

**ENCYCLOPEDIA OF
BIOINFORMATICS AND
COMPUTATIONAL BIOLOGY**

ENCYCLOPEDIA OF BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

EDITORS IN CHIEF

Shoba Ranganathan

Macquarie University, Sydney, NSW, Australia

Michael Gribskov

Purdue University, West Lafayette, IN, United States

Kenta Nakai

The University of Tokyo, Tokyo, Japan

Christian Schönbach

*Nazarbayev University, School of Science and Technology, Department of Biology,
Astana, Kazakhstan*

VOLUME 1

Methods

Mario Cannataro

The Magna Græcia University of Catanzaro, Catanzaro, Italy



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge MA 02139, United States

Copyright © 2019 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers may always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-12-811414-8

For information on all publications visit our
website at <http://store.elsevier.com>

		Working together to grow libraries in developing countries
www.elsevier.com • www.bookaid.org		

Publisher: Oliver Walter
Acquisition Editor: Sam Crowe
Content Project Manager: Paula Davies
Associate Content Project Manager: Ebin Clinton Rozario
Designer: Greg Harris

Printed and bound in the United States

EDITORS IN CHIEF



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. She hosted the Macquarie Node of the ARC Centre of Excellence in Bioinformatics (2008–2013). She was elected the first Australian Board Director of the International Society for Computational Biology (ISCB; 2003–2005); President of Asia-Pacific Bioinformatics Network (2005–2016) and Steering Committee Member (2007–2012) of Bioinformatics Australia. She initiated the Workshops on Education in Bioinformatics (WEB) as an ISMB2001 Special Interest Group meeting and also served as Chair of ICSB's Education Committee. Shoba currently serves as Co-Chair of the Computational Mass Spectrometry (CompMS) initiative of the Human Proteome Organization (HuPO), ISCB and Metabolomics Society and as Board Director, APBioNet Ltd.

Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books as well as articles for the 2013 Encyclopedia of Systems Biology. She is currently an Editor-in-Chief of the Encyclopedia of Bioinformatics and

Computational Biology and the Bioinformatics Section Editor of the Reference Module in Life Science as well as an editorial board member of several bioinformatics journals.



Dr. Gribskov graduated from Oregon State University in 1979 with a Bachelors of Science degree (with Honors) in Biochemistry and Biophysics. He then moved to the University of Wisconsin-Madison for graduate studies focused on the structure and function of the sigma subunit of *E. coli* RNA polymerase, receiving his Ph.D. in 1985. Dr. Gribskov studied X-ray crystallography as an American Cancer Society post-doctoral fellow at UCLA in the laboratory of David Eisenberg, and followed this with both crystallographic and computational studies at the National Cancer Institute. In 1992, Dr. Gribskov moved to the San Diego Supercomputer Center at the University of California, San Diego where he was lead scientist in the area of computational biology and an adjunct associate professor in the department of Biology. From 2003 to 2007, Dr. Gribskov was the president of the International Society for Computational Biology, the largest professional society devoted to bioinformatics and computational biology. In 2004, Dr. Gribskov moved to Purdue University where he holds an appointment as a full professor in the Biological Sciences and Computer Science departments (by courtesy). Dr. Gribskov's interests include genomic and transcriptomic analysis of model and non-model

organisms, the application of pattern recognition and machine learning techniques to biomolecules, the design and implementation of biological databases to support molecular and systems biology, development of methods to study RNA structural patterns, and systems biology studies of human disease.



Kenta Nakai received the PhD degree on the prediction of subcellular localization sites of proteins from Kyoto University in 1992. From 1989, he has worked at Kyoto University, National Institute of Basic Biology, and Osaka University. From 1999 to 2003, he was an Associate Professor at the Human Genome Center, the Institute of Medical Science, the University of Tokyo, Japan. Since 2003, he has been a full Professor at the same institute. His main research interest is to develop computational ways for interpreting biological information, especially that of transcriptional regulation, from genome sequence data. He has published more than 150 papers, some of which have been cited more than 1,000 times.



Christian Schönbach is currently Department Chair and Professor at Department of Biology, School of Science and Technology, Nazarbayev University, Kazakhstan and Visiting Professor at International Research Center for Medical Sciences at Kumamoto University, Japan. He is a bioinformatics practitioner interfacing genetics, immunology and informatics conducting research on major histocompatibility complex, immune responses following virus infection, biomedical knowledge discovery, peroxisomal diseases, and autism spectrum disorder that resulted in more than 80 publications. His previous academic appointments included Professor at Kumamoto University (2016–2017), Nazarbayev University (2013–2016), Kazakhstan, Kyushu Institute of Technology (2009–2013) Japan, Associate Professor at Nanyang Technological University (2006–2009), Singapore, and Team Leader at RIKEN Genomic Sciences Center (2002–2006), Japan. Other prior positions included Principal Investigator at Kent Ridge Digital Labs, Singapore and Research Scientist at Chugai Institute for Molecular Medicine, Inc., Japan. In 2018 he became a member of International Society for Computational Biology (ISCB) Board of Directors. Since 2010 he is serving Asia-Pacific Bioinformatics Network (APBioNet) as Vice-President (Conferences 2010–2016) and President (2016–2018).

VOLUME EDITORS



Mario Cannataro is a Full Professor of Computer Engineering and Bioinformatics at University “Magna Graecia” of Catanzaro, Italy. He is the director of the Data Analytics research center and the chair of the Bioinformatics Laboratory at University “Magna Graecia” of Catanzaro. His current research interests include bioinformatics, medical informatics, data analytics, parallel and distributed computing. He is a Member of the editorial boards of Briefings in Bioinformatics, High-Throughput, Encyclopedia of Bioinformatics and Computational Biology, Encyclopedia of Systems Biology. He was guest editor of several special issues on bioinformatics and he is serving as a program committee member of several conferences. He published three books and more than 200 papers in international journals and conference proceedings. Prof. Cannataro is a Senior Member of IEEE, ACM and BITS, and a member of the Board of Directors for ACM SIGBIO.



Bruno Gaeta is Senior Lecturer and Director of Studies in Bioinformatics in the School of Computer Science and Engineering at UNSW Australia. His research interests cover multiple areas of bioinformatics including gene regulation and protein structure, currently with a focus on the immune system, antibody genes and the generation of antibody diversity. He is a pioneer of bioinformatics education and has trained thousands of biologists and trainee bioinformaticians in the use of computational tools for biological research through courses, workshops as well as a book series. He has worked both in academia and in the bioinformatics industry, and currently coordinates the largest bioinformatics undergraduate program in Australia.



Mohammad Asif Khan, PhD, is an associate professor and the Dean of the School of Data Sciences, as well as the Director of the Centre for Bioinformatics at Perdana University, Malaysia. He is also a visiting scientist at the Department of Pharmacology and Molecular Sciences, Johns Hopkins University School of Medicine (JHUSOM), USA. His research interests are in the area of biological data warehousing and applications of bioinformatics to the study of immune responses, vaccines, inhibitory drugs, venom toxins, and disease biomarkers. He has published in these areas, been involved in the development of several novel bioinformatics methodologies, tools, and specialized databases, and currently has three patent applications granted. He has also led the curriculum development of a Postgraduate Diploma in Bioinformatics programme and an MSc (Bioinformatics) programme at Perdana University. He is an elected ExCo member of the Asia-Pacific Bioinformatics Network (APBioNET) since 2010 and is currently the President of Association for Medical and Bio-Informatics, Singapore (AMBIS). He has donned various important roles in the organization of many local and international bioinformatics conferences, meetings and workshops.

CONTENTS OF VOLUME 1

Editors in Chief	v
Volume Editors	vii
List of Contributors for Volume 1	xvii
Preface	xxi

VOLUME 1

Algorithms Foundations <i>Nadia Pisanti</i>	1
Techniques for Designing Bioinformatics Algorithms <i>Massimo Cafaro, Italo Epicoco, and Marco Pulimeno</i>	5
Algorithms for Strings and Sequences: Searching Motifs <i>Francesco Cauteruccio, Giorgio Terracina, and Domenico Ursino</i>	15
Algorithms for Strings and Sequences: Pairwise Alignment <i>Stefano Beretta</i>	22
Algorithms for Strings and Sequences: Multiple Alignment <i>Pietro H Guzzi</i>	30
Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins <i>Giuseppe Tradigo, Francesca Rondinelli, and Gianluca Pollastri</i>	32
Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins <i>Marco Wiltgen</i>	38
<i>Ab initio</i> Protein Structure Prediction <i>Rahul Kaushik, Ankita Singh, and B Jayaram</i>	62
Algorithms for Structure Comparison and Analysis: Docking <i>Giuseppe Tradigo, Francesca Rondinelli, and Gianluca Pollastri</i>	77
Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors <i>Lo Giudice Paolo and Domenico Ursino</i>	81
Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs <i>Paolo Lo Giudice and Domenico Ursino</i>	89
Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs <i>Clara Pizzuti and Simona E Rombo</i>	95
Algorithms for Graph and Network Analysis: Graph Alignment <i>Luigi Palopoli and Simona E Rombo</i>	102
Bioinformatics Data Models, Representation and Storage <i>Mariaconcetta Bilotta, Giuseppe Tradigo, and Pierangelo Veltri</i>	110
Data Storage and Representation <i>Antonella Guzzo</i>	117
Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing <i>Barbara Calabrese</i>	126

Standards and Models for Biological Data: Common Formats <i>Barbara Calabrese</i>	130
Standards and Models for Biological Data: FGED and HUPO <i>Barbara Calabrese</i>	137
Standards and Models for Biological Data: SBML <i>Giuseppe Agapito</i>	142
Standards and Models for Biological Data: BioPAX <i>Giuseppe Agapito</i>	147
Models for Computable Phenotyping <i>Alfredo Tirado-Ramos and Laura Manuel</i>	154
Computing for Bioinformatics <i>Mario Cannataro and Giuseppe Agapito</i>	160
Computing Languages for Bioinformatics: Perl <i>Giuseppe Agapito</i>	176
Computing Languages for Bioinformatics: BioPerl <i>Giuseppe Agapito</i>	187
Computing Languages for Bioinformatics: Python <i>Pietro H Guzzi</i>	195
Computing Languages for Bioinformatics: R <i>Marianna Milano</i>	199
Computing Languages for Bioinformatics: Java <i>Pietro H Guzzi</i>	206
Parallel Architectures for Bioinformatics <i>Ivan Merelli</i>	209
Models and Languages for High-Performance Computing <i>Domenico Talia</i>	215
MapReduce in Computational Biology Via Hadoop and Spark <i>Giuseppe Cattaneo, Raffaele Giancarlo, Umberto Ferraro Petrillo, and Gianluca Roscigno</i>	221
Infrastructure for High-Performance Computing: Grids and Grid Computing <i>Ivan Merelli</i>	230
Infrastructures for High-Performance Computing: Cloud Computing <i>Paolo Trunfio</i>	236
Infrastructures for High-Performance Computing: Cloud Infrastructures <i>Fabrizio Marozzo</i>	240
Infrastructures for High-Performance Computing: Cloud Computing Development Environments <i>Fabrizio Marozzo and Paolo Trunfio</i>	247
Cloud-Based Bioinformatics Tools <i>Barbara Calabrese</i>	252
Cloud-Based Bioinformatics Platforms <i>Barbara Calabrese</i>	257
Cloud-Based Molecular Modeling Systems <i>Barbara Calabrese</i>	261
The Challenge of Privacy in the Cloud <i>Francesco Buccafurri, Vincenzo De Angelis, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera</i>	265

Artificial Intelligence and Machine Learning in Bioinformatics <i>Kaitao Lai, Natalie Twine, Aidan O'Brien, Yi Guo, and Denis Bauer</i>	272
Artificial Intelligence <i>Francesco Scardcello</i>	287
Knowledge and Reasoning <i>Francesco Ricca and Giorgio Terracina</i>	294
Machine Learning in Bioinformatics <i>Jyotsna T Wassan, Haiying Wang, and Huiru Zheng</i>	300
Intelligent Agents and Environment <i>Alfredo Garro, Max Mühlhäuser, Andrea Tundis, Stefano Mariani, Andrea Omicini, and Giuseppe Vizzari</i>	309
Intelligent Agents: Multi-Agent Systems <i>Alfredo Garro, Max Mühlhäuser, Andrea Tundis, Matteo Baldoni, Cristina Baroglio, Federico Bergenti, and Paolo Torroni</i>	315
Stochastic Methods for Global Optimization and Problem Solving <i>Giovanni Stracquadanio and Panos M Pardalos</i>	321
Data Mining in Bioinformatics <i>Chiara Zucco</i>	328
Knowledge Discovery in Databases <i>Massimo Guarascio, Giuseppe Manco, and Ettore Ritacco</i>	336
Supervised Learning: Classification <i>Mauro Castelli, Leonardo Vanneschi, and Álvaro Rubio Largo</i>	342
Unsupervised Learning: Clustering <i>Angela Serra and Roberto Tagliaferri</i>	350
Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations <i>Massimo Cafaro, Italo Epicoco, and Marco Pulimeno</i>	358
Association Rules and Frequent Patterns <i>Giuseppe Di Fatta</i>	367
Decision Trees and Random Forests <i>Michele Fratello and Roberto Tagliaferri</i>	374
Data Mining: Classification and Prediction <i>Alfonso Urso, Antonino Fiannaca, Massimo La Rosa, Valentina Ravì, and Riccardo Rizzo</i>	384
Bayes' Theorem and Naive Bayes Classifier <i>Daniel Berrar</i>	403
Data Mining: Prediction Methods <i>Alfonso Urso, Antonino Fiannaca, Massimo La Rosa, Valentina Ravì, and Riccardo Rizzo</i>	413
Data Mining: Accuracy and Error Measures for Classification and Prediction <i>Paola Galdi and Roberto Tagliaferri</i>	431
Data Mining: Clustering <i>Alessia Amelio and Andrea Tagarelli</i>	437
Computation Cluster Validation in the Big Data Era <i>Raffaele Giancarlo and Filippo Utró</i>	449
Data Mining: Outlier Detection <i>Fabrizio Angiulli</i>	456

Pre-Processing: A Data Preparation Step <i>Swarup Roy, Pooja Sharma, Keshab Nath, Dhruba K Bhattacharyya, and Jugal K Kalita</i>	463
Data Cleaning <i>Barbara Calabrese</i>	472
Data Integration and Transformation <i>Barbara Calabrese</i>	477
Data Reduction <i>Barbara Calabrese</i>	480
Dimensionality Reduction <i>Italia De Feis</i>	486
Kernel Machines: Introduction <i>Italo Zoppis, Giancarlo Mauri, and Riccardo Dondi</i>	495
Kernel Methods: Support Vector Machines <i>Italo Zoppis, Giancarlo Mauri, and Riccardo Dondi</i>	503
Kernel Machines: Applications <i>Italo Zoppis, Giancarlo Mauri, and Riccardo Dondi</i>	511
Multiple Learners Combination: Introduction <i>Chiara Zucco</i>	519
Multiple Learners Combination: Bagging <i>Chiara Zucco</i>	525
Multiple Learners Combination: Boosting <i>Chiara Zucco</i>	531
Multiple Learners Combination: Stacking <i>Chiara Zucco</i>	536
Multiple Learners Combination: Cascading <i>Chiara Zucco</i>	539
Cross-Validation <i>Daniel Berrar</i>	542
Performance Measures for Binary Classification <i>Daniel Berrar</i>	546
Natural Language Processing Approaches in Bioinformatics <i>Xu Han and Chee K Kwoh</i>	561
Text Mining Basics in Bioinformatics <i>Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Mimmo Parente</i>	575
Data-Information-Concept Continuum From a Text Mining Perspective <i>Danilo Cavaliere, Sabrina Senatore, and Vincenzo Loia</i>	586
Text Mining for Bioinformatics Using Biomedical Literature <i>Andre Lamurias and Francisco M Couto</i>	602
Multilayer Perceptrons <i>Leonardo Vanneschi and Mauro Castelli</i>	612
Delta Rule and Backpropagation <i>Leonardo Vanneschi and Mauro Castelli</i>	621
Deep Learning <i>Massimo Guarascio, Giuseppe Manco, and Ettore Ritacco</i>	634

Introduction to Biostatistics <i>Antonella Iuliano and Monica Franzese</i>	648
Descriptive Statistics <i>Monica Franzese and Antonella Iuliano</i>	672
Measurements of Accuracy in Biostatistics <i>Haiying Wang, Jyotsna T Wassan, and Huiru Zheng</i>	685
Hypothesis Testing <i>Claudia Angelini</i>	691
Statistical Inference Techniques <i>Daniela De Canditiis</i>	698
Correlation Analysis <i>Monica Franzese and Antonella Iuliano</i>	706
Regression Analysis <i>Claudia Angelini</i>	722
Nonlinear Regression Models <i>Audrone Jakaitiene</i>	731
Parametric and Multivariate Methods <i>Luisa Cutillo</i>	738
Stochastic Processes <i>Maria Francesca Carfora</i>	747
Hidden Markov Models <i>Monica Franzese and Antonella Iuliano</i>	753
Linkage Disequilibrium <i>Barbara Calabrese</i>	763
Introduction to the Non-Parametric Bootstrap <i>Daniel Berrar</i>	766
Population-Based Sampling and Fragment-Based <i>De Novo</i> Protein Structure Prediction <i>David Simoncini and Kam YJ Zhang</i>	774
Ontology: Introduction <i>Gianluigi Greco, Marco Manna, and Francesco Ricca</i>	785
Ontology: Definition Languages <i>Valeria Fionda and Giuseppe Pirrò</i>	790
Ontology: Querying Languages and Development <i>Valeria Fionda and Giuseppe Pirrò</i>	800
Ontology in Bioinformatics <i>Pietro Hiram Guzzi</i>	809
Biological and Medical Ontologies: Introduction <i>Marco Masseroli</i>	813
Biological and Medical Ontologies: GO and GOA <i>Marco Masseroli</i>	823
Biological and Medical Ontologies: Protein Ontology (PRO) <i>Davide Chicco and Marco Masseroli</i>	832
Biological and Medical Ontologies: Disease Ontology (DO) <i>Anna Bernasconi and Marco Masseroli</i>	838

Biological and Medical Ontologies: Human Phenotype Ontology (HPO) <i>Anna Bernasconi and Marco Masseroli</i>	848
Biological and Medical Ontologies: Systems Biology Ontology (SBO) <i>Anna Bernasconi and Marco Masseroli</i>	858
Ontology-Based Annotation Methods <i>Pietro H Guzzi</i>	867
Semantic Similarity Definition <i>Francisco M Couto and Andre Lamurias</i>	870
Semantic Similarity Functions and Measures <i>Giuseppe Pirrò</i>	877
Tools for Semantic Analysis Based on Semantic Similarity <i>Marianna Milano</i>	889
Functional Enrichment Analysis Methods <i>Pietro H Guzzi</i>	896
Gene Prioritization Using Semantic Similarity <i>Erinija Pranckeviciene</i>	898
Gene Prioritization Tools <i>Marianna Milano</i>	907
Networks in Biology <i>Valeria Fionda</i>	915
Graph Theory and Definitions <i>Stefano Beretta, Luca Denti, and Marco Previtali</i>	922
Network Properties <i>Stefano Beretta, Luca Denti, and Marco Previtali</i>	928
Graph Isomorphism <i>Riccardo Dondi, Giancarlo Mauri, and Italo Zoppis</i>	933
Graph Algorithms <i>Riccardo Dondi, Giancarlo Mauri, and Italo Zoppis</i>	940
Network Centralities and Node Ranking <i>Raffaele Giancarlo, Daniele Greco, Francesco Landolina, and Simona E Rombo</i>	950
Network Topology <i>Giuseppe Manco, Ettore Ritacco, and Massimo Guarascio</i>	958
Network Models <i>Massimo Guarascio, Giuseppe Manco, and Ettore Ritacco</i>	968
Community Detection in Biological Networks <i>Marco Pellegrini</i>	978
Protein–Protein Interaction Databases <i>Max Kotlyar, Chiara Pastrello, Andrea EM Rossos, and Igor Jurisica</i>	988
Alignment of Protein-Protein Interaction Networks <i>Swarup Roy, Hazel N Manners, Ahed Elmsallati, and Jugal K Kalita</i>	997
Visualization of Biomedical Networks <i>Anne-Christin Hauschild, Chiara Pastrello, Andrea EM Rossos, and Igor Jurisica</i>	1016
Cluster Analysis of Biological Networks <i>Asuda Sharma, Hesham Ali, and Dario Gheri</i>	1036

Biological Pathways	
<i>Giuseppe Agapito</i>	1047
Biological Pathway Data Formats and Standards	
<i>Ramakanth C Venkata and Dario Gheri</i>	1063
Biological Pathway Analysis	
<i>Ramakanth Chirravuri Venkata and Dario Gheri</i>	1067
Two Decades of Biological Pathway Databases: Results and Challenges	
<i>Sara Rahmati, Chiara Pastrello, Andrea EM Rossos, and Igor Jurisica</i>	1071
Visualization of Biological Pathways	
<i>Giuseppe Agapito</i>	1085
Integrative Bioinformatics	
<i>Marco Masseroli</i>	1092
Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines	
<i>Maria T Di Martino and Pietro H Guzzi</i>	1099
Information Retrieval in Life Sciences	
<i>Pietro Cinaglia, Domenico Mirarchi, and Pierangelo Veltri</i>	1104

LIST OF CONTRIBUTORS FOR VOLUME 1

Giuseppe Agapito

*University "Magna Graecia" of Catanzaro,
Catanzaro, Italy*

Hesham Ali

*University of Nebraska at Omaha, Omaha, NE,
United States*

Alessia Amelio

University of Calabria, Rende, Italy

Claudia Angelini

*Istituto per le Applicazioni del Calcolo "M. Picone",
Napoli, Italy*

Fabrizio Angiulli

University of Calabria, Rende, Italy

Matteo Baldoni

University of Turin, Turin, Italy

Cristina Baroglio

University of Turin, Turin, Italy

Denis Bauer

CSIRO, North Ryde, NSW, Australia

Stefano Beretta

University of Milan-Biocca, Milan, Italy

Federico Bergenti

University of Parma, Parma, Italy

Anna Bernasconi

Politecnico di Milano, Milan, Italy

Daniel Berrar

Tokyo Institute of Technology, Tokyo, Japan

Dhruba K. Bhattacharyya

Tezpur University, Tezpur, India

Mariaconcetta Bilotta

*University of Catanzaro, Catanzaro, Italy; and Institute
S. Anna of Crotona, Crotona, Italy*

Francesco Buccafurri

University of Reggio Calabria, Italy

Massimo Cafaro

University of Salento, Lecce, Italy

Barbara Calabrese

*University "Magna Graecia" of Catanzaro,
Catanzaro, Italy*

Mario Cannataro

*University "Magna Graecia" of Catanzaro, Catanzaro,
Italy*

Maria Francesca Carfora

*Istituto per le Applicazioni del Calcolo CNR, Napoli,
Italy*

Mauro Castelli

*NOVA IMS, Universidade Nova de Lisboa, Lisboa,
Portugal*

Giuseppe Cattaneo

University of Salerno, Fisciano, Italy

Francesco Cauteruccio

University of Calabria, Rende, Italy

Danilo Cavaliere

Università degli Studi di Salerno, Fisciano, Italy

Davide Chicco

Princess Margaret Cancer Centre, Toronto, ON, Canada

Pietro Cinaglia

*Magna Graecia University of Catanzaro, Catanzaro,
Italy*

Francisco M. Couto

Universidade de Lisboa, Lisboa, Portugal

Luisa Cutillo

*University of Sheffield, Sheffield, United Kingdom; and
Parthenope University of Naples, Naples, Italy*

Vincenzo De Angelis

University of Reggio Calabria, Italy

Daniela De Canditiis

*Istituto per le Applicazioni del Calcolo "M. Picone",
Rome, Italy*

Italia De Feis

*Istituto per le Applicazioni del Calcolo CNR, Napoli,
Italy*

Carmen De Maio

University of Salerno, Fisciano, Italy

Luca Denti

University of Milan-Biocca, Milan, Italy

Giuseppe Di Fatta

University of Reading, Reading, United Kingdom

Maria T. Di Martino

*University "Magna Graecia" of Catanzaro, Catanzaro,
Italy*

Riccardo Dondi
University of Bergamo, Bergamo, Italy

Ahed Elmsallati
McKendree University, Lebanon, IL, United States

Italo Epicoco
University of Salento, Lecce, Italy

Giuseppe Fenza
University of Salerno, Fisciano, Italy

Antonino Fiannaca
Via Ugo La Malfa, Palermo, Italy

Valeria Fionda
University of Calabria, Rende, Italy

Monica Franzese
*Institute for Applied Mathematics "Mauro Picone",
Napoli, Italy*

Michele Fratello
DP Control, Salerno, Italy

Paola Galdi
University of Salerno, Fisciano, Italy

Alfredo Garro
University of Calabria, Rende, Italy

Dario Gherzi
*University of Nebraska at Omaha, Omaha, NE,
United States*

Raffaele Giancarlo
University of Palermo, Palermo, Italy

Gianluigi Greco
University of Calabria, Cosenza, Italy

Daniele Greco
University of Palermo, Palermo, Italy

Massimo Guarascio
ICAR-CNR, Rende, Italy

Yi Guo
Western Sydney University, Penrith, NSW, Australia

Pietro H. Guzzi
*University "Magna Graecia" of Catanzaro,
Catanzaro, Italy*

Antonella Guzzo
University of Calabria, Rende, Italy

Xu Han
Nanyang Technological University, Singapore

Anne-Christin Hauschild
Krembil Research Institute, Toronto, ON, Canada

Antonella Iuliano
*Institute for Applied Mathematics "Mauro Picone",
Napoli, Italy*

Audrone Jakaitiene
Vilnius University, Vilnius, Lithuania

B. Jayaram
IIT Delhi, New Delhi, India

Igor Jurisica
*University of Toronto, ON, Canada; and Slovak
Academy of Sciences, Bratislava, Slovakia*

Jugal K. Kalita
University of Colorado, Boulder, CO, United States

Rahul Kaushik
IIT Delhi, New Delhi, India

Max Kotlyar
University Health Network, Toronto, ON, Canada

Chee K. Kwoh
Nanyang Technological University, Singapore

Massimo La Rosa
Via Ugo La Malfa, Palermo, Italy

Kaitao Lai
CSIRO, North Ryde, NSW, Australia

Andre Lamurias
Universidade de Lisboa, Lisboa, Portugal

Francesco Landolina
University of Palermo, Palermo, Italy

Álvaro Rubio Largo
*NOVA IMS, Universidade Nova de Lisboa, Lisboa,
Portugal*

Gianluca Lax
University of Reggio Calabria, Italy

Paolo Lo Giudice
*University "Mediterranea" of Reggio Calabria, Reggio
Calabria, Italy*

Vincenzo Loia
University of Salerno, Fisciano, Italy

Max Mühlhäuser
*Darmstadt University of Technology, Darmstadt,
Germany*

Giuseppe Manco
ICAR-CNR, Rende, Italy

Marco Manna
University of Calabria, Cosenza, Italy

Hazel N. Manners
North-Eastern Hill University, Shillong, India

Laura Manuel
University of Texas Health at San Antonio, San Antonio, TX, United States

Stefano Mariani
University of Bologna, Bologna, Italy

Fabrizio Marozzo
University of Calabria, Rende, Italy

Marco Masseroli
Polytechnic University of Milan, Milan, Italy

Giancarlo Mauri
University of Milan-Biocca, Milan, Italy

Ivan Merelli
Institute for Biomedical Technologies (CNR), Milan, Italy; and National Research Council, Segrate, Italy

Marianna Milano
University of Catanzaro, Catanzaro, Italy

Domenico Mirarchi
Magna Graecia University of Catanzaro, Catanzaro, Italy

Keshab Nath
North-Eastern Hill University, Shillong, India

Serena Nicolazzo
University of Reggio Calabria, Italy

Antonino Nocera
University of Reggio Calabria, Italy

Aidan O'Brien
CSIRO, North Ryde, NSW, Australia

Andrea Omicini
University of Bologna, Bologna, Italy

Luigi Palopoli
Università della Calabria, Cosenza, Italy

Panos M.. Pardalos
University of Florida, Gainesville, FL, United States

Mimmo Parente
University of Salerno, Fisciano, Italy

Chiara Pastrello
Krembil Research Institute, Toronto, ON, Canada

Marco Pellegrini
Consiglio Nazionale delle Ricerche, Istituto di Informatica e Telematica, Pisa, Italy

Umberto Ferraro Petrillo
University of Rome "Sapienza", Rome, Italy

Giuseppe Pirrò
ICAR-CNR, Rende, Italy

Nadia Pisanti
University of Pisa, Pisa, Italy

Clara Pizzuti
Institute for High Performance Computing and Networking (ICAR), Cosenza, Italy

Gianluca Pollastri
University College Dublin, Dublin, Ireland

Erinija Pranckeviciene
Vilnius University, Vilnius, Lithuania

Marco Previtali
University of Milan-Biocca, Milan, Italy

Marco Pulimeno
University of Salento, Lecce, Italy

Sara Rahmati
University of Toronto, Toronto, ON, Canada; and Krembil Research Institute, Toronto, ON, Canada

Valentina Ravi
Via Ugo La Malfa, Palermo, Italy

Francesco Ricca
University of Calabria, Rende, Italy

Ettore Ritacco
ICAR-CNR, Rende, Italy

Riccardo Rizzo
ICAR-CNR, Rende, Italy

Simona E. Rombo
University of Palermo, Palermo, Italy

Francesca Rondinelli
Università degli Studi di Napoli Federico II, Napoli, Italy

Gianluca Roscigno
University of Salerno, Fisciano, Italy

Andrea E.M. Rossos
Krembil Research Institute, Toronto, ON, Canada

Swarup Roy
Sikkim University, Gangtok, India; and North-Eastern Hill University, Shillong, India

Francesco Scarcello
University of Calabria, Rende, Italy

Sabrina Senatore
Università degli Studi di Salerno, Fisciano, Italy

Angela Serra
University of Salerno, Salerno, Italy

Pooja Sharma
Tezpur University, Tezpur, India

Asuda Sharma
*University of Nebraska at Omaha, Omaha, NE,
 United States*

David Simoncini
*University of Toulouse, Toulouse, France; and
 RIKEN, Yokohama, Japan*

Ankita Singh
*IIT Delhi, New Delhi, India; and Banasthali
 Vidyapith, Banasthali, India*

Giovanni Stracquadanio
University of Essex, Colchester, United Kingdom

Andrea Tagarelli
University of Calabria, Rende, Italy

Roberto Tagliaferri
University of Salerno, Salerno, Italy

Domenico Talia
University of Calabria, Rende, Italy

Giorgio Terracina
University of Calabria, Rende, Italy

Alfredo Tirado-Ramos
*University of Texas Health at San Antonio, San
 Antonio, TX, United States*

Paolo Torroni
University of Bologna, Bologna, Italy

Giuseppe Tradigo
*University of Calabria, Rende, Italy; and University
 of Florida, Gainesville, United States*

Paolo Trunfio
University of Calabria, Rende, Italy

Andrea Tundis
*Darmstadt University of Technology, Darmstadt,
 Germany*

Natalie Twine
CSIRO, North Ryde, NSW, Australia

Domenico Ursino
*University "Mediterranea" of Reggio Calabria, Reggio
 Calabria, Italy*

Alfonso Urso
Via Ugo La Malfa, Palermo, Italy

Filippo Utro
*IBM Thomas J. Watson Research Center, Yorktown
 Heights, NY, United States*

Leonardo Vanneschi
*NOVA IMS, Universidade Nova de Lisboa, Lisboa,
 Portugal*

Pierangelo Veltri
*University "Magna Graecia" of Catanzaro, Catanzaro,
 Italy*

Ramakanth C. Venkata
*University of Nebraska at Omaha, Omaha, NE, United
 States*

Giuseppe Vizzari
University of Milano-Bicocca, Milan, Italy

Haiying Wang
*Ulster University, Newtonabbey, Northern Ireland,
 United Kingdom*

Jyotsna T. Wassan
*Ulster University, Newtonabbey, Northern Ireland,
 United Kingdom*

Marco Wiltgen
*Graz General Hospital and University Clinics, Graz,
 Austria*

Kam Y.J. Zhang
RIKEN, Yokohama, Japan

Huiru Zheng
*Ulster University, Newtonabbey, Northern Ireland,
 United Kingdom*

Italo Zoppis
University of Milan-Biocca, Milan, Italy

Chiara Zucco
*University "Magna Graecia" of Catanzaro, Catanzaro,
 Italy*

PREFACE

Bioinformatics and Computational Biology (BCB) combine elements of computer science, information technology, mathematics, statistics, and biotechnology, providing the methodology and *in silico* solutions to mine biological data and processes, for knowledge discovery. In the era of molecular diagnostics, targeted drug design and Big Data for personalized or even precision medicine, computational methods for data analysis are essential for biochemistry, biology, biotechnology, pharmacology, biomedical science, and mathematics and statistics. Bioinformatics and Computational Biology are essential for making sense of the molecular data from many modern high-throughput studies of mice and men, as well as key model organisms and pathogens. This Encyclopedia spans basics to cutting-edge methodologies, authored by leaders in the field, providing an invaluable resource to students as well as scientists, in academia and research institutes as well as biotechnology, biomedical and pharmaceutical industries.

Navigating the maze of confusing and often contradictory jargon combined with a plethora of software tools is often confusing for students and researchers alike. This comprehensive and unique resource provides up-to-date theory and application content to address molecular data analysis requirements, with precise definition of terminology, and lucid explanations by experts.

No single authoritative entity exists in this area, providing a comprehensive definition of the myriad of computer science, information technology, mathematics, statistics, and biotechnology terms used by scientists working in bioinformatics and computational biology. Current books available in this area as well as existing publications address parts of a problem or provide chapters on the topic, essentially addressing practicing bioinformaticists or computational biologists. Newcomers to this area depend on Google searches leading to published literature as well as several textbooks, to collect the relevant information.

Although curricula have been developed for Bioinformatics education for two decades now (Altman, 1998), offering education in bioinformatics continues to remain challenging from the multidisciplinary perspective, and is perhaps an NP-hard problem (Ranganathan, 2005). A minimum Bioinformatics skill set for university graduates has been suggested (Tan *et al.*, 2009). The Bioinformatics section of the Reference Module in Life Sciences (Ranganathan, 2017) commenced by addressing the paucity of a comprehensive reference book, leading to the development of this Encyclopedia. This compilation aims to fill the “gap” for readers with succinct and authoritative descriptions of current and cutting-edge bioinformatics areas, supplemented with the theoretical concepts underpinning these topics.

This Encyclopedia comprises three sections, covering Methods, Topics and Applications. The theoretical methodology underpinning BCB are described in the Methods section, with Topics covering traditional areas such as phylogeny, as well as more recent areas such as translational bioinformatics, cheminformatics and computational systems biology. Additionally, Applications will provide guidance for commonly asked “how to” questions on scientific areas described in the Topics section, using the methodology set out in the Methods section. Throughout this Encyclopedia, we have endeavored to keep the content as lucid as possible, making the text “... as simple as possible, but not simpler,” attributed to Albert Einstein. Comprehensive chapters provide overviews while details are provided by shorter, encyclopedic chapters.

During the planning phase of this Encyclopedia, the encouragement of Elsevier’s Priscilla Braglia and the constructive comments from no less than ten reviewers lead our small preliminary editorial team (Christian Schönbach, Kenta Nakai and myself) to embark on this massive project. We then welcomed one more Editor-in-Chief, Michael Gribskov and three section editors, Mario Cannataro, Bruno Gaeta and Asif Khan, whose toils have results in gathering most of the current content, with all editors reviewing the submissions. Throughout the production phase, we have received invaluable support and guidance as well as milestone reminders from Paula Davies, for which we remain extremely grateful.

Finally we would like to acknowledge all our authors, from around the world, who dedicated their valuable time to share their knowledge and expertise to provide educational guidance for our readers, as well as leave a lasting legacy of their work.

We hope the readers will enjoy this Encyclopedia as much as the editorial team have, in compiling this as an ABC of bioinformatics, suitable for naïve as well as experienced scientists and as an essential reference and invaluable teaching guide for students, post-doctoral scientists, senior scientists, academics in universities and research institutes as well as pharmaceutical, biomedical and biotechnological industries. Nobel laureate Walter Gilbert predicted in 1990 that “In the year 2020 you will be able to go into the drug store, have your DNA sequence read in an hour or so, and given back to you on a compact disk so you can analyze it.” While technology may have already arrived at this milestone, we are confident one of the readers of this Encyclopedia will be ready to extract valuable biological data by computational analysis, resulting in biomedical and therapeutic solutions, using bioinformatics to “measure” health for early diagnosis of “disease.”

References

- Altman, R.B., 1998. A curriculum for bioinformatics: the time is ripe. *Bioinformatics*. 14 (7), 549–550.
Ranganathan, S., 2005. Bioinformatics education—perspectives and challenges. *PLoS Comput Biol* 1 (6), e52.
Tan, T.W., Lim, S.J., Khan, A.M., Ranganathan, S., 2009. A proposed minimum skill set for university graduates to meet the informatics needs and challenges of the “-omics” era. *BMC Genomics*. 10 (Suppl 3), S36.
Ranganathan, S., 2017. *Bioinformatics. Reference Module in Life Sciences*. Oxford: Elsevier.

Shoba Ranganathan

Introduction

Biology offers a huge amount and variety of data to be processed. Such data has to be stored, analysed, compared, searched, classified, etcetera, feeding with new challenges many fields of computer science. Among them, algorithmics plays a special role in the analysis of biological sequences, structures, and networks. Indeed, especially due to the flood of data coming from sequencing projects as well as from its down-stream analysis, the size of digital biological data to be studied requires the design of very efficient algorithms. Moreover, biology has become, probably more than any other fundamental science, a great source of new algorithmic problems asking for accurate solutions. Nowadays, biologists more and more need to work with *in silico* data, and therefore it is important for them to understand why and how an algorithm works, in order to be confident in its results. The goal of this chapter is to give an overview of fundamentals of algorithms design and evaluation to a non-computer scientist.

Algorithms and Their Complexity

Computationally speaking, a *problem* is defined by an input/output relation: we are given an input, and we want to return as output a well defined solution which is a function of the input satisfying some property.

An *algorithm* is a computational procedure (described by means of an unambiguous sequence of instructions) that has to be executed in order to solve a computational problem. An algorithm solving a given problem is correct if it outputs the right result for every possible input. The algorithm has to be described according to the entity which will execute it: if this is a computer, then the algorithm will have to be written in a programming language.

Example: Sorting Problem

INPUT: A sequence S of n numbers $\langle a_1, a_2, \dots, a_n \rangle$.

OUTPUT: A permutation $\langle a'_1, a'_2, \dots, a'_n \rangle$ of S such that $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

Given a problem, there can be many algorithms that correctly solve it, but in general they will not all be equally efficient. The efficiency of an algorithm is a function of its input size.

For example, a solution for the sorting problem would be to generate all possible permutations of S and, per each one of them, check whether this is sorted. With this procedure, one needs to be lucky to find the right sorting fast, as there is an exponential (in n) number of such permutations and in the average case, as well as in the worst case, this algorithm would require a number of elementary operations (such as write a value in a memory cell, comparing two values, swapping two values, etcetera) which is exponential in the input size n . In this case, since the worst case cannot be excluded, we say that the algorithm has an exponential *time complexity*. In computer science, exponential algorithms are considered *intractable*. An algorithm is, instead, *tractable*, if its complexity function is polynomial in the input size. The *complexity of a problem* is that of the most efficient algorithm that solves it. Fortunately, the sorting problem is tractable, as there exist tractable solutions that we will describe later.

In order to evaluate the running time of an algorithm independently from the specific hardware on which it is executed, this is computed in terms of the amount of simple operations to which it is assigned an unitary cost or, however, a cost which is constant with respect to the input size. A constant running time is a negligible cost, as it does not grow when the input size does; moreover, a constant factor summed up with a higher degree polynomial in n is also negligible; furthermore, even a constant factor multiplying a higher polynomial is considered negligible in running time analysis. What counts is the growth factor with respect to the input size, i.e. the *asymptotic* complexity $T(n)$ as the input size n grows. In computational complexity theory, this is formalized using the *big-O* notation that excludes both coefficients and lower order terms: the asymptotic time complexity $T(n)$ of an algorithm is in $O(f(n))$ if there exist n_0 and $c > 0$ such that $T(n) \leq c f(n)$ for all $n \geq n_0$. For example, an algorithm that scans an input of size n a constant number of times, and then performs a constant number of some other operations, takes $O(n)$ time, and is said to have linear time complexity. An algorithm that takes linear time only in the worst case is also said to be in $O(n)$, because the big-O notation represents an upper bound. There is also an asymptotic complexity notation $\Omega(f(n))$ for the lower bound: $T(n) = \Omega(f(n))$ whenever $f(n) = O(T(n))$. A third notation $\Theta(f(n))$ denotes asymptotic equivalence: we write $T(n) = \Theta(f(n))$ if both $T(n) = O(f(n))$ and $f(n) = O(T(n))$ hold. For example, an algorithm that *always* performs a linear scan of the input, and not just in the worst case, has time complexity in $\Theta(n)$. Finally, an algorithm which needs to at least read, hence scan, the whole input of size n (and possibly also perform more costly tasks), has time complexity in $\Omega(n)$.

Time complexity is not the only cost parameter of an algorithm: *space complexity* is also relevant to evaluate its efficiency. For space complexity, computer scientists do not mean the size of the program describing an algorithm, but rather the data structures this actually keeps in memory during its execution. Like for time complexity, the concern is about how much memory the execution takes in the worst case and with respect to the input size. For example, an algorithm solving the sorting problem without

requiring any additional data structure (besides possibly a constant number of constant-size variables), would have linear space complexity. Also the exponential time complexity algorithm we described above has linear space complexity: at each step, it suffices to keep in memory only one permutation of S , as those previously attempted can be discarded. This observation offers an example of why, often, time complexity is of more concern than space complexity. The reason is not that space is less relevant than time, but rather that space complexity is in practice a lower bound of (and thus smaller than) time complexity: if an algorithm has to write and/or read a certain amount of data, then it forcibly has to perform at least that amount of elementary steps (Cormen *et al.*, 2009; Jones and Pevzner, 2004).

Iterative Algorithms

An *iterative algorithm* is an algorithm which repeats a same sequence of actions several times; the number of such times does not need to be known a priori, but it has to be finite. In programming languages, there are basically two kinds of iterative commands: the **for** command repeats the actions a number of times which is computed, or anyhow known, before the iterations begin; the **while** command, instead, performs the actions as long as a certain given condition is satisfied, and the number of times this will occur is not known a priori. What we call here an *action* is a command which can be, on its turn, again iterative. The cost of an iterative command is the cost of its actions multiplied by the number of iterations.

From now on, in this article we will describe an algorithm by means of the so-called *pseudocode*: an informal description of a real computer program, which is a mixture of natural language and keywords representing commands that are typical of programming languages. To this purpose, before exhibiting an example of an iterative algorithm for the sorting problem, we introduce the syntax of a fundamental elementary command: the assignment " $x \leftarrow E$ ", whose effect is to set the value of an expression E to the variable x , and whose time cost is constant, provided that computing the value of E , which can contain on its turn variables as well as calls of functions, is also constant. We will assume that the input sequence S of the sorting problem is given as an array: an array is a data structure of known fixed length that contains elements of the same type (in this case numbers). The i -th element of array S is denoted by $S[i]$, and reading or writing $S[i]$ takes constant time. Also swapping two values of the array takes constant time, and we will denote this as a single command in our pseudocode, even if in practice it will be implemented by a few operations that use a third temporary variable. What follows is the pseudocode of an algorithm that solves the sorting problem in polynomial time.

```

INSERTION-SORT( $S, n$ )
  for  $i = 1$  to  $n - 1$  do
     $j \leftarrow i$ 
    while ( $j > 0$  and  $S[j - 1] > S[j]$ )
      swap  $S[j]$  and  $S[j - 1]$ 
       $j \leftarrow j - 1$ 
    end while
  end for

```

INSERTION-SORT takes in input the array S and its size n . It works iteratively by inserting into the partially sorted S the elements one after the other. The array is indexed from 0 to $n - 1$, and a **for** command performs actions for each i in the interval $[1, n - 1]$ so that at the end of iteration i , the left end of the array up to its i -th position is sorted. This is realized by means of another iterative command, nested into the first one, that uses a second index j that starts from i , compares $S[j]$ (the new element) with its predecessor, and possibly swaps them so that $S[j]$ moves down towards its right position; then j is decreased and the task is repeated until $S[j]$ has reached its correct position; this inner iterative command is a **while** command because this task has to be performed as long as the predecessor of $S[j]$ is larger than it.

Example: Let us consider $S = [3, 2, 7, 1]$. Recall that arrays are indexed from position 0 (that is, $S[0] = 3$, $S[1] = 1$, and so on). INSERTION-SORT for $i = 1$ sets $j = 1$ as well, and then executes the while because $j = 1 > 0$ and $S[0] > S[1]$: these two values are swapped and j becomes 0 so that the while command ends with $S = [2, 3, 7, 1]$. Then a new **for** iteration starts with $i = 2$ (notice that at this time, correctly, S is sorted up to $S[1]$), and $S[2]$ is taken into account; this time the **while** command is entered with $j = 2$ and its condition is not satisfied (as $S[2] > S[1]$) so that the **while** immediately ends without changing S : the first three values of S are already sorted. Finally, the last **for** iteration with $i = 4$ will execute the **while** three times (that is, $n - 1$) swapping 1 with 7, then with 3, and finally with 2, leading to $S = [1, 2, 3, 7]$ which is the correct output.

INSERTION-SORT takes at least linear time (that is, its time complexity is in $\Omega(n)$) because all elements of S must be read, and indeed the **for** command is executed $\Theta(n)$ times: one per each array position from the second to the last. The invariant is that at the beginning of each such iteration, the array is sorted up to position $S[i - 1]$, and then the new value at $S[i]$ is processed. Each iteration of the **for**, besides the constant time (hence negligible) assignment $j \leftarrow i$, executes the **while** command. This latter checks its condition (in constant time) and, if the newly read element $S[j]$ is greater than, or equal to, $S[j - 1]$ (which is the largest of the so far sorted array), then it does nothing; else, it swaps $S[j]$ and $S[j - 1]$, decreases j , checks again the condition, and possibly repeats these actions, as long as either $S[j]$ finds its place after a smaller value, or it becomes the new first element of S as it is the smallest found so far. Therefore, the actions of the **while** command are never executed if the array is already sorted. This is the best case time complexity of INSERTION-SORT: linear in the input size n . The worst case is, instead, when the input array is sorted in

the reverse order: in this case, at each iteration i , the **while** command has to perform exactly i swaps to let $S[j]$ move down to the first position. Therefore, in this case, iteration i of the **for** takes i steps, and there are $n - 1$ such iterations for each $1 \leq i \leq n - 1$. Hence, the worst case running time is

$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \Theta(n^2)$$

As for space complexity, INSERTION-SORT works within the input array plus a constant number of temporary variables, and hence it has linear space complexity. Being n also a lower bound (the whole array must be stored), in this case the space complexity is optimal.

The algorithm we just described is an example of iterative algorithm that realises a quite intuitive sorting strategy; indeed, often this algorithm is explained as the way we would sort playing cards in one hand by using the other hand to iteratively insert each new card in its correct position. Iteration is powerful enough to achieve, for our sorting problem, a polynomial time – although almost trivial – solution; the time complexity of INSERTION-SORT cannot however be proved to be optimal as the lower bound for the sorting problem is not n^2 , but rather $n \cdot \log_2 n$ (result not proved here). In order to achieve $O(n \cdot \log_2 n)$ time complexity we need an even more powerful paradigm that we will introduce in next section.

Recursive Algorithms

A *recursive algorithm* is an algorithm which, among its commands, recursively calls itself on smaller instances: it splits the main problem into subproblems, recursively solves them and combines their solutions in order to build up the solution of the original problem. There is a fascinating mathematical foundation, that goes back to the arithmetic of Peano, and even further back to induction theory, for the conditions that guarantee correctness of a recursive algorithm. We will omit details of this involved mathematical framework. Surprisingly enough, for a computer this apparently very complex paradigm, is easy to implement by means of a simple data structure (the *stack*).

In order to show how powerful induction is, we will use again our Sorting Problem running example. Namely, we describe here the recursive MERGE-SORT algorithm which achieves $\Theta(n \cdot \log_2 n)$ time complexity, and is thus optimal. Basically, the algorithm MERGE-SORT splits the array into two halves, sorts them (by means of two recursive calls on as many sub-arrays of size $n/2$ each), and then merges the outcomes into a whole sorted array. The two recursive calls, on their turn, will recursively split again into subarrays of size $n/4$, and so on, until the base case (the already sorted sub-array of size 1) is reached. The merging procedure will be implemented by the function MERGE (pseudocode not shown) which takes in input the array and the starting and ending positions of its portions that contain the two contiguous sub-arrays to be merged. Recalling that the two half-arrays to be merged are sorted, MERGE simply uses two indices along them sliding from left to right, and, at each step: makes a comparison, writes the smallest, and increases the index of the sub-array which contained it. This is done until when both sub-arrays have been entirely written into the result.

```

MERGE-SORT(S,p,r)
  if p < r then
    q ← ⌊(p+r)/2⌋
    MERGE-SORT(S,p,q)
    MERGE-SORT(S,q+1,r)
    MERGE(S,p,q,r)
  end if

```

Given the need of calling the algorithm on different array fragments, the input parameters, besides S itself, will be the starting and ending position of the portion of array to be sorted. Therefore, the first call will be $\text{MERGE-SORT}(S, 0, n-1)$. Then the index q which splits S in two halves is computed, and the two so found subarrays are sorted by means of as many recursive calls; the two resulting sorted arrays of size $n/2$ are then fused by MERGE into the final result. The correctness of the recursion follows from the fact that the recursive call is done on a half-long array, and from the termination condition “ $p < r$ ”: if this holds, then the recursion goes on; else ($p = r$) there is nothing to do as the array has length 1 and it is sorted. Notice, indeed, that if S is not empty, then $p > r$ can never hold as q is computed such that $p \leq q < r$.

The algorithm MERGE-SORT has linear (hence optimal) space complexity as it only uses S itself plus a constant number of variables. The time complexity $T(n)$ of MERGE-SORT can be defined by the following recurrence relation:

$$T(n) = \begin{cases} \Theta(1) & \rightarrow n = 1 \\ 2 \cdot T(n/2) + \Theta(n) & \rightarrow n > 1 \end{cases}$$

because, with an input of size n , MERGE-SORT calls itself twice on arrays of size $n/2$, and then calls MERGE which takes, as we showed above, $\Theta(n)$ time.

We now show by induction on n that $T(n) = \Theta(n \cdot \log_2 n)$. The base case is simple: if $n = 1$ then S is already sorted and correctly MERGE-SORT does nothing and ends in $\Theta(1)$ time. If $n > 1$, assuming that $T(n') = \Theta(n' \cdot \log_2 n')$ holds for $n' < n$, then we have

$T(n) = 2 \cdot n/2 \cdot \log_2(n/2) + n = n(\log_2 n - \log_2 2 + 1)$, which is in $\Theta(n \cdot \log n)$. It follows that MERGE-SORT has optimal time complexity.

Closing Remarks

In this article we gave an overview of algorithms and their complexity, as well as of the complexity of a computational problem and how the latter should be stated. We also described two fundamental paradigms in algorithms design: iteration and recursion. We used as running example a specific problem (sorting an array of numbers) to exemplify definitions, describe algorithms using different strategies, and learning how to compute their complexity.

See also: Information Retrieval in Life Sciences

References

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. *Introduction to Algorithms*, third ed. Boston, MA: MIT Press.
Jones, N.C., Pevzner, P.A., 2004. *An Introduction to Bioinformatics Algorithms*. Boston, MA: MIT Press.

Further Reading

Mäkinen, V., Belazzougui, D., Cunial, F., 2015. *Genome-Scale Algorithm Design: Biological Sequence Analysis in the Era of High-Throughput Sequencing*. Cambridge: Cambridge University Press.

Biographical Sketch



Nadia Pisanti graduated cum laude in Computer Science at the University of Pisa in 1996. In 1998 she obtained a DEA degree at the University of Paris Est, and in 2002 a PhD in Informatics at the University of Pisa. She has been visiting fellow at the Pasteur Institute in Paris, ERCIM fellow at INRIA Rhone Alpes, research fellow at the University of Pisa, and CNRS post-doc at the University of Paris 13. Since 2006 she is with the Department of Computer Science of the University of Pisa. During the academic year 2012–2013 she was on sabbatical leave at Leiden University, and during that time she has been visiting fellow at CWI Amsterdam. Since 2015, she is part of the international INRIA team ERABLE. Her research interests fall in the field of Computational Biology and, in particular, in the design and application of efficient algorithms for the analysis of genomic data.

Techniques for Designing Bioinformatics Algorithms

Massimo Cafaro, Italo Epicoco, and Marco Pulimeno, University of Salento, Lecce, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

This article deals with design techniques for algorithms, a fundamental topic which deserves an entire book. Indeed, several books have been published, including Cormen *et al.* (2009), Kleinberg (2011), Knuth (1998), Kozen (1992), Levitin (2006), Manber (1989), Mehlhorn and Sanders (2010), Sedgewick and Wayne (2011), and Skiena (2010). Owing to space limits, we can not hope to provide an in-depth discussion and thorough treatment of each of the design techniques that shall be presented. Rather, we aim at providing a modern introduction that, without sacrificing formal rigour when needed, emphasizes the pro and cons of each design technique, putting it in context. The interested reader may refer to the provided bibliography to delve into this fascinating topic.

Informally, an algorithm is the essence of a computational procedure, and can be thought as a set of step-by-step instructions to transform the input into the output according to the problem's statement. The first algorithm known is the Euclidean algorithm for computing the greatest common divisor, circa 400–300 B.C. The modern study of algorithms dates back to the early 1960s, when the limited availability and resources of the first computers were compelling reasons for the users to strive to design efficient computer algorithms.

The systematic study of computer algorithms to solve literally thousands of problems in many different contexts had begun, with extensive progress made by a huge number of researchers active in this field. A large number of efficient algorithms were devised to solve different problems, and the availability of many correct algorithms for the same problem stimulated the theoretical analysis of algorithms.

Looking at the similarities among different algorithms designed to solve certain classes of problems, the researchers were able to abstract and infer general algorithm design techniques. We cover here the most common techniques in the design of sequential algorithms.

Exhaustive Search

We begin our discussion of design techniques for algorithms starting with *exhaustive search*, which is also known as the *brute force* approach. The technique, from a conceptual perspective, represents the simplest possible approach to solve a problem. It is a straightforward algorithmic approach which, in general, involves trying all of the possible candidate solutions to the problem being solved and returning the best one. The name exhaustive search is therefore strictly related to the modulus operand of the technique, which exhaustively examines and considers all of the possible candidate solutions. The actual number of solutions returned depends on the problem's statement.

For instance, consider the problem of determining all of the divisors of a natural number n . Exhaustive search solves the problem by trying one by one each integer x from 1 to n and verifying if x divides exactly n , i.e., if n modulo x returns a remainder equal to zero. Each x satisfying the problem's statement is outputted. Therefore, for this problem exhaustive search returns a set of solutions, according to the problem's statement.

However, it is worth noting here that the technique may also be used to solve other problems which admit one or more optimal solutions (e.g., the class of optimization problems). In this case, we are not usually concerned with determining all of the possible solutions, since we consider all of the solutions practically equivalent (from an optimality perspective with regard to the problem's statement). For these problems, exhaustive search consists of trying one by one all of the possible solutions and returning one of the satisfying candidate solutions, typically the first encountered. Once a solution is returned, remaining candidates (if any) are simply discarded from further consideration. Of course, if the problem admits exactly one solution, discarding the remaining candidates which can not be the solution allows avoiding a waste of time.

For instance, consider the sorting problem. We are given an input sequence $a_1, a_2, \dots, a_n$ of n elements, and must output a permutation $a_1^*, a_2^*, \dots, a_n^*$ such that $a_1^* \leq a_2^* \leq \dots \leq a_n^*$. One may try all of the possible permutations of the input sequence, stopping as soon as the one under consideration satisfies the output specification and can therefore be returned as the solution to the problem.

Exhaustive search is therefore a design technique characterized by its conceptual simplicity and by the assurance that, if a solution actually exists, it will be found. Nonetheless, enumerating all of the possible candidate solutions may be difficult or costly, and the cost of exhaustive search is proportional to the number of candidates. For instance, for the problem of determining all of the divisors of a natural number n , the number of candidates is n itself. The cost of exhaustive search for this problem depends on the actual number of bits required to store n and on the division algorithm used (it is worth recalling here that a division costs $O(1)$ only for sufficiently small n since we can not assume constant time arbitrary precision arithmetic when the size of n grows). Regarding the sorting problem, since there are $n!$ possible permutations of the input sequence, the worst case computational complexity is exponential in the input, making this approach to the problem unsuitable for large problems as well.

Since for many problems of practical interest a small increase in the problem size corresponds to a large increase in the number of candidates, the applicability of this technique is strictly confined to small size problems. Even though exhaustive search is often inefficient as an algorithmic design technique, it may be used as a useful complementary test to check that the results reported by other efficient algorithms - when run on small inputs - are indeed correct. Taking into account that exhaustive search is based on the enumeration of all of the possible candidate solutions, which are then checked one by one, in order to start applying the technique it is a useful discipline learning (by practice) how to identify the *structure* of a solution and how to rank a solution in order to select the best one.

A notable example of exhaustive search is the *linear search* algorithm for searching an element in an unsorted array (Knuth, 1998). A good example in bioinformatics is the so-called *restriction mapping problem* (Danna et al., 1973). Restriction enzyme mapping was a powerful tool in molecular biology for the analysis of DNA, long before the first bacterial genome was sequenced. Such a technique relied on restriction endonucleases, each one recognizing and reproducibly cleaving a specific base pair sequence in double-stranded DNA generating fragments of varying sizes. Determining the lengths of these DNA fragments is possible, taking into account that the rate at which a DNA molecule moves through an agarose gel during the electrophoresis process is inversely proportional to its size. Then, this information can be exploited to determine the positions of cleavage sites in a DNA molecule. Given only pairwise distances between a set of points, the restriction mapping problem requires recovering the positions of the points, i.e., in other words we are required to reconstruct the set of points.

Let X be a set of n points on a line segment in increasing order, and ΔX the multiset (i.e., a set that allows duplicate elements) of all pairwise distances between points in X : $\Delta X = \{x_j - x_i : 1 \leq i < j \leq n\}$. How to reconstruct X from ΔX ? We start noting here that the set of points giving rise to the pairwise input distances is not necessarily unique since the following properties hold:

$$\begin{aligned}\Delta A &= \Delta(A \oplus \{v\}) \\ \Delta A &= \Delta(-A) \\ \Delta(A \oplus B) &= \Delta(A \ominus B)\end{aligned}\tag{1}$$

where $A \oplus B \equiv \{a + b : a \in A, b \in B\}$ and $A \ominus B \equiv \{a - b : a \in A, b \in B\}$. More in general, two sets A and B are said to be *homometric* if $\Delta A = \Delta B$, and biologists are usually interested in retrieving all of the homometric sets.

Even though highly inefficient for large n , an exhaustive search algorithm for this problem is conceptually simple. Let L and n be respectively the input list of distances, and n the cardinality of X . The algorithm determines M , the maximum element in L and then for every set of $n - 2$ integers taken from L such that $0 < x_2 < \dots < x_{n-1} < M$, it forms $X = \{0, x_2, \dots, x_{n-1}, M\}$ and checks if $\Delta X = L$. Of course, the complexity of this algorithm is exponential in n . A better (slightly more practical) exhaustive search algorithm for this problem has been designed by Skiena in 1990 (Skiena et al., 1990) (it is an exponential algorithm as well). The first polynomial-time algorithm efficiently solving this problem was designed by Daurat et al. in 2002 (Daurat et al., 2002).

Decrease and Conquer

In order to solve a problem, *decrease and conquer* (Levitin, 2006) works by reducing the problem instance to a smaller instance of the same problem, solving the smaller instance and extending the solution of the smaller instance to obtain the solution to the original instance. Therefore, the technique is based on exploiting a relationship between a solution to a given instance of a problem and a solution to a smaller instance of the same problem. This kind of approach can be implemented either top-down (recursively) or bottom-up (iteratively), and it is also referred to as the *inductive* or *incremental approach*.

Depending on the problem, decrease and conquer can be characterized by how the problem instance is reduced to a smaller instance:

1. Decrease by a constant (usually by one);
2. Decrease by a constant factor (usually by half);
3. Variable-size decrease.

We point out here the similarity between decrease and conquer in which decrease is by a constant factor and *divide and conquer*.

Algorithms that fall into the first category (decrease by a constant) include for instance: insertion sort (Cormen et al., 2009), graph traversal algorithms (DFS and BFS) (Cormen et al., 2009), topological sorting (Cormen et al., 2009), algorithms for generating permutations and subsets (Knuth, 1998). Among the algorithms in the second category (decrease by a constant factor) we recall here exponentiation by squaring (Levitin, 2006), binary search (Knuth, 1998), the strictly related bisection method and the russian peasant multiplication (Levitin, 2006). Finally, examples of algorithms in the last category (variable-size decrease) are the Euclid's algorithm (Cormen et al., 2009), the selection algorithm (Cormen et al., 2009), and searching and insertion in a binary search tree (Cormen et al., 2009).

Insertion sort exemplifies the decrease by a constant approach (in this case, decrease by one). In order to sort an array A of length n , the algorithm assumes that the smaller problem related to sorting the subarray $A[1 \dots n - 1]$ consisting of the first $n - 1$ elements has been solved; therefore, $A[1 \dots n - 1]$ is a sorted subarray of size $n - 1$. Then, the problem reduces to finding the appropriate position (i.e., the index) for the element $A[n]$ within the sorted elements of $A[1 \dots n - 1]$, and inserting it. Even though this leads naturally to a recursive, top-down implementation, Insertion sort is often implemented iteratively using a bottom-up

approach instead: it is enough to start inserting the elements, one by one, from $A[2]$ to $A[n]$. Indeed, in the first iteration, $A[1]$ is already a sorted subarray, since an array consisting of just one element is already sorted. The worst-case complexity of Insertion sort is $O(n^2)$ to sort n elements; optimal sorting algorithms with worst-case complexity $O(n \lg n)$ are Merge sort and Heap sort.

Exponentiation by squaring is an example of an algorithm based on decrease by a constant factor (decrease by half). The algorithm is based on the following equation to compute a^n , which takes into account the parity of n :

$$a^n = \begin{cases} a^{n/2} \cdot a^{n/2} & \text{if } n \text{ is even and positive} \\ a^{(n-1)/2} \cdot a^{(n-1)/2} \cdot a & \text{if } n \text{ is odd} \\ 1 & \text{if } n = 0 \end{cases}$$

Therefore, a^n can be computed recursively by an efficient algorithm requiring $O(\lg n)$ iterations since the size of the problem is reduced in each iteration by about a half, even though at the expense of one or two multiplications.

The Euclid's algorithm for computing the greatest common divisor of two numbers m and n such that $m > n$ (otherwise, we simply swap m and n before starting the algorithm), provides an example of variable-size decrease. Denoting by $\gcd(m, n)$ the greatest common divisor of m and n , and by $m \bmod n$ the remainder of the division of m by n , the algorithm is based on repeated application of the following equation:

$$\gcd(m, n) = \gcd(n, m \bmod n)$$

until $m \bmod n = 0$. Since $\gcd(m, 0) = m$, the last value of m is also the greatest common divisor of the initial m and n . Measuring an instance size of the problem of determining $\gcd(m, n)$ by the size of m , it can be easily proved that an instance size will always decrease by at least a factor of two after two successive iterations of Euclid's algorithm. Moreover, a consecutive pair of Fibonacci numbers provides a worst-case input for the algorithm with regard to the total number of iterations required.

Transform and Conquer

A group of techniques, known as *transform and conquer* (Levitin, 2006), can be used to solve a problem by applying a *transformation*; in particular, given an input instance, we can transform it to:

1. a simpler or more suitable/convenient instance of the same problem, in which case we refer to the transformation as *instance simplification*;
2. a different representation of the same input instance, which is a technique also known in the literature as *representation change*;
3. a completely different problem, for which we already know an efficient algorithm; in this case, we refer to this technique as *problem reduction*.

As an example of instance simplification, we discuss *gaussian elimination*, in which we are given a system of n linear equations in n unknowns with an arbitrary coefficient matrix. We apply the technique and transform the input instance to an equivalent system of n linear equations in n unknowns with an upper triangular coefficient matrix. Finally, we solve the latter triangular system by back substitution, starting with the last equation and moving up to the first one. Another example is *element uniqueness*. We are given an input array consisting of n elements, and we want to determine if all of the elements are unique, i.e., there are no duplicate elements in the array. Applying the exhaustive search technique we could compare all pairs of elements in worst-case running time $O(n^2)$. However, by instance simplification we can solve the problem in $O(n \lg n)$ as follows. First, we sort the array in time $O(n \lg n)$ using Merge Sort or Heap Sort, then we perform a linear scan of the array, checking pairs of adjacent elements, in time $O(n)$. Overall, the running time is $O(n \lg n) + O(n) = O(n \lg n)$.

Heap sort (Williams, 1964) provides an excellent example of representation change. This sorting algorithm is based on the use of a *binary heap* data structure, and it can be shown that a binary heap corresponds to an array and vice-versa, if certain conditions are satisfied.

Regarding problem reduction, this variation of transform and conquer solves a problem by transforming it into a different problem for which an algorithm is already available. However, it is worth noting here that problem reduction is valuable and practical only when the sum of the time required by the transformation (i.e., the reduction) and the time required to solve the newly generated problem is smaller than solving the input problem by using another algorithm. Examples of problem reductions include:

- computing $\text{lcm}(x, y)$ via computing $\gcd(x, y)$: $\text{lcm}(x, y) = \frac{|xy|}{\gcd(x, y)}$
- counting the number of paths of length n in a graph by raising the graph's adjacency matrix to the n th power;
- transforming a linear programming maximization problem to a minimization problem and vice-versa;
- reduction to graph problems (e.g., solving puzzles via state-space graphs).

Divide and Conquer

Divide and conquer (from Latin *divide et impera*) is an important design technique and works as follows. When the input instance is too big or complex to be solved directly, it is advantageous to divide the input instance into two or more subproblems of roughly the same size, solve the subproblems (usually recursively, unless the subproblems are small enough to be solved directly) and finally combine the solutions to the subproblems to obtain the solution for the original input instance.

Merge sort, invented by John von Neumann in 1945, is a sorting algorithm based on divide and conquer. In order to sort an array A of length n , the algorithm divides the input array into two halves $A[1 \dots \lfloor n/2 \rfloor - 1]$ and $A[\lfloor n/2 \rfloor \dots n]$, sorts them recursively and then merges the resulting smaller sorted arrays into a single sorted one. The key point is how to merge two sorted arrays, which can be easily done in linear time as follows. We scan both arrays using two pointers, initialized to point to the first elements of the arrays we are going to merge. We compare the elements and copy the smaller to the new array under construction; then, the pointer to the smaller element is incremented so that it points to the immediate successor element in the array. We continue comparing pairs of elements, determining the smaller and copying it to the new array until one of the two input arrays becomes empty. When this happens, we simply add the remaining elements of the other input array to the merged array. Let p and q be respectively the sizes of the two input array to be merged, such that $n = p + q$. Then, the merge procedure requires in the worst case $O(n)$ time.

Recursive algorithms such as Merge sort are analyzed by deriving and solving a recurrence equation. Indeed, recursive calls in algorithms can be described using recurrences, i.e., equations or inequalities that describe a function in terms of its value on smaller inputs. For instance, the recurrence for Merge sort is:

$$T(n) = \begin{cases} O(1) & n = 1 \\ 2T(n/2) + O(n) & n > 1 \end{cases} \quad (2)$$

Actually, the correct equation should be

$$T(n) = \begin{cases} O(1) & n = 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + O(n) & n > 1 \end{cases} \quad (3)$$

but it can be shown that neglecting the floor and the ceil does not matter asymptotically.

There are many methods to solve recurrences. The most general method is the *substitution method*, in which we guess the form of the solution, verify it by induction and finally solve for the constants associated to the asymptotic notation. In order to guess the form of the solution, the *recursion-tree method* can be used; it models the cost (time) of a recursive execution of an algorithm. In the recursion tree each node represents a different substitution of the recurrence equation, so that each node corresponds to a value of the argument n of the function $T(n)$ associated with it. Moreover, each node q in the tree is also associated to the value of the nonrecursive part of the recurrence equation for q . In particular, for recurrences derived by a divide and conquer approach, the nonrecursive part is the one related to the work required to combine the solutions of the subproblems into the solution for the original problem, i.e., solutions related to the subproblems associated to the children of node q in the tree.

To generate the recursion tree, we start with $T(n)$ as the root node. Let the function $f(n)$ be the only nonrecursive term of the recurrence; we expand $T(n)$ and put $f(n)$ as the root of the recursion tree. We obtain the first level of the tree by expanding the recurrence, i.e. we put each of the recurrence terms involving the T function on the first level, and then we substitute them with the corresponding f terms. Then we proceed to expand the second level, substituting each T term with the corresponding f term. And so on, until we reach the leaves of the tree. To obtain an estimate of the solution to the recurrence, we sum the nonrecursive values across the levels of the tree and then sum the contribution of each level of the tree.

Equations of the form $T(n) = aT(n/b) + f(n)$, where $a \geq 1$, $b > 1$ and $f(n)$ is asymptotically positive can be solved immediately by applying the so-called *master theorem* (Cormen et al., 2009), in which we compare the function $f(n)$ with $n^{\log_b a}$. There are three cases to consider:

1. $\exists \epsilon > 0$ such that $f(n) = O(n^{\log_b a - \epsilon})$. In this case, $f(n)$ grows polynomially slower (by an n^ϵ factor) than $n^{\log_b a}$, and the solution is $T(n) = \Theta(n^{\log_b a})$;
2. $\exists k \geq 0$ such that $f(n) = \Theta(n^{\log_b a} \log^k n)$. Then, the asymptotic grow of both $f(n)$ and $n^{\log_b a}$ is similar, and the solution is $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$;
3. $f(n) = \Omega(n^{\log_b a + \epsilon})$ and $f(n)$ satisfies the regularity condition $af(n/b) \leq cf(n)$ for some constant $c < 1$. Then, $f(n)$ grows polynomially faster (by an n^ϵ factor) than $n^{\log_b a}$, and the solution is $T(n) = \Theta(f(n))$.

A more general method, devised by Akra and Bazzi (1998) allows solving recurrences of the form

$$T(n) = \sum_{i=1}^k a_i T(n/b_i) + f(n) \quad (4)$$

Let p be the unique solution to $\sum_{i=1}^k (a_i b_i^{-p}) = 1$; then the solution is derived exactly as in the master theorem, but considering n^p instead of $n^{\log_b a}$. Akra and Bazzi also prove an even more general result.

Many constant order linear recurrences are also easily solved by applying the following theorem.

Let $a_1, a_2, \dots, a_h \in \mathbb{N}$ and $h \in \mathbb{N}$, c and $\beta \in \mathbb{R}$ such that $c > 0$, $\beta \geq 0$ and let $a = \sum_{i=1}^h a_i$. Then, the solution to the recurrence

$$T(n) = \begin{cases} k \in \mathbb{N} & n \leq h \\ \sum_{i=1}^h a_i T(n-i) + cn^\beta & n > h \end{cases} \quad (5)$$

is

$$T(n) = \begin{cases} O(n^{\beta+1}) & a = 1 \\ O(a^n n^\beta) & a \geq 2 \end{cases} \quad (6)$$

Specific techniques for solving general constant order linear recurrences are also available.

Divide and conquer is a very powerful design technique, and for many problems it provides fast algorithms, including, for example, Merge sort, Quick sort (Hoare, 1962), binary search (Knuth, 1998), algorithms for powering a number (Levitin, 2006) and computing Fibonacci numbers (Gries and Levin, 1980), the Strassen's algorithm (Strassen, 1969) for matrix multiplication, the Karatsuba's algorithm (Karatsuba and Ofman, 1962) for multiplying two n bit numbers etc.

Since so many problems can be solved efficiently by divide and conquer, one can get the wrong impression that divide and conquer is always the best way to approach a new problem. However, this is of course not true, and the best algorithmic solution to a problem may be obtained by means of a very different approach.

As an example, consider the majority problem. Given an unsorted array of n elements, using only equality comparisons we want to find the majority element, i.e., the one which appears in the array more than $n/2$ times. An algorithm based on exhaustive search simply compares all of the possible pairs of elements and requires worst-case $O(n^2)$ running time. A divide and conquer approach provides an $O(n \log n)$ solution. However there exist an even better algorithm, requiring just a linear scan of the input array: the Boyer-Moore algorithm (Boyer and Moore, 1981, 1991) solves this problem in worst-case $O(n)$ time.

Randomized Algorithms

Randomized algorithms (Motwani and Raghavan, 2013) make random choices during the execution. In addition to its input, a randomized algorithm also uses a source of randomness:

- can flip coins as a basic step;
 - (i) can toss a fair coin c which is either Heads or Tails with probability $1/2$;
- can generate a random number r from a range $\{1 \dots R\}$;
 - (i) decisions and or computations are based on r 's value.

On the same input, on different executions, randomized algorithms may

- run for a different number of steps;
- produce different outputs.

Indeed, on different executions, different coins are flipped (different random numbers are used), and the value of these coins can change the course of executions. Why does it make sense to toss coins? Here are a few reasons. Some problems can not be solved deterministically at all; an example is the asynchronous agreement problem (consensus). For some other problems, only exponential deterministic algorithms are known, whereas polynomial-time randomized algorithms do exist. Finally, for some problems, a randomized algorithm provides a significant polynomial-time speedup with regard to a deterministic algorithm.

The intuition behind randomized algorithms is simple. Think of an algorithm as battling against an adversary who attempts to choose an input to slow it down as much as possible. If the algorithm is deterministic, then the adversary may analyze the algorithm and find an input that will elicit the worst-case behaviour. However, for a randomized algorithm the output does not depend only on the input, since it also depends on the random coins tossed. The adversary does not control and does not know which coins will be tossed during the execution, therefore his ability to choose an input which will elicit a worst-case running time is severely restricted. Where do we get coins from? In practice, randomized algorithms use pseudo random number generators.

Regarding the analysis of a randomized algorithm, this is different from average-case analysis, which requires knowledge of the distribution of the input and for which the expected running time is computed taking the expectation over the distribution of possible inputs. In particular, the running time of a randomized algorithm, being dependent on random bits, actually is a random variable, i.e., a function in a probability space Ω consisting of all of the possible sequences r , each of which is assigned a probability $Pr[r]$. The running time of a randomized algorithm A on input x and a sequence r of random bits, denoted by $A(x, r)$ is given by the expected value $E[A(x, r)]$, where the expectation is over r , the random choices of the algorithm: $E[A(x, r)] = \sum_{r \in \Omega} A(x, r) Pr[r]$.

There are two classes of randomized algorithms, which were originally named by Babai (1979).

- *Monte Carlo* algorithm: for every input, regardless of coins tossed, the algorithm always run in polynomial time, and the probability that its output is correct can be made arbitrarily high;

- *Las Vegas* algorithm: for every input, regardless of coins tossed, the algorithm is correct and it runs in expected polynomial time (for all except for a “small” number of executions, the algorithm runs in polynomial time).

The probabilities and expectations above are over the random choices of the algorithm, not over the input. As stated, a Monte Carlo algorithm fails with some probability, but we are not able to tell when it fails. A Las Vegas algorithm also fails with some probability, but we are able to tell when it fails. This allows us running it again until succeeding, which implies that the algorithm eventually succeeds with probability 1, even though at the expense of a potentially unbounded running time.

In bioinformatics, a good example of a Monte Carlo randomized algorithm is the *random projections* algorithm (Buhler and Tompa, 2001, 2002) for motif finding. Another common example of a Monte Carlo algorithm is the Freivald’s algorithm (Freivalds, 1977) for checking matrix multiplication. A classical example of Las Vegas randomized algorithm is Quick sort (Hoare, 1962), invented in 1962 by Charles Anthony Richard Hoare, which is also a divide and conquer algorithm. Even though the worst-case running time of Quick sort is $O(n^2)$, its expected running time is $O(n \lg n)$ as Merge sort and Heap sort. However, Quick sort is, in practice, much faster.

Dynamic Programming

The *Dynamic Programming* design technique provides a powerful approach to the solution of problems exhibiting (i) *optimal substructure* and (ii) *overlapping subproblems*. Property (i) (also known as *principle of optimality*) means that an optimal solution to the problem contains within it optimal solutions to related subproblems. Property (ii) tells us that the space of subproblems related to the problem we want to solve is small (typically, the number of distinct subproblems is a polynomial in the input size). In this context, a divide and conquer approach, which recursively solves all of the subproblems encountered in each recursive step, is clearly unsuitable and highly inefficient, since it will repeatedly solve all of the same subproblems whenever it encounters them again and again in the recursion tree. On the contrary, dynamic programming suggests solving each of the smaller subproblems only once and recording the results in a table from which a solution to the original problem can then be obtained.

Dynamic programming is often applied to optimization problems. Solving an optimization problem through dynamic programming requires finding an optimal solution, since there can be many possible solutions with the same optimal value (minimum or maximum, depending on the problem).

Computing the n th number of the Fibonacci series provides a simple example of application of dynamic programming (it is worth noting here that for this particular problem a faster divide and conquer algorithm, based on matrix exponentiation, actually exists). Denoting with $F(n)$ the n th Fibonacci number, it holds that $F(n) = F(n-1) + F(n-2)$. This problem is explicitly expressed as composition of subproblems, namely to compute the n th number we have to solve the same problem but with smaller instances $F(n-1)$ and $F(n-2)$. The divide and conquer approach would recursively compute all of the subproblems with a top-down approach, including also those subproblems already solved i.e. to compute $F(n)$ we have to compute $F(n-1)$ and $F(n-2)$; to compute $F(n-1)$ we have to compute again $F(n-2)$ and $F(n-3)$; in this example the subproblem $F(n-2)$ would be evaluated twice following the divide and conquer approach. Dynamic programming avoids recomputing the already solved subproblems. Typically dynamic programming follows a bottom-up approach, even though a recursive top-down approach with *memoization* is also possible (without memoizing the results of the smaller subproblems, the approach reverts to the classical divide and conquer).

As an additional example, we introduce the problem of *sequence alignment*. A common approach to infer a newly sequenced gene’s function is to find the similarities with genes of known function. Revealing the similarity between different DNA sequences is non trivial and comparing corresponding nucleotides is not enough; a sequence alignment is needed before comparison. Hirschberg’s space-efficient algorithm (Hirschberg, 1975) is a divide and conquer algorithm that can perform alignment in linear space (whilst the traditional dynamic programming approach requires quadratic space), even though at the expense of doubling the computational time.

The simplest form of a sequence similarity analysis is the Longest Common Subsequence (LCS) problem where only insertions and deletions between two sequences are allowed. We define a subsequence of a string v as an ordered sequence of characters, not necessarily consecutive, from v . For example, if $v = \text{ATTGCTA}$, then AGCA and ATTA are subsequences of v whereas TGTT and TCG are not. A common subsequence of two strings is a subsequence of both of them. The longer is a common substring between two strings, the more similar are the strings. We hence can formulate the Longest Common Substring problem as follows: given two input strings v and w , respectively of length n and m , find the longest subsequence common to the two strings.

Denoting with $s_{i,j}$ the longest common subsequence between the first i characters of v (denoted as i -prefix of v) and the first j characters of w (denoted as j -prefix of w), then the solution to the problem is $s_{n,m}$. We can solve the problem recursively noting that the following relation holds:

$$s_{i,j} = \begin{cases} s_{i-1,j-1} + 1 & \text{if } v_i = w_j \\ \max\{s_{i-1,j}, s_{i,j-1}\} & \text{if } v_i \neq w_j \end{cases} \quad (7)$$

Clearly, $s_{i,0} = s_{0,j} = 0 \forall 1 \leq i \leq n, 1 \leq j \leq m$. The first case corresponds to a match between v_i and w_j ; in this case, the solution for the subproblem $s_{i,j}$ is the solution for the subproblem $s_{i-1,j-1}$ plus one (since $v_i = w_j$ we can append v_i to the common substring we are building, increasing its length by one). The second case refers to a mismatch between v_i and w_j , giving rise to two possibilities: the

solution $s_{i-1,j}$ corresponds to the case in which v_i is not present in the LCS of the i -prefix of v and j -prefix of w , whilst the solution $s_{i,j-1}$ corresponds to the case when w_j is not present in LCS.

The problem has been expressed as composition of subinstances, moreover it can be easily proved that it meets the principle of optimality (i.e., if a string z is a LCS of v and w , then any prefix of z is a LCS of a prefix of v and w) and that the number of distinct LCS subproblems for two strings of lengths n and m is only nm . Hence the dynamic programming design technique can be applied to solve the problem.

In general, to apply dynamic programming, we have to address a number of issues:

1. Show optimal substructure, i.e. an optimal solution to the problem contains within it optimal solutions to subproblems; the solution to a problem is derived by:
 - making a choice out of a number of possibilities (look what possible choices there can be);
 - solving one or more subproblems that are the result of a choice (we need to characterize the space of subproblems);
 - show that solutions to subproblems must themselves be optimal for the whole solution to be optimal;
2. Write a recurrence equation for the value of an optimal solution:
 - $M_{opt} = \text{Min}$ (or Max , depending on the optimization problem) over all choices k of $\{(\text{sum of } M_{opt} \text{ of all of the subproblems resulting from choice } k) + (\text{the cost associated with making the choice } k)\}$;
 - show that the number of different instances of subproblems is bounded by a polynomial;
3. Compute the value of an optimal solution in a bottom-up fashion (alternatively, top-down with memoization);
4. Optionally, try to reduce the space requirements, by “forgetting” and discarding solutions to subproblems that will not be used any more;
5. Optionally, reconstruct an optimal solution from the computed information (which records a sequence of choices made that lead to an optimal solution).

Backtracking and Branch-and-Bound

Some problems require finding a feasible solution by exploring the solutions domain, which for these problems grows exponentially. For optimization problems we also require that the feasible solution is the best one according to an objective function. Many of these problems might not be solvable in polynomial time. In Section “Exhaustive Search” we discussed how such problems can be solved, in principle, by exhaustive search hence sweeping the whole solutions domain.

In this section we introduce the *Backtracking* and *Branch-and-Bound* design techniques which can be considered as an improvement of the exhaustive search approach. The main idea is to build the candidate solution to the problem adding one component at a time and evaluating the partial solution constructed so far. For optimization problems, we would also consider a way to estimate a bound on the best value of the objective function of any solution that can be obtained by adding further components to the partially constructed solution.

If the partial solution does not violate the problem constraints and its bound is better than the currently known feasible solution, then a new component can be added up to reach the final feasible solution. If during the construction of the solution no other component can be added either because it does not exist any feasible solution starting from the partially constructed solution or because its bound is worse than the currently known feasible solution, then the remaining components are not generated at all and the process backtracks, changing the previously added components. This approach makes it possible to solve some large instances of difficult combinatorial problems, though, in the worst case, we still face the same curse of exponential explosion encountered in exhaustive search.

Backtracking and Branch-and-Bound differ in the nature of problems they can be applied to. *Branch-and-Bound* is applicable only to optimization problems because it is based on computing a bound on possible values of the problem’s objective function. *Backtracking* is not constrained by this requirement and the partially built solution is pruned only if it violates the problem constraints.

Both methodologies require building the state-space tree whose nodes reflect the specific choices made for a solution’s components. Its root represents an initial state before the search for a solution begins. The nodes at the first level in the tree represent the choices made for the first component of a solution, and so on. A node in a state-space tree is said to be *promising* if it corresponds to a partially constructed solution that may still lead to a feasible solution; otherwise, it is called *nonpromising*. Leaves represent either *nonpromising* dead ends or complete solutions found by the algorithm.

We can better explain how to build the state-space tree by introducing the n -Queens problem. In the n -Queens problem we have to place n queens in an $n \times n$ chessboard so that no two queens attack each other. A queen may attack any chess piece if it is on the same row, column or diagonal. For this problem the *Backtracking* approach would bring valuable benefits with respect the exhaustive search. We know that only a queen must be placed in each row, we hence have to find the column where to place each queen so that the problem constraints are met.

A solution can be represented by n values $\{c_1, \dots, c_n\}$; where c_i represents the column of the i th queen. At the first level of the state-space tree we have n nodes representing all of the possible choices for c_1 . We make a choice for the first value of c_1 exploring

the first promising node and adding n nodes at second level corresponding to the available choices for c_2 . The partial solution made of c_1, c_2 choices is evaluated and the process continues visiting the tree in a depth-first manner. If all of the nodes on the current level are nonpromising, then the algorithm backtracks to the upper level up to the first promising node.

Several others problems can be solved by a backtracking approach. In the Subset-sum problem we have to find a subset of a given set $A = \{a_1, \dots, a_n\}$ of n positive integers whose sum is equal to a given positive integer d . The Hamiltonian circuit problem consists in finding a cyclic path that visits exactly once each vertex in a graph. In the n -Coloring problem we have to color all of the vertices in a graph such that no two adjacent vertices have the same color. Each vertex can be colored by using one of the n available colors. Subset-sum, Hamiltonian circuit and graph coloring are examples of NP-complete problems for which backtracking is a viable approach if the input instance is very small.

As a final example, we recall here the restriction map problem already described in Section “Exhaustive Search”. The restriction map problem is also known in computer science as Turnpike problem. The Turnpike problem is defined as follow: let X be a set of n points on a line $X = \{x_1, \dots, x_n\}$, given the ΔX multiset of the pairwise distances between each pair $\{x_i, x_j\}$, $\Delta X = \{x_j - x_i \mid i, j: 1 \leq i < j \leq n\}$, we have to reconstruct X .

Without loss of generality, we can assume that the first point in the line is at $x_1 = 0$. Let L be the input multiset with all of the distances between pairs of points; we have to find a solution X such that $\Delta X = L$. The key idea is to start considering the greatest distance in L ; let us denote it as δ_1 . We can state that the furthest point is at distance δ_1 , i.e. $x_n = \delta_1$. We remove δ_1 from L and consider the next highest distance δ_2 . This distance derives from two cases: $x_n - x_2 = \delta_2$ or $x_{n-1} - x_1 = \delta_2$; we can make an arbitrary choice and start building the state-space tree. Let us choose $x_2 = x_n - \delta_2$, we hence have a partial solution $\tilde{X} = \{0, \delta_1 - \delta_2, \delta_1\}$. In order to verify if this partial solution does not violate the constraints, we compute the $\Delta \tilde{X}$ and verify that $L \supset \Delta \tilde{X}$. If the constraint is satisfied, the node is a promising one and we can continue with the next point, otherwise we change the choice with the next promising node. The algorithm iterates until all of the feasible solutions are found. At each level of the state-space tree only two alternatives can be examined. Usually only one of the two alternatives is viable at any level. In this case the computational complexity of the algorithm can be expressed as:

$$T(n) = T(n-1) + O(n \log n) \quad (8)$$

being $O(n \log n)$ the time taken for checking the partial solution. In this case the computational complexity is $T(n) = O(n^2 \log n)$.

In the worst case both alternatives must be evaluated at each level; in this case the recurrence equation is:

$$T(n) = 2T(n-1) + O(n \log n) \quad (9)$$

whose solution is $T(n) = O(2^n n \log n)$. The algorithm remains an exponential time algorithm in the worst case like the one based on exhaustive search, but, usually, the backtracking approach greatly improves the computational time by pruning the non-promising branches. We recall finally that Daurat *et al.* in 2002 (Daurat *et al.*, 2002) proposed a polynomial algorithm to solve the restriction map problem.

In the majority of the cases, a state-space tree for a backtracking algorithm is constructed in a depth-first search manner, whilst Branch-and-Bound usually explores the state-space tree by using a best-first rule i.e. the node with the best bound is explored first. Compared to Backtracking, Branch-and-Bound requires two additional items:

- a way to provide, for every node of a state-space tree, a bound on the best value of the objective function on any solution that can be obtained by adding further components to the partially constructed solution represented by the node;
- the value of the best solution seen so far.

If the bound value is not better than the value of the best solution seen so far the node is *nonpromising* and can be pruned. Indeed, no solution obtained from it can yield a better solution than the one already available. Some of the most studied problems faced with the Branch-and-Bound approach are: the Assignment Problem in which we want to assign n people to n jobs so that the total cost of the assignment is as small as possible; in the Knapsack problem we have n items with weights w_i and values v_i , a knapsack of capacity W , and the problem consist in finding the most valuable subset of the items that fits in the knapsack; in the Traveling Salesman Problem (TSP) we have to find the shortest possible route that visits each city exactly once and returns to the origin city knowing the distances between each pair of cities. Assignment, Knapsack and TSP are examples of NP-complete problems for which Branch-and-Bound is a viable approach, if the input instance is very small.

Greedy Algorithms

The Greedy design technique defines a simple methodology related to the exploration of the solutions domain of optimization problems. The greedy approach suggests constructing a solution through a sequence of steps, each expanding a partially constructed solution obtained so far, until a complete solution to the problem is reached. Few main aspects make the greedy approach different from Branch-and-Bound. First, in the greedy approach no bound must be associated to the partial solution; second, the choice made at each step is irrevocable hence backtracking is not allowed in the greedy approach. During the construction of the solution, on each step the choice made must be:

- feasible: the partial solution has to met the problem's constraints;
- locally optimal: it has to be the best local choice among all of the feasible choices available on that step;

- irrevocable.

The Greedy approach is based on the hope that a sequence of locally optimal choices will yield an optimal solution to the entire problem. There are problems for which a sequence of locally optimal choices does yield an optimal solution for every input instance of the problem. However, there are others for which this is not the case and the greedy approach can provide a misleading solution. As an example, let us consider the change-making problem. Given a set of coins with decreasing value $C = \{c_i : c_i > c_{i+1} \forall i = 1, \dots, n\}$ and a total amount T , find the minimum number of coins to reach the total amount T . The solution is represented by a sequence of n occurrences of the corresponding coins. A greedy approach to the problem considers on step i the coin c_i and chooses its occurrences as the maximum possible subject to the constraint that the total amount accumulated so far must not exceed T . Let us suppose that we can use the following coins values $C = \{50, 20, 10, 5, 2, 1\}$ and that we have to change $T = 48$; a greedy approach suggests choosing on the first stage no coins of value 50, on the second step 2 coins of value 20 since this is the best choice to quickly reach the total amount T and so on until building the solution $S = \{0, 2, 0, 1, 1, 1\}$.

Greedy algorithms are both intuitively appealing and simple. Given an optimization problem, it is usually easy to figure out how to proceed in a greedy manner. What is usually more difficult is to prove that a greedy algorithm yields an optimal solution for all of the instances of the problem. The greedy approach applied to the change-making example given above provides optimal solution for any value of T , but what happens if the coins values are $C = \{25, 10, 1\}$ and the amount is $T = 40$? In this case following a greedy approach the solution would be $S = \{1, 1, 5\}$ but the best solution is instead $S = \{0, 4, 0\}$. Therefore, proving that the solution given by the greedy algorithm is optimal becomes a crucial aspect.

One of the common ways to do this is through mathematical induction, where we must prove that a partially constructed solution obtained by the greedy algorithm on each iteration can be extended to an optimal solution to the problem. The second way to prove optimality is to show that on each step it does at least as well as any other algorithm could in advancing toward the problem's goal. The third way is simply to show that the final result obtained is optimal based on the algorithm's output rather than on the way it operates. Finally, if a problem underlying combinatorial structure is a *matroid* (Cormen et al., 2009), then it is well known that the greedy approach leads to an optimal solution. The *matroid* mathematical structure has been introduced by Whitney in 1935; his *matric matroid* abstracts and generalizes the notion of linear independence.

In bioinformatic, one of the most challenging problem which can be solved through a greedy approach is genome rearrangement. Every genome rearrangement results in a change of gene ordering, and a series of these rearrangements can alter the genomic architecture of a species. The elementary rearrangement event is the flipping of a genomic segment, called a *reversal*. Biologists are interested in the smallest number of reversals between genomes of two species since it gives us a lower bound on the number of rearrangements that have occurred and indicates the similarity between two species.

In their simplest form, rearrangement events can be modelled by a series of reversals that transform one genome into another. Given a permutation $\pi = \pi_1 \pi_2 \dots \pi_{n-1} \pi_n$, a reversal $\rho(i, j)$ has the effect of reversing the order of block from i th to j th element $\pi_i \pi_{i+1} \dots \pi_{j-1} \pi_j$. For example the reversal $\rho(3, 5)$ of the permutation $\pi = 654298$ produces the new permutation $\pi \cdot \rho(3, 5) = 659248$. The Reversal Distance Problem can be formulated as follows: given two permutations π and σ , find the shortest series of reversals $\rho_1 \cdot \rho_2 \dots \rho_t$ that transforms π into σ . Without losing generality, we can consider the target permutation σ the ascending order of the elements. In this case the problem is also known as Sorting by Reversal. When sorting a permutation it hardly makes sense to move the elements already sorted. Denoting by $p(\pi)$ the number of already sorted elements of π , then a greedy strategy for sorting by reversals is to increase $p(\pi)$ at every step. Unfortunately, this approach does not guarantee that the solution is optimal. As an example we can consider $\pi = 51234$; following the greedy strategy we need four reversals for sorting π : $\{\rho(1,2), \rho(2,3), \rho(3,4), \rho(4,5)\}$ but we can easily see that two reversals are enough for sorting the permutation: $\{\rho(1,5), \rho(1,4)\}$.

Conclusions

We have presented a survey of the most important algorithmic design techniques, highlighting their pro and cons, and putting them in context. Even though, owing to space limits, we sacrificed in-depth discussion and thorough treatment of each technique, we hope to have provided the interested readers with just enough information to fully understand and appreciate the differences among the techniques.

See also: Algorithms Foundations

References

- Akra, M., Bazzi, L., 1998. On the solution of linear recurrence equations. *Comput. Optim. Appl.* 10 (2), 195–210. Available at: <http://dx.doi.org/10.1023/A:1018353700639>.
- Babai, L., 1979. Monte-carlo algorithms in graph isomorphism testing. Technical Report D.M.S. 79-10, Université de Montréal.
- Boyer, R., Moore, J., 1981. Mjty – A fast majority vote algorithm. Technical Report 32, Institute for Computing Science, University of Texas, Austin.
- Boyer, R., Moore, J.S., 1991. Mjty – A fast majority vote algorithm. In: *Automated Reasoning: Essays in Honor of Woody Bledsoe*, Automated Reasoning Series. Dordrecht, The Netherlands: Kluwer Academic Publishers, pp. 105–117.

- Buhler, J., Tompa, M., 2001. Finding motifs using random projections. In: Proceedings of the 5th Annual International Conference on Computational Biology. RECOMB '01. ACM, New York, NY, USA, pp. 69–76. Available at: <http://doi.acm.org/10.1145/369133.369172>.
- Buhler, J., Tompa, M., 2002. Finding motifs using random projections. *J. Comput. Biol.* 9 (2), 225–242. Available at: <http://dx.doi.org/10.1089/10665270252935430>.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to Algorithms, third ed. The MIT Press.
- Danna, K., Sack, G., Nathans, D., 1973. Studies of simian virus 40 dna. vii. a cleavage map of the sv40 genome. *J.Mol. Biol.* 78 (2), 1–10.
- Daurat, A., Gérard, Y., Nivat, M., 2002. The chords' problem. *Theor. Comput. Sci.* 282 (2), 319–336. Available at: [http://dx.doi.org/10.1016/S0304-3975\(01\)00073-1](http://dx.doi.org/10.1016/S0304-3975(01)00073-1).
- Freivalds, R., 1977. Probabilistic machines can use less running time. In: IFIP Congress. pp. 839–842.
- Gries, D., Levin, G., 1980. Computing Fibonacci numbers (and similarly defined functions) in log time. *Inform. Process. Lett.* 11 (2), 68–69.
- Hirschberg, D.S., 1975. A linear space algorithm for computing maximal common subsequences. *Commun. ACM* 18 (6), 341–343. Available at: <http://doi.acm.org/10.1145/360825.360861>.
- Hoare, C.A.R., 1962. Quicksort. *Comput. J.* 5 (1), 10–15.
- Karatsuba, A., Ofman, Y., 1962. Multiplication of many-digit numbers by automatic computers. *Dokl. Akad. Nauk SSSR* 145, 293–294. [Translation in *Physics-Doklady* 7, 595–596, 1963].
- Kleinberg, J., 2011. Algorithm Design, second ed. Addison-Wesley Professional.
- Knuth, D.E., 1998. The Art of Computer Programming, vol. 1–3, Boxed Set, second ed. Boston, MA: Addison-Wesley Longman Publishing Co., Inc..
- Kozen, D.C., 1992. The Design and Analysis of Algorithms. New York, NY: Springer-Verlag.
- Levitin, A.V., 2006. Introduction to the Design and Analysis of Algorithms, second ed. Boston, MA: Addison-Wesley Longman Publishing Co., Inc..
- Manber, U., 1989. Introduction to Algorithms: A Creative Approach. Boston, MA: Addison-Wesley Longman Publishing Co., Inc..
- Mehlhorn, K., Sanders, P., 2010. Algorithms and Data Structures: The Basic Toolbox, first ed. Berlin: Springer.
- Motwani, R., Raghavan, P., 2013. Randomized Algorithms. New York, NY: Cambridge University Press.
- Sedgwick, R., Wayne, K., 2011. Algorithms, fourth ed. Addison-Wesley Professional.
- Skiena, S.S., 2010. The Algorithm Design Manual, second ed. Berlin: Springer Publishing Company.
- Skiena, S.S., Smith, W.D., Lemke, P., 1990. Reconstructing sets from interpoint distances (extended abstract). In: Proceedings of the 6th Annual Symposium on Computational Geometry. SCG '90. ACM, New York, NY, pp. 332–339. Available at: <http://doi.acm.org/10.1145/98524.98598>.
- Strassen, V., 1969. Gaussian elimination is not optimal. *Numerische Mathematik* 13 (4), 354–356.
- Williams, J.W.J., 1964. Heapsort. *Commun. ACM* 7 (6), 347–348.

Algorithms for Strings and Sequences: Searching Motifs

Francesco Cauteruccio and Giorgio Terracina, University of Calabria, Rende, Italy

Domenico Ursino, University "Mediterranea" of Reggio Calabria, Reggio Calabria, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Ordered sequences of symbols (also called strings or sequences) play an important role in computer science. With an adequate semantics, they can be used to express several kinds of data. Data provided as sequences are constantly increasing in several areas of bioinformatics and computer science; think, for instance, of DNAs and proteins sequenced from organisms, but also at sensor networks, wearable devices, distributed agents, etc. Several general purpose techniques for string comparison have been proposed in the past literature.

In the specific context of bioinformatics, information encoded in biological sequences assumes an important role in identifying genetic-related diseases, and resulted useful for deciphering biological mechanisms. A gene is a sequence of DNA bases, used to generate proteins. The transformation from genes to proteins is based on transcription and gene expression mechanisms. Gene expression starts with one or more proteins, called transcription factors, binding to *transcription factor binding sites*; these are specific regions generally located before the gene sequence start. In fact, proteins may enhance or inhibit the transcription of a gene into a protein.

Regulation of gene expression, via the activation or inhibition of these transcription mechanisms, is a complex task, which is still under investigation by several researchers. However, it is well known that transcription factor binding sites are encoded as biologically significant patterns, called *motifs*, which occur in, or among, sequences frequently. As a matter of facts, researchers found that these are (usually small) sequences, which frequently occur in the same genome and are well conserved through evolution (see, for instance, [Pavesi et al., 2004](#); [Sandve and Drablos, 2006](#); [Sinha, 2002](#); [GuhaThakurta and Stormo, 2001](#)).

Several motif discovery approaches have been developed in the literature. These researches are strongly motivated by the availability of massive amounts of raw data resulting from the sequencing of the human genome, as well as of the genomes of other organisms. They significantly contributed to the growth of bioinformatics (see [Sandve et al., 2007](#); [Sandve and Drablos, 2006](#); [Pavesi et al., 2004](#), just to cite few papers).

Generally speaking, given a set of sequences, a motif discovery tool aims at identifying new, previously unknown, over-represented patterns; intuitively, these patterns should be common to all, or almost all, the sequences in the set. Motif discovery is not only used to detect transcription factor binding sites. As a matter of facts, the identification of over-represented patterns, through a set of protein sequences, might suggest that those proteins belong to the same family.

The search of frequent patterns has been also studied in other contexts, such as the analysis of time series (see [Torkamani and Lohweg, 2017](#)). However, the analysis of these approaches is out of the scope of this article.

The plan of this article is as follows. In Section Preliminary Definitions, we provide some preliminary notions related to motif discovery. In Section Approaches to Motif Search, we present an overview of motif search methods. In Section Assessing Motif Relevance, we illustrate the problems related to the assessment of motif relevance. Finally, in Section Closing Remarks, we provide some concluding remarks and we look at possible future directions of this research area.

Preliminary Definitions

Strings and Sequences

We start this section by introducing some notations and definitions, generally used in algorithms for motif discovery in strings and sequences.

An *alphabet* is a nonempty set (Σ) of elements called *symbols* (or *letters*).

A string s is an (finite) ordered sequence of symbols defined over (Σ), i.e., s consists of symbols of (Σ). We use the terms *strings* and *sequences* equivalently. For the sake of simplification, we represent a string by means of the juxtaposition of the symbols it consists of. For instance, suppose $(\Sigma) = \{a, b, c, d\}$, a string s , defined over (Σ), is $s = aabdbcdada$. The *length* of s is the number of symbols composing s , and is denoted by $|s|$. If $|s| = 0$, we call s the *empty string* and we denote it as $s = \varepsilon$. In biological contexts, relevant alphabets correspond to the four symbols representing DNA bases (A, C, G, T) and to the 20 symbols referring to aminoacids. We denote by $s[i]$, $1 \leq i \leq |s|$, the i -th symbol of s , i.e., the symbol at the position i in s .

A substring of s is a string that can be derived from s by deleting some symbols at the beginning and/or at the end of s . More formally, a string p is a *substring* (also called *factor*) of a string s whether $s = upw$, where u and w are two strings. If $u = \varepsilon$, p is called a *prefix* of s , whereas if $w = \varepsilon$, p is called a *suffix* of s . We denote by $s[i..j]$ the substring of s starting at index i and ending at index j . Let p be a nonempty string and let s be a string; we say that p has an *occurrence* in s (or, equivalently, that p *occurs in* s) if p is a substring of s .

A *subsequence* of s is a sequence that can be derived from s by deleting some symbols, without changing the order of the other symbols. Given two strings s and p , the Longest Common Subsequence between s and p (LCS) is the longest subsequence common to s and p .

Matches and Repetitions

An important problem on strings (strictly related to motif search) is that of *string matching*. Essentially, given two strings s and p , where s is called *text* and p is called *pattern*, string matching problem asks whether there exists an occurrence of p in s . This problem has been studied intensively in the literature, and we can distinguish between two main variants: *exact string matching* and *approximate string matching*. The former asks whether an exact occurrence of p can be found in s : here, “exact” means that $p = s[i..i + |p| - 1]$. Instead, the latter asks whether there exists a substring in s , which matches p via some matching function. Examples of simple matching functions are the Hamming distance, which counts the number of positions the two strings differ in, or the Longest Common Subsequence (LCS). In case of approximate matching with matching functions, some threshold is usually set (such as a maximum Hamming distance, or a minimum length of the LCS) to determine whether the match holds or not.

Approximate string matching can be carried out also by *string matching with don't care symbols*. In this case, occurrences of a special universal symbol $*$, called *don't care symbol*, can be included in the pattern p . A $*$ can match with any other (sequence of) symbol. Note that, in this context, the pattern becomes more complex, and harder computational issues may arise for determining the occurrences of p in s . For instance, given $s = \text{aabcdcd}$ and $p = a*c$, p occurs in s twice, one by substituting $*$ with bd and one by substituting $*$ with abd . Observe that, a don't care symbol may appear more than once in a pattern (e.g., $p = a*c*d$ occurs in s). A don't care symbol at the beginning and/or at the end of a pattern simply means that one may expect anything before and/or after the pattern; consequently, in pattern matching, don't care symbols in these positions are often omitted.

In some cases, in counting the number of occurrences of p in s , it may be important to distinguish between overlapping and non overlapping occurrences.

Motifs

Strings and matches are building blocks for the motif search problem (also called motif discovery). In bioinformatics, a motif is a biologically relevant pattern occurring in one or more input sequences; the set of sequences of interest for the motif search problem is also called sequence database. Two important research branches in motif discovery refer to *simple* and *structured* motifs. A simple motif is a single, usually short, sequence that significantly matches (either exactly or approximately) with the sequence database. A structured motif is a combination of simple motifs (called *boxes* or *blocks*), whose position and spacing (called also *gaps*) are relevant, and which are separated from one another by some regions not conserved in the evolution (Gross et al., 1992; Robin et al., 2003; Marsan and Sagot, 2000; Fassetti et al., 2008). Generally, the structures of motifs are *constrained*, which means that the minimum/maximum motif lengths, the box numbers, and the relative spacings are imposed.

In general, discovery methods are designed to identify motifs conforming to some model, which can capture the similarities of diverse sets of binding sites for the same transcription factor (see, for instance, Sandve et al., 2007; Pavese et al., 2004). Once the model is specified, it acts as a template that completely characterizes the search space for compatible motifs. Such a model is usually fixed by the biologist, who is willing to corroborate her hypothesis on the co-regulation of some given genes.

In the context of simple motifs, models can be fixed by the biologist in terms of Position Weighted Matrices (PWM) (Hughes et al., 2000), or IUPAC strings (Sinha and Tompa, 2003a,b), or consensus strings (Marsan and Sagot, 2000). These have been adopted by earlier approaches that analyzed the genome of simple prokaryotic organisms. However, when moving to eukaryotic transcription, things may become more complicated, and more complex motif templates have been considered in the literature (Werner, 1999; Sinha, 2002; Chen et al., 2005; Osanai et al., 2004; Tu et al., 2004). Indeed, conserved regions may consist of more than two short subsequences (in many cases, each one up to about 12 bases (Pavese et al., 2004)), and each consecutive pair of boxes might be separated by some specific gap (Werner, 1999; Sinha, 2002).

It is worth observing that, with the inclusion of “don't care” symbols in motif models, the distinction between simple and structured motifs is blurred, and constraining motif structures and/or matching functions for approximate matching is harder.

Approaches to Motif Search

In the literature outside bioinformatics, several algorithms for the discovery of frequent patterns have been proposed. As an example, the basic problem of finding frequent substrings has been deeply studied in the market-basket analysis. Here, sequences represent purchase transactions, and several algorithms to find frequent sequences have been proposed (see, for instance, Agrawal and Srikant, 1995; Zaki, 2001; Pei et al., 2001; Ayres et al., 2002). These approaches can be applied to general cases, where the alphabet is not fixed.

When dealing with the genomic domain and, in general, with the bioinformatics context, a plain adaptation of these approaches is generally unpractical (Wang et al., 2004; Ester and Zhang, 2004). For this reason, several specialized algorithms for motif search have been presented in the biological domain. These differ on several aspects, and an overview of all their typologies might be anyway incomplete.

Some authors (e.g., Brazma et al., 1998a; Das and Dai, 2007) tried to classify motif search approaches based on: (1) the underlying algorithms; (2) the kinds of derived motif; (3) the kinds of considered match. In the following, we examine these three taxonomies, one per subsection.

Classifying Motif Search Approaches Based on the Underlying Algorithms

Most literature on motif search categorizes the corresponding approaches by considering the underlying algorithm. In this case, it is possible to recognize three main categories, namely string-based, model-based, and phylogeny-based approaches. They are examined in more detail in the next subsections.

String-based approaches

String-based approaches rely on an exhaustive enumeration of potential motif candidates, followed by a counting of their occurrences. This kind of approach guarantees the exploration of the whole search space and, consequently, optimality. Obviously, they are appropriate only for short motifs. As a consequence, from a bioinformatics point of view, the kinds of subject under examination may significantly influence the possibility to adopt them. For instance, motifs in eukaryotic genomes are usually shorter than motifs in prokaryotes.

Optimized searches are achieved by applying advanced data structures, like suffix trees (Sagot, 1998), which, however, allow a perfect speedup for exact matches only.

String-based methods are also particularly suited for fixed structure motifs; in fact, when motifs have weakly constrained positions, or when their structure is not fixed, derived results may need some post-processing and refinement (Vilo *et al.*, 2000).

These methods guarantee global optimality, since they guarantee the generation and testing of all potential motifs. However, a relevant issue of them is that they may generate several irrelevant motifs; as a consequence, the validation phase is crucial, and it may become computationally expensive.

A slight variant of these approaches performs the enumeration of potential motifs actually expressed in the sequence database. This variant allows the reduction of the number of candidates. However, it may miss some motifs, when approximate matches are taken into account. As a matter of facts, it may happen that a true motif is actually not explicitly mentioned in the sequences, but it is represented by small variants of it.

A large (even if not exhaustive) variety of string-based approaches can be found in Bucher (1990), Bailey and Elkan (1995), Vilo *et al.* (2000), Van Helden *et al.* (2000), Tompa (1999), Sinha and Tompa (2000), (2003a,b), Mewes *et al.* (2002), Brzma *et al.* (1998b), Vanet *et al.* (2000), Carvalho *et al.* (2006), Pavesi *et al.* (2001), Eskin and Pevzner (2002), Pevzner and Sze (2000), Liang *et al.* (2004), and Fassetti *et al.* (2008).

Model-based approaches

Model-based approaches, also called probabilistic approaches, usually employ a representation of motif models by means of a position weight matrix (Bucher, 1990). Here, each motif position is weighed by the frequency of the corresponding symbols.

These models can be graphically represented by staking letters over each position, where the dimension of a letter is proportional to its information content in that position.

Probabilistic methods are often based on several forms of local search, such as Gibbs sampling and Expectation Maximization (EM), or on greedy algorithms that may converge to local optima. As a consequence, they cannot guarantee that the derived solution is globally optimal.

Model-based approaches are well suited to find long or loosely constrained motifs. Therefore, they can be useful for motif search in prokaryotes, where motifs are generally longer than the ones of eukaryotes.

A large (even if not exhaustive) variety of model-based approaches can be found in Hertz and Stormo (1999), Down and Hubbard, 2005, Liu (2008), Liu *et al.* (1995), (2001), Hughes *et al.* (2000), Thijs *et al.* (2002), Shida (2006), Buhler and Tompa (2002), and Kim *et al.* (1994).

Phylogeny-based approaches

Phylogeny-based approaches try to overcome the fact that classic motif search approaches consider input sequences independent from each other, i.e., they do not consider the possible phylogenetic relationships existing among input sequences. As a matter of facts, since sequence databases often contain data from closely related species, the choice of motifs to report should take this information into account.

One of the most important advantages of phylogenetic footprinting approaches is that they allow the identification of even single gene-specific motifs, if they have been conserved through sequences. However, one crucial task in these approaches is the choice of correlated sequences.

A naive method for searching phylogeny-based motifs consists in constructing a global multiple alignment of input sequences, which can be, then, used to identify conserved regions by means of well assessed tools, such as CLUSTALW (Thompson *et al.*, 1994).

However, this approach may fail in some cases; indeed, if the species are too correlated, non functional elements, along with functional ones, are conserved. On the contrary, if the sequence set is poorly correlated, it might be impossible to properly align sequences. To overcome this problem, some algorithms adapt standard motif search approaches, like Gibbs sampling, by including two important factors capable of measuring motif significance, namely over-representation and cross-species conservation. Some representative approaches belonging to this category are Carmack *et al.* (2007), Wang and Stormo (2003), Sinha *et al.* (2004), Siddharthan *et al.* (2005), Zhang *et al.* (2009, 2010), and Nettling *et al.* (2017).

Classifying Motif Search Approaches Based on the Kinds of Derived Motif

Taking this taxonomy into consideration, and based on what we have seen in Section Motifs, it is possible to distinguish approaches to extracting simple motifs from the ones to searching structured motifs.

Approaches to searching simple motifs

Simple motif extraction has been extensively studied in the literature. A former survey, which also introduces a formal framework to categorize patterns and algorithms, is presented in [Brazma et al. \(1998a\)](#). A more recent survey on this topic can be found in [Das and Dai \(2007\)](#). Among the most famous and best performing approaches in this context, we find MEME ([Bailey and Elkan, 1995](#)), CONSENSUS ([Hertz and Stormo, 1999](#)), Gibbs sampling ([Neuwald et al., 1995](#)), random projections ([Buhler and Tompa, 2002](#)) and MULTIPROFILER ([Keich and Pevzner, 2002](#)).

Approaches to searching structured motifs

When moving to structured motifs (also called composite motifs in the literature), the variety of existing approaches increases significantly. The simplest forms of these approaches are the ones for deriving “spaced dyads” (i.e., pairs of oligonucleotides at fixed distances from one another), or approaches for searching motifs composed of three boxes separated by a fixed length ([Van Helden et al., 2000](#); [Smith et al., 1990](#)). These approaches enumerate all possible patterns over the underlying alphabet, coherently with the chosen structure. Clearly, approaches like these are limited to the search of small motifs.

Some approaches for loosely structured motif extraction first derive simple motifs and then try to combine them for obtaining composite motifs. Other approaches try to obtain structured motifs directly. As an example, in [Eskin and Pevzner \(2002\)](#), an approach called MITRA is presented. It consists of two steps. The former pre-processes input data to obtain a new (bigger) input sequence by combining portions of simple motifs into virtual monads, which represent potentially structured motifs. The latter applies an exhaustive simple motif discovery algorithm to virtual monads for detecting patterns repeated significantly. Selected virtual monads are, then, decomposed back into structured patterns.

Other approaches, like SMILE ([Marsan and Sagot, 2000](#)), RISO ([Carvalho et al., 2006](#)), and L-SME ([Fassetti et al., 2008](#)), exploit efficient data structures, like suffix trees, factor trees, and other variants, for both explicitly representing structured motif candidates and efficiently counting their occurrences. In order to improve performances, [Fassetti et al. \(2008\)](#) introduces a randomized variant of the algorithm, based on sketches, which efficiently estimates the number of approximate occurrences of loosely structured patterns.

Considered motif structures may vary in several ways. In these cases, the concepts of boxes, skips, and swaps ([Marsan and Sagot, 2000](#); [Fassetti et al., 2008](#)) have been introduced as possible alternatives. However, also other classical structures, like tandem repeats or palindromes, received some interest.

Classifying Motif Search Approaches Based on the Kinds of Adopted Match

Taking this taxonomy into account, and based on what we have seen in Section Matches And Repetitions, it is possible to distinguish approaches based on exact matches from the ones adopting approximate matches. As a matter of facts, only former motif extraction approaches ([Van Helden et al., 2000](#); [Smith et al., 1990](#); [Brazma et al., 1998a,b](#); [Jonassen et al., 1995](#)), or approaches allowing arbitrary motif structures ([Wang et al., 2004](#)), consider exact matches. Indeed, since exception is the rule in biology, at least some approximation level is necessary.

The simplest approaches adopting approximate matches employ *degenerate alphabets* ([Brazma et al., 1998a](#)). These alphabets consider the fact that input symbols may be grouped in different classes, based on their meaning. For instance, aminoacids are either hydrophobic, neutral, or hydrophilic; as a consequence, they can be partitioned and mapped onto a three-symbol alphabet. A fully degenerate alphabet may include partially overlapping groups, where one symbol may be assigned to more than one class. Examples of approaches allowing degenerate alphabets can be found in [Neuwald and Green \(1994\)](#) and [Brazma et al. \(1998a,b\)](#).

More general motif search approaches adopting approximate matchings rely on some form of matching functions. One of the most common matching functions adopted by this kind of approach is the Hamming distance for its low computational cost and simplicity ([Neuwald and Green, 1994](#); [Eskin and Pevzner, 2002](#); [Marsan and Sagot, 2000](#); [Carvalho et al., 2006](#); [Fassetti et al., 2008](#)). Only few approaches consider the application of the Levenshtein distance (see, for instance, [Fassetti et al., 2008](#)), but at the price of a higher computational complexity.

Assessing Motif Relevance

Now, taking into account what we have seen in Section Motifs, given a pattern occurring in the sequence database, to determine whether it is a motif or not, it is necessary to assess its *relevance*. To carry out this task, the frequency of the pattern occurrences is clearly not enough. Indeed, there are patterns that, most probably, will be really frequent. As an extreme example, consider simple motifs composed of one symbol. Obviously, there is no surprise that all single DNA bases will be very frequent in a (portion of) genome. On the other side, a relatively high frequency of a very long pattern may be worth of deeper studies.

Unfortunately, there is still no widely accepted measure for assessing the relevance of a pattern yet, and, often, different fitness measures may result in quite different motif sets, returned by motif search approaches. Furthermore, the decision is

usually strongly dependent on some empirically chosen threshold. For all these reasons, assessing motif relevance is still an open issue.

An interesting attempt in this direction is described in [Apostolico et al. \(2003\)](#), where the authors introduce the “theory of surprise” to characterize unusual words. One of the most basic, and yet one of the most used, score functions for measuring the unexpectedness of a pattern is the z-score. This function is based on the (suitably normalized) difference between observed and expected pattern counts. Variants of the z-score differ for the normalization method and for the estimation of the number of expected patterns. For instance, this number might be obtained as an average on random strings or by a probabilistic estimation. However, counting pattern occurrences on random sequences might be computationally heavy. Other approaches are based on Expectation Maximization (EM) algorithms. In this case, motifs are incrementally enriched until to some maximal interest score is registered, and further enrichments would lead to a decrease of this score. Some recent approaches employ Markov chain models for functional motif characterization and evaluation (see, for instance, [Wang and Tapan, 2012](#)).

In [Tomba et al. \(2005\)](#), the authors use several statistic parameters (like sensitivity, specificity, etc.) to assess the performance quality of existing tools. However, in this case, some background knowledge on the real motifs, expected from the sequence database, is required. This knowledge could be based on information like the number of true/false positives, the number of motifs predicted in the right positions, etc. As a consequence, these statistics are more useful for an ex-post tool evaluation, rather than for guiding motif search. Interestingly, this analysis showed quite a poor performance of the state-of-the-art tools on real data sets. The reasons underlying this fact are presumably related to the poor estimation of the statistical relevance of returned motifs with respect to the actual biological relevance, and to the fact that motif search is actually based only on sequences.

As a final consideration about this issue, we evidence the popularity of several private and public databases, which report experimentally assessed and, in some cases, statistically predicted, motifs for transcription factor binding sites in different organisms. Some of these databases are TRANSFAC ([Matys et al., 2003](#)), JASPAR ([Sandelin et al., 2004](#); [Vlieghe et al., 2006](#)), SCPD ([Zhu and Zhang, 1999](#)), TRRD ([Kolchanov et al., 2000](#)), TRED ([Zhao et al., 2005](#)), and ABS ([Blanco et al., 2006](#)).

Closing Remarks

In this article, we have provided a presentation of algorithms for motif search. We have seen that this problem is very relevant in bioinformatics, since information encoded in biological sequences allow the identification of genetic-related diseases and resulted useful for deciphering biological mechanisms. Initially, we have defined the concepts of string, sequence, match, repetition and motif. Then, we have presented three possible taxonomies of approaches to motif search and, based on them, we have provided a description of these approaches. Finally, we have illustrated the issue of motif relevance assessment.

Research on motif discovery is moving in several directions. Several approaches are focusing on the application of existing computer science solutions to specific biological domains, such as the detection of transcription factors of specific genes. In this research area, biological insights on the analyzed organisms are strongly exploited to trim the number and quality of returned motifs. Other approaches are focusing on the reduction of the number of returned motifs; this objective is obtained by detecting closed sets of motifs (like non-redundant motifs or bases of motifs). Finally, since significant research efforts in bioinformatics are moving to the analysis of regulatory networks, a significant amount of research on motifs is shifting from the analysis of sequences to the search of motifs in networks.

See also: Biological Database Searching. Identification of Proteins from Proteomic Analysis. Mapping the Environmental Microbiome. Structure-Based Drug Design Workflow

References

- Agrawal, R., Srikant, R., 1995. Mining sequential patterns. In: *Proceedings of ICDE'95*, pp. 3–14.
- Apostolico, A., Bock, M.E., Lonardi, S., 2003. Monotony of surprise and large-scale quest for unusual words. *Journal of Computational Biology* 10 (3–4), 283–311.
- Ayres, J., Flannick, J., Gehrke, J., Yiu, T., 2002. Sequential pattern mining using a bitmap representation. In: *Proceedings of KDD'02*, pp. 429–435.
- Bailey, T.L., Elkan, C., 1995. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning* 21 (1), 51–80.
- Blanco, E., Farre, D., Alba, M.M., Messeguer, X., Guigo, R., 2006. ABS: A database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acids Research* 34 (Suppl. 1), D63–D67.
- Brzma, A., Jonassen, I., Eidhammer, I., Gilbert, D., 1998a. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology* 5 (2), 277–304.
- Brzma, A., Jonassen, I., Vilo, J., Ukkonen, E., 1998b. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research* 8 (11), 1202–1215.
- Bucher, P., 1990. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *Journal of Molecular Biology* 212 (4), 563–578.
- Buhler, J., Tomba, M., 2002. Finding motifs using random projections. *Journal of Computational Biology* 9 (2), 225–242.
- Carmack, C.S., McCue, L.A., Newberg, L.A., Lawrence, C.E., 2007. PhyloScan: Identification of transcription factor binding sites using cross-species evidence. *Algorithms for Molecular Biology* 2, 1.
- Carvalho, A.M., Freitas, A.T., Oliveira, A.L., Sagot, M.F., 2006. An efficient algorithm for the identification of structured motifs in DNA promoter sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3 (2), 126140.

- Chen, J.M., Chuzhanova, N., Stenson, P.D., Ferec, C., Cooper, D.N., 2005. Meta-analysis of gross insertions causing human genetic disease: Novel mutational mechanisms and the role of replication slippage. *Human Mutation* 25 (2), 207–221.
- Das, M.K., Dai, H.K., 2007. A survey of DNA motif finding algorithms. *BMC Bioinformatics* 8 (7), S21.
- Down, T.A., Hubbard, T.J., 2005. NestedMICA: Sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Research* 33 (5), 1445–1453.
- Eskin, E., Pevzner, P.A., 2002. Finding composite regulatory patterns in DNA sequences. *Bioinformatics* 18 (Suppl. 1), S354–S363.
- Ester, M., Zhang, X., 2004. A top-down method for mining most-specific frequent patterns in biological sequences. In: *Proceedings of SDM'04*.
- Fassetti, F., Greco, G., Terracina, G., 2008. Mining loosely structured motifs from biological data. *IEEE Transaction on Knowledge and Data Engineering* 20 (11), 1472–1489. (IEEE Computer Society, USA).
- Gross, C.A., Lonetto, M., Losick, R., 1992. Bacterial sigma factors. *Transcriptional Regulation* 1, 129–176.
- GuhaThakurta, D., Stormo, G.D., 2001. Identifying target sites for cooperatively binding factors. *Bioinformatics* 17 (7), 608–621.
- Hertz, G.Z., Stormo, G.D., 1999. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* 15 (7), 563–577.
- Hughes, J.D., Estep, A.F., Tavazoie, S., Church, G.M., 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *Journal of Molecular Biology* 296 (5), 1205–1214.
- Jonassen, I., Collins, J.F., Higgins, D.G., 1995. Finding flexible patterns in unaligned protein sequences. *Protein Science* 4, 1587–1595.
- Keich, U., Pevzner, P.A., 2002. Finding motifs in the twilight zone. In: *Proceedings of RECOMB'02*, pp. 195–204.
- Kim, J., Pramanik, S., Chung, M.J., 1994. Multiple sequence alignment using simulated annealing. *Computer Applications in the Biosciences* 10 (4), 419–426.
- Kolchanov, N.A., Podkolodnaya, O.A., *et al.*, 2000. Transcription regulatory regions database (TRRD): Its status in 2000. *Nucleic Acids Res* 28 (1), 298301.
- Liang, S., Samanta, M.P., Biegel, B.A., 2004. cWINNOWER algorithm for finding fuzzy DNA motifs. *Journal of Bioinformatics and Computational Biology* 2 (01), 47–60.
- Liu, J.S., 2008. *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.
- Liu, J.S., Neuwald, A.F., Lawrence, C.E., 1995. Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *Journal of the American Statistical Association* 90 (432), 1156–1170.
- Liu, X., Brutlag, D.L., Liu, J.S., 2001. BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pacific Symposium on Biocomputing* 6, 127–138.
- Marsan, L., Sagot, M.F., 2000. Algorithms for extracting structured motifs using a suffix tree with application to promoter and regulatory site consensus identification. *Journal of Computational Biology* 7, 345–360.
- Matys, V., Fricke, E., *et al.*, 2003. TRANSFAC: Transcriptional regulation, from patterns to profiles. *Nucleic Acids Research* 31 (1), 374378.
- Mewes, H.W., *et al.*, 2002. MIPS: A database for genomes and protein sequences. *Nucleic Acids Research* 30 (1), 31–34.
- Nettling, M., Treutler, H., Cerquides, J., Grosse, I., 2017. Combining phylogenetic footprinting with motif models incorporating intra-motif dependencies. *BMC Bioinformatics* 18 (1), 141.
- Neuwald, A.F., Green, P., 1994. Detecting patterns in protein sequences. *Journal of Molecular Biology* 239, 698–712.
- Neuwald, A., Liu, J., Lawrence, C., 1995. Gibbs motif sampling: Detection of bacterial outer membrane repeats. *Protein Science* 4, 1618–1632.
- Osanai, M., Takahashi, H., Kojima, K.K., Hamada, M., Fujiwara, H., 2004. Essential motifs in the 3' untranslated region required for retrotransposition and the precise start of reverse transcription in non-long-terminal-repeat retrotransposon SART1. *Molecular and Cellular Biology* 24 (19), 7902–7913.
- Pavesi, G., Mauri, G., Pesole, G., 2001. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* 17 (Suppl. 1), S207–S214.
- Pavesi, G., Mauri, G., Pesole, G., 2004. In silico representation and discovery of transcription factor binding sites. *Briefings in Bioinformatics* 5, 217–236.
- Pei, J., Han, J., Mortazavi-Asl, B., *et al.*, 2001. Prefixspan: Mining sequential patterns by prefix-projected growth. In: *Proceedings of ICDE'01*, pp. 215–224.
- Pevzner, P.A., Sze, S.H., 2000. Combinatorial approaches to finding subtle signals in DNA sequences. *ISMB* 8, 269–278.
- Robin, S., Daudin, J.J., Richard, H., Sagot, M.F., Schbath, S., 2003. Occurrence probability of structured motifs in random sequences. *Journal of Computational Biology* 9, 761–773.
- Sagot, M., 1998. Spelling approximate repeated or common motifs using a suffix tree. *LATIN'98: Theoretical Informatics*, 374–390.
- Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W.W., Lenhard, B., 2004. JASPAR: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research* 32, D91–D94.
- Sandve, G.K., Abul, O., Walseng, V., Drablos, F., 2007. Improved benchmarks for computational motif discovery. *BMC Bioinformatics* 8 (193), 1–13.
- Sandve, G.K., Drablos, F., 2006. A survey of motif discovery methods in an integrated framework. *Biology Direct* 1 (11), 1–16.
- Shida, K., 2006. GibbsST: A Gibbs sampling method for motif discovery with enhanced resistance to local optima. *BMC Bioinformatics* 7 (1), 486.
- Siddharthan, R., Siggia, E.D., Nimwegen, E., 2005. PhyloGibbs: A Gibbs sampling motif finder that incorporates phylogeny. *PLOS Computational Biology* 1, 534–556.
- Sinha, S., 2002. Composite motifs in promoter regions of genes: Models and algorithms. *General Report*.
- Sinha, S., Blanchette, M., Tompa, M., 2004. PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics* 5, 170.
- Sinha, S., Tompa, M., 2000. A statistical method for finding transcription factor binding sites. *ISMB* 8, 344–354.
- Sinha, S., Tompa, M., 2003a. Performance comparison of algorithms for finding transcription factor binding sites. In: *Proceedings of the Third IEEE Symposium on Bioinformatics and Bioengineering*, 2003, IEEE, pp. 214–220.
- Sinha, S., Tompa, M., 2003b. YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research* 31 (13), 3586–3588.
- Smith, H.O., Annau, T.M., Chandrasegaran, S., 1990. Finding sequence motifs in groups of functionally related proteins. *Proceedings of National Academy of Science of the United States of America* 17, 2421–2435.
- Thijs, G., Marchal, K., Lescot, M., *et al.*, 2002. A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *Journal of Computational Biology* 9 (2), 447–464.
- Thompson, J.D., Higgins, D.G., Gibson, T.J., 1994. CLUSTALW improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22, 4673.
- Tompa, M., 1999. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. *ISMB* 9, 262–271.
- Tompa, M., *et al.*, 2005. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology* 23 (1), 137–144.
- Torkamani, S., Lohweg, V., 2017. Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7 (2),
- Tu, Z., Li, S., Mao, C., 2004. The changing tails of a novel short interspersed element in *Aedes aegypti*: Genomic evidence for slippage retrotransposition and the relationship between 3' tandem repeats and the poly(da) tail. *Genetics* 168 (4), 2037–2047.
- Vanet, A., Marsan, L., Labigne, A., Sagot, M.F., 2000. Inferring regulatory elements from a whole genome. An analysis of *Helicobacter pylori* 80 family of promoter signals. *Journal of Molecular Biology* 297 (2), 335–353.
- Van Helden, J., Rios, A.F., Collado-Vides, J., 2000. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research* 28 (8), 1808–1818.
- Vilo, J., Brazma, A., Jonassen, I., Robinson, A.J., Ukkonen, E., 2000. Mining for putative regulatory elements in the yeast genome using gene expression data. *ISMB* 2000, 384–394.
- Vlieghe, D., Sandelin, A., Bleser, P., *et al.*, 2006. A new generation of JASPAR, the open-access repository for transcription factor binding site profiles. *Nucleic Acids Research* 34 (Database Issue), D95–D97.
- Wang, D., Tapan, S., 2012. MISCORE: A new scoring function for characterizing DNA regulatory motifs in promoter sequences. *BMC Systems Biology* 6 (2), S4.

- Wang, K., Xu, Y., Xu Yu, J., 2004. Scalable sequential pattern mining for biological sequences. In: Proceedings of CIKM'04, pp. 178–187.
- Wang, T., Stormo, G.D., 2003. Combining phylogenetic data with co-regulated genes to identify regulatory motifs. *Bioinformatics* 19, 2369–2380.
- Werner, T., 1999. Models for prediction and recognition of eukaryotic promoters. *Mammalian Genome* 10 (2), 168–175.
- Zaki, M.J., 2001. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning* 42 (1–2), 31–60.
- Zhang, S., Li, S., *et al.*, 2010. Simultaneous prediction of transcription factor binding sites in a group of prokaryotic genomes. *BMC Bioinformatics* 11, 397.
- Zhang, S., Xu, M., Li, S., Su, Z., 2009. Genome-wide de novo prediction of cis-regulatory binding sites in prokaryotes. *Nucleic Acids Research* 37 (10), e72.
- Zhao, F., Xuan, Z., Liu, L., Zhang, M.Q., 2005. TRED: A transcriptional regulatory element database and a platform for in silico gene regulation studies. *Nucleic Acids Research* 33 (Database Issue), D103–D107.
- Zhu, J., Zhang, M.Q., 1999. SCPD: A promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics* 15, 607–611.

Algorithms for Strings and Sequences: Pairwise Alignment

Stefano Beretta, University of Milan-Biocca, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The problem of aligning sequences is probably one of the classical and most important tasks in bioinformatics. In fact, in many studies one needs to compare strings or search for motifs such as, for example, nucleotide reads obtained from a sequencing process that must be aligned against a reference genome, or sequences of amino acids that must be compared to each other.

Recent advances in sequencing technologies have lead to a revolution in bioinformatics, since, with the introduction of Next-Generation Sequencing methods, a huge amount of data can be produced at a cheaper cost (Schuster, 2008). Thanks to this, it is now possible to perform new studies that were not even thinkable up to some years ago, such as the identification of organisms from a metagenomic survey (Rodrigue *et al.*, 2010; America, 2009), or the identification of structural variations present in genomes (Medvedev *et al.*, 2009; Tattini *et al.*, 2015). Although the computational problems arising from these studies require new algorithms and methods that are able to efficiently compute solutions by taking advantage of the different nature of the data, pairwise sequence alignment is still widely performed. In fact, after assembling transcripts or new genomes (e.g., from new species or from new individuals), it is important to determine the degree of similarity between two sequences, since high sequence similarity usually implies that they could have the same function or properties. For example, pairwise sequence alignment could also help in identifying the function of an unknown gene, by finding a similar gene (in terms of sequence composition) of known function.

In this work we will focus on the pairwise sequence alignment problem, in which two biological sequences are compared in order to identify their similarity. Intuitively, the process of aligning two sequences consists of lining them up, by possibly inserting spaces or gaps, while keeping the original order of the symbols, so that each symbol of one sequence is mapped to a symbol of the other sequence or to a null character or gap. More precisely, after introducing some basic notions related to this problem, we will present two types of pairwise alignments: global and local. The former alignment aims at finding the best alignment over the full length of the two input sequences, while the latter is used to identify the most similar parts (substrings) of the two sequences. We will also present two approaches, based on dynamic programming, to solve these two problems, and we will describe scoring functions used to achieve specific results in the alignment of biological sequences.

Background

In this section, we introduce the basic concepts of strings and pairwise alignment.

Let Σ be a non-empty finite set of symbols. A typical choice when dealing with nucleotide sequences is $\Sigma = \{A, C, G, T\}$. A string, $s = s_1 \dots s_n$, over an alphabet, Σ , is a sequence of symbols of Σ . We would like to point out that, although in some situations these terms are used as synonyms, in the rest of the paper we will use the term “sequence” only in the biological context, while for the mathematical and algorithmic ones we will use the term “string”.

Given a string, s , over Σ , we denote by $|s|$ the length of s , and by s_i , with $1 \leq i \leq |s|$, the symbol of s at position i . Moreover, we denote by $s_{i,j}$, with $1 \leq i < j \leq |s|$, the substring of s starting at position i and ending at position j , that is, the string $s_i \dots s_j$. We also denote by ϵ the empty string.

Consider now a symbol ‘-’, not in Σ , used to denote *gaps*, that is, ‘-’ $\notin \Sigma$. In order to define an alignment between two strings, s and t , over Σ , we consider the alphabet $\Sigma' = \Sigma \cup \{-\}$. At this point we are able to define the concept of alignment. Given two strings, s and t , over Σ , the (global) *alignment* of s and t is a pair (s', t') in which (i) s' is obtained from s by possibly inserting gaps, (ii) t' is obtained from t by possibly inserting gaps, (iii) $|s'| = |t'|$, and (iv) there is no position i , with $1 \leq i \leq |s'|$, such that both s'_i and t'_i are gaps.

As an example, consider the two strings, $s = \text{ATACAC}$ and $t = \text{AAGCGC}$, over the alphabet $\Sigma = \{A, C, G, T\}$; a possible alignment of s and t is:

$$s' = A \quad T \quad A \quad - \quad C \quad A \quad C$$
$$t' = A \quad - \quad A \quad G \quad C \quad G \quad C$$

In this example we obtained s' from s by adding a gap between the first A and C symbols (4th position), and we obtained t' from t by adding a gap between the two A symbols (2nd position). Considering this representation, in each position $i \in \{1, \dots, |s'| = |t'|\}$ of the alignment, we can find one of the following four possible situations:

- *insertion*: $s'_i = '-'$, corresponding to the fact that a gap has been inserted in the first string (position 4 of the example alignment);
- *deletion*: $t'_i = '-'$, corresponding to the fact that a gap has been inserted in the second string (position 2 of the example alignment);

- *match*: $s'_i = t'_j$, corresponding to the fact that the symbol in that position is the same in both the strings (positions 1, 3, 5, and 7 of the example alignment);
- *mismatch*: $s'_i \neq t'_j$, corresponding to the fact that the symbols in that position are different (position 6 of the example alignment).

It must be noticed that, from this definition, two input strings can always be aligned. Anyway, since the goal of pairwise alignment is to assess the similarity of the two input strings, it has to be scored. A common way to do this is to assign a value to each possible situation that can happen in a position when aligning two symbols, that is, to assign score values to insertion and deletion (gap alignments), match, and mismatch. Then, the score of the alignment is computed by summing up the values of the score at each position of the alignment. More precisely, we denote by w the *scoring function* that assigns to each possible combination of symbols in Σ' a score, that is $w : \Sigma' \times \Sigma' \rightarrow \mathbb{N}$ (or $\mathbb{Q}$, if we allow rational values).

From this point of view, it is possible to measure the *distance* or the *similarity* between the two input strings, depending on the specific values assigned by the scoring function. An example of the former type of measure is the *edit distance* in which the distance between the two strings is determined by the minimum number of edit operations required to transform the first string into the second one. On the other hand, different similarity measures can be adopted to quantify how much the two given input strings are similar. Recalling the previous example, it is possible to define a *similarity scoring function* by assigning a value 1 to match positions, and value -1 to mismatch and gap alignments, that is, to both insertions and deletions. The overall score of the alignment in the example is 1 (4 - 3). More precisely, in this case the (similarity) scoring function w is:

$$w(s'_i, t'_j) = \begin{cases} 1 & \text{if } s'_i = t'_j \wedge s'_i \neq '-' \\ -1 & \text{otherwise} \end{cases}$$

with $s'_i, t'_j \in \Sigma'$, and $1 \leq i, j \leq |s'| = |t'|$. In general, with this or any other similarity scoring function, higher values of the score will correspond to “better” alignments, i.e., shorter edit distances, and the optimal global pairwise alignment will have the highest score.

As anticipated, another possible choice for the scores of the positions is the *edit distance*, which measures the number of insertions, deletions, and substitutions required to transform one string into the other. It is interesting to notice that, when dealing with biological sequences, each edit operation corresponds to a biological operation: insertion, deletion, and mutation, respectively. A specific type of edit distance is the *Levenshtein distance*, which assigns 0 to matches, and 1 to mismatches and gap positions. In this case the scoring function modeling the edit distance w_{ed} is:

$$w_{ed}(s'_i, t'_j) = \begin{cases} 0 & \text{if } s'_i = t'_j \wedge s'_i \neq '-' \\ 1 & \text{otherwise} \end{cases}$$

with $s'_i, t'_j \in \Sigma'$, and $1 \leq i, j \leq |s'| = |t'|$. Since this is a distance measure, it corresponds to counting the number of non-matching positions, which means that the optimal alignment is the one that minimizes the score.

Obviously, there may be more than one solution, i.e., alignment, with the same optimal score.

In Section Methodologies a more detailed explanation of the *scoring function*, which assigns a score value to each possible combinations of symbols of the alphabet Σ' used in the alignment, will be provided.

Methodologies

As anticipated in Section Background, the goal of pairwise alignment of two input strings is to find the alignment having the best score. In this context, we can distinguish two type of alignments, namely *global* and *local*. In particular, the former type aligns each position of the two input strings by computing the best score that includes every character in both sequences, while the latter type aims at finding the regions of the input strings, that is, substrings, with the best match. To solve these problems, there exist different techniques, ranging from manual alignments made by inspection of the strings, to comparison using the dotplot matrix. Anyway, the most widely used techniques for the pairwise alignment, both global and local, rely on dynamic programming algorithms, in which the optimal alignment is obtained recursively by using the previously computed solutions of smaller substrings.

In the rest of this section, we will describe the *Needleman-Wunsch* algorithm (Needleman and Wunsch, 1970) which solves the global pairwise alignment problem, and the *Smith-Waterman* algorithm (Smith and Waterman, 1981), which solves the local pairwise alignment computation. Both algorithms are based on dynamic programming (Bellman, 1954). Moreover, we will also discuss different scoring functions which are commonly adopted to evaluate the alignments of biological sequences, i.e., nucleotide or protein.

Before presenting the two aforementioned algorithms, we will briefly explain the main concepts of dynamic programming, which is employed by both the methods. The idea is to recursively decompose the problem into subproblems and compute the solutions of the subproblems starting from partial solutions of smaller subproblems, computed at earlier stages. The dynamic programming technique can be applied when a solution to a problem can be obtained as a series of consecutive steps, each one working on a continuously growing subproblem. In this way, starting from the smallest subproblems, which are directly solved, subproblems are solved by combining previously computed solutions, following back the recursive decomposition. The solution of the final stage contains the overall solution.

The idea of applying this technique to the pairwise alignment problem of two input strings, s and t , is to consider the subproblems of aligning substrings, $s_{1,i}$ and $t_{1,j}$, with increasing values of i , ranging from 1 to $|s|$, and j , ranging from 1 to $|t|$. At each step, a new pair of positions, i and j , is considered, and the score $w(s_i, t_j)$ of their alignment is added to the score of the best alignment computed in the previous step. In this way, all the best alignments of two substrings ending at the last considered position can be stored in a matrix, as well as their scores. One of the key points about this dynamic programming method is that it is guaranteed to find at least one optimal alignment of the strings, s and t , for a given scoring function. On the other hand, although this technique can be used to simultaneously align several strings, it becomes quite slow, and is therefore not practical for large numbers of strings.

In the next two sections, we describe in detail the algorithmic procedures for the computation of the optimal global and local pairwise alignments. In this presentation, we adopt the scoring function w introduced in Section Background. More precisely, we define the following scoring function:

$$w(s'_i, t'_j) = \begin{cases} 1 & \text{if } s'_i = t'_j \wedge s'_i \neq '-' \quad (\text{Match}) \\ -1 & \text{if } s'_i \neq t'_j \wedge s'_i \neq '-' \wedge t'_j \neq '-' \quad (\text{Mismatch}) \\ -1 & \text{if } s'_i = '-' \vee t'_j = '-' \quad (\text{Gap}) \end{cases}$$

with $s'_i, t'_j \in \Sigma'$, and $1 \leq i, j \leq |s'| = |t'|$. With this similarity scoring function, the optimal alignment has the highest score and, consequently, both global and local algorithmic procedures maximize it.

Global Alignment

In this section we will describe the dynamic programming algorithm proposed by Needleman and Wunsch to solve the problem of computing an optimal global alignment between two strings (Needleman and Wunsch, 1970). Anyway, although we refer to the algorithm described in the rest of this section as that of Needleman and Wunsch, this is a bit imprecise. In fact, the first dynamic programming algorithm to solve the global pairwise alignment was introduced by Needleman and Wunsch in 1970 (having complexity $O(n^3)$ in time, although an efficient implementation can achieve an $O(n^2)$ in time), but the one described here is the one proposed later by Sellers (1974). This algorithm, which is also based on dynamic programming, requires $O(n^2)$ time, and uses a linear model for gap penalties (see later).

Let s and t be two strings over the alphabet, Σ , such that $|s| = n$ and $|t| = m$. The idea of the algorithm is to compute the optimal alignments of the prefixes of s and t , and to store the score in a matrix, M , having size $(n + 1) \times (m + 1)$, in which the '+' 1' is used to represent the initial empty substring ϵ . More precisely, $M[i, j]$ represents the score of an optimal alignment of two substrings $s_{1,i} = s_{\{1\} \dots s_{\{i\}}}$ and $t_{1,j} = t_{\{1\} \dots t_{\{j\}}}$. The dynamic programming algorithm computes $M[i, j]$ for i ranging from 0 (empty prefix) to n , and j ranging from 0 to m . At the end of the procedure, $M[n, m]$ is the score of the optimal global alignment of s and t .

Now, let us define the initialization of the dynamic programming, that is, the score of aligning the empty prefixes of the strings. More precisely, we start by setting $M[0, 0] = 0$, corresponding to the score of aligning two empty prefixes ϵ . Then, since we adopt a length-dependent model for gap penalties (Sellers, 1974), for every $i \in \{1, \dots, n\}$, set $M[i, 0] = i \cdot w(s_i, '-')$, corresponding to the score of aligning a prefix of s having length i , that is $s_{1,i}$, with the empty string ϵ . The same applies also for $j \in \{1, \dots, m\}$ for which we set $M[0, j] = j \cdot w('-', t_j)$, corresponding to the score of aligning a prefix of t having length j , that is $t_{1,j}$, with the empty string ϵ .

Now, it is possible to describe the recurrence. The idea, when considering the cell $M[i, j]$ of the matrix, which corresponds to the score of the optimal alignment of the prefixes $s_{1,i}$ and $t_{1,j}$, is that we take advantage of the previously computed scores for $i-1$ and $j-1$. Since, by definition, the goal is to maximize the final alignment score, we select the maximum score value among that of the alignments of the three possible prefixes of $s_{1,i}$ and $t_{1,j}$. The score of the alignment of the prefixes $s_{1,i}$ and $t_{1,j}$ is computed as the maximum of the following three cases:

- (1) s_i and t_j are aligned, and the score is that of the alignment of prefixes $s_{1,i-1}$ and $t_{1,j-1}$ plus the score $w(s_i, t_j)$ which is positive for a match and negative for a mismatch;
- (2) there is a deletion in t ; s_i is therefore aligned with a gap '-'. The score is that of the alignment of prefixes $s_{1,i-1}$ and $t_{1,j}$ plus the score of the gap alignment $w(s_i, '-')$;
- (3) there is an insertion in t ; t_j is aligned with a gap '-'. The score is that of the alignment of prefixes $s_{1,i}$ and $t_{1,j-1}$ plus the score of the gap alignment $w('-', t_j)$.

These cases are formalized in the following equation:

$$M[i, j] = \max \begin{cases} M[i-1, j-1] + w(s_i, t_j) & (1) \\ M[i-1, j] + w(s_i, '-') & (2) \\ M[i, j-1] + w('-', t_j) & (3) \end{cases}$$

for $i \in \{2, \dots, n\}$ and $j \in \{2, \dots, m\}$. Fig. 1 offers a graphical representation of the general case of the dynamic programming recurrence to compute the score $M[i, j]$ starting from the previously computed scores.

Finally, the score $Score_{OPT}$ of the optimal alignment of the two input strings s and t is in the cell $M[n, m]$:

$$Score_{OPT} = M[n, m]$$

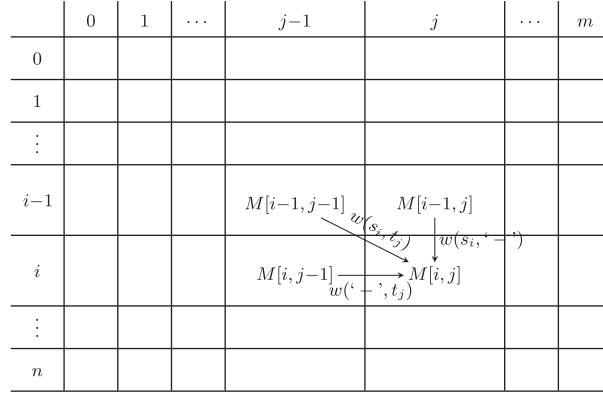


Fig. 1 Graphical representation of the computation of $M[i, j]$ in the matrix M using a linear gap penalties model (Sellers, 1974). Here, the 3 previously computed values on which the recurrence is based are highlighted, namely $M[i-1, j-1]$, $M[i-1, j]$, and $M[i, j-1]$. Labels on the arrows report the additional score that is added, corresponding to the pair of aligned symbols.

$s \backslash t$	ϵ	A	A	G	C	G	C
ϵ	0	-1	-2	-3	-4	-5	-6
A	-1	1	0	-1	-2	-3	-4
T	-2	0	0	-1	-2	-3	-4
A	-3	-1	1	0	-1	-2	-3
C	-4	-2	0	0	1	0	-1
A	-5	-3	-1	-1	0	0	-1
C	-6	-4	-2	-2	0	-1	1

$s \backslash t$	ϵ	A	A	G	C	G	C
ϵ	·	←	←	←	←	←	←
A	↑	↖	↖	←	←	←	←
T	↑	↑	↖	↖	↖	↖	↖
A	↑	↖	↖	←	←	←	←
C	↑	↑	↑	↖	↖	←	↖
A	↑	↖	↖	↖	↑	↖	↖
C	↑	↑	↑	↖	↖	↖	↖

Fig. 2 Example of the computation of optimal alignment matrices for the two input strings $s=ATACAC$ and $t=AAGCGC$. The left matrix represents the score of the optimal (maximum) alignment, while the right matrix contains the arrows to keep track of the best choice done to obtain the optimal score. The grey path corresponds to the optimal final alignment. In both the matrices, instead of reporting the values of $i \in \{0, \dots, 6\}$ and $j \in \{0, \dots, 6\}$ as row and column names, respectively, we used the corresponding symbols of the strings, $\{\epsilon, s_1, \dots, s_6\}$ and $\{\epsilon, t_1, \dots, t_6\}$, respectively.

Anyway, in order to reconstruct the optimal alignment (or alignments if there is more than one), it is necessary to keep track, for each pair of symbols at positions i and j , from which of the three aforementioned cases was obtained the value $M[i, j]$ during the dynamic programming. More precisely, since to compute $M[i, j]$ we select the maximum value among the 3 possible cases of the dynamic programming recurrence, in order to reconstruct the optimal alignment we have to compute from which value we obtained the maximum score $M[i, j]$ among $M[i-1, j-1]$, $M[i-1, j]$, and $M[i, j-1]$ (see Fig. 1). A possible way to represent this choice is to use arrows pointing to the corresponding matrix cell so that, once the matrix is completed, a path can be traced from the cell with the optimal value $Score_{OPT}$ (i.e., $M[n, m]$) to $M[0, 0]$, and the alignment reconstructed from the pairs of letters aligned at each step. This reconstruction process is usually called *traceback*. In fact, there is a pair of symbols corresponding to each possible arrow at a given position i, j of the matrix:

- ↖ corresponds to the alignment of symbols s_i and t_j ;
- ↑ corresponds to the alignment of symbols s_i and '-';
- ← corresponds to the alignment of symbols '-' and t_j .

The two matrices in Fig. 2 show the scores and the corresponding arrows for the computation of the optimal alignment of the two input strings used as an example in Section Background: $s=ATACAC$ and $t=AAGCGC$. By slightly abusing the notation, the row/column indices can be used as the symbols of the input string s/t , instead of the positions $i \in \{0, \dots, 6\}$ and $j \in \{0, \dots, 6\}$. Following the path in the right matrix of Fig. 2 we can reconstruct an optimal alignment having score equal to 1:

$$s' = A \quad T \quad A \quad - \quad C \quad A \quad C$$

$$t' = A \quad - \quad A \quad G \quad C \quad G \quad C$$

We would like to point out that, for ease of exposition, in the example in Fig. 2 only a single arrow for each cell of the matrix is shown, whereas, in reality, an equal optimal score can frequently be obtained from multiple "directions". This is due to the fact

that there could be multiple alignments of the two input strings having the same optimal score, $Score_{OPT}$. For this reason, if it is necessary to reconstruct all the optimal alignments of two given strings, for each cell, all the possible optimal choices should be stored. Moreover, the traceback step used to reconstruct the optimal alignment should explore all the possible paths (not just a single one) from $M[n, m]$ to $M[0, 0]$.

Local Alignment

In this section, we will describe the algorithm proposed by Smith and Waterman, which is used to solve the problem of computing the optimal local alignment of two input strings (Smith and Waterman, 1981). The goal of this procedure is to find the substrings of the input strings that are the most similar, with respect to a given scoring function, which usually represents a similarity measure (like the one used in Section Global Alignment to describe the global alignment).

The Smith–Waterman algorithm is based on a dynamic programming recurrence similar to that used to find the global pairwise alignment, and follows the same idea: compute a matrix with the optimal scores of the alignments of the prefixes of the input strings, keeping track of the previous optimal partial alignments, and at the end follow a traceback path to reconstruct an optimal alignment.

Formally, given two input strings, s and t , having lengths n and m , respectively, we compute a $(n + 1) \times (m + 1)$ matrix M , in which $M[i, j]$, with $i \in \{0, \dots, n\}$ and $j \in \{0, \dots, m\}$, represents the optimal (maximum) score of a local alignment between the substrings $s_{1,i}$ and $t_{1,j}$. Here, the interpretation of the matrix values is slightly different with respect to that of the global pairwise alignment: $M[i, j]$ represents the score of the best alignment of a suffix of $s_{1,i}$ and a suffix of $t_{1,j}$. In the local pairwise alignment matrix, the first row and the first column, i.e., those with index 0, are set to 0, since there is no penalty in introducing gaps. Moreover, as anticipated, for each cell of the matrix we store the path taken to obtain the maximum score value (i.e., through an arrow to the corresponding cell) in order to reconstruct the best local alignment.

The recurrence starts by setting the first row and the first column of the matrix M with 0, that is, $M[0, j] = 0$, with $j \in \{0, \dots, m\}$, and $M[i, 0] = 0$, with $i \in \{0, \dots, n\}$. Now, the recursive case of the recurrence, is:

$$M[i, j] = \max \begin{cases} M[i-1, j-1] + w(s_i, t_j) \\ M[i-1, j] + w(s_i, '-') \\ M[i, j-1] + w('-', t_j) \\ 0 \end{cases}$$

for $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$.

Notice that, while an alignment can be produced for any scoring system, if the average score for comparing the letters is not below zero or, more in general, for all positive scoring systems, it will result in a global alignment (and not in a local one).

and for an all positive scoring system

The value 0 at position (i, j) , i.e., $M[i, j] = 0$, indicates that a new alignment starts at this position. In fact, if the score of the best alignment up to (i, j) is negative, then it is better to start a new one. Differently from the global pairwise alignment procedure described before, all the negative scores are replaced by 0 s in the matrix M .

Finally, the score of the optimal local pairwise alignment between s and t can be found by searching for the highest value in the matrix M :

$$Score_{OPT} = \max\{M[i, j] : 1 \leq i \leq n, 1 \leq j \leq m\}$$

After that, similarly to the global alignment procedure, in order to reconstruct a local alignment having optimal score, it is necessary to follow the traceback path, from the matrix cell of M having score $Score_{OPT}$ up to a cell having score equal to 0.

The matrices in Fig. 3 show an example of optimal local alignment for the two input strings $s = ACTAAG$ and $t = CTCAAT$ w.r.t. the scoring function introduced at the beginning of Section Methodologies. More precisely, the left matrix reports the score values obtained with the dynamic programming recurrence, while the matrix on the right shows the arrows corresponding to the choices made, that is, the cells from which the values have been obtained. In this latter matrix, we use a dot (i.e., $\cdot$) for those cells with a 0 score obtained through the last case of the dynamic programming recurrence.

On the other hand, as in the example of the global pairwise alignment shown in Section Global Alignment, only one arrow for each cell is shown, following the order shown in the definition of the recurrence, when there is more than one possible choice. Finally, the grey path corresponds to the optimal local alignment having $Score_{OPT} = 3$:

$$\begin{aligned} s' &= C & T & - & A & A \\ t' &= C & T & C & A & A \end{aligned}$$

Extension to Affine Gap Model

Now, let us analyze the introduction of gap symbols in the alignment. Both the algorithms described in Section Global Alignment and Section Local Alignment use a linear gap model to assign a score to the alignment of a symbol in the alphabet Σ with the gap '-'. This usually consists of a negative score associated with a gap being dependent on the number of successive gap characters.

$s \backslash t$	ϵ	C	T	C	A	A	T
ϵ	0	0	0	0	0	0	0
A	0	0	0	0	1	1	0
C	0	1	0	1	0	0	0
T	0	0	2	1	0	0	1
A	0	0	1	1	2	1	0
A	0	0	0	0	2	3	2
G	0	0	0	0	1	2	2

$s \backslash t$	ϵ	C	T	C	A	A	T
ϵ	.	.	.	.	.	.	.
A	.	.	.	.	↖	↖	←
C	.	↖	←	↖	↑	↑	↖
T	.	↑	↖	←	↖	.	↖
A	.	.	↑	↖	↖	↖	↑
A	.	.	↑	↑	↖	↖	←
G	.	.	.	.	↑	↑	↖

Fig. 3 Example of the computation of optimal local alignment matrices for the two input strings $s = ACTAAG$ and $t = CTCAAT$. The left matrix represents the score of the optimal (maximum) alignment, while the right matrix contains the arrows to keep track of the best choice done to obtain the optimal score. The grey path corresponds to the optimal local final alignment. In both the matrices, instead of reporting the values of $i \in \{0, \dots, 6\}$ and $j \in \{0, \dots, 6\}$ as row and column names, respectively, we used the corresponding symbols of the strings, $\{\epsilon, s_1, \dots, s_6\}$ and $\{\epsilon, t_1, \dots, t_6\}$, respectively.

More precisely, in this *length-dependent* gap model, the score of introducing a sequence of k consecutive gap symbols in an aligned string s is $k \cdot w(t_j, '-')$, with $1 \leq j \leq m$, that is k times the score of aligning a symbol t_j of t with a gap (Sellers, 1974).

Anyway, in many biological analyses (but also in other research fields) having k consecutive gaps is preferable to having k gaps scattered along the strings in different positions. For this reason, a different gap model, which distinguishes between the score for opening the gap (length independent score) and the cost proportional to the gap length, was introduced by Gotoh (1982). This model, also known as the *affine* gap cost model, penalizes alignments with many scattered gap positions more than alignments with more consecutive ones.

Now, given a gap opening cost h , and a gap proportional cost similar to that used in Section Global Alignment and Section Local Alignment, the dynamic programming recursion to compute the global pairwise alignment can be modified as follows:

$$M[i, j] = \max \begin{cases} M[i-1, j-1] + w(s_i, t_j) \\ I_s[i-1, j-1] + w(s_i, t_j) \\ I_t[i-1, j-1] + w(s_i, t_j) \end{cases}$$

$$I_s[i, j] = \max \begin{cases} M[i-1, j] + h + w(s_i, '-') \\ I_s[i-1, j-1] + w(s_i, '-')$$

$$I_t[i, j] = \max \begin{cases} M[i, j-1] + h + w('-', t_j) \\ I_t[i, j-1] + w('-', t_j) \end{cases}$$

for $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. The matrices are initialized by setting:

- $M[0, 0] = 0$;
- $I_s[i, 0] = h + i \cdot w(s_i, '-')$, with $i \in \{1, \dots, n\}$;
- $I_t[0, j] = h + j \cdot w('-', t_j)$, with $j \in \{1, \dots, m\}$;
- all the other cells in the first rows and first columns of matrices M , I_s , and I_t to $-\infty$.

The additional matrices I_s and I_t are used to deal with the introduction of gaps in the alignment, to distinguish between the case in which a new gap is opened and the case in which it is extended.

Notice that the Smith-Waterman dynamic programming recurrence can be modified in the same way to adopt the affine gap model. Anyway, in both cases, the time complexity of the two algorithms still remains $O(n^2)$.

Scoring Functions

In this section we briefly discuss different scoring functions that are used in real case studies to perform pairwise alignments, especially when dealing with nucleotide and protein sequences. More precisely, another way to represent a scoring function $w : \Sigma' \times \Sigma' \rightarrow \mathbb{N}$ which assigns a value to each possible alignment of two symbols, is through a *scoring matrix*, that is a matrix having size $|\Sigma'| \times |\Sigma'|$. In this way, to each possible pair of symbols of $\Sigma' (= \Sigma \cup \{-'\})$ is assigned a numerical value corresponding to the score for aligning the corresponding symbols. This representation provides an easy way to assign different scores to the alignments of different symbols. This is useful especially when dealing with protein alignments (but also with nucleotide), in which the

different pairs of amino acids have different degrees of chemical similarity. For this reason it is important to choose a scoring function that reflects biological observations when aligning protein sequences.

For this purpose, a set of scoring matrices has been designed to model the fact that some pairs of symbols are more probable than others in related sequences, by encoding the log-odd scores of amino acid mutations. In fact, although different types of scoring matrices have been proposed, such as those based on the similarity in terms of chemical properties, the ones obtained from empirical observations have shown to perform better than the others. The two most widely used matrices adopted for the pairwise alignment of protein sequences are Percent Accepted Mutation (PAM) (Dayhoff and Schwartz, 1978) and BLOSUM (Henikoff and Henikoff, 1992) matrices.

The former, i.e., the PAM matrices, have been computed by observing point mutation rates of amino acids in closely related proteins, with known phylogenies. The scores in the PAM1 matrix have been obtained by aligning closely related protein sequences, and estimate the expected rate of substitution if 1% of amino acids undergo substitutions. Starting from the PAM1 matrix, the PAM2 matrix is created by assuming an additional 1% of changes of the amino acids and, mathematically, this is done by multiplying the PAM1 matrix by itself. In general, this process can be repeated n time in order to obtain the PAM n matrix. The most widely used PAM matrices are PAM30, PAM70, PAM120, and PAM250. It must be pointed out that the scores of a PAM matrix are the logarithms (usually base 2) of the rate, averaged over the forward and backward substitution, and normalized to the rate of seeing a given amino acid (*log-odds scores*).

The other set of scoring matrices commonly used in the pairwise alignment of biological sequences are the BLOSUM matrices (Henikoff and Henikoff, 1992), which are based on the blocks database. These matrices consist of the logarithms (usually base 2) of the relative substitution frequencies of the aligned blocks from different protein families, representing a variety of organisms (*log-odds scores*). As for the PAM matrices, different BLOSUM matrices have been computed, each one having a different threshold of minimum similarity among the sequences of the organisms. More precisely, the BLOSUM N matrix is built by considering sequences having more than the $N\%$ of similarity as corresponding to a single sequence, thereby downweighting the information from highly similar sequences. One of the most used matrix is the BLOSUM62 matrix, which is built by clustering together and downweighting sequences with more than the 62% of identity (Styczynski *et al.*, 2008).

Conclusions

In bioinformatics, one of the most classical problems is the alignment of two biological sequences, which is usually performed in order to assess similarities of DNA, RNA, or protein strings. Due to its importance, this problem has been thoroughly studied and many different variants proposed in literature. The two most important variants are global and local pairwise alignment. To solve these type of problems, the two most famous approaches are those proposed by Needleman and Wunsch, and by Smith and Waterman to compute optimal global and local pairwise alignment, respectively. Both techniques are based on dynamic programming algorithms, which find the optimal score of the pairwise alignment, with respect to a given scoring function, and also reconstruct an optimal alignment of the two input strings. In discussing the two recurrences on which the two algorithms are based, we provided also some an example for showing their functioning.

In this work, in addition to the linear gap model adopted in both the dynamic programming algorithms for solving the global and local pairwise alignments, we also discussed the affine gap model. Finally, we discussed some variants of the scoring function that are usually adopted in real case studies, in order to deal with biological sequences.

Further Reading

For a more detailed discussion on the pairwise alignment problem, we refer the reader to (Jones and Pevzner, 2004), to (Gusfield, 1997), and to (Setubal and Meidanis, 1997). These books provide a description of different methods for solving global and local pairwise alignment problems, in addition to the one presented in this work, and also give an extensive explanation of different scoring functions adopted for the alignment of biological sequences. In these books, heuristic methods for solving the aforementioned problems are discussed in detail. Moreover, a detailed analysis of gap penalties and different scoring functions is present in the aforementioned books.

See also: Algorithms for Strings and Sequences: Searching Motifs. Algorithms Foundations. Sequence Analysis

References

- America, N., 2009. Inc: Metagenomics versus moores law. *Nature Methods* 6, 623.
- Bellman, R., 1954. The theory of dynamic programming. Technical Report. RAND CORP SANTA MONICA CA.
- Dayhoff, M.O., Schwartz, R.M., 1978. Chapter 22: A model of evolutionary change in proteins. In: Dayhoff, M.O. (Ed.), *Atlas of Protein Sequence and Structure*. Washington, DC: National Biomedical Research Foundation, pp. 345–352.

- Gotoh, O., 1982. An improved algorithm for matching biological sequences. *Journal of Molecular Biology* 162, 705–708.
- Gusfield, D., 1997. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press.
- Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences* 89, 10915–10919.
- Jones, N.C., Pevzner, P., 2004. *An Introduction to Bioinformatics Algorithms*. MIT Press.
- Medvedev, P., Stanciu, M., Brudno, M., 2009. Computational methods for discovering structural variation with next-generation sequencing. *Nature Methods* 6, S13–S20.
- Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48, 443–453.
- Rodrigue, S., Materna, A.C., Timberlake, S.C., *et al.*, 2010. Unlocking short read sequencing for metagenomics. *PLOS ONE* 5, e11840.
- Schuster, S.C., 2008. Next-generation sequencing transforms today's biology. *Nature Methods* 5, 16.
- Sellers, P.H., 1974. On the theory and computation of evolutionary distances. *SIAM Journal on Applied Mathematics* 26, 787–793.
- Setubal, J.C., Meidanis, J., 1997. *Introduction to Computational Molecular Biology*. PWS Publishing.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* 147, 195–197.
- Styczynski, M.P., Jensen, K.L., Rigoutsos, I., Stephanopoulos, G., 2008. Bio-sum62 miscalculations improve search performance. *Nature biotechnology* 26, 274–275.
- Tattini, L., DAurizio, R., Magi, A., 2015. Detection of genomic structural variants from next-generation sequencing data. *Frontiers in Bioengineering and Biotechnology* 3, 92.

Algorithms for Strings and Sequences: Multiple Alignment

Pietro H Guzzi, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In computational biology and bioinformatics, the term sequence alignment refers to the identification of regions of similarity among the sequence of gene and gene products. Similarity may be a consequence of evolution, therefore alignment is a way to infer the impact of the evolution and the evolutionary relationship among sequences.

Sequence alignment algorithms fall into two main classes considering the number of aligned sequences: pairwise and multiple. A multiple sequence alignment (MSA) is a sequence alignment that receives as input three or more sequences and produces as output the analysis of similarity among the sequences. Sequence alignment algorithms are usually based on the concept of edit distances, i.e. how many changes are needed to transform one sequence into another one. The output of MSA is generally matrices for a visual depiction of the alignment to enhance the similarity and the differences. Differences or point mutations (single amino acid or nucleotide changes) that appear as different characters in a single alignment column, and insertion or deletion mutations (indels or gaps) that appear as hyphens in one or more of the sequences in the alignment.

From a computational point of view the multiple sequence alignment can be computationally expensive, and in general time-consuming, therefore multiple sequence alignment programs use heuristic methods rather than global optimization (Thompson *et al.*, 2002).

Multiple Sequence Alignment Algorithms

Initially, MSAs used dynamic programming to identify the best optimal alignment, extending the pairwise approaches. The optimization strategy was based on two main parameters: a penalty for differences (gap penalty) and a matrix used for assignment scores for the changes.

Given n sequences, the naive algorithm is based on the construction of an n -dimensional matrix extending the two-dimensional matrix used in pairwise algorithms. Consequently, the search space grows exponentially. It has been demonstrated that dynamic programming, in this case, give an NP-complete problem. The Carrillo-Lipman algorithm (Carrillo and Lipman, 1988) represents an evolution of this simple scheme yielding a reduction of the complexity preserving the global optimum (Lipman *et al.*, 1989).

The computational complexity of the problem caused the introduction of approximate algorithms based on a heuristic.

A big class of heuristic algorithms is usually defined to as progressive alignment (also known as hierarchical or tree-based methods). They are based on the progressive technique developed by Paulien Hogeweg and Ben Hesper (Hogeweg and Hesper, 1984). Each progressive alignment is based on two steps: (i) initially all the pairwise alignment are calculated, and the relationships are represented as a tree (guide tree), (ii) the building of MSA by adding the sequences sequentially to the growing MSA according to the guide tree. The initial guide tree is calculated using clustering methods, such as neighbour-joining or UPGMA. Examples of progressive alignments are the Clustal series and T-Coffee.

Consensus methods try to find the optimal MSA given multiple different alignments of the same set of sequences. Examples of consensus methods are M-COFFEE and MergeAlign. M-COFFEE uses multiple sequence alignments generated by seven different methods that are integrated to build consensus alignments. MergeAlign extends the previous approach by accepting as input a variable number of input alignments generated using different models of sequence evolution or different methods of multiple sequence alignment.

Iterative methods build a MSA by iteratively realigning the initial sequences as well as adding new sequences to the growing MSA. Examples of iterative methods are the software package PRRN/PRRP and DIALIGN.

Tools for Multiple Sequence Alignment

The Clustal is a series of computer programs used for multiple sequence alignment. Clustal series is composed by:

- Clustal: The original software for progressive alignment.
- ClustalW and ClustalX that are respectively a command line interface and the graphical user interface.
- Clustal ω (Omega): The current standard version that is available via a web interface.

To perform an alignment using ClustalW, you may select the sequences or alignment you wish to align. Then you have to choose the available options: Cost Matrix (the desired cost matrix for the alignment); Gap open cost and Gap extend cost; Free end gaps; Preserve original sequence order. After entering the desired options click OK and ClustalW will be called to align the selected sequences or alignment. Once complete, a new alignment document will be generated with the result as detailed previously.

See also: Algorithms for Strings and Sequences: Pairwise Alignment. Algorithms for Strings and Sequences: Searching Motifs. Algorithms Foundations. Sequence Analysis

References

- Carrillo, H., Lipman, D., 1988. The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics* 48 (5), 1073–1082.
- Hogeweg, P., Hesper, B., 1984. The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *Journal of molecular evolution* 20 (2), 175–186.
- Lipman, D.J., Altschul, S.F., Kececioglu, J.D., 1989. A tool for multiple sequence alignment. *Proceedings of the National Academy of Sciences* 86 (12), 4412–4415.
- Thompson, J.D., Gibson, T., Higgins, D.G., *et al.*, 2002. Multiple sequence alignment using clustalw and clustalx. *Current protocols in bioinformatics* 2–3.

Biographical Sketch

Pietro H. Guzzi is an assistant professor of Computer Science Engineering at the University Magna Graecia of Catanzaro, Italy. His research interests comprise semantic-based and network-based analysis of biological and clinical data.

Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins

Giuseppe Tradigo, University of Calabria, Rende, Italy

Francesca Rondinelli, Università degli Studi di Napoli Federico II, Napoli, Italy

Gianluca Pollastri, University College Dublin, Dublin, Ireland

© 2019 Elsevier Inc. All rights reserved.

Introduction

After the Human Genome Project, which had the goal of sequencing the whole genomes of various organisms, the focus of research moved towards proteomics. Proteins are long biomolecules which are responsible of the vast majority of the complex signaling and chemical interactions in the cell. The role of a protein is mainly due to its geometrical shape which determines which of its amino acids, also known as residues, are exposed towards other molecules and also the chemical potential of its binding sites, hence all of its properties. The role a protein has in the cell is often referred to as its function. Protein function prediction is also a very active research field in which the protein shape is a key information for achieving good prediction results. Many scientific expertises are involved in protein structure prediction research, among which: mathematicians, Chemists, Medicinal Chemists, Physicians, Biologists, Physicists and Computer Scientists.

The scientific community working on protein structure prediction formulated this problem as the Protein Folding Problem (PFP), which tries to give an answer to the following question: which are the rules behind protein folding? More precisely, PFP has the main goal to understand which is the function to map a sequence of amino acid (primary sequence) into the spatial description of all atoms in the molecule (tertiary structure). Proteins are described by a number of such structures, as we will see in the following.

Proteins are long molecules composed of a chain of amino acids. Each amino acid derives from a triplet on a coding portion of the genome. The ribosome (another protein) reads such triplets of bases from a messenger RNA (mRNA) and creates the chain of amino acids of the protein, according to the central dogma of biology (see [Crick, 1970](#)). While the polypeptide chain of a protein is being forming, the interactions among amino acids bend it in order to reach the structure with the lowest conformational energy. Such a process is called protein folding. The central dogma states that all information flows from the nucleus (i.e., DNA) towards the proteome and not vice versa. This implies that the sequence of amino acids translated from the genome implies its shape because of the function that has to be implemented in the cell. However, as we will see in the next section, exceptions to this rule have been observed in nature. In particular, under varying environmental conditions, the same protein folds into different shapes.

Protein structures can be determined experimentally, for instance by using X-ray crystallography or NMR (Nuclear Magnetic Resonance), but such experiments are both time consuming and extremely costly. Computer-based approaches are able to define statistical and mathematical models which can predict protein structures in high throughput pipelines, thus enabling investigations not feasible with classical approaches, such as: (i) Genome-wide studies, in which all proteins of a genome are transcribed *in silico* (ii) screenings of large datasets of molecules, used in chemoinformatics investigations, (iii) ligand docking screenings, with which researchers try to find the best target for a particular ligand, (iv) unknown target identification and classification, in which a novel protein is assigned to a class or labeled with its domains. Furthermore, structure predictions can be used to create vast curated and annotated repositories of molecules and their properties, which can be integrated with other existing data sources to extract new knowledge, explain and interpret data. Such an integration with ontologies and reasoning techniques could hopefully help in finding appropriate answers to unsolved biological and clinical problems.

Background

PFP has been proposed almost 50 years ago and it can be stated as three main questions: (i) What is the mechanism responsible for the folding of the linear sequence of amino acids composing a protein, (ii) how can proteins fold so fast and (iii) is there a way to create computer programs able to guess the 3D shape of a protein sequence will have in its folded spatial conformation ([Dill and MacCallum, 2012](#)). In nature folding is a spontaneous process driven by chemical bonds created among amino acids along the polypeptide chain of the protein. The folding process starts while the protein is being translated from the genetic code contained in messenger RNA filaments read by the ribosome.

One of the main online database containing resolved protein molecules is the Protein Data Bank (PDB), ([Berman et al., 2000](#)). The repository has been created in 1971 and originally contained as few as seven structures. This number has significantly increased over the years towards many thousands structures, mainly resolved with both X-ray crystallography and NMR. Each structure is stored in a dedicated file together with annotations about the authors, the publications describing the experiment and the list of atoms, coordinates and chemical properties of the entry. There currently exist 126,000,000 + identified protein primary sequences in the Uniprot database ([UniProt Consortium, 2016](#)) but just ~ 125,000 of them (less than one every a thousand) have an experimentally resolved 3D structure stored in the PDB Database, and the gap is rapidly widening over time. However,

according to the PDBSelect25 database, by only considering sequences being significantly different from each, i.e., having a mutual sequence identity lower than 25%, we obtain as low as ~ 4600 structures (Griep and Hobohm, 2009). This suggests that the effort of designing and implementing software systems able to perform genome-wide protein sequence structure prediction is still relevant and useful for the research community. Among the other online protein data sources, the Structural Classification of Proteins (SCOP) database contains information about evolutionary relationships among proteins and the principles that are behind their 3D structure (Reddy and Bourne, 2003). It also provides correlations with atoms coordinates resources, rendered images of the structure, sequence data and literature references.

Predicting atoms locations with high precision can be achieved by using computer based methods which: (i) Approximate the Schrödinger equation to capture chemical potentials for molecular orbitals or are based on the laws of physics to simulate forces among atoms (ii) learn from online databases containing known protein structures (e.g., the PDB), what are the statistical properties of similar sequences and create models for local (short peptides) and global (whole protein sequences) prediction. The former refers to methods based on Density Functional Theory (DFT) and on Molecular Dynamics (MD). The latter attains a large number algorithms and software tools, the vast majority of which have been created around the CASP (Critical Assessment of protein Structure Prediction) experiment research community (Moult *et al.*, 2014). We will give more details about these approaches and techniques in Section “Systems and/or Applications”.

The central dogma of molecular biology states that information flows from the coding portion of the genome towards the proteome and this flow is irreversible. Hence the protein structure is implied by the genetic information stored in the DNA/RNA and, since recently, such a structure was believed unique. However, experimental evidence of violations of such a rule has been found in the so-called metamorphic proteins (Murzin, 2008; Koonin, 2012). For these molecules, it has been noted that cell environmental factors (e.g., pH) can induce different protein structural rearrangements, even if it is still not clear if their shape can change again after the protein has folded in its first shape.

The process of genomic information translation into proteins starts once the mature messenger RNA exits from the nucleus and it is intercepted by a ribosome, a specialized protein molecule which associates an amino acid to each triplet of genes read from the mRNA sequence. Each amino acid attaches to the previous one forming a long chain which folds into a well-defined 3D structure. For the last 50 years Protein Folding Problem (PFP) has been one of the main driving forces for the investigation on tertiary structure prediction. The goal of PFP is to find the rules behind the folding of the amino acid chain of proteins. Such a function is thought to be directly implied by the primary structure of the protein, i.e., by the sequence of its amino acids.

A lot of recurrent patterns can be identified in a protein, which have been organized in well-known structures: (i) The primary structure, which is the sequence of its amino acids, (ii) the secondary structure, which stores the locations of highly organized substructures along the chain, such as alpha-helices and beta-sheets, (iii) the tertiary structure, which defines the coordinates of each atom constituting the protein and (iv) the quaternary structure, which takes into account how two or more proteins of the same kind pack together to form a complex. At the chemical level, protein molecules are long polypeptide chains in which the alpha-carbon atoms of adjacent amino acids are bond through by a covalent bond, called peptide bond.

Protein function is how a protein interacts with other molecules in the cell environment and is a direct consequence of the three-dimensional structure. There exist proteins which contribute in various roles in the cell, some of which are: (i) Catalysis of chemical reactions, where they contribute to accelerate some reactions, (ii) transport of nutrients or drugs, where their chemical shape or binding site is useful to host other molecules, (iii) mechanical functions, where their shape offers physical resistance to external forces, as for instance in hair or where they are used to achieve mechanical tasks such as the ribosome, the motor proteins or the flagellum of *Escherichia coli*, (iv) transmission and exchange of signals, where they help executing complex tasks in the cell through biochemical signaling described in cellular pathways.

Systems and/or Applications

Exact methods are widely used to precisely determine the 3D structure of protein molecules, the most used of which are NMR and X-ray crystallography. Both technologies are both costly and time consuming. The former limits the number of molecules which can be resolved with a certain budget. The latter is a compelling constraint when the number of molecules to be resolved is large.

NMR spectroscopy and X-ray crystallography are two of the most important techniques for gaining insight about the conformation of proteins. NMR spectroscopy elucidates the atomic structure of macromolecules in solution, in case of highly concentrated solutions (~ 1 mM, or 15 mg ml^{-1} for a 15-kd protein). This technique depends on certain atomic nuclei intrinsic magnetic properties. Only a limited number of isotopes display the well known property, called spin.

On the other hand, X-ray crystallography provides the finest visualization of protein structure currently available. This technique identifies the precise three-dimensional positions of most atoms in a protein molecule. The use of X-rays provides the best resolution because the wavelength of X-rays is in the same range of a covalent bond length. The three components in an X-ray crystallographic analysis are a protein crystal to be obtained, a source of X-rays, and a detector (Berg *et al.*, 2002).

In the last 30 years, Density Functional Theory (DFT) allowed great advances at the interface between Physical Chemistry and Life Science topics (Mardirossian and Head-Gordon, 2017; Medvedev *et al.*, 2017). The origin of DFT success both in academia and in industrial applications is its exact approach to the problem of electronic structure theory. According to the Born-Oppenheimer approximation, the electronic energy, $E_e[\rho(\mathbf{r})]$, can be expressed as a functional of the electron density.

$$E_e[\rho(\mathbf{r})] = T[\rho(\mathbf{r})] + V_{en}[\rho(\mathbf{r})] + J[\rho(\mathbf{r})] + Q[\rho(\mathbf{r})] \quad (1)$$

In Eq. (1), $T[\rho(\mathbf{r})]$ represents the electronic kinetic energy, $V_{en}[\rho(\mathbf{r})]$ the nuclear-electron attraction energy, $J[\rho(\mathbf{r})]$ the classical electron-electron repulsion energy, and $Q[\rho(\mathbf{r})]$ the quantum electron-electron interaction energy. The second and third terms in Eq. (1) can be evaluated by Eqs. (2) and (3), respectively.

$$V_{en}[\rho(\mathbf{r})] = - \sum_{A=1}^M \int \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \rho(\mathbf{r}) d\mathbf{r} \quad (2)$$

$$J[\rho(\mathbf{r})] = \frac{1}{2} \int \int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}} d\mathbf{r}_1 d\mathbf{r}_2 \quad (3)$$

DFT aims to develop accurate approximate functionals for $T[\rho(\mathbf{r})]$ and $Q[\rho(\mathbf{r})]$. Particular attention should be devoted to the accurate expression of the kinetic energy contribution, the most significant unknown term. DFT has given reliable results in almost every field of science, i.e., the prediction of chemical reactions and their rate-determining step to design better catalysts (Rondinelli *et al.*, 2006, 2007; Di Tommaso *et al.*, 2007; Chiodo *et al.*, 2008; Rondinelli *et al.*, 2008), the elucidation of molecular activity from natural compounds to novel generation drugs (Leopoldini *et al.*, 2010; Botta *et al.*, 2011), the investigation of microscopic features of new materials and the study of enzyme structure, function and dynamics.

Beside the quest for more accurate functionals in order to improve the accuracy of molecular energy and structure data, DFT applications are limited by the size of the chemical systems under investigations. If it is common to model the active site of an enzyme by considering the amino acids involved in the mechanism of catalysis, taking in consideration the whole protein would be prohibitive. Density functional theory, using approximate exchange-correlation functionals, allows successful application of quantum mechanics to a wide range of problems in chemistry at a fraction of the computational requirements needed by traditional Hartree-Fock theory methods. However, DFT approach is still too computationally expensive for common biological macromolecular systems involving thousands of atoms (Qu *et al.*, 2013).

The previously described approaches can give high resolution structures but show a low efficiency when applied to protein molecules, which are huge complex molecules. In fact, in the context of PFP, even computational approaches, which have been studied theoretically in order to find limitation for the implemented algorithms, have shown to be intractable. In Hart and Istrail (1997), the authors show how, even a simplified version of the PFP problem, where the α -carbon atoms of the protein are mapped on the nodes of a lattice, used to discretize the conformational space, the folding problem is NP-hard. In Berger and Leighton (1998), authors analyze that the hydrophobic-hydrophilic (HP) model, one of the most popular biophysical models used for protein folding. The HP model abstracts the hydrophobic interactions by labeling amino acids as hydrophobic (H for nonpolar) and hydrophilic (P for polar). In such a model, chains of amino acids are considered as self-avoiding walks on a 3D cubic lattice, where an optimal conformation maximizes the number of adjacencies between H's. The authors show that the protein folding problem, under the HP model on the cubic lattice, is NP-complete.

A predictor is a software system that, given the primary structure of a protein as the input, returns an estimated secondary or tertiary structure as the output. The output structure yielded by a predictor is called its prediction. Prediction methods are designed to imitate the protein folding process and combine different information (e.g., evolutionary, energetic, chemical) and/or predictions (e.g., secondary structure, contact maps) in order to guess the protein conformation (Palopoli *et al.*, 2013). We will now describe the workflow of a typical computational approach which tries to approximate a solution for the PFP problem. Computational approaches trying to predict the protein structure use the primary sequence as the input and then a set of intermediate structures are predicted before proceeding with the tertiary structure prediction. Algorithms usually use the sequence of amino acids in the primary sequence to decide how to proceed in the prediction phase. Only C- α atoms in amino acids are considered, in order to simplify the model. At the end of the process, the backbone and the side chains are reconstructed by using statistical approaches to obtain a full-atoms protein structure prediction. A typical tertiary structure prediction workflow starts from the primary structure of the target and predicts a set of intermediate structures describing various chemical and structural features of the target. 1D structures characterize local features along the amino acid sequence of the target, for instance: (i) Secondary structure, which highlights high ordered subsequences along the α -carbon trace, (ii) contact and solvent accessibility, which predicts which contacts will be exposed to ligands and solvents and which will be buried inside the structure, (iii) disorder, useful to predict which contacts will be in a highly disordered state (Mirabello and Pollastri, 2013; Walsh *et al.*, 2011). 2D structures contain translation and rotation invariant topological information of the target protein, some of which are: (i) Coarse topology, is a matrix containing information about amino acids being in contact or not, (ii) contact map is also a matrix representing distances between α -carbon atoms of each amino acid along the chain, usually in 4 classes (0–8Å, 8–13Å, 13–19Å, >19Å) (Tradigo *et al.*, 2011; Tradigo, 2013) and, more recently by using continuous real values (Kukic *et al.*, 2014). All of these intermediate predicted structures are then used by the folding algorithms to reconstruct a fold of the α -carbon trace by using an heuristic (e.g., simulated annealing). The obtained minimum energy structure is finally enriched by reconstructing its backbone (Maupetit *et al.*, 2006) and its complete side chain (Krivov *et al.*, 2009; Liang *et al.*, 2011). In the following we will give some more details about this process.

The primary sequence, or sub-portions of it, are used as queries to search for homologues in databases containing known sequences (e.g., the PDB). This step has been shown to be crucial in order to exploit evolutionary information, since similar structures imply similar functions for protein molecules, due to evolutionary mechanisms, which promote changes at the morphological level, at the DNA level, at the cell pathways level and at the protein level. However, protein functions have to be maintained hence specific 3D structures are preserved, leading to protein portions known as domains. Two naturally evolved

proteins with more than 25% identical residues (length > 80 amino acids) are very likely to have similar 3D structures (Doolittle, 1986). Results obtained during this phase can be classified into three main classes (Schwede *et al.*, 2009): (i) Homology modeling, also called comparative modeling, where proteins similar to the target with known sequence have been found hence an atomic model for the target protein can be calculated from the known template structures (Petrey and Honig, 2005; Schwede *et al.*, 2003); results obtained are often good with prediction accuracy $\sim 80\%$; (ii) fold recognition, where no satisfying sequence alignment has been found but there exists known proteins with similar shape, hence information on the structure can be gathered from alignment against these known folds (Ye and Godzik, 2003); prediction accuracy lowers and is usually dependent on the implemented method; (iii) ab-initio or de-novo modeling, where none of the above applies, hence the atomic model of the target has to be derived directly from the primary sequence (Das and Baker, 2008); results in this class are usually quite poor, with a prediction accuracy below 30%.

Secondary structures are 1D predicted features allowing for a drastic cut in the phase space search for models with minimal energy. Alpha-helices and beta-sheets, whose shape is implied by hydrogen bonds among amino acids along the c-alpha trace, can in fact be considered as “fixed” geometrical elements, since they represent almost rigid structures in the real protein, due to the hydrogen bonds between the amino acids forming them. As shown In Rost, Sander, Schneider (1994), also secondary structure prediction gains from exploiting evolutionary information, bringing 68% predictive accuracy of using basic neural network approaches, to 70% accuracy, as shown in Holly and Karplus (1989).

Contact Maps are 2D predicted features which can guide the folding algorithm during the search towards the minimal energy structure. They encode a prediction for the topology of the target structure in a bi-dimensional matrix in whose elements the distance between each residue is stored. Typically, the distance is predicted in four classes, as seen above, where smaller distance classes are usually more easy to predict than larger ones. However, the prediction of contact maps represents an unbalanced problem as far fewer examples of contacts than non-contacts exist in a protein structure (Kukic *et al.*, 2014).

When all of the intermediate structures have been predicted, the folding algorithm starts from an almost linear backbone and iteratively moves each atom to a new close position which minimizes energy and does not violate the topological and energetical constraints encoded in the intermediate structures. This process is repeated until the structure total energy falls below a predefined threshold.

Analysis and Assessment

A typical concern about predicted protein structures is about their reliability and about the resolution that can be achieved with state of the art algorithms, which is of utter importance for biologists or physicians. Predicted structures can be used in many application scenarios, but much depends on how the obtained structure quality is measured, which in the past has been quite controversial (Schwede *et al.*, 2009). To this end, a dedicated research topic deals with structural quality to assess the quality of predicted structure and their potential use in various applications. When errors, usually measured in RMSD (Root-Mean-Square Deviation), in the structure are around 1Å, the obtained models can be used to study catalytic mechanisms and functions. When errors are below 3Å, the predicted structure can be used from molecular replacement studies (higher quality) to modeling to low resolution density maps (lower quality), depending on the quality. When errors are above 3Å, which is probably due to a sub-optimal template selection or an ab-initio modeling, the resulting model can be used just for domain boundaries or for the identification of structural motives.

One of the most important research initiative about finding a principled solution to the PFP is the CASP experiment (Moult *et al.*, 2014). This large international experiment involves a large scientific research community working on state of the art protein structure prediction models and well principled algorithmic approaches. At present time, CASP has been held in 12 editions, both in Europe and in the United States, with a bi-annual schedule starting from 1994. CASP is held every two years in the form of a competition, in which participants enroll to a blind challenge for the prediction of a set of few hundreds “unknown” protein targets, which will be solved with exact methods (e.g., X-ray crystallography) by the end of the experiment. Predictions can be computer- and human-based, and also combined (i.e., human-curated predictions). Also, in the competition there exist raw methods, which build the structure prediction from scratch, and also metapredictors, which integrate prediction from other tools. At the end of the experiment, all targets get resolved and a dedicated team proceeds with the assessment by comparing predicted structures with the observed ones.

Like predictors, metapredictors take a primary structure as the input and return a prediction of its secondary or tertiary structure as the output (Palopoli *et al.*, 2013). Metapredictors compute their results based on prediction results taken from other prediction tools. Intermediate predictions are then elaborated by using two main approaches, leading to two kinds of metapredictors: (i) Consensus-based methods (see Mihășan, 2010; Bujnicki and Fischer, 2008), query different prediction servers and choose the best structure according to specific metrics, but they are only able to ideally perform as good as the best predictor they use; (ii) integration methods (see Bujnicki and Fischer, 2008; Palopoli and Terracina, 2004; Palopoli *et al.*, 2009) retrieve predicted structures from several prediction servers, extract relevant structural features and produce a supposedly superior model based on them, usually showing better performances than consensus-based methods, because they are able to combine local features, but when dealing with predictions being significantly different from one another. Furthermore, they go beyond the selection of the “best models”, because they generate a completely new structure.

In order to assess predictions for the target protein, different measures are usually used to compare the predicted structure with the observed one. Q3 and SOV (Rost *et al.*, 1994; Zemla *et al.*, 1999) are numerical measures which can assess the prediction

performance of a secondary structure predictor. Q3 measures the percentage of residues assigned to the correct class (i.e., alpha-helix, beta-sheet, loop) and is the average of several measures Q_i ($i = \{\text{helix, sheet, loop}\}$), where Q_i is the percentage of residues correctly predicted in state i with respect to the total number of residues experimentally observed in state i . SOV, which stands for Segment Overlap, gives a measure of correct assignments by-segment which is less sensitive to small variations. For the assessment of contact maps predictions, X_d and Z_{acc} measures are used (Moult, 2013). X_d takes into account the correctly assigned contacts in bands from the main diagonal (0–4Å, 4–8Å, 8–12Å, etc). Z_{acc} (z-accuracy) is the percentage of correctly predicted contacts with respect to total contacts. Contact maps prediction tools are then ranked according to the Z_{total} score, which averages the X_d and Z_{acc} measures calculated on each target predicted by the tool. One of the most diffuse tools for assessing 3D structures is the TMScore tool (Zhang and Skolnick, 2007). TMScore gives a length-independent score in the range (0,1] which allows predictions to be compared to each other and is less sensitive to local variations in the structure, being a global fold similarity. Other methods for the comparison of 3D structures do exist in literature (see Cristobal *et al.*, 2001; Siew *et al.*, 2000).

See also: Biomolecular Structures: Prediction, Identification and Analyses. Computational Protein Engineering Approaches for Effective Design of New Molecules. DNA Barcoding: Bioinformatics Workflows for Beginners. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Investigating Metabolic Pathways and Networks. Protein Structural Bioinformatics: An Overview. Protein Three-Dimensional Structure Prediction. Proteomics Mass Spectrometry Data Analysis Tools. Secondary Structure Prediction. Small Molecule Drug Design. Structural Genomics. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4)

References

- Berger, B., Leighton, T., 1998. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology* 5 (1), 27–40.
- Berg, J.M., Tymoczko, J.L., Stryer, L., 2002. *Biochemistry*. Macmillan.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28, 235–242.
- Botta, C.B., Cabri, W., Cini, E., *et al.*, 2011. Oxime amides as a novel zinc binding group in histone deacetylase inhibitors: Synthesis, biological activity, and computational evaluation. *Journal of Medicinal Chemistry* 54 (7), 2165–2182.
- Bujnicki, J.M., Fischer, D., 2008. 'Meta' approaches to protein structure prediction. In: *Practical Bioinformatics*, pp. 23–34. Berlin, Heidelberg: Springer.
- Chiodo, S., Rondinelli, F., Russo, N., Toscano, M., 2008. On the catalytic role of Ge + and Se + in the oxygen transport activation of N2O by CO. *Journal of Chemical Theory and Computation* 4 (2), 316–321.
- Crick, F., 1970. Central dogma of molecular biology. *Nature* 227 (5258), 561–563.
- Cristobal, S., Zemla, A., Fischer, D., Rychlewski, L., Elofsson, A., 2001. A study of quality measures for protein threading models. *BMC Bioinformatics* 2 (1), 5.
- Das, R., Baker, D., 2008. Macromolecular modeling with rosetta. *Annual Review of Biochemistry* 77, 363–382.
- Dill, K.A., MacCallum, J.L., 2012. The protein-folding problem, 50 years on. *Science* 338 (6110), 1042–1046.
- Di Tommaso, S., Marino, T., Rondinelli, F., Russo, N., Toscano, M., 2007. CO2 activation by Nb + and NbO + in the gas phase. A case of two-state reactivity process. *Journal of Chemical Theory and Computation* 3 (3), 811–815.
- Doolittle, R.F., 1986. Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences. University Science Books.
- Griep, S., Hobohm, U., 2009. PDBselect 1992–2009 and PDBfilter-select. *Nucleic Acids Research* 38 (suppl. 1), D318–D319.
- Hart, W.E., Istrail, S., 1997. Robust proofs of NP-hardness for protein folding: General lattices and energy potentials. *Journal of Computational Biology* 4 (1), 1–22.
- Holley, L.H., Karplus, M., 1989. Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences* 86 (1), 152–156.
- Koonin, E.V., 2012. Does the central dogma still stand? *Biology Direct* 7 (1), 27.
- Krivov, G.G., Shapovalov, M.V., Dunbrack, R.L., 2009. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics* 77 (4), 778–795.
- Kukic, P., Mirabello, C., Tradigo, G., *et al.*, 2014. Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. *BMC Bioinformatics* 15 (1), 6.
- Leopoldini, M., Rondinelli, F., Russo, N., Toscano, M., 2010. Pyrananthocyanins: A theoretical investigation on their antioxidant activity. *Journal of Agricultural and Food Chemistry* 58 (15), 8862–8871.
- Liang, S., Zheng, D., Zhang, C., Standley, D.M., 2011. Fast and accurate prediction of protein side-chain conformations. *Bioinformatics* 27 (20), 2913–2914.
- Mardirossian, N., Head-Gordon, M., 2017. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals. *Molecular Physics* 115 (19), 2315–2372.
- Maupetit, J., Gautier, R., Tuffery, P., 2006. SABBAC: Online structural alphabet-based protein backbone reconstruction from alpha-carbon trace. *Nucleic Acids Research* 34 (Suppl. 2), W147–W151.
- Medvedev, M.G., Bushmarinov, I.S., Sun, J., *et al.*, 2017. Density functional theory is straying from the path toward the exact functional. *Science* 355 (6320), 49–52.
- Mihășan, M., 2010. Basic protein structure prediction for the biologist: A review. *Archives of Biological Sciences* 62 (4), 857–871.
- Mirabello, C., Pollastri, G., 2013. Porter, PaleAle 4.0: High-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics* 29 (16), 2056–2058.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2014. Critical assessment of methods of protein structure prediction (CASP)-round X. *Proteins: Structure, Function, and Bioinformatics* 82 (S2), S1–S6.
- Murzin, A.G., 2008. Metamorphic proteins. *Science* 320 (5884), 1725–1726.
- Palopoli, L., Rombo, S.E., Terracina, G., Tradigo, G., Veltri, P., 2009. Improving protein secondary structure predictions by prediction fusion. *Information Fusion* 10 (3), 217–232.
- Palopoli, L., Rombo, S.E., Terracina, G., Tradigo, G., Veltri, P., 2013. Protein structure metapredictors. In: *Encyclopedia of Systems Biology*, pp. 1781–1785. New York: Springer.
- Palopoli, L., Terracina, G., 2004. Coopps: A system for the cooperative prediction of protein structures. *Journal of Bioinformatics and Computational Biology* 2 (03), 471–495.
- Petrey, D., Honig, B., 2005. Protein structure prediction: inroads to biology. *Molecular cell* 20 (6), 811–819.

- Qu, X., Latino, D.A., Aires-de-Sousa, J., 2013. A big data approach to the ultra-fast prediction of DFT-calculated bond energies. *Journal of Cheminformatics* 5 (1), 34.
- Reddy, B.V., Bourne, P.E., 2003. Protein structure evolution and the SCOP database. *Structural Bioinformatics* 44, 237–248.
- Rondinelli, F., Russo, N., Toscano, M., 2006. CO₂ activation by Zr⁺ and ZrO⁺ in gas phase. *Theoretical Chemistry Accounts* 115 (5), 434–440.
- Rondinelli, F., Russo, N., Toscano, M., 2007. On the origin of the different performance of iron and manganese monocations in catalyzing the nitrous oxide reduction by carbon oxide. *Inorganic Chemistry* 46 (18), 7489–7493.
- Rondinelli, F., Russo, N., Toscano, M., 2008. On the Pt⁺ and Rh⁺ catalytic activity in the nitrous oxide reduction by carbon monoxide. *Journal of Chemical Theory and Computation* 4 (11), 1886–1890.
- Rost, B., Sander, C., Schneider, R., 1994. Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology* 235 (1), 13–26.
- Schwede, T., Kopp, J., Guex, N., Peitsch, M.C., 2003. SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Research* 31 (13), 3381–3385.
- Schwede, T., Sali, A., Honig, B., *et al.*, 2009. Outcome of a workshop on applications of protein models in biomedical research. *Structure* 17 (2), 151–159.
- Siew, N., Elofsson, A., Rychlewski, L., Fischer, D., 2000. MaxSub: An automated measure for the assessment of protein structure prediction quality. *Bioinformatics* 16 (9), 776–785.
- Tradigo, G., 2013. Protein contact maps. In: *Encyclopedia of Systems Biology*, pp. 1771–1773. New York: Springer.
- Tradigo, G., Velti, P., Pollastri, G., 2011. Machine learning approaches for contact maps prediction in CASP9 experiment. In: SEBD, pp. 311–317.
- UniProt Consortium, 2016. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Walsh, I., Martin, A.J., Di Domenico, T., *et al.*, 2011. CSpritz: Accurate prediction of protein disorder segments with annotation for homology, secondary structure and linear motifs. *Nucleic Acids Research* 39 (suppl. 2), W190–W196.
- Ye, Y., Godzik, A., 2003. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics* 19 (suppl_2), ii246–ii255.
- Zemla, A., Venclovas, Č., Fidelis, K., Rost, B., 1999. A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics* 34 (2), 220–223.
- Zhang, Y., Skolnick, J., 2007. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics* 68 (4), 1020.

Further Reading

- Berg, J.M., Tymoczko, J.L., Stryer, L., 2002. *Biochemistry*. Macmillan.
- Gu, J., Bourne, P.E. (Eds.), 2009. *Structural Bioinformatics*, vol. 44. John Wiley & Sons.
- Tramontano, A., Lesk, A.M., 2006. *Protein Structure Prediction*. Weinheim: John Wiley and Sons, Inc.

Relevant Websites

- <http://predictioncenter.org/>
CASP website.
- <http://distillf.ucd.ie/>
Distill predictor website.
- <https://www.rcsb.org/>
RCS PDB Website.
- <http://scop.mrc-lmb.cam.ac.uk/scop/>
SCOP database website.

Biographical Sketch

Giuseppe Tradigo is a postdoc at the DIMES Department of Computer Science, Models, Electronics and Systems Engineering, University of Calabria, Italy. He has been a Research Fellow at University of Florida, Epidemiology Department, US, where he worked on a GWAS (Genome-Wide Association Study) project on the integration of complete genomic information with phenotypical data from a large patients dataset. He has also been a visiting research student at the AmMBio Laboratory, University College Dublin, where he participated to the international CASP competition with a set of servers for protein structure prediction. He obtained his PhD in Biomedical and Computer Science Engineering at University of Catanzaro, Italy. His main research interests are big data and cloud models for health and clinical applications, genomic and proteomic structure prediction, data extraction and classification from biomedical data.

Francesca Rondinelli is a young researcher. She obtained her PhD in Theoretical Chemistry at University of Calabria, Dept. of Chemistry. She has been visiting research student at KTH Royal Institute of Technology in Stockholm, Department of Chemistry. She has been a postdoc both at University of Calabria and at University of Naples, Federico II. Her research interest go from cyclodextrins, principled drug design and CO₂ activation.

Gianluca Pollastri is an Associate Professor in the School of Computer Science and a principal investigator at the Institute for Discovery and at the Institute for Bioinformatics at University College Dublin. He was awarded his M.Sc. in Telecommunication Engineering by the University of Florence, Italy, in 1999 and his PhD in Computer Science by University of California at Irvine in 2003. He works on machine learning and deep learning models for structured data, which he has applied to a cohort of problems in the bioinformatics and chemoinformatics space. He has developed some of the most accurate servers for the prediction of functional and structural features of proteins, which have processed over a million queries from all over the world and have been licensed to 75 different subjects, including pharmaceutical companies. His laboratory at UCD has been funded by Science Foundation Ireland, the Health Research Board, the Irish Research Council, Microsoft, UCD, the King Abdulaziz City for Science and Technology (Saudi Arabia) and Nvidia.

Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins

Marco Wiltgen, Graz General Hospital and University Clinics, Graz, Austria

© 2019 Elsevier Inc. All rights reserved.

Introduction

Protein structure is uniquely determined by its amino acid sequence. Knowledge of protein 3D structure is of crucial importance for understanding protein function, dynamics and interactions with ligands and other proteins (Lesk, 2001, 2002; Gibas and Jambeck, 2001; Chang, 2005). A protein cannot be seen, for example, by a microscope with X-ray focussing lenses. Therefore there exists no real image, such as a microscopic view of a cell, of a protein. Instead we must use structural models of the proteins, based on X-ray diffraction data, NMR data etc. (Wiltgen, 2009). Current experimental structure determination methods such as X-ray crystallography and nuclear magnetic resonance (NMR) are slow, expensive, and often difficult to perform (Kendrew *et al.*, 1960). X-ray crystallography delivers high resolution protein structures whereas NMR is restricted to limited size proteins in solution resulting in low-resolution structures. Many proteins are too large for NMR structure determination and others cannot be crystallized for X-ray diffraction. In addition in the experimental determination of a protein structure there occasionally arise problems concerning the cloning, expression, purification and crystallization of the protein. If the experimental techniques fail then protein modelling on a computer is the only way to obtain structural information (Chou, 2006). Although experimental methods are nowadays in some sense high throughput processes, they cannot keep up with the growth rate of known protein sequences. For example the protein structural database (PDB) today contains (06/2017) 131,485 macromolecular structures (Berman *et al.*, 2000). Of these, 36,528 entries are from *Homo sapiens*, 8563 from *Escherichia coli*, 5582 from *Mus musculus*, 3712 from *Saccharomyces cerevisiae* and so on (it should be noted that these numbers includes multiple solutions of the same structure). In contrast the UniProtKB/TrEMBL database contains 87,291,332 sequence entries. Therefore computational methods for the prediction of protein structure have gained much interest.

Among computational protein modelling methods, homology modelling, also known as template-based modelling (TBM) usually provides the most reliable results (Xu *et al.*, 2000; Kryshchak *et al.*, 2017). The method is based on the observation that two proteins belonging to the same family, and therefore sharing very similar amino acid sequences, have similar three-dimensional structures (Sanchez and Sali, 1997a, b). In homology modelling, a protein sequence with an unknown structure is aligned with sequences that have known protein structures (Schwede *et al.*, 2000). By exploiting structural information from the known configurations, the new structure can be predicted. Homology modelling is based on two important observations in nature: The structure of a protein is determined by its amino acid sequence, and the structure itself is more conserved than the sequence during evolution. Local changes in the amino acid sequence often cause only small structural changes. Therefore homology modelling first involves the finding of already known homologous protein structures and then building the query (target) sequence into the homologous template structures. Because most protein families contain at least one member with a known structure, the applicability of homology modelling is steadily increasing.

Homology-based models can be used to guide the design of new experiments, such as site-directed mutagenesis, and are valuable in structure-based drug discovery and design. In structural bioinformatics many public software tools and data bases are used for the analysis of macromolecules. Several of them are presented in this article.

Bioinformatics Tools for Protein Structure Prediction and Analysis

A number of bioinformatics tools for proteomic analysis are available at the ExPASy server at website provided in “Relevant Websites section” The software tools include functions such as protein identification and characterization such as: predicted isoelectric point, molecular weight, amino acid composition and predicted mass spectra, and others (Gasteiger *et al.*, 2003). Additionally similarity searches using pattern and profile searches can be used to identify homologous or structurally similar protein structures.

Two popular programs for homology modelling, that are free for academic research, are MODELLER (see “Relevant Websites section”) and SWISS-MODEL (See “Relevant Websites section”). The Modeller program can be used as standalone software whereas SWISS-MODEL is a Web-based fully automated homology modelling server (Eswar *et al.*, 2008; Webb and Sali, 2014). Homology modelling is initiated by inserting the target sequence into the entry field of the SWISS-MODEL server (Schwede *et al.*, 2003). After selecting the templates from a list proposed by the system, all the subsequent steps in homology modelling are carried out automatically. SWISS-MODEL is accessible via the ExPASy web server, or from the viewer program DeepView (Swiss PDB viewer). The purpose of this server is to make protein modelling accessible to all biochemists and molecular biologists worldwide. The Swiss PDB viewer can be freely downloaded from the Swiss PDB viewer server (see “Relevant Websites section”) and used for molecular analysis and visualization (Guex and Peitsch, 1997). As input, the program uses coordinate files from protein structures at the PDB database for the template structures. The atomic coordinates are then converted into a view of the protein. Some additional software tools for protein structure prediction based on homology modelling and threading are listed in Table 1 (this list is far from complete).

Table 1 Listed are some representative examples of Template Based Modelling (TBM) and Free Modelling (FM) software tools (also known as *ab initio* modelling)

Name	Method	Description	Web address
Swiss-Model	TBM	Webserver	http://swissmodel.expasy.org/
Modeller	TBM	Standalone	http://salilab.org/modeller/
ModWeb	TBM	Webserver	https://modbase.compbio.ucsf.edu/scgi/modweb.cgi
RaptorX	TBM	Webserver	http://raptorx.uchicago.edu/
HHpred	TBM	Webserver	http://hhpred.tuebingen.mpg.de/hhpred
Phyre2	TBM	Webserver	http://www.sbg.bio.ic.ac.uk/~phyre2/html/
I-TASSER	TBM + FM	Standalone	http://zhanglab.ccmb.med.umich.edu/I-TASSER/
Rosetta	FM	Standalone	http://www.rosettacommons.org/software/
Robetta	FM	Webserver	http://robetta.bakerlab.org/
QUARK	FM	Webserver	http://zhanglab.ccmb.med.umich.edu/QUARK/

The entry point to the structural protein database is the PDB web site: website provided in “Relevant Websites section”. The search for a particular protein structure of interest can be initiated by entering the 4 letter PDB identification code at the PDB main page (Sussman *et al.*, 1998). Alternatively, the PDB can be searched by use of keywords. Convenient access to the PDB and many other databases is enabled by the *Entrez* integrated search and retrieval system of the NCBI (National Centre for Biotechnology Information, see “Relevant Websites section”). Structures in the structural database can be identified by searching using specific keywords such as the name of the protein or organism, or other identifying features such as the names of domains, folds, substrates, etc. (Baxeianis, 2006). Keywords can be used to search for information in the most important fields in the PDB data header. The advantage of access via NCBI is the availability of public domain tools like BLAST (Basic Local Alignment Search Tool) which enables the user to compare an uncharacterized protein sequence with the whole sequence database (Altschul *et al.*, 1990, 1997).

Protein Families and the SCOP Database

Proteins always exist in aqueous solutions. To minimize the energy cost of solvating a protein, it is favourable that hydrophilic amino acids are arranged to be solvent exposed at the protein surface, whereas the hydrophobic amino acids are mainly concentrated in the protein core. This may limit the number of occurring structure motifs (folds) which are repetitively used in different proteins. Therefore the evolution of proteins is restricted to a finite set of folds resulting in proteins with structural (native conformation) and sequence similarities (Hubbard *et al.*, 1997).

The Structural Classification of Proteins (SCOP) database provides a detailed and comprehensive description of the structural and evolutionary relationships between all proteins whose structure is known. The SCOP is a database maintained by the MRC Laboratory of Molecular Biology, UK (see “Relevant Websites section”). (The prototype of a new Structural Classification of Proteins 2 (SCOP2) is available at see “Relevant Websites section”). Proteins are classified in a hierarchical way (Murzin *et al.*, 1995). At the top of the hierarchy are classes, followed by folds, superfamilies and families (Fig. 1). At the class level, folds are characterized by their secondary structure and divided into all-alpha (α), all-beta (β) or mixed alpha-beta (α/β , $\alpha + \beta$) structures.

Families: Proteins with a close common evolutionary origin are clustered together in families. The members of a family have significant sequence similarity leading to related structures and functions. The protein molecules have a clear homology.

Superfamilies: If the proteins of different families have only low sequence similarity, but their structures are similar (which indicates a possible common evolutionary origin), then these families are clustered together as a superfamily.

Folds: If the proteins included in superfamilies have the same major secondary structures arranged in the same way and with the same topological connections then they belong to the same fold (similar topology).

Class level: The different folds (with different characteristics) are clustered into classes. Thereby the folds are assigned to one of the following structural classes: folds whose structure contains essentially only α -helices, folds whose structure is essentially composed of only β -sheets, and folds composed of both α -helices and β -sheets.

In addition to the SCOP database, the CATH database provides a similar hierarchical classification of protein domains based on their folding patterns (Knudsen and Wiuf, 2010). The 4 main levels of the CATH hierarchy are: Class (equivalent to the class level in SCOP), Architecture (equivalent to the fold level in SCOP), Topology, and Homologous superfamily (equivalent to the superfamily level in SCOP).

Proteins that have grossly similar structures are classified as more closely related (Andreeva *et al.*, 2008). This plays an important role in the search and detection of homology. In the SCOP database, the amino acid sequence identity within a protein family must be at least > 15%. The prerequisite for successful homology modelling is sequence identity of at least 25%–30% between the target sequence and the template sequences. Therefore homology modelling works very well for proteins inside a family.

Substitution Matrix and Sequence Alignment

Sequence alignment is a way of arranging protein (or DNA) sequences to identify regions of similarity that may be a consequence of evolutionary relationships between the sequences. The rate at which an amino acid in a protein sequence changes to another

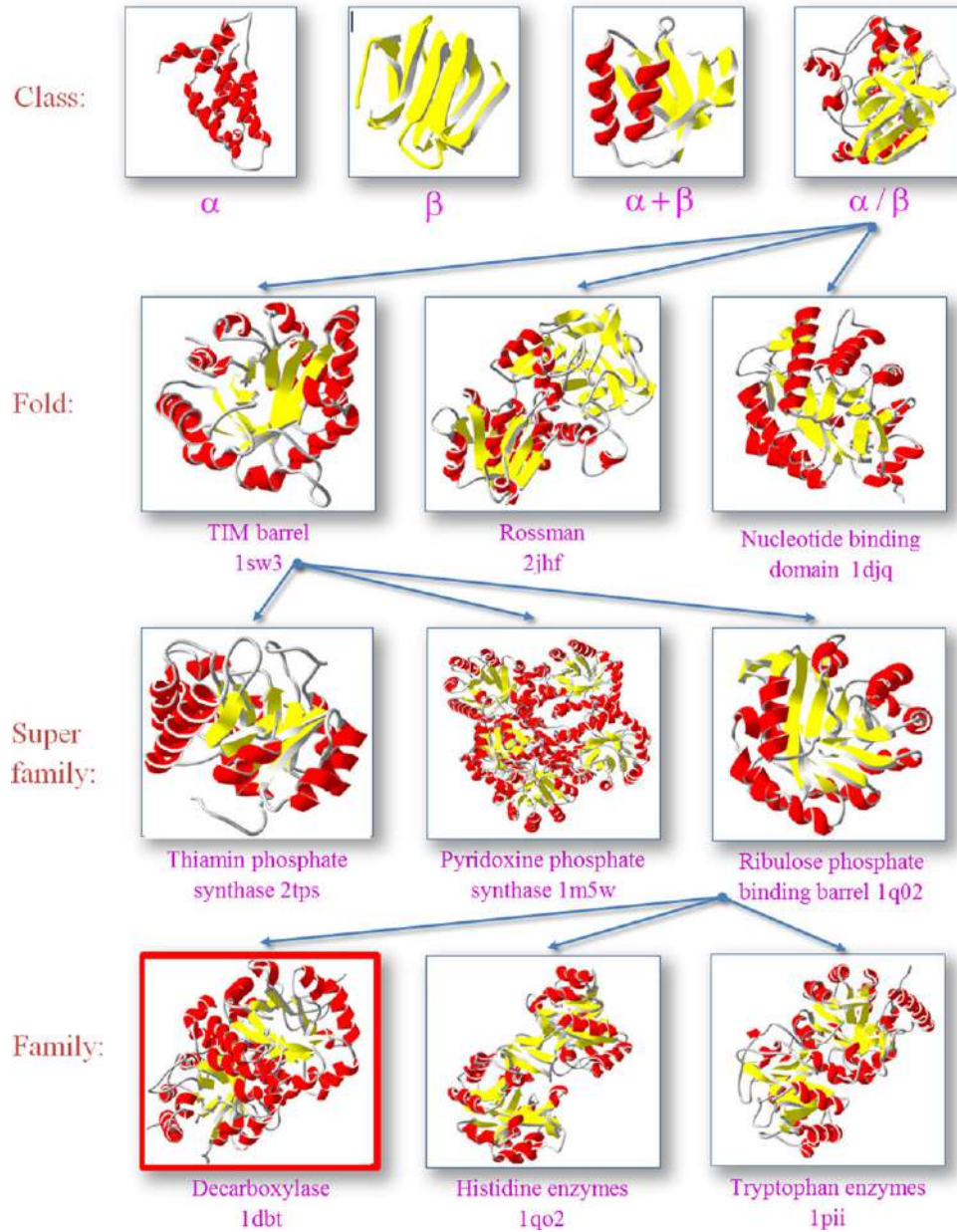


Fig. 1 The Structural Classification of Proteins (SCOP).

over time is described by substitution matrices. The elements of a (20×20) substitution matrix define the score for the alignment of any two of the 20 amino acids (Henikoff and Henikoff, 1992). Because the score for aligning residues A and B is normally the same as for B and A, the substitution matrix is symmetric.

The alignment score describes the overall quality of a sequence alignment. Higher numbers correspond to higher similarity. The calculation of the score is as follows. Given a sequence pair (S_1, S_2) of length N respectively M:

$$S_1 = a_1 a_2 \dots a_i \dots a_N \quad ; \quad S_2 = b_1 b_2 \dots b_j \dots b_M$$

The symbols in the sequences are elements of the set of amino acids:

$$a, b \in \{ \underbrace{\{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}}_{20 \text{ amino acids (1 letter symbol)}} \}$$

The indexes i and j refer to the position of an amino acid a_i and b_j in the respective sequences. To determine a score for the alignment we first consider the case that the two sequences are not related, and then the case that the sequences are related, which means they are descended from a common ancestor. Assuming the amino acid residues are randomly distributed, the probability

of an alignment of two sequences that are not related is given by the product of the probabilities of the independent amino acids:

$$P(R; S_1, S_2) = \prod_i q_{a_i} \cdot \prod_j q_{b_j}$$

The probability q_{a_i} is called the background frequency and reflects the natural occurrence of the amino acid a at position i . The alignment is given by chance (R : randomly). If the two sequences are related (belonging to the same family) the residue pairs in the alignment occur with a joint probability, p_{ab} . This probability reflects the natural exchange rate, over evolutionary time, of the considered amino acids in related proteins. In other words, p_{ab} describes the probability that the residues a and b are descended from a common ancestor.

The probability for a successful matching (M) of the two sequences is given by:

$$P(M; S_1, S_2) = \prod_i p_{a_i b_i}$$

The ratio of the probabilities for the exchange of residues a and b in the related and unrelated models is given by the odds-ratio:

$$\frac{P(M; S_1, S_2)}{P(R; S_1, S_2)} = \frac{\prod_i p_{a_i b_i}}{\prod_i q_{a_i} \cdot \prod_j q_{b_j}} = \prod_i \frac{p_{a_i b_i}}{q_{a_i} q_{b_j}}$$

To obtain an additive scoring system we take the logarithm of the odds ratio:

$$S = \log \prod_i \frac{p_{a_i b_i}}{q_{a_i} q_{b_j}} = \sum_i s(a_i, b_i)$$

The value S is called the log-odds ratio and $s(a_i, b_i)$ is the log-odds ratio for the residue pair a, b and is given by [Dayhoff et al. \(1978\)](#):

$$s(a, b) = \log \left(\frac{p_{ab}}{q_a q_b} \right)$$

If the joint probability is greater than the product of the background frequencies ($p_{ab} > q_a q_b$) then the value of the log-odds ratio is positive. If $q_a q_b > p_{ab}$ then the value is negative. The raw scores for every possible pairwise combination of the 20 amino acids are entered in the substitution matrix (also called scoring matrix). As an example the BLOSUM 62 (BLOCKS SUBstitution Matrix) matrix is shown in [Fig. 2](#). The log-odds ratio evaluates the probability that the residue pair a, b originates from an amino acid exchange in the equivalent position of an ancestral protein relative to the probability that the correlation is only by chance. The score of the alignment is then the sum of the individual log-odds ratios for every residue-pair in the alignment. If a gap occurs in the alignment, a penalty is subtracted from the total score. The above considerations about the probabilities of amino acid exchanges are the empirical basis for the understanding of substitution matrices, the values themselves are determined from

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R		5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N			6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D				6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C					9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q						5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E							5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G								6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H									8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I										4	2	-3	1	0	-3	-2	-1	-3	-1	3
L											4	-2	2	0	-3	-2	-1	-2	-1	1
K												5	-1	-3	-1	0	-1	-3	-2	-2
M													5	0	-2	-1	-1	-1	-1	1
F														6	-4	-2	-2	1	3	-1
P															7	-1	-1	-4	-3	-2
S																4	1	-3	-2	-2
T																	5	-2	-2	0
W																		11	2	-3
Y																			7	-1
V																				4

Fig. 2 BLOSUM 62 is a typical residue exchange or substitution matrix used by alignment programs. The values are shown in log-odds form based on a random background model.

Atom		Amino Acid		Coordinates			B-Factor	
Nr	Name	Name	Nr.	X	Y	Z		
51	CG1	VAL	A	6	20.917	64.652	7.314	1.00 21.94
52	CG2	VAL	A	6	19.040	65.291	5.669	1.00 22.81
53	N	ILE	A	7	19.628	65.019	10.345	1.00 20.41
54	CA	ILE	A	7	20.432	65.105	11.566	1.00 19.80
55	C	ILE	A	7	21.664	64.214	11.363	1.00 20.37
56	O	ILE	A	7	21.536	62.996	11.219	1.00 21.25
57	CB	ILE	A	7	19.674	64.605	12.799	1.00 19.60
58	CG1	ILE	A	7	18.336	65.307	12.935	1.00 20.34
59	CG2	ILE	A	7	20.529	64.864	14.080	1.00 18.44
60	CD1	ILE	A	7	17.422	64.684	13.996	1.00 19.33
61	N	VAL	A	8	22.856	64.818	11.327	1.00 19.83
62	CA	VAL	A	8	24.064	64.071	11.072	1.00 19.77

Fig. 3 The structural information is stored in the PDB atomic coordinates. The files have a standard format which can be read by most viewers and other protein-related software.

experimental data. A high alignment score indicates that the two sequences are very similar. The values along the diagonal in the substitution matrix are highest because, at relatively short evolutionary distances, identical amino acid residues are more likely to be derived from a common ancestor than to be matched by chance. Two residues have a positive score if they are frequently substituted for one other in homologous proteins. The alignment between residues that are rarely substituted in evolution, results in a lower (or negative) score.

The Protein Data Bank

Protein structural information is publicly available at the protein data bank (PDB), an international repository for 3D structure information (Sussman *et al.*, 1990; Rose *et al.*, 2017). At the moment, PDB contains more than 123,000 protein structures. The structural information is stored as atomic coordinates (Fig. 3). The B-factor, also called the temperature factor, defines the modelled isotropic thermal motion. Beside the B-factor, the occupancy value is stored in the PDB file. For structure determination with X-ray diffraction, macromolecular crystals are used. These crystals are composed of individual molecules which are symmetrically arranged. Because side chains on the protein surface may be differently orientated, or substrates may bind in different orientations in an active site, slight differences between the molecules in the crystal are possible. The occupancy is used to estimate the amount of conformations in the crystal. For most atoms, the occupancy value is 1.00, indicating that the atom is in all of the molecules in the same place in the crystal.

The database can be accessed via the Internet and the selected PDB data files, containing the atomic coordinates, are downloadable (Westbrook and Fitzgerald, 2003). Appropriate viewer programs, such as the Swiss PDB viewer, convert the atomic coordinates into a view of the protein. The coordinates of the templates are used for the homology modelling.

Homology Modelling

During protein evolution, protein structures show higher conservation than the corresponding sequences, so that distantly related sequences still fold into similar structures (Johnson *et al.*, 1994; Kaczanowski and Zielenkiewicz, 2010). Homology modelling is based on the principle that homologous proteins tend to have similar structures. Homology-based modelling is a knowledge-based prediction of protein structures that uses parameters extracted from existing structures to predict a new structure from its sequence (Sanchez and Sali, 2000). When no homologous sequences with known structure can be identified, then *ab initio* structure prediction, based on the first principle laws of physics and chemistry is used (Table 1). Thereby only relatively small protein structures can be predicted.

Therefore the determination of the tertiary structure of a given protein sequence (target) is based on an alignment of the target sequence with one or more sequences with known protein structures (templates). Hence homology modelling first involves the finding of known protein template structures, and then the target sequence is built into the homologous template structures. The steps required in homology modelling are as follows:

- 1) Identification of template structures: The target sequence (with unknown structure) is used as a query to find homologous sequences with known structures (the templates).
- 2) Sequence alignment: The amino acid sequences of the target and the templates are brought into an optimal (multiple) alignment.
- 3) Backbone generation: Information from the template backbones is used to model the structural backbone of the target sequence.
- 4) Generation of loops: Loop-modelling procedures are used to fill gaps in the alignment.

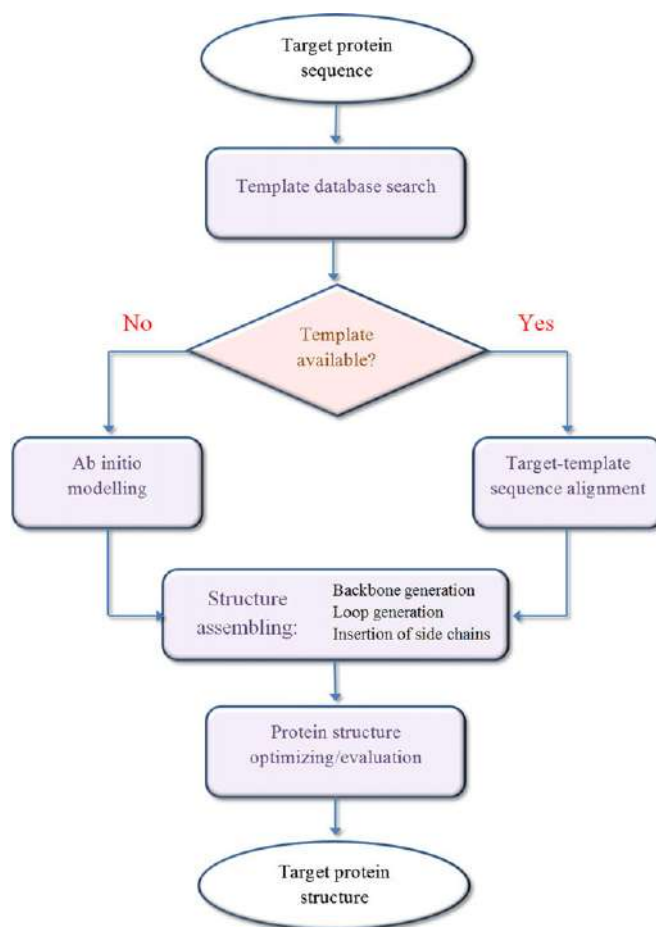


Fig. 4 Pipeline of a composite protein structure prediction: If homologous structures are available the prediction starts with an alignment of the target sequence and template sequences. If no homologous structures are available, then *ab initio* modelling is applied.

- 5) Insertion of side chains: The amino acid side chains are added to the backbone and their positions are optimized.
- 6) Model optimization: The generated structure is optimized by energy minimization

Homology modelling is based on a biological perspective: Homologous proteins have evolved by molecular evolution from a common ancestor. By identifying homology, the structure and function of a new protein can be predicted based on the homolog (Fig. 4). There is also a physical perspective: The structure of a protein corresponds to a global free energy minimum of the protein and solvent system. A compatible fold can be determined by threading the protein sequence through a library of folds, and empirical energy calculations used to evaluate compatibility (Ambrish and Yang, 2012).

If a homology between the target sequence and the template sequence is detected, then structural similarity can be assumed. In general, at least 30% sequence identity is required to generate a useful model. If the sequence identity in the alignment is below 30%, depending on the number of aligned amino acid pairs, then they fall in the twilight zone and random alignments begin to appear (Fig. 5).

Principles of Homology Modelling

In order to illustrate the principles of homology modelling, I present an example of predicting the structure of orotidine 5'-monophosphate decarboxylase from its sequence (Harris *et al.*, 2002). Orotidine 5'-monophosphate (OMP) decarboxylase is an enzyme which is essential for the biosynthesis of the pyrimidine nucleotides (cytosine, thymine, and cytosine uracil). It catalyzes the decarboxylation of orotidine monophosphate to form uridine monophosphate. OMP-decarboxylase belongs to the decarboxylase protein family, which is highlighted in Fig. 1.

Search for homologous template sequences

An optimal alignment can be calculated by the dynamic programming algorithm (Needleman and Wunsch, 1970). Such global alignments are mainly used for sequences of similar lengths and where a strong sequence homology is expected. (In contrast to most modern dynamic programming tools, the Needleman-Wunsch method uses a length independent penalty). To identify sequence

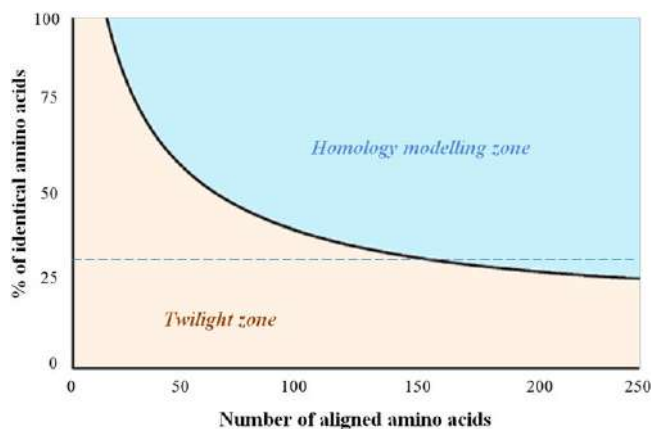


Fig. 5 The two zones of sequence alignments. Two sequences fold with high probability into a similar structure if their length and percentage sequence identity fall into the homology modelling zone.

motifs in protein sequences, local alignments are used (Smith and Waterman, 1981). Thereby the similarities of local regions in the sequences, which may differ in the rest of the sequence, are evaluated. In practice the methods of dynamic programming are too slow for searching the fast growing sequence databases. Therefore heuristic approaches are used instead of optimal solutions.

BLAST (Basic Local Alignment Search Tool) is a heuristic algorithm for comparing protein or DNA sequences (Altschul *et al.*, 1990). BLAST approximates the Smith-Waterman algorithm. A BLAST search compares a query sequence to sequences within sequence databases and retrieves sequences that resemble the query sequence, above a certain threshold. The basic idea of BLAST is that a significant alignment of two sequences contains short sections with a high score, called high-scoring segment pairs (HSP). First, from the protein query sequence a list of short words (containing 3 residues) is generated:

```
Sequence :  --LDYHNR--
word1 :    LDY
word2 :    DYH
word3 :    YHN
word4 :    HNR
```

These words are used for the database search. Actually, a key concept of BLAST is that it is not just identical matching; a neighborhood of matching words is generated for each word in the query sequence. If a word in the list is found in the database, then starting from the word the alignment between the query and target sequence is enlarged in both directions until a maximal score is reached (Fig. 6). The detected HSP's are then entered into the list of results. For a HSP to be selected its score must exceed a cut-off score S that is given by the predefined expectation-value (see below). To evaluate the significance of a BLAST search, the following question is of special importance: Given a particular scoring system, how many unrelated sequences would achieve an equal or higher score?

Almost all the relevant statistics for local alignment scores can be understood in terms of the expectation-value. The expectation-value (E-value) is the number of distinct alignments, with a score greater than or equal to S that are expected to occur in a database search by chance. The lower the E value, the more significant the alignment is. The E-value of an (un-gapped) alignment, between sequences of length m , respectively length n , is related to the score S by the Karlin-Altschul equation:

$$E = Kmn e^{-\lambda S}$$

Where S is the raw score of an alignment, obtained by summing the scores of each pair of amino acids in the alignment (see "Substitution matrix and sequence alignment"), and K and λ are constants that depend on the scoring matrix (Karlin and Altschul, 1990). The probability of finding at least one alignment with a score $\geq S$ is given by:

$$p = 1 - e^{-E}$$

This is called the p-value associated with S . The normalized bit-score S' is a rescaled version of the raw alignment score, expressed in bits of information. The parameters K and λ are folded into the bit-score by the following linear transformation:

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

(In current implementations of BLAST, the alignment score reported is the normalized bit score). By use of the bit-score the equation for the E-value reduces to:

$$E = mn 2^{-S'}$$

The orotidine 5'-monophosphate decarboxylase sequence is used as a query to search the PDB database for homologous sequences with known structures. To this end, the program BLAST was used, wherein the query sequence is entered via a simple web form (see "Relevant Websites section"). The search is restricted to the PDB database and therefore only sequences with known

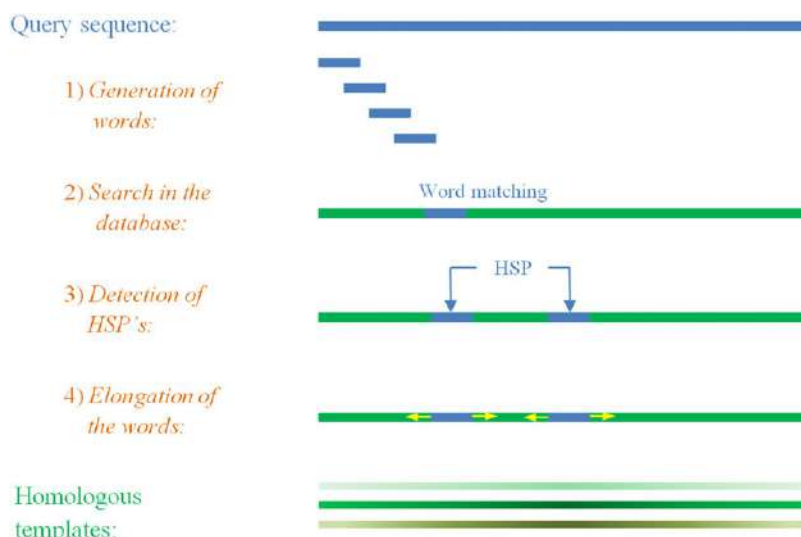


Fig. 6 BLAST search procedure.

Table 2 Template sequences

<i>OMP-decarboxylase</i>	<i>PDB code</i>	<i>Expectation value E</i>	<i>Residue identity</i>	<i>Similarity</i>	<i>Length</i>	<i>Gaps</i>
<i>Vibrio Cholera</i>	3LDV_A	5e-117	70%	80%	231	0%
<i>Coxiella Burnetii</i>	3TR2_A	7e-73	49%	64%	219	0%
<i>Lactobacillus Acidophilus</i>	3TFX_A	6e-50	39%	56%	231	2%

structures are taken into consideration. Blast compares the query sequence to all the sequences of known structures in the PDB database. The BLAST program lists several alignments of the query sequence with subject sequences with E-values below the selected threshold; The alignments are sorted by the E-values. Alignments with low E-values are very significant, which means that the sequences are likely to be homologous.

As a result, the sequences of orotidine 5'-monophosphate decarboxylase from *Vibrio cholerae*, *Lactobacillus acidophilus*, and *Coxiella Burnetii* were selected as likely to be homologs (Franklin *et al.*, 2015). The amino acid identity for all the template sequences is above 30% (Table 2). Similarity indicates the percentage of aligned amino acids that are not identical, but have positive values in the substitution matrix, and therefore frequently substituted. An optimal alignment shows a considerable number of identical or similar residues and only infrequent and small gaps.

Multiple alignments

A multiple sequence alignment is the alignment of three or more amino acid (or nucleic acid) sequences (Wallace *et al.*, 2005; Notredame, 2007). Multiple sequence alignments provide more information than pairwise alignments since they show conserved regions within a protein family which are of structural and functional importance.

Fig. 7 shows the target sequence arranged in a multiple alignment with the template OMP-decarboxylases from *V. cholerae*, *L. acidophilus*, and *C. burnetii*. The alignment was made with the MULTALIN multiple alignment tool (Corpet, 1988). The sequence alignment is used to determine the equivalent residues in the target and the template proteins.

The corresponding superposition of the template structures is shown in Fig. 8. After a successful alignment has been found, the actual model building can start.

Backbone generation

Proteins are polymers of amino acids, which consist of an amino group, a carboxyl group, and a variable side chain. The amino acids are connected by a peptide bond between the amino group and the carboxyl group of adjacent amino acid residues in the polypeptide chain. The amide nitrogen, α -carbon, and carbonyl carbon, are referred to as the backbone.

Creating the backbone scaffold can be trivially done by simply copying the coordinates of the template residues. More sophisticated procedures exploit structural information from combinations of the template backbones. For example, the target backbone can be modelled by averaging the backbone atom positions of the template structures, weighted by the sequence similarity to the target sequence (SWISSMODEL, Peitsch, 1997). Fig. 9 shows the target backbone superimposed on a ribbon diagram of one of the template structures.

Other homology modelling procedures rely on rules based on spatial restraints such as spacing between atoms, bond lengths, bond angles, and dihedral angles (Sali and Blundell, 1993). These rules are derived from observed values in known protein structures.

Target	VTNSP-VVVA	LDYHNRDDAL	AFVDKI-DPR	DCRLKVGKEM	FTLFGPQFVR
3LDV A	AMNDPKVIVA	LDYDNLADAL	AFVDKI-DPS	TCRLKVGKEM	FTLFGPDFVR
3TR2 A	---DPKVIVA	IDAGTVEQAR	AQINPL-TPE	LCHLKIGSIL	FTRYGPAFVE
3TFX A	---DRPVIVA	LDLDNEEQLN	KILSKLGDPH	DVFVKVGXEL	FYNAGIDVIK
Target	ELQQRGFDIF	LDLKFHDIPN	TAAHAVAAAA	DLGVWMVNVH	ASGGARMMTA
3LDV A	ELHKRGFSVF	LDLKFHDIPN	TCSKAVKAAA	ELGVWMVNVH	ASGGERMMAA
3TR2 A	ELXQGYRIF	LDLKFYDIPQ	TVAGACRAVA	ELGVWXXNIH	ISGGRTXXET
3TFX A	KLQQQGYKIF	LDLKHHDIPN	TVYNGAKALA	KLGITFTTVH	ALGGSQXIKS
Target	AREALVPFG--	-KDAPLLIAV	TVLTSMEASD	LVD-LGMTLS	PADYAERLA
3LDV A	SREILEPYG--	-KERPLLIGV	TVLTSMESAD	LQG-IGILSA	PQDHVLRLA
3TR2 A	VVNALQSITL-	-KEKPLLIGV	TILTSLDGSD	LKT-LGIQEK	VPDIVCRXA
3TFX A	AKDGLIAGTPA	GHSVPKLLAV	TELTSISDDV	LRNEQNCRLP	XAEQVLSLA
Target	ALTQKCGLDG	VVCSAQEAVRF	KQVFGQEFKL	VTPGIRPQGS	EAGDQRRIM
3LDV A	TLTKNAGLDG	VVCSAQEASLL	KQHLGREFKL	VTPGIRPAGS	EQGDQRRIM
3TR2 A	TLAKSAGLDG	VVCSAQEAALL	RKQFDRNFL	VTPGIR-----	RVX
3TFX A	KXAKHSGADG	VICSPLEVKKL	HENIGDDFLY	VTPGIRP-----	A
Target	TPEQALSAGV	DYMVIGRPVTQ	SVDPAQTLKA	INASLQ-----	
3LDV A	TPAQAIASGS	DYLVIGRPITQ	AAHPEVVLEE	INSSL-----	
3TR2 A	TPRAAIQAGS	DYLVIGRPITQ	STDPLKALEA	IDKDI-----	
3TFX A	TPKXAKEWGS	SAIVVGRPITL	ASDPKAAIEA	IKKEFNENLYFQS	

Fig. 7 Multiple alignment of the target sequence with the template sequences.

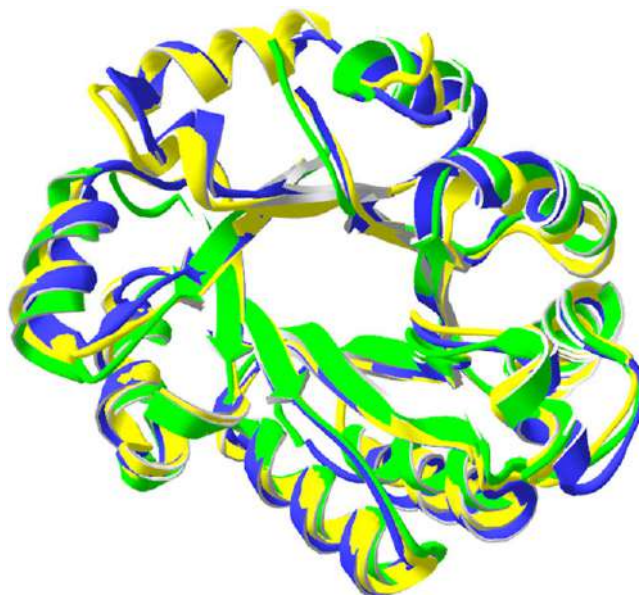


Fig. 8 Cartoon diagram of the superposition of the three template structures in Fig. 7.

Generation of loop structures

In most cases, the alignment between the query and template sequence contains gaps. Thereby the gaps can be either in the target sequences (deletions) or in the template sequence (insertions). Deletions create a hole in the target that must be closed. In the case of insertions, the backbone of the template is cut and the missing residues inserted. Both cases imply a conformational change of the backbone. For insertions in the target, no structural information can be derived from the template structures. In contrast the template structures provide a great deal of information for deletions in the target.

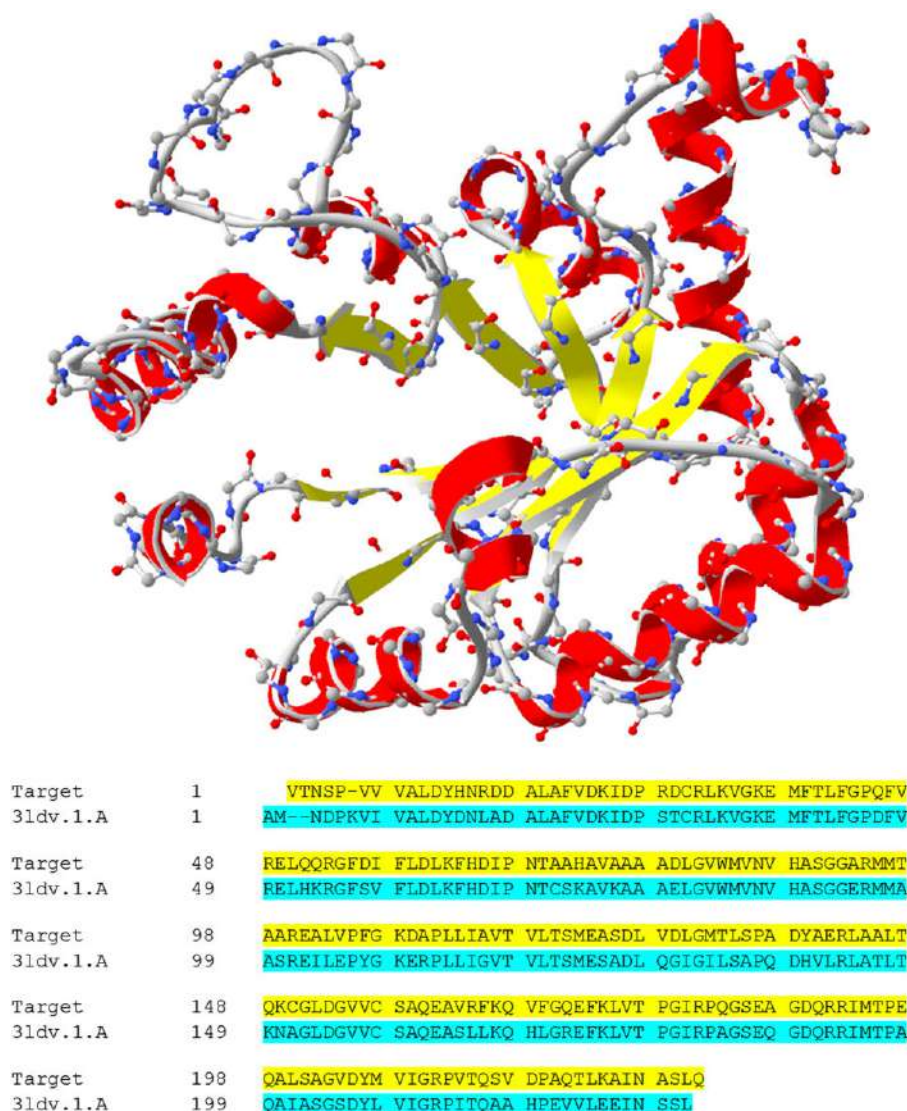


Fig. 9 In the figure, the target backbone (ball-and-stick representation) is superimposed on a cartoon representation of the 3ldv template structure (represented by a ribbon). Alpha helical regions of the template structure are shown in red, and beta strand regions are shown in yellow.

Fig. 10 illustrates the addition of a loop segment of appropriate length to the structural model (see below). It should be noted that this is only an illustrative example, and the alignment (with gap) was made with a template structure not contained in [Table 2](#).

The gaps in an alignment can be filled by searching for compatible loop fragments derived from experimental structures. Two main approaches to loop modelling can be distinguished:

1. Knowledge based approach: The PDB is searched for known loops with endpoints matching the residues between which the loop has to be inserted. Several Template Based Modelling programs such as: Swiss-Model and Modeller support this approach. If a suitable loop conformation is found, it can be simply copied and connected to the endpoint residues to fill the gap.
2. Energy based: Candidate loops are generated by constructing fragments, compatible with the neighbouring structural elements. An energy function (or force field) (Section “Molecular Force Fields”) is used to evaluate the quality of the loop. To get the optimal loop conformation the energy function is minimized, for example by Monte Carlo simulation ([Simons et al., 1999](#)) or molecular dynamics techniques, and evaluated by a scoring system ([Fiser et al., 2000](#)). The scoring system accounts for conformational energy, steric hindrance, and favourable interactions such as hydrogen bonds.

There are several reasons for different loop conformations in the template and model structures: Surface loops tend to be involved in inter-molecular interactions, resulting in significant conformational differences between the query and template structures. The exchange of small sidechains with bulky side chains pushes the loop aside. The exchange of a loop residue with a proline, or from glycine to any other residue, requires conformational changes in the loop. For short loops the methods listed above can predict a loop conformation that fits well in true structures ([Tappura, 2001](#)).

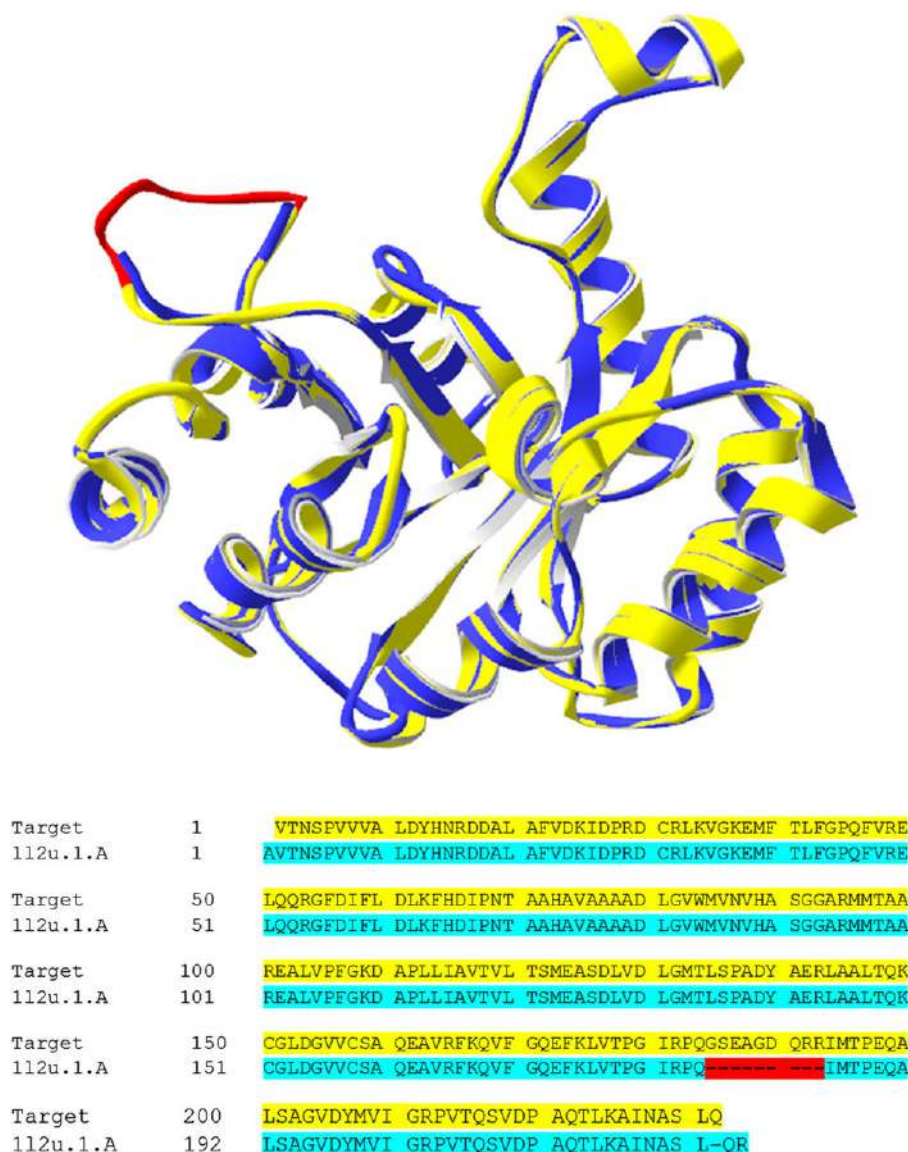


Fig. 10 Loop-modelling procedures are used to fill gaps (red) in the alignment.

Insertion of side chains

20 different amino acids are commonly found in protein sequences. Each of the 20 amino acids has a different side chain (R-group), which is responsible for its unique physicochemical properties. For instance, some of the side chains have ring like structures (aromatic), while others consist of unbranched carbon chains (aliphatic).

Side chain conformations are derived from those observed in similar structures and from steric considerations. Starting from the most conserved residues, the side chains are inserted by isosteric replacements (side chains with similar shapes and physicochemical properties) of the template side chains (Fig. 11). Amino acids that are highly conserved in structurally similar proteins often have similar side chain torsion angles. This is especially true when the amino acid residues form networks of contacts. Therefore such conserved residues can be copied in their entirety from the template structure to the model. Thereby a higher accuracy can be achieved than by copying just the backbone and predicting the side chain conformations (Sanchez and Sali, 1997a, b).

The most successful approaches to side-chain placement are knowledge based, relying on rotamer libraries tabulated based on high-resolution X-ray structures (Dunbrack and Karplus, 1994). Such libraries are built by taking high-resolution protein structures and collecting all stretches of a set of residues with a given amino acid at the centre. A preferred rotamer is identified by superimposing the corresponding backbone of an amino acid in the template on all the collected examples. Then the possible side-chain conformations are selected from the best backbone matches. A scoring function that evaluates favourable interactions, such as hydrogen bonds and disulphide bridges, and unfavourable interactions, such as steric hindrance, is used to select the most likely conformation. Various possible rotamers are successively tested and scored with energy functions (or force field, see Section

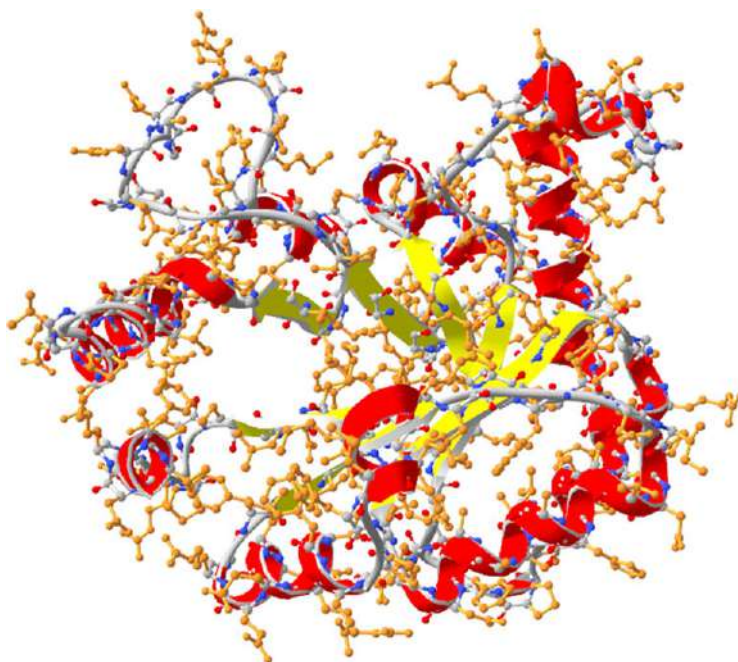


Fig. 11 The model structure after the side chains have been added to the target backbone by isosteric replacement of the side chains in the template structures. The replacement procedure relies on rules for dihedral and bond angles, and uses observed values in known protein structures to help optimize the positions of the side chains.

“Molecular Force Fields”). Because the choice of a rotamer automatically affects the rotamers of all the amino acid residues that make side chain contacts, this leads to a combinatorial explosion of the search space. Therefore, various methods have been developed to make this enormous search space tractable. For example a hydrogen bond between the side chain and backbone favors certain rotamers and thus greatly reduces the search space.

The rotamer prediction accuracy is usually high for residues in the hydrophobic protein structure core, but much lower for residues on the surface. The reasons are as follows: The flexible side chains on the surface tend to adopt multiple conformations. The energy functions used to score rotamers are well suited for evaluating the hydrophobic packing of the core, which results mainly from van der Waals interactions. But the electrostatic interactions on the surface, including hydrogen bonds with water molecules and associated entropic effects, are not handled with similar precision.

Some useful software tools for sequence alignment, loop prediction, and side chain modelling are listed in [Table 3](#).

Molecular Force Fields

In the rigid modelling procedure, described above, distortions such as: clashes between atoms; longer than normal bond lengths; unfavourable bond angles, etc., may occur, resulting in energetically unreasonable conformations. For example, steric hindrance results from overlapping of the van der Waals spheres associated with the residues, leading to strong repulsive forces. Such unrealistic distortions are removed by energy minimization techniques.

The determination of a geometrically optimal structure of a molecule, which means the structure with minimum free energy, is a further important step in homology modelling. The equilibrium free energy of a molecular structure is calculated by the use of molecular force fields. In computational physics and molecular modelling, a molecular force field is a mathematical function that describes the dependence of the potential energy of a molecule on the coordinates of its atoms. It is specified by an analytical form of the intermolecular potential energy: $U(r_1, r_2, \dots, r_N)$ and a set of input parameters. Force fields differ in their functional form as well as their fixed parameter sets ([Wang et al., 2000](#)). The parameter values are obtained either from quantum mechanical calculations (*ab initio* or semi-empirical methods), or by fitting to experimentally determined high resolution structures, determined by X-ray diffraction, nuclear magnetic resonance (NMR), infrared and Raman spectroscopy, and other methods. For the energy minimization of macromolecules, adequate molecular force fields such as AMBER, CHARMM, and GROMOS have been developed ([Christen et al., 2005](#)). In these approaches, the coordinates of all atoms of the macromolecule are treated as free variables ([Price and Brooks, 2002](#)). The basic functional form of a molecular force field includes terms for covalent bonds and terms for long-range forces (non-bonded interactions) inside the molecule:

$$U(r_1, r_2, \dots, r_N) = \underbrace{\sum_{\text{bonds}} K_r(r - r_0)^2 + \sum_{\text{bonds}} K_\theta(\theta - \theta_0)^2 + \sum_{\text{dihedrals}} V_n(1 + \cos(n\Phi))}_{\text{covalent bonds}} + \underbrace{\sum_i \sum_j \frac{q_i q_j}{r_{ij}} + \sum_i \sum_j \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right)}_{\text{non-bonded interactions}}$$

Table 3 List of useful software tools for the different operations in homology modelling

Program	Name/purpose	Web address
Sequence alignment tools		
BLAST	Basic local alignment tool	https://blast.ncbi.nlm.nih.gov/Blast.cgi
Clustal Omega	Multiple sequence alignment	http://www.ebi.ac.uk/Tools/msa/clustalo/
Vast	Vector alignment search tool structural alignment	https://structure.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml
MULTALIN	Multiple sequence alignment	http://multalin.toulouse.inra.fr/multalin/
Loop prediction and modelling tools		
Swiss-PDB Viewer	Visualization and analysis of protein structures	http://spdbv.vital-it.ch/
BRAGI	Angular trends of repeat proteins	https://bragi.helmholtz-hzi.de/index.html
RAMP a suite of programs to aid in the protein modelling		http://www.ram.org/computing/ramp/
BTPRED	The beta-turn prediction server	http://www.biochem.ucl.ac.uk/bsm/btpred/
CONGEN	CONformation GENerator	http://www.congenomics.com/congen/doc/index.html
Side chain modelling		
SCWRL4	Prediction of protein side-chain conformations	http://dunbrack.fccc.edu/scwrl4/index.php
SMD	Combinatorial amino acid sidechains optimization	http://condor.urbb.jussieu.fr/Smd.php

Table 4 Some popular molecular force fields for energy minimization

Program	Name	Web address
CHARMM	Chemistry at Harvard macromolecular mechanics	https://www.charmm.org/charmm/
AMBER	Assisted model building with energy refinement	http://ambermd.org/
GROMOS	GROningen MOlecular Simulation	http://www.gromos.net
OPLS	Optimized potentials for liquid simulations	http://zarbi.chem.yale.edu/oplsaam.html

The single terms and the symbols are explained in the following sections. Using such a force field model, macromolecules are reduced to a set of atoms held together by simple harmonic forces, Coulombic interactions, and van der Waals interactions. For practical calculations, the force field must be simple enough to be evaluated quickly, but sufficiently detailed to reproduce realistic structural properties.

Some popular molecular force fields are listed in [Table 4](#).

Covalent bond terms

For the covalent bond terms, parametrized by bond length, bond angles and dihedral angles, the potential energy is described relative to the atoms being in their equilibrium positions, for which the energy is taken to be zero.

- 1) The first term in the molecular force field describes the extension (stretching) of covalent bonds. Bond stretching is often represented by a simple harmonic function $K_r(r - r_0)^2$ that controls the length of covalent bonds ([Fig. 12](#)). This corresponds to a classical harmonic oscillator with spring constant: K_r . Realistic values for the equilibrium bond length r_0 are, for example, obtained experimentally by X-ray diffraction of small molecules. The spring constant (K_r) can be estimated from infrared or Raman spectra.

Bond lengths are determined by the electronic orbitals (s, p) of the involved atoms and the number of the shared electrons between the atoms. The harmonic potential is a poor approximation when the bond stretching exceeds displacements of more than 10% from the equilibrium value. Nevertheless, under most circumstances the harmonic approximation is reasonably good.

- 2) The second force field term describes the distortion of the bond angles ([Fig. 13](#)). Distortion of bond angles is described by the energy related to bending an angle, θ , formed by at least three atoms: A-B-C, where there is a chemical bond between A and B, and between B and C ([Fig. 14](#)). As in the case of bond stretching, the angle bending term is expanded as a Taylor series around the equilibrium bond angle, θ_0 , and terminated at the second order (harmonic approximation). The vibrational frequencies are in the near infrared spectrum, and the constant K_θ is measured by Raman spectra.
- 3) The third force field term describes the distortion of dihedral angles ([Fig. 15](#)) from their preferred values. If a molecule contains more than four atoms in a row, which is a given in macromolecules, the dihedral term must be included in the force

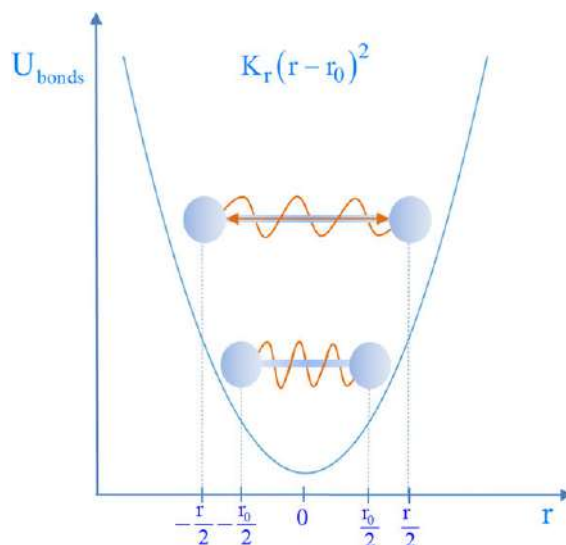


Fig. 12 Potential energy function for bond stretching.

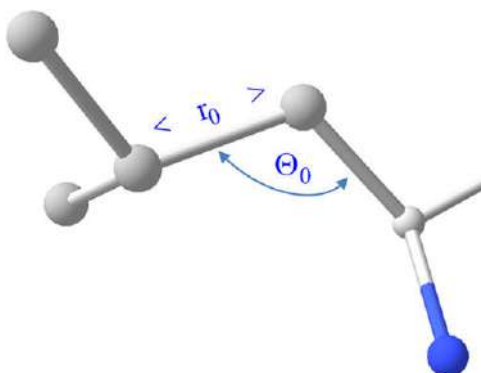


Fig. 13 Equilibrium bond length and bond angle on a part of a protein structure.

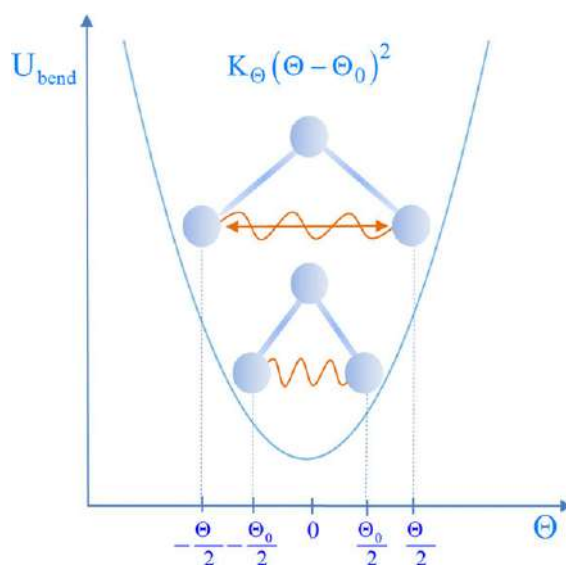


Fig. 14 Potential energy function for bond angle bending.

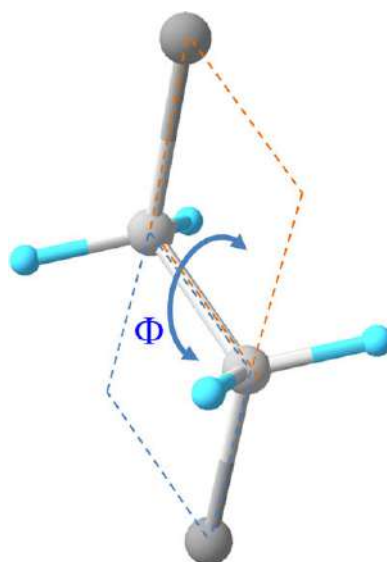


Fig. 15 Dihedral angle in 1,2-dichloroethane. This conformation corresponds to $\Phi = 180^\circ$. The energy of different conformations is shown in Fig. 16.

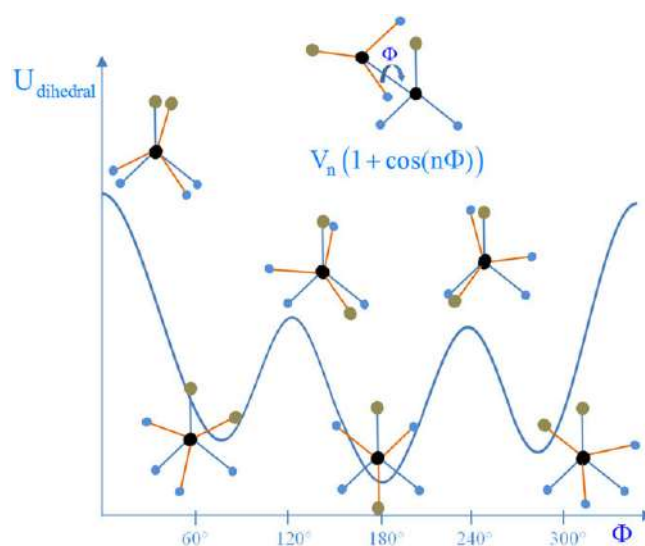


Fig. 16 Torsional potential of 1,2-dichloroethane as a function of the dihedral angle Φ . The potential energy minima are found at the gauche- ($\Phi = 60^\circ, 300^\circ$) and trans- ($\Phi = 180^\circ$) staggered conformations. The saddle points ($\Phi = 120^\circ, 240^\circ$) correspond to eclipsed (covered) conformations.

field. Dihedral angles are angles of rotation of the bonded atom pairs around the central bond. In stereochemistry, the dihedral is defined as the angle between planes through two sets of three atoms, which have two atoms in common (Fig. 16). Changes in dihedral angles often result in major conformational changes.

Fig. 16 shows the dihedral potential as a function of different conformations of 1,2-dichloromethane. The torsional motions in macromolecules determine the rigidity of the molecule. Therefore, they play an important role in determining local structures, such as reactivity centres, of a macromolecule. Additionally they play an important role in the relative stability of different molecular conformations. Bond stretching and angle bending motions are typically hundreds of times stiffer than torsional motions. Dihedral angles are mainly constrained by steric hindrance. Torsional energy is usually represented by a cosine function: $V_n(1 + \cos(n\Phi))$, where Φ is the torsional angle, and n defines the number of minima or maxima between 0 and 2π . The constant V_n determines the height of the potential energy barrier between torsional states.

All these terms make local contributions to the calculated potential energy. As seen in the figures, molecules are treated as consisting of balls (the atoms) connected by springs (the bonds). The building blocks in molecular force fields are atoms. The electrons are not treated as individual particles and no electronic details, which would require a quantum mechanical treatment,

are included. Additionally the quantum aspects of nuclear motion are neglected and the dynamics of the atoms in the molecule are treated by classical mechanics.

Non-bonded interactions

In common molecular force fields the non-bonded interactions include electrostatic forces and van der Waals forces.

- 4) The fourth term describes the electrostatic forces arising between atoms carrying a ionic charges, q_i and q_j : $\frac{q_i q_j}{r_{ij}}$. Charge-charge interactions between positive and negative ions are called salt bridges, which play a significant role in protein structure stabilization. Since substitution of basic residues for acidic residues changes the charge from positive to negative, such changes are extremely destabilizing when they occur in the interior of the protein. They tend to be more acceptable on the protein surface where the charged residues interact with polar water molecules and charged solutes.
- 5) The fifth term of the force field describes van der Waals forces: $\sum_i \sum_j (\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6})$. The movement of the electrons around the atomic nucleus creates an electric dipole moment. This dipole polarises neighbouring atoms, which results in a short-range attractive force between the non-bonded atoms (i, j).

The attractive part is described by: $-\frac{B_{ij}}{r_{ij}^6}$. Conversely, at short ranges a repulsive force between the electrons of the two atoms arises (Fig. 17). The radius at which the repulsive force begins to increase sharply is called the van der Waals radius.

The repulsive part of the energy is described by: $\frac{A_{ij}}{r_{ij}^{12}}$. Steric interactions arise when the van der Waals spheres of two non-bonded atoms are approach and interpenetrate (Fig. 18). The parameters A_{ij} and B_{ij} depend on the types of the involved atoms. This description of van der Waals forces is frequently referred to as a Lennard-Jones potential.

The equilibrium geometry of a molecule (with respect to bond lengths, angles, non-overlapping van der Waals spheres, etc.) describes the coordinates of a minimum on the potential energy surface. The problem is then reduced to determining the energy minima on this surface (Fig. 19).

The minimum of the potential energy function corresponds to the equilibrium geometry of the molecule.

An advantage of the molecular force fields method is the speed with which calculations can be performed, enabling its application to large biomolecules. With even moderate computer power, the energies of molecules with thousands of atoms can be optimized. This facilitates the molecular modelling of proteins and nucleotide acids, which is currently done by most pharmaceutical companies. There are different methods for the optimization the energy, such as: simulated annealing and conjugate gradients.

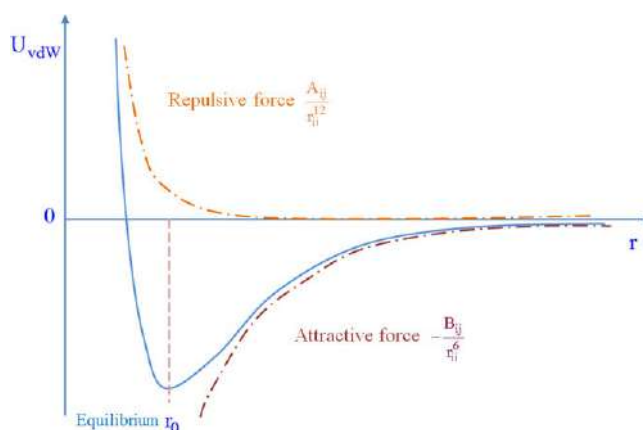


Fig. 17 Potential energy function for van der Waals interaction.

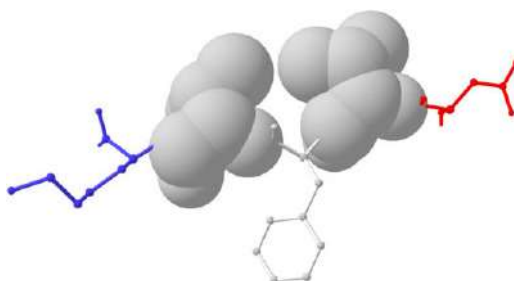


Fig. 18 Steric hindrance may result from overlapping of the van der Waals spheres associated with the residues. This leads to strong repulsive forces.

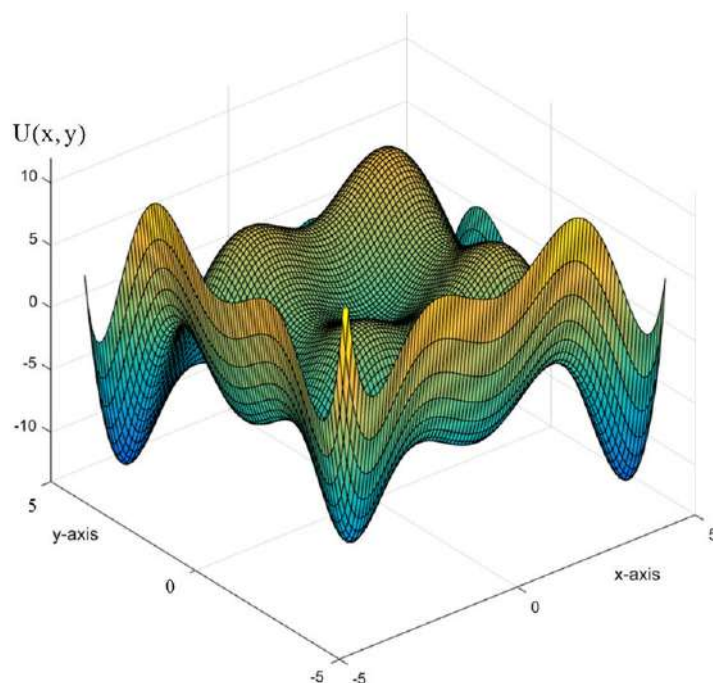


Fig. 19 The potential energy function of a molecule depends on the coordinates of its atoms.

Energy Optimization by Simulated Annealing

Simulated Annealing is an effective and general form of energy optimization. It is useful in finding the global minimum in the presence of several local minima (Agostini *et al.*, 2006; Cerny, 1985). The term annealing refers to the thermal process for obtaining low energy states of a solid in a heat bath. The annealing process consists of two steps: First the temperature of the heat bath is increased until the solid melts. Then the temperature of the heat bath is decreased slowly. If the melt is cooled slowly, large single crystals, representing the global minimum energy state, grow. Rapid cooling produces a disordered solid trapped in a local minimum energy state.

In statistical thermodynamics, the probability of a physical system being in the state with energy E_i at temperature T is given by the Boltzmann distribution:

$$p(E_i) = \frac{1}{Z} \cdot e^{-\frac{E_i}{k_B T}}$$

The parameter k_B is the Boltzmann constant that relates temperature to energy ($E \approx k_B T$). The function Z is the canonical partition function, which is the summation over all possible states, j , with energy E_j at temperature T :

$$Z = \sum_j e^{-\frac{E_j}{k_B T}}$$

Here we use the potential energy $U(r)$ of the molecular force field. Thereby, the parameter r is the set of all atomic coordinates, bond angles and dihedral angles. The probability of observing a particular molecular conformation is given by the above Boltzmann distribution. The probability of molecular transition from state r_k to state r_l is determined by the Boltzmann distribution of the energy difference, $\Delta E = U(r_k) - U(r_l)$, between the two states:

$$P = e^{-\frac{\Delta E}{k_B T}}$$

The different molecular configurations are given by different values of bond lengths, angles, dihedral angles, and non-bonded interactions in the protein. The states are modified by random modifications of these parameters. (*Mostly computational algorithms for pseudo-random number generators are used*). For example, consider random changes to a dihedral angle (Fig. 20). Then examine the energy associated with the resulting atom positions to decide whether or not to accept each considered move. T is a control parameter called *computational temperature*, which controls the magnitude of the random perturbations of the potential energy function. At high temperatures, large modifications of the molecular configuration, resulting in large changes in energy, are preferred. (At high T the Boltzmann distribution exhibits a uniform preference for all states, regardless of their energy). As T is lowered, the system responds mainly to small changes in the potential energy, and performs a fine search in the neighborhood of the already determined minimum to find a better minimum. When T approaches zero, only the states with lowest energies have nonzero probabilities of occurrence. At the initial high temperature, large conformational changes of the molecule are allowed. But as the temperature is decreased, the conformation of the molecule becomes trapped in an energy minimum (Fig. 21). The probabilities of uphill moves ($\Delta E > 0$) in the energy potential function

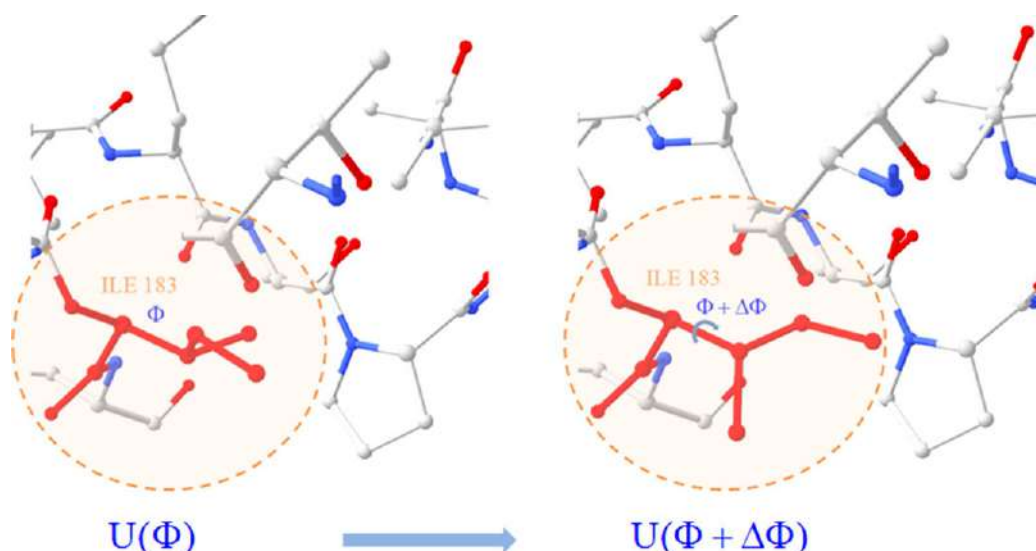


Fig. 20 The dihedral angle of the side chain of ILE 183 is changed randomly. Thereby results a change of the energy of the protein.

(of the molecular force field) are more likely at high temperature than at low temperature. Simulated annealing allows uphill moves in the energy potential function in a controlled fashion: It attempts to avoid a greedy movement to the next local minima by occasionally accepting a worse solution. This procedure is called the Metropolis criteria. Simulated annealing is a variant of the Metropolis algorithm, where the temperature is changing from high to low (Kirkpatrick *et al.*, 1983). The probability of accepting a conformational change that increases the energy decreases exponentially with the difference in the energies, ΔE , in the respective conformations.

The simulated annealing is basically composed of two stochastic processes: one process for the generation of a conformational change (for example a dihedral angle is modified) and the other for the acceptance of or rejection of the new conformation. The *computational temperature* is responsible for the correlation between the generated and the initial conformation.

In a typical simulated annealing optimization, T starts high and is gradually decreased according to the following algorithm:

```

Initialize the molecular configuration:  $r = r_{\text{init}}$ ;
Assign a large value to temperature:  $T = T_{\text{max}}$ ;
Repeat:
  Repeat:
    The configuration  $U(r)$  is modified by random perturbation:  $r = r + \Delta r$ ;
    The resulting energy difference is evaluated:  $\Delta E = U(r + \Delta r) - U(r)$ ;
    If  $\Delta E < 0$  then: keep the new configuration;
    otherwise: accept the new configuration with a probability:  $P = e^{-\frac{\Delta E}{k_B T}}$ ;
  until the number of accepted transitions is below a predefined threshold level.
Set:  $T = T - \Delta T$ ;
until  $T$  is small enough.
End
  
```

As the temperature decreases, uphill moves become more and more unlikely, until there are only downhill moves, and the molecular configuration converges (in the ideal case) to the equilibrium conformation. The size (ΔT) of the cooling steps for T is critical to the efficiency of simulated annealing. If T is reduced too rapidly, a premature convergence to a local potential minimum may occur. In contrast, if the step size is too small, the algorithm converges very slowly to a global minimum.

Model Verification

The dihedral angle, ω , of the protein backbone is restricted due to the planarity of the amide bond (C and N) and the hybridization of the involved atomic orbitals (Fig. 22). From this results a resonance structure with partial double binding and a permanent dipole moment (with negatively charged oxygen). Therefore rotation around ω requires a large amount of energy (80 kJ/mol). The values of the dihedral angles ϕ and ψ are restricted by the steric hindrance between the atoms of neighbouring peptide bonds and side chain atoms.

The Ramachandran plot shows the statistical distribution of the combinations of the backbone dihedral angles ϕ and ψ . In theory, the allowed regions of the Ramachandran plot show which values of the Phi/Psi angles are possible for an amino acid, X, in

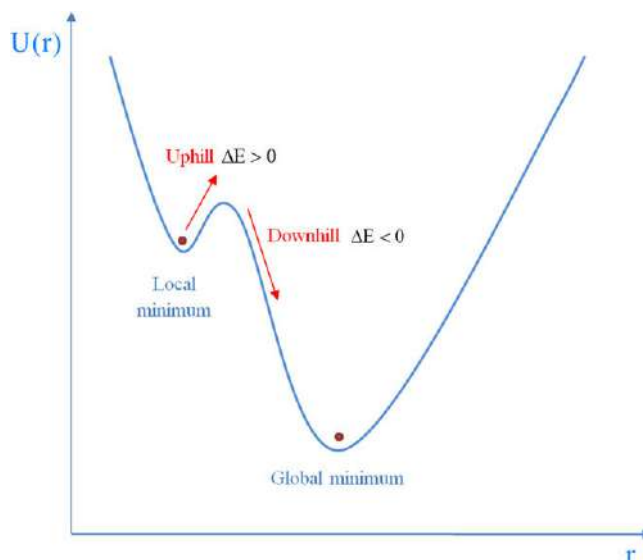


Fig. 21 The potential energy function with local and global minima depending on the molecule conformation.

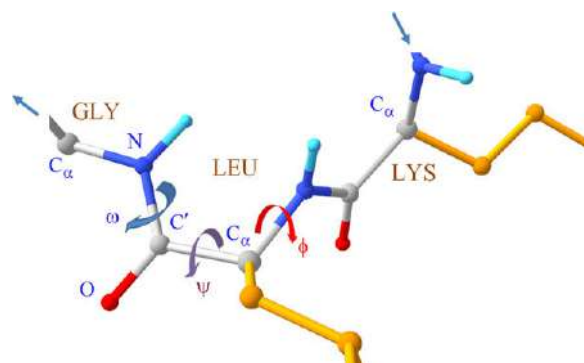


Fig. 22 Definition of the protein backbone dihedral angles ω , ψ and ϕ .

a ala-X-ala tripeptide (Ramachandran *et al.*, 1963). In practice, the distribution of the Phi/Psi values observed in a protein structure can be used for structure validation (Ramakrishnan *et al.*, 2007). The Ramachandran plot visualizes energetically allowed and forbidden regions for the dihedral angles. For poor quality homology models, many dihedral angles are found in the forbidden regions of the Ramachandran plot. Such deviations usually indicate problems with the structure.

Fig. 23 shows the Ramachandran plot of the homology model of the amino acid sequence of orotidine 5'-monophosphate decarboxylase. The plot is a visualization produced by Swiss-PDB viewer, and colors were added after the plot was generated. The dihedral angles of amino acid residues appear as crosses in the plot. The blue and red regions represent the favoured and allowed regions. The blue regions correspond to conformations where there are no steric clashes in the model tripeptide. These favoured regions include the dihedral angles typical of the alpha-helical and beta-sheet conformations. The orange areas correspond to conformations where atoms in the protein come closer than the sum of their van der Waals radii. These regions are sterically forbidden for all amino acids with side chains (the exception is glycine which has no side chain). In Swiss-PDB viewer, the Ramachandran plot can be used to interactively modify the Phi/Psi angles of an amino acid

A number of freely available software tools can be used to analyze the geometric properties of homology modelling results. These model assessment and validation tools are generally of two types: Programs of the first category (PROCHECK and WHATCHECK) perform symmetry and geometry checks (including bond lengths, bond angles, and torsion angles) and consider the influence of solvation. Those in the second category (VERIFY3D and ProSA) check the quality of the sequence to structure match and assign a score for each residue (Table 5).

ANOLEA (Atomic Non-Local Environment Assessment) is a server that performs energy calculations on a protein chain, by use of a distance-dependent knowledge-based mean force potential, derived from a database (Melo *et al.*, 1997). Thereby the Non-Local Environment (NLE: defined as all the heavy atoms, within an Euclidean distance of 7 Å and that belong to amino acids that are more distant than 11 residues) of each heavy atom in the molecule is evaluated.

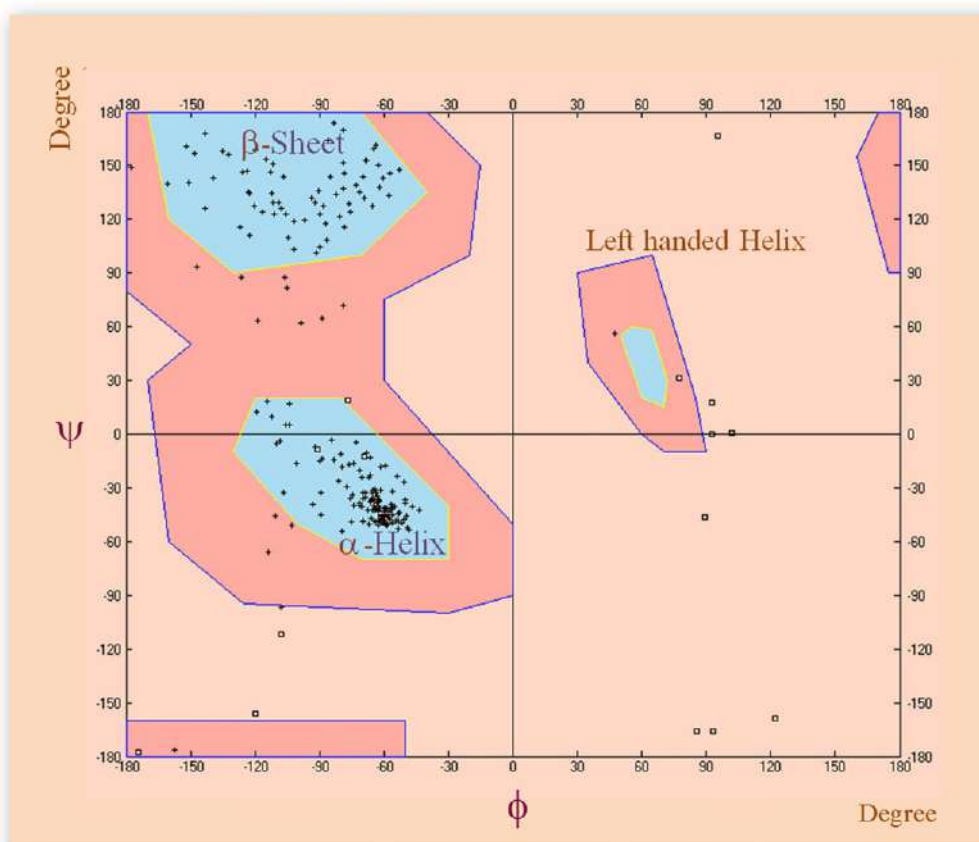


Fig. 23 Distribution of the amino acid Phi/Psi angles in the orotidine 5'-monophosphate decarboxylase.

Table 5 Software packages that can be used to evaluate the quality of the geometry and sequence to structure fitness of a homology-based model

Name	Method	Web address
PROCHECK	Checks the stereo-chemical quality	http://www.ebi.ac.uk/thornton-srv/software/PROCHECK/
WHATCHECK		http://swift.cmbi.ru.nl/gv/whatcheck/
ProSA	Checks the fitness of sequence to structure	https://prosa.services.came.sbg.ac.at/prosa.php
VERIFY3D		http://services.mbi.ucla.edu/Verify_3D/
ANOLEA	Energy calculations	http://melolab.org/anolea/

Fig. 24 shows the final homology model of the amino acid sequence of orotidine 5'-monophosphate decarboxylase. The target structure (represented by the ball-and-stick diagram) is superimposed on that of a template structure (represented by the ribbon diagram).

In conclusion it should be noted that a protein model is a tool that helps to interpret biochemical data. Models can be inaccurate or even completely wrong.

Applications of Homology Modelling in Human Biology and Medicine

Today homology modelling is one of the most common techniques used to build accurate structural models of proteins, and is used for rationalizing experimental observations. It is widely used in structure based drug design, and the study of inter-individual differences in drug metabolism (Cavasotto and Abagyan, 2004). Further applications include: (a) designing site-directed mutants to test hypotheses about protein function; (b) identification and structural analysis of small molecule binding sites for ligand design and search; (c) design and improvement of inhibitors for an enzyme based on its predicted substrate or product binding sites; (d) prediction and analysis of epitopes; (e) modelling of substrate specificity; (f) protein-protein docking simulations (see Table 6).

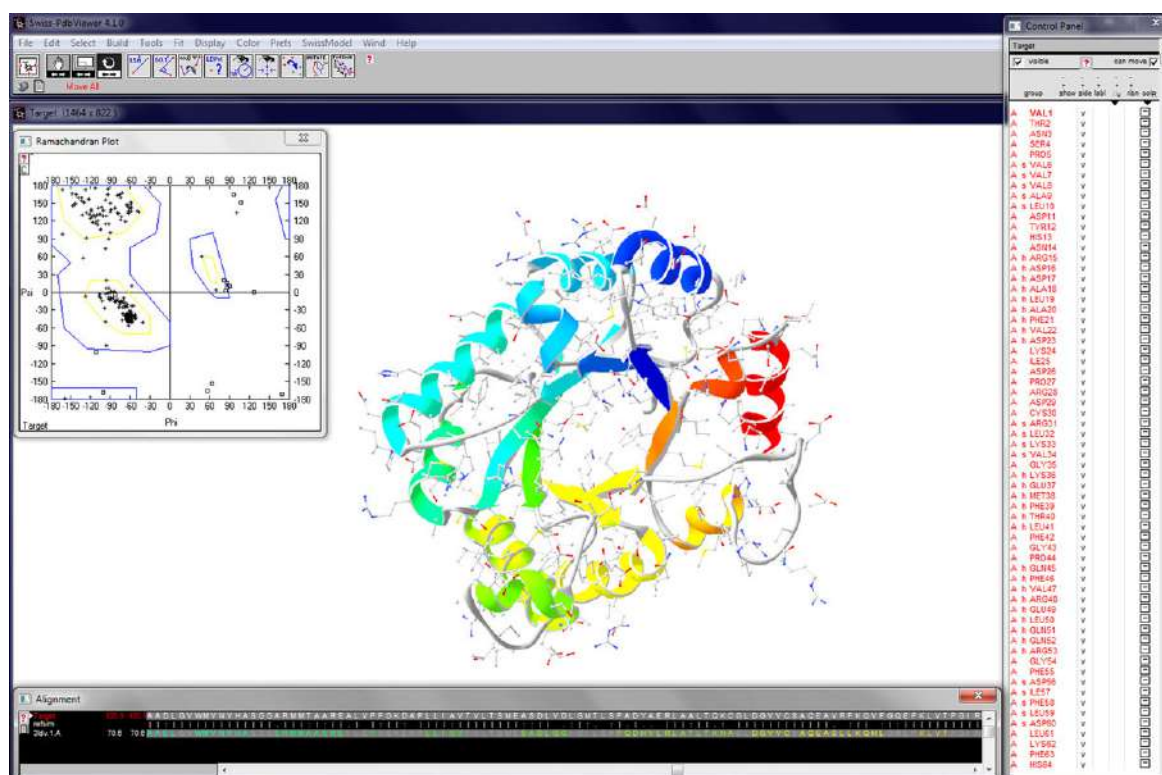


Fig. 24 Snapshot of the Swiss-PDB viewer showing the result of the homology modelling procedure.

Table 6 A selected list of homology modelling projects in medicine

Homology modelling projects	Authors
<i>P. falciparum</i> DHFR enzyme inhibitors	Adane and Bharatam (2008)
Beta2-adrenergic receptor	Costanzi (2008)
<i>Apis mellifera</i> nicotinic acetylcholine receptor	Rocher <i>et al.</i> (2008)
Histone deacetylases (HDACs)	Wang <i>et al.</i> (2005)
Human β 2-adrenergic G protein coupled receptor	Cherezov <i>et al.</i> (2007)
Serotonin 5-HT _M receptor	Nowak <i>et al.</i> (2006)
Anti-CD34 Monoclonal Antibody	Hou <i>et al.</i> (2008)
Human GAD65	Capitani <i>et al.</i> (2005)
Acetyl CoA carboxylase	Zhu <i>et al.</i> (2006)
Human serum carnosinase	Vistoli <i>et al.</i> (2006)
Cannabinoid receptor-2	Diaz <i>et al.</i> (2009)
C-terminal domain of human Hsp90	Sgobba <i>et al.</i> (2008)
Cytochrome sterol 14 alpha demethylase	Zhang <i>et al.</i> (2010)
Human adenosine A2A receptor	Michielan <i>et al.</i> (2008)
Adenosine A2a receptor complex	Katritch <i>et al.</i> (2010)
Melanin concentrating hormone receptor	Cavasotto <i>et al.</i> (2008)
Carbonic Anhydrase IX	Tuccinardi <i>et al.</i> (2007)
Brain lipid binding protein	Xu <i>et al.</i> (1996)
Human dopamine receptors	Wang <i>et al.</i> (2010)
Dopamine D ₃ receptor	Cui <i>et al.</i> (2010)
Alpha-1-adrenoreceptors	Li <i>et al.</i> (2008)
Human P-glycoprotein	Domicieva and Biggin (2015)
G Protein-coupled Estrogen Receptor	Bruno <i>et al.</i> (2016)
Trace amine-associated receptor 2	Cichero and Tonelli (2017)
Histamine Receptors	Strasser and Wittmann (2017)
Human tyrosinases	Hassan <i>et al.</i> (2017)
TSP0 protein	Bhargavi <i>et al.</i> (2017)
Cajanus cajan Protease Inhibitor.	Shamsi <i>et al.</i> (2017)
African horse sickness virus VP7 trimer	Bekker <i>et al.</i> (2017)
Cytochrome bc1 complex binding	Sodero <i>et al.</i> (2017)
Human P2X7 receptor	Caseley <i>et al.</i> (2017)

The structures of medically relevant proteins involved in malaria, nicotinic acetylcholine receptor, bacteria inhabiting the gastrointestinal tract, anticancer drugs, hematopoietic stem/progenitor cell selection, autoimmunity and many more, have been determined by homology modelling (Table 6).

See also: Algorithms for Strings and Sequences: Searching Motifs. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Biomolecular Structures: Prediction, Identification and Analyses. Computational Protein Engineering Approaches for Effective Design of New Molecules. DNA Barcoding: Bioinformatics Workflows for Beginners. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Investigating Metabolic Pathways and Networks. Metagenomic Analysis and its Applications. Protein Structural Bioinformatics: An Overview. Protein Three-Dimensional Structure Prediction. Proteomics Mass Spectrometry Data Analysis Tools. Secondary Structure Prediction. Small Molecule Drug Design. Structural Genomics

References

- Adane, L., Bharatam, P.V., 2008. Modelling and informatics in the analysis of *P. falciparum* DHFR enzyme inhibitors. *Curr. Med. Chem.* 15 (16), 155215–155269.
- Agostini, F.P., Soares-Pinto Dde, O., Moret, M.A., Osthoff, C., Pascutti, P.G., 2006. Generalized simulated annealing applied to protein folding studies. *J. Comput. Chem.* 27 (11), 1142–1155.
- Altschul, S.F., Madden, T.L., Schäffer, A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25 (1), 3389–3402.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.
- Andreeva, A., Howorth, D., Chandonia, J.M., *et al.*, 2008. Data growth and its impact on the SCOP database: New developments. *Nucleic Acids Res.* 36, 419–425.
- Baxeavanis, A.D., 2006. Searching the NCBI databases using Entrez. In: *Current Protocols in Human Genetics Chapter 6: Unit 6.10*.
- Bekker, S., Burger, P., van Staden, V., 2017. Analysis of the three-dimensional structure of the African horse sickness virus VP7 trimer by homology modelling. *Virus Res.* 232, 80–95.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Res.* 28, 235–242.
- Bhargavi, M., Sivan, S.K., Potlappally, S.R., 2017. Identification of novel anti cancer agents by applying insilico methods for inhibition of TSPO protein. *Comput. Biol. Chem.* 68, 43–55.
- Bruno, A., Aiello, F., Costantino, G., Radi, M., 2016. Homology modelling, validation and dynamics of the G protein-coupled estrogen receptor 1 (GPER-1). *Mol. Inform.* 35 (8–9), 333–339.
- Capitani, G., De Biase, D., Gut, H., Ahmed, A., Grütter, M.G., 2005. Structural model of human GAD65: Prediction and interpretation of biochemical and immunogenic features. *Proteins: Struct. Funct. Bioinform.* 59, 7–14.
- Caseley, E.A., Muench, S.P., Jiang, L.H., 2017. Conformational changes during human P2X7 receptor activation examined by structural modelling and cysteine-based cross-linking studies. *Purinergic Signal.* 13 (1), 135–141.
- Cavasotto, C.N., Abagyan, R.A., 2004. Protein flexibility in ligand docking and virtual screening to protein kinases. *J. Mol. Biol.* 337, 209–225.
- Cavasotto, C.N., Orry, A.J., Murgolo, N.J., *et al.*, 2008. Discovery of novel chemotypes to a G-protein-coupled receptor through ligand-steered homology modelling and structure-based virtual screening. *J. Med. Chem.* 51, 581–588.
- Cerny, V., 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Optim. Theory Appl.* 45, 41–51.
- Chang, P.L., 2005. Clinical bioinformatics. *Chang Gung Med. J.* 28 (4), 201–211.
- Cherezov, V., Rosenbaum, D.M., Hanson, *et al.*, 2007. High-resolution crystal structure of an engineered human β 2-adrenergic G protein coupled receptor. *Science* 318, 1258–1265.
- Chou, K.C., 2006. Structural bioinformatics and its impact to biomedical science and drug discovery. In: Atta-ur-Rahman, A., Reitz, B. (Eds.), *Frontiers in Medicinal Chemistry* 3. , pp. 455–502.
- Christen, M., Hünenberger, P.H., Bakowies, D., *et al.*, 2005. The GROMOS software for biomolecular simulation: GROMOS05. *J. Comput. Chem.* 26 (16), 1719–1751.
- Cichero, E., Tonelli, M., 2017. New insights into the structure of the trace amine-associated receptor 2: Homology modelling studies exploring the binding mode of 3-iodothyronamine. *Chem. Biol. Drug Des.* 89 (5), 790–796.
- Corpet, F., 1988. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.* 16 (22), 10881–10890.
- Costanzi, S., 2008. On the applicability of GPCR homology models to computer aided drug discovery: A comparison between in silico and crystal structures of the beta2-adrenergic receptor. *J. Med. Chem.* 51, 2907–2914.
- Cui, W., Wei, Z., Chen, Q., *et al.*, 2010. Structure-based design of peptides against G3BP with cytotoxicity on tumor cells. *J. Chem. Inf. Model.* 50, 380–387.
- Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model of evolutionary change in proteins. In: Dayhoff, M.O. (Ed.), *Atlas of Protein Sequence and Structure*, vol. 5, Suppl. 3. National Biomedical Research Foundation, pp. 345–352.
- Diaz, P., Phatak, S.S., Xu, J., *et al.*, 2009. 2,3-Dihydro-1-benzofuran derivatives as a novel series of potent selective cannabinoid receptor 2 agonists: Design, synthesis, and binding mode prediction through ligand-steered modelling. *Chem. Med. Chem.* 4, 1615–1629.
- Domicevica, L., Biggin, P.C., 2015. Homology modelling of human P-glycoprotein. *Biochem. Soc. Trans.* 5, 952–958.
- Dunbrack Jr., R.L., Karplus, M., 1994. Conformational analysis of the backbone dependent rotamer preferences of protein side chains. *Nat. Struct. Biol.* 5, 334–340.
- Eswar, N., Eramian, D., Webb, B., Shen, M.Y., Sali, A., 2008. Protein structure modelling with MODELLER. *Methods Mol. Biol.* 426, 145–159.
- Fiser, A., Kihl, G., Do, R., Sali, A., 2000. Modelling of loops in protein structures. *Protein Sci.* 9, 1753–1773.
- Franklin, M.C., Cheung, J., Rudolph, M.J., *et al.*, 2015. Structural genomics for drug design against the pathogen *Coxiella burnetii*. *Proteins* 83, 2124–2136.
- Gasteiger, E., Gattiker, A., Hoogland, C., *et al.*, 2003. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res.* 31, 3784–3788.
- Gibas, C., Jambeck, P., 2001. *Developing Bioinformatics Computer Skills: An Introduction to Software Tools for Biological Applications*. O'Reilly Media.
- Guex, N., Peitsch, M.C., 1997. SWISS-MODEL and the Swiss-Pdb Viewer: An environment for comparative protein modelling. *Electrophoresis* 18, 2714–2723.
- Harris, P., Poulsen, J.C., Jensen, K.F., Larsen, S., 2002. Substrate binding induces domain movements in orotidine 5'-monophosphate decarboxylase. *J. Mol. Biol.* 18, 1019–1029.
- Hassan, M., Abbas, Q., Raza, H., Moustafa, A.A., Seo, S.Y., 2017. Computational analysis of histidine mutations on the structural stability of human tyrosinases leading to albinism insurgence. *Mol. Biosyst.* 13 (8), 1534–1544.
- Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. *PNAS* 89 (22), 10915–10919.

- Hou, S., Li, B., Wang, L., *et al.*, 2008. Humanization of an anti-CD34 monoclonal antibody by complementarity-determining region grafting based on computer-assisted molecular modelling. *J. Biochem.* 144 (1), 115–120.
- Hubbard, T., Murzin, A., Brenner, S., Chothia, C., 1997. SCOP: A structural classification of proteins database. *Nucleic Acids Res.* 25 (1), 236–239.
- Johnson, M.S., Srinivasan, N., Sowdhamini, R., Blundell, T.L., 1994. Knowledge based protein modelling. *CRC Crit. Rev. Biochem. Mol. Biol.* 29, 1–68.
- Kaczanowski, S., Zielenkiewicz, P., 2010. Why similar protein sequences encode similar three-dimensional structures? *Theor. Chem. Acc.* 125, 643–650.
- Karlin, S., Altschul, S.F., 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA* 87, 2264–2268.
- Katritch, V., Rueda, M., Lam, P.C., Yeager, M., Abagyan, R., 2010. GPCR 3D homology models for ligand screening: Lessons learned from blind predictions of adenosine A2a receptor complex. *Proteins* 78, 197–211.
- Kendrew, J.C., Dickerson, R.E., Strandberg, B.E., *et al.*, 1960. Structure of myoglobin. A three-dimensional Fourier synthesis at 2 angstrom resolution. *Nature* 185, 422–427.
- Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Knudsen, M., Wiuf, C., 2010. The CATH database. *Hum. Genom.* 4 (3), 207–212.
- Kryshtafovych, A., Monastyrsky, B., Fidelis, K., *et al.*, 2017. Evaluation of the template-based modeling in CASP12. *Proteins*. 1–14. doi:10.1002/prot.25425. [Epub ahead of print].
- Lesk, A.M., 2001. Introduction to Protein Architecture. Oxford: Oxford University Press.
- Lesk, A.M., 2002. Introduction to Bioinformatics. Oxford: Oxford University Press.
- Li, M., Fang, H., Du, L., Xia, L., Wang, B., 2008. Computational studies of the binding site of alpha1A adrenoceptor antagonists. *J. Mol. Model.* 14, 957–966.
- Melo, F., Devos, D., Depiereux, E., Feytmans, E., 1997. ANOLEA: A www server to assess protein structures. *Intell. Syst. Mol. Biol.* 97, 110–113.
- Michielan, L., Bacilieri, M., Schiesaro, A., *et al.*, 2008. Linear and nonlinear 3D-QSAR approaches in tandem with ligand-based homology modelling as a computational strategy to depict the pyrazolo-triazolo-pyrimidine antagonists binding site of the human adenosine A2A receptor. *J. Chem. Inf. Model.* 48, 350–363.
- Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 24, 536–540.
- Needleman, Wunsch, 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453.
- Notredame, C., 2007. Recent evolutions of multiple sequence alignment algorithms. *PLOS Comput. Biol.* 3 (8), 1405–1408.
- Nowak, M., Koaczowski, M., Pawowski, M., Bojarski, A.J., 2006. Homology modelling of the serotonin 5-HT1A receptor using automated docking of bioactive compounds with defined geometry. *J. Med. Chem.* 49, 205–214.
- Peitsch, M.C., 1997. Large scale protein modelling and model repository. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 5, 234–236.
- Price, D.J., Brooks, C.L., 2002. Modern protein force fields behave comparably in molecular dynamics simulations. *J. Comput. Chem.* 23 (11), 1045–1057.
- Ramachandran, G.N., Ramakrishnan, C., Sasisekharan, V., 1963. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* 7, 95–99.
- Ramakrishnan, C., Lakshmi, B., Kurien, A., Devipriya, D., Srinivasan, N., 2007. Structural compromise of disallowed conformations in peptide and protein structures. *Protein Pept. Lett.* 14 (7), 672–682.
- Rocher, A., Marchand-Geneste, N., 2008. Homology modelling of the Apis mellifera nicotinic acetylcholine receptor (nAChR) and docking of imidacloprid and fipronil insecticides and their metabolites. *SAR QSAR Environ. Res.* 19 (3–4), 245–261.
- Rose, P.W., Plić, A., Altunkaya, A., *et al.*, 2017. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Res.* 45 (D1), 271–281.
- Roy, A., Zhang, Z., 2012. Protein structure prediction. In: eLS. Chichester: John Wiley & Sons, Ltd. Doi:10.1002/9780470015902.a0003031.pub2.
- Sali, A., Blundell, T.L., 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* 234, 779–815.
- Sanchez, R., Sali, A., 1997a. Advances in comparative protein-structure modelling. *Curr. Opin. Struct. Biol.* 7, 206–214.
- Sanchez, R., Sali, A., 1997b. Evaluation of comparative protein structure modeling by MODELLER-3. *Proteins. Suppl.* 1, 50–58.
- Sanchez, R., Sali, A., 2000. Comparative protein structure modelling. Introduction and practical examples with modeller. *Methods Mol. Biol.* 143, 97–129.
- Schwede, T., Diemand, A., Guex, N., Peitsch, M.C., 2000. Protein structure computing in the genomic era. *Res. Microbiol.* 151, 107.
- Schwede, T., Kopp, J., Guex, N., Peitsch, M.C., 2003. SWISS-MODEL: An automated protein homology-modelling server. *Nucleic Acids Res.* 31 (13), 3381–3385.
- Sgobba, M., Degliesposti, G., Ferrari, A.M., Rastelli, G., 2008. Structural models and binding site prediction of the C-terminal domain of human Hsp90: A new target for anticancer drugs. *Chem. Biol. Drug Des.* 71 (5), 420–433.
- Shamsi, T.N., Parveen, R., Ahamad, S., Fatima, S., 2017. Structural and biophysical characterization of Cajanus cajan protease inhibitor. *J. Nat. Sci. Biol. Med.* 8 (2), 186–192.
- Simons, K.T., Bonneau, R., Ruczinski, I., Baker, D., 1999. Ab initio structure prediction of CASP III targets using ROSETTA. *Proteins. Suppl.* 3, 171–176.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *JMB* 147, 195–197.
- Sodero, A.C., Abraham-Vieira, B., Torres, P.H., *et al.*, 2017. Insights into cytochrome bc1 complex binding mode of antimalarial 2-hydroxy-1,4-naphthoquinones through molecular modelling. *Mem. Inst. Oswaldo Cruz.* 112 (4), 299–308.
- Strasser, A., Wittmann, H.J., 2017. Molecular modelling approaches for the analysis of histamine receptors and their interaction with ligands. *Handb. Exp. Pharmacol.* 241, 31–61.
- Sussman, J.L., Abola, E.E., Lin, D., *et al.*, 1990. The protein data bank. Bridging the gap between the sequence and 3D structure world. *Genetica* 106 (1–2), 149–158.
- Sussman, J.L., Lin, D., Jiang, J., *et al.*, 1998. Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. *Acta Crystallogr. D* 54 (Pt 6 Pt 1), 1078–1084.
- Tappura, K., 2001. Influence of rotational energy barriers to the conformational search of protein loops in molecular dynamics and ranking the conformations. *Proteins* 44, 167–179.
- Tuccinardi, T., Ortore, G., Rossello, A., Supuran, C.T., Martinelli, A., 2007. Homology modelling and receptor-based 3D-QSAR study of carbonic anhydrase IX. *J. Chem. Inf. Model.* 47, 2253–2262.
- Vistoli, G., Pedretti, A., Cattaneo, M., Aldini, G., Testa, B., 2006. Homology modelling of human serum carnosinase, a potential medicinal target, and MD simulations of its allosteric activation by citrate. *J. Med. Chem.* 49, 3269–3277.
- Wallace, I.M., Blackshields, G., Higgins, D.G., 2005. Multiple sequence alignments. *Curr. Opin. Struct. Biol.* 15 (3), 261–266.
- Wang, D., Helquist, P., Wiest, N.L., Wiest, O., 2005. Toward selective histone deacetylase inhibitor design: Homology modelling, docking studies, and molecular dynamics simulations of human class I histone deacetylases. *J. Med. Chem.* 48, 6936–6947.
- Wang, J., Cieplak, P., Kollman, P.A., 2000. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J. Comput. Chem.* 21, 1049–1074.
- Wang, Q., Mach, R.H., Luedtke, R.R., Reichert, D.E., 2010. Subtype selectivity of dopamine receptor ligands: Insights from structure and ligand-based methods. *J. Chem. Inf. Model.* 50, 1970–1985.
- Webb, B., Sali, A., 2014. Comparative protein structure modelling using modeller. In: Current Protocols in Bioinformatics. John Wiley & Sons, Inc. 5.6.1–5.6.32.
- Westbrook, J.D., Fitzgerald, P.M., 2003. The PDB format, mmCIF, and other data formats. *Methods Biochem. Anal.* 44, 161–179.
- Wiltgen, M., 2009. Structural bioinformatics: From the sequence to structure and function. *Curr. Bioinform.* 4, 54–87.
- Xu, D., Xu, Y., Uberbacher, E.C., 2000. Computational tools for protein modelling. *Curr. Protein Pept. Sci.* 1, 1–21.
- Xu, L.Z., Sanchez, R., Sali, A., Heintz, N., 1996. Ligand specificity of brain lipid binding protein. *J. Biol. Chem.* 271, 24711–24719.
- Zhang, Q., Li, D., Wei, P., *et al.*, 2010. Structure-based rational screening of novel hit compounds with structural diversity for cytochrome P450 sterol 14 α -demethylase from *Penicillium digitatum*. *J. Chem. Inf. Model.* 50, 317–325.
- Zhu, X., Zhang, L., Chen, Q., Wan, J., Yang, G., 2006. Interactions of aryloxyphenoxypionic acids with sensitive and resistant acetyl-coenzyme A carboxylase by homology modelling and molecular dynamic simulations. *J. Chem. Inf. Model.* 46, 1819–1826.

Further Reading

- Breda, A., Valadares, N.F., Norberto de Souza, O., *et al.*, 2006. Protein structure, modelling and applications. In: Gruber, A., Durham, A.M., Huynh, C., *et al.* (Eds.), *Bioinformatics in Tropical Disease Research: A Practical and Case-Study Approach* [Internet]. Bethesda (MD): National Center for Biotechnology Information (US). (Chapter A06) <https://www.ncbi.nlm.nih.gov/books/NBK6824/>.
- Forbes J. Burkowski, 2008. Structural bioinformatics: An algorithmic approach. In: *Mathematical and Computational Biology*. Chapman & Hall/CRC ISBN 9781584886839 – CAT# C6838.
- González, M.A., 2011. Force fields and molecular dynamics simulations. *Collection Societe Francaise Neutronic SFN 12*, pp. 169–200. Available at: <https://doi.org/10.1051/sfn/201112009>.
- Haas, J., Roth, S., Arnold, K., *et al.*, 2013. The protein model portal – A comprehensive resource for protein structure and model information Database (Oxford), 2013: bat031. Published online 2013 Apr 19. doi: 10.1093/database/bat031.
- Krieger, E., Nabuurs, S.B., Gert Vriend, G., 2003. Homology modelling. In: Bourne, Philip E., Weissig, H. (Eds.), *Structural Bioinformatics*. Wiley-Liss, Inc.. <http://www.cmbi.ru.nl/edu/bioinf4/articles/homologymodelling.pdf>.
- Roy, A., Zhang, Y., 2012. Protein structure prediction. In: eLS. Chichester: John Wiley & Sons, Ltd. Available at: <http://dx.doi.org/10.1002/9780470015902.a0003031.pub2>.
- Vanommeslaeghe, K., Guvench, O., MacKerell Jr., A.D., 2014. Molecular mechanics. *Curr. Pharm. Des.* 20 (20), 3281–3292. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4026342/>.
- Vyas, V.K., Ukawala, R.D., Ghate, M., Chintia, C., 2012. Homology modelling a fast tool for drug discovery: Current perspectives *Indian. J. Pharm. Sci.* 74 (1), 1–17. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3507339/>.
- Wiltgen, M., 2009. Structural bioinformatics: From the sequence to structure and function. *Curr. Bioinform.* 4, 54–87.
- Wiltgen, M., Tilz, G., 2009. Homology modelling: A review about the method on hand of the diabetic antigen GAD 65 structure prediction. *Wien. Med. Wochenschr.* 159 (5–6), 112–125.

Relevant Websites

- <https://www.ncbi.nlm.nih.gov/blast>
Basic Local Alignment Search Tool.
- www.expasy.org
ExPASy.
- <https://salilab.org/modeller/>
MODELLER.
- <https://www.ncbi.nlm.nih.gov/>
National Center for Biotechnology Information.
- <http://www.rcsb.org/pdb>
RCSB PDB.
- <http://swissmodel.expasy.org/>
SWISS-MODEL.
- <http://spdbv.vital-it.ch>
Swiss PDB Viewer.
- <http://scop.mrc-1mb.cam.ac.uk/>
SCOP.
- <http://scop2.mrc-1mb.cam.ac.uk/>
SCOP2.

Ab initio Protein Structure Prediction

Rahul Kaushik, IIT Delhi, New Delhi, India

Ankita Singh, IIT Delhi, New Delhi, India and Banasthali Vidyapith, Banasthali, India

B Jayaram, IIT Delhi, New Delhi, India

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

Å	Angstrom	MC	Monte Carlo
AMBER	Assisted Molecular Building and Energy Refinement	MD	Molecular dynamics
CASP	Critical Assessment of Techniques for Protein Structure Prediction	MM	Molecular mechanics
CHARMM	Chemistry at HARvard Macromolecular Mechanics	NMR	Nuclear magnetic resonance
CNFs	Conditional Neural Fields	OPLS	Optimized potential for liquid simulations
CRFs	Conditional Random Fields	PDB	Protein Data Bank
Cryo-EM	Cryo Electron Microscopy	PSIBLAST	Position Specific Iterative Basic Local Alignment Search Tool
ENCAD	Energy Calculation and Dynamics	QM	Quantum mechanics
GPUs	Graphic processing units	RM2TS	Ramachandran Maps to Tertiary Structures
		RMSD	Root Mean Square Deviation
		SD	Structural difficulty
		UNIRES	United residue

Introduction

Recent successes in proteomics have led to a spate of sequence data. This data can benefit society only if one can decipher the functions and malfunctions of proteins, and this requires their structures, in addition to sequence information (Koga *et al.*, 2012; Bhattacharya, 2009; Grishin, 2001). Currently, almost a thousand fold gap exists between the number of known protein sequences in UniProtKB (~90 million sequences) (Boutet *et al.*, 2016) and the number of corresponding structures in the Protein Data Bank (PDB) (~0.13 million structures) (Berman *et al.*, 2007), as shown in Fig. 1. The urgency of determining protein structures is further underscored by various drug discovery endeavours. Protein structure elucidation from sequence is among the top hundred outstanding problems in modern science (Blundell, 1996).

Despite major developments in the field of experimental structure determination using X-ray crystallography, NMR and cryo-EM techniques (Shi, 2014; Chapman *et al.*, 2011; Raman *et al.*, 2010; Fernandez-Leiro and Scheres, 2016; Doerr, 2015), the mounting gap between known protein sequences and structures has created the need for reliable computational protein structure prediction methodologies (Marks *et al.*, 2012; Petrey and Honig, 2005; Baker and Sali, 2001). In the post-Anfinsen era, the field of protein structure prediction has made substantial progress, as chronicled through the biennial Critical Assessment of Techniques

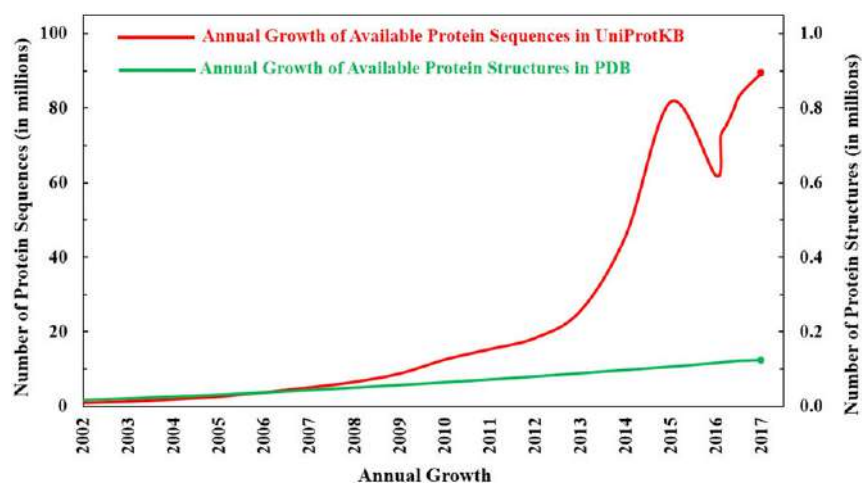


Fig. 1 A comparison of the annual growth rates of available protein sequences in UniProtKB and available protein structures in Protein Data Bank. Data source: UniProtKB and PDB.

for Protein Structure Prediction (CASP) experiments (Moult *et al.*, 2014), and by a continuous automated model evaluation with CAMEO (Haas *et al.*, 2013), among others.

The field of protein tertiary-structure prediction originated in physics-based molecular mechanics and dynamics (*ab initio* modeling) approaches (Pearlman *et al.*, 1995; Lindorff-Larsen *et al.*, 2012), but its success was restricted to small proteins because of its compute intensive nature (Kulik *et al.*, 2012; Lindorff-Larsen *et al.*, 2011; Jayaram *et al.*, 2006; DasGupta *et al.*, 2015). However, *ab initio* approaches offer the potential of predicting new folds (Huang *et al.*, 2016; Klepeis *et al.*, 2005; Mittal *et al.*, 2010). Subsequently, advances in bioinformatics, and data mining in particular, led to the development of some extremely popular knowledge-based methods (comparative modeling) which utilize information ingrained in experimentally solved protein structures (Ginalski, 2006; Shenoy and Jayaram, 2010). The success of comparative modeling approaches is limited by the availability of known reference structures, and this precludes discovery of new folds. Also, owing to the nature of the methodology adopted, comparative modeling approaches are not of much help in providing insights into the physico-chemical mechanism of protein folding (Dill and MacCallum, 2012; Garcia and Onuchic, 2005). A jump in the accuracy of protein structure prediction was realized with the development of integrated approaches (hybrid methods) that combine physical and knowledge-based methods (Zhang, 2008; Kim *et al.*, 2004; Jayaram *et al.*, 2014).

In this article, we discuss various methods for implementing *ab initio* approaches to protein structure prediction, mostly for small globular proteins. *Ab initio* structure prediction has several names such as *de novo* prediction, physics-based prediction, free modeling, *etc.*, which are used interchangeably. We will use the term *ab initio* prediction to encompass these diverse endeavours. Before we delve into the methodological details, we introduce here some basic concepts and assumptions important in protein structure prediction.

Ab initio Protein Structure Prediction

The pioneering work in the field molecular dynamics simulation by Karplus, Levitt and their coworkers laid the foundation for *ab initio* approaches (McCammon *et al.*, 1977; Levitt and Warshel, 1975; Levitt, 1976; Levitt and Sharon, 1988; Scheraga *et al.*, 2007). The field has been further accelerated by the landmark work on the folding of villin headpiece subdomain by Kollman and coworkers (Duan and Kollman, 1998). The efforts of Shaw and coworkers, with their special purpose Anton computer, which was designed to fold proteins in real time, have given a big boost to *ab initio* protein folding (Lindorff-Larsen *et al.*, 2011, 2012). Despite the recent successes enjoyed by *ab initio* approaches, a few road-blocks are yet to be circumvented, such as the methodology required for sampling and accurately identifying a native fold from the astronomical conformational space of larger proteins.

The axioms implicit in *ab initio* approaches, initially proposed by Anfinsen (1973) are that (i) the amino acid sequence of a protein uniquely determines its tertiary structure, and that (ii) the native conformation of a protein sequence corresponds to its global free-energy minimum. For *ab initio* protein structure prediction methods, it is obligatory to have a scoring function (typically a physics-based energy function) and a sampling method for searching the conformational space. In this section, we discuss some scoring functions used in *ab initio* approaches, and the most commonly used strategies for conformational sampling.

Scoring Functions

Essentially a scoring function has to mimic a free energy function. Scoring functions have to capture structures with minimum free energies. Noting that free energy which combines both enthalpy and entropy, is a statistical quantity which depends on an ensemble of structures and not a mechanical quantity which is calculable even for a single structure, formulating a scoring function to rank conformations free energetically is a difficult task. Based on their underlying principles, scoring functions, may be classified into two categories, *viz.* physics-based and statistics-based. Physics-based scoring functions are mathematical models that describe inter-atomic interactions (Duan and Kollman, 1998; Weiner *et al.*, 1984; Hagler and Lifson, 1974; Cornell *et al.*, 1995; Jorgensen and Tirado-Rives, 1988; Brooks *et al.*, 1983). Statistics-based functions, also known as knowledge-based functions or empirical-energy functions, are statistical models that are derived from various properties of native protein structures (Jayaram *et al.*, 2006; DasGupta *et al.*, 2015; Skolnick *et al.*, 1997; Samudrala and Moult, 1998; Shen and Sali, 2006). Usually, physics-based scoring functions account for bonded interactions *via* terms that describe bond lengths, bond angles, dihedral angles, *etc.*, and non-bonded interactions *via* terms that include van der Waals interactions, electrostatic interactions, hydrophobic effects, *etc.* Knowledge-based functions account for solvent accessibility, secondary structural preferences, torsion angle preferences, residue-residue pairwise potentials, packing fraction, *etc.*, and are derived from experimentally solved protein structures.

Physics-based functions

Ideally, atomic interactions can be best described using quantum mechanical (QM) calculations, and columbic interactions among the elementary particles involved (Kulik *et al.*, 2012). However, the potential use of quantum mechanical calculations in *ab initio* protein structure prediction cannot be explored, even for small proteins, because of their extremely compute intensive nature. Physics-based functions typically adopt the Born-Oppenheimer approximation and molecular mechanics (MM) instead,

Table 1 A list of force fields/software suites that implement physics-based energy scoring functions for *ab initio* protein structure prediction

Force field/software suite	Basic strategy for sampling and scoring	Availability
CHARMM	Molecular dynamics (MD)	https://www.charmm.org
AMBER	Molecular dynamics (MD)	http://www.ambermd.org
GROMOS	Molecular dynamics (MD)	http://www.gromos.net
OPLS	Molecular dynamics (MD)	http://zarbi.chem.yale.edu
ENCAD	Energy Calculation & Dynamics	http://depts.washington.edu
UNRES	Conformational space annealing (CSA)	http://www.unres.pl

giving rise to continual development of MM force fields, wherein the system (protein) may be an all-atom model or a coarse-grained model system. Some of the most commonly used all-atom model force fields (and software suites which adopt the MM force fields) include CHARMM (Chemistry at HARvard Macromolecular Mechanics) (Brooks *et al.*, 1983), AMBER (Assisted Model Building and Energy Refinement) (Pearlman *et al.*, 1995; Weiner *et al.*, 1984; Cornell *et al.*, 1995), OPLS (Optimized Potentials for Liquid Simulations) (Jorgensen and Tirado-Rives, 1988), and GROMOS (GROningen Molecular Simulation) (Brunne *et al.*, 1993; Van Der Spoel *et al.*, 2005). Additionally, CHARMM, AMBER and GROMOS also include united atom model force fields with a higher computational efficiency. These model systems differ from each other in their atom-type definitions, the force-field functional forms, and the parameters they use to account for inter-atomic interactions. Coarse-grained model systems include UNRES (UNited RESidue) (Liwo *et al.*, 1999), TOUCHSTONE (Skolnick *et al.*, 2003; Kihara *et al.*, 2001) and Martini (Marrink *et al.*, 2007). The UNRES coarse grained model system accounts only for two interaction sites, namely united side-chain and united peptide group, per residue, which offers ~1000 to 4000-fold time speed-up compared to all-atom model systems. The TOUCHSTONE coarse-grained model implements a reduced lattice-based model, which is used with replica exchange Monte Carlo approaches to account for short range interactions, local conformational stiffness, long range pairwise interactions, hydrophobic, and electrostatic interactions. The Martini coarse grain force field implements the mapping of four heavy atoms for individual coarse grained interaction, termed as bead. In Martini force field, four major bead categories are defined *viz.* charged, polar, non-polar and apolar. Energy Calculation and Dynamics (ENCAD) is another set of energy parameters designed for simulations of macromolecules with solvation, which focuses on energy conservation and reduces calculations by applying a distance cutoff (truncation) for nonbonded interactions (Levitt *et al.*, 1995). Table 1 summarizes some of the physics-based scoring functions along with their availability in the public domain.

Knowledge-based functions

Structural features and interaction patterns, derived from non-redundant datasets of experimentally determined high-resolution protein structures, are used for formulating various knowledge-based scoring functions. Knowledge-based scoring functions are also called statistics-based scoring functions because different statistical parameters are derived from the frequencies of interactions in experimentally determined structures that are favourable or seen in native proteins. These features and patterns may include pairwise interactions of residues, solvent accessible surface area, exposure of hydrophobic residues on the protein surface, packing density of protein secondary structural element and tertiary structures, etc. (Yang and Zhou, 2008; Laskowski *et al.*, 1993; Wiederstein and Sippl, 2007; Eisenberg *et al.*, 1997; Benkert *et al.*, 2009; Zhou and Skolnick, 2011). The probability of structural interaction patterns can be transformed to an energy function by implementing an inverse Boltzmann approach with a known probability reference state (derived from the reference dataset) as shown in Eq. (1).

$$\Delta E = -k_B T \ln \left\{ \frac{P_{obs}}{P_{ref}} \right\} \quad (1)$$

where, k_B is Boltzmann's constant, T is the thermodynamic temperature, P_{obs} is the probability of the predicted (observed) features, and P_{ref} is the probability of the reference feature, derived from the experimental dataset. The application of the inverse Boltzmann law assumes the mutual independence of the observed features and patterns, and their distribution in accordance with Boltzmann's Law. However, the mutual independence of the derived features is a severe approximation, and thus was not considered in early studies.

Statistics-based functions show a tendency to be skewed with respect to varying sizes of proteins and, thus may sometimes be misleading if they are not thoroughly validated on a large dataset of proteins of diverse sequence lengths. The selection and curation of the reference dataset, which is used for deriving structural features and interaction patterns, determines the accuracy of knowledge-based scoring functions. In Table 2, some successful knowledge-based scoring functions are listed along with their availability.

Physics and knowledge-based integrated functions

Owing to the individual limitations of physics-based and knowledge-based scoring functions, integrated scoring functions have been developed that couple the two approaches to improve prediction accuracies. These combined approaches extract certain

Table 2 A list of programs that implement knowledge-based potentials for *ab initio* protein structure prediction

Algorithm	Basic assumption	Availability
dDFIRE	Pair-wise atomic interactions and dipole-dipole interactions based energy scoring	http://sparks-lab.org/yueyang/DFIRE
Procheck	Stereo chemical quantification based on statistical potentials	http://services.mbi.ucla.edu/PROCHECK
ProSA	Sequence length dependent knowledge-based C α potentials of mean force	https://www.came.sbg.ac.at/prosa.php
Verify3D	1D–3D profiling via statistical potential derived from experimental structures	http://services.mbi.ucla.edu/Verify_3D
QMEAN	Qualitative model energy analysis composite scoring function	https://swissmodel.expasy.org/qmean
GOAP	Generalized Orientation-dependent, All-atom statistical Potential	http://cssb.biology.gatech.edu/GOAP

features and patterns from experimentally-solved protein structures, and implement them in physics-based energy scoring functions to identify correctly folded conformations (Davis *et al.*, 2007; Mishra *et al.*, 2013; Colovos and Yeates, 1993; Mishra *et al.*, 2014; Ray *et al.*, 2012; Singh *et al.*, 2016; Melo and Feytmans, 1998).

Conformation Sampling

Classical protein folding studies on small proteins implemented physics-based potential functions (force fields) for use in molecular dynamics (MD) simulations which provided insights into the molecular mechanisms of folding pathways. The increasing quality of the energy functions is a major strength of MD approaches, but purely MD based protein folding is feasible only for small proteins (less than 100 amino acid residues), unless simplified models are used. Continuously increasing compute power and resources offer the possibility of performing long (milliseconds and longer) MD simulations in order to explore the mechanisms of folding and unfolding of even larger proteins. For instance, MD algorithms implemented on graphics processing units (GPUs) have significantly contributed to accelerating such calculations. Also, the development of enormously parallel clusters (e.g., BlueWaters) (Mendes *et al.*, 2014) and special-purpose supercomputers (e.g., Anton) (Shaw *et al.*, 2014) has created opportunities for performing micro to millisecond simulations on biomolecules (Lindorff-Larsen *et al.*, 2012; Zhang, 2008). Another frequently used approach in molecular dynamics is Replica Exchange Molecular Dynamics (REMD), which implements multiple parallel simulations for the same biomolecule, with each simulation (termed a replica) running at a different temperature within a defined temperature scale. These parallel simulations may exchange their temperatures at intervals with non-zero probability (Sugita and Okamoto, 1999; Sugita *et al.*, 2000). The efficiency of REMD is strongly dependent upon the number of replicas and the range of selected temperatures. REMD approaches can address the multiple local minima problem more efficiently than conventional molecular dynamics simulations, which are carried out at fixed temperature. Further details of REMD-algorithm based approaches can be gleaned from more specific review articles (Zhou, 2007; Kar *et al.*, 2009; Sugita *et al.*, 2012; Chen *et al.*, 2015). However, the optimal use of computational resources for achieving consistent success for all types of proteins is highly dependent upon the accuracy of the physical model/force field.

In this section, we briefly summarize the chronological progress of the field of *ab initio* conformational sampling and the various force fields which have been implemented. In the protocol advanced by Beveridge and coworkers, medium accuracy structures of small proteins are identified by enabling the MD simulation to escape from meta-stable local minima by using an integrated energy function adopted from AMBER. The integrated energy function includes a solvent dielectric polarization function, van der Waals interactions, and cavitation effects, and uses a Monte Carlo simulated-annealing search scheme (Liu and Beveridge, 2002). The *ab initio* method of Gibbs *et al.* predicted structures of small proteins (up to 38 residues) to within 3 Å RMSD, allowing only the rotatable backbone dihedral angles (ϕ and Ψ) to change, considering the side chains as fixed, by implementing a physicochemical feature-based force field to evaluate the energies (Gibbs *et al.*, 2001). Scheraga and coworkers implemented a global optimization procedure based on a modified united-residue force field (UNRES) and conformational space annealing, with considerable success for helical proteins (Liwo *et al.*, 1999). Rose and coworkers proposed LINUS, a Monte Carlo simulation-based tool, which mainly focused on the impact of steric interactions and conformational entropy, with a simplified scoring function accounting for hydrophobic interactions and hydrogen bonds. Cubic or tetrahedral lattice representations have been used previously for reducing the size of the conformational space (Srinivasan *et al.*, 2004). Skolnick and coworkers proposed a face-centered cubic lattice model accounting for hydrophobic interactions, hydrophobic-polar repulsive interactions, and polar-polar interactions (Pokarowski *et al.*, 2003). Jayaram *et al.* proposed an *ab initio* methodology (christened *Bhageerath*) based on physics-based potentials integrated with biophysical filters, and predicted medium accuracy model structures (3–6 Å RMSD) for small proteins (Jayaram *et al.*, 2006; Jayaram *et al.*, 2012). Recently, fragment library approaches have been used to generate initial models for performing simulations, which circumvent the earliest steps in folding (such as local structure formation). Fragment libraries may be directly extracted from experimentally solved structures or on the basis of various features of amino acids. A small set of structural fragments is sufficient to accurately model protein structures, as demonstrated by Levitt and coworkers, who used different sized libraries with simulated-annealing k-means clustering (Kolodny *et al.*, 2002), and later by Jayaram and coworkers using tripeptide based backbone dihedral angle preferences derived from non-redundant experimental structures for predicting the tertiary structure of small proteins (DasGupta *et al.*, 2015).

Automation of *Ab initio* Structure Prediction

The different approaches for *ab initio* protein tertiary-structure prediction, as implemented in popular software/tools, are discussed here. It may be noted that the list of servers/ software/ tools presented here is not exhaustive, and does not cover all the available *ab initio* methodologies, for which the reader's indulgence is sought. Methodologies are listed in chronological order of their development and availability to the scientific community. Additionally, a brief summary of the various *ab initio* methods is provided in a tabular form for a quick reference.

Rosetta

This is an *ab initio* tertiary-structure prediction methodology for small proteins implemented as a web server. It is also made available for local installation on linux based computers, although, the standalone version requires huge compute resources and, single CPU machines may take ages to produce a structural model. The web server is more popular, and can predict a model structure in a few days, depending upon the queue. Initial versions of Rosetta implemented a simplified simulated annealing protocol for performing fragment assembly of peptide 9-mers and 3-mers derived from protein structures having similar local sequences, using Bayesian scoring functions to predict protein tertiary structures of small proteins (Das and Baker, 2008). A general workflow of the Rosetta *de novo* methodology is shown in Fig. 2. Since the Rosetta method is restricted to small proteins, a hybrid methodology has been developed that integrates the *ab initio* (Rosetta method) and homology-based methods in the Robetta server (Kim *et al.*, 2004). The integrated approach has achieved considerable success in the field of protein structure prediction.

QUARK

QUARK performs *ab initio* prediction using small fragment structural libraries of up to 20 amino acid residues, followed by replica-exchange Monte Carlo simulations with an atomic-level knowledge-based scoring function (Xu and Zhang, 2012). For a given target protein, secondary structure prediction is performed with PSSpred, (Yan *et al.*, 2013) and a sequence profile generated from a PSI-BLAST multiple sequence alignment (Altschul *et al.*, 1997). Further, neural networks are implemented for predicting residue-specific

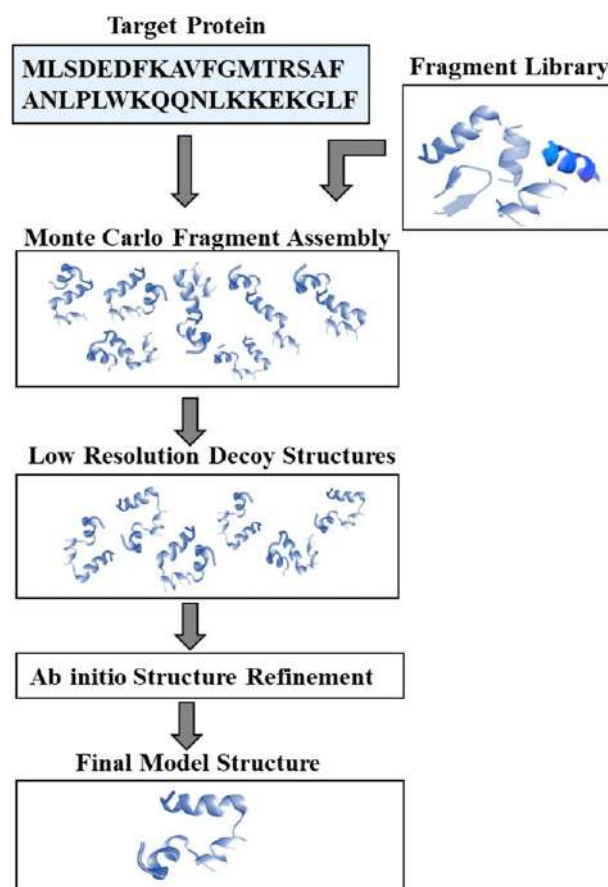


Fig. 2 A workflow of Rosetta methodology for performing *ab initio* protein tertiary structure prediction for a given target protein sequence. Based on methodology explained in Das, R., Baker, D., 2008. Macromolecular Modeling with Rosetta. Annual Review of Biochemistry, 77(1), 363–382.

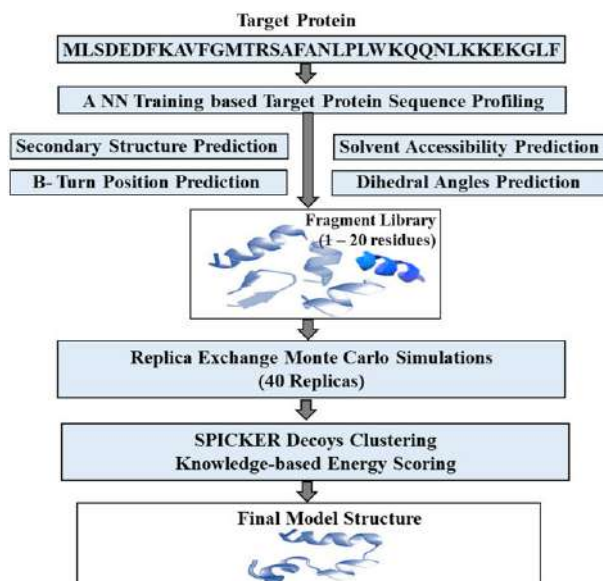


Fig. 3 A simplified flowchart of QUARK *ab initio* protein tertiary structure prediction. Based on methodology explained in Xu, D., Zhang, Y., 2012. *Ab initio* protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function and Bioinformatics*, 80(7), 1715–1735.

solvent accessibilities, backbone dihedral angles (ϕ and Ψ), and β -turn positions. The sequence profile, predicted secondary structure, solvent accessibilities, backbone dihedral angles, and β -turn positions are used for generating structural fragments of variable lengths (up to 20 residues) for segments of target sequence. The full length model structures are further scanned to filter out similar structures using SPICKER, followed by full atomic refinement with a knowledge-based scoring function (Zhang and Skolnick, 2004a). A workflow of QUARK *ab initio* prediction is shown in Fig. 3. Since success of protein structure prediction using the QUARK method is restricted to only small proteins (up to 200 amino acid residues), the QUARK *ab initio* methodology has been integrated with comparative-modeling based fold recognition and threading approaches in the I-TASSER software suite (Yang *et al.*, 2014) in order to allow the prediction of larger protein structures with improved efficiency.

TOUCHSTONE

TOUCHSTONE performs *ab initio* prediction by implementing a threading approach, which is based on secondary and tertiary structural parameters derived from experimentally solved protein structures (Skolnick *et al.*, 2003; Kihara *et al.*, 2001). These parameters include consensus contacts and local secondary structures at fragment levels. The conformational space is explored with the help of replica-exchange Monte Carlo to generate a reduced lattice-based protein model. Further, decoy structures are clustered and scored with a knowledge-based residue-specific heavy-atom pair potential to select representative structures. Confidence in the prediction accuracy is evaluated based on number of predicted contacts and the number of simulated contacts from the replica-exchange Monte Carlo simulations. The methodology showed considerable success on a validation dataset of 65 small proteins (up to 150 amino acid residues) in predicting model structures within 6.5 Å RMSD of their respective native structures. The updated version of the methodology (released two years later) implemented different short-range and long-range knowledge-based potentials and this resulted in improved predictions.

Bhageerath

Bhageerath is an energy-based software suite for restricting the conformational search space for small globular proteins (Jayaram *et al.*, 2006; Levitt *et al.*, 1995). The automated protocol of Bhageerath spans eight different modules that integrate physics-based potentials with knowledge-based biophysical filters, and produces ten representative model structures for an input protein sequence and its corresponding secondary structure. The performance of Bhageerath was benchmarked against homology-based methods and found to perform consistently better for predicting medium accuracy model structures for small globular proteins (up to 100 amino acid residues). The first module generates a coarse-grained model given the input target sequence and predicted secondary structure in the form of helices, strands and coils. Backbone dihedral angles (ϕ and Ψ) based conformational sampling of the amino acid residues representing coil region is performed in the second module to generate trial structures. Seven dihedral angles from each stretch of residues representing coil region are selected. For each selected dihedral angle, two preferred values for ϕ and Ψ are adopted from experimental protein structures which results into 2^7 conformation (128 conformations per loop).

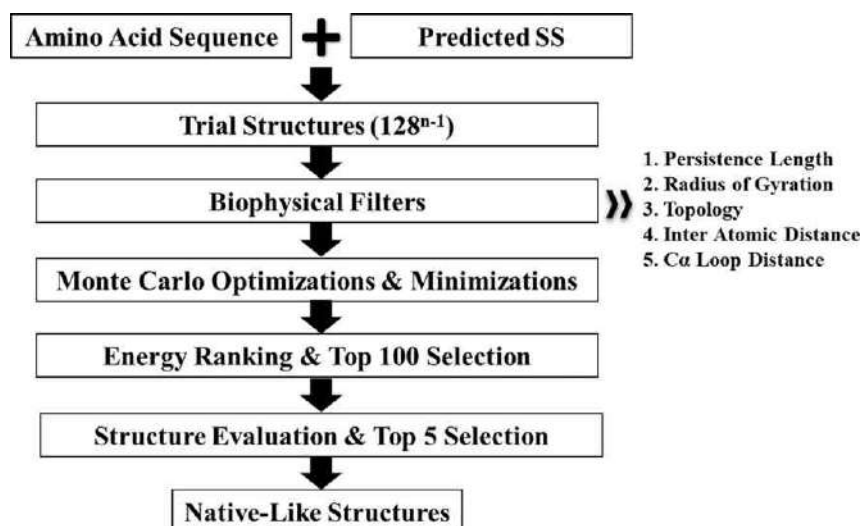


Fig. 4 A flowchart of Bhageerath *ab initio* protein structure prediction methodology. Based on methodology explained in Jayaram, B., Bhushan, K., Shenoy, S.R. *et al.*, 2006. Bhageerath: An energy based web enabled computer software suite for limiting the search space of tertiary structures of small globular proteins. *Nucleic Acids Research*, 34(21), 6195–6204.

For an individual stretch of amino acid residues representing a coil region in the predicted secondary structure, this module generates 128 different conformations. When the number of coil regions (excluding N-terminal and C-terminal coil regions) in the predicted secondary structure increases, the number of generated conformations increases exponentially. For instance, for a protein sequence with ‘*n*’ number of secondary structural elements in the predicted secondary structures (number of helices + number of strands), a total of $128^{(n-1)}$ trial conformations are generated. In the third module, the trial conformations are scanned through biophysical filters *viz.* persistence length and radius of gyration (Narang *et al.*, 2005), to exclude improbable conformations. Steric clashes and overlaps are rectified in the fourth module using Monte Carlo sampling in the backbone dihedral angle space. Further, an implicit solvent energy minimization with a distance dependent dielectric and side chain optimization is performed in the fifth module. The sixth module performs energy ranking based on an all-atom energy-based empirical scoring function, selecting the top 100 lowest energy structures (Narang *et al.*, 2006). A protein regularity index, which checks the compatibility of backbone dihedral angles of predicted conformations over experimentally solved protein structures (Thukral *et al.*, 2007), is implemented in the next module to reduce the number of candidate structures. Finally, in the eighth module, topologically equivalent structures are screened out and the top 10 structures are selected based on solvent surface accessibility (lower values are preferred), ranked and provided to the user. A flowchart of the automated Bhageerath pipeline is depicted in Fig. 4. The Bhageerath *ab initio* methodology can perform structure prediction for small proteins (up to 100 amino acid residues), and is integrated with homology-based methods in the BhageerathH⁺ software suite (Jayaram *et al.*, 2012, 2014; Singh *et al.*, 2016; Dhingra and Jayaram, 2013; Kaushik and Jayaram, 2016) to perform reliable structure prediction for large proteins.

ASTRO-FOLD

ASTRO-FOLD performs *ab initio* protein tertiary-structure prediction based on a combinatorial and global optimization framework (Klepeis and Floudas, 2003). The initial version of ASTRO-FOLD implemented a hierarchical method that integrated an all-atom energy function, global optimization algorithm, conformational space annealing, and MD simulations in dihedral angle conformational space. More recently, an updated version of ASTRO-FOLD has been released (christened ASTRO-FOLD 2.0) that predicts the secondary structure of a target protein using various statistical potentials, followed by contact prediction, and loop prediction (Subramani *et al.*, 2012). These predictions are used for deriving various restraints, such as dihedral angle and C α –C α distance restraints. The restraints are used for further conformational sampling using a combinatorial and global optimization algorithm. A simplified workflow of ASTRO-FOLD 2.0 is shown in Fig. 5.

RaptorX-FM

Probabilistic graphical models are used for deriving relationships among backbone dihedral angles, sequence profiles, and secondary structural elements, which results in more accurate backbone dihedral angle prediction and more efficient conformational sampling. The method performs better on all-alpha proteins, with up to 150 amino acid residues, and small all-beta proteins, with up to 90 amino acid residues. For conformation sampling, the probabilistic graphical models, Conditional Random Fields (CRF), which utilize a linear relationship between backbone dihedral angles and the sequence profile, and Conditional Neural Fields (CNF), which employ a neural-network based nonlinear relationship between backbone dihedral angles and the sequence profile, are coupled with

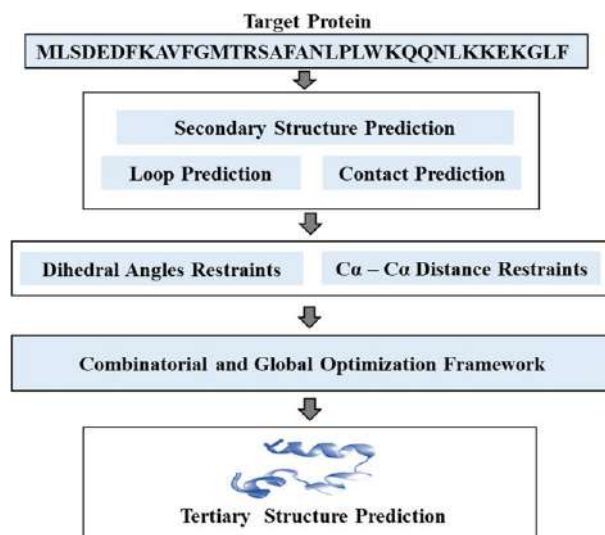


Fig. 5 A workflow of ASTRO-FOLD 2.0 for *ab initio* protein structure prediction. Based on methodology explained in Subramani, A., Wei, Y., & Floudas, C.A., 2012. ASTRO-FOLD 2.0: An enhanced framework for protein structure prediction. *AIChE Journal*, 58(5), 1619–1637.

replica exchange Monte Carlo methods (Zhao *et al.*, 2010). The latest RaptorX methodology implements an integrated framework for *ab initio* and comparative structure prediction using the RaptorX server (Källberg *et al.*, 2012).

RM2TS

RM2TS uses tripeptide based backbone dihedral angle preferences derived from non-redundant experimental structures as reference to predict the tertiary structure of small proteins (DasGupta *et al.*, 2015). The allowed backbone-dihedral angle conformational space of Ramachandran maps is divided into 27 classes and has been demonstrated to be sufficient for predicting a model structure to within 5 Å RMSD, for small globular proteins with up to 100 amino acid residues. The backbone dihedral angle preferences at the tripeptide level, when coupled with predicted secondary structural elements, reduce the conventional backbone dihedral angle conformational space by a factor of ten. This reduced conformational space results in a time efficient method for structure generation with reasonably high accuracy. The tertiary structure of a protein sequence can be predicted on the basis of the backbone dihedral angles, which are derived from a precomputed look-up table, within 2–3 min on a single processor computer. Further, a higher level of accuracy can be achieved if the target sequence is complemented by an accurate secondary structure prediction. The workflow of structure prediction using RM2TS is shown in Fig. 6. A modified RM2TS methodology (with backbone dihedral angle preferences considering 11 residues window) is implemented in the BhageerathH⁺ suite.

UniCon3D

UniCon3D is a *de novo* structure prediction methodology that implements united residue conformational sampling using hierarchical probabilistic sampling. The concept of protein folding *via* sequential stabilization is utilized in this method (Bhattacharya *et al.*, 2016). The local structural preferences, in terms of backbone and side chain angle parameters are used for conformational sampling, coupled with an integrated physics-based and knowledge-based energy scoring function. Since the backbone and side chain angle parameters are considered simultaneously, the energetics of side chain solvation/desolvation are accounted for, resulting in better conformational sampling. A simulated annealing algorithm is implemented for potential energy minimization of the united-residue polypeptide conformation. Conformational sampling proceeds by stepwise construction of small fragments and their assembly into full length models. The secondary structural information utilized in the methodology considers an eight-class secondary structural element classification (3₁₀ helices, α -helices, π -helices, β -strands, β -bridges, turns, bends and coils) instead of the conventional three-class secondary structural element classification (helices, strands and coils), which brings additional accuracy to the method. Fig. 7 shows a simplified workflow of the use of UniCon3D for structure prediction using its hierarchical probabilistic sampling. The UniCon3D methodology is combined with a machine-learning-based contact-prediction method for template-based modeling in MULTICOM (Cheng *et al.*, 2012).

Accuracy Measures for Comparing Structures

The measures for similarity among protein structures have been evolving continuously over the last two decades. Several measures for quantifying the structural differences between two structures used. In this section we briefly explain some of the frequently used measures.

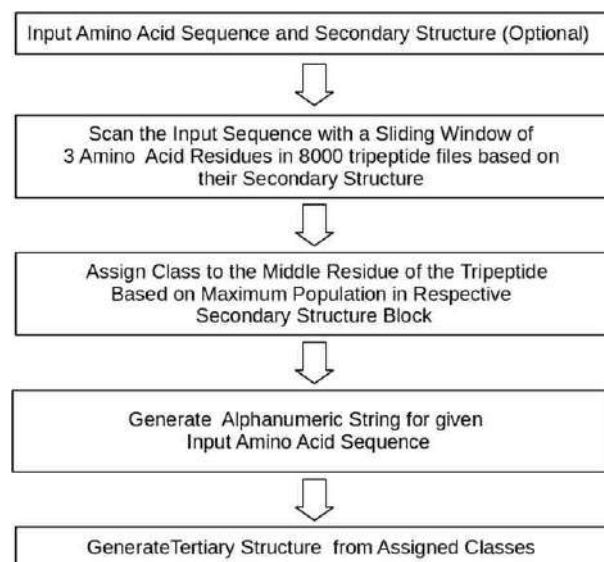


Fig. 6 A workflow of RM2TS protein tertiary structure prediction. Based on methodology explained in DasGupta, D., Kaushik, R., Jayaram, B., 2015. From Ramachandran maps to tertiary structures of proteins. *Journal of Physical Chemistry B*, 119(34), 11136–11145.

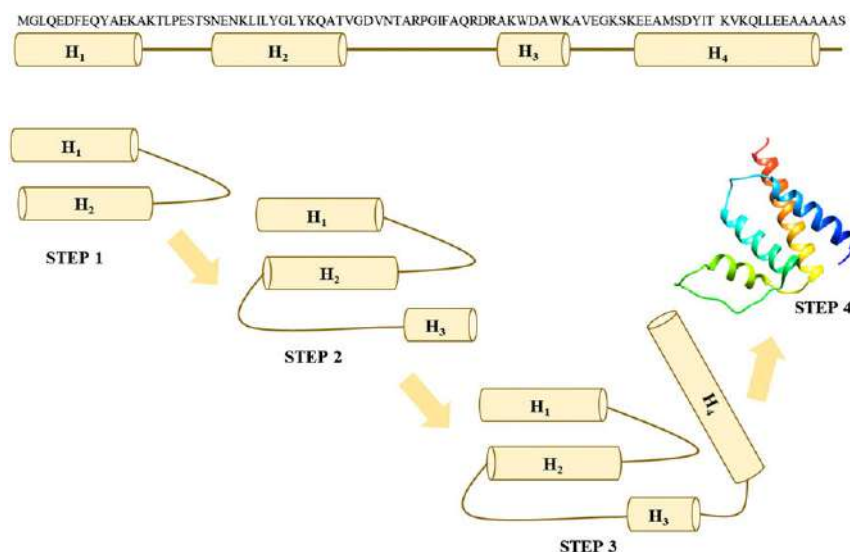


Fig. 7 A workflow of UniCon3D protein tertiary structure prediction where stepwise construction of small fragments and their assembly into full length models is depicted. Based on methodology explained in Bhattacharya, D., Cao, R., Cheng, J., 2016. UniCon3D: De novo protein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. *Bioinformatics*, 32(18), 2791–2799.

Root Mean Square Deviation (RMSD)

RMSD is a measure of average distance between the atoms of optimally superposed protein structures (rigid body superposition) (Coutsias *et al.*, 2004). The RMSD can be calculated for only C α atoms, backbone heavy atoms (N, C, C α and O), or all atoms of the superposed structures. The C α and backbone RMSD provide a measure of similarity without considering the side chain orientations. All atom RMSD is considered as the most informative measure among all RMSDs. For exactly identical protein structures, the RMSD should be zero. RMSD between related proteins increases as the level of similarity decreases. In the context of protein structure prediction, predicted structures within 3 Å RMSD of their native structure are considered as high accuracy models, which can be used for various protein structure-based studies such as function annotation and drug design. Also, the structures within 5 Å RMSD can provide insights into the overall topology/fold of protein structures. Despite the availability of several other measures, and despite its known sensitivity to outliers, RMSD is still the most popular and widely accepted measure of structural similarity, especially among non-computational biologists.

Global Distance Test (GDT) Score

The GDT Score is the percentage of C α pairs falling within a given distance cutoff (in Å) of two optimally superposed protein structures (Zemla, 2003), typically a predicted structure and a crystallographically determined native structure. Depending upon the distance cutoff, GDT is divided into two categories, GDT-HA (GDT- High Accuracy) where usually a 2 Å cutoff is used, and GDT-TS (GDT-Template Score) where usually a 4 Å cutoff is used. The GDT score varies from 0–100 with 0 being the worst and 100 being the best. The GDT Score is used as one of the assessment parameters in CASP experiments.

Template Modeling (TM) Score

The TM Score is another measure of structural similarity between two superposed protein structures. It varies from 0 to 1, where 0 is considered as the worst and 1 as the best match (Zhang and Skolnick, 2004b). The TM Score is more sensitive to global topology than to local sub-structures.

Apart from RMSD, GDT Score and TM Score, there is a long list of parameters/methods for measuring the structural similarity which have been used in the CASP experiments. Some such scores are the Sphere Grinder (SG) Score, Global IDDT Score, CAD Score, Quality Control (QC) Score, Accuracy Self Estimate (ASE) Score, *etc.* (see Relevant Website section). It is worth mentioning that the formulae used for assessment in CASP experiments are not fixed, and new scores are introduced while old ones are dropped in every round of the CASP experiments. For instance, ASE and CAD scores were introduced in CASP12, while Mol-Probity, QC, and Contact Scores used in CASP11, were dropped. Essentially, from a modeling perspective, these measures are supposed to help identify how far the model structure is from its native conformation (Table 3).

Analysis and Assessment

Large scale genome sequencing projects have resulted in various structural genomics initiatives that seek to determine the maximum number of possible protein folds. New folds can be either explored through time consuming and expensive experimental methods or *via* computational protein-structure prediction. Sequence-based protein structure predictions using *ab initio* approaches are governed by protein folding energetics or statistical preferences, and do not explicitly need an experimental template structure. The most common strategy in *ab initio* protein structure prediction comprises sampling conformational space, steered by a force field or a scoring function and/or various sequence features, to generate a large set of candidate structures, followed by selection of native-like structures using a scoring function. Alternatively, certain methods implement clustering algorithms to reduce the number of conformations that must be scored in the latter phase of the protocol. The representative conformations of these clusters are subjected to structure refinement, and rescored for native-like structure selection. In most of the purely physics-

Table 3 A list programs that implement integrated models of physics-based energy scoring and knowledge-based potentials for *ab initio* protein structure prediction

Algorithm	Basic assumption	Availability
MolProbity	An all atom contacts, steric clashes and dihedral based statistical scoring function	www.molprobity.biochem.duke.edu
pcSM	Euclidian distance, accessibility, secondary structure propensity, intramolecular energy	www.scfbio-iitd.res.in/pcSM
Errat	A quadratic error function based statistical potential for atomic interactions	www.services.mbi.ucla.edu/ERRAT
D2N	Known universalities in spatial organization of soluble proteins and	www.scfbio-iitd.res.in/D2N
ProQ	A Neural network based method using atom-atom and atom-residue contacts	www.sbc.su.se/~bjornw/ProQ
ProTSAV	An integrated scoring function accounting steric clashes, structural packing, dihedral distribution, solvent accessibility	www.scfbio-iitd.res.in/protsav.jsp
ANOLEA	A non-local energy profile calculations <i>via</i> atomic mean force potential and checks packing quality of protein conformations	http://www.melolab.org/anolea

Table 4 A list of *ab initio* protein structure prediction methodologies that are integrated with comparative modeling approaches to efficiently predict protein structures without sequence length restrictions

Ab initio method	Hybrid method	Availability of hybrid method
Rosetta	Robetta Server	www.rosetta.bakerlab.org
QUARK	Zhang Server	www.zhanglab.ccmb.med.umich.edu
Bhageerath	BhageerathH ⁺	www.scfbio-iitd.res.in/bhageerathH +
RaptorX-FM	RaptorX Server	www.raptorx.uchicago.edu
RM2TS	BhageerathH ⁺	www.scfbio-iitd.res.in/bhageerathH +
UniCon3D	MULTICOM	www.sysbio.rnet.missouri.edu/multicom_cluster

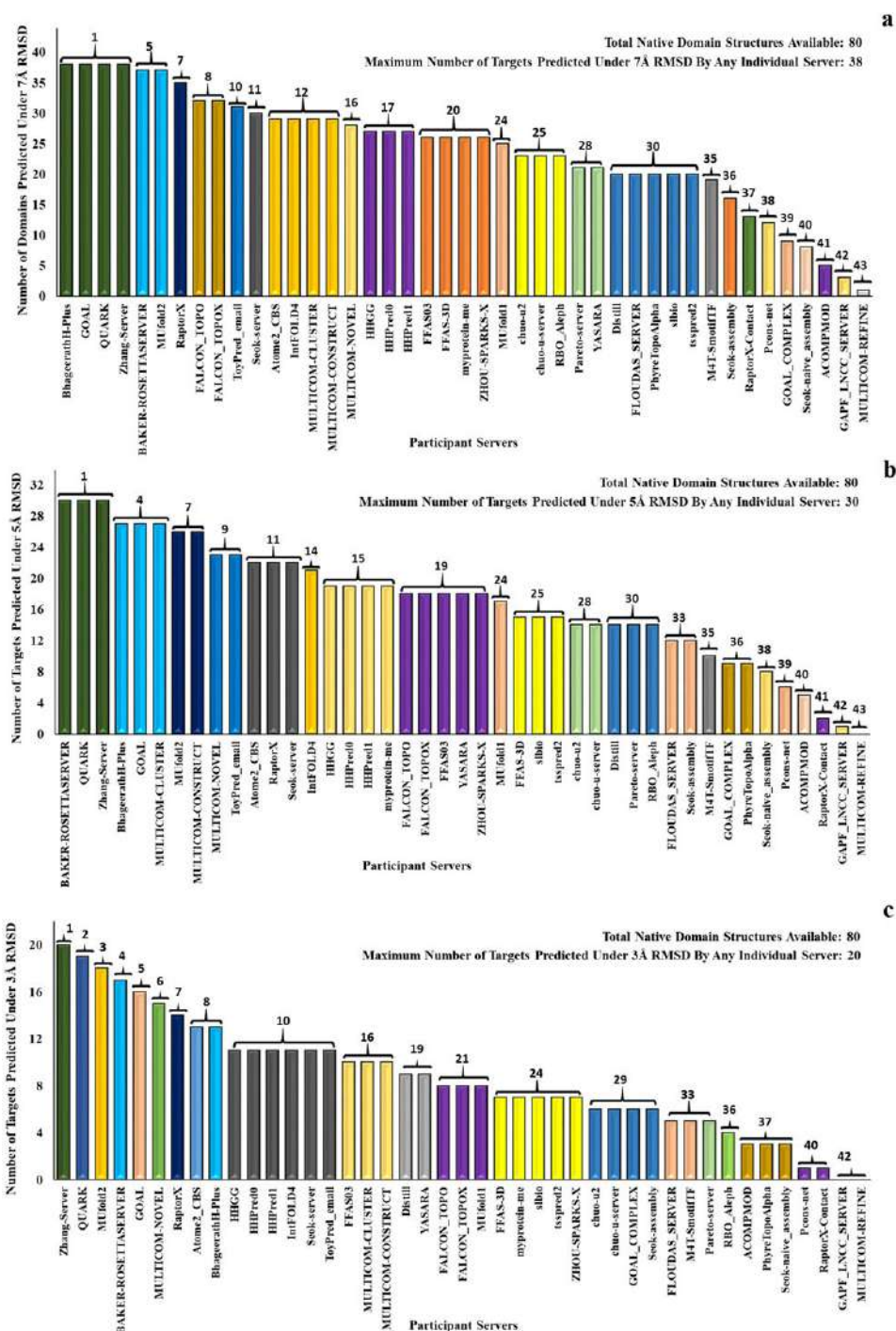


Fig. 8 A performance quantification of automated protein tertiary structure prediction servers which participated in the 12th round of CASP experiments. (a) For predicting low-accuracy model structures (under 7 Å RMSD), (b) For predicting medium-accuracy model structures (under 5 Å RMSD) and (c) For predicting high-accuracy model structures (under 3 Å RMSD). Data source: http://predictioncenter.org/download_area/CASP12/server_predictions/.

based methods, the conformational sampling space is explored using molecular dynamics and Monte Carlo simulations, guided by various force field parameters. At each step, the new conformation is evaluated using a scoring function, which makes these approaches computationally intensive and restricts their success to small proteins. The average length of protein sequences available in UniProtKB/Swiss-Prot (358 amino acid residues) suggests that *ab initio* methods may not be able to independently predict structures for most protein sequences. However, *ab initio* methods have been successfully integrated with comparative modeling approaches to enhance their time efficiency, and thus, their ability to predict structures of larger proteins. Table 4 lists

some *ab initio* methods (discussed above), which have been integrated into hybrid methods that achieve higher accuracy protein structure prediction without sequence length restrictions.

The performance of 43 protein structure prediction servers, participated in the 12th round of the CASP experiment (held from 2nd May to 12th July 2016), is compared *via* an in-house assessment in terms of number of structures predicted within specified RMSD ranges of the target structures: 7 Å (low-accuracy predictions), 5 Å (medium-accuracy predictions) and 3 Å (high-accuracy predictions). The assessment performed here accounts for consistency (in terms of frequencies) of accurate prediction in specified ranges of RMSD for CASP12 target proteins. The automated servers include purely *ab initio* servers, purely homology-based servers, and *ab initio*/homology-based hybrid servers. The performance comparison of different methodologies for predicting low, medium and high accuracy model structures is shown in Fig. 8. It can be observed that in all categories, servers that implement hybrid methodologies for structure prediction perform well. The different hybrid methodologies discussed here implement *ab initio* and comparative modeling components in different ways and thus perform differently on the same protein. It may happen that one server may gain while the others may have lost. Thus, in protein structure prediction regime, it is advantageous to adopt a consensus approach *via* predicting model structures from different servers, followed by a metasever approach for quality assessment.

Conclusions and Perspectives

Over the years, successful *ab initio* structure prediction strategies have metamorphosed into hybrid methodologies that can tackle proteins of any size and complexity. However, results from the recent CASP12 experiment suggest there is considerable room for further improvement. For instance, the best individual performance in the low-accuracy category (i.e., under 7 Å RMSD) was 38 out of a total of 80 domain targets (i.e. 48% success rate), which declined to 30/80 domain targets (38% success rate) for medium-accuracy predictions (i.e., under 5 Å RMSD), and to only 20/80 domain targets (25% success rate) for high accuracy predictions (i.e., under 3 Å RMSD). Considering that CASP targets are difficult to model, these success rates are of course the lower limits of the current status of the field. High accuracy predictions can be directly used for identifying ligands, modeling protein-protein interactions, functional characterization, and other structure-based drug discovery endeavours. Thus, there is a need for improved conformational sampling and scoring as well as structure refinement.

Acknowledgements

Support from the Department of Biotechnology, Govt. of India and SERB, Govt. of India to the Supercomputing Facility for Bioinformatics and Computational Biology (SCFBio), IIT Delhi, is gratefully acknowledged.

See also: Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Algorithms Foundations. Biomolecular Structures: Prediction, Identification and Analyses. Drug Repurposing and Multi-Target Therapies. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Protein Structural Bioinformatics: An Overview. Protein Three-Dimensional Structure Prediction. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Study of The Variability of The Native Protein Structure

References

- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25 (17), 3389–3402. Available at: <https://doi.org/10.1093/nar/25.17.3389>.
- Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. *Science* 181 (4096), 223–230. Available at: <https://doi.org/10.1126/science.181.4096.223>.
- Baker, D., Sali, A., 2001. Protein structure prediction and structural genomics. *Science* 294 (5540), 93–96. Available at: <https://doi.org/10.1126/science.1065659>.
- Benkert, P., Künzli, M., Schwede, T., 2009. QMEAN server for protein model quality estimation. *Nucleic Acids Res* 37 (Web Server issue), W510–W514. Available at: <https://doi.org/10.1093/nar/gkp322>.
- Berman, H.M., Henrick, K., Nakamura, H., *et al.*, 2007. Realism about PDB. *Nature Biotechnology* 25 (8), 845–846. Available at: <https://doi.org/10.1038/nbt0807-845>.
- Bhattacharya, A., 2009. Protein structures: Structures of desire. *Nature* 459 (7243), 24–27. Available at: <https://doi.org/10.1038/459024a>.
- Bhattacharya, D., Cao, R., Cheng, J., 2016. UniCon3D: De novo protein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. *Bioinformatics* 32 (18), 2791–2799. Available at: <https://doi.org/10.1093/bioinformatics/btw316>.
- Blundell, T.L., 1996. Structure-based drug design. *Nature* 384 (6604 Suppl.), S23–S26. Available at: <https://doi.org/10.1038/384023a0>.
- Boutet, E., Lieberherr, D., Tognolli, M., *et al.*, 2016. Uniprotkb/Swiss-Prot, the manually annotated section of the UniProt knowledgebase: How to use the entry view. *Methods in Molecular Biology* 1374, 23–54. Available at: https://doi.org/10.1007/978-1-4939-3167-5_2.
- Brooks, B.R., Brucoleri, R.E., Olafson, B.D., *et al.*, 1983. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry* 4 (2), 187–217. Available at: <https://doi.org/10.1002/jcc.540040211>.
- Brunne, R.M., van Gunsteren, W.F., Brüschweiler, R., Ernst, R.R., 1993. Molecular dynamics simulation of the proline conformational equilibrium and dynamics in antamanide using the GROMOS force field. *Journal of the American Chemical Society* 115 (11), 4764–4768. Available at: <https://doi.org/10.1021/ja00064a041>.

- Chapman, H.N., Fromme, P., Barty, A., *et al.*, 2011. Femtosecond X-ray protein nanocrystallography. *Nature* 470 (7332), 73–77. Available at: <https://doi.org/10.1038/nature09750>.
- Chen, C., Xiao, Y., Huang, Y., 2015. Improving the replica-exchange molecular-dynamics method for efficient sampling in the temperature space. *Physical Review E* 91 (5). Available at: <https://doi.org/10.1103/PhysRevE.91.052708>.
- Cheng, J., Li, J., Wang, Z., Eickholt, J., Deng, X., 2012. The MULTICOM toolbox for protein structure prediction. *BMC Bioinformatics* 13 (1), 65. Available at: <https://doi.org/10.1186/1471-2105-13-65>.
- Colovos, C., Yeates, T.O., 1993. Verification of protein structures: Patterns of nonbonded atomic interactions. *Protein Science* 2 (9), 1511–1519. Available at: <https://doi.org/10.1002/pro.5560020916>.
- Cornell, W.D., Cieplak, P., Bayly, C.I., *et al.*, 1995. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society* 117 (19), 5179–5197. Available at: <https://doi.org/10.1021/ja00124a002>.
- Coutsias, E.A., Seok, C., Dill, K.A., 2004. Using quaternions to calculate RMSD. *Journal of Computational Chemistry* 25 (15), 1849–1857. Available at: <https://doi.org/10.1002/jcc.20110>.
- DasGupta, D., Kaushik, R., Jayaram, B., 2015. From Ramachandran maps to tertiary structures of proteins. *Journal of Physical Chemistry B* 119 (34), 11136–11145. Available at: <https://doi.org/10.1021/acs.jpcb.5b02999>.
- Das, R., Baker, D., 2008. Macromolecular modeling with Rosetta. *Annual Review of Biochemistry* 77 (1), 363–382. Available at: <https://doi.org/10.1146/annurev.biochem.77.062906.171838>.
- Davis, I.W., Leaver-Fay, A., Chen, V.B., *et al.*, 2007. MolProbity: All-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Research* 35 (Suppl. 2), Available at: <https://doi.org/10.1093/nar/gkm216>.
- Dhingra, P., Jayaram, B., 2013. A homology/ab initio hybrid algorithm for sampling near-native protein conformations. *Journal of Computational Chemistry* 34 (22), 1925–1936. Available at: <https://doi.org/10.1002/jcc.23339>.
- Dill, K.A., MacCallum, J.L., 2012. The protein-folding problem, 50 years on. *Science* 338 (6110), 1042–1046. Available at: <https://doi.org/10.1126/science.1219021>.
- Doerr, A., 2015. Single-particle cryo-electron microscopy. *Nature Methods* 13 (1), 23. Available at: <https://doi.org/10.1038/nmeth.3700>.
- Duan, Y., Kollman, P.A., 1998. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science* 282 (5389), 740–744. Available at: <https://doi.org/10.1126/science.282.5389.740>.
- Eisenberg, D., Lüthy, R., Bowie, J.U., 1997. VERIFY3D: Assessment of protein models with three-dimensional profiles. *Methods in Enzymology* 277, 396–406. Available at: [https://doi.org/10.1016/S0076-6879\(97\)77022-8](https://doi.org/10.1016/S0076-6879(97)77022-8).
- Fernandez-Leiro, R., Scheres, S.H.W., 2016. Unravelling biological macromolecules with cryo-electron microscopy. *Nature* 537 (7620), 339–346. Available at: <https://doi.org/10.1038/nature19948>.
- Garcia, A.E., Onuchic, J.N., 2005. Folding a protein in the computer: Reality or hope? *Structure* 13 (4), 497–498. Available at: <https://doi.org/10.1016/j.str.2005.03.005>.
- Gibbs, N., Clarke, A.R., Sessions, R.B., 2001. Ab initio protein structure prediction using physicochemical potentials and a simplified off-lattice model. *Proteins: Structure, Function and Genetics* 43 (2), 186–202. Available at: [https://doi.org/10.1002/1097-0134\(20010501\)43:2<186::AID-PROT1030>3.0.CO;2-L](https://doi.org/10.1002/1097-0134(20010501)43:2<186::AID-PROT1030>3.0.CO;2-L).
- Ginalski, K., 2006. Comparative modeling for protein structure prediction. *Current Opinion in Structural Biology*. Available at: <https://doi.org/10.1016/j.sbi.2006.02.003>.
- Grishin, N.V., 2001. Fold change in evolution of protein structures. *Journal of Structural Biology* 134 (2–3), 167–185. Available at: <https://doi.org/10.1006/jsbi.2001.4335>.
- Haas, J., Roth, S., Arnold, K., *et al.*, 2013. The protein model portal – A comprehensive resource for protein structure and model information. *Database* 2013. Available at: <https://doi.org/10.1093/database/bat031>.
- Hagler, A.T., Lifson, S., 1974. Energy functions for peptides and proteins. II. The amide hydrogen bond and calculation of amide crystal properties. *Journal of the American Chemical Society* 96 (17), 5327–5335. Available at: <https://doi.org/10.1021/ja00824a005>.
- Huang, P.-S., Boyken, S.E., Baker, D., 2016. The coming of age of de novo protein design. *Nature* 537 (7620), 320–327. Available at: <https://doi.org/10.1038/nature19946>.
- Jayaram, B., Bhushan, K., Shenoy, S.R., *et al.*, 2006. Bhageerath: An energy based web enabled computer software suite for limiting the search space of tertiary structures of small globular proteins. *Nucleic Acids Research* 34 (21), 6195–6204. Available at: <https://doi.org/10.1093/nar/gkl789>.
- Jayaram, B., Dhingra, P., Lakhani, B., Shekhar, S., 2012. Bhageerath – Targeting the near impossible: Pushing the frontiers of atomic models for protein tertiary structure prediction. *Journal of Chemical Sciences* 124 (1), 83–91. Available at: <https://doi.org/10.1007/s12039-011-0189-x>.
- Jayaram, B., Dhingra, P., Mishra, A., *et al.*, 2014. Bhageerath-H: A homology/ab initio hybrid server for predicting tertiary structures of monomeric soluble proteins. *BMC Bioinformatics* 15 (Suppl. 16), S7. Available at: <https://doi.org/10.1186/1471-2105-15-S16-S7>.
- Jorgensen, W.L., Tirado-Rives, J., 1988. The OPLS potential functions for proteins. Energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society* 110 (6), 1657–1666. Available at: <https://doi.org/10.1021/ja00214a001>.
- Källberg, M., Wang, H., Wang, S., *et al.*, 2012. Template-based protein structure modeling using the RaptorX web server. *Nature Protocols* 7 (8), 1511–1522. Available at: <https://doi.org/10.1038/nprot.2012.085>.
- Kar, P., Nadler, W., Hansmann, U., 2009. Microcanonical replica exchange molecular dynamics simulation of proteins. *Physical Review E* 80 (5), 56703. Available at: <https://doi.org/10.1103/PhysRevE.80.056703>.
- Kaushik, R., Jayaram, B., 2016. Structural difficulty index: A reliable measure for modelability of protein tertiary structures. *Protein Engineering, Design and Selection* 29 (9), 391–397. Available at: <https://doi.org/10.1093/protein/gzw025>.
- Kihara, D., Lu, H., Kolinski, J., Skolnick, J., 2001. TOUCHSTONE: An ab initio protein structure prediction method that uses threading-based tertiary restraints. *Proceedings of the National Academy of Sciences of the United States of America* 98 (18), 10125–10130. Available at: <https://doi.org/10.1073/pnas.181328398>.
- Kim, D.E., Chivian, D., Baker, D., 2004. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Research* 32 (Web Server Issue), Available at: <https://doi.org/10.1093/nar/gkh468>.
- Klepeis, J.L., Floudas, C. a., 2003. ASTRO-FOLD: A combinatorial and global optimization framework for Ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. *Biophysical Journal* 85 (4), 2119–2146. Available at: [https://doi.org/10.1016/S0006-3495\(03\)74640-2](https://doi.org/10.1016/S0006-3495(03)74640-2).
- Klepeis, J.L., Wei, Y., Hecht, M.H., Floudas, C.A., 2005. Ab initio prediction of the three-dimensional structure of a de novo designed protein: A double-blind case study. *Proteins: Structure, Function and Genetics* 58 (3), 560–570. Available at: <https://doi.org/10.1002/prot.20338>.
- Koga, N., Tatsumi-Koga, R., Liu, G., *et al.*, 2012. Principles for designing ideal protein structures. *Nature* 491 (7423), 222–227. Available at: <https://doi.org/10.1038/nature11600>.
- Kolodny, R., Koehl, P., Guibas, L., Levitt, M., 2002. Small libraries of protein fragments model native protein structures accurately. *Journal of Molecular Biology* 323 (2), 297–307. Available at: [https://doi.org/10.1016/S0022-2836\(02\)00942-7](https://doi.org/10.1016/S0022-2836(02)00942-7).
- Kulik, H.J., Luehr, N., Ufimtsev, I.S., Martinez, T.J., 2012. Ab initio quantum chemistry for protein structures. *The Journal of Physical Chemistry B* 116 (41), 12501–12509. Available at: <https://doi.org/10.1021/jp307741u>.
- Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M., 1993. PROCHECK: A program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography* 26 (2), 283–291. Available at: <https://doi.org/10.1107/S0021889892009944>.
- Levitt, M., 1976. A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of Molecular Biology* 104 (1), 59–107. Available at: [https://doi.org/10.1016/0022-2836\(76\)90004-8](https://doi.org/10.1016/0022-2836(76)90004-8).
- Levitt, M., Hirshberg, M., Sharon, R., Daggett, V., 1995. Potential energy function and parameters for simulations of the molecular dynamics of proteins and nucleic acids in solution. *Computer Physics Communications* 91 (1–3), 215–231. Available at: [https://doi.org/10.1016/0010-4655\(95\)00049-L](https://doi.org/10.1016/0010-4655(95)00049-L).

- Levitt, M., Sharon, R., 1988. Accurate simulation of protein dynamics in solution. *Proceedings of the National Academy of Sciences of the United States of America* 85 (20), 7557–7561. Available at: <https://doi.org/10.1073/pnas.85.20.7557>.
- Levitt, M., Warshel, A., 1975. Computer simulation of protein folding. *Nature* 253 (5494), 694–698. Available at: <https://doi.org/10.1038/253694a0>.
- Lindorff-Larsen, K., Piana, S., Dror, R.O., Shaw, D.E., 2011. How fast-folding proteins fold. *Science* (New York, NY) 334 (6055), 517–520. Available at: <https://doi.org/10.1126/science.1208351>.
- Lindorff-Larsen, K., Trbovic, N., Maragakis, P., Piana, S., Shaw, D.E., 2012. Structure and dynamics of an unfolded protein examined by molecular dynamics simulation. *Journal of the American Chemical Society* 134 (8), 3787–3791. Available at: <https://doi.org/10.1021/ja209931w>.
- Liu, Y., Beveridge, D.L., 2002. Exploratory studies of ab initio protein structure prediction: Multiple copy simulated annealing, AMBER energy functions, and a generalized born/solvent accessibility solvation model. *Proteins* 46, 128–146.
- Liwo, A., Lee, J., Ripoll, D.R., Pillardy, J., Scheraga, H. A., 1999. Protein structure prediction by global optimization of a potential energy function. *Proceedings of the National Academy of Sciences of the United States of America* 96 (10), 5482–5485. Available at: <https://doi.org/10.1073/pnas.96.10.5482>.
- Marks, D.S., Hopf, T.A., Sander, C., 2012. Protein structure prediction from sequence variation. *Nature Biotechnology* 30 (11), 1072–1080. Available at: <https://doi.org/10.1038/nbt.2419>.
- Marrink, S.J., Risselada, H.J., Yefimov, S., Tieleman, D.P., De Vries, A.H., 2007. The MARTINI force field: Coarse grained model for biomolecular simulations. *Journal of Physical Chemistry B* 111 (27), 7812–7824. Available at: <https://doi.org/10.1021/jp071097f>.
- McCammon, J.A., Gelin, B.R., Karplus, M., 1977. Dynamics of folded proteins. *Nature* 267 (5612), 585–590. Available at: <https://doi.org/10.1038/267585a0>.
- Melo, F., Feytmans, E., 1998. Assessing protein structures with a non-local atomic interaction energy. *Journal of Molecular Biology* 277 (5), 1141–1152. Available at: <https://doi.org/10.1006/jmbi.1998.1665>.
- Mendes, C.L., Bode, B., Bauer, G.H., et al., 2014. Deploying a large petascale system: The Blue Waters experience. *Procedia Computer Science* 29, 198–209. Available at: <https://doi.org/10.1016/j.procs.2014.05.018>.
- Mishra, A., Rana, P.S., Mittal, A., Jayaram, B., 2014. D2N: Distance to the native. *BBA-Proteins and Proteomics* 1844 (10), 1798–1807. Available at: <https://doi.org/10.1016/j.bbapap.2014.07.010>.
- Mishra, A., Rao, S., Mittal, A., Jayaram, B., 2013. Capturing native/native like structures with a physico-chemical metric (pcSM) in protein folding. *BBA-Proteins and Proteomics* 1834 (8), 1520–1531. Available at: <https://doi.org/10.1016/j.bbapap.2013.04.023>.
- Mittal, A., Jayaram, B., Shenoy, S., Bawa, T.S., 2010. A stoichiometry driven universal spatial organization of backbones of folded proteins: Are there Chargaff's rules for protein folding? *Journal of Biomolecular Structure and Dynamics* 28 (2), 133–142. Available at: <https://doi.org/10.1080/07391102.2010.10507349>.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2014. Critical assessment of methods of protein structure prediction (CASP) – Round x. *Proteins: Structure, Function and Bioinformatics* 82 (SUPPL.2), 1–6. Available at: <https://doi.org/10.1002/prot.24452>.
- Narang, P., Bhushan, K., Bose, S., Jayaram, B., 2005. A computational pathway for bracketing native-like structures to small alpha helical globular proteins. *Physical Chemistry Chemical Physics* 7 (11), 2364–2375. Available at: <https://doi.org/10.1039/b502226f>.
- Narang, P., Bhushan, K., Bose, S., Jayaram, B., 2006. Protein structure evaluation using an all-atom energy based empirical scoring function. *Journal of Biomolecular Structure & Dynamics* 23 (4), 385–406. Available at: <https://doi.org/10.1080/07391102.2006.10531234>.
- Pearlman, D.A., Case, D.A., Caldwell, J.W., et al., 1995. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications* 91 (1–3), 1–41. Available at: [https://doi.org/10.1016/0010-4655\(95\)00041-D](https://doi.org/10.1016/0010-4655(95)00041-D).
- Petrey, D., Honig, B., 2005. Protein structure prediction: Inroads to biology. *Molecular Cell*. Available at: <https://doi.org/10.1016/j.molcel.2005.12.005>.
- Pokarowski, P., Kolinski, A., Skolnick, J., 2003. A minimal physically realistic protein-like lattice model: Designing an energy landscape that ensures all-or-none folding to a unique native state. *Biophysical Journal* 84 (3), 1518–1526. Available at: [https://doi.org/10.1016/S0006-3495\(03\)74964-9](https://doi.org/10.1016/S0006-3495(03)74964-9).
- Raman, S., Lange, O.F., Rossi, P., et al., 2010. NMR structure determination for larger proteins using backbone-only data. *Science* 327 (5968), 1014–1018. Available at: <https://doi.org/10.1126/science.1183649>.
- Ray, A., Lindahl, E., Wallner, B., 2012. Improved model quality assessment using ProQ2. *BMC Bioinformatics* 13 (1), 224. Available at: <https://doi.org/10.1186/1471-2105-13-224>.
- Samudrala, R., Moult, J., 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of Molecular Biology* 275 (5), 895–916. Available at: <https://doi.org/10.1006/jmbi.1997.1479>.
- Scheraga, H.A., Khalil, M., Liwo, A., 2007. Protein-folding dynamics: Overview of Molecular Simulation Techniques. *Annual Review of Physical Chemistry* 58 (1), 57–83. Available at: <https://doi.org/10.1146/annurev.physchem.58.032806.104614>.
- Shaw, D.E., Grossman, J.P., Bank, J.A., et al., 2014. Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer. In: *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, Vol. 2015–January, pp. 41–53. Available at: <https://doi.org/10.1109/SC.2014.9>.
- Shen, M., Sali, A., 2006. Statistical potential for assessment and prediction of protein structures. *Protein Science* 15 (11), 2507–2524. Available at: <https://doi.org/10.1110/ps.062416606>.
- Shenoy, S.R., Jayaram, B., 2010. Proteins: Sequence to structure and function – Current status. *Current Protein and Peptide Science* 11 (7), 498–514. Available at: <https://doi.org/10.2174/138920310794109094>.
- Shi, Y., 2014. A glimpse of structural biology through X-ray crystallography. *Cell*. Available at: <https://doi.org/10.1016/j.cell.2014.10.051>.
- Singh, A., Kaushik, R., Mishra, A., Shanker, A., Jayaram, B., 2016. ProTSAV: A protein tertiary structure analysis and validation server. *BBA-Proteins and Proteomics* 1864 (1), 11–19. Available at: <https://doi.org/10.1016/j.bbapap.2015.10.004>.
- Skolnick, J., Jaroszewski, L., Kolinski, A., Godzik, A., 1997. Derivation and testing of pair potentials for protein folding. When is the quasicheical approximation correct? *Protein Science* 6 (1997), 676–688. Available at: <https://doi.org/10.1002/pro.5560060317>.
- Skolnick, J., Zhang, Y., Arakaki, A.K., et al., 2003. TOUCHSTONE: A unified approach to protein structure prediction. *Proteins* 53 (S6), 469–479. Available at: <https://doi.org/10.1002/prot.10551>.
- Srinivasan, R., Fleming, P.J., Rose, G.D., 2004. Ab initio protein folding using LINUS. *Methods in Enzymology*. Available at: [https://doi.org/10.1016/S0076-6879\(04\)83003-9](https://doi.org/10.1016/S0076-6879(04)83003-9).
- Subramani, A., Wei, Y., Floudas, C.A., 2012. ASTRO-FOLD 2.0: An enhanced framework for protein structure prediction. *AIChE Journal* 58 (5), 1619–1637. Available at: <https://doi.org/10.1002/aic.12669>.
- Sugita, Y., Kitao, A., Okamoto, Y., 2000. Multidimensional replica-exchange method for free-energy calculations. *Journal of Chemical Physics* 113 (15), 6042–6051. Available at: <https://doi.org/10.1063/1.1308516>.
- Sugita, Y., Miyashita, N., Li, P., Yoda, T., Okamoto, Y., 2012. Recent applications of replica-exchange molecular dynamics simulations of biomolecules. *Current Physical Chemistry* 2 (4), 401–412. Available at: <https://doi.org/10.2174/1877946811202040401>.
- Sugita, Y., Okamoto, Y., 1999. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters* 314 (1–2), 141–151. Available at: [https://doi.org/10.1016/S0009-2614\(99\)01123-9](https://doi.org/10.1016/S0009-2614(99)01123-9).
- Thukral, L., Shenoy, S.R., Bhushan, K., Jayaram, B., 2007. ProRegIn: A regularity index for the selection of native-like tertiary structures of proteins. *Journal of Biosciences* 32, 71–81. Available at: <https://doi.org/10.1007/s12038-007-0007-2>.
- Van Der Spoel, D., Lindahl, E., Hess, B., et al., 2005. GROMACS: Fast, flexible, and free. *Journal of Computational Chemistry*. Available at: <https://doi.org/10.1002/jcc.20291>.

- Wiederstein, M., Sippl, M.J., 2007. ProSA-web: Interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Research* 35 (Suppl. 2), Available at: <https://doi.org/10.1093/nar/gkm290>.
- Weiner, S.J., Kollman, P.A., Case, D.A., *et al.*, 1984. A new force field for molecular mechanical simulation of nucleic acids and proteins. *Journal of American Chemical Society* 106 (17), 765–784. Available at: <https://doi.org/10.1021/ja00315a051>.
- Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function and Bioinformatics* 80 (7), 1715–1735. Available at: <https://doi.org/10.1002/prot.24065>.
- Yang, J., Yan, R., Roy, A., *et al.*, 2014. The I-TASSER Suite: Protein structure and function prediction. *Nature Methods* 12 (1), 7–8. Available at: <https://doi.org/10.1038/nmeth.3213>.
- Yang, Y., Zhou, Y., 2008. Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins: Structure, Function and Genetics* 72 (2), 793–803. Available at: <https://doi.org/10.1002/prot.21968>.
- Yan, R., Xu, D., Yang, J., Walker, S., Zhang, Y., 2013. A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Scientific Reports* 3 (1), 2619. Available at: <https://doi.org/10.1038/srep02619>.
- Zemla, A., 2003. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Research* 31 (13), 3370–3374. Available at: <https://doi.org/10.1093/nar/gkg571>.
- Zhang, Y., 2008. Progress and challenges in protein structure prediction. *Current Opinion in Structural Biology*. Available at: <https://doi.org/10.1016/j.sbi.2008.02.004>.
- Zhang, Y., Skolnick, J., 2004a. SPICKER: A clustering approach to identify near-native protein folds. *Journal of Computational Chemistry* 25 (6), 865–871. Available at: <https://doi.org/10.1002/jcc.20011>.
- Zhang, Y., Skolnick, J., 2004b. Scoring function for automated assessment of protein structure template quality. *Proteins* 57 (4), 702–710. Available at: <https://doi.org/10.1002/prot.20264>.
- Zhao, F., Peng, J., Xu, J., 2010. Fragment-free approach to protein folding using conditional neural fields. *Bioinformatics (Oxford, England)* 26 (12), i310–i317. Available at: <https://doi.org/10.1093/bioinformatics/btq193>.
- Zhou, H., Skolnick, J., 2011. GOAP: A generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophysical Journal* 101 (8), 2043–2052. Available at: <https://doi.org/10.1016/j.bpj.2011.09.012>.
- Zhou, R., 2007. Replica exchange molecular dynamics method for protein folding simulation. *Methods in Molecular Biology* 350, 205–223. Available at: [https://doi.org/10.1016/S0009-2614\(99\)01123-9](https://doi.org/10.1016/S0009-2614(99)01123-9).

Relevant Website

<http://predictioncenter.org/casp12/doc/help.html>
Protein Structure Prediction Center.

Algorithms for Structure Comparison and Analysis: Docking

Giuseppe Tradigo, University of Calabria, Rende, Italy and University of Florida, Gainesville, United States

Francesca Rondinelli, Università degli Studi di Napoli Federico II, Napoli, Italy

Gianluca Pollastri, University College Dublin, Dublin, Ireland

© 2019 Elsevier Inc. All rights reserved.

Introduction

In many scientific and technological fields there is the need to design molecules for specific goals. For instance, in Chemical Engineering many applications require to find novel materials with peculiar stress-resistance or temperature-resistance which may start from a known molecular compound, but often go towards new structures which show better fitting to the problem's specifications or constraints. In Organic Chemistry, a researcher may be interested in finding the protein responsible for a cellular process and how to accelerate the process in cases where the protein has been altered due to mutations in the genome. In Medicine contexts researchers are interested in finding new molecules or molecular approaches to cure diseases. The approach is often searching in a database of drugs to find which molecule is the best ligand for a target protein and has certain toxicity levels compatible with the treatment being designed.

Two types of molecules are usually involved in the docking process: (i) The ligand and (ii) the target molecule. Ligand, which comes from the latin term "ligare", refers to the property of the molecule to bind to another molecule. In modern chemistry, a ligand indicates a molecule that interacts with another molecule through noncovalent forces (Krumrine *et al.*, 2005). Hence, the interaction does not involve the formation of chemical bonds, which could lead to relevant chemical changes in both the ligand and the target, which is often more complex and larger in size. The final compound may be a supramolecular complex, containing multiple ligand and the target aggregates. The main forces involved in the process mainly depend on the shape of the two molecules and the influence of the solvent or the environment. In fact, the shape can be modified according to external forces (e.g., chemical bonds, solvent effect, other chemical species concentrations). These forces are usually studied using quantum mechanics. However, the direct application of quantum physics laws for such huge biological molecular systems remains limited due to computational resources limitations.

Due to the structural complexity of large molecules there is the need of smart ways to search the phase space in order to find candidate structures with minimal energy for the target-ligand molecular complex. Complications in this search can arise when dealing with metamorphic proteins, which have been observed folded in a different 3D shapes, depending on the cellular environment (Murzin, 2008). Furthermore, most computational approaches introduce significant simplifications which usually leads to a lack of generality of both the obtained model and the results.

For these reasons, the problem of finding an overall 3D structure for a ligand-protein or for a protein-protein complex is much more difficult than experimentally determining their individual 3D structures. Hence, computational techniques able to predict the interaction among proteins and between proteins and ligands are of utter importance because of the growing number of known protein structures (Vakser, 2014). Simplified computational approaches could consider the molecular shape of one or both of the chemical species as an invariant (or at least varied in a controlled way), which, albeit be a strong constraint, it helps to cut the search space for a solution with minimal energy but has to be carefully considered when adopting the resulting molecule complexes in critical applications. In fact, these results may need to be further modified to have the required features (e.g., solubility, toxicity) before being considered as viable clinical candidates.

There is a growing number of docking developers working in a wide community and producing algorithms which need assessments of methodologies. The development of more powerful docking algorithms and models can exploit the opportunity of having growing information and data resources, larger computational capabilities, and better understanding of protein interactions.

Background/Fundamentals

Structural bioinformatics is a research field which offers tools for the discovery, design and optimization of molecules (Krumrine *et al.*, 2005). However, no method exists being able to give a general solution for the many problems involved in the design of new materials.

Thousands of proteins carry out their intra- and extra-cellular functions interacting with each other (Khan *et al.*, 2013a). This observation leads to a relatively new research field called PPI, for protein-protein interaction, which models interacting protein molecules as nodes of a graph, where the arcs represent interactions between them. PPI is quite far from more classical chemical- and physical-based approaches, being a series of techniques and algorithms that map the problem of interacting molecules into the computer science problem. This allows for the efficient labeling of proteins with predicted functions inducted by similar neighbour nodes in the PPI graph.

Interactions among proteins play a fundamental role in almost every biological event such as: (i) Protein signaling, (ii) trafficking and their signal degradation (Castro *et al.*, 2005; Fuchs *et al.*, 2004), (iii) DNA repairing, replication and gene expression

(Neduva and Russell, 2006; Petsalaki and Russell, 2008). All of these cellular events require interactions between protein interfaces to function. The complexity of such iterations in the cell is huge and having complete map of them all will help in understanding how the regulatory networks work and the behaviour of the overall biological system.

Computational docking is widely used for the theoretical prediction of small molecule ligand-protein complex and is usually composed of two steps: (i) Generation of alternate shapes of a ligand molecule in order to model and simulate possible interactions with the target binding site and (ii) an energy or scoring function used to rank the various candidate structures generated at the previous step. In general, docking programs are not optimized for peptide docking, being designed and optimized to work with small molecules. Furthermore they can be deceived by the flexibility and alternative conformations of peptides which tend to rotate within the search space of the receptor site (Khan *et al.*, 2013a). For these reasons, many authors (Aloy and Russell, 2006; Gray, 2006; Russell *et al.*, 2004) recommend caution in adopting current protein-docking algorithms to detect interacting protein-ligands. Recently however, two high-throughput docking (HTD) experiments have been reported in literature, that demonstrate the ability of general docking methods in detecting protein–ligand interactions. Mosca *et al.* (2009) used docking to identify pairs of proteins interacting with each other in the *Saccharomyces cerevisiae* interactome. Furthermore, Wass *et al.* (2011) successfully distinguished between interacting native and non-interacting non-native proteins.

Computational Approaches

Optimization problems in chemistry vary from selecting the best wavelength for optimal spectroscopic concentration predictions to geometry optimization of atomic clusters and protein folding. While most optimization procedures maintain the ability to locate global optima for simple problems, few are effective against local optima convergence with regard to difficult or large scale optimization problems. Simulated annealing (SA) has shown a great tolerance to local optima convergence and it is often considered a global optimizer. The optimization algorithm has found wide use in numerous areas such as engineering, chemistry, biology, physics, and drug design. Recently, integrated approach of SA and DFT (Density Functional Theory) assured important results in developing novel drugs for diseases in different therapeutic areas (Persico *et al.*, 2017).

Car-Parrinello method represents a powerful tool of simulation in chemical engineering and biology applications. This popular approach is in fact an effective way to study phase diagrams and to identify new phases of materials, particularly at high pressure conditions. Besides, the elucidation of biological systems reactivity is extremely proficuous. It is well known that biological structures are very large and often surrounded by solvents contributing to the energy of the whole system. Taking in account the entire biomolecule would be highly time consuming. Nevertheless, a reduced model made up of atoms interested in chemical or conformational process can give a reliable insight in the system features and conversion.

Target-based drug design usually start with the 3d structure of a receptor site or the active site in a protein molecule (Krumrine *et al.*, 2005). Analog-based designs, such as pharmacophores and QSAR (quantitative structure-activity relationship), use the laws of mechanics describe the atomic level interactions and calculate an optimal molecular shape with a minimal energy. Structure-based design involves several steps: (i) 3d structure determinations, which usually involves finding it in online databases or predicting it with a prediction software tool, (ii) site representation and identification, which entails algorithms to automatically detect the binding site(s) on the external surface of the molecule (it is often the largest cavity) and its 3D shape, (iii) ligand docking, which, in case the ligand is not given, can search online databases of small molecules for ligand candidates matching the binding site; this phase could even generate novel chemical structures (de novo design) which however have to deal with the synthetic feasibility of the generated compound (iv) scoring, during which the strength of the interaction between the ligand and the binding site is evaluated.

The scoring function is a crucial element for docking. As for all the other aspects of systems containing chemical elements, a principled approach dealing with the calculation of the binding free energy between the ligand and the target protein would be computationally infeasible and extremely time consuming. The need for methods that can deal with HTD pipelines which can process a large number of (potentially large) molecules, has led to the implementation of approximate algorithmic approaches. First principle methods use a mechanics force field which represents the forces occurring between atoms and also weaker forces (e.g., Wand der Waals forces), without taking into account entropy. Nevertheless, such an approach is quite time consuming. Semiempirical approaches use a term from a linearized function to approximate absolute binding free energy, and a term from known data. Even if less time-consuming than first principle methods, these approaches are still quite time consuming. Empirically derived scoring functions are designed to be very fast in scoring ligands. Structural descriptors are selected and assigned weights calculated through regression methods using statistical models. In these approaches, the atomic details of the ligand and the binding sites are lost. Furthermore it may be difficult to find training data for particular binding sites and ligands. Knowledge-based potentials are methods where potential are not derived from experimental binding data, but from statistical analysis of atom binding frequencies measured in experimentally resolved protein-ligand complexes.

Analysis and Assessment

CAPRI (Critical Assessment of PRediction of Interactions) is an international experiment in which state of the art protein-docking methods to predict protein-protein interactions are assessed (Janin *et al.*, 2003). Many research groups participate worldwide and

submit structure predictions in a blind process for a set of protein-protein complexes based on the known structure of the component proteins. Predictions are compared to the unpublished structures of the complexes determined with X-ray crystallography at the end of the experiment. The analysis of the first editions of the competition lead to the observation that new scoring functions and for methods handling the conformation changes were needed, hence the state of the art was still not sufficient to deal with real-world molecular complexes.

In general, docking is a two phases process in which there is a search and a scoring. After the scoring has been performed, candidate complexes has to be assessed and tested to work correctly. One approach is the comparison of the RMSD (Root-Mean-Square Deviation) between the observed and the calculated complex structures. Nevertheless, RMSD can fail to detect conformational changes, not being invariant with respect to rotation and translation of the center-of-mass (Lee and Levitt, 1997). However it is widely adopted to test the scoring function results, with reported top scoring configurations being within 2 Å in 45% (Ewing *et al.*, 2001; Diller and Merz, 2001), 65% (Jones *et al.*, 1997) and 73% (Rarey *et al.*, 1996) of the cases.

Case Studies

In Søndergaard *et al.* (2009), the authors analyze the prevalence of crystal-induced artifacts and water-mediated contacts in protein-ligand complexes showing the effect they have on the performance of the scoring functions. They report that 36% of ligands in the PDBBind 2007 data set are influenced by crystal contacts and that the performance of a scoring function is affected by these.

In Khan *et al.* (2013b), the authors investigate if docking can be used to identify protein-peptide interactions with the objective of evaluating if docking could distinguish a peptide binding region from adjacent non-binding regions. They evaluated the performance of AutoDock Vina (Trott and Olson, 2010), training a bidirectional recurrent neural network using as input the peptide sequence, predicted secondary structure, Vina docking score and Pepsite score. They conclude that docking has only modest power to define the location of a peptide within a larger protein region. However, this information can be used in training machine learning methods which may allow for the identification of peptide binding regions within a protein sequence.

See also: Algorithms for Strings and Sequences: Searching Motifs. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Biomolecular Structures: Prediction, Identification and Analyses. Computational Protein Engineering Approaches for Effective Design of New Molecules. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Investigating Metabolic Pathways and Networks. Protein Structural Bioinformatics: An Overview. Small Molecule Drug Design. Structural Genomics. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow. Vaccine Target Discovery

References

- Aloy, P., Russell, R.B., 2006. Structural systems biology: Modelling protein interactions. *Nature Reviews Molecular Cell Biology* 7 (3), 188.
- Castro, A., Bernis, C., Vigneron, S., Labbe, J.C., Lorca, T., 2005. The anaphase-promoting complex: A key factor in the regulation of cell cycle. *Oncogene* 24 (3), 314.
- Diller, D.J., Merz, K.M., 2001. High throughput docking for library design and library prioritization. *Proteins: Structure, Function, and Bioinformatics* 43 (2), 113–124.
- Ewing, T.J., Makino, S., Skillman, A.G., Kuntz, I.D., 2001. DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases. *Journal of Computer-Aided Molecular Design* 15 (5), 411–428.
- Fuchs, S.Y., Spiegelman, V.S., Kumar, K.S., 2004. The many faces of β -TrCP E3 ubiquitin ligases: Reflections in the magic mirror of cancer. *Oncogene* 23 (11), 2028.
- Gray, J.J., 2006. High-resolution protein–protein docking. *Current Opinion in Structural Biology* 16 (2), 183–193.
- Janin, J., Henrick, K., Moult, J., *et al.*, 2003. CAPRI: A critical assessment of predicted interactions. *Proteins: Structure, Function, and Bioinformatics* 52 (1), 2–9.
- Jones, G., Willett, P., Glen, R.C., Leach, A.R., Taylor, R., 1997. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology* 267 (3), 727–748.
- Khan, W., Duffy, F., Pollastri, G., Shields, D.C., Mooney, C., 2013a. Potential utility of docking to identify protein-peptide binding regions. Technical Report UCDCSI-2013–01, University College Dublin.
- Khan, W., Duffy, F., Pollastri, G., Shields, D.C., Mooney, C., 2013b. Predicting binding within disordered protein regions to structurally characterised peptide-binding domains. *PLOS ONE* 8 (9), e72838.
- Krumrine, J., Raubacher, F., Brooijmans, N., Kuntz, I., 2005. Principles and methods of docking and ligand design. *Structural Bioinformatics* 44, 441–476.
- Lee, C., Levitt, M., 1997. Packing as a structural basis of protein stability: Understanding mutant properties from wildtype structure. *Pacific Symposium on Biocomputing*, 245–255.
- Mosca, R., Pons, C., Fernández-Reco, J., Aloy, P., 2009. Pushing structural information into the yeast interactome by high-throughput protein docking experiments. *PLOS Computational Biology* 5 (8), e1000490.
- Murzin, A.G., 2008. Metamorphic proteins. *Science* 320 (5884), 1725–1726.
- Nedeva, V., Russell, R.B., 2006. Peptides mediating interaction networks: New leads at last. *Current Opinion in Biotechnology* 17 (5), 465–471.
- Persico, M., Fattorusso, R., Tagliatala-Scafati, O., *et al.*, 2017. The interaction of heme with plakortin and a synthetic endoperoxide analogue: New insights into the heme-activated antimalarial mechanism. *Nature Scientific Reports* 7, 45485.
- Petsalaki, E., Russell, R.B., 2008. Peptide-mediated interactions in biological systems: New discoveries and applications. *Current Opinion in Biotechnology* 19 (4), 344–350.

- Rarey, M., Wefing, S., Lengauer, T., 1996. Placement of medium-sized molecular fragments into active sites of proteins. *Journal of Computer-Aided Molecular Design* 10 (1), 41–54.
- Russell, R.B., Alber, F., Aloy, P., *et al.*, 2004. A structural perspective on protein–protein interactions. *Current Opinion in Structural Biology* 14 (3), 313–324.
- Søndergaard, C.R., Garrett, A.E., Carstensen, T., Pollastri, G., Nielsen, J.E., 2009. Structural artifacts in protein – Ligand X-ray structures: Implications for the development of docking scoring functions. *Journal of Medicinal Chemistry* 52 (18), 5673–5684.
- Trott, O., Olson, A.J., 2010. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry* 31 (2), 455–461.
- Vakser, I.A., 2014. Protein–protein docking: From interaction to interactome. *Biophysical Journal* 107 (8), 1785–1793.
- Wass, M.N., Fuentes, G., Pons, C., Pazos, F., Valencia, A., 2011. Towards the prediction of protein interaction partners using physical docking. *Molecular Systems Biology* 7 (1), 469.

Further Reading

- Cannataro, M., Guzzi, P.H., 2012. *Data Management of Protein Interaction Networks*. vol. 17. John Wiley & Sons.
- Structural Bioinformatics*. Gu, J., Bourne, P.E. (Eds.), vol. 44. John Wiley & Sons.
- Adaption of Simulated Annealing to Chemical Optimization Problems*. Kalivas, J.H. (Ed.), vol. 15. Elsevier.
- Wei, J., Denn, M.M., Seinfeld, J.H., *et al.*, 2001. *Molecular Modeling and Theory in Chemical Engineering*. vol. 28. Academic Press.

Biographical Sketch

Giuseppe Tradigo is a postdoc at the DIMES Department of Computer Science, Models, Electronics and Systems Engineering, University of Calabria, Italy. He has been a Research Fellow at University of Florida, Epidemiology Department, US, where he worked on a GWAS (Genome-Wide Association Study) project on the integration of complete genomic information with phenotypical data from a large patients dataset. He has also been a visiting research student at the AmMBio Laboratory, University College Dublin, where he participated to the international CASP competition with a set of servers for protein structure prediction. He obtained his PhD in Biomedical and Computer Science Engineering at University of Catanzaro, Italy. His main research interests are big data and cloud models for health and clinical applications, genomic and proteomic structure prediction, data extraction and classification from biomedical data.

Francesca Rondinelli is a young researcher. She obtained her PhD in Theoretical Chemistry at University of Calabria, Dept. of Chemistry. She has been visiting research student at KTH Royal Institute of Technology in Stockholm, Department of Chemistry. She has been a postdoc both at University of Calabria and at University of Naples, Federico II. Her research interest go from cyclodextrins, principled drug design and CO₂ activation.

Gianluca Pollastri is an Associate Professor in the School of Computer Science and a principal investigator at the Institute for Discovery and at the Institute for Bioinformatics at University College Dublin. He was awarded his M.Sc. in Telecommunication Engineering by the University of Florence, Italy, in 1999 and his PhD in Computer Science by University of California at Irvine in 2003. He works on machine learning and deep learning models for structured data, which he has applied to a cohort of problems in the bioinformatics and chemoinformatics space. He has developed some of the most accurate servers for the prediction of functional and structural features of proteins, which have processed over a million queries from all over the world and have been licensed to 75 different subjects, including pharmaceutical companies. His laboratory at UCD has been funded by Science Foundation Ireland, the Health Research Board, the Irish Research Council, Microsoft, UCD, the King Abdulaziz City for Science and Technology (Saudi Arabia) and Nvidia.

Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors

Lo Giudice Paolo and Domenico Ursino, University "Mediterranea" of Reggio Calabria, Reggio Calabria, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Network Analysis (hereafter, NA) has been a multidisciplinary research field from its origin. Relationship represents the key concept in a network. Indeed, relationships, and not participants, give the main contribution to model a network (Hanneman and Riddle, 2005; Tsvetovat and Kouznetsov, 2011).

NA allows the identification of relationship patterns existing in a network. Moreover, it allows the detection, and the next investigation, of the information (and/or other resource) flow among participants. Finally, it focuses on interactions among participants, which differentiates it from other kinds of analysis that mainly investigate the features of a single participant.

Network analysis-based approaches allow interactions in a group to be mapped, as well as the connectivity of a network to be visualized and investigated. Furthermore, they make it possible to quantify the processes taking place among network participants (Knoke and Yang, 2008; Scott, 2012; Wasserman and Galaskiewicz, 1994).

Two research fields where the employment of NA is rapidly increasing are bioinformatics and biomedicine. Think, for instance, of Public Health (Luke and Harris, 2007; Berkman and Glass, 2000; House *et al.*, 1988). According to Luke and Harris (2007), it is possible to find three main typologies of network in this sector, namely: (i) transmission networks, (ii) social networks, and (iii) organizational networks. Transmission networks are particularly relevant and, therefore, largely investigated. They allow the analysis of the diffusion of both diseases (Friedman and Aral, 2001; Friedman *et al.*, 1997; Jolly *et al.*, 2001; Aral, 1999; Valente and Fosados, 2006) and medical information (Valente and Fosados, 2006; Katz and Lazarsfeld, 1955; Valente, 1995; Guardiola *et al.*, 2002; Valente and Davis, 1999). Two very relevant application contexts for transmission networks are social epidemiology (Berkman *et al.*, 2014; Haustein *et al.*, 2014) and information diffusion on social networks (Eysenbach, 2008; Scanfeld *et al.*, 2010; Hawn, 2009; Laranjo and Arguel, 2014; Xu *et al.*, 2015). Social networks investigate how social structures and relationships influence both public health and people behavior (Kessler *et al.*, 1985; Berkman, 1984; Cassel, 1976; Kaplan *et al.*, 1977; Lin *et al.*, 1999). Organizational networks represent the most recent research sector; in this case, researchers evaluate the impact of associations and/or agencies on public health (Leischow and Milstein, 2006; Borgatti and Foster, 2003; Becker *et al.*, 1998; Mueller *et al.*, 2004; Kapucu, 2005).

In bioinformatics, an important investigation regards the usage of "information-based" tools to analyze medical problems. In this case, two very important research areas are molecular analysis (Wu *et al.*, 2009; Cusick *et al.*, 2009; Gandhi *et al.*, 2006; Han, 2008; Sevimoglu and Arga, 2014) and brain analysis (Rubinov and Sporns, 2010; Greicius *et al.*, 2003; Achard *et al.*, 2006; Supekar *et al.*, 2008; Zalesky *et al.*, 2010). Another relevant investigation concerns the definition of software packages and analytic tools allowing extensive studies on large datasets (Huang *et al.*, 2009; Librado and Rozas, 2009; Zhang and Horvath, 2005; Langfelder and Horvath, 2008; Kears *et al.*, 2012; Chen *et al.*, 2009).

In this analysis, two of the most used indexes are: (i) centrality indicators (adopted, for instance, in Yoon *et al.* (2006) and Junker (2006)), and (ii) connection indicators (employed, for instance, in Girvan and Newman (2002), Estrada (2010), Wu *et al.* (2011)). An overview on the usage of these indicators can be found in Ghasemi *et al.* (2014). For instance, Closeness Centrality is used in Hahn and Kern (2005) to study the evolution of protein-protein networks. In Ozgur *et al.* (2008), the authors use eigenvector centrality to predict good candidate disease-related genes. In del Rio *et al.* (2009), the authors adopt 16 different centrality measures to analyze 18 metabolic networks. Finally, in Hsu *et al.* (2008), the authors employ both centrality and cohesion indexes for understanding how miRNAs influence the protein interaction network.

This article is organized as follows. In Section "Network Representation", we describe how networks can be represented. In Section "Index Description", we illustrate the main indexes employed in network analysis. Finally, in Section "Closing Remarks", we draw our conclusions.

Network Representation

A network $\mathcal{N} = \langle V, E \rangle$ consists of a set V of nodes and a set E of edges. Each edge $e_{ij} = (v_i, v_j)$ connects the nodes v_i and v_j . Edges can be either directed (when they can be traversed only in one direction) or undirected (when they can be traversed in both directions). Furthermore, networks can be weighted or unweighted. If a network is weighted, it can be represented as (v_i, v_j, w_{ij}) , where w_{ij} denotes the weight of the corresponding edge. On the basis of the reference context, this weight could represent strength, distance, similarity, etc.

Example 2.1: Consider the networks in Fig. 1. The one on the left is undirected and unweighted, whereas the one on the right is directed and weighted. For instance, the edge A–C in the network on the right indicates that there is a link from A to C and that the weight of this link is 34. □

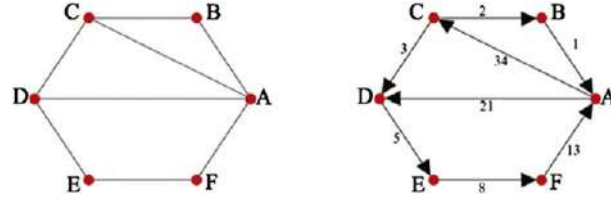


Fig. 1 An example of undirected and unweighted network (on the left), and an example of directed and weighted network (on the right).

	A	B	C	D	E	F
A	0	1	1	1	0	1
B	1	0	1	0	0	0
C	1	1	0	1	0	0
D	1	0	1	0	1	0
E	0	0	0	1	0	1
F	1	0	0	0	1	0

	A	B	C	D	E	F
A	0	0	34	21	0	0
B	1	0	1	0	0	0
C	0	2	0	3	0	0
D	0	0	0	0	5	0
E	0	0	0	0	0	8
F	13	0	0	0	0	0

Fig. 2 The Adjacency Matrixes corresponding to the networks of Fig. 1.

Table 1 The edge lists corresponding to the networks of Fig. 1

Undirected and unweighted network	Directed and weighted network
(A,B), (A,C), (A,D), (A,E), (B,C), (C,D), (D,E), (E,F)	(A,C, 34), (A,D,21), (B,A,1), (C,B,2), (C,D,3), (D,E,5), (E,F,8), (F,A,13)

Table 2 The adjacency lists corresponding to the networks of Fig. 1

Undirected and unweighted network	Directed and weighted network
A – {B, C, D, E}	A – {(C,34), (D,21)}
B – {C}	B – {(A,1)}
C – {D}	C – {(B,2), (D,3)}
D – {E}	D – {(E,5)}
E – {F}	E – {(F,8)}
F – {}	F – {(A,13)}

The basic way to represent a network $\mathcal{N}$ employs the so called adjacency matrix $\mathcal{A}$. This is a $|V| \times |V|$ matrix. Each row and each column correspond to a node. If $\mathcal{N}$ is unweighted, the generic element $\mathcal{A}[i, j]$ is set to 1 if there exists an edge from v_i to v_j ; otherwise, it is set to 0. By contrast, if $\mathcal{N}$ is weighted, $\mathcal{A}[i, j]$ is set to the weight of the edge from v_i to v_j . Finally, if $\mathcal{N}$ is undirected, the corresponding adjacency matrix is a lower triangular one. The adjacency matrix is very easy to be understood; however, in real cases, it is very sparse (i.e., most of its elements are set equal to 0) and, therefore, it wastes a lot of space.

To reduce the waste of space, $\mathcal{N}$ can be represented as an edge list $\mathcal{L}$. In this case, if $\mathcal{N}$ is unweighted, $\mathcal{L}$ consists of a list of pairs, each representing an edge with its starting and ending nodes. By contrast, if $\mathcal{N}$ is weighted, $\mathcal{L}$ consists of a list of triplets, each representing an edge with the corresponding starting node, ending node and weight. Clearly, edge list is more compact, but less clear, than adjacency matrix.

A further reduction of the space needed to represent $\mathcal{N}$ is obtained by adopting an adjacency list $\mathcal{L}^*$. If $\mathcal{N}$ is unweighted, $\mathcal{L}^*$ consists of a list of pairs $\langle v_i, \mathcal{L}'_i \rangle$, where v_i is a node of $\mathcal{N}$ and $\mathcal{L}'_i$ is the list of the nodes reachable from it. If $\mathcal{N}$ is weighted, $\mathcal{L}^*$ consists of a list of pairs $\langle v_i, \mathcal{L}''_i \rangle$, where v_i is a node of $\mathcal{N}$ and $\mathcal{L}''_i$ is, in turn, a list of pairs $\langle v_j, w_{ij} \rangle$, such that v_j is reachable from v_i and w_{ij} is the weight of the corresponding edge. Clearly, among the three structures presented above, the adjacency list is the most compact, but also the least clear one.

Example 2.2: (...cnt'd)

Consider the networks shown in Fig. 1. The corresponding adjacency matrixes are reported in Fig. 2. The associated edge lists are shown in Table 1. Finally, the corresponding adjacency lists are illustrated in Table 2. $\square$

Index Description

Basic Indexes

The most basic, yet extremely important, index for a network $\mathcal{N}$ is its *size*. It consists of the number of its nodes.

Given a node v_i of $\mathcal{N}$, if this last is *undirected*, the number of edges connecting v_i to the other nodes of $\mathcal{N}$ represents the *degree* of v_i . If $\mathcal{N}$ is *directed*, the number of edges incoming to (resp., outgoing from) v_i represents the *indegree* (resp., the *outdegree*) of v_i .

If $\mathcal{N}$ is *undirected*, it is possible to consider the *mean degree*, i.e., the ratio of the sum of the degrees of its nodes to the number of nodes. If $\mathcal{N}$ is *directed*, it is possible to define the *mean indegree* and the *mean outdegree* of $\mathcal{N}$ in an analogous fashion.

If $\mathcal{N}$ is unweighted, its *density* is simply the ratio of its edges to the number of all its possible edges. Recall that the number of all possible edges of $\mathcal{N}$ is $\frac{|V| \cdot (|V| - 1)}{2}$, if $\mathcal{N}$ is undirected, whereas it is $|V| \cdot (|V| - 1)$, if $\mathcal{N}$ is directed. If $\mathcal{N}$ is weighted, its *density* can be defined as the ratio of the sum of the weights of the existing edges to the number of all its possible edges.

Example 3.1: (...cnt'd)

Consider the undirected network of Fig. 1. Its size is 6. The degree of the node A of this network is 4. The mean degree of the network is 2.6, whereas its density is 0.53.

Consider, now, the directed network of the same figure. Its size is 6. The indegree of the node D is 2, whereas its outdegree is 1. The mean indegree (resp., outdegree) of the network is 1.33 (resp., 1.33). Finally, its density is 2.90. □

Given a network $\mathcal{N}$, a *walk* of $\mathcal{N}$ consists of an alternating sequence of nodes and edges that begins and ends with a node. If the starting and the ending nodes of a walk are different, it is said *open*; otherwise, it is said *close*. If no node is crossed twice, a walk is said *simple*.

Given a network $\mathcal{N}$, a *path* is an open simple walk. A *cycle* is a closed simple walk. A *trail* is a walk that includes each edge no more than once. A *tour* is a closed walk comprising each edge of $\mathcal{N}$ at least once.

If $\mathcal{N}$ is unweighted, the *length* of a walk of $\mathcal{N}$ consists of the number of its edges. If $\mathcal{N}$ is weighted, the length of a walk of $\mathcal{N}$ is the sum of the weights of its edges. Given two nodes v_i and v_j of $\mathcal{N}$, their *geodesic distance* is the length of the shortest path from v_i to v_j . Given a node v_i , the *eccentricity* of v_i is the maximum geodesic distance between v_i and any other node of $\mathcal{N}$. Finally, the *radius* of $\mathcal{N}$ is the minimum eccentricity over all its nodes, whereas, if $\mathcal{N}$ is connected, the *diameter* of $\mathcal{N}$ is the maximum eccentricity over all its nodes.

Example 3.2: (...cnt'd)

Consider the directed weighted network of Fig. 1. An example of walk is the one linking nodes A–D–E–F–A–C. This is an open walk. Vice versa, an example of close walk is B–A–C–B. Since no node is crossed twice, this walk is simple.

The walk A–C–D is an example of path, whereas the walk A–C–D–E–F–A is an example of cycle. The walk A–C–D–E is an example of trail. Finally, the walk A–C–B–A–C–D–E–F–A–D–E–F–A is an example of tour. The length of the walk A–D–E–F is 34. Consider, now, nodes C and A. Their shortest path is C–B–A. The distance of this shortest path is 3 and represents the geodesic distance from C to A. The eccentricity of the node C is 16. Finally, the diameter of this network is 62. □

Centrality Indexes

Centrality indexes aim at measuring power, influence, or other similar features, for the nodes of a network. In real life, there is an agreement on the fact that power and influence are strictly related to relationships. By contrast, there is much less agreement about what power and influence mean. Therefore, several centrality indexes have been proposed to capture the different meanings associated with the term “power”.

Degree centrality

In a network, nodes having more edges to other nodes may have an advantage. In fact, they may have alternative ways to communicate, and hence are less dependent on other nodes. Furthermore, they may have access to more resources of the network as a whole. Finally, because they have many edges, they are often third-parties in exchanges among others, and can benefit from this brokerage. As a consequence of all these observations, a very simple, but often very effective, centrality index is node degree. The corresponding form of centrality is called *degree centrality*.

In an undirected network, the degree centrality of a node is exactly the number of its edges. In a directed network, instead, it is important to distinguish centrality based on indegree from centrality based on outdegree. If a node has a high indegree, then many nodes direct edges to it; as a consequence, a high indegree implies *prominence* or *prestige*. If a node has a high outdegree, then it is able to exchange with many other nodes; as a consequence, a high outdegree implies *influence*.

Generally, in a network, degree centrality follows a power-law distribution. This implies that there are few nodes with a high degree centrality and many nodes with a low degree centrality.

Closeness centrality

A weak point of degree centrality is that it considers only the immediate edges of a node or the edges of the neighbors of a node, rather than indirect edges to all the other nodes. Actually, a node might be linked to a high number of other nodes, but these last

ones might be rather disconnected from the network as a whole. If this happens, the node could be quite central, but only in a local neighborhood.

Closeness centrality emphasizes the distance (or, better, the closeness) of a node to all the other nodes in the network. Depending on the meaning we want to assign to the term “close”, a number of slightly different closeness measures can be defined.

In order to compute the closeness centrality of the nodes of a network, first the length of the shortest path between every pair of nodes must be computed. Then, for each node: (i) the average distance to all the other nodes is computed; (ii) this distance is divided by the maximum distance; (iii) the obtained value is subtracted from 1. The result is a number between 0 and 1; the higher this number, the higher the closeness and, consequently, the lower the distance.

As for the distribution of the values of closeness centrality in a network, generally few nodes form a long tail on the right but all the other nodes form a bell curve residing at the low end of the spectrum.

Betweenness centrality

Betweenness centrality starts from the assumption that a node v_i of a network $\mathcal{N}$ can gain power if it presides over a communication bottleneck. The more nodes of $\mathcal{N}$ depend on v_i to make connections with other nodes, the more power v_i has. On the other side, if two nodes are connected by more than one shortest path and v_i is not on all of them, it loses some power. The betweenness centrality of v_i considers exactly the proportion of times v_i is in the shortest path between other nodes; the higher this proportion the higher betweenness centrality.

Betweenness centrality also allows the identification of boundary spanners, i.e., nodes acting as bridges between two or more subnetworks that, otherwise, would not be able to communicate to each other. Finally, betweenness centrality also measures the “stress” which v_i must undergo during the activities of $\mathcal{N}$.

Betweenness centrality can be measured as follows: first, for each pair of nodes of $\mathcal{N}$, the shortest path is computed. Then, for each node v_i of $\mathcal{N}$, the number of the shortest paths, which v_i is involved on, is computed. Finally, if necessary, the obtained results can be normalized to the range [0,1].

Eigenvector centrality

Eigenvector centrality starts by the assumption that, in order to evaluate the centrality of a node v_i in a network $\mathcal{N}$, instead of simply adding the number of edges to compute degree, one should weight each of the edges by the degree of the node at the other end of the link (i.e., well connected nodes are worth more than badly connected ones). In this case, v_i is central if it is connected to other nodes that, in turn, are central. A node with a high eigenvector centrality is connected to many nodes that are themselves connected to many nodes.

Eigenvector centrality allows the identification of the so called “gray cardinals”, i.e., nodes representing, for instance, advisors or decision makers operating secretly and unofficially. For instance, Don Corleone was a “gray cardinal” because he had an immense power, since he surrounded himself with sons and his trusted “capos”, who handled his affairs. By knowing well connected people, “gray cardinals” can use these relationships to reach their objectives while staying largely in the shadow.

The eigenvector centrality of v_i can be computed as follows: (i) a centrality score of 1 is assigned to all nodes; (ii) the score of each node is computed as a weighted sum of the centralities of all the nodes of its neighborhood; (iii) obtained scores are normalized by dividing them by the largest score; (iv) steps (ii) and (iii) are repeated until the node scores stop changing.

PageRank

PageRank overcomes the idea of centrality. In fact, instead of outgoing edges, PageRank centrality is determined through incoming edges. PageRank was originally developed for indexing web pages. In fact, it is the algorithm used by Google for this purpose. However, it can be applied to all directed networks.

PageRank follows the same ideas of eigenvector centrality, i.e., the PageRank of v_i depends on the number of edges incoming to it, weighted by the PageRank of the nodes at the other end of the edge.

Analogously to what happens for the computation of the eigenvector centrality, the computation of PageRank is iterative. However, differently from Eigenvector Centrality, PageRank computation is local in nature, because only immediate neighbors are taken into consideration; however, its iterative nature allows global influence to propagate through the network, although much more attenuated than eigenvector centrality. As a consequence of its local nature, the computation of PageRank scales much better to very large networks. Furthermore, at any time, it returns a result, but if more iterations are performed, the quality of results improves a lot.

Cohesion Indexes

One of the main issues in NA is the identification of cohesive subgroups of actors within a network. Cohesive subgroups are subsets of actors linked by strong, direct, intense, frequent and/or positive relationships. Cohesion indexes aim at supporting the identification of cohesive subnetworks in a network.

To introduce cohesion indexes, we must start with the concept of subnetwork. A *subnetwork* consists of a subset of nodes of a network and of all the edges linking them.

An *ego-network* is a subnetwork consisting of a set of nodes, called “alters”, connected to a focal node, called “ego”, along with the relationships between the ego and the alters and any relationships among the alters. These networks are important because the analysis of their structure provides information useful to understand and predict the behavior of ego.

Triads

A *triad* is a triple of nodes and of the possible edges existing among them.

With *undirected networks*, there are four possible kinds of relationship among three nodes (see Fig. 3), i.e., no edges, one edge, two edges or three edges. Triad census aims at determining the distribution of these four kinds of relationship across all the possible triads. It can give a good approximation of how much a population is characterized by “isolation”, “couples only”, “structural holes” or “closed triads”. In this context, “structural holes” represent a very interesting concept. Given three nodes v_i , v_j and v_k , there exists a structural hole if v_i is connected to v_j , v_j is connected to v_k but v_i is not connected to v_k . Structural holes have important implications in a network; in fact, they are nodes capable of using and handling asymmetric information; furthermore, they can also bridge two communities. The *ratio between structural holes and closed triads* is a very important index, because, if it is high, then the corresponding network tends to be hierarchic, whereas, if it is low, then the corresponding network tends to be egalitarian.

With *directed network*, there are 16 possible kinds of relationship among three nodes (see Fig. 4), including those exhibiting hierarchy, equality, the formation of exclusive groups or clusters. A very important scenario is that of *transitive triads* (i.e., relationships where, if there are edges from v_i to v_j and from v_j to v_k , then there is also an edge from v_i to v_k). Such triads represent the “equilibrium” toward which triadic relationships tend.

Cliques

In its most general definition, a clique is a subnetwork of a network $\mathcal{N}$ in which its nodes are more closely and intensely linked to each other than they are the other nodes of $\mathcal{N}$. In its most formal and rigorous form, a *clique* is defined as a maximal totally connected subnetwork of a given network. The smallest clique is the dyad, consisting of two nodes linked by an edge. Dyads can be extended to become more and more inclusive in such a way as to form strong or closely connected regions in graphs. A clique consists of several overlapping closed triads and, as such, it inherits many of the properties of closed triads.

This rigorous definition of a clique may be too strong in several situations. Indeed, in some cases, at least some members are not so strongly connected. To capture this cases, the definition of clique can be relaxed.

One way to do so is to define a node as a member of a clique if it is connected to each node of the clique at a distance greater than 1. In this case, the path distance 2 is used. This definition of clique is called *N-clique*, where N stands for the maximum length of the allowed path. The definition of N-clique presents some weaknesses. For instance, it tends to return long and stringy N-cliques. Furthermore, N-cliques have properties undesirable for many purposes. For instance, some nodes of N-cliques could be connected by nodes that are not, themselves, members of the N-clique. To overcome this last problem, it is possible to require that the path distance between any two nodes of an N-clique satisfies a further condition, which forces all links among members of an N-clique to occur by way of other members of the N-clique. The structure thus obtained is called *N-clan*.

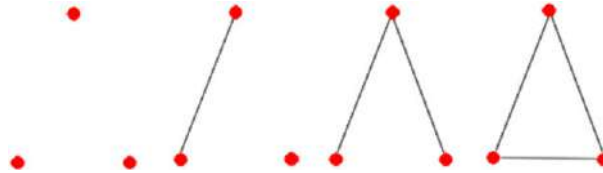


Fig. 3 The possible kinds of relationship involving a triad in an undirected network.

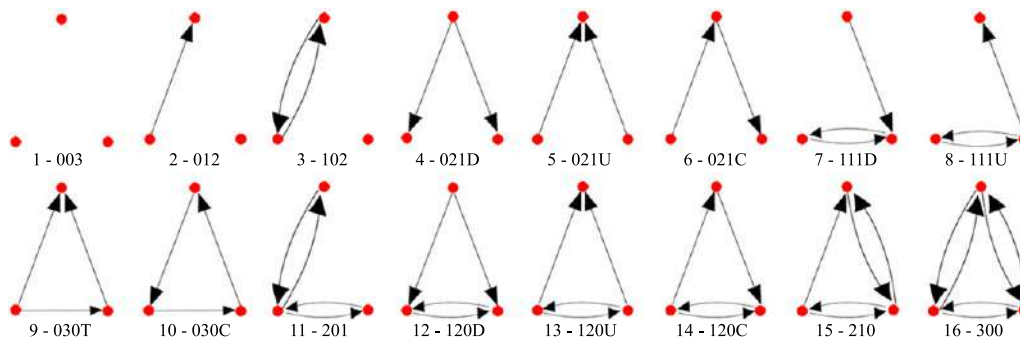


Fig. 4 The possible kinds of relationship involving a triad in a directed network.

An alternative way of relaxing the rigorous definition of clique consists of allowing nodes to be members of a clique even if they have edges to all but k other members. In other words, a node is a member of a clique of size N if it has edges to at least $N - k$ nodes of that clique. This relaxation leads to the definition of k -plex. While the N-clique approach often leads to large and stringy groupings, the k -plex approach often returns large numbers of smaller groupings. Depending on the goals of the analysis, both N-clique and k -plex could provide useful information about the sub-structure of groups.

A k -core is a maximal group of nodes all of whom connected to at least k other nodes of the group. The k -core approach is more relaxed than the k -plex one; indeed, it allows a node to join the group if it is connected to k other nodes, regardless of how many other nodes it may not be connected to. By varying the value of k different group structures can emerge. K-cores usually are more inclusive than k -plexes.

If a network is weighted, it is possible to introduce a last concept, i.e., the concept of F -groups. F -groups return the maximal groups formed by “strongly transitive” and “weakly transitive” triads. A strongly transitive triad exists if there is an edge (v_i, v_j, w_{ij}) , an edge (v_j, v_k, w_{jk}) and an edge (v_i, v_k, w_{ik}) and $w_{ij} = w_{jk} = w_{ik}$. A weakly transitive triad exists if w_{ij} and w_{jk} are both higher than w_{ik} , but this last is greater than some cut-off value.

Components

Components of a network are subnetworks that are connected within, but disconnected between networks. Interesting components are those dividing the network into separate parts such that each part has several nodes connected to one another (regardless of how much they are closely connected).

For directed networks, it is possible to define two different kinds of component. A weak component is a set of nodes that are connected, regardless of the direction of edges. A strong component also considers the direction of edges, when it verifies node connection.

Rather as the strict definition of clique may be too strong to capture the concept of maximal group, the notion of component may be too strong to capture all the meaningful weak points, holes and locally dense sub-parts of a larger network. Therefore, also for components, some more flexible definitions have been proposed. Due to space limitations, we do not illustrate these definitions in detail.

Other Indexes

In this section, we present some other concepts about NA that can be very useful in Bioinformatics and Computational Biology. The first concept regards the diffusion in a network. Several past studies show that the diffusion rate in a network is initially linear. However, if a *critical mass* is reached, this rate becomes exponential until the network is saturated. The same investigations show that the critical mass is reached when about 7% of nodes are reached by the diffusion process. From an economic point of view, in a diffusion process, critical mass is reached when benefits start outweighing costs. If benefits do not balance costs, the critical mass is not obtained, and the diffusion process itself will fail eventually.

If diffusion regards information, there are several indexes that can help to foresee if a node v_i will contribute to the diffusion process. These indexes are: (i) *relevance* (does v_i care at all?); (ii) *saliency* (does v_i care right now?); (iii) *resonance* (does information content mesh with what the actor associated with v_i believes in?); (iv) *severity* (how good or bad is information content?); (v) *immediacy* (does information require an immediate action?); (vi) *certainty* (does information cause pain or pleasure?); (vii) *source* (which did information come from and does v_i trust this source?); (viii) *entertainment value* (is information funny?).

To understand the behavior of actors in a network, a key concept is *homophily*. It states that two actors, who share some properties, will more likely form links than two actors, who do not. Other ways to express the same concept state that: (i) two actors being very close to a third one in a network often tend to link to each other; (ii) two actors sharing attributes are likely to be at closer distance to one another in networks.

In real networks, homophily is a major force, which, if left alone, would lead communities to become excessively uniform. To avoid this risk, two important elements act in real life, i.e., curiosity and weak ties. In particular, it was shown that weak ties are much more powerful than strong ties to stimulate innovation in the behavior of an actor or of a whole network.

A final important index to consider in this section is the so called *Dunbar number*. This index was determined by Robin Dunbar, who showed that, in real life, the average number of contacts that a person can really handle is 150 and that this number is limited by the size of our prefrontal cortex, as well as by human ability to reason about other people and relationships.

Closing Remarks

In this article, we have provided a presentation of several graph indexes and descriptors. We have seen that network analysis is largely employed in bioinformatics and biomedicine. Then, we have illustrated the most common network representations proposed in the past. Finally, we have presented a large variety of both basic and advanced indexes and descriptors. We think that the usage of graph-based indexes and descriptors in bioinformatics did not come to an end. On the contrary, in the future, the availability of large amounts of data in these contexts, along with the development of more and more powerful hardware, will lead to more and more complex and effective approaches for facing the new challenges that will appear in these sectors.

Acknowledgements

This work was partially supported by Aubay Italia S.P.A.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Strings and Sequences: Pairwise Alignment. Algorithms Foundations. Graphlets and Motifs in Biological Networks. Network-Based Analysis for Biological Discovery

References

- Achard, S., Salvador, R., Whitcher, B., Suckling, J., Bullmore, E.D., 2006. A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *The Journal of Neuroscience* 26 (1), 63–72.
- Aral, S.O. Sexual network patterns as determinants of std rates: Paradigm shift in the behavioral epidemiology of stds made visible, 1999.
- Becker, T., Leese, M., McCrone, P., *et al.*, 1998. Impact of community mental health services on users' social networks. *PRISM Psychosis Study. 7. The British Journal of Psychiatry* 173 (5), 404–408.
- Berkman, L., 1984. Assessing the physical health effects of social networks and social support. *Annual Review of Public Health* 5 (1), 413–432.
- Berkman, L., Glass, T., 2000. Social integration, social networks, social support, and health. *Social Epidemiology* 1, 137–173.
- Berkman, L.F., Kawachi, I., Glymour, M.M., 2014. *Social Epidemiology*. Oxford University Press.
- Borgatti, S., Foster, P., 2003. The network paradigm in organizational research: A review and typology. *Journal of Management* 29 (6), 991–1013.
- Cassel, J., 1976. The contribution of the social environment to host resistance. *American Journal of Epidemiology* 104 (2), 107–123.
- Chen, H., Ding, L., Wu, Z., *et al.*, 2009. Semantic web for integrated network analysis in biomedicine. *Briefings in Bioinformatics* 10 (2), 177–192.
- Cusick, M., Yu, H., Smolyar, A., *et al.*, 2009. Literature-curated protein interaction datasets. *Nature Methods* 6 (1), 39–46.
- del Rio, G., Koschutski, D., Coello, G., 2009. How to identify essential genes from molecular networks? *BMC Systems Biology* 3 (1), 102.
- Estrada, E., 2010. Generalized walks-based centrality measures for complex biological networks. *Journal of Theoretical Biology* 263 (4), 556–565. [Elsevier].
- Eysenbach, G., 2008. Medicine 2.0: Social networking, collaboration, participation, apomediation, and openness. *Journal of Medical Internet Research* 10 (3), e22.
- Friedman, S., Aral, S., 2001. Social networks, risk-potential networks, health, and disease. *Journal of Urban Health* 78 (3), 411–418.
- Friedman, S., Neaigus, A., Jose, B., *et al.*, 1997. Sociometric risk networks and risk for HIV infection. *American Journal of Public Health* 87 (8), 1289–1296.
- Gandhi, T., Zhong, J., Mathivanan, S., *et al.*, 2006. Analysis of the human protein interactome and comparison with yeast, worm and fly interaction datasets. *Nature Genetics* 38 (3), 285–293.
- Ghasemi, M., Seidkhani, H., Tamimi, F., Rahgozar, M., Masoudi-Nejad, A., 2014. Centrality measures in biological networks. *Current Bioinformatics* 9 (4), 426–441.
- Girvan, M., Newman, M.E., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Science of the United States of America* 99 (12), 7821–7826.
- Greicius, M., Krasnow, B., Reiss, A., Menon, V., 2003. Functional connectivity in the resting brain: A network analysis of the default mode hypothesis. *Proceedings of the National Academy of Sciences* 100 (1), 253–258.
- Guardiola, X., Diaz-Guilera, A., Perez, C., Arenas, A., Llas, M., 2002. Modeling diffusion of innovations in a social network. *Physical Review E* 66 (2), 026121.
- Hahn, M., Kern, A., 2005. Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. *Molecular Biology and Evolution* 22 (4), 803–806.
- Han, J., 2008. Understanding biological functions through molecular networks. *Cell research* 18 (2), 224–237.
- Hanneman, R., Riddle, M., 2005. Introduction to social network methods. <http://faculty.ucr.edu/widitdehanneman/nettext/>. Riverside: University of California.
- Haustein, S., Peters, I., Sugimoto, C., Thelwall, M., Lariviere, V., 2014. Tweeting biomedicine: An analysis of tweets and citations in the biomedical literature. *Journal of the Association for Information Science and Technology* 65 (4), 656–669.
- Hawn, C., 2009. Take two aspirin and tweet me in the morning: How Twitter, Facebook, and other social media are reshaping health care. *Health Affairs* 28 (2), 361–368.
- House, J., Landis, K., Umberson, D., 1988. Social relationships and health. *Science* 241 (4865), 540.
- Hsu, C., Juan, H., Huang, H., 2008. Characterization of microRNA-regulated protein-protein interaction network. *Proteomics* 8 (10), 1975–1979.
- Huang, D., Sherman, B., Lempicki, R., 2009. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols* 4 (1), 44–57.
- Jolly, A., Muth, S., Wylie, J., Potterat, J., 2001. Sexual networks and sexually transmitted infections: A tale of two cities. *Journal of Urban Health* 78 (3), 433–445.
- Junker, B., 2006. Exploration of biological network centralities with CentiBiN. *BMC Bioinformatics* 7 (1), 219.
- Kaplan, B., Cassel, J., Gore, S., 1977. Social support and health. *Medical Care* 15 (5), 47–58.
- Kapucu, N., 2005. Interorganizational coordination in dynamic context: Networks in emergency response management. *Connections* 26 (2), 33–48.
- Katz, E., Lazarsfeld, P., 1955. *Personal Influence*. New York: Free Press.
- Kearse, M., Moir, R., Wilson, A., *et al.*, 2012. Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics* 28 (12), 1647–1649.
- Kessler, R., Price, R., Wortman, C., 1985. Social factors in psychopathology: Stress, social support, and coping processes. *Annual Review of Psychology* 36 (1), 531–572.
- Knoke, D., Yang, S., 2008. *Social Network Analysis*, 154. Sage.
- Langfelder, P., Horvath, S., 2008. WGCNA: An R package for weighted correlation network analysis. *BMC Bioinformatics* 9 (1), 559.
- Laranjo, L., Arguel, A., 2014. The influence of social networking sites on health behavior change: A systematic review and meta-analysis. *Journal of the American Medical Association*, pages amiajn1–2014.
- Leischow, S.J., Milstein, B. *Systems thinking and modeling for public health practice*, 2006.
- Librado, P., Rozas, J., 2009. DnaSP v5: A software for comprehensive analysis of DNA polymorphism data. *Bioinformatics* 25 (11), 1451–1452.
- Lin, N., Ye, X., Ensel, W., 1999. Social support and depressed mood: A structural analysis. *Journal of Health and Social Behavior*. 344–359.
- Luke, D., Harris, J., 2007. Network analysis in public health: History, methods, and applications. *Annual Review of Public Health* 28, 69–93. [Annual Reviews].
- Mueller, N., Krauss, M., Luke, D., 2004. Interorganizational Relationships Within State Tobacco Control Networks: A Social Network Analysis. *Preventing Chronic Disease* 1 (4), 1–10.
- Ozgun, A., Vu, T., Erkan, G., Radev, D., 2008. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *24(13): i277–i285*.
- Rubinov, M., Sporns, O., 2010. Complex network measures of brain connectivity: Uses and interpretations. *Neuroimage* 52 (3), 1059–1069.
- Scanfeld, D., Scanfeld, V., Larson, E., 2010. Dissemination of health information through social networks: Twitter and antibiotics. *American Journal of Infection Control* 38 (3), 182–188.
- Scott, J., 2012. *Social Network Analysis*. Sage.
- Sevimoglu, T., Arga, K., 2014. The role of protein interaction networks in systems biomedicine. *Computational and Structural Biotechnology Journal* 11 (18), 22–27.

- Supekar, K., Menon, V., Rubin, D., Musen, M., Greicius, M.D., 2008. Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLOS Computational Biology* 4 (6), e1000100.
- Tsvetovat, M., Kouznetsov, A., 2011. *Social Network Analysis for Startups: Finding Connections on the Social Web*. O'Reilly Media, Inc.
- Valente, T., 1995. *Network Models of the Diffusion of Innovations*. Hampton Press.
- Valente, T., Davis, R., 1999. Accelerating the diffusion of innovations using opinion leaders. *The Annals of the American Academy of Political and Social Science* 566 (1), 55–67.
- Valente, T., Fosados, R., 2006. Diffusion of innovations and network segmentation: The part played by people in promoting health. *Sexually Transmitted Diseases* 33 (7), S23–S31.
- Wasserman, S., Galaskiewicz, J., 1994. *Advances in Social Network Analysis: Research in the Social and Behavioral Sciences*. 171. Sage.
- Wu, J., Vallenius, T., Ovaska, K., *et al.*, 2009. Integrated network analysis platform for protein-protein interactions. *Nature Methods* 6 (1), 75–77.
- Wu, K., Taki, Y., Sato, K., *et al.*, 2011. The overlapping community structure of structural brain network in young healthy individuals. *PLOS One* 6 (5), e19608.
- Xu, W., Chiu, I., Chen, Y., Mukherjee, T., 2015. Twitter hashtags for health: Applying network and content analyses to understand the health knowledge sharing in a Twitter-based community of practice. *Quality & Quantity* 49 (4), 1361–1380. [Springer].
- Yoon, J., Blumer, A., Lee, K., 2006. An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics* 22 (24), 3106–3108.
- Zalesky, A., Fornito, A., Bullmore, E., 2010. Network-based statistic: Identifying differences in brain networks. *Neuroimage* 53 (4), 1197–1207.
- Zhang, B., Horvath, S., 2005. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology* 4 (1), 1128.

Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs

Paolo Lo Giudice and Domenico Ursino, University “Mediterranea” of Reggio Calabria, Reggio Calabria, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Differently from other data analytics tasks, Network Analysis (NA) (Carrington *et al.*, 2005; Knoke and Yang, 2008; Wasserman and Galaskiewicz, 1994; Scott, 2012) focuses on *relationships* existing between actors, instead of on actors.

In Network Analysis, one of the most common issues regards the computational effort necessary for performing investigations. One of the most usual ways to face this issue consists in the adoption of sampling. Indeed, sampling approaches allow the extraction of knowledge about a network by investigating only a part of the network itself.

Clearly, the way the sample is chosen becomes crucial for extracting knowledge without errors or, at least, for minimizing the magnitude of errors. The problem of sampling from large graphs is discussed in Leskovec and Faloutsos (2006). Here, the authors investigate: (i) which sampling approaches must be used; (ii) how much the sampled graph can be reduced w.r.t. the original graph; (iii) how the measurements of a sample can be scaled up to get estimates for the corresponding (generally much larger) graph. The problem to obtain realistic samples with an as small as possible dimension is also described in Krishnamurthy *et al.* (2005). Here, the authors show that some of the analyzed methods can maintain the key properties of the original graph, even if the sample dimension is about 30% smaller than the original graph.

Sampling has been studied very much in the literature. For instance, in Gilbert and Levchenko (2004), the authors propose an approach that, given a communication network, determines the most important nodes and, then, links them each other. Instead, the authors of Rafei and Curial (2012) propose a technique, based on both sampling and the randomized notion of *focus*, to allow the visualization of very large networks. An analysis of the statistical properties of a sampled network can be found in Lee *et al.* (2006). In Ahn *et al.* (2007), the authors use the social network Cyworld to analyze the main features of the snowball sampling approach.

Other approaches, such as Chau *et al.* (2007), Gjoka *et al.* (2010), Kurant *et al.* (2010), Ye *et al.* (2010), focus mainly on sampling cost. Specifically, Ye *et al.* (2010) analyzes how much rapidly a crawler can reach nodes and links; Chau *et al.* (2007) proposes a framework of parallel crawlers based on Breadth First Search (BFS); Kurant *et al.* (2010) investigates the impact of different sampling techniques on the computation of the average node degree of a network; Gjoka *et al.*, (2010) studies several crawling strategies and determines the sampling quality guaranteed by them and the computation effort they require. Finally, in Buccafurri *et al.* (2014a,b), the authors describe how the crawling problem and its solutions change when passing from a social networking to a social internetworking scenario (i.e., in a scenario where several social networks interact each other through *bridge* nodes).

In bioinformatics and biomedicine, sampling of complex networks is a new and little investigated task. One of the main issues faced in these two contexts is the rapid growth of the scientific knowledge presented in the literature. Sampling is mainly used to classify such a knowledge. As a consequence, currently it is a supporting task for performing other activities, and is employed only rarely as the core task of an approach to facing issues in this context. For instance, in Coulet *et al.* (2010), Jin *et al.* (2008), Plaza *et al.* (2011), the authors present some approaches that employ sampling on the existing literature to create: (i) semantic maps based on relationships (Coulet *et al.*, 2010); (ii) summarizations (Plaza *et al.*, 2011); (iii) multi-label classifications (Jin *et al.*, 2008).

Sampling is also used to face a specific, yet extremely interesting, research problem, i.e., the search of motifs in a network. For instance, the authors of Kashtan *et al.* (2004) propose a new algorithm allowing the estimation of the subgraph concentration at runtime; furthermore, in Boomsma *et al.* (2008), the authors employ sampling to generate a probabilistic model of local protein structure; finally, in Alon *et al.* (2008), Wong *et al.* (2011), sampling is used to search motifs in biological networks.

In biomedical research, the most employed sampling approach is undoubtedly Random Walk (RW) and its variants. For instance, RW is adopted in Liu *et al.* (2016), Navlakha and Kingsford (2010) to evaluate the relationships between proteins, genes and diseases. In Leong and Morgenthaler (1995), the authors employ RW to investigate and plot DNA sequences. Finally, in Freschi (2007), Liu *et al.* (2013), Macropol *et al.* (2009), RW is used to discover functional models and to infer the pathway activity. In these last cases, RW allows users to capture the information embedded in structure and to represent it in the resulting graph.

This article aims at providing an exhaustive overview of the existing algorithms for traversing, searching and sampling networks. It is organized as follows. In Section Fundamentals, we illustrate some preliminary concepts and introduce the formalism adopted throughout this article. In Section Sampling Approaches, first we propose three taxonomies for sampling approaches and, then, we provide a brief description of each approach. In Section Analysis and Assessment, we present a comparison of sampling approaches based on property preservation and network property estimation. Finally, in Section Closing Remarks, we draw our conclusions and have a look at future possible developments of this research issue.

Fundamentals

A network $\mathcal{N} = \langle V, E \rangle$ consists of a set V of nodes and a set E of edges. We use $\mathcal{N}$ and m to denote $|V|$ and $|E|$. Each edge $e_{ij} = (v_i, v_j)$ connects the nodes v_i and v_j . Edges can be either directed (when they can be traversed only in one direction) or undirected (when they can be traversed in both directions). Furthermore, networks can be weighted or unweighted. If a network is weighted, the edge can be represented as (v_i, v_j, w_{ij}) , where w_{ij} denotes the weight of the edges. On the basis of the reference context, this weight could represent strength, distance, similarity, etc.

Let v_i be a node of V . The set of edges incident to v_i is defined as $\iota(v_i) = \{(v_j, v_i, w_{ji}) | (v_j, v_i, w_{ji}) \in E\}$. The neighborhood $\nu(v_i)$ is defined as $\nu(v_i) = \{v_j | (v_i, v_j, w_{ij}) \in E\}$.

A sampled network $\mathcal{N}_s = \langle V_s, E_s \rangle$ consists of a set $V_s \subseteq V$ of nodes and a set $E_s \subseteq E$ of edges such that $E_s \subseteq \{(v_i, v_j, w_{ij}) | v_i \in V_s, v_j \in V_s\}$. This last condition ensures that the sampled elements form a valid graph. We use the symbols $\mathcal{N}_s$ and m_s to denote $|V_s|$ and $|E_s|$, respectively. Clearly, $\mathcal{N}_s \leq \mathcal{N}$ and $m_s \leq m$. Each sampling activity has a cost and, often, a maximum budget B can be assigned to it.

Sampling Approaches

Taxonomies of Sampling Approaches

There exist several taxonomies of sampling approaches. A first classification considers the sampling objective. In this case, we can distinguish approaches that: (i) get a representative subset of nodes; (ii) preserve certain properties of the original network; (iii) generate a random network. As for this article, we will give more importance to the second type, i.e., property preservation.

A second taxonomy concerns the type of networks. In this case, we have: (i) Erdos-Renyi Network (ERN), also known as Random Graph, Exponential Random Graph, Poisson Random Graph, etc.; (ii) Power-Law Network (PLN), also called Scale-Free Network; (iii) Small-World Network (SMN); (iv) Fixed Degree Distribution Random Graph (FDDRG), also called "Configuration Model".

A third taxonomy is based on the adopted sampling techniques. In this case, we can consider:

- Node Sampling (NS).
- Edge Sampling (ES).
- Node Sampling with Neighborhood (NSN).
- Edge Sampling with Contraction (ESC).
- Node Sampling with Contraction (NSC).
- Traversal Based Sampling (TBS). This last is actually a family of techniques. In this case, the sampler starts with a set of initial nodes (and/or edges) and expands the sample on the basis of current observations. In this family, we can recognize:
 - Breadth First Search (BFS);
 - Depth First Sampling (DFS);
 - Random First Sampling (RFS);
 - Snowball Sampling (SBS);
 - Random Walk (RW);
 - Metropolis-Hastings Random Walk (MHRW);
 - Random Walk with Escaping (RWE);
 - Multiple Independent Random Walkers (MIRW);
 - Multi-Dimensional Random Walk (MDRW);
 - Forest Fire Sampling (FFS);
 - Respondent Driven Sampling (RDS) or Re-Weighted Random Walk (RWRW).

In the following, we use this last taxonomy and we give an overview to all the approaches mentioned above for it.

Description of Sampling Approaches

Node Sampling (NS)

This approach first selects V_s directly, i.e., uniformly or according to some distribution of V , determined on the basis of information about nodes already known. Then, it selects the edges of E_s in such a way that $E_s = \{(v_i, v_j, w_{ij}) | (v_i, v_j, w_{ij}) \in E, v_i \in V_s, v_j \in V_s\}$.

Edge Sampling (ES)

This approach first selects $E_s \subseteq E$ somehow. Then, it selects V_s as $V_s = \{v_i, v_j | (v_i, v_j, w_{ij}) \in E_s\}$. Alternatively, it can set $V_s = V$. In this last case, the edge sampling task reduces to a network sparsification task. As a matter of facts, network sparsification is a more general task than network sampling. Therefore, the latter can be considered as a specific case of the former.

Node Sampling with Neighborhood (NSN)

This approach first selects a set $\bar{V} \subseteq V$ directly, on the basis of available resources, without considering topology information. Then, it determines E_s as $E_s = \cup_{v_i \in \bar{V}} t(v_i)$ and $V_s = \{v_i, v_j | (v_i, v_j) \in E_s\}$. Finally, it returns $\mathcal{N}_s = \langle V_s, E_s \rangle$ as the sampled network.

Edge Sampling with Contraction (ESC)

This is an iterative process. At each step, it samples one edge $(v_i, v_j, w_{ij}) \in E$ and performs the following tasks: (i) it substitutes nodes v_i and v_j with only one node v_{ij} representing both of them; (ii) it substitutes each edge involving v_i or v_j with an edge involving v_{ij} ; (iii) it substitutes all the possible edges involving v_{ij} and the same node v_k with a unique edge involving the same nodes, whose weight is suitably determined from the weights of the merged edges, depending on the application context.

Node Sampling with Contraction (NSC)

This is an iterative process. At stage l , it samples one node v^l and contracts v^l and the nodes of $v(v^l)$ into one node. In carrying out this task, it suitably removes or modifies the corresponding edges. It is possible to show that NSC is a more constrained version of ESC.

Breadth First Search/Sampling (BFS), Depth First Search/Sampling (DFS), Random First Search/Sampling (RFS)

The Breadth First Sampling approach uses a support list L of nodes. Initially, it selects a starting node v^0 and sets L to $\{v^0\}$, V_s to $\{v^0\}$ and E_s to $\emptyset$. Then, it repeats the following tasks until the available budget B is exhausted: (i) it takes the first element v^l from L ; (ii) for each $v_j \in v(v^l)$ such that $v_j \notin V_s$ and $v_j \notin L$, it adds v_j to L ; v^l is called the “father” of v_j and is indicated as $f(v_j)$; (iii) it adds v^l to V_s ; (iv) it adds the edge (v^l, v_j) to E_s ; (v) it subtracts the cost of the current iteration from B .

DFS and RFS differ from BFS only in step (i) above. In fact, in DFS, the last element is selected from L , whereas, in RFS, a random element is chosen.

Snowball Sampling (SBS)

Snowball Sampling, or Network Sampling or Chain Referral Sampling, is often used in sociology when it is necessary to perform an investigation on a hidden population (e.g., alcoholists).

It starts from an initial set V^0 of nodes, which can be obtained randomly or based on the side knowledge of the hidden population.

At stage l , it first puts the set $\bar{V}^l$ of visited nodes and the set E^l of visited edges to $\emptyset$. Then, for each node $v^l \in V^{l-1}$, it selects k nodes belonging to the neighborhood $v(v^l)$ of v^l uniformly at random or according to some policy, adds them to $\bar{V}^l$ and adds the edges from v^l to each of these nodes to E^l . The methodology to perform the selection of the k nodes may depend on the application context. At the end of stage l , it constructs V^l as $V^l = \bar{V}^l - \cup_{j=0..l-1} V^j$.

The process is repeated for t stages until to the budget B is exhausted.

The final sampled network $\mathcal{N} = \langle V_s, E_s \rangle$ is constructed by setting $V_s = \cup_{j=0..t} V^j$ and $E_s = \cup_{j=1..t} E^j$.

Note that SBS is very similar to BFS. Indeed, the difference is that BFS considers the whole neighborhood of the current node, whereas SBS considers only k nodes of this neighborhood.

Random Walk (RW)

Random Walk starts from an initial node v^0 . Initially, it puts the set $\bar{E}_s$ of visited edges to $\emptyset$. At step l , it chooses one node v_j of the neighborhood $v(v^{l-1})$ of v^{l-1} . This choice can be performed uniformly at random or according to some policy. Then, it sets $v^l = v_j$ and adds to $\bar{E}_s$ the edge from v^{l-1} to v^l .

This process continues for t stages until to the budget B is exhausted. The final sampled network $\mathcal{N}_s = \langle V_s, E_s \rangle$ can be constructed in two different ways, namely:

- By setting $V_s = \{v^0, v^1, \dots, v^t\}$ and $E_s = \bar{E}_s$.
- By setting $\bar{V}_s = \{v^0, v^1, \dots, v^t\}$, $E_s = \cup_{v^l \in \bar{V}_s} t(v^l)$ and $V_s = \{v^l, v_j | (v^l, v_j) \in E_s\}$. In this case, RW reduces to Node Sampling with Neighborhood.

RW is also related to SBS. In fact, it can be considered as a specific case of SBS where $k=1$. However, there is an important difference between them because RW is memoryless. In fact, in SBS, the participants from previous stages are excluded, whereas, in RW, the same node can be visited more times.

It is possible to show that, when RW is applied on an undirected network, it returns a uniform distribution of edges. In this sense, it can be considered equivalent to ES.

Finally, it is worth pointing out that, if the choice of the next node to visit is performed uniformly at random, a node has a degree-proportional probability to be in V_s .

Metropolis-Hastings Random Walk (MHRW)

Metropolis-Hastings Random Walk is capable of returning a desired node distribution from an arbitrary undirected network. It uses two parameters, namely the probability P_{v^l, v_j} to pass from v^l to v_j and the desired distribution δ_v of a node v .

MHRW behaves analogously to RW. However, if v^l is the current node at stage l , the next node v_j to visit is determined according to the parameter P_{v^l, v_j} . The value of this parameter can be determined taking three possible cases into account. Specifically:

- If $v^l \neq v_j$ and $v_j \in \iota(v^l)$, then $P_{v^l, v_j} = M_{v^l, v_j} \cdot \min\left\{1, \frac{\delta_{v_j}}{\delta_{v^l}}\right\}$.
- If $v^l \neq v_j$ and $v_j \notin \iota(v^l)$, then $P_{v^l, v_j} = 0$.
- If $v^l = v_j$ then $P_{v^l, v_j} = 1 - \sum_{v_k \in V_s, v_k \neq v^l} P_{v^l, v_k}$.

Here, $M_{v^l, v_j} = M_{v_j, v^l}$ is a normalization factor for the pair $\langle v^l, v_j \rangle$. It allows the condition $\sum_{v_k \in V_s, v_k \neq v^l} P_{v^l, v_k} \leq 1$ to be satisfied. Since adding more higher-weight self-loops makes the mixing time longer, M_{v^l, v_j} should be selected as larger as possible. A possible choice for it is $M_{v^l, v_j} \min\left\{\frac{1}{|\iota(v^l)|}, \frac{1}{|\iota(v_j)|}\right\}$.

The application scenario of MHRW is more limited than the one of RW. In fact, to calculate P_{v^l, v_j} the degree of the neighboring nodes should be known. This information is often unavailable even if, in some cases, it is fixed (e.g., in P2P) or it can be obtained through a suitable API (e.g., in Online Social Networks).

Random Walk with Escaping (RWE)

Random Walk with Escaping, or Random Jump, is analogous to RW. However, if v^l is the current node, to determine the next node to visit, besides walking to a node of $\iota(v^l)$, RWE can jump to an arbitrary random node $v_j \in V$. RWE is not very meaningful as a sampling technique. Indeed, it is classified as a TBS technique. However, TBS generally operates when the whole network cannot be reached, or at least direct NS or ES is hard. By contrast, RWE needs an efficient NS as a support. As a consequence, it cannot be adopted in several scenarios. Furthermore, it is possible to show that, even when RWE can be adopted, it is hard to construct unbiased estimators for the properties of $\mathcal{N}$ starting from the ones of $\mathcal{N}_s$.

Multiple Independent Random Walkers (MIRW)

One problem of RW is that it tends to be trapped into local dense regions. Therefore, it could have high biases, according to different initial nodes. Multiple Independent Random Walkers was proposed to face this problem. First, it applies NS to choose l initial nodes. Then, it splits the budget B among l Random Walks and let them execute independently from each other. Finally, it merges the results produced by the l Random Walkers. As a matter of facts, it was shown that the estimation errors of MIRW are higher than the ones of MDRW (see Section Multi-dimensional random walk (MDRW)). As a consequence, we have mentioned MIRW only for completeness purposes.

Multi-Dimensional Random Walk (MDRW)

Multi-Dimensional Random Walk, or Frontier Sampling, starts by determining the number k of dimensions. Then, it initializes a list L of nodes by assigning k nodes, determined randomly via NS, to it. After this, it performs several iterations until to the Budget B is exhausted.

During one of these iterations, first it chooses one node v^l from L with the probability $p(v^l)$ proportional to $|\iota(v^l)|$. Then, it selects a node $v_j \in \iota(v^l)$. Finally, it adds the edge (v^l, v_j, w_{ij}) to E_s and substitutes v^l with v_j in L .

It was shown that: (i) MDRW provides very good estimations of some graph properties; (ii) when $l \rightarrow \infty$, MDRW obtains a uniform distributions of both nodes and edges.

Forest Fire Sampling (FFS)

Forest Fire Sampling can be considered as a probabilistic version of Snowball Sampling (see Section Snowball sampling (SBS)). Specifically, in SSB, k neighbors are selected at each round, whereas, in FFS, a geometrically distributed number of neighbors is selected at each round. If the parameter p of the geometric distribution is set to $\frac{1}{k}$, then the corresponding expectation is equal to k and FFS behaves very similarly to SBS.

An important common point between FFS and SBS, which differentiates both of them from RW and its variants, is that, in FFS and SBS, when a node is visited, it will no longer be visited again. By contrast, in RW and its variants, repeated nodes are included in the sample for estimation purposes.

Respondent Driven Sampling (RDS)

The original idea of Respondent Driven Sampling is to run SBS and to correct the bias according to the sampling probability of each node of V_s . Currently, SBS is often substituted by RW because the bias of RW can be more easily corrected. In this case, RDS is also called Re-Weighted Random Walk (RWRW). We point out that, actually, RDS itself is not a standalone network sampling technique. Indeed, it uses SBS or RW for sampling and, then, corrects the corresponding bias.

The principle underlying this approach is the following: it does not matter the adopted sampling technique (NS, ES or TBS); as long as the sample probability is known, the suitable bias correction technique can be invoked.

If we consider sampling and estimating tasks as a whole activity, RWRW and MHRW seem to have the same objective and similar results.

RWRW is a practical approach to estimate several properties without knowing the full graph.

Analysis and Assessment

In this section, we propose a comparison of network sampling approaches as far as network property preservation and network property estimation are concerned. Although these two goals are different, their results are strictly related and can transform to each other.

In the literature, it was shown that the Node Sampling or the Edge Sampling approaches and their variants (i.e., NS, ES, NSN, ESC and NSC) are completely dominated by Traversal Based Sampling approaches across all network features. Among the TBS approaches, there is no clear single winner. Each approach is the best one for at least a network feature of a particular network configuration.

More specifically, it was shown that, in presence of a Poisson degree distribution, approaches as SBS and FFS, configured with the mean of its geometric distribution set to 80% of the number of the remaining unselected neighbors (we call FFS80% this configuration), which can reconstruct a good representation of some local parts of the network nodes, perform relatively well. Furthermore, in presence of a power law degree distribution, approaches as RW and FFS, configured with the mean of the geometric distribution set to 20% of the number of the remaining unselected methods (we call FFS20% this configuration), which explore nodes farther away from the focal ones, perform better.

A closer examination of the approaches provides an, at least partial, motivation of these results. Indeed, SSB tends to return sampled networks, whose degree distributions contain inflated proportions of nodes with the highest and the lowest degrees. Clearly, this causes a poor performance of this approaches when applied to networks with a power-law degree distribution, which are characterized by a small proportion of high-degree nodes. On the contrary, RW tends to return sampled networks, whose nodes never have the highest degrees. Now, since the proportion of nodes with the highest degree is much lesser in the power-law degree distribution than in the Poisson distribution, RW performs better when applied to networks with the former distribution than to networks with the latter one. Furthermore, networks with Poisson distributions tend to be homogeneous throughout their regions; as a consequence, a locally-oriented approach, like SSB, can provide good results. On the contrary, networks with power-law degree distributions require a more global exploration; as a consequence, for this kind of network, FFS and RW appear more adequate.

Summarizing and, at the same time, deepening this topic, we can say that SBS is well suited for sampling social networks with Poisson degree distribution, RW is adequate for sparse social networks with power-law degree distribution and FF is well suited for dense social networks with power-law degree distribution. To implement this recommendation, the knowledge of the degree distribution of network nodes must be known. However, this information may be unavailable in many cases. In the literature, it was shown that FFS presents the best overall performance in determining degree distributions across different kinds of network and sample size. Therefore, it could be useful to exploit an adaptive sampling procedure using different sampling approaches at different stages. For instance, this procedure could start with FFS when no knowledge about the distribution of network nodes is available. Then, after a certain number of nodes have been included in the sample, it is possible to determine the degree distribution of the current sample and, based on it, to continue with FFS or to switch to SBS or RW.

Closing Remarks

In this article, we have provided a general presentation of algorithms for traversing, searching and sampling graphs. We have seen that these algorithms have been investigated very much in the past literature in many research fields. Instead, they were little employed in bioinformatics and biomedicine, where the most important adoption cases regard knowledge classification and motif search. In this article, we have introduced a formalism to represent a complex network, we have provided three taxonomies of sampling approaches, we have presented a brief description of each of them and, finally, we have compared them. We think that network traversing/searching/sampling approaches could have much more usage cases in the future. As a matter of fact, the amount of available data is increasing enormously. This fact could give origin to more and more sophisticated networks. In several cases, it could be impossible to perform the analysis of the whole network; when this happens, the possibility to have some reliable samples of them could be extremely beneficial.

Acknowledgement

This work was partially supported by Aubay Italia S.p.A.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms Foundations. Biological Database Searching. Graphlets and Motifs in Biological Networks. Network-Based Analysis for Biological Discovery

References

- Ahn, Y.Y., Han, S., Kwak, H., Moon, S., Jeong, H., 2007. Analysis of topological characteristics of huge online social networking services. In: Proceedings of the International Conference on World Wide Web (WWW'07), pp. 835–844. Banff, Alberta, Canada. ACM.
- Alon, N., Dao, P., Hajirasouliha, I., Hormozdiari, F., Sahinalp, S., 2008. Biomolecular network motif counting and discovery by color coding. *Bioinformatics* 24 (13), i241–i249.
- Boomsma, W., Mardia, K., Taylor, C., *et al.*, 2008. A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences* 105 (26), 8932–8937.
- Buccafurri, F., Lax, G., Nocera, A., Ursino, D., 2014a. Experiences using BDS, a crawler for Social Internetworking Scenarios. *Social Networks: Analysis and Case Studies*. Springer. (Lecture Notes in Social Networks).
- Buccafurri, F., Lax, G., Nocera, A., Ursino, D., 2014b. Moving from social networks to social internetworking scenarios: The crawling perspective. *Information Sciences*, 256. Elsevier. pp. 126–137.
- Carrington, P., Scott, J., Wasserman, S., 2005. *Models and Methods in Social Network Analysis*. Cambridge University Press.
- Chau, D.H., Pandit, S., Wang, S., Faloutsos, C., 2007. Parallel crawling for online social networks. In: Proceedings of the International Conference on World Wide Web (WWW'07), pp. 1283–1284. Banff, Alberta, Canada. ACM.
- Coulet, A., Shah, N., Garten, Y., Musen, M., Altman, R., 2010. Using text to build semantic networks for pharmacogenomics. *Journal of Biomedical Informatics* 43 (6), 1009–1019.
- Freschi, V., 2007. Protein function prediction from interaction networks using a random walk ranking algorithm. In: Proceedings of the International Conference on Bioinformatics and Bioengineering (BIBE 2007), pp. 42–48. Harvard, MA, USA. IEEE.
- Gilbert, A.C., Levchenko, K., 2004. Compressing network graphs. In: Proceedings of the International Workshop on Link Analysis and Group Detection (LinkKDD'04), Seattle, WA, USA. ACM.
- Gjoka, M., Kurant, M., Butts, C.T., Markopoulou, A., 2010. Walking in Facebook: A case study of unbiased sampling of OSNs. In: Proceedings of the International Conference on Computer Communications (INFOCOM'10), pp. 1–9. San Diego, CA, USA. IEEE.
- Jin, B., Muller, B., Zhai, C., Lu, X., 2008. Multi-label literature classification based on the Gene Ontology graph. *BMC Bioinformatics* 9 (1), 525.
- Kashtan, N., Itzkovitz, S., Milo, R., Alon, U., 2004. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics* 20 (11), 1746–1758.
- Knoke, D., Yang, S., 2008. *Social Network Analysis*, 154. Sage.
- Krishnamurthy, V., Faloutsos, M., Chrobak, M., *et al.*, 2005. Reducing Large Internet Topologies for Faster Simulations. In: Proceedings of the International Conference on Networking (Networking 2005), pp. 165–172. Waterloo, Ontario, Canada. Springer.
- Kurant, M., Markopoulou, A., Thiran, P., 2010. On the bias of BFS (Breadth First Search). In: Proceedings of the International Teletraffic Congress (ITC 22), pp.1–8. Amsterdam, The Netherlands. IEEE.
- Lee, S.H., Kim, P.J., Jeong, H., 2006. Statistical properties of sampled networks. *Physical Review E* 73 (1), 016102.
- Leong, P., Morgenthaler, S., 1995. Random walk and gap plots of DNA sequences. *Computer Applications in the Biosciences: CABIOS* 11 (5), 503–507.
- Leskovec, J., Faloutsos, C., 2006. Sampling from large graphs. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), pp. 631–636, Philadelphia, PA, USA: ACM.
- Liu, W., Li, C., Xu, Y., *et al.*, 2013. Topologically inferring risk-active pathways toward precise cancer classification by directed random walk. *Bioinformatics* 29 (17), 2169–2177.
- Liu, Y., Zeng, X., He, Z., Zou, Q., 2016. Inferring microRNA-disease associations by random walk on a heterogeneous network with multiple data sources. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- Macropol, K., Can, T., Singh, A., 2009. RRW: Repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics* 10 (1), 283.
- Navlakha, S., Kingsford, C., 2010. The power of protein interaction networks for associating genes with diseases. *Bioinformatics* 26 (8), 1057–1063.
- Plaza, L., Diaz, A., Gervas, P., 2011. A semantic graph-based approach to biomedical summarisation. *Artificial Intelligence in Medicine* 53 (1), 1–14.
- D. Rafiei, S. Curial, 2012. Effectively visualizing large networks through sampling. In: Proceedings of the IEEE Visualization Conference 2005 (VIS'05), p. 48. Minneapolis, MN, USA, 2005. IEEE.
- Scott, J., 2012. *Social Network Analysis*. Sage.
- Wasserman, S., Galaskiewicz, J., 1994. *Advances in Social Network Analysis: Research in the Social and Behavioral Sciences*, 171. Sage Publications.
- Wong, E., Baur, B., Quader, S., Huang, C., 2011. *Biological network motif detection: Principles and practice*. Briefings in Bioinformatics. Oxford Univ Press.
- Ye, S., Lang, J., Wu, F., 2010. Crawling online social graphs. In: Proceedings of the International Asia-Pacific Web Conference (APWeb'10), pp. 236–242. Busan, Korea. IEEE.

Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs

Clara Pizzuti, Institute for High Performance Computing and Networking (ICAR), Cosenza, Italy
Simona E Rombo, University of Palermo, Palermo, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Many studies have been performed on graphs modeling the complex interactions occurring among different components in the cell (Atias and Sharan, 2012; De Virgilio and Rombo, 2012; Ferraro *et al.*, 2011; Panni and Rombo, 2015; Pizzuti *et al.*, 2012; Sharan *et al.*, 2007). Most of these studies involve the analysis of graph topology, according to different points of view. In this manuscript, we consider two specific problems based on the analysis of graph topology, that are: Clustering of graphs representing biological networks, and searching for motifs in biological networks.

In particular, an important problem in Biology is the detection of molecular complexes, that can help in understanding the mechanisms regulating cell life, in describing the evolutionary orthology signal (e.g., (Jancura *et al.*, 2011)), in predicting the biological functions of uncharacterized proteins, and, more importantly, for therapeutic purposes. The problem of detecting molecular complexes from biological networks can be computationally addressed using clustering techniques. Clustering consists of grouping data objects into groups (also called *clusters* or *communities*) such that the objects in the same cluster are more similar to each other than the objects in the other clusters (Jain, 1988). Possible uncharacterized proteins in a cluster may be assigned to the biological function recognized for that module, and groups of proteins performing the same tasks can be singled out this way. As observed in Fortunato (2010), a generally accepted definition of “cluster” does not exist in the context of networks, since it depends on the specific application domain. However, it is widely accepted that a community should have more internal than external connections. For biological networks, the most common assumption is that clusters are groups of highly connected nodes, although recently the notion of community intended as a set of topologically similar links has been successfully used in Ahn *et al.* (2010) and Solava *et al.* (2012).

As for the search of motifs, the concept of *motif* has been exploited in different applications of computational biology (Apostolico *et al.*, 2008a,b; Furfaro *et al.*, 2017; Parida, 2008, 2014). Depending on the context, *what is a motif* may assume sensibly different meanings. In general, motifs are always associated to *interesting* repetitions in a given data set. Interestingness is the key concept for the definition of motif; for example, a repetition can be considered interesting when its frequency is greater than a fixed threshold, or instead when it is much different than expected (Apostolico *et al.*, 2003). Also in the context of biological networks, a motif can be defined according to its *frequency* or to its *statistical significance* (Ciriello and Guerra, 2008). In the first case, a motif is a subgraph that appears more than a threshold number of times in an input network; in the second case, a motif is a subgraph that appears more often than expected by chance. In particular, to measure the statistical significance of the motifs, many works compare the number of appearances of the motifs in the biological network with the number of appearances in a number of randomized networks (Erdos and Renyi, 1959, 1960), by exploiting suitable statistical indices such as *p-value* and *z-score* (Milo *et al.*, 2002).

Here we provide a compact overview of the main algorithms and techniques proposed in the literature to solve both clustering and motifs search in biological networks.

Algorithms for Network Clustering

Local Neighborhood Density Search

Many methods, including the most popular, are based on local neighbourhood density search. Their objective is to find dense subgraphs (that is, each node is connected to many other nodes in the same subgraph) within the input network. We summarize in the following seven representative methods in this class.

One of the most popular methods for finding modules in *protein – protein* interaction networks is MCODE (Bader and Hogue, 2003). This method employs a node weighting procedure by local neighbourhood density and outward traversal from a locally dense seed protein, in order to isolate the dense regions according to given input parameters. The algorithm allows fine-tuning of clusters of interest without considering the rest of the network and allows examination of cluster interconnectivity, which is relevant for protein networks. It is implemented as Cytoscape plug-in. With a user-friendly interface, it is suited for both computationally and biologically oriented researchers.

In Altaf-Ul-Amin *et al.* (2006) the DPCLUS method for discovering protein complexes in large interaction graphs was introduced. It is based on the concepts of *node weight* and *cluster property* which are used for selecting a seed node to be expanded by iteratively adding neighbours, and to terminate the expansion process, respectively. Once a cluster is generated, its nodes are removed from the graph and the next cluster is generated using only the remaining nodes until all the nodes have been assigned to a cluster. The algorithm also allows to generate overlapping clusters by keeping the nodes already assigned to clusters.

CFINDER is a program for detecting and analyzing overlapping dense groups of nodes in networks; it is based on the clique percolation concept (see (Adamcsek *et al.*, 2006; Derenyi *et al.*, 2005; Palla *et al.*, 2005)). The idea behind this method is that a cluster can be interpreted as the union of small fully connected sub-graphs that share nodes, where a parameter is used to specify the minimum number of shared nodes.

RANCoC (Pizzuti and Rombo, 2012), MF-PINCoC (Pizzuti and Rombo, 2008) and PINCoC (Pizzuti and Rombo, 2007) are based on greedy local expansion. They expand a single protein randomly selected by adding/removing proteins to improve a given quality function, based on the concept of co-clustering (Madeira and Oliveira, 2004). In order to escape poor local maxima, with a given probability, the protein causing the minimal decrease of the quality function is removed in MF-PINCoC and PINCoC. Instead RANCoC removes, with a fixed probability, a protein at random, even if the value of the quality function diminishes. This strategy is more efficient in terms of computation than that applied in the methods (Pizzuti and Rombo, 2007, 2008), and it is more efficacious in avoiding entrapments in local optimal solutions. All three algorithms work until either a preset of maximum number of iterations has been reached, or the solution cannot further be improved. Both MF-PINCoC and RANCoC allow for overlapping clusters.

DME (Georgii *et al.*, 2009) is a method for extracting dense modules from a weighted interaction network. The method detects all the node subsets that satisfy a user-defined minimum density threshold. The method returns only locally maximal solutions, i.e., modules where all the direct supermodules (containing one additional node) do not satisfy the minimum density threshold. The obtained modules are ranked according to the probability that a random selection of the same number of nodes produces a module with at least the same density. An interesting property of this method is that it allows to incorporate constraints with respect to additional data sources.

Cost-Based Local Search

Methods based on cost-based local search extract modules from the interaction graph by partitioning the graph into connected subgraphs, using a cost function for guiding the search towards a best partition. We describe here in short three methods based on this approach with different characteristics.

A typical instance of this approach is RNSC (King *et al.*, 2004), which explores the solution space of all the possible clusterings in order to minimize a cost function that reflects the number of inter-cluster and intra-cluster edges. The algorithm begins with a random clustering, and attempts to find a clustering with best cost by repeatedly moving one node from a cluster to another one. A list of tabular moves is used to forbid cycling back to previously examined solutions. In order to output clusters likely to correspond to true protein complexes, thresholds for minimum cluster size, minimum density, and functional homogeneity must be set. Only clusters satisfying these criteria are given as the final result. This obviously implies that many proteins are not assigned to any cluster.

Several community discovery algorithms have been proposed based on the optimization of a modularity-based function (see e.g. (Fortunato, 2010)). Modularity measures the fraction of edges falling within communities, subtracted by what would be expected if the edges were randomly placed. In particular, Qcut (Ruan and Zhang, 2008) is an efficient heuristic algorithm applied to detect protein complexes. Qcut optimizes modularity by combining spectral graph partitioning and local search. By optimizing modularity, communities that are smaller than a certain scale or have relatively high inter-community density may be merged into a single cluster. In order to overcome this drawback, the authors introduce an algorithm that recursively applies Qcut to divide a community into sub-communities. In order to avoid over-partitioning, a statistical test is applied to determine whether a community indeed contains intrinsic sub-community.

ModuLand (Kovacs *et al.*, 2010) is an integrative method family for determining overlapping network modules as hills of an influence function-based, centrality-type community landscape, and including several widely used modularization methods as special cases. Several algorithms obtained from ModuLand provide an efficient analysis of weighted and directed networks, determine overlapping modules with high resolution, uncover a detailed hierarchical network structure allowing an efficient, zoom-in analysis of large networks, and allow the determination of key network nodes. It is implemented as Cytoscape plug-in.

Flow Simulation

Methods based on the flow simulation approach mimic the spread of information on a network. We consider three methods based on this approach.

One of the first flow simulation methods for detecting protein complexes in a protein-protein interaction network is the *Markov Clustering algorithm* MCL (Enright *et al.*, 2002). MCL simulates the behaviour of many walkers starting from the same point, that move within the graph in a random way.

Another method based on flow simulation is RRW (Macropol *et al.*, 2009). RRW is an efficient and biologically sensitive algorithm based on repeated random walks for discovering functional modules, which implicitly makes use of network topology, edge weights, and long range interactions between proteins.

An interesting method based on flow simulation is STM (Hwang *et al.*, 2006), which finds clusters of arbitrary shape by modeling the dynamic relationships between proteins of a protein-protein interaction network as a signal transduction system. The overall signal transduction behaviour between two proteins of the network is defined in order to evaluate the perturbation of one protein on the other one, both biologically and topologically. The signal transduction behaviour is modelled using the Erlang distribution.

Statistical Measures

The two following approaches rely on the use of statistical concepts to cluster proteins. They are based on the number of shared neighbours between two proteins, and on the notion of preferential attachment of the members of a module to other elements of the same module, respectively.

Samantha and Liang (2003) proposed a clustering method, here called SL by the names of the authors, based on the idea that if two proteins share a number of common interaction partners larger than what would be expected in a random network, then they should be clustered together. The method assesses the statistical significance of forming shared partnership between a pair of proteins using the concept of p-value of a pair of proteins.

The p-values of all proteins pairs are computed and stored in a similarity matrix. The protein pair with the lowest p-value is chosen to form the first group and the corresponding rows and columns of the matrix are merged in a new row and column. The new p-value of the merged row/column is the geometric mean of the separate p-values of the corresponding elements. This process is repeated by adding new proteins to the actual cluster until a threshold is reached. The process is repeated on the remaining proteins until all the proteins have been clustered.

In Farutin *et al.* (2006) a statistical approach for the identification of protein clusters is presented, here called FARUTIN (the name of the first author). This method is based on the concept of preferential interaction among the members of a module. The authors use a novel metric to measure the community strength. The community strength is gauged by the preferential attachment of each member of a module to the other elements of the same module. This concept of preferential attachment is quantified by how unlikely it is observed in a random graph.

Population-Based Stochastic Search

Population-based stochastic search has been used for developing algorithms for community detection in networks (see e.g., (Pizzuti, 2008; Tasgin and Bingol, 2007)).

In Liu and Liu (2006) the authors proposed an algorithm based on evolutionary computation, here called CGA, for enumerating maximal cliques and apply it to the Yeast genomic data. The advantage of this method is that it can find as many potential protein complexes as possible.

In Ravae *et al.* (2010) an immune genetic algorithm, here called IGA, is described to find dense subgraphs based on efficient vaccination method, variable-length antibody schema definition and new local and global mutations. The algorithm is applied to clustering protein-protein interaction networks

In GA-PPI (Pizzuti and Rombo, 2014a,b) the adopted representation of individuals is the graph-based adjacency representation, originally proposed in Park and Song (1989), where an individual of the population consists of n genes, each corresponding to a node of the graph modeling the protein-protein interaction network. A value j assigned to the i th gene is interpreted as a link between the proteins i and j , and implies that i and j belong to the same cluster. In particular, in Pizzuti and Rombo (2014a) the fitness functions of *conductance*, *expansion*, *cut ratio*, *normalized cut*, introduced by Leskovec *et al.* (2010), are employed, while in Pizzuti and Rombo (2014b) the cost functions of the RNSC algorithm (King *et al.*, 2004) have been used.

Link Clustering

Link clustering methods group the set of edges rather than the set of nodes of the input network, often exploiting suitable techniques to compute edge similarity (Kuchaiev *et al.*, 2011; Milenkovic and Przulj, 2008; Przulj, 2007; Solava *et al.*, 2012). In Evans and Lambiotte (2010), Evans and Lambiotte (2009), Pizzuti (2009) link clustering is used to discover overlapping communities in complex networks different than protein-protein interaction networks. In the following we summarize two link clustering techniques applied to protein-protein interaction networks.

Given an input protein-protein interaction network N , the approach by Pereira *et al.* (2004) builds the corresponding line graph G . In particular, a vertex of G represents an edge of N , and two vertices are adjacent in G if and only if their corresponding edges in N share a common endpoint. Thus, each node of G represents an interaction between two proteins, and each edge represents pairs of interactions connected by a common protein. Pereira *et al.* apply MCL (Enright *et al.*, 2002) on G , and detect this way overlapping protein modules in N .

Ahn *et al.* (2010) propose an agglomerative link clustering approach to group links into topologically related clusters. The algorithm applies a hierarchical method based on the notion of *link similarity*, that is used to find the pair of links with the largest similarity in order to merge their respective communities. The similarity between two links takes into account the size of both the intersection and the union of their neighbourhoods. The agglomerative process is repeated until all the links belong to a single cluster. To find a meaningful community structure, it is necessary to decide where the built dendrogram must be cut. To this end, the authors introduce the concept of *partition density* to measure the quality of a link partitioning, and they choose the partitioning having the best partition density value.

Link clustering approaches have the main advantage that nodes are automatically allowed to be present in multiple communities, without the necessity of performing multiple clustering on the set of edges. As a negative point, if the input network is dense, then link clustering may become computationally expensive. We also observe that the performances of these techniques may depend on the link similarity measure they adopt. This issue is addressed by Solava *et al.* (2012), where a new similarity measure, extending that proposed in by Przulj (2007), has been defined. In particular, this measure is based on the topological

similarity of edges, computed by taking into account non-adjacent, though close, edges and counting the number of graphlets (i.e. are small induced subgraphs containing from 2 up to 5 nodes) each edge touches.

Algorithms for the Search of Motifs in Graphs

Given a biological network N , a *motif* can be defined according to its *frequency* or to its *statistical significance* (Ciriello and Guerra, 2008). In the first case, a motif is a subgraph appearing more than a threshold number of times in N ; in the second case, it is a subgraph occurring more often than expected by chance. In particular, to measure the statistical significance of a motif, many works compare its number of occurrences with those detected in a number of randomized networks (Erdos and Renyi, 1960), by exploiting suitable statistical indices such as *p-value* and *z-score* (Milo et al., 2002).

Milo et al. and its Extensions

The search of significant motifs in biological networks is pioneered by Shen-Orr et al. (2002), where *network motifs* have been defined as “patterns of interconnections that recur in many different parts of a network at frequencies much higher than those found in randomized networks”. The authors of Shen-Orr et al. (2002) studied the transcriptional regulation network of *Escherichia coli*, by searching for small motifs composed by three-four nodes. In particular, three highly significant motifs characterizing such network have been discovered, the most famous is the “feed-forward loop”, whose importance has been shown also in further studies (Mangan and Alon, 2003; Mangan et al., 2005).

The technique presented in Shen-Orr et al. (2002) laid the foundations for different extensions, the main of which are Berg and Lassig (2004), Cheng et al. (2008), Yeager-Lotem et al. (2004).

Milo et al. (2002) generalized the approach presented in Shen-Orr et al. (2002), in order to detect any type of connectivity graph in networks representing a broad range of natural phenomena. In particular, they considered gene regulatory networks, ecosystem food webs (Cohen et al., 1990), in where nodes represent groups of species and edges are directed from a node representing a predator to the node representing its prey; neuronal connectivity networks (Kashtan et al., 2004), where nodes represent neurons (or neuron classes), and edges represent synaptic connections between the neurons; technological networks as sets of sequential logic electronic circuits (Cancho et al., 2001), where the nodes in these circuits represent logic gates and flip-flops.

Also Berg et al. in Berg and Lassig (2004) analyzed the gene regulation network of *E. coli*, following the line of Shen-Orr et al. (2002). In particular, they developed a search algorithm to extract topological motifs called *graph alignment*, in analogy to sequence alignment, that is based on a scoring function. The authors observed that, in biological networks, functionally related motifs do not need to be topologically identical; thus, they discussed motifs derived from families of mutually similar but not necessarily identical patterns. Then, they compare the maximum-likelihood alignment in the *E. coli* data set with suitable random graph ensembles. They considered two different steps, in order to disentangle the significance of the number of internal links, and of the mutual similarity of patterns found in the data.

In Yeager-Lotem et al. (2004) composite network motifs are searched for. Such motifs consist of both transcription regulation and protein-protein interactions that recur significantly more often than in random networks. The authors developed algorithms for detecting motifs in networks with two or more types of interactions. In particular, they modelled an integrated cellular interaction network by two types (colors) of edges, representing protein-protein and transcription-regulation interactions, and developed algorithms for detecting network motifs in networks with multiple types of edges. Such a study may be considered as a basic framework for detecting the building blocks of networks with multiple types of interactions.

The most evolutive extension of Shen-Orr et al. (2002) has been presented in Cheng et al. (2008), where two types of motifs have been defined, that are, *bridge motifs*, consisting of weak links only, and *brick motifs*, consisting of strong links only. In particular, links are considered *weak* or *strong* according to the strength of the corresponding interaction (Girvan and Newman, 2002; Newman, 2003). The authors proposed a method for performing simultaneously the detection of global statistical features and local connection structures, and the location of functionally and statistically significant network motifs. They distinguished *bridge* motifs (consisting of weak links only) and *brick* motifs (consisting of strong links only), observing that brick motifs play a central role in defining global topological organization (Dobrin et al., 2004); bridge motifs include isolated motifs that neither interact nor overlap with other motifs. Cheng et al. examined functional and topological differences between *bridge* and *brick* motifs for predicting biological network behaviors and functions.

Motifs are “Not-Isolated”

In Dobrin et al. (2004) Dobrin et al. studied the transcriptional regulatory network of the bacterium *Escherichia coli*. The authors distinguish *coherent* motifs where all the directed links are activating, from *incoherent* ones, where one of the links inhibits the activity of its target node. They observed that in the analyzed network the vast majority of motifs overlap generating distinct topological units referred to as *homologous motif clusters*; then, they merged all the homologous motif clusters, finding that they form a single large connected component (i.e., *motif supercluster*) in which the previously identified homologous motif clusters are no longer clearly separable.

In Mazurie et al. (2005) the integrated network of *Saccharomyces cerevisiae*, comprising transcriptional and protein-protein interaction data, has been investigated. A comparative analysis has been performed with respect to *Candida glabrata*, *Kluyveromyces lactis*, *Debaryomyces hansenii* and *Yarrowia lipolytica*, which belong to the same class of *hemiascomycetes* as *S. cerevisiae* but span a

broad evolutionary range. The fact that the four analyzed organisms share many functional similarities with *S. cerevisiae* and yet span a broad range of evolutionary distances, comparable to the entire phylum of chordates, make them ideal for protein comparisons. Then, phylogenetic profiles of genes within different forms of the motifs have been analyzed, and the functional role in vivo of the motifs was examined for those instances where enough biological information was available.

Other Approaches Based on Topology Only

In Prill *et al.* (2005) performed an exhaustive computational analysis showing that a dynamical property, related to the stability or robustness to small perturbations, is highly correlated with the relative abundance of small subnetworks (network motifs) in several previously determined biological networks. They argued that robust dynamical stability can be considered an influential property that can determine the non-random structure of biological networks.

In Wernicke (2006), Wernicke presented *MFinder*, an algorithm overcoming the drawbacks of Kashtan *et al.* (2004), where a sampling algorithm to detect network motifs had been proposed, suffering from a sampling bias and scaling poorly with increasing subgraph size. The new approach described in Wernicke (2006) is based on randomized enumeration, and comprises a new way for estimating the frequency of subgraphs in random networks that, in contrast to previous approaches, does not require the explicit generation of random networks.

Chen *et al.* presented *NeMoFinder* (Chen *et al.*, 2006), a network motif discovery algorithm to discover repeated and unique meso-scale network motifs in a large protein-protein interaction network, since many of the relevant processes in biological networks have been shown to correspond to the meso-scale (5 – 25 genes or proteins) (Spirin and Mirny, 2003). The procedure is based on the search of repeated trees, to be exploited for partitioning the input network into a set of graphs, represented by their adjacency matrices. Then, they introduce the concept of *graph cousins* to facilitate the candidate generation and frequency counting processes.

In Grochow and Kellis (Network) an algorithm for discovering large network motifs is presented, based on subgraph query and symmetry-breaking. The size of the considered motifs would exceeds 15, since the exploited a symmetry-breaking technique eliminates repeated isomorphism testing. Such a technique reverses the traditional network-based search at the heart of the algorithm to a motif-based search, which also eliminates the need to store all motifs of a given size and enables parallelization and scaling.

Finally, a tool for the exploration of network motifs, namely, *MAVisto*, is described in Schreiber and Schwabermeyer (2005). Such a tool is based on a flexible motif search algorithm and different views for the analysis and visualization of network motifs, such that the frequency of motif occurrences can be compared with randomized networks, a list of motifs along with information about structure and number of occurrences depending on the reuse of network elements shows potentially interesting motifs, a motif fingerprint reveals the overall distribution of motifs of a given size and the distribution of a particular motif in the network can be visualized.

Approaches Based on Node Colors

The structural motifs defined above treat each component of the biological network as a unique and anonymous entity, ignoring any other useful biological information possibly known about them. In fact, the components are “unlabelled”, allowing for capturing only the topological shapes of the associated subgraphs, but not the biological context in which they occur.

An alternative definition of motif is possible, focusing on the functional nature of the components that form the motif. In such a case, the nodes of the input biological network, as well as the motifs to be searched for, are labelled. Each node label is representative of specific biological properties that can be shared by different nodes in the network. Thus, it is possible to *color* the nodes in such a way that each color is associated to a node class, and nodes with the same color belong to the same class. *Node-colors* motifs may be defined as frequent motifs, where the attention turns to the similarity between the corresponding pairs of nodes composing them.

The first time that colors on nodes have been introduced was in Moon *et al.* (2005), where a colored vertex graph model has been exploited, and different nodes and edges classes are considered. In particular, the class of a node is determined by its color, while the class of an edge is determined by the colors of the two nodes at its ends. The authors focused on protein interaction data, and modelled by nodes in the graph both domains and proteins belonging to such domains. Reflexive edges on domains or proteins indicate that they are self-interacting. Relationships which indicate that some protein has some domain are indicated by dotted lines, while interactions between domains or proteins are indicated by solid lines. The proposed algorithm searches for subgraphs of a given graph whose frequency is substantially higher than that of randomized networks. It first enumerates all of the possible subgraphs, then counts the frequency of each subgraph and finally compares their frequencies with those in randomized graphs. In order to count the frequency of each subgraph efficiently, the authors used a canonical labelling of a graph (McKay, 1978). Two graphs have the same canonically label graph if and only if they are isomorphic to each other.

In Lacroix *et al.* (2006) the authors introduce a definition of motif such that the components of the network play the central part and the topology can be added as a further constraint only. They specialized their approach to metabolic networks, calling the motifs they search for *reaction motifs*, and exploited hierarchical classification of enzymes developed by the International Union of Biochemistry and Molecular Biology (IUBMB) (Webb, 1992) to label the nodes. Then, they work on sets of nodes, instead of subgraphs, escaping in this way the necessity of recurring to subgraphs isomorphism.

In Chen *et al.* (2007) a method called *LaMoFinder* has been presented, to label network motifs with Gene Ontology terms (Asburner *et al.*, 2000) in a protein-protein interaction network. The authors followed the line of Kashtan *et al.* (2004) and Chen *et al.* (2006), first searching for the classes of isomorphic subgraphs that frequently occur in the input network, and then verifying which of

these subgraph classes are also displayed at a much higher frequency than in random graphs. Differently from the previous approaches, Chen *et al.* in [Chen *et al.* \(2007\)](#) exploits a further step, which consists of assigning biological labels to the vertices in the network motifs such that the resulting labelled subgraphs also occur frequently in the underlying labelled input network.

The last approach we discuss here is that presented by Parida in [Parida \(2007\)](#), where an exact three-step approach is presented exploiting the concept of *maximality*, suitably defined from the author in the context of network motifs. In particular, a compact notation has been introduced to handle the combinatorial explosion arising from isomorphisms. Such notation is based on grouping together sets of nodes that can be considered “indistinguishable”. Nodes within the same set, in fact, have the same color, thus each of them can be considered equivalent w.r.t. nodes in another set, and play the same role in the correspondent subgraph topology.

Conclusion

We considered two problems involving the analysis of topology in biological networks: Network clustering, aiming at finding compact subgraphs inside the input graph in order to isolate molecular complexes, and the search of motifs, i.e., sub-structures repeated in the input network and presenting high significance (e.g., in terms of their frequency). We provided a compact overview of the main techniques proposed in the literature to solve these problems.

The last group of techniques we presented for the search of motifs in biological networks, involve approaches that actually are not based on the only topology of the input networks, but they also consider additive information encoded as labels on nodes/edges of the network. This is an important aspect since an emergent trend is the construction of “functional networks” where the information obtained from physical interactions among the cellular components is enriched with functional information, coming from the knowledge of common biological functions of the components, or from their involvement in similar phenotypical effects, such as disorders or diseases. To this respect, also clustering techniques allowing the usage of labels on nodes/edges could be useful, as well as, more in general, analysis techniques able to manage heterogeneous networks, where nodes/edges encoding both components and associations of different types may coexist.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Biological Database Searching. Graphlets and Motifs in Biological Networks. Identification of Proteins from Proteomic Analysis. Mapping the Environmental Microbiome. Molecular Mechanisms Responsible for Drug Resistance. Network Inference and Reconstruction in Bioinformatics. Network-Based Analysis for Biological Discovery. Quantification of Proteins from Proteomic Analysis

References

- Adamcsek, B., *et al.*, 2006. CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* 22 (8), 1021–1023.
- Ahn, Y.-Y., Bagrow, J.P., Lehmann, S., 2010. Link communities reveal multiscale complexity in networks. *Nature* 466, 761–764.
- Altai-UI-Amin, M., *et al.*, 2006. Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics* 7 (207),
- Apostolico, A., Bock, M.E., Lonardi, S., 2003. Monotony of surprise and large-scale quest for unusual words. *Journal of Computational Biology* 10 (2/3), 283–311.
- Apostolico, A., *et al.*, 2008a. Finding 3d motifs in ribosomal RNA structures. *Nucleic Acids Reseach*.
- Apostolico, A., Parida, L., Rombo, S.E., 2008b. Motif patterns in 2D. *Theoretical Computer Science* 390 (1), 40–55.
- Atias, N., Sharan, R., 2012. Comparative analysis of protein networks: Hard problems, practical solutions. *Commun. ACM* 55 (5), 88–97.
- Bader, G., Hogue, H., 2003. An automated method for finding molecular complexes in large protein–protein interaction networks. *BMC Bioinformatics* 4 (2),
- Berg, J., Lassig, M., 2004. Local graph alignment and motif search in biological networks. *Proceedings of the National Academy of Sciences of the United States of America* 101 (41), 14689–14694.
- Cancho, R.F., Janssen, C., Solé, R.V., 2001. Topology of technology graphs: Small world patterns in electronic circuits. *Physical Review E* 64 (4), 046119.
- Chen, J., Hsu, W., Lee, M.L., *et al.*, 2006. NeMoFinder: Dissecting genome-wide protein–protein interactions with meso-scale network motifs. In: *KDD'06*, pp. 106–115.
- Chen, J., Hsu, W., Lee, M.L., *et al.*, 2007. Labeling network motifs in protein interactomes for protein function prediction. In: *ICDE'07*, pp. 546–555.
- Cheng, C.-Y., Huang, C.-Y., Sun, C.-T., 2008. Mining bridge and brick motifs from complex biological networks for functionally and statistically significant discovery. *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 38 (1), 17–24.
- Ciriello, G., Guerra, C., 2008. A review on models and algorithms for motif discovery in protein–protein interaction network. *Briefings in Functional Genomics and Proteomics*.
- Cohen, J., Briand, F., Newman, C., 1990. *Community Food Webs: Data and Theory*. Springer.
- De Virgilio, R., Rombo, S.E., 2012. Approximate matching over biological RDF graphs. In: *Proceedings of the ACM Symposium on Applied Computing*, pp. 1413–1414.
- Derenyi, I., Palla, G., Vicsek, T., 2005. Clique percolation in random networks. *Physical Review Letters* 94 (16), 160–202.
- Enright, A.J., Dongen, S.V., Ouzounis, C.A., 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research* 30 (7), 1575–1584.
- Erdos, P., Renyi, A., 1959. On random graphs. *Publicationes Mathematicae* 6, 290–297.
- Erdos, P., Renyi, A., 1960. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences* 5, 17–61.
- Dobrin, R., *et al.*, 2004. Aggregation of topological motifs in the escherichia coli transcriptional regulatory network. *BMC Bioinformatics* 5, 10.
- Asburner, S., *et al.*, 2000. Gene ontology: Tool for the unification of biology. the gene ontology consortium. *Nature Genetics* 25, 25–29.
- Evans, T.S., Lambiotte, R., 2009. Line graphs, link partitions, and overlapping communities. *Physical Review E* 80 (1), 016105:1–016105:8.
- Evans, T.S., Lambiotte, R., 2010. Line graphs of weighted networks for overlapping communities. *The European Physical Journal B* 77 (2), 265–272.
- Farutin, V., *et al.*, 2006. Edge-count probabilities for the identification of local protein communities and their organization. *Proteins: Structure, Function, and Bioinformatics* 62, 800–818.
- Ferraro, N., Palopoli, L., Panni, S., Rombo, S.E., 2011. Asymmetric comparison and querying of biological networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8, 876–889.
- Fortunato, S., 2010. Community detection in graphs. *Physics Reports* 486, 75–174.

- Furfaro, A., Groccia, M.C., Rombo, S.E., 2017. 2D motif basis applied to the classification of digital images. *Computer Journal* 60 (7), 1096–1109.
- Georgii, E., *et al.*, 2009. Enumeration of condition-dependent dense modules in protein interaction networks. *Bioinformatics* 25 (7), 933–940.
- Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America* 99 (12), 7821–7826.
- Grochow, J., Kellis, M., Network motif discovery using subgraph enumeration and symmetry-breaking.
- Hwang, W., *et al.*, 2006. A novel functional module detection algorithm for protein–protein interaction networks. *Algorithms for Molecular Biology* 1 (24).
- Jain, R.D.A., 1988. *Algorithms for Clustering Data*. Prentice Hall.
- Jancura, P., *et al.*, 2011. A methodology for detecting the orthology signal in a PPI network at a functional complex level. *BMC Bioinformatics*.
- Kashtan, N., Itzkovitz, S., Milo, R., Alon, U., 2004. Topological generalizations of network motifs. *Physical Review E* 70 (3), 031909.
- King, A.D., Przulj, N., Jurisica, I., 2004. Protein complex prediction via cost-based clustering. *Bioinformatics* 20 (17), 3013–3020.
- Kovacs, I.A., *et al.*, 2010. Community landscapes: An integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLOS One* 5 (9).
- Kuchaiev, O., Stevanovic, A., Hayes, W., Przulj, N., 2011. Graphcrush 2: Software tool for network modeling, alignment and clustering. *BMC Bioinformatics* 12, 24.
- Lacroix, V., Fernandes, C.G., Sagot, M.-F., 2006. Motif search in graphs: Application to metabolic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3 (4), 360–368.
- Leskovec, J., Lang, K., Mahoney, M.W., 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the International World Wide Web Conference (WWW)*, pp. 631–640.
- Liu, H., Liu, J., 2006. Clustering protein interaction data through chaotic genetic algorithm. *Simulated Evolution and Learning* 4247, 858–864.
- Macropol, K., Can, T., Singh, A., 2009. RRW: Repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics* 10 (1), 283.
- Madeira, S.C., Oliveira, A.L., 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE Trans. on Comp. Biol. and Bioinf.* 1 (1), 24–45.
- Mangan, S., Alon, U., 2003. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences of the United States of America* 100 (21), 11980–11985.
- Mangan, S., Itzkovitz, S., Zaslaver, A., Alon, U., 2005. The incoherent feed-forward loop accelerates the response-time of the gal system of *Escherichia coli*. *Journal of Molecular Biology* 356 (5), 1073–1081.
- Mazurie, A., Bottani, S., Vergassola, M., 2005. An evolutionary and functional assessment of regulatory network motifs. *Genome Biology* 6, R35.
- McKay, B., 1978. Computing automorphisms and canonical labelling of graphs. *Lecture Notes in Mathematics* 686, 223–232.
- Milenkovic, T., Przulj, N., 2008. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics* 6, 257–273.
- Milo, R., *et al.*, 2002. Network motifs: Simple building blocks of complex networks. *Science* 298 (5594), 824–827.
- Moon, H.S., Bhak, J., Lee, H.K., Lee, D., 2005. Architecture of basic building blocks in protein and domain structural interaction networks. *Bioinformatics* 21 (8), 1479–1486.
- Newman, M.E.J., 2003. The structure and function of complex networks. *SIAM Review* 45 (2), 167–256.
- Palla, G., *et al.*, 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435, 814–818.
- Panni, S., Rombo, S.E., 2015. Searching for repetitions in biological networks: Methods, resources and tools. *Briefings in Bioinformatics* 16 (1), 118–136.
- Parida, L., 2007. Discovering topological motifs using a compact notation. *J. Comp. Biol.* 14 (3), 46–69.
- Parida, L., 2008. *Pattern Discovery in Bioinformatics, Theory and Algorithms*. Chapman and Hall/CRC.
- Parida, L., Pizzi, C., Rombo, S.E., 2014. Irredundant tandem motifs. *Theoretical Computer Science* 525, 89–102.
- Park, Y.J., Song, M.S., 1989. A genetic algorithm for clustering problems. In: *Proceedings of 3rd Annual Conference on Genetic Algorithms*, pp. 2–9.
- Pereira, J.B., Enright, A.J., Ouzounis, C.A., 2004. Detection of functional modules from protein interaction networks. *Proteins: Structure, Functions, and Bioinformatics* 20), 49–57.
- Pizzuti, C., 2008. GA-NET: A genetic algorithm for community detection in social networks. In: *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, pp. 1081–1090.
- Pizzuti, C., 2009. Overlapped community detection in complex networks. In: *Proceedings of the 11th Annual conference on Genetic and Evolutionary computation, GECCO '09*, pp. 859–866.
- Pizzuti, C., Rombo, S.E., 2007. Pinoc: A co-clustering based approach to analyze protein–protein interaction networks. In: *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, pp. 821–830.
- Pizzuti, C., Rombo, S.E., 2008. Multi-functional protein clustering in ppi networks. In: *Proceedings of the 2nd International Conference on Bioinformatics Research and Development (BIRD)*, pp. 318–330.
- Pizzuti, C., Rombo S.E., 2012. Experimental evaluation of topological-based fitness functions to detect complexes in PPI networks. In: *Genetic and Evolutionary Computation Conference (GECCO 2012)*, pp. 193–200.
- Pizzuti, C., Rombo, S.E., 2014a. Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods. *Bioinformatics* 30 (10), 1343–1352.
- Pizzuti, C., Rombo, S.E., 2014b. An evolutionary restricted neighborhood search clustering approach for PPI networks. *Neurocomputing* 145, 53–61.
- Pizzuti, C., Rombo, S.E., Marchiori, E., 2012. Complex detection in protein-protein interaction networks: A compact overview for researchers and practitioners. In: *10th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology (EvoBio 2012)*, pages 211–223.
- Prill, R.J., Iglesias, P.A., Levchenko, A., 2005. Dynamic properties of network motifs contribute to biological network organization. *PLOS Biology* 3 (11), e343.
- Przulj, N., 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics* 23 (2), 177–183.
- Ravaee, H., Masoudi-Nejad, A., Omid, S., Moeini, A., 2010. Improved immune genetic algorithm for clustering protein-protein interaction network. In: *Proceedings of the 2010 IEEE International Conference on Bioinformatics and Bioengineering*, pp. 174–179.
- Ruan, J., Zhang, W., 2008. Identifying network communities with a high resolution. *Physical Review E* 77 (1).
- Samantha, M.P., Liang, S., 2003. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proceedings of the National Academy of Sciences of the United States of America* 100 (22), 12579–12583.
- Schreiber, F., Schwabermeyer, H., 2005. MAVisto: A tool for the exploration of network motifs. *Bioinformatics* 21 (17), 3572–3574.
- Sharan, R., Ulitsky, I., Shamir, R., 2007. Network-based prediction of protein function. *Molecular Systems Biology* 3 (88).
- Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U., 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature* 31, 64–68.
- Solava, R., Michaels, R.P., Milenkovic, T., 2012. Graphlet-based edge clustering reveals pathogen-interacting proteins. *Bioinformatics* 28 (18), 480–486.
- Spirin, V., Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America* 100 (21), 12123–12128.
- Tasgin, M., Bingol, H., 2007. Community detection in complex networks using genetic algorithm. *arXiv:0711.0491*.
- Webb, E.C., 1992. *Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. Oxford University Press.
- Wernicke, S., 2006. Efficient detection of network motifs. *IEEE/ACM Transactions on Computational Biology And Bioinformatics* 3 (4), 347–359.
- Yeger-Lotem, E., *et al.*, 2004. Network motifs in integrated cellular networks of transcription regulation and protein-protein interaction. *Proceedings of the National Academy of Sciences of the United States of America* 101 (16), 5934–5939.

Algorithms for Graph and Network Analysis: Graph Alignment

Luigi Palopoli, Università della Calabria, Cosenza, Italy

Simona E Rombo, Università degli studi di Palermo, Palermo, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

An important problem in biological network analysis is the comparison of different input networks, usually modelling the physical interactions among cellular components. Biological networks are represented by undirect or direct graphs. For example, protein-protein interaction networks are represented by undirect graphs, since the roles of two interacting proteins linked by an edge in the network are supposed to be equivalent, whereas direct graphs are associated with metabolic networks since chemical reactions have specified directions. However, graph alignment has been considered mainly in the context of undirect graphs, therefore we will refer to undirect graphs in the following if not differently specified.

Graph alignment aims at finding conserved portions of two or several input networks and has been applied to solve several problems involving biological networks. For instance, the alignment of biological networks associated with different organisms can be useful to uncover complex mechanisms at the basis of evolutionary conservations, or to infer the biological meaning of groups of interacting cellular components belonging to organisms not yet well characterized (Sharan and Ideker, 2006). Indeed, network alignment can guide the transfer of biological knowledge from model species to less studied species by highlighting conservation between network regions, thus complementing valuable insights provided by genomic sequence alignment (Faisal *et al.*, 2015).

In this manuscript, we illustrate how graph alignment has been defined in the literature, by distinguishing local alignment, global alignment and by also discussing network querying, that is, a specific instance of graph alignment where a small graph is searched for in another one (Section Ways to formulate Graph Alignment). In Section Algorithms for Graph Alignment we provide a comprehensive overview of the algorithms and techniques proposed in the literature to solve each of the specific considered types of graph alignment. Some of the available software tools implementing the techniques proposed in the literature are illustrated in Section Available Software Tools. A working example is provided in Section A Working Example in order to help understanding the application of the available algorithms for graph alignment. Finally, in Section Conclusion and Open Challenges, we discuss the main emerging research directions on this topic and we draw our conclusions.

The interested reader can find other surveys concerning specific aspects of problems involving graph alignment in Alon (2007), Fionda and Palopoli (2011), Hiram Guzzi and Milenkovic (2017), Panni and Rombo (2015), Sharan and Ideker (2006), and Zhang *et al.* (2008).

Ways to Formulate Graph Alignment

In this section we describe how graph alignment can be defined, according to different formulations of the problem. As already pointed out in the Introduction, we suppose that the input networks are always modelled by undirect graphs, since most of the alignment techniques refer to protein-protein interaction networks. Usually, some criterion is established in order to understand if two nodes (e.g., two proteins) are “similar” and then they can be paired during the alignment process. Different ways to compute node similarity can be adopted: for instance, vocabularies from the Gene Ontology (Ashburner *et al.*, 2000) may be used in order to understand the functional similarity of the cellular components associated with the nodes in the input networks, or the corresponding primary structures (e.g., amino acidic sequences) are aligned and the scores returned by the sequence alignment (e.g., BLAST score (Altschul *et al.*, 1997)) are used to measure the similarity between nodes.

Pairwise and Multiple Alignment

Let N_1 and N_2 be two input networks. The alignment problem consists in finding a set of conserved edges across N_1 and N_2 , leading to a (not necessarily connected) conserved subgraph occurring in both the input networks. In this case, the problem is referred to as *pairwise alignment*. *Multiple alignment* is an extension of pairwise alignment where a set of networks $N_1, \dots, N_n$ is considered in input, and it is usually computationally more difficult to be solved. Many of the algorithms proposed for pairwise graph alignment extend also to multiple alignment. For this reason, in the following subsections, we illustrate the graph alignment formulations by focusing on pairwise alignment.

Global Alignment

Given two graphs N_1 and N_2 in input, the aim of global alignment is that of superimposing them in such a way that a matching score involving both nodes and subgraph topology is maximized.

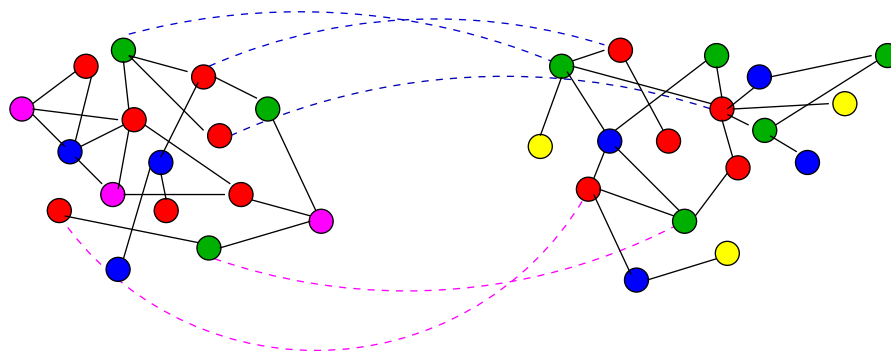


Fig. 1 Example of global alignment.

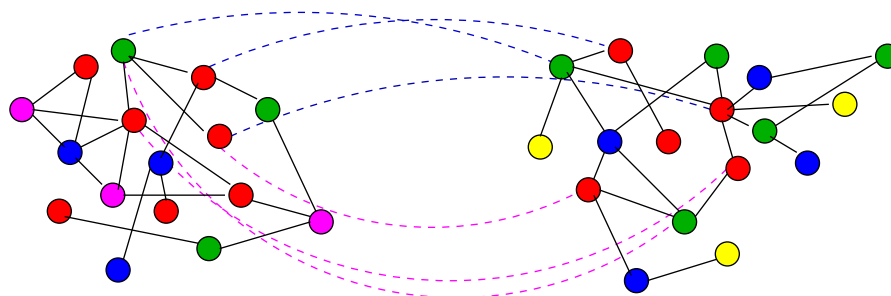


Fig. 2 Example of local alignment.

Global alignment returns a unique (possibly, the best one) overall alignment between N_1 and N_2 , in such a way that a one-to-one correspondence is found between nodes in N_1 and nodes in N_2 . The result is made of a set of pairs of non overlapping subgraphs of N_1 and N_2 .

Fig. 1 illustrates an example of global alignment between two simple graphs.

Local Alignment

Local alignment aims at finding multiple, unrelated regions of isomorphism among the input networks, each region implying a mapping independently of the others. Therefore, the computed correspondences may involve overlapping subgraphs.

The output of local network alignment is a set of pairs of (possibly) overlapping subgraphs of the input networks, as illustrated in **Fig. 2**.

Local network alignment may be applied to search for a known functional component, for example, pathways, complexes, etc., in a new species.

A Special Case of Local Alignment: Graph Querying

In some contexts, it may be useful searching for the “occurrences” of a specific, usually small, graph into another bigger graph. A typical application is studying how a specific module of a model organism differentiated in more complex organisms (Ferraro *et al.*, 2011). More in general, this problem “is aimed at transferring biological knowledge within and across species”, (Sharan and Ideker, 2006) since the result subnetworks may correspond to cellular components involved in the same biological processes or performing similar functions than the components in the query. Actually, it can be viewed as a specific formulation of local graph alignment, where one of the input networks is much smaller than the other ones.

In more detail, graph querying consists of analyzing an input network, called *target graph*, searching for subgraphs similar to a *query graph* of interest (see **Fig. 3**).

Algorithms for Graph Alignment

Graph alignment involves the problem of subgraph isomorphism checking, that is known to be NP-complete (Garey and Johnson, 1979). Therefore, the techniques proposed in the literature are often based on approximate and heuristic algorithms. Moreover, it is worth pointing out that in most cases the input networks are of the same kind, e.g., protein-protein interaction networks.

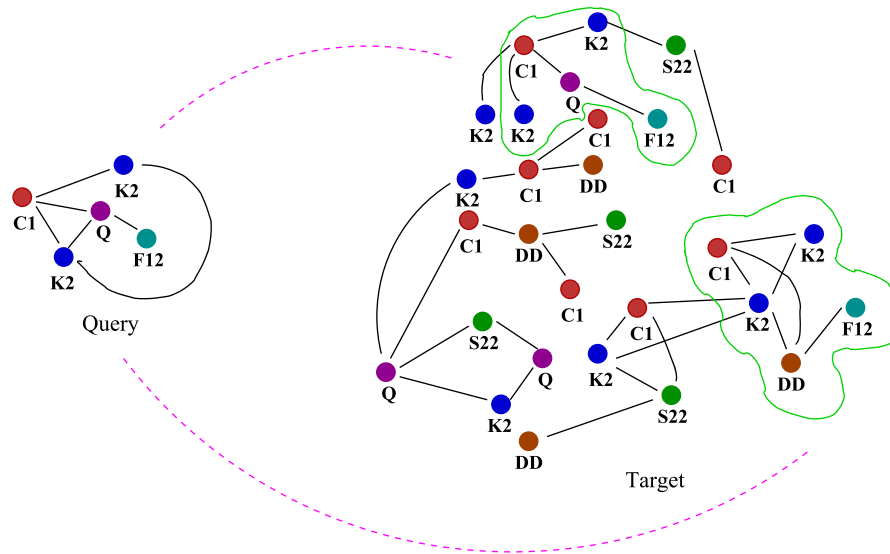


Fig. 3 Example of graph querying.

However, graph alignment can be approached also if the input networks are of different types, leading to a kind of *heterogeneous* alignment. Usually, in these cases, the two input networks are merged and statistical approaches are then applied to extract the most significant subgraphs from the integrated network (Wu *et al.*, 2009).

In the following subsections, we provide an overview of the main algorithms and techniques proposed in the literature to solve the different formulations of graph alignment referred to in the previous section.

Global Alignment

Singh *et al.* (2007) present *IsoRank*, an algorithm for pairwise global alignment of protein-protein interaction networks working in two stages: first it associates a score with each possible match between nodes of the two networks, and then it constructs the mapping for the global network alignment by extracting mutually-consistent matches according to a bipartite graph weighted matching performed on the two entire networks. *IsoRank* has been extended in Singh *et al.* (2008) to perform multiple alignment by approximate multipartite graph weighted matching. In Liao *et al.* (2009) the *IsoRankN* (IsoRank-Nibble) tool is proposed, that is, a global multiple-network alignment tool based on spectral clustering on the induced graph of pairwise alignment scores. In Klau (2009) a graph-based maximum structural matching formulation for pairwise global network alignment is introduced, combining a Lagrangian relaxation approach with a branch-and-bound method. MI-GRAAL (Kuchaiev and Przulj, 2011) can integrate any number and type of similarity measures between network nodes (e.g., sequence similarity, functional similarity, etc.) and finds a combination of similarity measures yielding the largest contiguous (i.e. connected) alignments. In Shih and Parthasarathy (2012) a scalable algorithm for multiple alignment is presented based on clustering methods and graph matching techniques to detect conserved interactions while simultaneously attempting to maximize the sequence similarity of nodes involved in the alignment. Finally, in Mongiov and Sharan (2013) an evolutionary-based global alignment algorithm is proposed, while in Neyshabur *et al.* (2013) a greedy method is used, based on an alignment scoring matrix derived from both biological and topological information about the input networks to find the best global network alignment. *ABiNet* (Ferraro *et al.*, 2010; Ferraro *et al.*, 2011) is an algorithm performing *asymmetric alignment*. In particular, given two input networks, the one associated with the best characterized organism (called *Master*) is exploited as a fingerprint to guide the alignment process to the second input network (called *Slave*), so that generated results preferably retain the structural characteristics of the Master network. Technically, this is obtained by generating from the Master a finite automaton, called *alignment model*, which is then fed with a (linearization of) the Slave for the purpose of extracting, via the Viterbi algorithm, matching subgraphs. *ABiNet* performs both querying and global alignment.

Local Alignment

Kelley *et al.* (2004) propose *PathBLAST*, that is, a procedure for pairwise alignment combining interaction topology and protein sequence similarity. They search for high scoring pathway alignments involving two paths, one for each network, in which proteins of the first path are paired with putative homologs occurring in the same order in the second path. *PathBLAST* is extended in Sharan *et al.* (2005) for multiple alignment, based on the generation of a network alignment graph where each node consists of a group of sequence-similar proteins, one for each species, and each link between a pair of nodes represents a conserved protein interaction between the corresponding protein groups. *PathBLAST* has also been used in Bandyopadhyay *et al.* (2006) to resolve

ambiguous functional orthology relationships in protein-protein interaction networks. In, [Koyuturk et al. \(2006\)](#) a technique for pairwise alignment is proposed based on duplication/divergence models and on efficient heuristics to solve a graph optimization problem. *Bi-GRAPPIN* ([Fionda et al., 2009a,b](#)) is based on maximum weight matching of bipartite graphs, resulting from comparing the adjacent nodes of pairs of proteins occurring in the input networks. The idea is that proteins belonging to different networks should be matched looking not only at their own sequence similarity but also at the similarity of proteins they significantly interact with. *Bi-GRAPPIN* allows for the exploitation of both quantitative and reliability information possibly available about protein interactions, thus making the analysis more accurate. *Bi-GRAPPIN* has been exploited in, [Fionda et al. \(2009a,b\)](#) as a preliminary step to apply a node-collapsing technique to extract similar subgraphs from two input networks. In [Flannick et al. \(2006\)](#) an algorithm for multiple alignment, named *Graemlin*, is presented. *Graemlin* aligns an arbitrary number of networks to individuate conserved functional modules, greedily assigning the aligned proteins to non-overlapping homology classes and progressively aligning multiple input networks. The algorithm also allows searching for different conserved topologies defined by the user. It can be used to either generate an exhaustive list of conserved modules in a set of networks (network-to-network alignment) or find matches to a particular module within a database of interaction networks (query-to-network alignment). In [Denielou et al. \(2009\)](#) the algorithm *C3Part-M*, based on a non-heuristic approach exploiting a correspondence multigraph formalism to extract connected components conserved in multiple networks, is presented and compared with *NetworkBlast-M*, ([Kalaev et al., 2008](#)) another technique recently proposed based on a novel representation of multiple networks that is linear in their size. *NetworkBlast-M* can align 10 networks with tens of thousands of proteins in few minutes. The two latter approaches represent the most efficient techniques proposed in the literature for local multiple alignment. Finally, *AlignNemo* ([Ciriello et al., 2012](#)) builds a weighted alignment graph from the input networks, extracts all connected subgraphs of a given size from the alignment graph and use them as seeds for the alignment solution, by expanding each seed in an iterative fashion.

Graph Querying

Network querying approaches may be divided in two main categories: those searching for efficient solutions under particular conditions, e.g., the query is not a general graph but it is a path or a tree, and other approaches where the query is a specific small graph in input, often representing a functional module of another well characterized organism. *MetaPathwayHunter* ([Pinter et al., 2005](#)) is an algorithm for querying metabolic networks by multi-source trees, that are directed acyclic graphs whose corresponding undirected graphs are trees where nodes may present both incoming and outgoing edges. *MetaPathwayHunter* searches the networks for approximated matching, allowing node insertions (only one node), whereas no deletions are allowed. In [Shlomi et al. \(2006\)](#) and [Dost et al. \(2007\)](#) *QPath* and *QNet* are presented, respectively. *QPath* queries a protein-protein interaction network by a query pathway consisting of a linear chain of interacting proteins belonging to another organism. The algorithm works similarly to sequence alignment, by aligning the query pathway to putative pathways in the target network, so that proteins in analogous positions have similar sequences. Protein-protein interaction networks interactions reliability scores are used, and insertions and deletions are allowed. *QNet* is an extension of *QPath* in which the queries are trees or graphs with limited treewidth. *GenoLink* ([Durand et al., 2006](#)) is a system able to integrate data from different sources (e.g., databases of proteins, genes, organisms, chromosomes) and to query the resulting data graph by graph patterns with constraints attached to both vertices and edges; a query result is the set of all the subgraphs of the target graph that are much similar to the query pattern and satisfy the constraints. In [Yang and Sze \(2007\)](#) the two problems of path matching and graph matching are considered. An exact algorithm called SAGA is presented to search for subgraphs of arbitrary structure in a large graph, grouping related vertices in the target network for each vertex in the query. Although the algorithm is accurate and also relatively efficient to be an exact one, the authors state that it is practical for queries having a number of nodes as large as 20, and its performances improve if the query is a sparse graph. *NetMatch* ([Ferro et al., 2007](#)) is a Cytoscape plugin allowing for approximated queries that come in the form of graphs where some nodes are specified and others are wildcards (which can match an unspecified number of elements). *NetMatch* captures the topological similarity between the query and target graphs, without taking into account any information about node similarities. In [Fionda et al. \(2008\)](#) protein-protein interaction networks are modelled using labelled graphs in order to taken into account interactions reliability, allowing for a rather accurate analysis, and a technique is proposed based on maximum weight matching of bipartite graphs. *Torque* ([Bruckner et al., 2009](#)) is an algorithm based on dynamic programming and integer linear programming to search for a matching set of proteins that are sequence-similar to the query proteins, by relaxing the topology constraints of the query. Finally, we note that, sometimes, methods for local alignment can be also successfully exploited to perform network querying, for example, [Ferraro et al. \(2011\)](#) [Kelley et al. \(2004\)](#) and [Koyuturk et al., 2006](#).

A Working Example

We now illustrate how the alignment tools work by discussing a simple example involving two networks. In particular, we focus on global alignment, and we consider two different approaches: one of the most popular, that is, *IsoRankN*, ([Liao et al., 2009](#)) and one performing asymmetric alignment, that is, *AbiNet* ([Ferraro et al., 2011](#)). The working example reported should allow the reader to understand how the available software tools can be exploited and, moreover, it provides some explanation of the main differences between asymmetric and symmetric alignment.

First of all, it is worth pointing out that interaction data are usually stored in a MiTab format, (Hermjakob *et al.*, 2004) that is a tab-separated text file where each column is associated with specific information, such as the id of interactors, the gene names corresponding to interactors, the experiments in which the interaction was demonstrated, etc. From this file, it is possible to select the information needed to build the networks to be fed to the alignment tools. For both the considered software, the input network can be stored in a tab-separated text file containing only the two columns associated with the interactors. A further file is needed, that is, a *basic dictionary*, storing the similarity score between pairs of nodes in input networks. We assume that the BLAST bit score is used as similarity and we also suppose that only similarity values satisfying a given threshold are kept in the dictionary.

Consider the two input networks Net_1 in Fig. 4(a) and Net_2 in Fig. 4(b). In Fig. 4(c) the input dictionary is reported (where the similarity threshold is set to 40.00).

We run AbiNet by setting first Net_1 and then Net_2 as the Master, and the associations between proteins in the two input networks resulting from the global alignment are shown in Fig. 5(a) and (b), respectively. Then, we run IsoRankN on the same networks and the result is shown in Fig. 5(c).

First of all, note that the three alignments share a central core made of three associations $((p_3, q_3), (p_5, q_5) \text{ and } (p_6, q_6))$, highlighted in bold in Fig. 5. Such associations correspond to a high conservation w.r.t. both protein basic similarity and topology, and this is the reason why they are intercepted in both versions of our Master-Slave alignment and also in the symmetric alignment carried out by IsoRank.

Let us now turn to the differences shown in the three alignment results. Consider the nodes p_2 and p_8 in Net_1 and the node q_2 in Net_2 . Node p_8 has a higher basic similarity to q_2 than p_2 , but p_2 is involved in a larger number of interactions that are *topologically similar* to those involving q_2 . By “topologically similar” we mean interactions in two different networks that involve pairs of proteins with a mutual basic similarity higher than the fixed threshold. As an example, the interaction between p_3 and p_5 in Net_1 can be considered topologically similar to the interaction between q_3 and q_5 in Net_2 . When Net_1 is the Master, the topology of Net_1 is almost completely kept, thus AbiNet associates p_2 with q_2 . On the contrary, when the Master is Net_2 , then the topology around p_2 and p_8 is flattened, and AbiNet associates q_2 with p_8 instead of p_2 , since these two nodes share a higher basic similarity. Looking at all the other discrepancies between the two alignments returned by AbiNet, it is easy to see that the situation is analogous to the

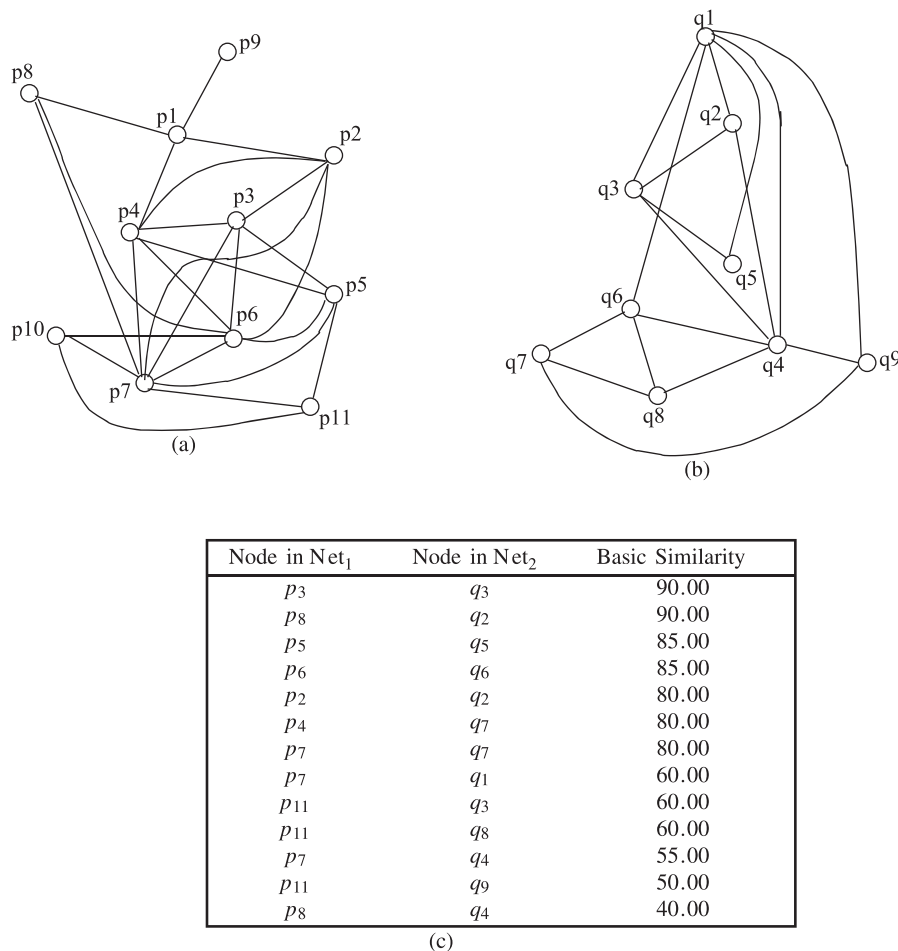


Fig. 4 (a) The input network Net_1 . (b) The input network Net_2 . (c) The input dictionary of protein basic similarities.

Node in Net_1	Node in Net_2	Node in Net_1	Node in Net_2
p_2	q_2	p_3	q_3
p_3	q_3	p_8	q_2
p_5	q_5	p_5	q_5
p_6	q_6	p_6	q_6
p_7	q_7	p_7	q_1
p_{11}	q_8	p_{11}	q_9
		p_4	q_7

(a)

Node in Net_1	Node in Net_2
p_3	q_3
p_8	q_2
p_5	q_5
p_6	q_6
p_7	q_7
p_{11}	q_8 / q_9

(b)

Node in Net_1	Node in Net_2
p_3	q_3
p_8	q_2
p_5	q_5
p_6	q_6
p_7	q_7
p_{11}	q_8 / q_9

(c)

Fig. 5 The global alignments returned by: (a) AbiNet, when Net_1 is the Master. (b) AbiNet, when Net_2 is the Master. (c) IsoRankN. Pairs in bold are common to all the three alignments.

Table 1 List of the publicly available software tools implementing the considered techniques for graph alignment

Method	Year	Problem	Software
PathBLAST (Kelley <i>et al.</i> , 2004)	2003	LA, GQ	http://www.pathblast.org/
MetaPathwayHunter (Pinter <i>et al.</i> , 2005)	2005	GQ	http://www.cs.technion.ac.il/olegro/metapathwayhunter/
NetworkBLAST (Sharan <i>et al.</i> , 2005)	2005	LA	http://www.cs.tau.ac.il/~bnet/networkblast.htm
Graemlin (Flannick <i>et al.</i> , 2006)	2006	LA	http://graemlin.stanford.edu/
NetMatch (Ferro <i>et al.</i> , 2007)	2007	GQ	http://ferrolab.dmi.unict.it/netmatch.html
IsoRank (Singh <i>et al.</i> , 2007)	2007	GA	http://groups.csail.mit.edu/cb/mna/
SAGA (Yang and Sze, 2007)	2007	GQ	http://www.eecs.umich.edu/saga
Torque (Bruckner <i>et al.</i> , 2009)	2009	GQ	http://www.cs.tau.ac.il/~bnet/torque.html
IsoRankN (Liao <i>et al.</i> , 2009)	2009	GA	http://groups.csail.mit.edu/cb/mna/
NATALIE (Klau, 2009)	2009	GA	http://www.mi.fu-berlin.de/w/LiSA/Natalie
C3Part-M (Denielou <i>et al.</i> , 2009)	2009	LA	http://www.inrialpes.fr/helix/people/viari/lxgraph/
AbiNet (Ferraro <i>et al.</i> , 2011)	2010	GA, GQ	http://siloe.deis.unical.it/ABiNet/
MI-GRAAL (Kuchaiev and Przulj, 2011)	2011	GA	http://bio-nets.doc.ic.ac.uk/MI-GRAAL/
AlignNemo (Ciriello <i>et al.</i> , 2012)	2012	LA	http://www.bioinformatics.org/alignnemo
NETAL (Neyshabur <i>et al.</i> , 2013)	2013	GA	http://www.bioinf.cs.ipm.ir/software/netal

one we just described. This confirms what we expected, that is, the network exploited as the Master “guides” the alignment. In particular, given a node of the Master, the topology around it is kept and it is associated to a node in the Slave sharing both a high basic similarity and some topologically similar interactions. Since the Slave is linearized, then less importance is given to the topology around the Slave candidate nodes and this can influence the final result. Obviously, there are cases for which these differences are immaterial (as for nodes in the core of Fig. 5), and the same associations are anyway returned.

As for the result returned by IsoRankN, we observe that it agrees with that of the first execution of AbiNet for the association (p_7, q_7), while it agrees with the second execution of AbiNet for the association (p_8, q_2). Furthermore, according to IsoRankN, node p_{11} can be equally associated with q_8 and with q_9 , while AbiNet associates p_{11} with q_8 when Net_1 is the Master and with q_9 when the Master is Net_2 . Finally, we observe that, in both executions, AbiNet is able to arrange at least one node more than IsoRankN (p_2 when Net_1 is the Master and q_1, p_4 when the Master is Net_2). In conclusion, a symmetric alignment can be viewed in part as a “mix” between the two asymmetric ones but, in this case, separating what of a network is more conserved in the other one is not easy.

Available Software Tools

In Table 1, the main software tools implementing the techniques proposed in the literature for graph alignment and discussed in the previous section are summarized. In particular, for each method it is specified the year in which it has been published, the

specific formulation of graph alignment it refers to (we denoted global alignment by “GA”, local alignment by “LA” and graph querying by “GQ”) and the web link where it is possible to access it.

Conclusion and Open Challenges

In this manuscript, we illustrated the problem of graph alignment, and the algorithms for its solution according to different formulations, by referring to the context of biological networks.

Looking at the number of techniques proposed in the literature to solve graph alignment, we can conclude that the problem has been now well studied and an analyst who needs to perform the alignment of biological network data can use several software tools that are publicly available.

However, biological networks intrinsically suffer of the difficulty in collecting an adequate amount of interaction data, therefore the biological results obtained by the alignment between different networks are seriously affected by the partiality of the information given in input. Very few model organisms have been extensively characterized and, even for them, the available interaction networks are far to be complete. Furthermore, to increase the coverage, methods used to reveal interactions have been automated to generate high throughput approaches, which, unfortunately, produce a significant fraction of false positives and reduce the accuracy of the data (von Mering *et al.*, 2002).

In this context, an interesting task would be that of applying techniques in order to both clean and make more accurate the available interaction networks. To this aim, the reliability scores provided by some of the interaction databases might be used, as well as additional information coming from curated vocabularies (e.g., the Gene Ontology (Ashburner *et al.*, 2000)).

A further open challenge is the alignment of “functional” networks, obtained as the integration of information coming from both physical interactions (e.g., protein-protein interactions) and functional annotations (e.g., coming from the Gene Ontology, or from genotype-phenotype associations). In this case the difficulty would be given not only by the heterogeneity of the input networks, which are made of nodes and edges of different types, but also by the very large sizes of the input networks, requiring the application of suitable classification and/or compression techniques, (Furfaro *et al.*, 2017; Hayashida and Tatsuya, 2010) or of big data technologies, that are not yet largely exploited for graph alignment.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Algorithms Foundations. Algorithms Foundations. Graphlets and Motifs in Biological Networks. Network-Based Analysis for Biological Discovery

References

- Alon, U., 2007. Network motifs: Theory and experimental approaches. *Nature* 8, 450–461.
- Altschul, S.F., Madden, T.L., Schaffer, A.A., *et al.*, 1997. Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Reserch* 25 (17), 3389–3402.
- Ashburner, M., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25 (1), 25–29.
- Bandyopadhyay, S., Sharan, R., Ideker, T., 2006. Systematic identification of functional orthologs based on protein network comparison. *Genome Research* 16 (3), 428–435.
- Bruckner, S., Huffner, F., Karp, R.M., Shamir, R., Sharan, R., 2009. Torque: Topology-free querying of protein interaction networks. *Nucleic Acids Research* 37 (Web-Server-Issue), 106–108.
- Ciriello, G., Mina, M., Guzzi, P.H., Cannataro, M., Guerra, C., 2012. AlignNemo: A local network alignment method to integrate homology and topology. *PLOS One* 7 (6), e38107.
- Denielou, Y.-P., Boyer, F., Viari, A., Sagot, M.-F., 2009. Multiple alignment of biological networks: A flexible approach. In: *CPM'09*.
- Dost, B., *et al.*, 2007. Qnet: A tool for querying protein interaction networks. In: *RECOMB'07*, pp. 1–15.
- Durand, P., Labarre, L., Meil, A., *et al.*, 2006. Genolink: A graph-based querying and browsing system for investigating the function of genes and proteins. *BMC Bioinformatics* 21 (7).
- Faisal, F.E., Meng, L., Crawford, J., Milenkovic, T., 2015. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology* 3.
- Ferraro, N., Palopoli, L., Panni, S., Rombo, S.E., 2010. Master-slave biological network alignment. In: *6th International symposium on Bioinformatics Research and Applications (ISBRA 2010)*, pp. 215–229.
- Ferraro, N., Palopoli, L., Panni, S., Rombo, S.E., 2011. Asymmetric comparison and querying of biological networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8, 876–889.
- Ferro, A., *et al.*, 2007. Netmatch: A cytoscape plugin for searching biological networks. *Bioinformatics*.
- Fionda, V., Palopoli, L., 2011. Biological network querying techniques: Analysis and comparison. *Journal of Computational Biology* 18 (4), 595–625.
- Fionda, V., Palopoli, L., Panni, S., Rombo, S.E., 2008. Protein–protein interaction network querying by a focus and zoom approach. In: *BIRD'08*, pp. 331–46.
- Fionda, V., Panni, S., Palopoli, L., Rombo, S.E., 2009a. Extracting similar sub-graphs across ppi networks. In: *ISCIS'09*.
- Fionda, V., Panni, S., Palopoli, L., Rombo, S.E., 2009b. A technique to search functional similarities in ppi networks. *International Journal of Data Mining and Bioinformation* 3.
- Flannick, J., *et al.*, 2006. Graemlin: General and robust alignment of multiple large interaction networks. *Genome Research* 16 (9), 1169–1181.
- Furfaro, A., Groccia, M.C., Rombo, S.E., 2017. 2D motif basis applied to the classification of digital images. *Computer Journal* 60 (7), 1096–1109.
- Garey, M., Johnson, D., 1979. *Computers and intractability: A guide to the theory of NP-completeness*. New York: Freeman.
- Hayashida, M., Tatsuya, A., 2010. Comparing biological networks via graph compression. *BMC Systems Biology* 4 (Suppl 2), S13.
- Hermjakob, H., *et al.*, 2004. The HUP0 PSI's molecular interaction format – A community standard for the representation of protein interaction data. *Nature Biotechnology* 22 (2), 177–183.

- Hiram Guzzi, P., Milenkovic, T., 2017. Survey of local and global biological network alignment: The need to reconcile the two sides of the same coin. *Briefings in Bioinformatics*.
- Kalaei, M., Bafna, V., Sharan, R., 2008. Fast and accurate alignment of multiple protein networks. In: RECOMB'08.
- Kelley, B.P., *et al.*, 2004. Pathblast: A tool for alignment of protein interaction networks. *Nucleic Acid Research* 32, W83–W88.
- Klau, G.W., 2009. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics* 10 (Suppl. 1), S59.
- Koyuturk, M., *et al.*, 2006. Pairwise alignment of protein interaction networks. *Journal of Computer Biology* 13 (2), 182–199.
- Kuchaiev, O., Przulj, N., 2011. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics* 27 (10), 1390–1396.
- Liao, C.-S., *et al.*, 2009. Isorankn: Spectral methods for global alignment of multiple protein networks. *Bioinformatics* 25, i253–i258.
- Mongiov, M., Sharan, R., 2013. Global alignment of protein–protein interaction networks. In: Mamitsuka, H., DeLisi, C., Kanehisa, M. (Eds.), *Data Mining for Systems Biology*, vol. 939 of *Methods in Molecular Biology*. Humana Press, pp. 21–34.
- Neyshabur, B., Khadem1, A., Hashemifar, S., Arab, S.S., 2013. NETAL: A new graph-based method for global alignment of protein? Protein interaction networks. *Bioinformatics* 29 (13), 11654–11662.
- Panni, S., Rombo, S.E., 2015. Searching for repetitions in biological networks: Methods, resources and tools. *Briefings in Bioinformatics* 16 (1), 118–136.
- Pinter, R., *et al.*, 2005. Alignment of metabolic pathways. *Bioinformatics* 21 (16), 3401–3408.
- Sharan, R., *et al.*, 2005. From the cover: Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America* 102 (6), 1974–1979.
- Sharan, R., Ideker, T., 2006. Modeling cellular machinery through biological network comparison. *Nature Biotechnology* 24 (4), 427–433.
- Shih, Y.-K., Parthasarathy, S., 2012. Scalable global alignment for multiple biological networks. *BMC Bioinformatics* 13 (Suppl. 3), S11.
- Shlomi, T., *et al.*, 2006. Qpath: A method for querying pathways in a protein–protein interaction network. *BMC Bioinformatics* 7.
- Singh, R., Xu, J., Berger, B., 2007. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In: RECOMB'07.
- Singh, R., Xu, J., Berger, B., 2008. Global alignment of multiple protein interaction networks. In: PSB'08.
- von Mering, D., Krause, C., *et al.*, 2002. Comparative assessment of a large-scale data sets of protein–protein interactions. *Nature* 417 (6887), 399–403.
- Wu, X., Liu, Q., Jiang, R., 2009. Align human interactome with phenome to identify causative genes and networks underlying disease families. *Bioinformatics* 25 (1), 98–104.
- Yang, Q., Sze, S.-H., 2007. Saga: A subgraph matching tool for biological graphs. *Journal of Computational Biology* 14 (1), 56–67.
- Zhang, S., Zhang, X.-S., Chen, L., 2008. Biomolecular network querying: A promising approach in systems biology. *BMC System Biology* 2, 5.

Bioinformatics Data Models, Representation and Storage

Mariaconcetta Bilotta, University of Catanzaro, Catanzaro, Italy and Institute S. Anna of Crotona, Crotona, Italy

Giuseppe Tradigo, University of Calabria, Rende, Italy and University of Florida, Gainesville, United States

Pierangelo Veltri, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A big amount of data is generated nowadays and consequently the necessity of storing and querying them is impelling. Bioinformatics can be defined as “the application of computational tools to organize, analyze, understand, visualize and store information associated with biological macromolecules” (Luscombe *et al.*, 2001; Pevsner, 2015). Three main perspectives defined from Pevsner (2015) about the field of bioinformatics and genomics are:

- The cell and the central dogma of molecular biology;
- The organism, which shows changes between the different stages of development and regions of the body;
- The tree of life, in which millions of species are grouped into three evolutionary branches. A computational view is presented by Luscombe (Luscombe *et al.*, 2001).

Goals of bioinformatics are to organize data so that researchers can access the information and create new entries.

- To develop tools and resources that help in the data analysis;
- To use these tools to analyze data and interpret them significantly.

Finally, the issues involved in bioinformatics (Fig. 1) can be classified into two classes: The first related to sequences and the second related to biomolecular structures (Diniz and Canduri, 2017).

Structural Bioinformatics

Several bioinformatics management systems exist and are usually classified according to the type of data (e.g., proteins, genes, transcriptome). Structural bioinformatics databases offer relevant features for the analysis of available information about particular biomacromolecules, for example, their 3D structure, sequence variations, function annotation, intrinsic flexibility, ligand binding cavity identification, interactions with ligands, membrane and subcellular localization (Koča *et al.*, 2016). Protein structural bioinformatics studies the role of the protein in structural, enzymatic, transport, and regulatory functions in the cell. Protein functions are implied by their structures:

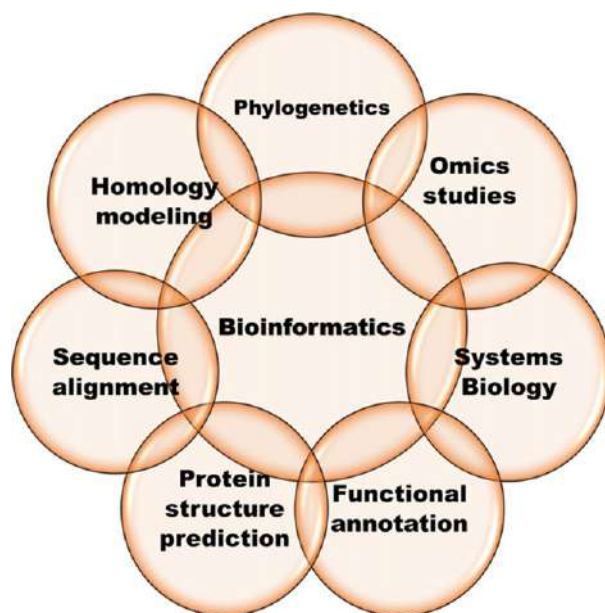


Fig. 1 Some of the bioinformatics applications. Figure modified from Diniz, W.J.S, Canduri, F., 2017. Bioinformatics: An overview and its applications. *Genetics and Molecular Research* 16 (1). (gmr16019645).

- Primary structures, arising from the sequence of amino acids residues;
- Secondary structures, (α -helices and β -sheets) that are the repeated main chain conformation stabilized by hydrogen bonds. Their prediction can be determined by short-range interactions representing the formation of α -helices and by long-range interactions characterizing the β -strands. Two methods can be used to predict secondary structures. The first is ab initio-based, which makes use of statistical calculations of the residues of a single query sequence while the second is homology-based, which makes use of common secondary structural patterns conserved among multiple homologous sequences;
- Tertiary structures, which are the three-dimensional conformation of a polypeptide chain and its determination can be obtained using X-ray crystallography and nuclear magnetic resonance spectroscopy;
- Quaternary structures, which are the complex arrangement of multiple polypeptide chains.

Once the structure of a particular protein is solved, a table of (x, y, z) coordinates representing the spatial position of each atom in the structure is created. The coordinate information is then submitted to the Protein Data Bank (PDB) repository, which uses the PDB format to store the structural details and other metadata (authors, publication details, experiment setup). PDB is a worldwide repository of protein structures being currently managed by the Research Collaboratory for Structural Bioinformatics and empowering extensive Structural Bioinformatics investigations also on protein–DNA interfaces. For instance, in Mount (2004); Gardini *et al.* (2017), the modes of protein binding to DNA have been explored by dividing 629 nonredundant PDB files of protein–DNA complexes into separate classes for structural proteins, transcription factors, and DNA-related enzymes.

Another important research area is structural and functional genomics. The first aims to study genomes referring to the initial phase of genome analysis (construction of genetic and physical maps of a genome, identification of genes, annotation of gene features, and comparison of genome structures) whereas the second refers to the analysis of global gene expression and gene functions in a genome. The latter is also called transcriptomics and uses either sequence- or microarray-based approaches (Xiong, 2006). Many genomes have been published because of the reduced costs in sequencing experiments. However, the new methodologies share the size and quality of the reads (150–300 bp) as a limitation, which represents a challenge for assembly software (Miller *et al.*, 2010). On the other hand, they produce much more sequences (Altmann *et al.*, 2012). Making sense of millions of sequenced base pairs is required in order to assemble the genome. The assembly consists of a hierarchical data structure that maps the sequence data to a supposed target reconstruction (Miller *et al.*, 2010). When a genome is sequenced, two approaches may be adopted: If the species' genome was previously assembled, (reference) mapping with the reference genome is performed. However, if a new genome has not been previously characterized, (de novo) assembly is required (Pevsner, 2015). Fig. 2 shows the typical steps for assembling a genome. The sequencer records sequencing data as luminance images captured during DNA synthesis. Therefore, the calling base refers to the acquisition of image data and its conversion into a DNA sequence by FASTA (Diniz and Canduri, 2017). Also, the quality of each base, called Phred score (Altmann *et al.*, 2012), is obtained. Quality control refers to the quality evaluation of the sequenced reads (Phred score) and the filtering of low quality bases and adapter sequences. Assembling each of the reads is mapped to each other while searching for identity or overlapping regions to construct contiguous fragments corresponding to the overlap of two or more reads (Staats *et al.*, 2014).

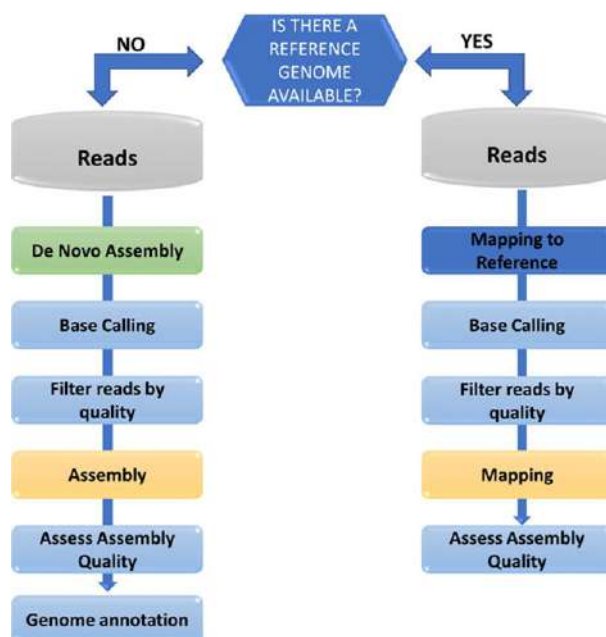


Fig. 2 Flowchart of genome assembly: De novo and based on the reference genome. Figure modified from Diniz, W.J.S, Canduri, F., 2017. Bioinformatics: An overview and its applications. Genetics and Molecular Research 16 (1), gmr16019645.

Management of the Information and Databases

Health Care Information Systems deliver value to individual health care organizations, patients, and providers, as well as during follow-up for entire communities of individuals (Wager *et al.*, 2017; Coronel and Morris, 2017). According to the latest update, published in January 2017, there exist 1739 biological databases. The information sources used by bioinformaticians can be divided into

- Raw DNA sequences;
- Protein sequences;
- Macromolecular structures;
- Genome sequencing.

Public databases store a big amount of information, and they are usually classified into primary and secondary databases (Diniz and Canduri, 2017). Primary databases are composed of experimental results and researchers submit nucleotide sequences, protein sequences, or macromolecular structures directly in this archive. Secondary databases contain more curated data (called content curation process) and a complex combination of computational algorithms and manual analysis are used to interpret and compile the public record of science.

GenBank is a primary database managed by the National Center for Biotechnology Information (NCBI). It was created in 1982 and has been growing at an exponential rate, almost doubling every 14 months. It contains nucleotide sequences obtained from volunteers and is part of the International Nucleotide Sequence Database Collaboration (INSDC) consortium, together with two other large databases: European Molecular Biology Laboratory (EMBL-Bank), and DNA Data Bank of Japan (DDBJ), in whose National Institute of Genetics archives contain over 110 million sequences each (Pevsner, 2015; Prosdocimi *et al.*, 2002). NCBI is the American database, a subsidiary of the National Library of Medicine. Compared to the EMBL of the European Bioinformatics Institute (EBI), which is a center for research and service in bioinformatics containing sequences of DNA, proteins, and macromolecules structures, NCBI also offers the possibility to perform bibliographic searches and to have a direct link between various biological databases obtaining sequence structure genetic article maps (Fig. 3). Thus, users can access sequences, maps, taxonomic information, and structural data of macromolecules. PubMed is an online repository that allows access to 9 million citations in MEDLINE. BLAST is a program developed at the NCBI that allows one to perform very fast similarity searches on whole DNA databases. DDBJ started its activity in 1984 and it is mainly used by Japanese researchers, but it is accessible worldwide through the Internet and together with the EBI and the NCBI, it is part of the International DNA Databases.

Protein Information Resource (PIR), UniProtKB/Swiss-Prot, PDB, Structural Classification of Proteins 2 (SCOP), and Prosite are secondary databases. They are curated and present only information related to proteins, describing aspects of their structures, domains, functions, and classification. In EMBL and DDBJ annotations are very limited, and there may be multiple entries for the same genes. If a sequence encodes a protein, the conceptual translation, or coding sequence, is shown together with a reference to the NCBI protein database.

Universal Protein Resource is a database managed by EBI, the Swiss Institute of Bioinformatics (SIB), and PIR. UniProtKnowledgeBase (UniProtKB) is one of the most complete sources of information on protein sequences and functions. It consists of the Swiss-Prot and TrEMBL sections: Swiss-Prot is manually curated, with very rich annotations while TrEMBL is automatically annotated and contains the conceptual translations of nucleic acid sequences with minor adjustments. Sequences are stored in TrEMBL before being manually annotated and transferred to SwissProt. UniProt Reference clusters groups closely linked sequences in a single document to speed up searches. UniProt archive is a huge parking lot of protein sequences and stores their history and evolution,

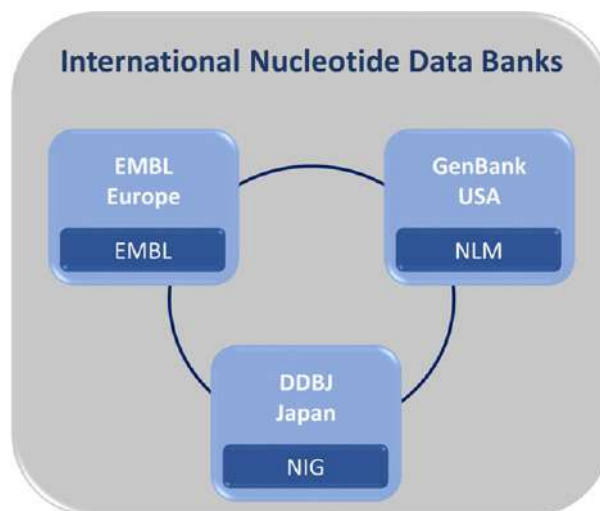


Fig. 3 International nucleotide data banks.

together with all the available related data. The PDB contains the structures of proteins and other biological macromolecules, and provides a variety of resources for the study of their sequences, their functions, and their possible pathological effect ([The UniProt Consortium, 2017](#)).

GenomeNet ([Kotera et al., 2015](#)) is a Japanese network of biocomputational data and services, created in 1991 and managed by the Kyoto University Bioinformatics Center, which hosts the KEGG portal (Kyoto Encyclopedia of Genes and Genomes), which includes gene and protein databases (KEGG genes), chemical components (KEGG ligand), molecular interactions and networks of biochemical reactions (KEGG pathway), of relationships of genomes with the external environment (KEGG brite).

Expert Protein Analysis System proteomics, created in 1993 by the SIB in Switzerland, offers a variety of IT tools for the analysis of protein data and hosts databases of sequences, domains and protein families, proteomic data, models of protein structures, and metabolic pathways ([Gasteiger et al., 2003](#)).

The main genome portals are Ensembl in Great Britain, created jointly in 1999 by the EBI and the Wellcome Trust Sanger Institute; and UCSC Genome Browser, in the United States, created in 2000 at the University of California, Santa Cruz ([Cunningham et al., 2015](#)).

Reference Sequence (RefSeq) by the NCBI is a large collection of annotated sequences that are more restricted. Unlike GenBank, in fact, RefSeq provides just one example of each biological molecule for the major organisms. RefSeq is limited to about 10,000 organisms, while GenBank has sequences obtained from about 250,000 different organisms ([Pruitt et al., 2007; 2014](#)).

The Third Party Annotation database allows authors who publish new experimental evidence to re-record sequences contained in the INSDC database. It is significantly smaller (about one sequence every 12,000) than GenBank ([Benson et al., 2008](#)).

The microRNA database (miRBase) is the central repository for microRNA sequences, very short RNA portions of about 21 nucleotides that seem to play an important role in gene regulation. MicroRNAs control the translation of mRNAs (messenger RNA) of numerous genes and have an important part in cell differentiation and proliferation, in the plasticity of the synapses of the nervous system and in various diseases, including cancer. miRBase hosts sequences of almost 11,000 microRNAs from 58 different species, treats the nomenclature and annotation, and provides computerized prediction programs for target mRNAs ([Kozomara and Griffiths-Jones, 2014](#)).

Research laboratories and scientific journals maintain widely used knowledge portals, which gather information on particular biological problems and provide IT tools to explore them. We report some of them created in the United States. Online Genes to Cognition (G2C), created in 2009 by the Cold Spring Harbor Laboratory, is a neuroscience portal focused on cognitive processes, related diseases, and research approaches ([Croning et al., 2009](#)). Nature Publishing Group hosts Omics gateway, for genomic-scale biology, and The Signaling Gateway (managed together with the University of California, San Diego), focused on signal transduction. *Science* magazine has developed *Science Signaling*, oriented on cellular regulation and signaling, which also maintains a database and various analysis tools organized in dynamically generated diagrams.

Databases of sequence motifs are needed to identify features that indicate specific functions (e.g., catalytic site). It has been observed that genes or proteins that perform a similar function have a similarity in some regions of their sequence. Thus, genes and proteins belonging to the same functional family should contain in their sequence a recurrent motif that characterizes the family and distinguishes it from others. The presence of such signatures is extremely useful to assign a new sequence to a specific family of genes or proteins, and thus to formulate hypotheses about its function.

PROSITE is a database of protein domains, families, and functional sites, integrated with IT tools to identify sequence motifs. It contains specific signatures for more than 1500 protein families and domains with extensive documentation on their structure and function. Through the computational tools hosted by PROSITE (e.g., ScanProsite) or other resources (such as PPSearch of the EMBL-EBI), it is possible to quickly identify which known protein family a given protein sequence belongs to [Hulo et al. \(2006\)](#).

The JASPAR database stores DNA sequences that regulate gene expression (promoters) located before the start of gene transcription sites and binding a variety of regulatory proteins, called transcription factors. The particular combination of factors related to the promoter determines whether the gene will be switched on or off. JASPAR contains 174 distinct pattern patterns representing preferential DNA binding sites of transcription factors, derived from scientific literature and carefully annotated, which can be used to scan genomic sequences ([Sandelin et al., 2004](#)).

Chemical Entities of Biological Interest at the EMBL site, KEGG compound (in the aforementioned GenomeNet network), and Public Chemical database, in the NCBI portal, are databases providing a molecular entities vocabulary, for millions of small chemical substances of biological interest, and descriptions of their structure and their activity ([Kim et al., 2016](#)).

Algorithms and Access to the Data

The choice of comparison algorithms should be based on the desired comparison type, the available computational resources, and the research goals. A rigorous implementation of the Smith–Waterman (SW) algorithm is available, as well as the FASTA program, within the FASTA package. The SW algorithm is one of the most sensitive but it also is computationally demanding. The FASTA algorithm is faster, and its sensitivity is similar to SW in many scenarios ([Brenner et al., 1998](#)). The fastest algorithm is BLAST (Basic Local Alignment Search Tool), the newest versions of which supports gapped alignments and provides a reliable and fast option (the older versions were slower, detected fewer homologs, and had problems with some statistics). Iterative programs like PSI-BLAST require extreme care with their options, as they can provide misleading results; however, they have the potential to find more homologs than purely pairwise methods ([Higgins and Taylor, 2000](#)). In order to achieve higher alignment performances, both the BLOSUM (BLOCKs of amino acid

Substitution Matrix) and the PAM (Point Accepted Mutation) scoring matrices can be used (Dayhoff *et al.*, 1978; Pevsner, 2009; Prosdociimi *et al.*, 2002), which relate the probability of substitution of one amino acid or nucleotide with another due to mutations (the best possible alignment will be one that maximizes the overall score (Junqueira *et al.*, 2014).

Comparative molecular modeling refers to the modeling of the 3D structure of a protein from the structure of an homologous one whose structure has already been previously determined (Capriles *et al.*, 2014). This approach is based on the fact that evolutionarily related sequences share the same folding pattern (Calixto, 2013). The access to biological information could be possible also by searching data banks of networks and models. The most common databases of networks and models are: CO-exPRESsed gene database (COXPRESdb), molecular InteAction database (IntAct), Human Protein Reference Database, Biomolecular Interaction Networks Database, Reactome, KEGG, GO, GeneNetWorks, KWS Online (Java Web Simulation), BioModels and Database Of Quantitative Cellular Signaling. Gene expression can be used to access Ensembl and UCSC Genoma Browser, Project ENCODE (ENCyclopedia of DNA Elements). Finally, Gene Expression Omnibus (GEO) by NCBI and ArrayExpress by EBI, store data in the standardized Minimum Information About a Microarray Experiment format and have online exploration tools (Davis and Meltzer, 2007). Besides hosting many transcriptomic experiments, they host data on the expression of microRNAs, genomic hybridizations, single nucleotide polymorphisms, chromatin immunoprecipitation and peptide profiles. The Allen Brain Atlas contains the three-dimensional, genomic-scale map of the expression of thousands of genes in all areas of the adult mouse brain and in the course of development, down to the cellular level (Hawrylycz *et al.*, 2014).

Data Elaboration

Data mining is the process of learning data with information technology in order to identify hidden structures in the data that allow one to obtain useful information (knowledge discovery) and to make accurate predictions on the evolution of a phenomenon (prediction). The data mining process takes place in several stages: The initial exploration, the construction of a model, and the execution of algorithms. The data mining process attempts to learn something meaningful from the data by highlighting models or groups of objects with similar characteristics. An important distinction is between learning with and without supervision. In the second case, no a priori question is asked about how to divide the data, and learning takes place without specific knowledge about the contents.

In bioinformatics, the unsupervised learning methods cannot be used wherever the biological problem lacks previously known classifications. There are several unsupervised techniques: Hierarchical grouping, k-means, principal component analysis, correspondence analysis and neural networks. Supervised learning, on the other hand, applies to cases in which a particular classification is already known for the training set and we want to build a model that predicts this classification in a new sample. There are various supervised techniques, including decision trees, discriminant analysis, support vector machines and neural networks (Witten *et al.*, 2017).

Expression cluster analysis denotes a number of unsupervised learning algorithms that distribute objects in groups according to similarity criteria where the number of groups can be determined automatically or chosen by the user. The similarity between objects is evaluated through a distance measure: The less objects are distant, the more similar they are and the more easily they will be assigned to the same group. There exist various distance measures, such as the Euclidean distance, which is simply the geometric distance in the multidimensional data space, or the Pearson correlation coefficient, which is a statistical similarity measure. In bioinformatics, an important problem is the extraction of information from large-scale gene expression data obtained by microarrays. The most common approach to gene expression data analysis is the hierarchical grouping, or tree grouping, whereby the relationships between genes are represented by a tree structure in which the proximity of the branches reflects their degree of similarity. The number of groups (clusters) is determined automatically by the algorithm. Sometimes it is convenient to divide the objects into a number of groups of choice, in which case the k-means algorithm can be used, in which attributes are represented as vectors and each group is represented by a point called centroid. The algorithm follows an iterative procedure in which objects are moved between groups in order to minimize the intragroup and maximize the intergroup one until the algorithm converges to a stable solution.

In many data mining problems, neural networks are adopted for their ability to approximate a large family of functions (Cybenko, 1989; Hornik *et al.*, 1989). Generally, neural networks are models used to implement supervised classification techniques. They are able to predict new observations on specific variables, after a learning phase on the preexisting data. The first step is to design the architecture of the network (layers), whose nodes are called neurons. Designing an optimal architecture for the network is not a trivial task, since it heavily depends on the problem and the statistics of the dataset. During training the weights of the network are updated until the examples shown to the network in the input layer give the desired result on the output layer giving a fitting with the desired error. After training, the network is ready to be used to generate predictions about new data. There also exist unsupervised neural networks, called self-organizing neural networks or Kohonen networks (Sarstedt and Mooi, 2014).

Pattern recognition is often performed using probabilistic models such as hidden Markov models, which are suitable to recognize event sequences (e.g., recognition of spoken language, manual writing). In bioinformatics, such models are widely used to identify homologies or to predict coding regions in the genome sequence and protein folding. They derive their name from the chain of Markov, a succession of states in which the transition from a present state to a future takes place with a probability that depends on the present state or from the past. Present state is useful to predict future behavior, while previous history gives insights about the trend of the signal. Markov's theory of processes is often used to give an order to web pages in an Internet search. For instance, Google uses the PageRank

algorithm to assign a numeric weight to web pages, in order to measure their relative relevance. The algorithm is based on the concept of popularity, that is on the frequency with which web pages and online documents are referenced (Zucchini *et al.*, 2016).

Algorithms for Data Elaboration

Statistical and mathematical techniques useful for the exploration of biological data are also adopted by various commercial packages (Wu *et al.*, 2014). MATLAB (MATrixLABoratory) has a section dedicated to bioinformatics tools that allow one to analyze and visualize genomic and proteomic data, and to build models of biological systems. There are specific programs for analyzing microarrays data, for example, GenePix, or for proteomic analysis. R is a widely used open source software environment, and also a software language, within which a variety of statistical and graphical techniques are available (e.g., linear and nonlinear modeling, classical statistical tests, time series analysis tools, classification and grouping algorithms). The tool can be expanded with a vast library of tools obtainable through the CRAN (Comprehensive R Archive Network) repository. Bioconductor provides tools for the analysis of genomic data written in the R language.

Use Cases

RNA-sequencing (RNA-seq) is currently the leading technology for genome-wide transcript quantification. While the volume of RNA-seq data is rapidly increasing, the currently publicly available RNA-seq data is provided mostly in raw form, with small portions processed nonuniformly. This is mainly because the computational requirements, particularly for the alignment step, are a significant barrier for the analysis. To address this challenge RNA-seq and ChIP-seq sample and signature search (ARCHS4) have been created. They are a web resource containing the majority of previously published RNA-seq data from human and mouse at the gene count level. Such uniformly processed data enables easy integration for analyses in various application contexts. For developing the ARCHS4 resource, all available FASTQ files from RNA-seq experiments were retrieved from the GEO, aligned and stored in a cloud-based infrastructure. A total of 137,792 samples are accessible through ARCHS4, with 72,363 mouse and 65,429 human samples. Through the efficient use of cloud resources, the alignment cost per sample has been dramatically reduced. ARCHS4 is updated automatically by adding newly published samples to the database as they become available (Lachmann *et al.*, 2017).

Another example of use of data mining in the bioinformatics scientific literature is biomarkers prediction, such as the prediction and diagnosis of diabetes mellitus (DM). Research efforts have been made to discover and suggest novel biomarkers and finally predict key aspects of the disease, such as its onset, with the bioinformatics tools described above. In general, the arising gaps and limitations of machine learning research in DM are closely related to the availability of data (Kavakiotis *et al.*, 2017).

New Frontiers

The role of data analytics in establishing an intelligent accounting function is to create the insights that help in making better corporate decisions. As organizations develop and adopt technologies related to big data, cognitive computing and the Internet of Things (IoT) applications are growing in both volume and complexity but also new opportunities arise (Pan *et al.*, 2015).

References

- Altmann, A., Weber, P., Bader, D., Preuss, M., *et al.*, 2012. A beginners guide to SNP calling from high-throughput DNA-sequencing data. *Human Genetics* 131, 1541–1554.
- Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L., 2008. GenBank. *Nucleic Acids Research* 36, D25–D30.
- Brenner, S.E., Choithia, C., Hubbard, T.J.P., 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences of the United States of America* 95, 6073.
- Calixto, P.H.M., 2013. Aspectos gerais sobre a modelagem comparativa de proteínas. *Ciencia Equatorial* 3, 10–16.
- Capriles, P.V.S.Z., Trevizani, R., Rocha, G.K., Dardenne, L.E., 2014. Modelos tridimensionais. In: Verli, H. (Ed.), *Bioinformática Da Biologia à Flexibilidade molecular*. São Paulo: SBBq, pp. 147–171.
- Coronel, C., Morris, S., 2017. *Database Systems: Design, Implementation, & Management*. Cengage Learning.
- Croning, M.D.R., Marshall, M.C., McLaren, P., Douglas, A.J., Grant, S.G.N., 2009. G2Cdb: The genes to cognition database. *Nucleic Acids Research* 37 (1), D846–D851.
- Cunningham, F., Ridwan, A.M., Barrell, D., *et al.*, 2015. Ensembl. *Nucleic Acids Research* 43 (D1), D662–D669.
- Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2 (4), 303–314.
- Davis, S., Meltzer, P.S., 2007. GEOquery: A bridge between the Gene Expression Omnibus (GEO) and bioConductor. *Bioinformatics* 23 (14), 1846–1847.
- Dayhoff, M.O., Schwartz, R., Orcutt, B.C., 1978. A model of evolutionary change in proteins. In: Dayhoff, M.O. (Ed.), *Atlas of Protein Sequence and Structure* 5. Washington, D.C.: National Association for Biomedical Research. Suppl. 3.
- Diniz, W.J.S., Canduri, F., 2017. Bioinformatics: An overview and its applications. *Genetics and Molecular Research* 16 (1), gmr16019645.
- Gardini, S., Furini, S., Santucci, A., Niccolai, N., 2017. A structural bioinformatics investigation on protein–DNA complexes delineates their modes of interaction. *Molecular BioSystems* 13, 1010–1017.
- Gasteiger, E., Gattiker, A., Hoogland, C., *et al.*, 2003. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Research* 31 (13), 3784–3788.
- Hawrylycz, M., Ng, L., Feng, D., *et al.*, 2014. The Allen brain atlas. In: Kasabov, N. (Ed.), *Springer Handbook of Bio-/Neuroinformatics*. Berlin, Heidelberg: Springer.
- Higgins, D., Taylor, W., 2000. *Bioinformatics: Sequence, structure and databanks*. Oxford University Press.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2 (5), 359–366.

- Hulo, N., Bairoch, A., Buliard, G., *et al.*, 2006. The PROSITE database. *Nucleic Acids Research* 34 (1), D227–D230.
- Junqueira, D.M., Braun, R.L., Verli, H., 2014. Alinhamentos. In: Verli, H. (Ed.), *Bioinformática Da Biologia à Flexibilidademolecular*. São Paulo: SBBq, pp. 38–61.
- Kavakiotis, I., Save, T., Salifoglou, A., *et al.*, 2017. Machine learning and data mining methods in diabetes research. *Computational and Structural Biotechnology Journal* 15, 104–116.
- Kim, S., Thiessen, P.A., Bolton, E., *et al.*, 2016. PubChem substance and compound databases. *Nucleic Acids Research* 44 (D1), D1202–D1213.
- Koča, J., *et al.*, 2016. Structural bioinformatics databases of general use. *Structural Bioinformatics Tools for Drug Design*. Springer Briefs in Biochemistry and Molecular Biology. Cham: Springer.
- Kotera, M., Moriya, Y., Tokimatsu, T., Kanehisa, M., Goto, S., 2015. KEGG and GenomeNet, new developments, metagenomic analysis. In: Nelson, K.E. (Ed.), *Encyclopedia of Metagenomics*. Boston, MA: Springer.
- Kozomara, A., Griffiths-Jones, S., 2014. miRBase: Annotating high confidence microRNAs using deep sequencing data. *Nucleic Acids Research* 42 (D1), D68–D73.
- Lachmann, A., Torre, D., Keenan, A.B., *et al.*, 2018. Massive mining of publicly available RNA-seq Data from human and mouse. *Nature Communications* 9.
- Luscombe, N.M., Greenbaum, D., Gerstein, M., 2001. What is bioinformatics? A proposed definition and overview of the field. *Methods of Information in Medicine* 40, 346–358. [10.1053/j.ro.2009.03.010](https://doi.org/10.1053/j.ro.2009.03.010).
- Miller, J.R., Koren, S., Sutton, G., 2010. Assembly algorithms for next-generation sequencing data. *Genomics* 95, 315–327.
- Mount, D.W., 2004. *Bioinformatics. Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press.
- Pan, G., Sun, S.P., Chan, C., Yeong, L.C., 2015. *Analytics and cybersecurity: The shape of things to come*.
- Pevsner, J., 2009. *Pairwise sequence alignment*. In: *Bioinformatics and Functional Genomics*, second ed. Wiley-Blackwell.
- Pevsner, J., 2015. *Bioinformatics and Functional Genomics*, third ed. Chichester: John Wiley & Sons Inc.
- Prosdociimi, F., Cerqueira, G.C., Binneck, E., Silva, A.F., 2002. *Bioinformática: Manual do usuário*. Biotecnologia Ciência & Desenvolvimento. 12–25.
- Pruitt, K.D., Brown, G.R., Hiatt, S.M., *et al.*, 2014. RefSeq: An update on mammalian reference sequences. *Nucleic Acids Research* 42 (D1), D756–D763.
- Pruitt, K.D., Tatusova, T., Maglott, D.R., 2007. NCBI reference sequences (RefSeq): A curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research* 35 (1), D61–D65.
- Sandelin, A., Alkema, W., Engström, P., Wasserman, W.W., Lenhard, B., 2004. JASPAR: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research* 32 (1), D91–D94.
- Sarstedt, M., Mooi, E., 2014. *Cluster analysis*. In: *A Concise Guide to Market Research*. Springer Texts in Business and Economics. Berlin, Heidelberg: Springer.
- Staats, C.C., Morais, G.L., Margis, R., 2014. Projetos genoma. In: Verli, H. (Ed.), *Bioinformática Da Biologia à Flexibilidademolecular*. São Paulo: SBBq, pp. 62–79.
- The UniProt Consortium, 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Wager, K.A., Lee, F.W., Glaser, J.P., 2017. *Health Care Information Systems: A Practical Approach for Health Care Management*. Jossey-Bass.
- Witten, I.H., Franke, E., Hall, M.A., Pal, C.J., 2017. *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
- Wu, X., Zhu, X., Wu, G., Ding, W., 2014. Data Mining with big data. *IEEE Transactions on Knowledge and Data Engineering* 26, 1.
- Xiong, Jin, 2006. *Essential Bioinformatics*. Cambridge University Press.
- Zucchini W., MacDonald I.L. and Langrock R. (2016); Hidden Markov models for time series. In: *An Introduction Using R*. CRC Press.

Relevant Websites

<http://www.ebi.ac.uk/>
EMBL-EBI.

<https://www.expasy.org/>
ExPASy.

<http://www.ncbi.nlm.nih.gov/>
NCBI.

<http://www.nig.ac.jp/home.html>
NIGINTERN2018.

<https://www.rcsb.org/>
RCSB PDB.

<http://www.uniprot.org/>
UniProt.Org.

Biographical Sketch

Mariaconcetta Bilotta has achieved a PhD in Biomedical Engineering at the University Magna Graecia of Catanzaro (IT) and now is a biomedical engineer at the Institute S. Anna of Crotona (IT). During the PhD has been visiting student at the WISB (Warwick Center for the Integrative Synthetic Biology) of the University of Warwick, Coventry (UK). Her research interests are modeling and analyzing chemical reactions, design and realization of control systems, synthetic and systems biology, embedded feedback control, automation and robotics in rehabilitation, neurorehabilitation of trunk, microRNA analysis, and health informatics.

Giuseppe Tradigo is a postdoc at the DIMES Department of Computer Science, Models, Electronics and Systems Engineering, University of Calabria, Italy. He has been a Research Fellow at University of Florida, Epidemiology Department, US, where he worked on a GWAS (Genome-Wide Association Study) project on the integration of complete genomic information with phenotypical data from a large patients dataset. He has also been a visiting research student at the AmMBio Laboratory, University College Dublin, where he participated to the international CASP competition with a set of servers for protein structure prediction. He obtained his Ph. in Biomedical and Computer Science Engineering at University of Catanzaro, Italy. His main research interests are big data and cloud models for health and clinical applications, genomic and proteomic structure prediction, data extraction and classification from biomedical data.

Pierangelo Veltri is associate professor in bioinformatics and computer science at Surgical and Clinical Science Department at University Magna Graecia of Catanzaro. He got his PhD in 2002 from University of Paris XI, and worked as researcher at INRIA from 1998 to 2002. His research interests regard database management systems, data integration, biomedical data management, health informatics. He has coauthored more than 100 papers, and he is editor of ACM SIGBioinformatics newsletter and associate editor of *Journal of Healthcare Informatics Research*.

Data Storage and Representation

Antonella Guzzo, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Data Storage of Biomedical Data

In the last decade, several techniques for biological data analysis and manipulation have been developed, including, methods for rapid genomic and RNA sequencing, mass spectrometry, microarray, yeast two-hybrid assay for protein–protein interactions, X-ray crystallography and NMR for protein structures. Moreover, due to these techniques, an enormous amount of biomedical data has been generated. In fact, to have an idea of the volume of data storage that these techniques have to deal with, we can just observe that, for the genomics alone, the first 20 of the largest biomedical institutions (Zachary *et al.*, 2015) currently consume more than 100 petabytes of storage. Moreover, it is estimated that by 2025 the data-storage demands for this application could run to as much as 2–40 exabytes (1 exabyte is 10¹⁸ bytes). Table 1 mentions just three of the most popular databases and their main characteristic.

As a matter of fact, the availability of this enormous amount of data raises important challenges in terms of scalability, complexity and costs of big data storage infrastructure. At the same time, privacy and security are important issues to be taken into account, too. For instance, the most immediate consequence of dealing with such enormous volume of data is, from a practical point of view, that the traditional way for bioinformatics analysis, involving to download the data from public sites (e.g., UCSC (Rosenbloom *et al.*, 2015) and Ensembl (Bronwen *et al.*, 2016)), to install software tools locally, and to run analysis on in-house computer resources is obsolete and evolving toward new and more efficient solutions and approaches. A specific solution approach that is practicable for data storage and transfer is a type of Dropbox for data scientists named Globus Online (Foster, 2011; Allen *et al.*, 2012), which provides storage capacity and secure solutions to transfer the data.

However, large amounts of data require also higher computational infrastructures, as High Performance Computing clusters (HPCs), that not only provide storage solutions but also parallel processing of computational tasks over the stored data. In fact, solutions based on HPCs tend to be difficult to maintain and lead to extremely high costs. A relatively more practical solution to handle big data is the usage of *cloud computing* infrastructures. Cloud computing exploits the full potential of multiple computers and delivers computation and storage as dynamically allocated virtual resources via the Internet. Specifically, bioinformatics clouds (Dai *et al.*, 2012) involve a large variety of services from data storage, data acquisition, and data analysis, which in general fall into four categories: (1) Data as a Service (like public datasets of Amazon Web Services (AWS) (Murty, 2009)) enables dynamic data access on demand and provides up-to-date data that are accessible by a wide range of Web applications; (2) Software as a Service delivers software services online and facilitates remote access to available bioinformatics software tools through the internet; (3) Platform as a Service (like Eoulsan (Jourdren *et al.*, 2012), the cloud-based-for high-throughput sequencing analyses and Galaxy Cloud (Afgan *et al.*, 2011), the cloud-scale-for large-scale data analyses) offers an environment for users to develop, test and deploy cloud applications where computer resources scale automatically and dynamically to match application demand, avoiding to know how many resources are required or to assign resources manually in advance and (4) Infrastructure as a Service (like Cloud BioLinux (Krampis *et al.*, 2012), the virtual machine that is publicly accessible for high-performance bioinformatics computing and CloVR (Angiuoli *et al.*, 2011), the portable virtual machine that incorporates several pipelines for automated sequence analysis) offers a full computer infrastructure by delivering all kinds of visualized resources via the internet, including hardware (e.g., CPUs) and software (e.g., operating systems). Fig. 1 shows the general abstract architecture of bioinformatics clouds.

Despite the advantages of bioinformatics clouds, in particular, despite the possibility for users to access visualized resources as a public utility and pay for the cloud resources that they utilize, only a tiny amount of biological data is accessible in the cloud at present (only AWS, including GenBank, Ensembl, 1000 Genomes, etc.), while the vast majority of data are still deposited in conventional biological databases. Another weakness in full exploiting the potentiality of the bioinformatics clouds is that transferring vast amounts of biological data to the cloud is a significant bottleneck in cloud computing. With this respect, it must be observed that choosing a proper data representations can drastically reduce the required storage and consequently the band of communication if data are transmitted over network, and impacts over the computation time obviously.

The data representation adopted by the most well-known public databases of biological data was initially just a sequence data, with some annotation within a text file format. With the introduction of the XML standard for data representation, a plethora of

Table 1 Few popular databases and their volume

Database	Description	Volume
ArrayExpress	Functional genomics data	45.51 TB of archived data
GenBank	Sequence data in term of DNA	213 billion nucleotide bases in more than 194 million sequences
Protein Data Bank (PDB)	Crystallographic db for 3-D structural data	More of 118192 released structures

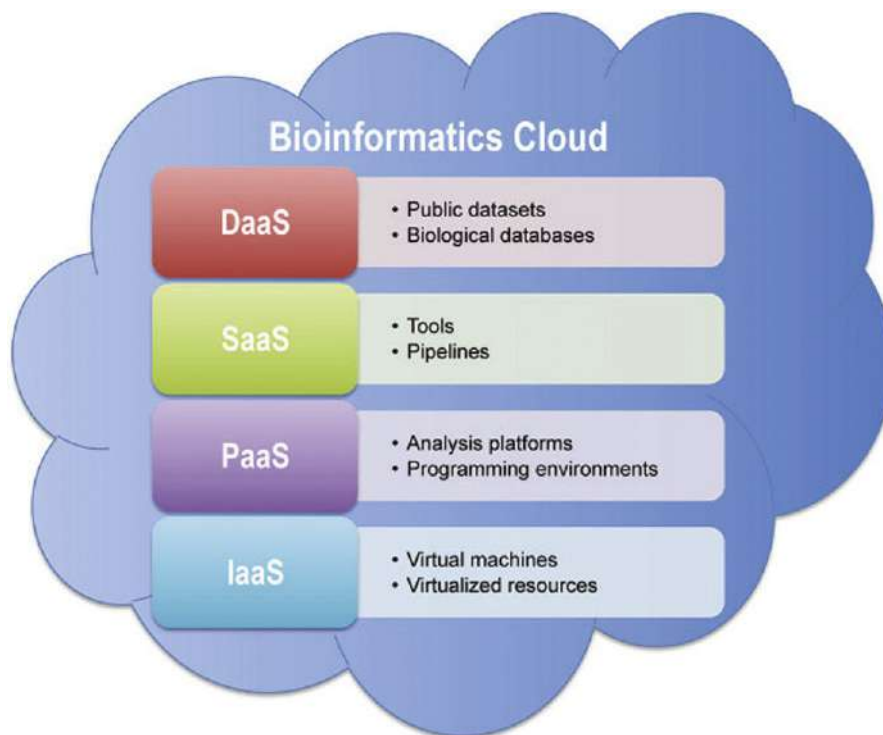


Fig. 1 Categorization of bioinformatics clouds. Reproduced from Dai, L., Gao, X., Guo, Y., *et al.*, 2012. Bioinformatics clouds for big data manipulation. *Biology Direct* 7, 43.

pseudostandards for XML biological data storage have been developed by different public databases, beginning with the flat file format of FASTA, moving to the Genbank and Swiss-Prot formats, and ending to the pure XML format – a review of these formats is behind the scope of this article, so that interested readers are referred to [Shahzad *et al.* \(2017\)](#) for a complete survey on modern data format. To our ends, instead, it is interesting to observe that, very recently, it is emerging the idea of using Synthetic DNA for data Storage and, in fact, several researchers have come up with a new way to encode digital data in the base sequence of DNA to create the highest-density large-scale data storage scheme. Unfortunately real applications are limited so far by high cost and errors in the data, while in a recent study researchers announced they have made the process error-free and 60 per cent more efficient compared to previous results, approaching the theoretical maximum for DNA storage ([Bornholt *et al.*, 2016](#)).

Beside the specific data format compliant with the different databases, it is relevant to discuss here some data model adopted for representing biological data, which come as single and/or multiple abstractions allowing us to carry out a large number of data processing and task analysis.

Basic Data Models for Biological Data

Background

To begin the discussion of data model representation, basic terms and concepts should be first introduced. Three are the key components in everything related to life: deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and proteins.

DNA is the basic hereditary material in all cells and contains all information, parts of DNA called genes, code for proteins which perform all the fundamental processes for living using biochemical reactions. The genetic information is represented both by DNA and RNA. In fact, while cells use only DNA, some viruses, the retroviruses, have their genome encoded into the RNA, which is replicated into the infected cells. The central dogma of molecular biology was first enunciated by Francis Crick in 1958 and re-stated in a paper appeared in the *Nature* journal published in 1970. It states that biological function is heavily dependent on the biological structure and it deals with the detailed residue-by-residue transfer of sequential information. Thus, it states that information cannot be transferred back from protein to either protein or nucleic acid. In other words, the central dogma of molecular biology is that genes may perpetuate themselves and work through their expression in form of proteins, but it is not possible to go the other way around and obtain the gene sequence from the protein. Note that the expression of a gene is its product, that is, the protein for which the gene encodes information.

The genetic information is encoded into the sequence of the bases of the DNA and perpetuate through the replication. This is represented by a loop in the [Fig. 2](#) from DNA to DNA, meaning that the molecules can be copied. Transcription, the next arrow, is



Fig. 2 The schematic central dogma.

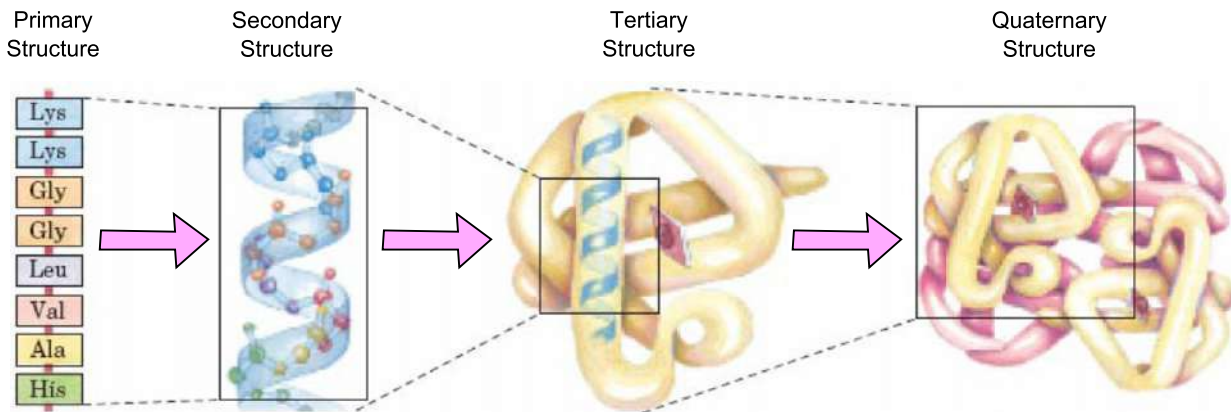


Fig. 3 Different levels of protein structures. Google images

the process by which the enzyme (that is a protein working as a biochemical catalysator) RNA polymerase, reads the sequence of bases on a gene and constructs an mRNA molecule from that sequence. Translation, the last arrow, is the process by which a ribosome, a macromolecular assembly, reads the information contained in the mRNA molecule and synthesizes a protein molecule from the sequence on the mRNA molecule. Thus, each protein molecule is a product of the gene that codes for it. In turn, proteins are responsible for carrying out various functions inside the cell. For instance, many proteins work as enzymes that catalyze the reactions that occur in living organisms or they can interact with other molecules for performing storage and transport functions. Moreover, these fundamental components provide mechanical support and shape to mechanical work as, for example, muscular contraction. Finally, several proteins have an essential role in the decoding of cellular information and also regulate the transcription of a gene to an mRNA molecule. Recently, it is worth observing that relations from RNA to DNA are also considered and studied, by capturing the mechanism via which retrovirus can copy their RNA genomes into DNA.

From a biological point of view, RNA is a polymer that contains ribose rather than deoxyribose sugars. The normal base composition is made up of guanine, adenine, cytosine, and uracil. Proteins are macromolecules composed by linear polymers, or chains, of amino acids. All organisms use the same set of 20 amino acids as building blocks in the protein synthesis. The variations of the order in which amino acids are connected and their total number let to obtain an almost unlimited number of proteins. The DNA uses four nucleotides: adenine (A), guanine (G), cytosine (C) e thymine (T). Since it is not possible to represent each of the 20 different amino acid by a nucleotide, each amino acid corresponds to a group of nucleotides. By choosing words composed by two nucleotides only a few number of combinations can be obtained. Instead, by choosing words composed by three nucleotides many more combinations can be obtained, that are sufficient to encode the 20 amino acids. Thus, a code of three or more nucleotides is necessary and the one made of three nucleotides seems to be valid for all organisms. Each triplet is called codon. All the 64 codons specify amino acids except three of them, that are stop triplets, and are stop signals in the transduction process. Since 61 codons are used to encode 20 amino acids, multiple triplets may encode for the same amino acid, and in general these have the same first two nucleotides and different third nucleotides. The starting triplet is the one encoding the methionine amino acid: all proteins start with this amino acid. The transduction process ends and the protein is released when one of the three stop triplets is recognized. The 20-amino acids are composed by an amide group and a carboxylic group, also known as α -carbon. The α -carbon also binds hydrogen atoms and a side chain. The side chain is distinctive to each amino acid. The amino acids are bound to one another by the condensation of a α -carboxylic group of one amino acid to the amide group of another amino acid to form a chain. This bound is known as peptidic bound and the involved amino acids are called residues. The free amide and carboxylic groups at the opposite extremities of the peptidic chain are called *N*-terminal (amide terminal) and *C*-terminal (carboxylic terminal). Conventionally, all the residues of a peptidic chain are numbered starting from *N*-terminals.

On the basis of protein complexity, a protein can have at most four levels of structural organization (see [Fig. 3](#)). The primary structure of a protein is the sequence of its amino acids, forming the polypeptidic chain, describing the one-dimensional structure of the protein. The other three levels encode the protein three-dimensional structure. In more detail, the polypeptidic chain patterns that regularly repeats into the protein denote the secondary structure. The tertiary structure is related to the three-dimensional structure of the whole polypeptide. The quaternary structure is related to the arrangement of two or more polypeptidic chains in one polymer. Alterations of the conditions of the environment, or some chemical treatments, may lead to a destruction of the native conformation of proteins with the subsequent loosing of their biological activities. This process is called denaturation. Proteins have different functions; they can provide structure (ligaments, fingernails, hair), help in digestion

(stomach enzymes), aid in movement (muscles), and play a part in our ability to see (the lens of our eyes is pure crystalline protein).

Strings and Sequences for Representing Biomolecules (DNA and Proteins)

Many biological objects can be interpreted as strings. As already pointed out in the previous subsection, biologically a strand of DNA is a chain of nucleotides adenine, cytosine, guanine, and thymine. The four nucleotides are represented by letters A, C, G, and T, respectively. Thus, a strand of DNA can be encoded as a string built from a 4-letter alphabet A, C, G, T, corresponding to the four nucleotides. Formally, let $\mathcal{A} = \{A, C, G, T\}$ be an alphabet and let k be a integer with $k \geq 1$, we call *DNA k -words* a string of length k over the letters A, C, G, and T. Specifically, let $\mathcal{W}_k$ be the set of all possible k -words formed using the alphabet $\mathcal{A}$, the size of $|\mathcal{W}_k|$ being 4^k . Moreover, for $k=1$ each 1-words denotes one of the four DNA bases (individual nucleotides), for $k=2$ each 2-words denotes one of the $|\mathcal{W}_k|=4^2=16$ possible dinucleotides (AA, AC, AG, AT, CA, ...), for $k=3$ each 3-words denotes a one of the $|\mathcal{W}_k|=4^3=64$ possible codons (AAA, AAC, ...). For $k \geq 4$, a k -words is a generic sequence that can have biological interest. It is possible that the biologists cannot determine some nucleotides in a DNA strand. In this case, the character N is used to represent an unknown nucleotides in the DNA sequence of the strand. In other words, N is a wildcard character for any one character among A, C, G, and T. A DNA sequence is an *incomplete sequence* if the corresponding strand contains one or more character N ; otherwise, it is said a *complete sequence*. A complete sequence is said to agree with an incomplete sequence if it is a result of substituting each N in the incomplete sequence with one of the four nucleotides. For example, ACCCT agrees with ACNNT, but AGGAT does not.

Similarly, proteins can also be seen as strings. In fact, we define a protein as the linear sequence of its amino acids. Formally, let $\mathcal{A}_p = \{a_1, \dots, a_j\}$ be a set of strings, where a_i encodes an amino acid recognized in a protein structure. Then, each protein can be represented by a string over the alphabet $\mathcal{A}_p$, where every letter in the alphabet correspond to a different amino acid. The total number of the recognized amino acids is 20 and thus the size of the $\mathcal{A}_p$ -alphabet is 20, too. Each protein sequence has a length varying from several tens to several thousands and usually contains long repeated sub-sequences. Moreover each protein sequence codes protein molecules, but not every string over the amino acid alphabet codes real protein molecules (Kertesz-Farkas, 2008). The amino acid sequence of a protein is determined by the gene that encodes for it. The differences between two primary structures reflect the evolutive mutations. The amino acid sequences of related species are, with high probability, similar and the number of differences in their amino acid sequences are a measure of how far in the time the divergence between the two species is located: the more distant the species are the more different the protein amino acid sequences are. The amino acid residues, essential for a given protein to maintain its function, are conserved during the evolution. On the contrary, the residues that are less important for a particular protein function can be substituted by other amino acids. It is important to note that some proteins have a higher number of substitutable amino acids than others, thus proteins can evolve at different speeds. Generally, the study of molecular evolution is focused on family of proteins. Proteins belonging to the same family are called homologous and the tracing of the evolution process starts from the identification of such families. Homologous are identified by using specialized amino acids sequence alignment algorithms that, by analyzing two or more sequences, search for their correspondences.

Recent studies have demonstrated that is easier to detect that two proteins share similar functions based on their structures rather than on their sequences. As a consequence, more attention is currently paid to the structural representation instead of the sequence one, and in fact there has been a growing interest in finding and analyzing similarities between proteins, with the aim of detecting shared functionality that could not be detected by sequence information alone.

Structures for Representing Biomolecules

The secondary structure of a protein is referred to the general three-dimensional form of local segments of proteins. It does not describe specific atomic positions in three-dimensional space, but is defined by patterns of hydrogen bonds between backbone amide and carboxylic groups. The secondary structure is related to the spacial arrangement of amino acid residues that are neighbors in the primary structure. The secondary structure is the repetition of four substructures that are: α helix, β sheet, β turn, Ω loop. The most common secondary structures are alpha helices and beta sheets (see Fig. 4). A common method for determining protein secondary structure is far-ultraviolet (far-UV, 170–250 nm) circular dichroism. A less common method is infrared spectroscopy, which detects differences in the bond oscillations of amide groups due to hydrogen-bonding. Finally, secondary-structure contents may be accurately estimated using the chemical shifts of an unassigned NMR spectrum.

The tertiary structure of a protein is its three-dimensional structure, as defined by the atomic coordinates. The function of a protein is determined by its three-dimensional structure and the three-dimensional structure depends on the primary structure. Efforts to predict tertiary structure from the primary structure are generally known as protein structure prediction. However, the environment in which a protein is synthesized and allowed to fold are significant determinants of its final shape and are usually not directly taken into account by current prediction methods.

The biological activity of a protein is related to the conformation the protein assumes after the folding of the polypeptidic chain. The conformation of a molecule is a spacial arrangement that depends on the possibility for the bonds to spin. In physiologic conditions a protein has only one stable conformation, known as native conformation.

On the contrary of secondary structure, the tertiary structure also takes into account amino acids that are far in the polypeptidic sequence and belong to different secondary structures but interact with one another. To date, the majority of known protein

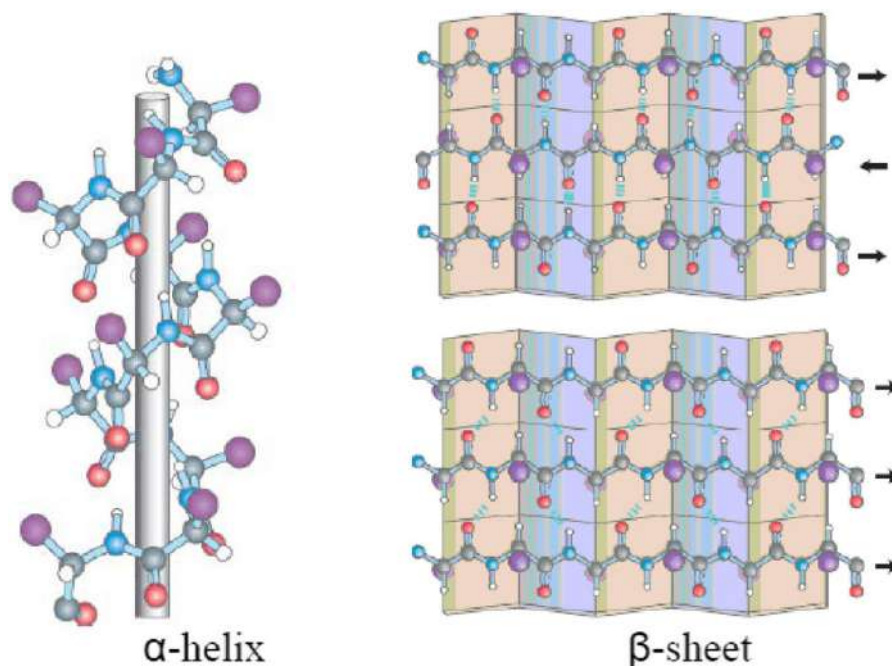


Fig. 4 Two examples of protein secondary structure: α helix and β sheet.

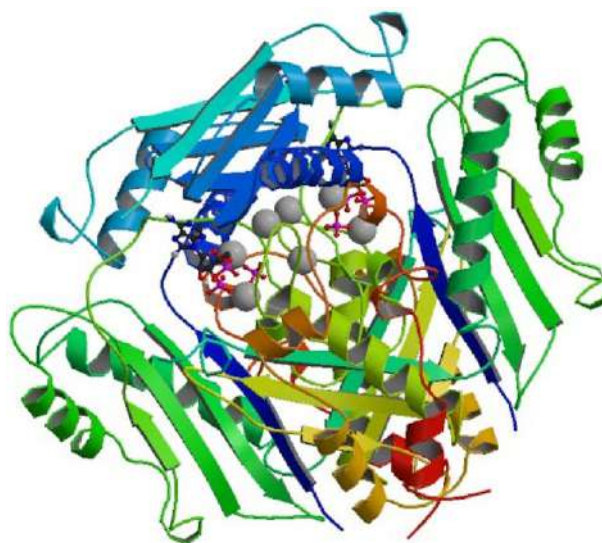


Fig. 5 An example of protein tertiary structure. Uniprot database

structures have been determined by the experimental technique of X-ray crystallography. A second common way of determining protein structures uses NMR, which provides somewhat lower-resolution data in general and is limited to relatively small proteins.

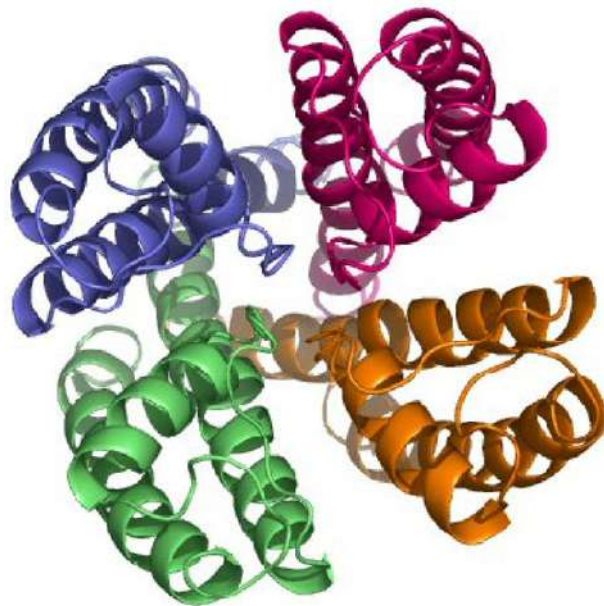
An example of tertiary structure as reported by the PDB database (see Relevant Website section) (Rose *et al.*, 2017) is shown in Fig. 5. The figure represents the tertiary structure of the *S-Adenosylmethionine Synthetase* with 8-BR-ADP.

Many proteins are assembled in more than one polypeptide chain, known as protein subunits. In addition to the tertiary structure of the subunits, multiple-subunit proteins possess a quaternary structure, which is the three-dimensional spatial arrangement of the several polypeptidic chains, corresponding to protein subunits. According to this structure, proteins can be subdivided in two groups: homo-oligomers and hetero-oligomers. The first group is made of proteins composed by only one type of subunit, while the second one is made of proteins that are composed by different types of subunits. The proteins belonging to the first group are those having structural and supporting roles, while the proteins belonging to the second one have dynamic functions.

Protein quaternary structures can be determined using a variety of experimental techniques that require a sample of proteins in a variety of experimental conditions. The experiments often provide an estimate of the mass of the native protein and, together

Table 2 The nomenclature used to identify protein quaternary structures

<i>Number of subunits</i>	<i>Name</i>
1	Monomer
2	Dimer
3	Trimer
4	Tetramer
5	Pentamer
6	Hexamer
7	Heptamer
8	Octamer
9	Nonamer
10	Decamer
11	Undecamer
12	Dodecamer
13	Tridecamer
14	Tetradecamer
15	Pentadecamer
16	Hexadecamer
17	Heptadecamer
18	Octadecamer
19	Nonadecamer
20	Eicosamer

**Fig. 6** An example of protein quaternary structure. Uniprot database

with knowledge of the masses and/or stoichiometry of the subunits, allow the quaternary structure to be predicted with a fixed accuracy. However, it is not always possible to obtain a precise determination of the subunit composition. The number of subunits in a protein complex can often be determined by measuring the hydrodynamic molecular volume or mass of the intact complex, which requires native solution conditions.

Table 2 reports the nomenclature used to identify protein quaternary structures. The number of subunits in an oligomeric complex are described using names that end in *-mer* (Greek for “part, subunit”).

Fig. 6 shows an example of the quaternary structure of a protein. The quaternary structure reported in the figure is a *tetramer* and is related to a potassium ion channel protein from *Streptomyces lividans*.

The quaternary structure is important, since it characterizes the biological function of proteins when involved in specific biological processes. Unfortunately, quaternary structures are not immediately deducible from protein amino acid sequences.

Biological Networks as Expressive Data Models

Graph Theory

Biological networks are data structures used to store information about molecular relations and interactions. Usually, they are conveniently represented as graphs (Huber *et al.*, 2007). A undirect graph $\mathcal{G}$ is a pair $\mathcal{G} = \langle V, E \rangle$, where V is the set of nodes and E is the set of edges $E = \{\{i, j\} | i, j \in V\}$, so that the elements from E are subsets of elements of V . A direct graph is defined as a pair $\mathcal{G} = \langle V, E \rangle$ where, instead, elements in E are ordered pairs – so that if $E = (i, j)$, then i is considered the source node and j a target node. The ordered pairs of vertices are called directed edges, arcs or arrows.

In graph-based system biology, nodes of the graph represent cellular building blocks (e.g., proteins or genes) and the set of edges represent interactions (see Fig. 7). Specifically, each edge represents a mutual interaction in the case of undirected graph, while, conversely, the flow of material or information from a source node to a target node, in a directed graph.

In computational biology, we usually use weighted graphs, that is a graphs $\mathcal{G} = \langle V, E \rangle$ associated with a weight function $w: E \rightarrow \mathbb{R}$, where $\mathbb{R}$ denotes the set of all real numbers and the weight w_{ij} represents the relevance of the connection among the nodes i and j . As an example, relations whose importance varies are frequently assigned to biological data to capture the relevance of co-occurrences identified by text mining, sequence or structural similarities between proteins or co-expression of genes (Jensen *et al.*, 2009; Lee *et al.*, 2014). Different types of graphs are used to represent different types of biological networks, each of which stores information about interactions related to specific entities or molecules.

Relevant kinds of networks include: transcriptional regulatory networks, signal transduction networks, metabolic networks, protein-protein interaction networks (or PPI network), domain interaction networks, Gene Co-Expression Networks and genetic interaction networks. Some of these networks will be discussed in the rest of the section.

Protein-Protein Interaction (PPI) Networks

PPI are powerful models to represent the pairwise protein interactions of the organisms, such as building of protein complexes and the activation of one protein by another protein. Their visualization aids biologists in pinpointing the role of proteins and in gaining new insights about the processes within and across cellular processes and compartments, for example, for formulating and experimentally testing specific hypotheses about gene function (Pavlopoulos *et al.*, 2011). A PPI network is common represented as a directed graph $\mathcal{G} = \langle V, E \rangle$ with an associated function t , where V is the set of proteins, E the set of directed interactions, and t a function $t: E \rightarrow T$ which defines the type of each edge (interaction type). PPI networks can be derived from a variety of large biological databases that contain information concerning PPI data. Some well-known databases are the Yeast Proteome Database (YPD) (Hodges *et al.*, 1999), the Munich Information Center for Protein Sequences (MIPS) (Mewes *et al.*, 2004), the Molecular Interactions (MINT) database (Zanzoni *et al.*, 2002), the IntAct database (Kerrien *et al.*, 2007), the Database of Interacting Proteins (DIP) (Xenarios *et al.*, 2000), the Biomolecular Interaction Network Database (BIND) (Bader *et al.*, 2009), the BioGRID database (Stark *et al.*, 2006), the Human Protein Reference Database (HPRD) (Keshava Prasad *et al.*, 2009), and the HPID (Han *et al.*, 2004) and the DroID (Yu *et al.*, 2008) database for Drosophila.

Regulatory Networks

This kind of network contains information about the control of gene expression in cells. Usually, these networks use a directed graph representation in an effort to model the way that proteins and other biological molecules are involved in gene expression

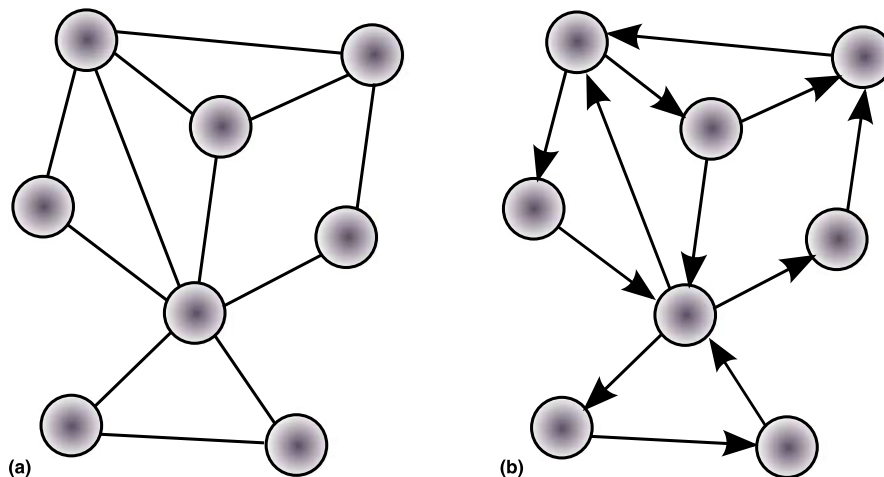


Fig. 7 Examples of graph structures.

and try to imitate the series of events that take place in different stages of the process. They often exhibit specific motifs and patterns concerning their topology. Databases collecting Protein-DNA interaction data are listed as follow: JASPAR ([Sandelin et al., 2004](#)), TRANSFAC ([Wingender et al., 1996](#)) or B-cell interactome (BCI) ([Lefebvre et al., 2007](#)), while post-translational modification can be found in databases like Phospho. ELM ([Diella et al., 2004](#)), NetPhorest ([Miller et al., 2008](#)) or PHOSIDA ([Gnad et al., 2007](#)).

Signal Transduction Networks

This kind of networks uses a graph representing both protein interactions and biochemical reactions, and their edges are mostly directed, indicating the direction of signal propagation ([Pavlopoulos et al., 2011](#); [Ma'ayan et al., 2005](#)). Often, these networks use multi-edged directed graphs to represent a series of interactions between different bioentities such as proteins, chemicals or macromolecules and to investigate how signal transmission is performed either from the outside to the inside of the cell, or within the cell.

Environmental parameters change the homeostasis of the cell and, depending on the circumstances, different responses can be triggered. Similarly to GRNs, these networks also exhibit common patterns and motifs concerning their topology ([Pavlopoulos et al., 2011](#)). Databases that store information about signal transduction pathways are MiST ([Ulrich and Zhulin, 2007](#)), TRANSPATH ([Krull et al., 2003](#)).

Metabolic and Biochemical Networks

This kind of networks is the complete network of metabolic reactions of a particular cell or organism, for example to produce energy or synthesize specific substances. A metabolic pathway is a connected sub-network of the metabolic network either representing a series of chemical reactions occurring within a cell at different time points ([Pavlopoulos et al., 2011](#); [Jeong et al., 2000](#)). The main role within a metabolic network is played by the enzymes, since they are the main determinants in catalyzing biochemical reactions. Often, enzymes are dependent on other cofactors, such as vitamins for proper functioning. The collection of pathways, holding information about a series of biochemical events and the way they are correlated, is called a metabolic network. Modern sequencing techniques allow the reconstruction of the network of biochemical reactions in many organisms, from bacteria to human ([Ma et al., 2007](#)). Some well-known databases collecting information about biochemical networks are listed as follow: the Kyoto Encyclopedia of Genes and Genomes (KEGG) ([Kanehisa et al., 2010](#)), EcoCyc ([Keseler et al., 2009](#)), BioCyc ([Karp et al., 2005](#)) and metaTIGER ([Whitaker et al., 2009](#)).

Evolutionary Trees and Networks

Hierarchical networks are considered in molecular biology. In particular, phylogenetic/evolutionary trees represent the ancestral relationships between different species, and thus they are widely used to study evolution, to describe and explain the history of species, i.e., their origins, how they change, survive, or become extinct ([Pavlopoulos et al., 2011](#)). Formally, a phylogenetic tree $T = (V, E, \delta)$ is a triple consisting of a set of nodes V (taxons), a set edges $E = V \times V$ (links) and a function δ mapping edges to real numbers, quantifying the biological divergence between the target node, for example, biological time or genetic distance. A node in the tree can be a leaf node representing species, sequences, or similar entities; on the other hand, it can be an internal node representing (hypothetical) ancestors generated from phylogenetic analysis. Phylogenetic trees are often stored in the Newick file format ([Cardona et al., 2008](#)), which makes use of the correspondence between trees and nested parentheses.

Databases that store phylogenetic information are TreeBASE ([Roderic, 2007](#)), which stores all kinds of phylogenetic data (e.g., trees of species, trees of populations, trees of genes) representing all biotic taxa, and TreeFam ([Li et al., 2006](#)), a database of phylogenetic trees of gene families found in animals.

As an extension of tree, we have phylogenetic networks that provide an explicit representation of the evolutionary relationships among sequences, genes, chromosomes, genomes, or species. They differ from phylogenetic trees because of the explicit modeling, by means of hybrid nodes instead of only tree nodes, of reticulate evolutionary events such as recombination, hybridization, or lateral gene transfer, and differ also from the implicit networks that allow for visualization and analysis of incompatible phylogenetic signals ([Huson and Bryant, 2006](#)).

Finally, signal transduction, gene regulatory, protein-protein interaction and metabolic networks interact with each other and build a complex network of interactions; furthermore, these networks are not universal but species-specific, i.e., the same network differs between different species.

Closing Remarks

We have described data storage methods for biological data. In particular, we have discussed basic data models and more advanced ones, namely biological networks, with special emphasis on PPI networks, regulatory networks, metabolic networks and evolutionary networks.

See also: Bioinformatics Data Models, Representation and Storage. Text Mining for Bioinformatics Using Biomedical Literature

References

- Afgan, E., *et al.*, 2011. Harnessing cloud computing with Galaxy Cloud. *Nat Biotechnol* 29, 972–974.
- Allen, B., *et al.*, 2012. Software as a service for data scientists. *Communications of the ACM* 55 (2), 81–88.
- Angiuoli, S.V., *et al.*, 2011. CloVR: A virtual machine for automated and portable sequence analysis from the desktop using cloud computing. *BMC Bioinformatics* 12, 356.
- Bader, G.D., *et al.*, 2009. BIND – The Biomolecular Interaction Network Database. *Nucleic Acids Res* 29 (1), 242–245.
- Bornholt J. *et al.*, 2016. A DNA-Based Archival Storage System ACM - Association for Computing Machinery, April 1.
- Brownwen, L., Aken, *et al.*, 2016. The Ensembl gene annotation system Database, <http://doi:10.1093/database/baw093>.
- Cardona, G., Rossella, F., Valiente, G., 2008. Extended Newick: It is Time for a Standard Representation of Phylogenetic Networks. *BMC Bioinformatics* 9.
- Dai, L., Gao, X., Gao, Y., Xiao, J., Zhang, Z., 2012. Bioinformatics Clouds for Big Data Manipulation. *Biology Direct* 7, 43.
- Diella, F.C.S., *et al.*, 2004. Phospho.ELM: A database of experimentally verified phosphorylation sites in eukaryotic proteins. *BMC Bioinformatics* 5.
- Foster, I., 2011. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. *Internet Computing, IEEE* 15 (3), 70–73.
- Gnad, F., *et al.*, 2007. PHOSIDA (phosphorylation site database): Management, structural and evolutionary investigation, and prediction of phosphosites. *Genome Biol.* 8 (11).
- Han, K., Park, B., Kim, H., Hong, J., Park, J., 2004. HPID: The Human Protein Interaction Database. *Bioinformatics* 20 (15), 2466–2470.
- Hodges, P.E., McKee, A.H., Davis, B.P., Payne, W.E., Garrels, J.I., 1999. The Yeast Proteome Database (YPD): A model for the organization and presentation of genome-wide functional data. *Nucleic Acids Res* 27 (1), 69–73.
- Huber, W., Carey, V.J., Long, L., Falcon, S., Gentleman, R., 2007. Graphs in molecular biology. *BMC Bioinformatics* 8.
- Huson, D.H., Bryant, D., 2006. Application of phylogenetic networks in evolutionary studies. *Mol Biol Evol* 23 (2), 254–267.
- Jensen, L.J., *et al.*, 2009. STRING 8 – A global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.* 412–416.
- Jeong, H., Tombor, B., Albert, R., Oltvai ZN, A.L., 2000. The large-scale organization of metabolic networks. *Nature.* 407, 651–654.
- Jourdren, L., Bernard, M., Dillies, M.A., Le Crom, S., 2012. Eoulsan: A cloud computing-based framework facilitating high throughput sequencing analyses. *Bioinformatics*, 28: 1542–1543.
- Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M., 2010. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res.* pp. 355–360.
- Karp, P.D., *et al.*, 2005. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Res* 33 (19), 6083–6089.
- Kerrien, S., *et al.*, 2007. IntAct—open source resource for molecular interaction data. *Nucleic Acids Res.* 561–565.
- Kertesz-Farkas, A., 2008. Protein Classification in a Machine Learning Framework PhD Thesis by Attila Kertesz-Farkas.
- Keseler, I.M., *et al.*, 2009. EcoCyc: A comprehensive view of *Escherichia coli* biology. *Nucleic Acids Res.* 464–470.
- Keshava Prasad, T.S., *et al.*, 2009. Human Protein Reference Database—2009 update. *Nucleic Acids Res.* 767–772.
- Krampis, K., *et al.*, 2012. Cloud BioLinux: Pre-configured and on-demand bioinformatics computing for the genomics community. *BMC Bioinformatics* 13, 42.
- Krull, M., *et al.*, 2003. TRANSPATH: An integrated database on signal transduction and a tool for array analysis. *Nucleic Acids Res.* 31 (1), 97–100.
- Lee, H.K., Hsu, A.K., Sajdak, J., Qin, J., Pavlidis, P., 2014. Coexpression analysis of human genes across many microarray data sets. *Genome Res* 14 (6), 1085–1094.
- Lefebvre, C., *et al.*, 2007. A context-specific network of protein-DNA and protein-protein interactions reveals new regulatory motifs in human B cells. *Lecture Notes in Bioinformatics (LNCS)* 4532, 42–56.
- Li, Heng, *et al.*, 2006. TreeFam: A curated database of phylogenetic trees of animal gene families. *Nucleic Acids Research* 34 (Database issue), 572–580.
- Ma'ayan, A., *et al.*, 2005. Formation of regulatory patterns during signal propagation in a Mammalian cellular network. *Science.* 309 (5737), 1078–1083.
- Ma, H., *et al.*, 2007. The Edinburgh human metabolic network reconstruction and its functional analysis. *Mol Syst Biol.* 3 (135),
- Mewes, H.W., *et al.*, 2004. MIPS: Analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.* 41–44.
- Miller, M.L., *et al.*, 2008. Linear motif atlas for phosphorylation-dependent signaling. *Sci Signal* 1 (35).
- Murty, J., 2009. Programming Amazon Web Services first ed., Book - O'Reilly.
- Pavlopoulos, M.L.A., Georgios, A., *et al.*, 2011. Using Graph Theory to Analyze Biological Networks. *BioData Mining* 4, 10.
- Rosenbloom, K.R., *et al.*, 2015. The UCSC Genome Browser database: 2015 update. *Nucleic Acids Res* 43 (Database issue), 670–681.
- Rose, P.W., *et al.*, 2017. The RCSB protein data bank: Integrative view of protein, gene and 3D structural information. *Nucleic Acids Research* 45, 271–281. <http://www.rcsb.org/pdb/home/home.do>.
- Sandelin, A., *et al.*, 2004. JASPAR: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 32, 91–94.
- Shahzad, A., *et al.*, 2017. Modern Data Formats for Big Bioinformatics Data Analytics. *Int. Journal of Advanced Computer Science and Applications* 8 (4).
- Stark, C., Breitkreutz, B.J., Reguly, T., *et al.*, 2006. BioGRID: A general repository for interaction datasets. *Nucleic Acids Res.* 535–539.
- Roderic, D.M., 2007. TBMMap: A taxonomic perspective on the phylogenetic database TreeBASE. *BMC Bioinformatics.* 8:158.
- Ulrich, L.E., Zhulin, I.B., 2007. MiST: A microbial signal transduction database. *Nucleic Acids Res.* 35, 386–390.
- Whitaker, J.W., Letunic, I., McConkey, G.A., Westhead, D.R., 2009. metaTIGER: A metabolic evolution resource. *Nucleic Acids Res.* 531–538.
- Wingender, E., Dietze, P., Karas, H., Knuppel, R., 1996. TRANSFAC: A database on transcription factors and their DNA binding sites. *Nucleic Acids Res.* 24 (1), 238–241.
- Xenarios, I., Rice, D.W., Salwinski, L., *et al.*, 2000. DIP: The database of interacting proteins. *Nucleic Acids Res* 28 (1), 289–291.
- Yu, J., Pacifico, S., Liu, G., Finley Jr., R.L., 2008. DroID: The *Drosophila* Interactions Database, a comprehensive resource for annotated gene and protein interactions. *BMC Genomics* 9, 461.
- Zachary, D., Stephens, *et al.*, 2015. Big data: Astronomical or genomics. *PLoS Biol* 13 (7).
- Zanzoni, A., *et al.*, 2002. MINT: A Molecular Interaction database. *FEBS Lett.* 513 (1), 135–140.

Relevant Website

<http://www.rcsb.org/pdb/home/home.do>
Protein Data Bank.

Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing

Barbara Calabrese, University "Magna Graecia", Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Omics refers to the characterization of pools of biological molecules (i.e., nucleic acids, proteins, metabolites). Omics studies have developed rapidly thanks to technological advances in high-throughput technologies, such as mass spectrometry, microarray and next generation sequencing, in the last two decades.

Microarray technology allows researchers to assess thousand of genes, proteins or other analytes through a variety of means. In addition to this technology, next-generation sequencing allows the sequencing of nucleic acids in millions of parallel reactions. Moreover, advancements in other technologies, such as mass spectrometry, enable researchers to collect data and gain major knowledge on biological and cellular processes.

The development of these technologies have led to a production of a vast amount of complex and multidisciplinary data that pose new challenges in terms of management, analysis and interpretation.

Mass Spectrometry

Mass spectrometry is a methodology aiming to (i) determine the molecular weight of a large variety of molecules and biomolecules (peptides, proteins), (ii) provide structural information on proteins, and on their post-translational modifications and (iii) carry out quantitative analysis of both small molecules and of biomolecules, providing high sensitivity and high specificity. Mass spectrometric measurements are carried out in the gas phase on ionized analytes (Aebersold and Mann, 2003). Specifically, molecules are ionized and then introduced into an electrical field where they are sorted by their mass to charge ratio (m/z). Thus, a mass spectrometer consists of an ion source, a mass analyzer that measures the mass-to-charge ratio of the ionized analytes and a detector, that registers the number of ions at each m/z value.

The two most common techniques to ionize the sample are:

- MALDI (Matrix Assisted Laser Desorption Ionization): a laser is used to ionize and vaporize a small amount of sample, which is then drawn into the mass spectrometer for analysis;
- ESI (Electrospray Ionization): a stream of liquid containing the sample is ionized by applying an electrical charge to it. This creates a stream of ions which repel each other upon exiting the capillary tubing creating a fine plume of ions which is then drawn into the mass spectrometer for analysis.

There are different types of mass analyzer (Matthiesen and Bunkenborg, 2013). One common is the quadrupole. This consists of 2 pairs of charged rods. There is an electrical potential between each pair of rods which draws the ions towards one rod. The polarity of this electrical field is oscillated rapidly, which causes the ions travel through the quadrupole in a spiral trajectory. Each oscillation frequency allows ions with a particular m/z to pass through, while the other ions crash into the poles and lose their charge, or are ejected from the quadrupole. By varying the oscillation frequency, ions with different m/z ratios will get through. The number of ions passing through at any given frequency is measured by the mass spectrometer's detector and a graph of intensity vs. m/z is created from this data. An MS Spectra represent the abundance of each ion as a function of its mass, i.e., the final mass spectrum, that is a long sequence of pairs of values ($i, m/z$), where i is the intensity or abundance and m/z is the mass of each detected molecule. Spectra are normally represented as histograms that report the abundance of each ion as a function of its mass, reasonably assuming that all the ions produced by the analysis have a single charge. The abundances are reported as a ratio to the base peak, which is the most abundant peak observed in the spectrum. This normalization allows to have spectra that are a function only of the analyte and of the conditions of analysis.

Another component used to filter ions is the ion trap. In an ion trap ions are collected and held either in a 3-dimensional space or a 2-dimensional plane. Once a certain number of ions have been collected, or after a set time, the ions are ejected from the trap. This ejection voltage is ramped in a way that allows different m/z ions to be ejected at slightly different times. This time difference creates an MS Spectra. Because a greater number of ions is collected, this method typically has a higher sensitivity than using a quadrupole mass filter.

One more common method of sorting ions is the Time-of Flight or TOF analyzer. In this analyzer the ions are collected in a similar manner to an ion trap, and then accelerated with one push into an empty chamber with an electrical field in it. The chamber is at a very low pressure, usually about $1 \cdot 10^{-7}$ torr, this allows the ions to fly freely with few collisions with other molecules. The ions are reflected by the electrical field (ion mirror) into a detector. Because larger m/z ions take longer to be turned around in the electrical field they arrive at the detector later, allowing for the creation of a MS Spectra. Because of the way the ions are sorted this method of analysis has high mass accuracy (Walther and Mann, 2010).

Tandem Mass Spectrometry

ESI and MALDI generate ions causing minimal fragmentation in molecules. This implies that the only information available from the analysis is a more or less accurate measurement of the molecular weight analyte. This is not sufficient for a detailed characterization structure of peptides and proteins (primary structure, modifications post-translational). It is necessary to use Tandem Mass Spectrometry, that consists in carrying out a double mass analysis on a given sample (Han *et al.*, 2008). To do this, it is possible to use two analyzers in series, or use the same analyzer at different times. It is an important technique, better known as MS/MS: the protein is pretreated with a chemical reagent to obtain various fragments, the mixture is injected into an instrument that is two spectrometers placed in series. The spectrometer consists of an MS-1 chamber which has the task of selecting among the various ions the desired one. The selected ion is put in contact with a gas (it can be of helium) in the collision cell. From the impact between the gas and the ion, fragments are obtained, which are separated into the MS-2 chamber, based on the mass to charge ratio.

This means that each group contains all the fragments loaded, due to the breaking of the same type of bond, even if in different positions, so that each successive peak has an amino acid less than that which precedes it. The difference in mass between one peak and another one identifies the amino acid that has been lost and therefore the peptide sequence. Through this method it is possible to catalog separate cell proteins through electrophoresis.

Peptide Mass Fingerprinting (PMF)

Peptide Mass Fingerprinting (PMF), also known as mass fingerprinting, was developed in 1993. It is a high throughput protein identification technique in which the mass of an unknown protein can be determined. PMF is always performed with Matrix-assisted laser/desorption ionization time of flight (MALDI-TOF) mass spectrometry (Thiede *et al.*, 2005).

Peptide means protein fragment, which is often generated by trypsin, mass means the molecular size of peptides and the fingerprinting presents the uniqueness of the masses of peptides. This technique means that the digestion of a protein by an enzyme can provide a specific fingerprint of great specificity, which can possibly identify the protein from this information alone.

In this technique, after separation of proteins by gel electrophoresis or liquid chromatography and being cleaved with a proteolytic enzyme, it is possible to get experimental peptide mass through mass spectrometry. On the other hand, the theoretical masses are achieved by using computer programs that translate the known genome of the organism into proteins or the proteins in the database, then theoretically cut the proteins into peptides, and calculate the absolute masses of the peptides. The experimentally obtained peptide masses are compared with theoretical peptide mass. The results are statistically analyzed to find the best match.

There are several steps for PMF:

- Protein separation: the proteins of interest from a sample are separated by gel electrophoresis.
- Digestion: the protein of interest is digested by the proteolytic enzyme. Trypsin is the favored enzyme for PMF. It is relatively cheap, highly effective and generates peptides with an average size of about 8–10 amino acids, which is suitable for MS analysis.
- Mass Spectrometric analysis: the peptides can be analyzed with different types of mass spectrometers, such as MALDI-TOF or ESI-TOF. The mass spectrometric analysis produces a peak list, which is a list of molecular weights of the fragments.
- In silico digestion: software performs in silico digestion on database proteins with the same enzyme used in the experimental digestion and generates a theoretical peak list. Mascot, MS-Fit, and Profound are the most frequently used search programs for PMF.
- Comparison: in this step, it is performed a comparison between peak list and theoretical peak list to get best match.

Microarray

A DNA microarray (commonly known as gene chip, DNA chip, or biochip) consists of a collection of microscopic DNA probes attached to a solid surface like glass, plastic, or silicon chips forming an array. These arrays are used to examine the profile expression of a gene, which is also known as the transcriptome or the set of messenger RNA (mRNA) transcripts expressed by a group of genes (Schulze and Downward, 2001).

To perform a microarray analysis, mRNA molecules are typically collected from both an experimental sample and a reference sample. The two mRNA samples are then converted into complementary DNA (cDNA), and each sample is labelled with a fluorescent probe of a different color. For instance, the experimental cDNA sample may be labelled with a red fluorescent dye, whereas the reference cDNA may be labelled with a green fluorescent dye. The two samples are then mixed together and allowed to bind to the microarray slide. The process in which the cDNA molecules bind to the DNA probes on the slide is called hybridization. Following hybridization, the microarray is scanned to measure the expression of each gene printed on the slide. If the expression of a particular gene is higher in the experimental sample than in the reference sample, then the corresponding spot on the microarray appears red. In contrast, if the expression in the experimental sample is lower than in the reference sample, then the spot appears green. Finally, if the expression are equal, then the spot appears yellow.

The measurement of gene expression by means of microarrays has a considerable interest both in the basic research field and in the medical diagnostics, in particular of genetic-based diseases, where the gene expression of healthy cells is compared with that of cells affected by the disease in question.

Other applications of microarrays are the analysis of SNPs (Single Nucleotide Polymorphisms), the comparison of RNA populations of different cells, and the use for new sequencing methods of the DNA.

There are different types of microarrays, catalogued depending on the material that is used as probes:

- cDNA microarray, with probes longer than 200 bases obtained for reverse transcription from mRNA, fragmented, amplified with PCR and deposited on a support glass or nylon;
- Oligonucleotide microarray, with probes with a length between 25 and 80 bases obtained from biological or artificial material and deposited on a glass support;
- Oligonucleotide microarray, with probes between 25 and 30 bases synthesized in situ with photolithographic techniques on silicon wafers.

For the analysis of gene expression two dominant technologies are present on the market: GeneChip, developed, marketed and patented by Affymetrix, Inc. and the “spotted” array, popularized by the Brown labs at Stanford.

Spotted Arrays

The first DNA micro-arrays used full length cDNA molecules (amplified using PCR) as probes. The probes were spotted onto glass slides with an activated surface capable of binding the DNA. The spotting process is the process of placing small droplets containing the cDNA probes in an organized grid on the micro-array slide. Spotted arrays can equally well be used with synthetic oligonucleotide probes.

Genechip

A completely different approach to the construction of DNA micro-arrays is to synthesize the probes directly on the surface of the array. The approach was initially commercialized by the company Affymetrix (California, USA) under the name “GeneChip”. The idea is to build the oligonucleotide one base at the time. Starting out with an empty activated silicon surface, the synthesis occurs during a series of coupling steps: in each step the four nucleotides are presented to the entire surface one at the time and will be coupled to the growing oligonucleotides in a tightly controlled manner. The individual positions on the array are targeted for coupling by a light-based deprotection of the oligonucleotides and the use of a series of lithographic masks to shield the rest of the array.

Next Generation Sequencing

Next generation technologies for DNA sequencing allow for high speed and throughput. The advantage of these technologies is the possibility of obtaining the DNA sequence by amplifying the fragment, without having to clone it (Metzker, 2010).

The Next Generation Sequencing (NGS) technology is based on the analysis of the light emitted from each nucleotide, which allows to identify the type. Unfortunately, the light that each of these emits is too small, thus, it must be amplified. To amplify it, PCR (polymerase chain reaction) is usually used. It is a technique that allows multiplication, and therefore amplification, of nucleic acid fragments of which the initial and final nucleotide sequences are known. After this stage, an amplified filament is obtained and it must be separated to be studied. Thus the second stage is the separation step. The separation can be performed using a picotiter plate (PTP), one specie of slide able to divide the various nucleotides. Once the nucleotides have been separated, it is possible to analyze them. The analysis is performed by detecting the light that each nucleotide emits, as the light emitted by each type of nucleotide is unique.

Commercial Systems for NGS

The main systems implementing the next generation sequencing techniques are described here (Buermans and den Dummen, 2014).

Roche 454 System. It was the first to be marketed in 2005. This system uses pyrosequencing and PCR amplification. Initially this system reached reads of 100–150 bp (bp means base pair), producing about 200,000 reads, with a throughput of 20 Mb per run.

In 2008 it was proposed an evolution, the GS4 GSX FLX Titanium sequencer, which reached 700 bp long reads, with an accuracy of 99.9% after filtering, with an output of 0.7 Gb per run in 24 h. In 2009, the combination of the GS Junior method with the 454 GS system led the output at 14 Gb per run. Further developments have led to the GS FLX +, able to sequence reads long up to 1 kb.

The high speed combined with the length of the reads produced, are the strong points of this system. However, the cost of the reagents remains a problem to be solved.

AB SOLiD System. was marketed by Applied Biosystems in 2006. The system uses the two-base sequencing method, based on the ligation sequencing, i.e., it performs filament analysis in both directions. Initially the length of the reads was only 35 bp, and

the output reached 3 Gb per run. Thanks to the two-base sequencing, the accuracy of the application reached 99.85% after filtering. In 2010 the SOLiD 5500x1 was released, with reads length of 85 bp, precision 99.99% and 30 Gb output per run. A complete run can be done in 7 days.

The main problem with this method is the length of the reads, which for many applications is not enough. The tool is used to sequence whole genomes, targeted sequences, epigenomic.

Illumina GA/HiSeq System. In 2006, Solexa released the genomic analyzer (GA), in 2007 the company was bought by Illumina. The system uses synthesis sequencing (SBS) and bridge amplification, an alternative at the PCR.

At first the output of the analyzer was 1 Gb per run, then brought to 50 Gb per run in 2009. In 2010 the HiSeq 2000 was released, which uses the same strategies as its predecessor, arriving for at 600 Gb per run, obtainable in 8 days. In the next future it should be able to even reach 1 Tb per run. Also the length of the reads has been improved, going from 35 bp to about 200 bp in the latest versions. The cost per operation is among the lowest, compared to the various sequencers.

Helicos single-molecule sequencing device (HeliScope) appeared for the first time in 2007. The method, unlike the previous ones, uses a technique that analyzes the molecules individually, this way it was possible to obtain even greater accuracy, not dirtying the genome with chemical reagents. Also for this system the throughput is in the order of Gigabases. However, the main disadvantage of this method remains the low capacitance to manage the indels (insertions-deletions) correctly, resulting in increased errors. Another problem is the length of the reads, which has never exceeded 50 bp.

Closing Remarks

High-throughput technologies allowed the development and proliferation of the 'omics' disciplines (i.e., genomics, proteomics, metabolic, transcriptomics, epigenomics, to cite a few). Moreover, they contribute to the generation of a high volume of data relative to different levels of biological complexity (DNA, mRNA, proteins, metabolites).

See also: Bioinformatics Data Models, Representation and Storage. Clinical Proteomics. Exome Sequencing Data Analysis. Genome Annotation: Perspective From Bacterial Genomes. Mass Spectrometry-Based Metabolomic Analysis. Metabolome Analysis. Next Generation Sequence Analysis. Next Generation Sequencing Data Analysis. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Proteomics Data Representation and Databases. Text Mining for Bioinformatics Using Biomedical Literature. Transcriptomic Databases. Utilising IPG-IEF to Identify Differentially-Expressed Proteins. Whole Genome Sequencing Analysis

References

- Aebersold, R., Mann, M., 2003. Mass spectrometry based proteomics. *Nature* 422 (6928), 198–207.
- Buermans, H.P.J., den Dummen, J.T., 2014. Next generation sequencing technology: Advances and applications. *Biochimica et Biophysica Acta – Molecular Basis of Disease* 1842 (10), 1932–1941.
- Han, X., Aslanian, A., Yates, J.R., 2008. Mass spectrometry for proteomics. *Current Opinion in Chemical Biology* 12 (5), 483–490.
- Matthiesen, R., Bunkenborg, J., 2013. Introduction to mass spectrometry-based proteomics. In: Matthiesen, R. (Ed.), *Mass Spectrometry Data Analysis in Proteomics. Methods in Molecular Biology (Methods and Protocols)*, vol. 1007. Totowa, NJ: Humana Press.
- Metzker, M.L., 2010. Sequencing technologies – The next generation. *Nature Reviews Genetics* 11 (1), 31–46.
- Schulze, A., Downward, J., 2001. Navigating gene expression using microarrays: A technology review. *Nature Cell Biology* 3 (8), E190:5.
- Thiede, B., Höhenwarter, W., Krah, A., *et al.*, 2005. Peptide mass fingerprinting. *Methods* 35 (3), 237–247.
- Walther, T.C., Mann, M., 2010. Mass spectrometry-based proteomics in cell biology. *The Journal of Cell Biology* 190 (4), 491–500.

Standards and Models for Biological Data: Common Formats

Barbara Calabrese, University “Magna Graecia”, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In bioinformatics and computational biology, the need to standardise the data is an important issue. Nowadays, there are many different ways of representing similar biological data and this makes the integration process and subsequent mining process more difficult.

Standards are defined as an agreed compliant term or structure to represent a biological activity (Lapatas *et al.*, 2015). The adoption of standards facilitates data re-use and sharing, allowing to overcome interoperability problems relative to different data formats. They provide uniformity and consistency in the data from different organizations and technologies. A standard within the omics data field, can be generally characterized by its domain and scope (Chervitz *et al.*, 2011). The domain refers to the type of experimental data (i.e., transcriptomics, proteomics, metabolomics), whereas the scope refers to the area of applicability of standard.

Standards for Biological Data

Numerous standard initiatives have been proposed, some of them are reported in the following paragraphs.

OBO, The Open Biological and Biomedical Ontologies: The Open Biological and Biomedical Ontology (OBO) Foundry (see “Relevant Websites section”) is a collective of ontology developers that are committed to collaboration and adherence to shared principles. The mission of the OBO Foundry is to develop a family of interoperable ontologies that are both logically well-formed and scientifically accurate. To achieve this, OBO Foundry participants voluntarily adhere to and contribute to the development of an evolving set of principles including open use, collaborative development, non-overlapping and strictly-scoped content, and common syntax and relations, based on ontology models that work well, such as the Gene Ontology (GO) (Smith *et al.*, 2007).

CDISC, Clinical data interchange standards consortium: The Clinical Data Interchange Standards Consortium (CDISC) (see “Relevant Websites section”) is a global, open, multidisciplinary, not-for-profit organization that has established standards to support the acquisition, exchange, submission and archive of clinical research data and metadata. The mission is to develop and support global, platform-independent data standards that enable information system interoperability to improve medical research and related areas of healthcare. CDISC standards are vendor-neutral and platform-independent freely available via the CDISC website (Huser *et al.*, 2015).

HUPO-PSI, Human Proteome Organization-Proteomics Standards Initiative: The Human Proteome Organization (HUPO) was formed in 2001 to consolidate national and regional proteome organizations into a single worldwide body (see “Relevant Websites section”). The Proteome Standards Initiative (PSI) was established by HUPO with the remit of standardizing data representation within the field of proteomics, to the end that public domain databases can be established where all such data can be deposited, exchanged, or downloaded and utilized by laboratory workers. The HUPO-PSI organized a series of meetings at which data producers, data users, instrumentation vendors, and analytical software producers, gathered to discuss the problem. As the HUPO-PSI is a completely voluntary organization with limited resources, activity is focussed on a few key areas of proteomics, constituting the PSI work groups.

GAGH, Global Alliance for Genomics and Health: The Global Alliance for Genomics and Health (GA4GH) is an international, nonprofit alliance formed in 2013 to accelerate the potential of research and medicine, to advance human health. Bringing together 500 + leading organizations working in healthcare, research, patient advocacy, life science, and information technology, the GA4GH community is working together to create frameworks and standards to enable the responsible, voluntary, and secure sharing of genomic and health-related data (see “Relevant Websites section”).

COMBINE, Computational Modeling in Biology: The ‘COMputational Modeling in Biology’ NETwork (COMBINE) is an initiative to coordinate the development of the various community standards and formats for computational models. By doing so, it is expected that the federated projects will develop a set of interoperable and non-overlapping standards covering all aspects of modeling in biology (see “Relevant Websites section”).

MSI, Metabolomics Standards Initiative: The Metabolomics Standards Initiative (MSI) was conceived in 2005, as an initiative of Metabolomics Society activities, now coordinated by the Data Standards Task Group of the Society. The MSI is an academic policy provider, to support the development of open data and metadata formats for metabolomics. MSI followed on earlier work by the Standard Metabolic Reporting Structure initiative (report) and the Architecture for Metabolomics consortium (ArMet). The early efforts of MSI were focused on community-agreed reporting standards, the so called minimal information (MI) checklists and data exchange formats to support the MIs reporting standards. MSI aim was to provide a clear description of the biological system studied and all components of a metabolomics study, as well as to allow data to be efficiently applied, shared and reused (see “Relevant Websites section”).

Clinical and Laboratory Standards Institute: The Clinical and Laboratory Standards Institute (see “Relevant Websites section”) is an international organization that develops and fosters clinical laboratory testing standards based on input from and consensus among industry, government, and health care professionals. The CLSI publishes standards for a wide range of biological specialties such as clinical chemistry and toxicology, hematology, method evaluation, microbiology, etc. It also provides guidance for obtaining accreditation and certifications as set by the International Organization for Standardization.

RDA, Research Data Alliance: The Research Data Alliance (RDA) was launched as a community-driven organization in 2013 by the European Commission, the United States Government's National Science Foundation and National Institute of Standards and Technology, and the Australian Government's Department of Innovation, with the goal of building the social and technical infrastructure to enable open sharing of data. With more than 6600 members from 135 countries (February 2018), RDA provides a neutral space where its members can come together through focused global Working and Interest Groups to develop and adopt infrastructure that promotes data-sharing and data-driven research, and accelerate the growth of a cohesive data community that integrates contributors across domain, research, national, geographical and generational boundaries (see "Relevant Websites section").

Formats for Biological Data

Biological data formats represent biological information in a file. High volume omics data cannot be manually analysed, thus there is the need of the adoption of commonly agreed formats to represent them in computer readable files.

Many different formats even to represent the same type of data have been proposed. However, as pointed out in the previous paragraphs, the adoption of standards in file formats is essential to data exchange and integration.

For example, in the case of NGS data, there are no standards, but a set of commonly used formats (FASTA/Q, SAM, VCF, GFF/GTF, etc.). This lack arises several issues relative to time and effort spent on converting raw files across multiple sequencing platforms to make these compatible (Lapatas *et al.*, 2015).

FASTA Format

FASTA format is a text-based format for representing either nucleotide sequences or amino acid sequences, in which base pairs or amino acids are represented using single-letter codes. Usually the first line starts with the ">" sign, followed by a sequence identification code, and optionally followed by a textual description of the sequence. Since it is not part of the official description of the format, software can choose to ignore this, when it is present. One or more lines contain the sequence itself. A file in FASTA format may comprise more than one sequence.

An example of a FASTA sequence (see "Relevant Websites section") is reported in the following:

BTBSCRYR

```

tgcaccaaacatgtctaaagctggaacccaaattactttcttgaagacaaaaactttca
aggccgccactatgacagcgattgcgactgtgcagattccacatgacctgagccgctg
caactccatcagatggaaggaggcacctgggctgtgtatgaaaggcccaatttgcctgg
gtacatgtacatcctacccggggcgagatcctgagtagcagcactggatgggcctcaa
cgaccgcctcagcctcctgcagggtgttcacctgtctagtggagccagataagcttca
gatctttgagaagggggttttaatggtcagatgcatgagaccaggaagactgcccttc
catcatggagcagttccacatgcgggaggtccactcctgttaagggtgctggaggcgctg
gatcttctatgagctgcccaactaccgaggcagggcagtagcctgtggacaagaaggagta
ccggaagcccgtcactgggtgcagcttcccagctgtccagcttccgcgcattgt
ggagtgtatcacagatgcggccaaacgctggctggcctgtcatccaataagcattat
aaataaacaattggcatgc

```

Sequences are expected to be represented in the standard IUB/IUPAC (see "Relevant Websites section") amino acid and nucleic acid codes.

The accepted nucleic acid codes are (see "Relevant Websites section"):

<i>IUPAC nucleotide code</i>	<i>Base</i>
A	Adenine
C	Cytosine
G	Guanine
T or U	Thymine or Uracil
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	Any base
or -	gap

The accepted amino acid codes are (see “Relevant Websites section”):

<i>IUPAC amino acid code</i>	<i>Three letter code</i>	<i>Amino acid</i>
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic Acid
E	Glu	Glutamic Acid
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine

Gen-Bank Format

GenBank is the National Institutes of Health (NIH) genetic sequence database, an annotated collection of all publicly available DNA sequences (Benson *et al.*, 2013). GenBank is part of the International Nucleotide Sequence Database Collaboration (see “Relevant Websites section”), which comprises the DNA DataBank of Japan (DDBJ), the European Nucleotide Archive (ENA), and GenBank at NCBI.

GenBank format (GenBank Flat File Format) consists of an annotation section and a sequence section. An annotated sample GenBank record could be examined at the following link provided in: “Relevant Websites section”. The start of the annotation section is marked by a line beginning with the word “LOCUS”. The LOCUS field contains a number of different data elements, including locus name, sequence length, molecule type, GenBank division, and modification date. Furthermore, the annotation section contain the following main elements:

- **Definition:** It is a brief description of sequence; includes information such as source organism, gene name/protein name, or some description of the sequence’s function (if the sequence is non-coding). If the sequence has a coding region (CDS), description may be followed by a completeness qualifier, such as “complete cds”.
- **Accession:** It is the unique identifier for a sequence record.
- **Version:** A nucleotide sequence identification number that represents a single, specific sequence in the GenBank database.
- **GI:** “GenInfo Identifier” sequence identification number, in this case, for the nucleotide sequence. If a sequence changes in any way, a new GI number will be assigned.
- **Keywords:** Word or phrase describing the sequence. If no keywords are included in the entry, the field contains only a period.
- **Source:** Free-format information including an abbreviated form of the organism name, sometimes followed by a molecule type.
- **Organism:** The formal scientific name for the source organism (genus and species, where appropriate) and its lineage, based on the phylogenetic classification scheme used in the NCBI Taxonomy Database.
- **Reference:** Publications by the authors of the sequence that discuss the data reported in the record. References are automatically sorted within the record based on date of publication, showing the oldest references first.
- **Authors:** List of authors in the order in which they appear in the cited article.
- **Title:** Title of the published work or tentative title of an unpublished work.
- **Features:** Information about genes and gene products, as well as regions of biological significance reported in the sequence. These can include regions of the sequence that code for proteins and RNA molecules, as well as a number of other features.
- **Source:** Mandatory feature in each record that summarizes the length of the sequence, scientific name of the source organism, and Taxon ID number. Can also include other information such as map location, strain, clone, tissue type, etc., if provided by submitter.
- **Taxon:** A stable unique identification number for the taxon of the source organism.
- **CDS:** coding sequence; region of nucleotides that corresponds with the sequence of amino acids in a protein (location includes start and stop codons). The CDS feature includes an amino acid translation. Authors can specify the nature of the CDS by using the qualifier “/evidence=experimental” or “/evidence=not_experimental”.

- **protein_id:** A protein sequence identification number, similar to the Version number of a nucleotide sequence. Protein IDs consist of three letters followed by five digits, a dot, and a version number.
- **GI:** “GenInfo Identifier” sequence identification number, in this case, for the protein translation.
- **Translation:** The amino acid translation corresponding to the nucleotide coding sequence (CS). In many cases, the translations are conceptual. Note that authors can indicate whether the CDS is based on experimental or non-experimental evidence.
- **Gene:** a region of biological interest identified as a gene and for which a name has been assigned.

The start of sequence section is marked by a line beginning with the word “ORIGIN” and the end of the section is marked by a line with only “//”.

EMBL Format

The EMBL Nucleotide Sequence Database at the EMBL European Bioinformatics Institute, UK, offers a large and freely accessible collection of nucleotide sequences and accompanying annotation. The database is maintained in collaboration with DDBJ and GenBank (Kulikova *et al.*, 2007). The flatfile format used by the EMBL to represent database records for nucleotide and peptide sequences from EMBL database (Stoesser *et al.*, 2002). The EMBL flat file comprises of a series of strictly controlled line types presented in a tabular manner and consisting of four major blocks of data:

- **Descriptions and identifiers.**
- **Citations:** citation details of the associated publications and the name and contact details of the original submitter.
- **Features:** detailed source information, biological features comprised of feature locations, feature qualifiers, etc.
- **Sequence:** total sequence length, base composition (SQ) and sequence.

An example of EMBL flat file is reported in the following:

```
ID   XXX;   XXX;   {'linear' or 'circular'};   XXX;   XXX;   XXX;   XXX.
XX
AC   XXX;
XX
AC   *   _{entry_name} (where entry_name=sequence name: e.g. _contig1 or _scaffold1)
XX
PR   Project :PRJEBNNNN;
XX
DE   XXX
XX
RN   [1]
RP   1–2149
RA   XXX;
RT   ;
RL   Submitted {(DD-MMM-YYYY)} to the INSDC.
XX
FH   Key                Location/Qualifiers
FH
FT   source              1..588788
FT                      /organism="{scientific organism name}"
FT                      /mol_type="{in vivo molecule type of sequence}"
XX
SQ   Sequence 588788 BP; 101836 A; 193561 C; 192752 G; 100639 T; 0 other;
    tgcgtactcg aagagacgcg cccagattat ataagggcgt cgtctcgagg ccgacggcgc
60  gccggcgagt acgcgtgac cacaaccga agcgaccgtc gggagaccga gggtcgtcga
120 ggggtgatac gttctgcct tcgtgccggg aaacggccga agggaacgtg gcgacctgcg
    [sequence truncated]...
```

SAM Format

SAM stands for Sequence Alignment/Map format. It is a TAB-delimited text format consisting of a header section, which is optional, and an alignment section. If present, the header must be prior to the alignments. Header lines start with ‘@’, while alignment lines do not. Each alignment line has 11 mandatory fields for essential alignment information such as mapping position, and variable number of optional fields for flexible or aligner specific information. In the following table, the mandatory fields are briefly described.

<i>No.</i>	<i>Name</i>	<i>Description</i>
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEquence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33 = Phred base quality)

They must be present but their value can be a '*' or a zero (depending on the field) if the corresponding information is unavailable. The optional fields are presented as key-value pairs in the format of TAG:TYPE:VALUE. They store extra information from the platform or aligner. The SAM format specification gives a detailed description of each field and the predefined TAGs (Li *et al.*, 2009).

VCF Format

The variant call format (VCF) is a generic format for storing DNA polymorphism data such as SNPs, insertions, deletions and structural variants, together with rich annotations (Danecek *et al.*, 2011). VCF is usually stored in a compressed manner and can be indexed for fast data retrieval of variants from a range of positions on the reference genome. The format was developed for the 1000 Genomes Project, but it has also been adopted by other projects. A VCF file consists of a header section and a data section. The header contains an arbitrary number of meta-information lines, each starting with characters '##', and a TAB delimited field definition line, starting with a single '#' character. The meta-information header lines provide a standardized description of tags and annotations used in the data section. It can be also used to provide information about the means of file creation, date of creation, version of the reference sequence, software used and any other information relevant to the history of the file. The field definition line names eight mandatory columns, corresponding to data columns representing the chromosome (CHROM), a 1-based position of the start of the variant (POS), unique identifiers of the variant (ID), the reference allele (REF), a comma separated list of alternate non-reference alleles (ALT), a phred-scaled quality score (QUAL), site filtering information (FILTER) and a semicolon separated list of additional, user extensible annotation (INFO). In addition, if samples are present in the file, the mandatory header columns are followed by a FORMAT column and an arbitrary number of sample IDs that define the samples included in the VCF file. The FORMAT column is used to define the information contained within each subsequent genotype column, which consists of a colon separated list of fields. All data lines are TAB delimited and the number of fields in each data line must match the number of fields in the header line.

The VCF specification includes several common keywords with standardized meaning. In the following some examples of the reserved tags:

Genotype columns:

- GT, genotype, encodes alleles as numbers: 0 for the reference allele, 1 for the first allele listed in ALT column, 2 for the second allele listed in ALT and so on. The number of alleles suggests ploidy of the sample and the separator indicates whether the alleles are phased ('/') or unphased ('/') with respect to other data lines.
- PS, phase set, indicates that the alleles of genotypes with the same PS value are listed in the same order.
- DP, read depth at this position.
- GL, genotype likelihoods for all possible genotypes given the set of alleles defined in the REF and ALT fields.
- GQ, genotype quality, probability that the genotype call is wrong under the condition that the site is being variant. Note that the QUAL column gives an overall quality score for the assertion made in ALT that the site is variant or no variant.

INFO column: Missing values are represented with a dot. For practical reasons, the VCF specification requires that the data lines appear in their chromosomal order. The full format specification is available at the VCFtools web site.

- DB, dbSNP membership.
- H3, membership in HapMap3.
- VALIDATED, validated by follow-up experiment.
- AN, total number of alleles in called genotypes.
- AC, allele count in genotypes, for each ALT allele, in the same order as listed.
- SVTYPE, type of structural variant (DEL for deletion, DUP for duplication, INV for inversion, etc. as described in the specification).
- END, end position of the variant.
- IMPRECISE, indicates that the position of the variant is not known accurately; and

- CIPOS/CIEND, confidence interval around POS and END positions for imprecise variants.

Other common biological data formats

Despite the large variety of computer readable formats, Lapatas *et al.* realised that the most commonly used ones are ascribable to four main different classes (Lapatas *et al.*, 2015):

- *Tables*: In table formats, data are organized in a table in which the columns are separated by tabs, commas, pipes, etc., depending on the source generating the file.
- *FASTA-like*: FASTA-like files utilise, for each data record, one or more “definition” or “declaration lines”, which contain metadata information or specify the content of the following lines. Definition/declaration lines usually start with a special character or keyword in the first position of the line – a “>” in FASTA files or a “@” in fastq or SAM files – followed by lines containing the data themselves. In some cases, declaration lines may be interspersed with data lines. This format is mostly used for sequence data.
- *GenBank-like*: In the GenBank-like format, each line starts with an identifier that specifies the content of the line.
- *tag-structured*: Tag-structured formatting uses “tags” “(“,”)”, “{“, “}”, etc. to make data and metadata recognisable with high specificity. Tag-structured text files, especially XML and JSON, are being increasingly employed as data interchange formats between different programming languages.

There are also examples of data files using different representations for data and metadata. This means that two or more format classes may be used in the same data file. Some authors propose to adopt XML for biological data interchange between databases.

Concluding Remarks

Standards adoption facilitates data integration and sharing of omics data. They improve interoperability by overcoming problems relative to different data formats, architectures, and naming conventions. The definition and usage of standards increases productivity and fosters availability of a major volume of data to researchers.

See also: Bioinformatics Data Models, Representation and Storage. Data Storage and Representation. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Text Mining for Bioinformatics Using Biomedical Literature

References

- Benson, D.A., Cavanaugh, M., Clark, K., *et al.*, 2013. GenBank. *Nucleic Acids Research* 41 (Database issue), D36–D42.
- Chervitz, S.A., *et al.*, 2011. Data standards for Omics data: The basis of data sharing and reuse. *Methods in Molecular Biology* (Clifton, N.J.) 719, 31–69.
- Danecek, P., *et al.*, 2011. The variant call format and VCF tools. *Bioinformatics* 27, 2156–2158.
- Huser, V., Sastry, C., Breymaier, M., Idriss, A., Cimino, J.J., 2015. Standardizing data exchange for clinical research protocols and case report forms: An assessment of the suitability of the Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model (ODM). *Journal of Biomedical Informatics* 57, 88–99.
- Kulikova, T., *et al.*, 2007. EMBL nucleotide sequence database in 2006. *Nucleic Acids Research* 35, D16–D20.
- Lapatas, V., Stefanidakis, M., Jimenez, R.C., Via, A., Schneider, M.V., 2015. Data integration in biological research: An overview. *Journal of Biological Research – Thessaloniki* 22 (9).
- Li, H., *et al.*, 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25, 2078–2079.
- Smith, B., *et al.*, 2007. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25, 1251.
- Stoesser, G., *et al.*, 2002. The EMBL nucleotide sequence database. *Nucleic Acid Research* 30, 21–26.

Relevant Websites

- www.cdisc.org
CDISC.
- <https://clsi.org/>
Clinical & Laboratory Standards Institute.
- http://www.bioinformatics.nl/tools/crab_fasta.html
FASTA format.
- www.ga4gh.org
Global Alliance for Genomics and Health (GA4GH).
- www.hupo.org
HUPO.
- www.insdc.org
INSDC: International Nucleotide Sequence Database Collaboration.
- <https://iupac.org/>
IUPAC.
- <https://www.bioinformatics.org/sms/iupac.html>
IUPAC Codes - Bioinformatics.org.

www.metabolomics-msi.org/

MSI (Metabolomics Standards Initiative).

www.rd-alliance.org

Research Data Alliance.

<https://www.ncbi.nlm.nih.gov/genbank/samplerecord/>

Sample GenBank Record - NCBI - NIH.

www.combine.org/

The 'COmputational Modeling in Biology' Network.

www.obofoundry.org/

The OBO Foundry.

Standards and Models for Biological Data: FGED and HUPO

Barbara Calabrese, University “Magna Graecia”, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The increase in the rate and amount of data currently being generated thanks to new high-throughput sequencing (HTS) technologies poses challenges in the data collection, management and sharing. Specifically, HTS is used not only for traditional applications in genomics, but also to assay gene expression (RNA-seq), transcription factor binding, DNA methylation. Availability of the data generated through HTS technologies in usable formats is essential not only for peer-review process and to guarantee experimental reproducibility, but also to allow integration of multiple experiments across multiple modalities (Brazma *et al.*, 2001).

Within the field of proteomics, mass spectrometry (MS) delivers ever-faster cycle times with high sensitivity and high quality MS spectra. Improved separation techniques have increased the rate at which samples can be fed into these machines and protein identification algorithms can rapidly search high quality protein sequence databases, and assign an ever-increasing proportion of spectra as being generated from a specific protein fragment. All such data is of value to the researcher and needs to be made accessible in an easily accessible form. Once validated, the data is then available to act as a reference set against which other experimental results can be compared. In order to do this, the original data needs to be stored in a format appropriate for the researcher to access, download and analyse.

This paper aims to examine the major standard initiatives in genomics and proteomics fields.

FGED

The Functional Genomics Data (FGED) Society, was founded in 1999 as the MGED (Microarray and Gene Expression Data) Society because its original focus is on microarrays and gene expression data. In July 2010, the society changed its name to the “Functional Genomics Data (FGED) Society” to reflect its current mission which goes beyond microarrays and gene expression to encompass data generated using any functional genomics technology applied to genomic-scale studies of gene expression, binding, modification (such as DNA methylation), and other related applications.

They work with other organizations to accelerate and support the effective sharing and reproducibility of functional genomics data. They facilitate the creation and use of standards and software tools that allow researchers to annotate and share their data easily. Finally, they promote scientific discovery that is driven by genome wide and other biological research data integration and meta-analysis.

The major standardization projects being pursued by the FGED Society include:

- MIAME – The formulation of the minimum information about a microarray experiment required to interpret and verify the results.
- MINSEQE – The development of the Minimum Information about a high-throughput SEQuencing Experiment standard for Ultra High-Throughput Sequencing experiments.
- MAGE-TAB – A simple spreadsheet-based, MIAME-supportive format for microarray experimental data called MAGE-TAB, based on a richer data exchange and object modelling format known as MAGE.
- Annotare – A stand-alone desktop application to help bench biologists annotate biomedical investigations and their resulting data.
- Ontology – The development of ontologies for microarray experiment description and biological material (biomaterial) annotation in particular.
- Collaborative standards – Engaging with and supporting the efforts of other relevant standards organizations, such as MIBBI (Minimum Information for Biological and Biomedical Investigations) (Taylor, 2008), ISA-TAB (Investigator/Study/Assay Infrastructure, see “Relevant Websites section”) (Sansone, 2012), OBI (Ontology for Biomedical Investigations, see “Relevant Websites section”).

MIAME

MIAME describes the Minimum Information About a Microarray Experiment that is needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment (Brazma *et al.*, 2001).

The six most critical elements contributing towards MIAME are:

- The raw data for each hybridisation (e.g., CEL or GPR files).
- The final processed (normalised) data for the set of hybridisations in the experiment (study) (e.g., the gene expression data matrix used to draw the conclusions from the study).

- The essential sample annotation including experimental factors and their values (e.g., compound and dose in a dose response experiment).
- The experimental design including sample data relationships (e.g., which raw data file relates to which sample, which hybridisations are technical replicates, which are biological replicates).
- Sufficient annotation of the array (e.g., gene identifiers, genomic coordinates, probe oligonucleotide sequences or reference commercial array catalog number).
- The essential laboratory and data processing protocols (e.g., what normalisation method has been used to obtain the final processed data).

MINSEQE

MINSEQE describes the Minimum Information about a high-throughput nucleotide SEQuencing Experiment, that is needed to enable the unambiguous interpretation and facilitate reproduction of the results of the experiment (MINSEQE, 2012). By analogy to the MIAME guidelines for microarray experiments, adherence to the MINSEQE guidelines will improve integration of multiple experiments across different modalities, thereby maximising the value of high-throughput research.

The five elements of experimental description considered essential when making data available supporting published high-throughput sequencing experiments are as follows:

- The description of the biological system, samples, and the experimental variables being studied: “compound” and “dose” in dose-response experiments or “antibody” in ChIP-Seq experiments, the organism, tissue, and the treatment(s) applied.
- The sequence read data for each assay: read sequences and base-level quality scores for each assay; FASTQ format is recommended, with a description of the scale used for quality scores.
- The ‘final’ processed (or summary) data for the set of assays in the study: the data on which the conclusions in the related publication are based, and descriptions of the data format. Currently there are no widely adopted formats for processed HTS data, thus the descriptions of the data format should be provided. For gene expression, in many cases, these data can be presented as a matrix, with each row corresponding to a genomic region (such as an gene), each column representing a particular biological state (e.g., a time point in a time b course experiment), and each element in the matrix representing a measurement of the particular genomic region in the particular biological state. Similarly, other applications like ChIP=Seq analyses typically generate tabular output.
- General information about the experiment and sample-data relationships: a summary of the experiment and its goals, contact information, any associated publication, and a table specifying sample-data relationships.
- Essential experimental and data processing protocols: how the nucleic acid samples were isolated, purified and processed prior to sequencing, a summary of the instrumentation used, library preparation strategy, labelling and amplification methodologies. Moreover, data processing and analysis protocols must be described in sufficient detail to enable unambiguous data interpretation and to enable scientists to reproduce the analysis steps. This should include, but is not limited to, data rejection methods, data correction methods, alignment methods, data smoothing and filtering methods and identifiers used for reference genomes to which the sequences were mapped.

MAGE-TAB

The MAGE project aims to provide a standard for the representation of microarray expression data that would facilitate the exchange of microarray information between different data systems. Sharing of microarray data within the research community has been greatly facilitated by the development of the MIAME and MAGE-ML (Microarray and Gene Expression Markup Language) standards by the FGED Society. However, the complexity of the MAGE-ML format has made its use impractical for laboratories lacking dedicated bioinformatics support. MAGE-TAB is a simple tab-delimited, spreadsheet-based format, which will become a part of the MAGE microarray data standard and can be used for annotating and communicating microarray data in a MIAME compliant fashion. MAGE-TAB enables laboratories without bioinformatics experience or support to manage, exchange and submit well-annotated microarray data in a standard format using a spreadsheet. The MAGE-TAB format is self-contained, and does not require an understanding of MAGE-ML or XML.

MGED Ontology and Ontology for Biomedical Investigations

The purpose of the MGED ontology was to provide, either directly or indirectly, the terms needed to follow the MIAME guidelines and referenced by MAGE. The MGED ontology was an ontology of experiments, specifically microarray experiments, but it was potentially extensible to other types of functional genomics experiments. Although the major component of the ontology involved biological descriptions, it was not an ontology of molecular, cellular or organismal biology. Rather, it was an ontology that included concepts of biological features relevant to the interpretation and analysis of an experiment (Stoeckert and Parkinson, 2003).

The original MGED Ontology (MO) is being incorporated into the Ontology for Biomedical Investigations (OBI). The Ontology for Biomedical Investigations (OBI, see “Relevant Websites section”) is build in a collaborative, international effort and will serve as a resource for annotating biomedical investigations, including the study design, protocols and instrumentation used,

the data generated and the types of analysis performed on the data. This ontology arose from the Functional Genomics Investigation Ontology (FuGO) and will contain both terms that are common to all biomedical investigations, including functional genomics investigations and those that are more domain specific (Bandrowski *et al.*, 2016).

ANNOTARE

Annotare (see “Relevant Websites section”) is a tool to help biologists annotate biomedical investigations and their resulting data. Annotare is a stand-alone desktop application that features the following:

- A set of intuitive editor forms to create and modify annotations,
- Support for easy incorporation of terms from biomedical ontologies,
- Standard templates for common experiment types,
- A design wizard to help create a new document, and
- A validator that checks for syntactic and semantic violation.

Annotare will help a biologist construct a MIAPE-compliant annotation file based on the MAGE-TAB format.

HUPO

The Human Proteome Organisation (HUPO) was formed in 2001 to consolidate national and regional proteome organizations into a single worldwide body. The Proteome Standards Initiative (PSI) was established by HUPO with the aim of standardizing data representation within the field of proteomics to the end that public domain databases can be established where all such data can be deposited, exchanged between such databases or downloaded and utilized by laboratory workers.

The HUPO-PSI organized a series of meetings at which data producers, data users, instrumentation vendors and analytical software producers gathered to discuss the problem. As the HUPO-PSI is a completely voluntary organisation with limited resources, activity is focused on a few key areas of proteomics, constituting the PSI work groups. Currently, there are the following work groups (Orchard and Henjakob, 2007):

- Molecular Interactions (MI): The Molecular Interactions working group is concentrating on: (i) improving the annotation and representation of molecular interaction data wherever it is published, e.g., in journal articles, authors web-sites or public domain databases (Orchard *et al.*, 2011, 2007, 2012; Bourbeillon *et al.*, 2010) and (ii) improving the accessibility of molecular interaction data to the user community by presenting it in a common standard data format (PSI-MI XML (Proteomics Standards Initiative-Molecular Interactions/MITAB (Molecular Interaction TAB, a tab-delimited data exchange format, developed by the HUPO Proteomics Standards Initiative). Thus, the data can be downloaded from multiple sources and easily combined using a single parser.
- Mass Spectrometry (MS): The PSI-MS working group defines community data formats and controlled vocabulary terms facilitating data exchange and archiving in the field of proteomics mass spectrometry (Mayer *et al.*, 2013, 2014).
- Proteomics Informatics (PI): The main current deliverable of the Proteomics Informatics working group is the mzIdentML data exchange standard (previously known as analysis XML).
- Protein Modifications (PSI-MOD): The protein modification workgroup focuses on developing a nomenclature and providing an ontology available in OBO format or in OBO.xml (Montecchi Palazzi *et al.*, 2008).
- Protein Separation (PS): The PSI Protein Separation work group is a collaboration of researchers from academia, industrial partners and software vendors. The group aims to develop reporting requirements that supplement the MIAPE parent document, describing the minimum information that should be reported about gel-based (Gibson *et al.*, 2008; Hoogland *et al.*, 2010), and non-gel based separation technologies (Domann *et al.*, 2010; Jones *et al.*, 2010) employed for proteins and peptides in proteomics. The group will also develop data formats for capturing MIAPE-compliant data about these technologies (Gibson *et al.*, 2010) and supporting controlled vocabularies.

The standard deliverables of each work group are:

- Minimum Information Specification: for the given domain, this specifies the minimum information required for the useful reporting of experimental results in this domain.
- Formal exchange format for sharing experimental results in the domain. This will usually be an XML format, capable of representing at least the Minimum Information, and normally significant additional detail.
- Controlled vocabularies.
- Support for implementation of the standard in publicly available tools.

PSI-MI XML

It is a community data exchange format for molecular interactions which has been jointly developed by major data providers from both academia and industry. This format is based on XML and is stable and used for several years (Kerrien, 2007). It can be used for storing any kind of molecular interactions data:

- Complexes and binary interactions;
- Not only protein-protein interactions, but also nucleic acids interactions and others;
- Hierarchical complexes hierarchical complexes modelling by using interaction Ref in participants instead of an interactor.

Data representation in PSI-MI 2.5 XML relies heavily on the use of controlled vocabularies in OBO format. These vocabularies are essential for standardizing not only the syntax, but also the semantics of the molecular interactions representation. PSI-MI 2.5 standard defines also a simple tabular representation (MITAB).

Two different flavours of this format have been developed: the **Compact** format and the **Expanded** format. In the Compact format, the description of experiments and interactors is done at the beginning of the **entry** using **experimentList** and **interactorList** elements. When describing the interactions (in the next element **interactionList**), we will use references to the previously described experiments and interactors using their **id** attributes.

In the Expanded format, the description of experiments and interactors is done within each interaction. The file doesn't contain any **experimentList** or **interactorList** at the **entry** level, it only contains an **interactionList** (see "Relevant Websites section").

mzML

mzML is the PSI standard file format for mass spectrometry output files (Martens *et al.*, 2011). The full technical details of the mzML standard are available online, together with complete specification documentation, graphical depictions of its structure, and various example files at "see Relevant Websites section". All of the information from a single MS run, including the spectra and associated metadata, is contained within the mzML file. mzML is encoded in XML. An XML schema definition defines the format structure, and many industry-standard tools are readily available to validate whether an XML document conforms to its XML schema definition.

The overall mzML file structure is as follows (elements presented top-to-bottom): **<cvList>** contains information about the controlled vocabularies referenced in the rest of the mzML document; **<fileDescription>** contains basic information on the type of spectra contained in the file; **<referenceableParamGroupList>** is an optional element that of groups of controlled vocabulary terms that can be referenced as a unit throughout defines a list the document; **<sampleList>** can contain information about samples that are referenced in the file; **<instrumentConfigurationList>** contains information about the instrument that generated the run; **<softwareList>** and **<dataProcessingList>** provide a history of data processing that occurred after the raw acquisition; **<acquisitionSettingsList>** is an optional element that stores special input parameters for the mass spectrometer, such as inclusion lists. These elements are followed by the acquired spectra and chromatograms. Both spectral and chromatographic data are represented by binary format data encoded into base 64 strings, rather than human-readable ASCII text for enhanced fidelity and efficiency when dealing with profile data.

Mzidentml

The Proteomics Informatic working group is developing standards for describing the results of identification and quantification processes for proteins, peptides and protein modifications from mass spectrometry. mzIdentML is an exchange standard for peptide and protein identification data. The mzIdentML standard is, similar to mzML. The mzIdentML format stores peptide and protein identifications based on mass spectrometry and captures metadata about methods, parameters, and quality metrics. Data are represented through a collection of protein sequences, peptide sequences (with modifications), and structures for capturing the scores associated with ranked peptide matches for each spectrum searched.

Concluding Remarks

Omics technologies represent a fundamental tool for discovery and analysis in the life sciences. With the rapid advances, it has become imperative to provide a standard output format for data that will facilitate data sharing and analysis. To resolve the issues associated with having multiple formats, vendors, researchers, and software developers established standard initiatives, such as FGED or HUPO PSI to develop a single standard. They propose new data formats for genomic and proteomics data, adding a number of improvements, including features such as a controlled vocabulary and/or ontologies with validation tools to ensure consistent usage of the format and immediately available implementations to facilitate rapid adoption by the community.

See also: Bioinformatics Data Models, Representation and Storage. Data Storage and Representation. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Text Mining for Bioinformatics Using Biomedical Literature

References

- Bandrowski, A., *et al.*, 2016. The ontology for biomedical investigations. PLOS ONE 11 (4), e0154556.
- Bourbeillon, J., *et al.*, 2010. Minimum information about a protein affinity reagent (MIAPAR). Nature Biotechnology 28, 650–653.
- Brazma, A., Hingamp, P., Quackenbush, J., *et al.*, 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nature Genetics 29 (4), 365–371.

- Domann, P.J., *et al.*, 2010. Guidelines for reporting the use of capillary electrophoresis in proteomics. *Nature Biotechnology* 28 (7), 654–655.
- Gibson, F., *et al.*, 2008. Guidelines for reporting the use of gel electrophoresis in proteomics. *Nature Biotechnology* 2 (68), 864.
- Gibson, F., *et al.*, 2010. The gel electrophoresis markup language (GelML) from the Proteomics Standards Initiative. *Proteomics* 10, 3073–3081.
- Hoogland, C., *et al.*, 2010. Guidelines for reporting the use of gel image informatics in proteomics. *Nature Biotechnology* 28, 655–656.
- Jones, A.R., *et al.*, 2010. Guidelines for reporting the use of column chromatography in proteomics. *Nature Biotechnology* 28 (7), 654.
- Kerrien, S., *et al.*, 2007. Broadening the horizon—level 2.5 of the HUPO-PSI format for molecular interactions. *BMC Biology* 5, 44.
- Martens, L., *et al.*, 2011. mzML — A community standard for mass spectrometry data. *Mol Cell Proteomics* 10 (1), R110.00013.
- Mayer, G., *et al.*, 2013. The HUPO proteomics standards initiative- mass spectrometry controlled vocabulary. *Database* (Oxford).
- Mayer, G., *et al.*, 2014. Controlled vocabularies and ontologies in proteomics: Overview, principles and practice. *Biochimica et Biophysica Acta* 1844, 98–107.
- MINSEQE, 2012. Minimum Information about a high throughput Nucleotide SeQuencing Experiment a proposal for standards functional genomic data reporting. Version 1.0.
- Montecchi Palazzi, L., *et al.*, 2008. The PSI-MOD community standard for representation of protein modification data. *Nature Biotechnology* 26 (8), 864–866.
- Orchard, S., Henjakob, H., 2007. The HUPO-proteomics standards initiative —easing communication and minimizing data loss in a changing world. *Briefings in Bioinformatics* 9 (2), 166–173.
- Orchard, S., *et al.*, 2007. The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nature Biotechnology* 25, 894–898.
- Orchard, S., *et al.*, 2011. Minimum information about a bioactive entity (MIABE). *Nature Reviews Drug Discovery* 10, 661–669.
- Orchard, S., *et al.*, 2012. Protein interaction data curation: The International Molecular Exchange (IMEx) consortium. *Nature Methods* 9, 345–350.
- Sansone, S.A., *et al.*, 2012. Toward interoperable bioscience data. *Genetics* 44, 121–126.
- Stoeckert, C.J., Parkinson, H., 2003. The MGED ontology: A framework for describing functional genomics experiments. *Comparative and Functional Genomics* 4 (1), 127–132.
- Taylor, C.F., *et al.*, 2008. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nature Biotechnology* 26, 889–896.

Relevant Websites

<http://code.google.com/p/annotate/>
Annotare.

<http://isa-tools.org/>
ISA tools.

<http://www.psidev.info/groups/molecular-interactions>
Molecular Interactions.

<http://obi-ontology.org/>
OBI.

<http://obi-ontology.org>
OBI.

<http://www.psidev.info/index.php?q=node/257>
psidev.

Standards and Models for Biological Data: SBML

Giuseppe Agapito, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Life science involves the scientific study of living organisms. Understanding how living organisms work is one of the greatest challenges that researchers are facing. Understanding how an organism work is challenging since it is necessary to identify all the interactions that happen inside the cells. Numerous experimental techniques (Abu-Jamous *et al.*, 2015) have been proposed such as, Polymerase Chain Reaction (PCR), Western Bolt, microarrays, Next Generation Sequencing (NGS) that have made possible to identify the components (known as biomolecules) inside the cell and how interacting to regulate the proper functioning of an organism. As well as the interactions between large molecules DNA, RNA, proteins and small molecules sugars, lipids, hormones, and vitamins. An in-depth knowledge of how these biomolecules interact among them is mandatory in order to understand how the cellular machinery works and the cellular processes are regulated. To develop a simple and efficient formalism able to convey how a cell work, how cellular processes are regulated and how the cell responds to internal and external stimuli, is mandatory. Thus, the interactions among the different types of molecules inside and outside the structure of the cell that regulate its functioning can be represented by means of networks. In particular, the variety of interactions between genes, proteins, and metabolites are well caught by network representations. In the simplest approximation, large and small biomolecules can be represented as nodes in a network whose edges represent various types of interaction between the nodes. This has given rise to the concept of system biology, which pursues the objective to provide the tools and the formalisms necessary to understand how the single biomolecules interact and evolve. The field of systems biology encompasses scientists with extremely various backgrounds comprising biologists, biochemists, clinicians, physiologists, mathematicians, physicists, computer scientists, and engineers. This has made it possible to realise models able to reproduce the behavior of a cell or even of a whole organism (Faro *et al.*, 2011; Eschrich *et al.*, 2009). Such models can speed up the investigation of the physiological mechanisms at the base of human diseases as well as improve the development and testing of new and more efficient drugs. To achieve this goal, it is essential that researchers can exchange data at their disposal, as this would allow for quicker explanations to complex illnesses such as cancer. Although data exchange would seem a very simple step to accomplish, it is not so immediate because of the absence of a single format for the representation of the biological system. This abundance of formats used to represent biological systems has lead to numerous problems, for example, (i) Researchers often need to use several different tools to make it possible to integrate data coming from different databases, or even worst manually re-encode data to be further analyzed, a time-consuming and error-prone process; (ii) Manually data manipulation made data stranded and unusable. Resulting in loss of re-usability, especially after data manipulation that makes data not more compatible with the original database. The current inability to exchange biological systems data models could be overcome by using a standard format for describing system biological models. To address those issues, it is necessary to develop a universal standard able to represent and exchange system biological models. In particular, to model ordinary differential equations (ODE) the standard language Systems Biology Markup Language (SBML) (Hucka *et al.*, 2003) has been proposed for efficient exchanging, storing and modeling of system biological models.

SBML is a computer-readable XML-based language for representing and exchanging models in a unique and universal format. The eXtensible Markup Language – XML (Bray *et al.*, 1997) is a markup language designed to be self-descriptive. XML stores data in plain text format, providing software and hardware independent method to store, transport, and share data. Due to this features, XML became like a standard data language even for bioinformatics. The adoption of SBML to store, exchange, and share data, would help to solve the problems of interoperability. In this way, users would be better capable of spending more time on research task rather than on struggling with data format issues.

Systems Biology Markup Language

SBML is a machine-readable format to represent biological models. In particular, SBML is focused on describing systems where biological entities are involved in, and modified by the processes that occur over time. An example of this is a biological network. SBML can represent models including cell signaling pathways, metabolic pathways, biochemical reactions, gene regulation, and many others. SBML allow describing biological models into a formal and computable format that can be analyzed rigorously by using scientific methods. As well as, SBML allow to represent models in several different representations related to its own biological scenarios. The primary target of SBML is to provide a standard format able to allow system biological data exchanging, storing and reusability. SBML is not a universal language for representing biological models. It would be impossible to produce a one-size-fits-all universal language for the biological systems. The use of SBML simplifies the sharing and analyzing of models among of multiple tools, without that users have to worry about to make it compatible with each software tool. SBML is independent of the programming languages and software tools, enabling the encoding of the biological models by means of XML.

By supporting SBML for reading and writing models, software tools can straightforwardly share, elaborate and store the biological models, as well as enabling the sharing among scientists of the outcomes obtained from well-defined models. Allowing to improve and speed-up the understanding of complex biological phenomena. SBML is a modular language, consisting of a comprehensive core that can be used alone. The most recent specification of SBML is SBML Level 3. The modular core of SBML allows adding new features in an easy way. SBML Levels are intended to coexist, that is SBML Level 3 does not make Level 2 obsolete, i.e., all models that have been writing by using SBML Level 2 are compatible with all software tools still continue to be used.

SBML Structure

SBML's markups define a set of rules for encoding biological models that are human and machine readable. A model definition in SBML Levels 3 Version 1 comprises a list of the following elements.

- Function definition: a mathematical function that could be used to describe the biological model;
- Unit definition: units can be used in the expression of quantities in a model where species may be located.
- Compartment: a container of limited dimensions where species may be placed. Compartments may or not describe real physical structures.
- Unit definition: a named definition of a new unit of measurement. Named units can be used in the expression of quantities in a model.
- Species: a set of entities of the same kind located in a compartment and participating in reactions process. SBML can represent any entity that makes sense in the context of a given model.
- Parameter: in SBML, the term parameter is used to indicate constants and or variables into a model. Moreover, SBML Level 3 provides the ability to define global parameters for a model as well as local parameters to a single process.
- Initial assignment: used to determine the initial conditions of a model by means of mathematical expressions. This value is used to define the value of a variable at the start of simulated time.
- Rule: mathematical expressions used to define how a variable's value can be calculated from other variables. Making possible to infer the behavior of the model with respect to time.
- Constraint: a means of detecting out-of-bounds conditions during a dynamical simulation and are defined by a general mathematical expression.
- Reaction: a statement describing transformation, transport or binding process that can change the amount of one or more elements.
- Event: a statement describing changes in one or more variables of any type (species, compartment, parameter, etc.) when a condition is satisfied.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2" level="2"
  version="1" xmlns:html="http://www.w3.org/1999/xhtml">
<model id="Ec_iAF1260" name="Ec_iAF1260">
<listOfUnitDefinitions>
  <unitDefinition id="mmol_per_gDW_per_hr">
    <listOfUnits>
      <unit kind="mole" scale="-3"/>
      <unit kind="gram" exponent="-1"/>
      <unit kind="second"
        multiplier=".00027777"
        exponent="-1"/>
    </listOfUnits>
  </unitDefinition>
</listOfUnitDefinitions>
<listOfCompartments>
  <compartment id="Extra_organism"/>
  <compartment id="Periplasm">
```

```

        outside="Extra_organism" />
    <compartment id="Cytosol" outside="Periplasm" />
</listOfCompartments>
<listOfSpecies>
    <species id="M_10fthf_c"
        name="M_10_Formyltetrahydrofolate_C20H21N7O7"
        compartment="Cytosol" charge="-2"
        boundaryCondition="false" />
    <species id="M_12dgr120_c"
        name="M_1_2_Diacyl_sn_glycerol__didodecanoyl
        .....n_C120__C27H52O5"
        compartment="Cytosol" charge="0"
        boundaryCondition="false" />
    <species id="M_12dgr120_p"
        name="M_1_2_Diacyl_sn_glycerol__
        .....didodecanoyl__n_C120__C27H52O5"
        compartment="Periplasm"
            charge="0"
            boundaryCondition="false" />
    .....

```

Code 2.1: This is a part of the genome-scale *E. coli* reconstruction and contains central metabolism reactions, encoded by using the SBML Level 2 format.

SBML Packages

SBML Level 3 has been designed in a modular fashion. This makes it possible to use the core specification independently or to add extra packages onto the core to provide further features.

- **Hierarchical Model Composition:** the Hierarchical Model Composition package provides the ability to combine models as submodels inside another model. A feature that makes it possible to decompose larger models into smaller ones, to avoid duplication of elements; and finally create reusable libraries.
- **Flux Balance Constraints:** the Flux Balance Constraints package provides support to analyze and study biological networks.
- **Qualitative Models:** the Qualitative Models package provide the tools to represent models partially that is, biochemical reactions and their kinetics are not entirely known.
- **Layout:** the SBML layout package contains the guidelines to represent a reaction network in a graphical form. SBML layout package only deals with the information necessary to define the position and other aspects of a graph's layout; the additional details regarding the rendering, are provided in a separate package called Rendering package.

SBML Specification Differences

SBML is defined in a set of Requirement documents describing the elements of the language, the syntax, and validation rules. SBML Levels are intended to coexist. Thus SBML Level 3 does not render Level 2 and Level 1 obsolete. As a good rule for each Level, the latest Version should be used. SBML is an active project under continuous updating and developed in collaboration with an international community of researchers and software developers.

SBML Level 1 Version 2

Systems Biology Markup Language (SBML) Level 1, Version 2 is a description language for simulations in systems biology. The goals pursuit by SBML are, allow to users to represent biochemical networks, including cell signaling pathways, metabolic pathways, biochemical reactions, and many others. SBML has been developed has an XML-based format for coding systems

biology models in a simple format that software tools can manage and exchange. However, for easier communication to human readers, there is a visual version of SBML developed upon the Unified Modeling Language graphical language (UML) the (Eriksson and Penker, 1998). UML-based definition is necessary to define the XML Schema (Biron and Malhotra, 2000) for SBML. The top-level components of the SBML Level 1, Version 2 model are the following:

- Unit definition: a name for a unit used in the expression of quantities in a model.
- Compartment: a container of finite volume for substances. In SBML Level 1, a compartment is primarily a topological structure with a size but no geometric qualities.
- Species: a substance or entity that takes part in a reaction. Some example species are ions such as and molecule such as glucose or ATP.
- Reaction: a statement describing transformation, transport or binding process that can change the amount of one or more elements.
- Parameter: in SBML, the term parameter is used to indicate constants and or variables into a model. Moreover, SBML Level 3 provides the ability to define global parameters for a model as well as local parameters to a single process.
- Rule: in SBML, a mathematical expression that is added to the differential equations constructed from the set of reactions and can be used to set parameter values, establish constraints between quantities, etc.

A software framework can read a model conveyed in SBML and translate it into its own internal format for model analysis. For instance, a framework might provide the tools to simulate a model by means of a set of differential equations representing the network, to perform a numerical integration to investigate the model's dynamic behavior. SBML allows representing models of arbitrary complexity. Each component type present in a model is described through a particular data structure type, able to organize the important information. The data structures determine how the resulting model is encoded in XML. To get more information on the items introduced before, it is advised to consult the specification reported at the following address: <http://co.mbine.org/specifications/sbml.level-1.version-2.pdf>.

SBML Level 2 Version 5 Release 1

The Systems Biology Markup Language (SBML) Level 2 Version 5 Release 1 is a model representation format for systems biology. The main intend of SBML Level 2 Version 5 Release 1 is to provide a formal language to represent biochemical networks. Also, SBML project is not an attempt to define a universal language for representing biological system models. SBML Level 2 Version 5 Release 1 allows to describe models of arbitrary complexity. Each type of component in a model is described using a particular type of data object that organizes the relevant information. The head level of an SBML model definition consists of lists of these components, with every list being optional, the meaning of each component is as follows:

- Function definition: a named mathematical function that may be used throughout the rest of a model.
- Unit definition: a named definition of a new unit of measurement, or a redefinition of an SBML predefined unit.
- Compartment type: a type of location where reacting entities such as chemical substances may be located.
- Species type: a type of entity that can participate in reactions. Typical examples of species types include ions such as Ca²⁺, molecules such as glucose or ATP, and more.
- Compartment: a container of finite volume for substances. In SBML Level 1, a compartment is primarily a topological structure with a size but no geometric qualities.
- Species: a pool of entities of the same species type located in a particular compartment.
- Parameter: in SBML, the term parameter is used to indicate constants and or variables into a model. Moreover, SBML Level 3 provides the ability to define global parameters for a model as well as local parameters to a single process.
- Initial assignment: used to determine the initial conditions of a model by means of mathematical expressions. This value is used to define the value of a variable at the start of simulated time.
- Rule: mathematical expressions used to define how a variable's value can be calculated from other variables. Making possible to infer the behavior of the model with respect to time.
- Event: a statement describing changes in one or more variables of any type (species, compartment, parameter, etc.) when a condition is satisfied.
- Constraint: a means of detecting out-of-bounds conditions during a dynamical simulation and are defined by a general mathematical expression.
- Reaction: a statement describing transformation, transport or binding process that can change the amount of one or more elements.

To get more detailed information on the items introduced before, it is advised to consult the specification reported are the following address: <http://co.mbine.org/specifications/sbml.level-2.version-5.release-1.pdf>.

SBML Level 3 Version 1 core release 2

Major editions of SBML are termed levels and represent substantial changes to the composition and structure of the language. The SBML Level 3 Version 1 core release 2 (Hucka *et al.*, 2010, 2015), represents an evolution of the first version of the language.

SBML Level 3 Version 1 core release 2 allows models of arbitrary complexity to be represented. Each type of component in a model is described using a particular type of data object that organizes the relevant information. The top level of an SBML model

definition consists of lists of these components, with every list being optional, the meaning of each component is as follows: The top level of an SBML model definition consists of lists of these components, with every list being optional, the meaning of each component is as follows:

- Function definition: a named mathematical function that may be used throughout the rest of a model.
- Unit definition: a named definition of a new unit of measurement. Named units can be used in the expression of quantities in a model.
- Compartment: a well-stirred container of a finite size where species may be located. Compartments may or may not represent actual physical structures.
- Species: a pool of entities of the same species type located in a particular compartment.
- Parameter: in SBML, the term parameter is used to indicate constants and or variables into a model. Moreover, SBML Level 3 provides the ability to define global parameters for a model as well as local parameters to a single process.
- Initial assignment: used to determine the initial conditions of a model by means of mathematical expressions. This value is used to define the value of a variable at the start of simulated time.
- Rule: mathematical expressions used to define how a variable's value can be calculated from other variables. Making possible to infer the behavior of the model with respect to time.
- Constraint: a means of detecting out-of-bounds conditions during a dynamical simulation and are defined by a general mathematical expression.
- Reaction: a statement describing transformation, transport or binding process that can change the amount of one or more elements.
- Event: a statement describing an immediate, discontinuous change in one or more symbols of any type (species, compartment, parameter, etc.) when a condition is satisfied.

To get more detailed information on the items introduced before, it is advised to consult the specification reported are the following address: <http://co.mbine.org/specifications/sbml.level-3.version-1.core.release-2>.

In brief, the main differences between the 3 versions of SBML are the number of available components. Compared to version 1 in subsequent releases, components have increased, making it possible to describe a larger number of biological systems. Also, from the version two, all the elements are optional, making it easier to explain phenomena in a partial way. From version 2 to version 3, the compartment types and species types have been removed, since in version 3 the compartment component can contain several different entities, thus rendering obsolete the use of components types and species types.

See also: Bioinformatics Data Models, Representation and Storage. Data Formats for Systems Biology and Quantitative Modeling. Data Storage and Representation. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Quantitative Modelling Approaches. Text Mining for Bioinformatics Using Biomedical Literature

References

- Abu-Jamous, B., Fa, R., Nandi, A.K., 2015. High-throughput technologies. In: *Integrative Cluster Analysis in Bioinformatics*, pp. 53–66.
- Biron, P., Malhotra, A., 2000. Xml schema part 2: Datatypes (w3c candidate recommendation 24 october 2000). Available at: <https://www.w3.org/TR/xmlschema-2/>.
- Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F., 1997. Extensible markup language (xml). *World Wide Web Journal* 2 (4), 27–66.
- Eriksson, H.-E., Penker, M., 1998. Uml Toolkit. John Wiley & Sons. Inc.
- Eschrich, S., Zhang, H., Zhao, H., *et al.*, 2009. Systems biology modeling of the radiation sensitivity network: A biomarker discovery platform. *International Journal of Radiation Oncology, Biology, Physics* 75 (2), 497–505.
- Faro, A., Giordano, D., Spampinato, C., 2011. Combining literature text mining with microarray data: Advances for system biology modeling. *Briefings in Bioinformatics*. bbr018.
- Hucka, M., Bergmann, F.T., Drager, A., *et al.*, 2015. Systems biology markup language (SBML) level 2 version 5: Structures and facilities for model definitions. *Journal of Integrative Bioinformatics* 12 (2), 731–901.
- Hucka, M., Bergmann, F.T., Hoops, S., *et al.*, 2010. The systems biology markup language (SBML): Language specification for level 3 version 1 core. *Journal of Integrative Bioinformatics* 12, 226.
- Hucka, M., Finney, A., Sauro, H.M., *et al.*, 2003. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19 (4), 524–531.

Standards and Models for Biological Data: BioPAX

Giuseppe Agapito, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

With the continuous evolution of biotechnologies, the study of biological systems is moving toward a molecular level. This changing is due to the capabilities of the new technologies to capture a significant amount of information on molecular changes associated with the particular condition of biological systems. Although the evolution of innovative technologies has made it possible to the sequencing of the whole human genome, the analysis of the vast amount of data available from the sequencing of the genome presents several difficulties. These difficulties are related with the intricate networks of interactions that regulate how a cell works, how cellular processes are regulated and how the cell responds to internal and external stimuli. Making difficult (often impossible), to map genome or proteome data together to obtain a clear vision of how to work an organism or a single cell. Thus, became essential to detect, store and model the interactions network among the different types of molecules inside and outside the structure of the cell. These interactions information known as biological pathways, are available in several internet public accessible databases. Researchers frequently need to exchange, integrate these information to improve their research activities, but the high heterogeneity among all this data source contribute to making difficult to retrieve, integrate and store data from multiple databases. To reach the target, it is crucial that researchers can easily exchange their data, making data collection and integration easier. This heterogeneity of formats used to represent biological pathways has to lead to several complications, for example, researchers have to use several different tools to integrate data coming from different databases. Even worst researchers manually have to re-encode data to be further analyzed, a time-consuming and error-prone process. All these issues, limiting the usability of the data, especially after manipulation, making data not more compatible with the original databases. The current inability to exchange, integrate and annotate biological pathways data models could be overcome by using a unique standard format for biological pathways. The Biological Pathway Exchange (BioPAX, see Relevant Website section) is an OWL-based data format that enables the integration of different pathway data by defining a data format for the exchanging and integration of biological pathways data. By using BioPAX (Demir *et al.*, 2010) data representation format, the data integration reduces to a semantic mapping of the data over the BioPAX, data model. Thus, data exchange makes it possible to obtain uniform pathway data spread in several databases. BioPAX is a computer-readable OWL-based language for representing and exchanging models in a unique and universal format. The W3C Web Ontology Language (OWL) (Bechhofer, 2009; Dean *et al.*, 2004) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, for example, to verify the consistency of that knowledge or to make implicit knowledge explicit. The adoption of BioPAX to store, exchange, and share data, would help to solve the problems of interoperability. In this way, users would be better capable of spending more time on research task rather than on struggling with data format issues.

Biological Pathway Exchange BioPAX

BioPAX is a machine-readable data format to represent and exchange biological Pathways data. Specifically, BioPAX is focused on describing systems where biological entities are involved in, and by the processes that occur over time. An example of this are biological pathways, gene regulation networks and so on. BioPAX can represent models including cell signaling pathways, metabolic pathways, biochemical reactions, gene regulation, and many others. BioPAX allows to describe biological pathways in a formal and computable format that can be analyzed rigorously by using scientific methods. Moreover, BioPAX allows to represent biological pathways as well as states of physical entities, generic physical entities, gene regulation and genetic interactions. The primary goal of BioPAX is to provide a standard format, to make easy exchange, store and integrate biological pathways data. BioPAX is a modular language, consisting of a comprehensive core that can be used alone.

The last version of BioPAX is BioPAX Level 3, that extends BioPAX including states of physical entities, generic physical entities, gene regulation and genetic interactions. BioPAX Level 3 supports the representation of many pathway data available in public databases. The modular core of BioPAX consents adding new features in an easy way. BioPAX Levels are designed to coexist, i.e., BioPAX Level 3 does not make Level 2 and Level 1 obsolete.

The core of BioPAX ontology is based on level and, has been developed to provide an easy and simple instrument to represent biological pathways. The fundamental element of the BioPAX ontology is the root class element. Thus, each level has been designed for the representation of specific types of pathway data, adding a new child component to the root class element. BioPAX Level 1 has been developed to represent only metabolic pathway data, trying to encode other kinds of pathway data with BioPAX Level 1 is possible but may not produce good outcomes. BioPAX Level 2 expands the scope of Level 1, including the representation of molecular binding interactions and hierarchical pathways. BioPAX Level 3 adds support for representation of signal transduction pathways, gene regulatory networks, and genetic interactions.

To get more detailed information it is possible to visit the following web-site: BioPAX.org (see Relevant Website section).

BioPAX Level 3 Ontology Structure

The BioPAX Level 3 ontology presents 5 essential classes: *Entity* class that is the root level and its four child classes: *Pathway*, *Interaction*, *PhysicalEntity* and *Gene*. Let analyze more in detail root class and its child classes.

- Entity class represents a single biological unit used to describe pathways, and comprises the following attributes: *Comment* attribute is used to better describe the data contained into the class. Since an element could have more than one name has been defined the *Synonyms* attribute, to handle multiple name identifier for an element. To track the source of data, the entity class presents the *dataSource* attribute, that contains a description of the data-source origin. Scientific evidence are handled by the *evidence* attribute whereas, multiple external references to the entity are defined by using the *xref* attribute. Finally an entity class can be identified by a name, characteristic managed by using the *name* attribute.

The name attribute contains the standard name and a short name useful to use in graphical contexts.

- Pathway class represents a network of interactions, comprising the following attributes: a pathways is a network of interactions, those interactions are represented by using the *pathwayComponent* attribute. The order with which these interactions happen in the pathway are handled by the *pathwayOrder* attribute. In addition Pathway class presents the availability, comment, dataSource, evidence, name, xref attributes that have the same meaning of the same attribute stated in the Entity class.
- Interaction class describes the interactions between two or more entities. The attribute *interactionType* allows to annotate the interaction by using, i.e., its name. In particular the annotation is usefull to be human-readable and cannot be used for computing tasks. Since an interaction can involve more elements, the involved elements are handled by the *participant* attribute. In addition Interaction class presents the availability, comment, dataSource, evidence, name, xref attributes that have the same meaning of the same attribute stated in the Entity class.
- PhysicalEntity class represents a set of entities, each one with its own physical structure. *cellularLocation* attribute specify a cellular location, i.e., cytoplasm whose characteristics are obtained by referring to Gene Ontology. *feature* specifies the only relevant features for the physical entity, i.e., binding site. *memberPhysicalEntity* attribute should be used to create generic groups of physical entities, however it is recommended do not use this attribute. *notFeature* attribute used to describe lacking features of the physical entity. In addition PhysicalEntity class presents the availability, comment, dataSource, evidence, name, xref attributes that have the same meaning of the respective attribute belonging to the Entity class.
- Gene contains information related to the inheritance properties. The organism where the gene has been found is handled by the attribute *organism*. In addition Gene class presents the availability, comment, dataSource, evidence, name, xref attributes that have the same meaning of the same attribute stated in the Entity class.

To get more information on all the other subclasses available in BioPAX Level 3, it is recommended to consult the BioPAX level 3 documentation available at the following web address: Biopax.org/release/biopax-/level3-documentation.pdf.

BioPAX Level 2 and Level 1 Ontology Structure

The BioPAX Level 2 and Level 1 ontology present 4 core classes: *Entity* class that is the root level and its three child classes: *Pathway*, *Interaction*, *PhysicalEntity*. Let analyze more in detail root class and its child classes.

- Entity class is used to represent a single biological element, i.e., pathway, and comprises the following attributes: *Comment* attribute is used to better describe the data encompassed in the class. The *Synonyms* attribute is used to handle multiple name for this element. To track the source of data provenance, the entity class presents the *Data-Source* attribute containing a text description of the source of provenance of this data, i.e., a database. Multiple external references to the entity are defined by using the *xref* attribute. The *name* attribute contains the full name of this element. *Short-Name* attribute contains the abbreviation of the full name, useful in a visualization context to display the name as label of a graphical element that represents this entity. All attributes that are defined in this class and are not inherited.
- Pathway class is a network of interactions, comprising the following attributes. The *pathwayComponents* attribute contains the set of interactions in this pathway or network. The *organism* attribute encompass information about the organism, i.e., *Felis-catus*. *Data-Source* attribute contains a text description of the source of provenance of this data, i.e., a database. Multiple external references to the entity are defined by using the *xref* attribute. *name* attribute contains the full name of this element. *Short-Name* attribute contains the abbreviation of the full name, useful in a visualization application to label a graphical element that represents this entity. Scientific evidence are handled by the *evidence* attribute. The *pathway Components*, *evidence* and *organism* attributes are defined in this class, whereas the other are inherited.
- Interaction class describes the interactions between two or more entities. Since an interaction can involve more elements, the involved elements are handled by the *participant* attribute. *Data-Source* attribute contains a text description of the source of provenance of this data, i.e., a database. Multiple external references to the entity are defined by using the *xref* attribute. *name* attribute contains the full name of this element. *ShortName* attribute contains the abbreviation of the full name, useful in a visualization context to display graphically the name of the element that represents this entity. The *participants* and *evidence* attributes are defined in this class, whereas the other are inherited.

- PhysicalEntity class represents a set of entities, each one with its own physical structure. *cellularLocation* attribute specify a cellular location, i.e., cytoplasm obtained by referring to Gene Ontology. *feature* specifies the only relevant features for the physical entity, i.e., binding site. *memberPhysicalEntity* attribute should be used to create generic groups of physical entities, however it is recommended do not use this attribute. *notFeature* attribute used to describe lacking features of the physical entity. Other attributes are availability, comment, dataSource, evidence, name, xref. The attributes defined in this class, are all inherited.

To get more information on all subclasses available in BioPAX, it is recommended to consult the BioPAX level 3 documentation available at the following web addresses: BioPAX.org (see Relevant Website section).

In brief, the most significant changes in BioPAX Level 3 respect to BioPAX Level 2 and Level 1 regard the following aspects. The PathwayStep class has been added as an attribute into the pathway class. Also, have been added a new class called BiochemicalPathwayStep, a child class of PathwayStep, making possible to order the biochemical processes. The physicalInteraction, which stores molecular interactions, has been moved to be a child of the Molecular Interaction class. Also, the openControlledVocabulary class has been renamed ControlledVocabulary, making possible to define a class for each controlled vocabulary. Finally, the confidence class has been renamed as Score, making it more flexible and suitable to describe genetic interactions. The major improvement introduced by BioPAX level 3 are the following: Better support for physical entities in diverse states: Proteins from sequence database like UniProt, IID and so on, are now expressed as a ProteinReference. ProteinReference stores the protein sequence, name, external references, and potential sequence features (similar in purpose to the class protein in BioPAX Level 1 and 2). The real proteins wrapped in a complex or present in a particular cellular compartment, in BioPAX Level 3 are represented through the class Protein (that is comparable in purpose to the class physicalEntityParticipant in BioPAX Level 1 and 2). Furthermore, stoichiometry attribute in this distribution is part of Conversion class, avoiding to duplicate proteins, as was done with physical Entity Participants in Level 1 and 2). Making it easier to create different types of protein without to duplicate common information to all kinds. Sequence features and stoichiometry have been significantly changed. Whereas DNA, RNA, and small molecule, have been redesigned. Only Complex has not been modified. Conversely, with the new design, the physicalEntityParticipant class has been removed, as it is no longer needed. BioPAX level 3 introduced a support to define generic physical entities, i.e., such as binding sites. That can be represented using the EntityReference class, or also supported using the EntityFeature class and its memberFeature property. BioPAX Level 3 can represent Gene regulation networks, by representing their target. Genetic interaction representation is possible in BioPAX level 3 through the GeneticInteraction class, which contains a set of genes and a phenotype (expressed using PATO or another phenotype controlled vocabulary).

BioPAX Condensing Example

In this section, are discussed how the known biological pathways contained into the organisms are encoded, stored, exchanged, visualized and analyzed by using BioPAX format. Pathway data models in BioPAX are generally encoded by using plain text as conveyed in Code 3.1. Code 3.1 shows how the pathways in the Mus-Musculus organism are coded by means of the BioPAX Level 3 format. Analyzing in detail the pathway data conveyed in Code 3.1 it is worth to note that data are encoded by using Resource Description Framework (RDF) that was developed to provide a meta language data model to the web. Nowadays, RDF is used as a standard data model for the Web of Data and the Semantic Web to support the representation, access, constraints, and relationships of objects of interest in a given application domain. Ontologies and their elements are identified using Internationalized Resource Identifiers (IRIs). URIs represent common global identifiers for resources across the Web and locally into the BioPAX file.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:bp="http://www.biopax.org/release/biopax-level3.owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xml:base="http://www.reactome.org/biopax/56/48892#">
  <owl:Ontology rdf:about="">
    <owl:imports
      rdf:resource="http://www.biopax.org/release/biopax-level3.owl" />
  </owl:Ontology>
```

```

<bp:Pathway rdf:ID="Pathway1">
  <bp:pathwayComponent rdf:resource="#Pathway2" />
  <bp:pathwayOrder rdf:resource="#PathwayStep3" />
  <bp:organism rdf:resource="#BioSource1" />
  <bp:displayName
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">3'
----UTR-mediated translational regulation
----</bp:displayName>
----<bp:xref rdf:resource="#UnificationXref479" />
----<bp:xref rdf:resource="#UnificationXref480" />
----<bp:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
----This event has been computationally inferred from....
----</bp:comment>
----<bp:evidence rdf:resource="#Evidence4" />
----<bp:dataSource rdf:resource="#Provenance1" />
--</bp:Pathway>
--<bp:Pathway rdf:ID="Pathway2">
----<bp:pathwayComponent rdf:resource="#BiochemicalReaction1" />
----<bp:pathwayComponent rdf:resource="#BiochemicalReaction2" />
----<bp:pathwayOrder rdf:resource="#PathwayStep2" />
----<bp:pathwayOrder rdf:resource="#PathwayStep1" />
----<bp:organism rdf:resource="#BioSource1" />
----<bp:displayName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
----L13a-mediated translational silencing of Ceruloplasmin expression
----</bp:displayName>
----<bp:xref rdf:resource="#UnificationXref477" />
----<bp:xref rdf:resource="#UnificationXref478" />
----<bp:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
----This event has been computationally inferred from... </bp:comment>
----<bp:evidence rdf:resource="#Evidence3" />
----<bp:dataSource rdf:resource="#Provenance1" />
--</bp:Pathway>
--<bp:BiochemicalReaction rdf:ID="BiochemicalReaction1">
----<bp:conversionDirection
----rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
----LEFT-TO-RIGHT

```

```

</bp:conversionDirection>
<bp:left_rdf:resource="#Complex1" />
<bp:right_rdf:resource="#Complex2" />
<bp:right_rdf:resource="#Protein57" />
<bp:displayName_rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
Dissociation of L13a from the 60s ribosomal subunit
</bp:displayName>
<bp:xref_rdf:resource="#UnificationXref246" />
<bp:xref_rdf:resource="#UnificationXref247" />
<bp:comment_rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
This event has been computationally
</bp:comment>
<bp:evidence_rdf:resource="#Evidence1" />
<bp:dataSource_rdf:resource="#Provenance1" />
</bp:BiochemicalReaction>
.....

```

Code 3.1. In Figure is partially displayed (for space reason) the known biological pathways belonging to the Mus-Msculus organism. The data are retrieved from Reactome database and are encoded by using BioPAX Level 3 format.

The main advantages to represent biological pathways data by means of BioPAX is that, data can be analyzed by means of computational approaches. As well as can be visualized from all the application compatible with the BioPAX file format. In [Fig. 1](#)

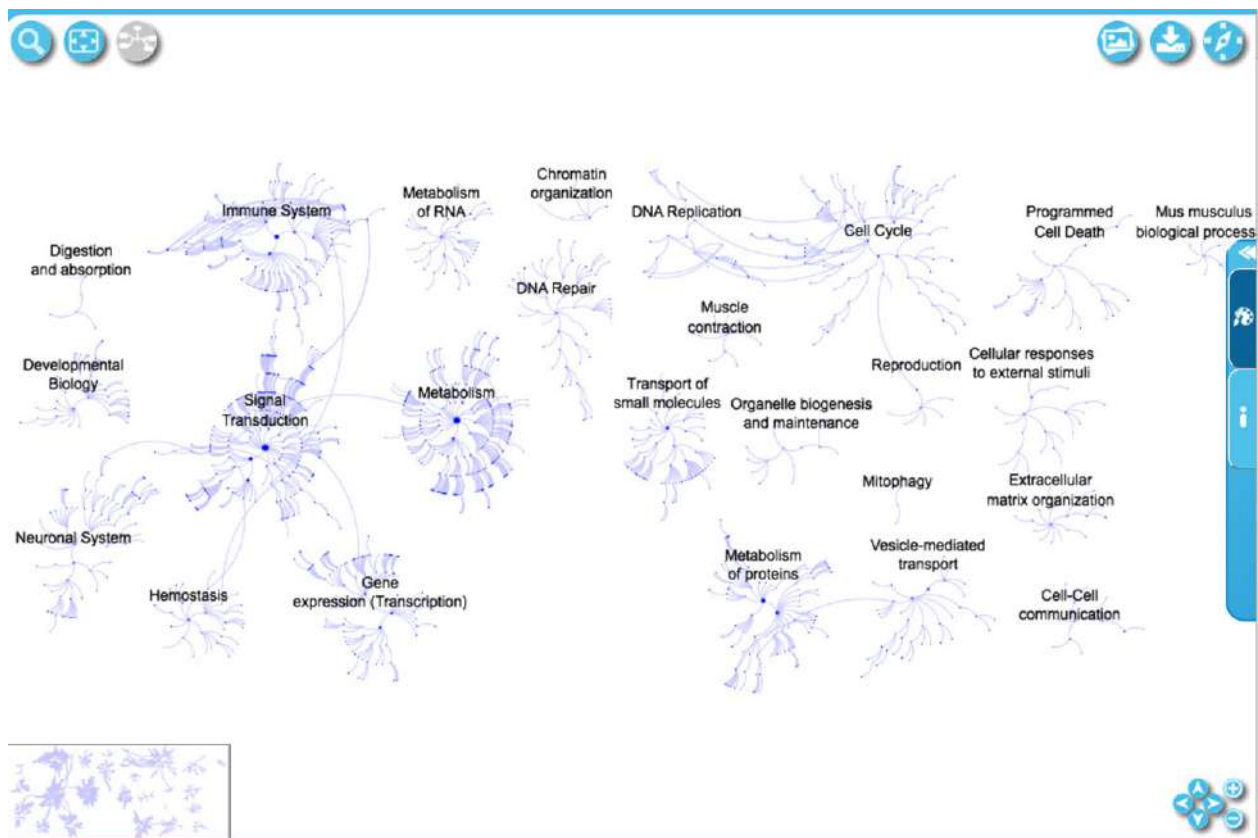


Fig. 1 Reactome Pathway Viewer.

is depicted the network of the biological pathways present in the Mus-Musculus organism, by using the web platform called Pathway browser available in the Reactome web site (see Relevant Website section).

In the top left viewer interface of Reactome Pathway Browser, are located buttons to provide basic navigation, zooming, and screen arrangement actions. On the top right are located some buttons to get in an easy way the illustration, export options, and pathway overview. In the left corner of the canvas, there is the navigation panel of the pathway browser (to have in one-shot a full pathway view), instead of on the right corner; there are the buttons to navigate the pathway and zoom in-out. The central canvas displays the pathway, making possible to navigate it by clicking and dragging or zooming. To get information regarding the shown elements, simple move the mouse over each node or edge and automatically the browser will show short information regarding the selected item. Clicking on the selected item will be highlighted in the event hierarchy situated on the left site of the canvas. From the event hierarchy, it is possible to go more into detail regarding the selected element. When a pathway is chosen in the canvas, in the tab "Download" will be shown some buttons to download the pathway in several formats, for reuse or reference. Formats include Word, PDF, SBML, SBN, BioPAX 2, BioPAX 3. Moreover, from the tab situated under the canvas, it is possible to get details of the select item in the Pathway Browser. For example, when a reaction is selected, will be shown details including molecules, summary and references, evidence and so on. The tab "Molecules" provides details regarding all the molecules involved with the selected one. If an item is selected in the diagram, the corresponding molecules are highlighted. The tab "Details" shows details for the selected item in the pathway diagram. For reactions, if available reaction diagrams from the Rhea database are retrieved and visualized. For simple molecules, information from ChEBI are extracted and visualized. For pathway items that contain proteins shows, 3D structure from PDB if available. Finally, the tab "Expression" displays gene expression information for genes corresponding to the selected item into the canvas. Expressions data are obtained from the Gene Expression Atlas. Over the canvas are located some tabs "Analysis," "Tour," and "Layout." The "Analyze" tab allows experimental data to be loaded, or paste text data, i.e., a sequence in the text-area. Following the step user will be guided to the results, that will be shown in a tabular format into the Analysis tab situated under the canvas. The "Layout" allows to users to chose what elements of the Reactome pathway browser to visualize or not. The "Tour" tab gives to the user a video guide explaining how to use the Pathway Browser.

Pathway Databases Using BioPAX

At today, BioPAX is used in several public pathways databases, as format to encode pathways data. Below is a list of the major biological pathways database that use BioPAX for pathway data encoding. The BioCyc ([Caspi et al., 2008](#)) databases contains metabolic and signaling pathways data encoded by using BioPAX Level 3. Data stored in BioCyc are computationally predicted. BioCyc provides tools for navigating, visualizing, and analyzing the underlying databases, and for analyzing omics data, to get access to data and tools provided by BioCyc subscription is required.

The BioModels ([Le Novère et al., 2006](#)) database contains models of biological processes. BioModels contains 630 manually curated models and 983 non curated models, providing access to 112,898 Metabolic models, 27,531 Nonmetabolic models and 2641 Whole genome metabolism models. The access to the model archives is free to everyone. Data in BioModel are encoded by using SBML and BioPAX Level 2.

EcoCyc ([Keseler et al., 2005](#)) is a scientific database for the bacterium *Escherichia coli* K-12 MG1655. EcoCyc provides access to literature-based curation of the entire genome, and of transcriptional regulation, transporters, and metabolic pathways. Metabolic and Signaling Pathways are encoded by using BioPAX, Level 3 and the access is free for everyone.

Kyoto Encyclopedia of Genes and Genomes KEGG ([Kanehisa and Goto, 2000](#)) is a Pathway database for understanding and analyzing gene functions, and linking genomic information providing utilities to understand how the biological systems work. Data in KEGG are encoded by using the BioPAX Level 1. The KEGG system is available free for academics purposes.

MetaCyc ([Caspi et al., 2008](#)) is a curated database of experimentally metabolic and signaling pathways. MetaCyc contains 2526 pathways from 2844 different organisms. Pathways data in MetaCyc are encoded by using the BioPAX Level 3. The MetaCyc is freely available for each user.

NetPath ([Kandasamy et al., 2010](#)) is a manually curated resource of signal transduction pathways in humans. Data in NetPath are available for download in BioPAX level 3.0, PSI-MI version 2.5 and SBML version 2.1 formats. The NetPath system is freely available for use.

Pathway Commons (see Relevant Website section) ([Cerami et al., 2011](#)) is a web resource for biological pathways data analysis, store and exchange. The access to Pathway Commons is free, data access are freely available under the licence terms of each involved database. Data are stored and retrieved by using BioPAX.

Reactome (see Relevant Website section) ([Croft et al., 2014](#)) is an open-source, curated and peer reviewed Metabolic and Signaling pathway database. The goal of Reactome is to provide intuitive bioinformatics tools for the visualization, exchanging, interpretation and analysis of pathway. Data are encoded by using BioPAX file format.

Rhea (see Relevant Website section) ([Alcántara et al., 2012](#)) is a freely available resource of curated biochemical reactions. It has been designed to provide a set of chemical transformations for applications such as the functional annotation of enzymes, pathway inference and metabolic network reconstruction. Data in Rhea are available in several different file format for the download in particular pathways data are encoded by using BioPAX Level 2. All data in Rhea is freely accessible and available for anyone to use.

See also: Bioinformatics Data Models, Representation and Storage. Data Formats for Systems Biology and Quantitative Modeling. Data Storage and Representation. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Quantitative Modelling Approaches. Text Mining for Bioinformatics Using Biomedical Literature

References

- Alcántara, R., Axelsen, K.B., Morgat, A., *et al.*, 2012. Rhea a manually curated resource of biochemical reactions. *Nucleic Acids Research* 40 (D1), D754–D760.
- Bechhofer, S., 2009. Owl: Web ontology language. *Encyclopedia of Database Systems*. Springer. pp. 2008–2009.
- Caspi, R., Foerster, H., Fulcher, C.A., *et al.*, 2008. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research* 36 (suppl 1), D623–D631.
- Cerami, E.G., Gross, B.E., Demir, E., *et al.*, 2011. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Research* 39 (Suppl 1), D685–D690.
- Croft, D., Mundo, A.F., Haw, R., *et al.*, 2014. The reactome pathway knowledgebase. *Nucleic Acids Research* 42 (D1), D472–D477.
- Dean, M., Schreiber, G., Bechhofer, S., *et al.*, 2004. Owl web ontology language reference. W3C Recommendation February 10.
- Demir, E., Cary, M.P., Paley, S., *et al.*, 2010. The biopax community standard for pathway data sharing. *Nature Biotechnology* 28 (9), 935–942.
- Kandasamy, K., Mohan, S.S., Raju, R., *et al.*, 2010. Netpath: A public resource of curated signal transduction pathways. *Genome Biology* 11 (1), R3.
- Kanehisa, M., Goto, S., 2000. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Keseler, I.M., Collado-Vides, J., Gama-Castro, S., *et al.*, 2005. Ecocyc: A comprehensive database resource for *escherichia coli*. *Nucleic Acids Research* 33 (Suppl 1), D334–D337.
- Le Novère, N., Bornstein, B., Broicher, A., *et al.*, 2006. Biomodels Database: A free, centralized Database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Research* 34 (Suppl 1), D689–D691.

Relevant Websites

- <http://biocyc.org>
BioCyc.
- <http://www.biopax.org>
Biological Pathway Exchange.
- <http://biomodels.net/>
BioModels.
- <http://www.biopax.org/release/biopax-level3-documentation.pdf>
BioPAX.
- <http://www.biopax.org/release/biopax-level2-documentation.pdf>
BioPAX.
- <http://www.biopax.org/release/biopax-level1-documentation.pdf>
BioPAX.
- <http://ecocyc.org/>
EcoCyc.
- <http://www.kegg.jp>
Kanehisa Laboratories.
- <http://metacyc.org/>
MetaCyc.
- <http://netpath.org/>
NetPath.
- <http://www.pathwaycommons.org>
Pathway Commons.
- <http://www.reactome.org/PathwayBrowser/>
Reactome.
- <http://reactome.org/>
Reactome.
- <http://www.ebi.ac.uk/rhea>
Rhea.

Models for Computable Phenotyping

Alfredo Tirado-Ramos and Laura Manuel, University of Texas Health at San Antonio, San Antonio, TX, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

The need for the creation of health research networks that comprise healthcare and pharmaceutical organizations, and are able to collaborate in order to facilitate clinical trial design and patient accrual, is currently driving a strong push in academic and industrial research. This push is being translated into federated clinical big data grids and their associated querying mechanisms. One of the fastest related growing fields of interest for researchers in this field is the design and development of technologies and processes for seamlessly querying logically and geographically dispersed patient data and metadata resources for cohort creation in such networks, while complying with diverse local and global governance requirements for patient data privacy. The currently de facto standard approach is called *computable phenotyping*, and is based on a varied ecosystem of technologies and tools for creating virtual organizations that share patient metadata in real time, while also allowing for participation in multi-site collaborations with local, state, national or international healthcare organizations. Computable phenotypes allow for well-defined processes for deepening phenotype querying, analysis and data de-identification, while pipelining the development of study protocols.

Background

A computable phenotype can be best described as a set of inclusion and exclusion criteria for a patient cohort. Criteria should be specific and objective enough to turn them into a machine-readable query, yet also generalized enough to make them portable between different data sources. Mere verbal descriptions are neither a computable phenotype, nor are they a set of proprietary vendor codes for a specific electronic health record. Nevertheless, a list of standardized medical terminology codes (ICD, HCPCS, LOINC, NDC, etc.) could be a computable phenotype.

Computable phenotypes are necessary for characterization of cohorts and reproducibility of clinical research. Yet currently, *Electronic Health Record* (EHR) systems do not have the ability to create and distribute computable phenotypes that can be utilized across multiple sites for research reproducibility. Standardized solutions are needed and are being defined. But even if a EHR vendor develops a system similar computable phenotype tool to existing open source program, they do not have a compelling reason to make such tools usable across platforms.

One of the utmost challenges, then, is to produce and make available such tools and algorithms for querying patient data in an open and scalable way, for applications ranging from clinical trial patient accrual to queries used for preparatory research work.

Several tools have been generated for creating and consuming computable phenotypes, each with their own strengths and weaknesses. These include OMOP, PCORNet Front Door, i2b2, and SHRINE, among others ([i2b2, 2017](#); [OMOP, 2017](#); [PCORnet, 2017](#); [SHRINE, 2017](#)).

Applications

There are a few research groups, mostly based in the United States, that are working on developing standardized solutions and tools for building scalable computable phenotype application frameworks. One such application framework is the set of tools developed by the *Patient-centered Research Institute* (PCORI), a research initiative started by the Obama administration focused on novel technologies and methods that stress patient-centered outcomes. Through the use of their *Common Data Model* (CDM) PCORI has created both a way of distributing queries via their main query tool, as well as a standardized format for research data warehousing, which provides a consistent framework for writing queries ([PCORnet, 2018](#)). In a quarterly manner, PCORI distributes queries throughout its national distributed network, that review data warehouses for consistency and compliance to its rules. By utilizing this standardized data warehouse PCORI can write a single query language script (e.g., SAS) or query and expect it to run without error on every instance of a data warehouse. PCORI leverages a number of relevant standards (ICD, HCPCS, RXCUI, NDC, etc.) as well as their own standardized data set to create reproducible computable phenotypes across its collection of data warehousing clusters, or *Clinical Data Research Networks* (CRDNs).

While PCORI's idea of a computable phenotype implementation is one of the most widely deployed across the United States, it is not the only one. Tool-specific approaches such as [i2b2 \(2017\)](#) allow for the straightforward creation of reproducible computable phenotypes, ranging from academic institutions' bare bones instantiations as well as at pharmaceutical industry-driven sophisticated frameworks. For instance, at the University of Texas Health San Antonio we have invested considerable resources into generating and processing computable phenotypes, analyzing the most efficient and concise manners through which they can be created and distributed. As specific context, we work on computable phenotypes for research grants, e.g., for our *Clinical and Translational Science Awards* (CTSA) Collaborative Innovation Award for surgical outcome disparities, as well as with a

number of externally-initiated clinical trials within our participation with the international TriNetX network and diverse national research groups.

In general, research groups like ours that are working on computable phenotyping face a number of serious limiting factors, such as hiring and training of skilled programmers with the ability to work with medical data, training medical researchers to understand the limitations of *Electronic Health Record* data, and coordinating with the local teams to correctly construct computable phenotypes out of raw data. That is, it is often necessary to train a cadre of scientists (both MD and PhD) to understand enough about the limitations and capabilities of medical data to be able to make effective use of computable phenotypes in their research, in an increasingly independent manner. Such scientists do their part by learning how to construct eligibility criteria and design studies that leverage their skills, expertise and needs in order to understand where specific technical support is needed.

Analysis

One of the main challenges in applying a computable phenotypes is the alignment of computable terms. It seems surprising that something as simple as a medication, e.g., “Tylenol[®]” would be complicated to process in an Electronic Health Record, but Tylenol[®] (acetaminophen) is an ingredient that can be found in more than 100 brand names and 200 drug combinations. Thus it becomes important for researchers to decide if a particular brand of Tylenol[®] or every medication that contains the ingredient acetaminophen, should be used in a particular computable phenotype. A mapping tool (e.g., RXNav) which can pull up all of the medications and their associated codes, may be necessary to find all brand names and combinations. Laboratory tests, diagnosis, and procedures all have similar issues, some of which can be solved through the use of international coding standards (e.g., ICD9, ICD10) or proprietary coding schemes (e.g., CPT), but even the use of these standardized coding schemes may present issues. That is, if a researcher provides a phenotype with just ICD9 codes, it might happen that these codes do not map exactly to ICD10 codes, and thus cause the potential for mapping errors. Medicare uses medical billing codes for procedures (e.g., CPT), however these codes are proprietary, so finding a list of applicable procedure codes can be difficult both for the researcher and for the technician trying to translate the query to their local site. Terms may be sent as free text such as: “Bariatric surgery” and leave the query writer to look up everything they can translate as bariatric surgery. But this also brings in the issue of the query builder needing to make decisions about the exact codes to be used. However, query builders are generally computer programmers with little to no medical knowledge who then would have to make decisions about medical terminology, which should be best left in the hands of medical coders, nurses and physicians.

A large part of our analysis at the University of Texas Health Science Center includes information that is not stored in the Electronic Health Record such as socioeconomic data (education, job, and relations) or difficult to extract from certain types of data. For example, many physicians do not code family or medical history diagnosis in the Electronic Health Record system and instead write them within unstructured, free-text notes. Free-text notes may contain *Protected Health Information* (PHI) and require specialized handling. They may also require complex *Natural Language Processing* programming, or manual curation to convert them into a structured format usable by researchers and data analysts.

Additionally, the limitations of Electronic Health Records should be taken into account when attempting to construct a “control” group. Electronic Health Records contain the necessary data for clinics or hospital departments in order to treat their current patient properly, but they may not contain crucial data for indepth research. Care must be taken to ensure that control groups are matched, not just by their lack of a diagnosis, but in the treatment areas they visited. An otolaryngologist would have no need to code diagnosis about infertility or other diagnosis that do not apply to their practice, and thus patients who have only visited a specialist outside of the scope of the targeted diagnosis should be excluded from control cohort selection. Thus the absence of a diagnosis in the Electronic Health Record system cannot be considered the absence of a condition.

The study of data quality metrics is a very important area within data analysis, which is in need of vigorous research and development. The Electronic Health Record system may or may not apply constraints on elements like plausible dates, foreign key constraints, or maintaining consistency between the various identifiers. Corrupted patient data can lead to invalid data such as one patient’s laboratory data being associated with another patient’s visit or account data. Variation in the structure of the Electronic Health Record system, the documentation habits of physicians and other healthcare workers who enter or structure data within the system, and the usage of coding schemes across sites provide increased complexity to the task of composing a computable phenotype for use across multiple sites.

Another important challenge is the conception of a coherent and technically informed vision with the input from all stakeholders about how informatics in general and computable phenotypes in particular could serve combined goals of research, quality improvement, and business intelligence. It requires the teamwork of experts from various disciplines to create a data warehouse that can provide researchers with access to high quality data; it should be noted that research data warehouses may provide deceptively easy access to complex data in a rapidly evolving discipline.

Case Study

A particularly descriptive case study on the development and use of computable phenotypes in the real world is provided by the ADAPTABLE trial by PCORI ([ADAPTABLE Home, 2017](#)). The ADAPTABLE trial is a first of its kind, looking at the benefit/risk ratio for different dosages of aspirin, weighing the risk of bleeding vs the benefit of reduced cardiac events, and it involves most of the nationwide Clinical Data Research Networks considerable resources. The initial ADAPTABLE phenotype was distributed with a text description


```

union
  Select distinct patid from pcori_cdm.diagnosis where dx_type = '09' and dx in (
    'V4581',
    '41000', '41001', '41002', '41010', '41011', '41012', '41020',
    '41021', '41022', '41030', '41031', '41032', '41040', '41041',
    '41042', '41050', '41051', '41052', '41060', '41061', '41062',
    '41070', '41071', '41072', '41080', '41081', '41082', '41090',
    '41091', '41092', '412',
    '4110', '4111', '41181', '41189', '4130', '4139', '41400', '41401', '41402',
    '41403', '41404', '41405', '41406', '41407', '4142', '4143', '4144', '4148',
    '4149',
    '25000', '25001', '25002', '25003', '25010', '25011', '25012', '25013', '25020',
    '25021', '25022', '25023', '25030', '25031', '25032', '25033', '25040', '25041',
    '25042', '25043', '25050', '25051', '25052', '25053', '25060', '25061', '25062',
    '25063', '25070', '25071', '25072', '25073', '25080', '25081', '25082', '25083',
    '25090', '25091', '25092', '25093',
    '430', '431', '4320', '4321', '4329', '43300', '43301', '43310', '43311', '43320',
    '43321', '43330', '43331', '43380', '43381', '43390', '43391', '43400', '43401',
    '43410', '43411', '43490', '43491', '4350', '4351', '4352', '4353', '4358', '4359',
    '436', '4370', '4371', '4373', '4374', '4375', '4376', '4377', '4378', '4379', '4380',
    '43810', '43811', '43812', '43813', '43814', '43819', '43820', '43821', '43822',
    '43830', '43831', '43832', '43840', '43841', '43842', '43850', '43851', '43852',
    '43853', '4386', '4387', '43881', '43882', '43883', '43884', '43885', '43889', '4389',
    '44020', '44021', '44022', '44023', '44024', '44029', '4439',
    '42820', '42821', '42822', '42823', '42840', '42841', '42842', '42843',
    '39891', '40201', '40211', '40291', '40401', '40403', '40411', '40413', '40491',
    '40493', '4280', '4281', '42820', '42821', '42822', '42823', '42830', '42831',
    '42832', '42833', '42840', '42841', '42842', '42843', '4289'
  )
union
  Select distinct refpop.patID
  From refpop
  join pcori_cdm.lab_result_cm l on refpop.patid = l.patid
  join pcori_cdm.demographic d on d.patid = refpop.patid
  where lab_loinc = '2160-0'
  and result_num between 1.5 and 61.3
  and d.birth_date <= current_date - (65*365.25); -- above 61.3(highest ever recorded) we can assume erroneous value
union
  select distinct refpop.patid
  from refpop
  join pcori_cdm.demographic d
  on d.patid = refpop.patid
  join pcori_cdm.vital v on d.patid = v.patid
  where v.systolic >=140 and d.birth_date <= current_date - (65*365.25) and v.measure_date >= current_date - 365
  union

```

Fig. 2 A section of the refined code created from the final ADAPTABLE phenotype. This phenotype contained lists of codes that removed the guess work out of building the query. Localized corrections for data quality issues may still be required, such as incorrectly entered lab values. This code was designed to work on the PCORI CDM tables. Reproduced from PCORnet Common Data Model (CDM), 2018.

controlled by the hospital's quality and informatics teams which provide an audit trail. Depending on the method of managing these changes, erroneous data may persist in the database even if it is not visible to caretakers.

Electronic Health Record data is by nature, incomplete. Physicians are paid to treat patients, not to maintain accurate problems lists, thus problems that have been resolved are often not removed from the database and simply persist until someone removes or otherwise remedies them. Patients may still have a broken bone in their record months or years after the injury has fully healed. Additionally, physicians have no incentive to add structured diagnosis data to a patient's record to indicate conditions which they are not treating the patient for. Insurance only reimburses for problems that the physician is treating, not for the completeness of the medical record.

A patient may have been diagnosed with conditions that have not been added to their records. Some researchers erroneously consider health data for a patient to be "complete" and assume that if a patient has been diagnosed with a condition that it is in the chart: an Ear Nose and Throat (ENT) specialist may not pay much attention to the fact that a patient may be infertile, since it may not affect his work; similarly, a fertility specialist may not pay much attention to the fact that a patient presents a deviated septum.

The quality of a phenotype definition requires a developer experienced with electronic medical records and cross disciplinary faculty who can bridge the expertise gap between the developers, the clinicians, the statisticians, and other scientists working on a project. Reviewers who can vet the queries are needed for objective, reliable performance metrics, and validation. The state of diagnostics has improved greatly over the last half decade, but much work remains to be done. There need to be subject matter experts who can recognize disease characteristics beyond explicit diagnoses, leveraging quantitative measures and indicators of disease including laboratory results, vital signs, and medication prescribing patterns insofar as they can be extracted from the Electronic Health Record. Different tactics may be optimal depending on whether the condition of interest is chronic, acute, or transient. These tactics will not necessarily be applicable at different healthcare organizations which have different approaches for the handling of chronic, acute, or transient problems.

Future Directions

The ability to identify novel research approaches for the creation of health research networks that comprise healthcare and pharmaceutical organizations is becoming increasingly dependent of the manner in which we create, develop and deploy computable phenotypes. Such computable phenotypes, once defined, should be applicable to multiple research problems utilizing multiple institution's data. There is indeed a need for a set of standardized processes and tools like those mentioned before, which can be leveraged when developing reliable phenotypes that are not sensitive to the specific enterprise Electronic Health Record vendors or institutions. Academic approaches that line up queries and reviews of the structure and consistency of a common data model, as it takes place in the PCORnet network, are a positive step forward. In these approaches queries are distributed to test common quality issues such as multiple patient's data being coded to the same hospital visit, resulting in improved quality within the data warehouses of participating institutions. Nevertheless, real channels by which to communicate the problems and solutions back to the enterprise Electronic Health Record teams are sorely needed.

Unlike system vendors, academic researchers do not divert their limited resources from solving scientific and engineering obstacles to focus on marketing. As a consequence, there will be a consistent bias toward over estimating the value of proprietary solutions and underestimating the value of open source solutions produced by researchers. Furthermore, vendors have strong economic incentives to not make their methodology transparent, reproducible, portable or interoperable. One way for institutions to protect themselves against vendor lock-in and data-blocking is to embrace not only open source, open standards compliant software but also for decision makers to educate themselves about the business models and collaborative practices that make open source software possible, so successful, and rapid in its evolution. We at the University of Texas Health Science Center have used our computable phenotyping capabilities for projects that include national collaborative research on network governance, surgical outcome disparities, Amyotrophic Lateral Sclerosis, Weight and Health, Antibiotics and Childhood Obesity, and so forth. Our efforts, as in many other small to medium size research outfits, have been mostly limited by the difficulty of recruiting and retaining faculty and staff programmers with the required skillset, and the very nature of work consisting in constructing computable phenotypes out of raw Electronic Health Record data.

Closing Remarks

Our own successful experience with this technology since 2014 has helped our Electronic Health Record system become more usable and relevant to our researchers, while helping our collaborative efforts with our external partners as well as our local clinical trials office. A pragmatic leveraging of such new resources has been of great importance for our institution to advance the design of studies and clinical trials with 21st century technology, and bring our clinicians and other stakeholders onboard. If the impact of data completeness issues and the preventable within-institutional siloing of business versus research informatics can be eventually overcome, there would be tremendous benefits seen for published studies, clinical registries, coordination of patient care, and fiscal sustainability of institutions by minimizing waste, readmission, complications and errors.

It is our belief that the ability to identify cohorts of people with particular health conditions, across healthcare organizations, by using common definitions has proven to have an intrinsic value for clinical quality measurement, health improvement, and research.

Acknowledgments

The authors would like to acknowledge the i2b2 transSMART foundation, PCORnet, and Harvard Catalyst for their contributions to open source computable phenotype platforms and the National Library of Medicine for its strides in producing and compiling standardized ontologies for medical science.

See also: Bioinformatics Data Models, Representation and Storage. Data Storage and Representation. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Text Mining for Bioinformatics Using Biomedical Literature

References

- ADAPTABLE Home. 2017. Available at: <http://theaspirinstudy.org/> (accessed 11.10.17).
 i2b2. 2017. Informatics for Integrating Biology & the Bedside. Available at: <https://www.i2b2.org/> (accessed 11.10.17).
 OMOP. Observational Medical Outcomes Partnership. 2017. Available at: <http://omop.org/> (accessed 11.10.17).
 PCORnet Common Data Model (CDM). PCORnet. CDM v4.0: Released January 3, 2018.
 PCORnet. 2017. The National Patient-centered Clinical Research Network. PCORnet. Available at: <http://www.pcornet.org/> (accessed 11.10.17).
 SHRINE. 2017. Harvard Catalyst. Available at: <https://catalyst.harvard.edu/services/shrine/> (accessed 11.10.17).

Biographical Sketch



As the chief and founder of the Clinical Informatics Research Division of the University of Texas Health Science Center at San Antonio, Dr. Tirado-Ramos leads a full-spectrum biomedical informatics program and explores the intersection between informatics, translational science, and clinically relevant data-centric problems including, but not limited to, computable phenotype-based research in health disparities, obesity, amyotrophic lateral sclerosis, ageing, and cancer. Under the umbrella of successful PCORI awards, he created and maintains an information research system for interdisciplinary collaboration between pediatric endocrinologists, cancer researchers and neurologists, creating new institutional governance frameworks along the way. He also co-directs the informatics core at the Claude Pepper Older Americans Independence Center, a National Institute on Aging award, where he works on state of the art informatics infrastructures to investigate innovative interventions that target the aging process as well as aging-related diseases, with a major focus on pharmacologic interventions. Previous to arriving at the University of Texas, he served at Emory University School of Medicine as Associate Director for the Biomedical Informatics Core at the Center for AIDS Research at the Rollins School of Public Health.



Laura Manuel received her BS in Computer Science from the School of Science at the University of Texas San Antonio. She works as the lead developer in the Clinical Informatics Research Division at the University of Texas Health Science Center San Antonio and oversees the work of the development team. She oversees the processing and deidentification of clinical data for the CIRD clinical data warehouse, has done research with geospatial analysis and is currently working on agent base simulation for HIV transmission.

Computing for Bioinformatics

Mario Cannataro and Giuseppe Agapito, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The strong data-driven and integrative nature of research in genomic, interactomics, and proteomics in the post genomic era, has been spurred by the continuing developments of high-throughput experimental assays and their capabilities in data-generation (Loman *et al.*, 2012). Advances in high-throughput innovative technologies continue to support the exponential growth in public and private available genomic data, interactomics data, and proteomics data. Thus, the integration and interpretation of these immense volumes of data could increase of some order of magnitude the enhancements in health and clinical outcomes, representing a challenge to move genomic, proteomics, interactomics research in the clinical activities (Kanehisa *et al.*, 2011). Even the extraction of information from the tissue through images is becoming an high-throughput methodology. In fact, considering the actual high resolution of microscopy, Radio-Magnetic Indicator (RMI), and Positron Emission Tomography (PET), bioimages present an high resolution that might allow the retrieval of phenotypic information that could be measured precisely and be linked to underlying molecular profiles and clinical outcome. Thus, high-throughput image analysis could serve as a suitable vector to assist clinical diagnosis and tissue-based research (Veta *et al.*, 2014; Caicedo *et al.*, 2009; Gurcan *et al.*, 2009). To promote such a large-scale image and integration analysis it is mandatory to develop scalable databases able to store a massive amount of data based on the NoSQL model, allowing to manage and query image analysis results systematically and efficiently. Thus, to process and analyze these various kinds of data it is necessary the support of computer scientists as well as of statisticians, physics and so on. Data obtained by the experimental high-throughput assays contains bias and noise. Indeed, it is mandatory to clean and uniform the input data, to be used in the knowledge extraction phase. Data clean is known as pre-processing, it is a step of the data mining process. Pre-processing is an iterative methodology that has to be performed many times demanding for a lot of computational power. A significant advantage of the preprocessing phase is that it can be easily parallelized which drastically reduces computation times thanks to the huge computational power that is available on various parallel computers. How highlighted by the previous examples, different kinds of data require different types of analysis and thus they are likely to have different computing requirements. The analysis of next-generation sequencing (NGS), a de novo assembly analysis step, might require vast quantities of RAM memory compared to a BLAST search, which needs less memory but is a CPU-bound process that, for the time to complete a BLAST job, is much more limiting than the speed of the CPU. Data analysis is thus tailored through a combination of resources availability, capacity, and configuration. On the other hand, to analyze high-resolution images of whole tissue slides, researchers need to outline spatial regions into the images representing these regions, as a set of values obtained from the pixels, which can be translated in the form of geometric shapes, surfaces, and fields. Modeling this spatial information provides the support for robust and scalable spatial queries. To support the massive data and computational demands of image analysis algorithms, a solution could be to employ high performance computing techniques to speed-up the image analysis (Kikinis *et al.*, 1998). The application can be built to exploit the modern hybrid computing systems equipped with multi-core CPUs and graphics processing units (GPUs). For example, the connection with the database from which run the query can be handled by the CPU, whereas the features extraction from the retrieved image can be done on the GPU. Because, GPUs provide high-speed memories and extensive multi-processing capabilities, which typically exceed those of CPUs, GPUs are perfect for performing massive parallel imaging analysis. Some of those challenges may be overcome by ad-hoc computational techniques. But still computational power and the efficiency remains the principal bottleneck that limits the execution of such analyses. Although the cost of hardware is decreasing considerably in recent years, investments of thousands of dollars are usually necessary to build and maintain a scientific computing infrastructure. For individual researchers and small labs, who not could have access to large funding, acquiring, configuring, and keeping working the required computing infrastructure is a hindrance and even a barrier to advance research. In addition to the hardware costs and maintenance, advanced software to facilitate parallel computation is typically needed, and a team must be hired to develop and maintain the software on the computing infrastructure. A possible alternative for buying and maintaining one's own computer cluster could be to use computational resources 'in the cloud', e.g., Amazon Web Services. In the recent years, cloud computing has risen as a viable option to quickly and easily obtain computational resources needed for analysis (Bajo *et al.*, 2010; Calabrese and Cannataro, 2016). Cloud computing offers network access to computational resources where CPUs, memory, and disks are accessible in the form of a virtual machine (i.e., a complete operating system) that a user has individual and full control on it. As a result, cloud computing has the potential to allow simple access to a variety of different types of machines, including large-memory machines, fast-CPU machines, or abundant disk space, without needing to build and later maintain the given infrastructure. There exist different forms for providing cloud resources such as the Infrastructure as a Service (IaaS) model. The virtualization technology allows entire operating systems to run independently of the underlying hardware. Due to the low level resources that such a model exposes, and the flexibility with which users can arrange the available components, a wide variety of configurations can be obtained. This feature primarily removes limitations imposed by physical resource availability and helps to enable open-ended analyses. In this way, to the user is given access to what seems to be a typical server computer. However, the server is just a 'virtual machine' running at any one point on the underlying hardware architecture, which is made up of many independent CPUs and storage devices. In the Software as a Service (SaaS) model, users can use the

applications provided by the cloud provider infrastructure, without exorbitant capital costs or infrastructure preparation efforts. SaaS provides to the users remote access to the resources, usually through a web browser. Users need not worry about storage or application management as only specific parameters are enabled for the users' control. Just in some cases, it is possible for the users to manage particular configurations of the application. Platform as a Service (PaaS) is a service model whereby users control the applications deployed, but not the underlying infrastructure. Applications need to be created using programming languages, libraries, services and tools supported by the provider that constitute the development platform provided as a service. An example is Google Apps Engine, which allows developing applications in Java and Python and supplies for both languages the software development kit (SDK). In this way, Cloud computing potentially provides an efficient and low-cost means to achieve the power and scale of computation required to facilitate large-scale efforts in data integration and analysis.

Recent studies showed that cloud computing and HPC are playing a strategic role in the improvement of healthcare services and biomedical research. In fact, they are making it possible to carry out and accelerate radical biological and medical breakthroughs that would directly translate into real benefits for the community.

The remaining of the manuscript is arranged as follows: Section "Parallel Architectures" describes the main parallel architectures used in current computers and suitable to run parallel programs. Section "Distributed Architectures" describes the most used distributed architectures with special focus on Bioinformatics. Section "Programming Languages for Parallel Architectures" summarizes the most used programming languages to write parallel code and illustrates some simple examples introducing how to write simply parallel code, through the presented languages. Section "Programming Languages for Distributed Architectures" introduces the main languages for writing distributed applications to be run on distributed architectures, such as Internet and the Cloud, and provides some simple examples on how to write distributed code. Section "Parallel and Distributed Bioinformatics" presents some uses of Cloud computing and high performance computing in computational biology, bioinformatics, and life sciences. Finally, Section "Closing Remarks" reports some closing remarks.

Parallel Architectures

The amounts of data obtained by new experimental technologies are growing exponentially in size (Marx, 2013; Aronova *et al.*, 2010). Besides, every minute scientific knowledge increases by thousands of pages and to read the new scientific material produced only in a single day, a researcher would take several years. To follow the produced scientific outcomes regarding a single disease, i. e., breast cancer, a researcher would have to examine more than a hundred different journals and data repositories and he/she has to read a lot of manuscripts per day. For example, genome sequencing gives precise information on the basic constituents of life; this massive quantity of data calls for a shift from a reductionist approach to a whole systematic view of biological systems. The whole systematic view of biological systems allows to produce an accurate description of the components and the interactions among them, leading to a better understanding of living systems, only if supported by efficient and scalable algorithms. Thus, the needs of computationally intensive applications able to deal with huge amount of data arise (O'Driscoll *et al.*, 2013). Parallel, distributed and cloud computing architectures are the perfect instruments to produce efficient and scalable tools to help scientist to spur light on biological systems.

Flynn Taxonomy

Flynn taxonomy (Flynn, 2011) identifies 4 classes of computers, taking into account only the instruction and data streams without considering the parallel machine architecture. To understand how the Flynn's taxonomy can classify parallel architectures through the data and instruction streams, it is necessary to understand how the instruction cycle of the CPU works. A generic statement in a program is composed of two parts *operands* and *opcode*. An opcode is used as the value of the statement, whereas the operands represent the address of memory where the data are stored. The CPU's instruction execution cycle comprises the following steps: (i) in the first step the address of the statement to be executed is calculated. (ii) In the second step the statement is fetched (a single statement at time is fetched), then (iii) the current statement is decoded by the Decoder. (iv) The operands address is calculate, thus (v) the fetch of the operands is possible, (vi) the statement is executed, (vii) the result is stored. (viii) If there are more statements jump to (i) otherwise stop. Thus, it can be asserted that the instruction streams includes the instructions executed by the CPU, whereas the data streams contain the data (operands) required for the execution of the instructions. Thus Flynn's classification is based on the multiplicity of the instruction streams and data streams performed by the CPU during the program execution.

- SISD – Single Instruction Single Data, in this category machines are conventional sequential computers that process only one stream of instructions and one stream of data at time by a single CPU, e.g., classical Von Neumann architectures.
- SIMD – Single Instruction Multiple Data, in this structure, multiple processing units (Arithmetic Logic Units – ALU) work under the control of a single Control Unit (CU). All the ALU of this organization receives the same instruction broadcast from the CU. Each ALU takes the data from its own memory, and hence it has on distinct data streams. An example of a dedicated SIMD processor is the GPU that relieves the CPU from time-consuming calculations related to the three-dimensional visualization.
- MISD – Multiple Instructions Single Data, In this class the work of multiple ALU is coordinate by multiple CUs. Each CU is supervising one instruction stream and elaborating it through its corresponding ALU, processing one data stream at the time.

All the CUs communicate and coordinate through the common shared memory for the arrangement of the single data stream. At this class belongs the vector processors, but most often said to be an empty class.

- MIMD – Multiple Instructions Multiple Data, autonomous processors execute simultaneously different instructions on different data. MIMD category comprises the general parallel machines e.g., computer clusters.

Multicore Computers

In a multicore computers, a single CPU is made up of several computational cores. Each core has its own instruction and data memories (i.e., L1 caches) and all cores share a second level on-chip cache (i.e., L2). The CPU is also connected through the bus to the main memory and all the system's peripherals. Furthermore, the L2 cache could be shared between sub-sets of cores, e.g., in a quad cores CPU, there are two L2 caches that are shared by two group of two cores, or is shared by 2 units of 4 cores in an octa-core CPU. Indeed, the external memories are often grouped into multiple levels and use different storage technologies.

Multiprocessor Computers

Multiprocessor Computers can be partitioned in three main architectures SMP, cluster and hybrid.

- A Shared Memory Multiprocessor (SMP) parallel computer consists in a multiprocessor system where, each processor has its CPU and cache, sharing the same central memory and peripherals. A parallel program executed on an SMP parallel computer allows to run multiple threads of one process, on each available processor. The process's program and data are stored in the shared main memory, in this way all the thread in the same process can use the shared memory to communicate. The communication among threads happens reading and writing values into the shared memory data structures.
- A cluster of computers consists of multiple interconnected computational nodes. In general, a cluster presents a dedicate node called frontend, from which users can log in to compile and run their programs, while the backend nodes do the computation. Each backend has its CPU, cache, main memory, and peripherals, such as a local disk drive. Moreover, each backend node is equipped with a dedicated high-speed backend network. The backend network is used only to allow communication between the nodes of the cluster. Other network traffic, i.e., remote logins goes to the frontend. Conversely, from an SMP parallel computer, there is no global shared memory, each backend node can access only its local memory. The cluster computer is known as distributed memory model. In a parallel cluster computer, a parallel program runs in parallel on each backend node. All processes perform its computation independently in its local environment, storing its results in the data structures in its local memory. If one process needs a piece of data that belong to another process's memory space, the process that owns the data sends a message containing the data through the backend network to the process that required the data. Conversely, from the SMP parallel program where the threads can merely access shared data in the shared memory space, in a cluster parallel program, the access to shared data must be explicitly coded enabling nodes to exchange messages among them.
- Hybrid parallel computers cluster and SMP models coexist together. In this architectures are present both shared memory (in each node) and distributed memory (among the nodes). A parallel program in a hybrid architecture runs on separate backend nodes with its central memory space. Moreover, each process has multiple threads, which like in an SMP parallel computer each thread can run on each local CPU. Thus, threads in the same process share the same memory space and can access their own shared data structures directly. Threads belonging to different backend nodes must send messages to each other to communicate in order to share information.

Graphic Processing Unit (GPU)

In recent years, much has been made of the computing industry's widespread shift to parallel computing. Today all consumer computers, smartphones, and tablets will ship with multicore central processors, and graphical process units are making parallel computing not to be only relegated to cluster, supercomputers or mainframes. GPU (Graphics Processing Units) computing (Owens *et al.*, 2008) is relatively new compared to CPU computing, essentially GPUs of early 2000 were developed to handle the color of every pixel on the screen by using a programmable arithmetics units called *pixel shaders*. Because all the arithmetics on input color, text pixel coordinate and so on was controlled by programmer, researcher notice that was possible to handle any data. Thus, have been coined the GPGPU General Purpose term. Because each pixel can be computed independently, the GPU typically presents an architecture with multiple processing cores making possible to calculate multiple pixels in parallel. In response, GPUs became a general-purpose massively parallel coprocessors to perform arbitrary computations on each kind of data. In this way, GPU performs non-rendering tasks by making those tasks appear as if they were a standard rendering (Pharr and Fernando, 2005).

Distributed Architectures

A distributed architecture consists of multiple programs running on various computers as a single system (Sunderam, 1990). The computers belonging to a distributed system can be physically close together and connected by a high-speed network, e.g., cluster, or they can be geographically distant and connected by a vast area network, e.g., a Grid of computers. A distributed architecture can

comprise any possible kind of machines, such as mainframes, workstations, minicomputers, and so on. Distributed architectures aim to hide the underlying computer network by producing a collaborative parallel environment, perceived by the user as a single computer (Rumelhart *et al.*, 1987).

Grid

A computer Grid presents an architecture similar to the clusters, except that the computational nodes are spread over the world and connected through a combination of local area networks and the Internet, instead of that nodes are connected on a single dedicated backend network (Bote-Lorenzo *et al.*, 2004). Grid applications often involve large amounts of computing and/or data. For these reasons, grid offers effective support for the implementation and use of Parallel and distributed computing systems (Cannataro and Talia, 2003a,b, 2004). Grid programming follows the same theoretical approach as cluster computers, with the only difference that programs run on grid machine spread over the internet network. But a parallel program that performs fine on a cluster does not mean that is suitable for the grid. Indeed, on the internet, the latency is orders of magnitude larger than a typical cluster's backend, and the internet's bandwidth is smaller than the cluster backend network. Thus, the messages exchanged among the Grid's nodes are sent over the internet need longer time respect communication in the cluster. Therefore well-suited problems for the Grid are those can be split into many independent pieces and computed independently among the nodes with a meager rate of communication. SETI@home (see "Relevant Websites section") (Search for Extraterrestrial Intelligence) project, and the Great Internet Mersenne Prime Search (GIMPS) (see "Relevant Websites section"), both are examples of a well-suited projects for the Grid environment. SETI@home is a scientific experiment, born at UC Berkeley, that uses Internet-connected computers in the Search for Extraterrestrial Intelligence. Users can participate by downloading and installing a free program that downloads and analyses radio telescope data. GIMPS is a scientific project aimed to discover Mersenne's prime numbers. A Mersenne's prime numbers is defined as: $2^P - 1$ where P is the exponent to be tested. Thus, if $2^P - 1$ is prime, P must also be prime. Thus, the first step in the search of Mersenne's prime number is to create a list of prime exponents to test. Both projects belong to the "voluntary computing" Grid model. In the voluntary computing model, users have to download a client software program and install it on his/her computer. The installed software is low-priority that exploits the idle CPU's cycles to execute parallel computation downloading data from the home site and synchronizing with the other nodes over the networks. In addition to the voluntary computing Grid, usually, a grid is set up by a consortium of companies, universities, research institutions, and government agencies. Examples are, the open source Grid project and Globus Toolkit (see "Relevant Websites section"), and the Open Science Grid (OSG) (see "Relevant Websites section"). Globus allows researchers and people to share computing power, databases. Globus includes software services and libraries for resource monitoring, discovery, and management, plus security and file management. The Open Science Grid (OSG) is a grid devoted to large-scale scientific computation, with thousands of institutional processors located in several countries, to facilitate the access to distributed high throughput computing for research.

Cloud

Cloud computing can be perceived as an evolution of the Grid computing, with the inclusion of virtualization and sharing of resources (Mell *et al.*, 2011). Distributed and Grid computing have long been employed for high-performance computing and scientific simulation and now Cloud computing architecture is becoming an attractive platform for scientific researchers and their computing needs. The World Wide Web (in short, WWW) spurs the development of web-supported services even known as web-services, where a user may request services through a website of the service provider, and the service provider provides the requested service. Examples of services on the web are: Buying an airline ticket or purchase a book from an e-commerce provider. An information system that supports service implementation is called service-oriented information system. An architecture that provides support for the implementation of web-services is known as Service-Oriented Architecture (SOA). With the integration of cloud computing and service-oriented computing, the services are now provided through a cloud. These services are not only related to booking travel or hotel, but they also include infrastructures as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Thus, Cloud Computing provides to the customers Softwares, Platforms and Infrastructures as a service via the Cloud (Zhang *et al.*, 2010). The idea behind cloud computing is to provide computing, infrastructures, and platform as a service in a very simple way just like we use electricity, or gas as a service. Thus, a cloud service supplier offers different types of service to the consumer. The service could be to use the cloud for computing, for database management, or for application support big data analysis such as biological data. The technological advances that characterize Cloud Computing respect to distributed computing and SOA, it is its adaptability to customer demand. Users can request new applications and infrastructure on demand. Indeed Clouds or, Cloud Computing is much more than accessing applications using the pay per view on-demand model. Cloud computing provides a virtual high-performance computing (HPC) or a high-performance environment for scientific simulations that can be handled through a simple web-browser (Buyya *et al.*, 2009). One of the major's keystones of Cloud resources is their quick access and easy-to-use nature. In fact, HPC resources have to be set up by expert figures with specific skills. Indeed, these skills are not widespread among all the possible users because HPC resources set up vary quite significantly. Also, a supercomputer will offer an environment tailored to dig with large parallel scientific applications, only after that the several different versions of compilers and libraries are available to users and software is kept up-to-date as part of the service, requiring a precise version of the operating system.

Furthermore, HPC systems present restrictive regulation policies to handle the resource (Mishra *et al.*, 2013). Given the shared nature of the resources among several users all competing for them, manage them through a scheduling system is mandatory. The resources that a user might employ are regulated by the scheduling system, and the policies are implemented through the scheduling system by the owners/operators of the infrastructure, for each user are necessary. Thus, scheduling policies have to take into account the number of active processes and how long a single process can run. The most used fashion in HPC is set a runtime threshold for all the computational process, after that the process is killed. Consequently, users have to adapt their programs to cope with the maximum runtime threshold by regularly saving data to disk to enable restart if the application is ended. These changes have a cost, both regarding the computational resources used (saving output data can require a long time to have written on disk) and, the increasing effort in the development necessary to modify programs to undertake this functionality. Cloud resources, on the other hand, generally have no such restrictions. User purchase computing resources are paying for them running them for as long as it is required. Users can also purchase as many resources as needed, dependent upon the available resources in the Clouds, and the associated cost. Indeed, cloud vendors, e.g., Amazon, Microsoft supplies computational resources via Cloud, which can be bought by anyone, just creating an account and paying for their use. To have access to an NGS analysis suite on the cloud users have to look for the service, for example, AWS (see “Relevant Websites section”) available on Amazon create an account and start to use the service. In this way is the Cloud environment to keep the different versions of compilers and libraries up-to-date as well the opportune version of the operating system. Finally, setup and usage of Cloud resources can be more straightforward than HPC, especially for less experienced users, the higher level of control over the environment by Graphical User Interface offers some advantages, to unskilled users to quickly reach their computing simulation needs.

Programming Languages for Parallel Architectures

A parallel program can be written by using a parallel programming language. Programmers to write parallel multithreaded programs can use the standard POSIX thread library (Pthreads) using the C language, whereas in Java programmers can use the native Thread class. If programmers execute this code on a shared memory parallel computer, each thread will run on a different processor simultaneously, yielding a parallel speedup. A better and more straightforward approach to write parallel programs is to use parallel programming libraries. Parallel libraries simplify the process of writing code abstracting low-level thread details reducing the effort needed to write a parallel program. Writing low-level multi-threads applications, it requires a great deal of effort to write the code that sets up and synchronize the multiple threads of an SMP parallel programs to ensure the consistency of information. Bioinformatics programmers are interested in solving problems in their domains, such to search a massive DNA sequence database or to predict the tertiary structure of an unknown protein, and they are not interested in writing low-level thread code. Indeed, many programmers may lack the expertise to write multi-thread code. Instead, parallel programming library encapsulates the low-level multi-thread code that is the same in any parallel program, presenting to the programmer easy-to-use, high-level parallel programming abstractions. In this way, programmers can focus on solving the domain problem by using the facilities provided by the parallel programming libraries.

Shared Memory

OpenMP (see “Relevant Websites section”) is the standard library to write parallel programming on a shared memory architectures. The first version of OpenMP was published in 1997 for the Fortran language and subsequently in 1998 for the C, and C++ languages, the latest version available today of OpenMP is the 4.5 released in 2015. The OpenMP API provides only user-directed parallelization, wherein the programmer explicitly defines the actions to be taken by the compiler and runtime system in order to execute the program in parallel. Standard OpenMP does not support Java. OpenMP is compatible with C, and C++, although these languages natively support multithread programming. Conversely, from C, and C++, Fortran language does not support multithreaded programming, a program wrote in Fortran can be made multithreading only by using OpenMP by inserting special OpenMP keywords called *pragmas* in the source code. OpenMP Pragmas are available even in C and C++, making easy to create multi-thread programs. By adding pragmas, programmers are designing which section of the program has to be parallelized, to be execute by multiple threads. When programmer located all the sections of code that is necessary to be executed in parallel, it is needed compile the annotated source code through an appropriate OpenMP compiler. The OpenMP compiler, finding the OpenMP pragmas through the code, rewrites the sequential source code in parallel code, adding the necessary low-level threading code. As the final step, the compiler compiles the now-multithreaded program as a regular Fortran, C or C++ program, that can be run as usual on an SMP parallel computer. Here is reported a simple example that explain how to use OpenMP pragmas. As example, suppose that it is necessary to figure out all the shortest path in a biological pathway. We can use the Floyd-Warshall’s algorithm. The sequential code is conveyed in [Listing 1](#).

Using the input presented in [Listing 2](#), we get the following output let see [Listing 3](#).

To make the same code be able to be execute on multiple threads we need to add in the code the OpenMP pragmas, as conveyed in [Listing 4](#).

Thus adding the OpenMP pragmas in the correct position in the code quickly we transform a sequential program in a parallel program. The “*#pragma omp parallel*” specifics to the compiler the start of a piece of code that can be executed in parallel on multiple threads. Because the number of threads is not specified in the pragma, the number of threads is defined at runtime. Inside

```

void floydWarshall(int graph[][VERTICES_COUNT]){
int distance[VERTICES_COUNT][VERTICES_COUNT];
for (int i = 0; i < VERTICES_COUNT; i++)
    for (int j = 0; j < VERTICES_COUNT; j++)
        distance[i][j] = graph[i][j];
        for (int k = 0; k < VERTICES_COUNT; k++){
            for (int i = 0; i < VERTICES_COUNT; i++){
                for (int j = 0; j < VERTICES_COUNT; j++){
                    if (distance[i][k] + distance[k][j] < distance[i][j])
                        distance[i][j] = distance[i][k] + distance[k][j];
                }
            }
        }
    Print(distance);
}

```

Listing 1 Floyd-Warshall's procedure.

```

int graph[VERTICES_COUNT][VERTICES_COUNT] = {
    { 0, 5, INF, 10 },
    { INF, 0, 3, INF },
    { INF, INF, 0, 1 },
    { INF, INF, INF, 0 }
};
FloydWarshall(graph);

```

Listing 2 Floyd-Warshall's Input example.

OUTPUT: Shortest distances between every pair of vertices

```

0    5    8    9
INF  0    3    4
INF INF  0    1
INF INF INF  0

```

Listing 3 Floyd-Warshall's output example.

the parallel region, each thread gets its copies of the i , j , and k variables, the other variables $VERTICES_COUNT$ and $DISTANCE$, which are declared outside the parallel region, are shared. All the threads execute the external cycle. However, when they reach the middle cycle, the “*for pragma*” states that the middle cycle is to be executed as sharing parallel cycle. That is, the iterations in the middle cycle are partitioned among the scheduled threads, so each thread executes a subset of iterations. Because no cycle schedule is specified, a default schedule is used. Thus, implicitly each thread at the end of the middle loop do an implicit wait, before proceeding to the next outer loop iteration.

```

#pragma omp parallel
void floydWarshall(int graph[][VERTICES.COUNT]){
    int distance[VERTICES.COUNT][VERTICES.COUNT];
    for (int i = 0; i < VERTICES.COUNT; i++)
        for (int j = 0; j < VERTICES.COUNT; j++)
            distance[i][j] = graph[i][j];
        for (int k = 0; k < VERTICES.COUNT; k++){
            #pragma omp for
            for (int i = 0; i < VERTICES.COUNT; i++){
                for (int j = 0; j < VERTICES.COUNT; j++){
                    if (distance[i][k] + distance[k][j] < distance[i][j])
                        distance[i][j] = distance[i][k] + distance[k][j];
                }
            }
        }
    Print(distance);
}

```

Listing 4 Floyd-Warshall's OpenMP parallel version.

```

#include <iostream>
__global__ void kernel( void ) {
}

int main( void ) {
    kernel<<<1,1>>>();
    printf( "Hello ,_World!_in_CUDA\n" );
    return 0;
}

```

Listing 5 Hello world code in CUDA.

CUDA

CUDA (see “Relevant Websites section”) is a parallel computing platform and application programming interface (API) developed by Nvidia. A CUDA program consists of host and device code. Usually, the host (CPU) executes the parts of the program that present low parallelism, i.e., the data partition among the device’s cores, whereas the statements with high data parallelism, they are performed in the device code. The NVIDIA C compiler (*nvcc*) extracts the two parts during the compilation process. The host code is ANSI C code, and it is further compiled with the host’s standard C compilers to be executed on an ordinary CPU. The device code is written using ANSI C at which are added keywords called kernels, for labeling data-parallel functions, and their associated data structures. The device code is typically additionally compiled by the *nvcc* to be executed on a GPU device. The kernel functions (or, kernels) generate a high number of threads to employ data parallelism. The CUDA threads are different from CPU threads because CUDA threads are of much lighter weight than the CPU threads. Thus, to generate and schedule CUDA threads require less cycle respect to create and schedule CPU threads that need thousands of clock cycles. These differences are due to the efficient hardware support of CUDA’s architecture. [Listing 5](#) introduce a very simple *Hello world* program in CUDA.

The differences between a standard hello world program and hello world in CUDA are: An empty function called `kernel()` and the keyword `__global__`, and a call to the “`kernel <<<1,1>>>()`” function.

The `__global__` qualifier informs the compiler that the function has to be compiled to be executed on the device instead of the host.

As example, suppose that we need to sum two sequence of genes expression to detect if the sum of the expression for each genes is over a threshold. We can model the problem as sum of two vectors. In particular we write the sum function to be execute on the device by using CUDA.

In the [Listing 6](#) we used `cudaMalloc()` to allocate the three arrays. Using `cudaMemcpy()`, we copy the input data from the host to the device with the kernel `cudaMemcpyHostToDevice`, and we copy back the results from the device to the host with by using the kernel `cudaMemcpyDeviceToHost`. Finally, by using `cudaFree()`, we release the allocated memory. The annotation `<<<N,1>>>` added to the `add()` methods, it allows to execute the `add()` method in the device code, from the host code in `main()`. Let see the meaning of the two parameters in the triple angle brackets annotation. Let define a simple CUDA kernel `cudaKernel <<<nBlocks, nThreadsPerBlock>>>` the first parameter `nBlocks` defines the number of thread blocks, and the second parameter `nThreadsPerBlock` defines the number of threads within the thread block. Launching the following kernel `<<<2,1>>>()`, the number of simultaneously running threads is given by the product of both parameters that is equals to 2 in this case. Launching the following kernel `<<<N,1>>>()` we are running `N` copies of kernel code, but how can we know which block is currently running? By using `blockIdx.x` to index arrays, each block handles different indices, thus Kernel can refer to its block. More example and more details on CUDA programming can found at the CUDA's web site provided in “Relevant Websites section”, and in [Sanders and Kandrot \(2010\)](#) and [Kirk and Wen-Mei \(2016\)](#).

Message Passing

Message Passing Interface (MPI) is a standard application program interface (API) with which write parallel programs in a distributed environment by sending and receiving messages. The first version of MPI has been released in the 1994 supporting Fortran and C. The MPI supports Fortran, C, and C++. Like OpenMP, even MPI is not compatible with Java. MPI does not require particular compiler; it is just a message passing protocol for programming parallel computers. Parallels programs are writing using the MPI library routines as needed to send and receive messages. Programs are executed on parallel cluster computers through an appropriate MPI launcher program. The MPI API takes care of all the low-level details, i.e., setting up network connections between processes and transferring messages back and forth. Conversely, form OpenMP that uses pragmas to specify to the compiler which section of code has to be parallelized, in MPI there no needs to use any tag in the code. MPI programmers write Fortran, C, and C++ code, as usual, including or importing the MPI library, and adding through the program the calls to the MPI subroutines to send and receive messages among the nodes. The final code is compiled using the Fortran, C or C++ standard compiler. To execute the compiled program on a parallel computer, it is mandatory use a specific MPI launcher application. In this way, the launcher takes care to run the program in multiple processes on the nodes of a cluster. We'll use OpenMPI (see “Relevant Websites section”) to present the programming examples in this section.

As simple example in [Listing 7](#), we report a cluster parallel program for the “HelloWorld” algorithm in MPI by using C++.

To compile the code in [Listing 7](#), open the terminal application go to the folder containing the “HelloWorldMPI.cpp” file and type the command in [Listing 8](#):

When the compiling process is done to run the program type the command in [Listing 9](#):

Programming Languages for Distributed Architectures

This Section summarizes some relevant programming approaches for distributed architectures, including the RESTful model, Hadoop, MapReduce and Globus.

Restful Model

REpresentational State Transfer REST is an architectural style established to develop, create and organize distributed systems. The term REpresentational State Transfer REST has been coined in 2000 by Roy Fielding in his doctoral thesis (see “Relevant Websites section”) REST is only an architectural style for building scalable web services; it is not a standard guidelines to follow in order to obtain a RESTful architecture ([Rodriguez, 2008](#)). In RESTful systems, communication occurs over the Hypertext Transfer Protocol (HTTP). Because REST is resource-based architecture, REST exploits the HTTP methods with which access to the resources. In REST resources are identified through a Unified Resource Identifier (URI), and resource should be compatible with the HTTP standard operations. Resources in REST have different representations, e.g., text, XML, JSON, etc. By using uniform resource identifiers (URIs) and HTTP verbs, it is possible to perform actions to the resources. If we define the resource “/office/agents”, it is possible to retrieve information about an agent resource by using the HTTP verb *Get*, create a new agent resource utilizing the HTTP verb *Post*, update an existing agent resource using the HTTP verb *Put*, and delete an agent resource through the HTTP verb *Delete*. Web service APIs that follow the REST architecture style, they are known as RESTful APIs. RESTful APIs uses XML and JSON as the data format

```

#include <iostream>
#include <stdio.h>
#include <split.h>

#define N 10
int main( void ) {
    int a[N], b[N], c[N];
    int *dev_a, *dev_b, *dev_c;

    // GPU memory allocation
    HANDLEERROR(cudaMalloc((void**)&dev_a, N * sizeof(int)));
    HANDLEERROR(cudaMalloc((void**)&dev_b, N * sizeof(int)));
    HANDLEERROR(cudaMalloc((void**)&dev_c, N * sizeof(int)));
    /** fill the arrays 'a' and 'b' on the CPU by reading the
    values into an input file */
    FILE *file;
    file = fopen("genesExpression.txt", "r");
    if (file) {
        int i = 0;
        while ((line = readLine(file)) != EOF){
            int exp [] = split(line);
            a[i] = exp[0];
            b[i] = exp[1];
            i++;
        }
        fclose(file);
    }
    // copy the arrays 'a' and 'b' to the GPU
    HANDLEERROR(cudaMemcpy(dev_a, a, N * sizeof(int), cudaMemcpyHostToDevice));
    HANDLEERROR(cudaMemcpy(dev_b, b, N * sizeof(int), cudaMemcpyHostToDevice));

    add<<<N,1>>>>( dev_a, dev_b, dev_c );
    // copy the array 'c' back from the GPU to the CPU
    HANDLEERROR(cudaMemcpy( c, dev_c, N * sizeof(int), cudaMemcpyDeviceToHost));
    // visualize the results
    for (int i=0; i<N; i++) {
        printf( "%d+=%d=%d\n", a[i], b[i], c[i] );
    }

    // free the memory allocated on the GPU
    cudaFree( dev_a );
    cudaFree( dev_b );
    cudaFree( dev_c );

    return 0;
}

//definition of add function
__global__ void add( int *a, int *b, int *c ) {
    int tid = blockIdx.x;
    if (tid < N)
        // sum the data at this index
        c[tid] = a[tid] + b[tid];
}

```

Listing 6 Sum of two arrays in CUDA.

```

// filename: HelloWorldMPI.cpp
# include <iostream>
using namespace std;
# include "mpi.h"
int main ( int argc , char *argv [] ) {
    int id , p, name_len;
    char processor_name[MPLMAXPROCESSOR_NAME];
    // Initialize MPI.
    MPI:: Init(argc , argv);
    // Get the number of processes.
    p = MPI::COMM_WORLD. Get_size ();
    // Get the single process ID.
    id = MPI::COMM_WORLD. Get_rank ();
    MPI_Get_processor_name(processor_name , &name_len);
    // Print off a hello world message
    cout << "Processor_" << processor_name << "_ID=" <<
        id << "_Hello_World_from_MPI!\n";
    // Terminate MPI.
    MPI:: Finalize ();
    return 0;
}

```

Listing 7 The HelloWorld program in MPI by using C++.

```
>$ mpic++ HelloWorldMPI.cpp -o HelloWorldMPI
```

Listing 8 The HelloWorld program in MPI by using C++.

```
>$ mpirun -np 4 HelloWorldMPI
Output:
Processor ProcessorMPI ID=0 Hello World from MPI!
Processor ProcessorMPI ID=3 Hello World from MPI!
Processor ProcessorMPI ID=1 Hello World from MPI!
Processor ProcessorMPI ID=2 Hello World from MPI!
```

Listing 9 The HelloWorld program in MPI by using C++.

for exchanging data between client applications and servers. AS example, in [Listing 10](#) we present a RESTful web services wrote in Java by using (Jersey/JAX-RS) APIs.

The class *HelloWorld* registers itself as a get resource by using the *@GET* annotation. Using the *@Produces* annotation, it defines that it can deliver several Multipurpose Internet Mail Extensions (MIME) types such as plain text, XML, JSON and HTML. *@Path* annotation defines the URI *"/HelloWorld"* where the service is available. MIME type is a standard with which to denote the nature and format of a document in internet.

```
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/HelloWorld")
public class HelloWorld {

    // This method is called if TEXT_PLAIN is request
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String helloWorldPlainText() {
        return "Hello_WORLD";
    }

    // This method is called if XML is request
    @GET
    @Produces(MediaType.TEXT_XML)
    public String helloWorldXML() {
        return "<?xml version='1.0'?>" + "<hello>_Hello_WORLD" + "</hello>";
    }

    // This method is called if HTML is request
    @GET
    @Produces(MediaType.TEXT_HTML)
    public String helloWorldHTML() {
        return "<html>_" + "<title>" + "Hello_WORLD" + "</title>"
            + "<body><h1>" + "Hello_WORLD" + "</body></h1>" + "</html>_";
    }

    // This method is called if JSON is request
    @GET @Produces(MediaType.APPLICATION_JSON)
    public String getMyBean() {
        return {"id":1,"content":"HelloWorld!"} ;
    }

}
```

Listing 10 The HelloWorld web server in Java by using (Jersey/JAX-RS).

Hadoop

The Apache Hadoop (see “Relevant Websites section”) software library is the basis for the development of advanced distributed and scalable applications, able to deal with big data. Hadoop consists of two main components: The Hadoop Distributed Filesystem (HDFS) and MapReduce. The HDFS is a data storage as well as a data processing filesystem. HDFS is designed to store and provide parallel, streaming access to large quantities of data (up to 100s of TB). HDFS storage is disseminated across a cluster of nodes; a single large file could be stored across multiple nodes in the cluster. A file is split into blocks called chunks, with a default size of 64MB. HDFS is designed to store and handle large files efficiently. The Hadoop ecosystem now comprises several components for databases, data warehousing, image processing, deep learning, and natural language processing, are only a few example.

Mapreduce

MapReduce (see “Relevant Websites section”) is a programming paradigm that allows to process big data with a parallel distributed algorithm on clusters providing massive scalability and fault-tolerant. MapReduce application consists of two steps: the Map and Reduce step, where data are processed using key/value pairs. In the Map step, the input dataset is treated by using the specified Map function. The Map function should be designed to count the number of unique occurrences of an SNP in a dataset, for example. The Map step produces an intermediate set of key/value pairs, which are partitioned in chunks and then assigned to the Reduce function to be elaborated. The Reduce function produces the output of the MapReduce application. The MapReduce framework consists of a master the ResourceManager, and one slave the NodeManager per single cluster-node. The ResourceManager is the master node, it accepts job submissions from clients, and it starts the process called ApplicationMaster to run the jobs, assigning the resources required for a job. A ResourceManager consists of two elements: Scheduler and ApplicationsManager. The Scheduler allocates resources and does not participate in running or monitoring the job. The ApplicationsManager accepts the job submissions from the clients and starts the ApplicationMaster to execute the submitted job and to restart failed ApplicationMaster. The ApplicationMasters are application specific, with one ApplicationMaster for each task. The NodeManager runs resource containers on the machine and monitors the resource usage of the applications running in the resource containers on the computer, reporting the resource usage to the ResourceManager. [Listing 11](#) shows a simple example of using MapReduce to count the occurrence of SNPs in a given dataset.

Globus

The open source Globus Toolkit (see “Relevant Websites section”) is a set of tools useful for building a grid infrastructure. The toolkit includes software services and libraries for resource monitoring, discovery, and management, plus security and file management. Globus Toolkit consists of three main components that are: Resource management, Data management, and Information Services. All the three components are built on top of the Grid Security Infrastructure (GSI), providing security functions, including authentication, confidential communication, authorization, and delegation. The resource management component provides support for, resource allocation, remote jobs submission, to collect results and monitoring job status and progress. Globus Toolkit does not have its job scheduler to allocate available resources and to dispatch jobs to proper machines. Globus Toolkit provides the tools and APIs necessary to implement its own scheduler. The information services component provides API and interfaces for collecting information in the grid by using the Directory Access Protocol (LDAP). The data management component provides support for transfer files among machines in the grid and for the supervision of these transfers.

Parallel and Distributed Bioinformatics

Classically High-performance computing (HPC) has been used in physics, mathematics, aerodynamics and so on, scientific areas where an intensive computational power it is necessary. Nowadays, HPC is being used even more often in Life Sciences, Medical Informatics, and Bioinformatics, to face the increasing amount of available experimental data ([Cannataro, 2009](#)). Current computational and systems biology involve a large number of biological data sources and database disseminated all over the world wide web, requiring efficient and scalable algorithms able to analyze these vast amounts of available experimental data. Besides, the intensification of use of high-throughput experimental assays, such as Next Generation Sequencing, microarray, Genome-Wide Study (GWAS) and mass spectrometry (Mass spectrometer), are producing massive volumes of data per single experiment, contributing significantly to increase the amount of data generated daily. Indeed, the storage and analysis of such data are becoming a bottleneck to bring to light useful knowledge hide in these apparently unrelated data. Thus, the developing of high-performance computing HPC is mandatory to make it possible to handle this vast amount of data, to be suitable in clinical practise or to develop tailored treatments for a single patient based on its own genetic features. Grids have been thought as universal tools whose aim is to provide a general solution for building scalable and ubiquity parallel collaborative environments. However, the high flexibility in the application of the Grid has limited the use of Grids only in the field of information technology. Indeed, the use of Grid-toolkits with which to develop Grid-services requires advanced computer skills. Thus, to encourage the use of grids in specific sectors, several dedicated Grids have been designed. The aim of dedicated Grids within a particular application domain is to provide ready to use Grid-services able to face the specific requirements and problems of that field. In the last times, many applications of Bioinformatics and Systems Biology have been developed on the Grids. The term BioGrids refers to the

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class SNPCount {

    public static class SNPMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable iw = new IntWritable(1);

    private Text snp = new Text();

    public void map(Object key, Text value, Context context
        ) throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            snp.set(itr.nextToken());
            context.write(snp, iw);
        }
    }
}
```

Listing 11 The SNP occurrences count application in Java by using MapReduce.

deployment of bioinformatics analysis pipeline to the Grid, yielding to a high-performance computing infrastructure dedicated to solve biomedical and bioinformatics problems exploiting the services of the Grid (Ellisman *et al.*, 2004). BioGrids goal is to provide the software and network infrastructures for Virtual Collaborative Laboratory integrating bioinformatics, biological and medical knowledge, through easy access and use of the Grid resources; integrated access to biological databases; and finally the support to application modeling and design, usually provided through workflow modeling.

Also, even Clouds are employed to host and deploy bioinformatics applications. Recent researches indicated that cloud computing could enhance healthcare services and biomedical study (Ahuja *et al.*, 2012; Rosenthal *et al.*, 2010), by allowing new potentialities. A significant thrust toward the adoption of Cloud in healthcare and bioinformatics is the growth of big data produced by high-throughput assays (Greene *et al.*, 2014). Because the amount of digital genomics, interactomics information increase, develop tools with the capability to dig with this flow of data is mandatory. Buried in this data there are the knowledge to extract to make clinical advances actuality, but at today that are not very accessible to the clinical researchers. Cloud Computing might enable data sharing and


```

public static class SNPReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
        Context context
        ) throws IOException, InterruptedException {

        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();

    Job job = Job.getInstance(conf, "SNP_count");
    job.setJarByClass(SNPCount.class);
    job.setMapperClass(SNPMapper.class);
    job.setCombinerClass(SNPReducer.class);
    job.setReducerClass(SNPReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

}
}

```

Listing 11 (Continued).

integration at vast scale in an easy and simple way. The volume of data produced by Medical imaging might reach the magnitude of petabytes due to high-resolution of the imaging instruments. Consequently, it is evident that the cloud computing will provide a possible contribution to satisfy computational needs related to the reconstruction and analysis of medical images, allowing a full sharing of imaging as well as advanced remote analysis. The cloud computing represents a solution for the problems of storing and processing data in the context of bioinformatics. Therefore, classical computational infrastructures for data processing have become ineffective and hard to maintain. The traditional bioinformatics analysis requires to download public data (e.g., NCBI, Ensembl), and the download and installation of the proper software or more than one with which to analyze the downloaded data. By porting data and software in the cloud, it is possible to provide them as a service, obtaining a level of integration that improves the analysis and the storage of bioinformatics big-data. In particular, as a result of this unusual growth of data, the requirement of data as a service (Data as a Service, DaaS) is of absolute importance. DaaS provides data storage in a dynamic virtual space hosted in the cloud, allowing users to update data through a browser-web. An example of DaaS is the Amazon Web Services (AWS) (see “Relevant Websites section”) ([Fusaro et al., 2011](#)), which provides a centralized repository of public data sets, including data from GenBank, Ensembl, 1000 Genomes Project,

Unigene, and Influenza Virus. There have been several efforts to develop cloud-based tools known as Software as a Service (SaaS), to perform several bioinformatics tasks through a simple web-browser. In this way, researchers can focus only on the definition of the data analysis methodology without worry about if the hardware available is powerful enough for the simulation and avoiding to take care of software updating, managing, and so on. Example of SaaS are: Cloud4SNP (Agapito *et al.*, 2013) is a private Cloud bioinformatics tool for the parallel preprocessing and statistical analysis of pharmacogenomics SNP DMET microarray data. It is a Cloud version of DMET-Analyzer (Guzzi *et al.*, 2012), that has been implemented on the Cloud employing the Data Mining Cloud Framework (Marozzo *et al.*, 2013a), a software environment for the design and execution of knowledge discovery workflows on the Cloud (Marozzo *et al.*, 2013b). Cloud4SNP allows to statistically test the significance of the presence of SNPs in two classes of samples using the well known Fisher test. ProteoCloud (see "Relevant Websites section") (Muth *et al.*, 2013) is a freely available, full-featured cloud-based platform to perform computationally intensive, exhaustive searches using five different peptide identification algorithms. ProteoCloud is open source, including a graphical user interface, making easy to interact with the application. In addition to DaaS and SaaS there are the Platform as a Service PaaS. The most known and used bioinformatics platform in the cloud is the Galaxy Cloud (see "Relevant Websites section"). Galaxy cloud-based is a platform for the analysis of big volume of data, allowing to the users customize the deployment as well as retain complete control on the instances and the associated data. The current version of Galaxy is available on Amazon Web Services. Another example of PaaS is CloudMan (see "Relevant Websites section") allows bioinformatics researchers quickly deployment, customize, and share of their entire cloud analysis environment, along with data, tools, and configurations.

Closing Remarks

High performance computing, distributed systems and database technology play a central role in Computational Biology and Bioinformatics. High-throughput experimental platforms such as microarray, mass spectrometry, and next generation sequencing, are producing the so called *omics* data (e.g., genomics, proteomics and interactomics data), that are at the basis of Computational Biology and Bioinformatics. These big omics data have an increasing volume due to the high resolution of such platforms and because biomedical studies involve an increasing number of biological samples. Moreover, the digitalization of healthcare data, such as laboratory tests and administrative data, is increasing the volume of healthcare data that is coupled to omics and clinical data. Finally, the use of body sensors and IoT (Internet of Things) devices in medicine is another source of big data. This big data trend poses new challenges for computing in bioinformatics related to the efficient preprocessing, analysis and integration of omics and clinical data. Main technological approaches used to face those challenges are: high-performance computing; Cloud deployment; improved data models for structured and unstructured data; novel data analytics methods such as Sentiment Analysis, Affective Computing, and Graph Analytics; novel privacy-preserving methods. This articles surveyed main computing approaches used in bioinformatics, including parallel and distributed computing architectures, parallel programming languages, novel programming approach for distributed architecture and Cloud.

Acknowledgement

"This work has been partially funded by the Data Analytics Research Center - University Magna Graecia of Catanzaro, Italy."

See also: Algorithms Foundations. Computational Immunogenetics. Computational Pipelines and Workflows in Bioinformatics. Dedicated Bioinformatics Analysis Hardware. Genome Databases and Browsers. Host-Pathogen Interactions. Learning Chromatin Interaction Using Hi-C Datasets. Techniques for Designing Bioinformatics Algorithms

References

- Agapito, G., Cannataro, M., Guzzi, P.H., *et al.*, 2013. Cloud4snp: Distributed analysis of snp microarray data on the cloud. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics, BCB'13, pp. 468–475. New York, NY: ACM. Available at: <http://doi.acm.org/10.1145/2506583.2506605>.
- Ahuja, S.P., Mani, S., Zambrano, J., 2012. A survey of the state of cloud computing in healthcare. *Network and Communication Technologies* 1 (2), 12. Available at: <https://doi.org/10.5539/nct.v1n2p12>.
- Aronova, E., Baker, K.S., Oreskes, N., 2010. Big science and big data in biology: From the international geophysical year through the international biological program to the long term ecological research (Iter) network, 1957–present. *Historical Studies in the Natural Sciences* 40 (2), 183–224.
- Bajo, J., Zato, C., de la Prieta, F., de Luis, A., Tapia, D., 2010. Cloud computing in bioinformatics. In: *Distributed Computing and Artificial Intelligence*. Springer, pp. 147–155.
- Bote-Lorenzo, M.L., Dimitriadis, Y.A., Gómez-Sánchez, E., 2004. Grid characteristics and uses: A grid definition. In: *Grid Computing*. Springer, pp. 291–298.
- Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., 2009. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 25 (6), 599–616.
- Caicedo, J.C., Cruz, A., Gonzalez, F.A., 2009. Histopathology image classification using bag of features and kernel functions. In: *Conference on Artificial Intelligence in Medicine in Europe*. Springer, pp. 126–135.
- Calabrese, B., Cannataro, M., 2016. Cloud computing in bioinformatics: Current solutions and challenges. Technical Report PeerJ Preprints.
- Cannataro, M., 2009. Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine, and Healthcare. vol. 1. IGI Global.
- Cannataro, M., Talia, D., 2003a. The knowledge grid. *Communications of the ACM* 46 (1), 89–93.
- Cannataro, M., Talia, D., 2003b. Towards the next-generation grid: A pervasive environment for knowledge-based computing. In: *Proceedings of the International Conference on Information Technology: Coding and Computing [Computers and Communications]*, ITCC 2003. IEEE, pp. 437–441.

- Cannataro, M., Talia, D., 2004. Semantics and knowledge grids: Building the next-generation grid. *IEEE Intelligent Systems* 19 (1), 56–63.
- Ellisman, M., Brady, M., Hart, D., *et al.*, 2004. The emerging role of biogrids. *Communications of the ACM* 47 (11), 52–57. Available at: <http://doi.acm.org/10.1145/1029496.1029526>.
- Flynn, M., 2011. Flynn's taxonomy. In: *Encyclopedia of parallel computing*. Springer, pp. 689–697.
- Fusaro, V.A., Patil, P., Gafni, E., Wall, D.P., Tonellato, P.J., 2011. Biomedical cloud computing with amazon web services. *PLOS Computational Biology* 7 (8), 1–6. Available at: <https://doi.org/10.1371/journal.pcbi.1002147>.
- Greene, C.S., Tan, J., Ung, M., Moore, J.H., Cheng, C., 2014. Big data bioinformatics. *Journal of Cellular Physiology* 229 (12), 1896–1900. Available at: <https://doi.org/10.1002/jcp.24662>.
- Gurcan, M.N., Boucheron, L.E., Can, A., *et al.*, 2009. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering* 2, 147–171.
- Guzzi, P.H., Agapito, G., Di Martino, M.T., *et al.*, 2012. Dmet-analyzer: Automatic analysis of affymetrix dmet data. *BMC Bioinformatics* 13 (1), 258. Available at: <https://doi.org/10.1186/1471-2105-13-258>.
- Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., Tanabe, M., 2011. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research* 40 (D1), D109–D114.
- Kikinis, R., Warfield, S., Westin, C.-F., 1998. High performance computing (hpc) in medical image analysis (mia) at the surgical planning laboratory (spl). In: *Proceedings of the 3rd High Performance Computing Asia Conference & Exhibition*. vol. 8.
- Kirk, D.B., Wen-Mei, W.H., 2016. *Programming Massively Parallel Processors: Hands-on Approach*. Morgan kaufmann.
- Loman, N.J., Constantinidou, C., Chan, J.Z., *et al.*, 2012. High-throughput bacterial genome sequencing: An embarrassment of choice, a world of opportunity. *Nature Reviews Microbiology* 10 (9), 599.
- Marozzo, F., Talia, D., Trunfio, P., 2013a. A cloud framework for big data analytics workflows on azure. *Cloud Computing and Big Data* 23, 182.
- Marozzo, F., Talia, D., Trunfio, P., 2013b. Using clouds for scalable knowledge discovery applications. In: Caragiannis, I., Alexander, M., Badia, R.M., *et al.* (Eds.), *Euro-Par 2012: Parallel Processing Workshops*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 220–227.
- Marx, V., 2013. Biology: The big challenges of big data. *Nature* 498, 255–260.
- Mell, P., Grance, T., 2011. The NIST definition of cloud computing. *Communications of the ACM* 53 (6), 463–511.
- Mishra, A., Mathur, R., Jain, S., Rathore, J.S., 2013. Cloud computing security. *International Journal on Recent and Innovation Trends in Computing and Communication* 1 (1), 36–39.
- Muth, T., Peters, J., Blackburn, J., Rapp, E., Martens, L., 2013. Proteocloud: A full-featured open source proteomics cloud computing pipeline. *Journal of Proteomics* 88, 104–108. Available at: <http://www.sciencedirect.com/science/article/pii/S1874391913000134>.
- O'Driscoll, A., Daugelaite, J., Sleator, R.D., 2013. 'Big data', hadoop and cloud computing in genomics. *Journal of Biomedical Informatics* 46 (5), 774–781.
- Owens, J.D., Houston, M., Luebke, D., *et al.*, 2008. GPU computing. *Proceedings of the IEEE* 96 (5), 879–899.
- Pharr, M., Fernando, R., 2005. *Gpu gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Addison-Wesley Professional.
- Rodriguez, A., 2008. *Restful web services: The basics*. IBM developerWorks.
- Rosenthal, A., Mork, P., Li, M.H., *et al.*, 2010. Cloud computing: A new business paradigm for biomedical information sharing. *Journal of Biomedical Informatics* 43 (2), 342–353. Available at: <http://www.sciencedirect.com/science/article/pii/S1532046409001154>.
- Rumelhart, D.E., McClelland, J.L., Group, P.R., *et al.*, 1987. *Parallel Distributed Processing*. vol. 1. Cambridge, MA: MIT Press.
- Sanders, J., Kandrot, E., 2010. *CUDA by example: An Introduction To General-purpose GPU Programming*. Addison-Wesley Professional.
- Sunderam, V.S., 1990. PVM: A framework for parallel distributed computing. *Concurrency and Computation: Practice and Experience* 2 (4), 315–339.
- Veta, M., Pluim, J.P., Van Diest, P.J., Viergever, M.A., 2014. Breast cancer histopathology image analysis: A review. *IEEE Transactions on Biomedical Engineering* 61 (5), 1400–1411.
- Zhang, Q., Cheng, L., Boutaba, R., 2010. Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications* 1 (1), 7–18.

Relevant Websites

<https://developer.nvidia.com/cuda-toolkit>
CUDA Toolkit.

http://www.ics.uci.edu/fielding/pubs/dissertation/rest_arch_style.htm
Fielding Dissertation.

<http://galaxy.psu.edu>
Galaxy Community Hub.

<http://toolkit.globus.org/toolkit/>
Globus Toolkit.

<https://aws.amazon.com/health/genomics/>
Genomics Cloud Computing - Amazon Web Services (AWS).

<http://toolkit.globus.org/toolkit/>
Globus Toolkit - Globus.org.

<http://www.mersenne.org/>
Great Internet Mersenne Prime Search - PrimeNet.

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
MapReduce Tutorial.

<http://www.openmp.org>
OpenMP: Home.

<http://www.open-mpi.org/>
Open MPI: Open Source High Performance Computing.

<https://www.opensciencegrid.org>
Open Science Grid.

<https://code.google.com/archive/p/proteocloud/>
Proteomics Cloud Computing Pipeline.

<http://aws.amazon.com/publicdatasets>
Registry of Open Data on AWS (Amazon Web Services).

<https://setiathome.berkeley.edu>
SETI@home - UC Berkeley.

<http://cloudman.irb.hr>
The CloudMan Project: Cloud clusters for everyone.

<http://hadoop.apache.org>
Welcome to Apache™ Hadoop®!

Computing Languages for Bioinformatics: Perl

Giuseppe Agapito, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Perl is a general-purpose scripting language introduced by Larry Wall in 1987 (Wall *et al.*, 1994, 2000). Perl was developed to connect different languages and tools together by making compatible the various data format between them. The reasons to spur Wall to create Perl have been, to gather together all the best features of C, C++, Lisp, awk, Smalltalk 80, Pascal and Unix Shell languages without their disadvantages. Perl became very popular as server-side script language, but with the time extended its application domain from system administrator tasks, managing databases, as well as object-oriented programming, finance, bioinformatics, and Graphical User Interface (GUI) programming. Perl is not an acronym, although we can refer to PERL as “Practical Extraction and Reporting Language.” The language used to develop Perl is C, Perl is a cross-platform language, and it is available to download under the General Public Licence (GNU). Perl is available to download at the following website: www.perl.org. At the time of writing, the current stable version of Perl is the 5.24.1. The major strength points of Perl are the following: (i) Perl is easy to learn and use, as it was designed to be easy to understand especially for humans rather than for computers. (ii) Perl is portable in the sense that, it is possible to run a script wrote in Windows operating system on several different other operating systems without change any line of code. Perl is a partially interpreted language since existing a compilation step through that a Perl script goes through before its execution. Before to execute a Perl script, is mandatory to compile the script that is translated in bytecode and finally, Perl interprets the bytecode. In Perl the compilation includes many of the same optimization steps like in Java, C, and C++ program, that is the elimination of unreachable code, replacing of constant expressions by their values, linking library and the built-in functions, and so on. Another characteristic of Perl is that variables do not have an intrinsic type in the sense that, conversely from languages such as Java, C or C++, a variable can be declared without any particular type. In this way, a variable previously used to store an integer can next be used to contain String or Double. Moreover, a variable can also contain an undefined value, by assigning to it the special object *undef*. The *undef* keyword in Perl's is the equivalent of the *null* object in Object Oriented Languages. A faster way to obtain further information regarding the Perl language features is to consult Perl's online documentation, commonly referred to as *perldoc*.

Perl is still broadly used for its original purpose: working like mediator among different tools, making data coming from one software in a format compatible with the format expected by the other tool. Going from processing and summarizing system logs (Newville, 2001), through manipulating databases (Wishart *et al.*, 2007), reformatting text files (Letunic and Bork, 2011), and simple search-and-replace operations, as well as in comparative studies (Warren *et al.*, 2010), life sciences data analysis (Ravel, 2001), handling data from the Human Genome Project as reported in Stein *et al.* (2002), Stein (1996), managing bioinformatics data see work of Arakawa *et al.* (2003), Lim and Zhang (1999) and all the tasks that require massive amounts of data manipulation.

Background

Perl is a language primarily intended to be used from command-line interface, shell or terminal because it was developed as a server-side scripting language (Wall *et al.*, 1994; Hall and Schwartz, 1998; Ousterhout, 1998). To take advantages of all the power of Perl, programmers are to know how to deal with a terminal interface. A terminal usually is a black/white screen displaying the prompt that looks like: \$, %, C:\>. After the prompt symbol, there is a flashing underscore meaning that the terminal is ready to get command. By using the terminal, it is possible to consult the Perl documentation by typing the command *perldoc*. From a terminal window type “*perldoc-h*” (the -h option prints more help) as conveyed in Fig. 1.

The *perldoc* command allows programmers to access to all the information about a particular function, including the implementation code. To get information about the *rename* function, the programmer has to type in the terminal: “*perldoc-f rename*” allowing to see the description and code of the *rename* function. As well as *perldoc* allows programmers to search for all the question-answer entries in the Perl FAQs for which the questions contain a particular keyword, for example, looking for *perldoc-q substr*” in this case the command allow to obtain more detailed information about *substr* function. The terminal is also used to write and execute Perl programs. Before to run Perl programs it is necessary to write them, by using an editor. Each operating system, Unix, OS X, Windows, and Linux, comes with several different text editors. Thus each programmer is free to use its favorite editor.

Install Perl on Your Machine

Perl has been developed to be used on many platforms. It will almost certainly build and run any UNIX-like systems such as Linux, Solaris, FreeBSD. Most other current operating systems are supported: Windows, OS/2, Apple Mac OS, and so on. Programmers can get the source release and/or the Binary distributions of Perl at the following web address <https://www.perl.org/get.html>.

```

[os-x:~ mac$ perldoc -h
perldoc [options] PageName|ModuleName|ProgramName|URL...
perldoc [options] -f BuiltinFunction
perldoc [options] -q FAQRegex
perldoc [options] -v PerlVariable

Options:
  -h    Display this help message
  -V    Report version
  -r    Recursive search (slow)
  -i    Ignore case
  -t    Display pod using pod2text instead of Pod::Man and groff
        (-t is the default on win32 unless -n is specified)
  -u    Display unformatted pod text
  -m    Display module's file in its entirety
  -n    Specify replacement for groff
  -l    Display the module's file name
  -F    Arguments are file names, not modules
  -D    Verbosely describe what's going on
  -T    Send output to STDOUT without any pager
  -d output_filename_to_send_to
  -o output_format_name
  -M FormatterModuleNameToUse
  -w formatter_option:option_value
  -L translation_code    Choose doc translation (if any)
  -X    Use index if present (looks for pod.idx at /System/Library/Perl/5.18/darwin-thread-multi-2le
vel)
  -q    Search the text of questions (not answers) in perlfaq[1-9]
  -f    Search Perl built-in functions
  -v    Search predefined Perl variables

```

Fig. 1 Using terminal to display the Perl documentation by using perldoc command.

- **Install Perl on Windows:** Make sure you do not have any version of Perl already installed. If you do uninstall Perl be sure if you still have a folder in C:\Strawberry to delete it. Download the Strawberry Perl version 5.12.3 from <http://strawberryperl.com>. Reboot your machine, after go to your start menu, then click the “Perl command” link to verify that the installation worked type: `perl-v`.
- **Install Perl on Unix/Linux:** Install a compiler (if not yet installed on your machine), such as `gcc` through your system package management (e.g., `apt`, `yum`). Open a Terminal and copy and paste the command “`curl-L http://xrl.us/installperlunix | bash`” then press `return` key to confirm.
- **Install Perl on OSX:** First, install “Command Line Tools for Xcode,” directly through Xcode, or through the Apple Developer Downloads (free registration required). Xcode can also be installed through the App Store application. Launch the Terminal Applications, copy and paste the command “`curl-L http://xrl.us/installperlunix | bash`” then press `return` key to confirm.

Write and Execute Perl Program by Using Interactive Development Environment

Despite Perl was intended to be used from terminal command line, now are available several tools that make it possible to write and run Perl program by using Interactive Development Environment (IDE). In Windows there are a plenty of IDE editors the most used are *Notepad++*, *Padre* and *Notepad* (Plain Text Editor) the link to download are available at the following web address https://learn.perl.org/installing/windows_tools.html. In Unix/Linux the most used IDE editor are *vim*, *emacs* and *Padre* you can get more information where download these editor to the following web address https://learn.perl.org/installing/unix_linux_tools.html. In Mac OSX the most used IDE editor are *vim*, *emacs*, TextEdit (Plain Text Editor), TextMate (Commercial) and *Padre*. More information about where to download these editor can get to the following web address https://learn.perl.org/installing/osx_tools.html.

Finally, users can use the NetBeans IDE wrote in Java making NetBeans platform independent that is, can run on each operating system in which Java is installed. NetBeans let users to quickly and easily develop Java desktop, mobile, and web applications, as well as HTML5, JavaScript, PHP, and CSS. The IDE also provides an excellent set of tools for PHP and C/C++ including Perl. They are free and open source and has a large community of users and developers around the world. To use Perl in NetBeans IDE is necessary to download and install the *Perl On NetBeans – plugin* from the NetBeans Plugin Portal. The *Perl On NetBeans – plugin* requires the following steps to be installed on your system for the IDE to work correctly:

- Perl v5.12 or greater installed on your machine;
- Java Runtime Environment 7 or higher installed on your computer;
- NetBeans 8 or higher installed on your computer;
- The Perl and Java binaries (in Windows system) should be available in the PATH variable.

After you match the following system requirements, the installation of *Perl On NetBeans* can be summarized in the following steps:

- Download the *Perl On NetBeans* plugin from the web site "<http://plugins.netbeans.org/plugin/36183/perl-on-netbeans>".
- In NetBeans 8 select *Tool* from the menu bar and select *Plugins*, showing the *Plugins Manager* window;
- In the *Plugins Manager* window select the *Downloaded* tab and click on the button "*Add Plugins ...*" will show the file system navigation window, to locate the *Perl On NetBeans* file (download previously);
- Select the file to add to NetBeans 8.

Write and Execute a Perl Program by Using the Terminal

Assuming that Perl is installed on the machine (if not it is mandatory to install Perl by following the instruction provided in the Perl web site (www.perl.org)), the next step is to set up a directory for all the codes where to save all the Perl programs. A Perl program will look like as depicted in Fig. 2.

Let's look more in detail line by line the program presented in Fig. 2.

The first line "`#!/usr/bin/perl`" is called *interpreter directive* (also called "*shebang*", specifying to the operating system the suitable interpreter to execute the script. Perl treats all lines starting with `#` as a comment, ignoring it. However, the concatenation between "`#`" and "`!`" at the start of the first line tells the operating system that it is an executable file, and compatible with perl, which is located at the `/usr/bin/` directory. The second line "`use warnings;`" activates warnings. The activation of warnings is useful because recall to the interpreter to highlight to the programmers possible mistakes that otherwise will be not visualized. As an example, suppose that we made the following error (print "`Perl is $Awesome !!!`") in the line 3 of the code presented in Fig. 2. Commenting the "`use warning`" line we get the following result in output (let see Fig. 3), without having any clue on why in the output misses the string "`Awesome`".

Instead, enabling the warnings visualization the perl interpreter give in out-put the message conveyed in Fig. 4. The interpreter informs the programmer that is using an uninitialized variable called "`$Awesome`" (in Perl the variables are preceded by the `$` symbol).

Fundamentals

Perl is a programming language with which it is possible to guide the computer to solve problems. Problem-solving is related to data elaboration, to elaborate data it needs to use a language that can be understood by the machine. Programming

```
#!/usr/bin/perl

use warnings;

print "Perl is Awesome !!!\n";

"firstPerlProgram.pl" 5L, 63C
```

Fig. 2 A simple Perl program wrote by using vi editor.

```
[os-x:~ mac$ ./firstPerlProgram.pl ]
Perl is !!!
os-x:~ mac$
```

Fig. 3 The output when we warnings are disabled.

```
[os-x:~ mac$ ./firstPerlProgram.pl ]
Name "main::Awesome" used only once: possible typo at ./firstPerlProgram.pl line 5.
Use of uninitialized value $Awesome in concatenation (.) or string at ./firstPerlProgram.pl line 5.
Perl is !!!
os-x:~ mac$
```

Fig. 4 The output when warnings are enabled.

languages such as Perl provide problem-solving capabilities to the computer. Perl uses statements often grouped together into blocks, that are easy to write for humans as well as are easy to be understood by machines. A Perl statement tells the computer how to deal with data, ending with a semicolon ";". To gather together any number of statements it is necessary to surround statements by curly braces {...}, that in Perl is called *block*, here's an example: "{print "Hello Perl.\n"; print "That's a block."}", this statement prints on video the message Hello Perl and That's a block, on different rows (without quotes). Statements are not enough to elaborate data, because the machine needs to store data somewhere generally into the main memory, to deal with them when it is necessary during the whole elaboration process. The memory locations where program languages store data for simplicity are identified through variables. In Perl as well as in the other program languages, a variable is defined through a name. In particular, a variable name has to start with the symbols "\$", and should not be a keyword of Perl language. For example, "\$var" is a correct name for a variable instead, "\$do" is not proper as variable name because can be confused with Perl's *do* keyword. The \$ symbol before the name makes it possible to know that the \$var is a *scalar* specifying that the variable can store a single value at the time. Whereas, variables starting with the symbol "@" can contain multiple values and are called *array* or *list*. A scalar variable can contain numbers that in Perl are classified in integer and floating-point numbers. Integers are whole numbers such as: "5, -9, 78" (without decimal part). Instead, floating points numbers has a decimal part, for example: "0.1, -0.12344" and so on. To put data into a variable, programmers have to use the assignment operator "=". An *array* or *list* is a variable that may holds zero or more primitive values. The elements stored in arrays and lists are numbered starting with zero, ranging from zero to the number of elements minus one. [Code 3.1](#) conveys a simple example of using the array variable.

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
$@age=(25, 30, 44);
@names=("Joseph", "AnnaMary", "Bob");
$firstElem=$age[0];
print("$firstElem");
}
```

Code 3.1: An simple example to assign the first element of an array to a variable.

To modify the content of a variable is mandatory to use the assignment operator. For example, \$num=12345, we defined a variable called "num" and assigned to it the value "12345", the content of num will successively modify assigning new value \$num=1.3; for example. In addition to the numbers, Perl allows to variables to contain *strings*. A string is a series of characters surrounded by quotation marks such as "Hello World". Strings contain ASCII characters and escape sequences such as the \n of the example, and there is no limitation on the maximum number of characters composing a Perl string. Perl provides programmers mechanisms called 'escape sequences' as an alternative way of getting all the *UTF8* characters as well as the *ASCII* characters that are not on the keyboard. A short list of escape sequence is presented in [Table 1](#).

There is another type of string obtained by using single-quotes: ". The difference between single and double quotes is that no processing is done within single quoted strings, that is variable names inside double-quoted strings are replaced by their contents, whereas single-quoted strings treat them as ordinary text. To better explain the differences between single and double quotes consider [Code 3.2](#) as example:

Table 1 Escape characters

Escape sequence	Function
\t	Tab
\n	New line
\b	Backspace
\a	Alarm

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
print '\t This is a single quoted string.\n';
print " \t This is a double quoted string.\n";
}
```

Code 3.2: Difference between the use of single and double quotes.

The differences between the double-quoted and the single-quoted string are that: the first one has its escape sequences processed, and the second one not. The output obtained is depicted in [Code 3.3](#):

```
#Console Output
\t This is a single quoted string.\n
This is a double quoted string.
```

Code 3.3: The difference of output due to the use of single and double-quotes.

This operation is called escaping, or more commonly, *backwhacking* allows programmers to put special character such as backslash into a string as conveyed in [Code 3.4](#), printing it on the screen in the correct format, let see [Code 3.5](#).

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
print 'C:\WINNT\Profiles\'.'.\n';
print "C:\\WINNT\\Profiles\\.\n";
}
```

Code 3.4: Combination of escaping character to print in output the special character \.

```
#Console Output
C:\WINNT\Profiles\
C:\\WINNT\\Profiles\\
```

Code 3.5: Output result using single and double-quotes respectively.

Perl besides to allow users to define numbers and strings, it provides operators and functions to deal with numbers and strings.

Arithmetic Operators

The *arithmetic operators* comprise the basic mathematics operators like adding, subtracting, multiplying, dividing, exponentiation and so on. As appear for mathematics each operator comes with a precedence, which establishes the order in which Perl performs operations. Multiply and divide have a higher precedence than adding and subtract, and so they get performed first. To coerce Perl to perform operations with low priority first, it is mandatory to use brackets, for example, the following operation $1 + 2 * 3$ will

produce as a result 7. To obtain 9, as a result, it is necessary to rewrite the expression by using brackets in this way $(1 + 2) * 3$. Other arithmetics operators are *exponentiation operator* `**` and *module* `%`. Where module operator has the same precedence as multiple and divide, whereas exponentiation operator has higher precedence than multiple and divide, but lower precedence than minus operator.

Bitwise Operators

Bitwise operators work on bits since computer represent the complete information using bits. Bitwise operators perform bit by bit operations, from right to left. Where the rightmost bit is called the 'least significant bit,' and the leftmost is called the 'most significant bit'. Given two numbers 9 and 5 that in binary using 4 bits are expressed as $9 = 1001$, and $5 = 0101$ let see, which are the bitwise operators available in Perl. Bitwise operators include: the *and* operator and is written `&` in Perl. The `&` operator compares pairs of bits, as follows: if bits are both 1 the `&` gives 1 as a result, otherwise if one of the bits or both is equal to 0 the `&` gives 0 as a result. For example, the result of $9 \& 5$ is: $\$a \& \$b = 0001$. The *or* operator in Perl is `|`, where 0|0 is always 0 whereas, 1|1 and 1|0 is always 1 (independently of the left-right operator values). The result of $9|5 = 1101$. To know if one bit or both bits are equals to 1 it is possible to use the *exclusive-or* operator `^`, the result of $5 \wedge 9$ is 1100. Finally, by using the *not operator* `~` it is possible to replace the value from 1 to 0 and vice versa, for example, ~ 5 is 1010.

Equality Operators

Perl provides users operators able to compare equality of numbers and strings.

Comparing numbers

The equality operator `==` checks if the value of two numerical operands are equal or not, if are equal the condition gives *true* as result *false* otherwise. In Perl *true* is represented as 1 and *false* as 0. The Inequality operator, `!=`, verifies if two operands are different, if left value and right value are different the condition becomes true ($5 \neq 9$ gives as result *true*). The compare operator `<=>` checks if two operands are equal or not providing as result `-1`, `0` or `1` if the left operand is numerically less than, equal to, or greater than the second operand. Finally, the operators `<`, `<=`, `>` and `>=`. The `<` operator give *true* if the left operand is less than the right operand (e.g., $5 < 9$ gives as a result *true*). The `<=` operator gives *true* if the left operand is less or equal than the right operand (e.g., $5 <= 5$ gives as result *true*). The `>` operator give *true* if the left operand is greater than the right operand (e.g., $5 > 9$ gives as a result *false*). Finally, The `>=` operator give *true* if the left operand is greater or equal than the right operand (e.g., $9 >= 5$ gives as a result *true*).

Comparing strings

To compare two strings in Perl is necessary to use the comparison operators `cmp`. `cmp` compares the strings alphabetically. `cmp` returns `-1`, `0`, or `1` depending on whether the left argument is less, equal to, or greater than the right argument. For example, "Bravo" comes after "Add", thus `("Bravo" cmp "Add")` gives as result `-1`. To test whether one string is less than another, use `lt`. Greater than becomes `gt`, equal to is `eq`, and not equal becomes `ne`. There are also the operators *greater than or equal* to referred to as `ge` and *less than or equal* to referred to as `le`.

Logical operators

Logical operators make possible to evaluate the truth or falsehood of some statements at the time. The logical operators supported by Perl are *and* referred to as `&&`. The `&&` operator evaluates the condition and if both the operands are *true* returns *true* as result, *false* otherwise. The *or* referred to as `||` evaluates the condition, returning *true* if at least one of the operands is *true*, *false* otherwise. Not operand `!` is used to negate the logical state of the condition. As an alternative it is possible to use logical operators through the easier to read versions, *and*, *or*, and *not*.

Other operators

Other useful operators available in Perl are: string concatenation `.` given two string `$a = "abc"` and `$b = "def"` `$a.$b` gives as result `"abcdef"`. Repetition operator `x` gives in output the left operand repeated *x*-times, for example `(print "ciao"x3)`, will print `"ciaociaociao"`. Range operator `..`, return a list of value starting from the left value to the right value included. For example `(2..6)`, will return the following values `(2,3,4,5,6)`. Finally, the auto-increment `++` and auto-decrement `--` operator, that increases and decreases integer value by one respectively.

Conditional Statement

The *if-else* statement is the fundamental control statement that allows Perl to make decisions executing statements conditionally. The simplest conditional statement in Perl has the form:

```
if(<condition>){<Statement 1>;<Statement 2>;...};
```

The if-else statement has an associated expression and statement. If the expression evaluates to true, the interpreter executes the statement. If the expression evaluates to false the interpreter skips the statement. An example of using if statement is presented in [Code 3.6](#).

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
if ($a >= $b){
    print ( 'The values of $a and $b is : ' . ($a) );
}
```

Code 3.6: An example illustrating a basic use of the if statement.

An if statement can include an optional `else` keyword. In this form of the statement, the expression is evaluated, and, if it is true, the first statement is executed. Otherwise, the second statement (`else`) is executed. The more general conditional in Perl presents the form:

```
if(<condition>){StatementsBlock1}; else{StatementsBlock2};
```

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
if ($a>$b){
    print ( '$a is greater than $b' . ($a->$b) );
} else {
    print ($b is greater than $a' . ($b->$a) );
}
```

Code 3.7: An example of if else statement.

When you use nested if/else statements, some caution is required to ensure that the else clause goes with the appropriate if statement. Nesting more than one condition could be difficult to read. Thus Perl provides to programmers the if `elsif` statement, which presents an easier to read form:

```
if ( < condition1 > ) < action > elsif ( < condition2 > ) < secondaction >
    elsif ( < condition3 > ) ... else <if all elsif fails>
```

Code 3.8: An example of if elsif statement:

Loops

The loops statement are the basic statement that allows Perl to perform repetitive actions. Perl programming language provides specific types of loop.

while loop that only executes the statement or group of statements only if the given condition is *true*. The general form of *while* loop is:

```
while (<condition>){<block of statements>}
```

Code 3.9 illustrates the of while loop.

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
use strict;
#initialisation of $a to 0
$a = 0;
# while loop execution
while($a<=9){
    printf "Value_of_a: %d\n";
    $a = $a + 1;
}
```

Code 3.9: An example of while loop that prints the numbers from 0 to 9.

until loop execute a statement or a group of statements till the given condition not becomes true. The general form of until loop is:

```
until (<condition>){<block of statements>}
```

Code 3.10 presents a simple use case of until loop.

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
use strict;
#initialisation of $a to 0
$a = 0;
# until loop execution
until($a==10){
    printf "Value_of_a: %d\n";
    $a++;
}
```

Code 3.10: An example of until loop that prints the numbers from 0 to 9.

do loop is very similar to the while loop, except that the loop expression is tested at the bottom of the loop rather than at the top. do ensures that the body of the loop is executed at least once. The syntax of do loop looks like:

```
do{<block of statements>}while(<condition>)
```

The for loop executes the statements in a block a determinate number of times. The for in is more general form is:

```
for (init; condition; increment){Block of statements; }
```

The `for` iterates on each element in an array as well as in a list as conveyed. As [Code 3.11](#) shows the use of `for` loop to print the elements of an array.

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
use strict;
my @array = (1,2,4,3);
my $i;
# for loop
for $i (@array){
    printf "\n_$i";
}
```

Code 3.11: An example of `for` loop that prints the values of an array.

It is possible to use `foreach` and `for` loop indistinctly on any type of list. It is worthy to note that the both loops create an alias, rather than a value. Thus, any changes made to the iterator variable, whether it be `$` or one you supply, will be reflected in the original array. For instance:

```
#!/usr/bin/perl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM
use warnings;
use strict;
# create an array of ten elements
my @array = (1..10);
foreach (@array) {$_++;}
print "Array is now: _@array\n";
```

Code 3.12: `foreach` loop that prints and modify the values of the array:

The code in [Example 3.12](#) will change the contents of the array, as follows:

Array is now: 2 3 4 5 6 7 8 9 10 11

Fibonacci Sequence Example

In this section, will be introduced the algorithm wrote in Perl to compute the Fibonacci sequence. The Fibonacci sequence is a recursive formulation where each element is equal to the sum of the first two. This sequence owes its name to the Italian mathematician Fibonacci. The purpose of the sequence was to identify a mathematical law to describe the growth of a population of rabbits. The [Code 3.13](#) shows the computation of the Fibonacci's sequence through Perl programming language.


```
#!/usr/bin/perl
# @File Fibonacci.pl
# @Author Perl
# @Created Mar 29, 2017 3:30:48 PM

use strict;
use warnings;

my $f1 = 0;
my $f2 = 1;

printf "\nHow many numbers of the Fibonacci's sequence
would you display?\n";
#We use <STDIN> library to get the data from stdin here
my $n = <STDIN>;
for (my $i=0;$i<$n;$i++){
    printf "%d\n", $f1;
    my $sum = $f1 + $f2;
    $f1 = $f2;
    $f2 = $sum;
}
```

Code 3.13: A simple script example to compute the Fibonacci's Sequence.

Closing Remarks

Perl first appeared in 1987 as a scripting language for system administration, but thanks to its very active community became a very powerful, flexibility and versatility programming language. The main strength of Perl's popularity is the Comprehensive Perl Archive Network (CPAN) library, that is very extensive and exhaustive collection of open source Perl code, ranging from Oracle to iPod to CSV and Excel file reader as well as thousands of pages of Perl's core documentation. Thus, Perl exploiting the knowledge and experience of the global Perl community, it provides help to everyone to write code, bug resolution and code maintenance. In summary, the key points of Perl are (i) management of Regular Expressions are natively handled through the regular expression engine. Regular expression engine is a built-in text process that, interpreting patterns and applying them to match or modify text, without requiring any additional module. (ii) The flexibility, Perl provides programmers only three basic variable types: Scalars, Arrays, and Hashes. That's it. Perl independently figures it out what kind of data developers are using (int, byte, string) avoiding memory leaks. Finally, (iii) the portability. Perl works well on several operating systems such as UNIX, Windows, Linux OSX, as well as on the web.

See also: Computing for Bioinformatics

References

- Arakawa, K., Mori, K., Ikeda, K., *et al.*, 2003. G-language genome analysis environment: A workbench for nucleotide sequence data mining. *Bioinformatics* 19 (2), 305–306.
- Hall, J.N., Schwartz, R.L., 1998. *Effective Perl Programming: Writing Better Programs With Perl*. Addison-Wesley Longman Publishing Co., Inc.
- Letunic, I., Bork, P., 2011. Interactive tree of life v2: Online annotation and display of phylogenetic trees made easy. *Nucleic Acids Research*. gkr201.
- Lim, A., Zhang, L., 1999. Webphyliip: A web interface to phylip. *Bioinformatics* 15 (12), 1068–1069.
- Newville, M., 2001. lfeffit: Interactive xafs analysis and feff fitting. *Journal of Synchrotron Radiation* 8 (2), 322–324.
- Ousterhout, J.K., 1998. Scripting: Higher level programming for the 21st century. *Computer* 31 (3), 23–30.
- Ravel, B., 2001. Atoms: Crystallography for the x-ray absorption spectroscopist. *Journal of Synchrotron Radiation* 8 (2), 314–316.
- Stein, L., 1996. How perl saved the human genome project. *Dr Dobb's Journal* (July 2001).
- Stein, L.D., Mungall, C., Shu, S., *et al.*, 2002. The generic genome browser: A building block for a model organism system database. *Genome Research* 12 (10), 1599–1610.
- Wall, L., Christiansen, T., Orwant, J., 2000. *Programming Perl*. O'Reilly Media, Inc.
- Wall, L., *et al.*, 1994. *The perl programming language*.
- Warren, D.L., Glor, R.E., Turelli, M., 2010. Enmtools: A toolbox for comparative studies of environmental niche models. *Ecography* 33 (3), 607–611.
- Wishart, D.S., Tzur, D., Knox, C., *et al.*, 2007. Hmdb: The human metabolome database. *Nucleic Acids Research* 35 (suppl 1), D521–D526.

Computing Languages for Bioinformatics: BioPerl

Giuseppe Agapito, University Magna Græcia of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Computational analysis is becoming an essential part of the modern biological research, to speed up the data analysis. This change is due to the high throughput methodologies (Abu-Jamous *et al.*, 2015) that are producing an enormous amount of data for each single experiment, requiring techniques able to analyze as much as possible information in the short possible time avoiding that become useless (Marx, 2013). The high-throughput methods comprise Protein Chips, Mass Spectrometry (for identification and quantification (LC-MS)), Yeast Two-Hybrid (that are techniques for investigating physical interactions among proteins), and Surface Plasmon Resonance technologies for studying Kinetic Dynamics in proteins. As well as, high-throughput sequencing includes Next Generation Sequencing (NGS), RNA sequencing, lipid profiling by GC-MS. Many computer software tools exist to perform biological data analyses, requiring the integration of several disciplines such as biology, physics, mathematics, computers science, statistics, engineering known as bioinformatics. These huge amount of data require a lot effort from computer scientists to face from several different points of view, from data storage (Barrett *et al.*, 2009) perspective where, there are a lot of public databases such as GenBank (Benson *et al.*, 2013), Protein Data Bank (PDB) (Bank, 1971), IID (Kotlyar *et al.*, 2015) Reactome (Joshi-Tope *et al.*, 2005), Panther (Mi *et al.*, 2005) and so on, that have been growing exponentially, in the last decade. As well as, computer scientist have to develop efficient tools and algorithm able to deal with this huge amount of data in the short possible time providing life science researchers practical tools able to simplifying their work (Guzzi *et al.*, 2014; Milano *et al.*, 2016; Agapito *et al.*, 2013a). Flanked to the data analysis, there is data visualization that makes it possible to highlight to the researcher's information not visible otherwise, especially reading strings or worst numbers (Pastrello *et al.*, 2013; Agapito *et al.*, 2013b). As a result, computer-science now play a more crucial role in the advancement into the life science research.

An example to demonstrate the power of bioinformatics to analyze a biological problem simple or complex. Let suppose that a research group discovers a fascinating segment of monkey's DNA from which could be possible extract some clue to explain the development of malignant brain neoplasms in humans. After DNA sequencing, researchers have to perform a search in public and private sequence databases and other data sources by using sequence alignment tools (i.e., BLAST) to get some match with known sequences. Although researcher will find some related sequences, this does not means that exist a link that could explain how brain neoplasms develop in human. To get this information, it is necessary to quotidianly query databases, task that could take several hours, days or worse months. Fortunately, bioinformatics propose several software tools, libraries that simplify the writing of a program, that in the previous example automatically conducts a daily BLAST search of databases for match with the new DNA sequence.

To help life science scientists several initiatives were born to simplify the analysis of life science area, including the BioPerl library (Stajich *et al.*, 2002).

The BioPerl is an international project involving several users and developers in the overall world of open source Perl tools for bioinformatics, genomics, and life science. BioPerl project is managed by the *Open Bioinformatics Foundation* (OBF) a non-profit, volunteer-run group devoted to promoting the Open Source software development and Open Science within the biological research community for more information visit the <http://BioPerl.org/index.html> web-site. The OBF foundation as well as BioPerl manages other open source projects including *BioJava*, *Biopython*, *BioRuby*, *BioSQL*, *DAS* (and related list of Global Sequence Identifiers), *MOBY*, *EMBOSS* and *OBDA*.

At the time of writing on CPAN at the following web-site <https://metacpan.org/release/BioPerl> and Github <https://github.com/BioPerl/BioPerl-live/releases/tag/release-1-7-0> is available the release of BioPerl v1.7.0, to reduce the number of dependencies required during the installation process as well as reducing the overhead maintenance process. In short, Bioperl is a set of Perl packages that promote the development of Perl scripts for bioinformatics applications.

BioPerl Installation

The BioPerl modules are distributed as a tar file that expands into a standard perl CPAN distribution. To install BioPerl on your Unix, Linux or MacOSX computer follow the instructions below.

Installing BioPerl on Linux and Mac OSX Machines

To install BioPerl on Linux, is preferred to use the repository, since many Linux distributions have already packaged BioPerl. Installing BioPerl from repository should be preferred since avoid to install out of date versions. Before proceeding with the

installation make sure that on your system is installed the release of BioPerl version 5.8 or higher, and make command (for the Mac OS system the make is not installed, and requires to install Xcode Developer Tools). BioPerl could be installed through CPAN or Github. The installation steps to install BioPerl from CPAN are the following: To test if on your system is already installed *cpan* alias, write from the command line the *cpan* command and press *enter*. If *cpan* is installed on your screen should be appear the following message: CPAN.pm requires configuration, but most of it can be done automatically. If you answer 'no'

below, you will enter an interactive dialog for each configuration option instead. Would you like to configure as much as possible automatically?

[yes]. After *cpan* is installed, it is necessary to find the latest BioPerl package entering the following command from the command line prompt: i) *cpan* and *enter*; ii) from the *cpan* terminal enter:

```
cpan> d /bioperl/ and enter; iii) will be displayed all the available version of BioPerl:
```

```
Distribution CDRAUG/Dist-Zilla-PluginBundle-BioPerl-0.25.tar.gz
```

```
Distribution CJFIELDS/BioPerl-1.007001.tar.gz
```

```
Distribution CJFIELDS/BioPerl-1.6.924.tar.gz
```

```
Distribution CJFIELDS/BioPerl-Network-1.006902.tar.gz
```

Finally, iv) install the most recent release in this case: *cpan> install CJFIELDS/BioPerl-1.007001.tar.gz*. Otherwise, to install BioPerl from Github the steps are the following: By the command line type the following command and press *enter*:

git clone https://github.com/bioperl/bioperl-live.git. After go to the folder *bioperl-live* through the command: *cd bioperl-live* and press *enter*.

More detailed and updated information about the installation can be retrieved at the following web address: <http://bioperl.org/INSTALL.html>.

Installing BioPerl on Windows Machines

In this section, we will illustrate how to Install Perl on Windows machines by using CPAN for Strawberry. Installing Bioperl from repository should be preferred since avoid to install out of date versions. Before proceeding with the installation make sure that on your computer is installed the release of BioPerl version 5.8 or higher, the accessory compiling program MinGW. MinGW provides tools such as *dmake* and *gcc*. MinGW has to be installed through the Perl Package Manager Index (PPM) available from the ActiveState PPM repository. In a command line window type: *C:\>ppm install MinGW* Be sure to choose the version of MinGW compatible with your ActivePerl version, because is ActivePerl is compatible only with a specific release of MinGW. To run CPAN shell into a command window, type the *cpan* command and press *enter*. As a result, the CPAN prompt will be displayed. At the *cpan>* prompt, typing the command "*cpan> install CPAN*" that will upgrade CPAN to the latest release. At the *cpan>* prompt, type *cpan> o conf prefer installer MB* to force CPAN to use Build.PL scripts for installation, and typing *cpan> o conf commit* to save the modify. Always from the CPAN prompt, type *cpan>install Module::Build* and press *enter*, *cpan>install Test::Harness* press *enter*, *cpan>install Test::Most*. Now it is possible to install BioPerl.

First install *local::lib* module by using the following command *perl -MCPAN -Mlocal::lib -e 'CPAN::install(LWP)'* more detailed information are available at <https://metacpan.org/pod/local::lib> web site. At the end of *local::lib* installation on your own machine, it is possible to install BioPerl using the following command: *perl -MCPAN -Mlocal::lib -e 'CPAN::install(LWP)'*.

More detailed and updated information about the installation can be retrieved at the following web address: <http://bioperl.org/INSTALL.WIN.html>.

Objects Management in Perl and BioPerl

Object-Oriented Programming (OOP for short) is the new paradigm that has supplanted the "structured," procedural programming techniques. BioPerl is entirely developed in Perl consequently it supports object-oriented programming as well as, procedural programming. OOP springs from the real world objects vision, extending it to the computer programs. In OOP a program is made of objects, each one with particular properties and operations that the objects can perform. In OOP, programmers have to only care about what the objects expose (the public methods). So, just as a television manufacturers don't care about the internals components of a power supply, most OOP programmers do not bother how an object is implemented. Structured programs consist of a set of procedures (or methods) to solve a problem. The first step regards the developing of the procedures, whereas only at the last step it is required to find appropriate ways to store the data. Conversely, OOP reverses the order putting data first and then looks at the methodologies that operate on the data. The key point in OOP is to make each object responsible for carrying out a set of related tasks. If an object relies on a job that isn't its responsibility, it needs to have access to another object whose responsibilities include that function. Information hiding in OOP is known as data encapsulation. Data encapsulation maximize reusability, reducing dependencies, and minimize the debugging time. Further, as good practice of OOP programming, an object should never directly manipulate the internal data of other objects, as well as should not it expose data for other objects

to access directly. An object or class in Perl and consequently in BioPerl is simple a package with subroutines that function as methods, modifying the object's state. In particular object's methods are the behaviour of an object (can a programmer can apply to it), and internal data (the fields) are the object's state this is, how the object reacts when those methods are applied to it. An example of class in BioPerl is presented in [Code 3.1](#).

```
#!/usr/bin/perl
# @Author Perl
package Lion;
sub roar {
    ....
}
sub manedColor {
    ....
}
```

Code 3.1: A simple example of Class in BioPerl.

To initiates an object from a class, is it mandatory to use the new method, and to use the methods provided by an object, has to be used the "`->`" operator, as shown in [Code 3.2](#).

```
#!/usr/bin/perl
# @Author Perl
...
$simba = new Lion;
...
print $simba->roar();
...
print "Simba's maned_color is : ". $simba->manedColor();
...
```

Code 3.2: A simple example of Object instantiation in BioPerl.

A Perl module is a reusable package enclosed into a library of files, whose name is the same of the package name. Each Perl module has a unique name. Moreover, Perl provides a hierarchical namespace for modules, to minimize namespace collision. Components of a module name are separated by double colons (`::`), for example, `Math::Complex`, `Math::Approx`, `String::BitCount` and `String::Approx`. Each module is contained in a single file, and all module files have `.pm` as an extension. To have a hierarchy in the name of the modules, Perl allow to stores files in subdirectories. A module can be loaded into any script by calling the `use` function, as conveyed from the [Code 3.3](#).

```
#!/usr/bin/perl
# @Author Perl

use Math::Trig;

$x = tan(0.9);
$y = acos(3.7);
$z = asin(2.4);

$rad = deg2rad(120);
# Import constants pi
use Math::Trig 'pi';
```

Code 3.3: A simple example of Module loading through the command use in BioPerl.

BioPerl Modules

BioPerl provide software modules for several of the typical activities to the analysis of life science data, including:

Sequence Objects

Bio::Seq is the main sequence object in BioPerl. A Seq object is a sequence and it contains a Bio::PrimarySeq object and annotations associated to the sequence. Seq objects can be created explicitly when needed through the new command, or implicitly by reading file containing sequences data by using the Bio::SeqIO object. PrimarySeq is the essential version of Seq object. PrimarySeq contains merely the sequence data itself and some identifiers (id, accession number, molecule type = DNA, RNA, or protein). Using PrimarySeq object can significantly speed-up the program execution and decrease the amount of central memory that the program requires to handle large sequences. Biological sequence's starting and ending points are represented in BioPerl by the Bio::LocatableSeq that is a Bio::PrimarySeq object able to represent the start and end points of any sequence, loaded manually. To handle very large sequences (i.e., greater than 100 MBases) BioPerl defines the LargeSeq object that is a special object of Seq type, with the capability to store very large sequences into the file system avoiding out of memory.

Alignment Objects

Conversely for the sequence handling, BioPerl provide only two modules to align sequences that are Bio::SimpleAlign and Bio::UnivAln. Multiple sequence alignment (MSA) in BioPerl are managed by the Bio::SimpleAlign object. The SimpleAlign objects allow aligning sequence with different length, providing a set of built-in manipulations and methods for reading and writing alignments. In the more recent versions of BioPerl, the use of Bio::UnivAln is not recommended. As good programming practice, the alignments have to be generally handled by using the SimpleAlign module where it is possible.

Illustrative Examples

In this section, will be presented some use case of BioPerl modules to accomplish any standard bioinformatics' task, by means of simple example codes. A way to manipulate sequences with BioPerl can be accomplished by using the Bio::Seq module. By default, the Bio::Seq module contains a sequence and a set of sequence features aggregate with its annotations. [Code 5.1](#) is a simple complete BioPerl script that demonstrate how to directly assign a sequence and related additional information to a BioPerl object.


```
#!/usr/bin/perl
# @Author Perl

use Bio::Seq;

$seq = Bio::Seq->new(-seq => 'agtagtagcagtagagct',
                    -desc => 'Simple_Bio::Seq_object',
                    -display_id => 'sequence',
                    -accession_number$
                    -alphabet => 'dna' );

print($seq->seq()."\n");
```

Code 5.1: The script is an example that demonstrates how to use Seq object to manually create a sequence and handling with the sequence, through the available methods.

Let's examine in details the script presented in [Code 5.1](#). The first line `#!/usr/bin/perl` is called interpreter directive (also called "shebang"), specifying to the operating system the suitable interpreter to execute the script. Second line contains a comment, whereas line 4 import the *Seq Bioperl* module. To create and initialize a Seq object it is mandatory to use `new` command (let see row 5 in [Code 5.1](#)). The `$seq` object is manually initialized by using the following commands: `seq` that sets the sequence to be handled, `-desc` that sets the description of the sequence, `display id` sets the *display id*, also known as the name of the Seq object to visualize, `accession number` sets the unique biological id for a sequence, commonly called the accession number. For sequences from established databases, should be used the accession number provided by the database curator. Alphabet the possible values is one of *dna*, *rna*, or *protein*, if the value is not provided will be automatically inferred.

However, in most cases, it is more indicate to access sequence data from some online data databases. BioPerl allowing users to access remote database among which: *Genbank*, *Genpept*, *RefSeq*, *Swissprot*, *EMBL*, *Ace*, *SWALL*, *Reactome* and so on. Accessing sequence data from the principal biological databases is straightforward in BioPerl. Data can be obtained employing the sequence's accession code (i.e., "`seq2 = gb -> get Seq by acc('AH012836');`" or open a stream on multiple sequences (i.e., "`seqio = gb -> get Stream by id(['AH012836','BA000005','AH011052']);`"). GenBank data can be retrieved by using the code proposed in [Code 5.2](#) for example.

```
#!/usr/bin/perl

$ENV{'PERLLWP_SSL_VERIFY_HOSTNAME'} = 0;

use Bio::DB::GenBank;

$gb = new Bio::DB::GenBank();
# this returns a Seq object from genbank:
$seq = $gb->get_Seq_by_id('AH012836');
print($seq->display_id()."\n". $seq->length()."\n");
print($seq->seq()."\n");
```

Code 5.2: Shows the code to get data of a particular sequence (AH012836 is the accession code of *Sus scrofa* organism) from GenBank that will be used to initialize an Seq object.

Let's examine in details the script presented in [Code 5.2](#). The first line `#!/usr/bin/perl` is called interpreter directive (also called "shebang"), specifying to the operating system the suitable interpreter to execute the script. The second line contains the definition of an environment variable. The definition of this variable is mandatory because, otherwise perl throws an exception like: "Can't verify SSL peers without knowing which Certificate Authorities to trust".

To disable verification of SSL peers set the PERL LWP SSL VERIFY HOSTNAME environment variable to 0. Line 5 create a new object suitable to contain data from GenBank, through the `new` command. Line 6 contains a comment, line 7 is the code used to

retrieve data from genbank, related to the organism identified by means of the *AH012836* value. Finally, lines 8, 9 print on screen the sequence identifier, the sequence length and the sequence as a string of letters. In addition to the methods directly available in the Seq object, BioPerl provides the Bio::Tools::SeqStats objects to compute simple statistical and numerical properties of the primary sequence. [Code 5.3](#) presents an case use of the Bio::Tools::SeqStat object, to compute the molecular weight of a sequence as well as the counts of each type of monomer (i.e., amino or nucleic acid).

```
#!/usr/bin/perl
$ENV{'PERLLWP_SSL_VERIFY_HOSTNAME'} = 0;
use Bio::DB::GenBank;
use Bio::Tools::SeqStats;

$gb = new Bio::DB::GenBank();
# this returns a Seq object :
$seq = $gb->get_Seq_by_id('AH012836');#('MUSIGHBA1');
print($seq->display_id()."\n".$seq->length."\n");
print($seq->seq()."\n");
$translation = $seq->translate;
#print($translation);

$weight = Bio::Tools::SeqStats->get_mol_wt($seq);
print "\nMolecular_weight_of_the_primary_sequence",
      $seq->id()."_is_between". $$weight[0]."_and".
      $$weight[1]."\n";

$hash_ref = Bio::Tools::SeqStats->count_monomers($seq);
foreach $base (sort keys %$hash_ref) {
    print "Number_of_bases_of_type". $base. "=".
          %$hash_ref->{$base}."\n";
}
```

Code 5.3: Shows the code to compute some analysis and statistic on the primary sequence.

Let's analyze in details the script presented in [Code 5.3](#). Line 14 declare and initialize the *weight* variable to contain a reference to the *SeqStats* object.

from the *seq* variable. In line 15 print the molecular weight of the primary sequence referred from the *seq* variable. Since the sequence may contain ambiguous monomers, the molecular weight is returned as range. Line 19 define an hash containing the count of each type of amino acid contained in the *seq* sequence. The cycle in line 20 allow to scan all the hash to get each amino acid and print its own occurrence in the sequence. Another common difficult and error prone bioinformatics task is manually converting sequence data among the several available data format. BioPerl Bio::SeqIO object can read as input several different file formats among which: Fasta, EMBL, GenBank, Swissprot, PIR, GCG, SCF, phd/phred, Ace, fastq, exp, or raw (plain sequence), and converting in another format and written to another file. The [Code 5.4](#) shows, how read a file from a directory on your computer in a format and convert it in another format.

```
#!/usr/bin/perl
use Bio::SeqIO;
$input = Bio::SeqIO->new(-file => "./GLA_prot.fasta",
                        -format => 'Fasta');
$output = Bio::SeqIO->new(-file => ">pfamFromat.pfam",
                        -format => 'EMBL');
while(my $seq=$input->next_seq()){
    $output->write_seq($seq);
}
print ("Conversion_job_ended!!!\n");
```

Code 5.4: Shows the how to convert an input file in FASTA format in EMBL format.

Analyzing in detail [Code 5.4](#), the statement in lines 3 and 4 allow to define the location on the disk where is stored the fasta file (the input file to convert) in particular, the parameter `-f => "name input file"` allow to the perl interpreter to load the file, whereas the parameter `-format => 'Fasta'` allow to specify to the interpreter the format of the input file. The statement wrote in lines 5 and 6 define the name and the location on the disk where to write the converted file `-file => ">pfamFromat.pfam"` and the new format `-format => 'EMBL'`. Block statement in line 7 allow to fetch a sequence at time (`$seq=$input->next_seq()`) until there are some, writing in output through the statement `"$output->write_seq($seq);"`.

See also: Computing for Bioinformatics. Computing Languages for Bioinformatics: Perl

References

- Abu-Jamous, B., Fa, R., Nandi, A.K., 2015. High-throughput technologies. In: Integrative Cluster Analysis in Bioinformatics.
- Agapito, G., Cannataro, M., Guzzi, *et al.*, 2013a. Cloud4snp: Distributed analysis of snp microarray data on the cloud. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. ACM, p. 468.
- Agapito, G., Guzzi, P.H., Cannataro, M., 2013b. Visualization of protein interaction networks: Problems and solutions. BMC Bioinform. 14 (1), S1.
- Bank, P.D., 1971. Protein data bank. Nat. New Biol. 233, 223.
- Barrett, T., Troup, D.B., Wilhite, S.E., *et al.*, 2009. NCBI GEO: Archive for high-throughput functional genomic data. Nucleic Acids Res. 37 (suppl 1), D885–D890.
- Benson, D.A., Cavanaugh, M., Clark, K., *et al.*, 2013. Genbank. Nucleic Acids Res. 41 (D1), D36–D42.
- Guzzi, P.H., Agapito, G., Cannataro, M., 2014. coreSNP: Parallel processing of microarray data. IEEE Trans. Comput. 63 (12), 2961–2974.
- Joshi-Tope, G., Gillespie, M., Vastrik, I., *et al.*, 2005. Reactome: A knowledgebase of biological pathways. Nucleic Acids Res. 33 (Suppl. 1), D428–D432.
- Kotlyar, M., Pastrello, C., Sheahan, N., Jurisica, I., 2015. Integrated interactions database: Tissue-specific view of the human and model organism interactomes. Nucleic Acids Res. gkv1115.
- Marx, V., 2013. Biology: The big challenges of big data. Nature 498 (7453), 255–260.
- Mi, H., Lazareva-Ulitsky, B., Loo, R., *et al.*, 2005. The panther database of protein families, subfamilies, functions and pathways. Nucleic Acids Res. 33 (Suppl. 1), D284–D288.
- Milano, M., Cannataro, M., Guzzi, P.H., 2016. Galign: Using global graph alignment to improve local graph alignment. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, pp. 1695–1702.
- Pastrello, C., Otasek, D., Fortney, K., *et al.*, 2013. Visual data mining of biological networks: One size does not fit all. PLOS Comput. Biol. 9 (1), e1002833.
- Stajich, J.E., Block, D., Boulez, K., *et al.*, 2002. The bioperl toolkit: Perl modules for the life sciences. Genome Res. 12 (10), 1611–1618.

Relevant Websites

<http://BioPerl.org/index.html>
 BioPerl.
<http://bioperl.org/INSTALL.html>
 BioPerl.
<http://bioperl.org/INSTALL.WIN.html>
 BioPerl.
<https://github.com/bioperl/bioperl-live.git>
 GitHub.

<https://github.com/BioPerl/BioPerl-live/releases/tag/release-1-7-0>

GitHub.

<https://metacpan.org/pod/local::lib>

Metacpan.

<https://metacpan.org/release/BioPerl>

Metacpan.

Computing Languages for Bioinformatics: Python

Pietro H Guzzi, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A concise description of Python is given by the Zen of Python that is available by typing `import this` on the Python console (Van Rossum *et al.*, 2007).

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one – and preferably only one – obvious way to do it.
Although that way may not be obvious at first unless you're Dutch. Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea – let's do more of those!
```

As a consequence Python is a highly readable language avoiding, for instance, the use of curly brackets to delimit blocks and the optional use of semicolons after statements. Blocks in Python are delimited by white space indentation: the increase in indentation delimited the start of statements while the decrease signifies the end of the current block (off-side rule). Moreover colons (:) are used to signal the start of the statement. For instance, following code shows the indentation used to represent the code of a sub-procedure (Zelle, 2004).

```
def hello():
    print('Hello World')
hello()
```

The simple program contains the definition of the subprocedure `hello()` and the subsequent invocation of the subprocedure.

Main Statements of Python

Python statements are similar to other programming languages and includes (Cai *et al.*, 2005):

- The if statement, which conditionally executes a block of code, along with else and elif (that stands for else-if);
- The while statement which conditionally executes a cycle;
- The for statement which executes a loop (using both iterators (for x in set) and indices (for x in range)).
- The try statement used to present code blocks that may raise exceptions to be caught and handled by except clauses; differently to other languages it also ensures that a finally block will always be run;
- The class statement used to define a class in object-oriented programming;
- The def statement, which defines a function or method;
- The import statement used to import external modules in a program.

Data Structures in Python

Python implements main data types such as int for integers, long for long integers, float for floating point numbers and bool for boolean. Moreover, it has some simple data structures in the standard library:

Tuple: Tuples are used to merge into a single structure multiple variables or objects. Tuples are immutable and are defined by specifying items separated by commas within an optional pair of parentheses. For instance the statement `t=(a,b,c)` creates a tuple, identified by `t`, containing `a`, `b`, and `c` objects. Tuples cannot be accessed by referencing the position. Following statements create two tuples and print their content.

```
b = ("one",)
```

```
# Two-element tuple.  
c = ("one", "two")  
if "one" is in c:  
    print("ok")  
# the statement print ok
```

Lists: A list is an ordered and indexed collection of heterogeneous object. Lists are created by using square brackets, e.g., `l=[]` or the `list()` statement i.e. `l=list()`. List are mutable and they may be accessed using indexes, e.g., `l[index]` or iterators e.g., `for x in l`. Python also offers main function for list operations, e.g., `append`, `delete`, `search`. Following statements create a list, put into the list 4 elements and finally write each element to the screen.

```
# creation of the list  
l=list()  
# Insertion of the elements into the list  
l.append(1)  
l.append(2)  
l.append('a')  
l.append('b')  
# print of the elements  
for x in l: print(x)
```

Sets. A set is an unordered list of objects in which each object compares once. Sets are created by using curly brackets e.g., `s=`. Python offers many functions to testing for membership, to test whether it is a subset of another set, to find the intersection between two sets.

```
# creation of the sets  
s1=set()  
s2=set()  
# Insertion of the elements into the sets  
s1.add(2)  
s1.add(4)  
s2.add(2)  
s2.add(4)  
s2.add(5)  
print(s1.intersection(s2))  
# Output: 2,4  
print(s1.union(s2))  
# Output: 2,4,5
```

Dictionaries: A dictionary is a ordered collection of key: values pair. Note that the key must be unique. It is mandatory to use only immutable objects (like strings) for the keys of a dictionary, while but you can use either immutable or mutable object for the values of the dictionary. Pairs of keys and values are specified by using the notation `dict = key1: value1, key2: value2`. Key-value pairs are separated by a colon, and the pairs are separated commas.

```
items = {'john': 4098, 'sean': 4139}  
items['guido'] = 4127  
print(items)  
Output: {'sean': 4139, 'guido': 4127, 'john': 4098}  
print(items['john'])  
Output: 4098  
del (items['sean'])  
items['irv'] = 4127  
print(items)  
{'guido': 4127, 'irv': 4127, 'john': 4098}  
print(items.keys())  
Output: ['guido', 'irv', 'john']  
'guido' in tel  
Output: True
```

Definition of Functions in Python

A function is a set of statement organised to perform a single, related action to provide modularity and code reusing.

Functions are defined using the keyword `def` followed by the function name and parentheses(`()`). Any input parameters or arguments should be placed within these parentheses. The first statement of a function may be a documentation string of the function

(or docstring). The code block within every function is indented and the end of the function is the end of the indentation or the state `return[expression]` that exits a function, optionally passing back an expression to the caller.

```
# definition of a function

# def triplicate(a):
# body of the function
#     return 3*a
# return statement
# use of the function

a=triplicate(3)
print(a)
Output: 9

def printinfo(name, age):
    "This prints a passed info into this function"
    print "Name: ", name
    print "Age ", age
    return;

# Now you can call printinfo function
printinfo(age=50, name="miki")
```

Object Oriented Programming in Python

Object-Oriented Programming in Python is substantially different from classical OOP languages like Java. The definition of a class in Python starts with the `class` command followed by a set of statements. Python does not admit the explicit definition of attributes, and they are defined as the `__init__()` function that also acts as a constructor. Moreover, attributes may also be added during the use of the object. Therefore two objects of the same class may have different attributes. Finally, Python does not admit an explicit definition of public and private functions. Private functions are defined as `__function(self)`, but their meaning is quite different from other languages like Java. Multiple inheritances is not admitted and interface cannot be defined. For instance, the simple definition of a `Point` class, storing the `x` and `y` coordinates of a point in a two-dimensional plane is the following.

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def getCoordinates(self):
        return self.x, self.y
```

The point has two attributes `x`, and `y` and the class have two functions, the constructor and the methods `getCoordinates`. The function called `__init__` is run when we create an instance of `Point`. The `self` keywords are similar to this keyword in java since it references to the implicit object. All the class methods need to use `self` as the first parameter, but during the use, the parameter (`self`) is omitted, thus one may invoke class methods, as usual, using the `(.)` style. `self` is the first parameter in any function defined inside a class. To access these functions and variables elsewhere inside the class, their name must be preceded with `self` and a full-stop (e.g., `name.method`).

Bioinformatics and Data Science in Python

Recently, data science has become a large field and its application in bioinformatics, in computational biology and in biomedicine are very popular. Data science, following a common accepted definition, is a novel field in which statistics, data analysis and computer science are mixed together to extract knowledge or insights from data. In particular, from computer science, data analysis uses methods and tools from machine learning, data mining, databases, and data visualisation. Python is largely used in data science, therefore we will present some project that offer useful libraries for data science in Python.

Among the others, one motivation for the use of Python in data science, is the possibility to easily distribute libraries through the community. Python community has developed some databases for distributing codes. Actually the Python Package Index (PyPI) is a freely available repository of softwares. PyPI offer the possibility for the developers to reach a big community and for the community to find and install packages and libraries. PyPI (see Relevant Websites section), that allows to users to find Python packages in various domains, including Bioinformatics and datascience. Using PyPI users may install libraries using a simple command from the prompt, in a similar way to `apt-get` for linux distributions.

Data Science Packages in Python: Scikit-Learn

Using the PyPI search engine for data science user may find more than 50 libraries for data science. One of the most popular libraries is sci-kit learn.

Scikit-learn includes implementations for various classification, regression and clustering algorithms. Implemented models span a large model including support vector machines, random forests, neural networks, and classical clustering algorithms like k-means and DBSCAN. It is based on Python numerical libraries and it easily interoperate with other libraries. Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Scikit-learn also offers a popular website that includes a large set of examples to learn the use of libraries. To install scikit-learn user may use pip and the command `pip install -U scikit-learn`.

Biopython

The Biopython Project (Cock *et al.*, 2009) is an association of developers that aim to develop tools for computational biology. Biopython is available for the user through the PyPI installers and through the web site <http://biopython.org>.

BioPython contains many classes for supporting researchers and practitioners in molecular biology. Using Biopython user may read biological sequences and annotations written in different file formats as well as he/she may interoperate with almost all the online databases of biological information. Biopython offer modules for sequence and structure alignment as well as some simply machine learning algorithms.

References

- Cai, X., Langtangen, H.P., Moe, H., 2005. On the performance of the python programming language for serial and parallel scientific computations. *Scientific Programming* 13 (1), 31–56.
- Cock, P.J., Antao, T., Chang, J.T., *et al.*, 2009. Biopy-thon: Freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25 (11), 1422–1423.
- Van Rossum, G., *et al.*, 2007. Python programming language. In: *USENIX Annual Technical Conference*, vol. 41, p. 36.
- Zelle, J.M., 2004. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates, Inc.

Relevant Websites

<http://biopython.org>
Biopython.
<https://pypi.python.org/pypi>
Python.

Biographical Sketch

Pietro H. Guzzi is an assistant professor of Computer Science Engineering at the University Magna Grcia of Catanzaro, Italy. His research interests comprise semantic-based and network-based analysis of biological and clinical data.

Computing Languages for Bioinformatics: R

Marianna Milano, University of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

R is a free software programming language and a software environment for statistical computing and graphics. The R language is widely used among the scientific community for developing statistical software and data analysis. It offers the widest range of available methodologies such as linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, and clustering and graphical techniques for understanding data, from the most basic to the most complex. R is an open source project, and there are freely available command line interfaces or graphical front-ends for different platforms, including Windows, Mac OS X, and Linux. R is under constant development, with new updates added daily. In fact, R is supported by scientists and programmers who can help and advise the users. In the programming language field, R is an important tool for development in the numerical analysis and machine learning spaces. Due to the high-level of interpreted languages, such as R, the user can easily and quickly prototype new computational methods. R has become very popular, and is now being used for projects that require substantial software engineering, as well continuing to be used as an interactive environment for data analysis. Over the years, the R has become one of the most used tools for scientific computation. The reasons are related to the existence of a collection of comprehensive statistical algorithms, access to high-quality numerical routines, and integrated data visualization tools. A second strong motivation for using R is its ability to interoperate with many other languages. Algorithms that have been written in another language rarely need to be reimplemented for use in R. Typically, one need write only a small amount of interface code, and the routines can be accessed from within R. Finally, R supports the creation and use of self-describing data structures.

R is the programming language on which the Bioconductor Project ([Reimers and Carey, 2006](#)) is founded. The goal of Bioconductor is the creation of extensible software for computational biology and bioinformatics. Most Bioconductor components are distributed as R packages. The Bioconductor project provides R packages for the analysis of genomic data, such cDNA microarrays, object-oriented data-handling, and for analysis of data from next-generation high-throughput sequencing methods.

R Environment

R is an integrated suite of software that enables calculation, data manipulation, and graphical display. In particular, R comprises an effective data handling and storage facility, a suite of operators for calculations on arrays, an integrated collection of tools for data analysis, graphical facilities for data analysis and display, and is an effective programming language that includes conditionals, loops, user-defined functions, and input and output facilities.

Analysis and Assessment

R Package System

The R environment includes a well-established package system, together with related software components and documentation. The package system represents the heart of the R System. There are several hundred packages that enable a wide range of statistical analyses and visualization objectives. An R package typically consists of a collection of functions and data structures that are appropriate for solving a specific problem. Each R package folder contains the R code, help pages, data, vignette documentation (see below), code written in other languages such as C or FORTRAN, and files that describe how to install the package. Packages should run on all platforms supported by R. There are currently three main repositories for R packages: Bioconductor ([Reimers and Carey, 2006](#)), CRAN ([Claes et al., 2014](#)), and Omegahat ([Lang, 2000](#)). CRAN contains over 1000 R packages, while Bioconductor and Omegahat are smaller. Packages can be downloaded as open source. The package system ensures that software modules are developed and distributed with clearly defined standards of test-based validation, version identification, and package interdependency.

Object-Oriented Programming Support

Object-oriented programming (OOP) ([Rumbaugh et al., 1991](#)) has become widely used in software engineering. R currently supports two internal OOP systems: S3 and S4. S3 does not require the specification of classes, and there is control of objects and inheritance. The emphasis of the S3 system was on generic functions and polymorphism. S4 requires formal class definitions. In S4, classes are defined to have specific structures and inheritance relationships, and methods are defined both generically and specifically. S4 is better suited for developing large software projects but has an increased complexity of use.

World Wide Web Connectivity

R contains a set of functions and packages that provide access to different databases and to web resources. For example, there are packages for dealing with XML ([W3C](#)), and a SOAP client package ([Mein et al., 2002](#)), SSOAP, both available from the Omegahat project.

Visualization Capability

R has built-in support for many simple plots and graphs, including pie charts, bar charts, histograms, line graphs, and scatterplots. In addition, there are many graphical tools more specifically designed for bioinformatics analysis, such as the heatmap function, which is used to plot the expression levels of genes. Generally, R provides interactive plots so that users can query and navigate through them, and our future plans involve more such development.

Documentation

Most R packages provide an optional supplemental documentation, known as a vignette ([Gentleman et al., 2004](#)). Vignettes are provided in addition to the required documentation for R functions and datasets. Vignettes are written in order to share knowledge, and assist new users in learning the purpose and use of a package. A vignette is a document that integrates code and text, and describes how to perform a specific task. Vignettes were developed as part of the Bioconductor Project. They have since been incorporated as part of the R system. For developers, the Sweave system ([Leisch, 2002](#)) is well suited to creating and processing vignettes, and is the most widely used tool for programming in R.

Support for Parallel Computing

R supports parallel computing. In general, parallel computing refers to simultaneous calculations across multiple cores of multi-processor computer. Theoretically, procedures are trivial to parallelize. However, the development of a parallelized implementation that it also is robust and reliable is far from trivial. R provides an easy and powerful programming interface for the computational clusters. The interface allows the rapid development of R functions that allocate the calculations across the computational cluster. The approach can be extended to complex parallelization. There are different packages such as Rmpi ([Yu, 2009](#)), rpvm ([Li and Rossini, 2001](#)), snow ([Tierney et al., 2009](#)), and nws ([Schmidberger et al., 2009](#)) that support parallel computing. These tools provide simple interfaces that allow parallel computation of functions in concurrent R sessions. For example, the snow package provides a higher level of abstraction, independent of inter-processor communication technology (for instance, the message-passing interface (MPI) ([MPI Forum](#)), or the parallel virtual machine (PVM). A parallel random number generation ([Mascagni et al., 1999](#)), essential when distributing parts of stochastic simulations across a cluster, is provided by rsprng. More details about the benefits and problems involved with programming parallel processes in R are described in [Rossini et al. \(2007\)](#).

Illustrative Example of R Use

In this section, the practical use of R is presented, starting from R installation, and introducing basic concepts such as the creation of vectors, matrices, lists, dataframes, arrays, as well as functions.

R Installation

To use R, the user needs to install the R program his/her computer. R software is available from the Comprehensive R Network (CRAN) website, see Relevant Website section, a server on which both R releases, and a whole range of tools and additional implementations developed by the various developers, are available. The R download is available for Linux, Windows, and (Mac) OS X. After installation, the user starts to use R-GUI.

R Syntax

R is similar to other programming languages, such as C, Perl, or Python, and provides similar sets of mathematical operators (+, -, *, /, ^ (exponentiation), %% (modulo)), and logical operators (<, <=, >, >+, ==, !=, | (OR), & (AND), and isTRUE). Familiar loop control statements such as repeat, while, and for are available in R. R supports a flexible set of data types: logical, numeric, integer, complex, character, and raw byte. In programming, the user uses variables to store different information of various data types. In R, the variables are stored as R-Objects, and there are several basic types: Vectors, Matrices, Arrays, Lists, Factors, and Data Frames.

Vector

Vectors are a fundamental data type in R, and are composed of logical, integer, double, complex, character, or raw byte values. Implicitly, single values in R are vectors of length 1. To create a vector, the user should use the `c()` function, which combines the

listed elements into a vector. Vectors with an ordered set of elements can also be created using the `seq()` function. The basic syntax for creating a vector in R is:

```
c(data) or seq(begin, end, step)
```

where *data* indicates the elements of the vector. For `seq()`, *begin* and *end*, indicate the first and last element (inclusive), and *step* specifies the spacing between the values. Examples of creating vectors are shown below:

```
# Create vector using c()
v <- c('red','white','yellow')
print(v)
[1] "red" "white" "yellow"
# Create vector using seq()
v <- seq(5, 7, 0.5)
print(v)
[1] 5.0 5.5 6.0 6.5 7.0
```

Matrix

A matrix in R is a two-dimensional array, usually of numerical values, but characters or logical values can also be used. To create a matrix, the user should use the `matrix()` function. The basic syntax for creating a matrix in R is:

```
matrix(data, nrow, ncol)
```

where *data* is the input vector, which becomes the data elements of the matrix, *nrow* is the number of rows to be created, and *ncol* is the number of columns to be created. An example of creating a 2 by 3 matrix is shown below:

```
# Create matrix
m = matrix(c('1','2','3','4','5','6'), nrow = 2, ncol = 3)
print(m)
[, 1] [, 2] [, 3]
[1,] 1 2 3
[2,] 4 5 6
```

Array

Arrays store data in two or more dimensions. In R, arrays can store only one datatype. To create an array, the user should use the `array()` function. The basic syntax for creating an array in R is:

```
array(data, dim)
```

where *data* are the input vectors, and *dim* specifies the dimensions of the array.

An example of creating an array is shown below:

```
# Create two vectors
v1 <- c(1,2,3)
v2 <- c(4,5,6,7,8,9)

# Combine the vectors to create an array
z <- array(c(v1,v2), dim=c(3,3,2))
print(z)
„1
[, 1] [, 2] [, 3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
„2
[, 1] [, 2] [, 3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

List

Lists may contain data of different types. To create a list, the user should use **list()** function.

The basic syntax for creating an array in R is:

```
list(data)
```

where *data* may be strings, numbers, vectors and a logical values. An example to create a list is reported below:

```
# Create a list containing a mixture of string, vector, logical, and numerical values
# Note that the vector is stored in the list as a two element vector
list_example <- list("Flower", c(1,2), TRUE, 0.23)
print(list_example)
[[1]]
[1] "Flower"

[[2]]
[1] 1 2

[[3]]
[1] TRUE

[[4]]
[1] 0.23
```

Factor

A factor is used to store a set of data as a series of levels or categories. Factors can store either character or integer values. They are useful for data that has an enumerable number of discrete values such as male/female or north/south/east/west. To create a factor, the user should use the **factor()** function. The basic syntax for creating a factor in R is:

```
factor(data)
```

An example of creating a factor is shown below:

```
# Create a vector as input
data <- c("East", "West", "East", "North", "North", "East", "West", "West", "West", "East", "North")

# Apply the factor function
factor_data <- factor(data)
print(factor_data)
[1] East West East North North East West West West East North
Levels: East North West
```

Data frame

A data frame is a table in which each column contains values of one kind, and the rows correspond to a set of values. Data frames are typically used for experimental data, where the columns represent information or measured values, and the rows represent different samples. To create a data frame, the user should use **data.frame()** function.

The basic syntax for creating an array in R is:

```
data.frame(data)
```

where *data* are a collection of variables, which share many of the properties of matrices and of lists. An example of creating a data frame is shown below:

```
# Create three vectors
sample <- c(1, 2, 3)
v2 <- c('a', 'b', 'c')
v3 <- c(TRUE, TRUE, FALSE)

# Create a dataframe from the three vectors
df <- data.frame(sample, v2, v3)
print(df)
sample v2 v3
1      1 a TRUE
2      2 b TRUE
3      3 c FALSE
```


R Functions

Functions are subprograms that perform a narrowly defined task. A function consists of a set of statements; the function receives information from the main program, its arguments, and sends information back to the main program after it finishes, its return value. To create a R function, the user should use the keyword **function**. The basic syntax for defining a R function is:

```
FunctionName <- function (arg1, arg2, ...argn) {
  statements
  return(object)
}
```

where *FunctionName* is the name of function, stored in R as object, *arg1* to *argn* are the arguments of the function (i.e., the values the function will use in its calculation), *statements* are the tasks to perform, and *object* is the value of the function which is returned to the main program.

In addition to the functions in-built in R, the user can build personalized functions.

An example of creating a function is shown below:

```
# Create a function that computes the area of a triangle
triangle_area <- function(height, base){
  area <- (height*base) / 2
  return(area)
}
```

R Packages

Packages are collections of functions, compiled code from other languages, and sample data. Packages are stored in the R environment library, and many packages are automatically downloaded and installed during R installation. As mentioned above, additional user contributed packages are often downloaded during the course of an R analysis. The R package system is one of the strengths of R because it makes hundreds of advanced statistical and computational analyses available. Part of the beauty of the system is that any user can create and contribute their own packages. While creating a package involves some extra effort to create the required documentation, it is not particularly different. Complete instructions can be found in [Leisch \(2009\)](#).

Case Studies

In this section, an example of bioinformatics issues managed with R is described.

Biological Annotation

Many bioinformatic analyses rely on the processing of sequences and their metadata ([Leipzig, 2016](#)). Metadata is data that summarizes information about other data. There are two major challenges related to metadata ([Duval et al., 2002](#)). First, is the evolutionary nature of the metadata. In fact, as biological knowledge increases, metadata also changes and evolves. The second major problem that concerns metadata data is its complexity. In R, these issues are tackled by placing the metadata into R packages. These packages are constructed by a semi-automatic process ([Zhang et al., 2003](#)), and are distributed and updated using the package distribution tools in the *reposTools* package ([Core, 2002](#)). R contains many different types of annotation packages. There are packages that contain Gene Ontology (GO) ([Ashburner et al., 2000](#)), Kyoto Encyclopedia of Genes and Genomes (KEGG) ([Kanehisa and Goto, 2000](#)), and other annotations, and R can easily access NCBI, Biomart, UCSC, and other sources. For example, the *AnnotationDbi* ([Milano et al., 2014](#)) package provides increased flexibility, and makes linking various data sources simpler. The *AnnotationDbi* package ([Pages et al., 2008](#)) contains a collection of functions that can be used to make new microarray annotation packages. For example, the building of a chip annotation package consists of two-steps. The first step regards the construction of a database that conforms to a standard schema for the metadata of the organism that the chip is designed for. The use of a standard schema allows the new package to integrate with other annotation packages, such as *GO.db* and *KEGG.db*. The construction of a chip-specific database requires two inputs, a file containing the mapping between the chip identifiers, and a special intermediate database. This database contains information for all genes in the model organism, and many different biological IDs for data sources, such as Entrez Gene, KEGG, GO, and Uniprot. In the second step, the chip-specific database is applied to build an R package. The typical metadata used in Bioinformatics are the annotations, which provide useful and relevant in different applications that analyze biological data, e.g., in functional enrichment analyses, the use of GO to describe biological functions and processes, or in model bioinformatics applications, to guide the composition of workflows ([Milano et al., 2016](#)). Furthermore, annotations allow the comparison of molecules on the basis of semantics aspects through semantic similarity measures (SSMs) ([Milano et al., 2014](#)). In R there are different packages that compute of many SS measures, such as, *GOSemSim* ([Yu, 2010](#)), *GOVis* ([Heydebreck et al., 2004](#)), *csbl.go* ([Ovaska, 2016](#)). Among these tools *csbl.go* is the most widely used R Package for semantic analysis. It contains a set of functions for the calculation of semantic similarity measures, as

well as for clustering SS scores. It requires as input a list of GO Terms, or (in the case of proteins) a list of proteins, as well as the related annotations for each protein. It currently runs only for a subset of proteomes, but it may be extended by the user for the evaluation of semantic similarities to other proteomes. An example of an ad hoc function for the computation of semantic similarity using `csbl.go` is shown below:

```
SSM <- function(inputfile, tax, ontology, measure) {
  set.prob.table(organism=tax, type="similarity")
  ent <- entities.from.text(inputfile)
  SSM_result = entity.sim.many(ent,ontology,measure)
  return(SSM_result)
}
```

where *inputfile* is the set of Gene Ontology annotations, *tax* is the taxonomy (e.g., *Homo sapiens*; *Saccharomyces cerevisiae*; *Caenorhabditis elegans*; *Drosophila melanogaster*; *Mus musculus*; *Rattus norvegicus*), *ontology* indicates the GO ontology on which the semantic similarity is computed, the *measure* argument is one the following semantic similarity measures implemented in `csbl.go`: Resnik, Lin, JiangConrath, Relevance, ResnikGraSM, LinGraSM, JiangConrathGraSM, Kappa, Cosine, WeightedJaccard, or CzekanowskiDice.

See also: Computing for Bioinformatics

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene Ontology: Tool for the unification of biology. *Nature Genetics* 25 (1), 25–29.
- Claes, M., Mens, T., Grosjean, P., 2014. On the maintainability of CRAN packages. In: 2014 Software Evolution Week – IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), IEEE, pp. 308–312.
- Core, B., 2002. Bioconductor: Assessment of current progress. *Biocore Technical Report* 2.
- Duval, E., Hodgins, W., Sutton, S., Weibel, S.L., 2002. Metadata principles and practicalities. *D-lib Magazine* 8 (4), 16.
- W3C. eXtensible markup language (XML). Available at: <http://www.w3.org/XML>.
- Gentleman, R.C., Carey, V.J., Bates, D.M., *et al.*, 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology* 5 (10), R80.
- Heydebreck, A., Huber, W., Gentleman, R., 2004. Differential expression with the bioconductor project. In: *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics*. New York, NY: Wiley.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Lang, D.T., 2000. The omegahat environment: New possibilities for statistical computing. *Journal of Computational and Graphical Statistics* 9 (3), 423–451.
- Leipzig, J., 2016. A review of bioinformatic pipeline frameworks. *Briefings in Bioinformatics* 18 (3), 530–536.
- Leisch, F., 2002. Sweave: Dynamic generation of statistical reports using literate data analysis. In: *Compstat 2002 – Proceedings in computational statistics*, pp. 575–580. Heidelberg: Physica-Verlag.
- Leisch, F., 2009. Creating R packages: A tutorial. Available at: <https://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf>.
- Li, M.N., Rossini, A.J., 2001 RPVM: Cluster statistical computing in R. Porting R to Darwin/X11 and Mac OS X 4. *R News*, p. 4.
- Mascagni, M., Ceperley, D., Srinivasan, A., 1999. SPRNG: A scalable library for pseudorandom number generation. *ACM Transactions on Mathematical Software* 26, 436–461.
- Mein, G., Pal, S., Dhondt, G., *et al.*, 2002. U.S. Patent No. 6,457,066. Washington, DC: U.S. Patent and Trademark Office.
- MPI Forum. Message-Passing Interface (MPI). Available at: <http://www.mpi-forum.org>.
- Milano, M., Agapito, G., Guzzi, P.H., Cannataro, M., 2014. Biases in information content measurement of gene ontology terms. In: 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, pp. 9–16.
- Milano, M., Agapito, G., Guzzi, P.H., Cannataro, M., 2016. An experimental study of information content measurement of gene ontology terms. *International Journal of Machine Learning and Cybernetics*. 1–13.
- Ovaska, K., 2016. Using semantic similarities and `csbl.go` for analyzing microarray data. *Microarray Data Analysis: Methods and Applications*. 105–116.
- Pages, H., Carlson, M., Falcon, S., *et al.*, 2008. Annotation Database Interface. R package version 1(2).
- PVM. Parallel Virtual Machine. Available at: <http://www.csm.ornl.gov/pvm/>.
- Reimers, M., Carey, V.J., 2006. Bioconductor: An open source framework for bioinformatics and computational biology. *Methods in Enzymology* 411, 119–134.
- Rossini, A.J., Tierney, L., Li, N., 2007. Simple parallel statistical computing in R. *Journal of Computational and Graphical Statistics* 16 (2), 399–420.
- Rumbaugh, J., Blaha, M., Premerlani, W., *et al.*, 1991. *Object-Oriented Modeling and Design*, vol. 99 (No. 1). Englewood Cliffs, NJ: Prentice-hall.
- Schmidberger, M., *et al.*, 2009. State-of-the-art in parallel computing with R. *Journal of Statistical Software* 47, 1.
- Tierney, L., Rossini, A.J., Li, N., 2009. Snow: A parallel computing framework for the R system. *International Journal of Parallel Programming* 37 (1), 78–90.
- Yu, H., 2009. Rmpi: Interface (wrapper) to mpi (message-passing interface). Available at: <http://CRAN.R-project.org/package=Rmpi>.
- Yu, G., 2010. GO-terms semantic similarity measures. *Bioinformatics* 26 (7), 976–978.
- Zhang, J., Carey, V., Gentleman, R., 2003. An extensible application for assembling annotation for genomic data. *Bioinformatics* 19 (1), 155–156.

Relevant Website

<http://cran.r-project.org/>
CRAN.

Biographical Sketch



Marianna Milano received the Laurea degree in biomedical engineering from the University Magna Græcia of Catanzaro, Italy, in 2011. She is a PhD student at the University Magna Græcia of Catanzaro. Her main research interests are on biological data analysis and semantic-based analysis of biological data. She is a member of IEEE Computer Society.

Computing Languages for Bioinformatics: Java

Pietro H Guzzi, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Java Programming language was designed by James Gosling, Mike Sheridan, and Patrick Naughton in early 1991. The syntax of Java is quite similar to C/C++ (Ellis and Stroustrup, 1990), therefore is based on the use of curly brackets to define code blocks and on the use of semicolons for each statement. The first public implementation of Java was released in 1995 by Sun Microsystems. Java was based on the use of a free Java Virtual Machine (JVM) available for almost all operating systems, able to execute the code, following the paradigm “Write Once, Run Anywhere” (WORA) (Arnold *et al.*, 2005).

Since first version, Java was designed with particular attention to security and network secure access to run java code (namely java applets) on web-browsers. The second version of Java (named Java 2 or J2SE), was more structured concerning Java 1 and more version of java were available. For instance, J2EE was designed for enterprise application while J2ME for mobile applications. JVM was initially released under GNU General Public License and on 2007 JVM code was available under free software/open-source distribution terms. Currently, Oracle detains the right of the Java.

The Structure of a Program in Java

As introduced before, the syntax of Java has been largely derived from C++ but it was built an object-oriented language. Consequently, all the data structures are represented as java objects (Bruegge and Dutoit, 2004).

All code is written inside classes, and every data item is an object except for primitive data types (i.e., integers, floating-point numbers, boolean values, and characters). Other differences with respect C++ are the absence of pointers and the lack of support for operator overloading and multiple inheritance for classes (Horstmann and Cornell, 2002). Java admits the definition of interfaces, i.e a reference type in Java that include a set of abstract methods. Classes implements the interfaces, i.e. they realise the abstract method, and a class may implement multiple interfaces, therefore realising a sort of multiple inheritance. Consequently, even a simple hello world application need the declaration of a class as follows.

```
public class helloworld {  
    public static void main (String args[]) {  
        System.out.println ("Hello World");  
    }  
}
```

The application code must be saved into a source files (with .java extension) that has the same name of the class containing the main method. Therefore the file containing the previous code must be named helloworld.java. It must first be compiled into bytecode, using a Java compiler, producing a file named HelloWorldApp.class. Only then can it be executed on a Java Virtual Machine.

It should be noted that three keywords are used for the main method: public, static and void. The keyword public it used for all the methods that may be called in other classes. This keyword is called access level modifier, other access level are private (a method that may be called only inside the same class) and protected (a method that is called inside classes from the same package). The keyword static indicates a method that is associated with the class and not with a specific object. Static methods cannot access any class members that are not also static. Methods that are not designated static are instance methods and require a specific instance of class to operate. The keyword void indicates that the main method does not return any value to the caller. The method name “main” is the name of the method the Java launcher calls to pass control to the program. The main method must accept an array of String objects.

Data Types in Java

Java is a statically-typed programming language, therefore each variable must be declared before the use by stating the type and the name of the variable. Java offers to the user eight primitive data types:

- Byte: The byte data type is an 8-bit signed two’s complement integer. It has a minimum value of -128 and a maximum value of 127 (inclusive).
- Short: The short data type is a 16-bit signed two’s complement integer. It has a minimum value of $-32,768$ and a maximum value of $32,767$ (inclusive).
- Int: The int data type is a 32-bit signed two’s complement integer, which has a minimum value of $-2,147,483,648$ and a maximum value of $2,147,483,647$.

- Long: The long data type is a 64-bit two's complement integer. The signed long has a minimum value of -2^{63} and a maximum value of $2^{63}-1$. (In Java 8 the programmer may also define a unsigned int and long).
- Float: The float data type is a single-precision 32-bit IEEE 754 floating point.
- Double: The double data type is a double-precision 64-bit IEEE 754 floating point.
- Boolean.
- Char: The char data type is a single 16-bit Unicode character.

Strings are represented by using the java.lang. String Class.

Data Structures in Java

Java does not define a set of primitive data structure, e.g. like Python, but many structure are available through the Java Collection Framework (JCF), a set of classes and interfaces that implement commonly reusable collection data structures. The JCF provides both interfaces that define various collections and classes that implement them.

Collections are derived from the java.util. Collection interface that defines the basic parts of all collections, including basic add(), remove() and contains() methods for adding to and removing from and to checks if a specified element is in the collection. All collections have an iterator that goes through all of the elements in the collection. From Java 1.6 any collection can be written to store any class. For example, Collection String can hold strings, and the elements from the collection can be used as strings.

Collections are subdivided into three main generic types: ordered lists, dictionaries/maps, and sets.

Two interfaces are included in the Ordered Lists which are the List Interface and the Queue Interface. Dictionaries/Maps store references to objects with a lookup key to access the object's values. One example of a key is an identification card. The Map Interface is included in the Dictionaries/Maps. Sets are unordered collections that can be iterated and where similar objects are not allowed. The Interface Set is included.

Creation and Running of a Program

Let suppose that a programmer wants to create a simple hello world application. He/she has to create a file (with .java extension) containing the main class with the same name of the class (helloworld.java in this case).

```
public class helloworld {
    public static void main (String args[] ) {
        System.out.println ("Hello World");
    }
}
```

It must first be compiled into bytecode, using a Java compiler, producing a file named helloworld.class, using the java compiler (javac helloworld.java). Finally, the program can be executed on a Java Virtual Machine, java helloworld.

Web Programming in Java

The possibility to run a code inside a web browser was one of the aims of the initial development of Java. Therefore a lot of effort has been made to ensure this possibility leading to the development of many Java-based technologies and programming models for the web. Among the others, we here report applets, servlets and Java Server Pages (JSP).

Java applets are programs that are embedded in other applications, typically in a Web page displayed in a web browser. Java Servlet technology provides a way to extend the functionality of a web server enabling the possibility to generate responses (typically HTML pages) to requests (typically HTTP requests) from clients. JavaServer Pages (JSP) are server-side Java EE components that generate responses, typically HTML pages, to HTTP requests from clients. JSP contains embedded java code in web pages. Each JSP page is compiled into a Java servlet the first time it is accessed. Then it is executed securely.

Concurrency in Java

The Java programming language and the Java virtual machine (JVM) have been designed to support concurrent programming. The programming model is based on threads: each thread has its path of execution. The programmer must take care of the safely read and write access to objects, accessible by many threads, in a synchronized way. Synchronization ensures that objects are modified by only one thread at a time. The Java language has built-in constructs to support this coordination.

Threads are also called lightweight processes, and a program in Java usually runs as a single process. Each thread is associated with an instance of the class Thread. Every application has at least one thread (defined as main thread) that may create additional

threads called Runnable objects (or Callable in recent versions). All the threads share the process's resources, including memory and open files. The mapping among java threads and OS threads is different on each operating systems and JVM implementation.

There are different ways two start a thread; we here present the use of class Thread and the use of the interface Runnable.

```
// using the interface Runnable

public class HelloRunnable implements Runnable {
    @Override
    public void run(){
        System.out.println("Hello from thread!");
    }
    public static void main(String[] args) {
        (new Thread(new HelloRunnable())).start();
    }
}

// Extension of the class Thread

public class HelloThread extends Thread {
    @Override
    public void run() {
        System.out.println("Hello from thread!");
    }
    public static void main(String[] args) {
        (new HelloThread()).start();
    }
}
```

Java for Bioinformatics: BioJava

BioJava ([Prlić *et al.*, 2012](#)) is an open-source software project offering libraries and tools to manage biological data such as sequences, protein structures, file parsers, Distributed Annotation System (DAS), and simple statistical algorithms. Using BioJava researches may transparently manage DNA and protein sequences as well as protein structures. BioJava is based on an application programming interface (API) supporting file parsers, database access, data models and algorithms.

See also: Computing for Bioinformatics

References

- Arnold, K., Gosling, J., Holmes, D., 2005. The Java Programming Language. Addison Wesley Professional.
- Bruegge, B., Dutoit, A.H., 2004. Object-Oriented Software Engineering Using UML, Patterns and Java-(Required), 2004. Prentice Hall.
- Ellis, M.A., Stroustrup, B., 1990. The Annotated C++ Reference Manual. Addison-Wesley.
- Horstmann, C.S., Cornell, G., 2002. Core Java 2: Volume I, Fundamentals. Pearson Education.
- Prlić, A., Yates, A., Bliven, S.E., *et al.*, 2012. Biojava: An open-source framework for bioinformatics in 2012. *Bioinformatics* 28 (20), 2693–2695.

Biographical Sketch

Pietro H. Guzzi is an assistant professor of Computer Science Engineering at the University Magna Gracia of Catanzaro, Italy. His research interests comprise semantic-based and network-based analysis of biological and clinical data.

Parallel Architectures for Bioinformatics

Ivan Merelli, Institute for Biomedical Technologies (CNR), Milan, Italy and National Research Council, Segrate, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The increasing availability of omic data resulting from improvements in molecular biology experimental techniques represents an unprecedented opportunity for Bioinformatics and Computational Biology, but also a major challenge (Fuller *et al.*, 2013). Due to the increased number of experiments involving genomic research, in particular due to the spreading of these techniques in hospitals, the amount and complexity of biological data is increasing very fast.

In particular, the high demand for low-cost sequencing has driven the development of high-throughput technologies that parallelize the sequencing process, producing millions of sequences concurrently (Church, 2006). High-throughput, or next-generation, sequencing (NGS) applies to genome sequencing, genome resequencing, transcriptome profiling (RNA-Seq), DNA-protein interactions (ChIP-sequencing), and epigenome characterization (de Magalhães *et al.*, 2010).

Such huge and heterogeneous amount of digital information is an incredible resource for uncovering disease associated hidden patterns in data (Merelli *et al.*, 2013), allowing the creation of predictive models for real-life biomedical applications (Alfieri *et al.*, 2008). But suitable analysis tools should be available to life scientists, biologists and physicians to properly treat this information in a fast and reliable way.

Due to the huge volume of information daily produced, it is almost impossible to process all data using an ordinary desktop machine in standalone executions. Since most of the analysis have non-linear complexity, the need for computational power to perform bioinformatic analysis grows very fast. Scientists need to use high-performance computing (HPC) environments together with parallel techniques to process all the produced data in a reasonable time.

Several large-scale bioinformatic projects already benefit from parallelization techniques and HPC infrastructures (Pérez-Sánchez *et al.*, 2015), in particular considering clusters of high-end servers connected by fast networks. Indeed, most of the modern supercomputers run, among the others, applications from the computational biology domain, since Bioinformatics provides impressive developing and testing opportunities for research in HPC applications. Some vast, rich, and complex bioinformatic areas related to genomics can also benefit from HPC infrastructures and parallel techniques, such as NGS, Proteomics, Transcriptomics, Metagenomics, and Structural Bioinformatics.

More recently, the development of cards harbouring hundreds of cores changed the paradigm of parallel computing, generating a large impact also in High Performance Bioinformatics. Starting from 2010, graphics processing units (GPUs), specialized computer processors addressing real-time compute-intensive 3D graphic tasks, evolved into highly parallel multi-core systems allowing very efficient manipulation of large blocks of data. These architectures are more effective than general-purpose central processing unit (CPUs) for algorithms that require processing large blocks of data in parallel. The drawback is that algorithms should be reimplemented to exploit the vectorized architecture of these devices and much work is necessary to optimize their performance.

Although many devices are available for GPU computing, the most popular cards are manufactured by NVIDIA, which also developed a parallel computing platform and application programming interface (API) called Compute Unified Device Architecture (CUDA) (Nickolls *et al.*, 2008). This framework allows software developers and software engineers to use CUDA-enabled GPUs, providing a software layer that gives direct access to the GPU parallel computational elements, for the execution of compute kernels. CUDA has been largely used in Bioinformatics, in particular in the field of structural biology, although not all the applications are suitable for this kind of implementation.

On the other hand, x86 compatible coprocessors have been developed to exploit hundreds of cores without the need of reimplementing any algorithm. The most popular of these cards is the Intel XeonPhi. Thanks to their x86 architecture, XeonPhi allows the use of standard programming language APIs, such as OpenMP (Dagum and Menon, 1998). Although these cards have a nominal performance that exceed 1 Tflops, their naive usage for bioinformatic applications is usually unsuccessful. As for GPU cards, the real challenge is the optimization of the code to exploit the architecture at the best of its capability, which has been achieved for few applications, in particular of molecular dynamics.

Parallelization Paradigms

Classification of parallel programming models can be divided broadly into two areas: process interaction and problem decomposition (Foster, 1995). Process interaction relates to the mechanisms by which parallel processes are able to communicate with each other. The most common forms of interaction are shared memory and message passing, although other approaches are possible (McBurney and Sleep, 1987).

Shared memory is an efficient means of passing data between processes. In a shared-memory model, parallel processes share a global address space that they read and write asynchronously. Asynchronous concurrent accesses can lead to race conditions and mechanisms such as locks, semaphores and monitors can be used to avoid these. Conventional multi-core processors directly

support shared memory, which many parallel programming languages and libraries, such as OpenMP (Dagum and Menon, 1998) and Threading Building Blocks (TBB) (Pheatt, 2008), are designed to exploit. This approach is typically used in Bioinformatics to parallelize applications within a single server or device.

In the message-passing model, parallel processes exchange data through passing messages to one another. These communications can be asynchronous, where a message can be sent before the receiver is ready, or synchronous, where the receiver must be ready. The Communicating Sequential Processes (CSP) formalisation of message passing uses synchronous communication channels to connect processes, leading to important languages such as Occam, Limbo and Go. In contrast, the actor model uses asynchronous message passing and has been employed in the design of languages such as Scala (Odersky *et al.*, 2004). However, the most popular approach for developing software using the message-passing model are language specific libraries that implement the Message Passing Interface standard, which defines the syntax and semantics of routines useful to write portable parallel programs (Gropp *et al.*, 1999).

Problem decomposition relates to the way in which the simultaneously executing processes of a parallel program are formulated (Foster, 1995). A task-parallel model focuses on processes, or threads of execution. These processes will often be behaviourally distinct, which emphasises the need for communication. Task parallelism is a natural way to express message-passing communication. On the other hand, a data-parallel model focuses on performing operations on a data set, typically a regularly structured array. A set of tasks will operate on this data, but independently on disjoint partitions. Data parallel applications are very common in Bioinformatics, in particular for sequence analysis.

Parallel Computing Platforms

The traditional platforms for operating parallel bioinformatic analysis are computer clusters (Merelli *et al.*, 2014), although parallel on-chip acceleration devices are commonly used to speed up the computation on desktop computers. In this section we will briefly review bioinformatic applications or projects exploiting these platforms.

Cluster Computing

The key issues while developing applications using data parallelism are the choice of the algorithm, the strategy for data decomposition, load balancing among possibly heterogeneous computing nodes, and the overall accuracy of the results (Rencuzogullari and Dwardadas, 2001).

The data parallel approach, that is the parallelization paradigm by which data are analysed by almost independent processes, is a suitable solution for many kinds of bioinformatic analysis. Indeed, the computation on biological data can be usually split in independent tasks, collecting and re-ranking results at the end. The possibility of working on each sequence independently makes data parallel approaches resulting in high scalability and performance figures for many bioinformatic applications.

This approach is generally compatible with clusters of computers, which can be combined to support the computational load. However, if processes need a lot of communication to accomplish their tasks, for example, due to complex post-processing analysis, it is important to have fast interconnecting networks, otherwise the scalability can be largely impaired. Importantly, the key feature to achieve good performance is to have low-latency networks (Fibre Channel, InfiniBand, Omni-Path, etc.), more than high-throughput networks (i.e., 10 Gigabit Ethernet).

An example of analysis that heavily relies on computer clusters and low-latency interconnecting networks is the *de novo* assembly of genomes. These approaches typically work finding the fragments that *overlap* in the sequencing reads and recording these overlaps in a huge diagram called de Bruijn (or assembly) graph (Compeau *et al.*, 2011). For a large genome, this graph can occupy many Terabytes of RAM, and completing the genome sequence can require days of computation on a world-class supercomputer. This is the reason why memory distributed approaches, such as Abyss (Simpson *et al.*, 2009), are now widely exploited, although algorithms that efficiently use multiple servers are difficult to implement and they are still under active development.

Virtual Clusters

Clusters can be also build in a virtualized manner on cloud, by using the on-demand paradigm. There are many tools for the automatic instantiation of clusters on virtual resources, which help users to manage node images, network connections and storage facilities. An example of such software is *AlcesFlight* (2016), which provides scalable High Performance Computing (HPC) environments, complete with job scheduler and applications, for research and scientific computing relying both on-demand and spot instances.

Concerning performance, virtual clusters should be also considered as very reliable in this cloud era: for example, a virtual infrastructure of 17,024 cores built using a set of Amazon Elastic Cloud Computing virtual machines was able to achieve 240.09 TeraFLOPS for the High Performance Linpack benchmark, placing the cluster at position 102 in the November 2011 Top500 list. A similar example was performed on Windows Azure, bringing together 8064 cores for a total of 151.3 TeraFLOPS, a virtual cluster that reached position 165 in the November 2011 Top500 list.

Virtual clusters are commonly used in Bioinformatics, for example, in the frame of drug discovery projects (D'Agostino *et al.*, 2013), which requires the screening of large datasets of ligands against a target protein (Chiappori *et al.*, 2013). Considering that each docking software is typically optimized for specific target families, it is usually a good idea to test many of them (Morris *et al.*, 2009; Mukherjee *et al.*, 2010; Friesner *et al.*, 2006; Merelli *et al.*, 2011), which increases the need for computational power. Moreover, side effects caused by off-target bindings should be avoided, therefore the most promising compounds are usually tested against many other proteins, which also requires time-consuming screenings.

GPU Computing

Driven by the demand of the game industry, GPUs have completed a steady transition from mainframes to workstations PC cards, where they emerge nowadays like a solid and compelling alternative to traditional parallel computing platforms. GPUs deliver extremely high floating-point performance and massively parallelism at a very low cost, thus promoting a new concept of the high performance computing market. For example, in heterogeneous computing, processors with different characteristics work together to enhance the application performance taking care of the power budget. This fact has attracted many researchers and encouraged the use of GPUs in a broader range of applications, particularly in the field of Bioinformatics. Developers are required to leverage this new landscape of computation with new programming models, which make easier the developers' task of writing programs to run efficiently on such platforms altogether (Garland *et al.*, 2008).

The most popular graphic card producers, such as NVIDIA and ATI/AMD, have developed hardware products aimed specifically at the heterogeneous or massively parallel computing market. The most popular devices are Tesla cards, produced by NVIDIA, and Fire-stream cards, which is the AMDs device line. They have also released software components, which provide simpler access to this computing power. Compute Unified Device Architecture (CUDA) is the NVIDIA solution for a simple block-based programming, while the AMDs alternative was called Stream Computing.

Although these efforts in developing programming models have made great contributions to leverage the capabilities of these platforms, developers have to deal with a massively parallel on-chip architectures (Garland and Kirk, 2010), which is quite different than working on traditional computing architectures. Therefore, programmability on these platforms is still a challenge, in particular concerning the fine-tuning of applications to get high scalability. Many research efforts have provided abstraction layers avoiding dealing with the hardware particularities of these accelerators and also extracting transparently high level of performance, providing portability across operating systems, host CPUs, and accelerators.

For example, OpenCL (Khronos Group, 2014) emerged as an attempt to unify all these models with a superset of features, being the best broadly supported multi-platform data parallel programming interface for heterogeneous computing, including GPUs, accelerators, and similar devices. However, other libraries and interfaces exist for developing with popular programming languages, like OpenMP or OpenACC, which describe a collection of compiler directives to specify loops and regions of code to parallelize in standard programming languages such as C, C++, or Fortran. Although the complexity of these architectures is high, the performance that such devices are able to provide justifies the great interest and efforts in porting bioinformatic applications on them (NVIDIA, 2016).

In Bioinformatics, one of the most successful application of GPUs concerns Molecular Dynamics simulations (Chiappori *et al.*, 2012). Molecular Dynamics is certainly the most CPU-demanding application in Computational Biology, because it consists in solving time step after time step the Newton's equations of motion for all the atoms of a bio-molecular system, taking as boundary conditions the initial macromolecular structure and a set of velocity taken from a Gaussian distribution.

Molecular Dynamics is often employed in combination with docking screenings, because while virtual screening is very useful for discarding compounds that clearly do not fit with the protein target, the identification of lead compounds is usually more challenging (Chiappori *et al.*, 2016). The reason is that docking software have biases in computing binding energy in the range of few kcal (Chiappori *et al.*, 2016). Therefore, best compounds achieved through the virtual screening process usually undergone to a protocol of energy refinement implemented using Molecular Dynamics (Alonso *et al.*, 2006).

Indeed, by employing specific simulation schemas and energy decomposition algorithms in the post-analysis, Molecular Dynamics allows to achieve more precise quantification of the binding energy (Huey *et al.*, 2007). Common techniques for energy estimation are MM-PBSA and MM-GBSA, which consist in the evaluation of the different terms that compose the binding energy taking into account different time points. For example, it is possible to estimate the binding energy as the sum of the molecular mechanical energies in the gas phase, the solvation contribute, evaluated using an implicit solvent model like the Generalized Born or solving the Poisson-Boltzman equations, and the entropic contribute, estimated with normal mode analysis approximation.

Moreover, Molecular Dynamics can be used to predict protein structures, ab-initio or refining models computed by homology, or to analyse protein stability, for example, verifying what happens in case of mutations. The simulation of proteins can be also very useful to verify the interactions of residues within a macromolecule, for example, to clarify why the binding of certain nucleotides (such as ATP) can change the structure of a particular binding site, a phenomenon that is usually referred as allostery.

The possibility of using NVIDIA cards for Molecular Dynamics in computational chemistry and biology propelled researches to new boundaries of discovery, enabling its application in wider range of situations. Compared to CPUs, GPUs run common molecular dynamics, quantum chemistry and visualization applications more than $5 \times$ faster. In particular, the team of AMBER has worked very hard to improve the performance of their simulator on GPUs, which is now extremely fast, between $5 \times$ and $10 \times$, depending on the number of atoms, the composition of the system and the type of simulation desired.

(Amber on GPUs, 2014). Also GROMACS7 has been ported on GPUs (GROMACS, 2012), with very good performance when the implicit solvent is used, while performance are less brilliant in case of explicit solvent.

XeonPhi Computing

Relying on Intel's Many Integrated Core (MIC) x86-based architecture, Intel Xeon Phi coprocessors provide up to 61 cores and 1.2 Teraflops of performance. These devices equip the second supercomputer of the TOP500 list (November 2016), Tianhe-2. In terms of usability, there are two ways an application can use an Intel Xeon Phi: in offload mode or in native mode. In offload mode the main application is running on the host, and it only offloads selected (highly parallel, computationally intensive) work to the coprocessor. In native mode the application runs independently, on the Xeon Phi only, and can communicate with the main processor or other coprocessors through the system bus.

The performance of these devices heavily depends on how well the application fits the parallelization paradigm of the Xeon Phi and in relation to the optimizations that are performed. In fact, since the processors on the Xeon Phi have a lower clock frequency with respect to the common Intel processor units (such as, e.g., the Sandy Bridge), applications that have long sequential algorithmic parts are absolutely not suitable for the native mode. On the other hand, even if the programming paradigm of these devices is standard C/C++ , which makes their use simpler with respect to the necessity of exploiting a different programming language such as CUDA, in order to achieve good performance, the code must be heavily optimized to fit the characteristics of the coprocessor (i.e., exploiting optimizations introduced by the Intel compiler and the MKL library).

Looking at the performance tests released by Intel (2015), the baseline improvement of supporting two Intel Sandy Bridge by offloading the heavy parallel computational to an Intel Xeon Phi gives an average improvement of 1.5 in the scalability of the application that can reach up to 4.5 of gain after a strong optimization of the code. For example, considering typical tools for bioinformatic sequence analysis: BWA (Burrows-Wheeler Alignment) (Li and Durbin, 2009) reached a baseline improvement of 1.86 and HMMER of 1.56 (Finn *et al.*, 2011). With a basic recompilation of Blastn for the Intel Xeon Phi (Altschul *et al.*, 1990) there is an improvement of 1.3, which reaches 4.5 after some modifications to the code in order to improve the parallelization approach. Same scalability figures for Abyss, which scales 1.24 with a basic porting and 4.2 with optimizations in the distribution of the computational load. Really good performance is achieved for Bowtie, which improves the code passing from a scalability of 1.3–18.4.

Clearly, the real competitors of the Intel Xeon Phi are the GPU devices. At the moment, the comparison between the best devices provided by Intel (Xeon Phi 7100) and Nvidia (Tesla K40) shows that the GPU is on average 30% more performing (Fang *et al.*, 2014), but the situation can vary in the future.

Low Power Devices

Over the recent years, energy efficiency has become a first order concern in the high performance computing sector. While high-end processors are rapidly shifting toward power-efficient technologies, the newer Systems-on-Chip (SoC), designed to meet the requirements of the mobile and embedded market, are gaining the interest of the scientific community for their increasing computing performances.

In addition, we should not underestimate the appealing low cost and low power consumption of SoCs. These novel hardware platforms are integrated circuits typically composed of low power multi-core processors combined with a small graphics-processing unit so, in principle, it is possible to run scientific applications on SoCs, with the aim of improving energy-to-performance ratios. However, such devices present a number of limitations for realistic scientific workloads, ranging from their 32-bit still present in some models to their small caches and RAM sizes.

There are examples of bioinformatic applications developed on SoCs architectures, although the diffusion of such cards is still limited and most of the available software is mainly demonstrative. For example, in the context of computational biology, a tool has been developed for the analysis of microRNA target (Morganti *et al.*, 2017), while in the context of systems biology a simulator has been developed to model reaction-diffusion systems (Beretta *et al.*, 2017).

Intriguingly, low-power architectures can be used to build portable bioinformatic applications for supporting portable sequencing machine such as the Oxford Nanopore Minion (Jain *et al.*, 2016). This will lead to the direct analysis of genomes of humans, animals or plants in remote regions of the world, or to analyse the composition of the microbioma in air-filters, water or soil samples in a simple and portable way.

Supercomputing

Many of the supercomputers in the TOP500 list are heavily involved in computational biology research. For example, Titan, one of the fastest system in the TOP500 list of November 2016, works for providing a molecular description of membrane fusion, one of the most common ways for molecules to enter or exit from living cells. Looking at the top supercomputers of this latest TOP500 list, the SuperMUC cluster, installed at the Leibniz Supercomputer Centre in Monaco, is often employed in Bioinformatics, for example, in analysis of linkage disequilibrium and genotyping, while Piz Daint, installed at the CSCS/Swiss Bioinformatics Institute in Lugano, has been successfully employed for a challenge of evolutionary genomics, for quickly calculating selection events in genes as consequence of mutations.

Looking at the November 2016 list, it is very interesting to see that two of the top three supercomputers in the world make use of co-processors to improve their performance. In particular, Tianhe-2 has more than 16,000 nodes, organized in unit of two Processor Intel Xeon E5 and three Coprocessor Intel Xeon Phi 31S1, while Titan uses NVIDIA K20 cards to improve its performance. Notably, in the top ten supercomputers, four make use of co-processors to enhance their performance. This choice should also be analysed in the view of the power consumption of these supercomputers: Tianhe-2 has a declared power consumption of 17 GW for a total of 33 PetaFLOPS, while Titan has a power consumption of 8 W for a total of 17 PetaFLOPS. For example, the K supercomputer, installed at the Riken Institute of Japan, has a power consumption of 12 GW for 10.5 PetaFLOPS. The possibility of saving energy using co-processors is therefore clear.

Conclusions

Omic sciences are able to produce, with the modern high-throughput techniques of analytical chemistry and molecular biology a huge amount of data. Next generation sequencing analysis of diseased somatic cells, novel bioactive compounds discovery and design, genome wide identification of regulatory elements and bio-markers, systems biology studies of biochemical pathways and gene networks are examples of the great advantages that high-performance computing can provide to omic sciences, accelerating a fast translation of bio-molecular results to the clinical practice.

Massive parallel clusters and supercomputers have huge capabilities, but their use by clinical and healthcare experts can be difficult. On the other hand, on-chip supercomputing such as GPU and Xeon Phi devices can represent good solutions to run custom bioinformatic algorithms, at least in institutions that perform routinely analysis. Virtual clusters on cloud can be a good trade-off between the necessity of computer power to run bioinformatic applications and the flexibility required to deal with heterogeneous data. Although this solution can be costly, a careful mix of on-demand resources and spot instances can be the key to face bioinformatic computational problems in the years to come.

See also: Computing for Bioinformatics. Dedicated Bioinformatics Analysis Hardware. Text Mining Applications

References

- Alces Flight, 2016. Effortless HPC is born (finally). Available at: <http://alces-flight.com/>.
- Alfieri, R., Merelli, I., Mosca, E., *et al.*, 2008. The cell cycle DB: A systems biology approach to cell cycle analysis. *Nucleic Acids Research* 36 (Suppl. 1), D641–D645.
- Alonso, H., Bliznyuk, A.A., Gready, J.E., 2006. Combining docking and molecular dynamic simulations in drug design. *Medicinal Research Reviews* 26 (5), 531–568.
- Altschul, S.F., Gish, W., Miller, W., *et al.*, 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3), 403–410.
- Amber on GPUs, 2014. Amber 16 GPU, acceleration support. Available at: <http://ambermd.org/gpus/benchmarks.htm>.
- Beretta, S., Morganti, L., Corni, E., *et al.*, 2017. Low-power architectures for miRNA-target genome wide analysis. In: *Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, pp. 309–312.
- Chiappori, F., Mattiazzi, L., Milanese, L., *et al.*, 2016. A novel molecular dynamics approach to evaluate the effect of phosphorylation on multimeric protein interface: The $\alpha\beta$ -Crystallin case study. *BMC Bioinformatics* 17 (4), 57.
- Chiappori, F., Merelli, I., Milanese, L., *et al.*, 2013. Static and dynamic interactions between GALK enzyme and known inhibitors: Guidelines to design new drugs for galactosemic patients. *European Journal of Medicinal Chemistry* 63, 423–434.
- Chiappori, F., Pucciarelli, S., Merelli, I., *et al.*, 2012. Structural thermal adaptation of β tubulins from the Antarctic psychrophilic protozoan *Euplotes focardii*. *Proteins: Structure, Function, and Bioinformatics* 80 (4), 1154–1166.
- Church, G.M., 2006. Genomes for all. *Scientific American* 294 (1), 46–54.
- Compeau, P.E., Pevzner, P.A., Tesler, G., 2011. How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology* 29 (11), 987–991.
- D'Agostino, D., Clematis, A., Quarati, A., *et al.*, 2013. Cloud infrastructures for in silico drug discovery: Economic and practical aspects. *BioMed Research International*, 138012.
- Dagum, L., Menon, R., 1998. OpenMP: An industry standard API for shared-memory programming. *IEEE Computational Science and Engineering* 5 (1), 46–55.
- de Magalhães, J.P., Finch, C.E., Janssens, G., 2010. Next-generation sequencing in aging research: Emerging applications, problems, pitfalls and possible solutions. *Ageing Research Reviews* 9 (3), 315–323.
- Fang, J., Sips, H., Zhang, L., *et al.*, 2014. Test-driving intel xeon phi. In: *Proceedings of the 5th ACM/SPEC international conference on Performance engineering*. ACM, pp. 137–148.
- Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Research*. gkr367.
- Foster, I., 1995. *Designing and Building Parallel Programs*. vol. 191. Reading, PA: Addison Wesley Publishing Company.
- Friesner, R.A., Murphy, R.B., Repasky, M.P., *et al.*, 2006. Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein ligand complexes. *Journal of Medicinal Chemistry* 49 (21), 6177–6196.
- Fuller, J.C., Khoeiri, P., Dinkel, H., *et al.*, 2013. Biggest challenges in bioinformatics. *EMBO Reports* 14 (4), 302–304.
- Garland, M., Kirk, D.B., 2010. Understanding throughput-oriented architectures. *Communications of the ACM* 53 (11), 58–66.
- Garland, M., Le Grand, S., Nickolls, J., *et al.*, 2008. Parallel computing experiences with CUDA. *IEEE Micro* 28 (4), 29–38.
- GROMACS, 2012. The GROMACS website. Available at: <http://www.gromacs.org/>
- Gropp, W., Lusk, E., Skjellum, A., 1999. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. vol. 1. MIT press.
- Huey, R., Morris, G.M., Olson, A.J., Goodsell, D.S., 2007. A semiempirical free energy force field with charge-based desolvation. *Journal of Computational Chemistry* 28 (6), 1145–1152.
- Intel, 2015. The Intel Xeon Phi coprocessor performance. Available at: <http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.html>
- Jain, M., Olsen, H.E., Paten, B., *et al.*, 2016. The oxford nanopore MinION: Delivery of nanopore sequencing to the genomics community. *Genome Biology* 17 (1), 239.
- Khronos Group, 2014. The Open Computing Language standard. Available at: <https://www.khronos.org/OpenGL/>

- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with BurrowsWheeler transform. *Bioinformatics* 25 (14), 1754–1760.
- McBurney, D., Sleep, M., 1987. Transputer-based experiments with the ZAPP architecture. *PARLE Parallel Architectures and Languages Europe*. Berlin/Heidelberg: Springer, pp. 242–259.
- Merelli, I., Calabria, A., Cozzi, P., *et al.*, 2013. SNPranker 2.0: A gene-centric data mining tool for diseases associated SNP prioritization in GWAS. *BMC Bioinformatics* 14 (1), S9.
- Merelli, I., Cozzi, P., D'Agostino, D., *et al.*, 2011. Image-based surface matching algorithm oriented to structural biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 8 (4), 1004–1016.
- Merelli, I., Pérez-Sánchez, H., Gesing, S., *et al.*, 2014. Latest advances in distributed, parallel, and graphic processing unit accelerated approaches to computational biology. *Concurrency and Computation: Practice and Experience* 26 (10), 1699–1704.
- Morganti, L., Corni, E., Ferraro, A., *et al.*, 2017. Implementing a space-aware stochastic simulator on low-power architectures: A systems biology case study. In: *Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, pp. 303–308.
- Morris, G.M., Huey, R., Lindstrom, W., *et al.*, 2009. Autodock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry* 16, 2785–2791.
- Mukherjee, S., Balias, T.E., Rizzo, R.C., 2010. Docking validation resources: Protein family and ligand flexibility experiments. *Journal of Chemical Information and Modeling* 50 (11), 1986–2000.
- Nickolls, J., Buck, I., Garland, M., *et al.*, 2008. Scalable parallel programming with CUDA. *Queue* 6 (2), 40–53.
- NVIDIA, 2016. GPU applications for bioinformatics and life sciences. Available at: http://www.nvidia.com/object/bio_info_life_sciences.html.
- Odersky, M., Altherr, P., Cremet, V., *et al.*, 2004. An overview of the Scala programming language (No. LAMP-REPORT-2004-006).
- Pérez-Sánchez, H., Fassihi, A., Cecilia, J.M., *et al.*, 2015. Applications of high performance computing in bioinformatics, computational biology and computational chemistry. In: *International Conference on Bioinformatics and Biomedical Engineering*. Switzerland: Springer International Publishing, pp. 527–541.
- Pheatt, C., 2008. Intel threading building blocks. *Journal of Computing Sciences in Colleges* 23 (4), 298.
- Rencuzogullari, U., Dwardadas, S., 2001. Dynamic adaptation to available resources for parallel computing in an autonomous network of workstations. *ACM SIGPLAN Notices* 36 (7), 72–81.
- Simpson, J.T., Wong, K., Jackman, S.D., *et al.*, 2009. ABySS: A parallel assembler for short read sequence data. *Genome Research* 19 (6), 1117–1123.

Models and Languages for High-Performance Computing

Domenico Talia, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Programming models, languages, and frameworks for parallel computers are required tools for designing and implementing high performance applications on scalable architectures. During recent years, parallel computers ranging from tens to hundred thousands processors became commercially available. They are gaining recognition as powerful instruments in scientific research, information management, and engineering applications. This trend is driven by parallel programming languages and tools that make parallel computers useful in supporting a wide range of applications, from scientific computing to business intelligence.

Parallel programming languages (called also *concurrent languages*) permit the design of parallel algorithms as a set of concurrent actions mapped onto different computing elements (Skillicorn and Talia, 1994). The cooperation between two or more actions can be performed in many ways according to the selected language. The design of programming languages and software tools for high-performance computing is crucial for the large dissemination and efficient utilization of these novel architectures (Skillicorn and Talia, 1998). High-level languages decrease both the design time and the execution time of parallel applications, and make it easier for new users to approach parallel computers.

Several issues must be solved when a parallel program is to be designed and implemented. Many of these are questions specifically related to the nature of parallel computation, such as process structuring, communication, synchronization, deadlock, process-to-processor mapping and distributed termination. For solving these questions in the design of an efficient parallel application it is important to use a programming methodology that helps a designer/programmer in all the stages of parallel software development. A parallel programming methodology must address the following main issues:

1. *Parallel process structuring*: how to decompose a problem in a set of parallel actions;
2. *Inter-process communication and synchronization*: how the parallel actions cooperate to solve a problem;
3. *Global computation design and evaluation*: how to see globally at the parallel program (as a whole) to improve its structure and evaluate its computational costs;
4. *Process-to-processor mapping*: how to assign the processes that compose a program to the processors that compose a parallel computer.

Parallel programming languages offer a user a support in all the phases of the parallel program development process. They provide constructs, mechanisms, and techniques that support a methodology for parallel software design that addresses the problems listed above. Although, early parallel languages and more recent low-level tools do not provide good solutions for all the mentioned problems, in the recent years significant high-level languages and environments have been developed. They can be used in all or many phases of the parallel software development process to improve scalability and portability of applications.

We discuss here representative languages and tools designed to support different models of parallelism and analyze both languages currently used to develop parallel applications in many areas, from numerical to symbolic computing, and novel parallel programming languages that will be used to program parallel computers in a near future.

Shared Memory Languages

Parallel languages of this class use the shared-memory model that is implemented by parallel machines composed by several processors that share the main memory. The concept of shared memory is a useful way to separate program control flow issues from data mapping, communication, and synchronization issues. Physical shared memory is probably difficult to provide on massively parallel architectures, but it is a useful abstraction, even if the implementation it hides is distributed. Significant parallel languages and environments based on the shared-memory model are OpenCL, Linda, OpenMP, Java, Pthreads, Opus, SHMEM, ThreadMarks, and Ease.

One way to make programming easier is to use techniques adapted from operating systems to enclose accesses to shared data in critical sections. Thus a single access to each shared variable is guaranteed at a given time. Another approach is to provide a high-level abstraction of shared memory. One way to do this is called *virtual shared memory* or *distributed shared memory*. In this case, the programming languages present a view of memory as if it is shared, but the implementation may or may not be. The goal of such approaches is to emulate shared memory well enough that the same number of messages travel around the system when a program executes as would have travelled if the program had been written to pass messages explicitly. In other words, the emulation of shared memory imposes no extra message traffic. Examples of these models and systems are Linda, Threadmarks, SHMEM, and Munin.

These systems emulate shared memory on distributed memory hardware by extending techniques for cache coherence in multiprocessors to software memory coherence. This involves weakening the implementation semantics of coherence as much as possible to make the problem tractable, and then managing memory units at the operating system level. Munin is a system that

supports different consistency models in the implementation of distributed shared memory (Bennett et al., 1990). It implements a type-specific memory coherence scheme that uses different coherence models. Thus, in Munin, each shared data object is supported by a memory coherence mechanism appropriate to the way in which the object is accessed. The other way is to build a programming model based on a useful set of sharing primitives for implementing shared data accessed through user-defined operations. This is the approach used by the Orca language.

Here we describe the main features of two shared-memory parallel languages: OpenMP and Java.

OpenMP

OpenMP is a library (*application program interface* or *API*) that supports parallel programming on shared memory parallel computers (OpenMP Consortium, 2002). OpenMP has been developed by a consortium of vendors of parallel computers (DEC, HP, SGI, Sun, Intel, etc.) with the aim to have a standard programming interface for parallel *shared-memory machines* (like PVM and MPI for distributed memory machines).

The OpenMP functions can be used inside Fortran, C and C++ programs. They allow for the parallel execution of code (*parallel DO loop*), the definition of shared data (*SHARED*), and synchronization of processes. OpenMP allows a user to.

- define regions of parallel code (*PARALLEL*) where it is possible to use local (*PRIVATE*) and shared variables (*SHARED*);
- synchronize processes by the definition of critical sections (*CRITICAL*) for shared variables (*SHARED*);
- define synchronization points (*BARRIER*).

A standard OpenMP program begins its execution as a single task, but when a *PARALLEL* construct is encountered, a set of processes are spawned to execute the corresponding parallel region of code. Each process is assigned with an iteration. When the execution of a parallel region ends, the results are used to update the data of the original process, which then resume its execution. From this operational way, it could be deduced that support for general task parallelism is not included in the OpenMP specification. Moreover, constructs or directives for data distribution control are absent from the current releases of OpenMP. This parallel programming library solves portability of code across different shared-memory architectures, however it does not offer a high-level programming level for parallel software development.

Java

Java is a language that is popular because of its connection with platform-independent software delivery on the Web (Lea, 2000). However, it is an object-oriented language that embodies also a shared-memory parallel programming model. Java supports the implementation of concurrent programs by process (called *threads*) creation (*new*) and execution (*start*). For example, the following Java instructions create three processes:

```
new proc (arg1a, arg1b, ..) ;
new proc (arg2a, arg2b, ..) ;
new proc (arg3a, arg3b, ..) ;
```

where *proc* is an object of the *Thread* class.

Java *threads* communicate and synchronize through *condition* variables. Shared variables are accessed from within *synchronized* methods. Java programs execute *synchronized* methods in a mutually exclusive way generating a critical section by associating a lock with each object that has synchronized methods. *Wait* and *notify* constructs have been defined to handle locks. The *wait* operation allows a thread to relinquish its lock and wait for notification of a given event. The *notify* operation allows a thread to signal the occurrence of an event to another thread that is waiting for that specific event. However, the *notify* and *wait* operations must be explicitly invoked within critical sections, rather than being automatically associated with section entry and exit as occurs in the *monitor* construct proposed about two decades ago. Thus a programmer must be careful to avoid deadlock occurrence among Java threads; the language offers no support for deadlock detection.

The shared-memory programming model has been defined for using Java on a sequential computer (*pseudo-parallelism*) or on shared-memory parallel computers. However, although the concurrent model defined in Java is based on the shared-memory model, Java was mainly designed to implement software on the Internet. Therefore it must be able to support the implementation of distributed programs on computer networks. To use Java on such platforms or on distributed-memory parallel computers there are different approaches presented in section "Object-Oriented Parallel Languages".

Distributed Memory Languages

A parallel program in a distributed-memory parallel computer (*multicomputer*) is composed of several processes that cooperate by exchanging data, e.g. by using messages. The processes might be executed on different processing elements of the multicomputer. In this environment, a high-level distributed concurrent programming language offers an abstraction level in which resources are defined like abstract data types encapsulated into cooperating processes. This approach reflects the model of distributed memory architectures composed of a set of processors connected by a communication network.

This section discusses imperative languages for distributed programming. Other approaches are available such as logic, functional, and object-oriented languages. This last class is discussed in Section "Object-Oriented Parallel Languages". Parallelism

in imperative languages is generally expressed at the level of processes composed of a list of statements. We included here both languages based on control parallelism and languages based on data parallelism. Control parallel languages use different mechanisms for process creation (e.g., *fork/join*, *par*, *spawn*) and process cooperation and communication (e.g., *send/receive*, *rendez-vous*, *remote procedure call*). On the other hand, data parallel languages use an implicit approach in solving these issues. Thus their run-time systems implement process creation and cooperation transparently to users. For this reason, data parallel languages are easier to be used, although they do not allow programmers to define arbitrary computation forms as occurs with control parallel languages.

Distributed memory languages and tools are: Ada, CSP, Occam, Concurrent C, CILK, HPF, MPI, C*, and Map-Reduce. Some of these are complete languages while others are APIs, libraries or *toolkits* that are used inside sequential languages. We discuss here some of them, in particular we focus on the most used languages for scientific applications, such as MPI and HPF.

MPI

The *Message Passing Interface* or MPI (Snir et al., 1996) is a de-facto standard message-passing interface for parallel applications defined since 1992 by a forum with a participation of over 40 organizations. MPI-1 was the first version of this message passing library that has been extended in 1997 by MPI-2 and in 2012 by MPI-3. MPI-1 provides a *rich set* of messaging primitives (129), including point-to-point communication, broadcasting, barrier, reduce, and the ability to collect processes in groups and communicate only within each group. MPI has been implemented on massively parallel computers, workstation networks, PCs, etc., so MPI programs are portable on a very large set of parallel and sequential architectures.

An MPI parallel program is composed of a set of similar processes running on different processors that use MPI functions for message passing. A single MPI process can be executed on each processor of a parallel computer and, according the SPMD (Single Program Multiple Data) model, all the MPI processes that compose a parallel program execute the same code on different data. Examples of MPI point-to-point communication primitives are.

```
MPI_Send (msg, leng, type,..., tag, MPI_COM);
MPI_Recv (msg, leng, type,0, tag, MPI_COM, &st);
```

Group communication is implemented by the primitives:

```
MPI_Bcast (inbuf, incnt, intype, root, comm);
MPI_Gather (outbuf, outcnt, outtype, inbuf, incnt,...);
MPI_Reduce (inbuf, outbuf, count, typ, op, root,...);
```

For program initialization and termination the *MPI_init* and *MPI_Finalize* functions are used. MPI offers a low-level programming model, but it is widely used for its portability and its efficiency. It is the case to mention that MPI-1 does not make any provision for process creation. However, in the MPI-2 and MPI-3 versions additional features have been provided for the implementation of.

- active messages,
- process startup, and
- dynamic process creation.

MPI is becoming more and more the first programming tool for message-passing parallel computers. However, it should be used as an *Esperanto* for programming portable system-oriented software rather than for end-user parallel applications where higher level languages could simplify the programmer task in comparison with MPI.

HPF

Differently from the previous toolkits, *High Performance Fortran* or HPF is a complete parallel language (Loveman, 1993). HPF is the result of an industry/academia/user effort to define a de facto consensus on language extensions for Fortran-90 to improve data locality, especially for distributed-memory parallel computers. It is a language for programming computationally intensive scientific applications. A programmer writes the program in HPF using the Single Program Multiple Data (SPMD) style and provides information about desired data locality or distribution by annotating the code with *data-mapping directives*. Examples of data-mapping directives are *Align* and *Distribute*:

```
!HPF$ Distribute D2 (Block, Block).
!HPF$ Align A(I,J) with B(I+2, J+2).
```

An HPF program is compiled by an architecture-specific compiler. The compiler generates the appropriate code optimized for the selected architecture. According to this approach, HPF could be used also on shared-memory parallel computers. HPF is based on exploitation of *loop parallelism*. Iterations of the loop body that are conceptually independent can be executed concurrently. For example, in the following loop the operations on the different elements of the matrix A are executed in parallel.

```
ForAll (I = 1: N, J = 1: M).
  A(I,J) = I * B(J).
End ForAll.
```

HPF must be considered as a high level parallel language because the programmer does not need to explicitly specify parallelism and process-to-process communication. The HPF compiler must be able to identify code that can be executed in parallel and

it implements inter-process communication. So HPF offers a higher programming level with respect to PVM or MPI. On the other hand, HPF does not allow for the exploitation of control parallelism and in some cases (e.g., irregular computations) the compiler is not able to identify all the parallelism that can be exploited in a parallel program, and thus it does not generate efficient code for parallel architectures.

Object-Oriented Parallel Languages

The parallel object-oriented paradigm is obtained by combining the parallelism concepts of process activation and communication with the object-oriented concepts of modularity, data abstraction and inheritance (Yonezawa *et al.*, 1987). An object is a unit that encapsulates private data and a set of associated operations or *methods* that manipulate the data and define the object behavior. The list of operations associated with an object is called its *class*. Object-oriented languages are mainly intended for structuring programs in a simple and modular way reflecting the structure of the problem to be solved.

Sequential object-oriented languages are based on a concept of passive objects. At any time, during the program execution only one object is active. An object becomes active when it receives a request (message) from another object. While the receiver is active, the sender is passive waiting for the result. After returning the result, the receiver becomes passive again and the sender continues. Examples of sequential object-oriented languages are Simula, Smalltalk, C++ , and Eiffel.

Objects and parallelism can be nicely integrated since object modularity makes them a natural unit for parallel execution. Parallelism in object-oriented languages can be exploited in two principal ways:

- using objects as the unit of parallelism assigning one or more processes to each object;
- defining processes as components of the language.

In the first approach languages are based on active objects. Each process is bound to a particular object for which it is created. When one process is assigned to an object, *inter-object* parallelism is exploited. If multiple processes execute concurrently within an object, *intra-object* parallelism is exploited also. When the object is destroyed the associated processes terminate.

In the latter approach two different kinds of entities are defined, objects and processes. A process is not bound to a single object, but it is used to perform all the operations required to satisfy an action. Therefore, a process can execute within many objects changing its address space when an invocation to another object is made.

Parallel object-oriented languages use one of these two approaches to support parallel execution of object-oriented programs. Examples of languages that adopted the first approach are ABCL/1, the Actor model, Charm++ , and Concurrent Aggregates (Chien and Dally, 1990). In particular, the Actor model is the best-known example of this approach. Although it is not a pure object-oriented model, we include the Actor model because it is tight related to object-oriented languages.

On the other hand, languages like HPC++ , Argus, Presto, Nexus, Scala, and Java use the second approach. In this case, languages provide mechanisms for creating and control multiple processes external to the object structure. Parallelism is implemented on top of the object organization and explicit constructs are defined to ensure object integrity.

HPC++

High Performance C++ (Diwan and Gannon, 1999) is a standard library for parallel programming based on the C++ language. The HPC++ consortium consists of people from research groups within Universities, Industry and Government Labs that aim to build a common foundation for constructing portable parallel applications as alternative to HPF. HPC++ is composed of two levels:

- Level 1 consists of a specification for a set of class libraries based on the C++ language.
- Level 2 provides the basic language extensions and runtime library needed to implement the full HPC++ .

There are two conventional modes of executing an HPC++ program. The first is *multi-threaded shared memory* where the program runs within one context. Parallelism comes from the parallel loops and the dynamic creation of threads. This model of programming is very well suited to modest levels of parallelism. The second mode of program execution is an *explicit SPMD model* where n copies of the same program are run on n different contexts. Parallelism comes from parallel execution of different tasks. This mode is well suited for massively parallel computers.

Distributed Programming in Java

Java is an object-oriented language that was born for distributed computing programming, although it embodies a shared-memory parallel programming model discussed in Section "Shared Memory Languages". To develop parallel distributed programs using Java, a programmer can use two main approaches:

- *sockets*: at the lowest programming level, Java provides a set of socket-based classes with methods (socket APIs) for inter-process communications using *datagram* and *stream* sockets. Java sockets classes offer a low-level programming interface that requires the user to specify inter-process communication details, however this approach offers an efficient communication layer.

- *RMI*: the Remote Method Invocation toolkit ([Sun Microsystems, 1997](#)) provides a set of mechanisms for communication among Java methods that reside on different computers having separate address spaces. The approach offers a user a higher programming layer that hides some inter-process communication details, but it is less efficient compared with the socket APIs.

In the latest years several efforts have been done to extend Java for high performance scientific applications. The most significant effort is represented by the *Java Grande* consortium that aimed at defining Java extensions for implementing computing intensive applications on high performance machines. The outcome of several research projects on the use of Java for distributed memory parallel computing are a set of languages and tools such as: *HPJava*, *MPIJava*, *JMPI*, *JavaSpace*, *jPVM*, *JavaPP*, and *JCSP*.

Composition Based Languages

This section describes novel models and languages for parallel programming that have properties that make them of interest. Some are not yet extensively used but they are very interesting high-level languages. The general trend that is observable in these languages, to those discussed in the previous sections, is that they are designed with stronger semantics directed towards software construction and correctness. There is also a general realization that the level of abstraction provided by a parallel programming language should be higher than was typical of languages designed in the past decade. This is no doubt partly due to the growing role of parallel computation as a methodology for solving problems in many application areas.

The languages described in this section can be divided into two main classes:

- Languages based on predefined structures that have been chosen to have good implementation properties or that are common in applications (*restricted computation forms* or *skeletons*);
- Languages that use a *compositional* approach – a complex parallel program can be written by composing simple parallel programs while preserving the original properties.

Skeletons are predefined parallel computation forms that embody both data- and control-parallelism. They abstract away from issues such as the number of processors in the target architecture, the decomposition of the computation into processes, and communication by supporting high-level *structured parallel programming*. However, using skeletons cannot be wrote arbitrary programs, but only those that are compositions of the predefined structures. The most used skeletons are *geometric*, *farm*, *divide&conquer*, and *pipeline*. Examples of skeleton languages are *SCL*, *SkieCL*, *BMF (Bird-Meertens formalism)*, *Gamma*, and *BSPlib*.

BSPlib

The Bulk Synchronous Parallel (BSP) model is not a pure skeleton model, but it represents a related approach in which the structured operations are single threads ([Skillicorn et al., 1997](#)). Computations are arranged in *supersteps*, each of which is a collection of these threads. A superstep does not begin until all of the global communications initiated in the previous superstep have completed (so that there is an implicit barrier synchronization at the end of each superstep). Computational results are available to all processes after each superstep. The *Oxford BSPlib* is a toolkit that implements the BSP operations in C and Fortran programs ([Hill et al., 1998](#)). Some examples of BSPlib functions are.

- *bsp_init(n)* to execute *n* processes in parallel,
- *bsp_sync* to synchronize processes,
- *bsp_push_reg* to make a local variable readable by other processes,
- *bsp_put* and *bsp_get* to get and put data to/from another process,
- *bsp_bcast* to send a data towards *n* processes.

The BSPlib provides automatic process mapping and communication patterns. Furthermore, it offers performance analysis tools to evaluate costs of BSP programs on different architectures. This feature is helpful in the evaluation of performance portability of parallel applications developed by BSP on parallel computers that are based on different computation and communication structures. BSPlib offers a minimal set of powerful constructs that make the programmer task easier than more complex models and, at the same time, it support costs estimation at compile time.

Significant examples of compositional languages are Program Composition Notation (PCN), Compositional C++, Seuss, and CYES-C++. PCN and Compositional C++, are second-generation parallel programming languages in the imperative style. Both are concerned with composing parallel programs to produce larger programs, while preserving the original properties.

Closing Remarks

Parallel programming languages support the implementation of high-performance applications in many areas: from computational science to artificial intelligence applications. However, programming parallel computers is more complex of programming sequential machines because in developing parallel software must be addressed several issues that are not present in sequential programming. Parallel processes structuring, communication, synchronization, process-to-processor mapping are problems that must be solved by parallel programming languages and tools to allows users to design complex parallel applications.

New models, tools and languages proposed and implemented in the latest years allow users to develop more complex programs with minor efforts. In the recent years we registered a trend from low-level languages towards more abstract languages to simplify the task of designers of parallel programs and several middle-level models have been designed as a trade off between abstraction and high performance. The research and development efforts presented here might bring parallel programming to the right direction for supporting a wider use of parallel computers. In particular, high-level languages will make the writing of parallel programs much easier than was possible previously and will help bring parallel systems into the general computer science community.

See also: Computing for Bioinformatics. Computing Languages for Bioinformatics: Java. Infrastructures for High-Performance Computing: Cloud Computing Development Environments. Infrastructures for High-Performance Computing: Cloud Infrastructures. MapReduce in Computational Biology via Hadoop and Spark. Parallel Architectures for Bioinformatics

References

- Bennett, J.K., Carter, J.B., Zwaenepoel, W., 1990. Munin: Distributed shared memory based on type-specific memory coherence. In: Proceedings of ACM PPOPP, pp. 168–176.
- Chien, A.A., Dally, W.J., 1990. Concurrent aggregates. In: Proceedings of 2nd SIGPLAN Symp. on Principles and Practices of Parallel Programming, pp. 187–196.
- Diwan, S., Gannon D., 1999. A capabilities based communication model for high-performance distributed applications: The Open HPC + + approach. In: Proceedings of IPPS/SPDP'99.
- Hill, J.M.D., *et al.*, 1998. BSPlib: The BSP programming library. *Parallel Computing* 24, 1947–1980.
- Lea, D., 2000. *Concurrent Programming in Java: Design Principles and Patterns*, 2nd ed. Addison-Wesley.
- Loveman, D.B., 1993. High Performance Fortran. *IEEE ParallelDistributed Technology*. 25–43.
- OpenMP Consortium, 2002. *OpenMP C and C + + Application Program Interface. Version 2.0*.
- Skillicorn, D.B., Hill, J.M.D., McColl, W.F., 1997. Questions and answers about BSP. *Scientific Programming* 6, 249–274.
- Skillicorn, D.B., Talia, D., 1994. *Programming Languages for Parallel Processing*. IEEE Computer Society Press.
- Skillicorn, D.B., Talia, D., 1998. Parallel programming models and languages. *ACM Computing Surveys* 30, 123–169.
- Snir, M., Otto, S.W., Huss-Lederman, S., Walker, D.W., Dongarra, J., 1996. *MPI: The Complete Reference*. The MIT Press.
- Sun Microsystems, 1997. *Java RMI Specification*.
- Yonezawa, A., *et al.*, 1987. *Object-Oriented Concurrent Programming*. MIT Press.

MapReduce in Computational Biology Via Hadoop and Spark

Giuseppe Cattaneo, University of Salerno, Fisciano, Italy

Raffaele Giancarlo, University of Palermo, Palermo, Italy

Umberto Ferraro Petrillo, University of Rome “Sapienza”, Rome, Italy

Gianluca Roscigno, University of Salerno, Fisciano, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the recent years there has been a huge increase in the number of genomic and proteomic sequences to be analyzed, mostly thanks to the sharp reduction of the sequencing costs made possible by NGS technologies. Unfortunately, computer hardware costs have not kept the same reduction pace, causing an economic problem to research areas that use sequencing, i.e., the entire Life Sciences. Those aspects are well summarized in [Kahn \(2011\)](#), [Mardis \(2010\)](#), where it is envisioned that the cost of analyzing sequence data may be a factor of 100 times more than the cost of its production. Apart from some particular cases, the Bioinformatics community is used to leverage on multi-processor architectures to accelerate the solution of computational intensive problems. Indeed, multi-processor calculators are inexpensive and do not require complex programming skills to be mastered. This is the case, e.g., of a simple task of counting the number of k -mers in a set of strings, a fundamental problem in Genomics and Epigenomics (see, e.g., [Compeau et al. \(2011\)](#), [Giancarlo et al. \(2015\)](#), [Utro et al. \(2015\)](#)), that is typically solved with the use of multi-processor architectures (see e.g., [Deorowicz et al. \(2015\)](#)). However, the ability of these architectures to timely analyze and solve decreasing for several reasons. Among these, the scalability of a multi-processor architecture is inherently limited by the (relative) small number of computing cores that can physically coexist on a same calculator. In addition, the computing parallelism achievable by a multi-processor architecture is heavily constrained by the intrinsic sequential nature of both the memory and the I/O subsystems. These problems can be overcome by resorting to Big Data distributed architectures, as these solutions allow for a virtual unbound scalability while requiring a programming approach easy to master. In order to foster further development of Bioinformatics and Computational Biology distributed software systems and platforms, here a short review of this area is provided. Moreover, we also identify some desirable properties that those software systems should have and provide a corresponding evaluation of them.

Specifically, Section MapReduce Fundamentals is dedicated to a short introduction to the architectural solutions available for the solution of Bioinformatics computational intensive problems and their potential issues. Then we focus on the recent approaches based on the usage of distributed systems for Big Data computing, with a particular emphasis on the MapReduce paradigm and its Hadoop and Spark implementations. In Section Systems and Applications: Description, Analysis and Assessment, we review a representative set of some of the most significant MapReduce implementations in bioinformatics, classified according to their application domain as well as to their ability to fully exploit the potential of the distributed system used to run them. In Section Closing Remarks, we focus on the alignment-free sequence comparison, by describing the first MapReduce solution to be proposed for this problem. In addition, we provide some insights on the engineering of this solution as well as an experimental assessment of its performance.

MapReduce Fundamentals

Bioinformatics problems can frequently be solved by decomposing them into simpler problems to be processed in parallel. This motivated the design of many algorithms able to exploit the several processing units available on multi-core shared-memory architectures for the effective execution of time-consuming bioinformatics tasks. Conversely, the scalability of this approach is inherently limited by some serious architectural bottlenecks, due to the competition between memory and I/O related resources. An algorithm can be considered *scalable* when its running time is inversely proportional to the number of computing units devoted to its execution.

The mentioned limits of multi-core shared-memory architectures is bypassed by resorting to *Distributed Systems*. They are architectures composed of a set of independent computers (also called *nodes*) that communicate over a network ([Tanenbaum and Van Steen, 2007](#)) and cooperate to solve a computational problem. Theoretically, with this approach, it is possible to assemble a system with an arbitrary computational capacity, as long as the corresponding number of nodes is available (i.e., *scale out*). The scalability of the system improves over multi-core shared-memory architectures as each node can access local resources without competing with other nodes.

However, such a theoretical advantage of distributed over multi-core comes at a price: the design and the development of a distributed algorithm are often a difficult task and involves complex algorithmic and programming skills. Fortunately, the adoption of one of the programming paradigms proposed in recent years for simplifying the transition toward distributed systems alleviates the mentioned problem. Among those, MapReduce ([Dean and Ghemawat, 2004](#)) paradigm is a *de facto* standard. It is described next, together with the two main middleware systems supporting it, i.e., [Apache Software Foundation \(2016a\)](#) and [Apache Software Foundation \(2016b\)](#).

MapReduce Paradigm

MapReduce (Dean and Ghemawat, 2008) is a paradigm for the processing of large amounts of data on a distributed computing infrastructure. Assuming the input data is organized as a set of $\langle \text{key}, \text{value} \rangle$ pairs, it is based on the definition of two functions. The *map* function processes an input $\langle \text{key}, \text{value} \rangle$ pair and returns a (possibly empty) intermediate set of $\langle \text{key}, \text{value} \rangle$ pairs. The *reduce* function merges all the intermediate values sharing the same key to form a (possibly smaller) set of values. These functions are run, as tasks, on the nodes of a distributed computing framework. All the activities related to the management of the lifecycle of these tasks as well as the collection of the map function results and their transmission to the reduce functions are transparently handled by the underlying framework (*implicit parallelism*), with no burden on the programmer.

Apache Hadoop

Apache Hadoop is the most popular framework supporting the MapReduce paradigm. It allows for the execution of distributed computations thanks to the interplay of two architectural components: *Yet Another Resource Negotiator* (YARN) (Vavilapalli et al., 2013) and *Hadoop Distributed File System* (HDFS) (Shvachko et al., 2010). YARN manages the lifecycle of a distributed application by keeping track of the resources available on a computing cluster and allocating them for the execution of application tasks modeled after one of the supported computing paradigms. HDFS is a distributed and block-structured file system designed to run on commodity hardware and able to provide fault tolerance through replication of data.

A basic Hadoop cluster is composed of a single *master node* and multiple *slave nodes*. The master node arbitrates the assignment of computational resources to applications to be run on the cluster and maintains an index of all the directories and the files stored in the HDFS distributed file system. Moreover, it tracks the slave nodes physically storing the data blocks making up these files. The slave nodes host a set of *workers* (also called *Containers*), in charge of running the map and reduce tasks of a MapReduce application, as well as using the local storage to maintain a subset of the HDFS data blocks.

One of the main characteristics of Hadoop is its ability to exploit data local computing. By this term, we mean the possibility to move applications closer to the data (rather than the vice-versa). This allows to greatly reduce network congestions and increase the overall throughput of the system when processing large amounts of data. Moreover, in order to reliably maintain files and to properly balance the load between different nodes of a cluster, large files are automatically split into smaller blocks, replicated and spread across different nodes.

Anatomy of a Hadoop MapReduce Application

MapReduce applications are run on the slave nodes of an Hadoop cluster in two consecutive (and potentially overlapping) phases: the map phase and the reduce phase. While in the map phase, a map task is executed for each distinct input pair by one of the workers running on the slave nodes of the Hadoop cluster. The output pairs generated during the execution of a map task are initially stored in a temporary memory buffer. When the task ends or when this buffer gets almost full, the output pairs it contains are sorted, and then saved on local disk in partitions for later processing (*spilling* step). When starting the reduce phase, output records generated during the previous spilling step are moved to the slave nodes designated for processing that partition (*shuffle* step). Once collected all records belonging to the same partition, these are sorted (*sort* step) and processed by a worker running the reduce task.

Apache Spark

Apache Spark is a new distributed framework used mainly to support programs with in-memory computing and acyclic data-flow model. In fact, Spark can be used for applications that reuse a working set of data across multiple parallel operations (e.g., iterative algorithms) and it allows the combination of streaming and batch processing (although Hadoop can be only used for batch applications). In addition, Spark is not limited to support only the MapReduce paradigm, although it preserves all the facilities of MapReduce-based frameworks, and it can have the Hadoop as the underlining middleware.

Spark has a *master/slaves* architecture where there are three main processes: *Worker*, *Executor* and *Driver*. A worker can run on any slave node, while an executor is a service launched for an application on a worker node that runs tasks and keeps data in memory or disk storage. The driver service, instead, runs on the master node and it manages the executors using a cluster resource manager (such as the default stand-alone cluster manager, Apache Mesos or YARN). In particular, the workers create executors for the driver, and then it can run tasks in those executors. In addition, each launched application has its own executors.

The *Resilient Distributed Dataset* (in short, RDD) is the main facility of Apache Spark, which is a read-only collection of objects partitioned across a set of nodes. This feature can be used to cache data in memory across nodes and they can be reused in iterative MapReduce runs. In addition, a RDD can provide many parallel operations, for example: *reduce*, *collect* and *foreach*. Spark also provides *Datasets* and *DataFrames* facilities. A Dataset is a distributed collection of data providing an interface that combines the benefits of RDDs with those of optimized execution engine of the query language. It can be constructed from objects and then manipulated using functional transformations (e.g., *map*, *flatMap*, *filter*). A DataFrame, instead, can be seen as a Dataset arranged in columns, as the concept of table in a relational database.

Finally, Spark application can be written in Java, Scala and Python language programming.

Systems and Applications: Description, Analysis and Assessment

Description

In this section, we report a classification of bioinformatics software tools developed according to the MapReduce paradigm. The vast majority of them runs on Hadoop and some on Spark. The proposed list does not aim to be exhaustive but to be representative of the most significant contributions proposed in this field. We also consider the particular case of an ad hoc framework expressly developed for the distributed genome analysis, i.e., GATK (McKenna *et al.*, 2010).

From the qualitative point of view, the properties that are desirable and that each of the considered software must have are the following.

- (a) **Programmability (PP)**. This property means that it is possible to instruct the software to solve an input problem using its own capabilities in a programmatic way. This property is held by any application whose functions can be recalled by means of a querying or a programming language. For example the Hadoop-based BioPig (Nordberg *et al.*, 2013) tool is programmable as it is possible to use its features by means of the Pig language to solve bioinformatics problems.
- (b) **Distributed Accelerator (DA)**. When facing problems that are easy to decompose, and provided that a sequential application/library already exist for solving those problems, it is possible to develop a distributed solution by just running on the nodes of a distributed system several instances of the sequential solution. In such cases, MapReduce is useful to schedule the execution of the several instances and to combine their output. Therefore, such a wrapping allows for an acceleration of the sequential package thanks to its distributed execution. For instance, CloudBLAST (Matsunaga *et al.*, 2008) is a distributed application that works by spanning multiple copies of a well-known existing application (namely BLAST (Altschul *et al.*, 1990)) on the nodes of a Hadoop cluster.
- (c) **Original Algorithm (OA)**. In this case, the proposed solution comes from the application of a new algorithm, developed according to the MapReduce paradigm to solve a classic or new bioinformatics problem. For instance, CloudAligner (Nguyen *et al.*, 2011) has been specifically designed according to the MapReduce paradigm and then implemented in Hadoop.
- (d) **Original Engineered Algorithm (OEA)**. This property holds if the MapReduce application is developed adopting an algorithm engineering approach (Cattaneo and Italiano, 1999; Demetrescu *et al.*, 2003) which, in addition to property (c), requires the outcoming code to be profiled and tuned so to allow for a full exploitation of the hardware resources of the system used to run the application.

The software can be divided into the following categories.

1. **Frameworks and support tools for the development of Bioinformatics applications**. We have included in this category software frameworks used to develop MapReduce pipelines and programs for bioinformatics (see Table 1), e.g., processing of HTS sequence data. In addition, we have included support tools and software libraries that high-level commodity functions useful for the fulfillment of auxiliary tasks in bioinformatics applications like managing files that have a standard format in bioinformatics, such as Hadoop-BAM (Niemenmaa *et al.*, 2012).
2. **Algorithms for Single Nucleotide Polymorphism Identification**. We have included in this category software that performs SNP identification and analysis (see Table 2).
3. **Gene Expression Analysis**. We have included in this category software for gene expression analysis (see Table 3), e.g., gene set analysis for biomarker identification.
4. **Sequence Comparison**. We have included in this category software for sequence comparison based on alignments and alignment-free methods (see Table 4).

Table 1 Frameworks and support tools for the development of Bioinformatics applications

Software	PP	Distributed accelerator	OA	OEA
ADAM (Nothaft <i>et al.</i> , 2015)	Yes	No	GE	GE
BioLinux (Krampis <i>et al.</i> , 2012)	SE	SE	NO	NO
BioPig (Nordberg <i>et al.</i> , 2013)	Yes	Yes	GE	NO
Biospark (Klein <i>et al.</i> , 2016)	Yes	No	ME	ME
Cloudgene (Schönherr <i>et al.</i> , 2012)	Yes	No	NO	NO
FASTdoop (Ferraro-Petrillo <i>et al.</i> , 2017)	Yes	No	GE	GE
GATK (McKenna <i>et al.</i> , 2010)	Yes	No	GE	NO
Hadoop-BAM (Niemenmaa <i>et al.</i> , 2012)	Yes	No	GE	NO
SeqInCloud (Mohamed <i>et al.</i> , 2013)	Yes	Yes	SE	ME
SeqPig (Schumacher <i>et al.</i> , 2014)	Yes	Yes	GE	NO
SparkSeq (Wiewiórka <i>et al.</i> , 2014)	Yes	No	SE	SE

Note: With reference to the main text, the software reported in this category is classified as shown. YES indicates that the property is held, while NO indicates otherwise. Other possible values indicate to what extent the property is held: GE stands for "to a great extent", ME stands for "to a moderate extent" and SE stands for "to a small extent." Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., *et al.*, 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science, vol. 708. Springer.

Table 2 Algorithms for Single Nucleotide Polymorphism Identification. This table is analogous to [Table 1](#)

Software	PP	Distributed accelerator	OA	OEA
BlueSNP (Huang <i>et al.</i> , 2013a,b)	Yes	Yes	NO	NO
Crossbow (Langmead <i>et al.</i> , 2009)	No	Yes	SE	NO
FALCO (Yang <i>et al.</i> , 2016)	No	Yes	NO	NO

Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., *et al.*, 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science*, vol. 708. Springer.

Table 3 Gene expression analysis. This table is analogous to [Table 1](#)

Software	PP	Distributed accelerator	OA	OEA
Eoulsan (Jourden <i>et al.</i> , 2012)	No	Yes	NO	NO
FX (Hong <i>et al.</i> , 2012)	No	No	GE	NO
MyRNA (Langmead <i>et al.</i> , 2010)	Yes	Yes	SE	SE
MRMSPolygraph (Kalyanaraman <i>et al.</i> , 2011)	No	No	NO	SE
YunBe (Zhang <i>et al.</i> , 2012)	No	No	GE	NO

Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., *et al.*, 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science*, vol. 708. Springer.

Table 4 Sequence comparison. This table is analogous to [Table 1](#)

Software	PP	Distributed accelerator	OA	OEA
Almeida <i>et al.</i> (Almeida <i>et al.</i> , 2012)	No	No	GE	NO
CloudBLAST (Matsunaga <i>et al.</i> , 2008)	No	Yes	NO	NO
CloudPhylo (Xu <i>et al.</i> , 2016)	No	No	ME	NO
HAFS (Cattaneo <i>et al.</i> , 2016)	Yes	No	GE	GE
HAlign (Zou <i>et al.</i> , 2015)	No	No	GE	ME
HBlast (O'Driscoll <i>et al.</i> , 2015)	No	Yes	NO	GE
K-mulus (Hill <i>et al.</i> , 2013)	No	Yes	NO	NO
MapReduce BLAST (Yang <i>et al.</i> , 2011)	No	No	ME	ME
Nephele (Colosimo <i>et al.</i> , 2011)	No	No	NO	NO
Strand (Drew and Hahsler, 2014)	No	No	GE	NO

Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., *et al.*, 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science*, vol. 708. Springer.

Table 5 Genome assembly. This table is analogous to [Table 1](#)

Software	PP	Distributed accelerator	OA	OEA
CloudBrush (Chang <i>et al.</i> , 2012)	No	No	GE	NO
Contrail (Schatz <i>et al.</i> , 2010)	No	Yes	GE	NO
Spaler (Abu-Doleh and Catalyjiirek, 2015)	No	No	ME	NO

Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., *et al.*, 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science*, vol. 708. Springer.

5. **Genome Assembly.** We have included in this category software for *de novo* genome assembly from short sequencing reads (see [Table 5](#)).
6. **Sequencing Reads Mapping.** We have included in this category software for mapping short reads to reference genomes (see [Table 6](#)).
7. **Additional Applications.** We have included in this category MapReduce bioinformatics applications for which there is only one implementation available (see [Table 7](#)).

Analysis and Assessment

As visible in [Tables 1–7](#) there is a large number of MapReduce distributed applications developed for solving bioinformatics problems, however, few of these have been expressly developed according to the MapReduce paradigm and even a fewer number has been engineered so to be able to take out the most from the underlying distributed systems where they are run.

Table 6 Sequencing reads mapping. This table is analogous to [Table 1](#)

Software	PP	Distributed accelerator	OA	OEA
BigBWA (Abuín et al., 2015)	No	Yes	NO	NO
BlastReduce (Schatz, 2008)	No	No	GE	NO
Bwasw-Cloud (Sun et al., 2014)	No	Yes	SE	SE
CloudAligner (Nguyen et al., 2011)	No	No	GE	NO
CloudBurst (Schatz, 2009)	No	No	GE	NO
DistMap (Pandey and Schlotterer, 2013)	No	Yes	NO	NO
Halvade (Decap et al., 2015)	No	Yes	GE	GE
MetaSpark (Zhou et al., 2017)	No	No	GE	NO
MRUniNovo (Li et al., 2016)	No	Yes	SE	NO
Mushtaq et al. (Mushtaq and Al-Ars, 2015)	No	Yes	GE	ME
Rail-RNA (Nellore et al., 2016)	No	Yes	SE	NO
SEAL (Pireddu et al., 2011)	No	Yes	NO	NO
SparkSW (Zhao et al., 2015)	No	No	GE	ME

Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., et al., 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science, vol. 708. Springer.

Table 7 Additional applications. This table is analogous to [Table 1](#)

Software	PP	Distributed accelerator	OA	OEA
BioDoo (Leo et al., 2009)	No	Yes	SE	SE
Codon counting (Radenski and Ehwerhemuepha, 2014)	No	No	GE	NO
GQL (Masseroli et al., 2015)	Yes	No	NO	NO
GRIMD (Piotto et al., 2014)	Yes	Yes	ME	ME
S-Chemo (Harnie et al., 2017)	No	Yes	SE	ME
MrMC-MinH (Rasheed and Rangwala, 2013)	No	No	GE	NO
MrsRF (Matthews and Williams, 2010)	No	No	GE	ME
PeakRanger (Feng et al., 2011)	No	No	GE	NO
VariantSpark (O'Brien et al., 2015)	No	No	ME	ME

Source: Modified and extended from Cattaneo, G., Giancarlo, R., Piotto, S., et al., 2017. MapReduce in computational biology – A synopsis. In: Rossi, F., Piotto, S., Concilio, S. (Eds.), Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry. WIVACE 2016. Communications in Computer and Information Science, vol. 708. Springer.

A Case Study: Alignment-Free Sequence Comparison

[Cattaneo et al. \(2016\)](#) present a Hadoop-based framework used to compute alignment-free sequence comparison using different dissimilarity measures, such as Squared Euclidean ([Vinga and Almeida, 2003](#); [Yang and Zhang, 2008](#)), based on exact k -mers counts. Given an input collection of sequences and a user-provided implementation of a dissimilarity measure, this framework operates in two subsequent steps.

Step 1: Indexing. For each input sequence, it extracts the k -mers that occur in that sequence and that are later used to compute the dissimilarity between sequences. The map function takes as an input a pair $\langle idSeq, S \rangle$, where $idSeq$ is a unique identifier for the input sequence S . Then, for each k -mer x present in S , it outputs the pair $\langle x, (idSeq, 1) \rangle$. At the end of that task, finally, each map function communicates to the reducers the length of the sequence via the pair $\langle idSeq, |S| \rangle$. The reduce function receives by the Hadoop framework a set of pairs $\langle x, L \rangle$, where L is the list of $(idSeq, 1)$ pairs corresponding to that particular k -mer, and outputs the number of times each k -mer x appears in every input sequence. That is encoded as a record $\langle x, L' \rangle$, where L' is the mentioned statistics. In addition, it returns also the size of each sequence it processed.

Step 2: Dissimilarity Measurement. Given as input a function implementing dissimilarity measure and the output of the indexing step, this step evaluates the pairwise dissimilarity for each pair of input sequences. Given a pair $\langle x, L' \rangle$, the map function computes the partial dissimilarity for each distinct pair of sequences in L' according to the input-provided measure D . As output, the map function emits a $\langle (idSeq_A, idSeq_B, D), pdiss \rangle$ pair, where $idSeq_A$ and $idSeq_B$ are the identifiers of two input sequences, while $pdiss$ is their partial dissimilarity. The reduce function receives by the Hadoop framework a set of pairs $\langle (idSeq_A, idSeq_B, D), list\{pdiss'\} \rangle$, where $list\{pdiss'\}$ is the list of all the partial dissimilarities among the sequences with identifiers $idSeq_A$ and $idSeq_B$, with respect to the measure D , and returns the pair $\langle (idSeq_A, idSeq_B, D), diss \rangle$, where $diss$ is the final value of D with respect to these two sequences.

Optimizations

Although being correct, the straightforward implementation of the Hadoop algorithm described in Section A Case Study: Alignment-Free Sequence Comparison suffered from some main performance issues when run on very long sequences. To alleviate some of these problems, the authors presented some significant optimizations:

Incremental in-mapper combining

The map tasks in the Indexing step use a hash table data structure to maintain a local index of the k -mers found while scanning an input sequence with their associated frequencies. Thanks to this optimization, the execution of a map task will produce just one single bulk output at the end of computation, including all the output pairs, rather than return a myriad of single pairs, one for each k -mer found, as required by the original implementation. As a further advantage, the overall number of output pairs of a map task will be significantly reduced because all occurrences of the same k -mer will be aggregated in a single output pair.

Input split strategies

The map tasks implement an optimized input split strategy able to manage very long multi-line sequences during the Indexing step. It works by splitting an input sequence S , encoded as a FASTA file, in several records $\langle id_S, (S_i, S_{i+1}) \rangle$, where id_S is an identifier for S , S_i is the i -th row of S and S_{i+1} contains the first $k-1$ characters of the $(i+1)$ -th row of S (S_{i+1} is empty if i is the last row of S). Differently from the standard Hadoop input strategies, this optimization allows each map task run during the Indexing step to extract autonomously all the k -mers that originate from a particular row of the input file thus allowing the processing of sequences of arbitrary length while exploiting the implicit parallelism available with Hadoop.

Experimental Analysis

The Hadoop-based framework presented in Section A Case Study: Alignment-Free Sequence Comparison has been subject to an experimental analysis aimed at assessing its ability to scale with the number of computing units devoted to its execution as well as its capacity to increase the span of the problems that can be solved with respect to the sequential approach. To this end, the authors used a dataset coming from meta-genomic studies, e.g., [Huang et al. \(2013a,b\)](#), and including all the sequenced microbial genomes (bacteria, archaea, viruses) available in public databases. It consists of 40,988 genomes, for a total of 172 GB.

The experiments have been conducted on a homogeneous cluster of 4 slave nodes and a master node, equipped each with 32 GB of RAM, 2 AMD Opteron @ 2.10 GHz processors, 1 TB disk drive and a Giga-Ethernet network card. A HDFS replication factor set to 2 and a block size set to 128 MB have been used.

The dataset has been processed by measuring the time required to evaluate the Squared Euclidean dissimilarity measure with $k=12$. Let S and Q be two sequences, this measure ([Vinga and Almeida, 2003](#); [Yang and Zhang, 2008](#)) evaluates their dissimilarity as:

$$d_{SE}(S, Q) = \sum_{i=1}^n (s_i - q_i)^2 \quad (1)$$

where s_i and q_i are the number of occurrences of the i -th k -mer in S and in Q , respectively.

During the tests, the authors measured the scalability of the proposed framework by increasing the number of CPU cores that are reserved to the cluster and, consequently, the number of map/reduce tasks that are run concurrently, while keeping fixed the input dataset. The outcoming performance have then been compared with those of the sequential version of the same algorithm. The overall results are reported in [Figs. 1 and 2](#), where it is also indicated the time spent for steps 1 and 2, respectively.

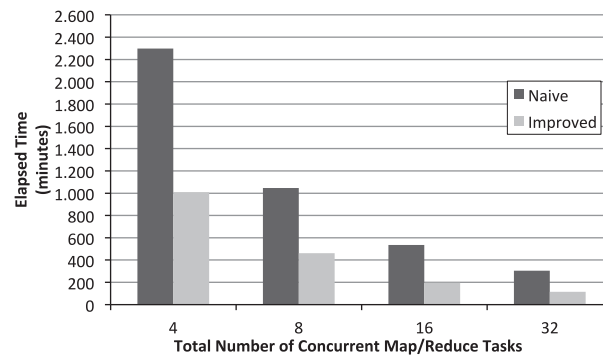


Fig. 1 Elapsed times for evaluating the Squared Euclidean dissimilarity measure among 60 sequences extracted from the dataset with a total size of $\approx 600,000,000$ characters, with $k=12$ and an increasing number of concurrent map/reduce tasks, using both the naive and improved versions of our distributed paradigm. Reproduced from Cattaneo, G., Ferraro Petrillo, U., Giancarlo, R., Roscigno, G., 2016. An effective extension of the applicability of alignment-free biological sequence comparison algorithms with Hadoop. *The Journal of Supercomputing* 1–17.

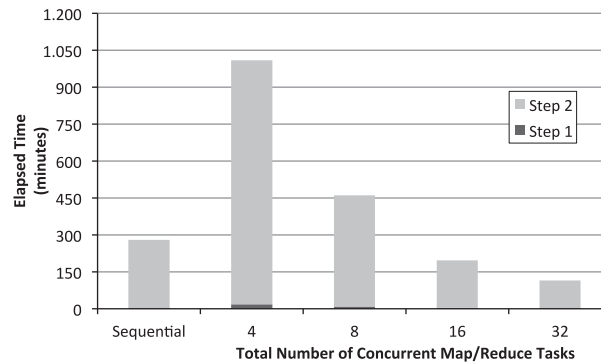


Fig. 2 Elapsed times for evaluating the Squared Euclidean dissimilarity measure among 60 sequences extracted from the dataset with a total size of $\approx 600.000.000$ characters, with $k=12$ and an increasing number of concurrent map/reduce tasks. For completeness, we also report the time of the sequential algorithm. Reproduced from Cattaneo, G., Ferraro Petrillo, U., Giancarlo, R., Roscigno, G., 2016. An effective extension of the applicability of alignment-free biological sequence comparison algorithms with Hadoop. *The Journal of Supercomputing* 1–17.

Results and Discussion

We have presented a review of some of the most significant MapReduce bioinformatics applications. The increasing number and the quality of the contributions proposed in this field show the arising interest toward the use of Distributed Computing, and of its popular MapReduce computing paradigm, for the solution of computational intensive bioinformatics problems. However, as witnessed by experiences like the ones documented in Cattaneo *et al.* (2016), Bertoni *et al.* (2015), an effective exploitation of the computation resources available with a distributed system can only be achieved through an accurate engineering activity of the MapReduce algorithms as well as a careful choice of the underlying software framework and support tools.

Closing Remarks

We have presented the state of the art regarding the use of Distributed Computing, in particular, software designed with MapReduce paradigm, in Bioinformatics and Computational Biology. Although such a body of work will grow in the future, it would be appropriate to follow good design and algorithm engineering approaches in order to use in full the available hardware and the scalability MapReduce offers.

Acknowledgement

We would like to thank the Department of Statistical Sciences of University of Rome – “La Sapienza”, for computing time on the TeraStat supercomputing cluster. Moreover, the authors are also grateful to the Consortium GARR for having made available a cutting-edge OpenStack Virtual Datacenter which will turn out to be instrumental in the continuation of this research that tries to establish the impact that MapReduce can have on bioinformatics research.

See also: Computational Pipelines and Workflows in Bioinformatics. Computing for Bioinformatics. Computing Languages for Bioinformatics: Java. Constructing Computational Pipelines. Models and Languages for High-Performance Computing. Parallel Architectures for Bioinformatics

References

- Abu-Doleh, A., Catalyürek, U.V., 2015. Spaler: Spark and GraphX based de novo genome assembler. In: *IEEE International Conference on Big Data (Big Data)*, 2015, IEEE, pp. 1013–1018.
- Abuin, J.M., Pichel, J.C., Pena, T.F., Amigo, J., 2015. BigBWA: Approaching the Burrows-Wheeler aligner to big data technologies. *Bioinformatics*. btv506.
- Almeida, J.S., Grieneberg, A., Maass, W., Vinga, S., 2012. Fractal MapReduce decomposition of sequence alignment. *Algorithms for Molecular Biology* 7, 1–12.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215, 403–410.
- Apache Software Foundation, 2016a. Hadoop. Available from: <http://hadoop.apache.org/>.
- Apache Software Foundation, 2016b. Spark. Available from: <http://spark.apache.org/>.
- Bertoni, M., Ceri, S., Kaitoua, A., Pinoli, P., 2015. Evaluating cloud frameworks on genomic applications. In: *2015 IEEE International Conference on Big Data, IEEE*, pp. 193–202.
- Cattaneo, G., Ferraro Petrillo, U., Giancarlo, R., Roscigno, G., 2016. An effective extension of the applicability of alignment-free biological sequence comparison algorithms with Hadoop. *The Journal of Supercomputing*. 1–17. <https://doi.org/10.1007/s11227-016-1835-3>.

- Cattaneo, G., Italiano, G.F., 1999. Algorithm engineering. *ACM Computing Surveys ((CSUR))* 31, 582–585.
- Chang, Y.-J., Chen, C.-L., Chen, C.-L., Ho, J.-M., 2012. A de novo next generation genomic sequence assembler based on string graph and MapReduce cloud computing framework. *BMC Genomics* 13, 1–17.
- Colosimo, M.E., Peterson, M.W., Mardis, S., Hirschman, L., 2011. Nephel: Genotyping via complete composition vectors and MapReduce. *Source Code for Biology and Medicine* 6, 1–10.
- Compeau, P.E.C., Pevzner, P.A., Tesler, G., 2011. How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology* 29, 987–991.
- Dean, J., Ghemawat, S., 2004. MapReduce: Simplified data processing on large clusters. *Operating Systems Design and Implementation*. 137–150.
- Dean, J., Ghemawat, S., 2008. MapReduce: Simplified data processing on large clusters. *Communications of the ACM* 51, 107–113.
- Decap, D., Reumers, J., Herzeel, C., Costanza, P., Fostier, J., 2015. Hal-vade: Scalable sequence analysis with MapReduce. *Bioinformatics*. btv179.
- Demetrescu, C., Finocchi, I., Italiano, G.F., 2003. Algorithm engineering. *Bulletin of the EATCS* 79, 48–63.
- Deorowicz, S., Kokot, M., Grabowski, S., Debudaj-Grabysz, A., 2015. KMC 2: Fast and resource-frugal k-mer counting. *Bioinformatics* 13 (10), 1569–1576.
- Drew, J., Hahsler, M., 2014. Strand: Fast sequence comparison using MapReduce and locality sensitive hashing. In: *Proceedings of the 5th ACM Conference on Bioinformatics Computational Biology, and Health Informatics*, ACM, pp. 506–513.
- Feng, X., Grossman, R., Stein, L., 2011. PeakRanger: A Cloud-enabled peak caller for ChIP-seq data. *BMC Bioinformatics* 12, 1–11.
- Ferraro-Petrillo, U., Roscigno, G., Cattaneo, G., Giancarlo, R., 2017. FASTdoop: A versatile and efficient library for the input of FASTA and FASTQ files for MapReduce Hadoop bioinformatics applications. *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/btx010>.
- Giancarlo, R., Rombo, S.E., Utro, F., 2015. Epigenomic k-mer dictionaries: Shedding light on how sequence composition influences in vivo nucleosome positioning. *Bioinformatics* 31 (18), 2939–2946.
- Harnie, D., Saey, M., Vapirev, A.E., *et al.*, 2017. Scaling machine learning for target prediction in drug discovery using Apache Spark. *Future Generation Computer Systems* 67, 409–417.
- Hill, C.M., Albach, C.H., Angel, S.G., Pop, M., 2013. K-mulus: Strategies for BLAST in the Cloud. In: *International Conference on Parallel Processing and Applied Mathematics*, Springer, pp. 237–246.
- Hong, D., Rhie, A., Park, S.-S., *et al.*, 2012. FX: An RNA-Seq analysis tool on the Cloud. *Bioinformatics* 28, 721–723.
- Huang, H., Tata, S., Prill, R.J., 2013a. BlueSNP: R package for highly scalable genome-wide association studies using Hadoop clusters. *Bioinformatics* 29, 135–136.
- Huang, K., Brady, A., Mahurkar, A., *et al.*, 2013b. MetaRef: A pan-genomic database for comparative and community microbial genomics. *Nucleic Acids Research* 42 (D1), 617–624.
- Jourden, L., Bernard, M., Dillies, M.-A., Le Crom, S., 2012. Eoulsan: A cloud computing-based framework facilitating high throughput sequencing analyses. *Bioinformatics* 28, 1542–1543.
- Kahn, S.D., 2011. On the future of genomic data. *Science* 331, 728–729.
- Kalyanaraman, A., Cannon, W.R., Latt, B., Baxter, D.J., 2011. MapRe-duce implementation of a hybrid spectral library-database search method for large-scale peptide identification. *Bioinformatics* 27 (21), 3072–3073.
- Klein, M., Sharma, R., Bohrer, C., Avelis, C., Roberts, E., 2016. Biospark: Scalable analysis of large numerical data sets from biological simulations and experiments using Hadoop and Spark. *Bioinformatics*. btw614.
- Krampis, K., Booth, T., Chapman, *et al.*, 2012. Cloud biolinux: Pre-configured and on-demand bioinformatics computing for the genomics community. *BMC Bioinformatics* 13 (1), 42. <https://doi.org/10.1186/1471-2105-13-42>.
- Langmead, B., Hansen, K.D., Leek, J.T., *et al.*, 2010. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biology* 11, 1–11.
- Langmead, B., Schatz, M.C., Lin, J., Pop, M., Salzberg, S.L., 2009. Searching for SNPs with cloud computing. *Genome Biology* 10, 1–10.
- Leo, S., Santoni, F., Zanetti, G., 2009. Biodoop: Bioinformatics on Hadoop. In: *International Conference on Parallel Processing Workshops, 2009 (ICPPW'09)*, IEEE, pp. 415–422.
- Li, C., Chen, T., He, Q., Zhu, Y., Li, K., 2016. MRUniNovo: An efficient tool for de novo peptide sequencing utilizing the Hadoop distributed computing framework. *Bioinformatics*. btw721.
- Mardis, E.R., 2010. The \$1,000 genome, the \$100,000 analysis? *Genome Medicine* 2 (1–3),
- Masseroli, M., Pinoli, P., Venco, F., *et al.*, 2015. GenoMetric query language: A novel approach to large-scale genomic data management. *Bioinformatics* 31 (12), 1881–1888.
- Matsunaga, A., Tsugawa, M., Fortes, J., 2008. CloudBLAST: Combining MapReduce and virtualization on distributed resources for bioinformatics applications. In: *IEEE Proceedings of the Fourth International Conference on eScience, eScience'08*, IEEE, pp. 222–229.
- Matthews, S.J., Williams, T.L., 2010. MrsRF: An efficient MapReduce algorithm for analyzing large collections of evolutionary trees. *BMC Bioinformatics* 11, 1–9.
- McKenna, A., Hanna, M., Banks, E., *et al.*, 2010. The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20, 1297–1303.
- Mohamed, N.M., Lin, H., Feng, W., 2013. Accelerating data-intensive genome analysis in the cloud. In: *Proceedings of the 5th International Conference on Bioinformatics and Computational Biology (BICoB)*, Honolulu, Hawaii.
- Mushtaq, H., Al-Ars, Z., 2015. Cluster-based Apache Spark implementation of the GATK DNA analysis pipeline. In: *IEEE International Conference on Bioinformatics and Biomedicine (BIBIM)*, IEEE, pp. 1471–1477.
- Nellore, A., Wilks, C., Hansen, K.D., Leek, J.T., Langmead, B., 2016. Rail-dbGaP: Analyzing dbgap-protected data in the cloud with amazon elastic MapReduce. *Bioinformatics* 32 (16), 2551–2553.
- Nguyen, T., Shi, W., Ruden, D., 2011. CloudAligner: A fast and full-featured MapReduce based tool for sequence mapping. *BMC Research Notes* 4, 171.
- Niemenmaa, M., Kallio, A., Schumacher, A., *et al.*, 2012. Hadoop-BAM: Directly manipulating next generation sequencing data in the Cloud. *Bioinformatics* 28, 876–877.
- Nordberg, H., Bhatia, K., Wang, K., Wang, Z., 2013. BioPig: A Hadoop-based analytic toolkit for large-scale sequence data. *Bioinformatics* 29, 3014–3019.
- Nothhaft, F.A., Massie, M., Danford, T., *et al.*, 2015. Rethinking data-intensive science using scalable analytics systems. In: *Proceedings of the 2015 ACM SIG-MOD International Conference on Management of Data*, ACM, pp. 631–646.
- O'Brien, A.R., Saunders, N.F., Guo, *et al.*, 2015. VariantSpark: Population scale clustering of genotype information. *BMC Genomics* 16 (1), 1052.
- O'Driscoll, A., Belogrudov, V., Carroll, *et al.*, 2015. HBLAST: Parallelised sequence similarity – a hadoop mapreducible basic local alignment search tool. *Journal of Biomedical Informatics* 54, 58–64.
- Pandey, R.V., Schlotterer, C., 2013. DistMap: A toolkit for distributed short read mapping on a hadoop cluster. *PLOS ONE* 8 (8), e72614.
- Piotto, S., Di Biasi, L., Concilio, S., Castiglione, A., Cattaneo, G., 2014. GRIMD: Distributed computing for chemists and biologists. *Bioinformatics* 10, 43–47.
- Pirreddi, L., Leo, S., Zanetti, G., 2011. SEAL: A distributed short read mapping and duplicate removal tool. *Bioinformatics* 27, 2159–2160.
- Radenski, A., Elwerhemuepha, L., 2014. Speeding-up codon analysis on the Cloud with local MapReduce aggregation. *Information Sciences* 263, 175–185.
- Rasheed, Z., Rangwala, H., 2013. A Map-Reduce framework for clustering metagenomes. *IEEE Proceedings of the 27th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, IEEE, pp. 549–558.
- Schatz, M.C., 2008. BlastReduce: High performance short read mapping with MapReduce. University of Maryland. Available from: <http://cgis.cs.umd.edu/Grad/scholarlypapers/papers/MichaelSchatz.pdf>.
- Schatz, M.C., 2009. CloudBurst: Highly sensitive read mapping with MapRe-duce. *Bioinformatics* 25, 1363–1369.
- Schatz, M.C., Sommer, D., Kelley, D., Pop, M., 2010. De novo assembly of large genomes using Cloud computing. In: *Proceedings of the Cold Spring Harbor Biology of Genomes Conference*.

- Schonherr, S., Forer, L., Weifiensteiner, *et al.*, 2012. Cloudgene: A graphical execution platform for MapReduce programs on private and public clouds. *BMC Bioinformatics* 13, 200.
- Schumacher, A., Pireddu, L., Niemenmaa, *et al.*, 2014. SeqPig: Simple and scalable scripting for large sequencing data sets in Hadoop. *Bioinformatics* 30, 119–120.
- Shvachko, K., Kuang, H., Radia, S., Chansler, R., 2010. The Hadoop distributed file system. In: *IEEE Proceedings of the 26th Symposium on Mass Storage Systems and Technologies*, IEEE Computer Society, Washington, DC, pp. 1–10.
- Sun, M., Zhou, X., Yang, F., Lu, K., Dai, D., 2014. Bwasw-Cloud: Efficient sequence alignment algorithm for two big data with MapReduce, In: *Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies*, IEEE, pp. 213–218.
- Tanenbaum, A.S., Van Steen, M., 2007. *Distributed systems*. Upper Saddle River, NJ: Prentice-Hall.
- Utro, F., Di Benedetto, V., Corona, D.F., Giancarlo, R., 2015. The intrinsic combinatorial organization and information theoretic content of a sequence are correlated to the DNA encoded nucleosome organization of eu-karyotic genomes. *Bioinformatics* 32 (6), 835–842.
- Vavilapalli, V.K., Murthy, A.C., Douglas, C., *et al.*, 2013. Apache Hadoop YARN: Yet another resource negotiator. In: *Proceedings of the 4th annual Symposium on Cloud Computing*, ACM, pp. 1–16.
- Vinga, S., Almeida, J., 2003. Alignment-free sequence comparison – A review. *Bioinformatics* 19, 513–523.
- Wiewiórka, M.S., Messina, A., Pacholewska, *et al.*, 2014. SparkSeq: Fast, scalable, cloud-ready tool for the interactive genomic data analysis with nucleotide precision. *Bioinformatics* 30, 2652–2653.
- Xu, X., Ji, Z., Zhang, Z., 2016. CloudPhylo: A fast and scalable tool for phylogeny reconstruction. *Bioinformatics*. btw645.
- Yang, A., Troup, M., Lin, P., Ho, J.W., 2016. Falco: A quick and flexible single-cell RNA-seq processing framework on the cloud. *Bioinformatics*. btw732.
- Yang, K., Zhang, L., 2008. Performance comparison between k-tuple distance and four model-based distances in phylogenetic tree reconstruction. *Nucleic Acids Research* 36, 1–9.
- Yang, X.-L., Liu, Y.-L., Yuan, C.-F., Huang, Y.-H., 2011. Parallelization of BLAST with MapReduce for long sequence alignment. In: *Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, IEEE, pp. 241–246.
- Zhang, L., Gu, S., Liu, Y., Wang, B., Azuaje, F., 2012. Gene set analysis in the cloud. *Bioinformatics* 28, 294–295.
- Zhao, G., Ling, C., Sun, D., 2015. SparkSW: Scalable distributed computing system for large-scale biological sequence alignment. In: *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, IEEE, pp. 845–852.
- Zhou, W., Li, R., Yuan, S., *et al.*, 2017. MetaSpark: A spark-based distributed processing tool to recruit metagenomic reads to reference genomes. *Bioinformatics*. btw750.
- Zou, Q., Hu, Q., Guo, M., Wang, G., 2015. HAlign: Fast multiple similar DNA/RNA sequence alignment based on the centre star strategy. *Bioinformatics*. btv177.

Infrastructure for High-Performance Computing: Grids and Grid Computing

Ivan Merelli, Institute for Biomedical Technologies (CNR), Milan, Italy and National Research Council, Segrate, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The lightening development of molecular biology techniques, with the increased capacity of producing high-throughput experiments, is posing important challenges for data storage, organization and analysis in Computational Biology and Bioinformatics. Handling the vast quantities of biological data generated by high-throughput experimental technologies is becoming increasingly difficult in a localized environment.

Small companies and small research groups produce uneven spikes of computationally intensive jobs, which makes a privately owned IT infrastructure a non-effective solution, since clusters can be underpowered during the actual spikes of work, while remaining underused for the majority of the time. Moreover, local clusters tend to be very expensive, since beside the hardware and software costs, electricity and cooling systems, the personnel for administration should also be considered.

Accordingly, Bioinformatics is experiencing a new trend in the way analysis is performed: computation is moving from in-house computing infrastructure to distributed computing delivered over the Internet. In this context, grid technology is a very interesting solution (Talbi and Zomaya, 2007), since it permits the transparent coupling of geographically dispersed resources (machines, networks, data storage, visualization devices, and scientific instruments) for large-scale distributed applications.

Starting from 2000s, grid computing has been for many years a resource of computational power suitable to support bioinformatic researches. Although developed in the field of high energy physics, the biomedical and bioinformatic communities worked as testbed of the infrastructure and many computational challenges have been performed. Beside large campaigns for the analysis of bio-sequences (Trombetti *et al.*, 2007), grid computing has been widely exploited in structural Bioinformatics, both for large scale virtual screenings, challenges of drugs discovery for neglected diseases (Lee *et al.*, 2006; Jacq *et al.*, 2007) and molecular dynamics of huge biomolecular systems (Merelli *et al.*, 2007).

The problem with grid computing is the low flexibility of the infrastructure, since there is only limited access to the servers, with little or no interactivity, due to remote batch scheduling of the jobs on the remote clusters. For desktop-based grid infrastructures the situation is even worst (Fedak, 2012). The management of the distributed data storage is very complex, not to mention the administration of geographically dispersed databases. The grid environment should be perfectly tuned in order to make an application running: a classical example is the presence of a mis-configured node on the grid, in which jobs fail continuously, emptying the grid queue, an effect known as *shrink-hole*.

More recently, grid computing embraced the idea of virtualization, moving towards a cloud computing paradigm. Different paradigms are available to implement this approach (Kee and Kesselman, 2008). This virtualization layer allows more flexibility, making grid computing able to address Big Data problems in many fields of Bioinformatics. For example, the Worker Nodes on Demand Service (WNoDeS) is a framework that makes possible to dynamically allocate virtual resources out of a common resource pool (Salomoni *et al.*, 2011).

Which Bioinformatic Applications Need Distributed Computing?

Considering bio-sequence analysis, short-read from Next-Generation Sequencing can be easily aligned against a reference genome on a single server using multi-thread implemented applications, such as Bowtie (Langmead and Salzberg, 2012). But the large number of experiments that are daily performed to identify genomic and epigenomic variations in patients affected by different diseases, and in particular cancer (which also requires a specific annotation of data; Alfieri *et al.*, 2008), generates huge computational loads, which boosted the development of grid tools to face this problem (Luyf *et al.*, 2010).

Nonetheless, more sophisticated analysis can be necessary from time to time. For example, to align protein or nucleic sequences against protein databases researchers must use tools such as BLAST (Altschul *et al.*, 1997), which has been ported in grid using many different approaches, such as BeoBlast (Grant *et al.*, 2002), Soap-HT-BLAST (Wang and Mu, 2003), mpiBLAST (Darling *et al.*, 2003), GridBLAST (Krishnan, 2005), ND-BLAST (Dowd *et al.*, 2005) and BGBlast (Trombetti *et al.*, 2007), and Crossbow (Gurtowski *et al.*, 2012).

Another example of sequence based analysis is the identification of regulatory patterns using HMMER (Finn *et al.*, 2011) (or similar Hidden Markov Model based software), which has been also ported on grid (Ciuffo and Mayo, 2009). The analysis of microRNA targets is also very complex, because the identification relies both on sequence similarity and on thermodynamic considerations (Ronchieri *et al.*, 2016). Moreover, microRNA seed sequences are very short, which contributes at generating a huge number of target predictions, also considering that the binding between the microRNA and the gene can have mismatches, at least in animals.

Huge efforts are also necessary to understand the impact of genome variations in the development of diseases. Genome wide association studies, involving thousands of patients, are routinely performed to discover possible biomarkers and design new

therapies (Merelli *et al.*, 2013). This requires large computational effort to achieve accurate statistical testing and Plink (Purcell *et al.*, 2007), which is one of the most used software for this kind of analysis, has been also ported to grid (Calabria *et al.*, 2010).

However, the most time consuming applications from the biological domain are those related to structural Bioinformatics (Chiappori *et al.*, 2013). Drug discovery and design are highly computational intensive, because large datasets of chemical compounds must be tested against molecular targets (usually proteins) to find molecules that can act as drugs. Considering that docking software are usually optimized for specific target families, it is usually a good idea to test many of them (Morris *et al.*, 2009; Mukherjee *et al.*, 2010; Friesner *et al.*, 2006; Merelli *et al.*, 2011). This approach, sometimes referred as virtual screening, requires massive computational resources, but it is very precious to reduce the number of wet experiments to perform, in particular for neglected diseases (D'Agostino *et al.*, 2013).

Moreover, side effects caused by off-target bindings should be avoided, therefore the most promising compounds are usually tested against many other proteins. At last, the docking conformations that describe the interactions between the compound and the targets should be optimized through molecular dynamics simulations to relax the system and improve the accuracy by which the binding energy is calculated. Molecular dynamics is a physical simulations in which the Newton's equations of motions are solved for each atom of the systems considering all the forces involved in the interaction (Merelli *et al.*, 2007) and, depending on the number of involved atoms, it can be very computationally demanding. Many attempts have been made to perform molecular dynamics on grid (Chiappori *et al.*, 2012, 2016), by using different paradigms, although one of the most popular is the Hamiltonian Replica Exchange (Jiang *et al.*, 2014).

Globus Based Grids

Most of the modern grid infrastructures rely on standard implementations of a middleware, such as the Globus Toolkit (Foster, 2006), an open source software for building Grid Services (GS). On the top of this middleware different grid implementations have been proposed, which usually follow two directions. The first is mainly application oriented and uses GS to provide utilities, such as authenticated and session enabled applications, providing high portability, but low scalability (Goble *et al.*, 2003). The second is a set of blank GS, that are computational resources where both data and software must be copied to perform computations, which present high scalability, but lower portability (Burke *et al.*, 2008).

These implementations cover different aspects, which can be both useful in Bioinformatics. The first is mainly related to the accessibility of distributed services, while the second concerns the possibility to perform wide range computations. While the service oriented implementation can be useful in terms of workflow composition, which can be remotely executed on distributed resources, the added value of the grid probably relies in the second implementation, which can be effectively used to mine the huge quantity of data that molecular biology is currently producing. In the rest of the section we will discuss this second aspect, as this is highly innovative, and with a particular attention to the EGEE project grid implementation, which is becoming the de-facto standard for grid computing in Europe.

The EGEE/EGI Grid Platform

The EGEE/EGI project infrastructure is a wide area grid platform for scientific applications composed of thousands of CPUs. This platform is a network of several Computing Elements, which constitute the gateways for the computer clusters on which jobs are performed, and an equal number of Storage Elements, that implement a distributed file system on which files are stored. The grid core is a set of Resource Brokers delegated for controlling submission and execution of the different jobs.

The computational resources are connected to a Resource Broker that routes each job on a specific Computing Element taking into account the directives of the submitting script, encoded using the Job Description Language (JDL), and implements a load balancing policy. Each Computing Element submits the incoming job to a batch system queue hiding many Working Nodes. A set of tools is used to manage data as in a distributed file systems. These tools allow the data to be replicated to different Storage Elements easily, and help using this information in an efficient way. The Resource Broker, in fact, is able to redirect the execution of an application to a Computing Element located as near as possible to the data being used, hence minimizing the communication time.

Apart from uploading data from the Storage Elements, a procedure that typically is reserved to large files and reference databases, in the JDL script it is possible specify an InputSandBox: a collection of files which are sent directly on the computational resources. This aspect is very important because it allows the subdivision of the input sequences to be analyzed and their direct upload with the related job to be computed. At the same way output data can be downloaded from the Working Nodes without copying data on the Storage Elements by using the OutputSandBox.

The EGEE grid enables an efficient data management, by coupling the Storage Element services, where files are stored, with the LHC File Catalogues where the physical handlers of files, which depend on the specific storage facility used, are recorder and associated to the logical file names. This system allows to have more replicas for each file, a feature that is crucial to implement redundancy of the reference databases, minimizing the transfer rate at running time. The system composed by the Storage Elements and the related LHC File Catalogue implements an effective virtual distributed file system, which allows redundancy and replicas of files.

All the elements of this grid computing environment are deployed as Grid Services, which envelope the main basic functionalities, enabling secure communications among the grid components. In particular, the gLite middleware must be installed on a local server within the institution that wants to join the EGEE grid in order to establish secure communications between the institution itself and the distributed infrastructure. This local server, which is usually called User Interface, through the gLite APIs enables to submit jobs, to monitor the state of advancement of the jobs, to retrieve the outputs when the computations have a normal termination and to resubmit the jobs in case of failure.

Due to the use of remote computational resources, the grid communication software must offer an efficient security system. Simply stated, security is guaranteed by the Grid Security Infrastructure (GSI), which uses public key cryptography to recognize users. GSI certificates are encoded in the X.509 format and accompany each job to authenticate the user. In other words, the access to remote clusters is granted by a Personal Certificate, which accompanies each job to authenticate the user. Moreover, users must be authorized to job submission by a Virtual Organization, a grid community having similar tasks, which grants for him. This procedure is indispensable for maintaining a high security level.

BIONC Based Grids

The other grid approach that gained progressive success in the scientific community is known as Desktop Grids (DGs), which often relies upon the general public to donate resources (volunteer computing). Unlike Grid Services, which are based on complex architectures, volunteer computing has a simple architecture and has demonstrated the ability to integrate dispersed, heterogeneous computing resources with simplicity, successfully scavenging cycles from tens of thousands of idle desktop computers. This paradigm represents a complementary trend concerning the original aims of Grid computing. In DG systems, anyone can bring resources into the Grid, installation and maintenance of the software is intuitive, requiring no special expertise, thus enabling a large number of donors to contribute into the pool of shared resources.

On the down-side, only a very limited user community (i.e., target applications) can effectively use Desktop Grid resources for computation. The most well-known DG example is the SETI@home ([Anderson et al., 2002](#)) project, in which approximately four million PCs have been involved. The middleware that was originally developed to support SETI@home is the Berkeley Open Infrastructure for Network Computing (BOINC, [Anderson, 2004](#)), which rapidly became a generalized platform for other distributed applications in areas as diverse as mathematics, linguistics, medicine, molecular biology, climatology, environmental science, and astrophysics, among others.

Rosetta@Home

Rosetta@home is a distributed computing project for protein structure prediction, relying on the BOINC platform, which aims at performing protein-protein docking and protein modelling. Though much of the project is oriented toward basic research to improve the accuracy and robustness of proteomic methods, Rosetta@home also does applied research on malaria, Alzheimer's disease, and other pathologies ([Das et al., 2007](#)).

Like all BOINC projects, Rosetta@home uses idle computer processing resources from volunteers' computers to perform calculations on individual workunits. Completed results are sent to a central project server where they are validated and assimilated into project databases. The project is cross-platform, and runs on a wide variety of hardware configurations. Users can view the progress of their individual protein structure prediction on the Rosetta@home screen saver.

In addition to disease-related research, the Rosetta@home network serves as a testing framework for new methods in structural Bioinformatics. Such methods are then used in other Rosetta-based applications, like RosettaDock and the Human Proteome Folding Project, after being sufficiently developed and proven stable on Rosetta@home's large and diverse set of volunteer computers. Two especially important tests for the new methods developed in Rosetta@home are the Critical Assessment of Techniques for Protein Structure Prediction (CASP) and Critical Assessment of Prediction of Interactions (CAPRI) experiments, biannual experiments that evaluate the state of the art in protein structure prediction and protein-protein docking prediction, respectively. Rosetta@home consistently ranks among the foremost docking predictors, and is one of the best tertiary structure predictors available ([Lensink et al., 2007](#)).

Virtual Paradigms

The main distinguishing feature between GSs and DGs is the way computations are initiated at the grid resources. In Grid Services a job submission or a service invocation is used to initiate the activity on a grid resource. Both can be considered as a specific form of the *push model* where the service requestor pushes jobs, tasks, service invocations on the passive resources.

Once such a request is pushed on the resource, it becomes active and the requested activity is executed. Desktop Grids work according to the *pull model*. Resources that have got spare cycles pull tasks from the application repository, which is typically placed on the DG server. In this way resources play an active role in the DG system, they initiate their own activity based on the task pulled from the server.

Both these approaches are quite rigid and can suffer from mis-configuration of the resources, which may cause many job failure. To solve this problem and implement a more interactive paradigm, grid computing has recently moved towards the idea of virtualization. The possibility to virtualize resources is critical for making grid computing able to address the complex workflow analysis systems that are necessary in Bioinformatics.

One of the most interesting approaches in this sense is Worker Nodes on Demand Services (WNoDes) framework (Salomoni *et al.*, 2011), which implement a Cloud over grid approach. It provides a cloud-based environment by instantiating resources on top of a grid infrastructure. WNoDes instantiates virtual machines from predefined images on top of Grid resources exploiting the local batch system of the grid site. Other examples are the CLEVER project (Tusa *et al.*, 2010), which supplies access to private and hybrid cloud, and PerfCloud (Mancini *et al.*, 2009), a novel architecture that creates configurable virtual clusters to be used in the grid environment.

The above mentioned paradigm can be exploited both through services and software frameworks, such as the distributed infrastructure with remote agent control (DIRAC, Fifield *et al.*, 2011), HTCondor (Thain *et al.*, 2006) and the gLite workload management system (Marco *et al.*, 2009), which are able to provide high-level job submission services as front end, and the possibility to interact with heterogeneous systems at the back end. In particular, the gLite WMS and DIRAC provides a complete solution to one (or more) user community requiring access to distributed resources. HTCondor supports distributed parallelised jobs on grid and cloud resources.

Performance

A critical point to justify the effort required for porting a bioinformatic application over grid is the possibility to achieve good performance. Clearly, scalability results are very important while considering the grid infrastructure as a possible solution to reduce the time required for a computation. In particular, we would discuss the job efficiency and the latency time, which are critical to decide the granularity of the computation, the overhead imposed by the grid and the global scalability of the infrastructure.

Job Efficiency and Latency Time

In literature are reported different analyses of the grid efficiency (Merelli *et al.*, 2015), which for bioinformatic applications seem to settle around the 70% of the jobs reported as successfully finished according to the status logged in the Resource Broker. A ratio that goes down to 60% after checking the existence of the output (Merelli *et al.*, 2015). The main cause of job failure seems to be the data transfer between the Storage Elements and the Computing Elements and, in few cases, the communication between the Computing Elements and the local results database.

This can be inferred by the differences in the statistics achieved by different computational challenges: challenges in which the data transfer rate is relatively important present a higher number of failures (Merelli *et al.*, 2015). Another frequent problem is related to errors during the job scheduling: a very high ratio of these failures came from the mis-configuration of Computing Elements. On the other hand, the number of jobs reported as correctly finished by the grid, but resubmitted because the results didn't match with the expected information is generally pretty low.

A very important factor for the reliability of the grid is the average latency time, which can be defined as the sum of the time spent for establishing communication with the Resource Broker and the scheduling time on the Computing Element queue. Depending on the average load of the grid, and considering only jobs that come to completion at the first round, the average latency time can be estimated in a few hours (Merelli *et al.*, 2015). Clearly this latency time has low impact on long term jobs, but heavily influences short jobs, making the grid quite unsuitable for short computations.

Granularity

A key parameter for the evaluation of the distribution of the computational load is the estimated duration of a single grid job. Apart from job failures due to data transfer or internal errors, the time needed for completing the whole computation can also be elongated by the execution of jobs too short, which greatly increase the latency in the execution queue. On the other hand the submission of a small number of jobs limits the scalability of the system and in case of failure may impose to wait long time for the fulfilment of a few resubmitted jobs. Hence, the length of the submitted jobs should be carefully planned.

Crunching Factor

Defining the scalability of a grid infrastructure is not easy as for parallel infrastructure, since it is not possible to compute the speed-up of an application. Nonetheless, in the grid domain, it is possible to estimate the crunching factor, which is defined as the expected single-core time required for the computational divided by the real computational time achieved on the distributed platform. This is the time needed to complete the last job. The CPU utilization peak is usually much higher, but due to overhead caused by queuing for accessing grid resources and job failures the scalability is generally lower. To avoid this problem, commonly named *last job syndrome*, it is possible to submit multiple replicas of the jobs that failed at the first round. Once the fastest replica is finished for each job failed at the first round, all the remaining computations can be dropped, which usually is a useful trick to accelerate the computation and improve the crunching factor.

Conclusions

In the last years, the role of grid computing has become increasingly important in scientific research. Concerning Bioinformatics, grid computing tackles challenges of processing power, large-scale data access and management, security, application integration, data integrity and curation, control/automation/tracking of workflows, data format consistency and resource discovery.

Grid technology allows computational intensive challenges to be accomplished, but due to its rigid constraints a capable submission and monitoring environment should be set-up in order to appropriately manage the volume of data. To mitigate these problems, where possible, the creation of a flexible virtual environment can play an important role to achieve good results in short time.

In conclusion, the grid is an effective system for coping with the increasing demand of computational power in Bioinformatics. The system scalability in case of independent computations is very high, even taking into account the time needed for scheduling jobs and transferring data. Such computing power is a concrete possibility to face challenges that have been thought to be impossible just few years ago.

See also: Computing Languages for Bioinformatics: Java. Dedicated Bioinformatics Analysis Hardware. Models and Languages for High-Performance Computing. Parallel Architectures for Bioinformatics. Text Mining Applications

References

- Alfieri, R., Merelli, I., Mosca, E., *et al.*, 2008. The cell cycle DB: A systems biology approach to cell cycle analysis. *Nucleic Acids Research* 36 (Suppl. 1), D641–D645.
- Altschul, S.F., Madden, T.L., Schffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25 (17), 3389–3402.
- Anderson, D.P., 2004. Boinc: A system for public-resource computing and storage. In: *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, 2004, IEEE, pp. 4–10.
- Anderson, D.P., Cobb, J., Korpela, E., *et al.*, 2002. SETI home: An experiment in public-resource computing. *Communications of the ACM* 45 (11), 56–61.
- Burke, S., Campana, S., Lorenzo, M.P., *et al.*, 2008. Glite 3.1 User Guide (2008).
- Calabria, A., Di Pasquale, D., Gnocchi, M., *et al.*, 2010. *Journal of Grid Computing* 8, 511. doi:10.1007/s10723-010-9163-y.
- Chiappori, F., Mattiazzi, L., Milanesi, L., *et al.*, 2016. A novel molecular dynamics approach to evaluate the effect of phosphorylation on multimeric protein interface: The B-crystallin case study. *BMC Bioinformatics* 17 (4), 57.
- Chiappori, F., Merelli, I., Milanesi, L., *et al.*, 2013. Static and dynamic interactions between GALK enzyme and known inhibitors: Guidelines to design new drugs for galactosemic patients. *European Journal of Medicinal Chemistry* 63, 423–434.
- Chiappori, F., Pucciarelli, S., Merelli, I., *et al.*, 2012. Structural thermal adaptation of tubulins from the Antarctic psychrophilic protozoan *Euplotes focardii*. *Proteins: Structure, Function, and Bioinformatics* 80 (4), 1154–1166.
- Ciuffo, L.N., Mayo, R., 2009. Biomedical applications in the EELA-2 project. In: *Proceedings of the Ninth Annual Workshop Network Tools and Applications in Biology*, pp. 35–38.
- D'Agostino, D., Clematis, A., Quarati, A., *et al.*, 2013. Cloud infrastructures for in silico drug discovery: Economic and practical aspects. *BioMed Research International* 2013, 1–19.
- Darling, A., Carey, L., Feng, W.C., 2003. The design, implementation, and evaluation of mpiBLAST. In: *Proceedings of ClusterWorld*, 2003, pp. 13–15.
- Das, R., Qian, B., Raman, S., *et al.*, 2007. Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. *Proteins: Structure, Function, and Bioinformatics* 69 (S8), 118–128.
- Dowd, S.E., Zaragoza, J., Rodriguez, J.R., Oliver, M.J., Payton, P.R., 2005. Windows .NET network distributed basic local alignment search toolkit (W.ND-BLAST). *BMC Bioinformatics* 6 (1), 93.
- Fedak, G. (Ed.), 2012. *Desktop Grid Computing*. Chapman and Hall/CRC.
- Fiefield, T., Carmona, A., Casajs, A., Graciani, R., Sevier, M., 2011. Integration of cloud, grid and local cluster resources with DIRAC. *Journal of Physics: Conference Series* 331 (6), 062009. IOP Publishing.
- Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Research*. doi:10.1093/nar/gkr367.
- Foster, I., 2006. Globus toolkit version 4: Software for service-oriented systems. *LNCS* 3779, 2–13.
- Friesner, R.A., Murphy, R.B., Repasky, M.P., *et al.*, 2006. Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein–ligand complexes. *Journal of Medicinal Chemistry* 49 (21), 6177–6196.
- Goble, C.A., Pettiier, S., Stevens, R., *et al.*, 2003. Knowledge Integration: In silico experiments in Bioinformatics. In: Foster, I., Kesselman, C. (Eds.), *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., pp. 121–124.
- Grant, J.D., Dunbrack, R.L., Manion, F.J., *et al.*, 2002. BeoBLAST: Distributed BLAST and PSI-BLAST on a Beowulf cluster. *Bioinformatics* 18 (5), 765–766.
- Gurtowski, J., Schatz, M.C., Langmead, B., 2012. Genotyping in the cloud with crossbow. *Current Protocols in Bioinformatics*. 15.3.1–15.3.15.
- Jacq, N., Breton, V., Chen, H.Y., *et al.*, 2007. Virtual screening on large scale grids. *Parallel Computing* 33 (4), 289–301.
- Jiang, W., Phillips, J.C., Huang, L., *et al.*, 2014. Generalized scalable multiple copy algorithms for molecular dynamics simulations in NAMD. *Computer Physics Communications* 185 (3), 908–916.
- Kee, Y.S., Kesselman, C., 2008. Grid resource abstraction, virtualization, and provisioning for time-targeted applications. In: *Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid*, 2008. CCGRID'08, IEEE, pp. 324–331.
- Krishnan, A., 2005. GridBLAST: A globus-based high-throughput implementation of BLAST in a Grid computing framework. *Concurrency and Computation: Practice and Experience* 17 (13), 1607–1623.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nature Methods* 9 (4), 357–359.
- Lee, H.C., Salzemann, J., Jacq, N., *et al.*, 2006. Grid-enabled high-throughput in silico screening against influenza A neuraminidase. *IEEE Transactions on Nanobioscience* 5 (4), 288–295.
- Lensink, M.F., Mndez, R., Wodak, S.J., 2007. Docking and scoring protein complexes: CAPRI 3rd Edition. *Proteins: Structure, Function, and Bioinformatics* 69 (4), 704–718.
- Luyf, A.C., van Schaik, B.D., de Vries, M., *et al.*, 2010. Initial steps towards a production platform for DNA sequence analysis on the grid. *BMC Bioinformatics* 11 (1), 598.

- Mancini, E.P., Rak, M., Villano, U., 2009. Perfcloud: Grid services for performance-oriented development of cloud computing applications. In: Proceedings of the 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE'09, IEEE, pp. 201–206.
- Marco, C., Fabio, C., Alvise, D., *et al.*, 2009. The glite workload management system. In: International Conference on Grid and Pervasive Computing. Berlin; Heidelberg: Springer, pp. 256–268.
- Merelli, I., Calabria, A., Cozzi, P., *et al.*, 2013. SNPranker 2.0: A gene-centric data mining tool for diseases associated SNP prioritization in GWAS. *BMC Bioinformatics* 14 (1), S9.
- Merelli, I., Cozzi, P., D'Agostino, D., *et al.*, 2011. Image-based surface matching algorithm oriented to structural biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 8 (4), 1004–1016.
- Merelli, I., Cozzi, P., Ronchieri, E., *et al.*, 2015. Porting bioinformatics applications from grid to cloud: A macromolecular surface analysis application case study. *International Journal of High Performance Computing Applications*. doi:10.1177/1094342015588565.
- Merelli, I., Morra, G., Milanesi, L., 2007. Evaluation of a grid based molecular dynamics approach for polypeptide simulations. *IEEE Transactions on NanoBioscience* 6 (3), 229–234.
- Morris, G.M., Huey, R., Lindstrom, W., *et al.*, 2009. Autodock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry* 16, 2785–2791.
- Mukherjee, S., Balias, T.E., Rizzo, R.C., 2010. Docking validation resources: Protein family and ligand flexibility experiments. *Journal of Chemical Information and Modeling* 50 (11), 1986–2000.
- Purcell, S., Neale, B., Todd-Brown, K., *et al.*, 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics* 81 (3), 559–575.
- Ronchieri, E., D'Agostino, D., Milanesi, L., *et al.*, 2016. MicroRNA–target interaction: A parallel approach for computing pairing energy. In: Proceedings of the 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), IEEE, pp. 535–540.
- Salomoni, D., Italiano, A., Ronchieri, E., 2011. WNoDeS, a tool for integrated grid and cloud access and computing farm virtualization. *Journal of Physics: Conference Series* 331 (5), 052017. IOP Publishing.
- Talbi, E.G., Zomaya, A.Y. (Eds.), 2007. *Grid Computing for Bioinformatics and Computational Biology*, vol. 1. John Wiley & Sons.
- Thain, D., Tannenbaum, T., Livny, M., 2006. How to measure a large open source distributed system. *Concurrency and Computation: Practice and Experience* 18 (15), 1989–2019.
- Trombetti, G.A., Merelli, I., Orro, A., Milanesi, L., 2007. BGBlast: A blast grid implementation with database self-updating and adaptive replication. *Studies in Health Technology and Informatics* 126, 23.
- Tusa, F., Paone, M., Villari, M., *et al.*, 2010. CLEVER: A cloud-enabled virtual environment. In: IEEE Symposium on Computers and Communications (ISCC), IEEE, pp. 477–482.
- Wang, J., Mu, Q., 2003. Soap-HT-BLAST: High throughput BLAST based on Web services. *Bioinformatics* 19 (14), 1863–1864.

Infrastructures for High-Performance Computing: Cloud Computing

Paolo Trunfio, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Cloud Computing abstracts computing resources to a utility-based model. This model is based on the virtualization of networks, servers, storage and services that clients can allocate on a *pay-per-use* basis to implement their distributed applications. From a client perspective, the cloud is an abstraction for remote, infinitely scalable provisioning of computing resources. From an implementation point of view, cloud systems are based on large sets of computing resources, located somewhere “in the cloud”, which are allocated to applications on demand (Barga *et al.*, 2011).

Thus, cloud computing can be defined as a distributed computing paradigm in which all the resources, dynamically scalable and virtualized, are provided as services over the Internet (Talia *et al.*, 2015). Virtualization is software-based technique that implements the separation of physical computing infrastructures and allows creating various “virtual” computing resources on the same hardware. It is a basic technology that powers cloud computing by making possible to concurrently run different operating environments and multiple applications on the same server. Differently from other distributed computing paradigms, cloud users are not required to have knowledge of, expertise in, or control over the technology infrastructure in the “cloud” that supports them. A number of features define cloud applications, services, data, and infrastructure:

- Remotely hosted: Services and/or data are hosted on remote infrastructure.
- Ubiquitous: Services or data are available from anywhere.
- Pay-per-use: The result is a utility computing model similar to that of traditional utilities, like gas and electricity, where you pay for what you use.

We can also refer to the popular National Institute of Standards and Technology (NIST) definition of cloud computing to highlight its main features (Mell and Grance, 2011): “Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. From the NIST definition, we can identify five essential characteristics of cloud computing systems:

1. *On-demand self-service*: The ability to allocate resources on-demand and through self-service interfaces.
2. *Broad network access*: The availability of high speed network access to all of the resources.
3. *Resource pooling*: A common pool of resources is transparently allocated to multiple users.
4. *Rapid elasticity*: The ability to scale up or scale down in the shortest possible time.
5. *Measured service*: The ability to control the use of resources by leveraging a resource metering system.

Cloud systems can be classified on the basis of their *service model* (Software as a Service, Platform as a Service, Infrastructure as a Service) and their *deployment model* (public cloud, private cloud, community cloud).

Service Models

Cloud computing vendors provide their services according to three main models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Software as a Service defines a delivery model in which software and data are provided through Internet to customers as ready-to-use services. Specifically, software and associated data are hosted by providers, and customers access them without need to use any additional hardware or software. Moreover, customers normally pay a monthly/yearly fee, with no additional purchase of infrastructure or software licenses. Examples of common SaaS applications are Webmail systems (e.g., Gmail), calendars (Yahoo Calendar), document management (Microsoft Office 365), image manipulation (Photoshop Express), customer relationship management (Salesforce), and others.

In *Platform as a Service* model, cloud vendors deliver a computing platform typically including databases, application servers, development environment for building, testing and running custom applications. Developers can just focus on deploying of applications since cloud providers are in charge of maintenance and optimization of the environment and underlying infrastructure. Hence, customers are helped in application development as they use a set of “environment” services that are modular and can be easily integrated. Normally, the applications are developed as ready-to-use SaaS. Google App Engine, Microsoft Azure, Salesforce.com are some examples of PaaS cloud environments.

Finally, *Infrastructure as a Service* is an outsourcing model under which customers rent resources like CPUs, disks, or more complex resources like virtualized servers or operating systems to support their operations (e.g., Amazon EC2, RackSpace Cloud). Users of an IaaS have normally skills on system and network administration as they must deal with configuration, operation and

maintenance tasks. Compared to the PaaS approach, the IaaS model has a higher system administration costs for the user; on the other hand, IaaS allows a full customization of the execution environment. Developers can scale up or down its services adding or removing virtual machines, easily instantiable from a virtual machine images.

In the following we describe how the three service models satisfy the requirements of developers and final users, in terms of *flexibility, scalability, portability, security, maintenance* and *costs*.

Flexibility

SaaS: Users can customize the application interface and control its behavior, but cannot decide which software and hardware components are used to support its execution.

PaaS: Developers write, customize, test their application using libraries and supporting tools compatible with the platform. Users can choose what kind of virtual storage and compute resources are used for executing their application.

IaaS: Developers have to build the servers that will host their applications, and configure operating system and software modules on top of such servers.

Scalability

SaaS: The underlying computing and storage resources normally scale automatically to match application demand, so that users do not have to allocate resources manually. The result depends only on the level of elasticity provided by the cloud system.

PaaS: Like the SaaS model, the underlying computing and storage resources normally scale automatically.

IaaS: Developers can use new storage and compute resources, but their applications must be scalable and allow the dynamic inclusion of new resources.

Portability

SaaS: There can be problems to move applications to other providers, since some software and tools could not work on different systems. For example, application data may be in a format that cannot be read by another provider.

PaaS: Applications can be moved to another provider only if the new provider shares with the old one the required platform tools and services.

IaaS: If a provider allows to download a virtual machine in a standard format, it may be moved to a different provider.

Security

SaaS: Users can control only some security settings of their applications (e.g., using https instead of http to access some Web pages). Additional security layers (e.g., data replication) are hidden to the user and managed directly by the system.

PaaS: The security of code and additional libraries used to build application is responsibility of the developer.

IaaS: Developers must take care of security issues from operating system to application layer.

Maintenance

SaaS: Users have not to carry maintenance tasks.

PaaS: Developers are in charge of maintaining only their application; other software components and the hardware are maintained by the provider.

IaaS: Developers are in charge of all software components, including the operating system; hardware is maintained by the provider.

Cost

SaaS: Users typically pay a monthly/yearly fee for using the software, with no additional fee for the infrastructure.

PaaS: Developers pay for the compute and storage resources, and for the licenses of libraries and tools used by their applications.

IaaS: Developers pay for all the software and hardware resources used.

Deployment Models

Cloud computing services are delivered according to three main deployment models: public, private and community cloud.

A *public cloud* provider delivers services to the general public through the Internet. The users of a public cloud have little or no control over the underlying technology infrastructure. In this model, services can be offered for free, or provided according to a *pay-per-use* policy. The main public providers, such as Google, Microsoft, Amazon, own and manage their proprietary data centers delivering services built on top of them.

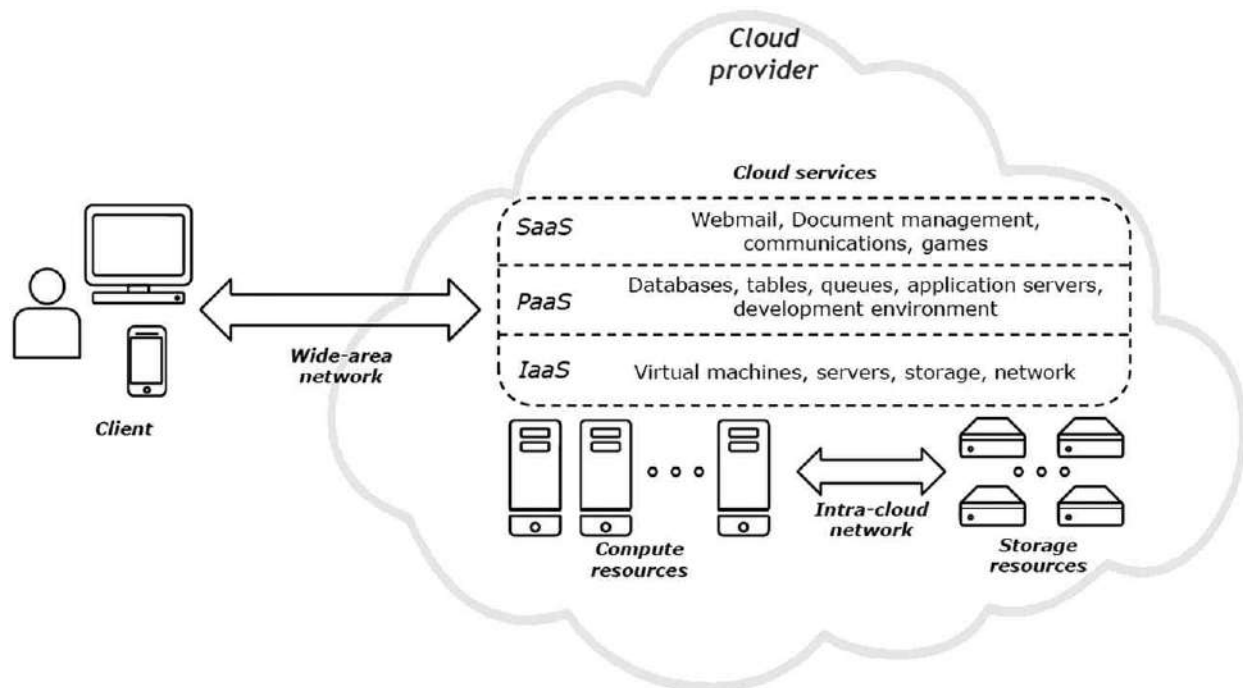


Fig. 1 General architecture of a public cloud.

A *private cloud* provider offers operations and functionalities “as a service”, which are deployed over a company intranet or hosted in a remote data center. Often, small and medium-sized IT companies prefer this deployment model as it offers advance security and data control solutions that are not available in the public cloud model.

A *community cloud* is a model in which the infrastructure is provisioned for reserved use of a given community of organizations that have shared objectives and requirements. This kind of cloud may be owned and managed by one or more of the organizations in the community, or by an organization that is not part of the community, or some combination of them.

Fig. 1 depicts the general architecture of a public cloud and its main components, as outlined in [Li et al. \(2010\)](#) and [Talia et al. \(2015\)](#). Users access cloud computing services using *client* devices, such as desktop computers, laptops, tablets and smartphones. Through these devices, users access and interact with cloud-based services using a Web browser or desktop/mobile app. The business software and user’s data are executed and stored on servers hosted in cloud data centers that provide *storage and computing resources*. Resources include thousands of servers and storage devices connected each other through an *intra-cloud network*. The transfer of data between data center and users takes place on *wide-area network*.

Several technologies and standards are used by the different components of the architecture. For example, users can interact with cloud services through SOAP-based or RESTful Web services ([Richardson and Ruby, 2007](#)). *HTML5* and *Ajax* technologies allow Web interfaces to cloud services to have look and interactivity equivalent to those of desktop applications. *Open Cloud Computing Interface* (OCCL, OCCI Working Group, see Relevant Websites section) specifies how cloud providers can deliver their compute, data, and network resources through a standardized interface. Another example is *Open Virtualization Format* (OVF, OVF Specification, see Relevant Websites section) for packaging and distributing virtual devices or software (e.g., virtual operating system) to be run on virtual machines.

Interconnected Clouds

Interconnection and interoperability of different clouds can be beneficial to both cloud providers and their clients for avoiding vendor lock-in, to increase scalability and availability, to reduce access latency, and to improve energy efficiency ([Toosi et al., 2014](#)). Interoperability between clouds can be obtained through either *provider-centric* or *client-centric* approaches. With a provider-centric approach, cloud providers adopt and implement standard interfaces, protocols, formats, and architectural components that facilitate collaboration between different clouds. With a client-centric approach, there is a user-side library or a third-party service broker that translates messages between different cloud interfaces, thus allowing clients to switch between different clouds.

The most important provider-centric approaches are *hybrid cloud*, *cloud federation*, and *Inter-cloud*:

- A *hybrid cloud* is the composition of two or more (private or public) clouds that remain different entities but are linked together. Companies can extend their private clouds using other private clouds from partner companies, or public clouds. In particular,

by extending the private infrastructure with public cloud resources, it is possible to satisfy peaks of requests, better serve user requests, and implement high availability strategies.

- A *cloud federation* allows providers to overcome resource limitation in their local cloud infrastructure, which may result in refusal of client requests, by outsourcing requests to other members of the federation. This is made by sharing the cloud resources through federation regulations, which allows providers operating at low utilization rates to lease part of their resources to other members of the federation, thus improving resource utilization and reducing costs.
- An *Inter-cloud*, or *cloud of clouds*, is a model in which all clouds are globally interconnected, forming a global cloud federation. This approach removes difficulties related to the migration of applications and supports their dynamic scaling across multiple clouds worldwide.

Client-centric interoperability approaches can be classified into *multicloud* and *aggregated service by broker* models:

- In the *multicloud* model, the client applications are run on several clouds with the help of a user-side library, thus avoiding the difficulty of migrating applications across clouds.
- In the *aggregated service by broker*, a third-party broker offers an integrated service to users by coordinating access and utilization of multiple cloud resources.

As discussed in Toosi *et al.* (2014), cloud interoperability is still a challenging issue that requires substantial efforts to overcome the existing functional and nonfunctional problems in important areas such as security, virtualization, networking and provisioning.

Closing Remarks

In this article we provided an overview of cloud computing concepts and models. Starting from the NIST definition of cloud computing, we discussed the main features of cloud systems, and analyzed the most important service models (Software as a Service, Platforms as a Service, Infrastructure as a Service) and deployment models (Private Cloud; Community Cloud; Public Cloud) currently adopted by cloud providers. Finally, we discussed the most important models for the interconnection and interoperability of cloud environments (Hybrid Cloud, Cloud Federation, and Inter-cloud).

See also: Computing Languages for Bioinformatics: Java. Dedicated Bioinformatics Analysis Hardware. Models and Languages for High-Performance Computing. Parallel Architectures for Bioinformatics. Text Mining Applications

References

- Barga, R., Gannon, D., Reed, D., 2011. The client and the cloud: Democratizing research computing. *IEEE Internet Computing* 15 (1), 72–75.
- Li, A., Yang, X., Kandula, S., Zhang, M., 2010. CloudCmp: Comparing public cloud providers. In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC'10)*, New York, NY.
- Mell, P., Grance, T., 2011. *The NIST Definition of Cloud Computing*. Gaithersburg, MD: National Institute of Standards and Technology (NIST), (NIST Special Publication 800-145).
- Richardson, L., Ruby, S., 2007. *RESTful Web Services*. California: O'Reilly & Associates.
- Talia, D., Trunfio, P., Marozzo, F., 2015. *Data Analysis in the Cloud*. The Netherlands: Elsevier.
- Toosi, A.N., Calheiros, R.N., Buyya, R., 2014. Interconnected cloud computing environments. *ACM Computing Surveys* 47 (1), 1–47.

Relevant Websites

- <http://www.occi-wg.org>
 OCCI Working Group.
- http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf
 OVF Specification.

Infrastructures for High-Performance Computing: Cloud Infrastructures

Fabrizio Marozzo, University of Calabria, Rende (CS), Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

This article describes the main services provided by the most popular cloud infrastructures currently in use. The systems presented in this article are either public clouds *Amazon Web Services (AWS)*, *Google Cloud Platform*, *Microsoft Azure* and *IBM Bluemix* or private clouds *OpenStack*, *OpenNebula* and *Eucalyptus*.

All the public cloud systems provide services according to the Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) models (Talía et al., 2015). With the IaaS model, the cloud system provide services that allow customers to rent virtualized resources like storage and servers; the PaaS model allows developers to focus on deploying of applications, since cloud providers are in charge of managing the underlying infrastructure; with the SaaS model, software and data are provided to customers as ready-to-use services.

As examples of IaaS services, AWS provides the popular *Elastic Compute Cloud (EC2)* service that allows creating and running virtual servers; Google Cloud Platform allows developers to run virtual machines through its *Compute Engine* service; Microsoft Azure provides a *VM role* service to manage virtual-machine images; IBM Bluemix *Virtual servers* can be used to deploy and scale virtual machines on demand.

As examples of PaaS services, AWS provides *Elastic Beanstalk* to create, deploy, and manage applications using a large set of infrastructure services; Google Cloud Platform includes *App Engine*, a PaaS that allows developers to create and host web applications in Google-managed data centers; Microsoft Azure's PaaS includes a *Web role* for developing Web-based applications and a *Worker role* for batch applications; IBM Bluemix includes *Mobile Foundation*, a set of APIs to manage the life cycle of mobile applications.

Finally, some examples of SaaS services are *Simple Email Service (AWS)*, *Gmail (Google Cloud Platform)*, *Office 365 (Microsoft Azure)* and *Natural Language Understanding (IBM Bluemix)*.

In general, the private cloud systems are more focused on the IaaS model, as they provide services for the management of large pools of processing, storage, and networking resources in a data center, by exploiting a variety of virtualization techniques. For example, OpenStack with its *Compute* component provides virtual servers and with its *Storage* component provides a scalable and redundant storage system; OpenNebula's *Core component* creates and controls virtual machines by interconnecting them with a virtual network environment; Eucalyptus includes a *Node Controller* that hosts the virtual machine instances and manages the virtual network endpoints, and a *Storage Controller* that provides persistent block storage and allows the creation of snapshots of volumes.

Amazon Web Services

Amazon offers compute and storage resources of its IT infrastructure to developers in the form of Web services. Amazon Web Services (AWS) is a large set of cloud services that can be composed by users to build their SaaS applications or integrate traditional software with cloud capabilities (see Fig. 1). It is simple to interact with these services since Amazon provides SDKs for the main programming languages and platforms (e.g., Java, .Net, PHP, Android).

AWS includes the following main services:

- Compute: *Elastic Compute Cloud (EC2)* allows creating and running virtual servers; *Amazon Elastic MapReduce* for building and executing MapReduce applications.
- Storage: *Simple Storage Service (S3)*, which allows storing and retrieving data via the Internet.
- Database: *Relational Database Service (RDS)* for relational tables; *DynamoDB* for non-relational tables; *SimpleDB* for managing small datasets; *ElasticCache* for caching data.
- Networking: *Route 53*, a DNS Web service; *Virtual Private Cloud* for implementing a virtual network.
- Deployment and Management: *CloudFormation* for creating a collection of ready-to-use virtual machines with pre-installed software (e.g., Web applications); *CloudWatch* for monitoring AWS resources; *Elastic Beanstalk* to deploy and execute custom applications written in Java, PHP and other languages; *Identity and Access Management* to securely control access to AWS services and resources.
- Content delivery: *Amazon CloudFront* makes easy to distribute content via a global network of edge locations.
- App services: *Simple Notification Service* to notify users; *Simple Queue Service* that implements a message queue; *Simple Workflow Service* to implement workflow-based applications.

Even though Amazon is best known to be the first IaaS provider (based on its EC2 and S3 services), it is now also a PaaS and SaaS provider, with services like *Elastic Beanstalk*, which allows users to quickly create, deploy, and manage applications using a large set of AWS services, *Amazon Machine Learning*, which provides visualization tools and wizards for easily creating machine learning models, and *Simple Email Service* providing a basic email-sending service.

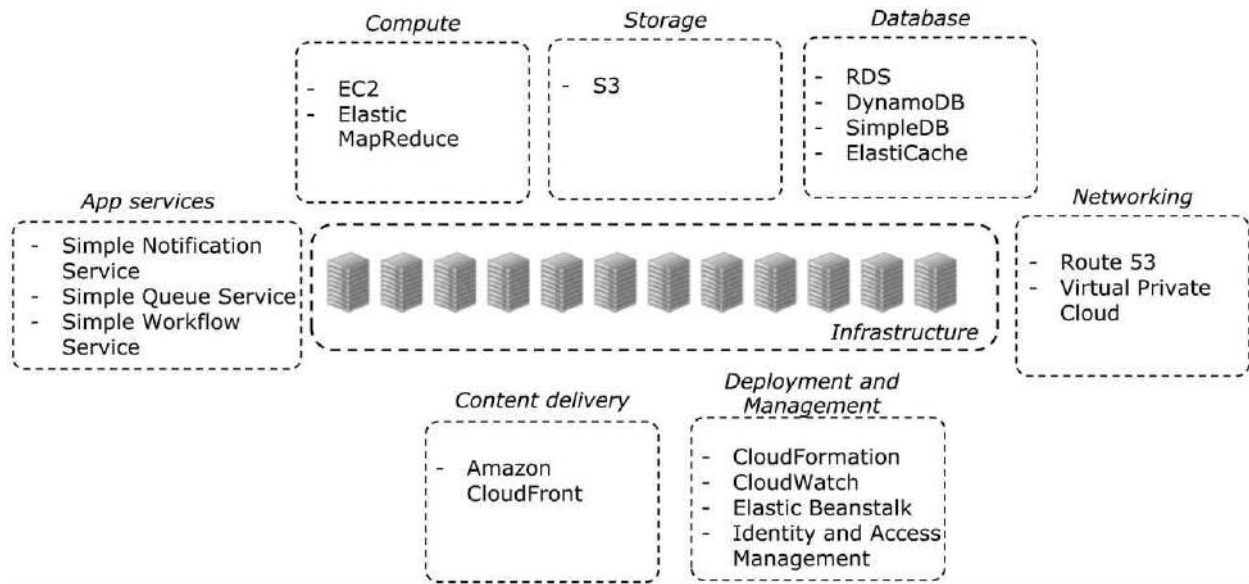


Fig. 1 Amazon Web Services.

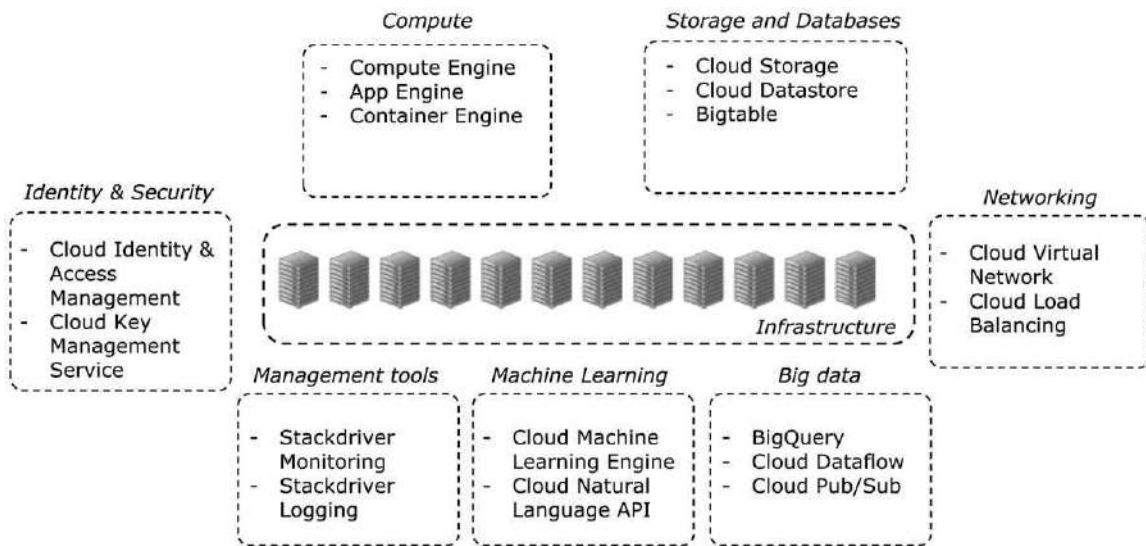


Fig. 2 Google Cloud Platform.

Google Cloud Platform

Google Cloud Platform is a set of cloud computing services provided by Google. The platform allows developers to define a wide range of web applications, from simple websites to scalable multitenancy applications. The services created by users run in the same supporting infrastructure that Google uses internally to run Google Search, YouTube, Maps, and Gmail. Google provides SDKs to define applications using several languages, libraries, and frameworks (e.g., GO, PHP, Java, Python and .Net).

The platform provides several services to build and scale an application (see Fig. 2):

- **Compute:** *Compute Engine*, an IaaS that allows developers to run virtual machine (VM) instances; *App Engine*, a PaaS that allows developers to create and host web applications in Google-managed data centers; *Container Engine*, a management and orchestration system for software container.
- **Storage and Databases:** *Cloud Storage*, a cloud storage web service for storing and accessing large and/or unstructured data sets; *Cloud Datastore*, a highly scalable NoSQL database service; *Cloud Bigtable*, a highly scalable NoSQL database service used by Google to power many core services, including Search, Analytics, Maps, and Gmail.

- Networking: *Cloud Virtual Network*, a managed networking functionality for web applications; *Cloud Load Balancing*, a load-balancer for compute resources in single or multiple regions.
- Big Data: *BigQuery*, a data warehouse enabling fast SQL queries; *Cloud Dataflow*, a data processing service for stream and batch execution of pipelines; *Cloud Pub/Sub*, a real-time messaging service to send and receive messages between applications.
- Machine Learning: *Cloud Machine Learning Engine*, a scalable machine learning service to train models and predict new data; *Cloud Natural Language API*, for extracting information mentioned in text documents, news articles or blog posts.
- Management Tools: *Stackdriver Monitoring*, to monitor cloud resources performance; *Stackdriver Logging*, for storing, monitoring, and alerting log data and events from google services.
- Identity and Security: *Cloud Identity & Access Management*, a service to establish a policy of access to the various services; *Cloud Key Management Service*, a key management for managing encryption of cloud services.

Microsoft Azure

Azure is an environment and a set of cloud services that can be used to develop cloud-oriented applications, or to enhance existing applications with cloud-based capabilities. The platform provides on-demand compute and storage resources exploiting the computational and storage power of the Microsoft data centers. Azure is designed for supporting high availability and dynamic scaling services that match user needs with a pay-per-use pricing model. The Azure platform can be used to perform the storage of large datasets, execute large volumes of batch computations, and develop SaaS applications targeted towards end-users.

Microsoft Azure includes three basic components/services as shown in Fig. 3:

- *Compute* is the computational environment to execute cloud applications. Each application is structured into roles: *Web role*, for Web-based applications; *Worker role*, for batch applications; *VM role*, for virtual-machine images.
- *Storage* provides scalable storage to manage: binary and text data (*Blobs*), non-relational tables (*Tables*), queues for asynchronous communication between components (*Queues*), and virtual disks (*Disks*).
- *Fabric controller* whose aim is to build a network of interconnected nodes from the physical machines of a single data center. The Compute and Storage services are built on top of this component.

Microsoft Azure provides also a set of services that can be used in many fields, like *Machine Learning*, to build cloud-based machine learning applications, or *IOT Hub*, to connect, monitor, and control IoT devices. Azure provides standard interfaces that allow developers to interact with its services. Moreover, developers can use IDEs like Microsoft Visual Studio and Eclipse to easily design and publish Azure applications.

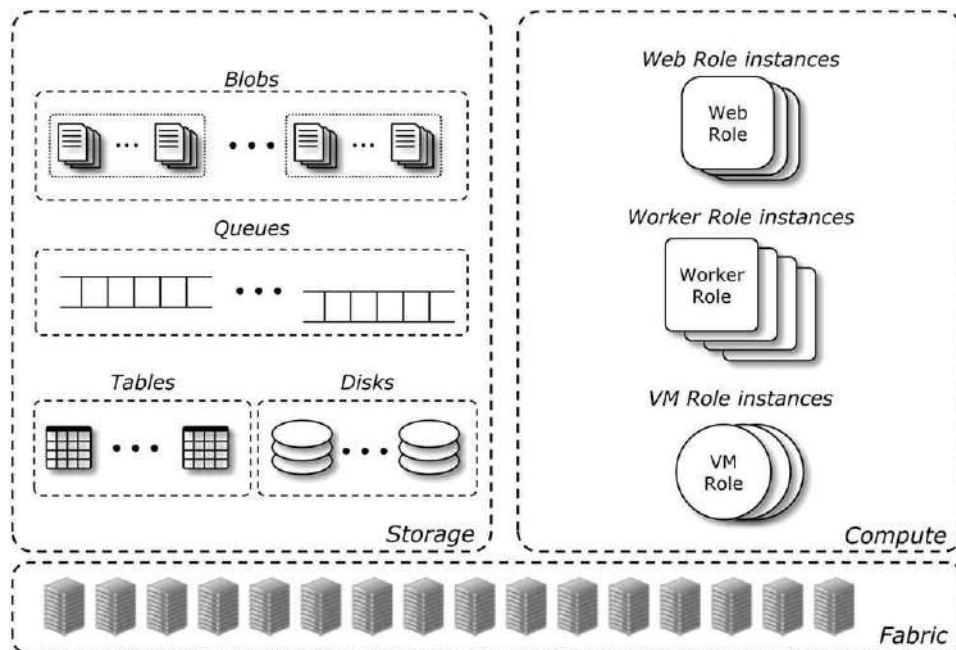


Fig. 3 Microsoft Azure.

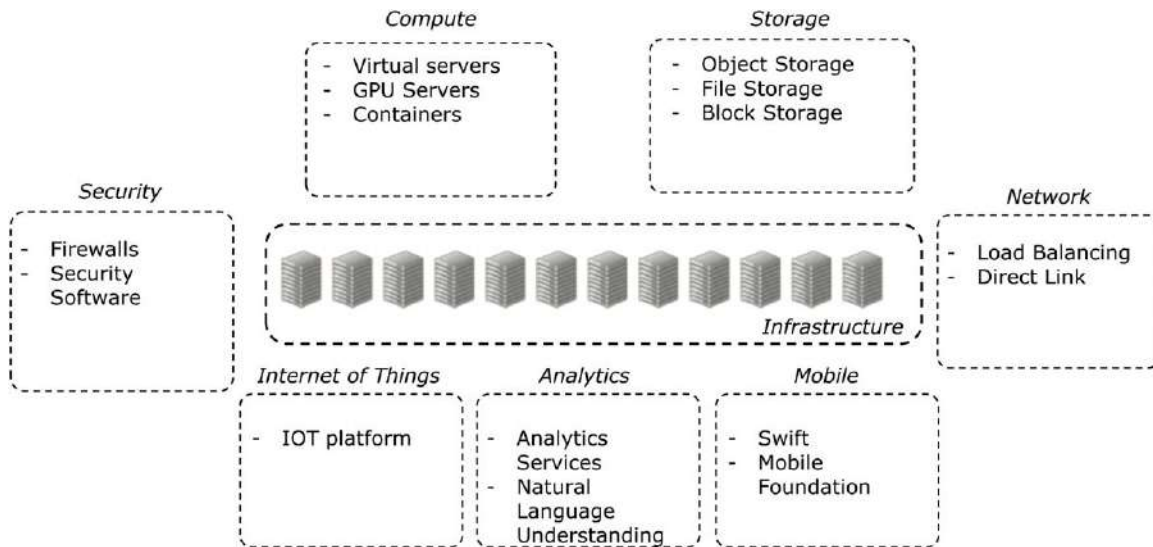


Fig. 4 IBM Bluemix.

IBM Bluemix

IBM Bluemix delivers services that can easily integrate with cloud applications without users needing to know how to install or configure them. It supports several programming languages and services for building, running and managing applications on the cloud (e.g., NET, Java, Node.js, PHP, Python, Ruby, Go).

With Bluemix customers can access three types of cloud computing platforms:

- *Bluemix public*, which provides all the necessary resources for an application development available to the general public;
- *Bluemix dedicated*, which provides a own Bluemix private environment that is hosted in an isolated container managed by Bluemix;
- *Bluemix Hybrid*, which allows customers to connect their dedicated services to the Public Bluemix services by maintaining data and sensitive services on premise.

IBM Bluemix offers mainly two compute services for defining applications: *Containers* and *OpenStack VMs*. *IBM Containers* permits developers to deploy, manage and run application on IBM platform by leveraging open-source Docker container technology (Merkel, 2014). Docker is an open platform used for building, shipping and running applications. *OpenStack Virtual Machines* allows running specific operating systems that users can customize.

The IBM Bluemix platform provides several services to build and scale an application (see Fig. 4):

- *Compute*: *Virtual servers* to deploy and scale virtual machines on demand; *GPU Servers* to deploy high-performance GPU servers; *Containers*, to deploy, manage and run application on IBM platform by leveraging open-source Docker container technology.
- *Storage*: *Object Storage*, to access unstructured data via with RESTful APIs; *File Storage*, a full-featured file storage; *Block Storage*, a scalable block storage services.
- *Network*: *Load Balancing*, a load-balancer for compute resources; *Direct Link*, to transfer data between a private infrastructure and IBM Bluemix services.
- *Mobile*: *Swift*, to write Swift mobile applications; *Mobile Foundation*, a set of APIs to manage the app life cycle.
- *Analytics*: *Analytics Services*, to extract insight from data; *Natural Language Understanding*, to extract meta-data from text such as sentiment, and relations.
- *Internet of Things*: *IOT platform*, which allows to communicate with and consume data from IOT devices.
- *Security*: *Firewalls* for protecting from malicious activities; *Security Software*, an intrusion protection system for the network and server/host level.

OpenStack

OpenStack is a cloud system that allows the management of large pools of processing, storage, and networking resources in a data center through a Web-based interface (Wen et al., 2012). The system has been designed, developed and released following four open principles:

- *Open source*: OpenStack is released under the terms of the Apache License 2.0;

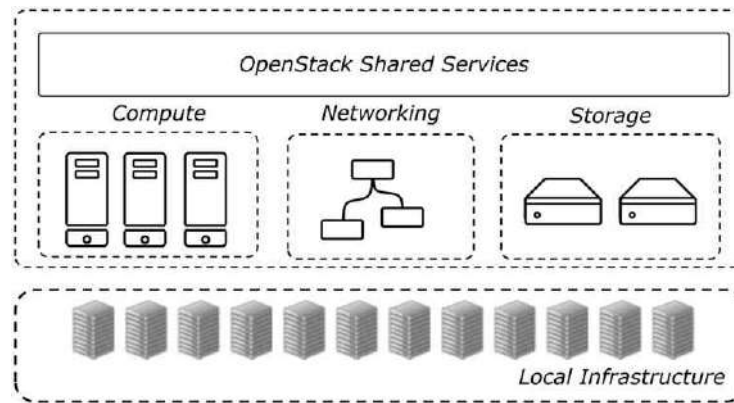


Fig. 5 OpenStack.

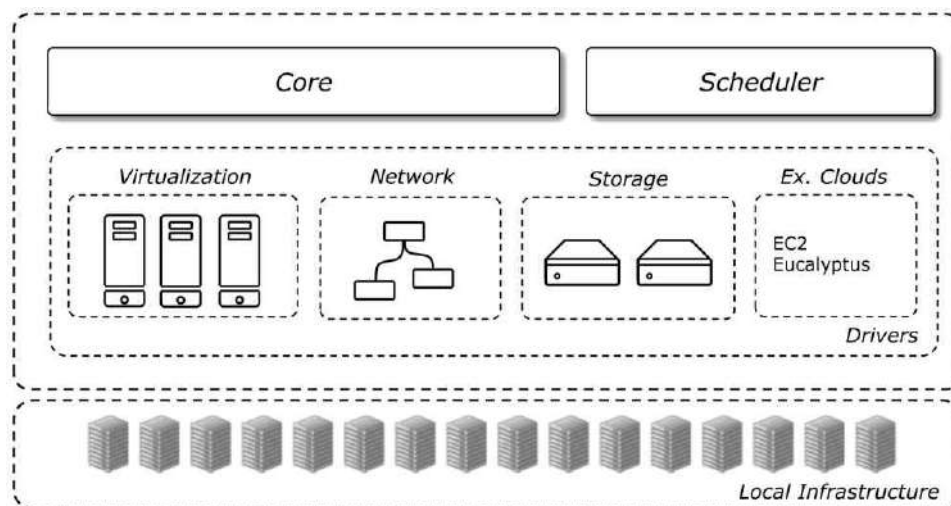


Fig. 6 OpenNebula.

- *Open design*: Every six months there is a design summit to gather requirements and define new specifications for the upcoming release;
- *Open development*: A publicly available source code repository is maintained for the entire development process;
- *Open Community*: Most decisions are made by the OpenStack community using a lazy consensus model.

The modular architecture of OpenStack is composed by four main components, as shown in Fig. 5.

OpenStack *Compute* provides virtual servers upon demand by managing the pool of processing resources available in the data center. It supports different virtualization technologies (e.g., VMware, KVM) and is designed to scale horizontally. OpenStack *Storage* provides a scalable and redundant storage system. It supports Object Storage and Block Storage: the former allows storing and retrieving objects and files in the data center; the latter allows creating, attaching and detaching of block devices to servers. OpenStack *Networking* manages the networks and IP addresses. Finally, OpenStack *Shared Services* are additional services provided to ease the use of the data center. For instance, Identity Service maps users and services, Image Service manages server images, Database Service provides a relational database.

OpenNebula

OpenNebula (Sotomayor et al., 2009) is an open-source framework mainly used to build private and hybrid clouds. The main component of the OpenNebula architecture (see Fig. 6) is the *Core*, which creates and controls virtual machines by interconnecting them with a virtual network environment. Moreover, the Core interacts with specific storage, network and virtualization operations through pluggable components called *Drivers*. In this way, OpenNebula is independent from the underlying infrastructure and offers a uniform management environment.

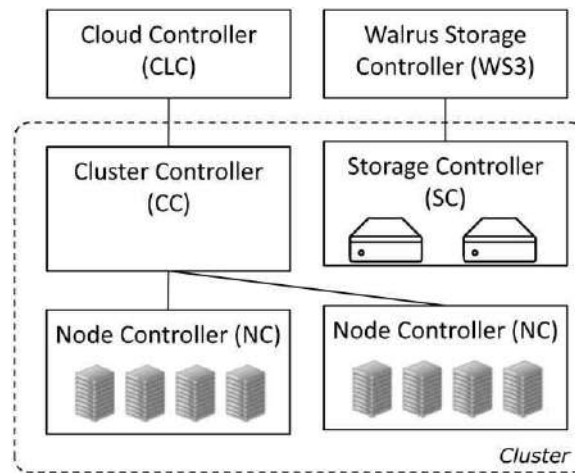


Fig. 7 Eucalyptus.

The Core also supports the deployment of *Services*, which are a set of linked components (e.g., Web server, database) executed on several virtual machines. Another component is the *Scheduler*, which is responsible for allocating the virtual machines on the physical servers. To this end, the Scheduler interacts with the Core component through appropriate deployment commands.

OpenNebula can implement a hybrid cloud using specific Cloud Drivers that allow interacting with external clouds. In this way, the local infrastructure can be supplemented with computing and storage resources from public clouds. Currently, OpenNebula includes drivers for using resources from Amazon EC2 and Eucalyptus.

Eucalyptus

Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) is an open source and paid software platform for implementing IaaS in a private or hybrid cloud computing environment (Endo *et al.*, 2010). Eucalyptus enables developers to use compute, storage, and network resources that can be dynamically scaled and can be used to build web-based applications. Companies can use AWS-compatible services, images, and scripts on their own on-premises IaaS environments, without using a public Cloud environment like AWS. It is simple to interact with Eucalyptus since it provides a console to self-service provision and configure compute, network, and storage resources.

Eucalyptus is based on five basic components as shown in Fig. 7. The *Node Controller* (NC) hosts the virtual machine instances and manages the virtual network endpoints. The *Cluster Controller* (CC) acts as the front end for a cluster and it is responsible for deploying and managing instances on NCs. The *Storage Controller* (SC) provides persistent block storage and allows the creation of snapshots of volumes. The *Cloud Controller* (CLC) is the front-end for managing cloud services and performs high-level resource scheduling and system accounting. CLC also allows interacting with the components of the Eucalyptus infrastructure providing an AWS-compatible web services interface to customers. *Walrus Storage Controller* (WS3) offers a persistent storage to all of the virtual machines in the Eucalyptus cloud and can be used to store machine images and snapshots. WS3 can be used with AWS S3 APIs.

Closing Remarks

This article described the main services provided by the most popular cloud infrastructures currently in use. The systems presented in this article were either public clouds Amazon Web Services (AWS), Google Cloud Platform, Microsoft Azure and IBM Bluemix or private clouds OpenStack, OpenNebula and Eucalyptus. As discussed above, the public cloud systems provide a variety of services that implement all the three main cloud service models, i.e., Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). On the other hand, the private cloud systems are more focused on the IaaS model, as they provide services for the management of large pools of processing, storage, and networking resources in a data center, by exploiting a variety of virtualization techniques.

See also: Computing Languages for Bioinformatics: Java. Dedicated Bioinformatics Analysis Hardware. Models and Languages for High-Performance Computing. Parallel Architectures for Bioinformatics. Text Mining Applications

References

- Endo, P.T., Gonçalves, G.E., Kelner, J., Sadok, D., 2010. A survey on open-source cloud computing solutions. In: Brazilian Symposium on Computer Networks and Distributed Systems, vol. 71.
- Merkel, D., 2014. Docker: Lightweight linux containers for consistent development and deployment. Linux Journal 2014, 239.
- Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I., 2009. Virtual infrastructure management in private and hybrid clouds. IEEE Internet Computing 13, 14–22.
- Talia, D., Trunfio, P., Marozzo, F., 2015. Data analysis in the Cloud. The Netherlands: Elsevier.
- Wen, X., Gu, G., Li, Q., Gao, Y., Zhang, X., 2012. Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In: Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2457–2461.

Relevant Websites

- <https://aws.amazon.com/>
Amazon Web Services.
- <https://www.docker.com/>
Docker.
- <http://www.eucalyptus.com/>
Eucalyptus.
- <https://cloud.google.com/>
Google Cloud Platform.
- <https://www.ibm.com/cloud-computing/bluemix/>
IBM Bluemix.
- <https://www.microsoft.com/azure>
Microsoft Azure.
- <https://www.openstack.org/>
OpenStack.
- <https://opennebula.org/>
OpenNebula.

Infrastructures for High-Performance Computing: Cloud Computing Development Environments

Fabrizio Marozzo and Paolo Trunfio, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Developing cloud applications may be a complex task, with specific issues that go beyond those of stand-alone application programming. For instance, cloud programming must deal with deployment, scalability and monitoring aspects that are not easy to handle without the use of ad-hoc environments (Talía *et al.*, 2015). In fact, to simplify the development of cloud applications, cloud computing development environments are often used. This article describes some of the most representative cloud computing development environments currently in use. The environment presented in this paper are classified into four types:

- *Integrated development environments*, which are used to code, debug, deploy and monitor cloud applications that are executed on a cloud infrastructure. The environments discussed in this article are *Eclipse*, *Visual Studio* and *IntelliJ*.
- *Parallel-processing development environments*, which are used to define parallel applications for processing large amount of data that are run on a cluster of virtual machines provided by a cloud infrastructure. The environments presented here are *Hadoop* and *Spark*.
- *Workflow development environments*, which are used to define workflow-based applications that are executed on a cloud infrastructure. The examples discussed here are *Swift* and *DMCF*.
- *Data-analytics development environments*, which are used to define data analysis applications through machine learning and data mining tools provided by a cloud infrastructure. The examples presented in this article are *Azure ML* and *BigML*.

Integrated Development Environments

Eclipse

Eclipse (see Section Relevant Websites) is one of the most popular integrated development environments (IDEs) for software programmers that can be used to define applications in C++ , Java, JavaScript, PHP, Python, R and so on, which can be run and deployed on multiple operating systems and computing platforms, including the most popular cloud computing infrastructures.

The Eclipse platform can be extended by installing plug-ins, such as development toolkits for novel programming languages and/or systems. Plug-ins can be programmed using Eclipse APIs and can be run on any of the supported operating systems. At the core of Eclipse is an architecture for discovering, loading, and running plug-ins. In addition to providing a development environment for programming languages, Eclipse supports development for most popular application servers (e.g., Tomcat, GlassFish) and is often capable of installing the required server directly from the IDE. It supports remote debugging that allows programmers to debug the code of applications running on servers.

Eclipse provides three types of products:

- *Desktop IDE*, for defining and running Java applications, C/C++ software, PHP web pages and so on in a desktop PC;
- *Cloud IDEs*, a Cloud IDE to develop software using a browser;
- *IDE Platforms*, a set of frameworks and common services to support the use of Eclipse as a component model.

Eclipse allows to program cloud applications for the main public and private cloud infrastructures. For example, *AWS Toolkit for Eclipse* is an open source plug-in that allows developers to define, debug, and deploy Java applications on Amazon Web Services. *IBM Eclipse Tools for Bluemix* enables the deployment and integration of many services from Bluemix into applications. Finally, *Google Plugin for Eclipse* simplifies the development of web applications that utilize Google cloud technology.

Visual Studio

Microsoft Visual Studio (see Section Relevant Websites) is a Microsoft IDE that allows users to develop, test, and deploy applications for the web, desktop, cloud, mobile, and game consoles. It is fully integrated with Microsoft technologies such as Windows API, Microsoft Office, Microsoft Azure and Windows Store.

Visual Studio includes a code editor for supporting code completion and refactoring. An integrated debugger helps to observe the run-time behaviour of programs and find problems. The functionality of the IDE can be enhanced through plug-ins, such as visual tools aiding in the development of GUI of web, desktop and mobile applications.

Visual Studio supports the most popular programming languages. For example, C++ for performance across a wide range of devices; Python for cross-platform scripting; R, for data processing; Node.js for scalable applications in JavaScript; C# as a multi-paradigm programming language for a variety of platforms, including the cloud.

Visual Studio allows to program cloud applications for Microsoft Azure and other cloud infrastructures. For example, *Visual Studio Tools for Azure* allows building, managing, and deploying cloud applications on Azure, whilst the *AWS Toolkit for Visual Studio* is a plugin that permits to develop, debug, and deploy .NET applications that use Amazon Web Services.

IntelliJ

IntelliJ (see Section Relevant Websites) is an IDE for web, mobile, cloud and enterprise development that supports languages like Java, JavaScript, Groovy and Scala. It is developed by the JetBrains company and is available in an Apache Licensed community edition, and in a commercial edition. The community edition allows Java and Android development. The commercial edition extends the community edition with support for web and enterprise development. Both community and commercial editions support cloud development with sponsorship of the main cloud providers.

One important feature of IntelliJ is code completion made by analyzing the source code of a user. Specifically, IntelliJ indexes user source code, for providing relevant code suggestions and on-the-fly code analysis. As the previous two systems discussed above, IntelliJ supports plugins for adding additional functionality to the IDE, such as version control systems (e.g., GIT), databases (e.g., Microsoft SQL Server) and automatic task runners (e.g., Grunt).

IntelliJ IDEA supports the most popular Java application servers, such as Tomcat, JBoss and Glassfish. A developer can deploy, debug and monitor an application onto an application server. Moreover, IntelliJ IDEA provides dedicated plug-ins that allows programmers to manage Docker virtual machines.

IntelliJ allows developers to create and interact with cloud applications using the API of the most popular cloud infrastructures. For example, through *AWS Manager Plugin* it provides integration with AWS services like EC2, RDS and S3, whilst *Cloud Tools for IntelliJ* is a Google-sponsored plugin that allows IntelliJ developers to interact with Google Cloud Platform services.

Parallel-Processing Development Environments

Hadoop

Apache Hadoop (see Section Relevant Websites) is commonly used to develop parallel applications that analyse big amounts of data. It can be adopted for developing parallel applications using many programming languages (e.g., Java, Ruby, Python, C++) based on the MapReduce programming model (Dean and Ghemawat, 2004) on a cluster or on a cloud platform. Hadoop relieves developers from having to deal with classical distributed computing issues, such as load balancing, fault tolerance, data locality, and network bandwidth saving. The Hadoop project is not only about the MapReduce programming model (Hadoop MapReduce module), as it includes other modules such as:

- *Hadoop Distributed File System (HDFS)*: a distributed file system providing fault tolerance with automatic recovery, portability across heterogeneous commodity hardware and operating systems, high-throughput access and data reliability.
- *Hadoop YARN*: a framework for cluster resource management and job scheduling.
- *Hadoop Common*: common utilities that support the other Hadoop modules.

With the introduction of YARN in 2013, Hadoop turns from a batch processing solution into a platform for running a large variety of data applications, such as streaming, in-memory, and graphs analysis. As a result, Hadoop became a reference for several other frameworks, such as: Giraph (see Section Relevant Websites) for graph analysis; Storm (see Section Relevant Websites) for streaming data analysis; Hive (see Section Relevant Websites), which is a data warehouse software for querying and managing large datasets; Pig (see Section Relevant Websites), which is as a dataflow language for exploring large datasets; Tez (see Section Relevant Websites) for executing complex directed-acyclic graph of data processing tasks; Oozie (see Section Relevant Websites), which is a workflow scheduler system for managing Hadoop jobs.

Hadoop is available in most cloud infrastructures. For example, the *HDInsight* service by Microsoft Azure, the *Amazon Elastic MapReduce (EMR)* service by AWS, and Google Cloud Dataproc service by Google Cloud.

Spark

Apache Spark (see Section Relevant Websites) is an open-source framework for in-memory data analysis and machine learning developed at UC Berkeley in 2009. It can process distributed data from several sources, such as HDFS, HBase, Cassandra, and Hive. It has been designed to efficiently perform both batch processing applications (similar to MapReduce) and dynamic applications like streaming, interactive queries, and graph analysis. Spark is compatible with Hadoop data and it can run in Hadoop clusters through the YARN module. However, in contrast to Hadoop's two-stage MapReduce paradigm in which intermediate data are always stored in distributed file systems, Spark stores data in a cluster's memory and queries it repeatedly so as to obtain better performance for several classes of applications (e.g., interactive jobs, real-time queries, and stream data) (Xin et al., 2013). The Spark project has different components:

- Spark Core contains the basic functionalities of the library such as for manipulating collections of data, memory management, interaction with distributed file systems, task scheduling, and fault recovery.

- Spark SQL provides API to query and manipulate structured data using standard SQL or Apache Hive variant of SQL.
- Spark Streaming provides an API for manipulating streams of data.
- GraphX is a library for manipulating and analyzing big graphs.
- MLlib is a scalable machine learning library on top of Spark that implements many common machine learning and statistical algorithms.

Several big companies and organizations use Spark for big data analysis purpose: for example, Ebay uses Spark for log transaction aggregation and analytics, Kelkoo for product recommendations, SK Telecom analyses mobile usage patterns of customers.

Similarly to Hadoop, most cloud infrastructures provide Spark as a service, like *IBM Analytics for Apache Spark*, *Azure HDInsight* and *Google Cloud Dataproc*.

Workflow Development Environments

Swift

Swift (Wilde *et al.*, 2011) is a implicitly parallel scripting language that runs workflows across several distributed systems, like supercomputers, clusters, grids and clouds. The Swift language has been designed at the University of Chicago and at the Argonne National Lab to provide users with a workflow-based language for cloud computing.

Swift separates the application workflow logic from runtime configuration. This approach allows a flexible development model. The Swift language allows invocation and running of external application code and allows binding with application execution environments without extra coding from the user. Swift/K is the previous version of the Swift language that runs on the Karajan grid workflow engine across wide area resources. Swift/T is a new implementation of the Swift language for high-performance computing. In this implementation, a Swift program is translated into an MPI program that uses the Turbine and ADLB runtime libraries for scalable dataflow processing over MPI. The Swift-Turbine Compiler (STC) is an optimizing compiler for Swift/T and the Swift Turbine runtime is a distributed engine that maps the load of Swift workflow tasks across multiple computing nodes. Users can also use Galaxy (Giardine *et al.*, 2005) to provide a visual interface for Swift.

The Swift language provides a functional programming paradigm where workflows are designed as a set of code invocations with their associated command-line arguments and input and output files. Swift is based on a C-like syntax and uses an implicit data-driven task parallelism (Justin *et al.*, 2014). In fact, it looks like a sequential language, but being a dataflow language, all variables are futures, thus execution is based on data availability. When input data is ready, functions are executed in parallel. Moreover, parallelism can be exploited through the use of the *for each* statement. The Turbine runtime comprises a set of services that implement the parallel execution of Swift scripts exploiting the maximal concurrency permitted by data dependencies within a script and by external resource availability. Swift has been used for developing several scientific data analysis applications, such as prediction of protein structures, modeling the molecular structure of new materials, and decision making in climate and energy policy.

Data Mining Cloud Framework

The Data Mining Cloud Framework (DMCF) is a software system developed at the University of Calabria for designing and executing data analysis workflows on clouds (Belcastro *et al.*, 2015). A Web-based user interface allows users to compose their applications and submit them for execution over cloud resources, according to a Software-as-a-Service (SaaS) approach.

The DMCF architecture has been designed to be deployed on different cloud settings. Currently, there are two different deployments of DMCF: (1) on top of a Platform-as-a-Service (PaaS) cloud, i.e., using storage, compute, and network APIs that hide the underlying infrastructure layer; (2) on top of an Infrastructure-as-a-Service (IaaS) cloud, i.e., using virtual machine images (VMs) that are deployed on the infrastructure layer.

The DMCF software modules can be grouped into *web components* and *compute components*. DMCF allows users to compose, check, and run data analysis workflows through a HTML5 web editor. The workflows can be defined using two languages: Visual Language for Cloud (VL4Cloud) (Marozzo *et al.*, 2016) and JavaScript for Cloud (JS4Cloud) (Marozzo *et al.*, 2015). Both languages use three key abstractions:

- *Data* elements, representing input files (e.g., a dataset to be analyzed) or output files (e.g., a data mining model).
- *Tool* elements, representing software tools used to perform operations on data elements (partitioning, filtering, mining, etc.).
- *Tasks*, which represent the execution of Tool elements on given input Data elements to produce some output Data elements.

The DMCF editor generates a JSON descriptor of the workflow, specifying what are the tasks to be executed and the dependency relationships among them. The JSON workflow descriptor is managed by the DMCF workflow engine that is in charge of executing workflow tasks on a set of workers (virtual processing nodes) provided by the cloud infrastructure. The workflow engine implements a data-drive task parallelism that assigns workflow tasks to idle workers as soon as they are ready to execute.

Data-Analytics Development Environments

Microsoft Azure Machine Learning

Microsoft Azure Machine Learning (Azure ML, see Section Relevant Websites) is a SaaS that provides a Web-based machine learning environment for the creation and automation of machine learning workflows. Through its user-friendly interface, data scientists and developers can perform several common data analysis/mining tasks on their data and automate their workflows.

Using its drag-and-drop interface, users can import their data in the environment or use special readers to retrieve data from several sources, such as Web URL (HTTP), OData Web service, Azure Blob Storage, Azure SQL Database, Azure Table. After that, users can compose their data analysis workflows where each data processing task is represented as a block that can be connected with each other through direct edges, establishing specific dependency relationships among them. Azure ML includes a rich catalog of processing tool that can be easily included in a workflow to prepare/transform data or to mine data through supervised learning (regression or classification) or unsupervised learning (clustering) algorithms. Optionally, users can include their own custom scripts (e.g., in R or Python) to extend the tools catalog. When workflows are correctly defined, users can evaluate them using some testing dataset. Users can easily visualize the results of the tests and find very useful information about models accuracy, precision and recall. Finally, in order to use their models to predict new data or perform real time predictions, users can expose them as Web services. Always through a Web-based interface, users can monitor the Web services load and use by time.

Azure Machine Learning is a fully managed service provided by Microsoft on its Azure platform; users do not need to buy any hardware/software nor manage virtual machine manually. One of the main advantage of working with Azure Machine Learning is its auto-scaling feature: models are deployed as elastic Web services so as users do not have to worry about scaling them if the models usage increased.

BigML

BigML (see Section Relevant Websites) is provided as a Software-as-a-Service (SaaS) for discovering predictive models from data sources and using data classification and regression algorithms. The distinctive feature of BigML is that predictive models are presented to users as interactive decision trees. The decision trees can be dynamically visualized and explored within the BigML interface, downloaded for local usage and/or integration with applications, services, and other data analysis tools. Extracting and using predictive models in BigML consists in multiple steps, as detailed in the following:

- *Data source setting and dataset creation.* A data source is the raw data from which a user wants to extract a predictive model. Each data source instance is described by a set of columns, each one representing an instance feature, or field. One of the fields is considered as the feature to be predicted. A dataset is created as a structured version of a data source in which each field has been processed and serialized according to its type (numeric, categorical, etc.).
- *Model extraction and visualization.* Given a dataset, the system generates the number of predictive models specified by the user, who can also choose the level of parallelism level for the task. The interface provides a visual tree representation of each predictive model, allowing users to adjust the support and confidence values and to observe in real time how these values influence the model.
- *Prediction making.* A model can be used individually, or in a group (the so-called ensemble, composed of multiple models extracted from different parts of a dataset), to make predictions on new data. The system provides interactive forms to submit a predictive query for a new data using the input fields from a model or ensemble. The system provides APIs to automate the generation of predictions, which is particularly useful when the number of input fields is high.
- *Models evaluation.* BigML provides functionalities to evaluate the goodness of the predictive models extracted. This is done by generating performance measures that can be applied to the kind of extracted model (classification or regression).

Closing Remarks

This article described four categories of cloud computing development environments. *Integrated development environments* are used to code, debug, deploy and monitor cloud applications that are executed on a cloud infrastructure. *Parallel-processing development environments* are used to define parallel applications for processing large amount of data that are run on a cluster of virtual machines provided by a cloud infrastructure. *Workflow development environments* are used to define workflow-based applications that are executed on a cloud infrastructure. *Data-analytics development environments* are used to define data analysis applications through machine learning and data mining tools provided by a cloud infrastructure. For each category, the article described some of the most representative cloud computing development environments currently in use, including their main features, provided services and available implementations.

See also: Dedicated Bioinformatics Analysis Hardware. Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Infrastructures for High-Performance Computing: Cloud Infrastructures. MapReduce in Computational Biology via Hadoop and Spark. Models and Languages for High-Performance Computing. Text Mining Applications

References

- Belcastro, L., Marozzo, F., Talia, D., Trunfio, P., 2015. Programming visual and script-based big data analytics workflows on clouds. *Advances in Parallel Computing* 26, 18–31.
- Dean, J., Ghemawat, S., 2004. MapReduce: Simplified data processing on large clusters. In: *Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI'04)*, San Francisco, CA.
- Giardine, B., Riemer, C., Hardison, R.C., *et al.*, 2005. Galaxy: A platform for interactive large-scale genome analysis. *Genome Res* 15, 1451–1455.
- Justin, M., Wozniak, J.M., Foster, I., 2014. Language features for scalable distributed-memory dataflow computing. In: *Proceedings Data-flow Execution Models for Extreme-scale Computing at PACT*.
- Marozzo, F., Talia, D., Trunfio, P., 2015. JS4Cloud: Script-based Workflow Programming for Scalable Data Analysis on Cloud Platforms. *Concurrency and Computation: Practice and Experience* 27 (17), 5214–5237.
- Marozzo, F., Talia, D., Trunfio, P., 2016. A workflow management system for scalable data mining on clouds. In: *IEEE Transactions On Services Computing (IEEE TSC)*.
- Talia, D., Trunfio, P., Marozzo, F., 2015. *Data analysis in the Cloud*. The Netherlands: Elsevier.
- Wilde, M., Hategan, M., Wozniak, J.M., *et al.*, 2011. Swift: A language for distributed parallel scripting. *Parallel Computing* 37 (9), 633–652.
- Xin, R.S., Rosen, J., Zaharia, M., *et al.*, 2013. Shark: SQL and rich analytics at scale. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13)*. New York, NY.

Relevant Websites

- <http://giraph.apache.org/>
Apache Giraph: The Apache Software Foundation.
- <http://hadoop.apache.org/>
Apache Hadoop: The Apache Software Foundation.
- <http://hive.apache.org>
Apache Hive: The Apache Software Foundation.
- <http://pig.apache.org>
Apache Pig: The Apache Software Foundation.
- <http://spark.apache.org>
Apache Spark: The Apache Software Foundation.
- <http://storm.apache.org>
Apache Storm: The Apache Software Foundation.
- <https://bigml.com>
BigML, Inc.
- <https://eclipse.org/>
Eclipse.
- <https://www.jetbrains.com/idea/>
IntelliJ.
- <https://azure.microsoft.com/en-us/services/machine-learning/>
Microsoft Azure.
- <https://www.visualstudio.com>
Microsoft Visual Studio.
- <http://oozie.apache.org/>
Oozie: The Apache Software Foundation.
- <http://tez.apache.org>
TEZ: The Apache Software Foundation.

Cloud-Based Bioinformatics Tools

Barbara Calabrese, University “Magna Graecia”, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The availability of high-throughput technologies, such as mass spectrometry, microarray, and next generation sequencing, and the application of genomics and pharmacogenomics studies of large populations, are producing an increasing amount of experimental and clinical data, as well as specialized databases spread over the Internet (Calabrese and Cannataro, 2015a,b). The availability of huge volumes of omics data poses new problems in the storage, preprocessing and analysis of data. Main challenges regard: (i) The efficient storage, retrieval and integration of experimental data; (ii) their efficient and high-throughput preprocessing and analysis; (iii) the building of reproducible “in silico” experiments; (iv) the annotation of omics data with pre-existing knowledge stored into ontologies (e.g., Gene Ontology) or specialized databases; (v) the integration of omics and clinical data.

Traditionally, the management and analysis of omics data was performed through the use of bioinformatics tools, often implemented as web services, whereas data were stored in geographically distributed biological databases. This computing and storage model do not allow to manage high volume data and therefore it is needed to adopt new computing models (Agapito *et al.*, 2017).

Cloud computing may play an important role in many phases of the bioinformatics analysis pipeline, from data management and processing, to data integration and analysis, including data exploration and visualization (Calabrese and Cannataro, 2015b; Dudley *et al.*, 2010). Currently, high performance computing is used to face the large processing power required when processing omics data, while web services and workflows are used to face the complexity of the bioinformatics pipeline that comprises several steps. Cloud computing may be the glue that put together those mainstreams technologies already used in bioinformatics (parallelism, service orientations, knowledge management), with the elasticity and ubiquity made available by the cloud (Schadt, 2011).

Moreover, by entering the data and software in the cloud and providing them as a service, it is possible to get a level of integration that improves the analysis and the storage of bioinformatics big-data. In particular, as a result of this unprecedented growth of data, the provision of data as a service (Data as a Service, DaaS) is of extreme importance. DaaS provides data storage in a dynamic virtual space hosted by the cloud and allows to have updated data that are accessible from a wide range of connected devices on the web. An example is represented by the DaaS of Amazon Web Services (AWS, see “Relevant Websites section”), which provides a centralized repository of public data sets (Fusaro *et al.*, 2011).

Despite the many benefits associated with cloud computing, it currently presents some issues and open problems such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks, that are particularly important when managing sensitive data such as the patients data stored and processed in genomics and pharmacogenomics studies, and more in general when clinical data are transferred to the cloud.

Cloud-Based Bioinformatics Tools

In recent years, there have been several efforts to develop cloud-based tools to execute different bioinformatics tasks (Dai *et al.*, 2012), e.g., read mapping applications, sequences alignment, gene expression analysis (Henry *et al.*, 2014). Some examples of academic and commercial SaaS (Software as a Service) bioinformatics tools are reported in the following section.

CloudBurst (Schatz, 2009) and *CloudAligner* (Nguyen *et al.*, 2011) are parallel read-mapping algorithms optimized for mapping next-generation sequence (NGS) data to the human genome and other reference genomes, for use in a variety of biological analyses including SNP discovery, genotyping and personal genomics. They use the open-source Hadoop implementation of MapReduce to parallelize execution using multiple compute nodes. Specifically, CloudAligner has been designed for more long sequences.

Balaur (Popic and Batzoglu, 2017) is a recent privacy preserving read mapping technique for hybrid clouds that securely outsources a significant portion of the read-mapping task to the public cloud, while being highly competitive with existing state-of-the-art aligners in speed and accuracy. At a high level, BALAUR can be summarized in the following two phases: (1) Fast identification of a few candidate alignment positions in the genome using the locality sensitive hashing (LSH) (on the secure private client) and (2) evaluation of each candidate using secure kmer voting (on the untrusted public server). The proposed approach can easily handle typical genome-scale datasets and is highly competitive with non-cryptographic state-of-the-art read aligners in both accuracy and runtime performance on simulated and real read data.

GENESIS (Gonzalez *et al.*, 2015) provides a cloud-based system that allows users to directly process and analyze next-generation sequencing (NGS) data through a user-friendly graphical user interface (GUI). The first objectives of the GENESIS (formerly GEM.app) platform are: (1) To assist scientists/clinicians in transferring and processing genomic data, (2) to produce accurate, high quality, and reproducible results, (3) to provide a highly available and scalable analytical framework for analyzing variant data, and (4) to provide tools for user-driven data-sharing and collaboration. Hereby, GENESIS enables users of varying

computational experience to iteratively test many different filtering strategies in a matter of seconds and to browse very large sets of full exomes and genomes in real-time.

DNANexus (see “Relevant Websites section”) allows next-generation sequence (NGS) analysis and visualization. It uses cloud computing from Amazon Web Services. Customers of DNANexus use those computational resources to run analysis programs on DNA sequence data and to store that data. The product includes applications for read mapping, RNA-seq, ChIP-seq, and genomic variant analysis.

Tute Genomics is a cloud-based clinical genome interpretation platform that enables researchers and clinicians to utilize human genome data for scientific discovery and individualized treatment. Tute Genomics has developed the most advanced analytical methods for genome analysis by incorporating proprietary machine-learning algorithms into a cloud based application that allows researchers to analyze and interpret entire human genomes and discover genes and biomarkers at an unprecedented rate.

TGex (see “Relevant Websites section”) provides rapid and accurate identification of causal mutations in rare diseases. TGex is a knowledge-driven Next Generation Sequencing (NGS) cloud-based analysis and interpretation solution based on the GeneCards Suite Knowledgebase. It uses VarElect, the NGS PhenoTyper, to score and prioritize variant-genes based on disease/phenotype of interest. TGex provides analysts with all the tools required for filtering and ranking variants, together with the ability to quickly assess evidence for the association between candidate genes and relevant phenotypes, all in one easy-to-use, consolidated view.

BaseSpace (see “Relevant Websites section”) offers a wide variety of NGS (next-generation sequencing) data analysis applications that are developed or optimized by Illumina, or from a growing ecosystem of third-party app providers. BaseSpace is a cloud platform to be directly integrated in to the industry’s leading sequencing platforms, with no cumbersome and time consuming data transfer steps.

CloudRS (Chen *et al.*, 2013) corrects sequencing errors in next-generation sequencing (NGS) data. CloudRS is a MapReduce application, based on the multiple sequence alignment (MSA) approach, that emulates the concept of the error correction algorithm of ALLPATHS-LG on the MapReduce framework. The software applies a two-stage majority voting mechanism for correcting errors in each ReadStack. It contains several modules with at least one mapper and/or reducer task.

Another Hadoop-based tool is *Crossbow* (Langmead *et al.*, 2009) that combines the speed of the short read aligner Bowtie with the accuracy of the SNP caller SOAPsnp to perform alignment and SNP detection for multiple whole-human datasets per day.

VAT (Variant Annotation Tool) (Habegger *et al.*, 2012) has been developed to functionally annotate variants from multiple personal genomes at the transcript level as well as to obtain summary statistics across genes and individuals. VAT also allows visualization of the effects of different variants, integrates allele frequencies and genotype data from the underlying individuals and facilitates comparative analysis between different groups of individuals. VAT can either be run through a command-line interface or as a web application. Finally, in order to enable on-demand access and to minimize unnecessary transfers of large data files, VAT can be run as a virtual machine in a cloud-computing environment.

Another cloud-computing pipeline for calculating differential gene expression in large RNA-Seq datasets is *Myrna* (Langmead *et al.*, 2010). Myrna integrates short read alignment with interval calculations, normalization, aggregation and statistical modeling in a single computational pipeline. After alignment, Myrna calculates coverage for exons, genes, or coding regions and differential expression using either parametric or non-parametric permutation tests. Myrna exploits the availability of multiple computers and processors where possible and can be run on the cloud using Amazon Elastic MapReduce, on any Hadoop cluster, or on a single computer (bypassing Hadoop entirely).

PeakRanger (Feng *et al.*, 2011) is a software package for the analysis in Chromatin Immuno-Precipitation Sequencing (ChIP-seq) technique. This technique is related to NGS and allows investigating the interactions between proteins and DNA. Specifically, PeakRanger is a peak caller software package that can be run in a parallel cloud computing environment to obtain extremely high performance on very large data sets.

For spectrometry-based proteomics research, *ProteoCloud* (Muth *et al.*, 2013) is a freely available, full-featured cloud-based platform to perform computationally intensive, exhaustive searches using five different peptide identification algorithms. ProteoCloud is entirely open source, and is built around an easy to use and cross-platform software client with a rich graphical user interface. This client allows full control of the number of cloud instances to initiate and of the spectra to assign for identification. It also enables the user to track progress, and to visualize and interpret the results in detail.

An environment for the integrated analysis of microRNA and mRNA expression data is provided by *BioVLAB-MMIA* (Lee *et al.*, 2012). Recently, a new version called BioVLAB-NGS, deployed on both Amazon cloud and on a high performance, public available server called MAHA, has been developed (Chae *et al.*, 2014). By utilizing next generation sequencing (NGS) data and integrating various bioinformatics tools and databases, BioVLAB-MMIA-NGS offers several advantages, such as a more accurate data sequencing for determining miRNA expression levels or the implementation of various computational methods for characterizing miRNAs.

Cloud4SNP (Agapito *et al.*, 2013) is a Cloud-based bioinformatics tool for the parallel pre-processing and statistical analysis of pharmacogenomics SNP microarray data. Cloud4SNP is able to perform statistical tests in parallel, by partitioning the input data set and using the virtual servers made available by the Cloud. Moreover, different statistical corrections such as Bonferroni, False Discovery Rate, or none correction, can be applied in parallel on the Cloud, allowing the user to choose among different statistical models, implementing a sort of parameter sweep.

AGX (see “Relevant Websites section”) is a cloud-based service that can analyze single vcf sample, compare tumour/normal samples and perform trio analysis. This online system permits to upload, access and analyze data anywhere with Internet connection. User could employ standard inheritance models like Autosomal dominant or improve it by checking quality of the genotype calls in trio samples.

Nephele (see “Relevant Websites section”) is a project from the National Institutes of Health (NIH) that brings together microbiome data and analysis tools in a cloud computing environment. *Nephele*’s advanced analysis pipelines include multiple stages of data processing, many of which can be configured by modifying parameters provided in the submission form.

Vivar (Sante *et al.*, 2014) represents a comprehensive analysis platform for the processing, analysis and visualization of structural variation based on sequencing data or genomic microarrays, enabling the rapid identification of disease loci or genes. *Vivar* allows user to scale analysis spreading the work load over multiple (cloud) servers, it has user access control to keep data safe but it is still easy to share, and it is easy expandable as analysis techniques advance.

GenomeVIP (Mashl *et al.*, 2017) performs variant discovery on Amazon’s Web Service (AWS) cloud or on local high-performance computing clusters. It provides a collection of analysis tools and computational frameworks for streamlined discovery and interpretation of genetic variants. The server and runtime environments can be customized, updated, or extended.

MyVariant.info uses a generalizable cloud-based model for organizing and querying biological annotation information. It utilizes the nomenclature from the Human Genome Variation Society (HGVS) to define its primary keys. The tool contains more than 500 variant-specific annotation types from dozens of resources, covering more than 334 million unique variants, including both coding and non-coding variants.

PorthoMCL (Tabari and Su, 2017) facilitates comparative genomics analysis through exploiting the exponentially increasing number of sequenced genomes. *PorthoMCL* is a parallel orthology prediction tool using the Markov Clustering algorithm (MCL). This application was developed to identify orthologs and paralogs in any number of genomes. It can be run on a wide range of high performance computing clusters and cloud computing platforms.

Falco (Yang *et al.*, 2017) is a cloud-based framework designed for multi-sample analysis of transcriptomic data in an efficient and scalable manner. *Falco* utilizes state-of-the-art big data technology of Apache Hadoop and Apache Spark to perform massively parallel alignment, quality control, and feature quantification of single-cell transcriptomic data in Amazon Web Service (AWS) cloud-computing environment.

Open Challenges

Omics data extracted by patients’ samples, as in pharmacogenomics studies, as well as other clinical data and exams (e.g., bio-images and/or bio-signals), are sensitive data and pose specific constraints in terms of privacy and security. On the other hand, cloud computing presents open issues such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks.

In general, omics and clinical data managed through a cloud are susceptible to unauthorized access and attacks. Specifically, in Johnson (2009), the author puts in evidence that storing huge volumes of patients’ sensitive medical data in third-party cloud storage is susceptible to loss, leakage or theft. The privacy risk of cloud environment includes the failure of mechanisms for separating storage, memory, routing, and even reputation between different tenants of the shared infrastructure. The centralized storage and shared tenancy of physical storage space means the cloud users are at higher risk of disclosure of their sensitive data to unwanted parties.

Moreover, threats to the data privacy in the cloud include spoofing identity, tampering with the data, repudiation, and information disclosure. In spoofing identity attack, the attacker pretends to be a valid user whereas data tampering involves malicious alterations and modification of the content. Repudiation threats are concerned with the users who deny after performing an activity with the data. Information disclosure is the exposure of information to the entities having no right to access information. The same threats prevail for the health data stored and transmitted on the third-party cloud servers.

Thus, it is needed to adopt secure protection schemes and cryptographic techniques to defend the sensitive information of the medical and biological records. There is considerable work on protecting data from privacy and security attacks. The NIST has developed specific guidelines to help consumers to protect their data in the Cloud (Grance and Jansen, 2012).

In Abbas and Khan (2014), a detailed discussion relative to the security and privacy requirements for cloud-based applications is reported. Specifically, the main requirements can be summarized in the following way:

- *Integrity*: It is needed to guarantee that the health data collected by a system or provided to any entity is true representation of the intended information and has not been modified in any way.
- *Confidentiality*: The health data of patients must be kept completely undisclosed to the unauthorized entities.
- *Authenticity*: The entity requesting access is authentic. In the healthcare systems, the information provided by the healthcare providers and the identities of the entities accessing such information must be verified.
- *Accountability*: An obligation to be responsible in light of the agreed upon expectations. The patients or the entities nominated by the patients should monitor the use of their health information whenever that is accessed at hospitals, pharmacies, insurance companies etc.
- *Audit*: It is needed to ensure that all the healthcare data is secure and all the data access activities in the e-Health cloud are being monitored.
- *Non-Repudiation*: Repudiation threats are concerned with the users who deny after performing an activity with the data. For instance, in the healthcare scenario neither the patients nor the doctors can deny after misappropriating the health data.
- *Anonymity*: It refers to the state where a particular subject cannot be identified. For instance, identities of the patients can be made anonymous when they store their health data on the cloud so that the cloud servers could not learn about the identity.

- *Unlinkability*: It refers to the use of resources or items of interest multiple times by a user without other users or subjects being able to interlink the usage of these resources. More specifically, the information obtained from different flows of the health data should not be sufficient to establish linkability by the unauthorized entities.

Finally, in the cloud, physical storages could be widely distributed across multiple jurisdictions, each of which may have different laws regarding data security, privacy, usage, and intellectual property. Data in the cloud may have more than one legal location at the same time, with differing legal consequences.

In the last years, to investigate these issues and propose secure solutions for genomic data, the NIH-funded National Center for Biomedical Computing iDASH (integrating Data for Analysis, ‘anonymization’ and SHaring) hosted the second Critical Assessment of Data Privacy and Protection competition. The aim is to assess the capacity of cryptographic technologies for protecting computation over human genomes in the cloud (Tang *et al.*, 2016). Data scientists were challenged to design practical algorithms for secure outsourcing of genome computation tasks in working software, whereby analyses are performed only on encrypted data. They were also challenged to develop approaches to enable secure collaboration on data from genomic studies generated by multiple organizations. The results of the competition indicated that secure computation techniques can enable comparative analysis of human genomes, but greater efficiency (in terms of compute time and memory utilization) are needed before they are sufficiently practical for real world environment.

Closing Remarks

Applications and services in bioinformatics data analysis pose quite demanding requirements. The fulfillment of those requirements could result in the development of comprehensive bioinformatics data analysis pipeline easy to use, available through the Internet, that may increase the knowledge in biology and medicine.

Naturally, the adoption of this technology with its benefits will determine a reduction of costs and the possibility of also providing new powerful services. However, a number of open issues and problems, such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks, that are particularly important when managing patients’ data stored and processed in the cloud, need to be considered in the adoption of cloud-based bioinformatics solutions.

See also: Computational Pipelines and Workflows in Bioinformatics. Dedicated Bioinformatics Analysis Hardware. Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Infrastructures for High-Performance Computing: Cloud Computing Development Environments. Infrastructures for High-Performance Computing: Cloud Infrastructures. Visualization of Biomedical Networks

References

- Abbas, A., Khan, S.U., 2014. A review on the state-of-the-art privacy preserving approaches in the e-Health Clouds. *IEEE Journal of Biomedical and Health Informatics* 18 (4), 1431–1441.
- Agapito, G., *et al.*, 2017. Parallel and cloud-based analysis of omics data: Modelling and simulation in medicine. In: *Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 519–526. St. Petersburg.
- Agapito, G., Cannataro, M., Guzzi, P.H., *et al.*, 2013. Cloud4SNP: Distributed analysis of SNP microarray data on the cloud. In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (BCB'13)*.
- Calabrese, B., Cannataro, M., 2015a. Bioinformatics and microarray data analysis on the cloud. In: Guzzi, P. (Ed.), *Microarray Data Analysis. Methods in Molecular Biology*, vol. 1375. New York, NY: Humana Press.
- Calabrese, B., Cannataro, M., 2015b. Cloud computing in healthcare and biomedicine. *Scalable Computing: Practice and Experience* 16 (1), 1–18.
- Chae, H., Rhee, S., Nephew, K.P., *et al.*, 2014. BioVLAB-MMIA-NGS: MicroRNA-mRNA integrated analysis using high throughput sequencing data. *Bioinformatics* 31 (2), 265–267.
- Chen, C., *et al.*, 2013. CloudRS: An error correction algorithm of high-throughput sequencing data based on scalable framework. In: *Proceedings of the IEEE International Conference on Big Data*, pp. 717–722. Santa Clara, California.
- Dai, L., Gao, X., Guo, Y., *et al.*, 2012. Bioinformatics clouds for big data manipulation. *Biology Direct* 7 (43).
- Dudley, J.T., Pouliot, Y., Chen, J.R., *et al.*, 2010. Translational bioinformatics in the cloud: An affordable alternative. *Genome Medicine* 2 (51).
- Feng, X., Grossman, R., Stein, L., 2011. PeakRanger: A cloud-enabled peak caller for ChIP-seq data. *BMC Bioinformatics* 12 (139).
- Fusaro, V.A., Patil, P., Gafni, E., *et al.*, 2011. Biomedical cloud computing with amazon web services. *PLOS Computational Biology* 7 (8).
- Gonzalez, M., Falk, M., Gai, X., Schüle, R., Zuchner, S., 2015. Innovative genomic collaboration using the GENESIS (GEM.app) platform. *Human Mutation* 36 (10), 950–956.
- Grance, T., Jansen, W., 2012. Guidelines on security and privacy in public cloud computing. National Institute of Standards and Technology (NIST), U.S. Department of Commerce. Special Publication, 800-144, Available at: <http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf>.
- Habegger, L., Balasubramanian, S., Chen, D.Z., *et al.*, 2012. VAT: A computational framework to functionally annotate variants in personal genomes within a cloud-computing environment. *Bioinformatics* 28 (17), 2267–2269.
- Henry, V.J., Bandrowski, A.E., Pepin, A.-S., Gonzalez, B.J., Desfeux, A., 2014. OMICtools: An informative directory for multi-omic data analysis. *Database: The Journal of Biological Databases and Curation*. bau069.
- Johnson, M.E., 2009. Data hemorrhages in the health-care sector. In: *Proceedings of Financial Cryptography and Data Security*, pp. 71–89. Barbados.
- Langmead, B., Hansen, K.D., Leek, J.T., 2010. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biology* 11 (R83).
- Langmead, B., Schatz, M.C., Lin, J., *et al.*, 2009. Searching for SNPs with cloud computing. *Genome Biology* 10, R134.

- Lee, H., Yang, Y., Chae, H., *et al.*, 2012. BioVLAB-MMIA: A cloud environment for microRNA and mRNA integrated analysis (MMIA) on amazon EC2. *IEEE Transactions on NanoBioscience* 11 (3). 266–272.
- Mashl, R.J., Scott, A.D., Huang, K., *et al.*, 2017. GenomeVIP: A cloud platform for genomic variant discovery and interpretation. *Genome Research* 27 (8). 1450–1459.
- Muth, T., Peters, J., Blackburn, J., *et al.*, 2013. ProteoCloud: A full-featured open source proteomics cloud computing pipeline. *Journal of Proteomics* 88, 104–108.
- Nguyen, T., Shi, W., Ruden, D., 2011. CloudAligner: A fast and full-featured MapReduce based tool for sequence mapping. *BMC Research Notes* 4 (171).
- Popic, V., Batzoglou, S., 2017. A hybrid cloud read aligner based on MinHash and kmer voting that preserves privacy. *Nature Communications* 8, 15311.
- Sante, T., Vergult, S., Volders, P.-J., *et al.*, 2014. ViVar: A comprehensive platform for the analysis and visualization of structural genomic variation. *PLOS ONE* 9 (12). e113800.
- Schadt, E.E., Linderman, M.D., Sorenson, J., *et al.*, 2011. Cloud and heterogeneous computing solutions exist today for the emerging big data problems in biology. *Nature Reviews Genetics* 12 (3). 224.
- Schatz, M.C., 2009. CloudBurst: Highly sensitive read mapping with MapReduce. *Bioinformatics* 25 (11). 1363–1369.
- Tabari, E., Su, Z., 2017. PorthoMCL: Parallel orthology prediction using MCL for the realm of massive genome availability. *Big Data Analytics* 2, 4.
- Tang, H., *et al.*, 2016. Protecting genomic data analytics in the cloud: State of the art and opportunities. *BMC Medical Genomics* 9 (63).
- Yang, A., Troup, M., Lin, P., Ho, J., 2017. Falco: A quick and flexible single-cell RNA-seq processing framework on the cloud. *Bioinformatics* 33 (5). 767–769.

Relevant Websites

- <https://agx.alapy.com>
ALAPY.
- <http://aws.amazon.com/publicdatasets>
AWS Free Tier.
- www.dnanexus.com
DNAnexus.
- www.illumina.com
Illumina.
- <https://nephele.niaid.nih.gov>
Nephele - NIH.
- <http://tgex.genecards.org/>
TGex - Knowledge Driven NGS Analysis.

Cloud-Based Bioinformatics Platforms

Barbara Calabrese, University “Magna Graecia”, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Research in life sciences are more and more based on the large datasets produced by high throughput platforms (e.g mass spectrometry, microarray, and next generation sequencing). The application of omics sciences, such as genomics, proteomics and interactomics, on large populations is producing an increasing amount of experimental data (“big data”), as well as specialized databases spread over the Internet ([Greene et al., 2014](#)).

However, the storage, preprocessing and analysis of this experimental data are becoming the main bottleneck of the bioinformatics analysis pipeline, by posing several issues. In fact, as the amount of digital information increases, the ability to manage this data becomes a growing problem. This data holds the keys to future clinical advances, but often remains inaccessible to researchers.

Cloud computing is a computing model that has spread very rapidly in recent years for the supply of IT resources of different nature, through services accessible via the network. Specifically, Cloud can provide Virtual Machines (VM) in the sense of hardware resources, platforms to deploy applications or ready-to-use services ([Mell and Grance, 2011](#)). This computing model guarantees new advantages related to massive and scalable computing resources available on demand, virtualization technology and payment for use as needed. Therefore, the role of cloud computing has become crucial in many steps of the bioinformatics analysis pipeline, from data management and processing, to data integration and analysis, including data exploration and visualization. Cloud Computing can be the enabling factor for data sharing and integration at a large scale. In the paper ([Merelli et al., 2014](#)), the authors discussed high performance computing (HPC) solutions in bioinformatics and Big Data analysis paradigms for computational biology. Specifically, the authors pointed out that cloud computing addresses big data storage and analysis issues in many fields of bioinformatics, thanks to the virtualization that avoid to move too big data.

Background/Fundamentals

A working definition of cloud computing comes from the work of [Mell and Grance \(2011\)](#) from the National Institute of Standards and Technology, United States Department of Commerce. Their definition focuses on computing resources which can be accessed from anywhere and may be provisioned online. It also specifies five characteristics of cloud computing, three service models and four deployment methods ([Calabrese and Cannataro, 2015a](#)). According to this definition, the five key characteristics of cloud services are:

- on demand self service: a consumer can acquire computing power unilaterally and automatically without requiring any human intervention by the service provider;
- broad network access: the computational capabilities are accessible on the Internet in accordance with standard mechanisms;
- resource pooling: users have the impression that the available resources are unlimited and can be purchased in any amount and at any time. Resources of the service provider come together to serve a variety of consumers, according to a multi-tenant model. The physical and virtual resources are dynamically assigned and reassigned to consumers, based on their requests without having any control or knowledge of the exact location of the resources assigned to him/her;
- rapid elasticity: resources are able to be quickly allocated and elastically;
- measured services: the cloud systems automatically control and optimize resource use by evaluating appropriate parameters (e.g. storage, processing power, bandwidth and active user accounts).

[Vaquero et al. \(2009\)](#) collected 22 excerpts from previous works and fused these into a single definition that emphasizes the importance of Service Level Agreements (SLA) in order to increase confidence in the cloud environment and defines virtualization as the key enabler of cloud computing. [Hill et al., \(2013\)](#) gives a recent definition of cloud computing that presents cloud computing as a utility.

Service Models

Cloud services can be classified into three main models:

- Infrastructure as a Service (IaaS): this service mode includes servers (typically virtualized) with specific computational capability and/or storage. The user can control all the storage resources, operating systems and applications deployed to, while he/she has limited control over the network settings. An example is Amazon's Elastic Compute Cloud (EC2), which allows the user to create virtual machines and manage them, and Amazon Simple Storage Service (S3), which allows to store and access data, through a web-service interface.

- Platform as a Service (PaaS): it allows the development, installation and execution on its infrastructure of user-developed applications. Applications must be created using specific programming languages, libraries and tools supported by the provider that constitute the development platform provided as a service. An example is Google Apps Engine, which allows to develop applications in Java and Python and provides for both languages the SDK or Microsoft Azure.
- Software as a Service (SaaS): customers can use the applications provided by the cloud provider. The applications are accessible as web-services through a specific interface. Customers do not manage the cloud infrastructure or network components, servers, operating systems or storage.

Delivery Models

Cloud services can be made available to users in different ways. In the following, a brief description of the delivery models is presented:

- Public cloud: public cloud services are offered by vendors who provide the users/customers the hardware and software resources of their data centers. Examples of public clouds are Amazon (Amazon EC2 provides computational services, such as Amazon S3 storage services); Google Apps (which provides software services like Gmail, Google Docs or Google Calendar, and development platform like Google App Engine); and Microsoft Azure, that is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters. It provides both PaaS and IaaS services and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.
- Private cloud: private cloud is configured by a user or by an organization for its exclusive use. Services are supplied by computers that are in the domain of the organization. To install a private cloud, several commercial and free tools are available, Eucalyptus, Open Nebula, Terracotta and VMWare Cloud.
- Community cloud: it is an infrastructure on which are installed cloud services shared by a community or by a set of individuals, companies and organizations that share a common purpose and that have the same needs. The cloud can be managed by the community itself or by a third party (typically a cloud service provider).
- Hybrid cloud: the cloud infrastructure is made up of two or more different clouds using different delivery models, which, while remaining separate entities, are connected by proprietary or standard technology that enables the portability of data and applications.

Bioinformatics Cloud-Based Services

The cloud computing represents a cost-effective solution for the problems of storing and processing data in the context of bioinformatics. Thanks to the progress of high-throughput sequencing technologies, there was an exponential growth of biological data. Therefore, classical computational infrastructure for data processing have become ineffective and difficult to maintain (Schadt *et al.*, 2011; Grossmann and White, 2011).

The traditional bioinformatics analysis involves downloading of public datasets (e.g., NCBI, Ensembl), installing software locally and analysis in-house. By entering the data and software in the cloud and providing them as a service, it is possible to get a level of integration that improves the analysis and the storage of bioinformatics big-data (Calabrese, 2015b; Dai *et al.*, 2012). Thus, bioinformatics research institutes no longer have to invest money in costly infrastructures to run expensive scientific analysis (Dudley *et al.*, 2010).

In recent years, several cloud-based tools (Software as a Services, SaaS) to execute different bioinformatics tasks, such as genomics analysis, sequences alignment, gene expression analysis, have been developed (Agapito *et al.*, 2017). A bioinformatics Platform as a Service (PaaS), unlike SaaS, allows the users to customize the deployment of their solutions and to completely control their instances and associated data. Currently, different Cloud-based platforms are available to researchers (Henry *et al.*, 2014). A bioinformatics Infrastructure as a Service (IaaS) model is offered in a computing infrastructure that includes servers (typically virtualized) with specific computational capability and/or storage.

Bioinformatics Platforms Deployed as PaaS

In the following section, the main features and functionalities of specific bioinformatics platforms are discussed. They use Hadoop MapReduce implementation for parallelization (O'Driscoll *et al.*, 2013). MapReduce algorithm splits large inputs for independent processes and then combines the results with HDFS (Hadoop Distributed File System).

Currently, the most used platform (PaaS) for bioinformatics applications is CloudMan (Afgan *et al.*, 2011). It is a public Galaxy Cloud deployment on AWS (Amazon Web Services) cloud; it is, also, compatible with Eucalyptus and other clouds. Galaxy is a bioinformatic platform that collects the main services and tools for several bioinformatics analysis, providing a simple user-friendly interface. CloudMan (see Relevant Websites section) is a CloudManager that orchestrates all of the steps required to the provision, configuration, management and sharing Galaxy on a cloud computing infrastructure, using a web browser. CloudMan combines advantages of Galaxy framework and the benefits of cloud computing model such as elasticity, and pay as you go (Afgan *et al.*, 2012).

cl-dash (see Relevant Websites section) is a complete framework that allows to researchers a quick setup of a Hadoop computing environment in AWS cloud environment. Critical features of the tool include rapid and simple creation of new clusters of different sizes, flexible configuration and custom software installation, data persistence during intermittent usage and low cost of initial investment and during operation (Hodor *et al.*, 2016).

DNAxenus (see Relevant Websites section) allows next-generation sequence (NGS) analysis and visualization. DNAxenus is a cloud-based solution to manage specific genomic data. User can create projects, share them with other DNAxenus users at various access levels, run apps and analysis and build a workflow.

CloudGene (see Relevant Websites section) is an open source platform to improve the usability of MapReduce programs in bioinformatics (Schönherr *et al.*, 2012). Cloudgene provides a graphical user interface for the execution, the import and export of data and the reproducibility of workflows on in-house (private clouds) and rented clusters (public clouds). The aim of Cloudgene is to build a standardized graphical execution environment for currently available and future MapReduce programs, which can all be integrated by using its plug-in interface.

CloudDOE (see Relevant Websites section) is a platform-independent software package implemented in Java. CloudDOE encapsulates technical details behind a user-friendly graphical interface, thus liberating scientists from having to perform complicated operational procedures. Users are guided through the user interface to deploy a Hadoop cloud within in-house computing environments and to run applications specifically targeted for bioinformatics (Chung *et al.*, 2014).

Arvados (see Relevant Websites section) is a platform for storing, organizing, processing, and sharing genomic and other big data. The platform is designed to make it easier for data scientists to develop analyses, developers to create genomic web applications and IT administrators to manage large-scale compute and storage genomic resources. Functionally, Arvados has two major sets of capabilities: (a) data management and (b) compute management. The data management services are designed to handle all of the challenges associated with storing and organizing large omic data sets. The compute management services are designed to handle the challenges associated with creating and running pipelines as large scale distributed processing jobs.

Eoulsan (see Relevant Websites section) is a framework based on the Hadoop implementation of the MapReduce algorithm, dedicated to high throughput sequencing data analysis on distributed computers (Jourden *et al.*, 2012). Specifically, it has been developed in order to automate the analysis of a large number of samples at once, simplify the configuration of the cloud computing infrastructure and work with various already available analysis solutions.

Closing Remarks

Omics disciplines are gaining an increasing interest in the scientific community due to the availability of high throughput platforms and computational methods which are producing an overwhelming amount of omics data. The increased availability of omics data poses new challenges both for the efficient storage and integration of the data and for their efficient preprocessing and analysis. Cloud computing is playing an important role in all steps of the life sciences research pipeline, from raw data management and processing, to data integration and analysis, up to data exploration and visualization.

See also: Computational Pipelines and Workflows in Bioinformatics. Dedicated Bioinformatics Analysis Hardware. Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Infrastructures for High-Performance Computing: Cloud Computing Development Environments. Infrastructures for High-Performance Computing: Cloud Infrastructures

References

- Afgan, E., Baker, D., Coraor, N., *et al.*, 2011. Harnessing cloud computing with Galaxy Cloud. *Nature Biotechnology* 29 (11), 972–974.
- Afgan, E., Chapman, B., Taylor, J., 2012. CloudMan as a platform for tool, data and analysis distribution. *BMC Bioinformatics* 13 (315).
- Agapito, G., Calabrese, B., Guzzi, P.H. *et al.*, 2017. Parallel and Cloud-Based Analysis of Omics Data: Modelling and Simulation. In: *Proceedings of the 25th Euromicro International Conference on Medicine, Parallel, Distributed and Network-based Processing (PDP)*.
- Calabrese, B., Cannataro, M., 2015a. Cloud computing in healthcare and biomedicine. *Scalable Computing: Practice and Experience* 16 (1), 1–17.
- Calabrese, B., Cannataro, M., 2015b. Bioinformatics and microarray data analysis on the cloud. In: Guzzi, P. (Ed.), *Microarray Data Analysis. Methods in Molecular Biology* 1375. New York, NY: Humana Press.
- Chung, W.C., Chen, C.C., Ho, J.M., *et al.*, 2014. CloudDOE: A user-friendly tool for deploying Hadoop clouds and analyzing high-throughput sequencing data with MapReduce. *PLOS One* 9 (6), e98146.
- Dai, L., Gao, X., Guo, Y., Xiao, J., Zhang, Z., 2012. Bioinformatics clouds for big data manipulation. *Biology Direct* 7 (43).
- Dudley, J.T., Pouliot, Y., Chen, J.R., Morgan, A.A., Butte, A.J., 2010. Translational Bioinformatics in the cloud: An affordable alternative. *Genome Medicine* 2, 51.
- Greene, C.S., Tan, J., Ung, M., Moore, J.H., Cheng, C., 2014. Big data bioinformatics. *Journal of Cell Physiology* 229 (12), 1896–1900.
- Grossmann, R.L., White, K.P., 2011. A vision for a biomedical cloud. *Journal of Internal Medicine* 271 (2), 122–130.
- Henry, V.J., Bandrowski, A.E., Pepin, A., Gonzalez, B.J., Desfeux, A., 2014. OMICtools: An informative directory for multi-omic data analysis. *Database (Oxford)*.
- Hill, R., Hirsch, L., Lake, P., Moshiri, S., 2013. *Guide to Cloud Computing*. Springer.
- Hodor, P., Chawla, A., Clark, A., Neal, L., 2016. cl-dash: Rapid configuration and deployment of Hadoop clusters for bioinformatics research in the cloud. *Bioinformatics* 32 (2), 301–303.

- Jourdren, L., Bernard, M., Dillies, M.A., Le Crom, S., 2012. Eoulsan: A cloud computing-based framework facilitating high throughput sequencing analyses. *Bioinformatics* 11 (28), 1542–1543.
- Mell, P., Grance T., 2011. The NIST Definition of Cloud Computing.
- Merelli, I., Prez-Sánchez, H., Gesing, S., D'Agostino, D., 2014. Managing, Analysing, and Integrating Big Data in Medical Bioinformatics: Open Problems and Future Perspectives. *BioMed Research International*.
- O'Driscoll, A., Daugelaite, J., Sleator, R.D., 2013. 'Big data', Hadoop and cloud computing in genomics. *Journal of Biomedical Informatics* 46, 774–781.
- Schadt, E.E., Linderman, M.D., Sorenson, J., Lee, L., Nolan, G.P., 2011. Cloud and heterogeneous computing solutions exist today for the emerging big data problems in biology. *Nature Reviews Genetics* 12 (3), 224.
- Schönherr, *et al.*, 2012. Cloudgene: A graphical execution platform for MapReduce programs on private and public clouds. *BMC Bioinformatics* 13, 200.
- Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M., 2009. A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Computer Communication Review* 39, 50–55.

Relevant Websites

<https://dev.arvados.org/projects/arvados>
Arvados.

<https://bitbucket.org/booz-allen-sci-comp-team/cl-dash/overview>
cl-dash.

<http://clouddoe.iis.sinica.edu.tw>
CloudDOE.

<http://cloudgene.uibk.ac.at>
CloudGene.

<https://galaxyproject.org/cloudman>
CloudMan.

<https://www.dnanexus.com>
DNAnexus.

<http://outils.genomique.biologie.ens.fr/eoulsan2/>
Eoulsan.

Cloud-Based Molecular Modeling Systems

Barbara Calabrese, University “Magna Graecia”, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Molecular modelling is based on the development of theoretical and computational methodologies, to model and study the behaviour of molecules, from small chemical systems to large biological molecules and material assemblies. The application fields of molecular modelling regard computational chemistry, drug design, computational biology and materials science. The basic computational technique to perform molecular modelling is simulation. Molecular simulation techniques requires specific additional computational and software requirements (Ebejer *et al.*, 2013).

Cloud computing could be used by computational scientists and engineers as a means to run molecular simulations and analysis tasks (Xu *et al.*, 2015). Running molecular simulation and analysis tasks in the Cloud could significantly lower the costs of molecular modelling analysis. The cloud computing model provides researchers the use of pay-on-demand computational equipment by a third-party (cloud provider), accessed through an Internet connection (Calabrese and Cannataro, 2015).

Cloud computing model guarantees scalability advantage because computing resources can be allocated dynamically according to the specific needs of molecular simulation projects. For medium size and little size research centres and/or companies, this issue is fundamental because as soon as the project is done, the cloud resources can be deallocated, avoiding the usage and maintenance of highly costly cluster installed locally. In fact, before, massive simulations were prohibitive due to very long run times (in the order of months) and because of the high cost to establish and maintain powerful computational equipments. In PaaS and SaaS solutions, cloud providers are responsible of maintenance and updating.

The use of cloud computing for molecular modelling applications is still in its infancy, because in order to use cloud computing, however, many technical tasks must be performed by the user such as configuring the compute nodes, installing the required software, and launching, monitoring and terminating the computation (Wong and Goscinski, 2012). Other bottleneck is the waiting time to push the data into the cloud. To overcome this limit, data and services could be pushed together on the cloud. In the last years, several SaaS front-ends that facilitate the use of cloud computing have been proposed.

Molecular Modelling Tasks

Molecular modelling and simulation imply a multi-step process and combine different analysis methods. Modelling can help scientists at many levels, from the precisely detailed quantum mechanics and classic molecular modelling to process engineering modelling.

In the following, some of the typical molecular modelling analysis are listed: for each class of analysis, an example of software system is described.

Ab initio protein structure modelling: Ab initio folding (or free-modelling, FM) in protein structure prediction refers to build 3D structure models without using homologous structures as templates. Although many efforts have been devoted to the study of the folding of protein structures based on physics-based force fields, including molecular dynamics simulations that determine each atom's position based on Newton's laws of motion, the most efficient approach for FM structure prediction still exploits the knowledge and information derived from the Protein Data Bank (PDB) library (Berman *et al.*, 2000) and sequence databases (Xu and Zhang, 2012).

QUARK (see “Relevant Websites section”) is an example of computer algorithm for ab initio protein structure prediction and protein peptide folding, which aims to construct the correct protein 3D model from amino acid sequence only (Xu and Zhang, 2011). QUARK models are built from small fragments (1–20 residues long) by replica-exchange Monte Carlo simulation under the guide of an atomic-level knowledge-based force field. QUARK was ranked as the No 1 server in Free-modelling (FM) in CASP9 and CASP10 experiments. Since no global template information is used in QUARK simulation, the server is suitable for proteins that do not have homologous templates in the PDB library (Zhang *et al.*, 2016).

Remote homology detection: In computational biology, protein remote homology detection is the classification of proteins into structural and functional classes given their amino acid sequences, especially, with low sequence identities (Chen *et al.*, 2016). Protein remote homology detection is a critical step for basic research and practical application, which can be applied to the protein 3D structure and function prediction.

HHpred (see “Relevant Websites section”) is a fast server for remote protein homology detection and structure prediction and is the first to implement pairwise comparison of Hidden Markov Models (HMMs) profiles (Söding *et al.*, 2005). It allows to search a wide choice of databases, such as the PDB, SCOP (Structural Classification of Proteins) (Murzin *et al.*, 1995), and Pfam (Finn *et al.*, 2016). It accepts a single query sequence or a multiple alignment as input. Within only a few minutes it returns the search results in a user-friendly format similar to that of PSI (Position Specific Iterated)-BLAST (Altschul *et al.*, 1997).

Fragment-based protein structure modelling: Computational methods for protein structure prediction include: (i) Simulated folding using physics-based or empirically-derived energy functions, (ii) construction of the model from small fragments of known

structure, threading where the compatibility of a sequence with an experimentally- derived fold is determined using similar energy functions, and (iii) template-based modelling, where a sequence is aligned to a sequence of known structure based on patterns of evolutionary variation. Template-based modelling has become the most reliable and used techniques in the molecular modelling field, for three main reasons: (i) The development of powerful statistical techniques to extract evolutionary relationships from homologous sequences; (ii) the enormous growth in sequencing projects which provides the raw information; and (iii) the power of computing to process large databases with a fast turn-around.

Today, the most widely used and reliable methods for protein structure prediction rely on some method to compare a protein sequence of interest to a large database of sequences, construct an evolutionary/statistical profile of that sequence, and subsequently scan this profile against a database of profiles for known structures.

Phyre and *Phyre2* (Protein Homology/AnalogY Recognition Engine; pronounced as ‘fire’) are web-based services for protein structure prediction that are free for non-commercial use. *Phyre* is among the most popular methods for protein structure prediction. Like other remote homology recognition technique, it is able to regularly generate reliable protein models when other widely used methods such as PSI-BLAST cannot. *Phyre2* (see “Relevant Websites section”) is the successor of *Phyre* and it was launched in January 2011. *Phyre2* is one of the most widely used protein structure prediction servers and serves approximately 40,000 unique users per year, processing approximately 700–1000 user- submitted proteins per day. *Phyre2* has been designed to ensure a user-friendly interface for users inexperienced in protein structure prediction methods (Kelley *et al.*, 2015).

Macromolecular visualization: Visualization of macromolecular structure in three dimensions is becoming ever more important for understanding protein structure-function relationship, functional consequences of mutations, mechanisms of ligand binding, and drug design (Krawetz and Womble, 2003).

VMD (Visual Molecular Dynamics, see “Relevant Websites section”) is designed for modelling, visualization, and analysis of biological systems such as proteins, nucleic acids, lipid bilayer assemblies, etc. (Humphrey *et al.*, 1996). It may be used to view more general molecules, as VMD can read standard Protein Data Bank (PDB) files and display the contained structure. VMD provides a wide variety of methods for rendering and coloring a molecule: simple points and lines, CPK spheres and cylinders, licorice bonds, backbone tubes and ribbons, cartoon drawings, and others. VMD can be used to animate and analyze the trajectory of a molecular dynamics (MD) simulation. In particular, VMD can act as a graphical front end for an external MD program by displaying and animating a molecule undergoing simulation on a remote computer.

UCSF Chimera (see “Relevant Websites section”) is another example of highly extensible program for interactive visualization and analysis of molecular structures and related data, including density maps, supramolecular assemblies, sequence alignments, docking results, trajectories, and conformational ensembles (Pettersen *et al.*, 2004). High-quality images and animations can be generated. Chimera includes complete documentation and several tutorials, and can be downloaded free of charge for academic, government, nonprofit, and personal use. Chimera is developed by the Resource for Biocomputing, Visualization, and Informatics (RBVI), funded by the National Institutes of Health.

Molecular dynamics: Molecular dynamics (MD) is a computer simulation method for studying the physical movements of atoms and molecules. The atoms and molecules are allowed to interact for a fixed period of time, giving a view of the dynamic evolution of the system.

NAMD (see “Relevant Websites section”) is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. Based on Charm++ parallel objects, NAMD scales to hundreds of cores for typical simulations and beyond 500,000 cores for the largest simulations (Phillips *et al.*, 2005). NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER (see “Relevant Websites section”) (Salomon-Ferrer *et al.*, 2013; Case *et al.*, 2005) and CHARMM (see “Relevant Websites section”) (Brooks *et al.*, 2009). NAMD is distributed free of charge with source code.

Cloud-Based Services for Molecular Modelling Tasks

One of the most sophisticated SaaS for cloud computing is provided by commercial providers such as Cycle-Cloud (cyclecomputing.com), which offers clusters with preinstalled applications that are widely used in the bioinformatics, proteomics, and computational chemistry areas. The most interesting ones, from a molecular modelling perspective, are: (i) Gromacs (Pronk *et al.*, 2013), to perform molecular dynamics (MD) simulations and energy minimization, and (ii) ROCS to perform shape-based screening of compounds. The idea is that molecules have similar shape if their volumes overlay well and any volume mismatch is a measure of similarity.

The Rosetta software suite (see “Relevant Websites section”) includes algorithms for computational modelling and analysis of protein structures. It has enabled notable scientific advances in computational biology, including de novo protein design, enzyme design, ligand docking, and structure prediction of biological macromolecules and macromolecular complexes. A Seattle biomedical software company named Insilicos announced in the 2012 the formation of RosettaATCloud LLC, a joint venture with the University of Washington, that developed Rosetta@cloud, which offers the Rosetta software suite as a cloud-based service hosted on Amazon Web Services. Rosetta@cloud was a startup that offered affordable, cloud-based pay-per-use molecular modelling and related services to the biotech and pharmaceutical industry. It is now inactive.

AutoDockCloud (Ellingson and Baudry, 2014) is a workflow system that enables distributed screening on a cloud platform using the molecular docking program AutoDock. AutoDockCloud is based on the open source framework Apache Hadoop, which

implements the MapReduce paradigm for distributed computing. In addition to the docking procedure itself, preparation steps, such as generating docking parameters and ligand input files, can be automated and distributed by AutoDockCloud.

Acellera (see "Relevant Websites section") announced AceCloud, that is a commercial on-demand GPU cloud computing service. The optimization of MD simulation packages for running on GPUs, such as Amber, Gromacs (see "Relevant Websites section"), LAMMPS (see "Relevant Websites section"), and NAMD is reported to result in large performance increases compared to calculations run on CPUs. AceCloud is designed to free the users from the constraints of their workstation and – though the use of cloud computing technology (specifically, Amazon Web Services) – allows to run hundreds of simulations as easily as running one without the need of any additional setup. The AceCloud interface abstracts all interactions with the supporting cloud computing infrastructure. It emulates the experience of running cloud molecular dynamics locally on one's own machine. AceCloud provides transparent execution of ACEMD (Amber, Charmm inputs) and Gromacs MD simulations. It has a built-in mechanism by which the user may run arbitrary code (such as NAMD and Amber) on an AceCloud instance.

A software plugin for VMD has been developed that provides an integrated framework for VMD to be executed on Amazon EC2, facilitating MD simulations of biomolecules using the NAMD program. This plugin allows use of the VMD Graphical User Interface to: (1) Create a compute cluster on Amazon EC2; (2) submit a parallel MD simulation job using the NAMD software; (3) transfer the results to the host computer for post-processing steps; and (4) shutdown and terminate the compute cluster on Amazon EC2.

Another SaaS example for MD is NCBI Blast (see "Relevant Websites section") on Windows Azure, that is a cloud-based implementation of the Basic Local Alignment Search Tool (BLAST) of the National Center for Biotechnology Information (NCBI). e-Science Cloud is a cloud based platform for data analysis developed at Newcastle University (Hiden *et al.*, 2013).

Closing Remarks

The adoption of cloud computing model in molecular modelling could guarantee several benefits to MD researchers, thanks to computational and storage resources allocated elastically and on demand.

See also: Computational Systems Biology Applications. Identification and Extraction of Biomarker Information. Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Infrastructures for High-Performance Computing: Cloud Computing Development Environments. Infrastructures for High-Performance Computing: Cloud Infrastructures. Molecular Dynamics and Simulation. Molecular Dynamics Simulations in Drug Discovery. Protein Design. Protein Structure Visualization

References

- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25, 3389–3402.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28 (1), 235–242.
- Brooks, B.R., *et al.*, 2009. CHARMM: The biomolecular simulation program. *Journal of computational chemistry* 30, 1455–1464.
- Calabrese, B., Cannataro, M., 2015. Cloud computing in healthcare and biomedicine. *Scalable Computing: Practice and Experience* 16 (1), 1–17.
- Case, D.A., *et al.*, 2005. The amber biomolecular simulation programs. *Journal of Computational Chemistry* 26, 1668–1688.
- Chen, J., Liu, B., Huang, D., 2016. Protein remote homology detection based on an ensemble learning approach. *BioMed Research International*, 11. (Article ID 5813645).
- Ebejer, J., Fulle, S., Morris, G.M., Finn, P.W., 2013. The emerging role of cloud computing in molecular modeling. *Journal of Molecular Graphics and Modelling* 44, 177–187.
- Ellingson, S.R., Baudry, J., 2014. High-throughput virtual molecular docking with AutoDockCloud. *Concurrency and Computation: Practice and Experience* 26 (4), 907–916.
- Finn, R.D., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research* 44, D279–D285.
- Hiden, H., Woodman, S., Watson, P., Cala, J., 2013. Developing cloud applications using the e-Science central platform. *Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences* 371.
- Humphrey, W., Dalke, A., Schulten, K., 1996. VMD – Visual molecular dynamics. *Journal of Molecular Graphics* 14, 33–38.
- Kelley, L.A., Mezulis, S., Yates, C.M., Wass, M.N., Sternberg, M.J.E., 2015. The Phyre2 web portal for protein modeling, prediction and analysis. *Nature Protocols* 10, 845–858.
- Krawetz, S.A., Womble, D.D., 2003. *Introduction to Bioinformatics: A Theoretical and Practical Approach*. Humana Press.
- Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology* 247, 536–540.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry* 25, 1605–1612.
- Phillips, J.C., *et al.*, 2005. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry* 26, 1781–1802.
- Pronk, S., *et al.*, 2013. GROMACS 4.5: A high throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* 29 (7), 845–854.
- Salomon-Ferrer, R., Case, D.A., Walker, R.C., 2013. An overview of the amber biomolecular simulation package. *WIREs Computational Molecular Sciences* 3, 198–210.
- Söding, J., Biegert, A., Lupas, A.N., 2005. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Research* 1 (33), W244–W248.
- Xu, D., Zhang, Y., 2011. Improving the physical realism and structural accuracy of protein models by a two-step atomic-level energy minimization. *Biophysical Journal* 101, 2525–2534.
- Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous fragments and optimized knowledge-based force fields. *Proteins* 80, 1715–1735.
- Xu, X., Dunham, G., Zhao, X., Chiu D., Xu, J., 2015. Modeling parallel molecular simulations on amazon EC2. In: *Proceedings of International Conference on Cloud Computing and Big Data*. Shanghai, China: Institute of Electrical and Electronics Engineers Inc.
- Wong, A., Goscinski, A.M., 2012. The design and implementation of the VMD plugin for NAMD simulations on the amazon cloud. *International Journal of Cloud Computing and Services Science* 1 (4), 155–171.
- Zhang, W., Yang, J., He, B., *et al.*, 2016. Integration of QUARK and I-TASSER for Ab Initio protein structure prediction in CASP11. *Proteins* 84 (Suppl. 1), 76–86. doi:10.1002/prot.24930.

Relevant Websites

www.acellera.com

Acellera.

<https://blast.ncbi.nlm.nih.gov/Blast.cgi>

BLAST.

<https://www.charmm.org>

CHARMM.

<http://www.gromacs.org>

Gromacs.

<http://lammps.sandia.gov>

LAMMPS.

<https://toolkit.tuebingen.mpg.de/#/tools/hhpred>

MPI Bioinformatics Toolkit.

<http://www.ks.uiuc.edu/Research/namd/>

NAMD-Scalable Molecular Dynamics.

<http://www.sbg.bio.ic.ac.uk/phyre2/>

Phyre.

www.rosettacommons.org

RosettaCommons.

<http://ambermd.org>

The Amber Molecular Dynamics Package.

<https://www.cgl.ucsf.edu/chimera/>

UCSF Chimera.

<http://www.ks.uiuc.edu/Research/vmd/>

VMD-Visual Molecular Dynamics.

<https://zhanglab.cmb.med.umich.edu/QUARK/>

Zhang Lab.

The Challenge of Privacy in the Cloud

Francesco Buccafurri, Vincenzo De Angelis, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera, University of Reggio Calabria, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Sharing health data can result in many advantages for patients, doctors, researchers, and medical institutions. Experiences like that of [Wicks et al. \(2010\)](#) show that patients can share data about symptoms, treatments, and health in order to learn from the experience of others and improve their outcomes. But the field in which the existence of platforms allowing world-wide sharing of medical data has the most potential benefit, is certainly research, as sharing research data can improve public health. As stated in [Walport and Brest \(2011\)](#), this can help to achieve both immediate and long-term goals. Among the immediate consequences, there is the possibility of setting standards of data and procedure management, so that research and clinical protocols can be routinely shared and effectively reused. Among the long-term benefits, we have to consider the availability of data for extensive analysis, also exploiting correlations with many attributes and features, as location, ethnic characteristics, habits, etc.

However, the ethical aspect plays a crucial role in this process. Indeed, the privacy of individuals as well as the dignity of communities must be carefully protected, as health data represent the most sensitive information in the person's sphere.

From the technological point of view, the cloud is the only possible response to the demand of sharing health data (consider the features of scalability, multilaterality, elasticity, availability, security, etc., required in this case). However, this opens a number of issues that must be dealt with, because it is not possible to exploit only trusted cloud providers. Indeed, the main stakeholders in this sector are private, and it is very difficult for even e-government services to forgo the outsourcing paradigm.

Therefore, security should be guaranteed against the cloud provider, whose full trustworthiness cannot be assumed.

The structure of this article is the following. In Section "Related Work", we discuss the related literature. In Section "The Problem of Privacy in the Cloud", we describe the problem of privacy in the cloud by presenting the main issues arising in this context. Among these issues, Section "The Problem of Access Linkage" focuses on a specific problem that is very relevant when the cloud delivers health services. A possible solution of this problem and its security analysis is presented Section "Proposed Solution". Finally, in Section "Conclusion", we draw our conclusion.

Related Work

The use of cloud-based approaches in medical applications is not new. [Pardamean and Rumanda \(2011\)](#) integrated conducted research on the implementation of cloud computing in health care organizations, focused on an integrated patient medical record information system using a health care standard information format for data exchange. This study was motivated by the fact that health care organizations use a variety of IT applications and infrastructure which always need to be updated as a result of rapid growth in health care services. The diversification of how health care organizations maintain their operations, especially with respect to how they maintain patient medical information caused difficulty with accessing patient's data. They concluded that the use of cloud computing in the Electronic Medical Record (EMR) design, in which leveraging common international health care standards enables integration among many types of health care organizations, whether they already have an application or want to develop a new one. Moreover, they observed that this can reduce complex business processes by automating manual processes.

[Yang et al. \(2010\)](#) implementation discussed the challenge in Medical Image exchanging, storing and sharing issues of EMR and presented a system called MIFAS (Medical Image File Accessing System) to solve the challenges of exchanging, storing and sharing medical images across different hospitals issues. Through this system, they enhanced the efficiency of sharing information between patients and their caregivers.

The problem of privacy when storing electronic medical records has been highlighted by [Li et al. \(2011\)](#). According to the definition set out in the Health Insurance Portability and Accountability Act (HIPPA), the confidential section of the electronic medical record needs to be protected. Thus, a mechanism to protect the patient's privacy is needed during electronic medical record exchange and sharing. The privacy protection mechanism can be categorized into four types, namely anonymity, pseudonymity, unlinkability, and unobservability. Typically, mathematical conversions and cross-reference tables have been utilized to conceal the confidential part of the electronic medical record in order to achieve privacy protection. However, these methods are harder to use than the unlinkability and unobservability mechanisms. [Li et al. \(2011\)](#) tried to improve the unlinkability mechanism between patient identity and the electronic medical record through cloud computing. According to this approach, the electronic medical record system in a hospital can be integrated, to facilitate the exchange and sharing of electronic medical records, and to provide smaller hospitals or clinics that have fewer resources with adequate electronic medical record storage space.

Starting from the observation that cloud computing paradigm is one of the most popular health infrastructures for facilitating electronic health record (EHR) sharing and EHR integration, [Zhang and Liu \(2010\)](#) discussed important concepts related to EHR sharing and integration in healthcare clouds and analyzed the security and privacy issues arising in access and management of

EHRs. Then, they described an EHR security reference model for managing security issues in healthcare clouds, which highlights three important core components in securing an EHR cloud. They also illustrated the development of an EHR security reference model through a use-case scenario, and described the corresponding security countermeasures and state-of-art security techniques that can be applied as basic security measures.

Radwan *et al.* (2012) focused on the importance of healthcare information technology in mobile, web, desktop, and enterprise e-health applications. They observed that the interoperability and data exchange between e-health applications and Health Information Systems (HIS) resulted in increased concerns for securing sensitive data and data exchange, and emphasized the importance of standardizing a data exchange mechanism and interoperability between HISs and e-health applications across a cloud-based service. The proposed service provides a single entry point for retrieving EMR from the various HISs in which a patient record is stored, and for any e-health application seeking such information. The authors also proposed a unified secure platform that allows developers, health care providers, and organizations access to a framework that support retrieving and management of medical records and Personal Health Records (PHR) among various subscribers. The proposed service and platform have been qualitatively evaluated by a sample of physicians, patients, and hospital administrators in a real case study.

In Liöhr *et al.* (2010), several shortcomings of e-health solutions and standards are pointed out, and, in particular, the fact that they do not address client platform security, which is a crucial aspect for the overall security of e-health systems. To fill this gap, the authors of this work defined an abstract model of e-health clouds, which comprises the common entities of healthcare telematics infrastructures. Based on this model, three main problem areas for security and privacy are outlined, which are data storage and processing, management of e-health infrastructures, and usability aspects for end-users. Then, they presented a security architecture for privacy domains in e-health systems, which extends the protection of privacy-sensitive data from centrally managed secure networks to the client platforms of the end-users. For each application area, a separate privacy domain is established and enforced both centrally and locally on each platform.

Although many works deal with the security aspects of cloud computing, few papers have focused on anonymous authentication. Malina and Hajny (2013) dealt with user anonymous access to cloud services and proposed a solution that provides registered users with anonymous access to cloud services. This solution offers anonymous authentication, meaning that users' personal attributes (age, valid registration, successful payment) can be proven without revealing users' identity. Thus, users can benefit from services without any threat of their behavior being profiled. Moreover, if users break the provider's rules, their access rights are revoked. This solution, which is based on a non-bilinear group signature, is not scalable because, when the number of users increases, the size of signature becomes too high, making this solution inefficient in many application contexts.

The recent study of Khan (2016) analyzed and classified the working mechanisms of the main security issues and the possible solutions that exist in the literature, focusing on trusted cloud computing and mechanisms for regulating security compliance among cloud service providers.

Pritam and Chatterjee (2016) proposed an encryption scheme that incorporates cryptographic approaches with role-based access control and also an anonymous control scheme to address the issue of data privacy as well as user identity privacy in access control schemes. A real-time method is provided to maintain a secure communication in cloud computing, which ensures security and trust-based access to cloud. The proposed model contains algorithms to explain data protection and user authentication problems. Their analysis suggested that the purpose of this work is to decrease cloud computing security concerns such as data protection, authentication, and securing data during communication.

All the techniques described above do not consider that the cloud provider may still associate service requests with the IP address of the user, thus linking different user requests and inferring more information. To the best of our knowledge, our proposal is the first one to consider such an attack and to provide a solution. The solution we propose leverages a P2P network for the IP obfuscation.

The Problem of Privacy in the Cloud

The security problems in cloud computing can be faced by following the classical CIA paradigm that consists of a set of rules to guarantee confidentiality, integrity, and availability. Being interested in the privacy problem, we focus our attention on the property of confidentiality. Confidentiality requires that data stored in the cloud can be accessed only by authorized people. Furthermore, the identity of a user accessing the cloud and his actions should be kept secret. Currently, many providers guarantee confidentiality only with respect to third parties but, there are many situations in which it is necessary that data are maintained confidential to the providers themselves.

When it is not possible to assume the trustworthiness of the cloud provider (because it may be malicious or simply curious), we need to apply some confidentiality mechanism.

A simple solution is to encrypt data before storing them in the cloud: if the provider does not know the encryption key, then data cannot be accessed.

Another solution is based on fragmentation. The idea is that data are not sensitive singularly but become sensitive if associated with other information. For example, the attributes NAME and ILLNESS are not sensitive per se but they become sensitive when linked each other (we do not want to declare which illness a person suffers from). Fragmentation enforces that these attributes are stored in different unlinkable fragments. Fragmentation can be used in combination with encryption (Ciriani *et al.*, 2010). For example, in a *two can keep a secret* approach, data are split into two fragments stored in two different databases belonging to non-communicating providers. Only attributes that are sensitive by themselves are encrypted.

Unfortunately, this approach presents some drawbacks. For example, when a user asks for some information and performs a query involving data values, if they are encrypted, it can be difficult for the cloud provider to resolve the query. Actually, cryptographic techniques exist that allow users to perform queries directly on encrypted data (see for example [Hacıgümüş et al., 2007](#)). But, generally, their computational effort is very high. A different approach is presented in [Di Vimercati et al. \(2014\)](#), which introduces indexes on encrypted data to perform queries. There are three types of index: direct index (there is a one-to-one correspondence between data in plaintext and index value), bucket-based-index (there is a many-to-one correspondence between data in plaintext and index value), flattened index (there is a one-to-many correspondence between data in plaintext and index value). However, these techniques are effective against static observation (observation of data stored in the database) but they do not guarantee enough protection against dynamic observation (observation of the queries). In fact, if an attacker observes a long enough sequence of queries, the mapping between index value and plaintext value can be inferred.

Another challenge in a cloud environment is related to the fact that users may have different levels of access to data (some user can access only to certain resources). A possible solution is to encrypt resources by using different keys depending on the level of access. Theoretically, a user should have a different key for each resource he can access, but if the number of resources is too high, it is an impracticable solution. So, it exists the *key derivation method* in which there is a single key per user, some upper level key and public tokens. Each user is able to derive an upper level key that allows him to access resources (or to derive another upper level key) by using his key and public tokens. The disadvantage of this approach is that, if the access policy changes, all keys must be changed. Another solution is proposed in [Di Vimercati et al. \(2007\)](#) and consists in applying two levels of encryption (a first level by user and a second level by provider) and a user that would to access data should pass both the levels.

As mentioned above, confidentiality does not refer to only privacy of data but also to the users who access to data. For example, a user may ask the provider for some information but the provider may not have this information, so it has to recover, transparently to the user, the data from another provider (collaboration between providers), and for this, the first provider should not provide information on the identity of the user to second provider.

In the literature, there are some anonymous techniques, such as attribute-based techniques, that allow users to access their data without revealing their identity, by using a *credential* that ensures users have the right of access. Instead, regarding the privacy of actions that users perform in the cloud, the challenge is to guarantee *access confidentiality* (the provider should not know the specific data which a user accesses) and *patter confidentiality* (the provider should not know if a user accesses two times to the same data). Private Information Retrieval (PIR) techniques are proposed to accomplish this. These techniques consider the database as a string of bits in which a user requests the *i*-th bit in the string, but the server ignores the position of this specific bit. To save computation, [Di Vimercati et al. \(2011\)](#) proposed the introduction of *shuffle indexes* that combine encryption to guarantee data confidentiality and shuffling to guarantee access and pattern confidentiality. Shuffling consists of changing, dynamically at each access, the physical location where data are stored.

A common approach that can be used, transparently, for all techniques, is to add some fake data in order to create confusion, in such a way that an attacker cannot distinguish fake data from real data. Furthermore, the fake information can be used in a probabilistic approach to ensure the trustworthiness of provider ([Xie et al., 2007](#)). Two other features that providers have to guarantee are an adequate level of isolation between data belonging to different users (e.g., if a user stores data influenced by virus, these virus should not be transferred to other data) and deletion confirmation (if a user deletes his data, no copy of them has to remain in cloud).

In general, privacy problems can be much more frequent in virtualization contexts, if the hypervisor does not guarantee the correct allocation and unallocation of resources to different users. Cloud computing is not just about storing data but it is also used by users to communicate with each other. In this case, even if an attacker does not intercept the message, he may infer some information from the knowledge of the identity of users or from the knowledge of the frequency of service invocation.

The challenges described so far concern technical problems in the cloud, but there are also other aspects that have to be considered. One of these is the location where data are stored, which is transparent to users. In some Countries, privacy laws are less strict than others so the Governments of these Countries may access data without the authorization of users. Again, financial problems may also occur if the provider, suddenly, goes out of business and the recovery of data may be difficult. Finally, regardless of what logical techniques providers use, they have to protect the physical location where data are stored. Although these problems may not be very significant for private users, they have a key role for companies or government organization.

The Problem of Access Linkage

Among all the challenges discussed in the previous section, here we focus on the problem related to the trustworthiness of the Cloud provider. We recall that same cloud provider may deliver very different services that have been outsourced by different parties. In a critical health service, we can imagine that strong measures are adopted, also based on cryptography, to protect both the content of processed data and user identity ([Wang et al., 2012](#); [Chaum and Van Heyst, 1991](#); [Malina and Hajny, 2013](#); [Li et al., 2013](#)). But, other services can be supplied to the same user with low or no protection measures including transactions that may reveal the user identity (e.g., a ticket sale service). In this case, an *honest but curious* provider can link the different accesses and thus discover the identity of the user.

The above problem can be better understood by means a real-life example, showing the possible weaknesses deriving from the use of cloud services in sensitive contexts. We assume that a cloud service is used for e-health services (sensitive context) and for flight reservations (typically not a sensitive context). We consider a citizen that connects to a health-care institute of another city,

say C , for a service, and this service is provided by a cloud. In this case, data are sensitive so that they are exchanged in a confidential way and no private information about the citizen is given to the cloud (usually such data are stored in a cyphered way). After this first interaction, the citizen connects to a flight reservation service to book a flight from his city to C (i.e., the city of the health-care institute).

Now, the cloud service provider, which we assume to be *honest but curious*, has the possibility of merging the above cross-domain information and to infer further information, possibly private and sensitive information about the citizen. In particular, by merging the acquired data, the cloud service provider can infer with a high probability, that the citizen has some e-health problem and he will go to the health-care institute of the city C . Moreover, it could even more accurately infer on the citizen's health problem in the case in which the type of health-care institute (e.g., mental hospital, sexual health clinic, or abortion clinic) is known. Clearly, there is also the possibility that this deduction is wrong, for example, because the citizen could be an employee of the health-care provider, but this possibility is less probable and can be excluded by using further information.

It is worth noting that solutions based on anonymous authentication do not solve this problem because the cloud service provider can first link the interaction between the citizen and the health-care institution and then the flight reservation, for example, by the IP address of the citizen requests.

In the next section, we present an approach to solve the above problem by ensuring (i) anonymous authentication, (ii) unlinkability of requests, and (iii) anonymous communication with the cloud service provider.

Proposed Solution

The technique defined in this section (derived by the proposal presented in [Buccafurri et al., 2015](#)), leverages the presence of a government party providing the health service through the cloud. Indeed, this fact makes it realistic to suppose that a trusted third party can be included in the authentication process, in order to simultaneously guarantee anonymity and access unlinkability, but also provide full accountability. The last feature is obviously necessary if we consider both the responsibilities coming from the low requirements of the involved parties and the general need of security against terrorism.

Our solution involves the presence of multiple parties. With no collusion of the involved parties, it is not possible to be aware about the identity of the user accessing the cloud. Accountability is obtained, in case of need, by merging information coming from multiple parties.

Interestingly, besides anonymity of user authentication, unlinkability of user requests is also supported. To do this, we combine a multi-party cryptographic protocol with a cooperative approach based on a Peer-To-Peer network (P2P). We believe that this new way to integrate P2P and cloud computing, in which the customers of the cloud cooperate with each other to obtain privacy features and increased efficiency, is also sustainable from a business point of view, due to the reciprocal advantages obtained by users. Conversely, a solution based on Tor ([Dingledine et al., 2004](#)), such as that proposed by [Laurikainen \(2010\)](#), [Khan and Hamlen \(2012\)](#) appears unrealistic due to legal problems which the subscription to such anonymization system may cause.

The proposed solution relies on a trusted third party to obtain anonymous authentication to a cloud provider. As an additional feature, to avoid the possibility of grouping requests of the same user by linking them to the source IP addresses, we adopt an IP obfuscation strategy based on the use of a private P2P network ([Buccafurri and Lax, 2004](#)).

The entities involved in our model are:

- (i) A cloud provider, P ;
- (ii) A user accessing its services, say U ;
- (iii) A grantor, say G , playing the role of a trusted third party (for instance, a trusted government institution);
- (iv) A set of P2P nodes, called $\mathcal{N}_S$.

As a prerequisite, the grantor and cloud provider must belong to a public key infrastructure so that they can securely interact with each other.

With that said, we are ready to describe our solution for anonymous and unlinkable authentication. As a first step, when a user, U , wants to access a cloud service, he starts an interaction with the grantor, G , and provides his identity and the information about the cloud provider, P , he wants to access. The response of G will contain both a set of *tokens* in the form of bit strings each composed of a *ticket* and a *key*, and the address of a randomly selected node of the private P2P network. U will use this node as an entry point of the P2P network to perform anonymous requests to the cloud provider, P . Each request will be associated with a ticket generated from some secret information known only to G and P . In this way, P can assess the validity of the corresponding token and uses the key to cypher the communication with U for service delivery through the P2P network. Clearly, to prevent P from linking different tokens to the same user, G builds and distributes them randomly. In this way, different requests cannot be linked together and associated with the same user.

We are ready to present our protocol for token generation. It consists of the following steps: registration, identification, and service request.

Registration. In this phase, U interacts with G to exchange all required information for establishing a secure communication channel (e.g., Diffie-Hellman key exchange).

Identification. The user, U , uses the secure channel built during the registration phase to send his identity to G . Furthermore, he specifies the public key certificate corresponding to the cloud provider with which he wants to interact.

Then, G checks the identity of U and his authorizations and produces a set, $\mathcal{TK}_S$, of n tokens as response, along with the address of a P2P node, $A \in \mathcal{N}_S$. Each token contains a pair $(ticket, key)$: more formally, the set of tokens is defined as: $\mathcal{TK}_S = \{(T_i, K_i) : T_i = E^{K_P}(\tau_i || r_i) \wedge K_i = \mathcal{H}(\tau_i || r_i)\}$, with $1 \leq i \leq n$. Here, τ_i is a (long) validity time, r_i is a nonce, K_P is the public key of P , $E^K(x)$ denotes the encryption of x with the key K , $\parallel$ denotes the concatenation operator, and $\mathcal{H}$ is a cryptographic hash function. All the tickets, T_i , are signed by G , so that their authenticity and integrity is guaranteed. Concerning the number of returned tokens, n , we observe that this implicitly establishes the maximum number of requests U can perform during a session. Of course, because the size of a token is very small, we can safely set n to a conveniently large number to decrease the required number of message exchanges between U and G .

As for the entry point, A , of the P2P network, the grantor selects it randomly from a set of the last t users who interacted with the grantor itself. The size t of this set is established by taking the dynamics of the P2P network into account in such a way as to have a suitably high probability that A is present in the network when U contacts it.

Service Request. A request of services to P can be performed once the identification procedure has been carried out. To do so, as a first step, U contacts the entry point, A , to join the P2P network.

Now, at the i -th request, U follows the following steps. First, he generates a secret value, say S , and encrypts it by using a key generated through a function, f , of the i -th key, K_i . This function is pre-agreed among all parties and is used to obtain a considerable change in K_i : typically, f is a cryptographic hash function (e.g., SHA-256). The result of this operation is $c_i = E^{f(K_i)}(S)$. After this, U uses the same procedure above to encrypt the timestamp, t_i , of the current time, thus obtaining a value, $v_i = E^{f(K_i)}(t_i)$.

Now, U is ready to perform a request to P . A request message, m_i , will contain the tuple $\langle T_i, c_i, v_i \rangle$ as credentials and will be delivered to P via the P2P network to obfuscate the originating IP address. As for this aspect, it is worth nothing that there exist several approaches in the literature for addressing this issue (Xu et al., 2003; Riahla et al., 2012). However, for the sake of clarity, we sketch one of the most simplest strategy to achieve this objective. Specifically, when a node of the P2P network receives a message for P , it will decide with a certain probability to deliver this message to P directly or to forward it to another P2P node in its neighbor list. For each message received, every node memorizes the previous hop so that a backwards route is automatically built. This route will be used to deliver the response from P back to the source of the request message. Observe that, the anonymous tunnel through the P2P network created for each request does not allow the cloud provider to link together different tickets of the same user by analyzing the source IP.

As for the reply of P , it runs the following procedure. After verifying the authenticity and integrity of the ticket by means of the public key of the grantor, P uses its private key to decipher T_i and obtains the timestamp, τ'_i , and the nonce, r'_i . Now, it verifies that τ'_i is before the current time, thus checking the temporal validity of this ticket. Moreover, it verifies whether r'_i has been ever received in the past. In the case that either the ticket is expired or r'_i is not valid (already used), this request will be discarded. It is worth noting that, long term temporal validity is useful in case of authorizations with validity time that must be reflected in the credentials sent by G to U .

Then, P obtains the secret value, S , and the timestamp, t_i , by computing $K'_i = \mathcal{H}(\tau'_i || r'_i)$ and using $f(K'_i)$ as the symmetric key to decipher c_i and v_i . Once again, if $t_i + \Delta t$ is less than the current time the request is denied. Observe that, Δt is a security system parameter set to a small value whose utility will be discussed in Section "Security Analysis".

If the request is considered valid, P verifies the validity of the key (i.e., if $K'_i = K_i$). Hence, the response of the provider will be sent through an encrypted communication, obtained using S , established with the anonymous source node of the P2P network.

As a final aspect, we consider the accountability issue. Logs generated by the cloud provider store requests by including the random number of the corresponding credential. This ensures that no information on the user can be obtained with the analysis of these logs. However, log entries can be associated with users' identities by using information stored by G , therefore full accountability is achievable only by merging information owned by both P and G .

Security Analysis

In this section, we discuss the most common types of attack and show how our protocol can withstand them.

Behavior-based deanonymization attack. This kind of attack can be perpetrated if a recurrent pattern in the usage of services during a user authenticated session is found. From the analysis of service logs, an attacker can guess the identity of a user on the basis of his learnt behavior inside the cloud. This attack can be contrasted by using different tickets for each required service. Moreover, the P2P network used to send messages to the cloud service provider adopts a routing protocol that guarantees full anonymity, hence an attacker is not able to correlate requests coming from the same IP address.

Man-in-the-middle attack. In this type of attack, both the provider and the user who accesses the service believe that they are directly communicating with each other. However, the attacker is intercepting and altering or injecting messages exchanged by them during their communication. Our protocol is able to contrast this attack because the secret value, S , used to encrypt the communication channel between P and U cannot be known by the attacker. Moreover, the attacker cannot alter the secret value, S , sent to P in the service request phase because K (which ciphers S) is known by U , and calculated by P . Thus, the attacker is not able to make them believe they are communicating directly to each other (the condition necessary for the success of the attack).

Denial-of-service attack. This type of attack happens when an attacker sends a stream of false requests at the same time to overload and interrupt the service provided by P . Our protocol detects fake requests because they are not signed by the grantor. Moreover, in case a signed message is sent to P more than one time, they will be discarded except for the first one. This attack also fails if it is

combined with a man-in-the-middle attack (this is performed by blocking and collecting a great number of tickets that are then delivered in a single burst), thanks to the ticket expiration time that is suitably set to prevent this possibility.

Replay attack. This type of attack is performed when a valid data transmission (a ticket to access a service in our case) is fraudulently delayed or repeated. Indeed, if the ticket has been already used by the legal owner, it will be detected as not valid thanks to the use of a nonce, r' , in the message. In the case an attacker intercepts the ticket going from user to provider, the expiration time Δt would force the attacker to use it immediately. However, the secret value, S , is necessary to the communication with the provider because the ticket is sent encrypted by $f(K)$. In any case, all the messages between U and P are encrypted by S , hence there is no advantage to the attacker in intercepting and replaying them.

Spoofing attack. The use of a PKI infrastructure for authentication of the grantor avoids the possibility of the attacker of impersonating the grantor in order to obtain the login information of the user.

Password guessing attack. This kind of attack is performed when an attacker tries to obtain user login information in one of the following ways: (i) *off-line*, the adversary guesses a secret (among τ , r , K and S) without the participation of any other party. Although τ is a timestamp, and hence it is easy to know, r is randomly generated and difficult to guess. Also the knowledge of K is hard, because it is a digest computed by a cryptographic hash function. Finally, the secret value, S , is encrypted by $f(K)$, so it is hard to guess without any key knowledge. (ii) *on-line*, submitting possible authentication credentials until the grantor accepts them. As this attack needs the participation of the grantor, it is averted by including a delay in the reply of grantor in order to limit the number of attempts that can be made by the attacker in a certain time.

Conclusion

The Cloud has become an effective tool for collaborative working and information sharing. However, when sensitive data (especially in the field of e-health) and services are involved, many privacy issues arise, because a cloud provider can be honest but curious. In this article, we analyzed this problem from a wide perspective, then focusing on a specific issue, which is the leakage of private information related to the access of users to services delivered by the cloud. Indeed, anonymous access mechanisms do not protect against this adversary situation, when user accesses to critical services are linkable. In this article, we highlighted this problem in the specific field of e-health, showing how the identity of a user can be inferred, thus breaking the privacy of the health service. To address this problem, we propose an authentication scheme supporting anonymity of users and unlinkability of service accesses. This goal is reached by combining a multi-party cryptographic protocol with a cooperative P2P-based approach to access services in the cloud. Moreover, by the cooperation of a trusted third party granting tickets, it is possible to monitor user access, so that users performing illegal actions can be identified.

See also: Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Infrastructures for High-Performance Computing: Cloud Computing Development Environments. Infrastructures for High-Performance Computing: Cloud Infrastructures

References

- Buccafurri, F., Lax, G., 2004. TLS: A tree-based DHT lookup service for highly dynamic networks. In: Proceedings of the On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, pp. 563–580. Springer.
- Buccafurri, F., Lax, G., Nicolazzo, S., Nocera, A., 2015. Accountability-preserving anonymous delivery of cloud services. In: Proceedings of the International Conference on Trust, Privacy and Security in Digital Business (TRUSTBUS 2015), pp. 124–135. Springer.
- Chaum, D., Van, E., 1991. Heyst group signatures. In: Proceedings of the Advances in Cryptology-EUROCRYPT 1991, pp. 257–265. Springer.
- Ciriani, V., Vimercati, S.D.C. Di, Foresti, S., *et al.*, 2010. Combining fragmentation and encryption to protect privacy in data storage. ACM Transactions on Information and System Security 13, 22.
- Dingledine, R., Mathewson, N., Syverson, P., 2004. Tor: The second-generation onion router. Technical Report DTIC Document.
- Di Vimercati, S.D.C., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P., 2007. Over-encryption: Management of access control evolution on outsourced data. In: Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 123–134. VLDB endowment.
- Di Vimercati, S.D.C., Foresti, S., Samarati, P., 2014. Selective and finegrained access to data in the cloud. In: Proceedings of the Secure Cloud Computing, pp.123–148. Springer.
- Hacıgümüş, H., Hore, B., Iyer, B., Mehrotra, S., 2007. Search on encrypted data. Secure Data Management in Decentralized Systems. 383–425.
- Khan, M.A., 2016. A survey of security issues for cloud computing. Journal of Network and Computer Applications 71, 11–29.
- Khan, S.M., Hamlen, K.W., 2012. Anonymouscloud: A data ownership privacy provider framework in cloud computing. In: Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 170–176. IEEE.
- Laurikainen, R., 2010. Secure and anonymous communication in the cloud. Aalto University School of Science and Technology. Department of Computer Science and Engineering, Tech. Rep. TTK-CSE-B10.
- Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W., 2013. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Transactions on Parallel and Distributed Systems 24, 131143.
- Löhner, H., Sadeghi, A.-R., Winandy, M., 2010. Securing the e-health cloud. In: Proceedings of the 1st ACM International Health Informatics Symposium, pp. 220–229. ACM.
- Li, Z.-R., Chang, E.-C., Huang, K.-H., Lai, F., 2011. A secure electronic medical record sharing mechanism in the cloud computing platform. In: Proceedings of the 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), pp. 98–103. IEEE.
- Malina, L., Hajny, J., 2013. Efficient security solution for privacy-preserving cloud services. In: Proceedings of the 2013 IEEE 36th International Conference on Telecommunications and Signal Processing (TSP), pp. 23–27. IEEE.

- Pardamean, B., Rumanda, R.R., 2011. Integrated model of cloud-based e-medical record for health care organizations. In: Proceedings of the 10th WSEAS International Conference on E-Activities, pp. 157–162.
- Pritam, D., Chatterjee, M., 2016. Enforcing role-based access control for secure data storage in cloud using authentication and encryption techniques. *Journal of Network Communications and Emerging Technologies* 6. Available at: www.jncet.org.
- Radwan, A.S., Abdel-Hamid, A.A., Hanafy, Y., 2012. Cloud-based service for secure electronic medical record exchange. In: Proceedings of the 2012 22nd International Conference on Computer Theory and Applications (ICCTA), pp. 94–103. IEEE.
- Riahla, M.A., Tamine, K., Gaborit, P., 2012. A protocol for file sharing, anonymous and confidential, adapted to p2p networks. In: Proceedings of the 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp. 549–557. IEEE.
- Walport, M., Brest, P., 2011. Sharing research data to improve public health. *The Lancet* 377, 537–539.
- Wang, C., Wang, Q., Ren, K., Cao, N., Lou, W., 2012. Toward secure and dependable storage services in cloud computing. *IEEE Transactions on Services Computing* 5, 220–232.
- Wicks, P., Massagli, M., Frost, J., *et al.*, 2010. Sharing health data for better outcomes on PatientsLikeMe. *Journal of Medical Internet Research*. 12.
- Xie, M., Wang, H., Yin, J., Meng, X., 2007. Integrity auditing of outsourced data. In: Proceedings of the 33rd International conference on Very large Data Bases, pp. 782–793. VLDB Endowment.
- Xu, Z., Min, R., Hu, Y., 2003. Hieras: A Dht based hierarchical p2p routing algorithm. In: Proceedings of the 2003 International Conference on Parallel Processing, pp. 187–194. IEEE.
- Yang, C.-T., Chen, L.-T., Chou, W.-L., Wang, K.-C., 2010. Implementation of a medical image file accessing system on cloud computing. In: Proceedings of the 2010 IEEE 13th International Conference on Computational Science and Engineering (CSE), pp. 321–326. IEEE.
- Zhang, R., Liu, L., 2010. Security models and requirements for healthcare application clouds. In: Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 268–275. IEEE.

Artificial Intelligence and Machine Learning in Bioinformatics

Kaitao Lai, Natalie Twine, and Aidan O'Brien, CSIRO, North Ryde, NSW, Australia

Yi Guo, Western Sydney University, Penrith, NSW, Australia

Denis Bauer, CSIRO, North Ryde, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Machine Learning is defined as a computer science discipline where algorithms iteratively learn from observations to return insights from data without the need for programming explicit tests. Machine Learning peaked in Gartner's Hype Cycle for Emerging Technologies in 2017 and received substantial attention in its broader context of Artificial Intelligence (AI) with Google AlphaGo and Tesla Autopilot showcasing the advanced decision-making ability of such techniques.

The recent technological leaps in this space are due to two enabling trends. Firstly, computing is increasingly being shifted into the cloud, which enables the money that has traditionally been spent on hardware to now be invested in computing. This rent-based access results in the use of more powerful and specialized hardware. As machines learn through an iterative process of evaluating and updating internal evaluation measures, performing this on more appropriate systems removes traditional limitations and allows for more sophisticated methods to be trained. Secondly, the digital revolution has seen a dramatic increase in data collection about almost every aspect of life. As the ability of a machine to gain generalizable insights from presented examples is directly correlated with the dataset size, the recent increase has enabled the training of very sophisticated Machine Learning models.

These datasets are not only growing vertically, by capturing more events, but also horizontally by capturing more information about these events. The challenge of "big" and "wide" data is especially pronounced in the biomedical space where, for example, whole genome sequencing (WGS) technology enables researchers to interrogate all 3 billion base pairs of the human genome. With an expected 50% of the world's population likely to have been sequenced by 2025, the resulting datasets may surpass those generated in Astronomy, Twitter and YouTube combined ([Stephens *et al.*, 2015](#)). Machine Learning approaches are hence necessary to gain insights from these enormous and highly complex modern datasets. Here we will discuss applications in sequence annotation, disease gene association, and drug discovery.

Specifically, for analysing next-generation-sequencing data, Machine Learning has been applied to analyse RNA sequencing (RNA-seq) expression data, data from chromatin accessibility assays, such as DNase I hypersensitive site sequencing (DNase-seq), or chromatin immunoprecipitation followed by sequencing (ChIP-seq), and data on histone modification or transcription factor binding, to name a few. For more details, please see the excellent review by [Libbrecht and Noble \(2015\)](#).

Machine Learning approaches have been applied in life science fields well before the genomic revolution. For example, Machine Learning algorithms can learn to recognize patterns in DNA sequences ([Libbrecht and Noble, 2015](#)), such as pinpointing the locations of transcription start sites (TSSs) ([Ohler *et al.*, 2002](#)), identifying the importance of junk DNA in the genome ([Algama *et al.*, 2017](#)), and identifying untranslated regions (UTRs), introns and exons in eukaryotic chromosomes ([Picardi and Pesole, 2010](#)).

Enhancing the annotation of genomic regions can be achieved by using Machine Learning to combine datasets for functional gene annotation. Here, the input data can include the genomic sequence, gene expression profiles across various experimental conditions or phenotypes, protein-protein interaction data, synthetic lethality data, open chromatin data, and ChIP-seq data on histone modification or transcription factor binding ([Libbrecht and Noble, 2015](#)). Specifically transcription factors, as the master regulators of gene expression, have received attention and models have built for profiling their binding behaviour ([Kummerfeld and Teichmann, 2006](#)).

Moving up from two-dimensional sequence space, Machine Learning has also found application in predicting the 3D structure of proteins and RNA molecules from sequence, the design of artificial proteins or enzymes, and the automated analysis and comparison of biomacromolecules in atomic detail ([Hamelryck, 2009](#)).

Moving from descriptive applications to interpretative areas, Machine Learning has been used to gain insights into the molecular mechanisms of genetic diseases and susceptibilities. This is because of the growing awareness that complex interactions among genes and environmental factors are important in common human disease etiology. Traditional statistical methods are not well suited for identifying such interactions, especially when interactions occur between more than two genes, or when the data are high-dimensional (many attributes or independent variables). Machine Learning algorithms, including artificial neural networks (ANNs), cellular automata (CAs), random forests (RF), and multifactor dimensionality reduction (MDR), have been used for detecting and characterising susceptibility genes and gene interactions in common, complex, multifactorial human diseases ([Mckinney *et al.*, 2006](#)). However, the traditional implementations of these technologies reach their limit with modern dataset sizes, which we will discuss in the context of VariantSpark ([O'Brien *et al.*, 2018](#)) in Sections Random Forests (RF) and Feature Selection.

Machine learning techniques have been used to elucidate the functional interactions of genes by modelling and analysing gene regulatory networks ([Schlauch *et al.*, 2017](#); [Ni *et al.*, 2016](#)). This machine learning discipline has helped answer biological questions about how molecular phylogenetics and evolution represents at whole-genome level, about how to identify protein biomarkers of diseases, and for disease-gene association discovery ([Leung *et al.*, 2016](#)).

Finally, arriving at applications in drug discovery, Machine Learning has been specifically applied in proteomics, which is the large-scale analysis of proteins, the main targets of drug discovery. Proteomics research provides applications for drug discovery,

including target identification and validation, identification of efficacy and toxicity biomarkers from readily accessible biological fluids, and investigation into the mechanisms of drug action or toxicity (Blunsom, 2004). Drug discovery is a continuous process that applies a variety of tools from diverse fields. Proteomics, genomics and some cellular and organismic approaches have been developed to accelerate the process (Abu-Jamous *et al.*, 2015a).

This article summarizes two main Machine Learning approaches: supervised learning and unsupervised learning. Apart from these predictive models, which aim to provide one result per sample, we also discuss generative methods, which are aimed at training a model to generate new data with similar properties to the training data. Lastly, we devote a section to Deep Learning, as one of the most recent developments in Machine Learning. However, we first discuss training regimes and quality control, for a more detailed description of this we refer the reader to the review by Libbrecht and Noble (2015).

Dataset for Machine Learning and Performance Measure

Choosing the right dataset is important for building a robust machine learning model. Typically, the dataset is divided into training set, validation set and test set. The training set is applied to build the models; the validation set is adopted to evaluate the generalization error of the final selected model and for optimizing the parameters; the test set is used as the final prove of the model's performance and used to report the generalization ability to unseen data.

Overfitting

In the machine learning task, a learning model is trained on a set of training data, but then it is applied to make predictions on new dataset. We need to consider the general situation that if the knowledge and data we have are not sufficient to completely determine the correct classifier. We take the risk of just constructing a classifier (or parts of this classifier) that is not grounded in reality, and is simply encoding random data points in the dataset (Domingos, 2012). The objective is to maximize its predictive accuracy on the new data set, and not necessary its accuracy on the training dataset. In fact, if we try our best to identify the very best fit to the training dataset, we run the risk of fitting the noise in the data by memorizing various peculiarities of the training data rather than finding a set of general predictive rules. This problem is called "overfitting" (Dietterich, 1995). Overfitting generally happens when the gap between the training and the test error is large (Valiant, 1984).

Domingos provided one way to understand overfitting by decomposing generalization error into bias and variance (Fig. 1). Consider a random guesser that chooses a number from a set of positive numbers irrespective of any input. Such a method will have high variance and high bias. Now if we limit the guesser to only returning the same number, it will have a low variance but is

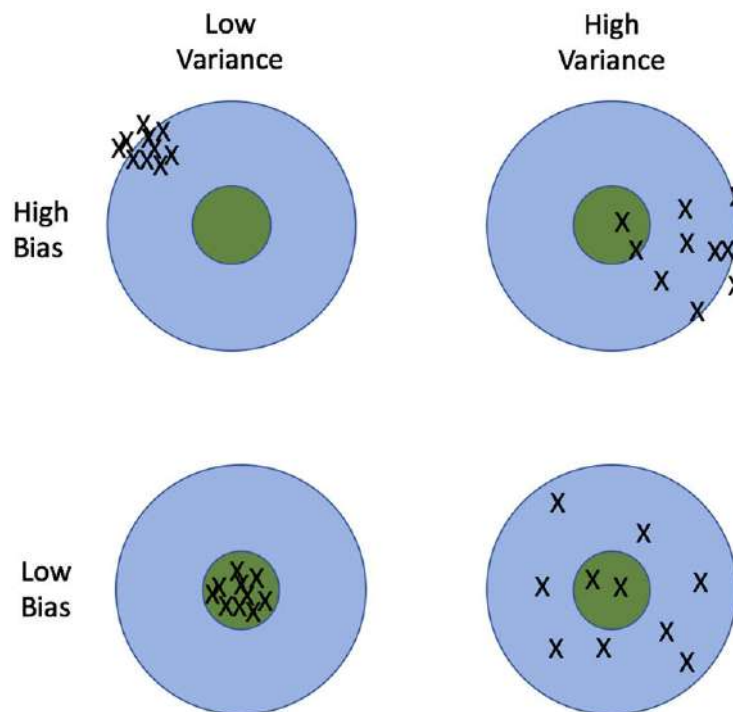


Fig. 1 Bias and variance in overfitting. Reproduced from Domingos, P., 2012. A few useful things to know about machine learning. Communications of the ACM 55, 78–87.

still biased. Conversely, we can reduce the bias by allowing the function to return positive and negative values. However, if we implement a function that actually takes the input into consideration and accurately predicts the target label, we can reduce both variance and bias.

Training Dataset

The most important consideration for generating a training dataset is to collect samples that span the entire problem space and represent predicted classes or values equally. Typically, it is easy to select samples for the commonly occurring classes or values (majority class). However, a classifier trained on such an imbalanced dataset would likely only predict the majority class. Consider a classifier for lung cancer with 351 patients, of which 95 had reoccurring cancer while 256 were cured. A model predicting no recurrence of lung cancer for all patients would reach an accuracy of 72.93% $(256/351) \times 100$. Despite this deceptively high classification accuracy it would tell 95 patients that their lung cancer was not going to reoccur (False Negatives), which is likely useless for a clinical setting. If the model needs to also accurately predict the minority class then it must have an equal representation of such samples in the dataset.

As collecting more of the minority class can be difficult, and presenting multiple copies of the same sample can lead to artefacts, undersampling the majority class is a common approach for processing imbalanced datasets. However, undersampling data has also drawbacks especially if the dataset is small to begin with. More specialized approaches have been developed taking the mechanisms in the different Machine Learning methods into account, e.g., for SVM (Akbari *et al.*, 2004).

Besides the equal representation of the prediction label (class or value), also important is the unbiased representation of samples. For example, the scenario of including the same sample multiple times can happen involuntarily especially if sample similarity cannot be determined easily (e.g., expression profile). Thorough analysis of the training dataset prior to training a Machine Learning method is hence strongly advisable.

Cross-Validation

Another potential source of bias can be the assignment of samples to training and test data. Any fixed split into training and testing data can lead to bias if the split happens to distribute samples unfavourable, e.g., all minority classes in the test set. N-fold cross-validation is a popular statistical method for randomizing the dataset and creating N equal size partitions: One from the Nth partition is picked for validation/testing, while the remaining N-1 sets are used for training the model then rotating the partitions until all have been used for testing. (Refaeilzadeh *et al.*, 2009).

However, in situations where the similarity between samples is hard to determine, standard N-fold cross validation may not be the right approach. Michiels *et al.* propose a multiple random validation strategy for prediction of cancer from microarrays. This strategy is about identifying a molecular signature (the subset of genes most differentially expressed in patients with different outcomes) in a training set of patients and to evaluate the proportion of misclassifications with this signature on an independent validation set of patients. They applied this strategy based on unique training and validation sets by using multiple random sets to study the stability of molecular signature and the proportion of misclassifications (Michiels *et al.*, 2005).

Measuring Performance of Classification

The use of the area under the receiver operating characteristic (ROC) curve (AUC) are generally used as a performance measure for machine learning algorithm on classification problem (Bradley, 1997). The receiver operating characteristic curve illustrates a binary classifier system as its discrimination threshold is varied. The ROC curve can be plotted with true positive rate (TPR) against the false positive rate (FPR) at various threshold settings from the so-called confusion matrix, which counts the correctly classified (True positives, True negative) as well as incorrect classifications (False positives, False negatives). The area under the ROC curve (AUROC) evaluates the accuracy of the test. An area of 1 represents a perfect test while an area of 0.5 demonstrates a random classifier. The precision-recall curve provides an alternative measure compensating for skewed datasets. The precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that have been retrieved over the total number of relevant instances. Both AUROC and the area under the precision-recall curve (AUPRC) can be measured using N-fold cross validation from training dataset [Table 1](#).

Two methods, correlation coefficient and root-mean-square error, are commonly applied to measure the performance of Machine Learning algorithms in regression problems. The correlation coefficient quantifies the relationship between the predicted and observed labels and ranges from -1 to $+1$. A value of 0 means no relationship exists between the two continuous variables, and a value of 1 means a perfect relationship, with -1 denoting a perfect anti-correlation. The Pearson correlation coefficient is

Table 1 Confusion matrix

	Positive	Negative
Positive	True positive	False positive
Negative	False negative	True negative

used when both variables are normally distributed, otherwise the Spearman correlation coefficient is more appropriate. The correlation coefficient represents the association, not causal relationships (Mukaka, 2012). The root-mean-square error (RMSE), on the other hand, quantifies the standard deviation of the residuals (prediction errors). The residuals are an indicator of how far from the regression line data points are. RMSE is a frequently used as indicator of the differences between values predicted by a model or an estimator and the values actually observed.

Supervised Learning

Supervised learning can be performed for training datasets where the labels or targets are known for each sample, e.g., the disease status or the category (Maglogiannis, 2007). Think of this as a child-teacher scenario, where the child learns a subject by receiving feedback on the accuracy of the answer he/she gives. Similarly, a supervised model will learn, from the truth provided in the training set, to build a generalized model that can then be applied to predict the labels for new data. The labels can be categorical, such as ethnicity (O'Brien *et al.*, 2018), or continuous, such as survival rates (Shipp *et al.*, 2002). The former case is called classification while the latter is called regression.

The inputs to the model are called features, which are information points that describe the sample, such as an expression or genotype profile. So, a more general statement is that a supervised learning approach identifies a mapping method from one data space (features) to another data space (targets, i.e., labels) (Yang, 2010).

Another subcategory of supervised learning is feature selection. The objective here is to identify the set of features that are most associated with the labels, to either gain insights into the underlying mechanisms (e.g., biological insights) or remove redundant or noisy variables in order to improve accuracy.

In the subsequent sections, we discuss the different methodologies, such as Bayesian networks, trees, random forest, support vector machines, and artificial neural networks in the classification context. However, those technologies can also be applied in regression, which we discuss in a separate section, below.

Classification

Classification is a fundamental task in data analysis and pattern recognition that constructs a classifier, which assigns a class label to an instance with a set of features/attributes. Similarly to categorization (Cohen and Lefebvre, 2005; Frey *et al.*, 2011), classification is a general process for recognizing, differentiating, understanding, and grouping ideas and objects into classes. It has been widely used in computer science, e.g., in natural language processing (Jackson and Moulinier, 2007), prediction, and decision making (Heekeren *et al.*, 2004).

The induction of classifiers from data sets of pre-classified instances, usually called training data, is one of the fundamental tasks in Machine Learning (Stanke and Waack, 2003). The process of modelling from training data, i.e., building up the mapping from observed features/attributes to correctly predict the class from the available data, is called training or learning. Some common classification approaches are Bayesian classification, classification trees, random forest, support vector machines, nearest neighbours, artificial neural networks, ensemble models (combining classifiers), and so on.

Bayesian classification

Bayesian classification is a statistical classification that minimizes the probability of misclassification (Devroye *et al.*, 2013). Many Bayesian classification algorithms are common, and they are traceable to a common ancestor (Langley *et al.*, 1992). They come originally from a supervised algorithm, which is simply referenced as a Bayesian classifier, in pattern recognition (Duda *et al.*, 1973), that assigns a simple probabilistic summary for the data; This summary includes the conditional probabilities of the class labels given the attribute values (Langley *et al.*, 1992; Rish, 2001), which is called the posterior distribution. Bayesian classifiers employ models connecting attributes to class labels and incorporate prior knowledge so that Bayesian inference can be utilized to derive the posterior distribution.

Classification trees

Tree-based algorithms form a decision tree for features in which nodes represent a series of decisions which leaves represent the final class labels. Given an observation with many features, the inference is then by passing those values of features through the decision tree from the top layer to a leave where the prediction is made. Models are constructed by recursively partitioning the feature space, and fitting a simple prediction threshold for each partition. The partitioning is represented graphically as a tree (Loh, 2011). Classification trees are very intuitive in the sense that they can be easily visualized and interpreted. They are also easy to train thanks to their simplicity. The depth of the tree and number of splits in each layer are the main parameters used to curb the overfitting problem.

Random forests (RF)

The random forests (RF) method constructs an ensemble of tree predictors, where each tree is constructed on a subset randomly selected from the training data, with the same sampling distribution for all trees in the forest (Breiman, 2001). Random forest is a

popular nonparametric tree-based ensemble Machine Learning approach that merges the ideas of adaptive nearest neighbour clustering with bagging (Breiman, 1996) for effective data adaptive inference. RF can be applied to “wide” data (“large p, small n”) problems, and accounts for correlation as well as interactions among features (Chen and Ishwaran, 2012). Data are identified as a fixed number of features which can be binary, categorical or continuous. Searching a good data demonstration is very domain specific and related to available measurements (Isabelle, 2006). In our gene editing CRISPR target sites on-target activity example, the features use the measurement of target site sequences, such as position-independent or position-dependent nucleotides and dinucleotides (Wilson, *et al.*, CRISPR Journal accepted).

RF has been used for gene selection and classification of microarray data (Korf, 2004). More recently, RF have been used for predicting CRISPR-Cas9 on-target activities (Wilson *et al.*, CRISPR Journal accepted). The RF model, called TUSCAN, is part of the GT-Scan2 suite for predicting target sites for CRISPR/Cas9 genome engineering. TUSCAN, uses sequence information describing the target site, such as global di-nucleotide frequency, or the presence of nucleotides at specific positions, to predict the activity of any given site. In total 621 features describe each site. For its final model TUSCAN performs feature selection to reduce the number of features to the 63 most important features (see Section Measuring Performance of Classification, Feature selection). TUSCAN can predict the activity of 5000 target sites in under 7 seconds, and is up to 7000 times faster than available methods, and is hence suitable for genome-wide screens.

Support vector machine (SVM)

Support vector machines (SVM) are a pattern classification technique proposed by Vapnik *et al.* (Boser *et al.*, 1992). SVM minimizes an upper bound on the generalization error through maximizing the margin between a hyperplane separating the data classes, and the data (Amari and Wu, 1999). The idea is to transform the data that is not linearly separable in its original space to a higher dimensional space where it can be separated by a simple hyperplane.

SVM has been used to identify target sequences in proteins and nucleic acids, for instance to identify SUMOylation sites (Bauer *et al.*, 2010) in proteins, or splice sites (Degroevae *et al.*, 2002) in primary mRNA. An accurate miR-E shRNA (the improved amiRNA backbone short hairpin RNAs) predictor has been developed using a sequential learning algorithm combining two support vector machine (SVM) classifiers trained on judiciously integrated data sets (Pelossof *et al.*, 2017).

A method has been developed using SVM for classification of tissue samples, consisting of ovarian cancer tissues, normal ovarian tissues and other normal tissues, and an exploration of the data for mislabelled or questionable tissue results. The tissue samples include ovarian cancer tissues, normal ovarian tissues and other normal tissues (Furey *et al.*, 2000).

Artificial neural networks

The motivation for neural networks was originally to mimic the working mechanism of the human brain. It is a graph computing model wherein computing units called neurons are organized in layers, and interconnected for passing information to each other (Kruse *et al.*, 2016). The first layer is the input layer which receives the raw data. The last layer is the output layer, which performs the final prediction task, e.g., classification, regression, and so on. The layers in between are hidden layers. To simplify optimization, the neurons in the same layer are not connected. Neural networks with three layers have the capability to approximate any function. However, the determination of the network architecture is not a trivial task, for example, the number of neurons in hidden layers and their activation functions. What about ANNs with no hidden layer?

Classic feedforward neural networks (FFNN) have been applied to predict protein site-directed recombination, which are breakpoint locations in a protein where introducing sequence from a homolog can yield improved activity (e.g., better heat stability) (Bauer *et al.*, 2006). Convolutional neural networks (CNN) uses a sequence of 2 operations, convolution and pooling, repeatedly on the input data. CNN are a subset of FFNN with a special structure, including sparse connectivity between layers and shared weights, which have surpassed conventional methods in modelling the sequence specificity of DNA-protein binding. In a systematic exploration of CNN architectures for predicting DNA sequence binding using a large compendium of transcription factor dataset, CNNs have been implemented as the best-performing architectures by varying CNN width, depth and pooling designs (Zeng *et al.*, 2016). A combination of embedding-based convolutional features (dense real value vectors, including word’s feature vector and syntax word embedding) and traditional features has been developed for use with a softmax classifier to extract DDIs from biomedical literature for detecting drug-drug interactions (DDI) (Zhao *et al.*, 2016).

Combined classification approaches

The quality of *de novo* sequence assembly can be improved by Machine Learning methods using comparative features, such as N50 score and percent match, and non-comparative features, such as mismatch percentage and the *k*-mer frequencies, to classify overlaps as true or false prior to contig (a set of overlapping DNA segments) construction (Palmer *et al.*, 2010). A comprehensive evaluation of multicategory classification methods has been performed for microarray gene expression cancer diagnosis. The multicategory support vector machines (MC-SVMs) have been demonstrated as the most effective classifiers in performing accurate cancer diagnosis from gene expression data and outperform other popular machine learning algorithms, such as backpropagation and probabilistic neural networks. The classification performance of both MC-SVMs and other non-SVM learning algorithms can be improved significantly by gene selection techniques (Statnikov *et al.*, 2005).

Microbiology is the study of microscopic organisms in numerous sub-disciplines including virology, mycology, parasitology, and bacteriology. In microbiology, it was formerly necessary to grow pure cultures in the laboratory in order to study an organism. Since many organisms cannot be cultured, this created a cultivation bottleneck that has limited our view of microbial diversity.

Metagenomics provides a relatively unbiased view of the community structure (species richness and distribution) and the functional (metabolic) potential of a community (Hugenholtz and Tyson, 2008). Metagenomic methodologies are recognized as fundamental for understanding the ecology and evolution of microbial ecosystems. The development of approaches for pathway inference from metagenomics data is crucial to connecting phenotypes to a complex set of interactions stemming from a series of combined sets of genes or proteins. The role of symbiotic microbial populations in fundamental biochemical functions have been investigated based on the modelled biochemical and regulatory pathways within one cell type, one organism, or multiple species (De Filippo *et al.*, 2012).

Machine Learning methodologies are often used for supervised classification. Feature representations and selection may improve microbe classification accuracy by producing better models and predictions (Ning and Beiko, 2015). Microbial communities are crucial to human health. Beck *et al.* have tested three Machine Learning techniques including genetic programming (GP) (Moore *et al.*, 2007; Eiben and Smith, 2003), random forests (RFs), and logistic regression (LR) for their ability to classify microbial communities into bacterial vaginosis (BV) categories. Before constructing classification models, the microbes were collapsed into groups, based on correlations, by reducing the number of factors, such as environmental variables, or dynamic microbial interactions, and to increase the interpretability of the classification models. Genetic algorithm uses computational simulations of evolutionary processes to explore highly fit models; RF is efficient but may not be as flexible as GP; LR fits a linear model, and produces a linear combination of features and regression coefficients whose value for a given set of microbial communities and patient behaviour quantifies the likelihood that the patient had BV (Beck and Foster, 2014).

Regression

In Machine Learning, regression can be seen as being more general than classification. In regression (except Poisson regression), the labels, i.e., the targets of the model, are continuous quantities, instead of discrete or categorical values. The modelling process here seeks to find a function that maps from feature to target, and which can then be used to predict the labels of new unseen data with some accuracy.

Regression methods attempt to model the relationship between input, the independent variables, and output, the dependent or response variables, by constructing parametric equations in which the parameters are estimated from the training data. The most commonly used regression methods are linear models, which include linear regression (Freedman, 2009), and regularized linear regression (McCullagh, 1984), as well as generalized linear models. Regularized regression methods fit linear models for which the number of coefficients are constrained. Regularized regression methods include ridge regression and the LASSO (Dasgupta *et al.*, 2011).

The LASSO technique was proposed by Tibshirani (1996). The LASSO and sparse least squares regression (SPLS) methods have been used for SNP selection in predicting quantitative traits. The performance in terms of r^2 (the square of Pearson correlation coefficient) for both LASSO and SPLS are almost identical in some scenarios. The LASSO produces a stable model, which is with less coefficients of each variable, because the LASSO method does not consider the effects of the correlation among SNPs, and also tends to reduce the coefficients of each variable with the shrinkage feature (shrink the feature vector) (Feng *et al.*, 2012). LASSO, along with other sparse regression models, shares the property of selecting variables and building the linear model at the same time. However, the caveat is the bias created by the regularization term, in terms of geometry, Bayesian statistics, and convex analysis, in the model.

Feature Selection

Data are demonstrated as a fixed number of features which can be binary, categorical or continuous. Identifying a good data demonstration is very domain specific and related to available measurements (Isabelle, 2006). Feature selection is crucial for the model building when there are many features in the data sampling process. Feature selection has been part of supervised learning in many real-world applications, although it can be applied to unsupervised learning scenarios as well. The high dimensional nature of many modelling tasks in bioinformatics, such as sequence analysis, microarray analysis, spectral analysis, and literature mining, has demanded the development of applications using feature selection techniques (Abu-Jamous *et al.*, 2015b). The goal is to select subset of features that can be used to better classify the given data objects (Abu-Jamous *et al.*, 2015a). A rigorous training and testing schema needs to be applied to find the statistical properties for the training set and test set, without biasing the approach by removing features that generalise well. (Libbrecht and Noble, 2015).

The three main motivations of feature selection are the following. First, to improve the overall accuracy of a prediction method by eliminating noisy or redundant features (Libbrecht and Noble, 2015). Second, to gain insights into the underlying mechanisms, e.g., answering an underlying biological question, such as identifying the genes that are associated with the corresponding functional label in order to gain insights into disease mechanisms (Glaab *et al.*, 2012; Tibshirani, 1996; Urbanowicz *et al.*, 2012). Both reasons were applied in TUSCAN (Wilson *et al.*, CRISPR Journal accepted) with an average AUC of 0.63, where the feature selection reduced the number of uninformative features from 621 to 63, and improved the average Cross-Validation result of the model by 12% to $R=0.6$ ($p<0.05$, t-test), which in turn gave insights into the most predictive features to be important to Cas9 on-target activity, including a Guanine at position 24 (G24), a depletion of Thymine within the seed region (5–12 bases preceding the PAM) and GC content.

However, there is a third motivation, which is adopted due to computational limitations rather than for improving accuracy or gaining insights. That is, to enable a more complicated model to be fitted, which has constraints on the number of features it can

process. Typical examples of this scenario include genome wide association studies (GWAS), where a preliminary feature selection based on linear regression models is performed, followed by a multi-variate model to capture complex interactions (Boyle *et al.*, 2017). The difficulty here is that this selection process can remove features that may individually not be associated with the traits of interest, but would have been major modulating factors in the multi-variate model.

To address this issue, VariantSpark (O'Brien *et al.*, 2018) was developed, which is a RF model implemented on a more powerful compute paradigm (Apache Spark), which overcomes the limitations of traditionally implemented models. VariantSpark allows a multi-variate model to be built directly on the full set of input features, thereby not discarding possibly important features during the feature selection process. It then allows features to be ranked according to their importance, thereby supporting the two main aims of feature selection, namely improving accuracy and gaining insights in the underlying mechanisms.

Unsupervised Learning

In contrast to supervised learning, learning from instances where label information is unavailable or is not used in the modelling process is called unsupervised learning (Maglogiannis, 2007). The purpose of unsupervised learning is to identify patterns in the data, such as finding groups of similar samples or identifying a trend line.

Typically, data collected through automated processes are unlabelled data, where, especially in the life-science space, the data volumes and speed with which the data is changing prevents the creation of an expert annotated subset on which to train supervised methods. As such, extracting information and gaining insights from large volumes of unlabelled data has recently become very important. Here, we will discuss Clustering approaches, such as hierarchical clustering, K-means, and model-based clustering.

Clustering

Clustering aims to group data into categories so that the data in each category share some commonalities or exhibit some uniformity. Clustering is a useful approach, especially in the exploratory data analysis phase, during decision-making, and when serving as a pre-processing step in Machine Learning. Clustering is widely used in data mining, document retrieval and image segmentation (Krogh, 2000). Hierarchical clustering, k-means clustering, and mixture models are the most important clustering families. Cluster assignment is typically quantitative in contrast to statistical dimensionality reduction methods like principal component analysis (PCA) or multidimensional scaling (MDS), where groupings are qualitatively based upon visual inspection in 2 or 3-dimensional space, or more depending on the data.

Hierarchical clustering

Hierarchical clustering builds clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. Hierarchical clustering methods can be divided into two types of clustering methods: agglomerative hierarchical clustering and divisive hierarchical clustering. In agglomerative hierarchical clustering, each object initially represents a cluster of its own, and clusters are successively merged to obtain the desired cluster structure using Ward's method. Ward recommended the criterion for selecting the pair of clusters to merge at each step is based on the optimized value of an objective function, which can be any function that reflects the investigator's purpose (Ward, 1963). In divisive hierarchical clustering, all objects initially belong to one cluster, and this cluster is successively divided into their own sub-clusters to obtain the desired cluster structure (Majoros *et al.*, 2004).

Hierarchical clustering algorithms have been used for gene sequences. A method named Unweighted Pair Group Method using arithmetic Averages (UPGMA) (average linking) is regarded the most widely used algorithm for hierarchical data clustering in computational biology. The entire collection of protein sequences has been automatically built a comprehensive evolutionary-driven hierarchy of proteins from sequence alone using a novel class of memory-constrained UPGMA (MC-UPGMA) algorithms (Loewenstein *et al.*, 2008).

k-means clustering

k-means clustering is a so-called partitional clustering algorithm, where samples for data sets are moved between clusters as the learning progresses. The algorithm begins by randomly selecting k samples and using their coordinates as the centroids (cluster centres). The algorithm then assigns each sample to its nearest centroid. This is based on a distance measure such as the Euclidian distance. Once every sample has been assigned to a centroid, each centroid is shifted to the mean of the samples assigned to that centroid. This process repeats iteratively (assign samples to centroids, adjust the centroids to reflect the mean of assigned samples) until certain stopping criteria are met. These criteria are either a maximum number of iterations, or a threshold defining a distance the centroids must move for them to not yet be considered as converged.

Many k-means implementations allow the user to define the above parameters (the value of k, the maximum number of iterations, and the distance measure). Defining the optimal value for k may be considered especially important, as it defines the number of clusters the algorithm will invariably build. Specifying the correct value for k can be a case of compromise, as increasing this value will decrease one of the error metrics (the within set sum of square error, or WSSSE), but may result in clusters

containing just one sample. One colloquial method to find the optimal value for k is the “elbow method”. This method plots the WSSSE as a function of k . According to this method, the optimal value for k is the point on the graph where the relative decrease in the WSSSE drops for each increase in k (i.e., the “elbow” in the plot).

k -means, as a heuristic algorithm, may not find the global optimum. To find the global optimum in partition-based clustering, an exhaustive enumeration process of all possible partitions is required. However, this is computationally infeasible, and therefore greedy heuristics such as k -means are ideal for these clustering problems (Majoros *et al.*, 2004).

k -means clustering was used in the VariantSpark suite to cluster individuals by their ethnicity based on their genomic profile (O’Brien *et al.*, 2015). When using genome sequencing data, this problem can be challenging as the number of variants across the input data can easily reach millions, even with only a moderate number (100–1000 s) of samples. This is due to many rare variants that are only present in a small set of samples. Because of this sparsity, variant data can be stored as sparse vectors. This is more efficient than using standard feature vectors, as sparse vectors need not store zero-values. Once VariantSpark has transformed variants from text files into sparse vectors, it can then cluster the individuals using the aforementioned K-means algorithm.

Generative Models

In this section we discuss generative models, which aim at generating novel data based on the patterns learned from the training data.

Mixture Models

Mixture models are also known as model-based clustering. Model-based clustering is a broad family of algorithms designed for modelling an unknown distribution as a mixture of simpler distributions, sometimes called basis distributions. The classification of mixture model clustering is based on the following four criteria: (i) the number of components in the mixture, including finite mixture model (parametric) and infinite mixture model (non-parametric); (ii) the clustering kernel, including multivariate normal models, or Gaussian mixture models (GMMs), the hidden Markov mixture models, Dirichlet mixture models, and other mixture models based on non-Gaussian distributions; (iii) the estimation method, including non-Bayesian methods and Bayesian methods; (iv) the dimensionality, including classes of factorising algorithms, such as mixture of factor analysers (MFA), MFA with common factor loadings, mixture of probabilistic principal component analysers, and so on (Abu-Jamous *et al.*, 2015b).

A widely used example of GMMs in bioinformatics is the VariantRecalibrator tool, which is part of the Genome Analysis Toolkit (GATK) (Depristo *et al.*, 2011). GATK is a suite of tools for genomic variant discovery in high-throughput sequencing data. VariantRecalibrator is, in fact, the first part of a two-stage process called VQSR (Variant Quality Score Recalibration). VariantRecalibrator uses GMMs to generate a continuous, co-varying estimate of the relationship between SNP annotations and the probability that a SNP is a genuine variant (versus a sequencing artefact). The GMM is estimated adaptively based on a set of true variants provided from highly validated sources (e.g., HapMap 3 sites, or the Omni 2.5M SNP chip array).

The second stage of VQSR is called ApplyRecalibration. Here, the adaptive GMM generated by VariantRecalibrator can be applied to both known and novel genetic variations discovered in the dataset at hand, thus evaluating the probability that each variant is real. Each variant is then annotated with a score called VQSLOD, which is the log-odds ratio of being a true variant versus being a false variant (sequencing error) under the trained GMM.

Probabilistic Graphical Models

Probabilistic graphical models use a graph-based representation to compactly encode a complex distribution over a high-dimensional space. The nodes correspond to the variables in the domain, and the edges correspond to direct probabilistic interactions between them (Burge and Karlin, 1997). Bayesian networks and Markov networks are important families of graphical representations of distributions.

Bayesian networks

Bayesian networks are directed acyclic graphs that efficiently represent a joint probability distribution over a set of random variables. In the graph, each vertex represents a random variable, and edges represent direct correlations between the variables. Each variable is independent of its non-descendants given the state of its parents. These independencies are then exploited to reduce the number of parameters for characterizing a probability distribution and computing posterior probabilities given the evidence (Stanke and Waack, 2003).

Bayesian networks have been applied to predict the cellular compartment to which a protein will be localized for its function (Bauer *et al.*, 2011). Here, the model integrates protein interactions (PPI), protein domains (Domains), post-translational modification sites (Motifs), and protein sequence data. For each protein, the network receives Boolean inputs of its random variables, e.g., ‘Protein-interacts-with-Pml’=False, ‘Protein-sequence-has-SUMO-site’=True, and ‘Protein-associates-with-PML-bodies’=False, which are processed in its unobserved latent variables, which represent PPI, Domains, and Motifs. The sequence information is provided by a SVM classification over the protein sequence, and presented to the network as an output variable. The

compartment variable itself is hence located between the latent variables and the output variables, and the probabilities are inferred during the training. This has the benefit that the input of the latent variables (the variables with missing data because they are unobserved or missing) is not required to be present for all variables, which makes the resulting network robust against missing information.

Markov networks

Markov networks are also called Markov random fields (MRF) or undirected graphical models, are commonly used in the statistical Machine Learning domain to succinctly model spatial correlations. A Markov random field includes a graph G ; the nodes represent random variables, and the edges define the independence semantics between the random variables. A random variable in a graph, G , is independent of its non-neighbours given the observed values for its neighbours (Krogh, 1997).

Probing cellular networks from different perspectives, using high-throughput genome-wide molecular assays, has become an important study in molecular biology. Probabilistic graphical models represent multivariate joint probability distributions through a product of terms with a few variables. These models are useful tools for extracting meaningful biological information from the resulting data sets (Friedman, 2004).

Hidden Markov Models

The hidden Markov model (HMM) is an important statistical tool for modelling data with sequential correlations in neighbouring samples, such as in time series data. Its most successful application has been in natural language processing (NLP). HMM have been applied with great success to problems such as part-of-speech tagging and noun-phrase chunking (Blunsom, 2004). In HMM, hidden variables are controlling the mechanism of how the data are generated. So, the attributes are directly affected by the hidden variables, for example, a segment of speech is dedicated to pronouncing a syllable, and this syllable can be seen as a value of a hidden variable.

HMMs have been used to resolve various problems of biological sequence analysis (Won *et al.*, 2007), including pairwise and multiple sequence alignment (Durbin *et al.*, 1998; Pachter *et al.*, 2002), base-calling (Liang *et al.*, 2007), gene prediction (Munch and Krogh, 2006), modelling DNA sequencing errors (Lottaz *et al.*, 2003), protein secondary structure prediction (Won *et al.*, 2007), fast ncRNA annotation (Weinberg and Ruzzo, 2006), ncRNA identification (Zhang *et al.*, 2006), ncRNA structural alignment (Yoon and Vaidyanathan, 2008), acceleration of RNA folding and alignment (Harmanci *et al.*, 2007), and many others (Yoon, 2009).

Pair HMMs can be used in dynamic programming (DP) for resolving alignment problems. A pair HMM emits a pairwise alignment in comparison with generalized HMMs (Durbin *et al.*, 1998). A combined approach named generalized pair HMM (GPHMM) has been developed in conjunction with approximate alignments, which allows users to state bounds on possible matches, for a reduction in memory (and computational) requirements, rendering large sequences on the order of hundreds of thousands of base pairs feasible. GPHMMs can be used for cross-species gene finding and have applications to DNA-cDNA and DNA-protein alignment (Pachter *et al.*, 2002).

HMMs have been widely applied for modelling genes. The *ab initio* HMM gene finders for eukaryotes include BRAKER1 (Hoff *et al.*, 2016), Seqping (Chan *et al.*, 2017), and MAKER-P (Campbell *et al.*, 2014). A procedure, GeneMarkS-T (Tang *et al.*, 2015), has been developed to generate a species-specific gene predictor from a set of reliable mRNA sequences and a genome. HMMs have demonstrated that species-specific gene finders are superior to gene finders trained on other species. Acyclic discrete phase-type distributions implemented using an HMM are well suited to model sequence length distributions for all gene structure blocks (Munch and Krogh, 2006). The state structure of each HMM is constructed dynamically from an array of sub-models that include only gene features from the training set. The comparison result from each individual gene predictor on each individual genome has demonstrated that species-specific gene finders are superior to gene finders trained on other species (Munch and Krogh, 2006).

A systematic approach, named EBSeq-HMM, using an HMM has been applied to modelling RNA-seq. In EBSeq-HMM, an auto-regressive HMM is developed to place dependence in gene expression across ordered conditions. This approach has been proved to be useful in identifying differentially expressed genes and in specifying gene-specific expression paths and inference regarding isoform expression (Leng *et al.*, 2015).

The prediction of the secondary structure of proteins is one of the most popular research topics in the bioinformatics community. The tasks of manual design of HMMs are challenging for the above prediction, an automated approach, using Genetic Algorithms (GA) has been developed for evolving the structure of HMMs. In the GA algorithm, the biologically meaningful building blocks of proteins (the set of 20 amino acids) are assembled as populations of HMMs. The space of Block-HMMs is discovered by mutation and crossover operators on 1662 random sequences, which are generated from the evolved HMM. The standard HMM estimation algorithm (the Baum-Welch algorithm) was applied to update model parameters after each step of the GA. This approach uses the grammar (probabilistic modelling) of protein secondary structures and transfers it into the stochastic context-free grammar of an HMM. This approach provides good performance of the probabilistic information on the prediction result under the single-sequence condition (Won *et al.*, 2007).

Non-coding RNAs (ncRNAs) are RNA molecules that are transcribed from DNA but not believed to be translated into proteins (Weinberg and Ruzzo, 2006). The ncRNA sequences play a role in the regulation of gene expression (Zhang *et al.*, 2006). The state-of-art methods, Covariance models (CMs), are an important statistical tool for identifying new members of a ncRNA gene family

in a large genome database using both sequence and RNA secondary structure information. Recent speed improvements through applying filters have been achieved (Weinberg and Ruzzo, 2006). In the development of detection methods for ncRNAs, Zhang *et al.* propose efficient filtering approaches for CMs to identify sequence segments and speed up the detection process. They built up the concept of a filter by designing efficient sequence based filters and provide figures of merit, such as G + C content, that allow comparison between filters. Zhang *et al.* (2006) also designed efficient sequence-based HMM filters to construct a new formulation of the CM that allows speeding up RNA alignment. This approach has been illustrated its efficiency and capability on both synthetic data and real bacterial genomes (Zhang *et al.*, 2006). Because many ncRNAs have secondary structures, an efficient computational method for representing RNA sequences and RNA secondary structure has been proposed for finding the structural alignment of RNAs based on profile context-sensitive hidden Markov models (profile-csHMMs) to identify ncRNA genes. The framework, based on profile-csHMMs, has been demonstrated to be effective for the computational analysis of RNAs and the identification of ncRNA genes (Yoon and Vaidyanathan, 2008).

The accuracy of structural predictions can be improved significantly by joint alignment and secondary structure prediction of two RNA sequences. Applying constraints that reduce computation by restricting the permissible alignments and/or structures further improves accuracy. A new approach has been developed for the purpose of establishing alignment constraints based on the posterior probabilities of nucleotide alignment and insertion. The posterior probabilities of alignment and insertion are computed for all possible pairings of nucleotide positions from the two sequences by a forward-backward calculation using a hidden Markov model. The co-incidence for nucleotide position pairs are obtained from these combined alignments, insertion posterior probabilities and the co-incidence probabilities are thresholded by a suitable alignment constraint, and this constraint is integrated with a free energy minimization algorithm for joint alignment and secondary structure prediction (Harmanci *et al.*, 2007).

A prediction method for a transcription factor prediction database has been implemented using profile HMMs of domains, and used for identifying sequence-specific DNA-binding transcription factors through sequence similarity. Transcription factor prediction based on HMMs of DNA-binding domains provides advantages. It is more sensitive than conventional genome annotation procedures because it uses the efficient multiple sequence comparison method of HMMs, and it recognizes only transcription factors that use the mechanism of sequence-specific DNA binding (Kummerfeld and Teichmann, 2006).

Deep Learning

Deep learning provides computational models composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These approaches have significantly improved the state-of-the-art in speech recognition, visual object recognition, and many other domains including biology, such as drug discovery and genomics. Deep learning has identified complex structures in large datasets by using the backpropagation algorithm. Backpropagation is an approach applied to estimate the error contribution of each neuron after the training data is processed. This algorithm can guide the network to change its internal parameters, to determine the representation in each layer from the representation in the previous layer (Lecun *et al.*, 2015).

As mentioned earlier, a standard neural network (NN) includes many simple, connected processors called neurons, each generating a sequence of real-valued activations. The input neurons become activated through sensors perceiving the environment, other neurons become activated through weighted connections from previously active neurons (Schmidhuber, 2015). However, simply stacking many layers of neurons together, i.e., making the model “deep”, will not increase model performance in accuracy by much. There is a need for an effective training algorithm in deep structures. In the late 1990s, there were some attempts to alleviate this problem. The convolutional neural network (CNN) (Lecun *et al.*, 1998) was proposed in 1998 in which parameter sharing and sparsity were introduced and implemented as convolutional operations in layers close to the input layer. Despite its outstanding performance, it was largely ignored. In 2006, the Restricted Boltzman Machine (RBM) was introduced (Hinton *et al.*, 2006) where neurons are treated as probabilistic units defined by energy functions (similar to a Markov Random Field). It was trained layer by layer. This provided a practical tool to make the networks really deep, i.e., containing many layers, and therefore representing very complex models with a large number of parameters to tune.

Meanwhile, computational platforms moved from vertical scaling to horizontal scaling, leading to fast growth in parallel computing and the emergence of GPUs (general computing units). These new platforms made the training of large-scale networks possible, which, in turn, stimulated the growth of deep learning. Many very deep CNNs were proposed and successfully applied to computer vision problems. Furthermore, many other neural networks designed for temporal data such as speech went deep as well, resulting in deep recurrent neural networks (DRNN) where connections between units form a directed cycle. A particular form of RNNs called long short-term memory (LSTM) was very successful in natural language processing. Now deep learning models are incorporating other learning techniques, such as reinforcement learning (Van Otterlo and Wiering, 2012), which combines learning and generative models to solve complex problems such as game playing (AlphaGo), and design (Autodesk generative design), to name a few. Reinforcement learning is a field of machine learning inspired by behaviourist psychology. Reinforcement learning is the problem that an agent resolves it by learning behaviour through trial-and-error interactions with a dynamic environment (Kaelbling *et al.*, 1996).

Deep learning requires a large amount of data for the training process. However, this can be alleviated by adopting some transfer learning techniques, i.e., localizing pre-trained models, which have been trained on readily available related data sets, and then fine-tuned on higher quality data.

Deep learning has been used in omics, biomedical imaging, and biomedical signal processing for bioinformatics (Min *et al.*, 2016). The generation of different transcripts from single genes is guided by the alternative splicing (AS) process. A model has been implemented using a deep neural network, trained on mouse RNA-Seq data, that can predict splicing patterns in individual tissues, and differences in splicing patterns across tissues. Their framework uses hidden variables jointly representing features in genomic sequences and tissue types for predictions (Leung *et al.*, 2014). In biomedical imaging research, deep learning approaches have been demonstrated to have the capability to learn physiologically important representations, such as independent component analysis (ICA) and restricted Boltzmann machine (RBM), and detect latent relations in neuroimaging data (Plis *et al.*, 2014). Buggenthin *et al.* present a deep neural network that predicts lineage choice in differentiating primary hematopoietic progenitors using millions of image patches from brightfield microscopy and cellular movement. They combine a CNN with an RNN architecture to automatically detect local image features and retrieve temporal information about the single-cell trajectories. This approach provides a solution for the identification of cells with differentially-expressed lineage-specific genes without molecular labelling (Buggenthin *et al.*, 2017). In biomedical signal processing research, brain signals have been decoded with Deep Belief Networks, probabilistic generative models that are composed of multiple layers of latent variables, and identified higher correlations with neural patterns than Principal Component Analysis (PCA).

Conclusion

Here we have provided an overview of different Machine Learning technologies and their application in the bioinformatics space. As also covered by Libbrecht and Noble, the choice of methodology depends on the properties of the available data as well as the intended outcome (Libbrecht and Noble, 2015).

We have covered supervised versus unsupervised learning, exemplified by FFNN and K-means methods respectively. The supervised FFNN approach has been used for designing novel proteins with site-directed recombination (Bauer *et al.*, 2006), where the Machine Learning models could learn from annotated examples. In contrast, the unsupervised K-means clustering in VariantSpark groups patients based on their genomic profile, which for a global population represents the different ethnicity groups (O'Brien *et al.*, 2015).

We also gave examples of choosing methodologies to achieve specific outcomes. For example we used a SVM for SUMOylation site prediction in protein sequence, as the intent was to develop the most accurate predictor (Bauer *et al.*, 2010). The trade-off for excellent performance here was the lack of insight into which biological feature contributes to the outcome. However, if gaining insights is the intent, as it was for the CRISPR target site prediction (Wilson *et al.*, CRISPR Journal accepted), then a methodology such as RF, is more advisable as it provides a feature-importance score after training.

Throughout the article we referred to "Big Data", which is particularly attractive for Machine Learning. This is because Machine Learning relies on the iterative process of learning from examples, which requires data to be kept close to the compute resources, and ideally in memory. Recent developments in the distributed computing space have enabled this in a standardized framework, namely Apache Hadoop, and for better memory management, Apache Spark.

We therefore also discussed VariantSpark, a Machine Learning framework for high-dimensional complex data, such as genomic information (O'Brien *et al.*, 2018). It offers supervised and unsupervised Machine Learning methods, and specifically its RF implementation exceeds other Spark-based parallelization attempts in the volume of data, e.g., Google's Plant implementation (Panda *et al.*, 2009). As such, it can be applied to datasets with millions of features to fit multi-variate models, e.g., to perform Genome wide association studies (GWAS) capturing the complex interaction between genomic loci.

In conclusion, the Machine Learning field is an exciting and rapidly evolving space especially in life sciences, as data volumes here will outpace those of traditional Big Data disciplines, like astronomy or retail. Hence, dramatic breakthroughs in advanced Machine Learning or Artificial Intelligence can be expected from this domain over the next years.

References

- Abu-Jamous, B., Fa, R., Nandi, A.K., 2015a. Feature Selection. Integrative Cluster Analysis in Bioinformatics. John Wiley & Sons, Ltd.
- Abu-Jamous, B., Fa, R., Nandi, A.K., 2015b. Mixture Model Clustering. Integrative Cluster Analysis in Bioinformatics. John Wiley & Sons, Ltd.
- Akbani, R., Kwek, S., Japkowicz, N., 2004. Applying support vector machines to imbalanced datasets. In: Proceedings of the Machine Learning, vol. 3201, pp. 39–50, ECML
- Algama, M., Tasker, E., Williams, C., *et al.*, 2017. Genome-wide identification of conserved intronic non-coding sequences using a Bayesian segmentation approach. BMC Genomics 18.
- Amari, S., Wu, S., 1999. Improving support vector machine classifiers by modifying kernel functions. Neural Networks 12, 783–789.
- Bauer, D.C., Boden, M., Thier, R., Gillam, E.M., 2006. STAR: Predicting recombination sites from amino acid sequence. BMC Bioinformatics 7.
- Bauer, D.C., Willadsen, K., Buske, F.A., *et al.*, 2011. Sorting the nuclear proteome. Bioinformatics 27, 17–114.
- Bauer, D.C., Buske, F.A., Bailey, T.L., Boden, M., 2010. Predicting SUMOylation sites in developmental transcription factors of *Drosophila melanogaster*. Neurocomputing 73, 2300–2307.
- Beck, D., Foster, J.A., 2014. Machine learning techniques accurately classify microbial communities by bacterial vaginosis characteristics. PLOS ONE 9.
- Blunsom, P., 2004. Hidden markov models. Lecture Notes 15, 18–19.
- Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152, ACM.
- Boyle, E.A., Li, Y.I., Pritchard, J.K., 2017. An expanded view of complex traits: From polygenic to omnigenic. Cell 169, 1177–1186.
- Bradley, A.P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159.

- Breiman, L., 1996. Bagging predictors. *Machine Learning* 24, 123–140.
- Breiman, L., 2001. Random forests. *Machine Learning* 45, 5–32.
- Buggenthin, F., Buettner, F., Hoppe, P.S., *et al.*, 2017. Prospective identification of hematopoietic lineage choice by deep learning. *Nature Methods* 14, 403–406.
- Burge, C., Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology* 268, 78–94.
- Campbell, M.S., Law, M.Y., Holt, C., *et al.*, 2014. MAKER-P: A tool kit for the rapid creation, management, and quality control of plant genome annotations. *Plant Physiology* 164, 513–524.
- Chan, K.L., Rosli, R., Tatarinova, T.V., *et al.*, 2017. Seqping: Gene prediction pipeline for plant genomes using self-training gene models and transcriptomic data. *BMC Bioinformatics* 18.
- Chen, X., Ishwaran, H., 2012. Random forests for genomic data analysis. *Genomics* 99, 323–329.
- Cohen, H., Lefebvre, C., 2005. *Handbook of Categorization in Cognitive Science*. Elsevier.
- Dasgupta, A., Sun, Y.V., Konig, I.R., Bailey-Wilson, J.E., Malley, J.D., 2011. Brief review of regression-based and machine learning methods in genetic epidemiology: The Genetic Analysis Workshop 17 experience. *Genetic Epidemiology* 35, S5–S11.
- Degroeve, S., De Baets, B., Van De Peer, Y., Rouze, P., 2002. Feature subset selection for splice site prediction. *Bioinformatics* 18, S75–S83.
- De Filippo, C., Ramazzotti, M., Fontana, P., Cavalieri, D., 2012. Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. *Briefings in Bioinformatics* 13, 696–710.
- Depristo, M.A., Banks, E., Poplin, R., *et al.*, 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics* 43, 491–498.
- Devroye, L., Györfi, L., Lugosi, G., 2013. *A Probabilistic Theory of Pattern Recognition*. Springer Science & Business Media.
- Dieterich, T., 1995. Overfitting and undercomputing in machine learning. *ACM Computing Surveys* 27, 326–327.
- Domingos, P., 2012. A few useful things to know about machine learning. *Communications of the ACM* 55, 78–87.
- Duda, R.O., Hart, P.E., Stork, D.G., 1973. *Pattern Classification*. New York: Wiley.
- Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G., 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge university press.
- Eiben, A.E., Smith, J.E., 2003. *Introduction to Evolutionary Computing*. Springer.
- Feng, Z.Z., Yang, X.J., Subedi, S., McNicholas, P.D., 2012. The LASSO and sparse least squares regression methods for SNP selection in predicting quantitative traits. *IEEE-ACM Transactions on Computational Biology and Bioinformatics* 9, 629–636.
- Freedman, D.A., 2009. *Statistical Models: Theory and Practice*. Cambridge university press.
- Frey, T., Gelhausen, M., Saake, G., 2011. Categorization of concerns: A categorical program comprehension model. In: *Proceedings of the 3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools*, pp. 73–82. ACM.
- Friedman, N., 2004. Inferring cellular networks using probabilistic graphical models. *Science* 303, 799–805.
- Furey, T.S., Cristianini, N., Duffy, N., *et al.*, 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16, 906–914.
- Glaab, E., Bacardit, J., Garibaldi, J.M., Krasnogor, N., 2012. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. *PLOS ONE* 7.
- Hamelryck, T., 2009. Probabilistic models and machine learning in structural bioinformatics. *Statistical Methods in Medical Research* 18, 505–526.
- Harmanci, A.O., Sharma, G., Mathews, D.H., 2007. Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign. *BMC Bioinformatics* 8.
- Heekeren, H.R., Marrett, S., Bandettini, P.A., Ungerleider, L.G., 2004. A general mechanism for perceptual decision-making in the human brain. *Nature* 431, 859–862.
- Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 1527–1554.
- Hoff, K.J., Lange, S., Lomsadze, A., Borodovsky, M., Stanke, M., 2016. BRAKER1: Unsupervised RNA-seq-based genome annotation with GeneMark-ET and AUGUSTUS. *Bioinformatics* 32, 767–769.
- Hugenholtz, P., Tyson, G.W., 2008. Microbiology – Metagenomics. *Nature* 455, 481–483.
- Isabelle, G., 2006. Feature extraction foundations and applications. *Pattern Recognition*.
- Jackson, P., Moulinier, I., 2007. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. John Benjamins Publishing.
- Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4, 237–285.
- Korf, I., 2004. Gene finding in novel genomes. *BMC Bioinformatics* 5.
- Krogh, A., 1997. Two methods for improving performance of an HMM and their application for gene finding. In: *Proceedings of the Ismb-97 Fifth International Conference on Intelligent Systems for Molecular Biology*, pp. 179–186.
- Krogh, A., 2000. Using database matches with HMMGene for automated gene detection in *Drosophila*. *Genome Research* 10, 523–528.
- Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., Steinbrecher, M., 2016. *Computational Intelligence: A Methodological Introduction*. Springer.
- Kummerfeld, S.K., Teichmann, S.A., 2006. DBD: A transcription factor prediction database. *Nucleic Acids Research* 34, D74–D81.
- Langley, P., Iba, W., Thompson, K., 1992. An analysis of Bayesian classifiers. In: *AAAI-92 Proceedings: Tenth National Conference on Artificial Intelligence*, pp. 223–228.
- Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444.
- Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- Leng, N., Li, Y., McIntosh, B.E., *et al.*, 2015. EBSeq-HMM: A Bayesian approach for identifying gene-expression changes in ordered RNA-seq experiments. *Bioinformatics*, 31, pp. 2614–2622.
- Leung, M.K.K., Delong, A., Alipanahi, B., Frey, B.J., 2016. Machine learning in genomic medicine: A review of computational problems and data sets. *Proceedings of the IEEE* 104, 176–197.
- Leung, M.K.K., Xiong, H.Y., Lee, L.J., Frey, B.J., 2014. Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30, 121–129.
- Liang, K.C., Wang, X.D., Anastassiou, D., 2007. Bayesian basecalling for DNA sequence analysis using hidden Markov models. *IEEE-ACM Transactions on Computational Biology and Bioinformatics* 4, 430–440.
- Libbrecht, M.W., Noble, W.S., 2015. Machine learning applications in genetics and genomics. *Nature Reviews Genetics* 16, 321–332.
- Loewenstein, Y., Portugaly, E., Fromer, M., Linial, M., 2008. Efficient algorithms for accurate hierarchical clustering of huge datasets: Tackling the entire protein space. *Bioinformatics* 24, 141–149.
- Loh, W.Y., 2011. Classification and regression trees. *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery* 1, 14–23.
- Lottaz, C., Iseli, C., Jongeneel, C.V., Bucher, P., 2003. Modeling sequencing errors by combining Hidden Markov models. *Bioinformatics* 19, li103–li112.
- Maglogiannis, I.G., 2007. *Emerging Artificial Intelligence Applications In Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Ios Press.
- Majoros, W.H., Pertea, M., Salzberg, S.L., 2004. TigrScan and GlimmerHMM: Two open source ab initio eukaryotic gene-finders. *Bioinformatics* 20, 2878–2879.
- McCullagh, P., 1984. Generalized linear-models. *European Journal of Operational Research* 16, 285–292.
- McKinney, B.A., Reif, D.M., Ritchie, M.D., Moore, J.H., 2006. Machine learning for detecting gene-gene interactions: A review. *Applied Bioinformatics* 5, 77–88.
- Michiels, S., Koscielny, S., Hill, C., 2005. Prediction of cancer outcome with microarrays: A multiple random validation strategy. *Lancet* 365, 488–492.
- Min, S., Lee, B., Yoon, S., 2016. Deep learning in bioinformatics. *Briefings in Bioinformatics*.
- Moore, J.H., Barney, N., Tsai, C.T., *et al.*, 2007. Symbolic modeling of epistasis. *Human Heredity* 63, 120–133.
- Mukaka, M.M., 2012. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal* 24, 69–71.

- Munch, K., Krogh, A., 2006. Automatic generation of gene finders for eukaryotic species. *BMC Bioinformatics* 7.
- Ning, J., Beiko, R.G., 2015. Phylogenetic approaches to microbial community classification. *Microbiome* 3.
- Ni, Y., Aghamirzaie, D., Elmarakeby, H., *et al.*, 2016. A machine learning approach to predict gene regulatory networks in seed development in arabidopsis. *Frontiers in Plant Science* 7.
- O'Brien, A.R., Saunders, N.F.W., Guo, Y., *et al.*, 2015. VariantSpark: Population scale clustering of genotype information. *BMC Genomics* 16.
- O'Brien, A., Szul, P., Dunne, R., *et al.*, 2018. Cloud-based machine learning enables whole-genome epistatic association analyses. in preparation.
- Ohler, U., Liao, G.C., Niemann, H., Rubin, G.M., 2002. Computational analysis of core promoters in the *Drosophila* genome. *Genome Biology* 3:RESEARCH0087.
- Pachter, L., Alexandersson, M., Cawley, S., 2002. Applications of generalized pair hidden Markov models to alignment and gene finding problems. *Journal of Computational Biology* 9, 389–399.
- Palmer, L.E., Dejori, M., Bolanos, R., Fasulo, D., 2010. Improving de novo sequence assembly using machine learning and comparative genomics for overlap correction. *BMC Bioinformatics* 11.
- Panda, B., Herbach, J.S., Basu, S., Bayardo, R.J., 2009. PLANET: Massively parallel learning of tree ensembles with MapReduce. *Proceedings of the VLDB Endowment* 2, 1426–1437.
- Pelosofo, R., Fairchild, L., Huang, C.H., *et al.*, 2017. Prediction of potent shRNAs with a sequential classification algorithm. *Nature Biotechnology* 35, 350–353.
- Picardi, E., Pesole, G., 2010. Computational methods for ab initio and comparative gene finding. *Methods in Molecular Biology* 609, 269–284.
- Plis, S.M., Hjelm, D.R., Salakhutdinov, R., *et al.*, 2014. Deep learning for neuroimaging: A validation study. *Frontiers in Neuroscience* 8.
- Rezaeilzadeh, P., Tang, L., Liu, H., 2009. Cross-validation. In: Liu, L., Özsu, M.T. (Eds.), *Encyclopedia of Database Systems*. Boston, MA: Springer US.
- Rish, I., 2001. An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, pp. 41–46, IBM.
- Schlauch, D., Paulson, J.N., Young, A., Glass, K., Quackenbush, J., 2017. Estimating gene regulatory networks with pandaR. *Bioinformatics* 33, 2232–2234.
- Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural Networks* 61, 85–117.
- Shipp, M.A., Ross, K.N., Tamayo, P., *et al.*, 2002. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* 8, 68–74.
- Stanke, M., Waack, S., 2003. Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics* 19, li215–li225.
- Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D., Levy, S., 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* 21, 631–643.
- Stephens, Z.D., Lee, S.Y., Faghri, F., *et al.*, 2015. Big data: Astronomical or genomics? *PLOS Biology* 13.
- Tang, S.Y.Y., Lomsadze, A., Borodovsky, M., 2015. Identification of protein coding regions in RNA transcripts. *Nucleic Acids Research* 43.
- Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B-Methodological* 58, 267–288.
- Urbanowicz, R.J., Granizo-Mackenzie, A., Moore, J.H., 2012. An analysis pipeline with statistical and visualization-guided knowledge discovery for Michigan-style learning classifier systems. *IEEE Computational Intelligence Magazine* 7, 35–45.
- Valiant, L.G., 1984. A theory of the learnable. *Communications of the ACM* 27, 1134–1142.
- Van Otterlo, M., Wiering, M., 2012. Reinforcement learning and Markov decision processes. In: Wiering, M., Van Otterlo, M. (Eds.), *Reinforcement Learning: State-of-the-Art*. Berlin, Heidelberg: Springer.
- Ward, J.H., 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58, 236.
- Weinberg, Z., Ruzzo, W.L., 2006. Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioinformatics* 22, 35–39.
- Won, K.J., Hamelryck, T., Pruegel-Bennett, A., Krogh, A., 2007. An evolutionary method for learning HMM structure: Prediction of protein secondary structure. *BMC Bioinformatics* 8.
- Wilson, L.O.W., Reti, D., O'Brien, A.R., Dunne, R.A., Bauer, D.C., 2018. High activity target-site identification using phenotypic independent CRISPR-Cas9 core functionality. *The CRISPR Journal* accepted.
- Yang, Z.R., 2010. *Machine Learning Approaches to Bioinformatics*. World scientific.
- Yoon, B.J., 2009. Hidden Markov models and their applications in biological sequence analysis. *Current Genomics* 10, 402–415.
- Yoon, B.J., Vaidyanathan, P.P., 2008. Structural alignment of RNAs using profile-csHMMs and its application to RNA homology search: Overview and new results. *IEEE Transactions on Automatic Control* 53, 10–25.
- Zeng, H.Y., Edwards, M.D., Liu, G., Gifford, D.K., 2016. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* 32, 121–127.
- Zhang, S.J., Borovok, I., Aharonowitz, Y., Sharan, R., Bafna, V., 2006. A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements. *Bioinformatics* 22, E557–E565.
- Zhao, Z.H., Yang, Z.H., Luo, L., Lin, H.F., Wang, J., 2016. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics* 32, 3444–3453.

Biographical Sketch



Dr. Kaitao Lai is a postdoctoral fellow at the Transformational Bioinformatics Team in Australian eHealth Research Centre and CSIRO Health and Biosecurity Business Unit. His expertise is in genome informatics, high throughput genomic data analysis, computational genome engineering, genome editing, as well as big data analysis and elastic cloud computing. He received his PhD in Bioinformatics for Plant Biotechnology and worked on microbial genomic and metagenomics data analysis, pan-genome analysis in previous postdoctoral fellow position. He is currently working on the development of predictors for CRISPR-Cpf1 in genome editing by training a Random Forests (RF) Machine Learning method, and facilitating the adaptation of the latest developments in computing infrastructure (e.g., Spark for big data analytics) and cloud technology (e.g., AWS Lambda) for research projects and commercial software products.



Dr. Natalie A Twine is a postdoctoral fellow at the Transformational Bioinformatics Team in Australian eHealth Research Centre and CSIRO Health and Biosecurity Business Unit. She works with big data technologies to understand the genetic basis of ALS. This is a collaborative project with Macquarie University and the international consortium, Project MinE. Dr. Twine has expertise in high throughput genomic and transcriptomic data analysis, clinical genomics, genetics and big data analysis. She obtained her PhD in Bioinformatics from University of New South Wales and has previously worked at UNSW, Kings College London and University College London. Natalie has 19 peer-reviewed publications (6 as senior author) with 835 citations and h-index of 12.



Aidan O'Brien is a joint PhD student between CSIRO and the John Curtin School of Medical Research at the Australian National University. His PhD project is aimed at developing sophisticated Machine Learning models to facilitate accurate CRISPR knock-in applications. He is working together with Australia's premier CRISPR facility to validate his models on novel datasets and enable new application areas. He graduated from the University of Queensland with a Bachelor of Biotechnology (1st class honours) in 2013. In his previous work, he developed GT-Scan and VariantSpark. Aidan has 4 journal publications (3 first author) with 61 citations (h-index 3). He received the "Best student and postdoc" award at CSIRO in 2015 and attracted \$180K in funding to date as AI.



Dr. Yi Guo received the B. Eng. (Hons.) in instrumentation from the North China University of Technology in 1998, and the M. Eng. in automatic control from Central South University in 2002. From 2005, he studied Computer Science at the University of New England, Armidale, Australia, focusing on dimensionality reduction for structured data with no vectorial representation. He received a PhD degree in 2008. From 2008 until 2016, he was with CSIRO, working as a computational statistician on various projects in spectroscopy, remote sensing and materials science. He joined the Centre for Research in Mathematics, Western Sydney University in 2016. His recent research interests include Machine Learning, computational statistics and big data.



Dr. Denis Bauer is the team leader of the transformational bioinformatics team in CSIRO's ehealth program. She has a PhD in Bioinformatics from the University of Queensland and held Post-doctoral appointments in biological Machine Learning at the Institute for Molecular Bioscience and in genetics at the Queensland Brain institute. Her expertise is in computational genome engineering and BigData compute systems. She is involved in national and international initiatives tasked to include genomic information into medical practice, funded with \$200M. She has 31 peer-reviewed publications (14 as first or senior author) with 7 in journals of IF > 8 (e.g., Nat Genet.) and H-index 12. To date she has attracted more than \$6.5 Million in funding as Chief investigator.

Artificial Intelligence

Francesco Scarcello, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Capturing the intrinsic nature of intelligence is one of the most fascinating questions in the history of humankind. Providing machines with its essence is one of the most fascinating ambitions in computer science.

The term Artificial Intelligence (or, simply, AI) was first used in 1956 by John McCarthy, one of the fathers of this field, to name a research conference. It is however intimately connected with logic, reasoning, cognitive science, and philosophy, so that in a sense it started with the Greek philosophers and has never stopped.

Alan Turing was probably the first to deal with the possibility of adding reasoning capability to machines, and to try to formalize the extent to which this may be possible. His paper “Computing Machinery and Intelligence” (Turing, 1950) begins with the following famous sentence: “I propose to consider the question: Can machines think?”. Instead of getting stuck in defining what we mean by an intelligent machine, he proposed a sort of imitation game, universally known as the Turing test, to identify such an “intelligent machine”. Roughly, the test is passed by a machine player if a human interrogator is not able to distinguish such an “artificial” player from human players, by putting questions (in natural language) to them. By 1966, the famous software “Eliza” performed well in the Turing test by just responding to the user’s sentences with either a generic reply or by repeating, in a slightly modified form, previous comments of the user. More recently, a similar (but malware) program called “CyberLover”, sought relationships on the Internet in order to steal personal data. Clearly enough, such programs are quite far from human intelligence, nevertheless the Turing test has been an important stimulus for developing machines exhibiting intelligent features.

Together with the contribution of many other researchers such as Gödel, Church, Karp, and Cook, just to name a few, Turing pointed out what can be computed and what cannot be computed by algorithmic procedures. In particular, since the 1970’s, it has been known that most forms of reasoning that are common to humans lead to problems that cannot be solved by our computers within an acceptable amount of time and with reasonable memory resources (unless some unlikely collapse occurs in complexity theory). Is it thus impossible for a machine to perform tasks where intelligence is needed? This classic question implicitly contains two further crucial questions: Which machine? Which intelligence? We know that there are different forms of (human) intelligence as well as different possible kinds of machines, such as quantum computers or biological machines, and new ones may come out in a not-so-distant future.

Anyway, even if no machine exhibits all facets of human intelligence, we now have machines that excellently perform many tasks previously possible only for (intelligent) humans, such as playing Chess or Go, and driving cars. Machines are starting to exhibit the ability to learn, which is a fundamental feature of intelligence. They have learned to recognize objects and faces, to process natural language sentences, and to perform many other difficult tasks. The availability of huge amounts of data that can be used for training such learning machines has led to an explosion of AI applications.

AI is becoming ubiquitous and it is going to support every human activity, from communication to healthcare, from business to entertaining. This paves the way for new questions about the ethics of AI and its impact on the economy, on the future of work, and on society, in general.

The rest of this contribution provides perspectives to some of the main facets of Artificial Intelligence. Each section below focuses on one facet, but recall that they are all part of the same puzzle, and their correlation is continuously increasing over the years.

Knowledge Representation, Reasoning, and Logic

Since the early days of AI, a crucial question has centered on how to move from raw data to information and knowledge. Indeed, *knowledge* is a precious resource that humans often keep implicitly somewhere in their brain. It is quite difficult to represent even the simple knowledge of a child, and consequently to perform basic human-like reasoning tasks, also known as commonsense reasoning (Van Harmelen *et al.*, 2008).

How can we formalize the semantics of data in a way that can be managed by machines? We next give a short overview.

Logic and Non-Monotonic Reasoning

The first approaches to knowledge representation were based on logic, starting from the ability of predicate calculus to state facts about the world, as well as to provide tools for reasoning about these facts. For instance, consider a fragment of a knowledge base (KB) encoding personal data about the inhabitants of some city. We may say

male(francesco). male(mario). female(alessia). female(sara).

$\forall x. \text{person}(x) \leftarrow (\text{male}(x) \vee \text{female}(x))$

The first line contains a list of facts representing extensional knowledge about the citizens. The first-order logic sentence in the second line encodes a form of intentional knowledge: the predicate *person* is represented by means of its logical connection to *male* and *female*, without explicitly listing who are all persons in the world. According to the *Closed World Assumption* (quite common in logic-based KBs), the extensional knowledge is considered complete. That is, in the above example, every instance of *male(X)* that does not occur as a fact in the KB is considered “false”.

By means of logical formulae we automatically infer new knowledge, and logic can be used for reasoning, too. It is crucial for an AI machine to perform *non-monotonic reasoning*, where, unlike classical logic, the addition of new knowledge may invalidate previous beliefs. Indeed, we are used to dealing with exceptions, to deducing, by default, conclusions that can be later invalidated, to defining theories that best explain a certain set of observations, and so on. For instance, most semantics for KBs are based on the so-called *negation-as-failure* concept, which says that we believe to be “false” every predicate that cannot be proved to be “true”. In the citizens example, we can infer that “*not person(pippo)*” is “true”, because both *male(pippo)* and *female(pippo)* cannot be proved to be true, which entails that, given the current knowledge, *person(pippo)* cannot be proved to be true, too. Further notable examples of non-monotonic frameworks in the huge AI literature include Default Reasoning, Circumscription, Abduction, Autoepistemic logics, and Belief Revision (Brewka, 1991).

In some contexts, it is useful to represent different modalities of qualifying facts and sentences. For example, for a given statement *p*, we may want to say that “*p* is possible”, or “*p* is necessary” (that is, “it is not possible that *p* does not hold”), or “*p* usually holds”, and so on. Modal logics provide a formal approach to representing this form of knowledge, and for reasoning on it. There are many modal logics, based on different modal operators and semantics. In particular, there are also temporal modal operators to represent statements such as “eventually *p* holds”, or “it has always been that *p*”, and so on (Blackburn et al., 2006).

Starting from knowledge of mathematical axioms, we can perform *Automated Theorem Proving*, see, e.g., (Gallier, 2015; Fitting, 2012) for an introduction to the subject.

Logic Programming

It turned out quite soon that logic could be used for quite general tasks, and the logic programming framework came out (Lloyd, 2012). One of the most important and widely known logic programming languages is Prolog (Sterling and Shapiro, 1994), which was initially developed for natural language processing, and later used for building expert systems, automated planning, and so on. The idea at the heart of logic programming is that problems can be solved automatically, if they are properly formalized in a logical way. Therefore, in contrast with the imperative paradigm of classical machine-oriented programming languages (think of C, Pascal, and the like), Prolog aims at a declarative approach where the user writes down a set of logical facts and rules encoding the problem at hand, and the system does the rest.

However, the semantic of Prolog is not fully declarative, so that further logic-based languages equipped with declarative semantics have been developed. See (Dantsin et al., 2001) for a study of the complexity and the expressive power of logic programming. In order to perform advanced queries to database management systems in a declarative way, the Datalog language has been proposed (Gottlob et al., 1990). Besides negation-as-failure, recently the core language has been extended with further powerful constructs, giving rise to so-called Answer-set Programming (ASP). Knowledge-base systems based on ASP have been implemented and engineered, and are currently used worldwide both for research and in commercial applications (Leone et al., 2006; Kaufmann et al., 2016).

Class-Based Representations and Description Logics

Another approach to knowledge representation in AI is based on network structures that encode individuals and their relationships. Starting with the notions of semantic networks (Lehmann, 1992) and frame systems (Fikes and Kehler, 1985), this line of research is currently best represented by Description Logics (DLs), see, e.g., (Baader, 2003). Contrasted with Datalog and its extensions, the domain of DLs is not finite, in general, and the closed-world assumption does not hold. That is, the individuals explicitly listed in the knowledge base are not necessarily all the relevant individuals in the world. DLs are useful for reasoning with ontologies and provide a powerful logical formalism for the Semantic Web. The standard Ontology Web Language (OWL), defined by the World Wide Web Consortium (W3C) Web Ontology Working Group, is, indeed, based on a DL (Horrocks, 2008). In particular, DLs (and OWL) find applications in knowledge representation in Bioinformatics, e.g., for terminology databases such as SNOMED CT, GALEN, and GO.

Rational Agents

In the age of the Internet of Things, where every device, even the smallest one, is programmable and equipped with some sort of intelligence, AI must deal not only with the intelligence of single machines but with the intelligence of many agents interacting with each other. Agents may be machines, humans, or a hybrid combination, and typically agents have contrasting preferences and goals.

Let us first consider the case of agents with a common goal, which cannot be reached by means of a classical central algorithm. Think for instance of sensor networks or other networks comprising cheap devices with limited sensing, processing, and communication capabilities. For these applications, distributed constraint satisfaction algorithms have been proposed in the literature (Faltings, 2006). Actually, in most cases, we would like to compute a best solution rather than any solution that meets the given

constraints, and we would like to get it from the agents in a distributed way. This is the case, e.g., of path-planning problems and of many economic applications involving weighted matching and scheduling problems (Yokoo, 2012).

Strategic Agents

Quite often, agents have diverging goals and preferences. *Game Theory* provides mathematical models and tools for studying both the case where the agents are selfish and play in a strategic way, and the case where agents are cooperative and form coalitions to obtain their best outcome (Von Neumann and Morgenstern, 1944; Osborne and Rubinstein, 1994).

In the former case, a classical result states that there always exists at least a vector of choices of all players (combined strategy) such that no player may increase her payoff by deviating from it (Nash, 1950). Such a vector is called Nash equilibrium, after the Nobel prize winner J.F. Nash, who defined it. However, its existence is guaranteed only holds if agents' choices are given in terms of probability distributions (called mixed strategies), rather than precise deterministic choices (called pure strategies). This paves the way for many "philosophical" interpretations of this result, as well as to connections among mixed strategies and the collective behavior of populations of living creatures in their quest for survival (Weibull, 1997).

Computing Nash equilibria is quite difficult from a computational point of view, as it is PPAD-complete for two-player games (Chen and Deng, 2006), as well as for multi-player games represented in a succinct way (Daskalakis et al., 2006). In the latter case, when pure strategies are considered, deciding the existence of a Nash equilibrium is an NP-complete problem (Gottlob et al., 2005).

In many cases, agents do not interact in a one-shot manner, and games consist of repeated turns. Then, the notions of strategies and of best responses do change, and any agent has the opportunity to modify her way of playing on the fly, depending of what the other players are doing. Moreover, often there is uncertainty about the knowledge of other agents' preferences, so that their actual moves in a repeated game provide precious information (we say the game does not have perfect knowledge).

Many algorithms for computing Nash equilibria, as well as variants of these equilibria, have been developed. We mention the library of tools called GAMBIT (McKelvey et al., 2006), and the suite for generating games and evaluating algorithms called GAMUT (Nudelman et al., 2004).

There are two-player games that end with a winner and a loser, think of chess or checkers. More generally, in a multiplayer game, agents may have opposite goals so that their possible outcomes always sum to zero. However, this is not always the case, and there are many situations where the agents can reach a best outcome by collaborating with each other. An important question is thus whether or not selfish agents looking for their personal highest profits are able to precisely choose those strategies that lead to the best (or at least a good) global outcome. A measure of the worst-case distance between any equilibrium and a desirable one is known as the *price of anarchy* (Koutsoupias and Papadimitriou, 1999).

The question is particularly interesting in repeated games, where a rational agent may learn from what happens in any turn of the game (Crandall and Goodrich, 2011). In particular, in repeated games, the necessity of cooperating with other rational agents to improve the overall performance may emerge as a prominent behavior. Recent experiments show that, from this point of view, machines seem to be better than human agents in learning to cooperate in repeated games, and they are also good at cooperating with humans (Crandall et al., 2017).

Coalitional Games

Another important field of AI and Game Theory focuses on Coalitional Games (von Neumann and Morgenstern, 1944), which were introduced as tools to reason about scenarios where players work together by forming coalitions, with the aim of obtaining a higher global worth than they would by acting in isolation. In contrast with the strategic games described above, a coalitional game is defined as a pair (N, v) where N is the set of agents and v is a function assigning a worth, $v(C)$, to any possible coalition of agents, C (while in strategic games values are assigned to combined strategies chosen by agents). Coalitional games find applications in voting systems, social choice, communication networks, and auctions, just to name a few.

A crucial problem in coalitional games is to find a way to distribute the total available wellness $v(N)$ to the agents in a way that is perceived as fair. A vector of values assigned to players is called an *imputation* if the available wellness is precisely distributed (efficiency), and if each agent gets at least what she would get by playing alone (individual rationality). Moreover, it is often required that such an imputation is stable, in that no subset of agents have an incentive to leave the group and play on their own (we say that the distribution belongs to the *Core* of the game). Many solution concepts have been defined in the literature to capture different notions of *fairness*, depending on the applications. Notable examples of solution concepts are the Shapley value and the Nucleolus, defined by the Nobel prize winners L.S. Shapley and R.J. Aumann, respectively. For any agent, i , her Shapley value is obtained as a weighted average of her marginal contributions, $v(C) - v(C \setminus \{i\})$, to every possible coalition, C , she may join. The Nucleolus minimizes the dissatisfaction of the coalitions as much as possible. Interestingly, some of these advanced notions of fair division have been known for thousands of years: it has been shown that the 2000-year old Babylonian Talmud describes a (unique) solution for the contested amount in bankruptcy problems that is precisely the nucleolus of a coalitional game that models the problem (Aumann and Maschler, 1985).

In Artificial Intelligence applications, the function v , which assigns a worth to each coalition, is usually represented in an implicit succinct way, for instance in terms of matching of weighted graphs, spanning trees, logical formalisms, and so on. Indeed,

whenever many agents are involved in the game, listing all entries of the worth function would be practically infeasible. The computational complexity of computing solutions for succinct games with non-fixed numbers of agents is intractable, in general (Deng and Papadimitriou, 1994; Greco *et al.*, 2009, 2015). However, there are many important classes of such games that can be dealt with in polynomial time.

For those games where cooperating with all participants is not necessarily the best option, agents may form a number of separate coalitions. This raises further problems about the structure of such coalitions, their stability, and their properties, in terms of solution concepts (Greenberg, 1994).

Mechanism Design

A dual aspect of dealing with autonomous rational agents is to design the rules of protocols for interacting agents, in such a way that some global goals of an external party (the designer) can be met (Nisan *et al.*, 2007). Agents are selfish and try to obtain their best outcome by playing in a strategic way, moreover their actual preferences are usually unknown. Therefore, the challenge in mechanism design is defining the rules of a game so that the desired properties are fulfilled at the end of the game, without having this precious information about the agents. It must be convenient for the agents to play as we would like them do. In particular, one of the basic properties of a mechanism is to make it convenient for agents to be *truthful*, that is, to reveal their actual preferences.

A typical application of mechanism design is in the field of *auctions*, where a given set of goods have to be allocated to selfish agents, who have hidden preferences on how much they value each good. Auctions currently occur everywhere, think for instance of the e-commerce websites, of the process of sharing computational power in a computing grid, and so on. In many cases, auctions on the Internet involve machines as bidders, rather than humans, or machines and humans are mixed.

Mechanism Design provides a general and powerful solution for dealing with such problems. For instance, consider the second-price bid-auction, where some good is to be sold and an agent wins the auction if she bids more than the other players, but she then pays the second highest-price to get the desired good. It turns out that this mechanism is truthful, that is, the strategy where every agent bids according to her actual private valuation of the good dominates all other strategies (Krishna, 2002).

Planning Agents

Consider an artificially intelligent machine consisting of a robot with a number of sensors that measure speed, position and so on. Assume that the robot has the ability to move itself, and possibly other objects in the space, and that it has some goal to achieve.

Such agents are typically implemented using the so-called triple-tower architecture, which comprises the following modules: the perception tower that receives the sensory signals; the model tower that manages knowledge and performs reasoning tasks; and the Action Tower that deals with actuators. In particular, by looking at the current status of the world, the model tower is in charge of defining a *plan*, that is, a sequence of actions allowing the robot to reach the desired goal.

Many planning algorithms have been described in AI literature to identify such action sequences. Some of them are based on logic and on the “Situation calculus”, which provides a predicate-calculus framework to represent states and actions, and to reason about the effects of actions on states. For instance, the language GOLOG, based on it, is used in robotics research (Levesque *et al.*, 1997).

A different and widely used approach to automated planning is based on the STRIPS action language, whose name comes from the very famous Stanford Research Institute Problem Solver (STRIPS) planner (Fikes and Nilsson, 1971).

Constraint Satisfaction

Most algorithms for AI problems search for solutions that range over huge search spaces and that are required to meet a large number of constraints. Sometimes these constraints are very mandatory constraints, and sometimes they just express preferences that we should try to satisfy, as much as possible (soft constraints).

Therefore, an important field of research in AI focuses on constraint satisfaction and constraint processing, and provides many algorithmic tools and theoretical results that are useful in many AI fields. In particular, algorithms for consistency enforcing and constraint propagation, combinatorial and stochastic search strategies, optimization tools for soft constraints, and other kinds of frameworks dealing with preferences (Dechter, 2003).

In general, solving a constraint satisfaction problem (CSP) is computationally intractable, to be precise NP-hard. However, a number of islands of tractability have been identified, that is classes of CSPs that can be solved in polynomial time. Such classes are mainly characterized either by restricting the languages used for defining the constraints (Jeavons *et al.*, 1997; Bulatov, 2013; Carbonnel and Cooper, 2015), or by restricting the (hyper)graph structure representing the way constraints interact with each other (Dechter and Pearl, 1989; Gottlob *et al.*, 2000, 2016; Greco and Scarcello, 2017).

For more information on the subject, see (Rossi *et al.*, 2006).

Machine Learning

The ability to learn is a fundamental feature of intelligent systems. It is becoming ubiquitous in AI applications, and it is at the heart of most of the recent impressive achievements of Artificial Intelligence, such as driving cars, playing chess, recognizing images and persons, and assisting humans in many tasks.

Machine learning started with the same flavor, as we have seen in the first section. That is, the original goal was learning knowledge encoded in some rule-based representation or in terms of decision trees, with a strong connection to cognitive science. A different line of research is based on tools from pattern recognition and statistics, whose outputs use simpler representations, based on attribute-value pairs or propositional representations (Michalski *et al.*, 2013). Another approach to learning by using logic, called Inductive Logic Programming, is based on the inference of logic programs from a given set of positive and negative facts about the subject to be learned (Nienhuys-Cheng and De Wolf, 1997; De Raedt *et al.*, 2008).

Many recent successful applications are based on neural networks, that is, networks of simple computational elements, inspired by the behavior of the neurons in the brain, and proposed since the early days of AI. Basically, the artificial neurons are arranged in layers and are connected through digital synapses. The bottom layer is in charge to processing the input signals; then, each neuron in a layer integrates, according to some (simple) function, the values received through its synapses in a new value, and propagates it to the next layer. The parameters of the functions acting at each neuron are repeatedly modified during the training process, depending on the network response to positive and negative examples of the subject to be learned.

There are now powerful tools, some are even open source (e.g., TensorFlow, originally developed by the Google Brain Team), that allow researchers and practitioners to easily develop powerful learning systems. From the uncountable successes of such technology, consider, for example, the AlphaGo software developed for playing Go by DeepMind. This is a nice example of AI software because it combines machine learning with randomized searches on trees. The deep learning part uses two neural networks, the value network and the policy network. The first training was performed using a database of 30 million moves from classical matches. Then, the algorithm continued to learn by playing against other AlphaGo machines.

A drawback of deep learning techniques lies in the so-called “black box problem”: unlike the alternative symbolic approach to learning, here the knowledge is hidden in the network parameters and we are not able to understand precisely what happens there and why. As a consequence, if something goes wrong it is not clear how to deal with the problem. There are experiments that show quite trivial cases that are misunderstood by the network, with no easy fix (Castelvecchi, 2016).

Natural Language Processing

Providing machines with the ability to communicate by written and spoken natural languages is one the first and most widely studied objectives in AI and computational linguistics. Recent advances in recognizing the human voice and in processing its features, allows us to use our voice and natural language for human-to-machine communications, such as asking questions to online and even mobile phone assistants, controlling cars or robots, and so on (Jurafsky and Martin, 2000).

In fact, natural language processing (NLP) does even more. By using real-time translation tools, we use NLP to communicate with other humans from different countries, by simply talking to a mobile phone and letting it pronounce, in a different (natural) language, what we have said. Eventually, in a similar way, we will also use NLP for communicating with different kinds of machines. In fact, large real-world organizations typically use many knowledge bases (KBs), encoded in different ways, and equipped with different querying and reasoning languages. Therefore, in practice, these KBs are not able to operate in a combined way, unless the organization invests (much) time and money in providing auxiliary tools for integrating them. NLP may be the key to leveraging such heterogeneous KBs by providing a common and highly accessible interface.

NLP is also used for sentiment analysis, that is, for detecting subjective information from documents, social networks, and so on. It is useful for identifying trends in users’ opinions that can be exploited, e.g., for marketing purposes or for politics.

NLP systems are quite complex and comprise many different tasks, whose implementation often relies on statistical and machine learning tools (Goldberg, 2016). These tasks include: *Syntactical tasks*, such as translating a natural language sentence to a representation in some formal grammar, by performing lemmatization, tagging the different parts of speech, and then parsing the resulting objects in order to identify the roles of words in the sentence and their relationships. *Semantical tasks*: moving from syntax to meaning by disambiguating words, performing entity recognition (by identifying proper names of persons and places, as well as their types), and identifying possible semantic relationships among these entities. *Discourse processing*: analyzing the structure of a discourse, determining which words refer to the same entities, and possibly producing automatic summaries. *Speech recognition*: determining a textual representation of a human speech by performing speech segmentation and recognizing allophones and phonemes in speech sounds.

See also: Artificial Intelligence and Machine Learning in Bioinformatics

References

- Aumann, R.J., Maschler, M., 1985. Game-theoretic analysis of a bankruptcy problem from the Talmud. *Journal of Economic Theory* 195–213.
- Baader, F., 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.
- Blackburn, P., van Benthem, J.F.A.K., Wolter, F., 2006. *Handbook of Modal Logic*. Elsevier.
- Brewka, G., 1991. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press.
- Bulatov, A.A., 2013. The complexity of the counting constraint satisfaction problem. *Journal of the ACM* 60 (5), 34:1–34:41.
- Carbonnel, C., Cooper, M.C., 2015. Tractability in constraint satisfaction problems: A survey. *Constraints* 21 (2), 115–144.
- Castelvecchi, D., 2016. Can we open the black box of AI. *Nature News* 538 (7623), 20.
- Chen, X., Deng, X., 2006. Settling the complexity of two-player Nash equilibrium. In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS'06*. pp. 261–272.
- Crandall, J.W., *et al.*, 2017. *Cooperating With Machines*. CoRR, abs/1703.0. Available at: <http://arxiv.org/abs/1703.06207>.
- Crandall, J.W., Goodrich, M.A., 2011. Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning. *Machine Learning* 82 (3), 281–314.
- Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A., 2001. Complexity and expressive power of logic programming. *ACM Comput. Surv.* 33 (3), 374–425.
- Daskalakis, C., Fabrikant, A., Papadimitriou, C.H., 2006. The game world is flat: The complexity of Nash equilibria in succinct games. *International Colloquium on Automata, Languages, and Programming*. 513–524.
- Dechter, R., 2003. *Constraint Processing*. San Francisco, CA: Morgan Kaufmann Publishers.
- Dechter, R., Pearl, J., 1989. Tree clustering for constraint networks. *Artificial Intelligence* 38 (3), 353–366.
- Deng, X., Papadimitriou, C.H., 1994. On the complexity of cooperative solution concepts. *Mathematics of Operations Research* 19 (2), 257–266.
- De Raedt, L., *et al.*, 2008. *Probabilistic Inductive Logic Programming*. Springer.
- Faltings, B., 2006. Distributed constraint programming. *Foundations of Artificial Intelligence* 2, 699–729.
- Fikes, R.E., Nilsson, N.J., 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2 (3), 189–208.
- Fikes, R., Kehler, T., 1985. The role of frame-based representation in reasoning. *Communications of the ACM* 28 (9), 904–920.
- Fitting, M., 2012. *First-Order Logic and Automated Theorem Proving*. Springer Science & Business Media.
- Gallier, J.H., 2015. *Logic for computer science: Foundations of automatic theorem proving*. Courier Dover Publications.
- Goldberg, Y., 2016. A primer on neural network models for natural language processing. *J. Artif. Intell. Res. (JAIR)* 57, 345–420.
- Gottlob, G., Greco, G., Leone, N., Scarcello, F., 2016. Hypertree decompositions: Questions and answers. In: *Proceedings of the 35th ACM Symposium on Principles of Database Systems, (PODS 2016)*, San Francisco, CA, USA, June 26–July 01, 2016. pp. 57–74.
- Gottlob, G., Greco, G., Scarcello, F., 2005. Pure Nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research* 24, 357–406.
- Gottlob, G., Leone, N., Scarcello, F., 2000. A comparison of structural CSP decomposition methods. *Artificial Intelligence* 124 (2), 243–282.
- Gottlob, G., Tanca, L., Ceri, S., 1990. *Logic Programming and Databases*. Verlag: Springer.
- Greco, G., Malizia, E., Palopoli, L., Scarcello, F., 2009. On the complexity of compact coalitional games. In: *Boutillier, C., (Ed.), Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*. Pasadena, CA, USA, p. 147–152.
- Greco, G., Malizia, E., Palopoli, L., Scarcello, F., 2015. The complexity of the nucleolus in compact games. *ACM Transactions on Computation Theory (TOCT)* 7 (1), 3.
- Greco, G., Scarcello, F., 2017. Greedy strategies and larger islands of tractability for conjunctive queries and constraint satisfaction problems. *Inf. Comput.* 252, 201–220. Available at: <https://doi.org/10.1016/j.ic.2016.11.004>.
- Greenberg, J., 1994. Coalition structures. In: *Aumann, R.J., Hart, S. (Eds.), Handbook of Game Theory With Economic Applications, vol.2*. Amsterdam, The Netherlands: North-Holland, pp. 1305–1337. *Handbooks in Economics*.
- Horrocks, I., 2008. Ontologies and the semantic web. *Communications of the ACM* 51 (12), 58–67.
- Jeavons, P., Cohen, D., Gyssens, M., 1997. Closure properties of constraints. *J. ACM* 44 (4), 527–548.
- Jurafsky, D., Martin, J.H., 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.
- Kaufmann, B., *et al.*, 2016. Grounding and solving in answer set programming. *AI Magazine* 37 (3), 25–32.
- Koutsoupias, E., Papadimitriou, C., 1999. Worst-case equilibria. In: *Proceedings of STACS*. pp. 404–413.
- Krishna, V., 2002. *Auction Theory*. San Diego, USA: Academic Press.
- Lehmann, F., 1992. *Semantic Networks in Artificial Intelligence*. Elsevier Science Inc.
- Leone, N., Pfeifer, G., Faber, W., *et al.*, 2006. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7 (3), 499–562.
- Levesque, H.J., *et al.*, 1997. GOLOG: A logic programming language for dynamic domains. *The Journal of Logic Programming* 31 (1–3), 59–83.
- Lloyd, J.W., 2012. *Foundations of Logic Programming*. Springer Science & Business Media.
- McKelvey, R.D., McLennan, A.M., Turocy, T.L., 2006. *Gambit: Software tools for game theory*.
- Michalski, R.S., Carbonell, J.G., Mitchell, T.M., 2013. *Machine Learning: An Artificial Intelligence Approach*. Springer Science & Business Media.
- Nash, J.F., 1950. Equilibrium points in n -person games. *Proceedings of the National Academy of Sciences of the United States of America* 36 (1), 48–49.
- Nienhuys-Cheng, S.-H., De Wolf, R., 1997. *Foundations of Inductive Logic Programming*. Springer Science & Business Media.
- Nisan, N., *et al.*, 2007. *Introduction to mechanism design (for computer scientists)*. *Algorithmic Game Theory* 9, 209–242.
- Nudelmann, E., *et al.*, 2004. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*. pp. 880–887.
- Osborne, M.J., Rubinstein, A., 1994. *A Course in Game Theory*. Cambridge, MA, USA: The MIT Press.
- Rossi, F., Beek, P. van, Walsh, T., 2006. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. New York, NY: Elsevier Science Inc.
- Sterling, L., Shapiro, E.Y., 1994. *The Art of Prolog: Advanced Programming Techniques*. MIT Press.
- Turing, A.M., 1950. Computing machinery and intelligence. *Mind* 59 (236), 433–460.
- Van Harmelen, F., Lifschitz, V., Porter, B., 2008. *Handbook of Knowledge Representation*. Elsevier.
- von Neumann, J., Morgenstern, O., 1944. *Theory of Games and Economic Behavior*, third ed. 1953 USA: Princeton University Press.
- Weibull, J.W., 1997. *Evolutionary Game Theory*. MIT press.
- Yokoo, M., 2012. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multiagent Systems*. Springer Science & Business Media.

Further Reading

- Bordeaux, L., Hamadi, Y., Kohli, P. (Eds.), 2014. *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press.
- Goodfellow, I.J., Bengio, Y., Courville, A.C., 2016. *Deep Learning*. MIT Press.
- Russell, S.J., Norvig, P., 2010. *Artificial Intelligence – A Modern Approach*. Pearson Education.
- Tegmark, M., 2017. *Life 3.0: Being Human in the Age of Artificial Intelligence*. Penguin Random House.

Biographical Sketch



Francesco Scarcello received the Ph.D. degree in Computer Science from the University of Calabria in 1997. He is a full professor of computer science (SSD ING-INF/05) at the University of Calabria, and he serves as an Associated Editor for the journal *Artificial Intelligence*, edited by Elsevier. His research interests are computational complexity, game theory, (hyper)graph theory, constraint satisfaction, logic programming, knowledge representation, non-monotonic reasoning, and database theory. He has extensively published in all these areas in leading conferences and journals. His paper "Pure Nash Equilibria: Hard and Easy Games" received the 2008 IJCAI-JAIR Best Paper Prize, awarded to an outstanding paper published in the Journal of Artificial Intelligence Research in the preceding five years. His paper "Hypertree Decompositions and Tractable Queries" received the 2009 ACM PODS Alberto O. Mendelzon Test-of-Time Award, awarded every year to a paper published in the proceedings of the ACM Symposium on Principles of Database Systems (PODS) ten years prior that had the most impact in terms of research, methodology, or transfer to practice over the intervening decade. In 2016, his paper "Hypertree Decompositions: Questions and Answers" was invited for the special *Gems of PODS* session, at the 35th ACM Symposium on Principles of Database Systems.

Introduction

Knowledge representation and reasoning (KR&R) is one of the foundations of Artificial Intelligence (AI). In particular, the key role of KR&R for AI relies on the hypothesis that symbolic reasoning coupled with explicit representations of knowledge may provide computers with the ability to algorithmically solve complex real world problems. In particular, the idea is to equip computer programs with a formal representation of problem domains (called *knowledge representation*) coupled with logical inference techniques in order to solve complex problems mimicking natural *reasoning*.

Former approaches for knowledge representation (KR) date before the advent of the computer era. However, they were mainly developed by mathematical logicians, who disregarded nonmathematical knowledge representation. Nonetheless, classical mathematical logic significantly influenced KR research. First-order (predicate) formulas form the basis of classical logic and declarative knowledge representation (McCarthy, 1958; Robinson, 1965). Transitive closure and related relevant concepts, however, revealed to be crucial in KR but needed more expressive formalisms; these have been introduced with the formalization of second-order formulas, which paved the way to modern logic. Some non “classical” approaches to knowledge representation introduce belief in their syntax (Gettier, 1963; Hintikka, 1962) or non classical semantics, like intuitionistic logic or super-intuitionistic logic of strong equivalence (Goble, 2001); these will not be considered in this article.

Several proposals exist among modern methods for knowledge representation and reasoning. In particular, Description Logics (DL) (Baader *et al.*, 2003; Franz and Sattler, 2001; Calvanese *et al.*, 2001) are a family of KR languages based on logic, where concept *descriptions* constitute the key ingredient. Descriptions are obtained from atomic concepts (unary predicates) and atomic roles (binary predicates). DLs are equipped with a formal, logic-based, semantics. They are currently the core representation languages for the Semantic Web and in particular for ontology representations.

Constraint programming (Apt, 2003; Frühwirth and Abdennadher, 2003; Hentenryck and Michel, 2005) is another powerful paradigm particularly suited for solving combinatorial search problems. It combines a wide range of techniques from artificial intelligence, operations research, algorithms and graph theory. In this case, the problem of interest must be expressed as a constraint satisfaction problem (CSP), where constraints are relations and the solution states which relations should hold among the given variables. General purpose constraint solvers can be used to get CSP, and consequently problem, solutions. However, expressing a generic problem in terms of constraints might be not obvious.

SAT solvers (Kautz and Selman, 2007) are based on propositional logic, and provide a framework for generic combinatorial reasoning. They are tailored to solve SAT formulas, but their full potential is evident when problems not directly encoded in propositional logics are expressed as boolean satisfiability, such as planning and verification problems. In fact, an increasing amount of practical applications can be solved with surprising efficiency by modern SAT solvers; as a matter of facts, they are able to solve hard problems with more than a million variables and several million constraints. Also in this case, one of the key issues is the proper reduction of the problem into a SAT formula.

Conceptual graphs (Michel and Mugnier, 2009) provide an expressive language and powerful reasoning capabilities; they provide a graph representation for logic, based on semantic networks. The main application domain of conceptual graphs is Natural Language Processing. Several extensions have been proposed for this model, such as contexts, metalanguage, plural nouns, etc.

In the context of declarative languages, one of the former languages proposed in the literature has been Prolog, developed since late 1970s. Here, knowledge is represented by clauses (specifically by definite clauses) and reasoning is carried out by the SLD resolution inference method. Solving a problem in this context, reduces to determining whether objects in the domain satisfy given properties or not. Prolog provides an easy way of representing knowledge and is endowed with efficient implementations. However, the language is not completely declarative and presents some issues related to the interpretation of the programs. As a matter of facts, it may happen that, given two slightly different variants of the same program, being completely equivalent from a declarative point of view, one of them solves the problem, whereas the other one goes into an infinite loop.

A less expressive, but fully declarative, knowledge representation language that emerged along Prolog is Datalog. In this context, the declarative language is used for a formal representation of the knowledge about the problem of interest and the result is a logic program. Queries to the program, then, define the reasoning. Results of the reasoning are eventually computed by an inference engine applied to the query and the program. It is important to point out that the combination of Datalog with an inference engine can provide answers to very hard problems in NP and beyond. As a consequence, if on the one hand Datalog is a subset of Prolog, on the other hand it significantly extends traditional query languages for relational databases. This is one of the reasons why Datalog showed up to be a particularly good candidate for practical applications.

Limitations of Datalog in terms of expressiveness have been addressed by Answer Set Programming (ASP). ASP is a declarative language for KR&R based on the Answer Set Semantics (Zbigniew and Truszczyński, 2006; Niemelä, 1999). It is influenced by former declarative programming languages, but also by database languages. Instead of providing a full general-purpose language for KR&R, it focuses on relevant language requirements for real-world applications. It includes nonmonotonic reasoning,

incomplete knowledge, hypothetical reasoning and default assumptions. ASP emerged in the logic programming context as one of the most influential tools for KR&R even in industrial applications, coupled with highly engineered ASP evaluation engines which are continuously improved and updated. As a matter of facts, thanks to the introduction of nonmonotonic reasoning, ASP can express all problems in the complexity class $(\Sigma)_2^P$ and its complement Π_2^P . The important role of ASP in the bioinformatics context has been recently evidenced in, (Erdem *et al.*, 2016) where several bioinformatics problem solutions based on ASP are reviewed.

Since the literature on KR&R is so vast, and since ASP emerged as a success story for the application of KR&R to bioinformatics, in the rest of this article we will focus on this formalism.

Background

In ASP, knowledge representation is based on rules, which are interpreted according to common sense principles. A program is, then, a collection of rules which represents a problem to be solved. This program, together with some input, may have several (but even none) solutions. Solutions to the program are then solutions to the problem. As for Datalog, solutions to the program are determined by an inference engine. In what follows, we provide an overview of the core syntax and semantics of Answer Set Programming, that is, disjunctive logic programming with strong negation and negation as failure, under the stable model semantics (Gelfond and Lifschitz, 1991).

Syntax

ASP language is based on *constants*, *predicate symbols* and *variable symbols*. A *term* is either a variable or a constant, whereas basic elements of ASP programs are *atoms*. An atom $p(t_1, \dots, t_n)$, is expressed by a *predicate* p of arity n having $t_1, \dots, t_n$ terms. A *classical literal* l can be either a positive atom p , or a negated atom $\neg p$. When using *Negation as failure* (NAF) literals can be positive literals, denoted as l , or negative literals, denoted as $\text{not } l$. Literals are said to be *ground* if no variables appear in them. An ASP program is composed of a finite set of (possibly *disjunctive*) rules, where each rule r is of the form:

$$a_1 \vee \dots \vee a_n \leftarrow b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m \quad (1)$$

Here $a_1, \dots, a_n, b_1, \dots, b_m$ are classical literals and $n \geq 0, m \geq k \geq 0$. The disjunction $a_1 \vee \dots \vee a_n$ is the *head* of r , whereas the conjunction $b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m$ is the *body* of r . A *normal rule* has precisely one head literal (i.e. $n = 1$), whereas if $n > 1$ the rule is called *disjunctive*. An *integrity constraint* is a rule without head literals (i.e. $n = 0$), whereas a *fact* is a rule with an empty body (i.e. $k = m = 0$). In order to simplify the notation, in this last case the symbol " $\leftarrow$ " is omitted. The following symbols can be useful as short hands for denoting the head $H(r) = \{a_1, \dots, a_n\}$ and the body $B(r) = B^+(r) \cup B^-(r)$ of a rule r , including the *positive body* $B^+(r) = \{b_1, \dots, b_k\}$ and the *negative body* $B^-(r) = \{b_{k+1}, \dots, b_m\}$.

Safety is an important property, which is required for all rules. A rule r is *safe* if every variable of r appears in at least one positive literal of the body.

A not-free ASP program $\mathcal{P}$ (i.e., such that it does not contain negative literals) is called *positive* (Observe that strong negation ($\neg$) may be present, instead), whereas a $\vee$ -free program $\mathcal{P}$ is called *normal logic program*.

Example 2.1: Consider the following program:

```

r1: root(a). edge(a,b). edge(a,c). edge(c,b). edge(c,d). edge(b,e). edge(d,a).
    node(a). node(b). node(c). node(d). node(e).
r2: inTree(X,Y)  $\vee$  outTree(X,Y)  $\leftarrow$  edge(X,Y), reached(X).
r3: reached(X)  $\leftarrow$  root(X)
r4: reached(Y)  $\leftarrow$  reached(X), inTree(X,Y).
r5:  $\leftarrow$  root(X), inTree(Z,X).
r6:  $\leftarrow$  inTree(X,Y), inTree(Z,Y), X  $\neq$  Z.
r7:  $\leftarrow$  node(X), not reached(X).

```

r_1 contains ground literals, which are also facts. r_2 is a disjunctive rule, whereas r_3 and r_4 are normal rules; r_4 is also recursive. Finally, rules r_5 – r_7 are constraints, where r_7 contains also a negative literal. $\square$

Semantics

Semantics of ASP is based on stable models formerly introduced in Gelfond and Lifschitz (1991).

Answer sets are defined over ground programs. Given a program $\mathcal{P}$ and the set $U_{\mathcal{P}}$ of all constants appearing in $\mathcal{P}$, let $B_{\mathcal{P}}$ indicate the set of all ground positive literals that can be obtained from predicates in $\mathcal{P}$ and constants in $U_{\mathcal{P}}$. The *ground instantiation* of a program $\mathcal{P}$ is the set $\text{Ground}(\mathcal{P})$ containing all possible ground rules that can be obtained by replacing each variable in $\mathcal{P}$ with constants in $U_{\mathcal{P}}$ for any rule r in $\mathcal{P}$.

The definition of answer sets is based on the notion of interpretations which are subsets of $B_{\mathcal{P}}$ satisfying certain properties. A positive literal l is true w.r.t. X if $l \in X$, otherwise it is false. A negative literal $\text{not } l$ is true whenever l is false. A rule r is satisfied if

$H(r) \cap X \neq \emptyset$ whenever all body literals are true w.r.t. X . An interpretation $X \subseteq B_P$ is a model if for every $r \in \text{Ground}(P)$ r is satisfied by X .

A model is an *answer set* for a positive disjunctive program $\mathcal{P}$, if it is minimal (under set inclusion) among all models for $\mathcal{P}$.

Now, in order to define answer sets for a general program $\mathcal{P}$, we need to introduce the *reduct* or *Gelfond-Lifschitz transform* of $\text{Ground}(P)$ w.r.t. a set $X \subseteq B_P$, which is the positive ground program $\text{Ground}(P)^X$, obtained from $\text{Ground}(P)$ by

- deleting all rules $r \in \text{Ground}(P)$ for which $B^-(r) \cap X \neq \emptyset$ holds;
- deleting the negative body from the remaining rules.

An *answer set* of a program $\mathcal{P}$ is a set $X \subseteq B_P$ such that X is an answer set of $\text{Ground}(P)^X$.

Example 2.2: Consider the program $\mathcal{P}$ of Example 2.1. $U_P = a, b, c, d, e$. Ground instantiations of rule r_3 are: $\text{reached}(a) \leftarrow \text{root}(a)$, $\text{reached}(a) \leftarrow \text{root}(b)$, $\text{reached}(a) \leftarrow \text{root}(c)$, ... $\text{reached}(e) \leftarrow \text{root}(c)$, $\text{reached}(e) \leftarrow \text{root}(d)$, $\text{reached}(e) \leftarrow \text{root}(e)$.

Observe that every answer set will contain the facts of $\mathcal{P}$. Constraints like r_5 prevent some interpretations to be closed and, consequently, models. As an example, every interpretation including $\text{root}(a)$ and $\text{inTree}(d, a)$ will be discarded.

$\mathcal{P}$ has two answer sets (we omit facts for simplicity):

$\{\text{inTree}(a, c), \text{inTree}(b, e), \text{inTree}(c, b), \text{inTree}(c, d), \text{outTree}(a, b), \text{outTree}(d, a), \text{reached}(a), \text{reached}(b), \text{reached}(c), \text{reached}(d), \text{reached}(e)\}$, and

$\{\text{inTree}(a, b), \text{inTree}(a, c), \text{inTree}(b, e), \text{inTree}(c, d), \text{outTree}(c, b), \text{outTree}(d, a), \text{reached}(a), \text{reached}(b), \text{reached}(c), \text{reached}(d), \text{reached}(e)\}$.

Observe that these are two *spanning trees* over the input graph defined by *edge* and *node* facts. $\square$

A Logic-Based Methodology to Knowledge Representation and Reasoning

ASP has been applied for solving a number of complex problems in several areas including databases and artificial intelligence. One of the advantages of ASP, as a knowledge representation and reasoning paradigm is the possibility to model problems and complex tasks in a declarative fashion.

Deductive database queries and complex reasoning tasks can be modeled by using a uniform solution over varying instances in the form of a fixed non-ground program coupled with input instances modeled by facts. More in detail, many reasoning tasks of comparatively high computational complexity can be solved in a canonical manner by following a knowledge representation methodology, called “Guess&Check” (Eiter et al., 2000; Leone et al., 2006).

In simple terms this methodology can be summarized as follows: a set of disjunctive rules (called the “guessing part”) is used to define candidate solutions; and another set of rules (called the “checking part”) imposes admissibility constraint to single out the solutions of the problem. Finally, a database of facts is used to specify an instance of the problem.

Basically, the answer sets of the combination of the input database with the guessing part, model “solution candidates”; the checking part filters candidates to guarantee that the answer sets of the resulting program represent precisely the admissible solutions for the input instance.

In what follows, we illustrate the usage of ASP for knowledge representation and reasoning by using two classical examples. In particular, we first show how to solve a classical problem in deductive database applications; then we adopt the “Guess&Check” methodology to solve well-known hard problems.

Reachability. A classical problem solved elegantly by deductive databases is called *reachability*. In more formal terms, given a directed graph $G = (V, E)$, the problem is to compute all pairs of nodes $(x, y) \in V \times V$ such that y can be reached from x by following a nonempty sequence of edges in E .

The input graph G is represented by the binary predicate $\text{arc}(X, Y)$, where a fact $\text{arc}(x, y)$ is in the input whenever $(x, y) \in E$ (i.e., G contains an arc from x to y). There is no need to represent explicitly V , since the nodes appearing in the transitive closure are implicitly given by these facts.

The solutions are modeled as follows:

$r_1: \text{reachable}(X, Y) \leftarrow \text{arc}(X, Y).$

$r_2: \text{reachable}(X, Y) \leftarrow \text{arc}(X, U), \text{reachable}(U, Y).$

where predicate $\text{reachable}(X, Y)$ will contain facts of the form $\text{reachable}(a, b)$ whenever b is reachable from a in G . Indeed, The first rule states that node Y is reachable from node X if there is an arc in the graph from X to Y , while the second rule states that node Y is reachable from node X if there exists a node U such that there is an arc from X to U and Y is reachable from U .

As an example, consider a graph represented by the following facts:

$\text{arc}(a, b), \text{arc}(b, c), \text{arc}(c, d).$

The program admits the single answer set $\{\text{reachable}(a, b), \text{reachable}(b, c), \text{reachable}(c, d), \text{reachable}(a, c), \text{reachable}(b, d), \text{reachable}(a, d), \text{arc}(a, b), \text{arc}(b, c), \text{arc}(c, d)\}$. The first three atoms are obtained by applying r_1 , while the other instances of reachable are obtained by applying r_2 .

N-Colorability. A classical NP-complete problem in graph theory is N-Colorability. Given a set of colors C and a finite directed graph $G = (V, E)$, does there exist a way of assigning precisely one color out of those in C to each node in V in such a way that nodes connected by an arc have assigned different colors?

The representation of the graph is analogous to the previous problem, and we add in the input a fact $color(c)$ for each color $c \in C$.

The following program models the *N-Colorability* problem:

```

 $r_1$ :  $node(X) \leftarrow arc(X, Y)$ .
 $r_2$ :  $node(Y) \leftarrow arc(X, Y)$ .
 $r_3$ :  $color(X, C) \vee differentColor(X, C) \leftarrow node(X), color(C)$ .
 $r_4$ :  $\leftarrow color(X, C1), color(X, C2), C1 < > C2$ .
 $r_5$ :  $\leftarrow color(X, C), color(Y, C), arc(X, Y)$ .

```

The first two rules compute the set of nodes out of the input that specifies all the arcs. Rule r_3 guesses an assignment of colors to nodes, whereas the remaining constraints ensure that only one color is assigned to each (rule r_4) and nodes connected by an arc have assigned different colors (rule r_5).

Hamiltonian Path. Given a finite directed graph $G = (V, E)$ and a node $a \in V$ of this graph, does there exist a path in G starting at a and passing through each node in V exactly once?

This is a classical NP-complete problem in graph theory. The representation of the graph is analogous to the previous problem, and we add in the input a fact modeling the starting node a as in instance of predicate *start*.

The following program models the *Hamiltonian Path* problem:

```

 $r_1$ :  $inPath(X, Y) \vee outPath(X, Y) \leftarrow arc(X, Y)$ .
 $r_2$ :  $reached(X) \leftarrow start(X)$ .
 $r_3$ :  $reached(X) \leftarrow reached(Y), inPath(Y, X)$ .
 $r_4$ :  $\leftarrow inPath(X, Y), inPath(X, Y1), Y < > Y1$ .
 $r_5$ :  $\leftarrow inPath(X, Y), inPath(X1, Y), X < > X1$ .
 $r_6$ :  $\leftarrow node(X), not\ reached(X), not\ start(X)$ .

```

The disjunctive rule (r_1) guesses a subset S of the arcs to be in the path, while the rest of the program checks whether S constitutes a Hamiltonian Path. The auxiliary predicate *reached* specifies the set of nodes which are reached from the starting node. This is very similar to the solution of reachability, but here *inPath* is made transitive using rule r_3 .

The requirements of an Hamiltonian Path are imposed in the checking part. In particular, r_4 and r_5 select solutions in which the set of arcs S selected by *inPath* is such that: (i) there must not be two arcs starting at the same node, and (ii) there must not be two arcs ending in the same node. Finally, r_6 acts so that all nodes in the graph are reached from the starting node.

Tools for Executing ASP Programs: Architecture and Assessment

The practical application of ASP has been made possible by the availability of efficient implementations. In this section we introduce the general architecture of ASP systems, (Kaufmann et al., 2016) and describe the ASP Competition, (Calimeri et al., 2016) a biannual event conceived to assess the state of the art in ASP evaluation.

Tools for Executing ASP Programs. The first reasonable implementations of the ASP language became available in the second half of the 1990ies. Since then the basic architecture of an ASP system is made of two main components: the Grounder (or instantiator) and the Solver (or model generator). In particular, the input program $\mathcal{P}$, usually read from text files, is first passed to the Grounder that implements a variable-elimination procedure. Basically, the grounder parses $\mathcal{P}$ and generates a ground program $Ground(\mathcal{P})$ that has the same answer sets as $\mathcal{P}$. The produced ground program can be exponentially larger in size with respect to $\mathcal{P}$; thus, concrete grounders employ intelligent procedures that aim at keeping the instantiated program as small as possible. The ground program produced by the Grounder is then evaluated by the Solver, which implements the non-deterministic part of an ASP system. Roughly, a Solver implements some backtracking algorithm to compute answer sets. Finally, once an answer set has been found, ASP systems typically print it in text format.

In the literature, several different techniques for implementing the Grounder and the Solver have been proposed. As a consequence, there are several alternative implementations one can resort to. Among the grounders we mention Gringo (Gebser et al., 2011) and i-DLV (Calimeri et al., 2017); whereas, among the more modern solvers we mention Clasp, (Gebser et al., 2011) WASP, (Alviano et al., 2015) and those from the lp2normal (Bomanson et al., 2014) family. Since, in practice, different combination of tools perform best in different application domains, the selection of the best evaluation strategy has been made automatic in portfolio and multi-engine solvers, such as Claspfolio (Hoos et al., 2014) and ME-ASP (Maratea et al., 2014). It is worth mentioning that, the easiest choice to execute an ASP program is the usage of a monolithic systems (i.e., a system that combines grounder and solver in the same tool), and in this case the most prominent alternatives are DLV, (Leone et al., 2006) and Clingo (Gebser et al., 2011).

Assessment of ASP Systems. The performance of ASP systems is evaluated in ASP Competitions since 2007 (Calimeri *et al.*, 2016). The ASP competitions aim at assessing and promoting the evolution of ASP systems and applications. Its growing range of challenging application-oriented benchmarks inspires and showcases continuous advancements of the state of the art in ASP.

In the following, we spotlight main aspects of the Sixth edition of the ASP competition (the most recent edition of the event at the time of this writing), and in particular we outline the improvements of participant systems carried out to assess the advancement of the state of the art. The competition was open in general to any ASP solving system, provided it is able to parse the standard format called ASP Core 2.0 (Calimeri *et al.*, 2016). Participant systems are run in a uniform setting, on a selection of instances taken from the official benchmarks suite, which is updated every edition with instances from various application domains. In particular, in the Sixth ASP Competition there were 13 competing ASP systems. The instance selection process, carried out according to a rigorous schema that ensures a balance of easy, hard, and very hard instances, ended up in the selection of 560 instances in total from 28 application domains. Each system was allotted 20 min per run, including the time for grounding, and solving. Up to 5 points could be earned per solved instance, corresponding to a perfect maximum total score of 2800. The first three places were taken by the top-performing systems from each team: the multi-engine system ME-ASP (Maratea *et al.*, 2014, 2015) with a score of 1971, the combined system WASP + DLV (Alviano *et al.*, 2015, 2014; Leone *et al.*, 2006) with score 1938, and LP2NORMAL + CLASP integrating dedicated preprocessing (Bomanson *et al.*, 2014) and search (Gebser *et al.*, 2015) with score 1760.

One of the main goals of an ASP competition is to measure the advancement of the state of the art. We report in the following the results of an empirical analysis of the state of the art in ASP solving that was published in Gebser *et al.* (2016). In this experiment the performance of the first three solvers occupying the podium position in the Fifth and the Sixth Competition have been compared on the instances of the more recent edition. Despite only one year has passed between the two events, the substantial improvements in solving technology were evident. All the participants of the Sixth Competition solved more instances than the overall winner of the Fifth edition (labeled CLASP-2014). Notably, the updated versions of LP2NORMAL + CLASP and WASP + DLV (wasp1.5 named the combination of wasp and dlw in 2014) could solve 35 and 171 more instances respectively, which corresponds to substantial improvements of 198%, and 114% respectively. It is also notable the performance of the me-asp system, that was able to solve more instances than any other by combining in a portfolio all solvers from the Fifth edition. This assessment proves that the advancement of the state of the art in ASP solving was consistent and continuous in the last few years.

Applications of ASP

ASP has been successfully applied for solving complex problems in several research areas (cf. (Erdem *et al.*, 2016) for a complete survey) such as Artificial Intelligence (Balduccini *et al.*, 2001; Gagli *et al.*, 2015) Databases (Manna *et al.*, 2015; Marileo and Bertossi, 2010) industrial applications (Grasso *et al.*, 2011) etc. To show the applicability in bioinformatics, in the following, we focus on the applications of ASP in this research area, such as phylogenies reconstruction, modeling biochemical reactions and genetic regulations, modeling biological signaling networks, prediction of protein structure, complex querying over biomedical ontologies and databases. In particular, ASP has been used to solve a number of problems connected to the reconstruction of phylogenies for specified taxa. In Brooks *et al.* (2007) ASP was used for reconstructing phylogenies for *Alcataenia* species; and the same methods have been extended for reconstructing phylogenetic networks in Erdem (2006). A general ASP-based solution to the supertree construction problem has been given in Koponen *et al.* (2015) and the validity of the proposed ASP-based method has been verified by computing a genus-level supertree for the family of cats (Felidae).

An ASP-based model of biochemical reactions and genetic regulations has been proposed in Gebser *et al.* (2011) to detect and explain inconsistencies between experimental profiles and influence graphs. As an example, they could find out inconsistencies in the profile data of SNF2 knock-outs by comparing it with the yeast regulatory network.

A model of a biological signaling network was proposed in Nam and Baral (2009) which can be used to generate hypotheses about the various possible influences of a tumor suppressor gene on the p53 pathway.

The problem of predicting the structure of a protein has been approached with ASP in Campeotto *et al.* (2015) with the goal of finding a folding in the 2D square lattice space, that maximizes the number of hydrophobic contacts between given amino acids.

Methods for solving the Haplotype Inference by Pure Parsimony (HIPP) problem and its variations with ASP have been studied in Erdem *et al.* (2009).

We finally mention some applications of ASP to query answering over biologic, and biomedical databases. In particular, ASP was used in the NMSU-PhyloWS project (Le *et al.*, 2012) to query the repository CDAOStore of phylogenies with the goals of finding similarities w.r.t. distance measures, or for computing optimal clades for taxa in specified trees. In Esra and Öztok (2015) ASP has been used to answer complex queries over biomedical ontologies and databases. The ASP-based query answering systems can find answers and explanations to some complex queries related to drug discovery over databases such as BIOGRID, CTD, DRUGBANK, PHARMGKB, and SIDER and DISEASE ONTOLOGY.

Future Directions and Closing Remarks

In this article we introduced a powerful knowledge representation and reasoning formalism, that can be used to solve complex problems. In particular, we have introduced language and tools for Answer Set Programming (ASP). We have also shown that ASP

has the potential to play a crucial role in the solution of complex problems in bioinformatics, as it is witnessed by a number of successful applications in this area. As far as future directions are concerned, given that ASP provides both a declarative problem solving framework for combinatorial optimization problems and a knowledge representation and reasoning framework for knowledge-intensive reasoning tasks, it can be envisaged that a number of similar problems and challenges can be effectively approached and solved with ASP. Real applications often point out weaknesses of implementations, and ASP systems will be subject of future developments and extensions to effectively answer to the computational challenges arising in bioinformatics.

See also: Algorithms for Graph and Network Analysis: Graph Alignment. Artificial Intelligence and Machine Learning in Bioinformatics. Artificial Intelligence

References

- Alviano, Mario, Dodaro, Carmine, Leone, Nicola, Ricca, Francesco, 2015. Advances in WASP. *LPNMR*, volume 9345 of *Lecture Notes in Computer Science*. Springer.
- Alviano, Mario, Dodaro, Carmine, Ricca, Francesco, 2014. Anytime computation of cautious consequences in answer set programming. *TPLP* 14 (4-5), 755–770.
- Apt, Krzysztof R., 2003. *Principles of Constraint Programming*. Cambridge University Press.
- Baader, Franz, Calvanese, Diego, McGuinness, Deborah L., Nardi, Daniele, Patel-Schneider, Peter F. (Eds.), 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- Baader, Franz, Sattler, Ulrike, 2001. An overview of tableau algorithms for description logics. *Studia Logica* 69 (1), 5–40.
- Balduccini, Marcello, Gelfond, Michael, Watson, Richard, Nogueira, Monica, 2001. The USA-advisor: A case study in answer set planning. *LPNMR*, volume 2173 of *Lecture Notes in Computer Science*. Springer.
- Bomanson, Jori, Gebser, Martin, Janhunen, Tomi, 2014. Improving the normalization of weight rules in answer set programs. *JELIA*, volume 8761 of *Lecture Notes in Computer Science*. Springer.
- Brooks, Daniel R., Erdem, Esra, Erdogan, Selim T., Minett, James W., Ringe, Donald, 2007. Inferring phylogenetic trees using answer set programming. *J. Autom. Reasoning* 39 (4), 471–511.
- Calimeri, Francesco, Fuscà, Davide, Perri, Simona, Zangari, Jessica, 2017. I-DLV: The new intelligent grounder of DLV. *Intelligenza Artificiale* 11 (1), 5–20.
- Calimeri, Francesco, Gebser, Martin, Maratea, Marco, Ricca, Francesco, 2016. Design and results of the fifth answer set programming competition. *Artif. Intell.* 231, 151–181.
- Calvanese, Diego, Giacomo, Giuseppe De, Lenzerini, Maurizio, Nardi, Daniele, 2001. Reasoning in expressive description logics. *Handbook of Automated Reasoning*. Elsevier and MIT Press.
- Campeotto, Federico, Dovier, Agostino, Pontelli, Enrico, 2015. A declarative concurrent system for protein structure prediction on GPU. *J. Exp. Theor. Artif. Intell.* 27 (5), 503–541.
- Eiter, Thomas, Faber, Wolfgang, Leone, Nicola, Pfeifer, Gerald, 2000. Declarative problem-solving using the DLV system. In: Minker, Jack (Ed.), *Logic-Based Artificial Intelligence*. Kluwer Academic Publishers, pp. 79–103.
- Erdem, Esra, 2006. Vladimir Lifschitz, and Donald Ringe. Temporal phylogenetic networks and logic programming. *TPLP* 6 (5), 539–558.
- Erdem, Esra, Erdem, Ozan, Türe, Ferhan, 2009. HAPLO-ASP: Haplotype inference using answer set programming. *LPNMR*, volume 5753 of *Lecture Notes in Computer Science*. Springer.
- Erdem, Esra, Gelfond, Michael, Leone, Nicola, 2016. Applications of answer set programming. *AI Magazine* 37 (3), 53–68.
- Erdem, Esra, Öztok, Umut, 2015. Generating explanations for biomedical queries. *TPLP* 15 (1), 35–78.
- Frühwirth, Thom W., Abdennadher, Slim, 2003. *Essentials of constraint programming*. Cognitive Technologies. Springer.
- Gaggl, Sarah Alice, Manthey, Norbert, Ronca, Alessandro, Wallner, Johannes Peter, Woltran, Stefan, 2015. Improved answer-set programming encodings for abstract argumentation. *TPLP* 15 (4-5), 434–448.
- Gebser, Martin, Kaminski, Roland, Kaufmann, Benjamin, Romero, Javier, Schaub, Torsten, 2015. Progress in clasp series 3. *LPNMR*, volume 9345 of *Lecture Notes in Computer Science*. Springer.
- Gebser, Martin, Kaufmann, Benjamin, Kaminski, Roland, *et al.*, 2011. Potassco: The potsdam answer set solving collection. *AI Commun* 24 (2), 107–124.
- Gebser, Martin, Maratea, Marco, Ricca, Francesco, 2016. What's hot in the answer set programming competition. AAAI. AAAI Press.
- Gebser, Martin, Schaub, Torsten, Thiele, Sven, Veber, Philippe, 2011. Detecting inconsistencies in large biological networks with answer set programming. *TPLP* 11 (2-3), 323–360.
- Gelfond, Michael, Lifschitz, Vladimir, 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput* 9 (3/4), 365–386.
- Gettier, Edmund L., 1963. Is justified true belief knowledge? *Analysis* 23 (6), 121.
- Goble, Lou, 2001. *The Blackwell guide to philosophical logic*. Oxford, UK; Malden, Mass., USA: Blackwell.
- Grasso, Giovanni, Leone, Nicola, Manna, Marco, Ricca, Francesco, 2011. ASP at work: Spin-off and applications of the DLV system. *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, volume 6565 of *Lecture Notes in Computer Science*. Springer.
- Hentenryck, Pascal Van, Michel, Laurent, 2005. *Constraint-based local search*. MIT Press.
- Hintikka, J., 1962. *Knowledge and Belief*. Cornell University Press.
- Hoos, Holger, Lindauer, Marius Thomas, Schaub, Torsten, 2014. claspfolio 2: Advances in algorithm selection for answer set programming. *TPLP* 14 (4-5), 569–585.
- Kaufmann, Benjamin, Leone, Nicola, Perri, Simona, Schaub, Torsten, 2016. Grounding and solving in answer set programming. *AI Magazine* 37 (3), 25–32.
- Kautz, Henry A., Selman, Bart, 2007. The state of SAT. *Discrete Appl Math* 155 (12), 1514–1524.
- Koponen, Laura, Oikarinen, Emilia, Janhunen, Tomi, Säilä, Laura, 2015. Optimizing phylogenetic supertrees using answer set programming. *TPLP* 15 (4-5), 604–619.
- Leone, Nicola, Pfeifer, Gerald, Faber, Wolfgang, *et al.*, 2006. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.* 7 (3), 499–562.
- Le, Tiep, Nguyen, Hieu, Pontelli, Enrico, Son, Tran Cao, 2012. ASP at work: An ASP implementation of phylows. *ICLP (Technical Communications)*, 17 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.
- Manna, Marco, Ricca, Francesco, Terracina, Giorgio, 2015. Taming primary key violations to query large inconsistent data via ASP. *TPLP* 15 (4-5), 696–710.
- Maratea, Marco, Pulina, Luca, Ricca, Francesco, 2014. A multi-engine approach to answer-set programming. *TPLP* 14 (6), 841–868.
- Maratea, Marco, Pulina, Luca, Ricca, Francesco, 2015. Multi-level algorithm selection for ASP. *LPNMR*, volume 9345 of *Lecture Notes in Computer Science*. Springer.
- Marileo, Mónica Caniupán, Bertossi, Leopoldo E., 2010. The consistency extractor system: Answer set programs for consistent query answering in databases. *Data Knowl. Eng.* 69 (6), 545–572.
- McCarthy, J., 1958. Programs with common sense. In: *Proceedings of the Teddington Conference on the Mechanisation of Thought Processes*, pp. 77–84.
- Michel, Chein, Mugnier, Marie-Laure, 2009. *Graph-based Knowledge Representation – Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing. Springer.
- Nam, Tran, Baral, Chitta, 2009. Hypothesizing about signaling networks. *J. Applied Logic* 7 (3), 253–274.
- Niemelä, Ilkka, 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Ann. Math. Artif. Intell.* 25 (3-4), 241–273.
- Robinson, J.A., 1965. A machine-oriented logic based on the resolution principle. *J. ACM* 12 (1), 23–41.
- Zbigniew, Lonc, Truszczynski, Mirosław, 2006. Computing minimal models, stable models and answer sets. *TPLP* 6 (4), 395–449.

Machine Learning in Bioinformatics

Jyotsna T Wassan, Haiying Wang, and Huiru Zheng, Ulster University, Newtonabbey, Northern Ireland, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

The study of biological data certainly found its wide application in improving the sustainability of biological processes and hence the life on our planet. There is increasing interest in investigating computational methods for storing, managing, and analysing biological data which is growing exponentially. ML techniques serve as potential computational methods for extracting knowledge from the biological data (Larranaga *et al.*, 2006). ML experiments are paramount for bioinformatics as they aid in building models for detecting patterns within biological inputs like genomes, health-care data, protein structures etc. and analysing them. ML models learn from and make predictions on biological data (Witten *et al.*, 2016), with a functional approximation (Eq. (1)).

$$f(X) = Y \quad (1)$$

where X acts as biological data, f acts as a functional approximation over X and Y is the new derived biological knowledge. ML techniques are applied mainly in two biological domains:- (i) OMIC Sciences and (ii) Systems Biology (Fig. 1; Larranaga *et al.*, 2006).

ML techniques can be applied in analysing data from varied biological areas (Tarca *et al.*, 2007) as follows:

- Predicting the gene locations and their biological roles
- Predicting protein or molecular functions and studying interactions.
- Predicting diseases from molecular samples
- Sequence analysis
- Drug discovery
- Phylogenetic analysis
- Analysing biological images and components and the list is dynamic and extensible.

However, data analysis is complex as a huge amount of heterogeneous data is generated in the biological processes. The current research needs include development and application of ML algorithms to further study the voluminous and varied biological compositions (Tarca *et al.*, 2007). The aim of this article is to provide insights on ML methods useful in bioinformatics categorizing their varied types, challenges, and applications to provide a general context on the importance of computational techniques in biological sciences for naïve readers. The article is organised as follows. Section “Definition” defines ML and its phases with enlisting of different kinds of ML. Section “Feature Selection and Extraction in Bioinformatics” is focussed on introducing feature selection procedures. Various representative algorithms for modelling ML procedures are described in Section “Representative ML Algorithms”. Section “Assessment of the Performance of ML Algorithms” describes validation indexes for supervised and unsupervised ML models. Section “Other Emerging Paradigms in ML” enlists the emerging paradigms of ML and Section “Machine Learning Software” enlists the emerging ML Tools useful in computational biology. Section “Applications in Bioinformatics and Systems Biology” highlights few of the biological applications of ML with key highlights on literature.

Definition

The term “Machine Learning” was coined by Arthur Samuel at IBM in 1959 (Kohavi and Provost, 1998). The field has evolved from the applied computational statistics, pattern recognition and artificial intelligence. Hence, ML is defined as an ensemble of computational techniques for predictive data analytics facilitating data-driven decisions based on learning from input data.

(1) Training and Testing

ML consists of two major phases of training and testing the model which aid in creating a system to answer the biological questions (Fig. 2).

In the training step, data is trained to incrementally improve the model’s ability for predicting the output. Once the training is complete, the built model is tested against data that has never been used for training and is evaluated to judge how the model might perform in the real-world application (Carbonell *et al.*, 1983).

(2) Types of Machine Learning

ML algorithms are categorized based on predictive, descriptive, or inductive modelling of functional tasks. The main categories are listed as follows (Ayodele, 2010).

- Supervised Learning:** This class is mainly targeted at predictive modelling of tasks based on predefined input data-labels (Kotsiantis *et al.*, 2007). A predictive model is constructed from training data to model functional relationships or dependencies between the input data and the target prediction output (labels). The target values (predictions/labels) for new data is based on the learned relationships from the previous data inputs. The supervised learning algorithms dealing with categorical targets are known as “classification” algorithms and the algorithms for continuous data are termed as

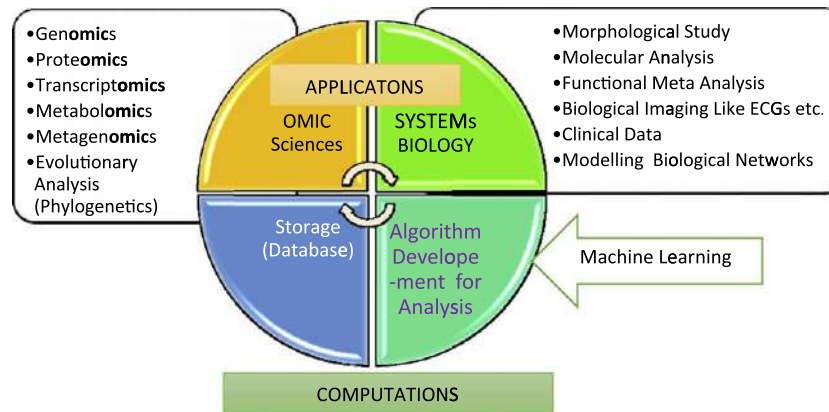


Fig. 1 Applications of ML in biology.

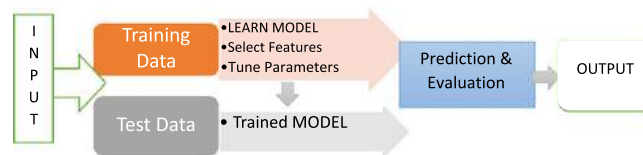


Fig. 2 Phases of ML modelling.

“regression” algorithms. The commonly used classification algorithms are: Decision trees, Random forests, Support vector machines, Neural Networks, Nearest Neighbor, and Naïve Bayes (Kotsiantis *et al.*, 2007). The regression algorithms include linear or multivariate regression logistic regression with lasso, ridge, or elastic-net penalties (Kotsiantis *et al.*, 2007; Carbonell *et al.*, 1983).

- ii. *Unsupervised Learning*: This is purely a data-driven approach and aids in descriptive modelling of data. There are no predefined target labels as described in supervised learning (Ayodele, 2010; Hyvärinen *et al.*, 2010). The unsupervised techniques are applied directly to the input data to detect patterns, rules or summaries which aid in deriving new meaningful insights from the data. Algorithms used to group input data points based on data properties into different classes (labels) are known as “clustering algorithms”. Some of the clustering algorithms include k-means, k-medoids, and hierarchical clustering (Hyvärinen *et al.*, 2010). The mining of association between different input variables (items) in data, aids in determining patterns and frequent item-sets and is known as “Association Rule Mining” (Karthikeyan and Ravikumar, 2014).
- iii. *Semi-Supervised Learning*: This kind of learning is a blend of supervised and unsupervised learning (Chapelle *et al.*, 2009). Labelled data requires expertise and unlabelled data is cheap; semi-supervised learning gains from both the factors. It is also known as inductive learning. The learning progresses using distances and densities in between data points and assumptions on the underlying distribution of data, such as points which are close to each other distance-wise, are more likely to share an output label.

Feature Selection and Extraction in Bioinformatics

Bioinformatic studies are influenced by choice of biological features and their importance across the huge number of biological samples. A natural question to ask is “which biological features are significant for predicting biological roles in analysis?” Feature selection is the process of reducing the size of biological sample space to attain only relevant biological features for improving the quality of the results is constantly emerging as a pre-cursor step in large-scale biological analyses. The commonly used strategies for feature selection are (Saeys *et al.*, 2007):- (a) Filter based approaches; (b) Wrapper based approaches and (c) Embedded Approaches.

Filter methods extract features based on general data properties, for example ranking features based on their significance or correlation with biological roles. The filters incorporate various methods such as correlation-based feature selection, ranking features based on probabilistic conditional distribution such as entropy, dependency between the joint distribution of the biological features and biological roles (Saeys *et al.*, 2007). Wrappers evaluate biological features by estimating their accuracy using a machine learning scheme itself (Kohavi and John, 1997). They are based on the interaction between an ML algorithm and the training data using some search algorithm or an optimization in either a forward or backward search over the biological input data. Embedded methods tend to in-cooperate the feature selection strategy as part of the ML model (Kohavi and John, 1997).

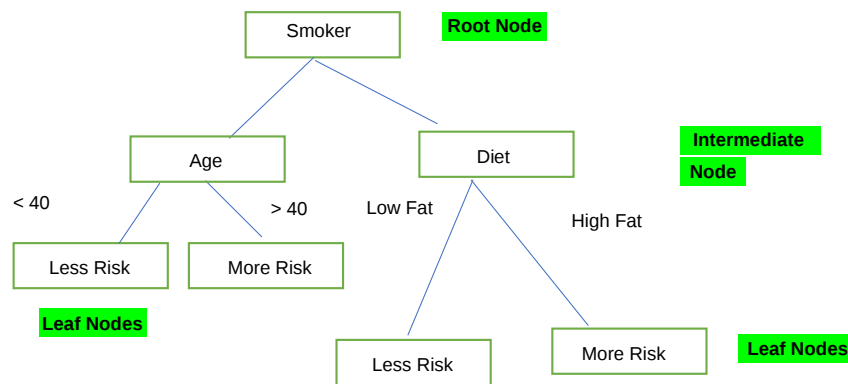


Fig. 3 A sample decision tree.

Feature extraction techniques like Principle Coordinate analysis (PCoA), Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA) etc., transform the data to a different space, reducing dimensionality and are useful in analysing and visualizing the data in bioinformatics (Saeyns *et al.*, 2007).

Representative ML Algorithms

This section presents a brief introduction to a list of representative ML algorithms for biological predictions.

(1) Classification

In a classification problem, the set of data items are needed to be divided into groups known as classes. Given an instance of the item-set, a class is assigned according to some of the features of items and a set of classification rules. Some of the famous classification algorithms are listed below.

a. Decision Trees

Decision trees represent hierarchical representation of relationships between various features in the biological data. The method depends on classifying a set of features through a series of questions on a tree traversal path, in which the answer of the current question determines the answer to the next question (Quinlan, 1986). Decision tree builds classification in the form of a tree structure as shown in Fig. 3. The sample figure (Fig. 3), depicts the risk of health hazards to a person depending on smoking status and diet plan. The leaf nodes represent the samples classified into categories/classes of subjected to less risk and more risk hazards. Popular decision tree algorithms, include ID3 (Iterative Dichotomiser 3), C4.5 Algorithm, CART (Classification and Regression Tree), CHAID and QUEST (Almana and Aksoy, 2014). The method simplifies complex relationships between input biological features and target classes by dividing original input variables into significant subgroups.

b. Random Forest and XGBoost

Random Forest creates the forest structure with randomized construction and integration of several decision trees to classify item sets. To classify a new sample based on biological features, each decision tree gives a classification and the tree "votes" for a class are counted for final decisions (Breiman, 2001). It takes a random sample of observation and randomly chosen initial items/features to build a decision tree model. The process is repeated many times and the final prediction is a function of each prediction derived from different constituent decision trees. This final prediction can simply be the mean of each prediction. It captures the variance of several input trees and hence supports good predictive ability even over large biological data sets. To make Random Forest faster, XGBoost (Chen and Guestrin, 2016) is proposed, which exploits out-of-core computation and scales to larger data with the least amount of computing resources (Chen and Guestrin, 2016).

c. Support Vector Machines

The algorithm starts with plotting each data/feature item as a point in n-dimensional space with the value of each nth feature being a coordinate in geometric plane. For example, if there are two sample features (e.g., age and smoking status) w.r.t. an individual, the two variables are plotted in a two-dimensional space; where each point has two co-ordinates known as Support Vectors (Drucker *et al.*, 1997). The aim of this model is to find a line that splits the data between the two classes such that it maximizes the distance-gap between the closest point in each of the two groups (Fig. 4). Hence, the line acts as a classifier.

d. Artificial Neural Networks

Artificial Neural Networks (Graupe, 2007), originated from the idea of mathematically modelling the human brain. In an Artificial Neural Network, the input biological features are forwarded to sequenced layers of neurons (the processing units), also known as nodes, in combination with associated weight thresholds (Eq. (2)). The combination drives the

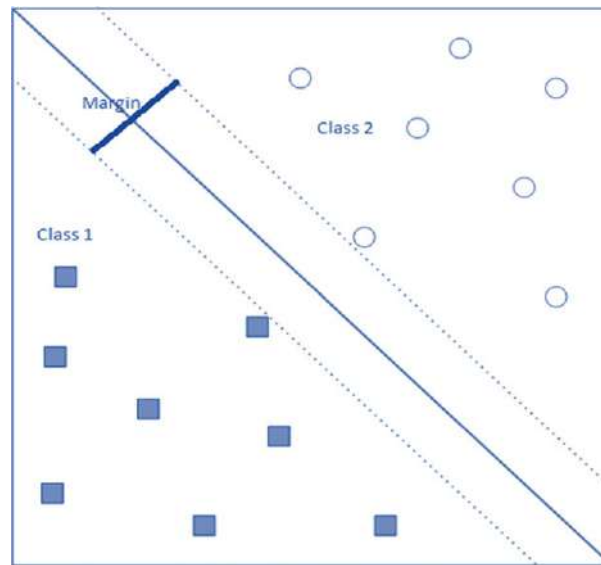


Fig. 4 Support vector machines.

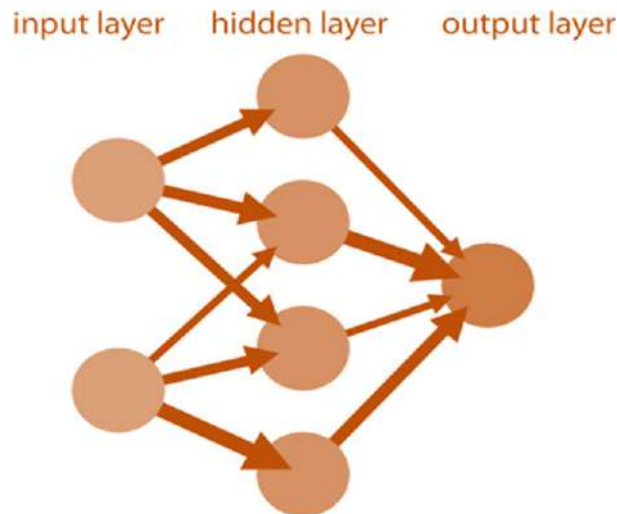


Fig. 5 The neural networks.

ability to perform classification at output layer (Graupe, 2007; Fig. 5).

$$OUTPUT = \sum W_i X_i \text{ where } i = 0 \text{ to } n_o : \text{ of inputs, } W_i : \text{ weights and } X_i : \text{ inputs} \quad (2)$$

The processing units are organised in layers (shown in Fig. 5). In multilayer perceptron the values of the parameters (weights) may be determined by the backpropagation algorithm.

Shen and Chou (2006), integrated multiple classifiers to create an ensemble for genomics studies to obtain a more concrete result in classification of samples.

(2) Clustering

Clustering is based on intrinsic grouping, i.e., it is a technique which arranges biological items into groups such that the members of a group are most like each other and are dissimilar to the items belonging to other groups. The groups are known as clusters. Clustering methods can be classified into following main categories (Gan et al., 2007).

- *Partitioning methods:* Given n data items, these methods are based on creating k partitions of the data, where each partition represents a cluster such that $k \leq n$. These use an iterative shifting or relocation technique for moving items from one cluster group to another, till the best results are achieved. Two popular heuristics methods are (Gan et al., 2007):- (i) K-means algorithm and (ii) K-medoids algorithm.

- **Hierarchical methods:** A hierarchical method is based on hierarchical decomposition of the given set of items. A hierarchical method could be either agglomerative (bottom-up) or divisive (top-down), based on the direction of decomposition. In each successive step, a cluster is split into smaller clusters, until eventually, each object is in any of the cluster. Other dominant methods for clustering are:- (i) *Density based methods*, which consider clustering until the density (number of items) in the cluster exceeds a given threshold and may result in clusters of an arbitrary shape; (ii) *Grid based methods*, which are based on quantizing and clustering the items into finite number of cells forming a grid structure and hence depends on number of cells in each dimension in the quantized space; (iii) *Model based methods* that hypothesize clustering by constructing model for each of the clusters and finding the best fit of the data to the given model (Gan et al., 2007).

(3) Association Rule Mining

The process is associated with link analysis, alternatively also known as affinity analysis or association analyses. It facilitates the data-mining tasks of uncovering relationships amongst items within the biological samples. An association rule is a model that captures the relationships among items based on their patterns of co-occurrence across samples. An implication expression for the association rule is of the form $X \rightarrow Y$, where X and Y are feature item-sets (Karthikeyan and Ravikumar, 2014). The analysis is useful in bioinformatics; e.g., to find co-expressed genes with their functions, to determine combinations of structural properties in biomolecules, to find patterns of frequently co-occurring annotations derived with frequent itemset in protein interactions, etc.

(4) Deep Convolutional Nets

The biologically inspired and enhanced variant of Neural Networks is *Convolutional Neural Deep Networks (ConvNets)*. They consider the input as mostly images and their architecture have neurons arranged in 3 dimensions of activation volume: width (w), height (h), depth (d), unlike 2D Neural Networks (Fig. 6; Anon, 2017).

Such networks have successfully been used in drug discovery. Potential treatments could be derived from predicting the interaction between molecules and biological proteins. The first deep learning Neural Network, *AtomNet* was introduced for structure-based rational drug design (Wallach et al., 2015).

Assessment of the Performance of ML Algorithms

Assessment methods determine the extent to which ML models demonstrate the desired learning outcomes. The main objective is to learn an ML model that attains a good generalization performance by maximizing the prediction accuracy or, in other words, minimizing the probability of making a wrong biological prediction. The most important mathematical modelling for evaluating supervised ML prediction is driven by a confusion matrix. A confusion matrix and related measures are shown in Fig. 7 (Powers, 2011). A confusion matrix is an $Num \times Num$ matrix, where *Num* represents the number of classes being predicted. In the given example (Fig. 7), *Num* is taken as 2 (i.e., binary classification case).

The quality of unsupervised learning (clustering) is determined by following numerical indexes (Halkidi et al., 2002):

- Cluster cohesion as the sum of the weight of all links between items within a cluster (Intra-Cluster Index)
- Cluster separation as the sum of the weights between items in the cluster and items outside of the cluster (Inter-Cluster Index)

Other Emerging Paradigms in ML

This section enlists two of the emerging ML techniques.

- (1) **Reinforcement Learning:** This field of ML is inspired by the systematic approach of behaviourism in psychology. The environment is majorly implemented as a Markov decision process (MDP) which relies on learning agents taking actions in a

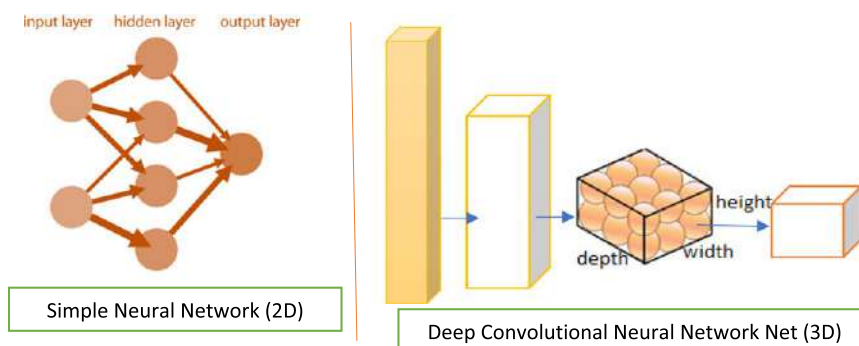


Fig. 6 Difference between simple and deep neural net. Available at: <http://cs231n.github.io/convolutional-networks/>.

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predicted Value	$a/(a+b)$
	Negative	c	d	Negative Predicted Value	$d/(c+d)$
		Sensitivity $a/(a+c)$	Specificity $d/(d+b)$	Accuracy = $(a+d)/(a+b+c+d)$	
AUC = plot between sensitivity and (1- specificity)					

Where a denotes the number of true positives (positive samples correctly classified as positive), b denotes the number of false negatives (positive samples incorrectly classified as negative), c represents the number of false positives (negative samples incorrectly classified as positive), and d represents the number of true negatives (negative samples correctly classified as negative).

Fig. 7 Assessment measures for supervised learning.

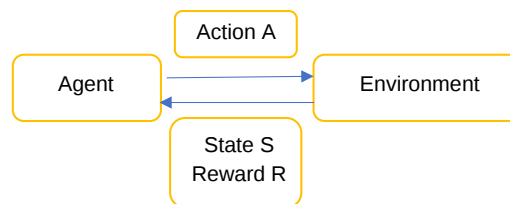


Fig. 8 The process of reinforcement learning.

given environment and moving from one problem state to another (Kaelbling *et al.*, 1996). Each action is associated with a reward. The aim of a reinforcement learning agent is to collect maximum rewards as possible by performing actions. To summarize, MPD consists of following (Puterman, 2014; Fig. 8):- a set of environment and agent states, S ; a set of actions, A , of the agent; the probability of transition from one state to another state belonging to S under an action $a \in A$; the reward while transition; rules describing the agents' observations.

- (2) *Extreme Machine Learning (ELM)*: ELM proposed by Huang *et al.* (2004), is a novel generalized learning algorithm for single-hidden-layer Neural Networks. In ELM, the input weights and hidden biases are generated randomly, and the output weights are calculated by regularized Least-Square method, which makes it faster. The model learns the output weights amounting to a linear model. There is no tuning of modelling parameters for propagation in the backward direction like in Neural Networks. Hence it is feed-forward Neural Network with only one hidden layer. Due to high performance, ELM is becoming a significant area in ML especially for pattern recognition in ML (Huang *et al.*, 2004).

Machine Learning Software

ML software/s have become integral part of data science solving problems of biological domain. The tools aid in data preparation, and predictive modelling. The four commonly used tools are listed below to enhance the learning experience for readers.

- (1) **RapidMiner**
RapidMiner, formerly known as YALE (Yet Another Learning Environment), was first developed in 2001 by Ralf Klinkenberg, Ingo Mierswa, and Simon Fischer at the Technical University of Dortmund in java programming language (Data-mining-blog.com, 2017). It serves as an open source Graphical User Interface (GUI) tool relying on a template-based block-diagram approach in which predefined blocks act as plug-ins. The product is based on ETL, i.e., extract, transform and load data phenomena. It was declared as the most popular data analytics software in 2016 by annual software poll conducted by KDnuggets (Jupp *et al.*, 2011). RapidMiner is suited for data related to biological Sciences. The RapidMiner Plugin for Taverna (Jupp *et al.*, 2011), allowed a mass application of ML tools to bioinformatics workflows (Data-mining-blog.com, 2017).
- (2) **BioWeka**
The Waikato Environment for Knowledge Analysis (Weka) supports various classification, regression, clustering algorithms, with various data pre-processing methods such as feature selection with good visualization (Hall, 1999). Gewehr *et al.* (2007), introduced bioinformatic methods to Weka as part of BioWeka project as shown in Fig. 9, which proved useful for biological studies.

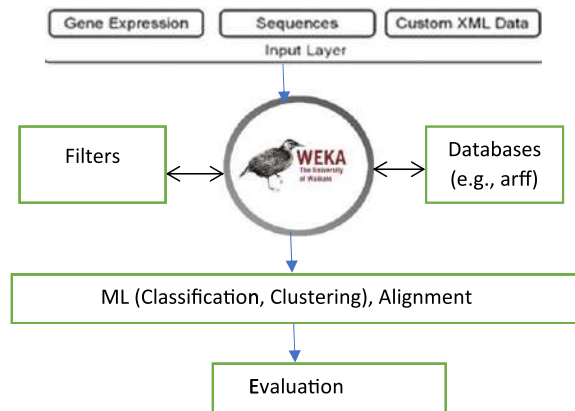


Fig. 9 BioWeka. Reproduced from Gewehr, J.E., Szugat, M., Zimmer, R., 2007. BioWeka – Extending the Weka framework for Bioinformatics. *Bioinformatics* 23 (5), 651–653.

(3) MATLAB

MathWorks product MATLAB ([Uk.mathworks.com](http://uk.mathworks.com), 2017), provides an integrated environment for modelling and simulating biological systems. Bioinformaticians are using MATLAB and related toolboxes to perform data analysis over sequences, microarrays etc., mass spectrometry for measuring bio-chemical compounds, medical image processing, bio-statistics, and to simulate biological networks among other computations. The SimBiology library in MATLAB provides a graphical and modelling tool for systems biology and a specialized field of pharmacokinetics ([Uk.mathworks.com](http://uk.mathworks.com), 2017). The tool is widely used in automating the biological workflows with computational techniques. It is also useful to acquire real data from medical instruments (like heart rate recorder), cards, sensors, read or write biological data including signals, images, omics data from files, databases, web and excel for simulating biological systems with an intuitive graphical interface. MATLAB is proved to be useful in biological visualizations. For example, BrainMaps are used to analyse massive, high-resolution brain image data at BrainMaps.org ([Uk.mathworks.com](http://uk.mathworks.com), 2017).

As an alternative to MATLAB, various open source programming languages like R, Python and Scala are emerging to support various ML packages, useful in biological systems ([Shameer et al., 2017](#); [Faghri et al., 2017](#)).

(4) R-Project

R-Project is emerging as a most powerful statistical tool and offers various useful packages for analysis and visualization of biological data. The Bioconductor-project ([Gentleman, 2006](#)), is an effective platform for the analysis of genomic data. Bioconductor supports various useful packages such as:- *ape* package aids in analysis of phylogenetics and evolution, *ade4* is useful for multivariate analysis of genetic markers, *picante* is useful for ecological studies, *affy* supports analysis of Affymetrix gene chip data at prob level, *DEGseq* is an R package for identifying differentially expressed genes from RNA sequence data, *ShortRead* is helpful in exploration of high throughput sequence data, *WGCNA* is an R package for weighted correlation network analysis, *limmaGUI* is aimed at linear modelling of microarray data, *DeepBlueR* is helpful in epigenomic analysis in R, etc. (Bioconductor.org, 2017). Recently cancer-subtypes has been developed for identifying molecular subtype identification, validation, and visualization ([Xu et al., 2017](#)).

Applications in Bioinformatics and Systems Biology

ML is revolutionizing the world of biological sciences. ML plays an important role in analysing the interpreting data from biological including DNA, RNA, protein structures, gene sequences (ATGC bases), microarray gene expression data, biological networks and pathways, biological signals, and biological images. Hence it supports both OMIC sciences and systems biology. The varied applications are highlighted in [Fig. 10](#). The field is also playing major role in detection of diseases and optimizing biological measures therapeutically. ML has been applied to diagnose and predict critical diseases like cancer prognosis ([Kourou et al., 2015](#)), autism diagnostic ([Bone et al., 2015](#)), chronic disease ([Chen et al., 2017](#)), etc. to support man-kind. [Plummer et al. \(2015\)](#), reviewed three bioinformatics pipelines:- (i) MG-RAST: Meta Genome Rapid Annotation using Subsystem Technology; (ii) QIIME: Quantitative Insights into Microbial Ecology; (iii) *mothur* for the analysis of preterm gut microbiota using 16S rRNA gene sequencing data.

ML methods are useful for analysing microarray gene expression data ([Pirooznia et al., 2008](#)). [Lin et al. \(2002\)](#), applied supervised ML in functional genomics for conserving codon composition of ribosomal protein coding genes in tuberculosis. Pharmacogenomics is an emerging field for understanding genetics in the context of how a patient respond to drugs. [Gacesa et al. \(2016\)](#) highlighted that ML can differentiate venom toxins from other proteins having non-toxic physiological functions. Biological sciences are not only useful in human health but also in field of agriculture to improve soil quality, crop yield, cattle health, etc. [Wassan et al. \(2017\)](#) contributed towards metagenomics pipeline for the prediction of impact of feed additives on cattle health.

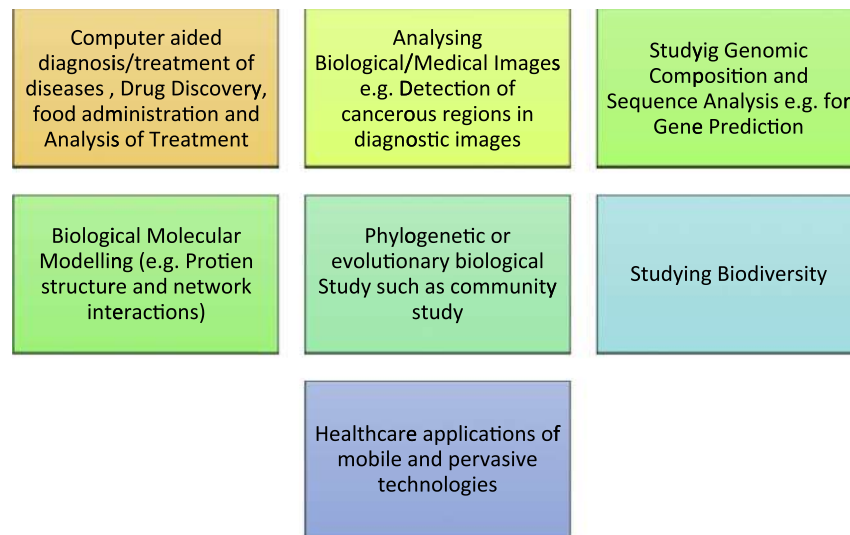


Fig. 10 Applicative areas of ML in bioinformatics.

Final Remarks

In this article, we summarized various ML techniques and tools that are useful for analysing biological data, ranging from training, testing, and validating the input data for novel predictions. The data science tools such RapidMiner, BioWeka etc.; ML techniques such as classification, clustering, association analysis etc.; various bioinformatics ML code libraries such as BioPython, BioPerl, BioRuby, and R-Bioconductor are also useful in varied applications of biology. An application of ML techniques helps to focus on finding the most plausible patterns and predictions from biological data including Next Generation Sequencing data analysis, gene assembly, annotation, phylogeny, molecular structural modelling, and gene expression analysis. The future in related context lies in developing scalable ML algorithms for bioinformatics as biological inspired data is growing exponentially (Faghri *et al.*, 2017).

See also: Algorithms for Graph and Network Analysis: Graph Alignment. Artificial Intelligence and Machine Learning in Bioinformatics

References

- Almana, A., Aksoy, M., 2014. An overview of inductive learning algorithms. *International Journal of Computer Applications* 88 (4), 20–28.
- Anon, 2017. Neural networks. [Online] Available at: <http://cs231n.github.io/convolutional-networks> (accessed 18.12.17).
- Ayodele, T., 2010. Types of machine learning algorithms. In: *New Advances in Machine Learning*. InTech.
- Bioconductor.org, 2017. Bioconductor – BioViews. [online] Available at: <https://bioconductor.org/packages> (accessed 28.12.17).
- Bone, D., Goodwin, M., Black, M., *et al.*, 2015. Applying machine learning to facilitate autism diagnostics: Pitfalls and promises. *Journal of Autism and Developmental Disorders* 45 (5), 1121–1136.
- Breiman, L., 2001. Random forests. *Machine Learning* 45 (1), 5–32.
- Carbonell, J., Michalski, R., Mitchell, T., 1983. An overview of machine learning. In: *Machine Learning*. Springer Berlin Heidelberg, pp. 3–23.
- Chapelle, O., Scholkopf, B., Zien, A., 2009. Semi-supervised learning. *IEEE Transactions on Neural Networks* 20 (3), 542.
- Chen, M., Hao, Y., Hwang, K., Wang, L., Wang, L., 2017. Disease prediction by machine learning over big data from healthcare communities. *IEEE Access* 5, 8869–8879.
- Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM.
- Data-mining-blog.com., 2017. RapidMiner at CeBIT 2010: The enterprise edition, rapid-I and cloud mining – Data mining – Blog.com. [Online] Available at: <http://www.data-mining-blog.com/cloud-mining/rapidminer-cebit-2010/> (accessed 20.12.17).
- Drucker, H., Burges, C.J., Kaufman, L., Smola, A.J., Vapnik, V., 1997. Support vector regression machines. *Advances in Neural Information Processing Systems*. 155–161.
- Faghri, F., Hashemi, S.H., Babaeizadeh, M., *et al.*, 2017. Toward scalable machine learning and data mining: The bioinformatics case. *arXiv preprint*.
- Gacesa, R., Barlow, D., Long, P., 2016. Machine learning can differentiate venom toxins from other proteins having non-toxic physiological functions. *PeerJ Computer Science* 2, e90.
- Gan, G., Ma, C., Wu, J., 2007. *Data Clustering*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Gentleman, R., 2006. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York: Springer.
- Gewehr, J.E., Szugat, M., Zimmer, R., 2007. BioWeka – Extending the Weka framework for Bioinformatics. *Bioinformatics* 23 (5), 651–653.
- Graupe, D., 2007. *Principles of Artificial Neural Networks*. World Scientific.
- Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2002. Cluster validity methods: Part I. *ACM SIGMOD Record* 31 (2), 40–45.
- Hall, M.A., 1999. Correlation-based feature selection for machine learning. Thesis, University of Waikato.
- Huang, G., Zhu, Q., Siew, C., 2004. Extreme learning machine: A new learning scheme of feedforward Neural Networks. In: *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990. IEEE.
- Hyvärinen, A., Gutmann, M., Entner, D., 2010. *Unsupervised machine learning*.

- Jupp, S., Eales, J., Fischer, S., *et al.*, 2011. Combining RapidMiner operators with bioinformatics services. A powerful combination. In: Proceedings of the RapidMiner Community Meeting and Conference. Shaker.
- Kaelbling, L., Littman, M., Moore, A., 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4, 237–285.
- Karthikeyan, T., Ravikumar, N., 2014. A survey on association rule mining. *International Journal of Advanced Research in Computer and Communication Engineering* 3 (1), 2278. ISSN: 2278-1021.
- Kohavi, R., John, G., 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97 (1–2), 273–324.
- Kohavi, R., Provost, F., 1998. Glossary of terms. *Machine Learning* 30 (2–3), 271–274.
- Kotsiantis, S., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. *Informatica* 31, 249–268.
- Kourou, K., Exarchos, T.P., Exarchos, K.P., *et al.*, 2015. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal* 13, 8–17.
- Larranaga, P., Calvo, B., Santana, R., *et al.*, 2006. Machine learning in bioinformatics. *Briefings in Bioinformatics*. 86–112.
- Lin, K., Kuang, Y., Joseph, J., Kolatkar, P., 2002. Conserved codon composition of ribosomal protein coding genes in *Escherichia coli*, *Mycobacterium tuberculosis* and *Saccharomyces cerevisiae*: Lessons from supervised machine learning in functional genomics. *Nucleic Acids Research* 30 (11), 2599–2607.
- MathWorks, 2017. Deep learning. [Online] Available at: <https://uk.mathworks.com/discovery/deep-learning.html> (accessed 18.12.17).
- Pirooznia, M., Yang, J.Y., Yang, M.Q., Deng, Y., 2008. A comparative study of different machine learning methods on microarray gene expression data. *BMC Genomics* 9 (1), S13.
- Plummer, E., Twin, J., Bulach, D.M., Garland, S.M., Tabrizi, S.N., 2015. A comparison of three Bioinformatics pipelines for the analysis of preterm gut microbiota using 16S rRNA gene sequencing data. *Journal of Proteomics & Bioinformatics* 8 (12), 283.
- Powers, D.M., 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies* 2 (1), 37–63.
- Puterman, M.L., 2014. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons.
- Quinlan, J.R., 1986. Induction of decision trees. *Machine Learning* 1 (1), 81–106.
- Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* 23 (19), 2507–2517.
- Shameer, K., Badgeley, M.A., Miotto, R., *et al.*, 2017. Translational bioinformatics in the era of real-time biomedical, health care and wellness data streams. *Briefings in Bioinformatics* 18 (1), 105–124. doi:10.1093/bib/bbv118.
- Shen, H., Chou, K., 2006. Ensemble classifier for protein fold pattern recognition. *Bioinformatics* 22 (14), 1717–1722.
- Tarca, A.L., Carey, V., Chen, X., Romero, R., Drăghici, S., 2007. Machine learning and its applications to biology. *PLOS Computational Biology* 3 (6), e.116.
- Wallach, I., Dzamba, M., Heifets, A., 2015. AtomNet: A deep convolutional Neural Network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*.
- Wassan, J., Wang, H., Browne, F., *et al.*, 2017. An integrative approach for the functional analysis of metagenomic studies. In: Proceedings of the International Conference on Intelligent Computing, pp. 421–427. Cham: Springer.
- Witten, I., Frank, E., Hall, M., Pal, C., 2016. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.
- Xu, T., Le, T.D., Liu, L., *et al.*, 2017. CancerSubtypes: An R/Bioconductor package for molecular cancer subtype identification, validation, and visualization. *Bioinformatics* 33 (19), 3131–3133.

Intelligent Agents and Environment

Alfredo Garro, University of Calabria, Rende, Italy

Max Mühlhäuser and Andrea Tundis, Darmstadt University of Technology, Darmstadt, Germany

Stefano Mariani and Andrea Omicini, University of Bologna, Bologna, Italy

Giuseppe Vizzari, University of Milano-Bicocca, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The analysis of complex systems, i.e., systems consisting of several interdependent and interacting entities that determine the system behavior, requires the exploitation of new and more effective solutions able address issues ranging from the definition of appropriate modeling formalisms to the use of advanced system analysis methods. Such kinds of systems can be classified in two main categories: Artificial Systems (ASs) that are built by humans and Natural Systems (NSs), already existing in nature without the intervention of humans. In particular,

- Examples of Artificial Systems are nowadays called Cyber-Physical Systems (CPSs) ([Chen and Lu, 2018](#)) or System of Systems (SoSs) ([Garro and Tundis, 2015](#)). Here different components belonging to various application domains (such as Software, Mechanical, Electrical, Electromechanical, and so on), which are originally conceived for working in isolation for a specific purpose, and are integrated in a common environment in order to achieve one or more complex goals ([Sampigethaya and Poovendran, 2012](#); [Zhabelova and Vyatkin, 2012](#)).
- Examples of Natural Systems are represented by Biological Systems (BSs) ([Anderson, 2015](#)), in which a complex network of biological entities belonging to different biological subsystems (such as the nervous system, circulatory system, respiratory system, and so on) work together in a synergistic manner.

Concerning the second example, Intelligent Systems represent a very promising solution in computational biology. They allow the development of complex applications centered on theoretical methods, for supporting data-analysis based on mathematical modeling, and computational simulation techniques for observing social systems and studying biological phenomena on the basis of the so-called emergent behavior ([Seekhao et al., 2016](#); [Adamatti, 2016](#)). Indeed, typically, in these systems it is not enough to observe and analyze the state and output of the single system or individual entity, but it is necessary to observe the way they interact and cooperate, in order to capture particular dynamics resulting from their interactions, which define their emergent behaviors.

It is clear that for such classes of systems, agents provide a suitable research approach, thanks to their key features of autonomy and cooperation. A key role, in the adoption of intelligent agents, is played by the environment, which represents the “problem space” in which the agents operate and in which the agents represent one possible resolution path. It is important to note that the environment can be partially or fully observable. Observability lets the agent retrieve information and perform computation on it in order to take decisions and consequent actions on the basis of the perceived information. Moreover, the environment can be deterministic or stochastic. As a consequence, if the environment is deterministic, its properties are well known and none of them are random, which means that the output of the model is fully determined by the values of its parameters and by its initial conditions. If the environment is stochastic, then some randomness and uncertainty is present in it.

Specific details about Intelligent Agents, Environment, and their Interactions are provided in the rest of the article.

Agents and Intelligent Agents

As stated in the previous section, different but similar definitions for agents have been provided in the literature. A weaker but more general definition of Agents, that may be suited to describe the extremely heterogeneous approaches encountered in the agent-based simulation context, is to see an Agent as an autonomous entity, having the ability to decide the actions to be carried out in the environment and interactions to be established with other agents, according to its perceptions and internal states.

In artificial intelligence, an Intelligent Agent (IA) is an autonomous entity that observes through sensors and acts upon an environment using actuators (i.e., it is an agent) and directs its activity towards achieving goals (i.e., it is “rational”, as defined in economics). Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an intelligent agent.

The most popular definitions of IA were provided (i) by [Smith et al. \(1994\)](#) who stated that “An agent is a persistent software entity dedicated to a specific purpose.”, (ii) by [Hayes-Roth \(1996\)](#) who said that “Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment, action to affect conditions in the environment, and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions”; (iii) by IBM, which said “Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user’s goals or desires.”

According to the vision described by [Russell and Norvig \(1995\)](#) an IA can be seen as an entity in a program or environment capable of generating action. From a more technical perspective, an Intelligent Agent is an entity in a program or environment capable of generating actions ([Magedanz et al., 1996](#)). It uses perception of the status of the environment in order to make decisions about specific actions to take. The perception is represented by the capability or sensitiveness which is typically achieved by sensors; whereas the actions are the reactions to a particular status of phenomena that can depend on the most recent perception or on the entire history (sequence of perceptions). An Intelligent Agent uses and provides functions. A function can be a mathematical function that maps a sequence of perceptions into one or more actions, which is implemented as an agent program. The part of the agent that is in charge of taking an action is called an actuator.

Other important characteristics of agents include the following:

- *Rationality*: an agent is supposed to act in order to achieve its goals and does not act in such a way as to prevent its goals being achieved, at least insofar as its beliefs permit. A rational agent is one that can take the right decision in every situation on the basis of a set of criteria/tests, used to measure the level of performance in term of the success of the agent's behavior. Such performance measures should be based on the desired effect of the agent on the environment. In particular, the agent's rational behavior depends on (i) the performance measure that defines success; (ii) the agent's knowledge of the environment; (iii) the actions that it is capable of performing; (iv) the current sequence of perceptions. In general, for every possible perception sequence, the agent is expected to take an action that will maximize its performance measure.
- *Autonomy*: an agent is able to take initiative and exercise a non-trivial degree of control over its own actions. Indeed, autonomy is the capacity to compensate for partial or incorrect prior knowledge, and it is usually achieved by learning. This is because an agent is not omniscient and, in practice, it does not know all the actual or possible outcomes of its actions. As a consequence, an exploration approach is necessary, through which an agent performs an action to gather information and to increase its perception

Based on such characteristics, well-known agent models have been identified such as Simple Reflex, Model-Based Reflex, Goal-Based, Utility-Based, and Learning.

- *Simple Reflex*: The decision of the action to take is only based on the current perception. The history and the perceptions gathered in the past are neglected. This model is based on *condition-action* rules. This model works if the environment is fully observable (stateless).
- *Model-Based Reflex*: this agent model works when the world is not fully observable. As a consequence, it is important that the agent remember previous observations about the parts of the environment which it cannot observe in a particular period of time. This requires a supporting model for representing the environment (stateful).
- *Goal-Based*: this agent model aims at driving the agent to achieve a specific purpose, and the action to be taken depends on the current state and on what it is trying to accomplish (the goal). Sometimes, achieving the goal requires a single action, in other cases the goal to be reached is complex and decomposed into multiple subgoals, each of which requires one or a set of actions. In this case, the achievement of all subgoals subsumes the achievement of the main goal. Usually in this case, strategies, planning, and sifting through a search space for possible solutions, are necessary.
- *Utility-Based*: this can be seen as a supporting or complementary model to the Goal-Based model previously described. In this case, the agent knows the utility function, which is continuously monitored and exploited to estimate the distance between the goal and the current state of the agent.
- *Learning*: this agent model has the capability of enriching its "knowledge" and abilities by observing and acting accordingly. This means that the agent is able to learn from past experiences in the environment to predict the future and, in some cases, (pro)actively affect the environment.

The next section focuses on the concepts of environment, the role that it plays, and its main properties.

Environment

While the notion of agent is obviously central to this subject, all of the most widespread definitions of agent at least mention the fact that a surrounding *environment* is present, for instance to provide percepts and a context in which actions are attempted. Whereas the awareness of the significance of the environment in which an intelligent agent is situated was already recognized in the earliest version of the most widely adopted book on Artificial Intelligence [Russell and Norvig \(1995\)](#), within the autonomous agents and multi-agent systems (MAS) research community the recognition of the environment as an explicit and essential part of a multiagent system required some time and a systematic analysis of the typical practice of researchers in the area. [Weyns et al. \(2007\)](#), in a foundational paper on this topic, provide the following definition:

The environment is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources.

The authors also highlight the fact that the environment is a first-class abstraction for agent oriented models not just providing the surrounding conditions for agents to exist, but also representing an exploitable design abstraction for building multiagent system applications.

Table 1 Example of different environments and their characterization

Example	Observable	Deterministic	Static	Discrete
Chess	Fully	Yes	Yes	Yes
Poker	Partly	No	Yes	Yes
Real-time strategy	Partly	No	No	No

Russell and Norvig, in the above cited book, provided several dimensions for the characterization of an environment, and in particular the most relevant in this context are:

- *observability*: agents can have complete or partial access to the state of the environment;
- *determinism*: in deterministic environments agents' actions have single, guaranteed effects;
- *dynamism*: in a static environment agents can assume that no change happens during their own deliberation;
- *discreteness*: discreteness can refer to different aspects of the environment, namely its state, the way time is represented and managed, and the perceptions and actions of agents; generally, in a discrete environment there are a fixed, finite number of actions and perceptions.

Examples of environments and their respective characterizations are shown in [Table 1](#). Clearly, the features of the environment heavily influence the design decisions about the agent architecture; it must also be clarified that it is sometimes possible to take a simplified but still acceptable perspective on certain aspects of an environment in order to actually come up with applicable and tractable solutions: for instance, to stay within the gaming example context, in a real-time strategic game the modeler could superimpose a discrete representation of the continuous environment to support the coordination of units patrolling an area.

Recently, [Russell and Norvig \(1995\)](#) also included an additional dimension of analysis that specifies whether the environment includes other agents and, in this case, also the cooperative or competitive attitude of agents (a more thorough analysis of the different types of interaction is provided by [Ferber \(1999\)](#)).

A more recent analysis, provided by Weyns *et al.*, instead of describing inherent features of an environment, considers it from the perspective of a design abstraction supporting the activities of the modeler or engineer. Their analysis considers that three different levels of support can be provided by the environment:

- *basic level*, essentially just enabling the agents to directly access their deployment context;
- *abstraction level*, filling the conceptual gap between the agent abstraction and low-level details of the deployment context (e.g., wrapping physical or software resources and providing access at agents' level of abstraction);
- *interaction/mediation level*, supporting both forms of regulation to the above mentioned resources, as well as mediating the interactions among agents to support forms of coordination.

Whereas most agent oriented platforms provide abstraction level support, support of the interaction/mediation level is generally not as comprehensive and systematic, as indicated by relatively recent works discussing meta-models for multiagent systems that explicitly include abstractions enabling this kind of high-level support (see, e.g., [Omicini et al., 2008a](#)).

Weyns *et al.* also stress the fact that the environment has a fundamental role in *structuring* the system: this is particularly relevant for applications such as the simulation of biological entities in which the defined model actually represents a physical spatial structure (e.g., portions of tissues of a human body), but it can also be relevant in situations in which the model must consider other conceptual structures such as organizational or societal ones (e.g., roles, groups, permissions, policies). Whenever the computational model needs to represent a physical environment, and to explicitly consider its spatial aspects and dynamics, even without involving the modeled agents, the need for a precise and systematic model of perception and action is even more apparent than in other situations. From this perspective, models such as the one described by [Ferber and Müller \(1996\)](#) represent relevant examples of a specific form of high-level support supplied by the environment model.

Finally, once again especially but not exclusively in the biological context, it is often the case that the modeler needs to consider distinct but related dynamics that are more reasonably or effectively represented by employing different spatial or temporal scales. The overall multiagent model could, therefore, include different environmental representations at different scales, potentially characterized by different features according to the above introduced schema, or the overall model could even employ completely different styles in a hybrid approach (as discussed by [Dada and Mendes \(2011\)](#)): the different dynamics must then be properly coupled by means of some form of interaction among the different submodels and scales.

Actions and Interactions

The notion of *action* directly contributes to the definition of agent: literally, *the one who acts*. Understanding the reciprocal *dependencies* between individual agents and their surrounding environment – either physical or computational, therein including the space-time fabric – amounts to understanding which kind of actions produce which kind of *effects* – either intended or not – and on *whom* – either another agent or (a portion of) the environment.

The distinction regarding the kind of actions taken as reference throughout the article is made by focusing on the *purpose* of an action (Kirsh and Maglio, 1994):

epistemic actions are meant to acquire/release information, and may or may not have a direct practical effect, either intended or not *practical actions*, on the contrary, are meant to directly affect a subject, and may or may not have not a direct epistemic effect, either intended or not.

From this stems the distinction w.r.t. the kind of effects caused by actions:

epistemic effects directly cause the acquisition/release of information *practical effects* directly cause a change in a subject

As the reader may notice, epistemic actions may have practical effects (possibly, not intended and indirect), and practical actions may have epistemic effects in turn (again, possibly not intended and indirect). Despite how odd this may seem, the popular distinction categorizing actions as either *communicative* or *practical* has been proven to be misleading by Conte and Castelfranchi (1995), who argue that practical actions are in all respects also communicative when they have an intended (although possibly implicit) communicative effect.

Anyway, epistemic actions are *mostly* based on *communication*, be it explicit or not, thus they likely require (FIPA, 1996) – at the very least:

- A *content language*, that is, a language for “talking about things”.
- A set of *communicative acts*, that is, the acts through which “communication happens”.

Besides, fruitful communication among *computational* agents is likely to require the content language to be shared among participants in a “conversation”, to require the communicative acts to have a well-defined shared semantics, and the conversations to adhere to prescribed interaction protocols guaranteeing some desired properties.

As far as practical actions are concerned, they are *mostly* based on *practical behaviors*, and thus are likely to require:

- *Perception* (Russell and Norvig, 1995) of the acting agent’s surroundings for detecting the subject of the action;
- *Practical reasoning* grounded upon *bounded rationality* (Bratman, 1987), to plan the course of actions to undertake in order to achieve a goal while considering feasibility, expected utility, likelihood of success, and the cost of the actions themselves.

Regardless of the purpose and the effects of actions, it is apparent that actions are influenced by their subject – who is it, an agent or the environment? – as well as by their surroundings – where and when is the action taking place? Which properties influence the action’s outcome, feasibility, etc.? In other words, actions are *situated* w.r.t. their *context*, which brings us to the next subsection.

Situated (Inter)Action

Situatedness is the property of being immersed within an *environment* (Suchman, 1987), that is, the property of being potentially influenced and, in turn, potentially capable of affecting someone or something.

Actions are then situated by definition: regardless of whether they are practical or epistemic, they have a target, a source, happen at a given time (and possibly have a duration, and/or a delay), affect a given space (either virtual or physical), and cause some change (at least, if successful) either intended or not (“side effects”). Thus, agents too are situated: through actions, they can be regarded as being “active” at a given time, in a precise space, for a given observer (e.g., the target of the action), either because through actions they affected someone or something, or because their course of action has been influenced by someone or something.

It is worth noting that situatedness directly relates to the notions of *perception* and *practical reasoning*: perception is the tool by which agents and the environment become *aware* of their surroundings (Russell and Norvig, 1995), and thus are potentially influenced by the activities therein; practical reasoning is the tool by which agents (and an intelligent environment? (Mariani, 2016) can *deliberate* about how to affect something or someone (Bratman, 1987).

Inter-actions then, are situated too, following the same reasoning: each of the participants in the interaction is situated, then the (inter-)actions they carry out *reciprocally affecting* each other are situated too – both because they are actions, although (possibly) *communicative*, and because the acting participants are situated. Besides, interactions also may be situated because they are *mediated* by some means external to the participants: e.g., the environment. In this case, since the environment is situated due to its very nature, any interaction it enables and constrains may be regarded as situated as well, through the properties of the environment – the flow of time, the topology of space, the existence of resources, and properties with their own dynamics.

Whether or not (inter-)actions are mediated by the environment, they always represent a *social* relationship between the parties involved, which brings us to the next subsection.

Social (Inter)Action

Castelfranchi discusses how the complex and distributed dependencies within the agents in a MAS – mostly regarding goals (Dennett, 1971), delegation and trust (Castelfranchi and Falcone, 1998) – are fundamental to the definition of intelligence as a social construct (Castelfranchi *et al.*, 1993). In particular, the notion of *social action*, which is meant to reconcile individual cognitive processes and social coordination, provides a conceptual foundation which all MAS social issues (cooperation, collaboration, competition) can be grounded on.

Not by chance, in fact, one of the first relevant acts of the Foundation for Intelligent Physical Agents (FIPA) – a world-wide organization devoted to agent-based technology – was to define a reference semantics for the FIPA Agent Communication Language (FIPA-ACL), and to also define the semantics of social actions – in this case, intended as message exchanging – which is now the standard for agent-oriented middleware (Bellifemine *et al.*, 2001).

Along the same lines, the many different kinds of social relationships expressed by social interactions have been recognized as deserving a first-class abstraction in the process of MAS engineering, especially when it comes to governing the *space of interactions*, a task which originated a whole research thread, branded *coordination models and languages* (Ciancarini, 1996). Accordingly, the SODA methodology explicitly accounts for *societies of agents* (Omicini, 2001), in line with the A&A meta-model (Omicini *et al.*, 2008a) which adds the notion of *artefact* – in particular, coordination artefacts – as a means for agents to augment their capabilities, both practical and cognitive, and to structure the societies they live in – as well as the MAS environment.

Agents in Bioinformatics and Computational Biology

It is not the aim of this article (nor it is considered actually possible) to provide a compact but comprehensive review of the most notable applications of agent models and technologies to bioinformatics and systems biology. Nonetheless, it is reasonable and useful to give an idea of the *categories* or areas of these applications.

First of all, as also noted in a general resource describing the state of the art and providing perspectives on agent-based computing (Luck *et al.*, 2005), simulation represents an application context in which the notion of autonomous agents have become almost ubiquitous: An *et al.* (2009), for instance, present a review of agent-based modeling approaches to translational systems biology, but a plethora of applications to other relevant areas and approaches could also be cited. What it worth noting is that, very often, the notion of agent employed in these works takes a very different perspective from the one provided by the most widely accepted definitions of intelligent agents, being more focused on studying the resulting or emergent behavior of the local actions and inter- actions of relatively simple agents, than on defining and employing knowledge-level agents involved in complicated patterns of coordination.

The latter, however, has become instead quite relevant to the design and implementation of solutions for managing specific parts or the overall workflow of scientists working in the field, in a more general e-science perspective, also as a consequence of results from the subfield of Agent Oriented Software Engineering (see, e.g., Jennings, 2001). For instance, Miles (2006) presents an example of an application in which agent-based approaches have been employed for performing data curation in the bioinformatics area, whereas Bartocci *et al.* (2007) describe a web-based Workflow Management System for bioinformatics that employs an agent-based middleware.

Whereas the previous systems, although actually applied to the specific fields of systems biology and bioinformatics, essentially accomplish quite general tasks, tasks that are quite specific to these research fields can also be fruitfully supported by agent-oriented applications: for instance, Garro *et al.* (2004) and Armano *et al.* (2006) describe agent-based approaches to the prediction of protein structures, in which agents are associated with different predictors, and their results are combined to achieve an improved collective overall performance compared the performance of the individual predictors.

Closing Remarks

The domain of Intelligent Agents is growing at an exponential rate (Singh *et al.*, 2017). The basic concepts offer powerful solutions for designing a variety of agent-based solutions. Over the past three decades, significant advances have been made in the domain, specifically pertaining to agent interaction and communication in complex distributed systems. The technology is now finding acceptance in commercial applications as well as in complex and mission critical applications. Efforts to improve the technology are still going on, and many issues, such as defining the limits on trust, intelligence, and their applicability are yet open. The full potential of agents is yet to be realized.

See also: Ecosystem Monitoring Through Predictive Modeling. Epidemiology: A Review. Intelligent Agents: Multi-Agent Systems

References

- Adamatti, D.F., 2016. Multi-agent-based simulations applied to biological and environmental systems. Information Science Reference, Hershey, PA.
- Anderson, P., 2015. On the origins of that most transformative of biological systems – the nervous system. *Journal of Experimental Biology* 218, 504–505.
- An, G., Mi, Q., Dutta-Moscato, J., Vodovotz, Y., 2009. Agent-based models in translational systems biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 1, 159–171.
- Armano, G., Orro, A., Vargiu, E., 2006. MASSP3: A system for predicting protein secondary structure. *EURASIP J. Adv. Sig. Proc.* 2006.
- Bartocci, E., Corradini, F., Merelli, E., Scortichini, L., 2007. BioWMS: A web-based workflow management system for bioinformatics. *BMC Bioinformatics* 8.
- Bellifemine, F., Poggi, A., Rimassa, G., 2001. Jade: A FIPA2000 compliant agent development environment. In: *Proceedings of the Fifth International Conference on Autonomous Agents*, ACM, New York, NY, pp. 216–217.

- Bratman, M., 1987. *Intention, Plans, and Practical Reason*. Center for the Study of Language and Information.
- Castelfranchi, C., Cesta, A., Conte, R., Miceli, M., 1993. *Foundations for Interaction: The dependence Theory*. Berlin, Heidelberg: Springer, pp. 59–64.
- Castelfranchi, C., Falcone, R., 1998. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In: *Proceedings of the International Conference on Multi Agent Systems* (Cat. No.98EX160), pp. 72–79.
- Chen, D., Lu, Z., 2018. A methodological framework for model-based self-management of services and components in dependable cyber-physical systems. *Advances in Intelligent Systems and Computing* 582, 97–105.
- Ciancarini, P., 1996. Coordination models and languages as software integrators. *ACM Computing Surveys* 28, 300–302.
- Conte, R., Castelfranchi, C., 1995. *Cognitive and social action*. Psychology Press.
- Dada, J.O., Mendes, P., 2011. Multi-scale modelling and simulation in systems biology. *Integrative Biology* 3, 86–96.
- Dennett, D.C., 1971. *Intentional systems*. The Journal of Philosophy 68, 87–106.
- Ferber, J., 1999. *Multi-Agent Systems*. Addison-Wesley.
- Ferber, J., Müller, J.P., 1996. Influences and reaction: A model of situated multiagent systems. In: *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*, AAAI, pp. 72–79.
- FIPA, 1996. Foundation for intelligent physical agents, www.fipa.org/.
- Garro, A., Terracina, G., Ursino, D., 2004. A multi-agent system for supporting the prediction of protein structures. *Integrated Computer-Aided Engineering* 11, 259–280.
- Garro, A., Tundis, A., 2015. On the reliability analysis of systems and SoS: The RAMSAS method and related extensions. *IEEE Systems Journal* 9, 232–241.
- Hayes-Roth, B., 1996. An architecture for adaptive intelligent systems. *Artificial Intelligence: Special Issue on Agents and Interactivity* 72, 329365.
- Jennings, N.R., 2001. An agent-based approach for building complex software systems. *Communications of the ACM* 44, 35–41.
- Kirsh, D., Maglio, P., 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science* 18, 513–549.
- Luck, M., McBurney, P., Sheory, O., Willmott, S. (Eds.), 2005. *Agent Technology: Computing as Interaction: A Roadmap for Agent Based Computing*. University of Southampton.
- Magedanz, T., Rothermel, K., Krause, S., 1996. Intelligent agents: An emerging technology for next generation telecommunications. *IEEE Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer Societies on Networking the Next Generation INFOCOM96* 2, pp. 464–472.
- Mariani, S., 2016. *Coordination of complex sociotechnical systems: Self- organisation of knowledge in MoK*. Berlin, Heidelberg: Springer.
- Miles, S., 2006. Agent-oriented data curation in bioinformatics. *ITSSA* 1, 43–50.
- Omicini, A., 2001. *SODA: Societies and Infrastructures in the Analysis and Design of Agent-Based Systems*. Berlin, Heidelberg: Springer, pp. 185–193.
- Omicini, A., Ricci, A., Viroli, M., 2008a. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 17, 432–456.
- Russell, S., Norvig, P., 1995. *Artificial Intelligence: A modern approach*. Prentice-Hall.
- Sampigethaya, K., Poovendran, R., 2012. Cyber-physical integration in future aviation information systems. In: *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st*, IEEE, pp. 7C2–1.
- Seekhao, N., Shung, C., Jaja, J., Mongeau, L., Li-Jessen, N., 2016. Real- time agent-based modeling simulation with in-situ visualization of complex biological systems: A case study on vocal fold inflammation and healing, *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International*, pp. 463–472.
- Singh, A., Juneja, D., Singh, R., Mukherjee, S., 2017. A thorough insight into theoretical and practical developments in multiagent systems. *International Journal of Ambient Computing and Intelligence* 8, 23–49.
- Smith, D.C., Cypher, A., Spohrer, J., 1994. KidSim: Programming agents without a programming language. *Commun. ACM* 37, 54–67.
- Suchman, L.A., 1987. *Situated actions. Plans and Situated Actions: The Problem of Human-Machine Communication*. New York, NYU, USA: Cambridge University Press, pp. 49–67. [chapter 4].
- Weyns, D., Omicini, A., Odell, J., 2007. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems* 14, 5–30.
- Zhabelova, G., Vyatkin, V., 2012. Multiagent smart grid automation architecture based on IEC 61850/61499 intelligent logical nodes. *IEEE Transactions on Industrial Electronics* 59, 2351–2362.

Intelligent Agents: Multi-Agent Systems

Alfredo Garro, University of Calabria, Rende, Italy

Max Mühlhäuser and Andrea Tundis, Darmstadt University of Technology, Darmstadt, Germany

Matteo Baldoni and Cristina Baroglio, University of Turin, Turin, Italy

Federico Bergenti, University of Parma, Parma, Italy

Paolo Torroni, University of Bologna, Bologna, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The agent is a metaphor that was natively originated in Artificial Intelligence (AI) (see works from Newell, and others). Its characteristics changed when the emphasis was put on multi-agent systems, rather than on single agents. A key event in the history of multi-agent systems was the recognition that agents can be used fruitfully to model and implement distributed systems. The creation of Foundation for Physical Intelligent Agents (FIPA) originates from the recognition that multi-agent systems are a novel and promising approach for the implementation and deployment of distributed systems. It is important to recognize that many applications of agents exist outside AI, because tools and techniques, which are intended to support one, are not necessarily good for the other. In particular, nowadays, multi-agent systems have two major applications outside of their traditional role in AI:

- Engineering of distributed systems, with the introduction of agent-oriented software engineering (AOSE) (e.g., agent-oriented methodologies, agent-oriented programming languages).
- Agent-based modeling and simulation (ABMS), which is what this article is mostly about.

Agents are entities that observe their environment and act upon it so as to achieve their own goals ([Russell and Norvig, 2003](#); [Wooldridge, 2009](#)). Two fundamental characteristics of agents are *autonomy* and *situatedness*. Autonomy means that agents have a sense-plan-act deliberative cycle, which gives them control of their internal state and behavior. Whereas, agents are situated because they can sense, perceive, and manipulate the environment in which they operate. The environment could be physical or virtual and it is understood by agents in terms of (relevant) data. Autonomy implies proactivity, i.e., the ability of an agent to take action toward the achievement of its objectives, without being solicited to do so.

From a programming perspective, agent-oriented programming was introduced by Shoham as “a specialization of *object-oriented programming*” ([Shoham, 1993](#)). The difference between agents and static objects is clear. Citing Wooldridge ([Wooldridge, 2009](#), Section 2.2): (1) objects do not have control over their own behavior (this is summarized by the well-known motto “Objects do it for free; agents do it because they want it”), (2) objects do not exhibit flexibility in their behavior, and (3) in standard object models there is a single thread of control, while agents are inherently multi-threaded.

The agent-based paradigm also differs from the Actor Model ([Hewitt et al., 1973](#)) (and from Active Objects, largely inspired by the latter). Actors, in fact, do not have neither goals nor purposes, even though their specification includes a process. Agents, instead, exploit their deliberative cycle (as control flow), possibly together with the key abstractions of belief, desire, and intention (as logic), so as to realize algorithms, for example, processes for acting in their environment to pursue their goals. In other words objects “do it” for free because they are data, agents are processes and “do it” because it is functional to their objectives.

The environment, in which agents are situated, does not exhibit the kind of autonomy that is typical of agents although it may evolve, also thanks to an internal process. Its activity, however, is not meant to pursue a goal and this makes environments more similar to active objects.

The binomial agent–environment is formalized by modeling approaches like ([Demazeau, 1995](#)), where the environment is seen as providing “the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources”, and in particular by the Agent & Artifact meta-model ([Omicini et al., 2008](#)).

A system comprising a number of possibly interacting agents is called a *multiagent system*. At this level, it is widely recognized that further abstractions become handy, like organizations and interactions, aimed at enabling a meaningful and fruitful coordination of the autonomous and heterogeneous agents in the system. Thus, agents are not only situated in a physical environment, they are also situated in a social environment where they get into relationships with other agents and are subject to the regulations of the society they belong to. A normative multiagent system is “a multiagent system together with normative systems in which agents on the one hand can decide whether to follow the explicitly represented norms, and on the other the normative systems specify how and in which extent the agents can modify the norms” ([Boella et al., 2007](#)). The impact on the agent’s deliberative cycle is that agents can reason about the social consequences of their actions.

Multi-Agent Systems

There is no single definition for the word agent, and there is no single definition for the term multi-agent system (MAS). Notably, major accepted definitions share commonalities, such as the way the agents interact in a system: via the shared environment, via structured messages (ontologies, interaction protocols). Indeed, a MAS can be defined in terms of interacting entities, and in

particular the agents. Communication may vary from simple forms to sophisticated ones. A simple form of communication is that restricted to simple signals, with fixed interpretations. Such an approach was used by Georgeff in multi-agent planning to avoid conflicts when a plan was synthesized by several agents. A more elaborate form of communication is by means of a blackboard structure. A blackboard is a shared resource, usually divided into several areas, according to different types of knowledge or different levels of abstraction in problem solving, in which agents may read or write the corresponding relevant information for their actions. Another form of communication is by message passing between agents.

Autonomy is another major characteristic of agents, when defining a MAS, also referred to as “self-organized systems”, enabling them to find the best solution for their problems “without intervention”. The main feature which is achieved when developing multi-agent systems, is flexibility, since a multi-agent system can be added to, modified and reconstructed, without the need for detailed rewriting of the application. The MAS also tends to prevent propagation of faults, self-recover and be fault tolerant, mainly due to the redundancy of components.

It is extremely important to distinguish between Automatic and Autonomous systems. Automatic systems are fully pre-programmed and act repeatedly and independently of external influence or control. It can be described as selfsteering or self-regulating and it is able to follow an externally given path while compensating for small deviations caused by external disturbances. However, it is not able to define the path according to some given goal or to choose the goal dictating its path. Whereas, Autonomous systems, as a MAS, are selfdirected toward a goal in that they do not require outside control, but rather they are governed through laws and strategies that clearly make a difference between traditional and multi agent systems. If machine learning techniques are utilized, autonomous systems can develop flexible strategies for themselves by which they select their behavior.

More in detail, *norms* are a fundamental ingredient of multi-agent systems that govern the expected behavior toward a specific situation. Through the norms, the desirable behaviors for a population of a natural or artificial community is represented. Indeed, they are generally understood as rules indicating actions that are expected to be pursued that are either obligatory, prohibitive, or permissive based on a specific set of facts. According to [Hollander and Wu \(2011\)](#), norms have been used to indicate constraints on behavior ([Shoham and Tennenholtz, 1992](#)), to create solutions to a macrolevel problem ([Zhang and Leezer, 2009](#)), and to serve as obligatory ([Verhagen, 2000](#)), regulatory, or control devices for decentralized systems ([Savarimuthu et al., 2008](#)). The most common norms are:

- Conventions, which are natural norms that emerge without any enforcement ([Villatoro, 2011](#)). Conventions solve coordination problems when there is no conflict between the individual and the collective interests; for example, everyone conforms to desired behavior. Essential Norms are used to solve or ease collective action problems when there is a conflict between an individual and the collective interests ([Villatoro, 2011](#); [Villatoro et al., 2010](#)). For example, the norm not to pollute urban streets is essential in that it requires individuals to transport their trash, rather than dispose of it on the spot, an act that benefits everyone.
- Regulative Norms. Regulative norms are intended for regulating activities by imposing obligation or prohibition in performing an action.
- Constitutive Norms, which are affirmed to produce new goal norms or states of affairs, for example, the rules of a game like chess.
- Procedural Norms that are categorized as objective and subjective. Objective procedural norms represent the rules that express how decisions are really made in a normative system, while subjective procedural norms represent the instrument for individuals working in a system, for instance, back-office procedures.

Coordination is another distinguishing factor of a MAS. In fact, an agent exists and performs its activity in a society in which other agents exist. Therefore, coordination among agents is essential for achieving the goals and acting in a coherent manner. Coordination implies considering the actions of the other agents in the system when planning and executing one agents actions. Coordination allows agents to achieve the coherent behavior of the entire system. Coordination may imply cooperation and in this case the agent society works toward common goals to be achieved, but may also imply competition, with agents having divergent or even antagonistic goals. In this later case, coordination is important because the agent must take into account the actions of the others, for example, competing for a given resource or offering the same service.

Another characterizing feature of a MAS is its *emergent behavior*. Emergent behavior in agents is commonly defined as behavior that is not attributed to any individual agent, but is a global outcome of agent coordination ([Zhengping et al., 2007](#)). This definition emphasizes that emergent behavior is a collective behavior. There are also other definitions. Emergent behavior is that which cannot be predicted through analysis at any level simpler than that of the system as a whole Emergent behavior, by definition, is what’s left after everything else has been explained ([Dyson, 1997](#)). This definition highlights the difficulty in predicting and explaining emergent behavior. If the behavior is predictable and explainable, then it will not be treated as emergent behavior and approaches could be designed to handle the behaviors. Emergence is also defined as the action of simple rules combining to produce complex results ([Rollings, 2003](#)). This definition states that the rules applied to the individuals can be quite simple, but the collective behavior of the group may turn out to be quite complex and unpredictable. Researchers have designed experiments to demonstrate this kind of situation. While it is true that all behavior comes from individuals, the interactions are what make things difficult to understand. Emergent behavior is essentially any behavior of a system that is not a property of any of the components of that system, and emerges due to interactions among the components of a system. Borrowing from biological models such as an ant colony, emergent behavior can also be thought of as the production of high level or complex behaviors through the interaction of multiple simple entities. Some

examples of emergent behaviors: Bee colony behavior where the collective harvesting of nectar is optimized through the waggle dance of individual worker bees; Flocking of birds cannot be described by the behavior of individual birds; Market crashes cannot be explained by "summing up" the behavior of individual investors.

Further details about multi-agent systems can be founded in Baldoni *et al.* (2010).

Agent-based Modeling and Simulation

Agent Based Modeling and Simulation (ABMS) refers to a category of computational models invoking the dynamic actions, reactions and intercommunication protocols among the agents in a shared environment, in order to evaluate their design and performance and derive insights on their emerging behavior and properties (Abar *et al.*, 2017). Agents and multi-agent systems are entities that can be effectively used to model complex systems made of interacting entities. This is why they have been adopted to study biological and chemical systems, especially when the systems become too complex for the analytic tools available from Chemistry, Physics and Mathematical Physics. The fact that agents and multi-agent systems are abstractions with executable counterparts, the agents and the multi-agent systems that many tools support, contributed to suggest the use of agent technology to simulate biological and chemical systems. The size of simulated systems, and the high level of accuracy of simulated phenomena, calls for dedicated tools capable enabling the domain expert to describe simulations with little, or no, interest on the engineering issues related to distributed systems. Agent-based technology provides such tools and it supports domain experts in the construction of effective distributed systems with minimal emphasis on the inherent issues of large-scale distributed systems. Notably, even if dedicated tools are available, it is common to adopt tools designed to support agent-oriented software engineering in the scope of ABMS. In particular, a number of agent-based tools to support modeling and simulation of complex and/or distributed systems are available (Allan, 2009):

1. Netlogo: it is a multi-agent programmable modeling environment which allows the simulation of natural and social phenomena. It is particularly well suited for modeling complex systems developing over time. Indeed, modelers can give instructions to hundreds or thousands of "agents" all operating independently. This makes possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from their interaction. It comes with a large library of existing simulations, both participatory and traditional, that one can use and modify in different domains such as social science and economics, biology and medicine, physics and chemistry, and mathematics and computer science. In the traditional NetLogo simulations, the simulation runs according to rules that the simulation author specifies. A further feature of NetLogo is HubNet, a technology that lets you use NetLogo to run participatory simulations. HubNet adds a new dimension to NetLogo by letting simulations run not just according to rules, but by direct human participation.
2. FLAME: it is a generic agent-based modeling system which can be used to development applications in many areas. Models are created based upon a model of computation called (extended finite) state machines. The framework can automatically generate simulation programs that can run models efficiently on HPCs. It produces a complete agent-based application which can be compiled and built on the majority of computing systems ranging from laptops to HPC super computers. Furthermore, FLAME provides a Model Library which is a collection of relatively simple models that illustrate the use of FLAME in different applications.
3. AnyLogic: it is a simulation tool that supports all the most common simulation methodologies in place today: System Dynamics, Process-centric (AKA Discrete Event), and Agent Based modeling. Its visual development environment significantly speeds up the development process. It has applicational models in ManufacturingLogistics, Supply ChainsMarkets, Competition Business Processes Modeling Healthcare, Pharmaceuticals Simulation Pedestrian Traffic Flows Information, Telecommunication Networks Simulation Modeling Social Process, Marketing Simulation Asset Management, Financial Operations with Simulation Modeling Warehouse Operations and Layout Optimization.
4. Repast: The Repast Suite is a family of advanced, free, and open source agent-based modeling and simulation platforms that have collectively been under continuous development for many years. Repast Symphony is a richly interactive and easy to learn Java-based modeling system that is designed for use on workstations and small computing clusters. An advanced version is called Repast for High Performance Computing, which is a lean and expert-focused C++-based modeling system, that is designed for use on large computing clusters and supercomputers.
5. Jason: It is an interpreter for an extended version of AgentSpeak, that has been one of the most influential abstract languages based on the BDI architecture. Jason implements the operational semantics of that language, Strong negation, so both closed-world assumption and open-world are available. Annotations in beliefs are used for meta-level information and annotations in plan labels. One of the best known approaches for the development of cognitive agents is the BDI (Beliefs-Desires-Intentions) architecture. It provides the possibility to run a multi-agent system distributed over a network.
6. Framsticks: it is a three-dimensional life simulation project. Both mechanical structures (bodies) and control systems (brains) of creatures are modeled. It is possible to design various kinds of experiments, including simple optimization, co-evolution, open-ended and spontaneous evolution, distinct gene pools and populations and modeling of species and ecosystems. Users of this software work on evolutionary computation, artificial intelligence, neural networks, biology, robotics and simulation, cognitive science, neuro-science, medicine, philosophy, virtual reality, graphics, and art.
7. Gephi: it is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. It allows (1) Exploratory Data Analysis: intuition-oriented analysis by networks manipulations in real time;

- (2) Link Analysis: revealing the underlying structures of associations between objects, in particular in scale-free networks; (3) Social Network Analysis: easy creation of social data connectors to map community organizations and small-world networks; (4) Biological Network analysis: representing patterns of biological data; (5) Poster Creation: scientific work promotion with hi-quality printable maps. It works mainly with metrics related to centrality, degree (power-law), betweenness, closeness, density, path length, diameter, HITS, modularity, clustering coefficient.
- 8. Stanford Network Analysis Platform (SNAP): it is a general purpose, high performance system for analysis and manipulation of large networks. It easily scales to massive networks with hundreds of millions of nodes, and billions of edges. It efficiently manipulates large graphs, calculates structural properties, generates regular and random graphs, and supports attributes on nodes and edges.

This is a list of the most popular simulation tools that can be used to model, simulate and analyzes complex systems by adopting agent oriented paradigm.

Agent-Oriented Software Engineering

Since its early days in late 1960s, *software engineering* has been constantly facing the problem of better understanding the sources of complexity in software systems. Over the last three decades, the complexity of interactions among parts has been progressively identified as one of the most significant sources of such a complexity. Software systems that contain a – possibly large – number of interacting parts are critical, especially when the graph of interactions changes dynamically, parts have their own thread of control, and parts are engaged in interactions governed by complex protocols (see, e.g., [Wooldridge and Ciancarini \(2001\)](#) for an in-depth discussion). As a consequence, a major research topic of software engineering has been the development of techniques and tools to understand, model, and implement systems in which interactions is the major source of complexity. This has led to the search for new computational abstractions, models, and tools to reason and to implement MASs, which have been recognized as prototypical examples of such systems. *Agent-Oriented Software Engineering* (AOSE) is an emerging paradigm of software engineering that has been developed to target the inherent complexity of analysing and implementing MASs (see, e.g., [Bergenti et al. \(2004\)](#) for a comprehensive reference on the subject). A number of AOSE methodologies have been proposed over the last two decades, and ([Kardas, 2013](#)) provides a recent survey of the state of the art of such methodologies.

Besides methodologies, the research on AOSE have been constantly interested in delivering effective tools to implement MASs. *Agent platforms* are examples of such tools intended to offer generic runtime environments for the effective deployment and execution of MASs. A number of platforms have been proposed over the years, and ([Kravari and Bassiliades, 2015](#)) proposes a recent attempt to enumerate the platforms that survived the test of time. One of the most popular agent platforms is *Java Agent DEvelopment framework* (JADE), as described in [Bellifemine et al. \(2007\)](#), which consists of a middleware and a set of tools that help the development of distributed, large-scale MASs. JADE is widely used for industrial and academic purposes and it can be considered as a consolidated tool. Just to cite a notable industrial example, JADE has been in daily use for service provision and management in Telecom Italia for more than six years, serving millions of customers in one of the largest and most penetrating broadband networks in Europe (see [Bergenti et al. \(2015\)](#) for further details). In order to effectively address the inherent issues of the high-profile scenarios that agent platforms target, specific tools are needed to assist the development of complex functionality, and to promote the effective use of the beneficial features of agent technology as a software development technology. Nevertheless, approaching AOSE with the help of agent platforms alone is often perceived as a difficult task for two main reasons. First, the continuous growth of agent platforms has been increasing their inherent complexity, and the number of implementation details that the programmer is demanded to master for the construction of MASs is equally grown. Second, the choice of mainstream programming languages as the unique option to use agent platforms is now considered inappropriate in many situations because such languages do not natively offer the needed abstractions for the effective concretization of AOSE.

The interest in *agent programming languages* dates back to the introduction of agent technologies and, since then, it has grown rapidly. As a matter of fact, agent programming languages turned out to be especially convenient to model and develop complex MASs. Nowadays, agent programming languages represent an important topic of research and they are widely recognized as important tools in the development of agent technologies, in contrast with mainstream languages, that are often considered not suitable to effectively implement AOSE. Agent programming languages are usually based on specific agent models and they aim at providing specific constructs to adopt such models at a high level of abstraction. The features of the various agent programming languages proposed over the years may differ significantly, concerning, for example, the selected agent mental attitudes (if any), the integration with an agent platform (if any), the underlying programming paradigm, and the underlying implementation language. Some classifications of relevant agent programming languages have already been proposed to compare the characteristics of different languages and to provide a clear overview of the state of the art. [Bădică et al. \(2011\)](#) classifies agent programming languages on the basis of the use of mental attitudes. According to such a classification, agent programming languages can be divided into: *Agent-Oriented Programming* (AOP) languages, *Belief Desire Intentions* (BDI) languages, hybrid languages – which combine the two previous classes – and other languages – which fall outside previous classes. It is worth noting that such a classification recognizes that BDI languages follow the *agent-oriented programming* paradigm, as defined in [Shoham \(1993\)](#), but it reserves special attention to them for their notable relevance in the literature. [Bordini et al. \(2006\)](#) proposes a different classification, where languages are divided into declarative, imperative, and hybrid. Declarative languages are the most common because they focus on automatic reasoning, both from the AOP and from the BDI points of view. Some relevant imperative languages have

also been proposed, and most of them were obtained by adding specific constructs to existing procedural programming languages. Finally, the presence (or absence) of a host language is an important basis of comparison among agent programming languages.

Even if early proposals dates back to late 1990s, AOSE is still at an early stage of evolution. While there are many good arguments to support the view that agents represent an important direction for software engineering, there is still need of actual experience to underpin these arguments. Methodologies and tools to support the deployment of MAS are beginning to become accepted for mission-critical applications, but slowly. Although a number of agent-oriented analysis and design methodologies have been proposed, there is comparatively little consensus among them. In most cases, there is not even agreement on the kinds of concepts the methodology should support. But, the research on AOSE is still active, and it has been recently revitalized by the view of AOSE in terms of *model-driven development*, as summarized in Kardas (2013).

Guidelines/Perspectives

Agents and multi-agent systems have been used to study and to simulate complex systems in different application domain where physical factor are present for energy minimizing, where physical objects tend to reach the lowest energy consumption possible within the physically constrained world. Furthermore, MAS have been intensively exploited for the analysis through simulation of biological and chemical systems. The literature reports on various successful uses of the abstractions and of their executable tools. Notably, the study of the benefits and the costs of adopting agents and multi-agent systems to support scientific studies of biological and chemical systems has not been approached extensively.

A well-known application field, where agents and MAS have been successful exploited regarding the study of biological phenomena, is the protein synthesis, a common and relevant phenomenon in nature. In this field, several approaches for predicting the three-dimensional structure of proteins are, for instance, available in literature Jennings *et al.* (1998). Several of them exploit the chemical or physical properties of the proteins (e.g., Standley *et al.* (1998), Dudek *et al.* (1998) work on energy minimization whereas Galaktionov and Marshall (1995), Sabzekar *et al.* (2017) uses intra-globular contacts); other ones use evolutionary information (Piccolbon and Mauri, 1998). Some tools for the prediction of the three-dimensional structure of proteins have been presented in Douguet and Labesse (2001), Gough *et al.* (2001), Meller and Elber (2001).

This shows that the autonomy of agents, their normed freedom of interaction, and the possibility of deploying large-scale systems with minimal care on issues related to distributed systems are few of the major benefits of approaching modeling and simulation of complex systems with agents. Furthermore, they closely represent how natural systems work by distributing a problem among a number of reactive, autonomous, deliberative, pro-active, adaptive, possibly mobile, flexible and collaborative entities. All these properties make agent technology more suited than other ones (e.g., a distributed system, an artificial intelligent system) for facing the challenges and high level of complexity of biological system and related phenomena.

See also: Computational Systems Biology Applications. Intelligent Agents and Environment. Studies of Body Systems

References

- Abar, S., Theodoropoulos, G.K., Lemariniér, P., OHare, G.M., 2017. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review* 24, 13–33.
- Allan, R., 2009. Survey of agent based modelling and simulation tools.
- Bădică, C., Budimac, Z., Burkhard, H.D., Ivanovic, M., 2011. Software agents: Languages, tools, platforms. *Computer Science and Information Systems* 8, 255–298.
- Baldoni, M., Baroglio, C., Mascardi, V., Omicini, A., Torroni, P., 2010. Agents, multi-agent systems and declarative programming: What, when, where, why, who, how? In: Dovier, A., Pontelli, E. (Eds.), *A 25-Year Perspective on Logic Programming: Achievements of the Italian Association for Logic Programming, GULP*. Springer, pp. 204–230.
- Bellifemine, F., Caire, G., Greenwood, D., 2007. *Developing multi-agent systems with JADE*. Wiley Series in Agent Technology. John Wiley & Sons.
- Bergenti, F., Caire, G., Gotta, D., 2015. Large-scale network and service management with WANTS. *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier. pp. 231–246.
- Bergenti, F., Gleizes, M.P., Zambonelli, F. (Eds.), 2004. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*. Springer.
- Boella, G., van der Torre, L.W.N., Verhagen, H., 2007. Introduction to normative multiagent systems. In: Boella, G., van der Torre, L.W.N., Verhagen, H. (Eds.), *Normative Multi-agent Systems*, 18–23, March 2007, Internationales Begegnung- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
- Bordini, R.H., Braubach, L., Dastani, M., *et al.*, 2006. A survey of programming languages and platforms for multi-agent systems. *Informatica* 30.
- Demazeau, Y., 1995. From interactions to collective behaviour in agent-based systems. In: *Proceedings of the 1st European Conference on Cognitive Science*, Saint-Malo, pp. 117–132.
- Douguet, D., Labesse, G., 2001. Easier threading through web-based comparisons and cross-validations. *Bioinformatics* 17, 752–753.
- Dudek, M., Ramnarayan, K., Ponder, J., 1998. Protein structure prediction using a combination of sequence homology and global energy minimization: II. Energy functions. *Journal of Computational Chemistry* 19, 548–573.
- Dyson, G.B., 1997. *Darwin Among the Machines: The Evolution of Global Intelligence*. Addison-Wesley Publishing Company.
- Galaktionov, S.G., Marshall, G.R., 1995. Properties of intraglobular contacts in proteins: An approach to prediction of tertiary structure, pp. 326–335.
- Gough, J., Karplus, K., Hughey, R., Chothia, C., 2001. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. *Journal of Molecular Biology* 313, 903–919.
- Hewitt, C., Bishop, P., Steiger, R., 1973. A universal modular ACTOR formalism for artificial intelligence. In: Nilsson, N.J. (Ed.), *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, Stanford, CA, August 1973, William Kaufmann, pp. 235–245.

- Hollander, C.D., Wu, A.S., 2011. The current state of normative agent-based systems. *J. Artificial Societies and Social Simulation* 14.
- Jennings, N., Sycara, K., Wooldridge, M., 1998. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1, 7–38.
- Kardas, G., 2013. Model-driven development of multiagent systems: A survey and evaluation. *The Knowledge Engineering Review* 28, 479–503.
- Kravari, K., Bassiliades, N., 2015. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation* 18, 11.
- Meller, J., Elber, R., 2001. Linear programming optimization and a double statistical filter for protein threading protocols. *Proteins: Structure, Function and Genetics* 45, 241–261.
- Omicini, A., Ricci, A., Viroli, M., 2008. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 17, 432–456. [Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems].
- Piccolbon, A., Mauri, G., 1998. Application of evolutionary algorithms to protein folding prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 1363, 123–135.
- Rollings, A., Adams, E., 2003. Andrew Rollings and Ernest Adams on Game Design.
- Russell, S.J., Norvig, P., 2003. *Artificial Intelligence: A Modern Approach*, second ed. Pearson Education.
- Sabzekar, M., Naghibzadeh, M., Eghdami, M., Aydin, Z., 2017. Protein-sheet prediction using an efficient dynamic programming algorithm. *Computational Biology and Chemistry* 70, 142–155.
- Savarimuthu, B., Purvis, M., Purvis, M., 2008. Social norm emergence in virtual agent societies, pp. 1485–1488.
- Shoham, Y., 1993. Agent-oriented programming. *Artificial Intelligence* 60, 51–92.
- Shoham, Y., Tennenholtz, M., 1992. On the synthesis of useful social laws for artificial agent societies (preliminary report), pp. 276–281.
- Standley, D., Gunn, J., Friesner, R., McDermott, A., 1998. Tertiary structure prediction of mixed/proteins via energy minimization. *Proteins: Structure, Function and Genetics* 33, 240–252.
- Verhagen, H., 2000. Norm autonomous agents.
- Villatoro, D., 2011. Self-organization in decentralized agent societies through social norms, pp. 1297–1298.
- Villatoro, D., Sen, S., Sabater-Mir, J., 2010. Of social norms and sanctioning: A game theoretical overview. *International Journal of Agent Technologies and Systems (IJATS)* 2, 115.
- Wooldridge, M., Ciancarini, P., 2001. Agent-oriented software engineering: The state of the art. *Agent-Oriented Software Engineering*. Springer Verlag. pp. 1–28.
- Wooldridge, M.J., 2009. *Introduction to multiagent systems*, second ed. Wiley.
- Zhang, Y., Leezer, J., 2009. Emergence of social norms in complex networks, pp. 549–555.
- Zhengping, L., Cheng, H., Malcolm, Y., 2007. A survey of emergent behavior and its impacts in agent-based systems, pp. 1295–1300.

Stochastic Methods for Global Optimization and Problem Solving

Giovanni Stracquadanio, University of Essex, Colchester, United Kingdom

Panos M. Pardalos, University of Florida, Gainesville, FL, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Many engineering and scientific problems require finding a solution that is optimal with respect to a predefined metric. Identifying an optimal solution with a mathematical guarantee is only possible when the Karush-Kuhn-Tucker (KKT) conditions are met (Horst and Pardalos, 2013). However, this is not always possible when gradient information is not available, the metric is difficult to compute, or the mathematical formulation of a problem is not available (e.g., simulators). In this case, we do not look for solutions with guaranteed optimality, instead we accept satisfactory solutions that are only putative or locally optimal. This is a common approach for solving optimization problems in science and engineering, and a plethora of algorithms have been proposed in the literature, both deterministic and stochastic. However, while deterministic methods can converge to a global optimum when certain conditions are met, stochastic methods are faster and more effective in practice, although they have weak convergence properties.

In this article, we present two stochastic optimization methods, namely Simulated Annealing (SA, Kirkpatrick *et al.*, 1983) and Genetic Algorithms (GA, Holland, 1975), which do not require gradient information to find putative optimal solutions to an arbitrary optimization problem and are straightforward to implement. The article is structured as follows; in Section Global Optimization, we give a brief overview of the fundamentals of global optimization; in Section Stochastic Optimization Algorithms, we outline the basic design principles of stochastic optimization methods, while in Sections Simulated Annealing and Genetic Algorithms we present the Simulated Annealing (SA) and the Genetic Algorithms (GAs), respectively. Finally, in Section Closing Remarks, we provide a brief comparative analysis of the two methods.

Global Optimization

Global optimization is the task of finding the set of parameters, $\mathbf{x}$, associated with the optimal value of an objective function, f , under a set of constraints, g . While local optimization looks for the optimal solution in a subregion of the feasible space, Ω , global optimization seeks an optimal solution in the entire feasible space. The change in scope between local and global optimization has tremendous theoretical consequences. Indeed, the $P \neq NP$ conjecture implies that there are no general algorithms to solve a global optimization problem in time polynomial in the problem description length (Neumaier, 2004).

Despite such strong theoretical limitations, in many engineering and scientific problems, finding the global optimal solution is not necessary; in practice, it is often sufficient to find a solution that improves the best known result. This is especially true for optimization problems where the objective and constraint functions are the output of simulators, when estimating the gradient is not possible, or numerical methods are not stable (Conn *et al.*, 2009). These problems belong to the class of black-box optimization problems, and many derivative-free algorithms have been proposed for this class of problems. Black-box functions are ubiquitous in science and engineering; for instance, in electronics, simulators that estimate the behavior of a circuit are treated as black-box functions, whereas in biology, an experiment is often modeled as a black-box function due to the lack of descriptive mathematical models.

We adopt a general formulation of an optimization problem subject to box, equality and inequality constraints, which tries to unify the canonical form with the extended formulation of black-box optimization introduced in Audet *et al.* (2010). Without loss of generality, we assume the problems are formalized as global minimization problems.

Let f be a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$. A global optimization problem is defined as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & f(\mathbf{x}) \\ g_k(\mathbf{x}) \quad & \leq \quad 0 & k = 1, \dots, l \\ h_z(\mathbf{x}) \quad & = \quad 0 & z = 1, \dots, m \\ \mathbf{x} \quad & \in \quad \mathbf{B} \text{ s.t. } \{b_i^- \leq x_i \leq b_i^+ | i = 1, \dots, n\} \end{aligned} \tag{1}$$

where n is the size of the problem, and the i -th variable, x_i , is bounded by b_i^- , b_i^+ , which are the lower and upper bound, respectively. $g_k: \mathbb{R}^n \rightarrow \mathbb{R}$ denotes a set of inequality constraints of size l , whereas $h_z: \mathbb{R}^n \rightarrow \mathbb{R}$ is a set of equality constraints of size m . Ω denotes the feasible region; if $\mathbf{x} \in \Omega$, then $\mathbf{x} \in \mathbf{B}$ and $g(\mathbf{x}) \leq 0$, $h(\mathbf{x}) = 0$ for all constraints. In general, each constraint is reformulated either in terms of equality or inequality to zero; this is a simple and convenient reformulation, which does not change the feasible region Ω .

Our definition accounts for different types of variables, including free variables ($x_i \in [-\infty, \infty]$), nonnegative variables ($x_i \in [0, \infty]$), and binary variables ($x_i \in [0, 1]$).

The constraint functions are usually classified as relaxable or unrelaxable; unrelaxable constraints are those constraints that a solution must not violate to be feasible. Conversely, relaxable constraints represent the degree of constraint violation of a solution. Relaxable constraints are often added as penalty terms to the objective function. An alternative approach is to reformulate the problem as a multi-objective optimization problem, where the relaxable constraints are modeled as additional objective functions. An additional class of constraints is the class of hidden constraints; this class includes those solutions, $\mathbf{x}$, in the feasible region, Ω , for which the objective function cannot be computed. This is a common scenario when the output is a simulator, which might fail to evaluate a solution despite its feasibility. If the problem defines a set of constraints and bounds s.t. $\mathbf{x} \notin \Omega$ for all $\mathbf{x}$, then the problem is deemed unsolvable and $\Omega = \emptyset$.

In our formulation, we assume that our problem is a single objective optimization problem, thus the objective function returns a single output value. A similar framework can be defined for multi-objective optimization problems, but it is outside the scope of this article. An extensive review on multi-objective optimization algorithms has been published by [Coello et al. \(2007\)](#).

Stochastic Optimization Algorithms

Although black-box global optimization problems cannot be solved exactly, several algorithms have been developed to identify satisfactory solutions to many practical problems.

Global optimization algorithms can be divided into two main classes ([Neumaier, 2004](#)); complete and incomplete search algorithms. Complete search methods can rigorously find all solutions to a global optimization problem if there are finitely many. Conversely, incomplete search algorithms are heuristic methods able to locate a putative optimal solution using heuristics. We will focus on incomplete search methods; an extensive introduction to complete search algorithms is presented in [Neumaier \(2004\)](#).

Heuristic global optimization methods include deterministic and stochastic algorithms; while the former comes with a theoretical convergence guarantee when the objective function meets specific mathematical conditions, the latter exploits stochastic sampling strategies to identify a putative optimal solution. It is important to point out that deterministic methods do not guarantee the identification of the global optimum, and the mathematical conditions for convergence can be rarely verified for black-box functions. An in depth review of deterministic global optimization methods is presented in [Floudas and Pardalos \(2008\)](#).

A plethora of stochastic methods have been proposed for global optimization, mostly based on two frameworks; Monte Carlo sampling and population-based sampling. Two algorithms have found applications in many engineering and scientific problems: Simulated Annealing (SA, [Kirkpatrick et al., 1983](#)) and Genetic Algorithms (GA, [Holland, 1975](#)). In the last 40 years, these methods have proven to be successful in identifying solutions to many optimization problems [Mavridou and Pardalos \(1997\)](#) where complete search methods are simply not feasible or deterministic algorithms perform poorly. SA and GA should be considered as two families of algorithms rather than distinct methods; indeed, although the underlying ideas have remained basically unaltered, there are several problem-specific implementations for both algorithms.

In the next sections, we describe in detail these two classes of algorithms and some of the most effective implementations.

Simulated Annealing

Simulated Annealing (SA, [Kirkpatrick et al., 1983](#)) is a global optimization method belonging to the class of Monte Carlo samplers. SA is able to locate a putative optimal solution of a function, f , by alternating a random sampling step with a probabilistic selection strategy. Specifically, SA is a Monte Carlo importance sampling technique, which was initially used to solve large scale physics integrals ([Metropolis et al., 1953](#)), and later in combinatorial optimization by [Kirkpatrick et al. \(1983\)](#). In their seminal paper, [Kirkpatrick et al. \(1983\)](#) also showed how statistical mechanics can be exploited to find satisfactory solutions to complex optimization problems. In this section, we describe the basic SA algorithmic framework and how to use statistical mechanics to study the behavior of this method.

Let f be a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$. SA alternates a sampling and selection step as described in [Alg. 1](#).

The algorithm starts by initializing a candidate solution, $\mathbf{x}$, at random ([Alg. 1](#), line 1); this initial candidate solution can be replaced by providing an initial solution obtained from domain experts or other optimization methods. SA then picks a new candidate solution, $\hat{\mathbf{x}}$, at random from the neighborhood of the current optimum, $\mathbf{x}$. To do that, SA uses a generating function $N(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$; in the case of continuous optimization, this is typically a multi-variate probability distribution [Alizamir et al. \(2008\)](#). As for standard Monte Carlo sampling methods, the generating function is selected such that the ergodic hypothesis is verified, therefore ensuring all possible solutions are equiprobable.

SA accepts the new solution, $\hat{\mathbf{x}}$, if this improves the current value of the objective function; however, it also accepts worse solutions probabilistically following a negative exponential distribution controlled by a temperature parameter, T , ([Alg. 1](#), line 5–6). Initially, T is set to T_{init} ; T is then updated at each iteration according to an update function, $U: \mathbb{R} \rightarrow \mathbb{R}$, also called the cooling schedule. The probability of accepting a worse solution is controlled by the current temperature and the difference in the objective function, ΔE , for the two solutions, $\mathbf{x}$ and $\hat{\mathbf{x}}$.

Typically, SA stops when an arbitrary low temperature is reached ([Alg. 1](#), line 3).

Algorithm 1: Simulated annealing (SA)

```

1:    $\mathbf{x} \leftarrow \text{Initialize}()$ 
2:    $T \leftarrow T_{\text{init}}$ 
3:   while  $T \geq T_{\text{min}}$  do
4:      $\hat{\mathbf{x}} \leftarrow N(\mathbf{x})$ 
5:      $\Delta E \leftarrow \exp(-(f(\hat{\mathbf{x}}) - f(\mathbf{x}))/T)$ 
6:     if  $\Delta E \leq \text{Random}(0,1)$  then
7:        $\mathbf{x} \leftarrow \hat{\mathbf{x}}$ 
8:        $T \leftarrow U(T)$ 
9:   return  $\mathbf{x}$ 

```

The probabilistic acceptance criterion of SA mimics the physical annealing process (Kirkpatrick *et al.*, 1983), which is a thermal process used to obtain low energy states of a solid in a heat bath. Interestingly, the annealing process can be simulated on a computer using Monte Carlo sampling (Chandler, 1987; Aarts and Lenstra, 1997). For this reason, many interesting properties of SA can be analyzed using concepts from statistical mechanics.

Given a solid in state s_i with energy E_i ; let s_j be a state obtained from s_i by applying a random perturbation. Let E_j be the energy of the solid in state s_j ; if $\Delta E = E_j - E_i$ is negative, the new configuration is accepted, otherwise the new state is accepted with probability:

$$P(s_i) = \exp(-\Delta E/k_B T) \quad (2)$$

where T is the temperature and k_B is the Boltzmann constant (Aarts *et al.*, 2005); This probabilistic acceptance function is the standard Metropolis criterion. The solid can reach thermal equilibrium at each temperature if T is decreased sufficiently slowly. Indeed, the probability of a solid of being in state s_i with energy E_i at temperature T is given by the Boltzmann distribution:

$$P(s_i) = \frac{\exp(-E_i/k_B T)}{\sum_j \exp(-E_j/k_B T)} \quad (3)$$

where the sum goes over all the possible states configurations Aarts *et al.* (2005).

Using this statistical mechanics framework, a solution $\mathbf{x} \in \mathbb{R}^n$ for an optimization problem with objective function, f , is equivalent to a state of a system with energy $E = f(\mathbf{x})$. The probability density of a n dimensional space can be defined as a Gaussian-Markovian system as follows:

$$g(x) = (2\pi T)^{-n/2} \exp[-\Delta x^2/(2T)] \quad (4)$$

where $\Delta x = x_k - x_{(k-1)}$ is the deviation between two solutions at iteration k (Ingber, 1989).

A solution is accepted probabilistically based on the difference in energy, ΔE , between two solutions at temperature T . Specifically, a new solution is accepted based on the probability of obtaining a new state with energy E_k from another state with energy E_{k-1} as follows:

$$\begin{aligned}
 h(\Delta E) &= \frac{\exp(-E_k/T)}{\exp(-E_k/T) + \exp(-E_{(k-1)}/T)} \\
 &= (1 + (\Delta E/T))^{-1} \\
 &\approx \exp(-\Delta E/T)
 \end{aligned} \quad (5)$$

which is the equivalent to Eq. (3), where the sum involves only the current and the new state.

It is important to note that SA can accept worse solutions probabilistically as a function of T ; however, the probability of accepting a worse solution goes to zero as $T \rightarrow 0$. Indeed, the initial temperature is typically set to a high value, then the update function, U , slowly decreases it to zero; thus, the U function effectively controls the exploration and exploitation ability of the algorithm (Fig. 1). Finding the optimal update scheme, U , is not trivial; however, if T is decreasing not faster than $\frac{T_{\text{init}}}{\ln k}$, it is possible to obtain a global optimal solution for a large enough k (Ingber, 1989).

The convergence of the SA can be slow, and therefore many efforts have been made to design versions that improve its speed either by changing the generating function, $N(\mathbf{x})$, or the temperature update function, U . Indeed, it has been shown that SA can work well for any reasonable generating function, $N(\mathbf{x})$, even without requiring ergodicity. Indeed, using the Cauchy distribution as the generating function, SA has been shown to perform better than the classical Gaussian for the Boltzmann distribution (Ingber, 1989).

An alternative approach to speeding up SA is to transform the objective function, such that algorithm can sample from a smoother solution space. Hamacher and Wenzel (1999) proposed a Stochastic Tunneling (STUN) approach; specifically, the objective function, f , is transformed into a cost function, $\hat{f}(\mathbf{x}) = 1 - \exp(-\lambda(f(\mathbf{x}) - f(\mathbf{x}')))$, where $\mathbf{x}'$ is the best solution found and λ is a scaling factor, which is a parameter of the algorithm. $\hat{f}$ smooths the landscape around the local minima and favors jumps into neighboring basins.

Recent advances in quantum computing showed that *Simulated Quantum Annealing* (SQA) outperforms SA on toy problems by requiring only polynomial time whereas the SA still takes exponential time (Crosson and Harrow, 2016).

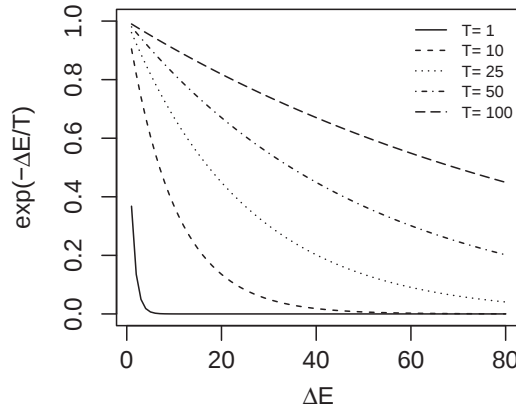


Fig. 1 Probability of accepting worse solutions as a function of the annealing temperature T . On the x-axis we show the difference in energy between two configurations (ΔE), while the y-axis depicts the probability of accepting a solution.

Genetic Algorithms

Genetic Algorithms (GAs) are bio-inspired optimization methods, which mimic the concept of natural selection. The canonical GA was presented in 1975 by [Holland \(1975\)](#), and later systematically described and analyzed by [Goldberg \(1989\)](#).

In their canonical form, GAs are derivative-free algorithms, as they do not require mathematical information about the objective function. Indeed, GAs use random sampling followed by a selection step based on iterative or probabilistic improvement, which we generally refer to as a evolutionary step.

The central idea of GAs is to use a population of candidate solutions, rather than a single candidate solution as in SA. Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be the objective function for an optimization problem; a candidate solution is represented as a vector $\mathbf{x} \in \mathbb{R}^n$, called chromosome, where the i -th variable, $x_i \in \mathbf{x}$, is called a gene. The admissible values of a gene are called alleles; however, while for combinatorial optimization problems this concept is well defined, for continuous optimization problems, it is not. Here, we assume that a solution, $\mathbf{x}$, is a real vector, although GAs can encode solutions in any format; e.g., for the Traveling Salesman Problem (TSP), a chromosome can represent a path over the graph defined by the roads connecting the cities.

To ensure the evolutionary process converges towards good solutions, GAs use a fitness function, F , which assesses the quality of a candidate solution. Obviously, the fitness function can be the objective function, f , itself, or can include additional terms to drive the algorithm towards promising configurations. For example, in the case of constrained optimization, a fitness function, F , can be defined as follows:

$$F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{x}) \quad (6)$$

where f is the objective function, but g and h are functions returning a penalty for violating inequality or equality constraints, respectively.

Let F be a fitness function for the optimization problem, f , and a coding scheme for the candidate solutions' GAs evolve a population of candidate solutions as described in [Alg. 2](#).

Algorithm 2: Genetic algorithm (GA)

```

1:    $P \leftarrow \text{Initialize}(m)$ 
2:   Evaluate( $F, P$ )
3:   while  $\neg \text{StoppingCondition}()$  do
4:      $M \leftarrow \text{Select}(P)$ 
5:      $M \leftarrow \text{Recombine}(M)$ 
6:      $M \leftarrow \text{Mutate}(M)$ 
7:     Evaluate( $F, M$ )
8:      $P \leftarrow \text{Replace}(P, M)$ 
9:   return  $P$ 

```

The algorithm starts by creating an initial population, M , of candidate solutions of size m ([Alg. 2](#), line 1), which is a parameter of the algorithm. Typically, the initial population is created at random, but domain-specific information can be introduced to let the algorithm start from promising solutions. In many cases, the initial population is initialized with the best known solution for a problem, often provided by an expert, or its random perturbations. This is a process called seeding, which is often used for engineering problems, where GAs are used to refine the best known solution; however, this operator could affect the behavior of the algorithm, which might work more like a local optimizer than a global one. The size of the population, m , is a key parameter of the algorithm, which affects both the speed and effectiveness of the GA; indeed, a small population makes the algorithm run faster,

but at the cost of increasing the risk of getting trapped in a local minimum. Conversely, a large population size improves the searching ability of the algorithm, at the expense of increased running time. Choosing an optimal population size is not easy, and typically is done empirically.

The GA evaluates the fitness function, F , for the initial population and then starts the evolutionary process. The first step is to select candidate solutions that will form the mating pool, M , picking solutions with the highest fitness value (Alg. 2, line 4). The underlying idea is to mimic the survival-of-the-fittest mechanism observed in Nature, driving the algorithm towards putative optimal solutions. There are many selection schemes reported in literature; here, we address the most commonly used, while a complete review of selection procedures is presented by [Goldberg \(1989\)](#); [Sastry et al. \(2014\)](#).

A classical selection scheme is the roulette-wheel procedure; according to this scheme, each individual in the population is selected probabilistically based on its fitness value, such that better solutions have higher probability of being selected. Thus, for each solution, i , we define a selection probability, $p_i = F_i / \sum_{j=1}^m F_j$, and a cumulative selection probability, $q_i = \sum_{j=1}^i p_j$. Then, the roulette-wheel procedure selects an individual, z , s.t.

$$z = \arg \min_{i=1 \dots m} r \leq q_i \quad (7)$$

with r being a uniform random value in $[0,1]$. The whole procedure is repeated m times to create a new population of m individuals.

An alternative selection strategy is tournament selection, where s solutions are picked at random, with or without replacement, to compete against each other; then, the best among the s solutions is added to the mating pool. A typical value for the size of the tournament is 2. In the case of selection without replacement, the best solution of the population is always selected and the worst is always discarded, while the others are selected probabilistically based on their fitness function value, with probability $p_i > 0$. Indeed, let ρ_i be the rank induced by the fitness of the i -th individual of the population, s.t. $\rho_i = 1$ for the best solution, and $\rho_i = m$ for the worst one. Let the tournament size be $s = 2$, the selection probability of the i -th individual is $p_i = 1 - ((\rho_i - 1) / (m - 1))$. It follows that the best solution has selection probability $p_1 = 1$, the worst has $p_m = 0$, whereas all other individuals have $p_i > 0$. It is possible to use the tournament selection to ensure that the best individual will go into the mating pool, making this an elitist strategy, whereas this is not guaranteed with the roulette wheel.

After m individuals are added to the mating pool, M , they will undergo a recombination, or crossover, step to create new offspring solutions (Alg. 2, line 5). In general, two solutions can undergo a recombination step with probability, $p_c > 0$, otherwise the offspring solutions will be simply a copy of their parents. The recombination probability p_c is a parameter of the GA. The crossover step is typically tailored to a specific optimization problem, however some schemes have been shown to perform consistently well in practice. The simplest recombination operators are the one- and two-point crossovers. Let $\mathbf{u}$ and $\mathbf{v}$ be two solutions of the mating pool, M , with $\mathbf{u} \neq \mathbf{v}$, one point crossover generates two new solutions $\mathbf{x}$, $\mathbf{y}$ such that:

$$\begin{aligned} \mathbf{x} &= [\mathbf{u}[1, \dots, r_p] \mathbf{v}[r_p + 1, \dots, n]] \\ \mathbf{y} &= [\mathbf{v}[1, \dots, r_p] \mathbf{u}[r_p + 1, \dots, n]] \end{aligned} \quad (8)$$

where r_p is the crossover point picked uniformly at random in $[1, n]$. Analogously, let $\mathbf{u}$ and $\mathbf{v}$ be two solutions of the mating pool, M , with $\mathbf{u} \neq \mathbf{v}$; two point crossover generates two new solutions, $\mathbf{x}$ and $\mathbf{y}$, such that:

$$\begin{aligned} \mathbf{x} &= [\mathbf{u}[1, \dots, r_p] \mathbf{v}[r_p + 1, \dots, r_q] \mathbf{u}[r_q + 1, \dots, n]] \\ \mathbf{y} &= [\mathbf{v}[1, \dots, r_p] \mathbf{u}[r_p + 1, \dots, r_q] \mathbf{v}[r_q + 1, \dots, n]] \end{aligned} \quad (9)$$

where r_p, r_q are the two crossover points picked uniformly at random in $[1, n]$ with $r_p \neq r_q$.

An alternative scheme is uniform crossover. Let $\mathbf{u}$ and $\mathbf{v}$ be two solutions in the mating pool, M , with $\mathbf{u} \neq \mathbf{v}$; uniform crossover exchanges the value in $\mathbf{u}[k]$, $\mathbf{v}[k]$ with probability p_s , which is a parameter of the algorithm. While both point and uniform crossover are crucial in generating diversity in the population, and enhancing the exploring ability of the GA, they produce very different offspring solutions; indeed, while point crossover tends to preserve the substructure of the two mating solutions, uniform crossover can potentially destroy any optimal substructure found by the algorithm.

A common problem of crossover operators is the potential flattening of the mating population; indeed, if two solutions have same value at a given position, crossover will keep it unchanged in the offspring. This is problematic since the algorithm may rapidly get stuck in a local optimum. To overcome this problem, GAs use mutation operators; these operators are applied to each solution in the mating pool to add diversity to the population and ensure the entire search space is explored. In general, a mutation operator is applied to each variable of a candidate solution with probability p_m , which is a parameter of the GA (Alg. 2, line 6). p_m is typically set to a low value to ensure that the probability of destroying any optimal substructure is minimized. In general, mutation operators are tailored to the optimization problem ([Bäck et al., 2000](#)). For combinatorial optimization, a typical mutation operator is the random flip operation, where a variable, x_i , is changed at random to another value in its domain; for example, in the case of binary optimization, this reduces to flipping a bit. For continuous optimization problems, mutation operators add a random value to the current value of a variable x_i following a random distribution; a common strategy is to apply a Gaussian noise, $z \sim N(\mu, \sigma)$, where the mean, μ , and standard deviation, σ , are parameters of the GA ([Yao et al., 1999](#)). Mutation operators based on statistical sampling have proven to be extremely effective in evolutionary strategies, where these are the primary search mechanisms. Indeed, [Hansen et al. \(1995\)](#) proposed an evolutionary strategy for continuous global optimization, which uses a multi-variate Gaussian mutation operator, where the mean and covariance matrix are updated at each step to maximize the likelihood of generating better solutions.

Finally, the GA evaluates the fitness of the mating pool (Alg. 2, line 7) and selects the solutions for the next iteration (Alg. 2, line 8). There are different strategies for selecting this population; a common one consists in picking the entire mating pool M as the new population. This strategy assumes that, since the mating pool was filled with the best solutions of the population, the offspring are likely to be better than their parents, somehow mimicking an elitist strategy. An alternative strategy is to replace k old solutions with k solutions from the mating pool; obviously, this procedure is more complex since different replacement strategies can be used (e.g., replace the k worst) and there is an additional parameter, k , to tune. Both strategies are widely used, but the former provides an easier way to preserve promising solutions from one iteration to the other, therefore favoring the convergence towards putative optimal solutions [Sastry et al. \(2014\)](#).

The GA stops when stopping conditions are met; since we do not have mathematical guarantee that the GA has converged to an optimal solution, the stopping criteria are typically heuristic. In general, the algorithm is stopped when a maximum number of iterations or function evaluations is attained, or when the solution is sufficiently close to a predefined objective function value (e.g., least square minimization). An alternative approach would be to stop the GA when the population diversity drops below a certain threshold. A complete overview of stopping criteria is presented in [Aytug and Koehler \(2000\)](#) and [Safe et al. \(2004\)](#).

Closing Remarks

Finding the global optimum of an arbitrary black-box objective function is impossible in polynomial time, and therefore many heuristic deterministic and stochastic methods have been proposed. In this article, we introduced two stochastic algorithms that have been shown to perform consistently well in practice, namely Simulated Annealing (SA) and Genetic Algorithms (GAs).

SA belongs to the family of Monte Carlo sampling methods, which allows to locate putative optimal solutions in extremely rugged objective function landscapes. Interestingly, despite being a stochastic method, statistical physics provides a mathematical guarantee on its convergence. However, years of research have shown that, despite theoretical and engineering efforts to define efficient sampling and cooling schemes for SA, the algorithm still requires a good understanding of the problem to work effectively. These results have decreased the appeal of this method, in favor of new heuristic algorithms that perform well in practice despite limited theoretical foundations.

GAs are stochastic algorithms that iteratively apply sampling procedures and elitism selection strategies to identify putative optimal solutions. In contrast to SA, GAs sample a population of candidate solutions, therefore increasing the probability of identifying a satisfactory solution. Despite limited advances in the analysis of the running time and convergence properties of these methods, GAs are routinely applied with success in many real-world optimization problems, often outperforming deterministic methods. However, similar to SA, GAs are controlled by significant number of parameters, which often require extensive tuning. It is safe to say that, based on recent empirical results, GAs provide a more general optimization framework for global optimization.

References

- Aarts, E., Korst, J., Michiels, W., 2005. Simulated annealing. *Search Methodologies*. 187–210.
- Aarts, E.H., Lenstra, J.K., 1997. *Local Search in Combinatorial Optimization*. Princeton University Press.
- Alizamir, S., Rebennack, S., Pardalos, P.M., 2008. Improving the neighborhood selection strategy in simulated annealing using the optimal stopping problem. *Simulated Annealing*. In Tech.
- Audet, C., Dennis, J.E., Le Digabel, S., 2010. Globalization strategies for mesh adaptive direct search. *Computational Optimization and Applications* 46 (2), 193–215. Available at: <https://doi.org/10.1007/s10589-009-9266-1>.
- Aytug, H., Koehler, G.J., 2000. New stopping criterion for genetic algorithms. *European Journal of Operational Research* 126 (3), 662–674.
- Bäck, T., Fogel, D.B., Michalewicz, Z., 2000. *Evolutionary Computation 1: Basic Algorithms and Operators*, vol. 1. CRC press.
- Chandler, D., 1987. Introduction to modern statistical mechanics. In: Chandler, D. (Ed.), *Introduction to Modern Statistical Mechanics*. Oxford University Press, p. 288. ISBN-10: 0195042778. ISBN-13: 9780195042771, 288.
- Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., et al., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. Springer.
- Conn, A.R., Scheinberg, K., Vicente, L.N., 2009. Introduction to derivative-free optimization. MPS-SIAM Book Series on Optimization, SIAM, Philadelphia, PA. doi:10.1137/1.9780898718768.
- Crosson, E., Harrow, A.W., 2016. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In: *IEEE Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 714–723.
- Floudas, C.A., Pardalos, P.M., 2008. *Encyclopedia of Optimization*. Springer Science & Business Media.
- Goldberg, D., 1989. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.
- Hamacher, K., Wenzel, W., 1999. Scaling behavior of stochastic minimization algorithms in a perfect funnel landscape. *Physical Review E* 59 (1), 938.
- Hansen, N., Ostermeier, A., Gawelczyk, A., 1995. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: *ICGA*, pp. 57–64.
- Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems. An Introductory Analysis With Application to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press.
- Horst, R., Pardalos, P.M., 2013. *Handbook of Global Optimization*. Dordrecht: Kluwer Academic Publishers.
- Ingber, L., 1989. Very fast simulated re-annealing. *Mathematical and Computer Modelling* 12 (8), 967–973.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., et al., 1983. Optimization by simulated annealing. *Science* 220 (4598), 671–680.
- Mavridou, T.D., Pardalos, P.M., 1997. Simulated annealing and genetic algorithms for the facility layout problem: A survey. *Computational Optimization and Applications* 7 (1), 111–126.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21 (6), 1087–1092.

- Neumaier, A., 2004. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica* 13, 271369.
- Safé, M., Carballido, J., Ponzone, I., Brignole, N., 2004. On stopping criteria for genetic algorithms. *Advances in Artificial Intelligence-SBIA 2004*, 405–413.
- Sastry, K., Goldberg, D.E., Kendall, G., 2014. Genetic algorithms. *Search Methodologies*. Springer. pp. 93–117.
- Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation* 3 (2), 82–102.

Biographical Sketch

Giovanni Stracquadanio is a lecturer in Computer Science and Artificial Intelligence at University of Essex. He is the Head of the Computational Intelligence Lab for Biology and Medicine, which focuses on the development of Computer Aided Design (CAD) methods for genome engineering and data-analysis methods for cancer genomics. He is also the recipient of the Wellcome Trust Seed Award in Science. His research work has been published in prestigious peer-reviewed journals, including *Science*, *Nature Reviews Cancer*, *PNAS* and *Genome Research*.

Panos M. Pardalos serves as distinguished professor of industrial and systems engineering at the University of Florida. Additionally, he is the Paul and Heidi Brown Preeminent Professor of industrial and systems engineering. He is also an affiliated faculty member of the computer and information science Department, the Hellenic Studies Center, and the biomedical engineering program. He is also the director of the Center for Applied Optimization. Pardalos is a world leading expert in global and combinatorial optimization. His recent research interests include network design problems, optimization in telecommunications, e-commerce, data mining, biomedical applications, and massive computing. Pardalos is a fellow of AAAS, INFORMS, and AIMBE. He is the founding editor of several journals, including *Optimization Letters*, the *Journal of Global Optimization*, and *Energy Systems*. In 2013 Pardalos, has been awarded the 2013 EURO Gold Medal prize, bestowed by the Association for European Operational Research Societies. This medal is the preeminent European award given to Operations Research (OR) professionals.

Data Mining in Bioinformatics

Chiara Zucco, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Thanks to the development of technologies capable of carrying out a massive number of tests and, consequently, of acquiring a massive amount of data at once, in recent decades the volume of extracted data related to biological sciences, has significantly increased in terms of complexity, heterogeneity and size. As a consequence, we have witnessed a deep transition of biological sciences that, from mainly driven-by-hypotheses disciplines, have now also become data-driven sciences.

Managing biological data and understanding its underlying correlations, structures and models is one of the main objectives of bioinformatics, an interdisciplinary field encompassing biology, biochemistry, mathematics and information technology and statistics.

Huang *et al.* (2013), define a general workflow for bioinformatics research. Naturally, this process, if necessary, can be cyclical and consists of 4 sub-processes:

- Laboratory collection of experimental data and biological samples.
- Data acquisition by high-throughput omics technologies.
- An analysis of the data is carried out using computational/mathematical/statistical methods.
- Validation of the results with further experimental tests.

The significantly volume's growth of biological data in terms of complexity, heterogeneity and size, constitutes a virtually unlimited potential for information growth but it also poses several challenges. In fact, paradoxically, it becomes increasingly difficult to extract relevant information from the superfluous ones and traditional methods of analysis that solely rely on manual and statistical analysis and domain expert's interpretation result impassable.

Data Mining is a discipline resulting from the combination of classical statistics tools and computer science algorithms, such as machine learning, which allows the extrapolation of knowledge from a large amount of data for the scientific, operational or industrial use of this knowledge.

Background and Fundamentals

Data mining is born as a particular step of a process defined in literature as *Knowledge Discovery in Databases* (KDD). Due to the centrality acquired by Data Mining task within this process, nowadays Data Mining is used as synonym for Knowledge Discovery, also known as Knowledge Discovery and Data Mining.

Using this identification, and reporting Fayyad *et al.* (1996) definition, we can say that Data Mining or KDD is

The non-trivial process of identifying patterns in data that are valid, original, potentially useful and understandable.

Better detailing this definition, it can be said that

Definition 2.1: Knowledge discovery is composed by various steps (*processes*) and has the aim of extracting models or, in general, recurrent structures in data (*patterns*) from data that are:

- *Valid* in the sense that the discovered patterns must occur on new data with a certain degree of certainty,
- *Original*, that is the more the extracted knowledge is not obvious or a priori guessed, the more valuable it is,
- *Potentially useful*, in the sense that it must provide some benefit to the final user or to the following step in the process;
- *Understandable*, i.e., patterns extracted from data must facilitate the understanding of data itself, if not immediately, after a post-processing step.

The search for valid, new, useful and understandable patterns implies that we can define quantitative measures for evaluating patterns, for example estimates of prediction's accuracy on new data, estimates of a suitable defined gain, estimates of complexity of the patterns extracted, etc.

Witten and Frank Witten *et al.* (2016) add to the previous definition that the process should be *automatic* or *semi-automatic*, emphasizing the relationship between Data Mining and Machine Learning.

As shown in Fig. 1, Han *et al.* (2011) synthesize the various steps that characterize the KDD in:

- *Data cleaning*. Given the increasing amount of data that we have said to be heterogeneous, datasets typically present missing data, inconsistent data or data which are susceptible to "noise". A low data quality affects in a negative fashion the information extraction process and, for this reason, as a first step is necessary to remove incomplete or inconsistent data.

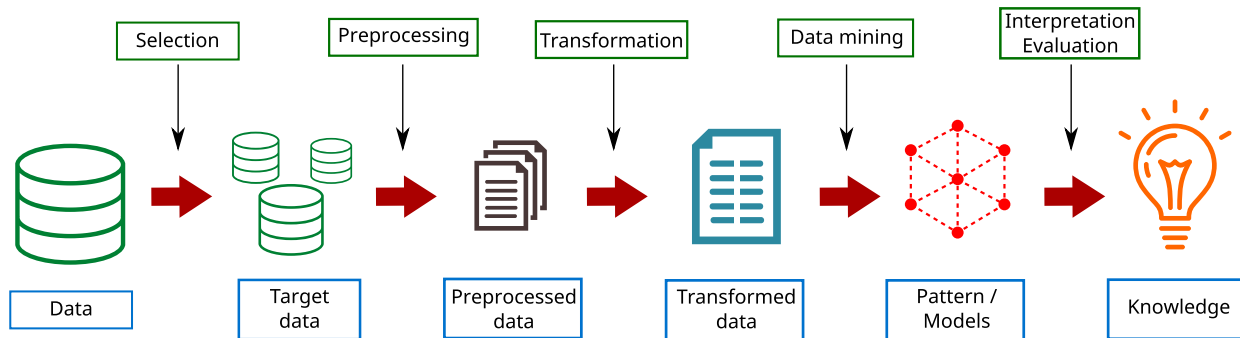


Fig. 1 The steps for Data Mining process.

- *Data integration.* This step is necessary if the data come from heterogeneous sources and have to be consistently aggregated.
- *Data selection.* Relevant data are selected.
- *Data transformation.* In this step the data is transformed or consolidated into an appropriate form for the following task. The data can be aggregated, generalized through the use of hierarchies, normalized in a specific range and finally can be integrated new attributes or features, useful for the process of data mining.
- *Data mining.* It consists in the search for models of interest to the user, with subsequent refinements, presented according to defined modes of representation (classification, decision trees, regression, clustering).
- *Pattern evaluation.* Once extracted, the patterns considered valid are selected based on the application of appropriate measures.
- *Knowledge presentation.* The extracted information is presented.

Each step of this process has developed independently from the others, providing a plethora of algorithms. For this reason, a common approach, also useful in bioinformatics environment, involves the modeling of the discovery process through the use of workflows, which help the automation of the process allowing, at the same time, the possibility of testing in parallel different algorithms belonging to the same task of that KDD process.

Data Mining Task

What are the final objectives of KDD? Still in Fayyad's article two categories of objectives are indicated, namely the *Verification* and the *Discovery*. Moreover, the goals for Discovery can be *Prediction* or *Description*. While descriptive discovery aims to find patterns and associations that can be interpreted by the expert of that domain, predictive discovery aims at finding patterns to predict the behavior of other data.

In Data Mining or in the Analysis task, new knowledge – thought as new high-level information expressed in terms of models, association rules, trends or predictions- is extracted from low-level data through approaches deriving from the combination of statistics and computer science tools, such as machine learning.

extraction of high-level information (knowledge) from low-level data (usually stored in large databases) In particular from Machine Learning, Data Mining has derived the distinction between supervised and unsupervised approaches. We briefly describe these techniques with reference to studies in the medical field.

Let suppose to have a set of cases or *instances* and each instance is represented by a set of variables or *attributes*. The first step of supervised learning techniques, is to build a model by training a machine learning algorithm on a set of examples, that are instances containing both input (called features or attributes), and output, consisting in an attribute called *learning target*. The learning algorithm uses these attributes to determine the set of parameters necessary to be able to learn a model that can correctly predict the output. Subsequently, this model is used to predict the unknown target values of new instances.

Two typical examples of supervised learning are *Classification* and *Regression*. If in *Classification* the target attribute is generally nominal and represents the class to which an instance belongs to, the aim of regression is to predict output value of a dependent numerical variable starting from one or more independent features, by inferring a real-valued function.

From this point of view, supervised learning techniques can be used, for example, to define criteria for diagnosis/prognosis or to predict a trend of clinical/vital factors that depend on the clinical subject's data.

Some among the most famous examples of supervised learning algorithms are

- Classification trees,
- Linear classifiers (SVM, Neural Network),
- Rule based classifiers,
- Probabilistic Classifiers (Naive Bayes, Bayesian network, Maximum Entropy).

In *unsupervised learning* structure in data is seek to be found without having examples or labeled outputs, but only relying on similar characteristics to try to make reasonings and forecasts on data and examples of unsupervised learning are, in general, *clustering* and *associative rules*.

In particular, the goal of clustering is to group in sets or *clusters* data that a similar in some way when, unlike classification, these groups are not known a priori. In order to define this similarity, clustering techniques are based on suitable *distance measures*.

Introduced in basket analysis, *associative rules* have found application in many domain, especially in the field known as frequent pattern mining. In fact association rules do not aim to predict values but to search for relationships and dependencies among data. In particular, an association rules can be defined as the relation between two sets of items (i.e., binary attributes) $X \rightarrow Y$ with the meaning that if X occurs, it's likely that Y also occurs.

A major issue of data mining step is to appropriately choose one or a combination of these techniques in order to find a efficiently and reliably solution to a given problem.

In recent years, predictive Data Mining has been widely used in bioinformatics, for example it finds application in genomics and proteomics for: Protein-protein interaction prediction problems (see for example Hood and Friend, 2011; Haider et al., 2015; Nafar and Golshani, 2006; Zhang et al., 2010); protein structure prediction (see Lexa et al., 2009; Fayeche et al., 2013); motif identification in biological data (see Narasimhan et al., 2002; Hoque et al., 2011; Qader and Al-Khafaji, 2014); in gene finding problems it's used in order to identify coding regions of genes etc.

Another important application of Data mining is the discovery of genomic biomarkers for disease discovery or progression, with the aim to obtain a diagnosis that can be more precocious and precise and a prognosis that can lead to more individualized therapies, see Mancilla et al. (2017) -this is for example the perspective of P4-medicine (predictive, personalized, preventive, participatory) Hood and Friend (2011), see also Guzzi et al. (2015).

One important example can be found in the work of Golub et al. (1999), in which is carried out a study focused on the early differential diagnosis of acute myeloid leukemia and acute limbo plastic leukemia.

Nevins et al. (2003) proposed a decision-driven approach to predict patient survival. Predictive data mining approaches have been widely used to predict the outcome in cases of prostate cancer, ovaries, lungs.

More recent examples can be found in the works of Yousefi et al. (2017); Shu et al. (2018); Mobadersany et al. (2018).

Feature Selection

So far, Data Mining process and Machine Learning approaches were presented without any further consideration on dataset dimensions and on the number of attributes in a dataset.

However, it is clear that some variables can be significant for the construction of a model while others maybe not. However, generally speaking, the Data Mining step take into account all the features and this, especially in cases where there are many features, as in bioinformatics, involves problems in terms of accuracy and, also, of performances.

Therefore, it makes sense to ask how to reduce the attributes or features, by selecting only those that are more significant than the others for the construction of a predictive model. Feature selection is the process by which a subset of attributes, regarded as relevant in the construction of a model, is selected.

Feature selection is distinguished by the so-called features reduction techniques, for example based on projection or compression methods, because although both aim to reduce the number of attributes in a dataset, while in the feature selection a subset of attributes is extracted, feature reduction produces a set of combined attributes while previous feature are discarded and this, especially in bioinformatics field, can lead to a loss of information or to interpretation issues.

Dash and Liu (1997), subdivide the feature selection process into 4 phases: *subset generation*, *subset evaluation*, *stopping criterion* and *result validation*.

The first step is based on a search criterion that can be complete, sequential or random. It is necessary to underline that in a n -dimensional feature space, the feature selection methods try to find the best subset, between the 2^n candidate subsets and therefore the problem has exponential complexity with respect to the data dimension.

Several heuristic strategies have been developed to overcome this problem. Guyon et al. (2002) proposed a recursive elimination function (RFE) that performs backward deletion, with the aim of removing the most irrelevant characteristics in a sequential and iterative way, until only a subset of a given dimension remains, according to some stop criteria.

Based on the second step, Yu and Liu (2003) subdivide the feature selection criteria into dependent and independent of the mining algorithm. Among the most common stopping criteria, Yu and Liu (2003) distinguish the following: the search is complete, a bound has been reached (for example a specific number of iterations), the addition or elimination of a feature does not produce a better subset.

The features selection algorithms can be divided into three general categories: *filter methods*, *wrapper methods* and *embedded methods*.

Filter methods select a subset of variables during preprocessing, taking into account only the information content and the intrinsic properties of the data without considering the mining algorithm used. In particular, filter methods apply statistical measures that classify features by assigning a "score" for each of these.

Moreover, these statistical methods are often univariate and consider features independent of one another or in relation to a dependent variable. Some filtering methods are: Pearson Correlation, Mutual Information, Chi Square, Fisher Score.

Wrapper methods consider selecting a subset of features in terms of the following search problem: “What is the subset of features that gives better results based on a chosen learning algorithm?”. So the learning algorithm is used as part of the subset prediction process, in a sort of black box that determines various combinations of the feature set, evaluates them according to the learning algorithm and classifies them in terms of accuracy. The subset that obtains a higher accuracy index is chosen as the final subset. This black box can apply methodological, stochastic or heuristic techniques. In general, wrapper methods are superior to filter methods in terms of performance, but they are slower and computationally more expensive.

In Embedded methods, variable selection takes place as part of the training process, making it more efficient than wrapper methods. Embedded methods incorporate the selection variable within the training process. The most common are the regularization methods. In addition, the CART methods have within them the mechanisms of embedded feature selection.

Data Mining Platforms

This section and, in general, the whole second part of this article, is devoted to give a brief overview of Data Mining and Machine Learning platforms and to introduce programming environments that, at present, are the most used for both bioinformatics and general domain applications.

In particular, in this Section the most popular Data Mining platforms are presented.

WEKA

One of the best known Data Mining software is WEKA, standing for Waikato Environment for Knowledge Analysis and developed by the University of Waikato, in New Zealand, see (Witten *et al.*, 2016; Hall *et al.*, 2009).

It is a free software environment for machine learning written in Java, available for download at: see “Relevant Websites section”.

Weka’s popularity lies in the fact that not only it contains a vast collection of tools for data preprocessing and modeling along with the most used state-of-the-art data mining algorithms, but it also allows an easy access to these functionalities, due to a user-friendly graphical interface that allows the use also for non-expert of Java programming language, in which Weka is written. On the other hand, for Java developers, it is possible to embed Weka algorithm in a Java application.

KNIME

KNIME, standing for Konstanz Information Miner, see Berthold *et al.* (2009), is an open source platform for data analysis, data reporting and integration, data manipulation and visualization, available for download at see “Relevant Websites section”. It integrates libraries of other suites, with an and easy to use interface. KNIME integrates various components of machine learning and data mining through its modular data pipelining concept, and allows data to be added in a workflow. The graphical user interface (GUI) facilitates the assembly of nodes for data pre-processing and the ETL process (Extraction, Transformation, Loading), for modeling, data analysis and visualization. It is written in Java and based on Eclipse and gives the possibility to develop and add plugins that provide additional functionality.

In Gartner’s 2017 magic quadrant (source Linden *et al.*, 2017), KNIME is still indicated as a Leader for data analysis platforms, counting among its strengths the availability of advanced features for data preparation, such as transformation and aggregation, or for feature selection task. A weakness point outlined in the same report is a lower rate of innovation than other leaders.

Rapid Miner

Another leader among the data science platforms that was recently confirmed by Gartner’s magic quadrant is Rapid Miner, see “Relevant Websites section”. RapidMiner, that offers a limited free-version, provides an integrated environment for data science, supporting all phases of the machine learning process, including data preparation, results visualization, model validation and optimization. This along with the user-friendliness and the adaptability, is one the strengths for this tool.

In fact Rapid miner offers different functionalities for different types of users. In particular, the template-based frameworks provide the possibility to develop guided models without the need of coding, and this is useful for user’s that don’t have confidence with programming languages, while for data and computer science experts, it is possible to access advanced analysis tools, for example by embedding algorithms in R and Python scripts. As for KNIME, RapidMiner provides a platform for developers to create data analysis algorithms and share it with other users.

Program Environment for Data Mining

In this section a quick overview of three of the major quantitative software programming languages, MATLAB, Python and R is discussed.

MATLAB

MATLAB (standing for MATrix LABoratory) is a proprietary programming language developed by MathWorks and a multi-paradigm numerical computing environment whose development was initiated by Cleve Moler in the late 1970s, see “Relevant Websites section”.

As the name suggests, MATLAB is one of the easiest-to-use languages when dealing with objects represented as a numeric matrix. Besides that, MATLAB allows the tracing of functions and data, the implementation of algorithms, supporting the interface with programs written in other languages. Popular both among engineering and image processing experts, MATLAB offers a range of program extensions and libraries, called toolboxes, integrated within the main MATLAB interface. This, in addition to the simplicity and power of the matrix manipulation language, is one of the reasons why MathWorks is mentioned as a Challenger in Gartner’s 2017 magic quadrant. A complete description of Matlab is beyond our intentions, so we will focus on the Data Mining Toolbox.

Statistics and Machine Learning Toolbox provides statistical functions and machine learning algorithms for data analysis and modeling. In particular, the toolbox provides descriptive statistics and plots tools and clustering algorithms for performing exploratory data analysis, feature transformation and feature selection algorithms for dimensionality reduction; supervised and unsupervised machine learning algorithms etc.

Among the issues reported by Gartner’s 2017 magic quadrant, one of the biggest is that the as the dataset dimension increases, the analysis performance has a decrease; furthermore, the lack of a visual pipeline makes it less user-friendly than other data science platforms and, therefore, less chosen by non-programmers data scientists. At the same time, data analyst and statisticians tend to prefer open source languages like Python or R.

Python and R

Both Python and R are popular programming languages, used for the development of statistical computing and Machine Learning applications.

While R (see “Relevant Websites section”) was developed in 1995 by Ross Ihaka and Robert Gentleman, [Lafaye de Micheaux et al. \(2013\)](#), as an open source tool with the aim of providing a better and more user-friendly tool for data analysis, statistical and graphics models, Python (see “Relevant Websites section”) is a general purpose programming language developed in late 80’s by Guido van Rossum, see [Zelle \(2004\)](#), with the aim of emphasizing code readability and an easy-to-understand syntax.

R is widely used for the development of statistical software and data analysis algorithms. This for numerous reasons, including:

- As already mentioned, R is designed as a tool to support a statistical analysis or data analysis.
- Being a free and open source development software and having an active development community, R offers a very large number of additional packages, available at the Comprehensive R Archive Network (CRAN) or other repositories such as Bioconductor or GitHub.
- The visualization packages, among all an example is the famous ggplot2, allow to obtain elegant and informative graphics that allow to understand data in a more efficient and effective way.

Notwithstanding these advantages, some issues can be found in the lower user-friendliness of R compared to software providing GUI interface and dealing with graphical workflows for data analysis process. Another weak point can be found in time performances, suffering the comparison with other languages.

Python is also very popular for Data Analysis and Data Mining applications, thanks to its simplicity, the reliability of the written code, the presence of an integrated test framework, the possibility to integrate code inside a notebook, given precisely from Jupyter Notebook for sharing code and material. Moreover, as already mentioned, Python is a general purpose programming language, which allows greater versatility of use.

However, in relation to R, Python suffers the comparison both in Data Visualization, often less elegant than the competitor, and in the number of packages proposed for the Data Analysis.

Data Mining Platforms for Bioinformatics

Ms-Analyzer

MS Analyzer is a tool for the analysis of mass spectrometry proteomics data, see ([Cannataro and Veltri, 2007](#)). One of the major challenge of mass spectrometry proteomics data analysis is the need of a combination of large storage systems, preprocessing techniques, and data mining and visualization tools.

Collection, storage and analysis of huge mass spectra produced in different laboratories can leverage the services of computational grids, that offer efficient data transfer primitives, effective management of large data stores, and large computing power.

The architecture of this tool is formed by two main components, called modules :

- Proteomics data storage system
- Ontology-based workflow editor.

The first module is called SpecDB and implements basic spectra management functions. It divides data in three different repositories:

- Raw spectra repository, or RSR, which stores the raw spectra coming from different experiments and instruments;
- Pre-processed spectra repository, or PSR, which stores the pre-processed spectra;
- Pre-processed and prepared spectra repository (PPSR), which stores spectra that have been pre-processed and are ready to be analyzed by data mining platforms and tools, like Weka.

The ontology-based workflow editor takes advantage of ontologies for modeling Data Mining steps, i.e., preprocessing, data preparation and data analysis tasks, in the view of supporting the design of bioinformatics workflows, which are assumed to be executed in a distributed environment and that every single activity can be performed by invoking both a Web service or another (sub)workflow.

In particular the ontology-based workflow editor is composed by:

1. A dataset manager, that allows to associate to raw spectra, produced by mass spectrometers, with further information, for example related to classes information, and then giving to the user the possibility to choose whether to perform pre-processing, data preparation and/or data analysis;
2. An ontology manager, that allows the user to perform, the browsing, searching and selection of bioinformatics tools. It is based on two ontologies, i.e., ProtOntology for the modeling concept, algorithms and tools related to the proteomic domain and the biological background, and WekaOntology used for the modeling of concept, algorithms and tools that are more related to the Data mining analysis of proteomic data.
3. The graphical Editor, that helps user to graphically design a bioinformatics workflow in UML (Unified modeling language), by dragging and dropping the needed blocks and then associating them one with another.

Dmet-Miner

DMET-Miner, [Agapito et al. \(2015\)](#) is a data mining tool able to extract Association Rules from DMET (Drug Metabolism Enzymes and Transporters) datasets. The Affymetrix DMET microarray platform allows to determine the ADME gene variants of a patient and to correlate them with drug dependency events.

Affymetrix DMET platform enables the simultaneous investigation of all the genes that are related to drug absorption, distribution, metabolism and excretion (ADME).

DMET-Analyzer, [Guzzi et al. \(2012\)](#), is a tool developed in order to perform an automatic association analysis among the variation of the patient genomes and the different patients response to drugs, in the perspective of P4 medicine.

DMET-Analyzer is able to correlate a single variant for each probe, its analysis strategy doesn't allow the discovery of multiple associations among allelic variants.

To bypass these limitations, DMET-Miner tool was developed, that extends the DMET-Analyzer tool performing data mining strategies and correlating the presence of a set of allelic variants with the conditions of patient's samples by exploiting association rules. It uses an efficient data structure and implements an optimized search strategy that reduces the search space and the execution time.

This tool is developed in JAVA and is available under Creative Commons License. It automatically filters useless rows by iteratively applying the Fishers test filter allowing to reduce the search space, improving the performance.

Bioconductor

Bioconductor is an open source and open development software project for computation biology, based on R programming Language see "Relevant Websites section". In particular, Bioconductor works with a high throughput genomic data from DNA sequence, microarray, proteomics, imaging and a number of other data types ([Gentleman et al., 2006](#)).

Following the release policy of R, Bioconductor generally provides two versions per year.

Most of the available packages are software packages performing some analytical calculation. The current available version of Bioconductor (Version 3.6) consists of 1477 software packages.

However, although in smaller numbers, there are also annotation packages (currently around 900) that are similar to databases and are intended to link identifying information (e.g., Affymetrix identifiers, PubMed), with additional information contained in databases such as GenBank, the Gene Ontology, LocusLink etc. Finally, there are also experimental data packages (about 300) and each package generally contains a single data set that can be related or not to some Bioconductor software packages, in order to illustrate particular case studies, or can be related to publications, etc.

Because of its open development nature, particular importance is given to the documentation relating to the packages.

More in details, the Bioconductor policy provides that for each package at least a vignette should be present, i.e., a document in PDF or HTML format that generally includes descriptive text, figures and bibliographic references and also input and output in R that make these vignettes to be easily executable and suitable for an interactive use.

Webmev

Originally developed by The Institute for Genomic Research (TIGR, that merged in 2006 into the J. Craig Venter Institute), the MultiExperiment Viewer (MeV) tool was born as a member of the TM4 suite, an open source suite for the management and analysis of data extracted from microarray experiments, see (Howe *et al.*, 2011). Howe, Sinha, Schlauch, and Quackenbush]. The TM4 suite's goal was to make data analysis more accessible and available for researchers belonging to different fields and consisted of 4 tools:

- The Microarray Data Manager (MADAM), for the management of microarray data through MySQL;
- SpotFinder, for image processing,
- Microarray Data Analysis System (MIDAS) for the normalization and the preprocessing of raw experimental data, in order to create a suitable input for the MeV tool,
- MeV, performing data analysis for the identification of gene expression patterns.

In particular, MeV was developed as a Java based standalone application. Some issue of this approach were related, for example, to the lack of portability and scalability, to the heavy dependence of the application on the computer environment and to the request of downloading datasets on the user's local machine.

For this reasons, a Web Server open-source tool, WebMeV (see "Relevant Websites section") has been recently developed (see Wang *et al.*, 2017), with the aim of taking advantage of modern cloud architectures, relying on a computing server deployed on Amazon Web Service (AWS). Then, through Rserve (an R package that allows to send binary requests to R playing the role of a socket server) the WebMeV application provides advanced analysis methods adapted from Bioconductor packages and related to Normalization, Clustering, Statistics, Meta Analysis.

Conclusions

In this article an introduction Data Mining process has been provided. In particular, Data Mining has been introduced as a central step in the process of Knowledge Discovery in Databases (KDD), with which it is often identified. A brief overview of the KDD steps was then presented, together with the main approaches used in Data Mining, also providing some applications in the bioinformatics field.

In the second part of this article, some of the most known platforms and programming environments related to Data Mining have been presented. Finally, some specific Platforms for Data Mining applications in Bioinformatics were discussed.

References

- Agapito, G., Guzzi, P.H., Cannataro, M., 2015. Dmet-miner: Efficient discovery of association rules from pharmacogenomic data. *Journal of Biomedical Informatics* 56, 273–283.
- Berthold, M.R., Cebron, N., Dill, F., *et al.*, 2009. Knime-the konstanz information miner: Version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter* 11 (1), 26–31.
- Cannataro, M., Veltri, P., 2007. Ms-analyzer: Preprocessing and data mining services for proteomics applications on the grid. *Concurrency and Computation: Practice and Experience* 19 (15), 2047–2066.
- Dash, M., Liu, H., 1997. Feature selection for classification. *Intelligent Data Analysis* 1 (1–4), 131–156.
- Fayech, S., Essoussi, N., Limam, M., 2013. Data mining techniques to predict protein secondary structures. In: *Proceedings of the 5th International Conference on IEEE Modeling, Simulation and Applied Optimization (ICMSAO)*, pp. 1–5.
- Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. From data mining to knowledge discovery in databases. *AI Magazine* 17 (3), 37.
- Gentleman, R., Carey, V., Huber, W., Irizarry, R., Dudoit, S., 2006. *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer Science & Business Media.
- Golub, T.R., Slonim, D.K., Tamayo, P., *et al.*, 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286 (5439), 531–537.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46 (1), 389–422.
- Guzzi, P.H., Agapito, G., Di Martino, M.T., *et al.*, 2012. Dmet-analyzer: Automatic analysis of affymetrix dmet data. *BMC Bioinformatics* 13 (1), 258.
- Guzzi, P.H., Agapito, G., Milano, M., Cannataro, M., 2015. Methodologies and experimental platforms for generating and analysing microarray and mass spectrometry-based omics data to support p4 medicine. *Briefings in Bioinformatics* 17 (4), 553–561.
- Haider, S., Lipinski, Z., Przewlaka, M.R., *et al.*, 2015. Dapper: A data-mining resource for protein-protein interactions. *BioData Mining* 8 (1), 30.
- Hall, M., Frank, E., Holmes, G., *et al.*, 2009. The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter* 11 (1), 10–18.
- Han, J., Pei, J., Kamber, M., 2011. *Data mining: Concepts and Techniques*. Elsevier.
- Hood, L., Friend, S.H., 2011. Predictive, personalized, preventive, participatory (p4) cancer medicine. *Nature Reviews Clinical Oncology* 8 (3), 184.
- Hoque, F., Mohebujaman, M., Noman, N., 2011. Informative motif detection using data mining. *Research Journal of Information Technology* 3 (1), 26–32.
- Howe, E.A., Sinha, R., Schlauch, D., Quackenbush, J., 2011. Rna-seq analysis in MeV. *Bioinformatics* 27 (22), 3209–3210.
- Huang, X., Bruce, B., Buchan, A., *et al.*, 2013. No-boundary thinking in bioinformatics research. *BioData Mining* 6 (1), 19.
- Lafaye de Micheaux, P., Drouilhet, R., Lique, B., 2013. The R software. In: *Proceedings of the Fundamentals of Programming and Statistical Analysis*.
- Lexa, M., Snášel, V., Zelinka, I., 2009. Data-mining protein structure by clustering, segmentation and evolutionary. In: *Proceedings of the Algorithms Foundations of Computational Intelligence*, vol. 4, pp. 221–248. Springer.
- Linden, A., Krensky, P., Hare, J., *et al.*, 2017. Magic quadrant for data science platforms. In: *Proceedings of the Gartner & Forrester & Aragon, Collection*, pp. 28–29.
- Mancilla, G., Oyarzun, I., Artigas, R., *et al.*, 2017. A data mining strategy identifies microRNA-15b-5p as a potential bio-marker in non-ischemic heart failure.
- Mobadersany, P., Yousefi, S., Amgad, M., *et al.*, 2018. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*. 201717139.
- Nafar, Z., Golshani, A., 2006. Data mining methods for protein-protein interactions. In: *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, CCECE'06*, pp. 991–994.

- Narasimhan, G., Bu, C., Gao, Y., *et al.*, 2002. Mining protein sequences for motifs. *Journal of Computational Biology* 9 (5), 707–720.
- Nevins, J.R., Huang, E.S., Dressman, H., *et al.*, 2003. Towards integrated clinico-genomic models for personalized medicine: Combining gene expression signatures and clinical factors in breast cancer outcomes prediction. *Human Molecular Genetics* 12 (suppl_2), R153–R157.
- Qader, N.N., Al-Khafaji, H.K., 2014. Motif discovery and data mining in bioinformatics. *International Journal of Advanced Computer Technology* 13 (1), 4082–4095.
- Shu, C., Wang, Q., Yan, X., Wang, J., 2018. Whole-genome expression microarray combined with machine learning to identify prognostic biomarkers for high-grade glioma. *Journal of Molecular Neuroscience*. 1–10.
- Wang, Y.E., Kutnetsov, L., Partensky, A., Farid, J., Quackenbush, J., 2017. Webmev: A cloud platform for analyzing and visualizing cancer genomic data. *Cancer Research* 77 (21), e11–e14.
- Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
- Yousefi, S., Amrollahi, F., Amgad, M., *et al.*, 2017. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports* 7 (1), 11707.
- Yu, L., Liu, H., 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 856–863.
- Zelle, J.M., 2004. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates, Inc.
- Zhang, S.-W., Li, Y.-J., Xia, L., Pan, Q., 2010. Pplook: An automated data mining tool for protein-protein interaction. *BMC bioinformatics* 11 (1), 326.

Relevant Websites

<https://www.bioconductor.org>
Bioconductor.

<https://www.knime.com/>
KNIME.

<https://uk.mathworks.com/>
MathWorks.

<https://www.python.org/>
Python.

<https://www.r-project.org/>
R.

<https://rapidminer.com/>
RapidMiner.

<http://mev.tm4.org>
TM4 MeV.

<https://www.cs.waikato.ac.nz/ml/weka/downloading.html>
WEKA.

Knowledge Discovery in Databases

Massimo Guarascio, Giuseppe Manco, and Ettore Ritacco, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Knowledge Discovery in Databases (*KDD*) (Fayyad and Uthurusamy, 1996) is an interdisciplinary science whose goal is to extract useful and actionable knowledge from very large data repositories. Mainly, given a data set, a KDD process aims at finding:

- *Classifiers*. A classifier is a decider able to divide data into predefined categories, often called classes.
- *Predictors*. A predictor is a fitting function able to predict a target feature exploiting the remaining data.
- *Clusterings*. A clustering is a process that divides data into unknown categories, called clusters, according to the *similarity* of the data points.
- *Patterns*. A pattern is a discernible regularity within the data, whose elements and/or features repeat in a predictable schema.
- *Anomalies*. An anomaly, often called outlier, is an unexpected piece of information that considerably deviates from the rest of the data.
- *Associations*. An association is a link between two or more phenomena, coded in pieces of information.
- *Models*. A model is a mathematical and/or logic sets of functions able to describe data distributions and behaviors.

In the last decades, KDD has proven to be an essential element of our research, economy and society due to the increasing availability of data in private, public and web databases. The application fields are vast and space in different domains, for instance:

- *Healthcare*. In the last decades, KDD was successfully used to improve several medical aspects: (i) The health system management, by identifying best practices, optimizing processes and reducing costs; (ii) the fraud and abuse detection; (iii) the disease recognition; (iv) the treatment decision making.
- *Market*. KDD is massively exploited in Business and Market analysis. The main objective is to find a model able to fit user segments and their interests in order to discover which products or services are the most likely to be bought, allowing the retailer to understand the buyer's purchase behavior.
- *Cyber Security*. Any activity that compromises the integrity and confidentiality of an information is a security threat. The defensive measures to prevent a data breach includes different procedures, that are currently benefiting from KDD applications, like: intrusion detection, access control, user profiling, malware discovery, anomaly detection.
- *Bioinformatics*. As a relatively recent research field, Bioinformatics leaves large space to research activities and it is rich of large data repositories, gathered in biology and in related life sciences areas. These data are actually analyzed via KDD procedures to try to give answers to different problems, e.g., gene identification, protein and gene interaction network reconstruction, protein function inference, disease diagnosis, prognosis and treatment optimization.

Focusing on biomedicine and bioinformatics, typically data exhibit strong complexities (Dugas *et al.*, 2002; Akil *et al.*, 2011; Holzinger, 2012, 2014) making manual analysis hard or even impossible: a big current challenge, in clinical practice and biomedical research is, hence, the information overload (Berghel, 1997; Noone *et al.*, 1998), i.e., the phenomenon which prevents a decision maker from taking any decision or action dealing with a target issue that produces too much information to analyze.

What is a KDD Process?

Summarizing and formalizing what we said in Section "Introduction", we can state that a KDD process is an activity of (semi-) automatic discovery of knowledge, models, patterns and anomalies in large data repositories. Knowledge is the understanding of the phenomenon that produces the observed data; models are mathematical and logic sets of functions describing the data and explaining their behavior; patterns and anomalies are respectively expected regularities and abnormal observations within the data.

The discovered knowledge should have the following properties:

- *Novelty*. The worthiness of a KDD process depends on which knowledge is discovered. An unknown piece of information has a big value in improving a decision making action, while a trivial information is worthless: *the more unexpected it is, the bigger its value*.
- *Utility*. The discovered knowledge should be useful in supporting the decision making process through suggesting possible and sustainable actions.
- *Generality*. The results of a KDD process, related to a target phenomenon, should be used in a repeatable manner when facing similar phenomena. This concept should be applicable also when considering the same phenomenon during time, unless the structure of the phenomenon has been changed.
- *Understandability*. The results of a KDD process should be clear and easy to read by non-expert users. However, there are many real applications that require that even the intermediate steps of the KDD process have to be intelligible by humans.

Why KDD in Medical and Biological Sciences?

Medicine, life science, biology and health care are sciences that span from the microscopic world (atoms, molecules, viruses, bacteria, cells, tissues) to the macroscopic one (organs, patients, population, health care processes) (Holzinger, 2012). The quantity of data they are able to collect is enormous and heterogeneous. Holzinger *et al.* (2014) propose a simple example for giving an order of magnitude: the omics world (e.g., genomics, proteomics, transcriptomics, microbiomics, etc.) produces many Terabytes (1 Tera Byte = 1000 Giga Bytes = 10^{12} Bytes) of genomics data for each individual, while merging, for instance, these data with the proteomics produces Exabytes of data (1 ExaByte = 1,000,000 Tera Byte = 10^{18} Bytes); things exacerbates when involving other kinds of information such as texts in natural language describing patients' status, laboratory and physiological sensor data, health care management processes, medical and biological research data.

Health sciences have always been data intensive (Ranganathan *et al.*, 2011; Kolker *et al.*, 2014), but their size and complexity are constantly growing up, providing future scenarios rich of possibilities. However there are some practical challenges to deal with when trying to get actionable knowledge from these data: heterogeneity, noise, inconsistency and missing information are typical issues (Kim *et al.*, 2003).

For these reasons, all the health sciences are moving towards the knowledge discovery research area trying to borrow or define methodologies, techniques and best practices to address their complex analytical problems. In Section "KDD Examples in Health Sciences" we will provide some examples where KDD helped in this direction.

How to Perform a KDD Process?

Currently, there is no dominant standard methodology for developing a KDD process. One of the most used approach is the Cross Industry Standard Process for Data Mining (CRISP-DM) Chapman *et al.* (2000). CRISP. It is a 6-phase model that describe a widely set of KDD problems:

1. Business understanding,
2. Data understanding,
3. Data preparation,
4. Modeling,
5. Evaluation,
6. Deployment.

The particularity of this methodology is the life cycle: at each phase, if necessary, the process can restart from any of the previous phases.

Business Understanding

The objective of this phase is to determine the business objectives by defining which are the business success criteria. The first step is to make a situation assessment, performing the resource inventory, analyzing the requirements, the assumptions and the constraints of the business activity, and drawing up a cost and benefit report. Once the initial setting is clear, the *Data Mining* goals and criteria must be defined, in other words we have to choose what we expect from the data analysis and how we can judge if the goals were reached. Finally, we have to produce a project plan stating all the objectives and the initial assessment of required resources, tools and techniques.

Data Understanding

The next phase is aiming at becoming acquainted with the available data. Typically, data are stored in different repositories with different structures; for this reason we need to start an initial data collection from all the data sources. Data are characterized by distributions that can be observed by *exploratory analysis* (Tukey, 1977), which is an approach for data analysis that employs a variety of techniques (mostly graphical) in order to build an overview about data insights, discover underlying structures, define important variables and detect anomalies. Some of the typical tools for exploratory data analysis (Chambers *et al.*, 1983) are: charts, graphs, data traces, histograms, probability plots, lag plots, scatter plots, Youden plots, deviation plots and box plots. The final objective of this phase is to verify the quality of the available data in order to understand if mining and business goals are achievable.

Data Preparation

Raw data must be manipulated (Rahm and Do, 2000) to be successfully analyzed. First of all we have to select the portion of the information that allows to reach the mining goals, from the different data sources: usually, data repositories contain much more data than necessary. Then, procedures of data cleaning are very common; this procedures aim at improving the data quality and readability by:

1. Identifying and dealing with outliers,
2. Smoothing out noisy data,

3. Filling in missing values,
4. Correcting inconsistent data,
5. Removing irrelevant/redundant data, in terms of features and samples.

Enriching data with new information could help in a deeper and fruitful data insight understanding. A common practice, at this stage, is the creation of new attributes and data features. General methodologies of new feature creations are:

1. Feature extraction: exploitation of domain-specific knowledge,
2. Mapping data to new space: algebraic transformations that allow a data domain change to better understand their behavior,
3. Aggregation: computing statistical operations over a subset of data,
4. Feature construction: combining two or more features,
5. Data transformation: applying mathematical functions to a subset of features.

Modeling

Prepared data are exploited to feed *Data Mining Models* (Witten and Frank, 2005), whose objective is to find the desired knowledge. At this stage the analyst has to choose which techniques better fit the data in order to achieve the target goals. A *Model Building Design* must be defined, which is a workflow that govern the model generation process and its parameter configuration. This workflow can be a complex graph composed by (parallel) cascades of different models linked like chains and combined at the end of the process.

Literature provides a vast selection of mining techniques able to solve different kind of problems, which are typically:

- *Classification*. It is the problem of identifying the belonging class of a new observation. Classes are a priori known and are characterized by the features of the data they contain. Some common classification techniques are: Naïve Bayes (Zhang, 2004), Bayesian Networks (Friedman *et al.*, 1997), Logistic Regression (Collins *et al.*, 2002), (Deep) Neural Networks (Haykin, 1998; Goodfellow *et al.*, 2016), Support Vector Machines (Steinwart and Christmann, 2008), KNN (Cover and Hart, 2006), Ripper (Cohen, 1995), PART (Frank and Witten, 1998), Decision Trees (Quinlan, 1986), Random Forest (Breiman, 2001).
- *Regression*. Regression is a statistical processes for estimating the relationships among numerical variables. Given a classical data generation function f , such that $\vec{y} = f(\vec{x}, \vec{\theta})$, where y is the output, $\vec{x}$ the input and $\vec{\theta}$ the function parameters, the objective of the regression is to make some assumption about the nature of f and estimate $\vec{\theta}$ only by observing $\vec{y}$ and $\vec{x}$. Some examples of regression are: Linear Regression (Yan and Su, 2009), Multi Linear Regression (Rao *et al.*, 2008) and Ridge Regression (Hoerl and Kennard, 2000).
- *Clustering*. Clustering is a similar but more complex task than Classification. The objective is the same, find the belonging classes (categories) of new observations; the difference consists in the missing knowledge about the classes: categories, their number and their characteristics are unknown. For these reasons, clustering techniques aim at grouping together the data points into clusters, according to some similarity statistics or distance: points that are similar or close should lie in same cluster, while different or distant points should belong to different clusters. Well-known algorithms in this field are: K-Means (Hartigan and Wong, 1979), K-Medoids (Li, 2009), BD-Scan (Ester *et al.*, 1996), Expectation Maximization (Dempster *et al.*, 1977).
- *Association Mining*. The objective of this research area is to find, within a data set, associations between data points or data point types that recur with a statistical significance. In other words, we are wondering if there are some elements in the data that appear together with high frequency. The best known algorithms for finding associations are Apriori (Agrawal and Srikant, 1994) and FP-Growth (Han *et al.*, 2000).
- *Anomaly detection*. Anomaly detection is the identification of items, events or observations which exhibit a strong difference in behavior from the expectation. Typically, data sources are governed by a set of unknown (or hard to be managed) generative processes that can explain the observed phenomena. These processes can be modeled as mathematical distributions (mixture of components) that generate data points. Each distribution is likely to represent a pattern, that is characterized by an expectation and a deviation. However, errors, noise or abnormal events can generate rare data points that do not follow the pattern schema, turning out to be anomalies different from the remaining ("normal") data points, due to an excessive deviation. Finding these points, called outliers, is the objective of the anomaly detection.

Anomaly detection techniques can be divided into two categories:

- Unbalanced classification: given a dataset whose outliers are known, it is possible to define a classification problem whose classes are "is_outlier" and "is_not_outlier", obviously the former has a much smaller size than the latter,
- Deviance from clusters: when outliers are unknown, a reasonable procedure is to find clusters and mark as anomalies all those points that deviate beyond a chosen threshold from the expectation of each cluster.

Popular algorithms are: BD-Scan (Ester *et al.*, 1996), Isolation Forest (Liu *et al.*, 2008) and CBOD (Jiang and An, 2008).

Evaluation

The Mining Workflow is able to produce results that have to be evaluated in order to understand if they match the Mining goals. A simple evaluation protocol, shown in Fig. 1, is described as follows:

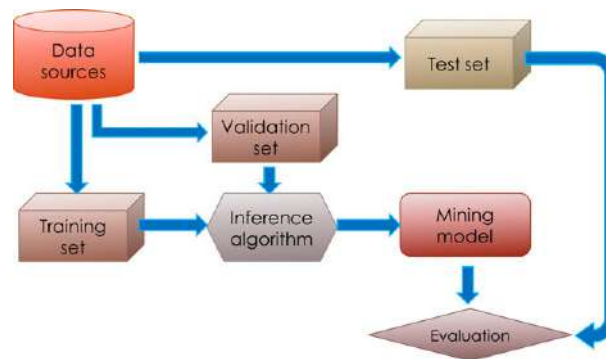


Fig. 1 Simple evaluation workflow.

- From the data sources three datasets are produced: Training set, Validation set and Test set,
- The Training set is provided to an Inference Algorithm whose objective is to build the Mining Model,
- The validation set will help the Inference Algorithm in finding the best algorithm's parameter set trying to avoid configuration of poor quality. Usually, the Validation Set is exploited to compute quality measures during the model building phase and to determine if the algorithm meets its stopping criterion,
- The Evaluation module will apply the Mining Model to the Test Set to generate the final statistics for judging the model quality.

It is important to notice that the sets of data, exploited to compute statistical measures (Test and Validation sets), should be different and independent from the Training Set, for ensuring the fairness of the statistical evaluation.

The evaluation metrics (very often graphical), that better fit business objectives, differ according to the mining goals (Witten and Frank, 2005). For instance, when dealing with classifiers, typical measures are: Confusion matrix, Precision, Recall, F-Measure, Cost, ROC analysis; while for clustering tasks some useful metrics are: Davies-Bouldin index, Dunn index and Silhouette coefficient.

Deployment

When the evaluation phase certifies that the mining goals are achieved, the final step of the CRISP-DM Methodology is the *Model Workflow Deployment*. This is a very crucial step, since in most cases, the end-user is not a data analyst, but the person who commissioned the entire KDD process to the data analysts: a clear and practical documentation is strongly recommended.

The deployment phase must match the business requirements and can range from simple process reports, to lists of useful insights, to complex implementations of the entire mining workflow.

KDD Examples in Health Sciences

One of the first examples of a knowledge discovery process in medical diagnostics is in Reeder and Felson (1977) where authors collected a big list of clinical differential diagnoses in radiology. The drawback here is that the data analysis process was manually performed, limiting the quantity of data that could be analyzed.

Nowadays, (semi-)automatic KDD processes are massively exploited in health sciences under many different perspectives such as semantic data annotation, genomics, protein structure modeling and biological system evolution (Brusic and Zeleznikow, 1999).

Wang *et al.* (2018) propose a recommender system (a cross among association and prediction) able to predict drug-gene-disease associations, breaking the traditional paradigm of "one-drug-one-gene-one-disease", which is not able to tackle diseases involving multiple malfunctioning genes. Thank to this novel technique, Wang *et al.* were able to discover that the diazoxide drug can inhibit malfunctions in multiple kinase genes, leading towards another step to develop a targeted therapy against breast cancer.

In (Wang *et al.*, 2017), KDD and electronic medical records were exploited to improve the hospital treatment quality and increase the patient survival rate. Wang *et al.* defined a convolutional neural networks, fed by patient vital signs and hospitalization history as time series, for the readmission prediction, a well-know problem in literature. An early readmission corresponds to the possibility of an early intervention, preventing dangerous events and reducing healthcare costs.

Monteiro *et al.* (2017) propose machine learning techniques and feature enrichment to improve the prediction of the functional outcome of ischemic stroke patients after three months from the emergency treatment. The motivation of this work lies in the fact that the prognosis is strongly correlated with the early stages of the treatment during the acute phase when a stroke occur.

The last example, we highlight, was proposed by Alhusain and Hafez (2017). They adapted the well-known prediction algorithm Random Forest (Breiman, 2001), to a clustering setting. The objective was to detect interesting patterns in genetic data in

order to address population structure analysis problem, aimed at finding genetic subpopulations based on shared genetic variations of individuals in a population.

Closing Remarks

In this discussion we gave an overview about the Knowledge Discovery in Database science. A general definition of KDD was provided, describing it as an effective set of methodologies, techniques and tools for data analysis aimed at unearthing useful knowledge buried within large amounts of information. We explored the principal KDD's areas, categorizing them according to their capabilities and objectives. A well-known methodology, for managing KDD processes, was presented, namely CRISP-DM. Each phase of this methodology has been discussed and enriched with definitions and bibliographic references for further and deeper readings, hoping to have intrigued the reader for continuing to study this fascinating discipline.

See also: Biological Database Searching. Knowledge and Reasoning. Mapping the Environmental Microbiome. Stochastic Methods for Global Optimization and Problem Solving

References

- Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487–499. San Francisco, CA: Morgan Kaufmann Publishers Inc.
- Akil, H., Martone, M.E., Van Essen, D.C., 2011. Challenges and opportunities in mining neuroscience data. *Science* 331, 708–712.
- Alhusain, L., Hafez, A.M., 2017. Cluster ensemble based on random forests for genetic data. *BioData Mining* 10, 37.
- Berghel, H., 1997. Cyberspace 2000: Dealing with information overload. *Communications of the ACM* 40, 19–24.
- Breiman, L., 2001. Random forests. *Machine Learning* 45, 5–32.
- Brusic, V., Zeleznikow, J., 1999. Knowledge discovery and data mining in biological databases. *The Knowledge Engineering Review* 14, 257–277.
- Chambers, J., Cleveland, W., Kleiner, B., Tukey, P., 1983. *Graphical methods for data analysis*. Wadsworth.
- Chapman, P., Clinton, J., Kerber, R., *et al.*, 2000. CRISP-DM 1.0 step-by-step data mining guide. In: *Proceedings of the Technical Report. The CRISP-DM consortium*. Available at: <http://www.crisp-dm.org/CRISPWP-0800.pdf>.
- Cohen, W.W., 1995. Fast effective rule induction. In: *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115–123. Morgan Kaufmann.
- Collins, M., Schapire, R.E., Singer, Y., 2002. Logistic regression, adaboost and bregman distances. *Machine Learning* 48, 253–285.
- Cover, T., Hart, P., 2006. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 21–27.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B* 39, 1–38.
- Dugas, M., Hoffmann, E., Janko, S., *et al.*, 2002. Complexity of biomedical data models in cardiology: The intranet-based AF registry. *Computer Methods and Programs in Biomedicine* 68, 49–61.
- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231. AAAI Press.
- Fayyad, U., Uthurusamy, R., 1996. Data mining and knowledge discovery in databases. *Communications of the ACM* 39, 24–26.
- Frank, E., Witten, I.H., 1998. Generating accurate rule sets without global optimization. In: Shavlik, J. (Ed.), *Fifteenth International Conference on Machine Learning*. Morgan Kaufmann, pp. 144–151.
- Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. *Machine Learning* 29, 131–163.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. The MIT Press.
- Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 1–12. New York, NY, USA: ACM.
- Hartigan, J.A., Wong, M.A., 1979. A k-means clustering algorithm. *JSTOR: Applied Statistics* 28, 100–108.
- Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation*, second ed. Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Hoerl, A.E., Kennard, R.W., 2000. Ridge regression: Biased estimation for non orthogonal problems. *Technometrics* 42, 80–86.
- Holzinger, A., 2012. Biomedical informatics: Computational sciences meets life sciences. *BoD*.
- Holzinger, A., 2014. *Biomedical Informatics: Discovering knowledge in Big Data*. Springer.
- Holzinger, A., Dehmer, M., Jurisica, I., 2014. Knowledge discovery and interactive data mining in bioinformatics - state-of-the-art, future challenges and research directions. *BMC Bioinformatics* 15, 11.
- Jiang, S., An, Q., 2008. Clustering-based outlier detection method. In: *Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 429–433. Shan-dong, China.
- Kim, W., Choi, B., Hong, E., Kim, S., Lee, D., 2003. A taxonomy of dirty data. *Data Mining and Knowledge Discovery* 7, 81–99.
- Kolker, E., Özdemir, V., Martens, L., *et al.*, 2014. Toward more transparent and reproducible omics studies through a common metadata checklist and data publications. *OMICS: A Journal of Integrative Biology* 18, 10–14.
- Li, X., 2009. K-means and k-medoids. In: Liu, L., Oszu, M.T. (Eds.), *Encyclopedia of Database Systems*. US: Springer, pp. 1588–1589.
- Liu, F.T., Ting, K.M., Zhou, Z.H., 2008. Isolation forest. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, IEEE Computer Society, pp. 413–422. Washington, DC.
- Monteiro, M., Fonseca, A.C., Freitas, A.T., *et al.*, 2017. Improving the prediction of functional outcome in ischemic stroke patients. In: *Proceedings of International Workshop on Data Mining in Bioinformatics (BIOKDD)*, p. 5.
- Noone, J., Warren, J.R., Brittain, M., 1998. Information overload: Opportunities and challenges for the gp's desktop. In: Cesnik, B., McCray, A.T., Scherrer, J. (Eds.), *MEDINFO '98 – 9th World Congress on Medical Informatics*. IOS Press, pp. 1287–1291.
- Quinlan, J.R., 1986. Induction of decision trees. *Machine Learning* 1, 81–106.
- Rahm, E., Do, H.H., 2000. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin* 23, 3–13.

- Ranganathan, S., Schönbach, C., Kelso, J., *et al.*, 2011. Towards big data science in the decade ahead from ten years of incob and the 1st iscb-asia joint conference. *BMC Bioinformatics* 12, S1.
- Rao, C., Toutenburg, H., Shalabh, Heumann, C., 2008. *The Multiple Linear Regression Model and its Extensions*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 33–141.
- Reeder, M., Felson, B., 1977. *Gamuts in Radiology: Comprehensive Lists of Roentgen Differential Diagnosis*. Audiovisual Radiology of Cincinnati.
- Steinwart, I., Christmann, A., 2008. *Support Vector Machines*, first ed. Springer Publishing Company, Incorporated.
- Tukey, J.W., 1977. *Exploratory Data Analysis*. Addison-Wesley.
- Wang, A., Lim, H., Cheng, S.Y., Xie, L., 2018. Antenna, a multi-rank, multi-layered recommender system for inferring reliable drug-gene-disease associations: Repurposing diazoxide as a targeted anti-cancer therapy, p.1.
- Wang, H., Cui, Z., Chen, Y., *et al.*, 2017. Predicting hospital readmission via cost-sensitive deep learning. In: *Proceedings of the Transactions on Computational Biology and Bioinformatics*, p. to appear.
- Witten, I.H., Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition (Morgan Kaufmann Series in Data Management Systems). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Yan, X., Su, X.G., 2009. *Linear Regression Analysis: Theory and Computing*. River Edge, NJ, USA: World Scientific Publishing Co., Inc.
- Zhang, H., 2004. The optimality of naive bayes. In: *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, AAAI Press.

Supervised Learning: Classification

Mauro Castelli, Leonardo Vanneschi, and Álvaro Rubio Largo, NOVA IMS, Universidade Nova de Lisboa, Lisboa, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the field of machine learning (ML) it is possible to distinguish two fundamental categories: supervised learning and unsupervised learning techniques. The main aim of the former category is the search for algorithms that reason from externally supplied observations (also called training instances) to produce general hypotheses, which then make predictions about future instances (Kotsiantis *et al.*, 2007). One important area of supervised machine learning relates to classification problems, a task that is usually performed in a supervised setting. Given k possible class labels, a training set of pairs (x_i, y_i) with $x_i \in \mathbb{X}^M$ denoting a training instance described by M features and $y_i \in 1, \dots, k$ representing the correct class label for the observation x_i , the objective of classification is to build a model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances (also called unseen instances) where the values of the predictor features are known, but the value of the class label is unknown. According to the number of possible labels, it is possible to distinguish binary classification problems and multi-class classification problems. In the former case, only two class labels exist, while in the latter the number of class labels is greater than 2. The aim of this article is to introduce the reader to supervised machine learning methods for classification. In particular, Section “Classification problems in Bioinformatics” reports some of the most important challenges in the field of Bioinformatics that can be solved as classification problems. Subsequently, Section “Methods” presents some of the existing supervised methods used to address a classification task, while Section “Performance Evaluation” presents measures that are commonly employed to evaluate the performance of a classifier. Section “Current research and Open Issues” outlines some recent advances and challenges in the area of supervised learning for classification problems, while Section “Conclusions” summarizes the main concepts presented in the article, suggesting further documents for more advanced concepts.

Classification Problems in Bioinformatics

Classification is a fundamental task in Bioinformatics, and a plethora of work using ML techniques to address the classification task has been proposed. While a complete review of the existing works is beyond the scope of this article, this section reports the main application of supervised machine learning techniques to address classification problems in the field of Bioinformatics.

The most known application of classification methods related to microarray data. In Díaz-Uriarte and Alvarez de Andrés (2006), authors propose the use of random forest (Breiman, 2001) for the classification of microarray data. In detail, authors investigate the use of random forest for classification of microarray data (including multi-class problems) and propose a new method of gene selection in classification problems based on random forest. Using simulated and nine microarray data sets they show that random forest has comparable performance to other classification methods and that the proposed gene selection procedure yields very small sets of genes (smaller than alternative methods) while preserving predictive accuracy.

In Furey *et al.* (2000), authors have developed a method to analyze microarray data using support vector machines (SVMs). This analysis consists of both classification of the tissue samples and an exploration of the data for mislabelled or questionable tissue results. The method is tested on samples consisting of ovarian cancer tissues, normal ovarian tissues, and other normal tissues. As a result of computational analysis, a tissue sample is discovered and confirmed to be wrongly labeled. To show the robustness of the SVM method, authors analyzed two previously published datasets from other types of tissues or cells, achieving comparable results.

An extensive comparison of classification tools applied to microarray data is presented in Lee *et al.* (2005). In this paper, authors evaluate the performance of different classification methods in microarray experiment, and provide the guidelines for finding the most appropriate classification tools in various situations. The comparison considers different combinations of classification methods, datasets and gene selection techniques. Among the different classification methods, there are random forest, classification and regression tree (CART) (Breiman *et al.*, 1984), bagging (Quinlan *et al.*, 1996), boosting (Quinlan *et al.*, 1996), SVMs and artificial neural networks (ANNs) (Patterson, 1998). The comparison study shows several interesting facts, providing some insights into the classification tools in microarray data analysis. In particular, the study shows that aggregating classifiers such as bagging, boosting and random forest improve the performance of CART significantly, being random forest the most excellent method among the tree methods when the number of classes is moderate. The experimental study also shows that SVM gives the best performance among the machine learning methods in most datasets regardless of the gene selection.

A review of other studies where supervised ML methods have been applied to the classification of genomic data can be found in Díaz-Uriarte (2005).

Other applications of supervised ML classification techniques in Bioinformatics include the identification of splice site (Zhang *et al.*, 2006), classification of proteins according to their function (Ding and Dubchak, 2001), the prediction of nucleosome positioning from primary DNA sequence (Peckham *et al.*, 2007), mass spectrometry analysis (Sauer *et al.*, 2008), classification of human genomic regions (Yip *et al.*, 2012).

Methods

This section is dedicated to the description of the most popular and used classification techniques. Specifically, the following techniques are outlined: decision trees, random forest, bagging and boosting, support vector machines, artificial neural networks. While the list does not cover all the existing methods, the presented techniques are by far the ones that are widely used in the Bioinformatics field. The interested reader can find additional information in the reference work of [Duda et al. \(2012\)](#).

Decision Tree

As reported in [Murthy \(1998\)](#), decision trees are a way to represent rules underlying data with hierarchical, sequential structures that recursively partition the data. Decision tree learning is used to approximate discrete-valued target functions, in which the learned function is represented by a decision tree. Learned trees can also be re-represented as sets of if-then rules to improve human readability ([Michalski et al., 2013](#)).

In particular, a decision tree for a classification problem can be represented in the form of a tree structure, where each node of the tree can be a leaf node or a decision node. A leaf node indicates the value of the target attribute (class), while a decision node specifies some test to be performed on a single feature of the available observation, with one branch for each possible outcome of the test. The process of classifying a given instance by means of a decision tree starts by evaluating of the test contained in the root node (a decision node) and moving through it until a leaf node, which provides the classification of the instance.

The most commonly used algorithm to build decision trees is the C4.5 proposed in [Quinlan \(1993\)](#) as an extension of the ID3 algorithm proposed in [Quinlan \(1993\)](#). The tree is constructed in a top-down recursive divide-and-conquer manner. At start, all the training examples are at the root and the attributes (features) are categorical (if continuous-valued, they are discretized in advance). Subsequently, samples are partitioned recursively based on selected attributes. After the partitioning step, test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain [Michalski et al. \(2013\)](#)). The algorithm ends when one of the following termination conditions holds: all samples for a given node belong to the same class, there are no remaining attributes for further partitioning, there are no samples left.

The pseudocode of the algorithm DecisionTreeBuilding (S, A) is the following, where S is the set of training instances and A the list of features:

1. Create a node N ;
2. If all samples belong to the same class C then label N with C and terminate;
3. If A is empty then label N with the most common class C in S (majority voting) and terminate;
4. Select $a \in A$, with the highest information gain and label N with a ;
5. For each value v of a :
 - (a) Grow a branch from N with condition $a=v$;
 - (b) Let S_v be the subset of instances in S with $a=v$;
 - (c) If S_v is empty then attach a leaf labeled with the most common class in S , else attach the node generated by DecisionTreeBuilding ($S_v, A - a$).

A decision tree can be used for data exploration to achieve the following objectives: (1) to reduce a volume of data by transforming it into a more compact form which preserves the essential characteristics and provides an accurate summary; (2) to discover whether the data contains well-separated classes of objects, such that the classes can be interpreted meaningfully in the context of a substantive theory; (3) to uncover a mapping from independent to dependent variables that is useful for predicting the value of the dependent variable in the future.

While decision trees are able to generate human-understandable rules without requiring much computational effort, they present some disadvantages. In particular, as reported in [Michalski et al. \(2013\)](#), practical issues in learning decision trees include handling continuous attributes, choosing an appropriate attribute selection measure, handling training data with missing attribute values, handling attributes with differing costs, avoiding overfitting of the learned model (i.e., model that is overspecialised on training instances).

Bagging and Boosting

Bagging and Boosting ([Quinlan et al., 1996](#)) are ensemble techniques that combine several machine learning methods into one predictive model in order to decrease the variance (bagging) or the bias (boosting). Both of them consist of the following steps: (1) producing a distribution of simple ML models on subsets of the original data; (2) combining the distribution into one "aggregated" model.

Bagging (whose name stands for Bootstrap Aggregating) starts by creating random samples of the training data set (using combinations with repetitions to produce multisets of the same cardinality of the original data). Then, it builds a classifier for each sample and, finally, results of these multiple classifiers are combined using average or majority voting. Bagging builds each model independently and, for this reason, it is considered as a parallel ensemble method. By increasing the size of the training set (i.e., considering different samples of the same cardinality) bagging decreases the variance of the final model,

tuning the prediction to the expected outcome (Bauer and Kohavi, 1999). As such, bagging is particularly useful when a model overfits the training instances. An example of bagging that makes use of tree models is represented by random forests (Breiman, 2001).

On the other, boosting is based on a two-step approach, where one first uses subsets of the original data to produce a series of averagely performing models and then “boosts” their performance by combining them together using a particular cost function (majority vote). Unlike bagging, in the classical boosting the subset creation is not random and depends upon the performance of the previous models: every new subset contains the elements that were (likely to be) misclassified by previous models (Bauer and Kohavi, 1999). Hence, differently from bagging, boosting provides sequential learning of the models, where the first model is learned on the whole data set, while the following are learned on the training set based on the performance of the previous one. More in detail, it starts by classifying original data set and giving equal weights to each observation. If classes are predicted incorrectly using the first learner, then it gives higher weight to the missed classified observation. Being an iterative process, it continues to add classifier learner until a limit is reached in the number of models or accuracy. Boosting has shown better predictive accuracy than bagging, but it also tends to overfit the training data. An example of this ensemble method is the AdaBoost algorithm (Rätsch *et al.*, 2001).

Random Forest

Random forest (Breiman, 2001) is an ensemble approach that can be used to perform both classification and regression tasks. The main principle behind ensemble methods is that a group of weak learners can be joined to form a strong learner. That is, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

As reported in Breiman (2001), random forests are a combination of tree predictors (i.e., a decision tree which, in ensemble terms, corresponds to the weak learner) such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Hence, random forest adds an additional level of randomness with respect to bagging. More in detail, in addition to constructing each tree using a different bootstrap sample of the data, random forests change how the classification trees are constructed. While in standard trees each node is split using the best split among all variables, in a random forest each node is split using the best among a subset of predictors randomly chosen at that node (Liaw and Wiener, 2002). While this strategy looks counterintuitive, it turns out to perform very well compared to many other classifiers and is robust against overfitting (Breiman, 2001).

Given N training samples and M variables and being B the number of decision trees in the forest, each one of the B trees is constructed as follows:

1. Choose a training set for this tree by choosing n times with replacement from all N available training cases (i.e., take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes;
2. For each node of the tree, randomly choose m variables (m should be much less than M) on which to base the decision at that node. Calculate the best split based on these m variables in the training set;
3. Each tree is fully grown and not pruned (as may be done in constructing a normal decision tree classifier).

Hence, the only difference with respect to bagging is that instead of using all the M variables in a decision node, only a small subset of them is considered to calculate the best split of the data. When a new instance is entered into the model, it is run down all of the trees. For prediction, a new sample is pushed down all the trees and the result (the class label for the instance), is obtained by means of a voting majority that considers the output of all the weak learners.

While random forests only require specifying two parameters (the number of variables in the random subset at each node and the number of trees in the forest) and the performance they achieved is usually not very sensitive to their values, they still present some disadvantages. In particular, the final model consists of a set of decision trees and it is not easy to interpret for a human-being. That is, random forest is a predictive modeling tool, not a descriptive tool. Hence, if a description of the relationships existing in the available data is fundamental, random forests are not a viable option to be considered.

Support Vector Machine

Support Vector Machines (Vapnik and Vapnik, 1998) are a set of supervised learning methods used for classification and regression. In the case of a classification task, given a set of data points in a n -dimensional space, each belonging to one of the possible classes, SVM aims at finding the separating hyperplanes that maximize the margin between sets of data. This should ensure a good generalization ability of the method, under the hypothesis of consistent target function between training and testing data. To calculate the margin between data belonging to two different classes (i.e., binary classification task), two parallel hyperplanes are constructed (one on each side of the separating hyperplane) and “pushed up against” the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighbouring data points of both classes, since in general the larger the margin the lower the generalization error of the classifier. The parameters of the maximum-margin hyperplane are derived by solving large quadratic programming (QP) optimization problems. As reported in

Vapnik and Vapnik (1998), there exist several specialized algorithms for quickly solving these problems that arise from SVMs, mostly reliant on heuristics for breaking the problem down into smaller, more manageable chunks (Platt, 1999).

While the original problem may be stated in a finite dimensional space, it often happens that in the original n -dimensional space the sets to be discriminated are not linearly separable. In this case, the original finite dimensional space is mapped into a much higher-dimensional space, in the hope that in this new space the data could become more easily separated or better structured. To achieve this, SVMs use a mapping into a larger space by means of a kernel function (Hofmann *et al.*, 2008) $K(x, y) = \langle f(x), f(y) \rangle$. Here K is the kernel function, x, y are n -dimensional inputs. f is a map from n -dimensional to m -dimensional space and $\langle x, y \rangle$ denotes the dot product. Usually, m is much larger than n . This mapping function, however, does not need to be computed because of a tool called the kernel trick: the idea is to define K in terms of original space itself without even defining what the transformation function f is (Hofmann *et al.*, 2008).

Artificial Neural Networks

Artificial neural networks were initially developed according to the observation that biological learning systems are built of very complex webs of interconnected neurons. While a large variety of networks have been proposed in the literature (Haykin and Network, 2004) the main structure shared by the different models is the one in which the network is composed of units (also called neurons), and connections between them, which together determine the behaviour of the network. The choice of the network type depends on the problem to be solved; the backpropagation gradient network is the most frequently used (Hecht-Nielsen *et al.*, 1988). As described in Reby *et al.* (1997), this network consists of three or more neuron layers: one input layer, one output layer, and at least one hidden layer. In most cases, a network with only one hidden layer is used to restrict calculation time, especially when the results obtained are satisfactory. All the neurons of each layer (except the neurons of the last one) are connected by an axon to each neuron of the next layer.

The back propagation algorithm is commonly used for training artificial neural networks (Haykin and Network, 2004). The training is usually performed by an iterative updating of the weights based on the error signal. This error is calculated, in the output layer, as the difference between the true class and the actual output values, multiplied by the slope of a sigmoidal activation function. Then the error signal is back-propagated to the lower layers. Under this light, back propagation is a descent algorithm, which attempts to minimize the error at each iteration. The weights of the network are adjusted by the learning algorithm such that the error is decreased along a descent direction. Traditionally, two parameters, called learning rate and momentum factor, are used for controlling the weight adjustment along the descent direction and for dampening oscillations (Hecht-Nielsen *et al.*, 1988). The training of a network with the back propagation algorithm can be summarized as follows:

1. Initialize the input weights for all neurons to some random numbers between 0 and 1;
2. Apply input to the network (i.e., provide one training instance to the network);
3. Calculate the output;
4. Compare the resulting output with the desired one (target class label) for the given input;
5. Modify the weights and threshold for all neurons using the error;
6. Repeat the process until the error reaches an acceptable value (e.g., accuracy is greater than a given threshold).

For all the details related to the back propagation algorithm and for a complete review of the existing networks, the reader is referred to Haykin and Network (2004).

ANNs have been used to perform both binary (Zhang, 2000) neural and multi-class classification (Ou and Murphey, 2007). In particular, Ou and Murphey (2007) presented an in-depth study on multi-class pattern classification using neural networks. Specifically, they discussed two different architectures, systems of multiple neural networks and single neural network systems, and three types of modeling approaches. The study shows that these different systems have their own strength over different application problems along the directions of system generalization within feature space, incremental class learning, learning from imbalanced data, large number of pattern classes, and small and large training data.

A complete guide to the design of ANNs to address classification problems can be found in Demuth *et al.* (2014), where the aforementioned issues are discussed.

Performance Evaluation

Once one of the existing techniques is selected to build a classifier, it is fundamental to evaluate its performance. Several measures have been defined to evaluate the performance of a classifier and this section presents the ones that are commonly used.

The discussion starts by considering binary classification problems and the related performance measures. After this presentation, measures used to evaluate multi-class classifiers are presented.

In a binary classification problem, each observation belongs to one element of the set $\{p, n\}$, where p denotes the positive class and n denotes the negative class. In this scenario, given a classifier and an instance, there are four possible outcomes. If the instance is positive and it is classified as positive, it is counted as a true positive (TP); if it is classified as negative, it is counted as a false negative (FN). If the instance is negative and it is classified as negative, it is counted as a true negative (TN); if it is classified as

Table 1 Confusion matrix. Across the top are the true class labels and down the side are the predicted class labels. Each cell contains the number of predictions made by the classifier that fall into that cell

		True class	
		p	n
Predicted class	p	True Positives	False Positives
	n	False Negatives	True Negatives

positive, it is counted as a false positive (FP) (Fawcett, 2006). Given a classifier and a set of observations, it is possible to build a two-by-two confusion matrix representing the dispositions of the set of instances. Table 1 represents a confusion matrix. Across the top are the true class labels and down the side are the predicted class labels. Each cell contains the number of predictions made by the classifier that fall into that cell.

Based on the information reported in the confusion matrix it is possible to derive the most often used measures for binary classification performance evaluation. More in detail, the numbers along the major diagonal of the confusion matrix represent the correct decisions made.

The measure that is commonly used is accuracy. Accuracy is defined as the number of correct predictions made divided by the total number of predictions made. Hence, $Accuracy = \frac{TP+TN}{TP+TN+FN+FP}$. Hence, accuracy represents the overall effectiveness of a classifier. Anyway, only considering the accuracy as a performance measure can lead to some misleading interpretation about the classifier. This is the case of a dataset that is very unbalanced (i.e., the observations belonging to one class are much more than the ones belonging to the second class). Considering a dataset where the 90% of the instances belongs to the class p , a classifier that classifies all the instances (also the one belonging to the class n) as belonging to class p reaches a 90% of accuracy, even though such a classifier is useless. To avoid this issue, other measures are also considered.

Precision is defined as the number of true positives divided by the number of true positives and false positives. Thus, $Precision = \frac{TP}{TP+FP}$. In other words, it is the number of positive predictions divided by the total number of positive class values predicted. Precision represents class agreement of the data labels with the positive labels given by the classifier. Hence, it can be thought of as a measure of a classifiers exactness.

Another measure that is typically used in conjunction with precision is the recall measure. Recall is defined as the number of true positives divided by the number of true positives and the number of false negatives. Hence, $Recall = \frac{TP}{TP+FN}$. Recall is used to express the effectiveness of a classifier to identify positive labels.

In order to evaluate the performance of a classifier with a measure that considers both precision and recall, it is possible to make use of the F_{score} . Generally speaking, F_{score} represents the relations between data's positive labels and those given by a classifier. F_{score} is defined as follows: $F_{score} = \frac{(\beta^2+1)TP}{(\beta^2+1)TP+\beta^2FN+FP}$ where β is a positive real constant. The F_{score} that is commonly used is the so-called $F1_{score}$ where the value of β is equal to 1. The $F1_{score}$ score can be interpreted as a weighted average of the precision and recall and it assumes a maximum value of 1 if and only if both precision and recall are equal to 1.

To introduce the AUC (area under the curve) measure, it is necessary to introduce the concept of ROC (Reception Operating Characteristic) curve. As reported in Fawcett (2006), ROC graphs are two-dimensional graphs in which TP rate ($\frac{TP}{TP+FN}$) is plotted on the Y axis and FP rate ($\frac{FP}{FP+TN}$) is plotted on the X axis. A ROC graph depicts relative tradeoffs between benefits (true positives) and costs (false positives). An example of a ROC graph is reported in Fig. 1 where each letter corresponds to a different classifier.

In the ROC space the point (0,0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives. The opposite strategy, of unconditionally issuing positive classifications, is represented by the upper right point (1,1) (Fawcett, 2006). The point (0,1) represents perfect classification. In order to combine FP rate and the TP rate into a single metric the following steps are performed: firstly the two former metrics are computed considering different threshold (e.g., from 0 to 1 with a step of 0.01), then the corresponding points are plotted on a single graph in the ROC space forming the ROC curve. The resulting metric is the AUC of this curve, which is also referred as AUROC. The value of the AUC ranges between 0 and 1 and the classifier's performance is as better as the AUC is different from 0.5 (this is the AUC of a random classifier).

The performance measures introduce for binary classification problems can be extended also to multi-class classification tasks. The idea is that for an individual class C_i , the assessment is defined by TP_i , FN_i , TN_i , FP_i , $Accuracy_i$, $Precision_i$, $Recall_i$. As reported in Sokolova and Lapalme (2009), the quality of the overall classification is usually assessed in two ways: a measure is the average of the same measures calculated for the possible class labels $C_1, \dots, C_j$ (macro-averaging). The second option is to consider the sum of counts to obtain cumulative TP, FN, TN, FP and then calculating a performance measure (micro-averaging). Macro-averaging treats all classes equally while micro-averaging favors bigger classes. As observed in Lachiche and Flach (2003) as there is no well-developed multi-class ROC analysis, AUC is typically not used to evaluate multi-classification performance.

For a complete review of existing measures for evaluating classifiers' performance, the reader is referred to Sokolova and Lapalme (2009), where authors analyze 24 performance measures used in the complete spectrum of ML classification tasks: binary, multi-class, multi-labeled, and hierarchical. In particular, the paper explains how different evaluation measures assess different characteristics of machine learning algorithms and choosing the correct measure is not a simple task.

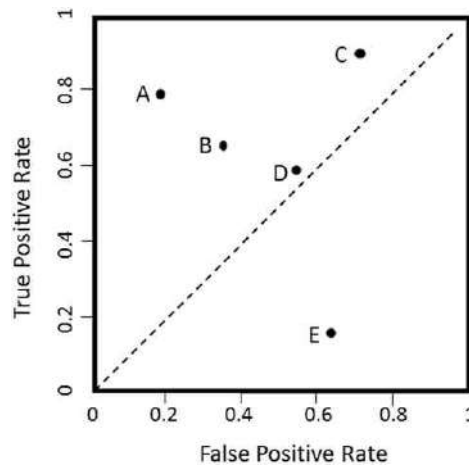


Fig. 1 Example of ROC curve.

Current Research and Open Issues

Despite the large number of techniques available to address a classification problem, researchers still continue to investigate the usage of more advanced methods to tackle this task. This section reports current research directions and open issues in this ML area.

A hot research topic is related to the use of evolutionary computation (Bäck *et al.*, 1997) to address classification problems. In fact, while evolutionary computation has been used to successfully solve a plethora of real world problems, its performance to perform multi-class classification is generally poorer with respect to the one produced by other classifiers. Under this light, the work of Ingalalli *et al.* (2014) represents a first attempt to improve the performance of genetic programming (GP) (Koza, 1992) in addressing multi-class classification tasks. GP is one of the youngest techniques inside the family of evolutionary computation and, while it has shown its ability in producing human-competitive results Koza (2010) human, its performance on multi-class problems was not competitive.

The basic idea of the M2GP (Multidimensional Multiclass Genetic Programming) system proposed by Ingalalli *et al.* (2014) is to find a transformation, such that the transformed data of each class can be grouped into unique clusters. In M2GP the number of dimensions in which the clustering is performed is completely independent of the number of classes, such that high dimensional datasets may be easily classified by a low dimensional clustering, while low dimensional datasets may be better classified by a high dimensional clustering (Silva *et al.*, 2016). To achieve this, M2GP uses a representation that is basically the same used for regular tree-based GP, except that the root node of the tree exists only to define the number of dimensions d of the new space. As reported in Silva *et al.* (2016), each candidate classifier is evaluated as follows: (1) all the n instances of the training set are mapped into the new d -dimensional space (each branch of the tree is one of the d dimensions); (2) on this new space, for each of the M classes in the data, the covariance matrix and the cluster centroid is calculated from the instances belonging to that class; (3) the Mahalanobis distance between each sample and each of the M centroids is calculated. Each sample is assigned the class whose centroid is closer.

This system, that has been subsequently extended to an improved method presented in Muñoz *et al.* (2015), has produced, on a large set of benchmarks, performance that are better than or comparable to the ones achieved by random forests, the technique that is able to generally produce the best accuracy with respect to other ML techniques (Fernández-Delgado *et al.*, 2014).

While this example underlines that ML for classification problems is still an open research topic, new challenges should be addressed in connection with the field of Big Data. A list of the most important issues and challenges is reported in Krawczyk (2016) and Kreml *et al.* (2014).

Conclusions

Supervised methods for classification represent an important sub-area of machine learning and they are a valuable tool for addressing problems over different domains. In the field of Bioinformatics, a large number of problems can be formulated as a classification task and successfully solved by means of the existing supervised methods. This article has presented some of the most relevant applications in this field where classification problems arise, describing the most well-known techniques to solve them. Moreover, different measures to evaluate the performance of a classifier have been introduced, showing that, generally, one single measure is not enough to get a meaningful inside about the performance produced by a classification model. The final part of the article was dedicated to recent advances in the field, providing some pointers to the most promising works in the area of machine learning methods for classification problems and discussing challenges and open issues in this field. In particular, one of the most challenging areas is the one related to the development of supervised methods to handle and classify Big Data.

While the provided information represents a gentle introduction to supervised methods for classification, the reader can find a more comprehensive and detailed discussion of the topics covered in this article in Further Reading section.

Acknowledgment

Álvaro Rubio-Largo is supported by the post-doctoral fellowship SFRH/BPD/100872/2014 granted by Fundação para a Ciência e a Tecnologia (FCT), Portugal.

See also: Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Data Mining in Bioinformatics. The Challenge of Privacy in the Cloud. Visualization of Biomedical Networks

References

- Bäck, T., Fogel, D.B., Michalewicz, Z., 1997. Handbook of Evolutionary Computation. New York: Oxford.
- Bauer, E., Kohavi, R., 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* 36 (1–2), 105–139.
- Breiman, L., 2001. Random forests. *Machine Learning* 45 (1), 5–32.
- Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. *Classification and Regression Trees*. Monterey, CA: Wadsworth & Brooks.
- Demuth, H.B., Beale, M.H., De Jess, O., Hagan, M.T., 2014. *Neural Network Design*. Martin Hagan.
- Díaz-Uriarte, R., 2005. Supervised methods with genomic data: A review and cautionary view. *Data Analysis and Visualization in Genomics and Proteomics*. New York: Wiley, pp. 193–214.
- Díaz-Uriarte, R., Alvarez de Andres, S., 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7 (1), 3.
- Ding, C.H., Dubchak, I., 2001. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* 17 (4), 349–358.
- Duda, R.O., Hart, P.E., Stork, D.G., 2012. *Pattern Classification*. John Wiley & Sons.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27 (8), 861–874.
- Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D., 2014. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research* 15 (1), 3133–3181.
- Furey, T.S., Cristianini, N., Duffy, N., *et al.*, 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16 (10), 906–914.
- Haykin, S., 2004. A comprehensive foundation. *Neural Networks* 2 (2004), 41.
- Hecht-Nielsen, R., *et al.*, 1988. Theory of the backpropagation neural network. *Neural Networks* 1 (Suppl 1), 445–448.
- Hofmann, T., Schölkopf, B., Smola, A.J., 2008. Kernel methods in machine learning. *The Annals of Statistics*, 1171–1220.
- Ingalalli, V., Silva, S., Castelli, M., Vanneschi, L., 2014. A multi-dimensional genetic programming approach for multi-class classification problems.
- Kotsiantis, S.B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques.
- Koza, J., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Bradford: A Bradford book.
- Koza, J.R., 2010. Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines* 11 (3–4), 251–284.
- Krawczyk, B., 2016. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence* 5 (4), 221–232.
- Krempel, G., Žliobaite, I., Brzeziński, D., *et al.*, 2014. Open challenges for data stream mining research. *ACM SIGKDD Explorations Newsletter* 16 (1), 1–10.
- Lachiche, N., Flach, P., 2003. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using roc curves. *ICML*, 416–423.
- Lee, J.W., Lee, J.B., Park, M., Song, S.H., 2005. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis* 48 (4), 869–885.
- Liaw, A., Wiener, M., 2002. Classification and regression by randomforest. *R News* 2 (3), 18–22.
- Michalski, R.S., Carbonell, J.G., Mitchell, T.M., 2013. *Machine learning: An Artificial Intelligence Approach*. Springer Science & Business Media.
- Muñoz, L., Silva, S., Trujillo, L., 2015. *M3GP – Multiclass Classification with GP*. Springer International Publishing.
- Murthy, S.K., 1998. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery* 2 (4), 345–389.
- Ou, G., Murphey, Y.L., 2007. Multi-class pattern classification using neural networks. *Pattern Recognition* 40 (1), 4–18.
- Patterson, D.W., 1998. *Artificial Neural Networks: Theory and Applications*. Prentice Hall PTR.
- Peckham, H.E., Thurman, R.E., Fu, Y., *et al.*, 2007. Nucleosome positioning signals in genomic dna. *Genome Research* 17 (8), 1170–1177.
- Platt, J.C., 1999. 12 fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, 185–208.
- Quinlan, J.R., 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Quinlan, J.R., *et al.*, 1996. Bagging, boosting, and c4. 5. *AAAI/IAAI vol. 1*, 725–730.
- Rätsch, G., Onoda, T., Müller, K.-R., 2001. Soft margins for adaboost. *Machine Learning* 42 (3), 287–320.
- Reby, D., Lek, S., Dimopoulos, I., *et al.*, 1997. Artificial neural networks as a classification method in the behavioural sciences. *Behavioural Processes* 40 (1), 35–43.
- Sauer, S., Freiwald, A., Maier, T., *et al.*, 2008. Classification and identification of bacteria by mass spectrometry and computational analysis. *PLoS One* 3 (7), e2843.
- Silva, S., Muñoz, L., Trujillo, L., *et al.*, 2016. Multiclass classification through multidimensional clustering. *Genetic Programming Theory and Practice, XIII*. Springer. pp. 219–239.
- Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management* 45 (4), 427–437.
- Vapnik, V.N., Vapnik, V., 1998. *Statistical Learning Theory*, vol. 1. New York: Wiley.
- Yip, K.Y., Cheng, C., Bhardwaj, N., *et al.*, 2012. Classification of human genomic regions based on experimentally determined binding sites of more than 100 transcription-related factors. *Genome Biology* 13 (9), R48.
- Zhang, G.P., 2000. Neural networks for classification: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30 (4), 451–462.
- Zhang, Y., Chu, C.-H., Chen, Y., Zha, H., Ji, X., 2006. Splice site prediction using support vector machines with a bayes kernel. *Expert Systems with Applications* 30 (1), 73–81. (Intelligent Bioinformatics Systems).

Further Reading

- Aggarwal, C.C., 2014. Data Classification: Algorithms and Applications, first ed. Chapman & Hall/CRC.
- Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A., Nielsen, H., 2000. Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics* 16 (5), 412–424.
- Dietterich, T.G., 2000. Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems. Berlin/Heidelberg: Springer. pp 1–15.
- Fielding, A., 2007. Cluster and Classification Techniques for the Biosciences. Cambridge: Cambridge University Press, pp. 179–199.
- Furey, T.S., Cristianini, N., Duffy, N., *et al.*, 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16 (10), 906–914.
- Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13 (2), 415–425.
- Kapetanovic, I.M., Rosenfeld, S., Izmirlian, G., 2004. Overview of commonly used bioinformatics methods and their applications. *Annals of the New York Academy of Sciences* 1020 (1), 10–21.
- Kotsiantis, S.B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques.
- Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D., Levy, S., 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* 21 (5), 631–643.
- Yang, P., Hwa Yang, Y., B Zhou, B., Y Zomaya, A., 2010. A review of ensemble methods in bioinformatics. *Current Bioinformatics* 5 (4), 296–308.

Unsupervised Learning: Clustering

Angela Serra and Roberto Tagliaferri, University of Salerno, Salerno, Italy

© 2019 Elsevier Inc. All rights reserved.

Clustering

Cluster analysis is an exploratory technique whose aim is to identify groups, or clusters, of high density in which observations are more similar to each other than observations assigned to different clusters.

This process requires to quantify the degree of similarity, or dissimilarity, between observations. The results of the analysis is strongly dependent on the kind of the used similarity metric.

A large class of dissimilarities coincides with the class of distance functions. The most used distance function is the Euclidean distance, defined as the sum of the squared differences of the features between two patterns x_i and x_j :

$$d(x_i, x_j) = \sum_{h=1}^p (x_{ih} - x_{jh})^2$$

where x_{ih} is the h -th feature of x_i .

Similarly, clustering algorithms can use also similarity measures between observations. The correlation coefficient is widely used as a similarity measure

$$\rho(x_i, x_j) = \frac{\sum_h (x_{ih} - \bar{x}_i)(x_{jh} - \bar{x}_j)}{\sqrt{\sum_h (x_{ih} - \bar{x}_i)^2 \sum_h (x_{jh} - \bar{x}_j)^2}}$$

where $\bar{x}_i = \frac{1}{p} \sum_{h=1}^p x_{ih}$ is the average value of the features of a single observation.

Clustering has been widely applied in bioinformatics to solve a wide range of problems. Two of the main problems addressed by clustering are: (1) identify groups of genes that share the same pattern across different samples (Jiang *et al.*, 2004); (2) identify groups of samples with similar expression profiles (Sotiriou and Piccart, 2007).

Many are the different clustering approaches proposed in literature. These methodologies can be divided into two categories represented by the hierarchical and partitive clustering as described in Theodoridis and Koutroumbas (2008).

Hierarchical Clustering

Hierarchical clustering seeks to build a hierarchy of clusters based on a proximity measure.

Hierarchical clustering methods need the definition of a pairwise dissimilarity function and a set of observations as inputs. Hierarchical clustering methods do not define just a partition of the observations, but they produce a hierarchical representation, in which the clustering at a determined level is defined in terms of the clusters of the level below. The lowest level of the hierarchy is made of n clusters, each containing a single observation, whereas at the top level, a single cluster contains all the observations. The absence of a single partition allows the investigator to find the most natural clustering, in the sense of comparing the within cluster similarity to the between cluster similarity.

Strategies for accomplishing hierarchical clustering can be divisive or agglomerative. The former is a “top down” approach where all the observations are in one cluster and then are splitted recursively moving down into the hierarchy. The latter is a “bottom up” approach where, initially, each pattern is a “singleton” cluster and then pairs of clusters are merged moving up into the hierarchy.

Agglomerative clustering is the most used approach to construct clustering hierarchies. It starts with n clusters, each one containing only one observation. At each step the two most similar clusters are merged and a new level of the hierarchy is added. The procedure ends in $n - 1$ steps, when, at the top of the hierarchy, the last two clusters are merged.

The most used methods to perform hierarchical cluster analysis are the single linkage, the complete linkage and the average linkage.

Single Linkage

In this method, the distance between two clusters is represented by the distance of the pair of the closest data patterns belonging to different clusters. More formally, given two clusters G and H and a distance function d , the single linkage dissimilarity is defined as:

$$d_S = \min_{x_i \in G, x_j \in H} d(x_i, x_j)$$

Single linkage method is able to identify clusters of different shapes and to highlight anomalies in the data distribution better than the other hierarchical techniques. On the other side, it is sensitive to noise and outliers. Moreover it can results in clusters that represents a chain between the patterns. This can happens when clusters are well-outlined but not well separated. An illustration of how the algorithm works is in the top-left part of Fig. 1.

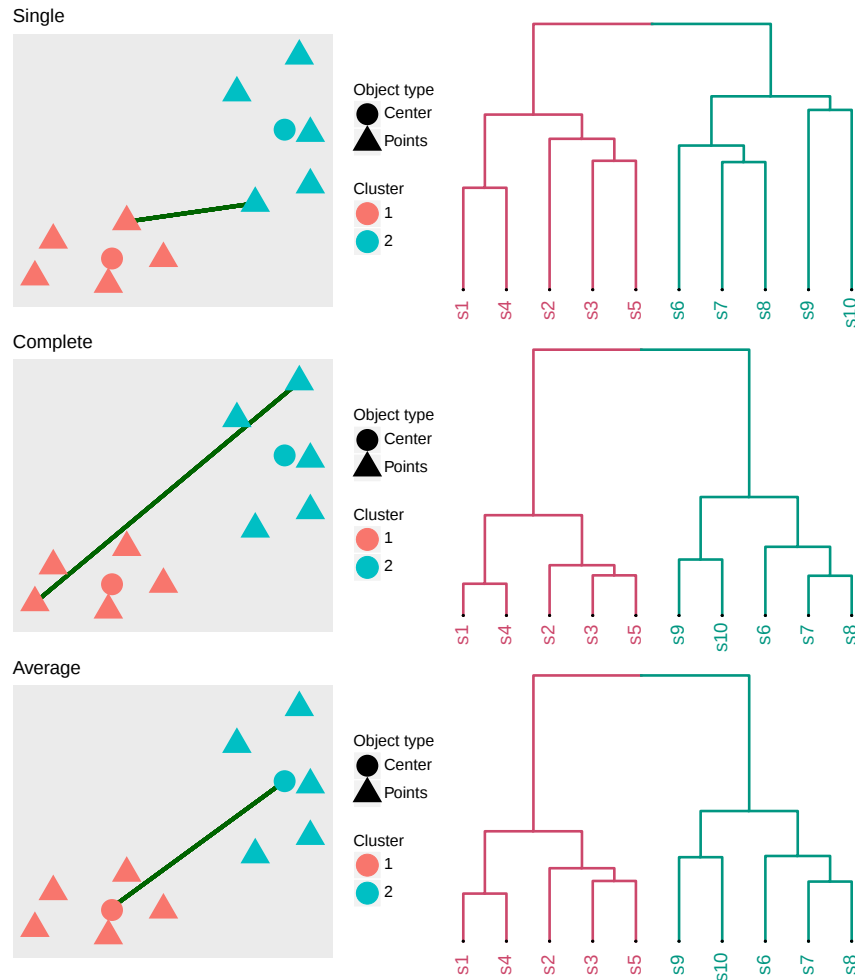


Fig. 1 The distance between any couple of clusters can be defined in different ways, resulting in different hierarchies. The left side of the figure shows two different clusters in two different colours, with the centroids of the clusters represented by circles. The green lines represent the different alternative distances between clusters. The corresponding dendrograms are shown next to each different distance definition.

Complete Linkage

In this method the distance between two clusters is represented by the distance of the pair of the farthest data points belonging to different clusters. More formally, given two clusters G and H and a distance function d , the complete linkage dissimilarity is defined as:

$$d_C = \max_{x_i \in G, x_j \in H} d(x_i, x_j)$$

This method identifies, above all, clusters of elliptical shape. This algorithm prefers the homogeneity between the elements of the group at the expense of differentiation between the groups. An illustration of how the algorithm works is in the center-left part of Fig. 1.

Average Linkage

The distance between two clusters is represented by the average distance of all pairs of data points belonging to different clusters. More formally given two clusters G and H and a distance function d , the average linkage dissimilarity is defined as:

$$d_A = \frac{1}{|G||H|} \sum_{i \in G} \sum_{j \in H} d(x_i, x_j)$$

where $|G|$ and $|H|$ are respectively the number of observations in clusters G and H . The average linkage produces clusters relatively compact and relatively separated. As a drawback, if the measurements of the two clusters to join are very different the distance will be very close to that of the largest cluster. Moreover, the average linkage is susceptible to strictly increasing transformations of the

dissimilarity, changing the results. On the contrary, single and complete linkages are invariant to such transformations as described in [Friedman et al. \(2001\)](#). An illustration of how the algorithm works is in the bottom-left part of [Fig. 1](#).

Dendrogram Representation

Regardless of the aggregation criterion, the produced hierarchy of clusters can be graphically represented by a binary tree called dendrogram. Each node on the dendrogram is a cluster while each leaf node is a singleton cluster (i.e., an input pattern or observation). The height of the tree measures the distance between the data points. An example is reported in [Fig. 1](#), where each tree corresponds to one of the aggregation criteria above defined. The clustering algorithms are performed on a toy dataset of ten data points. The height of each node is proportional to the dissimilarity between the corresponding merged clusters. Figures were accomplished by using the *ggplot2* package included in R. An in-depth description of the library can be found in [Wickham \(2009\)](#).

The dendrogram can be a useful tool to explore the clustering structure of the data at hand. Data clustering can be obtained by cutting the dendrogram at the desired height, then each connected component forms a cluster. Inspection of the dendrogram can help to suggest an adequate number of clusters presents in the data. This number and the corresponding data clustering can be used as the initial solution of more elaborate algorithms.

k-Means

The *k*-means, proposed by [MacQueen et al. \(1967\)](#), is one of the most used and versatile clustering algorithm. The goal of *k*-means is to partition the observations into a number *k* of clusters. Each cluster contains the patterns that are more similar to each other while those dissimilar to the observations are put into other clusters. A pattern, called centroid, that is computed as the centre of mass of all the observations belonging to the cluster, is the representative prototype of all these points.

Formally, let $X = \{x_1, \dots, x_n\}$ be a set of *N* points in a and let *K* be an integer value, the *k*-means algorithm seeks to find a set of *k* vectors μ_k that minimise the Within Cluster Sum of Squares (WCSS)

$$WCSS = \sum_{h=1}^k \sum_{x_i \in C_h} d(x_i, \mu_h)$$

where C_h is the *h*-th cluster and μ_h is the corresponding centroid.

The *k*-means algorithm is a two-step iterative procedure that starts by defining *k* random centroids in the feature space, each corresponding to a different cluster. Then, the patterns are assigned to the nearest centroid by using the Euclidean distance. Afterwards, each centroid is recomputed as the mean (center of mass) of the observations belonging to the same cluster. Each step is proven to minimise the WCSS until convergence to a local minimum. The assignment and recomputing steps alternate until no observation is reassigned.

The *k*-means algorithm, as described in [Jung et al. \(2014\)](#), is the following:

Algorithm 1: *k*-Means Clustering Algorithm

1. ***k*-means** (*k*, *X*);
 Input: The number *k* and a database *X* containing *n* patterns
 Output: A set of *k*-clusters that minimizes the squared-error criterion
 2. Randomly choose *k* patterns as the initial cluster centroids;
 3. repeat;
 4. (re)assign each pattern to the cluster corresponding to its nearest centroid;
 5. update the cluster centroid, i.e., for each cluster, calculate the mean value of its patterns;
 6. until no change.
-

It must be noted that this algorithm strongly depends on the initial assignment of the centroids as shown in [Fig. 2](#). Usually, if a prior knowledge about the location of the centroids does not exist, this algorithm can be run many times with different initial solutions, randomly sampled from the data observations and keeping the best solution. The number *k* of clusters is a hyper-parameter to be estimated, and different choices of it are usually compared. Note that the WCSS is a decreasing function with respect to *k*, so that care must be taken in performance comparisons with increasing values of *k*. Different heuristics to find an appropriate *k* have been proposed, ranging from graphical models to evaluate how well a point fits within its cluster, like the silhouette, described by [Rousseeuw \(1987\)](#), to information theoretic criteria that penalise the WCSS to find the right trade-off between maximum information compression (one cluster) and minimum loss of information (*n* clusters, one for each observation).

Expectation Maximisation (EM)

Another important category of clustering algorithms is the one that includes model based approaches. Here the main idea is that each cluster can be represented by a parametric distribution, such as a Gaussian or a Poisson for continuous or discrete data, respectively.

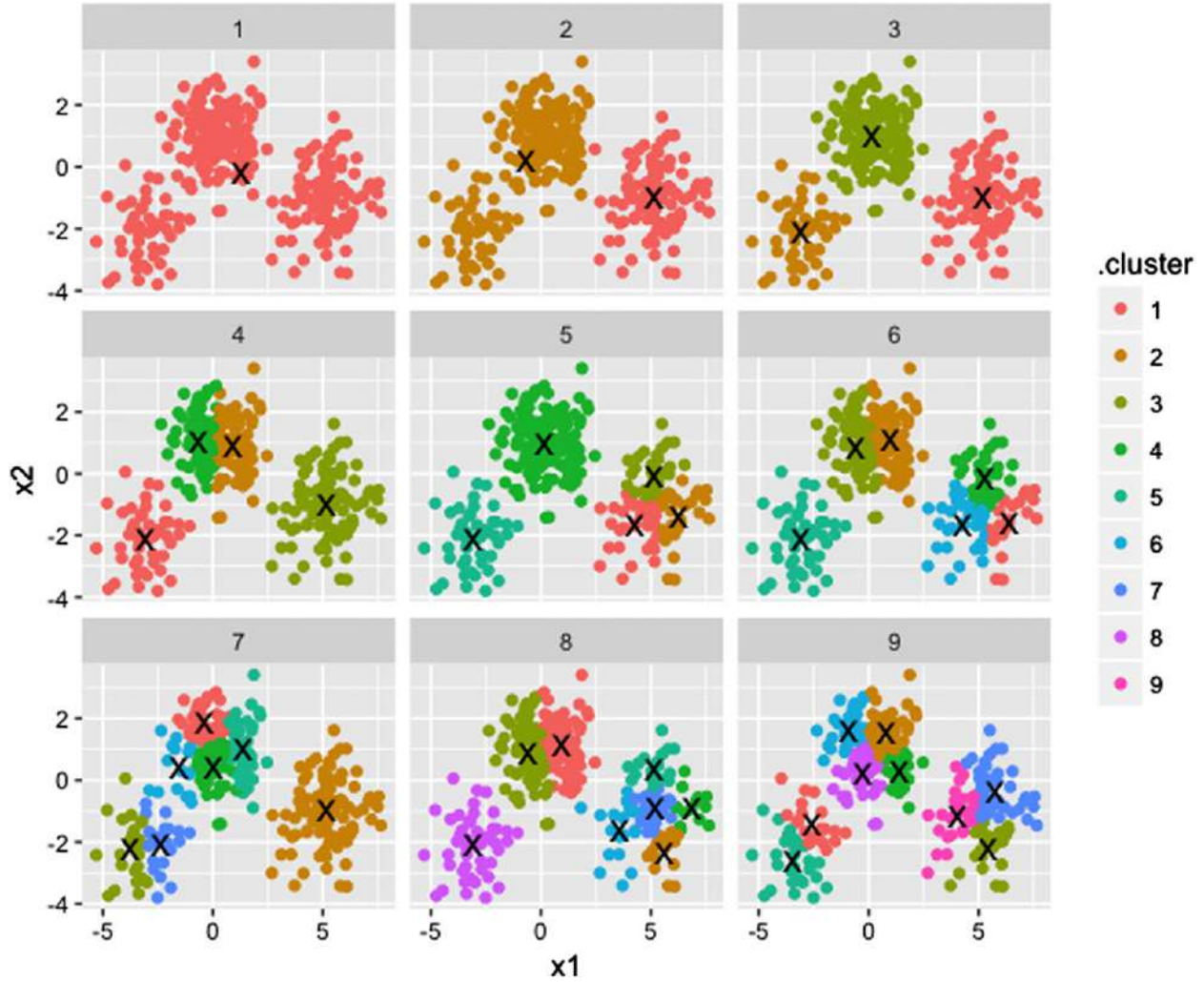


Fig. 2 *k*-Means results with different *k* values in input. A black X represents the cluster centroid.

More formally, let $X = \{x_1, \dots, x_n\}$ be n observed data vectors and let $Z = \{z_1, \dots, z_n\}$ be the n values taken by the hidden variables (i.e., the cluster labels). The expectation maximisation algorithm attempts to find the parameters θ that maximise the log probability $L(\theta) = \log P(X|\theta) = \log \sum_Z p(X, Z|\theta)$ of the observed data, where both θ and Z are unknown.

To deal with this problem, the task of optimising $\log p(X|\theta)$ is divided into a sequence of simpler sub-problems, that guarantees that their corresponding solutions $\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \dots$ converge to a local optimum of $\log P(X|\theta)$.

More specifically, the EM algorithm alternates between two phases. In the E-step, it chooses a function f_t that lower bounds $\log P(X|\theta)$ and for which $f_t(\hat{\theta}^{(t)}) = \log P(X|\hat{\theta}^{(t)})$. In the M-step, the EM algorithm find a new parameter set $\hat{\theta}^{(t+1)}$ that maximizes f_t . Since the value of f_t matches the objective function at $\hat{\theta}^{(t)}$, it follows that $\log P(X|\hat{\theta}^{(t)}) = f_t(\hat{\theta}^{(t)}) \leq f_t(\hat{\theta}^{(t+1)}) = \log P(X|\hat{\theta}^{(t+1)})$. This proves that the objective function monotonically increases at each iteration of EM leading to the algorithm convergence.

Similar to the *k*-means algorithm, EM is an iterative procedure: the E-step and M-step are repeated until the estimated parameters (means and covariances of the distributions) or the log-likelihood do not change anymore.

Mainly, we can summarize the EM clustering algorithm as described in [Jung et al. \(2014\)](#) as follows:

The EM technique is quite similar to the *k*-means. The EM algorithm extends this basic approach to clustering into two important ways:

- The EM clustering algorithm computes probabilities of cluster memberships based on one or more probability distributions. Therefore, the goal of the algorithm is to maximise the overall probability or likelihood of data, given the (final) clusters. See [Fig. 3](#).
- Unlike the classical implementation of *k*-means, the general EM algorithm can be applied to both continuous and categorical variables.

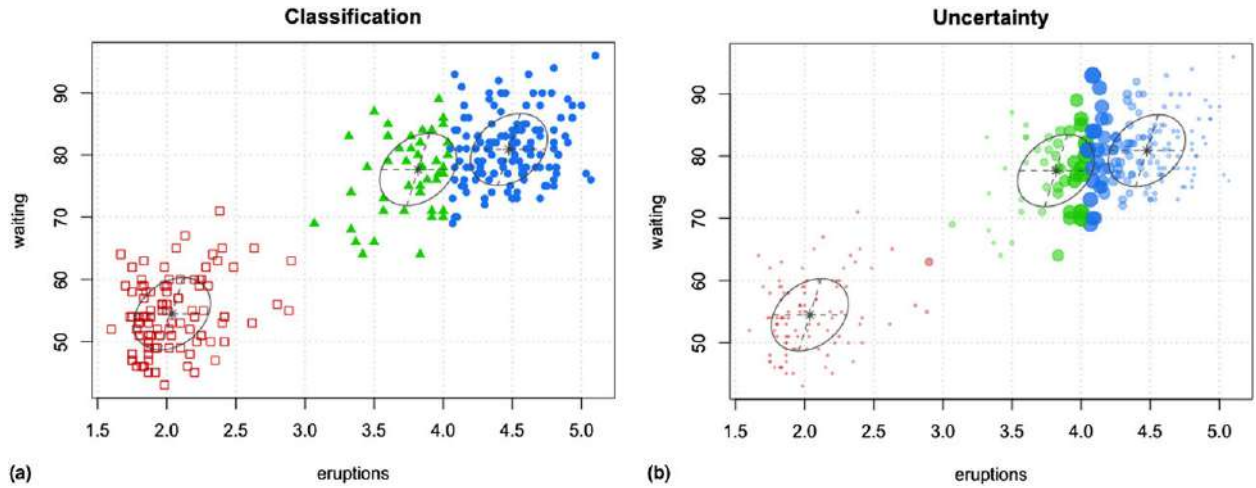


Fig. 3 EM clustering results. The (a) panel shows how the points are assigned to each cluster. The (b) panel shows the uncertainty in the assignment: as we can see, there is a higher confidence in the assignment of the points to the red cluster, while there is some uncertainty between the green and blue clusters. Smaller points have less uncertainty in the assignment.

Algorithm 2: EM clustering algorithm

1. $\underline{\text{EM}}(k, X, \text{eps})$;
Input: Cluster number k , a database X , stopping tolerance eps
Output: A set of k -clusters with weights that maximise the log-likelihood function
 2. Expectation step: For each database record x , compute the membership probability of x in each cluster $h = 1, \dots, k$.
 3. Maximization step: Update mixture model parameters (probability weights).
 4. Stopping criteria: If stopping criteria are satisfied (convergence of parameters and log-likelihood) then stop, else set $j = j + 1$ and go to (2).
-

Different from k -means, the EM algorithm does not compute actual assignments of observations to clusters, but classification probabilities. In other words, each observation belongs to each cluster with a certain probability. Of course, as a final result an actual assignment of observations to clusters can be computed based on the (highest) classification probability.

External Clustering Quality Measures

In order to compare the clustering results with prior knowledge related to real class membership values, several indexes can be used. Some examples are the Rand Index, the Purity Index and the Normalised Mutual Information.

Rand Index

Let $S = o_1, \dots, o_n$ be a set of n elements and X and Y be two of its partitions to compare, where $X = X_1, \dots, X_r$ is a partition of S into r subsets and $Y = Y_1, \dots, Y_s$ is a partition of S into s subsets, we can define the following measures: (i) a , is the number of element pairs in S that are in the same subset both in X and in Y ; (ii) b is the number of element pairs in S that are in different subsets both in X and in Y ; (iii) c is the number of element pairs in S that are in the same subset in X and in different subsets in Y ; (iv) d is the number of element pairs in S that are in different subsets in X and in the same subset in Y . The Rand Index R is defined as:

$$R = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}}$$

Intuitively, $a + b$ can be considered as the number of agreements between X and Y and $c + d$ as the number of disagreements between X and Y . Since the denominator is the total number of pairs, the Rand Index represents the occurrence frequency of agreements over the total pairs.

Purity

Purity is one of the most useful external criteria for clustering quality evaluation, when prior knowledge on real class memberships is available. To compute purity, each cluster is labelled with the class which represents the majority of the points it contains.

Then the accuracy of this assignment is measured by counting the number of correctly assignments divided by the number of points.

Given a set of N elements, their clustering partition $\omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ and their real set of classes $C = \{C_1, C_2, \dots, C_J\}$ the purity measure is defined as following:

$$\text{purity}(\omega, C) = \frac{1}{N} \sum_k \max_j |w_k \cap c_j|$$

where w_k is the set of elements in the cluster ω_k and c_j is the set of elements of class C_j .

Normalised Mutual Information

The Mutual Information (MI) of two random variables is a measure of their mutual dependence. It quantifies the amount of information that a variable can give about the other. Formally, the MI between two discrete random variables X and Y is defined as:

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right)$$

where $p(x, y)$ is the joint probability distribution function of X and Y and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y , respectively.

The concept of MI is intricately linked to that of Entropy of a random variable, a fundamental notion in Information Theory, that defines the “amount of information” held in a random variable.

The MI measure is not upper limited, then the Normalised Mutual Information (NMI) measure has been defined as:

$$R = \frac{I(X, Y)}{H(X) + H(Y)}$$

where $H(X)$ and $H(Y)$ are the entropies of X and Y , respectively. NMI ranges between 0 and 1, with 0 meaning no agreement between the two variables and 1 meaning complete agreement.

An Example of Cluster Analysis Application to Patient Sub-Typing

As noticed by [Saria and Goldenberg \(2015\)](#), many diseases, such as cancer or neurodegenerative disorders, are difficult to diagnose and to treat because of the high variability among affected individuals.

Precision medicine tries to solve this problem by identifying individual variability in gene expression, lifestyle and environmental factors ([Hood and Friend, 2011](#)). The final aim is to better predict disease progression and transitions between the disease stages and targeting the most appropriate medical treatments ([Mirmezami et al., 2012](#)).

A central role in precision medicine is played by patient sub-typing, whose aim is to identify sub-populations of patients that share similar behavioural patterns. This can lead to more accurate diagnostic and treatment strategies. From a clinical point of view, refining the prognosis for similar individuals can reduce the uncertainty in the expected outcome of a treatment on individuals.

In the last decade, the advent of high-throughput technologies has provided the means for measuring differences between individuals at the cellular and molecular levels. One of the main goals driving the analyses of high-throughput molecular data is

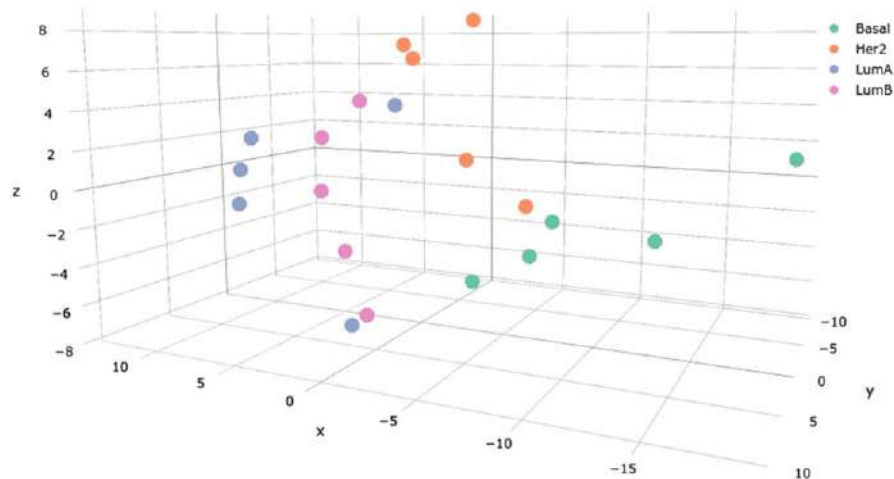


Fig. 4 3D-MDS dataset representation of the example described in the text.

the unbiased biomedical discovery of disease sub-types via unsupervised techniques, such as cluster analysis (Sorlie *et al.*, 2001; Serra *et al.*, 2015; Wang *et al.*, 2014).

Here we report an example of application of the hierarchical, *k*-means and EM algorithms to the clustering of patients affected by breast cancer.

The dataset is related to 20 breast cancer patients randomly selected from the TCGA repository (Breast invasive carcinoma (BRCA) – see Relevant Website section). The patients are divided into four classes (Her2, Basal, LumA, LumB), using the PAM50 classifier described in Tibshirani *et al.* (2002). Data are pre-processed as in Serra *et al.* (2015). Since gene expression data are high dimensional (more than 20 k genes), genes were grouped into 100 clusters and only the cluster prototypes were used as features for the patient clustering task. Moreover, for clarity purpose, only 5 random patients for each class are used in this example. In Fig. 4 the 3D multidimensional scaling projection of the dataset is reported.

All the experiments were performed by using the Euclidean distance and by setting the number of clusters to the same number of classes ($k=4$). Fig. 5 shows the results of the applied clustering algorithms. For each method, the true patient classes are represented by different shapes and the cluster assignments by different colours. As we can see from Fig. 5 and from Table 1 the clustering assignment that best resembles the real classes is the *k*-means, being that with the highest rand index (0.87), purity (0.85) and NMI (0.72), meaning that each cluster contains a majority of patients of the same class.

Figures were accomplished by using the *ggplot2* package included in R.

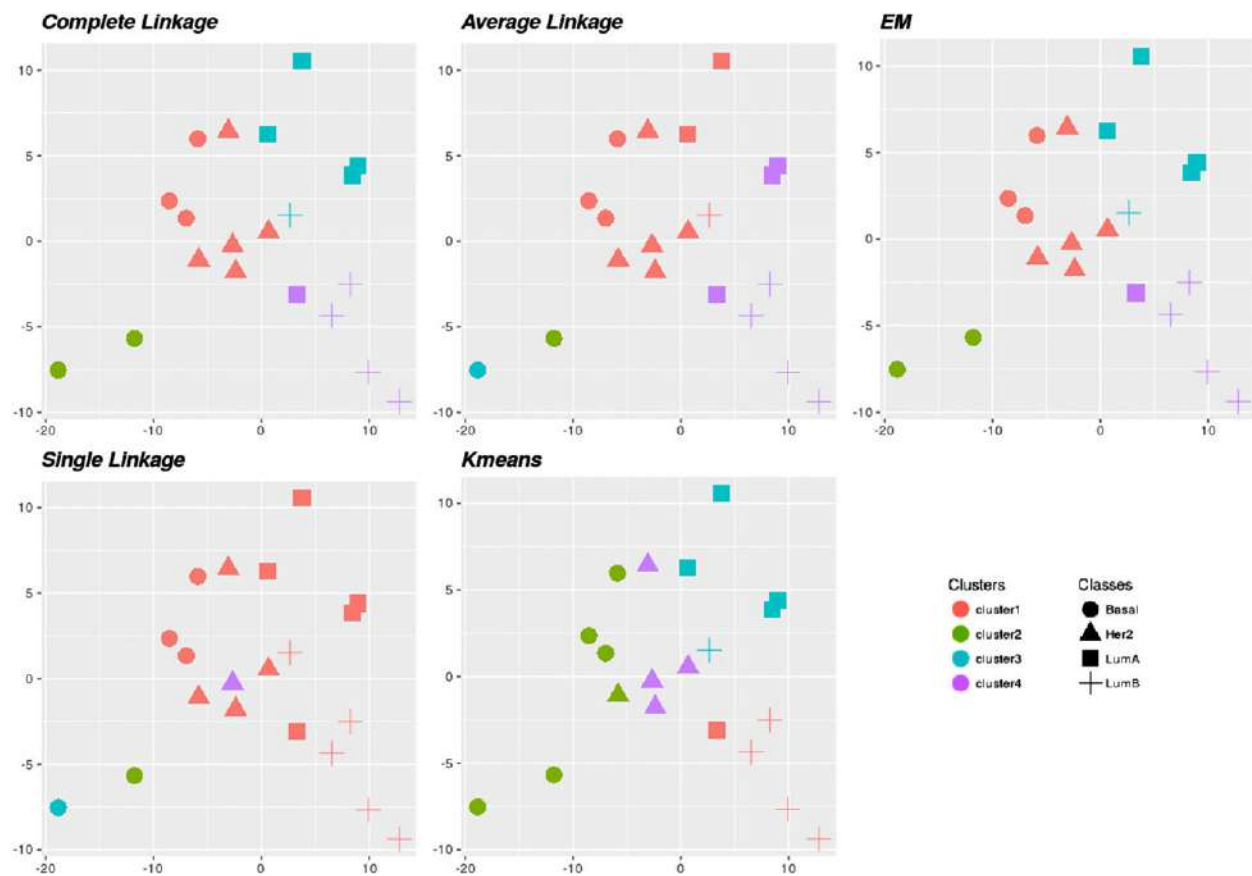


Fig. 5 *k*-Means, EM and Hierarchical clustering results with 4 clusters. In the scatters the shapes represent the true patient classes, while the colours represent the cluster assignments of each algorithm.

Table 1 External validation index for clustering evaluations

	<i>Complete</i>	<i>Single</i>	<i>Average</i>	<i>k-Means</i>	<i>EM</i>
Rand index	0.81	0.38	0.63	0.87	0.81
Purity	0.75	0.40	0.55	0.85	0.75
NMI	0.65	0.23	0.39	0.72	0.65

Note: The results of the different clustering algorithms were compared with the true patient classes.

See also: The Challenge of Privacy in the Cloud

References

- Friedman, J., Hastie, T., Tibshirani, R., 2001. The Elements of Statistical Learning. Springer Series in Statistics, vol. 1. Berlin: Springer.
- Hood, L., Friend, S.H., 2011. Predictive, personalized, preventive, participatory (p4) cancer medicine. *Nature Reviews Clinical Oncology* 8 (3), 184–187.
- Jiang, D., Tang, C., Zhang, A., 2004. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering* 16 (11), 1370–1386.
- Jung, Y.C., Kang, M.S., Heo, J., 2014. Clustering performance comparison using k-means and expectation maximization algorithms. *Biotechnology & Biotechnological Equipment* 28 (Suppl.), S44–S48.
- MacQueen, J. *et al.*, 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Oakland, CA, USA. vol. 1, pp. 281–297.
- Mirnezami, F.L., Nicholson, J., Darzi, A., 2012. Preparing for precision medicine. *New England Journal of Medicine* 366 (6), 489–491.
- Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20, 53–65.
- Saria, S., Goldenberg, A., 2015. Subtyping: What it is and its role in precision medicine. *IEEE Intelligent Systems* 30 (4), 70–75.
- Serra, A., Fratello, M., Fortino, V., *et al.*, 2015. MVDA: A multi-view genomic data integration methodology. *BMC Bioinformatics* 16 (1), 261.
- Sørbye, T., Perou, C.M., Tibshirani, R., *et al.*, 2001. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences* 98 (19), 10869–10874.
- Sotiriou, C., Piccart, M.J., 2007. Taking gene-expression profiling to the clinic: When will molecular signatures become relevant to patient care? *Nature Reviews Cancer* 7 (7), 545–553.
- Theodoridis, S., Koutroumbas, K., 2008. *Pattern Recognition*. Academic Press.
- Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America* 99 (10), 6567–6572.
- Wang, B., Mezlini, A.M., Demir, F., *et al.*, 2014. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods* 11 (3), 333–337.
- Wickham, H., 2009. *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer-Verlag.

Further Reading

- Handl, J., Knowles, J., Kell, D.B., 2005. Computational cluster validation in post-genomic data analysis. *Bioinformatics* 21 (15), 3201–3212.
- Hartigan, J.A., Hartigan, J., 1975. *Clustering Algorithms*, vol. 209. New York: Wiley.
- Zvelebil, M., Baum, J., 2007. Clustering methods and statistic. In: *Understanding Bioinformatics*. Oxford: Garland Science, (Chapter 16).

Relevant Website

<https://tcga-data.nci.nih.gov/tcga/>
Breast invasive carcinoma (BRCA).

Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations

Massimo Cafaro, Italo Epicoco, and Marco Pulimeno, University of Salento, Lecce, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Modern technologies allow collecting enormous amount of data in every field of human activity and regarding every aspect of human life. Industry, science, medicine, individuals in their day-to-day life produce data that can and must be analyzed because it often hides valuable knowledge, but this analysis can not be done by humans alone, even tough domain experts, since data generation happens at rates and in such volumes that traditional techniques are useless.

Data Mining is the discipline which deals with the process of extracting useful knowledge from large volumes of data (Han *et al.*, 2011). Its goal is to find knowledge that can ease comprehension and insight and can support decision-making with regard to the phenomena that have generated the data. In the context of data mining, data is the raw material which has to be excavated for knowledge, hence, differently from the meaning of similar expressions, for example, *gold mining*, data mining stands for mining *into* data and not *in search of* data.

Data Mining is an interesting and very active area of research, which has been received even more attention in the last decades. Although it is often confused with the general process of knowledge discovery from data, the term data mining refers to the core step of a wider pipeline of activities concerning the treatment of data for knowledge discovery. This pipeline includes the creation of the data set to analyze, by selecting and integrating data from possibly multiple sources; the cleaning and preprocessing of the data set, by deleting noisy and inconsistent data; the data transformation, which consists of reducing and transforming the data set to shape it in a form suitable for the analysis; the data mining step, which includes the choice of the mining task to accomplish (summarization, classification, clustering, pattern mining, etc.) and the selection and execution of the mining algorithm; at last the interpretation, evaluation and presentation of the results obtained (Aggarwal, 2015; Rajaraman *et al.*, 2012; Zaki and Wagner Meira, 2014).

The mining of frequent patterns, association rules, and correlations is one of the possible data mining tasks that one would want to accomplish, beyond being one of the most studied (Aggarwal and Han, 2014). In general, a pattern is some kind of scheme or configuration which recurs in the data. Based on the type of data analyzed, a pattern can consist of a set of items, or a subsequence, or a substructure, such as a subgraph or subtree. Frequent patterns are patterns that occur frequently in a data set, for example, set of items that appear together in more than a given fraction of the whole data set. Frequent patterns are important in order to determine associations and correlations among data and also as a possible useful component of other mining tasks, such as classification and clustering.

Association rules mining has been introduced in the context of market-basket analysis (Agrawal *et al.*, 1993), i.e., the analysis of products that customers purchase in a single visit to the store in search of sets of products frequently bought together. But since the first formulation, frequent pattern mining has become a data mining tool extensively used in a wide variety of domains and also employed as an intermediate step for other data mining tasks, such as classification and clustering. It has been applied to document sets analysis (Beil *et al.*, 2002; Boley *et al.*, 1999; Fung *et al.*, 2003), web log mining, for example, searching for the set of pages more likely to be accessed together during the same session by a user (Ivancsy and Vajk, 2006; Mobasher *et al.*, 1996), recommender systems (Lin *et al.*, 2002; Mobasher *et al.*, 2001), and anomaly detection (Lee *et al.*, 1998; Leung and Leckie, 2005).

Furthermore, recently, frequent pattern mining has been successfully adopted in life sciences data analysis. It has been applied to medical and biological data (Atluri *et al.*, 2009; Naulaerts *et al.*, 2015), for better disease diagnosis, or to improve the knowledge about what causes particular diseases, and for a wide range of specific bioinformatics problems, including annotation mining, structural motif discovery and biclustering of expression profiles.

In this article, we introduce in Section Basic Concepts the basic notions of frequent itemsets, associations and correlations in the context of transaction records. In the same Section, we also discuss some measures of interestingness and usefulness of associations among itemsets and in Section Frequent Itemset Mining algorithms we describe some algorithms that allow efficiently mining frequent itemsets.

Basic Concepts

In this section, we introduce and formalize the basic concepts which underlie the mining of frequent itemsets in transactional data sets and the derivation of interesting associations and correlations among the analyzed data. As a concrete example, we shall refer to the market-basket analysis, which consists in analyzing the business transaction records of a supermarket in order to find interesting associations among the items (products) that appear together in the shopping carts of customers.

A way to express co-occurrence relationships among items that appear in transactional data sets is by means of association rules. An association rule is an implication of the form $A \Rightarrow B$, where both A and B are set of items and $A \cap B = \emptyset$. It suggests that it is likely for a transaction which contains the itemset A , called the *antecedent* of the rule, to also contain the itemset B , called the *consequent* of the rule. In the context of market-basket analysis, for example, $\{bread\} \Rightarrow \{butter\}$ indicates that it is likely for a

customer who buys bread to also buy butter in the same trip to the supermarket. Finding association rules that describe the customer behaviours can greatly help in delineating successful marketing strategies, for example, a retailer can organize her shelves so that butter is positioned close to bread in order to ease the purchase of both the products. Otherwise, a completely different strategy could consist in placing the butter far from the bread so that a customer who buys bread needs to pass other shelves and is tempted by different products in its way toward the butter. The kind of knowledge brought to the retailer by association rules can also be helpful in planning a good price policy, for example, a discount on the bread can be paired with an increase in the price of butter: it is unlikely that a customer attracted by the discount on bread chooses to give up on butter owing to the higher price.

Frequent Itemsets and Association Rules

Let $\mathcal{U}$ be the universe of items, i.e., the set of all of the possible items that can appear in a data set. We also refer to a generic set of items drawn by $\mathcal{U}$ as an *itemset* and to a set of precisely k items as a *k-itemset*. A transaction is a set of items drawn by $\mathcal{U}$ and a transaction database consists of a set of transactions. Let $\mathcal{D} = \{T_1, T_2, \dots, T_n\}$ be a database of n transactions to be mined. Each transaction T_i in $\mathcal{D}$ is represented by a unique identifier TID_i , $i = 1, 2, \dots, n$, called *transaction ID*.

Definition 1: (the support of an itemset). Given an itemset I , the support of I with reference to a transaction database $\mathcal{D}$ is the fraction of transactions in $\mathcal{D}$ that contain I as a subset. The support of the itemset I , denoted by $sup(I)$, is defined as $sup(I) = \frac{|\{T_i \in \mathcal{D} \mid I \subseteq T_i\}|}{|\mathcal{D}|}$, where $|\mathcal{D}|$ is the total number of transactions in $\mathcal{D}$.

Alternatively, we can consider only the number of transactions that contains a given itemset. We refer to this number as the *support count* of the itemset, $sup_count(I) = |\{T_i \in \mathcal{D} \mid I \subseteq T_i\}|$.

The support of an itemset A represents an estimate of the probability $P(A)$ of finding that itemset in the data set. An itemset is *frequent* if it appears in a fraction of transactions which exceeds a minimum support threshold fixed by the data miner and specific to the mining task.

Definition 2: (frequent itemsets). Given a transaction database $\mathcal{D}$, an itemset I with support $sup(I)$ in $\mathcal{D}$, and a predefined minimum support threshold, $minsup$, the itemset I is said to be frequent if $sup(I) \geq minsup$.

A transaction data set can be easily represented as a table of transactions with each row consisting of a transaction ID and the set of items in the transaction. A transaction itemset, in turn, can be represented by a set of items or by a binary vector of length $|\mathcal{U}|$ where each bit represents an item in $\mathcal{U}$ and is set or unset on the basis of the presence of that item in the transaction. This representation of a data set is referred to as *horizontal*. Table 1 shows an example of database in horizontal form, borrowed as usual from the market-basket analysis domain. The table refers to the universe set of items $\mathcal{U} = \{\text{bread, butter, cheese, milk, water}\}$.

Alternatively, a database of transactions can be represented in *vertical* form, as a table that reports for each item the set of transactions (transaction IDs) that contain that item (this set is also referred to as *tidset*). We refer to the tidset of an itemset X as $T(X)$. Also for the vertical representation, the tidset can be written as a binary vector, where each transaction is mapped to a bit whose value is one if the item belongs to the transaction and zero otherwise. Table 2 shows the same database of Table 1 but in vertical form.

In the data set showed in Tables 1 and 2, the itemset $I = \{\text{bread, butter}\}$ has support count $sup_count(I) = 3$ and its support expressed as a percentage is $sup(I) = 50\%$, therefore the itemset occurs in 50% of all of the transactions of the database, i.e., three

Table 1 Example of a horizontal database

Transaction ID	Itemset	Binary representation
1	{milk, bread}	10010
2	{bread, butter}	11000
3	{bread, cheese}	10100
4	{milk, bread, butter}	11010
5	{bread, butter}	11000
6	{milk, water}	00011

Table 2 Example of a vertical database

Item	Tidset	Binary representation
Milk	{1, 4, 6}	100101
Bread	{1, 2, 3, 4, 5}	111110
Butter	{2, 4, 5}	010110
Cheese	{3}	001000
Water	{6}	000001

out of six transactions. If we set a minimum support threshold equal to 30%, then the itemsets {bread, butter} and {bread, milk} are frequent 2-itemsets, whilst the itemset {bread} is a frequent 1-itemset or frequent item.

Finding frequent itemsets is the first step toward the discovery of association rules. It is the most computationally intensive and, therefore, it has been the main focus of data mining researchers. Once the frequent itemsets of a data set are computed, the second step simply consists in considering each frequent itemset I and enumerating the possible association rules of the form $X \Rightarrow I \setminus X$, with $X \subset I$, discarding all of the rules judged as not useful or not interesting.

Two measures of interestingness usually adopted in the generation of association rules are *support* and *confidence*. Rules with support and confidence above specified thresholds are said to be strong and selected for output. Let us formally define these two metrics.

Definition 3: (support of an association rule). *Given an association rule $A \Rightarrow B$, the support of the rule is defined as*

$$\text{sup}(A \Rightarrow B) = \frac{\text{sup}(A \cup B)}{|D|}.$$

Definition 4: (confidence of an association rule) *Given an association rule $A \Rightarrow B$, the confidence of the rule is defined as*

$$\text{conf}(A \Rightarrow B) = \frac{\text{sup}(A \cup B)}{\text{sup}(A)}.$$

The support of an association rule coincides with the support of the itemset from which the rule is generated. It is an indication of how often the rule can be applied to a given input dataset. A rule whose support is very low may occur in the input dataset only by chance. Moreover, low support rules are likely to be, in general, not interesting (from a business oriented perspective, it may not be profitable promoting items rarely bought together by customers).

On the other hand, the confidence metrics measures how reliable the inference made by a rule is. For a given association rule $A \Rightarrow B$, a higher confidence value means that it is more likely for B to be present in transactions that contain A . By its definition, the confidence of a rule can also be interpreted as an estimate of the conditional probability of B given A , $P(B|A)$.

Definition 5: (strong association rules) *Let A and B be two sets of items and let minsup and minconf be respectively a predefined minimum support threshold and minimum confidence threshold. Then, the association rule $A \Rightarrow B$ is said to be strong if it satisfies both the following criteria:*

1. *The support of the itemset $A \cup B$ is at least minsup (itemset $A \cup B$ is frequent);*
2. *The confidence of the rule $A \Rightarrow B$ is at least minconf .*

The minimum support threshold and minimum confidence threshold are usually defined by the data miner on the basis of the specific domain and application. The former is used in the determination of the frequent itemsets, in the first step. The latter is employed in the second step, when association rules are generated from the frequent itemsets, in order to select only strong rules.

Closed and Maximal Frequent Itemsets

We note that if an itemset I occurs in a transaction, then all of its subsets will also be contained in the same transaction. This simple observation leads to the following property, referred to as the *support monotonicity property*.

Property 1: (support monotonicity). *The support of any non-empty subset J of an itemset I is always at least equal to the support of I :*

$$\text{sup}(J) \geq \text{sup}(I) \quad \forall J \subseteq I.$$

An immediate consequence of monotonicity of support is that every subset of a frequent itemset is also frequent. This property is referred to as *downward closure*.

Property 2: (downward closure). *Every non-empty subset of a frequent itemset is also frequent.*

The downward closure allows efficiently pruning the search space. In fact, if an itemset I is not frequent, then any superset $K \supseteq I$ cannot be frequent and can be discarded.

Following straight from the definition of confidence and the property of support monotonicity, a monotonicity property is also valid for the confidence of association rules.

Property 3: (confidence monotonicity). *Let X_1 , X_2 and I be itemsets such that $X_1 \subset X_2 \subset I$. Then the confidence of $X_2 \Rightarrow I \setminus X_2$ is at least that of $X_1 \Rightarrow I \setminus X_1$:*

$$\text{conf}(X_2 \Rightarrow I \setminus X_2) \geq \text{conf}(X_1 \Rightarrow I \setminus X_1).$$

Considering the downward closures, it is easy to see that the number of frequent itemsets of a data set can quickly grow and become unmanageable, especially if long frequent itemsets are present. In fact, if a 30-itemsets is frequent also its 2^{30} possible non-empty subsets are frequent. To overcome this problem, two particular classes of frequent itemsets have been introduced, the *closed frequent itemsets* and the *maximal frequent itemsets*.

Definition 6: (closed frequent itemsets). *Given a minimum support minsup , a frequent itemset I is closed if $\text{sup}(I) \geq \text{minsup}$ (i.e., I is frequent), and none of its supersets have exactly the same support $\text{sup}(I)$.*

Definition 7: (maximal frequent itemsets). *Given a minimum support minsup , a frequent itemset I is maximal if $\text{sup}(I) \geq \text{minsup}$ (i.e., I is frequent), and no superset of I is frequent as well.*

Fig. 1 shows the set relationships among frequent itemsets, closed frequent itemsets and maximal frequent itemsets.

It can be shown that from the set of closed frequent itemsets the complete set of frequent itemsets and their supports can be derived, so that closed frequent itemsets are a compressed and more manageable way of representing frequent itemsets. On the other hand, although smaller, the same can not be said for the set of maximal frequent itemsets, from which one can derive the identity of all of the frequent itemsets but not their supports.

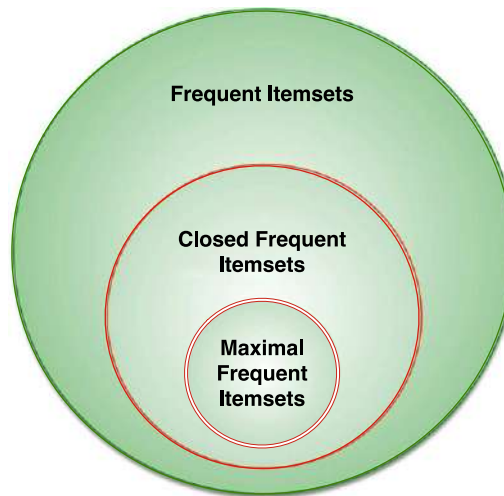


Fig. 1 Set relationships among frequent, closed and maximal itemsets.

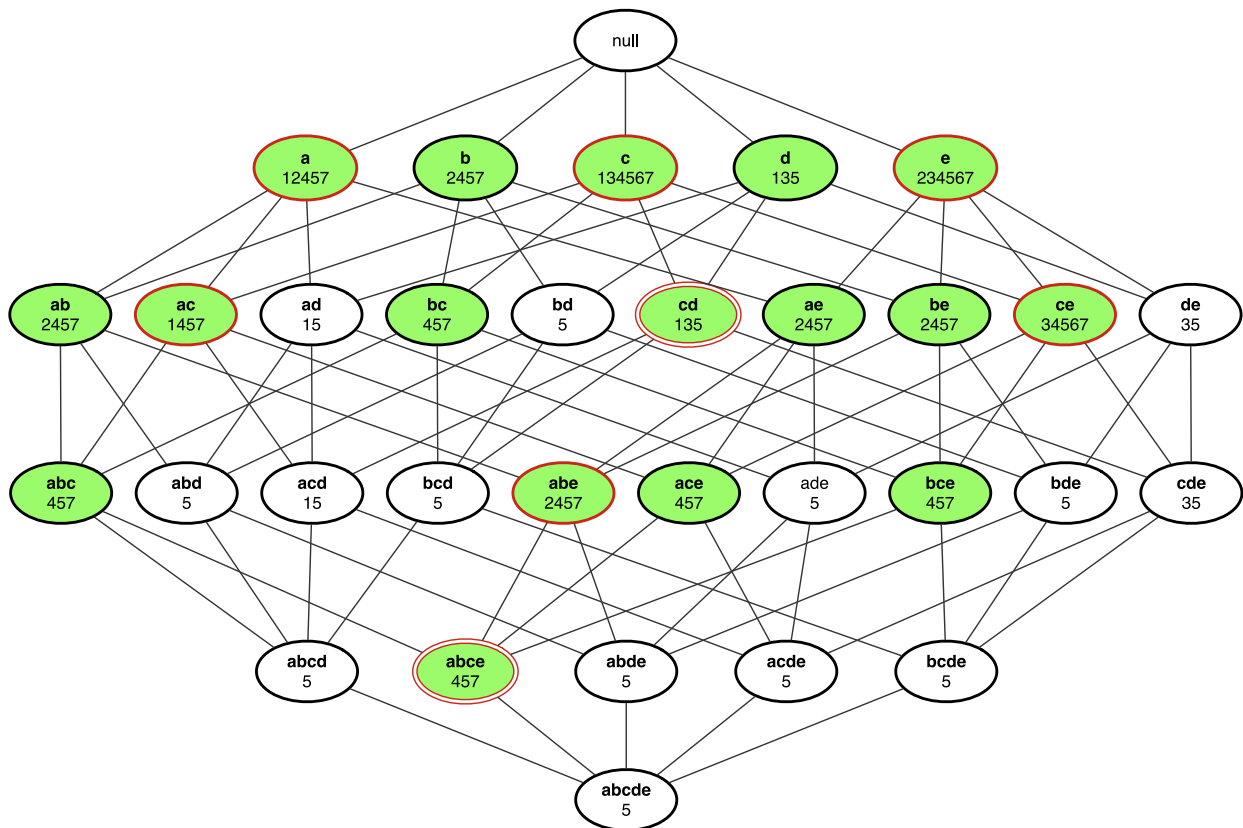


Fig. 2 Example of a lattice: green nodes are frequent itemsets, red marked green nodes are closed frequent itemsets, double red bordered green nodes are maximal frequent itemsets.

Table 3 Example of transaction database

Transaction ID	Itemset
1	{a, c, d}
2	{a, b, e}
3	{c, d, e}
4	{a, b, c, e}
5	{a, b, c, d, e}
6	{c, e}
7	{a, b, c, e}

There exists a simple graphical representation of itemsets, called a *lattice*, shown in Fig. 2. A lattice is a graph $G=(V, E)$ in which V is the set of vertices, each one representing an itemset, with $|V| = 2^{|\mathcal{U}|}$, and E is the set of edges, where an edge connects a pair of vertices *iff* (if and only if) the corresponding itemsets differ by exactly one item.

The lattice is a natural representation of the search space of frequent itemsets. Since the number of itemsets in the lattice is exponential (being the power set of the universe $\mathcal{U}$), a brute-force frequent pattern mining algorithm trying to explicitly traverse the lattice can not be efficient. Pruning the search space by implicitly traversing the lattice is therefore a common feature of all of the frequent pattern mining algorithms. Fig. 2 refers to the set of transactions in Table 3: for each itemset the IDs of the transactions containing it are indicated. Green nodes in Fig. 2 are frequent itemsets, whilst green nodes with a marked red border are closed frequent itemsets; nodes with a double red border are maximal frequent itemsets.

Correlation and Implication Rules

Although the minimum support and confidence thresholds help to discard many uninteresting association rules and these two measures are actually used by several algorithms, in many contexts they do not suffice to capture the degree of correlation and implication between antecedent and consequent of a rule. For this reason, additional correlation and implication measures can be evaluated along with support and confidence (Brin *et al.*, 1997a,b). When we augment the support-confidence framework with a correlation measure, we also refer to association rules as *correlation rules*.

We consider here a particular correlation measure called *lift* or *interest*. Lift allows quantifying the correlation of the two itemsets involved in an association rule. Let us consider the rule $A \Rightarrow B$, if the occurrence of A in a transaction is independent from the occurrence of B then $P(A \cup B) = P(A)P(B)$, otherwise the two occurrences are correlated. Lift is computed as following

$$\text{lift}(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{\text{conf}(A \Rightarrow B)}{\text{sup}(B)} \quad (1)$$

Unlike confidence, lift also takes into account $P(B)$ and allows overcoming the main weakness of the confidence metrics, which can be high and lead to valid rules also in case of uncorrelated or negatively correlated itemsets. In fact, lift reveals and measures both positive and negative correlations. If the lift value of an association rule is greater than 1, then antecedent and consequent are positively correlated, i.e., it is likely that the occurrence of one leads to the occurrence of the other. If the lift value is less than 1, then the antecedent is negatively correlated with the consequent, i.e., it is likely that the occurrence of one leads to the absence of the other. If the lift value is equal to 1, then antecedent and consequent are independent and there is no correlation between their occurrences.

Lift measurement allows improving on the support-confidence framework quantifying the degree of correlations between antecedent and consequent occurrences, but the computed correlation is not directional and does not measure the degree of implication. Another metric has been introduced for this purpose, called *conviction*, and the term *implication rule* is used in place of association rule when this measure is adopted. It is computed as:

$$\text{conv}(A \Rightarrow B) = \frac{P(A)P(\overline{B})}{P(A \cup \overline{B})} \quad (2)$$

Differently from lift, conviction is a measure of implication: it is directional and equal to one when antecedent and consequent are unrelated. A value greater than 1 indicates a relationship of implication between antecedent and consequent: the stronger the implication, the greater the value, up to infinity for implications which are always valid.

Frequent Itemset Mining Algorithms

In this section, we describe three classical algorithms for frequent itemsets and association rules mining, which represent three different fundamental approaches to this problem: the seminal algorithm by Agrawal and Srikant (1994) called Apriori; Eclat, an algorithm by Zaki (2000) exploiting the vertical database format; and FP-growth, by Han *et al.* (2000), which solves the problem of association rule mining without recurring to candidate generation. We shall focus on the first and more costly

step of association rule mining, i.e., how the three algorithms solve the problem of finding the frequent itemsets of a transaction database. The presentation is based on Han *et al.* (2011), Zaki and Wagner Meira (2014).

Apriori

The main characteristic of Apriori algorithm is its level-wise (breadth-first) exploration of the itemset lattice (see Fig. 2) in search of the frequent itemsets. Starting with the frequent 1-itemsets, for $k \geq 2$, it proceeds first by discovering all of the frequent itemsets of length k , and then computing the frequent $(k + 1)$ -itemsets. In fact, at each level, the computation leverages the knowledge acquired at the previous level.

The algorithm starts by identifying the set L_1 of the frequent 1-itemsets, i.e., the frequent items of the database to be mined. This step requires a first scan of the database. Then, at each subsequent value of $k \geq 2$, where k is the length of the itemsets computed in the current iteration, the set L_{k-1} of the frequent $(k - 1)$ -itemsets is used to generate the set of k -itemsets that are candidate to be frequent. The generated set, C_k , is a superset of the set of frequent k -itemsets, therefore, a scan of the database is required in order to compute the actual support of each candidate and filter out the candidates whose support is below the *minsup* threshold. The remaining itemsets, after the pruning, are all frequent and form the set L_k . The two phases of candidate generations and frequent itemsets determination alternates until no more frequent itemsets are found.

The candidate generation phase of each iteration is optimized by taking advantage of the downward closure property of frequent itemsets, i.e., the fact that all of the subsets of a frequent itemset must be frequent as well. The property is used to reduce the number of candidates in C_k by eliminating all of the candidates that can not be frequent since they have a $(k - 1)$ -subset which is not frequent, i.e., it is not present in L_{k-1} .

The pseudo-code for Apriori is shown in Algorithm 1. The procedure `APRIORI_GEN` deals with the generation of the candidate set C_k starting from the frequent $(k - 1)$ -itemsets already computed in the previous iteration. The candidate generation requires two steps. The first step consists in a join involving the set $\mathcal{L}_{k-1}$ with itself. Taking into account that there exists a lexicographic order among itemsets and items in itemsets are ordered, a new k -itemset is generated from two itemsets l_1 and l_2 in $\mathcal{L}_{k-1}$, where $l_1 < l_2$, and the first $(k - 2)$ items are the same in both l_1 and l_2 , by appending to l_1 the last item in l_2 . The second step checks whether all of the $(k - 1)$ -subsets of each generated k -candidate are frequent, i.e., they are in $\mathcal{L}_{k-1}$. If the candidate passes the test, it is added to C_k , otherwise it is discarded.

Eclat

The Equivalence CLASS Transformation (Eclat) algorithm is based on a vertical representation of the transaction database and adopts a divide and conquer approach along with a depth-first search strategy to explore the itemset lattice in search of frequent itemsets.

Thanks to the vertical database format, the cost of support counting is reduced and it becomes possible to determine it during the candidate generation. In fact, the support of a k -itemset can be computed by intersecting the tidsets corresponding to all of its items, or alternatively, to two of its $(k - 1)$ -subsets whose join can produce it. On the other hand, the divide and conquer approach and the depth-first search strategy allow reducing the database scans by dividing the itemset lattice, i.e., the search space, in sub-lattices formed by itemsets that share a common prefix (equivalence classes), and, correspondingly, by dividing the transaction database into conditional databases, each one containing the transactions in the original dataset that refer to the itemsets in the corresponding sub-lattice.

The pseudocode for `ECLAT` is shown in Algorithm 2. The algorithm recursively explores the sub-lattices (equivalence classes) formed by the itemsets with a common prefix and compute their support in order to detect the frequent itemsets. Initially, the frequent items are considered already computed and used to define the starting equivalence classes. Once the equivalence class corresponding to a given prefix is completely explored (in a recursively depth-first search fashion), the algorithm proceeds by considering another class (i.e., the class of itemsets having a different prefix).

One of the weaknesses of Eclat is that the tidsets can be too long to be stored in the main memory and to efficiently compute their intersection. It is worth noting here that, regarding this problem, a variant called `dEclat` of the original algorithm has been proposed in which the performances of Eclat are improved through a careful shrinking of the size of the intermediate tidsets. This algorithm keeps track of the differences in the tidsets as opposed to the full tidsets.

FP-Growth

The FP-growth algorithm exploits a divide and conquer approach similar to Eclat, but with reference to an horizontal database format. It uses a special data structure, an augmented prefix tree called frequent pattern tree (FP-tree) which contains, in a compressed form, all of the information of the input transaction database needed to compute the itemset supports.

The vertices of the tree are labeled with a single item, and each child vertex represents a different item. Vertices also store the support information related to the itemset comprising all of the items on the path from the root to a given vertex. To build the FP-tree, we start with a tree containing the null item $\emptyset$ as root. For each transaction $T_i \in \mathcal{D}$, the itemset T_i is inserted into the FP-tree, and the count of all of the vertices along the path representing T_i is incremented. When T_i shares a common prefix with some previously inserted transaction, T_i will follow the same path until the common prefix. Regarding the remaining items in T_i , new vertices are inserted under the common prefix, and their count is initialized to 1. The FP-tree construction is complete when all of the transactions have been processed and inserted.

The usefulness of the FP-tree lies in being a prefix compressed representation of the database $\mathcal{D}$. In order to compact the FP-tree as much as possible, so that the most frequent items are found at the top, near the root, the algorithm reorders the items in decreasing order of support. Starting from the initial database, FP-growth determines the support of all of the single items $i \in \mathcal{U}$. Then, infrequent items are discarded, and frequent items sorted by decreasing support. Finally, each transaction $T_i \in \mathcal{D}$ is inserted into the FP-tree after reordering T_i by decreasing item support.

The algorithm now uses the FP-tree as an index in place of the original database. The frequent itemsets are mined from the tree as shown in Algorithm 3. The algorithm takes as input a FP-tree R built from the database $\mathcal{D}$, and the current itemset prefix P , which is initially empty.

FP-GROWTH builds projected FP-trees for each frequent item i in R in increasing order of support. The projection of R on item i is determined by finding all of the occurrences of i in the tree, and for each occurrence, computing the corresponding path from the root to i . The count of each item i on a given path, is stored in the variable $cnt(i)$. The path is then inserted into the new projected tree RX , where X is the itemset obtained by extending the prefix P with the item i .

Algorithm 1: *Apriori* algorithm

Input: $\mathcal{D}$, a database of transactions; U , a universe set of items; $minsup$, the minimum support

Output: the set $\mathcal{L}$ of frequent itemsets in $\mathcal{D}$

```

1  procedure APRIORI( $\mathcal{D}, U, minsup$ )
2     $\mathcal{L}_1 \leftarrow \text{FIND\_FREQUENT\_ITEMS}(\mathcal{D}, minsup)$ 
3     $k \leftarrow 2$ 
4    while  $\mathcal{L}_{k-1} \neq \emptyset$  do
5       $C_k \leftarrow \text{APRIORI\_GEN}(\mathcal{L}_{k-1})$ 
6      foreach transaction  $t \in \mathcal{D}$  do
7        foreach candidate  $c \in C_k$  do
8          if  $c \subset t$  then
9             $sup(c) \leftarrow sup(c) + 1$ 
10         end
11       end
12     end
13      $\mathcal{L}_k \leftarrow \{c \in C_k \mid sup(c) > minsup\}$ 
14      $k \leftarrow k + 1$ 
15   end
16    $\mathcal{L} \leftarrow \cup_k(\mathcal{L}_k)$ 
17   return  $\mathcal{L}$ 
18 end

19 procedure APRIORI_GEN( $\mathcal{L}_{k-1}$ )
20   foreach  $l_1 \in \mathcal{L}_{k-1}$  do
21     foreach  $l_2 \in \mathcal{L}_{k-1}$  do
22       if  $l_1$  and  $l_2$  share the first  $k - 2$  items and  $l_1[k - 1] < l_2[k - 1]$  then
23          $c \leftarrow l_1 \cup l_2$ 
24         if HAS_INFREQUENT_SUBSETS( $c, \mathcal{L}_{k-1}$ ) then
25           // if the candidate has a  $(k - 1)$ -subset that is not in  $\mathcal{L}_{k-1}$  discard it
26           discard  $c$ 
27         else
28            $C_k \leftarrow C_k \cup c$ 
29         end
30       end
31     end
32   return  $C_k$ 
33 end

```

Algorithm 2: *Eclat* algorithm**Input:** a vertical database of transactions; *minsup*, the minimum support**Output:** the collection of frequent itemsets in $\mathcal{F}$ // Initial call: $\mathcal{F} \leftarrow \emptyset, P \leftarrow \{(i, T(i)) : i \in \mathcal{U}, |T(i)| \geq \text{minsup}\}$

```

1 procedure ECLAT( $P, \text{minsup}, \mathcal{F}$ )
2   foreach  $(X_a, T(X_a)) \in P$  do
3      $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X_a, \text{sup}(X_a))\}$ 
4      $P_a \leftarrow \emptyset$ 
5     foreach  $(X_b, T(X_b)) \in P$ , with  $X_b > X_a$  do
6        $X_{ab} \leftarrow X_a \cup X_b$ 
7        $T(X_{ab}) \leftarrow T(X_a) \cap T(X_b)$ 
8       if  $\text{sup}(X_{ab}) \geq \text{minsup}$  then
9          $P_a \leftarrow P_a \cup \{(X_{ab}, T(X_{ab}))\}$ 
10      end
11    end
12    if  $P_a \neq \emptyset$  then
13      ECLAT( $P_a, \text{minsup}, \mathcal{F}$ )
14    end
15  end
16 end

```

Algorithm 3: FP-GROWTH algorithm**Input:** $\mathcal{D}$, a database of transactions; *minsup*, the minimum support**Output:** the collection of frequent itemsets in $\mathcal{F}$ // Initial call: $R \leftarrow \text{FP-tree}(\mathcal{D}), P \leftarrow \emptyset, \mathcal{F} \leftarrow \emptyset$

```

1 procedure FP-GROWTH( $R, P, \mathcal{F}, \text{minsup}$ )
2   Remove infrequent items from  $R$ 
3   if ISPATH( $R$ ) then                                     // insert subsets of  $R$  into  $\mathcal{F}$ 
4     foreach  $Y \subseteq R$  do
5        $X \leftarrow P \cup Y$ 
6        $\text{sup}(X) \leftarrow \min_{x \in Y} \{\text{cnt}(x)\}$ 
7        $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
8     end
9   else                                                     // process projected FP-trees for each frequent item  $i$ 
10    foreach  $i \in R$  in increasing order of  $\text{sup}(i)$  do
11       $X \leftarrow P \cup \{i\}$ 
12       $\text{sup}(X) \leftarrow \text{sup}(i)$                                // sum of  $\text{cnt}(i)$  for all of the nodes labeled  $i$ 
13       $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
14       $R_X \leftarrow \emptyset$                                    // projected FP-tree for  $X$ 
15    end
16  end
17  foreach  $\text{path} \in \text{PATHFROMROOT}(i)$  do
18     $\text{cnt}(i) \leftarrow \text{count of } i \text{ in path}$ 
19    Insert  $\text{path}$ , excluding  $i$ , into FP-tree  $R_X$  with count  $\text{cnt}(i)$ 
20  end
21  if  $R_X \neq \emptyset$  then
22    FP-GROWTH( $R_X, X, \mathcal{F}, \text{minsup}$ )
23  end
24 end

```

When a path is inserted, the count of each vertex in RX along the given path is incremented by the corresponding path count $cnt(i)$. The item i is omitted from the path, since it now belongs to the prefix. The derived FP-tree is the projection of the itemset X that includes the current prefix extended with item i . The algorithm is then recursively invoked with projected FP-tree RX and the new prefix itemset X as input parameters. The recursion ends when the input FP-tree R is just a single path. In this case, all of the itemsets that are subsets of the path are enumerated, and the support of each such itemset is determined by the least frequent item in it.

Conclusions

We presented an overview of a fundamental data mining task, the mining of frequent itemsets and association rules of large data sets. We started with an introduction to the problem in the more general context of data mining and a sample of its possible applications. Then, we reviewed the basic concepts and definitions regarding frequent itemset mining and the main interestingness measures of the usefulness and informative capacity of association rules. At last, we described some of the most important classical algorithms that deal with frequent itemset mining.

See also: Data Mining in Bioinformatics. Identification of Homologs. Identification of Proteins from Proteomic Analysis. Population Analysis of Pharmacogenetic Polymorphisms. The Challenge of Privacy in the Cloud

References

- Aggarwal, C.C., 2015. Data Mining: The Textbook. Springer.
- Aggarwal, C.C., Han, J., 2014. Frequent Pattern Mining. Springer.
- Agrawal, R., Imielinski, T., Swami, A., 1993. Mining association rules between sets of items in large databases. *ACM SIGMOD Record* 22 (2), 207–216.
- Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. In: *Proceedings 20th International Conference Very Large Data Bases (VLDB)*, Vol. 1215, pp. 487–499.
- Atluri, G., Gupta, R., Fang, G., *et al.*, 2009. Association analysis techniques for bioinformatics problems. *Bioinformatics and Computational Biology*. Springer. pp. 1–13.
- Beil, F., Ester, M., Xu, X., 2002. Frequent term-based text clustering. In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 436–442.
- Boley, D., Gini, M., Gross, R., *et al.*, 1999. Partitioning-based clustering for web document categorization. *Decision Support Systems* 27 (3), 329–341.
- Brin, S., Motwani, R., Silverstein, C., 1997a. Beyond market baskets: Generalizing association rules to correlations. *ACM Sigmod Record*, 26. ACM. pp. 265–276.
- Brin, S., Motwani, R., Ullman, J.D., Tsur, S., 1997b. Dynamic itemset counting and implication rules for market basket data. *ACM SIGMOD Record*, 26. ACM. pp. 255–264.
- Fung, B.C., Wang, K., Ester, M., 2003. Hierarchical document clustering using frequent itemsets. *SDM*, vol. 3. SIAM. pp. 59–70.
- Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Elsevier.
- Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. *ACM SIGMOD Record* 29 (2), 1–12.
- Iváncsy, R., Vajk, I., 2006. Frequent pattern mining in web log data. *Acta Polytechnica Hungarica* 3 (1), 77–90.
- Lee, W., Stolfo, S.J., Mok, K.W., 1998. Mining audit data to build intrusion detection models. In: *KDD*, pp. 66–72.
- Leung, K., Leckie, C., 2005. Unsupervised anomaly detection in network intrusion detection using clusters. In: *Proceedings of the 28th Australasian Conference on Computer Science*, vol. 38. Australian Computer Society, Inc., pp. 333–342.
- Lin, W., Alvarez, S.A., Ruiz, C., 2002. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery* 6 (1), 83–105.
- Mobasher, B., Dai, H., Luo, T., Nakagawa, M., 2001. Effective personalization based on association rule discovery from web usage data. In: *Proceedings of the 3rd International Workshop on Web Information and Data Management*, ACM, pp. 9–15.
- Mobasher, B., Jain, N., Han, E.-H., Srivastava, J., 1996. Web mining: Pattern discovery from world wide web transactions. Technical Report TR96-050, Department of Computer Science, University of Minnesota.
- Naulaerts, S., Meysman, P., Bittremieux, W., *et al.*, 2015. A primer to frequent itemset mining for bioinformatics. *Briefings in Bioinformatics* 16 (2), 216–231.
- Rajaraman, A., Ullman, J.D., Ullman, J.D., Ullman, J.D., 2012. Mining of Massive Datasets, 1. Cambridge: Cambridge University Press.
- Zaki, M.J., 2000. Scalable algorithms for association mining. *Knowledge and Data Engineering, IEEE Transactions* 12 (3), 372–390.
- Zaki, M.J., Wagner Meira, J., 2014. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press.

Association Rules and Frequent Patterns

Giuseppe Di Fatta, University of Reading, Reading, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

Association Rule Mining (ARM) (Agrawal *et al.*, 1993; Agrawal and Srikant, 1994; Hipp *et al.*, 2000) is often referred to as frequent pattern mining (Goethals, 2003; Han *et al.*, 2007; Aggarwal and Han, 2014; Fournier-Viger *et al.*, 2017). Frequent pattern mining focuses on the extraction of the frequent patterns, ARM also offers a specific and more descriptive representation of the frequent patterns in the form of association rules: If the pattern X is present, then also the pattern Y is. Anyway, since ARM relies on the extraction of the frequent patterns, which is a particularly complex task, the two are often considered equivalent, at least, from the computational point of view. In the most common formulation of the ARM problem, patterns take the form of sets and, more specifically, of sets of items (itemsets). An item is a binary feature, such as the presence of a specific attribute or property. In general, ARM is applied to extract and explicitly represent events (attributes) that occur together in the data and more frequently than others. The assumption is that events that frequently occur together are more important of those not occurring together or not frequently enough. For example, in market basket analysis the itemset is the set of items purchased by a customer within a single transaction. The interesting relations to be exposed are the explicit or implicit associations made by customers in emerging shopping behaviours.

In the life sciences the adoption of ARM to address some data-intensive problems (Atluri *et al.*, 2009) did not receive the same attention as for other statistical and data mining techniques, such as correlation, regression, classification and clustering. However, since the widespread adoption of genomic sequencing and other high-throughput digital technologies, very large datasets have become more common and often publicly available. For example, mining association rules in genomic data (Hanash, 2003; Alves *et al.*, 2010; Oellrich *et al.*, 2014; Chen *et al.*, 2015) has become of more interest and adopted for two main tasks: identifying the significant patterns in subsets of genes and between gene regulation and phenotype.

Specific domain knowledge is useful to understand if the representation of the itemsets as binary attributes may force an undesired asymmetric semantics. The presence of an attribute in a sample is encoded by the presence of an item in the itemset: the item typically has an explicit positive meaning. The lack of an attribute often is not significant and is not explicitly encoded in the itemset. However, when this is not the case and the lack of some attribute is of interest, a specific 'negative' item can be used to encode this information explicitly in the itemsets. Alternatively Negative Association Rules (Antonie and Zañane, 2004) can be applied to extract relations, for example, between the absence of an item and the presence of others.

Arguably some limiting factors have hindered a more widespread adoption of ARM. Firstly, the discovery of interesting patterns from data requires complex combinatorial algorithms and often these benefit from high performance computing. For real-world problems the search space may be prohibitively large. Secondly, it is not uncommon that even for a relatively small input dataset, the set of discovered patterns is very large, even larger than the input dataset. In this case ARM is only a first step in more complex data workflows that include other data mining techniques. For these reasons, a multidisciplinary approach that combines expertise from both computer science and the specific application domain is often critical.

The rest of the article is organised as follows. Section "Problem Definition" introduces the basic ARM problem and discuss the combinatorial nature of the computational task. Some important ARM algorithms, the search space of patterns and some interestingness criteria are presented in Section "ARM Algorithms". Some extended ARM problems are briefly reviewed in Section "Extended Association Rule Mining" and, finally, Section "Conclusions" provides some conclusive remarks.

Problem Definition

Consider a set of binary attributes $\mathcal{I} = \{A, B, C, \dots\}$ ($|\mathcal{I}| = d$) called items. A proper subset $s \subset \mathcal{I}$ is referred to as *itemset*. A k -itemset s is an itemset of k items, i.e., $|s| = k$. A transaction over $\mathcal{I}$ is a pair $t = \langle id, s \rangle$, where id is the transaction identifier and s is an itemset. $\mathcal{T}$ is the set of transactions ($|\mathcal{T}| = n$), a transactional database. A transaction $t = \langle id, s \rangle$ is said to support an itemset x , if $s \supseteq x$. For example, the transaction $\langle 0F36A, \{BCDH\} \rangle$ supports the itemset $\{BD\}$ and it does not support the itemset $\{BDE\}$. The support σ of an itemset x is the fraction of transactions that support x , i.e., $\sigma(x) = \frac{|\{ \langle id, s \rangle \in \mathcal{T} : s \supseteq x \}|}{n}$.

An association rule (Agrawal and Srikant, 1994) is an implication expressed in the form $X \Rightarrow Y$ (" X implies Y ", " X then Y "), where X and Y are itemsets ($X, Y \subset \mathcal{I}$) with $X \cap Y = \emptyset$. The left term is referred to as the antecedent, the right term as the consequent of the rule. The 'support' of a rule is the fraction of transactions that support $X \cup Y$, which corresponds to the empirical joint probability $P(X \cup Y)$.

$$\text{support}(X \Rightarrow Y) = \sigma(X \cup Y) \quad (1)$$

The 'confidence' of a rule is the fraction of transactions supporting X that also support Y . The confidence of a rule corresponds to the empirical conditional probability $P(Y|X)$.

$$\text{confidence}(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \quad (2)$$

A rule is said 'frequent', if it is supported by at least a minimum fraction of transactions in $\mathcal{I}$. A user-defined parameter for the minimum support (*minsup*) is required for this purpose. A rule is said 'strong', if it has a confidence greater or equal to a user-defined parameter *minconf*.

Given a transactional database $\mathcal{I}$ and two user-defined parameters *minsup* and *minconf*, the Association Rule Mining problem is to find all frequent and strong rules.

$$\begin{aligned} X \Rightarrow Y, \quad \text{where } X, Y \subset \mathcal{I}, \quad X \cap Y = \emptyset, \\ \sigma(X \cup Y) \geq \text{minsup} \quad \text{and} \quad \frac{\sigma(X \cup Y)}{\sigma(X)} \geq \text{minconf} \end{aligned} \quad (3)$$

The Search Space

The search space of the ARM problem is limited by all possible combinations for the antecedent and the consequent of a rule. The total number of itemsets is the cardinality of the power set, $|\text{Power}(\mathcal{I})| = 2^d$.

Although the disjoint constraint reduced the number of possible combinations, they are still expected to be exponential in the number d of items. The total number R of possible association rules is given by all possible itemset combinations for the antecedent times all possible itemset combinations for the consequent, as expressed by Eq. (4). There are $\binom{d}{k}$ possible k -itemsets as antecedent and, given a k -itemset as antecedent, there are $\binom{d-k}{j}$ possible j -itemsets as consequent.

The total number R of possible association rules is given by

$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \cdot \sum_{j=1}^{d-k} \binom{d-k}{j} \right] = 3^d - 2^{d+1} + 1 \quad (4)$$

where the following equation derived from the binomial theorem has been applied for the reduction of the expression.

$$\sum_{k=0}^m \binom{m}{k} \cdot r^k = (1+r)^m \quad (5)$$

For example, a set of 10 items generates 57,002 rules; while 20 items generate almost 3.5 billion rules. Obviously a brute force approach that enumerates all possible rules and validates them against the minimum support and confidence constraints, is not a feasible solution to the ARM problem.

The computational complexity of the problem with binary attributes and its other variants (e.g., a quantitative approach) has been shown to be NP-complete (Wijzen and Meersman, 1998; Yang, 2004; Angiulli et al., 2001). The challenging combinatorial nature of the problem has given rise to an intense research (Agrawal et al., 1993; Agrawal and Srikant, 1994, 1986; Hipp et al., 2000; Zaki et al., 1997; Savasere et al., 1995; Toivonen, 1996; Brin et al., 1997; Han et al., 2000) and a number of efficient algorithms, the more representative of which are briefly reviewed in the next section.

ARM Algorithms

The observation that the computation of the support of the rule $\sigma(X \cup Y)$ is necessary in both constraints (Eqs. (1) and (2)) suggests an efficient problem decomposition into two sub-problems, which was originally introduced in the *Apriori* algorithm (Agrawal and Srikant, 1994).

1. Find all Frequent Itemsets (FI):

$$FI = \{x | x \subset \mathcal{I} \quad \text{and} \quad \sigma(x) \geq \text{minsup}\}$$

2. Generate all strong rules from FI.

In the second step only the confidence of the rules has to be tested, as any rule generated from the FI set is guaranteed to have sufficient support. The second problem can be solved by a straightforward polynomial time algorithm and most of the computational complexity lies in the first problem. All the possible itemsets correspond to the powerset $\mathcal{P}(\mathcal{I})$, where $|\mathcal{P}(\mathcal{I})| = 2^d$. This is still an exponential search space, though $3^d - 2^{d+1} + 1 > 2^d$ for $d > 2$, and it requires efficient combinatorial algorithms, optimisation techniques, high performance computing and, ultimately, additional domain-specific constraints to limit the search space.

The powerset $\mathcal{P}(\mathcal{I})$ can be represented as a lattice of sets. For example, Fig. 1 shows the lattice of all possible itemsets for a set of four items, which are represented as natural numbers ($\mathcal{I} = \{1, 2, 3, 4\}$). The binomial coefficients on the left indicates the number of subsets at each tier of the lattice and the edges indicate a subset-superset relation between itemsets in consecutive tiers. Fig. 2

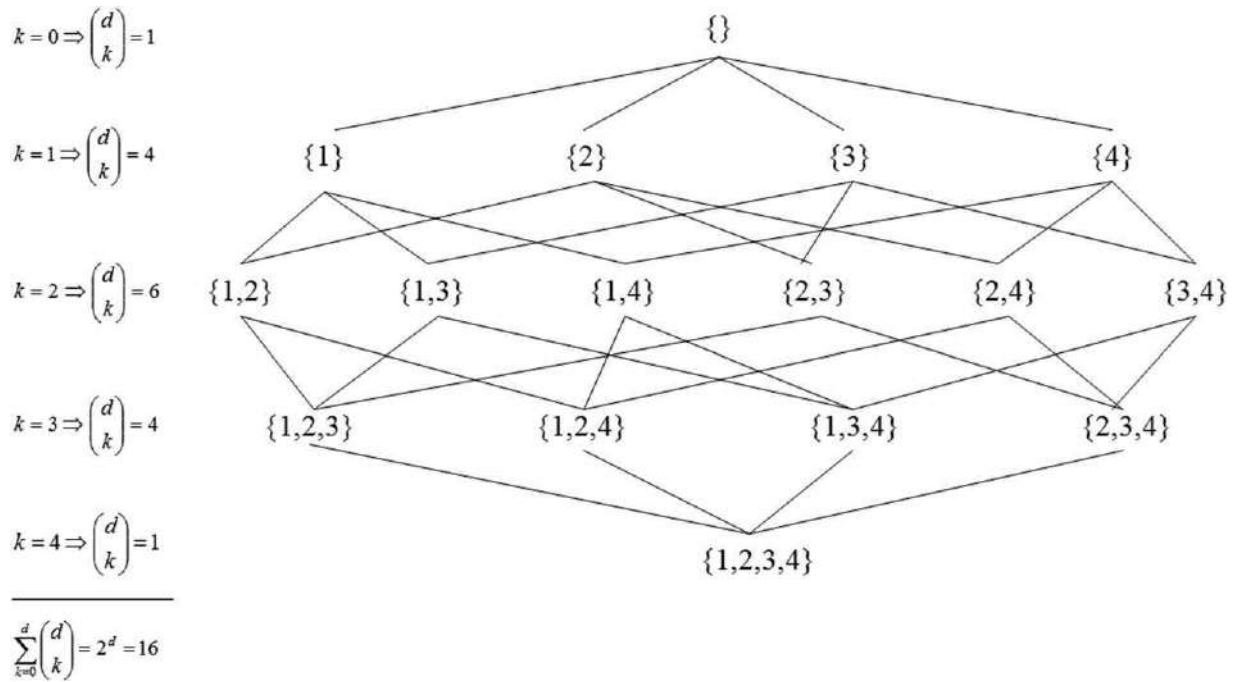


Fig. 1 Lattice of itemsets: Powerset for $\mathcal{I}=\{1,2,3,4\}$ ($d=4$).

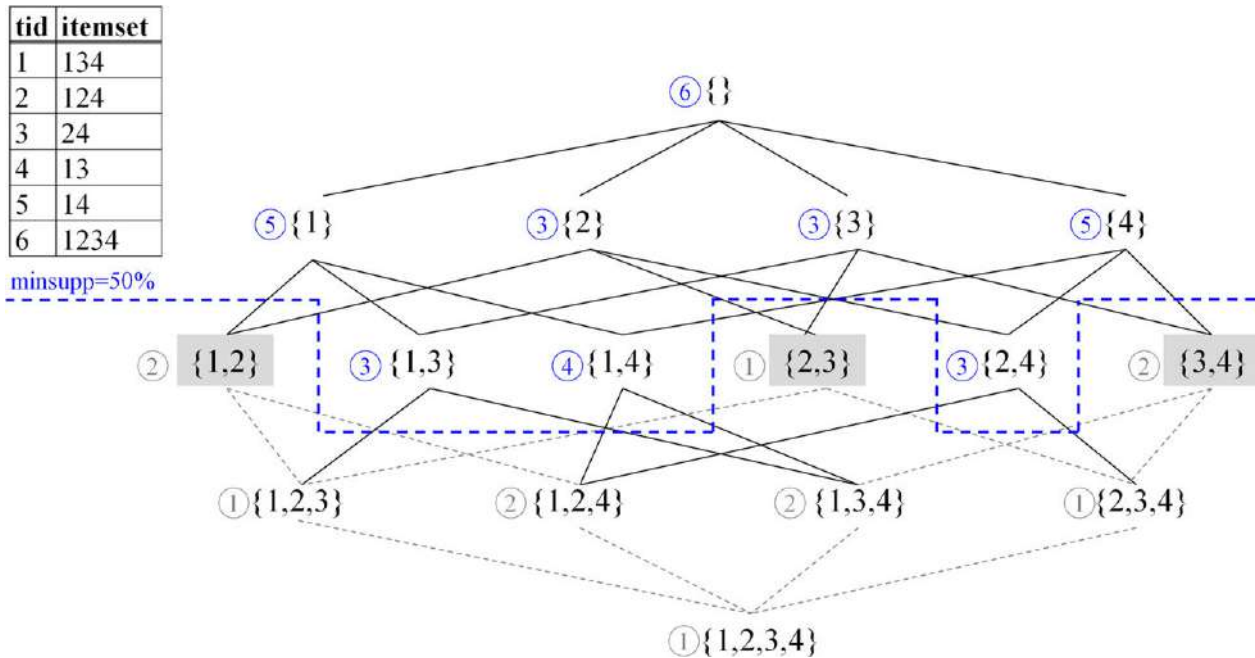


Fig. 2 Border between frequent and infrequent itemsets: The number in the circle indicates the support of the itemset.

provides an example of the border of the frequent itemsets for the transactions listed in the table and for a minimum support of 3 transactions (50%).

The main algorithmic strategy is to generate candidate frequent itemsets and test their support efficiently and without any redundancy, considering the computing system limitations. In most real-world applications the search space is too large to fit in main memory. Moreover, in spite of the exponential complexity optimisation techniques can provide a sufficient run time efficiency to tackle problems large enough to be of interest.

To this aim a number of optimisation techniques have been proposed, which refer to the strategy of visiting the lattice (breadth vs depth first), to the generation of the candidate itemsets, and to the data structure and the algorithm for testing the constraint of the minimum support of candidate itemsets.

In particular, one fundamental optimisation technique is based on the anti-monotonicity (or downward-closure) property of the support. Any superset of an itemset cannot have a support greater than its subset. In Fig. 2, the infrequent itemsets are highlighted in grey: the edges from them moving forward to the next tier are represented as dashed lines to indicate that their supersets are implicitly, a priori, known to fail the support constraint. The algorithm *Apriori* (Agrawal and Srikant, 1994) was the first to introduce and exploit these concepts with the so-called *Apriori* ‘pruning’ optimisation.

Among the many algorithms proposed in the last two decades or so, three of them have a particular importance as each introduced a specific novel approach that inspired many others. The algorithm *Apriori* (Agrawal and Srikant, 1994) adopts a breadth-first strategy by generating the itemsets of a single tier at a time. The algorithm *Eclat* (Zaki, 2000) adopts the opposite depth-first strategy reducing the memory requirements. The algorithm *FP-growth* (Han et al., 2000) also adopts a depth-first search and introduced a clever tree-based data structure to improve the running time significantly.

Efficient implementations of algorithms for mining frequent itemsets are available as executable or source code (Borgelt, 2003, 2012, 2017), or by means of extensions of data science development environments, such as *arules* (Hornik et al., 2005), which is an R package for generating, managing and analysing frequent itemsets and association rules.

Interestingness Measures and Criteria

Interesting correlations among items and itemsets in patterns and association rules may not always be represented only by frequency (support and confidence). For example, the support does not consider normalisation of the frequencies of the items and may be skewed towards highly frequent items. Moreover, when patterns are considered interesting only on their frequency, the number of discovered patterns is often very large with a significant redundancy.

For these reasons other interestingness criteria and measures have also been considered. In some cases additional domain-specific measures can be used for ranking the patterns and prioritise their analysis. In others, pruning unnecessary and redundant patterns is a possible approach.

In applications of frequent itemset mining, the user typically applies an exploratory and iterative approach starting from large values of the support threshold (*minsupp*). Such large values of *minsupp* are chosen so that the number of frequent itemsets is small. Unfortunately, this often leads to frequent itemsets of small size, which may not be particularly interesting. Hence, in order to find more interesting frequent itemsets, the support threshold needs to be set to smaller values, for which the number of the frequent itemsets can be quite large, possibly too large for user inspection and analysis.

It turns out that some of the frequent itemsets may not be of particular interest as they have identical support as their supersets. For example, consider the two frequent itemsets 3 and 13 in Fig. 2, they are supported by the same three transactions (*id* = 1, *id* = 4 and *id* = 6). In this case, the frequent itemset 13 is ‘maximal’ within the three supporting transactions, while 3 is not.

Following this consideration, a method of reducing the large amount of frequent itemsets is to use one of the so-called compact or condensed representations, such as the Maximal Frequent Itemsets (MFI) and the Closed Frequent Itemsets (CFI) (Uno et al., 2004).

MFI are frequent itemsets for which none of their supersets is also frequent. Clearly, any frequent itemset is a subset of a maximal frequent itemset. In other words, a frequent itemset is maximal if it is not a proper subset of any other frequent itemset.

A frequent itemset is closed if it contains all items that occur in all transactions in which it is bought, i.e., it is the intersection of its supporting transactions. In other words, a closed frequent itemset is a frequent itemset whose support is higher than the supports of all its proper supersets. In particular, all maximal frequent itemsets are also closed.

The underlying idea of both definitions is that the set of all maximal or closed frequent itemsets can be used as a compact representation of all frequent itemsets: It can be seen as a form of compression.

The set MFI allows to reconstruct all frequent itemsets by simply generating all their subsets. However, the support of these subsets cannot be reconstructed: the support of a frequent itemset can be different from the support of its maximal itemset. An additional database scan is needed to reconstruct this information, if necessary. MFI can be seen as a form of compression with loss of information.

This issue is overcome with the set CFI. The support of a non-closed itemset is the support of the smallest superset that is closed. CFI can be seen as a form of compression without loss of information.

Many other sampling criteria and definitions for subsets of frequent itemsets have also been proposed and include free frequent itemsets (Boulicaut et al., 2003), non-derivable itemsets (Calders and Goethals, 2007) and margin-closed frequent itemsets (Moerchen et al., 2011).

If the number of itemsets in a condensed subset is still too large for a manual analysis, then other data mining techniques can be applied to the complete set of frequent itemsets, or one of its subsets, in order to generate a model or a view at a higher level of abstraction (e.g., (Di Fatta et al., 2006)).

Extended Association Rule Mining

The basic concept of mining association rules and frequent patterns can be extended in a number of ways in order to gain more generality or to account for other attribute types (e.g., numerical or temporal attributes) in the data.

Generalised Association Rules (Sarawagi and Thomas, 1998) adopt a hierarchical taxonomy (concept hierarchy) of the items. Considering that coke and pepsi are soft drinks and assuming they are frequently bought with chips, nuts or crackers, a generalised rule may be express that “60% of transactions that contain soft drinks also contain snacks”.

Quantitative Association Rules (Srikant and Agrawal, 1996) considers both quantitative and categorical attributes in the data. For example, the rule “10% of married people between age 50 and 60 have at least 2 cars” is able to associate a qualitative attribute with a quantitative one. In Interval Data Association Rules (Miller and Yang, 1997) the range of quantitative attributes is partitioned: For example, age can be partitioned into 5-year-increment intervals.

Mining Maximal Association Rules and Closed Association Rules (Uno *et al.*, 2004) directly discovers the condensed subsets of the frequent itemsets, as discussed in Section “Interestingness Measures and Criteria”.

Sequential Association Rules (Sarawagi and Thomas, 1998) consider temporal data and discover subsequences: For example, users frequently buy first a PC, then a printer and, finally, a digital camera. Sequential pattern mining (Mabroukeh and Ezeife, 2010) can be applied to a variety of domains where the order of the items is relevant, e.g., Web log data and DNA sequences (Abouelhoda and Ghanem, 2010).

In Frequent Subgraph Mining (Kuramochi and Karypis, 2001) the data in the transactions are in the form of graphs. These may include many types of networks, such as social networks, protein-protein interaction networks (Shen *et al.*, 2012) and molecular compounds (Deshpande *et al.*, 2002). Finding frequent subgraphs in a set of graphs involves graph and subgraph isomorphism testing, which is more complex than subset testing required in frequent itemsets mining, and efficient and parallel computing approaches may be required for large datasets (Di Fatta and Berthold, 2005, 2006).

ARM Applications

Association rule and, in general, frequent pattern mining was originally inspired and motivated by market basket and customer behaviour analysis (Agrawal *et al.*, 1993; Agrawal and Srikant, 1994) and over the last two decades it has been successfully applied to many other applications domains. Various applications have been developed initially in other computer science fields, such as software bug discovery, failure and event detection in telecommunication and computer networks, WWW user behaviour analysis, and later in multidisciplinary domains including bioinformatics, chemoinformatics and data-driven medical applications.

Arguably the most popular example of association rules was extracted from supermarket transactions to reveal buying behaviour of customers: $IF\{DIAPERS\} \Rightarrow \{BEER\}$. This alleged association rule is often used to underline the nature of the correlations emerging from data: there is no implication of causality, only co-occurrence. The goal of ARM is to systematically extract all frequent and strong rules with an empirical (data-driven) approach. Domain experts may be able to gain useful insights from the generated rules and use them to support decision making, for example, in shelf management, marketing and sale promotions. The true story behind the example of diapers and beer is that in 1992, a retail consulting group at Teradata analysed 1.2 million customer transactions from Osco Drug stores. SQL self joins queries, not ARM, were used to identify correlations between (expensive) baby's products and any other product. The analysis discovered that between 5 and 7 p.m. consumers often bought diapers and beer. Apparently no attempt was made to exploit this correlation: Beers were not relocated near diapers. Nevertheless, this unexpected correlation raised much attention in the data analytics and mining community, and inspired a stream of research on association rules.

In telecommunication and computer networks, sequential pattern mining can be applied to the detection of events and anomalies (Cui *et al.*, 2014), to the analysis of user behaviour from Web logs (Pei *et al.*, 2000) and of online social media content (Adedoyin-Olowe *et al.*, 2013).

Detection of software bugs is another interesting example of ARM applications. Errors and faults leading to unexpected results, i.e., non-crashing bugs, is a difficult case of software bug detection (Liu *et al.*, 2005). For example in (Fatta *et al.*, 2016), function calls are recorded during software test executions and are analysed with a frequent pattern mining algorithm. Frequent subgraphs in failing test executions that are not frequent in successful test executions are used to rank the functions according to their likelihood of containing a fault.

Frequent pattern mining has been applied to biological data (Rigoutsos and Floratos, 1998) as these may come in the form of sequences (e.g., Microarray data (Cong *et al.*, 2004, 2005) and RNA (Chevalet and Michot, 1992)) and graphs (e.g., protein-to-protein interaction networks (Cho and Zhang, 2010), phylogenetic trees (Shasha *et al.*, 2004), molecular compounds (Di Fatta and Berthold, 2005, 2006; Deshpande *et al.*, 2005)).

In real-world applications frequent pattern and association rule mining is often one step in a more complex data analytics workflow, which may include other data mining techniques, such as classification and clustering, and data visualisation. The set of discovered patterns or rules can even be larger than the input dataset and a direct user inspection is not feasible, nor desirable. In some cases, frequent pattern mining can be seen as a feature generation step in the knowledge discovery process. For example, in (Di Fatta *et al.*, 2006) the frequent patterns are used to identify candidate drugs for a target activity and are considered as attributes in a very high dimensional space. A self-organising map is used to generate a 2-dimensional map of drugs, where proximity in the map indicates that activity against a target disease is due to a similar molecular substructure.

Conclusions

This article provides a brief survey of association rule and frequent pattern mining, which is one of the most important method in data mining for data-driven scientific discovery. A large body of literature has been produced in the past two decades or so. From the early introduction of the problem definition and the algorithms for market basket analysis, more efficient methods, extended

problem definitions, advanced interestingness criteria and a plethora of applications in many diverse domains are nowadays available. In particular, the application domains have gradually expanded to cover a wide range, including software engineering, computer networks, bioinformatics, chemoinformatics and many other life sciences and scientific domains.

A general problem definition is presented and the combinatorial complexity of the computational task discussed. In particular, the exact exponential number of possible rules that are implicitly defined by a set of items is derived. Some of the most relevant algorithms, the fundamental and advanced interestingness criteria are presented. The most relevant extended problems and some application domains are also discussed with examples.

ARM is arguably one of the most elegant data mining problems that has fascinated the data mining community for many years and has a great potential to contribute to many scientific domains. It can provide a formidable tool for exploratory investigations to reveal intriguing insights from data, and ultimately lead to unexpected data-driven scientific discoveries.

See also: Data Mining in Bioinformatics. Identification of Homologs. Identification of Proteins from Proteomic Analysis. Next Generation Sequencing Data Analysis. Population Analysis of Pharmacogenetic Polymorphisms. The Challenge of Privacy in the Cloud

References

- Abouelhoda, M., Ghanem, M., 2010. *String Mining in Bioinformatics*. Springer. pp. 207–247.
- Adedoyin-Olowe, M., Gaber, M.M., Stahl, F., 2013. TRCM: A methodology for temporal analysis of evolving concepts in twitter. In: *Proceedings of 12th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2013, Part II*, pp. 135–145. Zakopane, Poland: Springer.
- Aggarwal, C.C., Han, J., 2014. *Frequent Pattern Mining*. Springer.
- Agrawal, R., Imieliński, T., Swami, A., 1993. Mining association rules between sets of items in large databases. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD '93*, pp. 207–216. New York, NY: ACM.
- Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94*, pp. 487–499. Morgan: Kaufmann.
- Alves, R., Rodriguez-Baena, D.S., Aguilar-Ruiz, J.S., 2010. Gene association analysis: A survey of frequent pattern mining from gene expression data. *Briefings in Bioinformatics* 11 (2). 210–224.
- Angiulli, F., Ianni, G., Palopoli, L., 2001. On the complexity of mining association rules. In: *Proceedings of the Nono Convegno Nazionale su Sistemi Evoluti di Basi di Dati (SEBD)*, SEBD, pp. 177–184.
- Antonie, M.-L., Zaiiane, O.R., 2004. Mining positive and negative association rules: An approach for confined rules. In: *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, vol. 3202 of PKDD, pp. 27–38. Springer.
- Atluri, G., Gupta, R., Fang, G., *et al.*, 2009. Association analysis techniques for bioinformatics problems. In: *Proceedings of the First International Conference on Bioinformatics and Computational Biology*, pp. 1–13. Berlin, Heidelberg: Springer.
- Borgelt, C., 2003. Efficient implementations of apriori and eclat. In: *Proceedings of Workshop of Frequent Item Set Mining Implementations, FIMI*. Melbourne, FL.
- Borgelt, C., 2012. Frequent item set mining, *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery* 2 (6). 437–456.
- Borgelt, C., 2017. Implementations of various data mining algorithms. Available at: <http://www.borgelt.net/software.html>.
- Boulicaut, J.-F., Bykowski, A., Rigotti, C., 2003. Free-sets: A condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery* 7 (1). 5–22.
- Brin, S., Motwani, R., Ullman, J., Tsur, S., 1997. Dynamic itemset counting and implication rules for market basket data. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 255–264. ACM Press.
- Calders, T., Goethals, B., 2007. Non-derivable itemset mining. *Data Mining and Knowledge Discovery* 14 (1). 171–206.
- Chevalet, C., Michot, B., 1992. An algorithm for comparing rna secondary structures and searching for similar substructures. *Computer Applications in the Biosciences* 8 (3). 215–225.
- Cho, Y.R., Zhang, A., 2010. Predicting protein function by frequent functional association pattern mining in protein interaction networks. *IEEE Transactions on Information Technology in Biomedicine* 14 (1). 30–36.
- Cong, G., Tan, K.-L., Tung, A.K.H., Xu, X., 2005. Mining top-k covering rule groups for gene expression data. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD '05*, pp. 670–681. New York, NY: ACM Press.
- Cong, G., Tung, A.K.H., Xu, X., Pan, F., Yang, J., 2004. Farmer: Finding interesting rule groups in microarray datasets. In: *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD '04*, pp. 143–154. ACM.
- Cui, H., Yang, J., Liu, Y., Zheng, Z., Wu, K., 2014. Data mining-based dns log analysis, *Annals of Data. Science* 1 (3). 311–323.
- Deshpande, M., Kuramochi, M., Karypis, G., 2002. Automated approaches for classifying structures. In: *Proceedings of the 2nd International Conference on Data Mining in Bioinformatics, BIOKDD'02*, pp. 11–18. Springer.
- Deshpande, M., Kuramochi, M., Wale, N., Karypis, G., 2005. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering* 17 (8). 1036–1050.
- Di Fatta, G., Berthold, M.R., 2005. High performance subgraph mining in molecular compounds. In: *Proceedings of the International Conference on High Performance Computing and Communications (HPCC)*, LNCS, pp. 866–877. Springer.
- Di Fatta, G., Berthold, M.R., 2006. Dynamic load balancing in distributed mining of molecular compounds. In: *Proceedings of the IEEE Transactions on Parallel and Distributed Systems, Special Issue on High Performance Computational Biology*, pp. 773–785.
- Di Fatta, G., Fiannaca, A., Rizzo, R., *et al.*, 2006. Context-aware visual exploration of molecular databases. In: *Proceedings of the Sixth IEEE International Conference on Data Mining – Workshops (ICDMW'06)*, pp. 136–141.
- Di Fatta, G., Leue, S., Stegantova, E., 2016. Discriminative pattern mining in software fault detection. In: *Proceedings of the 3rd International Workshop on Software Quality Assurance (SOQUA)*, 14th ACM Symposium on Foundations of Software Engineering (ACM SIGSOFT), pp. 62–69. ACM.
- Fournier-Viger, P., Lin, J.C.-W., Vo, B., *et al.*, 2017. A survey of itemset mining, *Wiley interdisciplinary reviews. Data Mining and Knowledge Discovery* 7 (4).
- Goethals, B., 2003. Survey on frequent pattern mining. Technical report. Helsinki Institute for Information Technology.
- Hanashi, C.C.S., 2003. Mining gene expression databases for association rules. *Bioinformatics* 19 (1). 79–86.
- Han, J., Cheng, H., Xin, D., Yan, X., 2007. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery* 15 (1). 55–86.
- Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00*, pp. 1–12. ACM.

- Hipp, J., Guntzer, U., Nakhaeizadeh, G., 2000. Algorithms for association rule mining – A general survey and comparison. *SIGKDD Explorations Newsletter* 2 (1). 58–64.
- Hornik, K., Grün, B., Hahsler, M., 2005. *Arules – A computational environment for mining association rules and frequent item sets*, Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery 14 (15). 1–25.
- Kuramochi, M., Karypis, G., 2001. Frequent subgraph discovery. In: *Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM '01*, IEEE Computer Society, pp. 313–320.
- Liu, C., Yan, X., Yu, H., Han, J., Yu, P.S., 2005. Mining behavior graphs for “backtrace” of noncrashing bugs. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 286–297.
- Mabroukeh, N.R., Ezeife, C.I., 2010. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys* 43 (1). 1–3.
- Miller R.J., Yang Y., 1997. Association rules over interval data. In: *Proceedings of the ACM 1997 SIGMOD International Conference on Management of Data, SIGMOD '97*, pp. 452–461. ACM.
- Moerchen, F., Thies, M., Ultsch, A., 2011. Efficient mining of all margin-closed itemsets with applications in temporal knowledge discovery and classification by compression. *Knowledge and Information Systems* 29 (1). 55–80.
- Oellrich, A., Jacobsen, J., Papatheodorou, I., Smedley, D., 2014. Using association rule mining to determine promising secondary phenotyping hypotheses. *Bioinformatics* 30 (12).
- Pei, J., Han, J., Mortazavi-Asl, B., Zhu H., 2000. Mining access patterns efficiently from web logs. In: *Proceedings of the Knowledge Discovery and Data Mining, Current Issues and New Applications: 4th Pacific-Asia Conference, PAKDD 2000*, pp. 396–407. Kyoto, Japan: Springer.
- Rigoutsos, I., Floratos, A., 1998. Combinatorial pattern discovery in biological sequences: The teiresias algorithm. *Bioinformatics* 14 (1). 55–67.
- Sarawagi, S., Thomas, S., 1998. Mining generalized association rules and sequential patterns using sql queries. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '98*, ACM.
- Savasere, A., Omiecinski, E., Navathe, S., 1995. An efficient algorithm for mining association rules in large databases. In: *Proceedings of the 21th International Conference on Very Large Data Bases (VLDB)*, pp. 432–444.
- Shasha, D., Wang, J.T.L., Zhang, S., 2004. Unordered tree mining with applications to phylogeny. In: *Proceedings of 20th International Conference on Data Engineering*, pp. 708–719.
- Shen, R., Goonesekere, N., Guda, N., 2012. Mining functional subgraphs from cancer protein-protein interaction networks. *BMC Systems Biology* 6 (3).
- Chen, C.-H., Tsai, T.-H., Li, W.-H., 2015. Dynamic association rules for gene expression data analysis. *BMC Genomics* 16 (786).
- Srikant, R., Agrawal, R., 1996. Mining quantitative association rules in large relational tables. In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96*, pp. 1–12. ACM.
- Toivonen, H., 1996. Sampling large databases for association rules. In: *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB)*, pp. 134–145. Morgan: Kaufmann.
- Uno, T., Kiyomi, M., Arimura, H., 2004. Efficient mining algorithms for frequent/closed/maximal item sets. In: *Workshop of Frequent Item Set Mining Implementations, FIMI*. Brighton, United Kingdom.
- Wijsen, J., Meersman, R., 1998. On the complexity of mining quantitative association rules. *Data Mining and Knowledge Discovery* 2 (3). 263–281.
- Yang, G., 2004. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, pp. 344–353. ACM.
- Zaki, M., Parthasarathy, S., Ogihara, M., Li, W., 1997. New algorithms for fast discovery of association rules. In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 283–286. AAAI Press.
- Zaki, M.J., 2000. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 12 (3). 372–390.

Decision Trees and Random Forests

Michele Fratello, DP Control, Salerno, Italy

Roberto Tagliaferri, University of Salerno, Salerno, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Technological advancements of the last decades sparked an explosion in the amounts of data acquired in several scientific fields. In particular, high-throughput technologies have lead to an incredible growth of biological data. It is not uncommon, for instance, to measure simultaneously the levels of expression of thousands of genes thanks to microarray and sequencing technologies. As a natural consequence, several specialized databases accessible through the Internet have been populated. However, this data needs to be converted into knowledge to allow scientific progress. Machine Learning models allow to study the interactions among variables of interest, and have been employed in various scientific fields, including bioinformatics. Here, we review the theoretical and practical foundations of Decision Trees and Random Forests, as well as recent work from literature.

Before going on, we introduce the notation that we will use throughout this article. To keep the exposition clear, here, we focus on the context of classification problems. However, decision trees and random forests are flexible enough to be applied to regression problems, as well as to unsupervised learning. For an in depth discussion we will refer to the bibliographic references and to the additional resources useful for the interested reader at the end of this article.

Notation

The i -th observation of a dataset of samples is represented by the vector of its features, or variables, denoted by $\mathbf{x}_i$, and its dimensionality, i.e., the number of features, denoted by p .

The value of the j -th feature of the observation $\mathbf{x}_i$ is denoted by $x_{i,j}$, whereas we refer to the vector of all the values of a single feature as $\mathbf{x}_{:,j}$.

The outcome variable, or experimental condition or class label, corresponding to the observation $\mathbf{x}_i$ is represented by y_i which, in case of classification problems, is a discrete variable that can assume a finite number of different values, each corresponding to a different class.

A whole dataset is denoted by the $n \times p$ matrix X obtained by stacking all the feature vectors, where n is the number of samples of the dataset.

Similarly, the n dimensional vector of outcomes, denoted by $\mathbf{y}$, is obtained by concatenating the outcome values of each observation of the dataset.

Decision Trees

Decision trees are models based on the principle of *divide et impera*, i.e., the training dataset is recursively partitioned into non-overlapping rectangular regions (multi-dimensional boxes) and each derived region is associated with a decision rule responsible for the classification of that region.

This partitioning has the advantage of not limiting the relationship between the outcome variable (e.g., the class label of a sample) and the input variables (e.g., the features of a sample) of the dataset to a particular function family, like, for example, a linear relationship for Discriminant Analysis. These models are called non-parametric and their major strength is to automatically tune the complexity of the learned relationship directly from the data (Breiman *et al.*, 1984).

The resulting partitioning scheme can be represented by a binary tree as in Fig. 1. On the left, an artificial non-linearly separable dataset fitted with a decision tree is shown. On the right, the trained classifier is represented as a binary tree. Predictions can be performed by visiting the tree and performing the tests included inside the diamond shaped nodes. When a rectangular node is reached, the predicted label corresponds to the reported class.

A fitted decision tree can also be visualized as a set of intelligible decision rules. Each path from root to a leaf node can be translated into a chain of "IF-THEN" conditions that can be applied to observations to classify them. This makes decision trees particularly valuable in life sciences, where the inspection of decision rules can bring insights about the experimental question.

Training Algorithm

The general algorithm for training a decision tree involves two main phases: a growing phase and an optional pruning phase. In the growing phase, the decision tree is built in a top-down fashion. Starting with the whole dataset corresponding to the root node of the tree, a feature and a split value are identified according to a splitting criterion and the dataset is partitioned into two disjoint subsets depending on that feature value. This procedure is then recursively performed to each subset, until a stopping condition is reached.

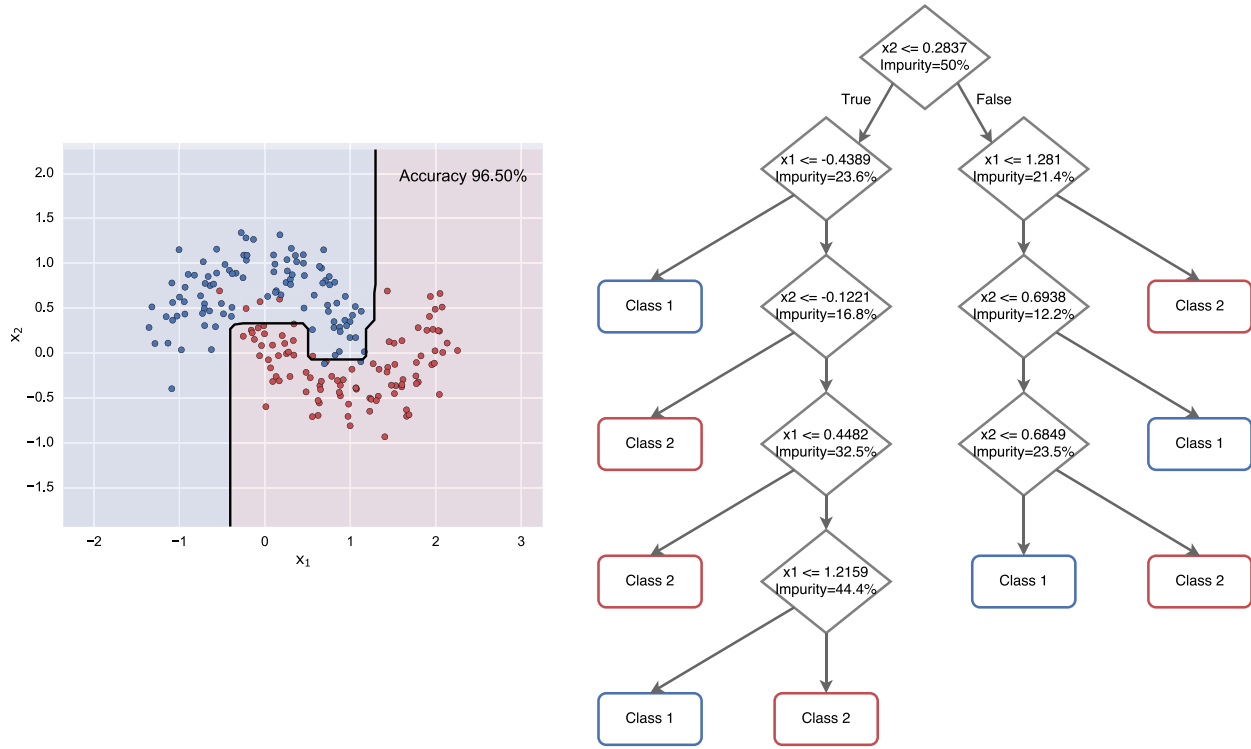


Fig. 1 Left: A two-classes synthetic dataset non-linearly separable. Right: The corresponding fitted classifier represented as a decision tree.

After growing, the tree may become too complex compared to the size of the training dataset. The optional pruning phase is used to avoid, or at least reduce, overfitting issues. In this case, each node is evaluated by a pruning criterion. When a node satisfies the pruning condition, its sub-tree is removed and the corresponding node becomes a leaf node. Pruning a sub-tree corresponds to merging together subsets of the training dataset. This procedure goes on until the tree is reduced to a single node and the best (in terms of accuracy) sub-tree built during the process is kept as the final tree. Lastly, each region of the dataset, represented by a leaf node, is associated with a decision rule, which usually corresponds to assign the most frequent class label in the corresponding region to the observations belonging to it (Breiman *et al.*, 1984).

Splitting criteria

During each step of the growing phase, the training procedure finds a feature $x_{i,j}$ and a split value s to divide the current set of observations X' into two smaller and more pure (or coherent) subsets $S_1 = \{x_i : |x_{i,j}| \leq s\}$ and $S_2 = \{x_i : |x_{i,j}| > s\}$.

Common purity criteria for classification problems are the Gini index and the Cross-entropy index:

$$\sum_{k=1}^K \hat{p}_k (1 - \hat{p}_k) \quad \text{Gini index}$$

$$\sum_{k=1}^K \hat{p}_k \log \hat{p}_k \quad \text{Cross entropy index}$$

Where $\hat{p}_k$ is the fraction of samples of each class k associated to a given subset S_i . Both indexes, equal 0 when $\hat{p}_k = 0$ or $\hat{p}_k = 1$ and reach the maximum value when $\hat{p}_k = \frac{1}{2}$ meaning that homogeneous subsets with the same value of the outcome variable (class label) are preferred.

Stopping criteria

Clearly, it is not possible to have more leaf nodes than samples in the training set. This configuration also corresponds to maximally pure leaf nodes. However, in this case, generalization performances to new samples will be poor. To avoid the explosion of the tree complexity, the growing phase must be stopped early in some nodes.

Common stopping criteria include limiting the minimum number of samples per node or the minimum level of impurity to split a node.

Pruning criterion

In Breiman *et al.* (1984) it was observed that imposing tight stopping criteria leads to underfit trees. On the other hand, training an unrestricted decision tree and then pruning it with the weakest link pruning procedure is shown to improve generalization performances.

The weakest link pruning procedure goes in the opposite direction of training: starting with the full tree, a sequence $T_0, T_1, \dots, T_k$ of smaller and smaller trees is built, where T_0 is the full tree and T_k is the tree with a single node. Each sub-tree T_{i+1} is built by collapsing each internal node of the sub-tree T_i into a leaf node, one at a time, and evaluating the error rate increase over the pruned leaves:

$$\alpha = \frac{\text{err}(T_{i+1}, \mathbf{X}') - \text{err}(T_i, \mathbf{X}')}{\text{leaves}(T_i) - \text{leaves}(T_{i+1})}$$

where $\text{err}(T, \mathbf{X}')$ is the error rate of tree T on the set of observations $\mathbf{X}'$, and $\text{leaves}(T)$ corresponds to the number of leaves for tree T . After the evaluation of all the collapsed sub-trees of T_i , the sub-tree that gets the lowest value of α is taken as T_{i+1} . The criterion α is also known as complexity parameter as it is used to determine the appropriate amount of complexity (the total number of leaves) to preserve from the original tree.

Finally, the optimally pruned tree is chosen from the sequence evaluating the performance of each sub-tree on a pruning dataset, not used during the training of T_0 . When data is not enough to be split into training and pruning datasets, cross-validation procedures can be used instead, although at a higher computational cost.

Random Forests

Despite the success and the desirable properties of decision trees, their discrete nature implies high variance in predictions, especially along the decision boundary. This means that small perturbations on the training dataset, may induce wild variations in predictions.

A Random Forest is an ensemble method based on bagging (bootstrap aggregating) (Breiman, 1996) where a large set of unstable (i.e., with high variance in predictions) but independent classifiers are aggregated using a majority vote to produce a more accurate classification with respect to each single model. The base predictor structure used in Random Forests is the decision tree, hence the name.

Ensemble learning methods allow constructing models that produce better predictions taking aggregations of predictions from a set of individual classifiers (Dietterich, 2000).

Two necessary assumptions for the ensemble model to be more accurate than its individual models are that (1) each of them must be more accurate than random guessing and (2) they need to produce as much independent errors as possible, i.e., there is some type of relation between the errors made by different classifiers in the ensemble.

In fact, an accuracy even slightly higher than random guess is sufficient to guarantee that the probability that the whole ensemble predicts the wrong class is reduced. The independence of errors is needed to ensure that possible wrong predictions are rejected by the rest of the classifiers which are expected to be higher in number, thereby increasing the overall accuracy.

More formally, consider an ensemble made of an odd number c of perfectly independent binary classifiers (i.e., there is not any statistical relation between the predictions of different classifiers), where each classifier has an error rate (i.e. probability of making a wrong prediction) of $p < \frac{1}{2}$. The ensemble then produces a wrong prediction when at least $\frac{c-1}{2} + 1$ individual classifiers are wrong, this happens with probability $(\frac{c-1}{2} + 1)^p + (\frac{c-1}{2} + 2)^p + \dots + (c)^p$, and this probability decreases very fast with an increasing number of classifiers.

As an example, let us consider an ensemble of 11 perfectly independent binary classifiers, each with a probability of error of $p = 0.499$, the probability that the whole ensemble is wrong drops to 0.03. Thus, even though the performance of each single classifier is near the random guess, ensembling even a small amount of independent classifiers will greatly improve the global performance.

Care must be taken since the above example is an ideal case and the computed probability is actually a lower bound. In practical applications, predictions within the ensemble are hardly perfectly independent. Nonetheless, good performances can be achieved, in practice, by maximizing the independency between the trained decision trees within the random forest.

Bagging and Random Subspace Selection

To enforce independency between predictors, the most common techniques are (Dietterich, 2000):

1. Train each predictor on a different subset of the original set of training samples.
2. Train each predictor on a different subset of features of the original training data.
3. Train different classes of predictors.

In Random Forests, independence among the predictors is ensured by training each of them on a bootstrapped training dataset, i.e., resampled with replacement from the original dataset and with the same number of samples. Also, to increment the

randomness needed to reduce dependencies between predictors in the ensemble, the search for the best split is performed each time on a different random subset of features of the dataset (Breiman, 2001).

The random selection of a subset of features has also the advantage of reducing the susceptibility of random forests to the curse of dimensionality, i.e., by considering only a few features at a time, the random forest is more robust to the noisy features in high-dimensional datasets, like gene expression datasets.

Out of Bag Error Estimate

When building the bootstrap dataset for each decision tree, each observation in the original dataset has a probability of $(1 - \frac{1}{n})^n$ of not appearing in a given bootstrapped dataset. In particular, this probability tends to $\frac{1}{e} \approx 0.3679$ as n increases, where n is the number of observations in the original dataset. This means that each decision tree is trained on a dataset that, on average, contains roughly two thirds of the observations in the original dataset, whereas the remaining are replicated observations.

Since each decision tree of the ensemble is trained on a bootstrapped dataset, then there is a set of approximately one third of samples belonging to the original dataset, different for each tree, and that is not used for training. Thus, it can be used to estimate the generalization performance of the tree. The generalization estimates of all trees of the ensemble are aggregated by averaging into the Out Of Bag (OOB) error estimate of the ensemble.

Through the OOB error, it is possible to estimate the generalization capabilities of the ensemble without the need of an hold-out test set (Breiman, 1996) that would be otherwise necessary as for any other machine learning model not based on a bagging mechanism.

Empirical studies showed that the OOB error is as accurate in predicting the generalization accuracy as using a hold-out test set given a sufficient number of estimators in the forest to make the OOB estimate stable (Breiman, 2001).

This characteristic becomes more relevant when the total amount of samples is not enough to be split in training, validation and test datasets, which is common in bioinformatics.

Variable Importance

The aim of each split when a decision tree or a random forest are trained is to decrease the impurity of each subset. The actual amount of impurity reduction can be seen as an index to identify which features are the most relevant in separating the dataset into homogeneous groups with respect to the class label. The impurity decrease is averaged across each tree of the forest during training.

High scores correspond to high impurity reduction, i.e., the corresponding features are more relevant to classification. Usually, these scores are normalized such that the sum of all importance values equals to 1.

Use-Cases

The following are two practical examples of how to use a decision tree or a random forest with real world datasets. They focus on the basics of data analysis in the common software tools R and Python. The full commented source code is available online (see Relevant Websites section), together with the instructions to setup the computational environments.

Predicting Heart Failures with Decision Trees in R

Here, we show how to perform an analysis using a Decision Tree classifier with the R environment. The example dataset is made of 270 patients, where each sample has 13 clinical features (details about the dataset can be found at the dataset webpage (see Relevant Websites section)). The objective of the analysis is to derive a model to detect heart failures, as well as determining which features are the most relevant by examining a small set of decision rules.

First, we load the dataset and split it into a training and a test sets with a 70%/30% ratio

```
data=read.table("heart.txt", header=TRUE, dec=".")
split<-sample(c(1, 2), size =nrow(data),
              replace=TRUE, prob=c(.7, .3))
data.train=data[split==1, ]
data.test=data[split==2, ]
```

Then, we load the R decision tree library and grow an unconstrained decision tree. Specifically, we allow the leaf nodes to contain at least one sample, and internal nodes to have at least two samples in order to be split; furthermore, we choose the Gini Index as splitting criterion. Finally, we don't want to prematurely prune the tree, so we disable this option by setting the parameter `cp` to 0.

```
library(rpart)
settings=rpart.control(minsplit=2,minbucket=1,cp=0)
tree=rpart(num~ .,data=data.train,method="class",
           parms=list(split="gini"),control=settings)
```

We expect the grown tree to be exceedingly large and to overfit the training data, with reduced generalization capabilities on the test set. We evaluate the trained model on both the training and test sets by printing the confusion matrices as well as the total accuracy

```
pred.train=predict(tree, data.train, type="class")
pred.test=predict(tree, data.test, type="class")

#The confusion matrix allows to visualize the type of errors made by the
model
conf.train=table(data.train$num, pred.train)
conf.test=table(data.test$num, pred.test)

#Finally, we evaluate the accuracy
accuracy.train=sum(diag(conf.train))/sum(conf.train)
accuracy.test=sum(diag(conf.test))/sum(conf.test)

cat("Predictions of the grown Classification Tree for the training set:\n")
print(conf.train)
cat(c("Total accuracy: ", round(accuracy.train, 4)*100, "%"))
>Output:
Predictions of the grown Classification Tree for the training set:
absence presence
absence      104         0
presence       0        85
Total accuracy: 100 %

cat("Predictions of the grown Classification Tree for the test set:\n")
print(conf.test)
cat(c("Total accuracy: ", round(accuracy.test, 4) * 100, "%"))
>Output:
Predictions of the grown Classification Tree for the test set:
absence presence
absence       29        17
presence       8        27
Total accuracy: 69.14 %
```

As expected, the performance on the training set is of 100%, whereas the generalization accuracy to samples never seen before is much lower. To reduce the overfitting and improve the generalization capabilities, we will prune the tree, at the expense of a deterioration of the performance on the training set.

```
plotcp(tree)
printcp(tree)
>Output:
      CP nsplit  relerror  xerror   xstd
1  0.4588235     0  1.000000  1.00000  0.080459
2  0.0705882     1  0.541176  0.63529  0.073066
3  0.0470588     3  0.400000  0.51765  0.068353
4  0.0352941     5  0.305882  0.50588  0.067805
5  0.0235294     6  0.270588  0.47059  0.066065
6  0.0176471     7  0.247059  0.43529  0.064176
7  0.0117647     9  0.211765  0.45882  0.065453
8  0.0098039    18  0.105882  0.50588  0.067805
9  0.0058824    25  0.035294  0.52941  0.068887
10 0.0000000    31  0.000000  0.52941  0.068887

#Then we limit the tree at the complexity parameter corresponding to the
#lowest cross-validation error
tree.pruned=prune(tree, cp=0.0176471)
```

The functions `plotcp` and `printcp` show respectively a graph (see Fig. 2) and a table of the cross-validated complexity parameter value (alpha) for each pruned sub-tree. The best pruned tree is selected as the one that achieves the lowest cross-validated error (the y-axis of the graph or the `xerror` column in the table above).

Now we evaluate the classification performance of the pruned tree

```
pred.train=predict(tree.pruned, data.train, type="class")
pred.test=predict(tree.pruned, data.test, type="class")

conf.train=table(data.train$num, pred.train)
conf.test=table(data.test$num, pred.test)

accuracy.train=sum(diag(conf.train))/sum(conf.train)
accuracy.test=sum(diag(conf.test))/sum(conf.test)

cat("Predictions of the pruned Classification Tree for the training
set:\n")
print(conf.train)
cat(c("Total accuracy: ", round(accuracy.train, 4)*100, "%"))
>Output:
Predictions of the pruned Classification Tree for the training set:
      absence presence
absence     96       8
presence    13      72
Total accuracy:  88.89 %

cat("Predictions of the pruned Classification Tree for the test set:\n")
print(conf.test)
cat(c("Total accuracy: ", round(accuracy.test, 4)*100, "%"))
>Output:
Predictions of the pruned Classification Tree for the test set:
      absence presence
absence     35      11
presence     9      26
Total accuracy:  75.31 %
```

By pruning the tree, we obtained two advantages. First, the model is smaller, hence demanding less computational power as well as memory. Second, the generalization capabilities of the model to new subjects have improved. However, we lost some accuracy on the training set.

As a last step, we want to distill the pruned model into a set of intelligible decision rules to gain insights about the most relevant features for the classification

```
#Rule Extraction
library(rattle)
asRules(tree.pruned, compact=TRUE)
>Output:
R 13 [ 6%,1.00] cp=D ca=A oldpeak>=15.5
R  7 [23%,0.95] cp=D ca=B,C,D
R 11 [ 6%,0.83] cp=A,B,C ca=B,C,D slope=2
R 49 [ 2%,0.75] cp=D ca=A oldpeak< 15.5 age< 58.5 age< 41
R 25 [ 5%,0.67] cp=D ca=A oldpeak< 15.5 age>=58.5
R 10 [10%,0.22] cp=A,B,C ca=B,C,D slope=1,3
R 48 [12%,0.13] cp=D ca=A oldpeak< 15.5 age< 58.5 age>=41
R  4 [36%,0.09] cp=A,B,C ca=A
```

For each row, the first number is the index of the corresponding leaf node in the tree. The percentage in the square bracket is the sample coverage of the rule, i.e., the number of samples to which the rule applies, whereas the second number in the square brackets is the probability of the presence of a heart failure.

The obtained decision rules highlight that of the 13 features, only 5 are actually relevant, specifically: the type of chest pain (cp), the number of vessels colored by fluoroscopy (ca), age and two electrocardiography related features (oldpeak and slope).

Predict Relapse of Primary Breast Cancer With Python

In this second example, we show how to build a random forest in Python to discriminate two subtypes of breast cancer. For the purpose, we use a subset of samples from the TCGA Breast Invasive Carcinoma dataset (see Relevant Websites section) already preprocessed as in [Serra et al. \(2015\)](#). The original dataset is heavily unbalanced, here, for the sake of exposition we focus only on the two most common classes, namely Luminal A and Luminal B.

The reduced dataset is made of 114 subjects and for each subject the mRNA expression of 4000 genes are measured by NGS technology. We start by loading the necessary components and the dataset files

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.feature_selection import VarianceThreshold
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix

from matplotlib import pyplot as plt

#init PRNG for reproducibility
np.random.seed(123456)

#Load gene expression data
gene_data = pd.read_csv("breast_subtyping.txt", sep='\t')

#Load clinical data
clin_data = pd.read_csv("patient_classes.txt", sep='\t', comment='#')

#Since the dataset is heavily unbalanced, we restrict our example
#to the two most common classes
to_remove = clin_data[(clin_data.x != "LumA") & (clin_data.x != "LumB")].index

Then, we remove any feature which contain NaNs and create the data matrix
and the labels vector.

#Create feature matrix where each row corresponds to a subject
#and each column corresponds to a gene
X = gene_data.T.drop(to_remove).sort_index()
X = X.dropna(axis=1)
print("Input Dataset size is {}".format(X.shape))
#As labels we want learn to predict, we use the status of relapse of each
subject
le = LabelEncoder()
y = le.fit_transform(clin_data.drop(to_remove).x.sort_index())
print("Output label size is {}".format(y.shape))
```

The data is then split into a training and a test sets using the ratio 70%/30%

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7,
stratify=y)
oob_score = True,
n_jobs=1,
)

oob_score = rf.fit(X_train, y_train).oob_score_
oob_scores.append(oob_score)

plt.plot(n_trees, oob_scores, "b")
plt.ylim(.5, 1)
plt.legend(["OOB Accuracy Estimate"])
```

To define the random forest model, we need to define a set of parameters, including the number of trees in the ensemble, the stopping criteria for each tree, the number of features to randomly select and the splitting criterion. These parameters are usually estimated, however, here for the sake of exposition, we will keep them fixed to reasonable values that are known to work well in practice.

We will grow an increasing number of unconstrained trees, like in the previous use case, expecting that overfit trees will produce more independent predictions. Also the size of the random subset of features to be selected is fixed to $\sqrt{p}$, where p is the total number of features, as suggested in Breiman (2001).

However, the number of trees to grow still needs to be estimated. As a rule of thumb, the more trees we grow, the better generalization performance we get until a point where the performance stop to improve. Using more trees than necessary results in a waste of resources.

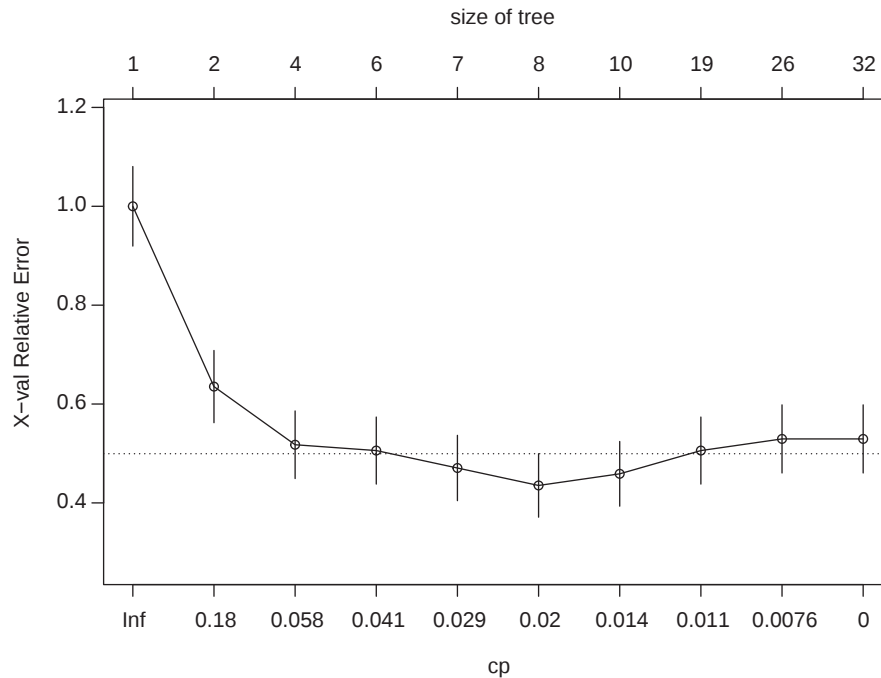


Fig. 2 Plot of the cross-validated relative error with respect to the complexity parameter. The best model is that corresponding to the lowest cross-validated error.

In the left plot of [Fig. 3](#), we show how the OOB accuracy prediction varies with the number of trained trees. To choose the optimal number of estimators, we inspect the OOB plot to look for the point where the OOB estimate stabilizes. Here, the estimates become more stable after 6001 trees are trained, and therefore we can fix the number of trees to 6001 and compare the OOB performances with the accuracy predictions in the test set.

```
best_rf=RandomForestClassifier(
    n_estimators=6001,
    criterion="entropy",
    min_samples_leaf=1,
    min_samples_split=2,
    max_features="sqrt",
    min_impurity_split=0,
    oob_score=True,
    n_jobs=1,
)
best_rf.fit(X_train,y_train)
print("Test set accuracy of the best model
{:.02f}%".format(best_rf.score(X_test,y_test)*100))
>Output:
Test set accuracy of the best model 80%
```

Finally, we show how to rank the most relevant features with respect to the impurity criterion across all the trees of the ensemble

```
top=10
importances=best_rf.feature_importances_
idx=importances.argsort()[::-1]
pos=np.arange(top)[::-1]+.5
plt.barh(pos,importances[idx[:top]], align="center")
plt.yticks(pos,X.columns[idx[:top]])
```

In the right plot of [Fig. 3](#) we show the 10 most relevant genes in the discrimination of the two subtypes of breast cancer and their relevance value, computed as the average decrease of the impurity criterion across the forest.

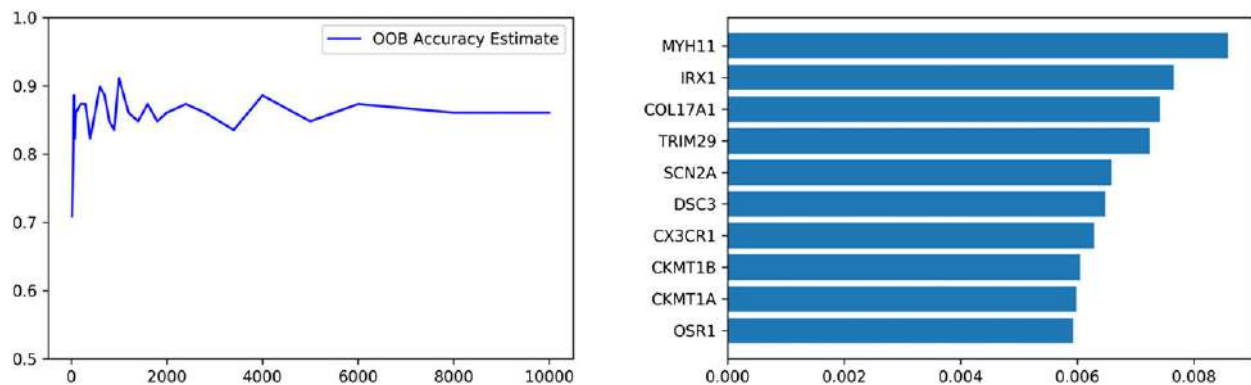


Fig. 3 Left: Plot of the out-of-bag accuracy estimate with respect to the number of trees in the forest. Right: The ten most relevant features for the classification of breast cancer subtypes.

Literature Applications

The ease of use, the flexibility and quality of results make Decision Trees and Random Forests two of the most used tools in the field of bioinformatics. Here, we give a few examples from recent literature of applications of the exposed models.

A major field of application is cancer classification, where each observation in the dataset is a subject and the objective is to predict the presence or absence of a cancer disease. Both decision trees and random forests have been applied to the classification of different types of cancer (Chen *et al.*, 2015; Podolsky *et al.*, 2016; Tsai *et al.*, 2016; Venkatesan and Velmurugan, 2015).

The most common features employed are the gene expression levels, measured by microarrays. The number of features that can be simultaneously measured with this technology varies between few hundreds and tens of thousands. Classification in this setting is ill-posed, given that noisy and/or correlated features can harm the final classification performances. To tackle this problem, classifiers are coupled with mechanisms of gene selection, based on variable importance, where the features are ranked based on how much they influence the prediction outcome (Kursa, 2014; Nguyen *et al.*, 2013). A second approach to gene selection is based on meta-heuristic search algorithms like Particle Swarm Optimization among others (Chen *et al.*, 2014), which explores the space of gene subsets searching for the smallest one with the best discriminative power.

Other applications include genome-wide association studies (Botta *et al.*, 2014; Li *et al.*, 2016; Nguyen *et al.*, 2015; Sapin *et al.*, 2015; Stephan *et al.*, 2015) in which genetic variations, usually SNPs, or single nucleotide polymorphisms, correspond to the features and a set of phenotypes of interest are the response variables, like particular traits or diseases. In this case not only it is important to determine a small set of relevant SNPs, but also to highlight which interactions among them trigger the phenotype. Here, the advantage of tree-based models is in the possibility of modeling non-linear interactions without the need to assume a predefined functional form. Furthermore, interactions are taken into account based on statistical or permutation based reasoning or, again, with the use of meta-heuristics like ant colony optimization (Sapin *et al.*, 2015).

Discussion

As every other machine learning model, Decision Trees and Random Forests have their advantages and disadvantages, which make them more or less suitable to tackle a problem. Nonetheless, they have become a major tool in the bioinformatics community, also thanks to a number of software implementations. Decision Trees' strengths are the relatively easy theoretical foundations, the capability to inspect the learned model and the reduced computational requirements. However, the high flexibility of Decision Trees is also a major drawback, since there is a high chance of overfitting.

Random Forests, on the other hand, have almost always better performances at the cost of higher memory requirements and losing model intelligibility, even though it is still possible to evaluate the feature relevance to the classification. Moreover, the lack of a predefined functional form of the interactions between the features and the outcome variable allows both models to automatically tune the complexity of the models on the data at hand.

Conclusions

We introduced the basic concepts of Decision Trees and Random Forests, with particular attention to the bioinformatics community, and we reviewed recent applications of these models while showing basic practical applications. We hope that this short review can be helpful to bioinformatics practitioners.

See also: Data Mining in Bioinformatics. The Challenge of Privacy in the Cloud

References

- Botta, V., Louppe, G., Geurts, P., Wehenkel, L., McGovern, D., 2014. Exploiting SNP correlations within random forest for genome-wide association studies. PLOS ONE 9 (4), e93379. Available at: <https://doi.org/10.1371/journal.pone.0093379>.
- Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140. Available at: <https://doi.org/10.1007/BF00058655>.
- Breiman, L., 2001. Random forests. Machine Learning 45 (1), 5–32. Available at: <https://doi.org/10.1023/A:1010933404324>.
- Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and regression trees. The Wadsworth Statisticsprobability Series, vol. 19. Available at: <https://doi.org/10.1371/journal.pone.0015807>.
- Chen, H., Lin, Z., Wu, H., *et al.*, 2015. Diagnosis of colorectal cancer by near-infrared optical fiber spectroscopy and random forest. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy 135, 185–191. Available at: <https://doi.org/10.1016/j.saa.2014.07.005>.
- Chen, K.-H., Wang, K.-J., Tsai, M.-L., *et al.*, 2014. Gene selection for cancer identification: A decision tree model empowered by particle swarm optimization algorithm. BMC Bioinformatics 15 (1), 49. Available at: <https://doi.org/10.1186/1471-2105-15-49>.
- Dietterich, T.G., 2000. Ensemble methods in machine learning. Multiple Classifier Systems, vol. 1857. Berlin, Heidelberg: Springer, pp. 1–15. Available at: https://doi.org/10.1007/3-540-45014-9_1.
- Kursa, M.B., 2014. Robustness of random forest-based gene selection methods. BMC Bioinformatics 15 (1), 8. Available at: <https://doi.org/10.1186/1471-2105-15-8>.
- Li, J., Malley, J.D., Andrew, A.S., Karagas, M.R., Moore, J.H., 2016. Detecting gene-gene interactions using a permutation-based random forest method. BioData Mining 9 (1), 14. Available at: <https://doi.org/10.1186/s13040-016-0093-5>.
- Nguyen, C., Wang, Y., Nguyen, H.N., 2013. Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. Journal of Biomedical Science and Engineering 6 (5), 551–560. Available at: <https://doi.org/10.4236/jbise.2013.65070>.
- Nguyen, T.-T., Huang, J., Wu, Q., Nguyen, T., Li, M., 2015. Genome-wide association data classification and SNPs selection using two-stage quality-based random forests. BMC Genomics 16 (Suppl. 2), S5. Available at: <https://doi.org/10.1186/1471-2164-16-S2-S5>.
- Podolsky, M.D., Barchuk, A.A., Kuznetsov, V.I., *et al.*, 2016. Evaluation of machine learning algorithm utilization for lung cancer classification based on gene expression levels. Asian Pacific Journal of Cancer Prevention: APJCP 17 (2), 835–838. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/26925688>.
- Sapin, E., Frayling, T., Keedwell, E., 2015. Ant colony optimisation of decision tree and contingency table models for the discovery of gene–gene interactions. IET Systems Biology 9 (6), 218–225. Available at: <https://doi.org/10.1049/iet-syb.2015.0017>.
- Serra, A., Fratello, M., Fortino, V., *et al.*, 2015. MVDA: A multi-view genomic data integration methodology. BMC Bioinformatics 16 (1), Available at: <https://doi.org/10.1186/s12859-015-0680-3>.
- Stephan, J., Stegle, O., Beyer, A., Büchse, A., Calus, M.P., 2015. A random forest approach to capture genetic effects in the presence of population structure. Nature Communications 6, 7432. Available at: <https://doi.org/10.1038/ncomms8432>.
- Tsai, M.-H., Wang, H.-C., Lee, G.-W., Lin, Y.-C., Chiu, S.-H., 2016. A decision tree based classifier to analyze human ovarian cancer cDNA microarray datasets. Journal of Medical Systems 40 (1), 21. Available at: <https://doi.org/10.1007/s10916-015-0361-9>.
- Venkatesan, E., Velmurugan, T., 2015. Performance analysis of decision tree algorithms for breast cancer classification. Indian Journal of Science and Technology 8 (29), Available at: <https://doi.org/10.17485/IJST/2015/V8I29/84646>.

Relevant Websites

<https://github.com/mfratello/ebcb-dtrf>

GitHub Inc.

<https://portal.gdc.cancer.gov/projects/TCGA-BRCA>

National Cancer Institute.

[https://archive.ics.uci.edu/ml/datasets/Heart + Disease](https://archive.ics.uci.edu/ml/datasets/Heart+Disease)

UCI, Machine Learning Repository.

Data Mining: Classification and Prediction

Alfonso Urso, Antonino Fiannaca, Massimo La Rosa, and Valentina Ravi, Via Ugo La Malfa, Palermo, Italy
Riccardo Rizzo, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Basic Concept of Classification and Prediction

Classification is the most common task in machine learning. It works on input data where each item is tagged with a predefined set of categorical labels or classes (Kesavaraj and Sukumaran, 2013). From such training records, the classifier learns a *model* able to distinguish classes or concepts for future predictions of discrete variables.

Using examples labelled with true function values (observation, measurements, etc.), a numerical predictor, or merely a predictor, learns to construct continuous-valued functions as model that predicts unknown or missing numeric values (Han et al., 2012).

Many applications of classification task solve several practical problems, for example, credit risk prediction, based on the knowledge acquired by collecting data from many loan applicants. The output from such a classifier will be one of two possible categorical labels “safe” or “risky”, asserting if a new applicant for a loan is reliable or not.

Similarly, regression models can be used in several fields of interest, for example, in marketing management to predict how much a given customer will spend during a sale. Unlike classification, numeric prediction models continuous-valued function f by mapping the input vector of attributes X to gives ordered continuous values y' as output: $y' = f(X)$.

Model Construction

Building a classifier or a predictor needs a training phase, in which a set of input labelled data, known as the *training set* is shown to the machine. Each item in training set is described by some *features* or *attribute* (qualitative or quantitative) and by a class or a numerical label. In this step, the candidate algorithms learn predetermined data classes or concepts: this is the operation mode of supervised learning. The *validation set* allows selecting the best model among possible candidates, comparing their performances over data never seen before. Finally, when the model was chosen, it needs to assess the machine's ability in classification task on a new data set, so called *test set*. A good classifier or predictor is a machine that inferred the “general” model underlying training data and can apply it to absolve his task on new unseen data.

Role of features

The term *features* holds the meaning of learning, representing the ensemble of characteristics useful to bring out the model from input data. The right model gets the best classification or prediction performances, for this reason, the skill of features extraction is considered a basic part of the classification/prediction task.

Classifier Versus Predictor: Learning, Testing, Accuracy, Overfit

Classification and prediction represent the most used supervised techniques of machine learning (Han et al., 2012). Classifiers and predictors develop their skills starting from training labelled data, discrete or numerical records, and produce categorical labels (classes) or numeric predictions (function values) respectively. The base of operation for both machines is the “model” that must connect the class (discrete or numerical) to the features.

Learning

The first step to constructing a classifier is learning the model from data. The classification model originates from training data, which must be suitably prepared. Data preparation involves (i) cleaning, (ii) relevance analysis and (iii) data transformation:

- (i) In the cleaning step, data are pre-processed to reduce noise and to handle missing values.
- (ii) Relevance analysis consists of correlation analysis and attribute selection: these procedures reduce redundant and irrelevant features, improving classification efficiency and scalability.
- (iii) Then the information contained in training records have to be generalized into concepts, with data dimensionality reduction.

In training phase, the machine is fed with pre-processed data, which is a list of instances, each of them described by some attributes and an assigned qualitative (for a classifier) or numerical (for predictor) label for each. This label represents the “true” class or the function values. Symbolically, the i -th instance is presented to the algorithm by the pair (X_i, y_i) , where $X = (x_1, \dots, x_n)$ is the attribute vector, which is associated to a predefined label y_i . The algorithm increases its knowledge using the information from the training records. The larger is the training set, the better is the classifier/predictor. In particular, a numerical predictor in training phase constructs a function that minimizes the difference between the predicted and true function values.

Model selection depends in comparing different models learned during the training by using validation data set.

Testing and Accuracy

When the training phase finish and the validation procedure selects the best among possible models, a test evaluates the learning degree of the machine. For this purpose, the algorithm will examine a set of data in the same form (X_i, y_i) of the training set. Usually, test sample is achieved randomly selecting 20%–30% of the complete labelled data set.

A comparison between the class or numerical label of each instance in the test set and the corresponding output is used to test the classifier. In particular, the percentage of cases correctly classified in the testing data set gets the accuracy of the classifier, as shown in the formula (1):

$$Accuracy = \frac{p}{t} \quad (1)$$

where p is the number of right classifications and t is the total number of test cases.

For a predictor the accuracy is provided by measuring the “distance” between the actual value y_i and the predicted value y'_i corresponding to each vector of input attribute X_i . *Loss functions* provide these kind of error measures. The following expressions (2,3) shows the most common loss functions:

$$Absolute\ error: |y_i - y'_i| \quad (2)$$

$$Squared\ error: (y_i - y'_i)^2 \quad (3)$$

Starting from these expression, the average loss over the test set is obtained by the error rates in expressions (4) and (5):

$$Mean\ absolute\ error: \frac{\sum_{i=1}^d |y_i - y'_i|}{d} \quad (4)$$

$$Mean\ squared\ error: \frac{\sum_{i=1}^d (y_i - y'_i)^2}{d} \quad (5)$$

where d is the total number of test instances.

Evaluation of classification and prediction methods can also consider other criteria like:

- speed: time to construct the model; time to use the model;
- robustness: noise and missing values;
- scalability: efficiency in disk-resident databases;
- interpretability: understanding and insight provided by the model.

Generalization and Overfitting

The construction of classification model is based on a suitable choice of features, in type and number. Performances of the classifier can be enhanced by raising the number of features and producing sophisticated *decision boundary* among data. This choice can lead to excellent performances on the test set but can result in poor performances on new patterns. This declining performance is a symptom of the problem of *overfitting*. It means that a complicated model with a complex decision boundary can give poor *generalization* capability.

To avoid this effect, it is desirable looking for a general model. Indeed, the robustness of the algorithm depends on the degree of generalization, namely the capacity of working with the crucial features of data and choosing general rules for the model.

Further Considerations

The simpler classification problem is the binary classification. The binary or two-class classification has several applications, for example, in problems regarding credit/loan approval or medical diagnosis. Moreover, classification techniques provides solution also for problems involving more than two classes, dealing with *multi-class classification* (see also the section about multi-class classification in SVM section) (Li et al., 2004; Zhou et al., 2017).

It is possible to forecast numeric values with regression. Three types of regression analysis can be conduct:

- (i) Linear and multiple regression.
- (ii) Non-linear regression.
- (iii) Other methods, such as generalized linear model (GLM), Poisson regression, regression trees.

A discussion of such methods will be subject to the following paragraphs and the next article.

Classification by Decision Tree Induction: ID3 (Iterative Dichotomiser), C4.5, CART (Classifier and Regression Trees)

In a classification task, the objects of the universe are described by a set of attributes and their values. To distinguish an object from others, the classifier has to absolve the *induction task* (Quinlan, 1986). The induction task consists in developing a classification rule able to identify the class of any object of the universe from the value of its attributes that represents some important characteristics or *features* of object.

A decision tree classifier is a simple and widely used classification technique. As other classification methods, it builds a classification model learning from a bunch of input data. By analogy with a botanical tree, the structure of the decision tree consists of a flowchart with a root, branches, internal nodes and leaves. Each structural element of a tree identifies a step in a splitting data process, whereby the tree algorithm separates input training items by testing their different characteristics, i.e., different values of *some* their attributes.

In particular, the classification process of an object starts from the *root* that is the feature always tested for each instance. The different values of root attribute lead to different possible paths on the tree, addressing the algorithm to a specific *branch*. The decision process continues in the next *internal node*, where a new attribute test run over another instance attribute until the reaching of a *leaf node*, representing the class that owns the instance considered.

Decision tree classifiers are employed extensively in several fields of interest, such as medical diagnosis, weather prediction, fraud detection, credit approval, target marketing, customer segmentation (Lavanya and Rani, 2011). Their hallmarks are: (i) providing intelligible classification rules, (ii) ease to interpret, (iii) speed of construction, (iv) high accuracy. Often expert systems use decision tree, because they offer results comparable to expert human skills.

Build a Tree: Divide-and-Conquer Algorithm

The build or growth phase of tree begins from the choice of the root. Numerical computations based on information theory (see Sections Information gain and Gini index) establish which attribute ought to be the root, or an internal node. Its values will be the branches that separate the training data in subsets (attribute test). In the next step, for each branched subset the algorithm iterates the numerical computation on remaining attributes in order to find the following node and further divides data. Tree expansion process stops when a leaf node arrives, namely when each item of the analyzed subset gives the same overcome of the attribute test: this value represents the *class label* (Lavanya and Rani, 2011). In this ideal case, leaf node is *pure*, including elements with the same category label. However, usually such a leaf node is not obvious, because it can contain a mixture of elements with different label class. In this more recurrently case, the algorithm uses a different stopping criterion, accepting such as imperfect decision in favor of a simpler, more generic tree structure. As an example, in Fig. 1 is shown a training set and the related tree, proposed by Quinlan (1986). Here, the classifier suggests if “Saturday morning” is suitable or not for some activity, respectively with the leaf node Yes or No, starting from its weather attributes.

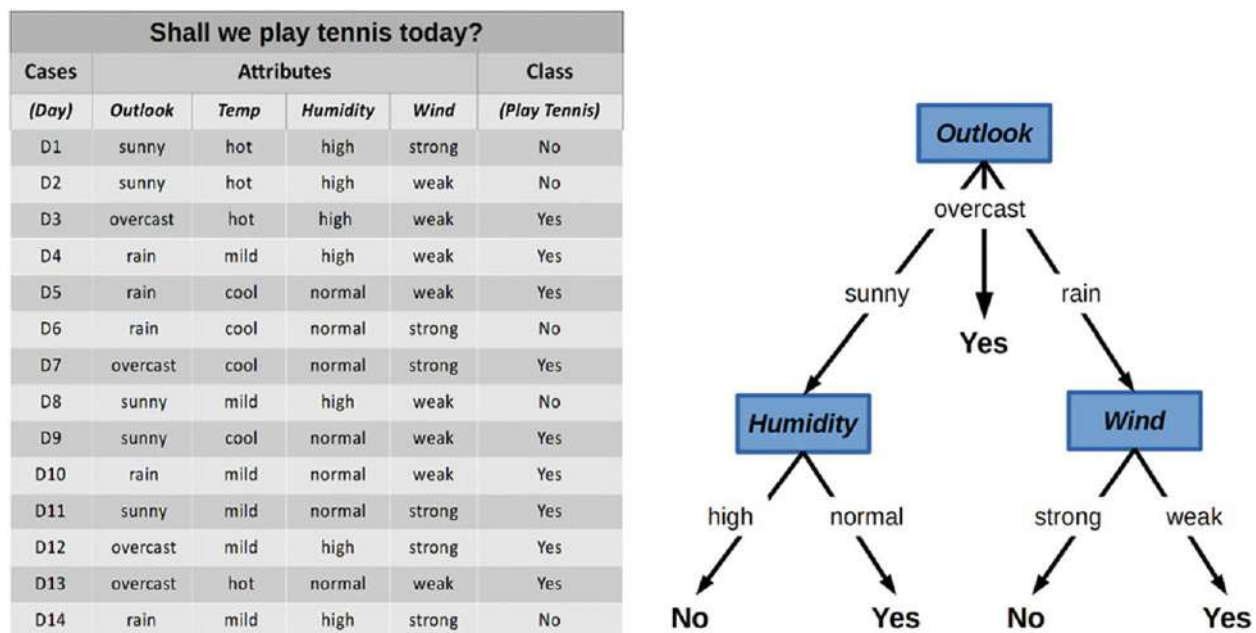


Fig. 1 Example of training set and related tree. The example is referred to the question if “Saturday morning” is suitable or not for some activity. Reproduced from Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1, 81–106.

An adequate choice of features allows build a decision tree that provides a correct classification for each object in the training data set. Nevertheless, the scope of induction task is implementing a classifier able to label correctly other unseen objects as well. Therefore, the decision tree must catch the underlying connection between an object's class and values of its attributes.

It is possible to build more than one tree starting from a training set; in general, it is better to choose the simplest tree in order to avoid overfitting problems and guarantee a proper classification of new objects. Indeed, it would be expected that the simpler tree works better than others tree trained on the same data set, because it more likely that correctly classifies more objects unseen (Quinlan, 1986). For this reason, in the most of cases, the algorithm uses a nontrivial stopping criterion in growing procedure, considering a mixture of elements with different label class as a leaf node and accepting such as imperfect decision in favor of a simpler, more generic tree structure. Furthermore, *pruning* operation generalizes the tree structure where branches are redundant, therefore removing noise and outliers.

The frequently used decision tree algorithm are Iterative Dichotomiser 3 (ID3), C4.5 and CART.

ID3

ID3 is a decision tree algorithm with an iterative structure based on Hunt's Concept Learning Systems (Hunt, 1962; Hunt et al., 1966). During the learning process, only a subset of training data is selected randomly to form a first decision tree. If this tree correctly classifies also all the other objects in the training set, then it represents the correct tree and the research process finish. Otherwise, a new subset or *window* of training records is chosen to form a new tree, including a selection of the incorrectly classified objects. This iterative process continues until the decision algorithm is able to classify correctly the entire training data set. The main advantage of ID3 strategy allows identifying the correct decision tree with few iterations even on training set containing thousands of items.

C4.5 algorithm

An improvement of ID3 is C4.5 algorithm, also developed by J. Ross Quinlan in 1993. C4.5 works with multiway splits in attribute test (Chen et al., 2015). C4.5 introduces some advantages with respect to ID3, working with both categorical and numerical (or continuous) attributes, and handling missing attribute values. Moreover, it employs information-based criteria as attribute selection measure. Furthermore, it uses pruning phase when the tree building is completed. Pruning allows handling the overfitting problem, wherein the decision tree algorithm may incur.

Classification and regression tree

Classifier and regression trees (CART) is a decision tree technique (Breiman et al., 1984) can perform both tasks of classification and regression, depending on the input data nature, categorical or numerical respectively. In particular, leaves of regression tree predict real numbers (Rokach and Maimon, 2005).

CART splits data by binary attribute tests, driven by pre-set criteria that establish when the growing process stops (Wu et al., 2008). The tree represents a series of rules opportunely fixed, learned by training data set, in order to predict a likely class label for a new examined case (Lawrence and Wright, 2001). Pruning is used to address the overfitting problem, which is recurrent also for CART.

Attribute Selection Measures

Information Gain and Gini Index are two split criteria widely used in the decision tree literature. Split criteria perform attribute selection for each node, when growing the tree.

Choosing correctly attributes is extremely important for proper classification, since from this depends the data split and the consequent performance of the tree. It is not obvious to determine which criterion is the best for a given data set (Raileanu and Stoffel, 2004).

Information gain

The decision tree algorithm uses a set of measures of entropy or impurities to evaluate the best choice for the node attribute. Indeed, a good attribute splits the data by grouping the greatest number of cases that belong to the same class, so that a successor node is as pure as possible. For this reason, it is indicative a measure of "order", defined as the number of items belonging to the same class; obviously, the more items belong to the same class, greater will be the order degree. Entropy is a measure of disorder, and represents the amount of information. Entropy is null for a pure node (leaf) and becomes maximum when all the classes at node N are equally likely. Therefore, given a set S of examples and n classes, the *entropy* is expressed by Eq. (6)

$$E(S) = \sum_{i=1}^n -p_i \log_2 p_i \quad (6)$$

where p_i is the portion of examples in i -th class (entropy of set S is constant for all attribute).

To compute the quality of the split on an entire attribute A, the weighted average is calculated over all sets resulting from the split on all values of A: $I(S,A)$ is the *average entropy for attribute A* and is defined as in Eq. (7):

$$I(S,A) = \sum_j \frac{|S_j|}{S} E(S_j) \quad (7)$$

where S_j represents the j -th subset of S depending on the j -th value of the attribute A . The *information gain* for attribute A is the information gained by branching on A is shown in Eq. (8):

$$\text{Gain}(S, A) = E(S) - I(S, A) \quad (8)$$

A good rule to tree growing consists of choosing that attribute for which the gain of information is maximum, or equivalently, that attribute with minimum average entropy (Han et al., 2012).

Information gain is used as attribute selection measure in ID3 e C4.5 algorithms.

Gini index

Gini Index is an inequality measure, employed in several scientific fields. In decision tree learning, Gini index is another method to assess purity/impurity of a node. Mathematical expression of Gini impurity of a set S_m of records at a given node m is the following Eq. (9):

$$i(S_m) = \text{Gini}(S_m) = 1 - \sum_{j=1}^n p_j^2 \quad (9)$$

where p_j is the portion of records belonging to j -th class, over a total of n classes. For Gini index, each attribute determines a binary data split. Gini index measures impurities of each possible pairs of partitions S_{m1} and S_{m2} for a given attribute A , as shown in formula (10):

$$\text{Gini}_A(S_m) = \frac{S_{m1}}{S_m} \text{Gini}(S_{m1}) + \frac{S_{m2}}{S_m} \text{Gini}(S_{m2}) \quad (10)$$

The attribute with the partition having minimum Gini index is chosen to continue the tree building process (Han et al., 2012). Gini index is usable in problem with two or more categories and is employed as attribute selection measure in CART algorithm.

Rule-Based Classification

Rule models are the second major techniques used in machine learning after the decision tree methods. Trees offer a first, although rigid example of rules as seen in branching process, since each branch dictates univocal way to browse the tree. Instead, in a rule model the algorithm may infer further information from the possible overlapping among several rules. As well as the decision trees, the Rule-Based Classifiers are highly expressive, easy to generate and to interpret. Their performances are comparable to decision trees. Moreover, rules methods can classify rapidly new instances and can easily handle missing values and numeric attributes (Tan et al., 2005). Rule-based methods form an integral part of expert systems in artificial intelligence.

If-Then Rules

A rule-based classifier uses a collection of rules of the type, "If.... Then..." to absolve the classification task. The expression following "If" states a condition or predicate (antecedent rule or left-hand side, LHS) that is a conjunction of attributes value. The class is defined after the keyword "then", which is the consequent (or right-hand side, RHS) of the rule. In a data set of records to classify, an instance is labelled according to the conditions verified by his attributes. The label class will be assigned to the instance whose attributes verify the condition of a rule r ; in this case, the rule r covers the considered instance. If attributes of the instance do not satisfy any rule, the item label will be that of a default class (Wu et al., 2008).

Building Classification Rules

A method to build a rule-based classifier is to extract rules directly from the data (Kesavaraj and Sukumaran, 2013), as done by PRISM (Cendrowska, 1987), RIPPER (Cohen, 1995), 1R (Holte, 1993) and CN2 (Clark and Niblett, 1989), which employ the so-called Direct Method, that will be explained below. Alternatively, it is possible to construct a rule classifier using an Indirect Method, which extract rules from other classification models, for example, decision trees. The algorithm learns first the tree structure and then converts it in rules. C4.5rules or PART are examples of rule-based classifier build using indirect method.

Direct method: Sequential covering

Sequential covering method aims to create rules that describe or cover many examples of a class and none or very few of other classes. Covering algorithm, descending from *covering strategy* (Michalski, 1969, 1975; Fürnkranz and Flach, 2005), is the typical rule learning method. It consists in iteratively learning one rule from training set and subsequently removing the example covered by that rule from the training set (Fürnkranz, 1999, 2010) (see Fig. 2). The training set is therefore *separate*, and recursively the algorithm learns another rule that covers some of the remaining examples *to conquer* (cf. separate-and-conquer method, Pagallo and Haussler (1990)). In particular, to learn a rule for a class, the algorithm starts from an empty rule, which is unable to discern among the entire input data set. Then, this rule is enriched by adding new attribute test that can describe a fraction of example in training set. As for the decision tree, the rule algorithm selects an attribute used to split data according to the amount of

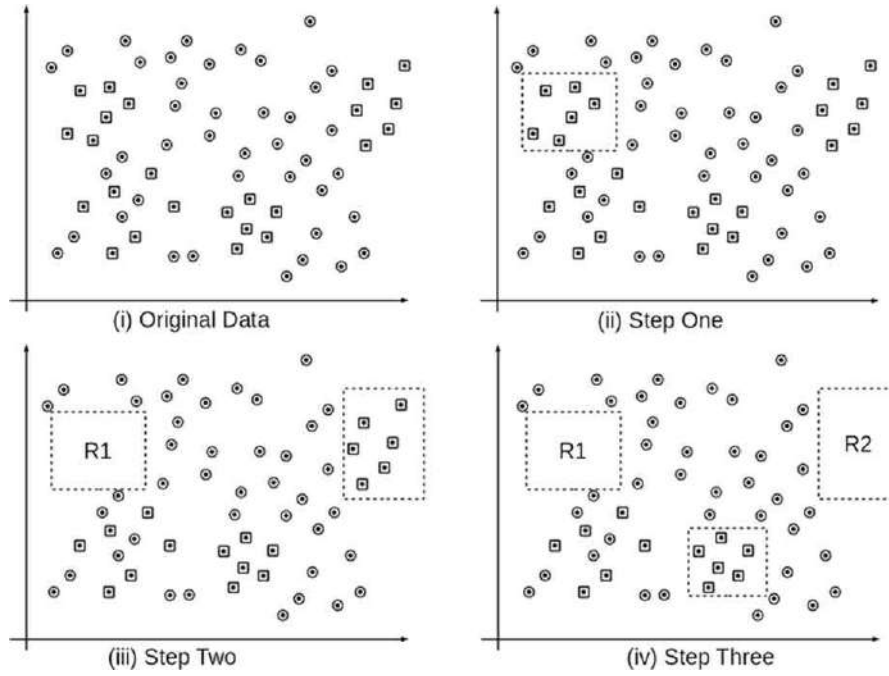


Fig. 2 Successive iterations of coverage learning: fraction of positive examples (squares), covered by a rule, discarded from original data set.

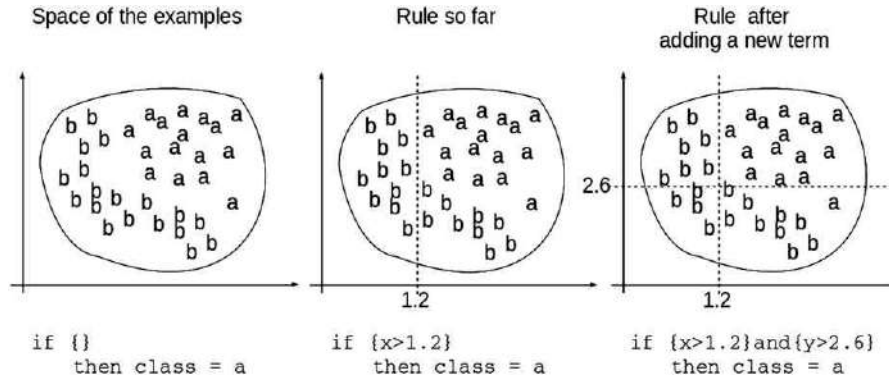


Fig. 3 Enrichment of a rule by adding a new combination of attribute test leads to a reduction of covered examples (left panel) and allows to improve the classification (right panel).

information contained in it by means of some test selection criteria such as accuracy or information gain (for accuracy, see below; for information gain, see the previous section). By adding more conditions to the rule, its coverage is reduced (see Fig. 3). The growing rule continues until training set cannot be split any further or another stopping criterion is met. The pseudo-code for sequential covering algorithm is shown in Fig. 4.

Evaluation of a rule and stopping criterion

Quality of a classification rule can be evaluated by Coverage and Accuracy. *Coverage* of a rule r represents the fraction of instances that satisfy the antecedent of a rule, as expressed by the Eq. (11):

$$\text{Coverage}(r) = \frac{|LHS|}{n} \quad (11)$$

where n is the total number of records in sample considered. An equivalent form for coverage is $C=t/n$, where $t=|LHS|$ is the total number of records covered by rules.

Accuracy of a rule r is the fraction of records covered by the rule that belong to the correct class, in RHS of rule, as shown in Eq. (12):

$$\text{Accuracy}(r) = \frac{|LHS \cap RHS|}{|LHS|} \quad (12)$$

Sequential Covering Algorithm

1. LearnedRules = { }
2. Rule = Learn1Rule (class, Attributes, Threshold)
3. While (Performance (Rule, Examples) > Threshold)
4. LearnedRules = LearnedRules + Rule
5. Examples = Example – Covered (Examples, Rule)
6. Rule = Learn1Rule(class, Attributes, Examples)
7. Sort (LearnedRules, Performance)
8. Return LearnedRules

Fig. 4 Pseudo-code for the sequential covering algorithm.

Rules accuracy can be expressed in the more compact form $A=p/t$, where p are positive examples, i.e., records with class correctly predicted by rule and t is the total number of records covered by rule. It is immediate to consider the difference $t - p$ as the number of errors made by rule. Other metrics of rules evaluation are illustrated in [Fürnkranz \(1999\)](#).

Often, accuracy of a rule is adopted as stopping criterion of rule construction. In particular, the building process stops when accuracy of the rule is maximal, or when increasing in accuracy gets below a given threshold.

When the rule classifier is constructed, *test set* allows evaluate model accuracy.

Conflict resolution and overlap problem

It can occur that one instance triggers more than one rules. In this case, conflict resolution is operated by three possible strategies:

- Size ordering: rules with wider LHS condition have the highest priority.
- Class-based ordering: rules belonging to the same class are grouped together and classes appear by decreasing of prevalence, being class prevalence the fraction of instances that belong to a particular class.
- Rule-based ordering (decision list): rule sets are ordered by priority list, depending on measures of rule quality or experts knowledge.

Some rules can become over specialized, incurring in the overfitting problem. A solution for this inconvenience is to prune the rules, by using *validation set*. Two main strategies are *incremental pruning* and *global pruning* ([Tan et al., 2005](#)).

Classification by Multilayer Feed-Forward Neural Networks (Backpropagation, Sigmoid Function, Learning Rate)

Conventional machine learning techniques require considerable domain expertise and engineering skills to extract right features from raw data and so to detect pattern or classify the input. Artificial Neural Networks (ANN) or also Neural Network (NN) allows solving problems where features detection is difficult. They are used in domains like biomedical field ([Dreiseitl and Ohno-Machado, 2002](#)), computer vision and speech and image recognition, drug discovery and genomics ([LeCun et al., 2015](#)).

Neural Networks are computational methods whose structure follows the information processing model of the biological brain ([McCulloch and Pitts, 1943](#); [Widrow and Hoff, 1960](#); [Rosenblatt, 1962](#); [Rumelhart et al., 1986](#)). Feedforward neural networks architecture is the first and simplest kind of neural network. It represents a powerful tool for handle non-linear classification (see note in the following section) and regression problems.

Difference Between Linear and Non-Linear Classifier

It can be helpful to visualize a geometric description of the classification problem, by plotting the data in a so-called *feature space*. The feature space is a mathematical formalization used to represent the data based on a characterizing collection of data features. The axis in a such space correspond to the features chosen for describe each instance in the data set. A two-class linear classifier discerns between possible cases using a linear combination of the features and a numerical threshold ([Fig. 5](#)). An item will belong to a particular class rather than to the other one if the weighted sum of its features is above or below a given threshold.

Neural network methods take advantage from composing simple non-linear computational modules that deform the input data space to make classes of data separable by a linear decision boundary.

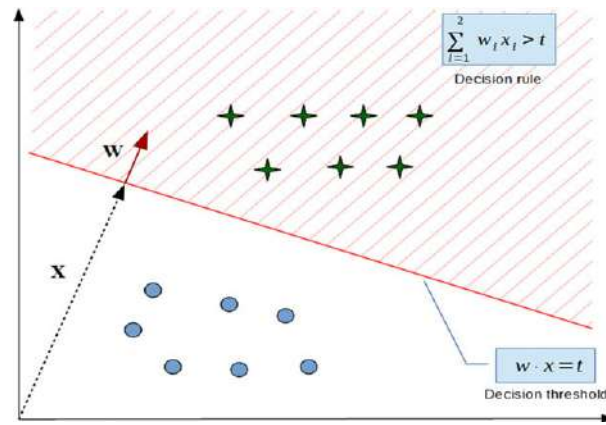


Fig. 5 Two-class linear classification, plotted in a 2-dimensional feature space. The red straight line is the decision boundary among positive and negative examples. $w=(w_1, w_2)$ is the vector of weights and $x=(x_1, x_2)$ is the vector of features on which to perform the classification for a given instance.

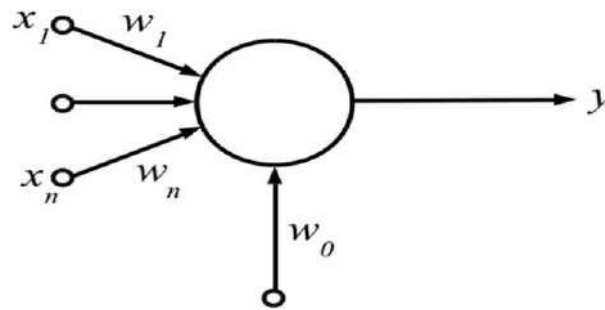


Fig. 6 Schematization of a single neuron.

NN: Perceptron, Multi-Layer Perceptron and Feedforward Neural Network Architecture

Neural network algorithms are used for both supervised and unsupervised learning. To absolve the classification task, supervised neural networks is trained with inputs data and their respective targets or classes.

Neural networks have a peculiar architecture inspired by the biological functioning of neurons. In this analogy, a single neuron, also called perceptron, works as a classifier: for each item to classify, it receives several feature inputs x_i with a corresponding weight w_i and produces an output y (see Fig. 6). The output value of a neuron is called activation a and it is a function of the sum of the weighted inputs: $a = \sum_i w_i x_i$, where the sum run over the total number n of features. There are several types of output or activation function $y=f(a)$, Table 1 shows some types of activation function.

A neural network is a connection of many neurons. In the example of Fig. 7, the outputs of a single neuron in a layer feed each neuron in the successive layer, in the “fully connected” model. The “depth” of the network depends on the number of hidden layers, rising from a simple model of two layer, where only a layer is hidden, until to a multilayer network.

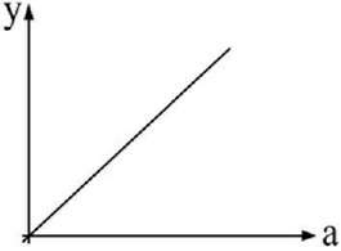
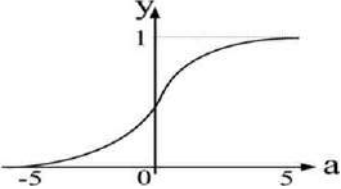
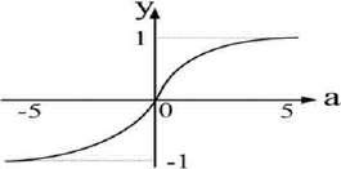
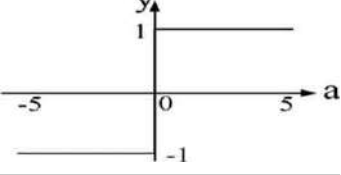
The most simple grouping is *single layer perceptron* (SLP), consisting of an input and an output layer (Jain et al., 1996). *Multilayer perceptron* (MLP) represents an advanced version of the standard SLP, where more layers of computing units connected to each other make a wider network. Unlike SLP, the MLP can discern data non-linearly separable by employing sigmoid non-linear activation function as shown in Fig. 8. The continuity property of sigmoid function will be central in network training.

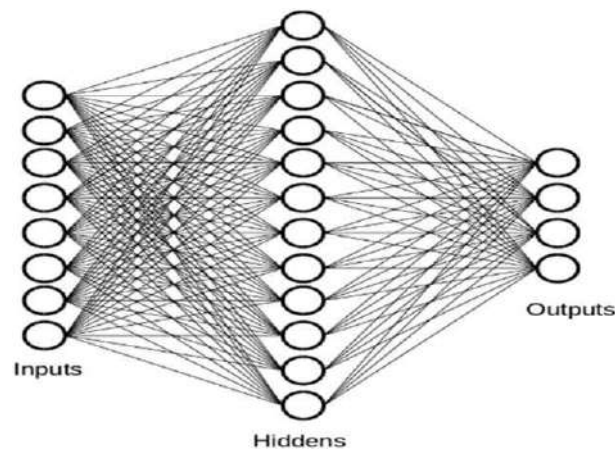
Feedforward neural network (FNN) is a multilayer perceptron where, as occurs in the single neuron, the decision flow is unidirectional, advancing from the input to the output in successive layers, without cycles or loops. This aspect makes FNN the simplest example of neural network.

Training the network

In training phase, a feedforward network examines examples that show the relationship between a feature vector x as input and a target t . The number of times the entire training set is fed to the network is known as *epoch*. From this training data, the network should learn the correct model of classification according to which a given input x correspond to an output y close enough to the right answer t . The ‘distance’ between the target t and the output $y(a)=y(x,w)$ is evaluated for each pair (x,t) by the *objective function*

Table 1 Synoptic table of deterministic activation functions

Activation function	Analytical expression	Graphic representation
Linear	$y(a) = a$	
Sigmoid logistic	$y(a) = \frac{1}{1+e^{-a}};$ $y \in (0,1)$	
Sigmoid tanh	$y(a) = \tanh(a);$ $y \in (-1,1)$	
Step	$y(a) = \begin{cases} 1 & \text{if } a > 0 \\ -1 & \text{if } a \leq 0 \end{cases}$	

**Fig. 7** Example of a two layer, fully connected network (do not count the input layer).

or error function $E = E(y, t)$, which reports how well the network solves the task. The training process consists in a minimization of objective function, by varying the weight parameters w . For function minimization is evaluated also the gradient of the objective function with respect to w by using the *backpropagation algorithm*.

Backpropagation and learning rate

The backpropagation algorithm applies chain rule for derivatives to the objective function $E(y, t)$ with respect to the weights w , relying on gradient descent. Thus $E(y, t)$ and consequently $y(x, w)$ must guarantee both continuity and differentiability. For

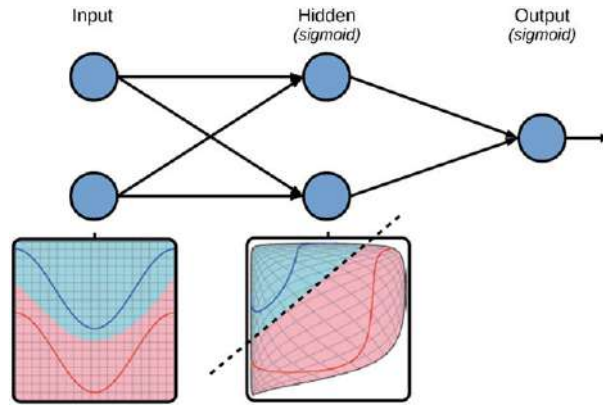


Fig. 8 Multilayer neural network schematized by fully connected black dots: from left to right – two input unit, two units in the hidden layer and one output unit. The hidden layer deforms the input space to obtain a linear decision boundary between classes of data, as highlighted from the shape of the border between colored areas. Reproduced from LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444.

backpropagation networks, Sigmoid activation function is preferred to step function used in perceptron; the objective function of the network is often defined as (using least squares methods as optimization technique. An alternative to optimization function is cross-entropy (with the relative maximum likelihood estimation method). $E = \frac{1}{2} \sum_{i=1}^p (y_i - t_i)^2$, for a training set containing p records. The learning rule for neural network relies on gradient descent procedure that allows minimizing E starting from its gradient $\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_l} \right)$ with respect to l weights. The weights, initialized with random values, represents the only tuning factor to minimize the error function through a change $\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$, where η is the *learning rate* (Royas, 1996). This latter can be a constant value or can be a decreasing function of simulation time τ such as η_0/τ , where τ is an integer counting the algorithm steps. Learning rate denotes the amount of the weights' variation or the length of the step of each iteration in the negative gradient descent.

Support Vector Machines

Support vector machine (SVM) is a supervised machine learning technique based on statistical learning theory (Vapnik, 1995; Scholkopf et al., 1995; Cristianini and Shawe-Taylor, 2000). SVM is a binary classifier (the term “machines” descends from the fact that SVM algorithm produces a binary output), but it can handle problems with many classes and also numerical prediction tasks (Wang and Xue, 2014; Wu et al., 2008). Furthermore, it can treat non-linearly separable data. Similarly to neural network, SVM employ a non-linear mapping of input vectors in a high-dimensional features space, where it can linearly separate data, by so-called optimal separating hyperplanes (Chen et al., 2015).

SVM algorithms allow approach many problems of different domains, such as pattern recognition, medical diagnosis, marketing (Yingxin and Xiaogang, 2005) or text classification (Minsky, 1961; Joachims, 1998).

Linearly Separable Data

Support vector machines mainly works as binary classifier, separating data in a linear way. Data are linearly separable when it is possible to split them in two groups by using a straight line in a two-dimensional input feature space (the feature space is a mathematical formalization used to represent the data based on a characterizing collection of data features) or a plane in a 3D feature space. More generally, for an N -dimensional space the *hyperplane* $N-1$ -dimensional divides data and it represents the decision boundary. There are several ways to separate linearly the same data set, as shown in Fig. 9.

SVM algorithm searches for the optimal separating hyperplane, considering the notion of *margin* m , defined as the distance between the separating hyperplane and the nearest training examples, at least one for each class. In particular, the points closest to the decision boundary, known as *support vectors*, should be far away from it as possible. Indeed, maximizing the margin in training phase makes the classifier more robust and gives better classification performance on test data. In general, training a SVM aims at finding the proper decision boundary or separating hyperplane with the largest margin. Support vectors identify the *optimal* decision boundary and determine the margin, evaluated as the distance between them (Cristianini and Shawe-Taylor, 2000).

As shown in Fig. 10, the margin has an amplitude equal to $m/\|w\|$, measured on w (vector of weights). Therefore, maximizing the margin corresponds to minimizing $\|w\|$, which results in solving a constrained optimization problem, considering a quadratic function such as that shown in formula (13):

$$w^*, t^* = \arg \min_{w,t} \frac{1}{2} \|w\|^2 \text{ subject to the constraint } y_i(w \cdot x_i - t) \geq 1 \quad (13)$$

where $1 \leq i \leq n$ and n is the number of training examples.

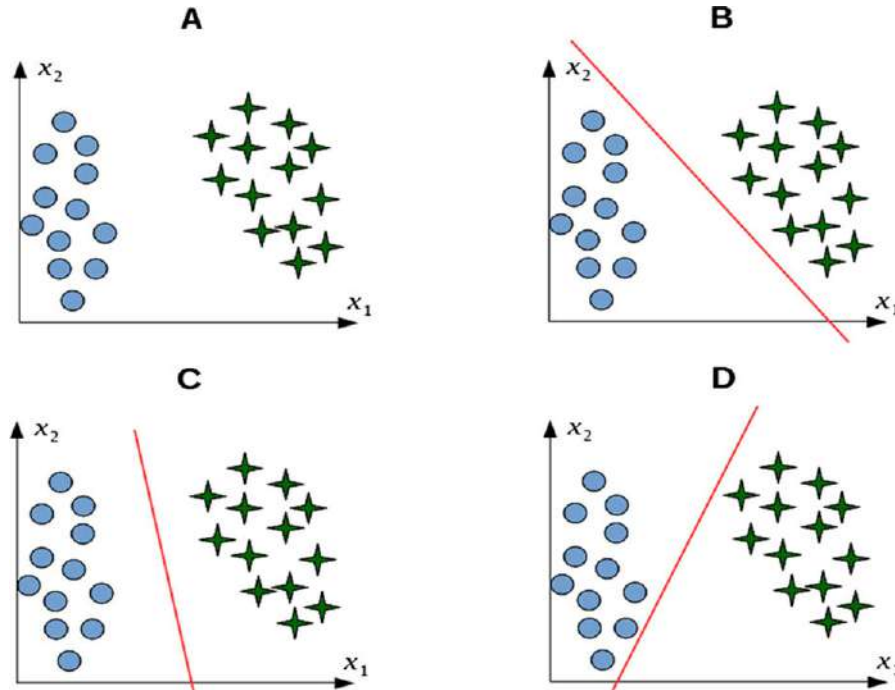


Fig. 9 Linearly separable data (A); different possibilities to separate data are shown in (B)–(D) panels.

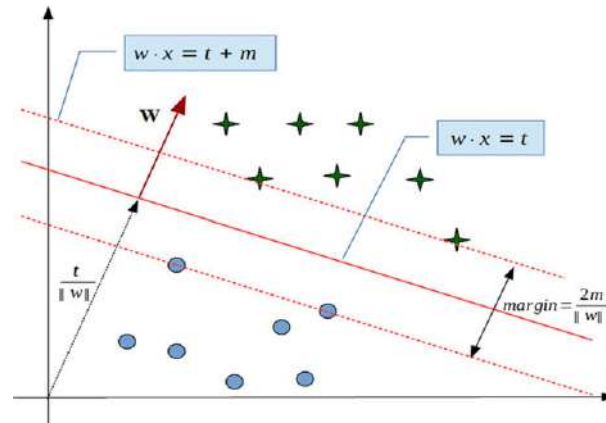


Fig. 10 Geometry of a SVM classifier. Dashed lines delimit the margin. The support vectors are training examples circled and nearest to decision boundary t .

The constraint expresses the request that instances remain outside the margin. In **Fig. 10**, the hyperplanes of equation $w \cdot x_i - t = \pm m$ delimit the margin of separation between two classes y_i of data, positives and negatives respectively. In particular, positives must verify the condition $(w \cdot x_i - t) \geq 1$ and negatives $(w \cdot x_i - t) \leq -1$, where m is set to 1, as usual (the parameters t , $\|w\|$ and m can be rescaled for convenience). The product between classes $y_i = \pm 1$ (assigned in binary supervised problems) and the previous conditions allows formulate a compact constraint (Eq. 13), valid for all instances.

The Lagrange multipliers method approaches the optimization problem applied to the function to minimize under the condition shown by Eq. (13). The Lagrange multipliers algebra allows to obtain the optimal value for the weight ($w = \sum_{i=1}^n \alpha_i y_i x_i$, where α_i are the Lagrange multipliers and n is the number of training examples) w and for the decision boundary t , starting from the pairs (x_i, y_i) , which are provided as input in training phase of SVM. Laid out these values, SVM will establish the class of a new instance according to the following formula (14):

$$y' = \text{sgn}[w \cdot x' - t] \quad (14)$$

The Lagrange function incorporates the constraint preceded by the Lagrange multipliers α_i , as shown in the Eq. (15) (according to the “hinge” loss function formulation (Hastie et al., 2008):

$$\begin{aligned}
\Lambda(\mathbf{w}, t, \alpha_1, \dots, \alpha_n) &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i - t) - 1) \\
&= \frac{1}{2} \|\mathbf{w}\|^2 - \mathbf{w} \cdot \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) + t \left(\sum_{i=1}^n \alpha_i y_i \right) + \sum_{i=1}^n \alpha_i
\end{aligned} \tag{15}$$

By setting to zero the partial derivative of the Lagrange function with respect to t and $\mathbf{w}$, it is possible obtain the following values:

$$\frac{\partial}{\partial \mathbf{w}} \Lambda(\mathbf{w}, t, \alpha_1, \dots, \alpha_n) = 0 \rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \tag{16}$$

$$\frac{\partial}{\partial t} \Lambda(\mathbf{w}, t, \alpha_1, \dots, \alpha_n) = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \tag{17}$$

Returning these values into the Lagrange function (15), the primal problem changes in the dual optimization problem expressed in Eq. (18), entirely depending from the Lagrange multipliers:

$$\begin{aligned}
\Lambda(\alpha_1, \dots, \alpha_n) &= -\frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) \cdot \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) + \sum_{i=1}^n \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^n \alpha_i
\end{aligned} \tag{18}$$

To solve the dual problem it is necessary maximize (the dual Lagrange function (19) is maximized being the α_i preceded by a minus sign in Eq. (15)) the function $\Lambda(\alpha_1, \dots, \alpha_n)$ as described in the following expression:

$$\begin{aligned}
\alpha_1^*, \dots, \alpha_n^* &= \arg \max_{\alpha_1, \dots, \alpha_n} = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^n \alpha_i \\
\text{subject to } \alpha_i &\geq 0, \quad 1 \leq i \leq n, \quad \sum_{i=1}^n \alpha_i y_i = 0
\end{aligned} \tag{19}$$

As shown in the Eq. (19), the optimization problem consists in pairwise dot products between training instances, which is expressed by the element of the Gram matrix $H_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$.

The condition $\alpha_i \geq 0$ is typical for support vector machines (the constraint in Eq. (15) being non-negative) and include two possible cases: (i) $\alpha_i = 0$ for an example $\mathbf{x}_i$ that does not contribute to learn the decision boundary; (ii) $\alpha_i > 0$ only for support vectors, namely, for examples, nearest to the decision boundary. The Lagrange multipliers non-zero determine the decision boundary through the expression (16) and (17) and consequently the class of each new instance through the Eq. (14).

Linearly Inseparable Data

Soft margin

SVM can be adapted to non-linearly separable cases (Cortes and Vapnik, 1995) introducing *margin errors* ξ_i , which allow to some of the training examples to be inside the margin or even at the wrong side of the decision boundary Fig. 11. The *slack variables* ξ_i leads to a *soft margin* optimization problem:

$$w^*, t^*, \xi_i^* = \arg \min_{w, t, \xi_i} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \text{ with the constraint } y_i (\mathbf{w} \cdot \mathbf{x}_i - t) \geq 1 - \xi_i \tag{20}$$

where $\xi_i \geq 0$ and $1 \leq i \leq n$.

C is a "cost" parameter that controls the tradeoff between margin maximization against slack variable minimization. Based on the penalization method, C represents the "modulator" of the slack variables, which work as penalty term for deviations on the wrong side of the margin.

In soft margin case, the Lagrange function becomes the Eq. (21):

$$\begin{aligned}
\Lambda(\mathbf{w}, t, \xi_i, \alpha_1, \dots, \alpha_n, \beta_i) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i - t) - (1 - \xi_i)) - \sum_{i=1}^n \beta_i \xi_i \\
&= \frac{1}{2} \|\mathbf{w}\|^2 - \mathbf{w} \cdot \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) + t \left(\sum_{i=1}^n \alpha_i y_i \right)
\end{aligned}$$

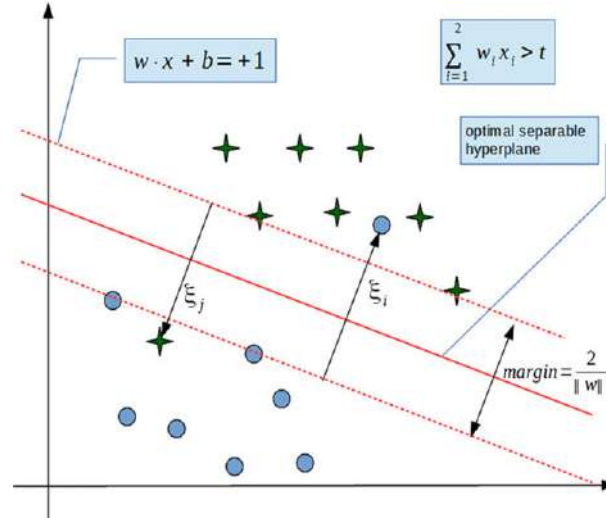


Fig. 11 Non-linear separable case: the distance between the red or blue example into the margin is $-\xi_i / \|w\|$.

$$\begin{aligned}
 & + \sum_{i=1}^n \alpha_i + \sum_{i=1}^n (C - \alpha_i - \beta_i) \xi_i \\
 & = \Lambda(w, t, \alpha_i) + \sum_{i=1}^n (C - \alpha_i - \beta_i) \xi_i.
 \end{aligned} \tag{21}$$

For an optimal solution every partial derivative with respect to ξ_i should be zero, and consequently for all i follows the Eq. (22):

$$C - \alpha_i - \beta_i = 0 \tag{22}$$

Therefore, vanishing the adding term in the last expression of Eq. (21), the non-linear dual problem is reduced to the linear one in formulation (19) with the addition of the upper bound on α_i derived from β_i (22): $0 \leq \alpha_i \leq C$.

With regard to the meaning of the upper bound of α_i , the equality to C implies that $\beta_i = 0$, concerning the case in which training examples are on or inside the margin, being respectively $\xi_i = 0$ or $\xi_i > 0$.

Kernel trick

Non linearly separable data can be treated plotting them in a higher dimensional feature space through a mathematical transformation known as *kernel function* k . Such function replaces the inner products between pairs of input instances $x_i \cdot x_j$ in the SVM algorithm for linear separable data, and allows to represent training examples without the explicit computation of their coordinates in the new feature space. The formulas (23) and (24) express that in symbol:

$$k : x \rightarrow \Phi(x) \tag{23}$$

and

$$H_{ij} = \Phi(x_i) \cdot \Phi(x_j) = k(x_i, x_j) \tag{24}$$

where $\Phi(x)$ is the feature mapping of the instance x in the new feature space, performed by the kernel function k and $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ is the Kernel matrix element, which provides the Gram matrix element H_{ij} already seen above transformed by k . Popular kernels are listed in Table 2.

With kernel function, the mapping in a higher dimensional space can make possible separate linearly data that in the starting feature space are non-linearly separable, as shown in Fig. 12.

Finally, for each new instance x' the class is established through the expression:

$$y' = \text{sgn}[w \cdot \Phi(x') - t] \tag{25}$$

where $w \cdot \Phi(x') = \sum_{i=1}^n \alpha_i y_i \Phi(x_i) \cdot \Phi(x') = \sum_{i=1}^n \alpha_i y_i k(x_i, x')$.

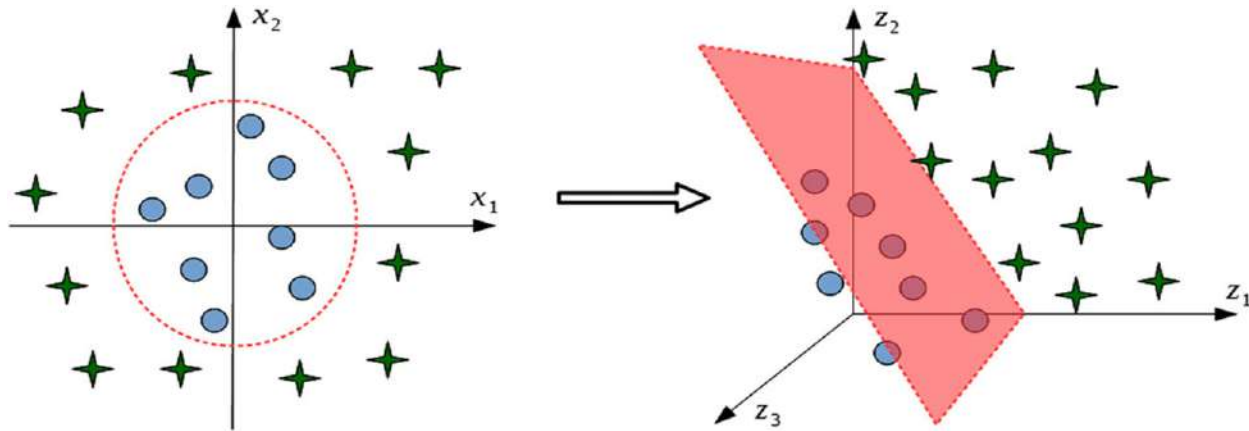
This means that classifying a new instances involves training examples with non-zero Lagrange multipliers.

Multiclass SVM

It is possible to use additional methods that extend the SVM classification to more than two classes. (Weston and Watkins, 1999; Crammer and Singer, 2001; Hsu and Lin, 2002; Wang and Xue, 2014). Among the strategies aimed to adapt SVM to multiclass classification, there are two common techniques: *one-versus-all* (Vapnik, 1998) and *one-versus-one* (Kreßel, 1999). Both approaches decompose the single multiclass problem into a set of binary classification problems. To solve a problem with k classes, the one-versus-all approach trains k SVM binary classifiers so that i -th model recognizes as positives the examples in the i -th class and as

Table 2 a, b, γ are parameters defining the behavior of kernel function

Kernel function	Analytical expression
Linear	$k_0(x_i, x_j) = x_i \cdot x_j$
Polynomial	$k_1(x_i, x_j) = (x_i \cdot x_j + a)^b$
Gaussian	$k_2(x_i, x_j) = \exp\left(-\frac{\ x_i - x_j\ ^2}{2\gamma^2}\right)$
Sigmoidal	$k_4(x_i, x_j) = \tanh(a(x_i \cdot x_j) - b)$

**Fig. 12** Left: in the starting 2D feature space data are not linearly separable; right: kernel trick allows to map data in a 3D new feature space, where data are linearly separable by means of a hyperplane.

negatives all the rest of examples. In the test phase for a given input, the binary classifier that provides positive output value assigns the class label.

The one-versus-one (Kreßel, 1999) method solves the k classes problem with a pairwise decomposition of classifiers. This approach uses $k(k-1)/2$ binary classifiers, each one of them is trained to identify as positive a given class and negative all the rest (in a similar way as described above). In the test phase, a given input instance is evaluated or “voted” from pairs of classifiers. The class label of this test instance will arise from the classifier that gives the highest number of votes.

Associative Classification: Classification Based on Association Rules

The rules syntax concerns different machine learning areas. As shown above (see Section Rule-Based Classification), supervised learning employs rules to absolve some classification tasks. But rules paradigm also take place in the unsupervised learning, with the association rules (Cios et al., 2007). Both approaches have a large use for many application domains (Gosain and Bhugra, 2013), according to their proper characteristics and differences.

Introduction of Association Rules

Association rules (AR) (Agrawal et al., 1993) represent a technique of unsupervised learning focused on the identification of relationship of co-occurrence, named associations, among items in a data set. The original application of this kind of rules is market basket data analysis, aimed at discovering possible regularities of association among items on sale and therefore to support market decision making. AR are relevant for cross-marketing. However, many other applications range in different contexts, from Web and text document (Kulkarni and Kulkarni, 2016) to Bioinformatics domain (Atluri et al., 2009).

Rule-Based Classification section is based on a “small set” of rules mined from a database in order to achieve as accurate as possible result. The objective of AR mining is finding in the database “all” rules that satisfy a given minimum support and minimum confidence threshold (Agrawal and Srikant, 1994). A basic difference between them is that the target of classification rule mining is pre-established (namely the class), unlike to what happens in association rule mining.

Associative classification originates from the idea of integrating classification rule mining and association rule mining in order to gain advantages from both methods, arising computational performances (Liu et al., 1998). This integration is focused on the subset of AR that have classification class attribute on the right-hand-side. These special rules are also known as *class association*

rules (CARs). The first implementation of associative classification is the *Classification Based on Association* algorithm (CBA) (Liu *et al.*, 1998).

Formalism of association rules

To introduce the definitions of this section, consider a market basket analysis context. The database is a set T of transactions each of which consists of a set X of items bought in that transaction. In turn, given an object or a set of objects, it is possible to list the transactions that contain them. An item set X covers a transactions set T if this item set appears in every transaction of T . *Support count* of an item set X , denoted by $X.count$, is the number of transactions in T covered by X . It represents the frequency with which an item set X appears in a database of transactions T (Liu, 2011).

An expression of the form $X \rightarrow Y$ represents an AR, where X and Y are disjoint sets of purchasable items. A such rule states that transactions of the database T which contain X tend to contain Y , as expressed by statement of the kind “98% of customers that purchase tires and automobile accessories also get automotive service” offered by Agrawal *et al.* (1995).

In association rule mining, all rules extract will be evaluated based on their strength, selecting only a subset that satisfy predetermined conditions. *Support* and *confidence* of a rule measure its strength. The rule support express the portion of transactions containing $X \cup Y$ over the total number n of transactions of T as shown by the formula (26):

$$support = \frac{(X \cup Y).count}{n} \quad (26)$$

The confidence of a rule $X \rightarrow Y$ is the per cent of transactions of T containing X that contain also Y :

$$confidence = \frac{(X \cup Y).count}{X} \quad (27)$$

The aim of mining AR is to find all association rules in T with support and confidence greater than or equal to a threshold value pre-specified named *minimum support* (*minsup*) and *minimum confidence* (*minconf*).

In order to highlight AR among basket data it is necessary to discover item sets that occurs recurrently (Gosain and Bhugra, 2013). Classical algorithms to mine AR are *Apriori* (Agrawal and Srikant, 1994) and *FP-growth* (Han *et al.*, 2000).

CBA

The CBA algorithm consists of two parts: i) a *rule generator* (CBA-RG), based on the *Apriori* algorithm (Agrawal and Srikant, 1994) appropriately modified to mine class association rules; ii) a *classifier builder* (CBA-CB) that classifies accurately the rules previously generated. Comparing CBA with other classifier based on rules, such as C4.5 (Quinlan, 1993) and CART (Breiman *et al.*, 1984), some differences emerge. Both approaches are based on the “covering method” (Michalski, 1980), but results are different, depending on the way of application. CBA algorithm (CBA-RG module) adopts covering method, running it over all the rules mined from the entire training set. Unlike, the traditional application of covering method works on one class at a time (see also previous section on rules-based classifier), whenever removing covered examples by best rules, and learning new rules from the remaining cases. The difference among results of CBA and the other classifiers arises from the general validity of rules mined by CBA that constructs all the rules from the entire training set and then makes the best selection of rules covering training data. Furthermore, rule-pruning technique is adopted in CBA to avoid overfitting problems. Pruning represents a difference between CBA algorithm and the association rule mining. Another difference lies in the presence of CBA-CB module, being the AR not based on the classifier building.

Classification Based on Multiple Association Rules

Classification based on Multiple Association Rules or CMAR is a method following to CBA (Li *et al.*, 2001). It introduces some advances with respect to CBA. First, CMAR identifies the class label analyzing a small set of rules with high confidence instead of mine the class from a single rule. CMAR uses *FP-growth* method (Han *et al.*, 2000), a frequent pattern mining algorithm, alternative to *Apriori* method, to mine class-association rules. Then, CMAR employs a *CR-tree* data structure to store and retrieve the large amount of classification rules previously generated. *CR-tree* represents a solution to the remarkable effort handling a substantial amount of rules. Finally, pruning make the classification process more effective and efficient, by removing redundant and noisy information.

To classify a new object, CMAR compares classification rules that match the given object. The object class is simply assigned when the class label is the same for all rules matching. Otherwise, CMAR groups together rules with the same class label. Then, it evaluates the strength of each group, comparing the “combined effect” of correlation (Brin *et al.*, 1997) and support (Baralis *et al.*, 2004) of contained rules.

Classification Based on Predictive Association Rules

Classification Based on Predictive Association Rules or CPAR (Yin and Han, 2003) is a classification approach that merges the advantages of (i) associative classification and (ii) traditional rule-based classification. It adopts different strategies with respect to (i) and (ii) about the

construction of rules, for example, it generates rules directly from training data, instead of creating a large set of candidate rules, as in associative classification; moreover, it produces a more complete set of rules than traditional rule-based approaches.

CPAR originates from First Order Inductive Learner (FOIL) learning system (Quinlan, 1990; Quinlan and Cameron-Jones, 1993), modified with Predictive Rule Mining (PRM) algorithm. PRM varies the covering method (see Section Rule-Based Classification) implemented by FOIL, keeping the rules that cover positive examples and associating a “weight” to these rules. In this way, positives examples are covered by more rules, achieving higher accuracy than FOIL. Furthermore, working with large database, PRM attains greater efficiency with respect to FOIL, thanks to reducing of time consuming part in rules building. PRM reaches higher efficiency but lower accuracy than associate classification. PRM results are exceeded by CPAR, that is the evolution of PRM. It gets better the rule building process, avoiding redundant rule generation and building several rules simultaneously. Finally, CPAR evaluates the prediction power of every rule, defining its expected accuracy as the probability that an example observing the antecedent of the rule actually belongs to class predicted in the consequent. So doing, in a set of k best rules for a class, the rule with the highest expected accuracy will be chosen to class prediction.

See also: Data Mining in Bioinformatics. The Challenge of Privacy in the Cloud

References

- Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I., 1995. Fast discovery of association rules. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds.), *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: AAAI/MIT Press, pp. 307–329.
- Agrawal, R., Srikant R., 1994. Fast algorithms for mining association rules. In: *Proceedings of International Conference on Very Large Data Bases (VLDB)*, San Jose, CA.
- Agrawal, R., Tomasz I., Swami, A., 1993. Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (Eds.), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207–216. New York, NY: ACM.
- Atluri, G., Gupta, R., Fang, G., *et al.*, 2009. Association analysis techniques for bioinformatics problems. In: Rajasekaran, S. (Ed.), *Bioinformatics and Computational Biology*. New Orleans, LA: Springer, pp. 1–13.
- Baralis, E., Chiusano, S., Garza, P., 2004. On support thresholds in associative classification. In: *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 553–558. New York, NY: ACM.
- Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. *Classification and Regression Trees*. Belmont, CA: Wadsworth.
- Brin, S., Motwani, R., Silverstein, C., 1997. Beyond market baskets: Generalizing associations rules to correlations. In: *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pp. 265–276. New York, NY: ACM.
- Cendrowska, J., 1987. PRISM: An algorithm for inducing modular rules. *International Journal of ManMachine Studies* 27, 349–370.
- Chen, F., Deng, P., Wan, J., *et al.*, 2015. Data mining for the internet of things: Literature review and challenges. *International Journal of Distributed Sensor Network* 2015, 1–15.
- Cios, K.J., Swiniarski, R.W., Pedrycz, W., Kurgan, L.A., 2007. Unsupervised learning: Association rules. In: Kecman, V. (Ed.), *Data Mining*. Springer, pp. 289–306.
- Clark, P., Niblett, T., 1989. The CN2 induction algorithm. *Machine Learning* 3, 261–283.
- Cohen, W.W., 1995. Fast effective rule induction. In: *Proceedings of the 12th International Conference on Machine Learning*, pp. 115–123. San Mateo, CA: Morgan and Kaufmann.
- Cortes, C., Vapnik, V., 1995. Support-vector networks. *Machine Learning* 20 (3), 273–297.
- Crammer, K., Singer, Y., 2001. On the algorithmic implementation of multiclass kernel-based machines. *Journal of Machine Learning Research* 2, 265–292.
- Cristianini, N., Shawe-Taylor, J., 2000. *An Introduction to Support Vector Machines*. New York, NY: Cambridge University Press.
- Reisert, S., Ohno-Machado, L., 2002. Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics* 35 (5), 352–359.
- Fürnkranz, J., 1999. Separate-and-conquer rule learning. *Artificial Intelligence Review* 13, 3–54.
- Fürnkranz, J., 2010. Rule learning. In: Sammut, C., Webb, G.I. (Eds.), *Encyclopedia of Machine Learning*. New York, NY: Springer, pp. 875–879.
- Fürnkranz, J., Flach, P.A., 2005. Roc 'n' rule learning – Towards a better understanding of covering algorithms. *Machine Learning* 58, 39–77.
- Gosain, A., Bhugra, M., 2013. A comprehensive survey of association rules on quantitative data in data mining. In: *IEEE Conference Information & Communication Technologies*, JeJu Island, pp. 1003–1008.
- Han, J., Kamber, M., Pei, J., 2012. *Data mining, Concepts and Techniques*, third ed. Waltham, MA: Morgan Kaufmann.
- Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD International conference on Management of Data*, pp. 1–12. New York, NY: ACM.
- Hastie, T., Tibshirani, R., Friedman J., 2008. *The Elements of Statistical Learning*, second ed. Stanford, CA: Springer.
- Holte, R.C., 1993. Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11, 63–90.
- Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13 (2), 415–425.
- Hunt, E.B., 1962. *Concept Learning: An Information Processing Problem*. New York, NY: Wiley.
- Hunt, E.B., Marin, J., Stone, P.J., 1966. *Experiments in Induction*. New York, NY: Academic Press.
- Jain, A.K., Mao, J., Mohiuddin, K.M., 1996. Artificial neural networks: A tutorial. *IEEE Computer-Special Issue in Neural Computing* 29 (3), 31–44.
- Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In: Hutchison, D., Kanade, T., Kittler, J., *et al.* (Eds.), *Machine Learning: European Conference on Machine Learning (in LNCS 1398)*, pp. 137–142. Berlin: Springer.
- Kesavaraj, G., Sukumaran, S., 2013. A study on classification techniques in data mining. In: *Proceedings of the International Conference on Computer Communication and Networking Technologies (ICCCNT'13)*, pp. 1–7. Tamil Nadu: IEEE.
- Kreßel, U., 1999. Pairwise classification and support vector machines. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), *Advances in Kernel Methods: Support Vector Learning*. Cambridge: MIT Press, pp. 255–268.
- Kulkarni, M., Kulkarni, S., 2016. Knowledge discovery in text mining using association rule extraction. *International Journal of Computer Applications* 143, 30–36.
- Lavanya, D., Rani, K.U., 2011. Performance evaluation of decision tree classifiers on medical datasets. *International Journal of Computer Applications* 26, 1–4.
- Lawrence, R.L., Wright, A., 2001. Rule-based classification systems using classification and regression trees (CART) analysis. *Photogrammetric Engineering & Remote Sensing* 67, 1137–1142.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444.
- Li, T., Zhang, C., Ogihara, M., 2004. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics* 20, 2429–2437.
- Liu, B., 2011. *Web data mining. Exploring Hyperlinks, Contents and Usage Data*. Chicago, IL: Springer.

- Liu, B., Hsu, W., Ma, Y., 1998. Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 80–86. New York, NY: AAAI Press.
- Li, W., Han, J., Pei, J., 2001. CMAR: Accurate and efficient classification based on multiple class-association rules. In: Proceedings of the IEEE International Conference on ICDM'01, pp. 369–376. IEEE: San Jose, CA.
- McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5 (4), 115–133. Reprinted in Anderson and Rosenfeld, Springer.
- Michalski, R., 1980. Pattern recognition as rule-guided induction inference. *IEEE Transaction On Pattern Analysis and Machine Intelligence* 2, 349–361.
- Michalski, R.S., 1969. On the quasi-minimal solution of the general covering problem. In: Proceeding of the V International Symposium on the Information Processing, Bled, Yugoslavia, pp. 125–128.
- Michalski, R.S., 1975. Synthesis of optimal and quasi-optimal variable-valued logic formulas. In: Proceedings of the 1975 International Symposium on Multiple-Valued Logic, pp. 76–87. Bloomington, Indiana.
- Minsky, M., 1961. Steps toward artificial intelligence. In: Hamburger, F. (Ed.), *Proceedings of the IRE*, vol. 49 (No. 1), pp. 8–30. New York: IEEE.
- Pagallo, G., Haussler, D., 1990. Boolean feature discovery in empirical learning. *Machine Learning* 5, 71–99.
- Quinlan, J.R., 1986. Induction of decision trees. *Machine Learning* 1, 81–106.
- Quinlan, J.R., 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann.
- Quinlan, J.R., 1990. Learning Logical Definitions from Relations. *Machine Learning*, 5. Boston: Kluwer Academic, pp. 239–266.
- Quinlan, J.R., Cameron-Jones, R.M., 1993. FOIL: A midterm report. In: Brazdil, P.B. (Ed.), *Proceedings of European Conference on Machine Learning*, pp. 3–20. Vienna: Springer-Verlag.
- Raileanu, L.E., Stoffel, K., 2004. Theoretical comparison between the Gini Index and Information Gain criteria. *Annals of Mathematics and Artificial Intelligence* 41, 77–93.
- Rokach, L., Maimon, O., 2005. Decision trees. In: Rokach, L., Maimon, O. (Eds.), *Data Mining and Knowledge Discovery Handbook*. New York, NY: Springer, pp. 165–192.
- Rosenblatt, F., 1962. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism*. Washington, DC: Spartan Books.
- Royas, R., 1996. *Neural Networks. A Systematic Introduction*. Berlin: Springer.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L., PDP Research Group. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition 1*. MIT Press, pp. 318–362.
- Scholkopf, B., Burges, C., Vapnik, V., 1995. Extracting support data for a given task. In: Fayyad, U.M., Uthurusamy, R. (Eds.), *Proceedings of the First International Conference on Knowledge discovery and Data Mining*, pp. 252–257. Menlo Park, CA: AAAI Press.
- Tan, P.N., Steinbach, M., Kumar, V., 2005. *Introduction to Data Mining*. Boston, MA: Pearson Addison Wesley.
- Vapnik, V., 1995. *The Nature of Statistical Learning Theory*. New York, NY: Springer.
- Vapnik, V., 1998. *Statistical Learning Theory*. New York, NY: Wiley.
- Wang, Z., Xue, X., 2014. Multi-class support vector machine. In: Ma, Y., Guo, G. (Eds.), *Support Vector Machines Applications*, first ed. Cham: Springer International Publishing, pp. 23–48.
- Weston, J., Watkins, C., 1999. Support vector machines for multi-class pattern recognition. In: Verleysen, M. (Ed.), *Proceedings of European Symposium on Artificial Neural Networks*, pp. 219–224. Bruges: ESANN.
- Widrow, B., Hoff, M.E., 1960. Adaptive switching circuits. In: *IRE WESCON Convention Record*, vol. 4, pp. 96–104. Reprinted in Anderson and Rosenfeld (1988).
- Wu, X., Kumar, V., Quinlan, R.J., *et al.*, 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems* 14, 1–37.
- Yingxin, L., Xiaogang, R., 2005. Feature selection for cancer classification based on support vector machine. *Journal of Computer Research and Development* 42 (10), 1796–1801.
- Yin, X., Han, J., 2003. CPAR: Classification based on predictive association rules. In: Barbara, D., Kamath, C. (Eds.), *Proceedings of the 2003 SIAM International Conference on Data Mining*, pp. 331–335. San Francisco, CA: Society for Industrial and Applied Mathematics.
- Zhou, L., Wang, Q., Fujita, H., 2017. One versus one multi-class classification fusion using optimizing decision directed acyclic graph for predicting listing status of companies. *Information Fusion* 36, 80–89.

Further Reading

- Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*. Singapore: Springer.
- Duda, R.O., Hart, P.E., Stork, D.G., 1991. *Pattern Classification*. New York, NY: John Wiley & Sons.
- Flach, P., 2012. *Machine Learning*. New York, NY: Cambridge University Press.
- Harrington, P., 2012. *Machine Learning in Action*. Shelte Island, NY: Manning.
- MacKay, D.J.C., 2005. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.
- McKinney, W., 2012. *Python for Data Analysis*. Sebastopol, CA: O'Reilly.
- Mitchell, T.M., 1997. *Machine Learning*. Portland, OR: McGraw Hill.
- Model, M.L., 2009. *Bioinformatics Programming using Python*. Sebastopol, CA: O'Reilly.
- Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B., 2001. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* 12 (2), 181–201.
- Phyu, T.N., 2009. Survey of classification techniques in data mining. In: Ao, S.I., Castillo, O., Douglas, C., Feng, D.D., Lee, J.A. (Eds.), *Proceedings of International Multi-Conference of Engineers and Computer Scientists*, pp. 727–731. Hong Kong: Newswood Limited.
- Schölkopf, B., Smola, A.J., 2002. *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.
- Shawe-Taylor, J., Cristianini, N., 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.
- Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6, 1453–1484.
- Witten, I.H., Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, second ed. San Francisco, CA: Morgan Kaufmann.
- Wu, X., Kumar, V., 2009. *The Top Ten Algorithms in Data Mining*. Boca Raton, FL: Chapman & Hall/CRC.

Biographical Sketch



Alfonso Urso received the Laurea degree in electronic engineering (*Summa cum Laude*), and the PhD degree in systems engineering from the University of Palermo, Italy, in 1992 and 1997, respectively. In 1998, he was with the Department of Computer and Control Engineering, University of Palermo, as Research Associate. In 2000, he joined the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR), where he is currently a Researcher in systems and computer engineering. From 2007 to 2015 he was coordinator of the research group “Intelligent Data Analysis for Bioinformatics” of the Italian National Research Council. Since January 2016 he is Head of the Palermo branch of the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR). Since 2001 he has been Lecturer at the University of Palermo. His research interests are in the area of machine learning, soft computing and applications to bioinformatics. He is member of the Institute of Electrical and Electronic Engineers (IEEE), and of the Italian Bioinformatics Society (BITS).



Antonino Fiannaca received the Laurea degree in computer science engineering (*Summa cum Laude*), and the PhD degree in computer science from the University of Palermo, Italy, in 2006 and 2011, respectively. In 2006, he joined the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR), where he is currently a Researcher. He is part of the Translational Bioinformatics Laboratory at ICAR-CNR. His research interests are in the area of machine learning, neural networks and bioinformatics.



Massimo La Rosa received the Laurea degree in computer science engineering (*Summa cum Laude*), and the PhD degree in computer science from the University of Palermo, Italy, in 2006 and 2011, respectively. In 2006, he joined the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR), where he is currently a Researcher. He is part of the Translational Bioinformatics Laboratory at ICAR-CNR. He is member of Bioinformatics Italian Society (BITS) from 2016. His research interests are in the area of machine learning, data mining and bioinformatics.



Valentina Ravi received her Master Degree in Physics, with Biophysical address (*Summa cum Laude*), at the University of Study of Palermo, Italy. She has held post-graduate internships at the same University, where she collaborated to tutoring, teaching and dissemination activities. She has done a post-degree training in the area of interest of data mining and support decision systems at ICAR-CNR of the National Research Council, Palermo. Here she is currently involved in research activities in bioinformatics domain, with the focus on the study of data mining techniques.



Riccardo Rizzo is staff researcher at Institute for high performance computing and networking, Italian National Research Council. His research is focused mainly on machine learning methods of sequence analysis, and biomedical data analysis, and on neural networks applications. He was Visiting Professor at the University of Pittsburgh, School of Information Science, in 2001. He participates in the program committee of several international conferences and has served as editor of some Supplements of the BMC Bioinformatics journal. He is author of more than 100 scientific, 33 of them in international journals, such as "IEEE Transactions on Neural Networks", "Neural Computing and Applications", "Neural Processing Letters", "BMC Bioinformatics". He participates in many national research projects and is co-author of one international patent.

Bayes' Theorem and Naive Bayes Classifier

Daniel Berrar, Tokyo Institute of Technology, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Bayes' theorem is of fundamental importance for inferential statistics and many advanced machine learning models. Bayesian reasoning is a logical approach to updating the probability of hypotheses in the light of new evidence, and it therefore rightly plays a pivotal role in science (Berry, 1996). Bayesian analysis allows us to answer questions for which frequentist statistical approaches were not developed. In fact, the very idea of assigning a probability to a hypothesis is not part of the frequentist paradigm.

The goal of this article is to provide a mathematically rigorous yet concise introduction to the foundation of Bayesian statistics: *Bayes' theorem*, which underpins a simple but powerful machine learning algorithm: the *naive Bayes classifier* (Lewis, 1998). This article is self-contained; it explains all terms and notations in detail and provides illustrative examples. As a tutorial, this text should therefore be easily accessible to readers from various backgrounds. As an encyclopedic article, it provides a complete reference for bioinformaticians, machine learners, and statisticians. Readers who are already familiar with the statistical background may find the practical examples in Section "Examples" most useful. Specifically, Section "Examples" highlights some caveats and pitfalls (and how to avoid them) in building a naive Bayes classifier using R, with additional materials available at the accompanying website <http://osf.io/92mes>.

Fundamentals

Basic Notation and Concepts

A statistical experiment can be broadly defined as a process that results in one and only one of several possible outcomes. The collection of all possible outcomes is called the *sample space*, denoted by Ω . At the introductory level, we can describe events by using notation from set theory. For example, our experiment may be one roll of a fair die. The sample space is then $\Omega = \{1, 2, 3, 4, 5, 6\}$, which is also referred to as *universal set* in the terminology of set theory. A simple event is, for instance, the outcome 2, which we denote as $E_1 = \{2\}$. The probability of an event E is denoted by $P(E)$. According to the classic concept of probability, the probability of an event E is the number of outcomes that are favorable to this event, divided by the total number of possible outcomes for the experiment, $P(E) = \frac{|E|}{|\Omega|}$, where $|E|$ denotes the cardinality of the set E , i.e., the number of elements in E . In our example, the probability of rolling a 2 is $P(E_1) = \frac{|E_1|}{|\Omega|} = \frac{1}{6}$. The event "the number is even" is a compound event, denoted by $E_2 = \{2, 4, 6\}$. The cardinality of E_2 is 3, so the probability of this event is $P(E_2) = \frac{3}{6}$.

The *complement* of E is the event that E does not occur and is denoted by E^c , with $P(E) = 1 - P(E^c)$. In the example, $E_2^c = \{1, 3, 5\}$. (In the literature, the complement of an event A is also often represented by the symbol $\bar{A}$.) Furthermore, $P(A|B)$ denotes the *conditional probability* of A given B . Finally, $\emptyset$ denotes the *empty set*, i.e., $\emptyset = \{\}$.

Let A and B be two events from a sample space Ω , which is either finite with N elements or countably infinite. Let $P: \Omega \rightarrow [0, 1]$ be a probability distribution on Ω , such that $0 < P(A) < 1$ and $0 < P(B) < 1$ and, obviously, $P(\Omega) = 1$. We can represent these events in a Venn diagram (Fig. 1(a)). The *union* of the events A and B , denoted by $A \cup B$, is the event that either A or B or both occur. The *intersection* of the events A and B , denoted by $A \cap B$, is the event that both A and B occur. Finally, two events, A and B , are called *mutually exclusive* if the occurrence of one of these events rules out the possibility of occurrence of the other event. In the notation of set theory, this means that A and B are disjoint, i.e., $A \cap B = \emptyset$. Two events A and B , with $P(A) > 0$ and $P(B) > 0$, are called *independent* if the occurrence of one event does not affect the probability of occurrence of the other event, i.e., $P(A|B) = P(A)$ or $P(B|A) = P(B)$, and $P(A \cap B) = P(A) \cdot P(B)$.

Note that the *conditional probability*, $P(A|B)$, is the *joint probability* $P(A \cap B)$ divided by the *marginal probability* $P(B)$. This is a fundamental relation, which has a simple geometrical interpretation. Loosely speaking, given that we are in the ellipse B (Fig. 1(a)), what is the probability that we are also in A ? To be also in A , we have to be in the intersection $A \cap B$. Hence, the probability is equivalent to the number of elements in the intersection, $|A \cap B|$, divided by the number of elements in B ,

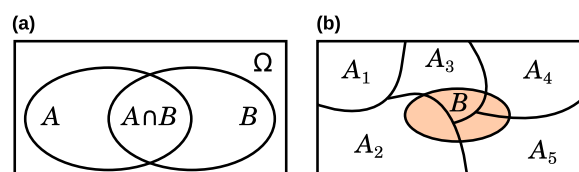


Fig. 1 (a) Venn diagram for sets A and B . (b) Illustration of the total probability theorem. The sample space Ω is divided into five disjoint sets A_1 to A_5 , which partly overlap with set B .

i.e., $|B|$. Formally,

$$P(A|B) = \frac{|A \cap B|}{|B|} = \frac{|A \cap B|/|\Omega|}{|B|/|\Omega|} = \frac{P(A \cap B)}{P(B)}.$$

Total Probability Theorem

Before deriving Bayes' theorem, it is useful to consider the *total probability theorem*. First, the addition rule for two events, A and B , is easily derived from Fig. 1(a):

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \quad (1)$$

We assume that the sample space can be divided into n mutually exclusive events A_i , $i = 1..n$, as shown in Fig. 1(b). Specifically,

1. $A_1 \cup A_2 \cup \dots \cup A_n = \Omega$
2. $A_i \cap A_j = \emptyset$ for $i \neq j$
3. $A_i \neq \emptyset$

From Fig. 1(b), it is obvious that B can be stated as

$$B = (B \cap A_1) \cup (B \cap A_2) \cup \dots \cup (B \cap A_n)$$

and we obtain the total probability theorem as

$$\begin{aligned} P(B) &= P(B \cap A_1) + P(B \cap A_2) + \dots + P(B \cap A_n) - \underbrace{P(B \cap A_1 \cap \dots \cap B \cap A_n)}_{= 0, \text{ because } A_i \cap A_j = \emptyset \text{ for } i \neq j} \\ &= P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \dots + P(B|A_n)P(A_n) \\ &= \sum_{i=1}^n P(B|A_i)P(A_i) \end{aligned} \quad (2)$$

which can be rewritten as

$$P(B) = P(B|A)P(A) + P(B|A^c)P(A^c) \quad (3)$$

because $A_2 \cup A_3 \cup \dots \cup A_n$ is the complement of A_1 (cf. conditions 1 and 2 above). Redefining $A := A_1$ and $A^c := A_2 \cup A_3 \cup \dots \cup A_n$ gives Eq. (3).

Bayes' Theorem

Assuming that $|A| \neq 0$ and $|B| \neq 0$, we can state the following:

$$P(A|B) = \frac{|A \cap B|}{|B|} = \frac{\frac{|A \cap B|}{|\Omega|}}{\frac{|B|}{|\Omega|}} = \frac{P(A \cap B)}{P(B)} \quad (4)$$

$$P(B|A) = \frac{|B \cap A|}{|A|} = \frac{\frac{|B \cap A|}{|\Omega|}}{\frac{|A|}{|\Omega|}} = \frac{P(A \cap B)}{P(A)} \quad (5)$$

From Eqs. (4) and (5), it is immediately obvious that

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) \quad (6)$$

and therefore

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (7)$$

which is the simplest (and perhaps the most memorable) formulation of Bayes' theorem.

If the sample space Ω can be divided into finitely many mutually exclusive events $A_1, A_2, \dots, A_n$, and if B is an event with $P(B) > 0$, which is a subset of the union of all A_i , then for each A_i , the generalized Bayes' formula is

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^n P(B|A_j)P(A_j)} \quad (8)$$

which can be rewritten as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)P(A^c)} \quad (9)$$

Both Eqs. (8) and (9) follow from Eq. (7) because of the total probability theorem (Eqs. (2) and (3))

Bayes' theorem can be used to derive the posterior probability of a hypothesis given observed data:

$$P(\text{hypothesis} | \text{data}) = \frac{P(\text{data} | \text{hypothesis})P(\text{hypothesis})}{P(\text{data})} \quad (10)$$

where $P(\text{data} \mid \text{hypothesis})$ is the *likelihood* of the data given the hypothesis ("if the hypothesis is true, then what is the probability of observing these data?"), $P(\text{hypothesis})$ is the *prior probability* of the hypothesis ("what is the *a priori* probability of the hypothesis?"), and $P(\text{data})$ is the probability of observing the data, irrespective of the specified hypothesis. The prior probability (short, *prior*) is also referred to as the (initial) *degree of belief* in the hypothesis. In other words, the prior quantifies the *a priori* plausibility of the hypothesis.

It is often assumed that the data can arise under two competing hypotheses, H_1 and H_2 , with $P(H_1) = 1 - P(H_2)$. Instead of "hypothesis", the term "model" is also frequently used. Let D denote the observed data. Then the posterior probability of the hypothesis (or model) H_1 is

$$P(H_1 \mid D) = \frac{P(D \mid H_1)P(H_1)}{P(D \mid H_1)P(H_1) + P(D \mid H_2)P(H_2)} \quad (11)$$

and the posterior probability of H_2 is

$$P(H_2 \mid D) = \frac{P(D \mid H_2)P(H_2)}{P(D \mid H_1)P(H_1) + P(D \mid H_2)P(H_2)} \quad (12)$$

From Eqs. (11) and (12), we obtain

$$\underbrace{\frac{P(H_1 \mid D)}{P(H_2 \mid D)}}_{\text{posterior odds}} = \underbrace{\frac{P(D \mid H_1)}{P(D \mid H_2)}}_{\text{Bayes factor } B_{12}} \cdot \underbrace{\frac{P(H_1)}{P(H_2)}}_{\text{prior odds}} \quad (13)$$

The *Bayes factor* is the ratio of the posterior odds of H_1 to its prior odds. The Bayes factor can be interpreted as a summary measure of the evidence that the data provide us in favor of the hypothesis H_1 against its competing hypothesis H_2 . If the prior probability of H_1 is the same as that of H_2 (i.e., $P(H_1) = P(H_2) = 0.5$), then the Bayes factor is the same as the posterior odds.

Note that in the simplest case, neither H_1 nor H_2 have any free parameters, and the Bayes factor then corresponds to the likelihood ratio (Kass and Raftery, 1995). If, however, at least one of the hypotheses (or models) has unknown parameters, then the conditional probabilities are obtained by integrating over the entire parameter space of H_i (Kass and Raftery, 1995),

$$P(D \mid H_i) = \int P(D \mid \theta_i, H_i)P(\theta_i \mid H_i)d\theta_i \quad (14)$$

where θ_i denotes the parameters under H_i .

Note that Eq. (13) shows the Bayes factor B_{12} for only two hypotheses, but of course we may consider more than just two. In that case, we can write B_{ij} to denote the Bayes factor for H_i against H_j . When only two hypotheses are considered, they are commonly referred to as *null hypothesis*, H_0 , and *alternative hypothesis*, H_1 . Jeffreys suggests grouping the values of B_{01} into grades (Jeffreys, 1961) (Table 1):

It is instructive to compare the Bayes factor with the p -value from Fisherian significance testing. In short, the p -value is defined as the probability of obtaining a result as extreme as (or more extreme than) the actually observed result, given that the null hypothesis is true. The p -value is generally considered an evidential weight against the null hypothesis: the smaller the p -value, the greater the weight against H_0 . However, the p -value can be a highly misleading measure of evidence because it overstates the evidence against H_0 (Berger and Delampady, 1987; Berrar, 2017; Berrar and Dubitzky, 2017). A Bayesian calibration of p -values is described by Sellke et al. (2001). This calibration leads to the *Bayes factor bound*, $\bar{B} = \frac{1}{-p \log p}$, where p is the p -value. Note that $\bar{B}$ is an upper bound on the Bayes factor over any reasonable choice of the prior distribution of the hypothesis " H_0 is not true", which we may refer to as "alternative hypothesis". For example, a p -value of 0.01 corresponds to an odds of, at most, about 8 to 1 in favor of " H_0 is not true". Note that the concept of "alternative hypothesis" does not exist in the Fisherian significance testing, which considers only one hypothesis, i.e., the null hypothesis H_0 . The idea of "alternative hypothesis" is firmly embedded in the Neyman-Pearsonian hypothesis testing, where the concept of the p -value does not exist. The two different schools of thought – the Fisherian and the Neyman-Pearsonian – should not be conflated; compare (Berrar, 2017).

So far, we have considered only the discrete case, i.e., when the sample space is countable. What if the variables are continuous? Let X and Y denote two continuous random variables with joint probability density function $f_{XY}(x, y)$. Let $f_{X|Y}(x \mid y)$ and $f_{Y|X}(y \mid x)$ denote their conditional probability density functions. Then

$$f_{X|Y}(x \mid y) = \frac{f_{XY}(x, y)}{f_Y(y)} \quad (15)$$

Table 1 Interpretation of Bayes factor B_{01} according to (Jeffreys, 1961)

Grade	B_{01}	Interpretation
0	$B_{01} > 1$	Null hypothesis H_0 supported
1	$1 > B_{01} > 0.32$	Evidence against H_0 , but not worth more than a bare mention
2	$0.32 > B_{01} > 0.10$	Evidence against H_0 substantial
3	$0.10 > B_{01} > 0.032$	Evidence against H_0 strong
4	$0.032 > B_{01} > 0.01$	Evidence against H_0 very strong
5	$0.01 > B_{01}$	Evidence against H_0 decisive

and

$$f_{Y|X}(y|x) = \frac{f_{XY}(x, y)}{f_X(x)} \quad (16)$$

so that Bayes' theorem for continuous variables can be stated as

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{f_Y(y)} \quad (17)$$

where $f_Y(y) = \int_{-\infty}^{+\infty} f_{Y|X}(y|x)f_X(x)dx = \int_{-\infty}^{+\infty} f_{XY}(x, y)dx$ because of the total probability theorem..

In summary, Bayes' theorem provides a logical method that combines new evidence (i.e., new data, new observations) with prior probabilities of hypotheses in order to obtain posterior probabilities for these hypotheses.

Naive Bayes Classifier

We assume that a data set contains n instances (or cases) $\mathbf{x}_i$, $i = 1..n$, which consist of p attributes, i.e., $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$. Each instance is assumed to belong to one (and only one) class $y \in \{y_1, y_2, \dots, y_m\}$. Most predictive models in machine learning generate a numeric score s for each instance $\mathbf{x}_i$. This score quantifies the degree of class membership of that case in class y_j . If the data set contains only positive and negative instances, $y \in \{0, 1\}$, then a predictive model can either be used as a *ranker* or as a *classifier*. The *ranker* uses the scores to order the instances from the most to the least likely to be positive. By setting a threshold t on the ranking score, $s(\mathbf{x})$, such that $\{s(\mathbf{x}) \geq t\} = 1$, the *ranker* becomes a (crisp) classifier (Berrar, 2014).

Naive Bayes learning refers to the construction of a Bayesian probabilistic model that assigns a posterior class probability to an instance: $P(Y = y_j | X = \mathbf{x}_i)$. The simple naive Bayes classifier uses these probabilities to assign an instance to a class. Applying Bayes' theorem (Eq. (7)), and simplifying the notation a little, we obtain

$$P(y_j | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | y_j)P(y_j)}{P(\mathbf{x}_i)} \quad (18)$$

Note that the numerator in Eq. (18) is the joint probability of $\mathbf{x}_i$ and y_j (cf. Eq. (6)). The numerator can therefore be rewritten as follows; here, we will just use $\mathbf{x}$, omitting the index i for simplicity:

$$\begin{aligned} P(\mathbf{x} | y_j)P(y_j) &= P(\mathbf{x}, y_j) = P(x_1, x_2, \dots, x_p, y_j) \\ &= P(x_1 | x_2, x_3, \dots, x_p, y_j) P(x_2, x_3, \dots, x_p, y_j) \text{ because } P(a, b) = P(a | b)P(b) \\ &= P(x_1 | x_2, x_3, \dots, x_p, y_j) P(x_2 | x_3, x_4, \dots, x_p, y_j) P(x_3, x_4, \dots, x_p, y_j) \\ &= P(x_1 | x_2, x_3, \dots, x_p, y_j) P(x_2 | x_3, x_4, \dots, x_p, y_j) \cdots P(x_p | y_j)P(y_j) \end{aligned}$$

Let us assume that the individual x_i are independent from each other. This is a strong assumption, which is clearly violated in most practical applications and is therefore *naive* – hence the name. This assumption implies that $P(x_1 | x_2, x_3, \dots, x_p, y_j) = P(x_1 | y_j)$, for example. Thus, the joint probability of $\mathbf{x}$ and y_j is

$$\begin{aligned} P(\mathbf{x} | y_j)P(y_j) &= P(x_1 | y_j) \cdot P(x_2 | y_j) \cdots P(x_p | y_j)P(y_j) \\ &= \prod_{k=1}^p P(x_k | y_j)P(y_j) \end{aligned} \quad (19)$$

which we can plug into Eq. (18) and we obtain

$$P(y_j | \mathbf{x}) = \frac{\prod_{k=1}^p P(x_k | y_j)P(y_j)}{P(\mathbf{x})} \quad (20)$$

Note that the denominator, $P(\mathbf{x})$, does not depend on the class – for example, it is the same for class y_j and y_l . $P(\mathbf{x})$ acts as a scaling factor and ensures that the posterior probability $P(y_j | \mathbf{x})$ is properly scaled (i.e., a number between 0 and 1). When we are interested in a crisp classification rule, that is, a rule that assigns each instance to exactly one class, then we can simply calculate the value of the numerator for each class and select that class for which this value is maximal. This rule is called the *maximum posterior rule* (Eq. (21)). The resulting “winning” class is also known as the *maximum a posteriori* (MAP) class, and it is calculated as $\hat{y}$ for the instance $\mathbf{x}$ as follows:

$$\hat{y} = \underset{y_j}{\operatorname{argmax}} \prod_{k=1}^p P(x_k | y_j)P(y_j) \quad (21)$$

A model that implements Eq. (21) is called a (simple) *naive Bayes classifier*.

A crisp classification, however, is often not desirable. For example, in ranking tasks involving a positive and a negative class, we are often more interested in how well a model ranks the cases of one class in relation to the cases of the other class (Berrar and Flach, 2012). The estimated class posterior probabilities are natural ranking scores. Applying again the total probability theorem (Eq. (3)), we can rewrite Eq. (20) as

$$P(y_j | \mathbf{x}) = \frac{\prod_{k=1}^p P(x_k | y_j) P(y_j)}{\prod_{k=1}^p P(x_k | y_j) P(y_j) + \prod_{k=1}^p P(x_k | y_j^c) P(y_j^c)} \quad (22)$$

Examples

Application of Bayes' Theorem in Medical Screening

Consider a population of people in which 1% really have a disease, D . A medical screening test is applied to 1000 randomly selected persons from that population. It is known that the sensitivity of the test is 0.90, and the specificity of the test is 0.91.

- If a tested person is really sick, then what is the probability of a positive test result (i.e., the result of the test indicates that the person is sick)?
- If the test is positive, then what is the probability that the person is really sick?

The probability that a randomly selected person has the disease is given as $P(D) = 0.01$ and therefore $P(D^c) = 0.99$. These are the marginal probabilities that are known *a priori*, that is, without any knowledge of the person's test result. The sensitivity of a test is defined as $\frac{TP}{TP+FN}$, where TP denotes the number of true positive predictions and FN denotes the number of false negative predictions. Sensitivity is therefore also known as *true positive rate*; in information retrieval and data mining, it is also called *recall*. The specificity of a test is defined as $\frac{TN}{TN+FP}$, where TN denotes the number of true negative predictions and FP denotes the number of false positive predictions. Let $\oplus$ denote a positive and $\ominus$ a negative test result, respectively. The answer to (a) is therefore simple – in fact, it is already given: the conditional probability $P(\oplus | D)$ is the same as the sensitivity, since the number of persons who are really sick is the same as the number of true positive predictions (persons are sick and they are correctly identified as such by the test) plus the number of false negative predictions (persons are sick but they are not identified as such by the test). Thus, $P(\oplus | D) = \frac{TP}{TP+FN} = 0.9$.

To answer (b), we use Bayes' theorem and obtain

$$P(D | \oplus) = \frac{P(\oplus | D)P(D)}{P(\oplus | D)P(D) + P(\oplus | D^c)P(D^c)} = \frac{0.9 \cdot 0.01}{0.9 \cdot 0.01 + 0.09 \cdot 0.99} = 0.092 \quad (23)$$

The only unknown in Eq. (23) is $P(\oplus | D^c)$, which we can easily derive from the given information: if the specificity is 0.91 or 91%, then the false positive rate must be 0.09 or 9%. But the false positive rate is the same as the conditional probability of a positive result, given the absence of disease, i.e., $P(\oplus | D^c) = 0.09$.

It can be insightful to represent the given information in a confusion matrix (Table 2). Here, the number of true negatives and false positives are rounded to the nearest integer. From the table, we can readily infer the chance of disease given a positive test result as $\frac{9}{9+89}$, i.e., just a bit more than 9%.

The conditional probability $P(D | \oplus)$ is also known as *positive predictive value* in epidemiology or as *precision* in data mining and related fields. What is the implication of this probability being around 0.09? The numbers in this example refer to health statistics for breast cancer screening with mammography (Gigerenzer et al., 2008). A positive predictive value of just over 9% means that only about 1 out of every 10 women with a positive mammogram actually has breast cancer; the remaining 9 persons are falsely alarmed. Gigerenzer et al. showed that many gynecologists do not know the probability that a person has a disease given a positive test result, even when they are given appropriate health statistics framed as conditional probabilities (Gigerenzer et al., 2008). By contrast, if the information is reframed in terms of natural frequencies (as in $\frac{9}{9+89}$ in this example), then the information is often easier to understand.

Naive Bayes Classifier–Introductory Example

We illustrate naive Bayes learning using the contrived data set shown in Table 3. This example is inspired by the famous “Play Tennis” data set, which is often used to illustrate naive Bayes learning in introductory data mining textbooks (Witten and Frank, 2005). The first 14 instances refer to biological samples that belong to either the class *tumor* or the class *normal*. These samples represent the training set. Each instance is described by an expression profile of only four genes. Here, the gene expression values are discretized into either underexpressed (-1), overexpressed ($+1$), or normally expressed (0). Sample #15 represents a new biological sample. What is the likely class of this sample? Note that the particular combination of features, $\mathbf{x}_{15} = (+1, -1, +1, +1)$, does not appear in the training set.

Table 2 Confusion matrix for the example on medical screening

	D	D^c	Σ
$\oplus$	$TP = 9$	$FP = 89$	98
$\ominus$	$FN = 1$	$TN = 901$	902
Σ	10	990	1000

Using Eq. (20), we obtain

$$P(\text{tumor} | \mathbf{x}_{15}) = \frac{P(A = +1 | \text{tumor}) \cdot P(B = -1 | \text{tumor}) \cdot P(C = +1 | \text{tumor}) \cdot P(D = +1 | \text{tumor}) \cdot P(\text{tumor})}{P(\mathbf{x}_{15})}$$

Let's begin with the prior probability of "tumor", $P(\text{tumor})$. This probability can be estimated as the fraction of tumor samples in the data set, i.e., $P(\text{tumor}) = \frac{9}{14}$. What is the fraction of samples for which gene A is overexpressed (+1), given that the class is "tumor"? As an estimate for this conditional probability, $P(\text{Gene A} = +1 | \text{tumor})$, the empirical value of $\frac{2}{9}$ (cf. samples #9 and #11) will be used.

Next, to calculate $P(B = -1 | \text{tumor})$, we proceed as follows: among the nine tumor samples, for how many do we observe $B = -1$? We observe $B = -1$ for cases #5, #7, and #9, so the conditional probability is estimated as $\frac{3}{9}$. The remaining conditional probabilities are derived analogously. Thus, we obtain

$$P(\text{tumor} | \mathbf{x}_{15}) = \frac{\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(\mathbf{x}_{15})} = \frac{0.00529}{P(\mathbf{x}_{15})}$$

$$P(\text{normal} | \mathbf{x}_{15}) = \frac{\frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(\mathbf{x}_{15})} = \frac{0.02057}{P(\mathbf{x}_{15})}$$

With the denominator $P(\mathbf{x}_{15}) = 0.00529 + 0.02057$, we then obtain the properly scaled probabilities $P(\text{tumor} | \mathbf{x}_{15}) = 0.02046$ and $P(\text{normal} | \mathbf{x}_{15}) = 0.79554$.

Laplace Smoothing

When the number of samples is small, a problem may arise over how to correctly estimate the probability of an attribute given the class. Let us assume that at least one attribute value of the test instance, $\mathbf{x}$, is absent in all training instances of a class γ_i . For example, assume that Gene A of instance #9 and #11 in Table 3 is underexpressed (-1) instead of overexpressed (+1). Then we obtain the following conditional probabilities,

$$\begin{aligned} P(\text{Gene A} = +1 | \text{tumor}) &= \frac{0}{9} \\ P(\text{Gene A} = 0 | \text{tumor}) &= \frac{4}{9} \\ P(\text{Gene A} = -1 | \text{tumor}) &= \frac{5}{9} \end{aligned}$$

which obviously leads to $P(\text{tumor} | \mathbf{x}_{15}) = 0$. If Gene A is underexpressed (-1) in instances #9 and #11 in Table 3, then $P(\text{Gene A} = +1 | \text{tumor}) = 0$, which implies that it is impossible to observe an overexpressed Gene A in a sample of class "tumor". Is it wise to make such a strong assumption? Probably not. It might be better to allow for a small, non-zero probability. This is what *Laplace smoothing* does (Witten and Frank, 2005). In this example, we simply add 1 to each of the three numerators above and then add 3 to each of the denominators:

$$\begin{aligned} P(\text{Gene A} = +1 | \text{tumor}) &= \frac{0+1}{9+3} \\ P(\text{Gene A} = 0 | \text{tumor}) &= \frac{4+1}{9+3} \\ P(\text{Gene A} = -1 | \text{tumor}) &= \frac{5+1}{9+3} \end{aligned}$$

Table 3 Contrived gene expression data set of 15 biological samples, each described by the discrete expression level of 4 genes. A sample belongs either to class "normal" or "tumor". Instance #15 is a new, unclassified sample

Sample	Gene A	Gene B	Gene C	Gene D	Class
1	+1	+1	+1	0	normal
2	+1	+1	+1	+1	normal
3	0	+1	+1	0	tumor
4	-1	0	+1	0	tumor
5	-1	-1	0	0	tumor
6	-1	-1	0	+1	normal
7	0	-1	0	+1	tumor
8	+1	0	+1	0	normal
9	+1	-1	0	0	tumor
10	-1	0	0	0	tumor
11	+1	0	0	+1	tumor
12	0	0	+1	+1	tumor
13	0	+1	0	0	tumor
14	-1	0	+1	+1	normal
15	+1	-1	+1	+1	unknown

However, instead of adding 1, we could also add a small positive constant c weighted by p_i ,

$$\begin{aligned} P(\text{Gene A} = +1 | \text{tumor}) &= \frac{0 + cp_1}{9 + c} \\ P(\text{Gene A} = 0 | \text{tumor}) &= \frac{4 + cp_2}{9 + c} \\ P(\text{Gene A} = -1 | \text{tumor}) &= \frac{5 + cp_3}{9 + c} \end{aligned}$$

with $p_1 + p_2 + p_3 = 1$, which are the prior probabilities for the states of expression for Gene A. Although such a fully Bayesian specification is possible, in practice, it is often unclear how the priors should be estimated, and simple Laplace smoothing is often appropriate (Witten and Frank, 2005).

Mixed Variables

In contrast to many other supervised learning algorithms, the naive Bayes classifier can easily cope with mixed-variable data sets. For example, consider Table 4. Here, Gene B has numeric expression values.

Assuming that the expression values of Gene B follow a normal distribution, we can model the probability density given class y_i as

$$f(x | y_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2} \quad (24)$$

where μ_i and σ_i denote the mean and standard deviation of the gene expression value for class y_i , respectively. Of course, in practice, other distributions are possible, and we need to choose the distributional model that best describes the data. In the example, we obtain $\mu_{\text{tumor}} = 21.9$, $\sigma_{\text{tumor}} = 7.7$, and $\mu_{\text{normal}} = 24.2$, $\sigma_{\text{normal}} = 8.5$. Note that the probability that a continuous random variable X takes on a particular value x is always zero for any continuous probability distribution, i.e., $P(X = x) = 0$. However, using the probability density function, we can calculate the probability that X lies in a narrow interval $[x_0 - \frac{\epsilon}{2}, x_0 + \frac{\epsilon}{2}]$ around x_0 as $\epsilon \cdot f(X = x_0)$. For the new instance $\mathbf{x}_{15}$ (Table 4), we obtain $f(12 | \text{tumor}) = 0.02267$ and $f(12 | \text{normal}) = 0.01676$, so that we can state the conditional probabilities as

$$\begin{aligned} P(\text{tumor} | \mathbf{x}_{15}) &= \frac{\frac{2}{9} \cdot 0.0227\epsilon \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(\mathbf{x}_{15})} = \frac{0.00036\epsilon}{P(\mathbf{x}_{15})} \quad \text{and} \\ P(\text{normal} | \mathbf{x}_{15}) &= \frac{\frac{3}{5} \cdot 0.01676\epsilon \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(\mathbf{x}_{15})} = \frac{0.00172\epsilon}{P(\mathbf{x}_{15})} \end{aligned}$$

The posterior probabilities are $P(\text{tumor} | \mathbf{x}_{15}) = \frac{0.00036\epsilon}{0.00036\epsilon + 0.00172\epsilon} = 0.17$ and $P(\text{normal} | \mathbf{x}_{15}) = \frac{0.00172\epsilon}{0.00036\epsilon + 0.00172\epsilon} = 0.83$. Note that ϵ cancels.

Missing Value Imputation

Missing values do not present any problem for the naive Bayes classifier. Let us assume that the new instance contains missing values (encoded as NA), for example, $\mathbf{x}_{15} = (+1, \text{NA}, +1, +1)$. The posterior probability for class y_i can then be calculated by simply omitting this attribute, i.e.,

$$\begin{aligned} P(\text{tumor} | \mathbf{x}_{15}) &= \frac{\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(\mathbf{x}_{15})} = \frac{0.016}{P(\mathbf{x}_{15})} \quad \text{and} \\ P(\text{normal} | \mathbf{x}_{15}) &= \frac{\frac{3}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(\mathbf{x}_{15})} = \frac{0.103}{P(\mathbf{x}_{15})} \end{aligned}$$

Table 4 Contrived gene expression data set from Table 3. Here, absolute expression values are reported for Gene B

Sample	Gene A	Gene B	Gene C	Gene D	Class
1	+1	35	+1	0	normal
2	+1	30	+1	+1	normal
3	0	32	+1	0	tumor
4	-1	20	+1	0	tumor
5	-1	15	0	0	tumor
6	-1	13	0	+1	normal
7	0	11	0	+1	tumor
8	+1	22	+1	0	normal
9	+1	14	0	0	tumor
10	-1	24	0	0	tumor
11	+1	23	0	+1	tumor
12	0	25	+1	+1	tumor
13	0	33	0	0	tumor
14	-1	21	+1	+1	normal
15	+1	12	+1	+1	unknown

If the training set has missing values, then the conditional probabilities can be calculated by omitting these values. For example, suppose that the value +1 is missing for Gene A in sample #1 (Table 4). What is the probability that Gene A is overexpressed (+1), given that the sample is normal? There are five normal samples, and two of them (#2 and #8, Table 4) have an overexpressed Gene A. Therefore, the conditional probability is calculated as $P(\text{Gene A} = +1 | \text{normal}) = \frac{2}{5}$.

R Implementation

We will now illustrate how to build a naive Bayes classifier using the function `naiveBayes()` of the package `e1073` (Meyer *et al.*, 2015) in the programming language and environment R (R Core Team, 2017), which is widely used by the bioinformatics community. Here, we use the data from Table 3.

```
library(e1071)

#' Load the data.
train <- read.csv("NB_train.csv", colClasses = c('factor', 'factor', 'factor', 'factor'))
test  <- read.csv("NB_test.csv", colClasses = c('factor', 'factor', 'factor', 'factor'))

# Build the model.
NB_model <- naiveBayes(Class ~ ., data = train)

# Predict the test case.
pred <- predict(NB_model, test, type = "raw")

pred

##           normal          tumor
## [1,] 0.000202459 0.9997975
```

Surprisingly, these probabilities differ from what we calculated above, namely $P(\text{tumor} | \mathbf{x}_{15}) = 0.2046$ and $P(\text{normal} | \mathbf{x}_{15}) = 0.7954$. Why? The problem can be quite hard to spot. The reason is that the factor levels (per attribute) are not the same in the training and the test set, which causes the function `predict()` to calculate incorrect probabilities. Internally, `predict()` converts attribute values into numbers, and it does not check whether the factor levels are consistent or not.

```
str(train)

## 'data.frame': 14 obs. of 5 variables:
## $ Gene_A: Factor w/ 3 levels "0","-1","+1": 3 3 1 2 2 2 1 3 3 2 ...
## $ Gene_B: Factor w/ 3 levels "0","-1","+1": 3 3 3 1 2 2 2 1 2 1 ...
## $ Gene_C: Factor w/ 2 levels "0","+1": 2 2 2 2 1 1 1 2 1 1 ...
## $ Gene_D: Factor w/ 2 levels "0","+1": 1 2 1 1 1 2 2 1 1 1 ...
## $ Class : Factor w/ 2 levels "normal","tumor": 1 1 2 2 2 1 2 1 2 2 ...

str(test)

## 'data.frame': 1 obs. of 5 variables:
## $ Gene_A: Factor w/ 1 level "+1": 1
## $ Gene_B: Factor w/ 1 level "-1": 1
## $ Gene_C: Factor w/ 1 level "+1": 1
## $ Gene_D: Factor w/ 1 level "+1": 1
## $ Class : Factor w/ 1 level "unknown": 1
```

As we can see, the factor levels are not the same in the training and test set. The user has to ensure factor level consistency. A simple solution consists in first appending the test case to the training set and then splitting them apart. Note that the class labels also have to be consistent. At the moment, the test case has the class label "unknown", but this is not a valid label. When we add the test case to the training set, we erroneously increase the factor level of "Class", which in turn will cause `naiveBayes()` to assume that there are three classes in total.

```
train_save <- train
test_save <- test
X <- rbind(train_save, test_save)
train <- X[1:14, ]
test <- X[15, ]

# Build the model again and apply it to the test case.
NB_model <- naiveBayes(Class ~ ., data = train)
pred <- predict(NB_model, test, type = "raw")
pred

##          normal tumor unknown
## [1,]      NaN   NaN   NaN

# We have to make sure that our test case has a class label that also appears in
# the training set, so we can just choose either "normal" or "tumor".
# It does not matter which one we choose, and it has no influence
# on predict() because predict() uses only the predictor variables
test$Class <- "normal"
X <- rbind(train_save, test)
train <- X[1:14, ]
test <- X[15, ]

# Build the model again and apply it to the test case.
NB_model <- naiveBayes(Class ~ ., data = train)
pred <- predict(NB_model, test, type = "raw")
pred

##          normal      tumor
## [1,] 0.7954173 0.2045827
```

When we use `naiveBayes()`, we have to make sure that the factor levels in the training and test set are consistent. Also, we need to make sure that data types are correct. Note that the values in [Table 3](#) could be interpreted as integers, which would of course lead to different results (see the R code at <https://osf.io/gtchm/> for more details). Both pitfalls can be easily overlooked and thereby cause `naiveBayes()` and `predict()` to produce results that may look plausible but that are, in fact, incorrect.

Discussion

In this article, we derived Bayes' theorem from the fundamental concepts of probability. We then presented one member of the family of machine learning methods that are based on this theorem, the naive Bayes classifier, which is one of the oldest workhorses of machine learning.

It is well known that the misclassification error rate is minimized if each instance is classified as a member of that class for which its conditional class posterior probability is maximal ([Domingos and Pazzani, 1997](#)). Consequently, the naive Bayes

classifier is optimal (cf. Eq. (21)), in the sense that no other classifier is expected to achieve a smaller misclassification error rate, provided that the features are independent. However, this assumption is a rather strong one; clearly, in the vast majority of real-world classification problems, this assumption is violated. This is particularly true for genomic data sets with many co-expressed genes. Perhaps surprisingly, however, the naive Bayes classifier has demonstrated excellent performance even when the data set attributes are not independent (Domingos and Pazzani, 1997; Zaidi *et al.*, 2013).

Another advantage of the naive Bayes classifier is that the calculation of the conditional probabilities is highly parallelizable and amenable to distributed processing, for example, in a MapReduce environment (Villa and Rossetti, 2014). Thus, the naive Bayes classifier is also interesting for big data analytics.

The performance of the naive Bayes classifier can often be improved by eliminating highly correlated features. For example, assume that we add ten additional genes to the data set shown in Table 4, where each gene is described by expression values that are highly correlated to those of Gene B. This means that the estimated conditional probabilities will be dominated by those values, which would “swamp out” the information contained in the remaining genes.

We illustrated some caveats and pitfalls and how to avoid them when building a naive Bayes classifier in the programming language and environment R (R Core Team, 2017). Further details with fully commented code and example data are available at the accompanying website <http://osf.io/92mes>.

Closing Remarks

Harold Jeffreys, a pioneer of modern statistics, succinctly stated the importance of Bayes' theorem: “[Bayes' theorem] is to the theory of probability what Pythagoras' theorem is to geometry.” (Jeffreys, 1973, p. 31). Indeed, Bayes' theorem is of fundamental importance not only for inferential statistics, but also for machine learning, as it underpins the naive Bayes classifier. This classifier has demonstrated excellent performance compared to more sophisticated models in a range of applications, including tumor classification based on gene expression profiling (Dudoit *et al.*, 2002). The naive Bayes classifier performs remarkably well even when the underlying independence assumption is violated.

See also: Data Mining in Bioinformatics. The Challenge of Privacy in the Cloud

References

- Berger, J., Delampady, M., 1987. Testing precise hypotheses. *Statistical Science* 2 (3), 317–352.
- Berrar, D., 2014. An empirical evaluation of ranking measures with respect to robustness to noise. *Journal of Artificial Intelligence Research* 49, 241–267.
- Berrar, D., 2017. Confidence curves: An alternative to null hypothesis significance testing for the comparison of classifiers. *Machine Learning* 106 (6), 911–949.
- Berrar, D., Dubitzky, W., 2017. On the Jeffreys-Lindley Paradox and the looming reproducibility crisis in machine learning, in: *Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics*, Tokyo, Japan, 19–21 October 2017, pp. 334–340.
- Berrar, D., Flach, P., 2012. Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them). *Briefings in Bioinformatics* 13 (1), 83–97.
- Berry, D., 1996. *Statistics – A Bayesian Perspective*. Duxbury Press.
- R Core Team, 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, Available at: <https://www.R-project.org/>.
- Domingos, P., Pazzani, M., 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29 (2), 103–130.
- Dudoit, S., Fridlyand, J., Speed, T., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* 97 (457), 77–87.
- Gigerenzer, G., Gaissmaier, W., Kurz-Milcke, E., Schwartz, L., Woloshin, S., 2008. Helping doctors and patients to make sense of health statistics. *Psychological Science in the Public Interest* 8 (2), 53–96.
- Jeffreys, H., 1961. *Theory of Probability*, third ed. Oxford: Clarendon Press, [Reprinted 2003, Appendix B].
- Jeffreys, H., 1973. *Scientific Inference*, third ed. Cambridge University Press.
- Kass, R., Raftery, A., 1995. Bayes factors. *Journal of the American Statistical Association* 90 (430), 773–795.
- Lewis, D.D., 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (Eds.), *Machine Learning: ECML-98*. In: *Proceedings of the 10th European Conference on Machine Learning*, Chemnitz, Germany, April 21–23, Springer, Berlin/Heidelberg, pp. 4–15.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., 2015. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien, R package version 1.6-7. Available at: <https://CRAN.R-project.org/package=e1071>.
- Sellke, T., Bayarri, M., Berger, J., 2001. Calibration of *p* values for testing precise null hypotheses. *The American Statistician* 55 (1), 62–71.
- Villa, S., Rossetti, M., 2014. Learning continuous time Bayesian network classifiers using MapReduce. *Journal of Statistical Software* 62 (3), 1–25.
- Witten, I., Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, second ed. Morgan Kaufmann.
- Zaidi, N.A., Cerquides, J., Carman, M.J., Webb, G.I., 2013. Alleviating naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research* 14, 1947–1988.

Relevant Website

<http://osf.io/92mes>
Open Science Framework.

Data Mining: Prediction Methods

Alfonso Urso, Antonino Fiannaca, Massimo La Rosa, and Valentina Ravi, Via Ugo La Malfa, Palermo, Italy
Riccardo Rizzo, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

k-Nearest Neighbor (*k*-NN)

A different approach to classification or regression is possible without any model construction first. This is known as the *lazy approach* because of the working procedure: the learner does not construct any general model before seeing the test instance. The learner stores all training instances and, only when a given test instance arrives, it generalizes the information coming from the examples to classify or predict the new class label or numeric value. *k*-nearest neighbor (*k*-NN) and *case-based reasoning* are examples of lazy learner.

k-NN method is based on the nearest neighbor decision rule (Fix and Hodges, 1951, 1952; Cover and Hart, 1967), according to which the class of an object never seen before by the classifier depends on a set of previously classified records. The nearest neighbor rule is widely applied in problems of pattern recognition, text categorization, ranking models, object recognition and event recognition (Bhatia, 2010).

How it Works

k-NN compares an unknown instance with those belonging to the training set and assigns it the class according to the similarity with training instances. Consider a training set of instances as pairs (X_i, y_i) , where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ is a tuple (i.e., a vector in the feature space representing a sequence of values concerning respectively fixed attributes) described by n attributes and y_i is the corresponding class label (qualitative or quantitative) (See the paragraphs 1 and 2 of the chapter "Data Mining: Classification and Prediction".); $i = 1, \dots, p$ is the total number of tuples in the data set. A tuple is represented by a point in a n -dimensional space of attributes (or feature space (The feature space concept was introduced in the chapter "Data Mining: Classification and Prediction", section 'Components of a Generalized Linear Model')). A new instance to be classified will take one among the class label of k training tuples that are nearest neighbors in the space of representation (see Fig. 1). By using a distance metric, the classifier establishes the neighbors in the feature space.

It is possible to use several *distances functions* to compute the similarities among tuples. This choice depends on the kind of the data set, if it includes numeric, symbolic or categorical attributes.

Distance functions

For numeric attributes, common distances are *Manhattan* and *Euclidean* distances, which represent two particular cases of the more general *Minkowski* distance.

Manhattan distance d_M between the tuples X_1 and X_2 is known also as Minkowski distance of order 1 and is given by Eq. (1)

$$d_M(X_1, X_2) = \sum_{i=1}^n |x_{1i} - x_{2i}| \quad (1)$$

The Euclidean distance d_E between two tuples X_1 and X_2 is computed as shown in Eq. (2):

$$d_E(X_1, X_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (2)$$

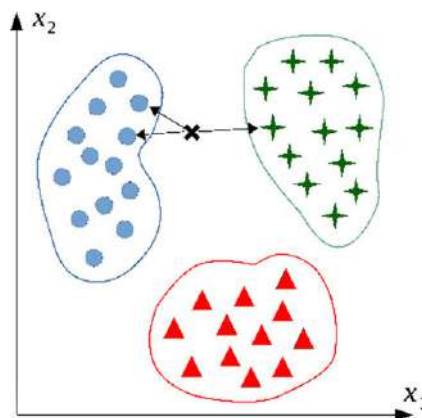


Fig. 1 *k*-NN classification in a 2-dimensional feature space; $k=3$. The black cross represents a new instance to classify. The arrows mark the distance among the new tuple and 3 nearest neighbors.

It is known also as Minkowski distance of order 2.

Hamming distance is used to compare two tuples expressed by strings of n characters (symbolic attributes). In this case, the distance reports the total number of correspondent positions where the symbols are different. For example, considering the strings "ATCG" and "ATGG", its Hamming distance is $d_M=1$, differing only in the third position.

The computation of their distance for categorical attributes is not straightforward. The distance or, conversely, the similarity can be evaluated considering schemes of numeric "translation". Starting from the trivial case, the similarity between two categorical attributes is 1 if they are identical or 0 if they are different. Considering instances described by more than one categorical attribute, the similarity between them will increase with the number of matching attributes. This is the so-called *overlap* measure (Stanfill and Waltz, 1986). Additional sophisticated procedures allow evaluating differential grading of similarities between instances described by more than one categorical attribute (Boriah et al., 2008). Other distances can be used, for example, *Jaccard* distance (Levandowsky and Winter, 1971), *Dice's* coefficient (Dice, 1945), *Tanimoto* coefficient (Tanimoto, 1958) and *cosine* distance (Qian et al., 2004).

The k value

Once calculated the distances from the other training examples, k -NN classifier establishes the class of the new instance by choosing the most frequent label among k nearest neighbors (see Fig. 2). In prediction task, for a new instance in input, k -NN predictor returns a real value, which is the average value of k numeric labels of nearest neighbors.

The choice of the k value influences the performance of the k -NN classifier. Indeed, a too small value of k makes the classifier more sensitive to noising data. Conversely, if k value is too large, the classifier will consider also examples from other classes, possibly deviating the result of classification. In the limit case, all the training instances will be considered and the most frequent class label will be assigned to the new instance.

The good value for the number k of neighbors is experimentally determined by estimating the error rate of the classifier/predictor with a start number $k^{(0)}$. If the performance needs to be improved, the number $k^{(0)}$ is increased, according to the number of new neighbors. The error rate can be estimated through probabilistic methods (Duda et al., 1991).

k -NN Regression

For dataset of instances (X_j, y_j) described by numeric attributes as component of the vector X_j and corresponding numeric value y_j , k -NN works as a regression method. Therefore, it allows to predict the numeric value $\hat{y}$ corresponding to an unseen instance vector X , on the basis of the information of k nearest neighbors in the training set. Consider the simple case of dataset described by a single feature. Hence, the j th instance of the dataset is represented by the pair (x_j, y_j) , where x_j is a scalar. The entire dataset can be plotted in a 2-dimensional graph, where the x -axis represent the 1-dimensional feature space (see Fig. 3).

Given a new instance $(x=4; \hat{y}=?)$, k -NN provides as output a number $\hat{y}$ arising from Eq. (3) (Hastie et al., 2008):

$$\hat{y} = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \quad (3)$$

where $N_k(x)$ is the set $\{x_1, \dots, x_k\}$ of the training neighbors of the new value x . Clearly, $N_k(x)$ depends on the value chosen for k . Referring to the example depicted in Fig. 3, for $k=1$, $N_k(x)$ will contain only one instance, which is the first in the list of the nearest neighbors. Consequently $\hat{y}=y_1=6$. For $k=2$, $\hat{y}=\frac{1}{2}(6+3)=4.5$ and so on. As shown in the Fig. 3, the suitable choice of k dictates the performance (with $k=3$ the predicted value moves away from the true value).

Unlike the regression methods described below, k -NN regression is said a "non-parametric" method, since it does not assume any explicit form for the prediction function f , meaning that it does not construct a model for f .

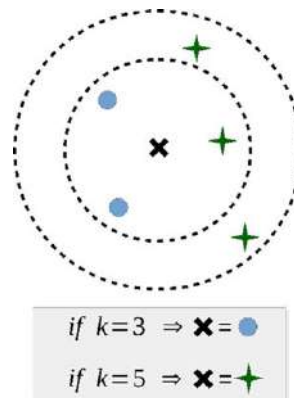


Fig. 2 The choice of the k value conditions the output of the classifier, including more or less example of different class.

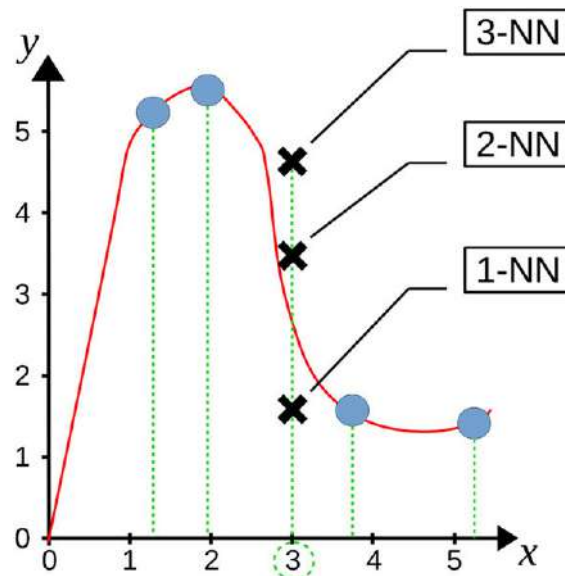


Fig. 3 k -NN regression: blue dots represent the training set plotted in a graph of y_i values against the feature values x_i . The red curve f is the true function underlying the training set. The regression method approximates the true (unknown) function. Therefore, given the new instance $x=3$, it provides as output a numeric value $\hat{y}$ (one black cross for each value of k) that should be as near as possible to the true value $f(x)$. $\hat{y}$ value changes for different values of k .

Conclusive Remarks

As resumed in the work of [Bhatia \(2010\)](#), k -NN technique includes advantages like a very fast training and a simple learning. It shows robustness to noisy training data coming from pruning noisy tuples ([Aha, 1992](#)). On the other hand, the main disadvantages concern mainly the computational complexity ([Preparata and Shamos, 1985](#)) and memory limitation of this technique. The slowness of classification can be improved by using a partial distance calculation, where only a subset of attribute is involved in the compute of the distance ([Gersho and Gray, 1992](#)). The editing of the stored tuples removes useless training examples, helping to speed up the process of classification ([Hart, 1968](#)). Furthermore, the performance of k -NN algorithm can be biased by the choice of the k value ([Guo et al., 2003](#)). Irrelevant attribute can easily mislead the decision of k -NN. For this reason, it is made more robust by a specific pruning on data tuples.

Case-Based Reasoning (CBR) Classifier

Case-based reasoning (CBR) ([Aamodt and Plaza, 1994](#); [Watson and Marir, 1994](#)), was born in response to several critical problems met by rule-based expert systems (RBES), in Artificial Intelligence field ([Schank, 1984](#)). CBR solves new problems according to a database of solved cases, namely of problem situations previously analyzed and stored. It works by comparing the characteristics of the new problem with the cases in its “historical archive” and looks for a match with stored solutions, eventually by adapting those that are similar to the examined issue ([Schank, 1982](#); [Riesbeck and Schank, 1989](#); [Kolodner, 1993](#)).

Classification is one of the possible applications of CBR in business applications, engineering and law area ([Watson and Marir, 1994](#); [Allen, 1994](#); [Rissland and Ashley, 1987](#)). In particular, CBR classifier for medical applications employs the data concerning patients’s cases and their treatments to assist in the diagnosis. First examples of applications are CASEY ([Koton, 1988](#)) heart diagnosis generator and PROTOS ([Bareiss et al., 1988](#)) for clinical audiology. There are also current applications in breast cancer domain ([Gu et al., 2017](#)).

How it Works

A crucial component of CBR is its database of information and knowledge developed with the experience that arises from previously examined situations. CBR learns from such past cases and stores them conveniently to create a *general background knowledge* database ([Aamodt and Plaza, 1994](#)). They represent the training instances of CBR, and they are employed to solve a new problem, called *new case*.

The solutions stored in the knowledge-base or *case-base* can be directly applied to the new problem or eventually adapted. Indeed, CBR can modify a retrieved solution for a new, different problem. Furthermore, it is able to enrich its case-base with each new solved case. Each solved case can be kept as single experience or grouped with others similar in order to form a generalized case. The problem-solving approach of CBR can be resumed in the following a cycle schema.

CBR cycle

CBR follows a cyclic process to solve a problem, learning from experience and reusing it to treat a new case.

A CBR cycle can be described by the four steps:

1. RETRIEVAL
2. REUSE
3. REVISION
4. RETAINMENT

First, CBR searches for a possible match of the new case with training instances and returns the solution eventually found (*retrieval*). Such solution will be reused for the new examined case (*reuse*). If the perfect match is not retrieved, CBR looks for neighbors, which are stored instances similar for a subset of attributes to the new case. Then, it proposes an answer by combining the solutions of neighbors (*revision*). Finally, CBR includes the new solved case in the case-base, incrementing its experience (*retainment*). The Fig. 4 depicts the schema of operation of a case-based reasoning. The following sub-sections shortly describe the steps of a CBR cycle.

Retrieval

The retrieval of cases consists of a set of subtasks such as:

- identification of features
- matching
- selection.

The identification of features is focused to detect the relevant characteristic of the new problem, to treat it comparing with the case-base. The matching task searches for a solution for the new problem recurring to the case-base. If any exact match is found, it provides a set of stored cases similar for features to the new instance, referring to a given similarity threshold. Selection step chooses the best match among the similar cases.

Reuse

The reuse procedure involves two fundamental steps: first, identifying the differences between the two cases and then establishing what part of the retrieved case can be used in the new case.

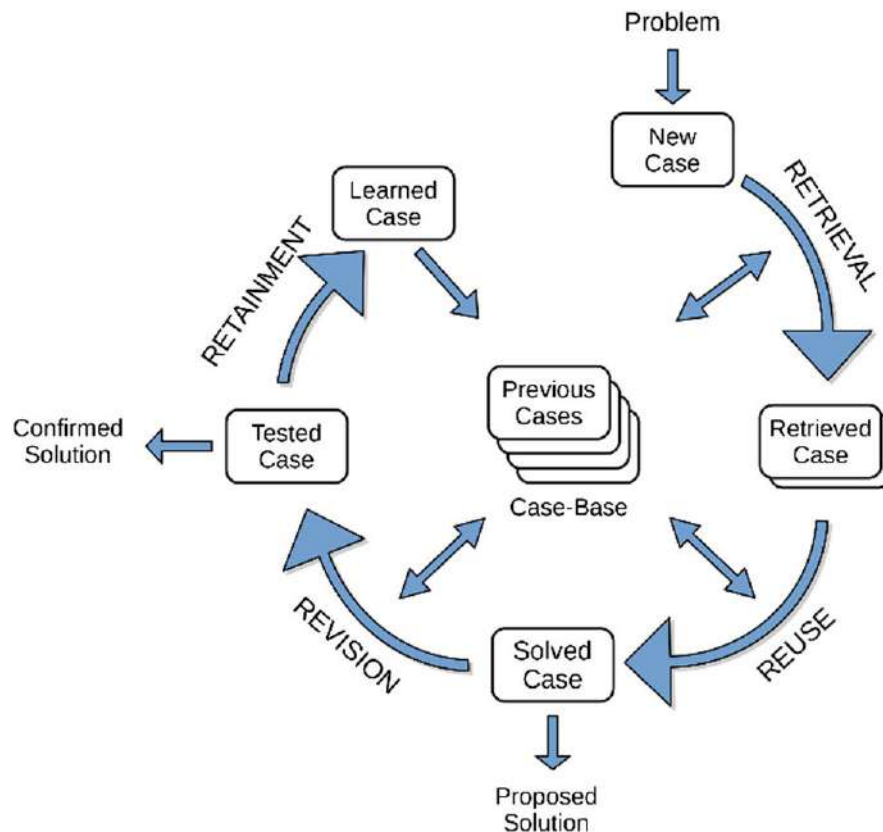


Fig. 4 Cycle of CBR.

Revision

In case revision step, CBR follows two procedures: first, it evaluates the solution provided from reuse step by applying that in the environment of the new problem; if the reuse solution is successful CBR retain it, otherwise CBR repair the solution, detecting errors and elaborating a new solution. In the repair step, the system recurs to a specific domain knowledge.

Retention

The retention step corresponds to the learning for CBR. It is focused on the ability to select the information to retain if a new solution is proposed; it is also important to establish how to encode the retained information, and consequently the indexing of the new entry in case-base for the future retrieval.

Operating Details

The CBR performance is conditioned by the content of its case-base and by its structure. Indeed, the effectiveness and the efficiency of CBR strongly depend on the search case and matching process. Hence, the structure of case-base, also called *case memory*, should be suitably organized and indexed, as a vocabulary to favor the retrieval and reuse. Consequently, the representation of cases is crucial for CBR, to describe the case content appropriately, by involving their features. A further challenge consists into integrating case-base structure to create a general domain knowledge model.

Two examples of case memory models are “Dynamic memory model” (Schank, 1982) on which it is based the first case-based system called CYRUS (Kolodner, 1983) and “Category & exemplar Model” (Porter and Bareiss, 1986), described by Aamodt and Plaza (1994).

Conclusive Remarks

CBR classifier performance depends on the choice of the similarity metric, which is involved in the identification of neighbors, and on the choice of suitable methods to combine solution. It is fundamental select remarkable features for indexing training cases.

Incrementing the number of the stored cases, the reasoner increases its abilities and accuracy. Conversely, beyond a given limit, the rising of computational time to find relevant cases determines a reduction of efficiency. In this case, an expedient is editing the database, discarding those cases redundant or useless.

Genetic Algorithms Classifier

Genetic algorithms (GAs) (Holland, 1975) are stochastic algorithms used to solve optimization and search problems, (see Michalewicz, 1992). They are a type of evolutionary computation techniques (Rechenberg, 1973) inspired by adaptation principles of natural selection. They are domain independent methods, therefore, GAs are applied in several fields of computer science, for example as optimization process in a number of problems such as routing and scheduling (the traveling salesman problem), game-playing, cognitive modeling, transportation problem and control problems (Janikow, 1993; Goldberg, 1989; DeJong, 1985) and also in the field of Neural Networks (Royas, 1996).

Vocabulary

GAs works on a set of possible solutions to a given problem, by applying stochastic methods that simulate natural ways of evolution. Biological world influences both computing methods and language of GAs. In biological language, the genetic information of each individual is contained in *chromosomes* (see Fig. 5, left side). A chromosome consists of units or *genes* that control hereditariness of characters. Each gene occupies a given position or *locus* on the chromosome. The same gene can be in several states or *alleles*, entailing the different ways in which an individual's character (for example eyes color) manifests itself.

In GAs, the chromosomes represent candidate solutions to a given problem. A chromosome is made up of a sequence of genes (see Fig. 5, right side). The genes encode a particular characteristic or feature of the candidate solution. For example, considering a problem of optimization, the optimization function is a chromosome. Each parameter of the optimization function is a gene, encoded by either a single bit or a short block of bits. A gene represented by single bit can be in two states, either “0” or “1”. These two possibilities are the alleles of that gene. The different alleles of the same gene represent different values of the same feature in GA vocabulary. For a gene that encodes more than two possible alleles, a block of bits is used.

How it Works

The basic idea of GA is to find the best individual in a population of chromosomes, namely candidates solutions of a given task (optimization function or classification rule), following a process of natural selection. During such process, the fitter individuals of the population are chosen to create the *offspring*. The offspring will result in a new successive population replacing the previous

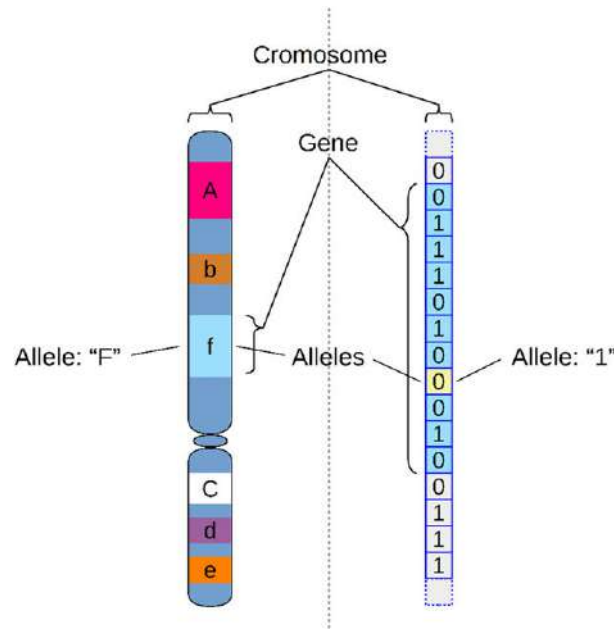


Fig. 5 (left) Biological representation of a chromosome: a string of genes (bands with letters), which are a portion of DNA. Each portion occupies a given locus on the chromosome. Uppercase or lowercase letter in a locus represents the allele of the same gene; (right) Computational representation of a chromosome: a sequence of genes, which are codified by one or more bits. The possibilities “0” or “1” at the same locus give rise to alleles of the same gene.

one. Each replacement gives rise to a new *generation*, potentially including the fittest one among several individuals and hence the best solution to the problem.

GAs employ the way of natural selection to allow proliferation and modification of good solutions and inhibit bad solutions, in perfect analogy with the biological environment (Chang and Lippmann, 1991).

The implementation of GA is based on its *genetic operators*. The simplest form of GA employs *creation*, *selection*, *crossover* and *mutation*. Genetic operators will be discussed in the following section. Furthermore, an *evaluate function* (Michalewicz, 1992) or *fitness function* (Mitchell, 1996) allows to evaluate each individual by assigning it a score, namely the *fitness*. The fitness of an individual depends on its goodness as a solution for the considered problem. The candidate who approaches the best solution will have the highest score. The following section introduces an example of the fitness function and a simple scheme of GA with genetic operators.

Fitness function

In the optimization problem, the aim is to find the set of parameters that maximize or minimize a given function. Consider for example the function of Eq. (4) (Riolo, 1992):

$$f(x) = x + |\sin(32x)|, \quad \text{with } 0 \leq x \leq \pi \quad (4)$$

The value of x that maximize the function are the candidate solutions. The possible values of x are encoded in strings of bit: they represent the chromosomes on which will operate the GA. To evaluate a candidate bit string, it is enough to convert it in the correspondent real number x' and calculate the value of the function in x' . This value is the fitness of the string.

Operators of GA: Creation, selection, crossover, mutation

The evolution process in GAs starts with the *creation* of the initial population by the specific homonym genetic operator. The individuals in a population are chromosomes encoded by the bit string that is initialized to “1” or “0” values.

Once the first population is created, the fitness of individuals is evaluated. Then, the *selection* operator identifies the fittest candidates to breed. The probability of selection is indeed an increasing function of fitness (Mitchell, 1996). The same chromosome can be selected more times to reproduce, if it is fitter than others.

A *crossover* operator acts on a couple of selected chromosomes, the *parents*, exchanging portions of these. In Fig. 6, it is shown the simplest crossover operator, which acts in a single locus of the chromosome (single point). First, the crossover operator randomly chooses a locus on the parents’ chromosomes. Then it exchanges the substrings, creating two offspring. The aim of this process is to mix the useful parts of both parents to produce the new better chromosomes.

Crossover can provide new chromosomes until that the individuals are not too similar to each other. For this scope, the new individual can be altered by the operator *mutation* that randomly selects bits in a string and then inverts them, as shown in Fig. 6.

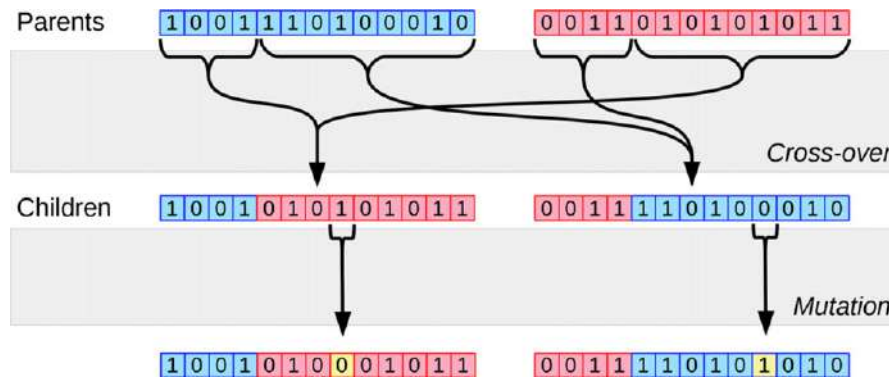


Fig. 6 The parents “blue” and “pink” strings breed through the crossover operator. It creates two new chromosomes children (offspring) from the single crossover point at the fourth-bit position. The mutation operator changes the offspring.

Table 1 Pseudocode summarizing the steps of a generic GA

1. Creation of a random population of n chromosomes (initialization of bit strings);
2. Evaluation of the fitness of each chromosome;
3. Intermediate step (to repeat until creating n offspring):
 - a) Selection of two chromosomes as parents (based on step 2)
 - b) Crossover with probability p_c
 - b) Mutation with probability p_m
4. Replacement of the current population with new population;
5. Go to step 2.

Crossover and mutation are random operators, meaning that they will act with a fixed probability, respectively p_c (crossover probability or crossover rate) and p_m (mutation probability or mutation rate).

A simple scheme of operation of GA is illustrated below.

Scheme of a simple GA

The process begins with the creation operator, which produces an initial population of chromosomes. The intermediate step provides a recursive modification of population, by applying other genetic operators such as selection, crossover and mutation. At the end of the evolution process, only the fittest offspring will survive, representing the selection of the best solutions.

The following scheme shown in [Table 1](#) summarizes the steps of a GA observing what has been reported by [Mitchell \(1996\)](#):

The iterations of this procedure are called *generations*. Their number varies typically from 50 to 500 or more ([Mitchell, 1996](#)). The set of generations is called *run*. After a run, the fitness function will indicate one or more individuals highly fit in the population. The performance of the GA often depends on the characteristic chosen to treat a given problem, for example, the size n of population and p_c and p_m value. Considering the importance of randomness in GA, a better information can results from the average over many different runs of the given problem.

Optimization Problems and Classification Tasks

In optimization problems, the individuals in a population are the possible solutions ([Sivanandam and Deepa, 2007](#)). Fitness function assesses each solution by a fitness value ([Chang and Lippmann, 1991](#)).

Genetic search algorithms show a suitable approach to feature creation, feature and examples selection. These skills make GAs suitable for pattern classification task ([Chang and Lippmann, 1991](#)).

In classification problems, the individuals in the initial population are rules randomly generated. Classification accuracy evaluates the fitness of a rule, considering a set of training instances ([Han et al., 2012](#)).

Conclusive Remarks

The main advantages of GAs are robustness and independence of their search mechanism from the application field. In particular, domain independence allows that a new employee needs only a proper encoding of the given problem ([Janikow, 1993](#)). In speech

recognition domain, GAs have a processing time for features selection similar than traditional methods but use a lower number of input feature (Chang and Lippmann, 1991). GAs can work on tasks of artificial machine-vision producing outputs of classification with error rate nearly 0%, obtaining results better than KNN and Neural Network Classifiers (Chang and Lippmann, 1991). The ease of application is also an advantage of GAs, in addition to the effectiveness in finding a proper solution to in-depth search problem (Chang and Lippmann, 1991).

On the other hand, run time of GAs can be long (Chang and Lippmann, 1991; Janikow, 1993), being this one of major drawbacks of this technique. Furthermore, domain independence entails as a disadvantage that the performance of GAs is heavily dependent on the quality of the problem coding.

Linear and Nonlinear Regression Prediction

The aim of this section and the successive one is to introduce numerical prediction methods (See also the first two paragraphs of the “Data Mining: Classification and Prediction” article.), which answers to given task by supplying continuous target values. Problems that require numerical predictors can be addressed by a *function estimator* or *regressor* (Flach, 2012). It provides real numeric values starting from a set of numeric input data such as observations or measurements. This kind of predictor builds a *model function* that fits training data of a given task. Such a function will be able to predict unknown numerical data for a new instance pertaining to the same domain of the learned problem (Han et al., 2012).

Regression predictors forecast data trends by applying the statistical tool of regression analysis in several domains of interest from marketing management (Stock and Watson, 2003) to bioinformatics analysis (Wu et al., 2009).

Regression is an approach mostly used for numerical prediction. According to the kind of the problem to treat, two different models can be used, linear and nonlinear regression. Both models describe a relationship between a dependent variable y and one or more independent variables X . The main difference between these two type of regression relies on the relationship between the independent variable and the parameters of regression, i.e., if it is a linear combination of those or not.

The curve obtained from regression analysis is straight-line for linear regression and can be of other types for nonlinear regression (polynomial curves with a degree greater than 1 are considered a special case of multiple regression and can be treated with linear regression; they are an example of functions called linear regression models) (Flach, 2012; Bishop, 2006).

How Regression Prediction Methods Work?

Regression prediction methods fall into the category of *geometric models*. Here, the techniques represent a given task and therefore data set in an n -dimensional geometric space (Introduced in section “Linear and nonlinear classifier” of the “Data Mining: Classification and Prediction” article.), called feature space, being each instance described by n features. Regression methods construct classification models by employing geometric concepts such as lines or planes or considering measures of distance as similarities among properties.

To train a regression prediction method, it is necessary to model training data with a function f . Training data are a set of examples, where each instance is described by a pair (X, y) , $X = (x_1, \dots, x_n)$ is the vector of the features and y is the predefined numeric label representing a “true” function value. Starting from the training data set, a function f that maps an input vector of attributes X from the feature space to the space of real number will be built, providing $f(X)$ as output. The input X is the vector of the independent or predictor variables, whereas $f(X)$ is the dependent or *response variable* (Flach, 2012).

During the training process, the predictor learns the best function that fits the data, identifying the suitable parameters $w = (w_1, \dots, w_n)$ for f . The learning relies on the difference or *residual* (ϵ_i) between the actual and the predicted values, for the i -th instance respectively y_i and $f_w(X_i)$, as shown in Eq. (5)

$$\epsilon_i = y_i - f_w(X_i) \quad (5)$$

In particular, to avoid that positive and negative errors mutually cancel out, it is usual to consider squares of residuals (ϵ_i)² to evaluate regression models. Therefore, the *error function* (The error function was introduced in the chapter “Data Mining: Classification and Prediction”.) is expressed from the *residual sum-of-squares* $RSS(w)$, given by $\sum_{i=1}^p \epsilon_i^2$, as shown in the following formula (6):

$$RSS(w) = \sum_{i=1}^p (y_i - f_w(X_i))^2 \quad (6)$$

where the sum run over the total number p of instances in training set.

The *method of the least squares* is used to find the best function, minimizing the error function RSS with respect to the parameters w . The function f_w looks for this procedure will be bound to satisfy the relation of minimum squares of residuals, and it will be hence as close to the data as possible.

Linear Regression Predictor

Linear is the simplest type of regression. The model is a linear function of a predictor variable X . When a scalar predictor variable is used rather than a vector, the independent variable X is a single number x . In this basic case, known as *simple linear regression*, it is hypothesized that the model function describing each instance i of the data is straight-line, expressed by Eq. (7).

$$f_w(x_i) = w_0 + w_1 x_i \quad (7)$$

where the weights w_0 and w_1 are respectively the intercept and the slope of the line, also known as *regression coefficients* or *regression parameters*. To get the coefficients that are suitable for the given problem, $RSS(w)$ will be minimized. Indeed, according to the least squares method, the weights values will be obtained by solving the system of two *normal Eqs. (8) and (9)* which put equal to zero the partial derivatives of RSS with respect to the parameters weights w_0 and w_1 :

$$\frac{\partial RSS}{\partial w_0} = -2 \sum_i^p (y_i - w_0 - w_1 x_i) = 0 \quad (8)$$

$$\frac{\partial RSS}{\partial w_1} = -2 \sum_i^p (y_i - w_0 - w_1 x_i) x_i = 0 \quad (9)$$

The system leads to the solutions shown in Eqs. (10) and (11):

$$w_1 = \frac{\sum_{i=1}^p (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^p (x_i - \bar{x})^2} \quad (10)$$

and

$$w_0 = \bar{y} - w_1 \bar{x} \quad (11)$$

where $(x_i; y_i)$ is one of the p pairs of data set, and $\bar{x}$ and $\bar{y}$ are the mean value of the set $\{x_i\}$ and $\{y_i\}$ where $i = 1, 2, \dots, p$.

In training phase, the predictor finds the weights w_1 and w_0 and therefore the builds the function f that models the data set. Linear regression predictor will employ this function to make a numerical prediction about a new numeric instance x' as input in Eq. (7).

Multiple linear regression

When the independent variable is a vector $X = (x_1, \dots, x_n)$, Eq. (7) changes in Eq. (12):

$$f_w(X) = w_0 + \sum_{j=1}^n w_j x_j \quad (12)$$

where the vector of weights w expresses the regression coefficients (or parameters). Now, the curve that fits the data change from a straight-line to a plane (for $n=2$) or a more general hyperplane (for $n>2$). Each component of the vector X is a predictor variable, and hence the regression is defined *multiple* instead of *simple*, see previously, where a single scalar predictor variable x is used. Even in this case, the residual sum-of-squares $RSS(w)$, expressed by Eq. (6) has to be minimized with respect to the parameters w to obtain their best values, according to the minimal squares method.

Polynomial regression

Polynomial regression adopts the function $f_w(x)$, defined in Eq. (13):

$$f_w(x) = w_0 + w_1 x + w_2 x^2 + \dots + w_m x^m = \sum_{j=1}^m w_j x^j \quad (13)$$

where m is the order of the polynomial and x is the only predictor variable. Although the polynomial curve is a nonlinear function of the independent variable x , it is a linear combination of the polynomial coefficients w . Such kind of functions can be treated as linear regression. Indeed, linear regression is used for models that are linear in the parameters (are in the predictor variables linear or not).

The least squares method can be applied to estimate the parameters, resorting to the technique of multiple regression. Indeed, it is possible to consider a set of new variables as expressed in Eq. (14):

$$x^j = x_j \quad (14)$$

where $j = 1, \dots, m$. Replacing these new variables in Eq. (13), each term of order higher than one is substituted with a new independent variable and the polynomial function takes the form of Eq. (12), resulting in a linear form of the multiple predictor variables.

Nonlinear Regression

Nonlinear regression is employed to describe observational data by using a function $f_w(X)$ that is a nonlinear combination of the model parameters. In such function, the independent variables can be greater or equal to one. An example of nonlinear regression

curve is in the following Eq. (15):

$$f_w(X) = w_0 x_1^{w_1} x_2^{w_2} \quad (15)$$

where $w = (w_0, w_1, w_2)$ is the vector of the parameters and $X = (x_1, x_2)$ is the vector of independent variables.

For nonlinear regression, there is not a general method to obtain the best parameters. Indeed, the least squares method for nonlinear case leads to a system of normal equations without an analytic solution for w . Therefore, numerical methods will be considered to solving nonlinear least squares.

In general, nonlinear regression entails an iterative procedure to estimate the best fit. A first fitting curve is generated from an initial value for each parameter. The RSS function, see Eq. (6), estimates the vertical distance between the data points and this curve. The next step tries to reduce such distance adjusting regression parameters iteratively, leading to a minimum of RSS. The obtained parameters depend on the choice of the initial values and also on the numerical optimization methods adopted to minimize RSS.

Examples of numerical methods used to minimize RSS are:

- (i) the gradient descent (or steepest descent) method;
- (ii) the Gauss-Newton method;
- (iii) the Levenberg-Marquardt method.

Gradient descent (or steepest descent) method

This technique (It was introduced in the “Data Mining: Classification and Prediction” article, about the backpropagation algorithm of feedforward neural networks.) exploits the information of the gradient of a function to find, step by step, the direction of steepest descent to its minimum value. The method begins with an initial arbitrary solution of the minimization problem and then changes it with a variation in the direction indicated by the gradient.

In the regression case, RSS is the function to minimize by varying the parameters w . The method starts generating a first fitting curve resulting from an initial value $w^{(0)}$ for the parameters. Then, it calculates the RSS between the actual value y_i of data set and the value resulting from the first fitting curve $f_{w^{(0)}}(x_i)$. To minimize this RSS value, the initial solution $w^{(0)}$ will be updated iteratively considering the relation $w^{(k+1)} = w^{(k)} + \alpha_k p_k$, where k is the step of procedure iteration, α_k is the small amount of variation for the parameter w , and $p_k = -\nabla \text{RSS}(w^{(k)})$ is the direction of steepest descent. RSS is computed at every new step, to monitor the procedure. The minimum will be reached when every further variation of w will produce an increase in the corresponding RSS value.

Gauss-Newton method

As for the previous method, Gauss-Newton starts from an initial estimate for the parameters $w^{(0)}$. In the next step, it approximates the fitting function f as a function of the parameter w with a Taylor-series about the point $w^{(0)}$. At the 1st order, the fitting function is approximated to a line, according to the local linear model. Hence, RSS' is computed considering the linear approximation of f and minimized following the usual linear squares methods. The resulting value for the parameters $w^{(1)}$, is used in the next iteration of the method, until the entire procedure converges, i.e., when the original RSS is minimized (Ruckstuhl, 2010).

Levenberg-marquardt

It consists of a blending of the previous methods, taking respective advantages. Indeed, the gradient descent works well in the initial iterations, but its performance decreases when parameters are near to the best values. On the other hand, Gauss-Newton method works better in successive iterations rather than in early ones. Levenberg-Marquardt starts its process by using a gradient descent approach and gradually changes it with the Gauss-Newton approach (Motulsky and Christopoulos, 2004).

Further considerations

Unlike the linear case, nonlinear regression can find a local minimum of the RSS curve that is not the “true” best values for the regression parameters. An example of a false minimum is shown in Fig. 7. Finding a local minimum does not depend on the numerical method chosen, since it can be determined by the initial choice of the parameters $w^{(0)}$. To overcome this problem, it is advisable to repeat nonlinear regression many times with different parameters, i.e. using different initial values. The optimization process can guarantee the best fit solution if different choice of the vector $w^{(0)}$ for the regression curve lead to the same minimum in RSS curve.

Goodness of a Model

Once the model is built, it can be useful to confirm the goodness of the fit. The meaning of the data, when it is known, suggests the type of regression to use and, as a general rule, the goodness of a fit relies on the acceptability of the parameters’ value found. In general, to confirm the goodness of a fit, it is used the r^2 or R^2 index (traditionally distinct for linear and nonlinear regression respectively) and the residual analysis.

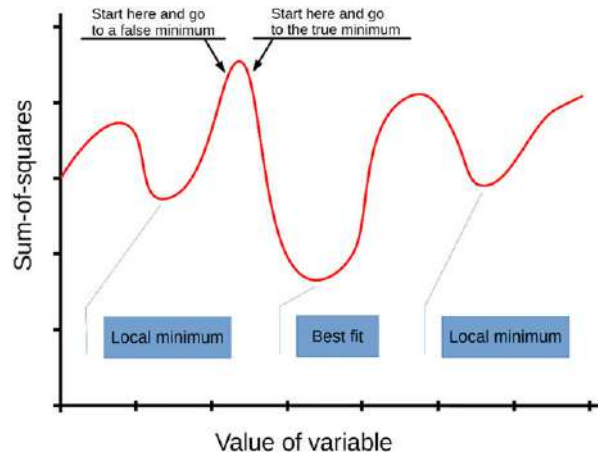


Fig. 7 Finding a local minimum is an intrinsic problem of nonlinear regression. It can be overcome by running nonlinear regression using different initial values for several computations.

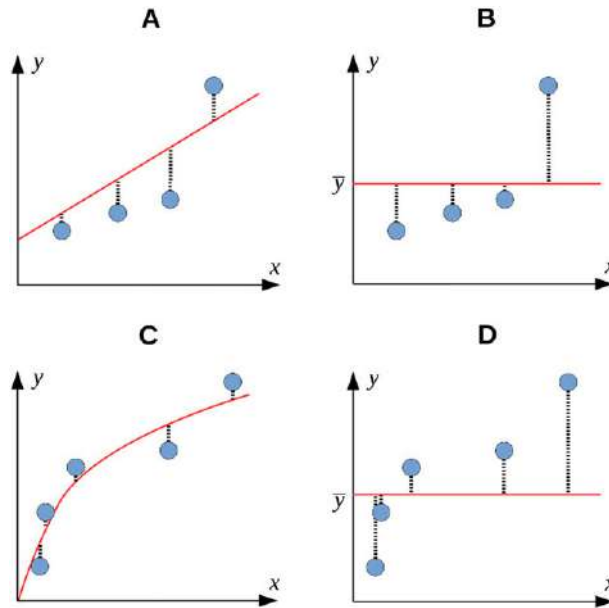


Fig. 8 Two cases of comparison between the fitting curve and the horizontal line passing through the mean value $\bar{y}$ of y_i data (B and D). Above: a case of linear regression (A); Below: a case of nonlinear regression (C).

r^2 or R^2

r^2 index is a number without units between 0.0 and 1.0. It is calculated through Eq. (16):

$$r^2 = 1.0 - \frac{RSS_{\text{reg}}}{RSS_{\text{tot}}} \quad (16)$$

where RSS_{reg} is the same of Eq. (6) and RSS_{tot} indicate the residual sum-of-squares of actual data point y_i from a horizontal line of equation $y = \bar{y}$, i.e., passing through the mean of all y_i values. r^2 , therefore, states if the found function of regression fits the data better than a horizontal line, in which case the r^2 value is close to 1. The comparison is shown in Fig. 8:

R^2 have the same meaning of r^2 but is capitalized to indicate nonlinear case.

Residual analysis

The residual analysis allows assessing if the model chosen to describe the data is suitable or not. This analysis evaluates the trend of residuals ϵ_i of Eq. (5) as a function of the independent variables X . The Fig. 9 illustrates three plots of residuals ϵ_i against the single independent variable x . A random pattern in the plot (left panel) confirms that a linear model is suitable for the regression analysis. Otherwise, another model would be used.

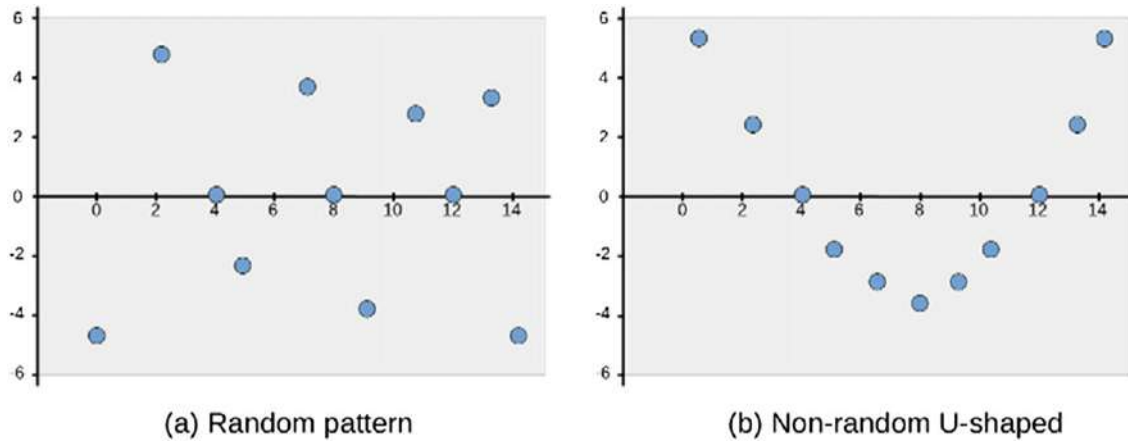


Fig. 9 Two typical patterns for residual plot: ϵ on the y-axis vs a single independent variable x . (a) the random plot suggest that the linear fitting model is a good choice to describe the data; (b) a non-random plot suggests a nonlinear model.

Conclusive Remarks

Method of regression rely on the assumption that the scatter of the data around the fitting line or curve follows a Gaussian or normal distribution. Therefore, the error of the dependent variable is random with mean zero. Furthermore, another hypothesis is that the variance of the error is a constant (*homoscedasticity*), i.e. the measure of the square variability of the error with respect to the mean of its distribution is uniform along the curve (Motulsky and Christopoulos, 2004). Therefore, there should not be any dependence of residuals from fitted values (otherwise, the method of *weighted* least squares has to be used; it will be discussed in Section “Estimating Parameters: Least Squares Estimation”).

Finally, the linear regression described above represents an appropriate prediction method when the dependent variable is a continuous variable (namely, it follows a Gaussian distribution). A generalization of linear regression can be made including non-continuous data for the dependent variable, and will be treated in Section “Generalized Linear Models”.

Generalized Linear Models

The regression analysis is a statistical method used to estimate the possible relationship between two types of variables. According to the regression method, a dependent variable Y or response variable (here, the response variable is indicated with uppercase to highlight that is a random variable) is expressed as a function of one or more independent variables, also known as *predictor* or *explanatory variables* of the form $X=(x_1, \dots, x_n)$. Generalized Linear Model (GLM) (Nelder and Wedderburn, 1972) is a generalization of the linear regression. It allows to model categorical response variables as a function of explanatory variables by using linear regression.

Categorical response variables or observations can be binary, having “yes” or “no” possibilities as outcomes or more generally, “success” or “failure”. To describe such binary observations of the response variable Y , it is usual to assume a binomial distribution for the data set y_i .

If the observations are counts, then a Poisson or negative binomial distribution is suitable to describe such data that will be nonnegative integers (Agresti, 1996).

Common types of GLMs are logistic regression and Poisson regression, which are employed respectively for the previous two cases. They will be described in the following sections.

Components of a Generalized Linear Model

A generalized linear model is composed of three components: i) random component, ii) systematic component, iii) link function. The *random component* specifies the response or dependent variable Y and the probability distribution hypothesized for it. The *systematic component* points out the explanatory or independent variables $(x_1, \dots, x_n)$, which describe each instance X_i of the data set, where $i=1, \dots, p$ is the total number of instances in the data set. Values of the explanatory variables are treated as fixed and not as random variable. The *link function* $g(\mu)$ indicates a function of mean μ of the probability distribution of Y , being $\mu=E(Y)$ the expected value or mean of Y . The expected value μ of a probability distribution can change depending on the explanatory variables. For example, the probability of the incidence of a disease can be considered as function of the presence of a risk factor. GLMs uses a *prediction equation* or *model equation* to relate this expected value or mean to the explanatory variables through the link function. Such model equation has a linear form, as shown in Eq. (17):

$$g(\mu) = \alpha + \beta_1 x_1 + \dots + \beta_n x_n \quad (17)$$

where the linear combination of the explanatory variables is known as *linear predictor*. α and β_j are the coefficient of regression, with $j=1, \dots, n$ if the independent variables are n .

The simplest link function is the *identity link* $g(\mu) = \mu$. If put in the left side of Eq. (17), it describe a linear model for the mean response as a function of the independent variables. Using only one explanatory variable, the model equation with identity link has the form shown in Eq. (18):

$$\mu(x) = \alpha + \beta x \quad (18)$$

This is known as ordinary linear regression model (the model Eq. (18) with regression coefficients α and β recalls Eq. (7) of the linear regression method with regression coefficients respectively to w_0 and w_1), employed for continuous responses. Therefore, linear regression is a special case of GLMs, known as *Normal GLM* because it assume a normal distribution for Y .

Other link functions consider a nonlinear relation between μ and the predictor variables. An example is the *log link function*, which considers the log of the mean $g(\mu) = \log(\mu)$. Prediction Eq. (17) employs log link function as left side when the mean μ can not be negative, as with count data. GLM using the log link function is known as the *loglinear model*.

Another nonlinear link function is the *logit link*: $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$, which models the log of an odds. It is convenient when $0 \leq \mu \leq 1$, as a probability. GLM using the logit link function is known as *logit model* or *logistic regression model*.

The next sections aim to describe GLMs for discrete responses in the two most important case: logistic regression models for binary data and loglinear models for count data.

Logistic Regression

Logistic regression (Cox, 1958) is the most common method for analysis of binary response data (Dobson, 2001). The logit model estimates the probability of a binary response Y (random component) as a function of one or more predictor or independent variables $\{x_j\}$ (systematic component), where $j=1, \dots, n$. For simplicity, it will be introduced the case of a single explanatory variable x .

The possible outcome for a response variable Y is denoted by 1 ("success") or 0 ("failure").

The distribution of the response variable Y is specified by the probabilities $P(Y=1) = \pi$ or $P(Y=0) = 1 - \pi$ for the single observation. The number y of successes in n independent observations follows the binomial distribution with index n and parameter π , as shown in Eq. (19):

$$P(y; \pi) = \binom{n}{y} \pi^y (1 - \pi)^{n-y} \quad (19)$$

where $y=1, 2, \dots, n$. The binomial distribution expected value (the expected value $E(Y)$ is the probability-weighted average of all possible outcomes of a random variable) or mean $E(Y) = \mu = n\pi$ and variance (the variance $\text{var}(Y) = \sigma^2 = E(Y - E(Y))^2$ is the expected value of the squared difference between a random variable and its mean) $\text{var}(Y) = \sigma^2 = n\pi(1 - \pi)$.

Binomial data are described by a GLM having the prediction Eq. (20):

$$g(\pi(x)) = \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \alpha + \beta x \quad (20)$$

where $\log\left(\frac{\pi(x)}{1 - \pi(x)}\right)$ is the *logit function* of π , also known as "logit(π)". Logit function is the link function between the mean probability π and the linear regression expression. From Eq. (20) it is possible to obtain the relationship between the success probability $\pi(x)$ and the predictor variable x . It is nonlinear, but varies continuously following a characteristic S-shaped curve, described by the so-called *logistic regression function* of Eq. (21):

$$\pi(x) = \frac{1}{1 + e^{-(\alpha + \beta x)}} \quad (21)$$

where x is the single independent or explanatory variable, α and β are the coefficients of regression. The β value establish the rate of sigmoid variation. As shown in Fig. 10, the increase or decrease of the curve depends on the positives or negative value of β . The rate of change of the sigmoid becomes steeper for greater values of $|\beta|$. For $\beta=0$ the binary response Y is independent of the explanatory variable x and the sigmoid becomes a horizontal line.

The outcomes of π are number between 0 and 1. They can be considered as the probability that the dependent variable Y is a success case or not (i.e., if the class label of Y is "yes" or "no" respectively).

Logistic regression classifier

Logistic regression can be used also to solve problems of classification. In general, logistic regression classifier can use a linear combination of more than one feature value or explanatory variable as argument of the sigmoid function. The corresponding output of the sigmoid function is a number between 0 and 1. The middle value is considered as threshold to establish what belong to the class 1 and to the class 0. In particular, an input producing an outcome greater than 0.5 is considered belong to the class 1. Conversely, if the output is less than 0.5, then the corresponding input is classified as belonging to 0 class (Harrington, 2012).

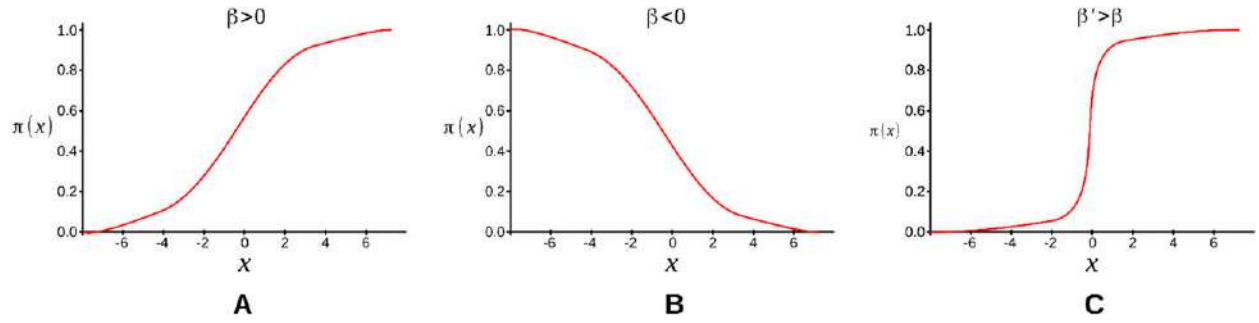


Fig. 10 Logistic regression functions, increasing and decreasing curve respectively for positives and negatives values of the β parameter. The second panel shows two different rate of change for two different positive value of β .

Nominal and ordinal logistic regression

Binomial or binary logistic regression is specific to treat a response variable with binary or binomial categories, as “true” or “false”, “male” or “female”, “healthy” or “sick” etc. Cases where the response variable considers more than two categories are described by nominal or ordinal logistic regression (Dobson, 2001), depending on whether they belong to ordered categories or not. An example of ordinal variable is one that can assume the values “low”, “medium” or “high” as the level of response to a medical treatment. It is an ordinal variables because follow ordered scales, unlike the nominal variables, for example “classical”, “rock”, “blues” or “jazz” as favorite type of music. Methods used for nominal variables can be used for ordinal ones. Conversely it is not true (Agresti, 1996).

Poisson Regression

Poisson regression is the GLM used to describe frequencies or count data. Such data usually exhibit a Poisson distribution for the probability to count a number y of events in a given temporal interval, knowing the average number θ of events in that interval. Poisson distribution is shown in Eq. (22):

$$P(y; \theta) = e^{-\theta} \frac{\theta^y}{y!} \quad (22)$$

where $y=0,1,2,\dots$. The only parameter of the distribution is θ and it can take positive values ($\theta > 0$); it expresses the mean and also the variance of the probability distribution, i.e., respectively $E(Y)=\theta$ and $var(Y)=\sigma^2=\theta$. This means that the variability of the counts tends to increase when the number of counts increases.

The link function for this GLM is the log link function. Therefore, the prediction equation follows the loglinear model shown in Eq. (23):

$$g(\theta) = \log(\theta) = \alpha + \beta x \quad (23)$$

The mean θ is an exponential function of the explanatory variable x , as shown in Eq. (24):

$$\theta(x) = \exp(\alpha + \beta x) = e^\alpha (e^\beta)^x \quad (24)$$

The values of the parameter β determine how the mean θ varies with the explanatory variable, increasing or decreasing respectively for positives or negatives β . If $\beta=0$ the means of Y is independent of x .

How GLMs Work

Estimating parameters: Maximum likelihood estimation

For a continuous random response variable Y , the probability density function or probability distribution is denoted as $P(y; \phi)$, where y is the observation of the Y variable for a fixed ϕ , which is the vector of parameters of the distribution, with components like the expected value $E(Y)$ and variance $var(Y)$. In general, the parameters values are unknown. Data sample allows to estimate these parameters by using the so-called *likelihood function*. The likelihood function $L(\phi; y)$ has the same algebraic form of the probability density function but conversely, L is a function of ϕ , for y fixed. The method of estimation of the parameter is based on the *Maximum Likelihood (ML) estimation*, which finds the parameter value that maximizes the likelihood function. The value of ϕ which maximizes the likelihood function is called *maximum likelihood estimate of ϕ* , and is indicated as $\hat{\phi}$. Frequently, it can be convenient to deal with the *log-likelihood function* $l(\phi; y) = \log L(\phi; y)$. Indeed, the parameter value sought $\hat{\phi}$ will maximize also the log-likelihood function. It will be found by differentiating the $L(\phi; y)$ or $l(\phi; y)$ with respect to each ϕ_j , and solving the simultaneous equations $\frac{\partial L(\phi; y)}{\partial \phi_j} = 0$ for $j=1,\dots,p$. In practical applications, numerical methods allow find $\hat{\phi}$ without solve these equations. GLMs works with such numerical approximations.

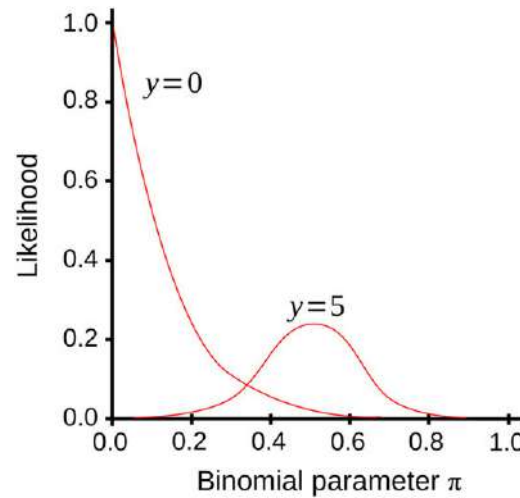


Fig. 11 Binomial likelihood functions for a number of success $y=0$ and $y=5$ successes in $n=10$ trials. Likelihood function says that the probability of $y=0$ for the head outcome in $n=10$ throws has a maximum for $\hat{\pi}=0$ (unfair coin).

Example of application

Considering the binomial case, whose probability distribution is expressed by Eq. (19), likelihood function says how the probability of a given number of successes y in n trials varies as a function of the success parameter π of the single trial. It is expressed by $L(\pi; y)$, where $0 \leq \pi \leq 1$ for binomial distribution. For example, the success parameter for the heads output ($Y=1$) in a single throw of a fair coin is $\pi=0.5$. The Fig. 11 shows the likelihood function for two fixed numbers of success $y=0$ and $y=5$ in $n=10$ throws. (The number of success could be considered as the head outcome in the coin throw). The likelihood function for $y=0$ has the form $L(\pi; 0) = \binom{10}{0} \pi^0 (1-\pi)^{10} = (1-\pi)^{10}$. For binomial outcomes of y successes in n trials, the likelihood function has a maximum for $\hat{\pi} = y/n$.

Estimating parameters: Least squares estimation

Another method of estimate for the parameter of a model is that of least squares, already introduced in the previous Section “Linear Regression Predictor” (linear and nonlinear regression). Here, considering Y_i independent random variable with $i=1, \dots, p$ and indicating with μ_i their respective expected values, the hypothesis is that each μ_i is a function of a regression parameters vector β . The estimator $\hat{\beta}$ is that minimizes RSS, namely the sum of squares of the differences between observed Y_i and expected values $\mu_i(\beta)$: $RSS = \sum [Y_i - \mu_i(\beta)]^2$ (as seen above in Eq. (6)). $\hat{\beta}$ is the parameter for which the RSS is minimized. Therefore, as seen in Eqs. (8) and (9), it is obtained by solving simultaneous equations of the type: $\frac{\partial RSS(\beta)}{\partial \beta_j} = 0$, where $j=1, \dots, n$.

If the variance σ_i^2 is not the same for each Y_i , then the weighted sum $RSS = \sum \frac{1}{\sigma_i^2} [Y_i - \mu_i(\beta)]^2$ is minimized, ensuring that the Y_i observations with greater variance, that mean less reliable, have a less effect on the estimates. This is the *weighted least squares method*.

Also for the least squares method, it is possible to resort to numerical approach to find the estimator.

Under assumption of Gaussian distribution for the value of Y , least squares estimates coincide with the ML estimates (Agresti, 1996).

Fitting generalized linear models

Newton-Raphson algorithm

In GLMs, a numerical algorithm is used to find ML estimates as distribution parameters values $\hat{\phi}$ of the model (as a function of regression coefficients $\hat{\beta}$). It starts assuming an initial value for the parameters maximizing the likelihood function. Through successive approximations, the parameters values tend to get closer to the ML estimates. For binomial logistic regression model and Poisson loglinear regression model, the above-mentioned method is the *Newton-Raphson algorithm* (Agresti, 1996). This algorithm is a simplification of the Fisher scoring algorithm. Through the Newton-Raphson algorithm, the log-likelihood function is approximated by a second – degree polynomial curve in the neighborhood of the initial parameter guess. Indeed, for a such parabola-shaped function it is easier to determine the value corresponding to the maximum. This value will be the second guess for the ML estimate. Hence, the algorithm iterates the approximation with the concave parabola in the neighborhood of this second guess. This procedure is repeated until the location of the maximum does not change anymore.

Model checking

It is possible to evaluate the goodness of a fit considering the distance between the observed value y_i and the fitted values $\hat{\mu}_i$, namely the residuals $y_i - \hat{\mu}_i$, for each observation i of the dataset.

To check a model involving a Normal distribution (Normal GLM), it is usual to consider a *standardized residual* obtained by dividing it by the estimate $\hat{\sigma}$ of its unknown σ parameter (the standard error $\sigma = \sqrt{\text{var}(Y)}$), namely the root square of the

variance), as shown in Eq. (25):

$$r_i = y_i - \hat{\mu}_i / \hat{\sigma} \quad (25)$$

If the assumption at the base of the model is correct, then the residuals should follow a Normal distribution with mean of zero and constant variance. They should be also independent from the explanatory variables. The goodness of the choice of the model adopted to describe the data is checked using suitable graphical methods. For example, a plot of residual against each explanatory variable should not show any pattern (as illustrate in Fig. 9). In addition, a plot of residuals against the fitted values $\hat{\mu}_i$ allows to detect an eventual change in variance, in violation of the assumption of constant variance (homoscedasticity) (as discussed in Section “Conclusive Remarks”) (under Linear and nonlinear Regression Prediction). Further, to examine the goodness of the model, the residuals can be aggregated and, as already seen above (Section “How Regression Prediction Methods Work”) it is preferable to consider the sum of the squares of residuals on the entire dataset: $\sum (y_i - \hat{\mu}_i)^2$.

A quantitative test to check the goodness of a fitting model is the χ^2 in the following Eq. (26):

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (26)$$

where the argument of the sum is the square of the standardized residuals of Eq. (25). In particular, Eq. (26) becomes Eq. (27) for the logistic model and Eq. (28) for the Poisson:

$$\sum r^2 = \sum \frac{(y_i - n\hat{\pi}_i)^2}{n\hat{\pi}_i(1 - \hat{\pi}_i)} \quad (27)$$

$$\sum r^2 = \sum \frac{(y_i - \hat{\theta}_i)^2}{\hat{\theta}_i} \quad (28)$$

where the sum run over the total number of data considered.

The χ^2 test evaluates the agreement between the observed distribution of data and the expected GLM employed to describe the data. It provides a numeric value that allows establishing if the supposed GLM describes adequately the data.

Conclusive Remarks

GLMs represent a unifying model of several statistical methods. They are a generalization of the linear regression (Section “Linear Regression Predictor”) because include response variables that follow a distribution model different from the Gaussian model (therefore not necessarily continuous but also categorical response variable) (Agresti, 1996; Dobson, 2001).

See also: Data Mining in Bioinformatics. The Challenge of Privacy in the Cloud

References

- Aamodt, A., Plaza, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7 (1), 39–59.
- Agresti, A., 1996. *An Introduction to Categorical Data Analysis*. John Wiley & Sons.
- Aha, D., 1992. Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies* 36, 267–287. Elsevier.
- Allen, B.P., 1994. Case-based reasoning: Business applications. *Communications of the ACM* 37, 40–42.
- Bareiss, E.R., Porter, B.W., Weir, C.C., 1988. Protos: An exemplar-based learning apprentice. *International Journal of Man-Machine Studies* 29, 549–561.
- Bhatia, N., 2010. Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security* 8 (2), 302–305.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*. Singapore: Springer.
- Boriah, S., Chandola, V., Kumar, V., 2008. Similarity measures for categorical data: A comparative evaluation. In: *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 243–254. Society for Industrial and Applied Mathematics.
- Chang, E.I., Lippmann, R.P., 1991. Using genetic algorithms to improve pattern classification performance. *Advances in Neural Information Processing Systems*. 797–803.
- Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. *IEEE Transaction on Information Theory* IT-13, 21–27.
- Cox, D.R., 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society, Series B (Methodological)*. 215–242.
- DeJong, K.A., 1985. Genetic algorithms: A 10 year perspective. In: *Proceedings of the First International Conference on Genetic Algorithms*, pp. 169–177.
- Dice, L.R., 1945. Measures of the amount of ecologic association between species. *Ecology* 26 (3), 297–302.
- Dobson, A.J., 2001. *An Introduction to Generalized Linear Models*, second ed. Chapman and Hall.
- Duda, R.O., Hart, P.E., Stork, D.G., 1991. *Pattern Classification*. New York: John Wiley & Sons.
- Fix, E., Hodges, J.L., 1952. Discriminatory analysis: Small sample performance. USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Rept. 11, August 1952.
- Fix, E., Hodges Jr., J.L., 1951. Discriminatory analysis, nonparametric discrimination. USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Rept. 4, Contract AF41(128)-31.
- Flach, P., 2012. *Machine Learning*. New York: Cambridge University Press.
- Gersho, A., Gray, R.M., 1992. *Vector Quantization and Signal Compression*. New York: Kluwer.
- Goldberg, D., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA: Addison-Wesley.
- Gu, D., Liang, C., Zhao, H., 2017. A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis. *Artificial Intelligence in Medicine* 77, 31–47.
- Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K., 2003. KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (Eds.), *Proceedings of OTM Confederated International Conferences on the Move to Meaningful Internet Systems*. pp. 986–996.

- Han, J., Kamber, M., Pei, J., 2012. *Data mining. Concepts and Techniques*, third edn. Waltham, MA: Morgan Kaufmann.
- Harrington, P., 2012. *Machine Learning in Action*. Shelte Island (NY): Manning.
- Hart, P.E., 1968. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory* 14, 515–516.
- Hastie, T., Tibshirani, R., Friedman, J., 2008. *The Elements of Statistical Learning*, second edn. Stanford, CA: Springer.
- Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.
- Janikow, C.Z., 1993. A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning* 13 (2–3), 189–228.
- Kolodner, J., 1983. Maintaining organization in a dynamic long-term memory. *Cognitive Science* 7, 243–280.
- Kolodner, J.L., 1993. *Case-Based Reasoning*. San Matteo, CA: Morgan Kaufmann.
- Koton, P., 1988. Reasoning about evidence in causal explanation. In: *Proceedings of the 7th National Conference of Artificial Intelligence (AAAI'88)*, pp 256–263.
- Levandowsky, M., Winter, D., 1971. Distance between sets. *Nature* 234 (5323), 34–35.
- Michalewicz, Z., 1992. *Genetic Algorithms + Data Structures = Evolution Programs*. USA: Springer Verlag.
- Mitchell, M., 1996. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.
- Motulsky, H., Christopoulos, A., 2004. *Fitting models to biological data using linear and nonlinear regression: A practical guide to curve fitting*. Place: Oxford University Press.
- Nelder, J., Wedderburn, R., 1972. Generalized Linear Models. *Journal of the Royal Statistical Society* 135, 370–384.
- Porter, B., Bareiss, R., 1986. PROTON: An experiment in knowledge acquisition for heuristic classification tasks. In: *Proceedings of the First International Meeting on Advances in Learning (IMAL)*, Les Arcs, France, pp. 159–174.
- Preparata, F.P., Shamos, M.I., 1985. *Computational Geometry: An Introduction*. New York: Springer-Verlag.
- Qian, G., Sural, S., Gu, Y., Pramanik, S., 2004. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In: *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 1232–1237. ACM.
- Rechenberg, I., 1973. *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog Verlag.
- Riesbeck, C., Schank, R., 1989. *Inside Case-Based Reasoning*. Hillsdale: Lawrence Erlbaum.
- Riolo, R.L., 1992. Survival of the fittest bits. *Scientific American* 267 (1), 114–116.
- Rissland, E.L., Ashley, K., 1987. HYPO: A case-based system for trade secret law. In: *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, pp. 60–66. Boston, MA: ACM.
- Royas, R., 1996. *Neural Networks. A systematic Introduction*. Berlin: Springer.
- Ruckstuhl, A., 2010. *Introduction to Nonlinear Regression*.
- Schank, R.C., 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge: Cambridge University Press.
- Schank, R.C., 1984. Memory-based expert systems. Technical Report No: AFOSR. TR. 84-0814, Yale University, New Haven, USA.
- Sivanandam, S.N., Deepa, S.N., 2007. *Introduction to Genetic Algorithms*. Heidelberg: Springer Science & Business Media.
- Stanfill, C., Waltz, D., 1986. Toward memory-based reasoning. *Communications of the ACM* 29 (12), 1213–1228.
- Stock, J.H., Watson, M.W., 2003. *Introduction to Econometrics*, vol. 104. Boston, MA: Addison Wesley.
- Tanimoto, T.T., 1958. *Elementary mathematical theory of classification and prediction*.
- Watson, I., Marir, F., 1994. Case-based reasoning: A review. *The Knowledge Engineering Review* 9, 327–354.
- Wu, T.T., Chen, Y.F., Hastie, T., Sobel, E., Lange, K., 2009. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics* 25 (6), 714–721.

Further Reading

- Agresti, A., 1990. *Categorical Data Analysis*. New York: Wiley.
- Bailey, T., Jain, A.K., 1978. A note on distance-weighted k-nearest neighbor rules. *IEEE Transactions on Systems, Man, and Cybernetics* 4, 311–313.
- Booker, L., B., Goldberg, D.E., Holland, J.H., 1989. Classifier systems and genetic algorithms. *Artificial Intelligence* 40, 235–282.
- Dasarathy, B., 1991. *Nearest Neighbor Pattern Classification Techniques*. Silver Spring, MD: IEEE Computer Society Press.
- Finnie, G.R., Wittig, G.E., Desharnais, J.M., 1997. A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models. *Journal of Systems and Software* 39 (3), 281–289.
- Friedman, J.H., 1991. Multivariate adaptive regression splines. *The Annals of Statistics* 19, 1–67.
- Harrell, F., 2015. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer.
- Kim, G.H., An, S.H., Kang, K.I., 2004. Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. *Building and Environment* 39 (10), 1235–1242.
- Leake, D.B., 1996. CBR in context: The present and future. In: Leake, D.B. (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park: AAAI Press, pp. 3–30.
- Whitley, D., 1994. A genetic algorithm tutorial. *Statistics and Computing* 4 (2), 65–85.

Biographical Sketch



Alfonso Urso received the Laurea degree in electronic engineering (*Summa cum Laude*), and the PhD degree in systems engineering from the University of Palermo, Italy, in 1992 and 1997, respectively. In 1998, he was with the Department of Computer and Control Engineering, University of Palermo, as Research Associate. In 2000, he joined the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR), where he is currently a Researcher in systems and computer engineering. From 2007 to 2015 he was coordinator of the research group “Intelligent Data Analysis for Bioinformatics” of the Italian National Research Council. Since January 2016 he is Head of the Palermo branch of the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR). Since 2001 he has been Lecturer at the University of Palermo. His research interests are in the area of machine learning, soft computing and applications to bioinformatics. Dr. Urso is member of the IEEE (Institute of Electrical and Electronic Engineers), and of the Italian Bioinformatics Society (BITS).



Antonino Fiannaca received the Laurea degree in computer science engineering (Summa cum Laude), and the PhD degree in computer science from the University of Palermo, Italy, in 2006 and 2011, respectively. In 2006, he joined the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR), where he is currently a Researcher. He is part of the Translational Bioinformatics Laboratory at ICAR-CNR. His research interests are in the area of machine learning, neural networks and bioinformatics.



Massimo La Rosa received the Laurea degree in computer science engineering (Summa cum Laude), and the PhD degree in computer science from the University of Palermo, Italy, in 2006 and 2011, respectively. In 2006, he joined the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR), where he is currently a Researcher. He is part of the Translational Bioinformatics Laboratory at ICAR-CNR. He is member of Bioinformatics Italian Society (BITS) from 2016. His research interests are in the area of machine learning, data mining and bioinformatics.



Valentina Ravi received her Master Degree in Physics, with Biophysical address (Summa cum Laude), at the University of Study of Palermo, Italy. She has held post-graduate internships at the same University, where she collaborated to tutoring, teaching and dissemination activities. She has done a post-Degree training in the area of interest of data mining and support decision systems at ICAR-CNR of the National Research Council, Palermo. Here she is currently involved in research activities in bioinformatics domain, with the focus on the study of data mining techniques.



Riccardo Rizzo is staff researcher at Institute for high performance computing and networking, Italian National Research Council. His research is focused mainly on machine learning methods of sequence analysis, and biomedical data analysis, and on neural networks applications. He was Visiting Professor at the University of Pittsburgh, School of Information Science, in 2001. He participates in the program committee of several international conferences and has served as editor of some Supplements of the BMC Bioinformatics journal. He is author of more than 100 scientific, 33 of them in international journals, such as "IEEE Transactions on Neural Networks", "Neural Computing and Applications", "Neural Processing Letters", "BMC Bioinformatics". He participates in many national research projects and is co-author of one international patent.

Data Mining: Accuracy and Error Measures for Classification and Prediction

Paola Galdi, University of Salerno, Fisciano, Italy

Roberto Tagliaferri, University of Salerno, Salerno, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The goal of data mining is that of building learning models to automatically extract knowledge from big amounts of complex data. In supervised learning, prior information is used to train a model to learn latent relationships between data objects. Once the model is trained, it is used to perform predictions on previously unseen data.

According to the type of prediction, we can distinguish between classification, where the output is a categorical class label, and regression, where the model learns a continuous function. In the former task, data are divided into groups according to some discriminative features. In the latter, input features are put into a functional relationship with a variable of interest. In both cases, a method is needed to quantify the performance of a model, i.e., to determine how accurate are its predictions.

During the training phase, measuring accuracy plays a relevant role in model selection: parameters are selected in order to maximize prediction accuracy on training samples. At the end of the learning step, accuracy is measured to assess the model predictive ability on new data. Being learning algorithms trained on finite samples, the risk is to overfit training data: the model might memorize the training samples instead of learning a general rule, i.e. the data generating model. For this reason, a high accuracy on unseen data is an index of the model generalization ability.

The remainder of the article is organized as follows: in Sections “Accuracy Measures in Classification” and “Accuracy Measures in Regression” the main accuracy measures used in classification and regression tasks are presented, respectively. Section “Model Selection and Assessment” explores how accuracy measures can be used in combination with validation techniques for model selection and model assessment. Section “Improving Accuracy” introduces bagging and boosting, two techniques for improving the prediction accuracy of classification models.

Accuracy Measures in Classification

The accuracy of a classifier is the probability of correctly predicting the class of an unlabelled instance and it can be estimated in several ways (Baldi *et al.*, 2000). Let us assume for simplicity to have a two-class problem: as an example, consider the case of a diagnostic test to discriminate between subjects affected by a disease (patients) and healthy subjects (controls). Accuracy measures for binary classification can be described in terms of four values:

- TP or true positives, the number of correctly classified patients.
- TN or true negatives, the number of correctly classified controls.
- FP or false positives, the number of controls classified as patients.
- FN or false negatives, the number of patients classified as controls.

The sum of TP, TN, FP and FN equals N , the number of instances to classify. These values can be arranged in a 2×2 matrix called contingency matrix, where we have the actual classes P and C on the rows, and the predicted classes $\tilde{P}$ and $\tilde{C}$ on the columns.

	$\tilde{P}$	$\tilde{C}$
P	TP	FN
C	FP	TN

The classifier sensitivity (also known as recall) is defined as the proportion of true positives on the total number of positive instances:

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (1)$$

In the above example, it is the number of correctly diagnosed patients on the total number of subjects affected by the disease.

The classifier specificity (also referred to as precision) is defined as the proportion of true positives on the total number of instances identified as positive (in some contexts, the term specificity is used to indicate the sensitivity of the negative class):

$$\text{specificity} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

In the example, it corresponds to the number of correctly diagnosed patients among all the positive diagnoses.

A low sensitivity corresponds to a high number of false negatives, while a low specificity indicates the presence of many false positives (see Fig. 1 for an example). According to the context, high rates of false negative or false positive predictions might have

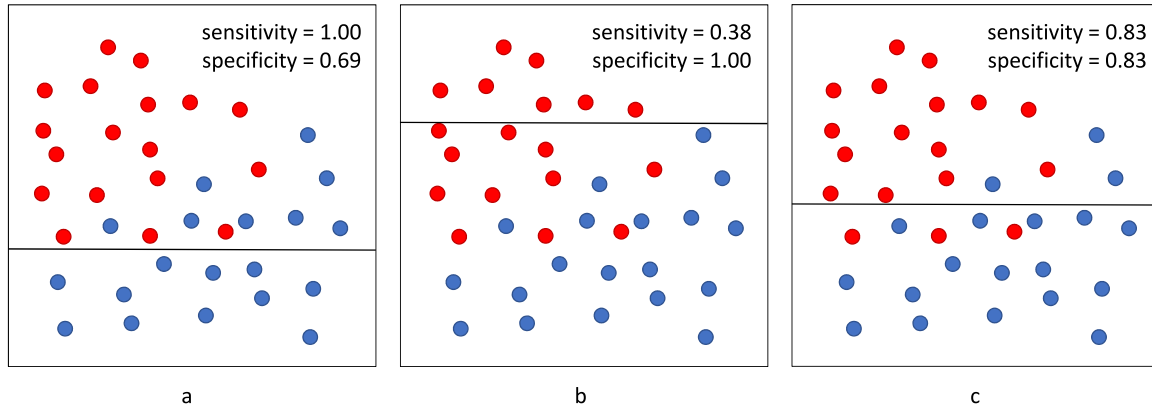


Fig. 1 The trade-off between specificity and sensitivity. In the first panel (a) the sensitivity for the red class is highest since all red dots have been assigned to the same class, but the specificity is low since there are many false positives (blue dots in the red class). In the second panel (b) the specificity for the red class is highest since all points assigned to that class are red, but the sensitivity is low for the presence of many false negatives (red dots in the blue class). In the last panel (c) there is a balance between sensitivity and specificity.

different implications: consider failing to diagnose a sick subject (false negative) versus assigning a treatment to a healthy subject (false positive).

The simplest way to estimate accuracy is to compute the percentage of correctly classified patients (positive instances):

$$PA_{pos} = 100 \cdot \frac{TP}{TP + FN} \quad (3)$$

which is complemented by the percentage of correctly classified controls (negative instances):

$$PA_{neg} = 100 \cdot \frac{TN}{TN + FP} \quad (4)$$

To account for all correctly classified instances, accuracy can be computed as:

$$PA = 100 \cdot \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Another option is to average PA_{pos} and PA_{neg} to obtain the average percentage accuracy on both classes.

$$PA_{avg} = \frac{PA_{pos} + PA_{neg}}{2} \quad (6)$$

F-measure (or F-score or F_1 -score) has been introduced to balance between sensitivity and specificity. It is defined as the harmonic mean of the two scores, multiplied by 2 to obtain a score of 1 when both sensitivity and specificity equal 1:

$$F = 2 \cdot \frac{1}{\frac{1}{sensitivity} + \frac{1}{specificity}} = 2 \cdot \frac{specificity \cdot sensitivity}{specificity + sensitivity} \quad (7)$$

Care must be taken when selecting an accuracy measure in presence of unbalanced classes. Following again the above example, consider the case of a dataset where 90% of the subjects are sick subjects and the remaining 10% consists of healthy controls. Taking into account only the percentage of correctly classified instances, a model which assigns every subject to the patients class would attain an accuracy of 90%. In the same case the F-measure would be equal to 0.95, while the average percentage accuracy on both classes PA_{avg} would yield a more informative score of 45%.

When working with more than two classes, say M , the resulting contingency matrix $X = \{x_{ij}\}$ is a $M \times M$ matrix where each entry x_{ij} is the number of instances belonging to class i that have been assigned to class j , for $i, j = 1, \dots, M$. The sensitivity for class i can then be computed as:

$$sensitivity_i = 100 \cdot \frac{x_{ii}}{n_i} \quad (8)$$

where n_i is the number of instances in class i . Similarly, the specificity for class i is:

$$specificity_i = 100 \cdot \frac{x_{ii}}{p_i} \quad (9)$$

where p_i is the total number of instances predicted to be in class i . The percentage of all correct predictions corresponds to:

$$PA = 100 \cdot \frac{\sum_i x_{ii}}{N} \quad (10)$$

Accuracy Measures in Regression

A regression model is accurate if it predicts for a given input pattern its target value with a low error. Given a vector $\hat{y}$ of N predictions and the vector y of N actual observed target values, the most commonly adopted error measure is the mean squared error, computed as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (11)$$

Another variant is the root mean squared error:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (12)$$

Other measures are based on the absolute difference between predicted and actual values, like the mean absolute error:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (13)$$

The above measures have the same scale as the input data. A scale-independent alternative is the relative absolute error, computed as:

$$\text{RAE} = \frac{\sum_{i=1}^N |\hat{y}_i - y_i|}{\sum_{i=1}^N |\bar{y} - y_i|} \quad (14)$$

where $\bar{y}$ is the mean of vector y , and therefore the quantity $\sum_{i=1}^N |\bar{y} - y_i|$ acts as a normalizing factor to get a value between 0 and 1. Multiplying this value by 100 yields the percentage absolute error.

The relative squared error is obtained in a similar fashion:

$$\text{RSE} = \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (\bar{y} - y_i)^2} \quad (15)$$

The choice between squared and absolute error depends on several considerations. For example, squared error emphasizes larger differences while absolute error is more robust to outliers causing occasionally large errors. Squared error is sometimes preferred because of its mathematical properties. Contrary to absolute error, it is continuously differentiable, making analytical optimization easier. In addition, when fitting a Gaussian distribution, the maximum likelihood fit is that which minimizes the squared error. On the other hand, absolute error uses the same unit as the data (instead of square units) and might be more straightforward to interpret.

Model Selection and Assessment

The standard approach when evaluating the accuracy of a model is the hold out method. It consists in splitting the available data in two disjoint sets: the training set, which is fed into the learning algorithm, and the test set, used only to evaluate the performance of the model. Commonly, two thirds of the data items are destined to the training and the remainder is reserved for the testing, but other proportions like 70:30, 80:20 and 90:10 are also adopted. There is a trade-off between the training set size and the test set size: using more data for the training allows to build more accurate models, but with fewer instances in the test set the estimate of prediction accuracy will have a greater variance.

When models depend on one or more parameters, a third set of data, the validation set, is required to perform model selection: several models are trained using only data coming from the training set, with different values for each parameter; from the pool of trained models, one winner is selected as that which achieves the best score according to a given accuracy measure on the validation set; finally, the generalization ability of the model is assessed on the test set. Two separated sets of data are necessary for model selection and assessment because using the validation set for the selection of the final model might introduce bias. Therefore, new unseen data are needed to avoid the underestimation of the true error.

Since the choice of the accuracy measure to optimize greatly affects the selection of the best model, then the proper score should be determined taking into account the goal of the analysis. When performing model selection in a binary classification problem, e.g. when selecting the best threshold for a classifier with a continuous output, a reasonable criterion is to find a compromise between the amount of false positives and the amount of false negatives. The receiver operating characteristic (ROC) curve is a graphical representation of the true positive rate (the sensitivity) as a function of the false positive rate (the so called false alarm rate, computed as $\text{FP}/(\text{FP} + \text{TN})$), as shown in Fig. 2. A perfect classifier, with a 100% true positive rate and no false positives would be represented by a (0, 1) coordinate in the ROC space, while chance level corresponds to the diagonal of the plot. Therefore, a good classifier would be represented by a point near the upper left corner of the graph and far from the diagonal. An indicator related to the ROC curve is the area under the curve (AUC), that is equal to 1 for a perfect classifier and to 0.5 for a random guess.

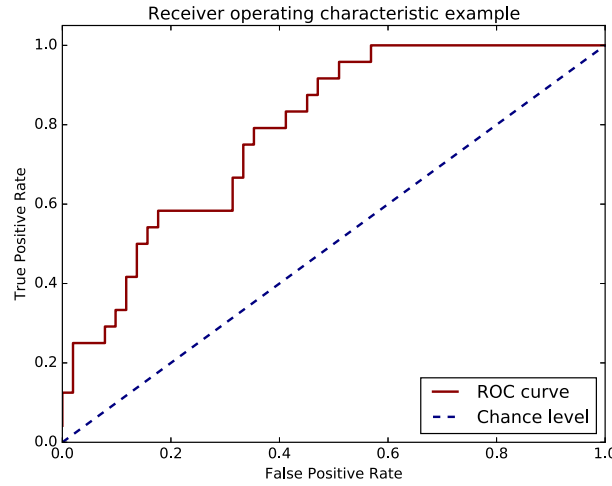


Fig. 2 Receiver Operating Characteristic (ROC) curve.

The main drawback of the hold out method is that it makes an inefficient use of data using only a part of them for the training, thus providing a pessimistic estimate of the model actual accuracy. For this reason, other approaches have been devised to take advantage of all available data, especially in the case of datasets with few samples. Another limitation of the hold out method is that the outcome depends on the specific choice of training and test set. To address this issue, random subsampling can be used to randomly generate multiple training/test splits and the accuracy is computed as the average accuracy across runs. However, if a class happens to be over-represented in the training set, it will be under-represented in the test set, and vice versa, thus affecting the prediction accuracy estimation. Therefore, stratification can be used to preserve the proportion of classes in both training and test set.

Cross-validation is a technique for accuracy estimation where the original dataset is randomly split into k disjoint sets of the same size. The model to be evaluated is trained k times, selecting in rotation $k - 1$ sets for the training and the k -th for the validation. The final accuracy is the overall number of correctly classified instances in each fold divided by N , the total number of instances in the dataset. When $k = N$, the scheme is called leave-one-out, since in each fold the test set is composed by a single object. As for the hold out method, cross-validation can be repeated multiple times using different random splits to obtain a better estimation of prediction accuracy (In a complete cross-validation scheme, all the $\binom{N}{k}$ possible splits are tested, but this is usually prohibitive in terms of computational costs.) and splits can be stratified to preserve class proportions. The advantage of this approach is that it exploits all available data. In particular, the leave-one-out method tests the model on all the possible N splits, and it is suitable in case of few samples, but it is characterized by a high variance. For this reason, small values for k are usually preferred, like in 5- and 10-cross-validation schemes (Kohavi *et al.*, 1995).

Bootstrap (Efron, 1983) is an alternative to cross-validation that relies on random subsampling with replacement (see Fig. 3 for an example). It is based on the assumption that samples are independently and identically distributed and it is especially recommended when the sample size is small. After n extractions, the probability that a given object has not been extracted yet is equal to $(1 - \frac{1}{n})^n \approx e^{-1} \approx 0.368$, therefore the expected number of distinct instances in the sample is $0.632n$. The 0.632 bootstrap estimate of accuracy, introduced in Efron and Tibshirani (1997), is defined as:

$$\text{acc}_{boot} = \frac{1}{b} \sum_{i=1}^b (0.632 \cdot \text{acc}_i + 0.368 \cdot \text{acc}_{train}) \quad (16)$$

where b is the number of generated bootstrap sample, acc_i is the accuracy obtained with a model trained on sample i and tested on the remaining instances, and acc_{train} is the accuracy of the model trained on the full training set.

Once the accuracy of a model has been estimated on a sample with one of the above methods, the amount of uncertainty associated with it can be described by a confidence interval, a range of values that is likely to contain the true value. Confidence intervals are computed for a given a confidence level expressed as a percentage (commonly adopted values are 90%, 95% and 99%), that is interpreted in the following way: if multiple samples are extracted from the same population and a confidence interval with a confidence level of, say, 90%, is computed for each sample, 90% of intervals will contain the true value (Fig. 4). The width of a confidence interval is affected by the size of the sample and its variability, i.e., a small sample characterized by a high variability would yield larger confidence intervals.

In binary classification problems, confidence intervals for the accuracy estimate can be calculated assuming a binomial distribution for the number of correct predictions on the test set; in other words, each prediction is modelled as a Bernoulli variable whose two possible values correspond to correct or incorrect prediction. In general, when a sufficient number of samples is available, approximated confidence intervals can be calculated assuming a Gaussian sampling distribution.

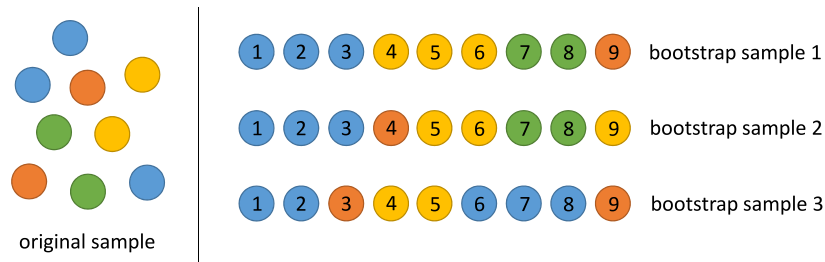


Fig. 3 An example of bootstrap sampling. Since objects are subsampled with replacement, some classes might be over-represented (yellow marbles in bootstrap samples 1 and 2), others might be under-represented (red marbles in bootstrap samples 1 and 2) or even missing (green marbles in bootstrap sample 3).

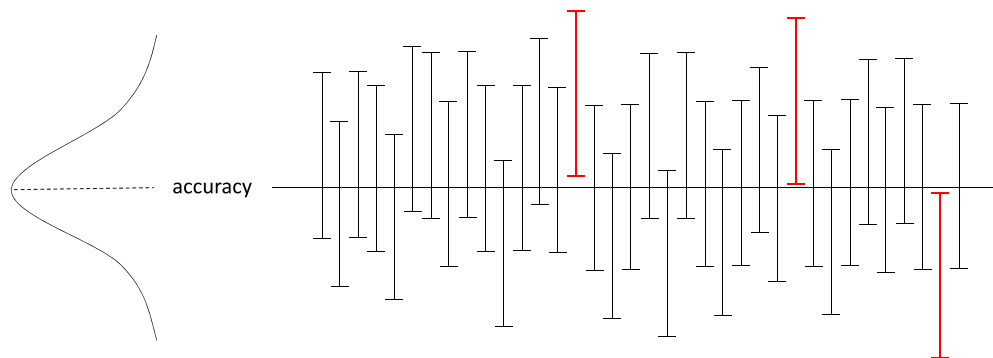


Fig. 4 Confidence intervals estimated with 90% confidence level: in 3 out of 30 samples from the same population the confidence intervals do not contain the true value for accuracy.

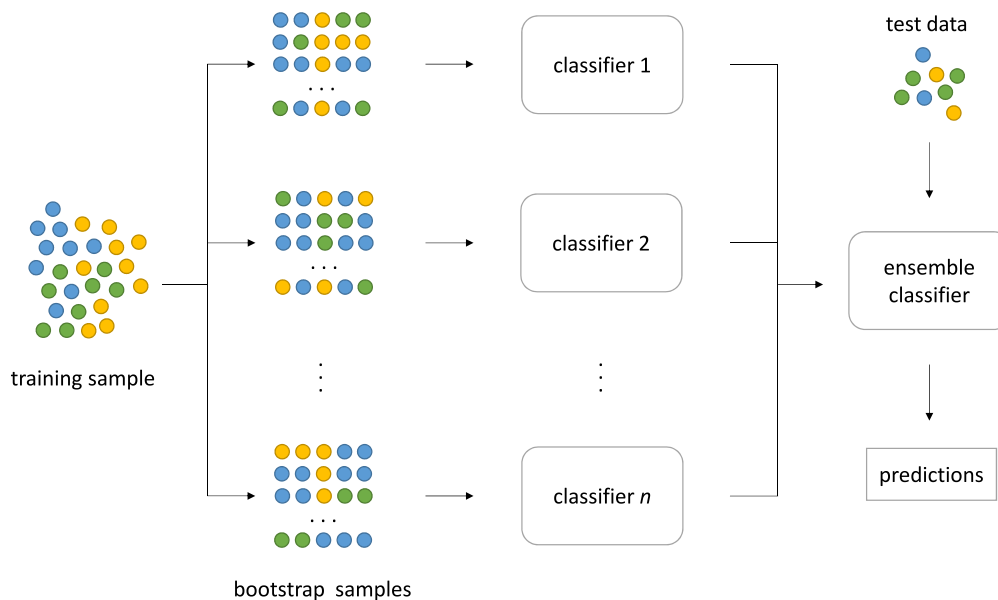


Fig. 5 The bagging approach. Several classifier are trained on bootstrap samples of the training data. Predictions on test data are obtained combining the predictions of the trained classifiers with a majority voting scheme.

Improving Accuracy

Ensemble methods are a class of algorithms that build models by combining predictions of multiple base classifiers. Two prerequisites for the ensemble of classifiers are that they should be diverse, i.e., predict with different accuracies on new data, and accurate, meaning that they should perform better than a random guess (Dietterich, 2000a). There are several approaches to this

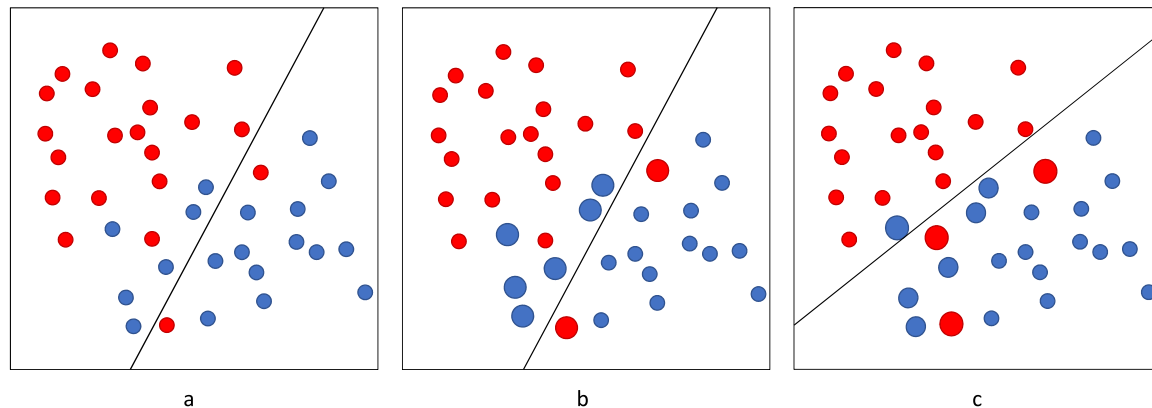


Fig. 6 The boosting approach. A classifier is trained on the original data (a). The weights of misclassified instances (dot size in the figure) are increased (b). A new classifier is trained on the new data set and weights are updated accordingly (c).

problem, the most relevant ones being bagging and boosting. The main idea behind these techniques is to generate many perturbed versions of the original training set on which different classifiers are built. Then, the final classification is obtained with a majority voting scheme, where each classifier expresses its vote for class membership of test instances. Both methods have been shown to outperform prediction accuracies of single classifiers (Quinlan *et al.*, 1996).

Bagging (as depicted in Fig. 5) works by creating several training sets by subsampling with replacement the input data, as in bootstrap sampling: the new sets have the same size of the original training set but with some instances missing and some present more than once. The performance of this method depends on how unstable the base models are: if small perturbations of the training set cause a significant variability in the trained classifiers, then bagging can improve the final accuracy (Breiman, 1996). This approach has been successfully applied in the Random Forest model (Breiman, 2001), where the outputs of an ensemble of Decision Tree classifiers are combined in a final solution.

While in bagging each sample is extracted independently and the different classifiers are trained in parallel, boosting follows an iterative scheme, where at each repetition each object is associated with a weight that reflects the performance of the current classifier. Specifically, weights of misclassified instances are increased while weights of correctly classified instances are decreased (see Fig. 6). Then, at the beginning of the next iteration, objects are sampled from the original training set with a probability proportional to their weights (Dietterich, 2000b). The goal is to guide the training of new models towards a solution that correctly predicts the instances that were misclassified in the original training set, thus affecting accuracy. AdaBoost is a prominent example of this class of methods (Freund and Schapire, 1995; Freund *et al.*, 1996).

See also: Data Mining: Classification and Prediction. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Machine Learning in Bioinformatics. Supervised Learning: Classification

References

- Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A., Nielsen, H., 2000. Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics* 16 (5), 412–424.
- Breiman, L., 1996. Bagging predictors. *Machine Learning* 24 (2), 123–140.
- Breiman, L., 2001. Random forests. *Machine Learning* 45 (1), 5–32.
- Dietterich, T.G., 2000a. Ensemble methods in machine learning. In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 1–15.
- Dietterich, T.G., 2000b. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning* 40 (2), 139–157.
- Efron, B., 1983. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association* 78 (382), 316–331.
- Efron, B., Tibshirani, R., 1997. Improvements on cross-validation: The 632+ bootstrap method. *Journal of the American Statistical Association* 92 (438), 548–560.
- Freund, Y., Schapire, R.E., 1995. A decision-theoretic generalization of online learning and an application to boosting. In: *European conference on computational learning theory*. Springer, pp. 23–37.
- Freund, Y., Schapire, R.E., *et al.*, 1996. Experiments with a new boosting algorithm. In: *ICML 96*, pp.148–156.
- Kohavi, R., *et al.*, 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *IJCAI*, vol. 14, Stanford, CA, pp. 1137–1145.
- Quinlan, J.R., *et al.*, 1996. Bagging, boosting, and C4.5. In: *AAAI/IAAI*, vol. 1, pp. 725–730.

Further Reading

- Efron, B., Tibshirani, R.J., 1994. *An Introduction to the Bootstrap*. CRC press.
- Molinaro, A.M., Simon, R., Pfeiffer, R.M., 2005. Prediction error estimation: A comparison of resampling methods. *Bioinformatics* 21 (15), 3301–3307.

Data Mining: Clustering

Alessia Amelio and Andrea Tagarelli, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The term “clustering” in its most general meaning refers to the methodology of partitioning elements in groups according to some common characteristics. Cluster analysis was introduced for the first time in anthropology by Driver and Kroeber in 1932 and in psychology by Zubin in 1938 and Robert Tryon in 1939 (Bailey, 1994; Tryon, 1939) with the need of characterizing the typology of different individuals and cultures. Later, it was used by Cattell in 1943 for trait theory classification in personality psychology (Cattell, 1943). However, clustering analysis developed as a major topic in the 1960 and 1970 when the monograph “The Principles and Practice of Numerical Taxonomy” appeared as a clear motivation on researching the clustering techniques (Sokal, 1963). Later, different books were published about the topic, such as “Les bases de la classification automatique” (Lerman, 1970), “Mathematical Taxonomy” (Jardine and Sibson, 1971) and “Cluster analysis for applications” (Anderberg, 1973) which formalized the problem and presented the clustering approaches.

Starting from 1960, clustering has been used in different disciplines, including biology, psychology, anthropology, political science, sociology and social sciences, geography, economics and literature. In biology, clustering has been applied for counting dust particles and bacteria. In geography, it has been employed for solving problems of land allocation in urban planning. In political science, clustering has been used for supporting tasks of political campaigning (Krippendorff, 1980).

Nowdays, clustering has become a valid instrument for solving complex problems of computer science and statistics. In particular, it is very used in data mining and effective for discovering patterns of specific interest from data in order to support the process of knowledge discovery. Generally, in order to optimize the clustering solution, data are pre-processed before applying a clustering approach. The specific type of pre-processing task mainly depends on the used clustering approach. Pre-processing tasks may include: (i) removal of noise or outliers from data, which is essential when the clustering approach is sensitive to noise or outliers, (ii) data normalization, which is important for distance-based clustering, and (iii) data reduction (sampling or attributes reduction), which can be useful when the clustering approach is computationally expensive. Normalization consists of scaling the data to fall within a given range. Data reduction can remove irrelevant attributes by attribute selection or principal component analysis which finds a lower dimensional space representing the data, or remove instances of data by using a sampling method.

In this article, we present the main concepts underlying the clustering methodology. We start by providing a background of the clustering problem, together with the key elements which are essential for its resolution. Among the others, we present some meaningful distance and similarity measures commonly adopted in the literature for clustering. Then, we provide a full categorization of the main clustering approaches, by describing some relevant and well-known algorithms for each category. It includes the description of clustering methods specifically designed for non-conventional data, such as the images and XML data. A section is also dedicated to the application of traditional clustering methods to data of different domains. Finally, different approaches for the evaluation of a clustering solution are presented and discussed.

The article is organized as follows. Section Background provides a background of the clustering problem. Section Methodologies focuses on the clustering methodologies and algorithms. Section Applications of Data Clustering presents different examples of application of the traditional clustering methods to data in different domains. Section Analysis and Assessment describes some relevant criteria for evaluation and assessment of a clustering solution. Finally, Section Closing Remarks draws conclusions about the presented topic.

Background

Clustering belongs to the category of unsupervised learning, whose aim is partitioning unlabeled data into clusters. The data belonging to the same cluster will be “close” to each other, and “far” from the data belonging to different clusters (Rokach and Maimon, 2005). Various proximity criteria can be used for evaluating how “close” the data are. Basically, the choice of a suitable proximity measure mainly depends on the data typology and on the descriptors which are used for representing the data.

Formally, given a collection of data points $\mathcal{X}$, the clustering consists of partitioning them into K disjoint groups $c_1, c_2, \dots, c_K$ such that the union of these groups provides the original collection of data points $\mathcal{X}$, and their intersection is empty:

- $\mathcal{X} = \bigcup_{i=1}^K c_i$,
- $\forall i \neq j, c_i \cap c_j = \emptyset$.

Fig. 1 shows an example of clustering 39 data points into 4 clusters.

In order to cluster a collection of data points, three elements are of prior importance (Ullman *et al.*, n.d.):

1. The proximity measure (similarity, dissimilarity or distance measure),
2. The function for evaluating the quality of the clustering, and
3. The algorithm for clustering computation.

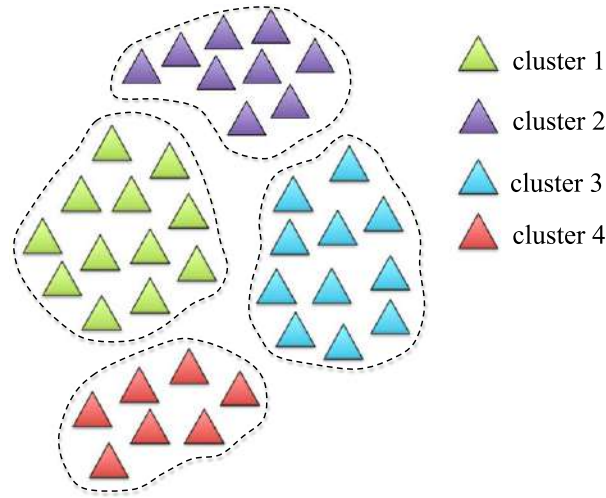


Fig. 1 A sample of clustering 39 data points into 4 clusters. Clusters are delimited by a dashed line and reported in the legend on the right.

In particular, given two data points x_i and x_j , a similarity measure is a proximity measure achieving large values when x_i and x_j are similar. On the contrary, a dissimilarity measure (or a distance measure) is a proximity measure which obtains small values when x_i and x_j are similar (Ullman *et al.*, n.d.). The function for evaluating the quality of the clustering should discriminate between a “good” and a “bad” clustering. Finally, the algorithms which are used for computing the clustering are based on the optimization of the evaluation function.

Different similarity, dissimilarity and distance measures have been introduced in the literature as proximity measures for the clustering problem. In particular, they can be classified as (Ullman *et al.*, n.d.):

- Euclidean, and
- Non-Euclidean.

The Euclidean measures are based on the concept of Euclidean space, which is characterized by a given number of dimensions and “dense” points. The average of two or more points can be evaluated in the Euclidean space and a proximity measure can be computed according to the location of the points in the space. Three Euclidean measures which have been used for clustering in multiple domains are (Ullman *et al.*, n.d.):

- The Euclidean distance,
- The Manhattan distance, and
- The Minkowski distance.

Let $x_i = \{x_i^1, x_i^2, \dots, x_i^n\}$ and $x_j = \{x_j^1, x_j^2, \dots, x_j^n\}$ be two n -dimensional data points. The Euclidean distance between x_i and x_j is defined as follows:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_i^k - x_j^k)^2} \quad (1)$$

An interesting property of the Euclidean distance is that it is invariant to translation. In particular, let $a = \{a^1, a^2, \dots, a^n\}$ be a translation vector. It is observed that the distance in the translated space is equal to the distance computed in the original space:

$$\sqrt{\sum_{k=1}^n ((x_i^k - a^k) - (x_j^k - a^k))^2} = \sqrt{\sum_{k=1}^n (x_i^k - a^k - x_j^k + a^k)^2} \quad (2)$$

The Manhattan (or city block) distance can be considered as an approximate and less expensive version of the Euclidean distance. Given the two n -dimensional data points x_i and x_j , it is defined as follows:

$$d(x_i, x_j) = \sum_{k=1}^n |x_i^k - x_j^k| \quad (3)$$

The generalization of the Euclidean and Manhattan distances is the Minkowski distance, which is defined as:

$$d(x_i, x_j) = \left[\sum_{k=1}^n (x_i^k - x_j^k)^p \right]^{\frac{1}{p}} \quad (4)$$

where p is a positive integer.

Table 1 Overview of some proximity measures adopted in the clustering context

Name	Type	Measure
Euclidean distance	Euclidean	$\sqrt{\sum_{k=1}^n (x_i^k - x_j^k)^2}$
Manhattan distance	Euclidean	$\sum_{k=1}^n x_i^k - x_j^k $
Minkowski distance	Euclidean	$[\sum_{k=1}^n (x_i^k - x_j^k)^p]^{\frac{1}{p}}$
Jaccard distance	Non-Euclidean	$1 - \frac{ x \cap y }{ x \cup y }$
Cosine similarity	Non-Euclidean	$\arccos \frac{x \cdot y}{ x y }$
Edit distance	Non-Euclidean	$ x + y - 2 LCS(x, y) $

The Non-Euclidean measures take into account some specific properties of the data points, which do not include the location of the points in the space. Three Non-Euclidean measures which are well-known in different clustering contexts are (Ullman, n.d.):

- The Jaccard distance,
- The Cosine distance, and
- The Edit distance.

Let x and y be two data points with n elements. The Jaccard distance is defined as follows:

$$d(x, y) = 1 - \frac{|x \cap y|}{|x \cup y|} \quad (5)$$

where $|x \cap y|$ is the size of the intersection and $|x \cup y|$ is the size of the union between the elements in x and the elements in y .

If x and y are considered as two vectors positioned from the origin until their location, the cosine distance is the angle between the two vectors, which is the arccosine of their normalized dot product:

$$d(x, y) = \theta = \arccos \frac{x \cdot y}{|x||y|} \quad (6)$$

Finally, the Edit distance is usually applied when x and y are strings and their elements are the characters composing the strings. It is defined as the number of inserts and deletes which are needed to turn one string into the other one:

$$d(x, y) = |x| + |y| - 2|LCS(x, y)| \quad (7)$$

where $|x|$ and $|y|$ are respectively the size of x and y , and $LCS(x, y)$ is the longest common subsequence of x and y . Table 1 shows an overview of all aforementioned proximity measures.

The evaluation of the clustering quality is generally a hard problem, which includes the two aspects of: (i) cluster compactness, and (ii) cluster isolation (Ullman et al., n.d.). The cluster compactness evaluates the nearness of the data points in a given cluster to its cluster centroid. In the Euclidean space, it is usually computed as the Sum-of-Squared-Error (SSE) measure. In particular, let $c_i = \{x_1, x_2, \dots, x_d\}$ be a cluster of d data points. The SSE computed on c_i is defined as follows:

$$SSE = \sum_{h=1}^d \sum_{k=1}^n (x_h^k - m_i^k)^2 \quad (8)$$

where m_i is the centroid of cluster c_i and m_i^k is the k -th element of the centroid m_i .

The cluster isolation evaluates how far two cluster centroids are in the clustering. In a “good” clustering, the different cluster centroids should be sufficiently distant to each other.

Another important aspect is the number K of clusters in which the data points should be partitioned. Usually, two common strategies can be followed: (i) the number K of clusters is an input parameter of the clustering and it is prior fixed, or (ii) the number K of clusters is dynamically obtained by detecting the best clustering according to the quality evaluation function (Ullman et al., n.d.).

In the next section, we focus on the methods which are currently adopted in the literature for clustering and their applications. Accordingly, we will describe the algorithms which are employed for finding the best clustering and the functions they use for evaluating the clustering quality.

Methodologies

The clustering methods can be partitioned in two broad categories (Xu and Tian, 2015; Ullman et al., n.d.):

- Partitional, and
- Hierarchical.

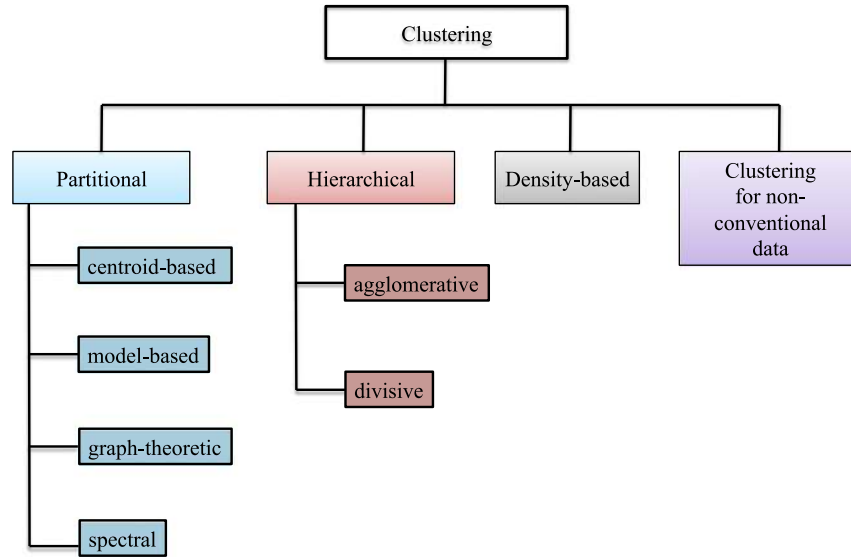


Fig. 2 Overview of the clustering categorization.

The partitional methods include: (i) centroid-based, (ii) model-based, (iii) graph-theoretic, and (iv) spectral approaches. The Hierarchical methods include: (i) divisive, and (ii) agglomerative approaches. Apart from these two categories, we find the density-based clustering methods and the methods for clustering non-conventional data. **Fig. 2** shows an overview of the presented clustering categories.

Partitional Methods

In the partitional methods, the number K of clusters is usually an input parameter. These methods are based on the relocation of the data points which are moved from one cluster to another one, starting from an initial cluster configuration. Because the global optimality cannot be achieved since it would require an exhaustive enumeration of all possible partitions, these methods use greedy heuristics based on iterative optimization (Rokach and Maimon, 2005). Next, we describe some relevant partitional methods and refer to Celebi (2014) for further details.

Centroid-based approaches

The centroid-based approaches are partitional methods characterized by the concept of cluster centroid, which is the central point (e.g. the mean, the median, etc.) computed for the clusters. Initially, the cluster centroids can be randomly selected. Then, an iterative refinement phase changes the centroids' position and the assignment of the data points to the centroids, until a termination condition is satisfied (Xu and Tian, 2015). Two methods belonging to this category are K-Means and K-Medoid.

K-Means. The K-Means algorithm partitions the data points into K non-uniform clusters, where K is an input parameter (MacQueen, 1967). The first step of the algorithm consists of selecting K initial centroids, one for each cluster. It is a critical choice which may influence the final clustering result. In the second step, each data point is assigned to the nearest centroid in terms of the Euclidean distance function. After that, K new centroids are computed as the mean of the data points for each cluster. The second and third steps are iterated until the centroids modify their location. The K-Means finds the K centroids which minimize the following objective function:

$$J = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^{(j)} - m_j\|^2 \quad (9)$$

where n_j is the size of cluster c_j , $x_i^{(j)}$ is the data point x_i belonging to cluster c_j , m_j is the centroid of cluster c_j , and $\|\cdot\|^2$ is the squared Euclidean distance. The time required by the algorithm is $O(I \times K \times M \times n)$, where I is the number of iterations, K is the number of clusters, M is the total number of data points, and n is the dimension of each data point. Because I is usually small and $K \ll M$, the algorithm is linear in the number M of data points (Tan et al., 2005). However, because it computes the mean value, it is particularly sensitive to the outlier data.

K-Medoid. The K-Medoid algorithm is a more robust version of the K-Means to noise and outliers (Kaufman and Rousseeuw, 1987). Instead of using the mean points as centroids, the algorithm selects the centroids among the data points, which are called medoids. The number K of clusters is the input of the algorithm. The first step randomly chooses K data points as the initial medoids. After that, each data point is assigned to the nearest or most similar medoid. The third step randomly selects a data point o to become the new medoid. Then, the total cost S of exchanging o with its old medoid is computed. If S is negative, then the old medoid is exchanged with o . Steps from 2 to 5 are iterated until the medoids change their location. The total cost S is computed as

the difference between the value of the cost function after and before the exchange of o with the medoid. In particular, the cost function J is defined as follows:

$$J = \sum_{j=1}^K \sum_{i=1}^{n_j} |x_i^{(j)} - \bar{m}_j| \quad (10)$$

where n_j is the size of cluster c_j , $x_i^{(j)}$ is the data point x_i belonging to cluster c_j , and $\bar{m}_j$ is the medoid of cluster c_j . Because the mean computation is not required, the algorithm can be used for clustering data points in Non-Euclidean spaces, too. One iteration of the algorithm takes $O(K \times (M - K)^2)$, which may be expensive for large K and M values.

Model-based approaches

The model-based approaches are partitional methods which associate a model with each cluster. The best fit of that model with the data points corresponds to the output of the procedure. These approaches can be partitioned in two sub-categories: (i) statistical learning, and (ii) neural network learning. A typical algorithm of statistical learning is Gaussian Mixture Model (GMM). A typical algorithm of neural network learning is Self-Organizing-Map (SOM) (Xu and Tian, 2015).

GMM. Let $\mathcal{X} = \{x_1, \dots, x_M\}$ be a set of M data points generated by an unknown probability distribution. The number K of clusters is an input parameter. The aim of this algorithm is to estimate the set of parameters θ of the GMM model which best fits the data points (Allili, n.d.). It is accomplished by maximizing the likelihood $p(\mathcal{X} | \theta)$ of the data points given the parameters θ of the model. Accordingly, the problem is defined as follows:

$$\theta^* = \arg \max_{\theta} p(\mathcal{X} | \theta) = \arg \max_{\theta} \prod_{i=1}^M p(x_i | \theta) \quad (11)$$

In order to maximize the likelihood, the Expectation-Maximization (EM) procedure is used. It is based on a hidden variable, which is added to simplify the maximization of the likelihood. The algorithm is composed of a step of expectation, where the distribution of the hidden variable given the data points and the value of the parameters are estimated. It is followed by a step of maximization, where the hidden variable is used for modifying the parameters in order to maximize the likelihood of the data points and of the hidden variable. The GMM algorithm takes $O(M^2 \times K \times I)$, where I is the number of iterations of the algorithm (Xu and Tian, 2015).

SOM. The Kohonen's SOM algorithm computes the output clusters after a learning phase which models an artificial neural network to follow the "shape" of the training data (Kohonen, 1982). The network is characterized by a set of n input neurons, corresponding to the dimension of the input data point, and by a set of K output neurons, which may correspond to the number of clusters. An output neuron j is associated with a n -dimensional vector of weights $w_j = [w_{j1}, \dots, w_{jn}]$, which is called prototype vector. The training of the network starts at the first epoch by initializing all the weights to small random values. Then, a data point x is randomly selected from the training data and given as input to the network. Hence, the output neuron with the nearest prototype vector from x in terms of Euclidean distance, called Best-Matching Unit (BMU) is selected. After that, all the prototype vectors in the neighborhood of BMU are updated by using the following rule:

$$w_j(t+1) = w_j(t) + \eta(t)(x - w_j(t)) \quad (12)$$

where η is the learning rate which monotonically decreases over time. The procedure is iterated by increasing the epoch number, until the computational bounds do not exceed. A new input data point belongs to that cluster which is associated with its BMU. The temporal cost of the SOM algorithm is $O(S^2)$, where S is the number of the map units.

Graph-theoretic approaches

The graph-theoretic approaches compute a partition-based clustering on a graph, where the nodes are the data points and the edges are the relationships among the data points. Weights can be associated with the nodes and/or the links of the graph, representing the similarity between nodes. Two algorithms belonging to this category are CLICK (Cluster Identification by Connectivity Kernels) and MST-based (Minimum Spanning Tree-based) clustering (Xu and Tian, 2015).

CLICK. The CLICK algorithm is characterized by two main steps (Sharan and Shamir, 2000). In the first one, the algorithm recursively splits the nodes of the graph in connected components according to a minimum weight cut criterion. In particular, if the connected component satisfies a given criterion, then it is considered as a kernel and returned. Otherwise, it is split in two parts based on the minimum weight cut criterion. Also, if the connected component is a singleton node, then it is separately managed. In the second step, an adoption procedure is performed, which iteratively finds the most similar pair of singleton and kernel and assigns the singleton to that kernel if their similarity overcomes a predefined threshold. After that, a merging procedure is applied, which iteratively merges pairs of clusters whose similarity is the highest and overcomes a predefined threshold. At the end, the adoption procedure is repeated on the new set of clusters for generating the final clusters. The temporal cost of the algorithm is $O(K \times f(M, E))$, where M is the number of nodes, E is the number of edges, K is the number of clusters, and $f(M, E)$ is the cost of computing a minimum weight cut (Xu and Tian, 2015).

MST-based clustering. A MST-based clustering algorithm generates a Minimum Spanning Tree (MST) from the graph in order to obtain the clusters (Jain and Dubes, 1988). The first step of the algorithm consists of building the MST from the graph of the data points. After that, the inconsistent edges are identified in the MST. Finally, the inconsistent edges are eliminated from the MST in order to keep the connected components representing the clusters. If the procedure is recursively applied on each

obtained cluster, other sub-clusters can be generated. An inconsistent edge can be found by considering if its weight is larger than the average weight of its neighbor edges, the number of standard deviations, or the ratio between its weight and the average weight of its neighbor edges. If the inconsistency factor of the edge is two, then the edge can be eliminated, because it usually connects two clusters. The algorithm takes $O(E \times \log M)$, which is basically the temporal cost of the Kruskal's algorithm for generating the MST (Xu and Tian, 2015).

Spectral approaches

Spectral clustering approaches are partitional methods which are based on the construction of a similarity graph, where the nodes are the data points, and the weighted edges represent the similarity between the nodes. Then, the graph is partitioned such that the total edge weight is as high as possible in each group and as low as possible between the groups. A typical algorithm belonging to this category is NJW (Ng, Jordan and Weiss) (Xu and Tian, 2015).

NJW. The NJW algorithm constructs the matrix $L = D^{-1/2} W D^{-1/2}$, where W is the similarity matrix corresponding to the graph adjacency matrix, and D is a diagonal matrix summing the weights for each node, $D(i) = \sum_j w(i, j)$. Then, the k largest eigenvectors $e_1, e_2, \dots, e_k$ of L are computed and set as the columns of a matrix $X = [e_1 e_2 \dots e_k] \in \mathbb{R}^{M \times k}$, where M is the number of the data points. After normalizing the rows of X to have unit length, the M rows are clustered by using the K-Means algorithm. Finally, each data point is assigned to that cluster which the corresponding X 's row was assigned to Ng et al. (2001). The algorithm can be quite demanding because of the high cost of an eigenvector method (Xu and Tian, 2015).

Hierarchical Methods

The hierarchical methods build a tree, which is called dendrogram, representing the relationships among the data. Then, they use the dendrogram in order to cluster the data. Hierarchical methods are based on two strategies: (i) agglomerative, and (ii) divisive (Tan et al., 2005). Ref. Murtagh and Contreras (2012) presents an overview of the main hierarchical clustering methods.

Agglomerative approaches

An agglomerative approach (or bottom up) starts by creating one cluster for each data point. Then, pairs of clusters are merged at each step, until all data points are grouped into a single cluster. It generates a hierarchy of clusters represented by a dendrogram from which the final clustering is obtained by cutting the dendrogram to a specific level. The merging procedure is performed between the two nearest clusters according to a linkage method. The three most common linkage methods are: (i) single linkage, (ii) complete linkage, and (iii) average linkage. In the single linkage, the distance between two clusters is equivalent to the distance between their two closest data points, one for each cluster. In the complete linkage, the distance between the two farthest data points, one for each cluster, is considered as the distance between the two clusters. Finally, in the average linkage, the distance between two clusters is the average distance of all pairs of data points in the two clusters (Tan et al., 2005). The temporal cost of this approach is $O(M^2 \times \log M)$, where M is the number of the data points (Tan et al., 2005).

Divisive approaches

A divisive approach (or top-down) starts by condensing all the data point into a single cluster. Then, at each step, the clusters are progressively divided in two new clusters, chosen to maximize the inter-cluster distance. The procedure is iterated until each data point becomes a single cluster. The inter-cluster distance can be computed by using similar linkage methods as in the agglomerative clustering: (i) single linkage, (ii) complete linkage, and (iii) average linkage (Forsyth, n.d.). The divisive approach takes at least $O(M^2)$, where M is the number of the data points.

Density-Based Methods

The density-based methods identify the clusters as regions of high-density, which are separated by regions of low-density. Consequently, the number of clusters does not need to be prior fixed. A well-known algorithm belonging to this category is the DBSCAN (Density-based Spatial Clustering of Applications with Noise) (Tan et al., 2005). A survey describing other density-based methods can be found in Loh and Park (2014).

DBSCAN. The DBSCAN algorithm uses a center-based approach for defining the density. In this approach, the density of a given data point is computed as the number of data points within a radius Eps of that point, which includes the point itself. The size Eps of the radius is an input parameter of the algorithm. According to this approach, a data point can be categorized as follows: (i) a core point, inside a dense region, (ii) a border point, on the edge of a dense region, (iii) a noise or background point, inside a region of low-density. The algorithm starts by categorizing all data points as core, border, or noise. All noise data points are discarded. Then, any two core data points which are within an Eps radius to each other are grouped into the same cluster. At the end of this procedure, all core points are separated in different clusters. Finally, each border data point is assigned to its nearest cluster. The algorithm takes $O(M^2)$ in the worst case, where M is the number of the data points (Tan et al., 2005).

Clustering Methods for Non-Conventional Data

In the last years, different methods have been specifically introduced for clustering non-conventional data, whose representation can be unstructured, such as the images, or semi-structured, such as the XML documents.

Image data clustering. Image data are usually represented by matrices of pixels, where each pixel can be characterized by brightness, color, texture, and shape information. According to this representation, different algorithms have been specifically proposed for clustering in this domain. They can solve the image segmentation problem or the image clustering problem. The first problem has the aim of partitioning the image pixels into uniform regions characterizing the objects of the image. The second problem has the aim of categorizing an image database according to the image properties (color, texture, etc.). In this context, Ref. [Shi and Malik \(2000\)](#) has introduced the SM (Shi and Malik) algorithm, which is a spectral approach for solving the image segmentation problem. It builds a similarity graph which is represented by its weighted adjacency matrix W . Each node is a pixel and edges link spatially close pixels. The algorithm runs by solving the system $(D - W)x = \lambda Dx$ for the eigenvectors with the smallest eigenvalues, where D is a diagonal matrix summing the weights for each node, $D(i) = \sum_j w(i, j)$. The eigenvector with the second smallest eigenvalue is used for finding a splitting point of the graph in two parts such that the normalized cut is minimized. Finally, the algorithm is recursively run on each part until the normalized cut exceeds a given threshold. It determines a division of the pixels into image regions. GeNCut (Genetic Normalized Cut) is a genetic graph-based clustering algorithm extending the SM algorithm ([Amelio and Pizzuti, 2014a](#)). Nodes of the graph correspond to image pixels and edges link the most similar image pixels within a spatial neighborhood in the image plan. The similarity is computed according to the brightness difference between the pixels. The fitness function is an extension of the normalized cut criterion, which does not require to prior set the number of clusters. The algorithm has been extended as C-GeNCut (Color-Genetic NCut) in order to include the contribution of color and texture in the similarity computation ([Amelio and Pizzuti, 2013a](#)). Finally, Ref. [Amelio and Pizzuti \(2014b\)](#) introduced GA-IC (Genetic Algorithms Image Clustering), which is a graph-based approach using a genetic algorithm for partitioning an image database. Each node of the graph is an image, while edges link the most similar images. Then, a genetic algorithm is applied on the graph for detecting the node communities which optimize the modularity function. Accordingly, the number of clusters is not an input parameter. The algorithm has been extended as GA-ICDA (Genetic Algorithms Image Clustering for Document Analysis) for partitioning document image databases by introducing the concept of spatial distance between the documents, a final step of clustering refinement, a generalization of the similarity measure, and a procedure for managing singleton clusters. In particular, it has been used for partitioning documents in different languages ([Brodić et al., 2016b](#)), in closely related languages ([Brodić et al., 2015](#)) and in evolving languages ([Brodić et al., 2017](#)). GA-ICDA has been also employed for partitioning documents given in different scripts of the same language ([Brodić et al., 2016a](#)), and in multiple languages and scripts [Brodić et al. \(2016c\)](#). Image and document clustering are two broad research areas which are out the scope of this article. Consequently, we refer to [Anastasiu et al. \(2013\)](#) and [Oikonomakou and Vazirgiannis \(2010\)](#) for further details about document clustering, and to [Ahmed \(2015\)](#) for further details about image clustering.

Semi-structured data clustering. The semi-structured data are typically modeled as tree data. In the last years, it has contributed to the development of different clustering methods for mining the XML data. In particular, Ref. [Costa et al. \(2004\)](#) has proposed a new approach for clustering XML documents, which is based on XML cluster representatives. They are XML documents providing the structural characteristics of a set of XML documents. Clustering is performed by comparison and modification of the cluster representatives as new clusters are found. In [De Meo et al. \(2005\)](#), an approach for clustering heterogeneous XML schema has been proposed. It evaluates the dissimilarity among XML schema by considering interschema semantic properties existing between concepts and builds the dissimilarity matrix to be used by a clustering algorithm. Also, Ref. [Lee et al. \(2002\)](#) has introduced a method for clustering Document Type Definitions (DTD) of XML sources. Semantically and structurally similar DTDs are grouped in the same clusters. Similarity considers not only the linguistic and structural information of the elements in the DTDs, but also the ancestors and descendants of the elements in the DTD trees. Finally, Ref. [Tagarelli and Greco \(2010\)](#) has proposed *SemXClust*, a new method for clustering semantically related XML documents according to their content and structure. XML documents are represented by tree tuples, which are semantically cohesive substructures of the documents represented as XML transactions. Then, two clustering algorithms are proposed to be applied on the XML transactions for clusters detection. An overview of the most recent trends in XML data mining can be found in [Tagarelli \(2011\)](#).

Applications of Data Clustering

Traditional clustering algorithms have been employed in multiple domains as a valid support for detecting classes of data in different contexts, including: (i) image processing, (ii) text mining, (iii) sensor data, (iv) web applications, (v) medical data, and (vi) natural language processing.

Image processing. In image processing, the K-Means algorithm has been used for solving the image segmentation problem, where each pixel is considered as a data point, characterized by color, texture and/or shape features. The algorithm finds groups of pixels corresponding to image regions ([Dhanachandra et al., 2015](#)). Also, Ref. ([Krishnamachari and Abdel-Mottaleb, 1999](#)) has applied an agglomerative hierarchical approach for clustering image collections according to the image content. Users can browse the hierarchy of images by scanning the obtained dendrogram. The Blobworld image retrieval system has applied the EM algorithm for dividing the image pixels into regions which can be used for retrieving similar images from an image query ([Carson et al., 2002](#)).

Text mining. In the text and document mining context, Ref. Wang *et al.* (2011) has applied an agglomerative hierarchical approach for clustering collections of text documents. Clustering is performed on a graph where the nodes are the words representing a description of the documents, and the edges are the relationships between the words. Also, in Bide and Shedge (2015) the K-Means algorithm has been used for clustering a collection of documents by following a Divide and Conquer strategy which divides the documents in smaller groups. Still, Ref. Li and Huang (2010) has analyzed the ability of K-Medoid and DBSCAN algorithms in clustering collections of text documents according to their content. Results demonstrate that DBSCAN performs considerably better than K-Medoid which is sensitive to the initial setting of the centroids. In Sarnovsky and Carnoka (2016), a distributed K-Means algorithm using Jbowl text mining library and GridGain framework has been designed and implemented for clustering collections of text documents.

Sensor data. Clustering has also been adopted for partitioning data generated by instrumentation measuring physical quantities. In Brodić and Amelio (2015), the K-Means algorithm has been applied on magnetic field data measured in the neighborhood of the laptop computers in order to detect the emission levels to which the laptop's users are exposed. The experiment has been extended to cases when the laptop is overloaded with heavy applications (under stress condition) Brodić and Amelio (2016). A similar experiment has been performed on magnetic field data measured in the neighborhood of the laptop's AC adapters for detecting their emission levels Brodić and Amelio (2017). In the same context, the SOM algorithm has been employed for evaluating the ability of an artificial neural network model in predicting the magnetic field levels from known laptop characteristics (Brodić *et al.*, 2016).

Web applications. In the web context, the K-Means algorithm has been used for clustering the search results in order to improve the browsing of the content Poomagal and Hamsapriya (2011). The web documents obtained by the search are partitioned into clusters according to the extracted words and their frequency. Also Ref. de Paiva and Costa (2012) has proposed the application of the SOM algorithm to web log files in order to detect the similarities between users' sessions. It has the aim to identify users presenting similar interests in order to create personalized navigation environments. In Morichetta *et al.* (2016), the DBSCAN has been employed for clustering the web URLs in order to identify suspicious traffic on the web.

Medical data. Clustering in the medical context has been used for supporting the diagnosis process. In particular, Ref. Albayrak (2003) has experimented the application of K-Means and SOM algorithms on thyroid gland data, in order to detect normal, hyperthyroid and hypothyroid functions. Also, in Amelio and Pizzuti (2013b) the C-GeNCut algorithm has been applied on dermatological image repositories for the extraction of skin lesions from the background, e.g. melanoma images. In Sharan and Shamir (2000), the CLICK algorithm has been used in the biological field for clustering genes in groups with similar expression patterns.

Natural language processing. In natural language processing, an approach for building a multilingual web directory based on the application of SOM algorithm has been introduced (Yang *et al.*, 2011). A network is used for learning each set of monolingual web pages. Then, a hierarchy generation and alignment process is applied on the networks to build the final multilingual hierarchy. Also, Ref. Kumar *et al.* (2011) has proposed the application of the bisecting K-Means for clustering multilingual documents according to the document similarities, where linguistic elements from Wikipedia are used for improving the document representation.

Analysis and Assessment

Different criteria have been introduced for evaluating the quality of a clustering solution. In particular, they can be divided in two main categories: (i) internal criteria, and (ii) external criteria (Béjar, n.d.). Internal criteria evaluate the clustering according to the quality of the partition. External criteria compare the clustering solution with the labeled data, also called ground-truth partitioning. In the following, we describe some criteria belonging to each category. Other evaluation criteria are described and analyzed in Halkidi *et al.* (2001).

Internal Criteria

Three known internal criteria for clustering evaluation are: (i) Silhouette coefficient, (ii) Davies-Bouldin index, and (iii) Ball-Hall index.

The Silhouette coefficient evaluates the clustering quality in terms of separation between the clusters and cohesion inside the clusters. For the i -th data point, it computes its average distance with the other data points in its cluster. It is denoted as a_i . Then, for any other cluster, it computes the average distance with its data points and finds the minimum distance over the clusters. It is denoted as b_i . Accordingly, the Silhouette coefficient is defined as follows (Desgraupes, n.d.):

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (13)$$

The value of the Silhouette coefficient ranges between -1 ($a_i > b_i$, worst case) and 1 ($a_i = 0$, best case).

The Davies-Bouldin index quantifies the average similarity between each cluster and its most similar cluster. Because a better solution corresponds to cohesive and well-separated clusters, a lower index value indicates a better clustering. In particular, let δ_k be the average distance between the data points belonging to cluster c_k and its centroid m_k :

$$\delta_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \|x_i^{(k)} - m_k\| \quad (14)$$

Also, let $\Delta_{kk'}$ be the distance between the centroids m_k and $m_{k'}$ of the clusters c_k and $c_{k'}$:

$$\Delta_{kk'} = \|m_k - m_{k'}\| \quad (15)$$

The Davis-Bouldin index is defined as (Desgraupes, n.d.):

$$\mathcal{D} = \frac{1}{K} \sum_{k=1}^K \max_{k' \neq k} \left(\frac{\delta_k + \delta_{k'}}{\Delta_{kk'}} \right) \quad (16)$$

The Ball-Hall index is the average of the squared distances between the data points of each cluster and its centroid (Desgraupes, n.d.). It is defined as follows:

$$\mathcal{B} = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} \sum_{i=1}^{n_k} \|x_i^{(k)} - m_k\|^2 \quad (17)$$

External Criteria

Three external criteria for clustering evaluation are: (i) Normalized Mutual Information, (ii) Adjusted Rand Index, and (iii) Jaccard index.

Let $C = \{c_1, c_2, \dots, c_K\}$ be a clustering solution composed of K clusters, and $P = \{p_1, p_2, \dots, p_K\}$ be the ground-truth partitioning in clusters. The overlap between C and P can be represented by the contingency table CM shown in Table 2 (Yeung and Ruzzo, n.d.):

According to the contingency table CM , the Normalized Mutual Information (NMI) can be computed as follows:

$$NMI = \frac{\sum_{ij} n_{ij} \log \frac{M n_{ij}}{n_i n_j}}{\sqrt{\left(\sum_i n_i \log \frac{n_i}{M} \right) \left(\sum_j n_j \log \frac{n_j}{M} \right)}} \quad (18)$$

where M is the total number of data points. The NMI is bounded between 0 (perfect mismatch) and 1 (perfect match). It represents the amount of information given by the clusters and useful to derive the membership of the data points in the ground-truth clusters (Vinh et al., 2010). The main limitation of this measure is the selection bias problem, which is the tendency to obtain higher values for clustering solutions with many clusters. Recently, an adjustment of the NMI has been introduced, which scales the value of the NMI by a scaling factor. It is computationally not expensive and avoids the selection bias problem by penalizing the clustering solutions with a very low or very high number of clusters (Amelio and Pizzuti, 2016).

Other measures compare C and P according to the pairs of data points which are in the same or different clusters. In particular, the Adjusted Rand Index (ARI) is defined as (Yeung and Ruzzo, n.d.):

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}} \quad (19)$$

It varies between -1 and 1 , and takes 0 for random partitioning and 1 for perfect agreement.

Finally, the Jaccard index is similar to ARI, with the difference that it does not consider the pairs of data points which are in different clusters. The Jaccard index is the following (Wagner and Wagner, n.d.):

$$\mathcal{J} = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \quad (20)$$

where n_{11} is the number of pairs which are in the same cluster in C and P , n_{10} is the number of pairs which are in the same cluster in C and in different clusters in P , and n_{01} is the number of pairs which are in different clusters in C and in the same cluster in P .

Table 2 Contingency table CM

C/P	p_1	p_2	...	p_K	Sums
c_1	n_{11}	n_{12}	...	n_{1K}	a_1
c_2	n_{21}	n_{22}	...	n_{2K}	a_2
...	...	...	...	...	...
c_K	n_{K1}	n_{K2}	...	n_{KK}	a_K
Sums	b_1	b_2	...	b_K	

Source: Modified from Yeung, K.Y. & Ruzzo, W. L., n.d. Details of the adjusted rand index and clustering algorithms supplement to the paper an empirical study on principal component analysis for clustering gene expression data (to appear in bioinformatics). Available at: <http://faculty.washington.edu/kayee/pca/supp.pdf>.

Closing Remarks

Clustering is the procedure of partitioning data into homogeneous groups such that data belonging to the same group are similar and data belonging to different groups are dissimilar. This article presented the main elements characterizing the clustering problem. It was realized by describing the most common measures for similarity or distance evaluation in a clustering approach, the most relevant clustering methodologies, and some important criteria for assessing the clustering solution. In particular, two broad categories of clustering approaches, partitional and hierarchical, were described. Also, density-based clustering approaches were discussed. Finally, some clustering approaches specifically designed for non-conventional data were presented. The focus of this work was on the methodological aspects of clustering, with a section dedicated to some application cases which contextualize the methodology. It can be of particular interest to students who approach for the first time to the clustering problem, as well as scientists, in academia and research institutes, as a valid support for designing new clustering methods advancing the proposed clustering methodologies in the state-of-the-art.

References

- Ahmed, N., 2015. Recent review on image clustering. *IET Image Processing* 9 (11), 1020–1032.
- Albayrak, S., 2003. Unsupervised Clustering Methods for Medical Data: An Application to Thyroid Gland Data. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 695–701.
- Allili, M., n.d. A short tutorial on gaussian mixture models. Available at: <http://www.computerrobotvision.org/2010/tutorialDay/GMMsaicrv1tutorial.pdf>.
- Amelio, A., Pizzuti, C., 2013a. A Genetic Algorithm for Color Image Segmentation. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 314–323.
- Amelio, A., Pizzuti, C., 2013b. Skin lesion image segmentation using a color genetic algorithm. In: *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '13 Companion*, ACM, New York, NY, USA, pp. 1471–1478.
- Amelio, A., Pizzuti, C., 2014a. An evolutionary approach for image segmentation. *Evolutionary Computation* 22 (4), 525–557.
- Amelio, A., Pizzuti, C., 2014b. A New Evolutionary-Based Clustering Framework for Image Databases. Cham: Springer International Publishing, pp. 322–331.
- Amelio, A., Pizzuti, C., 2016. Correction for closeness: Adjusting normalized mutual information measure for clustering comparison. *Computational Intelligence*. Available at: <https://doi.org/10.1111/coin.122100>.
- Anastasiu, D.C., Tagarelli, A., Karypis, G., 2013. Document clustering: The next frontier. In: *Data Clustering: Algorithms and Applications*. pp. 305–338.
- Anderberg, M.R., 1973. *Cluster Analysis for Applications*. Academic Press.
- Bailey, K., 1994. Typologies and Taxonomies: An Introduction to Classification Techniques. In: 'Quantitative Applications in t'. No. 102. SAGE Publications.
- Béjar, J., n.d. Clustering evaluation/model assessment. Available at: <http://www.cs.upc.edu/bejar/amlt/material/04-Validation.pdf>.
- Bide, P., Shedje, R., 2015. Improved document clustering using k-means algorithm. In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–5.
- Brodić, D., Amelio, A., 2015. Classification of the extremely low frequency magnetic field radiation measurement from the laptop computers. *Measurement Science Review* 15 (4), 202–209.
- Brodić, D., Amelio, A., 2016. Detecting of the extremely low frequency magnetic field ranges for laptop in normal operating condition or under stress. *Measurement* 91, 318–341.
- Brodić, D., Tanik, D., Amelio, A., 2016. An approach to evaluation of the extremely low-frequency magnetic field radiation in the laptop computer neighborhood by artificial neural networks. *Neural Computing and Applications*. 1–13.
- Brodić, D., Amelio, A., Milivojevic, Z.N., 2016a. Identification of fraktur and latin scripts in german historical documents using image texture analysis. *Applied Artificial Intelligence* 30 (5), 379–395.
- Brodić, D., Amelio, A., Milivojevic, Z.N., 2016b. Language discrimination by texture analysis of the image corresponding to the text. *Neural Computing and Applications*.
- Brodić, D., Amelio, A., Milivojevic, Z.N., 2016c. An approach to the language discrimination in different scripts using adjacent local binary pattern, *Journal of Experimental & Theoretical Artificial Intelligence* 29 (5), 1–19.
- Brodić, D., Amelio, A., 2017. Range detection of the extremely low-frequency magnetic field produced by laptop's ac adapter. *Measurement Science Review* 17 (1), 1–8.
- Brodić, D., Amelio, A., Milivojevic, Z.N., 2017. Clustering documents in evolving languages by image texture analysis. *Applied Intelligence* 46 (4), 916–933.
- Carson, C., Belongie, S., Greenspan, H., Malik, J., 2002. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (8), 1026–1038.
- Cattell, R., 1943. *The Description of Personality: Basic Traits Resolved Into Clusters*. American psychological association.
- Celebi, M.E., 2014. *Partitional Clustering Algorithms*, Springer Publishing Company, Incorporated.
- Costa, G., Manco, G., Ortale, R., Tagarelli, A., 2004. A tree-based approach to clustering xml documents by structure. In: *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD '04*, pp. 137–148. New York, NY, USA: Springer-Verlag New York, Inc.
- De Meo, P., Quattrone, G., Terracina, G., Ursino, D., 2005. An Approach for Clustering Semantically Heterogeneous XML Schemas. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 329–346.
- de Paiva, F.A.P., Costa, J.A.F., 2012. Using SOM to Clustering of Web Sessions Extracted by Techniques of Web Usage Mining. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 484–491.
- Desgraupes, B., n.d. Clustering indices, Available at: <https://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf>.
- Dhanachandra, N., Manglem, K., Chanu, Y.J., 2015. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm, *Procedia Computer Science* 54, 764–771; Eleventh International Conference on Communication Networks, ICCN 2015, August 21–23, 2015, Bangalore, India; Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21–23, 2015, Bangalore, India; Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21–23, 2015, Bangalore, India.
- Forsyth, D., n.d. Clustering. Available at: <http://luthuli.cs.uiuc.edu/daf/courses/probcourse/notesclustering.pdf>.
- Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. *Journal of Intelligent Information Systems* 17 (2), 107–145.
- Jain, A.K., Dubes, R.C., 1988. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- Jardine, N., Sibson, R., 1971. *Mathematical Taxonomy*, Wiley Series in Probability and Mathematical statistics. Wiley.
- Kaufman, L., Rousseeuw, P., 1987. Clustering by Means of Medoids, Reports of the Faculty of Mathematics and Informatics, Faculty of Mathematics and Informatics. Available at: <https://books.google.it/books?id=HK-4GwAACAAJ>.
- Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43 (1), 59–69.
- Krippendorff, K., 1980. In: *Clustering*, P.R. Monge, Cappella, J.N. (Eds.), *Multivariate Techniques in Human Communication Research*. New York, NY: Academic Press, pp. 259–308.

- Krishnamachari, S., Abdel-Mottaleb, M., 1999. Image browsing using hierarchical clustering. In: *Proceedings of the IEEE International Symposium on Computers and Communications (Cat. No. PR00250)*, pp. 301–307.
- Kumar, N.K., Santosh, G.S.K., Varma, V., 2011. Effectively Mining Wikipedia for Clustering Multilingual Documents. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 254–257.
- Lee, M.L., Yang, L.H., Hsu, W., Yang, X., 2002. Xclust: Clustering xml schemas for effective integration. In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pp. 292–299. New York, NY, USA: ACM.
- Lerman, I., 1970. *Les bases de la classification automatique*. Collection Programmation. Gauthier-Villars.
- Li, Q., Huang, X., 2010. Research on text clustering algorithms. In: *2010 2nd International Workshop on Database Technology and Applications*, pp. 1–3.
- Loh, W.-K., Park, Y.-H., 2014. *A Survey on Density-Based Clustering Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 775–780.
- MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1: Statistics, pp. 281–297. Berkeley, CA: University of California Press.
- Morichetta, A., Bocchi, E., Metwalley, H., Mellia, M., 2016. Clue: Clustering for mining web urls. 2016 28th International Teletraffic Congress (ITC 28) 01, 286–294.
- Murtagh, F., Contreras, P., 2012. Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2 (1), 86–97.
- Ng, A.Y., Jordan, M.I., Weiss, Y., 2001. On spectral clustering: Analysis and an algorithm. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 84–856. Cambridge, MA, USA: MIT Press.
- Oikonomakou, N., Vazirgiannis, M., 2010. A review of web document clustering approaches. In: *Data Mining and Knowledge Discovery Handbook*, second ed. pp. 931–948.
- Poomagal, S., Hamsapriya, T., 2011. K-means for search results clustering using url and tag contents. In: *2011 International Conference on Process Automation, Control and Computing*, pp. 1–7.
- Rokach, L., Maimon, O., 2005. Clustering methods. In: Maimon, O., Rokach, L. (Eds.), *The Data Mining and Knowledge Discovery Handbook*. Springer, pp. 321–352.
- Sarnovsky, M., Carnoka, N., 2016. Distributed Algorithm for Text Documents Clustering Based on k-Means Approach. Cham: Springer International Publishing, pp. 165–174.
- Sharan, R., Shamir, R., 2000. Center CLICK: A clustering algorithm with applications to gene expression analysis. In: Bourne, P.E., Gribskov, M., Altman, R.B., *et al.* (Eds.), *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, August 19–23, La Jolla/San Diego, CA, USA. AAAI, pp. 30–316. Available at: <http://www.aaai.org/Library/ISMB/2000/ismb00-032.php>.
- Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8), 888–905.
- Sokal, R.R., 1963. The principles and practice of numerical taxonomy. *Taxon* 12 (5), 190–199.
- Tagarelli, A., Greco, S., 2010. Semantic clustering of xml documents. *ACM Transactions on Information Systems* 28 (1), 3:1–3:56.
- Tagarelli, A., 2011. *XML Data Mining: Models, Methods, and Applications*, first ed. Hershey, PA, USA: IGI Global.
- Tan, P.-N., Steinbach, M., Kumar, V., 2005. *Introduction to Data Mining*, first ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Tryon, R., 1939. *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality*. Edwards brother, Incorporated, lithoprinters and publishers.
- Ullman, J.D., n.d. Clustering. Available at: <http://infolab.stanford.edu/ullman/mining/pdf/cs345-cl.pdf>.
- Ullman, S., Poggio, T., Harari, D., Zysman, D., Seibert, D., n.d. Unsupervised Learning, Clustering. Available at: <http://www.mit.edu/9.54/fall14/slides/Class13.pdf>.
- Vinh, N.X., Epps, J., Bailey, J., 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* 11, 2837–2854.
- Wagner, S., Wagner, D., n.d. Comparing Clusterings – An Overview. Available at: <https://publikationen.bibliothek.kit.edu/1000011477/812079>.
- Wang, Y., Ni, X., Sun, J.-T., Tong, Y., Chen, Z., 2011. Representing document as dependency graph for document clustering. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pp. 2177–2180. New York, NY, USA: ACM.
- Xu, D., Tian, Y., 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science* 2 (2), 165–193.
- Yang, H.-C., Hsiao, H.-W., Lee, C.-H., 2011. Multilingual document mining and navigation using self-organizing maps. *Information Processing & Management* 47 (5), 647–666. (Managing and Mining Multilingual Documents).
- Yeung, K.Y., Ruzzo, W.L., n.d. Details of the adjusted rand index and clustering algorithms supplement to the paper an empirical study on principal component analysis for clustering gene expression data (to appear in bioinformatics). Available at: <http://faculty.washington.edu/kayee/pca/supp.pdf>.

Further Reading

- Aggarwal, C.C., 2015. *Data Mining – The Textbook*. Springer International Publishing.
- Ambaye, M.S., 2009. *Effectiveness of Content-Based Image Clustering Algorithms: Measuring Cluster Quality*. VDM Publishing.
- Gan, G., Ma, C., Wu, J., 2007. *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Han, J., Kamber, M., Pei, J., 2011. *Data Mining: Concepts and Techniques*, The Morgan Kaufmann Series in Data Management Systems, third ed. Morgan Kaufmann Publishers.
- Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., Ponce Leon, A.C., de Carvalho, F., 2009. A Survey of Evolutionary Algorithms for Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 39 (2), 133–155.
- Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. *ACM Computing Surveys* 31 (3), 264–323.
- Witten, I.H., Frank, E., Hall, M.A., 2011. *Data Mining: Practical Machine Learning Tools and Techniques*, The Morgan Kaufmann Series in Data Management Systems, third ed. Elsevier.
- Zaki, M.J., Meira Jr., W., 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press.

Biographical Sketch



Alessia Amelio received her BSc and MSc magna cum laude in Computer Science Engineering from University of Calabria in 2005 and 2009, as well as Ph.D. in computer science engineering and systems from the Faculty of Engineering, University of Calabria in 2013. During her Ph.D., she was visiting research scholar at College of Computing, Georgia Institute of Technology, under the supervision of prof. Alberto Apostolico. From 2011 to 2014 she was research fellow at the National Research Council of Italy. From 2015 to 2016 she was researcher at the National Research Council of Italy. Now she is research fellow of computer science at the Department of Computer Science Engineering, Modeling, Electronics and Systems, University of Calabria, Italy. Her current research interests include different aspects of image processing, document classification, pattern recognition from sensor data, social network analysis, data mining and artificial intelligence methods for the web. She co-authored more than 20 journal papers indexed in Scopus and Web of Science, and more than 40 conference papers, book chapters and magazine papers. She was chair and co-organizer of two invited sessions as well as session chair in different international conferences. She has also served as a reviewer as well as a member of program committee for leading journals and conferences in the fields of data mining, knowledge and data engineering, artificial intelligence, and physics.



Andrea Tagarelli is an assistant professor of computer engineering at the University of Calabria, Italy. He graduated magna cum laude in computer engineering, in 2001, and obtained his Ph.D. in computer and systems engineering, in 2006. He was research fellow at the Department of Computer Science & Engineering, University of Minnesota at Minneapolis, United States, working in the George Karypis's Data Mining Lab, in 2007. In 2013, he obtained the Italian national scientific qualification to associate professor, and in 2017 the qualification to full professor, for the computer science and engineering research area (scientific disciplinary sector ING-INF/05). His research interests include topics in data mining, web and network science, information retrieval, artificial intelligence. On these topics, he has coauthored more than 100 peer-reviewed papers, including journal articles, conference papers and book chapters. He also edited a book titled XML Data Mining: Models, Methods, and Applications. He was co-organizer of three workshops and a mini-symposium on data clustering topics in premier conferences in the field (ACM SIGKDD, SIAM DM, PAKDD, ECML-PKDD). He has also served as a reviewer as well as a member of program committee for leading journals and conferences in the fields of databases and data mining, knowledge and data engineering, network analysis, information systems, knowledge based systems, and artificial intelligence. Since 2015, he has been in the editorial board of Computational Intelligence Journal and Social Network Analysis and Mining Journal.

Computation Cluster Validation in the Big Data Era

Raffaele Giancarlo, University of Palermo, Palermo, Italy

Filippo Utro, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

One of the main tasks of exploratory data analysis, and a common practice in statistics, is the grouping of items in a set in such a way that items in the same group (called a cluster) are more similar to each other than to those in other clusters.

The most fundamental issue to be addressed when clustering data consists of the determination of the number of clusters. Related issues are how to assign confidence levels to the selected number of clusters, as well as to the induced cluster assignments. Those issues are particularly important in data analysis and become quite difficult when the data available offer a relatively small sample size and are of very high dimensionality, making the clustering results especially sensitive to noise and susceptible to overfitting. Many data of interest to the Life Sciences exhibit those features, for example, microarrays.

In this article, we start by giving a formal definition of the basic issues concerning cluster analysis. Then, we present some related notions from Hypothesis Testing in Statistics.

Finally, we present a case study on microarray data, trying to answer to two fundamental questions: (1) What is the precision of a method, i.e., its ability to predict the correct number of clusters in a dataset? (2) Among a collection of methods, which is more accurate, less algorithm dependent, etc.?

It is to be remarked that although some of the methodologies described here have been validated only on biological data, namely microarrays, they are generic in the sense that they can be applied to any kind of data. Moreover, it is relevant to mention here that recently they are becoming quite popular in domains other than microarray data analysis (e.g., [Giroto et al. \(2016\)](#), [McMurdie and Holmes \(2013\)](#), and [Utro et al. \(2012\)](#)).

Background

The aim of cluster analysis is to divide data into groups (i.e., clusters) that are meaningful and lead to significant insights about the data. The partition of the data into clusters is based on information present in the data itself. Formally, consider a set of n items $(\Sigma) = \{\sigma_1, \dots, \sigma_n\}$, where σ_i is specified by m numeric values, referred to as features or conditions, for each integer $i \in [1, n]$. That is, each σ_i is an element in m -dimensional space. Let $C_k = \{c_1, c_2, \dots, c_k\}$ be a partition of (Σ) into k clusters, i.e., a set of subsets of (Σ) such that $\cup_{i=1}^k c_i = (\Sigma)$ and $c_i \cap c_j = \emptyset$ for $1 \leq i \neq j \leq k$. Each subset c_i , where $1 \leq i \leq k$, is referred to as a cluster, and C_k is referred to as a clustering solution. The aim of cluster analysis is to determine a partition of (Σ) according to a similarity/distance S , which is referred to as similarity/distance metric. It is defined on the elements in (Σ) . In particular, one wants that items in the same cluster to have “maximal similarity”, while items in different clusters are “dissimilar”. Typically, (Σ) is represented in one of two different ways: (1) a data matrix D , of size $n \times m$, in which the rows represent the items and the columns represent the feature values; (2) a similarity/dissimilarity matrix S , of size $n \times n$, in which each entry S_{ij} , with $1 \leq i \neq j \leq n$, is the value of similarity/dissimilarity of σ_i and σ_j . Particularly, the value of S_{ij} can be computed using rows i and j of D . Hence, S can be derived from D , but not vice versa. The specification and formalization of a similarity metric, via mathematical functions, depends heavily on the application domain and it is one of the key steps in clustering, in particular in the case of biological data (e.g., microarray). The state of the art, as well as some relevant progress in the identification of good distance functions for microarrays, is presented in [Giancarlo et al. \(2013\)](#), [Sera et al. \(2016\)](#) and references therein.

Typically, the partition of (Σ) into groups is accomplished via a clustering algorithm A . A classical classification of clustering algorithms is hierarchical versus partitional. The hierarchical clustering algorithms produce a nested sequence of partitions ([Jain and Dubes, 1988](#)). The partitional clustering algorithms directly decompose the dataset into a partition C_k . In this article, we limit ourselves to the class of clustering algorithms that take in input D and an integer k and return C_k .

Assessment of Cluster Quality: Main Problems Statement

Assume that one is given a set of genes. Then, a sensible biological question would be to find out how many functional groups of genes are present in a dataset. Since the presence of “statistically significant patterns” in the data is usually an indication of their biological relevance ([Leung et al., 1996](#)), it makes sense to ask whether a division of the items into groups is statistically significant. In what follows, the three problem statements in which that question can be cast ([Jain and Dubes, 1988](#)) are detailed. Let $\bar{C}_j$ be a reference classification for (Σ) consisting of j classes. That is, $\bar{C}_j$ may either be a partition of (Σ) into j groups, usually referred to as the gold standard, or a division of the universe generating (Σ) into j categories, usually referred to as class labels. An external index E is a function that takes as input a reference classification $\bar{C}_j$ for (Σ) and a partition C_k of (Σ) and returns a value assessing how close the partition is to the reference classification. It is external because the quality assessment of the partition is established via criteria external to the data, i.e., the reference classification. Notice that it is not required that $j=k$. An internal index I is a function defined

on the set of all possible partitions of (Σ) and with values in $\mathbb{R}$. It should measure the quality of a partition according to some suitable criteria. It is internal because the quality of the partition is measured according to information contained in the dataset without resorting to external knowledge. The first two problems are:

- (a) Given $\bar{C}_j$, C_k and E , measure how far is C_k from C_{k^*} according to E . That is, we are examining if k is the number of clusters one expects in D .
- (b) Given C_k and I , establish whether the value of I computed on C_k is unusual and therefore surprising. That is, significantly small or large.

We observe that these two problems try to assess the quality/goodness of a clustering solution C_k consisting of k groups, but give no indication whether k is the “right number” of clusters. In order to get such an indication, we are interested in the following:

- (c) Assume we are given: (a) A sequence of clustering solutions $C_1, \dots, C_s$, obtained for instance via repeated application of a clustering algorithm A ; (b) a function R , usually referred to as a relative index that estimates the relative merits of a set of clustering solutions. We are interested in identifying the partition C_{k^*} among the ones given in (a) providing the best value of R . Let k^* be the optimal number of clusters according to R .

The clustering literature is extremely rich in mathematical functions suited for the three problems outlined above (Handl et al., 2005). The crux of the matter is to establish quantitatively the threshold values allowing us to say that the value of an index is significant enough. That naturally brings us to briefly mention Hypothesis Testing in Statistics, from which one can develop methods to assess the statistical significance of an index.

Assessment of Cluster Significance: A Generic Procedure

A statistic T is a function that captures helpful information about the data, i.e., it can be one of the indices mentioned earlier. Formally, it is a random variable and its distribution describes the relative frequency with which values of T occur, on which, one implicitly assumes that there is a background/reference probability distribution for its values. That implies the existence of a sample space. A hypothesis is a statement about the frequency of events in the sample space. It is tested by observing a value of T and by deciding how unusual it is, according to the probability distribution one is assuming for the sample space. The most common hypothesis tested for in clustering is the null hypothesis, H_0 : *there is no structure in the data* (i.e., $k=1$). Testing for H_0 with a statistic T in a dataset D means to compute T on D and then decide whether to reject or not H_0 . Therefore, one needs to establish how significant is the value found with respect to a background probability distribution of the statistic T under H_0 via a formalization of the notion of “no structure” or “randomness” in the data. Among the many possible ways, commonly indicated to as null models, the most relevant proposed in the clustering literature are detailed in Bock (1985), Gordon (1996), Jain and Dubes (1988), and Sarle (1983).

After a null model has been selected, one would need to gain formulas giving the value of T under the null model and for a specific set of parameters. Disappointingly, since not too many such formulae exist, one typically resorts to a Monte Carlo simulation applied to the context of assessing the significance of C_k .

We anticipate that Section Internal indices provides methods that resort to the same principles and guidelines mentioned above, although they are more specific about the statistic that is relevant to identify the number of clusters in a dataset.

Fundamental Validation Indices

In this section, some essential validation methods are introduced. In particular, external and internal indices are outlined. In the literature concerning this topic, the terms index and measure are interchangeable. The external and internal indices differ fundamentally in their aims, and find application in distinct experimental settings. Recalling the previous section, external indices can be used to validate clustering algorithms, while internal measures can be used as data driven techniques to estimate the number of cluster in a data set.

We start by presenting three external indices that assess the agreement between two partitions. Then, we present the most prominent internal measures useful to estimate the correct number of clusters present in a dataset, and that fall into one of the following paradigms: (1) compactness (2) hypothesis testing in statistics; (3) stability-based techniques and (4) jackknife techniques.

External indices

As already mentioned, an external measure is a function that takes as input two partitions C_j and C_k and returns a value assessing how close C_k is to C_j . It is external because the quality assessment of the partition is established via criteria external to the data. Notice that it is not required that $j=k$. Three most prominent external measures known in the literature are: the Adjusted Rand Index (Hubert and Arabi, 1985), the F-index (Van Rijsbergen, 1979) and the Fowlkes and Mallows Index (Fowlkes and Mallows, 1983). All three can be computed via a contingency table described below, however, for brevity only the Adjusted Rand Index is considered here.

Given two partitions $\bar{C}_r$ and C_{k^*} with the notation of Section Assessment of Cluster Quality: Main Problems Statement, $\bar{C}_r$ is an external partition of the items, derived from the reference classification, while C_{k^*} is a partition obtained by some clustering

method. Let $n_{i,j}$ be the number of items in both $\bar{c}_i$ and c_j , $1 \leq i \leq r$ and $1 \leq j \leq t$. Moreover, let $|\bar{c}_i| = n_{i,*}$ and $|c_j| = n_{*,j}$. Those values can be conveniently arranged in a contingency table:

Class/Cluster	c_1	c_2	...	c_k	Sums
$\bar{c}_1$	$n_{1,1}$	$n_{1,2}$	...	$n_{1,k}$	$n_{1,*}$
$\bar{c}_2$	$n_{2,1}$	$n_{2,2}$	...	$n_{2,k}$	$n_{2,*}$
...	...	...	...	...	...
$\bar{c}_r$	$n_{r,1}$	$n_{r,2}$	...	$n_{r,k}$	$n_{r,*}$
Sums	$n_{*,1}$	$n_{*,2}$	...	$n_{*,k}$	$n_{*,*} = n$

Adjusted Rand Index R_A is derived from a generalization of a hypergeometric distribution as the null hypothesis. That is, it is assumed that the row and column sums in the contingency table are fixed, but the two partitions are picked at random. Formally, it is defined as:

$$R_A = \frac{\sum_{i,j} \binom{n_{i,j}}{2} - \frac{\sum_i \binom{n_{i,*}}{2} \sum_j \binom{n_{*,j}}{2}}{\binom{n}{2}}}{\frac{1}{2} \left[\sum_i \binom{n_{i,*}}{2} + \sum_j \binom{n_{*,j}}{2} \right] - \frac{\sum_i \binom{n_{i,*}}{2} \sum_j \binom{n_{*,j}}{2}}{\binom{n}{2}}}$$

R_A assumes values in the interval $[-\infty, 1]$. It assumes value one, when there is a perfect agreement between the two partitions, while its expected value of zero indicates a level of agreement due to chance. Therefore, the two partitions are in significant agreement if R_A assumes a non-negative value, substantially away from zero.

Internal indices

Internal indices assess the quality of a partition, without any use of external information. Then, a Monte Carlo simulation can determine if the value of such an index on the given partition is unusual enough for the user to gain confidence that the partition is good. Internal indices are also an essential building block in order to obtain relative indices that support to select, among a given set of partitions, the “best” one. For the state of the art on internal measures, the reader is referred to [Giancarlo et al. \(2008\)](#), [Handl et al. \(2005\)](#), and [Giancarlo and Utró \(2012\)](#). Some of the most prominent internal measures are based on: (1) compactness (2) hypothesis testing in statistics; (3) stability-based techniques and (4) jackknife techniques.

This also gives a natural division of the most prominent measures in the literature:

- Within Clusters Sum of Square (WCSS for short) ([Hastie et al., 2003](#)) and Krzanowski and Lai Index (KL for short) ([Krzanowski and Lai, 1985](#)).
- Gap Statistics (Gap for short) ([Tibshirani et al., 2001](#)).
- CLEST ([Dudoit and Fridlyand, 2002](#)), Model Explorer (ME for short) ([Ben-Hur et al., 2002](#)), Consensus Clustering (Consensus for short) ([Monti et al., 2003](#)).
- Figure of Merit (FOM for short) ([Yeung et al., 2001](#)).

We are selecting only one internal measure from each of the mentioned classes and highlight here a few key facts about them, referring the interested reader to [Giancarlo et al. \(2008\)](#) for a more in-depth presentation of each of the measures, as well as additional references to textbooks and papers covering additional aspects of those measures.

WCSS measures the “goodness” of a cluster solution via its compactness, one of the most fundamental indicators of cluster quality. Indeed, for each $k \in [1, k_{\max}]$, the method consists of computing the sum of the square distance between each element in a cluster and the centroid of that cluster. The “correct” number of clusters k^* is predicted according to the following rule of thumb. For each $k < k^*$, the value of WCSS should be substantially decreasing, as a function of the number of clusters k . On the other hand, for values of $k^* < k$, the compactness of the clusters will not increase as much, causing the value of WCSS not to decrease as much. The following heuristic approach comes out ([Hastie et al., 2003](#)): Plot the values of WCSS, computed on the given clustering solutions, in the range $k \in [1, k_{\max}]$; choose as k^* the abscissa closest to the *knee* in the WCSS curve.

Gap is one of the few measure that incarnates the Monte Carlo Confidence Analysis paradigm. It is based on WCSS. In particular, via a Monte Carlo simulation it tries to identify the knee in the WCSS curve. Intuitively, a null model is generated for each $k \in [1, k_{\max}]$, the “distance” (i.e., gap) between the null model and the WCSS curve is computed. Therefore, it returns as k^* the first value of k such that its gap is less than the its value at $k+1$ corrected by the standard deviation accounting for the null model.

Consensus builds, for each $k \in [2, k_{\max}]$, a consensus matrix indicating the level of agreement of clustering solutions, via a series of independent sampling of the dataset and their partitions via a clustering algorithm. Based on experimental observations and logic arguments, [Monti et al. \(2003\)](#) derived a *rule of thumb* to estimate k^* based on the empirical cumulative distribution function of the entries of the consensus matrix. The interested reader is referred to [Monti et al. \(2003\)](#) for details.

FOM is based on a root mean square deviation over all features. It uses the same heuristic methodology outlined for WCSS, i.e., one tries to identify the knee in the *FOM* plot as a function of the number of clusters.

It is worth pointing out that also some heuristics have been proposed in the literature to gain a speed-up of some of the mentioned measures, keeping at the same time good performance in term of the assessment of k^* . The interested reader is referred to [Giancarlo et al. \(2008\)](#) and [Giancarlo and Utro \(2011, 2012\)](#).

A Case Study on Microarray Data

In the remainder of this section, the experimental framework used in this manuscript is detailed, i.e., datasets, algorithms and hardware. The intention is not only to analyze the performance of each measure in term of its ability to estimate k^* , but to also account for the computational resources it needs for that task.

It is useful to recall the definition of “gold solution” that naturally yields a partition of the datasets in two main categories. Strictly speaking, a gold solution for a dataset is a partition of the data in a number of classes known *a priori*. For their extensive benchmarking, [Giancarlo et al. \(2008\)](#) and [Giancarlo and Utro \(2011\)](#) used several datasets with a gold solution. For the sake of conciseness, only one of their datasets is used in this article.

Lymphoma: It is an 80×100 data matrix, where each row corresponds to a tissue sample and each column to a gene. The dataset comes from the study of [Alizadeh et al. \(2000\)](#) on the three most common adult lymphoma tumors. There is a partition into three classes and it is taken as the gold solution. The dataset has been obtained from the original microarray experiments, consisting of an 80×4682 data matrix, following the same preprocessing steps detailed in [Dudoit and Fridlyand \(2002\)](#).

Clustering Algorithm

In this article, a suite of clustering algorithms is used. Among the hierarchical methods ([Jain and Dubes, 1988](#)) Hier-A (Average Link), Hier-C (Complete Link), and Hier-S (Single Link), are used. Moreover, K-means ([Jain and Dubes, 1988](#)) is used, both in the version that starts the clustering from a random partition of the data and in the version where it takes, as part of its input, an initial partition produced by one of the chosen hierarchical methods. For K-means, the acronyms of those versions are K-means-R, K-means-A, K-means-C and K-means-S, respectively. All of the algorithms use Euclidean distance in order to assess similarity of single elements to be clustered.

Hardware, Tools and Availability

All experiments for the assessment of the precision and timing were performed on a MacBook Pro (Retina, 15-in., Mid 2015), with 2.5 GHz Intel Core i7 processor and 16 GB of memory. All measures are implemented in ValWorkBench ([Giancarlo et al., 2015](#)) and see Section Relevant Website.

Analysis and Assessment

In this section, we are assessing the ability of the clustering algorithms as well as the performance of the internal validation measures.

Clustering Algorithms

External indices can be very useful in evaluating the performance of algorithms and internal/relative indices, with the use of datasets that have a gold standard solution. A brief illustration is given of the methodology for the external validation of a clustering algorithm, via an external index that needs to be maximized. The same methodology applies to internal/relative indices, as discussed in [Yeung et al. \(2001\)](#). For a given dataset, one plots the values of the index computed by the algorithm as a function of k , the number of clusters. Then, one expects the curve to grow to reach its maximum close or at the number of classes in the reference classification of the dataset. After that number, the curve should fall. In what follows, the results of the experiments are presented, with the use of the indices. For each dataset and each clustering algorithm, the *Adjusted Rand Index* computed for a number of cluster values in the range $[2,30]$ (see [Fig. 1](#)). The performance of the algorithms is somewhat mixed, and sometimes they are not precise, for example, the Hier-S algorithm.

The interested reader is referred to [Giancarlo and Utro \(2011\)](#) for a complete analysis of the other indices and datasets.

Internal Validation Measures

All internal validation measures mentioned in Section Background, in conjunction with any of the clustering algorithms detailed in Section A Case Study on Microarray Data have been used to estimate the number of cluster in the Lymphoma dataset. For brevity, we report in [Table 1](#) only the results for each method in conjunction with Hier-A, while the interested reader is referred to [Giancarlo et al. \(2008\)](#) for the complete list of experiment results.

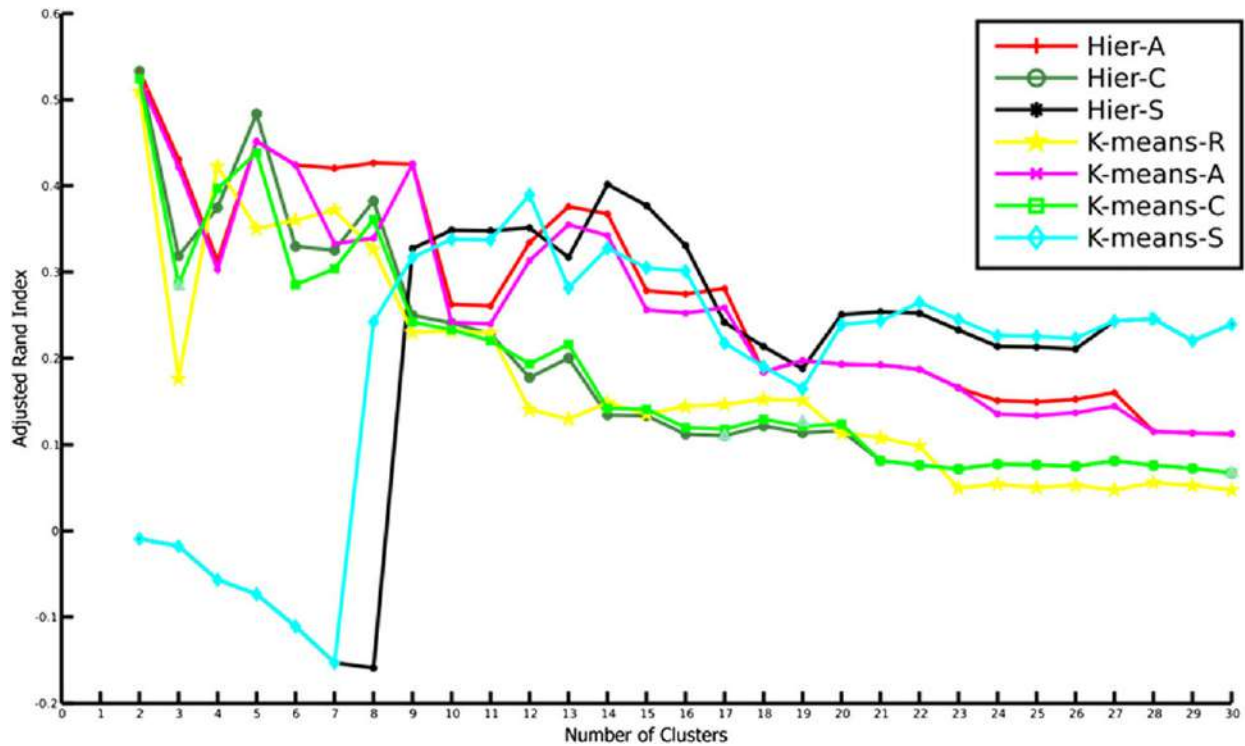


Fig. 1 The *Adjusted Rand Index* curves, for the Lymphoma datasets, as a function of the number of clusters, is plotted differently for each algorithm.

Table 1 A summary of the results of the Hier-A on all methods for the Lymphoma dataset. The columns precision indicates the number of clusters predicted by the measure, while the other one reports the timing in seconds for the execution of the corresponding experiment

	Precision	Timing (s)
WCSS-Hier-A	6	4.0×10^{-1}
Gap-Hier-A	6	5.6×10^0
Consensus-Hier-A	3	3.8×10^2
FOM-Hier-A	6	9.7×10^{-1}
Gold Solution	3	—

Results and Discussion

As it self evident from the results in Section Internal Validation Measures, and the broader results reported in [Giancarlo et al. \(2008\)](#), [Giancarlo and Utro \(2011\)](#), when computational time is taken into account, there is a hierarchy of measures, with WCSS being the fastest and *Consensus* the slowest. Overall, *Consensus* results to be the method of choice in terms of predictive power. From the experiments reported in [Giancarlo et al. \(2008\)](#) and [Giancarlo and Utro \(2011\)](#) *FOM* is the second best performer, although it may not be competitive since it has essentially the same predictive power of WCSS but it is much slower in time, depending on the dataset. Moreover, it is worth recalling that some approximation/heuristics of the mentioned measures have been proposed ([Giancarlo et al., 2008](#); [Giancarlo and Utro, 2011](#)) and that some of them seem quite competitive due to their precision and computational efficiency.

Finally, it is worth pointing out that all the measures we have considered show severe limitations on large datasets, either due to computational demand or to lack of precision.

Future Directions

Some possible future directions involve the design of fast approximations of other stability internal validation measures. It would be a quite remarkable accomplishment to design an internal validation measure that closes the gap between the time performance

of the fastest internal validation measures and the most precise. Moreover, high performance computing, i.e., shared memory multi-thread (Unold and Tagowski, 2015) may be also explored to improve the performance of validation measures.

Closing Remarks

We have provided a tutorial on techniques with grounds in the statistics literature and that are of use for (1) partitioning and (2) estimating the number of clusters in a data set. We have also demonstrated some of their uses on real case studies. It is worth pointing out that the analysis of microarray data, as all types of biological data, offers unique challenges due to their level of noise and their high-dimensionality.

See also: Data Mining: Clustering. Hidden Markov Models. Proteomics Mass Spectrometry Data Analysis Tools

References

- Alizadeh, A., *et al.*, 2000. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature* 403, 503–511.
- Ben-Hur, A., Elisseeff, A., Guyon, I., 2002. A stability based method for discovering structure in clustering data. In: *Pacific Symposium on Biocomputing (ISCB)*, pp. 6–17.
- Bock, H., 1985. On some significance tests in cluster analysis. *Journal of Classification* 2, 77–108.
- Dudoit, S., Fridlyand, J., 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology* 3.
- Fowlkes, E., Mallows, C., 1983. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 78, 553–584.
- Giancarlo, R., Lo Bosco, G., Pinello, L., Utro, F., 2013. A methodology to assess the intrinsic discriminative ability of a distance function and its interplay with clustering algorithms for microarray data analysis. *BMC Bioinformatics* 14, S6.
- Giancarlo, R., Scaturro, D., Utro, F., 2008. Computational cluster validation for microarray data analysis: Experimental assessment of cleft, consensus clustering, figure of merit, gap statistics and model explorer. *BMC Bioinformatics* 9, 462.
- Giancarlo, R., Scaturro, D., Utro, F., 2015. ValWorkBench: An open source Java library for cluster validation, with applications to microarray data analysis. *Computer Methods and Programs in Biomedicine* 108, 207–217.
- Giancarlo, R., Utro, F., 2011. Speeding up the consensus clustering methodology for microarray data analysis. *Algorithms for Molecular Biology* 6, 1.
- Giancarlo, R., Utro, F., 2012. Algorithmic paradigms for stability-based cluster validity and model selection statistical methods, with applications to microarray data analysis. *Theoretical Computer Science* 428, 58–79.
- Giroto, S., Pizzi, C., Comin, M., 2016. MetaProb: Accurate metagenomic reads binning based on probabilistic sequence signature. *Bioinformatics* 32, i567–i575.
- Gordon, A., 1996. Null models in cluster validation. *From Data to Knowledge: Theoretical and Practical Aspects of Classification*. 32–44.
- Handl, J., Knowles, J., Kell, D., 2005. Computational cluster validation in post genomic data analysis. *Bioinformatics* 21, 3201–3212.
- Hastie, T., Tibshirani, R., Friedman, J., 2003. *The Elements of Statistical Learning*. Springer.
- Hubert, L., Arabi, P., 1985. Comparing partitions. *Journal of Classification* 2, 193–218.
- Jain, A., Dubes, R., 1988. *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall, Inc.
- Krzanowski, W., Lai, Y., 1985. A criterion for determining the number of groups in a dataset using sum of squares clustering. *Biometrics* 44, 23–34.
- Leung, M.-Y., Marsch, G., Speed, T., 1996. Over and underrepresentation of short DNA words in Herpesvirus genomes. *Journal of Computational Biology* 3, 345.
- McMurdie, P., Holmes, S., 2013. Phyloseq: An R package for reproducible interactive analysis and graphics of microbiome census data. *PLOS ONE* 8, e61217.
- Monti, S., Tamayo, P., Mesirov, J., Golub, T., 2003. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning* 52, 91–118.
- Sarle, W., 1983. Cubic Clustering Criterion. SAS.
- Sera, F., Romualdi, C., F., F., 2016. Similarity measures based on the overlap of ranked genes are effective for comparison and classification of microarray data. *Journal of Computational Biology* 7, 603–614.
- Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the number of clusters in a dataset via the gap statistics. *Journal Royal Statistical Society B* 2, 411–423.
- Unold, O., Tagowski, T., 2015. A parallel consensus clustering algorithm. *Lecture Notes in Computer Science* 9432, 318–324.
- Utro, F., *et al.*, 2012. ARG-based genome-wide analysis of cacao cultivars. *BMC Bioinformatics* 13, S17.
- Van Rijsbergen, C., 1979. *Information Retrieval*, second ed. London: Butterworths.
- Yeung, K., Haynor, D., Ruzzo, W., 2001. Validating clustering for gene expression data. *Bioinformatics* 17, 309–318.

Further Reading

- Abonyi, J., Balázs, F., 2007. *Cluster analysis for data mining and system identification*.
- Bezdek, J.C., 2013. *Pattern Recognition With Fuzzy Objective Function Algorithms*. Springer Science & Business Media.
- Bhattacharyya, S., *et al.*, 2016. *Intelligent Multidimensional Data Clustering and Analysis*. IGI Global.
- Brian, S., *et al.*, 2011. *Cluster Analysis*. Wiley.
- Hennig, C., Meila, M., Murtagh, F., Rocci, R., 2015. *Handbook of Cluster Analysis*. Chapman and Hall/CRC.
- Jajuga, K., Bock, H.H., 2002. *Classification, Clustering, and Data Analysis: Recent Advances and Applications*. Springer Science & Business Media.
- Kaufman, L., Rousseeuw, P.J., 2005. *Finding Groups in Data*. Wiley.
- King, R.S., 2014. *Cluster Analysis and Data Mining*. Mercury Learning & Information.
- Tan, P.-N., Steinbach, M., Kumar, V., 2005. *Introduction to Data Mining*. Pearson.

Relevant Website

<http://www.math.unipa.it/~raffaele/valworkbench/>
Validation Work Benchmark.

Biographical Sketch



Raffaele Giancarlo is a Full Professor of Computer Science at University of Palermo. His research interests include Algorithms and Data Structures, Data Compression, Information Retrieval, Bioinformatics and Computational Biology. He has produced over 100 research papers in distinguished journals and international conferences. He holds 5 patents.



Dr. Filippo Utro is a Research Scientist in the Computational Genomics group at the IBM T.J. Watson Research Center. He joined IBM Research in 2011 after completing his PhD work in computer science at the University of Palermo, Italy. His research is focused around algorithm development and analysis, specifically on challenges in computational biology. He started his career at IBM as an investigator in the cacao genome project and has since continued to research on plant, population, epigenetics and cancer genomics. He has produced over 30 research papers in distinguished journals and international conferences, and holds 5 patents (pending).

Data Mining: Outlier Detection

Fabrizio Angiulli, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Outlier Detection

Outlier detection is one of the prominent data mining tasks, with a lot of applications in several domains. While the other three main data mining tasks, that are clustering, classification, and dependency detection, deal with regularities in the data, the goal of outlier detection is to isolate anomalous observations within the data. Clustering is the process of partitioning a set of objects into homogeneous groups, or clusters. Classification is the task of assigning objects to one of several predefined categories. Dependency detection techniques search for pairs of attribute sets which exhibit some degree of correlation in the dataset at hand.

As for the outlier detection task, it can be defined as follows: given a set of data points or objects, find the objects that are considerably dissimilar, exceptional or inconsistent with respect to the remaining data. These exceptional objects are also referred to in the literature as outliers.

It must be noticed that the other three categories of data mining techniques also deal with outliers, since many clustering, classification and dependency detection methods produce outliers as a by-product of their main task. Indeed, mislabeled points are considered outliers by classification techniques and they should be removed from the training set to improve the accuracy of the learned classifier. Also, points that do not well fit to any cluster are considered outliers by clustering techniques.

Despite these techniques can be able to treat some form of anomalies, it must be pointed out that the approach of searching for outliers through techniques which are not explicitly designed for detecting outliers presents different drawbacks. To illustrate, the quality of clusters returned by a clustering algorithm is usually affected by the presence of outliers and this in its turn affects the quality of the associated by-product outliers.

The advantage of outlier detection definitions relies on the fact that they are tailored on specific notions of abnormality. Moreover, other than guaranteeing an higher quality, outlier detection algorithms can be enormously more efficient than algorithms searching for regularities.

In the context of regularity-based data mining tasks, outliers are considered as noise that must be eliminated since they worsen the predictive or descriptive capabilities of algorithms. However, as already pointed in the literature, one person's noise could be another person's signal, and thus there are applications in which the outliers themselves represent the knowledge of interest. Outlier mining is used in telecom or credit card frauds, in intrusion detection, in medical analysis, in marketing and customer segmentation, in surveillance systems, in data cleaning, in biological data analysis and in many other fields.

Outlier Detection Approaches

Approaches to outlier detection can be grouped in three main families: supervised, semi-supervised, and unsupervised ([Aggarwal, 2013](#)).

Supervised methods exploit the availability of a labeled dataset, containing observations already labeled as normal and abnormal, in order to build a model of the normal class. Since usually normal observations are the great majority, these datasets are unbalanced and specific classification techniques must be designed to deal with the presence of rare classes.

Semi-supervised methods assume that only normal examples are given. The goal is to find a description of the data, that is a rule partitioning the point space into an accepting region, containing the normal points, and a rejecting region, containing all the other points. These methods are also called one-class classifiers or domain description techniques, and they are related to novelty detection since the domain description is used to identify points significantly deviating from the training examples.

Unsupervised methods search for outliers in an unlabelled dataset by assigning to each point a score which represents its degree of abnormality. Scores are usually computed by comparing each point with the points belonging to its neighborhood.

Data mining researchers have largely focused on unsupervised approaches. The rest of this section provides a detailed overview of the main unsupervised outlier detection approaches. Each family of approaches is presented in a separate subsection.

Statistical-Based Outliers

Early approaches to detect outliers originated in statistics. [Hawkins' \(1980\)](#) definition of outlier is usually reported to clarify the kind of approach: "An outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Statistical approaches are model-based, since a model is hypothesized for the data and the points are evaluated with respect to how well they fit the model or generating mechanism: an outlier is a point that has low probability to occur with respect to the probability distribution associated with the data.

Once a distribution model is selected, its parameters are estimated from data, e.g. the mean and the standard deviation for a normal distribution. A lot of statistical tests have been designed in the statistical literature, many of which are highly specialized. In [Barnett and Lewis \(1994\)](#) a comprehensive treatment is provided, listing about one hundred discordancy tests that differ for (*a*)

the distribution, (b) whether or not the distribution parameters are known, (c) the number of expected outliers, (d) the type of outliers, e.g. upper or lower.

For example, in a one-dimensional Gaussian distribution there is little chance that a value x will occur more than c standard deviations σ apart from the mean μ . Thus, given a significance level α , c can be obtained as the value such that

$$\Pr \left[\left| \frac{x - \mu}{\sigma} \right| \geq c \right] \leq \alpha$$

This approach can be generalized to higher dimensions by exploiting the Mahalanobis distance of a point to the mean of the distribution, which is directly related to the probability to observe the point.

When there is knowledge of the data and the kind of test to apply, these tests can be very effective. The problems of statistical tests are that in most cases no standard distribution can adequately model the observed values whose underlying distribution is in most cases unknown, that there may not be a test developed for the case at hand, and that they admit in general only one generating mechanism or cluster.

Distance-Based Outliers

Distance-based outlier detection has been introduced by Knorr *et al.* (2000) to overcome some limitations of statistical methods. According to the definition there provided, an object x is a *distance-based outlier* with respect to parameters k and R if less than k objects in the dataset lie within distance R from x . This definition corresponds to a form of density estimation based on counting the number of data points lying in a fixed-size neighborhood of the point, that is an hypersphere of radius R in the Euclidean space.

Subsequently, Ramaswamy *et al.* (2000) in order to provide a ranking of the outliers, modified the previous definition as follows: given two integers k and n , an object x is said to be the n -th top outlier if exactly $n-1$ objects have higher value for D_k than x , where D_k denotes the distance of the k -th nearest neighbor of the object.

Moreover, Angiulli and Pizzuti (2005) with the aim of taking into account the whole neighborhood of each point, proposed to rank points on the basis of the average distance from their k nearest neighbors, a measure called *weight*, and introduced the *HilOut* (Angiulli and Pizzuti, 2005) and *SolvingSet* algorithms (Angiulli *et al.*, 2006) for their efficient computation.

These two alternative definitions correspond to a different form of density estimation based on determining the radius of the smallest neighborhood of the point containing a fixed-size number of data points.

The advantage of distance-based definitions w.r.t. statistical tests is that there is no need to know the form of the distribution underlying the data and/or the parameters associated with that distribution. Moreover, distance-based outliers are suitable even in situations when the dataset does not fit any standard distribution.

As an interesting property, it can be shown that in some cases this definition generalizes the definition of outlier in statistics, that is for some known distributions there exists values for the parameters such that finding the distance-based outliers corresponds to determine the less probable observations according to the distribution function at hand, as discussed in Knorr *et al.* (2000) and Angiulli and Fasseti (2009).

Since the distance-based definition detects outliers on the basis of their absolute density, it corresponds to a notion of global outlier, as the outliers are the objects lying in the sparser regions of the feature space. The definition admits more than one generating mechanism or cluster, although they should have similar characteristics. Moreover, it works well even in situations where the geometric intuition is not available, e.g. non-metric spaces, and may exhibit high-ranking performances in high-dimensional spaces.

Distance-based outlier scores are monotonic non-increasing with respect to the portion of the dataset already explored. This property allows to design effective pruning rules and very efficient algorithms.

The DOLPHIN algorithm (Angiulli and Fasseti, 2009) is a state-of-the-art technique for detecting distance-based outliers w.r.t. parameters R and k . It is able to work with disk resident datasets and performs only two data scans. It has low main memory requirements and integrates pruning rules and indexing strategies. This method is very fast and able to manage large collections of data. Its temporal cost is linear in the dataset size, being $O(n d k/p)$, where n is the number of dataset points, d is the data dimensionality, and $p < 1$ is a constant which depends on the data distribution. It is very profitable for metric data, but can be used with any type of data for which a distance function is defined.

Density-Based Outliers

Despite the name usually employed to refer this family of techniques, density-based outlier definitions detect outliers on the basis of their *relative density*.

Indeed, differently from distance-based definitions, which declare as outliers the points where the estimated data density is low, density-based definitions score points on the basis of the degree of disagreement between the estimated density of the point and the estimated density of its surrounding or neighboring points.

Thus, this family of definition are better characterized as a notion of local outlier. Indeed, a relatively high density point could be a density-based outlier, provided that it is located close to a region of much higher density. This situation typically occurs when

the point lies on the border of a cluster. This makes these techniques very suitable to isolate outliers in data containing sub-populations or clusters of varying density and, hence, with markedly different characteristics.

A way to define the density of a point is to consider the average distance to its k -nearest neighbors, which is similar to the distance-based definition provided in Angiulli and Pizzuti (2005):

$$dens_k(x) = \left(\frac{\sum dist(x, y)}{|N_k(x)|} \right)^{-1}$$

where $N_k(x)$ denotes the set of points whose distance from x is not greater than the distance separating x from its k -th nearest neighbor.

Density-based outliers score points on the basis of their *relative density*, that is the ratio between the density of a point and the average density of its nearest neighbors:

$$score_k(x) = \frac{dens_k(x)}{(\sum dens_k(y))/|N_k(x)|},$$

Density-based outlier methods have been introduced in Breunig et al. (2000) with the Local Outlier Factor (LOF) measure. LOF measures the degree of an object to be an outlier by comparing the density in its neighborhood with the average density in the neighborhood of its neighbors. Thus, the formalization of the LOF score is very similar to the relative density score above reported, with some minor modifications aiming to mitigate the effect of statistical fluctuations of the distance.

Advantages of the LOF score are that it is able to identify local outliers, and that the definition can be applied in any context a dissimilarity function can be defined. Moreover, experimentally works well in different scenarios, e.g. in network intrusion detection.

However, the geometric intuition of LOF is only applicable to low dimensional vector spaces. Also, the LOF score is difficult to interpret: a value of 1 or even less indicates a clear inlier, but there is no clear rule for when a point is an outlier. Moreover, finding LOF outliers is costly, since in principle the exact k -nearest neighbors of each point are required, which in the worst case costs $O(n^2)$.

The Multi-granularity DEviation Factor (MDEF), introduced in Papadimitriou et al. (2003), represents an alternative definition of relative density. This time the density of an object is defined on the basis of the number of objects lying in its neighborhood, that is a fixed-radius hypersphere centered on the point.

The Gradient Outlier Factor (GOF) (Angiulli and Fassetti, 2016) represents an alternative notion of local outlier. Let $\rho(t)$ a function which is maximum in $t=0$ and tends to 0 for arbitrarily large t values, e.g. $\rho(t)=1/(1+t)$. Given a dataset, it is assumed that it is generated by a random variable X having unknown pdf f . The domain neighborhood $N_f(x)$ of x is the neighborhood of x having radius $\rho(\|\nabla f(x)\|)$, where $\nabla f(x)$ denotes the gradient of the function f evaluated in x . Intuitively the size of $N_f(x)$ is inversely proportional to the slope of f in x . The GOF is defined as the probability

$$GOF(x) = Pr[X \in N_f(x)] = f(x) \cdot \rho(\|\nabla f(x)\|)$$

to observe a data value in the domain neighborhood of x . The form of f is determined by exploiting Kernel Density Estimation (Scott, 1992).

This definition encompasses some desirable characteristics of various existing definitions that are the unification with the statistical definition, as in the case of the distance-based definitions, the ability to deal with different populations or to capture local outliers, as in the case of the density-based definitions, and the ability to detect some subtle form of anomalies, as in the case of abnormal concentrations of objects.

The Outlier Detection using Indegree Number (ODIN) algorithm (Hautamaki et al., 2004) exploits the notion of reverse nearest neighbor to isolate outliers and due to the nature of the detected outliers can be assimilated to local outlier definitions. A point y is a k -reverse nearest neighbor of x , if x is among the k -nearest neighbors of y . The score of a point x is the number of k -reverse nearest neighbors of x . Intuitively, points associated with low scores are unlikely to be selected as neighbors by any other point, and this witnesses for their outlierness.

The Concentration Free Outlier Factor (CFOF), introduced in Angiulli (2017), is a score particularly suitable for high dimensional data. Specifically, the CFOF score of the point x is given by the smallest integer k for which x is a k -nearest neighbor of at least the ρ fraction of the data, where $\rho \in (0, 1)$ is an input parameter. This definition is adaptive to different density levels and has the peculiarity to resist to curse of dimensionality related phenomena, such as the tendency of distances to become almost similar as the dimensionality increases.

Isolation-Based Outliers

The basic idea of *Isolation Forest* (iForest) (Liu et al., 2012) is to detect outliers on the basis of their inclination to be isolated from the rest of the data. Since anomalies are few and different, they are identified as the objects more susceptible to isolation.

With this aim the Isolation Forest technique builds a data-induced tree, also called Isolation Tree (or *iTree*) by recursively and randomly partitioning instances, until all of them are isolated. The random partitioning produces shorter paths for anomalies.

More formally, an *iTree* is a binary tree T associated with a subset of the data points, whose internal nodes consist of two childs, T_l and T_r , and one test, while external nodes (leafs) consist of one single point. The test consists of an attribute a and a split value v

such that the condition $a < v$ partitions data points associated with T into T_l and T_r . The *Path Length* $h(x)$ of a point x in T is the number of edges transversed in order to reach the external node containing x .

An iTree is built by recursively expanding non-leaf nodes (initially all the data is associated with a single internal node) by randomly selecting an attribute a and a split value v .

Due to the random nature of each iTree, to assess result quality a collection of iTrees is randomly built. The anomaly score $s(x)$ of x is then defined in terms of its average path length. Specifically:

$$s(x) = 2^{-\frac{E[h(x)]}{c(n)}}$$

where $E[h(x)]$ denotes the average path length of x in the collection of iTrees and $c(n)$ is a normalization constant which depends on the total number of data points.

To gain efficiency and ameliorate quality, subsampling is employed during collection building. Advantages of this approach are that it has a linear time complexity with a low constant and low memory requirements, due to subsampling.

However, the approach can be used only on ordered domains and does not support categorical attributes. Moreover there is no clear semantics, even if outliers may have different nature than those detected by other approaches, as distance- and density-based methods.

Angle-Based Outliers

In some applications dealing with high-dimensional data, angles are considered more reliable than distances. Think, e.g., to the text mining field, where the cosine distance, measuring the angle between two data points, is employed as a dissimilarity measure.

The Angle-Based Outlier Factor (ABOF) has been designed for high-dimensional data and, following the above perspective, scores data points on the basis of the variability of the angles formed by a point with each other pair of points.

The intuition is that for a point within a cluster, the angles between difference vectors to pairs of other points differ widely, while the variance of these angles will become smaller for points at the border of a cluster and for isolated points.

Formally, the angle-based outlier factor $ABOF(\vec{A})$ is the variance over the angles between the difference vectors of $\vec{A}$ to all pairs of points in the dataset D , weighted by the distance of the points:

$$ABOF(\vec{A}) = \text{VAR}_{\vec{B}, \vec{C} \in D} \left(\frac{\langle \vec{AB}, \vec{AC} \rangle}{\|\vec{AB}\|^2 \|\vec{AC}\|^2} \right)$$

where $\vec{BC}$ denotes the difference vector $\vec{C} - \vec{B}$ and the computation takes into account only triples $(\vec{A}, \vec{B}, \vec{C})$ where the three points are mutually different.

A problem with this definition is that since for each point all pair of other points must be considered, the temporal cost is very high, that is cubic $O(n^3)$.

The *approxABOF* score approximates the ABOF one by considering only the pairs of points with the strongest weight in the variance, that are the k nearest neighbors. This approximation results in an acceleration, with temporal cost $O(n^2 + n k^2)$. However, the quality of the approximation usually deteriorates with increasing dimensionality and the latter definition is only suitable for low-dimensional datasets.

Subspace-Based Outliers

Traditional outlier detection methods search for anomalies by taking into account the full feature space. However, in many situations points exhibit exceptional behavior only when the analysis is focused on some subset of the features, also called a subspace.

These techniques can be grouped in two categories, subspace-based outlier detection, which search for outliers in an unsupervised manner by restricting attention to subset of features, and outlier explanation techniques, which try to explain the abnormality of a user-provided outlier by singling out the subspace that most differentiates it from the rest of the data population.

In both cases, techniques have to deal with the enormous search space, having exponential size in the number of attributes, composed by all the subsets of the overall set of attributes.

Knorr and Ng (1999) focus on the identification of the intensional knowledge associated with distance-based outliers. First, they detect the outliers in the full attribute space and then, for each outlier o , they search for the subspace that better explains why it is exceptional, that is the minimal subspace in which o is still an outlier.

A popular method for finding outliers in subspaces is due to Aggarwal and Yu (2001), which detects anomalies in a d -dimensional dataset by identifying abnormal lower dimensional projections. Each attribute is divided into equi-depth ranges, and then k -dimensional hypercubes ($k < d$) in which the density is significantly lower than expected are searched for. The search technique is based on exploiting evolutionary algorithms.

Subspaces can be useful to perform *example-based* outlier detection: a set of example outliers is given as input and the goal is to detect the dataset objects which exhibit the same exceptional characteristics as the example outliers. Zhu et al. (2005) accomplish

this task by means of a genetic algorithm searching for the subspace minimizing the density associated with hyper-cubes containing user examples.

The approach (Dang *et al.*, 2014) considers the problem of detecting and *interpreting* local outliers, i.e., objects which are outliers relative to a subpopulation of neighbors, rather than the entire dataset. The outlierness is measured in a low-dimensional subspace capable of preserving the locality around the neighbors while at the same time maximizing the distance from the outlier candidate. Incidentally, the low-dimensional transformation also provides the insights for the relevant features which contribute most to the outlierness.

Outlier Ensembles

Ensemble analysis is a method used in data mining to reduce the dependence of the model on the specific dataset. The basic idea is to combine the results from different models in order to obtain a more robust one. This kind of approach has been widely applied to clustering and classification, while less attention has been devoted to its applicability in the context of outlier detection.

The basic ensemble algorithm (i) computes different outlier scores for a point using the output of different outlier detection algorithms and then (ii) combines the scores from different algorithms to obtain a more robust final score.

One of the first approaches to outlier ensembles is due to Lazarevic and Kumar (2005), which proposed to exploit subspaces to improve accuracy. Specifically, they experimented feature bagging: the same outlier detection algorithm is executed multiple times, each time considering a randomly selected set of features from the original feature set, and finally outlier scores are combined in order to find outliers of better quality.

Both Aggarwal (2012) and Zimek *et al.* (2013) report a comparison of existing outlier detection ensemble techniques and of the associated characteristics. Specifically, designing a good ensemble involves the following main challenges: the proper selection of the algorithms to employ, the normalization of the scores to be combined, and the selection of the strategy to combine scores.

The first issue, that is important for building good ensembles, should guarantee the diversity of the models. Aggarwal (2012) proposed a categorization of ensemble approaches to outlier detection by distinguishing between independent ensembles and sequential ensembles, and between model-centered ensembles and data-centered ensembles.

In independent ensembles the executions of the algorithms are independent each other, while in sequential ensembles the executions of the algorithms depend from the results of the past executions so that the models can be successively refined.

Moreover, a model-centered ensemble combines the output of different algorithms executed on the same data, while a data-centered ensemble combines the output of the same algorithm executed on different derivations, in the form of subsets and/or subspaces, of the same data.

As for other issues, since due to the different nature of the various unsupervised outlier detection approaches proposed in the literature the associated score are non-homogeneous, Gao and Tan (2006) proposed techniques for converting output scores from different outlier detection algorithms into probability estimates.

Outlier Explanation

In many real situations one is given a data population characterized by a certain number of attributes, and information are provided that one of the individuals in that data population is abnormal, but no reason whatsoever is given as to why this particular individual is to be considered abnormal. Outlier explanation techniques precisely try to explain the abnormality of an user-provided outlier by singling out the subspace that most differentiate it from the rest of the data population.

The technique introduced in Angiulli *et al.* (2009) provides explanations for a given outlier in a categorical dataset in the form of explanation-outlying property pairs. To illustrate, consider a table reporting some of the attributes collected for some animals. It is known that the platypus is an exceptional animal being it a mammal, but laying eggs. This knowledge can be formalized by noticing that among dataset objects having value “true” for the attribute “eggs”, the platypus is the only animal having value “true” for the attribute “milk”. Obviously the value “true” for the attribute “milk” is not an exceptional feature per se, but it is surprising when attention is restricted to the animals which lay eggs. This is a case where an outlying property is individuated, where the attribute “eggs” plays the role of explanation for the outlying property “milk” of the platypus.

More formally, a *property* (set of attributes) P is *outlying* for an object x , if the frequency of the combination of values assumed by x on the attributes in P is rare if compared to the frequencies associated with the other combinations of values assumed on the same attributes by the other objects of the dataset. Moreover, E is an *explanation* (set of attributes) for p and x , if restricting the attention to the subset of dataset objects sharing with x the same values on the attributes in E , makes p an outlying property for x . Thus, in the example above $E = \text{“eggs”}$ is an explanation for the outlying property $P = \text{“milk”}$ and the object $x = \text{“platypus”}$.

This approach, which consists in determining the typicality of frequency values, is more robust than directly considering absolute frequency values. As an example, consider a key attribute: obviously the value assumed by any object on that attribute occurs just once on the dataset, but this cannot be considered exceptional since all the other values occur once.

The technique was applied in Angiulli *et al.* (2009) to the analysis of a genetic dataset concerning genotype information from the DNA of subjects born in Calabria. Specifically, for each individual the stored information concerned ten polymorphic genetic loci and the analysis consisted in the explanation of longevous individuals.

The approach above depicted is extended to the handling of numerical attributes in Angiulli *et al.* (2017) and to the simultaneous explanation of groups of outliers in Angiulli *et al.* (2013).

In Micenkova *et al.* (2013), the authors assume that outliers are given in input and their objective is to find an *explanatory subspace*, that is a subspace of the original numerical attribute space where the outlier shows the greatest deviation from the other points. The basic idea of the algorithm is to encode the notion of outlierness as separability: given an object x deemed as an outlier, one can devise an artificial set of points X oversampled from a gaussian distribution centered in x . Then, the outlierness of x can be measured in terms of the accuracy in separating the artificial points X from the other points in the dataset. Having encoded the outlierness as a classification problem, the explanatory subspace can hence be reduced to feature selection relative to such a classification problem.

Duan *et al.* (2015) propose a method based on ranking and searching. Given a subspace, the method ranks the query object within the subspace according to a density-based outlierness measure. Then explanations are provided as those minimal subspaces for which the rank is minimum.

There are some substantial differences between these two approaches and the preceding ones. In the latter approaches it is assumed that outlierness is relative to the whole population, and methods return individual subspaces where the query object is mostly outlying comparing to the other subspaces. By contrast, the former approaches model the scenario where the outlierness can be expressed relative to a homogeneous subpopulation. In this respect, the meaning of the two types of explanations is fundamentally different.

Closing Remarks

Outlier detection is a fundamental data analysis tool having several applications in different domains. We surveyed the main families of unsupervised data mining techniques for the detection of anomalies. The discussed approaches are characterized by different modalities of perceiving the abnormality of a given observation. Understanding their peculiarities is of help to the analyst during the process of selection of the right technique to apply to the data at hand. The analysis included statistical-based, distance-based, density-based, isolation-based, angle-based, subspace-based, ensembles, and methods providing explanations. The above methods can be generally applied to data modelled by several features or on which a notion of distance is defined. The field of outlier detection is an active one, with several techniques developed also in the context of specific data types not explicitly considered here, including, e.g., spatial data (Chen *et al.*, 2008), sequential and temporal data (Chandola *et al.*, 2012), and network structured data (Akoglu *et al.*, 2015).

References

- Aggarwal, C.C., 2012. Outlier ensembles. *ACM SIGKDD Explorations* 14 (2), 49–58.
- Aggarwal, C.C., 2013. *Outlier Analysis*. Springer.
- Aggarwal, C.C., Yu, P.S., 2001. Outlier detection for high dimensional data. *Proceedings of the ACM International Conference on Management of Data SIGMOD*, pp. 37–46.
- Akoglu, L., Tong, H., Koutra, D., 2015. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery DAMI* 29 (3), 626–688.
- Angiulli, F., 2017. Concentration free outlier detection. In: *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases ECMLPKDD*.
- Angiulli, F., Basta, S., Pizzuti, C., 2006. Distance-based detection and prediction of outliers. *IEEE Transactions on Knowledge and Data Engineering TKDE* 18 (2), 145–160.
- Angiulli, F., Fasseti, F., 2009. DOLPHIN: An efficient algorithm for mining distance-based outliers in very large datasets. *ACM Transactions on Knowledge Discovery from Data TKDD* 3 (1), [4:1:57].
- Angiulli, F., Fasseti, F., 2016. Toward generalizing the unification with statistical outliers: The gradient outlier factor measure. *ACM Transactions on Knowledge Discovery from Data* 10 (3), 1–26. [27].
- Angiulli, F., Fasseti, F., Manco, G., Palopoli, L., 2017. Outlying property detection with numerical attributes. *Data Mining and Knowledge Discovery* 31 (1), 134–163.
- Angiulli, F., Fasseti, F., Palopoli, L., 2009. Detecting outlying properties of exceptional objects. *ACM Transactions on Database Systems TODS* 34 (1), 1–62. (Article No. 7).
- Angiulli, F., Fasseti, F., Palopoli, L., 2013. Discovering characterizations of the behavior of anomalous subpopulations. *IEEE Transactions on Knowledge and Data Engineering TKDE* 25 (6), 1280–1292.
- Angiulli, F., Pizzuti, C., 2005. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering TKDE* 17 (2), 203–215.
- Barnett, V., Lewis, T., 1994. *Outliers in Statistical Data*. John Wiley & Sons.
- Breunig, M.M., Kriegel, H., Ng, R.T., and Sander, J., 2000. LOF: Identifying Density-based Local Outliers. In: *Proceedings of the ACM International Conference on Management of Data SIGMOD*, pp. 93–104.
- Chandola, V., Banerjee, A., Kumar, V., 2012. Anomaly Detection for Discrete Sequences: a Survey. *IEEE Transactions on Knowledge and Data Engineering TKDE* 24 (5), 823–839.
- Chen, D., Lu, C.-T., Kou, Y., Chen, F., 2008. On Detecting Spatial Outliers. *Geoinformatica* 12 (4), 455–475.
- Dang, X.H., Assent, I., Ng, R.T., Zimek, A., Schubert, E., 2014. Discriminative features for identifying and interpreting outliers. In: *Proceedings of the IEEE International Conference on Data Engineering ICDE*, pp. 88–99.
- Duan, L., Tang, G., Pei, J., Bailey, J., Campbell, A., Tang, C., 2015. Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery DAMI* 29 (5), 1116–1151.
- Gao, J., Tan, P.N., 2006. Converting output scores from outlier detection algorithms into probability estimates. In: *Proceedings of the International Conference on Data Mining ICDM*, pp. 212–221.
- Hautamaki, V., Karkkainen, I., Franti, P., 2004. Outlier detection using k-nearest neighbour graph. In: *Proceedings of the International Conference on Pattern Recognition ICPR*, pp. 430–433.
- Hawkins, D., 1980. *Identification of Outliers*. London: Chapman and Hall.

- Knorr, E., Ng, R.T., Tucakov, V., 2000. Distance-based outlier: algorithms and applications. *VLDB Journal* 8 (3–4), 237–253.
- Knorr, E. M., Ng, R. T., 1999. Finding intensional knowledge of distance-based outliers. In: *Proceedings of the International Conference on Very Large Databases VLDB*, pp. 211–222.
- Lazarevic, A., Kumar, V., 2005. Feature bagging for outlier detection. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining KDD*, pp. 157–166.
- Liu, F.T., Ting, K.M., Zhou, Z.-H., 2012. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data* 6 (1), 1–39. (Article No. 3).
- Micenikova, B., Ng, R.T., Dang, X.H., Assent, I., 2013. Explaining outliers by subspace separability. In: *Proceedings of the IEEE International Conference on Data Mining ICDM*, pp. 518–527.
- Papadimitriou, S., Kitagawa, H., Gibbons, B., Faloutsos, C., 2003. LOCI: Fast outlier detection using the local correlation integral. In: *Proceedings of the International Conference on Data Engineering ICDE*, pp. 315–326.
- Ramaswamy, S., Rastogi, R., Shim, K., 2000. Efficient algorithms for mining outliers from large data sets. In: *Proceedings of the ACM International Conference on Management of Data SIGMOD*, pp. 427–438.
- Scott, D.W., 1992. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley.
- Zhu, C., Kitagawa, H., Faloutsos, C., 2005. *Proceedings of the IEEE International Conference on Data Mining ICDM*, pp. 829–832.
- Zimek, A., Campello, R.J.G.B., Sander, J., 2013. Ensembles for unsupervised outlier detection: challenges and research questions. *SIGKDD Explorations* 15 (1), 11–22.

Relevant Website

<https://elki-project.github.io/>

ELKI: Environment for Developing KDD-Applications Supported by Index-Structures.

Biographical Sketch



Fabrizio Angiulli received the Laurea degree in computer engineering in 1999 from the University of Calabria (UNICAL). He is an associate professor of computer engineering since 2011 at DIMES Department, University of Calabria, Italy. In 2013, he obtained the Italian Full Professor qualification. Previously, he held a research and development position at ICAR of the National Research Council of Italy and, after that, a tenured assistant professor position at DEIS, University of Calabria. His main research interests are in the area of data mining, notably outlier detection and classification techniques, knowledge representation and reasoning and database management and theory. He has authored more than 80 papers appearing in premier journals and conference proceedings. He regularly serves on the program committee of several conferences and, as an associate editor, on the editorial board of the *AI Communications* journal. He is a senior member of the IEEE.

Pre-Processing: A Data Preparation Step

Swarup Roy, Sikkim University, Gangtok, India and North-Eastern Hill University, Shillong, India

Pooja Sharma, Tezpur University, Tezpur, India

Keshab Nath, North-Eastern Hill University, Shillong, India

Dhruva K Bhattacharyya, Tezpur University, Tezpur, India

Jugal K Kalita, University of Colorado, Boulder, CO, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

High-throughput experimental processes in various data driven computing domains have led to the availability or production of massive amounts of data. Explosive data growth can definitely be witnessed in biological research, and is due to revolutionary changes in how high throughput experiments are conducted in biomedical sciences and in biotechnology. A wide spectrum of biomedical data is generated by experiments in clinical and therapeutic investigations. Omics research, involving high throughput Next Generation Sequencing (NGS) and Microarray technologies, have been instrumental in generating massive amounts of mRNA, miRNA and gene expression data, as well as Protein-Protein Interaction data. Rapid and massive data generation sources lead to subsequent challenges in effective data storage and transmission. Efficient and scalable exploratory data mining techniques provide an emerging and important set of tools for knowledge discovery from *in silico* data sources (Roy *et al.*, 2013). A plethora of data mining and machine learning methods have been proposed, over the last several decades, to explore and discover hidden and unknown biological facts and relationships among biological entities.

Inference and analysis outcomes produced by knowledge discovery methods are highly dependent on the quality of the input data. A highly effective method is generally incapable of producing reliable results in the absence of high quality data. Biological data are generated and distributed across the globe. Heterogeneity in data due to different data acquisition techniques and standards, with non-uniform devices in geographically distributed research laboratories, makes the task of producing high quality data a near impossibility in many cases.

In general, real world data are of poor quality and can not be directly input to sophisticated data mining techniques. Such data are also often incomplete. As a result, it may be difficult to discover hidden characteristics, which may be of interest to the domain expert, or the data may contain errors, technically referred to as outliers. The interpretation of this type of data may also require understanding of the background knowledge used when analyzing the data. The same set of data can be represented in multiple formats and the values may have been normalized to different ranges, depending upon the statistical validity and significance. To use data mining techniques to mine interesting patterns from such data, they need to be suitably prepared beforehand, using data Pre-processing. Data pre-processing is a sequence of steps comprising Data Cleaning, Data Reduction, Data Transformation and Data Integration. Each step is equally significant, independent and can be executed in isolation. It is the data and the tasks to be performed which determine the step(s) to be executed and when. A domain expert's intervention may be necessary to determine the appropriate steps for a particular dataset. It is not appropriate to use the steps simply as a black-box.

High throughput technologies such as Microarray and Next Generation Sequencing (NGS) typically assess relative expression levels of a large number of cDNA sequences under various experimental conditions. Such data contain either time series measurements, collected over a biological process, or comparative measurements of expression variation in target and control tissue samples (e.g., normal versus cancerous tissues). The relative expression levels are represented as ratios. The original gene expression data derived after scanning the array, i.e., the fluorescence intensities measured on a microarray, are not ready for data analysis, since they contain noise, missing values, and variations arising from experimental procedures. Similar situations may arise even in RNASeq data derived from NGS. The *in silico* analysis of large scale microarray or RNASeq expression data from such experiments commonly involves a series of preprocessing steps (Fig. 1). These steps are indispensable, and must be completed before any gene expression analysis can be performed.

Data cleaning in terms of noise elimination, missing value estimation, and background correction, followed by data normalization is important for high quality data processing. Gene selection filters out genes that do not change significantly in comparison to untreated samples. In reality, not all genes actively take part in a biological activity, and hence are likely to be irrelevant for effective analysis. The intention is to identify the smallest possible set of genes that can still achieve good performance for the analysis under consideration (Daz-Uriarte and De Andres, 2006). Gene expression levels of thousands of genes are stored in a matrix form, where rows represent the relative expression level of the gene w.r.t. to sample or time. Performing a logarithmic transformation on the expression level, or standardizing each row of the gene expression matrix to have a mean of zero and a variance of one, are common pre-processing steps.

We now discuss the major data pre-processing steps in detail.

Data Cleaning

Quality of data plays a significant role in determining the quality of the resulting output when using any algorithm. Data cleaning is one such step in creating quality data (Herbert and Wang, 2007). It usually handles two situations, which give rise to

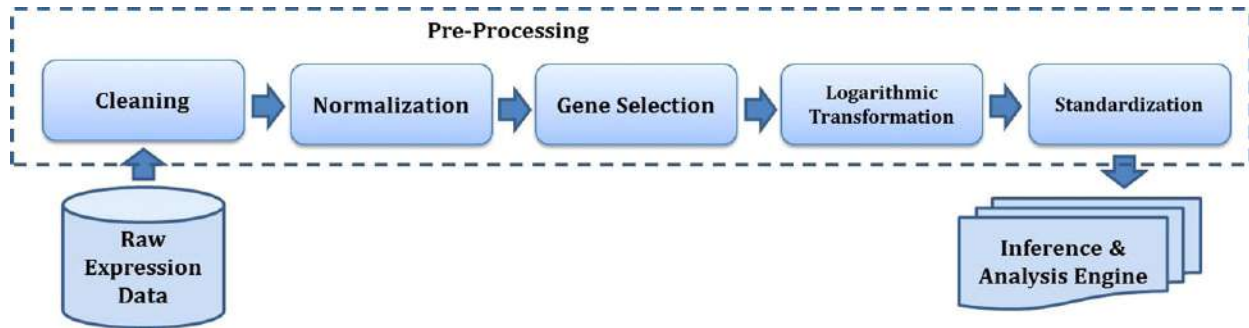


Fig. 1 Commonly used preprocessing steps for gene expression data analysis.

inconsistent input, missing values, misspellings, etc. The first deals with missing values, and the second is the data duplication problem. The presence of duplicate data adds to the computation time without adding any extra benefit to the result.

Handling Missing Value

A major issue that arises during pre-processing of data is that of missing values. A missing value occurs when the value of data example is not stored for a variable or feature of interest. Missing values can lead to serious issues, whether it is poor representation of the overall dataset in terms of its distribution, or bias in the results obtained (Hyun, 2013). Therefore, they need to be handled cautiously so as to enhance the performance of the methods. Missing data can occur because of a number of technical reasons. These may be one of the following:

- Malfunctioning or physical limitations of the measuring equipment may lead to some values being not recorded. Even interruptions in communication media during transportation of data may lead to missing values.
 - A few missing values may not seem to be important during the data collection phase, and corrective measures may not have been taken, at a later stage, to determine them.
 - Some values may have been removed due to inconsistency with other recorded data.
- There can be various types of missing data (Allison, 2012). These are broadly classified as follows.
- Missing Completely at Random (MCAR): When the probability of a missing observation is not related to the estimated value.
 - Missing at Random (MAR): When the probability of a missing observation is unrelated to its value, but depends on other aspects of the observed data.
 - Missing Not at Random (MNAR): If the missing data values do not belong to either of the above two categories, then it is said to be in MNAR, i.e., the probability of a missing observation is related to its value.

We explain the above types of missing data values with the help of an example. Suppose that we have two variables a and b , whose values are represented by vectors, A and B , respectively. Suppose that the individual values in vector B are always recorded, however, vector A has missing values. A missing value in A is said to be MCAR if the chance of it being missing is independent of values recorded in both A and B . On the other hand, if the probability of a missing value in A is likely to depend on the values in B , but is independent of the values of A , it is termed as MAR.

There are various approaches to handling missing data. It is possible to ignore the missing data completely, but if the amount of missing data is a large proportion of the dataset, this approach may severely affect the quality of the results. Also, if there are cases of MNAR missing values, they require parameter estimation to model the missing data. In order to deal with missing data, four broad classes of alternatives are available.

Ignoring and discarding data

Ignoring the missing values in the data is the simplest way to handle this issue. However, this is appropriate only for the MCAR situation. If we apply this treatment to other situations, we are likely to introduce bias in the data, and ultimately in the end results. Discarding missing data can be done in two ways: *complete case analysis* and *attribute discarding*. Complete case analysis is the *de facto* method, and is available in many programs. It discards all instances with missing data, and further analysis is carried out only on the remaining data.

The second way to discard attributes involves determining the extent to which the data is missing with respect to the number of instances, and the relevance of the attributes in the whole dataset. Based on that, one can discard or remove instances that are not widespread or whose relevance w.r.t. the whole dataset is not very significant. Note that if an attribute with missing values has very high relevance, it has to be retained in the dataset, as in Batista and Monard (2002).

Parameter estimation

This approach is suitable for MNAR missing data. It requires estimation of parameters using suitable models such as probabilistic, KNN, or SVD models. In order to estimate the parameters of a model, the maximum likelihood approach, using a variant of the

Expectation-Maximization algorithm can be used (Moon, 1996). The most common method using this approach is called BPCA (Oba *et al.*, 2003) which consists of three elementary steps – principal components regression, Bayesian estimation, and an iterative Expectation-Maximization (EM) algorithm. This is most commonly used in biological data cleaning.

Imputation

In the context of missing data, imputation refers to the consequence of actions taken to address the case of missing data. The aim of imputation is to replace the missing data with estimated values by using information in the measured values to infer population parameters. It is unlikely to give the best prediction for the missing values, but can certainly add to the completeness of the dataset (Little and Rubin, 2014). Imputation approaches can be broadly divided into *Single value imputation*, the *Hot deck and cold deck method*, and *Multiple imputation*.

1. Single value imputation: Single value imputation replaces the missing value with a single estimated value, which is assumed to be very close to the value that would have been observed if the dataset were complete. Various means for substituting a single value for the missing data are used. Some of these are as follows:
 - (i) Mean imputation: The mean of the existing values of an attribute is used to replace missing attribute value. This approach, reduces the diversity of data and therefore tends to produce a poorer estimation of the standard deviation of the dataset (Baraldi and Enders, 2010).
 - (ii) Regression imputation: A regression model can be used to predict observed values of a variable based on other variables, and that value is used to impute values in cases where that variable is missing. It preserves the variability in the dataset without introducing bias in the results. It overestimates the correlation between the attributes as compared to mean imputation, which underestimates this measure due to the loss of variability.
 - (iii) Least squares regression (LS impute): It is an extended form of the regression model. It is calculated by developing an equation for a function that minimizes the sum of squared errors from the model (Bø *et al.*, 2004).
 - (iv) Local least squares (LLS impute): It is similar to LS impute, except that it involves a *priori* step before applying the regression and estimation. It first represent a target gene that has missing values using a linear combination of similar genes by identifying the *K* nearest neighbors that have large absolute values of Pearson correlation coefficients (Kim *et al.*, 2004).
 - (v) K-Nearest Neighbor imputation (KNN impute): Missing data are replaced with the help of known data by minimizing the distance between the observed and estimated values of the missing data. The K-neighbors closest to the missing data are found, and the final result is estimated and replaces the missing value. This substitution of value depends on the type of missing data (Batista and Monard, 2002).
2. Hot deck and cold deck methods: Hot deck imputation involves filling in missing data on variables of interest from non-respondents (or recipients) using observed values from respondents (i.e., donors) within the same survey data set. Depending on the donor with which the similarity is computed, the hot deck approach, which is also known as Last Observation Carried Forward (LOCF), can be subdivided into *random hot deck prediction* and *deterministic hot deck prediction* (Andridge and Little, 2010). The hot deck approach offers certain advantages because it is free from parameter estimation and ambiguities in prediction of values. The most essential feature of this technique is that it uses only logical and credible values for estimating missing data, as these are obtained from the donor pool set (Andridge and Little, 2010). Cold-deck imputation, on the other hand, selects donors from another dataset to replace a missing value of a variable or data item with a constant value from an external donor.
3. Multiple imputation: Multiple imputation involves a series of calculations to decide upon the most appropriate value for the missing data. This technique replaces each missing value with a set of the most suitable values, which are then further analyzed to determine the final predicted value. It follows a monotonic missing pattern which means, for a sequence of values for a variable, if one value is missing then all subsequent values are also missing. Two common methods are used to find the set of suitable values.
 - Regression method: A regression model is fitted for attributes having missing values along with those from previous attributes in the sequence of dataset as covariates, following the monotone property. Based on this model, a new model is formed and this process is repeated sequentially for each attribute with missing values (Rubin, 2004).
 - Propensity score method: The propensity score is the conditional probability of assignment to a particular treatment given a set of observed covariates. In this method, a propensity score is calculated for each attribute with missing values to indicate the probability of that observation being missing. A Bayesian bootstrap imputation is then used on the grouped data based on the propensity score to get the set of values (Lavori *et al.*, 1995).

Duplicate Data Detection

Data redundancy due to the occurrence of duplicate data values for data instances or attributes leads to the issue of shortage of storage. The most common way of handling duplicacy is by finding chunks of similar values and removing the duplicates from the chunks. However, this method is very time consuming and hence a few other techniques have been developed for handling redundant data.

Knowledge-based methods

Incorporating domain dependent information from knowledge bases into the data cleaning task is one alternative for duplication elimination. A working tool based on this technique is Intelliclean (Low *et al.*, 2001). This tool standardizes abbreviations. Therefore, if one record abbreviates the word street as *St.*, and another abbreviates it as *Str*, while another record uses the full word, all three records are standardized to the same abbreviation. Once the data has been standardized, it is cleaned using a set of domain-specific rules that work with a knowledge base. These rules detect duplicates, merge appropriate records, and create various alerts for any other anomalies.

ETL method

The most popular method for duplicate elimination, these days, is the ETL method (Rahm and Do, 2000). The ETL method comprises three steps – extraction, translation, and loading. Two types of duplicate data elimination can be performed using this method – one at the instance-level and the other at the schema-level. Instance-level processing cleans errors within the data itself, such as misspellings. Schema-level cleaning usually works by transforming the database into a new schema or a data warehouse.

Data Reduction

Mining large scale data is time consuming and expensive in terms of memory. It may make the task of data mining impractical and infeasible. Using the entire dataset is not always important, and in fact, may contribute little to the quality of the outcome, compared to using a reduced version of it. Sometimes, the abundance of irrelevant data may lead to non-optimum results. Data reduction reduces the data either in terms of volume or the number of attributes (also called dimensions) or both, without compromising the integrity of the original data with regard to the results. A number of approaches are available for data reduction. Ideally, any data reduction method should be efficient and yet produce nearly identical analytical results to those obtained with the full data. Various approaches are briefly discussed below.

Parametric Data Reduction

In parametric data reduction, the volume of the original data is reduced by considering a relatively compact alternative way to represent the data. It fits a parametric model based on the data distribution to the data, and estimates the optimal values of the parameters required to represent that model. In this approach, only model parameters and the outliers are stored. Regression and non-linear models are two well-known approaches for parametric data reduction.

Unlike parametric, non-parametric approaches do not use any models. They summarize the data with sample statistics, as discussed below.

Sampling

Processing a large data set at one time is expensive, and time consuming. Instead of considering the whole data set, a small representative sample of k instances taken from a data set with G instances, where $|k| \leq |G|$. This process is called sampling. We can categorize sampling as follows.

- Simple random sampling without replacement: Here, a sample of size k is selected from a large data set of size $|G|$ with probability(p), and the samples, once chosen, are not placed back in the dataset. Sampling without replacement gives a non-zero covariance between two chosen samples, which complicates the computations. If the number of data instances are very large, the covariance is very close to zero.
- Simple random sampling with replacement: Here, the sample, once chosen, are placed back in the dataset and hence data may be duplicated in the sample. It gives a zero covariance between two chosen samples. In case of a skewed data distribution, simple random sampling with replacement usually produces poor results with any data analysis method. In such a case, sampling with replacement isn't much different from sampling without replacement. However, the precision of estimates is usually higher for sampling without replacement compared to sampling with replacement. Adaptive sampling methods such as stratified sampling and cluster sampling are likely to improve performance in the case of unbalanced datasets with relatively uneven sample distribution.
- Stratified sampling: In stratified sampling the original data is divided into strata (sub-groups), and from each stratum a sample is generated by simple random sampling.
- Cluster sampling: In cluster sampling, the original data is divided into clusters (sub-groups), and out of a total of N clusters, n are randomly selected, and from these clusters, elements are selected for the creation of the sample.

Dimension Reduction

Dimension or attribute reduction is simply the removal of unnecessary or insignificant attributes. The presence of irrelevant attributes in the dataset may make the problem intractable and mislead the analysis as well, which is often termed the *Curse of*

Dimensionality. Data sets with high dimensionality are likely to be sparse as well. Machine learning techniques such as clustering, which are based on data density and distance, may produce incorrect outcomes, in the case of sparse data. To overcome the low performance of machine learning algorithms with high dimensional data sets, dimension reduction is vital. Dimension reduction should be carried out in such a way that, as much as possible, the information content of the original data remains unchanged and that the use of the reduced dataset does not change the final outcome. Dimension reduction can be obtained by selecting only the relevant attributes and ignoring the rest. Principal Component Analysis (PCA) (Hotelling, 1933) is one of the most commonly used technique for dimensionality reduction. PCA is a statistical procedure that uses an orthogonal transformation to map a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It finds low dimensional approximations of the original data by projecting the data onto linear subspaces. It searches for k n -dimensional orthogonal vectors ($k < n$) from the original n dimensional data which best represent the original data. The original data are thus mapped onto a much smaller number of dimensions, resulting in dimensionality reduction.

Data Transformation and Normalization

Transformation is the process of converting data from one format or structure into another format or structure. It is applied so that the data may more closely meet the assumptions of a statistical inference procedure, or to improve interpretability. It uses a transformation function, $y=f(x)$, to change data from one domain to another. Often the term *normalization* is interchangeably used with transformation. Normalization means applying a transformation so that the transformed data are roughly normally distributed. Some transformation or normalization techniques are discussed below.

Log₂ Transformation

It is the most widely used transformation in the case of microarray (Quackenbush, 2002) or RNASeq data (Anders and Huber, 2010). It produces a continuous series of values for which it is easy to deal with extreme values, heteroskedasticity, and skewed data distributions.

Min-Max Normalization

It maps an attribute value, x , in the original dataset to a new value, x' , given by

$$x' = \frac{x - \min}{\max - \min}$$

Z-Score Normalization

It transforms the values of variables based on their mean and standard deviation. For a variable X represented by a vector $\{x_1, x_2, \dots, x_n\}$ each attribute can be transformed using the formula

$$x'_i = \frac{x_i - \bar{X}}{std_{dev}(\bar{X})}$$

where x'_i is the Z-score value of x , $\bar{X}$ is the row mean of X , std_{dev} is the standard deviation given by

$$std_{dev}(\bar{X}) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2}$$

Decimal Scaling Normalization

Decimal scaling moves the decimal point of a attribute value. The number of decimal points moved depends on the maximum absolute value of the attribute. A value v of an attribute is normalized to v' by following way:

$$v' = \frac{v}{10^j}$$

where j is the smallest integer such that $\max(v') < 1$. Suppose that the values of an attribute ranges from 523 to 237. The maximum absolute value is 523. To normalize by decimal scaling, it divides each value by 1000 (i.e., $j=3$) so that 523 normalizes to 0.523 and 237 normalizes to 0.237.

Quantile Normalization

It is a technique for making two or more distributions identical in terms of statistical properties. It is frequently used in microarray data analysis (Bolstad et al., 2003). To normalize gene expression data, it first assigns a rank to each column of values based on

lowest to highest values in that column. It then rearranges the column values based on their original values (before ranking), so that each column is in order, going from the lowest to the highest value. The average for each row is then calculated using the reordered values. The average value calculated in the first row will be lowest value (rank 1) from every column, the second average value will be the second lowest (rank 2) and so on. Finally, it replaces the original values with the average values based on the ranks assigned during first step. The new values in each column all have the same distribution.

The quantile normalization transforms the statistical distributions across samples to be the same and assumes there are no global differences in the distributions of the data in each column. However, it is not clear how to proceed with normalization if these assumptions are violated. Recently, a generalization of quantile normalization, referred to as smooth quantile normalization (qsmooth) (Hicks *et al.*, 2016), has been proposed that relaxes the assumptions by allowing differences in the distribution between the groups.

Data Discretization

Data collected from different sources are found in different formats depending on the experimental set-up, prevailing conditions, and variables of interest. To analyze such data, it is often convenient and effective to discretize the data (Dash *et al.*, 2011). Discretization transforms the domain of values for each quantitative attribute into discrete intervals. Using discrete values during data processing offers a number of benefits such as the following:

- Discrete data consumes less memory for storage.
- Discrete data are likely to be simpler to visualize and understand.
- Performance, in some systems, often becomes more efficient and accurate using discrete data.

The two terms frequently used in the context of discretization or interval generation are discussed below.

- Cut-points: Cut-points are certain values within the range of each discretized quantitative attribute. Each cut-point divides a range of continuous values into successive intervals (Liu *et al.*, 2002). For example, a continuous range $\langle a \dots b \rangle$ may be partitioned into intervals like $[a, c]$ and $[c, b]$, where the value c is a cut-point.
- Arity: Another term that is popular in discretization context is Arity. It is the total number of intervals generated for an attribute after discretization. Arity for an attribute is one more than the number of cut-points used for discretization (Liu *et al.*, 2002). An overly high Arity can make the learning process longer while a very low Arity may negatively affect the predictive accuracy.

Any discretization method follows the enumerated steps below.

1. Sort the domain of continuous values for each attribute to be discretized.
2. Evaluate the cut-points for each of these attributes.
3. Create intervals using the deduced cut-points.
4. Assign each value with the alternative value according to the interval in which it falls.

In order to accomplish this set of steps, a discretizer has to use a certain measure to evaluate cut-points and group different continuous values into separate discretized intervals. A few such measures are discussed below.

- Binning: The simplest method to discretize a continuous-valued attribute is by creating a pre-specified number of bins. These cut-points of the bins are determined by user input.
- Entropy: It is one of the most commonly used discretization measures in the literature. It is the average amount of information per event, where the information of an event is high for unlikely events and low otherwise. In every iteration, it calculates the entropy of the bin after splitting and calculates the net entropy of the split and the information gain (Ross Quinlan, 1986). It selects the split with the highest gain and iteratively partitions each split until the gain falls below a certain threshold.
- Dependency: It is a measure that finds the strength of association between a class and a attribute (depending upon certain computations or statistics) and accordingly determines the cut-points. In this measure, the Arity of each attribute has to be specified as a parameter.

Several discretization techniques have been proposed in the literature. They are broadly classified as Supervised and Unsupervised discretization techniques, depending on whether they use any class levels (Mahanta *et al.*, 2012). These are discussed below.

Supervised data discretization

Such discretization uses class labels to convert continuous data to discrete ranges derived from the original dataset (Dougherty *et al.*, 1995). These are of two types.

- (i) Entropy Based Discretization Method: This method of discretization uses the entropy measure to decide the boundaries. Chiu *et al.* (1990) proposed a hierarchical method that maximizes the Shannon entropy over the discretized space. The method starts with k -partitions and uses hill-climbing to optimize the partitions, using same entropy measure to obtain finer intervals.

- (ii) Chi-Square Based Discretization: It uses the statistical significance test (Chi-Square test) (Kerber, 1992) to determine the probability of the similarity of data in two intervals. It starts with every unique value of the attribute in its own interval. It computes χ^2 for each initial interval. Next, it merges the intervals with the smallest χ^2 values. It repeats the process of merging until no more satisfactory merging is possible based on the χ^2 values.

Unsupervised data discretization

The unsupervised data discretization process does not use any class information to decide upon the boundaries. A few discretization techniques are discussed below.

- (i) Average and Mid-Ranged value discretization: A binary discretization technique that divides the class boundaries using the average value of the data. Values less than the average correspond to one class, and values greater than the average correspond to the other class (Dougherty et al., 1995).

For example, given $A = \{23.73, 5.45, 3.03, 10.17, 5.05\}$ the average discretization method first calculates the average score, $\bar{A} = \sum_{i=1}^n A_i$. Based on this value, it discretizes the vector using following equation.

$$D_i = \begin{cases} 1, & \text{if } A(i) > \bar{A} \\ 0, & \text{otherwise} \end{cases}$$

In our example, $\bar{A} = 9.486$, and therefore, the discretized vector using average-value discretization is $D = [1 \ 0 \ 0 \ 1 \ 0]$.

On the other hand, mid-range discretization uses the middle or mid-range value of an attribute in the whole dataset to decide the class boundary. The mid-range of a vector of values is obtained using the equation

$$M = (H + U) / 2$$

where, H is the maximum value and U is the minimum value in the vector. Discretization of values occurs as follows.

$$D_i = \begin{cases} 1, & \text{if } A(i) > M \\ 0, & \text{otherwise} \end{cases}$$

For the vector A , we obtain M as 13.38. Therefore, the corresponding discretized vector of values are $D = [1 \ 0 \ 0 \ 0 \ 0]$. It is not of much use as it lacks efficiency and robustness (outliers change it significantly) (Dougherty et al., 1995).

- (ii) Equal Width Discretization: Divides the data into k equal intervals. The upper and lower bounds of the intervals are decided by the difference between the maximum and minimum values for the attribute of the dataset (Catlett, 1991).

The number of intervals, k , is specified by the user, and the lower and upper boundaries, p_r and p_{r+1} of the class are obtained from the data (H and U , respectively) in the vector according to the equation

$$p_{r+1} = p_r + (H - U) / k$$

For our example, if $k=3$, the class division occurs as given below.

$$D_i = \begin{cases} 1, & \text{if } p_{r_0} < A(i) < p_{r_1} \\ 2, & \text{if } p_{r_1} < A(i) < p_{r_2} \\ 3, & \text{if } p_{r_2} < A(i) < p_{r_3} \end{cases}$$

Therefore, the corresponding discretized vector is $D = [3 \ 1 \ 1 \ 2 \ 1]$.

Some of the frequently used discretizers, out of the plethora of techniques available in literature, are reported in Table 1.

Table 1 Some discretizers and their properties

Discretizer	Measure used	Procedure adopted	Learning model
Equal width	Binning	Splitting	Un-supervised
Equal frequency	Binning	Splitting	Un-supervised
ChiMerge (Kerber, 1992)	Dependency	Merging	Supervised
Chi2 (Liu and Setiono, 1997)	Dependency	Merging	Supervised
Ent-minimum description length principle (MDLP) (Fayyad and Irani, 1993)	Entropy	Splitting	Supervised
Zeta (Ho and Scott, 1997)	Dependency	Splitting	Supervised
Fixed frequency	Binning	Splitting	Un-supervised
Discretization (FFD) (Yang and Webb, 2009)			
Optimal flexible frequency discretization (OFFD) (Wang et al., 2009)	Wrapping	Hybrid	Supervised
Class-attribute interdependence	Dependency	Splitting	Supervised
Maximization (CAIM) (Kurgan and Cios, 2004)			
Class attribute dependent discretization (CADD) (Ching et al., 1995)	Dependency	Hybrid	Supervised
Ameva (Gonzalez-Abril et al., 2009)	Dependency	Splitting	Supervised

Data Integration

Biological data deal with properties of living organisms, i.e., data are associated with metabolic mechanisms. Such data result from the combined effect of complex mechanisms occurring in the cell, from DNA to RNA to Protein, and further to the metabolite level. Understanding cellular mechanisms from all these perspectives is likely to produce more biologically relevant results than simply analyzing them from mathematical or computational viewpoints. Such an analysis involves data integration. Data integration is the process of analyzing data obtained from various sources, and obtaining a common model that explains the data (Schneider and Jimenez, 2012). Such a model fits the prevailing conditions better and makes more accurate predictions. Literature has shown that the use of data integration in the biological sciences has been effective in tasks such as identifying metabolic and signaling changes in cancer by integrating gene expression and pathway data (Emmert-Streib and Glazko, 2011), thereby identifying protein related disorders through combined analysis of genetic as well as genomic interactions (Bellay *et al.*, 2011).

From the systems biological perspective, integration of data can be achieved at three levels, viz., at the data level, at the processing level, or at the decision level (Milanesi *et al.*, 2009). Integrating data at the first level requires combining data from heterogeneous sources and implementing a universal query system for all types of data involved. This is the most time-consuming step in data integration. One needs to consider various assumptions and experimental setup information before combining the data. The second level of data integration involves an understanding and interpretation of the datasets. In this stage of integration, one needs to identify associated correlations between the various datasets involved. The third stage of data integration is at the decision level. In this stage, different participating datasets are first dealt with using specific procedures and individual results are obtained. Finally, these individual results are mapped, according to some ontology or information known *a priori* to interpret the results with more confidence.

Conclusions

Real-life data tend to be incomplete in nature. To handle incompleteness of different forms such as noise, missing values, inconsistencies, and the curse of dimensionality, data preparation steps are essential. We presented a brief discussion of different data preprocessing steps required to make data suitable for analysis. It is well-known that real data are not static in nature and data distributions vary with time. State-of-the-art preprocessing steps are not appropriate to handle such changes of data distribution or characteristics. Dealing with dynamic data is particularly challenging when data are produced rapidly. An empirical study on the suitability and efficiency of pre-processing techniques applied to dynamic data is of great interest for the big data community.

See also: Data Mining in Bioinformatics. Knowledge Discovery in Databases. The Challenge of Privacy in the Cloud

References

- Allison, P.D., 2012. Handling missing data by maximum likelihood. SAS Global Forum, Paper 312-2012. Available at: <http://www.statisticalhorizons.com/wp-content/uploads/MissingDataByML.pdf>.
- Anders, S., Huber, W., 2010. Differential expression analysis for sequence count data. *Genome Biology* 11 (10), R106.
- Andridge, R.R., Little, R.J., 2010. A review of hot deck imputation for survey non-response. *International Statistical Review* 78 (1), 40–64.
- Baraldi, A.N., Enders, C.K., 2010. An introduction to modern missing data analyses. *Journal of School Psychology* 48 (1), 5–37.
- Batista, G.E.A.P.A., Monard, M.C., 2002. A study of K-nearest neighbour as an imputation method. *HIS* 87 (251-260), 48.
- Bellay, J., Han, S., Michaut, M., *et al.*, 2011. Bringing order to protein disorder through comparative genomics and genetic interactions. *Genome Biology* 12 (2), R14.
- Bø, T.H., Dysvik, B., Jonassen, I., 2004. LSImpute: Accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research* 32 (3), e34.
- Bolstad, B.M., Irizarry, R.A., Åstrand, M., Speed, T.P., 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19 (2), 185–193.
- Catlett, J., 1991. On changing continuous attributes into ordered discrete attributes. In: *Machine Learning, EWSL-91, LNAI*, vol. 26, pp. 164–178.
- Ching, J.Y., Wong, A.K.C., Chan, K.C.C., 1995. Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (7), 641–651.
- Chiu, D.K.Y., Cheung, B., Wong, A.K.C., 1990. Information synthesis based on hierarchical maximum entropy discretization. *Journal of Experimental & Theoretical Artificial Intelligence* 2 (2), 117–129.
- Dash, R., Paramguru, R.L., Dash, R., 2011. Comparative analysis of supervised and unsupervised discretization techniques. *International Journal of Advances in Science and Technology* 2 (3), 29–37.
- Daz-Uriarte, R., De Andres, S.A., 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7 (1), 3.
- Dougherty, J., Kohavi, R., Sahami, M., *et al.*, 1995. Supervised and unsupervised discretization of continuous features. *Machine Learning: Proceedings of the Twelfth International Conference* 12, 194–202.
- Emmert-Streib, F., Glazko, G.V., 2011. Pathway analysis of expression data: Deciphering functional building blocks of complex diseases. *PLOS Computational Biology* 7 (5), e1002053.
- Fayyad, U., Irani, K., 1993. Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. 1022–1027.
- Gonzalez-Abril, L., Cuberos, F.J., Velasco, F., Ortega, J.A., 2009. Ameva: An autonomous discretization algorithm. *Expert Systems With Applications* 36 (3), 5327–5332.
- Herbert, K.G., Wang, J.T.L., 2007. Biological data cleaning: A case study. *International Journal of Information Quality* 1 (1), 60–82.
- Hicks, S.C., Okrah, K., Paulson, J.N., *et al.*, 2016. Smooth quantile normalization. *BioRxiv*. 085175.

- Ho, K.M., Scott, P.D., 1997. Zeta: A global method for discretization of continuous variables. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD99), Newport, USA, pp. 191–194, AAAI Press.
- Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24 (6), 417.
- Hyun, K., 2013. The prevention and handling of the missing data. *Korean Journal of Anesthesiology* 64 (5), 402–406.
- Kerber, R., 1992. Chimerge: Discretization of numeric attributes. In: Proceedings of the tenth national conference on Artificial intelligence, pp. 123–128, AAAI Press.
- Kim, H., Golub, G.H., Park, H., 2004. Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics* 21 (2), 187–198.
- Kurgan, L.A., Cios, K.J., 2004. CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering* 16 (2), 145–153.
- Lavori, P.W., Dawson, R., Shera, D., 1995. A multiple imputation strategy for clinical trials with truncation of patient data. *Statistics in Medicine* 14 (17), 1913–1925.
- Little, R.J.A., Rubin, D.B., 2014. *Statistical Analysis With Missing Data*. John Wiley & Sons.
- Liu, H., Setiono, R., 1997. Feature selection via discretization. *IEEE Transactions on Knowledge and Data Engineering* 9 (4), 642–645.
- Liu, H., Hussain, F., Tan, C.L., Dash, M., 2002. Discretization: An enabling technique. *Data Mining and Knowledge Discovery* 6 (4), 393–423.
- Low, W.L., Lee, M.L., Ling, T.W., 2001. A knowledge-based approach for duplicate elimination in data cleaning. *Information Systems* 26 (8), 585–606.
- Mahanta, P., Ahmed, H.A., Kalita, J.K., Bhattacharyya, D.K., 2012. Discretization in gene expression data analysis: A selected survey. In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, pp. 69–75, ACM.
- Milanesi, L., Alfieri, R., Mosca, E., *et al.*, 2009. Sys-Bio Gateway: A framework of bioinformatics database resources oriented to systems biology. In: Proceedings of International Workshop on Portals for Life Sciences (IWPLS'09), Edinburgh, UK, CEUR.
- Moon, T.K., 1996. The expectation-maximization algorithm. *IEEE Signal Processing Magazine* 13 (6), 47–60.
- Oba, S., Sato, M.-A., Takemasa, I., *et al.*, 2003. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* 19 (16), 2088–2096.
- Quackenbush, J., 2002. Microarray data normalization and transformation. *Nature Genetics* 32, 496–501.
- Rahm, E., Do, H.H., 2000. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin* 23 (4), 3–13.
- Ross Quinlan, J., 1986. Induction of decision trees. *Machine Learning* 1 (1), 81–106.
- Roy, S., Bhattacharyya, D.K., Kalita, J.K., 2013. CoBi: Pattern based co-regulated biclustering of gene expression data. *Pattern Recognition Letters* 34 (14), 1669–1678.
- Rubin, D.B., 2004. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons.
- Schneider, M.V., Jimenez, R.C., 2012. Teaching the fundamentals of biological data integration using classroom games. *PLoS Computational Biology* 8 (12), e1002789.
- Wang, S., Min, F., Wang, Z., Cao, T., 2009. OFFD: Optimal flexible frequency discretization for naive Bayes classification. *Advanced Data Mining and Applications*. 704–712.
- Yang, Y., Webb, G.I., 2009. Discretization for naive-Bayes learning: Managing discretization bias and variance. *Machine Learning* 74 (1), 39–74.

Data Cleaning

Barbara Calabrese, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Raw experimental data is highly susceptible to noise (i.e., contain errors or outliers), missing values (i.e., data lack attribute values, lack certain attributes of interest, or contain only aggregate data) and inconsistency (i.e., data contain discrepancies in codes or names). The representation and quality of the data is crucial because the quality of data affects the data analysis pipeline and thus results. Irrelevant and redundant information in the data, or noisy and unreliable data, could prevent the correct analysis (Han *et al.*, 2012).

Data pre-processing is a fundamental step in the data analysis process to limit such problems. Data pre-processing methods are divided into following categories: (i) data cleaning, that aims to fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies; (ii) data integration, i.e., the merging of data from multiple data stores; (iii) data transformation, that includes normalization and aggregation methodologies, and (iv) data reduction, that attempts to reduce the volume, but produces the same or similar analytical results. In the following paragraphs, the main data cleaning methodologies are presented and discussed.

Data Cleaning

Data cleaning includes all methodologies whose aims are “detecting and removing errors and inconsistencies from data in order to improve the quality of data” (Lenzerini, 2002). Specifically, data quality methods “clean” the data by filling in missing values, smoothing noisy data, identifying or removing outliers and resolving inconsistencies (Van den Broeck *et al.*, 2005). Instance-level cleaning refers to errors within the data itself; schema level usually concerns integrating the databases into a new schema. Data cleaning is the major part of ETL (extraction, translation and loading) process in data warehouse.

Data Cleaning Methods for Missing Values in Bioinformatics

The presence of missing data may depend on (Herbert and Wang, 2007):

- Malfunctions of data collection systems;
- Inconsistency with values of other data set attributes;
- Not entered data, due to “misunderstanding”;
- Some data may not be considered important at the time of insertion;
- Failure to register changes in data.

There are several approaches to solve missing the data problems. A first approach is “do not do anything”. Some analysis algorithms are quite robust and insensitive to missing data. Another approach is to eliminate the records that contain missing data: this could introduce a distortion in the data. Otherwise, if only few columns have these characteristics it is better to ignore, but this is not very effective. An alternative approach is to predict new values by using different methods. It is possible to use mean or median value or to pad unknown values with zeros. These methods could cause some big imputation errors. To reduce these errors, more sophisticated methods, such as classification/regression techniques, could be used to estimate missing values.

In Liew *et al.* (2011), a comprehensive survey about missing value estimation techniques for microarray gene expression data is reported. A first approach for missing value estimation exploits the correlation structure between entries in the data matrix. In fact, in gene expression data matrix, rows are correlated because the genes involved in similar cellular processes usually have similar expression profiles. Moreover, there is correlation between columns because the set of genes is expected to behave similarly under similar conditions. Thus, missing values estimation can be based on subset of related genes or subset of related conditions. Another approach exploits the domain knowledge about the data or the process that generated the data. A recent work (Wei *et al.*, 2018) present a review and comparison of eight missing values imputation techniques (zero, half minimum, mean, median, random forest, singular value decomposition, KNN and QRILC quantile regression imputation of left-censored data) for different types of missing values using metabolomics datasets. Generally, missing values imputation methodologies could be grouped in the following classes:

- Global approach-based algorithms;
- Local approach-based algorithms;
- Hybrid approach-based algorithms;
- Knowledge assisted approach-based algorithms.

Global Approach-Based Algorithms

Global approach-based algorithms perform missing value estimation analysing the global correlation information of the entire data matrix. The most used algorithms are the Single Value Decomposition (*SVDimpute*) method and the Bayesian Principal Component Analysis (*BPCA*) method (Oba *et al.*, 2003).

SVD method employs singular value decomposition to obtain a set of mutually orthogonal expression patterns that can be linearly combined to approximate the expression of all genes in the data set (Troyauskaya *et al.*, 2001). These patterns, which in this case are identical to the principle components of the gene expression matrix, are referred to as eigengenes. *SVDimpute* first regresses the gene against the k most significant eigengenes and then use the coefficients of the regression to reconstruct the missing values from a linear combination of the k eigengenes.

In *BPCA*, the N -dimensional gene expression vectors γ is expressed as a linear combination of k principal axis vectors v_l . The factor scores w_l and the residual error ϵ are regarded as normally distributed random variables in the probabilistic PCA model (Eq. (1)).

$$\gamma = \sum_k w_l v_l + \epsilon \quad (1)$$

An EM-like algorithm is then used to estimate the posterior distributions of the model parameter and the missing values simultaneously (Liew *et al.*, 2011).

Local Approach-Based Algorithms

The second class of algorithms examine only local similarity structure in the dataset in order to estimate missing value. The most used algorithm is KNN (k -Nearest Neighbors). The KNN-based method (*KNNimpute*) selects genes with expression profiles similar to the gene of interest to impute missing values. If we consider gene A that has one missing value in experiment 1, this method would find K other genes, which have a value present in experiment 1, with expression most similar to A in experiments 2– N (where N is the total number of experiments). A weighted average of values in experiment 1 from the K closest genes is then used as an estimate for the missing value in gene A (Troyauskaya *et al.*, 2001). In the weighted average, the contribution of each gene is weighted by similarity of its expression to that of gene A. Euclidean distance is the most accurate metrics for gene similarity. In Troyauskaya (2001), KNN- and SVD-based methods were compared for DNA microarray missing value estimation. Both methods provide fast and accurate ways of estimating missing values for microarray data and far surpass the traditional accepted solutions, such as filling missing values with zeros or row average, by taking advantage of the correlation structure of the data to estimate missing expression values. Nevertheless, the authors recommend KNN-based method for imputation of missing values, because it shows less deterioration in performance with increasing percent of missing entries. In addition, the *KNNimpute* method is more robust than SVD to the type of data for which estimation is performed, performing better on non-time series or noisy data. *KNNimpute* is also less sensitive to the exact parameters used (number of nearest neighbors), whereas the SVD-based method shows sharp deterioration in performance when a non-optimal fraction of missing values is used.

A number of local imputation algorithms that use the concept of least square regression to estimate the missing values, have been proposed. In least square imputation (*LSimpute*) (Bo *et al.*, 2004), the target gene γ and the reference gene x are assumed to be related by a linear regression model. *LSimpute* first select the K most correlated genes based on absolute Pearson correlation values. Then a least square estimate of the missing value is obtained from each of the K selected genes using single regression. Finally, the K estimates are linearly combined to form the final estimate.

Unlike *LSimpute*, local least square imputation (*LLSimpute*) (Kim *et al.*, 2005) uses a multiple regression model to impute the missing values from all K reference genes simultaneously. Despite its simplicity, *LLSimpute* has been shown to be highly competitive compared to *KNNimpute* and *BPCA* (Brock *et al.*, 2008).

Sequential *LLSimpute* (*SLLSimpute*) is an extension of *LLSimpute* (Zhang *et al.*, 2008). The imputation is performed sequentially starting from the gene with the least missing rate, and the imputed genes are then used for later imputation of other genes. However, only genes with missing rate below a certain threshold are reused since genes with many imputed missing values are less reliable. *SLLSimpute* has been shown to exhibit better performance than *LLSimpute* due to the reuse of genes with missing values.

In iterated *LLSimpute* (*ILLSimpute*) (Cai *et al.*, 2006), different target genes are allowed to have different number of reference genes. The number of reference genes is chosen based on a distance threshold which is proportional to the average distance of all other genes to the target gene. *ILLSimpute* iteratively refines the imputation by using the imputed results from previous iteration to re-select the set of coherent genes to re-estimate the missing values until a preset number of iterations is reached. *ILLSimpute* has been shown to outperform the basic *LLSimpute*, *KNNimpute* and *BPCA* due to these modifications.

In Gaussian mixture clustering imputation (*GMCimpute*) (Ouyang *et al.*, 2004), the data is clustered into S components Gaussian mixtures using the EM algorithm. Then the S estimates of the missing value, one from each component, are averaged to obtain the final estimate of the missing value. The clustering and estimation steps are iterated until the cluster memberships of two consecutive iterations are identical. *GMCimpute* uses the local correlation information in the data through the mixture components.

In Dorri *et al.* (2012) an algorithm that is based on conjugate gradient (CG) method is proposed to estimate missing values. k -nearest neighbors of the missed entry are selected based on absolute values of their Pearson correlation coefficient. Then a subset of genes among the k -nearest neighbors is labeled as the best similar ones. CG algorithm with this subset as its input is then used to estimate the missing values.

MINMA (Missing data Imputation incorporating Network and adduct ion information in Metabolomics Analysis) (Jin *et al.*, 2017) implements a missing value imputation algorithm for liquid chromatography-mass spectrometry (LC MS) metabolomics. MINMA is an R package whose algorithm combines the afore-mentioned information and traditional approaches by applying the support vector regression (SVR) algorithm to a predictor network newly constructed among the features. The software provides a function to match feature m/z values to about 30 positive adduct ions, or over 10 negative adduct ions.

Hybrid Approach-Based Algorithms

The correlation structure in the data affects the performance of imputation algorithms. If the data set is heterogeneous, local correlation between genes are dominant and localized imputation algorithms such as KNNimpute or LLSimpute perform better than global imputation methods such as BPCA or SVDimpute. On the other hand, if the data set is more homogenous, a global approach such as BPCA or SVDimpute would better capture the global correlation information in the data (Liew *et al.*, 2011). Jornsten *et al.* (2005) proposes a hybrid approach called *LinCmb* that captures both global and local correlation information in the data.

In *LinCmb*, the missing values are estimated by a convex combination of the estimates of five different imputation methods: row average, KNNimpute and GMCimpute, that use local correlation information in the estimation of missing values, and SVDimpute and BPCA that are global-based methods for missing values imputation.

To obtain the optimal set of weights that combine the five estimates, *LinCmb* generates fake missing entries at positions where the true values are known and uses the constituent methods to estimate the fake missing entries. The weights are then calculated by performing a least square regression on the estimated fake missing entries. The final weights for *LinCmb* are found by averaging the weights obtained in 30 iterations (Liew *et al.*, 2011).

Knowledge Assisted Approach-Based Algorithms

The algorithms that belong to this category exploit the integration of domain knowledge or external information into the missing values estimation process. The use of domain knowledge has the potential to significantly improve the estimation accuracy, especially for data sets with small number of samples, noisy, or with high missing rate (Liew *et al.*, 2011). Knowledge assisted approach-based algorithms can make use of, for example, knowledge about the biological process in the microarray experiment (Gan *et al.*, 2006), knowledge about the underlying biomolecular process as annotated in Gene Ontology (GO) (Tuikkala *et al.*, 2006), knowledge about the regulatory mechanism (Xiang *et al.*, 2008), information about spot quality in the microarray experiment (Johansson and Hakkinen, 2006), and information from multiple external data sets (Sehgal *et al.*, 2008).

Data Cleaning Methods for Noisy Data in Bioinformatics

Noise is a random error or variance in a measured variable. In biological experiments, such as genomics, proteomics or metabolomics, noise could arise from human errors and/or variability of the system itself. An accurate experimental design and technological advances help to reduce noise, but some uncertainty remains always. Specifically, noisy data could include:

- Input errors due to operators;
- Incorrect data related to transcription operations;
- Input errors due to programs;
- Incorrect data related to programming errors;
- Errors due at a glance of insertion;
- Data stored in different formats for the same attribute.

In the following some methods to manage and pre-process noisy data are described. Binning methods smooth a sorted data value by examining the value of its “neighbourhood”. In smoothing by *bin means*, each value in a bin is replaced by the mean value of the bin. Similarly, smoothing by *bin medians* can be employed, in which each bin value is replaced by the bin median. In smoothing by *bin boundaries*, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value. In general, the larger the width, the greater the effect of the smoothing (Han *et al.*, 2012).

Data smoothing can also be performed by regression. Linear regression involves finding the “best” line to fit two attributes (or variables) so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fitted to a multidimensional surface.

Clustering techniques are also applied to noise detection tasks. Clustering algorithms, used for biological data reduction are particularly sensitive to the noise. In Sloutsky *et al.* (2012), a detailed study relative to the sensitivity of clustering algorithms to noise has been presented, by analysing two different case studies (gene expression and protein phosphorylation).

Another approach employs Machine Learning (ML) classification algorithms, which are used to detect and remove noisy examples.

The work presented in Libralon *et al.* (2009) proposes the use of distance-based techniques for noise detection. These techniques are named distance-based because they use the distance between an example and its nearest neighbors. The most popular distance-based technique is the k-nearest neighbor (k-NN) algorithm. Distance-based techniques use similarity measures to calculate the distance between instances from a data set and use this information to identify possible noisy data. Distance-based techniques are simple to implement and do not make assumptions about the data distribution. However, they require a large amount of memory space and computational time, resulting in a complexity directly proportional to data dimensionality and number of examples., which is the simplest algorithm belonging to the class.

Libralon *et al.* stated that for high dimensional data sets, the commonly used Euclidian metric is not adequate, since data is commonly sparse. Thus, they use the HVDM (Heterogeneous Value Difference Metric) metric to deal with high dimensional data. This metric is based on the distribution of the attributes in a data set, regarding their output values, and not only on punctual values, as is observed in the Euclidian distance and other similar distance metrics.

Outlier Detection

Outliers detection is a fundamental step in the preprocessing stage aiming to prevent wrong results. To detect anomalous measurements and/or observations from normal ones, data mining techniques are widely used (Oh and Gao, 2009). Generally, statistical methods often view objects that are located relatively far from the center of the data distribution as outlier. Several distance measures were implemented, such as Mahalanobis distance. Distance-based algorithms are advantageous since model learning is not required.

Outliers may be detected by clustering, for example, where similar values are organized into groups, or clusters. Values that fall outside of the set of clusters may be considered outliers. In Wang (2008), the authors proposed an effective cluster validity measure with outlier detection and cluster merging strategies for support vector clustering. In Oh and Gao (2009), the authors present an outlier detection method based on KL divergence. Angiulli *et al.* (2006) proposed a distance-based outlier detection method which finds the top outliers and provides a subset of the dataset called outlier detection solving set, that can be used to predict if new unseen objects are outliers.

Concluding Remarks

Data cleaning is a crucial step in data analysis pipeline because errors and noise can affect the quality of collected data, preventing an accurate subsequent analysis. Several methods could be used for data cleaning according to the specific required task.

See also: Data Mining in Bioinformatics. Knowledge Discovery in Databases. The Challenge of Privacy in the Cloud

References

- Angiulli, F., Basta, S., Pizzuti, C., 2006. Distance-based detection and prediction of outliers. *IEEE Transactions on Knowledge and Data Engineering* 18, 145–160.
- Bø, T.H., Dysvik, B., Jonassen, I., 2004. LSImpute: Accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research* 32, e34.
- Brock, G.N., Shaffer, J.R., Blakesley, R.E., *et al.*, 2008. Which missing value imputation method to use in expression profiles: A comparative study and two selection schemes. *BMC Bioinformatics* 9, 12.
- Cai, Z., Heydari, M., Lin, G., 2006. Iterated local least squares microarray missing value imputation. *Journal of Bioinformatics and Computational Biology* 45, 935–957.
- Dorri, F., Azmi, P., Dorri, F., 2012. Missing value imputation in DNA microarrays based on conjugate gradient method. *Computers in Biology and Medicine* 42, 222–227.
- Gan, X., Liew, A.W., Yan, H., 2006. Microarray missing data imputation based on a set theoretic framework and biological knowledge. *Nucleic Acids Research* 34, 1608–1619.
- Han, J., Kamber, M., Pei, J., 2012. *Data Mining: Concepts and Techniques*. Elsevier.
- Herbert, K., Wang, J.T., 2007. Biological data cleaning: A case study. *International Journal of Information Quality* 1, 60–82.
- Jin, Z., Kang, J., Yu, T., 2017. Missing value imputation for LC-MS metabolomics data by incorporating metabolic network and adduct ion relations. *Bioinformatics* 1–7.
- Johansson, P., Hakkinen, J., 2006. Improving missing value imputation of microarray data by using spot quality weights. *BMC Bioinformatics* 7, 306.
- Jornsten, R., Wang, H.Y., Welsh, W.J., *et al.*, 2005. DNA microarray data imputation and significance analysis of differential expression. *Bioinformatics* 21, 4155–4161.
- Kim, H., Golub, G.H., Park, H., 2005. Missing Value Estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics* 21, 187–198.
- Lenzerini, M., 2002. Data integration: A theoretical perspective. In: *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 233–246. Madison, Wisconsin.
- Libralon, G.L., Ferreira de Carvalho, A.C., Lorena, A.C., 2009. Pre-processing for noise detection in gene expression classification data. *Journal of the Brazilian Computer Society* 15, 3–11.
- Liew, A., Law, N., Yan, H., 2011. Missing value imputation for gene expression data: Computational techniques to recover missing data from available information. *Briefings in Bioinformatics* 12, 498–513.
- Oba, S., *et al.*, 2003. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* 19, 2088–2096.
- Oh, J.H., Gao, J., 2009. A kernel-based approach for detecting outliers of high-dimensional biological data. *BMC Bioinformatics* 10 (Suppl. 4), S7.
- Ouyang, M., Welsh, W.J., Georgopoulos, P., 2004. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics* 20, 917–923.
- Sehgal, M.S., Gondal, I., Dooley, L.S., *et al.*, 2008. Ameliorative missing value imputation for robust biological knowledge inference. *Journal of Biomedical Informatics* 41, 499–514.
- Sloutsky, R., *et al.*, 2012. Accounting for noise when clustering biological data. *Briefings in Bioinformatics* 14, 423–436.
- Troyanskaya, O., *et al.*, 2001. Missing value estimation methods for DNA microarray. *Bioinformatics* 17, 520–525.
- Tuikkala, J., Elo, L., Nevalainen, O., *et al.*, 2006. Improving missing value estimation in microarray data with gene ontology. *Bioinformatics* 22, 566–572.

- Van den Broeck, J., Argeseanu Cunningham, S., Eeckels, R., Herbst, K., 2005. Data cleaning: Detecting, diagnosing, and editing data abnormalities. *PLOS Medicine* 2, e267.
- Xiang, Q., Dai, X., Deng, Y., *et al.*, 2008. Missing value imputation for microarray gene expression data using histone acetylation information. *BMC Bioinformatics* 9, 252.
- Wei, R., *et al.*, 2018. Missing value imputation approach for mass spectrometry based metabolomics data. *Scientific Reports* 8.
- Zhang, X., Song, X., Wang, H., *et al.*, 2008. Sequential local least squares imputation estimating missing value of microarray data. *Computers in Biology and Medicine* 38, 1112–1120.

Data Integration and Transformation

Barbara Calabrese, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Recent technological advances in high throughput biology, have generated vast amounts of omics data. Data integration and data transformation are two fundamental steps of biological data preprocessing. Specifically, data integration can refer to different levels. For example it regards the merging of data from multiple data stores. Careful integration can help reduce and avoid redundancies and inconsistencies in the resulting data set (Han, 2012). This can help improve the accuracy and speed of the subsequent data mining process. Data integration is not a simple task, but it poses several issues, such as entity identification problem, redundancy and inconsistencies in attribute or dimension. In the following paragraphs, a further discussion about different levels of data integration is reported.

In the data transformation, the data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand. The main methods for data transformation are illustrated in this contribution.

Data Integration

Data integration can occur at different levels. The integration of data of the same type (e.g., nucleotide sequence or protein sequence) enables the user to use a single interface to access all possible data for analysis. Moreover, the integration of data of different types for analysis allows a deeper understanding of biological phenomena. Specifically, data integration collectively analyses all datasets and builds a joint model that captures all datasets. Thus, data integration process is crucial for the advancements in genomics, proteomics, metabolomics and, generally speaking, biological research because a comprehensive understanding of a biological system can come only from a joint analysis of all omics layers (i.e., genome, proteome, metabolome, etc...) Wanichthanarak *et al.* (2015).

Data integration methodologies have to meet many computational challenges. These challenges arise owing to different sizes, formats and dimensionalities of the data being integrated, as well as owing to their complexity, noisiness, information content and mutual concordance (i.e., the level of agreement between datasets) Gomez-Cabrero *et al.* (2014).

Schemas Integration

As pointed before, the aim of data integration in bioinformatics is to establish automated and efficient methods to integrate large, heterogeneous biological datasets from multiple sources. Biological databases are geographically distributed and heterogeneous in terms of their functions, structures, data access methods and dissemination formats (Zhang *et al.*, 2011). According to their functions, databases can be categorized into different classes: (i) sequence databases, e.g., GenBank (Benson *et al.*, 2015), RefSeq (O'Leary *et al.*, 2016); (ii) functional genomics databases, e.g., ArrayExpress (Kolesnikov *et al.*, 2014); (iii) protein-protein interaction databases, e.g., DIP (Database of Interacting Proteins) (Salwinski *et al.*, 2004), IntAct (Orchard *et al.*, 2013); (iv) pathway databases, e.g., KEGG (Kyoto Encyclopedia of Genes and Genomes) (Kanehisa *et al.*, 2017); (v) structure databases, e.g., CATH (Sillitoe *et al.*, 2015; Dawson *et al.*, 2017); (vi) annotation databases, e.g., GO (Gene Ontology) (Ashburner *et al.*, 2000).

In Lapatas *et al.* (2015), computational solutions allowing users to fetch data from different sources, combine, manipulate and re-analyse them as well as being able to create new datasets, have been discussed. Specifically, general computational schemas used to integrate data in biology include data centralisation, federated databases, linked data, service-oriented integration.

Methodologies and Tools for Integrative Analysis

It is becoming evident that integrative analyses across multiple omic platforms are needed to understand complex biological systems. Over the past several years, enrichment analyses methods such as gene set enrichment analysis (GSEA) have been widely used to analyse gene expression data. These methods integrate biological domain knowledge (e.g., biochemical pathways, biological processes) with gene expression results (Gligorivic and Przulj, 2015).

Network-based analyses are used to study a variety of organismal and cellular mechanisms. Biological networks represent complex connections among diverse types of cellular components such as genes, proteins, and metabolites.

Correlation analysis is useful for omic data integration when there is a lack of biochemical domain knowledge and to integrate biological and other meta data (Gligorivic and Przulj, 2015).

Data Transformation

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Strategies for data transformation include the following (Han, 2012):

- Smoothing, which works to remove noise from the data. Techniques include binning, regression, and clustering;
- Aggregation, where summary or aggregation operations are applied to the data;
- Normalization, where the attribute data are scaled so as to fall within a smaller range, such as -1.0 to 1.0 , or 0.0 – 1.0 ;
- Discretization, where the raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior).

In the subsequent subparagraphs, the main normalization and discretization techniques are reported.

Normalization

Normalization is required to avoid that the data model is not influenced by high or low values and falls into small specified range (Purshit *et al.*, 2018). There are several normalization techniques, such as:

- Min-max normalization. The following equation (Eq. (1)) refers to min-max normalization:

$$v'_i = \frac{v_i - \text{Min}_A}{\text{Max}_A - \text{Min}_A} (\text{newMax}_A - \text{newMin}_A) + \text{newMin}_A \quad (1)$$

where A is the input array, v_i is the i -th input value, Min_A e Max_A are respectively minimum and maximum values in the input array A . newMax_A e newMin_A are 1 and 0 since the mapping of original values is being to range between 0 and 1.

- Zero mean normalization. The equation for this type of normalization is (Eq. (2)):

$$z = \frac{x - \mu}{\sigma} \quad (2)$$

where x is the value in an array, μ is the mean, σ is the standard deviation of the array.

- Normalization by decimal scaling. Decimal scaling is the method which provides the range between -1 and 1 , according to the following formula (Eq. (3)):

$$v'_i = \frac{V}{10^j} \quad (3)$$

v'_i is scaled value, V is the value in an array in i th position and j is the smallest integer as $\max(|v_i|) < 1$.

Discretization

Data discretization aims to map the real data into a typically small numbers of finite values in order to allow the application of algorithms that require discrete data as an input (Gallo *et al.*, 2016). Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses class information. If the discretization process uses class information, then we say it is supervised discretization. Otherwise, it is unsupervised.

Data discretization refers also to data reduction. The raw data are replaced by a smaller number of interval or concept labels. This simplifies the original data and makes the mining more efficient. The resulting patterns mined are typically easier to understand. Moreover, discretized data show more robustness in the presence of noise. However this technique implies always a loss of information.

Closing Remarks

Data integration and data transformation are fundamental in the preprocessing stage of data analysis pipeline to achieve accurate mining results and a deeper understanding of biological phenomena. This contribution has presented the main approaches for data integration and the main data transformation techniques in biological context.

See also: Data Mining in Bioinformatics. Knowledge Discovery in Databases. The Challenge of Privacy in the Cloud

References

- Ashburner, M., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25 (1). 25–29.
- Benson, D.A., Clark, K., Karsch-Mizrachi, I., *et al.*, 2015. GenBank. *Nucleic Acids Research* 43, 30–35.
- Dawson, N.L., *et al.*, 2017. CATH: An expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research* 45 (D1). D289–D295.
- Gallo, C.A., Cecchini, R.L., Carballido, J.A., Micheletto, S., Panzoni, I., 2016. Discretization of gene expression data revised. *Briefings in Bioinformatics* 17 (5). 758–770.
- Gligoric, V., Przulj, N., 2015. Methods for biological data integration: Perspectives and challenges. *Journal of the Royal Society Interface* 12 (112).

- Gomez-Cabrero, D., Abugessaisa, I., Maier, D., *et al.*, 2014. Data integration in the era of omics: Current and future challenges. *BMC Syst Biology* 8 (Suppl. 2). S1.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K., 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research* 45, D353–D361.
- Kolesnikov, N., *et al.*, 2014. ArrayExpress update – Simplifying data submissions. *Nucleic Acid Research* 43, D1113–D1116.
- Han, J., 2012. *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series)*. Morgan Kaufmann.
- Lapatas, V., *et al.*, 2015. Data integration in biological research: An overview. *Journal of Biological Research-Thessaloniki* 22 (1). 9.
- O’Leary, N.A., *et al.*, 2016. Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research* 44, D733–D745.
- Orchard, S., *et al.*, 2013. The MintAct project-IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acid Research* 42, D358–D363.
- Purshitt, H.J., Kalia, V.C., More, R.P., 2018. *Soft Computing for Biological Systems*. Springer.
- Salwinski, L., Miller, C.S., Smith, A.J., *et al.*, 2004. The Database of Interacting Proteins 2004 update. *Nucleic Acid Research* 32, D449–D451.
- Sillitoe, I., Lewis, T.E., Cuff, A.L., *et al.*, 2015. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research* 43 (Database issue). D376–D381.
- Wanichthanarak, K., Fahrman, J.F., Grapov, D., 2015. Genomic, proteomic, and metabolomic data integration strategies. *Biomarker Insights* 10 (Suppl. 4). S1–S6.
- Zhang, Z., Bajic, V.B., Yu, J., Cheung, K.H., Townsend, J.P., 2011. Data integration in bioinformatics: Current efforts and challenges. In: Mahdavi, M.A. (Ed.), *Bioinformatics*. IntechOpen.

Data Reduction

Barbara Calabrese, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Data reduction techniques can be useful to obtain a smaller dataset in volume, avoiding to lose in the reduced representation significant information of the original data. The aim is to generate and process subsequently a dataset more efficiently. Data reduction methods include dimensionality reduction, numerosity reduction, and data compression. In bioinformatics analysis, the application of data reduction techniques is essential in order: (i) to eliminate the irrelevant or noise features; (ii) to reduce storage requirements, computation time and resource usage and, (iii) to decrease the complexity of identification of the main features (Tchitchek, 2018; Prasartvit *et al.*, 2013).

In the following paragraphs the main methodologies for biological data reduction are presented and discussed.

Dimensionality Reduction

Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration (Han, 2012; Meng *et al.*, 2016). Dimensionality reduction methods include wavelet transforms and Principal Components Analysis (PCA), which transform or project the original data onto a smaller space. Other approaches for dimension reduction for microarray data are based on Independent Component Analysis (ICA) (Aziz *et al.*, 2017). In the following subparagraphs, the main methodologies for biological data reduction are illustrated and discussed.

Linear Transformation

Linear transformations are applied to multivariate data to produce new datasets that are more meaningful or can be condensed into fewer variables (Semmlow, 2009). The data transformation used to produce new dataset is often a linear transformations that are easier to compute. A linear transformation can be represented as reported in Eq. 1:

$$y_i(t) = \sum_{j=1}^M w_{ij}x_j(t) \quad i = 1 \dots N \quad (1)$$

Where w_{ij} are constant coefficients that define the transformation. The two most used techniques for data reduction based on linear transformations are Principal Component Analysis (PCA) and Independent Component Analysis (ICA). In PCA, the aim is to transform the data set so as to produce a new set of variables (named principal components) that are uncorrelated. In ICA, the goal is to find new variables (named independent components), that are both statistically independent and non-Gaussian.

Principal Components Analysis

Principal Component Analysis (PCA) is considered a technique for reducing the number of variables without loss of information and for identifying new variables with greater meaning (Semmlow, 2009). PCA reduces data by geometrically projecting them onto lower dimensions, called principal components (Lever *et al.*, 2017). The first principal component is chosen to minimize the total distance between the data and their projection onto the principal component. The second (and subsequent) principal components are selected similarly, with the additional requirement that they are uncorrelated with all previous principal components. This requirement of no correlation means that the maximum number of principal components possible is either the number of samples or the number of features, whichever is smaller. The principal components selection process has the effect of maximizing the correlation between data and their projection. There are different ways of solving PCA. The most efficient algorithm uses Singular Value Decomposition (SVD) (Yao *et al.*, 2012).

In addition to being uncorrelated, the principal components are orthogonal and are ordered in terms of the variability they represent. Thus, the data size can be reduced by eliminating the weaker components, that is, those with low variance. The computation of principal components involves second order statistics that might not be always appropriate for biological data (Jolliffe, 2002).

Independent Component Analysis

Independent Component Analysis (ICA) is a technique that allows the separation of a mixture of signals into their different sources, by assuming non Gaussian signal distribution (Yao *et al.*, 2012). The ICA extracts the sources by exploring the independence underlying the measured data. Thus, it involves higher order statistics to recover statistically independent signals from

the observations of an unknown linear mixture. It turns out to be a technique more powerful than others, such as the Principal Component Analysis (PCA). The typical application of ICA is the “cocktail party problem”. In this situation, multiple people are speaking simultaneously within the same room. Their voices are recorded through multiple microphones, where the number of microphones are greater or equal to the number of speakers. Each microphone records a mixture of the voices of all speakers in the room. The application of ICA allows to recover the original signals produced by different speakers from recorded microphone signals (Hyvarinen and Oja, 2000).

Suppose that X is the input data matrix with dimension $[n \times p]$, where n is the number of samples and p is the number of measured variables or biological entities. The ICA generative model (i.e., a model that describes how the mixed signals are produced) is as follows (Eq. (2)):

$$X = AS \quad (2)$$

where A is the mixing matrix that indicates how the independent components (S) are linearly combined to build X (measured variables or signals). The model assumes that the mixed signals are the product of instantaneous linear combinations of the independent sources. The goal of the ICA is to recover the original signals only from the observations and consequently it is needed to construct an unmixing matrix, W which is the inverse matrix of the mixing matrix (Eq. (3)):

$$S = WX \quad (3)$$

The condition of independence implies the search for moments of superior order. These moments are null for a Gaussian variable. Two classical measures of gaussianity are kurtosis and negentropy. In Lee and Batzoglou (2003), the authors demonstrate that ICA methodologies outperform other leading methods, such as PCA for microarray data reduction.

Wavelet Transforms

In signal processing and analysis, a transform function permits to remapping the signal in order to provide more information than original. Fourier Transform (FT), for example, gives information about the different frequency components in the specific signal. Short-time Fourier Transform (STFT) is an example of time-frequency transform that provides information of the presence of various frequency components evolving time. The limit of STFT is that the analysis window frame is fixed. A more informative transform is represented by wavelet. Wavelet transforms (WT) in comparison to FT and STFT, offer the advantage of time frequency localisation of a signal by using windows of varying sizes and hence are capable of multi resolution analysis of signals. There are two types of wavelet transforms: continuous wavelet transforms (CWT) and discrete wavelet transforms (DWT).

A waveform of finite duration and zero average value is called a wavelet. WT is calculated using a mother wavelet function $\psi(t)$, by convolving the original signal $f(t)$ with the scaled and shifted version of the mother wavelet described by Eq. (4) where a is called the scaling parameter and b is called the translational parameter (Mallat, 2009).

$$Cab = \int_t f(t) \frac{1}{\sqrt{a}} \frac{\psi * (t - b)}{a} dt \quad (4)$$

Since continuous wavelet transforms are calculated at all possible scales and positions, they generate a large amount of data and require larger computation time. In discrete wavelet analysis, scales and positions are chosen based on powers of 2, called the dyadic scales. After discretization, the wavelet function is defined as given in Eq. (5):

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \psi * \frac{(t - nb_0 a_0^m)}{a_0^m} \quad (5)$$

where a_0 and b_0 are constants. The scaling term is represented as a power of a_0 and the translation term is a factor of a_0^m . Values of the parameters a_0 and b_0 are chosen as 2 and 1 respectively and is called as dyadic grid scaling.

The dyadic scaling scheme is implemented using filters developed by Mallat (1989). The original signal is filtered through a pair of one high pass filter $g(n)$ and one low pass filter $h(n)$, and then down sampled to get the decomposed signal through each filter which is half the length of the original signal. This process of filtering results in decomposition of the signal into different frequency components. The low frequency components are called *approximations* and high frequency components are called *details*. The dyadic scaling scheme is implemented using filters developed by Mallat (1989). This constitutes one level of decomposition, mathematically expressed as (see Eqs. (6) and (7)):

$$Y_{hp}(k) = \sum_n X(n)g(2k - n) \quad (6)$$

$$Y_{lp}(k) = \sum_n X(n)h(2k - n) \quad (7)$$

where $X(n)$ is the original signal, $h(n)$ and $g(n)$ are the sample sequences or impulse responses and $Y_{hp}(k)$ and $Y_{lp}(k)$ are the outputs of the high-pass and low-pass filters, respectively, after subsampling by 2. This procedure, known as sub-band coding, can be repeated for further decomposition. At every level, the filtering and subsampling result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence double the frequency resolution) (Saini and Dewan, 2016).

The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients (Liu, 2007). For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0. The work in Cannataro *et al.* (2001) presents a method for data compression based on wavelet transform.

Attribute Selection

Data sets for analysis may contain irrelevant and/or redundant attributes. Some of them can be very noisy and can cause errors in classification process. Moreover, they slope down the overall mining process. Attribute subset selection is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed (Han, 2012). Attribute selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions), guaranteeing that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all features/attributes. The appropriate selection increases the accuracy of the analysis and efficiency of the total process of data mining.

In the process of manual selection of attributes, the user decides which attributes should be eliminated. It is a subjective and time-consuming method.

Heuristic methods that explore a reduced search space are commonly used for attribute subset selection. Their strategy is to make a locally optimal choice in the hope that this will lead to a globally optimal solution. The choice of attributes is performed using tests of statistical significance, which assume that the attributes are independent of one another. Basic heuristic methods of attribute subset selection include stepwise forward selection, stepwise backward elimination, combination of the previous methods and decision tree induction (Han, 2012).

Numerosity Reduction

Numerosity reduction techniques replace the original data by alternative, smaller forms of data representation. These techniques may be parametric or nonparametric (Han, 2012).

For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. Regression and log-linear models are examples of parametric methods. Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling, and data cube aggregation.

Parametric Data Reduction: Regression

Regression and log-linear models can be used to approximate the given data. In (simple) linear regression, the data are modeled to fit a straight line. For example, a random variable, y (called a response variable), can be modeled as a linear function of another random variable, x (called a predictor variable), with the Eq. (8):

$$y = b + wx \quad (8)$$

where the variance of y is assumed to be constant. The coefficients, w and b (called regression coefficients), specify the slope of the line and the y -intercept, respectively. These coefficients can be solved by the method of least squares, which minimizes the error between the actual line separating the data and the estimate of the line. Multiple linear regression is an extension of (simple) linear regression, which allows a response variable, y , to be modeled as a linear function of two or more predictor variables (Eq. (9)):

$$y = b_0 + b_1x_1 + b_2x_2 \quad (9)$$

Log-linear model approximates discrete multidimensional probability distributions. This model estimates the probability of each point (tuple) in a multi-dimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations.

Non Parametric Data Reduction: Histograms, Clustering and Sampling

Histogram is a popular data reduction technique. It divides data into buckets and store average (sum) for each buckets.

Clustering techniques partition the objects into groups, or clusters, so that objects within a cluster are “similar” to one another and “dissimilar” to objects in other clusters. Similarity is commonly defined in terms of how “close” the objects are in a space, based on a distance function. The “quality” of a cluster may be represented by its diameter, the maximum distance between any two objects in the cluster. Centroid distance is an alternative measure of cluster quality and is defined as the average distance of each cluster object from the cluster centroid (denoting the “average object,” or average point in space for the cluster).

In data reduction, the cluster representations of the data are used to replace the actual data. The effectiveness of this technique depends on the data’s nature. It is much more effective for data that can be organized into distinct clusters than for smeared data.

Sampling techniques should create representative samples. A sample is representative, if it has approximately the same property (of interest) as the original set of data. Using a representative sample will work almost as well as using the full datasets (Han, 2012). There are several sampling techniques:

- Simple random sampling: every sample of size n has the same chance of being selected. It uses random numbers. Specifically, if the selected item cannot be selected again, this method is called sampling without replacement; in the sampling with replacement, items can be picked up more than once for the sample – not removed from the full dataset once selected.
- Stratified sampling: data are splitted into several partitions (strata); then draw random samples from each partition. Each strata may correspond to each of the possible classes in the data. The number of items selected from each strata is proportional to the strata size.
- Cluster sampling: is a technique in which clusters of data are identified and included in the sample.

Data Compression

Data compression includes all methods that are applied in order to obtain a reduced or “compressed” representation of the original data. If the original data can be reconstructed from the compressed data without any information loss, the data reduction is called lossless, otherwise the data reduction is called lossy. In (Hosseini, 2016) a review about proteomic and genomic data compression methods is reported.

Proteomic Data Compression

In Nevill-Manning and Witten (1999), the CP (Compress Protein) scheme is proposed, as the first protein compression algorithm, which employs probability to weight all contexts with the maximum of a certain length, based on their similarity to the current context. The XM (eXpert Model) method, presented in Cao *et al.* (2007), encodes each symbol by estimating the probability based on previous symbols. When a symbol is recognized as part of a repeat, the previous occurrences are used to find that symbol's probability distribution; then, it is encoded using arithmetic coding. “Approximate repeats” are utilized in the methods presented in Hategan and Tabus (2004). The ProtComp algorithm exploits approximate repeats and mutation probability for amino-acids to build an optimal substitution probability matrix including the mutation probabilities). Exploiting the “dictionary” is taken into consideration in the methods proposed in Hategan and Tabus (2007) and Daniels *et al.* (2013). In Hategan and Tabus (2007), the ProtCompSecS method is presented, which considers the encoding of protein sequences in connection with their annotated secondary structure information. This method is the combination of the ProtComp algorithm (Hategan and Tabus, 2004) and a dictionary based method, which uses the DSSP (Dictionary of Protein Secondary Structure) database. A heuristic method is introduced in Hayashida *et al.* (2014), that exploits protein domain compositions. In this method, a hyper-graph is built for the proteome, based on evolutionary mechanisms of gene duplication and fusion.

Genomic Data Compression

Genomic data compression methods are grouped in reference-free methods that are based only on the characteristics of the target sequences and reference-based methods, that exploit a set of references sequence (Lee *et al.*, 2007).

The basic idea of reference-free genomic sequence compression is exploiting structural properties, e.g., palindromes, as well as statistical properties of the sequences. The first algorithm which was specifically proposed for genomic sequences compression is biocompress (Grumbach and Tahi, 1993). In this algorithm, that is based on the Ziv and Lempel compression method (Ziv and Lempel, 1977), factors (repeats) and complementary factors (palindromes) in the target sequence are identified and then they are encoded using the length and the position of their earliest occurrences.

In DNA Compact algorithm (Gupta and Agarwal, 2011), the target sequence is first converted into words in a way that A, T, C and G bases are replaced with A, C, <space>A and <space>C, respectively, i.e., the four symbol space is transformed into a three symbol space. In the second phase, the obtained sequence is encoded by WBTC (Word Based Tagged Code).

The GeCo algorithm (Pratas *et al.*, 2016) exploits a combination of context models of several orders for reference-free, as well as for reference-based genomic sequence compression. In this method, extended finite-context models (XFCMs) are introduced and cache-hashes are employed. The cache-hash uses a fixed hash function to simulate a specific structure and takes into consideration only the last hashed entries in memory.

Reference-based genome sequence compression are based on the analysis of the similarity between a target sequence and a (set of) reference sequence(s). For this aim, the target is aligned to the reference and the mismatches between these sequences are encoded. Since the decompressor has access to the reference sequence(s), the reference-based methods obtain high compression rates (Zhu *et al.*, 2013; Wandelt *et al.*, 2014; Deorowicz and Grabowski, 2011). The DNazip algorithm (Christley *et al.*, 2009) was presented for compressing James Watson's genome, considering the high similarity between human genomes, implying that only the variation is required to be saved. In this method, variation data are considered in three parts: (i) SNPs (single nucleotide polymorphisms), which are saved as a position value on a chromosome and a single nucleotide letter of the polymorphism; (ii)

insertions of multiple nucleotides, which are saved as position values and the sequence of nucleotide letters to be inserted; and (iii) deletions of multiple nucleotides, which are saved as position values and the length of the deletion.

In the RLZ algorithm (Kuruppu *et al.*, 2010), each genome is parsed into factors, using the LZ77 approach (Ziv and Lempel, 1977). Then, the compression is done relative to the reference sequence, which is used as a dictionary.

A statistical compression method, GReEn (Genome Re-sequencing Encoding), is proposed in Pinho *et al.* (2012). In this method, the probability distribution of each symbol in the target sequence is estimated using an adaptive model (the copy model), which assumes that the symbols of the target sequence are a copy of the reference sequence, or a static model, relying on the frequencies of the symbols of the target sequence.

The GDC (Genome Differential Compressor) method (Deorowicz and Grabowski, 2011), is based on the LZ77 approach and considers multiple genomes of the same species. In this method, a number of extra reference phrases, that are extracted from other sequences, along with an LZ-parsing for detection of approximate repeats are used. The GDC 2 method (Deorowicz *et al.*, 2015) is an improved version of the GDC that considers short matches after some longer ones. It is worth pointing out that both the GDC and GDC 2 methods support random access to an individual sequence.

The ERGC (Efficient Referential Genome Compressor) method, based on a divide and conquer strategy, is proposed in Saha and Rajasekaran (2015). In this method, first, the reference sequence and the target sequence are divided into some parts with equal sizes; then, an algorithm, which is based on hashing, is used to find one-to-one maps of similar regions between the two sequences. Finally, the identical maps and dissimilar regions of the target sequence, are fed into the PPMD compressor for further compression of the results.

The iDoComp algorithm (Ochoa *et al.*, 2015) consists of three phases. In the first phase, named mapping generation, the suffix arrays are exploited to parse the target sequence into the reference sequence. In the second phase, the consecutive matches, which can be merged together to form an approximate match, are found and used for reducing the size of mapping. Finally, in the third phase, named entropy encoding, an adaptive arithmetic encoder is used for further compression of the mapping and then generation of the compressed file.

The DNA-COMPACT method (Li *et al.*, 2013) can be used for both reference-free and reference-based genome compression. In a first phase, an adaptive mechanism is presented which firstly, considers a sliding window for the subsequences of the reference sequence; then, it searches for the longest exact repeats in the current fragment of the target sequence, in a bi-directional manner. In the second phase, the non-sequential contextual models are introduced and then their predictions are synthesized exploiting the logistic regression model.

The CoGI method (Xie *et al.*, 2015) can be used for reference-free as well as for reference-based genome compression. Considering the latter case, at first, a proper reference genome is selected using two techniques based on base co-occurrence entropy and multi-scale entropy. Next, the sequences are converted to bit-streams, using a static coding scheme, and then the XOR (exclusive or) operation is applied between each target and the reference bit-streams. In the next step, the whole bit-streams are transformed into bit matrices like binary images (or bitmaps). Finally, the images are compressed exploiting a rectangular partition coding method.

Concluding Remarks

There are many data reduction methods, which can be applied to exploratory data analysis of a single data set, or integrated analysis of a pair or multiple data sets. In this contribution the main methods for dimensionality reduction, numerosity reduction and data compression have been discussed, specifically for proteomic and genomic data.

See also: Data Mining in Bioinformatics. Knowledge Discovery in Databases. The Challenge of Privacy in the Cloud

References

- Aziz, R., Verma, C.K., Srivastava, N., 2017. A novel approach for dimension reduction of microarray. *Computational Biology and Chemistry* 71, 161–169.
- Cannataro, M., Curci, W., Giglio, M., 2001. A priority-based transmission protocol for congested networks supporting incremental computations. *ITCC*, pp. 383–388.
- Cao, M., Dix, T., Allison, L., Mears, C., 2007. A simple statistical algorithm for biological sequence compression. In: *Proceedings of the DCC'07: Data Compression Conference*, pp. 43–52. Snowbird, UT, USA.
- Christley, S., Lu, Y., Li, C., Xie, X., 2009. Human genomes as email attachments. *Bioinformatics* 25, 274–275.
- Daniels, N., Gallant, A., Peng, J., *et al.*, 2013. Compressive genomics for protein databases. *Bioinformatics* 29, 283–290.
- Deorowicz, S., Grabowski, S., 2011. Compression of DNA sequence reads in FASTQ format. *Bioinformatics* 27, 860–862.
- Deorowicz, S., Danek, A., Niemiec, M., 2015. GDC 2: Compression of large collections of genomes. *Scientific Reports* 5.
- Grumbach, S., Tahi, F., 1993. Compression of DNA sequences. In: *Proceedings of the DCC'93: Data Compression Conference*, pp. 340–350. Snowbird, UT, USA.
- Gupta, A., Agarwal, S., 2011. A novel approach for compressing DNA sequences using semi-statistical compressor. *Int. J. Comput. Appl.* 33, 245–251.
- Han, J., 2012. *Data Mining. Concepts and Techniques*. Morgan Kaufmann.
- Hategan, A., Tabus, I., 2004. Protein is compressible. In: *Proceedings of the 6th Nordic Signal Processing Symposium*, pp. 192–195, Espoo, Finland.
- Hategan, A., Tabus, I., 2007. Jointly encoding protein sequences and their secondary structure. In: *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS 2007)*, Tuusula, Finland.

- Hayashida, M., Ruan, P., Akutsu, T., 2014. Proteome compression via protein domain compositions. *Methods* 67, 380–385.
- Hosseini, M., Pratas, D., Pinho, A.J., 2016. A survey on data compression methods for biological sequence. *Information* 7, 56.
- Hyvarinen, A., Oja, E., 2000. Independent component analysis: Algorithms and applications. *Neural Networks* 13 (4–5), 411–430.
- Jolliffe, I., 2002. *Principal Component Analysis*. New York: Springer.
- Kuruppu, S., Puglisi, S., Zobel, J., 2010. Relative Lempel-Ziv compression of genomes for large-scale storage and retrieval. In: *International Symposium on String Processing and Information Retrieval*, vol. 6393, pp. 201–206.
- Lee, G., Rodriguez, C., Madabhushi, A., 2007. An empirical comparison of dimensionality reduction methods for classifying gene and protein expression datasets. In: Măndoiu, I., Zelikovsky, A. (Eds.), *Bioinformatics Research and Applications. ISBRA 2007. Lecture Notes in Computer Science*, vol. 4463. Berlin, Heidelberg: Springer.
- Lee, S., Batzoglou, S., 2003. Application of independent component analysis to microarray. *Genome Biol.* 4, R76.
- Lever, J., Krzywinski, M., Altman, N., 2017. Points of significance: Principal component analysis. *Nat. Methods* 14, 641–642.
- Li, P., Wang, S., Kim, J., *et al.*, 2013. DNA-COMPACT: DNA compression based on a pattern-aware contextual modeling technique. *PLOS ONE* 8, e80377.
- Liu, Y., 2007. Dimensionality reduction for mass spectrometry data. In: Alhajj, R., Gao, H., Li, J., Li, X., Zaiane, O.R. (Eds.), *Advanced Data Mining and Applications. ADMA 2007. Lecture Notes in Computer Science*, vol. 4632. Berlin, Heidelberg: Springer.
- Mallat, S., 1989. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 674–693.
- Mallat, S., 2009. *A Wavelet Tour of Signal Processing*, third ed. Elsevier.
- Meng, C., Zeleznik, O.A., Thallinger, G.G., *et al.*, 2016. Dimension reduction techniques for the integrative analysis of multi-omics data. *Brief. Bioinform.* 17 (4), 628–641.
- Nevill-Manning, C., Witten, I., 1999. Protein is incompressible. In: *Proceedings of the DCC'99: Data Compression Conference*, pp. 257–266. Snowbird, UT, USA.
- Ochoa, I., Hernaez, M., Weissman, T., 2015. iDoComp: A compression scheme for assembled genomes. *Bioinformatics* 31, 626–633.
- Pinho, A., Pratas, D., Garcia, S., 2012. GREn: A tool for efficient compression of genome resequencing data. *Nucleic Acids Res.* 40.
- Prasartvit, C.T., *et al.*, 2013. Reducing bioinformatics data dimension with ABC-KNN. *Neurocomputing* 116, 367–381.
- Pratas, D., Pinho, A., Ferreira, P., 2016. Efficient compression of genomic sequences. In: *Proceedings of the DCC'16: Data Compression Conference*, pp. 231–240. Snowbird, UT, USA.
- Saha, S., Rajasekaran, S., 2015. ERGC: An efficient referential genome compression algorithm. *Bioinformatics* 31, 3468–3475.
- Saini, S., Dewan, L., 2016. Application of discrete wavelet transform for analysis of genomic sequences of *Mycobacterium tuberculosis*. *SpringerPlus* 5, 64.
- Semmlow, J.L., 2009. *Biosignal and Medical Image Processing*. CRC Press.
- Tchitchek, T., 2018. Navigating in the vast and deep oceans of high dimensional biological data. *Methods* 132, 1–2.
- Wandelt, S., Bux, M., Leser, U., 2014. Trends in genome compression. *Curr. Bioinform.* 9, 315–326.
- Xie, X., Zhou, S., Guau, J., 2015. CoGI: Towards compressing genomes as an image. *IEEE ACM Transaction on Computational Biology and Bioinformatics* 12 (6), 1275–1285.
- Yao, F., Coquery, J., Le Cao, K., 2012. Independent Principal Component Analysis for biologically meaningful dimension reduction of large biological data sets. *BMC Bioinform.* 13 (24).
- Zhu, Z., Zhang, Y., Ji, Z., He, S., Yang, X., 2013. High-throughput DNA sequence data compression. *Brief. Bioinform.* 16.
- Ziv, J., Lempel, A., 1977. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* 23, 337–343.

Dimensionality Reduction

Italia De Feis, Istituto per le Applicazioni del Calcolo CNR, Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the last 20 years the advances in high performance computing technologies, with more and more powerful and fast computers for data storage and processing, and in the engineering field, with the assembly of more and more sophisticated sensors, determined a huge development of “high resolution” offering new possibilities for scientific knowledge. We are now in the “big-data” era and new methodologies have emerged to perform real time descriptive and predictive analysis on massive amount of data, in order to formulate intelligent informed decisions. The word “big data” refers to a high volume of heterogeneous data formed by continuous or discontinuous information stream. It is a class of data sets so large that it becomes difficult to analyse using the standard methods of data processing. The problems of such data include collection, storage, search, sharing, transfer, visualization and analysis. Big data involves increasing volume, that is, amount of data; velocity, that is speed at which data is in and out; and variety, that is, range of data types and sources, indeed it has been characterized as 3Vs (Laney, 2001). But another V has been added and this 4th V refers to veracity, i.e. reliability of the accumulated data. For these reasons big data has unique features that are not shared by the traditional data sets: massive sample size, high-dimensionality and intrinsic heterogeneity.

This is particularly true for the biomedical/bioinformatics domain, that has experienced a drastic advance since the advent of complete genome sequences. In particular, in Stephens *et al.* (2015) the authors compared genomics with three other major generators of big data: Twitter, You-Tube, and astronomy, in terms of annual storage space required. Twitter requires 117 PB (Petabyte or 1 million GB), whereas You-Tube and astronomy require 1 EB (Exabyte or 1 billion GB) and 12 EB, respectively. Genomics, which is only a part of bioinformatics data, requires 240 EB per year! The authors further estimated that between 100 million and as many as 2 billion human genomes could be sequenced by the year 2025, representing 45 orders of magnitude growth in 10 years and far exceeding the growth in the 3 other big data domains. This explains the fundamental role of dimensionality reduction for the analysis of big data to extract meaningful knowledge. Dimensionality reduction techniques aim at finding and exploiting low-dimensional structures in high-dimensional data to overcome the “curse of dimensionality” and reduce computation and storage burden.

Curse of dimensionality is a term coined by Bellman (1961) and refers to the problem caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space. Indeed a local neighbourhood in higher dimensions is no longer local, i.e. the sparsity increases exponentially given a fixed amount of data points, hence to achieve the same accuracy or resolution and a statistically significant result, much larger data sets are needed.

For example, 100 equispaced sample points in a unit interval have 0.01 distance between them; to maintain the same distance among points in a 10-dimensional unit hypercube would require 10^{20} sample points!

Background/Fundamentals

Dimensionality reduction can be categorized into two main components: feature extraction and feature selection.

Feature extraction projects the original input data in a space of lower dimension. Each new variable is a linear or non-linear combination of the original variables, containing basically the same information as the input data. Three main representatives of this class are Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1933; Jolliffe, 2002), Fisher Discriminant Analysis, also called Linear Discriminant Analysis (LDA) (Fisher, 1936; Mika *et al.*, 1999; Duda *et al.*, 2000) and the ISOMetric feature MAPping (ISOMAP) (Tenenbaum *et al.*, 2000). PCA and LDA are both linear techniques, whereas ISOMAP is a non linear one.

Feature selection only keeps the most relevant variables from the original dataset to construct a model having the “best” predictive power. In the context of regression, representative methods of this class are subset selection techniques, i.e., best subset selection and stepwise selection (Miller, 2002). However, recently sparse learning based methods have received considerable attention due to their good performance and their theoretical robustness; a pioneer technique is the Least Absolute Shrinkage and Selection Operator (LASSO) introduced by Tibshirani in 1996 (Tibshirani, 1996).

Table 1 summarizes some common symbols used throughout this survey. Bold upper-case letters are used for matrices (e.g. $\mathbf{A}$) and bold lower-case letters for vectors (e.g., $\mathbf{a}$). The (i,j) -th entry of $\mathbf{A}$ is represented by a_{ij} and the transpose of $\mathbf{A}$ by $\mathbf{A}'$. For any vector $\mathbf{a} = (a_1, \dots, a_n)'$ its 2-norm is defined as $\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^n a_i^2}$, and its 1-norm is $\|\mathbf{a}\|_1 = \sum_{i=1}^n |a_i|$. $\mathbf{I}_k \in \mathbb{R}^{k \times k}$ is the identity matrix of dimension k and $\mathbf{1}_k \in \mathbb{R}^{k \times 1}$ is a vector of dimension k of all ones.

Methodologies

PCA

PCA uses a linear and orthonormal transform to project p input variables, possibly correlated, onto a set of linearly uncorrelated variables called principal components (PCs). The PCs preserve most of the information given by variances and correlations/

Table 1 Symbols

Notation	Definition or description
n	number of samples
p	number of variables/features
$\mathbf{x} = (x_1, \dots, x_p)' \in R^{p \times 1}$	random vector of dimension p representing variables
$\mathbf{x}_1, \dots, \mathbf{x}_n \in R^{p \times 1}$	n independent samples of $\mathbf{x}$
$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1' \\ \vdots \\ \mathbf{x}_n' \end{pmatrix} \in R^{n \times p}$	data matrix with element x_{ij} being variable j of sample i
$m_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$	sample mean of variable x_j , $j=1, \dots, p$
$\mathbf{m} = (m_1, \dots, m_p) = \frac{1}{n} \mathbf{X}' \mathbf{1}_n \in R^{p \times 1}$	vector of sample means
$\mathbf{S} = \frac{1}{n-1} (\mathbf{X} - \mathbf{1}_n \mathbf{m}') (\mathbf{X} - \mathbf{1}_n \mathbf{m}')' \in R^{p \times p}$	sample covariance matrix based on samples $\mathbf{x}_1, \dots, \mathbf{x}_p$

covariances of the original variables, and their number is less or equal than p . Moreover the PCs are the principal axes of a p -dimensional ellipsoid that fits the data. The first largest principal axis of such ellipsoid will then define the direction of maximum statistical variation. The second principal axis maximizes statistical variation, subject to being orthogonal to the first, and so on. Let's see in detail how PCA works.

Consider a data matrix $\mathbf{X} \in R^{n \times p}$ and assume that variables have been centred, i.e. $m_j = 0$, $j=1, \dots, p$.

In this case the sample covariance $\mathbf{S} = \frac{1}{n-1} \mathbf{X}' \mathbf{X}$.

The projection matrix $\mathbf{A} \in R^{p \times m}$, $m \leq p$, is given by m p -dimensional vectors $\mathbf{a}_k \in R^{p \times 1}$ (the loadings), $k=1, \dots, m$, that map each $\mathbf{x}_i \in R^{p \times 1}$, $i=1, \dots, n$ to a m dimensional vector $\mathbf{z}_i = (z_1^i, \dots, z_m^i)' \in R^{m \times 1}$ (the PCs), $i=1, \dots, n$ with $z_k^i = \mathbf{x}_i' \mathbf{a}_k$, $k=1, \dots, m$, such that the new variables z_k , $K=1, \dots, m$, account for the maximum variability in $\mathbf{X}$, with each $\mathbf{a}_k$, $k=1, \dots, m$ constrained to be a unit vector ($\|\mathbf{a}_k\|_2 = 1$).

Thus the first loading vector $\mathbf{a}_1 \in R^{p \times 1}$ must satisfy

$$\begin{aligned} \mathbf{a}_1 &= \arg \max_{\|\mathbf{a}\|_2 = 1} \left\{ \sum_{i=1}^n (z_1^i)^2 \right\} = \arg \max_{\|\mathbf{a}\|_2 = 1} \left\{ \sum_{i=1}^n (\mathbf{x}_i' \mathbf{a})^2 \right\} \\ &= \arg \max_{\|\mathbf{a}\|_2 = 1} \{ \|\mathbf{Xa}\|^2 \} = \arg \max \left\{ \frac{\mathbf{a}' \mathbf{X}' \mathbf{Xa}}{\mathbf{a}' \mathbf{a}} \right\} \end{aligned}$$

It can be shown that, since $\mathbf{X}' \mathbf{X}$ is symmetric, the maximum is reached for its maximum eigenvalue $\lambda_{\max}$ when $\mathbf{a}_1$ is $\mathbf{v}_{\max}$ the corresponding eigenvector.

Once the first component has been determined, the k -th component, $k \geq 2$, can be found by considering the residual of the projection of data on $\mathbf{a}_1, \dots, \mathbf{a}_{k-1}$, i.e.

$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \sum_{s=1}^{k-1} (\mathbf{x}_i' \mathbf{a}_s) \mathbf{a}_s, \quad i=1, \dots, n \Leftrightarrow \hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} (\mathbf{Xa}_s) \mathbf{a}_s'$$

and then finding the loading vector $\mathbf{a}_k$ that maximizes the variance of the new data

$$\arg \max_{\|\mathbf{a}\|_2 = 1} \{ \|\hat{\mathbf{X}}_k \mathbf{a}\|^2 \} = \arg \max \left\{ \frac{\mathbf{a}' \hat{\mathbf{X}}_k' \hat{\mathbf{X}}_k \mathbf{a}}{\mathbf{a}' \mathbf{a}} \right\}$$

Playing a bit with linear algebra, it is possible to show that $\mathbf{a}_k$ is the k -th eigenvector of $\mathbf{X}' \mathbf{X}$ corresponding to the k -th largest eigenvalue.

Thus the full principal components decomposition of $\mathbf{X}$ is given by

$$\mathbf{Z} = \mathbf{XA}, \mathbf{Z} \in R^{n \times p}$$

where $\mathbf{A}$ is a matrix of dimension $p \times p$ whose columns are the eigenvectors of $\mathbf{X}' \mathbf{X}$. From this it follows that

$$\text{cov}(\mathbf{Z}) = \mathbf{A}' \text{cov}(\mathbf{X}) \mathbf{A} = \mathbf{A}' \mathbf{S} \mathbf{A} = \frac{\mathbf{\Lambda}}{n-1}, \quad \text{where} \quad \mathbf{X}' \mathbf{X} = \mathbf{A} \mathbf{\Lambda} \mathbf{A}'$$

is the eigenvalue-eigenvector decomposition of $\mathbf{X}' \mathbf{X}$ and $\mathbf{\Lambda}$ is its eigenvalue matrix. This means that the PCs are decorrelated, but if data follow a normal distribution, the PCs are independent. Moreover it follows that

$$\text{Tr}(\mathbf{S}) = \frac{1}{n-1} \sum_{i=1}^p \lambda_i = \text{Tr}(\text{cov}(\mathbf{Z}))$$

i.e., the total variation in the data $\mathbf{X}$ equals the total variation in the PCs and the proportion of variation explained by each PC is $\lambda_s / (\sum_{i=1}^p \lambda_i)$, $s=1, \dots, p$. Obviously to reduce the original dimension p to m , with $m \leq p$, we can consider only the first m eigenvectors to project the original data, i.e.

$$\mathbf{Z}_m = \mathbf{XA}_m$$

and, by construction, this score matrix $\mathbf{A}_m$ maximizes the variance in the original data that has been preserved, while minimising the residual of the projection. In this case the total variance explained by the first m components will be $(\sum_{i=1}^m \lambda_i) / (\sum_{i=1}^p \lambda_i)$.

The PCA can also be built using the Singular Value Decomposition (SVD) (Golub and Van Loan, 1996) of input matrix $\mathbf{X}$, that decomposes $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}(\boldsymbol{\Sigma})\mathbf{V}', \mathbf{U} \in \mathbb{R}^{n \times r}, \mathbf{V} \in \mathbb{R}^{p \times r}, (\boldsymbol{\Sigma}) = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$$

being r the rank of $\mathbf{X}$. From this it follows the spectral decomposition of $\mathbf{X}$

$$\mathbf{X}'\mathbf{X} = \mathbf{V}(\boldsymbol{\Sigma})^2\mathbf{V}'$$

The PCs are therefore given by

$$\mathbf{Z} = \mathbf{X}\mathbf{V} = \mathbf{U}(\boldsymbol{\Sigma})\mathbf{V}'\mathbf{V} = \mathbf{U}(\boldsymbol{\Sigma})$$

and the variance of the scores for the k -th PC is $\sigma_k^2 / (n - 1)$.

A truncated $n \times m$ score matrix, $m \leq r$, $\mathbf{Z}_m$ can be obtained by considering only the first m largest singular values and their singular vectors

$$\mathbf{Z}_m = \mathbf{U}_m(\boldsymbol{\Sigma})_m$$

It can be shown that $\mathbf{Z}_m$ gives the best possible rank m approximation to $\mathbf{Z}$, in the sense of the difference between the two having the smallest possible Frobenius norm (Eckart and Young, 1936; Householder and Young, 1938; Gabriel, 1978).

An important question is the choice of m , i.e. the number of PCs. Different techniques exist, such as the Cumulative Percentage of Total Variation, the Kaiser's rule, the Scree plot, tests of hypothesis, cross-validation or bootstrapping. The first three are ad hoc rules-of-thumb, whose justification is mainly based on intuition and on the fact that they work in practise. Tests of hypothesis require distributional assumptions that often data don't satisfy. Computationally intensive methods, such as cross validation or bootstrapping are too expensive for high dimensional data. A detailed description of the possible proposals and all the mathematical and statistical properties of PCA can be found in Jolliffe (2002).

The derivation and properties of PCs are based on the eigenvalue-eigenvector decomposition of the covariance matrix. In practise, it is more common to standardize the variables $\mathbf{x}$ and define PCs as the eigenvectors of the correlation matrix. A major argument for using correlation is the sensitivity of the PCs derived from covariance matrix to the units of measurements of each element of $\mathbf{x}$. If the variances of elements of $\mathbf{x}$ differ too much each other, then variables with the largest variance will tend to dominate the first few PCs. On the contrary for correlation matrices the standardized variates are a-dimensional and can be happily combined to give PCs.

Fisher Linear Discriminant

Fisher Linear Discriminant/LDA, uses a linear transform to reduce the dimension seeking directions that are efficient for discrimination when it is known how data divide into K classes. The idea is to build a projection that preserves as much of the class discriminatory information as possible. Fig. 1 shows the conceptual difference between PCA and LDA.

Assume that $\mathbf{x}_1, \dots, \mathbf{x}_n$ are distributed in K classes, n_1 in C_1 , n_2 in $C_2, \dots, n_K$ in C_K , with $n_1 + \dots + n_K = n$. The idea is to obtain a transformation of $\mathbf{X}$ projecting the samples onto a hyperplane with dimension $K - 1$.

Consider first the case with two classes, i.e. $K=2$. In this case the hyperplane is a line and this means to obtain the following n scalars

$$y_i = \mathbf{w}'\mathbf{x}_i, \quad \mathbf{w} \in \mathbb{R}^{p \times 1}, \quad i = 1, \dots, n$$

where the optimal projection $\mathbf{w}$ is the one that maximizes the separability of the scalars.

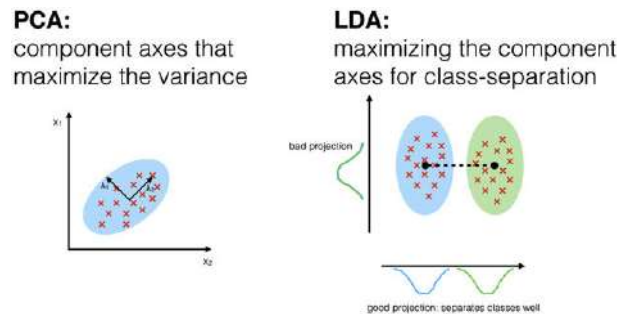


Fig. 1 PCA and LDA. Reproduced from Sebastianraschka. Available at: https://sebastianraschka.com/Articles/2014_python_lda.html.

A measure of the separation between the projected points is the difference of the sample means. The mean vector of each class in $\mathbf{x}$ and $\mathbf{y} = (y_1, \dots, y_n)'$ feature space is:

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}, \quad \text{and} \quad \tilde{\mathbf{m}}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \mathbf{w}' \mathbf{x} = \mathbf{w}' \mathbf{m}_i \quad i = 1, 2$$

It follows that the distance between the projected means is

$$|\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}_2| = |\mathbf{w}'(\mathbf{m}_1 - \mathbf{m}_2)|$$

and this difference can be made as large as wanted merely by scaling $\mathbf{w}$. Of course, to obtain a good separation of y_i , $i = 1, \dots, n$, it is important to take into account the variability of data within the classes. The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class variability, or the so-called *scatter*. For each class the scatter, an equivalent of the variance, is defined as

$$\tilde{s}_i^2 = \sum_{y \in C_i} (y - \tilde{\mathbf{m}}_i)^2, \quad i = 1, 2$$

$\tilde{s}_i$ measures the variability within class C_i after projecting it on the $\mathbf{y}$ -space. Thus $\tilde{s}_1 + \tilde{s}_2$ measures the variability within the two classes at hand after projection, hence it is called *within-class scatter* of the projected samples. The Fisher linear discriminant is defined as the linear function $\mathbf{w}'\mathbf{x}$ that maximizes the criterion function

$$J(\mathbf{w}) = \frac{|\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

Therefore, the maximization looks for a projection where samples from the same class are projected very close to each other and, at the same time, the projected means are as far apart as possible. To obtain $J(\cdot)$ as an explicit function of $\mathbf{w}$, it is useful to define the scatter matrices $\mathbf{S}_i$, $i = 1, 2$, and $\mathbf{S}_W$ by

$$\mathbf{S}_i = \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)' \quad \text{and} \quad \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

It easily follows

$$\tilde{s}_i^2 = \sum_{\mathbf{x} \in C_i} (\mathbf{w}'\mathbf{x} - \mathbf{w}'\mathbf{m}_i)^2 = \mathbf{w}'\mathbf{S}_i\mathbf{w} \quad i = 1, 2 \quad \text{and} \quad \tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}'\mathbf{S}_W\mathbf{w}$$

Similarly

$$(\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}_2)^2 = (\mathbf{w}'\mathbf{m}_1 - \mathbf{w}'\mathbf{m}_2)^2 = \mathbf{w}'\mathbf{S}_B\mathbf{w} \quad \text{with} \quad \mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)'$$

$\mathbf{S}_W$ is called the *within-class scatter matrix* and is proportional to the sample covariance matrix for the pooled p -dimensional data. $\mathbf{S}_B$ is called the *between-class scatter matrix* and its rank is at most one being the outer product of two vectors. Moreover for any $\mathbf{w}$, $\mathbf{S}_B\mathbf{w}$ is in the direction of $\mathbf{m}_1 - \mathbf{m}_2$. In terms of $\mathbf{S}_W$ and $\mathbf{S}_B$, the criterion function $J(\cdot)$ can be written as

$$J(\mathbf{w}) = \frac{\mathbf{w}'\mathbf{S}_B\mathbf{w}}{\mathbf{w}'\mathbf{S}_W\mathbf{w}}$$

and the vector $\mathbf{w}$ that maximizes it, satisfies

$$\mathbf{S}_B\mathbf{w} = \lambda \mathbf{S}_W\mathbf{w} \tag{1}$$

which is a generalized eigenvalue problem. If $\mathbf{S}_W$ is not singular then Eq. (1) can be rewritten as a common eigenvalue problem

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda \mathbf{w}$$

whose solution can be immediately written as

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

being $\mathbf{S}_B\mathbf{w}$ in the direction of $\mathbf{m}_1 - \mathbf{m}_2$, and the classification has been converted from a p -dimensional problem to a more manageable one-dimensional one.

Suppose now to have K classes. In this case there will be $K - 1$ discriminant functions and the projection is from a p -dimensional space to a $(K - 1)$ -dimensional space ($p \geq K$), i.e.

$$\mathbf{y}_i = \mathbf{W}'\mathbf{x}_i, \quad \mathbf{y}_i \in R^{K-1 \times 1}, \quad \mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_{K-1}] \in R^{p \times K-1}; \quad i = 1, \dots, n$$

The generalization for the within-class scatter matrix is

$$\mathbf{S}_W = \sum_{i=1}^K \mathbf{S}_i, \quad \text{where} \quad \mathbf{S}_i = \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)' \quad \text{and} \quad \mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

The between-class scatter matrix is defined as

$$\mathbf{S}_B = \sum_{i=1}^K n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})', \quad \text{with} \quad \mathbf{m} = \frac{1}{n} \sum_{\mathbf{x}} \mathbf{x} = \frac{1}{n} \sum_{i=1}^K n_i \mathbf{m}_i \quad \text{total mean}$$

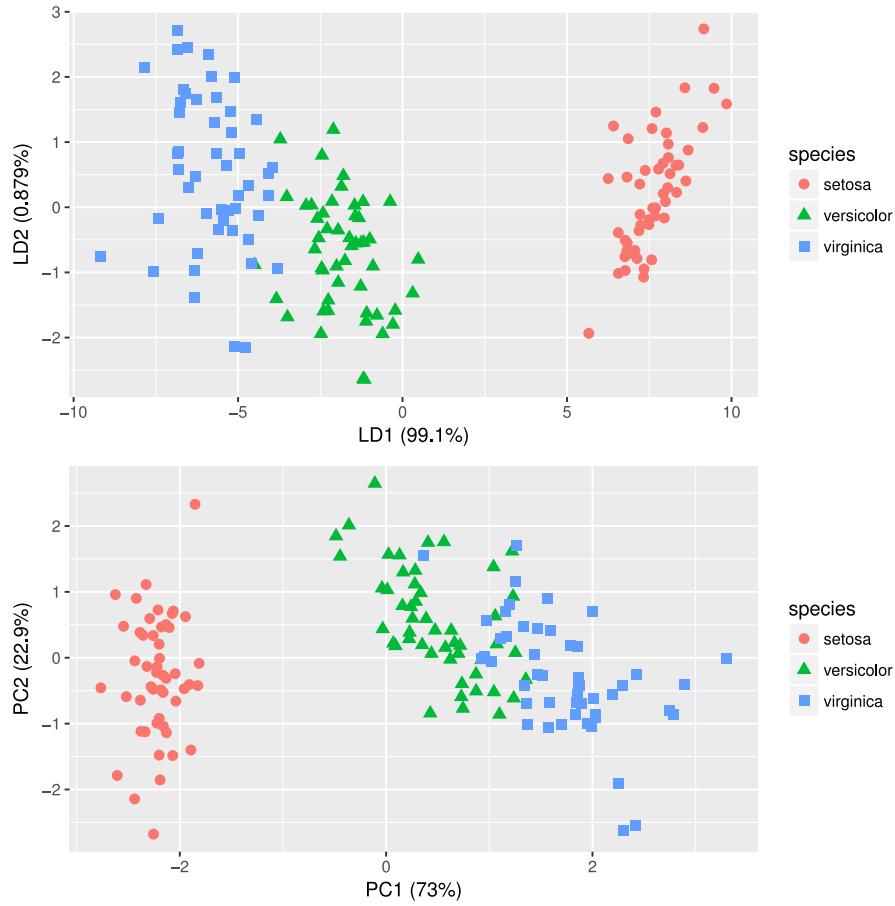


Fig. 2 LDA and PCA on the Iris data.

Similarly, the mean vectors and the scatter matrices for the projected samples $\mathbf{y}_i$ are defined as

$$\tilde{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^K n_i \tilde{\mathbf{m}}_i, \quad \tilde{\mathbf{m}}_i = \frac{1}{n_i} \sum_{\mathbf{y} \in C_i} \mathbf{y}$$

$$\tilde{\mathbf{S}}_W = \sum_{i=1}^K \sum_{\mathbf{y} \in C_i} (\mathbf{y} - \tilde{\mathbf{m}}_i)(\mathbf{y} - \tilde{\mathbf{m}}_i)', \quad \tilde{\mathbf{S}}_B = \sum_{i=1}^K n_i (\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})'$$

and it is easy to show that

$$\tilde{\mathbf{S}}_W = \mathbf{W}' \mathbf{S}_W \mathbf{W} \quad \text{and} \quad \tilde{\mathbf{S}}_B = \mathbf{W}' \mathbf{S}_B \mathbf{W}$$

The optimal transformation matrix $\mathbf{W}$ is the one that maximizes the ratio of the between-class scatter to the within-class scatter. This measure gives origin to the following criterion

$$J(\mathbf{W}) = \frac{|\tilde{\mathbf{S}}_B|}{|\tilde{\mathbf{S}}_W|} = \frac{|\mathbf{W}' \mathbf{S}_W \mathbf{W}|}{|\mathbf{W}' \mathbf{S}_B \mathbf{W}|}$$

It can be shown that the columns of $\mathbf{W}$ are the generalized eigenvectors that correspond to the largest eigenvalues in

$$\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i$$

Because $\mathbf{S}_B$ is the sum of K matrices of rank at most 1, and only $K - 1$ are independent, $\mathbf{S}_B$ is of rank at most $K - 1$. Thus, no more than $K - 1$ of the eigenvalues are non-zero, and the desired weight vectors correspond to these non-zero eigenvalues. Once the data have been projected, a full classifier can be created using the Bayesian decision theory.

Fig. 2 shows the IRIS data (Fisher, 1936) and their projection by LDA and PCA:

ISOMAP

ISOMAP is a dimensionality reduction method that combines the major algorithmic features of PCA and Multi Dimensional Scaling (MDS), i.e. computational efficiency, global optimality, and asymptotic convergence guaranteed, with the flexibility to

learn a broad class of non linear manifolds. In mathematics a manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighbourhood that is topologically the same as the open unit ball in R^n). An example of manifold is the Earth. We know it is round but on the small scales that we see, the Earth looks flat. In general, any object that is nearly “flat” on small scales is a manifold.

ISOMAP exploits geodesic paths for nonlinear dimensionality reduction, indeed it finds the map that preserves the global, nonlinear geometry of the data by maintaining the geodesic manifold inter-point distances. Recall that in the original sense a geodesic was the shortest route between two points on the Earth’s surface, then the term has been generalized to include measurements in much more general mathematical spaces.

The crucial point is the estimation of the geodesic distance between distant points, given only input-space distances (euclidean). For neighbouring points, input space distance provides a good approximation to geodesic distance. For distant points, geodesic distance can be approximated by adding up a sequence of short hops between neighbouring points. These approximations are computed efficiently by finding shortest paths in a graph with edges connecting neighbouring data points.

ISOMAP algorithm has the following three steps:

- identification of which points are neighbours on the manifold M , based on the euclidean distances $d_X(\mathbf{x}_i, \mathbf{x}_j)$ between pairs of points $\mathbf{x}_i, \mathbf{x}_j$ in the input space $X = R^{p \times 1}$. Two simple methods are to connect each point to all points within some fixed radius ε , or to consider all of its K nearest neighbours. These neighbourhood relations are represented as a weighted graph G over the data points, with edges of weight $d_X(\mathbf{x}_i, \mathbf{x}_j)$ between neighbouring points;
- estimation of the geodesic distances $d_M(\mathbf{x}_i, \mathbf{x}_j)$ between all pairs of points on the manifold M by the computation of their shortest path distances $d_G(\mathbf{x}_i, \mathbf{x}_j)$ in the graph G ;
- application of the classical MDS to the matrix of graph distances $D_G = \{d_G(\mathbf{x}_i, \mathbf{x}_j)\}$, constructing an embedding of the data in a d -dimensional Euclidean space $Y = R^{d \times 1}$ that best preserves the manifolds estimated intrinsic geometry. The coordinate vectors $\mathbf{y}_i$ for points in Y are chosen to minimize the cost function

$$E = \|\tau(D_G) - \tau(D_Y)\|_F \quad (2)$$

where D_Y denotes the matrix of euclidean distances $\{d_Y(\mathbf{y}_i, \mathbf{y}_j) = \|\mathbf{y}_i - \mathbf{y}_j\|_2\}$ and $\|A\|_F = \sqrt{\sum_i \sum_j a_{ij}^2}$ is the Frobenius norm. The operator τ converts distances to inner products which uniquely characterize the geometry of the data in a form that supports efficient optimization. τ is defined as

$$\tau(D) = -\frac{\mathbf{H}\mathbf{S}\mathbf{H}}{2}$$

with

$$\mathbf{S} = (s_{ij}) = (D_{ij}^2) \quad \text{and} \quad \mathbf{H} = (h_{ij}) = \left(\delta_{ij} - \frac{1}{N} \right), \quad \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$\mathbf{S}$ is the matrix of squared distances and $\mathbf{H}$ is the centering matrix. It can be shown that the global minimum of Eq. (2) is achieved by setting the coordinates $\mathbf{y}_i$ to the top d eigenvectors of the matrix $\tau(D_G)$. Note that, as for PCA, the true dimensionality of the data can be estimated from the decrease in error as the dimensionality of Y is increased.

ISOMAP recovers asymptotically the true dimensionality and geometric structure of a strictly larger class of nonlinear manifolds. Indeed it can be shown that as the number of data points increases, the graph distances $d_G(\mathbf{x}_i, \mathbf{x}_j)$ provide increasingly better approximations to the intrinsic geodesic distances $d_M(\mathbf{x}_i, \mathbf{x}_j)$, becoming arbitrarily accurate in the limit of infinite data (Tenenbaum *et al.*, 2000).

Fig. 3 shows the Swiss Roll data and their projection by ISOMAP:

Subset Selection

Aim of the regression is to model and quantify the relationship between groups of variables, i.e.

$$\mathbf{y} = f(\mathbf{x}), \quad \mathbf{y} = (y_1, \dots, y_q)' \in R^{q \times 1}, \quad \mathbf{x} = (x_1, \dots, x_p)' \in R^{p \times 1}$$

where the independent variables $\mathbf{x}$ are called predictors, regressors or explanatory variables and the dependent ones $\mathbf{y}$ are called responses. Regression consists of two steps: estimation of the existing connection and prediction of new values.

The most basic type of regression and commonly used predictive analysis is the multiple linear regression (Jobson, 1991). Multiple linear regression models the relation between a scalar response variable y , i.e. $q=1$, and one or more explanatory variables $\mathbf{x}$ by linear predictor functions whose unknown parameters are estimated from the data. Given a dataset $(y_i, \mathbf{x}_i)_{i=1, \dots, n}$ of n samples of $(y, \mathbf{x})$ a linear regression model assumes the following relationship

$$y_i = \beta_0 + \mathbf{x}_i' \boldsymbol{\beta} + \varepsilon_i \quad i = 1, \dots, n, \quad \boldsymbol{\beta} \in R^{p \times 1}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)'$ is the noise or the error term that captures all other factors which influence the dependent variable y_i other than the predictors $\mathbf{x}_i$. It is assumed independent from $\mathbf{x}$, with zero mean and covariance $\sigma^2 I_p$.

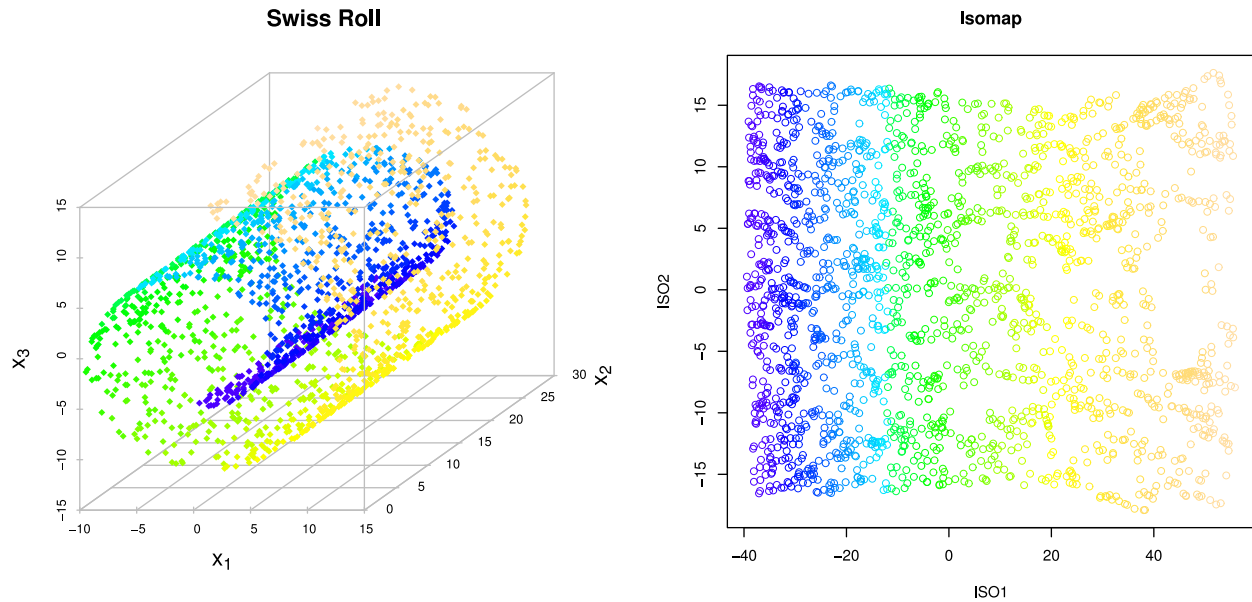


Fig. 3 3D Nonlinear Swiss Roll data and 2D Linear Manifold.

If $n \geq p$ and $\text{rank}(X) = p$, β can be efficiently estimated by the least squares method and used for future prediction. In the case these conditions are not verified, it is necessary to consider alternatives by replacing ordinary least squares fitting with some other fitting procedures that improve the prediction accuracy and the model interpretability. Indeed by “treating” the irrelevant features in some way, a model that is more easily interpreted can be obtained.

However, the selection of the “most informative” regressors from a larger set of observable variables to predict a particular quantity of interest y is an important problem independent from the conditions on n and p . First of all, it may be necessary because it is expensive to measure the variable y and it is hoped to be able to predict it with sufficient accuracy from other variables which can be measured cheaply. Unless the true form of the relationship between the x and y variables is known, it will be necessary for the data used to select the variables and to calibrate the relationship to be representative of the conditions in which the relationship will be used for prediction. Secondly, the variance of the predicted values increases monotonically with the number of variables used in the prediction. Indeed as more variables are added to a model we are “trading off” reduced bias against an increased variance. If a variable has no predictive value, then its inclusion just increases the variance. If the addition of a variable doesn’t reduce the biases significantly, then the increase in prediction variance may exceed the benefit from bias reduction.

Two classes of methods can be taken into account: subset selection and shrinkage. The subset selection identifies a subset of the p predictors that it is believed to be related to the response, then least squares on the reduced set of variables is performed. The shrinkage fits a model involving all p predictors, but the estimated coefficients are shrunk towards zero relative to the least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance and can also perform variable selection.

The subset selection divides into two procedures: best subset and stepwise model selection. The best subset selection consists of the following three steps:

- Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.
- For $k = 1, \dots, p$:
 - a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
 - b) Pick the best among these $\binom{p}{k}$ models, and call it M_k . Here best is defined as having the smallest Residual Sum of Squares (RSS), or equivalently largest R^2 .
- Select a single best model from among $M_0, \dots, M_p$ using cross-validated prediction error, Mallows C_p , Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), or adjusted R^2 .

Obviously, for computational reasons, best subset selection is infeasible with very large p . Moreover, the larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data. Thus an enormous search space can lead to overfitting and high variance of the coefficient estimates. For both of these reasons, stepwise methods, which explore a restricted set of models, are attractive alternatives. Stepwise methods are of two types: forward and backward.

Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model. In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model. In particular

- Let M_0 denote the null model, which contains no predictors.
- For $k=0, \dots, p$:
 - a) Consider all $p-k$ models that augment the predictors in M_k with one additional predictor.
 - b) Choose the best among these $p-k$ models, and call it M_{k+1} . Here best is defined as having smallest RSS or highest R^2 .
- Select a single best model from among $M_0, \dots, M_p$ using cross-validated prediction error, Mallows C_p , AIC, BIC, or adjusted R^2 .

It is important to stress that it is not guaranteed that the procedure finds the best possible model out of all 2^p models containing subsets of the p predictors.

The term stepwise regression is often used to mean an algorithm proposed by [Efroymson \(1960\)](#). This is a variation on forward selection. After each variable (other than the first) is added to the set of selected variables, a test is made to see if any of the previously selected variables can be deleted without appreciably increasing the residual sum of squares. As with forward selection, there is no guarantee that the Efroymson algorithm will find the best-fitting subsets, though it often performs better than forward selection when some of the predictors are highly correlated. The algorithm incorporates its own built-in stopping rule.

Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection. However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

- Let M_p denote the null model, which contains all p predictors.
- For $k=p, \dots, 1$:
 - a) Consider all k models that contain all but one of the predictors in M_k , for a total of $k-1$ predictors.
 - b) Choose the best among these k models, and call it M_{k-1} . Here best is defined as having smallest RSS or highest R^2 .
- Select a single best model from among $M_0, \dots, M_p$ using cross-validated prediction error, Mallows C_p , AIC, BIC, or adjusted R^2 .

Like forward stepwise selection, the backward selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection. Also backward stepwise selection is not guaranteed to yield the best model containing a subset of the p predictors. Backward selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

As an alternative to subset selection in the case of $p > n$, there exist the shrinkage methods that fit a model containing all p predictors using a technique that constrains the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero, and in some cases performs variable selection. This shrinking of the coefficient estimates can significantly reduce their variance, also if it increases the bias. However the total error estimate is lower than the one from the least square fitting.

A very recent alternative is the Lasso. The lasso coefficients, $\hat{\beta}_{\lambda}^L$, minimize the quantity

$$\min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Lasso shrinks the coefficient estimates towards zero, as with ridge regression, but the use of an ℓ_1 penalty instead of a ℓ_2 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter is sufficiently large. Hence, much like subset selection, the lasso performs variable selection and yields sparse model, that is, models that involve only a subset of the variables.

Future Directions

Machine learning and statistics for high dimensional data have been the most utilized tools for data analysis. Large scale data existed well before the big data era, particularly in bioinformatics, and different techniques have been successfully used to analyse both small scale as well as large scale. However, big data poses more challenges on the traditional learning methods in terms of velocity, variety, and incremental data. Traditional learning methods usually embed iterative processing and complex data dependency among operations consequently, they cannot be used to perform fast processing on massive data. Accomplishing the goal of data analysis will clearly require integration of different expertises, large-scale machine learning systems, and a computing infrastructure that can support flexible and dynamic queries to search for patterns over very large collections in very high dimensions.

Closing Remarks

The enormous quantity of data produced in every fields, social, economic, medical, cyber security, just to cite some, offered new challenges to scientists in order to produce significant knowledge. Machine learners, mathematicians and statisticians put a lot of effort in investigating new methodologies or extending the old ones to deal with the “curse of dimensionality” in the “big data era”. Indeed data analysis today is not an unsophisticated activity carried out by hand; it is much more ambitious and is now producing quite sophisticated objects. This is why dimensionality reduction is the hottest topic in data science today, aiming at building simpler and more comprehensible models, improving data mining performance, and helping prepare, clean, and understand data.

PCA, LDA and feature selection in linear regression models represents the basis knowledge each scientist should have.

See also: Data Mining in Bioinformatics. Knowledge Discovery in Databases. Next Generation Sequencing Data Analysis. The Challenge of Privacy in the Cloud

References

- Bellman, R., 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press.
- Duda, R.O., Hart, P.E., Stork, D.H., 2000. *Pattern Classification*, second ed. Wiley Interscience.
- Eckart, C., Young, G., 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1 (3), 211–218.
- Efroymson, M.A., 1960. Multiple regression analysis. In: Ralston, A., Wilf, H.S. (Eds.), *Mathematical Methods for Digital Computers*. New York, NY: John Wiley.
- Fisher, R.A., 1936. The use of multiple measurements in taxo-nomic problems. *Annals of Eugenics* 7, 179–188.
- Gabriel, K.R., 1978. Least squares approximation of matrices by additive and multiplicative models. *J. R. Statist. Soc. B* 40, 186196.
- Golub, G.H., Van Loan, C.F., 1996. *Matrix Computations*, third ed. Johns Hopkins.
- Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24, 417–441, 498–520.
- Householder, A.S., Young, G., 1938. Matrix approximation and latent roots. *Amer. Math. Mon.* 45, 165171.
- Jobson, J.D., 1991. *Applied Multivariate Data Analysis: Regression and Experimental Design*. vol. I. New York, NY: Springer-Verlag.
- Jolliffe, I., 2002. *Principal Component Analysis*. New York, NY: Springer-Verlag.
- Laney, D., 2001. 3D data management: Controlling data volume, velocity, and variety. Technical Report, META Group.
- Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R., 1999. Fisher discriminant analysis with kernels. In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop*, Madison, WI, pp. 41–48.
- Miller, A., 2002. *Subset Selection in Regression*. Chapman and Hall/CRC. p. 2002.
- Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 6 (2), 559–572.
- Stephens, Z.D., Lee, S.Y., Faghri, F., *et al.*, 2015. Big data: Astronomical or genetical? *PLoS Biol* 13 (7), e1002195.
- Tenenbaum, J.B., De Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290 (5500), 23192323.
- Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1), 267–288.

Kernel Machines: Introduction

Italo Zoppis and Giancarlo Mauri, University of Milan-Biocca, Milan, Italy
Riccardo Dondi, University of Bergamo, Bergamo, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Kernel methods are a class of machine learning algorithms implemented for many different inferential tasks and application areas (Smola and Schölkopf, 1998; Shawe-Taylor and Cristianini, 2004; Schölkopf and Burges, 1999). The main idea of these techniques is to map the observed examples (i.e., *training set*) into a new representation space (known as *feature space*), where object properties can be extracted more easily, and linear algorithms are exploited to find relationships that are difficult to identify within the original *pattern space*.

Importantly, it is not necessary to know the map which provides the new input representation, rather by reformulating the algorithm in a way that only requires the knowledge of inner products between the input data, one can evaluate a particular *kernel function* without explicitly expressing the embedding.

In this context, a kernel is a mathematical function, satisfying specific properties, that can be interpreted as a similarity measure between data items. Therefore, when a learning algorithm is adapted to access input only through inner products then, in that procedure, inner products can be replaced by a kernel function. This is the well known *kernel trick* which provides an implicit input transformation into the new space of representation.

Moreover, the interpretation of kernels as similarity measures has important consequences. First, as we will see, it motivates the design of specific kernels for particular inferential problems and data types. Then, it gives us the possibility to apply this approach in all distance-based (or dissimilarity-based) methods, as for example in Pekalska and Duin (2005), Ramakrishnan et al. (2010), Cava et al. (2014). In this article we will describe the kernel approach, following this interpretation.

The Kernel Function

Machine learning algorithms aim to identify functional dependencies within the observed data (i.e., experience or training set) to make useful inferences, using the dependencies that have been learned. In other words, machine learning allows one to build predictive models from observed data, and to apply these models to infer the properties of new items.

The inference models, trained using the kernel approach, are based on an interesting and powerful property. They allow to obtain new data representations for the training patterns, by embedding such observations, into a new *feature space* (Fig. 1). Then, in the new feature space, using techniques from optimization, mathematical analysis and statistics, one can extrapolate interesting properties, which will be useful for providing the required inference for new data.

Inference often takes the form of classification, which is probably one of the oldest and most studied machine learning activities. Classification aims to identify to which of a set of categories a new observation belongs, on the basis of a set of examples containing observations whose category labels (i.e., group membership) are known. A typical classification within healthcare is, for example, the determination of whether new subjects can be discriminated from patients (or cases) or classified as healthy (controls), according to a set of previously sampled attribute values, see for example, Chinello et al. (2014), Cava et al. (2013a,b). The discrimination is provided by a classification algorithm, which takes a set of labeled examples as input (i.e., patient records

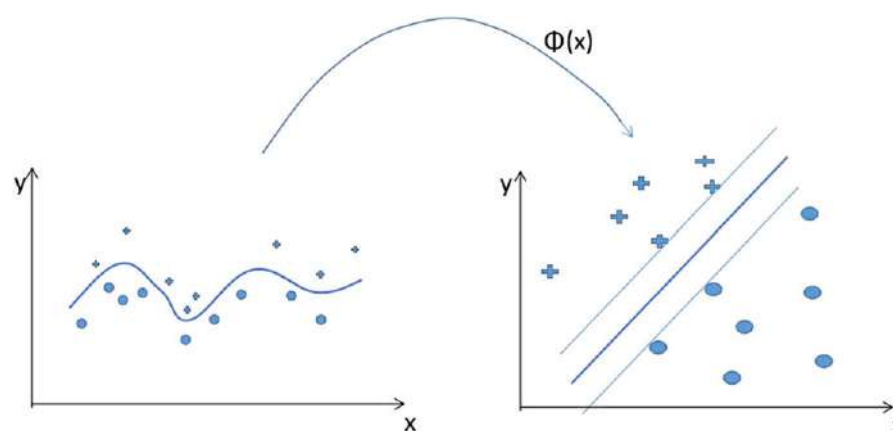


Fig. 1 Kernels use nonlinear maps to embed input patterns in a feature space where linear predictive models are applied.

with their associated labeled groups), and returns a *trained* model, generally defined by a set of fitted parameters. Suppose, for example, that we are given the following training set.

$$\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \in \mathcal{X} \times \mathcal{Y}$$

We refer to the objects denoted by x_i as patterns, while the $y_i \in \mathcal{Y}$ – the response variables, are generally called labels. The training set $\mathcal{T}$ of all pairs (x_i, y_i) represents past observations, whereas the collection of new examples, for which we would like to guess new labels, $\hat{y}_i \in \mathcal{Y}$, is generally called the *test set*. In other words, our aim is to generalize the correspondences (x_i, y_i) , using a function h which can be defined on all possible objects $\mathcal{X}$, and generate labels from the set of all possible labels $\mathcal{Y}$. The label predicted for a new test pattern, x , would then be $\hat{y} = h(\hat{x})$.

Notice that, no particular assumption is made about the nature of $\mathcal{X}$ (and $\mathcal{Y}$). The set $\mathcal{X}$ can be any collection of structured objects, such as graphs, strings, tree, etc. In this respect, it is important to emphasize that structured data are considered to be composed by simpler components combined into a more complex items. Frequently, this combination involves the recursive usage of simpler objects of the same type, e.g., [Comellas et al. \(2004\)](#).

It is not difficult now to intuitively see how h should behave to provide good generalization over all observations from $\mathcal{X}$. Indeed, any $x \in \mathcal{X}$ that is close to an observed input x_i , should have (target) label $y = h(x)$ close to the observed $y_i = h(x_i)$. Thus, the difficulty of this task consists in defining what we mean by “close” both within $\mathcal{X}$ and within $\mathcal{Y}$. (More precisely, we need to quantify the input similarities in $\mathcal{X}$ and the cost of incorrectly assigning outputs within $\mathcal{Y}$. We call this the loss function, which considers the sum of the costs (or losses) of all mistakes made, when we consider a particular possible solution $h(x_i)$ and applies it to all known items $(x_i, y_i) \in \mathcal{T}$. For instance we can choose):

$$\mathcal{L}(h(x_i), y_i) = \sum_{i=1}^N \|h(x_i) - y_i\| \quad (1)$$

Measuring similarity in $\mathcal{Y}$ can be quite easy, for example we could define a zero-one “loss function” such as,

$$\mathcal{L}(y, y_i) = I(y \neq y_i), \quad (2)$$

where I is the indicator function. On the other hand, quantifying similarities in $\mathcal{X}$ can be more difficult, in particular when one has to consider the different nature of objects in $\mathcal{X}$. Kernel methods introduced an interesting and targeted approach to this aim. It is required that, such a measure (let us denote it with k), should be defined as

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}, \quad (3)$$

$$(x_i, x_j) \mapsto k(x_i, x_j).$$

Moreover, k should satisfy the property that

$$k(x_i, x_j) := \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} \quad (4)$$

for all $x_i, x_j \in \mathcal{X}$ (The notation $\langle \cdot, \cdot \rangle$ represents an inner product. This operation is used to abstract the properties of a dot product that takes two equal-length sequences of numbers (usually, coordinate vectors) and returns a single number.). The function $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ in [Eq. \(4\)](#) can be thought as a new representation for the patterns $x \in \mathcal{X}$ in the new feature space $\mathcal{H}$. The advantage offered by the new representation is considerable. Due to this mapping, we can easily transform linear models into nonlinear models. Furthermore, the input domain does not necessarily need to be a vector space.

A symmetric function k , as defined by [Eq. \(4\)](#), for which the property expressed in [Eq. \(4\)](#) holds, is called a kernel function. Another advantage offered by [Eq. \(4\)](#) is expressed by the representation of $k(x_i, x_j)$ in terms of the inner product $\langle \Phi(x_i), \Phi(x_j) \rangle$ in $\mathcal{H}$. Due to this fact (i.e., *inner product representation*), we can think of the kernel as a measure of similarity between input patterns. Whenever [Eq. \(4\)](#) holds, one has an important mathematical tool. Since there exists a feature space $\mathcal{H}$, and a map Φ from $\mathcal{X}$ to the inner product feature space $\mathcal{H}$ (Hilbert space.), the inner product between $\Phi(x_i)$ and $\Phi(x_j)$ in $\mathcal{H}$ can be computed in terms of the kernel evaluations $k(x_i, x_j)$ in $\mathcal{X} \times \mathcal{X}$. Moreover, given a kernel k and a set $S = \{x_1, \dots, x_n\}$, we can define a corresponding *Gram matrix* (sometimes called the *Kernel matrix*) whose entries are expressed as follows.

$$G_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} = k(x_i, x_j). \quad (5)$$

Generally, the only information received by a machine learning algorithm, from the training set, comes from the Gram matrix and its related output values. In other words, all the information that a kernel-based learning algorithm can obtain about training data is provided by the kernel matrix together with any labeled examples. For this reason the Gram matrix can be considered as an information bottleneck that must transmit enough information about the data for the algorithm to be able to perform its task. The Kernel function ([Eq. \(4\)](#)) and the corresponding Gram matrix ([Eq. \(5\)](#)) are both the core and the critical parts of kernel methods. They imply that the observed patterns never have to be expressed explicitly in $\mathcal{H}$, as long as these data only appear as inner products in the considered formulations, i.e., the implemented machine learning algorithm. Since it is possible to construct a function $k(x, x')$, such that $k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$, then during learning, every time we need to compute $\langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$, we can just evaluate $k(x, x')$ - i.e., we don't need to actually apply and express the function $\Phi(x)$. In this way, we say that the transformation would exist only “implicitly”.

A fundamental and natural question, now, is to consider under what circumstances a function k may represent an inner product in order to be considered a valid kernel. It can be proved that every symmetric positive definite function k (i.e., satisfying *Mercer conditions*) represents a valid inner product in some feature space, as defined by [Eq. \(4\)](#), see e.g., [Vapnik \(1995\)](#), [Mercer \(1909\)](#).

Examples of Kernels

The choice of a particular kernel k is not straightforward. k should be chosen carefully, as it is the core of the generalization process. Each choice of kernel will define a different type of feature space and the resulting inference algorithm will perform differently. Moreover, since a kernel can be thought as a measure of similarity, this choice can also be informed by domain expertise, as domain experts already know the actual similarities between objects in their respective contexts. The following examples provide kernel functions that can be broadly applied to various situations. More application-specific kernels will be discussed in the following paragraph.

- Polynomial kernel

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z})^d \quad (6)$$

which, for $d=2$, squares the product between the two vectors $\mathbf{x}=(x_1, x_2)$ and $\mathbf{z}=(z_1, z_2)$. In this case, extending the equation, we get

$$\begin{aligned} (\mathbf{x} \cdot \mathbf{z})^2 &= (x_1 z_1 + x_2 z_2)^2 \\ &= (x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 x_2 z_2) \\ &= \langle (x_1, x_2, \sqrt{2}x_1 x_2), (z_1, z_2, \sqrt{2}z_1 z_2) \rangle \\ &= \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle \end{aligned} \quad (7)$$

The above example shows a typical situation, in all those circumstances e.g., pattern recognition, where even though the input examples $\mathbf{x}$ and $\mathbf{z}$ exist in a dot product space (i.e., $\mathfrak{R}^2$), we may still want to consider more general similarity measure by applying the map defined as

$$\Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2). \quad (8)$$

The function defined in Eq. (8) transforms the data in a higher dimensional space, embedding the input from a two dimensional space in a three-dimensional space. Notice that, the kernel function k in Eq. (7) corresponds to the inner product between two three-dimensional vectors expressed as the new representation, $\Phi(x_1, x_2)$. In other words, the inner product $\langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$ in $\mathfrak{R}^3$ can be easily obtained, in a totally equivalent way, by evaluating the function $(\mathbf{x} \cdot \mathbf{z})^2 \in \mathfrak{R}^2$, i.e., by using the kernel over $\mathcal{X} \times \mathcal{X}$, without explicitly re-writing the data using $\Phi(x_1, x_2)$. As expressed above, one can use the kernel trick to substitute inner products with kernels, whenever the selected function, k , gives rise to a positive definite matrix, $K_{ij}=k(x_i, x_j)$. This requirement ensures the existence of a mapping in some feature space, for which the representation $k(x, z)=\langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$ definitely holds. In general, if we had use a higher exponent, we can virtually embed the two vectors in a much higher dimensional space at a very low computational cost.

- Gaussian kernel also known as radial basis function. Gaussian kernels are one the most widely applied and studied kernels in literature. They are defined as follows

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{\mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{z} + \mathbf{z}^T \mathbf{z}}{2\sigma^2}\right). \end{aligned} \quad (9)$$

Here $\sigma > 0$ controls the flexibility of the kernel. Small values of σ allow classifiers to adapt any label, this way risking overfitting (See, e.g., [Cherkassky and Ma, 2004](#)) for setting the model parameters in order to obtain a good generalization performance.) We can observe that, in the feature space, the images of all points have norm equal to 1, as $k(x, z)=\exp(0)=1$. Moreover, the feature space can be chosen in such a way that all the images lie within a single orthant, since all the inner products between mapped points are positive.

Application-Specific Kernel

The capability to deal with different data is one of the advantages provided by kernel methods. The kernel representation $k(x_i, x_j) := \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}}$ does not assume any specific nature for the patterns x_i, x_j in the input domain $\mathcal{X}$, permitting the kernel to work with every data type our inference may require.

In particular, when input patterns are not defined as vectors (e.g., strings, trees, and graphs), kernels provide a general approach to making vector-based algorithms applicable (using input transformations) to all input data structures. This is a typical situation in real-world problems, where data can be “complex” structures, in some specific form. In these situations, it may be difficult to represent such data as vectors, and even when a vector-based representation is given, i.e., to apply the available vector-based

algorithms, part of information of the original patterns may be irretrievably lost, therefore reducing the discriminative power of the inference mechanism.

Moreover, since the kernel can be interpreted as a similarity measure, when the input structures are typical of some specific domain (for example, bio-sequences, images, or hypertext documents), it is also possible reveal which input data features are most relevant to the problem being considered. Hence, in these cases, the choice and the design of a specific kernel should be discussed together with domain experts, as they know the actual similarities between objects, and can help to properly define a kernel-based similarity measure.

Kernel for Strings

String algorithms and, in particular, string comparison algorithms are key topics in bioinformatics. Comparing DNA sequences can be indicative, for example, in the study of genetic variations with the aim whether two different DNA samples are from the same person, or whether genetic mutations may be critical for particular pathologies.

Measuring the similarity between sequences is therefore a very important task, and has become an essential tool for string comparison. Many approaches can be considered as referring to different ways of counting the sub-strings that two original strings have in common. Typical examples can be found in [Lodhi et al. \(2002\)](#), [Haussler \(1999\)](#). The standard idea is simple: the more sub-strings two strings have in common, the more similar they are. This quantity is generally referred as a *spectrum kernel*, and can be formulated as follows

$$k(s_i, s_j) = \sum_{q \in ((\Sigma))^n} \#(q \subset s_i) \cdot \#(q \subset s_j), \quad (10)$$

where $\#(q \subset s_i)$ is the frequency of sub-string $q \in ((\Sigma))^n$ in s_i . The kernel in [Eq. \(10\)](#) counts common sub-strings of length n in the two strings s_i and s_j . The advantage of this simple formulation is its fast computational time i.e., $O(|s_i| + |s_j|)$, where $|s|$ is the length of a string s .

The spectrum kernel has been extended in several ways, for example, by weighting the occurrences of sub-strings depending on how they are distributed within the original input. In this case, the kernel also takes into account the presence of gaps introduced to maximize the similarity of the sub-strings: see, for example, [Gordon et al. \(2003\)](#), [Rousu and Shawe-Taylor \(2005\)](#).

Kernel for Documents

Document analysis is certainly a challenging area of research within the machine learning community ([Joachims, 1998](#); [Yang and Liu, 1999](#)). A problem that most attracted the attention of researchers was the comparison and classification of the content of different documents (i.e., automatic assignment of a random document to one or more classes.). (Another important problem is the *document summarization* which refers to the process of extracting or generating shortened content from one or various sources, e.g., [Mamakis et al. \(2012\)](#)). Even in this case, the identification of appropriate similarity measures has had immediate results for many practical tasks, such as the office automation.

Significant results in this area have been obtained in Information Retrieval (IR) studies, which have provided the most commonly used representation to evaluate the content of documents. This representation is actually known as the Vector Space Model (VSM) (or bag-of-words) see, e.g., [Zhang et al. \(2010\)](#). The main idea of this method, can be given by introducing the following preliminary definitions.

Definition 4.1: (Bag). A Bag is a collection in which repeated elements are allowed.

Definition 4.2: (Words). Words are any sequence of letters from the basic alphabet separated by punctuation or spaces.

Definition 4.3: (Dictionary). A Dictionary can be defined by a predetermined set of terms.

Definition 4.4: (Corpus). Sometimes only those words belonging to the set of documents being processed are considered. In this case, we refer to this big collection of text (documents) as the corpus, and the set of terms occurring in the corpus as the dictionary.

Following the above terminology, a document can be viewed as a bag-of-words, and can be represented by a vector where each element is associated with one word from the dictionary. More formally,

$$(\text{tf}_d(t_1), \text{tf}_d(t_2), \dots, \text{tf}_d(t_n)), \in \mathfrak{R}^n \quad (11)$$

where $\text{tf}_d(t_i)$ is the frequency of the term t_i in the document d . In this way, we can consider the mapping $\Phi : d \rightarrow \Phi(d) \in \mathfrak{R}^n$ that represents a document as a point in a vector space of dimension n , which is the size of the dictionary. Typically, $\Phi(d)$ is a vector with many elements or features, and in many problems we can find more features than document examples within the training set. However, for a particular document, we have a sparse representation in which only a few elements have non zero entries.

Extending the representation in expression (11), we can define the following matrix.

$$D = \begin{pmatrix} \text{tf}(d_1, t_1) & \dots & \text{tf}(d_1, t_n) \\ \vdots & \ddots & \vdots \\ \text{tf}(d_n, t_1) & \dots & \text{tf}(d_n, t_n) \end{pmatrix} \quad (12)$$

The matrix D defined in Eq. (12) is known as a document-term matrix. Its rows are indexed by the documents of the corpus and its columns are indexed by the words of the text (terms). Hence, the (i, j) th value of D gives the frequency of term t_j in document d_i . Notice that, in IR, we can also find the so called term-document matrix, that is, the transpose D' of the document-term matrix defined in (12). With this definition, we can finally formalize the vector space kernel, as follows.

Definition 4.5: (Vector Space Kernel). Let us consider the matrix K defined as $K = DD'$. This matrix leads to the definition of the corresponding vector space kernel given by

$$k(d_1, d_2) = \langle \Phi(d_1), \Phi(d_2) \rangle = \sum_{j=1}^n \text{tf}(d_1, t_j) \text{tf}(t_j, d_2). \quad (13)$$

One problem with the above definition is that the ordering of the words is completely ignored. This is clearly a critical feature, since in most natural languages this circumstance modifies the semantics of the sentences. In addition, inferential aspects of several important tasks (e.g., speech recognition or short message classification), may be compromised if the relationship between words in the corpus is not taken into account. To address this problem, the vector space model has been extended by considering different ways to introduce semantic considerations in the formulation of the kernel (Shawe-Taylor and Cristianini, 2004).

Kernel for Graphs

Graph theory is a rich environment of powerful abstract techniques that allow modeling of several types of problems, both natural and human-made, ranging from biology to sociology (Mason and Verwoerd, 2007).

It is in this context that a recent focus in *machine learning* (Mitchell, 1997; Kolaczyk, 2014; Witten et al., 2011; Marsland, 2011) has been to extend traditional problems to systems of complex interactions, and more generally, to networks (Getoor and Taskar, 2007; Bansal et al., 2002). In this case, traditional algorithms may not only benefit from the information provided by relationships between instances (Zoppis and Mauri, 2008), but could even fail to achieve adequate performance when relational data are not properly supplied to a classification system.

A graph provides an abstract representation of the relationships among the elements of a set. Formally, we refer to a graph as $G = (V, E)$, where V is the set of elements (i.e., nodes or vertices) and E is the set of relationships between pairs of elements (i.e., edges). We use the notation $|V|$ to denote the number of nodes, and $|E|$ to denote the number of edges in the graph G . If u and v are two nodes and there is an edge from u to v , then we write that $(u, v) \in E$, and we say that v is a *neighbor* of u ; and u and v are adjacent.

The challenge of proposing a definition for a proper graph kernel is to find a suitable measure that captures a particular graph structure and, at the same time, is efficient to evaluate. For example, an elegant formulation, known as the *random-walk graph* kernel, can be found in Gartner et al. (2003), Kashima et al. (2003). To give a proper formulation for this kernel, let us consider the direct product (tensor product) of graphs, defined as a graph $G_x(G_1, G_2) = (V_x, E_x)$ whose vertices and edges are given as follows (See also Imrich and Klavzar, 2000.)

$$V_x = \{(v_1, w_1) : v_1 \in V, w_1 \in W, \text{Label}(v_1) = \text{Label}(w_1)\} \quad (14)$$

$$E_x = \{(v_1, w_1), (v_2, w_2) \in V_x \times V_x : (v_1, v_2) \in E, (w_1, w_2) \in F, \text{Label}(v_1, v_2) = \text{Label}(w_1, w_2)\}$$

where (V, E) and (W, F) are respectively, the vertices and the edges of the graphs $G_1 = (V, E)$ and $G_2 = (W, F)$. Using definition (14), the random-walk kernel can be defined as

$$k(G, G') = \sum_{i,j=1}^{|V_x|} \left[\sum_{k=0}^{\infty} \lambda_k A_x^k \right]_{i,j} \quad (15)$$

where A_x is the adjacency matrix of the product graph in (14). The sum in Eq. (15) converges for a suitable choice of the weights $\lambda_0, \lambda_1, \lambda_2, \dots$ (Gartner et al., 2003).

Kernel Composition

One of the attractiveness of kernel methods is due to the closure properties which kernels possess. This property can be summarized by the following proposition.

Proposition 1.: Let k_1, k_2 and k_3 be kernels over $\mathcal{X} \times \mathcal{X}$, $\mathcal{X} \subset \mathbb{R}^n$, $f(\cdot)$ a real-valued function on $\mathcal{X}$, $\Phi : \mathcal{X} \rightarrow \mathbb{R}^n$, B a positive semi-definite $n \times n$ matrix, and $p(x)$ a polynomial with positive coefficients. Then the following functions are kernels.

1. $k(x, z) = k_1(x, z) + k_2(x, z)$
2. $k(x, z) = \alpha k_1(x, z)$
3. $k(x, z) = k_1(x, z) + k_2(x, z)$
4. $k(x, z) = f(x)f(z)$
5. $k(x, z) = k_3(\Phi(x), \Phi(z))$
6. $k(x, z) = x' B z$
7. $k(x, z) = p(k_1(x, z))$
8. $k(x, z) = \exp(k_1(x, z))$

While the above property can be formally proven, from an applicative point of view, the importance of this result lies in the possibility of creating new kernels using already defined kernel functions. In this way, we can get more complex kernels from simple building blocks by simply applying a number of successive main operations.

This approach also has an important theoretical consequence: it provides a formal tool by which it is possible to show that new functions actually possess the property of being a kernel, i.e., of showing they are semi-positive defined functions. Often, this way of defining new kernel functions provides enough information for users to build suitable kernels for particular applications.

Kernel Algorithms

Many machine learning algorithms can be implemented in such a way that input patterns can be accessed only through inner product operations, using the representation $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}}$. Such algorithms are said to employ kernels, since the pairwise inner products can be computed directly from the original pattern space using the kernel function $k(x_i, x_j)$. In such a situation, a nonlinear version of a particular inference model can be easily obtained by simply replacing the inner products with the kernel k . This is called the kernel trick.

As an example, let us consider the case where the inference algorithm has to evaluate norms of vectors in the feature space. This computation, can be formulated in the following simple but important way

$$\|\Phi(x)\|_{\mathcal{H}} = \sqrt{\langle \Phi(x), \Phi(x) \rangle} = \sqrt{k(x, x)}. \quad (16)$$

The example in Eq. (16) shows that, despite the apparent impossibility of the evaluation of $\Phi(x)$, we can calculate this quantity by substituting the inner product with the kernel function k . Many well-known learning algorithms have been “kernelized” and have provided successful results in scientific disciplines such as bioinformatics, natural language processing, computer vision and robotics. Just to mention a few well known fundamental machine learning problems, we can refer to the following examples.

- **Classification.** As discussed above, classification is the task of identifying to which of a set of categories new observations belong. For a long time, the most popular approach to solving non-linear classification problems (i.e., the classification of non-separable data) was to apply a multilayer perceptron, i.e., an artificial neural network model that maps a set of input data onto a set of appropriate outputs (Rumelhart *et al.*, 1988). The introduction of SVMs – the most typical kernel based method (Boser *et al.*, 1992), lead to a completely different point of view. In this case, the novel idea was to detect, in the feature space, a discriminating hyperplane that maximize the distance between the nearest points of the two classes (i.e., the maximum margin hyperplane).
- **Regression.** When data are non-linearly related and the task is to fit, as much as possible, the observed examples with a smoothed function, kernel regression can be effectively applied. Similarly to SVM classification, the main mathematical tools of this technique come from mathematical optimization theory.
- **Principal Component Analysis (PCA).** PCA projects the input patterns into a new space spanned by the principal components. Each component is selected to be orthonormal to the others and to capture the maximum variance that is not already accounted for by the previous components.
- Similarly to other kernel algorithms, PCA can be extended to nonlinear settings by replacing PCA in the pattern space by a feature space representation. The final algorithm consists of solving an eigenvector problem for the kernel matrix (Mika *et al.*, 1999).
- **Independent Component Analysis (ICA).** The goal of ICA is to find a linear representation of non-Gaussian data in such a way that the components are statistically independent (or as independent as possible).

This representation can be used to capture the essential structure of the data in different applications, including feature extraction and signal separation.

The kernel ICA (Bach and Jordan, 2002) is based on the minimization of a contrast function, following typical kernel approach ideas. Specifically, a contrast function, in this case, measures the statistical association between components. It can be shown that minimizing these criteria leads to flexible and robust algorithms.

Moreover, a relatively recent research area concerns the study of deep neural networks (Hinton *et al.*, 2006; Deng *et al.*, 2014) and their relationships to kernel approaches (Yger *et al.*, 2011; Cho and Saul, 2009). In this respect, it is important to note that

generalization ability is limited by the choice of a proper kernel function, i.e., the choice of an appropriate kernel has a fundamental effect on the inference performance. For this reason, researchers – in particular from neural network community – attempted to solve this limitation by considering the so called deep kernel strategies.

This paradigm aims at learning useful features by stacking successive layers of transformations. The information represented at each layer is produced by a nonlinear projection of output of the previous layer using kernel method.

In general, researchers have advanced several motivations for these systems, for example, the ability to parametrize a wide number of functions through the combination of simple nonlinear transformations, the expressiveness of distributed hierarchical representations, and the ease of combining supervised and unsupervised methods. Experiments have also shown the benefits of deep learning in several interesting applications (Collobert and Weston, 2008; Bengio *et al.*, 2009).

Conclusion

Kernel methods have not only enriched the machine learning research by offering the opportunity to dealing with different tasks and different input structures, but have also provided new perspectives for solving typical problems with a methodology supported by strong intuition and well founded mathematical theory.

The discussion outlined in this article is intended to provide the fundamental idea of these techniques: whenever a procedure is adapted to use only inner products between input examples, then in that procedure, inner products can be replaced by a kernel function. By exploiting the so-called kernel trick, we have seen that one can straightforwardly perform nonlinear transformations of the original patterns into a (generally) higher dimensional feature space – which is consequently nonlinearly related to the input space. Many algorithms in the scientific literature have been “kernelized” (i.e., adapted) in the manner described above.

References

- Bach, F.R., Jordan, M.I., 2002. Kernel independent component analysis. *Journal of Machine Learning Research* 3, 1–48.
- Bansal, N., Blum, A., Chawla, S., 2002. Correlation clustering. *Machine Learning*, 238–247.
- Bengio, Y., *et al.*, 2009. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning* 2, 1–127.
- Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152. ACM.
- Cava, C., Zoppis, I., Gariboldi, M., *et al.*, 2013a. Copy-number alterations for tumor progression inference. In: Peek, N., Marn Morales, R., Peleg, M. (Eds.), *Artificial Intelligence in Medicine*. Berlin Heidelberg: Springer, pp. 104–109. (volume 7885 of *Lecture Notes in Computer Science*).
- Cava, C., Zoppis, I., Gariboldi, M., *et al.*, 2014. Combined analysis of chromosomal instabilities and gene expression for colon cancer progression inference. *Journal of clinical bioinformatics* 4, 2.
- Cava, C., Zoppis, I., Mauri, G., *et al.* 2013b. Combination of gene expression and genome copy number alteration has a prognostic value for breast cancer. In: *2013 Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 608–611. IEEE.
- Cherkassky, V., Ma, Y., 2004. Practical selection of svm parameters and noise estimation for SVM regression. *Neural Networks* 17, 113–126.
- Chinello, C., Cazzaniga, M., De Sio, G., *et al.*, 2014. Urinary signatures of renal cell carcinoma investigated by peptidomic approaches. *PLOS ONE* 9, e106684.
- Cho, Y., Saul, L.K., 2009. Kernel methods for deep learning. *Advances in Neural Information Processing Systems*, 342–350.
- Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th international conference on Machine learning*, ACM. pp. 160–167.
- Comellas, F., Fertin, G., Raspaud, A., 2004. Recursive graphs with small-world scale-free properties. *Physical Review E* 69, 037104.
- Deng, L., Yu, D., *et al.*, 2014. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing* 7, 197–387.
- Gartner, T., Flach, P., Wrobel, S., 2003. On graph kernels: Hardness results and efficient alternatives. *Learning Theory and Kernel Machines*. Springer. pp. 129–143.
- Getoor, L., Taskar, B., 2007. *Introduction to Statistical Relational Learning* (Adaptive Computation and Machine Learning). The MIT Press.
- Gordon, L., Chervonenkis, A.Y., Gammerman, A.J., Shahmuradov, I.A., So-lovyev, V.V., 2003. Sequence alignment kernel for recognition of promoter regions. *Bioinformatics* 19, 1964–1971.
- Haussler, D., 1999. Convolution kernels on discrete structures. Technical Report, Citeseer.
- Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 1527–1554.
- Imrich, W., Klavzar, S., 2000. *Product Graphs*. Wiley.
- Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML- 98*, 137–142.
- Kashima, H., Tsuda, K., Inokuchi, A., 2003. Marginalized kernels between labeled graphs. In: *ICML*, pp. 321–328.
- Kolaczyk, E., 2014. *Statistical Analysis of Network Data with R*. SpringerLink. Buocher: Springer.
- Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C., 2002. Text classification using string kernels. *Journal of Machine Learning Research* 2, 419–444.
- Mamakis, G., Malamos, A.G., Ware, J.A., Karelli, I., 2012. Document classification in summarization. *Journal of Information and Computing Science* 7, 025–036.
- Marsland, S., 2011. *Machine Learning: An Algorithmic Perspective*. CRC Press.
- Mason, O., Verwoerd, M., 2007. Graph theory and networks in biology. *IET Systems Biology* 1, 89–119.
- Mercer, J., 1909. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, Containing Papers of a Mathematical or Physical Character* 209, 415–446.
- Mika, S., Scholkopf, B., Smola, A.J., *et al.*, 1999. Kernel pca and de-noising in feature spaces. *Advances in Neural Information Processing Systems*, 536–542.
- Mitchell, T., 1997. *Machine Learning*. New York, NY: McGraw-Hill, Inc.
- Pekalska, E., Duin, R.P., 2005. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, vol. 64. World Scientific.
- Ramakrishnan, N., Tadepalli, S., Watson, L.T., *et al.*, 2010. Reverse engineering dynamic temporal models of biological processes and their relationships. *Proceedings of the National Academy of Sciences* 107, 12511–12516.
- Rousu, J., Shawe-Taylor, J., 2005. Efficient computation of gapped substring kernels on large alphabets. *Journal of Machine Learning Research* 6, 13231344.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors. *Cognitive Modeling* 5, 1.
- Scholkopf, B., Burges, C.J., 1999. *Advances in Kernel Methods: Support Vector Learning*. MIT press.

- Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge university press.
- Smola, A.J., Schölkopf, B., 1998. Learning with Kernels. Citeseer.
- Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. New York, NY: Springer-Verlag.
- Witten, I., Frank, E., Hall, M., 2011. Data Mining: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science.
- Yang, Y., Liu, X., 1999. A re-examination of text categorization methods. In: Proceedings of the 22nd Annual international ACM SIGIR Conference on Research and development in information retrieval, ACM. pp. 42–49.
- Yger, F., Berar, M., Gasso, G., Rakotomamonjy, A., 2011. A supervised strategy for deep kernel machine. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp. 501–506.
- Zhang, Y., Jin, R., Zhou, Z.H., 2010. Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics 1, 43–52.
- Zoppis, I., Mauri, G., 2008. Clustering dependencies with support vectors. Lecture Notes in Electrical Engineering 6, 155–165.

Further Reading

- Andrew, A.M., 2000. An introduction to support vector machines and other kernel-based learning methods by nello cristianini and john shawe-taylor. xiii + . Cambridge: Cambridge University Press, p. 189.
- Antoniotti, M., Carreras, M., Farinaccio, A., *et al.*, 2010. An application of kernel methods to gene cluster temporal metaanalysis. Computers & Operations Research 37, 1361–1368.
- Burges, C.J., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167.
- Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines. Cambridge: Cambridge University Press
- Hofmann, T., Schölkopf, B., Smola, A.J., 2008. Kernel methods in machine learning. The Annals of Statistics. 1171–1220.
- Schölkopf, B., Smola, A.J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT press.
- Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press.
- Wang, L., 2005. Support Vector Machines: Theory and Applications. vol. 177. Springer Science & Business Media.
- Zoppis, I., Merico, D., Antoniotti, M., Mishra, B., Mauri, G., 2007. Discovering relations among go-annotated clusters by graph kernel methods. In: International Symposium on Bioinformatics Research and Applications, pp. 158–169. Berlin: Springer.

Kernel Methods: Support Vector Machines

Italo Zoppis and Giancarlo Mauri, University of Milan-Biocca, Milan, Italy

Riccardo Dondi, University of Bergamo, Bergamo, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Among the most mentioned algorithms in the scientific literature of machine learning, Support Vector Machines (SVMs) have played an important role (Vapnik, 1995, 1998) since their introduction in 1992 at the Conference on Computational Learning Theory by Boser *et al.* (1992). They provide a general approach to different inferential tasks such as classification, clustering, and regression, based on a consistent mathematical theory and intuitive functional aspects.

To better understand the formal arguments presented in this article, we will first introduce three main key concepts on which the SVM's functional mechanism is conceptually motivated: i.e., maximum margin hyperplane, the kernel function, and non-linearly separable data. We will mainly refer to typical machine learning procedures, first of all considering the classification problem.

Subsequently, the fundamental mathematical issues for linear and nonlinear classification will be introduced. Notice that, while the mathematical tools of these techniques lie in optimization theory, here we will confine the formal arguments only to the most intuitive topics as well as, to the relationships that express the reasons why the SVMs and the kernel methods have found an effective coexistence.

Finally, considering different contributions in the literature, we will see how binary classification can be extended to clustering, regression and to multi-class classification.

Maximum Margin Hyperplane

Maximum margin hyperplane is one of the key concepts that best explains how SVMs work in practice. To better understand the main idea behind its usage, let us consider the following procedure for separating two kinds of objects (generally called *patterns*).

1. Represent input patterns in a high dimensional space, and
2. draw a (*decision*) linear boundary between them.

This procedure solves one of the most studied task in the machine learning community: the classification problem. The objective of classification is identifying to which of a set of categories new observations belong, on the basis of a sample of data representing observations whose category membership are known (*training set*).

The training set constitutes a collection of labeled examples that might be useful to build the decision boundary, i.e., to fit a prediction model for new unobserved cases. For example, in a typical healthcare classification problem, one might be asked to determine whether new subjects can be discriminated from case patients (diseased patients) or classified as controls (healthy) according to previous sets of sampled attribute values (see, for example, Chinello *et al.*, 2014; Cava *et al.*, 2013a,b when data are combined from different sources of information.).

In general, when the space of the representation (the input space, or as it is sometimes called, *the pattern space*) is $\mathcal{R}^2$, the boundary separates the plane into two half-planes, one that contains positive patterns, and one that contains negative ones. When it is assumed that the input data satisfy the condition of being linearly-separable, it is always possible to find such straight line in $\mathcal{R}^2$. When we have $\mathcal{R}^d$ with $d > 2$, we say that a separating hyperplane exists, and a linear separation boundary can discriminate the two classes of points. After successful training (i.e., identification of the decision boundary), the trained model should correctly predict the output membership class of new patterns, depending on whether new test points fall either within the positive or the negative semi-plane.

There are an infinite number of decision boundaries suitable to discriminate patterns, and many classifiers in literature, following some particular criterion (e.g., gradient descent for back-propagation networks) have achieved accurate performances. As we will see, compared to other methods, SVMs perform this task in a particularly targeted and intuitive manner. The discriminating boundary is given by a hyperplane suitably located to maximize the distance (the margin) that separate it from the closest positive and negative examples (known as support vectors). In this case, the separating hyperplane is called the *maximum margin hyperplane* (Fig. 1).

In order to explain intuitively why this mechanism works, we can consider a case where no preference is given to a particular input distribution, if we place, for example, the boundary too near to the negative points, we would over-predict positive labels (i.e., labeled as $y = +1$). Instead, we will risk for false assignments, on cases labeled as $y = -1$. In this way, a reasonable choice would place the separation boundary as far away as possible from both the closest positive and negative training examples, or in other words, as we will see more formally, right in the middle of these points. More importantly, this insight is supported by statistical learning theory (Vapnik, 1995), that shows that the ability to generalize, in this case, depends on the margin (defined as the distance of the decision hyperplane from the closest data), rather than the dimensionality of the feature space.

Soft Margin and Slack Variables

In practice, it is difficult to find linearly separable data. Noise is a typical characteristic of real-world contexts, which makes it unlikely to detect linear boundaries in classification problems. In such cases, the constraint of maximizing the margin of a linear

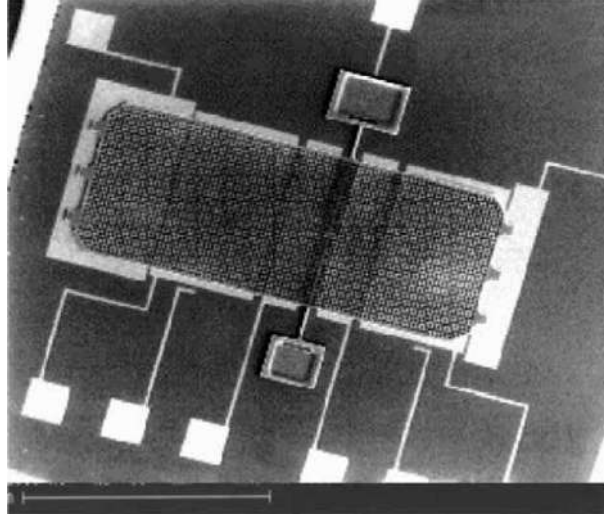


Fig. 1 The margin can be defined as the distance between the separating hyperplane and the hyperplanes through the closest points (i.e., support vectors). x_1 and x_2 are examples of support vectors from different classes.

separator can be relaxed. This simple solution is generally known as a *Soft Margin Classifier* (Shawe-Taylor and Cristianini, 2002). This problem motivates the development of more robust solutions that can tolerate noise and, more generally, variability and errors in the training set, without drastically altering the final results.

In order to train a Soft Margin Classifier, an additional set of coefficients is generally introduced that, in turn, allow margin variations for each defined space dimension. These coefficients are known as *slack variables*. Clearly, this increases the complexity of the model as there are more parameters to fit.

Kernel Function

Although the soft margin approach may work for data that are close to being linearly separable, it returns poor results when inputs are strongly affected by errors and variations. In this case, the solution offered by kernels comes into play. This is generally the case for real data sets, where training examples, are generally dispersed and overlapped up to some extent.

It is important to recall that kernel functions work by embedding input patterns into a vector space, where geometric and algebraic techniques can be properly applied to provide linear separation. In particular, by reformulating the learning algorithms in a way that only requires knowledge of the inner product between the input in the embedding space, one can use a kernel function without explicitly performing such embedding. There is no need to know the coordinates of each point in the embedding space, rather it is possible, thanks to the evaluation of the kernel function to directly obtain the inner products between points in such a space. The kernel approach of the SVMs is relatively simple, and can be described as follows.

1. Embed the training patterns into a new (generally, high dimensional) space, and
2. separate, in that space, the two classes of points using a maximum margin hyperplane.

SVM Formulation for Classification Problems

As mentioned above, the intuitive idea that describes how SVMs perform binary classification, is based on the maximum margin hyperplane criterion, i.e., SVM finds the plane that discriminates positive from negative labeled data with the maximum margin. This margin is defined as the distance from the closest input training examples (or support vectors). We now consider two main cases. The first, when data are linearly separable. The second, when they are not.

Separable Data

Let us assume, we are given the following set of examples (i.e., the training set).

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\} \in \mathcal{X} \times \mathcal{Y} \quad (1)$$

where $\mathbf{x}_i$ is a real vector in $\mathcal{R}^n$ and y_i is either 1 or -1 , representing the class to which point $\mathbf{x}_i$ belongs. Since the objective of SVM is to find a separating hyperplane, in this case, at least one such hyperplane must exist. This hyperplane is generally denoted as

$(\mathbf{w}, b)$, and can be described by the points that satisfy the following general equation.

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = \sum_{i=1}^N w_i x_i + b = 0 \quad (2)$$

Geometrically, $\mathbf{w}$ is a normal vector of the hyperplane that defines the hyperplane orientation, and b is a scalar called the intercept or bias.

Any hyperplane $\pi = (\mathbf{w}, b)$ divides the space $\mathfrak{R}^n$ into two parts, located on opposite sides of the hyperplane, called the positive and negative half-spaces. The positive half-space $(\mathfrak{R}^n)^+_\pi$ is specified by the normal vector of the hyperplane, and for any vector $\mathbf{x} \in (\mathfrak{R}^n)^+_\pi$ we have $\langle \mathbf{w}, \mathbf{x} \rangle + b > 0$, while for any $\mathbf{x} \in (\mathfrak{R}^n)^-_\pi$ we have $\langle \mathbf{w}, \mathbf{x} \rangle + b < 0$.

In other words, for linearly separable data, any hyperplane $(\mathbf{w}, b)$ in $\mathfrak{R}^n$, induces a linear decision boundary able to discriminate the two classes of points $C_1 = \{\mathbf{x} : (\mathbf{x}, \gamma) \in \mathcal{T}, \gamma = +1\}$ and $C_2 = \{\mathbf{x} : (\mathbf{x}, \gamma) \in \mathcal{T}, \gamma = -1\}$ if the following fact holds.

$$\langle \mathbf{w}, \mathbf{x} \rangle + b > 0 \quad \forall \mathbf{x} \in C_1$$

$$\langle \mathbf{w}, \mathbf{x} \rangle + b < 0 \quad \forall \mathbf{x} \in C_2$$

Therefore, we can associate to the separating hyperplane $(\mathbf{w}; b)$ the following (decision) function.

$$h_{\mathbf{w}, b}(\mathbf{x}) = \text{sgn}\{\langle \mathbf{w}, \mathbf{x} \rangle + b\} = \begin{cases} +1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle + b > 0 \\ -1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle + b < 0 \end{cases} \quad (3)$$

Function in Eq. (3) gives us a rule that correctly classifies all vectors from C_1 and C_2 , using the training examples in $\mathcal{T}$. The SVM training phase therefore has the task to find the weights, $\mathbf{w} = (w_1, w_2, \dots, w_N)$, and the bias, b , of the hyperplane $(\mathbf{w}, b)$.

Functional and Geometric Margin

To provide a formulation for the SVM algorithm, it is instructive to start with a formal definition of the notion of margin. The idea of margin is justified by the need to seek maximum separation between the classes, as described intuitively in the previous paragraph. For this reason, it should be useful to have a confidence measure, on the hyperplane's classification ability, to inform us whether a particular point is properly classified. Moreover, this measure should also quantify the magnitude of the distance between patterns and the separating hyperplane. Clearly, the greater the distance, the greater our confidence in the classification.

Let us assume that $\mathbf{x}_i$ is a vector in $\mathfrak{R}^n$, and $(\mathbf{x}_i, \gamma_i)$ an example in $\mathcal{T}$. We call *functional margin* $\hat{\gamma}_i$ of $\mathbf{x}_i$ with respect to the hyperplane $(\mathbf{w}, b)$ the following quantity.

$$\hat{\gamma}_i = \gamma_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \quad (4)$$

Notice that if $\gamma_i = 1$, then for the functional margin to be large (i.e., for our prediction to be confident and correct), we need $\langle \mathbf{w}, \mathbf{x}_i \rangle + b$ to be a large positive number. Conversely, if $\gamma_i = -1$, then for the functional margin to be large, we need $\langle \mathbf{w}, \mathbf{x}_i \rangle + b$ to be a large negative number. Therefore, if $\gamma_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$, our prediction of the example $(\mathbf{x}_i, \gamma_i)$ is correct. Hence, a large functional margin represents a confident and a correct prediction. However, we have also to consider that by scaling $\mathbf{w}$ and b by any constant factor α (for example, if we replace $\mathbf{w}$ by $3\mathbf{w}$ and b by $3b$), we could make the margin value much larger, but there would be no change in the performance of classifier itself. Indeed, when we try to optimize the expression 4, we can always make the functional margin as big as we wish by simply scaling up $\mathbf{w}$ and b , without really changing anything meaningful in the set of points that satisfy the following equation.

$$\alpha (\langle \mathbf{w}, \mathbf{x} \rangle + b) = 0$$

Thus, we can always obtain an infinite number of equivalent formulations for the same hyperplane. To remove this ambiguity, we consider the following expression.

$$\gamma_i = |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| / \|\mathbf{w}\| \quad (5)$$

Eq. (5) simply normalizes the functional margin by the euclidean norm, $\|\mathbf{w}\|$, of $\mathbf{w}$. Since $\gamma_i \in \{-1, +1\}$, Eq. (5) can also be written (for correctly classified patterns) as follows.

$$\gamma_i = \gamma_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) / \|\mathbf{w}\| \quad (6)$$

The quantity γ_i is known as *geometric margin* of $(\mathbf{w}, b)$ with respect to the training example $(\mathbf{x}_i, \gamma_i)$. Note that if $\|\mathbf{w}\| = 1$, then the functional margin equals the geometric margin, thereby relating the two different notions of margin. Also, the geometric margin is invariant to rescaling of the parameters: by replacing $\mathbf{w}$ with $2\mathbf{w}$ and b with $2b$, the geometric margin does not change. Moreover, we can also define the geometric margin of $(\mathbf{w}, b)$ with respect to the training set $\mathcal{T}$ as the smallest of the geometric margins on the individual training examples, i.e.,

$$\gamma = \min_{i=1, \dots, N} \gamma_i \quad (7)$$

A similar definition can also be given for any functional margin $\hat{\gamma}$.

SVM – Primal Optimization Problem

Geometric margin γ provides a natural measure to formulate the maximum margin criterion. This formulation can be described as the following optimization problem.

$$\begin{aligned} \max_{\mathbf{w}, b} \quad & \gamma \\ \text{s.t.} \quad & \gamma_i(\mathbf{w}'\mathbf{x}_i + b) \geq \gamma, \forall i \\ & \|\mathbf{w}\| = 1 \end{aligned} \quad (8)$$

Where, $\mathbf{w}'$ is the transpose of $\mathbf{w}$, which is needed to calculate Eq. (8).

The problem reported in expression 8 maximizes γ , subject to each training example having a functional margin of at least γ . The constraint $\|\mathbf{w}\| = 1$ ensures that the functional margin equals the geometric margin. In this way, we are also guaranteed that all the geometric margins (with reference to all training examples) are at least γ . Thus, solving problem 8 will result in the identification of a hyperplane $(\mathbf{w}, b)$ with the largest possible geometric margin, with respect to the training $\mathcal{T}$. Since the geometric and functional margins are related by $\gamma = \hat{\gamma}/\|\mathbf{w}\|$, we can write the following problem.

$$\begin{aligned} \max_{\mathbf{w}, b} \quad & \hat{\gamma}/\|\mathbf{w}\| \\ \text{s.t.} \quad & \gamma_i(\mathbf{w}'\mathbf{x}_i + b) \geq \hat{\gamma}, \forall i \end{aligned} \quad (9)$$

The objective of Eq. (9) is still non-convex, but considering our earlier discussion on margins, we can arbitrarily scale $\mathbf{w}$ and b without influencing the final computation. This is the key idea we'll apply now.

The scaling constraint can be introduced by requiring that the functional margin of $(\mathbf{w}, b)$, with respect to the training set, must be equal to 1, i.e., $\hat{\gamma} = 1$. Inserting this into Eq. (9), and noting that maximizing $\hat{\gamma}/\|\mathbf{w}\| = 1/\|\mathbf{w}\|$ is the same thing as minimizing $\|\mathbf{w}\|^2$, we have the following formulation.

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \gamma_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1, \forall i \end{aligned} \quad (10)$$

Eq. (10) is convex with a quadratic objective and linear constraints. Thus, we can finally obtain the solution by applying any quadratic optimization solver (Ding *et al.*, 2004).

SVM – Dual Optimization Problem

The concept of duality plays an important role in the theory of mathematical programming. It turns out that, for many optimization problems, we can easily provide an associated optimization problem, called a dual, whose solution is related to the original (primal) problem solution. Specifically, for a wide range of problems, the primal solutions can be computed from the dual ones.

In case of the maximum margin hyperplane, the dual formulation has two major benefits: its constraints are easier to handle than the constraints of the primal problem, and it is better suited to deal with the kernel function. Moreover, the theory of duality guarantees that the unique solution of the primal problem corresponds to the unique solution of the dual problem. Here, without discussing further mathematical details of the primal-dual theory (the readers interested in the derivation details are encouraged to see, for example, Rockafellar (1982)), we obtain the following dual formulation.

$$\begin{aligned} \max_{\alpha} \quad & \mathcal{W} = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \gamma_i \gamma_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & \sum_i \alpha_i \gamma_i = 0 \\ & \alpha_i \geq 0, i = 1, 2, \dots, N \end{aligned} \quad (11)$$

Specifically, in the dual formulation in Eq. (11), we have a maximization problem in which the parameters are the coefficients, α_i , known as *Lagrangian multipliers*, for the training points $i = 1, \dots, N$. Eq. (11) is still a quadratic program and can be solved using any quadratic programming algorithm. Solving this problem for α_i , $i = 1, \dots, N$, i.e., by finding the α_i 's that maximize $\mathcal{W}(\alpha)$, permits (again, using the primal-dual theory) one to find the optimal value for the maximum margin hyperplane, $(\mathbf{w}^*, b^*)$, leading to the following classification function for new test points.

$$h_{\mathbf{w}^*, b^*}(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^N \alpha_i^* \gamma_i \mathbf{x}_i^T \mathbf{x} + b^* \right) \quad (12)$$

The base mathematical theory (specifically, the Karush-Kuhn-Tucker conditions) provides also useful information about the structure of the solution. In our case, we have the following important facts.

- The points $\mathbf{x}_i$, for which $\alpha_i^* > 0$ contributes to the weight vector (i.e., the location of the large margin hyperplane), and have minimum margin, are called *support vectors*. Importantly, due to this property the number of support vectors is generally very small.
- All the other training examples do not contribute to the weight vector. If we remove all the examples $(\mathbf{x}_i, y_i) \in \mathcal{T}$ for which the α_i^* s are not support vectors, we get the same solution $(\mathbf{w}^*, b^*)$.

The support vectors give, in some sense, a compact representation of the data. SVMs neglect non-informative patterns and considers only those data near to the optimal hyperplane. This compact representation of the solution is useful, especially when large data-sets are considered. Finally, notice that both Eq. (11) and the Eq. (12) exclusively involve inner products on input patterns, thus the dual problem provides not only a different perspective for optimization, but also a way of employing kernel functions, as we will see in the next paragraphs.

Non-Separable Data

Since it is not unusual to find non-separable data, it is necessary to adapt the formulation of Eq. (8). The case we discuss here makes it possible to have violations (i.e., errors) of the discrimination ability of the linear decision boundary. This case is also note as a *soft margin* classifier. For this purpose, when using a linear discriminant, it is possible to provide constraint relaxations, in such a way that errors are allowed, up to some extent. This can be done by introducing *slack variables*, that characterize errors and, at the same time, a control parameter in the objective function to penalize such violations. More specifically, this can be formulated by taking into account, for each example, the following constraints.

$$y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i$$

where ξ_i are the slack variables that quantify the amount by which the discriminant function fails to reach the (functional) unit margin. The optimization problem then becomes,

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 - \xi_i, i = 1, \dots, N \\ & \xi_i \geq 0, i = 1, \dots, N \end{aligned} \quad (13)$$

Eq. (13) is still a quadratic program. Generally, C is a positive constant, introduced to control the tradeoff between the penalty and the margin.

Following a similar argument for the separable case, one can obtain the following dual formulation.

$$\begin{aligned} \max_x \quad \mathcal{W} = \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, i = 1, \dots, N \end{aligned} \quad (14)$$

This is almost the same optimization problem obtained previously, but with the further constraint on the multipliers, α , which now have an upper bound of C .

Indeed, it is noteworthy that the input data of Eq. (14) depend only on the inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$, thus ensuring the immediate possibility of introducing a kernel function, as we will see in the next paragraph.

Kernels in SVM Algorithms

According to the definition of a kernel, all operations that exclusively involve inner products $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, in some feature space, can be implicitly evaluated in the pattern space by an appropriate kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. A nonlinear version of an algorithm can be then obtained, by simply replacing the inner products with the kernel functions. This is called the kernel technique (kernel trick).

It should be clear now that the dual formulation is important for this task. In order to see how this is accomplished for the SVM approach, we can simply re-write the dual Problem's objective, Eq. (14), in the feature space, i.e., in terms of $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. In this way, we have the following expression.

$$\max_x \quad \mathcal{W} = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$$

Thus, we can obtain the final problem formulation, after the kernel substitution, as reported here.

$$\max_x \quad \mathcal{W} = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

Similarly, it can be shown that new examples can be classified by applying the following expression.

$$h(\mathbf{x}) = \text{sgnz.drule}; z.\text{drule}; \text{bigg}(\sum_i \alpha_i y_i (\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle + b) z.\text{drule}; z.\text{drule}; \text{bigg})$$

The new instance $\mathbf{x}$ depends on the data only through inner products in the feature space. Again, the inner product can be replaced with a (nonlinear) kernel function, performing maximum margin classification in the feature space using the kernel from the original pattern space.

Others SVM Extension Algorithms

The previous section has focused on classification problems where classes were labeled within the set $\{-1, +1\}$. Similar arguments can be extended to different typical tasks. Here, we briefly consider the cases of multi-class classification, clustering, and regression.

Multi-Class and One-Versus-All Classifiers

Real observations can belong to more than two groups or classes of objects. In this case, the *multi-class classification* algorithm extends the standard case we have considered so far.

One of the simplest approaches used in the multi-class classification is the so called *one-versus-all approach*, in which the observations belonging to an arbitrary class j have to be discriminated from all other classes, $j = 1, \dots, k$. Here, all the observations from classes other than j are combined to define one common class. The optimal hyperplane separating samples from the class j and the combined class is found using a standard SVM approach, as described above.

Generalizing to multi-class problems is straightforward. Multiple one-versus-all classifiers can be trained, and then the obtained answers can be combined with a general decision function. For example, denoting with $(\mathbf{w}^{(j)}, b^{(j)})$ the optimal hyperplane for class j , and $t^{(j)} = \text{sgn}(\langle \mathbf{w}^{(j)}, \mathbf{x} \rangle + b^{(j)})$, the decision function for j after all the k optimal separating hyperplanes have been found, the final classifier h is given by the following (decision).

$$h(\mathbf{x}) = \text{argmax}_j (t^{(j)}(\mathbf{x})) \quad (15)$$

This approach of assigning class labels using the rule in Eq. (15) is generally called voting. In this case, voting is performed by giving every classifier a vote of size one, and the unknown label is the index of the classifier giving only positive votes. If positive votes are not provided or more than one classifier, t_j , provide positive votes, then no decision about the class label is made.

Different more accurate approaches have been proposed in the literature, for example, binary classifiers can also be represented as a *tree*, such that multi-class classification is obtained by binary classifications at each node of the tree (Platt et al., 1999).

Support Vector Clustering

Differently from classification, clustering is the problem of partitioning the input data into groups in such a way that a particular criterion or relationships between items, belonging to the same group, is satisfied. In this way, the original input patterns are generally organized into a more meaningful form, allowing one to identify homogeneous set of items.

While the literature describes many ways to achieve this goal, Support Vector Clustering (SVC) (Ben-Hur et al., 2001) introduced an interesting and effective technique. First, following the kernel paradigm, inputs are mapped from the original space to a high dimensional feature space using a proper kernel function. Then, in the feature space, the smallest hypersphere enclosing most of the patterns is identified. This sphere, when mapped back to the original space, forms a set of contours that enclose the original samples. It is easy to consider these contours as cluster boundaries. In this way, the points enclosed by each contour are associated by SVC with the same group.

The cluster description algorithm does not differentiate between points that belong to different clusters. At this aim, we can apply a geometric approach that involves the following important observation. Given a pair of patterns belonging to different clusters, each segment that connects them must exit from the sphere in the function space. Such a path contains a point z with $R(z) \geq R$, i.e., its distance from the center of the sphere in the feature space, is greater than the ray of the enclosing sphere. Using this observation, we can define an adjacency matrix, $A_{i,j}$, between pairs of points, $\mathbf{x}_i$ and $\mathbf{x}_j$, whose images lie in or on the sphere in feature space. More formally,

$$A_{i,j} = \begin{cases} 1 & \text{for all } z \text{ on the line segment connecting } \mathbf{x}_i \text{ and } \mathbf{x}_j \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Clusters can be finally defined as the connected components of the graph induced by A .

Unfortunately, for practical purposes, in order to check whether any pair of inputs is within the same cluster, it is necessary to sample a number of points over the line segment that connects these data in the original pattern space. Hence, the time complexity of the cluster labeling part is generally a critical issue. For example, given M sampled points, on the segment connecting all pairs of input, one has to check their connectivity, based on the decision criterion: $R(z_i) \geq R$, for all $i = 1, \dots, M$ and input pairs $\mathbf{x}_i$ and $\mathbf{x}_j$. To

solve this issue different methodologies have been applied, for example in [Pozzi et al. \(2007\)](#) an appropriate simplification (derived from the approximation of a combinatorial optimization problem) on the adjacency matrix A (in [Eq. \(16\)](#)) is given to avoid useless sampling.

Support Vector Regression

Regression is the process of exploring the relationship between (dependent) variables, Y , and one or more (explanatory) variables, X . This relationship is generally provided by fitting a function, $h(\mathbf{x})$, using a sample of observations from X and Y . The function h then can be used to infer the output $\hat{y}=h(\mathbf{x})$ of new unobserved points. The general approach to this problem can be formulated using a loss function $\mathcal{L}(y, h(\mathbf{x}))$, which describes how the estimated $\hat{y}$ deviate from the true values, y , of the examples $(\mathbf{x}, y)$.

Following the most common approach to SV regression (Vapnik's ε -intensive loss function ([Vapnik, 1995](#))), we have

$$\mathcal{L}(y, h(\mathbf{x})) = \begin{cases} 0 & \text{if } |y - h(\mathbf{x})| \leq \varepsilon \\ |y - h(\mathbf{x})| - \varepsilon & \text{otherwise} \end{cases} \quad (17)$$

where $\varepsilon \geq 0$ is a constant that controls the noise tolerance. Using this loss function, we can view the regression problem as the task of finding a band (or tube), of size of at most ε , around the hypothesis function, h . More formally, let us assume to have the following training data.

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} \in \mathcal{X} \times \mathfrak{R},$$

where $\mathcal{X}$ denotes the space of the input patterns. [Eq. \(17\)](#) expresses the fact that, in order to estimate h , we can actually tolerate at most ε deviation from the obtained targets y , for all the training data. In other words, we don't care about errors, as long as they are less than ε , but will consider all deviations larger than this value. Moreover, it is generally required that h should be as flat as possible. If we assume h is a linear function, we have

$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (18)$$

where $\mathbf{x} \in \mathcal{X}$ and $b \in \mathfrak{R}$. In this case, flatness means that one seeks small $\mathbf{w}$. One way to ensure this flatness is to minimize the Euclidean norm $\|\mathbf{w}\|^2$. Thus, we can write the following convex optimization problem.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon, \forall i \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon, \forall i. \end{aligned}$$

Also in this case, we can introduce slack variables, ξ_i, ξ_i^* , to provide a soft margin, as described in the previous paragraphs. These slack variables are zero for points inside the band, and progressively increase for points outside the band, according to the loss function ([Fig. 2](#)). Hence we arrive at the following formulation.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \xi_i, \forall i \\ & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i^*, \forall i. \\ & \xi_i, \xi_i^* \geq 0. \end{aligned}$$

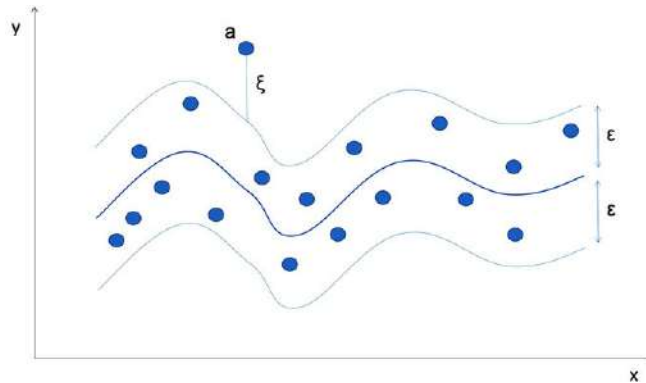


Fig. 2 Nonlinear regression function. The variable ξ measures the cost of a training error, corresponding to patterns outside the band.

The constant C determines the trade-off between the flatness of h and the degree to which deviations larger than ε are tolerated. Similarly to the previous formulations, it is possible to show that the dual form can be described in terms of inner products between the input patterns.

Conclusions

Support vector machines have provided a remarkable contribution, not only to the entire scientific community of machine learning, but also to the most diverse scientific and humanistic areas, where statistical inference is currently applied to many important problems. In fact, the exponential growth of SVM's application has shown the effectiveness of this method.

In this article we have discussed the rationale of such algorithms, their intuitive functional aspects, as well as the reasons why the kernel methods and SVMs have found an extremely natural and effective coexistence. Supported by well founded theoretical arguments, SVMs have definitely equaled and exceeded the accuracy level of the state-of-the-art inference models on numerous prediction tasks. Overall, SVMs are intuitive; By identifying a maximal margin hyperplane that separates positive from negative classes, they classify new items depending on the half-space they are located with respect to the separating hyperplane.

See also: Delta Rule and Backpropagation. Kernel Machines: Introduction

References

- Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V., 2001. Support vector clustering. *Journal of Machine Learning Research* 2, 125–137.
- Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers, in: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM. pp. 144–152.
- Cava, C., Zoppis, I., Gariboldi, M., *et al.*, 2013a. Copy-number alterations for tumor progression inference. In: *Conference on Artificial Intelligence in Medicine in Europe*, Springer Berlin Heidelberg. pp. 104–109.
- Cava, C., Zoppis, I., Mauri, G., *et al.*, 2013b. Combination of gene expression and genome copy number alteration has a prognostic value for breast cancer. In: *Engineering in Medicine and Biology Society (EMBC), 35th Annual International Conference of the IEEE, IEEE*. pp. 608–611.
- Chinello, C., Cazzaniga, M., De Sio, G., *et al.*, 2014. Urinary signatures of renal cell carcinoma investigated by peptidomic approaches. *PloS One* 9, e106684.
- Ding, X., Danielson, M., Ekenberg, L., 2004. Non-linear programming solvers for decision analysis. In: *Operations Research Proceedings 2003*. Springer, pp. 475–482.
- Platt, J.C., Cristianini, N., Shawe-Taylor, J., 1999. Large margin dags for multiclass classification. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*, MIT press. pp. 547–553.
- Pozzi, S., Zoppis, I., Mauri, G., 2007. Support vector clustering of dependencies in microarray data. *Lecture Notes in Engineering and Computer Science*, pp. 244–249.
- Rockafellar, R.T., 1982 (1970) *Convex Analysis*.
- Shawe-Taylor, J., Cristianini, N., 2002. On the generalization of soft margin algorithms. *IEEE Transactions on Information Theory* 48, 2721–2735.
- Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc.
- Vapnik, V.N., 1998. *Statistical Learning Theory*. New York, NY, USA: Wiley.

Further Reading

- Antoniotti, M., Carreras, M., Farinaccio, A., *et al.*, 2010. An application of kernel methods to gene cluster temporal metaanalysis. *Computers & Operations Research* 37, 1361–1368.
- Burges, C.J., 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 121–167.
- Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*. Cambridge university press.
- Cristianini, N., Shawe-Taylor, J., 2000. *An Introduction to Support Vector machines*.
- Deng, N., Tian, Y., Zhang, C., 2012. *Support Vector Machines: Optimization based Theory, Algorithms, and Extensions*. CRC press.
- Scholkopf, B., Smola, A.J., 2001. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press.
- Wang, L., 2005. *Support Vector Machines: Theory and Applications*, vol. 177. Springer Science & Business Media.
- Zoppis, I., Merico, D., Antoniotti, M., Mishra, B., Mauri, G., 2007. Discovering relations among go-annotated clusters by graph kernel methods. In: *International Symposium on Bioinformatics Research and Applications*, Springer. pp. 158–169.

Kernel Machines: Applications

Italo Zoppis and Giancarlo Mauri, University of Milano-Bicocca, Milano, Italy

Riccardo Dondi, University of Bergamo, Bergamo, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Kernel methods have received considerable attention in many scientific communities, mainly due to their capability of working with linear inference models, allowing at the same time to identify nonlinear relationships among input patterns (Smola and Schölkopf, 1998; Shawe-Taylor and Cristianini, 2004; Schölkopf and Burges, 1999). Moreover, the possibility offered by these techniques to work homogeneously with structured data has allowed to cope with different problems, even introducing within the inference process specific domain-knowledge.

The discovery of nonlinear relationships between input patterns and the ability to work with heterogeneous data have always been two important and desirable features to be found in machine learning. At this regard, it is necessary to emphasize that kernel methods not only provided these possibilities in a completely natural way, but they were largely founded on a strong and intuitive mathematics.

For a long time, the most popular approach to solve nonlinear problems was to apply *multi-layer perceptron*: a model, able to approximate any function (Hornik et al., 1989), that can be trained efficiently by using the back-propagation algorithm. The approach followed by the kernel methods is fundamentally different. The strategy is to design procedures, in the feature space, by addressing observations only through inner product operations. Such algorithms are said to employ a kernel, since the pairwise inner products can be computed directly from the original items, using only the so called *kernel function*. In other terms, whether an algorithm expressing operations which exclusively involve inner products in some feature space, can be implicitly done in the input space by using an appropriate kernel function. A nonlinear version of the considered algorithm can be readily obtained by simply replacing the inner products with the kernel function (this is generally referred as *kernel trick*). In this situation, it is not necessary to know the input transformation, instead the kernel trick is enough to ensure the existence of such transformation between the input patterns and the feature space.

In this article, we will show how to implement this technique for simple tasks. Moreover, successful applications in machine learning literature, bioinformatics and pattern recognition will be briefly described.

Implementation

Many machine learning algorithms are generally applied on input which are points in a vector space $\mathfrak{R}^n$. When these algorithms can be expressed in terms of inner products between pairs of input patterns, then they can “implicitly” provide input transformations, generally non-linearly related to the input space. As a consequence, linear algorithms can be easily transformed, into nonlinear algorithms, that in turn, can explore complex relationships between input observations. This operational mode is the fundamental feature of kernel methods.

More technically, such algorithms are said to employ kernel functions, since the pairwise inner products $\langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}}$ between input $\Phi(x_i)$, $\Phi(x_j)$, in the feature space $\mathcal{H}$, can be computed efficiently, directly from the original data items using an appropriate kernel function $k(x_i, x_j)$. In other terms, this function is able to represent, in the original space, the inner product in $\mathcal{H}$. This is formally stated as

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} \quad (1)$$

In this way, a nonlinear version of a given learning algorithm can be obtained by simply replacing the inner products with kernels. We will see now some examples of how this technique can be implemented to provide interesting properties of the input patterns, in the function space, and to solve both classification and regression problems.

Distances in Feature Space

Distances and dissimilarities have been important issues in machine learning and pattern recognition for many years, leading to many different known algorithms and important questions e.g., Pekalska and Duin (2005), Ramakrishnan et al. (2010), Cava et al. (2013), (2014). By applying the kernel trick, it is not difficult, to obtain distances between pair of points in the feature space. For this task, we denote with $\mathcal{H}$ the inner product space, where the features are represented through the mapping $\Phi: \mathcal{X} \rightarrow \mathcal{H}$. It is simple to define a distance $d(x_1, x_2)$ between points in $\mathcal{X}$, as the feature distance between input (images) in $\mathcal{H}$. This can be formulated as follows

$$d(x_1, x_2) = \|\Phi(x_1) - \Phi(x_2)\|_{\mathcal{H}}^2 \quad (2)$$

While it seems necessary to explicitly know the images $\Phi(x_1)$ and $\Phi(x_2)$, before providing the distance d , we can evaluate $d(x_1, x_2)$ by applying the kernel substitution. Hence we have,

$$\begin{aligned} \|\Phi(x_1) - \Phi(x_2)\|_{\mathcal{H}}^2 &= \langle \Phi(x_1) - \Phi(x_2), \Phi(x_1) - \Phi(x_2) \rangle \\ &= \langle \Phi(x_1), \Phi(x_1) \rangle - 2\langle \Phi(x_1), \Phi(x_2) \rangle + \langle \Phi(x_2), \Phi(x_2) \rangle \end{aligned}$$

Using the kernel trick, we have the following expression

$$\|\Phi(x_1) - \Phi(x_2)\|_{\mathcal{H}}^2 = k(x_1, x_1) - 2k(x_1, x_2) + k(x_2, x_2) \quad (3)$$

Thus, as expressed by Eq. (3), the distance in the feature space has been obtained assessing pairs of input in the pattern space through the kernel function.

Norm and Distance from the Center of Mass

As a more general example, we consider now the mean π_S (the center of mass) of a set of input patterns $S = \{x_1, x_2, \dots, x_N\}$. Suppose we want to evaluate π_S in the feature space. First of all, we can write the vector π_S as follows

$$\pi_S = \frac{1}{N} \sum_{i=1}^N \Phi(x_i) \quad (4)$$

Despite the apparent inaccessibility of all points $\Phi(x_i)$, and consequently of the mean π_S , we can easily compute $\|\pi_S\|$, by evaluating the kernel function over the input patterns. Indeed, let us consider the following expressions

$$\begin{aligned} \|\pi_S\|^2 &= \langle \pi_S, \pi_S \rangle \\ &= \left\langle \frac{1}{N} \sum_{i=1}^N \Phi(x_i), \frac{1}{N} \sum_{j=1}^N \Phi(x_j) \right\rangle \\ &= \frac{1}{N^2} \sum_{i,j=1}^N \langle \Phi(x_i), \Phi(x_j) \rangle \\ &= \frac{1}{N^2} \sum_{i,j=1}^N k(x_i, x_j) \end{aligned} \quad (5)$$

Notice that, from Eq. (5), the square of $\|\pi_S\|$ is equal to the average of the entries in the kernel matrix G , being defined as follows

$$G_{i,j} = \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} = k(x_i, x_j) \quad (6)$$

Similarly, we can compute the distance of the image of a given point x from the center of mass π_S . In this case, we obtain the following result

$$\begin{aligned} \|\Phi(x) - \pi_S\|^2 &= \langle \Phi(x), \Phi(x) \rangle + \langle \pi_S, \pi_S \rangle - 2\langle \Phi(x), \pi_S \rangle \\ &= k(x, x) + \frac{1}{N^2} \sum_{i,j=1}^N k(x_i, x_j) - \frac{2}{N} \sum_{i=1}^N k(x, x_i) \end{aligned}$$

Binary Classification Problems

Let us consider now a more complete case: the *Binary Classification* problem. Classification is a typical problem in machine learning. Given a training set of examples $\mathcal{T}$, it is asked to construct a specific inference model $\mathcal{M}$, using elements from $\mathcal{T}$. The target is to find the membership class of new items, which have not been used for the model construction. In other words, since any model has generally some free parameters, we can use elements from $\mathcal{T}$ – following particular mathematical techniques, to estimate such values. In binary classification, we assume that the label space $\mathcal{Y}$ contains only the two elements, $+1$ and -1 , while the training set consists of labeled examples of the two classes. When new items are given, the model should be able to answer, which of the two classes a new item belongs.

For classification, we will apply a simple argumentation to associate the class membership for new items. The class (label) will be given, taking into consideration the distance from the center of mass, as discussed in the previous section, i.e., we will assign a new point to the class whose distance is closest. More formally, let $\mathcal{T}$ be an arbitrary training set defined as follows

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} \in \mathcal{X} \times \mathcal{Y} \quad (7)$$

Let us assume to work directly in the feature space $\mathcal{H}$, and also assume that our pattern space of reference is $\mathfrak{R}^n$. The sentence, “working in the feature space” exactly means that we will refer to $\Phi(\mathbf{x}) \in \mathcal{H}$, instead of $\mathbf{x} \in \mathfrak{R}^n$, even if one cannot explicitly perceive its components. The reference to $\Phi(\mathbf{x})$ is simply done in order to apply the algorithmic ideas over the new space of representation, in our case, the classification algorithm. As the main intuition, on which our classification is based, is to assign, unseen patterns to classes with closer distance, we will apply this rule in the feature space $\mathcal{H}$. In this way, we first compute the distance of the observed examples from the center of mass — for each membership class, then assign a new point to the class whose distance is closest.

Let S_+ and S_- be the set of positive and negative examples from the training $\mathcal{T}$. Moreover, let $d_+(\mathbf{x}) = \|\Phi(\mathbf{x}) - \Phi_{S_+}\|$ be the distance, in the feature space, of a test point $\mathbf{x}$ from the center of mass Φ_{S_+} of S_+ , and $d_-(\mathbf{x}) = \|\Phi(\mathbf{x}) - \Phi_{S_-}\|$ be the distance of a point $\mathbf{x}$ from the center of mass Φ_{S_-} of S_- . The simple rule we apply here classifies a new item, following the decision function defined as

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } d_-(\mathbf{x}) > d_+(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases} \quad (8)$$

By expressing $h(\mathbf{x})$ in terms of the sign function, we can obtain the following,

$$\begin{aligned} h(\mathbf{x}) &= \text{sgn}(\|\Phi(\mathbf{x}) - \Phi_{S_-}\|^2 - \|\Phi(\mathbf{x}) - \Phi_{S_+}\|^2) \\ &= \text{sgn}(-k(\mathbf{x}, \mathbf{x}) - \frac{1}{m_+^2} \sum_{i,j=1}^{m_+} k(\mathbf{x}_i, \mathbf{x}_j) + \frac{2}{m_+} \sum_{i=1}^{m_+} k(\mathbf{x}, \mathbf{x}_i) \\ &\quad + k(\mathbf{x}, \mathbf{x}) + \frac{1}{m_-^2} \sum_{i,j=m_++1}^{m_++m_-} k(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{m_-} \sum_{i=m_++1}^{m_++m_-} k(\mathbf{x}, \mathbf{x}_i)) \\ &= \text{sgn}(\frac{1}{m_+} \sum_{i=1}^{m_+} k(\mathbf{x}, \mathbf{x}_i) - \frac{1}{m_-} \sum_{i=m_++1}^{m_++m_-} k(\mathbf{x}, \mathbf{x}_i) - b) \end{aligned} \quad (9)$$

where we have assumed to assign indexes ranging from 1 to m_+ to positive examples and indexes from $m_+ + 1$ to $m = m_+ + m_-$ to negative examples. Moreover, we denoted with b a constant being half of the difference between the average entry of the positive examples kernel matrix and the average entry of the negative examples kernel matrix. Finally notice that in Eq. (9), we applied the kernel trick to assess input data only through the kernel function.

Kernel Regression

Let us consider now the problem of finding a real-valued linear function

$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}'\mathbf{x} \quad (10)$$

that fits as better as possible a given set of points in $X \subseteq \mathcal{R}^n$. The points $\mathbf{x}_i \in X$ are generally provided by the set of observed data

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\} \quad (11)$$

where the corresponding labels y_i are elements of a set $Y \subseteq \mathcal{R}$. This is a typical regression problem, where it is asked to explore relationships between a dependent variable Y and one (or more) explanatory variable X . The function $h(\mathbf{x})$ in Eq. (10) represents one of the simplest relationships that can be found between X and Y . It expresses a linear function of the input patterns, that match the associated target values y_i . Using this function one can easily define a new pattern function f which, in turn, should assume ideally a value equal to 0. In this way, we can define f as follows

$$f((\mathbf{x}, y)) = |y - h(\mathbf{x})| \quad (12)$$

In general Eq. (12) evaluates an error $|\xi| = f((\mathbf{x}, y)) = |y - h(\mathbf{x})|$ of the linear function h on the particular training example in $(\mathbf{x}, y) \in \mathcal{T}$. Thus the problem can be formulated as the one of finding a function f for which all training errors are as small as possible. The standard classical approach is to use the sum of the squares of errors as measure of the discrepancy between the training data and the particular chosen function. In other terms, we could also say that, we consider here the problem of seeking the parameters $\mathbf{w}$ for which the following loss function is minimized

$$\mathcal{L}(h, \mathcal{T}) = \mathcal{L}(\mathbf{w}, \mathcal{T}) = \sum_{i=1}^N (y_i - h(\mathbf{x}_i))^2 = \sum_{i=1}^N \xi_i^2 = \sum_{i=1}^N \mathcal{L}((\mathbf{x}_i, y_i), h)$$

where we have used the same notation $\mathcal{L}((\mathbf{x}_i, y_i), h) = \xi_i^2$ to represent the squared error of h for the example $(\mathbf{x}_i, y_i)$ and $\mathcal{L}(h, \mathcal{T})$ to denote the collective loss on the training set $\mathcal{T}$. This problem, well-known as least squares approximation, is widely applied in many different discipline. Using the notation above, the vector of output discrepancies can be written as $\xi = \mathbf{y} - \mathbf{X}\mathbf{w}$. Hence, we can also write the loss function as follows

$$L(\mathbf{w}, \mathcal{T}) = \|\xi\|_2^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w})$$

The optimal $\mathbf{w}$ can be found by differentiating the loss function with respect to $\mathbf{w}$ and setting them equal to zero (vector), i.e.

$$\frac{\partial L(\mathbf{w}, \mathcal{T})}{\partial \mathbf{w}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\mathbf{w} = 0 \quad (13)$$

In this way, we get the following equation

$$\mathbf{X}'\mathbf{X}\mathbf{w} = \mathbf{X}'\mathbf{y} \quad (14)$$

If the inverse of $\mathbf{X}'\mathbf{X}$ exists, then we can write the solution of the least squares problem as follows

$$\mathbf{w} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \quad (15)$$

As usually, the inferred output on a new input pattern can be obtained applying the prediction function $h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$. Moreover, notice that if the inverse of $\mathbf{X}'\mathbf{X}$ exists, we can express $\mathbf{w}$ in the following way

$$\mathbf{w} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-2}\mathbf{X}'\mathbf{y} = \mathbf{X}'\boldsymbol{\alpha} \quad (16)$$

obtaining a linear combination of the training points, as follows

$$\mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{x}_i \quad (17)$$

Sometime, we can improve the generalization of the algorithm by adding a “penalty” term. In this case, we have the so called *Ridge Regression* that corresponds to solving the following optimization problem

$$\min_{\mathbf{w}} \mathcal{L}_{\lambda}(\mathbf{w}, T) = \min_{\mathbf{w}} (\lambda \|\mathbf{w}\|^2 + \sum_{i=1}^N (y_i - h(\mathbf{x}_i))^2) \quad (18)$$

where $\lambda > 0$ controls the “trade-off between norm and loss.

Similarly to the previous case, by taking the derivative of the loss function with respect to the parameters (13), we get the following equation

$$\mathbf{X}'\mathbf{X}\mathbf{w} + \lambda\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda I_N)\mathbf{w} = \mathbf{X}'\mathbf{y} \quad (19)$$

where I_N is a $N \times N$ identity matrix. In this case, the matrix $(\mathbf{X}'\mathbf{X} + \lambda I_N)$ is always invertible for $\lambda > 0$. Therefore, we have the solution

$$\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda I_N)^{-1}\mathbf{X}'\mathbf{y} \quad (20)$$

In order to solve this equation for $\mathbf{w}$ we have to solve a system of linear equations with N unknowns variables and N equations. We therefore obtain the final decision function expressed as follows

$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{y}'\mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda I_N)^{-1}\mathbf{x} \quad (21)$$

Equivalently, we can rewrite Eq. (19) in terms of $\mathbf{w}$. Thus we obtain

$$\mathbf{w} = \lambda^{-1}\mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{X}'\boldsymbol{\alpha} \quad (22)$$

Again, $\mathbf{w}$ can be written as a linear combination of the training data

$$\mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{x}_i \quad (23)$$

with $\boldsymbol{\alpha} = \lambda^{-1}(\mathbf{y} - \mathbf{X}\mathbf{w})$. Hence, we have

$$\begin{aligned} \boldsymbol{\alpha} &= \lambda^{-1}(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &\Rightarrow \lambda\boldsymbol{\alpha} = (\mathbf{y} - \mathbf{X}\mathbf{X}'\boldsymbol{\alpha}) \\ &\Rightarrow (\mathbf{X}\mathbf{X}' + \lambda I_N)\boldsymbol{\alpha} = \mathbf{y} \\ &\Rightarrow \boldsymbol{\alpha} = (\mathbf{G} + \lambda I_N)^{-1}\mathbf{y} \end{aligned} \quad (24)$$

where $\mathbf{G} = \mathbf{X}\mathbf{X}'$, or equivalently $G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Also in this case we can write the predicted output of a new points, obtaining the following result

$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \langle \sum_{i=1}^N \alpha_i \mathbf{x}_i, \mathbf{x} \rangle = \sum_{i=1}^N \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle \quad (25)$$

Notice that, we described two different methods for solving the Ridge regression optimization problem. In the first Eq. (20) the weight vector is computed explicitly (primal solution), while in the second case, reported in Eq. (24), we obtained the solution as a linear combination of the training examples (dual solution), in this last case, the parameters α are known as dual variables.

Finally, we observe an important fact. In the dual solution, the information provided by the training examples is given through the inner products between pairs of training points in the (Gram) matrix $\mathbf{G} = \mathbf{X}\mathbf{X}'$. Similarly, the information about a novel example $\mathbf{x}$ used by the predictive function is again provided as inner products between training points and the new example $\mathbf{x}$. Therefore, like the other examples reported previously, also in the case, the algorithm can be written in a form that only requires inner products between observations.

Kernels and Applications

The most typical and probably mentioned kernel-based algorithm is known as *Support Vector Machine* (Vapnik, 1995, 1998; Boser *et al.*, 1992). Many others algorithmic implementations have been developed and extended within the machine learning community since the introduction of kernel methods. Just to name a few, it is worthwhile to refer to the Kernel Fisher Discriminant Analysis, Kernel Principal Component Analysis and Kernel Independent Component Analysis. Such algorithms have been “kernelized”, in the sense that, starting from the original formulation, they have been adapted (as described above), to assess input

patterns only through inner product operations, in the feature space. This paradigm has been successfully applied to different application areas, providing accurate performances and relative robust results.

The exponential growth of the different applications and the many extensions of these techniques do not obviously allow to draw a full extensive characterization of the current literature on kernels. In this article, we focus on two very important domains: bioinformatics and pattern recognition.

Kernel Methods in Bioinformatics

The huge volume of data produced by biotechnologies (e.g., microarray) has required new methods for extracting and analyzing meaningful information. A multitude of works using the kernel approach emerged, especially for tasks concerning DNA/RNA sequences (Hua and Sun, 2001; Sonnenburg *et al.*, 2002, 2005, 2006; Sakakibara *et al.*, 2007; Sato *et al.*, 2008) and proteins (Tsuda *et al.*, 2002; Tsuda and Noble, 2004a; Ben-Hur and Brutlag, 2003; Liao and Noble, 2003).

The main reason of success of the kernel paradigm in this context is certainly due to both the ability of working homogeneously with many different structures and to the effectiveness with which non-linearity can be introduced into linear algorithms. For example, considering DNA-sequences, the problem is to deal with a variable length which make it hard to represent them e.g., as vectors. Since any valid kernel give rise to a positive semi-definite matrix, the representation of data with that matrix (being independent of the nature or the structure of the data to be analyzed) makes kernels a potential ideal technique for (heterogeneous) data integration and representation. Such issues are investigated e.g., in Daemen *et al.* (2007) where efficient integration of clinical and microarray data are focused.

Among the others omics science, Genomics and Proteomics are currently two successful applications where the requirements of homogeneous representation and nonlinear relationships identification are important to accomplish tasks such as classification, clustering and feature selection. In a sense, these tasks formalize and unify many targets problems of both these omics disciplines.

For example, in genomics, the *gene finding* problem is definitely one of the issues that has been more focused in literature. In particular, gene selection from microarray data is usually formalized as a feature selection task, where the relevant features (i.e., relevant for an accurate prediction) are extremely important for understanding the underlying disease or to get deeper knowledge of the underlying biological process. In this case, the knowledge discovery process, tries either to recursively eliminate irrelevant features or iteratively add most informative ones (Guyon *et al.*, 2008; Gratkowski *et al.*, 2009; Weston *et al.*, 2000).

While the literature on feature selection is older and goes behind the field of the kernel paradigm, several interesting development with kernels have been proposed, explicitly motivated by the problem of gene finding. For example in Su *et al.* (2003), it is proposed to evaluate the predictive power of each single gene for a given classification task by the value of the functional minimized by a SVM model, trained to classify samples from the expression of only the single gene of interest. This is the typical criterion that can be used to rank genes and then select only those with important predictive power. This procedure belongs to the so-called filter approach relevant to the feature selection problem. In this specific case, some score criterion (e.g., obtained using the SVM performances) measures the relevance of each feature, and only higher score features, according to this criterion, are kept.

A second general strategy, for feature selection, is the wrapper approach, where selection procedures alternate with the classifier's training phase. The widely used recursive feature elimination (RFE) procedure of Guyon *et al.* (2002), which iteratively selects smaller sets of genes and trains SVMs, follows this strategy. Starting from the full set of genes, a SVM is trained and the genes with smallest weight values are eliminated. The procedure is then repeated iteratively, using the current set of remaining genes. This process stops when either a specific criterion or a given number of genes is reached.

A third strategy for marker (i.e., gene) selection with the feature selection problem, is known as the embedded approach. In this case, the two phases i.e., learning of a classifier and the selection of features, are combined in a single step. Many kernel methods following this strategy have been implemented, for example in Krishnapuram *et al.* (2004) where, roughly speaking, a variant of SVM with a Bayesian formulation is proposed.

Kernel methods have been successfully applied also to splice site recognition. In this case, the prediction task is to discriminate between sequences that do contain a true splice site versus sequences with a decoy splice site (Sonnenburg *et al.*, 2002). Generally for these problem, particular kernels are employed (e.g., the so-called weighted degree kernel).

Structuring genes in groups (Eisen *et al.*, 1998; Antoniotti *et al.*, 2010; Zoppis *et al.*, 2007) is another important task which is generally formulated as a clustering problem. This task is possibly useful to gain insight into biological and regulatory processes. In Pozzi *et al.* (2007) an application of SVMs to cluster homogeneous features, such as pairs of gene-to-gene interactions is proposed. Specifically, a Support Vector Clustering (SVC) is applied (i.e. a novelty detection algorithm), to provide groups of similarly interacting pairs of genes with respect to some measure (i.e. kernel function) of their activation/inhibition relationships.

Kernel approach in proteomics mainly involved protein function prediction from data sources other than sequence or structure. Some representative studies in this direction can be found in Vert (2002), where a kernel on trees is introduced for function prediction from phylogenetic profiles of proteins. Tsuda and Noble (2004b), reported an inference task to predict the function of unannotated proteins in protein interaction or metabolomic networks. Successful application of SVMs in proteomics also include protein homology detection to determine the structural and functional properties of new protein sequences (Jaakkola *et al.*, 2000). Determination of these properties is achieved by relating new sequences to proteins with known structural features.

Finally, mass spectrometry and peptide identification represent a promising area of research in clinical analysis. They are primarily concerned with measuring the relative intensity (i.e., signals) of many protein/peptide molecules associated with their mass-to-charge ratios. These measurements provide a huge amount of information which requires adequate tools to be

interpreted. In Galli *et al.* (2016) is reported an accurate classification of thyroid bioptic specimens that can be obtained through the application of Support Vector Machines on MALDI-MSI data, together with a particular “wrapper” feature selection algorithm.

Kernel Methods and Pattern Recognition

Pattern Recognition is a mature and fast developing field, which forms the core of many other disciplines such as computer vision, image processing, clinical diagnostics, person identification, text and document analysis. It is closely related to machine learning, and also finds applications in fast emerging areas such as biometrics, bioinformatics, multimedia data analysis and most recently data science.

Within this context, object recognition (and detection) plays a key role in both computer vision and robotics. The task is difficult partly because images are in high-dimensional space and can change with viewpoint, while the objects themselves may be deformable, leading to large variation. The core of building object recognition systems is to extract meaningful representations (features) from high dimensional patterns such as images, videos, or even 3D clouds of points. For examples, moving people or traffic situation e.g., for surveillance or traffic control (Gao *et al.*, 2001). Here the tasks provided by kernel approaches are truly vast. Significant examples can be found in Nakajima *et al.* (2000), where people recognition and pose estimation were implemented as a multi-class classification problem. Many attention has been focused even with detailed situation and difficulties deriving e.g. from noise and occlusion (Pittore *et al.*, 1999).

Many other tasks and object recognition system have been implemented, just to cite a few in Li *et al.* (2000b) is provided a radar target recognition and in Pontil and Verri (1998) Roobaert and Van Hulle (1999) a more general SVM base 3D object recognition system are detailed.

A special topic of pattern recognition is the face recognition. This is also one of the most popular problem in biometrics, and it is of fundamental importance in the construction of security, control and verification systems. The main difficulties arising with these applications, are given by the needs of distinguishing different people who have roughly same facial features. Moreover complications arise due to variations in pose, illumination and facial expression (Wang *et al.*, 2002). Kernels and in particular SVMs have been widely applied in this context since the '90 s (Osuna *et al.*, 1997), collecting many successes both in the specific topics of face detection (Li *et al.*, 2000a; Ai *et al.*, 2001; Ng and Gong, 1999, 2002; Huang *et al.*, 1998) and face authentications (Tefas *et al.*, 2001; Jonsson *et al.*, 2002). While traditionally, face representation in the conventional Eigenface and Fisherface approaches is based on second order statistics of the image data-set, i.e., covariance matrix, and does not use high order statistical properties, much of the important information may be contained in these high order statistical relationships among image's pixels. Using the kernel tricks conventional methods have been extend to feature spaces, where it is possible to extract nonlinear features among more pixels. From these relevant works new specific kernel functions and concepts have become fundamental within the scientific community, i.e., Kernel Eigenface and Kernel Fisher face methods (Yang, 2002). Also in these cases, kernel approaches have shown to be effective and powerful for the development of face recognition platforms.

Conclusions

In this article, we have described how the kernel approach can be easily implemented and applied to simple machine learning problems. We have also discussed how this paradigm has offered new opportunities to the learning machine community. Kernel methods do not only provide two major needs driven by the Information and Communication Technologies i.e., the ability of working homogeneously with different structures and the effectively to introduce non-linearity within prediction models, but they have also given a new important capability: the one of transforming old procedures in new powerful methods for extracting and inferring meaningful information. In fact, many inference algorithms can be designed in such a way to access training data only through inner product operations between pair of input patterns. Such algorithms are said to employ kernels (or kernel function), since the pairwise inner products can be even computed efficiently directly from the original data items, using the kernel function. In this way, we could definitely say that, whether an old or a new methods expressing operations which exclusively involve inner products (in the feature space) can be implicitly done in the input space by an appropriate kernel. This is the way to design a kernel algorithm.

The exponential growth of different applications in many disciplines has shown the effectiveness of this paradigm. Here we focused on two examples of such disciplines: bioinformatics and patter recognition. Many other significant question and interesting challenges are emerging nowadays in the scientific community, mainly due to support limitations of the kernel approach. Just to cite a few of these challenges, the relationship between kernel methods (as well as the whole context of machine learning) and the new emerging learning paradigms on deep architectures and cloud technologies. In the first case (Hinton *et al.*, 2006; Deng *et al.*, 2014), researchers, in particular from neural network community, have advanced several reasons to design the so called *deep learning strategies*. Such paradigm aims at learning useful features by stacking successive layers of transformations. Each layer is issued from a nonlinear projection of output of the previous layer using kernel method. This approach should benefit the expressiveness of distributed hierarchical representation, and the ease of combining supervised and unsupervised methods. Experiments have shown promising results of this methodology in several cases (Collobert and Weston, 2008; Bengio *et al.*, 2009).

In the second case, distributed machine learning has become a fundamental requirement in all big data context. Most of machine learning algorithms have problems with computational complexity of training phase with large scale learning datasets.

Applications of classification algorithms for large volume data-set are computationally expensive to process. The computation time and storage space of kernel algorithms are mainly determined by large scale kernel matrix, which must transmit all the information contained within the pairwise similarity scores evaluated using the applied kernel. In this way, kernel is both the core and crucial information bottleneck of the kernel approach. Different solutions to overcome very large scale classification problems have been provided, for example using distributed SVMs by filtering out non-support vectors elements to optimize the final maximum margin hyperplane over the combined data sets (Graf *et al.*, 2004), or by exchanging in different way the support vectors among the system units (Ruping, 2001; Syed *et al.*, 1999; Lu *et al.*, 2008).

See also: Delta Rule and Backpropagation. Kernel Machines: Introduction

References

- Ai, H., Liang, L., Xu, G., 2001. Face detection based on template matching and support vector machines. In: Proceedings 2001 International Conference on Image Processing, IEEE. pp. 1006–1009.
- Antonioti, M., Carreras, M., Farinaccio, A., *et al.*, 2010. An application of kernel methods to gene cluster temporal meta-analysis. *Computers & Operations Research* 37, 1361–1368.
- Bengio, Y., *et al.*, 2009. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning* 2, 1–127.
- Ben-Hur, A., Brutlag, D., 2003. Remote homology detection: A motif based approach. *Bioinformatics* 19, i26–i33.
- Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory, ACM. pp. 144–152.
- Cava, C., Zoppis, I., Gariboldi, M., Castiglioni, I., Mauri, G., Antonioti, M., 2013. Copy-number alterations for tumor progression inference. In: Conference on Artificial Intelligence in Medicine in Europe. Berlin, Heidelberg: Springer. pp. 104–109.
- Cava, C., Zoppis, I., Gariboldi, M., *et al.*, 2014. Combined analysis of chromosomal instabilities and gene expression for colon cancer progression inference. *Journal of Clinical Bioinformatics* 4, 2.
- Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, ACM. pp. 160–167.
- Daemen, A., Gevaert, O., De Moor, B., 2007. Integration of clinical and mi-croarray data with kernel methods. In: Proceedings of the 29th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society, 2007. EMBS 2007. IEEE. pp. 5411–5415.
- Deng, L., Yu, D., *et al.*, 2014. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing* 7, 197–387.
- Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95, 14863–14868.
- M. Galli, M., Zoppis, I., De Sio, G., *et al.* A support vector machine classification of thyroid bioptic specimens using MALDI-MSI data In: *Advances in Bioinformatics* 2016.
- Gao, D., Zhou, J., Xin, L., 2001. Svm-based detection of moving vehicles for automatic traffic monitoring. In: Proceedings. 2001 IEEE Intelligent Transportation Systems, 2001. IEEE. pp. 745–749.
- Graf, H.P., Cosatto, E., Bottou, L., Durdanovic, I., Vapnik, V., 2004. Parallel support vector machines: The cascade svm. In: NIPS.
- Gratkowski, S., Brykalski, A., Sikora, R., Wilinski, A., Osowski, S., 2009. Gene selection for cancer classification. *COMPEL – The International journal for Computation and Mathematics in Electrical and Electronic Engineering* 28, 231–241.
- Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A., 2008. *Feature Extraction: Foundations and Applications*, 207. Springer.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422.
- Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 1527–1554.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359–366.
- Huang, J., Shao, X., Wechsler, H., 1998. Face pose discrimination using support vector machines (svm). In: Proceedings of the 14th International Conference on Pattern Recognition, IEEE. pp. 154–156.
- Hua, S., Sun, Z., 2001. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology* 308, 397–407.
- Jaakkola, T., Diekhans, M., Haussler, D., 2000. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology* 7, 95–114.
- Jonsson, K., Kittler, J., Li, Y., Matas, J., 2002. Support vector machines for face authentication. *Image and Vision Computing* 20, 369–375.
- Krishnapuram, B., Carin, L., Hartemink, A.J., 2004. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *Journal of Computational Biology* 11, 227–242.
- Liao, L., Noble, W.S., 2003. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology* 10, 857–868.
- Li, Y., Gong, S., Sherrah, J., Liddell, H., 2000a. Multi-view face detection using support vector machines and eigenspace modelling. In: Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. IEEE. pp. 241–244.
- Li, Z., Weida, Z., Licheng, J., 2000b. Radar target recognition based on support vector machine. In: Proceedings of the 5th International Conference on Signal Processing Proceedings, 2000. WCCC-ICSP 2000. IEEE. pp. 1453–1456.
- Lu, Y., Roychowdhury, V., Vandenbergh, L., 2008. Distributed parallel support vector machines in strongly connected networks. *IEEE Transactions on Neural Networks* 19, 1167–1178.
- Nakajima, C., Pontil, M., Poggio, T., 2000. People recognition and pose estimation in image sequences. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000. IJCNN 2000. IEEE. pp. 189–194.
- Ng, J., Gong, S., 1999. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In: Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. IEEE. pp. 14–21.
- Ng, J., Gong, S., 2002. Composite support vector machines for detection of faces across views and pose estimation. *Image and Vision Computing* 20, 359–368.
- Osuna, E., Freund, R., Girosi, F., 1997. Training support vector machines: an application to face detection. In: IEEE Computer Society Conference on Computer vision and pattern recognition, 1997. Proceedings. IEEE. pp. 130–136.
- Pekalska, E., Duin, R.P., 2005. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, 64. World scientific.
- Pittore, M., Basso, C., Verri, A., 1999. Representing and recognizing visual dynamic events with support vector machines. In: Proceedings of the International Conference on Image Analysis and Processing. IEEE. pp. 18–23.
- Pontil, M., Verri, A., 1998. Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 637–646.

- Pozzi, S., Zoppis, I., Mauri, G., 2007. Support vector clustering of dependencies in microarray data. *Lecture Notes in Engineering and Computer Science*. 244–249.
- Ramakrishnan, N., Tadepalli, S., Watson, L.T., *et al.*, 2010. Reverse engineering dynamic temporal models of biological processes and their relationships. *Proceedings of the National Academy of Sciences* 107, 12511–12516.
- Roobaert, D., Van Hulle, M.M., 1999. View-based 3d object recognition with support vector machines. In: *Proceedings of the 1999 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing IX*. IEEE. pp. 77–84.
- Ruping, S., 2001. Incremental learning with support vector machines. In: *Proceedings IEEE International Conference on Data Mining, 2001. ICDM 2001*. IEEE. pp. 641–642.
- Sakakibara, Y., Popendorf, K., Ogawa, N., Asai, K., Sato, K., 2007. Stem kernels for rna sequence analyses. *Journal of Bioinformatics and Computational Biology* 5, 1103–1122.
- Sato, K., Mituyama, T., Asai, K., Sakakibara, Y., 2008. Directed acyclic graph kernels for structural rna analysis. *BMC Bioinformatics* 9, 318.
- Schölkopf, B., Burges, C.J., 1999. *Advances in Kernel Methods: Support Vector Learning*. MIT press.
- Shawe-Taylor, J., Cristianini, N., 2004. *Kernel Methods for Pattern Analysis*. Cambridge university press.
- Smola, A.J., Schölkopf, B., 1998. *Learning with Kernels*. Citeseer.
- Sonnenburg, S., Rautsch, G., Jagota, A., Müller, K.R., 2002. New Methods for Splice Site Recognition. Berlin Heidelberg: Springer, pp. 329–336.
- Sonnenburg, S., Rautsch, G., Schuaf, C., 2005. Learning interpretable svms for biological sequence classification. In: *Annual International Conference on Research in Computational Molecular Biology*, Springer. pp. 389–407.
- Sonnenburg, S., Zien, A., Ratsch, G., 2006. Arts: Accurate recognition of transcription starts in human. *Bioinformatics* 22, e472–e480.
- Su, Y., Murali, T., Pavlovic, V., Schaffer, M., Kasif, S., 2003. Rankgene: Identification of diagnostic genes based on expression data. *Bioinformatics* 19, 1578–1579.
- Syed, N.A., Huan, S., Kah, L., Sung, K., 1999. Incremental learning with support vector machines.
- Tefas, A., Kotropoulos, C., Pitas, I., 2001. Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 735–746.
- Tsuda, K., Kin, T., Asai, K., 2002. Marginalized kernels for biological sequences. *Bioinformatics* 18, S268–S275.
- Tsuda, K., Noble, W.S., 2004a. Learning kernels from biological networks by maximizing entropy. *Bioinformatics* 20, i326–i333.
- Tsuda, K., Noble, W.S., 2004b. Learning kernels from biological networks by maximizing entropy. *ISMB/ECCB (Supplement of Bioinformatics)*. pp. 326–333.
- Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.
- Vapnik, V.N., 1998. *Statistical Learning Theory*. New York, NY, USA: Wiley.
- Vert, J.P., 2002. A tree kernel to analyse phylogenetic profiles. *Bioinformatics* 18, S276–S284.
- Wang, Y., Chua, C.S., Ho, Y.K., 2002. Facial feature detection and face recognition from 2d and 3d images. *Pattern Recognition Letters* 23, 1191–1202.
- Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V., 2000. Feature selection for svms. In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. MIT Press. pp. 647–653.
- Yang, M.H., 2002. Face recognition using kernel methods. *Advances in Neural Information Processing Systems* 2, 1457–1464.
- Zoppis, I., Merico, D., Antonioti, M., Mishra, B., Mauri, G., 2007. Discovering relations among go-annotated clusters by graph kernel methods. In: *International Symposium on Bioinformatics Research and Applications*. Springer. pp. 158–169.

Further Reading

- Borgwardt, K.M., 2011. Kernel methods in bioinformatics. *Handbook of Statistical Bioinformatics*. Springer. pp. 317–334.
- Burges, C.J., 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 121–167.
- Cristianini, N., Campbell, C., Burges, C., 2002. Editorial: Kernel methods: Current research and future directions. *Machine Learning* 46, 5–9.
- Gärtner, T., 2003. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter* 5, 49–58.
- Schölkopf, B., Smola, A.J., 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press.
- Schölkopf, B., Smola, A., Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10, 1299–1319.
- Schölkopf, B., Tsuda, K., Vert, J.P., 2004. *Kernel Methods in Computational Biology*. MIT press.

Multiple Learners Combination: Introduction

Chiara Zucco, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

With the term ensemble we usually refer to a group working for an overall result. Therefore, generally speaking, the idea behind the ensemble learning methods is to combine multiple learners in a suitable way that can improve the accuracy of a decision-making processes.

The idea underlying this methodology has many similarities with what happens in everyday life. Let's think, for example, to democratic electoral systems or medical teams formed by specialists from various fields, or to the requests for an external expertise or to a decision committee (Polikar, 2012).

There are innumerable cases where more opinions are heard to make a better decision and that decision is then made by taking into account and combining the opinions of those experts where, in general, with the term expert we mean a decision-maker who has a lot of knowledge or ability in a certain area. From a computational point of view, an accepted definition of expert can be a model induced by an algorithm that has a good number of past positive records in that particular area, at least more than the random guessing model. In particular, the term "weak" expert/learner/classifier is usually referred to learners having an accuracy that is slightly better than the random guessing algorithm, while a "strong" expert/learner/classifier has a high number of past positive records, so high accuracy.

In real life, if there was an expert of every area of knowledge, or "oracle", responding to any question in a correct way, we did not need to involve more experts, but it doesn't exist. Similarly, there is no single algorithm that induces the most accurate learning model in any domain or, in other words, there are no models that do not have negative records at least in one domain. This is basically expressed in a theorem, known as *No free lunch theorems for optimization* that, in some way, constitute the rationale behind the ensemble methods (Wolpert and Macready, 1997; Wolpert, 2012; Ho and Pepyne, 2002). Instead of looking for the learning model that is the most accurate for that problem, the ensemble learning approach is to combine weak (or base) learner to increase the accuracy of the combined model by compensating for the weaknesses of one learner by the strengths of the others.

One of the theoretical foundations underlying the combined methods, can be found in a result belonging to the political science field, known as the Condorcet's jury Theorem and dating back to 1785 (De Condorcet *et al.*, 2014).

The first approaches to ensemble approaches within the ML date back to the 1960s, in particular one of the first that presented an idealized architectures in which multiple models were built and their outputs were combined with a "demon decision", is the pandemonium paradigm presented by Selfridge (1958).

However, the major contributions regarding both the construction of ensemble models and the methodologies to combine multiple learners, arose in the '90s.

In particular, in 1990 Schapire presented boosting (Schapire, 1990) and in 1996 AdaBoost (Freund *et al.*, 1996) in 1992 Wolpert introduced the stacked generalization scheme, later called stacking (Wolpert, 1992). In Breiman (1996) introduced for the first time another lucky ensemble learning techniques known as Bagging, and in 2000 its most known variant, i.e., Random forest (Breiman, 2001).

The ensemble methods have gained a lot of popularity in the last twenty years because the results have shown both theoretically and practically that combining more weak learners over-emphasize the patterns induced by a single algorithm (also known as base or weak learners) (Polikar, 2012). Intuitively, however, it is evident that combining multiple models has the price of greater computational complexity. For example, if we think of a classification problem, it's easy to understand that training and testing of multiple algorithms requires greater spending on time and space than training and testing of just one algorithm. So it makes sense to wonder what hypotheses should be required for performance gains to be significant compared with the increase in complexity (Alpaydin, 2014).

In this article we will try to answer essentially two questions:

- How to choose the learner base so that the resulting model achieves better overall performance;
- How to combine the outputs of individual learners.

In particular, the article is organized as follows: Section "Motivation and Fundamentals" discusses in detail the theoretical motivation explaining why these methods should work and presents a significant example in which the majority vote of three classifiers does not induce a more accurate model. Following, will be discussed an example presenting the main issues when combining multiple classifiers; Section "Building an Ensemble Learning System" presents the most commonly used methods to combine multiple models. Finally, Section "Conclusion" concludes the article.

Motivation and Fundamentals

Among other theoretical reasons for supporting ensemble methods, Dietterich *et al.* (2000) gives three fundamental reasons: a *statistical* reason, a *computational* reason, and a third *representative* reason.

The *statistical* reason starts from the assumption that, indicating with L^* the optimal model for a given problem, because of lack of sufficient data, the learning algorithm can find multiple models having the same accuracy. The ensemble approach avoids the problem of choosing between these models by replacing an “average” aggregate model that may be closer to L^* , or have better performances on new data than the single models.

But even if the dataset has a sufficient number of data, it is demonstrated that for some machine learning algorithms the search for the optimal model L^* can be a computationally difficult task (e.g., NP-hard), as many algorithms rely on techniques of local optimization, that can converge towards a local optima. For this reason, a second computational motivation arises from the possibility of building an ensemble in which the local search starts from different points, in order to achieve a better approximation.

The latest motivation, the most subtle, is related to the fact that the optimal solution L^* for the real problem may not be represented by any model in the space of solutions. In this case, the multiple combination of algorithms can lead to a better approximation, for example a nonlinear optimal model can be approximated by a combination of linear models.

Before tackling in detail the methodologies and techniques of ensemble methods, it is useful to resume the analogy with the decision-making board mentioned in the Introduction.

Intuitively, it is clear that in order to form a good committee a first feature to be considered is the independence of the members. Indeed, if the committee is composed of elements that “influence” each other or that analyze the same aspects of a problem, the final decision will not be much more accurate than the decision taken individually by the members. If, on the contrary, the committee is composed of “independent” elements, or, roughly speaking, the members make mistakes on different data, it is expected that the other members of the committee will compensate for the mistake made by the individual member.

As mentioned in the introduction, this intuitive concept finds a theoretical validity in Condorcet’s *Jurys Theorem*. Let consider a jury formed by a group of n people having a probability p of voting correctly and a probability $1-p$ to vote incorrectly; the theorem states that if: (i) each member of the jury has a probability $p > 0.5$ and (ii) the members vote independently of each other, then the probability that the combined vote (in this case a majority vote) is correct tends to 1 as the number of voters increases.

Roughly speaking, the Condorcet’s Jury Theorem states that a jury whose jurors vote independently and have an individual competence greater than 0.5 (and we can consider them weak or strong experts), then the probability that the jury’s vote is correct tends to the certain event as the number of voters increase.

Looking at the problem from another point of view, let’s consider a binary classification problem and n generic C_i classifiers with $i=1..n$ having an error rate lower than the random guessing model, i.e. $err_{base,i} < err_{rand}=0.5$ for $i=1..n$. Under the assumption that the classifiers are independent, that is, their error rate are uncorrelated, then the probability distribution of the ensemble error rate follows a binomial distribution. In particular, if we set $n=9$ and $err=0.3$ and we suppose that the output of the ensemble model is given by the outcome having the majority vote of the base learners, then the probability that the ensemble classifies an element incorrectly is given by the probability that $\lfloor n/2 \rfloor + 1$ classifiers vote incorrectly, or in this case:

$$\begin{aligned} ens\ err &= \sum_{k=\lfloor n/2 \rfloor + 1}^n \binom{n}{k} err_{base}^k (1 - err_{base})^{(n-k)} \\ &= \sum_{k=11}^{21} \binom{21}{k} (0.3)^k (0.7)^{(21-k)} \\ &= 0.0988 \end{aligned}$$

that is the ensemble error is significantly lower than the base error rate.

In the [Fig. 1](#) the relationship between base learning errors and the resultant ensemble error is shown for an error rate in a range between 0 and 1 and considering different number of base learners. It can be seen how, under the assumptions of the Condorcet theorem, i.e., when $0 \leq Base_{err} < 0.50$, the curve lies below the bisector of the first quadrant meaning that the Ensemble error rate is lower than the single classifiers error. Also, as the number of “weak” classifiers increases, for $0 < err < 0.5$ it is evident that the ensemble error curve reaches zero faster.

This example gives an idea of how in ideal situations the use of ensemble learning approach can give very effective results.

However, a basic hypothesis that is used in this example is the assumption of independence of base learners from each others or, in other words, that the errors of base classifiers are not correlated.

Lets now consider the example of three dichotomic classifiers all having probabilities $p=0.6$ to classify correctly and, consequently, and error rate of $err=0.4 < 0.5$. This means that, considering 5 instances, each of the three classifiers makes two mistakes.

The [Table 1](#) shows the combinations of the different outcomes and the last column shows the majority vote accuracy for each combination of the three basic learners. As it can be seen from the table, only in two cases out of 16 the accuracy of the ensemble model is greater than the accuracy of the single classifiers, in 13 cases the accuracy of the combined model equals the accuracy of the single classifiers and in a case out of 16 the accuracy of the individual classifiers is greater than the ensemble model.

classifiers is greater than the ensemble model. Averaging the accuracy of the 16 cases considered, it results that the combined model has a probability $P=61.25\%$ that the majority vote output is correctly classified, i.e., the combined model improves the probability of individual classifiers in average of the 1.25% at the expenses of an increased complexity due to a training and testing phase for all three classifier and to the calculation of the majority vote for each instance.

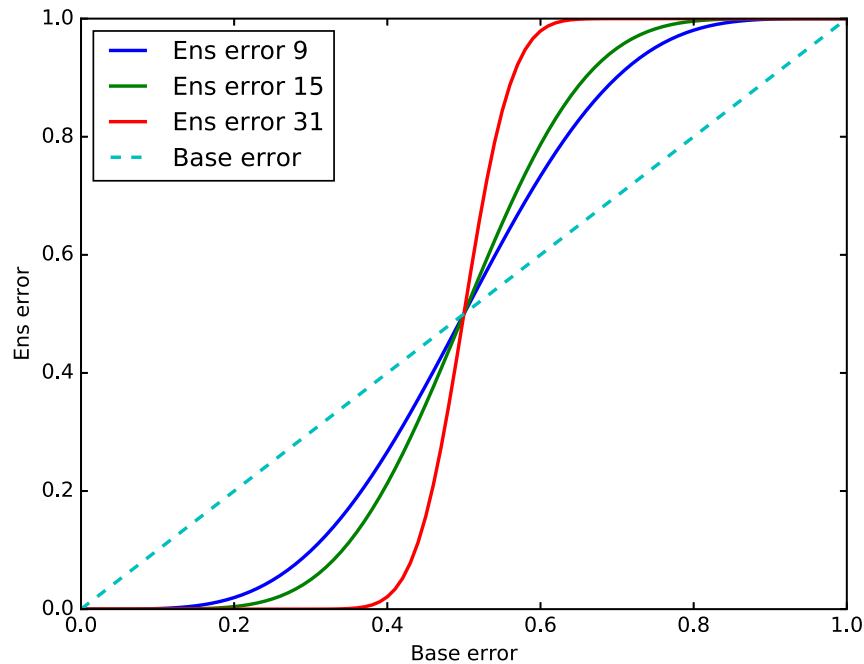


Fig. 1 The figure shows the error rate of three ensemble classifiers when combining 9 (in blue), 15 (in green) and 31 (in red) classifiers respectively. The x axis represent the single classifier error while the y axis represent the ensemble error. The bisector show the error rate of the random guess algorithm. It's easy to see that when each single classifier's error is lower than the random guessing error, then the ensemble error is lower than the single classifier's error and it tends to decrease to zero more faster when the number of classifier grow.

Table 1 The Table reports the possible output's combination of 3 classifier each of which correctly classifies 3 instances out of 5. Indicating with 1 or 0 if that classifier has classified correctly or it has misclassified that instance respectively, in the first row all the possible combination of the three classifiers' outputs are shown. The penultimate column shows for each case the accuracy rate when the classifiers are combined by a majority voting, and in the last column are reported the differences between the ensemble and the base learners accuracies. In the last row the overall accuracy of the ensemble classifier and the overall improvement with respect to the base classifier accuracy are reported

Possible outcomes										
	111	101	110	0 11	100	0 1 0	0 0 1	0 0 0	p_{ens}	$p_{ens} - p$
0	111	101	110	0 11	100	0 1 0	0 0 1	0 0 0	p_{ens}	$p_{ens} - p$
1	3	0	0	0	0	0	0	2	0,6	0
2	2	1	0	0	0	1	0	1	0,6	0
3	2	0	1	0	0	0	1	1	0,6	0
4	2	0	0	1	1	0	0	1	0,6	0
5	2	0	0	0	1	1	1	0	0,4	-0,2
6	1	1	1	1	0	0	0	1	0,8	0,2
7	1	1	1	0	0	1	1	0	0,6	0
8	1	1	0	1	1	1	0	0	0,6	0
9	1	0	1	1	1	0	1	0	0,6	0
10	1	0	1	1	1	0	1	0	0,6	0
11	1	1	0	1	1	1	0	0	0,6	0
12	1	1	1	0	0	1	1	0	0,6	0
13	1	2	0	0	0	2	0	0	0,6	0
14	1	0	2	0	0	0	2	0	0,6	0
15	1	0	0	2	2	0	0	0	0,6	0
16	1	1	1	1	0	0	0	1	0,8	0,2
									0,6125	0,0125

This unsatisfactory result is due to the non-diversity of the three classifiers, in the sense that they often commit the same mistakes on the same instances, and it follows that the error rates made by the three classifiers are somewhat correlated, against the hypotheses previously made.

Therefore the diversity request is very important for combining multiple classifiers or, in general, multiple learners. The next section will focus on the techniques used to do this.

Building an Ensemble Learning System

In the above example, it has been shown how the diversity of outputs of the classifier base greatly influences the performance of the assembly.

Although the concept of diversity is intuitively simple, in practice it is difficult to define a measure that expresses how “different” are learner.

For this reason, in addition to explicit approaches, in which to construct an ensemble, measures are taken to ensure the diversity of learners, also ensemble constructive approaches have been developed that are based on implicitly inducing diversity among learners, or without recourse to the calculation of a measure of “diversity”.

In Fig. 2, is generally represented the process of building an ensemble model. It can be seen how the diversity of learners base can be introduced at different levels, leading to different ensemble methods. For simplicity, let us now consider classification problems, but the following examples can easily be adapted to the case of generic learners:

- Variations in data: Starting from the initial dataset, different sub-sets are built and each one is trained in a single basic classifier;
- Variations of features: Through feature selection techniques, several feature sub-sets are built on which the classifiers are trained;
- Variation among learners: different classifiers are used (most of these approaches are explicit in that it is required that the classifiers be different). Alternatively, the same algorithm can be used setting different parameters, for example varying the k in kNN algorithm the kernel function in SVM etc;
- Variation in combinations methods: models are combined using different techniques that will be detailed in the next section;
- Variation in output labels, for example using the error correcting output code (ECOC).

Methods for Combining Multiple Learners

The approaches for combining multiple learners can be roughly divided into two groups: a fusion-type strategy, where the combination strategy is expected to consider that, for each instance, each model’s output contributes to the output of the ensemble, and a selection-type schema, in which each learner is supposed to be an “expert” in a specific domain, therefore the output of the induced pattern contributes to the output of the ensemble only for the elements of that domain. There are also combined strategies. Concerning fusion strategies, these mainly depend on the type of learner that constitute the ensemble. Below some of these approaches are presented, again considering only classification problem for sake of simplicity.

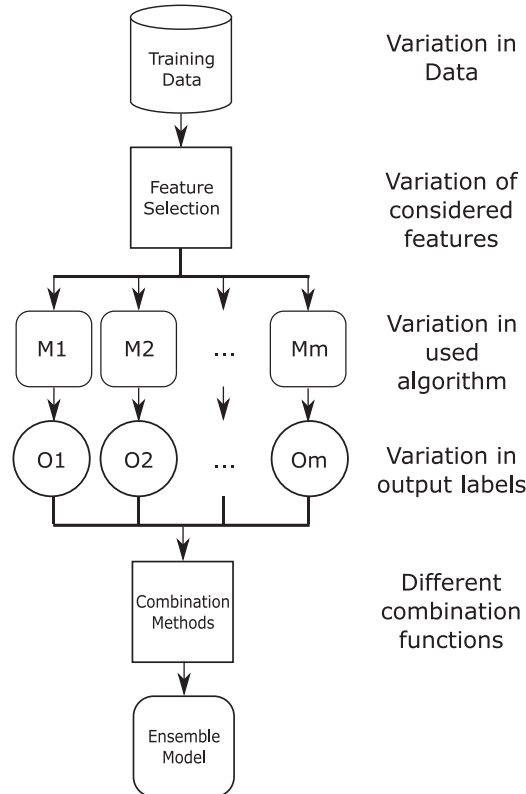


Fig. 2 A general Ensemble process representation. A variation at each different level can inject diversity in the construction of the ensemble’s classifiers.

Combining class label

Majority voting

In majority voting, given an instance, its final prediction label is given by the class having the highest number of concordant predicted labels of the individual base classifiers. There are three possible cases, as shown in the Fig. 3: all the classifiers agree on the forecast (unanimity vote), half plus one of the classifiers agree on the prediction (majority voting that can be applied only to dichotomic classifiers), or the largest number of classifiers agree on the prediction (plurality vote).

Let consider a classification problem, with label set $L = \{L_1 \dots L_m\}$ and N classifier $\{C_1 \dots C_N\}$. For each instance i let consider

$$\chi_{L_k}(C_j(i)) = \begin{cases} 1 & \text{if } C_j(i) = L_k \text{ with } L_k \in L \\ 0 & \text{if } C_j(i) = L_q \text{ with } L_k, L_q \in L \text{ and } q \neq k \end{cases}$$

then, combining the N classifiers with a majority vote, the ensemble prediction's label can be defined as

$$L_{ens} = \arg \max_{L_k} \sum_{j=1}^N \chi_{L_k}(C_j(i))$$

In this case all the N classifiers' votes have the same importance for the ensemble prediction.

A variant of the majority voting is the *weighted majority voting* in which to each classifier is assigned a weight w_j such that $\sum_{j=1}^m w_j = 1$ and the ensemble prediction label will be

$$L_{ens} = \arg \max_{L_k} \sum_{j=1}^N w_j \chi_{L_k}(C_j(i))$$

Borda count

While in voting given an instance i each classifier can votes only for one class label at time, and the ensemble class label obtaining the highest number of votes is considered, the Borda count approach is to let each classifier give a support rank to each class label, producing a ranking list. Then to each position in the list is assigned a decreasing score (for e.g., the class label occupying the first position in the list gets m vote, while the one in last position gets 1 point). Finally, the class label predicted by the ensemble will be the class label having the highest score overall.

Combining functions for continuous decision outputs

If the learners provides as output a numerical values, i.e. for each instance i the prediction of the j -th learner for $j = \dots N$, will be a value $d_j(i)$ in some range. In this case the most followed approach is to define the ensemble model prediction as an algebraic combination of each learner's prediction, that is

$$d_{ens}(i) = F[d_1(i) \dots d_N(i)]$$

Multiple are the possible choices of the algebraic function F , and some are listed below:

- **Mean rule:** $d_{ens}(i) = \frac{1}{N} \sum_{j=1}^N d_j(i)$
- **Weighted average** $d_{ens}(i) = \frac{1}{N} \sum_{j=1}^N w_j d_j(i)$, for w_j be such that $\sum_{j=1}^N w_j = 1$
- **Minimum rule** $d_{ens}(i) = \min_{j \in [1 \dots N]} d_j(i)$
- **Maximum rule** $d_{ens}(i) = \max_{j \in [1 \dots N]} d_j(i)$
- **Median rule** $d_{ens}(i) = \text{median}_{j \in [1 \dots N]} d_j(i)$
- **Product rule** $d_{ens}(i) = \frac{1}{N} \prod_{j=1}^N d_j(i)$

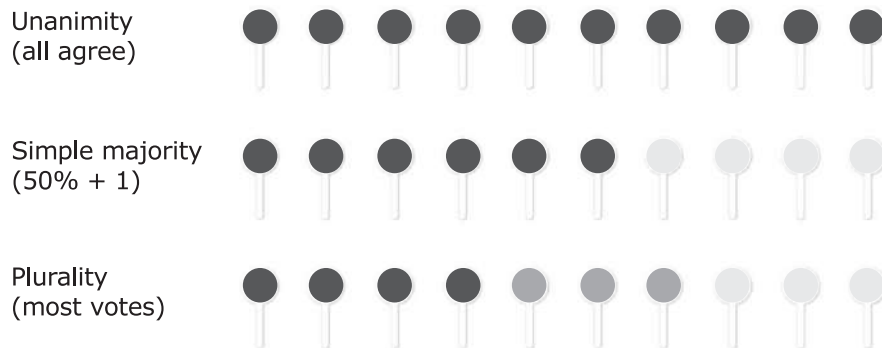


Fig. 3 A example of unanimity, majority and plurality voting.

Most Popular Ensemble Learning Algorithm

In general, it is evident that the class of ensemble methods is made up of many possible algorithms. However, 4 methods can be considered the most representative, namely: Bagging, boosting, Stacking and cascading.

In *Bagging*, i.e., Bootstrap AGGREGATING, n models $M_1 \dots M_n$ are constructed by applying a base learning algorithm A to n bootstrapping sample $D_1 \dots D_N$ (i.e., random samples with replacement) of the dataset D . The n models are then combined with a majority vote. For bagging to be effective, it is expected that small perturbations in the D_i samples produce large variations in the M_i models. Formally, it can be said that bagging is effective for *unstable* algorithms.

Boosting is a method that explicitly induces the diversity among the learners of ensembles, as it aims to maximize the accuracy of an algorithm on the set of instances in which previous models fail.

Lets consider the case of a classification problem. As for Bagging, the first Classifier C_1 is trained on a random sample of the Training Set producing an M_1 model. The second classifier C_2 is trained on a set datasets subset composed by 50% of instances correctly classified by the first classifier and by the other 50% by misclassified instances of the first classifier. A third M_3 model will then be built by training a third C_3 classifier on the subset of the dataset on which the M_1 and M_2 models disagree, and so on. Models thus produced are combined with a majority vote.

Two are the major differences between the previous methods and *Stacking*: the first is that if Bagging and Boosting follow the approach of varying the training set, stacking applies a variation a the algorithm level, i.e. diversity is built by using different algorithms; the second difference is that in stacking methods, models are combined through a Meta-Learner applied on a Meta-dataset containing for each instance the predictions of each base model.

Lastly, with the *Cascading* method, the ensemble is built by applying different algorithms like stacking and by varying the distribution of instances on the training set as with saw in boosting. However, the cascading method follows a multistage scheme, in which as long as an instance I_k is classified by a model M_i (induced by the classifier C_i) with a confidence rate lower than a defined threshold, it is more likely that instance I_k will be selected to be classified by a classifier $C(i+1)$, having a greater complexity than the classifier C_i .

Conclusion

In this article, the theoretical foundation of combining multiple learner's theory is discussed. In particular, through examples it was shown why one of the most important requests for building a good ensemble is that the base learners has to be "weak" expert and, moreover, they have to be "diverse" in a given way. Then, the most used function for combining learners' output and a brief overview of the most used ensemble approaches were presented. In the next articles each of there approaches will be discussed in details.

References

- Alpaydin, E., 2014. Introduction to Machine Learning. MIT press.
- Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140.
- Breiman, L., 2001. Random forests. Machine Learning 45 (1), 5–32.
- De Condorcet, N., *et al.*, 2014. Essai sur l'application de l'analyse a la probabilité des decisions rendues a la pluralite des voix. Cambridge University Press.
- Dietterich, T.G., *et al.*, 2000. Ensemble methods in machine learning. Multiple Classifier Systems 1857, 1–15.
- Freund, Y., Schapire, R.E., *et al.*, 1996. Experiments with a new boosting algorithm. ICML 96, 148–156.
- Ho, Y.-C., Pepyne, D.L., 2002. Simple explanation of the no-free-lunch theorem and its implications. Journal of Optimization Theory and Applications 115 (3), 549–570.
- Polikar, R., 2012. Ensemble learning. In: Ensemble Machine Learning. Springer, pp. 1–34.
- Schapire, R.E., 1990. The strength of weak learnability. Machine Learning 5 (2), 197–227.
- Selfridge, O.G., 1958. Pandemonium: A paradigm for learning in mechanisation of thought processes.
- Wolpert, D.H., 1992. Stacked generalization. Neural Networks 5 (2), 241–259.
- Wolpert, D.H., 2012. What the no free lunch theorems really mean; how to improve search algorithms. In: Santa fe Institute Working Paper. p. 12.
- Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1 (1), 67–82.

Multiple Learners Combination: Bagging

Chiara Zucco, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The development of omics sciences has led to the introduction of methodologies in order to identify, from experiments such as mass spectrometry and microarray, genes involved in a given disease. A bioinformatic approach to this problem involves the application of Machine Learning methods, for example in relation to microarray data or in gene-gene interactions, enabling high-performance analysis procedures.

In particular, the tasks of classifying and predicting genes that can discriminate particular tissue types, has been shown to play a central role for the effective diagnosis of cancers and other diseases.

Unfortunately, there not exist a classification algorithm, and in general, a machine learning algorithm, that always induces the most accurate model. For this reason, in order to improve the accuracy of the model, instead of searching for the best machine learning algorithm, an alternative can be to combine multiple learners having an accuracy slightly greater than random guessing.

These methods, also called ensemble learning methods, take inspiration from real life phenomenon like democratic process, expert teams etc. and, under suitable conditions, have shown to achieve valuable results, see for example [Kuncheva \(2004\)](#); [Polikar \(2012\)](#); [Dietterich et al. \(2000\)](#); [Duda et al. \(2012\)](#).

As will be discussed below, a key point in dealing with ensemble methods is that the base learners (that is, the learners composing the ensemble) have to be chosen or built in order to make different mistakes on different instances of the data set or, in other words, by implicitly or explicitly increasing the diversity between them.

One of the simplest yet effective ensemble approach is Bagging, standing for Bootstrapp AGGREGatING, and presented by [Breiman \(1996a\)](#).

In Bagging As the name suggests, in Bagging ensemble algorithm the base learners are built by training the same algorithm on different training sets, obtained as bootstrap samples of the original training set. In Bagging diversity is therefore enhanced implicitly.

There also exist variations of Bagging, the most popular of which is Random Forest, introduced by [Breiman \(2001\)](#).

Bagging and Random Forest found wide applications for improving accuracy in microarray data prediction.

One of the first work in this sense was done in [Ben-Dor et al. \(2000\)](#), while Tan and Gilbert first prove that ensemble learner can improve the coverage of the protein fold classification classifiers of a multi-class imbalanced sample ([Tan et al., 2003](#)), and then performed a comparative study of bagged and boosted decision trees over a single decision tree algorithm, for the classification of seven cancerous microarray dataset, showing that ensemble methods effectively improve the performances of a single decision algorithm ([Tan and Gilbert, 2003](#)). Another comparative study was carried out in [Wu et al. \(2003\)](#), where bagging boosting and random forest performances were compared with single classifiers such as kNN, SVM and others. The algorithm were evaluated for classifying ovarian cancer from mass spectra serum profiles and results shown that random forest achieved, on average, the lowest error rate.

These are only few among the examples on how successfully Ensemble methods and, in particular, bagging can be applied in bioinformatics, although an extensive review of studies in this field is beyond the purpose of this work.

The aim of this article is twofold: to provide the underlying fundamentals of Bagging and to present the Bagging and random forest algorithms.

In particular the starting points of this discussion will be resampling techniques and some theoretical results involving two variables useful for measuring the performances of a model and the relation between the to of them and the error rate. This is known as *bias-variance tradeoff/dilemma* and will be discussed in Section "Motivation and Fundamentals", while in Section "Bagging and Random Forest" will be outlined the step for constructing a bag of classifiers.

Motivation and Fundamentals

In this section some background information are provided. In particular, some theoretical results underlying machine learning algorithm are discussed and some resampling techniques are presented, dealing with their effect on a classification problem.

The Bias-Variance Trade-Off

When dealing with a classification or a regression problem, in addition to error due to noise in data, two are the main variables that can affect the performances of a model induced by a machine learning algorithm:

- Bias, that is, the measure of how the prediction model differs from the real value, in other words how the model is accurate.
- The variance, i.e., the measure of how much the model is sensitive to slight changes in the training set.

Although there are designing techniques devoted to adjusting the algorithm and reducing bias or variance, it will be shown below that bias and variance are not independent of each other but respond to a same relationship, called Bias-variance decomposition.

For sake of simplicity, a regression problem is considered in the following.

Let consider an arbitrary but fixed training set $\mathcal{D}$ and let $f(x)$ be a continuous valued function representing the unknown solution (also called true function) for the regression problem. Let $r_{\mathcal{D}}(x) = r(x, \mathcal{D})$ be the function modelling the regression problem or, that represents an estimation for $f(x)$.

In order to judge the quality of this estimation on $\mathcal{D}$, a useful theoretical measure is given by the *mean squared error*, i.e., the second moment of the deviation or, in other terms, the expected value of the quadratic error, $E[r_{\mathcal{D}}(x) - f(x)]^2$.

Applying well known properties of the expected values, the mean squared error can be expressed as

$$\begin{aligned} E[(r_{\mathcal{D}}(x) - f(x))] &= \underbrace{E[(r_{\mathcal{D}}(x) - f(x))]^2}_{bias^2} + \underbrace{E[(r_{\mathcal{D}}(x) - E[r_{\mathcal{D}}(x)])^2]}_{variance} \\ &= \quad \quad \quad + \end{aligned} \quad (1)$$

and it's known as the bias-variance decomposition [Sammut and Webb \(2011\)](#), [Duda et al. \(2012\)](#). Eq. (1) shows a general problem of machine learning: the more the model adds degrees of freedom to better adapt to data (low bias), the more it's sensitive to perturbation in the dataset (high variance) and this problem is referred to as *overfitting*. On the contrary, the lower are the degrees of freedom for the model, the greater will the sensitivity of the model, with a consequent "poor" adapting to data (underfitting). For example, if we consider as true function $f(x) = \sin(3\pi x)$ with noise, aiming to estimate $f(x)$ on a training set $\mathcal{D}$ of 15 points, with polynomial functions of degrees 1, 3 and 15 respectively. In [Fig. 1](#) is shown how the increase of the degrees of freedom tends to minimize the bias, fitting better the data, but incrementing the variance.

Of course, the ideal solution is to build a model having low bias and low variance (bias-variance trade-off).

Resampling Techniques

After introducing bias and variance, the next step is to determine these measurements to evaluate the performance of a learning algorithm for problems with unknown distribution. A class of statistical techniques useful in this context is called resampling. In particular, the focus will be made to the Bootstrapping that is a resampling technique that is also used for constructing the training subsamples on which the classifiers are trained in Bagging.

Bootstrap resampling

The term *Bootstrap* is referred to the resampling techniques of randomly selecting from a dataset $\mathcal{D}$ with N -cardinality, a set of $n \leq N$ instances with replacement. In the next Section it will see that this process can be repeated in order to generate m bootstrap dataset.

Example 2.1: Let now consider a set $\mathcal{D}$ of $n=N$ distinct values, then the bootstrapping process can generate $(2N - 1)$ distinct subsets. Moreover, in this case each instance of the dataset has probability $\frac{1}{N}$ of being randomly selected to populate the j^{th} subset, and has a probability $1 - \frac{1}{N}$ of not being selected. Therefore, each instance i has probability $P_N = (1 - \frac{1}{N})^N$ of not appearing in the j^{th} bootstrapped subset. As N increases, $P_N \rightarrow \frac{1}{e}$ that is, approximately, $\frac{1}{3}$ of the original dataset.

In the *balanced bootstrap sampling* the generation of the bootstrap sets takes into account also the request that the instances of the original dataset have to be present a number of time that is exactly the number m of generated bootstrap sets. This means that if an instance is present with repetition in a training subset, there will be another training subset in which that same instance will not appear. Balanced bootstrap samples are generated starting from a set with copies for every m instances, and then randomly altering the starting dataset by adding or deleting instances.

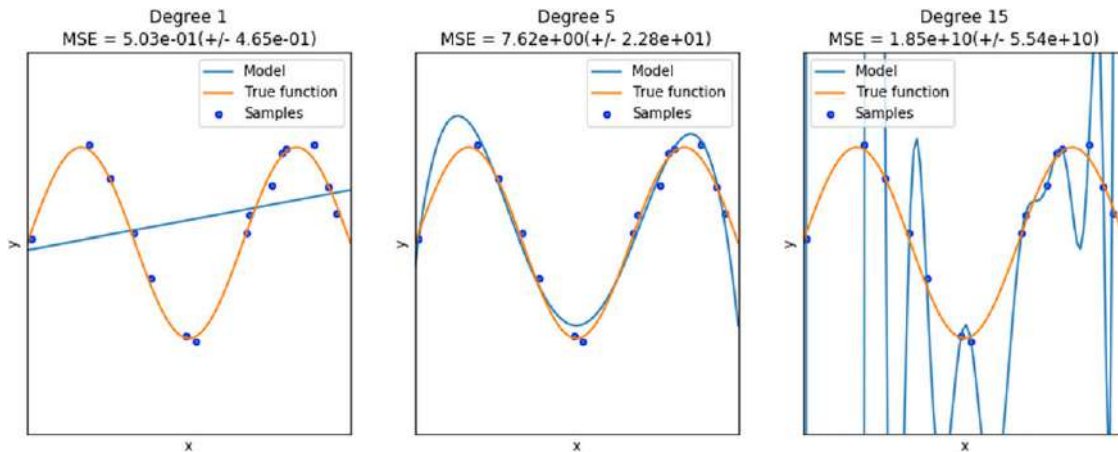


Fig. 1 The Figure shows how increasing the degrees of freedom, the curves estimating the true function $f(x)$ tends to fit more the data points (low bias) and to increase the oscillation (high variance).

A bias-variance decomposition holds true also in classification fields but here the proof is more subtle, because in this case can be shown that the relationship between the estimated error and the couple bias-variance isn't linear anymore but becomes multiplicative.

A last example shows how averaging behaves on bias and variance.

Example 2.2: Let d_j for $j=1\dots m$ be independent variables identically distributed (i.i.d.) and, therefore, having $E[d_j]$ as expected values. Let y be the variable obtained by averaging d_j for $j=1\dots m$. Then

$$E[y] = E\left[\sum_{j=1}^m \frac{1}{m} d_j\right] = \frac{1}{m} E\left[\sum_{j=1}^m d_j\right] = E[d_j]$$

and

$$Var(y) = Var\left(\sum_{j=1}^m \frac{1}{m} d_j\right) = \frac{1}{m^2} Var\left(\sum_{j=1}^m d_j\right) = \frac{1}{m} Var(d_j).$$

From the last series of equivalence, it can be seen that the variance of y decreases as the number of m increases without increasing the bias that, due to the i.i.d. hypothesis, remains stable. As a consequence, also the error decreases.

Applying an averaging function leads to lower the variance also when the variables $\{d_j\}$ are negatively correlated but, in this case, the bias tends to increase. On the contrary, if the variables $\{d_j\}$ are positively correlated, averaging leads to an increment of both variance and error.

Bagging and Random Forest

The Bagging Algorithm

The idea underlying bagging is related to the example of the last Section, that is to construct an ensemble of independent model with high variance and low bias and then to average over that models in order to decrease the variance without significative effects on the bias.

Bagging, as the other ensemble learning methods, can be seen as composed by two steps: in the first one, the single learning models are generated, in a way that can emphasizes diversity, and consequently keeping the correlation among models low; in the second step, the models are combined, generally by using a combining function.

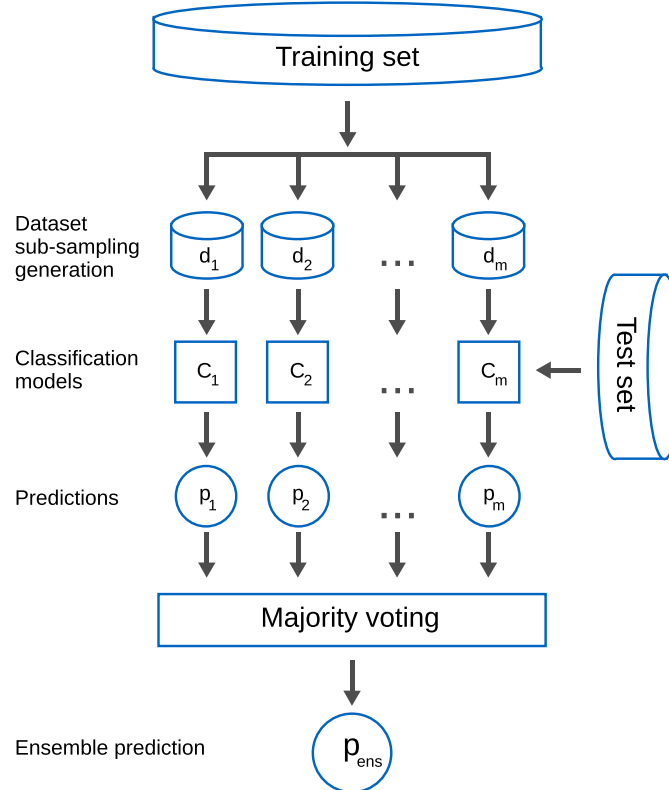


Fig. 2 A bagging process representation.

Specifically, referring to a classification on a numerical prediction problem, in bagging the models that will constitute the ensemble, are built by training the same learning algorithm on m subsets generated from the training dataset $\mathcal{D}$ by applying a balanced bootstrap samples process.

Then, the same test set will be used for testing every model and the ensemble prediction is finally obtained by a majority voting for a classification problem and by averaging on the models' output in numerical prediction.

As stated in the introduction, Bagging is one of the most simple ensemble methods, but yet can perform with outstanding results when used with *unstable* learners.

An unstable learner is a machine learning algorithm whose generated model tends to respond to slight changes in the training set with significant changes in the model. Therefore, it is clear that unstable learners trained on m bootstrap samples produce m different models, ensuring in general the diversity required for improving the overall accuracy. In reality, this is true only for not large training set.

Below, the pseudo-code for bagging is illustrated in [Algorithm 1](#), while [Fig. 2](#) summarizes the bagging process.

Algorithm 1: Bagging algorithm.

1 Models generation

2 Initialize the parameters:

- the ensemble $\mathcal{E} = \emptyset$;
- the number n of instances of the training data;
- the number m of learners;
- the algorithm $\mathcal{A}$;
- a set of prediction $P_1, \dots, P_q$;

For $j = 1 \dots m$:

- generate a bootstrap sample $\mathcal{D}_i$ instances from the training set;
- build the learner $\mathcal{C}_i$ training $\mathcal{A}$ on $\mathcal{D}_i$;
- add the learner $\mathcal{C}_i$ to the learner set, $\mathcal{E} = \mathcal{E} \cup \mathcal{C}_i$;
- return $\mathcal{E}$;

Testing

Apply $C_1 \dots C_m$ on the new instance $\mathbf{i}$ calculating the predictions $C_j(\mathbf{i})$

Calculate the ensemble prediction by:

$$P_{\mathcal{E}} = \arg \max_k \sum_{j=1}^n \chi(C_j(\mathbf{i}) = P_k)$$

Randomization

The strength point of bagging is to implicitly address the complex problem of the model diversity, by inducing randomness in the training subset of the learning algorithm and then taking advantage of the instability of the base learner. For example, an ensemble built using as base learner a stable algorithm as nearest neighborhood it will lead an increase in complexity without substantial improvement in performances.

Another issue of using bagging alone, is that the level of diversity achieved are lower if compared to other approaches and can be further increased only by combining bagging with other techniques.

Indeed, in general there are also other approaches for injecting randomness within the algorithms, especially in case of algorithms that have an embedded random component. In this cases, starting from a base learning algorithm, the idea is to obtain the ensemble learners from the base learner by random varying some parameters, for example the initial weights in neural networks, or in decision tree algorithm by randomly selecting the attribute from a list of best attribute in order to split-on at each node.

When dealing with randomization the goal is to increase diversity between learners but the risk is to obtain a significant decrease in the single learner's accuracy. Moreover, injecting randomness in general requires modifications in the learning algorithm, while it can lead benefits also with stable base algorithms.

An idea is to combine bagging with randomization technique in such a way they can complementary level of randomness and, therefore, emphasize the diversity among learners. An example of how this combination can bring to excellent results is *Random forest*.

Random Forest

Random forest is a variant of bagging in which the base learner is a decision tree, proposed by [Breiman \(2001\)](#). Below, is given a formal definition:

“Let $v_1, \dots, v_m$ be i.i.d. vectors. A random forest is a classifier built by combining through majority voting m decision trees $T_1, \dots, T_m$ grown with respect to $v_1, \dots, v_j$ respectively”.

So each random tree of the random forest can be grown by considering the bagging procedure of bootstrap subsampling of the training set, a random subsample from the features set, by varying some parameters or by a combination of these approaches. In order to alleviate the “course of dimensionality” in high-dimensional dataset and to improve the diversity among the decision trees of the ensemble, a common approach is to build each decision tree on a different bootstrap sample and on a different subset of features, without replacement.

The random forest algorithm can be summarized in few steps:

- Generate a bootstrap sample $\mathcal{I}$ of dimension n from the training set $\mathcal{D}$.
- For each $i = 1 \dots m$ grow the i -th decision tree by randomly selecting q features without repetition and then splitting the node based on the best feature. Supposing that the Feature set contains Q features, a common used value for setting the dimension of the features' subset is $q = \sqrt{Q}$.
- Combining the prediction by a suitable combination function (such as Majority vote for classification problems and by averaging prediction in regression problems).

One of the advantages of random forests compared to decision tree model is that, in this case, each decision tree grows with bootstrap samples of the original training set and with a subset of different features for each tree, then the generated model is more robust and less susceptible to overfitting, and this is the reason for which, in general, it's not required to prune the random forest. As previously mentioned, even averaging function and majority voting methods tend to further decrease variance without significant bias repercussions.

The pseudocode of the Random Forest algorithm is shown below 2.

Algorithm 1: Random Forest algorithm.

1 Random Forest generation

2 Initialize the parameters:

- the Forest $\mathcal{F} = \emptyset$;
- the Dataset $\mathcal{D}$;
- the set of Feature $\mathcal{F}$;
- the number m of trees in the forest;

For $i = 1 \dots m$:

- generate a bootstrap sample D_i of instances from $\mathcal{D}$;
- build the i -th tree $T_i = \text{tree}(D_i, \mathcal{F}_i)$:
 for each node:
 generate a subset F_i of feature from $\mathcal{F}$ without repetitions;
 split the node using the best feature;
 return $\mathcal{T}_i$
- generate a subset F_i of feature from $\mathcal{F}$;
- add the tree T_i to the forest, $\mathcal{F} = \mathcal{F} \cup T_i$;
- return $\mathcal{F}$;

Combining Prediction's trees

For each instance i calculating the predictions $P_j = T_j(i)$

Calculate the ensemble prediction by

$$P_{\mathcal{F}}(i) = \text{Average}(T_1, \dots, T_m, i) \text{ OR}$$

$$P_{\mathcal{F}}(i) = \text{Majority vote}(T_1, \dots, T_m, i)$$

Out-of-Bag Error

As seen in the example 2.1, each algorithm composing the ensemble learner is trained on a bootstrap subsample $\mathcal{D}_i$ containing, on average, the 64% of instances from the original training set $\mathcal{D}$, while the remaining 37% of instances of each subsample are replicated instances. Another 37% of instances belonging to the original training set do not belong to $\mathcal{D}_i$ and are called “out-of-bag” instances. So, when the number of instances of the dataset is suitable to be divided in training set, Out-Of-Bag (OOB) set and test set, the idea is to use the OOB instances predictions in order to have efficient estimates of the generalization performances of Bagging ensemble. The OOB error can be obtained by averaging the generalization estimates of all the learners of the ensemble. In particular OOB instances have been used to provide estimates of node probabilities and error rates for each tree of the random forest or, in regression trees, it was seen that by replacing OOB estimated prediction instead of the observation outputs can improve performances Breiman (1996b).

Conclusions

In this article the basic concepts of Bagging algorithms were introduced, with particular focus on theoretical background and on the motivations thanks to which Bagging ensemble algorithm and its variant can improve prediction problems.

See also: Kernel Machines: Applications. Multiple Learners Combination: Introduction

References

- Ben-Dor, A., Bruhn, L., Friedman, N., *et al.*, 2000. Tissue classification with gene expression profiles. *Journal of Computational Biology* 7 (3–4), 559–583.
- Breiman, L., 1996a. Bagging predictors. *Machine Learning* 24 (2), 123–140.
- Breiman, L., 1996b. Out-of-bag estimation.
- Breiman, L., 2001. Random forests. *Machine Learning* 45 (1), 5–32.
- Dietterich, T.G., *et al.*, 2000. Ensemble methods in machine learning. *Multiple Classifier Systems* 1857, 1–15.
- Duda, R.O., Hart, P.E., Stork, D.G., 2012. *Pattern Classification*. John Wiley & Sons.
- Kuncheva, L.I., 2004. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons.
- Polikar, R., 2012. Ensemble learning. In: *Ensemble Machine Learning*. Springer, pp. 1–34.
- Sammut, C., Webb, G., 2011. Bias – Variance decomposition. In: *Encyclopedia of Machine Learning*.
- Tan, A.C., Gilbert, D., 2003. Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics* 2 (Suppl. 3), S75–S83.
- Tan, A.C., Gilbert, D., Deville, Y., 2003. Multi-class protein fold classification using a new ensemble machine learning approach. *Genome Informatics* 14, 206–217.
- Wu, B., Abbott, T., Fishman, D., *et al.*, 2003. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics* 19 (13), 1636–1643.

Multiple Learners Combination: Boosting

Chiara Zucco, University of Magna Græcia of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Boosting is an iterative procedure for combining multiple learners, easy to implement and extremely powerful and, for this reason, represents an important element in the repertoire of learning mechanisms.

Since we will pose accent on supervised learning algorithm, besides the word learner, it will be also used the terms classifier or predictor.

A multiple learner combiner, also known as ensemble learner, can be viewed as a committee of learners whose predictions are combined in a suitable way to achieve better performances than any single learner in the committee, see Polikar (2012) and Alpaydin (2014).

Another way to see an ensemble learner method is as a procedure that allows to build a strong learner (having high performances in sense that we will formulate in the following) starting from a set of base or weak learners (slightly better than random guessing).

Two are the fundamental points in constructing an ensemble learner:

- To choose or build base learners to make different mistakes on different instances of dataset, by enhancing diversity among them;
- To choose a suitable rule for combining each learner's output.

As previously seen, if in Bagging diversity among learners is enhanced in an implicit way by training in parallel the same algorithm on n bootstrapped replicates of the training sample, the idea behind Boosting is to train learners in a sequential fashion, boosting the subsequent learner by acting on the training sample distribution and increasing weights on instances that were predicted wrong by the previously trained learner, see Kuncheva (2004) and Breiman *et al.* (1998).

The history of Boosting followed somehow a special way from that of the other ensemble algorithms. In fact, boosting finds its roots in the Computational learning theory and in particular in the Probably approximately correct learnability (PAC-learnability) framework, the first theoretical learning model presented by Valiant (1984). In particular, the first idea of "boosting" was proposed by Kearns (1988), while Schapire (1990), presented a first "Boosting" algorithm as a recursive process to show the equivalence of two sets of PAC-learner sets: strong learner and weak learner. A year later, Freund (1995), starting from the work of Schapire, proposed a non-recursive algorithm and introduced a majority-vote function for combining the weak learner's outputs. These approaches were both suffering of some practical drawbacks and, working together in the following years, Freund and Schapire (1995) presented an algorithm, called AdaBoost, thanks to which they won the Gödel award, one of the highest recognitions in theoretical computer science. Although Boosting can be considered as a family of algorithms, since there are many variations and "specializations" of Boosting, see Schapire and Freund (2012), AdaBoost still remains the boosting algorithm most used in practice, thanks to its simplicity and versatility.

In this article, Adaboost is introduced, trying to indicatively follow the path just described. In particular, in Section "Background and Fundamental" some basic principles of PAC-learnability are laid out and the classes of weak and strong learners are introduced. Following, a first approach to Boosting is presented, while in Section "Adaboost" AdaBoost is presented.

Background and Fundamental

In order to roughly explain the core idea of PAC-learnability, let's consider the following example.

Suppose that a buyer B who is looking for a new home, goes to a real estate agent E to find out what is the house price range of a specific neighborhood. The agent does not directly disclose the price range $[a, b]$, but he provides some examples by generating at random house prices by following a fixed way (or Distribution), and showing the buyer the value of that random price, let's say x , and a label y , that is saying if that house price x belongs to the house prices range of the neighborhood (positive examples, labeled as $+1$) or if it doesn't (negative examples, labeled as -1). Since E only provides finitely many examples and being a and b in general real numbers, B can not exactly determine the extremes of the interval. Therefore, from the examples provided by E, the objective of buyer B is to give an answer that is close enough to the real interval $[a, b]$, i.e., an approximation $[c, d]$.

For each real interval considered, it is possible to calculate the probability that, as the number of examples given by E increases, the range indicated by B does not represent a good approximation for $[a, b]$, and therefore that B will give an incorrect answer. If this probability is appropriately "small" or, in other words, if the error between the real range and the approximation given by B has high probability to be very small, it can be said that B has learned the interval. Assume then that this experiment is repeated several times, varying the interval and the strategy or distribution used by E for generating prices. If every time B finds, with high probability, a good approximation for the interval, it can be said that the problem is Probably Approximately Correct (PAC)-learnable.

With this example in mind, we can provide some formal definition.

Let X be a set or domain, also called *instance space* and $Y = \{-1, +1\}$ be a set of labels. It is assumed that there exist some probability distribution D , generating couples (x, y) . We assume that there also exists a *target function* $f: X \rightarrow Y$ be such that, for each (x, y) extracted from the distribution D , the following:

$$\mathbf{P}_{(x,y) \sim D}[y = f(x)|x] = 1$$

holds true, that is the examples are in the form $(x, f(x))$.

Let's suppose that to a learning algorithm A are given some *labeled examples* (x_i, y_i) , generated by the unknown distribution D . Starting from A , the goal is to construct a *learner* h (also called hypothesis, or model, or classifier) that, having access to the labeled examples generated by D , can approximate the target function f with high probability.

Given a learner $h: X \rightarrow \{-1, +1\}$ with respect to a target function $f: X \rightarrow \{-1, +1\}$ and the distribution D , the *error* is defined as

$$\text{err}_{f,D}(h) = P_D(h(x) \neq f(x))$$

We now give the definitions for an algorithm to be a strong or weak learner. For further details, see the works of [Valiant \(1984\)](#) and [Schapire \(1990\)](#).

Definition 2.1: Let X be a domain with size n and let $\mathcal{F}$ be a family of target functions $f: X \rightarrow \{-1, +1\}$. An algorithm $A = A_{\epsilon, \delta}$ is said to **strong PAC learn** the family of target functions F over the domain X if, for any target function $f \in \mathcal{F}$, for any distribution D over X and for any $0 < \epsilon, \delta < 1/2$, it holds true

$$\mathcal{P}_D(\text{err}_{c,D}(h) \leq \epsilon) > 1 - \delta$$

i.e., the model h built by the algorithm A has error at most ϵ with probability at least $1 - \delta$. Moreover, A must run in polynomial time in $\frac{1}{\epsilon}, \frac{1}{\delta}$ and n .

An algorithm that strong PAC learns A family F of target functions on X is said to be a *strong learner*.

Another important class of learner is the class of *weak PAC-learners*, introduced in the following.

Definition 2.2: Let X be a domain with size n and let $\mathcal{F}$ be a family of target functions $f: X \rightarrow \{-1, +1\}$. An algorithm $A = A_{\epsilon, \eta}$ is said to **weak PAC learn** the family of target functions F over the domain X if, for any target function $f \in \mathcal{F}$, for any distribution D over X and for any $0 < \delta < \frac{1}{2}$, and for some fixed $0 < \eta < 1/2$ it holds true that

$$\mathcal{P}_D\left(\text{err}_{c,D}(h) \leq \frac{1}{2} - \eta\right) > 1 - \delta$$

i.e., the model h built by the algorithm A has error which is slightly better than random guessing, with probability at least $1 - \delta$. Moreover, A must run in a polynomial time in $\frac{1}{\epsilon}, \frac{1}{\delta}$ and n .

An algorithm that weak PAC-learns A family F of target functions on X is said to be a *weak learner*.

Schapire roughly defines a weak learner as a “rule of thumb”, a rule that can be easily learned and applied but not accurate, most based on practical experience rather than theory. As example of this “rule of thumb” we can consider a decision stump, i.e., a decision tree having just one node and, therefore, in which the tree root is directly connected to the leaves.

The First Boosting, or Boosting by Filtering

As said before, the first version of Boosting was used as a constructive proof of the equivalence of strong and weak learners in free distribution context.

This equivalence naturally means that a concept can be weak-learned if and only if it can be strong-learned.

If the first implication is trivial because if a problem is strong-learned, it means that there exists a strong learner that learns it. But, by definition, a strong learner is also a weak learner, and this show the first implication.

It remains to show that if a concept is weak-learnable then, starting from this weak learner, a strong learner can be built.

The idea of the proof follows two steps:

- A first model, slightly more accurate than the weak model, is built by subsequentially training the same algorithm on three different subset of X , obtained by a suitable filtering methods.
- It is shown that the error can be made arbitrarily small, by recursively applying the first step.

As the whole demonstration is beyond our goals, we will only focus on the first step, that is constructing an algorithm E that simulates the algorithm A on three different distributions (building three different weak models), and then outputs a model H significantly closer to c .

Let A be an algorithm that produces with high-probability a weak learner h' whose error is $\text{err}_{c,D}(h') = \alpha \leq \frac{1}{2} - \eta$ for some fixed $0 < \eta < \frac{1}{2}$. The steps used for sequentially build the model H are the following:

- (1) From the Training Dataset T , with the original (uniform) distribution D , a subset T_1 is extracted. A weak learner h_1 is built by training the algorithm A on T_1 .

- (2) The algorithm A is forced to pay attention to the “harder” parts of the training set T_1 , that is on the examples that were misclassified by h_1 . To do this, a distribution D_2 is built in a way that an instance has equal chance of being correctly or incorrectly classified by h_1 , by filtering from T_1 those instances well classified by h_1 and then adding the same number of instances correctly classified by h_1 and a new training set T_2 is built starting from D_2 . A second weak learner h_2 is the model output of training A on T_2 .
- (3) Finally, a third weak learner h_3 is built by training A on a set T_3 containing instances on which models h_1 and h_2 disagree, with T_3 having a new distribution D_3 constructed by filtering from D those instances on which h_1 and h_2 agree.
- (4) At last the model H , output of E follows this rule:

$$H(i) = \begin{cases} h_1(i) & \text{if } h_1 = h_2 \\ h_3(i) & \text{if } h_1 \neq h_2 \end{cases}$$

It can be found that a bound for the error h is given by the function $err(H) \leq 3\alpha^2 - 2\alpha^3$ and, being $\alpha < \frac{1}{2}$ it follows that the error of H is smaller than α .

Adaboost

Unlike the constructive algorithm presented in the previous section where starting from a subset T , the following model was the result of training on a new subset obtained from the previous through a filter operation, in AdaBoost every weak learner is trained on the whole training set T . In order to encourage the weak learner of step $k+1$ to pay more attention to the instances that have been misclassified in step k , the distribution associated with the dataset is updated at each iteration and, consequently, the weights given to each attribute are updated, so that the outputs of the models are combined through a weighted majority vote.

AdaBoost stands for **ADaptive BOOSTing**, in the sense that the distribution of the dataset on which to train the algorithm A, is adaptively modified at each iteration, on the basis of the misclassified outputs.

Before introducing Adaboost's pseudocode, let's see with a well known example how this algorithm works. Let's consider a dataset T , composed by 10 instances, five of which are positive (shown with red points) and the remaining 5 are negative instances (shown in blue). And suppose to perform a binary classification. At a first step, the distribution of the dataset is uniform, so each point in the figure has equal weight $\frac{1}{10}$. On this set is trained a decision stump. The model M_1 is shown in Fig. 1, step1, subfigure A). At the second step, the distribution of the dataset T is updated (see Fig. 1, Step 2, subfigure 2) in order to train the decision stump to build a second model M_2 that pays more attention to the misclassified points (see Fig. 1, Step 2 subfigure B). At the third step, once again the distribution of T is updated and more weight is given to the three points misclassified by M_2 , then the algorithm is trained on that distorted version of T (see Fig. 1, Step 3, subfigure C). The three models are finally combined by a weighted majority vote, giving the ensemble model shown in Fig. 1 (see subfigure E).

If intuitively the steps of Adaboost are now clear and can be generally summarized in the Pipeline shown in Fig. 2, what remains to be defined is how the distribution D_k is updated at each step. We expect that, for each instance i in the training set T , at step $k+1$, an iterative punctual definition for the distribution function $D_{k+1}(i)$ has to take into account:

- The weight $\alpha_{k'}$ of learner $h_{k'}$, that will depend on the error $\epsilon_{k'}$;
- The punctual distribution defined in the previous step $D_k(i)$;
- The information that the model h_k has correctly classified the instances i or not.

Thanks to the assumption that for each k are defined the functions $f, h_k: X \rightarrow \{-1, +1\}$, it follows that the last information is given by the product $f(i)h_k(i)$, because:

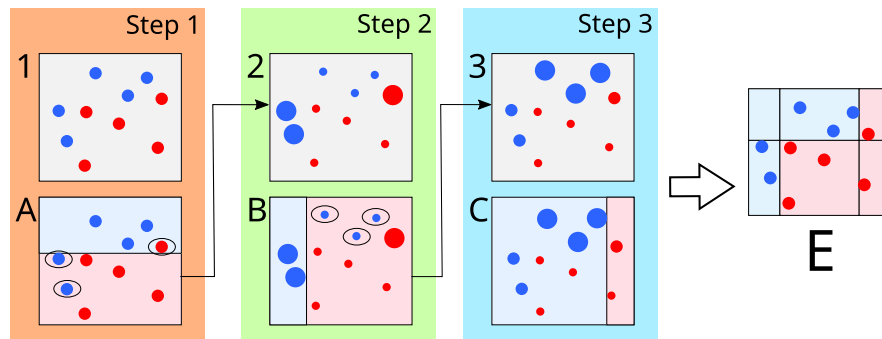


Fig. 1 A classic example of an AdaBoost classifier consisting of three decision stumps (the weak learners), shown in subfigures A, B and C, respectively. At each step, the training subsets distribution is updated, in order to pay more attention to misclassified instances (see subfigures 1, 2, 3). In the final step (subfigure E) the three classifiers are combined through a weighted majority vote.

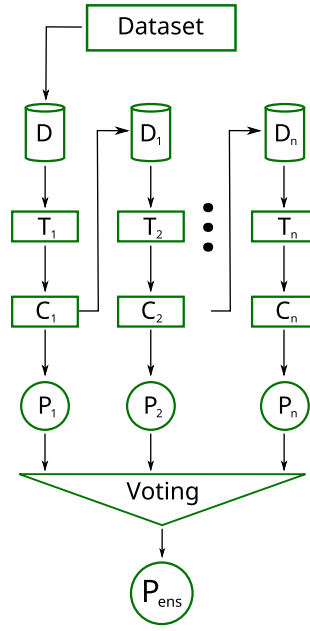


Fig. 2 A pipeline for AdaBoost algorithm.

$$f(x)h_k(x) = \begin{cases} 1 & \text{if } f(x) = h_k(x) \\ -1 & \text{if } f(x) \neq h_k(x) \end{cases}$$

In order to minimize the training error in the shortest possible time, it can be shown that (see [Freund and Schapire, 1997](#)), for every instance $i=1 \dots n$, a good recursive definition of D_{k+1} is the following:

$$D_{k+1}(i) = \frac{D_k(i)}{Z_t} e^{h_k(i)\gamma_i\alpha_k}$$

where we used that $f(i)=\gamma_i$ for $i=1 \dots n$, and Z_t is a suitable normalization factor for D_{k+1} to be a distribution, α_k is a parameter depending on the error ϵ_k of h_k on D_k . In particular a good choice for α for a binary classification problem is shown to be:

$$\alpha_k = \frac{1}{2} \ln \left(\frac{1 - \epsilon_k}{\epsilon_k} \right)$$

In the following Algorithm [1], a pseudocode for Boosting is presented.

Algorithm 1: *Bagging algorithm*

Input

- The training Set $T=(x_1, \gamma_1), \dots, (x_n, \gamma_n)$ where $\gamma_k \in -1, +1$ for $k=1, \dots, n$;
- The base algorithm A ;

Models generation

Initialize the following parameters:

- The initial distribution $D_1(i) = \frac{1}{n}$;
- The number of generating models m .

For $k=1 \dots m$:

- Build the learner h_k by training A on T using the distribution D_k ;
- Update the error $\epsilon_k = \sum_{i \in T} h_k(x_i) \neq \gamma_i D_k(i)$;
- Update the weights $\alpha_k = \frac{1}{2} \ln \left(\frac{1 - \epsilon_k}{\epsilon_k} \right)$;
- Update the distribution $D_{k+1}(i) = \frac{D_k(i)}{Z_t} e^{h_k(x)\gamma_i\alpha_k}$;

Output

For every x ,

$$H(x) = \text{sign} \left(\sum_{k=1}^m \alpha_k h_k(x) \right)$$

Conclusions

In this article the basic concepts of Boosting algorithms were introduced, with particular focus on theoretical background and on the motivations thanks to which Boosting ensemble algorithm and its famous variant, AdaBoost, can improve prediction problems.

See also: Kernel Machines: Applications. Multiple Learners Combination: Introduction

References

- Alpaydin, E., 2014. Introduction to Machine Learning. MIT press.
- Breiman, L., *et al.*, 1998. Arcing classifier (with discussion and a rejoinder by the author). *The Annals of Statistics* 26 (3), 801–849.
- Freund, Y., 1995. Boosting a weak learning algorithm by majority. *Information and Computation* 121 (2), 256–285.
- Freund, Y., Schapire, R.E., 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In: *European Conference on Computational Learning Theory*. Springer, pp. 23–37.
- Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55 (1), 119–139.
- Kearns, M., 1988. Thoughts on hypothesis boosting. Unpublished Manuscript 45, 105.
- Kuncheva, L.I., 2004. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons.
- Polikar, R., 2012. Ensemble learning. In: *Ensemble Machine Learning*. Springer, pp. 1–34.
- Schapire, R.E., 1990. The strength of weak learnability. *Machine Learning* 5 (2), 197–227.
- Schapire, R.E., Freund, Y., 2012. *Boosting: Foundations and Algorithms*. MIT press.
- Valiant, L.G., 1984. A theory of the learnable. *Communications of the ACM* 27 (11), 1134–1142.

Multiple Learners Combination: Stacking

Chiara Zucco, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Thanks to the development of more advanced high-throughput technologies, the last decades have seen the data extracted from the omic sciences become more and more numerous to assume the characteristics of “big data”. This is one of the reasons why Machine Learning and Data Mining have become an increasingly used ally in order to analyze this kind of data, especially for prediction and classification tasks.

Of course, the more accurate the models are the better the prediction will be. Unfortunately, there not exist an “oracle” algorithm, always outperforming other learning algorithm in free domains problems (Wolpert and Macready, 1997). Ensemble learning processes raise from the idea that in order to reach better performances, a set of multiple “weak” learner could be combined to form a “strong” learner (Kuncheva, 2004; Polikar, 2012; Alpaydin, 2014).

Stacked generalization, usually referred as Stacking, is an ensemble technique proposed in ‘92 by Wolpert (1992).

Although Stacking is less widely used than Bagging and Boosting, it was also applied in Bioinformatics context. For example in Wang *et al.* (2006), support vector machine and instance-based classifiers were combined using stacked process to predict membrane protein types based on pseudo-amino acid composition. Other application can be found in He *et al.*, (2013), where a stacking process was used for the automatic extraction of Drug-Drug interaction from biomedical literature and to build a decision-based fusion models for the classification of mutation for breast and ovarian cancer in (Sehgal *et al.*, 2004).

In Stacking process, the ensemble learner is generally built by combining models of different types. The procedure can be described as follows:

1. Divide the training set into two disjoint sets,
2. Train different basic “learners” on the first part,
3. Test the basic learnings on the second part,
4. Using the predictions of (3) as input and correct answers as output instruct a higher level learner.

Stacking can be considered an evolution of cross-validation that we’ll see more in details in the following subsection.

Background and Fundamentals

Before discussing the details of the stacking approach in combining multiple learner, in this Section attention is drawn to some preliminary concepts useful in the following, and the basis of constructing an ensemble of machine learning algorithms will be discussed.

Cross-Validation

It’s worth to remember that, in addition to error inducted by noise in data, two are the main variables that can affect the performances of a model inducted by a machine learning algorithm:

- Bias, that is, the measure of how the prediction model differs from the real value.
- The variance, i.e., the measure of how much the model is sensitive to slight changes in the training set.

It can be shown the bias and variance respond to a same relationship, called Bias-variance trade-off.

Models with low bias and high variance have overfitting problems to data while models with high bias and low variance underfit data. It is therefore clear that an important step for constructing a learning model is to estimate its performance.

Cross-validation includes a set of techniques used for the validation of a model or, in other words, to estimate the generalization performance of the considered model inducted by a given machine learning algorithm, taking into account that:

- A greater number of examples can allow to reduce the error rate but could lead to an overfitting model.
- A lower number of examples can lead to an underfitting model.

The goal of the various cross-validation approaches is therefore to split the original dates by defining: A subset on which the algorithm is trained (training set), a set on which the algorithm is tested (test set) and, if for example the model has parameters to be evaluated, a set to evaluate precisely how to set parameters optimally (validation set).

Because of its simplicity, the most used technique is the holdout approach, that is the random partition of the original dataset into training set, test set and, eventually, validation set.

One of the biggest drawbacks of the hold-out approach is that the estimate of generalization performance strongly depends on the choice of the training and the test set.

Cross-validation k-fold can be seen as an iterative extension of the holdout method. In particular in the k-fold approach, the original $\mathcal{D}$ dataset having N instances is partitioned into k subsets through a random sampling without replacement. Then the learning algorithm is trained on $k-1$ data sets and the remaining i -th set is used as a test set. Finally, to obtain an estimate that does not depend on the resampling of data, the average performance of the models is considered.

The most used variant of the k-fold cross validation are:

- Ten-fold cross-validation, in which $k=10$,
- Leave-one-out cross-validation, in which $k=n$ and so for each iteration the learning algorithm is trained on $N-1$ instances and tested on the remaining one.

Ensemble Learning

As stated in the introduction, the ensemble learning methods were introduced in order to induce a model with better performances or, in other words to build a “strong” learner by combining multiple “weak” or base learners, i.e., learners having an error rate that is slightly lower than the random guessing.

In fact, the theoretical basis shows that the model induced by more weak and independent learners combined by some suitable function (for example majority voting), can lead to an error rate that is lower than the error of each single base-learner (Caritat and de Concorcet, 1785).

Without going into detail, the concept of independence between learners can be made explicit by asking for the base-learners’ errors not to be correlated or, in other terms, that different base-learners tend to produce different errors. Unfortunately, there is a trade-off between diversity and accuracy and a key role for this problem is played by the choice of an suitable combining function (Kuncheva, 2004; Valentini and Masulli, 2002).

In summary, two are the main step that have to be considered in order to build a good ensemble:

- Build the ensemble in such a way that base-learners will be diverse and accurate,
 - Combining each base-learner result in a suitable way that can improve accuracy, i.e., lowering bias and/or variance.
- Diversity can be promoted in an implicit way, for example by training the same learning algorithm on different subset of the training set (variation in data), or explicitly choosing different learning algorithm (variation among learners) (Brazdil et al., 2008). Moreover, there are multiple choices for the combining function. Given N learners, i an instance, d_j the output of the j -th learner, $\mathcal{L}$ a class label set and indicating with d_{ens} the ensemble learner’s output and with $d_{j,c}(i) = \chi_{\text{mathcall}}(d_j(i) = c)$, with $c \in \text{mathcall}$, some examples for combining the base learners’ output are listed below:
- Combining function for class labels:
 - Majority voting: $d_{ens}(i) = \underset{c}{\operatorname{argmax}} \sum_j 1^N d_{j,c}(i)$
 - Borda count Van Erp and Schomaker (2000)
 - Combining function for numerical output:
 - Mean rule: $d_{ens}(i) = \frac{1}{N} \sum_j 1^N d_j(i)$
 - Weighted average $d_{ens}(i) = \frac{1}{N} \sum_j 1^N w_j d_j(i)$, for w_j be such that $\sum_{j=1}^N w_j = 1$
 - Minimum rule $d_{ens}(i) = \min_{j \in [1 \dots N]} d_j(i)$
 - Maximum rule $d_{ens}(i) = \max_{j \in [1 \dots N]} d_j(i)$
 - Median rule $d_{ens}(i) = \min_{j \in [1 \dots N]} d_j(i)$
 - Product rule $d_{ens}(i) = \frac{1}{N} \prod_j 1^N w_j d_j(i)$

The Stacking Algorithm

Compared to Bagging and Boosting, in Stacking two are the major differences:

- While Bagging and Boosting harness diversity with variation in data, Stacking exploit diversity among learners,
 - Instead of making use of one among the combining function listed in the previous Section, in Stacking a new learner is applied to a meta-dataset containing the base-learners output.
- In fact Bagging and Boosting, despite their simplicity, effectiveness and popularity, have as a weak point to subtract interpretability to the ensemble model, in the sense that these approach don’t allow to explicitly know which base-learners is likely to be accurate in which part of the feature space. For that reason in stacking two learning level are foreseen:
- At a first level the base-learners are trained and tested on the original dataset with a cross validation approach,
 - At a second level a new learning algorithm (meta-learner) is trained on a new meta-dataset.

More in details, the stacking process is shown in Fig. 1: At first n learning algorithms ($L_1, \dots, L_n$) are considered and trained on a partition of the dataset $\mathcal{D}$ obtained by applying cross validation to the original dataset. After this training phase, a series of models ($M_1, \dots, M_n$) are produced and tested with the remaining partition of the original dataset. Then, a new meta-dataset $\mathcal{D}_{\text{meta}}$ is built by adding or replacing each base-learner’s prediction for each instance i in the set used from testing the base-learners.

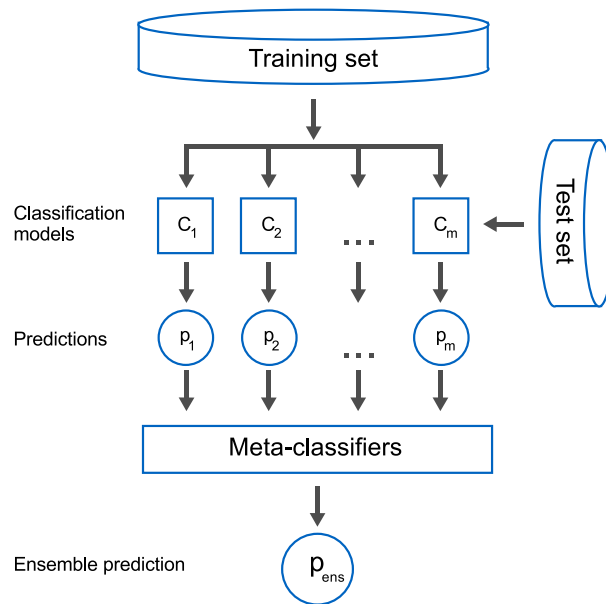


Fig. 1 The Stacking algorithm.

As in cross validation, the advantage of building the meta-dataset from the test set is that this new dataset reflects in some way the real performances of each base-learner, excluding the possibility of memorizing the dataset.

The “meta-dataset” become the training set for the meta-learner algorithm in order to build a new meta-model with the aim of mapping the prediction of each base-learner to a final output, for example a class label in a classification problem.

Then, for each new instance i , all the base-learners’ induced models make a prediction building the corresponding instance i_{meta} in the meta-dataset $\mathcal{D}_{meta}$ and then is sent to the meta-model in order to produce the final prediction for i .

Unlike a fixed rule, like majority voting, the presence of a meta-learner has the advantage of adding flexibility and lowering the bias, with the risk, however, of increasing variance. Another disadvantage is to increase the complexity in terms of time and need a larger dataset for a further training phase.

However, the main weakness of this technique is that there is not an accepted best way for doing stacking, because it’s not suitable for theoretical analysis. (Witten *et al.*, 2016). One of the major issue is what can be the most suitable algorithm used to construct the meta-model. Good results have been given by using linear meta-models (Wolpert, 1992). Stacking were also applied to linear regression problem (Breiman, 1996), while in (Ting and Witten, 1997), was shown that an improvement in performance is achieved by using the class probabilities rather than the individual outputs for each base-learner and a multi-response least-squares algorithm as meta-learner algorithm.

See also: Kernel Machines: Applications. Multiple Learners Combination: Introduction

References

- Alpaydin, E., 2014. Introduction to Machine Learning. MIT press.
- Brazdil, P., Carrier, C.G., Soares, C., Vilalta, R., 2008. Meta Learning: Applications to Data Mining. Springer Science & Business Media.
- Breiman, L., 1996. Stacked regressions. *Machine Learning* 24 (1), 49–64.
- Caritat, M.J.A.N., de Concorcet, M., 1785. Essai sur l’application de l’analyse à la probabilité des décisions: Rendues à la pluralité des voix. De l’imprimerie Royale.
- He, L., Yang, Z., Zhao, Z., Lin, H., Li, Y., 2013. Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach. *PLOS ONE* 8 (6), e65814.
- Kuncheva, L.I., 2004. Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons.
- Polikar, R., 2012. Ensemble learning. In: *Proceedings of the Ensemble Machine Learning*, pp. 1–34. Springer.
- Sehgal, M.S.B., Gondal, I., Dooley, L., 2004. Support vector machine and generalized regression neural network based classification fusion models for cancer diagnosis. In: *Proceedings of 4th International Conference on Hybrid Intelligent Systems (HIS’04)*, pp. 49–54. IEEE.
- Ting, K.M., Witten, I.H., 1997. Stacked generalization: When does it work?
- Valentini, G., Masulli, F., 2002. Ensembles of learning machines. In: *Proceedings of the Italian Workshop on Neural Nets*, pp. 3–20. Springer.
- Van Erp, M., Schomaker, L., 2000. Variants of the borda count method for combining ranked classifier hypotheses. In: *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition. Amsterdam Learning Methodology Inspired by Humans Intelligence Bo Zhang, Dayong Ding, and Ling Zhang*. Citeseer.
- Wang, S.-Q., Yang, J., Chou, K.-C., 2006. Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition. *Journal of Theoretical Biology* 242 (4), 941–946.
- Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.
- Wolpert, D.H., 1992. Stacked generalization. *Neural Networks* 5 (2), 241–259.
- Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1 (1), 67–82.

Multiple Learners Combination: Cascading

Chiara Zucco, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

With the development of high-throughput experimental technologies, in the last decades the omics sciences have seen a huge increase of data in biological databases, both in terms of numbers and in terms of complexity, to the point of considering bioinformatic data as “big data”. Machine learning and Data Mining approach have proved to play a key role in analyzing these vast amount of data and are widely used, especially for prediction and classification tasks.

In order to improve the prediction performances, accurate models are needed. The No free lunch Theorems (Wolpert and Macready, 1997) state that in free domains context, it doesn't exist an algorithm always outperforming another learning algorithm, even if we consider the random guessing algorithm as basis for comparison.

A possible solution that arise as an alternative option for looking for the algorithm that, in that context, induces the “optimal” model, is the idea that a combination of multiple “weak” learners, inducing more simple model, could approximate this optimal model allowing to reach better performances than a single weak or base learner. This kind of approaches are addressed as ensemble learning (Kuncheva, 2004; Polikar, 2012; Alpaydin, 2014; Valentini and Masulli, 2002).

In this article the Cascading process is presented, developed by Alpaydin and Kaynak (1998). As a first application, cascading was used for the handwriting digit recognition and compared to other ensemble approaches with positive results, see Alpaydin *et al.* (2000). Moreover, one of the most popular application of cascading can be found in the Viola-Jones algorithm for a real-time face recognition framework. In this work, visual features are extracted thanks to a particular image representation, called integral image, then features were selected using another ensemble learning approach discussed in a previous article, that is the Adaptive Boosting algorithm, or Adaboost. Adaboost algorithm is also used in order to train classifiers that are combined using the cascading approach.

A significant example of how cascading can be used for the construction of a gene expression data classification system related to lung cancer is presented by Iakovidis *et al.* (2004). Given N possible class labels, the system is built by combining in a cascade fashion $N - 1$ SVM classifiers. The system was then applied to a six classes classification problem of lung cancer data. Results shows an estimated accuracy of 98.5%.

Background and Fundamentals

Before discussing the details of the cascading approach in combining multiple learner, in this Section the basis of constructing an ensemble of machine learning algorithms will be summarised.

Ensemble Learning

As stated in the Introduction, one of the major implications of the No Free Lunch theorems is that, given two learning algorithms L_1 and L_2 , there are (on average) as many situations in which L_1 induced model fits data better than L_2 model as there are situations in which L_2 model fits data better than L_1 and this is also true if it's included the random guessing algorithm. In other words, for each algorithm there are many situations in which it induces a model that describes well observed data, and many other cases in which the model induced by that same algorithm fails.

Ensemble learning methods were introduced as a way to induce a model with better performances or, in other words, to build a “strong” learner by combining multiple “weak” or base learners, i.e., learners having an error rate that is slightly lower than the random guessing.

The basic principle of ensemble methods found many analogies in real life, let's think for examples to democratic processes. It's not a case that the theoretical basis for ensemble learning models can be found in a political science Theorem, known as the “Condorcet's Jury's Theorem”, showing that the model induced by more weak and independent learners combined by some suitable function (for example majority voting), can lead to an error rate that is lower than the error of each single base-learner, see Caritat and de Concorcet (1785).

Without going into detail, the concept of independence between learners can be made explicit by asking that different base-learners tend to produce errors on different instances (errors have to be uncorrelated). Unfortunately, the more the more learners are accurate, the less they are diverse, in the sense that the correlation between errors increases. In order to find a trade-off between diversity and accuracy, the choice of a suitable combining function plays a key role (Kuncheva, 2004; Valentini and Masulli, 2002).

The main issues to be considered in order to build a good ensemble are:

- To build the ensemble by choosing sufficiently accurate base learners and by encouraging diversity among base-learners;
- Combining each base-learner result in a suitable way that can improve accuracy, i.e., lowering bias and/or variance.

Diversity can be promoted both in an implicit way and in an explicit way. For example in Bagging algorithms the same learning algorithm is trained in parallel on different subsets of the training set and the diversity of the base learners is achieved by applying a variation in training data. Also in Boosting diversity is achieved by using the same learning algorithm on a different subset of the training data, but in this case the base-learners are trained in a sequence fashion on a subset containing also the instances misclassified by the previous learners. On the contrary, in stacking the ensemble is built by explicitly choosing different learning algorithms and training them in parallel on the same training set (for further details see Brazdil et al., 2008).

Moreover, there are multiple choices for the combining function. Let consider N learners, an instance i and the output d_j of the j -th learner. Furthermore, indicating with $\mathcal{L}$ a set of class labels, with d_{ens} the output of the ensemble classifier and with $d_{j,c}(i) = \chi_{\text{mathcall}}(d_j(i) = c)$ the characteristic function for $c \in \text{mathcall}$, some examples for combining the base learners' output are listed below:

- Combining function for class labels:
 - Majority voting: $d_{ens}(i) = \text{argmax} \sum_{j=1}^N d_{j,c}(i)$
 - Borda count (Van Erp and Schomaker, 2000)
- Combining function for numerical output:
 - Mean rule: $d_{ens}(i) = \frac{1}{N} \sum_{j=1}^N d_j(i)$
 - Weighted average $d_{ens}(i) = \frac{1}{N} \sum_{j=1}^N w_j d_j(i)$, for w_j be such that $\sum_{j=1}^N w_j = 1$
 - Minimum rule $d_{ens}(i) = \min_{j \in [1 \dots N]} d_j(i)$
 - Maximum rule $d_{ens}(i) = \max_{j \in [1 \dots N]} d_j(i)$
 - Median rule $d_{ens}(i) = \text{median}_{j \in [1 \dots N]} d_j(i)$
 - Product rule $d_{ens}(i) = \frac{1}{N} \prod_{j=1}^N w_j d_j(i)$

It can be proven that majority voting can lower the variance of the model without having strong effects on bias and for this reason is used as combining function in Bagging, because Bagging is more effective when using as learning algorithm an instable one. The weak point of these combining functions is that the resulting ensemble model has a loss in interpretability, because the information about which learner seems to be more accurate on a given part of the considered feature space is lost. Instead, in Stacking the combining function is a trainable arbiter, i.e., a new learning algorithm called meta-learner.

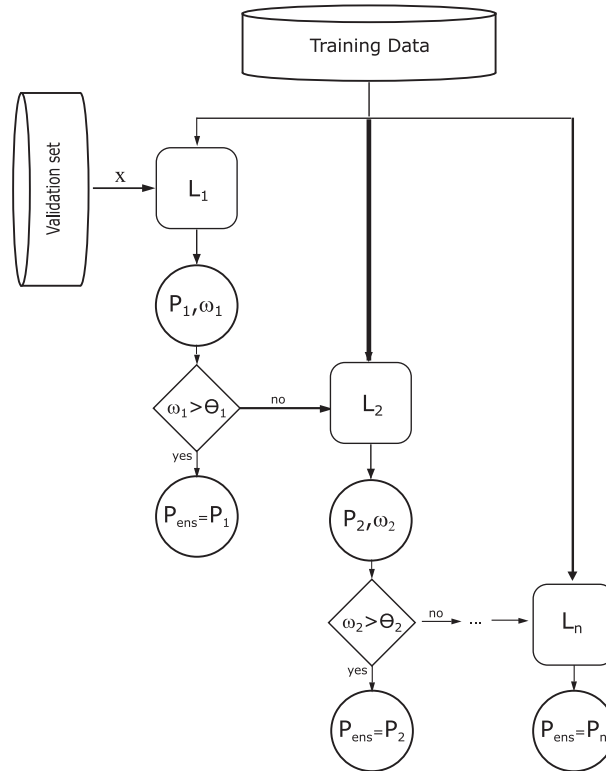


Fig. 1 The Cascading scheme for combining n Learners' predictions.

The Cascading Algorithm

Cascading is an ensemble method that uses different learning algorithms by combining them in sequence. In particular, in fact, in this case the base learners are ordered in an increasing way according to their spatial or temporal complexity or even on the basis of the cost of the representation used by the algorithm. Let's consider a classification problem. Unlike majority voting, for which each base-classifier's output is considered, in cascading, the "simpler" classifiers are those that are used to classify most of the instances, while more complex classifiers will be only used when the simplest ones do not give an output with certain confidence. In this sense, Cascading can be seen as a multistage method.

In the previous section we saw how Bagging, Boosting and Stacking mainly focus on the output of individual base-learners, which are then combined in some way. The main difference in Cascading is that, instead of considering only the base-learner's output, to each learner $\mathcal{L}_j$ is associated a confidence function ω_j and, as long as the base-learner's confidence on a given instance x is lower than a set threshold, then a following base-learner is used.

More in details, let's consider the initial dataset partitioned through a cross validation approach in a Training set T and in a Validation set V , a sequence $L_1 \dots L_n$ of classifiers and a sequence $0 < \theta_1 < \theta_n < 1$ of threshold values.

In the first step of cascading, the first classifier L_1 of the sequence is trained on the training set T and by considering on T a uniform distribution of the instances, as in the case of boosting. For the classifier L_1 and, more in general, for every classifier L_j it is possible to determine the posterior probability that the instance x is assigned to the class $c_k \in \{c_1 \dots c_m\}$, i.e.,

$$\rho_{k,j} = \mathcal{P}(c_k | x, L_j)$$

and it is also possible to assign a confidence function $\omega_j(x)$ of the classifier that is defined as

$$\omega_j(x) = \max_{c_k \in \{c_1 \dots c_m\}} \mathcal{P}(c_k | x, L_j) = \max_{c_k \in \{c_1 \dots c_m\}} \rho_{k,j}$$

If the x instance is predicted by the classifier L_1 with a confidence $\omega_1(x) > \theta_1$, then the output of L_1 will be accepted as the output of the ensemble L_{ens} , otherwise the output of the L_1 classifier will be rejected and then the instance must be classified by one of the following classifiers.

Technically, the subsequent classifiers L_{j+1} will be trained on the same dataset T having however a different distribution that depends on the confidence function ω_j of the previous classifier. In general the output of the ensemble for that instance will be given by

$$P_{ens}(x) = \arg \max_{c_k \in \{c_1 \dots c_m\}} \rho_{k,j}(x), \text{ if } \omega_j(x) > \theta_j \text{ and } \forall i < j, \omega_i(x) < \theta_i.$$

In **Fig. 1** is shown the process just described.

See also: Kernel Machines: Applications. Multiple Learners Combination: Introduction

References

- Alpaydin, E., 2014. Introduction to Machine Learning. MIT Press.
- Alpaydin, E., Kaynak, C., 1998. Cascading classifiers. *Kybernetika* 34 (4), 369–374.
- Alpaydin, E., Kaynak, C., Alimoglu, F., 2000. Cascading multiple classifiers and representations for optical and pen-based handwritten digit recognition. In: *Proceedings of International Workshop on Frontiers in Handwriting Recognition (IWFHR 00)*. Amsterdam, The Netherlands.
- Brazdil, P., Carrier, C.G., Soares, C., Vilalta, R., 2008. *Metalearning: Applications to Data Mining*. Springer Science & Business Media.
- Caritat, M.J.A.N., de Concorcet, M., 1785. *Essai sur l'application de l'analyse a la probabilité des décisions: Rendues a la pluralité des voix*. De l'imprimerie royale.
- Iakovidis, D.K., Flaounas, I.N., Karkanis, S.A., Maroulis, D.E., 2004. A cascading support vector machines system for gene expression data classification. In: *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, pp. 344–347.
- Kuncheva, L.I., 2004. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons.
- Polikar, R., 2012. Ensemble learning. In: *Ensemble Machine Learning*. Springer, pp. 1–34.
- Valentini, G., Masulli, F., 2002. Ensembles of learning machines. In: *Proceedings of Italian Workshop on Neural Nets*. Springer, pp. 3–20.
- Van Erp, M., Schomaker, L., 2000. Variants of the borda count method for combining ranked classifier hypotheses. In: *Proceeding of the Seventh International Workshop on Frontiers in Handwriting Recognition*. Amsterdam Learning Methodology Inspired by Humans Intelligence Bo Zhang, Dayong Ding, and Ling Zhang. Citeseer.
- Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1 (1), 67–82.

Cross-Validation

Daniel Berrar, Tokyo Institute of Technology, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Cross-validation is a data resampling method to assess the generalization ability of predictive models and to prevent overfitting (Hastie *et al.*, 2008; Duda *et al.*, 2001). Like the bootstrap (Efron and Tibshirani, 1993), cross-validation belongs to the family of Monte Carlo methods.

Consider a data set D , which consists of n labeled instances (or cases), x_i , $i = 1 \dots n$. Each case is described by a set of attributes (or features). We assume that each case x_i belongs to exactly one class y_i . A typical example from bioinformatics is a gene expression data set based on DNA microarray data, where each case represents one labeled tumor sample described by a gene expression profile. One of the common challenges concerns the development of a classifier that can reliably predict the class of new, unseen tumor samples based on their expression profiles (Berrar *et al.*, 2007). Conceptually, a predictive model, $f(\cdot)$, is a rule for assigning a class label to a case based on a data set D , i.e., $f(x_i, D) = \hat{y}_i$, where $\hat{y}_i$ is the predicted class label for case x_i . In machine learning, the construction of such a model is denoted as *supervised learning*.

A central question in supervised learning concerns the accuracy of the resulting model. Here, a key problem is *overfitting* (Berrar and Dubitzky, 2013). It is very easy to build a model that is perfectly adapted to the data set at hand but then unable to generalize well to new, unseen data. For example, consider a univariate regression problem where we wish to predict the dependent variable y from the independent variable x based on n observations, (x_i, y_i) , with $i = 1 \dots n$. We could use a polynomial of degree $n - 1$ to interpolate these points perfectly and then use the resulting curve to extrapolate the value y_{i+1} for a new case, x_{i+1} . However, this curve is very likely to be *overfitted* to the data at hand – not only does it reflect the relation between the dependent and independent variable, but it has also modeled the inherent noise in the data set. On the other hand, a simpler model, such as a least squares line, is less affected by the inherent noise, but it may not capture well the relation between the variables, either. Such a model is said to be *underfitted*. Neither the overfitted nor the underfitted model are expected to generalize well, and a major challenge is to find the right balance between over- and underfitting.

How can we assess the generalization ability of a model? Ideally, we would evaluate the model using new data that originate from the same population as the data that we used to build the model (Simon, 2003). In practice, new independent validation studies are often not feasible, though. Also, before we invest time and other resources for an external validation, it is advisable to estimate the predictive performance first. This is usually done by *data resampling methods*, such as cross-validation. This article describes the major subtypes of cross-validation and their related resampling methods.

Basic Concepts and Notation

The data set that is available to build and evaluate a predictive model is referred to as the *learning set*, D_{learn} . This data set is assumed to be a sample from a population of interest. Random subsampling methods are used to generate training set(s), D_{train} , and test set(s), D_{test} , from the learning set. The model is then built (or trained) using the training set(s) and tested on the test set(s). The various random subsampling methods differ with respect to how the training and test sets are generated.

Note that the term “training” implies that we apply the learning algorithm to a subset of the data. The resulting model, $\hat{f}(x, D_{\text{train}})$, is only an estimate of the final model that results from applying the same learning function to the entire learning set, $f(x, D_{\text{learn}})$. Model evaluation based on repeated subsampling means that a learning function is applied to several data subsets, and the resulting models, $\hat{f}_j$, are subsequently evaluated on other subsets (i.e., the test sets or validation sets), which were not used during training. The average of the performance that the models $\hat{f}_j$ achieve on these subsets is an estimate of the performance of the final model, $f(x, D_{\text{learn}})$.

Let us assume that with each case, exactly one target label y_i is associated. In the case of classification, y_i is a discrete class label. A classifier is a special case of a predictive model that assigns a discrete class label to a case. In regression tasks, the target is usually a real value, $y_i \in \mathbb{R}$. A predictive model $f(\cdot)$ estimates the target y_i of the case x_i as $f(x_i) = \hat{y}_i$. A *loss function*, $\mathcal{L}(y_i, \hat{y}_i)$, quantifies the estimation error. For example, using the 0-1 loss function in a classification task, the loss is 1 if $y_i \neq \hat{y}_i$ and 0 otherwise. With the loss function, we can now calculate two different errors, (1) the *training error* and (2) the *test error*. The training error (or resubstitution error) tells us something about the adaptation to the training set(s), whereas the test error is an estimate of the true prediction error. This estimate quantifies the generalization ability of the model. Note that the training error tends to underestimate the true prediction error, since the same data that were used to train the model are reused to evaluate the model.

Single Hold-Out Random Subsampling

Among the various data resampling strategies, one of the simplest ones is the *single hold-out method*, which randomly samples some cases from the learning set for the test set, while the remaining cases constitute the training set. Often, the test set contains about

10%–30% of the available cases, and the training set contains about 90%–70% of the cases. If the learning set is sufficiently large, and consequently, if both the training and test sets are large, then the observed test error can be a reliable estimate of the true error of the model for new, unseen cases.

***k*-fold Random Subsampling**

In *k*-fold random subsampling, the single hold-out method is repeated k times, so that k pairs of $D_{\text{train},j}$ and $D_{\text{test},j}$, $j=1\dots k$, are generated. The learning function is applied to each training set, and the resulting model is then applied to the corresponding test set. The performance is estimated as the average over all k test sets. Note that any pair of training and test set is disjoint, i.e., the sets do not have any cases in common, $D_{\text{train},j} \cap D_{\text{test},j} = \emptyset$. However, any given two training sets or two test sets may of course overlap.

***k*-fold Cross-Validation**

Cross-validation is similar to the repeated random subsampling method, but the sampling is done in such a way that no two test sets overlap. In *k*-fold cross-validation, the available learning set is partitioned into k disjoint subsets of approximately equal size. Here, *fold* refers to the number of resulting subsets. This partitioning is performed by randomly sampling cases from the learning set *without* replacement. The model is trained using $k-1$ subsets, which, together, represent the training set. Then, the model is applied to the remaining subset, which is denoted as the *validation set*, and the performance is measured. This procedure is repeated until each of the k subsets has served as validation set. The average of the k performance measurements on the k validation sets is the *cross-validated performance*. Fig. 1 illustrates this process for $k=10$, i.e., 10-fold cross-validation. In the first fold, the first subset serves as validation set $D_{\text{val},1}$ and the remaining nine subsets serve as training set $D_{\text{train},1}$. In the second fold, the second subset is the validation set and the remaining subsets are the training set, and so on.

The cross-validated accuracy, for example, is the average of all ten accuracies achieved on the validation sets. More generally, let $\hat{f}_{-k}$ denote the model that was trained on all but the k th subset of the learning set. The value $\hat{y}_i = \hat{f}_{-k}(x_i)$ is the predicted or estimated value for the real class label, y_i , of case x_i , which is an element of the k th subset. The cross-validated estimate of the prediction error, $\hat{e}_{\text{cv}}$, is then given as

$$\hat{e}_{\text{cv}} = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y_i, \hat{f}_{-k}(x_i)) \quad (1)$$

Cross-validation often involves *stratified* random sampling, which means that the sampling is performed in such a way that the class proportions in the individual subsets reflect the proportions in the learning set. For example, suppose that the learning set contains $n=100$ cases of two classes, the positive and the negative class, with $n_+=80$ and $n_-=20$. If random sampling is done without stratification, then it is quite possible that some validation sets contain only positive cases (or only negative cases). With stratification, however, each validation set in 10-fold cross-validation is guaranteed to contain about eight positive cases and two negative cases, thereby reflecting the class ratio in the learning set. The underlying rationale for stratified sampling is the following. The sample proportion is an unbiased estimate of the population proportion. The learning set represents a sample from the population of interest, so the class ratio in the learning set is the best estimate for the class ratio in the population. To avoid a biased evaluation, data subsets that are used for evaluating the model should therefore also reflect this class ratio. For real-world data sets, Kohavi recommends stratified 10-fold cross-validation (Kohavi, 1995).

To reduce the variance of the estimated performance measure, cross-validation is sometimes repeated with different k -fold subsets (*r times repeated k-fold cross-validation*). However, Molinaro et al. (2005) showed that such repetitions reduce the variance only slightly.

For the comparison of two different classifiers in cross-validation, the variance-corrected *t*-test was proposed (Nadeau and Bengio, 2003; Bouckaert and Frank, 2004). Note that the training sets in the different cross-validation folds overlap, which violates

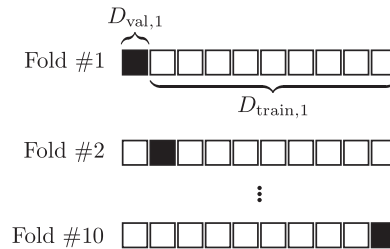


Fig. 1 10-fold cross-validation. The data set is randomly split into ten disjoint subsets, each containing (approximately) 10% of the data. The model is trained on the training set and then applied to the validation set.

the independence assumption of the standard t -test and leads to an underestimation of the variance. The variance-corrected t -statistic is

$$T = \frac{\frac{1}{kr} \sum_{i=1}^k \sum_{j=1}^r (a_{ij} - b_{ij})}{\sqrt{\left(\frac{1}{kr} + \frac{n_2}{n_1}\right) s^2}} \sim t_{kr-1} \quad (2)$$

This statistic follows approximately Student's t distribution with $k \cdot r - 1$ degrees of freedom. Here, a_{ij} and b_{ij} denote the performances achieved by two competing classifiers, A and B , respectively, in the j th repetition of the i th cross-validation fold; s^2 is the variance; n_2 is the number of cases in one validation set, and n_1 is the number of cases in the corresponding training set. This test should be used carefully, though; by increasing r , even a tiny difference in performance can be made significant, which is misleading because essentially the same data are analyzed over and over again (Berrar, 2017).

Leave-One-Out Cross-Validation

For $k=n$, we obtain a special case of k -fold cross-validation, called *leave-one-out cross-validation* (LOOCV). Here, each individual case serves, in turn, as hold-out case for the validation set. Thus, the first validation set contains only the first case, x_1 , the second validation set contains only the second case, x_2 , and so on. This procedure is illustrated in Fig. 2 for a data set consisting of $n=25$ cases.

The test error in LOOCV is approximately an unbiased estimate of the true prediction error, but it has a high variance, since the n training sets are practically the same, as two different training sets differ only with respect to one case (Hastie et al., 2008). The computational cost of LOOCV can also be very high for large n , particularly if feature selection has to be performed.

Jackknife

Leave-one-out cross-validation is very similar to a related method, called the *jackknife*. Essentially, these two methods differ with respect to their goal. Leave-one-out cross-validation is used to estimate the generalization ability of a predictive model. By contrast, the jackknife is used to estimate the bias or variance of a statistic, $\hat{\theta}$ (Efron and Stein, 1981). Note that the available data set is only a sample from the population of interest, so $\hat{\theta}$ is only an estimate of the true parameter, θ . The jackknife involves the following steps (Duda et al., 2001).

- (1) Calculate the sample statistic $\hat{\theta}$ as a function of the available cases $x_1, x_2, \dots, x_n$, i.e., $\hat{\theta} = t(x_1, x_2, \dots, x_n)$, where $t(\cdot)$ is a statistical function.
- (2) For all $i=1..n$, omit the i th case and apply the same statistical function $t(\cdot)$ to the remaining $n-1$ cases and obtain $\hat{\theta}_i = t(x_1, x_2, x_{i-1}, x_{i+1}, \dots, x_n)$. (NB: the index i means that the i th case is *not* used.)
- (3) The jackknife estimate of the statistic $\hat{\theta}$ is the average of all $\hat{\theta}_i$, i.e., $\bar{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^n \hat{\theta}_i$.

The jackknife estimates of bias and variance of $\hat{\theta}$ are

$$\text{bias}(\hat{\theta}) = (n-1)(\bar{\hat{\theta}} - \hat{\theta}) \quad (3)$$

$$\text{Var}(\hat{\theta}) = \frac{n-1}{n} \sum_{i=1}^n (\hat{\theta}_i - \bar{\hat{\theta}})^2 \quad (4)$$

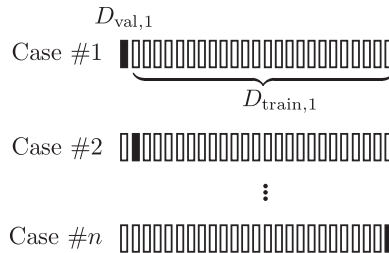


Fig. 2 Leave-one-out cross-validation, illustrated on a data set containing $n=25$ cases. In turn, each case serves as single hold-out test case. The model is built using the remaining $n-1$ cases.

Discussion

Cross-validation is one of the most widely used data resampling methods to assess the generalization ability of a predictive model and to prevent overfitting. To build the final model for the prediction of real future cases, the learning function (or learning algorithm) f is usually applied to the entire learning set. This final model cannot be cross-validated. The purpose of cross-validation in the model building phase is to provide an estimate for the performance of this final model on new data.

Feature selection is generally an integral part of the model building process. Here, it is crucial that predictive features are selected using only the training set, not the entire learning set; otherwise, the estimate of the prediction error can be highly biased (Ambroise and McLachlan, 2002; Simon, 2007). Suppose that predictive features are selected based on the entire learning set first, and then the learning set is partitioned into validation sets and training sets. This means that information from the validation sets was used for the selection of predictive features. But the data in the validation sets serve only to evaluate the model – we are not allowed to use these data in any other way; otherwise, the information leak would cause a downward bias of the estimate, which means that it underestimates the true prediction error.

Cross-validation is frequently used to tune model parameters, for example, the optimal number of nearest neighbors in a k -nearest neighbor classifier. Here, cross-validation is applied multiple times for different values of the tuning parameter, and the parameter that minimizes the cross-validated error is then used to build the final model. Thereby, cross-validation addresses the problem of overfitting.

Which data resampling method should be used in practice? Molinaro *et al.* (2005) compared various data resampling methods for high-dimensional data sets, which are common in bioinformatics. Their findings suggest that LOOCV, 10-fold cross-validation, and the .632+ bootstrap have the smallest bias. It is not clear, however, which value of k should be chosen for k -fold cross-validation. A sensible choice is probably $k=10$, as the estimate of prediction error is almost unbiased in 10-fold cross-validation (Simon, 2007). Isaksson *et al.* (2008), however, caution that cross-validated performance measures are unreliable for small-sample data sets and recommend that the true classification error be estimated using Bayesian intervals and a single hold-out test set.

Closing Remarks

Cross-validation is one of the most widely used data resampling methods to estimate the true prediction error of models and to tune model parameters. Ten-fold stratified cross-validation is often applied in practice. Practitioners should keep in mind, however, that data resampling is no panacea for fully independent validation studies involving new data.

See also: Data Mining: Accuracy and Error Measures for Classification and Prediction. Data Mining: Prediction Methods

References

- Ambroise, C., McLachlan, G.J., 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences* 99 (10), 6562–6566.
- Berrar, D., 2017. Confidence curves: An alternative to null hypothesis significance testing for the comparison of classifiers. *Machine Learning* 106 (6), 911–949.
- Berrar, D., Dubitzky, W., 2013. Overfitting. In: Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H. (Eds.), *Encyclopedia of Systems Biology*. Springer, pp. 1617–1619.
- Berrar, D., Granzow, M., Dubitzky, W., 2007. Introduction to genomic and proteomic data analysis. In: Dubitzky, W., Granzow, M., Berrar, D. (Eds.), *Fundamentals of Data Mining in Genomics and Proteomics*. Springer, pp. 1–37.
- Bouckaert, R., Frank, E., 2004. Evaluating the replicability of significance tests for comparing learning algorithms. *Advances in Knowledge Discovery and Data Mining* 3056, 3–12.
- Duda, R., Hart, P., Stork, D., 2001. *Pattern Classification*. John Wiley & Sons.
- Efron, B., Stein, C., 1981. The jackknife estimate of variance. *The Annals of Statistics* 9 (3), 586–596.
- Efron, B., Tibshirani, R., 1993. *An Introduction to the Bootstrap*. Chapman & Hall.
- Hastie, T., Tibshirani, R., Friedman, J., 2008. *The Elements of Statistical Learning*, second ed. New York/Berlin/Heidelberg: Springer.
- Isaksson, A., Wallman, M., Göransson, H., Gustafsson, M., 2008. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters* 29 (14), 1960–1965.
- Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 2, IJCAI'95*, pp. 1137–1143. Morgan Kaufmann Publishers Inc.
- Molinaro, A., Simon, R., Pfeiffer, R., 2005. Prediction error estimation: A comparison of resampling methods. *Bioinformatics* 21 (15), 3301–3307.
- Nadeau, C., Bengio, Y., 2003. Inference for the generalization error. *Machine Learning* 52, 239–281.
- Simon, R., 2003. Supervised analysis when the number of candidate features (p) greatly exceeds the number of cases (n). *ACM SIGKDD Explorations Newsletter* 5 (2), 31–36.
- Simon, R., 2007. Resampling strategies for model assessment and selection. In: Dubitzky, W., Granzow, M., Berrar, D. (Eds.), *Fundamentals of Data Mining in Genomics and Proteomics*. Springer, pp. 173–186.

Performance Measures for Binary Classification

Daniel Berrar, Tokyo Institute of Technology, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Classification problems can be categorized into (1) binary, (2) multiclass, and (3) multilabel tasks. In binary classification tasks, only two classes are considered, which are commonly referred to as the *positive* and *negative* class; for example, healthy vs. diseased, underexpressed vs. overexpressed, smoker vs. non-smoker, etc. By contrast, multiclass tasks include more than just two classes. Some of the measures for binary classification tasks can be easily extended to multiclass problems (Baldi et al., 2000; Ferri et al., 2009). “Single-label” means that an instance (or case) belongs to only one class, whereas “multi-label” means that an instance can simultaneously belong to more than just one class. This article focuses on performance measures for single-label, binary classification tasks, with the goal to provide an easily accessible introduction to the most commonly used quantitative measures and how they are related. Using a simplified example, we illustrate how to calculate these measures and give some general recommendations regarding their use.

In this article, the term “predictive model” should be understood to refer to not only fully specified models from machine learning; instead, the term also encompasses medical diagnostic tests, for example, a blood sugar test for diabetes.

We will begin with some basic notations. Let a data set D contain n instances (or cases) $\mathbf{x}_i$, $i = 1 \dots n$, and let each instance be described by k attributes (or features or covariates). We assume that each instance belongs to exactly one class y_i , with $y \in \{0, 1\}$, where 1 denotes the positive class and 0 denotes the negative class. A *scoring classifier* is a mapping $C: X \rightarrow \mathbb{R}$ that produces a class membership score for each instance, for example, a conditional probability $P(y = 1 | X = \mathbf{x}_i)$. This class membership score expresses the degree of class membership of that instance in the positive class. Here, we will assume that the scores are scaled from 0 to 1 and that they can be interpreted as estimated class posterior probabilities, $C(\mathbf{x}_i) = p_i = P(y = 1 | X = \mathbf{x}_i)$.

As D contains only positive and negative examples, the scoring classifier can either be used as a *ranker* or as a *crisp classifier*. A ranker uses the ordinal scores to order the instances from the most to the least likely to be positive. The ranker can be turned into a crisp classifier by setting a classification threshold t on the score: if $p_i > t$, then the predicted class label is $\hat{y} = 1$; otherwise, $\hat{y} = 0$.

The underlying concept of performance metrics are *scoring rules*, which assess the quality of probabilistic predictions (Buja et al., 2005; Gneiting and Raftery, 2007; Witkowski et al., 2017). Let a model be presented with an instance $\mathbf{x}_i$, which belongs to either the positive or negative class, $y \in \{0, 1\}$. Let the model's *probabilistic belief* be the same as the true probability $p \in [0, 1]$ that the class of $\mathbf{x}_i$ is $y = 1$. The model outputs the *belief report* $q \in [0, 1]$. A scoring rule $R(y, q) \in \mathbb{R}$ assigns a reward based on the reported q and the real class y . A scoring rule is called *proper* if the model maximizes the expected reward by truthfully reporting its belief p . A scoring rule is called *strictly proper* if the reported belief is the only report that maximizes the expected reward. For example, consider the quadratic scoring rule $R(y, q) = 1 - (y - q)^2$. The (true) probability that the instance $\mathbf{x}_i$ belongs to class 1 is p , and the (true) probability that it belongs to class 0 is $(1 - p)$. The expected reward is then $E(R) = [1 - (1 - q)^2]p + [1 - (0 - q)^2](1 - p) = 1 - q^2 + 2pq - p$. Setting the first derivative with respect to q to zero gives $\frac{\partial R}{\partial q} = 2p - 2q = 0$ or $q = p$. As the second derivative, $\frac{\partial^2 R}{\partial q^2} = -2$, is negative, the reward is indeed (uniquely) maximized if $q = p$. Therefore, the model is incentivized to report the true probability, p . The quadratic scoring rule is a strictly proper scoring rule and underlies various performance measures, for example, the Brier score (Definition 18).

As described in (Ferri et al., 2009), the different performance measures can be categorized into three broad families:

1. Performance measures based on a single classification threshold;
 - (a) Elementary performance measures;
 - (b) Composite performance measures;
2. Performance measures based on a probabilistic interpretation of error;
3. Ranking measures.

We will illustrate the performance measures using a contrived example (Fig. 1). Here, ten cases are described by two features, i.e., their x - and y -coordinates; five cases (#3, #6, #7, #9, and #10) belong to the positive class (represented by circles), while the five remaining cases (#1, #2, #4, #5, and #8) belong to the negative class (represented by squares). The classification task is to find a decision boundary, so that cases falling on one side are classified as members of the positive class, while cases falling on the opposite side are classified as members of the negative class.

In Fig. 1, the decision boundary is represented by the solid vertical line. Note that this line is certainly not the optimal decision boundary for this classification problem; nonetheless, it can be used to discern the two classes: the more a case is located to the right of the boundary, the more likely it is a member of the positive class, and vice versa. Case #6 lies exactly on the boundary, so it is reasonable to assign a class membership score of 0.5, with the probabilistic interpretation that the case is equally likely to belong to the positive or negative class. To quantify the degree of class membership of the other cases, we calculate the distance between them and the boundary. For example, the distance between case #8 and the boundary is 0.25, which leads to a score of

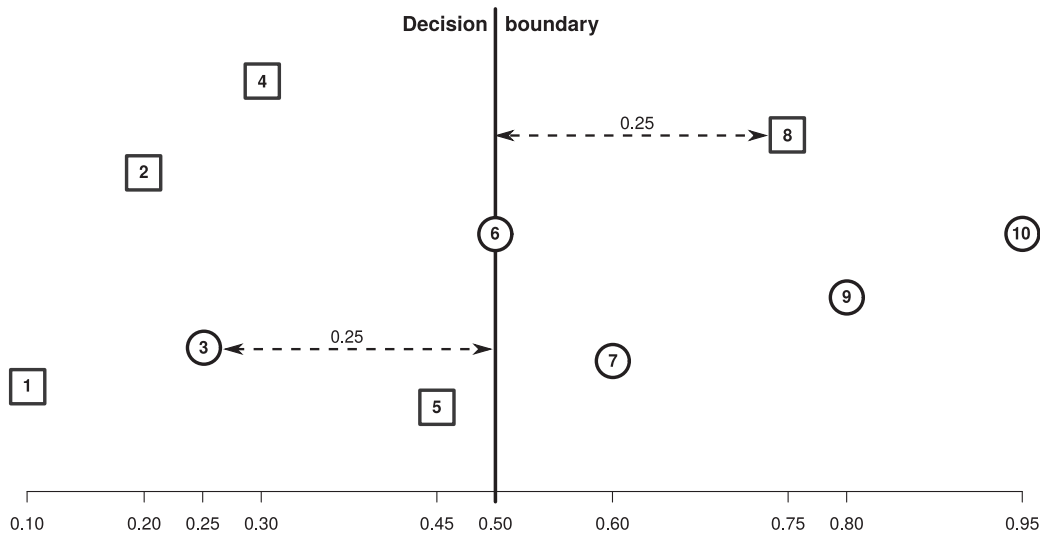


Fig. 1 Simplified example. Ten instances of two classes (circles: positive class; squares: negative class) are classified based on a decision boundary (solid black line). Instances to the right of the boundary are predicted as positives, while instances to the left are predicted as negatives. The distance to the boundary is used as class membership score.

	Case	Class	Score	TPR	FPR	Prec	Cum.class	Lift
t_1	10	1	0.95	1/5	0	1	1	2
t_2	9	1	0.8	2/5	0	1	2	2
t_3	8	0	0.75	2/5	1/5	2/3	2	4/3
t_4	7	1	0.6	3/5	1/5	3/4	3	3/2
t_5	6	1	0.5	4/5	1/5	4/5	4	8/5
•	5	0	0.45	4/5	2/5	4/6	4	4/3
•	4	0	0.3	4/5	3/5	4/7	4	8/7
•	3	1	0.25	5/5	3/5	5/8	5	5/4
	2	0	0.2	5/5	4/5	5/9	5	10/9
t_{11}	1	0	0.1	5/5	5/5	5/10	5	10/10

Fig. 2 Ranking table for the introductory example (Fig. 1), with performance measures resulting from 11 different classification thresholds $t_1 \dots t_{11}$. Each case *above* the threshold is classified as a positive. It is assumed that classification thresholds always fall between actual scores. TPR, true positive rate = recall = sensitivity; FPR, false positive rate = $1 - \text{specificity}$; Prec, precision = positive predictive value; Cum.class, cumulative class count.

$0.5 + 0.25 = 0.75$. Case #3 lies on the opposite side of the boundary but has the same distance, so we use $0.5 - 0.25 = 0.25$ as its membership score for the positive class. Analogously, we can derive the scores for all ten cases and rank them as shown in Fig. 2.

This contrived example is deliberately simplified, and real classification algorithms usually calculate the class membership scores in a more sophisticated way. But the example illustrates the key idea: a model separates positive and negative cases and quantifies their class membership by a score, which can be used to rank the cases from the most likely to be positive to the least likely to be positive.

Note that many performance measures are known under different names. The reason is that the same measures were developed in different fields of science; for example, in epidemiology and medicine, the term “positive predictive value” is widely used, whereas in machine learning and information retrieval, the term “precision” is more common. Similarly, “sensitivity” is commonly used in the context of biomedical tests, whereas “recall” is more common in information retrieval. Mathematically, there is of course no difference between these synonyms.

Performance Measured Based on a Single Classification Threshold

Consider Fig. 2. Here, all cases *above* threshold t_5 (dotted line) are classified as positive cases, while all cases *below* the line are classified as negative cases. Several elementary performance measures can now be derived from such a single classification threshold. The classification results are often represented in a 2×2 table or *confusion matrix*, as shown in Fig. 3(a), with the elementary concepts of true positives (TP, a case is really a positive case and predicted as positive); false positive (FP, a case is really a negative case but predicted as a positive); false negative (FN, a case is really a positive case but predicted as negative); and true negative (TN, a case is really a negative case and predicted as negative). The number of false positives is also known as Type I error,

(a)		Real		
		Positive class	Negative class	
Predicted	Positive class	TP	FP (Type I error)	TP + FP
	Negative class	FN (Type II error)	TN	FN + TN
		TP + FN	FP + TN	TP + FP + FN + TN

(b)		Real		
		Positive class	Negative class	
Predicted	Positive class	3	1	4
	Negative class	2	4	6
		5	5	10

Fig. 3 (a) Confusion matrix for a binary classification task. (b) Confusion matrix for Fig. 2.

and the number of false negatives is known as Type II error. The corresponding counts for the introductory example are shown in Fig. 3(b).

Elementary Performance Measures

From the confusion matrix in Fig. 3(a), several elementary performance measures can be derived.

Definition 1: Accuracy.

The *accuracy* is the proportion of correct classifications,

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

For the introductory example (Fig. 2), the accuracy is $\frac{3+4}{3+1+2+4} = 0.70$ for the classification threshold t_5 .

Definition 2: Error rate.

The *error rate* is the proportion of incorrect classifications,

$$\text{error rate} = \frac{FP + FN}{TP + FP + FN + TN} \quad (2)$$

For the introductory example (Fig. 2), the error rate is $1 - \text{accuracy} = 0.30$ for the classification threshold t_5 .

Definition 3: Sensitivity or recall or true positive rate.

The *sensitivity* (or *recall* or *true positive rate*, TPR) is the number of correctly predicted positive cases divided by the number of all positive cases,

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (3)$$

For the introductory example (Fig. 2), the sensitivity is $\frac{3}{5} = 0.60$. The sensitivity can also be stated as a conditional probability, $P(\hat{y} = 1|y = 1)$.

Definition 4: Specificity or true negative rate.

The *specificity* (or *true negative rate*, TNR) is the number of correctly predicted negative cases divided by the number of all negative cases,

$$\text{specificity} = \frac{TN}{FP + TN} \quad (4)$$

For the introductory example (Fig. 2), the specificity is $\frac{4}{5} = 0.80$. The specificity can also be stated as a conditional probability, $P(\hat{y} = 0|y = 0)$.

Definition 5: Precision or positive predictive value.

The *precision* (or *positive predictive value*) is the number of correctly predicted positive cases divided by the number of all cases that are predicted as positive,

$$\text{precision} = \frac{TP}{TP + FP} \quad (5)$$

For the introductory example (Fig. 2), the precision is $\frac{3}{4} = 0.75$. The precision can also be stated as a conditional probability, $P(y = 1|\hat{y} = 1)$.

Definition 6: Negative predictive value.

The *negative predictive value* is the number of correctly predicted negative cases divided by the number of all cases that are predicted as negative,

$$\text{negative predictive value} = \frac{TN}{TN + FN} \quad (6)$$

For the introductory example (Fig. 2), the negative predictive value is $\frac{4}{6} = 0.67$. The negative predictive value can also be stated as a conditional probability, $P(y = 0|\hat{y} = 0)$.

Definition 7: False discovery rate.

The *false discovery rate* (FDR) is the number of false positives divided by the number of cases that are predicted as positive,

$$\text{FDR} = \frac{FP}{FP + TP} \quad (7)$$

For the introductory example (Fig. 2), the false discovery rate is $\frac{1}{4} = 0.25$. The false discovery rate can also be stated as a conditional probability, $P(y = 0|\hat{y} = 1)$.

Composite Performance Measures Derived From Elementary Measures

From the elementary performance measures, several composite measures can be constructed, such as the Youden index, which is defined as follows (Youden, 1950).

Definition 8: Youden index.

The *Youden index* (or Youden's J statistic) is defined as

$$J = \text{sensitivity} + \text{specificity} - 1 \quad (8)$$

Often, the maximum Youden index is reported, i.e., $J_{\max} = \max_t \{\text{sensitivity}(t) + \text{specificity}(t) - 1\}$, where t denotes the classification threshold for which J is maximal (Ruopp et al., 2008). For the introductory example (Fig. 2), the sum of sensitivity and specificity is maximized for t_6 , for which both sensitivity and specificity are 0.8 (see Fig. 5(a)), and $J_{\max} = 0.6$.

Definition 9: Positive likelihood ratio.

The *positive likelihood ratio* (LR_+) is defined as

$$LR_+ = \frac{\text{sensitivity}}{1 - \text{specificity}} = \frac{P(\hat{y} = 1|y = 1)}{P(\hat{y} = 1|y = 0)} \quad (9)$$

For the introductory example (Fig. 2), the positive likelihood ratio is $\frac{3/5}{1-4/5} = 3.0$.

Definition 10: Negative likelihood ratio.

The *negative likelihood ratio* (LR_-) is defined as

$$LR_- = \frac{1 - \text{sensitivity}}{\text{specificity}} = \frac{P(\hat{y} = 0|y = 1)}{P(\hat{y} = 0|y = 0)} \quad (10)$$

For the introductory example (Fig. 2), the negative likelihood ratio is $\frac{1-3/5}{4/5} = 0.5$.

Definition 11: Balanced accuracy.

The *balanced accuracy* (BACC) is the average of sensitivity and specificity,

$$\text{BACC} = \frac{\text{sensitivity} + \text{specificity}}{2} \quad (11)$$

For the introductory example (Fig. 2), the balanced accuracy is $\frac{1}{2}(\frac{3}{5} + \frac{4}{5}) = 0.70$.

Definition 12: F-measure.

The *F-measure* (also known as F_1 -score or simply F -score) is the harmonic mean of precision and recall,

$$F\text{-measure} = 2 \cdot \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (12)$$

The multiplication by the constant 2 scales the measure to 1 when both precision and recall are 1. For the introductory example (Fig. 2), the F -measure is $2 \cdot \frac{3/4 \cdot 3/5}{3/4 + 3/5} = 0.67$.

Definition 13: F_β -measure.

The F_β -measure is the general form of the F -measure,

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \text{precision} + \text{recall}} \quad (13)$$

where the positive real constant β allows for an unequal weighting of precision and recall.

Definition 14: G-measure.

The G-measure is the geometric mean of precision and recall,

$$\text{G-measure} = \sqrt{\text{precision} \cdot \text{recall}} \quad (14)$$

For the introductory example (Fig. 2), the G-measure is $\sqrt{3/4 \cdot 3/5} = 0.671$ for the classification threshold t_5 .

Matthews correlation coefficient (Matthews, 1975) is a discretization of the Pearson correlation coefficient (Powers, 2011).

Definition 15: Matthews correlation coefficient.

Matthews correlation coefficient (MCC) is defined as

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \quad (15)$$

with $\text{MCC} \in [-1, 1]$, where -1 indicates perfect negative correlation (i.e., the model predicts all negatives as positives, and vice versa), 0 indicates no correlation (i.e., the model predicts randomly), and $+1$ indicates perfect positive correlation (i.e., the model predicts all real positives as positives and all real negatives as negatives).

For the introductory example (Fig. 2), $\text{MCC} = \frac{3 \cdot 4 - 1 \cdot 2}{\sqrt{(3+2)(3+1)(4+1)(4+2)}} = 0.408$. Note that Eq. (15) may lead to the indeterminate form $\frac{0}{0}$, for example, if $TP + FN = 0$, which means that the classifier predicts all cases as instances of the negative class. As the positive class is never predicted, this is most likely an indication that something is wrong with the model. The MCC is suitable for imbalanced data sets (Boughorbel et al., 2017); however, this measure is not easily generalizable to more than two classes (Baldi et al., 2000).

Definition 16: Lift.

The lift measures how much better the predictions by the model, C , are compared to a *baseline* or *null model*. The lift for the positive class is defined as

$$\text{lift}(y = 1) = \frac{P(y = 1 | \hat{y}_C = 1)}{P(\hat{y}_{\text{null}} = 1)} \quad (16)$$

where $P(y = 1 | \hat{y}_C = 1)$ denotes the probability that the case is really a positive, given that the model predicted that it is positive, and $P(\hat{y}_{\text{null}} = 1)$ is the probability that the null model predicts it as a positive.

Commonly, the null model is random guessing, so the probability of predicting a case as a positive is estimated as the proportion of positive cases in the training set, i.e., the prior probability of positive cases (which is also referred to as *prevalence*) (Dubitzky et al., 2001). Put simply, the lift tells us how much better our predictions are when we use our real model, compared to using just random guessing. To illustrate the lift, let us consider again the introductory example (Fig. 3). For the classification threshold t_5 , the model predicts 4 test cases as positives, and 3 of these predictions are correct, hence, $P(y = 1 | \hat{y}_C = 1) = \frac{3}{4}$. Let us assume that the class ratio of positives and negatives is the same in the training set, i.e., half of the cases are positives and the other half are negatives; hence, $P(\hat{y}_{\text{null}} = 1) = \frac{1}{2}$. Therefore, the lift for the positive class is $\frac{3/4}{1/2} = 1.5$. So loosely speaking, we are doing 1.5 times better with the model than with random guessing. This can also be expressed in terms of *gain*: using the model, we expect to predict 3 out of 4 positives correctly, whereas with random guessing, we expect to predict only 2 correctly – hence, we “gain” 1 correct prediction. The lift and gain are usually calculated for all possible classification thresholds and visualized in a *lift chart* and *gain chart*, respectively. These charts are discussed in Section “Ranking Measures”.

Performance Measures Based on a Probabilistic Understanding of Error

Suppose that a model C_1 produces the score $P(y = 1 | \mathbf{x}_-) = 0.9$ for a real negative test case $\mathbf{x}_-$, whereas another model C_2 produces the score $P(y = 1 | \mathbf{x}_-) = 0.8$. Both models misclassify $\mathbf{x}_-$ as a positive case, but which model is making the more serious error? Here, it is useful to consider the deviation of the predicted class posterior probability from the real class label, which is coded as 1 for the positive and 0 for the negative class. Performance measures that take this deviation into account are based on a probabilistic understanding of error.

Definition 17: Mean absolute error.

The *mean absolute error* (MAE) is defined as

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - p_i| \quad (17)$$

where $y_i \in \{0,1\}$ and $p_i = P(y_i = 1 | \mathbf{x}_i) = C(\mathbf{x}_i)$ is the predicted class membership score and n is the number of test cases.

For the introductory example (Fig. 2), the mean absolute error is calculated as $\text{MAE} = \frac{1}{10}(|1 - 0.95| + |1 - 0.80| + |0 - 0.75| + |1 - 0.6| + |1 - 0.5| + |0 - 0.45| + |0 - 0.3| + |1 - 0.25| + |0 - 0.2| + |0 - 0.1|) = 0.370$.

Definition 18: Mean squared error or Brier score.

The *mean squared error* (MSE) or *Brier score* (Brier, 1950) is defined as

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2 \quad (18)$$

where $y_i \in \{0,1\}$ and $p_i = P(y_i = 1 | \mathbf{x}_i) = C(\mathbf{x}_i)$ is the predicted class membership score and n is the number of test cases.

For the introductory example (Fig. 2), $\text{MSE} = \frac{1}{10}[(1 - 0.95)^2 + (1 - 0.80)^2 + (0 - 0.75)^2 + (1 - 0.6)^2 + (1 - 0.5)^2 + (0 - 0.45)^2 + (0 - 0.3)^2 + (1 - 0.25)^2 + (0 - 0.2)^2 + (0 - 0.1)^2] = 0.192$.

Definition 19: Root mean square error.

The *root mean square error* (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2} \quad (19)$$

where $y_i \in \{0,1\}$ and $p_i = P(y_i = 1 | \mathbf{x}_i) = C(\mathbf{x}_i)$ is the predicted class membership score and n is the number of test cases.

For the introductory example (Fig. 2), $\text{RMSE} = \sqrt{0.192} = 0.438$.

The *logarithmic loss* (logloss) or *cross-entropy* is an information-theoretic measure. Note that the Kullback-Leibler divergence is the cross-entropy minus the entropy.

Definition 20: Logloss or cross-entropy.

Let a model C produce scores $p_i \in]0, 1[$ for n instances $\mathbf{x}_i$, e.g., class posterior probabilities $C(\mathbf{x}_i) = P(y_i = 1 | \mathbf{x}_i) = p_i$. The *logarithmic loss* (logloss) or *cross-entropy* is defined as

$$\text{logloss} = -\frac{1}{n} \sum_{i=1}^n y_i \log_2 p_i + (1 - y_i) \log_2 (1 - p_i) \quad (20)$$

where $y_i \in \{0,1\}$, and $p_i \neq 1$ and $p_i \neq 0$.

The smaller the logloss, the better the predictions. If $p_i = 0$ or $p_i = 1$, then the logloss is not defined because of $\log_2 0$; therefore, $\log_2 p_i$ is then calculated as $\log_2(\max\{p_i, \varepsilon\})$, where ε is a small positive constant. Ferri *et al.* suggest $\varepsilon = 10^{-5}$ (Ferri *et al.*, 2009). For the introductory example (Fig. 2), the logloss is calculated as follows.

$$\begin{aligned} \text{logloss} &= -\frac{1}{10} [1 \cdot \log_2(0.95) + (1 - 1) \cdot \log_2(1 - 0.95) \\ &\quad + 1 \cdot \log_2(0.8) + (1 - 1) \cdot \log_2(1 - 0.8) \\ &\quad + 0 \cdot \log_2(0.75) + (1 - 0) \cdot \log_2(1 - 0.75) \\ &\quad + 1 \cdot \log_2(0.6) + (1 - 1) \cdot \log_2(1 - 0.6) \\ &\quad + 1 \cdot \log_2(0.5) + (1 - 1) \cdot \log_2(1 - 0.5) \\ &\quad + 0 \cdot \log_2(0.45) + (1 - 0) \cdot \log_2(1 - 0.45) \\ &\quad + 0 \cdot \log_2(0.3) + (1 - 0) \cdot \log_2(1 - 0.3) \\ &\quad + 1 \cdot \log_2(0.25) + (1 - 1) \cdot \log_2(1 - 0.25) \\ &\quad + 0 \cdot \log_2(0.2) + (1 - 0) \cdot \log_2(1 - 0.2) \\ &\quad + 0 \cdot \log_2(0.1) + (1 - 0) \cdot \log_2(1 - 0.1)] \\ &= 0.798 \end{aligned}$$

A further information-theoretic measure is the *information score* (Kononenko and Bratko, 1991).

Definition 21: Information score and relative information score.

Let the real class of instance $\mathbf{x}_i$ be y_i . Let the prior probability of that class be $p(y_i)$. Let the predicted score for that class be p_i . The *information score* (I) for the case $\mathbf{x}_i$ is defined as

$$I(\mathbf{x}_i) = \begin{cases} -\log_2 p(y_i) + \log_2 p_i & \text{if } p_i \geq p(y_i) \\ \log_2(1 - p(y_i)) - \log_2(1 - p_i) & \text{if } p_i < p(y_i) \end{cases} \quad (21)$$

The *relative information score* (I_r) is the ratio of the average information score over all n test cases $\mathbf{x}_i$ and the entropy of the prior class distribution,

$$I_r = \frac{\frac{1}{n} \sum_{i=1}^n I(\mathbf{x}_i)}{-\sum_{k=1}^K p(y_k) \log_2 p(y_k)} \quad (22)$$

where n denotes the number of test cases, and K denotes the number of classes.

$I(\mathbf{x}_i)$ is positive if $p_i > p(y_i)$, negative if $p_i < p(y_i)$, and zero if $p_i = p(y_i)$. The information score takes into account the class prior probabilities and thereby accounts for the fact that a high accuracy can be easily achieved in tasks with a very likely majority class (Lavrač et al., 1997). For the introductory example, the information score is calculated as follows.

$$I(\mathbf{x}_1) = -\log_2 0.5 + \log_2 0.95 = 0.926$$

$$I(\mathbf{x}_2) = -\log_2 0.5 + \log_2 0.80 = 0.678$$

$$I(\mathbf{x}_3) = \log_2(1 - 0.5) - \log_2(1 - 0.25) = -0.585$$

$$I(\mathbf{x}_4) = -\log_2 0.5 + \log_2 0.60 = 0.263$$

$$I(\mathbf{x}_5) = -\log_2 0.5 + \log_2 0.50 = 0$$

$$I(\mathbf{x}_6) = -\log_2 0.5 + \log_2 0.55 = 0.138$$

$$I(\mathbf{x}_7) = -\log_2 0.5 + \log_2 0.30 = 0.485$$

$$I(\mathbf{x}_8) = \log_2(1 - 0.5) - \log_2(1 - 0.25) = -0.585$$

$$I(\mathbf{x}_9) = -\log_2 0.5 + \log_2 0.80 = 0.678$$

$$I(\mathbf{x}_{10}) = -\log_2 0.5 + \log_2 0.90 = 0.848$$

$$I_r = \frac{\frac{1}{10}(0.926 + 0.678 - 0.585 + 0.263 + 0 + 0.138 + 0.485 - 0.585 + 0.678 + 0.848)}{-0.5 \log_2 0.5 - 0.5 \log_2 0.5} = 0.2846$$

Cohen's kappa measures the agreement between two raters on a categorical variable (Cohen, 1960). Thus, if we assume that the real class labels are due to some unknown generating process, we can use this measure to compare how well the predictions of a model agree with that process (i.e., reality). This agreement is of course already quantified in the accuracy (Eq. (1)); however, accuracy does not take into account that an agreement could be coincidental. Cohen's kappa adjusts the accuracy by calculating the probability of random agreements.

Definition 22: Cohen's kappa.

Cohen's kappa (κ) is defined as

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \quad (23)$$

where P_0 is the probability that the model's predicted class labels agree with the real class labels, and P_e is the probability of random agreement.

P_0 is estimated as the proportion of correctly predicted class labels, i.e., the accuracy. The probability of random agreement is estimated as follows: the probability that the classifier predicts a case as a positive is $P_+^c = \frac{TP+FP}{n}$. The probability that a case is really a positive is $P_+^r = \frac{TP+FN}{n}$. Hence, the probability that both agree just by chance on the positive cases is the product $P_+^c \cdot P_+^r$. It is of course assumed that the process which generates the real class labels is independent of the model. Analogously, we calculate the probability of random agreement on the negative cases, and we obtain the probability of total random agreement as

$$P_e = \frac{TP + FP}{n} \cdot \frac{TP + FN}{n} + \frac{FN + TN}{n} \cdot \frac{FP + TN}{n} \quad (24)$$

where n is the total number of test cases. For the introductory example (Fig. 2), $\kappa = \frac{3+1}{10} \cdot \frac{3+2}{10} + \frac{2+4}{10} \cdot \frac{1+4}{10} = 0.50$.

Ranking Measures

Ranking performance refers to how well the model orders the positive cases relatively to the negative cases based on the class membership score. There exist various graphical methods for representing ranking performance. Fig. 4(a) shows the gain chart of the introductory example (Fig. 2), which plots the cumulative class on the y -axis and the ranking index of the corresponding instance on the x -axis. For example, if we classify the 4 top-ranked instances as positives, then we classify 3 instances correctly, in contrast to only 2 instances that we expect to predict correctly by random guessing; hence, the gain is 1 correct prediction.

In Fig. 4(a), the red line indicates the expected number of correct positive predictions (for a given number of predicted instances) by using random guessing. The *average gain* is a summary statistic of the gain chart.

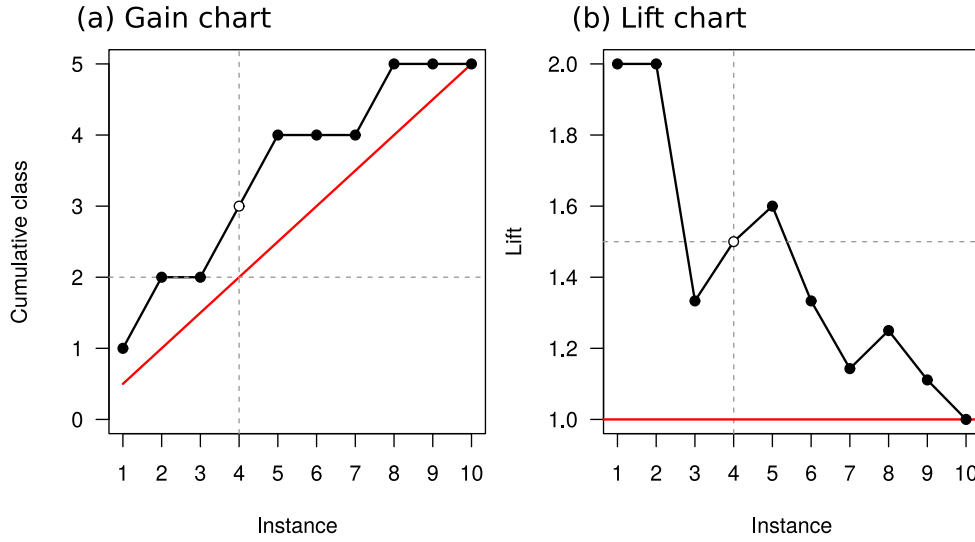


Fig. 4 (a) Gain chart and (b) lift chart for the introductory example (Fig. 2). The red line indicates the expected gain or lift for a random guesser.

Definition 23: Average gain.

Let the test cases be ranked based on decreasing values of their class membership scores $p_i = P(y_i = 1 | \mathbf{x}_i) = C(\mathbf{x}_i)$. Let Δg_j denote the difference between the gain of the trained model C , $g_c(j)$, and the gain of the null model, $g_{\text{null}}(j)$, when the j top-ranked instances are classified as positives,

$$\Delta g_j = g_c(j) - g_{\text{null}}(j) \quad (25)$$

The *average gain* is the average of all differences in gain,

$$\bar{g} = \frac{1}{n} \sum_{j=1}^n \Delta g_j \quad (26)$$

where n is the total number of test cases.

For example, assuming that the null model is random guessing, we calculate the average gain for the introductory example as follows: $\bar{g} = \frac{1}{10}[(1 - 0.5) + (2 - 1) + (2 - 1.5) + (3 - 2) + (4 - 2.5) + (4 - 3) + (4 - 3.5) + (5 - 4) + (5 - 4.5) + (5 - 5)] = 0.75$. Geometrically, the average gain is the average distance between the gain curve and the red line in Fig. 4(a).

Fig. 4(b) shows the corresponding lift chart. Note that the expected lift for random guessing is 1.

Definition 24: Average lift.

Let the test cases be ranked based on decreasing values of the class membership scores $p_i = P(y_i = 1 | \mathbf{x}_i) = C(\mathbf{x}_i)$. Let $\text{lift}(j)$ denote the lift when the j top-ranked cases are classified as positive. The *average lift* is defined as

$$\bar{\text{lift}} = \frac{1}{n} \sum_{j=1}^n \text{lift}(j) \quad (27)$$

where n is the total number of test cases.

For the introductory example, we calculate the average lift as follows (see Fig. 2):

$$\bar{\text{lift}} = \frac{1}{10}(2 + 2 + 4/3 + 3/2 + 8/5 + 4/3 + 8/7 + 5/4 + 10/9 + 1) = 1.427$$

The *receiver operating characteristic* (ROC) curve is one of the most widely used graphical tools to plot the performance of a binary classifier (Bradley, 1997; Fawcett, 2004; Fawcett, 2006; Flach, 2010; Berrar and Flach, 2012). The ROC curve depicts the trade-offs between the false positive rate (or 1 minus specificity, shown on the x-axis) and the true positive rate (or sensitivity or recall, shown on the y-axis). These trade-offs correspond to all possible binary classifications that result from any dichotomization of the ranking scores. The points connecting the segments of the ROC curve are called *operating points*, and they correspond to the possible classification thresholds.

Fig. 5(a) shows the ROC plot for the introductory example. The diagonal red line is the expected ROC curve of a random guesser. The area under the diagonal is 0.5. Hence, any “good” model should produce a ROC curve above the red line, with an *area under the curve* (AUC) larger than 0.5. The AUC is a commonly used summary statistic of the ROC curve and can be interpreted as a conditional probability because it is equivalent to a Wilcoxon rank-sum statistic (Bamber, 1975; Hanley and McNeil, 1982; Hanley and McNeil, 1983): given any randomly selected positive and negative case, the AUC is the probability that the model assigns a higher score to the positive case (i.e., ranks it before the negative case). Following Hilden’s notation (Hilden, 1991), the AUC is defined as follows.

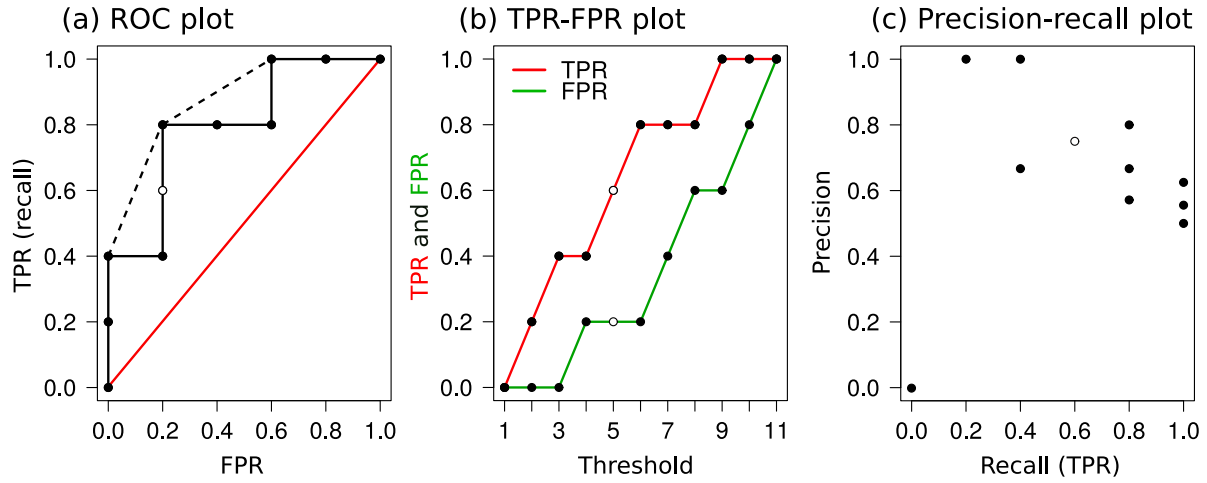


Fig. 5 Three different methods for the graphical representation of ranking performance. (a) ROC plot, showing the ROC curve (solid line) and the ROC convex hull (with dotted lines spanning the concavities). The red line is the expected performance of a random guesser. (b) TPR-FPR plot. (c) Precision-recall plot. The point marked by a white circle results from thresholding the ranking scores at t_5 .

Definition 25: Area under the ROC curve.

Let $P(p_+ > p_- | \mathbf{x}_+ \text{ and } \mathbf{x}_-)$ denote the probability that a randomly selected actual positive case, $\mathbf{x}_+$, has a higher ranking score, p_+ , than a randomly selected negative case, $\mathbf{x}_-$, i.e., $p_+ > p_-$. Here, a *higher* ranking score means that $\mathbf{x}_+$ is ranked *before* $\mathbf{x}_-$. Let $n_+ > 0$ be the number of positive instances and $n_- > 0$ be the number of negative instances, and $n = n_+ + n_-$. Let $\text{TPR}(t_i)$ and $\text{FPR}(t_i)$ denote the true positive rate and false positive rate for a threshold t_i , respectively, where $k = n + 1$ is the number of possible thresholds. The *area under the ROC curve* (AUC) is defined as

$$\text{AUC} = P(p_+ > p_- | \mathbf{x}_+ \text{ and } \mathbf{x}_-) = \int_0^1 \text{TPR}(t_i) d\text{FPR}(t_i) \quad (28)$$

From an empirical ROC curve, the AUC can be calculated as

$$\text{AUC} = \frac{S_- - 0.5n_-(n_+ + 1)}{n_+n_-} \quad (29)$$

where S_- is the sum of the ranks of the negative instances.

For the introductory example, we obtain $S_- = 3 + 6 + 7 + 9 + 10 = 35$, $n_- = 5$, $n_+ = 5$, and $\text{AUC} = \frac{35 - 0.5 \cdot 5 \cdot 6}{5 \cdot 5} = 0.8$, which corresponds to the shaded area in Fig. 6(a). Note that the expected AUC of a random model is 0.5, whereas the AUC of a perfect model is 1.

A closely related measure is the *Gini index*, which is defined as follows.

Definition 26: Gini index or Gini coefficient.

The *Gini index* (or *Gini coefficient*) is defined as

$$\text{Gini} = \frac{A}{B} \quad (30)$$

where A is the area between the ROC curve and the diagonal line from (0,0) to (1,1), and B is the area above that line, with $B = 0.5$.

The Gini index can be easily derived from the AUC. Consider Fig. 6(b). A is the purple area. As $\text{AUC} = A + 0.5$, we have $A = \text{AUC} - 0.5$. Then, since $\text{Gini} = \frac{A}{0.5} = 2A$, we obtain $\text{Gini} = 2 \cdot \text{AUC} - 1$. For the introductory example, we obtain $\text{Gini} = 2 \cdot 0.8 - 1 = 0.6$.

A further measure that can be derived from the ROC curve is the *area under the ROC convex hull* (AUCH). The ROC convex hull is the hull that encloses the operating points of the ROC curve (Flach, 2010; Provost and Fawcett, 2001). The line segment (0,0) to (0,0.4), the dotted lines, and the segment (0.6,1.0) to (1.0,1.0) in Fig. 5(a) represent the convex hull for the introductory example; the corresponding AUCH is shown in Fig. 6(c).

Definition 27: Area under the ROC convex hull.

The *area under the ROC convex hull* (AUCH) is the area under the curve that results from the interpolation between the following k points in ROC space, which are ordered based on increasing values of their abscissa: the origin $(x_i, y_i) = (0,0)$, the minimum set of points spanning the concavities, and the point (1,1). The AUCH can be calculated for an empirical ROC curve by using the trapezoidal rule,

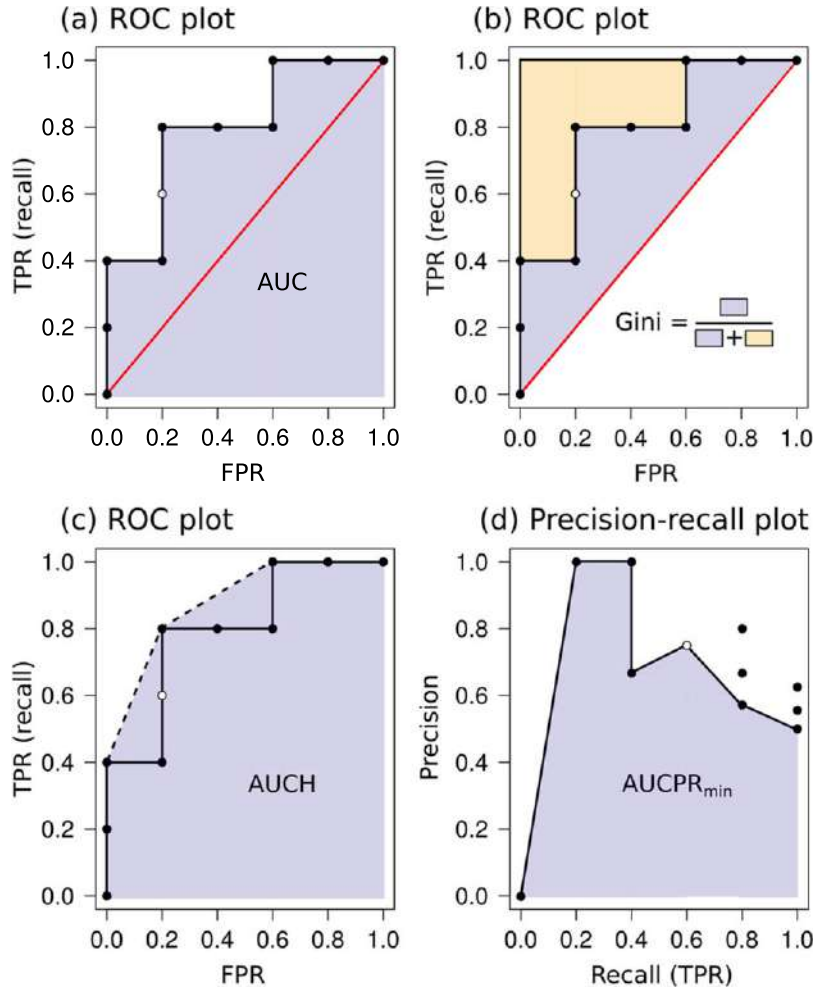


Fig. 6 Different graphical representations of ranking performance for the model from the introductory example (cf. Fig. 2). The point (0.2, 0.6) in ROC space and the point (0.6, 0.75) in precision-recall space result from thresholding the ranking scores at t_5 . (a) ROC plot, with $AUC=0.8$. The red line is the expected performance of a random guesser. (b) ROC plot, with Gini index $= \frac{0.3}{0.5} = 0.6 = 2 \cdot AUC - 1$. (c) ROC plot, with $AUCH=0.88$. (d) Precision-recall plot, with a trapezoidal estimator for the area under the curve, $AUCPR_{\min}=0.6476$.

$$AUCH = \sum_{i=1}^{k-1} y_i(x_{i+1} - x_i) + 0.5(y_{i+1} - y_i)(x_{i+1} - x_i) \quad (31)$$

For the introductory example, we obtain $AUCH=0.88$.

In Fig. 5(b), the true positive rate (TPR) and false positive rate (FPR) are plotted as a function of the classification threshold in a TPR-FPR plot, from which the Kolmogorov-Smirnov statistic can be derived.

Definition 28: Kolmogorov-Smirnov statistic.

The Kolmogorov-Smirnov statistic (KS) is the maximum value of the absolute difference between two cumulative distributions. When we assess the ranking ability of a model, these distributions are given by the true positive rates, $TPR(t_i)$, and false positive rates, $FPR(t_i)$, for all classification thresholds t_i .

$$KS = \max\{|TPR(t_i) - FPR(t_i)|\} \quad (32)$$

The KS statistic has a simple geometrical interpretation as the maximum distance between the TPR and FPR curves. In Fig. 5(b), the distance between the TPR and FPR curve is maximal for threshold t_6 , and $KS=0.8-0.6=0.2$. The KS statistic was shown to be sensitive to various types and levels of noise (Berrar, 2014; Berrar, 2016) and is therefore not recommended for model evaluation and comparison. A closely related but more robust measure is the truncated average Kolmogorov-Smirnov statistic (taKS) (Berrar, 2014), which calculates the average distance between the TPR- and FPR-curves.

Definition 29: Truncated average Kolmogorov-Smirnov statistic.

Let $\text{TPR}(t_i)$ and $\text{FPR}(t_i)$ denote the true positive and false positive rate for a classification threshold t_i , respectively, where $k=n+1$ is the number of possible thresholds. The *truncated average Kolmogorov-Smirnov statistic* (taKS) is the average distance between the TPR- and FPR-curves, excluding the start- and endpoints (0,0) and (1,1),

$$\text{taKS} = \frac{1}{k-2} \sum_{i=2}^{k-1} [\text{TPR}(t_i) - \text{FPR}(t_i)] \quad (33)$$

For the introductory example, we obtain $\text{taKS} = \frac{1}{9} \cdot [(0.2 - 0) + (0.4 - 0) + (0.4 - 0.2) + (0.6 - 0.2) + (0.8 - 0.2) + (0.8 - 0.4) + (0.8 - 0.6) + (1 - 0.6) + (1 - 0.8)] = 0.33$.

In precision-recall plots (Fig. 5(c)), the precision is plotted as a function of the recall (or true positive rate or sensitivity). A frequently used summary measure of a precision-recall plot is the *average precision*, which corresponds to the *area under the precision-recall curve* (AUCPR).

Definition 30: Average precision or area under the precision-recall curve.

Let n_+ be the number of positive instances and n_- be the number of negative instances, with $n_+ > 0$ and $n_- > 0$ and $n_+ + n_- = n$. Let $h_+(t_i)$ denote the hits, i.e., the number of positive instances at or above the threshold t_i , $i=1 \dots k$, where $k=n+1$ is the number of possible classification thresholds. Accordingly, let $h_-(t_i)$ denote the number of negative instances at or above the threshold. The recall at threshold t_i is $r(t_i) = \text{TPR}(t_i) = \frac{h_+(t_i)}{n_+}$. The precision is $p(t_i) = \frac{h_+(t_i)}{i-1}$ for $i > 1$ and 0 for $i=1$. The *average precision* (AP) is the area under the precision-recall curve,

$$\text{AP} = \int_0^1 p(t_i) dr(t_i) \quad (34)$$

From an empirical precision-recall curve, AP can be calculated by using the trapezoidal rule,

$$\text{AP} = \sum_{i=1}^k p(t_i) \Delta r_i = \sum_{i=1}^k p(t_i) [r(t_i) - r(t_{i-1})] \quad (35)$$

with $r(t_0)=0$.

For the introductory example, we calculate the average precision as follows (see Fig. 2): $\text{AP} = 1 \cdot 1/5 + 1 \cdot (2/5 - 1/5) + 2/3 \cdot (2/5 - 2/5) + 3/4 \cdot (3/5 - 2/5) + 4/5 \cdot (4/5 - 3/5) + 4/6 \cdot (4/5 - 4/5) + 4/7 \cdot (4/5 - 4/5) + 5/8 \cdot (5/5 - 4/5) + 5/9 \cdot (5/5 - 5/5) + 5/5 \cdot (5/5 - 5/5) = 0.835$.

The term “average precision” could be misunderstood as the average over the precisions resulting from all possible thresholds; in the introductory example, this average is $\frac{1}{10} \cdot (1 + 1 + 2/3 + 3/4 + 4/5 + 4/6 + 4/7 + 5/8 + 5/9 + 5/10) = 0.714$. Note that the expected AP of a random model is $\frac{n_+}{n}$, whereas the AP of a perfect model is 1.

From an empirical precision-recall plot, it is not obvious how the area should be calculated because the precision normally does not change monotonically with increasing recall (cf. Fig. 5(c)) (Boyd et al., 2013; Davis and Goadrich, 2006). From a plot like the one shown in Fig. 5(c), we can construct different curves by interpolating through different points, and consequently, we obtain (slightly) different areas under the curve, which can be calculated by using *trapezoidal estimators*.

Definition 31: Trapezoidal estimators of the area under the empirical precision-recall curve.

Let $p_{\min}(t_i)$ and $p_{\max}(t_i)$ denote the minimum and maximum precision, respectively, for a recall at threshold t_i , $i=1 \dots k$, where $k=n+1$ is the number of possible thresholds. The *area under the empirical precision-recall curve* (AUCPR) can be estimated by $\text{AUCPR}_{\min\max}$, $\text{AUCPR}_{\max}$ (upper trapezoid), or $\text{AUCPR}_{\min}$ (lower trapezoid), where

$$\text{AUCPR}_{\min} = \sum_{i=1}^{k-1} \frac{p_{\min}(t_i) + p_{\min}(t_{i+1})}{2} [r(t_{i+1}) - r(t_i)] \quad (36)$$

$$\text{AUCPR}_{\min\max} = \sum_{i=1}^{k-1} \frac{p_{\min}(t_i) + p_{\max}(t_{i+1})}{2} [r(t_{i+1}) - r(t_i)] \quad (37)$$

$$\text{AUCPR}_{\max} = \sum_{i=1}^{k-1} \frac{p_{\max}(t_i) + p_{\max}(t_{i+1})}{2} [r(t_{i+1}) - r(t_i)] \quad (38)$$

The *mean average precision* (MAP) is the extension of AP to more than two classes (Beitzel et al., 2009). MAP is simply the arithmetic mean of the average precisions, which we obtain by considering each class in turn as the positive class. MAP is a widely used performance measure in information retrieval (Beitzel et al., 2009).

Evaluating an Obtained Value of a Performance Measure

Once we have the value of our performance measure, how should we interpret it? For example, assume that we obtain a sensitivity of 0.80 – how do we know whether this is a good result or not? The performance may of course be evaluated based on domain

expert knowledge. If such knowledge is not available, however, then we could consider the statistical significance of the observed performance, keeping of course in mind the caveats and pitfalls of the p -value (Berrar, 2017).

Confidence intervals are always more meaningful than p -values. The statistical literature contains various approaches for calculating a confidence interval for a proportion. Here, we will present only two of them: a *simple asymptotic confidence interval* based on the normal approximation and the *exact Clopper-Pearson confidence interval*. We will assume that the prediction for an instance $\mathbf{x}_i$ is independent from the prediction for another instance $\mathbf{x}_j$, $i \neq j$. Also, we will assume that the test instances were randomly sampled from the target population. Note that, in practice, both assumptions may be violated.

Significance Testing With Random Permutation Tests

Random permutation tests are non-parametric Monte Carlo procedures that make fewer distributional assumptions than parametric significance tests. Such tests can be used to assess the statistical significance of an observed result. There exist various random permutation tests; for an overview, see (Good, 2000; Ojala and Garriga, 2010). In our context, the basic procedure of a random permutation test is as follows:

1. Train the classifier on the training set and apply it to the test set. Calculate the value v of the statistic of interest, i.e., the value of the performance measure.
2. Randomly permute the class labels of the training set. Thereby, any association between the data set attributes and the class labels is destroyed.
3. Retrain the classifier on the *permuted* training set and apply it to the test set. Calculate the statistic again (i.e., the value w_i of the performance measure of interest).
4. Repeat steps (2) and (3) many times (e.g., $k = 10000$ times) to generate the empirical distribution of the statistic under the null hypothesis of no association between the class labels and the data set attributes.

The p -value is defined as the probability of observing results as extreme as (or more extreme than) the observed results, given that the null hypothesis is true (Berrar, 2017; Sellke et al., 2001). The p -value quantifies the compatibility between the null hypothesis and the observed result: the smaller the p -value, the smaller is the evidence in favor of the null hypothesis (i.e., the less compatible is the null hypothesis with the observed result). We can calculate a p -value for our observed performance v by counting how many values w_i are at least as extreme as v . For example, assume that $v = 0.80$ is the observed sensitivity. Let us further assume that we performed 10000 random permutations and that among these, 300 values w_i are at least 0.80. Then the one-sided, empirical p -value is $\frac{300}{10000} = 0.03$. Thus, under the null hypothesis that there is no association between the attributes and the class labels, the probability of observing a sensitivity of at least 0.80 is 0.03.

Approximate Confidence Interval for a Proportion

Let p denote the (unknown) proportion in a population. For a sufficiently large sample, the sample proportion, $\hat{p}$, is approximately normally distributed with a mean of p . The standard deviation of the sampling distribution of the sample proportion (i.e., the standard error of the sample proportion) is $\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$, which is estimated as $s_{\hat{p}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$. The *Wald interval* is an approximate $(1-\alpha)100\%$ confidence interval for a population proportion (Wald and Wolfowitz, 1939).

Definition 32: Wald confidence interval.

Let $\hat{p}$ denote a proportion estimated from a sample of size n . If $n\hat{p} > 5$ and $n(1 - \hat{p}) > 5$, then an approximate asymptotic $(1-\alpha)100\%$ confidence interval for the population proportion p is given by the *Wald interval*,

$$\hat{p} \pm z_{1-\alpha/2} \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \quad (39)$$

where $z_{1-\alpha/2}$ is the quantile of the standard normal distribution for probability $1-\alpha/2$.

For example, a 95% Wald interval for the sensitivity in the introductory example is calculated as follows. With $TP=3$ and $FP=2$, we obtain $\hat{p} = \frac{3}{5}$ (= sensitivity). The interval is then

$$\hat{p} \pm z_{1-\alpha/2} \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.6 \pm 1.96 \cdot \sqrt{\frac{0.6 \cdot 0.4}{5}} = 0.6 \pm 0.4294 = [0.1706, 1.0294] \quad (40)$$

We see that the upper bound exceeds 1, which, obviously, is the maximum sensitivity; hence, the confidence interval includes values that are impossible. The approximation is not good because the conditions $n\hat{p} = 5 \cdot 0.6 > 5$ and $n(1 - \hat{p}) = 5 \cdot (1 - 0.6) > 5$ do not hold; therefore, an exact confidence interval should be calculated.

Exact Confidence Interval for a Proportion

An exact confidence interval based on the F -distribution is given by the Clopper-Pearson interval (Clopper and Pearson, 1934).

Definition 33: Exact Clopper-Pearson confidence interval.

Let $\hat{p} = \frac{r}{n}$ denote a proportion estimated from a sample of size n . An exact $(1-\alpha)100\%$ Clopper-Pearson confidence interval for the population proportion p is given by

$$\left[\frac{r}{r + (n - r + 1)F_{1-\alpha/2; 2(n-r+1), 2r}}, \frac{(r + 1)F_{1-\alpha/2; 2(r+1), 2(n-r)}}{(n - r) + (r + 1)F_{1-\alpha/2; 2(r+1), 2(n-r)}} \right] \quad (41)$$

where $F_{1-\alpha/2; df_1, df_2}$ is the quantile function of the F -distribution with df_1 and df_2 degrees of freedom.

For instance, a 95% confidence interval for the sensitivity in the introductory example is calculated as follows.

$$\left[\frac{3}{3 + (5 - 3 + 1)F_{1-0.05/2; 2(5-3+1), 2 \cdot 3}}, \frac{(3 + 1)F_{1-0.05/2; 2(3+1), 2(5-3)}}{(5 - 3) + (3 + 1)F_{1-0.05/2; 2(3+1), 2(5-3)}} \right] = [0.1466, 0.9473]$$

Bootstrap Percentile Confidence Interval

To derive a confidence interval analytically, it is necessary to calculate the standard error of the sample statistic, which is not trivial for the more intricate performance measures. When conventional parametric confidence intervals are difficult to calculate or when their assumptions are violated, deriving a *bootstrap confidence interval* is an interesting alternative. The *bootstrap* is a data resampling method for assessing the accuracy of statistical estimates (Efron, 1981; Efron and Tibshirani, 1994). There exist different types of bootstrap confidence intervals (Efron, 1987; DiCiccio and Efron, 1996). One of the simplest ones is the *percentile bootstrap* (Efron, 1981). The basic procedure for deriving a bootstrap percentile confidence interval is as follows (Berrar and Dubitzky, 2013):

1. From the available learning set, L , which contains a total of n instances, generate a bootstrap set, B , by randomly and uniformly sampling n instances with replacement.
2. Repeat step (1) b times to generate B_i bootstrap sets, $i = 1 \dots b$.
3. Build model C using the set B_i as training set, and apply C to the corresponding out-of-bag set T_i , which contains the elements from L that are not in B_i .
4. Calculate the value of the performance measure $\hat{\theta}_i$ from T_i .
5. Repeat steps (3) and (4) for all b bootstrap sets and calculate all $\hat{\theta}_i$, $i = 1 \dots b$.

Definition 34: Bootstrap percentile confidence interval.

Let L denote a learning set from a population. Let θ denote the unknown value of a performance measure for a predictive model C for that population. Let B_i denote the i^{th} bootstrap set (i.e., i^{th} training set) that was sampled uniformly with replacement from L , and let the number of instances in B_i and L be the same. Let T_i denote the i^{th} out-of-bag test set, with $T_i = L \setminus B_i$. The model C is trained on B_i and then applied to T_i . Let $\hat{\theta}_i$ denote the resulting value of the performance measure. A $(1-\alpha)100\%$ *bootstrap percentile confidence interval* (CI_{boot}) for θ is given by

$$CI_{\text{boot}} = [\hat{\theta}_{\alpha/2}, \hat{\theta}_{1-\alpha/2}] \quad (42)$$

where $\hat{\theta}_{\alpha/2}$ and $\hat{\theta}_{1-\alpha/2}$ are the $\alpha/2$ and $1 - \alpha/2$ percentiles, respectively, of the empirical distribution of $\hat{\theta}$.

In R, bootstrap confidence intervals can be generated with the function `boot.ci` of the package `boot` (Davison and Hinkley, 1997; Canty and Ripley, 2017).

Discussion

Which performance measure should be used in practice? A single “best” measure does not exist, as the different measures quantify different aspects of the performance. However, accuracy or error rate are not really meaningful when the classes are highly imbalanced. For example, assume that the class ratio is 9:1 in both the training and test set. A model that always predicts the majority class (i.e., a majority voter) is expected to achieve an accuracy of 90%, although it has not learned anything from the data. Performance measures that are based on a probabilistic interpretation of error are, in general, preferable to measures that rely on a single classification threshold. For example, let us assume that model A made the predictions shown in Table 2, and let us assume that model B made the same predictions, except that case #8 was misclassified with a score of 0.79. This is clearly a worse prediction than that of model A , which produced the score 0.75 for case #8. However, for t_5 , the threshold-based measures would not discriminate between A and B , in contrast to the measures that are based on a probabilistic interpretation of error.

Graphical tools such as ROC curves, lift charts, TPR-FPR plots, and precision-recall curves generally paint a clearer picture of the predictive performance than summary statistics. Clearly, it is often desirable to use a single value, for example, in order to tabulate the results of various models so that they can be ranked from best to worst, which is commonly done in data mining competitions. However, we have to keep in mind that we lose important information when we condense a two-dimensional plot into a single scalar. Thus, whenever possible, performance plots are preferable to scalars.

Ranking measures that are derived from such plots are popular evaluation measures; particularly, the AUC is widely used in machine learning. In contrast to accuracy, for example, the AUC is relatively robust to class imbalances and other types of noise (Berrar, 2014; Berrar, 2016). Therefore, the AUC is not a bad choice as a performance measure, although several shortcomings have

been pointed out (Hilden, 1991; Adams and Hand, 1999; Hand, 2009; Parker, 2013). For example, consider a drug discovery study that aims at ranking thousands of chemical compounds based on their toxicological effect. Here, it is typically possible to follow up on only a small number of top-ranked compounds, which should be predicted very accurately. By contrast, it may be irrelevant how well the lower-ranked compounds were predicted. This is also known as the *early retrieval problem*: although we are interested in only the top-ranked instances, the AUC also reflects the quality of the predictions of the remaining instances, which may not be interesting.

There exists an equivalence between ROC curves and precision-recall curves, in the sense that they contain the same points for the same predictive model (Davis and Goadrich, 2006). However, for data sets with highly imbalanced classes, precision-recall curves are more informative than ROC curves because precision-recall curves emphasize the performance with respect to the top-ranked cases (Saito and Rehmsmeier, 2015). In other words, if the correctness of the few top-ranked instances matters most (like in the mentioned drug discovery study), then precision-recall curves are preferable to ROC curves. This might explain why the average precision is more commonly used than the AUC in information retrieval (Su *et al.*, 2015). For data sets with highly imbalanced classes, the average precision (Definition 30) is therefore recommended as performance measure.

Closing Remarks

If a predictive model is developed for a concrete application, then it is usually clear which performance measure matters most. On the other hand, if the general usefulness of a new learning algorithm is to be assessed, then the choice of the most appropriate measure is often not obvious. In that case, reporting several measures can be meaningful. Graphical tools, such as ROC or precision-recall curves, are generally preferable to summary measures. For highly imbalanced classes, the average precision is the recommended performance measure, which is also easily extended to more than two classes as mean average precision (MAP).

Importantly, the performance measure in the training phase should be the same as the measure in the validation and test phase, since a model that was optimized to achieve, say, a high accuracy on the training set is not necessarily expected to achieve, say, a low logloss on the test set. There are further important aspects that need to be considered when we evaluate a predictive model or learning algorithm, such as the choice of benchmark data sets, data resampling strategies, statistical tools to compare different models or algorithms, etc. These aspects are beyond the scope of this article; for a more in-depth discussion, see (Japkowicz and Shah, 2011).

See also: Data Mining: Accuracy and Error Measures for Classification and Prediction. Data Mining: Prediction Methods

References

- Adams, N., Hand, D., 1999. Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition* 32 (7), 1139–1147.
- Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H., 2000. Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics* 16 (5), 412–424.
- Bamber, D., 1975. The area under the ordinal dominance graph and the area below the receiver operating characteristic curve. *Journal of Mathematical Psychology* 12, 387–415.
- Beitzel, S., Jensen, E., Frieder, O., 2009. MAP. In: Liu, L., Özsu, M.T. (Eds.), *Encyclopedia of Database Systems*. Boston, MA: Springer US, pp. 1691–1692.
- Berrar, D., 2014. An empirical evaluation of ranking measures with respect to robustness to noise. *Journal of Artificial Intelligence Research* 49 (6), 241–267.
- Berrar, D., 2017. Confidence curves: An alternative to null hypothesis significance testing for the comparison of classifiers. *Machine Learning* 106 (6), 911–949.
- Berrar, D., Flach, P., 2012. Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them). *Briefings in Bioinformatics* 13 (1), 83–97.
- Berrar, D., Dubitzky, W., 2013. Bootstrapping. In: Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H. (Eds.), *Encyclopedia of Systems Biology*. Springer, pp. 158–163.
- Berrar, D., 2016. On the noise resilience of ranking measures. In: Hirose, A., Ozawa, S., Doya, K., *et al.* (Eds.), *Proceedings of the 23rd International Conference on Neural Information Processing (ICONIP)*, Kyoto, Japan, Proceedings, Part II, Springer, pp. 47–55.
- Boughorbel, S., Jarray, F., El-Anbari, M., 2017. Optimal classifier for imbalanced data using Matthews correlation coefficient metric. *PLoS One* 12 (6), e0177678.
- Boyd, K., Eng, K.H., Page, C.D., 2013. Area under the precision-recall curve: Point estimates and confidence intervals. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (Eds.), *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23–27, 2013, Proceedings*, Part III, Springer, Berlin, Heidelberg, pp. 451–466.
- Bradley, A., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30 (3), 1145–1159.
- Brier, G., 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review* 78 (1), 1–3.
- Buja, A., Stuetzle, W., Shen, Y., 2005. Loss functions for binary class probability estimation and classification: Structure and applications. Available at: www-stat.wharton.upenn.edu/~buja. Accessed 15 March 2018.
- Canty, A., Ripley, B., 2017. Boot: Bootstrap R (S-Plus) functions. R package version 1.3–20.
- Clopper, C., Pearson, E., 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26 (4), 404–413.
- Cohen, J., 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20 (1), 37–46.
- Davis, J., Goadrich, M., 2006. The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning, ACM*, pp. 233–240.
- Davison, A., Hinkley, D., 1997. *Bootstrap Methods and Their Applications*. Cambridge University Press.
- DiCiccio, T., Efron, B., 1996. Bootstrap confidence intervals. *Statistical Science* 11 (3), 189–228.
- Dubitzky, W., Granzow, M., Berrar, D., 2001. Comparing symbolic and subsymbolic machine learning approaches to classification of cancer and gene identification. In: Lin, S., Johnson, K. (Eds.), *Methods of Microarray Data Analysis*. Kluwer Academic Publishers, pp. 151–166.
- Efron, B., 1981. Nonparametric standard errors and confidence intervals. *Canadian Journal of Statistics* 9 (2), 139–158.

- Efron, B., 1987. Better bootstrap confidence intervals. *Journal of the American Statistical Association* 82 (397), 171–185.
- Efron, B., Tibshirani, R., 1994. *An Introduction to the Bootstrap*. Chapman & Hall.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874.
- Fawcett, T., 2004. ROC graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories, pp. 1–38.
- Ferri, C., Hernández-Orallo, J., Modroiu, R., 2009. An experimental comparison of performance measures for classification. *Pattern Recognition Letters* 30, 27–38.
- Flach, P., 2010. ROC analysis. In: Sammut, C., Webb, G. (Eds.), *Encyclopedia of Machine Learning*. Springer, pp. 869–875.
- Gneiting, T., Raftery, A., 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102 (477), 359–378.
- Good, P., 2000. *Permutation tests: A practical guide to resampling methods for testing hypotheses*. Springer Series in Statistics.
- Hand, D., 2009. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning* 77, 103–123.
- Hanley, J., McNeil, B., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143 (1), 29–36.
- Hanley, J., McNeil, B., 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 148 (3), 839–843.
- Hilden, J., 1991. The area under the ROC curve and its competitors. *Medical Decision Making* 11 (2), 95–101.
- Japkowicz, N., Shah, M., 2011. *Evaluating Learning Algorithms – A Classification Perspective*. Cambridge University Press.
- Kononenko, I., Bratko, I., 1991. Information-based evaluation criterion for classifier's performance. *Machine Learning* 6 (1), 67–80.
- Lavrač, N., Gamberger, D., Džeroski, S., 1997. Noise elimination applied to early diagnosis of rheumatic diseases. In: Lavrač, N., Keravnou, E.T., Zupan, B. (Eds.), *Intelligent Data Analysis in Medicine and Pharmacology*. Boston, MA: Springer US, pp. 187–205.
- Matthews, B., 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta – Protein Structure* 405 (2), 442–451.
- Ojala, M., Garriga, G., 2010. Permutation tests for studying classifier performance. *Journal of Machine Learning Research* 11, 1833–1863.
- Parker, C., 2013. On measuring the performance of binary classifiers. *Knowledge and Information Systems* 35, 131–152.
- Powers, D., 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2, 37–63.
- Provost, F., Fawcett, T., 2001. Robust classification for imprecise environments. *Machine Learning* 42 (3), 203–231.
- Ruopp, M., Perkins, N., Whitcomb, B., Schisterman, E., 2008. Youden index and optimal cut-point estimated from observations affected by a lower limit of detection. *Biometrical Journal* 50 (3), 419–430.
- Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE* 10 (3), e0118432.
- Sellke, T., Bayarri, M., Berger, J., 2001. Calibration of p values for testing precise null hypotheses. *The American Statistician* 55 (1), 62–71.
- Su, W., Yuan, Y., Zhu, M., 2015. A relationship between the average precision and the area under the ROC curve. In: *Proceedings of the 2015 International Conference on the Theory of Information Retrieval, ICTIR 2015*, ACM, New York, NY, pp. 349–352.
- Wald, A., Wolfowitz, J., 1939. Confidence limits for continuous distribution functions. *The Annals of Mathematical Statistics* 10 (2), 105–118.
- Witkowski, J., Atanasov, P., Ungar, L., Krause, A., 2017. Proper proxy scoring rules. In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI Press, pp. 743–749.
- Youden, W., 1950. Index for rating diagnostic tests. *Cancer* 3 (1), 32–35.

Natural Language Processing Approaches in Bioinformatics

Xu Han and Chee K Kwoh, Nanyang Technological University, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

The quantity of biomedical literature is expanding at an increasing rate. As of 2017, over 23 million references to biomedical literature are included in the MEDLINE database, and this number is growing at the rate of more than 700,000 each year (see “Relevant Websites section”). With such high-speed growth, it has become more and more challenging for biomedical researchers to keep up to date with all the latest advancements within their own research fields. To lessen the burden of this information overload, more and more sophisticated *biomedical natural language processing* (BioNLP) systems are becoming available. Before we discuss the BioNLP in details, we introduce the historical evolution of the natural language processing (NLP) domain and the rise of statistical natural language processing, followed by some of the common NLP sub-problems and their applications to the biomedical text. Finally we discuss some of the advanced topics on the applications of active learning method to the biomedical NLP systems, in order to solve the lacking of training data issue for such systems.

The Historical Evolution of NLP

NLP began as the intersection of artificial intelligence and linguistics in the 1950s. Originally the NLP is distinct of the information retrieval (IR), which aims to establish principled approaches to search various contents such as scientific publications, library records, news-wires. And in Schütze *et al.* (2008) there is an excellent introduction of the topic of IR. In 1956, Chomsky published the theoretical analysis of language grammars (Chomsky, 1956), which estimated the difficulty of the problem in NLP. The analysis leads to the creation of the context-free grammar (CGF) (Chomsky, 1959), which is widely used to represent programming-language syntax.

Up to the 1980s, most parsing approaches in NLP systems were based on complex sets of symbolic and hand-crafted rules. However, there is the problem of ambiguous parses applying to the natural language, as the rules are of large size, unrestrictive nature and ambiguity. The ambiguous parses means that multiple interpretations of a word sequence were possible, as the grammar rules became unmanageably numerous and they were interacting unpredictably.

Starting in the late 1980s, more and more statistical NLP methods are applied to the NLP problems, these methods introduce machine learning algorithms for the language processing (Chomsky, 1957). For examples, in statistical parsing, the parsing-rule proliferation is addressed by the probabilistic CFGs (Klein and Manning, 2003a), where the individual rule is associated probabilities that is estimated through machine-learning on annotated corpora. Thus, the broader yet fewer rules replace the numerous detailed rules, and disambiguation is achieved through referring to the statistical-frequency information contained in the rules. There are other approaches that build the probabilistic rules based on the annotated data. For instance in Quinlan (2014), the researchers build decision trees from feature-vector data, and the statistical parser determine the ‘most likely’ parse of a sentence/phrase, where the statistical parser need to be trained on the domain-specific training data and is context-dependent. For instance, the Stanford Statistical Parser trained with the Penn Treebank, which is the annotated Wall Street Journal articles, will not be suitable for biomedical text. For the topic of statistical NLP, Manning provides an excellent introduction in Manning and Schütze (1999).

NLP Sub-Problems and Applications to Biomedical Text

The research area in the NLP can be broadly divided into two layers: the syntax layer and the semantics layer. A general summarization of the sub-problems and tasks that are covered in our discussions is illustrated in the Fig. 1. Our discussions in the subsequent sections follow the order of the tasks in the figure.

Pre-Processing and Syntactic Layer

In this section, we introduce a fundamental pre-processing step, text normalization, in the NLP. It is a set of tasks that convert the text into a more convenient and standard form. We then discuss a set of tasks that are targeted on the syntactic layer analysis, including language modeling, part-of-speech tagging, text chunking and syntactic parsing.

Text normalization

Text normalization refers to the tasks of converting the text into a more convenient and standard form. These tasks include *tokenization*, *lemmatization*, *stemming*, *sentence segmentation*. The task of *tokenization* is to separate out or tokenizing words from the running text. In the English language, the words are often separated from each other by the whitespace. However in some

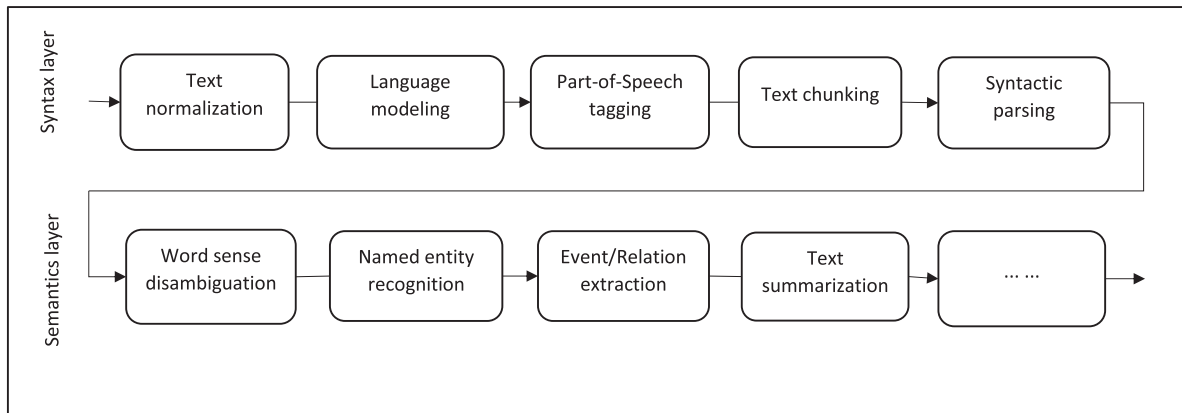


Fig. 1 The syntax layer and the semantics layer of natural language processing in the bioinformatics.

applications of NLP, separating by whitespace is not sufficient. For instance, we may need to separate the *I'm* into the two words: *I* and *am*. In this case, a tokenizer need to expand the *clitic* contractions that are marked by apostrophes. A clitic is a part of word that can't stand on its own, and only occurs when attached to another word. In other applications, a tokenizer need to treat a multi-word expression as a single token, although there is whitespace in the expression. For example, the 'United States' is recognized as a large token. This type of recognition is intimately correlated in the named entity recognition task, as discussed in Section "Named entity recognition".

One commonly used tokenization standard is the *Penn Treebank tokenization* standard, which is adopted in the parsed corpora by the Linguistic Data Consortium (LDC). In this standard, hyphenated words are kept together and the clitics and all punctuation are separated out. One example of the tokenization is given below:

Input: "The San Francisco-based restaurant," they said, "doesn't charge \$10".

Output: “The San Francisco-based restaurant”, “they said
, “does n’t charge \$10”.

For other languages such as Chinese or Japanese, where there is no spaces between words, the task of tokenization becomes more difficult.

Another task in text normalization is the *lemmatization*, which is the task to determine whether the input words have the same root, despite their surface differences. For instance, the words *eat*, *ate*, *eaten* and *eats* are the lemma *eat*. A lemmatizer maps from all these word to *eat*. One of the approaches for lemmatization is the *morphological parsing* of the word, where the word is regarded as built up from smaller meaning-bearing units called *morphemes*. Two broad types of morphemes are the stems and the affixes. The stems denotes the central morpheme of the word, and they defines the main meaning of the word; while the affixes carry the ‘additional’ information for the word. So the word ‘apples’ have two morphemes: the stem ‘apple’ and the affix ‘-s’. The task of *stemming* is a simpler form of lemmatization, where the suffixes are stripped from the end of the word. For instance, the word *looking* is converted to *look* after stemming. One of the commonly used stemmer is the *Porter Stemmer*, proposed in 1980 (Porter, 1980). The algorithm is based on a set of rewrite rules that applied to words in cascade, where the output of a certain step is fed as input for the subsequent step. Detailed rule lists of the Porter Stemmer and the source code of the program is provided in the author’s homepage (see “Relevant Websites section”).

Sentence segmentation is another step in the text processing, where the sentence splitter use the punctuation to segment a text into sentences. While the question marks and exclamation points are relative unambiguous for sentence segmentation, some punctuation mark, such as the period, may pose ambiguity between abbreviation and sentence boundary to the task. To solve this issue, sentence splitter work in general by building a binary classifier, either based on rules or machine learning, to decides if a period is the abbreviation or the sentence boundary.

Currently a wide variety of useful tools for tokenization and normalization is available, such as the Stanford Tokenizer (see “Relevant Websites section”) or specialized tokenizers for Twitter (O’Connor *et al.*, 2010), or for sentiment (see “Relevant Websites section”). NLTK (see “Relevant Websites section”) is an essential tool that offers useful Python libraries of many algorithms including text normalization and corpus interfaces for the NLP domain. We refer the interested reader to the publication in Indurkha and Damerau (2010) for a survey of text preprocessing.

Language modeling with N-grams

One strength for the statistical NLP is the ability to make statistical inference. For example, given a precedent word, the statistical NLP can assign a probability to each possible next word. The prediction of upcoming words, or assigning the probability of the whole sentence, is crucial in task like *handwriting recognition*, where the words need to be identified in noisy and ambiguous input. The models that assign probabilities to sequences of words are referred to as *language models*. The simplest model that assigns probabilities to sentences is the *N-gram* model. An N-gram is a sequence of N words: A 2-gram (or bi-gram) consists of two words; while a 3-gram (or trigram) have three words. One example is given as follows:

Input: Please give me that apple.

Bi-gram:

Please give

,

give me

,

me that

,

that apple

Trigram:

Please give me

,

give me that

,

me that apple

By splitting the text of the corpus into N-gram, we can get a statistical estimation for the conditional probability for a word given its preceding words. For instance in bi-gram model, we get the estimation of the probability $P(\text{me}|\text{give})$, where the word 'me' is preceded by the word 'give', by counting the occurrence of the bi-gram 'give me' in the corpus. In practical computation, we always use the log probability, to avoid the numerical underflow issue if too many probabilities are multiplied. To evaluate the language models, one commonly used metric is *perplexity*, which is the inverse probability of the test set, normalized by the number of words.

By using text from the web, it is possible to build extremely large language models. In 2006 Google released a very large set of N-gram counts, including N-grams (1-grams through 5-grams) from all the five-word sequences that appear at least 40 times from 1024, 908, 267, 229 words of running text on the web; this includes 1176, 470, 663 five-word sequences using over 13 million unique words types (Franz and Brants, 2006).

As the language modeling assign probabilities, one practical consideration is to keep a language model from assigning zero probability to the unseen N-grams, which may cause the whole sentence probability to be zero and affects the subsequent log-calculation. One remedy is to shave off a bit of probability mass from some more frequent N-grams and give it to the N-grams were never seen. This modification is called smoothing or discounting.

A wide variety of different language modeling and smoothing techniques were proposed in the 80s and 90s, including Good-Turing discounting first applied to the N-gram smoothing at IBM by Katz (Nadas, 1984; Church and Gale, 1991) Witten-Bell discounting (Witten and Bell, 1991), and varieties of class-based N-gram models that used information about word classes. Two commonly used toolkits for building language models are SRILM (Stolcke, 2002) and KenLM (Heafield, 2011; Heafield et al., 2013). SRILM offers a wider range of options and types of discounting, while KenLM is optimized for speed and memory size, making it possible to build web-scale language models. Another important technique is language model adaptation, where we want to adaptation combine data from multiple domains. For example, we might have less in-domain training data for a particular domain, but have much more data in the general domain. In this case, one practical solution is the language model adaptation, as reported in the publications (Bulyko et al., 2003; Bacchiani et al., 2004, 2006; Bellegarda, 2004; Hsu, 2007; Liu et al., 2013).

Part-of-speech tagging

Part-of-speech is also known as POS, *word classes* or *syntactic categories*. In the English language, some common types of *part-of-speech* include noun, verb, pronoun, preposition, adverb, conjunction, participle and article. Knowing whether a word is noun or verb can provide much information about the POS tag for its neighboring word. For instance, the nouns are usually preceded by determiners and adjectives, and verbs are often preceded by nouns. This makes the POS tagging an important component in the syntactic parsing. In applications such as named entity recognition and information extraction (introduced in Sections "Named entity recognition" and "Event and relation extraction"), the POS tags of the words are useful features for the NLP systems. *Part-of-speech tagging* is the process of assigning a POS tag to each word of the input text. Generally the step of tokenization is adopted before the POS tagging step in NLP, as the tags are also applied to punctuation.

Most current NLP applications on English use the 45-tag Penn Treebank tag set (Marcus et al., 1993). Generally the POS information is represented by placing the POS tags after each word, delimited by a slash. One example sentence after the POS tagging is shown below:

Input: The grand jury commented on a number of other topics.

Output: The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS/.

As each word can have more than one possible part-of-speech, the problem of POS-tagging is to choose the proper tag for the context. The training and testing datasets for the statistical POS tagging algorithms are often the corpora labeled with POS tags. For the English language, some of the most commonly used corpora are the *Brown* corpus, the *WSJ* corpus and the *Switchboard* corpus. The Brown corpus is based on 500 written texts from different genres published in the United States in 1961, while the WSJ corpus contains words published in the Wall Street Journal in 1989, and the Switchboard corpus mainly focus on telephone conversations

that are collected in 1990–1991. Regarding to the construction process of these corpora, they are created by applying an automatic POS tagger on the text of the corpus, and then the POS tags are manually curated by the human annotators. And there is two commonly used algorithms for POS-tagging: the Hidden Markov Model (HMM) (Church, 1988) and the Maximum Entropy Markov Model (MEMM) (Ratnaparkhi, 1996).

In the POS tagging for biomedical domain, there are the GENIA tagger (Tsuruoka *et al.*, 2005a) and the MedPost (Smith *et al.*, 2004), which are specifically designed for tagging biomedical text. And one commonly used corpus for training the POS tagger is the GENIA Corpus (Kim *et al.*, 2003). Recently, the FLORS POS tagger is reported in Schnabel and Schütze (2014), which is based on domain adaptation method and shows significantly better performance than some of the state-of-the-art POS taggers, including the SVMTool tagger (Giménez and Marquez, 2004) that based on support vector machine (SVM), the bidirectional log-linear model based Stanford tagger (Toutanova *et al.*, 2003), the HMM-based TnT tagger (Brants, 2000).

Text chunking

Text chunking is the process of identifying and classifying the flat, non-overlapping segments of a sentence that constitute the basic non-recursive phrases corresponding to the major parts-of-speech found in most wide-coverage grammars. This set typically includes noun phrases, verb phrases, adjective phrases, and prepositional phrases; in other words, the phrases that correspond to the content-bearing parts-of-speech. The most common chunking task is to simply find all the base noun phrases in a text.

Since chunked texts lack a hierarchical structure, a simple bracketing notation is sufficient to denote the location and the type of the chunks in the text. One example of a typical bracketed notation is illustrated below:

text chunking: [_{NP} The morning flight] [_{PP} from] [_{NP} Denver] [_{VP} has arrived.]

This bracketing notation makes clear the two fundamental tasks that are involved in chunking: finding the non-overlapping extents of the chunks and assigning the correct label to the identified chunks.

Note that in this example all the words are contained in some chunk. This will not be the case in all chunking applications, as the words in the input text will often fall outside of any chunk. For example, in systems searching for only NPs in their inputs, the output will be as follows:

text chunking: [_{NP} The morning flight] from [_{NP} Denver] has arrived.

State-of-the-art approaches to chunking use supervised machine learning to train a chunker by using annotated data as a training set. A common approach to text chunking is to treat chunking as a tagging task similar to part-of-speech tagging (Ramshaw and Marcus, 1999). In this approach, a small tag set simultaneously encodes both the segmentation and the labeling of the chunks in the input. The standard way to do this is called IOB tagging and is accomplished by introducing tags to represent the beginning (B) and internal (I) parts of each chunk, as well as those elements of the input that are outside (O) any chunk.

We give the IOB tagging result for the example sentence as follows:

input text: The morning flight from Denver has arrived.

IOB tagging: B_NP I_NP I_NP B_PP B_NP B_VP I_VP

Given such a scheme, building a chunker consists of training a classifier to label each word of an input sentence with one of the IOB tags from the tag set. Of course, training requires training data consisting of the phrases of interest delimited and annotated with the proper category. The direct approach is to annotate a representative corpus. However, annotation efforts can be both expensive and time consuming. It turns out that the best place to find such data for chunking is in an existing treebank such as the Penn Treebank (Marcus *et al.*, 1993), as mentioned in the Section “Part-of-Speech tagging”.

Context-free grammar and syntactic parsing

Before we introduce the syntactic parsing, we discuss about the context-free grammar, or CFG, which is formalized by Chomsky (1956) and, independently, Backus (1959). Context-free grammars are the backbone of many formal models of the syntax of natural language, and they are integral to many computational applications, such as grammar checking, semantic interpretation, dialog understanding, and machine translation. They are powerful enough to express sophisticated relations among the words in a sentence, yet computationally tractable that support efficient algorithms to parse sentences with them.

A context-free grammar consists of a set of rules or productions, each of which expresses how the symbols of the language can be grouped and ordered together, and a lexicon of words and symbols. For example, the following productions express that an NP (or noun phrase) can be composed of either a Proper Noun or a determiner (Det) followed by a Nominal; a Nominal in turn can consist of one or more Nouns.

context-free grammar:

$NP \rightarrow DetNominal$

$NP \rightarrow ProperNoun$

$Nominal \rightarrow Noun|NominalNoun$

Context-free rules can be hierarchically embedded, so we can combine the previous rules with others. For instance, the Fig. 2 illustrate how the context-free rules are embedded, to form a parse tree for the phrase “a promoter”.

Similarly, we can build the parse tree for a sentence, as shown in Fig. 3.

A parse tree or syntax tree is an ordered, rooted tree that represents the syntactic structure of a string according to the context-free grammar. The parse trees can be constructed based on the constituency relation of constituency grammars, which is also called the phrase structure grammars, as shown in Fig. 3. However, the parse trees can also be generated based on the dependency grammars. We refer the interested reader to the detailed discussions in Ágel (2006) and Carnie (2012).

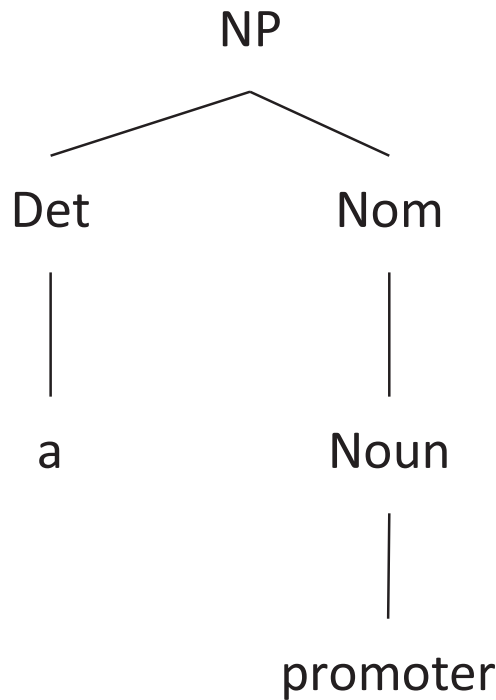


Fig. 2 A parse tree for “a promoter”.

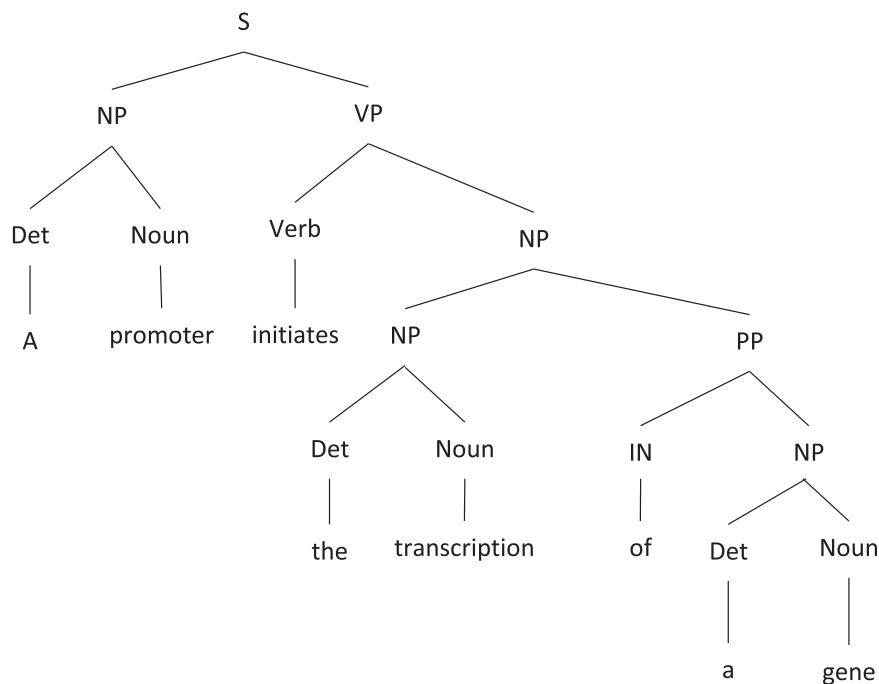


Fig. 3 A parse tree for sentence “A promoter initiates the transcription of a gene”.

A treebank is a syntactically annotated corpus where every sentence in the collection is paired with a corresponding parse tree. Treebanks play an important role in parsing, as well as in linguistic investigations of syntactic phenomena. To build a treebank it is possible to manually make the annotation only by linguists, yet such effort is expensive and costly. In general, a wide variety of treebanks have been created through the use of parsers to automatically parse each sentence, followed by linguists to manually correct the parses. One commonly used treebank for syntactic parsing is the Penn Treebank project (whose POS tagset we introduced in Section “Part-of-Speech tagging”).

In statistical NLP, there are many methods to augment the CFG in order to allow the statistical learning for the syntactic parser. One simplest augmentation of the context-free grammar is the Probabilistic Context-Free Grammar (PCFG), also known as the Stochastic Context-Free Grammar PCFG(SCFG), first proposed by Booth (1969). A PCFG differs from a standard CFG by augmenting each rule with a conditional probability. Then the issue is how to learn PCFG rule probabilities. The simplest way is to use a treebank, a corpus of already parsed sentences. Given a treebank, we can compute the probability of each expansion of a non-terminal by counting the number of times that expansion occurs and then normalizing. For the detailed discussions on the statistical learning for the PCFG, we refer the interested readers to the research work in Schabes *et al.* (1988), Schabes (1990), Hindle and Rooth (1993), Klein and Manning (2003b) and Petrov *et al.* (2006).

Semantical Layer

Word sense disambiguation

Word sense disambiguation (or WSD) is the task of selecting the correct sense for a word. In English, one word could have multiple senses. A *sense* (or *word sense*) is a discrete representation of one aspect of the meaning of a word. For instance, the word 'bank' have different aspects of meaning in the two phrases: 'the federal bank' and 'the river bank'. The word sense disambiguation is important to many NLP tasks, such as machine translation and information retrieval. Generally, the WSD algorithm takes a word in context along with a fixed inventory of potential word senses, and return the correct word sense for that use as the output. One approach to the WSD problem is the supervised learning, where the training data that has been hand-labeled with correct word senses are prepared, and then the WSD algorithm extracts features from the text, and a classifier is trained to assign the correct word sense given these features.

The supervised WSD algorithms based on sense-labeled corpora have the best-performing performance for sense disambiguation. However, such labeled training data is expensive and difficult to prepare. One alternative method is to get indirect supervision from dictionaries, thesauruses or similar knowledge bases. And this approach is also called knowledge-based WSD. The most well-studied dictionary-based algorithm for sense disambiguation is the Lesk algorithm, which choose the sense whose dictionary gloss or definition shares the most words with the target words neighborhood (Lesk, 1986; Kilgarriff and Rosenzweig, 2000). The recent high performing systems generally use POS tags and word collocations from a words window (Zhong and Ng, 2010).

One commonly used resource for this approach is the *WordNet*, which consists of three separate databases, one each for nouns and verbs and a third for adjectives and adverbs for the lexical relations. Each database contains a set of lemmas, each one annotated with a set of senses (Kilgarriff, 2000).

Both the supervised approach and the dictionary-based approaches to WSD require large hand-built resources: supervised training sets in one case, large dictionaries in the other.

We can instead use bootstrapping or semi-supervised learning, which needs only a very small hand-labeled training set. A classic bootstrapping algorithm for WSD is the Yarowsky algorithm for learning a classifier for a target word (in a lexical-sample task) (Yarowsky, 1995).

Modern interest in supervised machine learning approaches to disambiguation began with Black (1988), who applied decision tree learning to the task. The need for large amounts of annotated text in these methods led to investigations into the use of bootstrapping methods (Yarowsky, 1995), which needs only a very small hand-labeled training set. Diab and Resnik (2002) give a semi-supervised algorithm for sense disambiguation based on aligned parallel corpora in two languages. For a comprehensive survey article on WSD, we recommend the publication in Navigli (2009), and in Agirre and Edmonds (2007) there is a summary for the state-of-the-art progress in the WSD.

Named entity recognition

One of the fundamental research areas in biomedical information extraction is the named entity recognition (NER) task. Theoretically, the biomedical information extraction includes: (1) the recognition of terms expressing concepts for biomedical entities, for example, the gene names and protein names are to be recognized from the raw text, (2) the extraction of specific relations between biomedical entities, for instance, the discovery of the protein interactions based on the raw text, and (3) the extraction of the more complex types of biomedical information. One example is the widely known GENIA project (see "Relevant Websites section"), which is to uncover the metabolic pathways of the protein NF-kappa B. The NER is a crucial step for other sophisticated applications such as information extraction, since the steps of the entities recognition and linking them to the standardized ontological concepts in the NER are the foundations for accessing the textually described domain-specific information in the information extraction. Even besides information extraction, the NER supports the solution of more complicated tasks in text mining, such as event and relation extraction (de Bruijn and Martin, 2002; Hanisch *et al.*, 2003).

The linking of the terms in raw text with the concepts pre-defined in the ontology is a requirement for the text mining systems. This step is due to fact that the terms are more variable than concepts explicitly represented in the ontology. One example of the common variations is that EGR 1 and EGR 1 and EGR1 all express the same biological concept. Thus if the three variants all appear in the raw text, they all should be considered as the same concept. It is reported that such phenomenon of variance in raw text consists of one third of the total term occurrences (Wacholder, 2003), which suggests the identification of variance may help the recognition of terms in text.

On the other hand, the extraction of synonym and abbreviation comes from the inherent characteristics of terminologies in the biomedical domain. Paralleling the growth of the biomedical literature is the expanding volume of biomedical terminologies. In the current biomedical documents, many biomedical entities have more than one names or abbreviations. The issue has been attracting more and more attention of research, since it would be advantageous to extract the synonyms and abbreviations, so as to aid users in document retrieval in an automatic way. In the current study of biomedical text mining, the synonym and abbreviation extraction is mainly about the extraction of gene names and abbreviations of biomedical terms.

One application of the extraction of synonym and abbreviation is that it can be incorporated into the search engines to improve the search results. For example, one may use the medical abbreviation in using the PubMed to search relevant documents; the search engine, however, can integrate a dictionary of medical abbreviations, based on the dictionary of abbreviations to augment search queries and improve the search results. One thing worth noting is that if an abbreviation, which is well-accepted in a particular domain yet is not included in the maintained dictionary, is encountered in an article, the aim of improving in research results is becoming more difficult to achieve, since the solution of this issue requires the expert knowledge for that particular domain, together with the interpretation of the context of such abbreviation.

The extraction for gene name and protein name synonym is a more challenging problem. One possible way is to construct a synonym list and keep updating it. However, the quality of extracting the list is not guaranteed because of the low precision in extraction systems. (Cohen *et al.*, 2005) has used a pattern extraction approach and co-occurrence analysis to generate a synonym extraction system. Without any NER module for the gene names, the system has a score of 22% in F-measure. Yet with the identification of logical relation, the recall for the system improved by 10%.

The research progress on the NER domain is evaluated through the shared tasks, which is a community-wide competition for the state-of-the-art NER systems. We review some of the latest evaluation competition tasks on NER, including the CoNLL for the general domain, and the BioCreAtIvE for the biomedical domain.

The CoNLL-2003 shared task is one of the most widely used tasks for NER in the general domain. The dataset for the English language is taken from the Reuters Corpus, which consists of Reuters news stories between August 1996 and August 1997 (Sang and De Meulder, 2003). In the data file, one word is contained per line, and the empty lines represent the sentence boundaries. Within each line, four fields are contained: the word, the word's part-of-speech tag, the chunk tag for the word, and the named entity tag for the word.

The named entity tag is defined as follows: the word with O tag means it is outside of named entity; while words with I-XXX tag indicates the opposite, i.e., inside a named entity that is typed XXX. In addition, where two named entities that are of the same type XXX and are adjacent to each other, the named entity tag for the first word is B-XXX. In the shared task, four types of named entities are defined: persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC). An example sentence of the CoNLL-2003 NER task is given in Table 1.

The settings for the shared task are that, learning from the provided training dataset, the text mining system need to recognize the named entities that are contained in the separate evaluation dataset. In the shared task, 16 systems have employed a wide variety of machine learning techniques, including Maximum Entropy Model (Bender *et al.*, 2003), Hidden Markov Model (Florian *et al.*, 2003), AdaBoost.MH (Carreras *et al.*, 2003), memory-based learning (De Meulder and Daelemans, 2003), Transformation-based learning (Florian *et al.*, 2003), Support Vector Machines (Mayfield *et al.*, 2003) and Conditional Random Fields (McCallum and Li, 2003).

The evaluation results in the task are ranging from 69% to 88% in F-measure, while recently, the state-of-the-art Stanford NER system has reported the F-measure of around 90% in the task dataset (Finkel *et al.*, 2005).

Recently, more and more research work has been published on the application of natural language processing and information extraction to the biomedical literature. However, as Hirschman pointed out, each group had addressed different problems on different data sets (Hirschman *et al.*, 2002). Hence it became important to assess the progress of biomedical text mining systematically. Similar to the CoNLL shared tasks, the aim of organizing BioCreAtIvE challenge (Critical Assessment of Information Extraction in Biology) is to support a systematic assessment of the state-of-the-art information extraction systems. However, comparing to the CoNLL shared tasks, the BioCreAtIvE challenge is information extraction in the biomedical domain, which has its own specialty comparing to the general domain.

Two tasks are defined in the BioCreAtIvE II (Smith *et al.*, 2008). The first task assessed the extraction of gene names or protein names, and the linking of gene or protein names into the standardized gene identifiers in the databases for model organisms in biology, including fly, mouse and yeast. The second evaluates the functional annotation, where the participating systems were

Table 1 Example sentence for the CoNLL-2003 NER task

U.N.	NNP	I-NP	I-ORG
Official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
Heads	VBZ	I-VP	O
For	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

given full-text articles and should recognize specific text that supports the annotation for specific proteins with the Gene Ontology (Hirschman *et al.*, 2005).

One aim of BioCreAtIvE challenge is to support the curation process of biological databases. The BioCreAtIvE challenge was built on the KDD Challenge Cup, which is one of the earliest evaluations competitions in the biomedical text mining (Yeh *et al.*, 2003). Similar to the BioCreAtIvE, the KDD Cup also hosts tasks on the curation of biological literature, in particular, the recognition of articles that contain experimental evidence for the gene products of the database Flybase (see “Relevant Websites section”).

27 groups participated in the BioCreAtIvE challenge, and the evaluation results in task 1 are ranging from 80% to 90% in F-measure. Task 2, on the other hand, created a baseline for future research on functional annotation.

From the BioCreAtIvE challenge, interesting questions relevant to biology were raised. The gene name tagging task in BioCreAtIvE is compatible for comparison of similar tasks in other domains. The functional annotation task proposed an aim for biomedical text mining, which is to map complex concepts, which are expressed in raw text, to concepts defined in the ontologies. Overall, the BioCreAtIvE challenge was a major step forward in reflecting the advancements of the applications of text mining into biomedical domain (Krallinger *et al.*, 2008), in particular, the support of curation of biological databases that organize the raw biological data into amenable structures and associate biological data with experimental results contained in published literature.

Event and relation extraction

In the current post genomic era of the biology study, more and more studies have been focused on the relationships between genes and proteins. Through grouping genes by their functional relationships, both the analysis of the gene expression and database annotation can be aided. The grouping is currently divided into two ways. One way is based on how strong their associations are in the raw text. The other way is to use the relations between genes, proteins, or other biological entities and then to extract them from the text. Based on the previous accomplishment, it is suggested that the difficulty in extracting relations is different, as some can be extracted well but others cannot. The relationships that are very general, non-specific such as the gene groups relation between genes are easy to extract; on the other hand, specific relations that depend on their particular context, for example, the process of the GO code assignment in the annotation, remain challenging.

In contrast to NER, which is mainly related to the analysis of individual entity, event and relation extraction is relevant to a pair of, or even multiple, entities. The crucial characteristic of relation extraction is that it involves a pair of entities, which makes the analysis simpler than the event extraction, which is a more general *n*-ary prediction task for arbitrary numbers of arguments. In the structure of an event, there is a *trigger word*, which expresses the main event, and the arguments for the trigger word, which describe the detailed information relevant to the trigger word. The goal of relation extraction is to identify instances of a pre-defined type of relation between a pair of entities. While the types of the entity can be extremely specific, such as a gene, a protein or a drug, the type of relationship can be either highly general, for example, the expression of location for a pair of entities, or very specific such as the expression of the regulatory relationship, including inhibition or activation.

An example of the event and relation extraction is shown in Fig. 4. In the figure, different words are annotated with their corresponding ontological concepts shown in boxes. For instance, the word ‘activin’ is annotated with the ontological concept ‘Protein’. There may exist different types of pre-defined types of events and relations for the words, such as the ‘Activation’ (for the word ‘promotes’) and ‘Protein-Protein Interaction’ (for the word ‘interaction’). Such words with ontological annotations are nodes, while the edges between the nodes represent the detailed relationship between them. For instance, there is relationship of ‘Activation’ (for the word ‘promotes’) that ‘has Agent’ the ‘Protein’ (for the word ‘activin’).

Several approaches to extracting entity relationships have been used. To begin, some researchers use template-based syntactic patterns, which usually are regular expressions generated by domain experts, to extract specific relations from text (Yu *et al.*, 2002). One of the challenges in the hand-craft pattern is that a given event may be expressed in a variety of forms, such as the morphological, syntactic and lexical variations. To improve the performance, some syntactic generalization is often introduced. After the application of hand-craft pattern to extract candidate events, there is often an additional step of consolidation, where task-specific rules are applied to the candidate events. Another method is to use an automatic template. In this method, the templates, which are similar to manually generated regular-expression patterns, are automatically generalized patterns based upon the text close to the entity pairs, which have the relation of interest (Yu and Agichtein, 2003). The third method is a statistical approach that identifies the relationships by comparison to random predictions, evaluating the probability that the concepts will be found near to each other by chance (Friedman, 2002).

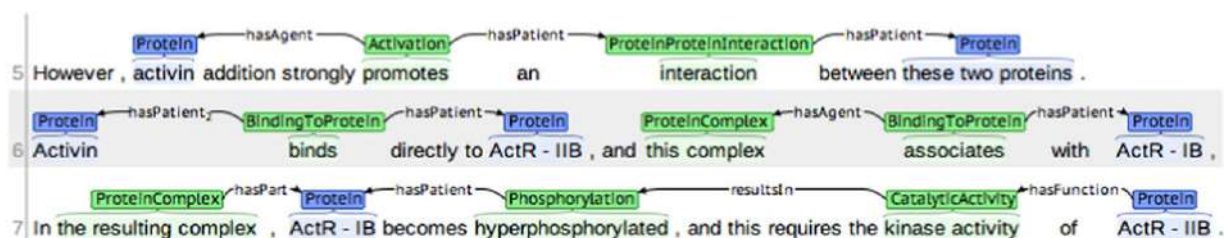


Fig. 4 Example of event and relation extraction.

BioNLP-shared tasks (BioNLP-ST)

Similar to the BioCreAtIvE challenge in the NER task, the BioNLP shared tasks series (BioNLP-ST) (Kim *et al.*, 2009, 2016, 2011a; Nédellec *et al.*, 2013) are organized to evaluate the progress of research on information extraction applications in the BioNLP domain. However, the focus of BioCreAtIvE and BioNLP-ST are different. While the BioCreAtIvE challenge assesses the simple representation of protein-protein interactions (PPI), the BioNLP-ST series are more centered on detailed biological relations, such as biological events (Kim *et al.*, 2011b). In addition, while the BioCreAtIvE challenges support the curation of PPI databases, the BioNLP-ST series are intended to support more detailed and structured databases, for example, pathway (Bader *et al.*, 2006) or Gene Ontology Annotation (GOA) databases (Camon *et al.*, 2004).

Among the participating systems in BioNLP-ST 13', only one system, TEES, submitted results for all the tasks except one (Nédellec *et al.*, 2013) and it was the first ranked system in four out of eight tasks, which showed its power in the event and relation extraction tasks. The TEES system is dependent on efficient machine learning approaches and features that are obtained from a full dependency analysis, generated by the applications of full dependency parsers into biomedical domain. For instance, many of the best-performing systems utilize the full dependency parses (Buyko *et al.*, 2009; Kilicoglu and Bergler, 2009; Sebastian *et al.*, 2009; Van Landeghem *et al.*, 2009). The pipeline for the system includes three steps: the first is *trigger recognition*, followed by the step of *argument detection*, the last is the step of *semantic post-processing*. For the trigger recognition step, the reason why the TEES separate the first step from the second step is that the TEES can use methods that are similar to the task of NER, with the exception of identifying the words for event trigger. With such approach, the problem of the detection of event arguments becomes the determination of whether the pairs, in the form of trigger-entity or trigger-trigger, are an instance of event arguments. Such steps are in essence classification tasks. The final step, semantic post-processing, is addressed through the rule-based approach by directly implementing the constraints defined the definition of each Shared Task. We refer the interested reader to the detailed explanation and implementation of TEES in publication in Björne *et al.* (2011) and Björne and Salakoski (2015).

Text summarization

Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks), and the generated summary can successfully imitate summaries generated by human beings, in other words, the summary should be concise and fluent while preserving key information content and overall meaning.

Generally, there are two main approaches to summarizing text documents, the extractive methods and abstractive methods. Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source. Abstractive text summarization involves generating entirely new phrases and sentences to capture the meaning of the source document. This is a more challenging approach, but is also the approach ultimately used by humans. Classical methods operate by selecting and compressing content from the source document.

We refer the interested reader to the book (Mani and Maybury, 1999) for detailed discussions of the text summarization, and the publication in Allahyari *et al.* (2017) for a survey of the text summarization techniques.

In the biomedical domain, the recent shared task on text summarization is the TAC BiomedSumm track (see "Relevant Websites section") held in 2014. The task challenged participants to leverage the set of citation sentences that reference a specific paper ('citations') for summarization, an important problem in BioNLP research (Lu *et al.*, 2007; Jin *et al.*, 2009).

The citations can be seen as a (community created) summary of an academic paper, and the set of citations is taken to summarize the key points of the referenced paper, and so reflects the importance of the paper within an academic community (Nakov *et al.*, 2004; Qazvinian and Radev, 2010), and they offers a new type of context that was not available at the time of authoring of the citation and the collection of citations to a reference paper adds an interpretative layer to the cited text.

Specifically, the track included identifying the text spans in the referenced papers reflecting the citations, classifying those spans into paper facets and then generating summary for the referenced papers based on the community discussion of their citations. This form of scientific summarization could be a component of a User Interface in which a user is able to hover over or click on a citation, which then causes a citance-focused faceted summary of the referenced paper to be displayed, or a full summary of the referenced paper taking into account the citations in all citing papers for that reference paper. Finally, this form of scientific summarization would allow a user to read the original reference paper, but with links to the subsequent literature that cites specific ideas of the reference paper.

Advanced Topics: Applications of Active Learning to Biomedical NLP Systems

The growth in the biomedical literature is paralleled by the growth in biomedical terminology, in which a single biomedical entity may have many different names and descriptions. The construction of an *ontology* is one way to address this issue of normalization. An ontology is a conceptual model that aims to support a consistent and unambiguous formal representation of a specific knowledge domain (Stevens *et al.*, 2000). Biomedical ontologies map biological concepts to their semantics, such as biological definitions and their relations to other biological concepts. By creating a commonly acceptable conceptual model, the adoption of an ontology ensures that data sets are machine readable and processable.

Many sophisticated BioNLP systems are based on machine learning. Machine learning studies the construction and analysis of systems that can learn from data. In BioNLP, machine learning algorithms can be designed and implemented to learn language patterns or rules from the limited amount of ontologically annotated biomedical documents. However, the research and implementation of BioNLP systems often require ‘gold-standard’ annotated corpora (Bada and Hunter, 2011), which contain biomedical documents that have been manually annotated with terms and relations defined in ontologies. The annotated corpora can subsequently be applied to the training of BioNLP systems (Tsuruoka *et al.*, 2005b).

To construct the ‘gold-standard’ annotated corpus, usually a group of human annotators first randomly select a pool of biomedical documents to be annotated. Based on their domain expertise, the annotators then manually identify and label the instances of document words or phrases with the proper ontological concepts defined in the ontology. After the manual annotation phase, an inter-annotator agreement (IAA) score, which measures the consistency of the expert annotators, is calculated for the annotated corpus. For instance, in a corpus construction project, acceptable IAA score may fall within the range of 66%–90%, depending on the difficulty of the annotation tasks and the level of annotator discrepancies (Thompson *et al.*, 2009). The annotated corpus could then be used for the training and evaluation of machine learning models. Such methodology is often referred as *passive learning*, and the documents are typically randomly chosen. However, the whole process of constructing annotated corpora is time-consuming and costly, as it requires substantial and tedious effort from the human annotators. Therefore, it is particularly valuable to construct the annotated corpus as quickly and economically as possible, using a directed sampling process (active learning), where the learner interactively queries the annotator for information about unannotated documents, and chooses the most informative document from which to learn, in order to generate as good a learner as possible with a minimum number of annotated documents.

Active learning is a subfield of supervised machine learning, in which the learning algorithm actively chooses the data it learns, in order to generate a learner that works as well as possible with the minimum amount of annotated data provided by the human annotator (Settles, 2012). This is especially useful for many supervised learning tasks, where it is very difficult, and time-consuming to obtain annotated data from an annotator. For instance, in the domain of information extraction, sophisticated information extraction systems should be trained using annotated documents. To produce such labeled documents, annotators who often have extensive knowledge in the specific domain markup entities and relations in text, which is a particularly time-consuming process. It is reported that, even for simple newswire stories, it can take half an hour or more to locate entities and relations (Settles *et al.*, 2008). And for tasks in other domains, such as annotating gene and protein mentions for biomedical information extraction, annotators must possess additional domain expertise.

Active learning has been studied in many natural language processing applications, such as word sense disambiguation (Chen *et al.*, 2013), named entity recognition (Tomanek and Hahn, 2009a,b, 2010), speech summarization (Zhang and Yuan, 2014) and sentiment classification (Kranjc *et al.*, 2015; Li *et al.*, 2012). Two popular approaches exist in the active learning method: the uncertainty-based approach (Lewis and Catlett, 1994) and committee-based approach (Seung *et al.*, 1992). The uncertainty-based approach is to label the most uncertain samples by using an uncertainty scheme such as entropy (Fu *et al.*, 2013). It has been shown, however, that the uncertainty-based approach may have worse performance than random selection (Schütze *et al.*, 2006; Tomanek *et al.*, 2009; Wallace *et al.*, 2010).

In biomedical information extraction, uncertainty-based active learning has been applied to the extracting PPIs task. For instance, Cui *et al.* (2009) adopted an uncertainty sampling-based approach to active learning, and Zhang *et al.* (2012) proposed maximum-uncertainty based and density-based sample selection strategies. While the extraction of PPI is concerned with a single event of type PPI, recent biomedical event extraction tasks (Kim *et al.*, 2009) involve multiple event types, and even hundreds of event types in the case of the Gene Regulation Ontology (GRO) task of BioNLP-ST’13 (Kim *et al.*, 2013).

In contrast, by using an ensemble approach, the committee-based active learning approach selects documents such that the documents’ label classification results mostly disagree with each other in the results of the committee of classifiers. The committee-based approach, however, has several issues in its application to event and relation extraction. As the event and relation extraction task is fundamentally different from applications of active learning in the other domains, as they are mainly focused on document classification. However, in a event extraction task, such as PPI extraction or gene regulation identification, the systems have to identify not only event keywords (also known as trigger words, e.g., regulation), but also event participants (also known as event arguments, e.g., gene or protein) within the input text; in addition, the systems need to extract relations between entities that are pre-defined in the given ontology. Hence, even if some systems provide confidence scores for their recognized events; these scores do not correspond to the probability that a particular event type is semantically present in a document, which is the probability of determining if a document belongs to that event type, as required in committee-based active learning.

Recently, there is research work, reported in Han *et al.* (2016a), of applying a committee-based active learning approach to biomedical event and relation extraction, by using an informativity score based system that evaluates the probability that a particular event type is semantically present in a document. The active learning method is evaluated using the BioNLP Shared Tasks datasets; and the results show that the event extraction system can be more efficiently trained than with other baseline methods.

The underlying principle in these methods is similar: to select the samples that are most ‘informative’, according to a certain measurement, to the underlying machine learning system. Moreover, there is a trend in active learning methods to consider whether the selected samples are ‘representative’ or not, as non-representative samples, such as outliers, may cause overfitting and undermine the training performance of the supervised learning framework. To select ‘representative’ examples and avoid outliers, different approaches have been proposed. For instance, information density may be used to guide the selection of documents (Settles and Craven, 2008). This approach explicitly models the distribution of average similarity for all the data point in the data

set with density weights, and has been shown to outperform uncertainty-based approaches. The informativeness of a sample is weighted by its average similarity to all other samples, and a sample with high similarity has high weight. This method was designed to avoid spending too much time on annotating outliers, which have low information density. We refer the interested reader to the detailed explanation in publication in [Settles and Craven \(2008\)](#).

On the other hand, there are also active learning methods that integrate unsupervised clustering techniques ([Kang et al., 2004](#); [Zhu et al., 2008](#); [Qian and Zhou, 2010](#)). For instance, [Kang et al. \(2004\)](#) they first clustered all samples, before active learning, by applying the k-means algorithm until it converges to form the clusters with pre-defined number, and selected documents close to the cluster centroids as the initial set for annotation, instead of beginning with a random subset of samples. In [Zhu et al. \(2008\)](#) they further develop this idea by incorporating information density to guide the document selection, implementing a k-nearest-neighbor-based density weight in order to circumvent the risk of selecting outliers.

Recently, there is research work of, reported in [Han et al. \(2016b\)](#), that select a mixture of both ‘representative’ and ‘informative’ documents from the clustering results. Rather than merely selecting the cluster centroids for annotation, the researchers select examples that are either near cluster centroids or near cluster boundaries, in order to improve the active learning performance.

Note that, the active learning method is similar to positive-unlabeled (PU) learning in that they both try to learn from the unlabeled dataset. In particular, the PU learning method first partitions the unlabeled dataset into four sets, namely, the reliable negative set, RN, the likely positive set, LP, the likely negative set, LN and the weak negative set, WN. Yet in active learning, such partitioning of the unlabeled dataset is not performed. We refer the interested reader to the publication in [Yang et al. \(2012, 2014\)](#).

Conclusion

In this article, we provide an overview of historical evolution of the natural language processing and summarize the common NLP sub-problems, together with their research progress in bioinformatics. Additionally, we discuss an advanced topic of applying the active learning into the NLP systems, in order to solve the practical issue of lacking training data for the systems.

See also: Data-Information-Concept Continuum From a Text Mining Perspective. Text Mining Basics in Bioinformatics. Text Mining for Bioinformatics Using Biomedical Literature. Text Mining Resources for Bioinformatics

References

- Ågel, V., 2006. Dependency and Valency: An International Handbook of Contemporary Research. 1. Walter de Gruyter.
- Agirre, E., Edmonds, P., 2007. Word Sense Disambiguation: Algorithms and Applications. 33. Springer Science & Business Media.
- Allahyari, M., Pouriyeh, S., Assefi, M., et al., 2017. Text summarization techniques: A brief survey. *International Journal of Advanced Computer Science and Applications* 8 (10), (arXivpreprint arXiv:1707.02268).
- Bacchiani, M., Riley, M., Roark, B., Sproat, R., 2006. Map adaptation of stochastic grammars. *Computer Speech & Language* 20 (1), 41–68.
- Bacchiani, M., Roark, B., Saraclar, M., 2004. Language model adaptation with map estimation and the perceptron algorithm. In: *Proceedings of HLT-NAACL 2004: Short Papers*, pp. 21–24. Association for Computational Linguistics.
- Backus, J.W., 1959. The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference. In: *Proceedings of the International Conference on Information Processing*.
- Bada, M., Hunter, L., 2011. Desiderata for ontologies to be used in semantic annotation of biomedical documents. *Journal of Biomedical Informatics* 44 (1), 94–101. doi:10.1016/j.jbi.2010.10.002. (Ontologies for Clinical and Translational Research).
- Bader, G.D., Cary, M.P., Sander, C., 2006. Pathguide: A pathway resource list. *Nucleic Acids Research* 34 (Suppl. 1), D504–D506. doi:10.1093/nar/gkj126.
- Bellegarda, J.R., 2004. Statistical language model adaptation: Review and perspectives. *Speech Communication* 42 (1), 93–108.
- Bender, O., Och, F.J., Ney, H., 2003. Maximum entropy models for named entity recognition. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 – vol. 4, CONLL '03*, pp. 148–151. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1119176.1119196>.
- Björne, J., Heimonen, J., Ginter, F., et al., 2011. Extracting complex biological events with rich graph-based feature sets. *Computational Intelligence* 27 (4), 541–557.
- Björne, J., Salakoski, T., 2015. Tees 2.2: Biomedical event extraction for diverse corpora. *BMC Bioinformatics* 16 (16), S4. doi:10.1186/1471-2105-16-S16-S4.
- Black, E., 1988. An experiment in computational discrimination of english word senses. *IBM Journal of Research and Development* 32 (2), 185–194.
- Booth, T.L., 1969. Probabilistic representation of formal languages. In: *Proceedings of the IEEE Conference Record of 10th Annual Symposium on Switching and Automata Theory*, pp. 74–81.
- Brants, T., 2000. Tnt: A statistical part-of-speech tagger. In: *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pp. 224–231. Association for Computational Linguistics.
- de Bruijn, B., Martin, J.D., 2002. Getting to the (c)ore of knowledge: Mining biomedical literature. *International Journal of Medical Informatics* 67 (1–3), 7–18. Available at: <http://dblp.uni-trier.de/db/journals/ijmi/ijmi67.html#BruijnM02>.
- Bulyko, I., Ostendorf, M., Stolcke, A., 2003. Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion volume of the Proceedings of HLT-NAACL 2003 – Short Papers*, vol. 2, pp. 7–9. Association for Computational Linguistics.
- Buyko, E., Faessler, E., Wermter, J., Hahn, U., 2009. Event extraction from trimmed dependency graphs. In: *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pp. 19–27. ACL.
- Camon, E., Magrane, M., Barrell, D., et al., 2004. The gene ontology annotation (GOA) database: Sharing knowledge in uniprot with gene ontology. *Nucleic Acids Research* 32 (Suppl. 1), D262–D266. doi:10.1093/nar/gkh021.
- Carnie, A., 2012. *Syntax: A Generative Introduction 3rd Edition and The Syntax Workbook Set*, *Introducing Linguistics*. Wiley. Available at: <https://books.google.com.sg/books?id=jhGKMAEACAAJ>.

- Carreras, X., Màrquez, L., Padró, L., 2003. A simple named entity extractor using adaboost. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 – vol. 4, CONLL '03, pp. 152–155. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1119176.1119197.
- Chen, Y., Cao, H., Mei, Q., Zheng, K., Xu, H., 2013. Applying active learning to supervised word sense disambiguation in medline. *Journal of the American Medical Informatics Association* 20 (5), 1001–1006. doi:10.1136/amiajnl-2012-001244.
- Chomsky, N., 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2 (3), 113–124.
- Chomsky, N., 1959. On certain formal properties of grammars. *Information and Control* 2 (2), 137–167.
- Chomsky, N., 1957. *Syntax Structures*. Walter de Gruyter.
- Church, K.W., Gale, W.A., 1991. A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilities of english bigrams. *Computer Speech & Language* 5 (1), 19–54.
- Church, K.W., 1988. A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the Second Conference on Applied natural language processing, pp. 136–143. Association for Computational Linguistics.
- Cohen, A.M., Hersh, W.R., Dubay, C., Spackman, K., 2005. Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts. *BMC Bioinformatics* 6 (1), 103. doi:10.1186/1471-2105-6-103. Available at: <http://www.biomedcentral.com/1471-2105/6/103>.
- Cui, B., Lin, H., Yang, Z., 2009. Uncertainty sampling-based active learning for protein-protein interaction extraction from biomedical literature. *Expert Systems with Applications* 36 (7), 10344–10350. doi:10.1016/j.eswa.2009.01.043.
- Diab, M., Resnik, P., 2002. An unsupervised method for word sense tagging using parallel corpora. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 255–262. Association for Computational Linguistics.
- Finkel, J.R., Grenager, T., Manning, C., 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, pp. 363–370. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1219840.1219885>.
- Florian, R., Ittycheriah, A., Jing, H., Zhang, T., 2003. Named entity recognition through classifier combination. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, CONLL '03, pp. 168–171. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1119176.1119201>.
- Franz, A., Brants, T., 2006. All our n-gram are belong to you, Google Machine Translation Team.
- Friedman, H.L.C., 2002. Mining terminological knowledge in large biomedical corpora. In: Proceedings of the Pacific Symposium on Biocomputing 2003, 3–7 January 2003, p. 415. Kauai, Hawaii: World Scientific.
- Fu, Y., Zhu, X., Li, B., 2013. A survey on instance selection for active learning. *Knowledge and Information Systems* 35 (2), 249–283. doi:10.1007/s10115-012-0507-8.
- Giménez, J., Márquez, L., 2004. Svmtool: A general pos tagger generator based on support vector machines. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Citeseer.
- Hanisch, D., Fluck, J., Mevissen, H.T., Zimmer, R., 2003. Playing biology's name game: Identifying protein names in scientific text. *Pacific Symposium on Biocomputing*, 403–414.
- Han, X., Kim, J.-j., Kwok, C.K., 2016a. Active learning for ontological event extraction incorporating named entity recognition and unknown word handling. *Journal of Biomedical Semantics* 7 (1), 22. doi:10.1186/s13326-016-0059-z.
- Han, X., Kwok, C.K., Kim, J.-j., 2016b. Clustering based active learning for biomedical named entity recognition. In: Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1253–1260. Available at: <https://doi.org/10.1109/IJCNN.2016.7727341>.
- Heafield, K., 2011. Kenlm: Faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, pp. 187–197. Association for Computational Linguistics.
- Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P., 2013. Scalable modified kneser-ney language model estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (vol. 2: Short Papers), vol. 2, pp. 690–696.
- Hindle, D., Rooth, M., 1993. Structural ambiguity and lexical relations. *Computational Linguistics* 19 (1), 103–120.
- Hirschman, L., Park, J.C., Tsujii, J., Wong, L., Wu, C.H., 2002. Accomplishments and challenges in literature data mining for biology. *Bioinformatics* 18 (12), 1553–1561.
- Hirschman, L., Yeh, A., Blaschke, C., Valencia, A., 2005. Overview of BioCreative: Critical assessment of information extraction for biology. *BMC Bioinformatics* 6 (Suppl. 1), S1. doi:10.1186/1471-2105-6-S1-S1.
- Hsu, B.-J., 2007. Generalized linear interpolation of language models. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU, pp. 136–140.
- Indurkha, N., Damerau, F.J., 2010. *Handbook of Natural Language Processing*. 2. CRC Press.
- Jin, F., Huang, M., Lu, Z., Zhu, X., 2009. Towards automatic generation of gene summary. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, pp. 97–105. Association for Computational Linguistics.
- Kang, J., Ryu, K.R., Kwon, H.-C., 2004. Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 384–388. Available at: http://dx.doi.org/10.1007/978-3-540-24775-3_46.
- Kilgarriff, A., Rosenzweig, J., 2000. Framework and results for english senseval. *Computers and the Humanities* 34 (1–2), 15–48.
- Kilgarriff, A., 2000. *Wordnet: An Electronic Lexical Database*. The MIT Press.
- Kilicoglu, H., Bergler, S., 2009. Syntactic dependency based heuristics for biological event extraction. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 119–127. ACL.
- Kim, J.-J., Han, X., Lee, V., Rebholz-Schuhmann, D., 2013. Gro task: Populating the gene regulation ontology with events and relations. In: Proceedings of the BioNLP Shared Task 2013 Workshop, pp. 50–57. Sofia, Bulgaria: Association for Computational Linguistics. Available at: <http://www.aclweb.org/anthology/W13-2007>.
- Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., Tsujii, J., 2009. Overview of BioNLP'09 shared task on event extraction. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 1–9. Boulder, Colorado: Association for Computational Linguistics. Available at: <http://www.aclweb.org/anthology/W09-1401>.
- Kim, J.-D., Ohta, T., Tateisi, Y., Tsujii, J., 2003. Genia corpora semantically annotated corpus for bio-textmining. *Bioinformatics* 19 (Suppl. 1), i180–i182.
- Kim, J.-D., Pyysalo, S., Ohta, T., et al., 2011a. Overview of bionlp shared task 2011. In: Proceedings of BioNLP Shared Task 2011 Workshop, pp. 1–6. Portland, Oregon, USA: Association for Computational Linguistics.
- Kim, J.-D., Wang, Y., Colic, N., et al., 2016. Refactoring the genia event extraction shared task toward a general framework for ie-driven kb development. In: Proceedings of the 4th BioNLP Shared Task Workshop, pp. 23–31. Berlin, Germany: Association for Computational Linguistics.
- Kim, J.-D., Wang, Y., Takagi, T., Yonezawa, A., 2011b. Overview of genia event task in BioNLP shared task 2011. In: Proceedings of BioNLP Shared Task 2011 Workshop, 2011, pp. 7–15. Portland, Oregon, USA: Association for Computational Linguistics. Available at: <http://www.aclweb.org/anthology/W11-1802>.
- Klein, D., Manning, C.D., 2003a. Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics.
- Klein, D., Manning, C.D., 2003b. A parsing: Fast exact viterbi parse selection. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 40–47. Association for Computational Linguistics.
- Krallinger, M., Morgan, A., Smith, L., et al., 2008. Evaluation of text-mining systems for biology: Overview of the second BioCreative community challenge. *Genome Biology* 9 (Suppl. 2), S1. doi:10.1186/gb-2008-9-s2-s1. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2559980&tool=pmcentrez&rendertype=abstract>.
- Kranjc, J., Smailović, J., Podpean, V., et al., 2015. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the cloudflows platform. *Information Processing & Management* 51 (2), 187–203. doi:10.1016/j.ipm.2014.04.001.

- Van Landeghem, S., Saeys, Y., De Baets, B., *et al.*, 2009. Analyzing text in search of bio-molecular events: A high-precision machine learning framework. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, pp. 128–136. Association for Computational Linguistics.
- Lesk, M., 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems documentation, pp. 24–26. ACM.
- Lewis, D.D., Catlett, J., 1994. Heterogenous uncertainty sampling for supervised learning. In: Proceedings of the Eleventh International Conference on International Conference on Machine Learning, ICML'94, pp. 148–156. San Francisco, CA: Morgan Kaufmann Publishers Inc.
- Liu, X., Gales, M.J.F., Woodland, P.C., 2013. Use of contexts in language model interpolation and adaptation. *Computer Speech & Language* 27 (1), 301–321.
- Li, S., Ju, S., Zhou, G., Li, X., 2012. Active learning for imbalanced sentiment classification. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 139–148.
- Lu, Z., Cohen, K.B., Hunter, L., 2007. Generif quality assurance as summary revision. In: Proceedings of the Biocomputing 2007, World Scientific, pp. 269–280.
- Mani, I., Maybury, M.T., 1999. Advances in Automatic Text Summarization. MIT press.
- Manning, C.D., Schütze, H., 1999. Foundations of Statistical Natural Language Processing. MIT Press.
- Marcus, M.P., Marcinkiewicz, M.A., Santorini, B., 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* 19 (2), 313–330.
- Mayfield, J., McNamee, P., Piatko, C., 2003. Named entity recognition using hundreds of thousands of features. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, CONLL'03, pp. 184–187. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1119176.1119205>.
- McCallum, A., Li, W., 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, CONLL'03, pp. 188–191. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1119176.1119206>.
- De Meulder, F., Daelemans, W., 2003. Memory-based named entity recognition using unannotated data. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, CONLL'03, pp. 208–211. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1119176.1119211>.
- Nadas, A., 1984. Estimation of probabilities in the language model of the IBM speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32 (4), 859–861.
- Nakov, P.I., Schwartz, A.S., Hearst M., 2004. Citances: Citation sentences for semantic analysis of bioscience text. In: Proceedings of the SIGIR, vol. 4, pp. 81–88.
- Navigli, R., 2009. Word sense disambiguation: A survey. In: Proceedings of the ACM Computing Surveys (CSUR), 41 (2), p. 10.
- Nédellec, C., Bossy, R., Kim, J.-D., *et al.*, 2013. Overview of bionlp shared task 2013. In: Proceedings of the BioNLP Shared Task 2013 Workshop, pp. 1–7. Sofia, Bulgaria: Association for Computational Linguistics. Available at: <http://www.aclweb.org/anthology/W13-2001>.
- O'Connor, B., Krieger, M., Ahn, D., 2010. Tweetmotif: Exploratory search and topic summarization for twitter. In: Proceedings of the International AAAI Conference on Web and Social Media, ICWSM, pp. 384–385.
- Petrov, S., Barrett, L., Thibaux, R., Klein, D., 2006. Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 433–440.
- Porter, M.F., 1980. An algorithm for suffix stripping. *Program* 14 (3), 130–137.
- Qazvinian, V., Radev, D.R., 2010. Identifying non-explicit citing sentences for citation-based summarization. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 555–564.
- Qian, L., Zhou, G., 2010. Clustering-based stratified seed sampling for semi-supervised relation classification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 346–355.
- Quinlan, J.R., 2014. C4.5: Programs for Machine Learning. Elsevier.
- Ramshaw, L.A., Marcus, M.P., 1999. Text chunking using transformation-based learning. In: Armstrong, S., Church, K., Isabelle, P., *et al.* (Eds.), *Natural Language Processing Using Very Large Corpora*. Springer, pp. 157–176.
- Ratnaparkhi, A., 1996. A maximum entropy model for part-of-speech tagging. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Schabes, Y., Abeille, A., Joshi, A.K., 1988. Parsing strategies with 'lexicalized' grammars: Application to tree adjoining grammars. In: Proceedings of the 12th Conference on Computational linguistics, Association for Computational Linguistics. vol. 2, pp. 578–583.
- Schabes, Y., 1990. Mathematical and Computational Aspects of Lexicalized Grammars. Philadelphia, PA: University of Pennsylvania.
- Schnabel, T., Schütze, H., 2014. FLORS: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics* 2, 15–26.
- Schütze, H., Manning, C.D., Raghavan, P., 2008. Introduction to Information Retrieval. 39. Cambridge University Press.
- Schütze, H., Velipasaoglu, E., Pedersen, J.O., 2006. Performance thresholding in practical text classification. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM'06, pp. 662–671. New York, NY, USA: ACM.
- Sebastian, R., Hong, W.C., Toshihisa, T., Jun'ichi, T., 2009. A Markov logic approach to bio-molecular event extraction. In: Proceedings of the Workshop on BioNLP, pp. 41–49.
- Settles, B., 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6 (1), pp. 1–114.
- Settles, B., Craven, M., 2008. An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP'08, pp. 1070–1079. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <http://dl.acm.org/citation.cfm?id=1613715.1613855>.
- Settles, B., Craven, M., Friedland, L., 2008. Active learning with real annotation costs. In: Proceedings of the NIPS Workshop on Cost-Sensitive Learning, pp. 1–10.
- Seung, H.S., Oppor, M., Sompolinsky, H., 1992. Query by committee. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT'92, pp. 287–294. New York, NY, USA: ACM. Available at: <https://doi.org/10.1145/130385.130417>.
- Smith, L., Rindfleisch, T., Wilbur, W.J., 2004. Medpost: A part-of-speech tagger for biomedical text. *Bioinformatics* 20 (14), 2320–2321.
- Smith, L., Tanabe, L., Ando, R., *et al.*, 2008. Overview of BioCreative II gene mention recognition. *Genome Biology* 9 (Suppl. 2), S2.
- Stevens, R., Goble, C.A., Bechhofer, S., 2000. Ontology-based knowledge representation for bioinformatics. *Briefings in Bioinformatics* 1 (4), 398–414. doi:10.1093/bib/1.4.398.
- Stolcke, A., 2002. SRILM – An extensible language modeling toolkit. In: Proceedings of the Seventh International Conference on Spoken Language Processing.
- Thompson, P., Iqbal, S.A., McNaught, J., Ananiadou, S., 2009. Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinformatics* 10 (1), 349. doi:10.1186/1471-2105-10-349.
- Sang, E.F.T.K., De Meulder, F., 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, CoNLL'03, vol. 4, pp. 142–147. Stroudsburg, PA, USA: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/1119176.1119195>.
- Tomanek, K., Hahn, U., 2009a. Reducing class imbalance during active learning for named entity annotation. In: Proceedings of the Fifth International Conference on Knowledge Capture, K-CAP'09, pp. 105–112. New York, NY, USA: ACM. Available at: <http://doi.acm.org/10.1145/1597735.1597754>.
- Tomanek, K., Hahn, U., 2009b. Semi-supervised active learning for sequence labeling. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1039–1047.
- Tomanek, K., Hahn, U., 2010. A comparison of models for cost-sensitive active learning. In: Proceedings of the International Conference on Computational Linguistics (Coling): Posters, pp. 1247–1255.

- Tomanek, K., Laws, F., Hahn, U., Schütze, H., 2009. On proper unit selection in active learning: Co-selection effects for named entity recognition. In: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing, HLT'09, pp. 9–17. PA, USA: Association for Computational Linguistics, Stroudsburg.
- Toutanova, K., Klein, D., Manning, C.D., Singer, Y., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 173–180. Association for Computational Linguistics.
- Tsuruoka, Y., Tateishi, Y., Kim, J.-D., *et al.*, 2005a. Developing a robust part-of-speech tagger for biomedical text. In: Proceedings of the Panhellenic Conference on Informatics, pp. 382–392. Springer.
- Tsuruoka, Y., Tateishi, Y., Kim, J.-D., *et al.*, 2005b. Developing a robust part-of-speech tagger for biomedical text. In: Bozanis, P., Houstis, E. (Eds.), *Advances in Informatics* 3746. Heidelberg: Springer Berlin, pp. 382–392. Available at: http://dx.doi.org/10.1007/11573036_36.
- Wacholder, N., 2003. Spotting and discovering terms through natural language processing. *Information Retrieval* 6 (2), 277–281. doi:10.1023/a:1023940422865.
- Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A., 2010. Active learning for biomedical citation screening. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'10, pp. 173–182. New York, NY, USA: ACM. Available at: <http://doi.acm.org/10.1145/1835804.1835829>.
- Witten, I.H., Bell, T.C., 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory* 37 (4), 1085–1094.
- Yang, P., Li, X., Chua, H.-N., Kwoh, C.-K., Ng, S.-K., 2014. Ensemble positive unlabeled learning for disease gene identification. *PLOS ONE* 9 (5), 1–11. doi:10.1371/journal.pone.0097079.
- Yang, P., Li, X.-L., Mei, J.-P., Kwoh, C.-K., Ng, S.-K., 2012. Positive-unlabeled learning for disease gene identification. *Bioinformatics* 28 (20), 2640–2647. doi:10.1093/bioinformatics/bts504.
- Yarowsky, D., 1995. Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 189–196.
- Yeh, A.S., Hirschman, L., Morgan, A.A., 2003. Evaluation of text data mining for database curation: Lessons learned from the KDD Challenge Cup. *Bioinformatics* 19 (Suppl. 1), i331–i339.
- Yu, H., Agichtein, E., 2003. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics* 19 (Suppl. 1), i340–i349. doi:10.1093/bioinformatics/btg1047.
- Yu, H., Hatzivassiloglou, V., Friedman, C., Rzhetsky, A., Wilbur, W.J., 2002. Automatic extraction of gene and protein synonyms from medline and journal articles. In: Proceedings of the AMIA Symposium, American Medical Informatics Association, p. 919.
- Zhang, H.-T., Huang, M.-L., Zhu, X.-Y., 2012. A unified active learning framework for biomedical relation extraction. *Journal of Computer Science and Technology* 27 (6), 1302–1313. doi:10.1007/s11390-012-1306-0.
- Zhang, J., Yuan, H., 2014. A certainty-based active learning framework of meeting speech summarization. *Computer Engineering and Networking* 277, 235–242.
- Zhong, Z., Ng, H.T., 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In: Proceedings of the ACL 2010 System Demonstrations, Association for Computational Linguistics, pp. 78–83.
- Zhu, J., Wang, H., Yao, T., Tsou, B.K., 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Coling 2008 Organizing Committee, pp. 1137–1144. Manchester, UK. Available at: <http://www.aclweb.org/anthology/C08-1143>.

Further Reading

- Ananiadou, S., Kell, D.B., Tsujii, J.I., 2006. Text mining and its potential applications in systems biology. *Trends in Biotechnology* 24 (12), 571–579.
- Hunter, L., Cohen, K.B., 2006. Biomedical language processing: What's beyond PubMed? *Molecular Cell* 21 (5), 589–594.
- Jurafsky, D., Martin, J.H., 2014. *Speech and Language Processing*. 3. London: Pearson.
- Krallinger, M., Leitner, F., Valencia, A., 2010. Analysis of biological processes and diseases using text mining approaches. *Bioinformatics Methods in Clinical Research*. 341–382.
- Nadkarni, P.M., Ohno-Machado, L., Chapman, W.W., 2011. Natural language processing: An introduction. *Journal of the American Medical Informatics Association* 18 (5), 544–551.
- Olsson, F., 2009. A literature survey of active machine learning in the context of natural language processing.
- Settles, B., 2012. Active learning, synthesis lectures on artificial intelligence and machine learning, 6 (1), pp. 1–114.

Relevant Websites

- https://www.nlm.nih.gov/bsd/index_stats_comp.html
Detailed Indexing Statistics.
- <http://flybase.org/>
FlyBase Homepage.
- <http://www.geniaproject.org/>
Genia Project.
- <http://www.nltk.org>
Natural Language Toolkit.
- <https://tartarus.org/martin/PorterStemmer/>
Porter Stemming Algorithm - Tartarus.
- <http://nlp.stanford.edu/software/tokenizer.shtml>
Stanford Tokenizer.
- <https://tac.nist.gov/2014/BiomedSumm/index.html>
TAC 2014 Biomedical Summarization Track.
- <http://sentiment.christopherpotts.net/tokenizing.html>
Tokenizing
Sentiment Symposium Tutorial
christopherpotts.net.

Text Mining Basics in Bioinformatics

Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Mimmo Parente, University of Salerno, Fisciano, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

One of the main goal of Text Mining (TM) is to discover relevant information in the text by transforming it into data that can then be analyzed. However, though correct, this definition does not convey the real relevance, efficacy, and role that TM plays in bioinformatics. Actually, the scientific production in this research field has reached huge proportions, even if we separately consider single research sub-fields. In the last decade, the research articles collected in public repositories, such as PubMed, are growing very much. Fig. 1 points out that the number of publications per year on Bioinformatics are doubled in 2014 with respect to 2008. In addition, if we consider the cumulative number of publications the growth became impressive. In this scenario, the availability of massive unstructured material raised the interest for literature-based applications in order to carry out useful findings for further scientific investigations in bioinformatics.

The excessive competition in science may also have adverse effects on research integrity, causing also the urgent need to discover and discard *bad science*, see e.g. Fanelli, 2009. In particular, in biomedicine it is fundamental to recover all the valid scientific literature concerning molecules, genes, proteins, tissues, cells, etc., to find the origin and causes of diseases. In Fig. 2, we can see the growing trend of applying text-mining.

TM, as a mere discipline, has already won the challenge for the identification, normalization, and disambiguation of key entities like the above mentioned, and it also has demonstrated to be a winning strategy to investigate the relations among those entities (Ananiadou et al., 2010). However, it is also important to note that recent developments and advances in the field of TM have put forward the limits of what is commonly considered *basics* knowledge in the area. The aim of this survey is hence to expose the up-to-date basics of TM, enhanced with recent developments in successful application scenarios.

Just to provide an evidence of the most trending topics related to text-mining in bioinformatics, we have collected articles published on PubMed in the period 2006–2017 by indexing abstracts and keywords, and we have extracted the word-cloud shown in Fig. 3. As we can see, the most relevant trends regard Machine Learning applications, Information and Relation Extraction, Named Entity Recognition, Information Retrieval and others.

Summary

The article is structured as follows: Section “Fundamentals” introduces the methods addressing linguistic analysis and other tasks for enabling text-mining applications; Section “Applications” provides a list of some existing solutions (e.g., Information Retrieval, Summarization, etc.) for each category of text-mining application in bioinformatics; Section “Evaluation Techniques and Metrics” provides the methods to evaluate text mining applications; then, Section “Case Study: Text Mining for Simplifying Biomedical Literature Browsing” illustrates a case study of literature-based application simplifying biomedical literature browsing. Finally, the conclusions summarizes the key points of the article.

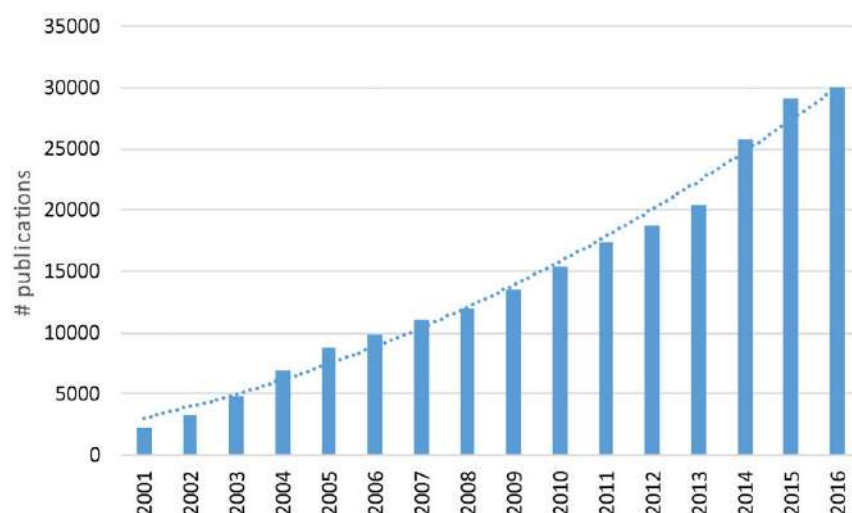


Fig. 1 The number of publications in PubMed on Bioinformatics.

Sentence Detection

Sentence detection (or Sentence Boundary Detection – SBD) is the problem of NLP to identify the start and the end of a sentence in a given document. Generally, it is a rather complex task because of the presence of misspellings, punctuation errors, and incomplete phrases. In bioinformatics, this task is even more complicated mainly due to the following concerns:

- The ambiguity in the use of periods, as closing an abbreviation (Prof.), marking the individual letters of an abbreviation (t.b.d.), indicating the rational parts of the real numbers (2.18) and so on.
- The mining of periods, like when the word *Q.E.D* is written at the end of a phrase. In this case, the punctuation defines the end of both the abbreviation and the sentence.
- Some general rules for delimiting the boundaries of a sentence are not always valid in biomedical texts, e.g., a sentence may start with a lowercase letter.
- The presence in the biomedical field of many acronyms and abbreviations.

Let us underline that correct sentence detection impacts also on other NLP tasks determining the structural role of words in sentences and phrases, such as Part-of-Speech tagging.

Tokenization

Tokenization is the problem of NLP devoted to dividing the text into a set of tokens. Tokens are approximately words but include also punctuation marks, date, time, fraction, person title, range and so on. The process uses a set of regular expressions to identify the token delimiters and special cases, occurring more frequently in biomedical text. The technique of tokenization can be deployed in different hierarchical flavours: text can be broken up into paragraphs, sentences, words, syllables, or phonemes. Generally, one token refers to one term or to one word, but it could also be an instance of a sequence of characters grouped together as a unique semantic unit. As an example, *type* is the class of tokens that contains the same sequence of characters.

As with sentence detection, even for tokenization, there are several problems to deal with. To understand whether to delete a punctuation character or not because it is not part of the word to be extracted as tokens: for example, the period at the end of the sentence. In many cases, when the period is part of an abbreviation or of a numeric expression, it should not be deleted. Even the case of the dashes is complex to deal with because in the biomedical field it can mean the absence of an element, that is, “- water” (less water) and thus it means to eliminate not only the dashes but also the next token. On the other hand, it can be a compound word and then everything becomes a unique token. The different applications have to hence find the right strategy to not cause errors in the mining of the text, thus a preliminary analysis could be foreseen, according to what counts as a token depending on the application domain.

Part-of-Speech Tagging

The technique of Part-of-Speech (PoS) tagging consists in assigning a role, among verb, noun or adjective, to every word occurring in the phrases. Most used approaches can be categorized as follows.

- *Rule-based taggers* (Greene and Rubin, 1971), tries to tag every word using hand-written rules: for example, a word which follows a determiner and an adjective must be a noun.
- *Stochastic (probabilistic) approach* (Màrquez and Rodriguez, 1998), adopts a training corpus to choose among the most feasible tags for a word. All these methods use massively the Markov models.
- *Transformation-based approach*, combine the two previous approaches. The most likely tag on a training corpus is considered, and then a particular set of rules is adopted to assess the validity of the tag, and in case it is modified to something else. The rules generated in this fashion, are then added to the set and taken into consideration in the next tag selection. One example of such tagger is the Brill tagger (Brill, 1992).

The PoS tagging on medical texts is complicated by the ambiguity of words. A term that can generally be an adjective, in the medical domain, may refer to a very precise medical condition, such as the word “cold”. Various specialized tagging tools for biomedical text exist in the literature, including MedPOST (Smith *et al.*, 2004) and LingPipe.

Stemming and Lemmatize

Stemming and Lemmatize both reduce inflectional and corresponding derivated related forms of a word to a common base form.

Stemming eliminates variations of grammatical forms of the same word (e.g., “connection” – “connect”, “computing” – “compute”, etc.). The stemming process is finalized to improve the effectiveness of the text process to avoid mismatches both in prefixes and suffixes. In fact, these can cause a mismatch or even miss of relevant information. Lemmatization is close to stemming. It removes inflectional endings and returns the lemma that is the base form or the dictionary form of a word. As an example, for the token “saw”, the lemmatization technique returns both “see” or “saw”, depending on whether the token is used as a verb or as a name.

Syntactic Structure

The syntactic structure is the problem of NLP of determining the structural role of words in sentences and phrases. It requires grammatical analysis of sentences and organizes the words in a manner showing the relationship among them. For example, in the text “Studying genetic circuit dynamics with fluorescence microscopy”, the phrase “with fluorescence microscopy” is associated with “studying”, not with “genetic circuit dynamics”. Obviously, this interpretation is very simple for humans, but more difficult for computers, because there are many syntactic ambiguities. The ambiguity in a sentence can contain an ambiguous phrase but with only one or many different interpretation.

Text Disambiguation

Text Disambiguation is the process that identifies the sense, i.e. the meaning, of a word in a sentence, when the word has multiple meanings. In the biomedical domain, it is a very critical task, since when the words are ambiguous, clearly the process of identification of key entities, like genes, proteins, cells, diseases, and so on is much more difficult.

An example of a very effective disambiguation tools is Wikify! (Mihalcea and Csomai, 2007). Used in biomedical scenarios, it extracts a model to represent the text. By exploiting the wikification services, it determines the meaning of the text and from this extracts the main concepts (De Maio *et al.*, 2015a). In particular, by using the Wikipedia knowledge base, it extracts a set of pairs $\langle \text{topic}, \text{relevance} \rangle$, where *topic* is a Wikipedia article, representing the meaning of the content, along with a corresponding membership degree, called *relevance*, ranging in the $[0,1]$ interval. For example, the following sentence is extracted from a PubMed article (the wikified terms will be bold-ed and underlined in the snippet):

“Extremely atrophic maxillae can be considered the most important indication for three-dimensional maxillary reconstruction. Different bone-augmentation techniques have been suggested to accomplish this. [...]”

For the above example, the following pairs $\langle \text{topic}, \text{relevance} \rangle$ are given as output which characterize the meaning of the text:

1. $\langle \text{Maxilla}; 0.93 \rangle$,
2. $\langle \text{Atrophy}; 0.83 \rangle$,
3. $\langle \text{Reconstructivessurgery}; 0.73 \rangle$,
4. $\langle \text{Threedimensionalspace}; 0.71 \rangle$,
5. ...

Applications

Some of the most investigated applications of text mining in Bioinformatics are listed below:

- Information Retrieval (IR), devoted to obtain relevant information from a collection of information resources and a user query;
- Document Classification (DC), which assigns one or more classes or categories to a document;
- Named Entity Recognition/Normalization (NER/NEN), devoted to extract named entity from unstructured or semi-structured machine-readable documents;
- Summarization (SUM), which synthesizes input text covering all contents of analyzed documents;

In this section, we detail text mining applications for the biomedical domain by introducing their underlying functional goals and existing solutions at the state of the art. In addition, we will describe (see Section “Others Applications”) some other applications of text-mining not categorized in the classes aforementioned.

Information Retrieval

Over the years different approaches have been studied and explored in the field of IR. The basic keyword-based IR approach simply checks the presence or absence of the words in the query input. More recent researches use mathematical techniques to calculate the relevance of a document and how much a given keyword reflects the actual relevance of a given document with respect to the query. Others expand the user’s queries with related synonyms or alternative names into one conceptual query, based on a controlled vocabulary.

The large volume and the rapid growth of biomedical literature represent a significant barrier for biomedical data retrieval. A set of IR applications in biomedical domain is listed in Table 1, each along with features and distinguishing functionalities.

Many applications such as PubMed and EuropePMC have a local database on which to search the scientific manuscripts. PubMed (see Relevant Website), gives access to open source full text articles and abstracts from the MEDLINE database, and does not include the full-text of journal articles but only provides short summaries. This is different from EuropePMC policy that processes full-text documents. Other applications, like PolySearch2, integrate multiple databases containing both text and sequence data. This application identifies relationships between biomedical entity such as human diseases, genes, proteins, and so on. A similar approach is PIE the search (Protein Interaction information Extraction) that retrieves Protein-protein interactions from PubMed or from other manually provided scientific articles (see Relevant Website section for PubMed, EuropePMC, PolySearch).

Table 1 Recent IR applications

Name	Content	Description
PubMed (McEntyre and Lipman, 2001)	Abstract	Given a user's query, retrieves related scientific publications.
GoPubMed (Doms and Schroeder, 2005)	Abstract	Retrieves information matching the query with biomedical background knowledge
PolySearch2 (Liu <i>et al.</i> , 2015)	Abstracts, databases	Retrieves information according to particular patterns of queries
Europe PMC (Consortium <i>et al.</i> , 2014)	Full text	Explore protein, gene, species and disease records directly from articles
PIE (Kim <i>et al.</i> , 2012)	Query	Finds protein-protein interaction articles from PubMed

Table 2 Recent DC applications

Name	Content	Description
MaxMatcher (Zhou <i>et al.</i> , 2006)	Documents	Dictionary-based biological concept extraction
BioDi (Chebil <i>et al.</i> , 2013)	Documents	Classifies biomedical documents with the MeSH ontology
MetaMap (Aronson and Lang, 2010)	Documents	Maps biomedical documents to the UMLS Metathesaurus concepts

Additional features are related to the classification of results based on contextual information. An example is provided by *GoPubMed* that hierarchically classifies retrieved documents. In particular it assigns each document to a concept of a domain specific ontology or a thesaurus, i.e., Gene Ontology or Medical Subject Headings (MeSH). This assignment to ontology concepts facilitates the navigation among the returned documents, by reducing the search time (see Relevant Website for *GoPubMed*, Gene Ontology, MeSH and Gene Ontology).

Further details on tools for information retrieval from the biomedical literature can be found in Roberts *et al.* (2016).

Document Classification

The search of biomedical documentations plays an important role in biomedical research because much information exists in the form of scientific publications. Furthermore, the number of biomedical publications grows rapidly and consequently it becomes difficult for researchers to quickly find the biomedical publications of interest. A solution is represented by the automatic classification of scientific publications.

Document Classification (DC) is the process of classifying a document as belonging to one or more categories or classes. Documents may be classified according to a very simple feature, like their subjects, or according to other attributes such as document type, author, printing year, etc. Thus, automated biomedical document classification, aiming to identify publications relevant for a specific research field, is an important task that has attracted much interest in the years (Huang and Lu, 2016; Jiang *et al.*, Effective; Lakiotaki *et al.*, 2013; Wu *et al.*, 2017). Table 2 lists recent DC applications in the biomedical domain. MaxMatcher and MetaMap are biological concept extractor, that extract all concepts belonging to Unified Medical Language System (UMLS) meta-thesaurus in order to accordingly classify the documents. Similarly, BioDi that classifies documents with the Medical Subject Headings (MeSH) thesaurus (see Relevant Website section for MaxMatcher and MetaMap).

Named Entity Recognition/Normalization

Named entity recognition (NER) arranges specific semantic classes, so called named entity, in a text such as the names of persons, organizations, places, etc. In biomedicine, the most studied entities have been gene and protein names. Also others have been studied as well, including disease and drug names, cells and types and other terms in the biomedical domain.

The application of NER activity in biomedical information extraction is one of the most important tasks (Han *et al.*, 2016; Lossio-Ventura *et al.*, 2016; Yu *et al.*, 2016). The accuracy of this process is crucial for follow-up researches.

The early biomedical NER approaches (listed in Table 3) are typically classified into three main categories, several of which are widely applied:

- *Rule-based approaches* consist in the manually or automatically construction of rules or patterns matching them in literature to find entities of interest (Yu *et al.*, 2016; Lai *et al.*, 2016). Example is *PPinterfinder*.
- *Lexicon-based approaches* consist in the usage of lexicons, dictionaries (e.g., names, aliases, symbols, etc.) to find specific terms in the analyzed text (Mrabet *et al.*, 2016; Dong *et al.*, 2016). Examples are *FACTA+*, and *G-Bean*.
- *Machine learning based approaches* consist in automatically learn to find entities using specific features that distinguish between features for the training set and those for the testing set (Vijay and Sridhar, 2016; Al-Hegami *et al.*, 2017). Examples are *PIE*, *PolySearch2*, *SCAIVIEW*, and *PCorral* (see Relevant Website section for *PPinterfinder*, *FACTA+*, *PIE*, *PolySearch2*, *SCAIVIEW*, *PCorral*).

Table 3 Recent NER applications

Name	Content	Description
PIE (Kim <i>et al.</i> , 2012)	Query	Finding protein-protein interaction articles from PubMed
PolySearch2 (Liu <i>et al.</i> , 2015)	Full-text	Extracts relations between diseases, genes, drugs, metabolites and toxin
G-Bean (Wang <i>et al.</i> , 2014)	Abstracts	Query documents in database
SCAIView (Malhotra <i>et al.</i> , 2015)	Abstracts	Searches for genes related to sclerosis and programs and drugs aimed
FACTA + (Tsuruoka <i>et al.</i> , 2011)	Abstracts	Discovers and visualizes indirect relations between biomedical concepts
PCorral (Li <i>et al.</i> , 2013)	Protein name	Merges IR and IE (information extraction) from MEDLINE database
PPInterFinder (Raja <i>et al.</i> , 2013)	Query + concepts	Extracts information about protein – protein Interactions from MEDLINE abstracts

Table 4 Recent SUM applications

Name	Content	Description
EntityRank (Schulze and Neves, 2016)	Abstract	Graph-based summarization algorithms for multi-document summarization for the biomedical domain
LTR (Shang <i>et al.</i> , 2014)	Full-text	A summarization method based on learning to rank

A research trend close to NER activity is the Named Entity Normalization (NEN) that is the activity to normalize all named entities in the text. It returns a specific label that is an entry of a domain specific database, ontology and so on. In biomedicine, this activity has been widely studied and explored in the research field of genes and proteins, with a certain success degree (e.g., many species have genes with the same name) (Leaman and Lu, 2016; Lou *et al.*, transition).

Summarization

Summarization (SUM) is the activity that takes in input a document or a set of documents, and outputs a synthesis of analyzed text that covers all the contents of the documents. The increasing amount of available biomedical publications makes the summarization a very important task because it is very hard to quickly find the right information. Considered that, generally, a search in PubMed for example for the gene *p53* returns about 50.000 publications, the summarization process provides more benefits regarding coverage and speed for future scientific searches. Furthermore, the use of domain knowledge to disambiguate domain specific terms is fundamental to create accurate summaries.

Examples (see Table 4) of recent SUM applications in biomedical domain are *EntityRank* and *LTR* (Learn To Rank Summary). The former works only on abstracts from articles retrieved on PUBMED and provides summaries specific to the users information needs. Differently, the latter provides a gene summary extraction analyzing the full-text document. Moreover it approaches the summarization activity as a ranking problem and applies learning to rank methods to automatically accomplish this task.

Others Applications

Others text-mining applications on biomedical literature are emerging, providing novel and useful services. One of this novel application is the expert finding whose goal is to select the right expert of a certain topic. Some solutions to find biomedical experts already exists, such as GoPubMed (Doms and Schroeder, 2005), eTBLAST (Errami *et al.*, 2007) and Jane (Schuemie and Kors, 2008), whose most important weakness is that they are not natively conceived to support experts finding, given as input the topic query. In fact, Jane and eTBLAST take a title or abstract as input, and output the most similar publications, relevant journals, and experts. More recently, BMExpert (Wang *et al.*, 2015) mines MEDLINE documents by considering the relevance of documents with respect to the input topic, and the associations between documents and experts, because, for the biomedical publication, the order of the authors is important.

Another application is the question-answering (Q&A), which returns an answer for a given question. Q&A initially found the best answer by performing data- and text-mining from the database, while more recently uses also unstructured resources. Q&A requires complex natural language queries understanding. It typically involves subtasks, such as, determining the type of questions (e.g., YES/NO, WH, choice, etc.) and, consequently, the type of the expected answers (e.g., time?, location?, person?), or finding documents containing the answer. Moreover, to address biomedical question-answering there is a need to use domain specific linguistic resources, such as Unified Medical Language System (UMLS), a controlled vocabulary unifying over 100 dictionaries, National Center for Biomedical Ontology (NCBO), a collaborative ontology repository, and so forth. For instance, UMLS is used in (Takahashi *et al.*, 2004) a question answering system using semantic information of the terms that were selected from the returned documents, to rank candidate answers and to determine their similarity degree. Let us note that the degree of difficulty varies by varying the question type. In the biomedical domain, the definition of questions has been extensively studied, see Lin

and Demner-Fushman (2005), Yu and Wei (2006), Yu *et al.* (2007). BioSquash (Shi *et al.*, 2007) is a question answering system based on multi-document summarization system for biological questions. Some other works address more specific biological questions, as in (Lin *et al.*, 2008) that focuses on questions about events in biomolecular scenarios, like the interactions between genes and proteins.

The last trend emphasizes the interest in extracting relations among entities cited for semantically annotating the text by using NLP and text-mining (De Maio *et al.*, 2014). Actually, to improve text-mining applications, based on biomedical literature analysis, there is a need to perform a deeper information extraction. In fact, it is required to extract assertions about protein-protein interactions, diseases and their treatments, nanomaterial and its toxicity because of the growing amount of unstructured findings published so frequently (Hunter *et al.*, 2008; Kilicoglu and Bergler, 2009). In addition, if we consider the impact of novel findings there is also a need to take into account during the text-mining the temporal dimension (De Maio *et al.*, 2016).

Evaluation Techniques and Metrics

In literature many methods to evaluate text mining applications exist. In this section, an overview of the most common is presented.

The most used measures for evaluating information retrieval, document classification, NER, and other applications are:

- *Precision*, is the fraction of all retrieved documents that are labeled *relevant*

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved}) \quad (1)$$

- *Recall*, is the fraction of all *relevant* labeled documents that have been effectively retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = R(\text{retrieved}|\text{relevant}) \quad (2)$$

These measures can be clarified by examining the following so called Confusion or Error Matrix: (Table 5).

Then:

$$P = \frac{tp}{(tp + fp)} \quad R = \frac{tp}{(tp + fn)} \quad (3)$$

- *Balanced F-measure*, attempts to reduce precision and recall to a single measure.

$$F = 2 \frac{(PR)}{(P + R)} \quad (4)$$

- *Accuracy*, is the number of correct answers divided by the total number of answers. In terms of the Confusion Matrix above,

$$\text{Accuracy} = \frac{(tp + tn)}{(tp + fp + fn + tn)} \quad (5)$$

- *Error*, is the proportion of incorrectly identified instances:

$$\text{Error} = 1 - \text{Accuracy} \quad (6)$$

In addition, text summarization systems are usually evaluated by using ROUGE metric for estimating the similarity of the resulting summary with respect to a, so called, *gold summary* at syntactic level by matching the n-grams, or at semantic level (De Maio *et al.*, 2016) by evaluating *concepts* covered by the *generated summary*. There are several variants of ROUGE, the following one attains with the metric that estimates the number of n-grams that are both in the gold and in the generated summary:

$$\text{ROUGE} = \frac{\#(\text{relevant } n - \text{gram retrieved in Generate Summaries})}{\#(\text{relevant } n - \text{gram in Gold Summaries})} \quad (7)$$

Table 5 Confusion Matrix

	<i>Relevant</i>	<i>Not relevant</i>
Retrieved	True positives (tp)	False positives (fp)
Not Retrieved	False negatives (fn)	True negatives (tn)

Case Study: Text Mining for Simplifying Biomedical Literature Browsing

This section illustrates an example of text-mining techniques for enabling a biomedical literature-based application. This application was previously experimented in [De Maio et al. \(2015a\)](#) to extract multi-faceted data model enabling user-friendly browsing of biomedical literature. Then, it was extended to address natural language query processing in [De Maio et al. \(2015b\)](#).

The motivation essentially is to simplify the exploration of biomedical data occurring on the World Wide Web in unstructured biomedical repositories, e.g.: PubMed ([NCBI, 2013](#)) an open access full-text archive of biomedical literature, WikiGenes ([Hoffmann, 2008](#)) a global collaborative knowledge base for the life sciences data, and so on (see Relevant Website for PubMed and WikiGenes).

The proposed framework defines a methodology for categorizing biomedical literature acquired from PubMed by adopting concepts of biomedical ontologies as category names. The overall workflow is composed of the following phases:

- **Knowledge Extraction.** In this phase a common sense ontology is extracted, called *Unsupervised Ontology*, categorizing unstructured biomedical literature in unsupervised. It executes biomedical content wikification, that is the practice of representing a content with a set of Wikipedia entities (e.g., articles) ([Mihalcea and Csomai, 2007](#)), and consequently apply Fuzzy Formal Concept Analysis (FFCA) to extract ontology concepts that may be meaningful for the biomedical domain. A successive step filters out concepts that are meaningless by evaluating a matching degree with respect to a set of entities included in some *Biomedical Ontologies*.
- **Biomedical Ontologies Matching.** This phase implements a matching strategy to find relations between *Unsupervised Ontology* and existing *Biomedical Ontologies* concepts in order to support advanced queries and visualization procedures. In fact, concept matching methods let one organize and visualize collected biomedical resources, providing thus a navigation model based on ad-hoc faceted browsing visualizations. In particular, a friendly multi-facets visualization engine will be defined enabling integrated access to the biomedical data sources driven by existing biomedical ontologies.
- **Natural Language Query Processing.** This phase processes natural language queries by executing an ad-hoc matching algorithm that fires the most relevant concepts of *Biomedical Ontologies*. Then, the biomedical resources categorized in the fired concepts will be retrieved and ranked as the results of the incoming user's query. Analogously to biomedical content, this step performs the wikification of the input query, in order to capture the meaning in the user's request. Then, the match between the wikified query and the concepts of the *Biomedical Ontologies* is evaluated to carry out a ranked list of results.

The screenshot displays a web application titled "MULTI-FACETS VISUALIZATION" running on a local host. The interface is divided into several regions:

- Region A (Left Panel):** Contains three faceted ontologies:
 - Ontology of Genes and Genomes:** Includes filters for "protein-coding gene of Homo sapiens" (selected), "THRB (13)", "THRA (31)", "SRSF1 (6)", "SCN5A (13)", and "NEDD4L (31)".
 - PRotein Ontology:** Includes filters for "cellular_component" (selected), "Endosome membrane (3)", "plasma membrane (41)", "organism", "Brucella (11)", "Avena sativa (1)", "Gallus gallus (34)", and "Homo sapiens (7)".
 - Gene Ontology:** Includes filters for "cell component" (selected), "cell (74)", "cell junction (3)", "oxidoreductase complex (23)", "membrane", "Apical plasma membrane (91)", "coated membrane (51)", and "side of membrane (14)".
- Region B (Top Right):** Displays "Current Search Results" and a "Concept Facet" showing the selected path: "gene of Homo Sapiens > protein-coding gene of Homo sapiens" (selected).
- Region C (Bottom Right):** Shows "Results 6 of 31" with a list of search results, including:
 - "[Mutation process in the protein-coding genes of human mitochondrial genome in context of evolution of the genus]."
 - "[No authors listed] Mol Biol (Mosk). 2013 Nov-Dec;47(6):927-33. Russian. PMID: 25509854 [PubMed - in process] Related citations PubMed"
 - "Conserved syntenic clusters of protein coding genes are missing in birds. Lovell PV, Wirthlin M, Wilhelm L, Minx P, Lazar NH, Carbone L, Warren WC, Mello CV. Genome Biol. 2014 Dec 18;15(12):565. [Epub ahead of print] PMID: 25518852 [PubMed - as supplied by publisher] Related citations PubMed"
 - "ECM1 – extracellular matrix protein 1 Gene: ... mutations in the extracellular matrix protein 1 gene(ECM1). Although the precise function of ECM1... the radiation hybrid mapping of human ECM1 to 1q21, and the gene structure and coding sequence of human ECM1...Disease relevance of Related citations Wikigenes"

Fig. 4 Ontology Driven Faceted Individuals Browser.

We adopted some *Biomedical Ontologies* available in BioPortal, such as *Ontology of Genes and Genomes* – OGG, a formal ontology of genes and genomes of biological organisms; *PROtein Ontology* – PRO, an ontological representation of protein-related entities; *Gene Ontology* – GO (see Relevant Website for BioPortal, OGG, PRO and GO).

The resulting multi-faceted data model, simplifies the browsing by providing domain specific categories for accessing the literature. In general, the facet-based navigation (Yee *et al.*, 2003) is an efficient technique for navigating a collection of information based on incremental filtering. The goal of this data exploration technique, is to restrict the search space to a set of relevant resources. Unlike a simple traditional hierarchical category scheme, the users have the ability to drill down to concepts based on multiple dimensions. These dimensions are called facets and represent peculiar features of the information units. Each facet has multiple restriction filters and the user selects a value to constrain relevant items in the search space. Specifically, the multi-facets visualization engine proposed here provides a prototype directory web site which associates data (i.e., a biomedical resource) to biomedical ontologies' categories. The user interface is shown in Fig. 4: *Region A* shows the facets derived by different subtrees of concepts available in the selected biomedical ontologies; *Region B* illustrates the selected constraints specified by the user to filter the search space shown in *Region C*, and *Region C* shows the ranked list of the retrieved results according to the selection of specific facets values. For experimental results and more details about the case study illustrated in this section one can refer to De Maio *et al.* (2015a,b).

Conclusions and Future Challenges

The article provides an overview of the text-mining in Bioinformatics introducing fundamentals methods, literature-based applications, and existing solutions. In addition, a case study implemented for simplifying the biomedical literature browsing has been described.

The state of the art emphasizes that last trend of text-mining is related to the extraction of relation among entities considering also temporal aspects and impact of novel findings on the existing knowledge base.

See also: Biomedical Text Mining. Data-Information-Concept Continuum From a Text Mining Perspective. Gene Prioritization Tools. Homologous Protein Detection. Natural Language Processing Approaches in Bioinformatics. Protein Functional Annotation. Text Mining Applications. Text Mining for Bioinformatics Using Biomedical Literature. Text Mining Resources for Bioinformatics

References

- Al-Hegami, A.S., Othman, A.M.F., Bagash, F.T., 2017. A biomedical named entity recognition using machine learning classifiers and rich feature set. *International Journal of Computer Science and Network Security (IJCSNS)* 17 (1), 170.
- Ananiadou, S., Pyysalo, S., Tsujii, J., Kell, D.B., 2010. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology* 28 (7), 381–390.
- Aronson, A.R., Lang, F.-M., 2010. An overview of metamap: Historical perspective and recent advances. *Journal of the American Medical Informatics Association* 17 (3), 229–236.
- Brill, E., 1992. A simple rule-based part of speech tagger. In: *Proceedings of the Workshop on Speech and Natural Language*, Association for Computational Linguistics, pp. 112–116.
- Chebil, W., Soualmia, L.F., Darmoni S.J., 2013. Bodi: A new approach to improve biomedical documents indexing. In: *International Conference on Database and Expert Systems Applications*, Springer, pp. 78–87.
- Consortium, E.P., *et al.*, 2014. Europe pmc: A full-text literature database for the life sciences and platform for innovation. *Nucleic Acids Research*. gku1061.
- De Maio, C., Fenza, G., Gallo, M., Loia, V., Senatore, S., 2014. Formal and relational concept analysis for fuzzy-based automatic semantic annotation. *Applied Intelligence* 40 (1), 154–177.
- De Maio, C., Fenza, G., Loia, V., Parente, M., 2015a. Biomedical data integration and ontology-driven multi-facets visualization. In: *International Joint Conference on Neural Networks (IJCNN)*, 2015, IEEE, pp. 1–8.
- De Maio, C., Fenza, G., Loia, V., Parente, M., 2015b. Natural language query processing framework for biomedical literature. In: *2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*, Atlantis Press.
- De Maio, C., Fenza, G., Loia, V., Parente, M., 2016. Time aware knowledge extraction for microblog summarization on twitter. *Information Fusion* 28, 60–74.
- Doms, A., Schroeder, M., 2005. Gopubmed: Exploring pubmed with the gene ontology. *Nucleic Acids Research* 33 (suppl 2), W783–W786.
- Dong, X., Qian, L., Guan, Y., *et al.*, 2016. A multiclass classification method based on deep learning for named entity recognition in electronic medical records. In: *Scientific Data Summit (NYSDS)*, 2016 New York, IEEE, pp. 1–10.
- Errami, M., Wren, J.D., Hicks, J.M., Garner, H.R., 2007. etblast: A web server to identify expert reviewers, appropriate journals and similar publications. *Nucleic Acids Research* 35 (suppl 2), W12–W15.
- Fanelli, D., 2009. How many scientists fabricate and falsify research? A systematic review and meta-analysis of survey data. *PLoS One* 4 (5), e5738.
- Greene, B.B., Rubin, G.M., 1971. Automated grammatical tagging of english.
- Han, X., Kwok, C.K., Kim, J.-J., 2016. Clustering based active learning for biomedical named entity recognition. In: *International Joint Conference on Neural Networks (IJCNN)*, 2016, IEEE, pp. 1253–1260.
- Hoffmann, R. 2008. A wiki for the life sciences where authorship matters (English) *Nature Genetics* 40 9 1047–1051 ([b1] (analytic)).
- Huang, C.-C., Lu, Z., 2016. Community challenges in biomedical text mining over 10 years: Success, failure and the future. *Briefings in Bioinformatics* 17 (1), 132–144.
- Hunter, L., Lu, Z., Firby, J., *et al.*, 2008. Opendmap: An open source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression. *BMC Bioinformatics* 9 (1), 78.
- Jiang, X., Ringwald, M., Blake, J.A., Shatkay, H., Effective biomedical document classification for identifying publications relevant to the mouse gene expression database (gxd).

- Kilicoglu, H., Bergler, S., 2009. Syntactic dependency based heuristics for biological event extraction. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, Association for Computational Linguistics, pp. 119–127.
- Kim, S., Kwon, D., Shin, S.-Y., Wilbur, W.J., 2012. Pie the search: Searching pubmed literature for protein interaction information. *Bioinformatics* 28 (4), gku1597. (arXiv:oup/backfile/content_public/journal/bioinformatics/28/4/10.1093/bioinformatics/btr702/btr702.pdf, <https://doi.org/10.1093/bioinformatics/btr702>).
- Lai, P.-T., Lo, Y.-Y., Huang, M.-S., Hsiao, Y.-C., Tsai, R.T.-H., 2016. Belsmile: A biomedical semantic role labeling approach for extracting biological expression language from text. *Database* 2016, baw064.
- Lakiotaki, K., Hliaoutakis, A., Koutsos, S., Petrakis, E.G., 2013. Towards personalized medical document classification by leveraging umls semantic network. In: *International Conference on Health Information Science*, Springer, pp. 93104.
- Leaman, R., Lu, Z., 2016. Taggerone: Joint named entity recognition and normalization with semi-Markov models. *Bioinformatics* 32 (18), 2839–2846.
- Li, C., Jimeno-Yepes, A., Arregui, M., Kirsch, H., Rebholz-Schuhmann, D., 2013. Pcorralinteractive mining of protein interactions from medline. *Database* 2013, bat030. (arXiv:oup/backfile/content_public/journal/database/2013/10.1093/database/bat030/2/bat030.pdf, <https://doi.org/10.1093/database/bat030>).
- Lin, J., Demner-Fushman, D., 2005. Automatically evaluating answers to definition questions. In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 931–938.
- Lin, R.T., Liang-Te Chiu, J., Dai, H.-J., *et al.*, 2008. Biological question answering with syntactic and semantic feature matching and an improved mean reciprocal ranking measurement. In: *IEEE International Conference on Information Reuse and Integration*, 2008. IRI 2008., IEEE, pp. 184–189.
- Liu, Y., Liang, Y., Wishart, D., 2015. Polysearch2: A significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Research* 43 (W1), W535. (arXiv:oup/backfile/content_public/journal/nar/43/w1/10.1093_nar_gkv383/2/gkv383.pdf, <https://doi.org/10.1093/nar/gkv383>).
- Lossio-Ventura, J.A., Hogan, W., Modave, F., *et al.*, 2016. Towards an obesity-cancer knowledge base: Biomedical entity identification and relation detection. In: *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, pp. 1081–1088.
- Lou, Y., Zhang, Y., Qian, T., *et al.*, A transition-based joint model for disease named entity recognition and normalization, *Bioinformatics* (Oxford, England).
- Malhotra, A., Gündel, M., Rajput, A.M., *et al.*, 2015. Knowledge retrieval from pubmed abstracts and electronic medical records with the multiple sclerosis ontology. *PIOS One* 10 (2), e0116718.
- Màrquez, L., Rodríguez, H., 1998. Part-of-speech tagging using decision trees, *Machine Learning: ECML-98* 25–36.
- McEntyre, J., Lipman, D., 2001. Pubmed: Bridging the information gap. *Canadian Medical Association Journal* 164 (9), 1317–1319.
- Mihalcea, R., Csoma, I., 2007. Wikify!: Linking documents to encyclopedic knowledge. In: *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, ACM, pp. 233–242.
- Mrabet, Y., Kilicoglu, H., Roberts, K., Demner-Fushman, D., 2016. Combining open-domain and biomedical knowledge for topic recognition in consumer health questions. In: *AMIA Annual Symposium Proceedings*, vol. 2016, American Medical Informatics Association, p. 914.
- NCBI, 2013. The NCBI Handbook [Internet], second ed. Bethesda, MD: National Center for Biotechnology Information. Available at: <https://www.ncbi.nlm.nih.gov/books/NBK143764/>.
- Raja, K., Subramani, S., Natarajan, J., 2013. Ppinterndera mining tool for extracting causal relations on human proteins from literature. *Database* 2013, bas052.
- Ramakrishnan, C., Patnia, A., Hovy, E., Burns, G.A., 2012. Layout-aware text extraction from full-text pdf of scientific articles. *Source Code for Biology and Medicine* 7 (1), 7.
- Roberts, K., Simpson, M., Demner-Fushman, D., Voorhees, E., Hersh, W., 2016. State-of-the-art in biomedical literature retrieval for clinical cases: A survey of the trec 2014 cds track. *Information Retrieval Journal* 19 (1–2), 113–148.
- Schuemie, M.J., Kors, J.A., 2008. Jane: Suggesting journals, finding experts. *Bioinformatics* 24 (5), 727–728.
- Schulze, F., Neves, M., 2016. Entity-supported summarization of biomedical abstracts. *BioTm 2016*, 40.
- Shang, Y., Hao, H., Wu, J., Lin, H., 2014. Learning to rank-based gene summary extraction. *BMC Bioinformatics* 15 (12), S10.
- Shi, Z., Melli, G., Wang, Y., *et al.*, 2007. Question answering summarization of multiple biomedical documents. *Advances in Artificial Intelligence*. Springer, pp. 284–295.
- Smith, L., Rindflesch, T., Wilbur, W.J., *et al.*, 2004. Medpost: A part-of-speech tagger for biomedical text. *Bioinformatics* 20 (14), 2320–2321.
- Takahashi, K., Koike, A., Takagi T., 2004. Question answering system in biomedical domain, In: *Proceedings of the 15th International Conference on Genome Informatics*, pp. 161–162.
- Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., Ananiadou, S., 2011. Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics* 27 (13), i111–i119.
- Vijay, J., Sridhar, R., 2016. A machine learning approach to named entity recognition for the. *Asian Journal of Information Technology* 15 (21), 4309–4317.
- Wang, B., Chen, X., Mamitsuka, H., Zhu, S., 2015. Bmexpert: Mining medline for finding experts in biomedical domains based on language model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12 (6), 1286–1294.
- Wang, J.Z., Zhang, Y., Dong, L., *et al.*, 2014. G-bean: An ontology-graph based web tool for biomedical literature retrieval. *BMC Bioinformatics* 15 (12), S1.
- Wu, Z., Zhu, H., Li, G., *et al.*, 2017. An efficient wikipedia semantic matching approach to text document classification. *Information Sciences* 393, 15–28.
- Yee, K.-P., Swearingen, K., Li, K., Hearst, M., 2003. Faceted metadata for image search and browsing. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, pp. 401–408.
- Yu, H., Lee, M., Kaufman, D., *et al.*, 2007. Development, implementation, and a cognitive evaluation of a definitional question answering system for physicians. *Journal of Biomedical Informatics* 40 (3), 236–251.
- Yu, H., Wei, Y., 2006. The semantics of a definiendum constrains both the lexical semantics and the lexicosyntactic patterns in the definiens. In: *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, Association for Computational Linguistics, pp. 1–8.
- Yu, H., Wei, Z., Sun, L., Zhang, Z., 2016. Biomedical named entity recognition based on multistage three-way decisions. In: *Chinese Conference on Pattern Recognition*, Springer, pp. 513–524.
- Zhou, X., Zhang, X., Hu, X., 2006. Maxmatcher: Biological concept extraction using approximate dictionary lookup, *PRICAI 2006: Trends in artificial intelligence*, pp. 1145–1149.

Relevant Websites

<http://opennlp.apache.org/>
 Apache OpenNLP.
<http://bioportal.bioontology.org/>
 BioPortal.
<http://bioportal.bioontology.org/ontologies/GO>
 BioPortal – Gene Ontology.
<http://bioportal.bioontology.org/ontologies/OGG>
 BioPortal – Ontology of Genes and Genomes.

<http://bioportal.bioontology.org/ontologies/PR>
BioPortal – Protein Ontology.

<http://www.biominigbu.org/ppinterfinder/about.html>
Data Mining and Text Mining Lab.

<http://dragon.ischool.drexel.edu/example/maxmatcher.zip>
Dragon Toolkit.

<http://www.ebi.ac.uk/Rebholz-srv/>
EMBL-EBI.

<https://europepmc.org/>
Europe PMC.

<http://www.nactem.ac.uk/facta/>
FACTA + .

<http://www.gopubmed.com/>
GoPubMed.

<http://www.geneontology.org/>
Gene Ontology Consortium.

<http://alias-i.com/lingpipe/>
LingPipe.

<http://mallet.cs.umass.edu/>
MALLET.

<http://mmtx.nlm.nih.gov/>
MetaMap.

<http://www.nltk.org/>
Natural Language Toolkit.

<https://www.ncbi.nlm.nih.gov/pubmed>
NCBI.

<https://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/IRET/PIE/>
NCBI.

<http://www.ncbi.nlm.nih.gov/pubmed>
NCBI.

<https://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/IRET/PIE/>
NCBI-NIH.

<https://www.nlm.nih.gov/mesh/>
NIH US National Library of Medicine.

<http://polysearch.cs.ualberta.ca/>
PIE.

<http://polysearch.cs.ualberta.ca/>
PolySearch.

<http://www.scaiview.com/>
SCAIVIEW.

<https://www.wikigenes.org/>
wikigenes.

Data-Information-Concept Continuum From a Text Mining Perspective

Danilo Cavaliere and Sabrina Senatore, Università degli Studi di Salerno, Fisciano, Italy

Vincenzo Loia, University of Salerno, Fisciano, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Nowadays, people communicate and share documents, opinions, ideas or anything they think or whatever happens to them in any moment of their lives.

Documents, posts, tweets, tutorials, etc., represent a substantial pool for unstructured data that need to be arranged to discover and retrieve relevant information as necessary.

The Web is indeed, the biggest existing source of information that still needs to be analysed in order to become accessible knowledge (Della Rocca *et al.*, 2017). In the era of big data, information mining is crucial to effectively access to potentially infinite, on-line knowledge, available from data-structured sources like Wikipedia but mainly from unstructured, natural language resources. There is an increasing need for analysing large amounts of texts in order to get the content meaning enclosed in them, not just as sequences of most frequent terms (whose comprehensive interpretation is left to the humans), but as a high-view synthesis of the content that represents conceptualization, feeling, opinions.

To this purpose, a complex data structuring, defined on discovered term relationships is required. The analysis of textual data is very tricky and needs an accurate processing to identify the relevant terms, their relationships, contextual information (i.e., surrounding text that can univocally characterize the meaning of such terms) to clearly recognize the conceptualizations. Many Concept Mining methodologies infer new concepts directly from the text. They generally employ syntactic and semantic relations among the most related terms or phrases to build contextual views, which are useful to extract and discriminate the concept. The complete process yields complex patterns in data to extract high level knowledge such as concepts, document topics, entities, feelings and behaviours.

This article presents an overview of the Concept Mining literature, discussing the main methodologies and the granularity level of the generated knowledge, i.e., the complexity in structuring the information adopted to define and express the concepts. Knowledge can be very basic involving simple information (individual terms), but it can also be more refined involving more articulated information (term-relationship structures). Generally, a lower-level knowledge is composed of elementary data, such as words and terms extracted from documents. Conversely, a higher-level knowledge, expressing more articulated information, is generally represented by more structured data.

The remainder of this article is structured as follows. Section A Layered Multi-Granule Knowledge Model introduces a new viewpoint of the knowledge representation, described by different granulation levels, according to the kind of knowledge modelled by Concept Mining approaches in the literature. Section Research Landscape presents an overview of the research landscape, focussing on the main research areas involved. Then, Section Approaches is devoted to introduce the main approaches in the Text Mining area, with a specific focus on concept-based approaches, classified according to their main information granule features: data, information, concept. Finally, based on the layer-based knowledge schema introduced in Section A Layered Multi-Granule Knowledge Model, Section A Schematic Description of a Multi-Layer Knowledge Representation introduces a granulation-based knowledge overview of the principal frameworks and tools, depicted through some knowledge structuring features, that are salient in the Concept Mining domain. Conclusion closes this article.

A Layered Multi-Granule Knowledge Model

The knowledge extraction is the complex activity of identifying valid and understandable patterns in data. These patterns are often related to the Natural Language Processing tools: the text content is parsed to identify topics that could be described by single terms, enhanced terms matching by adding phrases, complex key-phrases. Extracting the relations between terms, or verbs and their arguments in a sentence has the potential for identifying the context of terms within a sentence. The contextual information can be very informative for capturing the actual meaning of some terms, and the sense relations involved in the case of polysemy. The information about “who is doing what to whom” reveals the role of each term in a phrase and the higher level meaning of the phrase.

Simple terms or complex expressions represent different granularity levels of the knowledge that can vary depending on the formal methods used, the final conceptualizations and the intend meaning behind the sentences, whose interpretation often escapes automated machine-oriented approaches.

Fig. 1 shows our representation of the knowledge continuum, in an incremental layer-based transformation. The knowledge schema is composed of more granularity levels, starting from “atomic” entities, i.e., single words, to reach a high level representation of knowledge as ensemble of conceptualization and semantic correlations.

The lowest layer represents the primitive knowledge related to single words and terms in a document. Words collection per document is considered the basic data, or more simply, the *data*. The *Data* layer consists of raw data, which are generally composed by single words extracted from textual documents. These documents can be unstructured, i.e., the content is plain text, or structured, such as web resources with markup annotations (written in standard languages such as XML, HTML, CSS, etc.).

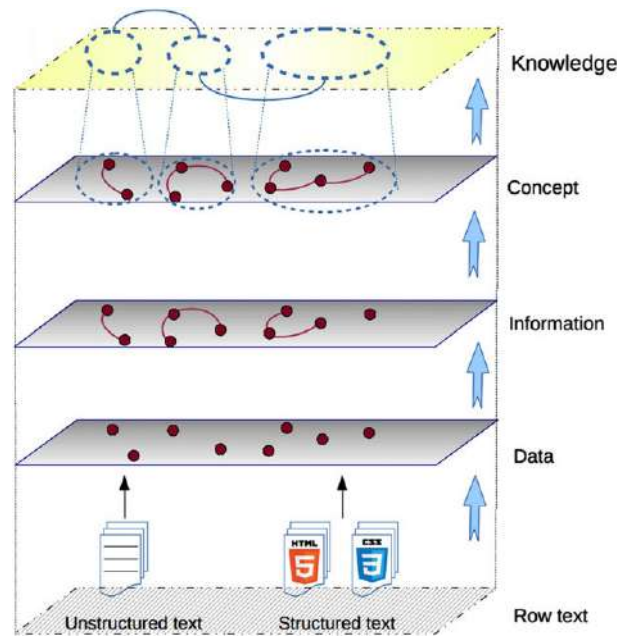


Fig. 1 Layered representation of the knowledge: from words to concepts.

In other words, the unstructured texts contain just words, while the structured ones contain meta-information like HTML tags, which can be used to extract more information about the concepts expressed by the text and the document structure. Let us consider the following document extract:

Canadian pop star Michael Buble married Argentine TV actress Luisana Lopilato in a civil ceremony on Thursday. The Grammy-winning singer of "Crazy Love" and his Argentine sweetheart posed for a mob of fans after tying the knot at a civil registry in downtown Buenos Aires.

Approaches modelling the *Data* layer directly work on the single words such as: "star", "actress", "TV", etc. and often only nouns (and adjectives in some cases) are processed: since these approaches work on the term frequency, proper nouns can be discarded if no named entity task is foreseen in the process. Usually, flat term ensembles are produced by this layer.

When data are furthermore processed or structured, the informative granule increases. The *Information* layer consists of more data structuring which considers linguistic and grammar relationships among atomic terms. More articulated sequences of words, often called keyphrases are extracted by the text analysis. Stemming, lemmatization, part of speech tagging are the basic NLP tasks involved; they allow the identification of terms that are linked by relations: sequences of only nouns, combinations of adjectives and nouns, named entities, etc. characterize the *information*.

By considering the previous extract, named entities such as "Michael Buble", "Buenos Aires", and a keyphrase such as "pop star", or syntactic relations between terms or entities, i.e., "civil ceremony" are the candidates to describe the information granule generated by this layer.

The highest layer of knowledge corresponds to a further and articulated structuring of the data which represent a more detailed and high-level knowledge. The informative granules of this layer define complex structure of terms that are supported by external sources, such as lexicons, knowledge bases and ontology-based schemas in order to provide a richer conceptualization, specialized thanks to contextual information and the term relationships. Compositions of simpler linguistic expressions yield complete and complex descriptions of concepts that, at this stage, are well-defined. The *Concept* Layer produces correlations of terms so rich that clearly identify a conceptualization, often specific in a given domain (according to the content of the processed document collection) and assumes a clear connotation for an authentic interpretation of the textual content. A crucial role is played by external knowledge-based sources, especially if they are semantically annotated, because, they yield additional (often inferred) information to extend, enrich and disambiguate concepts extracted by text. The extracted concepts, connected to each other by term-based relationships, compound a wider semantic network that represents the highest granulation layer.

The *Concept* layer can deduct more articulated concepts as facts from text; recalling the previous extract, the ex-novo concept <<marriage>> coming from the named entities "Michael Buble" and "Luisana Lopilato" could be extracted exploiting an ontological schema, thus enriching the initial more vague concept.

Each knowledge level introduces further relations among more articulated and high level data, which are taken into account. These relations are useful to provide a better contextual insight. To this purpose, let us consider a further example of knowledge stratification, given in Fig. 2, which considers the three words "code", "genetic" and "source". At the *Data* level, these words are considered, according to the proposed conceptualization, as simple three words or singleton terms <<code>>, <<genetic>>, <<source>> (see words beside the bullets at bottom of Fig. 2. Terms or words, at the *Information* level, are further structured according to term

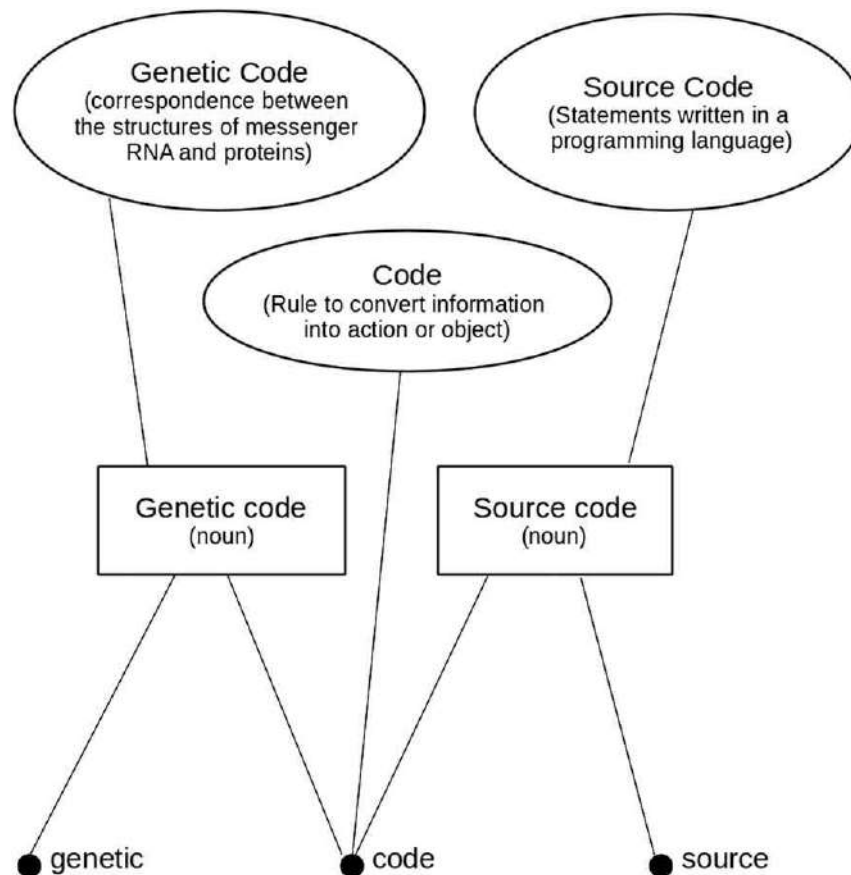


Fig. 2 Knowledge stratification example.

relationships and/or other transformations (e.g. POS tagging)). Thus, terms appearing close to each other within a document can be recognized as named entities or linguistic period structures (e.g. nouns, verbs, etc.). In this case if the words “genetic” and “code” are very close and used as phrase subject or object, <<Genetic Code>> will be recognized as proper noun (see texts in rectangles, Fig. 2). Finally, these more structured data can be further structured at *Concept* level. At this level, the data include more contextual information about terms on different documents. So, the word “code”, at this level, could refer to distinct concepts according to relations with other terms in the text. If the term is considered as single word it represents a concept (<<Code>>), which refers to a general rule for converting a piece of information into another object or action. If the term is close to the term “genetic”, together they form a fundamental concept in biology which refers to the set of rules by which information on genetic material (RNA or DNA sequences) is translated into proteins. The term “code” could be related with other terms, such as “source”. In this case, the two words unambiguously refer to a key concept (<<Source Code>>) in Computer Science, it refers to the sequence of statements written in a human-readable programming language (see texts in the ellipses, Fig. 2).

In a nutshell, a strongly connected structure may be figured from the layer-based knowledge model shown in Fig. 1: all the informative granules are linked, through all the granulation levels, in order to generate a large knowledge base that comprehensively describes the entire text documents domain.

Related literature provided overviews and perspectives on Text and Concept Mining, particularly focusing on specific methodologies (e.g., classification (Brindha *et al.*, 2016)), semantics (Ilakiya *et al.*, 2012; Kathuria *et al.*, 2016; Saiyad *et al.*, 2016), specific applicative Text Mining tasks (e.g., document clustering (Saiyad *et al.*, 2016)), opinion mining (Khan *et al.*, 2009; Ambika and Rajan, 2016), information retrieval (Kathuria *et al.*, 2016). To the best of our knowledge there is no study evaluating Concept Mining approaches by analysing the knowledge they model. In fact, our main goal is to provide an overview of Concept Mining approaches analysed by a knowledge modelling perspective.

Research Landscape

Concept Mining Background

Concept Mining represents a subfield of Text Mining, aimed at extracting concepts from text. Concept Mining approaches are largely used in many fields like Information Retrieval (IR) (Joel and Fagan, 1989) and Natural Language Processing (NLP)

(Cambria and White, 2014). The main applications include detecting indexing similar documents in large corpora, as well as clustering document by topic.

Most of the common techniques in this area are mainly based on the statistical analysis of term frequency, to capture the relevance of a term within a document (Yang *et al.*, 2014). More accurate methods need to capture the semantics beyond the terms.

Traditionally concept extraction methods employed thesauri like WordNet [WordNet], which transform words extracted from document in concepts. The main issue related to thesauri is that the word mapping to concepts is often ambiguous. Ambiguous terms can be generally related to more than one concept, only the human abilities allow contextualizing terms and find the right concept to which terms belong. Since thesauri do not describe the context along with the terms, further techniques have been introduced to face the word sense disambiguation; some of them perform linguistic analysis of text, based on term frequency similarity measures, some others employ knowledge-based models to generate a context useful to evaluate a semantic similarity between concepts.

Since documents are often described as a sequence of terms, a widely used data representation adopted by many methods is the vector space model (VSM) (Aas and Eikvil, 1999; Salton *et al.*, 1975; Salton and McGill, 1986). The VSM model represents each document as a feature vector of terms (words and/or phrases) present in the document. The vector usually contains weights of the document terms, defined mainly on the frequency with which the term appears in the document. Other techniques extend the VSM representation with context-related information for each term, transforming the VSM vector in a global text context vector. This way, a global context about a term is built by merging its local contexts, which are derived from each document where the term appears in Kalogeratos and Likas (2012).

Document similarity is evaluated by considering the similarity between their corresponding feature vectors. Well known similarity measures are euclidean, cosine, Jaccard, etc. Other techniques use document attribute information to extract inter-document information to calculate their similarity (Tombros and van Rijsbergen, 2004).

In Text Mining domain, the term importance in a document is based on the frequency with which the term appears in the document. But a high frequency does not mean that a term contributes more to the meaning of a sentence. There are some words with a low frequency which provide key concepts to the sentence. The importance of the term is also evident in summarization activities (Liu *et al.*, 2009). The sentence meaning depends strongly by verbs and their arguments. Verb analysis allows finding out who is doing something, or acting toward something or someone else, clarifying the role of each term in the sentence.

Whether extracting sophisticated information or simple ones, these techniques constitute the underlying methodological background of the Concept Mining research area and most approaches are modelled and developed on the basis of them.

Beyond the Knowledge: Connections and Relations Among the Main Research Domains

As the amount of available web resources continues to increase, the need of knowledge structuring is becoming predominant to enhance many activities, such as document categorization, indexing, web search. Structuring the knowledge means eliciting a structure informative (conceptualization), which connects the atomic elements (often linguistic terms that are relevant within a given collection) through reciprocal relations. Text Mining techniques are the methodological underpinnings on which the semantic infrastructure can be built to translate natural language text into concepts, and then into knowledge.

There are a lot of research domains that strictly affect and conversely are affected by the Text Mining issues. Our review inspects three different even though interrelated research streams, which are mostly involved in Text Mining tasks, to improve the access to unstructured data and alleviate the knowledge discovering. Fig. 3 shows the involved macro-areas: Natural Language Processing (NLP), Semantic Web (SW) and Artificial Intelligence, and Information Retrieval (IR) and their more specific sub-fields. Next sections will provide a brief overview of each research domain introducing the main characterizing approaches.

Natural language processing

Natural Language Processing (NLP) involves basic tasks in Text Mining activities, especially if they are targeted at concept extraction. One of the research goal in NLP is to generate computational models that simulate human linguistic abilities (reading, writing, listening and speaking). NLP aspects distinguish in low-level activities, such as tokenization and stemming of terms, shallow syntactic parsers to extract meaningful terms and term frequency index calculation; and more articulated activities, such as the utilization of a lexical resource (thesaurus) to support the word sense disambiguation (Navigli, 2009) or building ontology-based knowledge bases for topic extraction.

Modern approaches to NLP involve supervised and unsupervised methods from machine learning area, such as clustering, support vector machine (SVM) and undirected graphical models or decision trees.

The ultimate goal is to guarantee clear understanding of the word meaning, word role (i.e., sense) in the sentences, in order to correctly interpret the embedded semantics, associate the contextual information and finally elicit concept-targeted term relationship.

NLP tasks also contribute to build semantic networks (also known as conceptual graphs) that reveal correlations between terms (Kozareva and Hovy, 2010), and the text summarization (Erkan and Radev, 2004). Term relationships are inspected to support NLP applications, as well as to define domain models: graph models build semantic knowledge extracting term relationships, by analysing grammatical relations such as co-occurrence of noun (Widdows *et al.*, 2002). Standard NLP systems employ parsing methods in order to build and analyze the sentence structure. Traditional parsing methods focus on finding tree-based structures, capable of representing grammatical rules, in order to represent the syntax of phrases and clauses of a set of words in a document.

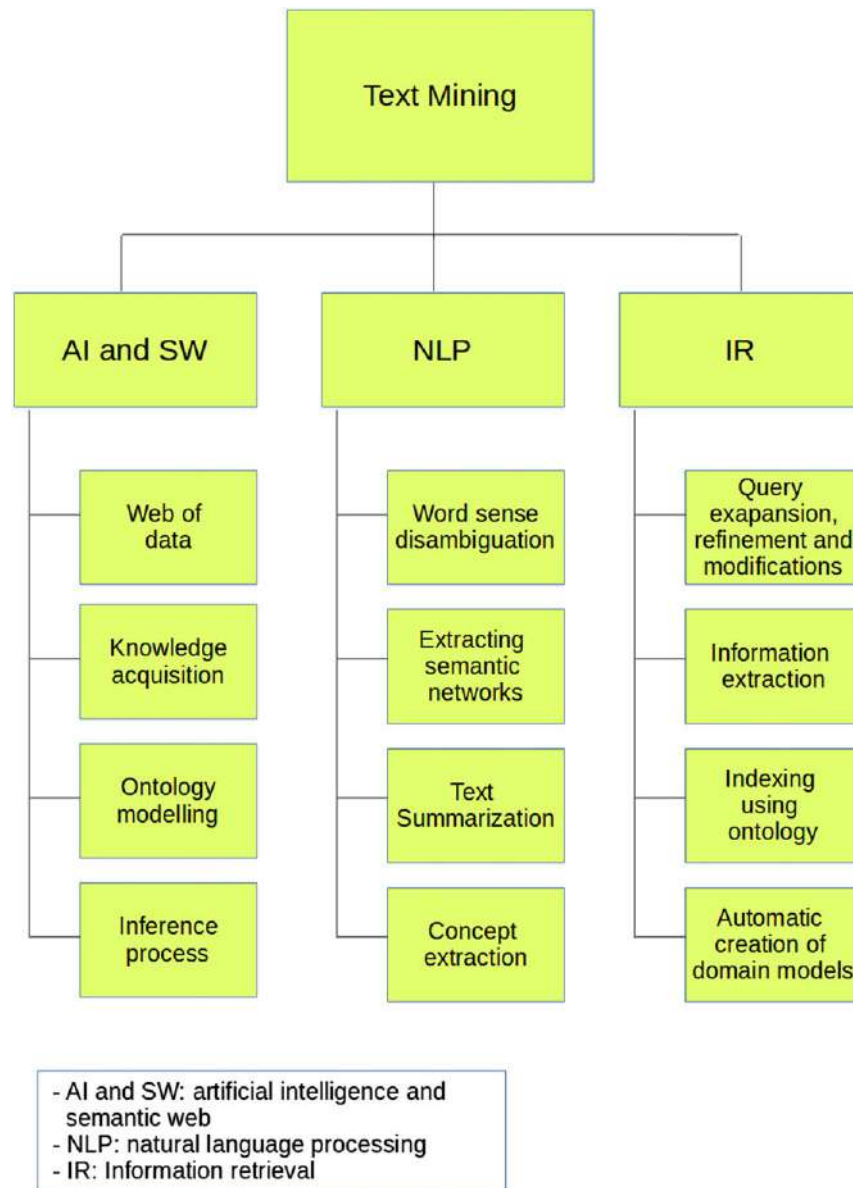


Fig. 3 Text Mining and the main correlated fields.

Therefore, words are tagged and related according to their role in a sentence (e.g., noun, verb, adjective, etc.) even though this activity can generate more and ambiguous syntactical schemas of the same sentence.

In order to reduce ambiguities, a traditional solution produces selectional restrictions, which are hand-coded syntactic constraints and preference rules on words, generally defined by human experts. These constraints restrict the general usage of the word in a sentence, identifying the correct sense. As an example of selectional restriction on a word, let us consider the word “play”, restrictions may be defined according to the terms the verb “play” appears with (e.g., “play guitar”, “play a role”), or on its meaning in the sentence as noun (“It’s your play”). The problem with selectional restrictions is that they are time-consuming, when ad-hoc defined by humans. Moreover, selectional restrictions are often not exhaustive and so they miss a lot of metaphorical and context-specific meanings. To address this issue, the current solutions use statistical or knowledge-based methods to learn automatically syntactical and structural preferences directly from text corpora. These methods produce a term relationship-based context on terms and sentences exploring the surrounding information and producing a better disambiguation. The rationale behind these methods is that the relationship between words, whose aim is to find which words group each other, can add information useful to learn semantic relationships among terms, and discover the word meaning in a sentence.

Besides the word sense disambiguation, another important NLP activity is the named-entity recognition (NER). It is traditionally based on linguistic grammar techniques as well as statistical models (Finkel *et al.*, 2005), aimed at discovering named entities in text.

With the advent of Semantic Web technologies, some projects enhanced named entity recognition with semantics; for instance, FOX [FOX] is an open-source framework that implements RESTful web services for providing users with disambiguated and linked named entities coded in RDF-based serialization formats.

Gate [Gate] is a text processing framework which combines data-driven (words that describe concepts) and knowledge-driven (relations that link concepts) approaches in order to find a link with Semantic Web approaches.

SHELDON (Recupero *et al.*, 2015) represents indeed a clear example of a NLP and SW hybridization tool, by implementing several machine reading tasks to extract meta-data from the text. It exploits a tool for automatically producing coded ontologies and linked data from text, adopting Ontology Design Patterns and Linked Data principles, relation extraction, frame detection, automatic typing of entities and the automatic labeling functions (Presutti *et al.*, 2012).

Semantic web

The proliferation of textual information makes the extraction and collection of relevant information a tricky task. Although search engines are recently enhanced with Artificial Intelligence techniques, the vagueness of natural language is still an open problem. At present, the ontology has proven itself to be an effective technology for representing as a form of concepts, the web information and then sharing common conceptualizations that are referenced as knowledge.

An ontology gathers concepts from the real world by means of unambiguous and concise coding. At the same time, it should capture the terminological knowledge that sometimes embeds imprecise information, should support the management of semantic data and the intrinsic ambiguity in their model theoretic representation, provide enhanced data processing and reasoning, and then supply a suitable conceptualization that bridges the gap between flexible human understanding and hard machine-processing (De Maio *et al.*, 2014). Simple ontology schemas, called taxonomies reflect vocabulary properties (such as term definitions, constraints and relationships) are often used as semantic models representing hierarchical classification of concepts. A taxonomy describes relations between related concepts as super-sub category or subsumption relationship. This schema enables to represent articulated concepts as subsets of more simple concepts, and create a layered structure based on concept complexity useful for concept analysis. In most cases, taxonomies are represented by hierarchical tree structure of classifications for a given set of objects. In order to represent concepts, an ontology can be seen as a semantic network composed of nodes and edges to link nodes, where the nodes represent the concepts and the edges represent the relationships among concepts.

Abstract concepts are defined as ontology classes, while a real concrete example of an abstract concept is represented by class instance (or individuals). The relations among two different concepts are characterized by ontological properties. Ontological axioms are represented in the form of triples subject-predicate-object, in order to represent that a resource (subject) has a property (predicate) which assumes as value another resource or a literal (object). The model-theoretic semantics behind the Semantic Web are based on Description Logics (DLs), in order to model assertions and perform reasoning on the ontological schema, which produces new axioms and increase the knowledge about the concepts.

Coding (meta-) data and relations between them into an ontology, starting from unstructured text is a necessary step towards the knowledge modelling (Navigli and Velardi, 2008). Ontologies and ontology-based applications (Navigli *et al.*, 2004) achieve language processing for extracting keywords inherent to domain concepts from natural language documents.

Semantic Web technologies yield knowledge in the form of concepts and relations among them; they translate the vagueness of natural language (embedded in the linguistic terms) by identifying conceptual entities in the resource content. Intelligent AI computing proactively supports these activities, modelling this ambiguity by more suitable methods and techniques that natively reflect the uncertainty and the reasoning of the human thought process (De Maio *et al.*, 2014).

The nature of the ontology modelling moves towards a shared conceptualization and a consequent reuse of the same data, reinforcing the request that data about concepts and their relationships must be specified explicitly and data needs a robust formalization. Due to the stringent reliance of applications on well-designed data structure, semi-automatic tools like OntoLearn [OntoLearn], AlchemyAPI [AlchemyAPI], Karlsruhe Ontology (KAON) framework [KAON], Open Calais [Open-Calais] and Semantria [Semantria] are widely developed, to automatically extract entities, keywords and concepts from unstructured texts.

Systems and applications as KnowItAll (Fader *et al.*, 2011), DBpedia (Auer *et al.*, 2007), Freebase (Bollacker *et al.*, 2008) also provide publicly semantically annotated knowledge resources; some others such as ConceptNet (Speer and Havasi, 2012), Yago (Suchanek *et al.*, 2007) aim at conceptually capturing common sense knowledge; sometimes, due to the quality bottleneck, projects like Cyc, OpenCyc (Matuszek *et al.*, 2006) and WordNet are often built on manually compiled knowledge collection.

Information retrieval

The query-based paradigm has a long and predominant tradition in Information Retrieval, whose main focus is the study to support users in retrieving information, that match their description. The user submits the query (that represents the information need) to the IR system, generally through a user interface, composed of a text entry box by which the user can type his/her requests in string format. The information item collection is a corpus of documents. Generally, this information depends by the selected engine: if the IR system deals with a web engine, the corpus may be provided by a web crawler. Otherwise in desktop search applications, the document collection is generally gathered from a hard drive.

A typical IR workflow is composed by some specific tasks, whose final output is a ranked list of information sources. The main tasks are as follows:

1. *Document indexing*: is a typical structure adopted by desktop and web search engine in order to speed up the search over large corpora. The structure is defined on an indexing of the textual contents.
2. *Query formulation*: once the document indexing is completed, user search query is also processed. Firstly, the query is parsed and then some transformations (extensions, refinements) are applied to the data in order to get a system representation of the user need.
3. *Query-Documents Matching*: the query is matched with terms composing the document representation, to find relevant documents that meet the user needs. The matching method is built according to a defined retrieval model.
4. *Result Ranking*: a ranked list of relevant documents is returned to the user. The ordering of this list is guaranteed by a ranking algorithm, which assigns a matching score to each document, from the most to the less relevant. The objective is to return those documents that meet the user need.

The effective understanding of user query needs the identification of relevant query terms and their term relationships. The importance of term in a document or in the whole corpus is generally estimated by a term weighting factor, often, expressed by frequency-based measures. The most used measures are indeed the term frequency (tf) and the term frequency/inverse document frequency (tf/idf); other measures involve probability and information theory, i.e., the mutual information which measures the mutual dependence between the two variables (terms).

A term relationship evaluation may involve the importance of the term (e.g., term weighting factor), as well as syntactical and semantic relations. Among them, grammatical relationships, such as synonym, hyponym, etc., have gained an increasing interest in IR area, because they can be used to better distinguish words for query expansion, and for calculate relevancy score of a document for a given query (Snow *et al.*, 2006; Giuliano *et al.*, 2007). These techniques are often based on term clustering, targeted at word sense disambiguation (Phillips, 1985; Schütze, 1998).

In the recent years, the IR process is becoming more interactive and dynamic, customized to the user preferences. Generally short-term queries personalization depends on recent searches and feedback provided by user, who pinpoints the document he/she finds most interesting. Therefore, feedbacks and recent activities are used to refine the result, generally, performed by a query reformulation. IR systems are increasingly oriented to the user personalization, taking into account the user navigation (visited sites, recurring web search, etc.), habits and preferences (Case, 2012; Golovchinsky *et al.*, 2009; Marchionini and White, 2009); also investigating on knowledge structures to support dynamic user queries (Ben-Yitzhak *et al.*, 2008).

Contextualized collected information assume a crucial role to the user profiling; to this purpose, Semantic Web technologies have been investigated to model human behaviours.

Semantic Web provides semantically enriched standards to provide a meta-data description of web resources. It provides a knowledge infrastructure to represent concepts and relations between concepts, sharable by humans by especially by machines, in a way that is cost-effective and consistent with adopted models and ontology schemas. Although there is a plethora of ontologies disseminated on the Web, they are often too general or specialized, sometimes covering only a portion of the domain of interest. At the same time, developing an ontology from scratch is a time-consuming, labor-intensive task that requires a deep understanding of the domain vocabulary (De Maio *et al.*, 2014). For this reason, IR community showed interest in automatically generated domain models from texts, which guarantee language independence. Some studies (Lau *et al.*, 2008; Kruschwitz, 2003; Sanderson and Croft, 1999) have yet propose some new approaches using these techniques for query expansion, suggestions, filtering information, etc.

Data-driven approaches as well as concept-driven approaches are mostly proposed in the IR approaches, even though several studies achieve, which build ontological model to capture domain knowledge for query expansion, some others extract semantic relationships from text to modify query (Clark *et al.*, 2012).

Approaches

According to the knowledge stratification model introduced in Section Introduction, Text Mining approaches have been organized in three granularity-based categories: data, information and concept driven, ranging from the lowest granulation level, the data, then to reach the highest one, the concepts whose ensembles compound the whole knowledge.

Data-driven approaches work on low-level data, i.e., raw data that need to be processed to extract atomic terms, verbs or key-phrases from the text. In these approaches linguistic regularities, such as term co-occurrences, term ensembles are automatically identified, generally with no human intervention. Relationships are indeed extracted in automatic way, through statistical or cluster-based methods that find correlation between term couples and term ensemble. These relations are often of different nature, specificity, type, sometimes based on grammatical dependencies, but in general they are flat, without any type of structuring.

Information-driven approaches work on slightly more complex data structures, that can include linguistic expressions composed of more than one term, named entities, key-phrases.

At this granularity level, discovered relationships aim at identifying higher level data, that can be considered embryonic conceptualizations. Sentence analysis techniques are widely inspected, especially part-whole relations and Part-of-speech (POS) relations, that support the knowledge structure. Unlike data-driven models, information-driven approaches are supported by supervised methodologies, sometimes exploiting external sources.

Concept-driven approaches instead, work on more specific types of relationship. Generally, these approaches extract lexical relationships like synonyms, hyponyms, etc. between (complex) terms, in order to extract concepts or topics expressing a conceptualization. Relationship types are a-prior defined, exploiting a thesaurus or dictionary such as WordNet, FrameNet which collect grammar relationships. Supervised learning methods use several levels of syntactic information, with more control over the identified relationship, which generally provide a deeper knowledge structuring about the concepts. Knowledge structures are often hierarchies of concepts at different levels, ranging from a very specificity to a high-level generalization. At the same time, the knowledge may lack of some specific notions related to the raw data (e.g. words), which could badly affect the adaptability of the system to specialized domains.

Some approaches included and classified according to the three granularity-based layers are described in the next sections.

Data-Driven Approaches

Data-driven approaches can be divided in three main categories: unsupervised, semi-supervised and supervised methods. Unsupervised methods do not need human annotation in the training phase even though some difficulties arise in producing annotation of explicit classes and relationships. As a consequence, the unsupervised methods need time-consuming training annotation tasks, even though the resulting annotations can be high quality and very specific data in the reference domain.

Semi-supervised methods reduce the amount of human effort for the annotation: these methods generally use unsupervised learning but need a human intervention to guarantee some minimal control on the data processing. Data-driven approaches usually exploit unsupervised methods, that basically work on raw data, and take the unstructured text (natural language documents, query logs, etc.) as input. In order to identify relevant terms, frequency analysis is usually used in text processing, coupled with statistical techniques that elicit term co-occurrences in the text, targeted at discovering term relationships in order to identify richer terms (Lauren and Doyle, 1961).

Co-occurrence analysis (Phillips, 1985) indeed, supports the building of conceptual networks and syntagmatic lexical networks to build phrasal patterns, useful to extract and cluster document words according to their vocabulary sense definitions. In Schütze (1998) words and their co-occurrence are mapped on a high dimensional space and clustered with the Expectation-Maximization (EM) clustering algorithm, initialized by group average agglomerative clustering on a sample. Other studies focus instead, on extracting hierarchical terms relationship built on subsumption relations between concepts, represented by the terms extracted from retrieved documents (Sanderson and Croft, 1999).

Most of these approaches extract “concepts” from textual collection, which generally are words, compound terms or term ensembles co-occurring in the documents. These approaches are based on typical NLP tasks such as stop-word removal, stemming, part-of-speech tagging, and then coupled with linguistic patterns selection and statistical analysis (Lau et al., 2007; Sanderson and Croft, 1999), supported by measures such as frequency, (balanced) mutual information, especially to select the domain concepts.

The vagueness of natural languages need more suitable methods and techniques that natively reflect the uncertainty and (fuzzy) reasoning of the human thought process. Fuzzy Logic is remarkably a simple way to process vague, ambiguous information, expressed in the natural language text. Fuzzy subsumption relations derived from term associations delineate hierarchical structures (Sanderson and Croft, 1999), semi-supervised fuzzy clustering methods return the special term ensembles, that are tuned with user suggestions on documents similarity (Loia et al., 2003).

A multi-view analysis of data is described in Loia et al. (2007), where a collaborative fuzzy clustering is applied on two different feature spaces extracted by the same dataset: one feature space is a more traditional, generated by the word frequency in the text, while the other space is built considering the mark-ups in the page; specifically, the semantic tags from the RDF [RDF] language appearing in the tagged documents.

Some other approaches work on structured text. Tags of the markup languages, such as HTML, XHTML, XML, etc., aim at capturing document structure to extract useful row information (Kruschwitz, 2003; Brunzel, 2008). Most data-driven approaches use clustering techniques to find term correlations that come from the document collection.

Agglomerative clustering (Chuang and Chien, 2005) for instance, produces hierarchical trees of term clusters; fuzzy clustering techniques (Loia et al., 2003, 2007) group the terms saving the fuzziness of the linguistic terms, and provide a membership degree to each generated concept-cluster.

Some approaches work on some special data, such as search log files, click data, etc. in order to extract concepts from user behaviour and collected profile (Baeza-Yates, 2007; Baeza-Yates and Tiberi, 2007; Boldi et al., 2008); techniques including both queries and URLs are also inspected: in Deng et al. (2009), Craswell and Szummer (2007), bipartite graphs are modelled starting by these data sources to find the related terminologies in documents. Some studies also focus on search links, especially the analysis of clicked links rather than the not clicked ones (Radlinski and Joachims, 2005).

Information-Driven Approaches

According to Fig. 1, the *Information* layer is composed of more complex data structures that come from the entities in the *Data* layer.

Information-driven approaches are mainly based on supervised approaches, targeted at extracting entity relationships for the concept extraction. Supervised methods usually require a training phase: syntactic and lexical aspects form the feature space to feed a learning algorithm, which classifies on the basis of a pre-annotated collection of relations between terms, named entities and

concepts. Several supervised methods are employed for Concept Mining (Brindha *et al.*, 2016) such as k-nearest neighbour, support vector machine, decision tree, etc. These methods are widely used by information-driven approaches because they work on a mid-level of syntactic data including part-of-speech (POS) tagging and named entities recognition.

A classification algorithm presented in Girju *et al.* (2006) recognizes true part-whole relations in text. This approach firstly retrieves candidate relationships from text, then the learning algorithm discovers the true relationships.

Some other methods extend pre-existing syntactic schemas. The algorithm proposed in Snow *et al.* (2006) incorporates the evidence from multiple classifiers over heterogeneous relationships to optimise a pre-existing taxonomy. Some studies (Reichartz *et al.*, 2009) present novel approaches based on the tree kernel method to extract association patterns, exploiting the complementary knowledge taken from grammar parse tree and dependency parse tree.

Other research focuses on kernel models to discover relations between named entities: in (Giuliano *et al.*, 2007), entity relations, WordNet synsets and hypernyms have been modelled as kernel functions, in order to extract relations among named entities.

Some other information-driven approaches adopt unsupervised methods. In particular, among the most common applications of these methods, there is document clustering, whose main aims are to extract topics and provide filtering on documents. Information-driven approaches perform document clustering employing named entities to capture document semantics, such as a presented fuzzy document clustering built on named entities (Cao *et al.*, 2008). Another approach prefers named entities to keywords for the definition of a vector space model, which is built on entity names, name-type pairs and identifiers. Then, the model generated is used to evaluate document similarity and perform hierarchical fuzzy clustering. In (Sinoara *et al.*, 2014) several experiments have demonstrated that hierarchical clustering based on named entities, as privileged information, gets better cluster quality and interpretation.

Semi-supervised methods are potentially very effective; a document clustering and classification is presented in Diaz-Valenzuela *et al.* (2016), where an automatic generation of the supervision is computed by the analysis of the data structure itself. The analysis is based on a partitional clustering algorithm that, discovering relationships between pairs of instances, has been used as a semi-supervision in the clustering process.

The approach described in Mintz *et al.* (2009) builds a hybrid framework to extract relations between named entities. This approach combines supervised and unsupervised methods, extracting relations from the semantic database Freebase [Freesbee] with the aim of providing supervision. All the sentences containing each pair of extracted entities, involved in relation, are retrieved in order to extract textual features to train a relation classifier.

As for automatic supervision, direct feedback from humans may also improve concept learning, especially in the IR field: a proposed agent-based framework provides support to user's web navigation, suggesting new web pages related to his/her previous searches. This framework retrieves similar pages to the ones visited by the user, who chooses his/her favourites, these remarks are transformed in semantic evaluations which are provided to a clustering algorithm aimed at extracting topics. Then term frequency or thesaurus-based techniques are employed to discover new terms useful to find new pages fitting user interests (Loia *et al.*, 2006).

Concept-Driven Approaches

Concept-driven models focus on building more refined data, expressing articulated concepts. They represent the upper layer of our abstract model described in Fig. 1.

These approaches generally achieve models, integrating the data, from local collections or sources with external knowledge sources, such as WordNet, Wikipedia, etc., in order to provide a complete and enriched domain knowledge model.

Domain model building is generally supported by using external tools. These tools are historically employed to extend lexicons with semantic relations, as well as to solve word-sense ambiguity and recognize right term meanings. Most of these tools are freely available, large-scale and provide high-quality data. Many methods require these tools to build knowledge bases on the domain (e.g., DBpedia, Yago), accomplish some specific tasks (e.g., word-sense disambiguation), perform linguistic and sentiment analysis (e.g., AlchemyAPI, WordNet) etc. One of the most used tools is WordNet, which is a lexical database providing knowledge about lexical features of text. Its capability of finding similar meaning words and put them in synonym groups (synsets) is widely used for disambiguation. Some studies have proposed methods to extend WordNet with further knowledge: sets of related words (topic signature) for topic discovery (Agirre *et al.*, 2000); further vocabulary, such as Longman dictionary of contemporary English [Longman]; ontology-based sentiment and emotion representation (Brença *et al.*, 2015); extension of complex concepts with relative descendants (e.g., restaurant for bistro, cybercafe) (Navigli, 2005).

Although the most of concept-driven methods focus on exploiting external sources, some solutions aim at enhancing the domain knowledge. The goal is to find how to enrich domain knowledge with new information and how new information can be included in the high-level knowledge representation. Some studies concentrate on searching a specific criterion to insert new information in the knowledge: the approach provided in Chen *et al.* (2008) evaluates the term distance from concept groups to map a new term within a concept. In this work, the Latent Semantic Analysis (LSA) technique is employed to strengthen the semantic characteristic of keywords and transform the term-document matrix into a high dimensional space based on collected web pages. Another latent model is Latent Dirichlet allocation (LDA), which is a generative statistical model for collection of discrete data, where each item of the collection is modelled as a finite mixture over an underlying set of topics. A document retrieval system has been built in Della Rocca *et al.* (2017), by exploiting both the latent models; it describes each document as an

ontology based network of concepts. In particular, this system adopts LDA model to extract topics from documents producing a granular knowledge on three levels (document, topic, word), enriched with semantic relations from WordNet and WordNet Domain [WordNet domain]. LSA provides instead a spatial distribution of the input documents, which allows to retrieve the most relevant documents and the correlated network of related topics and concepts.

Another formal framework used for the concept layer is Formal Concept Analysis (FCA). It supplies a basis for conceptual data analysis and knowledge processing. It enables the representation of the relationships between objects and attributes in a given domain. In the Text Mining domain, documents could be considered object-like and terms considered attribute-like (Cimiano *et al.*, 2005). FCA produces a one-to-one relationship between groups of similar concepts and a set of attributes, producing partially ordered relationships, like subsumption relation, between similar groups of concepts.

FCA has been used to produce formal ontologies, in fact many knowledge-based approaches employ FCA to produce high-level representations to model terms, concepts and their relationships; some fuzzy extension of FCA are proposed in (De Maio *et al.*, 2012) to build an ontology useful for web document view and organization. In (De Maio *et al.*, 2014), FCA has been flanked by the Relational Formal Context (RCA) with the purpose to automatically build an ex-novo ontology useful for semantic annotation of web pages.

Due to the nature of written languages, Fuzzy Logic has been used also for sentiment analysis and emotion detection. A framework for sentiment and emotion extraction from web resources has been introduced in Loia and Senatore (2014): it employs fuzzy sets to model sentiments and emotions extracted from text. The intensity of emotions, expressed according to Minsky's conception of emotions (Marvin, 2007), has been tuned by fuzzy modifiers, which act on the linguistic patterns recognized in the sentences.

Some other studies focus on ontology learning and enrichment. Ontolearn (Navigli *et al.*, 2004) for instance, is a framework capable of automatically generating an ontology, or extending a pre-existent schema, using external resources (e.g., dictionaries, thesauri), such as WordNet. Extending a pre-existing schema can be a non-trivial task, in fact an analysed framework enriches a core ontology with external glossary about named entities, such as WordNet and Dmoz taxonomy (Navigli and Velardi, 2006). Another study proposes a semi-automated model which uses a starting ontology, whose instances and lexicalizations are used to extract new candidates, in order to enhance the ontology knowledge (Valarakos *et al.*, 2004). Ontology learning is employed to reduce human involvement.

In order to discover new meaningful term and concept relations researchers have inspected inference methods, which enable to enhance the knowledge base about the term and concept context with new facts about term relationships. Artequkat (Alani *et al.*, 2003) is a framework able to extract knowledge from web, store knowledge and produce artist biographies. Artequkat extracts knowledge from web pages by some syntactic and semantic analyses and employs Gate and WordNet to enrich an a priori defined ontology which is stored. Then the biography generator can query the knowledge base by using an inference engine.

Some other studies present new high-level structures, other than ontologies, to represent and model high-level knowledge about terms, such as graphs or automata. An example is provided in Shehata *et al.* (2013), proposing a concept-based retrieval model which employs the conceptual ontological graph (COG) representation to capture semantic features of the term, such that each term is ranked according to its contribution to the meaning of the sentence. The top ranked concepts are chosen to build a concept-based document index for text retrieval.

A Schematic Description of a Multi-Layer Knowledge Representation

In the light of the layered stratification of the knowledge, described in Section A Layered Multi-Granule Knowledge Model, the main frameworks and tools presented across this article will be further analysed and classified, according to the given layered knowledge model. To this purpose, some salient features/aspects have been selected to evidence peculiarity and/or similarity in the basic approaches, at each knowledge layer.

Tables 1–3 show indeed the main frameworks, whose methodologies and implementation design produce a knowledge model that better reflects a knowledge layer of Fig. 1. The selected features are mainly five, described as follows.

1. Research sub-field: this feature identifies the research areas where the framework and tools are located, according to the methodologies, functionalities and techniques employed. The feature highlights the synergies between the different areas involved in the knowledge structuring, through an incremental informative granulation that yields the knowledge representation.
2. Knowledge representation: reports formal methods used for the knowledge extraction from text. It presents the methodologies and the technologies employed to represent and model the knowledge extracted.
3. Ontology-based support: evidences the role of external ontologies or ontology-based tools, in supporting the concept-based knowledge representation. Referencing to concepts of existing ontologies to describe entities is becoming a common practice in Text Mining.
4. Similarity measure: presents a primary feature, aimed at discovering low-level informative granules of knowledge. The similarity is the basic measure to compare text entities, such as words, terms, named entities, concepts, targeted at discovering syntactic or semantic relationships. Syntactic similarity concerns the sentence structure, exploiting for instance, the root or the lemma of a term; the semantic similarity is more complex to elicit: it discerns the correct sense (or concept) behind the term (or sentences) to get the contextual, actual meaning.

Table 1 Data-driven approaches test results

<i>Approach</i>	<i>Research sub-fields</i>	<i>Knowledge representation</i>	<i>Ontology-based support</i>	<i>Similarity measure</i>	<i>Semantic annotation support</i>
Phillips (1985)	NLP, clustering conceptual maps	Syntagmatic lexical networks	n.a.	Co-occurrence similarity	No
Schütze (1998)	IR, EM clustering	Word space model	n.a.	Semantic similarity	No
Sanderson and Croft (1999)	IR, subsumption relation	Subsumption relation-based hierarchy	n.a.	Semantic similarity	No
Loia <i>et al.</i> (2003)	SW, P-FCM	Word space model	n.a.	Proximity measure	Yes
Loia <i>et al.</i> (2007)	SW, P-FCM, Collaborative clustering	Word and RDF-tag space model	n.a.	Proximity measure	Yes
Lau <i>et al.</i> (2007)	NLP, POS tagging	Fuzzy subsumption relation hierarchy	n.a.	Term frequency, mutual information	No
Kruschwitz (2003)	IR	HTML tag structure	n.a.	Semantic similarity	No
Chuang and Chien (2005)	IR, agglomerative clustering	Suffix hierarchy	n.a.	Topological measure	No
Baeza-Yates (2007)	IR, graph theory	graph-based relations	n.a.	Semantic measure	No

Table 2 Information-driven approaches test results

<i>Approach</i>	<i>Research sub-fields</i>	<i>Knowledge representation</i>	<i>Ontology-based support</i>	<i>Similarity measure</i>	<i>Semantic annotation support</i>
Girju <i>et al.</i> (2006)	IR classification rule extraction	Part-whole relations	n.a.	Topological similarity	No
Snow <i>et al.</i> (2006)	IR classification	Hyponymy hypernymy	WordNet	Taxonomy	Yes
Reichartz <i>et al.</i> (2009)	IR, kernel methods	Parse tree	n.a.	Phrase similarity	No
Giuliano <i>et al.</i> (2007)	IR, kernel methods	Entity relations (Synsets, Hypernym)	WordNet	Semantic similarity, WordNet synsets	Yes
Cao <i>et al.</i> (2008)	IR, hierarchical fuzzy clustering	Fuzzy vector space model	n.a.	Fuzzy similarity measure	No
Diaz-Valenzuela <i>et al.</i> (2016)	IR, document clustering partitionial clustering	Clustering-based constraints	n.a.	Frequency measure	No
Mintz <i>et al.</i> (2009)	IR relation extraction classification	Free base relations	Freebase	Entity-relation model	Yes
Loia <i>et al.</i> (2006)	IR, topic extraction proximity-based fuzzy clustering	Fuzzy multiset	n.a.	Fuzzy measure	Yes

5. Semantic annotation support: semantic annotation is a new way to represent knowledge in the form of concepts, which is far from the textual annotations on the content of documents. The feature indicates if the annotation is retrieved by pre-existent or ad-hoc defined ontologies. Tools that achieve IR tasks using semantic web technologies often carry out annotation tasks.

These features are shown in **Tables 1–3** with respect to the main frameworks, tools, presented in the article, and classified according to the three knowledge layers.

As shown in **Table 1**, data-driven approaches work on low-level data. These approaches mainly adopt linguistic, statistical and unsupervised methods, such as clustering, word-space model and subsumption relations. The knowledge representation is strongly based on term-to-term relationship, often generating flat ensemble of terms. In some cases, other formal methods, such as Formal Concept Analysis (FCA), fuzzy set or graph theory, are also used, that produce a kind of term structuring. Data-driven approaches do not exploit semantic annotation or require external ontology-based tool support. Although the work presented in Loia *et al.* (2007) may seem an exception, it is a tentative to combine two different data spaces extracting from the same dataset, but describing terms and (semantic-oriented) RDF-tags, in order to mix the data with different a feature nature, in the clustering generation.

Table 2 shows information-driven approaches that usually extract relations between keywords or (more complex key-phrases) named entities adopting supervised or semi-supervised methods. The enhancement with respect to previous layer is that these approaches exploit topological representations for row data extracted from text. Formal models such as fuzzy set, part-whole

Table 3 Concept-driven approaches test results

<i>Approach</i>	<i>Research sub-fields</i>	<i>Knowledge representation</i>	<i>Ontology-based support</i>	<i>Similarity measure</i>	<i>Semantic annotation support</i>
Della Rocca <i>et al.</i> (2017)	IR, conceptual analysis	LDA-based concept learning	SKOS, WordNet	Statistical (LSA)	Yes
Cimiano <i>et al.</i> (2005)	IR, formal concept analysis (FCA)	FCA-based hierarchy	n.a.	Frequency measure, topological measure	No
De Maio <i>et al.</i> (2012)	IR, Conceptual analysis	Fuzzy FCA-based hierarchy	DBpedia, WordNet	Hierarchical	Yes
Loia and Senatore (2014)	Sentiment analysis, sentic computing	Fuzzy modifiers, fuzzy sets	WordNet, SentiWordNet	Semantic similarity, WordNet synsets	Yes
Navigli <i>et al.</i> (2004)	NLP, statistical approaches	Taxonomy learning	WordNet, FrameNet, VerbNet	Topological similarity	Yes
Navigli and Velardi (2006)	NLP, statistical approaches	Taxonomy learning	WordNet, Dmoz	Topological similarity, synsets	No
Valarakos <i>et al.</i> (2004)	NLP, knowledge discovery	Ontology learning, HMM	n.a.	Statistical similarity	Yes
Alani <i>et al.</i> (2003)	IR, knowledge extraction	Syntactic analysis, semantic analysis	Gate, WordNet	Semantic similarity	Yes
Shehata <i>et al.</i> (2013)	IR, concept extraction	Conceptual ontological graph (COG)	n.a.	Topological similarity	No
Agirre <i>et al.</i> (2000)	NLP, word-sense disambiguation	Topic signature WordNet concepts	WordNet	Topological similarity	No

relations, entity relations induct richer, semantic relations between named entities. The revealed tendency is to find patterns useful to group named entities and keywords, and clustering methods are largely used in order to fulfil this task. Some approaches gather additional information from external tools, such as thesauri and knowledge bases, in order to better identify the term sense and then produce more meaningful relations among NE and keywords. [Table 2](#) highlights that information-driven approaches are mainly used in Information Retrieval field.

Concept-driven approaches shown in [Table 3](#) are aimed at producing a more refined representation of the corpus in input, achieving a knowledge structuring that evidences a deeper granularity of the information. These approaches focus mainly on building a conceptual map or term-dependency network that allow the high-level knowledge description. At this purpose, they propose various methodologies based on formal models that reveal hierarchies or tree-based structures, such as Formal Concept Analysis (FCA), Conceptual Ontological Graph (COG); fuzzy modifiers are introduced to capture the vagueness in the written text, that is hidden behind lexical relations and grammar dependencies; then, they exploit these semantic and lexical connections to get the taxonomy and ontology learning.

Most of these approaches extract knowledge from external sources, but some of them build a conceptual structure based exclusively on the analysed text corpora. External support includes both syntactical and semantic sources and tools, for instance, WordNet, whose synsets yield synonyms for each term sense, useful to contextualize concepts. Other external sources are knowledge bases, such as DBpedia, verb-lexicon (e.g. VerbNet [VerbNet]), as well as more sophisticated semantic tools, such as semantic frameworks (e.g. FrameNet [FrameNet]) and ontology-based knowledge organization schema (SKOS [SKOS]). Concept-driven approaches are used as well in Information Retrieval and Natural Language Processing areas.

Some approaches seem to prefer methodologies, capable of producing a topological conceptualisation of data, such as Formal Concept Analysis (FCA), Fuzzy sets, Graph Theory, etc. which are largely used to extract relations, patterns and recognize articulated concepts and topics. Since the Concept Mining approaches aim at generating more complex topological structures, they often combine methodologies and technologies from the three fields introduced in Section A Layered Multi-Granule Knowledge Model, NLP, IR, SW.

In conclusion, the Text Mining approaches presented in the literature combine methodologies from the three fields and subfields (IR is mainly adopted in the information-driven ones). Topological semantic similarities are the most used similarity measures, since they are more effective than frequency-based and statistical measures to extract relations between terms or concepts. Although, data-driven approaches are based on term frequency-based measures, such as co-occurrence measure, tf-idf, mutual information, to assign a weight to each word, information and concept-driven approaches employ semantic similarity, especially topological measures, in order to better represent complex relationships among articulated concepts. The most used measure act on lexical similarity, to extract hyponymy, hypernymy and WordNet synsets, as well as fuzzy measures, which provide a more sensible evaluation of the ambiguousness about NE and concepts.

The use of external sources seems more useful when dealing with high-level input data, i.e., concepts, or when the modelling requires higher-level conceptualisations. The semantic annotation support is instead frequent in the concept-driven and information-driven approaches, while it is almost missing with data-driven approaches (see [Table 1](#)). The data-driven approaches

indeed, work mainly on row data that generate not very complex knowledge (often composed of term ensembles), so the semantic annotation tools are not required at this stage. The use of external ontology-based tools is instead predominant when the knowledge becomes articulated and generates specialized conceptualizations. Table 3 indeed provides a list of concept-driven approaches whose generated knowledge is enhanced by external semantic sources and databases.

Conclusions

The knowledge extraction and modelling need consolidated Text Mining techniques in order to achieve a clear understanding of semantics inside the text. The natural language interpretation is still an open issue that requires enhanced tasks from different but interrelated research domains, such as Artificial Intelligence, Semantic Web, NLP, etc. This paper presents an open-minded outlook on the main methodologies and new approaches to Concept Mining on text corpora, projected in the data-information-concept continuum. The knowledge has been described through a multi-layer schema that presents for each layer, a different knowledge granularity (from a single word to complex linguistic expressions or concepts) and shows the approaches implementing the corresponding information layer. A classification of the main approaches and works in Text Mining have been presented and discussed, according to these granulation levels, evidencing that generally, approaches modelling the same knowledge level have the same tasks and require the similar structuring of data. It is highlighted how, especially in the big data era, the knowledge structuring is a mandatory task to join ad-hoc methodologies and technologies, to improve the machine-oriented knowledge understanding and to guarantee a global knowledge enrichment.

See also: Biomedical Text Mining. Gene Prioritization Tools. Homologous Protein Detection. Natural Language Processing Approaches in Bioinformatics. Protein Functional Annotation. Text Mining Applications. Text Mining Basics in Bioinformatics. Text Mining for Bioinformatics Using Biomedical Literature. Text Mining Resources for Bioinformatics

References

- Agirre, E., *et al.*, 2000. Enriching very large ontologies using the WWW. In: Proceedings of the First International Conference on Ontology Learning – Volume 31, OL'00, pp. 25–30. Berlin, Germany: CEUR-WS.org.
- Alani, H., *et al.*, 2003. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems* 18, 14–21.
- Ambika, P., Rajan, M.R.B., 2016. Survey on diverse facets and research issues in social media mining. In: 2016 International Conference on Research Advances in Integrated Navigation Systems, (RAINS), pp. 1–6.
- Auer, S., *et al.*, 2007. DBpedia: A nucleus for a web of open data. In: 6th International the Semantic Web and Proceedings of the 2nd Asian Conference on Asian Semantic Web Conference. ISWC'07/ASWC'07. pp. 722–735. Busan: Springer-Verlag.
- Baeza-Yates, R., 2007. Graphs from search engine queries. In: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science. SOFSEM '07. pp. 1–8. Harrachov: Springer-Verlag.
- Baeza-Yates, R., Tiberi, A., 2007. Extracting semantic relations from query logs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '07. San Jose, pp. 76–85. CA: ACM.
- Ben-Yitzhak, O., *et al.*, 2008. Beyond basic faceted search. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. WSDM '08. pp. 33–44. Palo Alto, CA: ACM.
- Boldi, P., *et al.*, 2008. The query-flow graph: Model and applications. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. CIKM '08. pp. 609–618. Napa Valley, CA: ACM.
- Bollacker, K., *et al.*, 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. SIGMOD '08. pp. 1247–1250. Vancouver, BC: ACM.
- Brenga, C., *et al.*, 2015. SentiWordSKOS: A lexical ontology extended with sentiments and emotions. In: Conference on Technologies and Applications of Artificial Intelligence, TAAI 2015, Tainan, Taiwan, November 20–22, 2015, pp. 237–244.
- Brindha, S., Prabha, K., Sukumaran, S., 2016. A survey on classification techniques for text mining. In: 2016 Proceedings of the 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 01, pp. 1–5.
- Brunzel, M., 2008. The XTREEM methods for ontology learning from web documents. In: Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge. pp. 3–26. Amsterdam: IOS Press.
- Cambria, E., White, B., 2014. Jumping NLP curves: A review of natural language processing research. *IEEE Comput. Intell. Mag.* 9 (2), 48–57.
- Cao, T.H., *et al.*, 2008. Fuzzy named entity-based document clustering. In: 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pp. 2028–2034.
- Case, D.O., 2012. Looking for Information: A Survey of Research on Information Seeking, Needs and Behavior. Library and Information Science. Bingley: Emerald Group Publishing.
- Chen, R.C., *et al.*, 2008. Upgrading domain ontology based on latent semantic analysis and group center similarity calculation. In: 2008 IEEE International Conference on Systems, Man and Cybernetics, pp. 1495–1500.
- Chuang, S.-L., Chien, L.-F., 2005. Taxonomy generation for text segments: A practical web-based approach. *ACM Trans. Inf. Syst.* 23 (4), 363–396.
- Cimiano, P., Hotho, A., Staab, S., 2005. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Int. Res.* 24 (1), 305–339.
- Clark, M., *et al.*, 2012. Automatically structuring domain knowledge from text: An overview of current research. *Inf. Process. Manag.* 48 (3), 552–568.
- Craswell, N., Szummer, M., 2007. Random walks on the click graph. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '07. pp. 239–246. Amsterdam: ACM.
- Della Rocca, P., Senatore, S., Loia, V., 2017. A semantic-grained perspective of latent knowledge modeling. *Inf. Fusion* 36, 52–67.
- De Maio, C., *et al.*, 2014. Formal and relational concept analysis for fuzzy-based automatic semantic annotation. *Appl. Intell.* 40 (1), 154–177.

- De Maio, C., *et al.*, 2012. Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. *Inf. Process. Manag.* 48 (3), 399–418. (Soft Approaches to {IA} [on the Web]).
- Deng, H., King, I., Lyu, M.R., 2009. Entropy-biased models for query representation on the click graph. In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '09*. pp. 339–346. Boston, MA: ACM.
- Diaz-Valenzuela, I., *et al.*, 2016. Automatic constraints generation for semisupervised clustering: Experiences with documents classification. *Soft Comput.* 20 (6), 2329–2339.
- Erkan, G., Radev, D.R., 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* 22 (1), 457–479.
- Fader, A., Soderland, S., Etzioni, O., 2011. Identifying relations for open information extraction. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11*. pp. 1535–1545. Edinburgh: Association for Computational Linguistics.
- Finkel, J.R., Grenager, T., Manning, C., 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05*. pp. 363–370. Ann Arbor, MI: Association for Computational Linguistics.
- Girju, R., Badulescu, A., Moldovan, D., 2006. Automatic discovery of part-whole relations. *Comput. Linguist.* 32 (1), 83–135.
- Giuliano, C., *et al.*, 2007. FBK-IRST: Kernel methods for semantic relation extraction. In: *Proceedings of the 4th International Workshop on Semantic Evaluations. SemEval '07*. pp. 141–144. Prague: Association for Computational Linguistics.
- Golovchinsky, G., Qvarfordt, P., Pickens, J., 2009. Collaborative information seeking. *Computer* 42 (3), 47–51.
- Ilakiya, P., Sumathi, M., Karthi, S., 2012. A survey on semantic similarity between words in semantic web. In: *2012 International Conference on Radar, Communication and Computing (ICRCC)*, pp. 213–216.
- Fagan, L.J., 1989. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *J. Am. Soc. Inf. Sci.* 40 (2), 115. (Last updated – 24-02-13).
- Kalogeratos, A., Likas, A., 2012. Text document clustering using global term context vectors. *Knowl. Inf. Syst.* 31 (3), 455–474.
- Kathuria, M., Nagpal, C.K., Duhane, N., 2016. A survey of semantic similarity measuring techniques for information retrieval. In: *2016 Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 3435–3440.
- Khan, K., *et al.*, 2009. Mining opinion from text documents: A survey. In: *2009 Proceedings of the 3rd IEEE International Conference on Digital Ecosystems and Technologies*, pp. 217–222.
- Kjersti, A., Line, E., 1999. Text categorisation: A survey. Report No. 941. ISBN: 82-539-0425-8.
- Kozareva, Z., Hovy, E., 2010. A semi-supervised method to learn and construct taxonomies using the web. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. EMNLP '10*. pp. 1110–1118. Cambridge, MA: Association for Computational Linguistics.
- Kruschwitz, U., 2003. An adaptable search system for collections of partially structured documents. *IEEE Intell. Syst.* 18 (4), 44–52.
- Lau, R.Y.K., Bruza, P.D., Song, D., 2008. Towards a belief-revision-based adaptive and context-sensitive information retrieval system. *ACM Trans. Inf. Syst.* 26 (2), 8:1–8:38.
- Doyle, B.L., 1961. Semantic road maps for literature searchers. *J. ACM* 8 (4), 553–578.
- Lau, R.Y.K., *et al.*, 2007. Towards context-sensitive domain ontology extraction. In: *System Sciences, 2007. HICSS 2007. Proceedings of the 40th Annual Hawaii International Conference on*, pp. 60–60.
- Liu, X., Webster, J.J., Kit, C., 2009. An extractive text summarizer based on significant words. In: *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy. ICCPOL '09*. pp. 168–178. Hong Kong: Springer-Verlag.
- Loia, V., *et al.*, 2006. Web navigation support by means of proximity-driven assistant agents. *JASIST* 57, 515–527.
- Loia, V., Pedrycz, W., Senatore, S., 2003. P-FCM: A proximity-based fuzzy clustering for user-centered web applications. *Int. J. Approx. Reason.* 34 (2), 121–144.
- Loia, V., Pedrycz, W., Senatore, S., 2007. Semantic web content analysis: A study in proximity-based collaborative clustering. *IEEE Trans. Fuzzy Syst.* 15 (6), 1294–1312.
- Loia, V., Senatore, S., 2014. A fuzzy-oriented sentic analysis to capture the human emotion in Web-based content. *Knowl. Based Syst.* 58, 75–85. **Intelligent Decision Support Making Tools and Techniques: (IDSMT)**.
- Marchionini, G., White, R.W., 2009. Information-seeking support systems [Guest Editors' Introduction]. *Computer* 42 (3), 30–32.
- Minsky, M., 2007. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster.
- Matuszek, C., *et al.*, 2006. An introduction to the syntax and content of Cyc. In: *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pp. 44–49.
- Mintz, M., *et al.*, 2009. Distant supervision for relation extraction without labeled data. In: *Joint Conference Proceedings of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 – Volume 2. ACL '09*. pp. 1003–1011. Suntec: Association for Computational Linguistics.
- Navigli, R., 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.* 41 (2), 10:1–10:69.
- Navigli, R., *et al.*, 2004. Quantitative and qualitative evaluation of the OntoLearn ontology learning system. In: *Proceedings of the 20th International Conference on Computational Linguistics. COL-ING '04*. Geneva: Association for Computational Linguistics.
- Navigli, R., 2005. Semi-automatic extension of large-scale linguistic knowledge bases. In: *Proceedings of the 18th FLAIRS*, pp. 548–553.
- Navigli, R., Velardi, P., 2006. Ontology enrichment through automatic semantic annotation of on-line glossaries. In: Steffen, S., Svátek, V., *Managing Knowledge in a World of Networks: Proceedings of the 15th International Conference, EKAW 2006, Poděbrady, October 2–6, 2006*. pp. 126–140. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Navigli, R., Velardi, P., 2008. From glossaries to ontologies: Extracting semantic structure from textual definitions. In: *Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*. Press, pp. 71–87. Amsterdam: IOS.
- Phillips, M., 1985. *Aspects of Text Structure: An Investigation of the Lexical Organization of Text*, 52. Elsevier. (North-Holland linguistic series).
- Presutti, V., Draicchio, F., Gangemi, A., 2012. Knowledge extraction based on discourse representation theory and linguistic frames. In: ten Teije, A. (Ed.), *et al.*, *Knowledge Engineering and Knowledge Management: Proceedings of the 18th International Conference, EKAW 2012, Galway City, Ireland, October 8–12, 2012*. pp. 114–129. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Radlinski, F., Joachims, T., 2005. Query chains: Learning to rank from implicit feed-back. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. KDD '05*. pp. 239–248. Chicago, IL: ACM.
- Recupero, D.R., *et al.*, 2015. Extracting knowledge from text using SHELTON, a Semantic Holistic framework for Linked Ontology Data. In: *Proceedings of the 24th International Conference on World Wide Web. WWW '15 Companion*. pp. 235–238. Florence, Italy: ACM.
- Reichartz, F., Korte, H., Paass, G., 2009. Composite kernels for relation extraction. In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. ACLShort '09*. pp. 365–368. Suntec: Association for Computational Linguistics.
- Saiyad, N.Y., Prajapati, H.B., Dabhi, V.K., 2016. A survey of document clustering using semantic approach. In: *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 2555–2562.
- Salton, G., McGill, M.J., 1986. *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill, Inc.
- Salton, G., Wong, A., Yang, C.S., 1975. A vector space model for automatic indexing. *Commun. ACM* 18 (11), 613–620.
- Sanderson, M., Croft, B., 1999. Deriving concept hierarchies from text. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '99*. pp. 206–213. Berkeley, CA: ACM.
- Schütze, H., 1998. Automatic word sense discrimination. *Comput. Linguist.* 24 (1), 97–123.
- Shehata, S., Karray, F., Kamel, M.S., 2013. An efficient concept-based retrieval model for enhancing text retrieval quality. *Knowl. Inf. Syst.* 35 (2), 411–434.
- Sinoara, R.A., *et al.*, 2014. Named entities as privileged information for hierarchical text clustering. In: *Proceedings of the 18th International Database Engineering & Applications Symposium*. pp. 57–66. IDEAS '14. Porto: ACM, ISBN: 978-1-4503-2627-8.

- Snow, R., Jurafsky, D., Ng, A.Y., 2006. Semantic taxonomy induction from heterogeneous evidence. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. ACL-44. pp. 801–808. Sydney: Association for Computational Linguistics.
- Speer, R., Havasi, C., 2012. Representing general relational knowledge in ConceptNet 5. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23–25, 2012, pp. 3679–3686.
- Suchanek, F.M., Kasneci, G., Weikum, G., 2007. Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web. WWW '07. pp. 697–706. Banff, AB: ACM.
- Tombros, A., van Rijsbergen, C.J., 2004. Query-sensitive similarity measures for information retrieval. *Knowl. Inf. Syst.* 6.5, 617–642.
- Valarakos, A.G., *et al.*, 2004. Enhancing ontological knowledge through ontology population and enrichment. In: Motta, E. (Ed.), *et al.*, Engineering Knowledge in the Age of the Semantic Web: Proceedings of the 14th International Conference, EKAW 2004, Whittlebury Hall, UK, October 5–8, 2004. pp. 144–156. Berlin, Heidelberg: Springer Berlin Heidelberg, ISBN: 978-3-540-30202-5. doi: 10.1007/978-3-540-30202-5_10. Available at: http://dx.doi.org/10.1007/978-3-540-30202-5_10.
- Widdows, D., Cederberg, S., Dorow, B., 2002. Visualisation techniques for analysing meaning. In: Proceedings of the 5th International Conference on Text, Speech and Dialogue. TSD '02. pp. 107–114. London: Springer-Verlag, ISBN: 3-540-44129-8.
- Yang, C.L., Benjamasutin, N., Chen-Burger, Y.H., 2014. Mining hidden concepts: Using short text clustering and wikipedia knowledge. In: 2014 Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops, pp. 675–680.

Relevant Websites

<https://www.ibm.com/watson/alchemy-api.html>
AlchemyAPI.

<http://aksw.org/Projects/FOX.html>
FOX.

<https://framenet.icsi.berkeley.edu/fndrupal/>
FrameNet.

<https://developers.google.com/freebase/>
Freesbee.

<http://gate.ac.uk/>
Gate.

<http://kaon2.semanticweb.org/>
KAON.

<http://www.ldoceonline.com/>
Longman.

<http://ontolearn.org/>
OntoLearn.

<http://www.opencalais.com/>
OpenCalais.

<https://www.w3.org/RDF/>
RDF.

<https://www.lexalytics.com/semantria>
Semantria.

<https://www.w3.org/2004/02/skos/>
SKOS.

<http://verbs.colorado.edu/~mpalmer/projects/verbnet.html>
VerbNet.

<https://wordnet.princeton.edu/>
WordNet.

<http://wndomains.fbk.eu/>
WordNet domain.

Biographical Sketch



Danilo Cavaliere received master degree in computer science from University of Salerno, Italy in 2014. From 2015 he is predoctoral fellow at Computer Science Department and then at Department of Information and Electrical Engineering and Applied Mathematics (DIEM). He is currently a PhD student at the University of Salerno, Italy. His research interests are in the areas of artificial and computational intelligence, machine learning, data mining and knowledge discovery, areas in which he has published some paper.



Sabrina Senatore received MS degree in computer science from University of Salerno, Italy in 1999 and the PhD degree in computer science from University of Salerno, in 2004. From 2005 she is a faculty member at the University of Salerno. Her current position is associate professor of Computer Science at Department of Information and Electrical Engineering and Applied Mathematics, of the University of Salerno, Italy. She is a member of IEEE CIS Task Force on Intelligent Agents (TFIA) and she is also an editorial board member of the Applied Intelligence Journal and International Journal of Computational Intelligence Theory and Practice. Her current research interests include the development and application of intelligent systems based on the combination of techniques from Soft Computing, Computational Intelligence, Text Mining, Web Information Retrieval, Semantic Web, Machine Learning and Software Agents, areas in which she has published numerous papers.



Vincenzo Loia received BS degree in computer science from University of Salerno, Italy in 1985 and the MS and PhD degrees in computer science from University of Paris VI, France, in 1987 and 1989, respectively. From 1989 he is Faculty member at the University of Salerno. His current position is as Chair and Professor of Computer Science at Department of Management and Innovation Systems. He is the coeditor-in-chief of Soft Computing and the editor-in-chief of Ambient Intelligence and Humanized Computing, both from Springer. He is an associate editor of various journals, including the IEEE Transactions on System, Man and Cybernetics: Systems; IEEE Transactions on Fuzzy Systems; IEEE Transactions on Industrial Informatics; IEEE Transactions on the IEEE Transactions on Cognitive and Developmental Systems. His research interests include soft computing, agent technology, Web intelligence, Situational Awareness. He was principal investigator in a number of industrial R/&D projects and in academic research projects. He is author of over 390 original research papers in international journals, book chapters, and in international conference proceedings. He holds several roles in IEEE Society in particular for Computational Intelligence Society (Chair of Emergent Technologies Technical Committee, IEEE CIS European Representative, Vice-Chair of Intelligent Systems Applications Technical Committee).

Text Mining for Bioinformatics Using Biomedical Literature

Andre Lamurias and Francisco M Couto, Universidade de Lisboa, Lisboa, Portugal

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

DS	Distant supervision
IE	Information extraction
ML	Machine learning
NER	Named entity recognition

NLP	Natural language processing
POS	Part-of-speech
RE	Relationship extraction
UMLS	Unified Medical Language System

Introduction

Biomedical literature is one of the major sources of current biomedical knowledge. It is still the standard method researchers use to share their findings, in the form of articles, patents and other types of written reports (Hearst, 1999). However, it is essential that a research group working on a given topic is aware of the work that has been done on the same topic by other groups. This task requires manual effort and may take a long time to complete, due to the large quantity of published literature. One of the largest sources of biomedical literature is the MEDLINE database, created in 1965 and accessible through PubMed. This database contains over 23 million references to journal articles in the life sciences, and more than 860,000 entries were added in 2016. There are also other document repositories relevant to biomedicine, such as the European Patent Office, and [ClinicalTrials.gov](https://clinicaltrials.gov/).

Automatic methods for Information Extraction (IE) aim at obtaining useful information from large datasets, where manual methods would be unfeasible. Text mining aims at using IE methods to process text documents. The main challenge of text mining is in developing algorithms that can be applied to unstructured text to obtain useful structured information. Biomedical literature is particularly challenging to text mining algorithms for several reasons. The writing style differs from other types of literature since it is more formal and complex. Furthermore, different types of documents have different styles, depending on whether the document is a journal paper, patent or clinical report (Friedman *et al.*, 2002). Finally, there are a wide variety of terms that can be used, referring to genes, species, procedures, and techniques and, within each specific term, it is also common to have multiple spellings, abbreviations and database identifiers. These issues make biomedical text mining an interesting field for which to develop tools, due to the challenges that it presents (Cohen and Hunter, 2004).

The interactions found in the biomedical literature can be used to validate the results of new research or even to formulate new hypotheses to be tested experimentally. One of the first demonstrations of the hidden knowledge contained in a large literature was Swanson's ABC model (Swanson, 1990), who found that dietary fish oils might benefit patients with Raynaud's syndrome, by connecting the information present in two different sets of articles that did not cite each other. This inference has been independently confirmed by others in clinical trials (DiGiacomo *et al.*, 1989). In the same study, Swanson provided two other examples of inferences that could not be drawn from a single article, but only by combining the information of multiple articles. Considering that, since that study, the number of articles available has grown immensely, it is intuitive that many new chemical interactions might be extracted from this source of information.

More recently, bioinformatics databases have adopted text mining tools to more efficiently identify new entries. MirTarBase (Chou *et al.*, 2016) is a database of experimentally validated miRNA-target interactions published in journal papers. The curators of this database use a text mining system to identify new candidate entries for the database, which are then manually validated. This system was necessary due to the important role miRNAs have been found to play in human diseases over the last decade, leading to a high number of papers published about this subject. The introduction of the system as part of the workflow has led to a 7-fold increase in the number of interactions added to the database.

Text mining has generated much interest in the bioinformatics community in recent years. As such, several tools and applications have been developed, based on adaptations of text mining techniques to diverse problems and domains. This article provides a survey of biomedical text mining tools and applications that demonstrate the usefulness of text mining techniques. The rest of the paper consists of the following: Section Background/Fundamentals provides the basic concepts of text mining relevant to this article, Section Text Mining Toolkits describes some toolkits that can be used to develop text mining tools, Section Biomedical Text Mining Tools describes the most used text mining tools, and Section Applications describes applications built using those tools that have been distributed to the general public. Section Community Challenges provides a summary of the community challenges organized to evaluate biomedical text mining tools. Finally, Section Future Directions suggests future directions for biomedical text mining tools and applications, and Section Closing Remarks summarizes the main conclusions of the article.

Background/Fundamentals

When developing and using text mining tools, it is necessary to first define what type of information should be extracted. This decision will then influence the datasets to be considered, which text mining tasks will be explored, and which tools will be used. The objective of

this section is to provide an overview of the options available to someone interested in developing a new text mining tool or using text mining for their work. The concepts presented are simple to understand and applicable to various problems.

NLP Concepts

Natural Language Processing (NLP) has been the focus of many researchers since the 1950s (Bates, 1995). The main difference between NLP and text mining is the objective of the tasks. While NLP techniques aim at making sense of the text, for example, determining its structure or sentiment, the objective of text mining tasks is to obtain concrete structured knowledge from text. However, there is overlap between the two fields, and text mining tools usually make use of NLP concepts and tasks.

The following list defines NLP concepts relevant to text mining.

Token: a sequence of characters with some meaning, such as a word, number or symbol. The NLP task of identifying the tokens of a text is known as tokenization. It is of particular importance to text mining since most algorithms will not consider elements smaller than tokens.

Part-of-speech (POS): the lexical category of each token, for example, noun, adjective, or punctuation. The category imparts additional semantics to the tokens. Part-of-speech tagging is an NLP task that consists in classifying each token automatically.

Lemma and stem: the base form of a word. The lemma represents the canonical form of the word, corresponding to a real word. The stem does not always correspond to a real word, but only to the fragment of a word that never changes. For example, the lemma of the word “induces” is “induce” while the stem is “induc-”.

Sentence splitting: the NLP task consisting of identifying the sentence boundaries of a text. The methods used to accomplish this task should consider the difference between a period at the end of a sentence, and at the end of an acronym or abbreviation. It is desirable to break a document into sentences because they represent unique ideas. Although the context of the whole document is also important, extracting the knowledge of each sentence independently can provide useful results.

Entity: a segment of text with relevance to a specific domain. An entity may be composed of one or more tokens. Entity types relevant to biomedicine include genes, proteins, chemicals, cell lines, species, and biological processes.

Text Mining Tasks

Text mining tools focus on one or more text mining tasks. It is necessary to define these tasks properly so that it is possible to choose the type of tools that should be used for a given problem. Furthermore, these tasks are used to evaluate the performance of a tool on community challenges. The text mining tasks presented here are common to all domains and sources of text, although the performance of the methods on different domains may differ, i.e., a method that has a good performance on patent documents may not perform as well on clinical reports, due to the different characteristics of the text. The common final objective of these tasks, as to all text mining, is to extract useful knowledge from a high volume of documents, while the extracted knowledge can be useful for several applications, which will be described in Section Applications.

Topic modeling: the classification of documents according to their topics or themes. The objective of this task is to organize a set of documents to identify which documents are more relevant to a given topic (Blei, 2012). Related tasks include document triage (Buchanan and Loizides, 2007) and document clustering.

Named Entity Recognition (NER): consists of identifying entities that are mentioned in the text. In most cases, the exact location of each entity in the text is required, given by the offset of its first and last character. In some cases, discontinuous entities may be considered, therefore requiring multiple offset pairs. The classification of entity properties such as its type (e.g., protein, cell line, chemical) can be included in this task (Nadeau and Sekine, 2007).

Normalization: consists of matching each entity to an identifier belonging to a knowledge base that unequivocally represents its concept. For example, a protein may be mentioned by its full name or by an acronym; in this case, the normalization process should assign the same identifier to both occurrences. The identifiers can be provided by an external database or ontology (Tsuruoka *et al.*, 2008). Related tasks include named entity disambiguation (Bunescu and Pasca, 2006), entity linking, and harmonization.

Relationship Extraction (RE): the identification of entities that participate in a relationship described in the text. Most tools consider relations between two entities in the same sentence. Biomedical relations commonly extracted are protein-protein and drug-drug interactions, see (Segura-Bedmar *et al.*, 2014), for example.

Event extraction: can be considered an extension of the relationship extraction task, where the label of the relationship and role of each participant is specified. The events extracted should represent the mechanisms described in the text (Ananiadou *et al.*, 2010). Related task: slot-filling.

Text Mining Approaches

To accomplish the tasks described above, text mining tools employ diverse approaches. They may focus on one specific approach, or combine several approaches according to their respective advantages, the latter being more common. Most approaches can also be adapted for performing multiple tasks.

Classic approaches: approaches based on statistics that can be calculated on a large corpus of documents (Manning *et al.*, 1999). Some of the most popular approaches are term frequency – inverse document frequency for topic modeling, and

Table 1 Corpora relevant to biomedical text mining tasks

Name	Reference	Annotations	Document types
CRAFT	Bada <i>et al.</i> (2012)	Biomedical entities	Full-text articles
MedTag	Smith <i>et al.</i> (2005)	Biomedical entities	PubMed abstracts
Genia	Kim <i>et al.</i> (2003)	Biomedical entities and events	PubMed abstracts
CHEMDNER	Krallinger <i>et al.</i> (2015a)	Chemical compounds	PubMed abstracts
CHEMDNER-patents	Krallinger <i>et al.</i> (2015b)	Chemical compounds and proteins	Patent abstracts
DDI	Herrero-Zazo <i>et al.</i> (2013)	Drug-drug interactions	Drug descriptions and journal abstracts
SeeDev	Chaix <i>et al.</i> (2016)	Seed development events	Full-text articles
Thyme	Styler <i>et al.</i> (2014)	Events and time expressions	Clinical notes
MLEE	Pyysalo <i>et al.</i> (2012)	Biological events	PubMed abstracts

co-occurrence for relationship extraction. These approaches preceded the popularization of machine learning algorithms, although most current approaches still have a statistical background.

Rule-based methods: consist of defining a set of rules to extract the desired information. These rules can be a list of terms, regular expressions or sentence constructions. Due to the manual effort necessary to develop these rules, text mining tools based on this approach have limited applicability.

Machine learning (ML) algorithms: are used for automatically learning various tasks. In the specific case of text mining, it is necessary to convert the text to a numeric representation, which is the expected input of these algorithms. Text mining tools using ML contain models trained on a corpus, that can then be applied to other texts. In some cases, it may be possible to train additional models using other corpora. Several types of ML approaches can be considered, for example, supervised learning, in which the labels of each instance of the training data are known and used to train the classifier, and unsupervised learning, in which the algorithm learns to classify the data without a labeled training set.

Distant supervision (DS): a learning process which heuristically assigns labels to the data according to the information provided by a knowledge base. These annotations are prone to error, but using ML algorithms adapted to this method, it can provide effective classification models. Distant supervision is sometimes referred to as weak supervision.

Biomedical Corpora

Biomedical corpora are necessary to develop and evaluate text mining tools. The simplest corpora consist of a set of documents associated with a specific topic (e.g., disease, gene, or pathway). For some tasks, such as simple topic modeling tasks, it is enough to know which documents are relevant. However, most ML algorithms require annotated text to train their models. The type of annotations necessary to evaluate a task should be similar to the type of annotations to be extracted by the tools (NER tasks require text annotated with relevant entities, while relationship extraction requires the relations between the entities described in the text to be annotated). The annotations should be manually curated by domain experts according to established guidelines. Inter-annotator agreement measures, such as the kappa statistic (Carletta, 1996), can be used to assess the reliability of the annotations. However, text mining tools may also be used to help curators by providing automatic annotations as a baseline (Winnenburg *et al.*, 2008).

The size of an annotated corpus is limited by the manual effort necessary to annotate the documents. Simpler tasks, such as topic modeling, can be performed more quickly by human annotators, so it is less expensive to develop an annotated corpus for this task. Relationship extraction requires that the annotators first identify the entities mentioned in the text, and then the relationships described between the entities. For this reason, it is more expensive to develop an annotated corpus for this task. Biomedical text mining community challenges have contributed to the release of several annotated gold standards that can be used to evaluate systems. Section Community Challenges provides a summary of these challenges. Table 1 provides a list of annotated biomedical corpora relevant to various text mining tasks.

Text Mining Toolkits

Although biomedical text mining requires specialized approaches to deal with the characteristics of the biomedical literature, general text mining tools can be used as a starting point for more specialized approaches. These general tools can be adapted to specific domains, either by using models trained with biomedical datasets or by developing pre- and post-processing rules developed for this type of text. Text mining toolkits are a type of software that can perform various NLP and text mining tasks. The objective of these toolkits is to provide general-purpose methods for performing various text mining tasks, which can be adapted to specific problems. There are several toolkits available, that can be used to pre-process the data, compare the performance of various tools and approaches, and select the best combination for a specific problem. This section provides a survey of well-known text mining toolkits that have been used as frameworks of biomedical text mining tools. In addition to the toolkits presented here, tools can be developed from scratch using programming languages and libraries that implement specific algorithms.

One of the most widely used text mining toolkits is Stanford CoreNLP (Manning *et al.*, 2014), which aggregates various tools developed by the Stanford NLP team for processing text data. Biomedical text mining tools may use Stanford CoreNLP to pre-

process the data (e.g., for sentence splitting, tokenization and co-reference resolution) and to generate features for machine learning classifiers (e.g., for POS tagging, lemmatization, and dependency parsing).

NLTK (Bird *et al.*, 2009), another NLP toolkit, was implemented as a Python library. This toolkit provides interfaces to various NLP resources, such as Word-Net, tokenizers, stopwords lists, and datasets from community challenges. It is often used by developers who are getting started in text mining, due to its well-designed API, and to the availability of various online tutorials for this toolkit. More recently, another Python-based toolkit was released, spaCy, which is more focused on computational performance, using state-of-the-art algorithms.

ClearTK (Bethard *et al.*, 2014) is a text mining toolkit based on machine learning and the Apache Unstructured Information Management Architecture (UIMA). This framework provides interfaces to several machine learning libraries and feature extractors.

GATE (Cunningham *et al.*, 2013) is one of the few text mining toolkits which has features specially designed for biomedical text mining. This toolkit provides plugins for bioinformatics resources such as Linked Life Data and other ontologies, and specialized biomedical NLP tools. Furthermore, a graphical user interface is available to visualize and edit the data and system architecture.

Biomedical Text Mining Tools

This section describes text mining tools commonly used in bioinformatics. These tools generally focus on one specific task, presenting novel approaches, and are evaluated on gold standards. We focus on tools described in the literature and freely available to the community. Even though the current trend is to make software available on code repositories such as GitHub and Bitbucket, this has not always been the case, and past works may not be accessible if the source code was not shared with the community. The tools described in this section have been used in community challenges and may require considerable technical skill to apply to specific problems since the results provided by their developers often refer to gold standards and not to real-world use-cases. These tools are usually fine tuned to work with English texts, but automatic translation techniques have been shown to be effective when using texts in other languages (Campos *et al.*, 2017). Table 2 provides a list of biomedical text mining tools that are available to the community.

NER and Normalization

Biomedical text mining tools can be organized in terms of the text mining tasks performed. The biomedical community challenges organized in the last decade have motivated several teams to develop tools for bioinformatics and biomedical text mining. The focus of these challenges has been in recognizing genes, proteins and chemical compounds mentioned in texts, and linking those

Table 2 Text mining tools for bioinformatics and biomedical literature

Name	Reference	Tasks	Approaches	GUI
BANNER	Leaman <i>et al.</i> (2008)	NER	ML	N
ABNER	Settles (2005)	NER	ML	N
LingPipe	Carpenter (2007)	General NLP	ML and Rule-based	N
GNormPlus	Wei <i>et al.</i> (2015)	NER and Normalization	ML	N
DNorm	Leaman <i>et al.</i> (2013)	NER and Normalization	ML	N
tmChem	Leaman <i>et al.</i> (2015)	NER	ML	N
tmVar	Wei <i>et al.</i> (2013a)	NER	ML	N
GENIA tagger	Tsuruoka and Tsujii (2005)	NER and POS tagging	ML	N
GENIA sentence splitter	Sætre <i>et al.</i> (2007)	Sentence splitting	ML	N
Acronime	Okazaki and Ananiadou (2006)	Abbreviation resolution	Rule-based	Y
ONote	Lourenco <i>et al.</i> (2009)	NER, document retrieval	ML	Y
MetaMap	Aronson and Lang (2010)	NER and Normalization	Rule-based	Y
LDPMMap	Ren <i>et al.</i> (2014)	Normalization	Rule-based	N
SimSem	Stenetorp <i>et al.</i> (2011)	Normalization	Rule-based and ML	N
MER	Couto <i>et al.</i> (2017)	NER	Rule-based	N
IBEnt	Lobo <i>et al.</i> (2017)	NER and Normalization	Rule-based and ML	N
cTakes	Savova <i>et al.</i> (2010)	NER, normalization, and RE	Rule-based	Y
Neji	Campos <i>et al.</i> (2015)	NER and Normalization	ML and Rule-based	Y
jSRE	Giuliano <i>et al.</i> (2006)	RE	ML	N
DeepDive	Zhang (2015)	RE	ML/DS	N
IBRel	Lamurias <i>et al.</i> (2017)	RE	ML/DS	N
TEES	Björne <i>et al.</i> (2011)	Event extraction	ML and Rule-based	N
VERSE	Lever and Jones (2016)	Event extraction	ML	N
EventMine	Miwa <i>et al.</i> (2013)	Event extraction	ML	Y
Textpresso	Müller <i>et al.</i> (2004)	NER and RE	Rule-based	Y

terms to databases. This leads to an imbalance in the quantity and variety of tools available for NER and normalization when compared to other tasks.

BANNER (Leaman *et al.*, 2008) uses Conditional Random Fields (Sutton and McCallum, 2006) to perform NER of chemical compounds and genes. AB-NER (Settles, 2005) and LingPipe (Carpenter, 2007) use similar approaches, each one combining different techniques to improve the results on gold standards, by optimizing the system architecture and feature selection. LingPipe also performs other NLP tasks, such as topic modeling and part-of-speech tagging, while all three provide ways to train models on new data. More recently, other systems have combined machine learning algorithms and manual rules to achieve better results in the biomedical domain (Savova *et al.*, 2010; Campos *et al.*, 2015; Lobo *et al.*, 2017).

GNormPlus (Wei *et al.*, 2015) is a modular system for gene NER and normalization, performing mention simplification and abbreviation resolution to match each gene to an identifier, with higher accuracy, even when more than one species is involved. It is part of a set of NER tools developed by NCBI for various entity types, which includes tmChem (Leaman *et al.*, 2015), DNorm (Leaman *et al.*, 2013) and tmVar (Wei *et al.*, 2013a). These tools are often evaluated in text mining community challenges.

The GENIA project is responsible for various contributions to biomedical text mining, including an annotated corpus (Kim *et al.*, 2003) and various tools for text mining tasks. GENIA tagger (Tsuruoka and Tsujii, 2005) performs NER of several types of entities relevant to biomedicine (protein, DNA, RNA, cell line and cell types), as well as POS tagging. GENIA sentence splitter (Sætre *et al.*, 2007) is an ML-based tool for identifying sentence boundaries in biomedical texts, trained on the GENIA corpus. Acromine (Okazaki and Ananiadou, 2006) is another tool developed by the same team, with the purpose of providing definitions for abbreviations found in MEDLINE abstracts.

Since the vocabulary used in clinical records is quite different from other biomedical texts, tools have been developed specifically for this type of documents. These tools are based on the Unified Medical Language System (UMLS), a collection of vocabularies associated with the clinical domain. cTakes (Savova *et al.*, 2010) is a Java-based tool for processing clinical text, originally developed at the Mayo clinic, which performs several biomedical text mining tasks. It is possible to use this tool through a graphical user interface. Due to the large size and complex structure of UMLS, tools have been specifically developed just to find UMLS concepts in documents. Such tools include MetaMap (Aronson and Lang, 2010), and LDPMMap (Ren *et al.*, 2014). SimSem (Stenetorp *et al.*, 2011) is a tool for entity normalization, using string matching techniques and machine learning. This tool can match strings to a variety of bioinformatics knowledge bases, such as ChEBI, Gene Ontology, Entrez Gene, and UMLS. Couto *et al.* (2017) introduced a system, MER (Minimal Entity Recognizer), which can be easily adapted to different types of entities. This system requires only a file with one entity per line, and uses a simple matching algorithm to find those entities in text.

Relationship and Event Extraction

For RE, most tools use ML algorithms to classify which pairs of entities mentioned in the text constitute a relationship. In this task, kernel methods and Support Vector Machines are popular. jsRE (Giuliano *et al.*, 2006) uses a shallow linguistic kernel which takes into account the tokens, POS, and lemmas around each entity of the pair. It has been used for various problems, including drug-drug interaction extraction (Segura-Bedmar *et al.*, 2011).

Distant supervision has become particularly relevant to RE tasks because it is more expensive to develop a corpus annotated with relations. (Mallory *et al.*, 2016) developed an approach to gene RE using DeepDive, a general purpose system for training distantly supervised RE models. They applied this approach to a corpus of full-text documents from three journals, using the BioGRID and Negatome databases as reference. Another DS-based tool, IBRel (Lamurias *et al.*, 2017), uses TransmiR, a database of miRNA-gene associations, to extract the same type of relationships from text.

Biomedical event extraction is a complex task, but some tools have been developed. Turku Event Extraction System (TEES) (Björne *et al.*, 2011) identifies complex events based on trigger words and graph methods. This system has been evaluated on multiple community challenges, on event extraction and RE tasks, such as the BioNLP-ST 2011 event extraction task. In the 2016 edition of BioNLP-ST, (Lever and Jones, 2016) presented VERSE, a system for extracting relationships and events from text, and evaluated it on three different subtasks. This system is based on ML algorithms, and has the advantage of being able to extract relationships between entities in different sentences.

Textpresso (Müller *et al.*, 2004) is a system for biomedical information extraction based on regular expressions and ontologies. This system has been applied to various domains, and a portal to search the results obtained on each domain is provided in the web interface.

Applications

Even though it is important to develop methods for specific tasks, those methods will only be useful to the community if they can be easily used to help address biomedical problems. Since recent text mining tools have obtained good performance on evaluation corpora, efforts have been made deliver these tools to the general public. In this section, we present a survey of text mining applications that are available in the form of web pages and APIs that focus on the user experience. Table 3 provides a summary of these applications.

Some biomedical text mining applications simply provide access to a text mining tool via a web application. The user uploads one or more documents, which are processed by the tool in a server, and the results are delivered to the user. Even though this is an

Table 3 Bioinformatics applications that either use text mining tools or their results, accessible from the web

Name	Reference	API
Whatizit	Rebholz-Schuhmann et al. (2008)	Y
becas	Nunes et al. (2013)	Y
PubTator	Wei et al. (2013b)	Y
SciLite	Venkatesan et al. (2016)	Y
BEST	Lee et al. (2016)	N
STRING	Szklarczyk et al. (2017)	Y
STITCH	Szklarczyk et al. (2016)	Y
FACTA +	Tsuruoka et al. (2011)	N
Poly Search2	Liu et al. (2015)	Y
Evex	Hakala et al. (2013)	Y
MEDIE	Miyao et al. (2006)	N

important effort, it assumes that the user already has chosen the documents to be processed, and it depends on downstream applications to use the results. Whatizit ([Rebholz-Schuhmann et al., 2008](#)) is a text mining application that can be used to identify biomedical entities in text using a web browser or API. This application is based on a rule-based text mining system which annotates the documents submitted by users. The entities correspond to entries in biomedical knowledge bases, such as ChEBI and UniProt. The results are presented as a web page, where each entity type is marked with a different color. A similar application is BeCAS ([Nunes et al., 2013](#)), based on the Neji tool. With this application, it is also possible to access the results through a web browser or the API, which can then be exported to various file formats.

Other text mining applications provide pre-processed results, reducing the time necessary to obtain results. For example, PubTator ([Wei et al., 2013b](#)) contains every PubMed abstract, annotated with the NCBI NER tools, and it is updated as new abstracts are added to PubMed. Users can search for a list of abstracts or by keyword. It is possible to create a collection of abstracts, manually fix annotation errors, and download the results. PubTator provides access to the results through an API, for integration with other applications. For example, the Mark2Cure crowdsourcing project uses this API to provide a baseline of automatic annotations to its users, while the HuGE navigator knowledge base ([Yu et al., 2008](#)) relies on PubTator to improve its weekly update process. Another application based on pre-processed results is SciLite, a platform for displaying text mining annotations, which is integrated with the Europe PMC database ([Venkatesan et al., 2016](#)). This application shows a list of biomedical terms associated with each document, allowing users to endorse and report incorrect annotations to improve the text mining method. Biomedical Entity Search Tool (BEST) ([Lee et al., 2016](#)) uses text mining techniques to retrieve entities relevant to user queries. BEST is updated daily with the abstracts added to PubMed, and 10 types of entities are identified in each document.

The STRING database stores information about protein-protein interaction networks ([Szklarczyk et al., 2017](#)). It contains information obtained through various methods, including text mining. The interactions extracted using text mining methods are obtained from PubMed and a collection of full-text documents. The RE method used is based on co-occurrence of proteins in the same document, and presence of trigger words such as "binding" and "phosphorylation by." A related database, STITCH ([Szklarczyk et al. 2016](#)), uses a similar method to identify chemical-protein interactions based on the biomedical literature.

FACTA + ([Tsuruoka et al., 2011](#)) is a text mining application for identifying biomedical events described in PubMed abstracts. It uses both co-occurrence and machine learning approaches to extract relationships from text. The user can perform a keyword search to obtain associated documents and biomedical entities, such as genes, diseases, and drugs. Furthermore, FACTA + can be used to identify indirect relations between a concept and a type of biomedical entity. For example, it is possible to search for a disease name and obtain genes that are indirectly associated with that disease, through an intermediary disease, ranked by a novelty and reliability score.

PolySearch2 ([Liu et al., 2015](#)) can also identify relationships between biomedical concepts based on co-occurrence at the sentence level. With this application, it is possible to obtain all the entities, of a specific type, associated with the input query. The corpora and databases used by this application are stored locally and updated daily to ensure that the complete information is available to the users.

EVEX ([Hakala et al., 2013](#)) is a database of biomolecular events extracted from abstracts and full-text articles using text mining tools such as BANNER and TEES. This database contains more than 40 million associations between genes and proteins, and its data can be downloaded and accessed through an API, although it is not updated regularly. MEDIE ([Miyao et al., 2006](#)) contains biomolecular events extracted from MEDLINE, each event being composed of a subject, a verb, and an object. Using MEDIE, it is possible to search by subject, verb or object (or a combination of the three) and obtain all matching events extracted from the abstracts.

Community Challenges

Text mining challenges are organized regularly, by the community, with the purpose of evaluating the performance of text mining tools. These text mining challenges are open to the community, meaning that any academic or industry team can participate. Each

challenge usually comprises several tasks (sometimes referred to as tracks), each with a specific motivation, objective and gold standard. Each team may submit results to one or more tasks. Furthermore, the teams may develop their own tools, or adapt existing tools to the proposed task.

The task organizers announce the objectives of their task on the official websites of the challenge and on mailing lists. Since there are various data file formats used in text mining, a sample of the data may be provided to the participants at the same time as the announcement. This is also the case of datasets that require data use agreements. Afterward, the training set is provided to the participants, consisting of documents and annotations. This training set is used to develop or adapt tools and systems to the task. A development set may also be provided, similar in size to the training set, to further improve the systems. During the final phase of the challenge, a testing set is sent to the teams, without the gold standard annotations. The teams have a time period to submit the annotations obtained with their tools, which are then compared to the gold standard by the organizers. Each task has a defined set of measures to perform this evaluation and rank the teams. The results are then published on the challenge website and in a task overview paper.

One of the earliest NLP challenges, TREC, mainly focuses on the news domain, but it has included a bioinformatics task in some of its editions (TREC Genomics and TREC Chemistry). In 2003, this challenge had a task for retrieving documents related to gene functions (Hersh and Bhupatiraju, 2003), while in later years, more complex tasks have also been proposed (Hersh and Voorhees, 2009). Other NLP challenges, such as KDD Cup (Yeh *et al.*, 2002) and CoNLL (Farkas *et al.*, 2010), also include bioinformatics tasks. SemEval is a series of semantic analysis evaluations organized yearly, and in the most recent editions, there has been at least one task relevant to bioinformatics (Segura Bedmar *et al.*, 2013; Elhadad *et al.*, 2015; Bethard *et al.*, 2016).

Due to increasing interest in biomedical NLP and text mining, community challenges specifically for this domain have been organized. BioCreative was first organized in 2004, and it consisted of the identification of gene mentions and Gene Ontology terms in articles, and of gene name normalization (Hirschman *et al.*, 2005). Since then, five more editions of this challenge have been organized, with a wide variety of tasks. BioNLP-ST has organized various biomedical IE tasks, usually focused on a specific biological system such as seed development (Chaix *et al.*, 2016), epigenetics and post-translational modifications (Ohta *et al.*, 2011), and cancer genetics (Pyysalo *et al.*, 2015). Other community challenges relevant to biomedical text mining include JNLPBA (Kim *et al.*, 2004), BioASQ (Krithara *et al.*, 2016), i2b2 (Sun *et al.*, 2013), and ShARe/CLEF eHealth (Kelly *et al.*, 2014). Huang and Lu (2016) provides an overview of the community challenges organized over a period of 12 years.

Future Directions

More recent approaches to RE have explored deep learning techniques (Miwa and Bansal, 2016). Deep learning is an ML approach based on artificial neural networks that has become popular in the last few years due to its performance in fields such as speech recognition, computer vision, and text mining (LeCun *et al.*, 2015). In the case of text mining, deep learning is associated with word embeddings, which consist of vector representations of word frequencies, that are used as inputs to the networks. There are still few biomedical text mining systems using deep learning techniques. However, various resources are available for this purpose, such as software toolkits that implement these algorithms, as well as a set of resources generated from biomedical literature (Pyysalo *et al.*, 2013).

As NER, normalization, and relationship extraction tasks improve in terms of precision and recall, semantic and question answering techniques can be developed to explore the extracted information. Semantic similarity is a metric used to compare concepts, usually based on a text corpus or an ontology (Couto and Pinto, 2013). These measures can both improve text mining tools by estimating the coherency of the entities and relations extracted, and be improved by applications that can generate candidate entries that may be missing from the ontology (Pershina *et al.*, 2015). Furthermore, question-answering systems can use semantic similarity methods to provide answers with more accuracy (Lopez *et al.*, 2005).

Closing Remarks

There has been a considerable effort by the text mining community to develop and release tools and applications for bioinformatics and biomedical literature. The tools presented in this article use various methods to automate useful tasks, and they can be used by researchers who want to adapt it to their own needs. This article also presents various applications based on text mining results, which demonstrate real-world use-cases of text mining tools. The evolution of biomedical text mining methods has led to more efficient parsing of biomedical literature. These advances should affect how databases are created and maintained, and how documents are indexed by search engines. We expect that future bioinformatics search engines, instead of simply retrieving documents relevant to the query, will be able to directly answer user queries and generate new literature-based hypotheses.

Acknowledgements

This work was supported by the FCT through the PhD grant PD/BD/106083/2015 and LaSIGE Unit, ref. UID/CEC/00408/2013 (LaSIGE).

See also: Biomedical Text Mining. Data-Information-Concept Continuum From a Text Mining Perspective. Gene Prioritization Tools. Homologous Protein Detection. Natural Language Processing Approaches in Bioinformatics. Protein Functional Annotation. Text Mining Applications. Text Mining Basics in Bioinformatics. Text Mining Resources for Bioinformatics

References

- Ananiadou, S., Pyysalo, S., Tsujii, J., Kell, D.B., 2010. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology* 28, 381–390. doi:10.1016/j.tibtech.2010.04.005.
- Aronson, A.R., Lang, F.M., 2010. An overview of MetaMap: Historical perspective and recent advances. *Journal of the American Medical Informatics Association* 17, 229–236.
- Bada, M., Eckert, M., Evans, D., *et al.*, 2012. Concept annotation in the CRAFT corpus. *BMC Bioinformatics* 13, 161.
- Bates, M., 1995. Models of natural language understanding. *Proceedings of the National Academy of Sciences* 92, 9977–9982.
- Bethard, S., Ogren, P., Becker, L., 2014. ClearTK 2.0: Design patterns for machine learning in UIMA. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, European Language Resources Association (ELRA), Reykjavik, Iceland. pp. 3289–3293. Available at: http://www.lrec-conf.org/proceedings/lrec2014/pdf/218_Paper.pdf.
- Bethard, S., Savova, G., Chen, W.T., *et al.*, 2016. Semeval-2016 task 12: Clinical tempeval. *Proceedings of SemEval*, 1052–1062.
- Bird, S., Klein, E., Loper, E., 2009. *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*. "O'Reilly Media, Inc."
- Björne, J., Heimonen, J., Ginter, F., *et al.*, 2011. Extracting contextualized complex biological events with rich graph-based feature sets. *Computational Intelligence* 27, 541–557.
- Blei, D.M., 2012. Probabilistic topic models. *Communications of the ACM* 55, 77–84.
- Buchanan, G., Loizides, F., 2007. Investigating document triage on paper and electronic media. *Research and Advanced Technology for Digital Libraries*. 416–427.
- Bunescu, R.C., Pasca, M., 2006. Using encyclopedic knowledge for named entity disambiguation. *Eacl*. 9–16.
- Campos, D., Matos, S., Oliveira, J.L., 2015. A document processing pipeline for annotating chemical entities in scientific documents. *Journal of Cheminformatics* 7, S7.
- Campos, L., Pedro, V., Couto, F., 2017. Impact of translation on named-entity recognition in radiology texts. *Database*. 2017.
- Carletta, J., 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics* 22, 249–254.
- Carpenter, B., 2007. LingPipe for 99.99% recall of gene mentions. In: *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, pp. 307–309.
- Chaix, E., Dubreucq, B., Fathi, A., *et al.*, 2016. Overview of the regulatory network of plant seed development (seedev) task at the bionlp shared task 2016. In: *Proceedings of the 4th BioNLP shared task workshop*. Berlin: Association for Computational Linguistics, pp. 1–11.
- Chou, C.H., Chang, N.W., Shrestha, S., *et al.*, 2016. miRTarBase 2016: Updates to the experimentally validated miRNA-target interactions database. *Nucleic Acids Research* 44, D239–D247.
- Cohen, K.B., Hunter, L., 2004. Natural language processing and systems biology. In: Dubitzky, W., Azuaje, F. (Eds.), *Artificial Intelligence Methods and Tools for Systems Biology*. Springer, pp. 147–173.
- Couto, F., Campos, L., Lamurias, A., 2017. MER: A Minimal Named-Entity Recognition Tagger and Annotation Server. *BioCreative V.5 Challenge Evaluation*.
- Couto, F.M., Pinto, H.S., 2013. The next generation of similarity measures that fully explore the semantics in biomedical ontologies. *Journal of Bioinformatics and Computational Biology* 11, 1371001.
- Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K., 2013. Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLOS Computational Biology* 9, e1002854.
- DiGiacomo, R.A., Kremer, J.M., Shah, D.M., 1989. Fish-oil dietary supplementation in patients with Raynaud's phenomenon: A double-blind, controlled, prospective study. *The American Journal of Medicine* 86, 158–164.
- Elhadad, N., Pradhan, S., Chapman, W., Manandhar, S., Savova, G., 2015. Semeval-2015 task 14: Analysis of clinical text. In: *Proceedings of Workshop on Semantic Evaluation*. Association for Computational Linguistics, pp. 303–310.
- Farkas, R., Vincze, V., Móra, G., Csirik, J., Szarvas, G., 2010. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, Association for Computational Linguistics, pp. 1–12.
- Friedman, C., Kra, P., Rzhetsky, A., 2002. Two biomedical sublanguages: A description based on the theories of Zellig Harris. *Journal of Biomedical Informatics* 35, 222–235.
- Giuliano, C., Lavelli, A., Romano, L., 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. *EACL, Citeseer*. 401–408.
- Hakala, K., Van Landeghem, S., Salakoski, T., Van de Peer, Y., Ginter, F., 2013. EVEX in ST'13: Application of a large-scale text mining resource to event extraction and network construction. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*, Association for Computational Linguistics, pp. 26–34.
- Hearst, M.A., 1999. Untangling text data mining. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, Association for Computational Linguistics, pp. 3–10.
- Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., Declerck, T., 2013. The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions. *Journal of Biomedical Informatics* 46, 914–920.
- Hersh, W., Voorhees, E., 2009. TREC genomics special issue overview. *Information Retrieval* 12, 1–15.
- Hersh, W.R., Bhupatiraju, R.T., 2003. TREC genomics track overview. *TREC*. pp. 14–23.
- Hirschman, L., Yeh, A., Blaschke, C., Valencia, A., 2005. Overview of BioCreAtive: Critical assessment of information extraction for biology. *BMC Bioinformatics* 6, S1.
- Huang, C.C., Lu, Z., 2016. Community challenges in biomedical text mining over 10 years: Success, failure and the future. *Briefings in Bioinformatics* 17, 132–144. doi:10.1093/bib/bbv024.
- Kelly, L., Goeriot, L., Suominen, H., *et al.*, 2014. Overview of the share/clef ehealth evaluation lab 2014. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer. pp. 172–191.
- Kim, J.D., Ohta, T., Tateisi, Y., Tsujii, J., 2003. GENIA corpus – A semantically annotated corpus for bio-text mining. *Bioinformatics* 19, i180–i182.
- Kim, J.D., Ohta, T., Tsuruoka, Y., Tateisi, Y., Collier, N., 2004. Introduction to the bio-entity recognition task at JNLPBA. In: *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, Association for Computational Linguistics, pp. 70–75.
- Krallinger, M., Rabal, O., Leitner, F., *et al.*, 2015a. The ChEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics* 7, S2.
- Krallinger, M., Rabal, O., Lourenco, A., *et al.*, 2015b. Overview of the ChEMD-NER patents task, in: *Proceedings of the fifth BioCreative challenge evaluation workshop*, pp. 63–75.
- Krithara, A., Nentidis, A., Paliouras, G., Kakadiaris, I., 2016. Results proceedings of the 4th edition of BioASQ challenge. In: *Fourth BioASQ Workshop at the Conference of the Association for Computational Linguistics*, pp. 1–7.
- Lamurias, A., Clarke, L., Couto, F., 2017. Extracting microRNA-gene relations from biomedical literature using distant supervision. *PLOS ONE*. 12. <https://doi.org/10.1371/journal.pone.0171929>, Available at: <http://www.ncbi.nlm.nih.gov/pubmed/28263989>.
- Leaman, R., Islamaj Doğan, R., Lu, Z., 2013. DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics* 29, 2909–2917.

- Leaman, R., Wei, C.H., Lu, Z., 2015. tmChem: A high performance approach for chemical named entity recognition and normalization. *Journal of Cheminformatics* 7, S3.
- Leaman, R., Gonzalez, G., 2008. BANNER: An executable survey of advances in biomedical named entity recognition. In: *Pacific Symposium on Biocomputing*, pp. 652–663.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436.
- Lee, S., Kim, D., Lee, K., *et al.*, 2016. BEST: Next-generation biomedical entity search tool for knowledge discovery from biomedical literature. *PLOS ONE* 11, e0164680.
- Lever, J., Jones, S.J., 2016. VERSE: Event and relation extraction in the BioNLP 2016 shared task. *ACL 2016*, 42.
- Liu, Y., Liang, Y., Wishart, D., 2015. PolySearch2: A significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Research* 43, W535–W542.
- Lobo, M., Lamurias, A., Couto, F.M., 2017. Identifying human phenotype terms by combining machine learning and validation rules. *BioMed Research International*. 2017.
- Lopez, V., Pasin, M., Motta, E., 2005. Aqualog: An ontology-portable question answering system for the semantic web. In: *European Semantic Web Conference*, Springer, pp. 546–562.
- Lourenco, A., Carreira, R., Carneiro, S., *et al.*, 2009. @ note: A workbench for biomedical text mining. *Journal of Biomedical Informatics* 42, 710–720.
- Mallory, E.K., Zhang, C., Re, C., Altman, R.B., 2016. Large-scale extraction of gene interactions from full-text literature using DeepDive. *Bioinformatics* 32, 106–113.
- Manning, C.D., Schütze, H., *et al.*, 1999. *Foundations of Statistical Natural Language Processing*, 999. MIT Press.
- Manning, C.D., Surdeanu, M., Bauer, J., *et al.*, 2014. The Stanford CoreNLP natural language processing toolkit. *Association for Computational Linguistics (ACL) System Demonstrations*, 55–60. Available at: <http://www.aclweb.org/anthology/P14/P14-5010>.
- Miwa, M., Pyysalo, S., Ohta, T., Ananiadou, S., 2013. Wide coverage biomedical event extraction using multiple partially overlapping corpora. *BMC Bioinformatics* 14, 175.
- Miwa, M., Bansal, M., 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 10. doi:10.18653/v1/P16-1105, arXiv:1601.0770.
- Miyao, Y., Ohta, T., Masuda, K., *et al.*, 2006. Semantic retrieval for the accurate identification of relational concepts in massive textbases. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1017–1024.
- Müller, H.M.M., Kenny, E.E., Sternberg, P.W., 2004. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLOS Biology* 2, e309. doi:10.1371/journal.pbio.0020309.
- Nadeau, D., Sekine, S., 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes* 30, 3–26.
- Nunes, T., Campos, D., Matos, S., Oliveira, J.L., 2013. BeCAS: Biomedical concept recognition services and visualization. *Bioinformatics* 29, 1915–1916.
- Ohta, T., Pyysalo, S., Tsujii, J., 2011. Overview of the epigenetics and post-translational modifications (EPI) task of BioNLP shared task 2011. In: *Proceedings of the BioNLP Shared Task 2011 Workshop*, Association for Computational Linguistics, pp. 16–25.
- Okazaki, N., Ananiadou, S., 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics* 22, 3089–3095.
- Pershina, M., He, Y., Grishman, R., 2015. Personalized page rank for named entity disambiguation. In: *HLT-NAACL*, pp. 238–243.
- Pysalo, S., Ohta, T., Miwa, M., *et al.*, 2012. Event extraction across multiple levels of biological organization. *Bioinformatics* 28, i575–i581.
- Pysalo, S., Ohta, T., Rak, R., *et al.*, 2015. Overview of the cancer genetics and pathway curation tasks of bioNLP shared task 2013. *BMC Bioinformatics* 16, S2.
- Pysalo, S., Ginter, F., Moen, H., Salakoski, T., Ananiadou, S., 2013. Distributional semantics resources for biomedical text processing. In: *Proceedings of Languages in Biology and Medicine*, LBM.
- Rebholz-Schuhmann, D., Arregui, M., Gaudan, S., Kirsch, H., Jimeno, A., 2008. Text processing through web services: Calling Whatizit. *Bioinformatics* 24, 296–298.
- Ren, K., Lai, A.M., Mukhopadhyay, A., *et al.*, 2014. Effectively processing medical term queries on the UMLS metathesaurus by layered dynamic programming. *BMC Medical Genomics* 7, S11.
- Sætre, R., Yoshida, K., Yakushiji, A., *et al.*, 2007. AKANE system: Protein-protein interaction pairs in BioCreative2 challenge, PPI-IPS subtask. In: *Proceedings of the Second BioCreative Challenge Workshop*, Madrid, pp. 209–212.
- Savova, G.K., Masanz, J.J., Ogren, P.V., *et al.*, 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association: JAMIA* 17, 507–513. doi:10.1136/jamia.2009.001560.
- Segura Bedmar, I., Martinez, P., Herrero Zazo, M., 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In: *Proceedings of the Seventh International Workshop on Semantic Evaluation*, Association for Computational Linguistics.
- Segura-Bedmar, I., Martinez, P., de Pablo-Sánchez, C., 2011. Using a shallow linguistic kernel for drug-drug interaction extraction. *Journal of Biomedical Informatics* 44, 789–804.
- Segura-Bedmar, I., Martinez, P., Herrero-Zazo, M., 2014. Lessons learnt from the DDIEExtraction-2013 shared task. *Journal of Biomedical Informatics* 51, 152–164.
- Settles, B., 2005. ABNER: An open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics* 21, 3191–3192.
- Smith, L.H., Tanabe, L., Rindflesch, T., Wilbur, W.J., 2005. MedTag: A collection of biomedical annotations. In: *Proceedings of the ACL-ISMB workshop on linking biological literature, ontologies and databases: Mining biological semantics*, Association for Computational Linguistics, pp. 32–37.
- Stenetorp, P., Pyysalo, S., Tsujii, J., 2011. SimSem: Fast approximate string matching in relation to semantic category disambiguation. In: *Proceedings of BioNLP 2011 Workshop*, Association for Computational Linguistics, Portland, Oregon, USA, pp. 136–145. Available at: <http://www.aclweb.org/anthology/W11-0218>.
- Styler IV, W.F., Bethard, S., Finan, S., *et al.*, 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics* 2, 143–154.
- Sun, W., Rumshisky, A., Uzuner, O., 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association* 20, 806–813.
- Sutton, C., McCallum, A., 2006. An introduction to conditional random fields for relational learning. *Introduction to Statistical Relational Learning*, 93–128.
- Swanson, D.R., 1990. Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association* 78, 29.
- Szklarczyk, D., Santos, A., von Mering, C., *et al.*, 2016. STITCH 5: Augmenting protein-chemical interaction networks with tissue and affinity data. *Nucleic Acids Research* 44, D380–D384.
- Szklarczyk, D., Morris, J.H., Cook, H., *et al.*, 2017. The STRING database in 2017: Quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Research* 45, D362–D368.
- Tsuruoka, Y., McNaught, J., Ananiadou, S., 2008. Normalizing biomedical terms by minimizing ambiguity and variability. *BMC Bioinformatics* 9, S2.
- Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., Ananiadou, S., 2011. Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics* 27, 111–119. doi:10.1093/bioinformatics/btr214.
- Tsuruoka, Y., Tsujii, J., 2005. Bidirectional inference with the easiest-Proceedings of the first strategy for tagging sequence data. In: *Conference on human language technology and empirical methods in natural language processing*, Association for Computational Linguistics, pp. 467–474.
- Venkatesan, A., Kim, J.H., Talo, F., *et al.*, 2016. SciLite: A platform for displaying text-mined annotations as a means to link research articles with biological data. *Wellcome Open Research* 1, 25. doi:10.12688/wellcomeopenres.10210.1.
- Wei, C.H., Kao, H.Y., Lu, Z., 2013b. PubTator: A web-based text mining tool for assisting biocuration. *Nucleic Acids Research*. gkt441.
- Wei, C.H., Kao, H.Y., Lu, Z., 2015. GNormPlus: An integrative approach for tagging genes, gene families, and protein domains. *BioMed Research International*. 2015.
- Wei, C.H., Harris, B.R., Kao, H.Y., Lu, Z., 2013a. tmVar: A text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*. btt156.

- Winnenburg, R., Wächter, T., Plake, C., Doms, A., Schroeder, M., 2008. Facts from text: Can text mining help to scale-up high-quality manual curation of gene products with ontologies? *Briefings in Bioinformatics* 9, 466–478. doi:10.1093/bib/bbn043. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/19060303>.
- Yeh, A., Hirschman, L., Morgan, A., 2002. Background and overview for KDD cup 2002 task 1: Information extraction from biomedical articles. *ACM SIGKDD Explorations Newsletter* 4, 87–89.
- Yu, W., Gwinn, M., Clyne, M., Yesupriya, A., Khoury, M.J., 2008. A navigator for human genome epidemiology. *Nature genetics* 40, 124–125. doi:10.1038/ng0208-124.
- Zhang, C., 2015. DeepDive: A data management system for automatic knowledge base construction. PhD Thesis, The University of Wisconsin-Madison.

Relevant Websites

<https://www.epo.org/searching-for-patents.html>
European Patent Office.

https://www.nlm.nih.gov/bsd/index_stats_comp.html
NIH.

<https://spacy.io/>
spaCy.

Biographical Sketch

Andre Lamurias is a researcher at LaSIGE, currently enrolled in the BioSYS PhD programme at Universidade de Lisboa and holding a master's degree in Bioinformatics and Computational Biology from the same institution. His PhD thesis consists in developing text-mining approaches for disease network discovery and systems biology. More specifically, his research work is mainly focused on understanding how textual data from document repositories such as PubMed can be explored to improve our knowledge about biological systems.

Francisco M. Couto is currently an associate professor with habilitation and vice-president of the Department of Informatics of FCUL, member of coordination board of the master in Bioinformatics and Computational Biology, and a member of LASIGE coordinating the XLDB research group and the Biomedical Informatics research line. He graduated (2000) and has a master (2001) in Informatics and Computer Engineering from the IST. He concluded his doctorate (2006) in Informatics, specialization Bioinformatics, from the Universidade de Lisboa. He was on the faculty at IST from 1998 to 2001 and since 2001 at FCUL. He was an invited researcher at EBI, AFMB-CNRS, BioAlma during his doctoral studies. In 2003, he was one of the first researchers to study semantic similarity based on information content in biomedical ontologies. In 2006, he also developed one of the first systems to use semantic similarity to enhance the performance of text mining solutions. In 2011, he proposed the notion of disjunctive common ancestors. In 2013, he participated in the development one of the first similarity measure to exploit and demonstrate the usefulness of the description logic axioms in a biomedical ontology.

Multilayer Perceptrons

Leonardo Vanneschi and Mauro Castelli, NOVA IMS, Universidade Nova de Lisboa, Lisboa, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

Artificial Neural Networks are computational techniques that belong to the field of Machine Learning (Mitchell, 1997; Kelleher et al., 2015; Gabriel, 2016). The aim of Artificial Neural Networks is to realize a very simplified model of the human brain. In this way, Artificial Neural Networks try to learn tasks (to solve problems) mimicking the behavior of brain. The brain is composed by a large set of elements, specialized cells called *neurons*. Each single neuron is a very simple entity, but the power of the brain is given by the fact that neurons are numerous and strongly interconnected between them. The brain learns because neurons are able to communicate between each other. A picture of a biological neuron is shown in Fig. 1.

In analogy with the human brain, Artificial Neural Networks are computational methods that use a large set of elementary computational units, called themselves (artificial) *neurons*. And Artificial Neural Networks due their power to the numerous interconnections between neurons. Each neuron is able to only perform very simple tasks, and Artificial Neural Networks are able to perform complex calculations because they are typically composed by many artificial neurons, strongly interconnected between each other and communicating with each other. Before studying complex Artificial Neural Networks, that are able to solve large scale real-life problems, we must understand how the *single neurons* and *simple networks* work. For this reason, we begin the study of Artificial Neural Networks (from now on simply Neural Networks) with a very simple kind of network called *Perceptron*.

Perceptron

Perceptron (Rosenblatt, 1958; Demuth et al., 2014; Rashid, 2016) is a simple Neural Network, that can be composed by one single neuron, or several neurons arranged in a *single layer*. In the continuation of this document, we will study the important concept of “layers” in Neural Network and we will be able to understand why Neural Networks that are composed by more than one layer may be more powerful than Neural Networks composed by a single layer, like the Perceptron. Another limitation of the Perceptron, compared to some other kinds of networks, is that in the Perceptron the flow of the information is uni-directional (from input to outputs). For these reason, the Perceptron is also called a *Feed-Forward Neural Network*.

The Perceptron is composed by one or more output neurons, each of which is connected to several input units by means of connections that are characterized by weights. In analogy with Biology, these connections are called *synapsis*.

Single Layer Perceptron

Let us begin with the simplest possible Perceptron Neural Net: the one composed by just one neuron. The structure of a Single Neuron Perceptron can be represented as in Fig. 2.

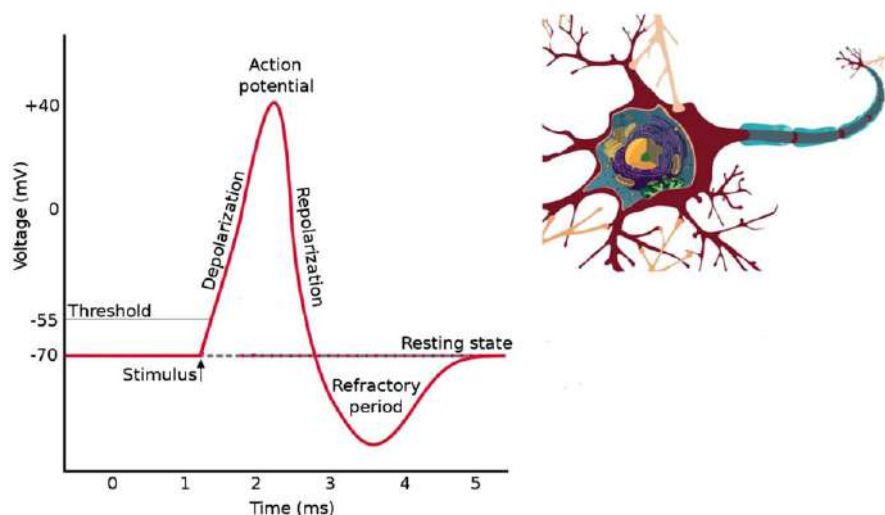


Fig. 1 Illustration of a biological neuron and its synapsis.

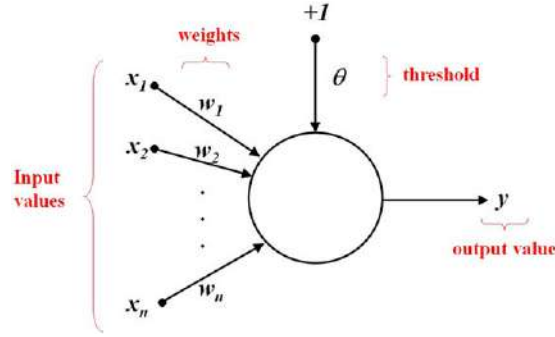


Fig. 2 Single Neuron Perceptron.

The output is calculated as follows:

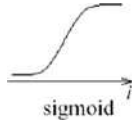
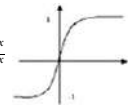
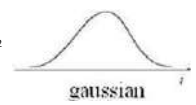
$$y = f\left(\sum_{i=1}^n w_i x_i + \theta\right)$$

The function f is called *activation function*. One of the most commonly used activation functions is the so called *threshold* (or “step”) function:

$$f(s) = \begin{cases} 1 & \text{se } s > 0 \\ -1 & \text{altrimenti} \end{cases}$$

This activation function is particularly useful when we want to solve classification problems. For instance, the Perceptron composed by one neuron, with this activation function, can be used to classify a set of vectors $\bar{x}$ in one among two possible classes C_1 and C_2 (binary classification). In such a situation, the single neuron Perceptron classifies the vector $\bar{x}$ into the class C_1 if its output is equal to 1 and into the class C_2 if its output is equal to -1 .

As an alternative to the threshold activation function, there are other possible activation functions that are commonly used:

- Logistic o sigmoidal: $f(x) = \frac{1}{1+e^{-ax}}$ 
sigmoid
- Hyperbolic tangent: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ 
- Gaussian: $f(x) = e^{-x^2}$ 
gaussian

Neural Networks are characterized by a learning (or training) phase, in which the network “learns” to solve the given problem modifying, if needed, the weights of its connections. Once the learning phase is complete, the Neural Network is ready to be used on new data (generalization phase). During the training phase, the single neuron Perceptron receives in input k vectors of data $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k\}$ and the class (C_1 or C_2) to which they belong (i.e., the class in which they must be classified). These data form the training set. During the learning, the Perceptron modifies the weights of its synopsis (with an algorithm, called *learning rule*) in such a way to be able to classify, if possible, all the vectors $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$ into the correct class (C_1 or C_2). This is supervised learning, given that the target is known for the data of the training set, and Perceptron is a supervised Neural Network.

The algorithm used by Perceptron to modify the weights (in other words, to learn) is the following.

Perceptron learning rule

1. Initialize the connections with a set of weights generated at random.
2. Select an input vector $\bar{x}$ from the training set. Let y be the output value returned by Perceptron for this input value and d the correct answer (target), that is associated to $\bar{x}$ in the dataset.
3. If $y \neq d$ (i.e. the output calculated by the Perceptron is wrong), then modify the weights of all the connections w_i as follows:

$$\text{if } y > d \quad w_i := w_i - \eta x_i$$

$$\text{if } y < d \quad w_i := w_i + \eta x_i$$

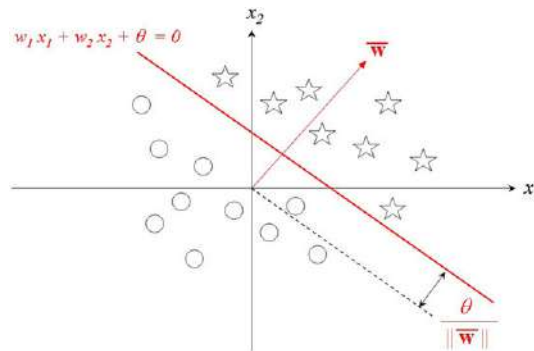


Fig. 3 A straight line separating points belonging to two classes.

4. Go to point 2.

The algorithm terminates when all the vectors in the training set are classified in a correct way or after a given prefixed number of iterations.

Remarks:

- Also the weight of the threshold θ must be modified together with all the other weights of the other connections. Thus, θ from now on will be treated like any other connection (in some cases we will call w_0 its weight) with the particularity that the input of this connection is always equal to 1.
- The parameter η is called *learning rate* and can be a constant or it can change at each iteration. In any case, it has always a positive value.

Let us consider the most possible simple case: the case in which the single neuron Perceptron has only two input units. In this case, once the learning phase is terminated, the values of the weights w_1 and w_2 and of the threshold θ will have been determined. In this situation, it is possible to identify a straight line (it would be a hyperplane in the general case, i.e., when the number of inputs is larger than two):

$$\text{or:} \quad w_1x_1 + w_2x_2 + \theta = 0$$

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{\theta}{w_2}$$

This straight line allows us to graphically *separate* the two classes C_1 and C_2 .

This straight line has the following properties:

- All the points that belong to class C_1 are “above” the straight line and all the ones that belong to C_2 are “below” (or viceversa).
- The weights vector $\bar{w} = [w_1, w_2]$ is perpendicular to the straight line.
- θ allows us to calculate the distance of the straight line to the origin.

These concepts are shown graphically in [Fig. 3](#) and will be clarified by the following example.

Example of Application of the Perceptron Learning Rule

This example should help us clarify the functioning of the algorithm and many of the concepts discussed so far. Let us consider the following “toy” training set:

(x_1, x_2)	d
$(-2, 6)$	1
$(6, 2)$	1
$(-4, 1)$	-1
$(-10, 4)$	-1

In such a situation, the Perceptron must classify $(-2, 6)$ e $(6, 2)$ into a class and $(-4, 1)$ and $(-10, 4)$ into the other. The first step is the weights initialization. It is done in a random way. Let us assume, as an example, that the generator of random numbers of our programming language has generated the following values for the initial weights:

$$w_1 = 1, w_2 = 2, w_0 = -12$$

Furthermore, let us suppose that we keep the learning rate constant for all the execution of the learning rule and let us consider a threshold (“step”) activation function. Let us draw the points of the training set (using two different symbols for the points that

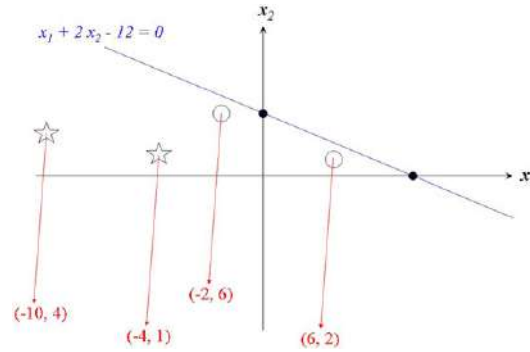


Fig. 4 The points of the example, and the straight line obtained using the initial weights.

must be classified into the two different classes) and the straight line:

$$x_1 + 2x_2 - 12 = 0$$

This graphical representation is shown in **Fig. 4**. From this graphical representation, it is clear that the straight line *does not* separate the points $(-2, 6)$ and $(6, 2)$ from $(-4, 1)$ and $(-10, 4)$, thus it is necessary to modify the weights (and in this way, also the straight line will be modified).

Let us begin the execution of the Perceptron learning rule in order to modify the weights. Let us consider, for instance, the first point in the training set: $(-2, 6)$ and let us calculate the output of the Perceptron for this point:

$$y = f(-2 + 12 - 12) = f(-2) = -1$$

The classification that the Perceptron does of this point is *not* correct: the result is different from the target value that appears in the training set in correspondence with the observation we are considering. Thus, it is necessary to modify the weights. The Perceptron learning rule does those modifications like this:

$$w_0 = w_0 + \eta x_0 = -12 + (1 \cdot 1) = -11$$

$$w_1 = w_1 + \eta x_1 = 1 + 1 \cdot (-2) = -1$$

$$w_2 = w_2 + \eta x_2 = 2 + (1 \cdot 6) = 8$$

Now let us consider, *for instance*, the second point in the training set and let us calculate the output of the Perceptron *with the new weights*. The point is $(6, 2)$ and the output calculated by the Perceptron is:

$$y = f(-6 + 16 - 11) = f(-1) = -1$$

Also in this case the classification is *not* correct (the output calculated by the Perceptron is different from the target that is the training set associated with the point $(6, 2)$). Given that the classification is not correct, it is necessary to once again modify the weights:

$$w_0 = w_0 + \eta x_0 = -11 + (1 \cdot 1) = -10$$

$$w_1 = w_1 + \eta x_1 = -1 + 1 \cdot 6 = 5$$

$$w_2 = w_2 + \eta x_2 = 8 + (1 \cdot 2) = 10$$

Let us now consider, *for instance*, the third point in the training set: $(-4, 1)$. The output calculated by the Perceptron is:

$$y = f(-20 + 10 - 10) = f(-20) = -1$$

the classification in this case is correct (the output calculated by the Perceptron is equal to the correct value that is in the training set associated with the point $(-4, 1)$). So, the weights do not have to be modified.

Now let us consider the fourth point in the training set: $(-10, 4)$. The output calculated by the Perceptron for this point is:

$$y = f(-50 + 40 - 10) = f(0) = -1$$

Also in this case the classification is correct, so the weights do not have to be modified. At this point, we must iterate once again considering all the points in the training set since the last modification of the weights, because the termination condition of the learning algorithm says that the algorithm terminates only when the Perceptron correctly classifies all the training instances.

Let us consider, *for instance*, the first point in the training set: $(-2, 6)$. We have:

$$y = f(-10 + 60 - 10) = f(40) = 1$$

The classification is correct, thus we do not modify the weights. Now let us consider the second point in the training set: $(6, 2)$. We have:

$$y = f(30 + 20 - 10) = f(40) = 1$$

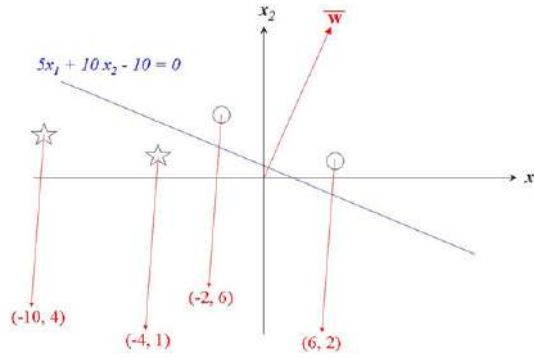


Fig. 5 The points of the example, and the straight line obtained using the weights obtained at termination of the execution of the Perceptron learning rule.

Also in this case the classification is correct, and so we do not modify the weights. The third and the fourth vector of the training set have already been tested with these weights and they have returned a correct classification. So now the algorithm terminates.

Let us now draw the points of the training set and the straight line with the new weights vector:

$$5x_1 + 10x_2 - 10 = 0$$

This graphical representation is shown in Fig. 5. It is clear that now the straight line correctly separates the points! Furthermore, it is possible to observe that the weights vector $(w_1, w_2) = (5, 10)$ is perpendicular to the straight line. In fact, the straight line can be expressed as follows:

$$x_2 = -\frac{1}{2}x_1 + 1$$

and thus its slope is: $m_1 = -\frac{1}{2}$. While the straight line that extends the weights vector is a straight line that passes through the origin and the point $(5, 10)$, so we can write this straight line as:

$$\begin{aligned} \frac{x_1 - 5}{-5} &= \frac{x_2 - 10}{-10} \\ -\frac{x_1}{5} + 1 &= -\frac{x_2}{10} + 1 \\ x_2 &= 2x_1 \end{aligned}$$

Its slope is: $m_2 = 2$. The condition for two straight lines with slopes m_1 and m_2 to be perpendicular is that: $m_2 = -\frac{1}{m_1}$. So, in our case the two straight lines are perpendicular. Finally, let us calculate the distance to the origin of the straight line:

$$5x_1 + 10x_2 - 10 = 0$$

The formula of the distance of a point (x_1, y_1) to a straight line:

$$w_1x + w_2y + w_0 = 0$$

is given by:

$$d = \frac{|w_1x_1 + w_2y_1 + w_0|}{\sqrt{w_1^2 + w_2^2}}$$

In the case in which (x_1, y_1) is the origin, the previous formula becomes:

$$d = \frac{|w_0|}{\sqrt{w_1^2 + w_2^2}}$$

or:

$$d = \frac{|\theta|}{\|\bar{w}\|}$$

The straight line:

$$w_1x_1 + w_2x_2 + w_0 = 0$$

can always be written in such a way that:

$$\|\bar{w}\| = \sqrt{w_1^2 + w_2^2} = 1$$

To obtain this, it is enough to divide the parameters of the straight line by a given constant k . For instance in the case of the previous straight line:

$$5x_1 + 10x_2 - 10 = 0$$

it can be transformed into:

$$x_1 + 2x_2 - 2 = 0$$

to calculate the constant k all we have to do is:

$$\sqrt{\frac{1}{k^2} + \frac{4}{k^2}} = 1 \Rightarrow \frac{\sqrt{5}}{k} = 1 \Rightarrow k = \sqrt{5}$$

So, if we write the straight line in the following form:

$$\frac{1}{\sqrt{5}} x_1 + \frac{2}{\sqrt{5}} x_2 - \frac{2}{\sqrt{5}} = 0$$

we have:

$$\|\bar{\mathbf{w}}\| = 1$$

This means that $\|\bar{\mathbf{w}}\|$ can always be considered equal to 1 *without loss of generality*.

Perceptron convergence theorem

If a weight vector $\bar{\mathbf{w}}^*$ that allows to obtain $y = d$ for all training instances exists, then the Perceptron converges to a solution (that can be equal to $\bar{\mathbf{w}}^*$ or not) in a finite number of steps *independently from the initial choice of the weights*.

If such a weight vector exists, then the classes are *linearly separable*.

Multiple Class Classification

So far, we have considered problems in which data have to be partitioned into *two* classes (*binary classification*). But what happens if data have to be partitioned in *more* than two classes? (*multiclass classification*). Given a problem in which data must be classified into more than two classes, more than two neurons have to be combined into a network. The Perceptron, by its very definition, can have only one layer of neurons, so, for instance, a Perceptron with m neurons will have a structure like the one shown in Fig. 6.

This Perceptron can separate data into 2^m distinct classes, depending on its binary output. In other words, $y_1, y_2, \dots, y_m$ can be interpreted as a binary number, which "encodes" one of the labels of the classes in which data have to be partitioned. During the learning phase, each single neuron is trained independently with the Perceptron Learning Rule that we have seen so far. So, at the end of this phase, we find a weight vector for each neuron. Using this vector, it is possible (in general) to define a *hyperplane*. In case the network has two inputs, the weights of each neuron identify a straight line. These straight lines can be represented on a Cartesian plane and, at the end of the learning phase, if this phase was successful, they separate the points that belong to the different classes, as in the example shown in Fig. 7. Given that all the separating functions are linear, also in the case of (single layer) Perceptron with more than one neuron, the Perceptron is able to correctly solve a classification problem into m different classes $C_1, C_2, \dots, C_m$ if and only if these classes are *linearly separable*.

Non Linearly Separable Problems. An Example

One of the most well-known and commonly diffused non linearly separable problem is the XOR problem. The XOR function is a boolean function that has two input arguments:

$$y = x_1 \text{ XOR } x_2$$

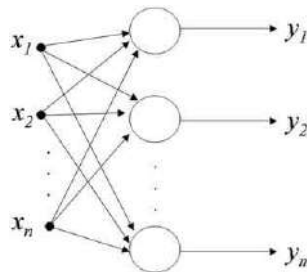


Fig. 6 Single-layer Perceptron with m output neurons.

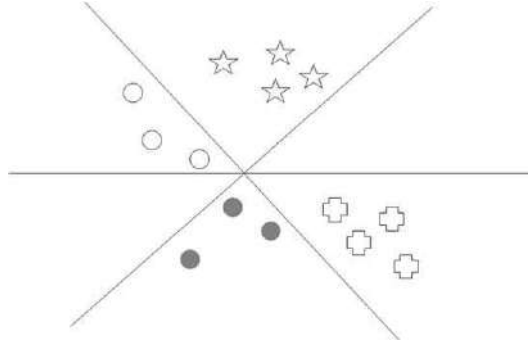


Fig. 7 Graphical representation of two straight lines able to separate points from four different classes.

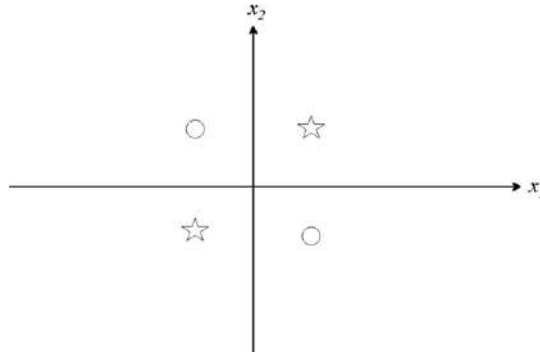


Fig. 8 Graphical representation of the XOR problem.

and whose value is expressed by the following truth table:

x_1	x_2	y
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

On a Cartesian plane, the XOR function can be represented as in [Fig. 8](#). Clearly, no set of straight lines can exist that can be able to separate the two classes.

Let us proof (by absurd) that the Perceptron is not able to correctly classify the XOR problem. Given that the problem consists into a classification of data which are vectors of cardinality equal to two into two classes, we can consider the case of the Perceptron with two inputs and one single neuron. Let us assume by absurd that this Perceptron is able to correctly classify the XOR problem. Then, we admit that a vector of weights $[w_0, w_1, w_2]$ exists such that the following system of disequations holds:

$$-w_1 - w_2 + w_0 \leq 0 \quad (1)$$

$$w_1 - w_2 + w_0 > 0 \quad (2)$$

$$-w_1 + w_2 + w_0 > 0 \quad (3)$$

$$w_1 + w_2 + w_0 \leq 0 \quad (4)$$

Adding (2) to (3) in the previous system, we obtain:

$$2w_0 > 0 \quad \text{i.e.} \quad w_0 > 0 \quad (5)$$

Replacing (5) in (1), we obtain: $-w_1 - w_2 \leq -w_0$ i.e. $-w_1 - w_2$ is smaller or equal to a quantity that is *strictly negative*, and for this reason it is also strictly negative:

$$-w_1 - w_2 < 0 \quad (6)$$

Now, we replace (5) in (4) and we obtain:

$$w_1 + w_2 \leq -w_0$$

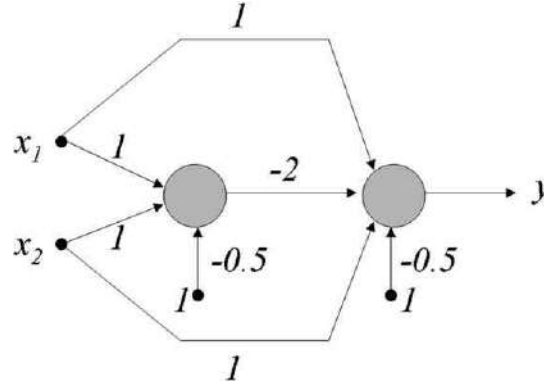


Fig. 9 A two layers Neural Network able to correctly classify all the instances of the XOR problem.

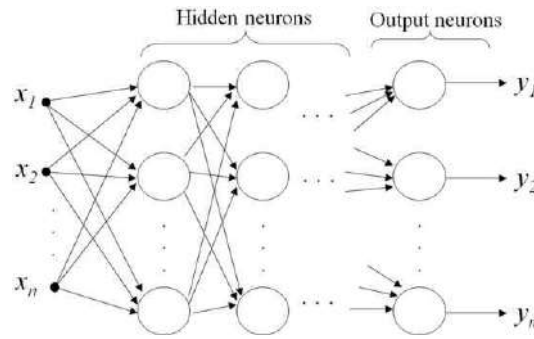


Fig. 10 Feed-forward, multi-layer Neural Network.

In other words, we have obtained that $w_1 + w_2$ is minor or equal to a strictly negative quantity, thus:

$$w_1 + w_2 < 0 \quad \rightarrow \quad -w_1 - w_2 > 0 \quad (7)$$

Inequalities (6) and (7) create a contradiction that allows us to terminate the proof.

We observe that a Neural Network composed by *two layers* of neurons (one neuron for each layer) solves the XOR problem. This network is represented in [Fig. 9](#).

In order to prove that the network shown in [Fig. 9](#) solves the XOR problem, let g the output of the left-most neuron, i.e.:

$$g = f(x_1 + x_2 - 0.5)$$

where f is the threshold (or step) activation function. Then, the output of all the network (that corresponds to the output of the right-most neuron) is:

$$y = f(x_1 + x_2 - 2g - 0.5)$$

For the different input data, we have:

$$\begin{aligned} (x_1, x_2) = (-1, -1) &\rightarrow g = f(-2.5) = -1 \rightarrow y = f(-0.5) = -1 \\ (x_1, x_2) = (-1, 1) &\rightarrow g = f(-0.5) = -1 \rightarrow y = f(1.5) = 1 \\ (x_1, x_2) = (1, -1) &\rightarrow g = f(-0.5) = -1 \rightarrow y = f(1.5) = 1 \\ (x_1, x_2) = (1, 1) &\rightarrow g = f(1.5) = 1 \rightarrow y = f(-0.5) = -1 \end{aligned}$$

Non Linearly Separable Problems and Multi-Layer Neural Networks

In general, non linearly separable problems can be solved using *multi-layer* Neural Networks, i.e., Neural Networks in which *one or more neurons are not directly linked to the output of the network* (we call these neurons hidden neurons, and given that they are generally organized into layers, we call these layers hidden layers) ([Demuth et al., 2014](#); [Rashid, 2016](#); [Minsky and Papert, 1988](#)).

A possible structure of a multi-layer Neural Network is shown in [Fig. 10](#).

This network is called Feed-Forward Neural Network and its main characteristics are:

- The neurons in a given layer receive as input the outputs of the neurons of the previous level.

- There are no inter-level connections.
- All the possible intra-level connections exist between two any subsequent layers.

Typically, these Neural Networks are trained using the Backpropagation learning rule, that is the subject of the next article.

See also: Algorithms for Graph and Network Analysis: Graph Alignment. Artificial Intelligence and Machine Learning in Bioinformatics. Artificial Intelligence. Data Mining in Bioinformatics. Data-Information-Concept Continuum From a Text Mining Perspective. Gene Prioritization Tools. Knowledge and Reasoning. Machine Learning in Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Stochastic Methods for Global Optimization and Problem Solving. Text Mining Basics in Bioinformatics. Text Mining for Bioinformatics Using Biomedical Literature. The Challenge of Privacy in the Cloud

References

- Demuth, H.B., Beale, M.H., De Jess, O., Hagan, M.T., 2014. Neural Network Design, second ed. USA: Martin Hagan.
- Gabriel, J., 2016. Artificial Intelligence: Artificial Intelligence for Humans, first ed. USA: CreateSpace Independent Publishing Platform.
- Kelleher, J.D., Namee, B.M., D'Arcy, A., 2015. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. Cambridge, MA: The MIT Press.
- Minsky, M.L., Papert, S.A., 1988. Perceptrons: Expanded Edition. Cambridge, MA: MIT Press.
- Mitchell, T.M., 1997. Machine Learning, first ed. New York, NY: McGraw-Hill, Inc.
- Rashid, T., 2016. Make Your Own Neural Network, first ed. USA: CreateSpace Independent Publishing Platform.
- Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386–408.

Delta Rule and Backpropagation

Leonardo Vanneschi and Mauro Castelli, NOVA IMS, Universidade Nova de Lisboa, Lisboa, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

As previously discussed, single-layer Artificial Neural Networks are not appropriate to solve non-linearly separable problems. Solid theoretical results tell us that this kind of problem can more suitably be solved using *multi-layer* Artificial Neural Networks (Minsky and Papert, 1988), i.e. Artificial Neural Networks (or simply Neural Networks from now on) in which one or more neurons are not directly linked to the output of the network. In other words, these neurons have the property that their outputs, instead of being part of the output of the network, are used as inputs for other neurons. We call these neurons *hidden neurons*, and given that this kind of networks are generally organized into layers, we call the layers that contain hidden neurons *hidden layers*, in contraposition with the *output layer* that is formed by neurons (that we call *output neurons*) whose outputs form the output of the network. A possible structure of a multi-layer Neural Network is shown in Fig. 1.

A network with such a structure is called *Feed-Forward Neural Network* and its main characteristics are:

- The neurons in a given layer receive as input the outputs of the neurons of the previous layer.
- There are no inter-layer connections.
- All the possible intra-layer connections exist between units of any two subsequent layers.

The motivation for the name we give to these networks (Feed-Forward Neural Networks) resides in the direction of the flux of information circulating in the network while the network itself is working: we can imagine that, in the graphical representation of Fig. 1, the information always goes from left to right. In other words, inputs $x_1, x_2, \dots, x_n$ are propagated *forward* into the network in order to produce the outputs $y_1, y_2, \dots, y_m$. This implies that Feed-Forward Neural Networks, by their very definition, do *not* contain cycles, or, which is equivalent, do not contain any connection from a neuron to the neuron itself, or from a neuron to any neuron of a previous layer (Neural Networks with cyclic connections exist. They are called Cyclic Neural Networks and they have several interesting typical practical applications, but their discussion is outside the scope of this article.).

Typically, feed-forward multi-layer Neural Networks are trained using the Backpropagation learning rule (Haykin, 1998), which is a generalization of another, older and simpler, learning rule called Delta Rule. The Delta Rule was originally defined for single-layer Neural Networks, or even Neural Networks composed by only one neuron. The Backpropagation can be seen as an extension of the Delta Rule to multi-layer Neural Networks. Under this perspective, in order to understand the functioning of the Backpropagation (which is the main objective of this paper), it is a necessary precondition to deeply understand the functioning of the Delta Rule. For this reason, in the next section we present the Delta Rule, and later in this document, we present the Backpropagation. Then, the paper is concluded by discussing the important issue of overfitting, and presenting ideas to appropriately set the parameters of feed-forward multi-layer Neural Networks in order to counteract overfitting.

Adaptive Linear Element, and Delta Rule

Adaptive Linear Element (ADALINE) is a Neural Network that has an architecture very similar to the one of the Perceptron (Rosenblatt, 1958). The learning algorithm used by ADALINE, called *Delta Rule* (Haykin, 1998), can be seen as a generalization and/or variation of the Perceptron learning rule. The difference between the Delta Rule and the Perceptron learning rule consists in

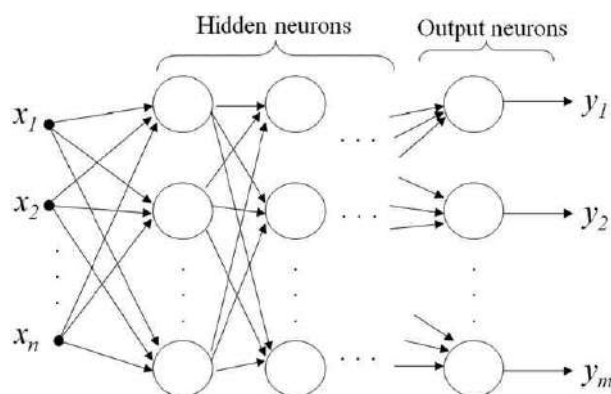


Fig. 1 Feed-forward, multi-layer Neural Network.

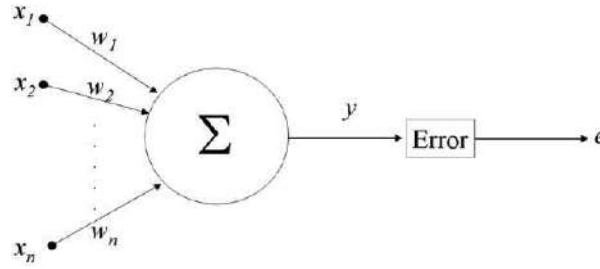


Fig. 2 Architectural structure of ADALINE.

the way the output is managed. The Perceptron uses as output the result of an activation function (for instance the threshold activation function) to learn. The Delta Rule uses the result of the summation of the inputs (weighted using the weights of the connections or synapsis), to which it can, or can not, be applied an activation function:

$$\gamma = \sum_{i=1}^m w_i x_i + \theta \quad \text{or} \quad \gamma = f\left(\sum_{i=1}^m w_i x_i + \theta\right)$$

In the continuation of this document, the activation function will always be used, but, for reasons that will be clearer later, when an activation function is used, it is better to use an activation function that does not necessarily return either 1 or -1; more specifically, it is generally better if it returns continuous values. The main difference between the Perceptron learning rule and the Delta Rule is that the Delta Rule compares this calculated output to the desired output, or target, by using an *error*, which is usually quantified by means of a distance between calculated outputs and targets (any distance metric can be used for this aim). Contrarily to the functioning of the Perceptron learning rule, the Delta Rule uses this error as the basis for learning. Given the importance of the concept of error between outputs and targets, the architecture of ADALINE is often represented as in [Fig. 2](#). The reader is invited to compare this architecture to the one of the Perceptron ([Rosenblatt, 1958](#)), and remark the presence of an error function applied to the output γ of the neuron (error function that is not present in the architecture of the Perceptron).

Let us assume that we want to map a set of input vectors $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ into a set of known desired output values $\{d_1, d_2, \dots, d_n\}$ (the target values, which are known, given that we are in presence of supervised learning (Unsupervised learning Neural Networks also exist and are much used in practice, but a discussion of these Neural Networks is outside the scope of this article.)). Let $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$ be the values respectively returned by ADALINE for each one of these input vectors. The most common error functions (also called *cost functions* in some references) are:

$$E = \frac{1}{n} \sum_{i=1}^n (d_i - \gamma_i)^2$$

and in this case the error is commonly called *mean square error*; or:

$$E = \sum_{i=1}^n |d_i - \gamma_i|$$

and in this case the error is commonly called *absolute error*. Many other error measures can be used besides these ones, but only these ones will be discussed in this document, since they are by far the most used ones.

The Delta Rule looks for the vector of weights $\bar{w}$ that *minimizes* the error function, using a method called *gradient descent* ([Haykin, 1998](#)). This method will be defined shortly. So far, it is important to notice that the *learning* of the Neural Network becomes an *optimization problem*, where the cost function, to be minimized, is the error function E , and the different possible solutions for this optimization problem are the different possible existing vectors of weights $\bar{w}$. The graphical representation of function E can be seen as a multidimensional surface and the objective is to find the point (i.e. the vector of weights) in which this surface has its minimum value. Under this perspective, the dynamics of the Delta Rule can be seen as a navigation strategy in this surface; a strategy that, typically starting from a random point, is able to move towards the minimum.

Before studying the functioning of the Delta Rule, the reader is invited to become aware of the existence of the following theorem, one of the consequences of which represents one of the most important limitations of the Delta Rule itself (and the motivation for the necessity of extending the Delta Rule to define the Backpropagation).

Theorem

If the problem is *linearly separable*, then the surface of the E function is *unimodal*, i.e. it contains *only one global minimum* and *no other local minima*.

The objective of the Delta Rule is simply modifying the weights in an iterative way, in order to guarantee that the error E decreases at each one of these steps. Thus, it is easy to understand that, if the problem is not linearly separable and thus the surface of the error function is multimodal, it is possible that ADALINE gets trapped into a local minimum, different from the global one.

On the other hand, for linearly separable problems, as for the Perceptron, also ADALINE is in general an appropriate method to find an optimal solution and thus solve the problem in a satisfactory way.

To decrease the error E at each step, the Delta Rule uses the gradient descent method. In other terms, the Delta Rule functions in a stepwise fashion, by at each step modifying the weights by a quantity $\Delta \bar{w}$ which is *proportional to the derivative* of function E (with a negative proportionality constant):

$$\Delta \bar{w} = -\eta \frac{\partial E}{\partial \bar{w}}$$

We say that, in this way, the weights are modified *in the negative direction of the gradient of E* . Of course, when the minimum of the E function is reached, its derivative is equal to zero, and thus weights are no more modified and the algorithm terminates.

Just as a matter of case study, let us focus in this section on the absolute error (we will focus on the mean square error later). The objective is to minimize:

$$E = \sum_{i=1}^n |d_i - y_i|$$

where $[d_1, d_2, \dots, d_n]$ are the known target values, and $[y_1, y_2, \dots, y_n]$ are the outputs calculated by a neuron (each output is calculated, as previously explained, using a different input vector). Minimizing the error means to find the vector of weights $[w_1, w_2, \dots, w_m]$ for which E is minimum. Before studying the details of the functioning of the Delta Rule, let us establish the following terminology.

Terminology

- n = number of observations in the training set (so, there are n different target values and the summation in the definition of E ranges from 1 to n);
- m = number of dimensions of each one of the input vectors $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ contained in the training set;
- x_i^j = the i th input value of the j th observation (i.e., the i th coordinate of the $\bar{x}_j$ input vector). Remark that: i ranges from 1 to m , j ranges from 1 to n and x_i^j is one of the values that are used as input to the neuron;
- w_i = weight of the connection between the i th input unit and the neuron. Remark that here i ranges from 0 to m (where w_0 represents the weight of the threshold connection (Rosenblatt, 1958));
- $v_j = \sum_{i=0}^m w_i x_i^j$ = what is calculated by the neuron before the application of the activation function. Remark that, again, j ranges from 1 to n (the neuron performs one different calculation, leading to one different output, for each different observation in the training set);
- $y_j = f(v_j)$ = output of the neuron for the j th observation (where f is the activation function);
- $e_j = |d_j - y_j|$ = error of the neuron of the j th observation (where d_j is the correct value, or target, for the j th observation, which is known in the dataset). Again, both in this and in the previous point, j ranges from 1 to n .

According to the previous terminology, E can be written as:

$$E = \sum_{j=1}^n e_j$$

If we consider again the formula of the modification of the weights, but considering this time only one coordinate w_i of the weight vector:

$$\Delta w_i = -\eta \frac{\delta E}{\delta w_i} \quad (1)$$

then our objective can be presented as finding an alternative (and mathematically “convenient”) way of expressing the term $\frac{\delta E}{\delta w_i}$.

To obtain this alternative expression, we first decompose the term $\frac{\delta E}{\delta w_i}$ as follows:

$$\frac{\delta E}{\delta w_i} = \frac{\delta E}{\delta e_j} \cdot \frac{\delta e_j}{\delta y_j} \cdot \frac{\delta y_j}{\delta v_j} \cdot \frac{\delta v_j}{\delta w_i} \quad (2)$$

We now consider the expression on the right of Eq. (2) and “transform” each term one-by-one. Before proceeding, the reader is invited to have a look back at the previous terminology to understand the meaning of e_j , y_j and v_j that have been introduced in Eq. (2). Furthermore, the reader is invited to become aware of the following notation, that will be used in the continuation.

Notation

From now on, given any function

$$\phi(x)$$

we will indicate its derivative with respect to variable x as:

$$\phi'(x)dx$$

This notation is synonymous of $\frac{\delta\phi}{\delta x}$.

Furthermore, when the variable with respect to which we are derivating is “clear from the context”, sometimes we will simply use notation ϕ' to express the derivative of ϕ .

Let us first consider the term $\frac{\delta E}{\delta e_j}$ in Eq. (2). We have;

$$\frac{\delta E}{\delta e_j} = \left(\sum_{k=1}^n e_k \right)' de_j = (e_1 + e_2 + \dots + e_j + \dots + e_n)' de_j = 1$$

In fact, e_j is the variable with respect to which we are performing the derivative, and all other terms are constant.

Now, let us develop the term $\frac{\delta e_j}{\delta y_j}$ in Eq. (2).

$$\frac{\delta e_j}{\delta y_j} = (|d_j - y_j|)' dy_j$$

We know that:

$$|d_j - y_j| = \begin{cases} y_j - d_j & \text{if } d_j < y_j \\ d_j - y_j & \text{otherwise} \end{cases}$$

So:

$$(|d_j - y_j|)' dy_j = \begin{cases} 1, & \text{if } d_j < y_j \\ -1, & \text{otherwise} \end{cases}$$

Now, let us develop the term $\frac{\delta y_j}{\delta v_j}$ in Eq. (2). Given that: $y_j = f(v_j)$, we have:

$$\frac{\delta y_j}{\delta v_j} = f'(v_j) dv_j = f'_j$$

The notation f'_j is used for simplicity, omitting the variable with respect to which the derivative is performed.

Finally, let us develop the term $\frac{\delta v_j}{\delta w_i}$ in Eq. (2). We have:

$$\frac{\delta v_j}{\delta w_i} = \left(\sum_{k=0}^m w_k x_k^j \right)' dw_i = (w_1 x_1^j + w_2 x_2^j + \dots + w_i x_i^j + \dots + w_m x_m^j)' dw_i = x_i^j$$

At this point, we are able to join all the developed terms of Eq. (2), and obtain a transformed version of that equation:

$$\frac{\delta E}{\delta w_i} = \frac{\delta E}{\delta e_j} \cdot \frac{\delta e_j}{\delta y_j} \cdot \frac{\delta y_j}{\delta v_j} \cdot \frac{\delta v_j}{\delta w_i} = \begin{cases} f'_j x_i^j, & \text{if } y_j < d_j \\ -f'_j x_i^j, & \text{otherwise} \end{cases}$$

So, also Eq. (1), i.e. the formula to update the weights, can be rewritten as:

$$\Delta w_i = -\eta \frac{\delta E}{\delta w_i} = \begin{cases} \eta f'_j x_i^j, & \text{if } y_j < d_j \\ -\eta f'_j x_i^j, & \text{otherwise} \end{cases} \quad (3)$$

Remark that, in this new equation, we do not have anymore the derivative of the error. Now we have the derivative of the activation function.

Now, let us consider the case in which the activation function is the *sigmoid*. The expression of the sigmoid function is:

$$f(x) = \frac{1}{1 + \exp(-x)}$$

So, the derivative of the sigmoid function is:

$$\begin{aligned} f'(x) &= \frac{\exp(-x)}{(1 + \exp(-x))^2} = \frac{1}{1 + \exp(-x)} \cdot \frac{\exp(-x)}{1 + \exp(-x)} \\ &= \frac{1}{1 + \exp(-x)} \cdot \left(1 - \frac{1}{1 + \exp(-x)} \right) = f(x)(1 - f(x)) \end{aligned}$$

Interestingly, it is possible to remark that the derivative of the sigmoid can be expressed as a function of the sigmoid itself. According to the terminology used so far, the output of the neuron on the j th observation is y_j , so if the activation function f is the sigmoid, we can write:

$$f'_j = y_j(1 - y_j)$$

So, if we use the sigmoid as an activation function, Eq. (3) can be rewritten as:

$$\Delta w_i = \begin{cases} \eta y_j(1 - y_j) x_i^j, & \text{if } y_j < d_j \\ -\eta y_j(1 - y_j) x_i^j, & \text{otherwise} \end{cases}$$

In other words, to update the weight of a connection we just need:

- The input entering in that connection (x_i^j);
- The output of the neuron (y_j).

In other terms, for each observation $j=1,2,\dots,n$ in the training set, given that for all $i=1,2,\dots,m$, x_i^j is known, all we need to perform the update of the weights is to calculate the output of the neuron y_j . Once this value is calculated, all the needed information to perform the update of the weights is known. The reader is also invited to remark that, after some mathematical passages, the gradient descent formulation has been transformed into a much simpler equation, which does not even contain any derivative calculation, in case the used activation function is the sigmoid.

As conclusion, the Delta Rule Algorithm for the sigmoid activation function is:

Delta Rule (in case of sigmoid activation function):

- Initialize all connections with random weights;
- **repeat until** a termination condition is satisfied:
 1. Select a training observation j
 2. Calculate the output of the neuron for this observation:

$$y_j = f\left(\sum_{i=0}^m w_i x_i^j\right)$$

3. **if** y_j approximates d_j is a satisfactory way

then do nothing and go to point 1.

else for each connection $i=1,2,\dots,m$:

if $y_j < d_j$

then $w_i = w_i + \eta y_j (1 - y_j) x_i^j$

else $w_i = w_i - \eta y_j (1 - y_j) x_i^j$

end if

end if

end repeat

Possible termination conditions of this algorithm are:

- the output of the neuron approximates the target value in a satisfactory way for each training instance (observation), or
- prefixed maximum number of iterations (parameter) was performed.

Also, remark that, contrarily to what typically happens for the Perceptron, the output of the activation function in the Delta Rule is a continuous value. So, our objective is that this value approximates the target "in a satisfactory way", more than being exactly identical to the target. In general, the definition of "satisfactory" is defined by means of a prefixed error threshold ε . In this way, for a given training instance j , y_j is considered to approximate "in a satisfactory way" the corresponding target d_j is and only if: $|d_j - y_j| \leq \varepsilon$.

In the continuation, we the objective of clarifying each step of the Delta Rule algorithm, we present a simulation of execution of that algorithm on a simple numeric example.

Example (application of the delta rule)

Let us consider the following "toy" training set:

(x_1, x_2)	d
$(-2, 6)$	1
$(6, 2)$	1
$(-4, 1)$	0
$(-10, 4)$	0

The objective of this example is to simulate, for some steps, the execution of the Delta Rule algorithm using this training set. Intuitively, this training set represents a simple classification problem, whose objective is to find a model able to classify observations $(-2, 6)$ and $(6, 2)$ into one class (labelled as 1) and observations $(-4, 1)$ and $(-10, 4)$ into another class (labelled as 0).

The first step is the weights initialization. It is typically done in a random way. Let us assume, as an example, that the generator of random numbers of our programming language has generated the following values for the initial weights (exactly as in the example that we have studied previously for the Perceptron):

$$w_1 = 1, \quad w_2 = 2, \quad w_0 = -12$$

Furthermore, let us suppose that we keep the learning rate constant for all the execution of the learning algorithm (and let that constant value be $\eta=0.1$) and let us consider a sigmoid activation function.

Let us assume that we select the first observation (first line) in the dataset (i.e. $\bar{x}_1 = (-2, 6)$). Let us calculate the output of the neuron for this observation (all the calculations will be truncated at the second decimal digit from now on):

$$y_1 = f(w_0 + w_1x_1 + w_2x_2) = f(-12.0 + 1.0 \cdot (-2.0) + 2.0 \cdot 6.0) = 0.12$$

The corresponding target d_1 is equal to 1.0, so we have that $y_1 < d_1$, so we update the weights like follows: for each $i=0,1,2$:

$$w_i = w_i + \eta \cdot y_1 \cdot (1 - y_1) \cdot x_i^1$$

So:

$$w_0 = w_0 + \eta \cdot y_1 \cdot (1 - y_1) \cdot 1 = -12.0 + 0.1 \cdot 0.12 \cdot 0.88 = -11.99$$

$$w_1 = w_1 + \eta \cdot y_1 \cdot (1 - y_1) \cdot x_1^1 = 1.0 + 0.1 \cdot 0.12 \cdot 0.88 \cdot (-2.0) = 0.98$$

$$w_2 = w_2 + \eta \cdot y_1 \cdot (1 - y_1) \cdot x_2^1 = 2.0 + 0.1 \cdot 0.12 \cdot 0.88 \cdot 6.0 = 2.06$$

The new weights are:

$$w_0 = -11.99, w_1 = 0.98, w_2 = 2.06$$

And the new output is:

$$y = f(w_0 + w_1x_1 + w_2x_2) = f(-11.99 + 0.98 \cdot (-2.0) + 2.06 \cdot 6.0) = 0.17$$

The new output is closer to the target than the previous one.

Let us now assume that at the next step we select *the same* input observation (which is possible, given that the Delta Rule algorithm does not impose any prefixed strategy, or order, to select the next observation), and let us perform another iteration of the algorithm. We are still in a situation where $y_1 < d_1$, so weights are modified as follows:

$$w_0 = w_0 + \eta \cdot y_1 \cdot (1 - y_1) \cdot 1 = -11.99 + 0.1 \cdot 0.17 \cdot 0.83 = -11.98$$

$$w_1 = w_1 + \eta \cdot y_1 \cdot (1 - y_1) \cdot x_1^1 = 0.98 + 0.1 \cdot 0.17 \cdot 0.83 \cdot (-2.0) = 0.95$$

$$w_2 = w_2 + \eta \cdot y_1 \cdot (1 - y_1) \cdot x_2^1 = 2.06 + 0.1 \cdot 0.17 \cdot 0.83 \cdot 6.0 = 2.15$$

The new weights are:

$$w_0 = -11.98, w_1 = 0.95, w_2 = 2.15$$

And the new output is:

$$y = f(w_0 + w_1x_1 + w_2x_2) = f(-11.98 + 0.95 \cdot (-2.0) + 2.15 \cdot 6.0) = 0.27$$

As we can see, this results is even closer to the target than at the previous iteration.

Discussion

The objective of the previous example is to convince the reader that:

- The Delta Rule is a very nice way of implementing gradient descent, fast and easy to execute and (in case the used activation function is the sigmoid) without having to calculate any derivative;
- Gradient descent actually works, in the sense that, at every step, the output is closer to the target than at the previous one. In other words, the error decreases at every step.

However, as mentioned earlier while we were studying the Perceptron, the problem of the previous example is linearly separable. So, as the previous Theorem tells us, the error surface is unimodal (no local optima), and decreasing the error at each step will bring us arbitrarily close to the target. But if the problem was non-linearly separable, as we already know, the error surface may contain local optima. For this reason, we need to generalize the Delta Rule to multi-layer Neural Networks. This will be done in the next section, presenting the Backpropagation.

Backpropagation

Looking at the structure of the multi-layer network represented in Fig. 1, a question arises natural: why cannot we use any of the learning rules that we have studied so far (like for instance the Perceptron learning rule, or the Delta rule) for all neurons independently? The answer is straightforward: all the learning rules that we have studied so far use the target value of the neurons as an information for learning. For instance, as we have seen in the previous section, the Delta Rule uses target values to calculate an error between outputs and targets, and the objective is to minimize that error. We obviously can do it for the output neurons in the network of Fig. 1, but we clearly cannot do it for the hidden neurons, where we do not have any notion of an existing target value. So, the Backpropagation learning rule works exactly like the Delta Rule on the output neurons, but extends the Delta Rule, so that it can function also for updating the weights entering in the hidden neurons.

The Backpropagation (Haykin, 1998) is the most common learning rule for multi-layer Neural Networks (sometimes also called multi-layer Perceptron or Feed-Forward Neural Networks). The basic idea of the Backpropagation is the following: the errors of the *output neurons* are *propagated backwards* to the *hidden neurons*. In other words: the error of a *hidden neuron* is defined as the sum of all the errors of all the *output neurons* to which it is directly connected.

The Backpropagation can be applied to networks with any number of layers, but the following theorem holds:

Universal Approximation Theorem

Only one layer of hidden neurons is sufficient to approximate any function with a finite number of discontinuities with arbitrary precision, provided that the activation functions of the hidden neurons are *non linear* (Hornik *et al.*, 1989).

This is one of the reasons why one of the most used configurations of a Feed-Forward multi-layer Neural Network is with only one layer of hidden units, and using a sigmoidal activation function which is non-linear:

$$f(x) = \frac{1}{1 + e^{-x}}$$

The reader is also invited to remark that this is the second good reason for using the sigmoid as an activation function. The first one has been seen in the previous section, and it consists in the fact that, in case the sigmoid is used as an activation function, we are able to transform the gradient descent method in a very simple equation, where no derivative needs to be calculated.

Backpropagation is composed by one *forward step* and one *backward step*. In the forward step, the weights of the connections remain unchanged and the outputs of the network are calculated propagating the inputs through all the neurons of the network. At this point, the error of each output neuron is calculated. The backward step consists in the modification of the weights of each connection. This calculation is made in the following way:

- for the output neurons, the modification is done by means of the Delta Rule;
- for the hidden neurons, the modification is done by *propagating backwards* the error of the output neurons: the error of each hidden neuron is considered as being equal to the sum of all the errors of the neurons of the subsequent layer.

The weights of the connections that enter in each neuron are updated using the formula:

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

where w_{ij} is the weight of the connection between unit i and unit j and:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

In order to calculate Δw_{ij} , as for the Delta Rule, we start from gradient descent, distinguishing the cases of update of the weights of the connections entering into the output neurons and the hidden neurons. These two different cases are presented in the next two sections, respectively.

Weights update – Output neurons

For the output neurons, the process is exactly identical to the one that we have seen for the Delta Rule. The reader is invited to repeat the mathematical passages, using this time the mean square error (instead of the absolute error, as it was done in the previous section for the Delta Rule). By doing this, the reader should be able to verify that the weight of each synapsis connecting the i th hidden neuron to the j th output neuron should be updated as:

$$\Delta w_{ij} = \eta (d_j - y_j) f'_j z_i$$

where d_j is the target value for the j th output neuron, y_j is the output of the j th output neuron, f'_j is the derivative of the activation function of the j th output neuron and z_i is the output of the i th hidden neuron.

If the activation function is the sigmoid, this gives:

$$\Delta w_{ij} = \eta (d_j - y_j) y_j (1 - y_j) z_i$$

This is straightforward to calculate, after that the forward pass of the algorithm has been performed, and so the outputs of all the neurons have been calculated.

If we now “isolate” all the terms concerning the j th output neuron and we define:

$$\beta_j = (d_j - y_j) y_j (1 - y_j) \quad (4)$$

We can conclude that

$$\Delta w_{ij} = \eta \beta_j z_i \quad (5)$$

This last equation will be used after that we will have learned how to modify the weights of the connections entering into the hidden neurons. More specifically, we will discover that those weights can be modified by means of an equation that is extremely similar to Eq. (5).

Weights update – Hidden neurons

In order to obtain a way of modifying the weights of the connections entering into the hidden neurons, first of all let us fix some terminology. Let us consider that our network contains just one layer of hidden neurons (the case of any number of hidden layers can be easily obtained from the previous one) and let p_{hi} be the weight of the connection between the h th input unit and the i th hidden neuron (this is the weight that we will have to update for each h and for each i). Also, like in Section Weights update – Output neurons, let z_i be the output of the i th hidden neuron, and let all the rest of the connections and output values follow the same terminology as in Section Weights update – Output neurons. Also, let v_i be the input of the activation function of the i th hidden neuron. In other words:

$$z_i = f(v_i)$$

where f is the activation function, and:

$$v_i = \sum_h p_{hi} IN_h$$

where IN_h is the value of the h th input unit.

As for the case of the output neurons (that use the Delta Rule), also for the hidden neurons, the starting point is given by the gradient descent formulation:

$$\Delta p_{hi} = -\eta \frac{\delta E}{\delta p_{hi}} \quad (6)$$

Also in this case, the previous equation will be transformed into a more “mathematically convenient” formulation. As for the Delta Rule, the first step is decomposing the previous equation into a number of terms. In this case, let us rewrite the term $\frac{\delta E}{\delta p_{hi}}$ as follows:

$$\frac{\delta E}{\delta p_{hi}} = \frac{\delta E}{\delta z_i} \frac{\delta z_i}{\delta v_i} \frac{\delta v_i}{\delta p_{hi}} \quad (7)$$

Let us begin by developing the term $\frac{\delta z_i}{\delta v_i}$ of Eq. (7). We have:

$$\frac{\delta z_i}{\delta v_i} = \frac{\delta f(v_i)}{\delta v_i} = f'(v_i)$$

As in the previous section, let us accept the simplified notation f'_i to indicate the derivative of the activation function of the i th hidden neuron, so:

$$\frac{\delta z_i}{\delta v_i} = f'_i$$

Now, let us develop the term $\frac{\delta v_i}{\delta p_{hi}}$ of Eq. (7). We have:

$$\frac{\delta v_i}{\delta p_{hi}} = [IN_1 p_{i1} + IN_2 p_{i2} + \dots + IN_h p_{ih} + \dots]' dp_{ih} = IN_h$$

In fact, the derivative is performed with respect to variable p_{hi} and all other terms in the summation are constant.

Finally, let us develop the term $\frac{\delta E}{\delta z_i}$ of Eq. (7). In this case, we have to remember that E , i.e., the error of the i th hidden neuron, is unknown. Also, we have to remember the fact that the Backpropagation algorithm uses the hypothesis that, by definition, the error of the hidden neurons is equal to the sum of the errors of all the output neurons. In other terms:

$$E = \sum_{k=1}^m EOUT_k$$

where $EOUT_k$ is the error of the k th output neuron. So:

$$\frac{\delta E}{\delta z_i} = \frac{\delta (\sum_{k=1}^m EOUT_k)}{\delta z_i}$$

and using the notion that the derivative of a sum is equal to the sum of the derivatives, we can write:

$$\frac{\delta E}{\delta z_i} = \sum_{k=1}^m \frac{\delta (EOUT_k)}{\delta z_i} \quad (8)$$

But the term $\sum_{k=1}^m \frac{\delta (EOUT_k)}{\delta z_i}$ can also be rewritten as:

$$\sum_{k=1}^m \frac{\delta (EOUT_k)}{\delta z_i} = \sum_{k=1}^m \frac{\delta (EOUT_k)}{\delta e_k} \frac{\delta e_k}{\delta z_i} \quad (9)$$

where e_k is the difference between the target value for the k th output neuron and its output, i.e.:

$$e_k = d_k - y_k$$

Let us now develop the term $\frac{\delta(EOUT_k)}{\delta e_k}$ of Eq. (9). We have:

$$\frac{\delta(EOUT_k)}{\delta e_k} = \left[\frac{1}{2} e_k^2 \right]' de_k = e_k$$

where, as for the case of the output neurons when the mean square error is considered, $\frac{1}{2}$ was used as a coefficient simply because it is “mathematically convenient”. In other terms, using this coefficient, it is possible to eliminate any multiplicative constant in the result of the derivative, simplifying the final result without any impact on the final algorithm.

Let us now develop the term $\frac{\delta e_k}{\delta z_i}$ of Eq. (9). That term can be rewritten as:

$$\frac{\delta e_k}{\delta z_i} = \frac{\delta e_k}{\delta v_k} \frac{\delta v_k}{\delta z_i} \quad (10)$$

where v_k is the input of the activation function of the k th output neuron (in other words: $y_k = f(v_k)$).

Let us first develop the term $\frac{\delta v_k}{\delta z_i}$ of Eq. (10). We have:

$$\frac{\delta e_k}{\delta v_k} = \frac{\delta(d_k - \gamma_k)}{\delta v_k} = \frac{\delta(d_k - f(v_k))}{\delta v_k} = -f'_{OUT_k}$$

where f'_{OUT_k} is the notation that we use to indicate the derivative of the activation function of the k th output neuron.

Finally, let us develop the term $\frac{\delta v_k}{\delta z_i}$ of Eq. (10). We have:

$$\frac{\delta v_k}{\delta z_i} = \frac{\delta(z_1 w_{1k} + z_2 w_{2k} + \dots + z_i w_{ik} + \dots)}{\delta z_i}$$

If we join the last two terms we have developed, we are now able to find a way of expressing $\frac{\delta E}{\delta z_i}$ by rewriting Eq. (8). We have:

$$\frac{\delta E}{\delta z_i} = \sum_{k=1}^m e_k (-f'_{OUT_k}) w_{ik} = - \sum_{k=1}^m e_k f'_{OUT_k} w_{ik}$$

We point out that, in this last expression, the summation runs over all output neurons and w_{ik} indicates the weight between the i th hidden neuron and the k th output neuron, that has already been updated by the Backpropagation algorithm (see Section Weights update – Output neurons).

Joining all the developments made so far, we are finally able to rewrite Eq. (7) as:

$$\frac{\delta E}{\delta p_{hi}} = \frac{\delta E}{\delta z_i} \frac{\delta z_i}{\delta v_i} \frac{\delta v_i}{\delta p_{hi}} = - \sum_{k=1}^m e_k f'_{OUT_k} w_{ik} f'_i IN_h$$

and so we are now able to rewrite Eq. (6), obtaining the following formula to update the weights of the connections entering into the hidden neurons:

$$\Delta p_{hi} = \eta \sum_{k=1}^m e_k f'_{OUT_k} w_{ik} f'_i IN_h \quad (11)$$

where:

- $e_k = d_k - \gamma_k$ is the difference between the target value and the output of the k th output neuron;
- f'_{OUT_k} is the derivative of the activation function of the k th output neuron;
- w_{ik} is the weight of the connection between the i th hidden neuron and the k th output neuron;
- f'_i is the derivative of the activation function of the i th hidden neuron;
- IN_h is the value of the h th input unit.

The reader should now try to convince herself that all the previous quantities are known and, if the activation functions of all the neurons are the sigmoid function, also easy to calculate.

If we now define:

$$\gamma_i = \sum_{k=1}^m e_k f'_{OUT_k} w_{ik} f'_i \quad (12)$$

Eq. (11) can be rewritten as:

$$\Delta p_{hi} = \eta \gamma_i IN_h \quad (13)$$

The reader should recognize that, unless for the name of some involved quantities, Eq. (13) is identical to Eq. (5) for updating the weights of the connections entering into the output neurons.

Furthermore, it is interesting to remark that, in Eq. (12), the term $e_k f'_{OUT_k}$ corresponds to the definition of the quantity β_k given in Eq. (4) for the output neurons. So, assuming that β_k has already been calculated, because the weights of the connections entering into the output neurons are updated before the ones of the connections entering into the hidden neurons, we can simplify the Eq. (12) by writing:

$$\gamma_i = f'_i \sum_{k=1}^m \beta_k w_{ik} \quad (14)$$

Let us now see how, despite the complexity of the previous mathematical steps, the modification of the weights of the connections entering into the hidden neurons is simple, in case we assume that also the hidden units use the sigmoid activation function. In this case, Eq. (14) can be rewritten as:

$$\gamma_i = z_i (1 - z_i) \sum_{k=1}^m \beta_k w_{ik}$$

while Eq. (13) remains:

$$\Delta p_{hi} = \eta \gamma_i IN_h$$

where:

- z_i is the output of the i th hidden neuron, and it is known because it has been calculated in the forward pass of the algorithm, where the outputs of all the neurons have been calculated;
- $\beta_1, \beta_2, \dots, \beta_m$ have already been calculated when updating the weights of the output neurons.
- $w_{i1}, w_{i2}, \dots, w_{im}$ are the weights of the connections entering into the output neurons, and they have already been updated (see Section Weights update – Output neurons).
- IN_h is the value of the h th input unit, and it is of course known from the training set.

As a conclusion, we can only recognize that the Backpropagation is able to update all the weights in the network in a simple and extremely efficient way. The complete Backpropagation algorithm is presented below.

The backpropagation algorithm

Summarizing all the achievements obtained so far, we have that:

- For the generic j th output neuron:
 1. Let d_j be the expected correct output (target);
 2. Let y_j be the calculated output

Calculate:

$$\beta_j = (d_j - y_j) y_j (1 - y_j)$$

1. Let y_i be the i th input of the j th output neuron (i.e. the output of the i th hidden neuron)

then the modification of the weight is:

$$\Delta w_{ij} = \eta \beta_j y_i$$

- For the generic i th output neuron:
 1. Let z_i be the calculated output;
 2. Let $\beta_1, \beta_2, \dots, \beta_m$ be the previously calculated quantities for all output neurons;
 3. Let $w_{i1}, w_{i2}, \dots, w_{im}$ be the weights of the connections joining the i th hidden neuron to all output neurons (already updated at previous step)

Calculate:

$$\gamma_i = z_i (1 - z_i) \sum_{k=1}^m \beta_k w_{ik}$$

1. Let INP_h be the h th input of the i th hidden neuron (i.e. the value of the h th input unit of the network)

then the modification of the weight is

$$\Delta p_{hi} = \eta \gamma_i INP_h$$

After some renaming, we are finally able to write an algorithm for the Backpropagation learning rule:

Backpropagation (in case of sigmoid activation function):

- Initialize all the weights in the network with random values.
- **repeat until** a given termination condition is satisfied
 1. Select a vector $\bar{x}$ of the training set (let $\bar{d}$ be correct output – or target – that can be found in the dataset, corresponding to vector $\bar{x}$). Calculate the output of the network $\bar{y}$ for input $\bar{x}$.

```

2. if  $\bar{y}$  approximates  $\bar{d}$  in a satisfactory way
   then do nothing and go to point 1.
   else:
     2.1 For each output neuron  $j$ , calculate  $\delta_j = (d_j - y_j) y_j (1 - y_j)$  and modify the weights of the connections that enter into
         neuron  $j$  as follows:  $w_{ij} = w_{ij} + \eta \delta_j y_i$ 
     2.2 For each hidden neuron  $j$ , calculate  $\gamma_j = y_j (1 - y_j) \sum_k \delta_k w_{jk}$  (where the sum is calculated over all output neurons
          $k$ ), and modify the weights of the connections that enter into neuron  $j$  as follows:  $w_{ij} = \eta \gamma_j y_i$ 
         (remark that both in both points 2.1. and 2.2.,  $y_i$  is the value entering in neuron  $j$  from source  $i$ , while  $y_j$  is the output of
         neuron  $j$ ).
   end if
end repeat

```

Possible termination conditions for this algorithm can be:

- The correct outputs $\bar{d}$ are approximated from the outputs $\bar{y}$ of the network in a “satisfactory” way for every vector $\bar{x}$ of the training set, or
- A prefixed number of maximum iterations has been executed.

Now that we have been able to obtain an efficient algorithm for training feed-forward multi-layer neural networks, let us discuss how to set some important parameters of this networks and one of the most well-known issues of these networks: overfitting.

Backpropagation – Parameter setting

In this section, ideas on how to use feed-forward multi-layer Neural Networks in practice are presented. In particular, the setting of some important parameters such as the learning rate, the momentum, the size of the training set, the number of layers and the number of hidden neurons is discussed. In this discussion, the focus is on the important issue of overfitting, and methods to limit overfitting in order to bestow on the network a reasonable generalization ability. Before continuing, however, the reader should be aware that, for obvious reasons of space, this section is only introductory and should be considered as being far from complete. In fact, given the huge amount of research published in the field, the objective of covering all the subject by revising all the important contributions would be absolutely utopic in such a short article. More specifically, only some of the most well-known and, so to say, “historical” results are discussed in this section, and summarized in a very simple and intuitive way. For a very recent publication, containing several recent advances on the subject, the reader is referred, for instance, to [Fengyu Cong \(2017\)](#).

Learning Rate η

The parameter η has an influence on the speed of learning of a Neural Network. In particular:

- If η is too small, the learning of the network can be slow.
- If η is too big, the network can be “unstable” (in other words, “oscillations” can happen around the optimal value of the vector of weights).

[Sorensen \(1994\)](#) proposes to begin the algorithm with a relatively big value such as $\eta = 0.1$, and to gradually decrease the value of η during the execution of the algorithm. In this way, for a large part of the applications, we are able to reduce (or even eliminate) the risk of having numerous oscillations around the optimal value of the weights.

Momentum α

In spite of all the achievements obtained so far, another problem remains: it is still possible to find a *local minimum* on the error surface. In fact, if the problem is not linearly separable, the surface of the error function can be multimodal, i.e. there can be the presence of several local minima. The Backpropagation, as it is the Delta Rule, is nothing but a convenient way of expressing the gradient descent formulation. As such, also the Backpropagation, such as it was for the Delta Rule, allows us to decrease the error at each step, but may get trapped into a local minimum of the error surface. To reduce the risk of getting trapped in a local minimum, the Delta Rule and the Backpropagation can be modified by inserting a parameter, called *momentum* α :

$$\Delta w_{ij} = \eta \delta y_i + \alpha \Delta w_{ij}$$

By its very definition, the momentum α is such that $0 \leq \alpha < 1$. The value of the momentum is typically a *random number*, drawn to escape from local minima with a given probability ([Sorensen, 1994](#)).

Local Minima and Hidden Neurons

Sometimes, another way of reducing the risk of getting stuck in local minima of the error surface consists in augmenting the *number* of hidden neurons. But it has also been experimentally shown that in some cases if this number is too high, the risk of

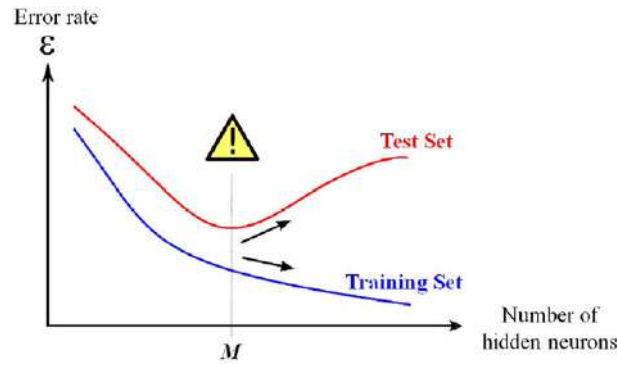


Fig. 3 Typical error rates on the training set and on the test set as the number of hidden units change.

getting stuck into local minima can increase instead of decreasing (Sorensen, 1994). To find the right number of hidden neurons to solve a problem is often an empirical task, based on experience and on the knowledge that we have of the problem. Often, some preliminary experiments are needed to find the right number. A little help (i.e. an heuristic formula that binds the dimension of the training set, the number of hidden neurons, the total number of connections of the network and the average admissible error) is given later in this document.

Number of Vectors in the Training Set

To obtain a good generalization from Neural Networks, we have to keep into account that:

- If the training set is too big, there can be a risk of *overfitting* (i.e., the network is too specialized for the training data).
- If the training set is too small, there can be a risk of *underfitting* (i.e., the network did not learn enough).
- In the case of a network with only one level of hidden neurons, some studies have been done regarding the suitable training set size to use to solve a problem (Baum and Haussler, 1989). This study reports that:
 - if we call ε the average of the admissible errors between outputs and targets on test data, and the average error of the network on the training set is smaller than $\varepsilon/2$,
 - if we call M the number of hidden neurons and W the total number of connections in the network,

then an appropriate number of training vectors is given by:

$$\text{Training set size} = K = O\left(\frac{W}{\varepsilon} \ln\left(\frac{M}{\varepsilon}\right)\right) \quad (15)$$

Number of Layers and Number of Neurons

In a large part of the practical cases, the surface of the error function contains a finite number of discontinuities and thus *only one* layer of hidden neurons is sufficient to approximate the target function with arbitrary precision, provided that the activation functions of the hidden neurons are non linear (Theorem of Universal Approximation (Hornik et al., 1989)). However, finding the right number of neurons to allocate in the hidden layer is still an open problem. Also for this parameter, we have to take into account that:

- If it is too big, we can have the risk of overfitting.
- If it is too small, we can have the risk of underfitting.

Generally, the error rates on the training set and on the test set change, during the learning phase and with the number of hidden neurons, as idealized in Fig. 3. Our objective is to find the number of hidden units M that is able to minimize the error on the test set. To find this value is not easy, because it depends on the problem, but Eq. (15) can be helpful. Often, only practice can help and it is necessary to perform a set of experiments to look for the suitable value M for the application at hand.

Conclusions

This paper has presented the Backpropagation learning rule, probably nowadays the most popular algorithm for training feed-forward multi-layer Neural Networks. Before studying the details of the Backpropagation, the Delta Rule was presented as a method to take advantage, in a practical and efficient way, of the gradient descent technique. Then, the Backpropagation has been presented, as a generalization of the Delta Rule to multi-layer Neural Networks, where the error of a hidden neuron is defined as the sum of the errors of the neurons in the subsequent layer, to which it is directly connected. In this phase, particular emphasis was

given to the method used to update the weights entering into the hidden neurons, while for the output neurons the method is identical to the Delta Rule. Finally, ideas to appropriately set some important parameters, in order to limit overfitting, have been discussed.

See also: Algorithms for Graph and Network Analysis: Graph Alignment. Artificial Intelligence and Machine Learning in Bioinformatics. Artificial Intelligence. Data Mining in Bioinformatics. Knowledge and Reasoning. Machine Learning in Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Stochastic Methods for Global Optimization and Problem Solving. The Challenge of Privacy in the Cloud

References

- Baum, E.B., Haussler, D., 1989. What size net gives valid generalization? *Neural Comput.* 1 (1), 151–160. doi:10.1162/neco.1989.1.1.151.
- Fengyu Cong, A., 2017. *Advances in Neural Networks – ISNN*. Cham: Springer International Publishing, Available at: <https://books.google.pt/books?id=apgnDwAAQBAJ>.
- Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation, second ed.* Upper Saddle River, NJ: Prentice Hall PTR.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Netw.* 2 (5), 359–366. doi:10.1016/0893-6080(89)90020-8.
- Minsky, M.L., Papert, S.A., 1988. *Perceptrons: Expanded Edition*. Cambridge, MA: MIT Press.
- Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65, 386–408.
- Sorensen, O., 1994. *Neural networks in control applications*. PhD Thesis, Aalborg Universitetsforlag.

Deep Learning

Massimo Guarascio, Giuseppe Manco, and Ettore Ritacco, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

With *Deep Learning* (DL) we refer to a wide family of machine learning techniques using deep architectures in order to learn complex (non-linear) functions, which can be used to tackle several difficult tasks in challenging application scenarios (e.g., *Computer Vision*, *Natural Language Processing*, *Speech Recognition*, *Recommender Systems* and other AI applications). In a nutshell, the approaches based on deep learning aim at training Artificial Neural Networks (ANNs) composed by a large number of hidden layers, so that the resulting models are both efficient and effective. The intuition behind Deep Neural Networks (DNNs) is that the hierarchical structure devised by the hidden layers provides subsequent levels of abstractions: each layer of the architecture learns a set of features with an higher level of abstraction compared to the previous one. In this way, it is possible to highlight the relevant discriminative features even from low level data, such as pixels of medical images or biological sequences.

Exploiting multiple levels of abstractions, *Deep Architectures* allow to discover highly accurate models, by capturing interactions between set of features directly from raw and noisy data. The latter represents one of the most important and disruptive aspects introduced by the DL framework: no feature engineering or interaction with domain experts are required to build good discriminative features.

Historically, DNNs do not represent a substantial architectural difference from the original neural network models introduced in the 1960s (Haykin, 1998). However, their exploitation was limited, due to the difficulty of training. Researchers reported positive results with architectures composed of two or three levels, but essentially training deeper networks did not yield significantly stronger results.

The impulse to new research in the field was given in the last decade. Particularly relevant in the scientific community is the paper by Hinton *et al.* (2006), where an unsupervised pre-training initialization step for the intermediate layers was introduced that would boost the performance of these networks. From there, new initialization methods were investigated (Bengio *et al.*, 2007; Erhan *et al.*, 2010) and other algorithms for deep architectures were proposed.

Since 2006 deep networks were successfully applied in several tasks and application domains. For the purpose of this book, relevant applications can be devised in the analysis of genomic sequences (Alipanahi *et al.*, 2015; Rizzo *et al.*, 2015) or mining of medical images (Ning *et al.*, 2005; Ciresan *et al.*, 2013; Ronneberger *et al.*, 2015; Gulshan *et al.*, 2016; Esteva *et al.*, 2017).

This article is aimed at introducing the basic concept that characterize DNNs and their principles. Clearly, due to the wide nature of the topic the content of this survey is limited. The interested reader can refer to some reference articles (Bengio, 2009; Le Cun *et al.*, 2015; Schmidhuber, 2015; Prieto *et al.*, 2016; Sze *et al.*, 2017; Liu *et al.*, 2017) and books (Deng and Yu, 2014; Goodfellow *et al.*, 2016; Patterson and Gibson, 2017).

Background

ANNs are models inspired by the neural structure of the brain. The brain basically learns from experience and stores information as patterns. These patterns can be highly complex and allow us to perform many tasks, such as recognizing individual from the image of their faces from many different angles. This process of storing information as patterns, applying those patterns and making inferences represents the approach behind ANNs.

The Perceptron

A neuron is the basic component of a neural network and encompasses some basic functionalities inspired from biology. Notably, a biological neuron is a cell connected to other neurons and acts as a hub for electrical impulses. Fig. 1 provides a graphical sketch of a neuron. A neuron has a roughly spherical cell body called soma, which processes the incoming signals and converts them into output signals. Input signals are collected from extensions on the cell body called dendrites. These output signals are transmitted to other neurons through another extension called axon, which prolongs from the cell body and terminates into several branches. The branches end up into junctions transmitting signals from one neuron to another, called synapses.

The behavior of a neuron is essentially electro-chemical. An electrical potential difference is maintained between the inside and the outside of the soma, due to different concentrations of sodium (Na) and potassium (K) ions. When a neuron receives inputs from a large number of neurons via its synaptic connections, there is a change in the soma potential. If this change is above a given threshold, it results in an electric current flowing through the axon to other cells. Then the potential drops down below the resting potential and neuron cannot fire again until the resting potential is restored (Fig. 2).

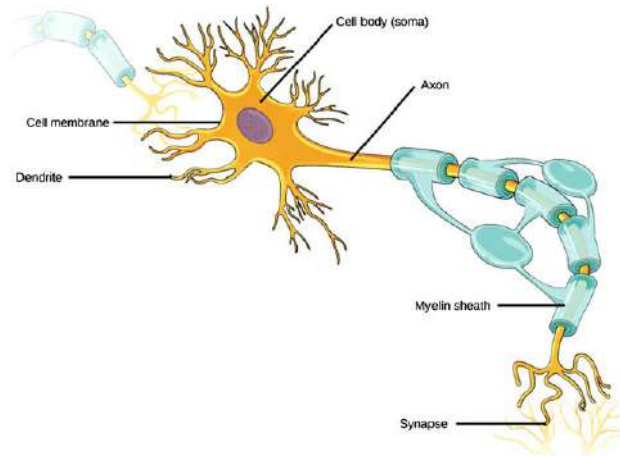


Fig. 1 Biological neuron. Modified from “Neurons and glial cells: Figure 2”, by OpenStax College, Biology, https://cnx.org/contents/GFy_h8cu@9.87:c9j4p0aj@3/Neurons-and-Glial-Cells.

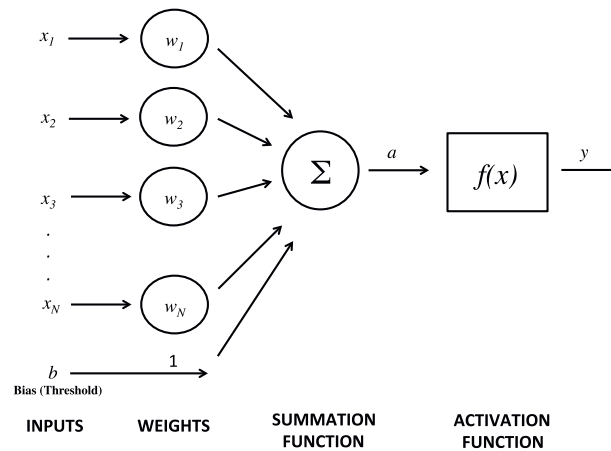


Fig. 2 Artificial neuron.

An artificial neuron (also called perceptron) emulates this behavior. There are essentially two phases, illustrated in Fig. 1. First, input signals $x_1, \dots, x_n$ are collected through the connections and a linear combination of such inputs is computed as $a = \sum_{i=1}^N x_i w_i + b$. The linear combination is the input for the second phase, where an activation function emits an output signal y . The activation function mimics the behavior of the threshold on the potential difference: when the value of a is substantially different from a steady state, a signal $y = f(a)$ can be emitted which is modeled according to an *activation function* f .

The simple structure of a perceptron is natural model for supervised learning within a linearly separable space (Mitchell, 1997): for example, by considering the activation function $f(x) = \text{sign}(x)$, a perceptron characterized by a vector $\mathbf{w} = (w_1, \dots, w_n)^T$ of weights and bias b models the binary classification problems where a given input tuple $\mathbf{x} = (x_1, \dots, x_n)^T$ can be classified as *positive* if $\mathbf{x} \cdot \mathbf{w} > -b$ and *negative* otherwise. That is, the pair $(\mathbf{w}, b)$ represent a hyperplane within the n -dimensional space, and each point above the hyperplane can be classified as positive, whereas points below it are classified as negative. Fig. 3 depicts a perceptron characterized by the parameters $\mathbf{w} = (1, -1)$ and $b = 0.5$.

An artificial neural network is essentially a combination of neurons to perform complex transformations of the input space. Fig. 4 shows a very simple ANN where the input space $\mathbf{x}$ is given as input to M different perceptrons. The graphical representation shows essentially two layers. The first layer is the input layer, where each node represents an input dimension x_i . In the second layer, nodes represent perceptrons and encode the structure of Fig. 2. That is, the j -th node in this layer represents a perceptron characterized by parameters $(\mathbf{w}_j, b_j)$ and outputting the value y_j . Thus, each perceptron is activated by the same input value $\mathbf{x}$, but the activation is characterized by different weight. Overall, the network accept an n -dimensional vector $\mathbf{x} = (x_1, \dots, x_n)^T$ as input, and outputs an M -dimensional vector $\mathbf{y} = (y_1, \dots, y_M)^T$. The characterization of the network is given by the parameters $(\mathbf{W}, \mathbf{b})$, where $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_M)^T$ is an $M \times N$ matrix where the j -th row represents vector of weights of the respective perceptron, and $\mathbf{b} = (b_1, \dots, b_M)^T$. Thus, in matrix notation, we have

$$\mathbf{y} = f(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

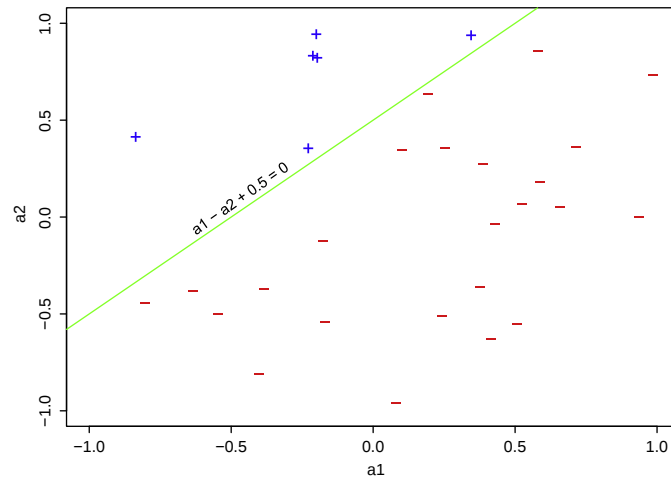


Fig. 3 Linear classification example.

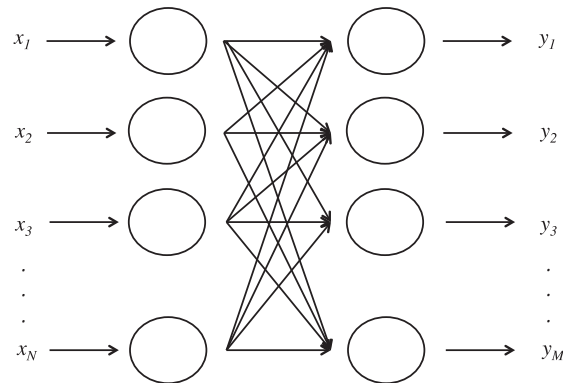


Fig. 4 Single layer ANN.

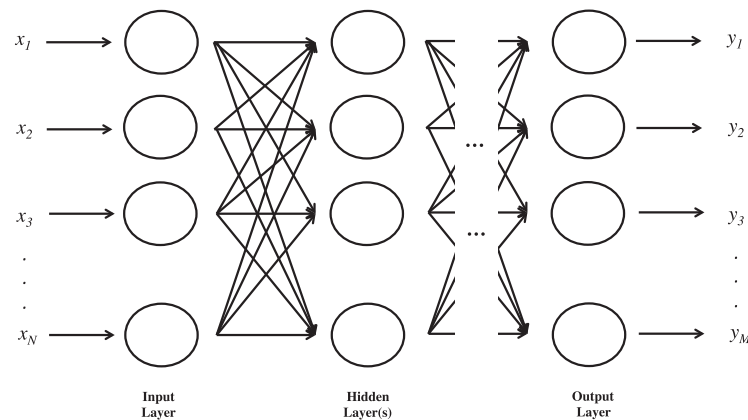


Fig. 5 Multi layer ANN.

Multi-Layer Networks and Deep Networks

The expressive power of a single-layer neural network is limited: for example, a perceptron is only effective for classification tasks where the input space is linearly separable. However, perceptrons can be combined and, in the same spirit of biological neurons, the output of a perceptron can feed a further perceptron in a connected architecture.

Stacking perceptrons into hierarchical layers represents an effective way of increasing the expressive power of a neural network.

Fig. 5 shows the general structure of a multilayer neural network with several hidden layers. The structure can be formally

characterized as follows. A neural network models a nonlinear function $y = \text{net}(\mathbf{x}; \mathbf{w}, \mathbf{b})$ from an input vector $\mathbf{x} \in \mathbb{R}^N$ to an output vector $\mathbf{y} \in \mathbb{R}^M$, controlled by two vectors $\mathbf{w}$ and $\mathbf{b}$ of adjustable parameters. Within a d -layer network, each layer is characterized by a size s_j , two vectors $\mathbf{z}^{(j)}$ and $\mathbf{a}^{(j)}$, where $s_0 = N$ is the size of the input data (i.e., number of units in the input layer), and $s_d = M$ is the size of the output (i.e., number of units in the output layer). Each node within the network is uniquely identified by a pair (h, k) , where h represents the corresponding layer and k the position of the node within the layer. To keep notation uncluttered, we assume that $S = \sum_{j=1}^m s_j$ and the vectors $\mathbf{z}$ and $\mathbf{a}$ span over $\mathbb{R}^S$ where $a_i \equiv a_v^{(h)}$ for some encoding $i = \langle h, v \rangle$. This induces a partial order $j < i$ which holds when $i = \langle h+1, v \rangle$, $j = \langle h, u \rangle$ and the two neurons are connected.

Within this representation, $\mathbf{w}$ is the vector of all the weights w_{ij} such that $j < i$. Furthermore, b_i represent the bias associated to unit i and f_i is the relative activation function. The network basically defines a chain of compositions of activation functions, the length of such chain being bounded by the number of layers. The recursive relationships for this composition are given by the equations

$$z_i = f_i(a_i) \quad (1)$$

$$a_i = \sum_{j < i} w_{ij} z_j + b_i \quad (2)$$

which are illustrated in Fig. 6. When j represents an input node, the value z_j corresponds to the value x_i of the relative input dimension. By contrast, when j represents to an output node, z_j maps directly into the corresponding output y_i for some given i .

Fig. 7(a) shows an example neural network with a single hidden layer. Starting from the input dimension (x_1, x_2) , the network iteratively computes the values a_3, a_4, a_5 (and the corresponding activations z_4, z_5 and z_6), and subsequently the combination a_7 which enables the activation on the output y . We call this iterative process *forward propagation*, and this type of architecture is an example of *feedforward neural network*, since the connectivity graph does not exhibit any directed loops or cycles. Moreover, since each neuron of a layer is connected with each neuron of the next layer, the network is *fully-connected*. The overall function computed by the network is

$$\begin{aligned} y = \text{net}(\mathbf{x}; \mathbf{w}, \mathbf{b}) = & f_6(w_{6,3}f_3(w_{3,1}x_1 + w_{3,2}x_2 + b_3) \\ & + w_{6,4}f_4(w_{4,1}x_1 + w_{4,2}x_2 + b_4) \\ & + w_{6,5}f_5(w_{5,1}x_1 + w_{5,2}x_2 + b_5) + b_6) \end{aligned}$$

To get an intuition why multi-layer neural networks increase the expressive power of a simple perceptron, recall that the latter can only model situations where the input space is linearly separable. The hidden layer in the network represented by Fig. 7(a) contains three perceptrons, where each of them can separate the input space according to a hyperplane. The output layer combines the separation in a unique decision, by enabling the overall architecture to separate regions in convex polygons, as shown in Fig. 7(b).

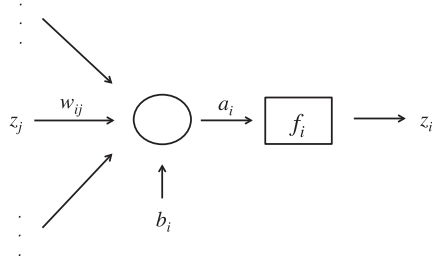


Fig. 6 Compositionality of a multi-layer NN.

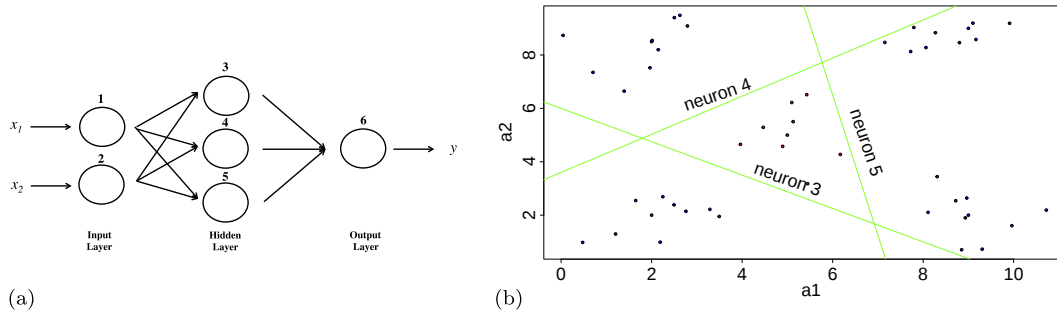


Fig. 7 Simple ANN architecture. (a) Example NN with one hidden layer. (b) Classification by separating convex polygons. each line in the two-dimensional space represents a perceptron. The intersection of the regions delimited by the lines represents the combination of the outputs of the perceptrons.

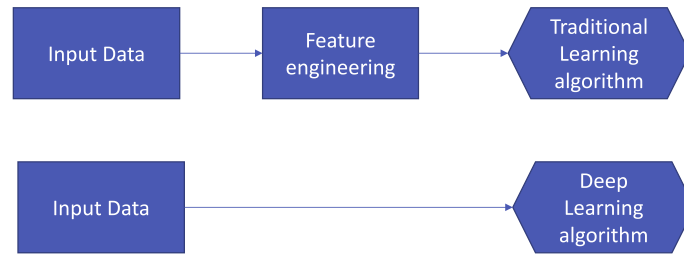


Fig. 8 Traditional machine learning versus (deep) neural network learning.

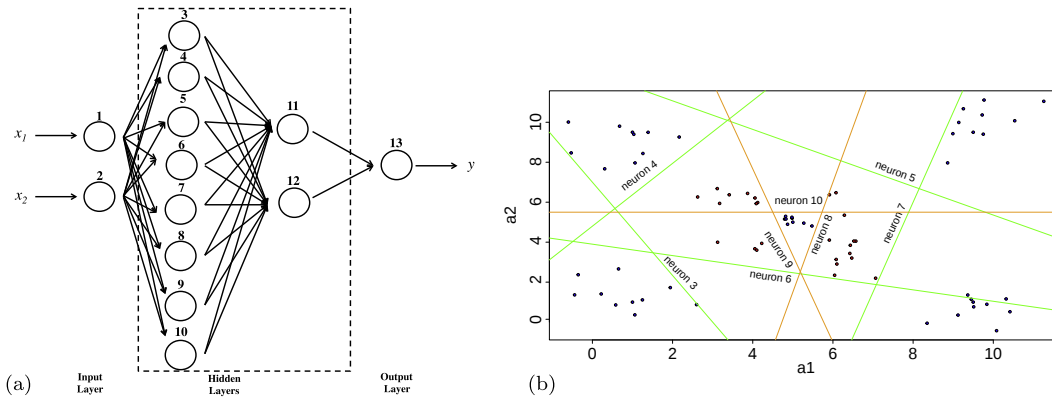


Fig. 9 Simple multi layer ANN architecture. (a) Example NN with two hidden layers. (b) Classification by separating non-convex regions. By assuming that each perceptron in the third level represents a convex region, the addition of a new level allows to combine decisions which involve such regions.

The above example helps in understanding the role of intermediate layers within a network. Intermediate layers represent natural progression from low level to high level structure within the data. Whereas input $\mathbf{x}$ represents a low level representation of the data, the vectors $\mathbf{z}^{(h)}$ represent a higher level of abstraction, and each $z_k^{(h)}$ represents a new feature which is devised at level h , based on the data coming at level $h - 1$. In this respect, Neural network learning is different from the traditional machine learning algorithm, as shown in [Fig. 8](#): the latter, indeed, require a manual feature engineering. By contrast, neural network adopt an approach where features are progressively learned directly from the low-level representation, and the feature learning is embedded within the training algorithm itself.

Deep Neural Networks (as opposed to *shallow networks*) exacerbate this feature learning general structure, by exploiting a large number of intermediate levels in order to learn complex functions and to devise several features. In this respect, a DNN emulates the mammal brain ([Thorpe and Fabre-Thorpe, 2001](#)) by processing a given input with several levels of abstractions. Clearly, multiple levels increase the expressive power accordingly: [Fig. 9\(b\)](#) provides an example of the expressive power of a relatively simple neural network with two hidden layers.

Learning a DNN

To the purpose of this article, neural networks represent machine learning models [Mitchell \(1997\)](#) which can be applied for either supervised or unsupervised learning problems. The scheme discussed in the previous section represents essentially a network which learns a function $net(\mathbf{x}; \mathbf{w})$ (From now on, we shall omit explicitly mentioning the bias terms and we shall assume that $\mathbf{W}$ represents the whole parameter set. In fact, biases can be considered as extra nodes with activation fixed at $+1$.) and hence it can be considered as an instance of a supervised learning problem where, given a training set $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, we aim at finding the weight vector $\mathbf{w}$ which best guarantees that the predicted values $\hat{\mathbf{y}}_i = net(\mathbf{x}_i; \mathbf{w})$ match the expected values $\mathbf{y}_i$ with minimal loss.

There are essentially four components which influence the specification of the model:

- The topology of the network, i.e., the number of layers and the connections among neurons which specify the $\prec$ partial order among nodes;
- The learning algorithm;
- The choice of the activation functions to exploit within the nodes;
- The loss function to minimize.

We shall discuss the topology of the next section and in the following subsections we shall examine the remaining aspects.

Gradient Descent and Backpropagation

Within the traditional machine learning jargon, the learning phase takes place by computing the value of the weights in the network, and it is referred as *training* of the network. Once the weights has been learnt, the output can be easily determined by applying the network on a specific input (*prediction*). The goal of the training phase is to compute the weights that minimize a given criterion, called *loss* and represented as a function $L(\mathbf{w}; \mathcal{D})$ of the weights given the training set. This is notably an unconstrained optimization problem which has been extensively studied and several characterizations and solution have been proposed (Nocedal and Wright, 2006). Typically, the approach is to work on iterative hill-climbing solutions which, starting from an initial solution $\mathbf{w}^{(0)}$, progressively update the weights in a way that at each step t , we can guarantee $\mathbf{w}^{(t)} < \mathbf{w}^{(t-1)}$. The algorithm stops when we reach a local minimum, i.e., when the improvement is negligible.

The standard approach in the DNN setting is a first-order method called *Gradient Descent* (Rumelhart et al., 1986). Basically, this method minimizes the loss $L(\mathbf{w}; \mathcal{D})$ by updating the model parameters in a direction opposite to the gradient $\nabla_{\mathbf{w}} L(\mathbf{w}; \mathcal{D})$. In its simplest form, the update is performed by considering the gradient w.r.t. the whole training set $\mathcal{D}$, and the update is given by $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} L(\mathbf{w}^{(t)}; \mathcal{D})$, where η is the *learning rate*.

The problem with the gradient descent is the relatively slow convergence to a local minimum. Second-order methods can also be applied (Shepherd, 1997), such as conjugate gradients and quasi-Newton methods. However, these approaches have not become popular, especially in the context of DNNs, due to issues such as numerical stability, poor approximation of the underlying hessian matrix, or the high computational costs associated. Some tricks can still be exploited, as investigated in Martens (2010). However, by far the most widely adopted optimization techniques for neural networks are those based on gradient descent plus heuristics.

Later in this section we shall discuss in details how the gradient descent algorithm is realized within a neural network. However, it is important to get into the details of the loss function first.

Loss functions

Recall that the training phase is meant to find a set $\mathbf{w}$ of weights that guarantee an optimal matching of the between the actual values $\mathbf{y}_i$ and the values $\hat{\mathbf{y}}_i$ predicted by the network, given $\mathbf{x}_i$. A loss function is meant to measure this matching, and to quantify how far we are from the theoretical optima. Typically, it can be defined incrementally,

$$L(\mathbf{w}; \mathcal{D}) = \sum_{i=1}^n \ell(\mathbf{y}_i, \hat{\mathbf{y}}_i)$$

by relying on a pointwise specification $\ell(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ of the loss for each data given point. In order to be suitable for the gradient descent algorithm and its variants, L has to be differentiable. Several alternative formulations can be devised, depending on the nature of the problem we are modeling. An extensive review be found in Rosasco et al. (2004). Here we review some major instantiations typically used in DNN realizations.

Numeric prediction. When $\mathbf{y}$ represents a real-valued vector, the simplest losses can be expressed in terms of the difference between the expected and the predicted values, either by means of the *Absolute Error*

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^M |y_i - \hat{y}_i|$$

or *Squared Error*

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^M (y_i - \hat{y}_i)^2$$

Particular cases can be adapted to model more complex situations. For example, when predicting one-dimensional count data, the *Poisson* loss can be applied:

$$\ell(y, \hat{y}) = (\hat{y} - y \log \hat{y})$$

Classification. When $\mathbf{y}$ represents a nominal value, typically both $\mathbf{y}$ and $\hat{\mathbf{y}}$ can be expressed as binary vectors and the absolute loss, also named 0 – 1 loss, can be applied. However, the predicted value $\hat{\mathbf{y}}$ are also likely to get an probabilistic interpretation: that is the output of a network can be interpreted as the response of a multinomial distribution across all the possible class values. Typically, $\mathbf{y}$ is a binary vector and $\hat{\mathbf{y}} \in (0, 1)^D$ is a numeric vector where each $\hat{y}_i$ represents the probability that the response associated to $\mathbf{x}$ is the i -th class. In such a case, the classification loss can be expressed in terms of *Cross Entropy*,

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^D y_i \log \hat{y}_i$$

which represents the log-likelihood of the data under the assumption that the response is, indeed, multinomial. Intuitively, cross entropy provides an estimate of the divergence between two probability distributions, if the cross entropy is large, which means that the difference between two distribution is large, while if the cross entropy is small, which means that two distribution is similar to each other.

A particular case can be devised with a binary classification problem: the true output can be expressed by means of a single binary variable y and the cross entropy loss can be expressed as

$$\mathcal{L}(y, \hat{y}) = (y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

Gradient descent optimization algorithms

We can now review the gradient descent algorithm and its realization in the context of neural networks, called *Backpropagation*. We discussed on how the gradient descent relies on the computation of the gradient $\nabla_{\mathbf{w}} \mathcal{L}$. It is useful to analyze in detail this gradient on specific nodes:

$$\frac{\partial \mathcal{L}}{\partial w_{u,v}} = \sum_{i=1}^n \frac{\partial L_i}{\partial w_{u,v}} \quad (3)$$

where L_i is a placeholder for $\mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$. Remember that $\hat{\mathbf{y}}_i = \text{net}(\mathbf{x}_i; \mathbf{w})$ is a function of the weights, which depends on the weight $w_{u,v}$ only via the input summation a_u , by virtue of the Eqs. (1) and (2). Thus, by applying the chain rule, we have

$$\frac{\partial L_i}{\partial w_{u,v}} = \frac{\partial L_i}{\partial a_u} \frac{\partial a_u}{\partial w_{u,v}}$$

By looking at Eq. (2) we can see that the second term simply corresponds to z_v . We can hence focus on the first term, which we shall denote as δ_u . We can distinguish two situations here:

- u is an output node. In such a case, the computation simplifies into

$$\delta_u = \frac{\partial L_i}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial a_u}$$

For example, in the case of 0–1 loss we have $\delta_u = 2(y_i - \hat{y}_i)z'_u$, where z'_u represents the derivative of f_u applied to a_u .

- u is an internal node. In such a case, we can backpropagate the effects of the gradient from the nodes directly connected to u , by applying again the chain rule:

$$\delta_u = \sum_{k: u < k} \frac{\partial L_i}{\partial a_k} \frac{\partial a_k}{\partial a_u} = \sum_{k: u < k} \delta_k z'_u w_{k,u}$$

As we can see, the computation of the gradient has a recursive specification, which starts from the output nodes and propagates backward to the internal nodes. To summarize, the contribution of each tuple to the update of the weights is given by the gradient $\partial L_i / \partial w_{u,v} = \delta_u z_v$, computed in two steps: first, a forward step is accomplished to compute values a_u , z_u and z'_u for each node in the network; second, these values are propagated backwards by means of the δ_u components.

Equipped with the above backpropagation formulas, Vanilla (or Batch) Gradient Descent method, described before in this section, is an intuitive and easy to train a network. However, it relies on the contributions of all tuples in the computation of the gradient, as shown in Eq. (3). This can be very slow or even unfeasible, if the whole amount of data cannot load in main memory. More practical variants of the algorithm have been proposed in the literature (Ruder, 2016). We briefly review some of them. *Stochastic Gradient Descent (SGD)*. This is probably the simplest variant of the GD algorithm. With this method, the update of the weights does not require the gradient on the full dataset: rather it focuses only on a random subset of reduced size, or even a single random tuple. This trick allows to speed-up the convergence. The adoption of a mini-batch typically introduces high variance in the updates, which can be mitigated by adding a *Momentum* (Qian, 1999), i.e., a smoother update which also takes into account the previous update:

$$\mathbf{v}^{(t+1)} = \gamma \mathbf{v}^{(t)} - \eta \nabla_{\mathbf{w}} L_i(\mathbf{w}^{(t)})$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \mathbf{v}^{(t+1)}$$

Adaptive learning rates. Both the vanilla and the stochastic version of the GD assume that the learning rate η is fixed for each weight component $w_{u,v}$. By contrast, Adagrad (Duchi et al., 2011) adapts the learning rate component-wise, by rescaling a fixed value proportionally to the sum of the magnitudes of the past component-wise gradients $\sum_t (\nabla_{w_{u,v}} L_i(\mathbf{w}^{(t)}))^2$. This has the natural effect of decreasing the effective step size as a function of time, and the principle is that the updates should be larger for infrequent parameters, and smaller for frequent ones.

Adadelta, RMSprop and Adam are variations of the original method, where different weighting factors are exploited instead of exploiting the sum of the magnitudes of the past component-wise gradients.

Activation Functions

An activation function f_u relative to a network unit u is used to encapsulate non linearity into the network. Non-linear activation functions enable neural networks to approximate arbitrarily complex functions (Cybenko, 1989). By contrast, without the non-linearity introduced by the activation function, multiple layers of a neural network are equivalent to a single layer neural network.

The most common forms of activation functions are the *sigmoids*, which are monotonically increasing functions that asymptotes to some finite value. The *logistic* function $\sigma(a) = (1 + \exp(-a))^{-1}$ is such an example, representing a continuous approximation of *sign* function. Another typical choice is the *hyperbolic function* $\tanh(a) = (\exp(a) - \exp(-a))(\exp(a) + \exp(-a))^{-1}$, which is preferable to the logistic for two main reasons (Le Cun et al., 1998): the derivatives are usually higher, thus allowing faster convergence, and the output space is broader and exhibits an average around zero.

One of the drawbacks of the above mentioned functions, especially in the context of deep networks, is that the gradient near the asymptotes is 0. During backpropagation through the network with sigmoid activation, the gradients in neurons whose output is near the asymptotes are nearly 0. Thus, the weights in these neurons do not update. Also, because of the backpropagation rule, the weights of neurons connected to such neurons are also slowly updated. This problem is also known as vanishing gradient. To alleviate this problem, alternative formulations were adopted. For example the *Rectified Linear Unit ReLu* ($a) = \max(0, a)$) has become popular in deep learning models because significant improvements of classification rates have been reported for speech recognition and computer vision tasks. This function only allows the activation if a neurons output is positive; and allows the network to compute much faster than a network with sigmoid or hyperbolic tangent activation functions, and allows sparsity on the network because random initialization approximately guarantees that half of the neurons in the entire network will be set to zero. Although it is not differentiable around 0, it allows a smooth approximation $\text{ReLU}(a) = \log(1 + \exp(a))$.

Previously in this article, we devised a network where each neuron can admit a different activation function. Although this is in principle possible, the common practice is that all nodes within a layer are equipped with the same activations and it's not unusual that the whole network is equipped with a unique activation function. The only exception is given by output nodes, which require specific activations. Typically, these nodes are equipped with linear activations for modeling numeric responses, or with the softmax function for modeling nominal response over k alternative unordered classes: $\text{softmax}_j(\mathbf{a}) = \exp(a_j) / (\sum_k \exp(a_k))^{-1}$.

Shallow Versus Deep Learning

Compared to shallow networks, DNNs suffered from two main issues: overfitting and training time. Overfitting is essentially due because DNNs represents extremely complex models with millions of parameters which are hence difficult to tune and optimize. An effective control of this phenomenon has been obtained by means of several tricks. We mention some of them and the interested reader can refer to Bengio (2012) and Goodfellow et al. (2016).

- *Regularization* allows to control overfitting by adding weight penalties within the loss function according to some criteria (Bengio et al., 2013), such as weight decay or sparsity.
- Other forms of regularization include *Dropout* (Hinton et al., 2014), i.e., resetting a random number of weights during training. Dropout also helps in decorrelating nodes within the network.
- *Gradient clipping* (Bengio et al., 2013) is another form of regularization, usually exploited to prevent the *gradient exploding* problem (the opposite of vanishing gradient).
- *Initialization and pretraining* by means of unsupervised techniques was also proved effective in the early stages of deep learning research (Hinton et al., 2006; Bengio et al., 2007; Erhan et al., 2010). Recently, better random initialization methods (Glorot and Bengio, 2010) were proved more effective, especially when combined with the adoption of rectified linear units.
- *Data augmentation*, i.e., artificially enlarging the training data by transforming the $\mathbf{x}_i$ inputs in the training set.

The problem of training a network with millions of parameters is also due to the computational time required by training procedure. The adoption of GPUs however has produced significant speedups and consequently has made the training phase more affordable.

Deep Learning Architectures

DL gathers features several layers of non-linear processing and according stacked into a hierarchical scheme. Despite this common characteristics, the DL architectures can vary according to the learning goals and the type of data to consider.

As seen in the previous section, a typical learning task for DNNs is supervised learning. DNNs for supervised learning allow to discover discriminative patterns for classification purposes and assume that labelled data are always available in direct or indirect forms. DNNs for supervised tasks are also called discriminative deep networks. Convolutional Neural Networks and Recurrent Neural Networks are relevant deep architectures for supervised tasks.

However, DNNs can also be used for unsupervised or generative learning, when no or little information about class labels is available. These architecture are aimed at extracting high-order correlation from data for pattern analysis or summarization purposes. Relevant architectures for these tasks are Autoencoders, (Restricted) Boltzmann Machines and Deep Belief Networks.

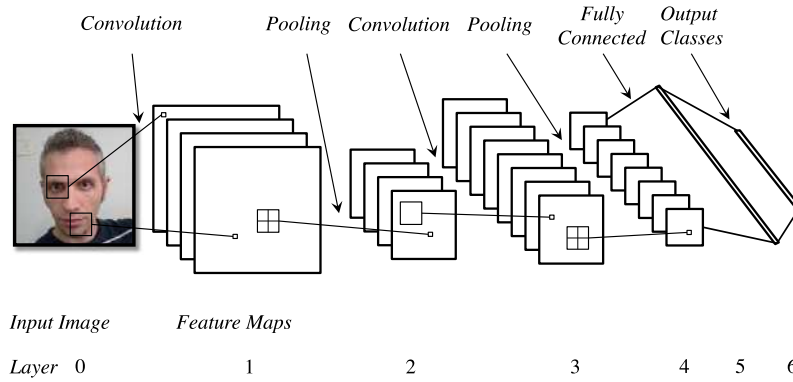


Fig. 10 Example CNN architecture.

Convolutional Neural Networks

Convolutional Neural Networks (CNN) (Le Cun *et al.*, 1990; Le Cun and Bengio, 1995) represent a biologically-inspired variant of feed-forward networks, where the connectivity between neurons tend to capture the invariance of patterns to distortion or shift in the input data. CNNs architectures commonly assume to work with bi-dimensional data (typically images) as input of the networks. A basic CNN can be devised as a stacking of layers where each of these layers transforms one volume of activations to another. Three main types of layers are used: *Convolutional Layers*, *Pooling Layers*, and *Fully-Connected Layers*. Fig. 10 shows an example CNN architecture that combines these layers to recognize objects within an image.

A convolutional layer produces an higher-level abstraction of the input data, called a *feature map*. Units in a convolutional Layer are arranged in feature maps, within which each unit is connected to local regions in the feature maps of the previous layer and represent a convolution of the input. Each neuron represents a receptive field, which receives as input a rectangular section (a filter) of the previous layer and produces an output according to the stimuli received from this filter. In practice, the activation of the (j,k) -th neuron in a feature map at layer h is given by the application of a filter of size $c \times d$ to the neurons of layer $h - 1$:

$$a_{j,k}^{(h)} = \sum_{l=1}^c \sum_{m=1}^d w_{m,l} z_{j+l,k+m}^{(h-1)}$$

Notice that weights in a feature map are shared. This allows to learn exactly a same feature, independently from the position of the feature within the input layer. Also, compared to fully connected layers, the number of parameters of convolutional layers is greatly reduced, since it only corresponds to the size of the filter multiplied by the number of feature maps we wish to detect.

Pooling layers merge semantically similar features, in order to progressively reduce the spatial size of the representation and the amount of parameters and computation in the network. A typical pooling unit computes an aggregation (e.g., the maximum or the average) of the values a local region within a feature map of a convolutional layer, as shown in figure.

The intuition within the architecture of a CNN is that convolutional/pooling layers detect high-level features within the input, which are hence used as input for a set of fully connected layers devoted to output the final classification. For example, within an image, convolutional layers can progressively detect edges, contours and borders, and the later can be finally used to recognize the object within the image.

Recurrent Neural Networks

The (D)NN models explored so far have no memory and the output for a given input does not assume a temporal dependency from the previous inputs to the network. However, the basic neural network model is flexible enough to model even such dependencies, which typically occur within sequences or time series.

In their most general form, Recurrent Neural Networks (RNNs) (Lipton *et al.*, 2015; Graves, 2012) are nonlinear dynamical systems mapping input sequences into output sequences. RNNs keep internal memory for allowing temporal dependencies to influence the output. In a nutshell, in these networks some intermediate operations generate values that are stored internally in the network and used as inputs to other operations in conjunction with the processing of a later input. Therefore RNNs receive as input two sources, the present and the recent past, which are combined to determine how they respond to new data. Fig. 11 clarifies these relationships.

Basically, RNNs can be thought as feed-forward neural networks featured by edges that join subsequent time steps, including the concept of time within the model. Like feedforward neural networks, these learning architectures do not exhibit cycles among the standard edges but, on the contrary, those that connect adjacent time steps (named *recurrent edges*) may form cycles, including cycles of length one that are self-connections from a node to itself across time. At time t , nodes with recurrent edges receive input from the current data point $\mathbf{x}^{(t)}$ and also from hidden node values $\mathbf{h}^{(t-1)}$ representing the network's

previous state. The output $\mathbf{y}^{(t)}$ at each time t is computed by exploiting as input the hidden node values $\mathbf{h}^{(t)}$ at time t . Input $\mathbf{x}^{(t-1)}$ at time $t-1$ can influence the output at time t and later by way the recurrent connections. The figure shows that the recursive representation on the left is a placeholder for the unfolded representation on the right, where the following recursive relationships hold:

$$\mathbf{h}^{(t)} = f_1(\mathbf{U}\mathbf{x}^{(t)} + \mathbf{W}\mathbf{h}^{(t-1)})$$

$$\mathbf{y}^{(t)} = f_2(\mathbf{V}\mathbf{h}^{(t)})$$

By considering the unfolded representation, it is immediate to notice that a RNN corresponds to a DNN where the depth is given by the sequence length, where the weights are shared among layers and the hidden layers $\mathbf{h}^{(t)}$ are connected to each other to capture temporal relationships within the data.

Learning of the weights can proceed by means of the standard backpropagation algorithm, which is instantiated to the case of the unfolded architecture shown in Fig. 11 and is called *Backpropagation Through Time* (Werbos, 1990).

An important issue with standard RNNs is the difficulty of learning long-range dependencies between data instances that are far from each other. This is due essentially to the deep nature of a RNN, which suffers of the already mentioned vanishing gradient (Pascanu et al., 2013).

To approach this issue, a different architecture was established and proven extremely successful: Long-Short Term Memory (LSTM, Hochreiter and Schmidhuber, 1997). Specifically, this model is an evolution of a standard RNN where each hidden layer is replaced by a *memory cell*, shown in Fig. 12. A memory cell contains a node with a self-connected recurrent edge, ensuring that the gradient can pass across many time steps without vanishing.

The idea behind LSTMs is to enable long-range dependencies by introducing an intermediate storage step within the memory cell, controlled by special neurons called gates. By assuming that a memory cell is associated with a cell state $C^{(t)}$, the gates can add or remove information to the cell state. Basically, gates are sigmoidal units associated with a pointwise multiplication operator, which optionally let information flow through the layer. In particular, the *input gate* $i^{(t)}$ controls how $\mathbf{x}^{(t)}$ contributes to the cell state, the *forget gate* $f^{(t)}$ controls the amount of information to gather from the previous cell state $C^{(t-1)}$, and the *output gate* $o^{(t)}$ controls the generation of the output $\mathbf{y}^{(t)}$ of the node.

Despite their simplicity, LSTMs work surprisingly well in several tasks and either them or their variants represent the most widely adopted application of RNNs.

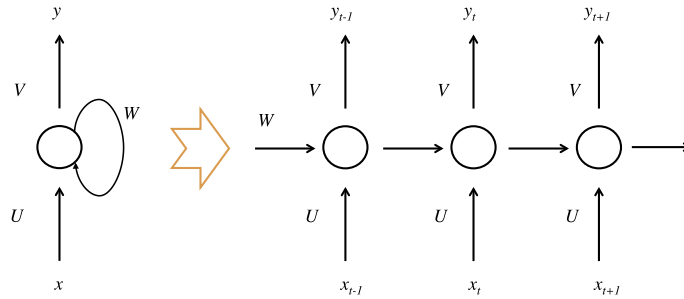


Fig. 11 Simple RNN architecture. The recurrent edge on the left represents the unfolding of the connections through the network following the length of the sequence. Modified from <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>.

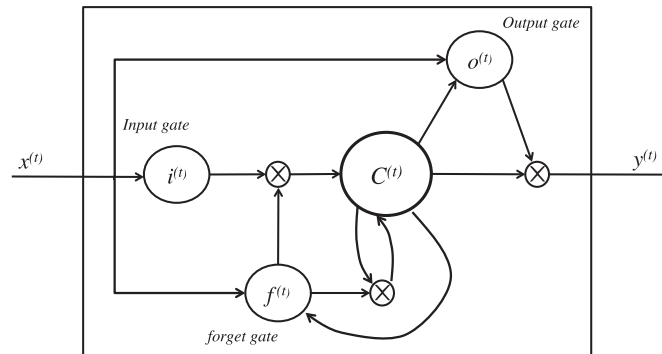


Fig. 12 Memory cell structure.

Autoencoders

The architectures discussed so far represent example models for supervised learning. However, deep architectures for generative learning have been investigated as well. Among them, Energy-Based Models (Ranzato *et al.*, 2006) and Deep Autoencoders (Hinton and Salakhutdinov, 2006; Bengio *et al.*, 2007) gained particular attention in recent years (Bengio, 2009; Ngiam *et al.*, 2011).

An autoencoder (also named autoassociator) is a particular feedforward NN architecture where the target of the network is the data itself, and it is mainly exploited for dimensionality reduction tasks or for learning efficient encoding. Basically, it can be considered as an unsupervised feature extraction method. The network structure of an Autoencoder is similar to a Multi-Layer ANN. The simplest structure (shown in Fig. 13(a)) includes three components: (i) an input layer of raw data to be encoded; (ii) a (stack of) hidden layer(s) featuring by a considerable smaller number of neurons; and (iii) an output layer with the same size (i.e. the same number of neurons) of the input layer. An autoencoder is called deep if the number of its hidden layers is large (Fig. 13(b)).

Specifically, an autoencoder can be conceptually thought as composed by two subnets, namely *encoder* and *decoder*. The former allows to learn a mapping $z = \text{enc}(x)$ between the input layer and the hidden layers (*encoding*), whereas the second learns a mapping $y = \text{dec}(z)$ between the features extracted by the encoder and the output layer (*decoding*). Both mappings are defined in terms of nonlinear activation functions that are governed by a set W of weights,

$$z = f_1(Wx)$$

$$y = f_2(W^T z)$$

Again, gradient descent can be applied to learn W that minimizes the reconstruction loss, for example, $\mathcal{L} = \sum_i \|x_i - y_i\|$ (for real-valued vectors x). Deep autoencoders represent essentially the multi-layered version of this basic model, where multiple intermediate representations can be stacked. Since we aim at minimizing the reconstruction error and it difficult to optimize the weights in nonlinear autoencoders that have multiple hidden layers, a typical strategy in to proceed in a greedy levelwise manner, by progressively learning the weights for each level separately.

An effective variant of this architecture is called (*Stacked*) *Denoising Autoencoder* (Vincent *et al.*, 2008). In a nutshell, the purpose of a denoising autoencoder is to act as a regularizer, by trying to reconstruct the original information from noisy data. To this purpose, noise can be applied stochastically to the input x , to obtain a new input $\tilde{x}$ used to compute z and y . By optimizing the reconstruction loss, the weight matrix W learns to regularize the content and to extract features even from noisy input. In this way, it can be used a pre-training strategy for learning DNNs.

Deep Belief Networks

An alternative approach to autoencoders is to assume that the observed data is the result of a stochastic process which can be interpreted in neural network terms.

A *Boltzmann Machine* (Fischer and Igel, 2012) is a parameterized generative model representing a probability distribution, modeled indeed as a neural network with two basic layers. Input units represent a first layer and corresponds to the components of the observations x . Hidden units are meant to model dependencies between the components of the observations. Fig. 14(a) shows an example where the structure is featured by symmetric and bidirectional edges connecting neurons even belonging to the

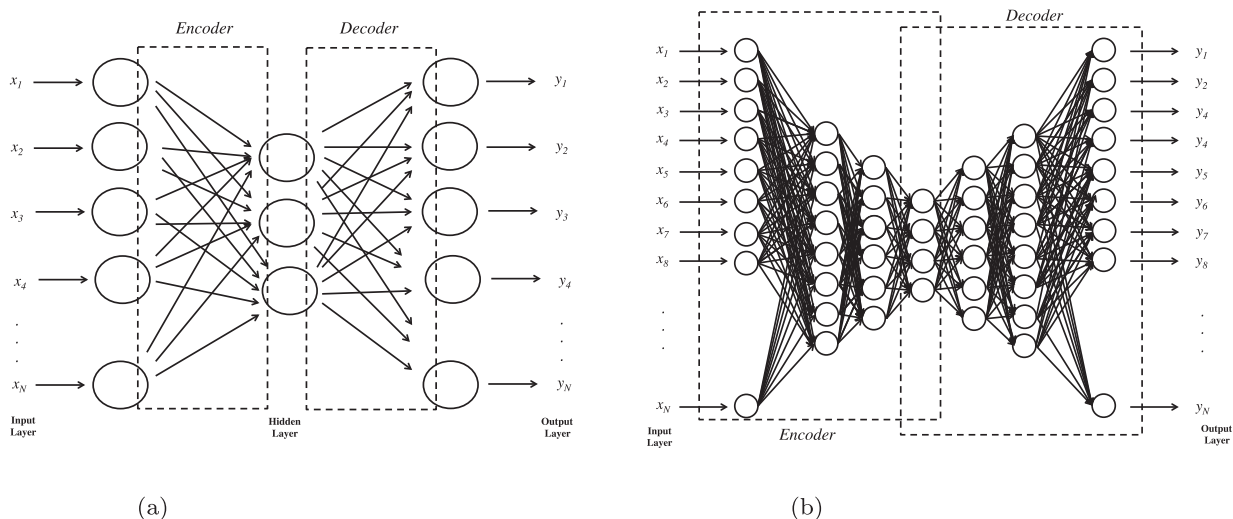


Fig. 13 Autoencoder architecture. (a) Simple Autoencoder. (b) Deep Autoencoder.

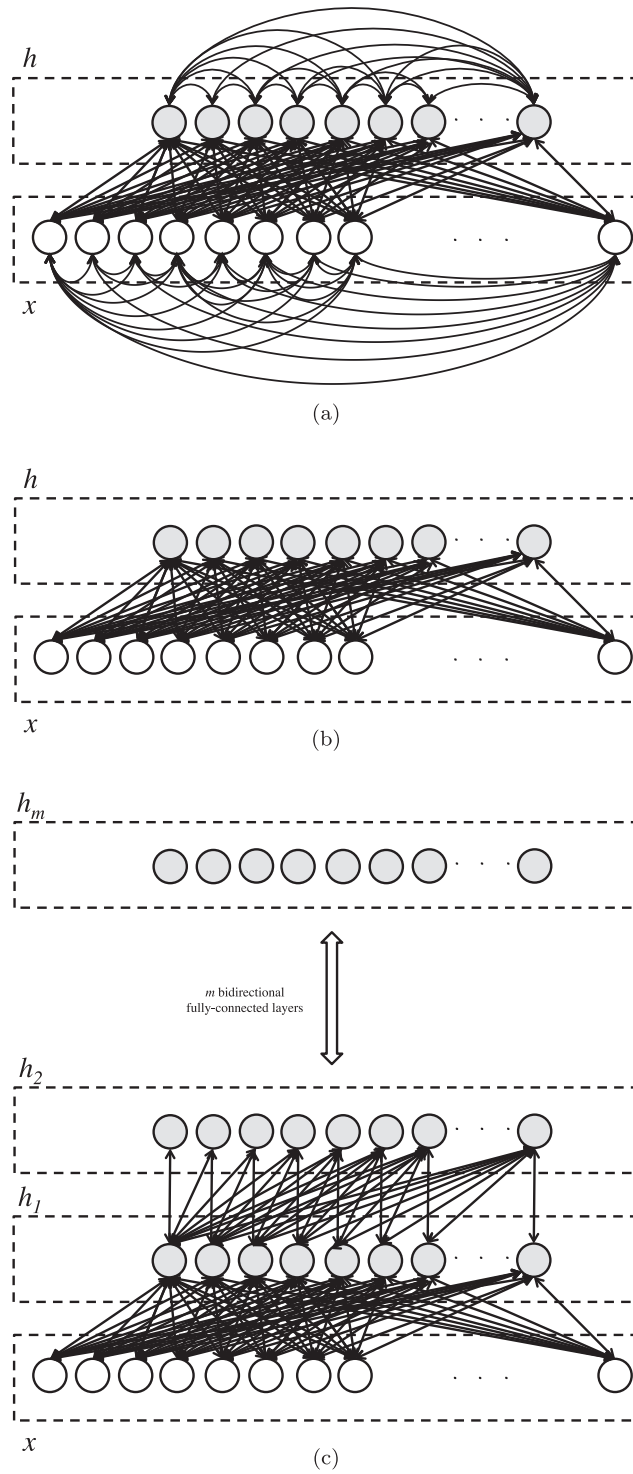


Fig. 14 General/restricted/deep Boltzmann machine architectures. (a) Boltzmann Machine. (b) Restricted Boltzmann Machine. (c) Deep Boltzmann Machine.

same layer. Within the figure, $\mathbf{x}$ represents the input vector, and $\mathbf{h}$ represents the vector corresponding to the hidden units. The network represents the probabilistic model given by the Gibbs distribution

$$p(\mathbf{x}, \mathbf{h}) = \frac{e^{-E(\mathbf{x}, \mathbf{h})}}{Z}$$

where $E(\mathbf{x}, \mathbf{h})$ represents the “energy” of the network, and Z is a normalization constant. In practice, a Boltzmann machine assumes that each possible configuration of the network is associated with a form of energy, so that low-energy configurations are more probable than high-energy configuration. According to this model, it is possible to devise the probability of the input data

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{h}) d\mathbf{h}$$

and to interpret the learning process as a way to adjust the interactions between variables in order to make the network more likely to generate the observed data. Notice that the approach is different from the supervised setting explored in the previous sections, where the objective was to learn $p(\mathbf{y} | \mathbf{x})$. In these respect, Boltzmann machines represent a particular case of belief networks, the latter being a directed acyclic graphs composed of stochastic variables (Koller and Friedman, 2009).

In its general formulation, the energy function $E(\mathbf{x}, \mathbf{h})$ expresses the interactions between the variables in the model:

$$E(\mathbf{x}, \mathbf{h}) = \mathbf{x}^T \mathbf{W} \mathbf{h} + \mathbf{h}^T \mathbf{U} \mathbf{h} + \mathbf{x}^T \mathbf{V} \mathbf{x} + \mathbf{b}^T \mathbf{x} + \mathbf{c}^T \mathbf{h}$$

Here, $\mathbf{W}$ represents the weights associated to the inter-layer interactions: that is, w_{ij} is the weight associated with the connection between the input variable x_i and the hidden variable h_j . Similarly, the matrices $\mathbf{U}$ and $\mathbf{V}$ represent the weights of the intra-layer interactions.

In practice, Boltzmann machines are barely used in this general formulation. The most common form assumes that the dependency graph only connections between the layer of hidden and visible variables but not between two variables of the same layer. In practice, the assumption is that the weights $\mathbf{U}$ and $\mathbf{V}$ are fixed to 0. This simpler version is called *Restricted Boltzmann Machine* (RBM) and it is shown in Fig. 14(b). Interestingly, an RBM with binary hidden variable can be interpreted as a stochastic neural network, where for a given hidden node i the conditional probability given the input can be expressed as $p(h_i = 1 | \mathbf{x}) = \sigma(\sum_j w_{ij} x_j + c_i)$. Also, restricting the connectivity only to inter-layer connection makes the learning of a Boltzmann machine easier.

In general, the learning of a (R)BM is accomplished by optimizing the log-likelihood

$$L(\theta; \mathcal{D}) = \sum_i \log p(\mathbf{x}_i)$$

with respect to the parameter set $\theta = \{\mathbf{W}, \mathbf{U}, \mathbf{V}, \mathbf{b}, \mathbf{c}\}$. However, computing such a likelihood or its gradient is computationally intensive. Fortunately, the simplified form of an RBM allows the adoption of sampling methods to approximate the gradient through an efficient algorithm called *Contrastive Divergence* (Hinton, 2002).

Deep Boltzmann Machines (DBM) represent a straightforward generalization of the RBM model (Hinton and Salakhutdinov, 2006; Hinton, 2007; Goodfellow et al., 2013), where several layers of hidden variables are stacked and no connection is established between neurons belonging to the same layer. Fig. 14(c) illustrates the general architecture for these networks. Once again, training these deep networks can be accomplished levelwise, by freezing the previously learned layer and re-applying the learning procedure to the next layer. Interestingly, the network corresponding to a DBM can be extended with an output layer. This way, the weights learned so far can be used as initial weights for the supervised training of the resulting NN.

Closing Remarks

The research on DL has received an extraordinary attention in the last decade. The combination of computational power, new methods for training and new architectures delivered impressive results in a wide range of applications.

Despite these impressive results, there are still several open issues. Tuning these networks is extremely complicated and, although several progress were made by the research in the field, the problem of properly interpreting the behavior of a network and of the hidden units is still open.

This survey did not cover several of these aspects, as well as several other promising directions in the current literature. In particular, reinforcement learning, or adversarial learning are themes of interest which can help improve the performance of these conceptually simple yet extremely powerful models. The interested reader can refer to the references suggested throughout this article as a starting point for continuing the journey into this field.

See also: Algorithms for Graph and Network Analysis: Graph Alignment. Artificial Intelligence and Machine Learning in Bioinformatics. Artificial Intelligence. Data Mining in Bioinformatics. Knowledge and Reasoning. Machine Learning in Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Stochastic Methods for Global Optimization and Problem Solving. The Challenge of Privacy in the Cloud

References

- Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J., 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* 33, 831–838.
- Bengio, Y., 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2, 1–127.
- Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*. Berlin; Heidelberg: Springer, pp. 437–478.

- Bengio, Y., Boulanger-Lewandowski, N., Pascanu, R., 2013. Advances in optimizing recurrent networks. In: *Procs. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8624–8628.
- Bengio, Y., Pascal, L., Dan, P., Larochelle, H., 2007. Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems* 19, 153–160.
- Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013 – Proceedings of the 16th International Conference, Nagoya, Japan, September 22–26, 2013, Proceedings, Springer*, pp. 411–418.
- Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2, 303–314.
- Deng, L., Yu, D., 2014. *Deep Learning: Methods and Applications*. Hanover, MA: Now Publishers Inc.
- Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, 2121–2159.
- Erhan, D., Bengio, Y., Courville, A., *et al.*, 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11, 625–660.
- Esteva, A., Kuprel, B., Novoa, R.A., *et al.*, 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542, 115–118.
- Fischer, A., Igel, C., 2012. An introduction to restricted boltzmann machines. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, CIARP 2012*, pp. 14–26.
- Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: *International Conference on Artificial Intelligence and Statistics*, pp. 249–256.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. Cambridge, MA: The MIT Press.
- Goodfellow, I., Mirza, M., Courville, A., Bengio, Y., 2013. Multi-prediction deep boltzmann machines. *Advances in Neural Information Processing Systems* 26, 548–556.
- Graves, A., 2012. *Supervised Sequence Labelling With Recurrent Neural Networks*. Studies in Computational Intelligence, vol. 385. Springer.
- Gulshan, V., Peng, L., Coram, M., *et al.*, 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Journal of the American Medical Association* 316, 2402–2410. Available at: <https://doi.org/10.1001/jama.2016.17216>.
- Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation*, second ed. Upper Saddle River, NJ: Prentice Hall PTR.
- Hinton, G., 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation* 13, 1771–1800.
- Hinton, G., 2007. Learning multiple layers of representation. *Trends in Cognitive Sciences* 11, 428–434.
- Hinton, G.E., Osindero, S., Teh, Y., 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 1527–1554.
- Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1929–1958.
- Hinton, G., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 504–507.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Computation* 9, 1735–1780.
- Koller, D., Friedman, N., 2009. *Probabilistic Graphical Models*. The MIT Press.
- Le Cun, Y., Bengio, T., 1995. Convolutional networks for images, speech, and time series. In: Arbib, M.A. (Ed.), *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, pp. 255–258.
- Le Cun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444.
- Le Cun, Y., Boser, B., Denker, J.S., *et al.*, 1990. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems* 2, 396404.
- Le Cun, Y., Bottou, L., Orr, G.B., Müller, K., 1998. Efficient backprop. In: *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, Springer-Verlag, London, pp. 9–50.
- Lipton, Z., Berkowitz, C., Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. CoRR abs/1506.00019. Available at: <http://arxiv.org/abs/1506.00019>; arXiv:1506.00019.
- Liu, W., Wang, Z., X, L., *et al.*, 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234, 11–26.
- Martens, J., 2010. Deep learning via hessian-free optimization. In: *Proceedings of the 27th International Conference on Machine Learning*, pp. 735–742.
- Mitchell, T.M., 1997. *Machine Learning*, first ed. New York, NY: McGraw-Hill, Inc.
- Ngiam, J., Khosla, A., Kim, M., *et al.*, 2011. Multi-modal deep learning. In: *Proceedings of the 28th International Conference on Machine Learning*, pp. 689–696.
- Ning, F., Delhomme, D., Le Cun, Y., *et al.*, 2005. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing* 14, 1360–1371.
- Nocedal, J., Wright, S., 2006. *Numerical Optimization*, second ed. New York, NY: Springer.
- Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning*, vol. 28, pp. III-1310–III-1318.
- Patterson, J., Gibson, A., 2017. *Deep Learning a Practitioners Approach*. Newton, MA: O'Reilly Media, Inc.
- Prieto, A., Prieto, B., Martínez-Ortíz, E., *et al.*, 2016. Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing* 214, 242–268.
- Qian, N., 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks* 12, 145–151.
- Ranzato, M., Poultney, C., Chopra, S., Le Cun, Y., 2006. Efficient learning of sparse representations with an energy-based model. In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pp. 1137–1144.
- Rizzo, R., Fiannaca, A., La Rosa, M., Urso, A., 2015. A deep learning approach to DNA sequence classification. In: *Computational Intelligence Methods for Bioinformatics and Biostatistics – Proceedings of the 12th International Meeting, CIBB 2015, Naples, Italy, September 10–12, 2015, Revised Selected Papers*, pp. 129–140.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 – Proceedings of the 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Springer*, pp. 234–241.
- Rosasco, L., De Vito, E.D., Caponnetto, A., Piana, M., Verri, A., 2004. Are loss functions all the same? *Neural Computation* 15, 1063–1076.
- Ruder, S., 2016. An overview of gradient descent optimization algorithms. CoRR abs/1609.04747. Available at: <http://arxiv.org/abs/1609.04747>; arXiv:1609.04747.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* 1, 318–362.
- Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural Networks* 61, 85–117.
- Shepherd, A., 1997. *Second-Order Methods for Neural Networks*. London: Springer-Verlag.
- Sze, V., Chen, Y., Yang, T., Emer, J.S., 2017. Efficient processing of deep neural networks: A tutorial and survey. CoRR abs/1703.09039. Available at: <http://arxiv.org/abs/1703.09039>.
- Thorpe, S., Fabre-Thorpe, M., 2001. Seeking categories in the brain. *Science Magazine* 12, 260–263.
- Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P., 2008. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103.
- Werbos, P., 1990. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*. 1550–1560.

Introduction to Biostatistics

Antonella Iuliano and Monica Franzese, Institute for Applied Mathematics “Mauro Picone”, Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biostatistics is a branch of applied statistics with applications in many areas of biology including epidemiology, medical sciences, health sciences, educational research and environmental sciences. The principles and methods of statistics, which is the science that deals with the collection, classification, analysis, and interpretation of numerical data for the purpose of data description and decision making, are applied to the biological areas. The first application of statistics appeared during the seventeenth century in political science to describe the various aspects of the affairs of a government or state (hence the term “statistics”). At the same time, the development of probability theory, thanks to the contribution of many mathematicians, such as Blaise Pascal (1623–1662), Pier Fermat (1601–1665), Jacques Bernoulli (1654–1705) and others, has provided the basis for the modern statistics. However, the first scientist to introduce biostatistics concepts was the astronomer and mathematician Adolphe Quetelet (1796–1874), who in his work combined the theory and practical methods of statistics in biological, medical and sociological applications. Later, Francis Galton (1822–1911) tried to solve the problem of heredity on the basis of Darwin’s genetic theories with the statistics. In particular, Galton’s contribution to biology was the application of statistics to the analysis of biological variation, using correlation and regression techniques. For this reason, he has been defined as the father of biostatistics and his methodology has become the basis for the use of statistics in biology. Karl Person (1860–1906) continued in the tradition of Galton’s theory contributing significantly to the field of biometrics, meteorology, theories of social Darwinism and eugenics. The dominant figure in biostatistics during the twentieth century was Ronald Fischer (1890–1962), who used mathematics to combine Mendelian genetics and natural selection. In particular, he developed the analysis of variance (ANOVA) to analyze large amount of biological data. Today, statistics is an active field whose applications touch many aspects of biology and medicine. In particular, we can distinguish two types of statistical approaches, which aim to provide precise conclusions and significant information from a set of data collected during a biological experiment. The first approach is called descriptive statistics and it is used to analyze a collection of data without assuming any underlying structure for such data ([Spriestersbach et al., 2009](#)), while the second one, called inferential statistics, works on the basis of a given structure for the observed data and involves hypothesis testing to draw conclusions about a population when only a part of the data is observed ([Altman and Krzywinski, 2017](#); [Gardner and Altman, 1986](#)). In fact, when a biologist conducts an experiment, he must make sure that the possible conclusions are statistically significant. In addition, the necessity to perform long and laborious arithmetic computations, as part of the statistical analysis of data with the use of computers, has contributed to improve the quality of the data and the interpretation of the results. In fact, the large amount of available statistical software programs, such as the R Project for Statistical Computing, SAS, SPSS and others, have further revolutionized statistical computing in the field of bioinformatics and computational biology.

The aim of this work is to provide statistical concepts that help biologists to correctly prepare experiments, verify conclusions and properly interpret results. We first introduce several descriptive statistical techniques for organizing and summarizing data, and then we discuss some procedures to infer the population parameters using the data contained in a sample that has been drawn from that population. Finally, an illustrative example is analyzed to give a general understanding of the nature and relevance of biostatistics in clinical research.

Statistical Analysis

The descriptive analysis is the starting point in any applied research providing a numerical summary of the collected data ([Spriestersbach et al., 2009](#)). The main steps of a scientific investigation are: collection of data, organization and visualization of data, calculation of descriptive statistics, and interpretation of statistics (see [Fig. 1](#)). Such data are usually available from one or more sources, such as routinely kept records (hospital medical records or hospital accounting records), surveys (questionnaires and interviews), experiments (treatment decision to investigate the effects of the assigned therapy, treatment and control groups), clinical trials (to test efficacy or toxicity of a treatment with respect to control group) and external sources (published reports or data banks). The set of all elements in a data is known as statistical population, while a sample consists of one or more observations extracted from the population. Each member (or element) of the data under investigated is called statistical unit. A statistical variable is each aspect or characteristic of the statistical unit that is considered for the study. A statistical variable can be qualitative or quantitative, depending on whether their nature is countable or not. Quantitative variables can be characterized as discrete or continuous. Examples of discrete variables are the number of daily admissions to a general hospital or the number of decayed, while examples of continuous variables are the diastolic blood pressure, the heart rate, the heights of the adult males or the ages of patients. On the other hand, qualitative or categorical variables involve observations that can be grouped into categories. In particular, these data can be statistically divided into three groups: nominal (when exist a natural ordering among the

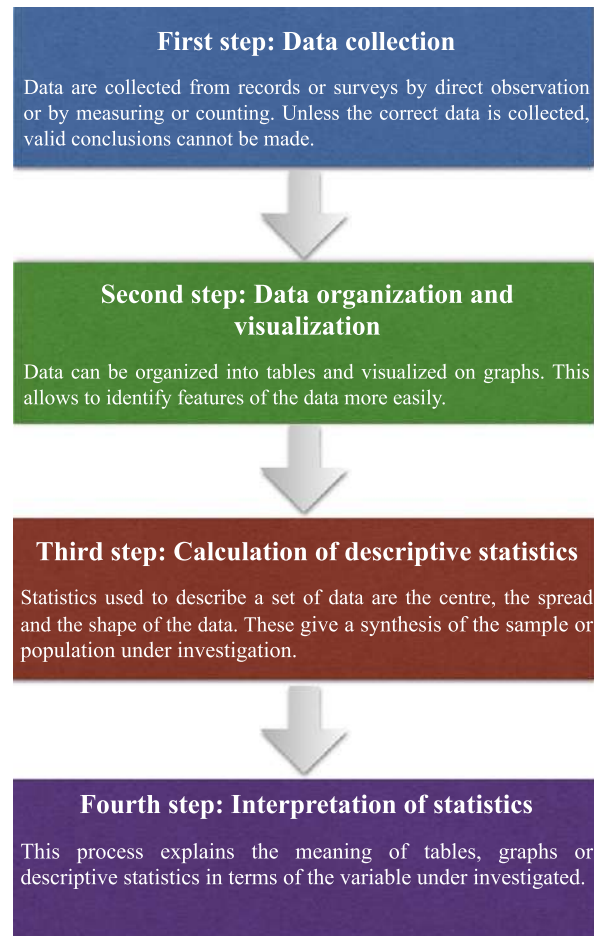


Fig. 1 Main steps of a statistical investigation.

categories), binary or dichotomous (when there are only two possible levels), and ordinal data (when there is a natural order among the categories). Examples of categorical variables involve the sex (male or female), the genotype (AA, Aa, or aa), or the ankle condition (normal, sprained, torn ligament, or broken).

Sometimes in statistics, we use the word measurement (or measurement scale) to categorize and quantify variables. In particular, a measurement is the assignment of numbers to objects or events according to a set of rules. Different measurement scales are distinguished on the relationships assumed to exist between object having different scale values. The most important types of measurement scales are: nominal, ordinal, interval, and ratio. The lowest measurement scale is the nominal scale. It consists of naming observations or classifying them into various mutually exclusive and collectively exhaustive categories. The practice of using numbers to distinguish among the various medical diagnoses constitutes measurement on a nominal scale. When the observations are ranked according to some criterion (from lowest to highest observations), they are said to be measured on an ordinal scale. Convalescing patients may be characterized as unimproved, improved, and much improved. The interval scale is a more sophisticated scale than the nominal and ordinal scale. Here, not only is it possible to order measurements, but also to know the distance between any two measurements. We know that the difference between a measurement of 20 and a measurement of 30 is equal to the difference between measurements of 30 and 40. The ability to do this implies the use of a unit distance and a zero point, both of which are arbitrary. A clear example is the Fahrenheit or Celsius scale. The unit of measurement is the degree of temperature, and the point of comparison is the arbitrarily chosen “zero degrees”, which does not indicate a lack of heat. The interval scale unlike the nominal and ordinal scales is a truly quantitative scale. The highest level of measurement is the ratio scale. This scale is characterized by the fact that equality of ratios as well as equality of intervals may be determined. Fundamental to the ratio scale is a true zero point. The measurement of such familiar traits as height, weight, and length makes use of the ratio scale. Interval and ratio data are sometimes referred to as parametric and nominal and ordinal data are referred to as nonparametric. Parametric means that it meets certain requirements with respect to parameters of the population (normal or bell curve). Parametric data are analyzed using statistical techniques called parametric statistics. Nonparametric data are lacking those same parameters and are investigated by using non-parametric statistics.

Collection, Organization and Visualization of Data

The first step of explaining a biological or biomedical phenomenon is the collection of data under investigated (first step in Fig. 1). Then, the second step is the organization of observed data into tables or data matrix in order to visualize their distribution (second step in Fig. 1). Let N be the number of element or individuals in the population and let X be a random variable assuming the values x_i , for $i = 1, 2, \dots, n$. We denote the number of individuals presenting the value (or characteristic) x_i as n_i . This number n_i is the absolute frequency of the observed value x_i , while the relative frequency f_i of the observed value x_i is the proportion on the total population N of values presenting the value x_i . In symbols, the relative frequency is the ratio

$$f_i = \frac{n_i}{N} \quad (1)$$

for $i = 1, \dots, n$. Note that the sum of the absolute frequencies is equal to the total number of data N and the sum of the relative frequencies is equal to 1:

$$\sum_{i=1}^n n_i = N, \quad \sum_{i=1}^n f_i = 1$$

The observed values x_i , the absolute and relative frequencies are usually organized in tables, called statistical tables or frequency distribution. These tables show the way in which the values of the variable are distributed among the specified characteristics. An example of a univariate statistical table is given in Table 1. When we have a large amount of information, data are grouped into class intervals. A common rule used to choose the number of intervals is to take no fewer than 5 intervals and no more than 15. The better way to decide how many class intervals to employ is to use the Sturges's formula given by $k = 1 + 3.322 \log_{10}(N)$, where k stands for the number of class intervals and N is the number of values in the data set. The width of the class intervals is determined by dividing the range R by k . The range R is given by the difference between the smallest $x_{\min}$ and the largest observation $x_{\max}$ in the data set. The number of values, falling within each class interval, is the absolute frequency n_i , while the proportion of values falling within each class interval is the relative frequency f_i (see Table 2). Starting from the definition of (absolute and relative) frequency distribution, it is possible to calculate the (absolute and relative) cumulative frequency (N_i and F_i , $i = 1, \dots, n$) distribution, which indicates the number of elements in the data set that lie above (or below) a particular value in a data set. For instance, see Tables 1 and 2. Usually, the information contained in these tables can be presented graphically under the form of histograms and cumulative frequency curves. A histogram is a graphical representation of the absolute or relative frequencies for each value of the characteristic or class intervals. It is commonly used for quantitative variables. A cumulative frequency curve is a plot of the number or percentage of individuals falling in or below each value of the characteristic or class intervals. Other types of graphical representations are the pie chart or the bar plot. The first graph is a circular chart divided into sectors, showing the relative magnitudes in frequencies or percentages. The second one, often used to display categorical data, is a

Table 1 Frequency distribution

Values of characteristics x_i	Absolute frequency n_i	Relative frequency f_i	Cumulative absolute frequency N_i	Cumulative relative frequency F_i
x_1	n_1	$f_1 = \frac{n_1}{N}$	$n_1 = N_1$	$f_1 = F_1$
x_2	n_2	$f_2 = \frac{n_2}{N}$	$n_1 + n_2 = N_2$	$f_1 + f_2 = F_2$
...	...	...	...	...
x_i	n_i	$f_i = \frac{n_i}{N}$	$n_1 + n_2 + \dots + n_i = N_i$	$f_1 + f_2 + \dots + f_i = F_i$
...	...	...	...	...
x_k	n_k	$f_k = \frac{n_k}{N}$	$n_1 + n_2 + \dots + n_k = N$	$f_1 + f_2 + \dots + f_k = 1$
Total	N	1		

Source: UF Health – UF Biostatistics. Available at: <http://bolt.mph.ufl.edu/2012/08/02/learn-by-doing-exploring-a-dataset/>.

Table 2 Frequency distribution based on class intervals. The symbol $-|$ means that only the superior limit is included into the class interval

Class intervals $x_i - x_{i+1}$	Absolute frequency n_i	Relative frequency f_i	Cumulative absolute frequency N_i	Cumulative relative frequency F_i
$x_1 - x_2$	n_1	$f_1 = \frac{n_1}{N}$	$n_1 = N_1$	$f_1 = F_1$
$x_2 - x_3$	n_2	$f_2 = \frac{n_2}{N}$	$n_1 + n_2 = N_2$	$f_1 + f_2 = F_2$
...	...	...	...	...
$x_{i-1} - x_i$	n_i	$f_i = \frac{n_i}{N}$	$n_1 + n_2 + \dots + n_i = N_i$	$f_1 + f_2 + \dots + f_i = F_i$
...	...	...	...	...
$x_{n-1} - x_n$	n_n	$f_n = \frac{n_n}{N}$	$n_1 + n_2 + \dots + n_n = N$	$f_1 + f_2 + \dots + f_n = 1$
Total	N	1		

Source: Reproduced from Daniel, W.W., Cross, C.L., 2013. Biostatistics: A Foundation for Analysis in the Health Sciences, tenth ed. John Wiley & Sons.

chart with rectangular bars with lengths proportional to the values they represent. They can be plotted vertically or horizontally. Histogram, pie chart and bar plot are graphs very useful for presenting data in a comprehensible way to a statistical and non-statistical audience.

Descriptive Measures

After the organization and visualization of data, several statistical functions can be computed to describe and summarize the set of data (third step in Fig. 1) and to interpret the obtained results (fourth step in Fig. 1). These functions are called descriptive measures or statistics. There are three general types of statistics: measures of central tendency (or location), measures of dispersion (or variability), and measures of symmetry (or shape). The measures of central tendency convey information about the average value of a data. The most commonly used measures of location are the mean, the median, and the mode (Manikandan, 2011a,b; Wilcox and Keselman, 2003). These descriptive measures are called location parameters, because they can be used to designate specific positions on the horizontal axis when the distribution of a variable is graphed. Let X be a random variable and let x be the observed values of the random variable X . The mean is obtained by adding all the values in a population or sample and dividing by the number of values that are observed. We use the Greek letter μ to stand for the population mean, while we use the symbol $\bar{x}$ to define the sample mean:

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^N x_i \quad (2)$$

The value N indicates the population size, the quantity n is the number of observed values in the sample. Similarly, we can compute the mean for (simple and grouped) frequency distributions. An alternative to the mean is the computation of the median. The median is a numerical value that divides the ordered set of values (from lowest to highest value) into two equal parts, such that the number of values equal to or greater than the median is equal to the number of values equal to or less than the median. If the number of values is odd, the median is the middle value of the ordered set of data. If the number of values is even, the median is the mean of the two middle values. In addition, when the median is close to the mean, then we use as statistics the mean, even if the median is usually the better choice. Similarly, we can compute the median for (simple and grouped) frequency distributions. Other location parameters include percentiles or quartiles. These descriptive measures divide the data set into four equal parts each containing 25% of the total observations. The 50th percentile Q_2 is the median. The 25th percentile is the first quartile Q_1 . The 75th percentile is the third quartile Q_3 . Finally, the mode of a variable is the value that occurs most frequently into the data. The mode may not exist, and even if it does, it may not be unique. This happens when the data set has two or more values of equal frequency, which is greater than the other values. The mode is usually used to describe a bimodal distribution. In a bimodal distribution, the taller peak is called the major mode and the shorter one is the minor mode. For continuous data, the mode is the midpoint of the interval with the highest rectangle in the histogram. If the data are grouped into class intervals, then the mode is defined in terms of class frequencies. The mode is used also for describing qualitative data. For example, suppose that the patients in a mental health clinic, during a given year, received one of the following diagnoses: mental retardation, organic brain syndrome, psychosis, neurosis, and personality disorder. The diagnosis, occurring most frequently in the group of patients, is called modal diagnosis. The computational formulas of location measures are shown in Table 3.

Dispersion measures describe the spread or variation present in a set of data. The most important statistics of variability are the range, the variance, the standard deviation and the coefficient of variation (Manikandan, 2011c). The range R is the difference between the largest and smallest value in a set of observations. The variance and the standard deviation are two very popular measures of dispersion.

The variance is defined as the average of the squared differences from the mean, while the standard deviation is a measure of how the data are spread out across the mean. We use the Greek letter σ^2 to indicate the population variance, while we use the symbol s^2 to define the sample variance:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \mu)^2, \quad s^2 = \frac{1}{n-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (3)$$

The standard deviation is the square root of the variance. The more variation there is into the data, the larger is the standard deviation. The standard deviation is useful as a measure of variation within a given set of data. When two distributions are taken into account and their measures are expressed in different units, compare the two standard deviations may lead to false results. For example, for a certain population we wish to know whether serum cholesterol levels, measured in milligrams per 100 mL, are more variable than body weight, measured in pounds. Therefore, in this case, we use a measure of relative variation rather than absolute variation. Such measure is called coefficient of variation CV , which expresses the standard deviation as a percentage of the mean: it is a unit-free measure. The CV is small if the variation is small and it is unreliable if the mean is near zero. Hence, if we consider two groups, the one with less CV is said to be more consistent.

Another measure of dispersion is the interquartile range (IQR). It is the difference between the third and first quartiles, i.e., $IQR = Q_3 - Q_1$. A large IQR indicates a large amount of variability among the middle 50% of the relevant observations, and a small IQR indicates a small amount of variability among the relevant observations. A useful graph for summarize the information contained in a data set is the box-and-whisker plot. The construction of a box-and-whisker plot makes use of the quartiles of a

Table 3 Summary of the main descriptive statistics used to describe the basic features of the data in a study

Statistics	For population	For sample	For simple frequency distribution	For class frequency distribution
Mean	$\mu = \frac{\sum_{i=1}^N x_i}{N}$ where N is the population size	$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$ where n is the sample size	$\bar{X} = \frac{\sum_{i=1}^n x_i n_i}{n}$ where n_i is the i th absolute frequency	$\bar{X} = \frac{\sum_{i=1}^n c_i n_i}{n}$ where $c_i = \frac{x_{i+1} + x_i}{2}$ are the central values of each class, for $i = 1, \dots, n$
Median	Odd size: $M_e = \frac{N+1}{2}$ th ordered observation X_i Even size: $M_e = \frac{(\frac{N}{2}) + (\frac{N}{2} + 1)}{2}$ th ordered observation X_i	Odd size: $M_e = \frac{n+1}{2}$ th ordered observation x_i Even size: $M_e = \frac{(\frac{n}{2}) + (\frac{n}{2} + 1)}{2}$ th ordered observation x_i	$M_e = x_i$, such that, $F_i \geq 0.50$ F_i is the i th relative cumulative frequency	$M_e \approx x_i + (x_{i+1} - x_i) \frac{0.5 - F_{i-1}}{F_i - F_{i-1}}$, where $F_i \geq 0.50$ F_i is the i th relative cumulative frequency
Quartiles	$Q_1 = \frac{N+1}{4}$ th ordered observation X_i $Q_2 = M_e$ $Q_3 = \frac{3(N+1)}{4}$ th ordered observation X_i	$Q_1 = \frac{n+1}{4}$ th ordered observation x_i $Q_2 = M_e$ $Q_3 = \frac{3(n+1)}{4}$ th ordered observation x_i	$Q_1 = x_i$, such that, $F_i \geq 0.25$ $Q_2 = M_e$ $Q_3 = x_i$, such that, $F_i \geq 0.75$ F_i is the i th relative cumulative frequency	$Q_1 \approx x_i + (x_{i+1} - x_i) \frac{0.25 - F_{i-1}}{F_i - F_{i-1}}$, where $F_i \geq 0.25$ $Q_2 = M_e$ $Q_3 \approx x_i + (x_{i+1} - x_i) \frac{0.75 - F_{i-1}}{F_i - F_{i-1}}$, where $F_i \geq 0.75$
Mode	$M_o = x_i$ is the value that occurs most frequently in the data	$M_o = x_i$ is the value that occurs most frequently in the data	$M_o = \{x_i : n_i = \max\}$, where n_i is the i th absolute frequency	$m_c = \{x_i - x_{i+1} - x_i : h_i = \max\}$, where $h_i = \frac{n_i}{x_{i+1} - x_i}$ is the intensity class $M_o \approx \frac{x_i + x_{i+1}}{2}$
Range	$R = X_{\max} - X_{\min}$	$R = X_{\max} - X_{\min}$	$R = X_{\max} - X_{\min}$	$R = X_{\max} - X_{\min}$
Interquartile range	$IQR = Q_3 - Q_1$	$IQR = Q_3 - Q_1$	$IQR = Q_3 - Q_1$	$IQR = Q_3 - Q_1$
Variance	$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$	$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$	$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2 n_i}{n-1}$	$s^2 = \frac{\sum_{i=1}^n (c_i - \bar{x})^2 n_i}{n-1}$
Deviazione standard	$\sigma = \sqrt{\sigma^2}$	$s = \sqrt{s^2}$	$s = \sqrt{s^2}$	$s = \sqrt{s^2}$
Correlation of variation	$CV = \frac{\sigma}{\mu} \times 100$	$CV = \frac{s}{\bar{x}} \times 100$	$CV = \frac{s}{\bar{x}} \times 100$	$CV = \frac{s}{\bar{x}} \times 100$
Skewness	$\Gamma_1 = \frac{1}{N\sigma^3} \sum_{i=1}^N (x_i - \mu)^3$	$\gamma_1 = \frac{\frac{\sqrt{n}}{(n-1)\sqrt{n-1}s^3} \sum_{i=1}^n (x_i - \bar{x})^3}{n_i}$	$\gamma_1 = \frac{\frac{\sqrt{n}}{(n-1)\sqrt{n-1}s^3} \sum_{i=1}^n (x_i - \bar{x})^3 n_i}{n_i}$	$\gamma_1 = \frac{\frac{\sqrt{n}}{(n-1)\sqrt{n-1}s^3} \sum_{i=1}^n (c_i - \bar{x})^3 n_i}{n_i}$
Kurtosis	$\Gamma_2 = \frac{1}{N\sigma^4} \sum_{i=1}^N (x_i - \mu)^4$	$\gamma_2 = \frac{\frac{n(n+1)}{(n-1)(n-2)(n-3)s^4} \sum_{i=1}^n (x_i - \bar{x})^4}{n_i}$	$\gamma_2 = \frac{\frac{n(n+1)}{(n-1)(n-2)(n-3)s^4} \sum_{i=1}^n (x_i - \bar{x})^4 n_i}{n_i}$	$\gamma_2 = \frac{\frac{n(n+1)}{(n-1)(n-2)(n-3)s^4} \sum_{i=1}^n (c_i - \bar{x})^4 n_i}{n_i}$

data set (see Fig. 2). An outlier is an observation whose value either exceeds the value of the third quartile by a magnitude greater than $1.5 \times (IQR)$ or is less than the value of the first quartile by a magnitude greater than $1.5 \times (IQR)$. Similarly, we can compute the (simple and grouped) dispersion measures for frequency distributions. The computational formulas of variability measures are shown in Table 3.

An attractive property of a data distribution occurs when the mean, median, and mode are all equal. The well-known “bell-shaped curve” is a graphical representation of a distribution for which the mean, median, and mode are equal among them. Much statistical inference is based on this distribution called normal distribution (see Fig. 3). Generally, a data distribution can be classified on the basis of their form (symmetric or asymmetric). A symmetric distribution is a type of distribution where the left side of the distribution mirrors the right side (see Fig. 3). When the left half and right half of the graph of a distribution are not mirror images of each other, the distribution is asymmetric. In this case, the distribution is said to be skewed. In other words,

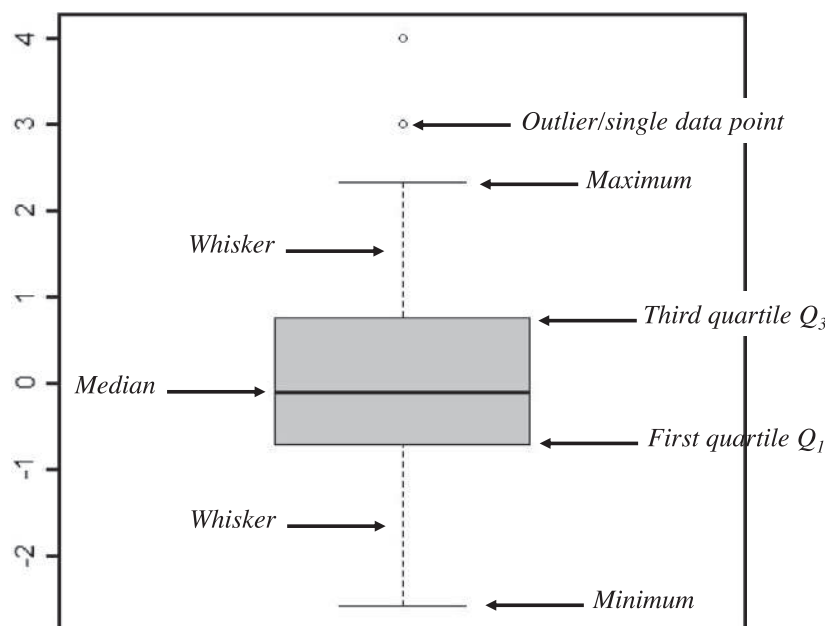


Fig. 2 Boxplot or box plot whisker diagram.

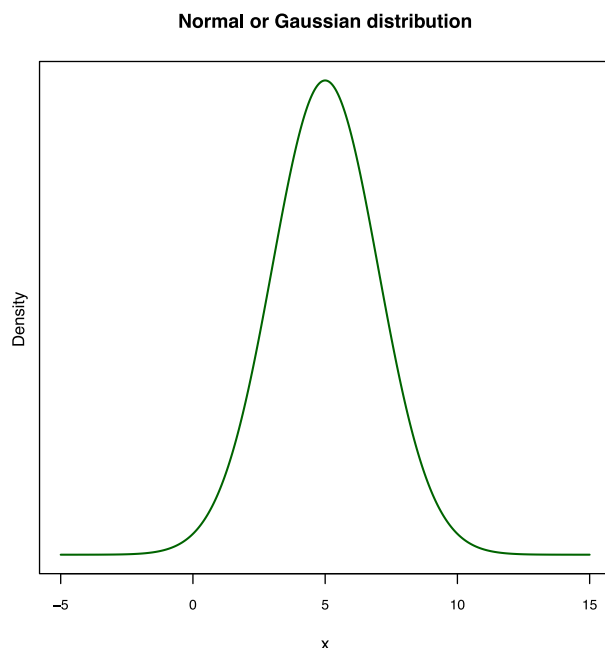


Fig. 3 Normal or Gaussian distribution with mean $\mu = 5$ and standard deviation $\sigma = 2$.

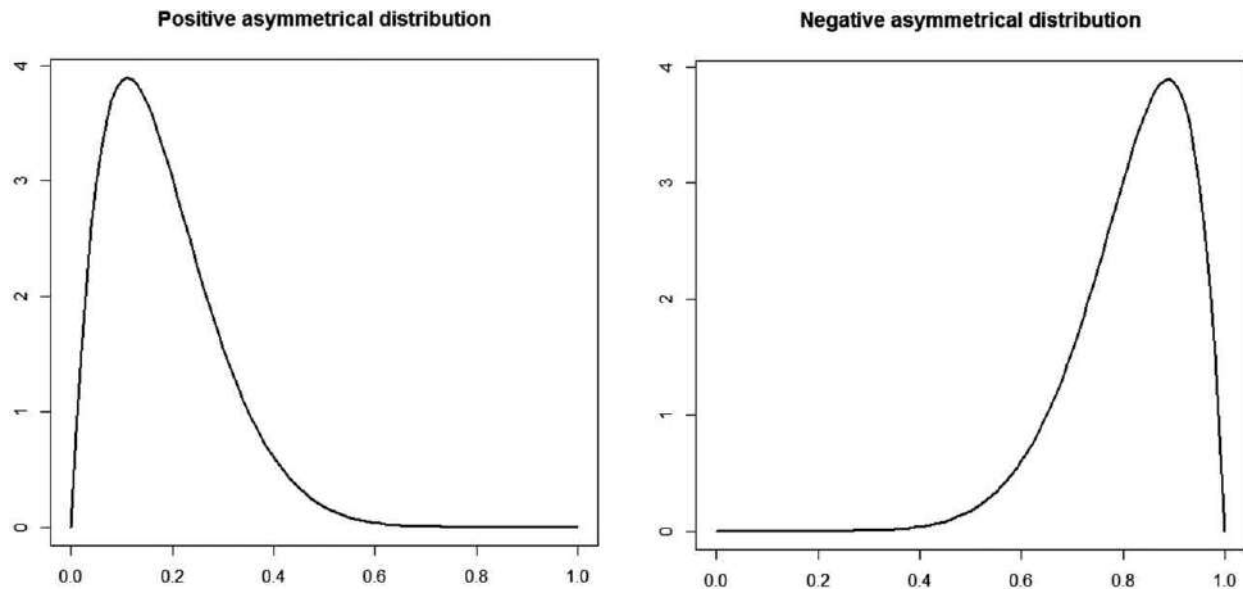


Fig. 4 Positively Skewed Distribution (to the right) and negatively Skewed Distribution (to the left).

mean, median and mode occur at different points of the distribution. In particular, there are two kinds of skewness. The distribution is said to be left-skewed (or negatively skewed) if the distribution appears to be skewed to the left, i.e. its mean is less than its mode. On the contrary, the distribution is said to be right-skewed (positively skewed) if the distribution is skewed to the right, i.e., its mean is greater than its mode. Most computer statistical packages (e.g., The R Project for Statistical Computing) include this statistic as part of a standard printout. A value of skewness > 0 indicates positive skewness and a value of skewness < 0 indicates negative skewness (see Fig. 4). As skewness involves the third moment of the distribution, kurtosis involves the fourth moment. Usually, kurtosis is quoted in the form of excess kurtosis (kurtosis relative to normal distribution kurtosis). Excess kurtosis is simply kurtosis less 3. In fact, kurtosis for a standard normal distribution is equal to three. There are three different ways to define the kurtosis. A distribution with excess kurtosis equal to zero (and kurtosis exactly 3) is called *mesokurtic*, or *mesokurtotic*. A distribution with positive excess kurtosis (and $\gamma_2 > 3$) is called *leptokurtic*, or *leptokurtotic*. A distribution with negative excess kurtosis (and $\gamma_2 < 3$) is called *platykurtic*, or *platykurtotic*. For instance, see Fig. 5. In terms of shape, a leptokurtic distribution has fatter tails while a platykurtic distribution has thinner tails. The computational formulas of skewness and kurtosis are shown in Table 3.

Inferential Statistics

Statistical inference is the procedure by which we obtain a conclusion about a population on the basis of the information contained in a sample drawn from that population. The basic assumption in statistical inference is that each element, within the population of interest, has the same probability of being included in a specific sample. Therefore, the knowledge of the probability distribution of a random variable provides the clinician and researcher with a powerful tool for summarizing and describing a set of data and for reaching conclusions about a population of data on the basis of a sample drawn from that population. In this section, we discuss two general areas of statistical inference, estimation and hypothesis testing, used to infer the population parameters under the assumption that the sample estimates follow the normal or Gaussian distribution. These types of statistical inference procedures are classified as parametric statistics.

Continuous Probability Distributions

To understand the nature of the distribution of a continuous random variable, we consider the probability density function which is the area under a smooth curve between any two points a and b , i.e., the definite integral between a and b . Thus, the probability of a continuous random variable to assume values between a and b is denoted by $P(a < X < b)$. The graph of probability density function is shown in Fig. 6.

The normal distribution is the most important continuous probability distribution in statistics. It describes well the distribution of random variables that arise in practice, such as the heights, weights, blood pressure, body mass, etc. Let X be a random variable normally distributed, the probability density function of X is given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < +\infty \quad (4)$$

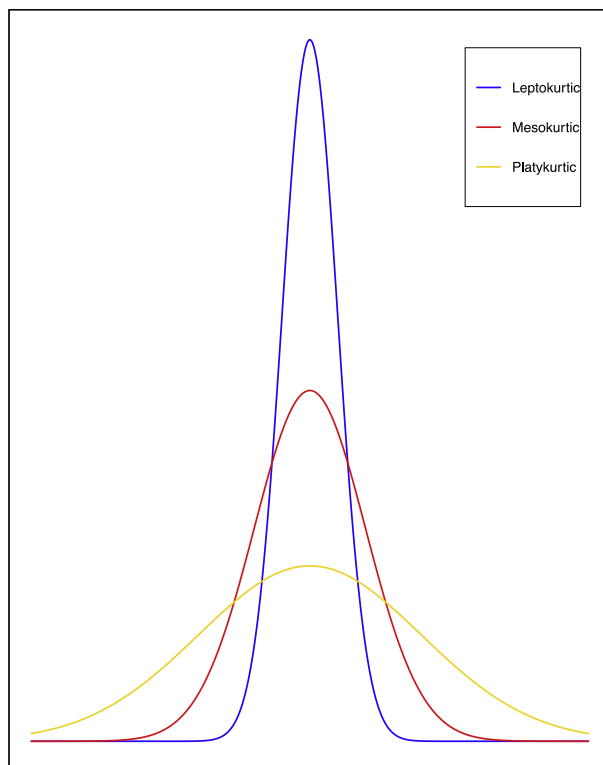


Fig. 5 Kurtosis distributions: a distribution with kurtosis equal to zero is called *mesokurtic*, or *mesokurtotic* (red line); a distribution with positive kurtosis is called *leptokurtic*, or *leptokurtotic* (blue line); a distribution with negative kurtosis is called *platykurtic*, or *platykurtotic* (gold line).

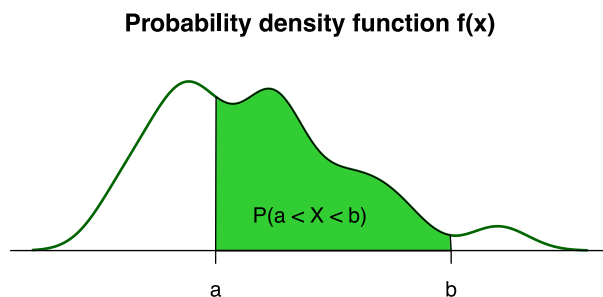


Fig. 6 Graph of a continuous distribution showing area between a and b . The probability of a continuous random variable to assume values between a and b is denoted by $P(a < X < b)$.

where the parameters μ and σ^2 are the mean and variance of X , respectively. Generally, we write $X \sim N(\mu, \sigma^2)$. Which means that X follows the normal distribution (or X is normally distributed) with mean μ , and variance σ^2 . The graph of the normal distribution produces the familiar bell-shaped curve shown in [Fig. 7](#). It is symmetrical about its mean μ . Mean, median and mode are equal. The total area under the curve above the x -axis is one square unit. In particular, the 68% of observations lie between $(\mu \pm \sigma)$, 95% of observations lie between $(\mu \pm 2\sigma)$ and 99.7% of observations lie between $(\mu \pm 3\sigma)$. For instance, see [Fig. 8](#). The normal distribution is completely determined by the parameters μ and σ^2 . Different values of μ and σ shift the graph of the distribution along the x -axis. Different values of σ determine the degree of flatness or peakedness of the graph of the distribution. Because of the characteristics of these two parameters, μ is often referred to as a location parameter and σ is often referred to as a shape parameter. The normal distribution with mean $\mu=0$ and $\sigma^2=1$ is called standard normal distribution. It is obtained from [Eq. \(5\)](#) by setting

$$z = \frac{(x - \mu)}{\sigma}$$

This value is called z -transformation (or z -score). Hence, the probability density function of the standard normal distribution is

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{z^2}{2}}, -\infty < x < +\infty \quad (5)$$

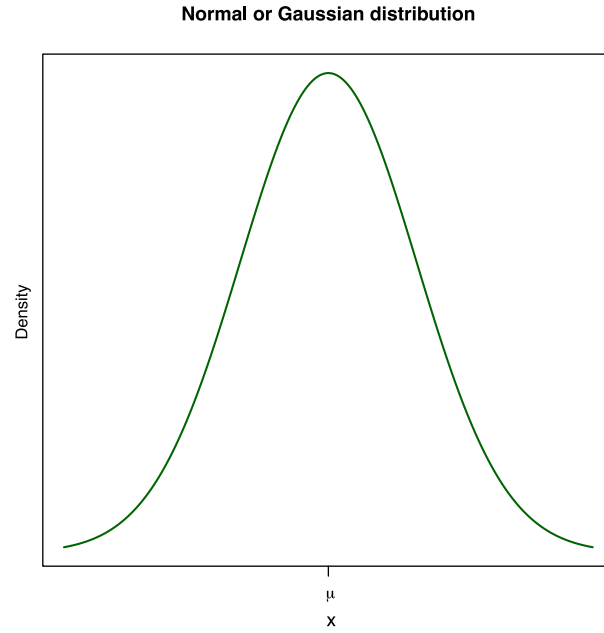


Fig. 7 Normal or Gaussian distribution with mean μ and standard deviation σ .

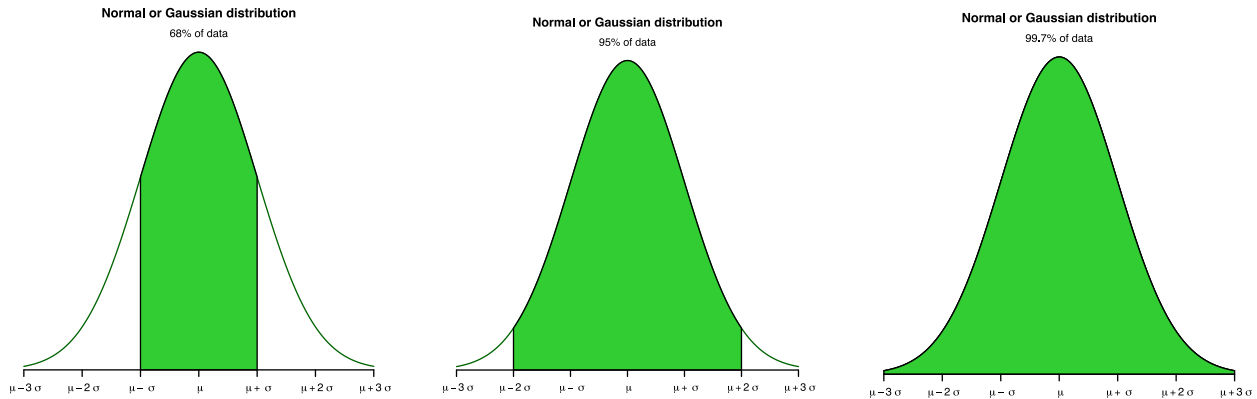


Fig. 8 Standard deviation and coverage. About 68% of values drawn from a normal distribution are within one standard deviation away from the mean; about 95% of the values lie within two standard deviations; and about 99.7% are within three standard deviations.

The graph of the standard normal distribution is shown in [Fig. 9](#). The probability of the random variables z between two points (or to the left/right of a given z -score) on the z -axis is the areas located under the curve of the standard normal distribution. The probability of z can be calculated by using standard normal distribution tables well known in literature.

Three important distributions related to the normal distribution are: Chi-square distribution, t distribution and F distribution. Let $X_1, X_2, \dots, X_m$ be m independent random variables having standard normal distribution, i.e., $X_i \sim N(0,1)$, the new random variable

$$Z = \sum_{i=1}^m X_i^2 \sim \chi_m^2$$

follows a Chi-Square distribution with m degrees of freedom (i.e., the number of random variables). Its mean is m , and its variance is $2m$. The probability density function of Z is given by

$$f(z) = \frac{1}{2^{\frac{m}{2}} \Gamma(\frac{m}{2})} \exp^{-\frac{z}{2}} z^{\frac{m}{2}-1}, \quad 0 < z < +\infty \quad (6)$$

where the gamma function Γ is the integral

$$\Gamma(x) = \int_0^{\infty} \gamma^{x-1} e^{-\gamma} d\gamma, \quad x > 0$$

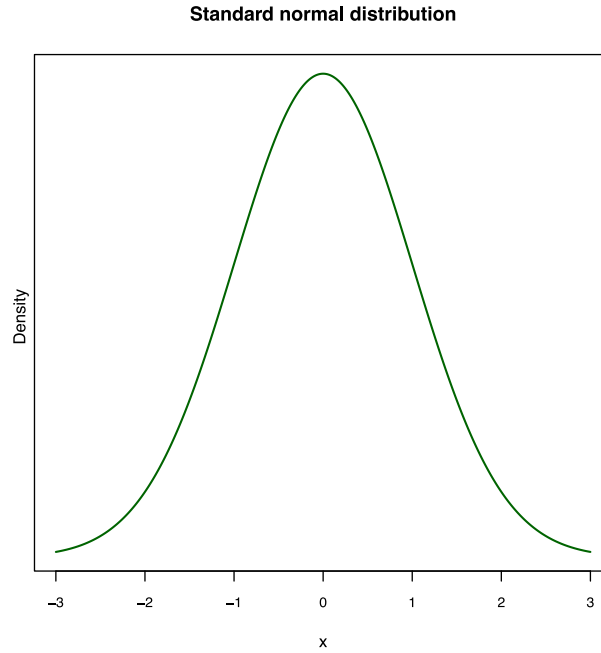


Fig. 9 Standard normal distribution with mean $\mu=0$ and standard deviation $\sigma=1$.

Note that as the degrees of freedom increase, the chi-square curve approaches a normal distribution. The graph of the Chi-squares X^2 distribution is shown in [Fig. 10](#).

The Student's t distribution is a probability distribution that is used to estimate population parameters when the sample size is small and/or when the population variance is unknown. Let Z be a random variable with standard normal distribution, i.e., $Z \sim N(0,1)$, and let V be a random variable having a Chi-square distribution with m degrees of freedom, i.e., $V \sim \chi_m^2$. Assume further that Z and V are independent. Define a new random variable T by

$$T = \frac{Z}{\sqrt{V/m}} \sim T_m$$

called Student t distribution with m degrees of freedom. The probability density function of the t distribution with m degrees of freedom is

$$f(t) = \frac{\Gamma(\frac{m+1}{2})}{\sqrt{\pi m} \Gamma(\frac{m}{2})} \left(1 + \frac{t^2}{m}\right)^{-\frac{m+1}{2}}, -\infty < t < +\infty \quad (7)$$

It is symmetrical about the mean equal to zero. It has a variance greater than 1, but the variance approaches 1 as the sample size becomes large, i.e., $\text{Var}(T) = \frac{m}{m-2}$. The shape of the t -distribution curve depends on the number of degrees of freedom. Compared to the normal distribution, the t distribution is less peaked in the center and has thicker tails. Finally, the t distribution approaches the standard normal distribution as m tends to infinity. The graph of the Student t distribution is shown in [Fig. 11](#).

Let U and V be independent chi-square random variables with m and n degrees of freedom, respectively. The variable

$$F = \frac{U/m}{V/n} \sim F_{m,n}$$

follows the Fisher F distribution with numerator degree of freedom m and denominator degree of freedom n . The probability density function of the F distribution is

$$f(x) = \frac{\Gamma(\frac{m+n}{2})}{\Gamma(\frac{m}{2})\Gamma(\frac{n}{2})} \left(\frac{m}{n}\right)^{\frac{m}{2}} x^{\frac{n}{2}-1} \left(1 + \frac{m}{n}x\right)^{-\frac{(m+n)}{2}}, 0 < x < +\infty \quad (8)$$

Then, the mean is $E(X) = \frac{n}{n-2}$ and the variance $\text{Var}(X) = \frac{2n^2(m+n-2)}{m(n-2)^2(m-4)}$. In general, the F distribution is skewed to the right. The graph of the Fisher F distribution is shown in [Fig. 12](#). The X^2, t, F distributions, like the standard normal, has been extensively tabulated.

Estimation

The estimation process consists of estimate sample statistics in order to give an approximation of the corresponding parameters of the population from which the sample is drawn. For example, we suppose that the administrator of a hospital is interested in the

Chi-square distributions

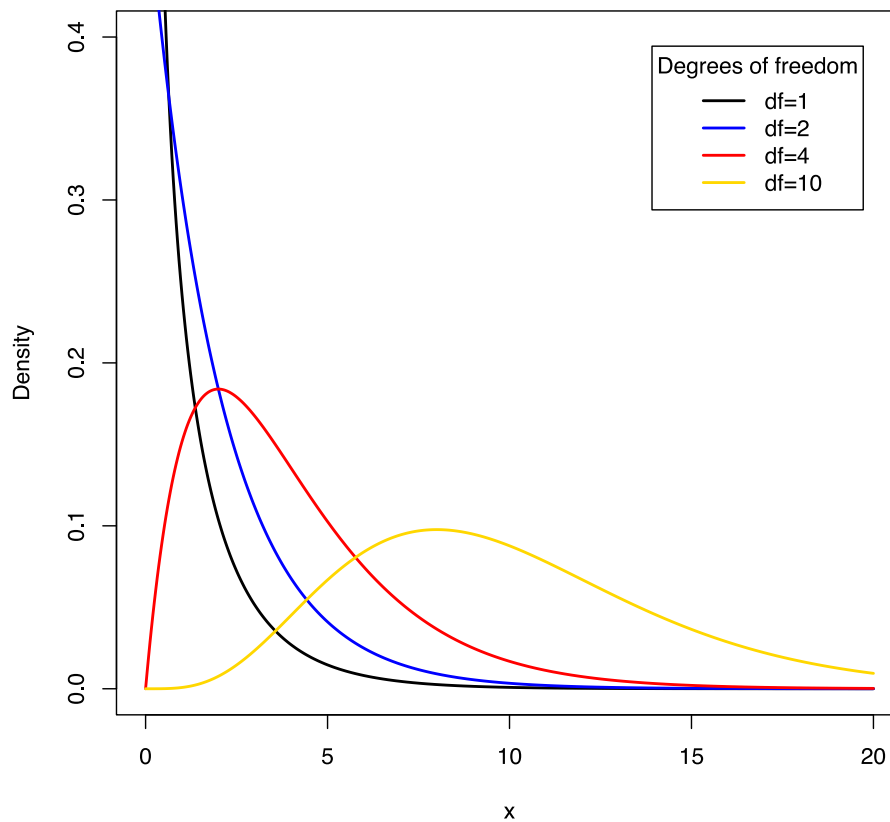


Fig. 10 Chi-square distribution for different degrees of freedom.

mean age of patients admitted to the hospital during a given year. He decides to examine only a sample of the records to conduct his analysis in order to determine a mean age estimation of all patients admitted to the hospital during the year. This statistic is an estimation of the corresponding population mean. Typically, we expect the estimate to differ by some amount from the parameter it estimates.

For each of the population parameters, we can compute two types of estimate: a point estimate and an interval estimate. A point estimate is a single numerical value used to estimate the corresponding population parameters, while an interval estimate is an interval that, with a specified degree of confidence, most likely includes the parameters being estimated ([Gardner and Altman, 1986](#)). For example, let X be a random variable that follows the normal distribution with mean μ , and variance σ^2 . The computed sample mean $\bar{x}$ is a point estimator of the population mean μ . Similarly, the computed sample variance s^2 is a point estimation of the of the population variance σ^2 . Interval estimation is an alternative procedure to point estimation. It consists to replace the point estimator of the population parameter by using a statistic that allows to calculate an interval of the parameter space. Let θ be the parameter space and let X be a random variable from a distribution that belongs to a family of distributions with a parameter $\theta \in \theta$. A confidence interval (CI) is an interval composed by two numerical values, called lower and upper limit, that with a specified degree of confidence $\alpha \in (0,1)$ includes the unknown parameter θ . In other words, it is an interval that with probability $1 - \alpha$, include the unknown parameter θ . The probability $1 - \alpha$ is called the confidence coefficient and represents the area under the probability distribution between the two limits of the CI. Usually, $1 - \alpha$ is taken to be 0.90, 0.95 or 0.99. To construct a confidence interval CI, we generally consider the following steps:

1. the sample statistic is identified to estimate a population parameter θ (for example, the population mean or the population variance);
2. the confidence level $1 - \alpha$ is fixed to compute the margin of error, i.e., the product between the critical value (which is a term that splits the area under the probability distribution in two regions) and the standard deviation;
3. the limits of the confidence interval are determined as follow.

$$\text{CI} = \text{sample statistic} \pm \text{margin of error} \quad (9)$$

In particular, when X is a random variable normally distributed with mean μ , and variance σ^2 , we can construct different type of CI for the mean μ with known or unknown variance σ^2 .

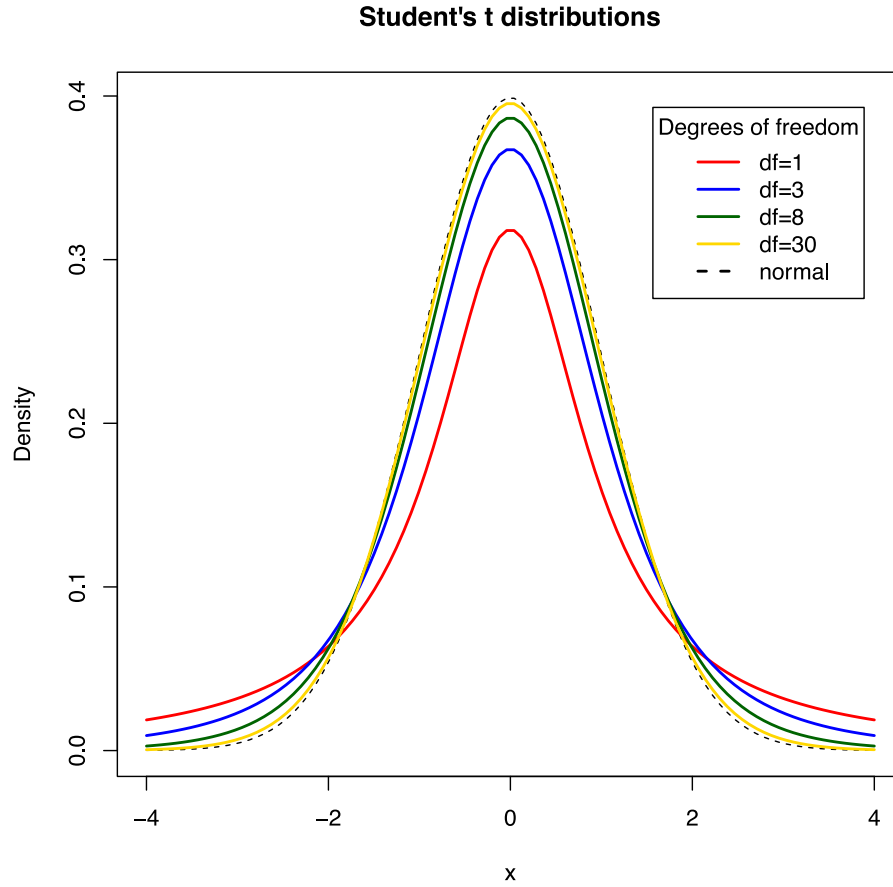


Fig. 11 Student distribution for different degree of freedom.

Confidence interval CI for the population mean μ

When the variance σ^2 is known, the statistic used to construct a $100(1 - \alpha)$ confidence interval CI for the population mean μ is the quantity

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1) \quad (10)$$

where σ is the known population standard deviation. Then, an interval estimate for μ is expressed as

$$\left(\bar{x} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}}, \bar{x} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \right) \quad (11)$$

where the critical value, denoted by $z_{\alpha/2}$, is the value of z to the left of which lies $-\alpha/2$ and to the right of which lies $\alpha/2$ of the area under its curve, i.e., $P(Z \geq |z_{\alpha/2}|) = \frac{\alpha}{2}$ with $Z \sim N(0, 1)$. For instance, see Fig. 13. When the variance σ^2 is unknown and the sample size n is large ($n > 30$), we consider the Student's t distribution. In this case, the statistic used to construct a $100(1 - \alpha)$ confidence interval CI for the population mean μ is given by

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} \sim T_{(n-1)} \quad (12)$$

where s is the sample standard deviation to replace σ in Eq. (11). This statistics follows a Student's distribution with $n - 1$ degrees of freedom. An interval estimate for μ is expressed as

$$\left(\bar{x} - t_{\alpha/2, n-1} \frac{s}{\sqrt{n}}, \bar{x} + t_{\alpha/2, n-1} \frac{s}{\sqrt{n}} \right) \quad (13)$$

where the critical value, denoted by $t_{\alpha/2, n-1}$, is the value of t to the left of which lies $-\alpha/2$ and to the right of which lies $\alpha/2$ of the area under its curve, i.e. $P(T \geq |t_{\alpha/2, n-1}|) = \frac{\alpha}{2}$ with $T \sim T_{(n-1)}$. For instance, see Fig. 14.

Confidence interval CI for the difference between the population means $\mu_1 - \mu_2$

Sometimes we are interested in estimating the difference between two population means. From each of the populations an independent random sample is drawn and, from the data of each, the sample means $\bar{x}_1$ and $\bar{x}_2$ respectively, are computed.

Fisher distributions

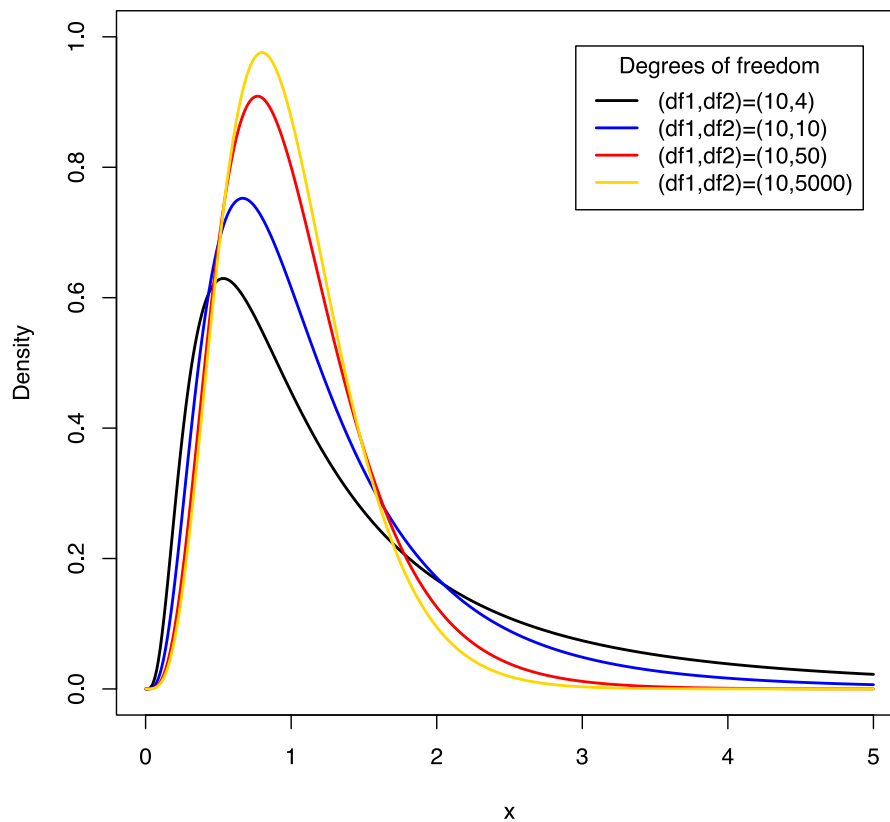


Fig. 12 Fisher distribution for different degree of freedom.

Critical regions – Standard normal distribution

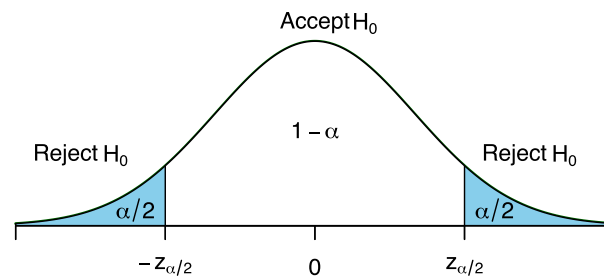


Fig. 13 Critical regions for the standard normal distribution.

Critical regions – Student t distribution

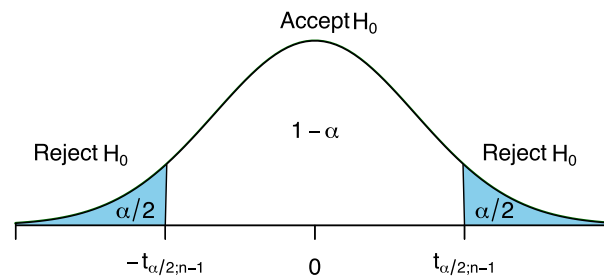


Fig. 14 Critical regions for the Student distribution.

An unbiased estimate of the difference between the population means, $\mu_1 - \mu_2$, is the difference between the sample means, $\bar{x}_1 - \bar{x}_2$. The variance of the estimator is $\left(\frac{\sigma_1^2}{n}\right) + \left(\frac{\sigma_2^2}{m}\right)$, where n and m are the sample sizes. The statistic used to construct a $100(1 - \alpha)$ confidence interval CI for the difference between the population means, $\mu_1 - \mu_2$ is

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}} \sim N(0, 1) \quad (14)$$

Hence, a confidence interval CI for $\mu_1 - \mu_2$ is given by

$$\left((\bar{x}_1 - \bar{x}_2) - z_{\alpha/2} \sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}, (\bar{x}_1 - \bar{x}_2) + z_{\alpha/2} \sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}} \right) \quad (15)$$

where the critical value, denoted by $z_{\alpha/2}$, is the value of z to the left of which lies $-\alpha/2$ and to the right of which lies $\alpha/2$ of the area under its curve, i.e. $P(Z \geq |z_{\alpha/2}|) = \frac{\alpha}{2}$ with $Z \sim N(0, 1)$. For instance, see Fig. 13. An investigation of a confidence interval CI for the difference between population means provides information that is helpful in deciding whether or not it is likely that the two population means are equal. When the constructed interval does not include zero, we say that the interval provides evidence that the two population means are not equal. When the interval includes zero, we say that the population means may be equal. When population variances are unknown, we use the t distribution to estimate the difference between two population means with a confidence interval CI. We assume that the two sampled populations are normally distributed. With regard to the population variances, we distinguish two cases: (1) the population variances are equal, and (2) the population variances are not equal. Let us consider each situation separately. If the population variances are equal, the two sample variances that we compute from two independent samples are estimates of the same quantity, the common variance. This estimation is called pooled estimate and it is obtained by computing the weighted average of the two sample variances. Each sample variance is weighted by its degrees of freedom. The pooled estimate is given by the formula

$$s_p^2 = \frac{(n-1)s_1^2 + (m-1)s_2^2}{n+m-2} \quad (15)$$

where n and m are the sample sizes. The statistic used to construct a $100(1 - \alpha)$ confidence interval CI for the difference between the population means, $\mu_1 - \mu_2$ is

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p \sqrt{\frac{1}{n} + \frac{1}{m}}} \sim T_{(n+m-2)} \quad (16)$$

Hence, a confidence interval CI for $\mu_1 - \mu_2$, when population variances are unknown and equal, is given by

$$\left((\bar{x}_1 - \bar{x}_2) - t_{\alpha/2; n+m-2} s_p \sqrt{\frac{1}{n} + \frac{1}{m}}, (\bar{x}_1 - \bar{x}_2) + t_{\alpha/2; n+m-2} s_p \sqrt{\frac{1}{n} + \frac{1}{m}} \right) \quad (17)$$

where the critical value, denoted by $t_{\alpha/2; n+m-2}$, is the value of t to the left of which lies $-\alpha/2$ and to the right of which lies $\alpha/2$ of the area under its curve, i.e., $P(T \geq |t_{\alpha/2; n+m-2}|) = \frac{\alpha}{2}$ with $T \sim T_{(n+m-2)}$.

If the population variances are not equal, the solution, proposed by Cochran (1964) consists of computing the reliability factor, $t'_{\alpha/2}$ by the following formula:

$$t'_{\alpha/2} = \frac{w_1 t_1 + w_2 t_2}{w_1 + w_2}$$

where $w_1 = \frac{s_1^2}{n}$, $w_2 = \frac{s_2^2}{m}$, $t_1 - t_{\alpha/2}$ for $n-1$ degrees of freedom, and $t_2 - t_{\alpha/2}$ for $m-1$ degrees of freedom. Hence, an approximate a $100(1 - \alpha)$ confidence interval CI for the difference between the population means, $\mu_1 - \mu_2$ is given by

$$\left((\bar{x}_1 - \bar{x}_2) - t'_{\alpha/2} s_p \sqrt{\frac{1}{n} + \frac{1}{m}}, (\bar{x}_1 - \bar{x}_2) + t'_{\alpha/2} s_p \sqrt{\frac{1}{n} + \frac{1}{m}} \right) \quad (18)$$

Confidence interval CI for a population proportion

Many questions of interest to the health science are related to population proportions. For example, the proportion of patients who receive a particular type of treatment, or the proportion of some population who has a certain disease or the proportion of a population who is immune to a certain disease. In this case, we consider the binomial distribution frequently used to model the number of successes p in a sample of size n drawn with replacement from a population of size n . Hence, the binomial distribution is characterized by two parameters, n and p . When the sample size is large, the distribution of sample proportions is approximately normally distributed by virtue of the central limit theorem. The mean of the distribution, $\mu_{\hat{p}}$, that is, the average of all the possible sample proportions, is equal to the true population proportion, p , and the variance of the distribution, $\sigma_{\hat{p}}^2$, is equal to $\frac{p(1-p)}{n}$. To estimate the population proportion, we compute the sample proposition $\hat{p}$. This sample proportion is used as the point estimator of the population proportion. In particular, when both np and $n(1-p)$ are greater than 5, we can say that the sampling distribution of $\hat{p}$ is approximately normally distributed with mean $\mu_{\hat{p}} = p$. Hence, the statistic used to construct a $100(1 - \alpha)$ confidence interval

CI for the population proportion p is given by

$$z = \frac{p - \hat{p}}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \sim N(0, 1) \quad (19)$$

A confidence interval CI for the population proportion p is

$$\left(\hat{p} - z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right) \quad (20)$$

where the critical value, denoted by $z_{\alpha/2}$, is the value of z to the left of which lies $-\alpha/2$ and to the right of which lies $\alpha/2$ of the area under its curve, i.e., $P(Z \geq |z_{\alpha/2}|) = \frac{\alpha}{2}$ with $Z \sim N(0,1)$. For instance, see Fig. 13.

Confidence interval CI for the difference between two population proportions

Often there are two population proportions in which we are interested and we desire to assess the probability associated with a difference in proportions computed from samples drawn from each of these populations. The relevant sampling distribution is the distribution of the difference between the two sample proportions. If independent random samples of size n and m are drawn from two populations of dichotomous variables where the proportions of observations with the characteristic of interest in the two populations are p_1 and p_2 , respectively, the distribution of the difference between sample proportions, $\hat{p}_1 - \hat{p}_2$, is approximately normal with mean and variance equal to

$$\mu_{\hat{p}_1 - \hat{p}_2} = p_1 - p_2, \text{ and } \sigma_{\hat{p}_1 - \hat{p}_2}^2 = \frac{p_1(1-p_1)}{n} + \frac{p_2(1-p_2)}{m}$$

respectively, when n and m are large (i.e., np_1 , mp_2 , $n(1-p_1)$ and $m(1-p_2)$ are greater than 5). Hence, an unbiased point estimator of the difference between two population proportions is provided by the difference between sample proportions, $\hat{p}_1 - \hat{p}_2$. The statistic used to construct a $100(1-\alpha)$ confidence interval CI for the difference between two population proportions $p_1 - p_2$ is

$$z = \frac{(\hat{p}_1 - \hat{p}_2) - (p_1 - p_2)}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n} + \frac{\hat{p}_2(1-\hat{p}_2)}{m}}} \sim N(0, 1) \quad (21)$$

A confidence interval CI for $p_1 - p_2$ is given by

$$\left((\hat{p}_1 - \hat{p}_2) - z_{\alpha/2} \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n} + \frac{\hat{p}_2(1-\hat{p}_2)}{m}}, (\hat{p}_1 - \hat{p}_2) + z_{\alpha/2} \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n} + \frac{\hat{p}_2(1-\hat{p}_2)}{m}} \right) \quad (22)$$

where the critical value, denoted by $z_{\alpha/2}$, is the value of z to the left of which lies $-\alpha/2$ and to the right of which lies $\alpha/2$ of the area under its curve, i.e., $P(Z \geq |z_{\alpha/2}|) = \frac{\alpha}{2}$ with $Z \sim N(0,1)$. For instance, see Fig. 13.

Confidence interval CI for the population variance σ^2

The statistic used to construct a $100(1-\alpha)$ confidence interval CI for the population variance σ^2 is

$$\chi^2 = \frac{(n-1)s^2}{\sigma^2} \sim \chi^2(n-1) \quad (23)$$

This statistics follows a chi-square χ^2 distribution with $n-1$ degrees of freedom. An interval estimate for σ^2 is expressed as

$$\left(\frac{(n-1)s^2}{\chi_{1-\alpha/2; n-1}^2}, \frac{(n-1)s^2}{\chi_{\alpha/2; n-1}^2} \right) \quad (24)$$

where s is the sample variance and $\chi_{1-\alpha/2; n-1}^2$ and $\chi_{\alpha/2; n-1}^2$ are the values from the χ^2 table to the left and right of which, respectively, lies $\alpha/2$ of the area under the curve, i.e., $P(Y \leq \chi_{1-\alpha/2; n-1}^2) = 1 - \frac{\alpha}{2}$ and $P(Y \geq \chi_{\alpha/2; n-1}^2) = \frac{\alpha}{2}$ with $Y \sim \chi_{(n-1)}^2$. For instance, see Fig. 15. If we take the square root of each term in Eq. (23), we have the confidence interval for σ , the population standard deviation.

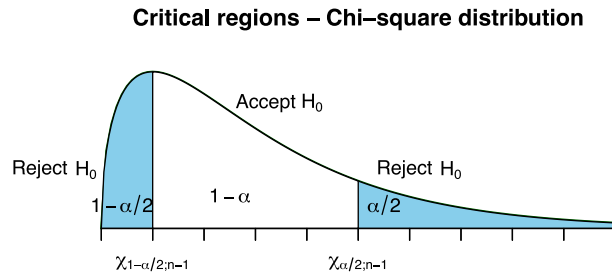


Fig. 15 Critical regions for the Chi-square distribution.

Critical regions – Fisher distribution

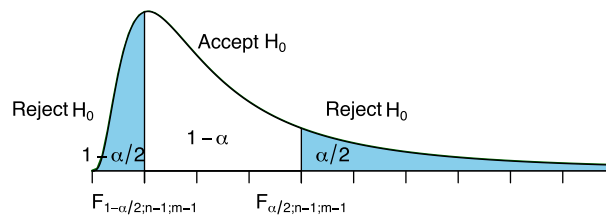


Fig. 16 Critical regions for the Fisher distribution.

Confidence interval CI for the ratio of the variances of two normally distributed populations

Generally, we consider the ratio σ_1^2/σ_2^2 to compare the variances of two normally distributed populations. If two variances are equal, their ratio will be equal to 1. We usually will not know the variances of populations of interest, and, consequently, any comparisons we make will be based on sample variances. In other words, we may wish to estimate the ratio of two population variances. The assumptions are that s_1^2 and s_2^2 are computed from independent samples of size n and m respectively, drawn from two normally distributed populations. The variance s_1^2 is designed as the larger of the two sample variances. Hence, the statistic used to construct a $100(1-\alpha)$ confidence interval CI for the ratio of the variances of two normally distributed populations is

$$F = \frac{s_1^2/\sigma_1^2}{s_2^2/\sigma_2^2} \sim F_{(n-1; m-1)} \quad (25)$$

This statistic follows a F distribution depending on two-degrees-of-freedom values, one corresponding to the value of $n-1$ used in computing s_1^2 and the other corresponding to the value of $m-1$ used in computing s_2^2 . These are usually referred to as the numerator degrees of freedom and the denominator degrees of freedom. An interval estimate for the ratio σ_1^2/σ_2^2 is expressed as

$$\left(\frac{s_1^2/s_2^2}{F_{1-\alpha/2; n-1; m-1}}, \frac{s_1^2/s_2^2}{F_{\alpha/2; n-1; m-1}} \right) \quad (26)$$

where $F_{1-\alpha/2; n-1; m-1}$ and $F_{\alpha/2; n-1; m-1}$ are the values from the F table to the left and right of which, respectively, lies $\alpha/2$ of the area under the curve i.e., $P(F \leq F_{1-\alpha/2; n-1; m-1}) = 1 - \frac{\alpha}{2}$ and $P(Y \geq F_{\alpha/2; n-1; m-1}) = \frac{\alpha}{2}$ with $F \sim F_{(n-1; m-1)}$. For instance, see Fig. 16.

Note that the tables of all the critical values can be used for both one-sided (lower and upper) and two-sided tests with specific values of α .

Hypothesis Testing

The aim of hypothesis testing is to aid the clinician and researcher in reaching a conclusion concerning a population by examining a sample from that population. Interval estimation, discussed in the preceding section, and hypothesis testing are based on similar concepts. In fact, confidence intervals can be used to obtain the same conclusions that are reached through the use of hypothesis tests. There are two statistical hypotheses involved in hypothesis testing: the null hypothesis and the alternative hypothesis. The null hypothesis is the hypothesis to be tested and it is designated by the symbol H_0 . The null hypothesis is the hypothesis of no difference (or equality, either $=$, $\leq$, or $\geq$), since it is a statement of agreement with conditions supposed to be true in the population under investigated. Consequently, the conclusion that the researcher is seeking to reach is to reject the null hypothesis. If the null hypothesis is not rejected, we conclude that the data not provide sufficient evidence that the null hypothesis is not in reality true. The alternative hypothesis, designed by the symbol H_1 , is the statement that researchers hope to be true. In other words, it is the hypothesis of effect or real difference. Based on the sample data, the test determines whether to reject the null hypothesis. In particular, we can follow two types of decisional strategies. The first approach is based on the computation of test statistic, the second one is called p -value approach (Altman and Krzywinski, 2017; Gardner and Altman, 1986). All possible values that the test statistic can assume are arranged on the horizontal axis of the probability distribution and are divided into two regions: the rejection region and non rejection region. The values of the test statistic forming the rejection region are those values that are less likely to occur if the null hypothesis is true, while the values making up the acceptance region are more likely to occur if the null hypothesis is true. The decision rule tells us to reject the null hypothesis if the value of the test statistic that we compute from the sample falls in the rejection region or not. The decision to reject or accept the null hypothesis is based on the level of significance α . A computed value of the test statistic that falls in the rejection region is said to be significant. Generally, a small value of α is selected to make the probability of rejecting a true null hypothesis small. The more frequently values used for α are 0.01, 0.05, and 0.10. The relationship between the (unknown) reality if the null hypothesis is true or not and the decision to accept or reject the null hypothesis is shown in Table 4. The error committed when a true null hypothesis is rejected is called the type I error. The type II error is the error committed when a false null hypothesis is not rejected. The probability of committing a type II error is designated by β . The II error is know as the statistical power, which is the ability of a test to detect a true effect, i.e., reject the null hypothesis if the alternative hypothesis is true. The second strategy is based on the concept of p -value, which is the probability that the computed test statistic is at least as extreme as a specified value of the test statistic when the null

Table 4 Conditions under which type I and type II errors may be committed

Decision rules	The truth	
	H_0 true	H_1 true
Accept H_0	Correct decision	Type II error
Reject H_0	Type I error	Correct decision

Table 5 Confidence intervals (CI) and hypothesis test for the single population mean μ and for the difference between two population means μ_1 and μ_2 when sampling from normally distributed populations

Statistics	σ^2 Known	Hypothesis test	Critical regions
	CI at level α		
$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} \sim N(0, 1)$ z -transformation	$\left(\bar{X} - Z_{\alpha/2} \frac{\sigma}{\sqrt{n}}, \bar{X} + Z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \right)$	$H_0 : \mu = \mu_0$ $H_1 : \mu \neq \mu_0$ $(\mu > \mu_0 \text{ or } \mu < \mu_0)$	Reject H_0 $z \leq -Z_{\alpha/2} \text{ or } z \geq Z_{\alpha/2}$ $p\text{-value} < \alpha$ Accept H_0 $-Z_{\alpha/2} \leq z \leq Z_{\alpha/2}$ $p\text{-value} > \alpha$
$z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}} \sim N(0, 1)$ n and m sample sizes	$\left(\bar{X}_1 - \bar{X}_2 - Z_{\alpha/2} \sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}, \bar{X}_1 - \bar{X}_2 + Z_{\alpha/2} \sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}} \right)$	$H_0 : \mu_1 = \mu_2$ $H_1 : \mu_1 \neq \mu_2$ $(\mu_1 > \mu_2 \text{ or } \mu_1 < \mu_2)$	Reject H_0 $z \leq -Z_{\alpha/2} \text{ or } z \geq Z_{\alpha/2}$ $p\text{-value} < \alpha$ Accept H_0 $-Z_{\alpha/2} \leq z \leq Z_{\alpha/2}$ $p\text{-value} > \alpha$

hypothesis is true. Thus, the p -value is the smallest value (typical less than 5%) for which we reject a null hypothesis. To calculate the p -value, we first calculate the value of the test statistic, and then, using the known distribution of the test statistic, calculate the p -value. Note that the probability of rejecting a true null hypothesis is the significance level α , we conclude that, if the p -value is less than (or equal to) α , then the null hypothesis is rejected, while, if the p -value is greater than α , then the null hypothesis is not rejected (or accepted). For instance, see [Tables 5](#) and [6](#).

Non-Parametric Statistics

In this section, we explore the most common non-parametric techniques used when the underlying assumptions of traditional hypothesis tests are violated. These statistical procedures allow for the testing of hypotheses that are not statements about population parameter values and are applied when the form of the sampled population is unknown.

Wilcoxon signed-rank test for location

Suppose to test a null hypothesis about a population mean, but neither z nor t is an appropriate test statistic because the sampled population does not follow or approximate a normal distribution ([Wilcoxon, 1945](#)). When confronted with such a situation we use a non-parametric statistical procedure called Wilcoxon signed-rank test for location. It makes use of the magnitudes of the differences between measurements and a hypothesized location parameter rather than just the signs of the differences. The Wilcoxon test is based on the following assumptions about the data: (i) the sample is random; (ii) the variable is continuous; (iii) the population is symmetrically distributed about its mean μ ; (iv) the measurement scale is at least interval. After the formulation of null mean H_0 and alternative hypothesis H_1 ,

$$H_0 : \mu = \mu_0 (\leq, \geq) \text{ vs } H_1 : \mu \neq \mu_0 (>, <)$$

we perform the Wilcoxon test when the population mean μ_0 is unknown.

1. Subtract the Hypothesized Mean μ_0 from Each Observation x_i to Obtain

$$d_i = x_i - \mu_0.$$

If any x_i is equal to the mean, so that, $d_i = 0$, eliminate that d_i from the calculations and reduce n accordingly.

2. Rank the usable d_i from the smallest to the largest without regard to the sign of d_i . That is, consider only the absolute value of the d_i , designated $|d_i|$, when ranking them. If two or more of the $|d_i|$ are equal, assign each tied value the mean of the rank positions the tied values occupy. If, for example, the three smallest $|d_i|$ are all equal, place them in rank positions 1, 2, and 3, but assign each a rank of $(1 + 2 + 3)/3 = 2$.

Table 6 Confidence intervals (CI) and hypothesis test for the single population mean μ , for the difference between two population means μ_1 and μ_2 , for population variance σ^2 and for the ratio of the variances when sampling from normally distributed populations

σ^2 Unknown			
Statistics	CI at level α	Hypothesis test	Critical regions
$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \sim T_{(n-1)}$ <i>t</i> -transformation $t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \sqrt{\frac{1}{n} + \frac{1}{m}}} \sim T_{(n+m-2)}$ where $s_p^2 = \frac{(n-1)s_1^2 + (m-1)s_2^2}{n+m-2}$ n and m sample sizes	$\left(\bar{X} - t_{\alpha/2, n-1} \frac{s}{\sqrt{n}}, \bar{X} + t_{\alpha/2, n-1} \frac{s}{\sqrt{n}} \right)$ $\left(\bar{X}_1 - \bar{X}_2 - t_{\alpha/2, n+m-2} s_p \sqrt{\frac{1}{n} + \frac{1}{m}}, \bar{X}_1 - \bar{X}_2 + t_{\alpha/2, n+m-2} s_p \sqrt{\frac{1}{n} + \frac{1}{m}} \right)$	$H_0 : \mu = \mu_0$ $H_1 : \mu \neq \mu_0$ $(\mu > \mu_0 \text{ or } \mu < \mu_0)$ $H_0 : \mu_1 = \mu_2$ $H_1 : \mu_1 \neq \mu_2$ $(\mu_1 > \mu_2 \text{ or } \mu_1 < \mu_2)$	Reject H_0 $t \leq -t_{\alpha/2, n-1}$ or $t \geq t_{\alpha/2, n-1}$ $p\text{-value} < \alpha$ Accept H_0 $-t_{\alpha/2, n-1} \leq t \leq t_{\alpha/2, n-1}$ $p\text{-value} > \alpha$ Reject H_0 $t \leq -t_{\alpha/2, n+m-2}$ or $t \geq t_{\alpha/2, n+m-2}$ $p\text{-value} < \alpha$ Accept H_0 $-t_{\alpha/2, n+m-2} \leq t \leq t_{\alpha/2, n+m-2}$ $p\text{-value} > \alpha$
$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2} \sim \chi_{n-1}^2$	$\left(\frac{n-1}{\chi_{\alpha/2, n-1}^2} s^2, \frac{n-1}{\chi_{1-\alpha/2, n-1}^2} s^2 \right)$	$H_0 : \sigma^2 = \sigma_0^2$ $H_1 : \sigma^2 \neq \sigma_0^2$ $(\sigma^2 < \sigma_0^2 \text{ or } \sigma^2 > \sigma_0^2)$	Reject H_0 $0 \leq \chi^2 < \chi_{1-\alpha/2, n-1}^2$ or $\chi^2 \geq \chi_{\alpha/2, n-1}^2$ Accept H_0 $\chi_{1-\alpha/2, n-1}^2 \leq \chi^2 \leq \chi_{\alpha/2, n-1}^2$
$F = \frac{s_1^2/\sigma_2^2}{s_2^2/\sigma_1^2} \sim F_{(n-1, m-1)}$	$\left(\frac{s_1^2/s_2^2}{F_{1-\alpha/2, n-1, m-1}}, \frac{s_1^2/s_2^2}{F_{\alpha/2, n-1, m-1}} \right)$	$H_0 : \sigma_1^2 = \sigma_2^2$ $H_0 : \sigma_1^2 \neq \sigma_2^2$ $(\sigma_1^2 > \sigma_2^2, \sigma_1^2 < \sigma_2^2)$	Reject H_0 $0 \leq F \leq F_{1-\alpha/2, n-1, m-1}$ or $F \geq F_{\alpha/2, n-1, m-1}$ Accept H_0 $F_{1-\alpha/2, n-1, m-1} \leq F \leq F_{\alpha/2, n-1, m-1}$

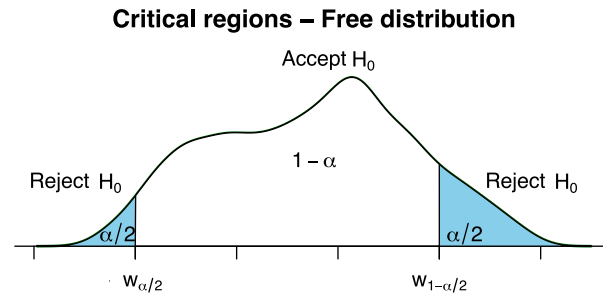


Fig. 17 Critical regions for a free distribution.

3. Assign each rank the sign of the $|d_i|$ that yields that rank.
4. Find T_+ the sum of the ranks with positive signs, and, T_- , the sum of the ranks with negative signs.

The Wilcoxon test statistic is either T_+ or T_- , depending on the nature of the alternative hypothesis. If the null hypothesis is true, that is, if the true population mean is equal to the hypothesized mean, the probability of observing a positive difference $d_i = x_i - \mu_0$ of a given magnitude is equal to the probability of observing a negative difference of the same magnitude. Then, in repeated sampling, when the null hypothesis is true and the assumptions are met, the expected value of T_+ is equal to the expected value of T_- . However, when H_0 is true, we do not expect a large difference in their values. Consequently, a sufficiently small value of T_+ or a sufficiently small value of T_- will cause rejection of H_0 . When the alternative hypothesis is two-sided ($\mu \neq \mu_0$), either a sufficiently small value of T_+ or a sufficiently small value of T_- will cause rejection of H_0 . The test statistic, then, is T_+ or T_- , whichever is smaller. To simplify notation, we call the smaller of the two T . Similarly, when the one-sided alternative hypothesis is true a sufficiently small (or large) value of T_+ (or T_-) will cause rejection of H_0 , and T_+ (or T_-) is the test statistic. Critical values of the Wilcoxon test statistic are given in probability tables well known in literature. The following are the decision rules for the three possible alternative hypotheses:

1. $H_1: \mu \neq \mu_0$. Reject H_0 at the level of significance α if the calculated T is smaller than or equal to the tabulated T for n and preselected $\alpha/2$ (see Fig. 17).
2. $H_1: \mu < \mu_0$. Reject H_0 at the level of significance α if T_+ is less than or equal to the tabulated T for n and preselected α .
3. $H_1: \mu > \mu_0$. Reject H_0 at the level of significance α if T_- is less than or equal to the tabulated T for n and preselected α .

The Mann-Whitney test

Another important non-parametric test is the Mann-Whitney test based on the ranks of the observations (Mann and Whitney, 1947). The assumptions underlying the Mann-Whitney test are as follows: (i) the two samples, of size n and m , respectively, available for analysis have been independently and randomly drawn from their respective populations; (ii) the measurement scale is at least ordinal; (iii) the variable of interest is continuous; (iv) if the populations differ at all, they differ only with respect to their medians. When these assumptions are met we test the null hypothesis that the two populations have equal medians against either of the three possible alternatives: (1) the populations do not have equal medians (two-sided test), (2) the median of population 1 is larger than the median of population 2 (one-sided test), or (3) the median of population 1 is smaller than the median of population 2 (one-sided test). If the two populations are symmetric, so that within each population the mean and median are the same, the conclusions we reach regarding the two population medians will also apply to the two population means. In particular, the null and alternative hypotheses are given by:

$$H_0 : M_X = M_Y (\leq, \geq) \text{ vs } H_1 : M_X \neq M_Y (>, <)$$

where M_X is the median of a population of population 1 and M_Y is the median of population 2. For a fixed significance level α , we compute the test statistic combining the two samples and rank all observations from smallest to largest while keeping track of the sample to which each observation belongs. Tied observations are assigned a rank equal to the mean of the rank positions for which they are tied. The test statistic is

$$T = S - \frac{n(n+1)}{2}$$

where n is the number of sample X observations and S is the sum of the ranks assigned to the sample observations from the population of X values. The choice of which sample's values we label X is arbitrary. If the median of the X population is smaller than the median of the Y population, as specified in the alternative hypothesis, we would expect (for equal sample sizes) the sum of the ranks assigned to the observations from the X population to be smaller than the sum of the ranks assigned to the observations from the Y population. A sufficiently small value of T will cause rejection of H_0 . Critical values of the Mann-Whitney test statistic are given in probability table well known in literature. The following are the decision rules for the three possible alternative hypotheses:

1. $H_1: M_X \neq M_Y$. Reject H_0 if the computed T is either less than $w_{\alpha/2}$, or greater than $w_{1-\alpha/2}$, where $w_{\alpha/2}$ is the tabulated critical value of T for n , the number of X observations; m , the number of Y observations; and $\alpha/2$, the chosen level of significance, and $w_{1-\alpha/2} = nm - w_{\alpha/2}$. For instance, see Fig. 17.

Table 7 Clinical depression data. For each patient, the dataset contains the following characteristic or variables. Hospt: the patient's hospital with 1, 2, 3, 5, or 6; Treat: the treatment received by the patient (Lithium, Imipramine, or Placebo); Outcome: recurrence or no recurrence occurred during the patient's treatment; Time: the length (in days) of the patient's participation in the study in terms of recurrence or no recurrence; AcuteT: the time (in days) that the patient was depressed prior to the study; Age: the age of the patient in years, when the patient entered the study; Gender: The patient's gender (1 = Female, 2 = Male). The number of patients are 109

<i>Hospt</i>	<i>Treat</i>	<i>Outcome</i>	<i>Time</i>	<i>AcuteT</i>	<i>Age</i>	<i>Gender</i>
1	Lithium	Recurrence	36,143	211	33	1
1	Imipramine	No Recurrence	105,143	176	49	1
1	Imipramine	No Recurrence	74,571	191	50	1
1	Lithium	Recurrence	49,714	206	29	2
1	Lithium	No Recurrence	14,429	63	29	1
1	Placebo	Recurrence	5	70	30	2
1	Lithium	No Recurrence	104,857	55	56	1
1	Placebo	Recurrence	2,857	512	48	1
1	Placebo	No Recurrence	102,429	162	22	2
1	Placebo	Recurrence	55,714	306	61	2
1	Imipramine	No Recurrence	106,429	165	58	1
1	Imipramine	No Recurrence	105,143	129	31	1
1	Imipramine	No Recurrence	83	428	44	1
1	Imipramine	Recurrence	27,286	256	55	2
1	Lithium	No Recurrence	105,857	197	57	2
1	Lithium	Recurrence	5,571	227	46	1
1	Imipramine	No Recurrence	98	168	58	1
1	Lithium	No Recurrence	16,286	194	57	1
2	Lithium	Recurrence	1,286	173	54	1
2	Lithium	No Recurrence	2,143	48	23	1
2	Imipramine	No Recurrence	100	47	65	1
2	Imipramine	Recurrence	27,143	95	27	1
2	Lithium	Recurrence	4	148	50	1
2	Lithium	Recurrence	74,143	127	41	2
2	Placebo	No Recurrence	104,857	129	65	1
2	Placebo	Recurrence	0,143	182	52	1
2	Placebo	Recurrence	1,429	90	60	1
2	Placebo	Recurrence	45,857	177	25	2
2	Imipramine	Recurrence	17,429	234	27	2
2	Imipramine	No Recurrence	78	322	32	1
2	Imipramine	Recurrence	66,857	141	43	2
2	Placebo	No Recurrence	78,429	165	20	2
2	Lithium	No Recurrence	78,429	239	23	2
2	Imipramine	No Recurrence	78,143	147	36	2
2	Imipramine	No Recurrence	15,857	348	22	2
3	Lithium	No Recurrence	79	274	49	2
3	Imipramine	No Recurrence	32,571	130	40	2
3	Lithium	Recurrence	9	98	54	2
3	Lithium	Recurrence	3,286	77	26	1
3	Imipramine	No Recurrence	206	90	48	1
3	Lithium	Recurrence	30	280	51	2
3	Placebo	Recurrence	7,143	167	35	2
3	Placebo	Recurrence	31	181	28	1
3	Placebo	Recurrence	17,286	399	23	1
3	Placebo	Recurrence	0,143	289	57	2
5	Lithium	Recurrence	3,286	182	47	1
5	Imipramine	No Recurrence	1,571	159	31	2
5	Lithium	Recurrence	19,714	122	27	1
5	Imipramine	No Recurrence	126,714	115	61	1
5	Placebo	Recurrence	8	343	60	1
5	Lithium	Recurrence	71,714	114	28	1
5	Placebo	Recurrence	63,714	249	36	1
5	Placebo	No Recurrence	96,286	140	29	1
5	Lithium	No Recurrence	50,857	110	34	1
5	Imipramine	No Recurrence	155	214	49	1
5	Imipramine	No Recurrence	39,571	224	45	1
5	Lithium	Recurrence	36,286	294	28	1

(Continued)

Table 7 Continued

<i>Hospt</i>	<i>Treat</i>	<i>Outcome</i>	<i>Time</i>	<i>AcuteT</i>	<i>Age</i>	<i>Gender</i>
5	Placebo	No Recurrence	102,571	162	24	2
5	Placebo	Recurrence	8,143	140	33	2
5	Imipramine	No Recurrence	28	147	34	1
5	Imipramine	No Recurrence	38	138	60	1
5	Imipramine	No Recurrence	111,571	196	23	2
5	Lithium	No Recurrence	165	139	35	1
5	Placebo	Recurrence	16	246	45	1
5	Lithium	No Recurrence	124,571	105	46	1
5	Lithium	No Recurrence	68	160	38	2
5	Placebo	No Recurrence	39,571	146	32	2
5	Placebo	No Recurrence	131	187	33	1
5	Imipramine	Recurrence	3,429	372	52	1
5	Lithium	No Recurrence	42	146	50	2
5	Imipramine	No Recurrence	26	131	38	1
5	Lithium	Recurrence	37,857	237	47	1
5	Imipramine	Recurrence	92,714	105	23	1
5	Imipramine	No Recurrence	106,714	140	31	1
5	Placebo	Recurrence	11,143	136	55	1
5	Placebo	No Recurrence	115	147	39	1
5	Placebo	Recurrence	44	160	41	1
5	Imipramine	No Recurrence	75	175	62	2
5	Placebo	No Recurrence	77,857	261	50	2
5	Lithium	Recurrence	0,286	146	46	1
5	Imipramine	No Recurrence	86	195	33	2
5	Placebo	No Recurrence	12,429	476	22	1
5	Lithium	No Recurrence	22	441	37	2
6	Lithium	Recurrence	5,429	86	40	2
6	Lithium	No Recurrence	67	201	22	1
6	Imipramine	Recurrence	3,429	130	30	2
6	Lithium	Recurrence	6,286	86	63	2
6	Imipramine	No Recurrence	5	209	40	1
6	Lithium	Recurrence	5,286	214	23	1
6	Imipramine	Recurrence	1	72	52	1
6	Placebo	Recurrence	3,429	238	23	1
6	Placebo	Recurrence	6,571	133	22	2
6	Placebo	Recurrence	1	128	23	1
6	Placebo	No Recurrence	45	139	30	2
6	Imipramine	No Recurrence	109,571	148	26	2
6	Lithium	Recurrence	0,857	285	46	1
6	Placebo	Recurrence	4,714	141	61	1
6	Imipramine	Recurrence	0,571	212	30	2
6	Imipramine	No Recurrence	9,143	168	39	1
6	Imipramine	No Recurrence	102	305	49	1
6	Lithium	Recurrence	46,286	204	57	1
6	Lithium	Recurrence	0,571	140	51	1
6	Lithium	Recurrence	6,429	182	53	1
6	Placebo	Recurrence	0	162	31	1
6	Placebo	Recurrence	20,857	207	43	1
6	Placebo	Recurrence	18,286	102	29	1
6	Imipramine	Recurrence	31,857	154	28	1
6	Imipramine	Recurrence	22	203	51	1
6	Lithium	Recurrence	2	176	33	1

2. $H_1: M_X > M_Y$. Reject H_0 if the computed T is less than $w_{1-\alpha}$, where $w_{1-\alpha} = nm - w_\alpha$ is the tabulated critical value for n , the number of X observations; m , the number of Y observations; and α , the chosen level of significance.
3. $H_1: M_X < M_Y$. Reject H_0 if the computed T is less than w_α , where w_α is the tabulated critical value of T for n , the number of X observations; m , the number of Y observations; and α , the chosen level of significance.

When either n or m is greater than 20 we compute the following test statistic

$$z = \frac{T - nm/2}{\sqrt{nm(n+m+1)/12}} \sim N(0, 1)$$

and compare the result, for significance, with critical values of the standard normal distribution. Finally, many computer packages give the test value of both the Mann–Whitney test (U) and the Wilcoxon test (W). These two tests are algebraically equivalent tests, and are related by the following equality when there are no ties in the data:

$$U + W = \frac{m(m + 2n + 1)}{2}$$

Case Studies

In this section, we consider two datasets as case studies. The first is the Clinical depression dataset downloaded from <http://bolt.mph.ufl.edu> (for instance, see [Table 7](#)). The depression is the most common mental illness in the United States, affecting 19 million adults each year (Source: NIMH, 1999). Nearly 50% of individuals who experience a major episode will have a recurrence within 2–3 years. In a study conducted by the National Institutes of Health, 109 clinically depressed patients were separated into three groups, and each group was given one of two active drugs (imipramine or lithium) or no drug at all. For each patient, the dataset contains the following characteristic or variables:

Hospt: The patient's hospital, represented by a code for each of the 5 hospitals (1, 2, 3, 5, or 6).

Treat: The treatment received by the patient (Lithium, Imipramine, or Placebo).

Outcome: Whether or not a recurrence occurred during the patient's treatment (Recurrence or No Recurrence).

Time: Either the time (days) till recurrence, or if no recurrence, the length (days) of the patient's participation in the study.

AcuteT: The time (days) that the patient was depressed prior to the study.

Age: The age of the patient in years, when the patient entered the study.

Gender: The patient's gender (1=Female, 2=Male).

Using these data, researchers are interested in comparing therapeutic solutions that could delay or reduce the incidence of recurrence.

In the second dataset a researcher designed an experiment to assess the effects of prolonged inhalation of cadmium oxide. Fifteen laboratory animals served as experimental subjects, while 10 similar animals served as controls. The variable of interest was hemoglobin level following the experiment. The results are shown in [Table 8](#). We wish to know if we can conclude that prolonged inhalation of cadmium oxide reduces hemoglobin level.

Results

In this section, we describe the main results obtained from the descriptive and inferential analysis using the information contained [Tables 7](#) and [8](#).

The first dataset (Clinical depression dataset) is composed by four categorical variables (Hospt, Treat, Outcome, Gender) and two numerical variables (Time, Time, AcuteT). Bar plots for Hospt, Outcome, and Gender variables are plotted for a qualitative analysis of these data (see [Fig. 18](#)). On the contrary, a quantitative descriptive analysis is performed for the variable age of patients. [Table 9](#) summarizes the main results of some statistical measures computed using the formulas shown in [Table 3](#). Finally, the confidence intervals (CI) and hypothesis test for the difference between two population means μ_1 and μ_2

Table 8 Hemoglobin determinations (grams) for 25 laboratory animals

<i>Exposed animals</i>	<i>Unexposed animals</i>
14.4	17.4
14.2	16.2
13.8	17.1
16.5	17.5
14.1	15.0
16.6	16.0
15.9	16.9
15.6	15.0
14.1	16.3
15.3	16.8
15.7	–
16.7	–
13.7	–
15.3	–
14.0	–

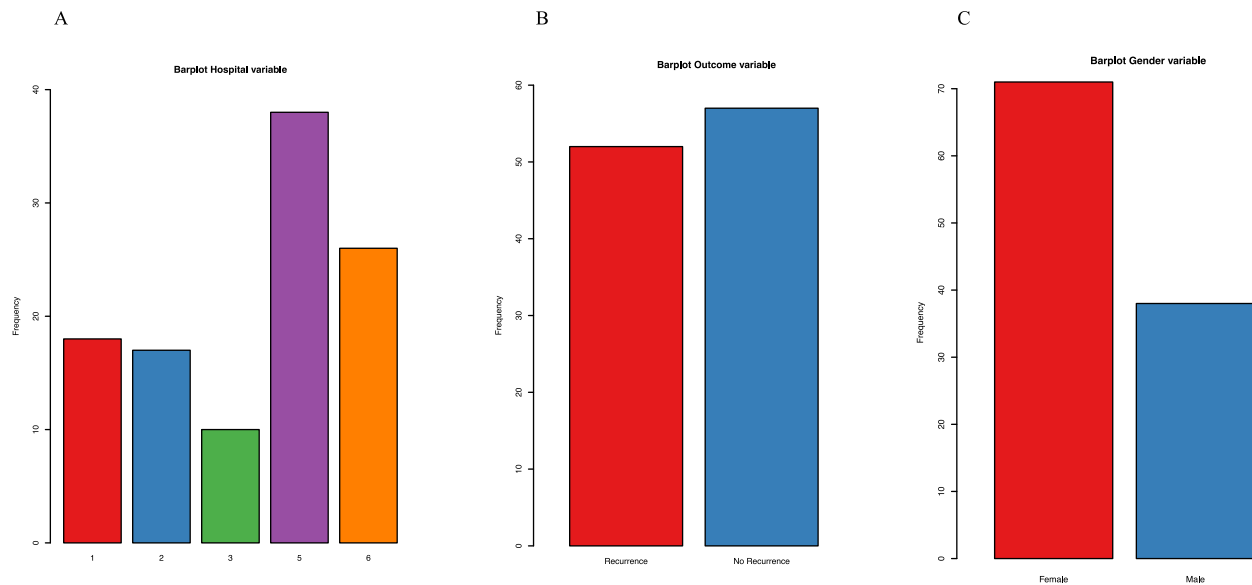


Fig. 18 Bar plots of the qualitative variables: Host (A), Outcome (B) and Gender (B). The first plot indicates that the number of patients from hospital 5 is greater than the others. The second plot shows that the outcome of the treatment for patients with no recurrence exceeds that with recurrence. The third plot displays that the number of female patients is greater than that of males.

Table 9 Data synthesis of patients grouped by age

Class intervals	n_i	f_i	$f_i(\%)$	F_i	c_i	$c_i n_i$	$c_i^2 n_i$	
19 – 125	16	0.15	15	0.15	22	352	7,744	
25 – 131	21	0.20	20	0.34	28	588	16,464	Mode class
31 – 137	14	0.13	13	0.47	34	476	16,184	
37 – 143	11	0.10	10	0.57	40	440	17,600	Median class
43 – 149	15	0.14	14	0.70	46	690	31,740	
49 – 155	15	0.14	14	0.84	52	780	40,560	
55 – 171	17	0.15	15	1	63	1,071	67,473	
Total	109	1	100			4,397	197,765	
Mean	Median	Mode	Variance	Standard deviation	First quartile	Third quartile	IQR	
40.34	38.8	28	187.04	13.68	28.16	51.14	22.98	

are detected using the second listed test statistic illustrated in Table 6. In particular, we test the effect of the treatment (Imipramine) with respect to the control group (Placebo) during the participation of patients in the study (Time). The sample mean estimates are 37.58 and 63.06 for each group under investigated ($n=34$ -Imipramine and $m=38$ -Placebo). The test statistic (two-tailed test) is $t = -2.38$ with 70 degree of freedom, the confidence interval at level $\alpha=0.05$ is $(-46.84, -4.12)$ and the p -value is significant ($p\text{-value}=0.0201 < 0.05$). Hence, the null hypothesis H_0 is rejected which means that the true difference in means is not equal to zero. This means that there is an evidence on the effects of therapy in the treatment of patients with Imipramine during the study.

In the second dataset, we consider the hemoglobin levels (measured in grams) for 25 laboratory animals, divided in two groups: exposed (X) and not exposed (Y) to cadmium oxide. We assume that the assumptions of the Mann-Whitney test are applicable. Therefore, with $n=15$, $m=10$ and $\alpha=0.05$, we find the statistic test (two-tailed test) $T=25$ and the p -value equal to 0.006008 ($p\text{-value} < 0.05$, statistically significant). We conclude that M_X is smaller than M_Y . This leads to the conclusion that prolonged inhalation of cadmium oxide does reduce the hemoglobin level.

Software

We use the R statistical software (see Relevant Websites section) to plot the graphs and to perform the descriptive statistics and statistical inference. In particular, we apply the common used statistical packages in R.

Conclusions

Biostatistics can be defined as the application of mathematics used in statistics to the fields of biological sciences and medicine. When research activities involve data collection on a sample of a population, an understanding of descriptive and inferential analysis become essential for an accurate study of the phenomenon to draw conclusions and make inferences about the entire population. The two major areas of statistics are the descriptive statistics and the inferential statistics. The aim of the first areas is to collect data and obtain a synthesis of this information in order to give a descriptive overview of the data. On the other hand, the goal of the statistical inference is to decide whether the findings of an investigation reflect chance or real effects at a given level of probability. Both estimation and testing hypothesis are covered. These statistical tools are useful for researchers in order to decide what type of study to use for their research project, how to execute the study on patients and well people, and how to evaluate the final results.

See also: Natural Language Processing Approaches in Bioinformatics

References

- Altman, N., Krzywinski, M., 2017. Points of significance: P values and the search for significance. *Nature Methods* 14 (1), 3–4.
- Cochran, W.G., 1964. Approximate significance levels of the Behrens–Fisher test. *Biometrics* 20, 191–195.
- Gardner, M.J., Altman, D.G., 1986. Confidence intervals rather than P values: Estimation rather than hypothesis testing. *British Medical Journal (Clinical Research Edition)* 292 (6522), 746–750.
- Manikandan, S., 2011a. Measures of central tendency: The mean. *Journal of Pharmacology and Pharmacotherapeutics* 2 (2), 140.
- Manikandan, S., 2011b. Measures of central tendency: Median and mode. *Journal of Pharmacology and Pharmacotherapeutics* 2 (3), 214.
- Manikandan, S., 2011c. Measures of dispersion. *Journal of Pharmacology and Pharmacotherapeutics* 2 (4), 315–316.
- Mann, H.B., Whitney, D.R., 1947. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics* 18, 50–60.
- Spiestersbach, A., *et al.*, 2009. Descriptive statistics: The specification of statistical measures and their presentation in tables and graphs. Part 7 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt International* 106 (36), 578–583.
- Wilcox, R.R., Keselman, H.J., 2003. Modern robust data analysis methods: Measures of central tendency. *Psychological Methods* 8 (3), 254.
- Wilcoxon, F., 1945. Individual comparisons by Ranking methods. *Biometrics* 1, 80–83.

Further Reading

- Daniel, W.W., Cross, C.L., 2013. *Biostatistics: A Foundation for analysis in the Health Sciences*, tenth ed. John Wiley & Sons.
- Dehmer, M., Emmert-Streib, F., Graber, A., Salvador, A., 2011. *Applied Statistics for Network Biology: Methods in Systems Biology*. Wiley-Blackwell.
- Dunn, O.J., Clark, V.A., 2009. *Basic Statistics: A Primer for the Biomedical Sciences*. John Wiley & Sons.
- Heumann, C., Schomaker, M., 2016. *Introduction to Statistics and Data Analysis. With Exercises, Solutions and Applications in R*. Springer.
- Hoffman, J.I.E., 2015. *Biostatistics for Medical and Biomedical Practitioners*. Elsevier.
- Indrayan, A., Malhotra, R.K., 2017. *Medical Biostatistics*, fourth ed. Chapman and Hall/CRC.

Relevant Websites

- <https://www.r-project.org>
The R Project for Statistical Computing.
- <http://bolt.mph.ufl.edu>
UF Health: Biostatistics.

Descriptive Statistics

Monica Franzese and Antonella Iuliano, Institute for Applied Mathematics “Mauro Picone”, Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Statistics is a mathematical science for collecting, analyzing, interpreting and drawing conclusions from a set of data. The first instance of descriptive statistics was given by the “bills of mortality” collected by John Graunt’s in 1662. Around 1749, the meaning of statistics was limited to information about “states” to design the systematic collection of demographic and economic data. In the early 19th century, accumulation of information intensified, and the definition of statistics extended to include disciplines concerned with the biology and the biomedicine. The main important areas of statistics are the descriptive statistics and the inferential statistics. The first statistical method gives numerical and graphic procedures to summarise a collection of data in a clear and understandable way without assuming any underlying structure for such data, the second one provides procedures to draw inferences about a population on the basis of observations obtained from samples. Therefore, the use of descriptive and inferential methods enables researchers to summarize findings and conduct hypothesis testing. The descriptive statistics is the primary step in any applied scientific investigation to simplify large amounts of data in a sensible way. Indeed, the goal of descriptive statistics is to give a clear explanation and interpretation of the information collected during an experiment. For instance, in medicine and biology, the observations obtained by a phenomenon are large in number, dispersed, variable and heterogeneous preventing the researcher from directly understanding it. To have a full knowledge of the under investigated phenomenon, it is first necessary to arrange, describe, summarize and visualize the collected data (Spriestersbach *et al.*, 2009).

In this article, we present two statistical descriptive methods: graphical and numerical. The graphs and tables are used to organize and visualize the collected data. Numerical values are computed to summarize the data. Such numbers are called parameters if they describe population; they are called statistics if they describe a sample. The most useful numerical value or statistics for describing a set of observations are the measures of location, dispersion and symmetry. Generally, graphical methods are better suited than numerical methods for identifying patterns in the data, although the numerical approaches are more precise and objective. Since the numerical and graphical approaches complement each other, it is wise to use both. In the following sections, we first introduce the statistical data types (quantitative or qualitative), and then, the way to organize and visualize collected data using tables and graphs. Several kinds of statistics measures (location, dispersion and symmetry) are also discussed to provide a numerical summary of data (Manikandan, 2011a,b,c; Wilcox and Keselman, 2003). Finally, clinical data are elaborated and discussed as an illustrative example.

Statistical Data Types

The goal of descriptive statistics is to gain understanding from data. Population and sample are two basic concepts of statistics. Population can be defined as the set of individuals or objects in a statistical study. While, a sample is a subset of the population from which information is collected. In other words, a statistical population is the set of measurements corresponding to the entire collection of units for which inferences are to be performed, a statistical sample is the set of measurements that are collected in the course of an investigation from the statistical population. Each measurement is defined as statistical unit. This denomination inherited from demography that was the first application field of statistics. A statistical variable is each aspect or characteristic of the statistical unit and it varies from one individual member of the population to another. A statistical variable can be qualitative or quantitative, depending on whether their nature is countable or not. Examples of variables for humans are height, weight, sex, status, and eye color. The first two variables are quantitative variables, the last three are qualitative variables. Quantitative variables can be classified as either discrete or continuous, while qualitative variables can be divided into categorical and ordinal. We define a discrete variable as a finite or countable number of values, while, a continuous variable as a measurement that can take any value in an interval of the real line.

Organization and Visualization of Data

We define each individual or object of data as observation. The collection of all observations for particular variables is called data set or data matrix. Data set is composed by the values of variables recorded for a set of sampling units. Note that in the case of qualitative variable, we assign numbers to the different categories, and thus transform the categorical data to numerical data in a trivial sense. For example, cancer grade can be coded using the values 1, 2 and 3 depending on the amount of abnormality (see Table 1).

Frequency Distribution

Let N be the number of individuals in the population and let X be a variable assuming the values x_i , $i = 1, 2, \dots, k$. We denote with n_i the number of times the value x_i appears in the data set. This value is called absolute frequency of the observed value

Table 1 Tumor data table, with different classification grades

Unit	Gender	Age	Tumor	Grade	Year first diagnosis
1	M	34	Prostate cancer	II	2000
2	M	45	Brain cancer	II	2003
3	F	50	Breast cancer	III	2005
...	...	...	...	...	...
90	F	62	Ovarian cancer	I	2001

Table 2 Frequency distribution

Statistical units i	Values of characteristics x_i	Absolute frequency n_i	Relative frequency f_i	Cumulative absolute frequency N_i	Cumulative relative frequency F_i
1	x_1	n_1	$f_1 = \frac{n_1}{N}$	$n_1 = N_1$	$f_1 = F_1$
2	x_2	n_2	$f_2 = \frac{n_2}{N}$	$n_1 + n_2 = N_2$	$f_1 + f_2 = F_2$
...	...	...	...	...	...
i	x_i	n_i	$f_i = \frac{n_i}{N}$	$n_1 + n_2 + \dots + n_i = N_i$	$f_1 + f_2 + \dots + f_i = F_i$
...	...	...	...	...	...
k	x_k	n_k	$f_k = \frac{n_k}{N}$	$n_1 + n_2 + \dots + n_k = N$	$f_1 + f_2 + \dots + f_k = 1$
Total		N	1		

Table 3 Frequency distribution of tumor grades for 60 patients

Tumor	n_i	f_i
I	36	0,6
II	9	0,15
III	15	0,25
Total	60	1,00

with x_i , and the ratio

$$f_i = \frac{n_i}{N} \quad (1)$$

indicates the relative frequency of the observed value with n_i . In other words, f_i is the proportion on the total population N of individuals presenting the value with x_i . In particular, Eq. (1) satisfies the following conditions:

$$\sum_{i=1}^k n_i = N, \quad \sum_{i=1}^k f_i = 1$$

The observed values of each variable, their absolute and relative frequencies are usually organized in a table called frequency distribution (see Table 2). Sometimes, we are often interested in the percentage frequency that is obtained by dividing the relative frequency by the total number of observations N and multiplying the result by 100. The sum of absolute (or relative) frequencies of all the values equal to or less than the considered value is called cumulative absolute (or relative) frequency. This is represented as N_i (or F_i). If cumulative frequencies are represented in a table then it is called as cumulative frequency distribution (see Table 2). The frequency distribution is useful when data sets are large and the number of different values is not too large. For example, if we consider the brain tumor stage in a sample of 60 patients, then we can use the frequency distribution to compute the absolute and relative frequency (see Table 3).

Qualitative variable

The number of observations that fall into a particular class (or category) of a qualitative variable indicates the frequency (or count) of the class. In this case, cumulative frequencies make sense only for ordinal variables, not for nominal variables. The qualitative data can be represented graphically either as a pie chart or as a horizontal or vertical bar graph. A pie chart is a disk divided into pie-shaped pieces proportional to the relative frequencies of the classes. To obtain angle for any class, we multiply the relative frequencies by 360° , which corresponds to the complete circle. A horizontal bar graph displays the classes on the horizontal axis and the absolute frequencies (or relative frequencies) of the classes on the vertical axis. The absolute frequency (or relative frequency) of each class is represented by vertical bar whose height is equal to the absolute frequency (or relative frequency) of the

Table 4 Frequency distribution based on class intervals. The symbol -| means that only the superior limit is included into the class interval

Class intervals $x_i - x_{i+1}$	Absolute frequency n_i	Relative frequency f_i	Cumulative absolute frequency N_i	Cumulative relative frequency F_i
$x_1 - x_2$	n_1	$f_1 = \frac{n_1}{N}$	$n_1 = N_1$	$f_1 = F_1$
$x_2 - x_3$	n_2	$f_2 = \frac{n_2}{N}$	$n_1 + n_2 = N_2$	$f_1 + f_2 = F_2$
...	...	...	...	...
$x_{i-1} - x_i$	n_i	$f_i = \frac{n_i}{N}$	$n_1 + n_2 + \dots + n_i = N_i$	$f_1 + f_2 + \dots + f_i = F_i$
...	...	...	...	...
$x_{k-1} - x_k$	n_k	$f_k = \frac{n_k}{N}$	$n_1 + n_2 + \dots + n_k = N$	$f_1 + f_2 + \dots + f_k = 1$
Total	N	1		

class. In a bar plot, its bars do not touch each other. In a vertical bar graph, the classes are displayed on the vertical axis and the frequencies (absolute and relative) of the classes on the horizontal axis. Nominal data is best displayed by pie chart and ordinal data by horizontal or vertical bar graph.

Quantitative variable

Quantitative variable can also presented by a frequency distribution. The data of a discrete variable can be summarised in a frequency distribution as in [Table 2](#). Differently, continuous data are first grouped into classes (or categories) and, then collected into a frequency distribution (see [Table 4](#)). The main steps in a process of grouping quantitative variable into classes are:

1. Find the minimum x_{min} and the maximum x_{max} values into the data set.
2. Construct intervals of equal length that cover the range between x_{min} and x_{max} without overlapping. These intervals are called class intervals, and their end points are defined class limits. The magnitude of class interval depends on the range and the number of classes. The range is the difference between x_{max} and x_{min} . A class interval is generally in multiples of 5, 10, 15 and 20. The magnitude of class is given by

$$d = \frac{x_{max} - x_{min}}{K}$$

where K is the number of classes equal to $K = 1 + 3.322 \log_{10}(k)$.

3. Count the number of observations in the data that belongs to each class interval. The count in each class is the absolute class frequency.
4. Calculate the relative frequencies of each class by dividing the absolute class frequency by the total number of observations into the data.

In this case, the information contained in [Table 4](#) can be illustrated graphically using histograms (or vertical bar graph) and cumulative frequency curves. A histogram is a graphical representation of the absolute or relative frequencies for each value of the variables. It is like a horizontal bar graph where the bars are closed each other. In particular, a histogram is composed by rectangles over each class where the area of each rectangle is proportional to its frequency. The base of rectangles are the range of each class, i.e. the difference between x_{i+1} and x_i , for $i = 1, \dots, k$, while the height of the rectangle, called class intensity, is given by

$$h_i = \frac{n_i}{x_{i+1} - x_i}, i = 1, \dots, k$$

A cumulative frequency curve is a plot of the number or percentage of individuals falling in or below each value of the characteristic. If quantitative data is discrete, then the variable should graphically be presented by a bar graph. Also in the case in which the frequency table for quantitative variable is composed by unequal class intervals, the variable can be represented by a bar graph.

Statistical Measures

In this section we present three types of statistical measures that describe and summarise the observed data: the measures of central tendency, the measures of dispersion, and the measures of symmetry. They are called statistics, i.e., functions or modifications of the obtained data. These descriptive measures are a direct consequence of the frequency distribution of the data. In fact, they provide a numerical summary of the frequency distribution.

Measures of Central Tendency

The central tendency of a frequency distribution is a statistical measure that identifies a single value as representative of an entire distribution ([Manikandan, 2011a](#)). It aims to provide an accurate description of the data and it is a numerical value that is most representative of the collected data. The mean, median, quartiles and mode are the commonly used measures of central tendency. The mean is the most used measure of central tendency. The (arithmetic) mean is the sum of all the values $x_1, x_2, \dots, x_k$ in the data

set divided by the number of observations k ($k \leq N$). Denoting the sample mean by $\bar{x}$, it is given by the formula

$$\bar{x} = \frac{1}{k} \sum_{i=1}^k x_i \quad (2)$$

Sometimes the prefix “sample” is dropped, but it is used to avoid the confusion of $\bar{x}$ with the population mean μ on the entire population. In this case, the mean is computed by adding all the values in the data set divided by the number of observations N . The formula is

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (3)$$

In terms of frequency distribution, the sample mean is given by

$$\bar{x} = \frac{1}{k} \sum_{i=1}^k x_i n_i \quad (4)$$

where n_i is the absolute frequency. In the case of class frequency distribution, we first calculate the central value $c_i = \frac{x_{i+1} + x_i}{2}$ and, then, the sample mean $\bar{x}$ with the following formulas

$$\bar{x} = \frac{1}{k} \sum_{i=1}^k c_i n_i \quad (5)$$

The median is a measure of location that separates the set of values in half, so that the data in one half are less than or equal to the median value and the data in the other half are greater or equal to the median value (Manikandan, 2011b). It divides the frequency distribution exactly into two halves. Fifty percent of observations in a distribution have scores at or below the median. Hence, median is the 50th percentile. To compute the median, we first collect the data in increasing order and then determine the middle value in the ordered list. In particular, we can distinguish two cases:

1. If the number of observations is odd, then the median is the value that occupies the middle position into the data. If we let k denote the number of observations, then the sample median is at position $\frac{k+1}{2}$ into the ordered list of data.
2. If the number of observations is even, then the median is the mean of the two middle observations into the ordered data, i.e., the mean of the values at position $\frac{k}{2}$ and $\frac{k}{2} + 1$ into the ordered data.

In the case of (discrete) frequency distribution, we first compute the value $\frac{k}{2}$ and, then we consider the absolute cumulative frequency that is greater than $\frac{k}{2}$, i.e. $N_i \geq \frac{k}{2}$, for $i = 1, 2, \dots, k$. The corresponding value x_i is the median. In terms of grouped value, we first identify the median class as the class that includes the 50% of cumulative relative frequencies, i.e., $F_i \geq 0.50$, for $i = 1, 2, \dots, k$. Then, under the assumption that the frequencies are uniformly distributed, we compute the median M_e as the following approximation

$$M_e \approx x_i + (x_{i+1} - x_i) \frac{0.5 - F_{i-1}}{F_i - F_{i-1}} \quad (6)$$

The quartiles (or percentiles) are location measures that divide the data set into four equal parts, each quartile contains the 25% of the total observations. The first quartile Q_1 (lower quartile) is the number below which lies the 25% of the bottom data. Q_1 is at position $\frac{k+1}{4}$ into the ordered sample data. The second quartile Q_2 is the median of the data. The data are divided into two equal parts, the bottom 50% and the top 50% and it is at position $\frac{k+1}{2}$ into the ordered sample data. The third quartile Q_3 (upper quartile) is the number above which lies the 75% of the top data. Q_3 is at position $\frac{3(k+1)}{4}$ into the ordered sample data. In the case of class frequency, the first quartile class Q_1 is the class that includes the 25% of cumulative relative frequencies, i.e. $F_i \geq 0.25$, for $i = 1, 2, \dots, k$, while the third quartile class Q_3 is the class that includes the 75% of cumulative relative frequencies, i.e., $F_i \geq 0.75$, for $i = 1, 2, \dots, k$. Under the assumption that the frequencies are uniformly distributed, the first quartile class Q_1 and the third quartile class Q_3 are approximately equal to

$$Q_1 \approx x_i + (x_{i+1} - x_i) \frac{0.25 - F_{i-1}}{F_i - F_{i-1}}, Q_3 \approx x_i + (x_{i+1} - x_i) \frac{0.75 - F_{i-1}}{F_i - F_{i-1}} \quad (7)$$

Generally, a frequency distribution can be represented constructing a graph called boxplot (or whisker diagram) which is a standardized way of displaying the data distribution based on five number summary: x_{min} , Q_1 , $Q_2 = M_e$, Q_3 , x_{max} . The central rectangle spans the first quartile to the third quartile (the interquartile range). The segment inside the rectangle shows the median and “whiskers” above and below the box show the positions of the minimum and maximum (see Fig. 1).

The mode (or modal value) of a variable is the value that occurs most frequently in the data (Manikandan, 2011b). It is given by

$$M_o = \{x_i : n_i = \max\}, i = 1, 2, \dots, k \quad (8)$$

The mode may not exist, and even if it does, it may not be unique. This happens when the data set has two or more values of equal frequency that is greater than the other values. The mode is usually used to describe a bimodal distribution. In a bimodal distribution, the taller peak is called the major mode and the shorter one is the minor mode. For continuous data, the mode is the midpoint of the interval with the highest rectangle in the histogram. If the data are grouped into class intervals, than the mode is defined in terms of

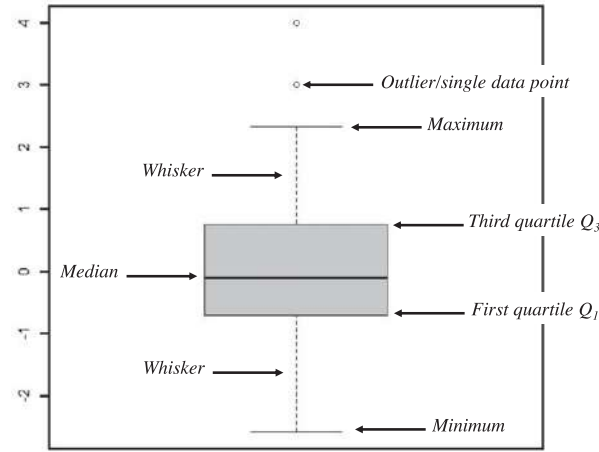


Fig. 1 Boxplot or box and whisker diagram.

class frequencies. With grouped quantitative variable, the mode class is the class interval with highest frequency and it is given by

$$m_c = \{x_i - |x_{i+1} : h_i = \max\} \quad (9)$$

where $h_i = \frac{n_i}{x_{i+1} - x_i}$ is the class intensity. Under the assumption that the frequencies are uniformly distributed, the mode is approximately equal to

$$M_o \approx \frac{x_i + x_{i+1}}{2} \quad (10)$$

We observe that median and mode are not influenced by extreme values or outliers. On the contrary, the mean suffers of them. An outlier is an observation that lies an abnormal distance from other values in a sample from a population. It is a value that exceeds the third quartile by a magnitude greater than $1.5 \times (Q_3 - Q_1)$ or is less than the first quartile by a magnitude greater than $1.5 \times (Q_3 - Q_1)$. In addition, for qualitative and categorical data, the mode can be calculated, while the mean and median do not. On the other hand, if the data is quantitative one, we can use any one of the three averages presented. For symmetric data the mean, the median and the mode can be approximately equal; for skew (or asymmetric) data the median is less sensitive than the mean to extreme observations (outliers). As the mean, the mode and the median have the corresponding population median and population mode, which are all unknown. In fact, the sample mean, the sample median, and the sample mode can be used to estimate the values of these corresponding unknown population values.

Measures of Dispersion

The measures of central tendency are representatives of a frequency distribution but they are not sufficient to give a complete representation of a frequency distribution (Manikandan, 2011c). Two data sets can have the same mean but they can be entirely different. Thus, to describe data, one needs to know also the extent of variability. This is given by the measures of dispersion. A measure of dispersion (called also variability, scatter, or spread) is a statistics that indicates the degree of variability of data. Range, interquartile range, variance, standard deviation and coefficient of variation are the commonly used measures of dispersion. The (sample) range is obtained by computing the difference between the largest observed value $x_{\max}$ of the variable in a data set and the smallest one $x_{\min}$. This measures is easy to compute even if a great deal of information is ignored. In fact, only the largest and smallest values of the variable are considered while the other observed values are disregarded. In addition, the range is always increase, when additional observations are included into the data set which means that the range is overly sensitive to the sample size. The (sample) interquartile range (IQR) is equal to the difference between 75th and 25th percentiles, i.e. the upper and lower quartiles

$$\text{IQR} = Q_3 - Q_1 \quad (11)$$

The (sample) interquartile range represents the length of the interval covered by the center half of the observed values of the data. This measure is not distorted if a small fraction of the observed values are very large or very small.

The variance and the standard deviation are two very popular measures of dispersion. They measure the spread of data across the mean. The population variance σ^2 is the mean of the square of all deviations from the mean. Mathematically it is given as:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (12)$$

where x_i is the value of the i th observation, μ is the population mean and N is the population size. The population standard deviation σ is defined as $\sigma = \sqrt{\sigma^2}$. In terms of frequency distribution, the (sample) variance s^2 is given by

$$s^2 = \frac{1}{k-1} \sum_{i=1}^k (x_i - \bar{x})^2 n_i \quad (13)$$

where x_i is the value of the i th observation, $\bar{x}$ is the sample mean and k is the sample size. In the last formula, the sum of the squared deviations from the mean provides a measure of total deviation from the mean of the data. If k is large the difference between the formulas (12) and (13) is minimal; if k is small, the difference is very sensitive. Generally, for calculations an easier formula is used. The equation of this formula is given by

$$s^2 = \frac{1}{k} \sum_{i=1}^k x_i^2 n_i - \bar{x}^2 \quad (14)$$

where $\bar{x}_2 = \frac{1}{k} \sum_{i=1}^k x_i^2 n_i$ is called second order statistics. This computational formula avoids the rounding errors during calculation. The (sample) standard deviation SD is given by $s = \sqrt{s^2}$. The more variation there is into the data, the larger is the standard deviation. However, the standard deviation does have its drawbacks. For instance, its values can be strongly affected by a few extreme observations. We observe that factor $k-1$ is present in both formulas (13) and (14) instead of k in the denominator, this produces a more accurate estimate of standard deviation.

In the case of class frequency distribution, we first calculate the central value c_i and, then the (sample) variance s^2 given by

$$s^2 = \frac{1}{k-1} \sum_{i=1}^k (c_i - \bar{x})^2 n_i \left(\text{or } s^2 = \frac{1}{k} \sum_{i=1}^k c_i^2 n_i - \bar{x}^2 \right) \quad (15)$$

When the two distributions are expressed in the same units and their means are equal or nearly equal, the variability of data can be compared directly by using the relative standard deviations. However, if the means are widely different or if they are expressed in different units of measurement, we cannot use the standard deviations as such for comparing the variability of both data. Therefore, we use as measure of dispersion the coefficient of variation (CV) that is the ratio of the standard deviation to the mean. The population CV is given by

$$CV = \frac{s}{|\bar{x}|} \quad (16)$$

Formula (16) for population is the ratio of the population standard deviation σ and the population mean μ . The CV is a unit-free measure and it is always expressed as percentage. The CV is small if the variation is small and it is unreliable if the mean is near zero. Hence, if we consider two groups, the one with less CV is said to be more consistent.

Measures of Symmetry

An important aspect of the data is the shape of its frequency distribution. Generally, we are interested to observe if the frequency distribution can be approximated by the normal distribution ($\mu = M_e$). The normal distribution is a continuous random variable and its density curve is symmetric, bell-shaped curve and characterised by its mean μ and standard deviation σ (Fig. 2).

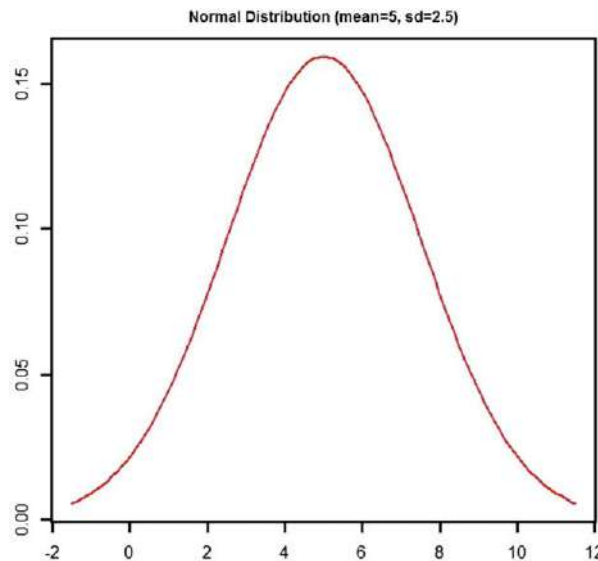


Fig. 2 Normal distribution with mean=5 and standard deviation=2.5.

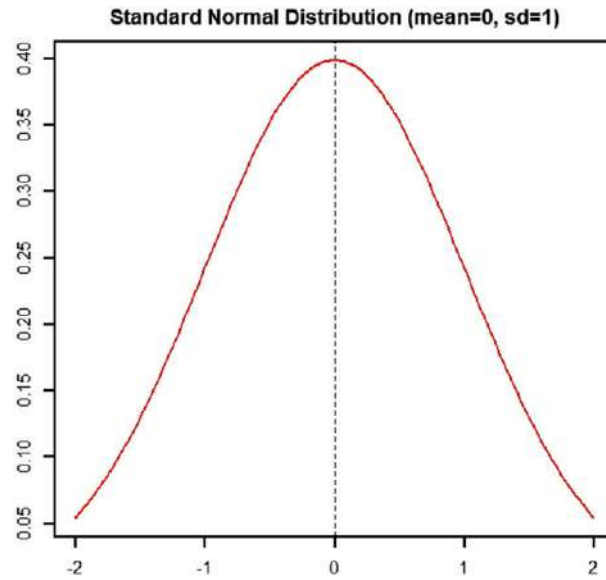


Fig. 3 Standard normal distribution with mean=0 and standard deviation=1.

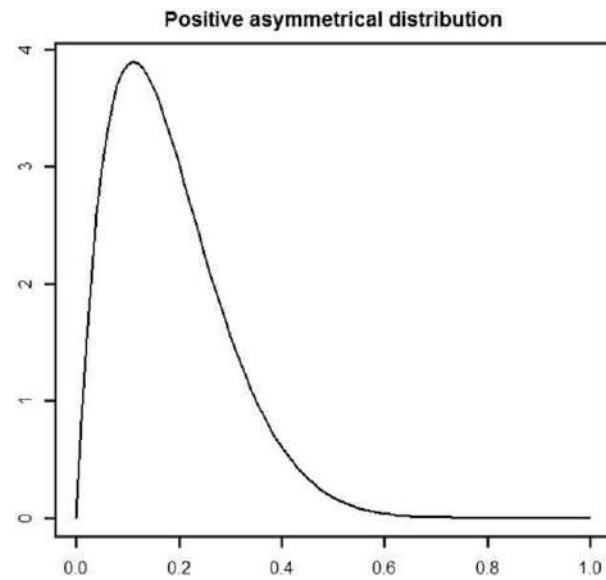


Fig. 4 Positively skewed distribution (to the right).

A continuous random variable follows a standard normal distribution if the variable is normally distributed with mean $\mu=0$ and standard deviation $\sigma=1$ (see Fig. 3). Two important measures of shape are skewness and kurtosis. The first measure is the deviation of the distribution from symmetry (departure from horizontal symmetry), the second one measures the peakedness of the distribution (how tall and sharp the central peak is, relative to a standard bell curve). In particular, if the skewness is different from zero, then that distribution is asymmetrical, while normal distributions are perfectly symmetrical. If the kurtosis is not equal to zero, then the distribution is either flatter or more peaked than normal. We define the moment coefficient of skewness of a sample as

$$\gamma_1 = \frac{\frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^3}{\left(\frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^2 \right)^{3/2}} \quad (17)$$

where $\bar{x}$ is the mean and k is the sample size, as usual. The numerator is called the third moment of the variable x . The skewness can also be computed as the average value of z^3 , where z is the familiar z-score, i.e. $z = \frac{x - \bar{x}}{\sigma}$. In terms of frequency distribution, the

skewness is

$$\gamma_1 = \frac{\sqrt{k}}{(k-1)\sqrt{k-1}s^3} \sum_{i=1}^k (x_i - \bar{x})^3 n_i \quad (18)$$

where s is the sample standard deviation and k the sample size. Similarly, for class frequency distribution, the skewness is

$$\gamma_1 = \frac{\sqrt{k}}{(k-1)\sqrt{k-1}s^3} \sum_{i=1}^k (c_i - \bar{x})^3 n_i \quad (19)$$

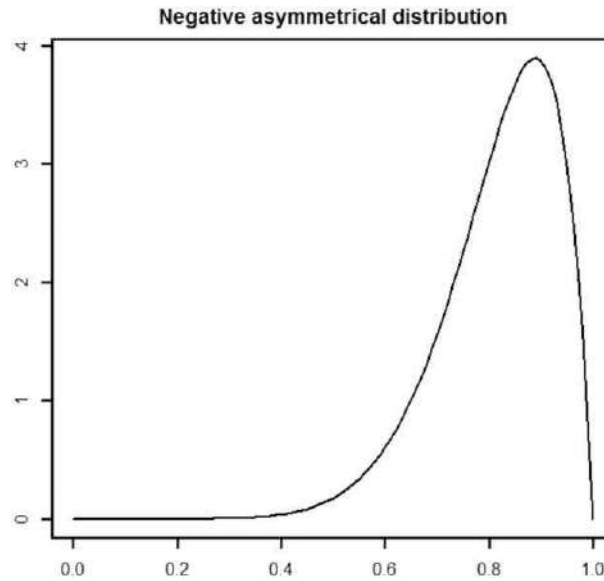


Fig. 5 Negatively skewed distribution (to the left).

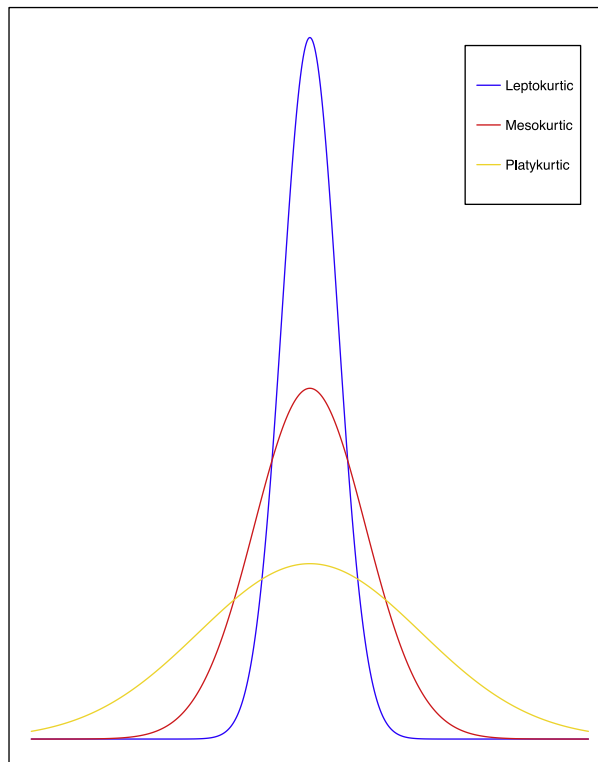


Fig. 6 Kurtosis distributions: a distribution with kurtosis equal to zero is called mesokurtic or mesokurtotic (line red); a distribution with positive kurtosis is called leptokurtic, or leptokurtotic (line blue); a distribution with negative kurtosis is called platykurtic, or platykurtotic (line gold).

Negative values of γ_1 indicate that the data are skewed left, positive values of γ_1 show that the data are skewed right. By skewed left, we mean that the left tail is long relative to the right tail (see Fig. 4). Similarly, skewed right means that the right tail is long relative to the left tail (see Fig. 5). If the data are multi-modal, then this may affect the sign of the skewness. An alternative formula is the Galton skewness (also known as Bowley's skewness)

$$\gamma_1 = \frac{Q_1 + Q_3 - 2Q_2}{Q_3 - Q_1} \quad (20)$$

where Q_1 is the lower quartile, Q_3 is the upper quartile, and Q_2 is the median.

Table 5 Simulated data. The dataset contains the following variables: Age, Sex (M= male, F= female), Alcohol (1= yes, 0= no), Smoke (1= yes, 0= no) and Nationality (Italian and Asiatic) of 50 patients

	<i>Age</i>	<i>Sex</i>	<i>Alcohol 1= yes, 0= no</i>	<i>Smoke 1= yes, 0= no</i>	<i>Nationality</i>
Patient 1	50	M	1	0	Italian
Patient 2	55	F	0	1	Italian
Patient 3	25	F	0	1	Asiatic
Patient 4	30	M	1	1	Asiatic
Patient 5	37	M	0	1	Asiatic
Patient 6	52	M	0	1	Italian
Patient 7	42	M	0	1	Italian
Patient 8	60	F	1	0	Asiatic
Patient 9	41	F	1	0	Asiatic
Patient 10	38	F	1	0	Asiatic
Patient 11	62	M	0	0	Asiatic
Patient 12	65	M	0	0	Asiatic
Patient 13	32	M	0	0	Asiatic
Patient 14	33	M	0	1	Italian
Patient 15	50	M	0	1	Italian
Patient 16	44	M	1	1	Italian
Patient 17	45	M	0	0	Italian
Patient 18	50	M	0	1	Asiatic
Patient 19	60	M	0	1	Asiatic
Patient 20	37	M	1	0	Asiatic
Patient 21	56	F	0	0	Italian
Patient 22	57	F	0	0	Italian
Patient 23	48	M	0	1	Asiatic
Patient 24	50	F	1	1	Asiatic
Patient 25	27	M	0	1	Asiatic
Patient 26	43	F	1	0	Italian
Patient 27	53	F	0	0	Italian
Patient 28	30	M	0	0	Asiatic
Patient 29	25	F	0	0	Italian
Patient 30	28	M	1	1	Asiatic
Patient 31	50	M	1	1	Asiatic
Patient 32	69	F	1	1	Asiatic
Patient 33	50	M	1	1	Italian
Patient 34	50	M	1	1	Italian
Patient 35	50	M	1	0	Italian
Patient 36	50	M	1	0	Italian
Patient 37	50	M	0	0	Asiatic
Patient 38	50	F	0	1	Asiatic
Patient 39	50	F	0	1	Italian
Patient 40	39	F	0	1	Italian
Patient 41	50	M	0	1	Italian
Patient 42	46	F	0	1	Italian
Patient 43	50	F	0	1	Asiatic
Patient 44	50	F	0	0	Asiatic
Patient 45	50	M	0	0	Asiatic
Patient 46	67	M	0	0	Italian
Patient 47	70	M	1	0	Asiatic
Patient 48	61	M	0	0	Italian
Patient 49	68	F	1	1	Italian
Patient 50	45	M	0	1	Asiatic

The kurtosis can be explained in terms of the central peak. Higher values indicate a higher, sharper peak; lower values indicate a lower, less distinct peak. As skewness involves the third moment of the distribution, kurtosis involves the fourth moment and it is defined as

$$\gamma_2 = \frac{\frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^4}{\left(\frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^2 \right)^2} \quad (21)$$

Similarly, in terms of frequency distribution, the kurtosis is

$$\gamma_2 = \frac{k(k+1)}{(k-1)(k-2)(k-3)s^4} \sum_{i=1}^k (x_i - \bar{x})^4 n_i \quad (22)$$

$$\gamma_2 = \frac{k(k+1)}{(k-1)(k-2)(k-3)s^4} \sum_{i=1}^k (c_i - \bar{x})^4 n_i \quad (23)$$

Usually, kurtosis is quoted in the form of excess kurtosis (kurtosis relative to normal distribution kurtosis). Excess kurtosis is simply kurtosis less 3. In fact, kurtosis for a standard normal distribution is equal to three. There are three different ways to define the kurtosis. A distribution with excess kurtosis equal to zero (and kurtosis exactly 3) is called mesokurtic, or mesokurtotic. A distribution with positive excess kurtosis (and $\gamma_2 > 3$) is called leptokurtic, or leptokurtotic. A distribution with negative excess kurtosis (and $\gamma_2 < 3$) is called platykurtic, or platykurtotic. For instance, see Fig. 6. In terms of shape, a leptokurtic distribution has fatter tails while a platykurtic distribution has thinner tails. Finally, if you have the whole population, then γ_1 and γ_2 are the measure of skewness, formula (17), and kurtosis, formula (21).

Data Analysis and Results

In this section, we describe an illustrative example and the results obtained by using the main descriptive tools and measures. We simulate a clinical random dataset composed by 50 patients and different type of variables (quantitative and qualitative). More precisely, the data matrix, in Table 5, shows information about age, sex (M-male; F-female), alcohol (1=yes, 0=no), smoke (1=yes, 0=no) and nationality (Italian, Asiatic) for each patient. Hence, the dataset is composed by four categorical variables (sex, alcohol, smoke, and nationality) and one numerical variable (age). In the first step of our analysis, we plot each variable in order to examine the relative distribution. In particular, we use the pie chart for the qualitative variables alcohol and smoke (see Fig. 7). In the graph A, we observe that the percentage of patients who alcoholic drinks is 36%, while it is 64% for patients do not consume alcoholic drinks. In the graph B, we notice that the percentage of patients who smoke is 54%, while it is 46% for patients do not smoke. Fig. 8 shows the vertical bar plot of the patient's frequencies grouped by nationality, while Fig. 9 displays the horizontal bar plot of the patient's frequencies grouped by gender. The first plot indicates that the community of Asiatic people is greater than of Italian one, while the second graph shows that the males are more numerous than females. In the second step of the analysis, we analyze the quantitative variable: age. We organize the relative data in class frequency. In particular, we first divide the variable age in 7 classes according to the procedure illustrated in Section Quantitative Variable, then, we compute the relative absolute frequencies. Based on this information, formula (1) is applied to calculate the relative frequencies and percentages for each class. The cumulative relative frequency is also computed. Therefore, the table frequency distribution, Table 6, is created. Using the data of this table, the histogram and the cumulative frequency curve of patients grouped by age are plotted (see Figs. 10 and 11).

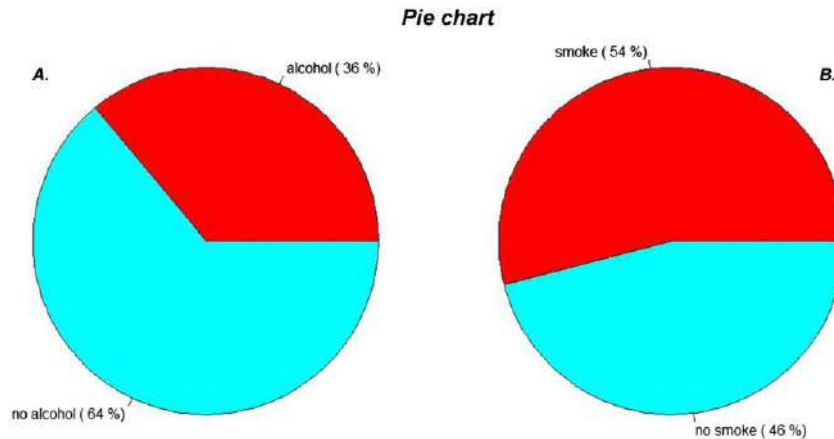


Fig. 7 Pie chart A shows the percentage of *alcohol/no alcohol* patients; pie chart B shows *smoke/no smoke* patients.

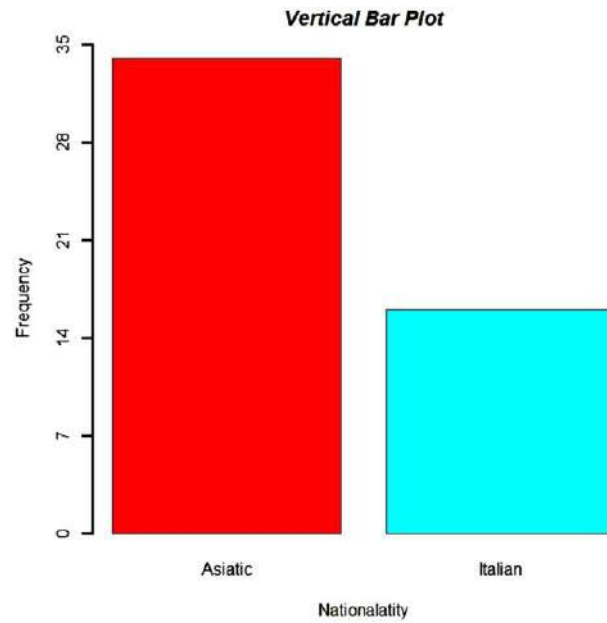


Fig. 8 Vertical bar shows the frequency of patients grouped by nationality.

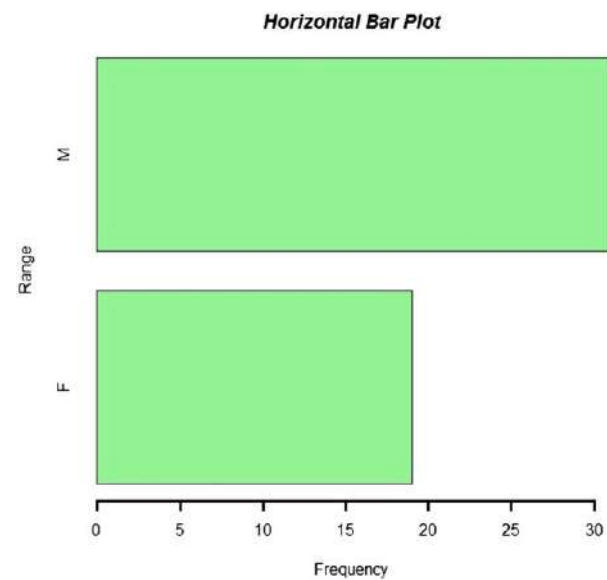


Fig. 9 Horizontal bar shows the frequency of patients grouped by gender.

In addition, the density plot of the variable age grouped in class is shown in [Fig. 12](#). We note that the density shows a bimodal distribution with two different modes that are two different peaks (local maxima). Finally, to conclude our descriptive analysis the most important statistical measures are determined. By considering the information collected in [Table 6](#), we construct the [Table 7](#). In particular, using formula (5), the mean is equal to 47.19, which means that the center of the age distribution is around 47.19 years. The median class of the variable age is the interval $(46,53]$ corresponding to $F_i \geq 0.50$, for $i=1, \dots, 7$. Hence, the median, in according to formula (6), is approximately to 48.21. Using formula (10), the first quartile Q_1 is approximately equal to 39.5 ($F_i \geq 0.25$, for $i=1, \dots, 7$), while the third quartile Q_3 is approximately equal to 52.3 ($F_i \geq 0.75$, for $i=1, \dots, 7$). The mode class is $(46,53]$. Hence, by using formula (9), the mode is approximately equal to 49.5. Median class and mode class coincide.

Finally, we calculate the measure of dispersion for variable age. Using data collected in [Table 7](#), we first compute the range, which is equal to 45. A larger range value indicates a greater spread of the data. Then, by considering formula (11), we calculate the interquartile range IQR, which is equal to 12.8. In addition, the variance and the standard deviation are equal to 132.88 and 11.52, respectively. While the coefficient of variation CV is 0.24. This means that exists a low variability among the data, hence the existing patterns can be seen clearly. To obtain these last results we use formula (15) and (16).

Table 6 Class Frequency Distribution of patients grouped by age. The absolute, relative and cumulative relative frequency are computed

Class intervals	n_i	f_i	$f_i(\%)$	F_i
24 -I 32	7	0.14	14	0.14
32 -I 39	5	0.10	10	0.24
39 -I 46	7	0.14	14	0.38
46 -I 53	19	0.38	38	0.76
53 -I 60	5	0.10	10	0.86
60 -I 67	4	0.08	8	0.94
67 -I 74	3	0.06	6	1
Total	50	1	100	

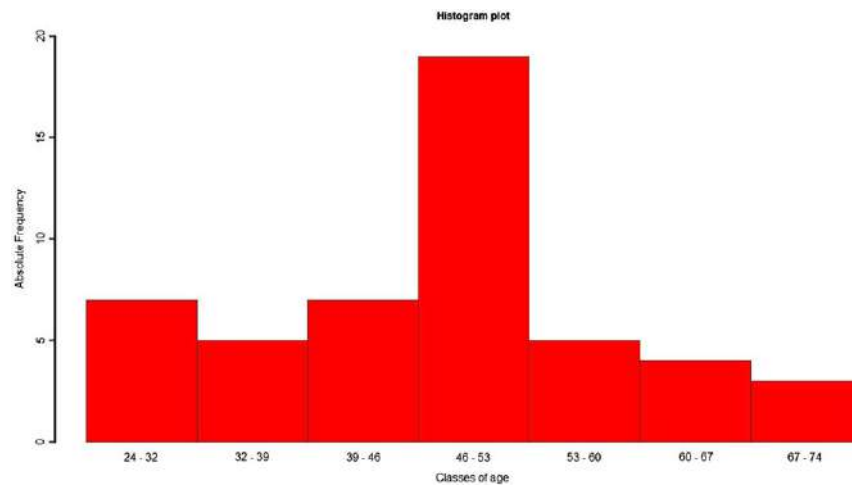


Fig. 10 Histogram shows the absolute frequency of patients grouped for classes of age.

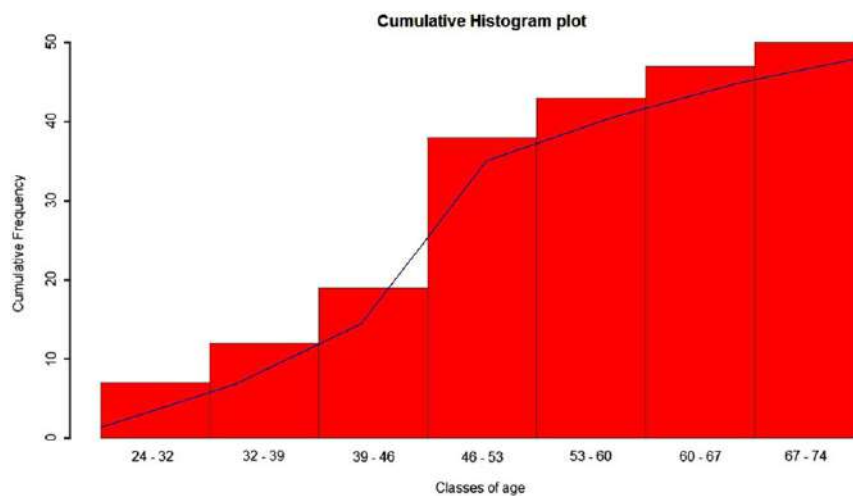


Fig. 11 Cumulative histogram shows the cumulative frequency of patients grouped for classes of age.

Software

We use the R statistical software (see Section Relevant Website) to plot the graphs and to compute the descriptive statistics. In particular, we apply the common used statistical packages in R.

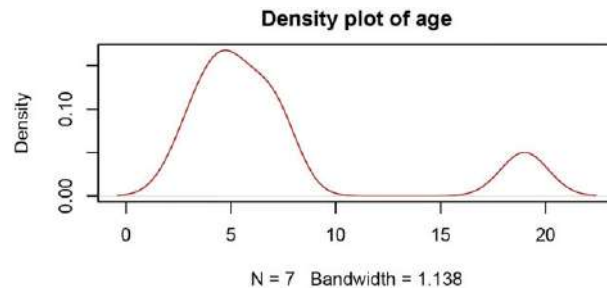


Fig. 12 Density plot of patients grouped by age.

Table 7 Data synthesis of patients grouped by age. The interval (46,53) is the median class. This interval coincides with the mode class

Class intervals	n_i	f_i	$f_i(\%)$	F_i	c_i	c_i/n_i	c_i^2/n_i
24 - 32	7	0.14	14	0.14	28	196	5,488
32 - 39	5	0.10	10	0.24	35.5	177.5	6,301.25
39 - 46	7	0.14	14	0.38	42.5	297.5	12,643.75
46 - 53	19	0.38	38	0.76	49.5	940.5	46,554.75
53 - 60	5	0.10	10	0.86	56.5	282.5	15,961.25
60 - 67	4	0.08	8	0.94	63.5	254	16,129
67 - 74	3	0.06	6	1	70.5	211.5	14,910.75
Total	50	1	100			2359.5	117,988.8

Conclusion

The description of data is the first step for the understanding of statistical evaluations. In fact, if the data are of good quality and well presented, we can draw valid and important conclusions. In this work different descriptive statistical procedures are explained. These include the organization of data, the frequency distribution and the graphical presentations of data. The concepts of central tendency, dispersion and symmetry, called summary statistics, are deeply investigated for a complete exploratory data analysis.

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Manikandan, S., 2011a. Measures of central tendency: The mean. *Journal of Pharmacology and Pharmacotherapeutics* 2.2, 140.
- Manikandan, S., 2011b. Measures of central tendency: Median and mode. *Journal of Pharmacology and Pharmacotherapeutics* 2.3, 214.
- Manikandan, S., 2011c. Measures of dispersion. *Journal of Pharmacology and Pharmacotherapeutics* 2 (4), 315–316.
- Spiersbach, Albert, *et al.*, 2009. Descriptive statistics: The specification of statistical measures and their presentation in tables and graphs. Part 7 of a Series on Evaluation of Scientific Publications. *Deutsches Ärzteblatt International* 106.36, 578–583.
- Wilcox, R.R., Keselman, H.J., 2003. Modern robust data analysis methods: Measures of central tendency. *Psychological Methods* 8 (3), 254.

Further Reading

- Box, G.E., Hunter, W.G., Hunter, J.S., 1978. *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, vol. 1. New York: Wiley.
- Daniel, W.W., Cross, C.L., 2013. *Biostatistics: A Foundation for Analysis in the Health Sciences*, 10th edition John Wiley & Sons.
- Dunn, O.J., Clark, V.A., 2009. *Basic Statistics: A Primer for the Biomedical Sciences*. John Wiley & Sons.

Relevant Website

<https://www.r-project.org>
The R Project for Statistical Computing.

Measurements of Accuracy in Biostatistics

Haiying Wang, Jyotsna Wassan, and Huiru Zheng, Ulster University, Newtownabbey, Northern Ireland, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biostatistics is essentially the application of statistics to problems in biology and medicine. It has been playing an important role in a wide range of applications. Examples include the design of biological experiments, evaluation of the spread of a disease (Deardon *et al.*, 2014), modelling and hypothesis generation (Shmueli, 2010).

Biological data (both quantitative or qualitative) are highly variable and diverse. Range, mean deviation, standard deviation (SD), and coefficient of variation serve as measures of variability for biological samples (Bhuyan *et al.*, 2016). Range is defined as the difference between the highest and the lowest figures. Mean deviation is the average value of the deviations from the arithmetic mean of a biological sample. SD depicts the root mean square of the mean deviation. Coefficient of variation is a measure for comparing relative variability in biological sample. Some of the fundamental concepts in biostatistics as listed by Lopes *et al.* (2014), for quality measure are indicated in Fig. 1 and described below.

- Error indicates deviation of the biological results from the “true” value. Random deviations/errors are quantifiable with standard deviation and systematic deviations/errors are quantifiable with “difference of mean values”.
- Accuracy measure the closeness of obtained biological results to the “true” value.
- Precision serves as the measure of dispersion around the mean value and may be represented by standard deviation or range.
- Bias is the measure of consistent deviation of biological results from “true value” caused by systematic errors.

The biological studies often demand performance analysis by comparison of data (for example, results obtained for a case or control samples). The statistical tools to quantify such biological studies are the correlation-tests, the t-tests and regression analysis etc. The tests are dependent on probabilistic values, which represent the chance of occurrence of an event. The commonly used statistical tests are enlisted in Section “Relevant Statistical Tests for Significance of Biological Results”.

Recent advances in data generation and computer technology have a great impact on biostatistics (see “Relevant Websites section”). For example, the ability to generate data on a high-throughput scale leads to the accumulation of tremendous amounts of biological data. One major application has been in the realm of classification aiming to design a prediction model. An important contribution from biostatistics is an appreciation of the variability of the classification results and the need for good experimental design (Stuart and Barnett, 2006). Thus there is a need to understand metrics used to evaluate the performance of a predictor, which is generally assessed by the extent to which the correct class labels have been assigned. From the biological point of view, it is important not only to examine how many cases have been correctly classified in relation to a particular class, but also to indicate how well a classifier can classify an unknown case as not belonging to such a class. A standard tool used is a confusion matrix (Kohavi and Provost, 1998), from which a number of widely used metrics can be derived.

Confusion Matrix

A confusion matrix is a specific table allowing visualisation of the performance of a classification algorithm. While each row contains information associated with actual classifications, each column indicated the number of predicted classification done by a classification model.

Let TP denote the number of true positives (positive samples correctly classified as positive), FN denote the number of false negatives (positive samples incorrectly classified as negative), FP represent the number of false positives (negative samples incorrectly classified as positive), and TN represent the number of true negatives (negative samples correctly classified as negative). The confusion matrix (Kohavi and Provost, 1998) for a binary classifier is shown in Table 1. As can be seen, such a representation allows visual inspection of prediction errors with little effort as they are clearly located outside the diagonal of the table.

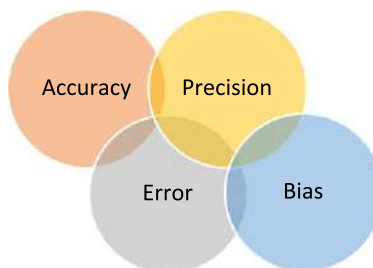


Fig. 1 Fundamentals for biostatistics.

Table 1 A confusion matrix for two possible outcomes positive and negative

		<i>Predicted</i>	
		<i>Positive</i>	<i>Negative</i>
Actual	Positive	TP	FN
	Negative	FP	TN

A number of widely-used statistical measures are often derived from a confusion matrix including:

- Accuracy (Ac), which is defined as the proportion of correct classifications among the total number of samples examined and can be computed using the following equation (Eq. (1)).

$$Ac = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

- Precision (Pr), which is defined as the ratio of true positives to all predicted positives and can be derived using Eq. (2).

$$Pr = \frac{TP}{TP + FP} \quad (2)$$

- Sensitivity (Se), which is also called True Positive Rate and Recall. It is defined as the proportion of actual positives that are correctly identified as positives which can be computed using Eq. (3). Examples include the percentage of patients that are correctly identified as having the condition.

$$Se = \frac{TP}{TP + FN} \quad (3)$$

- Specificity (Sp), which is also called True Negative Rate. It is defined as the proportion of actual negatives that are correctly identified as negatives as shown in Eq. (4). Examples include the percentage of healthy population correctly identified as healthy.

$$Sp = \frac{TN}{TN + FP} \quad (4)$$

- F-score or F measure which is defined as the harmonic mean of precision and recall (Eq. (5)). As can be seen from Eq. (5), it equally weights precision and recall and reaches the highest value of 1 when precision and recall are equal.

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

- *Receiver Operating Characteristic (ROC) curve and Area Under the ROC Curve*

A ROC space is defined by false positive rate (1 – specificity) and true positive rate (sensitivity) as x and y axes as depicted in 2. Each point in the space such as A, B, and C corresponds to an instance of a confusion matrix representing a classification result. A ROC curve is generated by connecting data points (1 – specificity, Sensitivity) obtained at different threshold levels for a given classification model.

As can be seen, the ROC space shown in Fig. 2 highlights some key properties:

1. The top left point with coordinate (0, 1) represents the perfect classification with 100% sensitivity and 100% specificity.
2. The diagonal dashed line represents prediction based on a completely random guess. Any points above the diagonal indicates a better-than-random performance while appearing in the lower right triangle suggesting the classifier performs worse than random guessing (Fawcett, 2006).
3. The ROC curve represents a tradeoff between sensitivity and specificity. Any increase in sensitivity might occur at the cost of a decrease in specificity.
4. The area under the curve (AUC) is a combined measure of sensitivity and specificity, reflecting the overall performance of a classification model. The larger the AUC value is, the better the overall performance is. While theoretically the AUC value lies between 0 and 1, the practical lower limit for the AUC is 0.5 as any classifier that has a worse-than-random performance, e.g., lies in the lower right triangle can be negated to produce a point in the upper left triangle (Park et al., 2004; Fawcett, 2006).

As an illustration, suppose a statistical classification model is to be built using gene expression data to predict the presence of a disease (positive class). In total, the model made 100 predictions, 45 of which were predicted as positives, e.g., having the disease. In reality, a total of 50 were known to have the disease, 40 of which were found in predicted positives. The resulting confusion matrix is illustrated in Table 2. The corresponding value of TP, FP, TN, and FN is 40, 5, 45, and 10 respectively. Thus, the value of Ac, Pr, Se, Sp, and F-score are 85%, 88.9%, 80%, 90%, and 0.84 respectively. The performance of the model can be represented by Point A (0.11, 080) in Fig. 2.

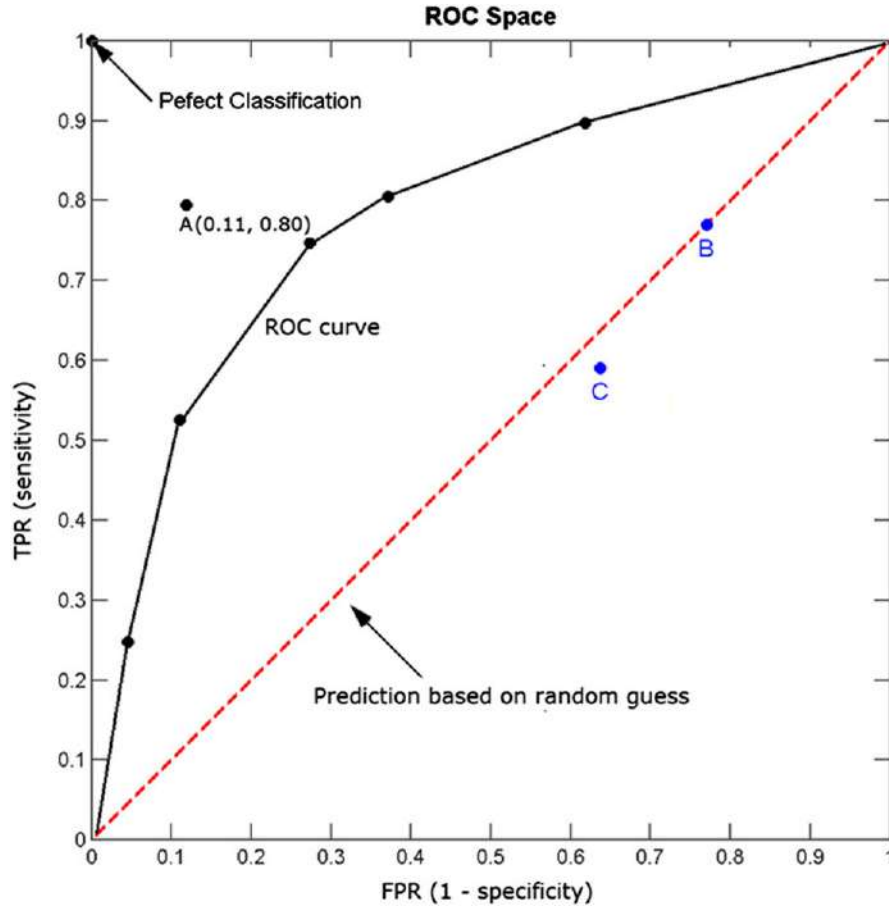


Fig. 2 An illustration of a ROC curve.

Table 2 An example confusion matrix for a binary classification

		Predicted	
		Positive	Negative
Actual	Positive	40	10
	Negative	5	45

Application to Multi-Class Domains

For a multi-class prediction problem with c classes, a $c \times c$ confusion matrix, $Z = (z_{ij})$, is constructed, where z_{ij} represents the number of samples predicted as members of class j while belonging in reality to class i .

Let $x_i = \sum_{1 \leq j \leq c} z_{ij}$ be the number of input samples associated with class i , and $y_j = \sum_{1 \leq i \leq c} z_{ij}$ be the number of inputs predicted to be in class j . The total number of samples is then $N = \sum_{1 \leq i, j \leq c} z_{ij} = \sum_{1 \leq i \leq c} x_i = \sum_{1 \leq j \leq c} y_j$ and the overall accuracy can be defined as: $Ac(\%) = \frac{\sum_{1 \leq i \leq c} z_{ii}}{N} \times 100$. For a class i , its individual precision, sensitivity and specificity can be written as:

$$Pr_i(\%) = \frac{z_{ii}}{y_i} \times 100 \quad (6)$$

$$Se_i(\%) = \frac{z_{ii}}{x_i} \times 100 \quad (7)$$

$$Sp_i(\%) = \frac{\sum_{\substack{k \neq i \\ j \neq i}} z_{kj}}{\sum_{\substack{k \neq i \\ 1 \leq j \leq c}} z_{kj}} \times 100 \quad (8)$$

Turning to multi-class ROC curve, things become more complex as we now need to manage c correct classification and $c^2 - c$ possible errors. One possible solution is to produce a ROC curve for each class. Thus, a total of c ROC curves are generated, each using class c_i as the positive class and the union of the remaining $(c - 1)$ classes as the negative class. While the formulation is straightforward, the solution may compromise one of key advantages exhibited by ROC curve, e.g., insensitive to class skew (Fawcett, 2006). Lane (2000) outlined key issues when extending ROC analysis to multi-class domains.

Application to Imbalanced Classification

The imbalanced classification problem is concerned with the performance of classifiers in handling highly skewed datasets in which one class significantly out represents another (He and Garcia, 2009) and is common in computational biology and bioinformatics. For example, the number of non-interacting protein pairs far outnumber interacting proteins (Jansen *et al.*, 2003). Clearly, any performance metric calculated using information from both rows in confusion matrix (actual positives and negatives) illustrated in Table 1 such as accuracy, precision and F-measure will be inherently sensitive to class distribution thus may not be suitable in the case of imbalanced classification. Utilizing two single-column-based metrics, e.g., True Positive Rate and False Positive Rate, the ROC curve assessment provides a more credible evaluation in the presence of highly skewed data (Fawcett, 2006; He and Garcia, 2009).

However, it has been suggested that when applied to biological domain, the significance of AUC values needs to be interpreted with caution as the majority of the AUC of a ROC curve may not represent biologically meaningful predictions (Scott and Barton, 2007; Browne *et al.*, 2010). Rather than measuring the AUC under the entire ROC curve, it may be more informative to consider the area under a portion of the curve referred to as the partial ROC (Collins *et al.*, 2007). For example, Browne and her colleagues developed a knowledge-driven Bayesian network (KD-BN) for the prediction of protein–protein interaction networks in which a ROC curve was constructed based on identification of TPs and FPs against specific likelihood cutoff ratios (Browne *et al.*, 2010). They compared the proposed KD-BN to a Naive Bayesian (NB)-based model using both the whole and partial ROC curves as shown in Fig. 3. No significant improvement was observed when comparing the KD-BN with NB classifiers when using the traditional ROC curve. However, as it has been estimated that only 1 in ~600 possible protein pairs interact in yeast (Jansen *et al.*, 2003), the results derived with a likelihood ratio less than 600 may not be biologically relevant and thus a large portion of the ROC curve shown in Fig. 3 may not represent biologically relevant results. This is indeed the case when examining the partial ROCs in which only the predictions made by using a selected threshold (600 for yeast based on posterior odds of an interacting protein pair) were included in the calculation of the area of the ROC. As highlighted in the inset of Fig. 3, an improvement can be observed for the KD-BN approach when the partial ROC curves are employed. This can be further confirmed with the TP/FP ratio at the corresponding likelihood ratio (Browne *et al.*, 2010). For instance, a TP/FP rate of 4.5 was obtained using the NB with the threshold set at 600. A much higher TP/FP ratio was obtained at the same threshold by the KD-BN.

Relevant Statistical Tests for Significance of Biological Results

The term “test of significance” was given by statistician Ronald Fisher (Fisher, 1929). The significance test starts with a claim to be compared. The claim tested by a statistical test is called the null hypothesis (H_0). Significance is reported as ‘p’ value

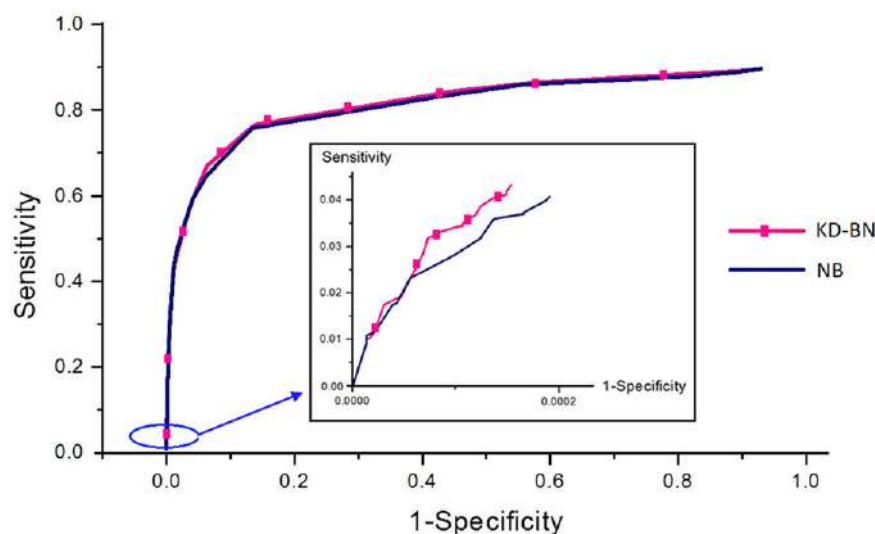


Fig. 3 Illustration of the classification performance of KD-BN and NB when inferring PPIs in yeast using both whole and partial ROC curves (inset).

(i.e., probability value). Thus p value account for differences in the sample estimates. The test of significance validates the inferences drawn from biological observations. The cutoff value for p is known as alpha, or the significance level.

- If the p-value is larger than alpha, the null hypothesis is accepted and any alternatives are rejected.
- If the p-value is smaller than alpha, the null hypothesis is rejected and any alternative is accepted.

The alpha value is typically set at 0.05 (or 5%) representing the amount of acceptable error, or the probability of rejecting a null hypothesis. The Handbook of Biological Statistics evolved at the University of Delaware, is very useful for biologists to choose statistical tests for validating significance (see “Relevant Websites section”).

The statistical analysis is useful in variety of biological applications. For example, [Storey and Tibshirani \(2003\)](#) used statistical significance for genomewide studies. [Cui and Churchill \(2003\)](#) reported various statistical tests for differential expression in cDNA microarray experiments, [Mitchell-Olds and Shaw \(1987\)](#) used regression analysis for biological interpretation of natural selection. Recently, [Schurch et al. \(2016\)](#) provided an insightful study for determining biological replicates to ensure valid biological interpretation of the results and which statistical tools are best for analyzing biological data.

Correlation is useful in analyzing the relationships between two biological quantities and is provided by the Pearson coefficient of correlation, typically denoted by ‘r’ (Eq. (9)).

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (9)$$

where x and y denote the biological sample variables; $\bar{x}$ and $\bar{y}$ are respective sample means and n denotes the total number of observations.

Regression analysis is useful in cause and effect relationships as it helps in predicting the value of dependent variable (y) based on the given value of independent variable (x). Likelihood ratio of a test is used to determine the likelihood of finding a positive or negative test result, for example to determine whether a person is suffering from a diseased condition or not ([Park, 2011](#)).

Various statistical software such as R, SYSTAT, statistical package for the social sciences (SPSS), statistical analysis system (SAS), STATA, biomedical package (BMDP), etc. are useful in statistical analysis ([Tabachnick and Fidell, 1991](#)).

Final Remarks

Biological Statistics is useful in validating the significance of biological conditions and relevance (see “Relevant Websites section”). A confusion matrix is a standard tool used to evaluate the performance of a statistical classification model and various metrics including ROC curves and AUC values can be derived from the matrix. It has been recommended that the interpretation of the values of these metrics should be cautious when applied to biological domain. For example, it might be the case in which a large portion of AUC curves may fail to encode any biologically relevant results ([Scott and Barton, 2007](#); [Browne et al., 2010](#)). Given that the value of a precision depends on the prevalence of a disease, it has been argued that the precision value should not be used as a sole indicator when evaluating a diagnostic model ([Hardesty et al., 2005](#)). Along with standard evaluation techniques, some metrics specifically designed for a application domain could be also considered such as TP/FP ratio as a measure of the probability of a positive protein interaction and TP/P ratio as a measure of coverage ([Jansen et al., 2003](#)).

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Bhuyan, D., Dua, N., Kothari, T., 2016. Epidemiology and biostatistics: Fundamentals of research methodology. *Dysphrenia* 7 (1), 87–93.
- Browne, F., Wang, H.Y., Zheng, H., Azuaje, F., 2010. A knowledge-driven probabilistic framework for the prediction of protein–protein interaction networks. *Computers in Biology and Medicine* 40 (3), 306–317.
- Collins, S.R., Kemmeren, P., Zhao, X.C., et al., 2007. Towards a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Molecular & Cellular Proteomics* 6 (3), 439–450.
- Cui, X., Churchill, G.A., 2003. Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology* 4 (4), 210.
- Deardon, R., Fang, X., Kwong, G.P.S., 2014. Statistical modeling of spatiotemporal infectious disease transmission. In: Chen, D., Moulin, B., Wu, J. (Eds.), *Analyzing and Modeling Spatial and Temporal Dynamics of Diseases*. Hoboken, NJ, USA: John Wiley & Sons, Inc., pp. 221–231.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27 (8), 861–874.
- Fisher, R.A., 1929. Tests of significance in harmonic analysis. *Proceedings of the Royal Society of London Series A, Containing Papers of a Mathematical and Physical Character* 125 (796), 54–59.
- Hardesty, L.A., Klym, A.H., Shindell, B.E., et al., 2005. Is maximum positive predictive value a good indicator of an optimal screening mammography practice? *American Journal of Roentgenology* 184 (5), 1505–1507.
- He, H., Garcia, E., 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21 (9), 1263–1284.
- Jansen, R., Yu, H., Greenbaum, D., et al., 2003. A Bayesian networks approach for predicting protein-protein interactions from genomic Data. *Science* 302 (5644), 449–453.
- Kohavi, R., Provost, F., 1998. Glossary of term. *Machine Learning* 30, 271–274.
- Lane, T., 2000. Extensions of ROC analysis to multi-class domains. In: Dietterich, T., Margineantu, D., Provost, F., Turney, P. (Eds.), *Proceedings of ICML2000 Workshop on Cost-Sensitive Learning*.

- Lopes, B., Ramos, I.C.D.O., Ribeiro, G., *et al.*, 2014. Biostatistics: Fundamental concepts and practical applications. *Revista Brasileira de Oftalmologia* 73 (1), 16–22.
- Mitchell-Olds, T., Shaw, R.G., 1987. Regression analysis of natural selection: Statistical inference and biological interpretation. *Evolution* 41 (6), 1149–1161.
- Park, K., 2011. Park's Textbook of Preventive and Social Medicine. India: Bhanot Publishers.
- Park, S., Goo, J., Jo, C., 2004. Receiver Operating Characteristic (ROC) Curve: Practical review for radiologists. *Korean Journal of Radiology* 5 (1), 11–18.
- Schurch, N.J., Schofield, P., Gierliński, M., *et al.*, 2016. How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA* 22 (6), 839–851.
- Scott, M.S., Barton, G.J., 2007. Probabilistic prediction and ranking of human protein-protein interactions. *BMC Bioinformatics* 8, 239.
- Shmueli, G., 2010. To explain or to predict? *Statistics Science* 25 (3), 289–310.
- Storey, J.D., Tibshirani, R., 2003. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences* 100 (16), 9440–9445.
- Stuart, G.B., Barnett, S.K., 2006. Biostatistics – Why biotechnologists should care about biostatistics. *Asia-Pacific Biotech News* 10 (22), 1275–1278.
- Tabachnick, B.G., Fidell, L.S., 1991. Software for advanced ANOVA courses: A survey. *Behavior Research Methods, Instruments, & Computers* 23 (2), 208–211.

Relevant Websites

<https://en.wikipedia.org/wiki/Biostatistics>

Biostatistics.

<http://www.biostathandbook.com/>

Handbook of Biological Statistics: Introduction.

Hypothesis Testing

Claudia Angelini, Istituto per le Applicazioni del Calcolo "M. Picone", Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A statistical hypothesis is a statement on a population (or on several populations) and hypothesis testing is a classical and well-established framework of statistical inference (Abramovich and Ritov, 2013; Casella and Berger 2001; Lehmann, 1986; Stuart *et al.*, 2008) that allows one to decide, on the basis of an observed samples drawn from that population, which of the two complementary hypotheses to consider as true and, also, to quantify the uncertainty in this decision (Krzywinski and Altman, 2013a). For example, one can establish whether a coin is fair or not after counting the number of heads observed in a series of tosses, or one can assess whether a novel drug is more effective in lowering the cholesterol levels than an existing one, by comparing the reduction of cholesterol in two groups of individuals treated with the two drugs, and so on.

The underlying idea consists of defining an initial assumption (statistical hypothesis), observing sampled data in order to accumulate evidence in favor or against such an initial assumption, and then, given the observed data, deciding whether to reject or not reject the initial assumption.

More formally, a statistical hypothesis is an assertion regarding the distribution(s) of one or several random variables that describe the population(s). When taking a decision two complementary hypotheses are compared: the *null hypothesis* (denoted with H_0) represents the current or conventional experimental condition (e.g., the coin is fair); the *alternative hypothesis* (denoted with H_1) describes the condition one is interested to assess (e.g., the coin is not fair).

The concept of hypothesis can be rather general and may concern the parameters of the given distribution(s) (i.e., say the probability, p , of observing a head in tossing a coin) or other assumptions on the probability distribution(s) of the population(s) under study (e.g., the independence between two random variables).

The choice of which of the two hypotheses to accept is taken after performing a statistical test on samples drawn from the population(s) under study. Specifically, by looking at the value of a test statistic on the observed data, one can decide whether to accept or reject the null hypothesis. A test statistic is a random variable depending on the observations (e.g., the number of observed heads in a series of tosses) and its mathematical form depends on the specific hypotheses one is interested in. Since the value of the test statistic depends on random samples, there is a source of uncertainty in this decisional process. In particular, two types of errors might occur, i.e., rejecting the null hypothesis when it is true (*Type I error*) or accepting the null hypothesis when it is false (*Type II error*). Unfortunately, it is not possible to minimize both types of errors. Therefore, by convention one controls the probability of Type I error at some conventional (*significance*) level, α , and then, sets the decision rule accordingly. In practice, there are two ways to set a decision rule: (i) defining a *critical or rejection region* on the basis of the significance level α , and then rejecting H_0 if the observed test statistic falls in this critical regions, or (ii) computing the *p-value* for the observed data, and then rejecting H_0 if the p-value is smaller than α . Note that, roughly speaking, a rejection of H_0 is also called discovery, since usually a statistical test is designed with the aim of finding evidence for rejecting H_0 .

In brief, within the classical statistical literature, several hypothesis-testing procedures (both parametric and non parametric) have been designed to support researchers in taking suitable decisions under different experimental contexts. Such procedures have been studied in terms of their mathematical properties (i.e., power, etc.) and required assumptions (Abramovich and Ritov, 2013; Casella and Berger 2001; Lehmann, 1986; Stuart *et al.*, 2008) for a wide range of statistical hypotheses and experimental designs. Moreover, many of these tests are implemented in statistical software either as command line functions or within computational platforms with user friendly interfaces.

Nowadays, many inference problems in research areas such as genomics require simultaneously testing a large number (from hundreds to thousands or even millions) of null hypotheses (Sham and Purcell, 2014). Typical examples are the detection of differentially expressed genes from high-throughput assays such as microarrays, the identification of polymorphisms associated with a certain disease, and so on. This framework is known as *multiple comparison* (Dudoit and van der Laan, 2008; Noble, 2009) and represents one of the most widely studied research areas in the last few decades (Benjamini, 2000). In this context, different overall decision errors can be defined and adjustment procedures can be designed to control these specific error types.

In the following, we briefly summarize the key concepts and definitions related to hypothesis testing, moving from the classical approach concerning parametric tests on a single population, to non-parametric tests, to the most relevant procedures for multiple comparisons.

Background/Fundamentals

To fix the notation and introduce the general concepts, assume we have collected a random sample, $X_1, \dots, X_n$, from a population of interest, where X_i are independent and identically distributed random variables (i.e., $X_i \sim f_0(x)$, where f_0 denotes the probability density function or the probability mass function). Moreover, assume that the parameter, $\theta \in \Theta$, is unknown and that we want to perform a statistical test on its value.

The problem can be formulated as follows

$$H_0 : \theta \in \Theta_0 \text{ vs } H_1 : \theta \in \Theta_1$$

where H_0 and H_1 are the so-called *null hypothesis* and *alternative hypothesis*, respectively. The two hypotheses are complementary and mutually exclusive (i.e., $\Theta_0 \cup \Theta_1 = \Theta$ and $\Theta_0 \cap \Theta_1 = \emptyset$), with Θ_0 and Θ_1 representing the two possible values or ranges of values that the parameter θ can assume. If $\Theta_0 = \theta_0$ (or $\Theta_1 = \theta_1$), the corresponding hypothesis is said to be *simple*, otherwise the hypothesis is said to be *composite*. Examples of composite null hypotheses are $\Theta_0 \geq \theta_0$, or $\Theta_0 \leq \theta_0$ (analogously, for the alternative hypothesis). Note that in most of practical applications in biomedicine, at least one of the hypotheses is composite.

A *hypothesis testing* procedure is a decisional rule that, on the basis of the observed sample $\mathbf{x} = (x_1, \dots, x_n)$, can suggest whether to accept H_0 as true, or to reject H_0 in favor of H_1 .

In particular, to decide which of the two hypotheses to accept as true, one has to choose a function, $T(X_1, \dots, X_n)$, known as a *test statistic*, and evaluate it on the sampled data.

In many cases, large values of $t_{\text{obs}} = T(\mathbf{x}) = T(x_1, \dots, x_n)$ can be seen as a measure of the evidence against H_0 ; this means the larger $T(\mathbf{x})$ is, the less likely the observed samples were drawn under the null hypothesis. Therefore, the decision can be taken by looking at $t_{\text{obs}} = T(\mathbf{x})$.

More precisely, one can define a *critical value*, C , such that, if $T(\mathbf{x})$ is greater or equal than C , H_0 will be rejected, otherwise H_0 will be accepted (see Fig. 1, Panel a). In this set up, C divides the range of values that the test statistic, $T(X)$, can assume into two regions: an *acceptance region*, $\mathcal{A} = \{\mathbf{x} : T(\mathbf{x}) \leq C\}$, where one accepts H_0 , and a *critical or rejection region*, $\mathcal{R} = \{\mathbf{x} : T(\mathbf{x}) > C\}$, where one rejects H_0 .

It is worth noting that the partition of the decisional space into the two regions such as those reported in Fig. 1 (Panel a) has to be considered merely descriptive and it is usually associated with a so-called one-sided test. However, the symmetric partition in Fig. 1 (Panel b) could also be possible, as well as the partition shown in Fig. 1 (Panel c), which is usually associated with a so-called two-sided test.

Under the above-mentioned framework, the distributions of the test statistic under H_0 , (i.e., $P_{H_0}(T(X))$), and under H_1 , (i.e., $P_{H_1}(T(X))$), constitute the key points. In particular, we can rewrite $P_{H_0}(T(X))$ as $P_{H_0}(T(X)) = P(T(X)|H_0) = P(\text{Data}|H_0)$ where the notation $P(\text{Data}|H_0)$ has to be interpreted as the probability of the observed data assuming that H_0 is true.

Since the acceptance or the rejection of H_0 is based on the evidence collected from observed data, two types of errors might occur (Table 1): the *Type I error* (also known as the *significance level* of a test) is defined as

$$\alpha = P_{H_0}(\text{reject } H_0) = \underset{\theta \in \Theta_0}{z.drule; z.drule; raise6ptsup} P_{\theta}(T(X) \geq C) = \underset{\theta \in \Theta_0}{z.drule; z.drule; raise6ptsup} P_{\theta}(X \in \mathcal{R})$$

and the *Type II error* is defined as

$$\beta = P_{H_1}(\text{accept } H_0) = \underset{\theta \in \Theta_1}{z.drule; z.drule; raise6ptsup} P_{\theta}(T(X) < C) = \underset{\theta \in \Theta_1}{z.drule; z.drule; raise6ptsup} P_{\theta}(X \in \mathcal{A})$$

More precisely, we also define $\alpha(\theta) = P_{\theta}(X \in \mathcal{R})$ and $\beta(\theta) = P_{\theta}(X \in \mathcal{A})$

Note that Type I and Type II errors are also called the probability of false positives and the probability of false negatives, respectively.

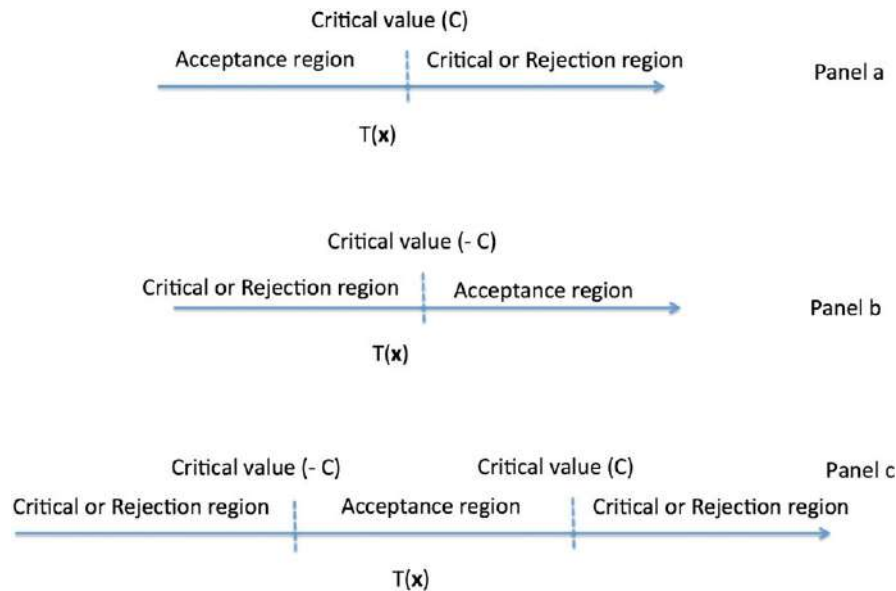


Fig. 1 Graphical representation of acceptance and rejection regions.

Table 1 Errors in hypothesis testing

	H_0 is true	H_1 is true
Decision: Accept H_0	Correct decision $1-\alpha$	Type II error β
Decision: Reject H_0	Type I error α	Correct decision $\pi = 1-\beta$

Then, the *power* of a test, as function of θ , is given by

$$\pi(\theta) = \begin{cases} \alpha(\theta) & \theta \in \Theta_0 \\ 1 - \beta(\theta) & \theta \in \Theta_1 \end{cases},$$

and it represents the probability of rejecting H_0 , or equivalently, the probability of discovering H_1 when it is true.

In principle, the critical value of C should be chosen to minimize the two error types. Unfortunately, it is not possible to simultaneously minimize both types of errors, since when one error decreases the other increases. Moreover, when the alternative is composite, it is difficult to compute P_{H_1} (accept H_0) and then to minimize the sums of the two errors. Therefore, the classical approach to hypothesis testing suggests choosing C so as to guarantee that the Type I error is below a prespecified significance level, α .

Typical significance levels are $\alpha = 0.05$ or $\alpha = 0.01$, although such choices are rather conventional (Wasserstein and Lazar, 2016 and therein). Indeed, the choice of the actual significance level α to be used in a test depends on the impact that false positives have in that specific context, and has to be decided before observing the data.

Significance, power and sample size are strongly interplaying factors in hypothesis testing (Efron, 2007; Krzywinski and Altman, 2013b,c; Nuzzo, 2016), and not accounting for such relationships can lead to incorrect results, waste of resources, and other ethical issues (Ioannidis, 2005). For example, on the one hand, for a fixed sample size, n , decreasing the significance level, α , (say, from 0.05 to 0.01) results in a loss of power for the testing procedure. The lack of power implies that many small effects might not be detected (false negatives) by the testing procedure, and usually only the strongest effects are detected. On the other hand, for a fixed significance level, α , increasing the sample size, n , produces an increase in power. Therefore, balancing sample size, significance, and power becomes a crucial part of experimental design, in particular in omics studies (Krzywinski and Altman, 2013c; Sham and Purcell, 2014).

Another important concept in hypothesis testing is the *p-value*, which can be defined as

$$p\text{-value} = P_{H_0}(T(X)) = t_{\text{obs}} \text{ or } T(X) \text{ more extreme than } t_{\text{obs}} \text{ assuming } H_0 \text{ hold}$$

that is, the probability (under the null hypothesis) of observing a test statistic equal or even more extreme than the one observed on the sample x . Unfortunately, in the so-called “*prosecutor’s fallacy*,” the *p-value* is often misinterpreted as the probability that H_0 is true (Altman and Krzywinski, 2017a,b; Greenland et al., 2016; Krzywinski and Altman, 2013b; Nuzzo, 2015), while it is only a measure of evidence against the null hypothesis. In practice, the smaller the *p-value*, the more “unlikely” the observed data come from the null hypothesis. Therefore, one can still reject H_0 if the *p-value* is smaller or equal to the significance level, α , but he/she has to remember that $P(\text{Data}|H_0) \neq P(H_0|\text{Data})$. In particular, while $P(\text{Data}|H_0)$ is related to the concept of *p-value*, $P(H_0|\text{Data})$ is a posterior probability that is not evaluated within the classical statistical framework. In order to define $P(H_0|\text{Data})$, one has to redefine mathematical hypothesis testing within the Bayesian statistical framework (Berger, 1985).

Systems and/or Applications

The formulation of a statistical hypothesis test can be generalized to more than one statistical population (Krzywinski and Altman, 2014b). However, while the formalism and the definitions do not change, the way in which is possible to design the statistical experiment (i.e., the type of hypothesis and the data collection) might lead one to distinguish different testing procedures. For example, when testing whether two populations have or do not have the same mean, one has to distinguish whether the experimental design is independent or paired. In the case of paired samples, instead of the simple one-sample T-test, one can consider the two-sample T-test, then (depending on the experimental design (Krzywinski and Altman, 2014a)) an independent two-sample T-test, or a paired-two-sample T-test. In the same scenario, when there are more than two populations, the F-test within the ANOVA model could be considered. Obviously, the T-test requires the data to be normally distributed. If such an assumption is not satisfied, non-parametric tests (such as, for example, the Mann–Whitney–Wilcoxon test) have to be considered.

More formally, within hypothesis testing we can distinguish two types of tests: parametric tests and nonparametric tests (Krzywinski and Altman, 2014d).

Parametric Tests

A parametric hypothesis test assumes that observed data are distributed according to distributions of a well-known form (e.g., normal, binomial, and so on) up to some unknown parameter(s) on which we want to make inferences (say the mean, or the success probability). Examples of this type of test are the T-test for comparing mean(s), the binomial test for comparing proportions, and the F-test for ANOVA models, among many others. It is important to understand that the validity of a parametric test depends on the assumptions one can make about the underlying population(s). If the assumptions are satisfied, then the decision rule will control the Type I error at the chosen significance level. Otherwise, the results may be over-optimistic and the decision may not be taken under proper error control. In the case of the classical T-test, the samples, $X_1, \dots, X_n$, are assumed to be drawn from normal populations, with the same variance and that they are uncorrelated. Corrections such as Welch's variant have to be applied in case of unequal variances.

Nonparametric Tests

A nonparametric test is a hypothesis test where it is not necessary (or not possible) to specify the parametric form of the distribution(s) of the underlying population(s). Therefore, they require few assumptions about the population under study, and can be used to test much wider cases of possibilities. Examples of this type of test include the sign test, Mann-Whitney-Wilcoxon test, Chi-square test, Kruskal-Wallis test, and the Kolmogorov-Smirnov test, among many others.

Hypothesis Testing and Other Statistical Approaches

It should be noted that statistical hypothesis testing procedures are not only interesting in their own right, but they can be also combined with other inferential statistical procedures such as regression analysis, model/variable selection, and so on. For example within linear regression, the significance of a specific regression coefficient can be assessed by a hypothesis test.

Testing Procedure Step-by-Step

A testing procedure involves the following steps:

1. Formulate the two hypotheses: the null hypothesis H_0 and the alternative hypothesis H_1 :
2. Decide the significance level, α , for the test.
3. Choose the test statistic, $T(X)$, on the basis of the hypothesis to be tested and the assumptions about the data probability distribution.
4. For the given significance level, α , calculate the critical value, C , of the test statistic under H_0 . Then, deduce the critical region, $\mathcal{R}$, or the acceptance region, $\mathcal{A}$.
5. Collect your data X and compute the observed test statistic $t_{\text{obs}} = T(x)$
6. Take the decision to accept or to reject the null hypothesis on the basis of t_{obs} as follows: if t_{obs} is located in the critical region $\mathcal{R}$, then reject the null hypothesis, H_0 , in favor of the alternative hypothesis H_1 , otherwise accept H_0 .

When the p-value approach is used, after above steps (1)–(3) one has to

4. Collect your data and compute the observed test statistics t_{obs}
5. Compute the *p-value* associated with t_{obs}
6. Take the decision to accept or to reject the null hypothesis on the basis of the *p-value* as follows: if *p-value* $< \alpha$, then reject the null hypothesis, H_0 , in favor of the alternative hypothesis H_1 , otherwise accept H_0 .

Results, Discussion and Advanced Approaches

Reporting Results and Misinterpretation of p-Values

In order to report the result of a statistical test, one should explicitly formulate the null and the alternative hypotheses, the test statistic that was used, and the p-value observed on the sample data. As previously stated, the p-value is one of the most misinterpreted concepts in statistics (Altman and Krzywinski, 2017a,b; Nuzzo, 2015, among many others), in particular in biomedical science. Indeed, in the last few years there has been an increasing debate in the statistical community on the misuse of the p-value concept in many scientific publications (Halsey *et al.*, 2015; Huber, 2016; Lazzeroni *et al.*, 2016; van Helden, 2016, among many others). The reason is that a small p-value is often associated with the possibility of publishing a result. Therefore, there has been increasing attention paid to its value. Although a small p-value can be seen as a measure of evidence against the null hypothesis, a small p-value can also be observed in a poorly designed study or when the test assumptions are violated, hence the p-value alone is not sufficient to guarantee good reproducibility of the findings (Ioannidis, 2005).

Additionally, it is important to remember that the *p-value* of a given test is computed from the observed data assuming that H_0 is true (i.e., it is related to $(Data|H_0)$), regardless of the form of the alternative hypothesis, H_1 , hence it does not provide support

for the alternative. Finally, $P(\text{Data}|H_0) \neq P(H_0|\text{Data})$, where $P(H_0|\text{Data})$ denotes the (posterior) probability that H_0 is true. Therefore, the p-value is not the probability that the null hypothesis is true. To better understand these points (Greenland *et al.*, 2016) provided a guide to properly reporting and interpreting the p-values.

Multiple Hypothesis Testing

To fix notations and introduce the general concepts, suppose that one is interested in performing a series of M simultaneous tests where the null hypotheses, H_{0i} , are compared with H_{1i} , as follows

$$H_{0i} : \theta_i \in \Theta_{0i} \text{ vs } H_{1i} : \theta_i \in \Theta_{1i} \quad i = 1, \dots, M.$$

Such circumstances are typical of many omics studies (Sham and Purcell, 2014), where the number of comparison, M , can be as large as thousands, hundreds of thousands or even more. In these cases, the use of p-values and/or individual decisional rules can result in a series of uncontrolled errors (Krzywinski and Altman, 2014c; Noble, 2009). In fact, suppose that one decides to apply an individual testing procedure at the same significance level, α , to each of the M tests, then the probability of “not making any Type I error in M tests” is $(1 - \alpha)^M$, (≈ 0 when M increases) and hence the probability of “taking at least a wrong rejection in M ” tests is $1 - (1 - \alpha)^M$, (≈ 1 when M increases). Not only that, by applying an individual testing procedure, it is expected to have on average $M\alpha$ false discoveries in M tests.

More precisely, as depicted in Table 2, we can assume that out of M tested null hypotheses, H_{0i} , M_0 are true (hence $M - M_0$ are false). M is known, while M_0 is an unknown parameter. In the decisional process we can reject R null hypotheses, (hence, $M - R$ hypotheses will be accepted). Out of the R discoveries, let S denote the number of true discoveries and V the number of false positives. Out of the $M - R$ accepted null hypotheses, let U denote the number of correctly accepted null hypotheses and T the number of false negatives. R is an observable random variable; V , S , U and T are not observable variables.

Several global measures of errors can be defined (Dudoit *et al.*, 2004a,b; Dudoit and van der Laan, 2008).

For the sake of brevity we report here only FWER (family wise error rate) and FDR (false discovery rate), that are defined as follows:

$$\text{FWER} = P(V \geq 1) = P(\text{at least one false positive discovery})$$

$$\text{FDR} = E[V/R] = E[V/(V + S)], \text{ where } E[\cdot] \text{ denotes the expected value of the random variable.}$$

Out of these error types, FDR (originally proposed in Benjamini and Hochberg (1995)) has had a significant impact in genomics since it better compromises between false positives and false negatives when the number of hypotheses to be tested is very large. For each of these global error types, there are several controlling procedures for guaranteeing the overall error is kept below some pre-specified significance level, α .

Bonferroni Correction

In order to guarantee that $\text{FWER} \leq \alpha$, one can test each hypothesis (H_{0i} versus H_{1i}) at the individual significance level, $\alpha' = \alpha/M$. Equivalently, one can define the so-called *adjusted p-values* defined as

$$p_i^{\text{adj}} = \min(1, Mp_i)$$

where p_i is the p-value of the i -th test, and then reject only those hypotheses, H_{0i} , such that $p_i^{\text{adj}} \leq \alpha$.

Benjamini and Hochberg Correction

In order to guarantee that $\text{FDR} \leq \alpha$, one can use the BH step-up procedure, which works as follows:

1. Order the individual p-values, $p_{(i)}$, from smallest to largest
 $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(M)}$, where $p_{(i)}$ denotes the i -th ordered p-value.
2. Find the largest k such that $p_{(k)} \leq \frac{k}{M}\alpha$
3. If $k > 0$, reject the null hypotheses $H_{0(1)}, \dots, H_{0(k)}$, otherwise accept all null hypotheses.

Table 2 Errors in multiple hypothesis testing

	# true Null hypotheses	# true Alternative hypotheses	# total hypotheses
# Accepted null hypotheses	U	T	M-R
# Rejected null hypotheses	V	S	R
Total	M_0	$M - M_0$	M

Source: Reproduced from Dudoit, S., van der Laan, M.J., Pollard, K.S., 2004a. Multiple testing. Part I. Single-step procedures for control of general Type I error rates. *Statistical Applications in Genetics and Molecular Biology* 3 (1), 1040; Dudoit, S., van der Laan, M.J., Pollard, K.S., 2004b. Multiple testing. Part II. Step-down procedures for control of the family-wise error rate. *Statistical Applications in Genetics and Molecular Biology* 3 (1), 1041; Dudoit, S., van der Laan, M.J., 2008. *Multiple Testing Procedures with Applications to Genomics*. Springer Series in Statistics.

The corresponding *adjusted p-values* are defined as $p_i^{adj} = \min\left(\frac{M \cdot p_{(i)}}{i}, 1\right)$.

The above-mentioned BH procedure requires that the M tests are independent, variants of such a correction procedure are available under dependency (Benjamini and Yekutieli, 2001). Moreover, it can be proved that the BH controlling procedure assures that $FDR \leq (M_0 / M) * \alpha \leq \alpha$. If $M_0 \ll M$ (as in the case of microarray or RNASeq data analysis), the procedure could be suboptimal, and an adaptive procedure can be used in order to improve the overall power (Benjamini *et al.*, 2006).

Other Procedures and Error Types

After the seminal paper concerning FDR (Benjamini and Hochberg, 1995), several related concepts and controlling procedures have been proposed, such as pFDR (Storey, 2002, 2003), and fdr (Efron, 2008), among many others. As for individual testing procedures, many of these controlling procedures are implemented in available statistical software (Dudoit and van der Laan, 2008; Bretz *et al.*, 2011) and are of common used when analyzing omic data. However, although the general idea behind such methods is similar, there could be important differences either in the choice of the global decisional errors and in the type of procedures used to keep such error under control (see, Benjamini, 2000, and discussion therein).

Future Directions

Large-scale hypothesis testing is a typical problem faced when analyzing biomedical data. In particular, the analysis of omic data is challenged by the fact that hundreds of thousands or even millions of hypotheses have to be compared, and only a limited number of samples can be collected (often referred to as a large p small n situation). Moreover, the data (e.g., gene expression) are usually highly correlated and parametric assumptions may not be satisfied. The problem is particularly relevant when hypothesis testing becomes part of more comprehensive inferential procedures, such as regression, model selection, and dimension reduction. To face such challenges, high dimensional testing procedures in complex experimental designs still have to be improved.

Closing Remarks

Hypothesis testing constitutes one of the most studied topics in modern statistical inference, with several applications in the analysis of biomedical data, since it allows one to study the effect of a certain drug in a group of individuals, to monitor gene expression under different conditions, to detect genomic variants associated with a certain pathology, and so on. The key concept is to support researchers in choosing (on the basis of observed data) between two possibilities, and to quantify the uncertainty in this decisional process by means of the two types of errors. Results are often reported in terms of p -values, and adjustment procedures have to be applied when multiple hypotheses are compared (Noble, 2009). Recently the American Statistical association has produced a statement on the way in which p -values should be reported and interpreted (Wasserstein and Lazar, 2016, and reference therein) in order to avoid misinterpretation and incorrect results (Ioannidis, 2005).

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Abramovich, F., Ritov, Y., 2013. Statistical Theory: A Concise Introduction. Chapman & Hall/CRC.
- Altman, N., Krzywinski, M., 2017a. P values and the search for significance. *Nature Methods* 14 (1), 3–4.
- Altman, N., Krzywinski, M., 2017b. Interpreting P values. *Nature Methods* 14, 213–214.
- Benjamini, Y., 2000. Discovering the false discovery rate. *Journal of the Royal Statistical Society, Ser. B* 72 (4), 405–416.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the fast discovery rate: A practical and powerful approach to multiple testing. *Journal of Royal Statistical Society, Ser B* 57, 289–300.
- Benjamini, Y., Krieger, A.M., Yekutieli, D., 2006. Adaptive linear step-up procedures that control the false discovery rate. *Biometrika* 93 (3), 491–507.
- Benjamini, Y., Yekutieli, D., 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics* 29 (4), 1165–1188.
- Berger, J.O., 1985. Statistical Decision Theory and Bayesian Analysis. Berlin: Springer-Verlag.
- Bretz, F., Westfall, P.H., Hothorn, T., 2011. Multiple Comparisons Using R. CRC Press.
- Casella, G., Berger, R., 2001. Statistical Inference, second ed. Duxbury.
- Dudoit, S., van der Laan, M.J., Pollard, K.S., 2004a. Multiple testing. Part I. Single-step procedures for control of general Type I error rates. *Statistical Applications in Genetics and Molecular Biology* 3 (1), 1040.
- Dudoit, S., van der Laan, M.J., Pollard, K.S., 2004b. Multiple testing. Part II. Step-down procedures for control of the family-wise error rate. *Statistical Applications in Genetics and Molecular Biology* 3 (1), 1041.
- Dudoit, S., van der Laan, M.J., 2008. Multiple Testing Procedures With Applications to Genomics. Springer Series in Statistics.
- Efron, B., 2007. Size, power and false discovery rates. *Annals of Statistics* 35 (4), 1351–1377.
- Efron, B., 2008. Microarrays, empirical Bayes and the two groups model. *Statistical Science* 23, 1–22.
- Greenland, S., Sen, S.J., Rothman, K.J., *et al.*, 2016. Statistical tests, P values, confidence intervals, and power: A guide to misinterpretations. *European Journal of Epidemiology* 31, 337–350.

- Halsey, L.G., Curran-Everett, D., Vowler, S.L., Drummond, G.B., 2015. The fickle P value generates irreproducible results. *Nature Methods* 12 (3), 179–185.
- Huber, W., 2016. A clash of cultures in discussions of the P value. *Nature Methods* 13, 607.
- Ioannidis, J.P., 2005. Why most published research findings are false. *PLOS Medicine* 2, e124.
- Krzywinski, M., Altman, N., 2013a. Importance of being uncertain. *Nature Methods* 10 (9), 809–810.
- Krzywinski, M., Altman, N., 2013b. Significance, P values and *t*-tests. *Nature Methods* 10 (11), 1041–1042.
- Krzywinski, M., Altman, N., 2013c. Power and sample size. *Nature Methods* 10 (12), 1139–1140.
- Krzywinski, M., Altman, N., 2014a. Designing comparative experiments. *Nature Methods* 11 (6), 597–598. doi:10.1038/nmeth.2974.
- Krzywinski, M., Altman, N., 2014b. Comparing samples- part I. *Nature Methods* 11 (3), 215–216.
- Krzywinski, M., Altman, N., 2014c. Comparing samples- part II. *Nature Methods* 11 (4), 355–356.
- Krzywinski, M., Altman, N., 2014d. Nonparametric tests. *Nature Methods* 11 (5), 467–468.
- Lazzeroni, L.C., Lu, Y., Belitskaya-Lévy, I., 2016. Solutions for quantifying P-value uncertainty and replication power. *Nature Methods* 13, 107–108.
- Lehmann, E.L., 1986. *Testing Statistical Hypotheses*, second ed. New York: Wiley.
- Noble, W.S., 2009. How does multiple testing correction work? *Nature Biotechnology* 27, 1135–1137.
- Nuzzo, R.L., 2015. The inverse fallacy and interpreting P values. *PM R* 7, 311–314.
- Nuzzo, R.L., 2016. Statistical power. *PM&R* 8, 907–912.
- Sham, P.C., Purcell, S.M., 2014. Statistical power and significance testing in large-scale genetic studies. *Nature Reviews Genetics* 15, 335–346.
- Storey, J.D., 2002. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Ser B* 64 (3), 479–498.
- Storey, J.D., 2003. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics* 31 (6), 2013–2035.
- Stuart, A., Ord, K., Arnold, S., 2008. *Kendall's advanced theory of statistics: Volume 2A – Classical inference & the linear model*, sixth ed. Wiley.
- van Helden, J., 2016. Confidence intervals are no salvation from the alleged fickleness of the P value. *Nature Methods* 13, 605–606.
- Wasserstein, R.L., Lazar, N.A., 2016. The ASA's statement on p-Values: Context, process, and purpose. *The American Statistician* 70 (2), 129–133.

Statistical Inference Techniques

Daniela De Canditiis, Istituto per le Applicazioni del Calcolo “M. Picone”, Rome, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A common bioinformatics problem is to determine, in terms of probability, whether an observed sample is drawn from a specific population (i.e., distribution). Another common situation is to decide whether differences between two or more groups of samples are due to the differences in the populations or matter of chance. All such kind of problems can be stated in terms of a statistical hypothesis test as described in many standard statistical book (Abramovich and Ritov, 2013; Casella and Berger, 2001; Ross, 2009) among many others.

In this article, we present the most commonly used statistical inference techniques, trying to organize them in a simple way for quick consultation. The plethora of hypothesis tests available in the modern statistics can be divided essentially into two categories: parametric and nonparametric. The first category includes those tests based on the assumption of knowing the distribution of the underlying population(s) up to some unknown parameters; the second category contains those tests that are “distribution-free”. When a parametric test is used, then it must be stated that “If the assumptions regarding the shape of the population(s) are valid, then we conclude...”, on the other hand if the same conclusions are drawn using a nonparametric test it is enough to state that “regardless of the shape of the population we may conclude that...”. At a first glance, it seems that nonparametric tests are preferable to parametric ones, however this is not always the case, because when the assumptions are met, then a parametric test is always more powerful than a nonparametric one. Hence, as a rule of thumb, it is a good practice to use a parametric test instead of a nonparametric one when assumptions are satisfied. Moreover, the validity of the assumptions can be tested.

With this distinction in mind, this article is divided in two sections: parametric tests and nonparametric tests. Inside each section, the techniques are assigned to subsections according to the experimental design for which they are appropriate. Specifically, each section is divided into three subsections: the first containing techniques employed when one statistical sample is available, the second containing techniques employed when two statistical samples are available, the last one containing techniques employed when more than two statistical samples are available. Fig. 1 summarizes these different experimental designs.

We use classical notations as described in Casella and Berger (2001) and adopt the p-value approach. Hence, for each test, we present the model assumptions and its test statistics, being the evaluation of the p-value demanded to common statistical software. All the tests discussed in this article along with their respective built in functions in Matlab and R are summarized in Table 1.

Parametric Tests

A parametric test is a hypothesis testing procedure based on the assumption that observed data are distributed according to some distributions of well-known form (e.g., normal, Bernoulli, and so on) up to some unknown parameter(s) on which we want to make inference (say the mean, or the success probability). Examples of this type of test are the T-test for comparing the mean(s), the Binomial test for comparing the proportions, the F-test for the ANOVA models. It is important to understand that the validity of a parametric test depends on the assumptions one can elicit on the underlying population(s). If the assumptions are satisfied, then the decision rule will control the Type I error at the chosen significance level. Otherwise, results might be over-optimistic and the decision might not be under a proper error control.

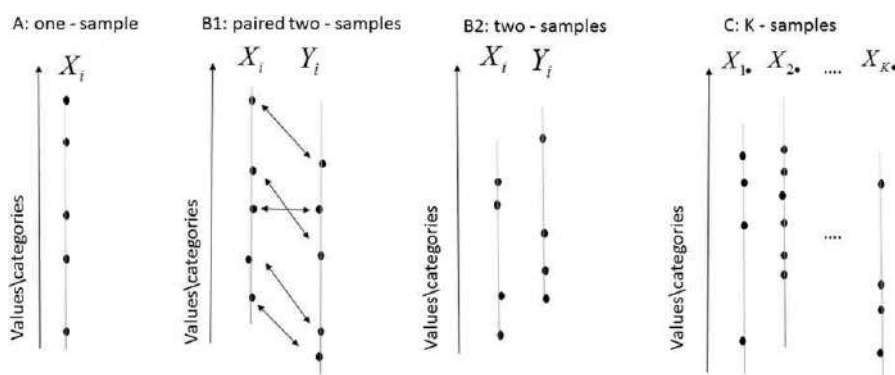


Fig. 1 Sketch of experimental data in one sample design (A); two samples design (B1 and B2); K-samples design (C).

Table 1 Most common hypothesis testing procedures and respective built in Matlab and R functions

Test	decription	Name of Matlab function	Name of R function
T-test	Tests if a gaussian sample with unknown variance has a specified mean	ttest	t.test
Binomial test	Tests if a Bernoulli sample has a specified success probability	cdf('bino',...); ztest	binom.test
Two sample T-test	Tests if two gaussian samples have the same mean	ttest2	t.test
Fisher-Irwin	Tests if two Benoulli samples have the same success probability	cdf('hyge',...); ztest	prop.test
One-way ANOVA	Tests if multiple gaussian samples have the same mean	anova1	oneway.test
Sign test	Tests if an ordinal sample has a specified median	signtest	binom.test
Signed Rank test	Tests if a continuos sample is symmetric around a specified median	signrank	wilcox.test(one-sample)
One-sample chi-square	Tests if a sample fits a specified discrete probabilistic model	chi2gof	chisq.test
One-sample Kolmogorov-Smirnov	Tests if a sample fits a specified continuos probabilistic model	kstest	ks.test
Shapiro-Wilk	Tests if a sample fits a gaussian distribution	swtest	shapiro.test
Mann-Whitney	Tests if two samples have the same median	ranksum	wilcox.test(two-sample)
Chi-square test of independence	Tests if two characteristics are independent	cdf('chi2',...)	fisher.test
Two-sample Kolmogorov-Smirnov	Tests if two samples come from the same continuous distribution	kstest2	ks.tests
Kruskal-Wallis	Tests if multiple samples have the same median	kruskalwallis	kruskal.test

Statistical Tests for the “One-Sample” Case

Assume we have collected a random sample $X_1, \dots, X_n$ from a population of interest, where X_i are independent and identically distributed random variables (i.e., $X_i \sim f_\theta(x)$, with f denoting the probability distribution function). See [Fig. 1\(A\)](#).

One sample T-test

This test allows making inference on the mean of a given normal population.

Under the hypothesis that $f_\theta(x)$ follows a Gaussian distribution, with unknown parameters $\vartheta = (\mu, \sigma^2)$, the one-sample T-test is used to determine whether our sample has a given mean $\mu = \mu_0$ (i.e., $H_0: \mu = \mu_0$ vs $H_1: \mu \neq \mu_0$) or if our sample has a mean not smaller than a given value (i.e., $H_0: \mu \geq \mu_0$ vs $H_1: \mu < \mu_0$) or if our sample has a mean not larger than a given value (i.e., $H_0: \mu \leq \mu_0$ vs $H_1: \mu > \mu_0$). In all cases, the test statistics results to be:

$$T(X_1, \dots, X_n) = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}, \text{ with}$$

$$\bar{X} = \sum_{i=1}^n X_i/n \quad \text{and} \quad S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

denoting the sample mean and sample variance, respectively.

Under the null hypothesis, the test statistics is distributed like a T-Student variable with $n - 1$ degree of freedom. Hence, the test is called T-test.

The one-sample T-test appears in many bioinformatics contexts, such as measuring protein expression, the quantity of drug delivered by a medication and so on; since its central role in inference all statistical software have a built in T-test function, see [Krzywinski and Altman \(2013\)](#) for more discussion. [Table 2](#) summarizes the p-values for each of the three hypothesis tests, with T_{n-1} denoting a Student variable and t_{obs} the observed value of the test statistics.

Finally, let us note that when the sample size is sufficiently large (usually $n \geq 30$), then this test can be applied without the normality assumption by the mean of the Central Limit Theorem.

Table 2 p -values for the one-sample T -test; the test statistics being a Student variable.

H_0	H_1	p -Value
$\mu = \mu_0$	$\mu \neq \mu_0$	$2P(T_{n-1} > t_{obs})$
$\mu \geq \mu_0$	$\mu < \mu_0$	$P(T_{n-1} < t_{obs})$
$\mu \leq \mu_0$	$\mu > \mu_0$	$P(T_{n-1} > t_{obs})$

Table 3 p -values for the binomial test; the test statistics being a Binomial variable.

H_0	H_1	p -value
$p = p_0$	$p \neq p_0$	$2 \min\{P(T \geq t_{obs}), P(T \leq t_{obs})\}$
$p \geq p_0$	$p < p_0$	$P(T < t_{obs})$
$p \leq p_0$	$p > p_0$	$P(T > t_{obs})$

Binomial test

This test allows making inference on the success probability of a given Bernoulli population. Under the hypothesis that the underling population is Bernoulli distributed, with unknown $p = P(X=1)$, then it is possible to test whether our sample has a given probability of success p_0 (i.e., $H_0: p = p_0$ vs $H_1: p \neq p_0$) or if our sample has a probability of success not smaller than a given value (i.e., $H_0: p \geq p_0$ vs $H_1: p < p_0$) or if our sample has a probability of success not larger than a given value (i.e. $H_0: p \leq p_0$ vs $H_1: p > p_0$). In all these cases, the test statistics results to be:

$$T(X_1, \dots, X_n) = \sum_{i=1}^n X_i.$$

Under the null hypothesis, the test statistics T is distributed like a Binomial random variable with number of trials n (the sample size) and probability of success p_0 , denoted $B(n, p_0)$. This test is known as the Binomial test and the p -value is evaluated according to **Table 3** with T distributed as a $B(n, p_0)$ variable and t_{obs} the observed value of the test statistics.

Finally, we observe that, when $np_0(1-p_0) > 20$, the Binomial distribution is well approximated by a Gaussian one, then evaluation of the p -value simplifies and the test is known as approximated Binomial test.

Statistical Tests for the “Two-Sample” Case

The two-sample statistical tests are used when the researcher wishes to establish whether two treatments are different, or whether one treatment is “better” than another. The treatments can be any diverse variety of conditions and the two samples come from two groups that underwent to two different conditions. Two different experimental designs can be used for this aim: the paired experimental design, where the two samples have the same number of data and each member of the first sample is connected to a specific member of the second sample serving as its own control, and the independent experimental design, where the members of the two groups are independent and the sample sizes might be different. Sketch of these experimental designs are shown in **Fig. 1 (B1)** and **(B2)**, respectively.

Paired two-samples T -test

This test allows making inference on the means of two normal populations when the sampling design is paired.

Under the assumption that the pairs $\{(X_i, Y_i)\}_{i=1, \dots, n}$ are sampled from two Gaussian distributions with unknown mean, μ_1 and μ_2 , and unknown variance (not necessarily the same), it is easy to show that the problem reduces to the one-sample T -test applied to the data $\{W_i = (X_i - Y_i)\}_{i=1, \dots, n}$ for testing $H_0: \mu_1 - \mu_2 = 0$ vs $H_1: \mu_1 - \mu_2 \neq 0$; or for testing $H_0: \mu_1 - \mu_2 \geq 0$ vs $H_1: \mu_1 - \mu_2 < 0$; or $H_0: \mu_1 - \mu_2 \leq 0$ vs $H_1: \mu_1 - \mu_2 > 0$. The test statistics is $T(W_1 \dots W_n) = \sqrt{n} \bar{W} / S_W \sim T_{n-1}$, with $\bar{W}$ and S_W the sample mean and the sample variance of $\{W_i\}_{i=1, \dots, n}$. Under H_0 , the test statistics is distributed like a T -student with $n-1$ degree of freedom and the p -value is evaluated according to **Table 2**.

Independent two-samples T -test

This test allows making inference on the means of two normal populations when the sampling design is independent.

Under the assumption that the data $X_1, \dots, X_n$ and $Y_1, \dots, Y_m$ are sampled from two independent Gaussian distributions with unknown mean, μ_1 and μ_2 , and unknown equal variance σ^2 , it is possible to test $H_0: \mu_1 - \mu_2 = 0$ vs $H_1: \mu_1 - \mu_2 \neq 0$, or $H_0: \mu_1 - \mu_2 \geq 0$ vs $H_1: \mu_1 - \mu_2 < 0$ or $H_0: \mu_1 - \mu_2 \leq 0$ vs $H_1: \mu_1 - \mu_2 > 0$. The test statistics results to be:

Table 4 p -values for the independent two-sample T -test; the test statistics being a Student variable.

H_0	H_1	p -value
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$2P(T_{n+m-2} > t_{obs})$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$	$P(T_{n+m-2} < t_{obs})$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$	$P(T_{n+m-2} > t_{obs})$

	Group 1	Group 2
Number of success	$\sum_{i=1}^n X_i = n_1$	$\sum_{i=1}^m Y_i = m_1$
Number of failures	$n - n_1$	$m - m_1$
Sample size	n	m

Fig. 2 Contingency table for the Fisher-Irwin test.

$$T(X_1, \dots, X_n, Y_1, \dots, Y_m) = \frac{\bar{X} - \bar{Y}}{S_p \sqrt{1/n + 1/m}}, \text{ where}$$

$$\bar{X} = \sum_{i=1}^n X_i/n, S_X^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2; \quad \bar{Y} = \sum_{i=1}^m Y_i/m, S_Y^2 = \frac{1}{m-1} \sum_{i=1}^m (Y_i - \bar{Y})^2$$

and the pooled variance is defined by $S_p = ((n-1)S_X^2 + (m-1)S_Y^2)/(n+m-2)$.

Under the null hypothesis, the test statistics has a Student distribution with $n+m-2$ degree of freedom and [Table 4](#) reports the p -value for each situation.

See [Ross \(2009\)](#), [Krzywinski and Altman \(2014b,c\)](#) for more details and discussion. Further, we stress that when the hypothesis of equal variance is not met, then the Welch's modification can be applied ([Lehmann 1999](#)). The test statistics is given by:

$$T(X_1, \dots, X_n, Y_1, \dots, Y_m) = \frac{\bar{X} - \bar{Y}}{\sqrt{S_X^2/n + S_Y^2/m}}$$

and it results to be again a Student variable T_ν with degree of freedom given by the following formula, $(\nu) = (S_X^2/n + S_Y^2/m^2)/(S_X^4/n^2(n-1) + S_Y^4/m^2(m-1))$. In all the common statistical software, the two-independent t-test is built in with the options equal or unequal variance.

Finally, let us note that when the samples' size are sufficiently large (usually $n, m \geq 30$), then both paired and independent two-sample T-test can be applied without the normality assumption by the mean of the Central Limit Theorem.

Fisher-Irwin test

This test allows making inference on the success probability parameters of two Bernoulli populations.

Suppose a research has scores from two independent samples all falling into one or the other of two mutually exclusive classes. In other words, suppose the researcher has two groups, experimental and control, or female and male, or with disease and not-disease, and for each group the experimenter measures a dichotomous variable. Then, the two groups of scores $X_1, \dots, X_n$ and $Y_1, \dots, Y_m$ can be assumed to be sampled from two independent Bernoulli distributions with unknown parameter p_1 and p_2 . [Fig. 2](#) outlines the type of data in this specific case. The Fisher-Irwin test serves to test the hypothesis $H_0: p_1 = p_2$ vs $H_1: p_1 \neq p_2$, and it is based on a statistics whose distribution under H_0 is exactly known to be hypergeometric. Sometimes this test is also presented as a nonparametric test because it can be adapted to several other situations when data are discrete (nominal or ordinal), however according to our definition the test assumes the data distribution known except for two parameters about which the hypothesis is formulated. See [Ross \(2009\)](#) for more details. Finally, we note that when the size of the two samples are large, then it is possible to approximate the Binomial variables $\sum_{i=1}^n X_i$ and $\sum_{i=1}^m Y_i$ with Gaussian variables and evaluation of p -value simplifies. See [Ross \(2009\)](#) and [Siegel \(1956\)](#) for more details and one-tail situations.

Statistical Tests for k Samples

The k-samples statistical test is used when the researcher wishes to establish if k groups, that underwent k different conditions, are different. See Fig. 1(C) for an experimental design sketch. The case $k=2$ falls into the two-sample T-test, then here we suppose $k>2$. The ANOVA model furnishes a well known parametric test when the k samples are supposed to come from Gaussian distributions with unknown means $\mu_1, \mu_2, \dots, \mu_k$ and unknown (necessarily) common variance σ^2 .

One-way ANOVA

This test allows making inference on the means of more than two normal populations.

Assume we have collected a sample $X_{i1}, \dots, X_{ini}$ from $N(\mu_i, \sigma^2)$ for each condition $i=1, \dots, k$ and assume we want to test $H_0 : \mu_1 = \mu_2 = \dots = \mu_k$ (i.e., the mean in all groups is the same) versus the alternative hypothesis that H_0 is not true (at least one group's mean is different). If the k samples have the same size, $n_1 = n_2 = \dots = n_k$ the model is called balanced; otherwise the model is called unbalanced. In both cases, the test statistics is the following:

$$T(\dots X_{ij} \dots) = \frac{SS_b/k-1}{SS_w/N-k}$$

with $N = \sum_{i=1}^k n_i$ and where SS_b denotes the so called *Between groups Sum of Squares* and SS_w the *Within group Sum of Squares* defined as:

$$SS_b = \sum_{i=1}^k n_i (X_{i*} - X_{**})^2 \quad \text{and} \quad SS_w = \sum_{i=1}^k \sum_{j=1}^{n_i} (X_{ij} - X_{i*})^2$$

In this notation, the symbol * stands for the empirical mean with respect to the variable *, (for example $X_{3*} = \frac{1}{n_3} \sum_{j=1}^{n_3} X_{3j}$). The test statistics makes a comparison between two different estimators of σ^2 , $SS_w/N - k$ which is always a good estimator of σ^2 and $SS_b/k - 1$ which is a good estimator of σ^2 when H_0 is true, while it tends to overestimate σ^2 when H_0 is not true. This observation implies that the larger the observed statistic the more likely the observed scores did not come from the population described by the null hypothesis. Under the null hypothesis, the test statistics is distributed like a Fisher variable with $k - 1$ degree of freedom at the numerator and $N - k$ degree of freedom at the denominator, the p-value results to be $P(F_{k-1, N-k} > t_{obs})$, where $F_{k-1, N-k}$ is a Fisher variable and t_{obs} is the observed value of the test statistics.

If there is a significant difference between the groups, further tests (known as *post hoc tests*) are performed to verify which groups are really different from each other, see Klockars and Hancock (2000) for more details.

When possible the balanced scenario is always preferable to the unbalanced scenario since it is more robust with respect to departure from the hypothesis.

Finally, we observe that the one-way ANOVA model allows you to study the effect of a single factor on data distribution, i.e. it determines whether the group's average is influenced by belonging to the group. It is also possible to study the effect of two factors simultaneously using the two-way ANOVA model that lets you determine whether the average of the groups is affected by either of the two factors. This last model can be used with or without interactions and for a more complete discussion we demand to Krzywinski and Altman (2014b,c), Ross (2009), and Casella and Berger (2001).

Nonparametric Tests

Nonparametric tests are hypothesis testing procedures where it is not necessary (or not possible) to specify the parametric form of the underlying population(s) distribution(s). Therefore, they require few assumptions on the population under study and can be used to test much wider ranges of possibilities. Moreover, many nonparametric tests adapt to the case of ordinal and categorical random variables, for the first type of data being possible to evaluate the rank, for the second type of data being possible to evaluate only frequencies of occurrence.

Examples of this type of such tests include the sign test, Mann-Whitney-Wilcoxon test, Chi-square test, Kruskal-Wallis test, Kolmogorov-Smirnov test, among many others.

Statistical Tests for the “One-Sample” Case

Sign test

This test allows making inference on the median of a population.

Suppose we have a random sample $X_1, \dots, X_n$ extracted from an unknown population with an unknown median m . The sign test is used to test the hypothesis $H_0: m = m_0$ vs $H_1: m \neq m_0$. The data are supposed to be at least of ordinal type so that for each X_i it is possible to evaluate the following sign function:

$$I_i = \begin{cases} 1 & \text{if } X_i < m_0 \\ 0 & \text{if } X_i \geq m_0 \end{cases}$$

Under the null hypothesis, the new random variables $I_1, \dots, I_n$ are independent Bernoulli variables with probability of success $1/2$. It is then possible to apply the Binomial test presented in Section Statistical Tests For The "One-Sample" Case for evaluating the p-value of the test statistics $T(X_1, \dots, X_n) = \sum_{i=1}^n I_i$, which is called sign sum statistics. See Siegel (1956), Ross (2009), Krzywinski and Altman (2014a).

Signed rank test

This test allows making inference on the symmetry of a distribution around its median. Suppose we have a random sample $X_1, \dots, X_n$ extracted from an unknown continuous population with unknown median m . The signed rank test is used to test the hypothesis $H_0 : P(X < m_0 - a) = P(X \geq m_0 + a) \forall a > 0$ vs H_1 : the population is not symmetric around its median. While, the sign test serves to test the hypothesis that the population is centered around m_0 , using only the information if the data is smaller or bigger than m_0 ; the signed rank test serves to test the hypothesis that the population is centered and symmetric around m_0 , using the information how much the data is smaller or bigger than m_0 .

Define $Y_i = X_i - m_0$, for $i = 1, \dots, n$, and rank all the Y_i 's without regard to sign: give rank 1 to the smallest $|Y_i|$, give rank 2 to the next smallest $|Y_i|$, etc. Like in the sign test define for each $i = 1, \dots, n$ the sign function $I_i = \begin{cases} 1 & \text{if } X_i < m_0 \\ 0 & \text{if } X_i \geq m_0 \end{cases}$ and sum the rank having a sign equals 1, i.e. sum the rank of the observations $X_i < m_0$, the test statistics results to be:

$T(X_1, \dots, X_n) = \sum_{i=1}^n I_i \text{rank}(|Y_i|)$, the sum of rank statistics. Evaluation of the p-value is demanded to standard statistical software. See Ross (2009) for more details.

One-sample chi-square test

This test is as goodness of fit test, i.e. it is used for verifying whether the data fit a given discrete probabilistic model.

Suppose the researcher is interested in the number of subjects, objects or responses, which fall in various categories. For example, a group of patients can be classified according to their response to a certain drug: "good", "indifferent", "bad", and the researcher is interested in testing the hypothesis that the frequencies of these three categories of response is equal to some putative frequencies. The number of categories may be $k \geq 2$, however the case $k=2$ results in the above-mentioned Binomial test. This test is a non-parametric test, since data are supposed to be sampled from an unknown discrete population with unknown expected frequencies for the k categories and one wants to test $H_0 : (p_1, \dots, p_k) = (p_1^0, \dots, p_k^0)$ vs $H_1 : (p_1, \dots, p_k) \neq (p_1^0, \dots, p_k^0)$.

Let us denote O_i the observed number of data categorized into i -th category, i.e., $O_i = \sum_{j=1}^n \text{Ind}(X_j = i)$ with $\text{Ind}(A) = 1$, if A is true; 0 otherwise, and denote $E_i = np_i^0$ the expected number of data into i -th category under H_0 ; then the test statistics is defined by the following expression

$$\chi^2(X_1, \dots, X_n) = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

Under the null hypothesis is possible to show that the test statistics is distributed like a χ_{k-1}^2 with $k-1$ degree of freedom.

The p-value results to be $P(\chi_{k-1}^2 > t_{\text{obs}})$; roughly speaking, the larger the observed statistic the more likely the observed frequencies did not come from the population described by the null hypothesis. This test is valid when the expected number of data falling into category i , E_i , satisfy the hypothesis that at least 80% of them are such that $E_i \geq 5$ and at least 20% of them are such that $E_i \geq 1$. See Siegel (1956), Ross (2009) for more details and examples.

One-sample Kolmogorov-Smirnov test

This test is as goodness of fit test, i.e. it is used for verifying whether the data fit a given continuous probabilistic model.

Suppose we have a sample $X_1, \dots, X_n$ of scalar values and suppose we want to test the hypothesis that this sample is drawn from a putative distribution $F_0(x)$. Then, if $F_X(x)$ is the cumulative distribution function of the sampled random variable, we want to test $H_0: F_X = F_0$ vs $H_1: F_X \neq F_0$. The rationale behind this test consists in evaluating the empirical cumulative distribution and computing the greatest difference between observed and expected cumulative distribution functions. More specifically, if $F_e(x) = \frac{1}{n} \sum_{i=1}^n \text{Ind}(X_i \leq x)$ is the empirical cumulative distribution function (i.e., based on the observed values), the test statistic is the maximum deviation between them, $D = \max_{-\infty < x < +\infty} |F_e(x) - F_0(x)|$. The larger the observed statistic, d_{obs} , the more likely the observed data did not come from the putative population, so the p-value is $p\text{-value} = P(D \geq d_{\text{obs}})$. The p-value is usually evaluated by the mean of any statistical software and it is obtained by simulations since, under H_0 , the sampling distribution of D does not depend from the specified distribution $F_0(x)$, then a uniform distribution can be employed to evaluate the p-value.

It is worth to observe that it is always possible to apply the One-sample chi-square test described in the previous paragraph to this case, however the Kolmogorov-Smirnov one-sample test treating individual observation separately do not lose information through the binning of categories. Hence it is always preferable to apply Kolmogorov-Smirnov test instead of chi-square test when possible. See Siegel (1956), Ross (2009) for more details.

Finally, we observe that if a goodness of fit test has to be performed with F being normal, then some more powerful tests are available, like Shapiro-Wilk test or Lilliefors test, see Razali and Wah (2011) for discussions and comparisons.

Statistical Tests for the “Two-Samples” Case

Since a paired two sample design can be reduced to a one-sample design considering $\{W_i = (X_i - Y_i)\}_{i=1, \dots, n}$, in this section we only present those tests which are specific for the two independent samples situation, schematically represented in [Fig. 1\(B2\)](#)

Mann-Whitney test

This test allows making inference on the medians of two independent populations.

Assumes we have two sets of data $X_1, \dots, X_n$ and $Y_1, \dots, Y_m$ sampled from two unknown continuous populations with cumulative distribution functions F and G , then this test serves to test $H_0: F=G$ vs $H_1: F \neq G$. Alternatively, if we know that the two populations have the same distribution shape, then this test serves to test the hypothesis that the two distribution have the same median, see [Krzywinski and Altman \(2014a\)](#).

As a first step combine the data from the two samples and sort these $n + m$ values. The value of the U-statistics is given by the number of times that a data from the first population (X) precedes a data from the second population (Y) in the ranking. The sampling distribution of U under H_0 is known and any statistical software can determine the p-value, i.e. the probability associated with the occurrence under H_0 of any U as extreme as the observed value. This test is also known as the Wilcoxon test or the rank sum test and it is the most useful alternative to the two-sample (independent) T-test; see [Krzywinski and Altman \(2014a\)](#) for a detailed comparison between two sample T-test and Mann-Whitney test.

Chi-square test of independence

This test allows making inference on the independence/association of two traits in the paired two-sample experimental design of [Fig. 1\(B1\)](#), where each subject/data of the sample is categorized according to two type of categories/traits. Still, this test allows making inference on two categorical variables in the two independent samples experimental design of [Fig. 1\(B2\)](#), where each data of each of the two samples is categorized according to one type of category/trait.

More formally, suppose we have a sample in which each data can be classified according to two criteria or according to two characteristics, indicated by X and Y , respectively. Suppose that the characteristic X can assume r different values and that the characteristic Y can assume s different values. Denote p_{ij} the probability that a random element from the population takes the value i for the characteristic X and the value j for the characteristic Y .

Let n be the total number of measured subjects/data, evaluate the value of the two characteristics for each element of it, then it is possible to define a contingency table for better representing it. In the contingency table with r rows and s columns, in position ij will be the number of sample elements that simultaneously have the characteristic $X=i$ and the characteristic $Y=j$, denote it O_{ij} . On the margin of the table the observed absolute frequencies for each of the two characteristic will be obtained, denote it np_i for $i=1, \dots, r$ and nq_j for $j=1, \dots, s$. Then, it is possible to test the independency hypothesis $H_0: p_{ij} = p_i q_j \forall i=1, \dots, r \ j=1, \dots, s$ vs $H_1: p_{ij} \neq p_i q_j$ for same i and j . The test statistics results to be:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^s \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} is the observed number of cases categorized in i -th row and j -th column and $E_{ij} = np_i q_j$ is the expected number of cases under H_0 to be categorized in the i -th row and j -th column. Under the null hypothesis, the test statistics is distributed like a chi-square with $(r-1)(s-1)$ degree of freedom, then the p-value results to be $P(\chi_{(r-1)(s-1)}^2 \geq t_{obs})$. See [Siegel \(1956\)](#), [Ross \(2009\)](#) for more details.

Finally, let us note that in case $r=s=2$, the same test can be exactly performed using the Fisher-Irwin test which tests the null hypothesis that the probability of being in one or the other of two categories does not depend from belonging to one or the other of the two groups.

Two-sample Kolmogorov-Smirnov test

This test allows making inference on the distributions of two continuous distributions. It tests whether two independent samples have been drawn from the same population (or population with the same distribution). The two populations can have any kind of difference: in location (central tendency), in dispersion, in skewness, etc. Like the Kolmogorov-Smirnov one-sample test, this two-sample test is concerned with the agreement between two cumulative distributions. Specifically, if $F_e(x) = \frac{1}{n} \sum_{i=1}^n \text{Ind}(X_i \leq x)$ is the empirical cumulative distribution function of the first sample and $G_e(x) = \frac{1}{m} \sum_{i=1}^m \text{Ind}(Y_i \leq x)$ is the empirical cumulative distribution function of the second sample, the test statistic is the maximum deviation between them, $D = \max_{-\infty < x < +\infty} |F_e(x) - G_e(x)|$. Under the null hypothesis, the sampling distribution of the test statistics can be obtained, evaluation of the p-value being demanded to specific statistical toolbox. See [Siegel \(1956\)](#) for more details and examples.

Statistical Tests for K Samples

The chi-square test and the median test for k independent samples are the same as the ones for two independent samples, for that reason we demand to the literature for appropriate extension, see for example ([Siegel, 1956](#)). In this section, we present the Kruskal-Wallis one-way analysis of variance by rank, which is the nonparametric counterpart of the one-way ANOVA model presented in Section Statistical Tests for k Samples.

Kruskal-Wallis one-way ANOVA

This non-parametric test allows making inference on the median of more than two populations. Assume we have collected k independent samples each of size n_j , $j = 1, \dots, k$, the Kruskal-Wallis one-way ANOVA tests the hypothesis that the samples come from the same continuous population, or more generally it tests the hypothesis that the k populations have the same median under the assumptions they share the same shape distribution.

The rationale of the method consists of ranking the pooled data from all the k samples and replacing data by their rank. When this has been done, the sum of the ranks in each sample is found and the test determines whether these sums of ranks are so disparate that they are not likely to have come from samples drawn from the same population. More specifically, if $R_j = \sum_{i=1}^{n_j} \text{rank}(X_{ij})$ is the sum of ranks in j th sample and $n = \sum_{j=1}^k n_j$ is the total number of data available into the k samples, then the test statistics is defined by

$$H = \frac{12}{n(n+1)} \sum_{j=1}^k \frac{R_j^2}{n_j} - 3(n+1)$$

Under the null hypothesis, the distribution of the test statistics can be exactly evaluated (in small sample situation) or well approximated by a chi-square with $k - 1$ degree of freedom (in large sample situation); in any case evaluation of the p-value is demanded to specific statistical toolbox. See [Kruskal and Wallis \(1952\)](#), [Siegel \(1956\)](#) for more details and examples.

Finally, let us observe that it is possible to specialize this test to the case of repeated measures or k matched samples, i.e., when the number of observations in each of the k groups are the same. This is known as the Friedman test, see [Siegel \(1956\)](#) for details and examples.

Closing Remarks

This article presents some of the most well-known statistical tests available in literature. For easy of usage, all the tests presented have been summarized in [Table 1](#) along with classical built in functions of Matlab and R. However, the list is not exhaustive and many other functions and statistical tests are available. The most important message is that before applying an hypothesis testing procedure, it is always a good practice to revisit the mathematical assumptions required and look for its documentation, which can be found into the instruction manuals of the most common statistical software, see for example [Krijnen Wim \(2009\)](#), [Mathworks \(R2017a\) \(2017\)](#). Finally, it is important to stress that the above mentioned procedures can be applied for testing individual statistical hypotheses. In the case of multiple tests, the multiplicity correction should be properly taken into account. We demand [Abramovich and Ritov \(2013\)](#) for discussion and methods.

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Abramovich, F., Ritov, Y., 2013. Statistical Theory: A Concise Introduction. Chapman & Hall/CRC.
- Casella, G., Berger, R., 2001. Statistical Inference, 2nd ed. Duxbury.
- Klockars, A.J., Hancock, G.R., 2000. Scheffé's more powerful F-protected post hoc procedure. *Journal of Educational and Behavioral Statistics* 25 (1), 13–19.
- Krijnen, Wim P., 2009. Applied Statistics for Bioinformatics Using R.
- Kruskal and Wallis, 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* 47 (260), 583–621.
- Krzywinski, M., Altman, N., 2013. Significance, P values and t-tests. *Nature Methods* 10 (11), 1041–1042.
- Krzywinski, M., Altman, N., 2014a. Nonparametric tests. *Nature Methods* 11 (5), 467–469.
- Krzywinski, M., Altman, N., 2014b. Comparing samples – Part I. *Nature Methods* 11 (3), 215–216.
- Krzywinski, M., Altman, N., 2014c. Analysis of variance and blocking. *Nature Methods* 11 (7), 699–700.
- Lehmann, E.L., 1999. Elements of Large Sample Theory. New York: Springer.
- Mathworks documentation. Statistics and Machine Learning User's Guide. Available at: <https://it.mathworks.com/help/stats/>.
- Razali, N.M., Wah, Y.B., 2011. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics* 2 (1), 21–33.
- Ross, S.M., 2009. Probability and Statistics for Engineers and Scientists, 4th ed Elsevier.
- Siegel, S., 1956. Nonparametric Statistics for the Behavioral Sciences. Tokyo: McGraw-Hill, Kogakusha Company, Ltd.

Correlation Analysis

Monica Franzese and Antonella Iuliano, Institute for Applied Mathematics “Mauro Picone”, Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the analysis of health sciences disciplines, usually we are interested to understand the type of relationship that exists between two or more variables. For example, the association between blood pressure and age, height and weight, the concentration of an injected drug and heart rate or the intensity of a stimulus and reaction time. Generally, the strength of relationship between two continuous variables is examined by using a statistical technique called correlation analysis. The concept of correlation was introduced by Sir Francis Galton (1877) in the mid 19th century as the most important contribution to psychological and methodology theory. In 1896, Karl Pearson published his first rigorous treatment of correlation and regression in the *Philosophical Transactions of the Royal Society of London* (Pearson, 1930). Here, he developed the Pearson product-moment correlation coefficient (PPMCC), using an advanced statistical proof based on the Taylor expansion. Today, the term correlation is used in statistics to indicate an association, connection, or any form of relationship, link or correspondence between two or more random variables. In particular, the PPMCC is used to study the linear relationship between two sets of numerical data. The correlation is connected to the concept of covariance that is the natural measure of the joint variability of two random variables. Usually, when we use measures of correlation or covariance from a set of data, we are interested in the degree of the correlation between them. In fact, we cannot prove that one variable causes a change in another if there are no connections between the two variables analyzed. An example is to test if the efficacy of a specific treatment is connected with the dose for the drug in a patient. In this case, a change in the drug variable changes the treatment variable, then the two variables are correlated. Therefore, correlation analysis provides information about the strength and the direction (positive or negative) of a relationship between two continuous variables. No distinction between the explaining variable and the variable to be explained is necessary. On the other hand, regression analysis is used to model or estimate the linear relationship between a response variable and one or more predictor variables (Gaddis and Gaddis, 1990). The simplest regression models involve a single response variable (dependent variable) and a single predictor variable (independent variables). For example, the blood pressure measured at the wrist depends on the dose of some anti-hypertensive drug administered to the patient.

In this work, our discussion is limited to the exploration of the linear (Person correlation) and non-linear relationship (Spearman and Kendall correlations) between two quantitative variables. In particular, we first introduce the concepts of covariance and correlation coefficient, then we discuss how these measures are used in statistics for estimating the goodness-of-fit of linear and non-linear trends and for testing the relationship between two variables. Finally, we explain how the regression analysis is connected to the correlation one. Simulated and real data sets are also presented for the identification and characterization of relationship between two or more quantitative variables.

Measures of Correlation Analysis

In this section, we discuss three correlation measures between two quantitative variables X and Y to determine the degree of association or correlation between them. For example, if we are interested to investigate the association between body mass index and systolic blood pressure, the first step for studying the relationship between these two quantitative variables is to plot a scatter diagram of the data. The points are plotted by assigning values of the independent variable X (body mass index) to the horizontal axis and values of the dependent variable Y (systolic blood pressure) to the vertical axis. The pattern made by the points plotted on the scatter plot usually suggests the nature and strength of the relationship between two variables. For instance, in Fig. 1, the first plot (A) shows that there is no relationship between the two variables X and Y , the second one (B) displays that exist a positive linear relationship between the two variables X and Y , the third one (C) exhibits a negative linear trend between the two variables X and Y . In the last two cases (B and C) the strength of linear relationship is the same but the direction is different, i.e., in (B) the values of Y increase as the values of X increase while in (C) the values of Y decrease as the values of X increase. In addition, the higher the correlation in either direction (positive or negative), the more linear the association between two variables.

Covariance

The covariance quantifies the strength of association between two or more sets of random variables.

Let X and Y be two random variable with the same population size N , we define the covariance as the following expectation value

$$\text{Cov}(X, Y) = E[(X - \mu_x)(Y - \mu_y)] \quad (1)$$

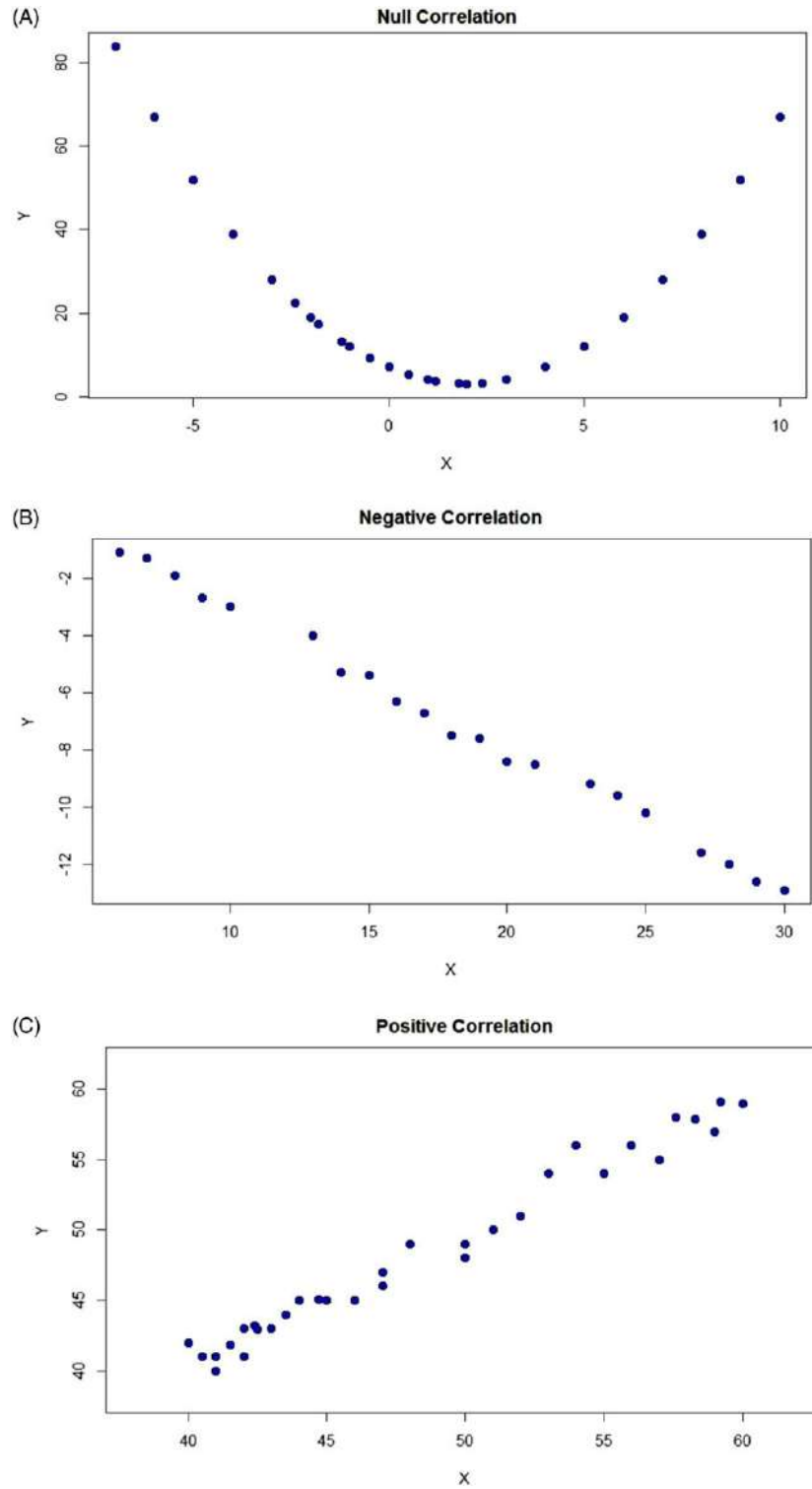


Fig. 1 The plot (A) shows that there is no relationship between the two variables X and Y ; the plot (B) displays that exist a positive linear relationship between the two variables X and Y ; the plot (C) exhibits a negative linear trend between the two variables X and Y .

where $\mu_X = E[X]$ and $\mu_Y = E[Y]$ are the population means of X and Y , respectively. We often denote covariance by σ_{XY} . The variance is a special case of the covariance when the two variables X and Y are identical. In fact, let $\mu = E[X]$ the population mean of X . Then, the covariance is given by

$$\sigma_{XX} = \text{Cov}(X, X) = E[(X - \mu_X)^2] = \text{Var}(X) = \sigma_X^2 \quad (2)$$

By using the linearity property of expectations, formula (1) can be further simplified as follow

$$\begin{aligned}\sigma_{XY} &= \text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])] \\ &= E[XY] - XE[X] - E[X]Y + E[X]E[Y] \\ &= E[XY] - E[X]E[X] - E[X]E[Y] + E[X]E[Y] = E[XY] - E[X]E[Y]\end{aligned}$$

In other words, the covariance is the population mean of the pairwise cross-product XY minus the cross-product of the means. The formula is given by

$$\sigma_{XY} = \text{Cov}(X, Y) = \mu_{XY} - \mu_X \mu_Y \quad (3)$$

where μ_{XY} is the joint population mean of X and Y . Given a set of paired variables (x, y) with simple size n , the sample covariance is given by

$$\sigma_{xy} = \text{Cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (4)$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ and } \bar{y} = \frac{1}{n} \sum_{j=1}^n y_j$$

are the sample means of variable x and y , respectively. Positive values of covariance, $\sigma_{xy} > 0$, means that the variables are positively related, i.e., the terms $(x_i - \bar{x})(y_i - \bar{y})$ in the sum are more likely to be positive than negative; a negative covariance, $\sigma_{xy} < 0$, indicates that the variables are inversely related, i.e., the terms $(x_i - \bar{x})(y_i - \bar{y})$ in the sum are more likely to be negative than positive. If $\sigma_{xy} = 0$, then the variables x and y are uncorrelated or independent between them. In other words, the sample covariance is positive if y increases with increasing x , negative if y decreases as x increases, and zero if there is no linear tendency for y to change with x . An alternative formula of sample covariance is the following formula (similar to that for a sample variance),

$$\sigma_{xy} = \text{Cov}(x, y) = \frac{n(\overline{xy} - \bar{x}\bar{y})}{n-1} \quad (5)$$

where

$$\overline{xy} = \frac{1}{n} \sum_{i=1}^n x_i y_i$$

is the joint sample mean of x and y . In probability theory, the covariance is a measure of the joint variability of two random variables. In particular, if X and Y are discrete random variables, with joint support S , then the covariance of X and Y is:

$$\text{Cov}(X, Y) = \sum \sum_{(x,y) \in S} (x_i - \mu_x)(y_j - \mu_y) f(x, y) \quad (6)$$

While on the contrary, if X and Y are continuous random variables with supports S_1 and S_2 , respectively, then the covariance of X and Y is:

$$\text{Cov}(X, Y) = \int_{S_1} \int_{S_2} (x_i - \mu_x)(y_j - \mu_y) f(x, y) dx dy \quad (7)$$

In Eqs. (6) and (7), the function $f(x, y)$ is a joint probability distribution, i.e., a probability distribution that gives the probability that X and Y fall in any particular range or discrete set of values specified by the two variables. For instance, we consider two discrete random variable X and Y (bivariate distribution) as illustrated in Table 1. By using the formula (6) the absolute value of covariance, is equal to 0.005. This value indicates a weak degree of association between X and Y .

Correlation Coefficients

In this section, we discuss the most widely used measures of associations between variables: Pearson's product moment correlation coefficient, Spearman's rank correlation coefficient and Kendall's correlation coefficient (Chok, 2010). The correct use of correlation coefficient depends on the type of variables. Pearson's product moment correlation coefficient is used only for continuous variables while the Spearman's and Kendall's correlation coefficients are adopted for either ordinal or continuous variables

Table 1 Joint probability mass function (or bivariate distribution) of two discrete random variables X and Y

$x \backslash y$	1	2	3	$f_X(x)$
1	0.25	0.00	0.25	0.50
2	0.15	0.10	0.25	0.50
$f_Y(y)$	0.40	0.10	0.50	1

Note: The function $f_X(x)$ and $f_Y(y)$ are called marginal distributions. The mean of X and Y are equal to 1.5 and 2, respectively.

(Mukaka, 2012). In addition, the first correlation coefficient is used to quantify linear relationships, the last two correlation coefficients are applied for measuring non-linear (or monotonic) associations.

Pearson's product moment correlation coefficient

Pearson's product moment correlation coefficient r , called also linear correlation coefficient, measures the linear relationship between two continuous variables (Pearson, 1930). Let x and y be the quantitative measures of two random variables on the same sample of n . The formula for computing the sample Pearson's correlation coefficient r is given by

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (8)$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ and } \bar{y} = \frac{1}{n} \sum_{j=1}^n y_j$$

are the sample means of variable x and y , respectively. In other words, assuming that the sample variances of x and y are positive, i.e., $s_x^2 > 0$ and $s_y^2 > 0$, the linear correlation coefficient r can be written as the ratio of the sample covariance of the two variables to the product of their respective standard deviations s_x and s_y ,

$$r = \frac{\text{Cov}(x, y)}{s_x s_y} \quad (9)$$

Hence, the correlation coefficient is a scaled version of covariance. The sample correlation measurement r ranges between -1 and $+1$. If the linear correlation between x and y is positive (i.e., higher levels of one variable are associated with higher levels of the other) results $r > 0$, while if the linear correlation between x and y is negative (i.e., higher levels of one variable are associated with lower levels of the other) results $r < 0$. The value $r = 0$ indicates absence of any association (positive or negative) between x and y . The sign of the linear correlation coefficient indicates the direction of the association, while the magnitude of the correlation coefficient denotes the strength of the association. If the correlation coefficient is equal to $+1$ the variables have a perfect linear positive correlation. This means that if one variable increases, the second increases proportionally in the same direction. If the correlation coefficient is zero, no relationship exists between the variables. If correlation coefficient is equal to -1 , the variables are perfectly negatively correlated (or inversely correlated) and move in opposition to each other. If one variable increases, the other one decreases proportionally. In addition, when two random variables X and Y are normally distributed, the population Pearson's product moment correlation coefficient is given by

$$\rho = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (10)$$

where σ_x and σ_y are the population standard deviations of X and Y , respectively. This coefficient is affected by extreme values and it is therefore not significant when either or both variables are not normally distributed.

Spearman's rank correlation coefficient

Spearman's correlation coefficient evaluates the monotonic relationship between two continuous or ordinal variables (Spearman, 1904). In a monotonic relationship, the variables tend to change together, but not necessarily at a constant rate. Given two random variables x and y , Spearman's rank correlation coefficient computes the correlation between the rank of the two variables. The sample Spearman's rank correlation coefficient r_s is given by the following expression

$$r_s = \frac{\sum_{i=1}^n (x'_i - \bar{x}')(y'_i - \bar{y}')}{\sqrt{\sum_{i=1}^n (x'_i - \bar{x}')^2} \sqrt{\sum_{i=1}^n (y'_i - \bar{y}')^2}} \quad (11)$$

where x' is the rank of x and y' is the rank of y . In other words, it is a rank-based version of the Pearson's correlation coefficient. It ranges from -1 to $+1$. A strong monotonically increasing (or decreasing) association between two variables usually leads to positive (or negative) values of all correlation coefficients simultaneously. Moreover, for weak monotone associations, different correlation coefficients could also be of a different sign. Similar to the Pearson correlation coefficient, Spearman's correlation coefficient is 0 for variables that are correlated in a non-monotonic way. Unlike the Pearson's correlation coefficient, r_s is equal to $+1$ for both linearly and not linearly correlated variables. In addition, there is no requirement of normality for the variables. The corresponding population Spearman's rank correlation coefficient, denoted as ρ_s , describes the strength of a monotonic relationship. This coefficient is computed when one or both variables are skewed or ordinal and it is robust when extreme values are present. An alternative formula used to calculate the Spearman rank correlation is

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (12)$$

where d_i is the difference between the ranks of corresponding values x_i and y_i .

Kendall's correlation coefficient

Kendall's correlation coefficient τ is used to measure the monotonic association between two ordinal (not necessarily continuous) variables (Kendall, 1970). The formula to compute τ is

$$\tau = \frac{\sum_{i=1}^n \sum_{j=1}^n \text{sign}(x_i - x_j) \text{sign}(y_i - y_j)}{n(n-1)} \quad (13)$$

where

$$\text{sign}(x_i - x_j) = \begin{cases} 1 & \text{if } \text{sign}(x_i - x_j) > 0 \\ 0 & \text{if } \text{sign}(x_i - x_j) = 0 \\ -1 & \text{if } \text{sign}(x_i - x_j) < 0 \end{cases} \quad \text{and} \quad \text{sign}(y_i - y_j) = \begin{cases} 1 & \text{if } \text{sign}(y_i - y_j) > 0 \\ 0 & \text{if } \text{sign}(y_i - y_j) = 0 \\ -1 & \text{if } \text{sign}(y_i - y_j) < 0 \end{cases}$$

This coefficient measures the discrepancy between the number of concordant and discordant pairs. Any pairs of ranks (x_i, y_i) and (x_j, y_j) are said to be concordant when $x_i < x_j$ and $y_i < y_j$, or $x_i > x_j$ and $y_i > y_j$, or $(x_i - x_j)(y_i - y_j) > 0$. Similarly, any pairs of ranks (x_i, y_i) and (x_j, y_j) are said to be discordant when $x_i < x_j$ and $y_i > y_j$, or $x_i > x_j$ and $y_i < y_j$, or $(x_i - x_j)(y_i - y_j) < 0$. As the two previous correlation coefficients, also τ ranges from -1 to $+1$. It is equal to $+1$ for concordant pairs and -1 for discordant pairs. Both Kendall and Spearman coefficients are formulated as special cases of the Person correlation coefficient.

Confidence Intervals and Testing Hypothesis

Suppose that x and y are two normally distributed variables (mean 0 and standard deviation 1), then the joint distribution is still a normal distribution with probability density

$$f(x) = \frac{1}{2\pi(1-r^2)} \exp \frac{x^2 + 2xy + y^2}{2(1-r^2)}$$

where r is the linear correlation coefficient. This function is called bivariate normal distribution. If the form of the joint distribution is not normal, or if the form is unknown, inferential procedures are invalid, although descriptive measures may be computed. Under the assumption of bivariate normality, given a sample correlation coefficient r , estimated from a sample size of n , we are interested to test if two variables X and Y are linearly correlated, i.e., if $\rho \neq 0$. To estimate ρ (usually unknown), we use the sample correlation statistic r . In particular, we consider the following test statistics

$$r_{obs} = r \sqrt{\frac{n-2}{1-r^2}} \sim T_{(n-2)} \quad (14)$$

which is a statistics distributed as a Student's t variable with $n-2$ degrees of freedom. To test if $\rho \neq 0$, the statistician and biologist Sir Ronald Aylmer Fisher (1921) developed a transformation of r that tends to become normal quickly as the population size n increases. Fisher's z -transformation is a function of r whose sampling distribution of the transformed value is close to normal. It is also called the r to z transformation and it is defined as

$$z = 0.5 \ln \left(\frac{1+r}{1-r} \right) = \text{arctanh}(r) \quad (15)$$

where $\ln$ is the natural logarithm function and arctanh is the inverse hyperbolic tangent function. Then, z is approximately normally distributed with mean

$$\bar{z} = 0.5 \ln \left(\frac{1+\rho}{1-\rho} \right) \quad (16)$$

and standard error

$$\sigma_z = 1/\sqrt{n-3} \quad (17)$$

where n is the sample size. Fisher's z -transformation and its inverses

$$r = \frac{\exp(2z) - 1}{\exp(2z) + 1} = \text{arctanh}(z) \quad (18)$$

can be used to construct a confidence interval for r using standard normal theory and derivations. If the distribution of z is not strictly normal, it tends to be normal rapidly as the sample size increases for any values of ρ . A confidence interval gives an estimated range of r values which is likely to include an unknown population parameter ρ . Generally, it is calculated at a confidence level, usually 95% (i.e., the significance level α is equal to 0.05). If the confidence interval includes 0 we can say that the

Critical regions – Standard normal distribution

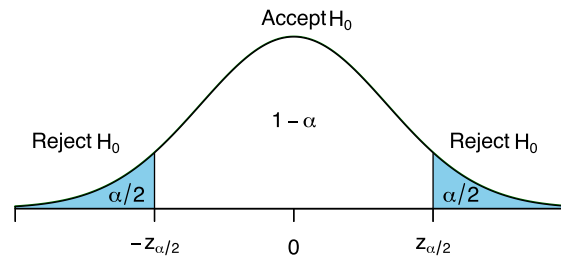


Fig. 2 Critical regions of the standard normal distribution.

Scatter plot

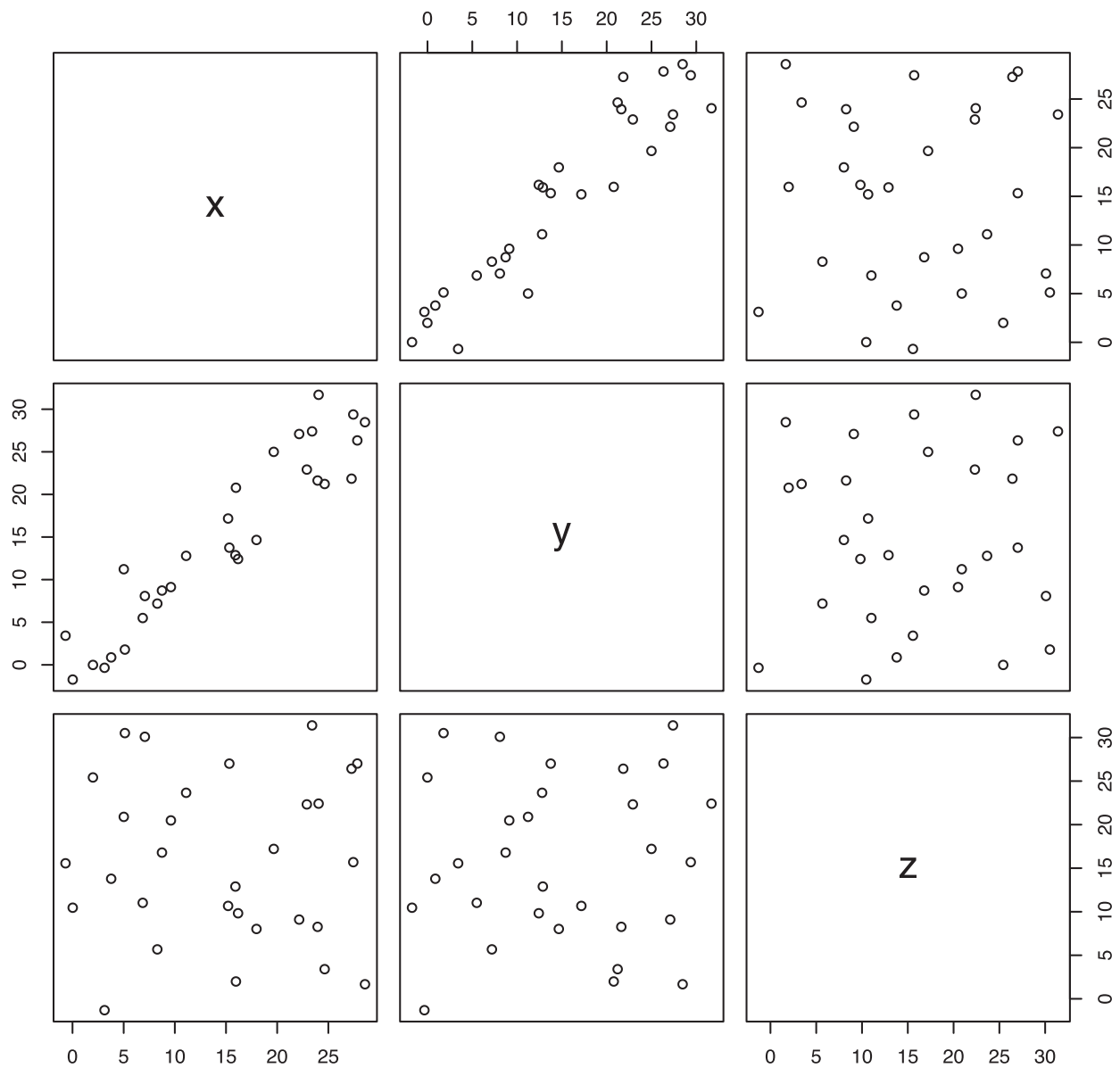


Fig. 3 Scatterplot shows a positive strong linear relationship between x and y variables (on the left of panel); while, it shows a weak association in the other cases; in particular, there is a negative correlation between variables x and z and a positive correlation between variables y and z (on the right of panel).

Table 2 Vital capacity data

<i>Number</i>	<i>Sex</i>	<i>Height</i>	<i>PEFR</i>	<i>vc</i>
1	1	180.6	522.1	4.74
2	1	168	440	3.63
3	1	163	428	3.40
4	1	171	536.6	3.75
5	1	177	513.3	3.81
6	1	169.4	510	2.80
7	1	161	383	2.90
8	1	170	455	3.88
9	1	158	440	2.40
11	1	161	461	2.60
11	1	163	370	2.72
12	1	155	503	2.20
13	1	171	430	3.38
14	1	171.5	442	2.99
15	1	167.6	595	3.06
17	1	166.6	455	3.06
18	1	167	500	3.72
19	1	163	548	2.82
20	1	172	463	2.83
21	1	155.4	475	3.06
22	1	165	485	3.07
23	1	174.2	540	4.27
24	1	167	415	3.80
25	1	162	475	2.88
26	1	172	490	4.47
27	1	161	470	3.40
28	1	155	450	2.65
29	1	162	450	3.12
30	1	174	540	4.02
31	1	161	475	2.80
32	1	166	430	3.69
33	1	166	510	3.66
34	1	161	470	2.56
35	1	168	430	2.78
36	1	167	470	3.48
37	1	166	440	3.03
38	1	164	485	2.90
39	1	162	550	2.96
40	1	176	535	3.77
41	1	166	485	3.50
42	1	160	360	2.30
43	1	161.2	480	3.39
44	1	167.8	480	3.70
45	2	181	580	—
46	2	170	560	—
47	2	171	460	—
48	2	184	611	—
49	2	184	600	—
50	2	188	590	—
51	2	186	650	—
52	2	187	600	—
53	2	181	630	—
54	2	181	670	—
55	2	177	515	—
56	2	167	470	—
57	2	182	550	—
58	2	172	620	—
59	2	190	640	—
60	2	178	680	—
61	2	184	600	—
62	2	170	510	—

Table 2 Continued

<i>Number</i>	<i>Sex</i>	<i>Height</i>	<i>PEFR</i>	<i>vc</i>
63	2	174	550	–
64	2	167	530	–
65	2	178	530	–
66	2	182	590	–
67	2	176	480	–
68	2	175	620	–
69	2	181	640	–
70	2	168	510	–
71	2	178	635	–
72	2	174	615.8	–
73	2	180.7	547	–
74	2	168	560	–
75	2	183.7	584.5	–
76	2	188	665	–
77	2	189	540	–
78	2	177	610	–
79	2	182	529	–
80	2	174	550	–
81	2	180	545	–
82	2	178	540	–
83	2	177	792	–
84	2	170	553	–
85	2	177	530	–
86	2	177	532	–
87	2	172	480	–
88	2	176	530	–
89	2	177	550	–
90	2	164	540	–
91	2	181	570	–
92	2	178	430	–
93	2	167	598	–
94	2	171.2	473	–
95	2	177.4	480	–
96	2	171.3	550	–
97	2	183.6	540	–
98	2	183.1	628.3	–
99	2	172	550	–
100	2	181	600	–
101	2	170.4	547	–
102	2	171.2	575	–

Note: The dataset contains the following variables: sex (1 = female, 2 = male), height, peak flow (PEFR), and vital capacity (vc) for 44 female and 58 male medical students.

population ρ is not significantly different from zero, at a given level of confidence α . More precisely, using Fisher's z transformation, we calculate the confidence interval at a confidence level α as following:

1. Given the observed correlation r , use Fisher's z transformation to compute the transformed sample correlation z , formula (15), the relative mean $\bar{z}$, formula (16), and standard deviation σ_z , formula (17).
2. Calculate the two-sided confidence limits (lower and upper) for z

$$\left(z - z_{\alpha/2} \sqrt{\frac{1}{n-3}}, z + z_{\alpha/2} \sqrt{\frac{1}{n-3}} \right) \quad (19)$$

where the critical value $z_{\alpha/2}$ depends on the significance level α . The value $\alpha/2$, is the area under the two tails of a standard normal distribution (see Fig. 2).

3. Un-transform the end points of the CI above using the arc tangent transformation, formula (19), in order to derive the confidence limits for the population correlation ρ as (r_{lower}, r_{upper}) .

Confidence intervals are more informative than the simple results of hypothesis tests since they provide a range of possible values for the unknown parameter.

In order to perform hypothesis testing for the population correlation coefficient ρ , we first formulate the null and alternative hypothesis as follow

$$H_0 : \rho = 0 \text{ (X and Y are uncorrelated)} \text{ versus } H_1 : \rho \neq 0 \text{ (X and Y are correlated)}.$$

Then, we use the computed test statistic r_{obs} , formula (14), to find the relative p -value and to make a decision. If the p -value is smaller than the significance level α , then we reject the null hypothesis H_0 and accept the alternative H_1 . In this case, we conclude that exists a linear relationship in the population between the two variables at the level α . If the p -value is larger than the significance level α , we accept the null hypothesis H_0 . In this case, we deduce that there is no linear relationship in the population between the two variables at the level α . Typically, the significant level α is equal to 0.05 or 0.01. This approach is called p -value approach. An alternative method to make a decision is the critical value approach. We find the critical value $t_{\alpha/2, n}$ using the Student's t distribution or t -table (where α is the significance level and n is the degree of freedom) and compare it to the observed r . If the test statistic is more extreme than the critical value, then the null hypothesis is rejected in favor of the alternative hypothesis. If the test statistic is not as extreme as the critical value, then the null hypothesis is not rejected. In other words, we reject the null hypothesis H_0 at the level α if $r_{obs} \leq t_{\alpha/2, n-2}$ and $r_{obs} \geq t_{\alpha/2, n-2}$. Vice versa, we accept the null hypothesis H_0 at the level α if $-t_{\alpha/2, n-2} \leq r_{obs} \leq t_{\alpha/2, n-2}$. For instance, see Fig. 2.

Note that the Fisher z -transformation and the hypothesis test explains above are mainly associated with the Person correlation coefficient for bivariate normal observations, but similar consideration can also be applied to Spearman (Bonett and Wright, 2000; Zar, 2014) and Kendall (Kendall, 1970) correlation coefficients in more general cases.

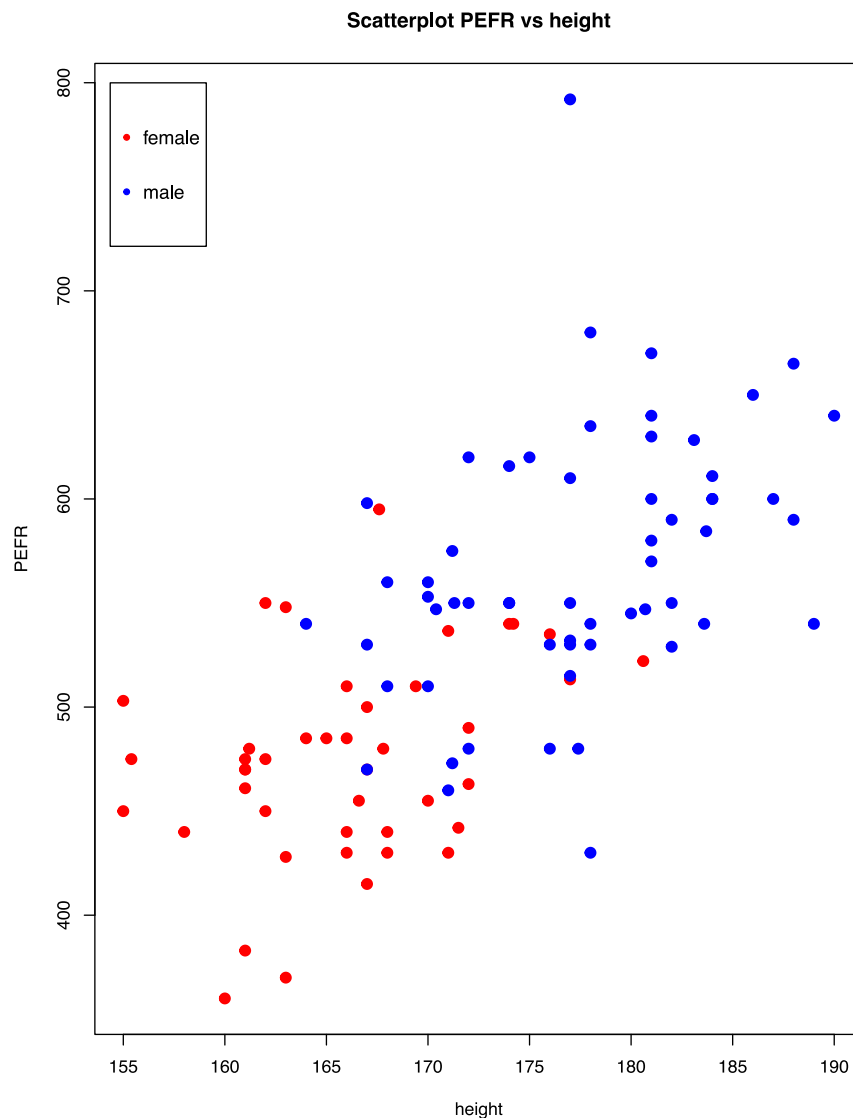


Fig. 4 Scatterplot between the peak expiratory flow rate (PEFR) and the height of medical students (male and female).

Table 3 In the table are reported the covariance and the three different correlation coefficients (Pearson, Spearman and Kendall) for the medical students dataset (female and male). The regression coefficients are also shown

		Height	PEFR
Covariance	height	69.02	396.20
	PEFR	396.20	5467.74
Pearson Correlation	height	1.00	0.64
	PEFR	0.64	1.00
Spearman Correlation	height	1.00	0.67
	PEFR	0.67	1.00
Kendall Correlation	height	1.00	0.48
	PEFR	0.48	1.00
Regression coefficients		$a = -461.84, b = 5.74$	

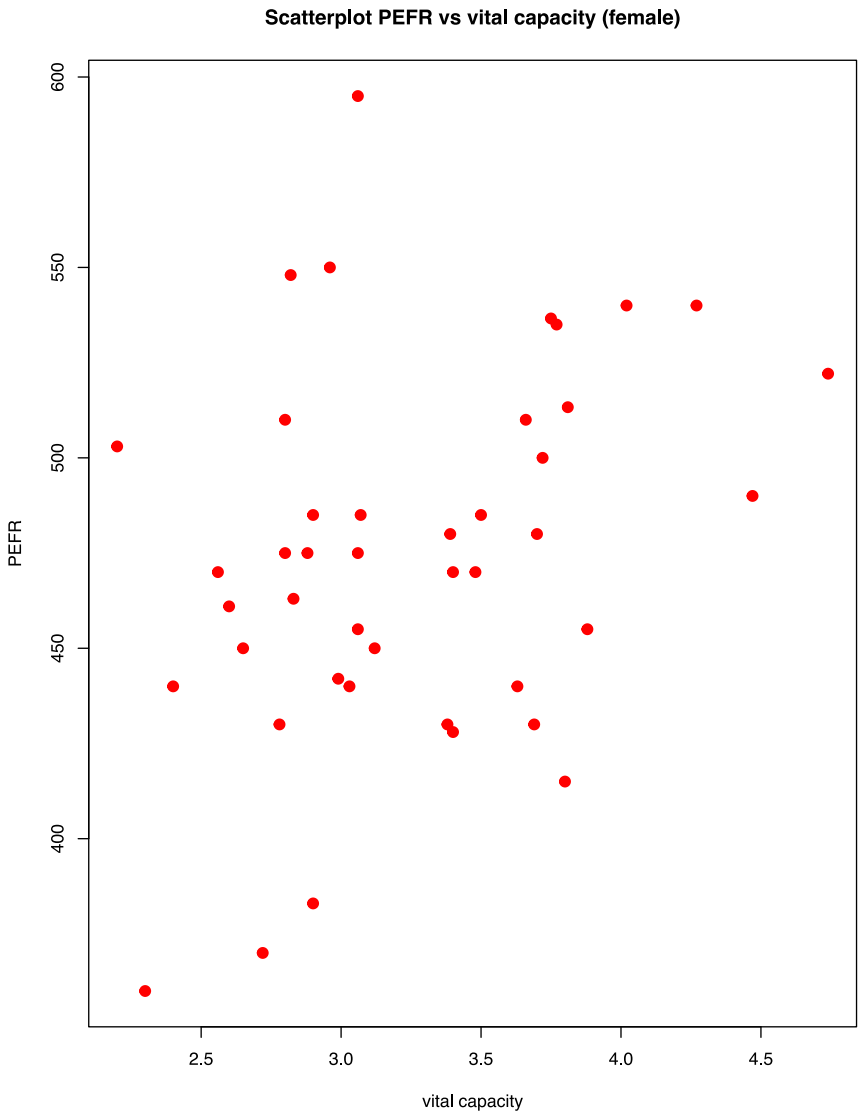


Fig. 5 Scatterplot between the vital capacity and the peak expiratory flow rate (PEFR) of female medical students.

Correlation and Regression

Regression analysis is used to model the relationship between a response variable (dependent variable) and one or more predictor variables (independent variables). Denoting by y the dependent variable and by x the independent variable, the simple linear correlation can be represented using the least squares regression line

$$y = a + bx + e \quad (20)$$

where a is the point where the line crosses the vertical axis y , and b shows the amount by which y changes for each unit change in x . We refer to a as the y -intercept and b as the slope of the line (or regression coefficient). The value e is the residual error. Letting $\hat{y} = \hat{a} + x\hat{b}$ be the value of y predicted by the model, then the residual error is the deviation between observed and the predicted values of the outcome y , i.e., $e = y - \hat{y}$. The aim of linear regression analysis is to estimate the model parameters a and b , in order to give the best fit for the joint distribution of x and y . The mathematical method of least-squares linear regression provides the best-fit solutions. Without making any assumptions about the true joint distribution of x and y , the least-squares linear regression minimizes the average value of the squared deviations of the observed y from the values predicted by the regression line $\hat{y}$. That is, the least-squares solution yields the values of a and b that minimize the mean squared residual, i.e., $e^2 = (y - \hat{y})^2$. The residual is the vertical distances of the data points. The least-squares regression line equation is obtained from sample data by simple arithmetic calculations. In particular, the estimations of a and b that

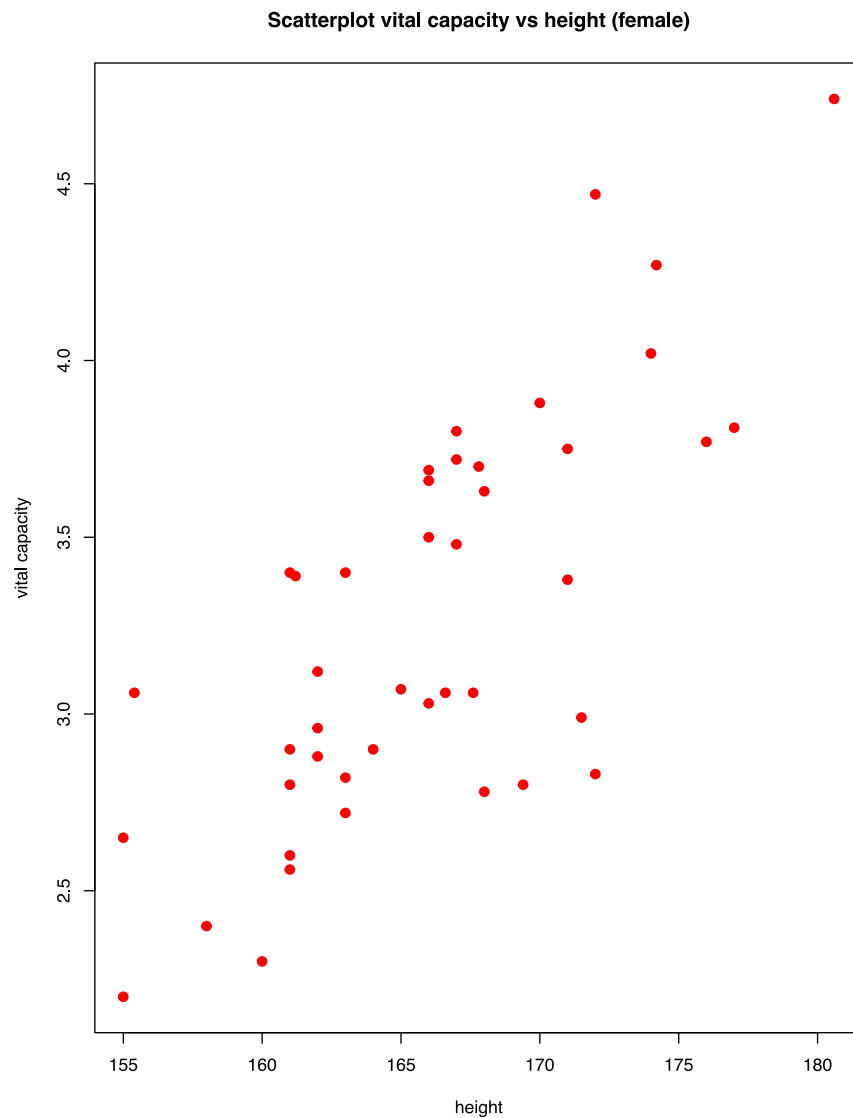


Fig. 6 Scatterplot between the height and the estimated vital capacity of female medical students.

minimizes the mean squared residual $e^2 = (y - \hat{y})^2$ are given by

$$\hat{b} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \hat{a} = \bar{y} - b\bar{x} \quad (21)$$

where x_i and y_i are the corresponding values of each data point (X, Y) , $\bar{x}$ and $\bar{y}$ are the sample means of the X and Y , respectively, and n the sample size. In other words, the estimate of intercept $\hat{a}$ and slope $\hat{b}$ are

$$\hat{b} = \frac{\text{Cov}(x, y)}{\text{Var}(x)}, \hat{a} = \bar{y} - b\bar{x} \quad (22)$$

Thus, the least-squares estimators for the intercept a and slope b of a linear regression are simple functions of the covariances, variances and observed means and define the straight line that minimizes the amount of variation in y explained by a linear regression on x . In addition, the residual errors around the least squares regression are uncorrelated with the predictor variable x . Unlike the correlation analysis, that involves the relationship between two random variables, in the linear regression only the dependent variable is required to be random, while the independent variable is fixed (nonrandom or mathematical). In addition, as for the linear regression, in the correlation analysis we fit a straight line to the data either by minimizing

$$\sum_{i=1}^n (x_i - \hat{x}_i)^2 \text{ or } \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

In other words, we apply a linear regression of X on Y as well as a linear regression of Y on X . These two fitted lines in general are different. Also in this case we use the scatter diagram to plot the regression line. A negative relationship is represented by a falling regression line (regression coefficient $b < 0$), a positive one by a rising regression line ($b > 0$).

Data Analysis and Results

In this section, simulated and clinical data are presented for the study of correlation and regression analysis.

Simulated Data

We generate three random variables x , y , z from three bivariate normal distribution of 30 observations with means equal to zero and standard deviations 2, 3 and 4, respectively. For each pairs of variables, we first plot the scatter diagram in order to visualize the type of correlation that exists among them, and then we compute the covariance and the three correlation coefficients. Fig. 3 shows that the variables x and y have a positive covariance and a positive correlation, while the variables y and z have a negative covariance and a negative correlation. In particular, the Person correlation coefficient $r_{xy} = 0.94$ indicates that there is a strong positive association between the variables x and y , whereas Person correlation coefficient $r_{yz} = 0.03$ shows a weak positive relationship between the variables y and z . These results are confirmed by Spearman correlation coefficients $r_{sxy} = 0.93$ and $r_{syx} = 0.05$ and Kendall correlation coefficients $\tau_{xy} = 0.78$ and $\tau_{yz} = 0.04$. On the contrary, the variables x and z present a negative covariance and a negative correlation. In fact, Person correlation coefficient is $r_{xz} = -0.01$ which indicates a

Table 4 In the table are reported the covariance and the three different correlation coefficients (Pearson, Spearman and Kendall) for the female medical students dataset. The regression coefficients are also shown

		Height	PEFR	Vital capacity
Covariance	Height	34.30	100.16	2.50
	PEFR	100.16	2407.32	9.62
	Vital capacity	2.50	9.62	0.33
Pearson Correlation	Height	1.00	0.35	0.74
	PEFR	0.35	1.00	0.34
	Vital capacity	0.74	0.34	1.00
Spearman Correlation	Height	1.00	0.34	0.70
	PEFR	0.34	1.00	0.33
	Vital capacity	0.70	0.33	1.00
Kendall Correlation	Height	1.00	0.23	0.54
	PEFR	0.23	1.00	0.24
	Vital capacity	0.54	0.24	1.00
Regression coefficients	PEFR vs. vital capacity	$a = 380.10, b = 28.87$		
	Vital capacity vs. height	$a = -8.81, b = 0.07$		

weak negative correlation. This result is validated by Spearman correlation coefficient $r_{sz} = -0.02$ and Kendall correlation coefficient $r_{xz} = -0.02$. Finally, we test if these three correlation coefficients are statistically significant. Using parametric assumptions (Pearson, dividing the coefficient by its standard error, giving a value that follow a t -distribution), for the pair (x,y) the confidence interval at level 95% ($\alpha=0.05$) is $(0.88, 0.97)$. In other words, we reject the null hypothesis H_0 . In fact, the p -value $= 1.118e-14$ is statistically significant, since it is less than 0.05. Similar results are obtained when data violate parametric assumptions (Spearman and Kendall). Also in these cases the p -value is statistically significant (p -value $= 5.562e-08$ and p -value $= 1.953e-12$). On the contrary, for the pairs (x,z) and (y,z) , we accept the null hypothesis H_0 for the three type of correlation coefficients (Person, Spearman and Kendall) take into account. In fact, the three tests are not statistically significant (p -value > 0.05) (Table 2).

Clinical Data

The clinical data is downloaded (See Section Relevant Websites). The data is composed by 44 female and 58 male medical students. It contains information about the peak expiratory flow rate (PEFR, measured in litre/min), which is a form of pulmonary function test used to measure how fast a person can exhale, the height of students and the vital capacity (vc), which is the maximum amount of air a person can expel from the lungs after a maximum inhalation (measured in litre). In particular, the vital capacity (vc) is reported only for female students. The scatter plot in Fig. 4 shows a positive correlation between the peak expiratory

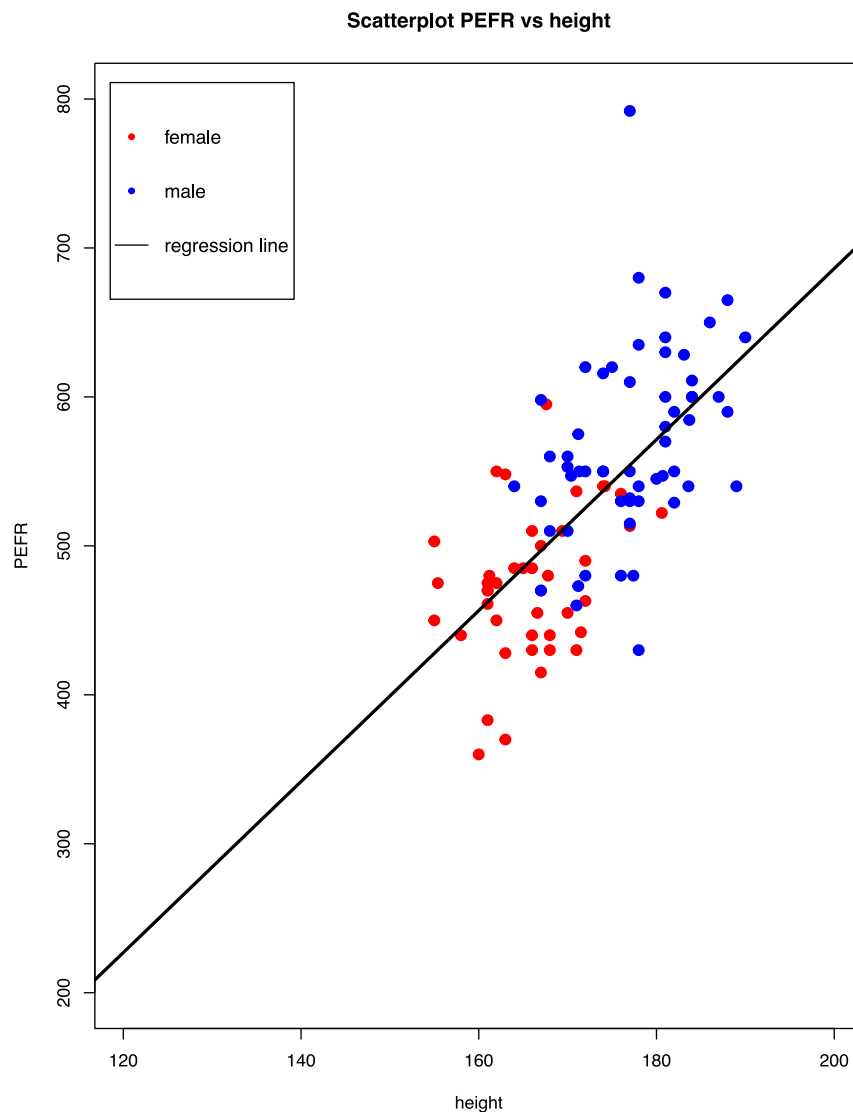


Fig. 7 Least squares line between the variables PEFR and height of the medical students (female and male).

flow rate (PEFR) and the height of all students (male and female). This result is confirmed by the computation of covariance and correlation coefficients (Person, Spearman and Kendall) as shown in [Table 3](#). The Person and Spearman correlation coefficients indicate a moderate positive relationship ($r=0.64$ and $r_s=0.67$), while the Kendall correlation coefficient indicates a weak positive relationship ($\tau=0.48$). In particular, for the female group of medical students, [Figs. 5](#) and [6](#) show a positive association for the variables (vc, PEFR) and (height, vc), respectively. Also in this case the positive relationship is validated by the computation of covariance and correlation coefficients (Person, Spearman and Kendall) as shown in [Table 4](#). In particular, the three correlation coefficients indicate a weak positive relationship ($r=0.34$, $r_s=0.33$ and $\tau=0.24$) for the variables (vc, PEFR), while they indicate a strong positive relationship ($r=0.74$, $r_s=0.70$ and $\tau=0.54$) for the variables (height, vc). Finally, the regression analysis is also performed. In [Fig. 7](#), we plot the least-squares line between the dependent variable PEFR and the independent variable height when female and male are considered together. The least-squares equation is $\hat{y} = -461.84 + 5.74\hat{x}$. On the other hand, in [Figs. 8](#) and [9](#), the least-squares line between PEFR and vital capacity (vc) and vital capacity and height are plotted, respectively. The least-squares equations are $\hat{y} = 380.10 + 28.87\hat{x}$ and $\hat{y} = -8.81 + 0.07\hat{x}$.

Software

We use the R statistical software (see Relevant Website section) to elaborate the strength of correlation and to analyze the regression relationship between the variables under investigated in both cases studies. In particular, we apply the common used statistical packages in R.

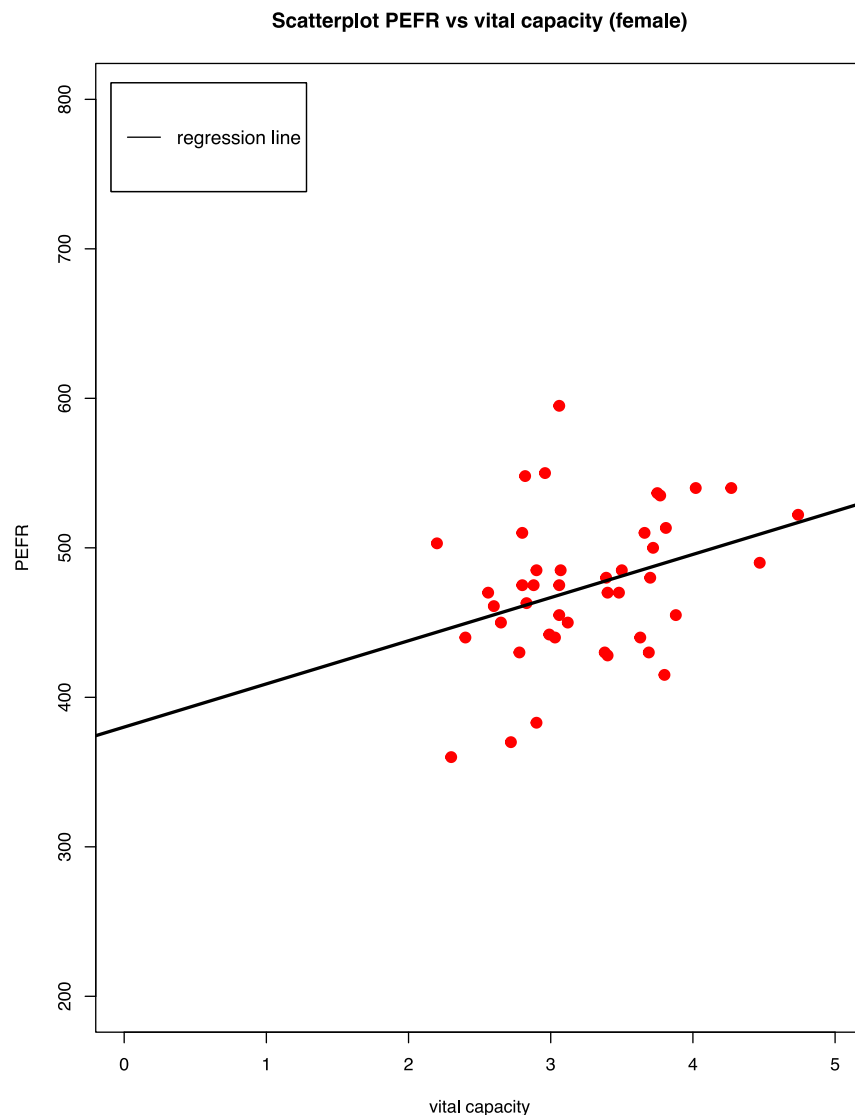


Fig. 8 Least squares line between the variables PEFR and vital capacity of the female medical students.

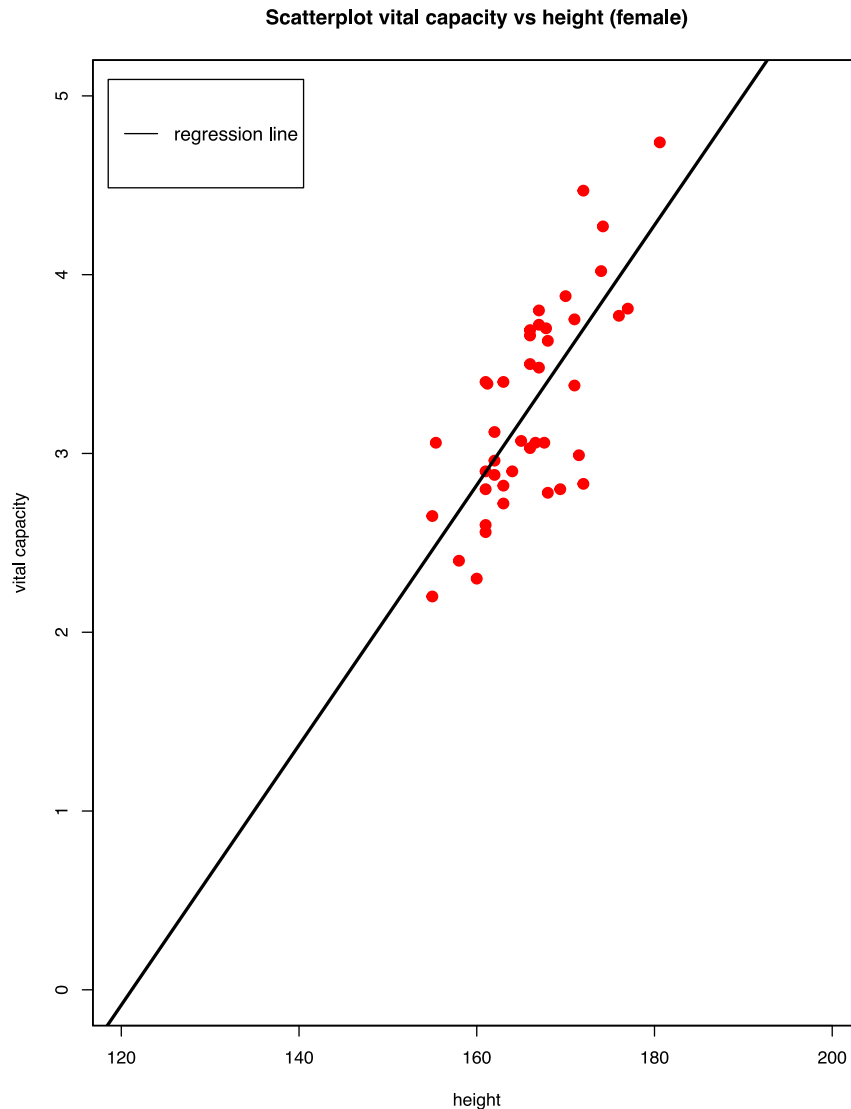


Fig. 9 Least squares line between the variables vital capacity and height of the female medical students.

Conclusions

Correlation analysis is an important statistical method for the analysis of medical data. It is used to investigate the relationship between two quantitative continuous variables. In particular, a correlation coefficient measures the strength of the relationship (magnitude) between two variables and the direction of the relationship (sign). We analyzed three different type of correlation coefficient. Pearson correlation coefficient quantifies the strength of a linear relationship between two variables. Spearman and Kendall correlation coefficients are two rank-based (or non-parametric) version of the Pearson coefficient. When two variables are normally distributed we use Pearson coefficient, otherwise, we apply Spearman or Kendall coefficient. Moreover, Spearman coefficient is more robust to outliers than is Pearson coefficient. Finally, since the correlation analysis does not establish if one variable is dependent and the other is independent, we introduce the regression analysis which is another statistical method used to describe a linear relationship between a depend variable (response or outcome) and one or more independent variables (predictors or explanatory variables). Therefore, the correlation analysis can be defined as a double linear regression of X on Y and of Y on X .

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Bonett, D.G., Wright, T.A., 2000. Sample size requirements for Pearson, Kendall, and Spearman correlations. *Psychometrika* 65, 23–28.
- Chok, N.S., 2010. Pearson's versus Spearman's and Kendall's correlation coefficients for continuous data. Dissertation, University of Pittsburgh.
- Gaddis, M.L., Gaddis, G.M., 1990. Introduction to biostatistics: Part 6, correlation and regression. *Annals of Emergency Medicine* 19 (12), 1462–1468.
- Kendall, M.G., 1970. *Rank Correlation Methods*, fourth ed. London: Griffin.
- Mukaka, M.M., 2012. A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal* 24 (3), 69–71.
- Pearson, K., 1930. *The Life, Letters and Labors of Francis Galton*. Cambridge University Press.
- Spearman, C., 1904. The proof and measurement of association between two things. *American Journal of Psychology* 15, 72–101.
- Zar, J.H., 2014. Spearman Rank Correlation: Overview. Wiley StatsRef: Statistics Reference Online.

Further Reading

- Bland, M., 2015. *An Introduction to Medical Statistics*. Oxford: Oxford University Press.
- Cox, D.R., Shell, E.J., 1981. *Applied Statistics. Principles and Examples*. London: Chapman and Hall.
- Daniel, W.W., Cross, C.L., 2013. *Biostatistics: A Foundation for Analysis in the Health Sciences*, tenth ed. John Wiley & Sons.
- Dunn, O.J., Clark, V.A., 2009. *Basic Statistics: A Primer for the Biomedical Sciences*. John Wiley & Sons.
- Gibbons, J.D., Kendall, M.G., 1990. *Rank Correlation Methods*. Edward Arnold.

Relevant Websites

- <https://www.users.york.ac.uk/~mb55/datasets/datasets.htm>
Selected Data-sets from Publications by Martin Bland.
- <https://www.r-project.org>
The R Project for Statistical Computing.

Regression Analysis

Claudia Angelini, Istituto per le Applicazioni del Calcolo “M. Picone”, Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

$Y = (Y_1, Y_2, \dots, Y_n)^T$ denotes the $n \times 1$ vector of responses

$\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T$ denotes the $n \times 1$ vector of errors/noise

$X = \begin{pmatrix} 1, x_{11}, x_{12}, \dots, x_{1p} \\ 1, x_{21}, x_{22}, \dots, x_{2p} \\ \dots \\ 1, x_{n1}, x_{n2}, \dots, x_{np} \end{pmatrix}$ denotes the $n \times (p+1)$ matrix of regressors (i.e., *design matrix*)

$x_i = (x_{i1}, \dots, x_{ip})^T$ (or $x_i = (1, x_{i1}, \dots, x_{ip})^T$) denotes the $p \times 1$ (or $(p+1) \times 1$) vector of covariates observed on the i th statistical unit

$X_k = (x_{1k}, \dots, x_{nk})^T$ denotes the $n \times 1$ vector of observations for the k th covariate

$\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$ denotes the $(p+1) \times 1$ vector of unknown regression coefficients

$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)^T$ denotes the $(p+1) \times 1$ vector of estimated regression coefficients

$\hat{Y} = (\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_n)^T$ denotes the $n \times 1$ vector of responses

$H = X(X^T X)^{-1} X^T$ denotes the $n \times n$ hat-matrix.

$E()$ denotes the Expected value

T_{n-p-1} denotes a Student distribution with $n-p-1$ degree of freedom

$t_{n-p-1, \alpha}$ denotes the $(1-\alpha)$ quantile of a T_{n-p-1} distribution

$\|\beta_i\|_2^2 = \sum_{i=1}^p \beta_i^2$ denotes the l_2 vector norm

$\|\beta_1\|_1 = \sum_{i=1}^p |\beta_i|$ denotes the l_1 vector norm

Introduction

Regression analysis is a well-known statistical learning technique useful to infer the relationship between a *dependent variable* Y and p *independent variables* $X = [X_1 | \dots | X_p]$. The dependent variable Y is also known as *response variable* or *outcome*, and the variables X_k ($k=1, \dots, p$) as *predictors*, *explanatory variables*, or *covariates*. More precisely, regression analysis aims to estimate the mathematical relation $f()$ for explaining Y in terms of X as, $Y=f(X)$, using the observations $(x_i, Y_i), i=1, \dots, n$, collected on n observed *statistical units*. If Y describes a univariate random variable the regression is said to be *univariate regression*, otherwise it is referred as *multivariate regression*. If Y depends on only one variable x (i.e., $p=1$), the regression is said *simple*, otherwise (i.e., $p>1$), the regression is said *multiple*, see (Abramovich and Ritov, 2013; Casella and Berger, 2001; Faraway, 2004; Fahrmeir et al., 2013; Jobson, 1991; Rao, 2002; Sen and Srivastava, 1990).

For the sake of brevity, in this chapter we limit our attention to *univariate regression* (simple and multiple), so that $Y = (Y_1, Y_2, \dots, Y_n)^T$ represents the vector of observed outcomes, and $X = \begin{pmatrix} x_1^T \\ \dots \\ x_n^T \end{pmatrix}$ represents the *design matrix* of observed covariates, where $x_i = (x_{i1}, \dots, x_{ip})^T$ (or $x_i = (1, x_{i1}, \dots, x_{ip})^T$). In this setting $X = [X_1 | \dots | X_p]$ is a p -dimensional variable ($p \geq 1$).

Regression analysis techniques can be organized into two main categories: non-parametric and parametric. The first category contains those techniques that do not assume a particular form for $f()$; while the second category includes those techniques based on the assumption of knowing the relationship $f()$ up to a fixed number of parameters β that need to be estimated from the observed data. When the relation between the explanatory variables, X , and the parameters, β , is linear, the model is known as *Linear Regression Model*.

The Linear Regression Model is one of the oldest and more studied topics in statistics and is the type of regression most used in applications. For example, regression analysis can be used for investigating how a certain phenotype (e.g., blood pressure) depends on a series of clinical parameters (e.g., cholesterol level, age, diet, and others) or how gene expression depends on a set of transcription factors that can up/down regulate the transcriptional level, and so on. Despite the fact that linear models are simple and easy to handle mathematically, they often provide an adequate and interpretable estimate of the relationship between X and Y . Technically speaking, the linear regression model assumes the response Y to be a continuous variable defined on the real scale and each observed data is modeled as

$$Y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i = x_i^T \beta + \varepsilon_i \quad i = 1, \dots, n$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$ is a vector of unknown parameters called *regression coefficients* and ε represents the *errors* or *noise term* that accounts for the randomness of the measured data or the residual variability not explained by X . The regression coefficients β can be estimated by fitting the observed data using the least squares approach. Under the Gauss-Markov conditions (i.e., the ε_i are assumed to be independent and identically distributed random variables, with zero mean and finite variance σ^2), the ordinary least squares estimates $\hat{\beta}$ are guaranteed to provide the *best linear unbiased estimator* (BLUE). Moreover, under the further assumptions that $\varepsilon \sim N(0, \sigma^2 I_n)$, $\hat{\beta}$ allows statistical inference to be carried out on the model, as described later. The validity of both the

Gauss-Markov conditions and the normal distribution of the error term are known as “white noise” conditions. In this context, linear regression model is also known as the regression of the “mean”, since it models the conditional expectation of Y given X , as follows $\hat{Y} = E(Y|X) = X^T \hat{\beta}$, where $E(Y|X)$ denotes the conditional expected value of Y for fixed values of the regressors X .

The linear regression model not only allows estimating the regression coefficients β as $\hat{\beta}$ (and hence quantifying the strength of the relationship between Y and each of the p explanatory variables when the remaining $p-1$ are fixed), but also selecting those variables that have no relationship with Y (when the remaining ones are fixed), as well as identifying which subsets of explanatory variables have to be considered in order to explain sufficiently well the response Y . These tasks can be carried out by testing the significance of each individual regression coefficient when the others are fixed, by removing the coefficients that are not significant and re-fitting the linear model and/or by using model selection approaches. Moreover, linear regression model can be also used for prediction. For this purpose, given the estimated values, $\hat{\beta}$, it is possible to predict the response, $\hat{Y}_0 = x_0^T \hat{\beta}$, corresponding to any novel value x_0 and to estimate the uncertainty of such prediction. The uncertainty depends on the type of prediction one wants to make. In fact, it is possible to compute two types of confidence intervals: the one for the expectation of a predicted value at a given point x_0 , and the one for a future generic observation at a given point x_0 .

As the number, p , of explanatory variables increases, the least squares approach suffers from a series of problems, such as lack of prediction accuracy and difficulty of interpretation. To address these problems, it is desirable to have a model with only a small number of “important” variables, which is able to provide a good explanation of the outcome and good generalization at the price of sacrificing some details. Model selection consists in identifying which subsets of explanatory variables have to be “selected” to sufficiently explain the response Y making a compromise referred as the *bias-variance trade-off*. This is equivalent to choosing between competing linear regression models (i.e., with different combinations of variables). On the one hand, one has to consider that including too few variables leads to so-called “underfit” of the data, characterized by poor prediction performance with high bias and low variance. On the other hand, selecting too many variables leads to so-called “overfit” of the data, characterized by poor prediction performance with low bias and high variance. Stepwise linear regression is an attempt to address this problem (Miller et al., 2002; Sen and Srivastava, 1990) constituting a specific example of subset regression analysis. Although model selection can be used in classical regression context, it is one of the most effective tool in high dimensional data analysis.

Classical regression deal with the case $n \geq p$ where n denotes the number of independent observations (i.e., the sample size) and p the number of variables. Nowadays, in many applications especially in biomedical science, high-throughput assays are capable of measuring from thousands to hundreds of thousands of variables on a single statistical unit. Therefore, one has often to deal with the case $p \gg n$. In such a case, ordinary least squares cannot be applied, and other types of approaches (for example, including the use of a penalization function) such as Ridge regression, Lasso or Elastic net regression (Hastie et al., 2009; James et al., 2013; Tibshirani, 1996, 2011) have to be used for estimating the regression coefficients. In particular, Lasso is very effective since it also performs also variable selection and has opened the new framework of high-dimensional regression (Bühlmann and van de Geer, 2011; Hastie et al., 2015). Model selection and high dimensional data analysis are strongly connected, and they might also benefit from dimension reduction techniques such as principal component analysis, or feature selection.

In the classical framework, Y is treated as a random variable, while X are considered fixed, hence, depending on the distribution of Y , different types of regression models can be defined. With X fixed, the assumptions on the distribution of Y are elicited through the distribution of error term $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^T$. As above mentioned, classical linear regression requires the error term to satisfy the Gauss-Markov conditions and be normally distributed. However, when the error term is not normally distributed, linear regression might be not appropriate. Generalized linear models (GLM) constitute a generalization of classical linear regression that allows the response variable Y to have an error distribution other than normal (McCullagh and Nelder, 1989). In this way, GLM generalize linear regression by allowing the linear model to be related to the response variable via a *link function*, and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. In this way GLM represent a wide framework that includes linear regression, logistic regression, Poisson regression, multinomial regression, etc. In this framework the regression coefficients can be estimated using the *maximum likelihood* approach, often solved by iteratively reweighted least squares algorithms.

In the following, we briefly summarize the key concepts and definitions related to linear regression, moving from the simple linear model, to the multiple linear model. In particular, we discuss the Gauss-Markov conditions and the properties of the least squares estimate. We discuss the concepts of model selection and also provide suggestions on how to handling outliers and deviation from standard assumptions. Then, we discuss modern forms of regression, such as Ridge regression, Lasso and Elastic Net, which are based on penalization terms and are particularly useful when the dimension of the variable space, p , increases. We conclude by extending the linear regression concepts to the Generalized Linear Models (GLM).

Background/Fundamentals

In the following we first introduce the main concepts and formulae for simple linear regression ($p=1$), then we extend the regression model to the case $p>1$. We note that, when $p=1$, simple linear regression is the “best” straight line through the observed data points, for $p>1$ it represents the “best” hyper-plane across the observed data points. Moreover, while Y has to be a quantitative variable, the X_k can be either quantitative or categorical. However, categorical covariates have to be transformed into a series of dummy variables using indicators.

Simple Linear Regression

As mentioned before, simple linear regression (Altman and Krzywinski, 2015; Casella and Berger, 2001) is a statistical model used to study the relationship $f()$ between two (quantitative) variables x and Y , where Y represents the response variable and x the explanatory variable (i.e., x is one particular X_k), assuming that the mathematical relation can be described by a straight line. In particular, the aim of simple linear regression is estimating the straight line that best fits the observed data, by estimating its coefficients (i.e., the intercept and the slope), quantifying the uncertainty of such estimate, testing the significance of the relationship, and finally using the estimated line to predict the value of Y using the observed values of x . Typically the x variable is treated as fixed, and the aim is estimating the relationship between the mean of Y and x (i.e., $E[Y|x]$ where E denotes the conditional expectation of Y given x).

More formally, to fix the notation and introduce the general concepts, assume we have collected n observations (x_i, Y_i) , $i = 1, \dots, n$. A simple linear regression model can be written as

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad i = 1, \dots, n$$

where the ε_i are independent and identically distributed random variables (with zero mean and finite variance σ^2) and $\beta_0 + \beta_1 x$ represents a straight line, β_0 being the intercept and β_1 the slope. In particular, when $\beta_1 > 0$, x and Y vary together; when $\beta_1 < 0$ then x and Y vary in opposite directions.

Let $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)^T$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T$ denote the vectors of the observed outcomes and the noise, respectively, $\mathbf{X} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{pmatrix}^T = \begin{pmatrix} 1_n \\ \mathbf{x} \end{pmatrix}^T$, the matrix of regressors (usually called the *design matrix*), and $\boldsymbol{\beta} = (\beta_0, \beta_1)^T$, the vector of unknown coefficients, then the regression problem can be rewritten in matrix form as,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The aim is estimating the coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1)^T$ that provide a “best” fit for the observed data. This “best” fit is often achieved by using the *ordinary least-squares* approach, i.e., by finding the coefficients that minimizes the sum of the squared residuals, or in mathematical terms by solving the following problem

$$\underset{\beta_0, \beta_1}{\operatorname{argmin}} \sum_{i=1}^n e_i^2 = \underset{\beta_0, \beta_1}{\operatorname{argmin}} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)^2$$

After straightforward calculations, it is possible to prove that

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$$

where $\bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}$ and $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ denote the sample mean of Y and x , respectively. The estimate of $\hat{\beta}_1$ can be rewritten as

$$\hat{\beta}_1 = \frac{s_{xY}}{s_{xx}} = r_{xY} \frac{s_{YY}}{s_{xx}}$$

where s_{xx} and s_{YY} denote the sample variance of x and Y , respectively; r_{xY} and s_{xY} denote the sample correlation and sample covariance coefficient between x and Y , respectively.

Given the estimated parameters $\hat{\beta}_0, \hat{\beta}_1$, it is possible to estimate the response, $\hat{Y}_i$, as

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \quad i = 1, \dots, n$$

The least squares approach provides a good estimate of the unknown parameters if the so-called Gauss-Markov conditions are satisfied (i.e., i. $E(\varepsilon_i) = 0$, $\forall i = 1, \dots, n$, ii. $E(\varepsilon_i^2) = \sigma^2$, $\forall i = 1, \dots, n$ and iii. $E(\varepsilon_i \varepsilon_j) = 0$, $\forall i \neq j$, where $E()$ denotes the expected value). In particular, according to the Gauss-Markov theorem, if such conditions are satisfied, the least squares approach provides the *best linear unbiased estimator* (BLUE) of the parameters, i.e., the estimate with the lowest variance compared to all linear unbiased estimators. Furthermore, if we also assume that $\varepsilon_i \sim N(0, \sigma^2)$ then, the least squares solution provides the best estimator among all unbiased estimators, and also the maximum likelihood estimator. Additionally, under such assumptions (“white noise” assumptions) it is possible to prove that

$$\frac{\beta_i - \hat{\beta}_i}{s_{\hat{\beta}_i}} \sim T_{n-2}, \quad i = 0, 1$$

where $s_{\hat{\beta}_i} = \sqrt{\frac{1}{n-2} \sum_{i=1}^n \frac{e_i^2}{(x_i - \bar{x})^2}}$ is the *standard error* of the estimator $\hat{\beta}_i$, and T_{n-2} denotes the Student T-distribution with $n-2$ degree of freedom. On the basis of this result, the $(1-\alpha)$ % *percent confidence interval* for the coefficient, β_i , is given by

$$\beta_i = \left[\hat{\beta}_i - s_{\hat{\beta}_i} t_{n-2, \frac{\alpha}{2}}, \hat{\beta}_i + s_{\hat{\beta}_i} t_{n-2, \frac{\alpha}{2}} \right], \quad i = 0, 1$$

where $t_{n-2, \frac{\alpha}{2}}$ denotes the $(1-\alpha/2)$ quantile of T_{n-2} .

Moreover, the significance of each coefficient can be evaluated by testing the null hypothesis $H_{0,i} : \beta_i = 0$ versus the alternative $H_{1,i} : \beta_i \neq 0$. To perform such test, it is possible to use the following test statistics $\frac{\hat{\beta}_i}{s_{\hat{\beta}_i}} \sim T_{n-2}$, $i = 0, 1$.

Finally, the *coefficient of determination*, R^2 , is commonly used to evaluate the *goodness of fit* within a simple linear regression model. It is defined as follows

$$R^2 = \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$$

We have that $R^2 \in [0, 1]$, where 0 (or a value close to 0) indicates that the model explains none (or little) of the variability of the response data around its mean; 1 (or a value close to 1) indicates that the model explains all (or most) of the variability of the response data around its mean. Moreover, if $R^2 = 1$ the observed data are distributed on the regression line (perfect fitting). Note that $R^2 = r^2$ were, r , denotes the Pearson correlation coefficient that can be computed as

$$r = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \sqrt{n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2}}$$

Ordinary least squares regression is the most common approach to defining the “best” straight line that fit the data. However, there are other regression methods that can be used in place of ordinary least squares, such as the least absolute deviations (i.e., the line that minimizes the sum of the absolute values of the residuals)

$$\operatorname{argmin}_{\beta_0, \beta_1} \sum_{i=1}^n |e_i| = \operatorname{argmin}_{\beta_0, \beta_1} \sum_{i=1}^n |Y_i - \beta_0 - \beta_1 x_i|$$

see (Sen and Srivastava, 1990) for a more detailed discussion.

Multiple Linear Regression

Multiple linear regression generalizes the above mentioned concepts and formulae to the case where more predictors are used (Abramovich and Ritov, 2013; Casella and Berger, 2001; Faraway, 2004; Fahrmeir *et al.*, 2013; Jobson, 1991; Krzywinski and Altman, 2015; Rao, 2002; Sen and Srivastava, 1990).

Let $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)^T$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T$ denote the vectors of the observed outcomes and the noise, respectively,

$$\mathbf{X} = \begin{pmatrix} 1, x_{11}, x_{12}, \dots, x_{1p} \\ 1, x_{21}, x_{22}, \dots, x_{2p} \\ \dots \\ \dots \\ 1, x_{n1}, x_{n2}, \dots, x_{np} \end{pmatrix}$$

is the *design matrix*, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^T$ is the vector of unknown coefficients, then the regression problem can be rewritten in matrix form as,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The least squares estimate of $\boldsymbol{\beta}$ can be obtaining by solving the following problem

$$\operatorname{argmin}_{\beta_0, \beta_1, \dots, \beta_p} \sum_{i=1}^n e_i^2 = \operatorname{argmin}_{\beta_0, \beta_1, \dots, \beta_p} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} \dots - \beta_p x_{ip})^2 = \operatorname{argmin}_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

It can be proved that if $\mathbf{X}^T \mathbf{X}$ is a non-singular $p \times p$ matrix, and $(\mathbf{X}^T \mathbf{X})^{-1}$ denotes its inverse matrix, then there exists a unique solution given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

If $\mathbf{X}^T \mathbf{X}$ is singular, the solution is not unique, but it can be still computed in terms of the pseudo-inverse matrix $(\mathbf{X}^T \mathbf{X})^+$ as follows

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^+ \mathbf{X}^T \mathbf{Y}$$

Once the estimate $\hat{\boldsymbol{\beta}}$ has been computed, it is possible to estimate $\hat{\mathbf{Y}}$ as

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = \mathbf{H}\mathbf{Y}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ is called the *hat-matrix* or *projection matrix*. From a geometrical point of view, in a n -dimensional Euclidean space, $\hat{\mathbf{Y}}$ can be seen as the orthogonal projection of $\mathbf{Y}$ in the subspace generated by the columns of $\mathbf{X}$, and the vector of *residuals*, defined as $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$, is orthogonal to the subspace generated by the columns of $\mathbf{X}$.

Moreover, under the assumption that the Gauss-Markov conditions hold, the least squares estimate, $\hat{\boldsymbol{\beta}}$, is the BLUE, and an unbiased estimate of the variance is given by

$$\hat{\sigma}^2 = \frac{1}{n - p - 1} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Analogously to simple linear regression, the coefficient of determination, $R^2 = \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$, can be used to evaluate the goodness of fit ($R^2 \in [0, 1]$). However, the value of R^2 increases as p increases. Therefore, it is not a useful measure for selecting parsimonious models. The *adjusted* R^2 , R_a^2 , is defined as

$$R_a^2 = 1 - \frac{\sum_{i=1}^n \frac{(Y_i - \hat{Y}_i)^2}{(n-p-1)}}{\sum_{i=1}^n \frac{(Y_i - \bar{Y})^2}{(n-1)}} = 1 - \frac{SSE/(n-p-1)}{SST/(n-1)}$$

which adjusts for the number of explanatory variables in the model.

Overall we have the following partition of the errors: $SST = SSR + SSE$ where $SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$ denotes the total sum of squares, $SSR = \sum_{i=1}^n (\bar{Y}_i - \hat{Y}_i)^2$, the sum of squares explained by the regression, and $SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$, the residual sum of squares due to the error term.

Gauss-Markov Conditions and Diagnostics

Linear regression and the ordinary least squares approach are based on the so-called Gauss-Markov conditions

- 1) $E(\varepsilon_i) = 0$ (*Zero-mean error*)
- 2) $E(\varepsilon_i^2) = \sigma^2 \forall i = 1, \dots, n$ (*Homoscedastic error*)
- 3) $E(\varepsilon_i \varepsilon_j) = 0 \forall i \neq j$ (*Independent errors*)

that guarantee $\hat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y$ is the *best linear unbiased estimator* (BLUE) of the coefficients (Gauss-Markov theorem, Jobson, 1991; Sen and Srivastava, 1990).

When using linear regression functions it is important to verify the validity of such assumptions using so-called residual plots (Altman and Krzywinski, 2016b). Moreover, in order to perform inference on the significance of the regression coefficients and prediction of future outcomes, it is necessary that $\varepsilon \sim N(0, \sigma^2 I_n)$ also holds. This condition can be verified using the Shapiro-Wilk test or the Lilliefors test, see (Razali and Wah, 2011) or the Chapter *Statistical Inference Techniques* in this volume.

Tests of significance and confidence regions

One of the most important aspects of regression is that it is possible to make inference on the estimated model, this means it is possible to test the significance of each coefficients (i.e., $H_0 : \beta_j = 0$ vs $H_1 : \beta_j \neq 0$), assuming the other are fixed, and to build confidence intervals. In particular, under the “white noise” assumptions, it is possible to prove that

$$\hat{\beta} \sim N(\beta, (X^T X)^{-1} \sigma^2)$$

and

$$(n-p-1)\hat{\sigma}^2 \sim \sigma^2 \chi_{n-p-1}^2$$

where χ_{n-p-1}^2 denotes a chi-squared distribution with $n-p-1$ degree of freedom. Moreover the two estimates are statistically independent. Hence, it is possible to show that (under the null hypothesis)

$$\frac{\hat{\beta}_j - \beta_j}{s_{\hat{\beta}_j}} \sim T_{n-p-1}$$

is a Student distribution T_{n-p-1} with $n-p-1$ degree of freedom. These results can be used to decide whether β_j are significant (where, $s_{\hat{\beta}_j}$ denotes the standard deviation of the estimated coefficients, i.e., $s_{\hat{\beta}_j} = \sqrt{\text{var}(\hat{\beta}_j)}$).

Moreover the $100(1-\alpha)\%$ confidence interval for the regression coefficient, $\hat{\beta}_j$, is given by $\hat{\beta}_j \pm s_{\hat{\beta}_j} t_{n-p-1, \frac{\alpha}{2}}$, where $t_{n-p-1, \frac{\alpha}{2}}$ is the corresponding quantile of the T distribution with $n-p-1$ degree of freedom. More details about significance testing, including the case of contrasts between coefficients, can be found in Sen and Srivastava (1990), Jobson (1991).

Inference for the model

The overall *goodness of fit* of the model can be measured testing the null assumption $H_0 : \beta_1 = \beta_2 = \dots = \beta_p = 0$ using the F-test

$$F = \frac{SSR/p}{SSE/(n-p-1)}$$

which has a Fisher distribution with p and $n-p-1$ degree of freedom.

Confidence interval (CI) for the expectation of a predicted value at x_0

Let $x_0 = (x_{00}, x_{01}, \dots, x_{0p})$ be a novel observation of the independent variables X , and let y_0 be the unobserved response. Then, the predicted response is $\hat{y}_0 = x_0^T \hat{\beta}$ and the $100(1-\alpha)\%$ confidence interval of its expected value is

$$\hat{y}_0 \pm t_{n-p-1, \alpha} \hat{\sigma}^2 \left[x_0^T (X^T X)^{-1} x_0 \right]$$

CI for a future observation x_0

Under the same settings, the $100(1-\alpha)\%$ confidence interval for the predicted response is

$$\hat{y}_0 \pm t_{n-p-1, \alpha} \hat{\sigma}^2 \left[1 + \mathbf{x}_0^T (X^T X)^{-1} \mathbf{x}_0 \right]$$

Subset Linear Regression

In many cases, the number of observed variables, X , is large and one seeks for a regression model with a smaller number of “important” variables in order to gain explicative power and to address the so-called variance-bias trade off. In these cases a small subset of the original variables must be selected, and the regression hyper-plane fit using the subset. In principle, one can choose a goodness criterion and compare all potential models using such criteria. However if p denotes the number of observed variables, there will be 2^p potential models to compare. Therefore, for large p such an approach becomes unfeasible. Stepwise linear regression methods allow one to select a subset of variables by adding or removing one variable at a time and re-fitting the model. In this way, it is possible to choose a “good” model by fitting only a limited number of potential models. *Mallows' C_p* , *AIC* (Akaike's information criterion), and *BIC* (Bayesian information criterion) are widely used for choosing “good” models. More details can be found in [Sen and Srivastava \(1990\)](#), [Jobson \(1991\)](#), [Miller \(2002\)](#), [Lever et al., \(2016\)](#).

Regression Analysis in Practice

How to Report the Results of a Linear Regression Analysis and Diagnostic Plots

After fitting a linear regression model one has to report the estimates of the regression coefficients and their significance. However, this is not sufficient for evaluating the overall significance of the fit. The R^2 (or the R_a^2) are typical measures of quality that should be reported, as well as the overall significance, F , of the regression model. Moreover, results have to be accompanied by a series of diagnostic plots and statistics on residuals that can be used to validate the Gauss-Markov conditions. Such analyses include normal-quantile residual and fitted-residual plots. The key aspect to keep in mind is that when fitting a linear regression model by using any software, the result will provide the “best” hyper-plane or the “best” straight line, regardless of the fact that the model is inappropriate. For example, in the case of simple linear regression the best line will be returned even though the data are not linear. Therefore, to avoid false conclusions, all of the above-mentioned points have to be considered. The leverage residual, DFFITS and Cook's distance of observations are other measures that can be used to assess the quality and robustness of the fit, see ([Sen and Srivastava, 1990](#); [Jobson, 1991](#); [Altman and Krzywinski, 2016b](#)) for more details. In cases where the assumptions are violated, one can use data transformations to mitigate the deviation from the assumptions, or use more sophisticated regression models, such as generalized linear models, non-linear models, and non-parametric regression approaches depending on the type of assumptions one can place on the data.

Outliers and Influential Observations

The presence of outliers can be connected to potential problems that may inflate the apparent validity of a regression model ([Sen and Srivastava, 1990](#); [Jobson, 1991](#); [Altman and Krzywinski, 2016a](#)). In particular, an outlier is a point that does not follow the general trend of the rest of the data. Hence it can show either an extreme value for the response Y , or for any of the predictors X_k , or both. Its presence may be due either to measurement error, or to the presence of a sub-population that does not follow the expected distribution. One might observe one or few outliers when fitting a regression model. An influential point is an outlier that strongly affects the estimates of the regression coefficients. Anscombe's quartet ([Anscombe, 1973](#)) is a typical example used to illustrate how influential points can inflate conclusions. Roughly speaking, to measure the influence of an outlier, one can compute the regression coefficients with and without the outlier. More formal approaches for detecting outliers and influential points are described in [Sen and Srivastava \(1990\)](#), [Jobson \(1991\)](#), [Altman and Krzywinski \(2016a\)](#).

Transformations

As already mentioned above, when the Gauss-Markov conditions are not satisfied ordinary least squares cannot be used. However, depending on deviation from the assumptions, either a transformation can be applied in order to match the assumptions, or other regression approaches can be used. More in general, transformations can be used on the data for wider purposes, for example for centering and standardizing observations when the variables have different magnitude. Other widely used transformations are logarithmic and square-root transforms that can help in linearizing relationships. Additionally, in the context of linear regression, variance-stabilizing transformations can be used to accommodate heteroscedasticity, and normalizing transformations to better match the assumption of normal distribution. Although several transformations have been proposed in the literature, there is no standard approach. Moreover, an intensive use of transformations might be questionable. The reader is referred to ([Sen and Srivastava, 1990](#)) for more details.

Beyond ordinary least squares approaches

As previously stated, linear regression is also known as the regression of the “mean” since the conditional mean of Y given X is modeled as linear combination of the regressors, X , plus an error term. In this case, the least squares approach constitutes the BLUE when the Gauss-Markov conditions hold. However, other types of criteria, such as *quantile regression*, have been proposed (McKean, 2004; Koenker, 2005; Maronna et al., 2006). In particular, quantile regression aims at estimating either the conditional median or other conditional quantiles of the response variable. In this case, the estimates are obtained using linear programming optimization algorithms that solve the corresponding minimization problem. Such types of regression models are more robust than least squares with respect to the presence of outliers.

Advanced Approaches

High Dimensional Regression ($p \gg n$)

Classical linear regression, as described above, requires the matrix $(X^T X)$ to be invertible, this implies that $n \geq p$, where n denotes the number of independent observations (i.e., the sample size) and p the number of variables. Current advance in science and technology allow the measurement of thousands or millions of variables on the same statistical unit. Therefore, one has often to deal with the case of $p \gg n$, where ordinary least squares cannot be applied, and other types of approaches must be used. The solution in such cases is to use penalized approaches such as Ridge regression, Lasso regression, or Elastic net or others (Hastie et al., 2009; James et al., 2013; Tibshirani, 1996, 2011; Zou and Hastie, 2005), where a penalty term, $P(\beta)$, is added to the fitting criteria, as follows

$$(Y - X\beta)^T(Y - X\beta) + \lambda P(\beta)$$

where λ is the so-called *regularization parameter*, which controls the trade-off between bias and variance. In practice, by sacrificing the unbiased property of the ordinary least squares, one can achieve generalization and flexibility. Different penalty terms lead to different regressions methods (Bühlmann and van de Geer, 2011; Hastie et al., 2015) and are also known as *regularization techniques*. The regularization parameter, λ , is usually selected using criteria such as cross-validation.

Other possible approaches include the use of dimension reduction techniques such as principal component analysis or feature selection, such as described in the Chapter *Dimension Reduction Techniques* in this volume.

Ridge regression

The estimates of the regression coefficients are obtained by solving the following minimization problem

$$\hat{\beta}^{\text{Ridge}} = \underset{\beta}{\operatorname{argmin}} \left[\sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} \dots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p \beta_j^2 \right] = \underset{\beta}{\operatorname{argmin}} [(Y - X\beta)^T(Y - X\beta) + \lambda \|\beta\|_2^2]$$

where $\lambda > 0$ is a suitable regularization parameter. It is easy to show, since the minimization problem is convex, it has a unique solution. The solution can be computed in a closed form as

$$\hat{\beta}^{\text{Ridge}} = (X^T X + \lambda I)^{-1} X^T Y$$

which corresponds to shrinking the ordinary least square coefficients by an amount that is controlled by λ . Ridge regression can be used when $(X^T X)$ is singular or quasi-singular and ordinary least squares does not provide a unique solution. In fact, in such circumstance $(X^T X + \lambda I)$ is still invertible. Note that, as $\lambda \rightarrow 0$ the penalty plays a “minor” role, thus $\hat{\beta}^{\text{Ridge}}$ tends to $\hat{\beta}$ (i.e., those coefficients obtained by using ordinary least squares), when $\lambda \rightarrow +\infty$ the $\hat{\beta}^{\text{Ridge}}$ tends to zero (i.e., to the so-called intercept-only model). Different values of λ provide a trade-off between bias and variance (larger λ increases the bias, but reduces the variance). A suitable value of λ can be chosen from the data by cross-validation. Although ridge regression can be used for fitting high dimensional data in a more effective way than using ordinary least squares approach, there are however some limitations such as a large bias toward zero for large regression coefficients, and a lack of interpretability of the regression solution since “unimportant” coefficients are shrunk towards zero, but they’re still in the model instead of being killed to zero. As a consequence, the ridge regression does not act as model selection.

Lasso regression

The estimates of the regression coefficients are obtained by solving the following minimization problem

$$\hat{\beta}^{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \left[\sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} \dots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p |\beta_j| \right] = \underset{\beta}{\operatorname{argmin}} [(Y - X\beta)^T(Y - X\beta) + \lambda \|\beta\|_1]$$

where $\lambda > 0$ is a suitable regularization parameter. The underlying idea in lasso approach is that it seeks a set of sparse solutions meaning that it will set some regression coefficients exactly equal to 0. As a consequence, lasso also performs model selection. Larger is the value of λ , more will be the coefficients set to zero. Unfortunately, the solution of lasso minimization problem is not available in closed form, however it can be obtained by using convex minimization approaches and several algorithms have been

proposed such as the least-angle regression (LARS) (Efron *et al.*, 2004) to efficiently fit the model. Analogously to ridge regression, a suitable value of λ can be chosen from the data by cross-validation.

Overall, lasso has opened a new framework in the so-called *high-dimensional regression models* and several generalizations have been proposed (Bühlmann and van de Geer, 2011; Hastie *et al.*, 2015) to overcome its limitations and to extend the original idea to different regression contexts.

Elastic net regression

The estimates of the regression coefficients are obtained by solving the following minimization problem

$$\begin{aligned}\hat{\beta}^{\text{Elastic net}} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n \left(Y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} \dots - \beta_p x_{ip} \right)^2 + \lambda_1 \sum_{j=1}^p \beta_j^2 + \lambda_2 \sum_{j=1}^p |\beta_j| \\ &= \underset{\beta}{\operatorname{argmin}} \left[(Y - X\beta)^T (Y - X\beta) + \lambda_1 \|\beta\|_2^2 + \lambda_2 \|\beta\|_1 \right]\end{aligned}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are suitable regularization parameters (Zou and Hastie, 2005). Note that for $\lambda_1 = 0$ one can obtain lasso regression, while the case $\lambda_2 = 0$ corresponds to ridge regression. Different combinations of λ_1 and λ_2 compromise between shrinking and selecting coefficients. Indeed, elastic net was designed to overcome some of the limitations of both ridge regression and lasso regression, since the quadratic penalty term shrinks the regression coefficients toward zero, while the absolute penalty term act as model selection by keeping or killing regression coefficients. For example, elastic net is more robust than lasso when correlated predictors are present in the model. In fact, when there is a group of highly correlated variables, lasso usually selects only one variable from the group (ignoring the others), elastic net allows more variables to be selected. Moreover, when $p > n$ lasso can select at most n variables before saturating, whereas, thanks to the quadratic penalty, elastic net can select a larger number of variables in the model. On the other hand, “unimportant” regression coefficients are often set to zero to perform variable selection.

Other Types of Regressions

Generalized linear models (GLM) are a well-known generalization of the above-described linear model. GLM allow the dependent variable, Y , to be generated by any distribution $f()$ belonging to the exponential family. The exponential family includes normal, binomial, Poisson, and gamma distribution among many others. Therefore GLM constitute a general framework in which to handle different type of relationships.

The model assumes that the mean of Y depends on X by means of a link function, $g()$,

$$E(Y) = \mu = g^{-1}(X\beta)$$

where $E()$ denotes expectation and $g()$ the *link function* (an invertible, continuous and differentiable function). In practice, $g(\mu) = X\beta$, so there is a linear relationship between X and a function of the mean of Y . Moreover, in this context the variance is also function of the mean $\operatorname{Var}(Y) = V(\mu) = V(g^{-1}(X\beta))$.

In this context, the linear regression framework can be reformulated choosing the identity as the link function, Poisson regression corresponds to $\operatorname{Log}()$ as the link function, and binomial regression to choosing Logit , and so on.

The unknown regression coefficients β are typically estimated with maximum likelihood, maximum quasi-likelihood, or Bayesian techniques. Inference can then be carried out in a similar way as for linear regression. A formal description and mathematical treatment of GLM can be found in McCullagh and Nelder (1989), Madsen and Thyregod (2011). GLM are very important for biomedical applications since they include logistic and Poisson regression, which are often used in biomedical science to model binary outcomes or counts data, respectively.

Recently, penalized regression approaches, as those described in section high dimensional regression, have been extended to the generalized linear models. In this context, the *regularization* is achieved by penalizing the log-likelihood function; see (Bühlmann and van de Geer, 2011; Hastie *et al.*, 2015) for more details.

Closing Remarks

Regression constitutes one of the most relevant frameworks of modern statistical inference with many applications to the analysis of biomedical data, since it allows one to study the relationship between a dependent variable Y and a series of p independent variables $X = [X_1 | \dots | X_p]$, from a set of independent observations (x_i, Y_i) , $i = 1, \dots, n$. When the relationship is linear in the coefficients one has linear regression. Despite its simplicity, linear regression allows testing the significance of the coefficients, estimating the uncertainty and predicting novel outcomes. The Gauss-Markov conditions guarantee that the least squares estimator has the BLUE property, and the normality of the residuals allows carrying out inference. When $p > n$, classical linear regression cannot be applied, and penalized approaches have to be used. Ridge, lasso and elastic net regression are the most well known approaches in the context of high dimensional data analysis (Bühlmann and van de Geer, 2011; Hastie *et al.*, 2015). When the

white-noise conditions are violated data transformation of other types of models have to be considered. Analogously, when the relationship is not linear other approaches or non-parametric models should be used.

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Abramovich, F., Ritov, Y., 2013. *Statistical Theory: A Concise Introduction*. Chapman & Hall/CRC.
- Altman, N., Krzywinski, M., 2015. Simple linear regression. *Nature Methods* 12 (11), 999–1000.
- Altman, N., Krzywinski, M., 2016a. Analyzing outliers: Influential or nuisance? *Nature Methods* 13 (4), 281–282.
- Altman, N., Krzywinski, M., 2016b. Regression diagnostics. *Nature Methods* 13 (5), 385–386.
- Anscombe, F.J., 1973. Graphs in statistical analysis. *American Statistician* 27 (1), 17–21.
- Bühlmann, P., van de Geer, S., 2011. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer series in statistics.
- Casella, G., Berger, R., 2001. *Statistical Inference*, second ed. Duxbury.
- Efron, B., Hastie, T., Johnstone, J., Tibshirani, R., 2004. Least angle regression. *Annals of Statistics* 32 (2), 407–499.
- Faraway, J.J., 2004. *Linear Models with R*. Chapman & Hall/CRC.
- Fahrmeir, L., Kneib, T., Lang, S., Marx, B., 2013. *Regression: Models, Methods and Applications*. Springer.
- Hastie, T., Tibshirani, R., Friedman, J., 2009. *The Elements of Statistical Learning*, second ed. Springer series in Statistics.
- Hastie, T., Tibshirani, R., Wainwright, M., 2015. *Statistical Learning with sparsity: The Lasso and Generalizations*. CRC Press.
- James, G., Witten, D., Hastie, T., Tibshirani, T., 2013. *An Introduction to Statistical Learning: With Applications in R*. Springer.
- Jobson, J.D., 1991. *Applied Multivariate Data Analysis. Volume I: Regression and Experimental Design*. Springer Texts in Statistics.
- Koenker, R., 2005. *Quantile Regression*. Cambridge University Press.
- Krzywinski, M., Altman, N., 2015. Multiple linear regression. *Nature Methods* 12 (12), 1103–1104.
- Lever, J., Krzywinski, M., Altman, N., 2016. Model selection and overfitting. *Nature Methods* 13 (9), 703–704.
- Madsen, H., Thyregod, P., 2011. *Introduction to General and Generalized Linear Models*. Chapman & Hall/CRC.
- Maronna, R., Martin, D., Yohai, V., 2006. *Robust Statistics: Theory and Methods*. Wiley.
- McCullagh, P., Nelder, J., 1989. *Generalized linear models*, second ed. Boca Raton, FL: Chapman and Hall/CRC.
- McKean, Joseph W., 2004. Robust analysis of linear models. *Statistical Science*. 19 (4), 562–570.
- Miller, A., 2002. *Subset Selection in Regression*. Chapman and Hall/CRC.
- Rao, C.R., 2002. *Linear Statistical Inference and its Applications*, Wiley series in probability and statistics, second ed. New York: Wiley.
- Razali, N.M., Wah, Y.B., 2011. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics* 2 (1), 21–33.
- Sen, A., Srivastava, M., 1990. *Regression Analysis: Theory, Methods and Applications*. Springer Texts in Statistics.
- Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal statistical Society B* 58 (1), 267–288.
- Tibshirani, R., 2011. Regression shrinkage and selection via the lasso: A retrospective 73 (3), 273–282.
- Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B* 67 (2), 301–320.

Nonlinear Regression Models

Audrone Jakaitiene, Vilnius university, Vilnius, Lithuania

© 2019 Elsevier Inc. All rights reserved.

Introduction

Regression was, and still is, a very common tool used by researchers in various fields to express a relationship between one or more explanatory variables (also called predictors or covariates) and a response variable (or outcome). The most frequently employed regression models are linear. However, a linear regression model cannot handle discrete responses, such as categorical outcomes or counts. Sometimes the data itself does not follow a linear pattern. This motivates the use of nonlinear models. One option is to use generalized linear model (GLM), which can represent categorical, binary or other types of responses. Another option is to set up nonparametric regression models, or to perform curve fitting. Therefore, in this article, we will focus on nonlinear regression models, such as logistic, Poisson and negative binomial regressions, with some insights to more generalized nonlinear models.

GLM provide a statistical framework for diverse research problems in various fields, such as medicine, biology, bioinformatics and others. Background information about the most widely used GLMs, such as binary logistic, and Poisson regression can be found in almost any statistical textbook (e.g., Bowers, 2008; Kirkwood and Sterne, 2010; Motulsky, 2014). More advanced information about GLMs can be obtained from recent books in which various modelling peculiarities are addressed, and implementation discussed (e.g., Agresti, 2015; Faraway, 2016; Hosmer *et al.*, 2013). However nowadays, in the era of big data, GLM faces new challenges and still is a significant research topic. A few challenges come from genetic data analysis. Using new DNA sequencing technologies, it is feasible, in terms of cost and time, to sequence all the exomes or complete genomes of large numbers of people. However, as for now, the number of sequenced individuals is much smaller as the number of potential covariates (each individual may have millions of genetic variants, which each may be rare in the population), leading to high dimension and low sample size problems that might not be handled with classical models (Dasgupta *et al.*, 2011). For large p and small n microarray data, Liao and Chin (2007) discuss covariates selection and accurate error estimation issues for logistic regression models. For rare events in high dimensional data, sparse logistic regression is applicable as well (Liu *et al.*, 2009). Qiu *et al.* (2013) discuss bias issues in logistic regression and propose methods for bias adjustment for rare events in large scale data. New methodologies for Poisson and Negative binomial regression have been developed in light of big data and/or rare events, as well (Papastamoulis *et al.*, 2016; Anders and Huber, 2010).

We start with standard notation for linear models. Then we introduce GLM for nominal and count variables, with some insights to other nonlinear modelling technics. The implementation of GLM in R is presented. The main assessment aspects of the models, and the regularization methods are described. Discussion and future directions finalize the article.

Fundamentals

General Linear Model

In a general linear model (LM), we model the response variable, Y , as a linear function of the explanatory or predictor variables, X_k , where $k=1, \dots, K$ and ε is an error term

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_K X_K + \varepsilon \quad (1)$$

where β_k are regression parameters, estimated using ordinary least squares. Y and each explanatory variable, X_k , as well as the error term have n observations. The model is *linear in the parameters*. We assume that the errors, ε , are independent and identically distributed such that $E(\varepsilon)=0$, $Var(\varepsilon)=\sigma^2$ and $\varepsilon \sim N(0, \sigma^2)$.

Although linear models are a useful approach, there are some cases when general linear models are inappropriate, namely when possible Y values are restricted (such as categorical or count data), or the variance of Y depends on the mean. Generalized linear models extend the linear model approach to address these issues.

Generalized Linear Models

The purpose of generalized linear model (GLM), as for LMs, is to identify a relationship between a number of covariates and response variables. Following Agresti (2015), a GLM is composed of

- A *random component*, which specifies the response variable, Y , and its probability distribution from an *exponential family* (the most commonly used are Normal, Binomial, and Poisson). The observations of Y are treated as independent.
- A *linear predictor* for k explanatory variables

$$\eta = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k \quad (2)$$

The linear predictor is linear in the parameters. The explanatory variables can be nonlinear functions of underlying variables, such as interaction term (e.g., $X_4 = X_1 X_3$). In GLM, Y is treated as random, and explanatory variables as fixed, therefore the linear predictor is sometimes called the *systematic component*.

- A smooth and invertible linearizing *link function* $g(\cdot)$, which describes how the mean of the response variable, $E(Y) = \mu$, depends on the linear predictor, $g(\mu) = \eta$.
- A *variance function* $V(\cdot)$, which describes how the variance of the response variable, $Var(Y)$, depends on the mean, i.e., $Var(Y) = \phi V(\mu)$. ϕ is a constant dispersion parameter.

A GLM with a link function $g(\mu) = \mu$, known as the *identity link function*, is called a linear model (Agresti, 2015). Therefore, LM is a special case of GLM.

Next we introduce the two most important GLM for nonlinear modelling of discrete response variables, such as categorical and count data.

Modelling categorical data

The *logistic regression* (LR) model is a common approach for modelling a categorical outcome variable. Categorical variables could be binary decision variables (for example, Yes/No, Present/Absent, Healthy/Disease), or variables with more than two levels (for example, cause of a death – cancer, heart disease, other). If response variable is binary, one will set up a univariable (one explanatory variable) or multivariable (more than one covariate) *binary* (also called binomial) *logistic regression* (BLR). If response variable has more than two levels, one would conduct univariable or multivariable *multinomial logistic regression* (MLR). Nowadays, LR is a classical approach for building prediction or classification methods, and has been widely used in many applications, such as bioinformatics (Tsuruoka et al., 2007), gene classification (Liao and Chin, 2007), genetic association (Cantor et al., 2010; Jostins and McVean, 2016), and neural signal processing (Philiastides et al., 2006).

Binomial logistic regression (BLR). Suppose a response variable, Y , takes the values zero or one and $P(Y=1) = p$. For example, presence (value 1) or absence (0) of a species in a site. We need a model that relates k explanatory variables and the probability, p . As described above, a linear predictor has the form shown in equation (2). We cannot use $\eta = p$ as a link function, as it does not guarantee $0 \leq p \leq 1$. The most popular choice is the *logit* link function

$$\eta = g(\mu) = \text{logit}(\mu) = \log(p/(1-p)) \quad (3)$$

and

$$p = \frac{e^\eta}{1 + e^\eta}$$

Thus, the combination of a linear predictor with the logit link function gives BLR (Faraway, 2016). One obtains the logarithm of the odds of a positive outcome, and straightforward algebraic manipulation transforms this into the probability of the outcome. We use the method of maximum likelihood to obtain estimates of the parameters.

Multinomial logistic regression (or *multinomial regression*, MLR) is an extension of BLR to nominal outcome variables with more than two levels. Following Faraway (2016), suppose random variable Y can have values of a finite number of categories, labeled $1, 2, \dots, J$. Let $p_j = P(Y=j)$ and $\sum_{j=1}^J p_j = 1$. In MLR, as in BLR, we want to set up a model that links the probabilities, p_j , with the explanatory variables, X_k . Similarly

$$\eta_j = \beta_{0j} + \beta_{1j}X_1 + \dots + \beta_{Kj}X_K = \log \frac{p_j}{p_c}, j = 1, \dots, J-1$$

where we use category J as a reference and

$$p_j = \frac{e^{\eta_j}}{1 + \sum_{j=1}^{J-1} e^{\eta_j}}$$

$$p_J = 1 - \sum_{j=1}^{J-1} p_j$$

It makes no difference which category we choose as the reference, although there may be choices that are more convenient for interpretation. The MLR model estimates a separate binary logistic regression model for each category variable. The result is $J-1$ binary logistic regression models. Each one specifies the effect of the predictors on the probability of success in that category, in comparison to the reference category. As for BLR, we use the method of maximum likelihood (ML) to obtain estimates of the parameters.

Modelling count data

For count data, for example, the number of emerging seedlings in a plot, the number of ticks on red grouse chicks, or clutch sizes of storks (Bolker et al., 2009), the response variable is discrete, positive and, typically, not bounded. If the count is bounded, logistic regression could be considered or, if the count is sufficiently large, linear regression might be used.

Variables that denote the number of occurrences of an event, or object in a certain unit of time or space, are distributed according to the Poisson distribution (Motulsky, 2014). We assume that the events occur randomly, independently of one another, and with an average rate that does not change over time. Therefore, if $Y \sim \text{Pois}(\mu)$, we might set up a *Poisson regression*

model in which the linear predictor is linked via the log link function, that is:

$$\log(\mu) = \eta = \beta_0 + \beta_1 X_1 + \dots + \beta_K X_K$$

For a Poisson distribution, the variance must be equal the mean, i.e., $E(Y) = \text{Var}(Y) = \mu$. However, count data are often highly variable, due to heterogeneity, and exhibit variance that is larger than mean; this is called *overdispersion*.

The *Negative binomial distribution* handles overdispersion by using an extra parameter to account for it. Typically the mean response μ is linked to the linear predictor as follows:

$$\eta = \beta_0 + \beta_1 X_1 + \dots + \beta_K X_K = \log \frac{\mu}{\mu + \phi}$$

ϕ is fixed, and is additional parameter that must be estimated.

Some datasets show greater frequencies of zero counts than the Poisson or Negative binomial models allow. For this case, zero-inflated Poisson or Negative binomial regressions can be employed (see [Agresti, 2015](#)).

Other Nonlinear Techniques

As described above, LM is special case of GLM. A GLM is special case of a *generalized additive model* (GAM), where the linear predictor is replaced by smooth functions of the explanatory variables

$$g(\mu) = \sum_{k=1}^K s_k(X_k)$$

where $s_k(\cdot)$ is an unspecified smooth function of predictor k . So, when s_k is a linear function, GAM becomes GLM.

More generally, we have *nonlinear regression* when the response mean is a nonlinear function of the parameters

$$E(Y) = f(X; \beta)$$

where f is a known function of the explanatory variables and parameters. For example, we can consider a model for tumor or population growth ([Agresti, 2015](#))

$$E(Y) = \frac{\beta_0}{1 + e^{-(\beta_1 + \beta_2 X)}}$$

Up to now f linearly or nonlinearly depends on parameters, which we call a parametric approach. Parameters have an empirical interpretation. In a nonparametric approach, we can select f from some smooth family of functions. All we have to assume are the degree of smoothness and continuity of f ([Faraway, 2016](#)). There is no formulaic way of describing the relationship between the covariates and the response. The nonparametric approach might be a relevant tool for modelling unknown data, when we have little idea about an appropriate form of the model. Nonparametric regression estimators, also known as *smoothers*, could be kernel or spline methods. Both of them suffer from outlier issues in the data. [Cleveland \(1979\)](#) proposed the algorithm *lowess* or *loess* (locally weighted scatter plot smoothing) as an outlier resistant method based on local polynomial fits. For the implementation of the latter algorithm the function *loess()* has been implemented in R.

However, if one has proper information about the appropriate model family, a parametric model should be considered.

GLM Implementation in R

Linear and GLM analysis can be conducted using R, SAS, Stata, SPSS and other general purpose statistical software. We discuss only R, and give links to examples. However, the choice of the software is less essential when one understands and can interpret the results, as the outputs are quite similar irrespective of the software selected.

In R, a standard GLM can be fitted using the *glm* function from the package *stats*, which is similar to the *lm* function for fitting linear models.

The syntax of the *glm* function is

$$\text{glm}(\text{formula}, \text{family}, \text{data}, \text{weights}, \text{subset}, \dots)$$

We will give some details only on the *formula*, *family* and *data* arguments as the remaining list of arguments and their usage can be found on the *glm* function webpage.

The argument *formula* should be specified as follows

$$Y \sim X_1 + X_2$$

where Y is the response variable, coded 0/1 for BLR, and X_1 and X_2 are the explanatory variables, which could be continuous or categorical variables. The latter should be coded as a dummy variable with values 0/1, as well. If not encoded by a dummy variable, by default, the first level alphabetically will be associated with $Y=0$, and the second with $Y=1$. One should be careful with categorical data that have more than 2 levels ([Bowers, 2008](#); [Suits, 1957](#)) (Additional reading about how to code dummy variables in R: see Relevant Website section). All specified variables must be in the workspace or in the data frame passed in the *data* argument.

For the *family* argument, one has various choices of distributions of response variables, together with options for a link function. Normal, binomial and Poisson regression would use the following *family* specifications

```
gaussian(link="identity")
binomial(link="logit")
poisson(link="log")
```

The implementation of a binary logistic regression (BLR) model would be as follows

```
Mod_BLR <- glm(Response ~ Expl_1 + Expl_2, data=inputData, family=binomial(link="logit"))
summary(Mod_BLR)
predicted <- predict(Mod_BLR, newData, type="response")
```

The *glm* function will build the logistic regression model, *Mod_BLR*, based on the given formula. When we use the *predict* function on this model, it will predict the *log(odds)* of the response variable *Y*.

We cannot implement a multinomial logistic model (MLM) using the *glm* function. Instead, we need to employ the *multinom* function from the *nnet* package

```
library(nnet)
Mod_MLM <- multinom(Response ~ Expl_1 + Expl_2, data=inputData)
summary(Mod_MLM)
```

Implementation of standard Poisson regression is possible using the *glm* function, e.g.,

```
Mod_Pois <- glm(Response ~ Expl_1 + Expl_2, family="poisson", data=inputData)
summary(Mod_Pois)
predict(Mod_Pois, newData, type="response")
```

To implement a negative binomial model (NBM), we need the *glm.nb* function from the *MASS* package, for which we need to provide only the formula and data frame

```
library(MASS)
Mod_NBM <- glm.nb(Response ~ Expl_1 + Expl_2, data=inputData)
summary(Mod_NBM)
predict(Mod_NBM, newData, type="response")
```

The implementation of zero-inflated and zero-truncated models, with examples in R, for example, can be found in the webpages of The Institute for Digital Research and Education, UCLA (see Relevant Websites).

For the implementation of nonlinear regression models in R, one can consider using the *nlstools* package (Baty *et al.*, 2015).

Fitting Regression Models

The following steps are necessary when fitting regression models:

1. *Model selection/specification.* When one considers multivariable analysis, the selection of variables (or features) is essential. In general, there are two approaches to feature selection: *forward* and *backward*. In the forward approach, we start with a single explanatory variable in a model, and extend it by including additional variables. We retain only the variables whose p-values are less than confidence level in the model. In the backward approach, we start with a model including all explanatory variables. We sequentially eliminate the variables with the largest p-values (those greater than confidence level). Inclusion or exclusion of variables is performed sequentially. If there are no further variables to include or exclude, the model selection process is terminated. For model selection, the Akaike Information Criteria (AIC) or Bayesian information criterion (BIC) are often used. AIC penalizes a model for having many parameters. A model with better fit has smaller AIC/BIC (for more details, see Symonds and Moussalli, 2011). BIC penalizes a model more severely for the number of parameters compared to AIC. Detailed discussion about the use of AIC vs BIC is provided in Aho *et al.* (2014). To obtain the p-values of the parameters, formula and AIC in R, one can use the *summary()* function for the model. The BIC value can be retrieved using the function *BIC()* for the model.
2. *Model estimation.* Parameter estimation of the selected model will be performed by the selected function. Parameter estimates will be reported in the output of the *summary()* function. For the parameter information alone, one can apply *coef()* function.
3. *Adequacy.* One should check how well the estimated model fits the data. In R there are several functions that can be used, such as *residuals()*, to extract information about residuals of the estimated model; *fitted()* returns the fitted values of the response, using the estimated model, and *predict()* returns the predicted values of the response, using the estimated model. The default predictions are provided on the logit scale (i.e., predictions are made in terms of log odds), while using *type="response"* gives the predicted probabilities.

4. *Inference*. As the last step, we can calculate confidence intervals, perform hypothesis testing, and interpret the results. We calculate confidence intervals for the estimated parameters using the `confint()` function. For two model differences one can use `anova()` function, with the additional optional `test="Chisq"`. A likelihood ratio test comparing candidate models can be performed using the `lrtest()` function from the package `lmtest`.

More information about reporting and selection of LR is provided in Bagley *et al.* (2001) and Bursac *et al.* (2008), respectively. R has a separate package `glmulti` for automated GLM feature selection and multi-model inference. It handles high dimensional data and parallel version is available (Calcagno and de Mazancourt, 2010).

Regularization Methods

In fitting GLM, typically, ML is applied, however sometimes *regularization methods* are used to modify ML in order to avoid interpretability and overfitting issues. ML cannot distinguish covariates with little or no influence, and the variance of parameters becomes infinite for large p and small n problems. Of course, we can think about smaller subsets of the predictors, or apply some method of dimension reduction. Other way, keeping all covariates in the analysis, is to add a term when maximizing the log-likelihood, $L(\beta)$, of the model, which smooths the ordinary estimates. Thus we maximize

$$L^*(\beta) = L(\beta) - s(\beta)$$

where $s(\cdot)$ is a function, defined such that $s(\beta)$ decreases when the elements of β are smoother, in some sense, such as uniformly closer to 0. This method is called the *penalized likelihood* (Agresti, 2015).

A variety of penalized-likelihood methods use L_q -norm smoothing function

$$s(\beta) = \lambda \sum_{k=0}^K |\beta_k|^q$$

for $q \geq 0$ and $\lambda \geq 0$. The response and explanatory variables should be standardized.

Regularization methods that use a quadratic penalty term, $q=2$, are called L_2 -norm penalty methods. In the case of LM, this is known as ridge regression. Fitting a model with the L_1 -norm penalty, $q=1$, it is called the *lasso* (least absolute shrinkage and selection operator) method. For ridge regression, lasso, and other regularization methods, the `glmnet` package can be used in R.

Discussion and Future Directions

Some data has a grouped, nested or hierarchical structure. Repeated measures, longitudinal, and multilevel data, consist of several observations taken on the same individual or group. This may induce a correlation structure in the error terms, ϵ . Also random effects may not be negligible. Therefore, for such data, a mixed effect modelling strategy might be considered (Bolker *et al.*, 2009; McCulloch and Neuhaus, 2001).

The implementation of GLM might result overfitting for high-dimensional data, which might come not only from genetic but also biomedical imaging, signal processing, image analysis, language recognition, and financial data. Regularization methods are especially useful when the number of model parameters is extremely large (Wu *et al.*, 2009; Meier *et al.*, 2008; Roth and Fischer, 2008; Friedman *et al.*, 2010; Ogutu *et al.*, 2012). Tran *et al.* (2015) point out that for large p and huge n , ordinary ML fitting may not be possible, and new methods might be needed. According to Agresti (2015), variable selection methods for large p fall into two types: dimension-reduction methods (stepwise methods, penalized regularization), and identification of relevant effects using significance tests, with adjustment for multiplicity (such as the false discovery rate (FDR)). As another dimension reduction method, one might consider principal component analysis.

As another approach to modelling, one can use Bayesian approach (Jostins and McVean, 2016). However, when dealing with large p , Bayesian inference is perhaps even more challenging than frequentist inference (Agresti, 2015).

Closing Remarks

Nonlinear regression models are a very broad topic. The most frequently used approaches have been concisely discussed, above. Some insights about which methods are appropriate for high-dimensional data, specifically when sample size is small and the parameter number is large, have been provided. All of the described methods are widely used, in many fields, such as biology, medicine, epidemiology, image analysis, language detection, finance. We expect that, due to rapid technological development, nonlinear models will play an even larger role in both theoretical and applied research in the future.

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Agresti, A., 2015. Foundations of Linear and Generalized Linear Models. John Wiley & Sons.
- Aho, K., Derryberry, D., Peterson, T., 2014. Model selection for ecologists: The worldviews of AIC and BIC. *Ecology* 95 (3), 631–636.
- Anders, S., Huber, W., 2010. Differential expression analysis for sequence count data. *Genome Biology* 11 (10), R106.
- Bagley, S.C., White, H., Golomb, B.A., 2001. Logistic regression in the medical literature: Standards for use and reporting, with particular attention to one medical domain. *Journal of Clinical Epidemiology* 54 (10), 979–985.
- Baty, F., Ritz, C., Charles, S., *et al.*, 2015. A toolbox for nonlinear regression in R: The package nlstools. *Journal of Statistical Software* 66 (5), 1–21.
- Bolker, B.M., Brooks, M.E., Clark, C.J., *et al.*, 2009. Generalized linear mixed models: A practical guide for ecology and evolution. *Trends in Ecology & Evolution* 24 (3), 127–135.
- Bowers, D., 2008. Medical Statistics From Scratch: An Introduction for Health Professionals. John Wiley & Sons.
- Bursac, Z., Gauss, C.H., Williams, D.K., Hosmer, D.W., 2008. Purposeful selection of variables in logistic regression. *Source Code for Biology and Medicine* 3 (1), 17.
- Calcagno, V., de Mazancourt, C., 2010. glmulti: An R package for easy automated model selection with (generalized) linear models. *Journal of Statistical Software* 34 (12), 1–29.
- Cantor, R.M., Lange, K., Sinsheimer, J.S., 2010. Prioritizing GWAS results: A review of statistical methods and recommendations for their application. *The American Journal of Human Genetics* 86 (1), 6–22.
- Cleveland, W.S., 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* 74 (368), 829–836.
- Dasgupta, A., Sun, Y.V., König, I.R., Bailey-Wilson, J.E., Malley, J.D., 2011. Brief review of regression-based and machine learning methods in genetic epidemiology: The Genetic Analysis Workshop 17 experience. *Genetic Epidemiology* 35 (S1).
- Faraway, J.J., 2016. Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models, 124. CRC press.
- Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33 (1), 1.
- Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X., 2013. Applied Logistic Regression, 398. John Wiley & Sons.
- Jostins, L., McVean, G., 2016. Trinculo: Bayesian and frequentist multinomial logistic regression for genome-wide association studies of multi-category phenotypes. *Bioinformatics* 32 (12), 1898–1900.
- Kirkwood, B.R., Sterne, J.A., 2010. Essential medical statistics. John Wiley & Sons.
- Liao, J.G., Chin, K.V., 2007. Logistic regression for disease classification using microarray data: Model selection in a large p and small n case. *Bioinformatics* 23 (15), 1945–1951.
- Liu, J., Chen, J., Ye, J., 2009. Large-scale sparse logistic regression. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 547–556. ACM.
- McCulloch, C.E., Neuhaus, J.M., 2001. Generalized Linear Mixed Models. John Wiley & Sons, Ltd.
- Meier, L., Van De Geer, S., Bühlmann, P., 2008. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70 (1), 53–71.
- Motulsky, H., 2014. Intuitive Biostatistics: A Nonmathematical Guide to Statistical Thinking. USA: Oxford University Press.
- Ogutu, J.O., Schulz-Streeck, T., Piepho, H.P., 2012. Genomic selection using regularized linear regression models: Ridge regression, lasso, elastic net and their extensions. *BMC proceedings* 6 (2), S10. (BioMed Central).
- Papastamoulis, P., Martin-Magniette, M.L., Maugis-Rabusseau, C., 2016. On the estimation of mixtures of Poisson regression models with large number of components. *Computational Statistics & Data Analysis* 93, 97–106.
- Philastides, M.G., Ratcliff, R., Sajda, P., 2006. Neural representation of task difficulty and decision making during perceptual categorization: A timing diagram. *Journal of Neuroscience* 26 (35), 8965–8975.
- Qiu, Z., Li, H., Su, H., Ou, G., Wang, T., 2013. Logistic regression bias correction for large scale data with rare events. In: International Conference on Advanced Data Mining and Applications, pp. 133–144. Berlin, Heidelberg: Springer.
- Roth, V., Fischer, B., 2008. The group-lasso for generalized linear models: Uniqueness of solutions and efficient algorithms. In: Proceedings of the 25th International Conference on Machine Learning, pp. 848–855. ACM.
- Suits, D.B., 1957. Use of dummy variables in regression equations. *Journal of the American Statistical Association* 52 (280), 548–551.
- Symonds, M.R., Moussalli, A., 2011. A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using Akaike's information criterion. *Behavioral Ecology and Sociobiology* 65 (1), 13–21.
- Tran, D., Toulis, P., Airolidi, E.M., 2015. Stochastic gradient descent methods for estimation with large data sets. *arXiv:1509.06459*.
- Tsuruoka, Y., McNaught, J., Tsujii, J.I.C., Ananiadou, S., 2007. Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics* 23 (20), 2768–2774.
- Wu, T.T., Chen, Y.F., Hastie, T., Sobel, E.M., Lange, K., 2009. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics* 25, 714–721.

Further Reading

- Anderson, D., Feldblum, S., Modlin, C., *et al.*, 2004. A practitioner's guide to generalized linear models. *Casualty Actuarial Society Discussion Paper Program*. 1–116.
- Lee, J.W., Lee, J.B., Park, M., Song, S.H., 2005. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis* 48 (4), 869–885.
- McCullagh, P., 1973. Nelder. JA (1989), Generalized Linear Models. CRC Monographs on Statistics & Applied Probability. New York: Springer Verlag.
- Motulsky, H., Christopoulos, A., 2004. Fitting Models to Biological Data Using Linear And Nonlinear Regression: A Practical Guide to Curve Fitting. Oxford University Press.
- Wood, S.N., 2017. Generalized Additive Models: An Introduction With R. CRC press.

Relevant Websites

- <https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/>
Multinomial logistic regression. R data analysis examples.
- <https://stats.idre.ucla.edu/r/dae/negative-binomial-regression/>
Negative binomial regression. R data analysis examples.
- <https://stats.idre.ucla.edu/r/dae/poisson-regression/>
Poisson regression. R data analysis examples.

<https://www.r-project.org>

R project.

<https://stats.idre.ucla.edu/r/modules/coding-for-categorical-variables-in-regression-models/>

UCLA, Institute for Digital Research and Education – IDRE.

<https://stats.idre.ucla.edu/r/dae/zinb/>

Zero-inflated negative binomial regression. R data analysis examples.

<https://stats.idre.ucla.edu/r/dae/zip/>

Zero-inflated Poisson regression. R data analysis examples.

<https://stats.idre.ucla.edu/r/dae/zero-truncated-negative-binomial/>

Zero-truncated negative binomial. R data analysis examples.

<https://stats.idre.ucla.edu/r/dae/zero-truncated-poisson/>

Zero-truncated Poisson. R data analysis examples.

Biographical Sketch

Audrone Jakaitiene is professor and senior researcher at Vilnius university. She defended her PhD thesis in physical sciences (Informatics 09P) in 2001. Title of the thesis “The Algorithms of Competing Risk Regression Models”, in which competing risks were forecasted using regression models and neural networks. Later A. Jakaitiene worked in Lithuanian and EU institutions developing her modelling skills in social, physical and biomedical sciences. As of 2005, her research paper topics could be split in three fields: (1) modelling and forecasting of Lithuanian, European Union and global economic indicators; (2) Analysis of biomedical data, with special interest to modelling of large-scale genetic data; (3) Educational data research covering international databases and an integrated analysis of Lithuanian data. In 2005–2016, she published 40 scientific papers (15 in Clarivate Analytics Web Of Science, H-index 4), participated in more than 30 conferences. From 2011 she is head of Bioinformatics and Biostatistics center at Human and Medical Genetics Department, Medical Faculty, Vilnius university.

Parametric and Multivariate Methods

Luisa Cutillo, University of Sheffield, Sheffield, United Kingdom; and Parthenope University of Naples, Naples, Italy

© 2019 Elsevier Inc. All rights reserved.

Parametric Statistical Inference

Statistics can be broken into two basic types. The first is known as descriptive statistics. This is a set of methods used to describe data that we have collected. The second type of statistics is inferential statistics. This is a set of methods used to make a generalization, estimate, prediction, decision. In inferential statistics a population is a general class of objects (genes, tissues, cell, etc.) for which we want to make an inferential statement. A set of observed items from the population is called a sample. The need for statistical inference comes from the existence of measurement errors. As an example, in experimental biological data there is both biological and technical variability, and hence the need to control and quantify variability. The use appropriate inferential tools may leverage both the current study and previous information to provide a reproducible inference. In statistical inference, the sample is used to make inferences about the population, meaning that the sample needs to be representative of the population and not biased. The majority of the inferential methods are based on the assumption that the samples are independent of each other. However, correlation is often induced. Examples of biological correlations are those between animals from the same litter (i.e., family correlation) or repeated measurements on the same subject. Another example of correlation is the one related to the so called batch effects. Suppose you replicate an experiment in two different labs; it is likely that the measurements taken within each lab are more similar to each other than to the measurements taken in different labs. The cause of these similarities might be the lab environment, batches of reagents, feed, or the operator who performed the experiment and made the measurements. These similarities induce correlation and if we neglect it by treating samples as if they are independent, we can introduce large errors into the analysis, which typically leads to a high error rate.

Parametric statistics is that part of statistics that assumes sample data follow a probability distribution based on a fixed set of parameters. On the other hand, in a non-parametric model, the parameter set is not fixed and can vary in case new relevant information is collected. A parametric model relies on specific assumptions about a given population and, when such assumptions are correct, parametric methods will produce more reliable estimates than non-parametric methods. Conversely, when the assumptions are not correct they have a greater chance of failing, and hence are not a robust statistical method. However the great advantage of parametric formulae is that they are simple and fast to compute. The simplicity of a parametric model, when accompanied by suitable diagnostic statistics, can compensate for the lack of robustness.

Maximum Likelihood Estimation

Given of a subset of individuals from within a statistical population, we need to estimate the characteristics of the whole population. The likelihood function expresses the evidence contained in a sample that is relevant for inferring the parameters of a given a statistical model. A likelihood function is a probability distribution considered as a function of its distributional parameterization argument. As an example, consider an observable random variable, X , modelled by a probability density function depending on the parameter θ .

We can consider the function $L(\theta|x) = P(X=x|\theta)$ as a likelihood function of θ , for a specific value x of X . This gives a measure of how likely any particular value of θ is, given that we know x is a realization of X . Suppose now that, $x_1, x_2, \dots, x_n$ is a sample of n independent and identically distributed observations, distributed according to an unknown probability density function, $f_0(\cdot)$. The function, f_0 , is usually assumed to belong to a specific parametric model, a family of distributions, $\{f(\cdot|\theta), \theta \in \Theta\}$ (where θ is a vector of parameters for this family), so that $f_0 = f(\cdot|\theta_0)$. The values of the set of parameters θ_0 are unknown. It is desirable to find an estimator which would be as close to the true value θ_0 as possible. To solve this task we can use the method of maximum likelihood that can be summarized in the following steps.

First we need to specify the joint density function for all observations. Hence, given an independent and identically distributed sample, $x_1, x_2, \dots, x_n$, the joint density function is

$$f(x_1, x_2, \dots, x_n|\theta) = f(x_1|\theta)f(x_2|\theta) \dots f(x_n|\theta) \quad (1)$$

If we consider the observed values $x_1, x_2, \dots, x_n$ to be fixed parameters, and θ as the function's variable, we can rewrite (1) as the likelihood function:

$$L(\theta) = f(x_1, x_2, \dots, x_n|\theta) = \prod_{i=1}^n f(x_i|\theta) \quad (2)$$

Usually, it is convenient working with the log-likelihood, the natural logarithm of the likelihood function:

$$l(\theta) = \ln(L(\theta)) = f(x_1, x_2, \dots, x_n|\theta) = \sum_{i=1}^n \ln(f(x_i|\theta)) \quad (3)$$

Indeed we can use the fact that the logarithm is an increasing function so it will be equivalent to maximizing the log-likelihood. The maximum likelihood estimator (MLE) of θ is the value, θ_0 , that maximizes (3). In the following we provide a few examples of MLE under three different parametric models, namely Normal, Gamma, and Poisson.

Normal Example

Consider the random variables, $X_1, X_2, \dots, X_n$ to be independent and identically distributed (iid) as a normal, $N(\mu, \sigma)$. Their density can be written as:

$$f(x_1, x_2, \dots, x_n | \mu, \sigma) = \prod_{i=1}^n \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \quad (4)$$

Interpreting this density as a function of the two parameters, μ and σ , the log-likelihood can be written as:

$$l(\mu, \sigma) = -n \ln \sigma - \frac{n}{2} \ln(2\pi) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 \quad (5)$$

Given that:

$$\begin{aligned} \frac{\partial l}{\partial \mu} &= \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu) \\ \frac{\partial l}{\partial \sigma} &= -\frac{n}{\sigma} + \sigma^{-3} \sum_{i=1}^n (x_i - \mu)^2 \end{aligned} \quad (6)$$

it comes that the MLE of μ is equal to the sample mean, $\hat{\mu}$, and the MLE of σ is equal to the unadjusted sample variance, $\hat{\sigma}$:

$$\begin{aligned} \hat{\mu} &= \frac{1}{n} \sum_{i=1}^n x_i \\ \hat{\sigma} &= \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \end{aligned} \quad (7)$$

Poisson Example

Assume, $X_1, X_2, \dots, X_n$ are iid Poisson random variables with the marginal frequency function, $P(X=x) = \frac{\lambda^x e^{-\lambda}}{x!}$. The log likelihood will thus be:

$$l(\lambda) = \sum_{i=1}^n (x_i \ln \lambda - \lambda - \ln x_i!) = \lambda \sum_{i=1}^n x_i - n \lambda - \sum_{i=1}^n \ln x_i! \quad (8)$$

whose derivative would be:

$$l(\lambda)' = \frac{1}{n} \sum_{i=1}^n x_i - n = \bar{X} - n \quad (9)$$

hence the MLE of λ , maximum of Eq. (8), is $\hat{\lambda} = \bar{X}$. Note that the MLE agrees with the method of moments in this case.

Gamma Example

In the last two examples the function l was actually concave and, equating the derivatives to zero, we were able to find a maximum of the log-likelihood in closed form. This is not always the case. Consider a final example where $X_1, X_2, \dots, X_n$ are iid Gamma distributed variables with marginal density:

$$f(x|\alpha, \lambda) = \frac{1}{\Gamma(\alpha)} \lambda^\alpha x^{\alpha-1} e^{-\lambda x}$$

The log-likelihood can be written as:

$$l(x|\alpha, \lambda) = \sum_{i=1}^n [\alpha \ln \lambda + (\alpha - 1) \ln x_i - \lambda x_i - \ln \Gamma(\alpha)] \quad (10)$$

If we try to maximize (10), we end up with a nonlinear equation in α that cannot be solved in closed form. In this situation we could employ a so-called *root-finding* method, exploiting the fact that when a function is continuous, and changes signs on an interval, it admits at least one zero on that interval. For this particular problem there are already coded (e.g., in Matlab and e.g., in Matlab and R) general optimization methods, that also provide a confidence interval.

Evaluating an Estimator

Supposing that we are given a random sample, $X_1, X_2, \dots, X_n$, such that each random variable, X_i , has the same distribution as $X \sim f(X|\theta)$. Let's assume that θ is an unknown parameter to be estimated. For example, θ might be the expected value of a random variable, $\theta = E(X)$. To estimate θ , we collect some data and we define a point estimator, $\hat{\theta}$, that is a function of the random sample, i.e., $\hat{\theta} = h(X_1, X_2, \dots, X_n)$. For example if $\theta = E(X)$, we might choose an estimator to be the sample mean, $\hat{\theta} = \frac{1}{n} \sum_{i=1}^n X_i$.

There are many possible estimators for θ , and in the previous paragraph we described the MLE approach. The questions we want to address here are: how can we make sure that an estimator is a good one? How do we compare different possible estimators? As an answer there is a list of some desirable properties that we would like our estimators to have. Indeed, a good estimator should be able to give us values that approach the real value of θ . To clarify this notion, we provide some definitions and examples in the following section, defining three main desirable properties for point estimators.

Bias

The first desirable property of a point estimator is related to the estimator's bias. The bias, $B(\hat{\theta})$, of an estimator, $\hat{\theta}$, tells us on average how far $\hat{\theta}$ is from the real value of θ . Let $\hat{\theta} = h(X_1, X_2, \dots, X_n)$ be a point estimator of θ . The bias of the point estimator $\hat{\theta}$ is defined by

$$B(\hat{\theta}) = E[\hat{\theta}] - \theta. \quad (11)$$

Intuitively, we would expect that a *good* estimator has a bias that is close to 0, indicating that on average, $\hat{\theta} \sim \theta$. We would like to point out that the value of the *bias* might depend on the actual value of the parameter θ , implying that $B(\hat{\theta})$ might be large or small according to the specific value θ assumes. Hence, another desirable property is that an estimator is *unbiased*. If $\hat{\theta} = h(X_1, X_2, \dots, X_n)$ is a point estimator of θ , we say that it is an *unbiased* estimator of θ if $B(\hat{\theta}) = 0$, for all possible values of θ . It is very important to note that, if an estimator is unbiased, it is not necessarily a good estimator. To clarify this concept consider the following example.

Example 1

As before assume that we are given a random sample, $X_1, X_2, \dots, X_n$, such that each random variable, X_i , has the same distribution as X . It is easy to show that the sample mean, $\hat{\theta} = \bar{X}$, and $\hat{\theta}_1 = X_1$ are both unbiased estimators of $\theta = E(X)$. Indeed we have:

$$\begin{aligned} B(\hat{\theta}) &= E[\hat{\theta}] - \theta \\ &= E[\bar{X}] - \theta \\ &= E[X] - \theta = 0 \end{aligned}$$

and

$$\begin{aligned} B(\hat{\theta}_1) &= E[\hat{\theta}_1] - \theta \\ &= E[X_1] - \theta \\ &= E[X] - \theta = 0 \end{aligned}$$

Mean Squared Error

We can guess that $\hat{\theta}_1$ should be a worse estimator with respect to the sample mean $\hat{\theta} = \bar{X}$. Therefore, we need to introduce other measures to ensure the goodness of an estimator. A very common measure is the mean squared error, (MSE). The mean squared error of a point estimator $\hat{\theta}$, referred to as $MSE(\hat{\theta})$, is defined as

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] \quad (12)$$

Note that, as it is defined, the MSE is a measure of the distance between $\hat{\theta}$ and θ , thus a smaller MSE usually suggests of a better estimator.

Example 2

As in the *Example 1*, assume that we are given a random sample, $X_1, X_2, \dots, X_n$, such that each random variable, X_i , has the same distribution as X . Assume now we know that $E(X) = \theta$ and $var(X) = \sigma^2$. We have already shown that the sample mean $\hat{\theta} = \bar{X}$ and $\hat{\theta}_1 = X_1$ are both unbiased estimators of θ . In order to compare the two estimators, let's compute $MSE(\hat{\theta})$ and the $MSE(\hat{\theta}_1)$:

$$\begin{aligned} MSE(\hat{\theta}) &= E[(\hat{\theta} - \theta)^2] \\ &= E[(\bar{X} - \theta)^2] \\ &= var(\bar{X} - \theta) + (E[\bar{X} - \theta])^2 \end{aligned}$$

and

$$\begin{aligned} \text{MSE}(\hat{\theta}_1) &= E\left[(\hat{\theta}_1 - \theta)^2\right] \\ &= E[(X_1 - E(X_1))^2] \\ &= \text{var}(X_1) = \sigma^2 \end{aligned}$$

Note that, since θ is constant, $\text{var}(\bar{X} - \theta) = \text{Var}(\bar{X})$. Moreover it is easy to see that $E[\bar{X} - \theta] = 0$. As a consequence we can write:

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= \text{var}(\bar{X}) \\ &= \frac{\sigma^2}{n} \end{aligned} \quad (13)$$

In conclusion, $\forall n > 1$

$$\text{MSE}(\hat{\theta}_1) > \text{MSE}(\hat{\theta}) \quad (14)$$

Hence, even if $\hat{\theta}$ and $\hat{\theta}_1$ are both unbiased estimators of the population mean, θ , $\hat{\theta}$ is a better estimator due to the fact that it has a smaller MSE. In general, if $\hat{\theta}$ is a point estimator for a population parameter, θ , we can write:

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= E\left[(\hat{\theta} - \theta)^2\right] \\ &= E[(\bar{X} - \theta)^2] \\ &= \text{var}(\hat{\theta} - \theta) + \left(E[\hat{\theta} - \theta]\right)^2 \\ &= \text{var}(\hat{\theta}) + B(\hat{\theta})^2 \end{aligned} \quad (15)$$

Consistency

Given a point estimator, another important property that we should discuss is *consistency*. We could say that an estimator $\hat{\theta}$ is consistent if it converges to the real value of θ as the sample size n increases. More formally, Let $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_n$, be a sequence of point estimators of θ . We say that $\hat{\theta}_n$ is a *consistent estimator* of θ , if

$$\lim_{n \rightarrow \infty} P(|\hat{\theta}_n - \theta| \geq \epsilon) = 0, \forall \epsilon > 0. \quad (16)$$

An important result, that enables an easier evaluation of consistency for an estimator, is that if:

$$\lim_{n \rightarrow \infty} \text{MSE}(\hat{\theta}_n) = 0 \quad (17)$$

then $\hat{\theta}_n$ is a consistent estimator of θ .

To illustrate how we can prove the consistency of an estimator, consider the results in *Example 2*. We could show the consistency of $\hat{\theta}_n = \bar{X}$ by looking at its MSE. Indeed we found previously that in this case $\text{MSE}(\hat{\theta}_n) = \frac{\sigma^2}{n}$. Hence, the (17) is verified. From this, we can conclude that $\hat{\theta}_n = \bar{X}$ is a consistent estimator for θ .

Multivariate Statistics

Data generally will consist of a number of variables recorded on a number of individuals. As an example consider weights, ages, and sexes of a sample of mice analyzed in an experiment. Usually, data collected on n individuals over q variables, say $x_1, \dots, x_p$, will be arranged in an $n \times q$ data matrix, with rows denoting individuals and the columns denoting variables:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,q} \\ x_{2,1} & x_{2,2} & \dots & x_{2,q} \\ \vdots & \ddots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,q} \end{bmatrix}$$

where $x_{i,j}$ is the value of variable x_j for individual i . Of course, the measurements in the i th row, namely, $x_{i1}, \dots, x_{ip}$, which are the measurements on the same individual, are correlated. If we arrange them as a column vector $\mathbf{x}_i$ defined as

$$\mathbf{x}_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,q} \end{bmatrix}$$

then $\mathbf{x}_i$ can be viewed as a multivariate observation (Zelterman, 2015; Johnson and Wichern, 2007; Rencher, 2012). Thus, the n rows of matrix X correspond to n multivariate observations, and the measurements within each $\mathbf{x}_i$ are usually correlated. On the other hand usually, $x_1, \dots, x_n$ are assumed to be uncorrelated or statistically independent but this may not always be true. Note that if the set of these n units constitutes the entire finite set of all possible individuals, then we have data available on the entire reference population. However, usually the data are obtained through a survey in which, on each of the individuals, all p characteristics are measured. Such a situation represents a multivariate sample. A sample represents the underlying population from which it is taken adequately or poorly. As a consequence, any summary derived from it merely represents the true population summary.

A very intuitive and immediate way of getting in touch with a multivariate dataset is to visualise it, before even computing any descriptive statistics. A two-dimensional scatter plot is a scatter plot matrix that arranges all possible two-way scatter plots in a $q \times q$ matrix. These displays can be enhanced with brushing, in which individual points or groups of points can be selected in one plot, and simultaneously highlighted in the other plots. See as a reference Fig. 1, where we show a scatter plot of the Fisher-Anderson iris data using the *R* function *pairs*. The most famous data set in multivariate analysis is the Iris data, analyzed by Fisher (1936) based on data collected by Anderson (1935). There are fifty specimens each of three species of iris: *setosa*, *versicolor*, and *virginica*. There are four variables measured on each plant: sepal length, sepal width, petal length, and petal width. Thus $n=150$ and $q=4$. Fig. 1 contains the corresponding scatter plot matrix, with species indicated by colours. We can use Fig. 1 as a starting diagnostic on the data. Indeed, we can note that the three species separate fairly well, with *setosa* especially distinct from the other two.

Basics

To analyse a multivariate data set derived from several mutually dependent variables, we need to find few basic quantities to summarise it. In univariate data there is only one variable under study, and we usually describe it by the population or sample mean, variance, skewness, and kurtosis. Similarly, for multivariate data we can extend such concepts. Let $\mathbf{x} = (x_1, \dots, x_q)^t$ be the $q \times 1$ random vector corresponding to the multivariate population under consideration. Let $\mu_i = E(x_i)$ and $\sigma_{i,i} = \text{var}(x_i)$ be respectively the population mean and variance of x_i . Furthermore, let $\sigma_{i,j} = \text{cov}(x_i, x_j)$ be the population covariance between x_i and x_j . A mean vector $E(\mathbf{x})$ of a multivariate variable is defined as the element-wise application of the expectation to a vector of random variables. That is:

$$E(\mathbf{x}) = \begin{bmatrix} E(x_1) \\ E(x_2) \\ \vdots \\ E(x_q) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_q \end{bmatrix} = \boldsymbol{\mu}$$

Moreover, the population variance can be generalised to be the matrix of all the population variances and covariances. Specifically, we can write the variance-covariance matrix of $\mathbf{x}$ as:

$$\begin{aligned} (\boldsymbol{\Sigma}) = (\sigma_{i,j}) &= \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \dots & \sigma_{1,q} \\ \sigma_{2,1} & \sigma_{2,2} & \dots & \sigma_{2,q} \\ \vdots & \ddots & & \\ \sigma_{q,1} & \sigma_{q,2} & \dots & \sigma_{q,q} \end{bmatrix} \\ &= \begin{bmatrix} \text{var}(x_1, x_1) & \text{cov}(x_1, x_2) & \dots & \text{cov}(x_1, x_q) \\ \text{cov}(x_2, x_1) & \text{var}(x_2, x_2) & \dots & \text{cov}(x_2, x_q) \\ \vdots & \ddots & & \vdots \\ \text{cov}(x_q, x_1) & \text{cov}(x_q, x_2) & \dots & \text{var}(x_q, x_q) \end{bmatrix} \end{aligned}$$

As observed previously, there usually exists dependence between $x_1, \dots, x_q$, hence we are interested in measuring the degree of linear dependence between them. Specifically, this is often measured using correlations. Let

$$\rho_{i,j} = \frac{\text{cov}(x_i, x_j)}{\sqrt{\text{var}(x_i)\text{var}(x_j)}} = \frac{\sigma_{i,j}}{\sqrt{\sigma_{i,i}\sigma_{j,j}}} \quad (18)$$

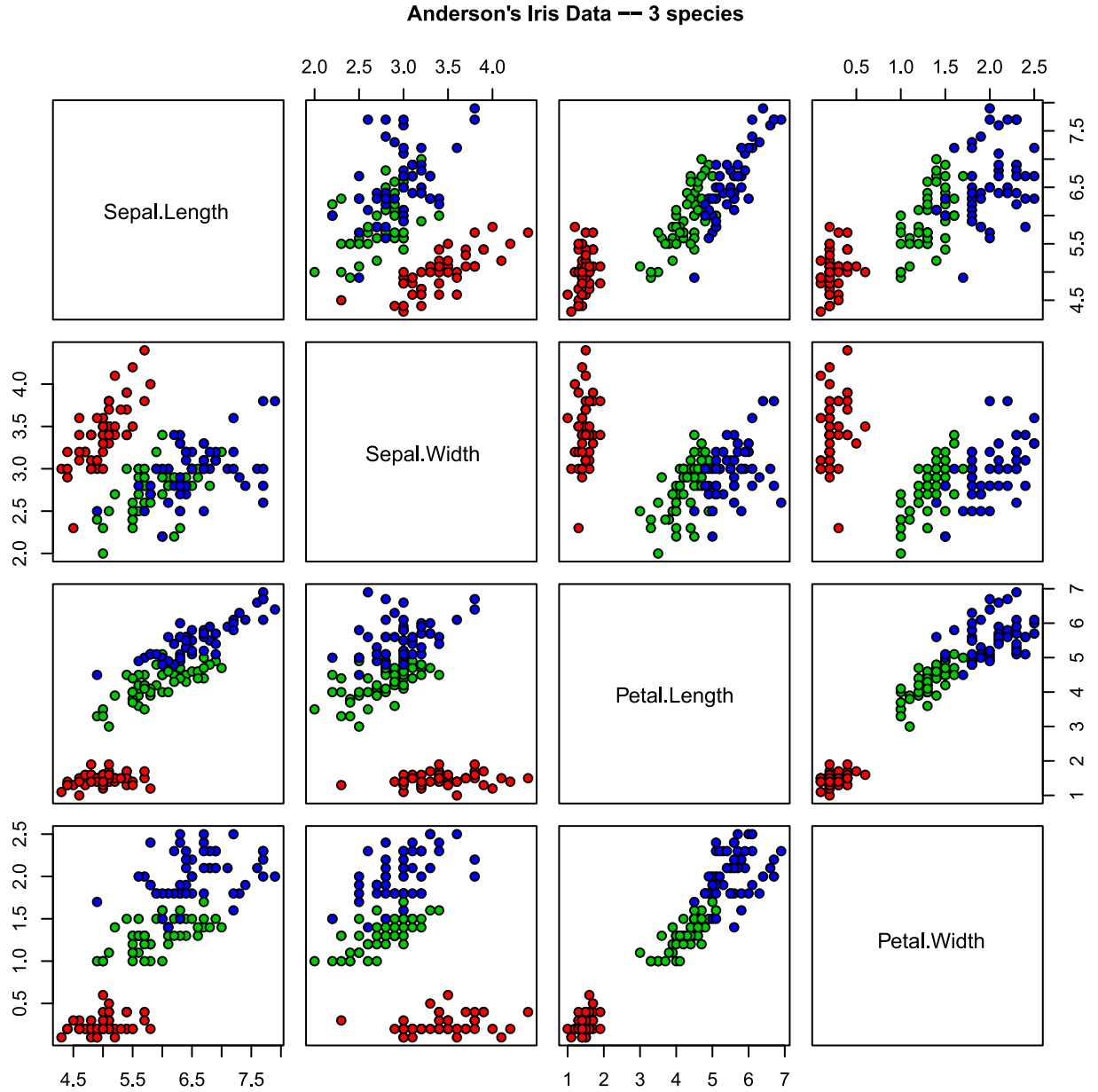


Fig. 1 A scatter plot matrix for the Fisher-Anderson iris data. In each plot, red indicates *setosa* plants, green indicates *versicolor*, and blue indicates *virginica*.

be the Pearson's population correlation coefficient between x_i and x_j . We define the population correlation matrix as:

$$\rho = (\rho_{ij}) = \begin{bmatrix} \rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,q} \\ \rho_{2,1} & \rho_{2,2} & \cdots & \rho_{2,q} \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{q,1} & \rho_{q,2} & \cdots & \rho_{q,q} \end{bmatrix} = \begin{bmatrix} 1 & \rho_{1,2} & \cdots & \rho_{1,q} \\ \rho_{2,1} & 1 & \cdots & \rho_{2,q} \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{q,1} & \rho_{q,2} & \cdots & 1 \end{bmatrix}$$

Note that both (Σ) and ρ are symmetric by construction. Further, the concept of skewness is generalised as:

$$\text{multivariate skewness } \beta_{1,q} = E[(\mathbf{x} - \mu)'(\Sigma)(\mathbf{x} - \mu)]^3 \quad (19)$$

and the kurtosis is generalised as:

$$\text{multivariate kurtosis } \beta_{2,q} = E[(\mathbf{x} - \mu)'(\Sigma)(\mathbf{x} - \mu)]^2 \quad (20)$$

The quantities just introduced, provide a basic summary of a multivariate population. Of course, when we have a q -variate random sample, $\mathbf{x}_1, \dots, \mathbf{x}_n$, of size n , with the n by p data matrix, $\mathbf{X}$, defined as

$$\mathbf{X} = \begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{bmatrix} \in \mathcal{R}^{n \times q}$$

we define,

$$\text{sample mean vector : } \bar{\mathbf{x}} = n^{-1} \sum_i^n \mathbf{x}_i \quad (21)$$

$$\text{sample variance – covariance matrix : } \mathbf{S} = (n - 1)^{-1} \sum_i^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})' \quad (22)$$

$$= (n - 1)^{-1} \mathbf{X}'\mathbf{X}$$

Similarly we can derive skewness and kurtosis.

Many multivariate problems involve data reduction, description and estimation. As a general theme, most of the situations either require some matrix decomposition and transformation or use a distance-based approach. Distributional assumptions, such as multivariate normality, are very helpful in assessing the quality of estimation and make the multivariate statistics *parametric*.

Main Multivariate Techniques

The key to understanding multivariate statistics is related to the kind of problems each technique is suited for, the objectives of each technique, the data structure required for each technique, and the underlying mathematical model of each technique. In order to understand multivariate analysis, it is important to understand the basic multivariate statistical concepts we described in the previous section. In the following, we will attempt to give a basic description of the main parametric multivariate methods, with the aim of providing a way to choose between them, according to the objective of the analysis and data structure. Many of the methods introduced in the following were already described and addressed in other articles of this book.

Data Quality Assessment

Before deciding on an analysis technique, it is important to have a clear understanding of the best distributional assumption, and of the quality of the data. Estimation of skewness, and kurtosis are helpful in examining the distribution. Nevertheless it is important to understand the magnitude of missing values in the observations, and to determine whether to discard them or to use specific methods to estimate the missing observations. Also, a very important data quality measure is represented by *outliers*. It is, indeed, important to determine whether the outliers should be retained in the analysis or removed. The shortcoming of keeping them is that they may cause a distortion to the data. The advantage of eliminating outliers is that they might support the assumptions of a specific distribution, e.g., normality. Although it is crucial to understand what the outliers represent.

Multiple Regression Analysis

Multiple regression examines the relationship between a single dependent variable, and two or more independent variables. It is the most commonly utilised multivariate technique and is often used as a forecasting tool. Multiple regression analysis aims to determine the linear relationship that minimises the sum of squared variances. As a consequence, the assumptions of normality, linearity, and equal variance are crucial. This method implies the estimation of model coefficients (called weights) that represent the marginal impacts of each variable.

Logistic Regression Analysis

This technique is a variation of multiple regression that allows for the prediction of an event. It is sometimes referred to as *choice model*. This method's objective is to make a probabilistic assessment of a binary choice and hence it utilises (typically) binary dependent variables. The independent variables can be either discrete or continuous. The outcome of this analysis can be provided in terms of a contingency table. This will show the classification of observations as matching or not matching the predicted events. As a measure of the effectiveness of this model we can take the sum of events that were predicted to occur and which actually did occur (true positives) and the events that were predicted to not occur which actually did not occur (false negatives), divided by the total number of events. This measure helps predicting, for example the choices consumers might make when presented with alternatives.

Linear and Quadratic Discriminant Analysis

The purpose of discriminant analysis is to correctly classify observations or subjects into homogeneous groups. In the parametric approach, the independent variables must have a high degree of normality. Discriminant analysis builds a linear discriminant function in which normal variates are assumed to have unequal mean and equal variance. If normal variates are assumed to have unequal mean and unequal variance, the discriminant function is constructed to be quadratic. The discriminant function can then be used to classify the observations. The overall fit is assessed by looking at the degree to which the group means differ, and how well the model classifies. To determine which variables have the most impact on the discriminant function, it is possible to look at F values. The higher the F , the more impact that variable has on the discriminant function. This helps categorize (i.e., discriminate) subjects, between different homogeneous groups (e.g., mice that respond to a drug, mice that do not respond to a drug).

Multivariate Analysis of Variance

It is well known that analysis of variance (ANOVA) assesses the differences between groups (by using T tests for two means, and F tests between three or more means). Similarly, in the case of a multivariate dataset, Multivariate Analysis of Variance (MANOVA) examines the dependence relationship between a set of dependent measures across a set of groups. More explicitly, this technique examines the relationship between several categorical independent variables, and two or more dependent variables. Typically MANOVA is based on a specific hypothesis of relationship between dependent measures, and is often used to validate experimental designs. The null hypothesis is of no difference between categories (e.g., different treatments). In this technique the independent variables are categorical and the dependent variable is not. A limitation of this method is sample size, usually needing 15–20 observations needed per cell. However, too many observations per cell (over 30) often cause overfitting, and cell sizes also should be roughly equal. This is due to the normality assumption of the dependent variables. The model fit is determined by examining the mean vector across groups. If there is a significant difference in the means, the null hypothesis can be rejected and treatment differences can be determined.

Factor Analysis

Factor Analysis (FA) is an independence technique, in which there is no dependent variable. The motivation of FA relies on the fact that often there are many variables involved in a research design, and it is usually helpful to reduce the variables to a smaller set of factors, aiming mainly to understand the underlying structure of the data matrix. The relationship of each variable to the underlying factor is expressed by the so-called factor loading. Again, in the parametric approach, the independent variables are usually assumed to be normal and continuous, with at least three to five variables loading onto a factor. The sample size should be large (i.e., the number of observations should be greater than 50), with at least five observations per variable.

We can broadly define two main factor analysis methods: principal component analysis, which extracts factors based on the total variance of the factors, and common factor analysis, which instead extracts factors based on the variance shared by the factors. Principal component analysis is used to find the fewest number of variables that explain the most variance, whereas common factor analysis is used to look for the latent underlying factors. Usually the first factor extracted explains most of the variance. The factor loadings express the correlations between the variables and the factor. As rule of the thumb, a factor loading of at least 0.4 strongly suggests that a specific variable can be attributed to a factor.

Cluster Analysis

Cluster analysis of a multivariate dataset aims to partition a large data set into meaningful subgroups of subjects. Based on a similarity measure between different subjects, data are divided according to a set of specified characteristics. In this case, outliers also play a pivotal role. The problem of outliers is often caused by variables that might be irrelevant. Moreover it is desirable that the sample under study provide a good representation of the underlying population, and that the variables are uncorrelated.

Clustering methods can be: *hierarchical*, meaning that are based on a tree approach and more appropriate for smaller data sets; *non-hierarchical*, having the shortcoming of requiring the user to specify a priori the number of clusters. Different algorithms are based on specific notion of a cluster, with specific properties. Typical cluster models include: connectivity models, centroid models, distribution models, density models, subspace models, group models, graph-based models and neural models.

Conclusions

In this article we have described the main features of parametric statistics. Then we provided the basics of multivariate statistics, and finally we described briefly a few main multivariate techniques. In particular, we focused on the specific type of research question each method is best suited. It is important to figure out the main strengths and weaknesses of each multivariate technique before trying to interpret any results of its application. Current statistical packages (e.g., in the *R* programming language and *Matlab*) allow one to use built-in procedures features the state of the art of multivariate statistical methods, but the results can be disastrously misinterpreted without adequate care.

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Anderson, E., 1935. The irises of the Gaspé Peninsula. Bulletin of the American Iris Society 59, 2–5.
- Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7 (Part II), 179–188.
- Johnson, R.A., Wichern, D.V., 2007. Applied Multivariate Statistical Analysis, sixth ed. Duxbury.
- Rencher, A.C., 2012. Methods of Multivariate Analysis. Wiley.
- Zelterman, D., 2015. Applied Multivariate Statistics With R. Springer.

Stochastic Processes

Maria Francesca Carfora, Istituto per le Applicazioni del Calcolo CNR, Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Stochasticity plays a lead role in biological modelling: from the earliest models applying branching processes to genealogical questions to birth-and-death processes in population dynamics and, more recently, to simulation of stochastic networks in systems biology, stochastic phenomena have always been central (Schuster, 2016). Nowadays, many processes in genetics, epigenetics and cellular metabolism are ruled and controlled by small numbers of molecules. At such extremely low concentrations the deterministic approach fails to capture the discrete and stochastic nature of chemical kinetics. In such a context, noise is increasingly seen as a force shaping biology (Bressloff, 2014). Just to cite a few examples, noise-induced transitions affect stochastic ion channels; molecular motors exploit thermal noise to realize fluctuation-driven transport; the numerous amount of protein-protein interactions (PPIs) causes noise in biochemical signaling; noise can cause genes to express differently. Stochasticity in gene expression can have strong consequences on cell behaviour and function, and its potential advantages are also to be considered. For example, intrinsic noise can provide the flexibility needed by cells to adapt to fluctuating environments or respond to sudden stresses and can also support a mechanism by which population heterogeneity is established during cell differentiation and development. Similarly, in biochemical signaling the promiscuity of PPIs that create signal noise may help the recovery of cell functions through homologous pathways. This signaling noise may also be important for increasing robustness of signaling between cells by dampening incorrect events.

Stochastic aspects of cellular function are hence crucial. The rapid advance in experimental techniques for imaging and probing cells at the molecular level (Single Particle Tracking), as well as the advances in single-molecule techniques for measuring the force-dependent dynamics of molecular motors, DNA, and other macromolecules vital for cell function, stimulated a constantly growing interest in stochastic modelling (Ullah and Wolkenhauer, 2011; Wilkinson, 2012). However, it is worth noting that almost all stochastic models in science assume the so-called Markov property: full knowledge of the present state of the system allows prediction of the future (or of the past). For example, a simple two-state continuous-time Markov process can be used to model the opening and closing of an ion channel, or the binding and unbinding of a ligand molecule to a protein receptor.

Aim of this contribution will be a clear and concise introduction to the topic; we will not adopt the mathematically oriented exposition or the measure-theoretic framework of many classical textbooks (e.g., Karlin and Taylor, 1975) to whom we refer for further reading. Following the approach of many basic and intermediate textbooks (Cox and Miller, 2001; Gardiner, 2004; Van Kampen, 2007; Sheldon, 2009; Stirzaker, 2005), we present the theory essentially restricting ourselves to Markov processes; applications to biology are outlined where possible, also through a wide review of the recent literature on stochastic modelling. Finally, numerical simulation of stochastic processes is faced, by description of the simplest and most popular algorithms and also by reference to more advanced techniques.

Fundamentals

Loosely speaking, a stochastic process is the mathematical representation of a system which evolves probabilistically. Then, it is a collection of random variables $\{X(t), t \in T\}$, where both the index t and the random variable X can be discrete or continuous. In the following, for simplicity we generally refer to t as time, even if it can be useful to consider stochastic processes in different formal spaces, such as the genotype or sequence space. There, the points represent individual genotypes and the distance between genotypes counts the minimal number of point mutations required to bridge the interval between them. Neutral evolution and Darwinian selection, for example, can be visualized as processes in genotype space (Ewens, 1979).

A listing of the values assumed by the random variable at specific times, i.e., a sequence of time-ordered pairs (x, t) , is called a *trajectory* or a *sample path* of the process. If the set T is countable (finite or enumerable), we call X a discrete-time process; if, on the contrary, T is (an interval of) the real axis, X is called a continuous-time process. The set of possible values X , called the *state space* of the process, can be uni- or multidimensional; moreover, it can be discrete or continuous.

The complete specification of a process requires knowledge of the joint distributions (or the joint densities, if they exist) of the random variables $X_1, \dots, X_k$ for any k ; in the following we assume that the finite joint probability densities $p(x_1, t_1; \dots; x_n, t_n)$ exist for any n . Note that, in such a notation, we have not yet assumed anything about the ordering of the random variables. The marginalization of the joint density over one of the random variables gives the so-called Chapman-Kolmogorov Equation:

$$p(x_1, t_1; \dots; x_{n-1}, t_{n-1}) = \int p(x_1, t_1; \dots; x_n, t_n) dx_n \quad (1)$$

obviously, for discrete variables, the integral in Eq. (1) is replaced by a sum.

Model equations for stochastic processes can be grouped in two classes: the first group deals with the representation of the trajectories of the process, given by the random variable $X(t)$. Mathematically speaking, these trajectories are described by

stochastic differential equations (SDE), that are differential equations in which one or more of the terms are stochastic, so that the solution itself is a stochastic process. The second group describes the evolution of the probability $p(x, t) = \text{Prob}[X(t)=x]$ by means of deterministic differential equations, such as the Chapman-Kolmogorov Equation in its differential form and the derived ones (Fokker-Planck, Chemical Master Equation,...).

Some Definitions

Let us introduce the main properties characterizing some classes of stochastic processes that will be useful in the applications.

Independence

The simplest class of stochastic processes is characterized by complete independence of events: $X(t)$ does not depend on any of the $X(s)$ for $s < t$. Then the joint density fully factorizes:

$$p(x_1, t_1; \dots; x_n, t_n) = p(x_1, t_1)p(x_2, t_2) \dots p(x_n, t_n)$$

Markov

The Markov property assumes that prediction of the future requires knowledge of the present alone. In other terms, a stochastic process is a Markov process if for arbitrary times $t_1 < t_2 < \dots < t_n$ the conditional probability reduces to a transition probability between the last two states:

$$p(x_n, t_n | x_1, t_1; \dots; x_{n-1}, t_{n-1}) = p(x_n, t_n | x_{n-1}, t_{n-1}) \quad (2)$$

Then, any joint probability density can be expressed as the product of transition probabilities and an initial probability:

$$p(x_1, t_1; x_2, t_2; \dots; x_n, t_n) = p(x_n, t_n | x_{n-1}, t_{n-1}) \dots p(x_2, t_2 | x_1, t_1) p(x_1, t_1) \quad (3)$$

Stationarity

A stochastic process is said to be *strictly* or *strongly* stationary if the joint probability densities are invariant under time translations:

$$p(x_1, t_1; x_2, t_2; \dots; x_n, t_n) = p(x_1, t_1 + \Delta t; x_2, t_2 + \Delta t; \dots; x_n, t_n + \Delta t)$$

Wide-sense or *weak* stationarity requires instead only the invariance under time translation of the mean and covariance of the process:

$$E[X(t)] = \mu_X(t) = \mu_X(t + \Delta t)$$

$$\text{Cov}[X(t_1), X(t_2)] = C_X(t_1, t_2) = C_X(t_1 - t_2, 0)$$

In this case, the mean of the process is a constant and its covariance only depends on the time difference $t_1 - t_2$.

Continuity

A process is continuous if each of its sample paths is a continuous function of t . For a Markov process, this definition specifies as

$$\forall \varepsilon > 0 \quad \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int_{|x-z| > \varepsilon} p(x, t + \Delta t | z, t) dx = 0$$

uniformly in z , t and Δt .

Gaussianity

A process is Gaussian if every finite joint density is a multivariate Gaussian. Then the process is completely specified by its mean $E[X(t)]$ and covariance $E[X(t), X(s)]$ functions.

Markov Processes

Mathematical models generally make simplifying assumptions about properties of the phenomena being modelled. Such modelling assumptions often do not hold exactly; however they frequently allow a good compromise between accuracy and tractability of models. The common assumption when applying stochastic processes to modelling is the Markov property; thus, in the following, we restrict our attention to Markov processes. Methods for dealing with non-Markov processes, generally by their embedding in suitable Markov processes, can be found in classical textbooks (e.g., [Cox and Miller, 2001](#)).

When the stochastic process under consideration is Markov, the Chapman-Kolmogorov [Eq. \(1\)](#) is equivalent to an identity on the transition probability densities:

$$p(x_3, t_3 | x_1, t_1) = \int p(x_3, t_3 | x_2, t_2) p(x_2, t_2 | x_1, t_1) dx_2 \quad (4)$$

informally, this says that the probability of a transition from x_1 to x_3 can be expressed by adding up the probabilities of transitions from x_1 to any possible intermediate state x_2 and then from x_2 to x_3 . Note that the integral notation in [Eq. \(4\)](#) has to be intended wide-sense: it represents an integral only for continuous processes under suitable regularity hypotheses. For processes with discrete

state space it will be replaced by a summation while for discontinuous processes the notation will include both integration on continuity intervals and summation on single discontinuity points.

In any case, we can obtain from Eq. (4) differential equations for the transition probabilities. To simplify the notation, we assume fixed and sharp initial conditions (x_0, t_0) so that the unconditioned probability of the state (x, t) is the same as the probability of the transition from (x_0, t_0) to (x, t) :

$$p(x, t) = p(x, t | x_0, t_0)$$

Then if we replace t_1, t_2, t_3 with $\tau, t - dt, t$ and pass to the limit we obtain the forward equation

$$\frac{\partial}{\partial t} p(x, t) = - \sum_i \frac{\partial}{\partial x_i} (A_i(x, t) p(x, t)) + \frac{1}{2} \sum_{ij} \frac{\partial^2}{\partial x_i \partial x_j} (B_{ij}(x, t) p(x, t)) + \int dz (W(x|z, t) p(z, t) - W(z|x, t) p(x, t)) \quad (5)$$

it is important to notice that the integral sign stands actually for a principal value integral, since the transition probability may approach infinity.

The three terms $A(x, t)$, $B(x, t)$ and $W(x|z, t)$ represent drift, diffusion and transition for discontinuous jumps, respectively; in other words, if $X(t)$ is a continuous process, in the limit $z \rightarrow x$ the expectation value of the increment $X(t + dt) - X(t)$ approaches $A(X(t), t)dt$ and its covariance converges to $B(X(t), t)dt$. Let us consider some specific cases: when $W=0$ we obtain continuous processes, specifically pure drift processes ($B=0$, Liouville equation), pure diffusion processes – Brownian motion ($A=0$, Wiener equation), or drift-diffusion processes ($A \neq 0$ and $B \neq 0$, Fokker-Planck equation). On the contrary, when only $W \neq 0$ the variables x and z evolve in jumps and the process is described by a so-called master equation.

While the Fokker-Planck equation is used with problems where the initial distribution is known, if the problem is to know the distribution at previous times, the Kolmogorov backward equation can be used:

$$\frac{\partial}{\partial t} p(x_0, t_0 | x, t) = - \sum_i A_i(x, t) \frac{\partial p(x_0, t_0 | x, t)}{\partial x_i} - \frac{1}{2} \sum_{ij} B_{ij}(x, t) \frac{\partial^2 p(x_0, t_0 | x, t)}{\partial x_i \partial x_j} \quad (6)$$

here the final condition (x, t) is given while the initial state and time (x_0, t_0) are variable and integration is performed backward in time. Such a formulation is particularly useful in problems involving *hitting times* or *escape times*, since the time to reach a specific target (or to leave a specific region), the so-called First Passage Time, FPT, is a random variable whose probability density satisfies a Kolmogorov backward equation.

Finally, a number of important cellular processes at the macromolecular level involve a coupling between continuous external variables and discrete internal variables, and this coupling is modelled using a stochastic hybrid system (Bressloff, 2014; Wilkinson, 2012). Consider, for example, molecular motors, that undergo a cyclic sequence of conformational changes after reacting with one or more molecules of a chemical such as ATP, resulting in the release of chemical energy; another example is given by voltage-gated or ligand-gated ion channels, in which the opening and closing of the channel depend on an external variable such as membrane voltage or calcium concentration.

Main Markov Processes

Let us introduce now, by some examples, the main Markov processes and some of their applications.

Markov chains

A Markov chain is a process that occurs in a series of time-steps in each of which a random choice is made among a finite (or also enumerable) number of states; since both the index set and the state space are discrete, we denote by $X_n = X(t_n)$; the transition probability can then be represented by a matrix $P = (p_{ij})$, where p_{ij} is the probability of moving from state i to state j : $p_{ij} = \text{Prob}[X_{n+1} = j | X_n = i]$. For homogeneous chains, these probabilities do not depend on t , i.e., they are stationary. Then, the initial distribution, together with the transition matrix P , determines the probability distribution for any state at all future times.

All the elements of the matrix P are non-negative and its row sums are unity; such matrices are called stochastic matrices. Now, the states of a Markov chain are *recurrent*, if the probability of eventual return to that state is 1, or *transient*, otherwise. State types and properties of the chain are related to properties of the stochastic matrix (Cox and Miller, 2001).

The classical model of enzyme activity, Michaelis-Menten kinetics, can be viewed as a Markov chain, where at each time step the reaction proceeds in some direction (Ingalls, 2013). Markov chains also have many applications in biological modelling, particularly for population growth processes or epidemics models (Allen, 2010). Branching processes are other examples. These chains have been used in population genetics to model variety-increasing processes such as mutation as well as variety-reducing effects such as genetic drift and natural selection. Markov chains take part in the modelling of hybrid stochastic systems (Ullah and Wolkenhauer, 2011; Wilkinson, 2012). Finally, they are the basis for Markov chain Monte Carlo (MCMC) methods, a class of algorithms for sampling from a probability distribution based on constructing a Markov chain that has the desired distribution as its equilibrium distribution.

The Poisson process

We consider point events occurring randomly in time and, denoting by $N(t, t + \Delta t)$ the number of events in $(t, t + \Delta t]$, we assume that there is a positive constant ρ (rate of occurrence) such that, as $\Delta t \rightarrow 0$,

$$\text{Prob}[N(t, t + \Delta t) = 0] = 1 - \rho\Delta t + o(\Delta t)$$

$$\text{Prob}[N(t, t + \Delta t) = 1] = \rho\Delta t + o(\Delta t)$$

where, as usual, $o(\Delta t)$ denotes a general and unspecified remainder term of smaller order than Δt . Moreover, we assume that $N(t, t + \Delta t)$ is independent of occurrences in $(0, t]$. In such a process, the number of events in any interval of length Δt has a Poisson distribution of mean $\rho\Delta t$:

$$\text{Prob}[N(t, t + \Delta t) = k] = \frac{(\rho\Delta t)^k e^{-\rho\Delta t}}{k!} \quad k = 0, 1, \dots$$

while the intervals between successive events are independent and exponentially distributed. Evolutionary models in continuous time (Ewens and Grant, 2005; cap. Kloeden and Platen, 1992) use Poisson processes; the activity of nerve cells and neuron firing in LIF model are described by Poisson processes (Tuckwell, 1989). Generalizations of the Poisson process, allowing the rate ρ to vary with time, or also to depend on the number of events already occurred, for example, leads to pure-birth processes; to epidemic models (Allen, 2010).

Brownian motion

The most important continuous time stochastic process is Brownian motion. It originated in physics as a description for the motion of small particles immersed in a fluid. Today, along with its many generalizations and extensions, it occurs in numerous and diverse areas of pure and applied science. Brownian motion is a mean zero, continuous process in both time and state space with independent Gaussian increments. Then, $W(t)$ is a Brownian motion (or a Wiener process) with diffusion coefficient σ^2 if:

1. $W(0) = 0$;
2. $W(t)$ has independent, stationary increments; and
3. for every t , $W(t)$ is normally distributed with zero mean and variance $\sigma^2 t$.

Then, an initially sharp distribution spreads in time: individual trajectories are extremely variable and diverge after short times. A simple generalization is the process with drift, where the increments in a small interval dt have mean μdt and variance $\sigma^2 dt$. The probability density of the Wiener process is the solution of the Fokker-Planck Eq. (5), that in this case reads

$$\frac{\partial p(x, t)}{\partial t} = \frac{1}{2} \sigma^2 \frac{\partial^2 p(x, t)}{\partial x^2}$$

Passive, as well as facilitated diffusion in cells can be modelled by an overdamped Brownian motion; the theory of Brownian ratchets is based on Brownian particles moving in a periodic ratchet (asymmetric) potential (Bressloff, 2014, for both).

Methodologies for Numerical Investigation

Numerical investigation on a stochastic process can involve Monte Carlo simulation of the trajectories to obtain approximate solutions of the process SDE; alternatively, the probability distribution function of the process as a function of time can be obtained by solving the related equations (Fokker-Planck Equation (Eq. (5)), Chemical Master Equation, CME (Eq. (6)). Let us briefly describe some tools for both approaches.

Methods for SDEs

In physical science, SDEs are usually written as Langevin equations. Those forms consist of an ordinary differential equation containing a deterministic function and an additional term which represents random white noise calculated as the derivative of Brownian motion or the Wiener process:

$$dX(t) = a(X(t))dt + b(X(t))dW(t) \quad (7)$$

However, other types of random behaviour are possible, such as jump processes.

Numerical solution of SDEs is a quite young field; algorithms developed for the solution of ODEs can be generalized but usually they work very poorly for SDEs. We describe here the simplest time-discrete approximation, the Euler-Maruyama method that is the stochastic generalization of Euler method for ODEs. A good reference text for other and more sophisticated algorithms is Kloeden and Platen (1992).

Consider the stochastic differential Eq. (8) with initial condition $X(0) = x_0$, and suppose that we wish to solve this SDE on some time interval $[0, T]$. Then the Euler-Maruyama approximation to the true solution $X(t)$ is the Markov chain Y defined as follows:

1. partition the interval $[0, T]$ into N equal subintervals of width $\Delta t > 0$: $0 = \tau_0 < \tau_1 < \dots < \tau_N = T$ and $\Delta t = T/N$;

2. set $Y(0)=x_0$; and
3. recursively define Y_n for $1 \leq n \leq N$ by $Y_{n+1}=Y_n+a(Y_n)\Delta t+b(Y_n)\Delta W_n$, where $\Delta W_n=W_{\tau_{n+1}}-W_{\tau_n}$.

The random variables ΔW_n are independent and identically distributed normal random variables with expected value zero and variance Δt . In simulations, to construct Y_n we only need to generate realizations of such random variables via a pseudo-random number generator at any timestep. Note that, with similar procedures (Sheldon, 2009; cap. Ingalls, 2013), it is also possible to simulate random variables having a different (non Gaussian) distribution, when the adopted model assumes correlated, or “colored”, noise instead of white noise.

Simulation of Reacting Systems

The Stochastic Simulation Algorithm (SSA) proposed by Gillespie (1977) is a numerical procedure for the exact simulation of the time evolution of a reacting system. In the limit of large number of reactants it converges (as the CME) to the deterministic solution of the law of mass action. Now, while traditional continuous and deterministic biochemical rate equations, typically modelled as a set of coupled ordinary differential equations, rely on bulk reactions that require the interactions of millions of molecules, the Gillespie algorithm allows a discrete and stochastic simulation of a system with few reactants because every reaction is explicitly simulated. The physical basis of the algorithm is the collision of molecules within a reaction vessel. It is assumed that collisions are frequent, but collisions with the proper orientation and energy are infrequent. Therefore, all reactions within the Gillespie framework must involve at most two molecules. Reactions involving three molecules are assumed to be extremely rare and are modelled as a sequence of binary reactions. It is also assumed that the reaction environment is well mixed.

The algorithm, in its original version, proceeds through these steps: Set the initial values for the number of molecules in the system, the reaction constants, and the random number generators. Then repeat the following two steps until the number of reactants is zero or the simulation time has been exceeded:

1. Two independent random numbers are generated to determine which reaction will take place as well as the time interval to this occurrence. The probability of a given reaction to be chosen depends on the different rate constants and population size of the chemical species.
2. The current time is updated by the randomly generated time and so is the molecule count based on the reaction that occurred.

The algorithm is computationally expensive and thus many modifications and adaptations exist, including the Next Reaction Method (Gibson and Bruck, 2000), more efficient both in terms of number of operations and number of random numbers used; the tau-leaping (Daniel, 2001), performing all reactions for an interval of length tau before updating the propensity functions; hybrid stochastic-deterministic techniques have also been proposed (see Wilkinson (2012) and references therein), where abundant reactants are modelled with deterministic behaviour. Another alternative to the SSA is the Stochastic Simulation algorithm (StochSim) (Le Novère and Shimizu, 2001; Morton-Firth and Bray, 1998), where time is quantized in intervals whose size depends on the most rapid reaction in the system. Then in each interval one or two objects (species or pseudo-species) are selected at random and the corresponding reaction will occur or not according to probabilities given in the look-up tables.

Future Directions/Closing Remarks

Stochastic processes and biological/biochemical modelling are increasingly merging in recent years as technology has started giving real insight into intra-cellular processes: quantitative real-time imaging of expression at the single-cell level and improvement in computing technology are allowing modelling and stochastic simulation of such systems at levels of detail previously impossible. “The message that keeps being repeated is that the kinetics of biological processes at the intra-cellular level are stochastic, and that cellular function cannot be properly understood without building that stochasticity into *in silico* models” (Wilkinson, 2012, Preface). The relevance of noise and stochastic modelling to state-of-the-art molecular and cell biology is thus unquestionable. Just to cite a few advanced topics that are currently considered: models of passive and active transport in cells; models of self-organization of cytoskeletal structures; models for the interplay between diffusion and nonlinear chemical reactions.

Understanding and modelling of biological noise, that plays an important role in cell fate decisions, environmental sensing and cell-cell communication, is another issue of main interest, as reported by Eldar and Elowitz in a recent review (Eldar and Elowitz, 2010).

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics

References

- Allen, L.J.S., 2010. *An Introduction to Stochastic Processes With Applications to Biology*, second ed. Chapman and Hall/CRC.
- Bressloff, P.C., 2014. *Stochastic Processes in Cell Biology*. Springer.

- Cox, D.R., Miller, H.D., 2001. *The Theory of Stochastic Processes*. Boca Raton, FL: Chapman and Hall/CRC.
- Eldar, A., Elowitz, M.B., 2010. Functional roles for noise in genetic circuits. *Nature* 467 (7312), 167–173.
- Ewens, W.J., 1979. *Mathematical Population Genetics*. Berlin; New York, NY: Springer-Verlag.
- Ewens, W.J., Grant, G., 2005. *Statistical Methods in Bioinformatics: An Introduction*, second ed. Springer.
- Gardiner, C.W., 2004. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, third ed. Springer-Verlag.
- Gibson, M.A., Bruck, J., 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A* 104 (9), 1876–1889.
- Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* 81 (25), 2340–2361.
- Gillespie, D.T., 2001. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics* 115 (4), 1716–1733.
- Ingalls, B.P., 2013. *Mathematical Modelling in Systems Biology: An Introduction*. MIT Press.
- Karlin, S., Taylor, H.M., 1975. *A First Course in Stochastic Processes*, second ed. Academic Press.
- Kloeden, P.E., Platen, E., 1992. *Numerical Solution of Stochastic Differential Equations*. Berlin; New York: Springer-Verlag.
- Le Novère, N., Shimizu, T., 2001. Stochsim: Modelling of stochastic biomolecular processes. *Bioinformatics* 17 (6), 575–576.
- Morton-Firth, C.J., Bray, D., 1998. Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology* 192 (1), 117–128.
- Schuster, P., 2016. *Stochasticity in Processes. Fundamentals and Applications to Chemistry and Biology*. Springer.
- Sheldon, M.R., 2009. *Introduction to Probability Models*, tenth ed. Academic Press.
- Stirzaker, D., 2005. *Stochastic Processes and Models*. Oxford University Press.
- Tuckwell, H., 1989. Stochastic processes in the neurosciences. In: CBMS-NSF Regional, Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Ullah, M., Wolkenhauer, O., 2011. *Stochastic Approaches for Systems Biology*. Springer-Verlag.
- Van Kampen, N.G., 2007. *Stochastic Processes in Physics and Chemistry*, third ed. North Holland: Elsevier.
- Wilkinson, D.J., 2012. *Stochastic Modelling for Systems Biology*, second ed. Chapman and Hall/CRC Press.

Hidden Markov Models

Monica Franzese and Antonella Iuliano, Institute for Applied Mathematics "Mauro Picone", Napoli, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Hidden Markov models (HMMs), named after the Russian mathematician Andrey Andreyevich Markov, who developed much of relevant statistical theory, are introduced and studied in the early 1970s. They were first used in speech recognition and have been successfully applied to the analysis of biological sequences since late 1980s. Nowadays, they are considered as a specific form of dynamic Bayesian networks, which are based on the theory of Bayes. HMMs are statistical models to capture hidden information from observable sequential symbols (e.g., a nucleotidic sequence). They have many applications in sequence analysis, in particular to predict exons and introns in genomic DNA, identify functional motifs (domains) in proteins (profile HMM), align two sequences (pair HMM). In a HMM, the system being modelled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. A good HMM accurately models the real world source of the observed real data and has the ability to simulate the source. A lot of Machine Learning techniques are based on HMMs have been successfully applied to problems including speech recognition, optical character recognition, computational biology and they have become a fundamental tool in bioinformatics: for their robust statistical foundation, conceptual simplicity and malleability, they are adapted fit diverse classification problems. In Computational Biology, a hidden Markov model (HMM) is a statistical approach that is frequently used for modelling biological sequences. In applying it, a sequence is modelled as an output of a discrete stochastic process, which progresses through a series of states that are 'hidden' from the observer. Each such hidden state emits a symbol representing an elementary unit of the modelled data, for example, in case of a protein sequence, an amino acid. In the following sections, we first introduce the concepts of Hidden Markov Model as a particular type of probabilistic model in a Bayesian framework; then, we describe some important aspects of modelling Hidden Markov Models in order to solve real problems, giving particular emphasis in its use in biological context. To show the potentiality of these statistical approaches, we present the stochastic modelling of an HMM, defining first the model architecture and then the learning and operating algorithms. In this work we illustrate, as example, applications in computational biology and bioinformatics and, in particular, the attention is on the problem to find regions of DNA that are methylated or un-methylated (CpG-islands finding).

Stochastic Process

The basic idea of the process modelling is to construct a model of a process starting from a set of sequences of events typically generated by the process itself. Subsequently, the model could be also used to discover properties of the process, or to predict future events on the basis of the past history. From a general point of view, a model can be used for three main purposes: describing the details of a process, predicting its outcomes, or for classification purposes, i.e., predicting a single variable k , which takes values in a finite unordered set, given some input data $x = (x_1, \dots, x_n)$. Against the deterministic model, which predicts outcomes with certainty, with a set of equations that describe the system inputs and outputs exactly, a stochastic model represents a situation where uncertainty is present. In other words, it's a model for a process that has some kind of randomness. The word "stochastic" derives from the Greek and means *random* or *chance*. A deterministic model predicts a single outcome from a given set of circumstances. A stochastic process is a sequence of events, in which the outcome at any stage depends on some probabilities. It means that a stochastic model predicts a set of possible outcomes weighted by their likelihoods, or probabilities. In modelling stochastic processes the key role is played by time; in fact, the stochastic model is a tool for predicting probability distributions of potential outcomes by allowing a random variation in its inputs over time. A stochastic process is defined as a collection of random variables $X = \{X_t; t \in T\}$ defined on a common probability space, taking values in a common set S (the *state space*), and indexed by a set T , often either $\mathbb{N}$ or $[0, \infty)$ and thought of as *time* (discrete or continuous respectively) (Oliver, 2009).

Markov Processes and Markov Chains

Important classes of stochastic processes are Markov processes and Markov chains. A Markov process is a process that satisfies the Markov property (*memoryless*), i.e., it does not have any memory: the distribution of the next state (or observation) depends exclusively on the current state. Formally, a stochastic process $X(t)$ is a Markov process, if it has the following properties:

1. The number of possible outcomes or states is finite.
2. The probabilities are constant over time.
3. It satisfies *memoryless* property.

$$P\{X(t_{n+1}) = x_{n+1} | X(t_n) = x_n, \dots, X(t_1) = x_1\} = P\{X(t_{n+1}) = x_{n+1} | X(t_n) = x_n\}$$

for any choice of time instants t_i , con $i = 1, \dots, n$ where $t_j > t_k$ for $j > k$.

Markov chain is a specific Markov process with a finite or countable state-space. Considering a set of states, $S = \{s_1, \dots, s_r\}$, we describe Markov chain as a process that starts in one of these states and moves successively from one state to another. Each move is called *step*. If the chain is currently in state s_i , then it moves to state s_j at the next step with a probability denoted by p_{ij} ; this probability does not depend upon which states the chain was in before the current state. The probabilities p_{ij} are called transition probabilities and are defined as the probabilities that the Markov chain is at the next time point in state j , given that it is at the present time point at state i . The matrix P with elements p_{ij} is called the transition probability matrix of the Markov chain. Since the state space is countable (or even finite), we can use the integers Z or a subset such as Z_+ (non-negative integers), the natural numbers $N = \{1, 2, 3, \dots\}$ or $\{0, 1, 2, \dots, m\}$ as the state space. We refer to Markov chains as *time homogeneous* or having stationary transition probabilities. Then, the probability of the transition from i to j between time point n and $n + 1$ is given by conditional probability function $P(X_{n+1} = j | X_n = i)$. We will assume that the transition probabilities are the same for all time points so that there is no time index needed on the left hand side. Given that the process X_n is in a certain state, the corresponding row of the transition matrix contains the distribution of X_{n+1} , implying that the sum of the probabilities over all possible states equals one. The transition probability matrix (see Table 1) contains a conditional discrete probability distribution on each of its rows. Formally,

$$p_{ij} \geq 0, \forall i, j \in S$$

$$\sum_{j \in S} p_{ij} = 1, \forall i \in S$$

In general, we define Markov chain as k th-order Markov model, when the probability of X_{j+k} conditioned on all previous elements in the sequence is identical to the probability of X_{j+k} conditioned on the previous k elements only:

$$P(X_{j+k} | X_1, \dots, X_j) = P(X_{j+k} | X_{j+k-1}, \dots, X_j)$$

In particular, in zeroth-order Markov chain $k=0$. It means that the variables X_i are independent, i.e., $P(X_j | X_1, \dots, X_{j-1}) = P(X_j)$.

It is always possible to represent a time-homogeneous Markov chain by a transition graph. Then, any transition probability matrix P (see Table 1) can be visualized by a transition graph, where the circles are nodes and represent possible states s_i , while edges between nodes are the transition probabilities p_{ij} (see Fig. 1).

Markov chains are probabilistic models, which can be used for the modelling of sequences given a probability distribution and then, they are also very useful for the characterization of certain parts of a DNA or protein string given, for example, a bias towards the AT or GC content. DNA sequences consist of one of four possible bases $\{A, T, C, G\}$ at each position. Each sequence is always read in a specific direction, from the 5' to the 3' end. Each place in the sequence can be thought of as a state space, which has four

Table 1 Transition probability matrix

		Time t + 1				
State		S ₁	S ₂	S ₃	S ₄	Sum
Time t	S ₁	p ₁₁	p ₁₁	p ₁₂	p ₁₃	1
	S ₂	p ₂₁	p ₂₁	p ₂₂	p ₂₃	1
	S ₃	p ₃₁	p ₃₁	p ₃₂	p ₃₃	1
	S ₄	p ₄₁	p ₄₁	p ₄₂	p ₄₃	1

$\rho_{ij}=p(q_{t+1}=s_j|q_t=s_i)$

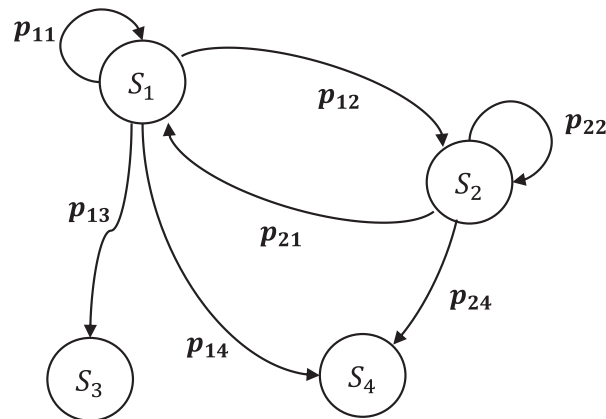


Fig. 1 Transition probability graph.

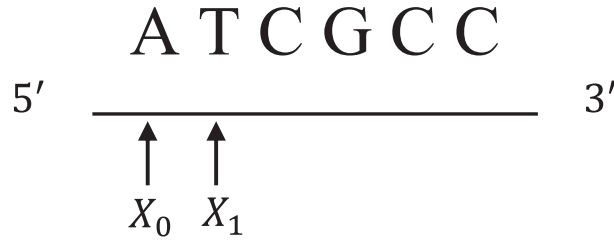


Fig. 2 Nucleotide sequence (5' to 3') as a first-order Markov Chain.

Table 2 Transition probability matrix for the ATCG sequence

A	0.3	0.2	0.2	0.3
T	0.1	0.2	0.4	0.3
C	0.2	0.2	0.2	0.4
G	0.1	0.8	0.1	0

possible states $\{A, T, C, G\}$, where each state of the nucleotide is sequentially dependent on the nucleotide adjacent and immediately upstream of it, and only that nucleotide. Each position in the sequence can be represented with a random variable $X_0, X_1, \dots, X_n$ that takes a value of one of the states in the state space for a particular place in the sequence. In a Markov Chain of zero order, the current state (or nucleotide) is totally independent of the previous state, so it's no memory and every state is untied. For the first order Markov Chain the case is different, because the current state actually depends only on the previous state. In the **Fig. 2** a first-order Markov chain is shown: the sequence goes in the 5' to 3' direction and $X_0=A, X_1=T$ and so forth. In this case, we can express the probability of this sequence using the Markov Property as follows:

$$P(X_5 = C | X_4 = C, X_3 = G, X_2 = T, X_1 = T, X_0 = A) = P(X_5 = C | X_4 = C)$$

and its transition probability matrix is shown in **Table 2**.

Hidden Markov Model

Now, we can define the Hidden Markov Models as probabilistic models, in which sequences are generated from two coexistent stochastic processes: the process of moving between states and the process of emitting an output sequence, characterized by Markov property and the output independence. The first one is a Markov model represents by a finite set of states, which generates the sequence of states of variables, specified by the initial state probabilities and state transition probabilities between variables; the second is characterized by the emission of one character of a given alphabet from each state, with a probability distribution that only depends from the state. The sequence of state transitions is a hidden process; it means that the variable states cannot be directly observed but they are observed through the sequences of emitted symbols, therefore the name *Hidden Markov Model*. Then, an Hidden Markov Model is defined by states, state probabilities, transition probabilities, emission probabilities and initial probabilities. They constitute the architecture of an HMM. Formalizing the definition, an HMM is a quintuple (S, V, π, A, B) , characterized by the following elements (Rabiner and Juang, 1986a):

- $S = \{S_1, \dots, S_N\}$ is the set of *states*, where N is the number of states. The triplet (S, π, A) represents a Markov chain; the states are hidden and we never observe them directly.
- $V = \{v_1, \dots, v_M\}$ is the *vocabulary*, the set of symbols that may be emitted.
- $\pi: S \rightarrow [0,1] = \{\pi_1, \dots, \pi_N\}$ is the *initial probability distribution* on the states. It gives the probability of starting in each state. We expect that

$$\sum_{s \in S} \pi(s) = \sum_{i=1}^N \pi_i = 1$$

- $A = (a_{ij})_{i \in S, j \in S}$ is the *transition probability* of moving from state S_i to state S_j . We expect that $a_{ij} \in [0,1]$ for each S_i and S_j , and that $\sum_{j \in S} a_{ij} = 1$, for each S_i .
- $B = (b_{ij})_{i \in V, j \in S}$ is the *emission probability* that symbol v_i is seen when we are in state S_i .

HMMs provide great help when there is the need of modelling a process in which there is not a direct knowledge about the state in which the system is. The key idea is that an HMM is a sequence “generator”. In general, we talk about emission of

observable events because we can think of HMMs as generative models that can be used to generate observation sequences. Algorithmically, a sequence of observations $O = o_1, \dots, o_T$, with $o_t \in V$ can be generated by an HMM, described by the algorithm in [Fig. 3](#) ([Rabiner and Juang, 1986a](#)). Two assumptions are made by the model. The first is called the Markov assumption and represents the memory of the model; it means that the current state is dependent only on the previous state; formally:

$$P(q_t | q_1^{t-1}) = P(q_t | q_{t-1})$$

The second is the independence assumption, i.e., the output observation at time t is dependent only on the current state and it is independent of previous observations and states:

$$P(o_t | o_1^{t-1}, q_1^t) = P(o_t | q_t)$$

A simple HMM for generating a DNA sequence is specified in [Fig. 4](#). In this model, state transitions and their associated probabilities are indicated by arrows; and symbol emission probabilities for A, C, G, T at each state are indicated below the state. For clarity, we omit the initial and final states as well as the initial probability distribution. For instance, this model can generate the state sequence given in [Fig. 5](#) and each state emits a nucleotide according to the emission probability distribution. The logical idea of an HMM suggests the potential ability of this approach for modelling problems in computational biology. In particular, [Baldi and Brunak \(1998a\)](#) define three main groups of problems in computational biology for which HMMs have been useful. The first problem is the multiple alignments of DNA sequences, which is more difficult using a dynamic programming approach. The second is to discover periodic sequences within specific regions of biological data from the knowledge of consensus patterns. The third is the problem to classify each nucleotide according to which structure it belongs. HMMs have also been used in protein

```

- Set  $t=1$ 
- Choose an initial state  $S_i(t)$ , according to the initial state probability distribution  $\pi$ 
- while  $t \leq T$  do
    Choose  $O_t = V_k$ , according to the emission probability  $b_{ik}$  in state  $S_i$ 
    if  $t \leq T$ 
        Transit to a new state  $S_j(t+1)$ , according to the transition probability
        distribution  $a_{ij}$  for state  $S_i$ 
        Set  $S_t = S_j$ 
    end if
    Set  $t = t + 1$ 
end while

```

Fig. 3 Generator algorithm for a sequence of observations by an Hidden Markov Model.

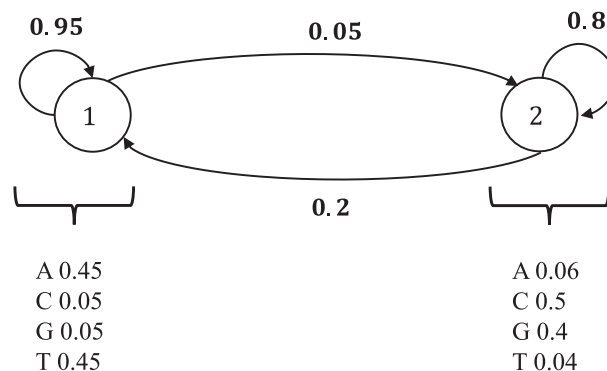


Fig. 4 A Hidden Markov model generator for DNA sequence.

State Sequence	...	1	1	1	2	2	2	1	1	...
Transition probabilities	?	0.95	0.95	0.05	0.8	0.8	0.2	0.95		
Observable Sequence	A	T	A	C	G	C	A	T		
Emission probabilities	0.45	0.45	0.45	0.5	0.4	0.5	0.45	0.45		

Fig. 5 Scheme for HMM emission probabilities and states.

profiling to discriminate between different protein families and predict a new protein's family or subfamily and in the problem of gene finding in DNA.

Statistical Inference in Hidden Markov Models

Once the architecture of an HMM has been decided, to analyze and describe data in almost all applications of HMMs, three distinct questions must to be solved:

1. What is the probability of an observed sequence according to a given HMM?
2. How to find the optimal state sequence that the HMM would use to generate the observed sequence?
3. How to find the structure and parameters of the HMM that best accounts for the data?

In general, in order to use the HMMs in application contexts, for example, in computational biology, to solve these questions, we need to be able to:

- 1) *Evaluate* the likelihood of the model given the observations, i.e., to compute the probability of the observation sequence for a model.
- 2) *Decode* the most likely state sequence given the observations, i.e., to find the optimal corresponding state sequence given the observation sequence and the model.
- 3) *Learning or training* to estimate the model parameters (initial probabilities, transition probabilities, and emission probabilities), that best explains the observation sequences given the model structure, describing the relationships between variables.

Evaluation Problem

Given a sequence of observations and an HMM, to solve the first problem, we would like to be able to compute the probability (or likelihood), $P(O|\lambda)$, that the observed sequence is produced by the model. This problem could be viewed as one of evaluating how well a model predicts a given observation sequence and thus allow us to choose the most appropriate model from a set. For this purpose, the forward-backward algorithm and the result can be used (Rabiner and Juang, 1986a; Stratonovich, 1960a). We consider the probability of the observations O for a specific state sequence Q

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t|q_t, \lambda) = b_{q_1}(o_1) \times b_{q_2}(o_2) \dots b_{q_T}(o_T)$$

and the probability of the state sequence

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

The join probability that O and Q occur simultaneously is simply the product of above two terms $P(O|Q, \lambda) P(Q|\lambda)$. Then, we can obtain the probability of the observations given the model by summing this join probability over all possible state sequences

$$P(O|\lambda) = \sum_Q P(O|Q, \lambda) P(Q|\lambda) = \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

To evaluate the last equation the forward-backward approach, that reduced complexity, is used. The key idea is to process a sequence, consider a forward or backward loop that processes it one element at a time. The sequence $X(1 \dots T)$ is broken into two parts, a "past" sequence $X(1 \dots t)$ and a "future" sequence $X(t+1 \dots T)$. In the Hidden Markov Model, each symbol emission and each state transition depend only on the current state; there is no memory of what happened before, no lingering effects of the past. This means that we can work on each half separately. Splitting the sequence into two parts, the inductive calculation on t can be used: if t advances from 1 towards T , it is called *forward* calculation, while if t is decremented down from T towards 1, it is called the *backward* calculation. In formal terms, the forward algorithm calculates the probability of being in a state i at time t and having emitted the output $o_1, \dots, o_t$. These probabilities are named the *forward variables*. By calculating the sum over the last set of forward variables, one obtains the probability of the model having emitted the given sequence. Both the forward algorithm and the backward algorithm involve three steps: initialization, recursion (or induction), and termination. We define the forward probability variable α as the probability of the partial observation sequence $o_1, \dots, o_t$ and state s_i at time t :

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = s_i | \lambda)$$

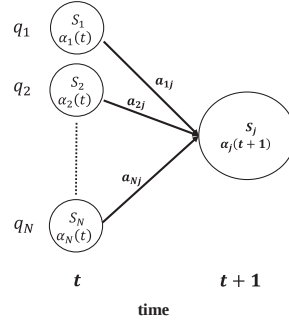


Fig. 6 The recursion step of the forward algorithm.

The forward algorithm is as follows:

1. *Initialization*

Calculate the forward probability at the first position $\alpha_1(i) = \pi_i b_i(o_1)$, $1 \leq i \leq N$

2. *Recursion*

Compute the forward probability

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N$$

The recursion step is the key to the forward algorithm (see Fig. 6). For each state s_i , $\alpha_j(t)$ stores the probability of arriving in that state, having observed the observation sequence up until time t .

3. *Termination*

When $i=N$, the forward recursion stops. Then, the probability of the whole sequence of observations can be found by summing the forward probabilities over all the states at the final variable

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

This approach reduces the complexity of calculations involved to N^2T rather than $2TN^T$. Similarly, backwards algorithm can be defined. It is the exact reverse of the forwards algorithm, with the backwards variable

$$\beta_t(i) = P(o_{t+1}o_{t+2}\dots o_T, q_t = s_i|\lambda)$$

as the probability of the partial observation sequence from $t+1$ to T , starting in state s_i .

Decoding Problem

The aim of decoding is to find the optimal state sequence associated with a given observation sequence. There are several possible ways to solve this problem. One possible optimality solution is proposed by the *Viterbi algorithm* (Viterbi, 1967a). The Viterbi algorithm uses a dynamic programming approach to find the most likely sequence of states Q given an observed sequence O and model λ . It works similarly to the forward algorithm. The goal is to get the most likely path through the HMM for a given observation; then, only the most likely transition from a previous state to the current one is important. Therefore, the transition probabilities are maximized at each step, instead of summed. To implement this solution, we define the variable

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = s_i, o_1, o_2, \dots, o_t|\lambda)$$

It is the probability of the most probable state path for the partial observation sequence. Corresponding to each node, two storage variables are used to cache the probability of the most likely path for the partial sequence of observations and the state at the previous variable to lead this path, denoted $\delta_t(i)$ and $\psi_t(i)$, respectively. The Viterbi algorithm consists of four steps: initialization, recursion, termination, and backtracking. It is described as follows:

1. *Initialization*

Calculate $\delta_1(i) = \pi_i b_i(o_1)$, $1 \leq i \leq N$ and set $\psi_1(i) = 0$

2. *Recursion*

Calculate

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N$$

and record the state

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N$$

3. Termination

The recursion ends when $i=N$. The probability of the most likely path is found by

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

The state of this path at variable N is found by

$$q^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

4. Backtracing

The last observation frame at time T is needed in order to decode the global optimal state path from T back to the first time index. The state of the optimal path at variable i is found by

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1$$

The backtracking allows the best state sequence to be found from the back pointers stored in the recursion step.

Learning Problem

How can we adjust the HMM parameters in a way that a given set of observations, called *training set*, is represented by the model in the best way for our purposes? Training involves adjusting the transition and output probabilities until the model sufficiently fits the process. These adjustments are performed using techniques to optimize $P(O|\lambda)$, the probability of observed sequence $o_1, \dots, o_T$, given model λ over a set of training sequences. How do we learn the parameters of an HMM from observations? Depending on the application, the “quantity” that should be optimized during the learning process differs. There isn’t known way to analytically solve this problem. Therefore iterative procedures or gradient techniques for optimization can be used. Here we will only present the iterative procedure. Therefore, we can choose an HMM such that $P(O|\lambda)$ is locally maximized. In literature, we can find several optimization criteria for learning. We present an iterative procedure, the Baum-Welch or Expectation Maximization (EM) method (Rabiner and Juang, 1986a; Baldi and Brunak, 1998a), which adapts the transition and output parameters by continually estimating these parameters until $P(O|\lambda)$ has been locally maximized. The EM algorithm is an iterative method to find the maximum likelihood estimate (MLE). It will let us train both the transition probabilities A and the emission probabilities B of the HMM. It works by computing an initial estimate for the probabilities, then using those estimates to compute a better estimate, and so on, iteratively improving the probabilities that it learns. This iterative algorithm alternates between performing an expectation step (E-step) and a maximization step (M-step). In an E-step, the expected number of times each transition is computed (the expected log-likelihood) and emission is used for the training set. In an M-step, the transition and emission parameters, that maximize the expected log-likelihood in the E-step are updated, using re-estimation formulas. The EM algorithm includes three steps of initialization, a series of iterations, and termination. In order to describe how to re-estimate HMM parameters, we first define the probability $\xi_t(i, j)$ of being in state S_i at time t and in state S_j at time $t+1$, as following:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

1. Iteration

Each cycle of EM iteration involves two steps, an E-step followed by an M-step, alternately optimizing the log-likelihood with respect to the posterior probabilities and parameters, respectively. In E-step we calculate the posterior probability of latent data using the current estimate for the parameters (π, α, β) of the model λ at time t . To perform the E-step, the expected log-likelihood function is maximized under the current estimated parameters and is computed, by using the estimated posterior probabilities of hidden data. In M-step: the new parameters that maximize the expected log-likelihood found in the E-step are estimated. From the definitions of forward and backward variables, we can calculate $\xi_t(i, j)$, which represents the posterior state probabilities of a variable and of the state combinations of two adjacent variables. We can write

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

where the numerator term is $P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$ and $P(O|\lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$ is the proper normalization factor for $\xi_t(i, j)$. We need to introduce another auxiliary variable, $\delta_t(i) = P(q_t = S_i | O, \lambda)$. It is the probability of being in the state S_i at time i , given the observation sequence and the model. In forward and backward variables this can be expressed by

$$\delta_t(i) = \frac{\alpha_t(i) \beta_{t+1}(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

Then, we can see that the relationship between $\delta_t(i)$ and $\xi_t(i, j)$ is given by

$$\delta_t(i) = \sum_{j=1}^N \xi_t(i, j), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T$$

If we sum over $\delta_t(i)$ over the time index t , we obtain the expected number of transitions from state S_i , while, summing $\xi_t(i, j)$ over the time, we obtain the expected number of transitions from state S_i to state S_j . Then, we can use again the Baum-Welch method to re-estimate the HMM parameters as follows

$$\begin{aligned} \bar{\pi}_i &= \text{expected frequency in state } S_i \text{ at time } t = 1 \\ \bar{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} \\ \bar{b}_i(k) &= \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} \end{aligned}$$

2. Termination

If we apply this procedure iteratively, (Baum and Sell, 1968a; Baum et al., 1970a) using $\bar{\lambda}(\bar{\pi}, \bar{a}, \bar{b})$ in place of λ , and we repeat the calculation of the parameters, we can improve that the probability of O being observed from the model until convergence is reached. The final result of this re-estimation procedure is called maximum likelihood estimate of the HMM. The re-estimation parameters can be derived by maximizing the auxiliary function

$$Q(\lambda, \bar{\lambda}) = \sum_Q P(Q|O, \lambda) \log [P(O|Q, \bar{\lambda})]$$

over λ . The maximization of $Q(\lambda, \bar{\lambda})$ leads to increased likelihood

$$\max_{\bar{\lambda}} Q(\lambda, \bar{\lambda}) \Rightarrow P(O|\bar{\lambda}) \geq P(O|\lambda)$$

Case Study

We consider an application of HMMs to the finding problem of the CpG islands in a DNA sequence (Ron, Lecture Notes). For this purpose, we define CpG islands as regions of DNA with a high frequency of CpG sites, where a cytosine nucleotide is followed by a guanine nucleotide in the linear sequence of bases along its 5' → 3' direction. The CG pair of nucleotides is the most infrequent dinucleotide in many genomes. This is because cytosines C in CpG dinucleotides are vulnerable to a process, called methylation, that can change with a high chance in mutating to a T. The methylation process is suppressed in areas around genes and hence these areas contain a relatively high concentration of the CpG dinucleotide. Such regions are called CpG islands, whose length varies from few hundreds to few thousands bases, with a GC content of greater than 50% and a ratio of observed-to-expected CpG number above 60%. The presence of a CpG island is often associated with the start of the gene (promoter regions) in the most mammalian genome and thus the presence of a CpG island is an important signal for gene finding. Then, we consider an HMM for detecting CpG islands in a DNA sequence. In this case, the model contains eight states corresponding to the four symbols of the alphabet $\Sigma = \{A, C, G, T\}$, where the states and emitted symbols are shown in Fig. 7 and we consider two possible conditions: a DNA sequence is a CpG island (labeled +) or a DNA sequence is non-CpG island (labeled -). Considering a DNA sequence $X = (x_1, \dots, x_L)$, the question is to decide whether X is a CpG island.

Results

In the case study we ask if given a short sequence, is it from a CpG island or not. To answer this question, we can estimate the transition probabilities from statistical data about CpG islands and non-CpG islands and then we can build two Markov chains, one for each. Then given a sequence, we compute the probability p of obtaining the sequence in the CpG island Markov chain, and the probability q of obtaining the sequence in the non-CpG island Markov chain. The odds ratio or log-odds ratio of these two

State:	A ⁺	C ⁺	G ⁺	T ⁺	A ⁻	C ⁻	G ⁻	T ⁻
Emitted Symbol:	A	C	G	T	A	C	G	T

Fig. 7 Hidden Markov Model for identification of CpG islands and non-CpG islands.

Table 3 Transition probability matrix for CpG islands (+) and non-CpG islands (−)

		A	C	G	T
p^+	A	0.18	0.27	0.43	0.12
	C	0.17	0.37	0.27	0.19
	G	0.16	0.34	0.37	0.13
	T	0.08	0.36	0.38	0.18
		A	C	G	T
p^-	A	0.30	0.20	0.29	0.21
	C	0.32	0.30	0.08	0.30
	G	0.25	0.25	0.29	0.21
	T	0.18	0.24	0.29	0.29

Table 4 Transition probability matrix in the CpG islands for HMM

$a_{\pi_i, \pi_{i+1}}$	A ⁺	C ⁺	G ⁺	T ⁺	A [−]	C [−]	G [−]	T [−]
A ⁺	0.18 <i>p</i>	0.27 <i>p</i>	0.43 <i>p</i>	0.12 <i>p</i>	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
C ⁺	0.17 <i>p</i>	0.37 <i>p</i>	0.27 <i>p</i>	0.19 <i>p</i>	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
G ⁺	0.16 <i>p</i>	0.34 <i>p</i>	0.37 <i>p</i>	0.13 <i>p</i>	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
T ⁺	0.08 <i>p</i>	0.36 <i>p</i>	0.38 <i>p</i>	0.18 <i>p</i>	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$	$\frac{1-p}{4}$
A [−]	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.30 <i>q</i>	0.20 <i>q</i>	0.29 <i>q</i>	0.21 <i>q</i>
C [−]	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.32 <i>q</i>	0.30 <i>q</i>	0.08 <i>q</i>	0.30 <i>q</i>
G [−]	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.25 <i>q</i>	0.25 <i>q</i>	0.29 <i>q</i>	0.21 <i>q</i>
T [−]	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	$\frac{1-q}{4}$	0.18 <i>q</i>	0.24 <i>q</i>	0.29 <i>q</i>	0.29 <i>q</i>

probabilities can be used to determine whether the sequence is coming from a CpG island or not. Then, for the CpG island Markov chain, we estimate a_{ij}^+ as follows

$$a_{ij}^+ = \frac{c_{ij}^+}{\sum_k c_{ik}^+}$$

where c_{ij}^+ is the number of times nucleotide j follows nucleotide i in the sequences labeled +. For the non-CpG island Markov chain, a_{ij}^- is estimated in a similar way. Now given a sequence X , we can compute $p(x)$ for each Markov chain; denoted these by $p(x|+)$ and $p(x|-)$, we can use the log-odds ratio $\log \frac{p(x|+)}{p(x|-)}$ to determine if X is coming from a CpG island or not; in fact, if $\log \frac{p(x|+)}{p(x|-)} > 0$, the sequence X is coming from a CpG island. Assuming that the transitions from the start state and to the end state are the same in both cases, the log-odds ratio can be expressed as following:

$$\log \frac{p(x|+)}{p(x|-)} = \log \frac{\prod_{i=0}^n a_{x_i x_{i+1}}^+}{\prod_{i=0}^n a_{x_i x_{i+1}}^-} = \sum_{i=1}^{n-1} \log \frac{a_{x_i x_{i+1}}^+}{a_{x_i x_{i+1}}^-}$$

We consider, as example, the short sequence CGCG; the transition probabilities for each of the two chains shown in **Table 3**. From the **Table 3**, we note that for the '+' Markov chain (CpG islands), the transition probabilities to C and G are higher. The log-odds ratio for this sequence is $\log \frac{0.27}{0.08} + \log \frac{0.34}{0.25} + \log \frac{0.27}{0.08} > 0$. Therefore, in this case, CGCG is coming from a CpG island. Instead, to verify if given a long sequence, does it contain a CpG island or not, we can incorporate both models (CpG islands and non-CpG islands) into one model. Then, we build a single Markov model consisting of both chains (+) and (−) described above as sub-chains, and with small transition probabilities between the two sub-chains. As before, we can estimate the transition probabilities between the two sub-chains by relying on known annotated sequences with all their transitions between CpG and non-CpG islands. is that there it not a one-to-one correspondence between the states and the symbols of the sequence. For instance, the symbol C can be generated by both states C^+ and C^- . For our purpose, we use an HMM (see **Fig. 7**): a sequence $X=(x_1, \dots, x_L)$ does not uniquely determine the path in the model and the states are hidden in the sense that the sequence itself does not reveal how it is generated. In this model, we define the probability for staying in a CpG island as p and the probability of staying outside as q , then the transition probabilities can be described in **Table 4**, derived from the transition probabilities given in **Table 3** under the assumption that we lose memory when moving from/into a CpG island, and that we ignore background probabilities. In this particular case, the emission probability of each state X^+ or X^- is exactly 1 for the symbol X and 0 for any other symbol.

Conclusions

HMM methods are used to solve a variety of biological problems, such as, for example, gene prediction, protein secondary structure prediction; in the last years, HMM-based profiles was applied to the protein-structure prediction and large-scale genome sequence analysis. In this paper we have presented an introduction to statistical approach of an HMM to show as these methods provide a conceptual framework for building complex models, just by drawing an intuitive picture. In particular, we have described the three problems of this theory with applications to the problem of finding specific patterns in biological sequences.

See also: Deep Learning. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics. Nonlinear Regression Models. Stochastic Processes

References

- Baldi, P., Brunak, S., 1998a. *Bioinformatics – The Machine Learning Approach*. Massachusetts Institute of Technology.
- Baum, L.E., Petrie, T., Soules, G., Weiss, N., 1970a. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.* 41 (1), 164–171. doi:10.1214/aoms/1177697196.
- Baum, L.E., Sell, G.R., 1968a. Growth transformations for functions on manifolds. *Pacific J. Math.* 27 (2), 211–227.
- Oliver, K., 2009. *Probability Theory and Stochastic Processes With Applications* Paperback. Overseas Press India Private Limited: New Delhi
- Rabiner, L., Juang, B., 1986a. An introduction to hidden Markov models. *IEEE ASSP Magazine* 3 (1), 4–16.
- Shamir, R. Lecture notes: Algorithms in molecular biology – Hidden Markov models. Tel Aviv University – Blavatnik School of Computer Science. Available at: <http://www.cs.tau.ac.il/~rshamir/>.
- Stratonovich, R., 1960a. Conditional Markov processes. *Theory of Probability & Its Applications* 5 (2), 156–178.
- Viterbi, A.J., 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. In: *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269.

Further Reading

- Chung, K.L., 1998a. *Markov Chains With Stationary Transition Probabilities*, second ed. Berlin: Springer-Verlag.
- Doob, J.L., 1953. *Stochastic Processes: Stochastic Modelling for Systems Biology*. New York, NY: John Wiley & Sons.
- Howard, R.A., 1971a. *Dynamic Probabilistic Systems*, vol. 1. New York, NY: John Wiley and Sons.
- Taylor, H.M., Samuel, K., 1998. *An Introduction to Stochastic Modeling*, third ed. San Diego, CA: Academic Press.
- Revuz, D., 1984a. *Markov Chains*, second ed. Amsterdam: North-Holland.

Linkage Disequilibrium

Barbara Calabrese, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Throughout the human genome, a correlation structure exists across genetic variation of different loci. Such a correlation structure means that knowing the genotype at one locus might make available information about the genotype at a second locus. This correlation between variation at different loci is named linkage disequilibrium (LD) (Neale, 2010). Linkage disequilibrium refers to the non-random association of alleles at two or more loci in a general population. Linkage disequilibrium between two alleles is related to the time of the mutation events, genetic distance, and population history. Specifically, it can provide valuable information in locating disease variants from marker data.

In the following, linkage disequilibrium analysis is defined and the main methods and software tools for LD estimation are presented.

Linkage Disequilibrium

Consider two linked loci Locus 1 has alleles $A_1, A_2, \dots, A_m$ occurring at frequencies $p_1, p_2, \dots, p_m$ and Locus 2 has alleles $B_1, B_2, \dots, B_n$ occurring at frequencies $q_1, q_2, \dots, q_n$ in the population. The possible haplotypes can be denoted as $A_1B_1, A_1B_2, \dots, A_mB_n$ with frequencies $h_{11}, h_{12}, \dots, h_{mn}$. The two linked loci are said to be in linkage equilibrium (LE), if the occurrence of allele A_i and the occurrence of allele B_j in a haplotype are independent events. Conversely, alleles are in linkage disequilibrium (LD) when they do not occur randomly. Under linkage disequilibrium, haplotypes do not occur at the frequencies expected when the alleles were independent. Positive linkage disequilibrium exists when two alleles occur together on the same haplotype more often than expected, and negative LD exists when alleles occur together on the same haplotype less often than expected (Barnes, 2007).

Several linkage disequilibrium measures have been defined. The Linkage Disequilibrium Coefficient D is one measure of LD. For biallelic loci with alleles A and a at locus 1; B and b at locus 2, D is defined as (Eq. 1):

$$D_{AB} = P_{(AB)} - P_{(A)}P_{(B)} \quad (1)$$

LD is a property of two loci, not their alleles. Thus, the magnitude of the coefficient D is important and it does not depend on the choice of alleles. The range of values the linkage disequilibrium coefficient can take on varies with allele frequencies. The calculation of this parameter is quite simple, but it is very sensitive to allele frequencies at the extreme values of 0 to 1. Lewontin (1964) suggested D_0 , a normalized D calculated by dividing D by its theoretical maximum for the observed allele frequencies. D_0 can assume values from -1 to $+1$, but, generally, its absolute value is used.

Another measure of linkage disequilibrium is r^2 which is equivalent to the Pearson correlation coefficient (Pritchard and Przeworski, 2001). Define a random variable X_A to be 1 if the allele at the first locus is A and 0 if the allele is a. Define a random variable X_B to be 1 if the allele at the second locus is B and 0 if the allele is b. Then the correlation between these random variables is (Eq. 2):

$$r_{AB}^2 = \frac{D_{AB}^2}{p_A(1-p_A)p_B(1-p_B)} \quad (2)$$

Other methods for LD estimation include more sophisticated methods, such as the maximum likelihood methods (Gomez-Raya, 2012; Feder et al., 2012).

Tools for LD Estimation

Numerous software packages facilitate LD estimations (Henry et al., 2014). LDHub (Zheng et al., 2017) automates the linkage disequilibrium score regression analysis pipeline. LD Hub calculates the single nucleotide polymorphisms (SNP) heritability for the uploaded phenotype(s), and a genetic correlation matrix across traits. It can be considered as a useful hypothesis generating tool, providing an easy method of screening hundreds/thousands of traits for interesting genetic correlations that could subsequently be followed up in further detail by other approaches such as pathway analysis.

Overall LD (Zhao et al., 1999) offers a platform for estimating linkage disequilibrium (LD) between both markers and groups of markers. Overall LD is a standalone software with the aim of providing a permutation-based assessment based on a measure of the overall deviation from random association. It was tested with simulated data.

gwasMP (Wu et al., 2017) queries users' results with different thresholds of physical distance and/or Linkage Disequilibrium (LD). gwasMP is an online method that quantifies the physical distance, LD and difference in Minor Allele Frequency (MAF)

between the top associated single-nucleotide polymorphism (SNPs) identified from genome wide association studies (GWAS) and the underlying causal variants. The results are from simulations based on whole-genome sequencing data.

LDx (Feder *et al.*, 2012) is a computational tool for estimating linkage disequilibrium (LD) from pooled resequencing data.

Haploview comprehensive suite of tools for haplotype analysis for a wide variety of dataset sizes (Barrett, 2009; Barrett *et al.*, 2005). Haploview generates marker quality statistics, LD information, haplotype blocks, population haplotype frequencies and single marker association statistics in a user-friendly format. All the features are customizable and all computations performed in real time, even for datasets with hundreds of individuals and hundreds of markers.

Ldlink allows users to query pairwise linkage disequilibrium (LD) between single nucleotide polymorphisms (SNPs) (Machiela and Chanock, 2015; Machiela and Chanock, 2017). Ldlink is a web-based LD analysis tool providing access to several bioinformatics modules. The software integrates expanded population reference sets, updated functional annotations, and interactive output to explore possible functional variants in high LD. It can facilitate mapping of disease susceptibility regions and assist researchers in characterizing functional variants based on genotype-phenotype associations with potential clinical utility.

LDetect is a method to identify approximately independent blocks of linkage disequilibrium (LD) in the human genome (Berisa and Pickrell, 2016). These blocks enable automated analysis of multiple genome-wide association studies.

LDheatmap (Shin *et al.*, 2006) produces a graphical display, as a heat map, of measures of pairwise linkage disequilibria (LD) between single nucleotide polymorphisms (SNPs). From a data set that provides information on pairwise LD between SNPs in a genomic region, it plots color-coded values of the pairwise LD measurements and returns an object containing several components. The package also contains two functions which can be used to highlight or mark with a symbol pairwise LD measures on an LD heat map.

Ld.pairs (Schaid, 2004) calculates composite measures of linkage disequilibrium (LD). *ld.pairs* determines the variance, covariance and statistical tests for all pairs of alleles from two loci when linkage phase is not known. This application is based on the works of Weir and Cockerham on multiallelic loci. *ld.pairs* permits to allow more than two alleles at either of the loci, and so this general composite statistics is a competitor to the traditional likelihood-ratio statistic.

The main technical features of the cited tools have summarized in the following:

<i>Tool</i>	<i>Programming language</i>	<i>Interface</i>	<i>Operating System</i>
LDHub	Python	Command line	Unix/Linux
Overall LD	–	Command line	Unix/Linux
Haploview	Java	Graphical	Unix/Linux, MAC OS, Windows
Ldlink	–	Web	–
Ldetect	Python	Command line	Unix/Linux
LDHeatmap	R	Command line	Unix/Linux, MAC OS, Windows
Ld.pairs	R	Command line	Unix/Linux, MAC OS, Windows
gwasMP	–	Web	–
LDx	–	Command line	Unix Linux

Concluding Remarks

LD plays a fundamental role in gene mapping, both as a tool for fine mapping of complex disease genes and in proposed genomewide association studies. LD is also of interest for what it can reveal about evolution of populations. In this contribution, linkage disequilibrium has been formally defined. Different methods for linkage disequilibrium estimation have been introduced and discussed.

See also: Comparative Genomics Analysis. Detecting and Annotating Rare Variants. Gene Mapping. Genetics and Population Analysis. Genome Informatics. Genome-Wide Haplotype Association Study. Natural Language Processing Approaches in Bioinformatics. Population Analysis of Pharmacogenetic Polymorphisms. Quantitative Immunology by Data Analysis Using Mathematical Models. Single Nucleotide Polymorphism Typing

References

- Barnes, M.R., 2007. *Bioinformatics for Geneticists*. Wiley.
- Barrett, J.C., 2009. Haploview: Visualization and analysis of SNP genotype data. *Cold Spring Harb Protoc.* 10.
- Barrett, J.C., Fry, B., Maller, J., Daly, M.J., 2005. Haploview: Analysis and visualization of LD and haplotype maps. *Bioinformatics* 21 (2), 263–265.

- Berisa, T., Pickrell, J.K., 2016. Approximately independent linkage disequilibrium blocks in human populations. *Bioinformatics* 32 (2), 283–285.
- Feder, A.F., Petrov, D.A., Bergland, A.O., 2012. LDx: Estimation of linkage disequilibrium from high-throughput pooled resequencing data. *PLoS One* 7 (11), e48588.
- Gomez-Raya, L., 2012. Maximum Likelihood Estimation of Linkage Disequilibrium in Half-Sib Families. *Genetics* 191 (1), 195–213.
- Henry, V.J., *et al.*, 2014. OMICtools: An informative directory for multi-omic data analysis. *Database (Oxford)* 2014 (2014), bau069.
- Lewontin, R.C., 1964. The interaction of selection and linkage. I. General considerations; heterotic models. *Genetics* 49 (1), 49–67.
- Machiela, M.J., Chanock, S.J., 2015. LDlink: A web-based application for exploring population-specific haplotype structure and linking correlated alleles of possible functional variants. *Bioinformatics* 31 (21), 3555–3557.
- Machiela, M.J., Chanock, S.J., 2017. LDassoc: An online tool for interactively exploring genome-wide association study results and prioritizing variants for functional investigation. *Bioinformatics* 34 (5), 887–889.
- Neale, B.M., 2010. Introduction to linkage disequilibrium, the HapMap and imputation. *Cold Spring Harb Protoc.* 2010 (3), pdb.top74.
- Pritchard, J.K., Przeworski, M., 2001. Linkage Disequilibrium in Humans: Models and Data. *American Journal of Human Genetics* 69 (1), 1–14.
- Schaid, D.J., 2004. Linkage disequilibrium testing when linkage phase is unknown. *Genetics* 166 (1), 505–512.
- Shin, J.H., Blay, S., McNeney, B., Graham, J., 2006. LDheatmap: An R Function for Graphical Display of Pairwise Linkage Disequilibria Between Single Nucleotide Polymorphisms. *Journal of Statistical Software* 16.
- Wu, Y., Zheng, Z., Visscher, P.M., Yang, J., 2017. Quantifying the mapping precision of genome-wide association studies using whole-genome sequencing data. *Genome Biology* 18, 86.
- Zhao, H., Pakstis, A.J., Kidd, J.R., Kidd, K.K., 1999. Assessing linkage disequilibrium in a complex genetic system. I. Overall deviation from random association. *Ann Human Genet* 63, 167–179.
- Zheng, J., *et al.*, 2017. LD Hub: A centralized database and web interface to perform LD score regression that maximizes the potential of summary level GWAS data for SNP heritability and genetic correlation analysis. *Bioinformatics* 33 (2), 272–279.

Introduction to the Non-Parametric Bootstrap

Daniel Berrar, Tokyo Institute of Technology, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

The *bootstrap* is a computation-intensive data resampling methodology for assessing the accuracy of statistical estimates and for making statistical inferences about unknown population parameters (Efron, 1979, 1981, 1987; Efron and Tibshirani, 1986, 1993; Davison and Hinkley, 1997). A central question of inferential statistics is the following: How accurate is the sample statistic $\hat{\theta}$ as an estimator for an unknown population parameter θ ? The key to this question is the *sampling distribution* of $\hat{\theta}$. Suppose that we are interested in the population mean, μ . It is well known that the sample mean, $\bar{x}$, is the best point estimate for μ , and its sampling distribution can be analytically derived. However, for more intricate statistics (for example, performance measures of predictive models), it may be very difficult or even impossible to find the theoretical sampling distribution. Here, the bootstrap provides a solution.

In theory, to obtain the sampling distribution of our statistic of interest, we would need to draw all or infinitely many random samples from the population and compute the statistic for each sample. In practice, we (usually) have only one sample from the population. At the heart of the bootstrap is the idea that the available sample is a good estimate of the population of interest. So instead of drawing random samples from the population, we draw random samples from our estimate. These “samples from a sample” are called *bootstrap sets* or *bootstrap samples*. From each bootstrap sample, we then compute our statistic of interest. The spread and the shape of the distribution of these *bootstrap statistics* tell us something about the sampling distribution of the sample statistic, which provides useful information for making inferences about the population parameter. Now the etymological roots of the methodology also become clear: The name is derived from the phrase “to pull oneself up by one’s own bootstrap,” which implies that a thing is being built by using the thing itself – by drawing samples from the sample, we are building a bootstrap distribution.

There are essentially two different types of bootstraps. Without specific assumptions or a particular model for the population under investigation, the bootstrap is called *non-parametric*; otherwise, it is called *parametric* (Berrar and Dubitzky, 2013). Like random permutation methods and the jackknife, the bootstrap belongs to the family of Monte Carlo resampling methods (Baguley, 2012). Essentially, these methods differ with respect to how the sampling is done. Both the jackknife and random permutation methods perform sampling *without* replacement, whereas sampling is done *with* replacement in the bootstrap.

This article provides an introduction to the ordinary non-parametric bootstrap, which is arguably the most fundamental type. Here, we consider only the case that the population parameter and sample statistic are scalars. The focus is on the key ideas and two applications that are particularly relevant for bioinformatics: How to derive basic bootstrap confidence intervals, and how to assess the prediction error of a statistical model (for example, a classifier). For more theoretical details, see Efron and Tibshirani (1986, 1993), Dixon (2006). For an excellent discussion of the role of the bootstrap in statistics education, see Hesterberg (2015).

Notation

We begin with some basic terms, which largely follow the standard statistical notation (Davison and Hinkley, 1997). Greek lower case letters usually denote population parameters, for example, μ denotes the population mean. Roman upper case letters, such as X , usually denote sample statistics. For instance, $X = x_i$ means that the random variable X assumes a particular value x_i . The exception is N , which refers to the population size. $E(X)$ denotes the expected value or mean of X .

The probability distribution of X is called *sampling distribution* of X . The probability that a real-valued random variable X is smaller than, or equal to, a particular value x_0 is denoted by $P(X \leq x_0) = \mathcal{F}_X(x_0)$, which is called the *cumulative distribution function* (CDF) of X . The unknown parameter of interest is commonly denoted by θ , which could represent the mean or any other parameter. The symbol $\hat{\cdot}$ indicates an estimate. For example, the sample statistic $\hat{\theta}$ is an estimate for θ ; therefore, $\hat{\theta}$ is also called the *estimator* and θ is called the *estimand*. The standard deviation of the sampling distribution of the sample statistic $\hat{\theta}$ is called the *standard error* of $\hat{\theta}$. Finally, the symbol $*$ indicates a bootstrap statistic.

Note that a parameter has a fixed but (usually) unknown value and is therefore a constant. By contrast, a sample statistic is a random variable because it depends on the random makeup of the sample from which it was calculated. In the frequentist paradigm, sample statistics are random variables, and consequently, they have a sampling distribution; parameters, on the other hand, do not, since they are not random variables.

Sampling Distribution of a Sample Statistic

Let us assume that we have a data set D , which is a random sample from the population of interest. The data set has n elements, $x_1, x_2, \dots, x_n$. We will assume that the elements in our random sample are the real-valued outcomes of independent and identically distributed (iid) random variables $X_1, X_2, \dots, X_n$. The *probability density function* (PDF) of these random variables is denoted by f , and their cumulative distribution function is denoted by $\mathcal{F}$. Both f and $\mathcal{F}$ are characteristics of the population and therefore unknown,

just like the population parameter that we are interested in. We use the sample to make inferences about the unknown population parameter θ by calculating the statistic $\hat{\theta}$ from the sample data.

The population parameter can be thought of as a function of the population data, i.e., $\theta = t(x)$, where x denotes the population data and $t(\cdot)$ is a statistical function. This function tells us how to calculate θ from the data. For instance, let us assume that θ denotes the population mean, μ , and let us assume that the population elements are countably finite. Then $\mu = t(x) = \frac{1}{N} \sum_{i=1}^N x_i$, where N is the number of elements in the population. The same statistical function may be used to calculate the sample statistic, $\hat{\theta} = t(x)$, where x now denotes the sample data. For example, the sample mean is calculated as $\bar{x} = t(x) = \frac{1}{n} \sum_{i=1}^n x_i$, where n is the number of elements in the sample. In non-parametric analysis, the *empirical distribution function* (EDF), denoted by $\hat{\mathcal{F}}$, is an estimate of the cumulative distribution function, $\mathcal{F}$. Many simple statistics can be regarded as properties of the empirical distribution function (Davison and Hinkley, 1997). For example, let us consider the discrete case, where each x_i has the probability $\frac{1}{n}$ of being sampled. The sample mean, $\bar{x}$, is the same as the mean of the empirical distribution function, $\hat{\mathcal{F}}$,

$$\hat{\mathcal{F}}(x) = \frac{1}{n} \sum_{i=1}^n H(x - x_i), \quad \text{with} \quad (1)$$

$$H(u) = \begin{cases} 1, & \text{if } u \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where $H(\cdot)$ is the Heaviside function. Both the population parameter and the sample statistic can be expressed as a function of their underlying distribution functions: $\theta = t(\mathcal{F})$ and $\hat{\theta} = t(\hat{\mathcal{F}})$. The reason is that the parameter can be considered a characteristic of the population, which is described by its cumulative distribution function, $\mathcal{F}$. Similarly, the value of the sample statistic is a function of the empirical distribution of the sample elements, $\hat{\mathcal{F}}$. Not all estimators, however, are exactly of the form $\hat{\theta} = t(\hat{\mathcal{F}})$ (Davison and Hinkley, 1997); for example, the unbiased sample variance is $\frac{n}{n-1} t(\hat{\mathcal{F}})$. More precisely, therefore, it should be stated that $\hat{\theta} = t_n(\hat{\mathcal{F}})$ and $t_n \rightarrow t$ as $n \rightarrow \infty$. Here, we will ignore this detail, as it is irrelevant for the explanation of the non-parametric bootstrap.

The idea of random sampling from a population is illustrated in Fig. 1. As we have seen, the sample statistic, $\hat{\theta}$, can be thought of as a function of the empirical distribution, $\hat{\mathcal{F}}$, of the data in that particular sample. Clearly, if we drew another random sample, then most certainly we would obtain a slightly different value of $\hat{\theta}$. This idea is illustrated by $\hat{\theta}_2$ and $\hat{\theta}_k$ in Fig. 1. Assume that $t(\cdot)$ is the function for the arithmetic mean. It is plausible that sometimes, $\hat{\theta} = \bar{X}$ is a bit larger than $\theta = \mu$, sometimes a bit smaller, but on average, the sample mean and the population mean are the same. In fact, it can be proved that the expected value of the sample mean is the population mean, i.e., $E(\bar{X}) = \mu$.

Two characteristics of an estimator are its bias, β , and variance, $\sigma_{\hat{\theta}}^2$. The bias is the systematic error, i.e., the difference between the estimator's expected value, $E(\hat{\theta})$, and the true value of the estimand, θ , i.e., $\beta = E(\hat{\theta}) - \theta$. An estimator is said to be *unbiased* if its bias is zero. Hence, the sample mean $\bar{X}$ is an unbiased estimator of μ .

Consider a sample statistic $\hat{\theta}$ with mean $\mu_{\hat{\theta}} = \theta + \beta$ and standard deviation $\sigma_{\hat{\theta}}$, and assume that $\hat{\theta}$ is normally distributed, $\hat{\theta} \sim \mathcal{N}(\mu_{\hat{\theta}}, \sigma_{\hat{\theta}}^2)$. The probability that $\hat{\theta}$ takes on a value smaller than, or equal to, a particular value $\hat{\theta}_0$ is

$$P(\hat{\theta} \leq \hat{\theta}_0) = \Phi\left(\frac{\hat{\theta}_0 - \mu_{\hat{\theta}}}{\sigma_{\hat{\theta}}}\right) = \Phi\left(\frac{\hat{\theta}_0 - (\theta + \beta)}{\sigma_{\hat{\theta}}}\right) \quad (2)$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function. Furthermore,

$$P((\theta + \beta) - z_{1-\frac{\alpha}{2}} \cdot \sigma_{\hat{\theta}} \leq \hat{\theta} \leq (\theta + \beta) + z_{1-\frac{\alpha}{2}} \cdot \sigma_{\hat{\theta}}) = 1 - \alpha \quad (3)$$

where $z_{1-\frac{\alpha}{2}} = \Phi^{-1}(1 - \frac{\alpha}{2})$ is the quantile of the standard normal distribution for probability $1 - \frac{\alpha}{2}$. For example, if $\alpha = 0.05$, then $z_{0.975} = 1.96$. Rearranging Eq. (3) leads to the following identity $\hat{\theta}$ for the population parameter θ ,

$$P((\hat{\theta} - \beta) - z_{1-\frac{\alpha}{2}} \cdot \sigma_{\hat{\theta}} \leq \theta \leq (\hat{\theta} - \beta) + z_{1-\frac{\alpha}{2}} \cdot \sigma_{\hat{\theta}}) = 1 - \alpha \quad (4)$$

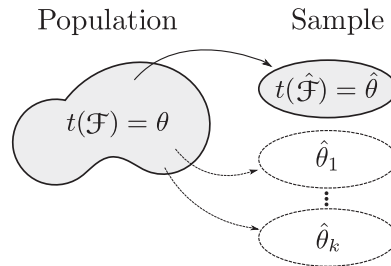


Fig. 1 Random sampling from a population with unknown parameter θ . The sample statistic $\hat{\theta}$ is the estimator for the parameter (i.e., the estimand). Both the statistic and the parameter depend on their underlying distribution functions $\hat{\mathcal{F}}$ and $\mathcal{F}$, respectively.

For example, let $\hat{\theta}$ be the sample mean, $\bar{X}$. It is well known that the sampling distribution of the sample mean is a normal distribution with mean $\mu\bar{X} = E(\bar{X}) = \mu$ and standard deviation $\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$, if the population has a normal distribution (with standard deviation σ) or if the sample size is sufficiently large (usually, $n \geq 30$) (Fig. 2). The sample mean is an unbiased estimator of the population mean, as $\beta = E(\bar{X}) - \mu = \mu - \mu = 0$. A $(1-\alpha) \times 100\%$ confidence interval for the population mean μ is now given by

$$\bar{x} \pm z_{1-\frac{\alpha}{2}} \cdot \frac{\sigma}{\sqrt{n}} \quad (5)$$

In practice, the population standard deviation is usually not known, and the confidence interval is calculated based on Student's t -distribution. The Student- t confidence interval for the population mean is calculated as

$$\bar{x} \pm t_{v, 1-\frac{\alpha}{2}} \cdot \frac{s}{\sqrt{n}} \quad (6)$$

where $t_{v, 1-\frac{\alpha}{2}}$ is the quantile of the t -distribution with v degrees of freedom, with $v = n - 1$, and $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ is the sample standard deviation.

So if we know the bias and variance of the sample statistic, then we can make inferences about the population parameter. For the sample mean, this is no problem, but what about more intricate statistics? Let us consider bias and variance as a function of the underlying distribution,

$$\beta = E(\hat{\theta}|\mathcal{F}) - t(\mathcal{F}) \quad (7)$$

$$\sigma\hat{\theta}^2 = \text{Var}(\hat{\theta}|\mathcal{F}) \quad (8)$$

Using $\hat{\mathcal{F}}$ as an estimate for $\mathcal{F}$, we obtain estimates of bias and variance,

$$\hat{\beta} = E(\hat{\theta}|\hat{\mathcal{F}}) - t(\hat{\mathcal{F}}) \quad (9)$$

$$\hat{\sigma}\hat{\theta}^2 = \text{Var}(\hat{\theta}|\hat{\mathcal{F}}) \quad (10)$$

which are called *bootstrap estimates* (Davison and Hinkley, 1997). This illustrates the *plug-in principle* (Hesterberg, 2015) of the bootstrap: When something is unknown, an estimate is plugged in instead.

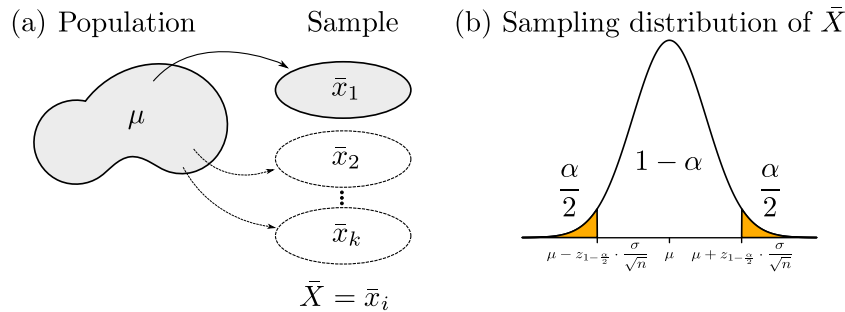


Fig. 2 (a) Random sampling from a population with unknown mean μ . The estimator for the population mean is the sample mean, $\bar{X}$. The sample mean is a random variable and takes on the values $\bar{x}_i$, which depend on the concrete makeup of the random sample. (b) Sampling distribution of the sample mean, $\bar{X}$. The distribution is normal, $\bar{X} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$, i.e., the distribution is centered at the population mean, μ . The value $z_{1-\frac{\alpha}{2}}$ is the quantile of the standard normal distribution for probability $1 - \frac{\alpha}{2}$. For example, $z_{0.975} = 1.96$ for $\alpha = 0.05$.

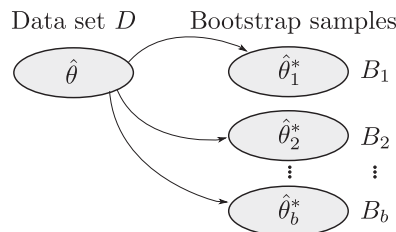


Fig. 3 Ordinary non-parametric bootstrap. The available data set (i.e., original sample) is repeatedly sampled (with replacement) to generate b bootstrap sets B_i , $i = 1..b$. Each bootstrap set has the same number of elements, n , as the original data set D . Because of the sampling with replacement, some elements can occur more than once in one bootstrap set, while others do not occur at all.

Ordinary Non-Parametric Bootstrap

The *ordinary non-parametric bootstrap* procedure can be described as follows:

1. The available data set D is assumed to be a representative sample from the population of interest. This data set contains a total of n elements, $x_1, x_2, \dots, x_n$.
2. From D , calculate the statistic $\hat{\theta}$, which is the estimate for the population parameter, θ .
3. Generate a bootstrap set, B , by randomly sampling n instances with replacement from D . The sampling is uniform, which means that each of the n elements in D has the same probability, $\frac{1}{n}$, of being selected. (More intricate sampling approaches, such as sampling with noise, are beyond the scope of this article.)
4. Repeat step (3) b times to generate b bootstrap sets, $B_1, B_2, \dots, B_b$ (Fig. 3).
5. Calculate the statistic $\hat{\theta}_i^*$ from the i th bootstrap set B_i .
6. Repeat step (5) for all b bootstrap sets.

From a data set with n elements, we could theoretically generate $\binom{2n-1}{n}$ distinct bootstrap sets. If n is not too large, we could perform an *exhaustive bootstrap* by constructing all possible sets instead of random sampling. This approach is also referred to as the *theoretical bootstrap* (Hesterberg, 2015). In practice, however, n is usually too large, and we generate several hundreds or thousands of bootstrap samples. This approach is an example of *Monte Carlo sampling*. Hesterberg suggests $b=1000$ for a rough approximation and $b \geq 10000$ for more accuracy (Hesterberg, 2015).

After the described procedure, we obtain the *empirical* or *bootstrap distribution* of all $\hat{\theta}_i^*$, from which we can infer the *bootstrap cumulative distribution function*, $\hat{\mathcal{F}}^*$. The mean, bias, and variance of $\hat{\theta}^*$ are:

$$\bar{\hat{\theta}}^* = \frac{1}{b} \sum_{i=1}^b \hat{\theta}_i^* \quad (11)$$

$$\text{bias}\{\hat{\theta}^*\} = E(\hat{\theta}_i^*) - \hat{\theta} = \bar{\hat{\theta}}^* - \hat{\theta} \quad (12)$$

$$\hat{\sigma}_{\hat{\theta}^*}^2 = \frac{1}{b-1} \sum_{i=1}^b (\hat{\theta}_i^* - \bar{\hat{\theta}}^*)^2 \quad (13)$$

It is important to note that the bootstrap distribution is centered at the observed statistic, $\hat{\theta}$, not at the population parameter, θ . Our best estimate for the population parameter (say, the mean μ) is therefore still our original sample statistic (i.e., $\bar{x}$ for μ), not the average of all bootstrap estimates, $\bar{\hat{\theta}}^*$. As the bootstrap distribution is not centered at the population parameter, the quantiles of the bootstrap distribution are usually different from the quantiles of the theoretical sampling distribution of the sample statistic, $\hat{\theta}$. This means that the bootstrap quantiles are not meaningful estimates for the quantiles of $\hat{\theta}$. However, the bootstrap quantiles are useful to estimate the quantiles and CDF of $\hat{\theta} - \theta$ and to estimate the standard deviation of $\hat{\theta}$. This idea is illustrated in Fig. 4 using the sample mean $\bar{X}$ as an example. Note that in Fig. 4(d), the standard deviation of the bootstrap statistic is only slightly larger than the standard deviation of the theoretical sampling distribution of the sample statistic. The example in Fig. 4 is just an illustration of the principle of bootstrapping. For the sample mean, the bootstrap is actually not needed because we already know the theoretical sampling distribution of $\hat{\theta}$. For many other statistics, however, that is not the case.

Basic Bootstrap Confidence Intervals

We now describe two basic bootstrap confidence intervals: The *bootstrap percentile interval* and the *bootstrap-t interval* (Efron and Tibshirani, 1993). More advanced intervals are described in Efron (1987), DiCiccio and Efron (1996).

Bootstrap Percentile Confidence Interval

The bootstrap percentile confidence interval is arguably the simplest and most intuitive bootstrap interval. It is derived as follows.

1. Generate b bootstrap sets by repeatedly sampling with replacement from the available data set (i.e., original sample) D . For each bootstrap sample B_i , calculate the sample statistic $\hat{\theta}_i^*$.
2. Find the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ percentiles of the distribution of $\hat{\theta}_i^*$. These percentiles are denoted by $\hat{\theta}_{\frac{\alpha}{2}}^*$ and $\hat{\theta}_{1-\frac{\alpha}{2}}^*$, respectively. (In R, the function `quantile(Y, c(0.025, 0.975))` gives the 2.5% and 97.5% percentile for the data in Y .)
3. A $(1-\alpha) \times 100\%$ bootstrap percentile confidence interval for θ is given by

$$\left[\hat{\theta}_{\frac{\alpha}{2}}^*, \hat{\theta}_{1-\frac{\alpha}{2}}^* \right] \quad (14)$$

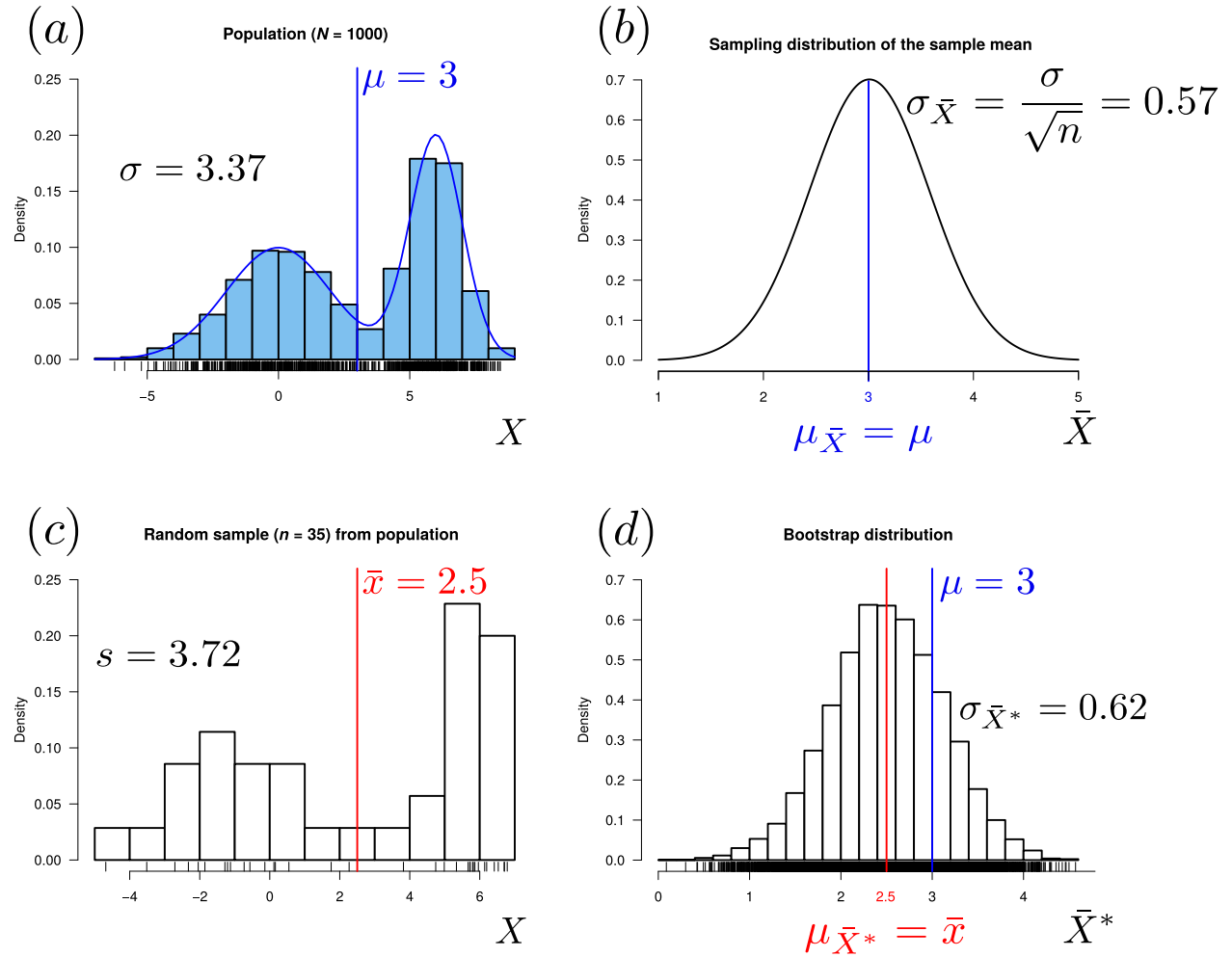


Fig. 4 (a) An example of a population of interest, representing $N=1000$ instances x_i from a normal mixture of two distributions, $\mathcal{N}_1(0, 4)$ and $\mathcal{N}_2(6, 1)$. The population mean and standard deviation are $\mu=3$ and $\sigma=3.37$, respectively. (b) The sampling distribution of the sample mean is normal, with mean $\mu_{\bar{X}} = \mu$ and standard deviation $\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = 0.57$. (c) A random sample of $n=35$ instances from the population. The sample mean is $\bar{x}=2.5$ and the standard deviation is $s=3.72$. (d) Bootstrap (empirical) distribution of the bootstrap estimate $\bar{X}^*$ based on $b=10000$ bootstrap sets sampled from the data in (c). The bootstrap distribution has mean $\mu_{\bar{X}^*} = \bar{x}=2.5$ and standard deviation $\sigma_{\bar{X}^*} = 0.62$.

Bootstrap- t Confidence Interval

The bootstrap- t interval is constructed as follows.

1. Draw b bootstrap samples by repeatedly sampling with replacement from the available data set D . For each bootstrap sample B_i , calculate

$$Z_i^* = \frac{\hat{\theta}_i^* - \hat{\theta}}{\widehat{\text{SE}}_i^*} \quad (15)$$

where $\widehat{\text{SE}}_i^*$ is the estimate of the standard error of $\hat{\theta}^*$ based on the data in B_i . This estimate is calculated as the standard deviation of all values in B_i divided by the square root of the sample size, $\widehat{\text{SE}}_i^* = \frac{\hat{\sigma}_i^*}{\sqrt{n}}$. The random variable Z_i^* is called an *approximate pivot* (Efron and Tibshirani, 1993).

2. Find the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ percentiles of the distribution of Z_i^* . Denote these percentiles by $\hat{t}_{\frac{\alpha}{2}}^*$ and $\hat{t}_{1-\frac{\alpha}{2}}^*$, respectively.
3. A $(1-\alpha) \times 100\%$ bootstrap- t interval for the parameter θ is given by

$$\left[\hat{\theta} - \hat{t}_{1-\frac{\alpha}{2}}^* \cdot \frac{s}{\sqrt{n}}, \hat{\theta} - \hat{t}_{\frac{\alpha}{2}}^* \cdot \frac{s}{\sqrt{n}} \right] \quad (16)$$

where s denotes again the standard deviation of the sample (i.e., the original data set D).

Table 1 95% confidence intervals for the population mean based on the random sample in Fig. 4(a)

Method	95% confidence interval	Width
Bootstrap percentile interval	[1.238, 3.696]	2.458
Bootstrap- <i>t</i> interval	[1.184, 3.769]	2.585
Student- <i>t</i> interval	[1.224, 3.780]	2.556
Normal- <i>z</i> interval	[1.387, 3.617]	2.230

Eq. (16) is similar to the common Student-*t* interval, except that the *t*-value is substituted by the bootstrap estimates $\hat{t}_{1-\frac{\alpha}{2}}^*$ and $\hat{t}_{\frac{\alpha}{2}}^*$. These estimates are not necessarily symmetric about 0, in contrast to the percentiles of the standard normal and *t*-distribution. Whereas the common *z*- and *t*-intervals are always symmetric about θ , bootstrap-*t* intervals (and bootstrap percentile intervals) are not necessarily symmetric.

It may be surprising that Eq. (16) does not include a term for the bias (cf. Eq. (4)). Indeed, we could calculate a bias-corrected estimate as $\hat{\theta} - \text{bias}\{\hat{\theta}^*\} = \hat{\theta} - (\hat{\theta}^* - \hat{\theta}) = 2\hat{\theta} - \hat{\theta}^*$. However, bias estimates can have a high variability (Efron and Tibshirani, 1993; Hesterberg, 2015), so it is not advisable to perform this correction.

Table 1 shows the two bootstrap 95% confidence intervals for the population mean μ of the example population (Fig. 4(a)). For comparison, the common *z*- and *t*-intervals are also shown.

For the example data in Fig. 4, $n=35$ seems to be sufficient for a reliable bootstrap confidence interval. The width of both bootstrap intervals is comparable to that of the Student-*t* interval. The *z*-interval is of course the most accurate interval in this example because it uses the population standard deviation, σ . For real-world data sets, however, it is usually not possible to derive a *z*-interval, as σ is almost never known.

Both the bootstrap-*t* interval and the bootstrap percentile interval are only first-order accurate, which means that the actual one-sided coverage probabilities differ from the nominal values by $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. Hence, both intervals tend to be too narrow for small n (Hesterberg, 2015). For small sample sizes, the bootstrap intervals offer therefore no improvement over the Student-*t* interval. On the other hand, when n is large, bootstrap intervals usually perform better, particularly for skewed distributions (Hesterberg, 2015). The reason is that common confidence intervals based on the standard normal and *t*-distribution are symmetric about zero, whereas bootstrap percentile intervals and bootstrap-*t* intervals may be asymmetric, which can improve coverage (Efron and Tibshirani, 1993). In fact, the Student-*t* interval was shown to be surprisingly inaccurate when the population distribution is skewed, even if the sample size is much larger than 30 (Hesterberg, 2011). For large samples from skewed populations, bootstrap-*t* intervals tend to have a better coverage than the Student-*t* intervals and are therefore preferable. The bootstrap-*t* interval is particularly suitable to location statistics such as the mean or median (Efron and Tibshirani, 1993).

When the bootstrap distribution is approximately normal and the bias is small, the bootstrap-*t* interval and the percentile bootstrap interval will agree closely. If that is not the case, Hesterberg et al. (2010) caution against the use of either interval.

Bootstrap Estimates of Prediction Error

When only one data set is available to develop and test a statistical model (for example, a classifier or regression model), the bootstrap is an effective technique to estimate the true prediction error (Berrar and Dubitzky, 2013). This true prediction error represents the unknown population parameter. Let us assume that a data set D contains n instances (or cases) x_i , $i=1..n$, and each instance is described by a set of features. Let us further assume that with each instance, exactly one target label y_i is associated. In the case of classification, y_i is a discrete class label, $y_i \in \{y_1, y_2, \dots, y_k\}$, where k indicates the number of distinct classes. In the case of regression, the target is a real value, $y_i \in \mathbb{R}$. A statistical model $f(\cdot)$ estimates the target y_i of the case x_i as $f(x_i) = \hat{y}_i$. The estimation error is quantified by a *loss function*, $\mathcal{L}(y_i, \hat{y}_i)$. For example, in the case of classification and the 0–1 loss function, the loss is 1 if $y_i \neq \hat{y}_i$ and 0 otherwise. In the case of a regression problem, the loss function generally involves a form of squared error. The *bootstrap estimate of the prediction error* is calculated as follows.

1. Generate b bootstrap sets B_j , $j=1..b$, by repeatedly sampling (with replacement) n cases from D .
2. Use the bootstrap set B_j as training set to build the model f_j^* .
3. Calculate $f_j^*(x_i) = \hat{y}_i$, and then compute the loss function $\mathcal{L}(y_i, f_j^*(x_i))$.
4. The *bootstrap estimate of the prediction error*, $\hat{\varepsilon}^*$, is the average over all cases and all bootstrap sets,

$$\hat{\varepsilon}^* = \frac{1}{n} \sum_{i=1}^n \frac{1}{b} \sum_{j=1}^b \mathcal{L}(y_i, f_j^*(x_i)) \quad (17)$$

As the training sets B_j partially overlap with D , the error estimate $\hat{\varepsilon}^*$ is biased downward, which means that it is smaller than the true prediction error. A simple approach to alleviate the optimistic bias is to exclude those cases from the evaluation that served

already as training cases. Let S_{-i} denote the set of indices of the bootstrap samples that do not contain the case x_i , and let $|S_{-i}|$ denote the number of these indices. The *leave-one-out bootstrap estimate*, $\hat{e}_{loob}^*$, is defined as follows (Hastie et al., 2008).

$$\hat{e}_{loob}^* = \frac{1}{n} \sum_{i=1}^n \frac{1}{|S_{-i}|} \sum_{j \in S_{-i}} \mathcal{L}(y_i, \hat{f}_j^*(x_i)) \quad (18)$$

It is possible that all bootstrap samples contain the case x_i , and consequently that $|S_{-i}|=0$. To avoid a division by zero, the number of bootstrap samples, b , has to be sufficiently large to ensure that at least one of them does not contain x_i , or those cases that appear in all bootstrap samples should be omitted from Eq. (18).

The leave-one-out bootstrap estimate tends to overestimate the true prediction error; in other words, it has an upward bias. The reason is that the model is trained on only a subset of the available data because each bootstrap sample contains, on average, only about 63% of the data: The probability that a case x_i is selected for a bootstrap set in one random sampling is $\frac{1}{n}$, and the probability of not being selected is therefore $1 - \frac{1}{n}$. When the sampling is done n times, the probability that the case is not selected at all is $(1 - \frac{1}{n})^n$. Consequently, the probability that it is selected is $1 - (1 - \frac{1}{n})^n$. Note that $(1 - \frac{1}{n})^n \rightarrow e^{-1}$ for $n \rightarrow \infty$. Therefore, for sufficiently large n , the probability that a case x_i appears in a bootstrap set is approximately $1 - e^{-1} = 0.632$. This means that each bootstrap set has, on average, about 63% of distinct cases only – so about 37% of the available data are not used for training, which results in an overestimation of the prediction error. This overestimation can be corrected by considering the resubstitution error. The *bootstrap resubstitution error* is defined as follows (Hastie et al., 2008).

$$\hat{e}_{resub}^* = \frac{1}{n} \sum_{i=1}^n \frac{1}{|S_{+i}|} \sum_{j \in S_{+i}} \mathcal{L}(y_i, \hat{f}_j^*(x_i)) \quad (19)$$

where S_{+i} is the set of bootstrap set indices that contain the case x_i . The resubstitution error is also referred to as the *training error*. It has a downward bias, which means that it underestimates the true prediction error.

The *0.632 bootstrap error*, $\hat{e}_{0.632}^*$, is a weighted average of the overly optimistic resubstitution error and the overly pessimistic leave-one-out bootstrap error,

$$\hat{e}_{0.632}^* = 0.368 \cdot \hat{e}_{resub}^* + 0.632 \cdot \hat{e}_{loob}^* \quad (20)$$

The error estimate $\hat{e}_{0.632}^*$ still has a downward bias (Breiman et al., 1984), which the *.632 + bootstrap* corrects by adjusting the weights for the resubstitution error and leave-one-out bootstrap error,

$$\hat{e}_{0.632+}^* = (1 - w) \cdot \hat{e}_{resub}^* + w \cdot \hat{e}_{loob}^*, \quad \text{with} \quad (21)$$

$$w = \frac{0.632}{1 - 0.368\hat{R}} \quad (22)$$

$$\hat{R} = \frac{\hat{e}_{loob}^* - \hat{e}_{resub}^*}{\hat{e}_{random} - \hat{e}_{resub}^*} \quad (23)$$

The quantity $\hat{e}_{random}$ in Eq. (23) is an estimate of the no-information error rate, i.e., the expected error when the cases are statistically independent of their class labels. $\hat{R}$ is a measure of the relative overfitting rate, which ranges from 0 if $\hat{e}_{loob}^* = \hat{e}_{resub}^*$ (i.e., there is no overfitting) to 1 if $\hat{e}_{loob}^* = \hat{e}_{random}$. The weight w ranges from 0.632 (if $\hat{R} = 0$) to 1 (if $\hat{R} = 1$), and consequently, $\hat{e}_{0.632+}^*$ ranges from $\hat{e}_{0.632}^*$ to $\hat{e}_{loob}^*$. Hence, the weights in the *0.632 + bootstrap* error are not fixed, but they depend on the estimated degree of overfitting.

Discussion

The bootstrap is an extremely versatile methodology for statistical inference. It can be applied to problems that are intractable for standard approaches. For example, consider the problem of assessing the discriminatory power of a classification model based on genomic profiling. Typically, only a relatively small data set is available for both training and testing the model. The observed classification performance may be measured by balanced accuracy, area under the ROC curve, or any other metric that is deemed suitable for the problem at hand. But the observed performance value is only an estimate of the true discriminatory power for new, unseen cases from the same population. How accurate is this estimate? The bootstrap enables us to address this question.

The bootstrap is not the only data resampling methodology for this purpose, though. Molinaro et al. (2005) compared different resampling techniques for estimating the prediction error in the context of the small- n -large- p problem, i.e., in problems where the number of cases (n) is much smaller than the number of features (p). Such problems are not at all uncommon in the life sciences, for example, in classification studies involving gene expression data. The bias of the *0.632 + bootstrap* was found to be comparable to that of leave-one-out cross-validation and 10-fold cross-validation (Molinaro et al., 2005; Simon, 2007). According to Isaksson et al. (2008), however, both cross-validation and bootstrapping are unreliable for estimating the true prediction error when the data set is small, and they recommend Bayesian intervals based on a holdout test set.

The bootstrap spares us theoretical analysis, but at the expense of considerable computational costs due to the required repeated sampling. On the other hand, computational costs become increasingly negligible with the availability of relatively cheap computing power. Most statistical software tools nowadays include functions for bootstrapping; for example, the widely used

language and environment R (R Core Team, 2017) provides the package `boot` (Canty and Ripley, 2017) for parametric and non-parametric bootstrapping. In particular, a variety of bootstrap confidence intervals can be calculated with the function `boot.ci`. The R package `resample` provides various resampling functions for bootstrapping, jackknifing, and random permutation testing (Hesterberg, 2015).

The bootstrap can also serve as a useful teaching tool in introductory statistics courses (Hesterberg, 2015). Concepts such as repeated random sampling, standard errors, etc. might be explained in a more accessible way if the instructor complements statistical theory and formulas with histograms of sampling distributions.

Closing Remarks

The bootstrap is a powerful methodology that allows statistical inferences for a wide range of problems that are extremely difficult or even impossible to tackle by other means. The bootstrap therefore plays an important role in the sciences, both in research and education. At the heart of the bootstrap is the idea that the available data set is an estimate of the population of interest, and that we can repeatedly take random samples from that estimate. But if the available data set is too small, it is unlikely to be a good estimate. Researchers need to keep in mind that no amount of resampling will ever be a panacea for the lack of data. Also, error estimates based on resampling are no substitute for independent validation studies using new data.

See also: Natural Language Processing Approaches in Bioinformatics

References

- Baguley, T., 2012. *Serious Stats: A Guide to Advanced Statistics for the Behavioral Sciences*. Palgrave Macmillan.
- Berrar, D., Dubitzky, W., 2013. Bootstrapping. In: Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H. (Eds.), *Encyclopedia of Systems Biology*. Springer, pp. 158–163.
- Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. *Classification and Regression Trees*. Chapman and Hall.
- Canty, A., Ripley, B., 2017. *Boot: Bootstrap R (S-Plus) Functions*. R Package Version 1.3-20. Available at: <https://CRAN.R-project.org/package=boot>.
- Davison, A., Hinkley, D., 1997. *Bootstrap Methods and Their Applications*. Cambridge University Press.
- DiCiccio, T., Efron, B., 1996. Bootstrap confidence intervals. *Statistical Science* 11 (3), 189–228.
- Dixon, P., 2006. Bootstrap resampling. In: El-Shaarawi, A., Piegorisch, W. (Eds.), *Encyclopedia of Environmetrics*. Wiley, pp. 212–220.
- Efron, B., 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* 7 (1), 1–26.
- Efron, B., 1981. Nonparametric standard errors and confidence intervals. *Canadian Journal of Statistics* 9 (2), 139–158.
- Efron, B., 1987. Better bootstrap confidence intervals. *Journal of the American Statistical Association* 82 (397), 171–185.
- Efron, B., Tibshirani, R., 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science* 1 (1), 54–75.
- Efron, B., Tibshirani, R., 1993. *An Introduction to the Bootstrap*. Chapman & Hall.
- Hastie, T., Tibshirani, R., Friedman, J., 2008. *The Elements of Statistical Learning*, second ed. New York/Berlin/Heidelberg: Springer.
- Hesterberg, T., 2011. Bootstrap. *Wiley Interdisciplinary Reviews: Computational Statistics* 3 (6), 497–526.
- Hesterberg, T., 2015. What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum. *The American Statistician* 69 (4), 371–386.
- Hesterberg, T., Moore, D., Monaghan, S., et al., 2010. Bootstrap methods and permutation tests. In: Moore, D., McCabe, G., Craig, B. (Eds.), *Introduction to the Practice of Statistics*, seventh ed. N.Y.: W.H. Freeman.
- Hesterberg, T., 2015. *Resample: Resampling Functions*, R Package Version 0.4. Available at: <https://CRAN.R-project.org/package=resample>.
- Isaksson, A., Wallman, M., Göransson, H., Gustafsson, M., 2008. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters* 29 (14), 1960–1965.
- Molinaro, A., Simon, R., Pfeiffer, R., 2005. Prediction error estimation: A comparison of resampling methods. *Bioinformatics* 21 (15), 3301–3307.
- R Core Team, 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, Available at: <https://www.R-project.org/>.
- Simon, R., 2007. Resampling strategies for model assessment and selection. In: Dubitzky, W., Granzow, M., Berrar, D. (Eds.), *Fundamentals of Data Mining in Genomics and Proteomics*. Springer, pp. 173–186.

Population-Based Sampling and Fragment-Based *De Novo* Protein Structure Prediction

David Simoncini, University of Toulouse, Toulouse, France and RIKEN, Yokohama, Japan
Kam YJ Zhang, RIKEN, Yokohama, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteins are among the most studied macromolecules in Biology and are central to the functioning of living organisms. They play a major role in many critical mechanisms such as signaling, transport, enzymatic reactions, and immune defense. The dysfunction or the over-expression of a protein can be responsible for various lethal diseases, such as cancer. The function of a protein is determined by its structure, which is determined by its sequence (Anfinsen, 1973). Understanding the relationship between the sequence and structure of proteins would help to unveil the mechanisms which underlie the functioning of living organisms, and allow designing drugs in order to regulate the expression of proteins in case of abnormal activity. In addition, the abilities of proteins to self-assemble and to serve as chemical reaction catalysts make proteins a first choice target for applications in bio-nanotechnologies (Voet *et al.*, 2014; Bale *et al.*, 2016) or enzyme design (Siegel *et al.*, 2010; Khersonsky *et al.*, 2011).

Protein structure prediction aims at determining the three-dimensional structure of a protein starting from its sequence. The methods for addressing this problem can be classified into two categories: homology modeling (template-based modeling) and *de novo* modeling. Homology modeling methods aim at identifying known protein structures with a similar sequence, relying on the observation that similar sequences generally fold into similar structures. The known structure is then used as a template to model the target protein. *De novo* modeling methods predict protein structures solely based on their amino acid sequences. While the main bottleneck for homology methods is the identification of a suitable template, the main difficulty for *de novo* methods is dealing with the astronomical size of the conformational search space.

As pointed out by Cyrus Levinthal in 1968, the number of degrees of freedom of proteins renders infeasible the exhaustive enumeration of all states (Levinthal, 1968). When dealing with *de novo* protein structure prediction, it is therefore crucial to devise strategies that allow exploring a meaningful portion of plausible states, such as the fragment-based approach, which has encountered great success in the last two decades as epitomized by the Rosetta macromolecular modeling software (Rohl *et al.*, 2004; Bradley *et al.*, 2005b; Leaver-Fay *et al.*, 2011). In this approach, small structural fragments, taken from proteins with known structures, are assembled together in order to construct a complete protein model. Using small structural fragments which are matched with the target protein sequence helps to reduce the search space size by discretization. However, with a reasonable setting using 25 fragments of 9 residues per overlapping sliding window, the size of the search space for a 100 residue long protein sequence is of the order of 10^{128} . With such a search space size, advanced sampling algorithms are needed. The discretization of the search space provided by the fragment-based approach facilitates the use of combinatorial optimization sampling methods, such as simulated annealing, and population-based techniques. Simulated annealing (Kirkpatrick *et al.*, 1983) is the default sampling algorithm in Rosetta. It is commonly used in hard optimization problems with rugged energy landscapes due to its well-known ability to escape local minima. *De novo* protein structure prediction typically requires the production of tens of thousands of models in order to be reliable. Parallel and distributed computer architectures are employed in order to perform the calculations in an acceptable time. Simulated annealing, being a single solution method, cannot share information between parallel search trajectories. Information sharing is one of the strength of population-based sampling methods such as Evolutionary Algorithms, which use it to enhance the exploration of the search space and to lower the risks of convergence toward local minima.

Evolutionary algorithms, and population-based meta-heuristics in general, are known to provide near-optimal solutions to many NP-hard optimization problems, such as *de novo* protein structure prediction. The scoring functions associated with protein structure prediction typically describe a rugged funnel-like landscape with many local minima. Combined with the huge size of the search space, this makes the protein structure prediction problem a suitable target for population-based meta-heuristics. Furthermore, NP-hardness proofs exist even for simplified protein structure prediction problem representations (Hart and Istrail, 1997). A variety of population-based sampling methods have been reported in the literature. Genetic algorithms (GA) certainly are the most well-known, and their application to protein structure prediction have been extensively studied in the last three decades. Simple lattice models, such as the well-known HP model proposed by Dill (1985), have been used to confirm the potential of GA for protein structure prediction and to design efficient variants (Unger and Moulton, 1993; Hoque *et al.*, 2005; Lin and Hsieh, 2009; Zhang *et al.*, 2010). However, even though these models are helpful in understanding the main driving forces behind protein folding and for devising new algorithms, they cannot be used to predict the 3D structure of real-world proteins. To this end, many studies have proposed GA variants working on more realistic representations of proteins, beginning in the mid-nineties (Dandekar and Argos, 1992, 1994, 1996; Sun, 1993; Pedersen and Moulton, 1995;) and continuing to the present (Sakae *et al.*, 2011; Saleh *et al.*, 2013; Olson and Shehu, 2013; Clausen and Shehu, 2014; Kandathil *et al.*, 2016; Garza-Fabre *et al.*, 2016). Besides GA, many other meta-heuristics have been

Table 1 Metaphorical comparison of the Theory of Evolution and EA

<i>Theory of Evolution</i>	<i>Evolutionary algorithms</i>
Environment	Problem
Species	Search space
Individuals	Solutions
Fitness	Objective function

proposed such as Hidden Markov Models (HMM) (Park, 2005; Hamelryck *et al.*, 2006; Zhao *et al.*, 2008; Shmygelska and Levitt, 2009; Zhao *et al.*, 2010) or Estimation of Distribution Algorithms (Simoncini *et al.*, 2012, 2017; Simoncini and Zhang, 2013).

Evolutionary Algorithms

Evolutionary algorithms (EA) are general population-based optimization methods. Their search space sampling mechanisms and dynamics are inspired by the Theory of Evolution (Darwin, 1859). The metaphor employed in EA is summarized in Table 1. Similar to individuals evolving inside a species according to their fitness in the environment, solutions in a EA evolve in the search space in order to optimize an objective function and answer a problem.

Any EA follows the general scheme depicted in Algorithm 1. First, the population μ is initialized (line 1) and the solutions are evaluated (line 2). Then a cycle starts during which, at each iteration, a subpopulation λ is selected (line 4). The subpopulation λ is used to generate new solutions with stochastic variation operators (line 5). Each solution in the subpopulation λ is evaluated (line 6). Finally, the population μ is partially or totally replaced by the newly generated population. The iterative cycle continues until a stop criterion is met (line 3).

Algorithm 1 Evolutionary algorithm

```

1:  $\mu \leftarrow \text{InitializePopulation}()$ 
2:  $\text{EvaluateFitness}(\mu)$ 
3: while  $\text{Continue}()$  do
4:    $\lambda \leftarrow \text{Selection}(\mu)$ 
5:    $\lambda \leftarrow \text{StochasticVariations}(\lambda)$ 
6:    $\text{EvaluateFitness}(\lambda)$ 
7:    $\mu \leftarrow \text{ReplacePopulation}(\mu, \lambda)$ 
8: end while

```

In this section, we give an overview of evolutionary algorithms and discuss the key parameters that influence the search trajectories. We will focus on three subclasses of evolutionary algorithms: genetic algorithms, estimation of distribution algorithms, and memetic algorithms. We conclude this section with one of the main issues one must address when using evolutionary algorithms: the balance between exploration and exploitation.

Key Elements of Evolutionary Algorithms

The key elements of an EA are the problem representation, the choice of stochastic variation operators, and the population selection and replacement strategy.

Problem representation. Coming back to the evolutionary metaphor, the link between phenotype and genotype transposes to the link between problem statement and problem encoding. The two main elements of the problem encoding are the representation of solutions and the design of the objective function. The choice of these elements have a critical impact on the size of the search space and the shape of the *fitness landscape*, which can be seen as a map of the search space, with hills and valleys representing local minima attraction basins. Solutions can be encoded in many different ways, the most common being vectors of bits or integers. The objective function is derived from the problem statement. It can be exact, as for the well-known Traveling Salesman Problem, or approximate and empirically defined, as for the protein structure prediction problem.

Stochastic variation operators. Inspired by the theory of evolution, stochastic variation operators can be divided into two classes: crossover operators and mutation operators. Crossover operators are functions that take two solutions as input, whereas mutation operators are functions that take a single solution as input. The idea behind crossover operators is to randomly combine two good solutions with the hope of capturing good features from each of them, thereby generating a better solution. Many different crossover operators exist, the most popular being the one point crossover illustrated in Fig. 1. The one point crossover operator assumes that the solutions are vectors. Its principle is to randomly select a cutting point in

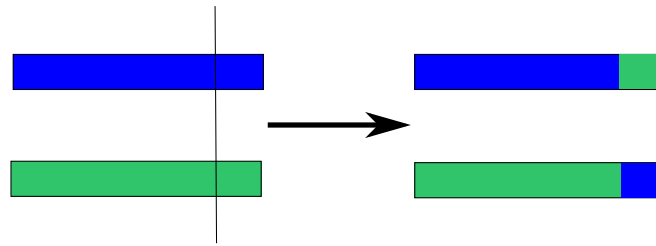


Fig. 1 Illustration of a one point crossover. A cutting point is arbitrarily chosen in the blue and green solutions. The part after the cutting point is swapped in order to create two new solutions.

order to delimit which part of the two solutions will be exchanged. Mutation operators typically randomly select a position in the solution and change its value. It can be seen as a neighborhood exploring move, whereas the crossover operation can be seen as a big jump in the fitness landscape.

Selection and replacement strategy. The way solutions are selected for recombination, and the way the new generation of solutions replaces the old, impacts the search dynamics and the convergence of EA. The two main approaches are the steady-state and generational schemes. In steady-state EA, two solutions are selected for recombination/mutation and the resulting solution is directly inserted in the population if it satisfies the prerequisite conditions. In generational EA, some solutions are selected in order to form a transition population from which a pool of new solutions will be generated by recombination and mutation. The new pool of solutions replaces the population totally or partially, once again according to some conditions. The conditions for replacement are user-defined, the most common being an elitist scheme where a fraction of the best solutions are preserved.

Genetic Algorithms

Genetic Algorithms (GA) were introduced by John Holland in 1975 (Holland, 1975). As with any evolutionary algorithm, GA rely on a metaphor of the Theory of Evolution (see Table 1). As suggested by Charles Darwin, a species evolves and adapts to its environment by means of variation and natural selection (Darwin, 1859). Good solutions to a problem can be seen as individuals well adapted to their environment. The basic idea of GA thus consists in combining solutions (genetic crossover), using local moves (mutations) and renewing the population with the best solutions (natural selection). Algorithm 2 describes a typical GA process. This algorithm is derived from the general EA scheme presented in Algorithm 1 by specifying the stochastic variation operators as crossover and mutation operations.

Algorithm 2 Genetic algorithm

```

1:  $\mu \leftarrow \text{InitializePopulation}()$ 
2:  $\text{EvaluateFitness}(\mu)$ 
3: while  $\text{Continue}()$  do
4:    $\lambda \leftarrow \text{Selection}(\mu)$ 
5:    $\lambda \leftarrow \text{Crossover}(\lambda)$ 
6:    $\lambda \leftarrow \text{Mutation}(\lambda)$ 
7:    $\text{EvaluateFitness}(\lambda)$ 
8:    $\mu \leftarrow \text{ReplacePopulation}(\mu, \lambda)$ 
9: end while

```

Estimation of Distribution Algorithms

Estimation of distribution algorithms (EDA) were proposed by Muhlenbein and Paaß in 1996 (Muhlenbein and Paaß, 1996). Unlike GA and other EA, EDA does not use any crossover or mutation operators. Still, EDA shares enough similarities with EA to be considered to belong to this class. The general behavior of EDA is given in Algorithm 3. This algorithm follows the general EA scheme of Algorithm 1. The unique feature of EDA concerns the choice of the stochastic variation operators which do not explicitly apply recombination or mutation to generate solutions. A pool of good solutions is selected (line 4) and used to estimate their probability distribution (line 5). The population is then renewed by drawing solutions according to this distribution (line 6). Another difference with other EA is that the steady-state replacement scheme does not make sense in an EDA: the pool of solutions must contain enough solutions to permit the estimation of the probability distribution.

Algorithm 3 Estimation of distribution algorithm

```

1:  $\mu \leftarrow \text{InitializePopulation}()$ 
2:  $\text{EvaluateFitness}(\mu)$ 
3: while  $\text{Continue}()$  do
4:    $\lambda \leftarrow \text{Selection}(\mu)$ 
5:    $\delta \leftarrow \text{EstimationOfDistribution}(\lambda)$ 
6:    $\nu \leftarrow \text{DrawFromDistribution}(\delta)$ 
7:    $\text{EvaluateFitness}(\nu)$ 
8:    $\mu \leftarrow \text{ReplacePopulation}(\mu, \nu)$ 
9: end while

```

Memetic Algorithms

Memetic algorithms, also known as Lamarckian EAs, were proposed by Moscato in 1989 (Moscato, 1989). A memetic algorithm follows the generic EA scheme described in Algorithm 4, and adds a local search step after the application of stochastic variation operators (line 6). The local search method can be any single solution neighborhood exploring algorithm, e.g., simulated annealing (Kirkpatrick *et al.*, 1983), Tabu search (Glover and Laguna, 1999), Iterated local search (Lourenco *et al.*, 2001). Memetic algorithms are often called hybrid genetic algorithms because of the hybridization between classic GA stochastic variation operators (crossover, mutation) and heuristics, i.e., domain specific local search methods. This kind of algorithm is very successful because of its natural tendency to efficiently balance search space exploration with stochastic variation and local intensification with heuristics.

Algorithm 4 Memetic algorithm

```

1:  $\mu \leftarrow \text{InitializePopulation}()$ 
2:  $\text{EvaluateFitness}(\mu)$ 
3: while  $\text{Continue}()$  do
4:    $\lambda \leftarrow \text{Selection}(\mu)$ 
5:    $\lambda \leftarrow \text{StochasticVariations}(\lambda)$ 
6:    $\lambda \leftarrow \text{LocalSearch}(\lambda)$ 
7:    $\text{EvaluateFitness}(\lambda)$ 
8:    $\mu \leftarrow \text{ReplacePopulation}(\mu, \lambda)$ 
9: end while

```

Exploration Versus Exploitation

The balance between exploration and exploitation, or diversification and intensification, is a longstanding and well documented issue in population-based sampling (Črepinšek *et al.*, 2013). Exploration refers to search space sampling, and exploitation refers to local optimization. The fact that these opposite concepts have to be reconciled in order to solve any optimization problem gave birth to the so-called exploration/exploitation trade-off. The optimal trade-off being problem dependent, it is important to devise methods able to tune the balance between them. In EA, the influence of exploitation can be linked to the convergence of the population. If the influence of exploitation is too high, the algorithm cannot escape the local optimum basin it is currently exploring, and it converges prematurely. On the other hand, if the influence of exploitation is too low, the algorithm randomly explores the search space without going deep enough into the optima basins. Both of these extreme behaviors are unwanted and lead to poor performance. If we come back to the evolutionary metaphor, the balance between exploration and exploitation can be seen as the selective pressure exerted on individuals competing for survival. When the selective pressure is high, only individuals expressing specific vital genetic traits survive and reproduce. In this scenario the diversity in the population is reduced, which limits the exploration of the search space by recombination of the individuals, and favors exploitation of specific attraction basins. When the selective pressure is low, there are no vital genetic traits. The lack of constraints on the genes of the individuals allows them to freely explore the fitness landscape. Many different models have been proposed in order to achieve optimal trade-offs. In the island model, solutions are separated into subpopulations. This separation allows an algorithm to follow several search trajectories in parallel, and solutions can be periodically traded between islands (Whitley *et al.*, 1998). In the cellular model, solutions are spatially distributed on a grid and can only interact for crossover with their neighbors (Alba and Dorronsoro, 2008). This spatial arrangement of solutions allows one to tune the balance between exploration and

exploitation by playing on the size and the definition of neighborhoods (Simoncini *et al.*, 2006, 2009). Besides these generic schemes, problem specific heuristics can be designed in order to adapt to the search space properties and provide efficient sampling. The protein structure prediction problem, with a huge search space and rugged landscape, is an example of an important application for which many heuristics have been designed looking for the best trade-off between exploration and exploitation.

Application to Fragment-Based Protein Structure Prediction

This section uses EdaFold as an example of a population-based sampling algorithm as it is applied to protein structure prediction. EdaFold is an Estimation of Distribution Algorithm (EDA) built on top of the Rosetta fragment-based *ab initio* protocol. After a brief introduction to fragment-based protein structure prediction, and a presentation of the Rosetta modeling software, the EDA mechanism implemented in EdaFold will be described and a practical use case discussed.

Fragment-Based Protein Structure Prediction

Proteins are chains of amino acids connected by peptide bonds. A protein chain naturally folds into a three-dimensional structure known to be responsible for its role or function. Each amino acid residue in a protein chain has three angular degrees of freedom on its main chain: the ϕ , ψ and ω angles. These angles determine the global fold of a protein. The ω angle is highly constrained by the planar nature of the peptide bond, and is therefore often neglected by protein structure prediction methods, which focus on the values of angles ϕ and ψ (see Fig. 2). For a protein composed of 100 residues, assuming we only consider 180 discrete angular values for ϕ and ψ , the size of the search space is about 10^{255} .

The principle of fragment-based protein structure prediction is to discretize and diminish the size of the search space by assembling small protein fragments together instead of considering all possible degrees of freedom induced by the protein main chain torsion angles. Besides reducing the size of the search space, fragment-based approaches have the advantage of extracting structural information from known protein structures that locally resemble the target protein. Since the relationship between sequence and structure is well known, the idea makes sense and has encountered a lot of success in the last two decades. Fragment-based methods drew the attention of researchers after the results obtained by David Baker's group during the CASP protein structure prediction blind challenge (Bradley *et al.*, 2005a).

The fragment-based approach implemented in Rosetta consists of two phases: fragment assembly and refinement. The fragment assembly phase uses a coarse-grained representation of proteins where the residue side chains are replaced by a center of mass centroid. It is decomposed in 4 stages. In the first three stages, fragments of length 9 (9-mers) are used. The last stage uses fragments of length 3 (3-mers). The fragments are overlapping: the 9-mers (respectively 3-mers) library contains 25 (respectively 200) fragments for each residue from 1 to N-8 (respectively N-2) where N is the length of the protein sequence. What differentiates the stages is the score function that is used. The score function becomes more and more complex as the search proceeds, and the number of fragment insertion trials increases. The optimization is handled by a Monte Carlo simulated annealing process, which randomly selects a position in the protein sequence, randomly selects a fragment from the library, and evaluates the energy variation induced by the insertion of the fragment. The fragment insertion is accepted with probability $\min\left(1, e^{-\frac{E'-E}{T}}\right)$ where E' is the energy of the model after fragment insertion, E the energy of the model before fragment insertion, and T a temperature parameter. Using this technique, called the Metropolis criterion (Hastings, 1970), the fragment insertion is

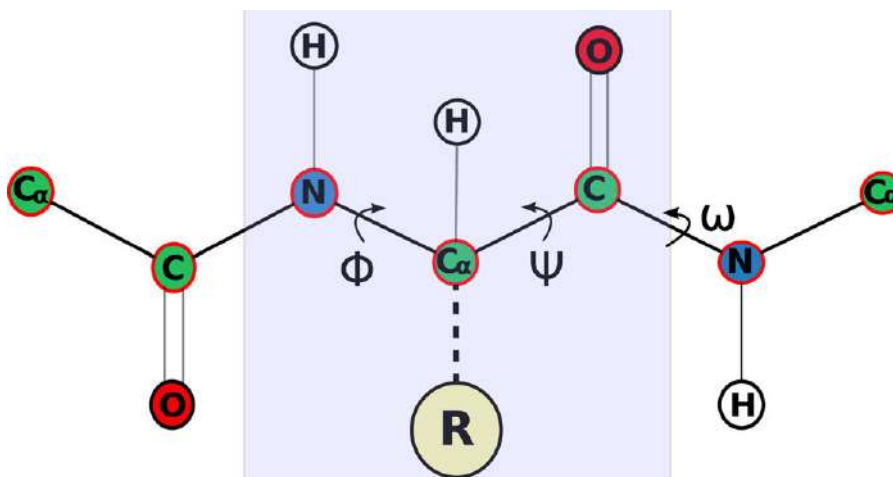


Fig. 2 Amino acid residues in a protein and main chain ϕ , ψ and ω torsion angles. One amino acid is shaded in blue. The R group is the side chain which differentiates the amino acids.

automatically accepted if the new energy is better. Otherwise, it is accepted with a probability that depends on the temperature and the energy difference. The temperature parameter allows one to calibrate which models that worsen the energy will be accepted. T gradually decreases over the course of the simulation. This biases the process toward exploration in the early (high T) stages, and toward exploitation in the late (low T) stages. After the fragment insertion phase, the resolution of the protein models is switched to an all-atom representation for the refinement phase. This phase, known as the Relax protocol, alternates between gradient descent minimization and residue side chain packing evaluation.

Due to the size of the search space and the number of local minima, many independent simulated annealing trajectories (or runs) have to be performed in order to have a chance of finding a model with the correct fold. While they may be globally wrong, many of the models that are produced contain correct and useful local topologies that favorably contribute to the total energy of the protein. Identifying and exploiting this information is the driving idea behind the population-based sampling method, EdaFold, and its latest implementation, EdaRose.

EdaRose

The EDA algorithm implemented in EdaRose adjusts the probability distribution over the fragment libraries in order to guide the search toward native-like protein models. This probability distribution is typically uniform in single solution simulated annealing, such as in Rosetta, where each fragment has the same probability of being selected during the fragment assembly phase. EdaRose modifies the probability distributions empirically, accordingly to the observed frequency of the fragments in a pool of pre-selected models. The EDA only changes the 9-mer fragment assembly phase. The final, and much shorter, 3-mer assembly phase remains unchanged. The search space sampling method implemented in EdaRose does not rely solely on the EDA: it includes a local search method as well. EdaRose can thus also be classified as a memetic algorithm since it combines an EDA with a local search method, even though it has many unique properties which will be described below. EdaRose is built on top of Rosetta and uses its protein representation and simulated annealing method (Simoncini *et al.*, 2017).

Protocol design particularities

Before presenting the algorithm, we focus on some unique characteristics of EdaRose which are, to some extent, dictated by the application to protein structure prediction. Some of these particularities are related to the representation of protein models and the way they are constructed, some others are related to the search dynamics.

Indirect representation of solutions. The first particularity of EdaRose is that the EDA does not try to capture the probability distribution describing the solutions: it is intended to capture the probability distribution over the fragments that are used to build the solutions. The data contained in the fragments are not 3D coordinates of amino acid residues, but ϕ and ψ torsion angles, which are sufficient to determine the fold of a protein. The 3D Cartesian coordinates can be easily obtained from the torsion angle values. Since the fragments overlap, the order in which they are inserted has an influence on the fold of the final model: the insertion of a fragment that partially overlaps another, previously inserted, fragment will overwrite pre-existing torsion angle data, and modify the global fold of the model. In EdaRose, each model keeps track of the list of fragments that were used to build it, in order to transmit this information to the EDA. However, the information about the order in which the fragments were inserted is lost. This particularity can lead to extreme cases where the distribution is a Dirac delta function that assigns a probability of 1 to a single value and 0 to all other values. In such a case, there should be a unique solution, but the actual number of solutions is factorial (W) where W is the number of fragment windows. This is due to the fact that the EDA acts on an indirect representation of the solutions, the fragments, and because the final model fold relies on the order in which fragments were inserted.

EDA and local search interplay. Another particularity of EdaRose is the interlacing of the EDA and the simulated annealing, which is used as a local search method. In a standard scheme, solutions would be generated based on the EDA, and would then be used as starting points in order to run simulated annealing search trajectories. Instead, EdaRose uses the EDA to directly bias the search trajectories performed by the simulated annealing algorithm. The solutions are not generated with the EDA but with the local search method, which uses the EDA to decide which move to make, i.e., which fragment to insert in the protein model. At the end of an iteration, the solutions are gathered and examined in order to refine the probability distribution used by the simulated annealing. Even if this scheme is not specific to protein structure prediction, and could be implemented for any application, it facilitates the integration of EdaRose in the Rosetta macromolecular modeling software.

Algorithm

Algorithm 5 presents an outline of EdaRose as implemented on top of Rosetta. The algorithm takes as inputs the sequence of the protein which is to be modeled and a desired number of iterations. The EDA estimates the probability distribution over the fragments once per iteration. It is initialized with a uniform distribution (line 4) and a first batch of solutions is generated (lines 5 and 6). The *fast_relax* function is an implementation of the Rosetta Relax protocol that minimizes the energy of the model in the all-atom representation, whereas the *simulated_annealing* function performs fragment insertion using a coarse-grained representation. The relaxation in all-atom representation is optional and can be skipped. The iterative process then begins with the selection of the models (line 8), which will be used by the EDA in order to refine the probability distribution (line 9). The model selection is a crucial part of the algorithm, and will be discussed in more detail in the following section. Models are then generated using fragment insertion with the EDA-based simulated annealing, and an optional all-atom relaxation of the models (lines 10

and 11). Contrary to most EA, the population of solutions is not replaced at the end of an iteration. All the solutions are kept in the population which grows linearly from the first to the last iteration. This particular behavior was adopted mainly because of the uncertainty associated with the inaccuracy of the objective function. As is the case for many practical real-world applications, the objective function is empirical and there is no guarantee that its global minimum corresponds to a protein model with the correct fold. It is thus wiser to save all models, and to implement a post-filtering strategy in order to identify the native-like models in the solutions produced throughout the entire process. The main phases of the algorithm are recapitulated in Fig. 3: fragment assembly, model relaxation, model selection, and estimation of distribution.

Algorithm 5 EdaRose

```

1: input :  $s$  {sequence of the target protein}
2: input :  $n$  {number of iterations}
3: output :  $p$  {set of potential solutions}
4:  $pmf \leftarrow \text{init\_with\_uniform\_distributions}()$ 
5:  $p \leftarrow \text{simulated\_annealing}(s, pmf)$ 
6:  $p \leftarrow \text{fast\_relax}(p)$  {optional}
7: for  $i$  in  $[1..n - 1]$  do
8:    $sub_p \leftarrow \text{select\_solutions}(p)$ 
9:    $pmf \leftarrow \text{estimate\_pmf}(sub_p)$ 
10:   $p \leftarrow p \cup \text{simulated\_annealing}(s, pmf)$ 
11:   $p \leftarrow \text{fast\_relax}(p)$  {optional}
12: end for
13: return  $p$ 
  
```

The selection of solutions plays a central role in determining the diversity of the population and the quality of the estimation of the probability distribution. Protein structure prediction offers many possibilities for the design of selection strategies. Two strategies are currently implemented in EdaRose: energy selection, and energy-based clustering selection. The first strategy selects protein models according to the evaluation of their energy as calculated by the scoring function. A parameter controls the proportion of the population that will be retained for the estimation of the distribution over the fragments at each iteration. The

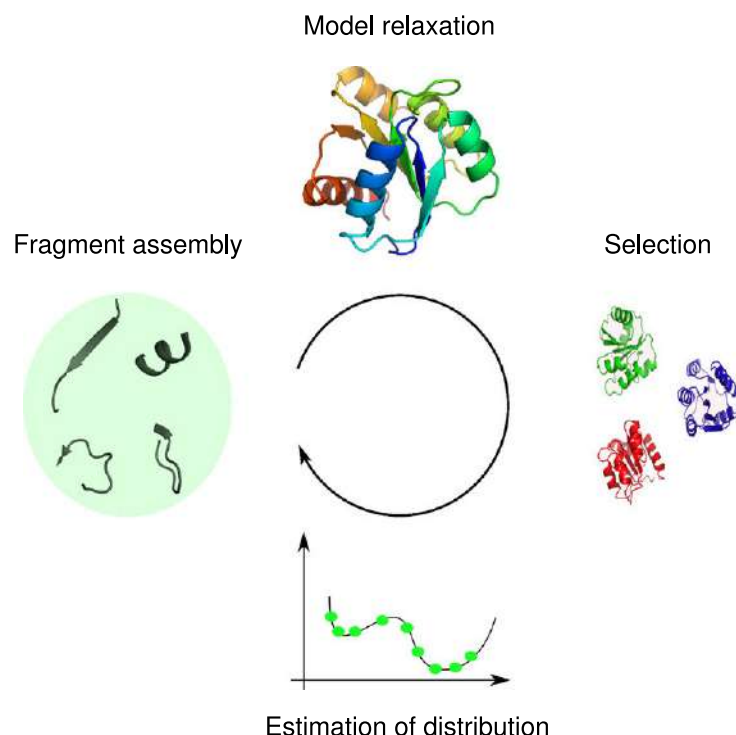


Fig. 3 Illustration of the four main steps of EdaRose. The iterative process starts with fragment assembly of protein models. The models are then relaxed in all-atom representation. Representatives are then selected and exploited in order to refine the probability distribution over the fragments used during assembly.

second strategy uses a hierarchical clustering approach to select protein models that cover large regions of the search space. The energy being an important selection criterion, it is used to guide the clustering algorithm. This strategy is outlined in [Algorithm 6](#). The desired number of clusters and the clustering radius (the maximum distance between a cluster center and a member of the same cluster) have to be specified. By default, the number of clusters is set to 100 and the clustering radius to 4 Å RMSD. The algorithm selects the lowest energy solution from the set of potential solutions as a cluster center (line 7), and adds it to the set of selected models (line 8). At the beginning of the process, the set of potential solutions contains all of the models that were produced during one iteration of EdaRose. Then, the algorithm finds all the solutions belonging to the same cluster and removes them from the set of potential solutions (line 9). This procedure is repeated until the desired number of solutions (i.e., number of clusters) is reached (line 6). Energy-based clustering is the default selection method in EdaRose. The ability to customize selection methods is an attractive feature of population-based sampling algorithms because it offers a way to control the search dynamics and the convergence rate of the algorithm. Furthermore, it allows one to introduce problem specific information that may be critical for the success of the sampling process. For protein structure prediction, many features could be exploited in order to design efficient selection methods. For example, residue contacts derived from co-evolutionary analysis could be used by a selection method in order to favor models that satisfy those contact restraints.

Algorithm 6 Model selection scheme for energy-based clustering.

```

1: input :  $P$  {set of potential solutions}
2: input :  $nbclusters$  {number of clusters to extract}
3: input :  $d$  {clustering radius}
4: output :  $M$  {set of selected protein models}
5:  $M \leftarrow \emptyset$ 
6: for  $i$  in  $[1..nbclusters]$  do
7:    $m \leftarrow find\_lowest\_energy\_model(P)$ 
8:    $M \leftarrow M \cup m$ 
9:    $P \leftarrow remove\_cluster(P, m, d)$ 
10: end for
11: return  $M$ 

```

Test Case

In this section, we present the whole process of a population-based protein structure prediction with EdaRose: from protein fragment library creation to protein model generation and analysis. In this example, we will try to predict the structure of an IgG-binding B1 domain of protein L (pdb code 1HZ5) ([O'Neill et al., 2001](#)). We will go through the whole process step by step.

Fragment library creation. The first step is to generate the fragment libraries. Two libraries need to be generated: a library of 9-mers and a library of 3-mers. The simplest way to do this is to use the Robetta server online ([Kim et al., 2004](#)): <http://rosetta.bakerlab.org/>. In order to generate the fragment libraries, you will need to submit a fragment library task: log in, enter a target name and copy/paste the fasta sequence of the target protein (1hz5 in our case). If you want to test the reliability of EdaRose ignoring the homologous structures in the PDB, remember to tick the *exclude homologues* box. Once the job is completed, download the two fragment libraries (the first two files in the results) and rename them to *1hz5.frag3* and *1hz5.frag9*.

Parameters setting and command line. The main parameters of EdaRose are listed in [Table 2](#). Some of these parameters are specific to EdaRose: the number of iterations, the damping factor of the EDA (which controls the convergence speed) and parameters relative to the selection method. The other parameters are mandatory Rosetta input files and options. Many other Rosetta options such as the use of constraints or symmetry are compatible with EdaRose, please refer to the Rosetta documentation (<https://www.rosettacommons.org/>) for more details. A typical EdaRose command line would be:

```

mpirun -np NP ./EdaRose -in:file:fasta 1hz5.fasta -in:file:frag3 1hz5.frag3
-in:file:frag9 1hz5.frag9 -eda:nbiterations 6 -eda:popthreshold 0.6
-nstruct 60000 -out:pdb

```

where NP is the number of cores used by Open MPI, an open source implementation of the Message Passing standard for parallel computing, which is mandatory. With the default setting, no all-atom relaxation will be performed and the predicted models will be produced in the coarse-grain representation. All-atom relaxation can be included in the process with the appropriate option (see [Table 2](#)), even though it is a time consuming step. Another possibility is to use the Rosetta relax application as a post-processing step in order to relax only a fraction of the best energy models (see the Rosetta documentation at https://www.rosettacommons.org/docs/latest/application_documentation/structure_prediction/relax for more details about Rosetta relax protocol). In this example, we opted for the latter solution and relaxed the best 1000 energy models. For this test-case, the computation time was roughly 4 h, using 100 cores for the prediction of 60,000 models, and 30 min for the relaxation of 1000 models

Table 2 EdaRose options. Parameters prefixed by *eda* are specific to EdaRose

-eda:nb_iterations	Number of iterations
-eda:distrib_k	EDA conservation rate
-eda:clustering	clustering-based model selection (default: true)
-eda:pop_threshold	Top ranked model threshold for energy-based selection
-in:file:fasta	target sequence in fasta format
-in:file:frag9	9-mers fragments library file
-in:file:frag3	3-mers fragments library file
-nstruct	Total number of models
-relax:fast	Specify this option to include model relaxation

The other ones are mandatory Rosetta options.

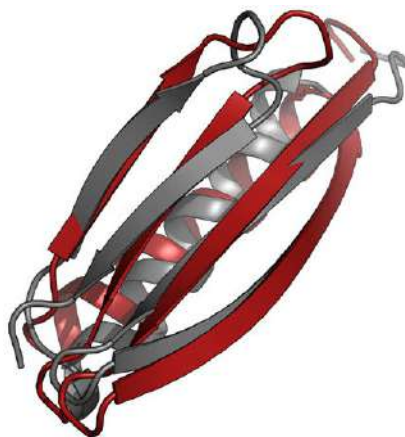


Fig. 4 Protein cartoon model generated with EdaRose in red superimposed on the target native structure (pdb code 1hz5) in gray. The ribbons represent a smooth curve fit through the C_{α} carbons of the backbone; the arrows represent beta strands; the ribbons without arrowheads indicate alpha helices, and irregular regions are shown as narrow tubes. The protein model has the second best energy in the population of models and has a root mean square deviation to the native structure of 3 Å.

using 20 cores. The number of models can be adapted to the computational power available, keeping in mind that the size of the population plays an important role in the performance of the EDA.

Results analysis. The computational job produces a *score file* (score.fsc by default) containing the breakdown of the energy terms and the total score of each model produced during the run. With the flag *out:pdb* on the command line, each model will be output in PDB format. In our example, we generated 60,000 models in 6 iterations (10,000 per iteration) and relaxed the top 1000 lowest energy models using the all-atom representation. Fig. 4 shows the most accurate model, in terms of root mean square deviation to the native structure, out of the 5 lowest energy models (please note that the all-atom relaxation re-ranks the models energy-wise). When no homologous protein can be identified in the PDB, a prediction run can be considered successful if there is at least one accurate model out of the five lowest energy models. This quality criterion is widely accepted: during the biennial blind protein structure prediction contest, CASP, the participants are allowed to send five models per target. In addition to the energy of models, several options exist for assessing the quality of the predictions. The quality assessment of protein models is a research field on its own, and the most popular methods employ clustering and pairwise analysis (consensus scores) in order to re-rank the protein models.

Conclusion

Protein structure prediction remains an open problem of great importance in structural biology. The huge size of the search space, and the ruggedness of the fitness landscape, make it a challenging NP-hard problem, even for modern computer hardware and state-of-the-art optimization algorithms. Popular methods, such as Rosetta, need to generate a substantial number of protein models due to the typically large number of local minima. In this context, and with the democratization of parallel computer architectures, population-based meta-heuristics such as evolutionary algorithms represent an interesting alternative. Population-based methods have the ability of sharing information between the solutions. This ability provides multiple possibilities: mainly, it allows the steering of the search by extracting and exploiting information from good solutions, and it enhances the exploration of the search space by promoting diversity in the population. Of course, exploitation of good solutions and exploration of the

search space are in competition during a search trajectory. Population-based methods can be tuned and customized in order to adjust the balance between exploitation and exploration. The ideal balance is problem dependent and expert knowledge is often needed in order to improve the efficiency of the algorithms. Evolutionary algorithms are highly customizable at many different levels, from the selection of solutions to the stochastic operators, and even the population replacement strategy. Many features and useful information can be extracted from proteins and used to make the protein structure prediction problem highly compatible with population-based methods. EdaRose is an estimation of distribution algorithm built on top of Rosetta. It has been customized in order to exploit some information about the structure of the protein models. Many research studies put in evidence the benefits of population-based methods in terms of sampling and performance compared to single solution methods. The introduction of additional expert knowledge, or the design of novel concerted search strategies, would probably further improve the gain in performance, taking advantage of the flexibility of population-based sampling methods.

See also: Assessment of Structure Quality (RNA and Protein). Investigating Metabolic Pathways and Networks. Natural Language Processing Approaches in Bioinformatics

References

- Alba, E., Dorronsoro, B., 2008. Cellular Genetic Algorithms, 1st ed. Springer Publishing Company. (Incorporated. ISBN: 0387776095, 9780387776095).
- Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. *Science* 181.
- Bale, J.B., *et al.*, 2016. Accurate design of megadalton-scale two-component icosahedral protein complexes. *Science* 353 (6297), 389–394. (ISSN: 0036-8075).
- Bradley, P., Malmstrom, L., *et al.*, 2005a. Free modeling with Rosetta in CASP6. *Proteins: Structure, Function, and Bioinformatics* 61 (S7), 128–134. (ISSN: 1097-0134).
- Bradley, P., Misura, K.M.S., Baker, D., 2005b. Toward high-resolution de novo structure prediction for small proteins. *Science* 309 (5742), 1868–1871. (ISSN: 1095-9203).
- Clausen, R., Shehu, A., 2014. A multiscale hybrid evolutionary algorithm to obtain sample-based representations of multi-basin protein energy landscapes. In: *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. BCB '14*. Newport Beach, CA: ACM, pp. 269–278. ISBN: 978-1-4503-2894-4.
- Črepinšek, M., Liu, S.-H., Mernik, M., 2013. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.* 45 (3), 35:1–35:33. (ISSN: 0360-0300).
- Dandekar, T., Argos, P., 1992. Potential of genetic algorithms in protein folding and protein engineering simulations. *Protein Eng* 5 (7), 637–645. (ISSN: 0269-2139).
- Dandekar, T., Argos, P., 1994. Folding the Main Chain of Small Proteins with the Genetic Algorithm. *Journal of Molecular Biology* 236 (3), 844–861. (ISSN: 0022-2836).
- Dandekar, T., Argos, P., 1996. Identifying the tertiary fold of small proteins with different topologies from sequence and secondary structure using the genetic algorithm and extended criteria specific for strand regions. *Journal of Molecular Biology* 256 (3), 645–660. (ISSN: 0022-2836).
- Darwin, C., 1859. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favored Races in the Struggle for Life*. London: Murray.
- Dill, K.A., 1985. Theory for the folding and stability of globular proteins. *Biochemistry* 24 (6), 1501–1509.
- Garza-Fabre, M., *et al.*, 2016. Generating, maintaining and exploiting diversity in a memetic algorithm for protein structure prediction. *Evolutionary Computation*. (ISSN: 1063-6560).
- Glover, F., Laguna, M., 1999. Tabu search. In: Du, D.-Z., Pardalos, P.M. (Eds.), *Handbook of Combinatorial Optimization* 1-3. MA: Springer US, pp. 2093–2229. (ISBN: 978-1-4613-0303-9).
- Hamelryck, T., Kent, J.T., Krogh, A., 2006. Sampling realistic protein conformations using local structural bias. *PLOS Comput Biol* 2 (9), e131. (ISSN: 1553-7358).
- Hart, W.E., Istrail, S., 1997. Robust proofs of NP-hardness for protein folding general lattices and energy potentials. *Journal of Computational Biology* 4.
- Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57 (1), 97–109.
- Holland, J.H., 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press. (ISBN: 0472084607).
- Hoque, M.T., Chetty, M., Dooley, L.S., 2005. A new guided genetic algorithm for 2D hydrophobic-hydrophilic model to predict protein folding. In: *2005 IEEE Congress on Evolutionary Computation*. Vol. 1, pp. 259–266.
- Kandathil, S., Lovell, S., Handl, J., 2016. Toward a detailed understanding of search trajectories in fragment assembly approaches to protein structure prediction. *Proteins: Structure, Function and Bioinformatics* 84 (4), 411–426. (ISSN: 0887-3585).
- Khersonsky, O., *et al.*, 2011. Optimization of the In-silico-designed kemp eliminase (KE70) by computational design and directed evolution. *Journal of Molecular Biology* 407 (3), 391–412. (ISSN: 0022-2836).
- Kim, D.E., Chivian, D., Baker, D., 2004. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Research* 32 (suppl_2), W526.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Leaver-Fay, A., *et al.*, 2011. ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol* 487, 545–574. (ISSN: 1557-7988).
- Levinthal, C., 1968. Are there pathways for protein folding? *Journal de Chimie Physique* 65 (1), 44–45.
- Lin, C.-J., Hsieh, M.-H., 2009. An efficient hybrid Taguchi-genetic algorithm for protein folding simulation. *Expert Syst. Appl.* 36 (10), 12446–12453. (ISSN: 0957-4174).
- Lourenco, H., Martin, O., Stutzle, T., 2001. Iterated local search. In: Glover, F., Kochenberger, G. (Eds.), In: *Handbook of Metaheuristics* 57. Kluwer, pp. 321–353. (ISORMS).
- Moscato, P., 1989. On evolution, search, optimization, genetic algorithms and martial arts – towards memetic algorithms. in: *Technical report*.
- Muhlenbein, H., Paaß, G., 1996. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature – PPSN IV*. Springer-Verlag. pp. 178–187.
- Olson, B., Shehu, A., 2013. Multi-objective stochastic search for sampling local minima in the protein energy surface. In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. BCB'13*. Washington DC: ACM, 430:430-430:439. ISBN: 978-1-4503-2434-2.
- O'Neill, J.W., *et al.*, 2001. Structures of the B1 domain of protein L from *Peptostreptococcus magnus* with a tyrosine to tryptophan substitution. *Acta Crystallographica D* 57 (4), 480–487.
- Park, S.-J., 2005. A study of fragment-based protein structure prediction: Biased fragment replacement for searching low-energy conformation. *Genome Inform* 16 (2), 104–113. (ISSN: 0919-9454).
- Pedersen, J.T., Moulton, J., 1995. Ab initio structure prediction for small polypeptides and protein fragments using genetic algorithms. *Proteins: Structure, Function, and Bioinformatics* 23 (3), 454–460. (ISSN: 1097-0134).
- Rohl, C.A., *et al.*, 2004. Protein structure prediction using Rosetta. *Methods Enzymol* 383, 66–93.
- Sakae, Y., *et al.*, 2011. Protein structure predictions by parallel simulated annealing molecular dynamics using genetic crossover. *Journal of Computational Chemistry* 32 (7), 13531360. (ISSN: 1096-987X).
- Saleh, S., Olson, B., Shehu, A., 2013. A population-based evolutionary search approach to the multiple minima problem in de novo protein structure prediction. *BMC Structural Biology* 13 (1), S4. (ISSN: 1472-6807).

- Shmygelska, A., Levitt, M., 2009. Generalized ensemble methods for de novo structure prediction. *Proc Natl Acad Sci USA* 106 (5), 1415–1420. (ISSN: 1091-6490).
- Siegel, J.B., *et al.*, 2010. Computational Design of an Enzyme Catalyst for a Stereoselective Bimolecular Diels-Alder Reaction. *Science* 329 (5989), 309–313. (ISSN: 0036-8075).
- Simoncini, D., *et al.*, 2006. Anisotropic selection in cellular genetic algorithms. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. GECCO '06*. Seattle, Washington, USA: ACM, pp. 559–566. ISBN: 1-59593-186-4.
- Simoncini, D., *et al.*, 2009. Centric selection: A way to tune the exploration/exploitation tradeoff. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. GECCO '09*. Montreal, Quebec, Canada: ACM, pp. 891–898. ISBN: 978-1-60558-325-9.
- Simoncini, D., Schiex, T., Zhang, K.Y., 2017. Balancing exploration and exploitation in population-based sampling improves fragment-based de novo protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*. (ISSN: 1097-0134).
- Simoncini, D., Berenger, F., *et al.*, 2012. A probabilistic fragment-based protein structure prediction algorithm. *PLOS One* 7 (7), e38799. (SSN: 1932-6203).
- Simoncini, D., Zhang, K.Y.J., 2013. Efficient sampling in fragment-based protein structure prediction using an estimation of distribution algorithm. *PLOS ONE* 8 (7), 1–10.
- Sun, S., 1993. Reduced representation model of protein structure prediction: Statistical potential and genetic algorithms. *Protein Science* 2 (5), 762–785. (ISSN: 1469-896X).
- Unger, R., Moul, J., 1993. Genetic algorithm for 3D protein folding simulations. In: *Proceedings of the 5th International Conference on Genetic Algorithms*. San Francisco, CA: Morgan Kaufmann Publishers Inc., pp. 581–588. ISBN: 1-55860-299-2.
- Voet, A.R.D., *et al.*, 2014. Computational design of a self-assembling symmetrical beta-propeller protein. *Proceedings of the National Academy of Sciences* 111 (42), 15102–15107.
- Whitley, D., Rana, S., Heckendorn, R.B., 1998. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology* 7, 33–47.
- Zhang, X., *et al.*, 2010. 3D Protein structure prediction with genetic tabu search algorithm. *BMC Systems Biology* 4 (1), S6. (ISSN: 1752-0509).
- Zhao, F., Li, S., *et al.*, 2008. Discriminative learning for protein conformation sampling. *Proteins* 73 (1), 228–240. (ISSN: 1097-0134).
- Zhao, F., Peng, J., Xu, J., 2010. Fragment-free approach to protein folding using conditional neural fields. *Bioinformatics* 26 (12), i310-7. (ISSN: 1367-4811).

Further Readings

- Alba, E., Tomassini, M., 2002. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 6 (5).
- Bowie, J.U., Eisenberg, D., 1994. An evolutionary approach to folding small alpha-helical proteins that uses sequence information and an empirical guiding fitness function. *Proceedings of the National Academy of Sciences, USA* 91, 4436–4440.
- Branden, C., Tooze, J., 1998. *Introduction to protein structure*. Garland Publishing.
- Defoin-Platel, M., Schliebs, S., Kasabov, N., 2009. Quantum-inspired evolutionary algorithm: A multimodel EDA. *IEEE Transactions on Evolutionary Computation* 13 (6).
- Goldberg, D.E., 1989. *Genetic algorithms in search, optimization and machine learning*, first ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Goldberg, D.E., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers.
- Krasnogor, N., Blackburne, B.P., Burke, E.K., Hirst, J.D., 2002. Multimeme algorithms for protein structure prediction. In: *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature (PPSN VII)*. Springer-Verlag, London, pp. 769–778.
- Pelikan, M., Goldberg, D.E., Cantú-Paz, E., 1999. BOA: The Bayesian optimization algorithm. In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation (GECCO'99)*, Vol. 1. Morgan Kaufmann Publishers.
- Talbi, E., 2009. *Metaheuristics: Design to Implementation*. Wiley Publishing.
- Tramontano, A., 2006. *Protein Structure Prediction Concepts and Applications*. Wiley Publishing.

Relevant Websites

- <https://www.rosettacommons.org/>
Rosetta Commons.
- https://www.rosettacommons.org/docs/latest/application_documentation/structure_prediction/
Rosetta Commons.

Ontology: Introduction

Gianluigi Greco and Marco Manna, University of Calabria, Cosenza, Italy

Francesco Ricca, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Overview

Ontology – from the Greek ὄν, genitive ὄντος: *of being* (present participle of ἔIMI: *to be*), and ΛΟΓΙΑ for ΛΟΓΟΣ: *science, study, theory* – is the study of the nature of being, existence, or reality in general and of its basic categories and their relations.

Moving from the above original definition, nowadays, an ontology is pragmatically viewed in computer science as structure that specifies a conceptualization or, more accurately, *a specification of a shared conceptualization of a domain* (Thomas, 1993). In fact, this term is very often used to refer to a wide range of different formal representations, including taxonomies, hierarchical terminology vocabularies or detailed logical theories describing a domain (Noy and Klein, 2004). Therefore, providing the reader with an accurate and univocally determined definition is rather difficult (Thomas, 1993, 1995; Guarino, 1998). For the sake of concreteness and by following a rather accepted abstract formalization (de Bruijn et al., 2004; Predoiu et al., 2005), we present here an ontology as a quadruple $\mathcal{O} = \langle C, R, \iota, \xi \rangle$, where C is a set of concepts, R is a set of relations, ι is a set of instances and ξ is a set of axioms. In fact, some authors prefer not to include ι in $\mathcal{O}$ and to talk about a *knowledge base* when an ontology (with no instance) is paired with instances.

A concept in an ontology is a category of real or virtual objects of interest, which are the instances. Relations specify how different concepts (or instances) are interconnected, and axioms are formulae that allow to further define dependencies between all mentioned ontology entities. Moreover, the sets C and ι are not necessarily disjoint (for instance, the same term can denote both a class and an instance), although it is often required that this property holds. Moreover, in an ontology, concepts are usually organized in a subclass hierarchy, through the *is-a* (or sub-concept-of) relationship. More general concepts reside higher in the hierarchy. Such a special kind of relationship is called *taxonomic*, as well as the *part-of* relationship, which both define hierarchies of concepts.

The specification of an ontology is usually done by using a formal knowledge representation language. This means that concepts, relations, instances, and axioms are specified in some logical language.

As an example, by adopting a first-order like abstract syntax, consider the ontology $\mathcal{O} = \langle C, R, \iota, \xi \rangle$ where $C = \{Exam, Student, Person, Number\}$, $R = \{pass\}$, $\iota = \{Exam(e), Student(s_1), Student(s_2), Number(1), \dots, Number(100), pass(s_1, e, 18), pass(s_2, e, 30)\}$, and finally $\xi = \{pass(S, E, G) \wedge G < 18 \rightarrow \perp, pass(S, E, G) \wedge G > 30 \rightarrow \perp, pass(S, E, g) \wedge \neg Number(g) \rightarrow \perp, pass(S, E, G) \rightarrow Student(S), pass(S, E, G) \wedge \neg Exam(E) \rightarrow \perp, Student(S) \rightarrow Person(S)\}$.

Note that, in this example, the ontology states that e is an exam, that s_1 and s_2 are students, that 1...100 are numbers, that s_1 passes exam e with grade 18, and that s_2 passes exam e with grade 30. Moreover, it formalizes (via ξ the facts that the grade of an exam is a number that may range from 18 to 30, that if someone passes an exam then he/she is a student, that nobody can pass something which is not an exam, and that every student is a person.

An ontology specification should allow one to model the definitions of terms and relations holding in a domain. For this purpose, several ontology specification languages have been proposed in the literature. These languages feature different syntactic, semantic, and expressivity properties. The choice of the most appropriate ontology language depends on the intrinsic characteristic of the target domain, and on the area of application, that is, it depends on the *ontological commitments* (Thomas, 1993) of those who are expected to use the specification.

In the following section, some basic ontology specification languages are surveyed. Notably, some of them have been defined for general knowledge representation tasks, others have been conceived to support the development of ontologies for the semantic Web, and some others were intended for developing enterprise knowledge bases. In the rest of the article, we then overview the most important steps for designing an ontology (see Section Ontology Design), we discuss the tools available to develop and manage ontologies (see Section Tools for Ontology Management), and we present some examples of real ontologies of interest in biological domains (see Section Ontologies in Bioinformatics and Biology).

Ontology Languages

In this section, we mention some ontology specification languages appeared in the recent literature (Calì et al., 2009; Corcho and Gómez-Pérez, 2000; Ricca and Leone, 2007): F-Logic, RDF, OWL, OntoDLP, and Datalog[±].

F-Logic

Frame Logic is a popular logic-based object-oriented language which includes most aspects of object-oriented and frame-based languages (Kifer et al., 1995). It was conceived as a language for intelligent information systems based on the logic programming paradigm. A main implementation of F-logic is the Flora-2 system (Yang et al., 2003) which is devoted to Semantic Web reasoning tasks. Flora-2 integrates F-Logic with other novel formalisms such as HiLog (Chen et al., 1993) (a logical formalism that provides

higher-order and meta-programming features in a computationally tractable first-order setting) and Transaction Logic (Bonner and Kifer, 1994) (that provides a logical foundation for state changes and side effects in a logic programming language).

RDF

A number of formalisms for specifying ontologies have been originally proposed by the World Wide Web Consortium (W3C), in particular, RDF and RDFS. The *Resource Description Framework* (RDF) (Ralph, 1998; W3C, 2006) is a knowledge representation language for the Semantic Web. It is a simple assertional logical language which allows for the specification of binary properties expressing that a resource (entity in the Semantic Web) is related to another entity or to a value. RDF has been extended with a basic type system; the resulting language is called RDF Vocabulary Description Language (RDF Schema or RDFS). RDFS introduces the notions of class and property, and provides mechanisms for specifying class hierarchies, property hierarchies, and for defining domains and ranges of properties. Basically, RDF(S) allows for expressing knowledge about the resources (identified via URI), and features a rich data-type library.

OWL

The *Ontology Web Language* (Smith et al., 2004) is an ontology representation language built on top of RDFS. Ontologies defined in this language consist of concepts (or classes) and roles (binary relations also called class properties). OWL has a logic-based semantics and, in general, it allows us to express complex statements about the domain of discourse (OWL is undecidable in general) (Smith et al., 2004). More precisely, Description Logics (DLs) (Baader et al., 2003) represent the key logics underpinning OWL. The largest decidable subset of OWL, called OWL-DL, basically coincides with $\mathcal{SHOIN}(\mathcal{D})$, one of the most expressive Description Logics (Baader et al., 2007). Finally, OWL and DLs are based on classical logic; in fact, there is a direct mapping from $\mathcal{SHOIN}(\mathcal{D})$ to first-order logic (FOL).

OntoDLP

This is a language based on Disjunctive Logic Programming (DLP) and extended with object-oriented features (Ricca and Leone, 2007). In particular, DLP is an advanced formalism for Knowledge Representation and Reasoning expressing all problems belonging to the complexity class $(\Sigma)_2^P$ – the class of all decision problems solvable in nondeterministic polynomial time by a Turing machine with an oracle in NP. In fact, the language OntoDLP includes, besides the concept of *relation*, the object-oriented notions of *class*, *object* (class instance), *object-identity*, *complex object*, *(multiple) inheritance*, and the concept of modular programming by means of *reasoning modules*.

Datalog[±]

This is a family of knowledge representation languages to specify ontologies (Calì et al., 2009), whose intent is to collect all expressive extensions of Datalog which are based on existential quantification, equality-generating dependencies, negative constraints, negation, and disjunction. In particular, the “plus” symbol refers to any possible combination of these extensions, while the “minus” one imposes at least decidability of the main reasoning tasks (Calì et al., 2012a,b, 2013; Fagin et al., 2005; Leone et al., 2012). Originally, this family was introduced with the aim of “closing the gap between the Semantic Web and databases” (Calì et al., 2012a) to provide the *Web of Data* with scalable formalisms that can benefit from existing database technologies. And, in fact, it generalizes well-known subfamilies of Description Logics – such as $\mathcal{EL}$ (Brandt, 2004) and *DL-Lite* (Artale et al., 2009) – collecting the basic tractable languages in the context of the Semantic Web and databases. Currently, Datalog[±] has evolved as a major paradigm and an active field of research.

Ontology Design

The design of an ontology is a complex process. Therefore, it comes with no surprise that an entire area of computer science is devoted to study and define methods and methodologies for building ontologies (Thomas, 1993). By abstracting from the specific proposals in the literature, we can see that the development of an ontology takes place by considering some fundamental steps:

- Analysis of the domain of interest, where all the necessary information regarding domain and entities is collected. All the information gathered must be translated into a common language chosen for that particular ontology. The formalization process must be carried out by adhering to the domain and by taking into account the purpose of ontology and the users who will handle it;
- Reuse of existing resources, where parts of ontologies already available at hand have to be identified for possible reuse. This step guarantees benefits both in terms of development and time of implementation;
- Planning the conceptual structure of the domain, which corresponds to defining its concepts and properties and the various relationships between them. This step is crucial for developing a flat glossary and elaborating a glossary structure;

- Coherence and consistency checking, which means verifying any semantic, logical and syntactic inconsistencies.

By looking at the above tasks, the design of an ontology can be approached in different ways and the choice of the most appropriate one depends on the characteristics of the domain that has to be modeled. By speaking in general terms, there are three main approaches: bottom-up, top-down and mixed:

- Bottom-Up Approach. In this type of approach, we first define the most specific concepts and then move to characterize the more generic ones. This approach is performed to a degree of generalization and abstraction that allows us to achieve the purposes for which the ontology is designed. As a result, concepts are represented according to a certain hierarchy.
- Top-Down Approach. In this case, we first identify the most general concepts and then move to the more specific ones, reaching an appropriate level for the purposes of ontology. Even in this case, concepts are represented by a taxonomy and their specialization must respect the purpose of ontology.
- Mixed approach. In this type of approach, we first identify all the concepts that are considered important for the domain to be represented, then we look for the hierarchical ties that bind the concepts to organize them from the most general to the most specific.

No matter of the specific design approach being selected, an ontology should be designed to meet some desirable criteria and the ontology designer is in charge to accept trade-offs among them (Thomas, 1993) in order to obtain the best results for a particular application. A few desiderata for ontologies are summarized in the following. First, an ontology should be *clear*, that is, it should be objective and communicate the intended meaning of defined terms. This should be obtained by properly constraining the possible interpretations for the defined terms. An ontology should be *coherent*, that is, the entire specification should be logically consistent, so that it will allow to perform correct inferences and reasonings on the knowledge it specifies. An ontology should be *extensible*, that is, it should be designed to be easily extended with new terms that will be required by future applications. Extensions should be possible by reusing the existing vocabulary. An ontology designed for knowledge-sharing should minimize the *encoding bias*, that is, it should not depend on a particular encoding or notation, in order to maximize usability in different systems and styles of representation. Moreover, a general purpose ontology should have *minimal ontological commitment*, that is, it should make as few claims as possible about the world being modeled and provide definitions for those terms that are essential to model the target domain.

Tools for Ontology Management

Numerous tools have been developed in the past few years, both in the university and commercial fields, in order to provide support for the specification and manipulation of ontologies. The use of such tools that have advanced graphical capabilities is essential for applying ontologies in real-world applications, in particular because users are very often not expert about specific linguistic formalisms for the specification of ontologies.

Various kinds of tools have been developed to cope with various developmental requirements and ontologies. A classification can be made based on the particular use for which such tools have been conceived and developed:

- Tools for the development of ontologies. This category includes all the tools that can be used to build new ontologies from scratch or reuse pre-existing ontologies;
- Tools for integrating and fusing ontologies. These are tools born to deal with the integration of different ontologies concerning the same domain;
- Tools for evaluating ontologies. This category refers to the tools that must ensure the quality of the ontologies and their technologies;
- Ontological-based annotation tools. These tools are designed to allow users to insert and maintain semi-automatic markup-based ontologies (especially) on web pages;
- Tools for querying and storing ontologies. These tools have been designed to simplify the use and the querying of ontologies;
- Tools for learning ontologies. These tools are capable of deriving, semi-automatically, ontologies (usually) from text written in natural language.

For each of the above categories, currently available ontology tools are rather numerous, even though their usage is limited by a number of issues limiting their diffusion, especially in the commercial sphere. Very often, in fact, these tools are complex and are not accompanied by appropriate documentation. Another crucial issue in the use of ontology tools is the need to combine different solutions that are not conceived to natively cooperate with each other. At present, there is no ontology management tool that is a complete platform for ontologies management and that, therefore, can offer in one environment all the features needed to manage ontologies: specifying, manipulating, storing and query.

Among the available solutions, Protégé (Noy *et al.*, 2001), developed at the University of Stanford, is one of the most well known and used. It provides a graphical and interactive editor for ontology design and acquisition of knowledge. The tools has been designed to help knowledge engineers and domain experts to realize knowledge management goals. Ontologists can access information quickly whenever they need it and can modify ontology directly by navigating the tree that represents it. The knowledge model used by Protégé is compatible with current standard and one of the major advantages of its architecture is that it

guarantees an open and modular environment. Indeed, programmers can add new features simply by creating the appropriate plug-ins. Protégé comes equipped with libraries containing display plug-ins, knowledge-based inference engines (to verify first-order logic constraints), and methods to capture information from remote and heterogeneous resources.

In Protégé, a class can be concrete or abstract. In the latter case, the class serves to convey a series of attributes possessed by different classes, but it admits no instances. Protégé offers a panel that manages the building part of the classes and defines the relationships between the concepts that are part of it. In this panel, users can: define the taxonomy of the ontology by pointing out what are the elements of their hierarchy of concepts and relationships (represented by a tree); manage the inheritance relationship between concepts, allowing users to add or remove relatives to a tree node; view and manage attributes related to the selected concept on the tree; managing individuals (instances). Moreover, Protégé implements a panel where conceptual instances can be defined and managed. In this case, the panel is divided into sections that allow: to explore the defined taxonomy; to handle instances linked to the selected concept; to view information related to a single instance. Finally, Protégé supports an ontology query management mechanism. A query can be used to verify, for example, existing relationships between defined concepts, or links to instances.

Ontologies in Bioinformatics and Biology

The use of ontologies in bioinformatics and biology has a long tradition. Indeed, this domain can be considered as one of the prototypical examples for the application of ontologies to concrete real-worlds problems. In the following, we mention some well-known projects that are aimed at developing ontologies in this context as well as some pointers for obtaining these ontologies.

The Gene Ontology (GO) (see Relevant Websites section) is by far the most widely used ontology in biology. GO defines biological processes, molecular functions and cell components. The GO vocabulary is designed to be general and applicable from prokaryotes to multicellular organisms. The vocabulary is organized in three orthogonal hierarchies representing gene product properties: (i) Cellular Component, modeling the parts of a cell or its extracellular environment; (ii) Molecular Function, modeling the elemental activities of a gene product at the molecular level; (iii) Biological Process, modeling operations or sets of molecular events related to the functioning of integrated living units: cells, tissues, organs, and organisms. GO is linked to a database of hundred of thousands genes of animals, plants, fungi, bacteria and viruses. The user can find, for each gene, detailed information about the role it plays and its products.

The Systems Biology Ontology (SBO) is a set of controlled, relational vocabularies of terms commonly used in Systems Biology (see Relevant Websites section). It defines reaction participants roles, quantitative parameters, mathematical expressions describing the system, the nature of the entities, the type of interaction. The aim of SBO is to annotate the results of biochemical experiments in order to facilitate their efficient reuse.

The Ontology of Physics for Biology (OPB) is a reference ontology of classical physics as applied to the dynamics of biological systems (see Relevant Websites section). It is designed to cover multiple physical domains at different structural scales, and to support the study and analysis of biological organisms.

A hub collecting references to many ontologies describing the domains of biology is provided by the Open Biomedical Ontologies (OBO) Foundry (see Relevant Websites section), a network of ontology developers that are committed to the development of a family of interoperable ontologies that are both logically well-formed and scientifically accurate.

Closing Remarks

In this article, we provided an overview of ontologies in computer science. In this area, an ontology is a formal way of representing knowledge in which concepts are described both by their meaning and their relationship with each other. A key component in an ontology specification is the selection of a formal language to model the definitions of the terms and the relations that hold in the given domain. For this purpose, we reviewed several languages featuring different syntactic, semantic, and expressivity properties. For readers interested in ontology engineering, we have discussed the main approaches for designing an ontology, and listed some desirable properties that an ontology is expected to enjoy. Finally, we provided links to existing ontologies that have been developed in the field of biology, in order to provide the reader with references to concrete ontologies that are currently used in practical applications.

See also: Natural Language Processing Approaches in Bioinformatics. The Gene Ontology

References

- Artale, A., Calvanese, D., Kontchakov, R., Za-kharyashev, M., 2009. The DL-Lite family and relations. *J. Artif. Intell. Res.* 36, 1–69.
- Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (Eds.), 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

- Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., 2007. *The Description Logic Handbook*. New York, NY, USA: Cambridge University Press.
- Bonner, A.J., Kifer, M., 1994. An overview of transaction logic. *Theor. Comput. Sci.* 133 (2), 205–265.
- Brandt, S., 2004. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and – What else? In: *Proceedings of ECAI 2004*, pp. 298–302.
- Calì, A., Gottlob, G., Kifer, M., 2013. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.* 48, 115–174.
- Calì, A., Gottlob, G., Lukasiewicz, T., 2009. Datalog[±]: A unified approach to ontologies and integrity constraints. In: *Proceedings of ICDT 2009*, pp. 14–30.
- Calì, A., Gottlob, G., Lukasiewicz, T., 2012a. A general datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* 14, 57–83.
- Calì, A., Gottlob, G., Pieris, A., 2012b. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.* 193, 87–128.
- Chen, W., Kifer, M., Warren, D.S., 1993. Hilog: A foundation for higher-order logic programming. *J. Log. Program.* 15, 187–230.
- Corcho, Ó., Gómez-Pérez, A., 2000. A roadmap to ontology specification languages. In: *EKA'00: Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, pp. 80–96. London, UK: Springer-Verlag.
- de Bruijn, J., Martín-Recuerda, F., Manov, D., Ehrig, M., 2004. State-of-the-art survey on ontology merging and aligning v1. Technical report, SEKT project deliverable D4.2.1. Available at: <http://www.sekt-project.com/rd/deliverables/wp04/sekt-d-4-2-1-Mediation-survey-final.pdf>
- Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L., 2005. Data exchange: Semantics and query answering. *Theor. Comput. Sci.* 336 (1), 89–124.
- Guarino, N., 1998. Formal ontology and information systems. In: *International Conference On Formal Ontology In Information Systems FOIS'98*, pp. 3–15. Trento, Italy; Amsterdam: IOS Press.
- Kifer, M., Lausen, G., Wu, J., 1995. Logical foundations of object-oriented and frame-based languages. *J. ACM* 42 (4), 741–843.
- Leone, N., Manna, M., Terracina, G., Veltri, P., 2012. Efficiently computable datalog programs. In: *Proceedings of KR 2012*.
- Noy, N.F., Klein, M., 2004. Ontology evolution: Not the same as schema evolution. *Knowl. Inf. Syst.* 6 (4), 428–440.
- Noy, N.F., Sintek, M., Decker, S., *et al.*, 2001. Creating semantic web contents with protege-2000. *IEEE Intell. Syst.* 16 (2), 60–71.
- Predoiu, L., de Bruijn, J., Feier, C., *et al.*, 2005. State-of-the-art survey on ontology merging and aligning v2. Deliverable D4.2.2, SEKT.
- Ralph, R. Swick, 1998. Rdf, the resource description framework (tutorial). In: *QL*.
- Ricca, F., Leone, N., 2007. Disjunctive Logic Programming with types and objects: The DLV+ system. *J. Appl. Log.* 5 (3), 545–573.
- Smith, M.K., Welty, C., McGuinness, D.L., 2004. Owl Web Ontology Language Guide. W3C Recommendation. World Wide Web Consortium.
- Thomas, R.G., 1993. A translation approach to portable ontology specifications. *Knowl. Acquis.* 5 (2), 199–220.
- Thomas, R.G., 1995. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.–Comput. Stud.* 43 (5–6), 907–928.
- W3C, 2006. The resource description framework. Available at: <http://www.w3.org/RDF/>.
- Yang, Guizhen, Kifer, M., Zhao, Chang, 2003. flora-2: A rule-based knowledge representation and inference infrastructure for the semantic web. In: *Meersman, Robert, Tari, Zahir, Schmidt, Douglas C. (Eds.), CoopIS/DOA/ODBASE*, pp. 671–688.

Relevant Websites

- <http://geneontology.org>
Gene Ontology Consortium.
- <http://purl.bioontology.org/ontology/OPB>
Ontology of Physics for Biology.
- <http://www.ebi.ac.uk/sbo/main/>
System Biology Ontology.
- <http://www.obofoundry.org/>
The OBO Foundry.

Ontology: Definition Languages

Valeria Fionda and Giuseppe Pirrò, University of Calabria, Rende, Italy and ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Ontologies are formal artifacts that allow to gain a shared understanding of a particular domain of interest. Ontologies model knowledge about individuals, their attributes, and their relationships by providing a level of abstraction that goes beyond classical data models (e.g., the relational model). Indeed, one of the main advantages of ontologies is their ability to provide abstraction from data structures, implementation strategies and lower-level data models. Ontologies are also a useful support for integrating heterogeneous knowledge bases, thus enabling interoperability among different data providers and applications. This is particularly relevant in the context of life sciences where different ontologies are often used by biologists to model overlapping knowledge domains that characterize, for instance, experiments. From a concrete point of view, the definition of ontologies is supported by a variety of ontology languages with different levels of expressiveness; expressiveness refer to the capability of a language to model a particular aspect. Ontology definition languages, indeed, also allow to specify semantic rules to reason about the knowledge encoded in the ontology.

The massive amount of data characterizing today's applications calls for automatic techniques that can process and interpret ontologies. In other words, ontologies need to be interpreted and processed by machines. This will foster the possibility to perform quick and complex types of inferences that are helpful in a variety of tasks, including search and knowledge base integration. A typical context where inference is of utmost importance is the detection (and resolution) of knowledge base inconsistencies. A key ingredient toward the definition of inferences is the definition of logic-based semantics for ontologies.

In this article we provide an overview of some ontology definition languages that have been defined under the umbrella of the W3C consortium. We will focus our attention on three languages: the Resource Description Framework (RDF), RDF Schema (RDFS) and the Ontology Web Language (OWL).

While the RDF language is mostly a data-model for expressing facts, i.e. it does not bring (much) semantics or meaning to such facts, RDFS and OWL allow to make claims about things described by an RDF document. These formal languages (RDFS is quite simple, whereas OWL is much richer) offer a meta-vocabulary with well-defined semantics to express constraints.

Resource Description Framework

The Resource Description Framework (RDF) (Consortium *et al.*, 2014) is a data model to represent information. In particular, RDF has been proposed with the aim of describing annotations (facts) about Web resources. The abstract RDF syntax (Consortium *et al.*, 2014) has several concrete implementations such as Turtle, (Beckett *et al.*, 2014) n-triples (Beckett, 2014) or XML (Gandon and Schreiber, 2014). For sake of readability, in the examples of this section we will use the Turtle syntax. In the last few years there have been several efforts to define extensions of RDF (e.g. for dealing with time (Gutierrez *et al.*, 2007) or trust values (Fionda and Greco, 2015)). Web resources can be referenced by using RDF terms.

Definition 1 (RDF terms). *The set of RDF terms is the union of three disjoint sets: the set of all Uniform Resource Identifiers (URIs) ($\mathcal{U}$), the set of all literals ($\mathcal{L}$) and the set of all blank nodes ($\mathcal{B}$).*

URIs are global identifiers. For example, <http://dbpedia.org/resource/Mouse> is used to identify the mouse mammal specie in the DBpedia (DBpedia is a crowd-sourced community effort to express Wikipedia information into RDF) knowledge base. (Lehmann *et al.*, 2015) Importantly, RDF does not take the Unique Name Assumption; this means that two different terms can refer to the same entity. For example, the URI <http://rdf.freebase.com/ns/m.04rmv> is used by Freebase (another data provider) to identify the mouse mammal specie. To avoid to write full URI strings, prefixes can be used. For example, by using the prefix `dbr` to denote the DBpedia knowledge base (i.e., @prefix dbr: <http://dbpedia.org/resource/>) we can simply write `dbr:Mouse`. Literals are a set of lexical values such as strings, numbers, and dates. Literals can be of two types: *lexical forms* that is a plain string (e.g. "Mouse") or *datatype IRI* that consist of a lexical string and a datatype (e.g., "1"^^xs: integer). Blank Nodes are defined as existential variables and denote some resource whose reference is not made explicit. Blank nodes are indicate by an underscore prefix (i.e., _:b1) and have local scope (i.e., limited to an RDF document).

Facts or statements about entities are encoded in RDF triples. An RDF triple is a 3-tuple of RDF terms, where the first element is called the *subject*, the second one the *predicate*, and the third element the *object*.

Definition 2: (RDF Triple). *An RDF triple t is defined as $t = (s, p, o)$ where $s \in \mathcal{U} \cup \mathcal{B}$ is called the subject, $p \in \mathcal{U}$ is called the predicate and $o \in \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$ is called the object.*

Informatively, the subject refers to the primary entity being described by the triple, the predicate identifies the relation between the subject and the object, which represents the value of the relation.

Example 3. The following example reports some RDF triples talking about species:

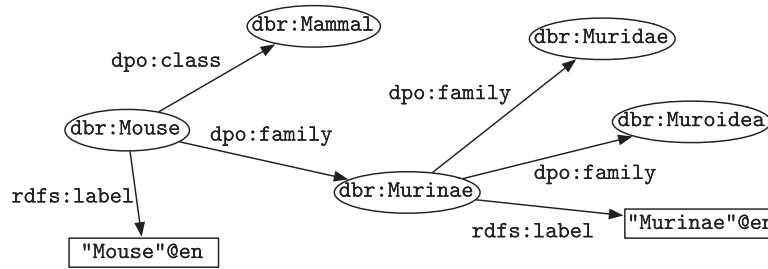
```
# PREFIX DECLARATIONS
@prefix dbr: <http://dbpedia.org/resource/>.
@prefix dbo: <http://dbpedia.org/ontology/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
# RDF TRIPLES
dbr: Mouse rdfs: label "Mouse"@en.
dbr: Mouse dbo: class dbr: Mammal.
dbr: Mouse dbo: family dbr: Murinae.
dbr: Murinae rdfs: label "Murinae"@en.
dbr: Murinae dbo: family dbr: Muridae.
dbr: Murinae dbo: family dbr: Muroidea.
```

□

Here we see three prefix declarations and then six RDF triples (delimited by periods). As discussed in [Definition 2](#), each triple is composed by three RDF terms. The subject contains URIs that identify the primary resource being described (in this case, `dbr:Mouse` and `dbr:Murinae`). The predicate position contains a URI that identifies the relation being described for that resource (e.g., `rdfs:label`, `dbo:family`). The object position contains URIs and literals that refer to the value for that relation (e.g., `"Mouse"@en`, `dbr:Muroidea`).

A set of RDF triples, as those in [Example 3](#), is graphically represented as directed labeled graph, where subjects and objects are drawn as labeled vertices and predicates are drawn as directed, labeled edges. By convention, literal vertices are drawn as rectangles, and URI vertices (and blank nodes) are drawn as ellipses.

Example 4. The following picture represents the RDF triples of [Example 3](#) as a directed labeled graph.



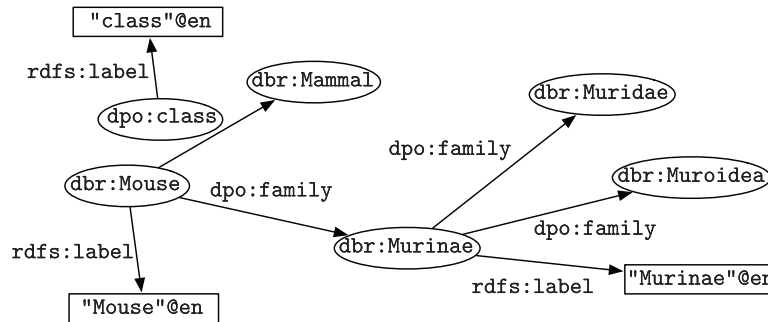
□

It is worth nothing that since edge labels are URIs they can themselves be vertices. This happens when a predicate URI also appears in a subject or object position.

Example 5. Consider the additional following triple to the set of triples reported in [Example 3](#):

```
...
dbo:class rdfs:label "class"@en.
```

The corresponding graphical representation is the following:



A set of RDF triples is formally referred to as an RDF graph.

□

Definition 6. (RDF Graph). A set of RDF triples $G \subseteq (U \cup B) \times U \times (U \cup B \cup \mathcal{L})$ is called an RDF graph.

Since RDF graphs are defined as sets of RDF triples, it follows that the ordering of triples does not matter; moreover, duplicate triples are not allowed.

The RDF data model provides a set of predefined terms, called RDF vocabulary, having as the common prefix the core RDF namespace (rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>). The most widely used term in the RDF vocabulary is `rdf:type`, which encodes the fact that a given resource belongs to a particular class. Via `rdf:type` resources sharing certain commonalities are assigned to the same classes.

Example 7. Consider the following example, three instances (resources) are assigned to two different classes:

```
@prefix dbr: <http://dbpedia.org/resource/>.
@prefix dbo: <http://dbpedia.org/ontology/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

dbr:Murinae rdf:type dbo:Species.
dbr:Murinae rdf:type dbo:Mammal.
dbr:Muridae rdf:type dbo:Species.
dbo:family rdf:type rdf:Property.
```

□

In general, the same resource can belong to multiple classes and each class can have multiple instances. The previous example shows in the last triple another RDF vocabulary term that is the `rdf:Property` class used to denote the set of all URIs that can appear in predicate position. For further details about the RDF standard the reader is referred to. [Consortium et al. \(2014\)](#), [Hogan \(2013\)](#).

RDF Semantics and RDF Schema

In the previous section we have covered the syntactic and structural aspects of RDF. This section describes how RDF data are *interpreted* by machines thus enabling the automation of some tasks based on the content of data.

RDF Semantics. The semantics of RDF has been defined in terms of a model-theoretic semantics ([Consortium et al., 2014](#); [Franconi et al., 2013](#)) providing a formal specification of when truth is preserved by transforming RDF data or deriving new RDF content. In particular, the RDF model theory defines RDF interpretations as possible configuration of the world and, in this setting, RDF triples are a means to make claims about the world. The more RDF triples the more specific is the world described. The interpretation provides the mathematical basis necessary to state what entities, the URIs identify in the world, what things in the world are related and by which relations (properties), what values (datatype) maps to, what entities the blank nodes refers to. The most important notion deriving from the RDF Semantics is that of *entailment*. Given an RDF graph that states a set of claims as true, the entailment helps in determining which other claims can be derived, as a consequence.

Example 8. Consider the RDF graph composed by the following RDF triples:

```
dbr:Mouse dbo:family dbr:Murinae.
dbr:Murinae dbo:family dbr:Muridae.
```

Since each RDF graph trivially entails all its subgraphs, the above RDF graph entails the following RDF graphs:

```
dbr:Mouse dbo:family dbr:Murinae.
```

and

```
dbr:Murinae dbo:family dbr:Muridae.
```

Moreover, due to the existential nature of blank nodes, it also entails the following RDF graph:

```
dbr:Mouse dbo:family dbr:Murinae.
dbr:Murinae dbo:family dbr:Muridae.
_:b1 dbo:family _:b2.
```

This last RDF graph says that there is something having some family.

□

Such form of entailment, that considers subgraphs or blank nodes, is called *simple entailment*. A more sophisticated form of entailment, called *RDF entailment*, builds upon the simple entailment.

Example 9. Consider the following RDF graph composed by the triple:

```
dbr:Mouse dbo:family dbr:Murinae.
```

This entails the following triple:

```
dbo:family rdf:type rdf:Property.
```

□

The above example shows a typical RDF entailment stating that any term appearing in the predicate position of a triple is a property, i.e., an instance of the class `rdf:Property`. The set of triples deriving from the RDF entailment can be automatically discovered by using inference rules. An inference rule has a premise, called body, and a conclusion, called head; its application generate a new conclusion (additional triples) every time the body is matched in the RDF graph.

Example 10. An inference rule to support the inference of [Example 9](#) is the following:

$?s ?p ?o \rightarrow ?p \text{ rdf:type } \text{rdf:Property}.$

□

In the above example `?s`, `?p` and `?o` are variables (variables are always prefixed with a '?'). Variables can match any RDF term in an RDF graph. In all inference rules, the variables that appear in the head must appear in the body, too. The semantics of RDF is based on the Open World Assumption (OWA), which assumes data incompleteness and all pieces of information that are not provided are unknown. In this settings, if an RDF term that appears in predicate position is not explicitly stated to be an `rdf:Property` this is not a problem. Indeed, due to the OWA, since data are incomplete a triple stating the type of such an RDF term can always be added. For further details about the RDF Semantics the interested reader is referred to [Consortium et al. \(2014\)](#), [Hogan \(2013\)](#)

RDF Schema. The RDF Schema (RDFS) specification ([Brickley et al., 2014](#)) has been proposed in 1998 to extend the RDF vocabulary. RDFS extends RDF with several key terms (see [Brickley et al. \(2014\)](#) for a complete list). The most important are the following four terms that are used to specify relationships among classes and properties: `rdfs:subClassOf` (`sc`), `rdfs:subPropertyOf` (`sp`), `rdfs:domain` (`dom`) and `rdfs:range` (`range`). `rdfs:subClassOf` is used to state that all the instances of one class are instances of another. `rdfs:subPropertyOf` is used to state that all resources related by one property are also related by another. `rdfs:domain` is used to state that any resource that is subject of a given property is an instance of one or more classes. `rdfs:range` is used to state that the objects of relations having a given property are instances of one or more classes. Among the other RDFS terms it should be mentioned the class `rdfs:Class`, which refers to the class of all classes (including itself).

The RDFS vocabulary is equipped with a model-theoretic semantics that allows to delineate RDFS entailment and define a set of RDFS inference rules. ([Consortium et al., 2014](#)). A selection of the most important (from a practical point of view) RDFS inference rules is reported in [Table 1](#); the full list is available in the RDF Semantics document 5.

Example 11. Consider the following RDF graph:

```

dbr:Mouse rdf:type dbo:Mammal.
dbr:Mus_nitidulus dbo:genus dbr:Mouse.
dbo:Mammal rdfs:subClassOf dbo:Animal.
dbo:Animal rdfs:subClassOf dbo:Eukaryote.
dbo:genus rdfs:domain dbo:Species.
dbo:Species rdfs:subClassOf owl:Thing.

```

The following triples can be RDFS-entailed from it:

```

dbo:Mammal rdfs:subClassOf dbo:Eukaryote. #rule 1 (b)
dbr:Mouse rdf:type dbo:Animal. #rule 1(a)
dbr:Mouse rdf:type dbo:Eukaryote. #rule 1(a)
dbr:Mus_nitidulus rdf:type dbo:Species. #rule 3 (a)
dbr:Mus_nitidulus rdf:type owl:Thing. #rule 3 (a)

```

Next to each inferred triple is indicated the rule from [Table 1](#) according to which it can be inferred. Note that the inference rules can also be applied to the inferred triples to infer further valid triples.

Table 1 A selection of the RDFS rule system

1. Subclass:

(a) $\frac{(?a.sc,?b) \quad (?c.type,?a)}{(?c.type,?b)} \quad (b) \quad \frac{(?a.sc,?b) \quad (?b.sc,?c)}{(?a.sc,?c)}$

2. Subproperty:

(a) $\frac{(?a.sp,?b) \quad (?c,?a,?d)}{(?c,?b,?d)} \quad (b) \quad \frac{(?a.sp,?b) \quad (?b.sp,?c)}{(?a.sp,?c)}$

3. Domain:

(a) $\frac{(?a.dom,?b) \quad (?c,?a,?d)}{(?c.type,?b)}$

4. Range:

(a) $\frac{(?a.range,?b) \quad (?c,?a,?d)}{(?d.type,?b)}$

RDFS entailment is layered on top of RDF entailment, which in turn is layered on top of simple entailment. Another form of entailment layered on top of RDF entailment and specified by the RDF Semantics document ([Consortium et al., 2014](#)) is datatype entailment or D-entailment. Such entailment has the purpose of mapping, for a set of datatypes, lexical strings to the values they denote. For example, the RDF term "25"^^xsd:decimal can be mapped to the value of the number 25.

The Web Ontology Language

The definition of Web Ontology Language (OWL) dates back to 2001 and the first version of the language was recognized as a W3C Recommendation ([McGuinness et al., 2004](#)) in 2004. Then, this first version (OWL 1) was subsequently extended by the OWL 2 W3C Recommendation ([Motik et al., 2009](#)). OWL 2 has a very similar overall structure to the first version OWL 1 and backward compatibility is kept (meaning that OWL1 ontologies are OWL 2 valid ontologies). However, OWL 2 adds new functionality with respect to OWL 1. Some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressiveness (e.g., property chains and asymmetric, reflexive, and disjoint properties). OWL 2 extends RDFS with the possibility to express additional constraints and is able to represent richer and more complex knowledge about things, groups of things, and relations between things. OWL 2 is a much more intricate language than RDFS and thus, in this article we only give a high-level overview and discuss only some relevant details. Like RDFS, OWL2 statements can be expressed as RDF triples ([Grau et al., 2008](#)). Indeed, OWL 2 reuses the core RDFS vocabulary and adds some specific predicates and objects with a particular meaning. In the following a subset of the novel language constructs introduced by OWL (1/2) is summarized:

Class disjointness constraints. The predicate owl:disjointWith allows for stating that two classes are disjoint, i.e., the sets of their members have an empty intersection. As an example, the following set of RDF triples

```
:Animal rdf:type owl:Class.
:Plant rdf:type owl:Class.
:Animal owl:disjointWith:Plant.
```

states that the classes Animal and Plant cannot have any individual in common.

Class equivalence constraints: the predicate owl:equivalentClass allows for stating that two classes are equivalent, i.e., the sets of their members are the same. As an example, the following set of RDF triples

```
:Human rdf:type owl:Class.
:Person rdf:type owl:Class.
:Human owl:disjointWith dbr:Person.
```

states that the classes Human and Person are equivalent, that is very individual of type Human is also of type Person.

Property disjointness constraints: the owl:disjointPropertyWith predicate allows for stating that two properties can never interlink the same two individuals (e.g., the properties hasParent and hasSpouse cannot relate the same two people). As an example, the following set of RDF triples

```
:hasSpouse rdf:type owl:Property.
:hasParent rdf:type owl:Property.
:hasSpouse owl:disjointPropertyWith dbr:hasParent.
```

states that the properties hasSpouse and hasParent are disjoint, meaning that if an RDF document contains the triple (:a :hasSpouse :b), it cannot contain the triple (:a :hasParent :b).

Property equivalence constraints: the owl:equivalentProperty predicate allows for stating that two properties relate precisely the same individuals (e.g., the properties parentOf and hasChild are said to relate the same resources). As an example, the following set of RDF triples

```
:parentOf rdf:type owl:Property.
:hasChild rdf:type owl:Property.
:parentOf owl:equivalentProperty dbr:hasChild.
```

states that the properties parentOf and hasChild are equivalent, meaning that if an RDF documents contains the triple (:a :parentOf :b), that the triple (:a :hasChild :b) can be inferred.

Individual equivalence constraints: the predicate owl:sameAs allows for stating that two resources refer to the same individual, thus, the information for one resource applies to the other. As an example, consider the following RDF triples:

```
dbr:Mouse rdfs:label "Mouse"@en.
dbr:Mouse dbo:class dbr:Mammal.
```

```
dbr:Mouse dbo:family dbr:Murinae.
dbr:Mouse owl:sameAs wikidata:Mus.
```

The last triple establishes an `owl:sameAs` relation to another RDF resource referring to the mouse species, published by an external exporter of RDF (in this example Wikidata). On the Wikidata site, we can find, among the other, the following triples about the mouse species:

```
wikidata:Mus wikidata:taxon_name "Mus"@en.
wikidata:Mus wikidata:parent_taxon wikidata:Murinae.
```

The `owl:sameAs` link provided by DBpedia states that the two URIs refer to the same thing, in this case the mouse species. Hence, information published under one of the URIs also applies to the other URI.

Individual inequality constraints: the `owl:differentFrom` predicate allows for stating that two resources necessarily refer to different individuals; as a consequence they cannot be in an `owl:sameAs` relation and a reasoner will always deduce that they refer to distinct individuals. As an example, consider the following RDF triples:

```
dbr:Mouse rdfs:label "Mouse"@en.
dbr:Mouse dbo:class dbr:Mammal.
dbr:Mouse dbo:family dbr:Murinae.
dbr:Mouse owl:differentFrom dbr:Yeast.
```

The `owl:differentFrom` link states that the two URIs refer to different things.

Property inverse constraints: the predicate `owl:inverseOf` allows for stating that one property relates the same things as another property, but in the opposite direction; that is, If the property P1 is stated to be the inverse of the property P2, then if X is related to Y by the P2 property, then Y is related to X by the P1 property. As an example, consider the following RDF triples:

```
:a :hasChild :b.
:hasChild owl:inverseOf :hasParent.
```

The `owl:inverseOf` link states that `:hasChild` is the inverse of `hasParent`. Thus, it is possible to infer the additional triple (`:b :hasParent :a`).

Property transitive constraints: the `owl:TransitiveProperty` predicate is used to state that a property is transitive; that is, if the pair (a,b) is an instance of a transitive property P, and the pair (b,c) is an instance of P, then the pair (a,c) is also an instance of P. As an example, consider the following RDF triples:

```
:a :ancestorOf :b.
:b :ancestorOf :c.
:ancestorOf rdf:type owl:TransitiveProperty.
```

The last triple states that `:ancestorOf` is a transitive property. Thus, it is possible to infer the additional triple (`:a:ancestorOf:c`).

Property symmetric constraints: the `owl:SymmetricProperty` predicate is used to state that a property is symmetric; that is, if the pair (a,b) is an instance of the symmetric property P, then the pair (b,a) is also an instance of P. As an example, consider the following RDF triples:

```
:a :friend :b.
:friend rdf:type owl:SymmetricProperty.
```

The last triple states that `:friend` is a symmetric property. Thus, it is possible to infer the additional triple (`:b :friend :a`).

Property functional constraints: the `owl:FunctionalProperty` predicate allows to state that a subject resource can have at most one value for such property; that is, `FunctionalProperty` is shorthand for stating that the property's minimum cardinality is zero and its maximum cardinality is 1. As an example, consider the following RDF triples:

```
:a :hasBiologicalFather :b.
:a :hasBiologicalFather :c.
:hasBiologicalFather rdf:type owl:FunctionalProperty.
```

The last triple states that `:hasBiologicalFather` is a functional property. Thus, it is possible to infer the additional triple (`:b owl:sameAs :c`).

Inverse functional constraints of properties: this type of constrain, expressed via the predicate `owl:InverseFunctionalProperty`, allows for stating that the inverse of a property is functional, meaning that the value of a property is unique to the subject of the relation

(e.g., ISBN values uniquely identify books). If two subjects share the same object for a given inverse-functional property it means that they refer to the same individual. As an example, consider the following RDF triples:

```
:b :isBiologicalFather :a.  
:c :isBiologicalFather :a.  
:isBiologicalFather rdfs:type owl:InverseFunctionalProperty.
```

The last triple states that `:isBiologicalFather` is an inverse functional property. Thus, it is possible to infer the additional triple (`:b owl:sameAs :c`).

Intentional class definition: A main feature of OWL is the intentional definition of new classes from existing ones; the keyword `owl:Restriction` is used for defining a new class by using some specific restriction properties.

Property and Cardinality restrictions: Allows to express restrictions on how properties can be used by the instances of a class and are used within the context of an `owl:Restriction`.

- The predicate `owl:someValuesFrom` allows to state that at least one value of the property `P` comes from a given class `C`. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction.  
_:a owl:onProperty :leads.  
_:a owl:someValuesFrom :Scientist.  
:Department rdfs:subClassOf :a.
```

The above triples constraint that every department must be led at least by one scientist.

- The predicate `owl:allValuesFrom` enables to restrict the range of a property for a given class. This means that for all individuals of such a class the value of such property must be an individual respecting the range restriction criteria. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction.  
_:a owl:onProperty :leads.  
_:a owl:allValuesFrom :Scientist.  
:Department rdfs:subClassOf :a.
```

The above triples constraint that every department can be led only by scientists.

- The predicates `owl:minCardinality` and `owl:maxCardinality` allow to indicate the minimum number and maximum number of relations of a given type that all individuals of a given class must have, respectively. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction  
_:a owl:onProperty :leads  
_:a owl:minCardinality 3  
_:b rdfs:subClassOf owl:Restriction  
_:b owl:onProperty :leads  
_:b owl:maxCardinality 6  
:Department rdfs:subClassOf :a.  
:Department rdfs:subClassOf :b.
```

The above triples constraint that every department can have at least 3 and at most six leaders.

- The predicate `owl:cardinality` allows to indicate the exact number of relations of a given type that all individuals of a given class must have. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction  
_:a owl:onProperty :leads  
_:a owl:cardinality 3  
:Department rdfs:subClassOf :a.
```

The above triples constraint that every department must have exactly 3 leaders.

Set operations over classes: OWL contains some set-based constructors that allow to combine classes.

- The constructor `owl:intersectionOf` enables to specify the intersection of classes; the intersection of classes denotes the individuals that belong to both classes. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction
_:a owl:onProperty :teachesTo
_:a owl:someValuesFrom :Graduated
_:b owl:intersectionOf (:Professor, :Lecturer)
_:a rdfs:subClassOf _:b
```

The above triples constraint that only professors and lecturers may teach to graduate students.

- The constructor `owl:unionOf` allows to specify the union of classes; the union of classes denotes the individuals that belong to some classes. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction
_:a owl:onProperty :teachesTo
_:a owl:someValuesFrom :Undergraduated
_:b owl:unionOf (:Professor, :Lecturer)
_:a rdfs:subClassOf _:b
```

The above triples constraint that both professors and lecturers may teach to undergraduate students.

- The constructor `owl:complementOf` enables to specify the complement of a class; the complement of a class denotes the individuals that do not belong to the class. As an example, consider the following RDF triples:

```
_:a rdfs:subClassOf owl:Restriction
_:a owl:onProperty :teachesTo
_:a owl:someValuesFrom :Undergraduated
_:b owl:complement (:Professor)
_:a rdfs:subClassOf _:b
```

The above triples constraint that professors cannot teach to undergraduate students.

In **Table 2** some of the inference rules that correspond to such OWL features (and are a subset of the complete list of the OWL 2 RL/RDF rules ([Motik et al., 2009](#))) are reported. In the table, the head of the rule FALSE indicates that data are inconsistent.

Now, we give an example of how the above inference rules are applied. Consider the following two RDF graphs:

```
dbr:Mouse rdfs:label "Mouse"@en.
dbr:Mouse dbo:class dbr:Mammal.
dbr:Mouse dbo:family dbr:Murinae.
dbr:Mouse owl:sameAs wikidata:Mus.
```

and

```
wikidata:Mus wikidata:taxon_name "Mus"@en.
wikidata:Mus wikidata:parent_taxon wikidata:Murinae.
```

The `owl:sameAs` link provided in the first graph states that the two URIs `dbr:Mouse` and `wikidata:Mus` refer to the same thing. Hence, information published about `dbr:Mouse` or about `wikidata:Mus` also applies to the other URI. This feature of OWL is axiomatized by rules `eq-trans`, `eq-rep-s`, `eq-rep-p` and `eq-rep-o` in the above Table. Applying these rules, one can see that the above two graphs entail:

```
dbr:Mouse wikidata:taxon_name "Mus"@en.
dbr:Mouse wikidata:parent_taxon wikidata:Murinae.
wikidata:Mus owl:sameAs dbr:Mouse.
wikidata:Mus rdfs:label "Mouse"@en.
wikidata:Mus dbo:class dbr:Mammal.
wikidata:Mus dbo:family dbr:Murinae.
wikidata:Mus owl:sameAs wikidata:Mus.
```

Among the typical reasoning tasks that can be performed over an OWL ontology there are consistency checking (checking if the data lead to some inconsistency), satisfiability checking (checking if a class can have a member without causing a logical contradiction), subsumption checking (checking if a class is necessarily a sub-class of another), instance checking (checking if a resource is a member of a class) and conjunctive query answering (posing complex queries against the ontology and its entailments). To solve this

Table 2 A subset of OWL inference rules

Name	Rule
eq-sym	$?x \text{ owl:sameAs } ?y. \rightarrow ?y \text{ owl:sameAs } ?x.$
eq-trans	$?x \text{ owl:sameAs } ?y. ?y \text{ owl:sameAs } ?z. \rightarrow ?x \text{ owl:sameAs } ?z.$
eq-rep-s	$?s \text{ owl:sameAs } ?x. ?s ?p ?o. \rightarrow ?x ?p ?o.$
eq-rep-p	$?p \text{ owl:sameAs } ?x. ?s ?p ?o. \rightarrow ?s ?x ?o.$
eq-rep-o	$?o \text{ owl:sameAs } ?x. ?s ?p ?o. \rightarrow ?s ?p ?x.$
eq-diff1	$?x \text{ owl:sameAs } ?y. ?x \text{ owl:differentFrom } ?y. \rightarrow \text{FALSE}$
prp-dom	$?p \text{ dom } ?c. ?x ?p ?y. \rightarrow ?x \text{ rdf:type } ?c.$
prp-rng	$?p \text{ rng } ?c. ?x ?p ?y. \rightarrow ?y \text{ rdf:type } ?c.$
prp-fp	$?p \text{ a FP. } ?x ?p ?y_1, ?y_2. \rightarrow ?y_1 \text{ owl:sameAs } ?y_2.$
prp-ifp	$?p \text{ a IFP. } ?x_1 ?p ?y. ?x_2 ?p ?y. \rightarrow ?x_1 \text{ owl:sameAs } ?x_2.$
prp-symp	$?p \text{ rdf:type owl:SymmetricProperty. } ?x ?p ?y. \rightarrow ?y ?p ?x.$
prp-trp	$?p \text{ rdf:type owl:TransitiveProperty. } ?x ?p ?y. ?y ?p ?z. \rightarrow ?x ?p ?z.$
prp-spo1	$?x \text{ rdfs:subPropertyOf } ?y. ?s ?x ?o. \rightarrow ?s ?y ?o.$
prp-eqp1	$?x \text{ owl:equivalentProperty } ?y. ?s ?x ?o. \rightarrow ?s ?y ?o.$
prp-eqp2	$?x \text{ owl:equivalentProperty } ?y. ?s ?y ?o. \rightarrow ?s ?x ?o.$
prp-pdw	$?x \text{ owl:propertyDisjointWith } ?y. ?s ?x ?o ; ?s ?y ?o. \rightarrow \text{FALSE}$
prp-inv1	$?x \text{ owl:inverseOf } ?y. ?s ?x ?o. \rightarrow ?o ?y ?s.$
prp-inv2	$?x \text{ owl:inverseOf } ?y. ?s ?y ?o. \rightarrow ?o ?x ?s.$
cls-com	$?x \text{ owl:complementOf } ?y. ?s \text{ rdf:type } ?x. ?s \text{ rdf:type } ?y \rightarrow \text{FALSE}.$
cls-svf1	$?x \text{ owl:someValuesFrom } ?y. ?x \text{ owl:onProperty } ?p. ?u ?p ?v. ?v \text{ rdf:type } ?y \rightarrow ?u \text{ rdf:type } ?x.$
cls-avf	$?x \text{ owl:allValuesFrom } ?y. ?x \text{ owl:onProperty } ?p. ?u ?p ?v. ?u \text{ rdf:type } ?x \rightarrow ?v \text{ rdf:type } ?y.$
cax-sco	$?x \text{ rdfs:subClassOf } ?y. ?s \text{ rdf:type } ?x. \rightarrow ?s \text{ rdf:type } ?y.$
cax-ecq1	$?x \text{ owl:equivalentClass } ?y. ?s \text{ rdf:type } ?x. \rightarrow ?s \text{ rdf:type } ?y.$
cax-ecq2	$?x \text{ owl:equivalentClass } ?y. ?s \text{ rdf:type } ?y. \rightarrow ?s \text{ rdf:type } ?x.$
cax-dw	$?x \text{ owl:disjointWith } ?y. ?s \text{ rdf:type } ?x. ?s \text{ rdf:type } ?y. \rightarrow \text{FALSE}$
scm-sco	$?x \text{ rdfs:subClassOf } ?y. ?y \text{ rdfs:subClassOf } ?z. \rightarrow ?x \text{ rdfs:subClassOf } ?z.$
scm-ecq1	$?x \text{ owl:equivalentClass } ?y \rightarrow ?x \text{ rdfs:subClassOf } ?y. ?y \text{ rdfs:subClassOf } ?x.$
scm-ecq2	$?x \text{ rdfs:subClassOf } ?y. ?y \text{ rdfs:subClassOf } ?x. \rightarrow ?x \text{ owl:equivalentClass } ?y.$
scm-spo	$?p_1 \text{ rdfs:subPropertyOf } ?p_2. ?p_2 \text{ rdfs:subPropertyOf } ?p_3. \rightarrow ?p_1 \text{ rdfs:subPropertyOf } ?p_3.$
scm-eqp1	$?p_1 \text{ owl:equivalentProperty } ?p_2 \rightarrow ?p_1 \text{ rdfs:subPropertyOf } ?p_2. ?p_2 \text{ rdfs:subPropertyOf } ?p_1.$
scm-eqp2	$?p_1 \text{ rdfs:subPropertyOf } ?p_2. ?p_2 \text{ rdfs:subPropertyOf } ?p_1. \rightarrow ?p_1 \text{ owl:equivalentProperty } ?p_2.$

reasoning tasks OWL defines two standard and compatible semantics. The first semantics is called the “RDF-Based Semantics”, and it is backwards-compatible with RDF. However, all the above reasoning tasks are undecidable. This means that it cannot be guaranteed that the automated reasoning process ever terminates. The second semantics, called the Direct Semantics, can only interpret OWL ontologies satisfying certain restrictions. These restrictions are such that certain reasoning tasks are known to be decidable. This introduces a trade of some expressive power of the language (features it supports) for the efficiency of reasoning. OWL 2 define three profiles (Motik *et al.*, 2009) that achieve efficiency in a different way and are useful in different application scenarios.

OWL 2 EL. This was the first OWL profile to be defined and is particularly useful in applications employing ontologies that contain very large numbers of properties and/or classes. The basic reasoning problems for this profile can be performed in time that is polynomial with respect to the size of the ontology. As an example, OWL 2 EL, among the others, does not allow to use inverse or symmetric properties.

OWL 2 QL. This was the second OWL profile and is aimed at applications that use very large volumes of instance data where query answering is the most important reasoning task. Query answering is solved by rewriting queries over relational database systems.

OWL 2 RL. This is the third OWL profile and is aimed at applications that require scalable reasoning without sacrificing too much expressive power. The ontology consistency, class expression satisfiability, class expression, subsumption, instance checking, and conjunctive query answering problems can be solved in time that is polynomial with respect to the size of the ontology.

Concluding Remarks

Ontologies are a key ingredient toward modeling and reasoning over knowledge domains. In this article we gave an overview of the ontology language defined under the W3C (the World Wide Web Consortium) umbrella. We have started from RDF, a simple language to express statements, and reached OWL, which offers very sophisticated reasoning capabilities. Instead of focusing on too much technicalities and formalism, our treatment gave special attention to concrete examples. This approach will hopefully allow to better grasp the core ideas beyond these ontology languages thus making them useful in concrete application scenarios.

See also: Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Ontology: Introduction

References

- Beckett, D., 2014. Rdf 1.1 n-triples. W3C Recommendation.
- Beckett, D., Berners-Lee, T., Prud'hommeaux, E., Carothers, G., 2014. Rdf 1.1 turtle-terse rdf triple language W3C Recommendation.
- Brickley, D., Guha, R.V., McBride, B., 2014. Rdf schema 1.1 W3C Recommendation 25, 2004-2014.
- Consortium, W.W.W. *et al.*, 2014. Rdf 1.1 concepts and abstract syntax.
- Consortium, W.W.W. *et al.*, 2014. Rdf 1.1 semantics.
- Fionda, V., Greco, G., 2015. Trust models for rdf data: Semantics and complexity. In: AAAI, pp. 95–101.
- Franconi, E., Gutierrez, C., Mosca, A., Pirro, G., Rosati, R., 2013. The logic of extensional RDFS. In: International Semantic Web Conference, Springer.
- Gandon, F., Schreiber, G., 2014. Rdf 1.1 xml syntax W3C Recommendation 25.
- Grau, B.C., Horrocks, I., Parsia, B., Rutenberg, A., Schneider, M., 2008. Owl 2 web ontology language: Mapping to rdf graphs.
- Gutierrez, C., Hurtado, C.A., Vaisman, A., 2007. Introducing time into rdf. IEEE Transactions on Knowledge and Data Engineering 19 (2).
- Hogan, A., 2013. Linked data and the semantic web standards. Linked Data and the Semantic Web Standards. Chapman and Hall/CRC Press.
- Lehmann, J., Isele, R., Jakob, M., *et al.*, 2015. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web 6 (2), 167–195.
- McGuinness, D.L., Van Harmelen, F., *et al.* 2004. Owl web ontology language overview. W3C Recommendation 10(10) 2004.
- Motik, B., Grau, B.C., Horrocks, I., *et al.* 2009. Owl 2 web ontology language profiles. W3C Recommendation 27, 61.

Relevant Websites

<http://dbpedia.org/resource/DBpedia>.
DBpedia.

<http://rdf.freebase.com/ns/>
Freebase API.

Ontology: Querying Languages and Development

Valeria Fionda, University of Calabria, Rende, Italy

Giuseppe Pirrò, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The use of ontologies as a support to model knowledge domains and enable a new form of knowledge-based services has significantly increased in the last decade. Ontologies represent an underlying knowledge support for modeling domains (Gruber *et al.*, 1993), and thus is of utmost importance the definition of ontology-building methodologies (Fernández-López *et al.*, 1997). Ontologies have played a key role in a variety of tasks from knowledge management/retrieval (e.g., ontology based data access (Calvanese *et al.*, 2011)) to distributed systems (e.g., ontology-based query routing (Pirrò *et al.*, 2009) or resource finding (Atencia *et al.*, 2011)). A very prominent example of the importance of ontologies derives from the Semantic Web project (Berners-Lee *et al.*, 2001).

In this context new problems related to the creation and management of ontologies raised along with new opportunities to use ontologies in a variety of tasks including querying and reasoning. A significant contribution to the spread of ontologies has been given by standardization bodies such as the World Wide Web Consortium (W3C) (see Relevant Website section). Nowadays, ontologies languages like RDF, RDFS (Franconi *et al.*, 2013), and OWL have now reached a significant level of maturity that make them suitable for the usage in different practical contexts. To support the ontology management a plethora of tools has been defined (Perez *et al.*, 2002). Retrieval and reasoning are among the most two important functionalities enabled by ontologies.

Focusing on these two aspects, in this article, we first discuss the SPARQL language (Section SPARQL), a W3C standard for querying RDF data and then illustrate how it can also be used to query an ontology. Then, we provide (Section Ontology Management Tools) a high level overview of about some popular ontology management tools. We conclude in Section Concluding Remarks.

SPARQL

The SPARQL standard defines a query language proposed specifically for RDF data. The first proposal for the SPARQL specification became a W3C Recommendation in 2008 (Prud'hommeaux and Seaborne, 2008). Then, in 2013, an extension of the standard, called SPARQL 1.1, became a W3C Recommendation (Harris *et al.*, 2013). The SPARQL language is orthogonal to RDFS and OWL; indeed, it is built directly on top of the RDF data model and does not provide direct support for inferencing. Before introducing SPARQL we introduce the following notation; I is the set of all URIs, L be the set of all literals and $\mathcal{V}$ be a countably infinite set of variables, such that $\mathcal{V} \cap (I \cup L) = \emptyset$. Variables in SPARQL are string preceded by the ? symbol.

SPARQL is similar in spirit and structure to the Structured Query Language (SQL) used for querying relational databases, but designed to work specifically on RDF data. In what follows we will discuss both the syntax and semantics of SPARQL queries. The SPARQL semantics is formalized along the lines of Angles and Gutierrez (Angles and Gutierrez, 2016).

By analyzing SPARQL queries on a high level of details, we can identify five main parts: (i) **Prefix Declarations** that allow for defining URI prefixes that can be used as shortcuts in a SPARQL query; (ii) **Dataset Clause** that allows for specifying the dataset over which the query is to be executed; (iii) **Result Clause** that allows to specify the type of SPARQL query and what results should be returned; (iv) **Query Clause** that allows to specify the query patterns; (v) **Solution Modifiers** that allows to express some ordering and partitioning of the results.

Example 1. Consider the SPARQL query reported below. Comments lines are prefixed with a sharp ('#'). The first block defines the prefixes that can be used later in the query. The second block (Dataset Clause) specifies the dataset on which the query should be executed: in this case an RDF document on the DBpedia site about the mouse species. The third block (the Result Clause) allows to define which kind of results should be returned (in this case the keyword *DISTINCT* forces the result pairs matching the variables *?class* and *?family* to be unique). The fourth block (Query Clause) defines the patterns that the query should match against the data. In this example we are looking for the values of the *dbo: class* and *dbo: family* of *dbr: Mouse*. The last block (Solution Modifiers) is used to set a limit on the number of results returned via the keyword *LIMIT*.

```
# PREFIX DECLARATIONS
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

# DATASET CLAUSE
FROM <http://dbpedia.org/data/Mouse.xml>

# RESULT CLAUSE
SELECT DISTINCT ?class ?family
```

```
# QUERY CLAUSE
WHERE {
  dbr:Mouse dbo:class ?class.
  dbr:Mouse dbo:family ?family.
}
```

```
# SOLUTION MODIFIER
LIMIT 2
```

Assuming the existence of the following triples:

```
dbr : Mouse dbo : class dbr : Mammal.
dbr : Mouse dbo : family dbr : Murinae.
dbr : Mouse dbo : family dbr : Muridae.
dbr : Mouse dbo : family dbr : Muroidea.
```

in the RDF graph `<http://dbpedia.org/data/Mouse.xml>`, a result for the above query is the binding of the variables `?class` and `?family` as follows:

<code>?class</code>	<code>?family</code>
dbr : Mouse	dbr : Murinae
dbr : Mouse	dbr : Muridae

where the header indicates the variables for which the respective terms are bound in the results. □

In what follows the Dataset Clause, Result Clause, Query Clause and Solution Modifiers are detailed.

Query Types

A SPARQL query can be one of the following four types:

SELECT [DISTINCT]: the result is a list of bindings for variables specified in the query. By default the result can contain duplicate bindings; the use of the keyword **DISTINCT** has the effect of eliminating duplicates.

ASK: the result is a boolean value indicating whether or not there was a match in the data for the specified query.

CONSTRUCT: the result is an RDF graph (instead of a (multi)set of bindings). An RDF template is provided where variable bindings can be inserted.

DESCRIBE: the result is the RDF description of a particular RDF term. Common types of descriptions are all the triples in which the RDF term appears at (i) any position, (ii) as a subject, or (iii) as a subject or object.

Dataset Clause and Named Graphs

In general, a SPARQL dataset is composed by several RDF graphs.

Definition 2. A SPARQL named graph is a pair (u, G) where $u \in I$ is a URI representing the name of the RDF graph G .

A SPARQL dataset is then composed of a default graph, which has no name, and a set of named graphs.

Definition 3. A SPARQL dataset $D = \{G, (u_1, G_1), \dots, (u_n, G_n)\}$ is a set of RDF graphs where $u_1, \dots, u_n$ are distinct URIs and $G, G_1, \dots, G_n$ are RDF graphs. The unnamed graph G is the default graph. Each pair (u_i, G_i) is a named graph.

Since a SPARQL dataset is composed by several graphs, it is possible to query different partitions of it, by using the names of the individual graphs. The selection of the partition to use is done in the dataset clause of the query, by using one of the two following optional features:

- **FROM:** this keyword is used to specify the default graph;
- **FROM NAMED:** this keyword is used to specify a set of named graphs via their URIs.

Query patterns in the query clause are matched against the default graph. If another graph should be used it must be explicitly declared by using in the query clause the keyword **GRAPH**. For queries that do not contain any dataset clause, usually SPARQL query engines use as a default graph the union of all the RDF graphs in the dataset.

Query Clause

The query clause specifies the SPARQL graph patterns that query variables must match. It starts with the keyword **WHERE** and is surrounded by curly brackets.

Definition 4. A SPARQL graph pattern is defined recursively as follows:

- A triple from $(I \cup L \cup V) \times (I \cup V) \times (I \cup L \cup V)$ is a graph pattern called a triple pattern. (We do not consider blank nodes since in RDF triple patterns they are handled as existential variables.).
- If $\mathcal{P}_1$ and $\mathcal{P}_2$ are graph patterns then $(\mathcal{P}_1 \text{ AND } \mathcal{P}_2)$, $(\mathcal{P}_1 \text{ UNION } \mathcal{P}_2)$, $(\mathcal{P}_1 \text{ OPTIONAL } \mathcal{P}_2)$, $(\mathcal{P}_1 \text{ MINUS } \mathcal{P}_2)$ and $(\mathcal{P}_1 \text{ NOT-EXISTS } \mathcal{P}_2)$ are graph patterns.
- If $\mathcal{P}_1$ is a graph pattern and C is a filter constraint (as defined below) then $(\mathcal{P}_1 \text{ FILTER } C)$ is a graph pattern.

Triple patterns and basic graph patterns

Triple patterns can be executed against RDF graphs to produce a (multi)set of solution mappings. Given a triple pattern t we next indicate with $\text{var}(t)$ the set of variables that appear in t .

Definition 5. A (solution) mapping μ is a partial function $\mu : V \rightarrow I \cup L$, where $\text{dom}(\mu)$ indicates the set of variables over which the mapping is defined.

Each mapping applied to the triple pattern returns a triple belonging to the RDF graph. Given a triple pattern t and a mapping μ such that $\text{var}(t) \subseteq \text{dom}(\mu)$, we denote by $\mu(t)$ the triple obtained by replacing the variables in t according to μ . Overloading the above definition, we denote by $\mu(\mathcal{P})$ the graph pattern obtained by the recursive substitution of variables in every triple pattern and filter constraint occurring in the graph pattern $\mathcal{P}$ according to μ . The *empty mapping*, denoted μ_0 , is the mapping satisfying that $\text{dom}(\mu_0) = \emptyset$.

A set of triple patterns is called basic graph pattern (BGP). An example of BGP is reported in Example 1, where two triple patterns (i.e., `dbr:Mouse dbo:class ?class` and `dbr:Mouse dbo:family ?family`) are embedded in the WHERE clause. The set of triple patterns included in a basic graph pattern are considered as a conjunction. This conjunction leads to a join operation over compatible mappings belonging to the sets of mappings produced by evaluating individual triple patterns.

Definition 6. Two mappings μ_1 and μ_2 are compatible (resp., not compatible), denoted by $\mu_1 \sim \mu_2$ (resp., $\mu_1 \not\sim \mu_2$), if $\mu_1(?X) = \mu_2(?X)$ for all variables $?X \in (\text{dom}(\mu_1) \cap \text{dom}(\mu_2))$ (resp., if $\mu_1(?X) \neq \mu_2(?X)$ for some $?X \in (\text{dom}(\mu_1) \cap \text{dom}(\mu_2))$).

If $\mu_1 \sim \mu_2$ then we write $\mu_1 \cup \mu_2$ for the mapping obtained by extending μ_1 according to μ_2 on all variables in $\text{dom}(\mu_2) \setminus \text{dom}(\mu_1)$. Note that two mappings with disjoint domains are always compatible, and that the empty mapping μ_0 is compatible with any other mapping.

Given a finite set of variables $W \subseteq V$, the restriction of a mapping μ to W , denoted $\mu|_W$, is a mapping μ' satisfying that $\text{dom}(\mu') = \text{dom}(\mu) \cap W$ and $\mu'(?X) = \mu(?X)$ for every $?X \in \text{dom}(\mu) \cap W$.

Filter constraints

A filter constraint is defined recursively as follows:

1. If $?X, ?Y \in V$ and $c \in I \cup L$ then $(?X=c)$, $(?X=?Y)$ and $\text{bound} (?X)$ are *atomic filter constraints*;
2. If C_1 and C_2 are filter constraints then $(!C_1)$, $(C_1 \parallel C_2)$ and $(C_1 \&\& C_2)$ are *complex filter constraints*.

Given a filter constraint C , we denote by $f(C)$ the selection formula obtained from C . A *selection formula* is defined recursively as follows: (i) If $?X, ?Y \in V$ and $c \in I \cup L$ then $(?X=c)$, $(?X=?Y)$ and $\text{bound} (?X)$ are atomic selection formulas; (ii) If F and F' are selection formulas then $(F \wedge F')$, $(F \vee F')$ and $\neg(F)$ are boolean selection formulas. The evaluation of a selection formula F under a mapping μ , denoted $\mu(F)$, is defined in a three-valued logic (i.e. with values *true*, *false*, and *error*) as follows:

- If F is $?X=c$ and $?X \in \text{dom}(\mu)$, then $\mu(F) = \text{true}$ when $\mu(?X) = c$ and $\mu(F) = \text{false}$ otherwise. If $?X \notin \text{dom}(\mu)$ then $\mu(F) = \text{error}$.
- If F is $?X=?Y$ and $?X, ?Y \in \text{dom}(\mu)$, then $\mu(F) = \text{true}$ when $\mu(?X) = \mu(?Y)$ and $\mu(F) = \text{false}$ otherwise. If either $?X \notin \text{dom}(\mu)$ or $?Y \notin \text{dom}(\mu)$ then $\mu(F) = \text{error}$.
- If F is $\text{bound} (?X)$ and $?X \in \text{dom}(\mu)$ then $\mu(F) = \text{true}$ else $\mu(F) = \text{false}$.
- If F is a complex selection formula then it is evaluated following the three-valued logic presented in [Table 1](#).

Note that there exists a simple and direct translation from filter constraints to selection formulas and viceversa.

Combining graph patterns

On top of Basic Graph Patterns and Filter constraints, SPARQL defines six other core features that can be used to create complex query patterns in query clauses:

- **GRAPH:** If used in conjunction with a URI, this keyword specifies a named graph against which a graph pattern should be matched. If used in conjunction with a variable, such a variable is used to bind the named graph for which the graph pattern is matched.
- **AND:** (In the SPARQL syntax the keyword AND is omitted.) Allows the definition of a conjunction of graph patterns that the query should match. The result is the join of the solution mappings generated for each conjunctive query pattern.
- **UNION:** Allows the definition of a disjunction of graph patterns that the query should match. The result is the union of the solution mappings generated for each disjunctive query pattern.

Table 1 Three-valued logic for evaluating selection formulas

p	q	$p \wedge q$	$p \vee q$		
true	true	true	true		
true	false	false	true		
true	error	error	true		
false	true	false	true	p	$\neg p$
false	false	false	false	true	false
false	error	false	error	false	true
error	true	error	true	error	error
error	false	false	error		
error	error	error	error		

- **OPTIONAL**: It is used to specify optional graph patterns. If an optional graph pattern cannot be matched the variables that are unique in the optional pattern are mapped to UNBOUND (a SPARQL keyword denoting the fact that a variable is not bound).
- **MINUS**: it is used to specify negation. This operator evaluates both its arguments, then calculates solutions in the left-hand side that are not compatible with any solution on the right-hand side.
- **NOT EXISTS**: it is another type of negation that filter expression tests whether a graph pattern does not match the dataset, given the values of variables in the graph pattern in which the filter occurs.

Property paths

Property paths (PPs) have been incorporated into the SPARQL 1.1 standard with two main motivations; first, to provide explicit graph navigational capabilities thus allowing the writing of SPARQL navigational queries in a more succinct way; second, to introduce transitive closure $*$ previously not available in SPARQL. The design of PPs was influenced by earlier proposals (e.g., nSPARQL (Pérez et al., 2010)) and inspired several extensions (e.g., Extended Property Paths (Fionda et al., 2015), Property Paths over the Web (Hartig and Pirrò, 2015), and languages for the Web (Fionda et al., 2015)).

Definition 7. A property path pattern (or PP pattern for short) is a tuple $\mathcal{P} = \langle \alpha, \text{elt}, \beta \rangle$ with $\alpha \in (I \cup L \cup \mathcal{V})$, $\beta \in (I \cup L \cup \mathcal{V})$, and elt is a property path expression (PP expression) that is defined by the following grammar (where $u, u_1, \dots, u_n \in I$):

$$\text{elt} := u | (u_1 | \dots | u_n) | !(\wedge u_1 | \dots | \wedge u_n) | !(\wedge u_1 | \dots | \wedge u_j | \dots | \wedge u_q | \dots | \wedge u_n) | \text{elt} / \text{elt} | (\text{elt} | \text{elt}) | (\text{elt})^* | \wedge \text{elt}$$

The SPARQL standard introduces additional types of PP expressions (Harris et al., 2013); since these are merely syntactic sugar (they are defined in terms of expressions covered by the grammar given above), we ignore them in this article. As another slight deviation from the standard, we do not consider blank nodes in PP patterns. PP patterns with blank nodes can be simulated using fresh variables. The SPARQL standard distinguishes between two types of property path expressions: *connectivity patterns* or recursive PPs that include closure ($*$), and *syntactic short forms* or non-recursive PPs (nPPs) that do not include it. As for the evaluation of PPs, the W3C spec. informally mentions the fact that nPPs can be evaluated via a translation into equivalent SPARQL basic expressions (see Harris et al., 2013, Section 9.3). Property path patterns can be combined with graph patterns inside SPARQL patterns (using PP expressions in the middle position of a pattern).

SPARQL semantics

The evaluation of a SPARQL graph pattern $\mathcal{P}$ over an RDF graph G is defined as a function $\llbracket \mathcal{P} \rrbracket_G$ which returns a multiset of solution mappings. We use the symbol Ω to denote a multiset and $\text{card}(\mu, \Omega)$ to denote the cardinality of the mapping μ in the multiset Ω . Moreover, it applies that $\text{card}(\mu, \Omega) = 0$ when $\mu \notin \Omega$. We use Ω_0 to denote the multiset $\{\mu_0\}$ such that $\text{card}(\mu_0, \Omega_0) > 0$ (Ω_0 is called the join identity). The domain of a solution mapping Ω is defined as $\text{dom}(\Omega) = \cup_{\mu \in \Omega} \text{dom}(\mu)$. The SPARQL algebra for multisets of mappings is composed of the operations of projection, selection, join, difference, left-join, union and minus. Let Ω_1, Ω_2 be multisets of mappings, W be a set of variables and F be a selection formula.

Definition 8. (Operations over multisets of mappings). Let Ω_1 and Ω_2 be multiset of mappings, then:

- Projection:** $\pi_W(\Omega_1) = \{\mu' | \mu \in \Omega_1, \mu' = \mu|_W\}$ where $\text{card}(\mu', \pi_W(\Omega_1)) = \sum_{\mu' = \mu|_W} \text{card}(\mu, \Omega_1)$.
- Selection:** $\sigma_F(\Omega_1) = \{\mu \in \Omega_1 | \mu(F) = \text{true}\}$ where $\text{card}(\mu, \sigma_F(\Omega_1)) = \text{card}(\mu, \Omega_1)$.
- Union:** $\Omega_1 \cup \Omega_2 = \{\mu | \mu \in \Omega_1 \vee \mu \in \Omega_2\}$ where $\text{card}(\mu, \Omega_1 \cup \Omega_2) = \text{card}(\mu, \Omega_1) + \text{card}(\mu, \Omega_2)$.
- Join:** $\Omega_1 \bowtie \Omega_2 = \{\mu = (\mu_1 \cup \mu_2) | \mu_1 \in \Omega_1, \mu_2 \in \Omega_2, \mu_1 \sim \mu_2\}$, $\text{card}(\mu, \Omega_1 \bowtie \Omega_2) = \sum_{\mu = (\mu_1 \cup \mu_2)} \text{card}(\mu_1, \Omega_1) \times \text{card}(\mu_2, \Omega_2)$.
- Difference:** $\Omega_1 \setminus_F \Omega_2 = \{\mu_1 \in \Omega_1 | \forall \mu_2 \in \Omega_2, (\mu_1 \sim \mu_2) \vee (\mu_1 \sim \mu_2 \wedge (\mu_1 \cup \mu_2)(F) = \text{false})\}$ where $\text{card}(\mu_1, \Omega_1 \setminus_F \Omega_2) = \text{card}(\mu_1, \Omega_1)$.
- Minus:** $\Omega_1 - \Omega_2 = \{\mu_1 \in \Omega_1 | \forall \mu_2 \in \Omega_2, \mu_1 \sim \mu_2 \vee \text{dom}(\mu_1) \cap \text{dom}(\mu_2) = \emptyset\}$ where $\text{card}(\mu_1, \Omega_1 - \Omega_2) = \text{card}(\mu_1, \Omega_1)$.
- Left Join:** $\Omega_1 \bowtie_L \Omega_2 = \sigma_F(\Omega_1 \bowtie \Omega_2) \cup (\Omega_1 \setminus_F \Omega_2)$ where $\text{card}(\mu, \Omega_1 \bowtie_L \Omega_2) = \text{card}(\mu, \sigma_F(\Omega_1 \bowtie \Omega_2)) + \text{card}(\mu, \Omega_1 \setminus_F \Omega_2)$.

Let $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ be graph patterns and C be a filter constraint. The evaluation of a graph pattern $\mathcal{P}$ over a graph G is defined recursively as follows:

- If $\mathcal{P}$ is a triple pattern t_p then $\llbracket t_p \rrbracket_G = \{\mu \mid \text{dom}(\mu) = \text{var}(t_p) \text{ and } \mu(t_p) \in G\}$ where $\text{var}(t_p)$ is the set of variables in t_p and the cardinality of each mapping is 1.
- If $\mathcal{P} = (\mathcal{P}_1 \text{ AND } \mathcal{P}_2)$, then $\llbracket \mathcal{P} \rrbracket_G = \llbracket \mathcal{P}_1 \rrbracket_G \bowtie \llbracket \mathcal{P}_2 \rrbracket_G$
- If $\mathcal{P} = (\mathcal{P}_1 \text{ UNION } \mathcal{P}_2)$, then $\llbracket \mathcal{P} \rrbracket_G = \llbracket \mathcal{P}_1 \rrbracket_G \sqcup \llbracket \mathcal{P}_2 \rrbracket_G$
- If $\mathcal{P} = (\mathcal{P}_1 \text{ OPTIONAL } \mathcal{P}_2)$, then: then
 - (a) if $\mathcal{P}_2$ is $(\mathcal{P}_3 \text{ FILTER } C)$ then $\llbracket \mathcal{P} \rrbracket_G = \llbracket \mathcal{P}_1 \rrbracket_G \bowtie_{f(C)} \llbracket \mathcal{P}_3 \rrbracket_G$
 - (b) else $\llbracket \mathcal{P} \rrbracket_G = \llbracket \mathcal{P}_1 \rrbracket_G \bowtie_{(true)} \llbracket \mathcal{P}_3 \rrbracket_G$
- If $\mathcal{P} = (\mathcal{P}_1 \text{ MINUS } \mathcal{P}_2)$, then $\llbracket \mathcal{P} \rrbracket_G = \llbracket \mathcal{P}_1 \rrbracket_G - \llbracket \mathcal{P}_2 \rrbracket_G$
- If $\mathcal{P} = (\mathcal{P}_1 \text{ NOT- EXISTS } \mathcal{P}_2)$, then $\llbracket (\mathcal{P}_1 \text{ NOT- EXISTS } \mathcal{P}_2) \rrbracket_G = \{\mu \mid \mu \in \llbracket \mathcal{P}_1 \rrbracket_G \wedge \llbracket \mu(\mathcal{P}_2) \rrbracket_G = \emptyset\}$
- If $\mathcal{P} = (\mathcal{P}_1 \text{ FILTER } C)$, then $\llbracket \mathcal{P}_1 \text{ FILTER } C \rrbracket_G = \sigma_{f(C)}(\llbracket \mathcal{P}_1 \rrbracket_G)$

The semantics of PPs is shown in [Fig. 1](#). The semantics uses the evaluation function $\llbracket \langle a, elt, b \rangle \rrbracket_G$, which takes as input a PP pattern and a graph and returns a multiset of solution mappings. In [Fig. 1](#) we do not report all the combinations of types of patterns as they can be derived in a similar way. For connectivity patterns the SPARQL standard introduces auxiliary functions called ALP, which stands for Arbitrary Length Paths (see [Fig. 2](#)); in this case the evaluation does not admit duplicates thus solving a problem in an early version of the semantics that was based on counting and lead to intractability ([Arenas et al., 2012](#)).

The solution mappings from the query clause, that is the solution mappings obtained by evaluating the query pattern contained in query clause, can be further chopped and changed through the solution modifiers discussed in Section Solution Modifiers.

Solution Modifiers

Solution modifiers can be used to post-process results produced from the query clause. SPARQL offers the following solution modifiers:

- ORDER BY [ASC | DESC]: it allows to sort results. Sorting is performed lexicographically based on variable order. The optional ASC and DESC keywords allow to specify whether for specific variables the sorting should be ascending or descending.
- LIMIT: it allows to indicate a non-negative integer n that specifies the maximum number of results to return.
- OFFSET: it allows to specify a non-negative integer indicating how many results should be skipped. In combination with LIMIT, this allows for a form of pagination of results. However, since no default order can be assumed in the results, OFFSET is only useful in combination with ORDER BY.

Using SPARQL to Query Ontologies

The SPARQL language is orthogonal to the RDFS and OWL, indeed, it is built directly on top of the RDF data-model and does not provide direct support for inferencing. However, being a query language proposed for RDF data, SPARQL can be used to query

$$\begin{aligned}
 \llbracket \langle \alpha, u, \beta \rangle \rrbracket_G &:= \{ \mu \mid \text{dom}(\mu) = (\{\alpha, \beta\} \cap \mathcal{V}) \text{ and } \mu(\langle \alpha, u, \beta \rangle) \in G \}, \text{card}(\mu, \Omega) = 1 \\
 \llbracket \langle \alpha, !(u_1 \mid \dots \mid u_n), \beta \rangle \rrbracket_G &:= \{ \mu \mid \text{dom}(\mu) = (\{\alpha, \beta\} \cap \mathcal{V}), \exists u \in \mathbf{I} \text{ such that } u \notin \{u_1, \dots, u_n\} \text{ and} \\
 &\quad \mu(\langle \alpha, u, \beta \rangle) \in G \} \text{ and } \text{card}(\mu, \Omega) = |\{u \mid u \in \mathbf{I}, u \notin \{u_1, \dots, u_n\}, \\
 &\quad \mu(\langle \alpha, u, \beta \rangle) \in G \}| \\
 \llbracket \langle \alpha, \hat{elt}, \beta \rangle \rrbracket_G &:= \llbracket \langle \beta, elt, \alpha \rangle \rrbracket_G \\
 \llbracket \langle \alpha, elt_1 / elt_2, \beta \rangle \rrbracket_G &:= \pi_{\{\alpha, \beta\} \cap \mathcal{V}} \left(\llbracket \langle \alpha, elt_1, ?v \rangle \rrbracket_G \bowtie \llbracket \langle ?v, elt_2, \beta \rangle \rrbracket_G \right) \\
 \llbracket \langle \alpha, (elt_1 \mid elt_2), \beta \rangle \rrbracket_G &:= \llbracket \langle \alpha, elt_1, \beta \rangle \rrbracket_G \sqcup \llbracket \langle \alpha, elt_2, \beta \rangle \rrbracket_G \\
 \llbracket \langle x_L, (elt)^*, ?v_R \rangle \rrbracket_G &:= \{ \mu \mid \text{dom}(\mu) = \{?v_R\} \text{ and } \mu(?v_R) \in \text{ALP1}(x_L, elt, G) \}, \text{card}(\mu, \Omega) = 1 \\
 \llbracket \langle ?v_L, (elt)^*, ?v_R \rangle \rrbracket_G &:= \{ \mu \mid \text{dom}(\mu) = \{?v_L, ?v_R\} \text{ and } \mu(?v_L) \in \text{terms}(G) \text{ and} \\
 &\quad \mu(?v_R) \in \text{ALP1}(\mu(?v_L), elt, G) \}, \text{card}(\mu, \Omega) = 1 \\
 \llbracket \langle ?v_L, (elt)^*, x_R \rangle \rrbracket_G &:= \llbracket \langle x_R, (\hat{elt})^*, ?v_L \rangle \rrbracket_G \\
 \llbracket \langle x_L, (elt)^*, x_R \rangle \rrbracket_G &:= \begin{cases} \{\mu_\emptyset\} & \text{if } \exists \mu \in \llbracket \langle x_L, (elt)^*, ?v \rangle \rrbracket_G : \mu(?v) = x_R, \\ \emptyset & \text{else} \end{cases} \text{ and } \text{card}(\mu, \Omega) = 1
 \end{aligned}$$

Fig. 1 Standard semantics of SPARQL Property Paths, where $\alpha, \beta \in (\mathbf{I} \cup \mathbf{L} \cup \mathcal{V})$; $u, u_1, \dots, u_n \in \mathbf{I}$; $x_L, x_R \in (\mathbf{I} \cup \mathbf{L})$; $?v_L, ?v_R \in \mathcal{V}$; $?v \in \mathcal{V}$ is a fresh variable.

Function $\text{ALP1}(\gamma, elt, G)$
Input: $\gamma \in (\mathbf{I} \cup \mathbf{B} \cup \mathbf{L})$,
 elt is a PP expression,
 G is an RDF graph.

- 1: $Visited := \emptyset$
- 2: $\text{ALP2}(\gamma, elt, Visited, G)$
- 3: **return** $Visited$

Function $\text{ALP2}(\gamma, elt, Visited, G)$
Input: $\gamma \in (\mathbf{I} \cup \mathbf{B} \cup \mathbf{L})$, elt is a PP expression,
 $Visited \subseteq (\mathbf{I} \cup \mathbf{B} \cup \mathbf{L})$, G is an RDF graph.

- 4: **if** $\gamma \notin Visited$ **then**
- 5: add γ to $Visited$
- 6: **for all** $\mu \in \llbracket \langle ?x, elt, ?y \rangle \rrbracket_G$ such that $\mu(?x) = \gamma$ and
 $?x, ?y \in \mathcal{V}$ **do**
- 7: $\text{ALP2}(\mu(?y), elt, Visited, G)$

Fig. 2 Auxiliary functions used for defining the semantics of PP expressions of the form elt^* .

RDF ontologies. In this section we report some examples of SPARQL queries used to retrieve information from some ontology. Suppose to have a SPARQL dataset containing one or more ontologies, stored as named graphs.

Example 2: *Since, typically, every ontology contains at least one class, in this example we want to find all graphs that contain at least one class. This can be done in SPARQL by running the following query:*

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?ontology
WHERE
{
  GRAPH ?ontology
    { ?c rdf:type owl:Class }.
}
```

In this example, the keyword GRAPH is followed by a variable (i.e., $?ontology$). Such a variable is used to bind the named graph for which the graph pattern is matched; in this case that contains at list one triple matching the triple pattern $?c \text{ rdf:type owl:Class}$.

Example 3: *Find all the subclasses of a given class. This can be done in SPARQL by running the following query:*

```
SELECT DISTINCT ?sc
WHERE
{
  ?sc rdfs:subClassOf :myClass.
}
```

Example 4: *Find all the subclasses of a given class, by making a recursive search. This means that the transitivity of the subclass relation must be made explicit. This can be done in SPARQL by running the following query that makes use of SPARQL property paths:*

```
SELECT DISTINCT ?sc
WHERE
{
  ?sc (rdfs:subClassOf)* :myClass.
}
```

Example 5: *Find the number of all classes of an ontology. The SPARQL query is the following:*

```
SELECT count(?c)
WHERE
{
  ?c rdfs:type owl:Class.
}
```

```
?c rdf:type owl:Class.  
}
```

This script can be easily modified to find the number of subclasses of a given class:

```
SELECT count(?sc)  
WHERE  
{  
  ?sc (rdfs:subClassOf)* :myClass.  
}
```

Example 6: Retrieve the information associated to the classes of an ontology. Below is the SPARQL query:

```
SELECT DISTINCT ?c ?p, ?o  
WHERE{  
  ?c rdf:type owl:Class.  
  {?c ?p ?o .}  
  UNION  
  {?o ?p ?c .}  
}
```

The UNION operator allows to consider both the triples in which the particular class appears as the subject and those in which it appears as the object.

Example 7: Find all the instances of a given class. It can be done by executing the following SPARQL query:

```
SELECT DISTINCT ?i  
WHERE{  
  ?i rdf:type :myClass.  
}
```

This query can be easily modified to find the instances of a given class or of all its subclasses. The following query uses a property path pattern:

```
SELECT DISTINCT ?i  
WHERE{  
  {?i rdf:type :myClass.}  
  UNION  
  {?i rdf:type ?c .  
   ?c (rdfs:subClassOf)* :myClass.  
  }  
}
```

Ontology Management Tools

There are a lot of software tools proposed for the creation and manipulation of ontologies. In what follows we provide a high level overview of the most popular ontology management tools. We will focus on the following three main aspects:

- Interoperability with ontology definition languages: considering the definition languages reported in the previous article (RDF, RDFS, OWL);
- Inference and consistency checking: discussing whether the tool has the possibility to use an inference engine and can perform constraint/consistency checking.
- Usability: by considering the existence of the graphical editors for the creation and querying of ontologies.

The result of the analysis is reported in [Table 2](#).

Protégé

Protégé ([Knublauch et al., 2004](#)) is a free, open source ontology editor and a knowledge management system developed at Stanford University. Protégé provides a graphical user interface to define ontologies. It also includes deductive classifiers to validate that models are consistent and to infer new information. Protégé currently exists in a variety of frameworks. A desktop system supports many advanced features to enable the construction and management of OWL ontologies. A Web-based system offers distributed access over the Internet using any Web browser.

Table 2 Comparison among various tools and environments for ontology development and engineering

<i>Tool</i>	<i>Input format</i>	<i>Output format</i>	<i>GUI</i>	<i>Consistency Checking</i>
Protégé	RDF(S), OWL	RDF(S), OWL	Yes	Yes
Swoop	RDF(S), OWL	RDF(S), OWL	Yes	Yes
OntoEdit	RDF(S)	RDF(S)	Yes	Yes
WebODE	RDF(S)	RDF(S)	Yes	Yes
KAON	RDF(S), OWL	RDF(S), OWL	No	Yes
DOE	RDF(S), OWL	RDF(S), OWL	No	Yes
NeOn	RDF(S), OWL	RDF(S), OWL	Yes	Yes

Swoop

Swoop ([Kalyanpur et al., 2006](#)) is an open-source, Web-based ontology editor and browser, tailored specifically for OWL ontologies. It supports multiple ontologies (so that entities and relationships across various ontologies can be compared, edited and merged), uses various OWL presentation syntax to render ontologies, and supports OWL reasoners for consistency checking. The layout of the GUI resembles a familiar frame-based website viewed through a Web Browser. It has formerly been maintained at the University of Maryland only, but is now jointly developed together with Clark & Parsia, IBM Watson Research and the University of Manchester

OntoEdit

OntoEdit ([Sure et al., 2002](#)) is an ontology editor integrating various aspects of ontology engineering, developed by Ontoprise, a provider of Semantic Web infrastructure technologies and product of Germany. Motivated by the fact that typically the development of ontologies involves collaborative efforts of multiple persons, it combines methodology-based ontology development with capabilities for collaboration and inferencing. OntoEdit has been developed by providing high extensibility through plug-in structure and an inference engine.

WebODE

WebODE ([Arpírez et al., 2001](#)) is an integrated ontological engineering workbench that helps users with ontology development and management activities such as ontology edition, browsing, learning, merge and alignment. Moreover, it includes middleware services that permit interoperation with other information systems, such as services for ontology library administration and access and ontology upgrading, querying, and metrics. WebODE is developed by the Technical School of Computer Science in Madrid, Spain.

KAON2

KAON2 (Karlsruhe ontology) ([Bozsak et al., 2002](#)) is an ontology infrastructure developed by the University of Karlsruhe, the Research Center for Information Technologies in Karlsruhe and the University of Manchester. Its first version (KAON1) was developed in 2002 and supported an enhanced version of RDF ontologies. In 2005, the second version (KAON2) was released, offering fast reasoning support for OWL ontologies. KAON2 supports answering conjunctive queries expressed using SPARQL syntax (even if not entire SPARQL specification is supported). Reasoning in KAON2 is implemented by algorithms that allow applying well-known deductive database techniques, such as magic sets or join-order optimizations.

DOE

Differential Ontology Editor (DOE) ([Bachimont et al., 2002](#)) is a simple ontology editor which allows the user to build ontologies via a particular methodology divided in three steps: (i) the user is invited to build two taxonomies of concepts and relations respectively, and for each notion he/she builds a definition following the principles that come from the Differential Semantics theory; (ii) the two taxonomies are considered from an extensional semantics point of view; (iii) the ontology can be translated into a knowledge representation language, which allows to use it in an appropriate ontology-based system or to import it into another ontology-building tool. DOE is developed by the National Audiovisual Institute in France.

NeOn

The NeOn Toolkit ([Haase et al., 2008](#)) aims at providing an infrastructure for networked ontology management and for engineering contextualized networked ontologies and semantic applications. It features methods and tools for managing knowledge that is distributed, heterogeneous, contextualized, and developed collaboratively. The NeOn Toolkit is open source and multi-platform and provides an extensive set of plug-ins covering a variety of ontology engineering activities such as development,

matching, reasoning and inference. The NeOn Toolkit was originally developed as part of the NeOn Project and is now supported by the NeOn Foundation.

Concluding Remarks

In this article, we have provided an overview of the SPARQL query language, a W3C standard. We have outlined the main features of the language via some examples and underlined both its pros (e.g., a flexible query language with specific supports for recursion) and cons (e.g., the fact that in its current form does not allow to perform automatic inference). In the last part, we provided an overview of popular ontology management tools.

See also: Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Ontology: Introduction

References

- Angles, R., Gutierrez, C., 2016. The multiset semantics of sparql patterns. In: International Semantic Web Conference, pp. 20–36, Springer.
- Arenas, M., Conca, S., Perez, J., 2012. Counting beyond a yottabyte, or how sparql 1.1 property paths will prevent adoption of the standard. In: Proceedings of the 21st international conference on World Wide Web, pp. 629–638, ACM.
- Arpíez, J.C., Corcho, O., Fernández-López, M., Gómez-Pérez, A., 2001. We-bode: A scalable workbench for ontological engineering. In: Proceedings of the 1st international conference on Knowledge capture, pp. 6–13, ACM.
- Atencia, M., Euzenat, J., Pirro, G., Rousset, M.-C., 2011. Alignment-based trust for resource finding in semantic p2p networks. In: The Semantic Web – ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, pp. 51–66.
- Bachimont, B., Isaac, A., Troncy, R., 2002. Semantic commitment for designing ontologies: A proposal. In: International Conference on Knowledge Engineering and Knowledge Management, pp. 114–121, Springer.
- Berners-Lee, T., Hendler, J., Lassila, O., *et al.*, 2001. The semantic web. *Scientific American* 284 (5), 28–37.
- Bozsak, E., Ehrig, M., Handschuh, S., *et al.*, 2002. Kaon – Towards a large scale semantic web. *E-Commerce and Web Technologies*. 231–248.
- Calvanese, D., De Giacomo, G., Lembo, D., *et al.*, 2011. The mastro system for ontology-based data access. *Semantic Web* 2 (1), 43–53.
- Fernández-López, M., Gómez-Pérez, A., Juriso, N., 1997. Methontology: From ontological art towards ontological engineering. *AAAI Technical Report SS-97-06*.
- Fionda, V., Pirrò, G., Consens, M.P., 2015. Extended property paths: Writing more SPARQL queries in a succinct way. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI).
- Fionda, V., Pirrò, G., Gutierrez, C., 2015. NautiLOD: A formal language for the web of data graph. *ACM Transactions on the Web (TWEB)* 9 (1), 5.
- Franconi, E., Gutierrez, C., Mosca, A., Pirrò, G., Rosati, R., 2013. The logic of extensional RDFS. In: International Semantic Web Conference, pp. 101–116, Springer.
- Gruber, T.R., *et al.*, 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5 (2), 199–220.
- Haase, P., Lewen, H., Studer, R., *et al.*, 2008. The neon ontology engineering toolkit. In: WWW 2008 Developers Track.
- Harris, S., Seaborne, A., Prud'hommeaux, E., 2013. Sparql 1.1 query language. W3C Recommendation 21 (10). Available at: <https://www.w3.org/TR/sparql11-query/#propertypath-syntaxforms>.
- Hartig, O., Pirrò, G., 2015. A context-based semantics for SPARQL property paths over the web. In: European Semantic Web Conference (ESWC), pp. 71–87, Springer.
- Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C., Hendler, J., 2006. Swoop: A web ontology editing browser. *Web Semantics: Science, Services and Agents on the World Wide Web* 4 (2), 144–153.
- Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A., 2004. The protégé owl plugin: An open development environment for semantic web applications. In: International Semantic Web Conference, pp. 229–243, Springer.
- Perez, A.G., Angele, J., Lopez, M.F., *et al.*, 2002. A survey on ontology tools. Deliverable 1.3, EU IST Project IST-2000-29243 OntoWeb.
- Pérez, J., Arenas, M., Gutierrez, C., 2010. nsparql: A navigational language for RDF. *Web Semantics: Science, Services and Agents on the World Wide Web* 8 (4), 255–270.
- Pirrò, G., Ruffolo, M., Talia, D., 2009. Secco: On building semantic links in peer-to-peer networks. *Journal on Data Semantics XII*. pp. 1–36. doi:10.1007/978-3-642-00685-2_1.
- Prud'hommeaux, E., Seaborne, A., 2008. Sparql query language for RDF. W3C Recommendation. Available at: <http://www.w3.org/TR/rdf-sparql-query/>.
- Sure, Y., Erdmann, M., Angele, J., *et al.*, 2002. Ontoedit: Collaborative ontology development for the semantic web. In: International Semantic Web Conference, pp. 221–235, Springer.

Relevant Website

<http://w3c.org>
W3C.

Ontology in Bioinformatics

Pietro H Guzzi, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Ontology is the study of existing entities or *things* there are in the universe (Sowa, 2000). In computer science, an ontology is defined as a specification of conceptualisations about a domain. Conceptualisation is the formalisation of knowledge about a domain, i.e., concepts, the relationships they hold, and the constraints between or among them. A specification is the concrete representation of this conceptualisation through the use of a formal language to represent knowledge, for example, description logic. So, an ontology is a shared understanding of some domain of interest, which is often realized as a set of **classes (concepts)**, **relations, functions, axioms, and instances** (Gruber, 1993a; Uschold *et al.*, 1998; Gruber, 1993b).

Formally, an ontology comprises:

- a set of strings describing lexical entries L for concepts and relations;
- a set of classes C representing the main concepts of the domain;
- a taxonomy of concepts with multiple inheritances (**heterarchy**) H_C ;
- a set of non-taxonomic relations R described by their domain and range restrictions;
- a heterarchy of relations, H_R ;
- a set of axioms A describing additional constraints on the ontology and making implicit facts explicit.

A class represents a set of entities or things within a domain, (e.g., protein and genes are concepts of the proteomics domain). Relations describe the interactions between two or more concepts or a concept's properties. The taxonomies of classes organise them into a tree-like structure.

For example, let us consider three concepts: (i) protein, (ii) molecule, a generalisation of the previous concept, and (iii) binding site, a portion of the three-dimensional structure of a protein deputed to link to another molecule. A simple type of organization can be imposed using the relations is a, has Component linking the concept in such a form: protein is a molecule, (specialization); protein has Component a binding site (association).

Even the relations, like concepts, can be structured into taxonomies, and they are annotated with specialized properties that capture some quantification, for example, a property that holds for all concepts, or for only one concept, or for a limited number of concepts belonging to a class.

Instances are the objects represented by a concept, for example, the P53 human protein is an instance of the concept of protein.

Axioms are constraints applied to the values of classes or instances, for example, a protein can have at least k PhosphorylationSite. The combination of an ontology and the instances of its concept is called a knowledge base.

Ontologies could be categorised by their characteristics, considering, for example, the levels of detail and generality of the proposed conceptualisation. Different works have proposed many guidelines to categorise ontologies, but for this study, we can consider only the level of generalisation in classifying ontologies. From these considerations, we can consider: (i) *top-level ontologies*, also known as foundational ontologies or standard upper ontologies (SUO), as formalized by the IEEE P1600.1 standard (see Relevant Website section), (ii) *generic ontologies*, and (iii) *domain ontologies*. The ontologies that belong to the first class contain specifications of the domain and problem-independent concepts and relations, such as the DOLCE Ontology (Gangemi *et al.*, 2003) or the Suggested Upper Merged 50 Ontology (SUMO) (Niles and Pease, 2001). The second class of ontology contains general knowledge about a certain domain, such as medicine or biology. The last class contains a specialised conceptualisation of a particular domain, for example, proteomics.

Languages for Modeling Ontologies

We will discuss only two languages specifically developed to model and reason on ontologies: the **Darpa Agent Markup Language plus Ontology Inference Layer** (DAML+OIL) (Hendler and McGuinness, 2001; Fensel *et al.*, 2001) and the **Ontology Web Language** (OWL) (Smith *et al.*, 2004) languages. Both ontology languages are based on **Resource Description Framework** (RDF) language. RDF (Decker *et al.*, 2000) is a standard for describing resources, i.e., anything that can be identified. RDF has been used as a data model to describe resources on the web. Its evolution, namely RDF Schema, allows the definition of elementary ontology elements, for example, classes and hierarchy, properties and constraints.

DAML was developed by the RDF Core Working Group to represent on-tological elements not captured by RDF. Successively, DAML was extended with OIL to enable the reasoning on ontologies. DAML+OIL consists of class elements, property elements, and instances, but was limited in its support of restrictions and concepts. Thus, Ontology Web Language (OWL) took the place of DAML+OIL as the semantic web standard.

OWL was developed from the concepts behind DAML+OIL and is the current W3C standard for ontology languages; it has been extended to provide more explicit description logic. OWL permits the writing of ontologies with different levels of

expressivity: OWL Lite, OWL DL, and OWL Full. The OWL syntax employs URIs for naming and implements the description framework for the web provided by RDF. Changes from DAML + OIL to OWL include the removal of DAML + OIL restrictions concerning RDF. OWL also supports the construction of distributed ontologies, which is beneficial in many ways.

Thus, the integration of distributed ontologies becomes an important design implication. Thus, the support of a distributed ontology system in which specialised ontologies can be maintained as separate entities becomes an attractive option. Moreover, in distributed environments, such as the GRID (Foster and Kesselman, 2003), the management of distributed ontologies can be helpful in addressing different problems.

Some Ontologies for Life Sciences

In this section, a representative sample of existing bio-ontologies will be presented. For this work, we will focus on ontologies used helping data analysis or querying. In the remainder of the section, we will discuss: (i) Gene Ontology (Ashburner *et al.*, 2000; Harris *et al.*, 2004), its usage in gene expression data analysis, (ii) TAMBIS Ontology (TaO) (Baker *et al.*, 1998; Paton *et al.*, 1999), the federation of different databases through its classes, and (iii) DAMON (Cannataro and Comito, 2003), a data mining ontology for grid programming.

The TAMBIS Ontology (TaO)

The Transparent Access to Multiple Bioinformatics Information (TAMBIS 95 Sources) system merges different biological databases through a common query interface on the web. The unification of such databases is realised by the use of an ontology called **TAMBIS ontology** (TaO), which describes bioinformatics tasks and resources. This ontology does not contain any instances, but only describes the application domain through its classes and properties. The instances 100 of this ontology are contained in external databases. In fact, TaO describes the resources, such as protein sequences, that are stored in the federated databases. For example, the concept of receptor protein represents the proteins with a receptor function, and the instances of this concept are the proteins stored in databases. In fact, a set of rules governs the joining of new concepts and the formation of relationships between them.

To explain the usage of TaO in such a system, let us consider a typical query that involves semantically different concepts. For instance, let us consider a user who starts a search by inserting the term *gene*. The system retrieves the relations that are connected to *gene*, (e.g., *hasAccessionNumber*, *AccessionNumber*, and *is Homologous*). This user can expand his or her query using the last terms producing this new concept (*Gene homologue of Gene with Accession Number*) that describes genes that are homologous to a gene with a particular accession number. The TAMBIS system takes this query and processes it, producing a set of queries for the external data sources.

Gene Ontology

The Gene Ontology project is maintained by the Gene Ontology (GO) Consortium. The consortium aims to develop a controlled vocabulary of the molecular-biology domain to describe and organise hierarchical concepts. Gene Ontology maintains the vocabularies of three domains: (i) molecular functions (GO:0008639, MF or F), (ii) biological processes (GO:0008150, BP or P), and (iii) cellular components (GO:0005575, CC or C). Moreover, every term has a unique identifier, for example, (GO:0000001), and a name.

The rationale of this classification is that this organizational structure can be applied to all living organisms, considering a generic eukaryotic cell. The first one is defined considering the biochemical level and what a product of a gene does. The second one mentions the biological goal. The biological process domain is not semantically equivalent to a biological pathway, as stated explicitly by the GO consortium. The last one references the place where a gene product plays its role. The vocabularies are often updated, so two types of terms have been introduced to manage the evolution. An **unknown term**, a child of the root of each vocabulary, it is meant to hold molecules that need more investigation to reveal their role in the domain. The term **obsolete** marks terms that have evolved and are no longer used in the domain.

The Gene Ontology vocabularies are structured hierarchically to form different directed acyclic graphs. In such a graph, a node corresponds to a biological term, and a directed edge links two nodes that are hierarchically related. This representation models the hierarchical structure like a tree, but allows multiple inheritances, i.e., each child node may have more than one parent (Ashburner *et al.*, 2000). Two types of parent-child relationships are defined in GO: *is-a*, and *part-of*. The first describes the specialization of a concept parent in its child. The second denotes that the child term is a component of the parent. A child term may have different relationships with its parents.

There exist many applications of GO in computational molecular biology and bioinformatics, such as Muro *et al.* (2006), Lu *et al.* (2004) and Zhang *et al.* (2004). For example, the GOSSIP framework (Bluthgen *et al.*, 2005) tests, statistically, whether the functions, processes, or locations described in the ontology could be significantly enriched by the use of a group of genes. The GOStat (Beissbarth and Speed, 2004) tool utilizes the information of GO to automatically infer what annotations are typical for a particular list of genes. The tool takes in input from a group of genes, finds the corresponding annotations in the ontology, and

generates statistics for each of these, considering, for example, over-representation. Finally, it sorts the GO terms according to their specificity to the considered group.

DAMON: A Data Mining Ontology for Grid Programming

The Data Mining Ontology for Grid programming (DAMON) (Cannataro and Comito, 2003) has been introduced to model data mining tools, concepts, and resources. DAMON aims to characterise the data mining scenario, considering the process of knowledge discovery in a distributed scenario, such as the GRID (Foster and Kesselman, 2003). The main goal of this ontology was to enable the semantic search of data mining resources and to suggest the use of the resources themselves. The conceptualisation has been made by these parameters:

Task: the data mining task performed by the software, for example, the results produced or the knowledge discovered;

Method: the type of methodologies that the software uses in the data mining process;

Algorithm: the algorithm that uses such methodologies;

Software: the software implementing the algorithms;

Data Source: the kind of data sources the software works on;

Human interaction the degree of required interaction with the user, for example, if the mining process is completely autonomous, or if it requires user intervention;

Suite: a collection of many types of data mining software tools.

Starting with these concepts, a set of taxonomies has been induced. Many non-taxonomic relationships and different constraints complete the 175 organization of concepts. For instance, the software and algorithm concepts are connected by **Implements Algorithm** property. This relationship has to verify some imposed constraints, for example, Implements Algorithm property of classification software must have a classification algorithm as filler.

By browsing DAMON, a researcher can choose the optimal data mining 180 techniques to utilize. Given a specified kind of data, a user can browse the ontology taxonomy and find the pre-processing phase, the algorithms, and its software implementation. In summary, the DAMON ontology allows the semantic search of data mining software and other data mining resources and suggests to the user the methods and software used to stored knowledge and the user's 185 requirements.

Appendix.

See also: Natural Language Processing Approaches in Bioinformatics. Ontology: Introduction. The Gene Ontology

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics* 25 (1), 25–29. Available at: <https://doi.org/10.1038/75556>
- Baker, P., Brass, A., Bechhofer, S., *et al.*, 1998. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. An Overview, vol. 25, p. 34. Available at: <http://www.cs.man.ac.uk/~stevensr/papers/baker98.pdf>.
- Beissbarth, T., Speed, T.P., 2004. Gostat: find statistically overrepresented gene ontologies within a group of genes. *Bioinformatics* 20 (9), 1464–1465. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=14962934.
- Blthgen, N., Brand, K., Cajavec, B., *et al.*, 2005. Biological profiling of gene groups utilizing gene ontology. *Genome Inform.* 16 (1), 106–115. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=16362912.
- Cannataro, M., Comito, C., 2003. A data mining ontology for grid programming. In: *Workshop on Semantics in Peer-to-Peer and Grid Computing (in conj. with WWW2003)*. Budapest-Hungary.
- Decker, S., Melnik, S., van Harmelen, F., *et al.*, 2000. The semantic web: The roles of xml and rdf. *Internet Computing, IEEE* 4 (5), 63–73. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=877487.
- Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., 2001. Oil: An ontology infrastructure for the semantic web. *Intelligent Systems, IEEE* (see also *IEEE Intelligent Systems and Their Applications*) 16 (2), 38–45. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=920598.
- Foster, I., Kesselman, C., 2003. The Grid 2: Blueprint for a New Computing Infrastructure (The Morgan Kaufmann Series in Computer Architecture and Design). Morgan Kaufmann. Available at: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/1558609334>
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., 2003. Sweetening wordnet with dolce. *AI Magazine* 24 (3), 13–24. Available at: <http://citeseer.ist.psu.edu/uschold95enterprise.html>.
- Gruber, T.R., 1993a. A translation approach to portable ontologies. *Knowledge Acquisition* 5 (2), 199–220.
- Gruber, T.R., 1993b. Towards principles for the design of ontologies used for knowledge sharing. In: Guarino, N., Poli, R. (Eds.), *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Dordrecht, The Netherlands: Kluwer Academic Publishers. Available at: <http://citeseer.ist.psu.edu/gruber93toward.html>.
- Harris, M.A., Clark, J., Ireland, A., *et al.*, 2004. The gene ontology (go) database and informatics resource. *Nucleic Acids Research* 32 (Database Issue), 258–261. Available at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=14681407.
- Hendler, J., McGuinness, D.L., 2001. The darpa agent markup language. *IEEE Intelligent Systems* 15 (6), 67–73.
- Lu, X., Zhai, C., Gopalakrishnan, V., Buchanan, B.G., 2004. Automatic annotation of protein motif function with gene ontology terms. *BMC Bioinformatics* 5 (1), Available at: <https://doi.org/10.1186/1471-2105-5-122>.
- Muro, E.M., Perez-Iratxeta, C., Andrade-Navarro, M.A., 2006. Amplification of the gene ontology annotation of affymetrix probe sets. *BMC Bioinformatics* 7. Available at: <https://doi.org/10.1186/1471-2105-7-159>.
- Niles, I., Pease, A., 2001. Towards a standard upper ontology. In: Welty, C., Smith, B. (Eds.), *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems FOIS-2001*. URL Ogunquit, Maine, October 17–19.

- Paton, N., Stevens, R., Baker, P., *et al.*, 1999. Query processing in the TAMBIS bioinformatics source integration. In: Ozsoyoglu, Z. e. a. (Ed.), Proceedings of the 11th International Conference on Scientific and Statistical Database Management (SSDBM), pp. 138–147. Los Alamitos, California: IEEE Press.
- Smith, M., Welty, C., McGuinness, D., 2004. Owl web ontology language guide. Available at: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- Sowa, J.F., 2000. Ontology, metadata, and semiotics. In: ICCS '00: Proceedings of the Linguistic on Conceptual Structures, pp. 55–58. London, UK: Springer-Verlag. Available at: <http://portal.acm.org/citation.cfm?id=657527>.
- Uschold, M., King, M., Moralee, S., Zorgios, Y., 1998. The enterprise ontology. The Knowledge Engineering Review, Special Issue on Putting Ontologies to Use 13 (1), 31–89. Available at: <http://citeseer.ist.psu.edu/uschold95enterprise.html>.
- Zhang, W., Morris, Q.D., Chang, R., *et al.*, 2004. The functional landscape of mouse gene expression. Journal of Biology 3 (5), 21. Available at: <https://doi.org/10.1186/jbiol16>.

Relevant Website

<http://suo.ieee.org/>
IEEE Standards Association.

Biographical Sketch

Pietro H. Guzzi is an assistant professor of Computer Science Engineering at the University Magna Grecia of Catanzaro, Italy. His research interests comprise 190 semantic-based and network-based analysis of biological and clinical data.

Biological and Medical Ontologies: Introduction

Marco Masseroli, Politecnico di Milano, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Post-genomic era is producing an incredible amount of biological, bio-molecular and biomedical data, as well as a lot of valuable information extracted from them. This has increasingly stressed the need for more standardized ways to describe such information in order to be able to leverage on them to extract new biomedical knowledge. Many controlled vocabularies (i.e., terminologies) have been developed by different groups to define structural, functional, and phenotypic features of biomolecular and biological entities in a controlled way; more importantly, several of these terminologies are also being actively used world-wide to state and describe such features of specific genes, proteins, cells, tissues and even entire organisms, which are progressively becoming more and better known. This has produced and is keeping generating a growing corpus of precious controlled information; yet, the proliferation of different terminologies, often regarding the same or similar topics, has created the absolute necessity to identify equivalences, similarities, and in general relations between the terms used in different controlled vocabularies, triggering the development of mapping efforts among them. On the other hand, the need to more properly describe the increasing knowledge about the complex biomolecular and biomedical domains in a way that would allow to take advantage of it also computationally, prompted the definition of semantic relations between the terms within these controlled vocabularies, with the development of a number of biomolecular and biomedical ontologies; some of them have become widely used to describe the features of the biomedical-molecular entities in different databases, representing a key element for the interoperability of these repositories.

After this introduction, this article first illustrates what an ontology is and what is its relevance from the knowledge discovery point of view. Then, it focuses on the terminologies and ontologies in the life science domain, highlighting their main issues. Finally, it describes and discusses two important mapping and interoperability efforts among many terminologies and ontologies in the biomedical and biomolecular domains, whose main ones are individually described in the following sections.

Terminologies and Ontologies

A *controlled vocabulary*, or *terminology*, is a collection of the precise and universally comprehensible terms that univocally define and identify different concepts in a specific subject field or domain of activity. Terms are single words, compound words, or multi-word expressions that in a specific context (i.e., domain) have specific meanings; such meanings may be different from those that the same terms have in other contexts. Thus, a terminology regards the concepts in a specific domain and the labels (terms) used in that domain to refer to them. Particularly, such terms are controlled since they are defined and maintained by groups of experts of the specific domain of the terminology.

In a terminology, the concepts are defined as independent and unrelated, however often semantic relations exist between the concepts used in a domain; such relations can be expressed through *semantic networks*. They are logical structures used to represent relationships between concepts in a specific domain through a graph structure; such structure is composed of a set of elements, the graph nodes, representing the domain concepts, and the graph arches, representing the relations between the domain concepts. Each arch in the graph defines a type of relation; the main relation types are *IS A* and *PART OF*. The former one describes a *specification*, and defines a subtype of a concept or class (e.g., a growth factor *IS A* protein). The latter one expresses *membership* and represents a part-whole relationship; it relates two concepts A and B only if B is part of A whenever B is present and its existence implies the existence of A, although the presence of A does not guarantee the presence of B (e.g., a gene is *PART OF* DNA, since all genes are part of some DNA).

IS A and *PART OF* model hierarchical relations, with concepts related at different levels of specification. Both of them held the transitive property, which can be used to make useful inferences. Furthermore, the *IS A* relation allows the inheritance of attributes among related concepts, i.e., all attributes of a more general concept apply also to a more specific one, related to the former one through an *IS A* relation.

Other more specific relations present in the biomedical-molecular domain are: *ASSOCIATED WITH*, *EXPRESSED BY*, *REGULATED BY*, *TREATED BY*, *CAUSED BY*, etc.

Semantic networks are also a reasoning tool, since they allow finding relations between concepts which are not directly related. For example, through a semantic network which expresses that the *DNA* is *PART OF* a *cell*, and a *cell* is *PART OF* an *organism*, it can be inferred that the *DNA* is *PART OF* an *organism*, although in the semantic network it is not explicitly expressed (i.e., there is no a direct relation between *DNA* and *organism*) (Fig. 1). Interestingly from a computational point of view, semantic networks can be implemented in software and automatically processed, i.e., knowledge inference can be performed automatically.

An *ontology* is a semantic structure used to describe the knowledge of a domain in a textual and computable form, as well as standardize and give precise definitions for the terminology employed in the domain. According to the philosopher Barry Smith, it is "any theory or system that aims to describe, standardize or provide rigorous definitions for terminologies used in a domain" (Smith, 2003).

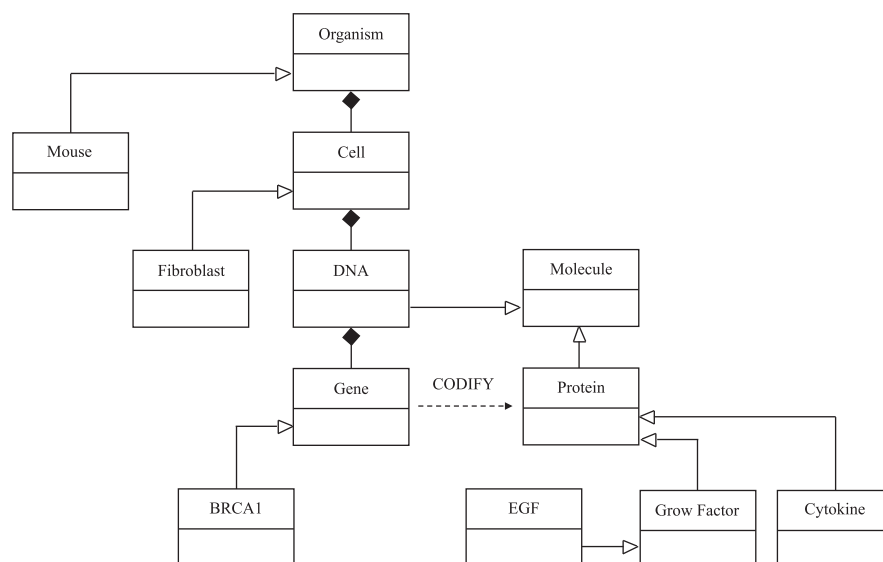


Fig. 1 Example excerpt of the DAG of an ontology in the biomolecular domain, with its semantic network relating concepts defined by the ontology controlled vocabulary.

Ontologies are composed by a controlled vocabulary (i.e., terminology) and a semantic network; leveraging on the latter one, they are very useful for automatic classification, inference and reasoning. They are generally represented through a Directed Acyclic Graph (DAG), where the DAG nodes are the domain concepts expressed by the ontology controlled vocabulary, and the DAG edges are the semantic relations between them, described by the semantic network of the ontology (Fig. 1). Since in a DAG relations are directed, a hierarchy exists among the concepts of the ontology, where one of two related concepts is more general (i.e., the parent) and the other more specific (i.e., the child). In a DAG not only a parent node can have multiple children, but also a child node can have multiple parent nodes; this is the main aspect that differentiates a DAG from a tree. The simplest version of an ontological DAG is a logical tree; in this case, when all relationships of the ontological tree are of IS A type, the ontology is a taxonomy.

Bio-Terminologies and Bio-Ontologies

When a terminology or an ontology regards the biological, biomolecular, or biomedical domain, it is said to be a bio-terminology or bio-ontology. While the aim of a bio-terminology is to collect names of entities (i.e., substances, qualities and processes) used in the life sciences, the purpose of a bio-ontology is to characterize classes of entities, existing in the reality, which are of biological/biomedical significance; thus, a bio-ontology is concerned with the principled definition of biological, biomolecular, or biomedical classes, and the relations among them.

Bio-terminologies and bio-ontologies have an important role in e-science, since they provide formal and explicit declarations of the entities and their relationships in the life sciences. Built by humans, they can be automatically processed by machines; they have the capabilities of relating disparate data and enabling data interoperability (Fig. 2), data summarization, and data mining. They favor information and knowledge management, sharing, and analysis through the integration of sparse and heterogeneous information, the identification and grouping of “similar” bio-entities, as well as the statistical analysis and data mining of their controlled descriptions. In doing so, they can highlight most relevant biological/biomedical features, and help unveiling knowledge from data; furthermore, they support translational research, to quickly bring in the clinical practice new biomolecular knowledge.

Several bio-terminologies and bio-ontologies have been and are being developed and used to define, describe and suitably identify life science information. They are increasing in number, coverage and use in molecular biology and biomedicine; some notable examples, among the many that should be cited, regard biomolecular sequences (Sequence Ontology (Cunningham *et al.*, 2015)), Protein Ontology (Natale *et al.*, 2017), protein families and domains (Pfam (Finn *et al.*, 2014), InterPro (Finn *et al.*, 2017)), biological processes, molecular functions, cellular components (Gene Ontology (Ashburner *et al.*, 2000)), biochemical and metabolic pathways (KEGG (Ogata *et al.*, 1998), Reactome (Fabregat *et al.*, 2016)), genetic diseases (OMIM (Hamosh *et al.*, 2005), Disease Ontology (Schröml *et al.*, 2012)), and phenotypes (Human Phenotype Ontology (Köhler *et al.*, 2016)). They are very useful to enrich lists of genes and gene products with biological and biomedical information, in a way that then such lists and information can be automatically processed and effectively used in computational evaluations, also based on data mining and machine learning approaches.

Bio-Ontology Issues

Bio-ontology development is fragmented: separate communities of biomedical researchers create and maintain diverse ontologies, and different model organism databases use diverse ontologies to annotate their experimental data. When these activities are not

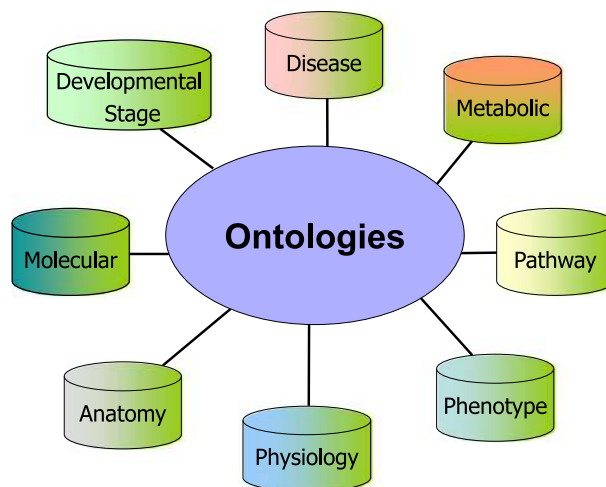


Fig. 2 Ontology driven interoperability of bio-knowledge databases.

unified, they produce not matching ontologies, and consequently not comparable annotations; whereas, unification of efforts could allow integration of each other and with other data, as well as cross-species analyses, also thanks to the algorithms that bioinformaticians are creating to analyze these valuable ontological annotations.

Problems exist at both ontology curation and experimental data annotation level. For ontology content curation, the faced issues rise from many different groups/consortia creating ontologies with uncoordinated efforts, and many different ontologies overlapping in content and with variable quality. This ends in having ontologies that are not interoperable; a strong limitation that can be smoothed only through ontology (and terminology) mapping efforts, which however are laborious and time consuming. Thus, instead of driving interoperability, as they are supposed to do, available ontologies become barriers to accessing, effectively using, and expanding data repositories.

Regarding experimental data annotation (i.e., controlled description of experimental data features), a growing number of data in biomedical resources are annotated with ontologies (e.g., Gene Ontology, Disease Ontology), but current data resources are confined to use single ontology for annotations. Furthermore, issues to be faced concern the difficulty to relate different annotation repositories to each other, with data integration efforts made laborious by mapping difficulties.

With the aim of overcoming these issues, the US National Institute of Health (NIH) funded the National Center for Biomedical Ontology (NCBO), with the mission to advance biomedicine with tools and methodologies for the structured organization of knowledge. Its strategy is to develop, disseminate, and support open-source ontology development and data annotation tools, as well as resources enabling scientists to access, review, and integrate disparate knowledge resources in biomedicine. NCBO supports biomedical researchers in their knowledge-intensive activities, by giving online tools and a Web portal allowing them to access, review, and integrate diverse ontological resources in all aspects of biomedical research and clinical practice. A major focus regards the use of biomedical ontologies to aid in the management and evaluation of data from complex experiments. Among the tools and resources provided by the NCBO, the main ones are:

1. *Open Biological and Biomedical Ontologies* (OBO): An integrated virtual library of biomedical ontologies
2. *Open Biomedical Database* (OBD): An online repository of OBO annotations on experimental data accessible via BioPortal
3. *BioPortal*: A Web-based portal to allow investigators and intelligent computer programs to access the OBO, use them to annotate experimental data in OBD, and visualize and analyze OBO annotations in OBD.

Open Biological and Biomedical Ontologies

The OBO are a set of well-structured, orthogonal, fully interoperable, reference ontologies for shared use across different biological and medical domains; they get these great features by virtue of a common design philosophy and implementation, with sharing principles, including the use of a unique identifier space and enclosure of definitions for all ontology concepts. They were created, and are developed and maintained, by the OBO Foundry, an open, inclusive and collaborative experiment engaging developers of science-based ontologies, which is involved in establishing principles for proper ontology development, and applying them in the OBO (Smith *et al.*, 2007). Mission of the OBO Foundry is developing a family of interoperable ontologies which are both logically well-formed and scientifically accurate to incorporate precise representations of the biological reality. Furthermore, it aims at fostering interoperability of ontologies within the wider OBO framework, and also guaranteeing a gradual improvement of quality and formal rigor in bio-ontologies. The Foundry supports community members who are developing and publishing ontologies in the biomedical domain, so that the created and used ontologies enable scientists and their instruments to communicate with minimum ambiguity.

The OBO Foundry provides a new paradigm for biomedical ontology development by creating gold standard reference ontologies for individual domains of investigation. In order to be part of OBO, an ontology must satisfy several requirements, i.e., it

must be: (i) *open*, i.e., accessible to everyone without any constrain other than its origin must be recognized and its subsequent modifications must be distributed under different names and identifiers; (ii) *expressed in a common and shared syntax and relations* (i.e., using a formal language in an accepted concrete syntax such as the OBO syntax (see in Relevant Websites), its extension, or Web Ontology Language (OWL) (Knublauch *et al.*, 2006)), in order to ease use of the same tools and shared implementation of software applications; (iii) with a *unique identifier space* within OBO; (iv) *clearly specified* and with a *well defined content*, with each OBO ontology that must be orthogonal to the other OBO ontologies; (v) *not overlapping other OBO ontologies* (although partial overlapping can be allowed in order to enable combination of ontology terms to form new terms); (vi) *strictly-scoped in content*, i.e., the ontology scope must be clearly specified and its content must adhere to that scope; (vii) *able to include textual definitions of all terms*, i.e., since several biomedical terms can be ambiguous, the concepts they represent must be precisely defined with their meaning within the specific ontology domain they refer, which must be also specified; (viii) using *relations* which are *unambiguously defined* and follow the pattern of definitions characterized in the OBO Relation Ontology (Guardia *et al.*, 2012); (ix) *well documented*; (x) *collaborative*, i.e., its development must be carried out in a collaborative way; and (xi) with a *plurality of independent users*.

Furthermore, all ontologies that are part of the OBO standard must discriminate:

1. continuants (cells, molecules, organisms, ...)
2. occurrents (events, processes)
3. dependent entities (qualities, functions, ...)
4. independent entities (their bearers)
5. universals (types, kinds)
6. instances (tokens)

This is performed also thanks to the many semantic relations defined between OBO concepts; examples of the different semantic types of such relations included in OBO, and thus that are common to all the ontologies that share the OBO standard, are the *foundational* (is a, part of), *spatial* (located in, contained in, adjacent to), *temporal* (transformation of, derives from, preceded by), and *participation* (has participant, has agent) types.

OBO ontologies tackle several different biological and biomedical aspects, including organism taxonomies, anatomies, cell types, genotypes, sequence attributes, temporal attributes, phenotypes, diseases, and others. The most developed and used OBO include the Gene Ontology (Ashburner *et al.*, 2000), Cell Ontology (Meehan *et al.*, 2011), Foundational Model of Anatomy (Rosse and Mejino, 2003), Disease Ontology (Schriml *et al.*, 2012), Human Phenotype Ontology (Köhler *et al.*, 2016), Phenotypic Quality Ontology (Gkoutos *et al.*, 2005), Sequence Ontology (Cunningham *et al.*, 2015), Protein Ontology (Natale *et al.*, 2017), Relation Ontology (Guardia *et al.*, 2012), Ontology for Biomedical Investigations (Bandrowski *et al.*, 2016), etc.; some of them are detailed described and discussed in the next sections.

Unified Medial Language System

The Unified Medial Language System (UMLS) (Bodenreider, 2004) is another extremely relevant effort to overcome the important issues deriving from the dispersed and unsynchronized works devoted to the development of terminologies and ontologies in the biomedical domain, which consequently result not interoperable. Designed, maintained and disseminated by the National Library of Medicine (NLM), a division of the US National Institute of Health (NIH), since the 1986 the UMLS is an evolving mapping endeavor among the major terminologies and ontologies used in the biomedical and biological domain, with updates twice a year. Its goal is to provide support for the integration of biomedical textual annotations scattered in distinct databases and biomedical resources; its major aim is to overcome two significant barriers towards the effective retrieval of machine-readable information: i) the variety of ways the same concepts are expressed in different machine-readable sources and by different people, and ii) the distribution of useful information among many disparate databases and systems. Other UMLS aims are: facilitating the communication between different biomedical systems, supporting information retrieval and medical record systems, enhancing the access to the biomedical literature by facilitating the development of computer systems that understand biomedical language, developing automatic systems for parsing the biomedical literature, and building enhanced electronic information systems able to create, process, retrieve, integrate, and/or aggregate biomedical and health data and information, both for clinical and informatics research. UMLS strategy towards these goals and aims is providing a set of multi-purpose tools for system developers, i.e., the UMLS is not an end-user application, but enhances the possibility to develop effective end-user applications. UMLS approach is based on leveraging on Knowledge Sources to overcome disparities in language and language format (e.g., *atrial fibrillation*, vs. *auricular fibrillation*, vs. *af*), as well as disparities in granularity (e.g., *contusions*, vs. *hematoma*, vs. *bruise*) and perspective (e.g., when instructing a patient to promptly report *excessive bruising* and *nosebleeds* instead of *epistaxis*). Among UMLS key points there are a broad coverage of medical terms and the absence of multiple inheritance (i.e., each term has only a parent).

UMLS Content and Main Activities

UMLS is a compendium of many controlled vocabularies used in the biomedical domain, which provides a mapping structure among these vocabularies and allows the translation of a term among the various terminology systems mapped (Fig. 3). It may also be viewed as a comprehensive thesaurus and ontology of biomedical concepts.

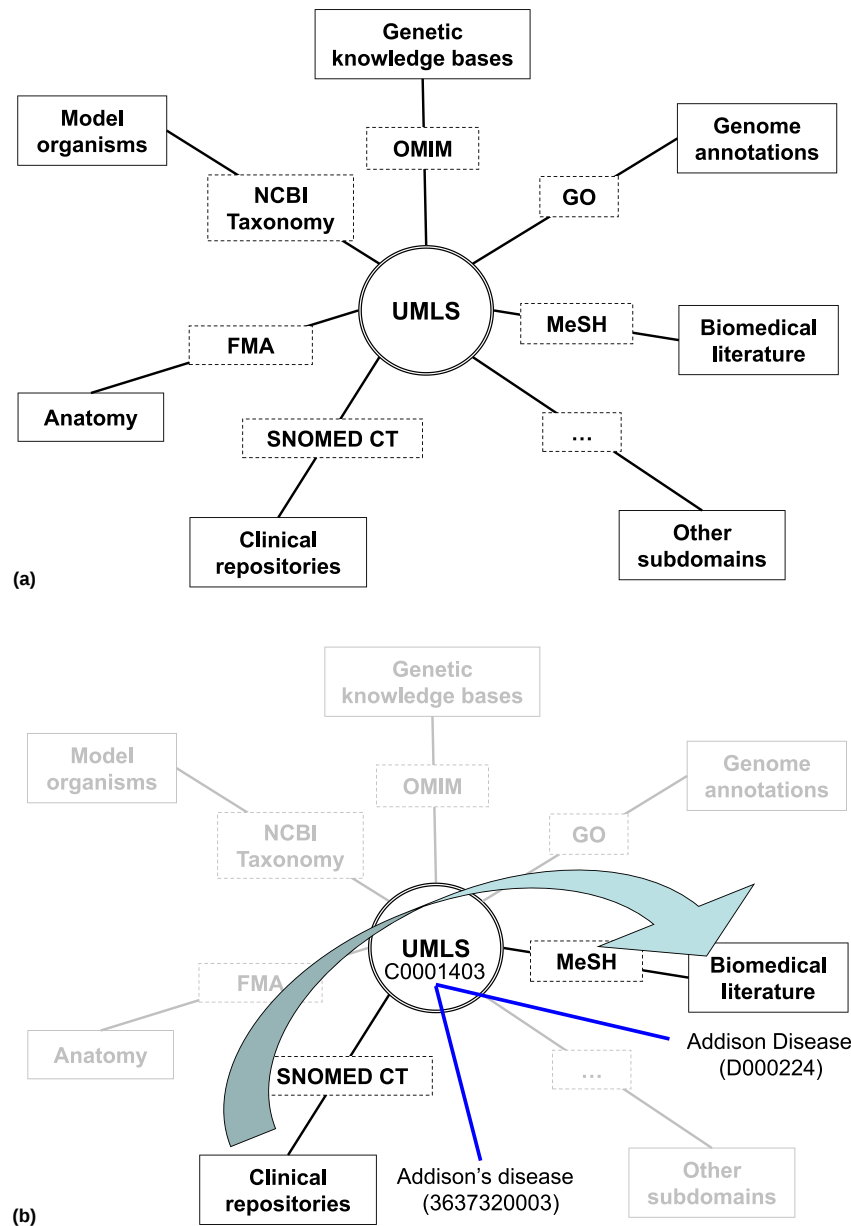


Fig. 3 (a) UMLS as a mapping structure among many biomedical controlled vocabularies. (b) UMLS as a tool for the translation of a term among mapped terminologies.

UMLS organizes terms and concepts of all the mapped controlled vocabularies and ontologies, as well as relates them to other concepts, and categorizes them in broader concepts.

It organizes terms by clustering synonymous terms into a concept, and for each of such concepts it identifies a *preferred term* and a *concept unique identifier* (CUI) (Table 1).

UMLS also organizes concepts described in multiple ontologies with different hierarchies, by using one graph instead of multiple trees (multiple inheritance); in doing so, it keeps inter-concept relations existing in the different hierarchies from the original ontologies, representing redundancy with multiple paths (Fig. 4(a) and (b)).

Furthermore, UMLS relates the organized concepts also to other concepts, through additional hierarchical relationships (i.e., linking to other trees, or making relationships explicit), non-hierarchical relationships, relations to co-occurring concepts, or through relationships' mapping.

Finally, it categorizes concepts by defining high-level categories (*semantic types*) assigned by the UMLS editors independently on the hierarchies where the concepts are located (Fig. 5).

As an example, Fig. 6 shows the representation of the *Addison's disease* concept in UMLS, with its many synonyms, eponyms and clinical variants in different medical vocabularies, and also in different languages, and with its definition, CUI, and semantic type.

Table 1 Example of different synonymous terms in distinct controlled vocabularies for the concept with UMLS preferred term Addison's disease and CUI C0001403

Term	Vocabulary	ID
Addison Disease	MeSH	D000224
Primary hypoadrenalism	MedDRA	10036696
Primary adrenocortical insufficiency	ICD-10	E27.1
Addison's disease (disorder)	SNOMED CT	363732003
Bronzed disease	SNOMED International	DB-70620

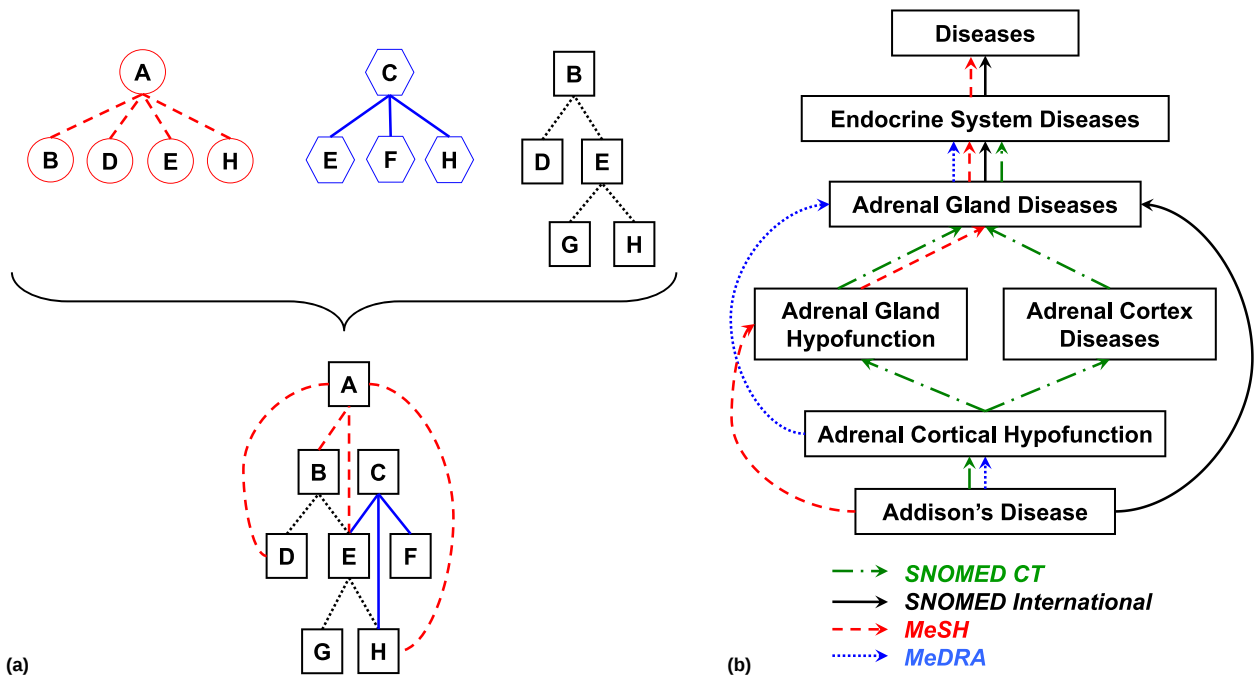


Fig. 4 (a) UMLS single graph approach to describe concepts and their relations in multiple different hierarchies. (b) Example application to the Addison's disease and its relations in the SNOMED CT, SNOMED International, MeSH, and MeDRA hierarchies.

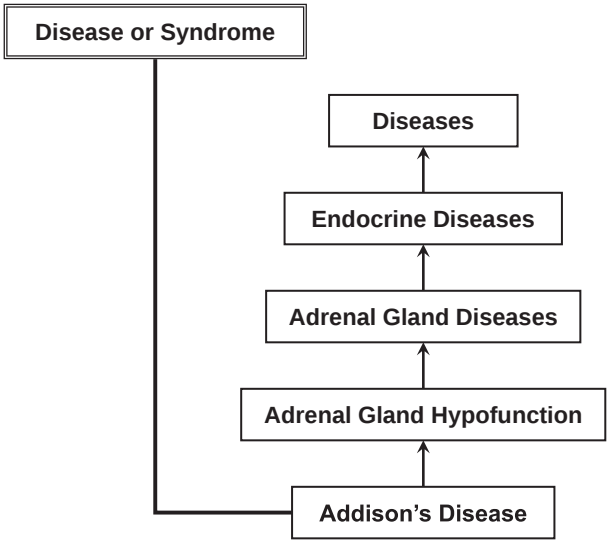


Fig. 5 Example of Disease or Syndrome semantic category applied to the Addison's Disease UMLS concept.

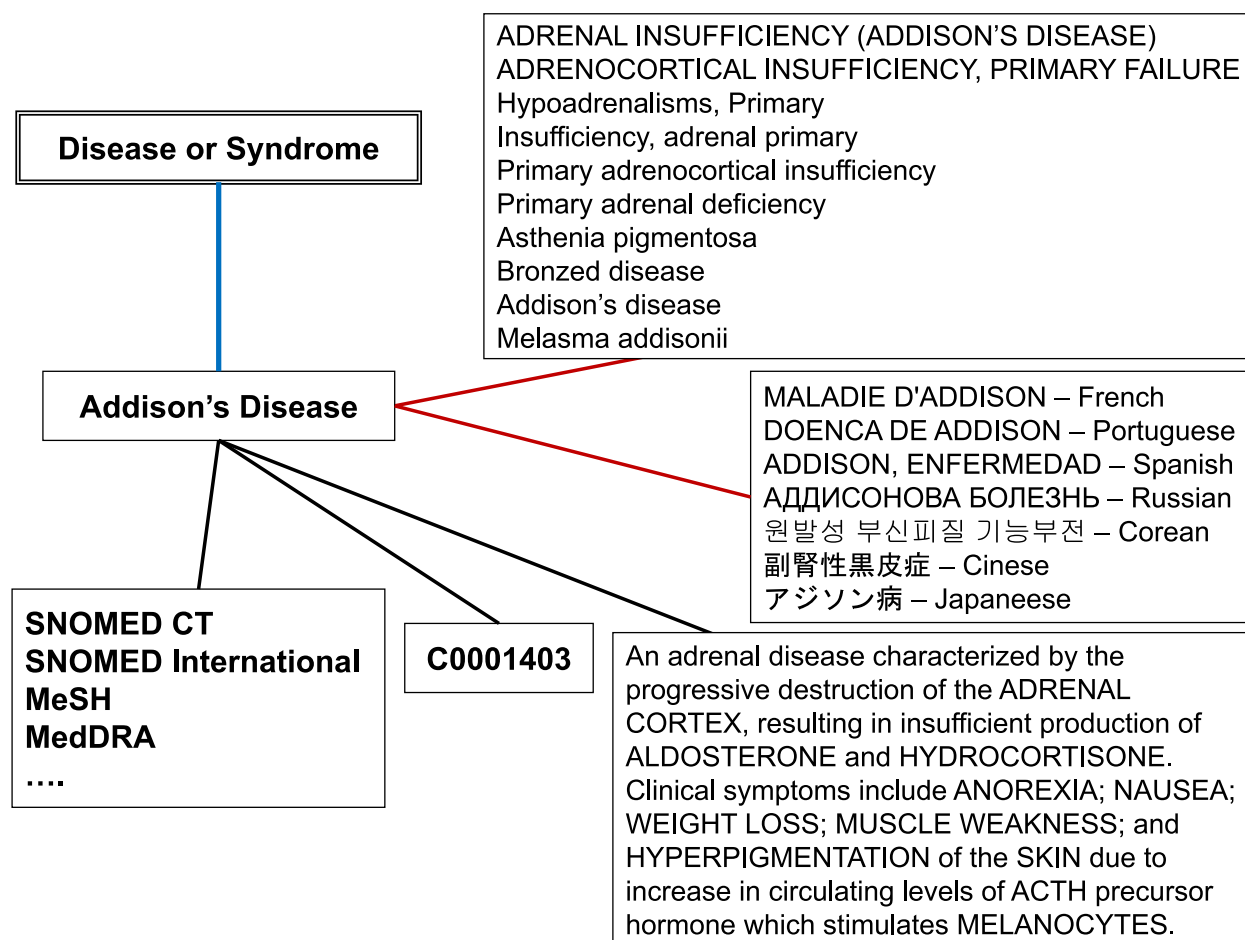


Fig. 6 Example of a concept in UMLS: Addison's Disease.

To perform all these activities (i.e., clustering synonymous terms into concepts, defining a concept unique identifier, refining granularity, broadening scope, adding hierarchical relationships, and providing semantic categorization), UMLS takes advantage of UMLS editors' activity, available lexical knowledge, and semantic pre-processing; the latter one takes into account also the metadata in the source vocabularies, and positive (or negative) evidence for tentative synonymy relations based on lexical features, providing an automatic tentative categorization which is then revised by the UMLS editors.

UMLS Knowledge Sources

The UMLS is composed of three main parts: *Metathesaurus*, *Semantic Network*, and *SPECIALIST Lexicon*, which constitute the UMLS Knowledge Sources, as well as of several *lexical tools*. By design, the UMLS Knowledge Sources are multi-purpose, i.e., they are not optimized for particular applications, but can be applied in systems that perform a range of functions involving one or more types of information (e.g., patient records, scientific literature, guidelines, and public health data).

UMLS Metathesaurus

The UMLS *Metathesaurus* contains information about biomedical and health related concepts, their various names and associated codes, and the relationships among them. It covers a wide range of general and specialized biomedical terminologies, and constitutes the base of UMLS biomedical concepts and their names. It is very large, including at the time of writing this article nearly 3.5 million of biomedical concepts (133 semantic types), more than 13.5 millions of concept names, and 8 millions of relationships (54 semantic relationships); it is multi-source (150 sources), multi-lingual (25 languages), and multi-purpose. The contained source vocabularies and classifications are of many different types, comprising general vocabularies (e.g., RxNorm for drugs), clinical vocabularies, some in multiple languages (e.g., SNOMED CT), administrative code sets (e.g., ICD-9-CM, CPT), and thesauri used to index and retrieve scientific literature and biomedical information in general (e.g., MeSH, CRISP, NCI).

The Metathesaurus is used in a variety of settings and purposes, such as clinical, administrative, public health reporting, and research; its uses span information retrieval, thesaurus construction, natural language processing, automated indexing, and

electronic medical / personal health records. It is organized by concept; each concept has specific attributes defining its meaning, and is linked to the corresponding concept names in the various source vocabularies. Numerous relationships are represented between the concepts, including hierarchical relationships (i.e., *is a*, *is part of*), and associative relationships (e.g., *is caused by*, *in the literature often occurs close to*). It clusters terms from multiple sources by meaning, assigns them to a single concept for which it defines a Concept Unique Identifier, chooses the concept preferred term (as a default, which can be changed), and sets concept synonymous terms (Table 1).

UMLS Semantic Network

Each concept in the Metathesaurus is assigned to one or more semantic types, linked each other through semantic relationships; they broadly and consistently categorize the biomedical domain, and form the UMLS *Semantic Network*, which includes 133 semantic types and 54 semantic relationships.

Semantic types define broad, high level subject categories such as *Clinical Drug*, or *Virus* (e.g., *Disease or Syndrome* is the semantic type of the concept *Addison's Disease*); they are assigned to Metathesaurus concepts independently of concept position in source hierarchies. The major upper level semantic types regard entities (i.e., *Physical object*, or *Conceptual entity*) and events (i.e., *Activity*, *Phenomenon* or *Process*); the main physical object types are *Organism*, *Anatomical structure*, *Manufactured object*, and *Substance*. Major groupings of semantic types comprise *Organism*, *Anatomical structure*, *Biologic function*, *Chemical*, *Physical object*, *Idea or concept*. The information about each semantic type includes a unique identifier, a tree number that represents its position in an *is a* hierarchy, a definition, examples, and its related parent and children types.

Semantic relationships are links between semantic types, such as *is a*, *causes*, or *treats* (e.g., *Virus causes Disease or Syndrome*). They define the structure of the semantic network, and represent important relationships in the biomedical domain between concept categories and between concepts; thus semantic relationships may hold also at the concept level, where also other relationships may apply.

The primary semantic relationship is the *is a*, which establishes a hierarchy of types within the Semantic Network, either among types (e.g., *Animal is a Organism*, *Enzyme is a Biologically active substance*, *Mental process is a Physiological function*), or among relationships (e.g., *prevents is a affects*); furthermore, it eases the allocation of the most specific semantic type available for a Metathesaurus concept.

The Semantic Network also has five major categories of non-hierarchical (or associative) relationships, which group the remaining 53 semantic relationships; these five categories are *physically related to*, *spatially related to*, *temporally related to*, *functionally related to*, and *conceptually related to*. Examples of non-hierarchical relationships are *diagnoses* (e.g., *Sign or Symptom diagnoses Pathologic function*), and *treats* (e.g., *Pharmacologic substance treats Pathologic function*).

The information about each semantic relationship includes a unique identifier, a digit number representing its position in an *is a* hierarchy, a definition, examples, and the set of semantic types that can be linked by the relationship.

UMLS Specialist Lexicon

The *SPECIALIST Lexicon* is a syntactic English lexicon of common words and biomedical terms, which includes at the time of writing this article more than 250 thousand words, and more than 400 thousand variants. It contains information about common English vocabulary, biomedical terms, terms in MEDLINE (PubMed), and terms in the UMLS Metathesaurus. Each of its entries contains *morphological information* (word form and structure, i.e., inflection or derivation), *orthographic information* (word spelling, i.e., spelling variants), and *syntactic information* (how words are put together, i.e., complementation of verbs and nouns) (Table 2).

It is used by the *SPECIALIST Natural Language Processing* system to process text and terms in a customizable way, which is employed to maintain Metathesaurus, indexes.

UMLS Lexical Tools

UMLS *lexical tools* are software programs, in Java programming language, to manage lexical variation, indexing, and normalization in biomedical text. They aim to assist developers in customizing or using the UMLS Knowledge Sources for particular purposes; for example, researchers can use the UMLS products in investigating knowledge representation and retrieval questions. The main lexical tools are Wordind, Lexical Variant Generation, NORM, and MetaMap.

Wordind creates word indexes by breaking a text string into a unique list of lowercased words, whereas *Lexical Variant Generation* (LVG) (Divita et al., 1998) generates normalized terms for word indices, by performing various atomic lexical transformations producing inflectional variants (e.g., the input terms *swim*, *swam*, and *swum* all normalize to the term *swim*).

NORM is a lexical program that generates the normalized strings for the terms in the *SPECIALIST Lexicon*; it includes a selection of LVG transformations, and produces Metathesaurus normalized word and string indexes used to access those terms. The normalization process involves many steps, including stripping possessives, replacing punctuation with spaces, removing stop words (e.g., "No Other Specification" or NOS), lower-casing each word, breaking a string into its constituent words, and sorting the words in alphabetic order.

MetaMap (Aronson, 2001) automatically maps biomedical text to concepts in the UMLS Metathesaurus, i.e., it finds Metathesaurus concepts in text. It does it through a set of subsequent steps which include parsing (using *SPECIALIST* minimal commitment parser, *SPECIALIST Lexicon* and a part of speech tagger), variant generation (using *SPECIALIST Lexicon*, and *Lexical Variant Generation*), candidate concept retrieval (from the Metathesaurus), candidate concept evaluation, and mapping construction. A distributable version of *MetaMap* is available with the name *MetaMap Transfer* (MMTx).

Table 2 Type of information provided for each word in the UMLS SPECIALIST Lexicon

<i>Morphological information</i>	
<i>Word inflection</i>	
Noun:	Nucleus, nuclei
Verb:	Cauterize, cauterizes, cauterized, cauterizing
Adjective:	Red, reddest, redder
<i>Word derivation</i>	
Verb <-> noun:	Cauterize <-> cauterization
Adjective <-> noun:	Red <-> redness
<i>Orthographic information</i>	
<i>Word spelling variants</i>	
oe/e:	Oesophagus–esophagus
ae/e:	Anaemia–anemia
ise/ize:	Cauterise–cauterize
Genitive mark:	Addison's disease Addison disease Addisons disease
<i>Syntactic information</i>	
<i>Word complementation</i>	
<i>Verbs</i>	
Intransitive:	I'll treat
Transitive:	He treated the patient
Ditransitive:	He treated the patient with a drug
<i>Nouns</i>	
Prepositional phrase:	Valve of coronary sinus

UMLS Availability

UMLS can be freely used both locally and remotely. The former access mode is available through the MetamorphoSys software, by installing files locally, and creating customized Metathesaurus subsets that can then be searched, browsed, and viewed. The latter access mode is available via the UMLS Terminology Services (UTS), which provide UMLS Metathesaurus and Semantic Network browsers, as well as application programming interfaces (APIs), downloadable files and programs, guides and documentation to UMLS resources. UTS purpose is to make the UMLS resources more accessible to both users and systems developers.

The use of the UMLS resources, both locally and remotely, is subject to the sign of an online Web-based license, which is free, but with some limitations, particularly in its appendices.

See also: Data Mining: Clustering. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Introduction. The Gene Ontology

References

- Aronson, A.R., 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. In: Proceedings AMIA Symposium, pp. 17–21.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* 25 (1), 25–29.
- Bandrowski, A., Brinkman, R., Brochhausen, M., *et al.*, 2016. The ontology for biomedical investigations. *PLOS One* 11 (4), e0154556.
- Bodenreider, O., 2004. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research* 32, D267–D270.
- Cunningham, F., Moore, B., Ruiz-Schultz, N., *et al.*, 2015. Improving the Sequence Ontology terminology for genomic variant annotation. *Journal of Biomed Semantics* 6, 32.
- Divita, G., Browne, A.C., Rindfleisch, T.C., 1998. Evaluating lexical variant generation to improve information retrieval. In: Proceedings AMIA Symposium, pp. 775–779.
- Fabregat, A., Sidiropoulos, K., Garapati, P., *et al.*, 2016. The reactome pathway knowledgebase. *Nucleic Acids Research* 44 (D1), D481–D487.
- Finn, R.D., Attwood, T.K., Babbitt, P.C., *et al.*, 2017. InterPro in 2017–beyond protein family and domain annotations. *Nucleic Acids Research* 45 (D1), D190–D199.
- Finn, R.D., Bateman, A., Clements, J., *et al.*, 2014. Pfam: The protein families database. *Nucleic Acids Research* 42 (Database issue), D222–D230.
- Gkoutos, G.V., Green, E.C.J., Mallon, A.-M., *et al.*, 2005. Using ontologies to describe mouse phenotypes. *Genome Biology* 6, R8.
- Guardia, G.D., Vêncio, R.Z., de Farias, C.R., 2012. UML profile for the OBO relation ontology. *BMC Genomics* 13 (Suppl 5), S3.
- Hamosh, A., Scott, A.F., Amberger, J.S., *et al.*, 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research* 33, D514–D517.
- Knublauch, H., Oberle, D., Tetlow, P., Wallace, E., 2006. A Semantic Web primer for object-oriented software developers. W3C. 2006-03-09. Available at: <http://www.w3.org/2001/sw/BestPractices/SE/ODSD/>.
- Köhler, S., Vasilevsky, N.A., Engelstad, M., *et al.*, 2016. The human phenotype ontology in 2017. *Nucleic Acids Research* 45, gkw1039.
- Meehan, T.F., Masci, A.M., Abdulla, A., *et al.*, 2011. Logical development of the cell ontology. *BMC Bioinformatics* 12, 6.
- Natale, D.A., Arighi, C.N., Blake, J.A., *et al.*, 2017. Protein Ontology (PRO): Enhancing and scaling up the representation of protein entities. *Nucleic Acids Research* 45 (D1), D339–D346.
- OGata, H., Goto, S., Fujibuchi, W., Kanehisa, M., 1998. Computation with the KEGG pathway database. *Biosystems* 47 (1–2), 119–128.

- Rosse, C., Mejino Jr., J.L., 2003. A reference ontology for biomedical informatics: The foundational model of anatomy. *Journal of Biomedical Informatics* 36 (6), 478–500.
- Schriml, L.M., Arze, C., Nadendla, S., *et al.*, 2012. Disease ontology: A backbone for disease semantic integration. *Nucleic Acids Research* 40, 940–946.
- Smith, B., 2003. Ontology. In: Floridi, L. (Ed.), *Blackwell Guide to the Philosophy of Computing and Information*. Oxford, UK: Blackwell, pp. 155–166.
- Smith, B., Ashburner, M., Rosse, C., *et al.*, 2007. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25 (11), 1251–1255.

Relevant Websites

- <http://bioportal.bioontology.org/>
'BioPortal'.
- <https://metamap.nlm.nih.gov/>
'MetaMap'.
- <http://mmtx.nlm.nih.gov/>
'MetaMap Transfer (MMTx)'.
- <http://www.cs.man.ac.uk/~horrocks/obo/>
'OBO flat file format syntax and semantics and mapping to OWL Web Ontology Language'.
- <http://www.obofoundry.org/>
'Open Biological and Biomedical Ontologies'.
- <https://www.nlm.nih.gov/research/umls/>
'Unified Medical Language System'.
- <http://www.bioontology.org/>
'US National Institute of Health, National Center for Biomedical Ontology'.
- <https://uts.nlm.nih.gov/home.html>
'UTS: UMLS Terminology Services'.

Biographical Sketch



Marco Masseroli received the Laurea degree in Electronic Engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in Biomedical Engineering in 1996, from the Universidad de Granada, Spain. He is associate professor in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano, and lecturer of Bioinformatics and Biomedical Informatics. He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomía Patológica, Facultad de Medicina at the Universidad de Granada, Spain, and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health, Bethesda. His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Biological and Medical Ontologies: GO and GOA

Marco Masseroli, Politecnico di Milano, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Many ontologies exist to describe concepts and their relationships in biology; the Gene Ontology (GO) (Ashburner *et al.*, 2000) is the most developed and used one. This article focuses on this paramount ontology, first describing its development, composition, structure, included concepts and relations, as well as their amounts, available formats and the tools available to browse them. Then, it shows the annotations that can be provided through the GO, and the ways to define their supporting evidence and rules to express them. Furthermore, it illustrates the relevant use of such annotations in computational applications, and the available repositories containing them. Finally, it describes and discusses issues still present in the GO and their possible solutions.

The Gene Ontology

Out of the many ontologies available to describe concepts and their relationships in biology, the Gene Ontology (GO) (Ashburner *et al.*, 2000) is the most developed and used one. The GO project was born in November 1998 as a collaborative effort among the databases of three model organisms: FlyBase (*Drosophila*), *Saccharomyces* Genome Database (SGD) and the Mouse Genome Database (MGD). It addresses the necessity to have consistent descriptions of gene products across databases, in order to ease the search for all available information regarding the same biological entity, but stored in different databases, and to be able to compare them and to support comparative genomic and proteomic studies. This important task, common to almost all biological studies, is difficult due to the dispersion of the information in many distributed and heterogeneous resources, often using different data organizations. Additionally, it is hampered by the use of multiple different terms in distinct databases to describe the same concept or entity, and also the same term to address different concepts, particularly in different species; this prevents obtaining effective results through automatic searches, and even by not highly expert human beings. As an example, the different terms “protein synthesis” or “translation” are equally used to refer to the same biological process of protein production; conversely, the same term “bud initiation” is equally used to refer to the initial area of outgrowth from a cell in the formation of a new cell, or the primordial structure from which a tooth is formed (in animals), or the embryonic shoot of a new branch (in plants).

Since its constitution, the GO Consortium (GOC) has incorporated an increasing number of databases, including many of the most relevant repositories for plant, animal, and microbial genomes. On the basis of the variety of biological entities of different organisms contained in such databases, it has developed three associated sub-ontologies (i.e., structured controlled vocabularies and the relationships between the concepts described by their terms) that express, in a species-independent manner, the *biological processes*, *cellular components*, and *molecular functions* in which the biological entities of different organisms are involved. Specifically, the cellular component sub-ontology describes the parts and the extracellular environment of cells, the molecular function sub-ontology defines the gene product basic activities at molecular level (e.g., binding or catalysis), and the biological process sub-ontology characterizes the operations or the sets of molecular events with a defined start and stop which regard the functioning of integrated living units (e.g., cells, tissues, organs, and organisms). Thus, these three ontologies, which are encompassed within the GO and constitute its sub-ontologies, are very effective in expressing the functional characteristics of genes and proteins across organisms, eukaryotic and single or multi-cellular prokaryotic organisms, even as knowledge keeps accumulating and evolving (Ashburner *et al.*, 2000).

GO Structure

Within each GO sub-ontology, the concepts that the GO terms define and the relationships between them form a logical structure, named Directed Acyclic Graph (DAG) (Fig. 1), which is a graph made of nodes and edges.

Within the GO DAG, each node represents a single concept (i.e., a GO category) expressed by a unique *term* of the controlled vocabulary, whose *name* may be a word or string of words. A term may have one or more *synonyms*, which represent other word or phrase forms to express exactly the same or a close concept, with indication of the relationship between the term and the synonym given by the synonym scope (i.e., EXACT, BROAD, NARROW, or RELATED); furthermore, a term may have one or more *references* to equivalent concepts in other databases (*xref*), and a *comment* about the term meaning or use. Each GO DAG node is identified by a unique alphanumeric *identifier* (ID), formed by the “GO:” characters followed by seven digits (e.g., GO:0033926); furthermore, each DAG node has a unique *definition* with cited sources (e.g., [EC:3.2.1.108], to the ExPASy Enzyme database), and a *namespace* specifying the sub-ontology to which it belongs. An example of the elements of a GO DAG node is the following:

identifier: GO:0033926.

name: glycopeptide alpha-N-acetylgalactosaminidase activity.

namespace: molecular_function.

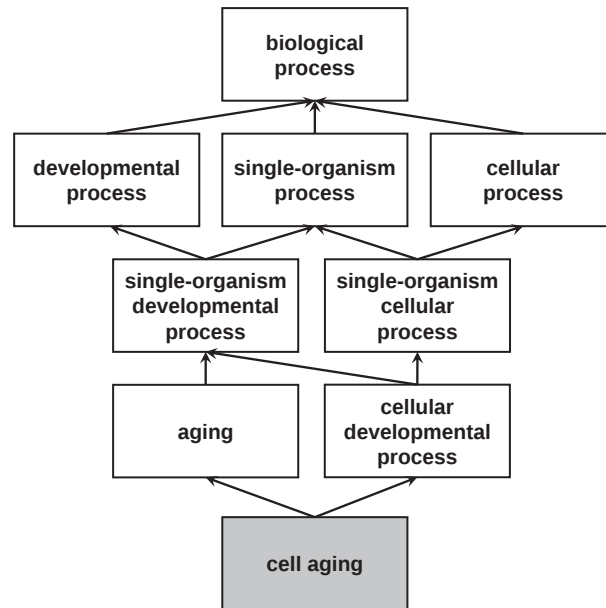


Fig. 1 Example of Gene Ontology Directed Acyclic Graph, for the GO term 'cell aging' (GO:0007569).

definition: "Catalysis of the reaction: D-galactosyl-3-(N-acetyl-alpha-D-galactosaminyl)-L-serine + H₂O=D-galactosyl-3-N-acetyl-alpha-D-galactosamine + L-serine." [EC:3.2.1.97].

synonym: "D-galactosyl-N-acetyl-alpha-D-galactosamine D-galactosyl-N-acetyl-galactosaminohydrolase activity" EXACT [EC:3.2.1.97].

synonym: "endo-alpha-acetylgalactosaminidase activity" BROAD [EC:3.2.1.97].

synonym: "endo-alpha-N-acetylgalactosaminidase activity" BROAD [EC:3.2.1.97].

xref: EC:3.2.1.97.

xref: MetaCyc: 3.2.1.97-RXN.

xref: InterPro: IPR024746.

xref: InterPro: IPR025706.

Each edge of the GO DAG connects two nodes of the same GO sub-ontology, or sometimes of different GO sub-ontologies, and represents a relation between the linked nodes/concepts, i.e., between the vocabulary terms that express them.

As its name indicates, there are no cycles between the nodes of a DAG, and its edges have a one-way meaning. This allows classifying the DAG nodes in top-level, mid-level, and bottom-level nodes, where the level denotes the grade of specialization of each node, being upper-level nodes the most generic and lower-level ones the most specialized. Thus, a DAG is very similar to a hierarchical tree structure; yet, differently from a hierarchy, in a DAG a "child" node, i.e., a more specialized concept, can have multiple "parents" nodes, i.e., more generic concepts, related to it through different types of relations. This defines different paths from a DAG node to the DAG root-node (that is unique), which can be formed by a different number of mid-level nodes (Fig. 2); thus, the notion of level of a DAG node is not unique, since a DAG node can be at different levels simultaneously.

GO relations

Any number of relations can exist between two nodes of the GO DAG, and each of such nodes can have any number and type of relations to other nodes in the GO graph. Currently, the relations described in the Gene Ontology are mainly hierarchical of two types: *specification* (i.e., the *IS A* relation) and *membership* (i.e., the *PART OF* relation).

The *IS A* relation constitutes the main structure of the GO DAG. It defines a subtype, not an instance, i.e., not a specific example of something; in fact, GO terms represent classes of entities or phenomena, rather than specific manifestations thereof, which are defined through the annotation of such manifestations with the GO terms. Examples are: *mitotic cell cycle IS A cell cycle*, or *lyase activity IS A catalytic activity*.

The *PART OF* relation represents a part-whole relationship. In the GO it relates two concepts A and B only if B is necessarily part of A whenever B exists, and its presence implies the presence of A; yet, the existence of A does not mean for certain the presence of B. An example is: *replication fork PART OF chromosome*, since all replication forks are part of some chromosome (but only some chromosomes have a replication fork as their part).

Other kinds of relations that are present, although limitedly, in the GO are *HAS PART*, *REGULATES* and *OCCURS IN*.

HAS PART is the logical counterpart of the *PART OF* relation; it represents a part-whole relationship from the view point of the whole. As *PART OF*, it is used to relate two concepts A and B only when A always necessarily has B as a part; yet, if B exists, it does not mean for certain that A exists. For example: *nucleus HAS PART chromosome*, since all nuclei contain a chromosome, although only some chromosomes are part of a nucleus.

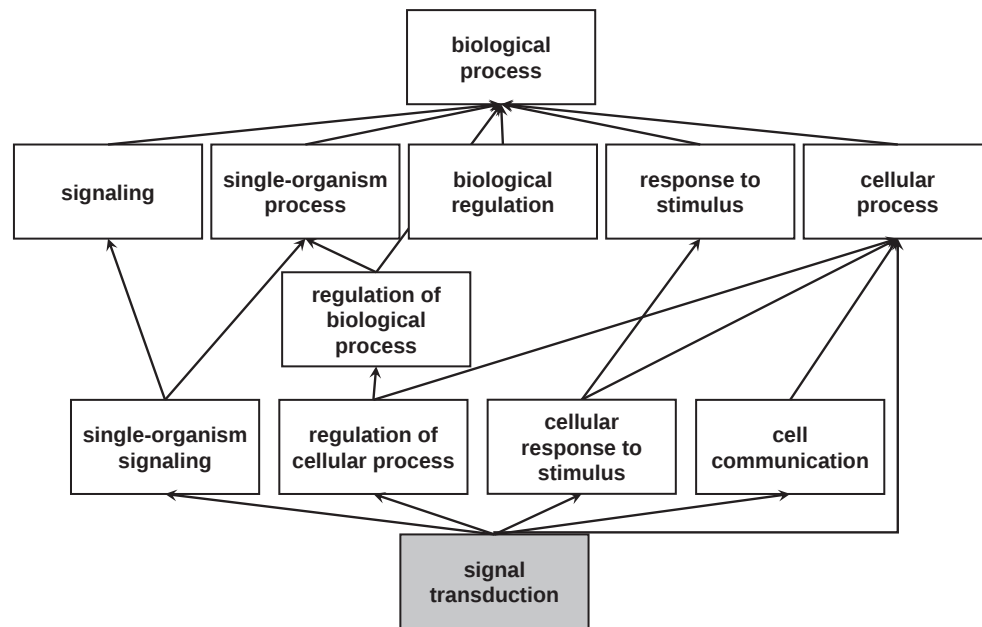


Fig. 2 Example of the DAG of a GO term ('signal transduction', GO:0007165) with multiple paths to the GO root, including a different number of nodes.

The *REGULATES* relation states that a process directly affects the manifestation of another process or quality, i.e., when both are present, the former one necessarily regulates the latter one (which may not regulate the former one). For example, *cell cycle checkpoint REGULATES cell cycle* describes that, when a cell cycle checkpoint occurs, it always regulates the cell cycle, but the cell cycle is regulated also by other processes besides cell cycle checkpoints.

The *REGULATES* relation has two sub-relations, *POSITIVELY REGULATES* and *NEGATIVELY REGULATES*, to denote these more specific types of regulation; if A *POSITIVELY* or *NEGATIVELY REGULATES* B, it also *REGULATES* B.

The *OCCURS IN* relation specifies where a biological process or a molecular function takes place. Differently from the other relations, which connect two concepts of the same sub-ontology, it is used for inter-ontology links, mainly to specify relations between a molecular function or a biological process and a cellular component. Such relation can exist only if the occurrent (process or function) always necessarily takes place in the specified location (component), which must exist when the occurrent exist; yet, the existence of the component does not for certain implies that the specific process or function occurs, and in case it may occur only in some instances of that component.

All mentioned relations constitute the *semantic network* of the GO vocabulary; through them, either very general or very precise concepts can be represented.

GO browsers

The Gene Ontology DAG structure describes the intricate underlying biology, which determines the DAG complexity. Thus, the Gene Ontology structure is pretty more complex than a tree. To help inspecting the complex GO DAG structure, several GO browsers have been constructed; the most relevant ones are AmiGO (Carbon *et al.*, 2009) by the Gene Ontology Consortium, QuickGO (Binns *et al.*, 2009) by the European Molecular Biology Library – European Bioinformatics Institute (EMBL-EBI), and GO Browser (Eppig *et al.*, 2017) by the Mouse Genome Informatics (MGI). All of them provide tools for text searching within the GO vocabulary and for displaying graphically the DAG structure of user specified GO terms.

GO Statistics

The Gene Ontology coverage is very ample, with some parts more detailed than others, reflecting the amount of research activities and outcomes reached and increasingly performed in the different biological and biomolecular areas. Since the GO covers the biology of many organisms, the increasing number of studied organisms and the rising knowledge of their biology make continuously growing the already large number of GO terms, i.e., covered concepts, and relations between the expressed concepts.

On June 9th, 2016, the GO included 43,787 terms overall, subdivided in 28,648 biological processes (64.51%), 10,161 molecular functions (22.88%), and 3907 cellular components (8.80%); additionally, there were 1692 obsolete terms (3.81%) not included in the above statistics. In fact, while the GO evolves, previously defined terms which have been refined in their meaning or discarded are kept as obsolete within the ontology, so that to ease compatibility between different GO versions and applications using them. Among the three GO sub-ontologies, the biological process one is the largest and most developed, with terms as much

specific as the *positive regulation of sevenless signaling pathway* (GO:0045874), which is on the seventeenth level of the GO biological process DAG.

Given the increasing size of the GO, more simple subsets of it, named *GOslim* subsets, have been defined to ease its use and interpretability; they provide a broad overview of the ontology content without the details of specific fine grained terms.

Besides a generic GO slim developed by the GO Consortium which is not species specific as the entire GO, available GO slims are created by users based on their needs, and may be specific to species (e.g., plant, yeast, viruses, or a particular model organism) or to particular areas of the GO.

GO Formats and Availability

Being an ontology, the GO is represented with ontology languages and available in file formats able to encode the knowledge about the specific domain, i.e. to describe the nodes and relations, with their multiple features, of the GO DAG. Two file formats are used to this aim: the OBO and the RDF-XML format.

The OBO format uses an ontology representation language and is the text file format used by OBO-Edit, an open source, platform-independent application for viewing and editing ontologies. It attempts to reach the goals of human readability, ease of parsing, extensibility, and minimal redundancy. The OBO 1.2 version is the flat file format used and recommended by the GO Consortium; the concepts that it models represent a subset of the concepts in the Web Ontology Language (OWL) (Knublauch *et al.*, 2006), a family of knowledge representation languages for authoring ontologies, characterized by formal semantics and built upon a W3C XML standard for objects called the Resource Description Framework (RDF) (OWL2, 2009). For the RDF-XML format of the GO, the document type definition (DTD) is also available, which defines the XML document structure and legal building blocks with a list of permissible elements and attributes, allowing consistency checking of the provided GO files.

The latest version of the whole Gene Ontology, as well as its *slim* subsets and mappings to other controlled vocabularies (e.g., InterPro, KEGG, MetaCyc, Pfam, PRINT, ProSite, Reactome, SMART), are publicly available for downloading in the mentioned formats at the GO website and from the GO database archive (see 'Gene Ontology Consortium. Gene Ontology' and 'Gene Ontology database archive' in Relevant Websites Section).

Gene Ontology Annotations

The most relevant use of the GO is in the controlled and computable description of the features of genes and gene products regarding the biological processes, molecular functions and cellular components in which they occur. Such description, known as *annotation*, involves the association of a gene or gene product with a GO concept (term), where usually each gene or gene product is associated with more GO terms, since it has more features, and each GO term (category) is assigned to many genes or gene products, since many genes (gene products) have same features.

Annotations are usually associated with a specific reference, which describes the work or analysis upon which the annotation is based; they can be given in different ways: assigned by experts, i.e., *human curated*, or automatically predicted, i.e., *computationally assigned*, with or without human supervision (curation) (Hennig *et al.*, 2003). Several evidence codes exist and are associated with each annotation to indicate how the annotation to a particular GO term is supported; they are subdivided in two major groups: *automatically* or *manually assigned* evidence codes. The former one includes the IEA (*Inferred from Electronic Annotation*) evidence code, which states that the annotation is based only on automatic computation. The manually assigned (curated) evidence codes are categorized in: *Experimental* (i.e., results exist from a physical characterization that has supported the annotation), *Computational analysis* (i.e., the annotation is based on an *in silico* analysis of the annotated gene or gene product sequence, and/or other data, also according to a varying degree of curatorial input), *Author statement* (i.e., the annotation is funded on a statement made by the author(s) of the associated reference cited), and *Curatorial statement* (i.e., the annotation is based on a curatorial judgment that does not fit in any of the other evidence code classifications). All the elements of each of these categories are as follows:

1. *Experimental*:

EXP: Inferred from Experiment; better to use one of the following more specific experimental codes:

IDA: Inferred from Direct Assay

IPI: Inferred from Physical Interaction

IMP: Inferred from Mutant Phenotype

IGI: Inferred from Genetic Interaction

IEP: Inferred from Expression Pattern

2. *Computational analysis*:

ISS: Inferred from Sequence or structural Similarity

ISO: Inferred from Sequence Orthology

ISA: Inferred from Sequence Alignment

ISM: Inferred from Sequence Model

IGC: Inferred from Genomic Context

IBA: Inferred from Biological aspect of Ancestor

IBD: Inferred from Biological aspect of Descendant
 IKR: Inferred from Key Residues
 IRD: Inferred from Rapid Divergence
 RCA: Inferred from Reviewed Computational Analysis

3. *Author statement:*

TAS: Traceable Author Statement
 NAS: Non-traceable Author Statement

4. *Curatorial statement:*

IC: Inferred by Curator
 ND: No biological Data available; it is usually associated with annotations to GO sub-ontologies route terms, when no data exist to annotate the gene or gene product to that sub-ontology.

A detailed flow chart is provided by the Gene Ontology Consortium to illustrate the steps that lead the curators to decide what evidence code should be assigned to an annotation.

Although related to different types of evidence, evidence codes should not be considered as statements about the quality of the annotation. Each evidence code encompasses multiple different methods that produce annotations, some of higher confidence or greater specificity than other methods, whose application (or result interpretation) can also affect the quality of the annotation. Despite of this, automatically assigned annotations are usually considered to be the ones with the lowest quality, although recently some specific automatic annotation procedures have been demonstrated to be equally reliable than curated ones, which often can be subjective and not highly robust.

When a reference for a GO annotation describes multiple methods, each of which providing evidence for the annotation, then multiple equal annotations with identical GO identifiers and reference identifiers, but different evidence codes, may be made; thus, different evidence codes, although apparently not coherent (e.g., IEA and EXP) can be found associated with the same annotation.

Besides reference and evidence, also an evidence modifier attribute, named *qualifier*, is assigned to GO annotations; it can assume values such as CONTRIBUTES TO, COLOCALIZES WITH, or null, but also NOT or NOT CONTRIBUTES TO, which actually negate the existence of the annotation. Thus, to correctly evaluate a GO annotation, be always careful to consider also the value of its qualifier attribute.

In summary, from a computational point of view, each GO annotation is represented by a record including:

1. Gene (or gene product) ID
2. Gene Ontology ID
3. Reference ID(s) (e.g., PubMed ID(s))
4. Evidence code(s)
5. Evidence modifier (qualifier)

Two examples of GO annotation records are shown in [Fig. 3](#).

True-Path-Rule

Regardless the way in which the annotation is assigned (automatically or manually curated), the assignment must be done to the most specific concept in the ontology that describes the gene or gene product feature that the annotation states; in fact, each annotation to a GO term must satisfy what is known as the *true path rule*: "if a child term of the ontology describes a gene or gene product, then all its parent terms in the ontology must also apply to that gene or gene product" ([Rhee et al., 2008](#)). Thus, thanks to the ontological structure of the GO, this allows automatically inferring the annotation of a gene or gene product also to all the terms that in the ontology are parents of the terms to which the gene or gene product has been annotated to. For example, the biological process *hexose biosynthetic process* has two parents: *monosaccharide biosynthetic process* and *hexose metabolic process*, since *hexose* is a kind of monosaccharide and *biosynthetic process* is a type of metabolic process ([Fig. 4](#)). When a gene or gene product is annotated the *hexose biosynthetic process*, it is automatically annotated also to all its parents terms, and in particular to both *hexose metabolic process* and *monosaccharide biosynthetic process*.

Such automatic unfolding of the ontological annotations allows to store in biological databases only the direct, most specific annotations, thus saving space without losing information power. Furthermore, leveraging on the ontology structure, it allows applying automatic reasoning procedures to predict new unknown gene or gene product features.

Protein ID	GO ID	Evidence code	PubMed ID	Qualifier
P05147	GO:0047519	IDA	PMID:2976880	null

Protein ID	GO ID	Evidence code	PubMed ID	Qualifier
P98194	GO:0005388	IDA	PMID:16192278	NOT

Fig. 3 Example of GO annotation records of two proteins.

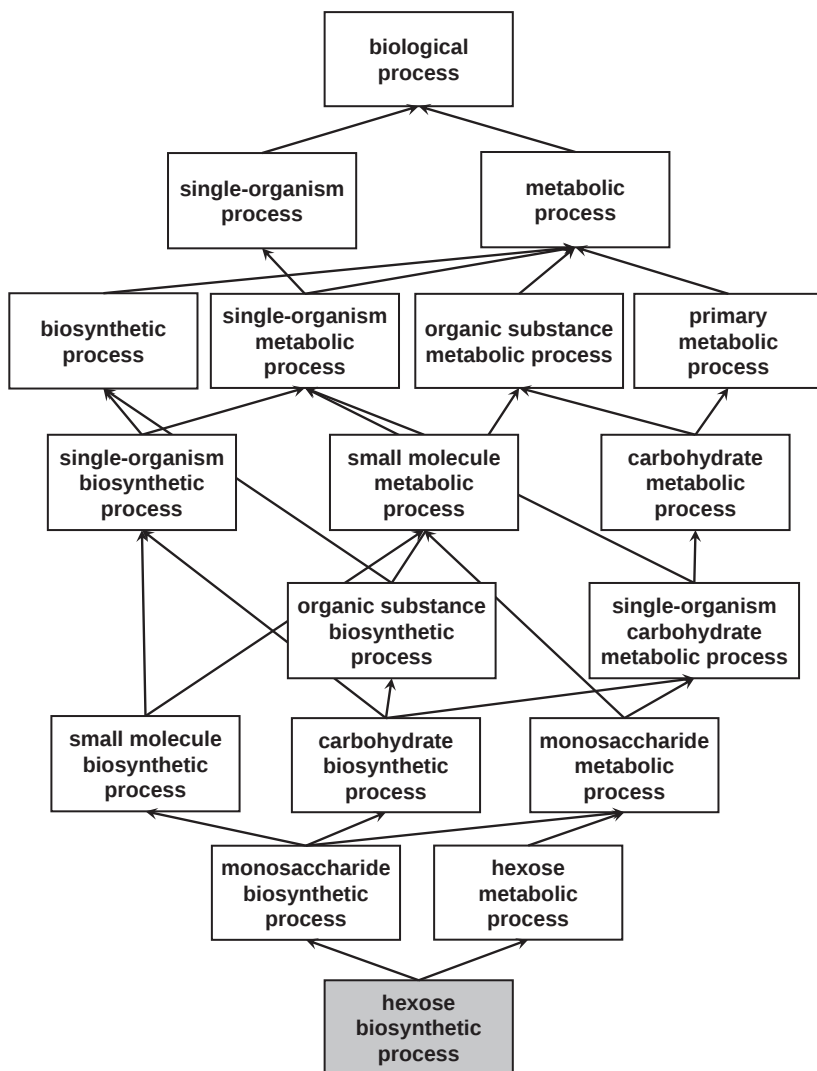


Fig. 4 DAG of the GO term ‘hexose biosynthetic process’ (GO:0019319).

Usage in Computational Applications

Gene Ontology annotations are central for many computationally intensive bioinformatics analyses, including *annotation enrichment* analysis (Masseroli *et al.*, 2004; Huang *et al.*, 2007; Masseroli, 2007; Al-Shahrour *et al.*, 2007; Huang *et al.*, 2009) and *semantic similarity* analysis (Pesquita *et al.*, 2009; Schlicker *et al.*, 2010; Jain and Bader, 2010; Tedder *et al.*, 2010; Falda *et al.*, 2012) of genes or proteins, and for interpretation of biomolecular test results, extraction of new information useful to generate and validate biological hypotheses, and for discovering new knowledge.

For reliable GO annotation prediction and prioritization based on an associated likelihood value, several computational methods have been proposed; they include decision trees (King *et al.*, 2003), Bayesian networks (King *et al.*, 2003), k-nearest neighbour (*k*-NN) (Tao *et al.*, 2007) and support vector machine (SVM) classifiers (Minnecci *et al.*, 2013; Mitsakakis *et al.*, 2013), hidden Markov models (HMM) (Mi *et al.*, 2013; Deng and Ali, 2004), and biological network analysis (Warde-Farley *et al.*, 2010; Li *et al.*, 2007). Also simple latent semantic approaches have been suggested to this purpose (Khatri *et al.*, 2005; Done *et al.*, 2010), and then extended with multiple improvements (Chicco and Masseroli, 2015, 2016; Done *et al.*, 2010, 2007; Pinoli *et al.*, 2014b, 2015). More sophisticated latent semantic indexing (LSI) (Dumais *et al.*, 1988) have been proposed to predict new GO annotations based on available ones; they include the probabilistic latent semantic analysis (pLSA) (Hofmann, 1999; Masseroli *et al.*, 2012), also enhanced with weighting schemes (Pinoli *et al.*, 2013), and latent Dirichlet allocation (LDA) (Blei *et al.*, 2003; Bicego *et al.*, 2010; Perina *et al.*, 2010). The LDA technique, associated with Gibbs sampling (Griffiths, 2002; Casella and George, 1992; Porteous *et al.*, 2008), was used to predict gene annotations to GO terms in Pinoli *et al.* (2014a). Lately, other supervised methods were proposed for the prediction of gene GO annotations (Cheng *et al.*, 2014; Stojanova *et al.*, 2013), although with limited predictive accuracy. Overall, such annotation prediction techniques are general and flexible, but provide only limited accuracy, or improve predictions by using a complex model. The latter ones usually are difficult and time consuming to be set up and slow. Recently, a novel representation of the annotation discovery problem

and a random perturbation method of the available annotations were proposed (Domeniconi *et al.*, 2014); taking advantage of supervised algorithms, they allow accurate predictions of novel GO annotations, also regarding the genes of an organism different from the one whose annotations are used for the prediction (Domeniconi *et al.*, 2016).

Gene Ontology Annotation Repositories

The very numerous gene and gene product annotations to Gene Ontology terms are stored in many heterogeneous repositories, including integrative repositories (e.g., GeneCards (Rebhan *et al.*, 1997), and Genomic and Proteomic Knowledge Base (GPKB) (Masseroli *et al.*, 2016)). Above all, the Gene Ontology Annotation (GOA) database (Huntley *et al.*, 2015) is the largest and most comprehensive open-source archive of GO annotations of gene products in multiple species, providing more than 386 million GO annotations to almost 59 million proteins in more than 707,000 taxonomic groups. Managed by the European Bioinformatics Institute (EBI), its aim is to provide electronic and manual annotations of high-quality, ensured by the use of quality control checks, to the proteins from a wide range of species in the UniProt Knowledgebase (Magrane, 2011); furthermore, its use of the standardized GO vocabulary for such annotations promotes a high level of integration of the knowledge in UniProt with other databases. The annotations from GOA are freely available in a variety of annotation file formats, and are also accessible through FTP and Web browser.

Gene Ontology Issues and Solutions

During its development the Gene Ontology faced some problems, mainly due to its growing complexity and the fast increasing knowledge of the domain it describes. The main problem always at risk is that, to be enough comprehensive, several molecular functions, biological processes, and cellular components that are not common to all life forms are described in the GO, despite it should not contain species-specific concepts. Although the current principle is to define in the GO any concept that can apply to more than one type of organisms, as the GO vocabularies expand species-specific terms become problematic; especially, the use of species-specific anatomical terms stays unresolved.

Conflicts in the GO generally manifest when curators from different organism-specific databases add species-specific terms to describe the gene products of their organism. An example of such conflict previously occurred with the introduction in the GO of the biological process term *chitin metabolic process*, which describes cell wall biosynthesis in fungi and cuticle synthesis in insects. Making this concept a sub-process of both *cell wall biosynthesis* and *cuticle synthesis* generated errors, since searches for genes implied in cell wall biosynthesis found also genes concerned with insect cuticle biosynthesis that were annotated to chitin metabolic process. This conflict was solved by creating two types of chitin metabolic processes: one regarding cell wall biosynthesis (i.e., *cell wall chitin metabolic process*), and one related to cuticle synthesis (i.e., *cuticle chitin metabolic process*) (Hill *et al.*, 2002). Yet, as more species-specific terms are progressively included in the GO vocabularies, it is increasingly difficult to keep both the global interspecies relations for which GO struggles, and the accurate terms needed for intra-species gene annotation.

In spite of problems of this and other types (Smith *et al.*, 2003), the Gene Ontology is extremely useful to relate at best many annotations regarding individual biological concepts, but in different species or merely dispersedly described in heterogeneous databases, supporting automatic searches and semantic clustering of different gene annotations.

Conclusion

The GO project is a very important effort regarding three separate aspects: first, the development and maintenance of the GO sub-ontologies; second, the annotation of gene and gene products, which involves making associations between the GO terms and the genes and gene products in collaborating databases; and third, the development of tools that ease the creation, maintenance and use of the GO sub-ontologies.

The use of GO terms by collaborating databases enables the possibility to perform uniform queries across all of them. Moreover, the GO controlled vocabularies are structured in a way that they can be queried at different levels; for instance, users may query GO annotations to find all gene products in the mouse genome that are involved in signal transduction, or zoom in on all receptor tyrosine kinases that have been annotated. Such structure also enables annotators to characterize features of genes or gene products at different granularity levels, depending on the depth of knowledge available about the entity.

Shared standardized vocabularies are an important aspect to unify biological databases, but further work is still needed as knowledge changes, updates hang back, and individual curators evaluate data differently. The GO aims to be a platform where curators can agree on how and why using a specific term, and how to consistently apply it, e.g., to create relationships between gene products.

All these aspects make the GO a very useful instrument both to annotate and describe gene and gene product functions across species, and to connect at best several functional annotations concerning individual biological concepts but related to different species, or simply sparsely stored in heterogeneous databases. Thus, it allows automatically performing controlled searches, functional enrichment analyses, and semantic clustering of diverse gene annotations, which is very useful in many genomic analyses (e.g., high-throughput gene expression) to group genes and gene products to some biological high level concept/term. Besides helping in better interpreting results of such experimental analyses, this ability is fundamental also in characterizing less known genes: if in the same cellular component an uncharacterized gene is co-expressed with well-characterized genes annotated to some GO biological process, one can infer that the “unknown” gene’s product is likely to act in the same process.

See also: Biological and Medical Ontologies: Introduction. Data Mining: Clustering. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Querying languages and development. The Gene Ontology

References

- Al-Shahrour, F., Minguéz, P., Tárraga, J., *et al.*, 2007. FatiGO + : A functional profiling tool for genomic data. Integration of functional annotation, regulatory motifs and interaction data with microarray experiments. *Nucleic Acids Research* 35 (Web Server Issue), W91–W96.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* 25 (1), 25–29.
- Bicego, M., Lovato, P., Oliboni, B., Perina, A., 2010. Expression microarray classification using topic models. In: *Proceedings of the ACM Symposium on Applied Computing*, pp. 1516–1520.
- Binns, D., Dimmer, E., Huntley, R., *et al.*, 2009. QuickGO: A web-based tool for Gene Ontology searching. *Bioinformatics* 25 (22), 3045–3046.
- Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022.
- Carbon, S., Ireland, A., Mungall, C.J., *et al.*, 2009. AmiGO: Online access to ontology and annotation data. *Bioinformatics* 25 (2), 288–289.
- Casella, G., George, E.I., 1992. Explaining the Gibbs sampler. *American Statistical Association* 46 (3), 167–174.
- Cheng, L., Lin, H., Hu, Y., Wang, J., Yang, Z., 2014. Gene function prediction based on the Gene Ontology hierarchical structure. *PLoS One* 9 (9), e107187.
- Chicco, D., Masseroli, M., 2015. Software suite for gene and protein annotation prediction and similarity search. *IEEE/ACM Transaction on Computational Biology and Bioinformatics* 12 (4), 837–843.
- Chicco, D., Masseroli, M., 2016. Ontology-based prediction and prioritization of gene functional annotations. *IEEE/ACM Transaction on Computational Biology and Bioinformatics* 13 (2), 248–260.
- Deng, X., Ali, H., 2004. A hidden Markov model for gene function prediction from sequential expression data. In: *Proceedings of the IEEE Computational Systems Bioinformatics Conference*, pp. 670–671.
- Domeniconi, G., Masseroli, M., Moro, G., Pinoli, P., 2014. Discovering new gene functionalities from random perturbations of known gene ontological annotations. In: *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR 2014)*, pp. 107–116.
- Domeniconi, G., Masseroli, M., Moro, G., Pinoli, P., 2016. Cross-organism learning method to discover new gene functionalities. *Computer Methods and Programs in Biomedicine* 126, 20–34.
- Done, B., Khatri, P., Done, A., Draghici, S., 2007. Semantic analysis of genome annotations using weighting schemes. In: *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)*, pp. 212–218.
- Done, B., Khatri, P., Done, A., Draghici, S., 2010. Predicting novel human gene ontology annotations using semantic analysis. *IEEE/ACM Transaction on Computational Biology and Bioinformatics* 7 (1), 91–99.
- Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S., Harshman, R., 1988. Using latent semantic analysis to improve access to textual information. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 281–285.
- Eppig, J.T., Smith, C.L., Blake, J.A., 2017. Mouse Genome Informatics (MGI): Resources for mining mouse genetic, genomic, and biological data in support of primary and translational research. *Methods in Molecular Biology* 1488, 47–73.
- Falda, M., Toppo, S., Pescarolo, A., *et al.*, 2012. Argot2: A large scale function prediction tool relying on semantic similarity of weighted Gene Ontology terms. *BMC Bioinformatics* 13 (Suppl. 4), S14.
- Griffiths, T., 2002. Gibbs sampling in the generative model of Latent Dirichlet allocation. *Stanford University* 518 (11), 1–3.
- Hennig, S., Groth, D., Lehrach, H., 2003. Automated Gene Ontology annotation for anonymous sequence data. *Nucleic Acids Research* 31 (13), 3712–3715.
- Hill, D.P., Blake, J.A., Richardson, J.E., Ringwald, M., 2002. Extension and integration of the Gene Ontology (GO): Combining GO vocabularies with external vocabularies. *Genome Research* 12, 1982–1990.
- Hofmann, T., 1999. Probabilistic latent semantic indexing. In: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (RDIR 1999)*, pp. 50–57.
- Huang, D., Sherman, B., Tan, Q., *et al.*, 2007. David bioinformatics resources: Expanded annotation database and novel algorithms to better extract biology from large gene lists. *Nucleic Acids Research* 35 (Web Server Issue), W169–W175.
- Huang, D.W., Sherman, B.T., Lempicki, R.A., 2009. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research* 37 (1), 1–13.
- Huntley, R.P., Sawford, T., Mutwou-Meullenet, P., *et al.*, 2015. The GOA database: Gene Ontology annotation updates for 2015. *Nucleic Acids Research* 43 (Database issue), D1057–D1063.
- Jain, S., Bader, G.D., 2010. An improved method for scoring protein–protein interactions using semantic similarity within the Gene Ontology. *BMC Bioinformatics* 11 (1), 562.
- Khatri, P., Done, B., Rao, A., Done, A., Draghici, S., 2005. A semantic analysis of the annotations of the human genome. *Bioinformatics* 21 (16), 3416–3421.
- King, O.D., Foulger, R.E., Dwight, S.S., White, J.V., Roth, F.P., 2003. Predicting gene function from patterns of annotation. *Genome Research* 13 (5), 896–904.
- Knublauch, H., Oberle, D., Tetlow, P., Wallace, E., 2006. A Semantic Web Primer for Object-oriented Software Developers. W3C. 2006-03-09. Available at: <http://www.w3.org/2001/sw/BestPractices/SE/ODSD/>.
- Li, X., Zhang, Z., Chen, H., Li, J., 2007. Graph kernel-based learning for gene function prediction from gene interaction network. In: *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2007)*, pp. 368–373.
- Magrane, M., UniProt Consortium, 2011. UniProt Knowledgebase: A hub of integrated protein data. *Database* 2011, bar009.
- Masseroli, M., 2007. Management and analysis of genomic functional and phenotypic controlled annotations to support biomedical investigation and practice. *IEEE Transaction on Information Technology in Biomedicine* 11 (4), 376–385.
- Masseroli, M., Canakoglu, A., Ceri, S., 2016. Integration and querying of genomic and proteomic semantic annotations for biomedical knowledge extraction. *IEEE/ACM Transaction on Computational Biology and Bioinformatics* 13 (2), 209–219.
- Masseroli, M., Chicco, D., Pinoli, P., 2012. Probabilistic Latent Semantic Analysis for prediction of Gene Ontology annotations. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2012)*, pp. 2891–2898.
- Masseroli, M., Martucci, D., Pincioli, F., 2004. GFINDER: Genome Function INtegrated Discoverer through dynamic annotation, statistical analysis, and mining. *Nucleic Acids Research* 32 (Web Server Issue), W293–W300.
- Mi, H., Muruganujan, A., Thomas, P.D., 2013. PANTHER in 2013: Modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Research* 41 (Database Issue), D377–D386.
- Minnecci, F., Piovesan, D., Cozzetto, D., Jones, D.T., 2013. FFPred 2.0: Improved homology-independent prediction of Gene Ontology terms for eukaryotic protein sequences. *PLoS One* 8 (5), e63754.
- Mitsakakis, N., Razak, Z., Escobar, M.D., Westwood, J.T., 2013. Prediction of *Drosophila melanogaster* gene function using Support Vector Machines. *BioData Mining* 6 (1), 8.
- OWL2. 2009. Web Ontology Language Document Overview. W3C. 2009-10-27. Available at: <http://www.w3.org/TR/owl2-overview/>.
- Perina, A., Lovato, P., Murino, V., Bicego, M., 2010. Biologically-aware Latent Dirichlet allocation (BaLDA) for the classification of expression microarray. In: *IAPR International Conference on Pattern Recognition in Bioinformatics*, pp. 230–241.
- Pesquita, C., Faria, D., Falcao, A.O., Lord, P., Couto, F.M., 2009. Semantic similarity in biomedical ontologies. *PLOS Computational Biology* 5 (7), e1000443.

- Pinoli, P., Chicco, D., Masseroli, M., 2013. Enhanced probabilistic latent semantic analysis with weighting schemes to predict genomic annotations. In: Proceedings of the IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2013), pp. 1–4.
- Pinoli, P., Chicco, D., Masseroli, M., 2014a. Latent Dirichlet allocation based on Gibbs Sampling for gene function prediction. In: Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2014), pp. 1–8.
- Pinoli, P., Chicco, D., Masseroli, M., 2014b. Weighting scheme methods for enhanced genomic annotation prediction. *Computational Intelligence Methods for Bioinformatics and Biostatistics. LNCS (Lecture Notes in Bioinformatics)*, vol. 8452. pp. 76–89.
- Pinoli, P., Chicco, D., Masseroli, M., 2015. Computational algorithms to predict Gene Ontology annotations. *BMC Bioinformatics* 16 (Suppl. 6), S4.
- Porteous, I., Newman, D., Ihler, A., *et al.*, 2008. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDDM 2008), pp. 569–577.
- Rebhan, M., Chalifa-Caspi, V., Prilusky, J., Lancet, D., 1997. GeneCards: Integrating information about genes, proteins and diseases. *Trends in Genetics* 13 (4), 163.
- Rhee, S.Y., Wood, V., Dolinski, K., Draghici, S., 2008. Use and misuse of the Gene Ontology annotations. *Nature Reviews Genetics* 9 (7), 509–515.
- Schlicker, A., Lengauer, T., Albrecht, M., 2010. Improving disease gene prioritization using the semantic similarity of Gene Ontology terms. *Bioinformatics* 26 (18), i561–i567.
- Smith, B., Williams, J., Schulze-Kremer, S., 2003. The ontology of the gene ontology. In: Proceedings of the AMIA Symposium, pp. 609–613.
- Stojanova, D., Ceci, M., Malerba, D., Dzeroski, S., 2013. Using PPI network autocorrelation in hierarchical multi-label classification trees for gene function prediction. *BMC Bioinformatics* 14, 285.
- Tao, Y., Sam, L., Li, J., Friedman, C., Lussier, Y.A., 2007. Information theory applied to the sparse Gene Ontology annotation network to predict novel gene function. *Bioinformatics* 23 (13), 529–538.
- Tedder, P.M., Bradford, J.R., Needham, C.J., *et al.*, 2010. Gene function prediction using semantic similarity clustering and enrichment analysis in the malaria parasite *Plasmodium falciparum*. *Bioinformatics* 26 (19), 2431–2437.
- Warde-Farley, D., Donaldson, S.L., Comes, O., *et al.*, 2010. The GeneMANIA prediction server: Biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Research* 38 (Web Server Issue), W214–W220.

Relevant Websites

<https://www.ebi.ac.uk/QuickGO/>
European Molecular Biology Library – European Bioinformatics Institute. QuickGO.

<http://www.genecards.org/>
GeneCards.

http://amigo.geneontology.org/amigo/software_list
Gene Ontology Consortium. AmiGO.

<http://www.ebi.ac.uk/GOA/>
Gene Ontology Annotation database.

<http://archive.geneontology.org/latest-termdb/>
Gene Ontology database archive.

<http://www.geneontology.org/GO.evidence.shtml>
Gene Ontology Consortium. Gene Ontology evidence codes.

<http://www.geneontology.org/page/evidence-code-decision-tree>
Gene Ontology Consortium. Gene Ontology evidence code decision tree.

<http://www.geneontology.org/>
Gene Ontology Consortium. Gene Ontology.

<http://www.bioinformatics.deib.polimi.it/GPKB/>
Genomic and Proteomic Knowledge Base (GPKB).

http://www.informatics.jax.org/vocab/gene_ontology/
Mouse Genome Informatics. GO Browser.

Biographical Sketch



Marco Masseroli received the Laurea degree in Electronic Engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in Biomedical Engineering in 1996, from the Universidad de Granada, Spain. He is associate professor in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano, and lecturer of Bioinformatics and Biomedical Informatics. He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomía Patológica, Facultad de Medicina at the Universidad de Granada, Spain, and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health, Bethesda. His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Biological and Medical Ontologies: Protein Ontology (PRO)

Davide Chicco, Princess Margaret Cancer Centre, Toronto, ON, Canada

Marco Masseroli, Polytechnic University of Milan, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The Protein Ontology (PRO) is a structured controlled vocabulary for protein and protein-related entities, and their corresponding relationships. Developed by Natale and colleagues at Georgetown University Medical Center (Washington, DC, United States), the main goal of PRO is to provide a protein-centric ontology, as an alternative to the multiple gene-centric ontological resources already available in the bioinformatics community (Natale *et al.*, 2007, 2017); an example of a popular ontology is the Gene Ontology (Gene Ontology Consortium, 2013, 2015), which often is exploited by methods related to genes (Pinoli *et al.*, 2015; Chicco and Masseroli, 2015, 2016). One of the goals of the Protein Ontology, in fact, is to fill this gap and provide a complete ontological resource about proteins and proteomics to the scientific community.

PRO integrates the protein information already available in UniProtKB (Boutet *et al.*, 2007) by adding semantic ontological relations and representations of proteins and protein complexes, which are currently missing in other databases. Especially, PRO provides specific information related to what proteins and protein complexes can exhibit as a result of biological processes. Authors fill the PRO resource with information obtained by manual curation from the scientific literature, and by large-scale processing of datasets which supply curated protein and pathway information (largely from UniProtKB). PRO curators use UniProtKB to set the name, the synonyms and the definition of the protein product. Term name and definition follow the conventions used for the human proteins. If no human entry exists, the curators assign the term name from the mouse or *Escherichia coli* (Natale *et al.*, 2011).

In addition to the common protein and protein products, PRO also supplies the representation of specific protein types, which permits precise definition of protein products in pathways, complexes, or in disease modeling. This feature can be exploited in proteomics studies where isoforms and modified forms have to be clearly identified, and for biological network representation where event series can be related to specific protein modifications.

We organize this article as follows. After the Introduction section, first we mention the development and statistics of the PRO ontology (see Section Development and Statistics), describe its content and structure (see Section Content and Structure), and the PRO tools and annotations (see Section Tools and Annotations). We then explain a use case related to PRO (see Section Use Case), and mention some projects which took advantage of PRO (see Section Related Applications). We finally list some recent enhancements of PRO (see Section Recent Enhancements) and draw some conclusions (see Section Conclusions).

Development and Statistics

Natale and colleagues first introduced the Protein Ontology in 2006 (Natale *et al.*, 2007), and then released new versions in 2011 (Natale *et al.*, 2011), in 2014 (Natale *et al.*, 2014) and in 2017 (Natale *et al.*, 2017). Bult *et al.* (2011) described a representation of protein complexes in PRO in 2011, and Arighi *et al.* (2011) published a tutorial on PRO in the same year.

The number of PRO protein entities has always augmented, since its first release in 2007. At the time of writing this article, in the current PRO version 52.0, its stats Web page lists 213,457 total protein terms, 1674 annotations derived from curated papers, 4419 annotations derived from Gene Ontology (GO) terms, and 130,442 protein sequences derived from UniProt.

Content and Structure

Authors designed the Protein Ontology following the Open Biological and Biomedical Ontology (OBO) Foundry principles (Smith *et al.*, 2007). Thanks to some of these principles, it has an open copyright license that allows anyone to use and reuse it; it is written in common format and syntax, whose rules are respected in each aspect and data of the ontology; each data element has an associated unique Uniform Resource Identifier (URI); it has a large documentation available. In addition, each Protein Ontology entry has a specific ID code, and an associated Web page and URL address.

The ontology is organized in five meta-classes, in hierarchical order (Arighi, 2011):

- *Family*: representing classes of proteins derived from a specific set of common ancestor genes. An example of this class entity is PRO:000000676. This protein entity is “A protein with amino- and carboxyl-terminal intra-cellular domains separated by a domain (common with other ion channels) containing six transmembrane helices (S1–S6) in which the last two helices (S5 and S6) flank a loop, called the pore loop, which determines ion selectivity”;

- *Gene*: representing proteins as direct products of a specific gene. An example of this class entity is PR:000000705. This entity is: “A potassium/sodium hyperpolarization-activated cyclic nucleotide-gated channel protein that is a translation product of the human HCN1 gene or a 1:1 ortholog thereof”;
- *Sequence*: representing protein products with a distinct sequence on initial translation. An example of this class entity is PRO:000003420, which is “A PC4 and SFRS1-interacting protein isoform 1 that has not been subjected to any co- or post-translational residue modification or peptide bond cleavage” corresponding to isoform 1 (p75) derived from gene LEDG.
- *Modification*: representing protein products evolved from a single messenger RNA (mRNA) species that differ because of some change that occurs after the initiation of translation. An example of this class entity is PR:000003423, which is “A PC4 and SFRS1-interacting protein isoform 2 that has not been subjected to any co- or post-translational residue modification or peptide bond cleavage”;
- *Complex*: representing protein complexes with a specific defined subunit composition. An example of this class entity is PR:000035566, that is “A protein complex that is composed of BUB1 and BUB3. This complex is involved in the inhibition of anaphase-promoting complex or cyclosome (APC/C) when spindle-assembly checkpoint is activated”.

In PRO, two different types of relations can link two protein entities (Natale *et al.*, 2011):

- *is_a* indicates the subtype relation between child and parent term;
- *derives_from* indicates the relation between a proteolytic cleavage product and its fully-formed precursor.

PRO takes advantage of the popular sequence database UniProtKB (Boutet *et al.*, 2007), and relates with the biological ontology Gene Ontology. In PRO, the organism-specific clusters of proteins encoded by a specific gene link to protein products documented in UniProtKB. The Protein Ontology both uses UniProtKB as a source of protein entities, and complements this database by supplying formal definition of protein products, together with their relationships in the ontological context. PRO links also to the Gene Ontology, regarding its complex meta-class. The representations of organism-specific protein complexes in PRO are sub-classes of the organism-independent protein complex terms in the Cellular Component (CC) GO sub-ontology: multiple GO Cellular Component terms are parent terms for PRO complex terms.

The Protein Ontology furnishes protein data related not only to human genes, but also references to other organisms such as mouse, *Escherichia coli*, *Saccharomyces cerevisiae*, and *Arabidopsis thaliana*. In the last years, the Protein Ontology curators also added the mappings of the PRO entities to external databases such as UniProtKB (Boutet *et al.*, 2007), Reactome (Croft *et al.*, 2014), EcoCyc (Keseler *et al.*, 2005), plus gene mappings for Mouse Genome Database (Bult *et al.*, 2008), HUGO Gene Nomenclature Committee (HGNC) (Povey *et al.*, 2001), EcoGene (Rudd, 2000), and the Saccharomyces Genome Database (Issel-Tarver *et al.*, 2002). Multiple PRO entities, in fact, are associated with entities of the aforementioned resources through identity relationships. The same proteomics concept can be present in both PRO and Reactome, with different ID's, linked through a PRO external relation.

We report an example of Protein Ontology entity in Fig. 1.

Tools and Annotations

The Protein Ontology can be accessed through the internet, can be explored through Cytoscape (Smoot *et al.*, 2011), or can be downloaded into plain files. PRO also provides a set of annotations related to protein and protein entities. PRO curators, in fact, analyze the scientific literature to find references to proteins and protein complexes in order to collect protein annotations and their associated evidence. PRO includes annotations to attribute (such as protein function, localization, process and involvement in disease) in the PRO association file (PAF), available for download.

PRO expresses each annotation as an association between a protein entity and a term from controlled vocabularies (such as the Gene Ontology and others), and their corresponding relationship (Natale *et al.*, 2011). In the aforementioned example (Fig. 1), the annotations of the protein *smad2 isoform 1 phosphorylated 1* are listed by PRO in the lower part of the record. The example shows that this isoform is linked to the “transcription coactivator activity” (GO:0003713), which is a molecular function GO term, through a “has_function” relation.

In addition, the Protein Ontology allows the users to submit their own protein products and protein annotations, through a tool called Rapid Annotation interfaCE for PRotein Ontology (RACE PRO). On its online Web page, anyone can suggest a protein-associated entity by supplying a scientific paper related to it. This way, inexperienced users with no specific knowledge of proteomics can contribute to PRO as well. After the receiving of a request, the PRO editor checks the information and, if the request is accepted, converts it into the proper PRO format. Finally, he/she sends it back to the original submitter for confirmation and, after his/her final approval, the editor adds it to the Protein Ontology.

PRO is also accessible through BioPortal, an online resource of biomedical ontologies, provided by the National Center for Biomedical Ontology (Musen *et al.*, 2012).

Use Case

An interesting and complete use case for the usage of the Protein Ontology has been described by Ross *et al.* (2013), which took advantage of PRO to analyze a biological process called *spindle checkpoint*. The spindle checkpoint is a highly conserved



Protein Ontology Report - hSMAD2/iso:1/Phos:1

PR:000025934 - http://purl.obolibrary.org/obo/PR_000025934[Annotations](#)

Ontology Information	
PRO ID	PR:000025934 Show OBO stanza / PAF
PRO name	smad2 isoform 1 phosphorylated 1 (human)
Synonyms	PRO-short-label: hSMAD2/iso:1/Phos:1 PRO-proteform-std: UniProtKB:Q15796-1, Ser-465/Ser-467, MOD:00046
Definition	A smad2 isoform 1 phosphorylated 1 in human. This form is phosphorylated in the two last Ser of the C-terminal SSxS motif. UniProtKB:Q15796-1 , Ser-465/Ser-467, MOD:00046 . [PMID:8980228 , PMID:9346966]
Comment	Kinase=(“TGFB1”; PR:000025963; Ser-465/Ser-467).
PRO Category	organism-modification
Parent	PR:000000650 smad2 isoform 1 phosphorylated 1
Term Hierarchy Visualization	DAG OLS Cytoscape

Related Cross References

Db Identifiers [Reactome:R-HSA-177099](#)

Functional Annotation

[PRO Centric View](#)[GO Centric View](#)

PRO Term	GO Annotation	Evidence
PR:000025934 hSMAD2/iso:1/Phos:1 Ser-465/Ser-467, MOD:00046	has_function GO:0046332 SMAD binding with PR:Q13485 hSMAD4	PMID:9311995
	has_function GO:0003713 transcription coactivator activity	PMID:9689110
	has_function GO:0030618 transforming growth factor beta receptor, pathway-specific cytoplasmic mediator activity	PMID:8980228 , PMID:9346966
	has_modification MOD:00046 O-phospho-L-serine	PMID:9136927
	located_in GO:0005634 nucleus	PMID:9006934 , PMID:9346966
	participates_in GO:0007183 SMAD protein complex assembly	PMID:9346966
	participates_in GO:0006355 regulation of transcription, DNA-templated	PMID:15690394 , PMID:9689110
	participates_in GO:0007165 signal transduction	PMID:8752209 , PMID:8980228 , PMID:9346966
	participates_in GO:0007179 transforming growth factor beta receptor signaling pathway	PMID:8752209 , PMID:8980228

Fig. 1 Example of a Protein Ontology entry in the PRO browser: PR:000025934, corresponding to “A smad2 isoform 1 phosphorylated 1 in human. This form is phosphorylated in the two last Ser of the C-terminal SSxS motif”. Its category is organism-modification. This isoform is also present in Reactome with the listed ID code R-HSA-177099 and the corresponding hyper-link. The browser lists the Gene Ontology terms associated to this protein entity in the lower part of the screenshot, including the hyper-links for the Gene Ontology terms, and the references to the PubMed scientific papers describing the relation between the GO term and the protein entity (e.g., PMID:9311995).

biological process strongly related to protein modification and protein complex formation. The authors took advantage of the Protein Ontology and of other bioinformatics tools to explore the cross-species conservation of spindle checkpoint proteins, including phosphorylated forms and complexes. The authors also studied the impact of phosphorylation on spindle checkpoint function, and the interactions of spindle checkpoint proteins with the site of checkpoint activation (kinetochore) (Ross *et al.*, 2013).

Authors started their analysis with a PubMed literature search of the keywords *BubR1* and *Bub1* (which are the names or synonyms of the checkpoint protein *BUB1*), and *Mad3*, which is the closest yeast relative of *BUB1B*. The authors collect all the information related to the proteins for which there were experimental data available, from PubMed and other datasets, and inputted them into the previously introduced RACE-PRO. All the information submitted by the authors to RACE-PRO were checked by the PRO editor and converted into PRO terms through a semi-automated process which also establishes the hierarchy of each term in the ontology. Once these terms were available on PRO, the authors browsed them through the PRO website and its graphical display tools, such as the Cytoscape view (Smoot *et al.*, 2011).

Through the Cytoscape view, the authors were able to identify the kinetochore protein-protein interaction (PPI) network for the *spindle checkpoint*. Using a script, the authors extracted the name, definition, category, and label of the PRO terms retrieved and of the kinase-substrate relationships. The authors extracted the protein binding related annotations (identified by the GO evidence code “inferred from physical interaction” or IPI) from the PAF file provided by PRO. Their script then generated two tab-delimited text files, which are importable into Cytoscape: a network file containing each pair of interacting proteins, its interaction type, and the corresponding evidence and a PRO entry information file containing PRO ID and entity description (Ross *et al.*, 2013). By analyzing this list of PRO terms, authors were able to infer interesting relationships and new terms regarding the input spindle checkpoint proteins.

Related Applications

Many scientific projects have used the Protein Ontology in the past; here we list the main applications available in literature.

Ross *et al.* (2013) described a case study in which they took advantage of PRO to analyze a biological process called *spindle checkpoint*. We explore more in depth this case study in Section Use Case. Rocca-Serra *et al.* (2011), on the other hand, took advantage of PRO to associate chemical and enzymatic probes to protein products. In this project, the authors propose a standard method for reporting the results of mapping experiments for the characterization of single nucleotide resolution nucleic acid structure. One of the steps of this method is the association of protein entity information to chemical enzymes: authors perform this operation by querying the Protein Ontology.

The Protein Ontology can also be used as a reference support for text mining. Funk *et al.* (2014) in fact, exploited it to test dictionary-based systems for biomedical concept recognition. Maiorana (2012) used the Protein Ontology to retrieve protein information related to Huntington Disease. A typical usage of the Protein Ontology was shown by Li *et al.* (2016) recently. The authors of this article, studying the types of human dental follicle cells (DFCs) and periodontal ligament cells (PDLs), took advantage of PRO and identified 2138 proteins related to those cell types. They then identified 39 of these proteins as consistently differentially expressed between DFCs and PDLs. To complete their study, the authors analyzed these 39 proteins by querying the Protein Ontology, inferring additional information about them and their products.

Recent Enhancements

The Protein Ontology curators released a new version of PRO in October 2016, featuring several enhancements (Natale *et al.*, 2017). One of the goals of the latest and current version of PRO at the time of writing this article, in fact, is to provide a standard representation of proteoforms by employing UniProtKB as a sequence reference, and Protein Structure Initiative Modification Ontology (PSI-MOD) as a post-translational modification reference. This feature is able to let users retrieve information about proteoforms more easily across several resources.

Another enhancement regards scalability and text mining. Authors included iPTMnet, a tool for the integration of post-translational modification (PTM) information from curated databases, and the results of broad text mining of PubMed abstracts performed with RLIMS-P (Torii *et al.*, 2015).

Also the Histone database (Khare *et al.*, 2012), which provides a collection of human histone modifications, has been included in the last version of PRO. The PRO curators downloaded the Histone data and converted them by a script to generate their PRO instances. In the first integration, authors produced 468 Histone evidenced PRO terms.

Another novelty of the latest version of PRO regards the dynamic generation of terms. Up to the previous PRO version, data inclusion requests were filled fully manually by the PRO curators. To quicken this process, the PRO authors have added the capability for some specific terms (gene-level and sequence-level terms) to be generated dynamically, from UniProtKB. As the authors explain (Natale *et al.*, 2017), UniProtKB-derived terms can be defined as “a protein that is a translation product of gene X in organism Y”. Since these items of information are already available in the UniProtKB database for that entry, PRO takes advantage of UniProtKB's Web service and dynamically adds the entry as a PRO term in its database.

While other novelties refer to the Web interface of PRO, it is worth mentioning that the last version of PRO also links to SPARQL (Quilitz and Leser, 2008). The PRO curators produced a resource description framework (RDF) linked data repository that contains information about the PRO term file and the associated PRO annotation file. Upon this RDF, the authors developed a SPARQL endpoint server for PRO through OpenLink Virtuoso (Erling and Mikhailov, 2009). This feature allows the PRO users to query against PRO data following the W3C SPARQL specification, and therefore retrieve terms, subclasses, and functional annotations from PRO, with or without inference. The PRO SPARQL endpoint is publicly accessible over the Internet.

Conclusions

The Protein Ontology curators started this project with an ambitious goal: create a new ontological resource for proteomics, which might have been as useful and handy as the popular Gene Ontology for all the scientific community. Even if PRO contains multiple proteomics entities and is linked to multiple external resources, it is still not as spread as GO.

The PRO curators are currently working on developing new collaborations with external scientists for future PRO extensions and integrations. One of the future integrations of PRO will be with the Immunology and Data Analysis Portal (ImmPort) (Bhattacharya *et al.*, 2014), an immunology data resource whose integration will make PRO able to associate protein entities to antibodies, among others. The curators are also working on linking together PRO and the Proteoform Repository, a new databank for proteoforms developed at the Northwestern University (Chicago, Illinois, United States). With this new extension, PRO will enable users to analyze conserved modifications among *Homo sapiens* and other organisms, to analyze the associations between proteins and function information contained in Proteoform Repository, and to analyze relationships between proteoforms.

With these new integrations and a better communication campaign, we believe that the Protein Ontology can become the reference proteomics ontology for the scientific community.

See also: Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: Introduction. Data Mining: Clustering. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Querying languages and development

References

- Arighi, C.N., 2011. A tutorial on protein ontology resources for proteomic studies. *Bioinformatics for Comparative Proteomics*. 77–90.
- Bhattacharya, S., Andorf, S., Gomes, L., *et al.*, 2014. ImmPort: Disseminating data to the public for the future of immunology. *Immunologic Research* 58 (2–3), 234–239.
- Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bairoch, A., 2007. UniProtKB/Swiss-Prot: The manually annotated section of the UniProt knowledgebase. *Plant Bioinformatics: Methods and Protocols*. 89–112.
- Bult, C.J., Drabkin, H.J., Eysikov, A., *et al.*, 2011. The representation of protein complexes in the Protein Ontology PRO. *BMC Bioinformatics* 12 (1), 371.
- Bult, C.J., Eppig, J.T., Kadin, J.A., Richardson, J.E., Blake, J.A., 2008. The mouse genome database mgd: Mouse biology and model systems. *Nucleic Acids Research* 36 (Suppl. 1), D724–D728.
- Chicco, D., Masseroli, M., 2015. Software suite for gene and protein annotation prediction and similarity search. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12 (4), 837–843.
- Chicco, D., Masseroli, M., 2016. Ontology-based prediction and prioritization of gene functional annotations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13 (2), 248–260.
- Croft, D., Mundo, A.F., Haw, R., *et al.*, 2014. The reactome pathway knowledgebase. *Nucleic Acids Research* 42 (D1), D472–D477.
- Erling, O., Mikhailov, I., 2009. RDF support in the Virtuoso DBMS. In: Pellegrini, T., Auer, S., Tochtermann, K., Schaffert, S. (Eds.), *Networked Knowledge-Networked Media* 221. Heidelberg: Springer, pp. 7–24.
- Funk, C., Baumgartner, W., Garcia, B., *et al.*, 2014. Large-scale biomedical concept recognition: An evaluation of current automatic annotators and their parameters. *BMC Bioinformatics* 15 (1), 59.
- Gene Ontology Consortium, 2013. Gene ontology annotations and resources. *Nucleic Acids Research* 41 (D1), D530–D535.
- Gene Ontology Consortium, 2015. Gene ontology consortium: Going forward. *Nucleic Acids Research* 43 (D1), D1049–D1056.
- Issel-Tarver, L., Christie, K.R., Dolinski, K., *et al.*, 2002. *Saccharomyces* genome database. *Methods in Enzymology* 350, 329–346.
- Keseler, I.M., Collado-Vides, J., Gama-Castro, S., *et al.*, 2005. EcoCyc: A comprehensive database resource for *Escherichia coli*. *Nucleic Acids Research* 33 (Suppl. 1), D334–D337.
- Khare, S.P., Habib, F., Sharma, R., *et al.*, 2012. Histone relational knowledgebase of human histone proteins and histone modifying enzymes. *Nucleic Acids Research* 40 (D1), D337–D342.
- Li, J., Li, H., Tian, Y., *et al.*, 2016. Cytoskeletal binding proteins distinguish cultured dental follicle cells and periodontal ligament cells. *Experimental Cell Research* 345 (1), 6–16.
- Maiorana, F., 2012. A semantically enriched medical literature mining framework. In: *Proceedings of the 25th International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, pp. 1–4.
- Musen, M.A., Noy, N.F., Shah, N.H., *et al.*, 2012. The national center for biomedical ontology. *Journal of the American Medical Informatics Association* 19 (2), 190–195.
- Natale, D.A., Arighi, C.N., Barker, W.C., *et al.*, 2007. Framework for a protein ontology. *BMC Bioinformatics* 8 (9), S1.
- Natale, D.A., Arighi, C.N., Barker, W.C., *et al.*, 2011. The protein ontology: A structured representation of protein forms and complexes. *Nucleic Acids Research* 39 (Suppl. 1), D539–D545.
- Natale, D.A., Arighi, C.N., Blake, J.A., *et al.*, 2014. Protein ontology: A controlled structured network of protein entities. *Nucleic Acids Research* 42 (D1), D415–D421.
- Natale, D.A., Arighi, C.N., Blake, J.A., *et al.*, 2017. Protein ontology PRO: Enhancing and scaling up the representation of protein entities. *Nucleic Acids Research* 45 (D1), D339–D346.
- Pinoli, P., Chicco, D., Masseroli, M., 2015. Computational algorithms to predict Gene Ontology annotations. *BMC Bioinformatics* 16 (Suppl. 6), S4.
- Povey, S., Lovering, R., Bruford, E., *et al.*, 2001. The HUGO gene nomenclature committee HGNC. *Human genetics* 109 (6), 678–680.
- Quillit, B., Leser, U., 2008. Querying distributed RDF data sources with SPARQL. In: *European Semantic Web Conference*. Springer, pp. 524–538.
- Rocca-Serra, P., Bellaousov, S., Birmingham, A., *et al.*, 2011. Sharing and archiving nucleic acid structure mapping data. *RNA* 17 (7), 1204–1212.
- Ross, K.E., Arighi, C.N., Ren, J., *et al.*, 2013. Use of the protein ontology for multi-faceted analysis of biological processes: A case study of the spindle checkpoint. *Frontiers in Genetics* 4.
- Rudd, K.E., 2000. EcoGene: A genome sequence database for *Escherichia coli* K-12. *Nucleic Acids Research* 28 (1), 60–64.
- Smith, B., Ashburner, M., Rosse, C., *et al.*, 2007. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25 (11), 1251–1255.
- Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.-L., Ideker, T., 2011. Cytoscape 2.8: New features for data integration and network visualization. *Bioinformatics* 27 (3), 431–432.
- Torii, M., Arighi, C.N., Li, G., *et al.*, 2015. RLIMS-P 2.0: A generalizable rule-based information extraction system for literature mining of protein phosphorylation information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12 (1), 17–29.

Relevant Websites

http://pir.georgetown.edu/cgi-bin/pro/sta_pro
 PRO Statistics.
http://purl.obolibrary.org/obo/PR_000035566
 Protein Ontology Report - BUB1:BUB3 PR:000035566.
http://purl.obolibrary.org/obo/PR_000000705
 Protein Ontology Report - HCN1 PR:000000705.
http://purl.obolibrary.org/obo/PR_000000676
 Protein Ontology Report - potassium/sodium hyperpolarization-activated cyclic nucleotide-gated channel protein PR:000000676.
http://purl.obolibrary.org/obo/PR_000003423
 Protein Ontology Report - PSIP1/iso:2/UnMod PR:000003423.
http://purl.obolibrary.org/obo/PR_000003420
 Protein Ontology Report - PSIP1/iso:1/UnMod PR:000003420.
http://pir.georgetown.edu/cgi-bin/pro/race_pro
 RACE-PRO: Rapid Annotation interfaCE for PRotein Ontology.
http://proconsortium.org/pro/pro_sparql.shtml
 SPARQL Endpoint for Protein Ontology.
<http://pir.georgetown.edu/pro/>
 The Protein Ontology (PRO) website.

Biographical Sketch

Davide Chicco obtained his Bachelor of Science and Master of Science degrees in computer science at Università di Genova (Genoa, Italy), respectively in 2007 and 2010. He then started the PhD program in computer engineering at Politecnico di Milano university (Milan, Italy), where he graduated in Spring 2014. He also spent a semester as visiting doctoral scholar at University of California Irvine (Irvine, California, United States). Since September 2014, he has been a post-doctoral researcher at the Princess Margaret Cancer Center and guest at University of Toronto (Toronto, Ontario, Canada). His research topics focus upon mainly machine learning algorithms applied to bioinformatics.

Marco Masseroli received the Laurea degree in electronic engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in biomedical engineering in 1996, from the Universidad de Granada (Granada, Spain). He is associate professor and lecturer of bioinformatics and biomedical informatics in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano university (Milan, Italy). He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomía Patológica, Facultad de Medicina at the Universidad de Granada (Granada, Spain), and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health (Bethesda, Maryland, United States). His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Biological and Medical Ontologies: Disease Ontology (DO)

Anna Bernasconi and Marco Masseroli, Politecnico di Milano, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The nomenclature of human diseases has traditionally been defined according to the name of the scientist who discovered the disease (e.g., *Larsen Syndrome*), or where the disease was first revealed (e.g., *Japanese spotted fever*), or the animal in which the disease was first identified (e.g., *Cowpox*). Across the multitude of different resources, this tradition highlights a strong need for a standardized way of representing human diseases, in order to boost interoperability between resources, to connect gene variation to phenotype targets, and to support the use of computational tools that can use disease information to perform important data analysis and integration.

The Disease Ontology (DO) provides a single unifying representation of biomedical data that are associated with human diseases. It contains standardized disease descriptors that are both human- and computer-readable. DO terms are well-defined and linked (with frequent updates) to established terminologies that contain disease and disease-related concepts, such as Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) (Donnelly, 2006), International Classification of Diseases (ICD-9 and ICD-10 versions) (see 'ICD, World Health Organization' in Relevant Websites section), Medical Subject Headings (MeSH) (Lipscomb, 2000), and Unified Medical Language System (UMLS) (Bodenreider, 2004).

The disease descriptors of the DO have been created by combining project specific genetic/genomic disease terms with resources deriving from the study of human disease through the analysis of model organisms. The actual corpus of DO diseases includes genetic, environmental, and infectious diseases of interest both for clinical research and medicine. It encompasses multiple mammalian genomic resources enabling inference between different datasets that use the included standard terminologies to describe disease.

The activities that mostly benefit from semantically consistent disease annotations (like the ones made possible by the DO) include diagnostic evaluation, treatment and clinical or experimental data comparisons over time and between different studies. Before the DO was developed, many vocabularies and ontologies included disease and disease-related concepts, but none of these was structured around the concept of disease.

The DO has become an established reference point for any reasoning that involves finding relationships between disease concepts, genes contributing to disease and the plethora of symptoms, findings, and signs associated with disease. Its authority is by now widely recognized by the research community, since it encapsulates a comprehensive theory of disease.

The Disease Ontology was first developed by the Center for Genetic Medicine of Northwestern University (Chicago, Illinois, United States) in 2003. The initial drive was the data curation need of the NUGene Project at the Center for Genetic Medicine. The first releases (in 2003 and 2004) referred to the International Classification of Diseases in its 9th version (ICD-9) as their foundational vocabulary, reorganizing its terms by process, affected system, and cause (including metabolic and genetic disorders, as well as infectious diseases). Since ICD-9 is a set of billing codes rather than a proper disease terminology, it features highly variable granularity of terms; thus, the DO development team soon realized that it was not the right choice as basis for the DO. A re-visitation of the initial choices led to building a new version of the DO based on disease concepts in UMLS, taking advantage of them to create mappings to SNOMED-CT and ICD-9.

The version of the ontology described in Schriml *et al.* (2012) had a strong impact on the development of the following biomedical resources, as documented by the high number of references to the paper in the biology and bioinformatics literature. The new DO followed the Open Biomedical Ontology (OBO) and OBO Foundry principles. The DO became open source and maintained in a well-defined exchange format (OBO and OWL), with clearly specified content, definitions, and relations (i.e., *is_a* and *part_of*) that are formatted according to the OBO Relations Ontology standards.

Between 2012 and 2015 the content of the DO has been revised 192 times with the addition of many terms. In the version of the ontology described in Kibbe *et al.* (2015) the content has grown notably, the data structure has undergone improvements, the community-based organization has switched from OBO to Web Ontology Language (OWL) based curation of the DO.

Advances have been mainly dedicated to data content: representing common and rare diseases (thanks to the collaboration with OMIM (Hamosh *et al.*, 2005), Orphanet (Rath *et al.*, 2012), and the National Organization for Rare Disorders (see 'NORD' in Relevant Websites section)), assessing genomic variants of cancers, defining human disease according to model organism databases, homogenizing unequal disease representations, and connecting disease and phenotype information.

The curatorial efforts have targeted in particular rare genetic diseases, cardiovascular diseases, neurodegenerative diseases, inherited metabolic disorders, diseases related to intellectual disabilities and cancer. Still at the time of writing, the research community is always at the forefront in guiding the curation of the DO by requesting the inclusion of new terms and the revision of groups of terms utilized in the research.

Mission and scope of the DO project are clearly stated on its main Web page, aiming at gathering an always greater plurality of users, who participate to the DO collaborative, community-driven, development. It also features a Wiki page that provides an overview of the project, interesting links and contacts. In August 2006 it was submitted for inclusion and review to the OBO Foundry, and it is now an OBO Foundry ontology.

The DO has been funded by the National Center for Research Resources (NCRR) of the National Institutes of Health (NIH). The comprehensive project is hosted at the Institute for Genome Sciences at the University of Maryland School of Medicine (Baltimore, Maryland, United States).

In May 2017, at the time of writing this article, the BioPortal (Noy *et al.*, 2009) website listed 6 projects using this ontology: Biomedical Semantic QA (Balikas *et al.*, 2015), DisGeNET-RDF (Piñero *et al.*, 2017), Epidemic Marketplace (Silva *et al.*, 2010), FlyBase (Gramates *et al.*, 2017), Gemma (Zoubarev *et al.*, 2012), and Neurocarta (Portales-Casamar *et al.*, 2013).

After this introduction, this article first describes the DO with focus on its scope, structure, terms, and relations. The following sections include: a quantitative perspective on the coverage and mappings provided by the ontology in its different releases, an overview of the technological aspects regarding its access, Web browser and API, a list of notable applications that employ the ontology in their work, and information about the ongoing changes and expected future developments, along with a use case scenario describing a simple user activity.

The Disease Ontology

The DO is intended as a single relational structure to organize the classification of disease, able to unify the representation of disease among various terminologies. It allows inference and reasoning using the disease concepts and their relationships, and it is optimized to facilitate disease annotations of other sources.

Scope of the DO

The Disease Ontology is based on the definition of disease: “a disposition to undergo pathological processes that exists in an organism because of one or more disorders in that organism”. This definition was given by the Ontology of General Medical Science (OGMS), as documented in Scheuermann *et al.* (2009).

The DO, like the model Gene Ontology, arranges *terms* (also referred to as *classes*, which is used as a synonym in this article) in a Directed Acyclic Graph (DAG) in such a way that traversing it from the root toward the leaves corresponds to progressively moving toward more specific terms. Its organization ensures that the true-path rule (Gene Ontology Consortium, T.G.O., 2001) always holds.

The DO includes only concepts that strictly represent diseases. Note that concepts related to the progression of disease (e.g., early, late, metastasis, stages) and manifestations of disease (e.g., transient, acute, chronic) are not included as part of a disease definition. Moreover, terms describing the combination of two disease concepts (e.g., *glaucoma associated with pupillary block*) are not included, but rather the two diseases are represented through two separate terms (i.e., *glaucoma* and *pupillary block*).

Structure of the DO

The DO is logically divided by major types of disease to enable a guided exploration of the ontology (e.g., using its Web browser).

The DO is organized according to a clinical perspective of disease etiology (i.e., disease origins or causes) and location. Its graph-structure presents terms linked by hierarchical and computable relationships (e.g., *brain sarcoma* is_a *brain cancer*, and *brain cancer* is_a *central nervous system cancer*). The terms are divided into eight high-level bins to represent cellular proliferation (i.e., *benign neoplasm*, *cancer*, or *pre-malignant neoplasm*), mental health (e.g., *cognitive disorder*), anatomical entities (e.g., *endocrine*, *cardiovascular*, or *gastrointestinal disease*), infectious agents (i.e., *bacterial*, *fungal*, *parasitic*, or *viral infectious disease*), metabolism diseases (i.e., *acquired* or *inherited metabolic disease*), genetic diseases (e.g., *chromosomal disease*), medical disorders (e.g., *visceral heterotaxy*), and syndromes (e.g., *chronic fatigue syndrome*).

In the ontology, all terms are anchored by stable and traceable identifiers (DOIDs) that consist of the prefix DOID: followed by number. As an example, *Parkinson's disease* has the identifier DOID:14330.

Example term

A relevant advantage of expressing information on the phenotype in form of an ontology is that search procedures can exploit the semantic relationships between the used ontology terms as the example below shows. The Disease Ontology entry for *Lateral sclerosis* in OBO format is given in Table 1, featuring its identification number (DOID) within the ontology, name, textual definition, external links to other classification schemes, subsets it belongs in (i.e., subsets of biomedical ontologies, called *slims*, created to provide customized high-level views (Davis *et al.*, 2010)), synonyms, and relations it participates in.

Relations in the DO

Textual definitions of the ontology terms include *relations* (also called *properties*), taken from a defined set of possibilities. The DO features a total of 15 properties (as defined in the OWL version of the ontology).

Table 2 shows the properties that are used to annotate concepts in the OWL version of the ontology that is released at the time of writing this article. Note that the column *Relation* reports the name of the ObjectProperty as it is used in the OWL version file of the ontology. The column *Use* indicates the description of the AnnotatedTarget field in the OWL file data type description, and the column *Example* shows a sample value of the AnnotatedTarget field of one ontology entry.

Table 1 Example of DO term: Lateral sclerosis

<i>Id</i>	<i>DOID:230</i>
Name	Lateral sclerosis
Definition	A motor neuron disease characterized by painless but progressive weakness and stiffness of the muscles of the legs. Available at: http://en.wikipedia.org/wiki/Primary_lateral_sclerosis/ , https://rarediseases.org/rare-diseases/primary-lateral-sclerosis/
Xrefs	ICD10CM: G12.29 ICD9CM:335.24 MESH: D016472 OMIM:611637 ORDO:35689 SNOMEDCT_US_2016_03_01:81211007 UMLS_CUI:C0154682
Subsets	DO_rare_slim
Synonyms	Adult-onset primary lateral sclerosis [EXACT] Primary lateral sclerosis [EXACT] [ICD9CM_2006:335.24]
Relationships	is_a DOID:231 motor neuron disease

Table 2 List of relations in the disease ontology

<i>Relation</i>	<i>Use</i>	<i>Example</i>
complicated_by		DOID:6905, A herpes zoster that is <i>complicated_by</i> AIDS.
composed_of	Component parts of anatomy of tissue made up of certain cells or other body area/system or tissue types.	DOID:2661, A sweat gland neoplasm that is <i>composed_of</i> outgrowths of myoepithelial cells from a sweat gland.
derives_from	Type of tissue or cell/the source of the material.	DOID:2671, A carcinoma that <i>derives_from</i> transitional epithelial cells.
has_material_basis_in		DOID:2762, A bone cancer that <i>has_material_basis_in</i> abnormally proliferating cells <i>derives_from</i> epithelial cells.
has_symptom	Symptom(s) associated with disease.	DOID:3049, A vasculitis that is systemic vasculitis realized as blood vessel inflammation and <i>has_symptom</i> asthma along with hay fever, rash and gastrointestinal bleeding.
is_a		DOID:1311, A renal infectious disease <i>is_a</i> Human immunodeficiency virus infectious disease [...].
located_in	Anatomical location.	DOID:1324, A respiratory system cancer that is <i>located_in</i> the lung.
results_in	Development process/cause and effect/disease progress, process.	DOID:0050068, A bubonic plague that <i>results_in</i> a benign form of bubonic plague [...].
results_in_formation_of	Formation of structure, cancer, etc./cause and effect.	DOID:0050096, A dermatophytosis that [...] <i>results_in_formation_of</i> noninflammatory superficial perfollicular pustules.
transmitted_by	Pathogen is transmitted.	DOID:0050082, A viral infectious disease that [...] is <i>transmitted_by</i> blood transfusion.

Additional relations, defined in the OWL file as ObjectProperty are *inheres_in*, *occurs_with*, *part_of*, *realized_by*, and *realized_by_suppression_with*. These are not used in any textual definition yet.

An example of a definition that combines multiple relations is the textual definition of the term DOID:13021, *AIDS-related cryptococcosis*: “A cryptococcosis that *is_a* disease associated with AIDS *has_material_basis_in* *Cryptococcus neoformans* which *results_in* a systemic infection in individuals with HIV.”

Quantitative Aspects

In the version of the DO documented in the 2012 Nucleic Acid Research paper (Schriml *et al.*, 2012), the ontology represented a knowledge base of 8043 human diseases (divided into inherited, developmental and acquired) (DO version 3, revision 2510). The same version featured textual definitions for 22% of the DO terms. Additionally, 932 logical definitions were provided (in the HumanDO_xp.obo file) to express the annotation of disease attributes (e.g., symptom, phenotype, anatomical or cellular location, and pathogenic agent) linking the DO terms to orthogonal ontologies (such as the Foundational Model of Anatomy (FMA) (Rosse and Mejino, 2003), Human Phenotype Ontology (HPO) (Köhler *et al.*, 2016), NCBI organismal classification

vocabulary (NCBI Resource Coordinators, 2013), PATO (Gkoutos *et al.*, 2005), and Gene Ontology (Ashburner *et al.*, 2000) among others).

The version of the DO documented in the 2015 Nucleic Acid Research paper (Kibbe *et al.*, 2015) resulted after 192 revisions of the ontology OBO version file. This release was marked as the revision 2702 (6 October 2014), and it featured 8803 classes, i.e., diseases (which corresponded to an increment of 760 with respect to the previous NAR publication version), 2384 of which to be considered obsolete. This newer version featured textual definitions for the 32% of DO terms (2087 out of the 6419 non-obsolete ones).

At the time of writing this article, in May 2017, according to the most updated version, the Disease Ontology features 10,831 concepts in total. According to the BioPortal profiling, these are arranged in 12 levels of depth. The maximum number of children for a single class is 107. On average, a class has around four children. There exist 554 classes with only one child, which represents a specialization nomenclature for that class. There are 34 classes that have more than 25 children classes. 7206 classes have no definition.

Mapping with Other Ontologies

The DO addresses the complexity of disease nomenclature by semantically integrating disease and medical vocabularies through extensive cross-mapping of MeSH, ICD-9, ICD-10, NCI's thesaurus (Sioutos *et al.*, 2007), and SNOMED-CT. All terms are extracted from the Unified Medical Language System (UMLS) based on UMLS Concept Unique Identifiers (CUIs). Note that the DO updates vocabulary mappings twice yearly through the extraction of term CUIs from the UMLS vocabulary mapping file. Additionally, the DO also includes concepts extracted directly from Orphanet, OMIM, and the Experimental Factor Ontology (EFO) (Malone *et al.*, 2010). These mappings provide a rich resource for semantically connecting phenotypic, gene and genetic information related to human disease.

Through the mapping process, 91% (7845 out of 8588) of DO terms were mapped to UMLS CUIs (DO version 3, revision 2490, August 2011), a result that represented a 7% reduction of UMLS mappings since the May 2010 DO-UMLS mapping. This reflected DO's increased utilization of logical definitions to define complex disease relationships, which had decreased the number of unique DOIDs. Indeed, note that logical definitions allow to define the relationship between a type of organ disease, such as a *gallbladder adenocarcinoma*, and a cell type, such as the *adenoma*, as a type of adenoma. Moreover, they allow to define the anatomical location of a disease (e.g., *gallbladder adenocarcinoma* is located_in the *gallbladder*).

In the DO version documented in Schriml *et al.* (2012), multiple is_a relationships, which had been inherited from the UMLS vocabularies, were greatly reduced.

As a study on cross-comparison performed in Kibbe *et al.* (2015) shows, the DO provides the largest number of unique cross-references (35,895) among seven different disease vocabularies. This number represents around the 64% of the total unique cross-references in each disease vocabulary, with more than four cross-references on average for each term. Such result highlights the DO's role as a resource rich in cross-references and useful as a disease-centric scaffold for data.

As reported by the interoperability study shown in BioPortal website (Salvadores *et al.*, 2013), as of 15 September 2014, BioPortal mapped the DO classes to classes within 128 other biomedical ontologies also included in its repository. At the time of the writing of this article, DO has increased these mappings of 43 units (for a total of 171 ontologies).

Notable overlapping is observed with OMIM, where 3602 concepts of DO coexist, SNOMED-CT (3062 overlapping terms), and Human Phenotype Ontology (2154 overlapping terms). The highest number of overlapping classes is reached in Medical Dictionary for Regulatory Activities (8996) (Brown *et al.*, 1999), National Cancer Institute Thesaurus (8986), and Cigarette Smoke Exposure Ontology (8234) (Younesi *et al.*, 2014). Conversely, the International Classification of Diseases (Version 10) features 2130 overlapping terms with the DO.

Technological Details

Access Modes

The Disease Ontology can be downloaded in alternative ways. The OBO Foundry portal hosts a version which is updated daily. A GitHub repository and a Sourceforge website host synchronized versions of the ontology. Both OBO and OWL formats are available.

On the Disease Ontology webpage, several services are available for users. These include a Term Tracker to submit new terms, definitions or suggestions, a Wiki documentation, the link to the Twitter, LinkedIn, Facebook, Google+ pages, and other resources.

A curation style guide provides documentation on defining new terms and standardizing their format, as well as on creating definitions, IDs for external references, and URLs for provenance of term definition sources.

Disease ontology files are available under the Creative Commons Public Domain Dedication CC0 1.0 Universal license, which allows copying, modification and distribution of the Disease Ontology content, without asking for permission. They are available in three formats: the OBO formatted Disease Ontology (HumanDO.obo), the Disease Ontology file without cross-references (HumanDO_no_xrefs.obo), and an enhanced Disease Ontology file containing logical definitions to other OBO Foundry ontologies (HumanDO_xp.obo).

The Disease Ontology can also be explored using the Web browsers in EBI's Ontology Lookup Service and NCBO's BioPortal (this also exhibits a service to query the ontology using the SPARQL standard, see 'BioPortal SPARQL queries' in Relevant Websites section).

The DO Web Browser

The DO browser is a Web 2.0 application designed with Semantic Web technologies to allow an easy exploration of the DO, through both a full-text search and a visualization function.

The database

The ontology metadata are stored in Neo4j (Webber, 2012), a NoSQL graph database server. Neo4j is an embedded, disk-based, transactional Java persistence engine that stores data structured in graphs.

Each DO term is modelled as a Neo4j graph node containing the following properties: Name, DOID, Definition, Synonym(s), Alternate ID(s), Subset(s), Cross-Reference(s) and Relationships. The relationships between terms of the ontology are represented as edges of the graph (with a relationship type).

Note that representing graph structures that include multiple relationships is very easy in a graph-based persistence engine such as Neo4j. While a relational database would store ontology terms in one table with a one-to-many connection to another table containing relationships, a graph database stores terms as nodes that are connected to each other by edges (relationships). This paradigm provides several robust mechanisms to retrieve data very fast. Built into Neo4j there are optimized functions for retrieving individual nodes, but also the path between two terms (all paths, the shortest one, user-defined variants), a quite useful feature to make ontology visualization more meaningful.

The rich RESTful API of Neo4j enables advanced users familiar with the Neo4j RESTful API to request data from the Disease Ontology database in a programmatic way and facilitates integration into external projects. The DO browser exploits them to retrieve nodes and their properties (i.e., relationships) via HTTP service calls.

The visualization

The DO browser presents the ontology tree, query results, and terms metadata in a unified point of visualization, where multiple tabs can be selected on a single web page. This page is divided into three areas: the search panel, the navigation panel and the content panel (Fig. 1).

The navigation panel allows visualizing the ontology in a hierarchical structure, i.e., a navigation tree that permits double clicks to expand classes in order to see the related subclasses, and to move from relation to relation.

The search panel supports a basic search (allowing query terms that correspond to DOIDs, names, synonyms, external references, or free text that can be found in definition fields), or an advanced kind of search; this additional feature allows creating targeted and complex queries. Boolean operators are featured to compose conjunctive/disjunctive queries, where the type of data (such as Name, Synonym, DOID, etc.) can be set.

The search for terms is based on concept mapping, with a Lucene (see 'Apache Lucene' in Relevant Websites section) scoring index of matches based on word matches. The Lucene scoring index weighs matches in order (highest to lowest): name, synonym, definition, subset, and ID.

The submitted search returns an output which is visualized in the content panel, in pages of 30 terms from which it is possible to navigate forward or backward. By clicking on a DOID entry belonging to a query result, or from the navigation tree, it is possible to visualize its detailed description (metadata).

A graph interactive view of a selected term can be opened by choosing the functionality *Visualize* from the metadata panel. This creates a new tab that hosts an interactive canvas upon which the target term and any of its children or parents are rendered and can be explored. A red node keeps memory of the first target node of the visualization. Its child or parent nodes have a green label and can be expanded by double clicking on them. Leaf nodes are marked through a grey label and cannot be further expanded. Nodes with more than 5 relatives are compacted into a yellow circle node indicating how many nodes are not being shown. The user can explore these compacted branches by clicking on the yellow circle and selecting to add to the graph only certain children/parents (or all of them). When this choice is made, they are shown in the visualization canvas. The user must be aware that requesting a visualization of more than 50 nodes could slow down the canvas update process.

The DO REST API

A RESTful API service is offered as a programmatic way to access the information found in the database. Metadata (in a richer version compared to the one accessible through the Web browser) can be downloaded in JSON format by using an HTTP request. As of May 2017, the request path can be built by concatenating to the DO's Web page URL (see 'Disease Ontology Web page' in Relevant Websites section) the string `"/api/metadata/"`, followed by the DOID of the term of interest. A query to this URL returns a JSON packet containing all the metadata for this term including parents, children, definition, external references, synonyms, name, alternate IDs and DO identifier. The DO team intends to broaden the types of services provided through API commands in the near future.



The screenshot shows the DO Web interface. At the top, there is a search bar with the text 'atypical autism' and a 'Go' button. To the right of the search bar is a link for 'Advanced Search'. Below the search bar, there is a 'Navigation' panel on the left and a 'Metadata' panel on the right.

Navigation Panel: It shows a hierarchical tree view of the Disease Ontology. The tree starts with 'disease' at the top, followed by 'disease of mental health', 'developmental disorder of mental health', 'pervasive developmental disorder', 'autism spectrum disorder', and finally 'atypical autism' which is highlighted. Other branches include 'disease by infectious agent', 'disease of anatomical entity', 'disease of cellular proliferation', 'adjustment disorder', 'cognitive disorder', 'childhood disintegrative disorder', 'specific developmental disorder', 'dissociative disorder', 'factitious disorder', 'gender identity disorder', 'impulse control disorder', 'personality disorder', 'sexual disorder', 'sleep disorder', 'somatoform disorder', 'substance-related disorder', 'disease of metabolism', 'genetic disease', 'physical disorder', and 'syndrome'.

Metadata Panel: It displays information for the selected term 'atypical autism'. The information is organized into a table-like structure:

Metadata	
DOID	DOID:0060042
Name	atypical autism
Definition	An autism spectrum disorder that involves some autistic symptoms occurring after age 3 with an absence of all the traits necessary for a diagnosis of autism. http://counsellingresource.com/distress/autistic/autism-atypical.html , http://www.wisegeek.com/what-is-atypical-autism.htm , www.neurodevnet.ca
Synonyms	PDD [EXACT]
Relationships	is_a autism spectrum disorder

Below the metadata, there is a button labeled 'Visualize' and a link 'Add an item to the term tracker'.

Fig. 1 DO Web interface with search, navigation and display functions. The disease tree view displays the DO's hierarchical structure and top level parent nodes expandable to view subgraphs. The disease of mental health subgraph with its direct child terms is shown. Term metadata are displayed for the selected term atypical autism from the tree view (the term has been reached by expanding progressively developmental disorder of mental health, then pervasive developmental disorder, finally autism spectrum disorder).

Applications in Other Works

One of the first applications of the DO is documented in Osborne *et al.* (2009), where it was used as a controlled vocabulary to annotate the human genome in terms of diseases. The ontology was used to identify relevant diseases in Gene Reference Into Function (GeneRIF) (Mitchell *et al.*, 2003), with the goal to provide a comprehensive disease-to-gene annotation. This initiative used a preliminary version of the ontology, which at the time corresponded to a manually inspected subset of UMLS and included concepts from outside the UMLS disease/disorder semantic network (i.e., cancers, congenital abnormalities, deformities, and mental disorders). The DO was preferred to OMIM and MeSH for this purpose because it was able to provide larger coverage and more accuracy (its hierarchical structure brought more detailed information about the disease of interest).

The authors argue that the annotation of genes with the DO can enable many graphical and statistical applications (following the model of the Gene Ontology). As an example, (Osborne *et al.*, 2009) showed how visualizing all genes in the human genome with established links to four cancers types led to the identification of eleven genes in common between these four cancers. Such studies facilitate the identification of relevant targets or markers for diseases with common etiology or pathology.

Another use of the Disease Ontology has been described in Du *et al.* (2009), and presents DOLite, a lighter version of the ontology. The work is based on the observation that the DO Directed Acyclic Graph is highly appropriate for organizing a wide spectrum of data, but it is not optimal for specific applications, such as identifying disease-gene relationships. For this specific purpose, a pruned (slimmer) version of the graph is more useful for building a high-level functional summary from a gene list.

Human genes annotations using the DO were very verbose and difficult to interpret, while DOLite allowed a more compact annotation process (larger number of genes assigned to individual disease categories and, conversely, smaller number of disease categories assigned to individual genes).

The DO group manages other software development projects (mainly data access and exploration tools) working in parallel with the DO main project.

The Functional Disease Ontology (FunDO) (Osborne *et al.*, 2009) is a Web application used to measure the internal consistency of the DO, and its ability to functionally annotate a gene list with diseases. The input of FunDO is a list of genes; the output includes relevant diseases based on the statistical analysis of the Disease Ontology annotation database.

GeneAnswers (Feng *et al.*, 2017) is a Bioconductor package that provides an integrative interpretation of genes, examining the relationships between diseases, genes, and biological pathways.

Current Work and Future Goal

At the time of writing this article, the Disease Ontology team is focusing on improving the ontological soundness of the ontology, such as removal of plural terms and of upper case terms.

Work is being dedicated especially to the cancer node and the infectious disease node: restructuring the nodes to avoid redundancy in terms, making terms consistent with the style guidelines, creating textual definitions from other sources such as Wikipedia and the National Cancer Institute, and creating logical definitions that will allow the Disease Ontology to effectively communicate with other ontologies on a computational level. These logical definitions reference the ontologies: FMA, HPO, Cell Line Ontology (CLO) (Sarntivijai *et al.*, 2014), and other OBO Foundry candidate ontologies.

The DO team has chosen to move towards a multi-editor model, utilizing Protégé (Noy *et al.*, 2001) to organize the DO using the Web Ontology Language. This will enable more interoperability with the Human Phenotype Ontology, EBI's Ontology Working Group, Mouse Genome Informatics, and Monarch Initiative, among others. It will also facilitate DO's multiple inferred views through reasoning. The DO team also recognized the imperative need to provide definitions for all DO terms. As a future commitment, the curatorial and development teams will work to add a significant number of definitions to the ontology.

Furthermore, the DO team intends to use the DO paradigm for the creation of non-human organism disease ontologies. The paradigm is expandable: disease defined by the etiology and the affected body system are universally applicable principles for describing pathologies in model organism, plants with viral, bacterial, fungal, or parasitic infections, as well as inherited and acquired disease.

The DO will try and integrate disease prevalence (the proportion of a population that is affected) to augment DO's disease classification.

Finally, an enhanced DO API will allow users to perform any action that they can do interactively on the website via the API.

Use Case

The Disease Ontology diseases can be browsed using a visual graphical tool. Let us suppose that the user is interested in exploring the terms connected to the notion of "heart hypertension". He/she uses the full-text search, but no result is returned. He/she might then search only for the word "heart" and thus identify the entry "DOID:114, *heart disease*". By pressing the functionality "Visualize" on the right top corner of the metadata tab of the DO Web browser, the user can visualize on the canvas the portion of the DO graph that concerns *heart disease* as a central concept (red rectangle); its parent *cardiovascular system disease* is shown (green rectangle) linked through an *is_a* relation (Fig. 2).

Eight children of *heart disease* are shown through an aggregation (yellow circle with the '8' figure). By clicking on the yellow circle, the user can decide which child nodes he/she wishes to visualize. At this level, the entry *Hypertensive heart disease*, originally

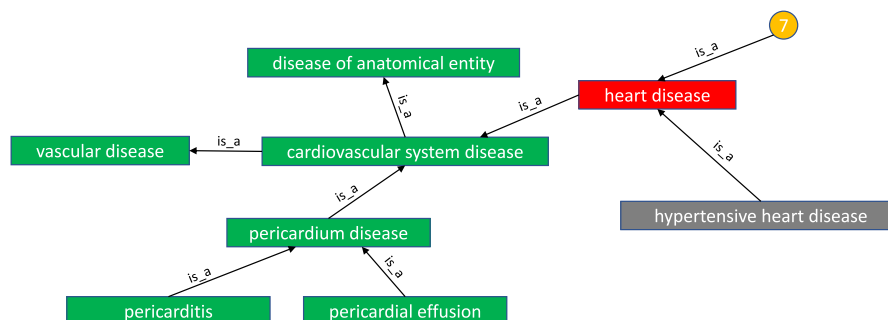


Fig. 2 Term visualization in the DO Web browser, as a result of the full-text search of "heart".


```
{
  "definition": "\"A heart disease that is caused by high blood pressure.\""
  [url:http\\:\\en.wikipedia.org/wiki/Heart_disease,
  url:http\\:\\en.wikipedia.org/wiki/Hypertensive_heart_disease]",
  "xrefs": ["ICD10CM:I11", "ICD10CM:I11.9", "ICD9CM:402", "ICD9CM:402.9",
  "SNOMEDCT_US_2016_03_01:155297007", "SNOMEDCT_US_2016_03_01:194769003",
  "SNOMEDCT_US_2016_03_01:194772005", "SNOMEDCT_US_2016_03_01:64715009",
  "UMLS_CUI:C0152105"],
  "parents": [{"is_a", "heart disease", "DOID:114"}],
  "id": "DOID:11516",
  "name": "hypertensive heart disease"
}
```

Fig. 3 Example response to API request in JSON format.

meant by the user, appears as a possible choice. Let us assume that the user only selects this term of interest. Furthermore, he/she continues to expand the green nodes (ancestors of *heart disease*) to find out more information about the searched disease portion of the ontology. In this way, he/she might obtain, as an example, the excerpt of the graph shown in Fig. 2.

On a different scenario, let us suppose the user already knows the DOID of the disease of interest; in this case, he/she might as well use the DO REST metadata API by constructing an HTTP request in the format indicated in Section The DO REST API, by adding at the end of the URL "DOID:11516" (which identifies *Hypertensive heart disease*), he/she obtains the JSON object shown in Fig. 3.

As already mentioned in Section Current Work and Future Goal, the DO team intends to expand the functionalities that can be accessed through API requests.

See also: Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: Introduction. Characterizing and Functional Assignment of Noncoding RNAs. Data Mining: Clustering. Identification and Extraction of Biomarker Information. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Querying languages and development

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene Ontology: Tool for the unification of biology. *Nature Genetics* 25, 25–29.
- Balikas, G., Krithara, A., Partalas, I., Paliouras, G., 2015. BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. Cham: Springer, pp. 26–39.
- Bodenreider, O., 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research* 32, D267–D270.
- Brown, E.G., Wood, L., Wood, S., 1999. The medical dictionary for regulatory activities (MedDRA). *Drug Safety* 20, 109–117.
- Davis, M.J., Sehgal, M., Ragan, M.A., 2010. Automatic, context-specific generation of Gene Ontology slims. *BMC Bioinformatics* 11, 498.
- Donnelly, K., 2006. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in Health Technology and Informatics* 121, 279–290.
- Du, P., Feng, G., Flatow, J., *et al.*, 2009. From disease ontology to disease-ontology lite: Statistical methods to adapt a general-purpose ontology for the test of gene-ontology associations. *Bioinformatics* 25, 63–68.
- Feng, G., Du, P., Kibbe, W.A., Lin, S., 2017. GeneAnswers, Integrated Interpretation of Genes.
- Gene Ontology Consortium, 2001. Creating the gene ontology resource: Design and implementation. *Genome Research* 11, 1425–1433.
- Gkoutos, G.V., Green, E.C.J., Mallon, A.-M., *et al.*, 2005. Using ontologies to describe mouse phenotypes. *Genome Biology* 6, R8.
- Gramates, L.S., Marygold, S.J., Santos, G. dos, *et al.*, 2017. FlyBase at 25: Looking to the future. *Nucleic Acids Research* 45, D663–D671.
- Hamosh, A., Scott, A.F., Amberger, J.S., *et al.*, 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research* 33, D514–D517.
- Kibbe, W.A., Arze, C., Felix, V., *et al.*, 2015. Disease ontology 2015 update: An expanded and updated database of Human diseases for linking biomedical knowledge through disease data. *Nucleic Acids Research* 43, D1071–D1078.
- Köhler, S., Vasilevsky, N.A., Engelstad, M., *et al.*, 2016. The human phenotype ontology in 2017. *Nucleic Acids Research* 45, gkw1039.
- Lipscomb, C.E., 2000. Medical subject headings (MeSH). *Bulletin of the Medical Library Association* 88, 265–266.
- Malone, J., Holloway, E., Adamusiak, T., *et al.*, 2010. Modeling sample variables with an experimental factor ontology. *Bioinformatics* 26, 1112–1118.
- Mitchell, J.A., Aronson, A.R., Mork, J.G., *et al.*, 2003. Gene indexing: Characterization and analysis of NLM's GeneRIFs. In: AMIA, Annual Symposium Proceedings. AMIA Symposium 2003, pp. 460–464.
- NCBI Resource Coordinators, 2013. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 41, D8–D20.
- Noy, N.F., Shah, N.H., Whetzel, P.L., *et al.*, 2009. BioPortal: Ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research* 37, W170–W173.
- Noy, N.F., Sintek, M., Decker, S., *et al.*, 2001. Creating semantic web contents with protege-2000. *IEEE Intelligent Systems* 16, 60–71.
- Osborne, J.D., Flatow, J., Holko, M., *et al.*, 2009. Annotating the human genome with Disease Ontology. *BMC Genomics* 10 (Suppl 1), S6.
- Piñero, J., Bravo, Á., Queralt-Rosinach, N., *et al.*, 2017. DisGeNET: A comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research* 45, D833–D839.
- Portales-Casamar, E., Ch'ng, C., Lui, F., *et al.*, 2013. Neurocarta: Aggregating and sharing disease-gene relations for the neurosciences. *BMC Genomics* 14, 129.
- Rath, A., Olry, A., Dhombres, F., *et al.*, 2012. Representation of rare diseases in health information systems: The orphanet approach to serve a wide range of end users. *Human Mutation* 33, 803–808.
- Rosse, C., Mejino, J.L.V., 2003. A reference ontology for biomedical informatics: The Foundational Model of Anatomy. *Journal of Biomedical Informatics* 36, 478–500.
- Salvadores, M., Alexander, P.R., Musen, M.A., Noy, N.F., 2013. BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF. *Semantic Web* 4, 277–284.
- Santivijai, S., Lin, Y., Xiang, Z., *et al.*, 2014. CLO: The cell line ontology. *Journal of Biomedical Semantics* 5, 37.
- Scheuermann, R.H., Ceusters, W., Smith, B., 2009. Toward an ontological treatment of disease and diagnosis. *Summit on Translational Bioinformatics* 2009, 116–120.

- Schriml, L.M., Arze, C., Nadendla, S., *et al.*, 2012. Disease ontology: A backbone for disease semantic integration. *Nucleic Acids Research* 40, 940–946.
- Silva, F.A.B., Silva, M.J., Couto, F.M., 2010. Epidemic marketplace: An e-science platform for epidemic modelling and analysis. *ERCIM News* 82, 43–44.
- Sioutos, N., Coronado, S., de Haber, M.W., *et al.*, 2007. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics* 40, 30–43.
- Webber, J., 2012. A programmatic introduction to Neo4j. In: *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity – SPLASH '12*. ACM Press: New York, NY, p. 217.
- Younesi, E., Ansari, S., Guendel, M., *et al.*, 2014. CSEO – The cigarette smoke exposure ontology. *Journal of Biomedical Semantics* 5, 31.
- Zoubarev, A., Hamer, K.M., Keshav, K.D., *et al.*, 2012. Gemma: A resource for the reuse, sharing and meta-analysis of expression profiling data. *Bioinformatics* 28, 2272–2273.

Further Reading

- Hoehndorf, R., Schofield, P.N., Gkoutos, G.V., 2015. Analysis of the human diseasesome using phenotype similarity between common, genetic, and infectious diseases. *Scientific Reports* 5, 10888.
- LePendu, P., Musen, M.A., Shah, N.H., 2011. Enabling enrichment analysis with the Human Disease Ontology. *Journal of Biomedical Informatics* 44, S31–S38.
- Li, J., Gong, B., Chen, X., *et al.*, 2011. DOSim: An R package for similarity between diseases based on Disease Ontology. *BMC Bioinformatics* 12, 266.
- Schriml, L.M., Mittraka, E., 2015. The Disease Ontology: Fostering interoperability between biological and clinical human disease-related data. *Mammalian Genome* 26, 584–589.
- Yu, G., Wang, L.G., Yan, G.R., He, Q.Y., 2015. DOSE: An R/Bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics* 31, 608–609.

Relevant Websites

- <http://lucene.apache.org/java/docs/index.html/>
Apache Lucene.
- <https://bioportal.bioontology.org/ontologies/D0ID/>
BioPortal.
- <http://sparql.bioontology.org/>
BioPortal SPARQL queries.
- http://do-wiki.nubic.northwestern.edu/do-wiki/index.php/Style_Guide/
DO Curation Style Guide.
- <http://disease-ontology.org/>
Disease Ontology Web page.
- http://do-wiki.nubic.northwestern.edu/do-wiki/index.php/Main_Page/
DO Wiki Page.
- <https://www.ebi.ac.uk/ols/ontologies/doid/>
EBI.
- <https://github.com/DiseaseOntology/HumanDiseaseOntology/>
Github repository.
- <http://www.who.int/classifications/icd/>
ICD, World Health Organization.
- <https://www.rarediseases.org/>
NORD.

Biographical Sketch



Anna Bernasconi received her Master's Degree in Computer Engineering in 2015 from Politecnico di Milano and a Master in Computer Science at the University of Illinois at Chicago in 2016. She is currently a PhD candidate at Politecnico di Milano, working in the field of data-driven genomic computing. Her research interests are in the area of bioinformatics data and metadata integration in order to support answering to complex biological queries. She is currently focusing on public repositories of open data, controlled biomedical vocabularies and ontologies.



Marco Masseroli received the Laurea degree in Electronic Engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in Biomedical Engineering in 1996, from the Universidad de Granada, Spain. He is associate professor in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano, and lecturer of Bioinformatics and Biomedical Informatics. He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomía Patológica, Facultad de Medicina at the Universidad de Granada, Spain, and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health, Bethesda. His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Biological and Medical Ontologies: Human Phenotype Ontology (HPO)

Anna Bernasconi, Politecnico di Milano, Milan, Italy

Marco Masseroli, Polytechnic University of Milan, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The Human Phenotype Ontology (HPO) is an open source ontology developed collaboratively, that provides a standardized, structured, and controlled vocabulary of all phenotypic features that are encountered in human hereditary and other diseases, with focus on monogenic ones.

The initial goal of the HPO has been to provide researchers and clinicians that operate in the human genetics field with a well-defined, shared, ontological framework. This allows to both share clinical data and knowledge in a computer-readable standardized way and to apply computational analysis algorithms to explore the human phenotype.

Phenotypical abnormalities are meant to serve as a bridge that translates genome-scale biology concepts into disease-centered human biology concepts. Understanding human health and disease requires a systematic human and machine readable representation of the current knowledge on phenotypic variation. Since an ontology provides a conceptualization of the knowledge of a domain, the HPO can be used effectively in such a context.

To understand the use of the ontology in supporting computational analysis, let us suppose that different experiments are tagged with synonymous descriptions such as *generalized amyotrophy*, *generalized muscular atrophy*, and *muscular atrophy, generalized*. While a human user would have no uncertainties in attributing all these terms to the same concept, a regular programmatic search function would. This is a case in which the automatic attribution of all these terms to a unique standardized ontological term becomes necessary. Indeed, an ontology that captures phenotypic information allows the use of computational algorithms that exploit semantic similarity between related phenotypic abnormalities, in correlation with other cellular phenomena associated with human disease.

In short, relevant uses of this ontology include: 1. providing a standardized vocabulary for clinical databases, 2. mapping between model organism and human phenotypes, 3. Bioinformatics research on the relationships between cellular/biochemical networks and human phenotypic abnormalities, 4. clinical diagnostics in human genetics.

All terms of the HPO have a unique stable identifier (e.g. HP:0000478), along with a label and a set of synonyms. Most terms also feature a textual definition manually curated by clinical experts. Moreover, HPO terms contain external references to other resources, aiming at an improved interoperability among different research areas in the biomedical field.

In its last release (Köhler *et al.*, 2016), the HPO is organized in five independent sub-ontologies that represent different aspects: *Mode of inheritance*, *Mortality/Aging*, *Clinical modifier*, *Frequency*, and the largest category of *Phenotypic abnormalities*.

When the first version of the HPO was developed (Robinson *et al.*, 2008), the most important source of information in the field of hereditary diseases was the database Online Mendelian Inheritance in Man (OMIM) (Hamosh *et al.*, 2005), which represented an essential result for the human genetics community. The HPO was therefore based on OMIM's terminology. However, OMIM did not use a controlled vocabulary to describe phenotypic features (signs and symptoms). This shortly made OMIM a wrong candidate for computational analysis. Note that, even if the OMIM entries were grouped by organ system, its data structure did not reflect that, for example, *Hypoplastic philtrum* and *Smooth philtrum* were more closely related to one another than to *Hypoplastic nasal septum* (all these descriptions are in the NOSE category of OMIM's clinical synopsis).

The creation of the HPO has thus answered to the need for a uniform and internationally accepted set of terms describing the human phenotype, and to the need for resources that would allow computational analyses on human phenotype data. With respect to the former one, it is noted that the terms used by clinicians to describe phenotypic expression have evolved in disorganized ways, which has resulted in the absence of databases of systematically collected phenotypic information about humans with hereditary disease. With respect to the latter one, it is clear intuitively that certain hereditary disorders are phenotypically similar to each other because of shared phenotypic features. However, these kinds of inference would have been impossible without a proper way to correlate the phenotype to features of genetic networks on genome-wide scale.

The HPO was born with the purpose to overcome previous efforts on computational analysis of phenotypic data in human hereditary disease. These works have used text mining techniques, but they all used concepts from thesauri (such as Unified Medical Language System (UMLS) (Bodenreider, 2004) or Medical Subject Headings (MeSH) (Lipscomb, 2000)) whose indexing terms are not designed for describing hereditary diseases and their related phenotypes.

Compared to previous works, HPO authors contend that a manually curated ontological approach to computational phenotype analysis has its own advantages. To mention one, in HPO terms are based on medical knowledge instead of text-mining inferences. These can be refined and extended in the future.

Note that integrative analysis like the one performed by the molecular biology community following the successful model proposed by the Gene Ontology (Ashburner *et al.*, 2000), was not possible in the human genetics community until the contribute of the HPO. Just as Gene Ontology, which is keeping expanding today, the HPO continuously undergoes active improvement. It is

currently being developed within the context of the Monarch Initiative (McMurry *et al.*, 2016), a collaboration between various research institutions supported by the National Institute of Health, which directs its efforts towards the integration of genotype-phenotype data from many species and sources. Moreover, the HPO participates in the OBO (Open Biological Biomedical Ontologies) Foundry project (Smith *et al.*, 2007), which holds an updated version of the ontology file on its server, available for download.

Furthermore, it is worth mentioning that the HPO project has been completely incorporated into the UMLS and is partnering with Orphanet (Rath *et al.*, 2012), a European Consortium of 40 countries providing a unique resource that gathers and improves knowledge on rare diseases. Their disease terms are being re-annotated with HPO terms.

The ontology is exposed as an open source resource in several platforms such as BioPortal (Noy *et al.*, 2009), the OBO Foundry (Smith *et al.*, 2007), MSeqDR browser (Falk *et al.*, 2015), the Ontology Lookup Service (Cote *et al.*, 2008), Ontobee (Ong *et al.*, 2017), and the Monarch Initiative, that provide intuitive Web-tools to browse its content.

The following of this chapter is organized as follows: Section “Development” defines the historical development of the ontology since its first version, born in 2007, until its release at the time of writing this chapter, in 2017. Section “The Human Phenotype Ontology” describes the core part of the HPO project, the structure of the ontology. Section “Phenotype Annotation Data” presents its main employment, the annotations of diseases. Section “Computable Definitions Of HPO Terms” includes relevant information on computer readable definitions of the ontology terms. Section “Quantitative Aspects” gives a quantitative perspective on the coverage and mappings provided by the HP ontology in its different releases. Section “Technological Details” overviews the technological aspects regarding its access. Section “Applications In Other Works” presents a list of notable applications that employ the HPO in their work. Finally, Section “Use Case: The Phenomizer” concludes with a sample use case scenario describing a simple activity of a user taking advantage of the HPO.

Development

The story of the HPO is made of three milestones, along with important literature material. This section proposes a brief overview of the phases that have brought the ontology from its birth to how it is today.

Robinson *et al.* (2008) and Robinson and Mundlos (2010) document the first release and use of the Human Phenotype Ontology. The initial intent was to enable the integration of phenotype information across scientific fields and databases. That version of the HPO made possible to annotate and analyze hereditary disease, using computational assertions stating that a disease is associated with a given phenotypic abnormality.

In 2014 the HPO upgraded its relevance, becoming a tool that allowed more interoperability, being utilized as a world-wide reference for describing phenotype information in various datasets (Köhler *et al.*, 2014a). At that stage, the ontology gained the dignity of a multi-dimension project (which still is), for linking molecular biology and disease through phenotype data. The project provided a richer standard phenotype terminology and collection of annotations of diseases with phenotypes. The project had clearly grown in terms of coverage, complexity, usage, and cross-linking with other projects. The 2014 version of the ontology was organized in three initial independent sub-ontologies which covered different categories. The largest is the *phenotypic abnormalities* sub-ontology, followed by the *mode of inheritance* and the *onset and clinical course* sub-ontologies. With respect to previous versions, the HPO at this point featured detailed textual definitions for 65% of the terms (created by clinical experts) and cross-references to other ontologies for 39% of the terms. In addition, to improve semantic interoperability with other ontologies in the OBO Foundry, the HPO project had started since 2009 to create logical definitions for each term. These are formal descriptions that are computer processable, useful for automated logical inference and reasoning. Finally, this release of the ontology also provided a number of negative annotations (NOT-modifier) denoting which clinical features have *not* been observed in a patient with a certain disease.

The release described in Köhler *et al.* (2016) has further reviewed the progress of the HPO project by adding specific areas of interest (e.g., common diseases), new algorithms for genomic discovery driven by phenotype information, and mapping with cross-species ontologies (e.g., Mammalian Phenotype Ontology (Smith *et al.*, 2005)). Also the quality control pipeline has been improved and user-friendly terminology has been added.

At the beginning of 2017 continues improving. As of May 2017, the BioPortal page lists 10 projects using this ontology: National Data Bank for Rare Diseases (BNDMR), the Clinical Genomicist Workstation (Sharma *et al.*, 2013), DisGeNET-RDF (Piñero *et al.*, 2017), Epidemic Marketplace (Silva *et al.*, 2010), Gemma (Zoubarev *et al.*, 2012), International Fanconi Anemia Registry (Kutler *et al.*, 2003), Linking Open Data for Rare Diseases (Choquet *et al.*, 2015), Neurocarta (Portales-Casamar *et al.*, 2013), SKELETOME (Groza *et al.*, 2012), and the Monarch Initiative.

The Human Phenotype Ontology

The Human Phenotype Ontology is arranged as a directed acyclic graph (DAG). The *terms* (or *concepts*) of the ontology are here interchangeably referred to as *classes*. Each term describes a phenotypic abnormality, while the ontology does not describe diseases themselves, but the abnormalities connected to them - therefore to be explored. Each term is related to its parent term through an “is a” relationship, meaning that it represents a *subclass* (i.e., more specific instance) of a more general parent term. Since a term in a DAG can have multiple parents, then a phenotypic feature can be considered as a more specific aspect of one or more parental

terms. As an example, from [Fig. 1](#): *Atrial septal defect* is a term which is more specific than both *Abnormality of cardiac atrium* and *Abnormality of the atrial septum*.

The HPO was first constructed with OBO-Edit ([Day-Richter et al., 2007](#)), an ontology editor for biologists. As a start, concepts and their representative terms were extracted by parsing the textual descriptions provided in omim.txt, a file version of the OMIM database.

Since OMIM does not use a controlled vocabulary, the terms were then curated manually, taking advantage of domain knowledge in human genetics (especially for the task of fusing synonyms into a single term). As an example, the three OMIM descriptions *Carpal bone hypoplasia*, *Hypoplasia of carpal bones*, and *Hypoplastic carpal bones* were fused into the single term *HP:0001498, Carpal bone hypoplasia*. Finally, additional descriptions were mapped using an adapted version of the Smith-Waterman algorithm of identification of common molecular subsequences ([Smith and Waterman, 1981](#)), then examined manually.

Such ontological structure for expressing information of the phenotype has a relevant advantage: search procedures can exploit the semantic relationships between terms.

Organization

Note that the majority of HPO classes describe abnormalities, but additional sub-ontologies are included to describe other aspects. At the time of writing this chapter, the HPO is divided into five independent sub-ontologies, which cover the following contents:

1. The Phenotypic abnormality sub-ontology is used to describe organ abnormalities and contains most terms; examples of terms at the first level are *Abnormality of the musculoskeletal system*, or *Hematological abnormality*;
2. The Mode of inheritance sub-ontology defines disease models according to Mendelian/non-Mendelian inheritance modes and contains terms like *Autosomal dominant*;
3. The Mortality/Aging sub-ontology expresses the age of death associated with a disease or observed in a specific individual to be annotated; example classes are *Neonatal death*, or *Sudden death*;
4. The Clinical modifier sub-ontology provides terms to characterize and specify the phenotypic abnormalities defined in the Phenotypic abnormality sub-ontology, with respect to various aspects including the speed of progression, the variability, or the onset; example classes include: *Onset in childhood*, *Rapidly progressive*, or *Incomplete penetrance*;
5. The Frequency ontology indicates the frequency of clinical feature display from patients, such as *Obligate*, *Frequent*, or *Occasional* (as identically defined in Orphanet).

Phenotype Annotation Data

The HPO is used for the annotation of databases such as OMIM, which is a catalogue of human genes and genetic disorders, Orphanet, a knowledge base about rare diseases, and DECIPHER ([Bragin et al., 2014](#)), an RNAi screening project.

Diseases are annotated to terms of the HPO, which are used to comprehensively describe signs, symptoms, and phenotypic manifestations that characterize the referred disease.

The HPO does not describe disease entities, but only the phenotypic abnormalities associated with them. To annotate all entries of OMIM that feature a clinical-synopsis section (nearly 6000 out of 8500), the most specific possible terms of the ontology have been used.

The true-path rule ([Gene Ontology Consortium, 2001](#)) applies to the terms of the HPO, i.e., a disease that is directly annotated to a specific term is also annotated (implicitly) to all its ancestors. As an example, *LARSEN SYNDROME* (OMIM:150250) is directly annotated to *Atrial septal defect* and is thereby implicitly annotated to *Abnormality of the atrial septum*, and to all its other ancestors' terms shown in [Fig. 1](#). Such structure of the HPO allows flexible searches for disease entries based on phenotypic abnormalities, that can be specified with different granularity focus (from broad to narrow).

The HPO project uses an ontological similarity measure to define a significant phenotypic similarity metric among hereditary diseases listed in OMIM. Following the example of the Gene Ontology, in the HPO terms that are closer to the root are more general than terms further away from the root (in [Fig. 1](#), *Atrial septal defect* is more specific than *Abnormal heart morphology*).

The information content of a HPO term is estimated using its frequency among the annotations of the entire OMIM corpus. The higher a term is located in the ontology, the lower information content it holds. For example, when two diseases such as *FIBROCHONDROGENESIS 1* and *TRANSALDOLASE DEFICIENCY* are both annotated to a term with high information content like *Patent foramen ovale*, then the degree of similarity calculated between these diseases is higher than the degree of similarity calculated between *FIBROCHONDROGENESIS 1* and *HYPERCALCEMIA, INFANTILE*, since these are annotated to the more general term *Abnormal heart morphology* (with lower information content). For a term t of the ontology, given its probability p (which is retrieved as its frequency among annotations to all the annotated diseases), the information content is equal to $(-\ln p(t))$.

A measure of similarity which is comparable to the one between two proteins, expressed in [Pesquita et al. \(2008\)](#), was applied on the diseases listed in the OMIM database.

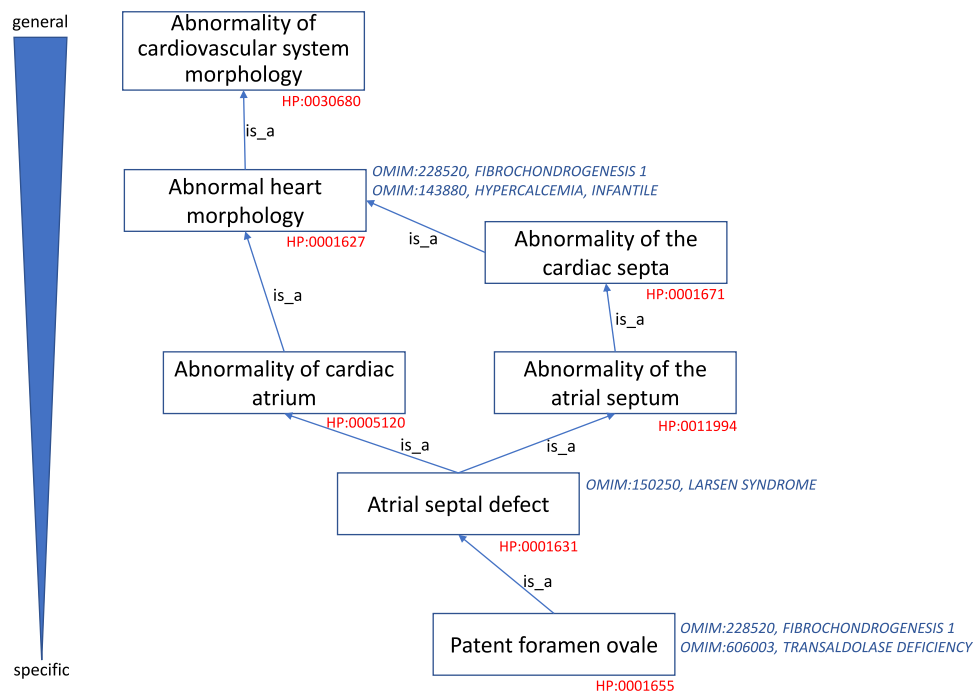


Fig. 1 Example excerpt of the HPO structure regarding the Patent foramen ovale specific concept and all its associated general concepts in the HPO, together with the OMIM diseases annotated to them.

An ontology like the HPO is not intended to capture quantitative aspects such as the specific height of an adult body or the blood glucose level in mg/dl. Instead, its terms often express qualitative information regarding the excess or reduction in quantity of the entity in question (e.g., *Tall stature*, or *Hypoglycemia*). In some cases, it has been considered useful for clinical purposes to divide entities in multiple categories. For this purpose, a set of meta-annotations has been included as a support to HPO phenotype annotations. These include:

- Qualifier/Modifier. As an example, the degree of Intellectual disability is reported in six possible categories (i.e., Mild, Moderate, Severe, Profound, Borderline, or Progressive);
- Evidence Code (i.e., inferred by text mining, from published clinical study, individual clinical experience, electronical annotation, traceable author statement);
- Onset modifier (i.e., any term from the HPO sub-ontology Clinical modifier such as Ameliorated by, Triggered by, Aggravated by, etc.);
- Frequency modifier (i.e., any term from the HPO sub-ontology Frequency, such as percentage values, ratios, very rare, rare, occasional, frequent, typical, etc.).

Applications of the Annotations

A study from [Goh et al. \(2007\)](#), performed on 727 diseases (divided among 21 physiological-disorder classes), has led to an interesting visualization of the human phenome, a clustering of diseases based on their phenotypic similarities. The similarity of diseases is shown on its phenotypic level by linking all pairs of diseases exceeding a similarity score threshold. To better visualize these relationships, a graph representation has been used. The graph features a node for every disease and the distance between nodes expresses their similarity measure. A specific algorithm was used to show the clustered structure of the phenotypic network.

The study has demonstrated that, although diseases are generally independent of the disorder classes, the resulting phenotypic network displays clusters corresponding to many of these classes.

This representation additionally allows to visualize interconnections between one disease cluster and the other (e.g., muscular disorders are strongly connected to neurological disorders) ([Fig. 2](#)).

Another interesting application of HPO terms annotations was performed in a clinical setting. In medical genetics, clinical diagnosis is often pursued by combining multiple features. Automatized procedures must consider various dimensions: the number of hereditary disorders, the multitude of partially overlapping clinical features associated with them, and the different ways and levels of detail that clinicians may use to express clinical features.

Taking all these aspects into consideration, the aim is for diagnostic algorithms that are completely automatized, able to search medical diagnosis at varying levels of detail, weighting specific features more than general ones, and not very sensitive to the

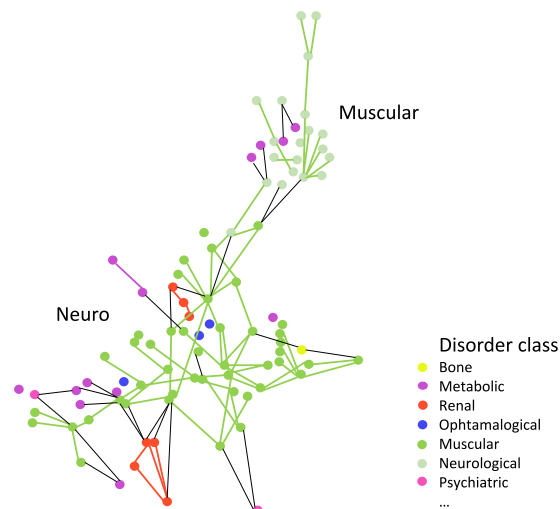


Fig. 2 Visualization of an excerpt of the human phenome. Each disease listed in OMIM is shown as a node in the graph. The colors are assigned based on the membership to a disorder classes. The nodes are clustered in a phenotypic network. Connections between nodes are shown starting from a certain similarity score.

absence of individual features. An ontological approach appears appropriate in this context. Experiments on a set of 2116 OMIM diseases have shown in [Robinson et al. \(2008\)](#) that the HPO is able to capture phenotypic similarity at various levels of detail and that its calculation is not overly sensitive to the set of initial terms chosen for the search of the diagnosis. Given a set of features and asking for a possible diagnosis, in the optimal case the algorithm assigns a top rank to the original disease or at least place it in the first positions.

Computable Definitions of HPO Terms

An HPO term is explicitly related to other classes of the ontology through hierarchical relationships. However, no information is encoded about its relationships with concepts from cellular physiology, biochemistry, pathology, histology, and anatomy. For example, there is no link between a concept such as *Cerebral calcification* and the anatomical site *human brain*. This connection needs to be instructed through a computer-readable explanation. This is given through logical definitions created using the Phenotypic Quality Ontology (PATO) ([Gkoutos et al., 2005](#)). The basic methodology, described in [Köhler et al. \(2013\)](#), states that classes in the HPO are logically equivalent to descriptions based on Qualities (i.e., types of characteristic that the genotype affects), Entities (types of entities that bear the quality), and Modifiers. An example of such definition (in Manchester syntax) follows:

```
Class: Hypoglycemia
EquivalentTo: 'decreased concentration' (PATO: 0001163)
and towards some 'glucose' (ChEBI: 17234)
and inheres_in some 'portion of blood' (UBERON: 0000178)
and qualifier some 'abnormal' (PATO:0000460)
```

This statement means that the HPO term *Hypoglycemia* is defined as the intersection of all classes that are “a concentration which is lower relative to the normal or average”, “deviate from the normal or average”, with respect to (towards) glucose and inhering in “blood”.

This definition uses terms from multiple ontologies, including PATO for describing qualities, UBERON ([Mungall et al., 2012](#)) for describing anatomy, and ChEBI ([Hastings et al., 2013](#)) for describing small biological molecules. Definitions in this form can be used in various applications, such as automation of ontology construction, computational quality control, and integrative cross-species phenotype comparisons.

Quantitative Aspects

Literature contributions report precise quantitative measures regarding the coverage of the HPO ontology terms and of its annotations.

In [Robinson and Mundlos \(2010\)](#) the HPO contained over 9500 terms describing phenotypic features. On the website, they also provided about 50,000 annotations to HPO terms for 4779 diseases. This was a first step, which was immediately refined and expanded (including new terms definitions for concepts not originally found in the OMIM data).

The first Nucleic Acids Research database article regarding the HPO (Köhler *et al.*, 2014a) referred that the HPO included 10,088 terms and 13,326 relations between them. Additionally, the ontology provided textual definitions for 6603 terms and 6265 synonyms. Logical definitions were developed for 46% (4591) of all HPO classes using terms from other ontologies regarding anatomy, cell types, function, embryology, pathology and other domains. Also, the annotation database was updated to include 110,301 annotations of 7354 human hereditary syndromes present in OMIM, Orphanet and DECIPHER to classes of the HPO. On average, each disease entry had 15 HPO annotations.

The HPO has shown a substantial growth in the September 2016 release (documented in Köhler *et al.* (2016)). 11,813 terms have been included, linked through 15,595 relationships. The textual definitions have reached 8627 units, and the synonyms are 14,328. Logical definitions are included for 5717 HPO classes, referring to additional ontologies for biochemistry, gene function, and others.

As a novelty, also annotations of HPO terms to common disease have been added (until 2014, the focus of the HPO had been on rare disease). In this release, there are 123,724 annotations of HPO terms to rare diseases and 132,620 to common diseases.

At the time of writing of this chapter, in May 2017, according to the most updated version, the HP ontology has 12,358 concepts in total. According to the BioPortal profiling, the maximum number of children for a single term is 32. On average, a term has around three children. There exist 1037 terms with only one child, which represents a specialization nomenclature for that term. There are 22 terms with more than 25 children. 6702 classes have no definition, while the other 9102 present a definition curated by a clinical expert.

Mapping with Other Ontologies

The HPO has mappings to 185 other ontologies (as reported by the interoperability study shown in BioPortal webpage (Salvadores *et al.*, 2013)). The biggest overlapping is reported with OMIM, where 3608 concepts of HPO coexist, followed by the Suggested Ontology for Pharmacogenomics (Coulet *et al.*, 2006) (2633 overlapping concepts) and the Human Disease Ontology (2154).

As to other biomedical ontologies of broad interest, we mention that the HPO has 709 overlapping concepts with the Mammalian Phenotype Ontology (Smith *et al.*, 2005), 423 with the Mouse Pathology Ontology (Sundberg *et al.*, 2008), and 12 with the Foundational Model of Anatomy (Rosse and Mejino, 2003).

Köhler *et al.* (2014a) points out that 39% (3956) of the HPO terms contain cross-references, whose 98% points to Unified Medical Language System and Medical Subject Headings. Above all, these references are helpful for linking to the Disease Ontology resource.

Since the 2014 release, equivalence mappings to other phenotype vocabularies have been made available. These include the London Dysmorphology Database (LDDb) (Guest *et al.*, 1999) – which provides a direct conversion from LDDb phenotypic data into HPO terms, Orphanet – which is annotating all its diseases with HPO terms, MedDRA (Brown *et al.*, 1999), and phenoDB (Hamosh *et al.*, 2013).

These can be retrieved on the GitHub repository of the HPO in csv, and tsv formats.

The OBO-version files of the ontology contain HPO terms annotated with cross-references to UMLS, SNOMED CT (Donnelly, 2006) and MeSH concepts (e.g., “xref: SNOMEDCT_US:32659003”, or “xref: UMLS:C0266295”).

Coverage

Until 2013 not much attention has been dedicated to the terminological characteristics of the HPO and to the representation of phenotypes in standard terminologies. In Winnenburger and Bodenreider (2014) the authors have evaluated the 2014 release of the HPO against the UMLS and its source vocabularies, including SNOMED CT and MeSH, observing that the coverage of the HPO phenotype concepts in the UMLS is 54% and only 30% in SNOMED CT.

Technological Details

HPO Workflow and Resources

Hudson, a system for continuous integration, is used for the management of stable releases of the data of the HP ontology (in addition to other related metadata). The major focus is on the phenotype ontology and the annotation data, but closely related projects are included too.

The files kept available through Hudson include the OBO and OWL formats of the ontology, the logical definitions of HPO terms, mappings to other phenotype vocabularies (e.g., Orphanet, LDDb, MedDRA), manual and semi-automatic annotations of syndromes from OMIM, DECIPHER, and Orphanet, and disease-HPO term associations asserted not to be associated with the corresponding disease (negative annotation: NOT-modifier).

In addition, the system offers access to: the mapping of human genes to phenotypic features and vice versa, a precomputed disease-disease similarity matrix for all diseases with annotations to HPO terms, and a cross-species phenotype ontology (human, mouse, zebrafish).

Access Mode

The HPO data content can be currently accessed through flat files (with OBO or OWL extensions). To download the most recent stable releases, the users should use persistent URLs that redirect to the continuous integration systems Hudson and Jenkins.

Other methods to access the HPO information include a MySQL database (which is a dump of all HPO related data) and Web-based tools (among others, the HPO features a browser with free text search functionality on its main Web page).

Choosing concepts through the Web interface leads to an individual page for each HPO term. Alternatively, the same page is accessible by adding an id parameter to the home page URL of the HPO (e.g., id=HP:0000127 shows the *Renal salt wasting* page).

Each page presents the term label, its synonyms, the textual and logical definitions, a list of super- and sub-classes, lists of associated diseases and genes (exportable to Excel or CSV format), and the associated graph view.

The last stable release files of the ontology can be also downloaded through the BioPortal page (in the formats OBO, CSV, and XRDF) and the OBO Foundry portal (OBO and OWL formats).

Furthermore, the Monarch Initiative provides a useful integrated interface in which each phenotype trait is directly shown as linked to the diseases that feature it, the genes that characterize it, the related genotypes, and the liked variants.

In addition to simple browsing, the HPO official Web page allows requesting new classes, suggesting structural changes of the HPO hierarchy (through the HPO tracker), accessing the GitHub repository, and reading the documentation.

Applications in Other Works

Apart from being a resource for computational analysis of human disease phenotypes, the HPO has been used as a basis for a wide range of studies and tools that perform analysis in both research and clinical settings.

One of the main application fields of the HPO corresponds to the development of algorithms and computational tools for phenotype-driven differential diagnosis.

The *Phenomizer* is a Web-based application used for clinical diagnostics which uses searches in ontologies based on semantic similarity. Köhler *et al.* (2009) presents the initial version of the tool, which has now been updated to the latest HPO.

In April 2017, the HPO website has also made available a new beta-prototype called *Orphamizer*, which uses a different algorithm (the Bayesian Ontology Query Algorithm – BOQA (Bauer *et al.*, 2012)), a different set of diseases and phenotype-associations.

In the context of phenotype-driven exome and genome analysis, novel tools that provide results of Next Generation Sequencing results include *PhenIX* (Zemojtel *et al.*, 2014), which provides effective diagnosis of genetic disease by computational phenotype analysis of the disease-associated genome, *Exomiser* (Robinson *et al.*, 2014), which uses a set of phenotypes encoded using the HPO, and *Phevor* (Singleton *et al.*, 2014), which combines biomedical ontologies to identify disease-causing alleles.

Cross-species phenotype analysis include *MouseFinder* (Chen *et al.*, 2012), *PhenoDigm* (Smedley *et al.*, 2013), *PhenomeNET* (Hoehndorf *et al.*, 2011), as well as *Uberpheno* (Köhler *et al.*, 2013). These enable searches for novel disease genes based on the analysis of model-organism phenotypes. *PhenogramViz* (Köhler *et al.*, 2014b) allows to explore in a visual fashion gene-to-phenotype relations in copy number variants (CNVs).

Clinical data management and analysis tools include *ECARUCA* (Vulto-van Silfhout *et al.*, 2013), *DECIPHER* (Bragin *et al.*, 2014), and *PhenoTips* (Girdea *et al.*, 2013).

As a method for Functional and network analysis (inferring novel drug indications) we mention *PREDICT* (Gottlieb *et al.*, 2011).

Use Case: The Phenomizer

An indirect use of the ontology can be made through the *Phenomizer*, described in the previous section. As an example of how the HPO has been designed to be used in practical cases, this section reports a procedure for using an ontological semantic similarity analysis in clinical diagnosis.

A search procedure has been designed to take phenotype features as input and provide possible disease (diagnosis) as output.

The semantic structure of the HPO is used to weight the importance of the input phenotype features and disease terms according to their clinical specificity. Intuitively, when a physician enters the query term *downward slanting palpebral fissures*, the amount of clinical information about the patient is higher than when the physician enters the term *abnormality of the eyelid*.

The designed search procedure takes this fact into account by weighting the best match between a query term to the terms of any given disease according to the information content of that term. A statistical model assigns a p-value to the results of each search. Not only a ranking, but also a significance threshold for search results is provided. This distinguishes this search procedure from other search routines commonly used in clinical genetics that are designed to show all diseases that are characterized by a certain number of query terms.

In a practical scenario, the user enters the HPO term in the *Features* window. Let us suppose he/she selects various phenotypic features of the disease “ADACTYLIA, UNILATERAL” (e.g., *Brachydactyly syndrome*, *Clinodactyly of the 5th finger*, *Cone-shaped epiphyses of the middle phalanges of the hand*, *Polydactyly*, *Short 1st metacarpal*, *Short 2nd finger*, *Short 3rd finger*, *Abnormality of the fingernails*). Each feature may be marked as mandatory if the clinician is certain of its presence and especially of its relation with the underlying disease. Logically, when a feature is marked as mandatory, all the diseases that are not annotated to it are removed from the reasoning base.

Suppose the user marks as mandatory the features “Short 2nd finger” and “Short 3rd finger”.

When the *Phenomizer* is initiated, a list of differential diagnoses ranked according to p-value is shown.

The list of the first 10 candidates indeed includes “ADACTYLIA, UNILATERAL” as foreseen, with the same 0.0020 p-value as “CAMPTOBRAHYDACTYLY”, “BRACHYDACTYLY TYPE A3” and other 5 diseases, along with two other entries with 0.0032 p-value.

Such output returned by the *Phenomizer* can prompt more appropriate clinical examination or technical examinations with respect to the one given by the clinician in a non-automatized way.

See also: Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: Introduction. Data Mining: Clustering. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Querying languages and development

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25, 25–29.
- Bauer, S., Köhler, S., Schulz, M.H., Robinson, P.N., 2012. Bayesian ontology querying for accurate and noise-tolerant semantic searches. *Bioinformatics* 28, 2502–2508.
- Bodenreider, O., 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research* 32, D267–D270.
- Bragin, E., Chatzimichali, E.A., Wright, C.F., *et al.*, 2014. DECIPHER: Database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation. *Nucleic Acids Research* 42, D993–D1000.
- Brown, E.G., Wood, L., Wood, S., 1999. The medical dictionary for regulatory activities (MedDRA). *Drug Safety* 20, 109–117.
- Chen, C.-K., Mungall, C.J., Gkoutos, G.V., *et al.*, 2012. MouseFinder: Candidate disease genes from mouse phenotype data. *Human Mutation* 33, 858–866.
- Choquet, R., Maaroufi, M., Fonjallaz, Y., *et al.*, 2015. LORD: A phenotype-genotype semantically integrated biomedical data tool to support rare disease diagnosis coding in health information systems. *AMIA Annual Symposium Proceedings. AMIA Symposium* 2015, pp. 434–440.
- Cote, R.G., Jones, P., Martens, L., *et al.*, 2008. The ontology lookup service: More data and better tools for controlled vocabulary queries. *Nucleic Acids Research* 36, W372–W376.
- Coulet, A., Smal-Tabbone, M., Napoli, A., Devignes, M.-D., 2006. Suggested ontology for pharmacogenomics (SO-Pharm): Modular construction and preliminary testing. Berlin, Heidelberg: Springer, pp. 648–657.
- Day-Richter, J., Harris, M.A., Haendel, M., Lewis, S., 2007. OBO-Edit an ontology editor for biologists. *Bioinformatics* 23, 2198–2200.
- Donnelly, K., 2006. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in Health Technology and Informatics* 121, 279–290.
- Falk, M.J., Shen, L., Gonzalez, M., *et al.*, 2015. Mitochondrial Disease Sequence Data Resource (MSeqDR): A global grass-roots consortium to facilitate deposition, curation, annotation, and integrated analysis of genomic data for the mitochondrial disease clinical and research communities. *Molecular Genetics and Metabolism* 114, 388–396.
- Gene Ontology Consortium, T.G.O., 2001. Creating the gene ontology resource: Design and implementation. *Genome Research* 11, 1425–1433.
- Girdea, M., Dumitriu, S., Fiume, M., *et al.*, 2013. PhenoTips: Patient phenotyping software for clinical and research use. *Human Mutation* 34, 1057–1065.
- Gkoutos, G.V., Green, E.C.J., Mallon, A.-M., *et al.*, 2005. Using ontologies to describe mouse phenotypes. *Genome Biology* 6, R8.
- Goh, K.-I., Cusick, M.E., Valle, D., *et al.*, 2007. The human disease network. *Proceedings of the National Academy of Sciences* 104, 8685–8690.
- Gottlieb, A., Stein, G.Y., Rupp, E., Sharan, R., 2011. PREDICT: A method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology* 7, 496.
- Groza, T., Hunter, J., Zankl, A., 2012. The bone dysplasia ontology: Integrating genotype and phenotype information in the skeletal dysplasia domain. *BMC Bioinformatics* 13, 50.
- Guest, S.S., Evans, C.D., Winter, R.M., 1999. The online london dysmorphology database. *Genetics in Medicine* 1, 207–212.
- Hamosh, A., Scott, A.F., Amberger, J.S., *et al.*, 2005. Online mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research* 33, D514–D517.
- Hamosh, A., Sobreira, N., Hoover-Fong, J., *et al.*, 2013. PhenoDB: A new web-based tool for the collection, storage, and analysis of phenotypic features. *Human Mutation* 34, n/a/n/a.
- Hastings, J., de Matos, P., Dekker, A., *et al.*, 2013. The ChEBI reference database and ontology for biologically relevant chemistry: Enhancements for 2013. *Nucleic Acids Research* 41, D456–D463.
- Hoehndorf, R., Schofield, P.N., Gkoutos, G.V., 2011. PhenomeNET: A whole-phenome approach to disease gene discovery. *Nucleic Acids Research* 39, e119.
- Köhler, S., Doelken, S.C., Mungall, C.J., *et al.*, 2014a. The Human Phenotype Ontology project: Linking molecular biology and disease through phenotype data. *Nucleic Acids Research* 42, 966–974.
- Köhler, S., Doelken, S.C., Ruef, B.J., *et al.*, 2013. Construction and accessibility of a cross-species phenotype ontology along with gene annotations for biomedical research. *F1000Research* 2, 30.
- Köhler, S., Schoeneberg, U., Czeschik, J.C., *et al.*, 2014b. Clinical interpretation of CNVs with cross-species phenotype data. *Journal of Medical Genetics* 51, 766–772.
- Köhler, S., Schulz, M.H., Krawitz, P., *et al.*, 2009. Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *American Journal of Human Genetics* 85, 457–464.
- Köhler, S., Vasilevsky, N.A., Engelstad, M., *et al.*, 2016. The human phenotype ontology in 2017. *Nucleic Acids Research* 45, gkw1039.
- Kutler, D., Singh, B., Satagopan, J., *et al.*, 2003. A 20-year perspective on the international fanconi anemia registry (IFAR). *Blood*.
- Lipscomb, C.E., 2000. Medical subject headings (MeSH). *Bulletin of the Medical Library Association* 88, 265–266.
- McMurry, J.A., Köhler, S., Washington, N.L., *et al.*, 2016. Navigating the phenotype frontier: The monarch initiative. *Genetics* 203.
- Mungall, C.J., Torniai, C., Gkoutos, G.V., *et al.*, 2012. Uberon, an integrative multi-species anatomy ontology. *Genome Biology* 13, R5.
- Noy, N.F., Shah, N.H., Whetzel, P.L., *et al.*, 2009. BioPortal: Ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research* 37, W170–W173.
- Ong, E., Xiang, Z., Zhao, B., *et al.*, 2017. Ontobee: A linked ontology data server to support ontology term dereferencing, linkage, query and integration. *Nucleic Acids Research* 45, D347–D352.
- Pesquita, C., Faria, D., Bastos, H., *et al.*, 2008. Metrics for GO based protein semantic similarity: A systematic evaluation. *BMC Bioinformatics* 9, S4.
- Piñero, J., Bravo, A., Queralt-Rosinach, N., *et al.*, 2017. DisGeNET: A comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research* 45, D833–D839.
- Portales-Casamar, E., Chng, C., Lui, F., *et al.*, 2013. Neurocarta: Aggregating and sharing disease-gene relations for the neurosciences. *BMC Genomics* 14, 129.
- Rath, A., Olry, A., Dhombres, F., *et al.*, 2012. Representation of rare diseases in health information systems: The orphanet approach to serve a wide range of end users. *Human Mutation* 33, 803–808.
- Robinson, P.N., Köhler, S., Bauer, S., *et al.*, 2008. The human phenotype ontology: A tool for annotating and analyzing human hereditary disease. *American Journal of Human Genetics*. doi:10.1016/j.ajhg.2008.09.017.
- Robinson, P.N., Köhler, S., Oellrich, A., *et al.*, 2014. Improved exome prioritization of disease genes through cross-species phenotype comparison. *Genome Research* 24, 340–348.
- Robinson, P.N., Mundlos, S., 2010. The human phenotype ontology. *Clinical Genetics* 77, 525–534.
- Rosse, C., Mejino, J.L.V., 2003. A reference ontology for biomedical informatics: The Foundational Model of Anatomy. *Journal of Biomedical Informatics* 36, 478–500.
- Salvadores, M., Alexander, P.R., Musen, M.A., Noy, N.F., 2013. BioPortal as a dataset of linked biomedical ontologies and terminologies in RDF. *Semantic Web* 4, 277–284.
- Sharma, M., Phillips, J., Agarwal, S., 2013. Clinical genomicist workstation AMIA Summits on Translational Science Proceedings.
- Silva, F.A.B., Silva, M.J., Couto, F.M., 2010. Epidemic marketplace: an e-science platform for epidemic modelling and analysis. *ERCIM News* 82, 43–44.
- Singleton, M.V., Guthery, S.L., Voelkerding, K.V., *et al.*, 2014. Phevor combines multiple biomedical ontologies for accurate identification of disease-causing alleles in single individuals and small nuclear families. *The American Journal of Human Genetics* 94, 599–610.
- Smedley, D., Oellrich, A., Köhler, S., *et al.*, 2013. PhenoDigm: Analyzing curated annotations to associate animal models with human diseases. *Database* 2013, bat025.

- Smith, B., Ashburner, M., Rosse, C., *et al.*, 2007. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25, 1251–1255.
- Smith, C.L., Goldsmith, C.-A.W., Eppig, J.T., 2005. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology* 6, R7.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* 147, 195–197.
- Sundberg, J.P., Sundberg, B.A., Schofield, P., 2008. Integrating mouse anatomy and pathology ontologies into a phenotyping database: Tools for data capture and training. *Mammalian Genome: Official Journal of the International Mammalian Genome Society* 19, 413–419.
- Vulto-van Silfhout, A.T., van Ravenswaaij, C.M.A., Hehir-Kwa, J.Y., *et al.*, 2013. An update on ECARUCA, the european cytogeneticists association register of unbalanced chromosome aberrations. *European Journal of Medical Genetics* 56, 471–474.
- Winnenburg, R., Bodenreider, O., 2014. Coverage of Phenotypes in Standard Terminologies. *Proceedings of the Joint Bio-Ontologies and BioLINK ISMB'2014 SIG Session Phenotype Day*.
- Zemotitel, T., Köhler, S., Mackenroth, L., *et al.*, 2014. Effective diagnosis of genetic disease by computational phenotype analysis of the disease-associated genome. *Science Translational Medicine* 6.
- Zoubarev, A., Hamer, K.M., Keshav, K.D., *et al.*, 2012. Gemma: A resource for the reuse, sharing and meta-analysis of expression profiling data. *Bioinformatics* 28, 2272–2273.

Further Reading

- Deans, A.R., Lewis, S.E., Huala, E., *et al.*, 2015. Finding our way through phenotypes. *PLOS Biology* 13, e1002033.
- Gkoutos, G.V., Schofield, P.N., Hoehndorf, R., 2017. The anatomy of phenotype ontologies: Principles, properties and applications. *Briefings in Bioinformatics* 9, 601–605.
- Groza, T., Köhler, S., Doelken, S., *et al.*, 2015a. Automatic concept recognition using the human phenotype ontology reference and test suite corpora. *Database: The Journal of Biological Databases and Curation* 2015, 1–13.
- Groza, T., Köhler, S., Moldenhauer, D., *et al.*, 2015b. The human phenotype ontology: Semantic unification of common and rare disease. *American Journal of Human Genetics* 97, 111–124.
- Westbury, S.K., Turro, E., Greene, D., *et al.*, 2015. Human phenotype ontology annotation and cluster analysis to unravel genetic defects in 707 cases with unexplained bleeding and platelet disorders. *Genome Medicine* 7, 36.

Relevant Websites

- <https://bioportal.bioontology.org/ontologies/HP/>
'BioPortal'.
- <https://github.com/obophenotype/human-phenotype-ontology/>
'GitHub repository'.
- <http://human-phenotype-ontology.org/>
'HPO official Web page'.
- <https://monarchinitiative.org/phenotype/>
'Monarch Initiative'.
- https://mseqdr.org/hpo_browser.php/
'MSeqDR browser'.
- <http://purl.obolibrary.org/obo/hp.obo>
'OBO download URL'.
- <http://www.obofoundry.org/ontology/hp.html/>
'OBO Foundry'.
- <http://www.ebi.ac.uk/ols/ontologies/hp/>
'Online Lookup Service'.
- <http://www.ontobee.org/ontology/HP/>
'Ontobee'.
- <http://purl.obolibrary.org/obo/hp.owl>
'OWL download URL'.

Biographical Sketch



Anna Bernasconi received her Master's Degree in Computer Engineering in 2015 from Politecnico di Milano and a Master in Computer Science at the University of Illinois at Chicago in 2016. She is currently a PhD candidate at Politecnico di Milano, working in the field of data-driven genomic computing. Her research interests are in the area of bioinformatics data and metadata integration in order to support answering to complex biological queries. She is currently focusing on public repositories of open data, controlled biomedical vocabularies and ontologies.



Marco Masseroli received the Laurea degree in electronic engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in biomedical engineering in 1996, from the Universidad de Granada, Spain. He is associate professor in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano, and lecturer of bioinformatics and biomedical informatics. He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomía Patológica, Facultad de Medicina at the Universidad de Granada, Spain, and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health, Bethesda. His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Biological and Medical Ontologies: Systems Biology Ontology (SBO)

Anna Bernasconi and Marco Masseroli, Politecnico di Milano, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the last decades, data relevant to molecular and cellular biology has grown enormously, encouraging Systems Biology approaches, which use computational modelling as a means to understand large amounts of diverse information. In order to exchange and reuse models in appropriate ways, the community relies on standards for biological quantitative models' curation with increasing interest. Using computational modelling to describe and analyze biological systems has thus become a fundamental activity of Systems Biology.

While common data formats such as Systems Biology Markup Language (SBML) (Hucka, 2013) or Cell Markup Language (CellML) (Lloyd *et al.*, 2004) have answered to the need for exchange and integrate models on a syntactic level (encoding them in agreed formats), the community has highlighted a growing need for an additional semantic layer. Note that the semantic layer is not necessary in general for model simulations; however, it enhances the ability to understand and analyze models in a programmatic way. One of the efforts in this direction has been concretized with the Minimal Information Requested in the Annotation of Models (Le Novère *et al.*, 2005), a set of rules for curating quantitative models of biological systems. Nevertheless, semantic integration is a goal typically pursued through the use of an ontology.

The Systems Biology Ontology (SBO) has been therefore designed to provide semantic information and characterization of the model components, in addition to annotate the results of biochemical experiments to facilitate their resourceful use. While existing ontologies already covered the biological aspects described by models, the SBO takes care of encompassing this idea to model-related aspects. Indeed, one of the objectives of the SBO is to simplify the direct identification of the relation between one of the components of a model and the structure of the overall model, increasing the usability and re-usability of the model. The annotation of a model can be performed at all stages of its lifecycle, when considered convenient. As an example, the model creator might be interested in providing information about its constructions, built-in assumptions, and the framework in which the model was conceived. A following modeler, after running multiple simulations, may find useful to add information such as a definition of the nature of the parameters, specific rate laws that can be applied, the mathematical expressions that govern the behavior of the instantiated system. Finally, also a biological investigator could enrich the data about the model by specifying the participants' role in reactions encoded in the model.

At the time of writing this article, the SBO is a set of seven controlled, relational vocabularies of terms that are hierarchically organized. The effort made by the SBO team to standardize the encoding of semantics for models (both discrete and continuous) in systems biology was first conceived during the 9th SBML Forum Meeting in Heidelberg on October 14–15, 2004. In that occasion, it was observed that much of the knowledge necessary to understand and simulate a model was only held by modelers (it was not encoded in SBML, indeed). A solution to this problem, by creating a controlled vocabulary to store this information, was proposed by Nicholas Le Novère shortly after. The idea of the SBO has appeared for the first time in Le Novère (2005) and the SBO is now a well-established software tool, publicly available, and shared with free consensus.

SBO is a candidate ontology of the Open Biomedical Ontologies (OBO) Foundry, available to use for free. SBO is a project of the BioModels.net effort, which defines agreed-upon standards for model curation. Its development is organized as a collaboration among the BioModels.net team (EMBL-EBI, United Kingdom), the Le Novère lab (Babraham Institute, United Kingdom), and the SBML team (Caltech, United States). The ontology is also open to contributions toward its development from the community. For requesting new terms creation or suggesting existing terms modifications, the users can exploit dedicated trackers that are available on SBO SourceForge project (see 'SourceForge page' in Relevant Websites section). As to contacting the team that develops SBO, they manage a mailing list for general discussions about BioModels.net projects which works by email subscription. SBO benefited from the funds of the European Molecular Biology Laboratory and the National Institute of General Medical Sciences.

In June 2017, at the time of writing this article, the BioPortal (Noy *et al.*, 2009) website lists 3 projects using this ontology: MaSyMoS, a Neo4J instance containing SBO data along with other bio-ontologies, models, and simulation descriptions (Henkel and Waltemath, 2014), Systems Biology Graphical Notation (SBGN), a visual language that allows the representation of networks of biochemical interactions (Le Novère *et al.*, 2009), and SBML, the standard exchange format for computational models already mentioned.

The following of this article is organized as follows: Section Development defines the historical development of the ontology since its 2006 version, until its state at the time of writing this article, in June 2017. Section The Systems Biology Ontology describes the SBO with focus on its structure, terms, and relations. Section Interoperability and Quantitative Aspects overviews interoperability with other resources used in Systems Biology and outlines quantitative aspects on the coverage and mappings provided by the SBO. Section Technological Details describes the technological aspects regarding its access, search functionalities, implementation, API, and changes management. Section Applications in Other Works presents a list of other applications that employ the SBO, or contribute to common pipelines. Finally, Section Use Case concludes with a use case scenario describing a simple activity of a user taking advantage of the SBO.

Development

The version of the SBO documented in Le Novère paper (Le Novère, 2005) mentions the introduction of the SBO within a big international initiative: BioModels.net, along with MIRIAM and BioModels Database (Li *et al.*, 2010a).

The SBO intended to be a first hierarchical structuring of knowledge on the matters of Systems Biology and the underlying models. Its main purpose was to facilitate the direct identification of the relation between a model component and the model structure. It presented three different vocabularies: a classification of rate laws (e.g., *Mass action*, *Henri-Michaelis-Menten kinetics*), a taxonomy of the roles of reaction participants (e.g., *substrate*, *catalyst*, *inhibitor*), and a terminology for parameter roles in quantitative models (e.g., *Hill coefficient*, *Michaelis coefficient*). The innovative aspect of such ontology, compared to more classical ones such as the Gene Ontology, was that the links in SBO crossed the vocabulary barriers. For instance, a term which defined a rate-law was allowed to have children representing the relevant reacting species parameters.

Using SBO terms to annotate model components was an essential step for annotation methods to be compliant with the standard indicated by MIRIAM and also influenced the search strategies used by models databases (such as BioModels Database).

Le Novère (2006) presented an improved version of the ontology, which featured four different vocabularies. One new branch, referring to the three previous ones from Le Novère (2005), was added to include a list of modelling frameworks that precise how to interpret a mathematical expression (e.g., *deterministic*, *stochastic*, *boolean*). This subset of terms, used within SBML, would allow to get rid of explicit mathematics in the model itself, downloading a correct and updated rate-law in its place.

Le Novère *et al.* (2006) documents a renovated version of the SBO which had by then become part of the OBO Foundry. The development team planned to export the ontology not only in OBO flat format, but also in OBO-XML (containing the same information that OBO-flat, expressed in eXtensible Markup Language) and in OWL (Web Ontology Language). At that moment, the ontology was able to index, define, and relate terms based on five different vocabularies. A branch for the classification of events represented by biochemical models (e.g., *binding*, *transport*, *degradation*) had been added with respect to the version of Le Novère (2006).

Li *et al.* (2010b) described the increased use of the SBO through a solid implementation and organization of their Web services, which improved the interoperability of the ontology.

At the time of writing this article, the SBO website exhibits an updated 'News' page containing detailed chronological facts regarding the ontology evolution, from its first steps in 2006 until the latest major updates, in 2013.

Courtot *et al.* (2011) is a reference community paper which covers the ontologies developed by BioModels.net, including the SBO. In this article, other domain ontologies, such KiSAO and TEDDY (Knüpfer *et al.*, 2009) are described. They are employed to characterize aspects of the systems' dynamic behaviors.

The Systems Biology Ontology

The SBO consists of seven orthogonal vocabularies each represented by a *branch* of the ontology, connected to the root of the ontology by a *is_a* relationship. As of June 2017, they appear on the Tree browsing section on SBO page (see 'SBO website' in Relevant Websites section) in the following form (also synthesized in Fig. 1):

1. SBO:0000236 – *physical entity* representation branch: the physical nature of the entity, that can be either *material* (e.g., *macromolecule*, *physical compartment*, *observable*) or *functional* (e.g., *transporter*, *enzyme*, *receptor*);
2. SBO:0000003 – *participant role* branch: the roles played by reaction participants (e.g., *reactant*, *product*, *modifier*);

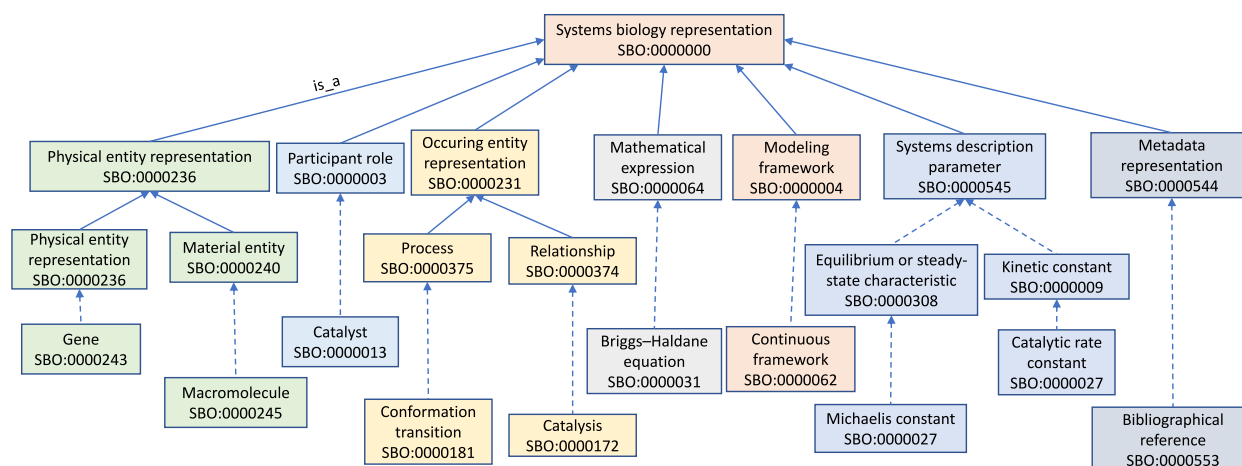


Fig. 1 SBO terms are distributed in seven orthogonal branches, as described in Section The Systems Biology Ontology. In the graph, the arrows with a continuous line represent simple *is_a* relationships, while the arrows with a dashed line indicate that intermediate terms are omitted.

3. SBO:0000231 – *occurring entity representation* branch: the type of occurring interaction, in terms of types of *process* (e.g., *composite biochemical process, molecular, or genetic interaction*) and the *relationship* that an entity represents with respect to another (e.g., *inhibition or stimulation*);
4. SBO:0000064 – *mathematical expression* branch: the mathematical expressions describing the system (e.g., *enzymatic rate law, mass-action rate law, Hill-type rate law*);
5. SBO:0000004 – *modelling framework* branch: the modelling framework within which the model was constructed, critical to recognize the possible limitations under different simulation conditions (e.g., *discrete framework, concrete framework, logical framework*);
6. SBO:0000545 – *system description parameter* branch: quantitative parameters to describe the systems (e.g., *Michaelis constant, catalytic constant, thermodynamic temperature*);
7. SBO:0000544 – *metadata representation* branch: the types of metadata represented within model components (e.g., *annotation, bibliographical reference*).

The user should note that the terms in the *system description parameter* and *mathematical expression* branches contain Mathematical Markup Language (MathML) Version 2 mathematical expressions (Ausbrooks *et al.*, 2003), which allow to remove ambiguities in the semantic meaning. These are used to relate the symbols used in the model expressions to the actual parameters referenced in the ontology.

SBO Term Fields

Table 1 reports all the fields regarding a single term that the ontology stores and makes available for manual and programmatic exploration. As an example, the specific values for the term *catalytic rate constant* are shown.

The identifiers are made by the “SBO” prefix, followed by a colon separator and a 7-digit identifier (e.g., SBO:1234567). A SBO identifier is assigned to a term when it is introduced in the ontology and characterizes that concept term permanently, although some of the information connected to it (such as its name, definition, or other fields) may change over time. When a term needs refinement, the central concept is not affected, only further clarified. When new terms are introduced, it often becomes necessary to change relationships with existing *child* and *parent* terms that must be refined to allow conceptual distinction between them.

In sporadic cases, a term is considered obsolete because it does not fit anymore in the SBO domain of interest. Such situation can happen when the term in question: (1) is out-dated, (2) becomes redundant because the same concept is represented by an improved term, (3) becomes redundant because another term covers the same concept space, (4) was created by mistake.

When a term becomes obsolete, it is not visible anymore in the ontology. Nevertheless, it is not removed from the downloadable ontology files (in which it is marked with the property *is_obsolete*: true) and Web browser (in which they are marked with the tag “(OBSOLETE TERM)” in red beside the SBO identifier), to still allow for its search. When applicable, terms that can be used in alternative to the obsolete term are proposed.

Table 1 Description of SBO term fields and their example values for the term SBO:0000025 – catalytic rate constant

Field	Meaning	Optional	Example
Identifier	The SBO identifier appears at the top of the pop up display (marked obsolete in case of obsolete terms).		SBO:0000025
Name	The full name of the SBO term.		catalytic rate constant
Synonyms	Known synonyms for the term name.	X	kcat, turnover number
Definition	Textual definition describing the precise meaning and context of use for the selected term.		Numerical parameter that quantifies the velocity of an enzymatic reaction.
MathML	A standardised XML-based mathematical definition of the term.	X	
Rendered equation	The graphically rendered representation of the mathematical definition presented in the MathML field, if applicable.	X	
Comment	Clarification comments inserted by the curator to further define or clarify the use of the term		“irreversible” removed on March 11, 2007 by Nicolas Le Novère
Miscellaneous	Creation and last modification dates for the term.		<i>Date of creation</i> : 13 February 2006, 23:41 <i>Date of last modification</i> : 11 March 2007, 21:39
Parents	Hyperlinked SBO identifier(s), and term name(s) of the immediate parent(s) of the displayed term.		SBO:0000035 forward unimolecular rate constant, continuous case (is_a)
Children	Hyperlinked SBO identifier(s), and term name(s) of the immediate children of the displayed term		SBO:0000320 product catalytic rate constant (is a) SBO:0000321 substrate catalytic rate constant (is a)
History	Lists the dates, actions and details of all changes to the displayed term, since its introduction to the ontology.		11/03/200: term updated. Some information about this term have been updated 13/02/2006: term created. This term has been newly created

```

<math xmlns="http://www.w3.org/1998/Math/MathML">
  <lambda>
    <bvar><ci definitionURL="http://biomodels.net/SBO/#SBO:0000338">koff</ci></bvar>
    <bvar><ci definitionURL="http://biomodels.net/SBO/#SBO:0000341">Kon</ci></bvar>
    <apply>
      <divide/>
      <ci>koff</ci>
      <ci>Kon</ci>
    </apply>
  </lambda>
</math>

```

Fig. 2 MathML expression for SBO:0000282 – dissociation constant term, contained in the definition field of the OBO file of the SBO.

SBO Relationships

Terms are arranged within a branch of the SBO ontology using only *is_a* relationships (subsumptions), which entail that each term is a particular version of its parent term (e.g., *boolean switch* is *a switch value*).

The terms contained in different SBO vocabularies (of the available seven ones) are not directly linked to each other. However, some terms of the ontology contain MathML (see ‘MathML 2.0’ in the Relevant Websites section) lambda functions that contain links to terms belonging to other branches. As an example, [Fig. 2](#) reports the MathML entry for the *dissociation constant*.

Interoperability and Quantitative Aspects

At the time of writing this article, in June 2017, according to the most updated version, the Systems Biology Ontology featured 648 concepts in total, 21 of which are considered obsolete.

Among the various defined formats within which computational models can be encoded in order to be annotated with SBO concepts, SBML, CellML, and GENESIS ([Bower and Beeman, 1998](#)) are worth mentioning. SBML, in particular, provides a specific integration mechanism to accept SBO terms that can be integrated into its file format (the attribute *sboTerm* is specifically used for linking SBO term annotations with model components).

According to the interoperability study shown in BioPortal website ([Salvadores et al., 2013](#)), as of June 2017, notable overlapping is observed with the Computational Neuroscience Ontology ([Le Franc et al., 2012](#)), MeSH (in its Robert Hoehndorf Version) ([Lipscomb, 2000](#)), and the National Cancer Institute (NCI) Thesaurus ([Sioutos et al., 2007](#)), where many concepts of SBO coexist. These are followed by the Neuroscience Information Framework (NIF) Standard Ontology ([Bug et al., 2008](#)), the BioModels Ontology ([Hoehndorf et al., 2011](#)), and SNOMED CT ([Donnelly, 2006](#)). Overlaps with other notable ontologies can be found in: Gene Expression Ontology (126 terms) ([Barrett et al., 2013](#)), Gene Ontology (50 terms), International Classification of Diseases, Version 10 (32 terms) (see ‘ICD, World Health Organization’ in Relevant Websites section) ([Rath et al., 2012](#)), and the Mammalian Phenotype Ontology (24 terms) ([Smith et al., 2005](#)).

Technological Details

The back-end system on which the SBO is based is a MySQL relational database management system. This is accessed through a front-end Web interface built on Java Server Pages and Java Beans. The content uses UTF-8 encoding, thus supporting a broad set of characters within terms and their definitions.

Access Mode

SBO is an open-resource ontology (therefore listed on SourceForge), developed and maintained by the scientific community, whose files are reusable under the terms of the Artistic License 2.0. This means that it is permitted to copy and distribute verbatim copies of it, but changes are not allowed (see ‘License’ in Relevant Websites section).

Instant exports are automatically generated daily at 7.00 am United Kingdom time (or on specific request), and they can be directly downloaded through the specific page in the website (see ‘SBO download’ in the Relevant Websites section). They are available in OBO (full and no MathML versions), OWL and XML formats. In addition, the XML (and MathML) schemas used by the SBO are provided. In the same Web page, a minimal ‘Statistics’ tab is provided, i.e., a summary display which specifies the date of the last modification to the ontology.

The Systems Biology Ontology can also be explored by using the public interface Web browsers in the European Bioinformatics Institute (EBI) website (see ‘SBO website’ in Relevant Websites section), the National Center for Biomedical Ontology (NCBO) BioPortal (see ‘SBO BioPortal’ in Relevant Websites section) ([Noy et al., 2009](#)), or the OBO Foundry website (see ‘SBO OBO Foundry’ in Relevant Websites section), which provides the download of the OWL version of the ontology.

Search and visualization of a specific SBO term

The EBI SBO website offers a dedicated resource to curate and maintain the SBO. The SBO search box is available in all pages of the SBO website in the top right corner. The text entered in the search box can be used to find keywords in the fields identifier, name, definition, synonyms, MathML, and comment of SBO terms. Results are provided as a list of tables created based on what the entered text has matched in the term information (i.e., Terms with a match in the *accession*, Terms with a match in the *name*, Terms with a match in the *definition*, *comment* or *mathML*, Terms with a match in the *synonym*). Each table features the SBO Accession, its term name, and clickable links to the tree and entry view of that term.

The search is conjunctive (searched words are combined through the logical operator AND), meaning that when more than one word is entered in the search box, the results will only include terms where all words appear within an SBO term. The disjunctive search (which allows the logical OR between words) and more advanced kinds of searches are envisioned as future works.

The SBO can be also browsed as an ontology tree, accessible through the Browse function on the Web page (see 'SBO website' in Relevant Websites section). The tree can be navigated by expanding and collapsing its branches, using the icons marked with plus and minus symbols, respectively, positioned on the left side of each sub-tree. By clicking on a term, the user can open a pop up display that contains the specific information for that particular entry (known as the entry view), which includes the fields specified in Section SBO Term Fields.

Alternatively, the entry view of specific terms can be directly accessed using dedicated URLs, built by appending the SBO identifier of the term to the SBO website URL.

The SBO Rest API

To allow programmatic access to the terms and sub-trees of the ontology from other software tools, the SBO team provided a first (soon to be discontinued) version of Web Services whose communication layer had been implemented based on Apache Axis. The updated version uses instead a new Web Services stack (JAX-WS). From the SourceForge project Web page, a runnable Java library can be downloaded, useful to communicate with the SBO Web Services in a Java program.

The services use official SBO identifiers (such as SBO:0000135) as input parameters for all available methods. The implementation is based on a simple *Term* Java Object class, which represents the concept of the ontology and features all the fields of the ontology terms as described in Section SBO Term Fields. The output returned by each service is given as a null or not-empty list of objects.

The possible queries include the ability to retrieve SBO terms (either only one or a list of them, as the subtree of a given term) described in XML and OWL format, or just as a simple OBO object. As an example, the method *getTerm*(SBO:0000192) retrieves the term corresponding to the given identifier, i.e., *Hill-type rate law, generalised form*. This will be in the form of an object which contains all the information stored in the ontology about the associated identifier. On the other hand, *getOWLTerm*(SBO:0000192) retrieves the OWL format of the same term with all its information.

It is possible to request: direct children and parents of a term, the whole sub-tree starting from a term, a check that a term is a (direct) child of another one, a check that a term is obsolete, or a check that a term corresponds to the SBO root. For instance, *getTreeOWL*(SBO:0000009) retrieves the subtree starting from the term *kinetic constant* (whose identifier is SBO:0000009), in the OWL format.

Search for terms can be made on the whole content of the ontology (e.g., with *searchOWL()*), on the possible completions of the word(s) given in the search parameter (e.g., with *searchPossibleCompletions()*), on the Details, MathML, Name, or Synonyms sections of the terms (e.g., with *searchStringTermMath()* and *searchTermDetailsOWL()*).

Changes and Tracking

Distributed curation is made possible by using a custom-tailored locking system allowing concurrent access. This system allows a continuous update of the ontology with immediate availability and suppress merging problems.

Requests for new terms, changes, and additional features can be made through the tracker tool provided in the SourceForge page, along with reports of errors in existing terms. They are evaluated by an SBO curator, with possible contact of the submitter for clarification purposes. Other issues should be addressed to the email contacts provided on the SBO website.

At the time of writing this article, the SBO presented no versioning system, which is considered as one of the future additional features. The only way to identify what version an ontology file represents is by looking at the *data-version* field at the top of the file downloadable on the website (see 'SBO Downloads' in the Relevant Websites section).

In the Browse menu of the main website the user can access the "History" section, a list of recently added or updated terms. The same information can be found in the single SBO terms, under their history and comment fields.

Applications in Other Works

As mentioned in Section Introduction, the SBO is part of the BioModels.net project, that aims at integrating the collaborative use of the following three technologies: (1) MIRIAM, as a standard to curate and annotate models to facilitate their reuse, (2) the SBO, as a set of controlled vocabularies to be used to characterize the models' components, and (3) the BioModels Database, as a resource to store, search, and retrieve published models of interest. These resources, together with the language SBML, are supporting the productive exchange and reuse of computational models.

The interoperability of the SBO with the SBML has been set since the Level 2 Version 2 of the SBML, which now provides a mechanism to annotate model components with SBO terms. This mechanism increases the semantics of a model, enriching the information about its interaction topology and mathematical expression. *SBMLsqueezer* (Dräger *et al.*, 2008) is a modelling application that helps to identify SBO terms based on existing model components, with the aim to generate suitable mathematical relationships over biochemical maps. This is based on the automation of equations generation, overcoming the highly error-prone and cumbersome process of manually assigning kinetic equations, such as the Henri-Michaelis-Menten equation.

SemanticSBML (Krause *et al.*, 2010) is a tool that uses the SBO annotation to integrate individual models into a larger one. Furthermore, all graphical symbols used in the SBGN languages are linked to an SBO term, thus helping to generate SBGN maps from SBML models.

The Systems Biology Pathway Exchange (SBPAX) is an integration effort that allows SBO terms to be added to the Biological Pathway Exchange (BioPAX) format, which is linked to information useful for computational modelling, such as the quantitative descriptions contained in SBO.

Use Case

Li *et al.* (2010b) introduced the integrated use of the technologies of MIRIAM, SBO and BioModels Database as a unified toolkit to analyze biological mathematical models. Let us consider a typical example workflow that uses all three tools, by querying their services. The studied biological problem corresponds to the one introduced in Hong *et al.* (2009), i.e., the molecular mechanism underlying how DNA damage causes predominantly phase advances in the circadian clock.

Since the peer-reviewed journal where the (Hong *et al.*, 2009) study is published suggests to their authors to deposit their data into a public repository, Hong and colleagues submitted their model to the BioModels Database, where it is freely available (identified by the ID BIOMD0000000216). By means of the appropriate method of the BioModels Web Services, with the specification of the BioModels Database ID BIOMD0000000216 as a parameter, the user can retrieve the model encoded in SBML. Fig. 3 shows an excerpt from the retrieved XML file describing the model, specifically a part that is contained inside its *listOfSpecies* XML element tag.

In such a SBML representation, the user can find lists with all the computationally interesting components of the model, such as *listOfFunctionDefinitions*, *listOfCompartments*, *listOfSpecies*, *listOfParameters*, *listOfRules*, and *listOfReactions*, just to mention some.

Let us suppose that, among these many elements, the user is interested in understanding the nature of the SBML *species* denoted as “CP” (the one shown in Fig. 3, indeed). In order to examine it more in depth, he/she can look at the SBO term associated with it. As it can be observed in the *species* element of Fig. 3, the SBML *species* is associated with the attribute *sboTerm*: “SBO:0000252”. As indicated in Section The SBO Rest API, the SBO Web Service method *getTerm*(‘SBO:0000296’) can be used in such a case to retrieve more information about the term. It returns the full record of this term, which includes its name (i.e., *polypeptide chain*) along with other information, as shown in Fig. 4.

Whenever this should not be enough, the user should notice that an entity (i.e., the species “CP” in this example) is associated with some annotations (contained within the *annotation* XML element tags, see Fig. 3), encoded in RDF following the SBML specifications. This information can be used to further investigate the properties of this entity. For instance, in this example the user might want to find “CP” in the UniProt database (UniProt Consortium, 2015), whose entries can be retrieved from the MIRIAM Web Services. The user will need an identifier to search for further details on the UniProt Knowledge Base. In particular, the UniProtKB identifier O15516 can be extracted as the final part of the URL in the *rdf:li* XML element tag (see Fig. 3). By using the appropriate MIRIAM Web Service, the user can learn that “CP” is a *polypeptide chain* made of *Circadian locomotor output cycles protein kaput*.

This is an example of the information that can provide a better biological understanding of the biological mathematical model. The integrated use of the three modules of BioModels.net (i.e., MIRIAM, SBO and BioModels Database), in case together with

```
<species id="CP" initialConcentration="0.037" name="CP" metaid="metaid_0000029" sboTerm="SBO:0000252" compartment="system">
  <annotation>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
      xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
      <rdf:Description rdf:about="#metaid_0000029">
        <bqbiol:isHomologTo>
          <rdf:Bag>
            <rdf:li rdf:resource="http://identifiers.org/uniprot/O15516"/>
          </rdf:Bag>
        </bqbiol:isHomologTo>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</species>
```

Fig. 3 Excerpt of the SBML encoded model from Hong *et al.* (2009). Note the usage of the attribute *sboTerm*.

Term: *SBO:0000252*

Name

polypeptide chain

Definition

Naturally occurring macromolecule formed by the repetition of amino-acid residues linked by peptidic bonds. A polypeptide chain is synthesized by the ribosome. CHEBI:16541

Comment

Name changed on January 10 2007 by Nicolas Le Novère, to disambiguate from non-ribosomal peptides. Created on November 10 2006 by Nicolas Le Novère.

Parent(s)

[SBO:0000246](#) information macromolecule (is a)

Miscellaneous

Date of creation: 10 November 2006, 15:38

Date of last modification: 10 January 2007, 13:49

History

Date	Action	Details
10/01/2007	term updated	Some information about this term have been updated
10/11/2006	term created	This term has been newly created

Fig. 4 Details of the SBO term: polypeptide chain (SBO: 0000252).

others, queried through dedicated Web Services, guarantees a way to fully understand the biology behind a computational biology model (which is specified in SBML and curated with the use of the SBO).

See also: Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: Introduction. Computational Systems Biology Applications. Coupling Cell Division to Metabolic Pathways Through Transcription. Data Mining: Clustering. Identification and Extraction of Biomarker Information. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Querying languages and development. Studies of Body Systems

References

- Ausbrooks, R., Buswell, S., Carlisle, D., *et al.*, 2003. Mathematical markup language (MathML) version 2.0., second ed. World Wide Web Consortium, Recommendation.
- Barrett, T., Wilhite, S.E., Ledoux, P., *et al.*, 2013. NCBI GEO: Archive for functional genomics data sets—update. *Nucleic Acids Research* 41, D991–D995.
- Bower, J.M., Beeman, D., 1998. The book of GENESIS: Exploring realistic neural models with the GEneral NEural Simulation System. New York, NY: Springer-Verlag, Inc.
- Bug, W.J., Ascoli, G.A., Grethe, J.S., *et al.*, 2008. The NIFSTD and BIRNLex vocabularies: Building comprehensive ontologies for neuroscience. *Neuroinformatics* 6, 175–194.
- Courtot, M., Juty, N., Knüpf, C., *et al.*, 2011. Controlled vocabularies and semantics in systems biology. *Molecular Systems Biology* 7, 543.
- Donnelly, K., 2006. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in Health Technology and Informatics* 121, 279–290.
- Dräger, A., Hassis, N., Supper, J., *et al.*, 2008. SBMLsqueezer: A CellDesigner plug-in to generate kinetic rate equations for biochemical networks. *BMC Systems Biology* 2, 39.
- Henkel, R., Waltemath, D., 2014. MaSyMoS: Finding hidden treasures in model repositories. *Semantic Web Applications and Tools for the Life Sciences*.
- Hoehndorf, R., Dumontier, M., Gennari, J., *et al.*, 2011. Integrating systems biology models and biomedical ontologies. *BMC Systems Biology* 5, 124.
- Hong, C.I., Zamborsky, J., Csikász-Nagy, A., 2009. Minimum criteria for DNA damage-induced phase advances in circadian rhythms. *PLOS Computational Biology* 5, e1000384.
- Hucka, M., 2013. Systems biology markup language (SBML). *Encyclopedia of Systems Biology* SE 1091, 2057–2063.
- Knüpf, C., Köhn, D., Le Novère, N., 2009. Beyond structure: Kisao and TEDDY – two ontologies addressing pragmatical and dynamical aspects of computational models in systems biology. *Nature Proceedings*. doi:10.1038/npre.2009.3137.
- Krause, F., Uhlendorf, J., Lubitz, T., *et al.*, 2010. Annotation and merging of SBML models with semanticSBML. *Bioinformatics* 26, 421–422.
- Le Franc, Y., Davison, A.P., Gleeson, P., *et al.*, 2012. Computational neuroscience ontology: A new tool to provide semantic meaning to your models. *BMC Neuroscience* 13, P149.
- Le Novère, N., 2005. BioModels.net, tools and resources to support computational systems biology. In: *Proceedings of the 4th Workshop on Computation of Biochemical Pathways and Genetic Networks*.
- Le Novère, N., 2006. Model storage, exchange and integration. *BMC Neuroscience* 7 (Suppl. 1), S11.
- Le Novère, N., Courtot, M., Laibe, C., 2006. Adding semantics in kinetics models of biochemical pathways. In: *Proceedings of the 2nd International Symposium on Experimental Standard Conditions of Enzyme Characterizations*.
- Le Novère, N., Finney, A., Hucka, M., *et al.*, 2005. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature Biotechnology* 23, 1509–1515.

- Le Novère, N., Hucka, M., Mi, H., *et al.*, 2009. The systems biology graphical notation. *Nature Biotechnology* 27, 735–741.
- Li, C., Courtot, M., Le Novère, N., Laibe, C., 2010b. BioModels.net Web Services, a free and integrated toolkit for computational modelling software. *Briefings in Bioinformatics* 11, 270–277.
- Li, C., Donizelli, M., Rodriguez, N., *et al.*, 2010a. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology* 4, 92.
- Lipscomb, C.E., 2000. Medical subject headings (MeSH). *Bulletin of the Medical Library Association* 88, 265–266.
- Lloyd, C.M., Halstead, M.D.B., Nielsen, P.F., 2004. CellML: Its future, present and past. *Progress in Biophysics and Molecular Biology* 85, 433–450.
- Noy, N.F., Shah, N.H., Whetzel, P.L., *et al.*, 2009. BioPortal: Ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research* 37, W170–3.
- Rath, A., Olry, A., Dhombres, F., *et al.*, 2012. Representation of rare diseases in health information systems: The orphanet approach to serve a wide range of end users. *Human Mutation* 33, 803–808.
- Salvadores, M., Alexander, P.R., Musen, M.A., Noy, N.F., 2013. BioPortal as a dataset of linked biomedical ontologies and terminologies in RDF. *Semantic Web* 4, 277–284.
- Sioutos, N., Coronado, S., de, Haber, M.W., *et al.*, 2007. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics* 40, 30–43.
- Smith, C.L., Goldsmith, C.-A.W., Eppig, J.T., 2005. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology* 6, R7.
- UniProt Consortium, 2015. UniProt: A hub for protein information. *Nucleic Acids Research* 43, D204–D212.

Further Reading

- Funahashi, A., Morohashi, M., Kitano, H., Tanimura, N., 2003. CellDesigner: A process diagram editor for gene-regulatory and biochemical networks. *Biosilico* 1, 159–162.
- Hucka, M., Finney, A., Sauro, H.M., *et al.*, 2003. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.
- Juty, N., Le Novère, N., Waltemath, D., Knuepfer, C., 2010. Ontologies for use in systems biology: Sbo, KISAO and TEDDY. *Nature Proceedings*. doi:10.1038/npre.2010.5122.1.
- Waltemath, D., Adams, R., Beard, D.A., *et al.*, 2011. Minimum information about a simulation experiment (MIASE). *PLOS Computational Biology* 7, e1001122.
- Wittig, U., Kania, R., Golebiewski, M., *et al.*, 2012. SABIO-RK – Database for biochemical reaction kinetics. *Nucleic Acids Research* 40, D790–D796.

Relevant Websites

<http://www.who.int/classifications/icd/>
ICD, World Health Organization.

<http://www.opensource.org/licenses/artistic-license.php>
Licence.

<https://www.w3.org/TR/MathML2/>
MathML 2.0.

<https://bioportal.bioontology.org/ontologies/SBO/>
SBO BioPortal.

<http://www.ebi.ac.uk/sbo/main/download/>
SBO Downloads.

<http://www.obofoundry.org/ontology/sbo.html>
SBO OBO Foundry.

<http://www.ebi.ac.uk/sbo/>
SBO website.

<http://www.biomodels.net/sbo/>
SBO website.

<https://sourceforge.net/projects/sbo/>
SourceForge page.

Biographical Sketch



Anna Bernasconi received her Master's Degree in Computer Engineering in 2015 from Politecnico di Milano and a Master in Computer Science at the University of Illinois at Chicago in 2016. She is currently a PhD candidate at Politecnico di Milano, working in the field of data-driven genomic computing. Her research interests are in the area of bioinformatics data and metadata integration in order to support answering to complex biological queries. She is currently focusing on public repositories of open data, controlled biomedical vocabularies and ontologies.



Marco Masseroli received the Laurea degree in Electronic Engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in Biomedical Engineering in 1996, from the Universidad de Granada, Spain. He is associate professor in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano, and lecturer of Bioinformatics and Biomedical Informatics. He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomía Patológica, Facultad de Medicina at the Universidad de Granada, Spain, and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health, Bethesda. His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Ontology-Based Annotation Methods

Pietro H Guzzi, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The production of experimental data in molecular biology has been accompanied by the accumulation of functional information about biological entities. Terms describing such knowledge are usually structured by using formal instruments such as controlled vocabularies and ontologies (Guzzi *et al.*, 2012). The Gene Ontology (GO) project (Harris *et al.*, 2004) has developed a conceptual framework based on ontologies for organizing terms (namely GO Terms) describing biological concepts. It consists of three ontologies: Molecular Function (MF), Biological Process (BP), and Cellular Component (CC) representing different aspects of biological molecules. Each GO Term may be associated with many biological concepts (e.g. proteins or genes) by a process also known as annotation. The whole corpus of annotations is stored in publicly available databases, such as the Gene Ontology Annotation (GOA) database (Camon *et al.*, 2004a).

In such a way records representing the associations of biological concepts, e.g. proteins, and GO terms may be easily represented as $\{P, T_1, \dots, T_n\}$, e.g. {P06727, GO:0002227, GO:0006810, GO:0006869} or {ApolipoproteinA-IV, innate immune response in mucosa, transport, lipid transport}.

The whole set of annotated data represents a valuable resource for the existing approach of analysis. From those, the use of association rules (AR) (Hipp *et al.*, 2000; Guzzi *et al.*, 2014; Zaki *et al.*, 1997) is less popular with respect to other techniques, such as statistical methods or semantic similarities (Cannataro *et al.* 2013, 2015). Existing approaches span from the use of AR to improve the annotation consistency, as presented in Faria *et al.* (2012), to the use of AR to analyse microarray data (Carmona-Saez *et al.*, 2006; Ponzoni *et al.*, 2014; Tew *et al.*, 2014; Benites *et al.*, 2014; Nguyen *et al.*, 2011; Manda *et al.*, 2013; Agapito *et al.*, 2016), (see Naulaerts *et al.*, 2013 for a detailed review).

As we pointed out in a previous work (Guzzi *et al.*, 2014), the use of AR presents two main issues due to the Number and Nature of Annotations (Huttenhower *et al.*, 2009). The number of annotation is for each protein or gene is highly variable within the same GO taxonomy and over different species. The variability is caused by two main facts: (i) The presence of various methods of annotations of data; and (ii) the use of different data sources.

Regarding the *Nature of Annotations*, it should be evidenced that the association between a biological concept and its related GO Term can be performed with 14 different methods. These methods are in general grouped into two broad categories: experimentally verified (or manuals) and Inferred from Electronic Annotation (IEA). IEA annotations are usually derived using computational methods that analyse literature. Each annotation is labelled with an evidence code (EC) to keep track of the technique used to annotate a protein with GO Terms. Manual annotations are, in general, more precise and specific than IEA ones (see Guzzi *et al.*, 2012). Unfortunately, their number is lower, and the ratio among IEA versus non-IEA is variable.

Often many generic GO terms (this is particularly evident when considering novel or not well-studied genes) annotate genes and proteins, and the problem is also referred to *Shallow Annotation Problem*. The role of these non-specific annotations is to suggest an area in which the proteins or genes operate. This phenomenon affects especially IEA annotations derived by using computational methods.

The Process of Ontology-Based Annotation

The process of annotation based on ontologies is structured on two main steps: from one side a structured vocabulary, or ontology, is continuously updated and stores annotation terms, from the other hand a set of methods (manually or automated) associated to biological terms (e.g. proteins) the set of related annotations. In the rest of the article, we explain this process using Gene Ontology and Disease Ontology.

Annotations in Gene Ontology

The GO is a structured vocabulary of terms (namely GO-terms) organised in three ontologies: Molecular Function (MF), Biological Process (BP) and Cellular Component (CC). Each GO-term is uniquely associated with a code. Each ontology describes a particular aspect of a gene or gene product functionality. Each gene or gene product is associated with a set of GO terms (Cho *et al.*, 2013).

Each gene is associated with a set of terms that describe its functionality, and this association is stored into the Gene Ontology Annotation Database (GOA) (Camon *et al.*, 2004b). Each annotation in the GO has a source which can be a literature reference, a database reference or computational evidence (Guzzi *et al.*, 2014). Each annotation is associated with an evidence code describing the process of annotation. There exist 18 evidence codes divided into four categories. It possible to see whether an annotation is supported by more than one type of evidence. The most reliable annotations are those inferred directly from experimental evidence. Since most researchers do add their findings to the GO, professional curators examining the literature

add these findings. The second big class of annotation method is the computational inference, out of which six imply manual curation (ISS, ISO, ISA, ISM, IGC, RCA). Finally, the evidence code IEA is used for all inferences made without any human supervision, regardless of the method used. The IEA evidence code is by far the most abundantly used evidence code. For an in-depth treatment of the topic, we refer the interested reader to some recent reviews. There exist seven different electronic annotation pipelines: Ensembl Compara, InterPro2GO, UniProtKB-Keyword2GO, UniProtKB-Subcellular, Location2GO, UniPathway2GO, EC2GO, and HAMAP2GO.

Retrieving Gene Ontology Annotations: The Gene Ontology Annotation Database

The Gene Ontology Annotation Database (GOA) stores the corpus of electronic and manual associations (annotations) of Gene Ontology (GO) terms to UniProt Knowledgebase (UniProtKB) entries. The GOA is accessible through the QuickGO interface. QuickGO is a web-based tool for searching and viewing GO terms and annotations from the GOA database. Through the use of the software, the user may see, filter and download all the annotation data. Annotation data within QuickGO are updated on a weekly basis. In parallel, the consortium offer web-services for accessing GOA-data at <http://www.ebi.ac.uk/QuickGO-Beta/webservices>.

Annotations in Disease Ontology

The Disease Ontology (DO) database (Schriml *et al.*, 2012) is a knowledge base related to human diseases. The current version stores information about 8043 diseases. DO aims to connect biological data (e.g. genes) considering a disease-centred point of view. The DO semantically integrates a disease and existing medical vocabularies. DO terms, and their DOIDs have been utilised to annotate disease concepts in several primary biomedical resources. The DO is organised into eight central nodes to represent cellular proliferation, mental health, anatomical entity, infectious agent, metabolism and genetic diseases along with medical disorders and syndromes anchored by traceable, stable identifiers (DOIDs). Genes may be annotated with terms coming from DO that may be freely downloaded from the website. DO is structured into a directed acyclic graph (DAG), and the terms are linked by relationships in a hierarchy organised by interrelated subtypes. DO has become a disease knowledge resource for the further exploration of biomedical data, including measuring disease similarity based on functional associations between genes, and it is a disease data source for the building of biomedical databases.

To transfer annotations from Disease Ontology to genes, the MetaMap Transfer tool (MMTx) was used. These mappings are stored in the Open Biomedical Ontologies format and can be manually edited using the open source graph editor DAGEdit <http://geneontology.sourceforge.net>.

See also: Biological and Medical Ontologies: GO and GOA. Computational Tools for Structural Analysis of Proteins. Data Mining: Clustering. Information Retrieval in Life Sciences. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics. Ontology: Definition languages. Ontology in Bioinformatics. Ontology: Introduction. Ontology: Querying languages and development. Protein Functional Annotation

References

- Agapito, G., Milano, M., Guzzi, P.H., Cannataro, M., 2016. Extracting cross-ontology weighted association rules from gene ontology annotations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13 (2), 197–208. Available at: <https://doi.org/10.1109/TCBB.2015.2462348>.
- Benites, F., Simon, S., Sapozhnikova, E., 2014. Mining rare associations between biological ontologies. *PIOS ONE* 9 (1), e84475.
- Camon, E., Magrane, M., Barrell, D., *et al.*, 2004a. The gene ontology annotation (goa) database: Sharing knowledge in uniprot with gene ontology. *Nucleic Acids Research* 32 (suppl_1), D262–D266. Available at: <https://doi.org/10.1093/nar/gkh021>.
- Camon, E., Magrane, M., Barrell, D., *et al.*, 2004b. The Gene Ontology Annotation (GOA) Database: Sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Research* 32 (Database issue).
- Cannataro, M., Guzzi, P.H., Milano, M., 2015. God: An r-package based on ontologies for prioritization of genes with respect to diseases. *Journal of Computer Science* 9, 7–13. Available at: <https://doi.org/10.1016/j.jocs.2015.04.017>.
- Cannataro, M., Guzzi, P.H., Sarica, A., 2013. Data mining and life sciences applications on the grid. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3 (3), 216–238. Available at: <https://doi.org/10.1002/widm.1090>.
- Carmona-Saez, P., Chagoyen, M., Rodriguez, A., *et al.*, 2006. Integrated analysis of gene expression by association rules discovery. *BMC Bioinformatics* 7 (1), 54.
- Cho, Y.-R., Mina, M., Lu, Y., Kwon, N., Guzzi, P.H., 2013. M-Finder: Uncovering functionally associated proteins from interactome data integrated with GO annotations. *Proteome Science* 11 (1), 1.
- Faria, D., Schlicker, A., Pesquita, C., *et al.*, 2012. Mining go annotations for improving annotation consistency. [07]. *PLOS ONE* 7 (7), e40519.
- Guzzi, P., Mina, M., Guerra, C., Cannataro, M., 2012. Semantic similarity analysis of protein data: Assessment with biological features and issues. *Briefings in Bioinformatics* 13 (5), 569–585. Available at: <http://bib.oxfordjournals.org/content/early/2011/12/02/bib.bbr066.short>.
- Guzzi, P.H., Milano, M., Cannataro, M., 2014. Mining association rules from gene ontology and protein networks: Promises and challenges. *Procedia Computer Science* 29, 1970–1980.
- Harris, M.A., Clark, J., Ireland, A., Lomax, J., Ashburner, M., *et al.*, 2004. The gene ontology (go) database and informatics resource. *Nucleic Acids Research* 32 (Database issue), 258–261.
- Hipp, J., Guntzer, U., Nakhaeizadeh, G., 2000. Algorithms for association rule mining: a general survey and comparison. *ACM Sigkdd Explorations Newsletter* 2 (1), 58–64.
- Huttenhower, C., Hibbs, M.A., Myers, C.L., *et al.*, 2009. The impact of incomplete knowledge on evaluation: An experimental benchmark for protein function prediction. *Bioinformatics* 25 (18), 2404–2410.

- Manda, P., McCarthy, F., Bridges, S.M., 2013. Interestingness measures and strategies for mining multi-ontology multi-level association rules from gene ontology annotations for the discovery of new go relationships. *Journal of Biomedical Informatics* 46 (5), 849–856.
- Naulaerts, S., Meysman, P., Bittremieux, W., *et al.*, 2013. A primer to frequent itemset mining for bioinformatics. *Briefings in Bioinformatics*.
- Nguyen, C.D., Gardiner, K.J., Cios, K.J., 2011. Protein annotation from protein interaction networks and gene ontology. *Journal of Biomedical Informatics* 44 (5), 824–829.
- Ponzoni, I., Nueda, M.J., Tarazona, S., *et al.*, 2014. Pathway network inference from gene expression data. *BMC Systems Biology* 8 (2), 1–17.
- Schriml, L.M., Arze, C., Nadendla, S., *et al.*, 2012. Disease Ontology: A backbone for disease semantic integration. *Nucleic Acids Research* 40 (D1), D940–D946.
- Tew, C., Giraud-Carrier, C., Tanner, K., Burton, S., 2014. Behavior-based clustering and analysis of interestingness measures for association rule mining. *Data Mining and Knowledge Discovery* 28 (4), 1004–1045.
- Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W., *et al.*, 1997. New algorithms for fast discovery of association rules. *KDD*. Vol. 97, 283–286.

Relevant Websites

<http://disease-ontology.org>
Disease Ontology.

<http://www.ebi.ac.uk/QuickGO>
Gene Ontology and GO Annotations.

<http://www.ebi.ac.uk/QuickGO-Beta/webservices>
Gene Ontology and GO Annotations.

<http://geneontology.sourceforge.net>
The Gene Ontology.

<http://obo.sourceforge.net>
The OBO Foundry.

Biographical Sketch

Pietro H. Guzzi is an assistant professor of Computer Science Engineering at the University Magna Grecia of Catanzaro, Italy. His research interests comprise semantic-based and network-based analysis of biological and clinical data

Semantic Similarity Definition

Francisco M Couto and Andre Lamurias, Universidade de Lisboa, Lisboa, Portugal

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

CA	Common ancestors	$IC_{d_{shared}}$	Disjoint shared information content
CE	Common entries	IC_{shared}	Shared information content
DCA	Disjunctive common ancestors	KOS	Knowledge organization systems
F_D	Frequency in a external dataset D	MICA	Most informative common ancestors
IC	Information content	SB	Semantic-base
		SSM	Semantic similarity measure

Introduction

The biological role of an entity is considered to be its semantics, which has been increasingly being represented through common vocabularies. The entries in these vocabularies represent biological features, that are often connected with each other by semantic relations, such as subsumption. The availability of these common vocabularies, and their usage to semantically annotate entities enabled the development of computational semantic similarity measures (Batet *et al.*, 2014). Before defining semantic similarity, we should start by defining why bioinformatics needs semantic similarity in the first place, then what it is, to finally describe how it can be calculated.

Why?

Biomedical entities, such as proteins or chemical compounds, are frequently compared to each other to find similarities that may able us to transfer knowledge from one another. In the case of proteins, one of the most popular techniques is to calculate sequence similarity by locating short matches between sequences and then generate local alignments (Smith and Waterman, 1981). In the case of compounds, one of the most popular techniques is to calculate the number of 2D substructural fragments (molecular fingerprints) that they have in common (Willett, 2011). The above techniques are popular mainly because they can be implemented by high performance tools, such as BLAST (Altschul *et al.*, 1997), and are based on simple, unambiguous and widely available digital representations. However, these digital representations result from observations of how these biomedical entities look like, and not about their semantics. This means that from these digital representations we cannot have a direct insight about their biological role. Sequence similarity and common fingerprints measure how close two entities are in terms of what they look like, which may differ from their biological role.

There is an association between what an entity looks like and its biological role, i.e., proteins with similar sequence tend to have similar molecular functions, as well as with compounds with similar molecular shapes. However, there are many exceptions. For example, crystallins have a high sequence similarity to several different enzymes due to evolution, but in the eye lens their role is to act as structural proteins, not enzymes (Petsko and Ringe, 2004). Another example is caffeine and adenosine. These two molecules have a similar shape, so similar that caffeine is able to bind to adenosine receptors (Gupta and Gupta, 1999). However, adenosine induces sleep and suppresses arousal while caffeine makes you more awake and less tired. Semantic similarity addresses the above exceptions, by comparing biomedical entities based on what they do and not on what they look like. This means that when looking for similar compounds to caffeine, other central nervous system stimulants, such as doxapram, will appear before adenosine that has the opposite effect.

What?

Digital representations of biomedical entities based on structure can normally be expressed using a simple syntax. For example, ASCII strings are used to represent: the nucleotide sequences of genes; the amino acid sequences of proteins, and also the structure of compounds using SMILES. Semantics is however more complex since it may have different interpretations according to a given context. For example, the meaning of a biological role of a given gene may differ from a biological or medical perspective. For humans the easiest way to represent semantics is to use free text due to its flexibility to express any concept. For example, short text comments are usually valuable semantic descriptions to understand the meaning of a piece of information. However, for computers free text is not the most effective form of encoding semantics, making semantic similarity measurement between different text descriptions almost unfeasible.

In recent years, the biomedical community made a substantial effort in representing the semantics of biomedical entities by using common vocabularies, which vary from simple terminologies to highly complex semantic models. These vocabularies are instantiated by Knowledge Organization Systems (KOS) in the form of classification systems, thesauri, lexical databases, gazetteers, and taxonomies, and ontologies (Barros *et al.*, 2016). Perhaps the most well-known KOS is the Gene Ontology, which has been

extensively used to annotate gene-products with terms describing their molecular functions, biological processes and cellular components, and the source of most semantic similarity studies in bioinformatics. This manuscript will denote a KOS used in a semantic similarity measure as its Semantic-Base (SB). Semantic similarity measures become feasible when a biomedical community accepts a SB as a standard to represent the semantics of the entities in their domain. Semantic similarity is therefore a measure of how close are the semantic representations of different biomedical entities in a given SB. This means that the semantic similarity between two entities depends on their SB representation and also on a similarity measure that calculates how close these representations are in the SB.

How?

We may think that given a SB, we should be able to find the optimal quantitative function to implement semantic similarity. However, the notion of semantic similarity is dependent on what are the objectives of the study. For example, a biologist and a physician may have two different expectations about the semantic similarity between the biological roles of two genes.

In bioinformatics, ontologies have been the standard SB for calculating semantic similarity. The SB provides an unambiguous context on where semantic representations can be interpreted. A semantic representation is sometimes referred as a set of annotations, i.e., a link between the entity and an entry in the SB. Each entity can have multiple annotations. This means that the similarity measure may be applied for multiple entries in the SB. There are also different types of annotations. For example, an annotation can represent a finding with experimental evidence, or just a prediction from a computational method. Semantic similarity can explore the different types of annotations, for example to filter out annotations in which we have lower confidence.

A similarity measure is a quantitative function between entries in the SB, which explores the relations between its entries to measure their closeness in meaning. An entry is normally connected to the other entries by different types of relations represented in the SB. The similarity measure calculates the degree of shared meaning between two entries, resulting in a numerical value. For example, this can be performed by identifying a path between both entries in the SB, and calculating the semantic gap encoded in that path. This means that a semantic similarity measure can be defined by the SB and the quantitative measure used, which will be formulated in the following sections.

Semantic Base

Definition 1: (Semantic-Base). A Semantic-Base is a tuple $SB = \langle E, R \rangle$, such that E is the set of entries, and R is the set of relations between the entries. Each relation is pair of entities (e_1, e_2) with $e_1, e_2 \in E$.

When using biomedical ontologies, the entries represent the classes, terms or concepts. This definition ignores the type of relations that may be present in the ontology, since semantic similarity measures are normally restricted to subsumption relations (*is-a*). Nevertheless, a measure may use other type of relation, or even use different types of relations. The interpretation of its results should take this into consideration. One of the reasons why subsumption relations are used is because they are transitive, i.e. if $(e_1, e_2) \in R$ and $(e_2, e_3) \in R$ then we can implicitly assume that (e_1, e_3) is also a valid relation. This enables us to define the ancestors and descendants of a given entry.

Definition 2: (Ancestors). Given a SB represented by the tuple $\langle E, R \rangle$, and T the transitive closure of R on the set E (i.e. the smallest relation on E that contains R and is transitive), the Ancestors of a given entry $e \in E$ are defined as $Anc(e) = \{a : (e, a) \in T\}$.

Definition 3: (Descendants). Given a SB represented by the tuple $\langle E, R \rangle$, and T the transitive closure of R on the set E , the Descendants of a given entry $e \in E$ are defined as $Des(e) = \{d : (d, e) \in T\}$.

There are multiple successful semantic similarity measures being used in bioinformatics. Many of them are inspired on the contrast model proposed by Tversky (1977), in the sense that they balance the importance of common features versus the exclusives. Thus, a semantic similarity measure can be categorized by how it defines the common features, and how it calculates the importance of each feature. The first step in most measures is to find the common ancestors in the SB to define the common features.

Definition 4: (Common Ancestors). Given a SB represented by the tuple $\langle E, R \rangle$, the Common Ancestors of two entries $e_1, e_2 \in E$ is defined as $CA(e_1, e_2) = Anc(e_1) \cap Anc(e_2)$.

Information Content

This manuscript follows an information-theoretic perspective of semantic similarity (Sanchez and Batet, 2011). To calculate the importance of each entry the measures identify the information content of each entry. Resnik (1995) defined the information content of an entry based on the notion of the entropy of the random variable X known in information theory (Ross, 2009). The intuition is to measure the surprise evoked by having an entry $e \in E$ in the semantic representation.

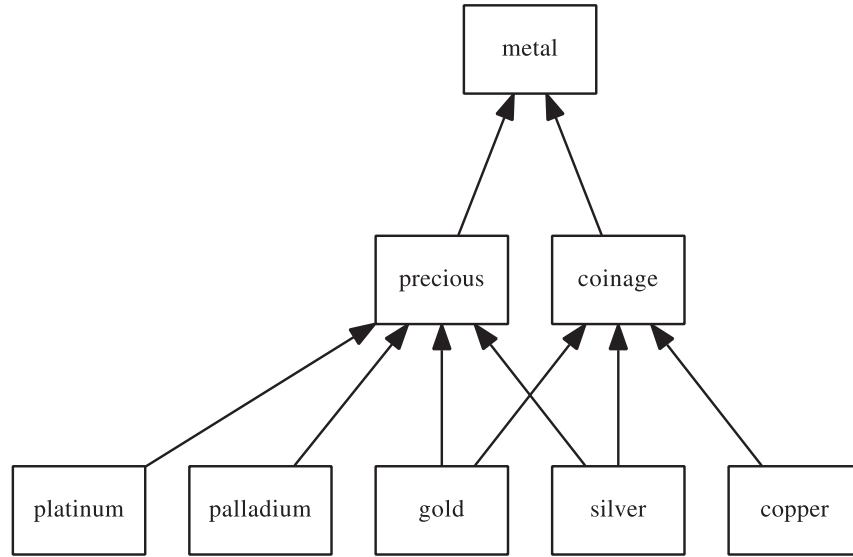


Fig. 1 This graph represents an example of a classification of metals with multiple inheritance, since *gold* and *silver* are considered both precious and coinage metals.

Definition 5: (Information Content). Given a SB represented by the tuple $\langle E, R \rangle$, and a probability function $p : E \rightarrow]0,1]$, the information content of an entry $e \in E$ is defined as $IC(e) = -\log(p(e))$.

The probability function should be defined in a way that bottom-level entries in the SB become more informative than top-level entries, making the $IC(e)$ correlated with the specificity of e in the SB.

The definition of the probability function p can follow two different approaches:

Intrinsic: p is based only on the internal structure of the SB.

Extrinsic: p is based on the frequency of each entry in an external dataset.

Considering the graph represented in **Fig. 1** as our SB, and assuming an intrinsic approach $p(e) = \frac{Desc(e)+1}{|E|}$, then we have all the bottom entries with p equal to $\frac{1}{8}$, $p(coinage) = \frac{4}{8}$, $p(precious) = \frac{5}{8}$, and $p(metal) = \frac{8}{8}$. Thus, we have $IC(metal) < IC(precious) < IC(coinage) < IC(platinum) \dots < IC(copper)$. Note also that the addition of 1 to avoid having a zero probability for the entries without descendants.

Definition 6: (Frequency). Given a SB represented by the tuple $\langle E, R \rangle$, and an external dataset D , and a predicate $refer(d, e)$ that is true when a data element $d \in D$ refers the entry $e \in E$, then the frequency of a given entry in that dataset is defined as

$$F_D(e) = |\{d : refer(e, d) \wedge d \in D \wedge e_1 \in Desc(e) \cup \{e\}\}|$$

Note that when using subsumption relations, i.e., an occurrence of an entry, it is also an implicit occurrence of all its ancestors.

Definition 7: (Extrinsic Probability). Given a SB represented by the tuple $\langle E, R \rangle$, and a frequency measure F_D the extrinsic probability function of an entry $e \in E$ is defined as

$$p(e) = \frac{F_D(e) + 1}{\max\{F_D(e_1) : e_1 \in E\} + 1}$$

Note that top-level entries have high frequency values due the occurrences of their descendants, so their IC is close to zero. Note again the addition of 1 this time in both parts of the fraction to avoid having a zero probability.

Considering again the graph represented in **Fig. 1** as our SB, and assume an external dataset D containing exactly one occurrence of each entry, then we have all the bottom entries with F_D equal to $\frac{2}{9}$, $F_D(coinage) = \frac{5}{9}$, $F_D(precious) = \frac{6}{9}$, and $F_D(metal) = \frac{9}{9}$. Thus, we again have $IC(metal) < IC(precious) < IC(coinage) < IC(platinum) \dots < IC(copper)$. We will assume this IC instantiation for the remainder examples in this manuscript.

Shared Ancestors

Not all ancestors are relevant when calculating semantic similarity since some of them are already subsumed by others and do not represent any new information. So normally the measures select only the most informative ones.

Definition 8: (Most Informative Common Ancestors). Given a SB represented by the tuple $\langle E, R \rangle$, and an IC measure, the Most Informative Common Ancestors of two entries $e_1, e_2 \in E$ is defined as

$$MICA(e_1, e_2) = \{a : a \in CA(e_1, e_2) \wedge IC(a) = \max\{IC(a_1) : a_1 \in CA(e_1, e_2)\}\}$$

Considering again the graph represented in [Fig. 1](#) as our SB, and the extrinsic IC defined above, then we have $MICA(palatium, copper) = \{metal\}$, $MICA(silver, gold) = \{coinage\}$, and $MICA(palatium, gold) = \{precious\}$.

Sometimes the most informative common ancestors are not sufficient, since they may neglect multiple inheritance relations. Thus, instead of MICA, the measures can use the disjunctive common ancestors ([Couto and Silva, 2011](#)).

Definition 9: (Disjunctive Common Ancestors). Given a SB represented by the tuple $\langle E, R \rangle$, and an IC measure, and a function $PD: E \times E \times E \rightarrow \mathbb{N}$, that calculates the difference between the number of paths from the two entries to one of their common ancestors, the Disjunctive Common Ancestors of two entries $e_1, e_2 \in E$ is defined as

$$\begin{aligned} DCA(e_1, e_2) = \{a : \\ a \in CA(e_1, e_2) \wedge \\ \forall a_x \in CA(e_1, e_2) PD(e_1, e_2, a) = PD(e_1, e_2, a_x) \\ \Rightarrow IC(a) > IC(a_x)\} \end{aligned}$$

Considering again the graph represented in [Fig. 1](#) as our SB, the extrinsic IC defined above, then we have $DCA(silver, gold) = \{coinage, precious\}$, and $DCA(platinum, gold) = \{precious, metal\}$.

Shared Information

The importance of common features is defined by the shared IC present in the common ancestors, normally its average.

Definition 10: (Shared Information Content). Given a SB represented by the tuple $\langle E, R \rangle$, and an IC measure, the Shared Information Content of two entries $e_1, e_2 \in E$ is defined as $IC_{shared}(e_1, e_2) = \overline{IC(a) : a \in DCA(e_1, e_2)}$.

Note that DCA can be replaced by MICA, however since all ancestors in MICA have the same IC value by definition only that IC value is used in practice.

Considering again the graph represented in [Fig. 1](#) as our SB, the extrinsic IC defined above, then when using MICA we have $IC_{shared}(platinum, gold) = -\log(\frac{6}{9})$. If we use DCA then we have $IC_{shared}(platinum, gold) = (-\log(\frac{6}{9}) - \log(\frac{2}{9}))/2$.

More recently, [Ferreira et al. \(2013\)](#) proposed the usage of the disjointness axioms in semantic similarity by defining the disjoint shared information content. The idea is that if we know that two entries are disjoint, then we should decrease their amount of shared information.

Definition 11: (Disjoint Shared Information Content). Given a SB represented by the tuple $\langle E, R \rangle$, a set of axioms A , and an IC_{shared} measure, the Disjoint Shared Information Content of two entries, $e_1, e_2 \in E$ is defined as $IC_{dshared}(e_1, e_2) = IC_{shared}(e_1, e_2) - k(e_1, e_2)$ with $k: E \times E \rightarrow \mathbb{N}$ satisfying the following conditions: i) $k(e_1, e_2) > 0$ if e_1 and e_2 are disjoint according to A ; ii) $k(e_1, e_2) = 0$ if otherwise.

Similarity Measure

Definition 12: (Semantic Similarity Measure). Given a SB represented by the tuple $\langle E, R \rangle$, a Semantic Similarity Measure is a quantitative function $SSM: E \times E \rightarrow \mathbb{R}$.

Note that a semantic similarity measure is not expected to be instantiated by the inverse of a metric or distance function, but the following conditions are normally satisfied:

non-negativity: $SSM(e_1, e_2) \geq 0$ with $e_1, e_2 \in E$;

symmetry: $SSM(e_1, e_2) = SSM(e_2, e_1)$ with $e_1, e_2 \in E$.

Many measures are also normalized, i.e., $SSM(e_1, e_2) \in [0..1]$ with $e_1, e_2 \in E$; and $SSM(e, e) = 1$ with $e \in E$.

The seminal work based on Resnik's measure ([Resnik, 1995](#)) was one of the first measures to be successfully applied to a biomedical ontology, namely the Gene Ontology ([Lord et al., 2003](#)). The measure was defined as:

$$SSM_{resnik}(e_1, e_2) = IC_{shared}(e_1, e_2)$$

Another well-known measure, was defined by Lin *et al.* (1998) as:

$$SSM_{lin}(e_1, e_2) = \frac{2 \times IC_{shared}(e_1, e_2)}{IC(e_1) + IC(e_2)}$$

where the denominator represents the exclusive features.

Note that both measures are independent of using *MICA* or *DCA* as the common features.

Considering again the graph represented in Fig. 1 as our SB, the extrinsic *IC* defined above, and *MICA*, then we have $SSM_{resnik}(platinum, gold) = -\log(\frac{6}{9})$ and $SSM_{lin}(platinum, gold) = (2 \times -\log(\frac{6}{9})) / (-\log(\frac{2}{9}) - \log(\frac{2}{9}))$.

Entity Similarity

Until now we only defined *SSM* in terms of entries, but a biomedical entity may not be directly represented in the SB, but instead linked to the SB through annotations. For example in the case of proteins, they are not represented as entries of the Gene Ontology but through annotations. In opposition, chemical compounds are represented as entries of the ontology Chemical Entities of Biological Interest (ChEBI).

Definition 13: (Annotation). Given a SB represented by the tuple $\langle E, R \rangle$ and a set of biomedical entities B , a predicate $annotates(b, e)$ that is true when the entity $b \in B$ is annotated with the entry $e \in E$, then the annotation set of a biomedical entity (or concept) $b \in B$ is defined as

$$AS(b) = \{e : e \in E \wedge annotates(b, e)\}$$

This definition ignores the type of annotation, e.g. with experimental or computational evidence, since the similarity measure calculation is usually independent of this information. It is up to the user to decide which type of annotations to include.

To compare biomedical entities we need to extend the *SSM* definition so it applies to the two sets of entries of each entity, instead of a single entry for each entity. For readability we will use the same function name *SSM*, to represent different functions according to the input domain, i.e. two entries or two sets of entries.

There are multiple successful instantiations of entity semantic similarity measures, and most of them use two aggregate functions (e.g., average, maximum) on the results from comparing each pair of entries annotated to each entry.

Definition 14: (Aggregate Measure). Given a SB represented by the tuple $\langle E, R \rangle$, a set of biomedical entities B , two aggregate functions f and g , and two biomedical entities $b_1, b_2 \in B$ the Aggregate Similarity Measure is defined as

$$SSM_{aggregate}(AS(b_1), AS(b_2)) = f(\{g(\{SSM(e_1, e_2) : e_1 \in AS(b_1)\}) : e_2 \in AS(b_2)\})$$

Considering again the graph represented in Fig. 1 as our SB, f as the average function, g as the maximum function, two entities containing metals $B = \{\alpha, \beta\}$, and their respective annotation set $AS(\alpha) = \{platinum, palladium\}$ $AS(\beta) = \{copper, gold\}$, then we have

$$\begin{aligned} SSM_{aggregate}(\{platinum, palladium\}, \{copper, gold\}) &= avg\{ \\ &\quad max\{SSM(platinum, copper), SSM(platinum, gold)\}, \\ &\quad max\{SSM(palladium, copper), SSM(palladium, gold)\}\} \end{aligned}$$

Another popular approach is to apply the Jaccard coefficient to all common entries vs. the exclusive ones.

Definition 15: (Jaccard Measure). Given a SB represented by the tuple $\langle E, R \rangle$, a set of biomedical entities B , an annotation set AT , and two biomedical entities $b_1, b_2 \in B$ the similarity measure is defined as

$$\begin{aligned} SSM_{jaccard}(AS(b_1), AS(b_2)) &= \\ &\frac{\sum \{IC(e) : e \in \{Anc(e_1) : e_1 \in AS(b_1)\} \cap \{Anc(e_2) : e_2 \in AS(b_2)\}\}}{\sum \{IC(e) : e \in \{Anc(e_1) : e_1 \in AS(b_1)\} \cup \{Anc(e_2) : e_2 \in AS(b_2)\}\}} \end{aligned}$$

Considering the example above of α and β when using Jaccard we will have

$$\begin{aligned} SSM_{jaccard}(\{platinum, palladium\}, \{copper, gold\}) &= \\ &\frac{IC(precious) + IC(metal)}{IC(coinage) + IC(precious) + IC(metal)} \end{aligned}$$

Future Directions

This manuscript is focused on defining semantic similarity using a single KOS, however a large amount of biomedical resources use multiple KOS describing a single domain from different perspectives or even distinct domains. Calculating semantic similarity

using multiple KOS as SB is a complex problem, and only a few works have addressed it (Solae-Ribalta *et al.*, 2014). Thus, a future formulation of multiple domain semantic similarity is much required.

Another issue is about the incompleteness of KOS. They normally represent work in progress, being updated as our knowledge of the domain becomes more sound and comprehensive. Keeping a KOS up-to-date is also a daunting task in terms of human effort, especially in large KOS, so we should always expect to have a delay until new knowledge is incorporated. This means that the common features identified in a KOS may be incomplete, and the exclusives features may not even be exclusive in the future. If a biomedical entity is not annotated with a specific feature, that does not mean that the entity does not have that feature, it only means that we do not know if it has or not. Thus, a future formulation of semantic similarity that takes in account the incompleteness of KOS is also much required.

Closing Remarks

This manuscript presented a definition of semantic similarity following an information-theoretic perspective that covers a large number of the measures currently being used in bioinformatics. It defined the amount of information content two entries share in a SB, and how it can be extended to compare biomedical entities represented outside the SB but linked through a set of annotations.

The manuscript aims at providing a generic and inclusive formulation that can be helpful to understand the fundamentals of semantic similarity and at the same time be used as a guideline to distinguish between different approaches. The formulation did not aim at providing a one size fits all definition, i.e. trying to represent all measures being proposed.

The manuscript presented well-known measures in bioinformatics, Resnik, Lin and Jaccard coefficient, according to the proposed definitions. It also presented their results when applied to simple example of a classification of metals, which is used along the text to clarify the definitions being presented. Finally, a software repository (see Relevant Website section) is available to test and learn more on how semantic similarity works in practice.

Acknowledgement

This work was supported by FCT through the PhD grant PD/BD/106083/2015 and LaSIGE Research Unit, ref. UID/CEC/00408/2013.

See also: Computational Tools for Structural Analysis of Proteins. Natural Language Processing Approaches in Bioinformatics. Protein Functional Annotation

References

- Altschul, S., Madden, T., Schäffer, A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25, 3389–3402.
- Barros, M., Couto, F.M., *et al.*, 2016. Knowledge representation and management: A linked data perspective. *IMIA Yearbook*. pp. 178–183.
- Batet, M., Harispe, S., Ranwez, S., Sanchez, D., Ranwez, V., 2014. An information theoretic approach to improve semantic similarity assessments across multiple ontologies. *Information Sciences* 283, 197–210.
- Couto, F., Silva, M., 2011. Disjunctive shared information between ontology concepts: Application to Gene Ontology. *Journal of Biomedical Semantics* 2, 5.
- Ferreira, J.D., Hastings, J., Couto, F.M., 2013. Exploiting disjointness axioms to improve semantic similarity measures. *Bioinformatics* 29, 2781–2787.
- Gupta, B.S., Gupta, U., 1999. Caffeine and Behavior: Current Views & Research Trends: Current Views and Research Trends. CRC Press.
- Lin, D., *et al.*, 1998. An information-theoretic definition of similarity. In: *ICML*. Citeseer. pp. 296–304.
- Lord, P., Stevens, R., Brass, A., Goble, C., 2003. Investigating semantic similarity measures across the Gene Ontology: The relationship between sequence and annotation. *Bioinformatics* 19, 1275–1283.
- Petsko, G.A., Ringe, D., 2004. *Protein Structure and Function*. New Science Press.
- Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.
- Ross, S., 2009. *A First Course in Probability* 8th Edition. Pearson.
- Sánchez, D., Batet, M., 2011. Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. *Journal of Biomedical Informatics* 44, 749–759.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* 147, 195–197.
- Solé-Ribalta, A., Sanchez, D., Batet, M., Serratos, F., 2014. Towards the estimation of feature-based semantic similarity using multiple ontologies. *Knowledge-Based Systems* 55, 101–113.
- Tversky, A., 1977. Features of similarity. *Psychological Review* 84, 327.
- Willett, P., 2011. Similarity searching using 2d structural fingerprints. *Chemoin Formatics and Computational Chemical Biology*. 133–158.

Further Reading

- Batet, M., Sánchez, D., 2015. A review on semantic similarity, *Encyclopedia of Information Science and Technology*, Third Edition IGI Global. pp. 7575–7583.
- Couto, F.M., Pinto, H.S., 2013. The next generation of similarity measures that fully explore the semantics in biomedical ontologies. *Journal of Bioinformatics and Computational Biology* 11, 1371001.

- Harispe, S., Ranwez, S., Janaqi, S., Montmain, J., 2015. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies* 8, 1–254.
- Pedersen, T., Pakhomov, S.V., Patwardhan, S., Chute, C.G., 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics* 40, 288–299.
- Pesquita, C., Faria, D., Falcao, A., Lord, P., Couto, F., 2009. Semantic similarity in biomedical ontologies. *PLOS Computational Biology* 5, e1000443.

Relevant Website

<https://github.com/lasigeBioTM/DiShIn/> and <http://labs.fc.ul.pt/dishin/>
GitHub.

Biographical Sketch

Francisco M. Couto is currently an associate professor with habilitation and vice-president of the Department of Informatics of FCUL, member of the coordination board of the master in Bioinformatics and Computational Biology, and a member of LASIGE coordinating the Health and Biomedical Informatics research line. He graduated (2000) and has a master (2001) in Informatics and Computer Engineering from the IST. He concluded his doctorate (2006) in Informatics, specialization Bioinformatics, from the Universidade de Lisboa. He was on the faculty at IST from 1998 to 2001 and since 2001 at FCUL. He was an invited researcher at EBI, AFMB-CNRS, BioAlma during his doctoral studies. In 2003, he was one of the first researchers to study semantic similarity based on information content in biomedical ontologies. In 2006, he also developed one of the first systems to use semantic similarity to enhance the performance of text mining solutions. In 2011, he proposed the notion of disjunctive common ancestors. In 2013, he participated in the development of the first similarity measure to exploit and demonstrate the usefulness of the description logic axioms in a biomedical ontology.

Andre Lamurias is a researcher at LaSIGE, currently enrolled in the BioSYS PhD programme at Universidade de Lisboa and holding a master's degree in Bioinformatics and Computational Biology from the same institution. His PhD thesis consists in developing text-mining approaches for disease network discovery and systems biology. More specifically, his research work is mainly focused on understanding how textual data from document repositories such as PubMed can be explored to improve our knowledge about biological systems.

Semantic Similarity Functions and Measures

Giuseppe Pirrò, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The origin of similarity studies stems from psychology and cognitive science where, during the years, different models have been postulated. The *geometric* model enables to assess similarity between entities by considering them as points in a dimensionally organized metric space. Similarity is given by the distance between objects in this space; the closer together two objects, the more similar. The *feature-based* model, relies on features (i.e., characteristics) of the examined objects and assumes that similarity is a function of both common and distinctive features (Tversky, 1977). Other techniques for assessing similarity, such as Information Content (Resnik, 1995), Pointwise Mutual Information (PMI), Information Retrieval (IR) (Turney, 2001; Pirrò and Talia, 2008) Normalized Google Distance (Cilibrasi and Vitanyi, 2007) and Latent Semantic Analysis (Landauer and Dumais, 1997), leverage probability co-occurrence of words in large text corpora such as the World Wide Web. Yet others rely on structured information (Pirrò, 2012).

Upon these models a plethora of computational methods has been developed, which includes very intuitive approaches where similarity is computed by counting the number of edges/nodes separating two words in a network structure (Rada et al., 1989) but also more sophisticated ones where similarity is assessed by projecting the feature-based model of similarity into the information-theoretic domain (Pirrò, 2009). For a more practical point of view, semantic similarity is relevant in many research areas in computer science and artificial intelligence (Rissland, 2006). For instance, in databases, it is used to estimate similarity between entities (Kashyap and Sheth, 1996). In information retrieval, it is used to improve accuracy of the vector-space model (Lee et al., 1993; Hliaoutakis et al., 2006). In the biomedical domain, there exist some applications to compute semantic similarity between concepts of ontologies such as Gene Consortium (2004) or MeSH (e.g., Pedersen et al., 2007; Lord et al., 2003) with the aim to assess, for instance, protein functional similarity. In Zhang et al. (2007), an approach for clustering research papers annotated with MeSH descriptors is presented. In Natural Language Processing, similarity is useful in many contexts such as word sense disambiguation (Leacock et al., 1998; Ravi and Rada, 2007). Word similarity has been also used to compute similarity between short sentences (Li et al., 2006), determine semantic orientation of adjectives (Hatzivassiloglou and McKeown, 1997) and classify reviews (Turney, 2002).

The Semantic Web is one of the most active community in which similarity is being used. Similarity helps to compute mappings between different ontologies (Pirrò et al., 2008; Atencia et al., 2011), repair ontology mappings (Meilicke et al., 2007) or compute similarity between whole ontologies (Araujo and Pinto, 2007). Similarity and Formal Concept Analysis can be exploited to structure a particular domain of interest useful in ontology development (Formica, 2008). Similarity measures have also been revisited in the context of Description Logics (DLs), which are expressive languages with a formal semantic (Borgida et al., 2005).

Similarity is such a versatile tool that it has found its way even in Peer to Peer networks where it can be exploited to build semantic overlay networks of peers. Concepts of a shared taxonomy, used to semantically annotate content, can be seen as indicators of the expertise of a peer. Semantic similarity allows to compute neighbours on a semantic basis, that is, by computing similarity among peer expertises. The neighbours to route a given message to can be chosen by computing the semantic similarity between concepts in a query and those reflecting neighbours' expertises (Hai and Hanhua, 2008; Pirrò et al., 2010; Penzo et al., 2008).

This paper is built around two main objectives. The first is to survey on computational methods to assess semantic similarity. This will hopefully pave the way toward a comprehensive overview on the most popular similarity measures. In particular, similarity measures exploiting different *sources of knowledge* in the process of likeness estimation will be reviewed. The second objective is to describe a concrete implementation of the analyzed measures in an off-the-shelf tool extensible and freely available. In this respect, although some initiatives exist, they only deal with similarity measures exploiting a single source of knowledge therefore lacking in integrating and providing a flexible environment to be easily embedded in concrete applications.

Knowledge-based similarity measures will be reviewed in Section "Knowledge-Based Approaches" whereas those relying on search engines as source of knowledge will be analyzed in Section "Search-Engine-Based Similarity Measures". A possible extension of similarity measures to compute similarity between sentences is described in Section "A Strategy to Compute Similarity between Sentences". The architecture of the *Similarity Library* will be sketched in Section "The Similarity Library" whereas in Section "Related Work" it will be compared with related initiatives. Some concluding remarks and possible ongoing work will be discussed in Section "Concluding Remarks and Future Work".

Similarity Measures Between Words

This section provides an overview on the similarity measures implemented in the *Similarity Library*. For each measure, the main characteristics will be described. As an example accompanying the presentation of the different measures, the excerpt of the Gene Ontology Consortium (2004) depicted in Fig. 1 will be used.

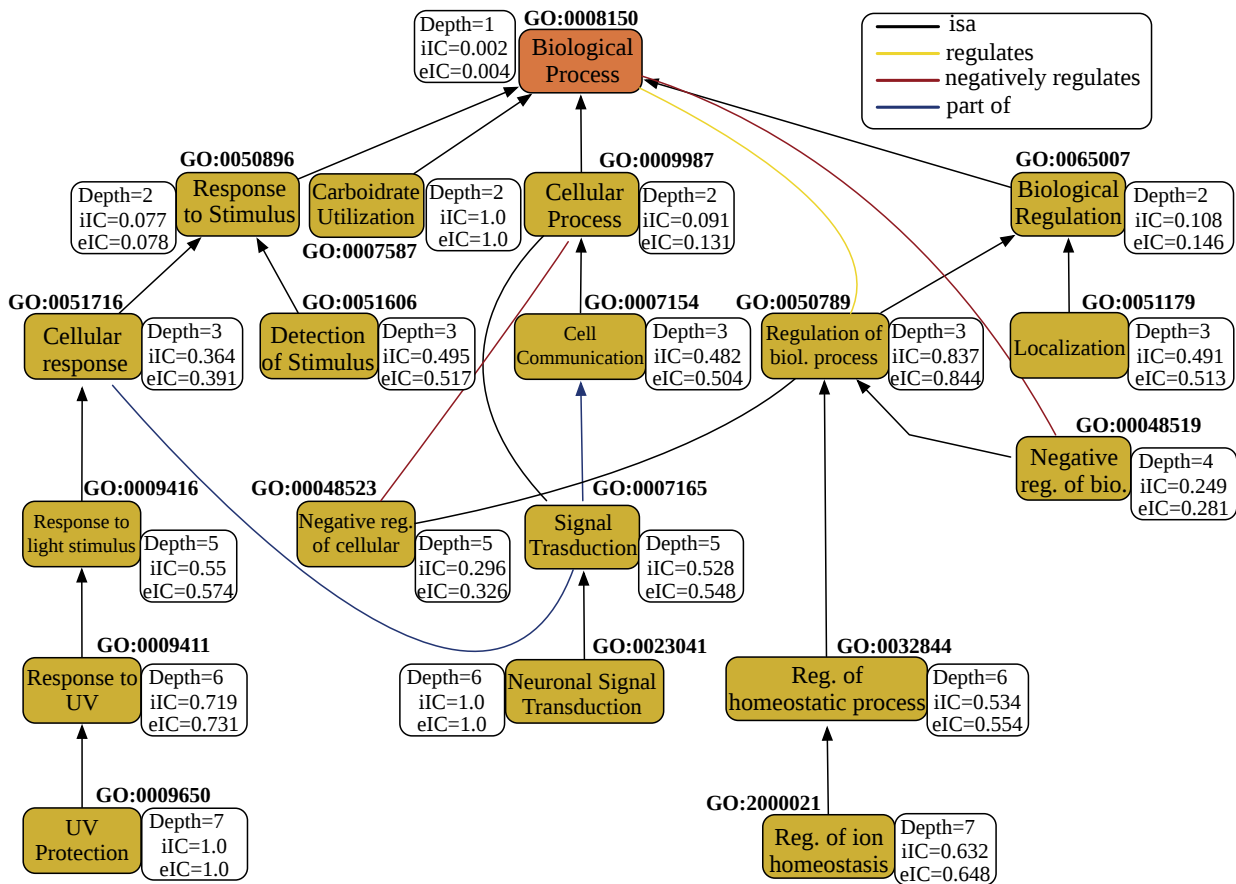


Fig. 1 An excerpt of ontology.

Knowledge-Base Approaches

Knowledge-based approaches can be divided in two main categories, that is, network-based approaches and information-theoretic approaches. In the following, network-based approaches will be reviewed.

Network-based approaches

Network-based approaches to similarity exploit some semantic artifacts closely akin to the semantic model devised by Quillian (1968), where concepts are represented by nodes and relations by links. Usually in semantic networks, concepts are stored within a hierarchical structure following the principle of *cognitive economy*. Cognitive economy refers to the fact that *properties* of concepts are stored at the highest possible level in the hierarchy and not re-represented at lower level.

An important definition when working with semantic networks is that of most specific common abstraction (MSOA), which is the most specific ancestor concept subsuming two concepts in the structure. As an example, in Fig. 1 the msca between *Cellular Response* and *Detection of Stimulus* is *Response to stimulus*. However, in some cases two concepts can have more than one common ancestor since it is common for semantic networks to be modeled as a Directed Acyclic Graph (DAG).

The Rada measure

The work by Rada et al. (1989) computes distance between concepts in a hierarchical structure by considering the number of links separating them. In particular, the lower the number of links the more similar the two concepts. The next equation represents this measure.

$$D_{rada}(c_1, c_2) = \min_{edge}(c_1, c_2) \quad (1)$$

For instance, in Fig. 1, the distance between the concepts *UV Protection* and *Detection of Stimulus* is 5 whereas that between *Cellular Response* and *Detection of Stimulus* is 1, which indicates that *Cellular Response* is more similar to *Detection of Stimulus* than to *UV Protection*.

The Leacock and Chodorow measure

The Leacock and Chodorow measure (L&C) Leacock and Chodorow (1998) assesses similarity by considering the shortest path between the two concepts being compared (using node counting) along with the depth of the taxonomy. The following formula

reflects this measure.

$$sim_{L\&C}(c_1, c_2) = -\log \frac{\min_{node}(c_1, c_2)}{2 \cdot Depth} \quad (2)$$

It can be noted that even if this measure takes into account the depth, it only considered the maximum depth in the taxonomy and not the depth of the concepts being compared. Hence, the couples *Cellular Response- Detection of Stimulus* and *Regulation of bio. process-Localization* will be rated at the same similarity level.

The Wu and Palmer measure

The Wu & Palmer measure (W&P) [Wu and Palmer \(1994\)](#) calculates similarity by considering the depths of the two nodes in the taxonomy, along with the depth of the msca. The following formula reflects this measure.

$$sim_{W\&P}(c_1, c_2) = \frac{-2 \cdot Depth(msca(c_1, c_2))}{Depth(c_1) + Depth(c_2)} \quad (3)$$

Returning to our previous example, with this measure the similarity between *Cellular Response* and *Detection of Stimulus* is 0.66.

Information Theoretic Approaches

Information-theoretic approaches provide an alternative way to compute similarity in a semantic network. Here, the notion of Information Content (IC), which quantifies the informativeness of concepts, is adopted. IC values are obtained by associating probabilities to each concept on the basis of its occurrences in large text corpora. In the specific case of hierarchical structures, probabilities are cumulative as we travel up from specific concepts to more abstract ones. This means that every occurrence of a concept in a given corpus is also counted as an occurrence of each concept containing it. IC values are obtained by computing the negative likelihood of encountering a concept in a given corpus as reported in the following formula.

$$IC(c) = -\log p(c) \quad (4)$$

where c is a concept and $p(c)$ is the probability of encountering c in a given corpus. Note that in the case of hierarchical structures this method ensures that IC is monotonically decreasing as we move from the leaves of the structure to its roots. In fact, the concept corresponding to the root node of the isa hierarchy has the maximum frequency count, since it includes the frequency counts of every other concept in the hierarchy. For instance, in [Fig. 1](#) it can be noted that the IC (In [Fig. 1](#) values of *intrinsic* and *extended* IC are reported. These two notions will be discussed in Sections “Intrinsic information content” and “Extended information content”) of the concept *Biological Process* is almost equal to zero since it is a very abstract concept (the top of the hierarchy), whereas that of more specific concepts such as *Signal Transduction* or *Localization* is higher. Note also that leaves have IC equal to 1 as they represent completely described concepts (no children).

The Resnik measure

[Resnik \(1995\)](#) was the first to leverage IC values for the purpose of semantic similarity. The basic intuition behind the use of the negative likelihood is that the more probable a concept is of appearing in a corpus the less information it conveys, in other words, infrequent words are more informative than frequent ones. Once IC values are available for each concept in the considered ontology, semantic similarity can be calculated. According to Resnik, the similarity depends on the amount of information two concepts have in common. This shared information is given by the msca, that is, their subsumer. As an example, in [Fig. 1](#) the concept *Regulation of Biological Process* subsumes both *Negative Regulation of Cellular Process* and *Regulation of Homeostatic Process*. In order to find a quantitative value of the shared information we must first discover the msca. If one does not exist then the two concepts are maximally dissimilar, otherwise the shared information is equal to the IC value of their msca. Resnik’s formula is modeled as follows:

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} IC(c) \quad (5)$$

where $S(c_1, c_2)$ is the set of concepts that subsume c_1 and c_2 . Returning to our previous example, the similarity between the couples *Carbohydrate Utilization-Cellular Process* and *Localization-Regulation of Biological Process* are 0.002 and 0.1088 respectively. Starting from Resnik’s work, the [Jiang and Conrath \(1997\)](#) and the [Lin \(1998\)](#) similarity measures, which take into account the IC of the two concepts as well, have been proposed.

The Jiang and Conrad measure

The Jiang and Conrad measure (J&C) [Jiang and Conrath \(1997\)](#) is a semantic distance measure and is derived from the edge-based notion of distance with the addition of the IC as a decision factor ([Jiang and Conrath, 1997](#)). The next equation represents this measure.

$$D_{J\&C}(c_1, c_2) = 2 \cdot IC(msca(c_1, c_2)) - (IC(c_1) + IC(c_2)) \quad (6)$$

Returning to our example, according to the J&C measure the distances between the couples *Signal Transduction-Regulation of Homeostatic Process* and *Response to UV-Negative Regulation of Cellular Process* are 0.603 and 0.516 respectively. Note that in this case the lower the distance the higher the similarity. The J&C measure solves the problem faced with Resnik’s measure with which every two couples having the same msca have the same similarity.

The Lin measure

According to Lin “The similarity between c_1 and c_2 is measured by the ratio between the amount of information needed to state the commonality of c_1 and c_2 and the information needed to fully describe what c_1 and c_2 are”. Formally this formula is given in the following equation:

$$sim_{Lin}(c_1, c_2) = \frac{2 \cdot sim_{res}(c_1, c_2)}{IC(c_1) + IC(c_2)} \quad (7)$$

The Lin measure represents another variant of Resnik’s measure that takes into account the IC of the two concepts being compared. This measure of similarity has been designed according to the following principles: i) The similarity between c_1 and c_2 is related to their commonality; ii) the similarity between c_1 and c_2 is related to the differences between them and iii) the maximum similarity between c_1 and c_2 is reached when c_1 and c_2 are identical. The Lin measure returns 0.71 and 0.838 as values of similarity for the couples *Signal Transduction-Regulation of Homeostatic Process* and *Response to UV-Negative regulation of Cellular Process*, respectively.

Intrinsic information content

The classical way of measuring information content of concepts is based on the idea of combining knowledge of the hierarchical structure in which they are defined with statistics on their actual usage in text derived from a large corpus. The intuitive motivation behind this type of reasoning is that rare concepts are more specific and therefore much more expressive. However, from a practical point of view, this approach has two main drawbacks:

1. It is time consuming since it implies that large corpora should be parsed and analyzed. Moreover, the considered corpora have to be adequate in term of content to the considered semantic structure,
2. It heavily depends on the type of corpora considered and its size. It is arguable that IC values obtained from very general corpora maybe different than those obtained with more specialized ones. As for WordNet, the Brown corpus is typically exploited, which is a good source of general knowledge. However, if we were calculating similarity between concepts of very specialized ontologies such as MeSH, it is likely that Brown corpus does not contain many of the terms included in that ontology and then IC values and corresponding similarity assessments could be affected.

Research toward mitigating these drawbacks has been proposed by [Seco et al. \(2004\)](#). Here, values of IC of concepts rest on the assumption that the taxonomic structure of the ontology is organized in a meaningful and structured way, where concepts with many hyponyms (i.e., sub-concepts) convey less information than concepts that have a lower number of hyponyms. With this reasoning leaves are maximally informative. The intuition is: More abstract concepts are more probable of being present in a corpus because they subsume so many other ones. If a concept has many hyponyms, then it has more of a chance of appearing since the subsuming concept is implicitly present when reference to one of its hyponyms is made. So if one wanted to calculate the probability of a concept it would be the number of hyponyms it has plus one (for itself) divided by the total number of concepts that exist. The *intrinsic* IC for a concept c is defined as:

$$iIC(c) = 1 - \frac{\log(hypo(c) + 1)}{\log(max_{con})} \quad (8)$$

where the function *hypo* returns the number of hyponyms of a given concept c . Moreover max_{con} is a constant that indicates the total number of concepts in the considered structure. In [Fig. 1](#), values of *iIC* are reported for each concept. As it can be noted, concepts higher in the hierarchy have lower *iIC* values. Moreover, the more hyponyms a concept has the lower is its *iIC*. For instance, *Cellular Process* has an *iIC* equals to 0.091 whereas *Biological Regulation* has a higher *iIC* (i.e., 0.108).

The intrinsic IC formulation is based on the assumption that the ontology is organized according to the principle of *cognitive saliency* ([Zavaracky, 2003](#)). Cognitive saliency states that humans create concepts when there is a need to differentiate from what already exists. As an example, in WordNet the concept *cable car* only exists because its lexicographers agree that *cable* is a sufficiently salient feature (This feature indicates that the *cable car* operates on a cableway or cable railway as the WordNet gloss mentions.) allowed it to be differentiated from *car* and promoted to a concept in its own right. Obviously, what is cognitively salient to one community may not be to another and consequently these communities will have different similarity judgments. Hierarchies are usually created trying to cover lexical concepts following principles of general knowledge ([Seco, 2005](#)). The *Similarity Library* exploits this formulation for all the IC-based measures.

Extended information content

The intrinsic IC described above, results very useful in determining to what extent two concepts are related by subsumption relations. However, an ontology usually contains relations beyond inheritance that can be very useful to better refine the extent to what two concepts are alike. With this reasoning in mind, the Extended Information Content (*eIC*) has been introduced ([Pir  and Euzenat, 2010a](#)). The *eIC* is a score assigned to each concept computed by investigating the different kinds of relations a given concept has with other concepts. For instance, by only focusing on isa relations, in the example in [Fig. 1](#) we would lose some important information (e.g., that *Signal Transduction* is part-of *Cell Communication* or that *Signal Transduction* is part-of *Cellular Response*) that can help to further characterize commonalities and differences between concepts. For each concept, the coefficient

EIC is defined as follows:

$$EIC(c) = \sum_{j=1}^m \frac{\sum_{k=1}^n iIC(c_k \in C_{R_j})}{|C_{R_j}|} \quad (9)$$

This formula takes into account all the m kinds of relations that connect a given concept c with other concepts. Moreover, for all the concepts at the other end of a particular relation (i.e., each $c_k \in C_{R_j}$) the average iIC is computed. This enables to take into account the expressiveness of concepts to which a given concept is related in terms of their information content. The final value of *Extended Information Content* (eIC) is computed by weighting the contribution of the iIC and EIC coefficients thus leading to:

$$eIC(c) = \zeta iIC(c) + \eta EIC(c) \quad (10)$$

Where the two parameters ζ and η are used to weight the contribution of isa relations and other kinds of semantic relations. In Fig. 1, the eIC of the different concepts is reported. As it can be noted, in many cases the value is different from the iIC since in computing eIC all the semantic relations between concepts are considered. This score offers the same nice properties as the iIC as it can be directly extracted from the ontological structure. The *Similarity Library* also provides eIC values for each concept.

Hybrid approaches

Hybrid approaches usually combine multiple information sources or different models to assess similarity.

The Li measure

An approach combining structural semantic information in a nonlinear model is that proposed by Li *et al.* (2003). This work borrows some ideas from Shepard's law of generalization, which states that similarity decreases with distance following an exponential function Shepard (1987). Li *et al.* empirically defined a similarity measure that uses shortest path length, depth and local density in a taxonomy. If the two concepts are the same the similarity is equal to 1 else it is given by the formula reported in the following equation:

$$sim_{Li}(c_1, c_2) = e^{-\alpha l} \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \quad \text{if } c_1 \neq c_2 \quad (11)$$

Where l is the length of the shortest path between c_1 and c_2 in the graph spanned by the isa relation and h is the level in the tree of the msca from c_1 and c_2 . The parameters α and β represent the contribution of the shortest path length l and depth h . The optimal values for these parameters, determined experimentally, are: $\alpha=0.2$ and $\beta=0.6$ as discussed in Li *et al.* (2003).

Returning to our previous example, the similarity between *Detection of Stimulus* and *Cell Communication* is 0.67. However it should be noted that this measure requires the tuning of two parameters.

Features and information content for similarity

This category of similarity measures, combines two models of similarity, that is, features and information content. The main idea is to exploit the IC formulation of similarity, in which the msca of two concepts c_1 and c_2 reflects the information these concepts share, and to extend it for considering also specific concept features (refer to Pirrò, 2009 for a detailed discussion on the feature-based model of similarity). Recently, a similarity measure combining features and information content that adopts Tversky's contrast model has been defined (Pirrò, 2009). The next equation reflects this measure.

$$sim_{P\&S}(c_1, c_2) = 3 \cdot IC(msca(c_1, c_2)) - IC(c_1) - IC(c_2) \quad (12)$$

In the case in which $c_1 = c_2$, the measure returns 1 as similarity value. This measure treats similarity between identical concepts as a special case and can give as output negative values, which makes it difficult the interpretation of results. Therefore, a new similarity measure, called *FaITH*, based on the Tversky ratio-model, has been defined in Pirrò and Euzenat (2010a). The next equation reports this measure.

$$sim_{FaITH}(c_1, c_2) = \frac{IC(msca)}{IC(c_1) + IC(c_2) - IC(msca)} \quad (13)$$

Similarity values returned by the P&S measure for the couples *Signal Transduction-Regulation of Homeostatic Process* is 0.456 whereas with FaITH the similarity value is 0.366. The main contribution of these two measures is that they enable to project the well-known feature-based similarity model into the information content domain and then to provide an alternative way to treat concept features.

Search-Engine-Based Similarity Measures

Search engines are without any doubt a huge source of knowledge that can be exploited in the process of similarity estimation. Differently from semantic-network based approaches, similarity measures exploiting search engines have to deal with unstructured knowledge. Even if there exist a huge amount of literature concerning distributional approaches to semantic similarity such as Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997), here we will focus on more practical approaches for two main reasons. First, methods such as LSA require a corpus of content to be indexed on which applying the *Singular Valued Decomposition*

method. Since the *Similarity Library* is not tied to an specific corpus or application, implementing the LSA approach would make the library losing its generality. Second, the aim of the *Similarity Library* is to provide a scalable and portable tool to be used even in large scale applications and therefore, it has to be simple and immediately usable. Implementing LSA, on the other hand, would require to use large information repositories.

In the remainder of this section, two strategies implemented in the *Similarity Library* that make usage of search engines to estimate similarity will be reviewed.

The PMI-IR measure

PMI-IR [Turney \(2001\)](#) is a unsupervised learning algorithm for recognizing synonyms, based on statistical data acquired by querying a Web search engine. PMI-IR uses Pointwise Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of pairs of words. PMI-IR has been designed with the aim of identifying synonyms. In particular, PMI-IR has been evaluated on TOEFL synonymy tests where, given a base word (called *problem*), the goal is that of finding the more similar word among four choices (each of which is referred to as a *choice*). In the general case, PMI-IR computes its score according to the following formula:

$$PMI - IR_{score}(c_1, c_2) = \frac{p(c_1 AND c_2)}{p(c_2)} \quad (14)$$

where the function $p(\cdot)$ returns the probability of the problem/choice and it is computed by counting the number of query hits returned by a search engine (i.e., AltaVista in the original formulation). The author starting from this basic formula defined four different scores of increasing complexity. The first score assesses word similarity by focusing on the simple co-occurrence of words. The second score looks at words that co-occur in the same document and are close together. The third score faces the problem of antonym (i.e., opposite) noun scoring. Finally, the fourth score takes context into account.

As an example let's consider the following TOEFL question: Given the four words *thief*, *traitor*, *shark* and *discourage* which of these words is the most similar in meaning to the word *turncoat*? In the following table the steps performed by PMI-IR for answering this question are reported ([Table 1](#)).

As it can be noted, PMI-IR $score_1$ returns the higher score for the word *traitor*, which is indeed the most similar word to *turncoat*. This approach has been extensively compared with latent Semantic Analysis (LSA) in [Turney \(2001\)](#) and an improvement in performance in the same task of synonymy recognition has been reported.

The NGD measure

Normalized Google Distance (NGD) [Cilibrasi and Vitanyi \(2007\)](#) distance is a measure of semantic interrelatedness derived from the number of hits returned by the Google search engine for a given set of keywords. Keywords with the same or similar meanings in a natural language sense tend to be "close" in units of Google distance, while words with dissimilar meanings tend to be distant. In more detail the NGD between c_1 and c_2 is defined as follows:

$$NGD(c_1, c_2) = \frac{\max\{\log f(c_1), \log f(c_2) - \log f(c_1, c_2)\}}{\log N - \min\{\log f(c_1) - \log f(c_2)\}} \quad (15)$$

where $f(c)$ denotes the number of pages containing c , and $f(c_1, c_2)$ denotes the number of pages containing both c_1 and c_2 while N the number of pages indexed by the search engine. Returning to the example presented in the previous section, NGD returns the scores reported in [Table 2](#).

As it can be noted, even the NGD choices the correct alternative among the four words.

A Strategy to Compute Similarity Between Sentences

The similarity measures described in Section "Similarity Measures between Words" can be adopted to compute semantic similarity between sentences. Sentence similarity, is very useful in many research areas such as text summarization. However, most of current approaches exploit the classical vector space model ([Salton et al., 1975](#)), which fails when there are sentences semantically similar even if syntactically different as in the case, for instance, of the two sentences *I own a car* and *I have an automobile*. Therefore, by leveraging measures of semantic similarity, the aim is as follows: given two input sentences, calculate their similarity at semantic level. In particular, similarity between single words, is extended in the case of multiple words given that, the similarity between two

Table 1 An example of PMI-IR computation by using Google

Query	Hits X and Y	Hits Y	Score ₁
turncoat AND thief	2360	1,770,000	0.0013
turncoat AND traitor	5590	1,880,000	0.0029
turncoat AND shark	2280	2,740,000	$8 \cdot 10^{-4}$
turncoat AND discourage	1290	4,330,000	$2.97 \cdot 10^{-4}$

Table 2 An example of NGD computation by using Google

<i>Couple</i>	<i>NGD</i>
turncoat-thief	0.340
turncoat-traitor	0.4219
turncoat-shark	0.296
turncoat-discourage	0.1913

sentences can be seen as a function of the similarity of their component words. The approach described here is based on the following observations:

- Current approaches to compute sentence similarity using knowledge-based methods (e.g., Mihalcea *et al.*, 2006; Li *et al.*, 2006) only compute similarity between nouns in the sentences underestimating the importance of other parts of speech (e.g., adjectives and adverbs). To cope with this aspect the *Similarity Library* implements strategies to compute similarity also between verbs, adjectives and adverbs, respectively (refer to Pirrò and Euzenat, 2010b for more details).
- In a sentences there are some words that can contribute to a greater extent to the similarity of the sentences. For instance, specific words should have a greater impact than more general ones.

Let consider the following two sentences as example:

- Sentence 1 (s_1): *When the team and his trainer entered the city, some of the fans started to blame the latter.*
- Sentence 2 (s_2): *When the players returned into the city with the trainer, some supporters begun to yell at the latter.*

The general approach, adopted to compute sentence similarity, is inspired by the maximum weighted matching problem in a bipartite graph and consists in comparing all the words in two sentences to find their best coupling. In the maximum weighted matching problem, a non-negative weight $w_{i,j}$ is assigned to each edge $e(x_i, y_j)$ and the aim is to seek a perfect matching M to maximize the total weight $w(M)$. However, couples of nodes that are not matched should not participate in the final similarity calculation.

The process starts by construing a similarity matrix which has as rows words in the first sentence, and as columns words in the second sentence. In particular, to each element (representing a pair of words) is assigned a score, which is the similarity between the words. The score can be computed using one, or a combination, of similarity measures described in Section “Similarity Measures between Words”.

This matrix is fed in input at the *Hungarian* algorithm, which gives as output the best coupling of elements in rows and columns. After having obtained the best pairing, in order to give more emphasis to specific words and less to more general ones, the following coefficient of specificity is adopted:

$$spec_{e(i,j)} = \frac{IC(w_i) + IC(w_j)}{2} \quad (16)$$

This coefficient takes into account the average IC of the two words that have been matched (refer to Pirrò and Euzenat, 2010b for more details). Since the IC is an indicator of the informativeness of words, a pair with a higher *spec* is more specific and then can be more significant in estimating sentence similarity. At this point, we can compute sentence similarity as follows:

$$sem_{SEN}(s_1, s_2) = \sum_{i \in S_1} \sum_{j \in S_2} \frac{e(i,j) * spec_{e(i,j)}}{K} \quad (17)$$

where each $e(i,j)$ is the best weight for a couple of words $w_i \in S_1$ and $w_j \in S_2$ and K is the number of pairs with $e(i,j) > 0$. Recalling our previous example, Fig. 2 shows the pair matching found by the algorithm. The similarity between s_1 and s_2 is therefore $sem_{SEN}(s_1, s_2) = 0.7113$ by using the P&S measure. The result obtained, for the same sentence by PMI-IR and NGD are 0.641 and 0.643 respectively.

The Similarity Library

This section describes the implementation of a similarity library gathering the techniques reviewed in this paper. The library is available at (see “Relevant Websites section”).

Architecture

The *Similarity Library* is composed by a set of software modules implemented in Java. Table 3 summarizes the similarity measures implemented.

The overall architecture is depicted in Fig. 3. As it can be noted the architecture is built upon three layers. The *Data Layer* interacts with the different sources of knowledge that can be used to estimate similarity. The following table details the version of

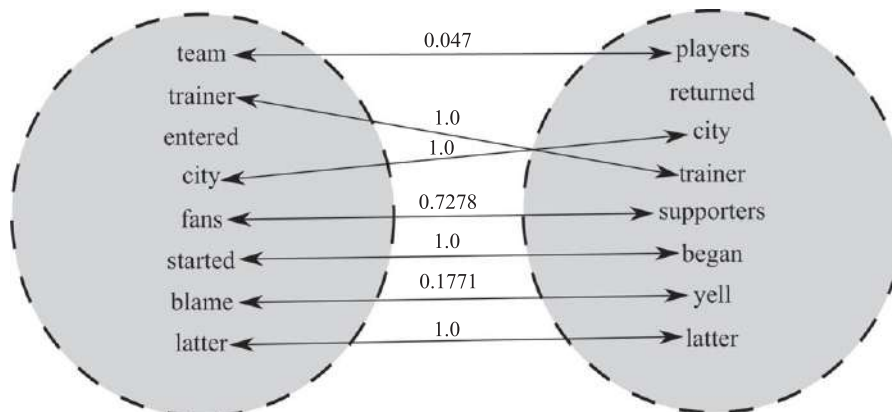


Fig. 2 Example of sentence pair matching.

Table 3 Measures implemented by the *Similarity Library*

	Information exploited				Specific techniques
	Depth	Length	IC	SE	
Rada <i>et al.</i> (1989)		X			Shortest path counting edges
L&C Leacock and Chodorow (1998)	X				Path counting nodes and depth of the hierarchy
W&P Wu and Palmer (1994)	X				Depth of concepts and msca
Li <i>et al.</i> (2003)	X	X			Hybrid measure
Resnik (1995)			X		IC of the msca
Lin (1998)			X		IC of the msca and concepts
J&C Jiang and Conrath (1997)			X		IC of the msca and concepts
P&S Pirrò and Seco (2008); Pirrò (2009)			X		Mapping of features into IC
FaITH Pirrò and Euzenat (2010a)			X		Mapping of features into IC
PMI-IR Turney (2001)				X	Search engines hits
NGD Cilibiasi and Vitanyi (2007)				X	Search engines hits

the knowledge sources used in the current release of the library. The various knowledge sources can be easily updated since the library provides the proper parsers (Table 4).

The *Wrapping Layer* receives the information parsed by the *Data Layer* and transforms it into an internal model suitable for indexing. In more detail each data item (e.g., ontology concept) is assigned a Lucene (see “Relevant Websites section”) document composed by different fields. Subsequently, all the documents are stored in a disk index. This will be useful to speed up the lookup process needed during the similarity estimation. This will also enable an easier use of the library as compared to other initiatives (e.g., WordNet::Similarity Pedersen *et al.*, 2004) that require a relational databases and other software to be installed.

Finally, the *Similarity Layer* takes care of computing the similarity between the words or sentences passed in input. This is the highest layer of the architecture and represents the interface by which an application requires a similarity computation and receives the answer.

Software Architecture

This section provides some details about the classes implemented in the *Similarity Library* with the aim to give a hint on its internal software architecture. This will help in extending it with both new word and sentence similarity measures. Fig. 4 shows how the classes involved in the process of sentence similarity computation interact. The *Sentence Similarity Measure* interface provides basic definitions of methods that can be implemented by any sentence similarity strategy.

Common methods to any implementation are provided in the *Abstract Sentence Similarity Measure* class. In particular this class uses the *Sentence Utilities* class that provides methods to perform some preprocessing on the sentences being compared (e.g., tagging). Moreover, this class also uses the *Word Similarity Calculator* class that provides similarity scores for couples of words according to different similarity strategies such as search engine, by using the *Search Engine* assessor class, or WordNet, by using the *Word Net Assessor* class.

Fig. 5 details the structure of classes involved in the process of similarity estimation for knowledge-based approaches. Even in this case, an interface (i.e., *Similarity Measure*) provides some guidelines on the methods to be implemented by either IC-based and

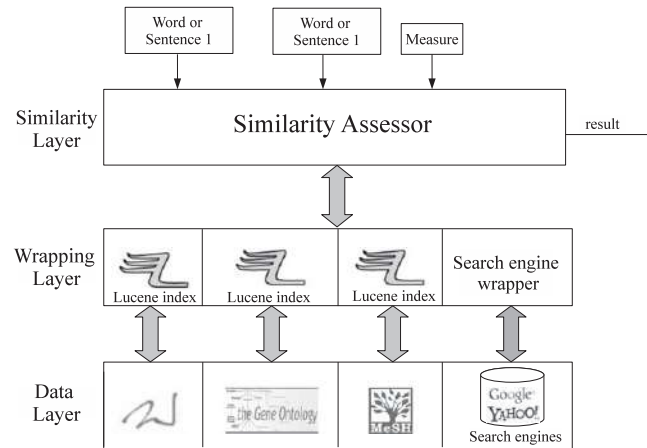


Fig. 3 Architecture of the similarity library.

Table 4 Current version of the knowledge sources in the *Similarity Library*

Knowledge source	Version
WordNet	3.0
MeSH	2009
Gene Ontology	OBO 1.12

path-based approaches. Common methods useful to any implementation are implemented by the two abstract classes *Abstract ICE*

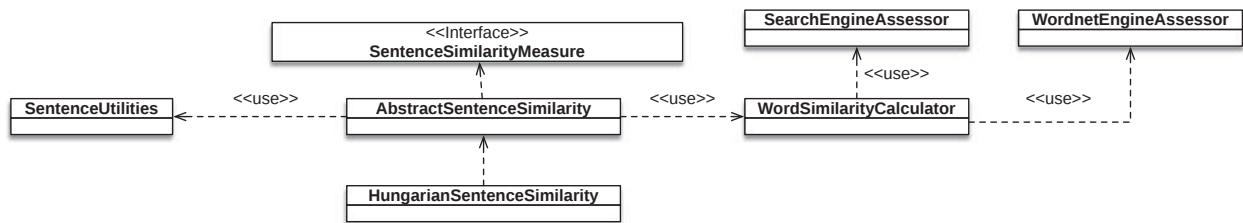


Fig. 4 Classes involved in sentence similarity.

Similarity Measure and *Abstract Topological Similarity Measure*. Note that both classes use the *Abstract Index Broker* class, which provides access to the different ontologies stored in the corresponding Lucene indexes. Upon these two abstract classes, six additional classes provide similarity computation for each of the considered categories and ontologies. Finally, for each ontology a corresponding assessor class encompasses both the IC-based and path-based approaches.

In Fig. 6 it is reported the class diagram for search-engine based approaches. As in the case of knowledge-based approaches, an interface is defined (i.e., *Search Engine Similarity*) to provide some guidelines for concrete implementations. The *Abstract Search Engine Similarity Measure* provides common functionalities shared by the two approaches implemented and described in Section "Search-Engine-Based Similarity Measures". Note that the library provides implementations relying on two search engines (i.e., Google and Yahoo). The *Search Engine Assessor* provides a common point for using the different measures. Finally, Fig. 7 shows the main classes involved in the indexing process. As it can be noted, there is an interface and an abstract class along with different concrete implementations. In particular, for the WordNet ontology, four different indexes are constructed, one for nouns, one for verbs, one for adjectives and one for adverbs. Two additional indexes, one for the Gene Ontology and one for MeSH are also constructed.

Related Work

There exist some work in the literature related to the *Similarity Library*. The WordNet::Similarity library described in Pedersen et al. (2004) is implemented in Python and aims at computing relatedness between concepts. An online interface is made available (see "Relevant Websites section"), which enables to compute the similarity between words and to choose the strategy to be used. It is also possible to locally install this library after installing Python along with a relational database.

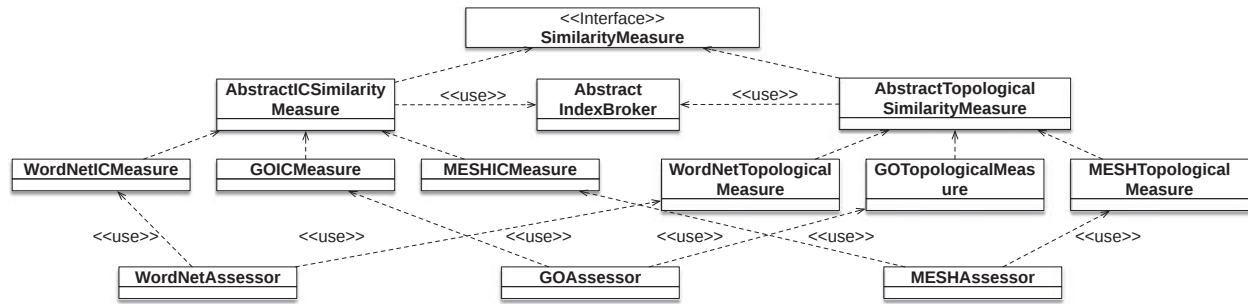


Fig. 5 Classes involved in knowledge-based word similarity.

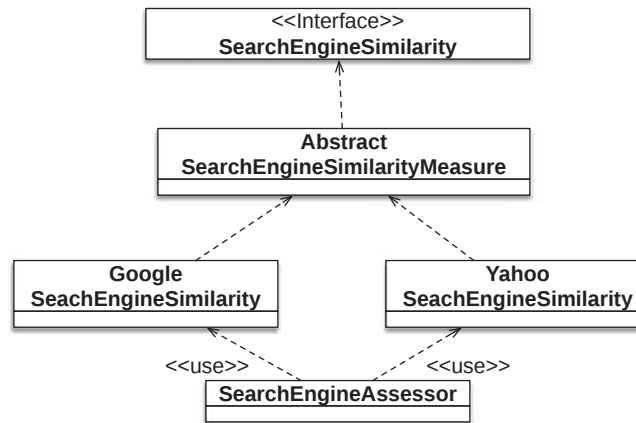


Fig. 6 Classes involved in the similarity computation for search-engine-based approaches.

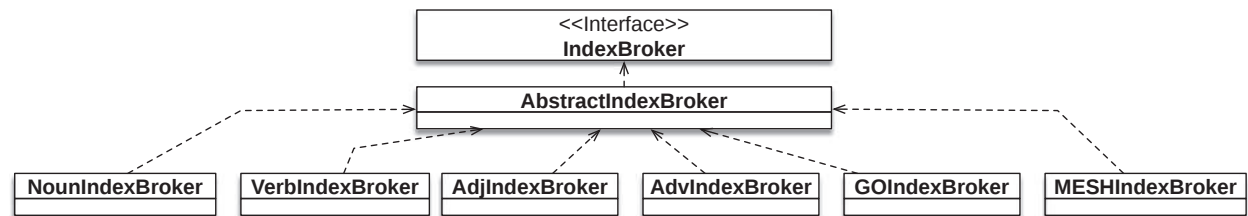


Fig. 7 Classes involved in the indexing process.

While sharing some commonalities with the *Similarity Library* such as the usage of WordNet and the implementation of several similarity measures, there are several differences. First, the *Similarity Library* is entirely written in Java and the it can be easily embedded in an existing piece of standalone Java code. Second, it is an off-the-shelf tool, and therefore there is no need to install interpreters (Python) or databases. This is because the ontologies used by the *Similarity Library* have been indexed through the Lucene IR library and then are portable without requiring any installation procedure. Third, the *Similarity Library* is a comprehensive tool to compute similarity between concepts belonging to different ontologies, not only WordNet. Besides, the *Similarity Library* also provides two search-engine based similarity strategies that can easily be combined with ontology based approaches.

Sim Pack (Ziegler et al., 2006), is a Java based library that aims at encompassing several classes of measures ranging from strategies to compare string to information theoretic approaches (e.g., Resnik, Lin). The *Similarity Library* as compare to SimPack is more focuses on similarity between ontology concepts since it provides ad-hoc indexes for WordNet, MeSH and the Gene Ontology. Moreover, it provides two search-engine based strategies and a strategy to compute similarity between sentences.

Concluding Remarks and Future Work

This paper described the *Similarity Library*, a Java based tool encompassing several similarity measures relying on different sources of knowledge. The library includes knowledge-based methods that usually exploit some ontology as source of knowledge in the

process of similarity estimation. Moreover, also some approaches exploiting search engine indexes are implemented. The library can be easily extended to include other similarity measures thanks to its modular architecture, which provides a set of simple interfaces to be implemented.

As compared to existing approaches, the Similarity Library has some nice features: i) It speeds up similarity computation by exploiting ad-hoc indexes maintaining ontological knowledge; ii) it includes both ontology and search-engine based approaches that can also be combined; iii) it encompasses different ontologies, ranging from generic one such as WordNet to domain specific such as the Gene Ontology. Finally, the library can be easily integrated in an piece of Java code and then it is easily portable.

As future work, there are two main extensions. The first is the integration of similarity approaches exploiting Wikipedia such as WikiRelate! (Ponzetto and Strube, 2007). Wikipedia has recently emerged as a viable source of knowledge to compute similarity since it provides a huge amount of unstructured knowledge, on the form of Wikipedia articles, and a categorization mechanism through which articles are arranged in a hierarchy. This integration will also enable to combine knowledge-based approaches, which can rely on knowledge manually encoded by experts, with knowledge defined by a community. The second is to include FrameNet, which provides an on-line lexical resource for English, based on frame semantics and supported by corpus evidence Ruppenhofer *et al.* (2005).

See also: Biological and Medical Ontologies: Systems Biology Ontology (SBO). Computational Tools for Structural Analysis of Proteins. Natural Language Processing Approaches in Bioinformatics. Ontology-Based Annotation Methods. Protein Functional Annotation. Semantic Similarity Definition

References

- Araujo, R., Pinto, H.S., 2007. Towards semantics-based ontology similarity. In: *Proceedings of Ontology Matching Workshop*.
- Atencia, M., Euzenat, J., Pirro, G., Rousset, M.-C., 2011. Alignment-Based Trust for Resource Finding in Semantic P2P Networks. *ISWC*. pp. 51–66.
- Borgida, A.T.W., Hirsh, H., 2005. Towards measuring similarity in description logics. In: *Proceedings of DL2005*.
- Cilibrasi, R.L., Vitanyi, P.M.B., 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering* 19 (3), 370–383.
- Consortium, G.O., 2004. The gene ontology (go) database and informatics resource. *Nucleic Acids Research* 32.
- Formica, A., 2008. Concept similarity in formal concept analysis: An information content approach. *Knowledge Based Systems* 21, 80–87.
- Hai, J., Hanhua, C., 2008. SemreX: Efficient search in semantic overlay for literature retrieval. *Future Generation Computer Systems* 24 (6), 475–488.
- Hatzivassiloglou, V., McKeown, K.R., 1997. Predicting the semantic orientation of adjectives. In: *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, pp. 174–181. Morristown, NJ, United States.
- Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G.M., Milios, E.E., 2006. Information retrieval by semantic similarity. *International Journal on Semantic Web and Information System* 2 (3), 55–73.
- Jiang, J., Conrath, D., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In: *Proceedings of ROCLING X*.
- Kashyap, V., Sheth, A., 1996. Schematic and semantic similarities between database objects: A context-based approach. *Vldb Journal* 5, 276–304.
- Landauer, T.K., Dumais, S.T., 1997. Solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review* 104).
- Leacock, C., Chodorow, M., 1998. Combining local context and wordNet similarity for word sense identification. In: Fellbaum, C. (Ed.), *WordNet: A Lexical Reference System and its Application*. MIT Press, pp. 265–283.
- Leacock, C., Chodorow, M., Miller, G.A., 1998. Using corpus statistics and wordNet relations for sense identification. *Computational Linguistics* 24 (1), 147–165.
- Lee, J., Kim, M., Lee, Y., 1993. Information retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation* 49, 188–207.
- Li, Y., Bandar, A., McLean, D., 2003. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering* 15 (4), 871–882.
- Li, Y., McLean, D., Bandar, Z., O'Shea, J., Crockett, K., 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering* 18 (8), 1138–1150.
- Lin, D., 1998. An information-theoretic definition of similarity. In: *Proceedings of Conference on Machine Learning*, pp. 296–304. San Francisco, CA: Morgan Kaufmann. Available at: citeseer.ist.psu.edu/95071.html.
- Lord, P.W., Stevens, R.D., Brass, A., Goble, C.A., 2003. Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation. *Bioinformatics* 19 (10), 1275–1283.
- Meilicke, C., Stuckenschmidt, H., Tamilin, A., 2007. Repairing Ontology Mappings. *AAAI*. pp. 1408–1413.
- Mihalcea, R., Corley, C., Strapparava, C., 2006. Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity. *AAAI*.
- Pedersen, T., Pakhomov, S.V.S., Patwardhan, S., Chute, C.G., 2007. Measures of semantic similarity and relatedness in the Biomedical Domain. *Journal of Biomedical Informatics* 40 (3), 288–299.
- Pedersen, T., Patwardhan, S., Michelizzi, J., 2004. WordNet: Similarity – Measuring the Relatedness of Concepts. *AAAI*. pp. 1024–1025.
- Penzo, W., Lodi, S., Mandreoli, F., Martoglia, R., Sassatelli, S., 2008. Semantic Peer, Here are the Neighbors you Want!. *EDBT*. pp. 26–37.
- Pirró, G., 2009. A semantic similarity metric combining features and intrinsic information content. *Data and Knowledge Engineering* 68 (11), 1289–1308.
- Pirró, G., 2012. REWOrd: Semantic Relatedness in the Web of Data. *AAAI*.
- Pirró, G., Euzenat, J., 2010a. A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness. *ISWC*.
- Pirró, G., Euzenat, J., 2010b. A semantic similarity framework exploiting multiple parts-of speech. In: *OTM Conferences*. pp. 1118–1125.
- Pirró, G., Ruffolo, M., Talia, D., 2008. SECCO: On building semantic links in peer to peer networks. *Journal on Data Semantics XII*, 1–36.
- Pirró, G., Seco, N., 2008. Design, Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content. *ODBASE*. pp. 1271–1288.
- Pirró, G., Talia, D., 2008. LOM: A linguistic ontology matcher based on information retrieval. *Journal of Information Science* 34 (6), 845–860.
- Pirró, G., Trunfio, P., Talia, D., Missier, P., Goble, C., 2010. ERGOT: A Semantic-Based System for Service Discovery in Distributed Infrastructures. *CCGrid*. pp. 263–272.
- Ponzetto, S.P., Strube, M., 2007. Knowledge derived from wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research* 30, 181–212.
- Quillian, M., 1968. Semantic memory. In: Minsky, M. (Ed.), *Semantic Information Processing*. Cambridge, MA: MIT Press.
- Rada, R., Mili, H., Bicknell, E., Blettnet, M., 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics* 19, 17–30.

- Ravi, S., Rada, M., 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: Proceedings of ICSC.
- Resnik, P., 1995. Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of IJCAI. pp. 448–453.
- Rissland, E.L., 2006. AI and similarity. *IEEE Intelligent Systems* 21, 39–49.
- Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J., 2005. FrameNet II: Extended theory and practice. Technical Report, ICSI. Available at: <http://framenet.icsi.berkeley.edu/book/book.pdf>.
- Salton, G., Wong, A., Yang, C.S., 1975. A vector space model for automatic indexing. *Communication of the ACM* 18 (11), 613–620.
- Seco, N., 2005. Computational models of similarity in lexical ontologies. Master's thesis, University College Dublin.
- Seco, N., Veale, T., Hayes, J., 2004. An intrinsic information content metric for semantic similarity in wordNet. In: Proceedings of ECAI. pp. 1089–1090.
- Shepard, R., 1987. Toward a universal law of generalization for psychological science. *Science* 237, 1317–1323.
- Turney, P.D., 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. *ECML*. pp. 491–502.
- Turney, P.D., 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *ACL*. pp. 417–424.
- Tversky, A., 1977. Features of similarity. *Psychological Review* 84 (2), 327–352.
- Wu, Z., Palmer, M., 1994. Verb semantics and lexical selection. In: Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics. pp. 133–138.
- Zavaracky, A., 2003. Glossary-based semantic similarity in the wordNet ontology. Master's thesis, University College Dublin.
- Zhang, X., Jing, L., Hu, X., Ng, M.K., Zhou, X., 2007. A Comparative Study of Ontology Based Term Similarity Measures on PubMed Document Clustering. *DASFAA*. pp. 115–126.
- Ziegler P., Kiefer C., Sturm, C., Dittrich, K., Bernstein, A., 2006. Generic similarity detection in ontologies with the SOQA-SimPack toolkit. In: Proceedings of SIGMOD Conference. pp. 751–753.

Relevant Websites

<http://www.lucene.org>
 Apache Lucene.

<http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi>
 Similarity - maraca.

<http://simlibrary.wordpress.com>
 The Similarity Library.

Biographical Sketch



Giuseppe Pirrò is a researcher at the Institute for High Performance Computing and Networking (ICAR-CNR). He got his PhD in Computer and System Engineering and Master Degree in Computer Engineering at the University of Calabria. His research interests include Semantic Web, graph databases, and distributed systems. He published papers in top journals and conferences including AIJ, TWEB, PVLDB, ISWC, AAAI, WWW, CIKM.

Tools for Semantic Analysis Based on Semantic Similarity

Marianna Milano, University of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Bioinformatics approaches have led to the introduction of different methodologies and tools for the analysis of different types of data related to proteins, ranging from primary, secondary and tertiary structures to interaction data (Cannataro *et al.*, 2010).

The same is not true of for functional knowledge. The reason is related to an objective representation and measurable properties of protein sequences and structures, whereas functional aspects have neither. Thus, to compare functional aspects they must be expressed in a common and objective form. One of the most advanced tools for encoding and representing functional knowledge in a formal way is the Gene Ontology (GO) (Shah and Rubin, 2006; Ashburner *et al.*, 2000). GO is divided in three ontologies, named Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). Each ontology consists in a set of GO terms representing different biological processes, functions and cellular components within the cell. GO terms are connected to each other to form a hierarchical graph. Terms representing similar functions are close to each other within this graph.

Biological molecules are associated with GO terms that represent their functions, biological roles, and localization through a process also known as annotation process. The annotation process can be performed under the supervision of an expert or in a fully automated way. For this reason, every annotation is labeled with an Evidence Code (EC) that keeps track of the type of process used to produce the annotation itself.

All the annotations involving a set of proteins or genes are commonly called the annotation corpus, and usually, refer to the whole proteomes and genomes (i.e., the annotation corpus of yeast). The availability of well formalized functional data enabled the use of computational methods to analyze genes and proteins from the functional point of view. An interesting problem is how to express quantitatively the relationships between GO terms. In order to evaluate the similarity among terms belonging to the same ontology, a set of formal instruments called Semantic Similarity Measures (SSM) have been developed. A SSM takes in input two or more terms of the same ontology and produces as output a numeric value representing their similarity. Semantic similarity applied to the GO annotations of gene products provides a measure of their functional similarity. Consequently, the use of SSMs to analyze biological data is gaining a broad interest from researchers (Milano *et al.*, 2014, 2016). The semantic similarity has become a valuable tool for validating the results drawn from biomedical studies such as gene clustering, gene expression data analysis, prediction and validation of molecular interactions, and disease gene prioritization. Over the years several approaches are proposed to quantify the semantic similarity between terms or annotated entities in ontologies.

Background/Fundamentals

A SS measure is a formal instrument that ensures the quantification of the relatedness of two or more terms within an ontology. There are two different measure approaches: the measures that quantify the similarity among two terms, often known as pairwise measures, and the measures able to describe the relatedness of two sets of terms, yielding a global similarity of sets, referred to as groupwise measures.

The similarity measures have been extended to gene and gene products that are annotated with terms belonging to the ontology, allowing to draw conclusions on the relationship of two proteins relying on the similarity of GO terms.

SS measures can be classified according to the properties of GO terms and annotation corpora on which they are based, and the strategies and models on which they rely. A typical categorization groups the SS measures which are based on: Term Information Content (IC), Term Depth, a common ancestor, all common ancestors, Path Length and Vector Space Models (VSM).

The measures based on Term Depth and IC evaluate terms similarity on the basis of the specificity of the terms. One concept commonly used in these approaches is information content (IC), which gives a measure how specific and informative a term is.

More formally, given an annotation corpus, the IC of a term c is defined as:

$$IC(c) = -\log p(c)$$

where $p(c)$ is the fraction of gene products that are annotated with term c or its descendants in the annotation corpus.

Measures based on a common ancestor select a common ancestor of two terms according to its properties and then evaluates the semantic similarity on the basis of the distance between the terms and their common ancestor and the properties of the common ancestor. IC is used to select the proper ancestor, yielding to the development of methods based on the information content of common ancestor: for instance, the Maximum Informative Common Ancestor (MICA)-based approaches select the common ancestor of two terms t_1 and t_2 with highest IC:

$$MICA(t_1, t_2) = \operatorname{argmax}_i IC(t_i)$$

$t_i \in \text{ancestors}(t_1, t_2)$

Resnik's measure (Resnik, 1995) (*simRes*), one of the most popular SS measures, is an exponent of this category. The semantic similarity between two terms t_1 and t_2 is simply the IC of the MICA:

$$\text{sim}_{\text{Res}}(t_1, t_2) = \text{IC}(\text{MICA}(t_1, t_2))$$

Lin's measure (Lin, 1998), *simLin*, considers both the information content of the MICA and of the input terms:

$$\text{sim}_{\text{Lin}}(t_1, t_2) = \frac{\text{IC}(\text{MICA}(t_1, t_2))}{\text{IC}(t_1) + \text{IC}(t_2)}$$

In a similar way, Jiang and Conrath's measure (Jiang and Conrath, 1997), *simJC*, takes into account the MICA and the input terms:

$$\text{sim}_{\text{JC}}(t_1, t_2) = 1 - \text{IC}(t_1) + \text{IC}(t_2) - 2\text{XIC}(\text{MICA}(t_1, t_2))$$

Also *simGIC* (Pesquita *et al.*, 2009) is a measure based on IC, but instead of focusing on only the most informative common ancestor of a pair of terms, it considers all the common ancestors of two sets A and B of GO terms:

$$\text{sim}_{\text{GIC}}(A, B) = \frac{\sum_{t \in \{\text{GO}(A) \cap \text{GO}(B)\}} \text{IC}(t)}{\sum_{t \in \{\text{GO}(A) \cup \text{GO}(B)\}} \text{IC}(t)}$$

where $\text{GO}(X)$ is the set of terms within X and all their ancestors in the GO hierarchy.

The measures based on all common ancestors (Gentleman, 2005; Sheehan *et al.*, 2008; Yu *et al.*, 2007; Couto *et al.*, 2007; Wu *et al.*, 2006; Li *et al.*, 2010) collect all the ancestors of terms, and then evaluate the overlap between the two sets, sometimes using also other characteristics, for example, IC or edge distance to determine term similarity. The measures based on Path Length (Yu *et al.*, 2007; Al-Mubaid and Nagar, 2008; Wang *et al.*, 2007; Othman *et al.*, 2008; Pekar and Staab, 2002; Wu *et al.*, 2006; Wu and Palmer, 1994) consider the length of the path from terms to their common ancestor that can be lowest common ancestor (LCA) – or maximum common ancestor – MCA. The measures based on VSM (Popescu *et al.*, 2006; Chabalier *et al.*, 2007; Bodenreider *et al.*, 2005) evaluated the similarity by considering the distance among vectors that are defined using topological considerations (e.g., the cosine of the induced angle), as well as semantic considerations (e.g., IC-based). The vectors are composed by the annotations of proteins. The calculation of protein semantic similarity consists of the evaluation of the semantic similarity between all the terms annotating two proteins, and then combine them in some way. Several measures for calculating the functional similarity between gene products have been proposed, which can be divided into two categories: pairwise and groupwise approaches. Groupwise Term SS measures can be directly extended to measure protein similarity, simply considering as input the two sets of GO terms annotating the proteins. Instead, Pairwise Term SS measures evaluate the similarity of pairs of terms and therefore, are not directly applicable to genes and proteins. Consequently, it is necessary to define a strategy, called mixing strategy, that transforms all the pairwise term similarities into a single representative value. There are six mixing strategies reported in the literature:

- Average (avg) strategy that considers the average of all term pairwise similarities (Lord *et al.*, 2003);
- Maximum (max) strategy that is based on the maximum of all term pairwise similarities (Sevilla *et al.*, 2005);
- Best Match Average (BMA) that relies on the average of similarity between best-matching terms (Azuaje *et al.*, 2005);
- funSim strategy determines the protein semantic similarities in MF and BP ontologies using max, avg or BMA mixing strategies and then they are combined together in a non-linear way (Schlicker *et al.*, 2006);
- Information Theory-based Semantic Similarity strategy filters the best matching pairs on the basis of their similarities, then the average is calculated (Tao *et al.*, 2007);
- FuSSiMeG strategy similar to max strategy; and the maximum of all term pairwise similarities weighted by the ICs of the terms is selected (Couto *et al.*, 2007).

SS measures are substantially different from all the other classical measures such as sequence similarity because they use information regarding the functions and roles of proteins themselves. However, the comparison of gene product using semantic similarity requires a well-defined ontology and a complete and reliable annotation corpus.

Furthermore, the SS measures have been verified to be good predictors of protein–protein interactions (Ivanov *et al.*, 2011). The robustness and generality of the analysis are certified by the heterogeneity of data used in different assessment methods. As expected, the best results are obtained when using BP ontology, while CC ontology has proven to be not particularly suited for this task (Guo *et al.*, 2006).

Different works (Chen *et al.*, 2009) identified Resnik as one of the best semantic similarity measures (Guo *et al.*, 2006), especially when combined with the max mixing strategy (Xu *et al.*, 2008). This is not unexpected since max strategy favors protein pairs sharing even only a part of their functions and, as we reported previously, two proteins are likely to interact even when they only have in common some of their aspects.

Systems and/or Applications

The current scenario is characterized by the different tools that ensure the computation of many SS measures; however, there is none tool that implements all the SS measures or that is easily extendible. Considering the distribution, tools are mainly available as web servers, i.e., FuSSiMeG, ProteInOn, functional similarity matrix (FunSimMat), GOToolBox, G-SESAME or as packages for the R platform, csbl.go, GOSemSim, GOVis. However, FuSSiMeG, protein interactions and ontology (ProteInOn), FunSimMat, csbl.go and SemSim together cover almost all the similarity measures. In general, tools are based on GO and annotation corpora. Some tools, such as the web servers, include their own copy of annotation corpora and GO, offering user-friendly solutions. However, they rely on maintainers for updated data and generally do not offer many possibilities of customization or extension. On the contrary, other tools such as stand-alone R-packages, are generally more flexible and often easily extendable, but they require the intervention of expert users. Usually, they require the user to provide annotations and ontologies as input data in more or less common formats. While this enables the full control over data used and guarantees the possibility to use most-updated data, the preparation of input datasets may result in an error-prone waste of time.

Analysis and Assessment

In this section the web servers tools and R platform tools are described.

FuSSiMeG-Functional Semantic Similarity Measure Between Gene-Products

FuSSiMeG (Couto *et al.*, 2003) is a tool that measures the semantic similarity between all the GO terms and functional similarity of gene products by comparing the semantic similarity between the annotations of genes products provided by Gene Ontology Annotation (GOA).

Given a gene product from SwissProt/TrEMBL, a list of GO terms assigned to a gene product in GOA is defined as:

$$T(g) = \{t : (g, t) \text{GOA}\}$$

where GOA is a set of pairs consisting of gene products and GO terms assigned in the GOA database.

FuSSiMeG assumes that two gene products have a functional similarity when they are annotated with terms functionally similar. The measure of similarity between gene products is calculated from the maximum of the similarity among their terms. FuSSiMeG is defined as:

$$FuSSiMeG(g_1, g_2) = \max\{SSM(t_1, t_2) : t_1 T(g_1) \wedge t_2 T(g_2)\}$$

where g_1 e g_2 are two gene products.

FuSSiMeG assumes that two gene products have functional similarities when they are annotated with similar functional terms and when these terms have a significant information content. Then, the previous expression is redefined as:

$$FuSSiMeG(g_1, g_2) = \max\{SSM(t_1, t_2)XIC(t_1)XIC(t_2) : t_1 T(g_1) \wedge t_2 T(g_2)\}$$

To compute the semantic similarity of gene products or the terms GO, FuSSiMeG requires as input the pair of proteins or GO terms of interest; FuSSiMeG computes the semantic similarity by selecting one measure among Resnik, Lin, JiangConrath, ResnikGraSM, LinGraSM, JiangConrathGraSM. The output consists of: the similarity score in percent and a weighted similarity score. The weighted similarity considers the information content of each term that is proportional to how many times the term is annotated.

ProteInOn- Protein Interactions and Ontology

ProteInOn (Faria *et al.*, 2007) is an online tool for exploring and comparing proteins within the context of Gene Ontology. It implements several semantic similarity measures for calculating protein and term similarity, and combines information on protein-protein interactions and GO term assignment for protein characterization. The tool can be used to compute semantic similarity between proteins or GO terms, using one of the three measures implemented (Resnik's, Lin's and Jiang and Conrath's measures) with or without the GraSM approach. Furthermore, it introduces a preliminary weighting factor to improve the specificity of these measures for protein semantic similarity, addressing the issue of their displacement from the GO graph. It can also be used to find GO terms assigned to one or more proteins or GO terms representative of a set of proteins. A preliminary score is used to measure the representativeness of a term for a set of proteins, based on the number of proteins the term is annotated to and its probability of annotation. The ProteInOn database contains protein and GO terms and is structured according to a relational model, implemented in MySQL. The main entities are, therefore, protein and GO terms, and major reports are the protein-GO terms annotations, protein-protein interactions and GO term-GO term ancestry. The data corresponding to these entities and relationships are imported from four public databases: UniProt, for the data source of protein which includes the accession numbers, names, and sequences, GO, for GO term data and a GO term-GO term ancestor; GOA for the annotation; IntAct for protein-protein interactions. The ProteInOn interface consists of a web page divided into four stages: entry input, query

selection, query options and results. ProteinOn takes as input up to five proteins or GO terms. The query selection depends on the kind of input. For the GO term, there are two available queries: 'get information content' which returns a list with the accession number, name and information content of each of the terms entered and 'calculate term similarity' which is available for two or more terms only, and returns for each pairwise combination of the terms entered, the semantic similarity score between them. The queries for proteins are:

'Find assigned GO terms' which lists all GO terms directly assigned to each of the proteins entered, and the evidence code(s) for that assignment; 'find interacting proteins' which returns a list of all proteins known to interact with each of the proteins entered; 'calculate protein similarity' which is available for two or more proteins, and returns for each pairwise combination of the entered proteins a set of semantic similarity scores between them. Each query ensures to select number of options for further specificity. For example, the user can limit the queries to one of the three aspects of GO or he/she can decide to ignore the IEA annotation from the analysis. Furthermore there is the options that limits the results to proteins that interact with all of the proteins entered only or the options that lists the GO terms that better represent the set of proteins entered, according to the representativeness score. Then, for the similarity queries, the user can select one of the three semantic similarity measures available, i.e., Resnik's, Lin's, and Jiang and Conrath's. The results are presented in tables, in which the output of the selected queries and the Similarity scores in percentage are reported.

FunSimMat

FunSimMat is a database that provides several different semantic similarity measures for GO terms. It offers various precomputed functional similarity values for proteins contained in UniProtKB and for protein families in Pfam and SMART. FunSimMat also provides several semantic and functional similarity measures for each pair of proteins with the GO annotation from UniProtKB and offers different types of queries available through a web interface. The first query consists of the semantic all-against-all comparison of GO terms contained in an input list provided by the user by using four different measures, simRel, Lin, Resnik, e Jiang e Conrath. The second query regards the comparison of one query protein or protein family with a list of proteins or protein families, which can be compiled in different ways. The simplest one is to enter the corresponding accession numbers into the query form of the website. The third query option is the definition of a functional profile. A functional profile consists of a list of GO terms from the biological process, molecular function or cellular component. This functional profile is treated as an annotation class and it is compared to a list of proteins or protein families. The web front-end provides HTML forms for all of the different query options that FunSimMat offers. The results are displayed in a table and can be downloaded as a tab-delimited text file or printed.

GOToolBox

The web server GOToolBox (Martin *et al.*, 2004) provides a set of programs that allow a functional study of groups of genes based on GeneOntology. It consists of a set of methods and tools that process the GO annotations for any species. All ontology data and the GO terms associated to genes are gathered from GO Consortium web site. GOToolBox performs five main operations: the building of gene dataset, the selection, and testing of the ontology level (GODiet), statistical analysis of the terms associated with the sets of genes (GO-Stats), gene clustering based on GO (GO-Proxy) and the recovery of genes based on the similarity of GO annotations (GO-Family). The dataset creation consists of retrieving, for each individual gene of the dataset, all the corresponding GO terms and their parent terms using the Dataset creation program. The genomic frequency of each GO term associated with genes in the dataset is then calculated. GO-Diet is an optional tool useful to reduce the number of GO terms associated with a set of genes data, to facilitate the analysis of the results, in particular when the list the gene of input and/or the number of associated GO terms is large. GO-Stats tool permits the automatic ranking of all annotation terms, and the evaluation of the significance of their occurrences within the dataset obtained by computing the frequencies of terms within the dataset and by comparing with reference frequencies. GO-Proxy tool groups together functionally related genes on the basis of their GO terms. GO-Family aims at finding genes having shared GO terms with a user-entered gene, on the basis of a functional similarity calculation.

G-SESAME

The web tool G-SESAME (Du *et al.*, 2009) is composed of four tools which includes the tool for measuring the semantic similarities of two and multiple GO terms, the tool for the semantic comparison of two gene products from two different species, and multiple gene comparisons and clustering tools. The tool for measuring the semantic similarities of two GO terms computes the semantic similarity by using the Wang measures among GO terms or GO terms set provided. The tool for measuring the semantic similarities of two genes from two species compares the functional similarities of two genes. The tool for multiple genes comparison computes the gene functional similarity measurement by applying Jiang's, Lin's and Resnik's methods. The output consists of the similarities of genes and also the clustering results of these genes (Table 1).

Table 1 Overview of web tools and their related functions and available semantic similarity measures

<i>Tool</i>	<i>Functions</i>	<i>Measures</i>
FuSSiMeG	SS measures, statistical tests	Resnik, Lin, JiangConrath, GraSM
ProteinOn	SS measures, search for assigned GO Terms and annotated proteins, representative of GO Terms	Resnik, Lin, JiangConrath, simGIC, GraSM, simU
FunSimMat	SS measures, disease-related genes prioritization	simRel, Lin, Resnik, JiangConrath
GOToolBox	SS measures, clustering	Si, Sp, SCD
G-SESAME	SS measures, clustering	G-SESAME

GOSemSim

The GOSemSim (Yu, 2010) is a R package developed to compute semantic similarity among GO terms, sets of GO terms, gene products, and gene clusters, providing five methods, four of which are based on the IC (Resnik, Lin, Jiang and Corrado and Schlicker) and one on the graph (Wang). The package provides six functions of which goSim, mgoSim, Genesim, cluster Sim calculate the semantic similarity between GO terms, a term set GO, GO descriptions of gene products and GO descriptions of gene cluster, while mgeneSim and mclusterSim are designated to calculate the matrix of similarity scores of a set of genes and gene clusters. For all six functions, the analysis is restricted to one of the three ontologies, “BP”, “MF”, “CC”. The goSim function provides the semantic similarity score for a pair of GO terms. The output consists of value between 0 and 1. The mgoSim function generates a score of similarity between two lists of GO terms. The genesim function estimates the semantic similarity between two genes. The mgeneSim function calculates the pairwise similarity scores for a list of genes. The mgeneSim automatically removes genes that have no annotations. The clusterSim function was developed to calculate the similarity between two gene clusters. This function computes the pairwise similarity between the gene products from different clusters and then provides the average of all similarities as the similarity of two gene clusters. The mclusterSim function calculates the pairwise similarity of a set of gene clusters.

GOvis

The GOvis package (Heydebreck *et al.*, 2004) calculates the similarity between GO graphs or between Entrez identifiers gene on their induced GO graphs. The relations between different terms of GO within one of the three ontologies are represented by a directed acyclic graph. The leaves of the graph represent the most specific terms and the connections between two nodes represent the relationship between the less specific and more specialized. The graph induced is a graph that is obtained from a specific node to the root, considering the various parents. Given a set of genes, it is possible to derive from GOA the specific set of annotations for these genes and, therefore, the graph induced by each of them. Starting from two graphs induced, three operations, the union, intersection and complement can be performed. GOvis implements two methods based on the graph structure, namely simUI and simLP. Both methods take as codominio the 0–1 range (a higher value implies a higher level of similarity). SimUI computes the intersection of the induced graphs for two genes. SimLP considers the depth of the path from a specified term to the root. More the paths are overlapping, more the terms will be similar.

Csbl.go

Csbl.go (Ovaska, 2016) package computes similarities for arbitrary number of genes and supports the following measures: Czekanowski-Dice, Kappa, Resnik (with GraSM as an option), Jiang-Conrath (GraSM), Lin (GraSM), Relevance, Cosine and SimGIC. The MICA-based measures (Resnik, Lin, Jiang-Conrath, Relevance and GraSM enhancements) are implemented as a combination of R and C++ code. Similarity computation needs GO term probabilities for the reference gene set, for this reason csbl.go provides precomputed probability tables for Homo sapiens, Saccharomyces cerevisiae, Caenorhabditis elegans, Drosophila melanogaster, Mus musculus and Rattus norvegicus. The tables are computed based on all gene and protein annotations for the given organism found in the geneontology.org database. The package also has an option to use custom tables. The taxonomy ID of the organism is stored along with probability tables as metadata, which enables selection of a table by organism ID. The package also includes an option to compute GO term enrichment using Fisher's Exact Test (Al-Mubaid and Nagar, 2008) (Table 2).

Discussion

During the years, an important component of biomedical research have become the ontologies. The reason is related to the characteristics of ontologies to provide the formalism, and common terminology necessary for researchers to describe their results. This ensures to easily share the results and reused ones by both humans and computers. The main advantage of the use of ontologies is the comparison of concepts and entities annotated with those concepts by using of semantic similarity measure. SS measures, i.e., the quantification of the similarity of two or more terms belonging to the same ontology, is a well- established field.

Table 2 Overview of R packages and their related functions and available semantic similarity measures

<i>Tool</i>	<i>Functions</i>	<i>Measures</i>
csbl.go	SS measures, clustering based on SS	Resnik, Lin, JiangConrath, GRaSM, simRel, Kappa Statistics, Cosine, Weighted Jaccard, Czekanowski-Dice
GOSemSim	SS measures	Resnik, Lin, Jiang, simRel, G-SESAME
GOvis	SS measures	simLP, simUI

There are two main approaches for this comparison: pairwise, in which entities are represented as lists of concepts that are then compared individually; and groupwise, in which the annotation graphs of each entity are compared as a whole. Until now, most research efforts in this area have been conducted to the development of different tools that ensure the computation of SS measures. However, there is a lack of tools that implements all the SS measures or that is easily extendible. Future efforts will address to implementation of tools that enable the computation all the SS measures and that ensure to integrate the next generation of SS measures. Furthermore, the work will direct to design faster solutions for the calculation of semantic similarity measures.

See also: Biological and Medical Ontologies: Systems Biology Ontology (SBO). Computational Tools for Structural Analysis of Proteins. Natural Language Processing Approaches in Bioinformatics. Ontology-Based Annotation Methods. Protein Functional Annotation. Semantic Similarity Definition

References

- Al-Mubaid, H., Nagar, A., 2008. Comparison of four similarity measures based on GO annotations for gene clustering. In: Symposium on Computers and Communications, 2008, IEEE, pp. 531–536.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25 (1), 25–29.
- Azuaje, F., Wang, H., Bodenreider, O., 2005. Ontology-driven similarity approaches to supporting gene functional assessment. In: Proceedings of the ISMB'2005 SIG meeting on Bio-ontologies pp. 9–10.
- Bodenreider, O., Aubry, M., Burgun, A., 2005. Non-lexical approaches to identifying associative relations in the gene ontology. In: Pacific Symposium on Biocomputing, NIH Public Access, p. 91.
- Cannataro, M., Guzzi, P.H., Veltri, P., 2010. Protein-to-protein interactions: Technologies, databases, and algorithms. *ACM Computing Surveys (CSUR)* 43 (1), 1.
- Chabalier, J., Mosser, J., Burgun, A., 2007. A transversal approach to predict gene product networks from ontology-based similarity. *BMC Bioinformatics* 8 (1), 235.
- Chen, G., Wang, J., Li, M., 2009. GO semantic similarity based analysis for human protein interactions. In: International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, 2009, IJCBS'09, IEEE, pp. 207–210.
- Couto, F.M., Silva, M.J., Coutinho, P.M., 2003. Implementation of a functional semantic similarity measure between gene-products. Technical Report DI/FCULTR, Department of Informatics, University of Lisbon.
- Couto, F.M., Silva, M.J., Coutinho, P.M., 2007. Measuring semantic similarity between gene ontology terms. *Data & Knowledge Engineering* 61 (1), 137–152.
- Du, Z., Li, L., Chen, C.F., Philip, S.Y., Wang, J.Z., 2009. G-SESAME: Web tools for GO-term-based gene similarity analysis and knowledge discovery. *Nucleic Acids Research*. gkp463.
- Faria, D., Pesquita, C., Couto, F.M., *et al.*, 2007. Proteinon: A web tool for protein semantic similarity. Technical Report DI/FCULTR, Department of Informatics, University of Lisbon.
- Gentleman, R., 2005. Visualizing and distances using GO. Available at: <http://www.bioconductor.org/docs/vignettes.html>.
- Guo, X., Liu, R., Shriver, C.D., *et al.*, 2006. Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics* 22 (8), 967–973.
- Heydebreck, A., Huber, W., Gentleman, R., 2004. Differential expression with the bioconductor project. In: Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics. New York, NY: Wiley.
- Ivanov, A.S., Zgoda, V.G., Archakov, A.I., 2011. Technologies of protein interactomics: A review. *Russian Journal of Bioorganic Chemistry* 37 (1), 4–16.
- Jiang, J.J., Conrath, D.W., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. ROCLING X, arXiv:cmp-lg/9709008.
- Li, B., Wang, J.Z., Feltus, F.A., *et al.*, 2010. Effectively integrating information content and structural relationship to improve the GO-based similarity measure between proteins. arXiv:1001.0958. Available at: <http://arxiv.org/abs/1001.0958>.
- Lin, D., 1998. An information-theoretic definition of similarity. *ICML* 98 (1998), 296–304.
- Lord, P.W., Stevens, R.D., Brass, A., Goble, C.A., 2003. Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation. *Bioinformatics* 19 (10), 1275–1283.
- Martin, D., Brun, C., Remy, E., *et al.*, 2004. GOToolBox: Functional analysis of gene datasets based on Gene Ontology. *Genome Biology* 5 (12), R101.
- Milano, M., Agapito, G., Guzzi, P.H., Cannataro, M., 2014. Biases in information content measurement of gene ontology terms. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, pp. 9–16.
- Milano, M., Agapito, G., Guzzi, P.H., Cannataro, M., 2016. An experimental study of information content measurement of gene ontology terms. *International Journal of Machine Learning and Cybernetics*. 1–13.
- Othman, R.M., Deris, S., Illias, R.M., 2008. A genetic similarity algorithm for searching the gene ontology terms and annotating anonymous protein sequences. *Journal of Biomedical Informatics* 41 (1), 65–81.
- Ovaska, K., 2016. Using semantic similarities and csbl. go for analyzing microarray data. *Microarray Data Analysis: Methods and Applications*. 105–116.
- Pekar, V., Staab, S., 2002. Taxonomy learning: Factoring the structure of a taxonomy into a semantic classification decision. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, Association for Computational Linguistics, pp. 1–7.
- Pesquita, C., Faria, D., Couto, F.M., 2009. Measuring coherence between electronic and manual annotations in biological databases. In: Proceedings of the 2009 ACM Symposium on Applied Computing, ACM, pp. 806–807.
- Popescu, M., Keller, J.M., Mitchell, J.A., 2006. Fuzzy measures on the gene ontology for gene product similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3 (3), 263–274.

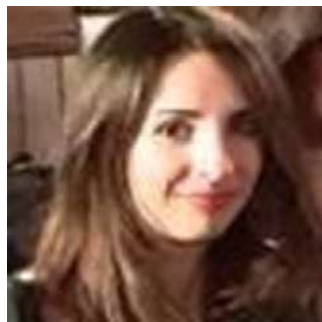
- Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, arXiv:cmp-lg/9511007.
- Schlicker, A., Domingues, F.S., Rahnenführer, J., *et al.*, 2006. A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics* 7 (1), 302.
- Sevilla, J.L., Segura, V., Podhorski, A., *et al.*, 2005. Correlation between gene expression and GO semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 2 (4), 330–338.
- Shah, N., Rubin, D., 2006. *Ontologies for bioinformatics (Computational Molecular Biology)*. Kenneth Baclawski and Tianhua Niu The MIT Press; ISBN: 0-262-02591-4; Hardcover; 440pp; 2005; £29.95. Available at: <https://doi.org/10.1093/bib/bbl011>.
- Sheehan, B., Quigley, A., Gaudin, *et al.*, 2008. A relation based measure of semantic similarity for gene ontology annotations. *BMC Bioinformatics* 9 (1), 468.
- Tao, Y., Sam, L., Li, J., *et al.*, 2007. Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics* 23 (13), i529–i538.
- Wang, J.Z., Du, Z., Payattakool, R., Philip, S.Y., Chen, C.F., 2007. A new method to measure the semantic similarity of GO terms. *Bioinformatics* 23 (10), 1274–1281.
- Wu, X., Zhu, L., Guo, J., *et al.*, 2006. Prediction of yeast protein–protein interaction network: Insights from the gene ontology and annotations. *Nucleic Acids Research* 34 (7), 2137–2150.
- Wu, X., Zhu, L., Guo, J., Zhang, D.Y., Lin, K., 2006. Prediction of yeast protein–protein interaction network: Insights from the gene ontology and annotations. *Nucleic Acids Research* 34 (7), 2137–2150.
- Wu, Z., Palmer, M., 1994. Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 133–138.
- Xu, T., Du, L., Zhou, Y., 2008. Evaluation of GO-based functional similarity measures using *S. cerevisiae* protein interaction and expression profile data. *BMC Bioinformatics* 9 (1), 472.
- Yu, G., 2010. GO-terms semantic similarity measures. *Bioinformatics* 26 (7), 976–978.
- Yu, H., Jansen, R., Gerstein, M., 2007. Developing a similarity measure in biological function space. *Bioinformatics*. 1–18.

Further Reading

FastSemSim. Available at: <http://sourceforge.net/projects/fastsemsim/>.

Schlicker, A., Albrecht, M., 2008. FunSimMat: A comprehensive functional similarity database. *Nucleic Acids Research* 36 (Suppl. 1), D434–D439.

Biographical Sketch



Marianna Milano received the Laurea degree in biomedical engineering from the University Magna Græcia of Catanzaro, Italy, in 2011. She is a PhD student at the University Magna Græcia of Catanzaro. Her main research interests are on biological data analysis and semantic-based analysis of biological data. She is a member of IEEE Computer Society.

Functional Enrichment Analysis Methods

Pietro H Guzzi, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The development of high-throughput experimental techniques, such as microarrays or next-generation sequencing, allow the screening of large number of genes and or proteins. Bioinformatics analysis, made on the gene input list, enable the identification of a subset of products that exhibit an “interesting” pattern of behaviour. The last step of the bioinformatics analysis consists of the interpretation of results from a biological perspective helping researchers to group genes. Such process is often referred to as enrichment, or functional enrichment, and consists of the individuation of the functions of the input list of genes by using annotations, i.e. known functions of the genes stored into biological ontologies. Consequently many algorithms and tools have been proposed, referred to as functional enrichment tools (Huang *et al.*, 2009). The increase of the use of such algorithms is clearly shown by the trend of the increase of the citations on search engines considering both articles describing novel methods and articles describing the use of existing ones. In parallel the scientific community has developed methodologies for classifying existing methods and website for suggesting the use of the appropriate algorithms.

Here we follow the classification proposed in Huang *et al.* (2009), that propose three main classes for such algorithms on the basis of the schema used for analysing input data. All the algorithms are based on three main pillars: (i) a list of genes or proteins, (ii) a source of annotations (i.e., formal descriptions of the functions) based on a biological ontology, e.g., Gene Ontology (Ashburner *et al.*, 2000), (iii) a way for testing the relevance of the annotations in the input list using statistical or semantic-based methods (Guzzi *et al.*, 2012).

The first class, called Singular Enrichment Analysis (SEA), groups traditional algorithms that receives as input a list of user preselected genes or proteins, and than tests the enrichment of each annotation term extracted from an ontology one by one and in iterative way. The second class, Gene Set Enrichment Analysis (GSEA), is similar to SEA from which differ considering the strategy to calculate the relevance (statistical and biological) compared to SEA. The last class, Modular Enrichment Analysis (MEA), calculates the enrichment using network based algorithms (Roy and Guzzi, 2015).

SEA algorithms take as input a list of genes or gene products that have been produced by the user (e.g. differentially expressed genes). Then they iteratively test the enrichment of each annotation term for each gene and finally the select the enriched annotation terms that satisfy the P-value threshold. The P-value threshold measure that probability that genes of the input list have a given annotation term with respect to a random chance, measured against a null model. The strength of SEA algorithms is their simple approach and the efficiency (Huang *et al.*, 2009). However as noted by Huang *et al.* a weakness is that the annotation term that pass the enrichment threshold may be very large.

The GSEA tools have the same principle of SEA but they differ on the calculation of the enrichment P-values (Cannataro *et al.*, 2015). The main idea of GSEA is to select all the genes in an experiment (not only the user-selected), therefore they reduce possible artefacts due to the gene selection step. Then they use all the genes to calculate the enrichment score and genes are ranked. MEA use the enrichment calculation method of SEA and incorporates extra network discovery algorithms by considering the term-to-term relationships (Di Martino *et al.*, 2015).

Singular Enrichment Analysis Algorithms and Tools

Singular Enrichment has been the first strategy for enrichment analysis. All the algorithms that implements this strategy take the user's preselected genes, and then iteratively test the enrichment of each annotation term one-by-one in a linear mode. After this step, each annotation term is associated to an enrichment probability (enrichment P-value) and each term passing the P-value threshold is reported to the user. Results are usually reported in a tabular format.

The enrichment P-value calculation is performed by using well-known statistical methods such as including Chi-square, Fisher's exact test, Binomial probability and Hypergeometric distribution. Examples of SEA tools are GoMiner, Onto-Express, DAVID, and GOEAST (see Relevant Websites section). Main weakness of tools in this class is that the linear output of terms can be very large and interrelationships of relevant terms are not evidenced.

GSEA Algorithms and Tools

GSEA approach is similar to SEA, i.e., the analysis of each annotation term separately, and uses a distinct algorithm to calculate enrichment P-values. The rationale of GSEA is to takes into account all the genes of an experiments while SEA selected only a subset of user selected genes. This strategy reduces the arbitrary factors in the typical gene selection step. GSEA algorithms calculates a maximum enrichment score (MES) from the rank order of all gene members in the annotation category. Then enrichment P-values are calculated by matching the MES to randomly shuffled MES distributions. Examples of enrichment tools in

the GSEA class are ErmineJ, and FatiScan (see Relevant Websites section). A complete list may be found at <https://omictools.com/gene-set-analysis-category>.

MEA Algorithms and Tools

MEA inherits the basic enrichment calculation found in SEA and incorporates extra network discovery algorithms. In this way MEA approaches are able to discover term-to-term relationships. The key advantage of this approach is that the researcher can take advantage of term-term relationships. The rationale is that relationships takes into account the biological meaning of the set of terms for a study. Examples of MEA tools are Ontologizer, topGO, and GENECODIS (see Relevant Websites section). A complete list may be found at <https://omictools.com/onttool>.

See also: Biological and Medical Ontologies: Systems Biology Ontology (SBO). Computational Tools for Structural Analysis of Proteins. Expression Clustering. Functional Enrichment Analysis. Functional Genomics. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Ontology-Based Annotation Methods. Protein Functional Annotation. Semantic Similarity Definition

References

- Ashburner, M., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* 25 (11), 25–29. doi:10.1038/75556.
- Cannataro, M., Guzzi, P.H., Milano, M., 2015. GoD: An R-package based on ontologies for prioritization of genes with respect to diseases. *Journal of Computational Science* 9, 7–13.
- Di Martino, M.T., *et al.*, 2015. Integrated analysis of microRNAs, transcription factors and target genes expression discloses a specific molecular architecture of hyperdiploid multiple myeloma. *Oncotarget* 5.
- Guzzi, P.H., *et al.*, 2012. Semantic similarity analysis of protein data: Assessment with biological features and issues. *Briefings in Bioinformatics* 13 (55), 569–585. doi:10.1093/bib/bbr066.
- Huang, D.W., Sherman, B.T., Lempicki, R.A., 2009. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research* 37 (11), 1–13. doi:10.1093/nar/gkn923.
- Roy, S., Guzzi, P.H., 2015. Biological network inference from microarray data, current solutions, and assessments. *Microarray Data Analysis* 1375, 155–167. doi:10.1007/7651_2015_284.

Relevant Websites

<https://bioconductor.org/packages/release/bioc/html/topGO.html>
Bioconductor.

<https://biportal.bioontology.org/>
BioPortal.

<https://david.ncifcrf.gov/>
DAVID Bioinformatics Resources.

<http://erminej.msl.ubc.ca/erminej>

<http://genecodis.cnb.csic.es/>
GeneCodis.

<https://github.com/phenotips/ontologizer>
GitHub.

<http://omicslab.genetics.ac.cn/GOEAST/>
GOEAST.

<https://discover.nci.nih.gov/gominer/index.jsp>
GOMINER.

<http://genome.crg.es/GOToolBox/>
GOToolBox.

<https://omictools.com/fatiscan-tool>
OMIC Tools.

<https://omictools.com/gene-set-analysis-category>
OMIC Tools.

<https://omictools.com/onttool>
OMIC Tools.

Gene Prioritization Using Semantic Similarity

Erinija Pranckevičienė, Vilnius University, Vilnius, Lithuania

© 2019 Elsevier Inc. All rights reserved.

Introduction

The year 2003 marked the completion of the Human Genome Project, and along with it a new scientific frontier. What immediately followed was an explosion of new research, possibilities, and substantial challenges. Where exactly do we find ourselves after more than a decade?

We are realizing previously unthinkable practical applications. Given the speed of molecular profiling technologies, personal genomics based medicine is becoming increasingly accessible (Dudley and Karczewski, 2013). Personalized precision medicine already offers individualized approaches to treatments according to individual genetic makeups (Fernald *et al.*, 2011). Beyond general public healthcare, direct to consumer (DTC) companies can sequence personal genomes upon request (Angrist, 2016). Today, in almost every area of biology and medicine the central question becomes: which gene and molecular markers associate with, or explicitly cause observed traits/phenotypes, and how? As of this writing, different modalities of high-throughput molecular profiling technologies are used to collect data and tackle these research areas. Microarrays and RNA-sequencing are used to analyze expression of genes in changing conditions. Chromatin-immunoprecipitation followed by parallel next generation sequencing (ChIP-Seq) is used for chromatin analysis and gene regulation. Genome and exome sequencing are used to study variation in the genome. Single cell sequencing is used to understand processes simultaneously occurring in a cell. All these technologies also generate huge amounts of data that has to be processed to extract useful knowledge (Tiffin *et al.*, 2009; Hawkins *et al.*, 2010).

The true value of this data depends on our ability to make correct links between molecular entities and the information stored in databases and literature (DeIaIglesia *et al.*, 2013). An unprecedented amount of information has to be processed to elucidate smaller subsets of essential genes or genomic variants related to the phenotype or condition. Just as a brief example - a single microarray experiment on mice brains and genetic susceptibility to stress may produce hundreds of differentially expressed genes. However, only a small subset of genes will actually contribute to the increased stress sensitivity. Hence the challenge - in genetic diagnostics, many single nucleotide polymorphisms (SNPs) can be identified in patient's genome/exome, yet only few are ever related to the observed trait and have explanatory power when it comes to the symptoms (Eilbeck *et al.*, 2017).

Here we come to the particularly difficult point of the research and methodologies described below. Reviewing all potential candidates or following a wrong lead might be extremely time consuming and costly (Hassani-Pak and Rawlings, 2017; Moreau and Tranchevent, 2012). However, selecting a small subset of best candidate genes might be very challenging, because one needs to interpret many complex relationships between observed phenotypes and candidate genes. To aid our search for the best candidate genes related to our target traits and phenotypes, we can now leverage modern computational tools. What follows is a thorough description of one such powerful method. The hope is that by utilizing this methodology in gene prioritization and combining it with other state-of-the-art tools, modern researchers and bioinformaticians will be able to substantially reduce time and errors in genetic discovery.

The importance of computational gene prioritization is evidenced by dynamics of scientific publications. Web of Science database has over two hundred articles published since 2005 (accessed in September 2017) whose primary topic is computational gene prioritization and 30% of these articles were published starting 2014 showing increasing interest in this field.

Computational gene prioritization aims to identify the most promising gene candidates related to a phenotype of interest from a large list of genes. In this process the connections between genes and biological processes of interest are derived and the genes are ranked according to the strength of those connections. The top ranking genes are the suggested best candidates (Moreau and Tranchevent, 2012; Bomberg, 2013).

Fundamental features that characterize a gene prioritization algorithm are (i) how a phenotype is defined and (ii) how an algorithm identifies links between genes and phenotypes. The definition of a phenotype will determine rules by which the algorithm will mine available data resources to retrieve gene-phenotype links. In practice, researchers try to use as many resources as possible. Information types and databases used to guide gene prioritization consist of already known genotype-phenotype relationships and its supporting evidence, homology between different species, gene expression data, interactions between the molecular entities and functional annotations of genes and gene variants (Aerts *et al.*, 2006; Hassani-Pak and Rawlings, 2017). Research teams may not only use conventional genomic data to prioritize gene candidates, but also images of gene expression maps in tissues (McClelland and Yao, 2017).

Recently established Monarch Initiative started integrating data from a large number of diverse resources for human and model organisms to aid in disease mechanism discovery and diagnosis (Mungall *et al.*, 2017) and the Monarch Initiative website. It unites large research centers and organizations worldwide leveraging resources of International Mouse Phenotyping Consortium (IMPC), Mouse Genome Database (MGI), Online Mendelian Inheritance in Man (OMIM) and portal of rare diseases Orphanet to integrate data describing diseases, phenotypes, environmental factors, drugs, literature, research resources and other.

Algorithmic approaches to gene prioritization can be classified into several broad categories – those that query existing resources and try to establish a chain of evidence linking genes to phenotypes (Perez-Iratxeta *et al.*, 2002; Pranckeviciene, 2015) and those that are based on principle of “guilty by association” (Wang and Marcotte, 2010). The latter is a dominating approach in most popular gene prioritization tools such as ToppGene (Chen *et al.*, 2009) and Endeavour (Tranchevent *et al.*, 2016; Moreau and

Tranchevent, 2012). Using functional properties of genes known to be associated with a disease phenotype of interest called training genes, the candidate genes are ranked by comparing them to the training genes (Valentini *et al.*, 2014; Bomberg, 2013). The comparison is carried out in all domains describing gene functions by assessing similarity of sequences, molecular interactions, biological pathways, gene product protein family domains and functional annotations and then aggregating the evidence (Aerts *et al.*, 2006) to prioritize the candidates (Moreau and Tranchevent, 2012). Extensive review of gene prioritization tools is presented in another article of this compendium by Marianna Milano.

This chapter is about analysis of complex biomedical data and prioritization of causative genes using semantic similarity searches in large hierarchical databases. We will outline a detailed example how Semantic Similarity can be applied to measure functional closeness of phenotypes and genes. We will demonstrate that it can serve as a powerful instrument in comparing annotations and groups of annotations organized as a hierarchical taxonomy.

Ontology Represents Organized Knowledge

First definition of ontology in computer science was given by Tom Gruber as “explicit specification of conceptualization” (Gruber, 2009). In philosophy an ontology is a way to understand and describe world. In computational and information sciences it is an instrument that enables modeling of knowledge about some real or imagined domain. Ontology provides machine readable language and method to structure and semantically describe a knowledge domain independently from its computational implementation. It can be understood as a controlled vocabulary which defines and specifies concepts, entities, relationships between them and other information relevant in modeling of that knowledge domain (Gruber, 2009; Hoehndorf *et al.*, 2015). Curated biomedical ontology contains organized knowledge that was reviewed by human field specialists so that ontology terms and their relationships are well defined. Most of biomedical ontologies are available for download from OBO Foundry – Open repository of biomedical ontologies (Smith *et al.*, 2007) and OBO Foundry website.

Gene Ontology

The most important ontology in biomedical and molecular sciences is Gene Ontology (GO). In GO knowledge about functions, compartments and biological processes in which known genes of all researched organisms take part is organized into three domains: Molecular Function, Biological Process and Cellular Component. These three domains are structured as taxonomic graphs in which the general terms precede the more specific terms. Each term in the taxonomy is assigned a unique identifier (for example cell communication is GO:0007154) and it can have multiple parents. The root concepts Molecular Function, Biological Process and Cellular Component are at the top of the hierarchy (Ashburner *et al.*, 2000) and Gene Ontology website.

A major purpose of Gene Ontology is to organize functional molecular knowledge about genes so that each gene can be characterized by what it does through Gene Ontology annotations/terms. As new gene functions are discovered or previously known functions are revised the corresponding genes are assigned a new or revised annotation. Each gene has its GO annotations listed in National Center of Biotechnology Information (NCBI) Gene database, Ensembl and other major gene information databases.

The graph structure of GO allows comparisons of GO terms and GO-annotated gene products by Semantic Similarity. The closer the two terms are in the GO graph, the more similar meanings they have. Comparison of GO term meanings helps in predicting/identifying protein–protein interactions, suggesting candidate genes involved in diseases and evaluating the functional coherence of gene sets (Pesquita *et al.*, 2009; Couto and Pinto, 2013).

Human Phenotype Ontology

Human phenotype ontology (HPO) is central in medical genetics and genomics. It provides “comprehensive bioinformatic resource for analysis of human diseases and phenotypes and serves as a computational bridge between genome biology and clinical medicine” (Köhler *et al.*, 2017) and HPO website. HPO provides a standardized vocabulary of abnormalities observed in human diseases. Its terms stand for well defined anatomical and functional properties that are used by medical practitioners to describe syndromes and clinical phenotypes. For the first time Semantic Similarity was used on HPO terms practically to aid in differential diagnosis of genetic diseases (Köhler *et al.*, 2009). HPO is under constant maintenance and development and its terms are being integrated into major online resources for medical genetics such as Decipher (Firth *et al.*, 2009) and PhenoTips (Girdea *et al.*, 2013) to annotate human diseases and to describe clinical symptoms.

HPO became a part of Monarch initiative (Mungall *et al.*, 2017) directed at integration of heterogeneous molecular biology data resources to aid diagnostics of genetic disorders. Because of its significant overlap with Mammalian Phenotype Ontology, HPO enables comparison of phenotypes between model organisms of different species by estimating their Semantic Similarity (Robinson *et al.*, 2014; Smedley and Robinson, 2015). This is very important because knowledge of gene-phenotype relationships discovered and confirmed in model organisms can inform unresolved gene-phenotype relationship cases in humans (Oellrich *et al.*, 2012; Robinson and Webber, 2014; Haendel *et al.*, 2015).

Semantic Similarity

Semantics (originating from Greek) denotes a philosophical and linguistic study of meaning. Semantic Similarity (SS) measures likeness of meaning of two concepts. For example, “an ocean” and “a sea” are semantically similar since both mean a large

continuous body of salt water. Concepts of “pears” and “apples” are semantically similar because both are tree-grown fruits. Proximity between meanings of concepts can be estimated numerically. It is computed using mathematical formulas that define Semantic Similarity (Sánchez and Batet, 2011). Notable domains in which a Semantic Similarity has found useful applications are: Natural Language Processing (NLP), Knowledge Engineering (KE) and Bioinformatics (Harispe et al., 2015):

- In NLP Semantic Similarity is used to compare units of language (words, phrases, sentences). It is applied to word sense disambiguation, text summarization, building textual corpora, semantic annotation and designing question answering systems.
- KE focuses on building machine-understandable knowledge representations such as controlled vocabularies and ontologies. Semantics of concepts in ontologies is already encoded by making relationships between the concepts explicit. Semantic Similarity in KE domain mostly is used to integrate, link and align different ontologies.
- In Bioinformatics a majority of the data is already structured and organized into ontologies (GO, HPO and other) that are available from Bio Portal (Whetzel et al., 2011). Semantic Similarity here mostly is used to analyze functional similarity of genes or phenotypes, to aid interpretation of poorly characterized gene functions, to guide clinical diagnostics and to help drug design (Harispe et al., 2015; Sánchez and Batet, 2011).

Formal Definitions

A problem of Semantic Similarity estimation in taxonomic networks dates back to seventies. The similarity in meaning of two concepts used to be estimated by counting edges connecting the concepts in the taxonomy. It was based on assumption that concepts connected by shorter paths have also similar meanings. However, this is not entirely true.

Philip Resnik was first to propose a different simple measure of SS of two concepts based on information content (IC) (Resnik, 1995). It measured how much information two concepts share in a hierarchical taxonomy. His approach was based on the generality of the concept which is expressed as a probability of its occurrence. For example, a root concept of the taxonomy such as Biological Process (BP) in Gene Ontology has a probability of occurrence equal to 1, since all other concepts in BP ontology are subsumed by this concept and also are biological processes. Therefore, Biological Process does not convey any information and has zero information content. More specific concepts down the hierarchy become narrower in meaning and have gradually higher information content. The SS between two concepts then is defined as the information content of the most informative (closest) common ancestor (subsumer) of both concepts. In this case, the closeness in meaning originates from the structure of the knowledge graph and a length of a path connecting the concepts becomes secondary. Therefore, a structure of the taxonomy/hierarchical graph became central in numerical estimation of SS between terms (Resnik, 1999). Recent theoretical work postulates an equivalence between information-theoretic definitions of Semantic Similarity and definitions in set theory (Sánchez and Batet, 2011).

Mathematical Definition

Given an ontology organized as a taxonomic graph its terms are used to annotate entities in some domain. For example, genes are annotated by the terms of Gene Ontology and human diseases are annotated by the terms of Human Phenotype Ontology. The purpose of annotation is to characterize entities in the domain. Some terms in the ontology are more specific and some are more general. Suppose that the terms in ontology O annotate entities in the domain D . Information Content (IC) of the term suggested by (Resnik, 1995) measures how specific and informative the i -th ontology term $_i$ is in its domain:

$$IC(\text{term}_i) = -\log(p(\text{term}_i)) \quad (1)$$

The $p(\text{term}_i)$ in Eq. (1) is a probability (relative frequency) of occurrence of the term $_i$ from the ontology O in the domain D . Information content of the terms in the ontology is always relative to the domain entities of which they annotate. Consider genes of some organism. Each gene is annotated by GO terms of Molecular Function, Biological Process and Cellular Component. The GO terms are organized hierarchically meaning that if a gene is annotated by the term $_i$, then it is also annotated by all ancestors of that GO term $_i$ and has all functional properties that the ancestor annotations designate. The probability of the term's occurrence is measured by a ratio of a number of genes the term and its descendants annotate over a number of all annotated genes in the domain:

$$p(\text{term}_i) = (\text{genes_annotated_by_the_term}_i \text{ and_its_descendants}) / (\text{all_annotated_genes_in_the_domain}) \quad (2)$$

Since every gene is annotated by the root term Biological Process and by all descendants of the root term, then the probability of occurrence of the Biological Process term in the Gene Ontology equals to 1 because every gene is involved in some Biological Process. Therefore, the Information Content of Biological Process annotation is 0.

The hierarchical structure of ontology makes possible a computation of Semantic Similarity. Usually, the annotations that have related meanings are also in close proximity in the hierarchical taxonomy. The most general concepts are at the top of the hierarchy. As we go from the top to the bottom the terms become more and more specific. Resnik suggested to measure SS between the ontology terms u and v as:

$$\text{Sim}_{\text{Resnik}}(u, v) = IC(\text{MICA}_{u,v}) \quad (3)$$

MICA abbreviates Most Informative Common Ancestor. Resnik's measure does not take into account the IC of the estimated concepts. Lin, Jiang and Conrath, Pirro and Euzénat and Mazandu and Mulder refined Resnik's measure to incorporate the IC of measured concepts. These modifications are reviewed in depth by (Harispe et al., 2014; Mazandu et al., 2016). More on SS

measures can be found in another article of this compendium “Tools for semantic analysis based on semantic similarity”. Out of many measures of Semantic Similarity the one that is used in all practical applications is Lin’s measure (Lin, 1998). It takes into account Information Content of both: the terms and their MICA:

$$\text{Sim}_{\text{Lin}}(u, v) = 2 * \text{IC}(\text{MICA}_{u,v}) / (\text{IC}(u) + \text{IC}(v)) \quad (4)$$

Measures of Semantic Similarity and their biomedical applications are extensively surveyed in work of Couto and Pinto, (2013); Sánchez and Batet, (2011); Sánchez *et al.*, (2012); Mazandu *et al.*, (2016); Pesquita *et al.*, (2009); Pesquita, (2017).

Computation in Ontologies

Semantic Similarity can be computed between the two terms in the ontology as well as between the annotated entities of the domain. Köhler *et al.* (2009) explains in depth how a principle of Semantic Search in Ontologies can be applied to differentiate diagnosis between Noonan Syndrome and Optic Syndrome using observed phenotypic abnormalities of Hypertelorism and Downward slanting palpebral fissures in an individual. In the following we compute explicitly and apply Information Content (Eq. 1) and Semantic Similarity (Eqs. 3 and 4) to compare syndromes defined by terms of Human Phenotype Ontology and to compare their associated genes annotated by Gene Ontology terms.

Semantic Similarity in HPO

The HPO curated terms currently annotate 1,0204 OMIM and ORPHANET diseases (HPO Website accessed December 2017). An example of a graph relating concepts in HPO is presented in Fig. 1. Left panel A in Fig. 1. shows a hierarchical structure relating a root term Phenotypic abnormality with the Abnormality of cerebral artery.

The terms become more and more specific going from the top to the bottom of the hierarchy. Each term represents some abnormality that may be encountered in some disease. The phenotypic abnormality specific terms are assigned to the diseases in which that abnormality may be manifesting. Therefore, each term in HPO annotates some number of the diseases. The root – Phenotypic abnormality – is assigned to all diseases currently annotated by HPO terms. Information Content of each term reflects its specificity and can be computed using Eq. 1. In order to compute Information Content of a term we need to know how many diseases are annotated by this term. A frequency of the HPO term means a proportion of diseases it annotates among all annotated diseases. Outline of computation of Information Content for five terms shown in Fig. 1 is presented in Table 1.

The terms *Abnormality of the vasculature* (HP:0002597) and *Abnormality of brain morphology* (HP:0012443) have broader meanings and are super classes for other terms. Therefore, they are annotated to more diseases than their descendants. This fact is reflected also in a lower IC of these terms compared to their descendant subclass terms.

Diseases in HPO are annotated with both: specific terms and their super class terms. For example, the term *Interrupted aortic arch* is a phenotype annotated to OMIM:616920 disease *Heart and brain malformation syndrome*. Another phenotype *Abnormality of the vasculature* is also annotated to the same OMIM:616920 disease just because it is a superclass of the *Interrupted aortic arch* phenotype. Similarly, the phenotype *Abnormality of brain morphology* is annotated to OMIM:616875 disease *Cerebral atrophy, visual impairment and psychomotor retardation* because this phenotype is a superclass of *Cerebral atrophy* that occasionally manifests in this OMIM:616875 disease (Source Monarch Initiative website). Superclass term is descriptive but usually it has a broader meaning.

Symptoms and syndromes organized into HPO ontology offer a framework to differentiate between competing diagnosis. The ontology enables to ask and answer questions such as: if a patient has *Abnormality of cerebral artery* (HP:0009145) and other symptoms that can be described by HPO terms, then which syndrome offers a best explanation of the observed symptoms? To find the matching syndrome, HPO terms of the patient’s phenotype are compared to HPO terms annotated to all diseases. A syndrome/disease that has highest Semantic Similarity to the patient’s phenotypic terms is suggested as a most viable.

We will use the *Abnormality of cerebral artery* to prioritize between the two diseases - *Heart and brain malformation syndrome* and *Cerebral atrophy, visual impairment and psychomotor retardation*. We will compare this *Abnormality of cerebral artery* phenotype with the HPO terms that are annotated to the mentioned diseases (shown in HPO graph in Fig. 1).

Semantic Similarity between the two terms in ontology is measured by Information Content of their most informative common ancestor (MICA) as defined in Eqs. (3) and (4). To start differentiating between the diseases we first find MICA of $u = \text{Abnormality of cerebral artery}$ and $v = \text{Abnormality of the vasculature}$. The terms u and v are connected by the two paths in Fig. 1. One path passes through the *Abnormality of the systemic arterial tree*. Another path goes through the *Abnormality of the cerebral vasculature*. Both these terms are closest common ancestors for u and v . However, we seek the most informative common ancestor and it would be the term with the highest Information Content. Calculations presented in Table 1 show that the term *Abnormality of the cerebral vasculature* has higher IC=3.87 than the IC=3.17 of *Abnormality of the systemic arterial tree*. Therefore, the most informative common ancestor is *Abnormality of the cerebral vasculature*. Through this ancestor only the patient’s phenotype *Abnormality of cerebral artery* connects with the term *Abnormality of brain morphology* that is a syndrome that is manifesting in and annotated to the disease *Cerebral atrophy, visual impairment and psychomotor retardation*. And through this ancestor only the same patient’s phenotype connects with another syndrome of *Abnormality of vasculature* that manifests in and is annotated to the other disease *Heart and brain malformation syndrome*.

Assuming that this is all that we know and applying Eq. (3) we compute Information Content of the most informative common ancestor $\text{IC}(\text{Abnormality of the cerebral vasculature}) = 3.87$ that is Resnik’s Semantic Similarity. From it alone we can’t decide whether the *Abnormality of cerebral artery* is closer in meaning to the *Abnormality of the vasculature* or to the *Abnormality of brain morphology* because its IC is the same 3.87 for both cases. However, using Lin’s Semantic Similarity defined by Eq. (4) we can

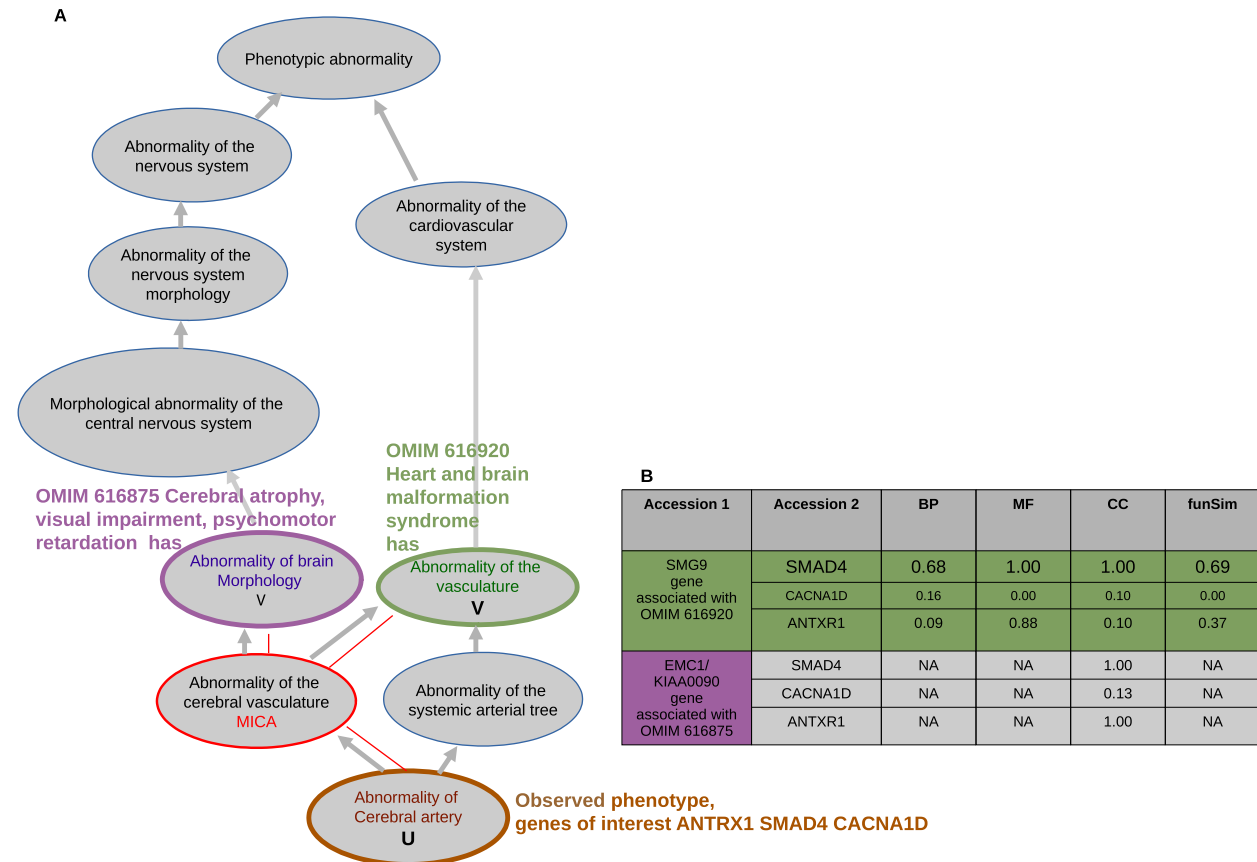


Fig. 1 Hierarchy of relationships between Human Phenotype Ontology terms and SS between genes. (A) Graph shows semantic relationships between the observed phenotype Abnormality of cerebral artery (U) and its ancestors. Among them, the Abnormality of brain morphology (V) is associated with OMIM disease Cerebral atrophy, visual impairment and psychomotor retardation which has an identified causative gene EMC1 (alternative name KIAA0090). The ancestor Abnormality of the vasculature (V) is a symptom observed in OMIM disease Heart and brain malformation syndrome. Its associated gene is SMG9. The observed phenotype U is linked to the phenotypes manifesting in diseases through the Abnormality of cerebral vasculature that is the Most Informative Common Ancestor (MICA) for the linked terms V, V and U. The observed phenotype is hypothesized to be associated with ANTRX1, SMAD4 and CACNA1D genes which we want to prioritize with respect to OMIM diseases. (B) Table shows Lin's SS estimates. Columns represent similarity between gene annotations of Biological Process (BP), Molecular Function (MF) and Cellular Component (CC) and the funSim column is an aggregated similarity score. Since EMC1 gene is poorly annotated it can't be meaningfully compared with other genes. It is represented by NA cells of similarity values between genes.

Table 1 Human Phenotype Ontology terms and their computed Information Content

HPO identifier	HPO term	# annotated diseases	IC = $-\log(\# \text{ annotated diseases}/n)$
HP:0009145	Abnormality of cerebral artery	72	4.95
HP:0100659	Abnormality of the cerebral vasculature	212	3.87
HP:0011004	Abnormality of the systemic arterial tree	429	3.17
HP:0002597	Abnormality of the vasculature	1578	1.87
HP:0012443	Abnormality of brain morphology	2956	1.24

Source: Total number of diseases annotated by HPO terms is $n=1,0204$ (last accessed September 2017).

take into account the informativeness of u and v :

$$\begin{aligned} \text{Sim}_{\text{Lin}}(u = \text{Abnormality of cerebral artery}, v = \text{Abnormality of the vasculature}) \\ = 2 * \text{IC}(\text{MICA}_{u,v}) / (\text{IC}(u) + \text{IC}(v)) = 2 * 3.87 / (4.95 + 1.87) = 1.13. \end{aligned}$$

$$\begin{aligned} \text{Sim}_{\text{Lin}}(u = \text{Abnormality of cerebral artery}, v = \text{Abnormality of brain morphology}) \\ = 2 * \text{IC}(\text{MICA}_{u,v}) / (\text{IC}(u) + \text{IC}(v)) = 2 * 3.87 / (4.95 + 1.24) = 1.25. \end{aligned}$$

With Lin's measure we are able to differentiate between these two terms. Based on a current knowledge existing in HPO, the *Abnormality of cerebral artery* is semantically closer to the *Abnormality of brain morphology* and therefore, the observed phenotypic evidence is better explained by the *Cerebral atrophy, visual impairment and psychomotor retardation* disease. We arrived at this conclusion using estimates of Semantic Similarity given that all that is known is *the phenotype observed in the individual that is Abnormality of cerebral artery*. In practice the phenotypes of the individuals are described by more HPO terms. The same reasoning and computation of Sim_{Lin} is applied to all pairs of HPO terms: where the pair contains the HPO term describing patient's phenotype and a term annotated to a disease. Then individual Sim_{Lin} estimates between the pairs of the terms computed for each disease are averaged for that disease (Köhler *et al.*, 2009). The disease characterized by a maximum average Sim_{Lin} would be suggested as the one which explains the observed symptoms best, based on the current knowledge that is present in the ontology.

Semantic Similarity in GO

Gene Ontology annotations describe functional properties of genes and biological processes in which genes are involved. Semantic Similarity can measure functional closeness of Gene Ontology (GO) annotations and functional similarity of annotated gene products. In most applications similarity between gene products is measured aiming to prioritize disease causing genes. Major sources of known associations between genes and diseases are the Online Mendelian Information about Man (OMIM) and the database of rare diseases ORPHAnet both of which document genetic diseases and their causes. Phenotypes in HPO have associated genes based on evidence in OMIM and ORPHAnet. For example, the *Abnormality of the cerebral vasculature* (Fig. 1) is associated with 552 genes. The SMAD4 and CACNA1D are among those genes and will serve as an example here. SMAD4 also is associated with the *Abnormality of the systemic arterial morphology* phenotype together with other 972 genes. Descriptions of these genes are shown in Table 2 (Monarch Initiative and Gene Ontology websites accessed 2018 February).

The SMAD4 and CACNA1D genes are associated with vascular phenotypes but not only. Gene ontology annotations of these genes show that they are related to the function and development of heart. These GO annotations are summarized in Table 3. Other GO annotations of these genes can be reviewed in Gene Ontology, Quick GO or NCBI Gene or UniProt websites.

Using previously described approach we can compute functional similarity between individual GO annotations of genes and their products. To aid this computation we use online FunSimMat tool (Schlicker *et al.*, 2010). Results of comparison of functional similarity between the selected GO terms from the Table 3 employing Lin's measure are shown in Table 4.

Two terms closest in meaning by $Sim_{Lin}=0.77$ are the *Regulation of atrial cardiac muscle cell membrane repolarization* and the *Cardiac conduction*. Semantic Similarity value computation relies on the knowledge encoded in the ontology (Table 4 Row 1) and

Table 2 Characteristics of genes associated with HPO phenotypes

Gene	Name	Total number of GO annotations	Entrez ID	UniProtKB ID	Number of associated phenotypes
SMAD4	SMAD family member 4	119	4089	Q13485	209
CACNA1D	Calcium voltage-gated channel subunit alpha1 D	36	776	Q01668	36

Table 3 Selected biological process annotations of CACNA1D and SMAD4 genes

Gene association with phenotype	Annotation	GO identifier	Annotated gene products
SMAD4 <i>Abnormality of the systemic arterial tree</i>	Positive regulation in cell proliferation involved in heart valve morphogenesis	GO:0003251	30
	Any process that increases the rate, frequency or extent of cell proliferation that contributes to the shaping of a heart valve		
	Cardiac septum development	GO:0003279	2663
	The progression of a cardiac septum over time, from its initial formation to the mature structure		
CACNA1D <i>Abnormality of the cerebral vasculature</i>	Brainstem development	GO:0003360	101
	The progression of the brainstem from its formation to the mature structure. The brainstem is the part of the brain that connects the brain with the spinal cord		
	Lipoprotein particle mediated signalling	GO:0055095	87
	A series of molecular signals mediated by the detection of a lipoprotein particle		
	Regulation of atrial cardiac muscle cell membrane repolarization	GO:0060372	127
	Any process that modulates the establishment or extent of a membrane potential in the polarizing direction towards the resting potential in an atrial cardiomyocyte		
	Cardiac conduction	GO:0061337	4264
	Transfer of an organized electrical impulse across the heart to coordinate the contraction of cardiac muscles. The process begins with generation of an action potential (in the sinoatrial node (SA) in humans) and ends with a change in the rate, frequency, or extent of the contraction of the heart muscles		

Table 4 Lin's Semantic Similarity between of selected GO annotations

Row	GO term 1	GO term 2	Lin SS
1	GO:0060372 Regulation of atrial cardiac muscle cell membrane repolarization (CACNA1D)	GO:0061337 Cardiac conduction (CACNA1D)	0.77
2	GO:0003279 Cardiac septum development (SMAD4)	GO:0003360 Brainstem development (SMAD4)	0.38
3	GO:0003279 Cardiac septum development (SMAD4)	GO:0061337 Cardiac conduction (CACNA1D)	0.36
4	GO:0003251 Positive regulation in cell proliferation involved in heart valve morphogenesis (SMAD4)	GO:0061337 Cardiac conduction (CACNA1D)	0.34
5	GO:0003251 Positive regulation in cell proliferation involved in heart valve morphogenesis (SMAD4)	GO:0060372 Regulation of atrial cardiac muscle cell membrane repolarization (CACNA1D)	0.31

this value fairly reflects that knowledge. We would expect the terms GO:0060372 *Regulation of atrial cardiac muscle cell membrane repolarization* and GO:0061337 *Cardiac conduction* to be semantically close since cardiac conduction involves membrane repolarization. Indeed, we see that these terms are closer semantically than the terms GO:0060372 *Regulation of atrial cardiac muscle cell membrane repolarization* and GO:0003251 *Positive regulation in cell proliferation involved in heart valve morphogenesis* (Table 4 Row 5). In the latter instance the two processes are biologically remote because morphogenesis belongs to development but regulation of membrane repolarization is characteristic to the continuous functioning of the heart. These terms are also far from each other in the GO hierarchy where their common ancestor GO:0050789 *Regulation of Biological Processes* is located almost at the root of the ontology. Since Gene Ontology has very complex structure of relationships between annotations, we refer interested reader to the Gene Ontology website Tools section AmiGo2. We recommend to use 'Visualization Creator' to obtain a visualization of hierarchical representation of relationships between a given subset of GO terms.

In gene prioritization by semantic similarities the annotations of different genes are compared to each other in pair wise manner. The maximum values of estimated semantic similarities are averaged resulting in a single similarity value for two genes/gene products that are compared (Schlicker et al., 2010). If a subset of genes is compared to a single gene, then the genes in the subset are ranked by a magnitude of the estimated similarity value. Top ranking genes will be most similar functionally to the gene with which they are compared. The actual measure of SS can vary. Most tools that are available have Resnik's, Lin's and Yang's measures of Semantic Similarity.

Lets assume a hypothetical situation that we observed a phenotype of *Abnormality of cerebral artery* and candidate genes SMAD1 (UniProt accession Q13485), CACNA1D(Q01668) and ANTXR1(Q9H6X2). We are interested in how similar these genes are to the causative disease genes - what is the extent of their functional similarity to the known causative disease genes for: OMIM:616920 disease *Heart and brain malformation syndrome* (associated gene SMG9 UniProt accession Q9H0W8) and OMIM:616875 disease *Cerebral atrophy, visual impairment and psychomotor retardation* (associated gene EMC1 gene UniProt accession Q8N766). To perform prioritization we use FunSimMat tool again that computes also semantic similarities between GO annotations of gene products given their UniProtKB accessions.

Results of prioritization are summarized on the right panel B in Fig. 1. Two Accessions columns show the compared genes. The BP, MF and CC columns show Lin's SS values in Biological Process, Molecular Function and Cellular Component classes. The funSim column is an aggregated overall similarity score (Schlicker et al., 2010). For the SMG9 gene known to be associated with the *Heart and brain malformation syndrome* the most functionally similar gene in the list of candidates is SMAD4. We note that this example is used only to demonstrate application of Semantic Similarity. A simplistic gene prioritization exercise serves only as a guide to illustrate computations. Though a relationship between the SMAD4 gene and the *Heart and brain malformation syndrome* may exist, its proof would require thorough analysis which is beyond of the scope of this reference work. As for the other case the causative EMC1 gene (whose alternative name is KIAA0090) does not have annotations for Biological Process and Molecular Function. Therefore, no meaningful comparison can be performed. Note, that EMC1 was identified as the causative gene in the exome sequencing assay (OMIM #616875). The EMC1 gene is an example of a less investigated and annotated gene at a time of writing. However, its annotation status will likely change in future. These examples demonstrate that Semantic Similarity search in ontologies is most powerful in well annotated and documented application domains.

Conclusion and Prospects

Gene prioritization field is actively researched and developed (Valentini et al., 2014). First ideas and papers on inferring links between genes and phenotypes in gene prioritization were published by Dr. P. Bork group early starting 1996. A notable example of the development in this field is Genes to Diseases system G2D (Perez-Iratxeta et al., 2002) to prioritize genes in inherited diseases. G2D system used knowledge in literature and RefSeq database as evidence to infer strongest links between genes and diseases. It used Semantic Similarity as an additional option to prioritize genes.

Another important system discussed widely in literature is gene prioritization tool Endeavour developed by Prof. Y. Moreau group (Tranchevent *et al.*, 2016). This system is based on “guilty by association” paradigm and requires training genes to define a functional characteristics of a phenotype of interest. Semantic Similarity is used to compare GO annotations as one channel of evidence. Other evidence comes from interactions, pathway, protein domain data and similarity scores are aggregated known as genomic data fusion algorithm (Aerts *et al.*, 2006).

Some systems were developed and supported no longer. Worth of mentioning are specific and broader systems that withstood challenges of time: Genie (Fontaine *et al.*, 2011), ToppGene (Chen *et al.*, 2009) and systems that utilize rich framework of R statistical computing environment (<https://www.r-project.org/>) (Cannataro *et al.*, 2015). Interested reader is also referred to the extensive review of gene prioritization tools in another article of this compendium.

Increasing use of next generation sequencing exome/genome data in clinical diagnostics is raising interest in tools of prioritization of genes and genomic variants (Eilbeck *et al.*, 2017; Yang *et al.*, 2015; Smedley and Robinson, 2015; Smedley *et al.*, 2015). Recently the phenotypic descriptions of symptoms in human diseases are becoming better formalized, organized and maintained as ontologies (Köhler *et al.*, 2017; Hoehndorf *et al.*, 2015; Girdea *et al.*, 2013). This makes easier to apply computational algorithms to gene prioritization (Masino *et al.*, 2014). Some genes in human are poorly annotated. However, this is not necessarily a case in model organisms (Haendel *et al.*, 2015). Semantic Similarity can be applied cross-species to translate known gene-phenotype relationships in model organisms to human organism and help to explain difficult cases (Oellrich *et al.*, 2012; Robinson *et al.*, 2014; Robinson and Webber, 2014). Application of Semantic Similarity to facilitate a use of biological knowledge cross-species proved to be very successful and underlies development of information resources for medical genetics in particular in Monarch initiative (Mungall *et al.*, 2017).

See also: Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Natural Language Processing Approaches in Bioinformatics. Ontology-Based Annotation Methods. Semantic Similarity Definition

References

- Aerts, S., Lambrechts, D., Maity, S., *et al.*, 2006. Gene prioritization through genomic data fusion. *Nature Biotechnology* 24 (5), 537.
- Angrist, M., 2016. Personal genomics: Where are we now? *Applied Translational Genomics* 8, 1–3.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25, 25–29.
- Bomberg, Y., 2013. Chapter 15: Disease gene prioritization. *PLOS Computational Biology* 9 (14), 1–16.
- Cannataro, M., Guzzi, P.H., Milano, M., 2015. GoD: An R-package based on ontologies for prioritization of genes with respect to diseases. *Journal of Computational Science* 9, 7–13.
- Chen, J., Bardes, E.E., Aronow, B.J., Jegga, A.G., 2009. ToppGene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Research* 37 (suppl_2), W305–W311.
- Couto, F.M., Pinto, H.S., 2013. The next generation of similarity measures that fully explore the semantics in biomedical ontologies. *Journal of Bioinformatics and Computational Biology* 11 (05), 1371001.
- Delalgesia, D., Garcia-Remesal, M., de la Calle, G., *et al.*, 2013. The impact of computer science in molecular medicine: Enabling high-throughput. *Current Topics in Medicinal Chemistry* 13 (5), 526–575.
- Dudley, J., Karczewski, K.J., 2013. *Exploring Personal Genomics*. Oxford University Press.
- Eilbeck, K., Quinlan, A., Yandell, M., 2017. Settling the score: Variant prioritization and Mendelian disease. *Nature Reviews Genetics* 18, 599–612.
- Fernald, G.H., Capriotti, E., Daneshjou, R., *et al.*, 2011. Bioinformatics challenges for personalized medicine. *Bioinformatics* 27 (13), 1741.
- Firth, H., Richards, S.M., Bevan, A.P., *et al.*, 2009. DECIPHER: Database of chromosomal imbalance and phenotype in humans using Ensembl resources. *American Journal of Human Genetics* 84 (4), 524–533.
- Fontaine, J.F., Priller, F., Barbosa-Silva, A., Andrade-Navarro, M.A., 2011. Genie: Literature-based gene prioritization at multi genomic scale. *Nucleic Acids Research* 39 (suppl_2), W455–W461.
- Girdea, M., Dumitriu, S., Fiume, M., *et al.*, 2013. PhenoTips: Patient phenotyping software for clinical and research use. *Human Mutation* 34, 1057–1065.
- Gruber, T., 2009. Ontology. In: Liu, Ling., Özsu, M. Tamer. (Eds.), *Encyclopedia of Database Systems*. US: Springer-Verlag, pp. 1963–1965.
- Haendel, M.A., Vasilevsky, N., Brush, M., *et al.*, 2015. Disease insights through cross-species phenotype comparisons. *Mammalian Genome* 26 (9–10), 548–555.
- Harispe, D., Ranwez, S., Janaqi, S., Montmain, J., 2015. Semantic similarity from natural language and ontology analysis. In: *Proceedings of the Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.
- Harispe, S., Sánchez, D., Ranwez, S., Janaqi, S., Montmain, J., 2014. A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain. *Journal of Biomedical Informatics* 48, 38–53.
- Hassani-Pak, K., Rawlings, C., 2017. Knowledge discovery in biological databases for revealing candidate genes linked to complex phenotypes. *Journal of Integrative Bioinformatics* 14 (1), 1–9.
- Hawkins, R.D., Hon, G.C., Ren, B., 2010. Next-generation genomics: An integrative approach. *Nature Reviews Genetics* 11 (7), 476–486.
- Hoehndorf, R., Shofield, P.N., Gkoutos, G.V., 2015. The role of ontologies in biological and biomedical research: A functional perspective. *Briefings In Bioinformatics* 16 (6), 1069–1080.
- Köhler, S., Vasilevsky, N.A., Engelstad, M., 2017. The human phenotype ontology in 2017. *Nucleic Acids Research* 45 (D1), D865–D876.
- Köhler, S., Schulz, M.H., Krawitz, P., *et al.*, 2009. Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *The American Journal of Human Genetics* 85 (4), 457–464.
- Lin, D., 1998. An information-theoretic definition of similarity. In: *ICML'98 Proceedings of the Fifteenth International Conference on Machine Learning*, 296–304. Morgan Kaufmann Publishers Inc.
- Masino, A.J., Dechene, E.T., Dulik, M.C., *et al.*, 2014. Clinical phenotype-based gene prioritization: An initial study using semantic similarity and the human phenotype ontology. *BMC Bioinformatics* 15 (248), 1–11.
- Mazandu, G.K., Chimusa, E.R., Mulder, N.J., 2016. Gene Ontology semantic similarity tools: Survey on features and challenges for biological knowledge discovery. *Briefings in Bioinformatics* 18 (5), 886–901. doi:10.1093/bib/bbw067.

- McClelland, K.S., Yao, H.H.-C., 2017. Leveraging online resources to prioritize candidate genes for functional analyses: Using the fetal testis as a test case. *Sexual Development* 11, 1–20.
- Moreau, Y., Tranchevent, L.C., 2012. Computational tools for prioritizing candidate genes: Boosting disease gene discovery. *Nature Reviews Genetics* 13 (8), 523–536.
- Mungall, C.J., McMurtry, J.A., Köhler, S., *et al.*, 2017. The Monarch Initiative: An integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Research* 45, D712–D722.
- Oellrich, A., Hoehndorf, R., Gkoutos, G.V., Rebholz-Schuhmann, D., 2012. Improving disease gene prioritization by comparing the semantic similarity of phenotypes in mice with those of human diseases. *PLOS ONE* 7 (6), e38937.
- Perez-Iratxeta, C., Bork, P., Andrade, M.A., 2002. Association of genes to genetically inherited diseases using data mining. *Nature Genetics* 31 (3), 316.
- Pesquita, C., 2017. Semantic similarity in the gene ontology. *The Gene Ontology Handbook*. 161–173.
- Pesquita, C., Faria, D., Falcao, A.O., Lord, P., Couto, F.M., 2009. Semantic similarity in biomedical ontologies. *PLOS Computational Biology* 5 (7), e1000443.
- Pranckeviciene, E., 2015. Procedure and datasets to compute links between genes and phenotypes defined by MeSH keywords [version 1; referees : 2 approved with reservations]. *F1000 Research* 4 (47).
- Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. In: Mellish, C.S. (Ed.), *IJCAI-95 Proceedings of the Fourteen International Joint Conference on Artificial Intelligence*, vols. 1–2. San Francisco, CA: Morgan Kaufmann Publishers Inc., pp. 448–453.
- Resnik, P., 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research* 11, 95–130.
- Robinson, P.N., Köhler, S., Oellrich, A., *et al.*, 2014. Improved exome prioritization of disease genes through cross-species phenotype comparison. *Genome Research* 24 (2), 340–348.
- Robinson, P.N., Webber, C., 2014. Phenotype ontologies and cross-species analysis for translational research. *PLOS Genetics* 10 (4), e1004268.
- Sánchez, D., Batet, M., 2011. Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. *Journal of Biomedical Informatics* 44, 749–759.
- Sánchez, D., Batet, M., Isern, D., Valls, A., 2012. Ontology-based semantic similarity: A new feature-based approach. *Expert Systems With Applications* 39 (9), 7718–7728.
- Schlicker, A., Lengauer, T., Albrecht, M., *et al.*, 2010. Improving disease gene prioritization using the semantic similarity of gene ontology terms. *Bioinformatics* 20 (18), 561–567.
- Smedley, D., Jacobsen, J.O., Jager, M., *et al.*, 2015. Next-generation diagnostics and disease-gene discovery with the Exomiser. *Nature Protocols* 10 (12), 2004.
- Smedley, D., Robinson, P.N., 2015. Phenotype-driven strategies for exome prioritization of human Mendelian disease genes. *Genome Medicine* 7 (1), 81.
- Smith, B., Ashburner, M., Rosse, C., *et al.*, 2007. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25, 1251–1255.
- Tiffin, N., Andrade-Navarro, M., Perez-Iratxeta, C., 2009. Linking genes to diseases: It's all in the data. *Genome Medicine* 1 (8), 77.
- Tranchevent, L.C., Ardeshirdavani, A., ElShal, S., *et al.*, 2016. Candidate gene prioritization with Endeavour. *Nucleic Acids Research* 44 (W1), W117–W121.
- Valentini, G., Paccanaro, A., Caniza, H., Romero, A.E., Re, M., 2014. An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. *Artificial Intelligence in Medicine* 61 (2), 63–78.
- Wang, P.L., Marcotte, E.M., 2010. It's the machine that matters: Predicting gene function and phenotype from protein networks. *Journal of Proteomics* 73 (11), 2277–2289.
- Whetzel, P.L., Noy, N.F., Shah, N.H., *et al.*, 2011. BioPortal: Enhanced functionality via new web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Research. Web Server Issue*, W541–W545.
- Yang, H., Robinson, P.N., Wang, K., 2015. Phenolyzer: Phenotype-based prioritization of candidate genes for human diseases. *Nature Methods* 12 (9), 841–843.

Relevant Websites

<http://www.funsimmat.de/>
FunSimMat.

<http://geneontology.org/>
Gene Ontology.

<http://human-phenotype-ontology.github.io/>
Human Phenotype Ontology.

<https://www.ncbi.nlm.nih.gov/>
National Center of Biotechnology Information Gene Database.

<https://www.genome.gov/10001772/all-about-the-human-genome-project-hgp/>
National Human Genome Institute.

<https://omictools.com/>
Omics Tools.

<https://www.ebi.ac.uk/QuickGO/>
Quick GO.

<https://monarchinitiative.org/>
The Monarch Initiative.

<http://www.obofoundry.org/>
The Open Biomedical Ontologies Foundry.

<http://www.uniprot.org/>
UniProt.

Gene Prioritization Tools

Marianna Milano, University of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The experiment platforms, such as mass spectrometry and microarray, usually produce a set of candidate genes potentially involved in different diseases (Cannataro *et al.*, 2013, 2010). Consequently, different approaches for selecting and filtering the genes have been introduced. These approaches are particularly used in clinical scenarios when the researchers are interested in the behavior of few molecules related to some specific disease.

Such approaches are known as gene prioritization methods. Gene prioritization aims at identifying the most promising candidate genes from a large list of candidates with respect to a biological process of interest, through the use of computational methods (Moreau and Tranchevent, 2012). Gene prioritization involves the following steps: the selection of a list of genes from the omics experiment, e.g. microarray or NGS; the gathering information about a disease from literature or from existing knowledge bases e.g. biological ontologies or repositories; the selection of a prioritization method; the production of a list of ranked genes. The Fig. 1. summarizes the Gene prioritization process.

In the past few years, many gene prioritization methods have been proposed, some of which have been implemented into publicly available tools that users can freely access and use.

These existing gene prioritization approaches can be classify according to the kind of input data, the used knowledge bases, and the selected prioritization methods. A fundamental step in gene prioritization consists of the integration of heterogeneous data sources related to biological knowledge.

Data sources are at the core of the gene prioritization problem since the quality of the predictions directly correlates with the quality of the data used to make these predictions. There exist different data sources such as protein-protein interactions, functional annotations, pathways, expression, sequence, phenotype, conservation, regulation, disease probabilities and chemical components. About functional annotations, reliable ontologies may be used as computer readable source of knowledge. Important examples of ontologies that may help the prioritization process are Gene Ontology (GO) (Gene Ontology Consortium, 2004), Human Phenotype Ontology (HPO) (Robinson *et al.*, 2008), and Disease Ontology (DO) (Schriml *et al.*, 2012). They offer both the organization of concepts through taxonomies and the association of a concept to gene products through annotations.

Generally, existing gene prioritization methods use ontologies only in a post-processing step after mining of textual data sources, e.g. the web. Moreover, different tools often use Gene Ontology, and one tool offers the possibility to use more than one ontology (Moreau and Tranchevent, 2012). Consequently, the need for the introduction of software tools able to help researchers to filter these lists on the basis of biological or clinical considerations (e.g. which molecules are more related to a given disease) arises.

Background/Fundamentals

This section recall background information on ontologies and semantic similarity related to gene prioritization process. Different prioritization tools use several ontologies during the prioritization by evaluating the similarity among the annotations of each

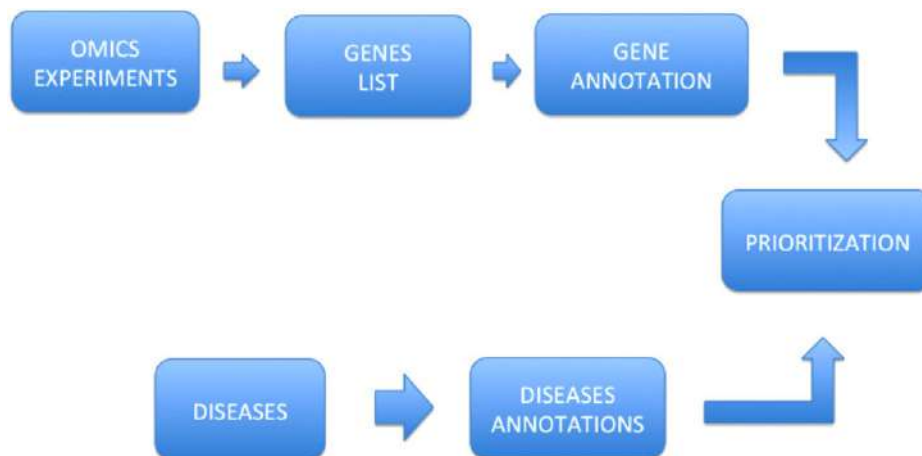


Fig. 1 Workflow prioritization process.

gene or gene product and of a selected disease. This is done by the use of semantic similarities (Guzzi *et al.*, 2012; Milano *et al.*, 2014). Semantic similarities are formal instruments to quantify the similarity of concepts in a semantic space. In the field of prioritization, semantic similarities are applied to quantify the similarity of a gene to a disease using the annotations of both elements on the same ontology.

Ontologies

Gene ontology

Gene Ontology (GO) is one of the main resources of biological information since it provides a specific definition of protein functions. GO is a structured and controlled vocabulary of terms, called GO terms. GO is subdivided into three non-overlapping ontologies: Molecular Function (MF), Biological Process (BP) and Cellular Component (CC). The structure of GO is a Directed Acyclic Graph (DAG), where the terms are the nodes and the relations among terms are the edges. The structure allows for more flexibility than a hierarchy since each term can have multiple relationships to broader parent terms and more specific child terms (du Plessis *et al.*, 2011). Genes or proteins are connected with GO terms through annotations by using a procedure also known as annotation process. Each annotation in the GO has a source and a database entry attributed to it. The source can be a literature reference, a database reference or computational evidence. Each biological molecule is associated with the most specific set of terms that describe its functionality. Then, if a biological molecule is associated with a term, it will connect to all the parents of that term (du Plessis *et al.*, 2011).

Human phenotype ontology

The Human Phenotype Ontology (HPO) is a standardized, controlled vocabulary that contains phenotypic information about genes or product genes. The HPO contains over 12,000 terms describing phenotypic features. The ontology is organized as three independent ontologies that include different categories: the mode of inheritance, the onset, and clinical course and the largest category of phenotypic abnormalities. The HPO is structured into a directed acyclic graph (DAG) in which the terms represent subclasses of their parent term and each term in the HPO describes a distinct phenotypic abnormality. About 110,301 annotations to HPO terms for almost 8000 diseases listed in the OMIM (Online Mendelian Inheritance in Man) database are also available at the website (see Relevant Website section). Diseases are annotated with terms of the HPO, meaning that HPO terms are used to describe all the signs, symptoms, and other phenotypic manifestations that characterize the disease in question. Since HPO contains information related to phenotypic abnormalities, the computation of semantic similarities among concepts annotated with HPO terms may enable database searches for clinical diagnostics or computational analysis of gene expression patterns associated with human diseases.

Disease ontology

The Disease Ontology (DO) database (see Relevant Website section) is a knowledge base related to human diseases. The current version stores information about 8000 diseases. The aim of DO is to connect biological data (e.g. genes) considering a disease-centred point of view. The DO semantically integrates a disease and existing medical vocabularies. DO terms and their DOIDs have been utilized to annotate disease concepts in several major biomedical resources. The DO is organized into eight main nodes to represent cellular proliferation, mental health, anatomical entity, infectious agent, metabolism and genetic diseases along with medical disorders and syndromes anchored by traceable, stable identifiers (DOIDs). Genes may be annotated with terms coming from DO that may be freely downloaded from the website. DO is structured into a directed acyclic graph (DAG), and the terms are linked by relationships in a hierarchy organized by interrelated subtypes. DO has become a disease knowledge resource for the further exploration of biomedical data, including measuring disease similarity based on functional associations between genes, and it is a disease data source for the building of biomedical databases.

Semantic Similarities

A semantic similarity measure (SSMs) is a formal instrument to quantify the similarity of two or more terms of the same ontology. Measures comparing only two terms are often referred to as pairwise semantic measures, while measures which compare two sets of term yielding a global similarity among sets, are referred to as groupwise measures. Since proteins and genes are associated with a set of terms coming from Gene Ontology, SSMs are often extended to proteins and genes. The similarity of proteins is then translated into the determination of similarity of a set of associated terms (Pesquita *et al.*, 2009; Wang *et al.*, 2010). Many similarity measures have been proposed (see for instance Guzzi *et al.* (2012) for a complete review) that may be categorized according to different strategies used for evaluating similarity.

For instance, many measures are based on information content (IC) of ontology terms. The information content of a term T of an ontology O is defined as $\log(p(c))$ where $p(c)$ is the number of concepts that are annotated with T and its descendants, divided by the number of all concepts that are annotated with a term of the same ontology. Measures based on a common ancestor first select a common ancestor of two terms according to its properties, and then evaluate the semantic similarity on the basis of the distance between the terms and their common ancestor and the properties of the common ancestor. IC can be used to select the proper ancestor, yielding to the development of mixed methods. The Resnik's similarity measure, sim_{res} , of two terms T_1 and T_2 of

GO is based on the determination of the information content (IC) of the their most informative common ancestor (MICA), where MICA is the common ancestor with the highest IC (Resnik, 1995):

$$sim_{Res} = IC(MICA(T_1, T_2))$$

A drawback of the Resnik's measure is that it considers mainly the common ancestor, and it does not take into account the distance among the compared terms and the shared ancestor. Lin's measure (Lin, 1998), sim_{Lin} , faces with this problem by considering both terms and yielding to the following formula:

$$sim_{Lin} = \frac{IC(MICA(T_1, T_2))}{IC(T_1) + IC(T_2)}$$

Jiang and Conrath's measure, sim_{JC} , takes into account this distance by calculating the following formula:

$$sim_{JC} = 1 - IC(T_1) + IC(T_2) - 2 \times IC(MICA(T_1, T_2))$$

Completely different from previous approaches, are the techniques based on path length (edge distance). In this case, similarity measures are correlated to the length of the path connecting two terms. Finally, many integrative approaches of two different categories have recently been proposed to achieve higher accuracy in measuring functional similarity of proteins. For example, Wang *et al.* (2010) proposed a combination of the normalized common term-based method and the path-length-based method. Their semantic similarity measure scores a protein pair by the common GO terms having the annotations of the proteins, but gives different weights to the common GO terms according to their depth.

Pesquita *et al.* (2008) proposed sim_{GIC} that integrates the normalized common-term-based method with information contents. Instead of counting the common terms, sim_{GIC} sums the information contents of the common terms.

$$sim_{GIC}(T_1, T_2) = \frac{\sum_{T_i \in C(T_1) \cap C(T_2)} \log P(T_i)}{\sum_{T_j \in C(T_1) \cap C(T_2)} \log P(T_j)}$$

where $C(T_1)$ is a set of all ancestor terms of T_1 . Finally, two recent IC based measures were proposed by Cho *et al.* (2013). The rationale is to integrate two orthogonal features. Since Resnik's method computes the information content of SCA (where SCA is the most specific common ancestor term) of two GO terms T_1 and T_2 , it focuses on their commonality, not the difference between them. In contrast, Lin's and Jiang's methods measure their difference only. sim_{ICNP} (information content of SCA normalized by path-length of two terms) uses the information content of common ancestors normalized by the shortest path length between T_1 and T_2 as the distance.

$$sim_{ICNP} = \frac{-\log P(T_0)}{len(T_1, T_2) + 1}$$

where T_0 is SCA of T_1 and T_2 . This method gives a penalty to Resnik's semantic similarity if T_1 and T_2 are located farther from their SCA. sim_{ICND} (information content of SCA normalized by difference of two terms' information contents) employs the information content of SCA normalized by the difference of information contents from the two terms to SCA, as Jiang's method does.

$$sim_{ICND} = \frac{-\log P(T_0)}{-\log P(T_0) - \log P(T_1) - \log P(T_2) + 1}$$

This method gives a penalty to Resnik's semantic similarity if the information contents of T_1 and T_2 are higher than the information content of their SCA.

Systems and/or Applications

The whole process of prioritization consists of different steps:

1. The first step consists of an omics experiment that produces a list of candidate genes identified through different technologies such as genomic microarray, expression microarray, next generation sequencing, micro-RNA microarray. The list can be filtered by using computational techniques, such as evidencing differentially expressed genes or potential mutations different in case/control studies.
2. The second step involves the collection of prior knowledge about the disease. Prior knowledge may be represented as keywords extracted from the literature using text-mining tools or as key- words selected by experts e.g. biologists or medical doctors. Analogously prior knowledge may be gathered from ontologies that represent a standardization of knowledge sources.
3. The third step consists of selecting the appropriate prioritization methods. The correct choice is crucial to obtain the best results.
4. The fourth step consists in the statistical evaluation of obtained results. After the validation ontologies may be used as a post-processing tool to evaluate the biological significance or the enrichment of selected candidate genes.

Analysis and Assessment

There exist many available approaches for gene prioritization that offer different functionalities to the community. Informations about these tools are summarized in The Gene Prioritization Portal (see Relevant Website section) (Börmigen *et al.*, 2012) that currently describes 46 tools that ensure the calculation of gene list prioritization.

This web site has been designed to help researchers to carefully select the tools that best correspond to their needs.

In general, the tools can differ in term of the inputs they require and the outputs they provide. There exist two types of inputs, the first one is the prior knowledge about the genetic disorder of interest. The prior knowledge regards the retrieval of a training set about at least, one disease causing gene or a complete set of keywords that covers most aspects of the disease.

The second type of input is the candidate search space, that consists of locus, or a differentially expressed genes (DEG) list, or the whole genome.

Regarding the outputs, two types were considered: a ranking of the candidate genes such that the most promising candidate can be found at the top and selection of the candidate genes that consists of a subset of the original candidate set, containing only the most promising candidates.

The first tool represents in The Gene Prioritization Portal is aGeneApart.

aGeneApart (Van Vooren *et al.*, 2007) takes as input a list of functional annotations from a variety of controlled vocabularies, including disease, dysmorphology, anatomy, development and Gene Ontology and creates as output a prioritized list of candidates.

BioGraph (Liekens *et al.*, 2011) allows prioritization of putative disease genes, supported by functional annotations from Gene Ontology.

Biomine (Eronen and Toivonen, 2012) integrates data from several publicly available biological databases on genes and phenotypes and returns a selection of candidates.

BITOLA (Hristovski *et al.*, 2005) mines MEDLINE database to discover new relations between biomedical concepts and returns a prioritized list of candidates and selection of candidates.

CANDID (Hutz *et al.*, 2008) uses several heterogeneous data sources, some of them chosen to overcome bias and returns a prioritized list of candidates.

CGI (Ma *et al.*, 2007) is a method for prioritizing genes associated with a phenotype by Combining Gene expression and protein Interaction data (CGI).

DIR (Chen *et al.*, 2011) is a tool based on an expandable framework for gene prioritization that can integrate multiple heterogeneous data sources by taking advantage of a unified graphic representation.

The DomainRBF (Zhang *et al.*, 2011) tool ensures the prioritization of genes and gene products associated with specific diseases. The user may select a disease phenotype and the prioritization is performed with Bayesian regression with respect to proteins contained in the Pfam database (Aerts *et al.*, 2009).

Endeavour (Aerts *et al.*, 2009) is a software application for the computational prioritization of candidate genes in multiple species. The process requires a training set of genes and it is based on functional annotations, protein-protein interactions, regulatory information, expression data. Endeavour can be used both as a web server and as a Java application.

G2D (Teber *et al.*, 2009) is a tool for prioritizing candidate genes for inherited diseases. The candidate genes are ranked according to their possible relation to an inherited disease by applying a combination of data mining on biomedical databases and gene sequence analysis.

GeneDistiller (Seelow *et al.*, 2008) uses information from various data sources such as gene-phenotype associations, gene expression patterns and protein-protein interactions to obtain a prioritized list of candidates.

GeneFriends (van Dam *et al.*, 2012) a seed list of genes and functional annotation for candidate gene prioritization.

Gene Prospector (Yu *et al.*, 2008) prioritizes potential disease-related genes by using literature database of genetic association studies.

GeneRank (Morrison *et al.*, 2005) combines gene expression information with a network structure derived from gene annotations and returns a prioritized list of candidates.

GeneRanker (Gonzalez *et al.*, 2008) combines gene-disease associations with protein-protein interactions extracted from the literature to obtain a ranked list of genes potentially related to a specific disease or biological process.

GeneSeeker (Van Driel *et al.*, 2005) returns the genes that are located in the specified location and expressed in the specified tissue by accessing simultaneously at different databases.

GeneWanderer (Köhler *et al.*, 2008) is an algorithm that uses a global network distance measure to define similarity in protein-protein interaction networks. A candidate disease gene prioritization is obtained.

Genie (Fontaine *et al.*, 2011) ranks genes using a text-mining approach based on gene-related scientific abstracts and return a prioritized list of candidates and a selection of candidates.

Gentrepid (Jourquin *et al.*, 2012) predicts candidate disease genes based on their association to known disease genes of a related phenotype.

GLAD4U (Jourquin *et al.*, 2012) is a prioritization tool based that applies a ranking algorithm based on the hypergeometric test.

GPSy (Britto *et al.*, 2012) is a gene prioritization system that enables the rank process of a gene list or prioritization of the entire genome for the chosen species. In GPSy, only two types of diseases are available for the analysis.

GUILD (Guney and Oliva, 2012) is a network-based framework that computes the prioritization of genes using seven different algorithms. The ranking process is based on a priori gene-disease associations and protein interactions. GUILD can be downloaded.

MedSim tool (Schlicker *et al.*, 2010) ranks a list of genes for a known disease with a valid OMIM id, involving the Gene Ontology annotations. The classification is computed using two semantic similarity measures, (Lin and SimRel) on tree ontologies, Biological Process (BP), Molecular Function (MF), Cellular Component (CC). It receives as input a list of genes and a valid OMIM id.

MetaRanker (Pers *et al.*, 2011) enables the prioritization of the genome in relation to the phenotype of interest by integrating data sources for a given risk phenotype or disease.

MimMiner (Van Driel *et al.*, 2006) gathers the human phenome by text mining and ranks phenotypes according to their similarity to a given disease phenotype.

PGMapper (Xiong *et al.*, 2008) combines gene function information from the OMIM and PubMed databases and returns the candidate gene.

PhenoPred (Radivojac *et al.*, 2008) applies a supervised algorithm for detecting gene-disease associations based on the human protein-protein interaction network, known gene-disease associations, protein sequence and returns a prioritized list of candidates.

Pinta (Nitsch *et al.*, 2011) gathers information about the phenotype from experimental data in order to identify the most promising candidates.

PolySearch (Cheng *et al.*, 2008) ranks the relationships between diseases, genes, mutations, drugs, pathways, tissues by using a text mining system.

Posmed (Yoshida *et al.*, 2009) makes use of genomewide networks to obtain a Prioritized list of candidates and a selection of candidates.

PRINCE (Vanunu *et al.*, 2010) predicts causal genes and protein complexes involved in a disease of interest by using a network-based approach.

ProDiGe (Mordelet and Vert, 2011) uses a machine learning strategy which ensures to integrate various sources of information about the genes. The machine learning strategy is based on learning from positive and unlabeled examples and enable the prioritization of Disease Genes.

ProphNet (Martínez *et al.*, 2012) integrates information from data networks to return a prioritization gene list.

S2G (Gefen *et al.*, 2010) enables to search all diseases containing similar phenotypes given a disease and genes associated with these diseases. Furthermore, S2G enables to seek other genes that are associated with selected genes through the entire human genome.

SNPs3D (Yue *et al.*, 2006) identifies genes that are candidates for being involved in a specified disease based on literature.

TargetMine (Chen *et al.*, 2011) ensures a target prioritization within the framework of a data warehouse.

ToppGene (Chen *et al.*, 2009) is a web-based software able to rank candidate genes based on a similarity score for each annotation of each candidate with respect to selected disease. ToppGene requires an initial set of training genes to compare gene candidate terms.

Then, in the Gene Prioritization Portal are listed eight tools such as, Caesar, DGP, eResponseNet, Pandas, Pocus, Prioritizer, Suspects, Tom that are yet obsolete.

Moreover, there exists a gene prioritization tool, namely GoD (Gene on Diseases) (Cannataro *et al.*, 2015) that is not listed in the Gene Prioritization Portal but is reported in Omics Tool Page (see Relevant Website section).

GoD is provided as an R package. The current version of GoD enables the prioritization of a list of input genes with respect to a selected disease. The ranking is based on ontology annotations. In specific, GoD orders genes by the semantic similarity computed with respect to a disease among the annotations of each gene and those describing the selected disease. It is based on three ontologies: HPO ontology, GO ontology and DO ontology for the calculation of ranking. It takes as input a list of genes or gene products annotated with GO Terms, HPO Terms, DO Terms and a selected disease described in terms of annotation of GO, HPO or DO. It produces as output the ranking of those genes with respect to the input disease. The package consists of three main functions: hpoGoD (for HPO based prioritization), goGoD (for GO based prioritization) and doGoD (for DO based prioritization).

The hpoGoD function enables the prioritization of a list of input genes with respect to a disease described in terms of a list of HPO Terms using semantic similarity measures. For HPO, are applied the following semantic similarity measures: Resnik, JiangConrath, Lin, simIC, relevance, GIC and Wang (Wang *et al.*, 2010). Semantic similarities are calculated by using the HPOSim package (see Relevant Website section).

The goGoD function enables the prioritization of a list of input genes with respect to a disease described in terms of a list of GO Terms using semantic similarity measures. For this function, Resnik, Lin, Rel, Jiang and Wang measures are provided. The Semantic similarities calculated by using the GOSemSim package (Yu *et al.*, 2010).

The doGoD function computes the prioritization of a list of input genes in terms of a list of DO Terms with respect to a known disease with a valid DO ID and using one semantic similarity measure among Resnik, Jiang, Lin, Schlicker, and Wang, with BMA approach calculated by using the DOSE package (Yu *et al.*, 2015).

GoD algorithm guarantees a flexibility of data, dealing with different types of annotations from different ontologies. The user can adapt the input dataset according to the study to be investigated, and can analyze the data through tree kind of annotations,

choosing one function among hpoGoD.R, goGoD.R, doGoD.R. In fact, GoD, is able to make use of multi ontologies managing all the Gene Ontology sub-ontology (BP, MF, CC) and different ontologies like HPO and DO.

Illustrative Example(s) or Case Studies

This section presents the prioritization process of selected genes with respect to a selected disease by applying one tool among those cited, GoD. GoD enables the prioritization of a list of input genes with respect to a selected disease. It is based on GO, DO and HPO ontology for the calculation of ranking. It takes as input a list of genes or gene product annotated with GO Terms, HPO Terms, DO Terms and a selected disease. It produces as output the ranking of those genes with respect to the input disease. Package consists actually of three main functions: hpoGoD (for HPO based prioritization) , doGoD (for DO based prioritization), goGoD (for GO based prioritization). For these functions, the input file is a simple text file tab-separated containing for each line the gene identifier and its related annotations. The GoD web site provide an example input file that consists in an integrated dataset of DMET (Deeken, 2009; Arbitrio 2016) genes, related GO, HPO, DO annotations. The Affymetrix DMET (drug metabolism enzymes and transporters) Plus Premier Pack is a microarray able to analyze allelic variants in 225 genes related to, genes known to be related to drug absorption, distribution, metabolism and excretion (ADME). An experiment DMET consists of the selection of a set cases and a set of controls and the comparison of their genotypes. The result is a list of probes genes whose allelic variants are different in two classes.

To perform the prioritization analysis, the user may choose a subset of probe identifiers to analyze, a known disease with a valid OMIM ID and one semantic similarity measure.

Let an experiment has selected as candidate SNPs those identified by the following probes: AM_10183, AM_10178, AM_10174, AM_10168, the studied pathology is DUBIN-JOHNSON SYNDROME and the user want rank these genes by exploiting the HPO ontology. Thus, the user gives as input the list of these probes and the disease using the OMIM identifiers (see Relevant Website section) and select a semantic similarity from those provided by GoD. Since genes are annotated with HPO annotations, GoD calculates the semantic similarity among each of the input genes and the selected disease. Finally, GoD presents as output a histogram showing the semantic similarity values sorted from the higher to the lower in order to enhance the readability of results. A gradient of colors is also used to improve the readability.

In another example, an experiment has selected the following candidate identifiers set: AM_10163, AM_10155, AM_10143, AM_10121 and the user want to compute the prioritization of these input probes with respect to the breast cancer that is identified by D01216 in DO. In this case the user exploits the doGoD ontology to rank these genes. Similarly to the previous example, the user provides as input the list of identifiers, the disease identifier in DO and the selected semantic similarity. GoD calculates the semantic similarity among each of the input genes annotated with DO annotations and the selected disease.

Furthermore, It is also possible to compute the prioritization of the four probes identifiers with respect to Gene Ontology.

Results and Discussion

The results of omics experiments consist of a large list of genes, or gene products. However only to the behavior of few molecules is primary interest. Consequently, the need for the introduction of software tools able to help researchers to filter these lists on the basis of biological or clinical considerations e.g. which molecules are more related to a given disease arises. Identifying the most promising candidates among such large lists of genes is a challenge. In the past, the biologists manually obtained a candidates list by checking what is currently known about each gene, and assess whether it is a promising candidate. However, the bioinformatics community has introduced the concept of gene prioritization to take advantage of both the progress made in computational biology and the large amount of genomic data publicly available. Gene prioritization consists of the finding the most promising genes among large lists of candidate genes. Consequently, several gene prioritization tools have been developed to tackle this task, and some of them have been implemented and made available through freely available web tools. The informations about these tools are summarized in a website termed Gene Prioritization Portal that describes in detail the prioritization tools and it has been designed to help researchers to carefully select the tools that best correspond to their needs. These tools ensure the prioritization of human gene and differ by the inputs they need, the computational methods they implement, the data sources related to biological knowledge they use and the output they present to the user. Different tools such as MedSim, GUILD, ToppGene, GPSy, DomainRBF and GoD uses the knowledge contained in public ontologies to guide the selection process. In the case study, GoD is used to obtain the prioritization of DMET genes based on the use of annotation and semantic similarity measures. The advantages of using GoD for the prioritization process covers many aspects. GoD guarantees a flexibility of data, both dealing with different types of annotations from different ontologies and making use of multi ontologies managing all the Gene Ontology sub-ontology (BP, MF, CC). The use of annotations allows to evaluate any gene set and any disease without a training data set. Furthermore GoD uses the semantic similarity measures in the prioritization process of genes with respect to a selected disease. In this way, GoD assigns a value of similarity to ranked genes with respect to a known disease based on GO, HPO or DO annotations. GoD is also the only tool that provides a broad range of measures of similarity that the user can select. Finally, differently from the other tools, GoD is available for download as an R-package and its approach is customisable; thus the user may easily improve the analysis capabilities by inserting novel ontologies.

Future Directions

The progress of high-throughput technologies as well as the growth of genomic data direct the future directions of research towards the improvement of prioritization tools. In particular, the improvements have been conducted on the interface, which is sometimes overlooked in the software development process. Furthermore, these tools computed the prioritization process only among human genes; a future work regards the application of data available for species close to human.

See also: Biological and Medical Ontologies: Systems Biology Ontology (SBO). Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Gene Prioritization Using Semantic Similarity. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Ontology-Based Annotation Methods. Semantic Similarity Definition. Tools for Semantic Analysis Based on Semantic Similarity

References

- Aerts, S., Vilain, S., Hu, S., *et al.*, 2009. Integrating computational biology and forward genetics in *Drosophila*. *PLOS Genet* 5 (1), e1000351.
- Arbitrio, M., Di Martino, M.T., Scionti, F., *et al.*, 2016. DMET™ (Drug Metabolism Enzymes and Transporters): A pharmacogenomic platform for precision medicine. *Oncotarget* 7 (33), 54028–54050.
- Börnigen, D., Tranchevent, L.C., Bonachela-Capdevila, F., *et al.*, 2012. An unbiased evaluation of gene prioritization tools. *Bioinformatics* 28 (23), 3081–3088.
- Britto, R., Sallou, O., Collin, O., *et al.*, 2012. GPSy: A cross-species gene prioritization system for conserved biological processes – application in male gamete development. *Nucleic Acids Research* 40 (W1), W458–W465.
- Cannataro, M., Guzzi, P.H., Milano, M., 2015. God: An R-package based on ontologies for prioritization of genes with respect to diseases. *Journal of Computational Science* 9, 7–13.
- Cannataro, M., Guzzi, P.H., Sarica, A., 2013. Data mining and life sciences applications on the grid. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3 (3), 216–238.
- Cannataro, M., Guzzi, P.H., Veltri, P., 2010. Protein-to-protein interactions: Technologies, databases, and algorithms. *ACM Computing Surveys ((CSUR))* 43 (1), 1.
- Chen, J., Bardes, E.E., Aronow, B.J., Jegga, A.G., 2009. ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Research* 37 (suppl 2), W305–W311.
- Chen, Y., Wang, W., Zhou, Y., *et al.*, 2011. In silico gene prioritization by integrating multiple data sources. *PIOS One* 6 (6), e21137.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2011. TargetMine, an integrated data warehouse for candidate gene prioritisation and target discovery. *PIOS One* 6 (3), e17844.
- Cheng, D., Knox, C., Young, N., *et al.*, 2008. PolySearch: A web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Research* 36 (suppl 2), W399–W405.
- Cho, Y.R., Mina, M., *et al.*, 2013. M-finder: Uncovering functionally associated proteins from interactome data integrated with go annotations. *Proteome Science* 11 (1), S3.
- Deeken, J., 2009. The Affymetrix DMET platform and pharmacogenetics in drug development. *Current Opinion in Molecular Therapeutics* 11 (3), 260–268.
- du Plessis, L., Škunca, N., Dessimoz, C., 2011. The what, where, how and why of gene ontology – a primer for bioinformaticians. *Briefings in Bioinformatics*. bbr002.
- Eronen, L., Toivonen, H., 2012. Biomine: Predicting links between biological entities using network models of heterogeneous databases. *BMC Bioinformatics* 13 (1), 119.
- Fontaine, J.F., Priller, F., Barbosa-Silva, A., *et al.*, 2011. Genie: Literature-based gene prioritization at multi genomic scale. [A.] *Nucleic Acids Research* 39 (suppl 2), W455–W461.
- Gefen, A., Cohen, R., Birk, O.S., 2010. Syndrome to gene (S2G): In-silico identification of candidate genes for human diseases. *Human Mutation* 31 (3), 229–236.
- Gene Ontology Consortium, 2004. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research* 32 (suppl 1), D258–D261.
- Gonzalez, G., Uribe, J.C., Armstrong, B., *et al.*, 2008. GeneRanker: An online system for predicting gene-disease associations for translational research. *Summit on Translat Bioinforma* 2008, 26–30.
- Guney, E., Oliva, B., 2012. Exploiting protein-protein interaction networks for genome-wide disease-gene prioritization. *PIOS One* 7 (9), e43557.
- Guzzi, P.H., Mina, M., Guerra, C., Cannataro, M., 2012. Semantic similarity analysis of protein data: Assessment with biological features and issues. *Briefings in Bioinformatics* 13 (5), 569–585.
- Hristovski, D., Peterlin, B., Mitchell, J.A., *et al.*, 2005. Using literature-based discovery to identify disease candidate genes. *International Journal of Medical Informatics* 74 (2), 289–298.
- Hutz, J.E., Kraja, A.T., *et al.*, 2008. CANDID: A flexible method for prioritizing candidate genes for complex human traits. *Genetic Epidemiology* 32 (8), 779.
- Jourquin, J., Duncan, D., Shi, Z., Zhang, B., 2012. GLAD4U: Deriving and prioritizing gene lists from PubMed literature. *BMC Genomics* 13 (8), S20.
- Köhler, S., Bauer, S., Horn, D., Robinson, P.N., 2008. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics* 82 (4), 949–958.
- Liekens, A.M., De Knijf, J., Daelemans, W., *et al.*, 2011. BioGraph: Unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biology* 12 (6), R57.
- Lin, D., 1998. An information-theoretic definition of similarity. In *ICML* 98 (1998), 296–304.
- Ma, X., Lee, H., Wang, L., Sun, F., 2007. CGI: A new approach for prioritizing genes by combining gene expression and protein–protein interaction data. *Bioinformatics* 23 (2), 215–221.
- Martínez, V., Cano, C., Blanco, A., 2012. Network-based gene-disease prioritization using PROPHNET. *EMBNet. Journal* 18 (B), 38.
- Milano, M., Agapito, G., Guzzi, P.H., Cannataro, M., 2014. Biases in information content measurement of gene ontology terms. In: 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 9–16. IEEE.
- Mordelet, F., Vert, J.P., 2011. ProDiGe: Prioritization of Disease Genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics* 12 (1), 389.
- Moreau, Y., Tranchevent, L.C., 2012. Computational tools for prioritizing candidate genes: Boosting disease gene discovery. *Nature Reviews Genetics* 13 (8), 523–536.
- Morrison, J.L., Breitling, R., Higham, D.J., *et al.*, 2005. GeneRank: Using search engine technology for the analysis of microarray experiments. *BMC Bioinformatics* 6 (1), 233.
- Nitsch, D., Tranchevent, L.C., Goncalves, J.P., *et al.*, 2011. PINTA: A web server for network-based gene prioritization from expression data. *Nucleic Acids Research* 39 (suppl 2), W334–W338.
- Pers, T.H., Hansen, N.T., Lage, K., *et al.*, 2011. Meta-analysis of heterogeneous data sources for genome-scale identification of risk genes in complex phenotypes. *Genetic Epidemiology* 35 (5), 318–332.
- Pesquita, C., Faria, D., Bastos, H., *et al.*, 2008. Metrics for GO based protein semantic similarity: A systematic evaluation. *BMC Bioinformatics* 9 (5), S4.
- Pesquita, C., Faria, D., Falcao, A.O., Lord, P., Couto, F.M., 2009. Semantic similarity in biomedical ontologies. *PLOS Comput Biol* 5 (7), e1000443.

- Radivojac, P., Peng, K., Clark, W.T., *et al.*, 2008. An integrated approach to inferring gene–disease associations in humans. *Proteins: Structure, Function, and Bioinformatics* 72 (3), 1030–1037.
- Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.
- Robinson, P.N., Köhler, S., Bauer, S., *et al.*, 2008. The Human Phenotype Ontology: A tool for annotating and analyzing human hereditary disease. *The American Journal of Human Genetics* 83 (5), 610–615.
- Schlicker, A., Lengauer, T., Albrecht, M., 2010. Improving disease gene prioritization using the semantic similarity of Gene Ontology terms. *Bioinformatics* 26 (18), i561–i567.
- Schriml, L.M., Arze, C., Nadendla, S., *et al.*, 2012. Disease Ontology: A backbone for disease semantic integration. *Nucleic Acids Research* 40 (D1), D940–D946.
- Seelow, D., Schwarz, J.M., Schuelke, M., 2008. GeneDistiller – distilling candidate genes from linkage intervals. *PLOS One* 3 (12), e3874.
- Teber, E.T., Liu, J.Y., Ballouz, S., *et al.*, 2009. Comparison of automated candidate gene prediction systems using genes implicated in type 2 diabetes by genome-wide association studies. *BMC Bioinformatics* 10 (1), S69.
- van Dam, S., Cordeiro, R., Craig, T., *et al.*, 2012. GeneFriends: An online co-expression analysis tool to identify novel gene targets for aging and complex diseases. *BMC Genomics* 13 (1), 535.
- Van Driel, M.A., Bruggeman, J., Vriend, G., *et al.*, 2006. A text-mining analysis of the human phenome. *European Journal of Human Genetics* 14 (5), 535–542.
- Van Driel, M.A., Cuelenaere, K., Kemmeren, P.P.C.W., *et al.*, 2005. GeneSeeker: Extraction and integration of human disease-related information from web-based genetic databases. *Nucleic Acids Research* 33 (suppl 2), W758–W761.
- Vanunu, O., Magger, O., Ruppin, E., *et al.*, 2010. Associating genes and protein complexes with disease via network propagation. *PLoS Comput Biol* 6 (1), e1000641.
- Van Vooren, S., Thienpont, B., Menten, B., *et al.*, 2007. Mapping biomedical concepts onto the human genome by mining literature on chromosomal aberrations. *Nucleic Acids Research* 35 (8), 2533–2543.
- Wang, H., Zheng, H., Azuaje, F., 2010. Ontology-and graph-based similarity assessment in biological networks. *Bioinformatics* 26 (20), 2643–2644.
- Xiong, Q., Qiu, Y., Gu, W., 2008. PGMapper: A web-based tool linking phenotype to genes. *Bioinformatics* 24 (7), 1011–1013.
- Yoshida, Y., Makita, Y., Heida, N., *et al.*, 2009. PosMed (Positional Medline): Prioritizing genes with an artificial neural network comprising medical documents to accelerate positional cloning. *Nucleic Acids Research* 37 (suppl 2), W147–W152.
- Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., Wang, S., 2010. GOSemSim: An R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* 26 (7), 976–978.
- Yu, G., Wang, L.G., Yan, G.R., He, Q.Y., 2015. DOSE: An R/Bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics* 31 (4), 608–609.
- Yu, W., Wulf, A., Liu, T., *et al.*, 2008. Gene Prospector: An evidence gateway for evaluating potential susceptibility genes and interacting risk factors for human diseases. *BMC Bioinformatics* 9 (1), 528.
- Yue, P., Melamud, E., Moulit, J., 2006. SNPs3D: Candidate gene and SNP selection for association studies. *BMC Bioinformatics* 7 (1), 166.
- Zhang, W., Chen, Y., Sun, F., *et al.*, 2011. DomainRBF: A Bayesian regression approach to the prioritization of candidate domains for complex diseases. *BMC Systems Biology* 5 (1), 55.

Relevant Websites

<http://disease-ontology.org>
DISEASE ONTOLOGY.

<http://www.esat.kuleuven.be/gpp>
ESAT.

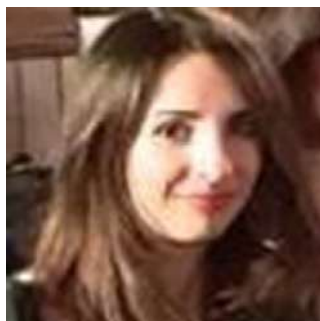
<https://omictools.com/gene-ranking-based-on-diseases-tool>
HPOSim package.

<http://www.human-phenotype-ontology.org>
Human Phenotype Ontology.

<https://omictools.com/gene-ranking-based-on-diseases-tool>
OMIC TOOLS.

<http://www.omim.org/>
OMIM.

Biographical Sketch



Marianna Milano received the Laurea degree in biomedical engineering from the University Magna Græcia of Catanzaro, Italy, in 2011. She is a Ph.D student at the University Magna Græcia of Catanzaro. Her main research interests are on biological data analysis and semantic-based analysis of biological data. She is a member of IEEE Computer Society.

Networks in Biology

Valeria Fionda, University of Calabria, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Research in bioinformatics was initially focused on the study of individual molecules and cellular components. Indeed, computers became essential to compare into an automated way multiple protein sequences, a task that was impractical to be made manually. Even if the reductionist method (Van Regenmortel, 2004) of studying the individual components (such as proteins, DNA, RNA and small molecules) and their functions resulted in explaining the chemical basis of numerous processes within living organisms, during the years the limitations of such approach have arisen. Indeed, biological systems are extremely complex and some behaviors cannot be explained by studying their individual parts only, since they frequently function in different ways in different processes. Instead, new biological characteristics, which are absent in the isolated components, arise as network behavior, that is from the way in which these components interact with one another. Such emergent properties of biological systems cannot be understood or predicted by simply analyzing their components. This is why bioinformaticians has increasingly shifted their interest from individuals to large-scale networks of interactions. In this respect, a key challenge is to understand and analyze the structure and the dynamics of such networks of interactions that contribute to the living functions of cells.

The study of biological networks has been facilitated by two factors. On the one hand, high-throughput data-collection techniques allowed to identify active molecules in biomolecular pathways. On the other hand, some technology platforms, such as yeast two-hybrid screening, helped to discover how and when these molecules interact with each other. As a result, by combining the discovered interactions, several types of biological networks has been generated, including protein-protein interaction, metabolic, signaling and transcription regulatory networks. It is important to note that even if such networks are often studied independently from one another, they actually form a 'network of networks'. In fact, they altogether determine the behavior of the cell.

It is interesting to point out that by applying the theory of complex networks to biological networks, it has been detected that the organizing principles that govern the formation and evolution of molecular interaction networks within a cell are similar to those of other complex systems, such as the Internet and social networks. This suggested that the laws that govern most complex networks in nature are similar and, thus, the knowledge from non-biological systems can help in characterizing biological networks. Indeed, similarly to other non-biological networks, networks of biological interactions are often represented as graphs, the nodes of which represents individual molecules and edges represent the interactions between them.

This article introduces the different types of biological networks. The rest of the article is organized as follows. In Section Networks in Biology the various type of biological networks are introduced and then discussed in depth in the subsequent sections, in particular: gene regulatory network are discussed in Section Gene Regulatory Networks, signaling network in Section Signaling Networks, metabolic network in Section Metabolic Networks, protein-protein interaction networks in Section Protein-Protein Interaction Networks and gene co-expression networks in Section Gene Co-Expression Networks. In Section Applications some applications of biological networks are discussed. Finally, in Section Closing Remarks some conclusions are drawn.

Networks in Biology

Molecules inside the cell interact with each other determining the biological processes at the basis of the cell's life. A set of molecules that are connected by physical interactions form a molecular interaction networks. Since molecular interactions can occur between molecules belonging to different biochemical families (e.g., proteins, genes, metabolites) and also within a given family, interaction networks are usually classified by the nature of the molecules involved. As an example, a well studied interactome is the protein-DNA interactome, that results in gene-regulatory networks. Such networks are formed by transcription factors, chromatin regulatory proteins, and their target genes. Gene-regulatory networks are discussed in Section Gene Regulatory Networks.

A key issue in biology is cell signaling, that is the response of a cell to internal and external stimuli and coordinate the regulation of its activity. Signaling networks are made up of highly connected modules that regulate some functions in the cell. Signaling networks are described in Section Signaling Networks.

A slightly different type of biological network are metabolic networks. Such networks are composed by cellular constituents and reactions that convert metabolites, i.e., chemical compounds in a cell, into each other. Such process is regulated by enzymes, which have to bind their substrates physically. Metabolic networks are presented in Section Metabolic Networks.

Another example of biological networks is the interactome of an organisms that refers to its set of protein-protein interactions. Such interactions generates protein-protein interaction networks. In many application it makes sense to study a subset of PPI interactions that generate a subnetwork. Protein-protein interaction networks are illustrated in Section Protein-Protein Interaction Networks.

Another type of biological network is obtained by using gene expression data. Gene expression is the process by which the gene products are synthesised by using the information encoded into genes. In particular, co-expression analysis allows to study existing patterns in gene expression data and to identify groups of genes with correlated expression profiles. Gene co-expression networks are discussed in Section Gene Co-Expression Networks.

Gene Regulatory Networks

The transcription of a gene to an mRNA molecule is regulated by proteins referred to as transcription factors. Transcription factors are proteins or protein complexes that contain at least one DNA-binding domain (DBD), which can bind to a specific sequence of DNA referring to the genes that they regulate. In particular, by binding to such DNA regions they activate (or upregulate) or inhibit (or downregulate) the production of another protein. In more details, transcription factors activate (resp., inhibit) the expression of a gene inside the cell by binding to regions upstream (resp., downstream) of the gene on the DNA molecule. This process may, in turn, facilitate or prevent RNA polymerase from binding and initiating the transcription of the gene. Thus, the genes inside cells interact with each other via intermediate transcription factors to influence each others expression.

In unicellular organisms, gene regulatory networks are essential to regulate the cell activity to survive to external environmental conditions. In multicellular organisms, such networks serve the purpose of regulating cell activity depending on the particular function the cell has to perform. This is essential when a cell divides and the two resulting cells, even containing the same genome, can differ in functions depending on which genes are activated or not. Moreover, some proteins resulting from the activation due to some transcription factors can pass through cell membranes and create signaling paths to other cells.

The regulation process can involve more than one transcription factor and a gene product can indirectly downregulate or upregulate another gene product through a chain of regulations. Moreover, cycles of regulation can generate feedback loops through which a gene product can downregulate or upregulate its own production. Negative feedback loops result in down-regulation, indicating for example that the transcript levels are kept constant or proportional to a factor. Positive feedback loops result in upregulation.

The set of genes interactions, both activations and inhibitions, inside the cell is referred to as the gene regulatory network.

Modeling

Various models have been proposed for gene regulatory networks. These models can be divided into three classes. The first class is that of *logical models*, which describe gene regulatory networks from the qualitative point of view. They can be adapted to fit biological phenomena but can only answer qualitative questions such as to understand if a gene is activated or not at some stage of the life of the system. For a finer grain analysis, when also quantitative information are needed (e.g., the concentrations of molecules) a second class of models, that is *continuous models*, has been developed. A third class of models, that is *single molecule level models*, takes into account the noise that can affect the functionality of regulatory networks. Some review works on the different types of modeling techniques can be found in the literature (Alakwaa, 2014; Karlebach and Shamir, 2008).

The basic logical model to represent gene regulatory networks is a Boolean network (Glass and Kauffman, 1973; Kauffman, 1993). A Boolean network is a directed graph the nodes of which are in one-to-one correspondence with genes and have associated Boolean variables. A value equals to 0 is associated to nodes whose corresponding gene is not expressed; a value equals to 1 is associated to nodes whose corresponding gene is expressed. In particular, each node (i.e., gene) has associated a Boolean function, that determine its Boolean value, that depends only from the Boolean values of its parent nodes. The state of the Boolean network is the set of values of the nodes and can change during time to model the evolution of the biological system. However, very often, due to insufficient understanding of the biological system to model, for each node in the network there is more than one possible boolean function that generate its state. To overcome the limitation imposed by Boolean networks, that allow to use one function only, another model, called Probabilistic Boolean networks (Shmulevich et al., 2002, 2003), has been proposed. Probabilistic Boolean networks modify Boolean networks by associating to each node several regulation Boolean functions each of which has associated a probability. Thus, at each time step the function to apply at each node is determined randomly according to the given probabilities.

Another logical model used for gene regulatory networks are Bayesian networks (Chen et al., 2006; Friedman et al., 2000), that are probabilistic directed acyclic graph models that represent a set of random variables (i.e., the genes' expressions) and their conditional dependencies (i.e., activation or inhibition) via a directed acyclic graph.

To analyze the dynamics of gene regulatory network a good model to use are Petri nets (Steggles et al., 2007). A Petri net is a directed bipartite graph having two types of nodes, that are transitions, represented by bars, and places, represented by circles. The directed arcs, denoted by arrows, describe which places are pre- and/or post-conditions for which transitions. Arcs run from a place to a transition or vice versa. Moreover, tokens, represented by black dots inside some places, enable transitions to fire.

A shortcoming of logical models is that they require the discretization of the input data. However, experiments that measure gene-expression intensities produce real values. This motivated the adoption of continuous models that use real-valued parameters. A basic type of continuous model are linear models (Weaver et al., 1999), that are based on the assumption that each regulator contributes to the regulation of the same gene independently from the other regulators. This translates into regulation functions that are weighted linear sums of the levels of the regulators.

A more general model is based on differential equations instead of linear functions (Chen *et al.*, 2004; Li *et al.*, 2008). Differential equations allow to explicitly model the concentration changes of molecules over time. Thus, it is a better model to study the dynamics of the network since it allows to follow concentration changes continuously, even when the time increase considered is infinitely small.

To conclude, it has been observed that the behavior of biological systems is not deterministic, in the sense that it can evolve differently even starting from the same initial state. Such source of non-determinisms can be modeled by stochastic approaches (Gillespie, 2007). Such approaches attempt to describe the time evolution of the system by taking into account the systems stochasticity by using probability functions.

Signaling Networks

Cells have the ability to receive and process signals that originate outside their borders in order to respond to changes in their immediate environment. In particular, inside the cell there are particular proteins, called receptors, that bind to signaling molecules and initiate the response process. Receptors are generally transmembrane proteins that are able to bind signaling molecules that are outside the cell and then transmit the stimuli inside the cell. There are different types of receptors that can bind different signaling molecules. Moreover, different cell types have different populations of receptors.

Individual cells often receive many signals simultaneously and integrate the different stimuli to define a unified action plan. Most cell signals correspond to chemical characteristics such as growth factors, hormones or neurotransmitters. However, some cells are also able to respond to mechanical stimuli, such as sensory cells in the skin that respond to the pressure of touch. The response reaction of a cell can be for example the activation of a gene or the production of energy. The capacity of the cell to receive and correctly respond to the environmental stimuli is essential for cell development. Indeed, malfunctioning in stimuli processing is responsible for diseases such as cancer and diabetes.

Cells are usually specialized to perform some specific functions and, thus, to carry out complex biological processes several cells need to cooperate and coordinate their activities. Hence, cells have to communicate with each other to respond in an appropriate manner to specific environmental stimuli. Such communication is made possible by a process called cell signaling. The overall process that transform the external signals into changes that occur inside the cell is called signal transduction. Signal conversions usually involve sequences of chemical reactions among proteins. The signal transduction networks or signaling networks store information about the processes through which cells respond to stimuli. Signaling can be subdivided in the following five classes:

- Intracrine: refers to signals that are produced inside a cell and remain inside the cell;
- Autocrine: refers to signals produced inside a cell, secreted to the external environment, and that affect the same cell (or close-by cells of the same type) via receptors;
- Juxtacrine: refers to signals produced by a cell that affect adjacent cells via cell membrane contact;
- Paracrine: refers to signals produced by a cell that affect nearby cells (does not require cell contact);
- Endocrine: refers to signals produced by a cell that can reach cells in other parts of the body via the circulatory system.

Another way to classify signaling is with respect to the signaling molecule. Three major classes of signaling molecules can be identified:

- Hormones: they are the major signaling molecules of the endocrine system and, often, they regulate each other's secretion. They are used to communicate for physiological regulation and behavioral activities such as metabolism or sleep.
- Neurotransmitters: they are signaling molecules of the nervous system.
- Cytokines: they are signaling molecules of the immune system.

Modeling

Signaling networks can be represented at a qualitative level by a graph the nodes of which are molecules and the edges of which represent the capability of molecules to activate or deactivate other molecules. The qualitative approach allows to perform statistical analysis of signaling networks and to identify some structural properties about single elements of the network, clusters of elements, or the entire network, such as node degree (the number of edges connected to a node) or the shortest path between two nodes.

The qualitative approach, however, does not allow to take into account the temporal evolution of signaling networks occurring under different conditions. Quantitative models extend qualitative models in this direction. Systems of Ordinary Differential Equations (ODE) have been largely used for the quantitative modeling of signal transduction networks. They represent a biological system by a set of mathematical equations, each of which rules the temporal evolution of a molecule. Even if they are widely used, ODE models have some drawbacks among which the fact that defining the equations is a very difficult task. To overcome the drawbacks of ODE models, several network-oriented and bio-inspired models have been introduced that allow to symbolically describe signaling processes. Among them, there are P systems (Paun, 2000), or membrane systems, a computational model which takes its inspiration from the structure and functioning of the cell, and in particular from the way in which chemicals interact and cross cell membranes. The P system has a particular membrane structure that subdivides it compartments. The compartments

contain objects (described by symbols or by strings of symbols) which evolve according to some given rules. By applying the rules in a nondeterministic, parallel manner, transitions between the system configurations are determined.

Metabolic Networks

The growth and maintenance of the cell is performed via biochemical reactions responsible for the up-taking of nutrients from the external and their conversion into other molecules. Each reaction, that is a transformation of chemical substances or metabolites (reactants) into other substances (products), is usually catalyzed by enzymes. In general metabolic reactions are reversible, that is, they occur in both directions; but, if considered in isolation, each reaction reaches a steady state that is a state where the amount of change in both directions is equal. However, the cell continuously exchange substances with the environment and other factors can affect the reactions. Thus, the steady states can be shifted and it can be recognized a main direction for a reaction (this is why usually one distinguish between reactants and products). Moreover, metabolic reactions interact with each other, that is the product of one reaction can be a reactant of another reaction. A sequence of metabolic reactions $P = (R_1, \dots, R_n)$ such that at least one product of the reaction R_i is a reactant for the reaction R_{i+1} , for all $i \in \{1, \dots, n-1\}$, is called a path of reactions.

The metabolic network of a cell or an organism is its complete set of metabolic reactions. The complete metabolic networks of several organisms, from bacteria to human, can be reconstructed thanks to the sequencing of complete genomes. Indeed, several of these networks are available online and can be freely downloaded from several databases such as Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000) or EcoCyc (Keseler et al., 2005). A metabolic pathway is a connected sub-network of a metabolic network that usually represents a specific process. Metabolic pathways can be also constructed by defining their functional boundaries, for example, by defining an initial and final metabolite.

The substances that participate to a metabolic pathway are often divided into two types: the main substances and the co-substances, that are small metabolites, such as ATP or ADP. However, such subdivision is not a global property but depends on the reaction under consideration, meaning that the co-substances for a reaction can be considered as main substances for another reaction. Such a distinction is mainly used by visualization tools that generally visualize main and co-substances in a different manner.

During the years, many simplifications of metabolic networks have been used in the literature such as:

- Simplified metabolic networks (Bachmaier et al., 2014) that are networks that contains reactions, enzymes and main substances, but no co-substances.
- Metabolite networks and simplified metabolite networks (Risler et al., 2006) that are networks consisting only of substances (metabolites) and, in the simplified case, only of main substances.
- Enzyme networks (Xiong et al., 2016), that are networks consisting only of the enzymes catalyzing the reactions.

Modeling

The simplest model to represent a metabolic network is via an hyper-graph. The nodes of the hyper-graph represent the substances and the hyper-edges represent the reactions. A hyper-edge involves all substances of a reaction, is directed from reactants to products and can be labeled with the enzymes that catalyze the reaction. Hyper-graphs can be represented by directed bipartite graphs. A directed bipartite graph, indicated as $G = (V_S, V_R, E)$, is a directed graph having two partitions of nodes, V_S and V_R . Here, nodes in V_S represent substances, nodes in V_R represent reactions (and are labeled with the enzymes that catalyze the reaction) and directed edges in $E \subseteq (V_S \times V_R) \cup (V_R \times V_S)$ represent the transformation of substances. In particular, the set of edges $(u_1, r), \dots, (u_n, r), (r, v_1), \dots, (r, v_m)$ encodes the fact that the reaction r transforms the reactants $u_1, \dots, u_n$ into the set of products $v_1, \dots, v_m$ (and encode the hyper-edge involving $u_1, \dots, u_n, v_1, \dots, v_m$ representing r). In the bipartite metabolic graph, there are no direct links between either two metabolites or two reactions.

Another bipartite graph representation of metabolic networks $G = (V_S, V_E, E)$ considers as the two partitions of nodes the chemical compounds (V_S) and the enzymes (V_E), respectively. For each enzyme node, an incoming edge occurs with each of its substrate nodes and an outgoing edge occurs with each of its product nodes, i.e., $E \subseteq (V_S \times V_E) \cup (V_E \times V_S)$.

Metabolic networks can be also modeled via weighted tripartite graphs $G = (V_S, V_R, V_E, E_R, E_E)$, that have three types of nodes representing metabolites (V_S), reactions (V_R) and enzymes (V_E), respectively, and two types of edges representing mass flow (E_R) and catalytic regulation (E_E), respectively. The first type of edge connects reactants to reactions and reactions to products (i.e., $E_R \subseteq (V_S \times V_R) \cup (V_R \times V_S)$). The second type connects enzymes to the reactions they catalyze (i.e., $E_E \subseteq (V_E \times V_R)$).

Another possible simpler model is based on the use of graphs (which could be directed or undirected) having one type of node only. For instance, if one considers nodes as representing enzymes, directed edges connecting pairs of enzymes represent the fact that the product of the first enzyme is a reactant of the second enzyme. If one considers nodes as representing metabolites the directed edges represent enzymes that catalyze a reaction having the first metabolite as a reactant and the second metabolite as a product.

More complex models involve the use of Ordinary Differential Equations (ODE) or P systems and, in particular, of Metabolic P systems, shortly MP systems (Manca and Luca, 2008), that are particularization of P systems applied to metabolic networks.

Protein-Protein Interaction Networks

Proteins are the main agents of biological function. Indeed, proteins control molecular and cellular mechanisms and, thus, determine healthy and diseased states of organisms. However, they are not functional in isolated forms but they interact with each other and with other molecules (e.g., DNA and RNA) to perform their functions. Thus, the study of proteins' interactions is crucial to understand their role inside the cell. Since this type of interactions can be of several types, the term *protein-protein interaction* refers to a variety of events happening inside the cell. As an example, a protein-protein interaction can indicate the formation of either stable or transient protein complexes as well as either physical or functional interactions.

A protein-protein interaction network stores the information about the protein-protein interactome of a given organism, that is the whole set of its protein-protein interactions. Even if it has been suggested that the size of protein-protein interactomes proportionally grows with the biological complexity of the organisms, none protein-protein interactome has been completely identified and, thus, this correlation only remains a conjecture. Moreover, the available protein-protein interaction networks are error-prone due to the fact that experimental methods used to discover interactions may include false positives or there may unveil some existing interactions.

There are a multitude of methods that have been proposed to detect protein-protein interactions and populate protein-protein interaction networks. In general, none of them is better than the others but each method has its own strengths and weaknesses. The goodness of a method is measured in terms of its sensitivity (high sensitivity corresponds to the ability of the method to discover many real interactions) and specificity (high specificity indicates that most of the interactions detected are real interactions). Thus, to each protein-protein interaction can be associated a weight that takes into account the sensitivity and specificity of the method used to discover it. The methods commonly used to detect protein-protein interactions can be classified in two main groups: experimental and computational methods. The experimental methods can be divided in their turn in two classes: (i) biophysically methods that represent the main source of knowledge about protein-protein interactions and are based on structural information (e.g., X-ray crystallography, NMR spectroscopy, fluorescence, atomic force microscopy); and (ii) high-throughput methods, that can be either direct or indirect. Among the direct high-throughput methods there is the Yeast two-hybrid (Y2H) that examines the interaction of two given proteins by fusing each of them to a transcription binding domain. Indirect high-throughput methods are based on the use of other types of experimental data such as gene co-expression data. Experimental methods, even if being accurate, have some severe drawbacks: they are expensive and time consuming. This motivated the adoption of computational methods that are fast and cheap. Computational methods can be subdivided in two classes: (i) empirical predictions that use experimental data to infer new protein-protein interactions; and (ii) theoretical predictions that use some accepted assumptions to predict protein-protein interactions. The main drawback of empirical prediction is that by exploiting experimental data they naturally propagate errors and inaccuracies.

Although protein-protein interaction networks are incomplete and error-prone, they are particularly important in many contexts. For example, the analysis of such networks facilitates the understanding of the mechanisms that trigger the beginning and progression of diseases. Moreover, protein-protein interaction networks have been used to discover novel protein function ([Fionda et al., 2009](#)) and to identify functional modules and conserved interaction patterns ([Fionda and Palopoli, 2011](#)).

Some studies on the structure of the protein-protein interaction networks of several species allowed to discover that independently from the species, protein-protein interaction networks are scale-free. It means that some hub proteins have a central role participating in the majority of the interactions while most proteins, that are not hubs, only participate to a small fraction of interactions.

In the literature there exists another type of biological network that is strictly related to protein-protein interaction networks: domain-domain interaction (DDI) networks. Domains are independently folded modules of a protein and a domain-domain interaction (DDI) network is constructed by replacing each protein in a protein-protein interaction network by one or more nodes representing its constituent domains. In this type of network, each edge that in the protein-protein interaction network connected two proteins is transformed in an edge connecting the corresponding domain nodes. Since most of the known proteins are composed by more than one domain, a domain-domain interaction network usually is much larger than the protein-protein interaction network from which is extracted.

Modeling

Protein-protein interaction networks are commonly modeled via graphs, whose nodes represent proteins and whose edges, that are undirected and possibly weighted, connect pairs of interacting proteins. Edge weights may be used to incorporate reliability information associated to the corresponding interactions.

It is important to point out that, since protein-protein interactions are often obtained from protein complex detection and not really as binary interactions, a more complex model may be more suitable for the representation of protein-protein interaction networks. In fact, the use of hyper-graphs, instead of simple graphs, allows to model protein complexes, where each hyper-edge involves all proteins belonging to the same complex.

Gene Co-Expression Networks

Gene expression is the process by which the information encoded in a gene is used to synthesize a gene product (i.e., a protein or RNA). Transcription is the first step of gene expression, in which the DNA of the gene is copied into RNA. Gene co-expression

networks store information about transcription that takes place at the same time or under the same conditions. Differently from gene regulatory networks (see Section Gene Regulatory Networks), gene co-expression networks do not provide any information about the causality relationships between genes (e.g., activation or inhibition) and edges only represent the fact that there exists a correlation or dependency relationship among genes.

Several methods have been developed for constructing gene co-expression networks and they are all composed by two main steps: (1) calculating a co-expression measure, and (2) selecting a significance threshold. The first step is usually carried out by exploiting high-throughput gene expression profiling technologies such as large-scale DNA microarray experiments. In particular, starting from microarray gene expression data for several samples or experimental conditions, a gene co-expression measure can be obtained by looking for pairs of genes that show a similar expression pattern across samples. The input data is often represented as a matrix, whose columns represent genes and whose rows represent the different samples or conditions. Such a matrix is called expression matrix. A numerical value to quantify the similarity of the expression profiles of genes can be obtained according to several measures such as the Pearson's correlation coefficient, Mutual Information, Spearman's rank correlation coefficient or Euclidean distance. In particular, the correlation of two genes is computed by comparing different rows of the expression matrix and computing a numerical similarity value for the two genes according to the selected measure. If all pairs of genes are considered a correlation matrix (whose rows and columns both represent genes) can be built, where each element shows how similar the expression levels of two genes are. As for the second step, a threshold is selected such that gene pairs which have a correlation score higher than the threshold are considered to have significant co-expression relationship. Then, the elements in the correlation matrix which are above then threshold are replaced by 1 (meaning that the corresponding genes are similarly co-expressed) and the remaining elements are replaced by 0.

Clearly, each of the above mentioned correlation measures have its own advantages and disadvantages. However, the Pearson's correlation coefficient is the most popular co-expression measure. The Pearson's correlation coefficient takes a value between -1 and 1 where values close to 1 or -1 show strong correlation. In particular, positive values correspond to positive correlation meaning that the expression value of one gene increases with the increase in the expression of its co-expressed gene and vice versa. Negative values correspond to negative correlation meaning that the expression value of one gene decreases with the increase in the expression of its co-expressed gene.

There exist also several methods to select the threshold value. The simplest approach consists in choosing a co-expression cutoff without looking at the data. Another approach is to use Fisher's Z-transformation that first calculates a z-score based on the number of samples and then converts the z-score into a p-value used to set the threshold.

The study of gene co-expression networks allows to identify genes that are controlled by the same transcriptional regulatory program, that are functionally related, or whose gene products are involved in a common biological process.

Modeling

According to the way gene co-expression networks are built the simplest model to represent them is by using the correlation matrix (in particular the matrix containing only 0 and 1 s obtained after applying the threshold) to build a graph. In particular, such binary adjacency matrix corresponds to an unweighted graph the nodes of which represent the genes and whose edges connect genes that are co-expressed. According to such a model all connections are equivalent.

A more complex model takes into account edge weights to store information about the corresponding values in the correlation matrix (before applying the threshold). In particular, in such a representation the graph is complete, meaning that there exists an edge between each pair of nodes, and the strength of the connection (i.e., co-expression) is given by the edge weights.

Applications

Biological networks found application is several research areas. As an example, the analysis of biological networks with respect to human diseases has led to the field of network medicine ([Barabási et al., 2011](#)). Indeed, a disease is usually a consequence of an abnormality in a complex intracellular and intercellular network and the study of biological networks can help in characterizing diseases. For instance, biological networks are analyzed to discover disease-gene associations with the aim of identifying relationships between disease phenotypes and genes. Even if traditional approaches, that do not use biological networks, are successful, often it is necessary to analyze experimentally tens or even hundreds of genes. To reduce experimental cost and efforts, protein-protein interaction networks or metabolic networks are used by disease gene prioritization methods to rank candidate genes that are probably related to a disease.

Another application of biological networks in network medicine is for the genome-wide association study (GWAS). The GWAS aims at studying DNA sequence variations in the human genome to discover their associations with diseases (or phenotypes). In particular, the network-based GWAS methods are based on two types of information: (i) interactions between genes or proteins and (ii) association information available from an existent GWAS.

Protein-protein interaction networks are widely used for predicting protein function. The standard way to predict protein function is based on the computation of sequence homology that is used to annotated proteins. However, for the great majority of proteins their function still remains unknown. Protein-protein interaction networks have been successfully used in this context and several approaches to predict protein function by analyzing this type of network have been proposed (e.g., [Fionda et al., 2009](#)).

Another area of application of biological networks regards complex networks. Complex networks research helps in representing, characterizing and modeling complex systems and phenomena and, thus, found a natural application in the analysis of biological networks (Costa *et al.*, 2008).

Closing Remarks

The various types of biological networks have been discussed in the previous sections independently from one another. However, it is important to underline that they are not independent inside the cell. For instance, on the one hand the state of the genes in the transcriptional regulatory network determines the activity of the metabolic network. On the other hand, the concentration of metabolites in the metabolic network determines the activity of transcription factors or proteins which regulate the expression of genes in the regulatory network. Thus, all the different types of biological networks form together a network of networks inside the cell that determines the overall behavior of the corresponding organism.

Indeed, several research groups have integrated different types of networks in their studies. As an example, gene expression networks have been integrated with protein-protein interaction networks in order to understand how one of them affect the other in different biological states. For example, the protein-protein interaction network of the yeast was merged with the gene expression network and the results showed that on the one hand most proteins of interacting complexes are expressed in a similar manner in the various stages of the cell cycle, and on the other hand, only a single or a small number of key proteins are expressed in a single phase. Thus, the complementing of these two types of networks allowed to identify such key elements.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Challenges in Creating Online Biodiversity Repositories with Taxonomic Classification. Disease Biomarker Discovery. Ecological Networks. Gene Regulatory Network Review. Investigating Metabolic Pathways and Networks. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Pathway Informatics. Protein-DNA Interactions. Quantification of Proteins from Proteomic Analysis

References

- Alakwaa, F.M., 2014. Modeling of gene regulatory networks: A literature review. *Journal of Computational Systems Biology* 1 (1), 102.
- Bachmaier, C., Brandes, U., Schreiber, F., 2014. Biological networks. In: Tamassia, R. (Ed.), *Handbook of Graph Drawing and Visualization*. CRC Press, pp. 621–651.
- Barabási, A.-L., Gulbahce, N., Loscalzo, J., 2011. Network medicine: A network-based approach to human disease. *Nature Reviews Genetics* 12 (1), 56–68.
- Chen, K.C., Calzone, L., Csikasz-Nagy, A., *et al.*, 2004. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell* 15 (8), 3841–3862.
- Chen, X.-w., Anantha, G., Wang, X., 2006. An effective structure learning method for constructing gene networks. *Bioinformatics* 22 (11), 1367–1374.
- Costa, L.F., Rodrigues, F.A., Cristino, A.S., 2008. Complex networks: The key to systems biology. *Genetics and Molecular Biology* 31 (3), 591–601.
- Fionda, V., Palopoli, L., 2011. Biological network querying techniques: Analysis and comparison. *Journal of Computational Biology* 18 (4), 595–625.
- Fionda, V., Palopoli, L., Panni, S., Rombo, S.E., 2009. A technique to search for functional similarities in protein–protein interaction networks. *International Journal of Data Mining and Bioinformatics* 3 (4), 431–453.
- Friedman, N., Linial, M., Nachman, I., Pe'er, D., 2000. Using Bayesian networks to analyze expression data. *Journal of Computational Biology* 7 (3–4), 601–620.
- Gillespie, D.T., 2007. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry* 58, 35–55.
- Glass, L., Kauffman, S.A., 1973. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology* 39, 103–129.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Karlebach, G., Shamir, R., 2008. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology* 9 (10), 770–780.
- Kauffman, S.A., 1993. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford: Oxford University Press.
- Keseler, I.M., Collado-Vides, J., Gama-Castro, S., *et al.*, 2005. EcoCyc: A comprehensive database resource for *Escherichia coli*. *Nucleic Acids Research* 33 (suppl 1), D334–D337.
- Li, S., Brazhnik, P., Sobral, B.W.S., Tyson, J.J., 2008. A quantitative study of the division cycle of *Caulobacter crescentus* stalked cells. *PLOS Computational Biology* 4 (1), Manca, V., Luca, B., 2008. Biological networks in metabolic P systems. *BioSystems* 91 (3), 489–498.
- Paun, G., 2000. Computing with membranes. *Journal of Computer and System Sciences* 61 (1), 108–143.
- Rischer, H., Oresic, M., Seppanen-Laakso, T., *et al.*, 2006. Gene-to-metabolite networks for terpenoid indole alkaloid biosynthesis in *Catharanthus roseus* cells. *Proceedings of the National Academy of Sciences* 103 (14), 5614–5619.
- Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W., 2002. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18, 261–274.
- Shmulevich, I., Gluhovsky, I., Hashimoto, R.F., Dougherty, E.R., Zhang, W., 2003. Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks. *Comparative and Functional Genomics* 4, 601–608.
- Steggles, L.J., Banks, R., Shaw, O., Wipat, A., 2007. Qualitatively modelling and analysing genetic regulatory networks: A Petri net approach. *Bioinformatics* 23 (3), 336–343.
- Van Regenmortel, M.H., 2004. Reductionism and complexity in molecular biology. *EMBO Reports* 5 (11), 1016–1020.
- Weaver, D.C., Workman, C.T., Stormo, G.D., 1999. Modeling regulatory networks with weight matrices. In: *Proceedings of the Pacific Symposium on Biocomputing*, pp. 112–123.
- Xiong, L., Wenjia, S., Chao, T., 2016. Adaptation through proportion. *Physical Biology* 13 (4), 046007.

Graph Theory and Definitions

Stefano Beretta, Luca Denti, and Marco Previtali, University of Milan-Biocca, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Nowadays, even in our daily life, we encounter several different situations in which objects, or more in general elements, are put into relations or simply connected with other objects, creating a network. Although in most cases it is not easy to spot these kind of structures, they are widely adopted and used for different purposes. As an example, if we consider a city map, cities and towns are the elements and streets represent the connections among them. The same is also valid for maps of the underground of a city or of the railway connections among different places. These very intuitive and easy to understand structures are called *graphs* and, although simple, they have a formal mathematical formulation on which several properties are defined. Moreover, in fields like computer science they constitute the basis of different data structures, which are then used by algorithmic procedures to efficiently solve specific problems. As an example, satellite navigators exploit a graph structure representing the map of the area of interest to find the best way to reach a specific location from the actual position. Recalling the previously mentioned example, in a simpler way, we solve the same problem when we decide which trains to take to go from a point A to a destination B, following the railway map.

As anticipated before, graphs are widely used in several different fields, ranging from mathematics to biology, and from computer science to physics. One of the main reasons of their widespread adoption is due to the fact that graphs are able to capture and model different real case scenario, helping scientists in finding solutions to specific problems. In fact, although graphs offer a simple and very intuitive graphical representation, they have a formal mathematical definition, on which several properties and algorithms are based (Cormen *et al.*, 2009; Diestel, 2012).

Mainly thanks to these aspects, two of the most influenced research fields in which graphs are employed are bioinformatics and systems biology (Doncheva *et al.*, 2012). In fact, since graphs are usually adopted to model relations between elements in biological datasets, several real case studies, like the analysis of the chromatin conformation inside the nucleus of cells (Merelli *et al.*, 2013; Tordini *et al.*, 2016) or the identification of biological pathways in gene regulatory networks (Karlebach and Shamir, 2008), take advantage of this formalism to efficiently represent and analyze the data. Other examples are their application to protein-protein interaction networks (Pizzuti and Rombo, 2014), in which are mainly employed for clustering analysis.

Graphs are also extensively studied, from a formal point of view, in computer science where they are fundamental tools for solving computational problems. For this reason, several bioinformatic approaches and tools rely on them to implement algorithms for solving biological problems, like genome assembly, sequence alignment, or Next-Generation Sequencing reads error correction. More precisely, and as an example, mapping nucleotidic reads to a gene transcript can be modeled as the alignment of sequences to a graph that represent the possible alternative splicing events (Beretta *et al.*, 2014). In the same context, to perform the assembly of molecular sequences, the majority of the developed techniques rely on a graph structure to face this problem, like building an overlap graph of the input short fragments, and then extracting the sequence of nodes that better represents the original one (Bonizzoni *et al.*, 2016; Myers *et al.*, 2000; Simpson and Durbin, 2012), or constructing the de Bruijn graph of the input long fragments, and then finding the different paths corresponding to the expressed gene transcripts (Grabherr *et al.*, 2011; Trapnell *et al.*, 2010).

Moreover, due to the novel developments in the aforementioned fields that lead to the production of huge quantities of data to be analysed, the role of structures like graphs is becoming more and more central. In fact, in order to develop more efficient techniques to deal with such data, graphs are often employed to design optimal procedures and also to achieve good performances from the applicative point of view (Gonnella and Kurtz, 2012).

For these reasons, a good knowledge of the graphs is fundamental to understand the majority of the state-of-the-art methods that deal with problems in bioinformatics and systems biology. In this article we will introduce the basic concepts and definitions of graphs and we will highlight their usage in bioinformatics and systems biology.

Background

A graph G is a pair $G=(V, E)$ where V is the set of vertices (also called nodes) of size $|V|$, and $E \subseteq V \times V$ is the set of edges. Each edge $e \in E$ is an element that *connects* two vertices in V and that represents some kind of relation between them. If e connects the nodes u and v , then it is common to refer to e as the pair $\{u, v\}$. The vertex set of the graph G is also referred to as $V(G)$, while its edge set as $E(G)$. Depending on the meaning of the relations between nodes, a graph can be either *directed* or *undirected*. In the former case, the connections between pairs of nodes are usually called *arcs*, and each arc e has an *orientation* so that e is represented as an ordered pair of vertices (u, v) with $u, v \in V$. In the latter case, edges do not have orientations and an edge e can be represented either as $\{u, v\}$ or $\{v, u\}$. Note that in directed graphs the ordering between the nodes in the arcs has a fundamental role and, therefore, two arcs $e_i=(u, v)$ and $e_j=(v, u)$ represent two different arcs (if $u \neq v$). Recalling the previously introduced examples, directed graphs can be used in biological networks and especially in metabolic networks to model the pathways of enzymes and metabolites (Bourqui *et al.*, 2007).

Moreover, given a vertex $v \in V$, the edge $\{v, v\}$ is called *loop* and, usually, graphs without loops are called *simple graphs*.

Following the previous distinction based on fact that arcs and edges are different concepts, a graph $G = (V, E)$ is graphically represented by drawing for each vertex $v \in V$ a dot and, for each edge $e \in E$ between vertices u and v , either a line connecting u and v if the graph is undirected, or an arrow directed towards v if the graph is directed.

Fig. 1 shows an example of an undirected graph (**Fig. 1(a)**) and an example of a directed graph (**Fig. 1(b)**). As one can see from these figures, the undirected graph in **Fig. 1(a)** has an edge connecting nodes u and v , while the one in **Fig. 1(b)** has two arcs linking the same nodes, i.e., (u, v) and (v, u) . The same also applies to nodes w and x of the examples in the same figures.

In the rest of this article, we will refer all the concepts and the definitions to a graph $G = (V, E)$ using the notations for undirected graphs if they apply to both directed and undirected graphs, while we will explicitly highlight the differences between the two when required.

The graph $G = (V, E)$ such that $V = \emptyset$ and $E = \emptyset$ is called *empty graph*.

Given an edge $e \in E$, with $e = \{u, v\}$, the vertices u and v are called *adjacent* or *neighbor* vertices and are said to be *incident* with e . More precisely, if G is an undirected graph, $e = \{u, v\}$ is usually said to be *incident on* both vertices u and v , whereas if G is a directed graph, $e = (u, v)$ is said to be *incident from* or *leaving* vertex u and *incident to* or *entering* vertex v . The set of all edges entering or leaving a vertex v is usually denoted by $E(v)$.

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs, if $V' \subseteq V$ and $E' \subseteq E$, then we say that G' is a *subgraph* of G and that G is a *supergraph* of G' . In the following we will denote the subgraph and supergraph relations using the $\subseteq$ and $\supseteq$ symbols, respectively. More precisely, if G' is a subgraph of G then we will denote it by $G' \subseteq G$. Let $G = (V, E)$ be a graph and let V' be a subset of V , that is $V' \subseteq V$. Then, the subgraph of G induced by V' is the graph $G' = (V', E')$, where E' is the set of edges incident in the vertices of V' . More formally, we say that E' is the set $\{\{u, v\} \in E : u \in V' \wedge v \in V'\}$. An example of induced subgraph is shown in **Fig. 2**.

A *weighted* graph is a graph with an associated function f_w that assigns to every vertex or to every edge a value that is usually referred to as the *weight*. The function f_w is referred to as the *weight function*. Depending on whether the weight function's domain is the set of vertices or the set of edges, we can refer to a weighted graph either as a *vertex-weighted* graph or as an *edge-weighted* graph. Formally, a vertex-weighted graph is the union of a graph $G = (V, E)$ and a weight function $f_w: V \rightarrow \mathbb{R}$, while an *edge-weighted* graph is the union of a graph $G = (V, E)$ and weight function $f_w: E \rightarrow \mathbb{R}$. In the former case, given a vertex $v \in V$, the weight associated with v can be denoted as $f_w(v)$, while in the latter case of edge-weighted graph, given an edge $\{u, v\} \in E$, the corresponding weight can be denoted as $f_w(u, v)$. In bioinformatics, one of the applications of weighted graphs is in the assembly of sequences (genomes, transcripts, or meta-genomes) starting from a set of short reads which must be assembled. In such a case, the majority of the

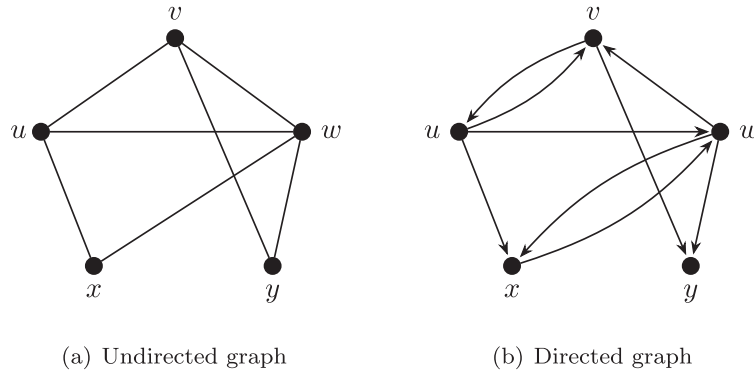


Fig. 1 Examples of both undirected (a) and directed (b) graphs.

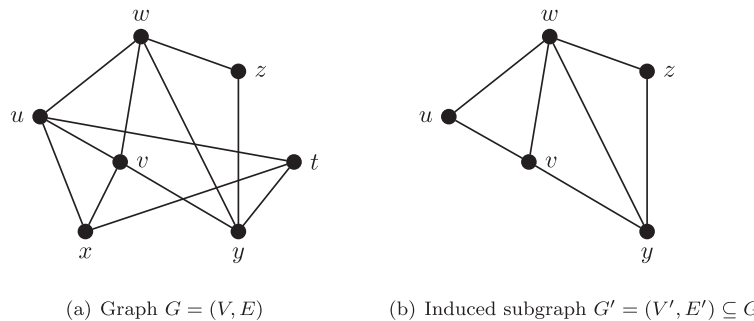


Fig. 2 Example of induced subgraph: starting from a graph $G = (V, E)$ (a), the subgraph $G' = (V', E')$ induced by the vertex set $V' = \{u, v, w, y, z\}$ is shown (b).

assembly algorithms employs a graph structure to reconstruct one or more paths corresponding to the solution and, to do so, weights are used to represent the support in terms of number of reads, which helps in discarding wrong solutions and reconstruct the correct one(s) (Zerbino and Birney, 2008).

A *bipartite* graph is a graph $G=(V, E)$ in which V can be partitioned into two disjoint subsets V_1 and V_2 such that the vertices contained in the same subset are not adjacent, i.e., all the edges of E are incidents on a vertex of V_1 and on a vertex of V_2 . Formally, given a graph $G=(V, E)$, then G is a bipartite graph if there exists two sets $V_1 \subseteq V$ and $V_2 \subseteq V$ such that the following properties are true:

- $V_1 \cap V_2 = \emptyset$;
- $V_1 \cup V_2 = V$;
- $\forall \{u, v\} \in E: (u \in V_1 \wedge v \in V_2) \vee (v \in V_1 \wedge u \in V_2)$.

An example of bipartite graph is shown in Fig. 3. Bipartite graphs are usually adopted to model two categories of elements, and the relations between elements belonging to the two different categories. Examples of this situation can be found, for example, when analyzing long non-coding RNAs (lncRNAs) with respect to their binding proteins, which results in a graph connecting nodes representing lncRNAs with others representing targeted proteins (Ge et al., 2016).

Finally, a *multigraph* $G=(V, E)$ is an undirected graph in which multiple edges can connect the same pair of vertices, that is E is a multi-set of edges. An example of multigraph is shown in Fig. 4. Multigraphs are important because in some cases there is the need of representing different types of information between two elements, each one with a different meaning.

One of the most important aspects when dealing with graphs is their representation since, in many cases, it can affect the performance of the methods that employ them. From this point of view, the two most commonly used representations of a graph $G=(V, E)$ are a collection of *adjacency lists* and an *adjacency matrix*.

The adjacency lists representation of a graph $G=(V, E)$ consists of an array *Adj* of lists having size $|V|$, that is, one for each vertex in the vertex set V , such that for each vertex $v \in V$, $Adj[v] = \{u \in V: \{v, u\} \in E\}$. This representation is recommended for *sparse* graphs, which are graphs having a small number of edges, i.e., $|E| \ll |V|^2$, where $|V|^2$ is an approximation of the maximum number of edges of a graph having $|V|$ vertices. In fact, this approach allows to store only the edges present in the graph and to keep the space required to store the graph G linear with its size. The drawback of this representation is that to search whether the edge $\{u, v\}$ is in

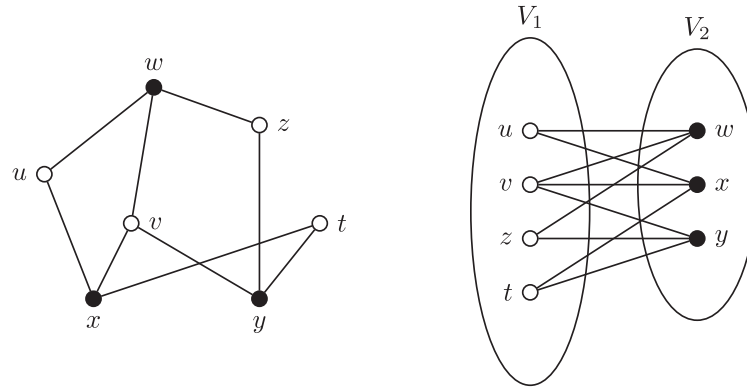


Fig. 3 Example of bipartite graph $G=(V, E)$, in which nodes can be partitioned into two subsets: $V_1=\{u, v, z, t\}$ (empty circle nodes) and $V_2=\{w, x, y\}$ (black circle nodes). Here, two different ways of drawing the bipartite graph G are shown.

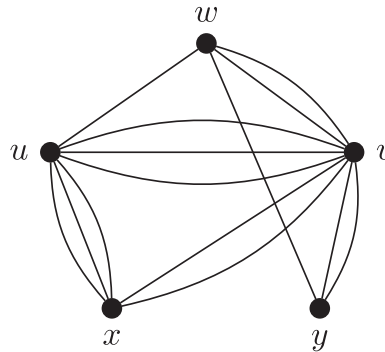


Fig. 4 Example of multigraph.

the graph it is necessary to scan the whole adjacency list of u . This means that in the worst case scenario, that is, when each vertex is connected to all the others, this test requires $\mathcal{O}(V)$ time. The adjacency list of the graph shown in Fig. 1(a) is:

u	$= \{v, w, x\}$
v	$= \{u, w, y\}$
w	$= \{u, v, x, y\}$
x	$= \{u, w\}$
y	$= \{v, w\}$

On the other hand, for *dense* graphs in which $|E| \simeq |V|^2$, or in situations when the existence of an edge must be checked quickly, the adjacency matrix representation is more suitable. This representation of a graph $G=(V, E)$ consists of a $|V| \times |V|$ matrix A , such that

$$A[u, v] = \begin{cases} 1 & \text{if } \{u, v\} \in E \\ 0 & \text{otherwise} \end{cases}$$

This way of representing a graph requires a fixed $|V|^2$ bits space but allows to test the presence of an edge in $\mathcal{O}(1)$ time. For this reason, storing a sparse graph using an adjacency matrix leads to a matrix having the majority of values set to zero, wasting space.

Moreover, if $G=(V, E)$ is a *edge-weighted* graph, the matrix A can be adapted to store the weights of the edges, instead of the only presence or absence of a connection, i.e., 0 and 1, respectively. More precisely, $A[u, v]=f_w(u, v)$, that is the weight associated with the edge $\{u, v\} \in E$, and 0 otherwise, that is, if $\{u, v\} \notin E$.

The adjacency matrix of the graph shown in Fig. 1(a) is:

	u	v	w	x	y
v	0	1	1	1	0
w	1	0	1	1	1
x	1	1	0	0	1
x	1	1	0	0	0
y	0	1	1	0	0

Notice that both these representations can be applied to undirected and also to directed graphs. Anyway, it must be also noticed that for undirected graphs the adjacency matrix is symmetric and so $A[u, v]=A[v, u]$, for each $u, v \in V$, meaning that it is possible to consider only the upper triangular part (with respect to the diagonal). On the other hand, this is not valid for directed graphs, for which the two triangular parts (upper and lower) of the matrix contain the direction of the arcs, since $A[u, v]$ refers to the arc $(u, v) \in E$, while $A[v, u]$ refers to the arc $(v, u) \in E$.

Methodologies

In this section we will focus on bipartite graphs and, in particular, we will describe an algorithmic procedure to verify if a given graph is bipartite. This will be done without entering into the formal aspects of this problem, nor the technical algorithmic details of the procedure. In fact, the main aim of this part is to give an idea of a possible solution to the bipartite check problem, but also to give an example of how it is possible to design an algorithm working on graphs.

Bipartite graphs are quite interesting and are employed in several different studies and, for this reason, there is often the necessity to check if a given graph $G=(V, E)$ is bipartite or not. Although for small examples this check is quite easy and sometimes it could be done just by looking at the graphical representation of the graph, when the number of vertices grows this problem becomes harder. Since, by definition, in a bipartite graph the set of vertices V can be partitioned into two subsets, namely V_1 and V_2 , in such a way that there are no edges connecting two vertices belonging to the same subset, it is possible to assign a color to each vertex according to the subset it belongs. As an example, it is possible to color vertices of V_1 with "red" and those of V_2 with "blue", so that each edge will connect a red and a blue vertex. More precisely, in a bipartite graph the vertices can be colored with two colors, corresponding to the two subsets, so that the vertices incident to each edge will always have two different colors. This property is useful especially in determining if a given graph $G=(V, E)$ is bipartite or not. In fact, it must be observed that if there is a subgraph composed of three vertices, in which every vertex is adjacent to the other two, i.e., a "triangle", it is not possible to

assign two colors, say red and blue, in such a way that adjacent vertices will have different colors. Let us consider the subgraph composed of three vertices u, v, z , and three edges $\{u, v\}, \{u, z\}, \{v, z\}$; if we assign to the first vertex, say u the color red, then its neighbours v and z should be blue, but, since they are adjacent each other, i.e., there exists the edge $\{v, z\}$, this will not be possible. For a more formal and detailed characterization of bipartite graphs, we refer the reader to [Diestel \(2012\)](#).

Based on the above property, one of the algorithms to test whether a given graph $G = (V, E)$ is bipartite tries to assign different colors to the vertices of G in such a way that each edge will connect vertices having different colors. Without entering into the details of the algorithmic procedure to visit the graph, which is not part of this paper, we will describe the intuitive idea to verify if the graph $G = (V, E)$ is bipartite. The idea of the visit is the following: once you visit a vertex you put all its neighbours in the queue of the vertices that must be visited, so that, at the next step you pop out the next vertex from this queue, visit it, and put its neighbours in the queue. This operation is repeated until the queue is empty, that is, until all the vertices of the graph are visited. Let us denote as c_1 and c_2 the two possible colors to be assigned to a vertex. Initially, none of the vertices of V is colored. First of all, consider the vertex set V of the input graph G , and select a vertex $v \in V$ as a starting point of the visit. Then, color v with c_1 , select the set of its neighbours, color each of them with c_2 , and add them to the queue of vertices to be visited. Now, pop out the first vertex in the queue, say u , select its neighbours, and when considering them, each vertex z can be in one of the three possible situations:

- z is not yet colored, so color it with c_1 (the opposite of the color of u) and put it into the queue of the vertices to be visited;
- z is colored with c_1 , that means that it has been already visited and its color agrees with that of u , so the visit continues (i.e., since u and z are adjacent they must have different colors);
- z is colored with c_2 , that means that it has been already visited but its color does not agree with that of u , and so the visit ends denoting the fact that the graph is not bipartite.

This latter case model the fact that an edge $\{u, z\}$ connects two vertices u and z having the same color, violating the property of bipartite graphs. This situation results from the fact that, during the visit, one of vertices adjacent to u and one of the vertices adjacent to z was previously colored with the other color. If the visit of the graph with the vertex coloring ends without entering in the third case, then the graph $G = (V, E)$ is bipartite, and the two subset of vertices can be obtained based on the two colors assigned during the visit.

It is important to notice that the previously described procedure to check whether a given graph is bipartite does not depend on the choice of the initial vertex of the visit.

Finally, we would like to point out that, although simple, this is a good example of algorithmic procedure designed to solve a specific problem on graphs. Formal details will require the introduction of different concepts and properties that are not part of this article.

Conclusions

Graphs are very important mathematical concepts that are employed in several different fields and so, due to their importance, several properties and theorems have been developed. Moreover, thanks also to their intuitive and easy-to-understand structure, they are often used to model biological networks, like protein-protein interaction networks, gene regulatory networks, or metabolic networks. In addition to systems biology, also bioinformatics takes advantage of graph structures, both to model specific phenomena, like the alternative splicing events occurring in an expressed gene, or to represent specific data structures in the computational procedures designed to solve specific problems. In this latter case, graphs play a fundamental role, especially in guaranteeing the efficiency of the implemented procedure. As an example, one could think at the problem of assembling genomic sequences starting from a set of Next-Generation Sequencing reads, which is usually composed of millions and even billions of short, and also erroneous, short (sub)strings of the sequenced genome. In order to reconstruct the original sequence, it is necessary to deal with this huge amount of information, but, thanks also to the use of graphs in representing the strings (which are usually the so called k -mers), the overall process can be run on a workstation in a quite fast time.

Although these are only a small number of the examples that highlight the importance of graphs in systems biology and bioinformatics, it is quite easy to guess that graphs can be found in many problems. For this reason, a good knowledge of them is fundamental and could be very helpful in doing specific studies or when facing many biological problems.

This article is intended to be an introductory review of the basic concepts about graphs, which could be useful for those that are approaching graphs for the first time. Here, we started with the definition of graphs and we introduced the notions of vertex and edge, adjacency and incidence of vertices, and we also highlighted the difference between directed and undirected graphs. Starting from this point, we explained the meaning of subgraphs of a given graph, induced by a subset of its vertices, and of weighted graphs, both the node- and edge-weighted ones. Then, we introduced two type of graphs: bipartite graphs, in which the set of vertices can be partitioned into two subsets so that there is no edge connecting two vertices in the same subset, and multigraphs, in which there is the possibility of having multiple edges connecting the same pair of vertices. Moreover, due their relevance in different applications, we focused our attention on bipartite graphs by describing a procedure to check whether a given graph is bipartite. Finally, we explained two possible representations of graphs, which are usually adopted by the computational methods using them. The first one is the adjacency lists of the vertices, in which for each vertex there is a list of its neighbours, while the second one is the adjacency matrix in which each cell correspond to a pair of vertices, and there is a 1 value if the corresponding

edge exists, and 0 otherwise. We also highlighted advantages and disadvantages of both the representations, and their extension to weighted graphs.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Natural Language Processing Approaches in Bioinformatics

References

- Beretta, S., Bonizzoni, P., Vedova, G.D., Pirola, Y., Rizzi, R., 2014. Modeling alternative splicing variants from RNA-seq data with isoform graphs. *Journal of Computational Biology* 21, 16–40.
- Bonizzoni, P., Vedova, G.D., Pirola, Y., Previtali, M., Rizzi, R., 2016. LSG: An external-memory tool to compute string graphs for next-generation sequencing data assembly. *Journal of Computational Biology* 23, 137–149.
- Bourqui, R., Cottret, L., Lacroix, V., *et al.*, 2007. Metabolic network visualization eliminating node redundancy and preserving metabolic pathways. *BMC Systems Biology* 1, 29.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. *Introduction to Algorithms*, third ed. MIT Press.
- Diestel, R., 2012. *Graph Theory*, Graduate Texts in Mathematics, fourth ed., 173. Springer.
- Doncheva, N.T., Assenov, Y., Domingues, F.S., Albrecht, M., 2012. Topological analysis and interactive visualization of biological networks and protein structures. *Nature Protocols* 7, 670–685.
- Ge, M., Li, A., Wang, M., 2016. A bipartite network-based method for prediction of long non-coding RNA–protein interactions. *Genomics, Proteomics & Bioinformatics* 14, 62–71.
- Gonnella, G., Kurtz, S., 2012. Readjoinder: A fast and memory efficient string graph-based sequence assembler. *BMC Bioinformatics* 13, 82.
- Grabherr, M.G., Haas, B.J., Yassour, M., *et al.*, 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* 29, 644–652.
- Karlebach, G., Shamir, R., 2008. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology* 9, 770–780.
- Merelli, I., Liò, P., Milanese, L., 2013. Nuchart: An R package to study gene spatial neighbourhoods with multi-omics annotations. *PLOS ONE* 8, e75146.
- Myers, E.W., Sutton, G.G., Delcher, A.L., *et al.*, 2000. A whole-genome assembly of *Drosophila*. *Science* 287, 2196–2204.
- Pizzuti, C., Rombo, S.E., 2014. Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods. *Bioinformatics* 30, 1343.
- Simpson, J.T., Durbin, R., 2012. Efficient de novo assembly of large genomes using compressed data structures. *Genome Research* 22, 549–556.
- Tordini, F., Aldinucci, M., Milanese, L., Liò, P., Merelli, I., 2016. The genome conformation as an integrator of multi-omic data: The example of damage spreading in cancer. *Frontiers in Genetics* 7.
- Trapnell, C., Williams, B., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Biotechnology* 28, 511–515.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18, 821–829.

Network Properties

Stefano Beretta, Luca Denti, and Marco Previtali, University of Milan-Bicocca, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A graph is a mathematical structure that represents relationships (the edges) between elements of a domain (the nodes) in a very intuitive way. In particular, a graph can be represented as a set of nodes, corresponding to the elements, and edges connecting pairs of nodes, representing the fact that two connected nodes have a relation. Taking advantage of its formal definition, several properties and notions have been defined in literature during the years, contributing to the widespread diffusion of graphs in different fields such as physics, computer science, and biology. Strictly related to this latter research area, those fields that involve specific computational aspects, such as bioinformatics and systems biology, make use of graphs in many different applications and studies.

In the context of Next Generation Sequencing consider, for example, the problem of reconstructing the original sequence starting from a huge set of reads, corresponding to substring obtained through a sequencing process. To solve this problems, the majority of the bioinformatic approaches use a graph to represent the input reads, which are usually connected based on their sequence overlap, and tries to assemble them (see Gnerre *et al.*, (2011); Zerbino and Birney (2008) for some examples). Although they vary in the type of adopted graph and in the technique used to reconstruct the sequence, the cores of the methods rely on basic notions coming from graph theory. In fact, several studies have been proposed in literature, ranging from the more theoretical ones (see Beerenwinkel *et al.*, 2015) to more applicative techniques (see Pertea *et al.*, 2015).

On the other hand, graphs are also used to directly model specific biological phenomena on which, then, analysis is performed in order to obtain relevant results. Examples of these applications are the metabolic networks, in which the biochemical reactions of a specific organism are analyzed to extract the most relevant pathways (see Kanehisa *et al.*, 2010; Feist *et al.*, 2009) or the protein-protein interaction networks, in which proteins are connected to each other based on their interactions within a cell or in a specific organism (see Rual *et al.*, 2005).

Due to the high importance of graphs, in this work we will present some basic properties and definitions that are fundamental for a good understanding of this mathematical structure. More precisely, we will start by recalling some basic definitions and then we will describe some notions related to the concept of node degree for undirected graphs. We will also refine this definition for directed graphs for which we can distinguish between in- and out-degree. Starting from this property, we will introduce the notion of k -regular graph and of complete graph, and we will highlight some properties and case studies involving this latter kind of graphs. After that, we will focus on the density measure by giving its definition on both directed and undirected graphs and, finally, based on this measure we will describe two kind of graphs, namely dense (highly connected) and sparse (poorly connected).

Background

Before introducing the main concepts of this work, let us recall some basic notions on graphs. A graph is usually defined as $G=(V,E)$, in which V represents the set of vertices, while E is the set of edges or arcs between pairs of vertices. More precisely, if G is an undirected graph, then the elements of E are usually called edges and are denoted as unordered pairs $\{u,v\} \in E$, with $u,v \in V$. In this case, since there is no orientation associated with the edges, they are sets of two vertices of V , and consequently $\{u,v\} = \{v,u\}$. On the other hand, if G is a directed graph, then the elements of E are usually called arcs and are denoted as ordered pairs of vertices $(u,v) \in E$, with $u,v \in V$. In this case the arc (u,v) is oriented from vertex u to vertex v , and as a consequence $(u,v) \neq (v,u)$, since (u,v) and (v,u) represent two distinct arcs.

Moreover, given an edge $e \in E$, with $e = \{u,v\}$, the vertices u and v are said to be adjacent or neighbor vertices, and are incident on e . Again, if G is an undirected graph, the edge $e = \{u,v\}$ is usually said to be incident on both vertices u and v , whereas if G is a directed graph, then the arc $e = (u,v)$ is said to be incident from (or leaving) vertex u and incident to (or entering) vertex v .

Now, let us consider an undirected graph $G=(V,E)$. The *degree* of a vertex corresponds to the number of edges incident on it. The degree of a vertex $v \in V$ is usually denoted as $\deg(v)$ and can be formally defined as the cardinality of the set $\{\{v,u\} \in E : u \in V\}$. A vertex with degree equals to 0 is said to be *isolated*, since it has no connections with the other vertices of the graph. As an example, consider the graph in Fig. 1(a): the vertex u has degree 3, while the vertex x is isolated.

Let us now consider a directed graph $G=(V,E)$. In a directed graph, for each vertex it is possible to consider three different degrees: *in-degree*, *out-degree*, and *degree*. The in-degree of a vertex $v \in V$, denoted as $\deg^-(v)$, is the number of edges incident to v and, formally, it corresponds to the cardinality of the set $\{\{u,v\} \in E : u \in V\}$. The out-degree of a vertex $v \in V$, denoted as $\deg^+(v)$, is the number of edges incident from v and, formally, it corresponds to the cardinality of the set $\{\{v,u\} \in E : u \in V\}$. Finally, the degree of a vertex $v \in V$, denoted in the same way as for undirected graphs as $\deg(v)$, is equal to its in-degree plus its out-degree, that is, $\deg(v) = \deg^-(v) + \deg^+(v)$. Starting from the previous definitions, in a directed graph, a vertex v can be classified as:

- isolated vertex, if $\deg(v)=0$ (vertex x in Fig. 1(b));
- source vertex, if $\deg^-(v)=0$ (vertex u in Fig. 1(b));
- sink vertex, if $\deg^+(v)=0$ (vertex y in Fig. 1(b));
- internal vertex, if $\deg^-(v) \neq 0$ and $\deg^+(v) \neq 0$ (vertex w in Fig. 1(b)).

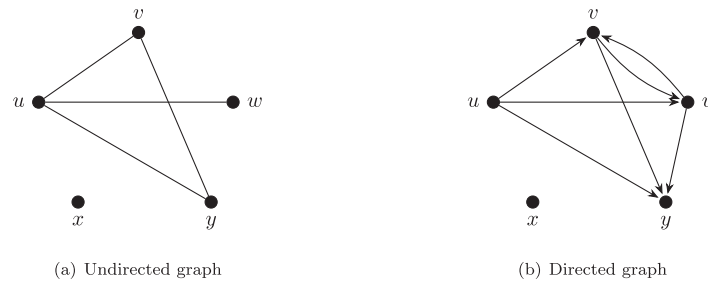


Fig. 1 In (a) an example of undirected graph is shown. In this example vertices u, v, w, y have degree 3, 2, 1, 2, respectively. On the other hand, in (b) an example of directed graph is reported. Here, vertex u has in-degree 0 and out-degree 3, while vertex y has in-degree 3 and out-degree 0. The other two vertices v and w have in-degree 2 and out-degree 2. In both the graphs vertex x has degree 0.

The node degree is a fundamental concept, since it is exploited in several different biological applications, and can be used to analyze biological networks. As an example, in Piraveenan *et al.* (2012) new measures are defined to assess the tendency of a node to connect with other nodes that have similar degree, extending the previously defined notion of “assortativity”. Similarly, in Milenković *et al.* (2011) new measures based on the degree of vertices are introduced to identify the most relevant genes in a protein-protein interaction network. In fact, as pointed out in Ideker and Sharan (2008), when studying diseases, genes that play an important role have a high number of connections in the protein-protein interaction network. In addition, it has been shown in Dyer *et al.* (2008) that the majority of the pathogens that are viral and bacterial have the tendency to interact with proteins having a high-degree, or with those that are central in many paths of the network.

In addition to the aforementioned examples in which the concept of node degree plays a central role, there are other bioinformatic applications that take advantage of it. More precisely, one of the ways in which graphs are used, is for representing the input reads obtained with a sequencing process. In this context, and in particular in studies involving the analysis of the DNA of specific bacteria having circular chromosomes, one of the goals is to reconstruct this sequence by assembling the input reads (Blattner *et al.*, 1997; Loman *et al.*, 2015). This is usually done by taking advantage of the overlap among them, but, mainly due to errors in the input sequence, the solution of this problem is not so straightforward. For a more complete introduction on the analysis of Next-Generation Sequencing data for bacterial genomes, we refer the reader to Edwards and Holt (2013). From a computational point of view, this problem corresponds to the that of finding a circular path in the graph that passes through all the edges. Although we do not formally define these concepts, the idea is that we want to find a way in the graph to return to the starting point, by traversing all the edges the minimum number of times. Recalling the assembly problem, this circular path will correspond to the assembled (circular) genome, obtained by merging the overlapping reads found along the path. The variant of the problem in which we want to pass through each edge only once, is called *Eulerian walk*, and it has a solution only if all the vertices in the graph have an even degree. Moreover, for directed graphs, other conditions for the existence of an Eulerian walk are that: (i) at most one vertex v has $\deg^+(v) - \deg^-(v) = 1$, (ii) at most one vertex u has $\deg^-(u) - \deg^+(u) = 1$, and (iii) all the other vertices have equal in-degree and out-degree. Once assessed its existence, the walk can be reconstructed by exploiting some algorithms proposed in literature (Fleury, 1883; Fleischer, 1991).

A very well-known type of graphs, which are involved in different studies, are the so called regular graphs. Given a graph $G=(V,E)$, if all its vertices have the same degree k , then G is said to be k -regular. One of the examples of these kind of graphs are the lattices, in which vertices can be thought as disposed as a (usually regular) grid, and each vertex is connected with its grid neighbours, that is, it has 4 edges. To be more precise, in order to be regular a graph, the lattice must connect the vertices on the border, that is, those in first/last column/row with the corresponding ones in the last/first column/row, respectively. This graph is called *toroidal*.

Starting from the previously introduced concepts, we are now able to define the concept of *complete graphs*. More precisely, an undirected graph $G=(V,E)$ is complete if for every pair of distinct vertices in V , there exists an edge in E connecting them. Formally, for each pair of vertices $u, v \in V$ such that $u \neq v$, then $\{u, v\} \in E$. From this definition it is easy to observe that a complete graph $G=(V,E)$ having $|V|$ vertices has exactly $\frac{|V|(|V|-1)}{2}$ edges. Since in a complete graph there exists an edge between every pair of vertices, that number is the maximum number of possible (distinct) edges in a graph. Fig. 2 shows some representations of complete undirected graphs having vertex sets of sizes ranging from 2 to 9.

The definition of complete graph we introduced for undirected graphs, can be extended also to the directed ones. In particular, in a directed complete graph $G=(V,E)$, for every pair of distinct vertices $u, v \in V$, that is $u \neq v$, there exist two arcs connecting them, i. e. $(u, v) \in E$ and $(v, u) \in E$. In this case, a complete directed graph $G=(V,E)$ having $|V|$ vertices has $|V|(|V|-1)$ arcs. As an example, Fig. 3 shows some possible representations of complete directed graphs having vertex sets of sizes ranging from 2 to 5.

Complete graphs, which are usually referred to as *cliques*, are a very important type of graphs, and several problems are focused on the identification of complete (and usually maximal) subgraphs. Let us recall the notion of subgraph. Given a graph $G=(V,E)$ and a subset $V' \subseteq V$ of vertices of V , then the subgraph of G induced by V' is the graph $G'=(V',E')$, where E' is the set of edges incident in the vertices of V' . More formally, we say that E' is the set $\{\{u, v\} \in E : u \in V' \wedge v \in V'\}$. A very important problem related to the concept of complete (sub)graphs asks to extract, from a given graph, the maximal clique, that is, the maximum subset of vertices $V' \subseteq V$ for which the induced subgraph $G'=(V',E')$ is a clique.

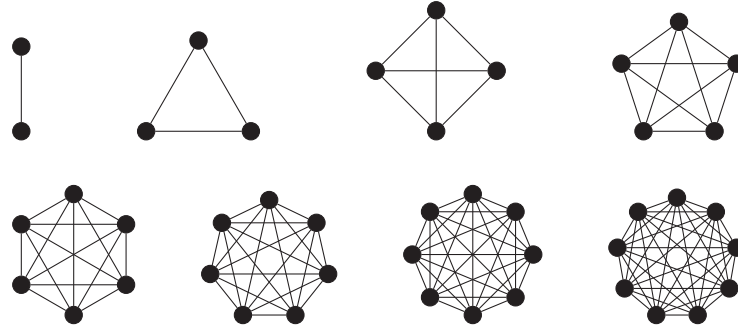


Fig. 2 Examples of complete undirected graphs, having the number of vertices ranging from 2 (upper leftmost graph) to 9 (lower rightmost graph).

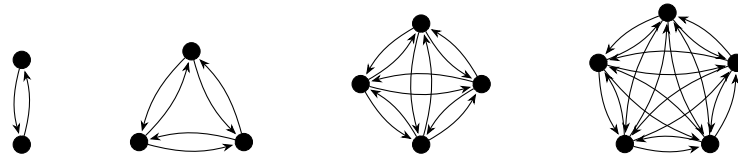


Fig. 3 Examples of complete directed graphs, having the number of vertices ranging from 2 (leftmost graph) to 5 (rightmost graph).

This latter problem has several applications in both systems biology and bioinformatics. More precisely, one of the main problems for which protein-protein interaction networks are adopted is the identification of the complete set of interactions among the proteins in a cell. In this context, cliques represent a subset of vertices, that is proteins, having pairwise connections allowing the possibility to identify the interactions among a great number of proteins (Yu *et al.*, 2006). Other examples of maximal complete subgraph identifications are in gene network analyses, in which cliques represent local clusters, containing proteins of likely similar biological function (Ciriello *et al.*, 2012) or when studying the effect of microRNA with cancer networks (Volinia *et al.*, 2010). Moreover, in bioinformatics complete subgraphs can be used to reconstruct the structure of a viral quasispecies from Next-Generation Sequencing data obtained from a set of mixed virus samples (Tpfer *et al.*, 2014) or to compute the distance between tree structures, such as those resulting from the folding of RNA molecules (Fukagawa *et al.*, 2011).

Another notion that is strictly connected with that of complete graphs is the *complete bipartite graph*, also called *biclique*. More precisely, a bipartite graph $G=(V,E)$ in which $V=V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, and no edge connects two vertices of the same subset, is a biclique if for every pair of vertices of the two subsets there exists an edge in E connecting them. Formally, for each $v \in V_1$ and $u \in V_2$, $\{u,v\} \in E$. From this definition it is easy to observe that the number of edges in G is $|V_1| \cdot |V_2|$. Examples of this kind of graph are the so called *stars*, which are complete bipartite graphs in which one of the two subsets is composed of only one vertex.

Methodologies

In this section we will introduce some notions related to the concept of density, which is very useful especially when studying biological networks.

Now, for ease of exposition, in the rest of the section we will consider only graphs $G=(V,E)$ without loops, that is, without edges $\{v,v\}$, with $v \in V$.

The density of a graph $G=(V,E)$, which is usually denoted as δ_G , is a measure to express the relationship between its number of edges and its number of vertices. If G is an undirected graph, the density δ_G can be computed as

$$\frac{2 \cdot |E|}{|V| \cdot (|V| - 1)}$$

since each single edge $\{u,v\} \in E$, with $u \neq v$, ideally represents two different arcs, namely (u,v) and (v,u) .

On the other hand, if G is a directed graph, instead, the density can be computed as

$$\frac{|E|}{|V| \cdot (|V| - 1)}$$

For example, the density of the graph shown in Fig. 1(a) is equal to $8/20=0.4$, while the density of graph shown in Fig. 1(b) is equal to $6/20=0.3$.

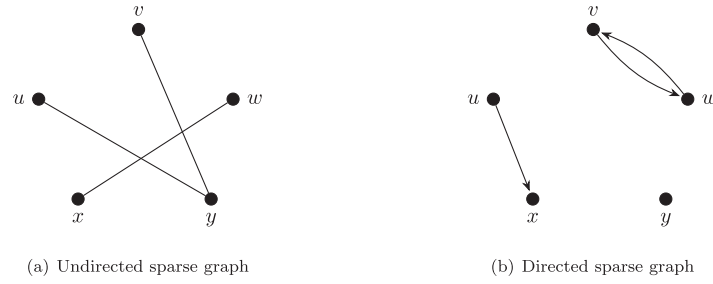


Fig. 4 Examples of sparse graphs. In (a) an example of undirected graph having low density ($\delta_G=6/20=0.3$) is shown. In (b) an example of directed graph having low density ($\delta_G=3/20=0.15$) is reported.

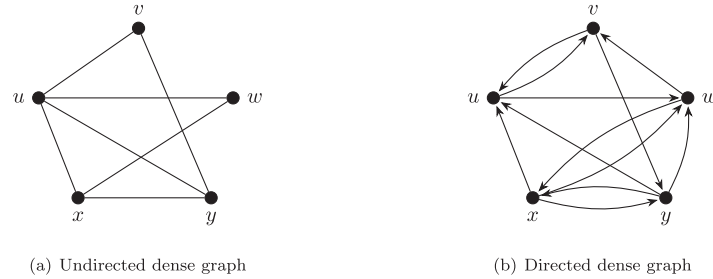


Fig. 5 Examples of dense graphs. In (a) an example of undirected graph having high density ($\delta_G=14/20=0.7$) is shown. In (b) an example of directed graph having high density ($\delta_G=12/20=0.6$) is reported.

Based on this measure, it is possible to define two types of graphs. More precisely, a graph $G=(V,E)$ with only a few edges i.e., a graph with low density ($\delta_G \leq 0.5$, or $|E| \ll |V|^2$, or $|E| \approx |V|$) is said to be *sparse*. [Fig. 4\(a\)](#) shows a sparse undirected graph, while [Fig. 3](#) a sparse directed graph.

On the other hand, a graph $G=(V,E)$ with a number of edges close to the maximal number of (possible) edges i.e., a graph with high density ($\delta_G \geq 0.5$, or $|E| \approx |V|^2$) is said to be *dense*. [Fig. 5\(a\)](#) shows a dense undirected graph, while [Fig. 5\(b\)](#) a dense directed graph.

Finally, notice that a complete graph $G=(V,E)$ has density equal to 1, i.e. $\delta_G=1$.

Starting from the previous classification of graphs, based on the density measure, we would like to point out that, the fact that a graph is sparse or dense can influence the way it should be represented. In particular, although it strictly depends on the specific application and also on the operations that must be performed on the graph, it is possible to prefer different representations for these two kind of graphs. If a graph is sparse then it could be convenient to explicitly store the edges with a set of *adjacency lists* in order to save some space, although each operation (usually) requires to scan through those lists. On the other hand, if a graph is dense, then an *adjacency matrix* representation could be more suitable since it allows to store all the possible edges (that are $|V| \times |V|$), also guaranteeing good performance on some operations (like to check the existence of an edge). More details on these aspects can be found in [Cormen et al. \(2009\)](#).

As for complete graphs, also for these two kind of density-based graphs, in most of the studies and the applications, the interest is in identifying subgraphs that are dense (or sparse, depending on the final goal). In fact, considering for example the previously mentioned biological networks and in particular the protein-protein interaction ones, although the entire network could be sparse, there could exist some dense subgraphs which constitute the most relevant interactions. Several approaches have been proposed to solve this problem, exploiting different techniques such as clustering and subgraphs enumerations (see [Adamcsek et al., 2006](#); [Nepusz et al., 2012](#)), or data mining algorithms (see [Bahmani et al., 2012](#); [Tsourakakis et al., 2013](#)).

Conclusions

Thanks to the combination of an intuitive representation of relations between elements, and a formal mathematical structure, graphs are used in many fields, ranging from physics to compute science, and from mathematics to biology. As a consequence of this fact, they play a central role in several studies, especially in areas like systems biology and bioinformatics in which different properties and methods are exploited to efficiently solve different problems. These notions, although very intuitive and easy to understand, constitute a strong basis for a better understanding of graph structures, and are fundamental to develop new methods and to perform specific studies.

In this work we presented the basic properties of graphs and, in particular, we explained the concept of node degree, starting from its definition with respect to both directed and undirected graphs. Based on this definition, we also introduced two types of graphs, that are the so called k -regular ones and the complete graphs. Moreover, when explaining these concepts we provided some examples of their applications in different studies present in literature, introducing also some property arising from them.

After that, we focused our attention on a very important measure that has been defined on graphs, that is, the density. We provided the formal definition, on both directed and undirected graphs, and we described two types of graphs that it is possible to identify, based on this latter measure: dense and sparse graphs. More precisely, the former ones are graphs having a density greater than 0.5, that correspond to fact of having a high number of edges, while the latter ones are graphs having density smaller than 0.5, that correspond to the fact of having a small number of edges. Finally, we gave some additional notions on these types of graphs and we provided some examples.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Quantification of Proteins from Proteomic Analysis

References

- Adamcsek, B., Palla, G., Farkas, I.J., Derényi, I., Vicsek, T., 2006. Cfinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* 22, 1021–1023.
- Bahmani, B., Kumar, R., Vassilvitskii, S., 2012. Densest subgraph in streaming and mapreduce. *Proceedings of the VLDB Endowment* 5, 454–465.
- Beerenwinkel, N., Beretta, S., Bonizzoni, P., Dondi, R., Pirola, Y., 2015. Covering pairs in directed acyclic graphs. *The Computer Journal* 58, 1673–1686.
- Blattner, F.R., Plunkett, G., Bloch, C.A., *et al.*, 1997. The complete genome sequence of *Escherichia coli* k-12. *Science* 277, 1453–1462.
- Ciriello, G., Cerami, E., Sander, C., Schultz, N., 2012. Mutual exclusivity analysis identifies oncogenic network modules. *Genome Research* 22, 398–406.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. *Introduction to Algorithms*, third ed. MIT Press.
- Dyer, M.D., Murali, T., Sobral, B.W., 2008. The landscape of human proteins interacting with viruses and other pathogens. *PLOS Pathog* 4, e32.
- Edwards, D.J., Holt, K.E., 2013. Beginner's guide to comparative bacterial genome analysis using next-generation sequence data. *Microbial Informatics and Experimentation* 3, 2.
- Feist, A.M., Herrgård, M.J., Thiele, I., Reed, J.L., Palsson, B., 2009. Reconstruction of biochemical networks in microorganisms. *Nature Reviews Microbiology* 7, 129–143.
- Fleischner, H., 1991. X. 1 algorithms for eulerian trails. Eulerian graphs and related topics: Part 1. *Annals of Discrete Mathematics* 50, 1–13.
- Fleury, M., 1883. Deux problemes de geometrie de situation. *Journal de Mathematiques Elementaires* 2.
- Fukagawa, D., Tamura, T., Takasu, A., Tomita, E., Akutsu, T., 2011. A clique based method for the edit distance between unordered trees and its application to analysis of glycan structures. *BMC Bioinformatics* 12, S13.
- Gnerre, S., MacCallum, I., Przybylski, D., *et al.*, 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences* 108, 1513–1518.
- Ideker, T., Sharan, R., 2008. Protein networks in disease. *Genome Research* 18, 644–652.
- Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M., 2010. Kegg for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research* 38, D355–D360.
- Loman, N.J., Quick, J., Simpson, J.T., 2015. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature Methods* 12, 733–735.
- Milenkovi, T., Memievi, V., Bonato, A., Pruli, N., 2011. Dominating biological 300 networks. *PLOS ONE* 6, 1–12.
- Nepusz, T., Yu, H., Paccanaro, A., 2012. Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods* 9, 471–472.
- Pertea, M., Pertea, G.M., Antonescu, C.M., *et al.*, 2015. Stringtie enables improved reconstruction of a transcriptome from rna-seq reads. *Nature Biotechnology* 33, 290–295.
- Piraveenan, M., Prokopenko, M., Zomaya, A., 2012. Assortative mixing in directed biological networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 9, 66–78.
- Rual, J.F., Venkatesan, K., Hao, T., *et al.*, 2005. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 37, 1173–1178.
- Tpfer, A., Marschall, T., Bull, R.A., *et al.*, 2014. Viral quasispecies assembly via maximal clique enumeration. *PLOS Computational Biology* 10, 1–10.
- Tsourakakis, C., Bonchi, F., Gionis, A., Gullo, F., Tsiarli, M., 2013. Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guaran tees. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM. pp. 104–112.
- Volinia, S., Galasso, M., Costinean, S., *et al.*, 2010. Reprogramming of mirna networks in cancer and leukemia. *Genome Research* 20, 589–599.
- Yu, H., Paccanaro, A., Trifonov, V., Gerstein, M., 2006. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics* 22, 823.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research* 18, 821–829.

Graph Isomorphism

Riccardo Dondi, University of Bergamo, Bergamo, Italy

Giancarlo Mauri and Italo Zoppis, University of Milan-Biocca, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Graphs (or networks) have played a relevant role in computational biology in the last few years. Several biological problems have been modeled using graphs, from protein interaction networks (Spirin and Mirny, 2003; Scott *et al.*, 2006) to the representation of the relations among orthologous and paralogous genes (Hellmuth *et al.*, 2014; Lafond *et al.*, 2016).

In this contribution we will focus on the main concepts related to graph isomorphism, graph traversal and graph/network measures. First, we introduce in Section Background/Fundamentals some concepts related to graph traversals (paths, walks, cycles, circuits), graph connection properties and partition (diameter, connected components and strongly connected components), and graph isomorphism. In Section Methodologies, we investigate two main measures of a graph/network: network topology measures (average path length, diameter, cluster coefficient, degree distribution) and centrality measures (degree centrality, closeness centrality, betweenness centrality, eigenvector centrality). We then introduce the methods for the detection of recurrent subgraphs (motifs) inside a network and the related subgraph isomorphism problem. We present in Section Illustrative Examples some examples of application of these measures and concepts in computational biology.

Background/Fundamentals

In what follows we denote by $G=(V, E)$ an undirected graph, where V is the set of vertices and E is the set of (undirected) edges, and by $D=(V, A)$ a directed graph (or digraph), where V is the set of vertices and A is the set of (directed) arcs. We represent an edge of $G=(V, E)$ between vertices $u, v \in V$ as $\{u, v\}$, while we denote a directed arc of $D=(V, A)$ between vertices $u, v \in V$ as (u, v) . Given a set S , we denote by $|S|$ the size or cardinality of S .

Given a vertex v of G , we denote by $N(v)$ the set of vertices adjacent to v , or the neighborhood of v . Formally:

$$N(v) = \{u : \{u, v\} \in E\}$$

For example in Fig. 1, $N(v_1) = \{v_2, v_3\}$

In a directed graph $D=(V, A)$, given a vertex $v \in V$, we denoted by $N_{in}(v)$ (by $N_{out}(v)$, respectively) the in-neighborhood (the out-neighborhood, respectively) of v , that is the set of vertices that have an incoming arc in v (an outgoing arc from v , respectively). Formally:

$$N_{in}(v) = \{u : (u, v) \in A\}$$

$$N_{out}(v) = \{u : (v, u) \in A\}$$

Consider Fig. 2, $N_{in}(v_3) = \{v_2, v_5\}$, while $N_{out}(v_3) = \{v_1\}$.

Let $V' \subseteq V$, we denote by $G[V']$ the subgraph induced by V' , that is $G[V'] = (V', E')$, where

$$E' = \{\{u, v\} : \{u, v\} \in E, u, v \in V'\}$$

For example, consider Fig. 1. The graph induced by $V' = \{v_1, v_2, v_3\}$ has edges

$$E' = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_2, v_3\}\}$$

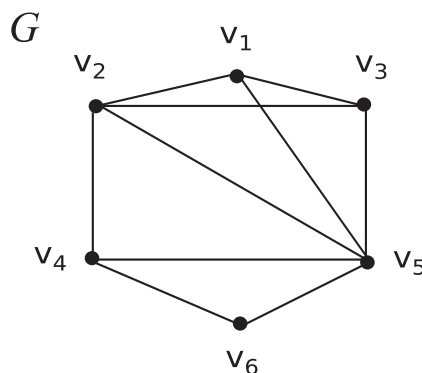


Fig. 1 An undirected graph $G=(V, E)$.

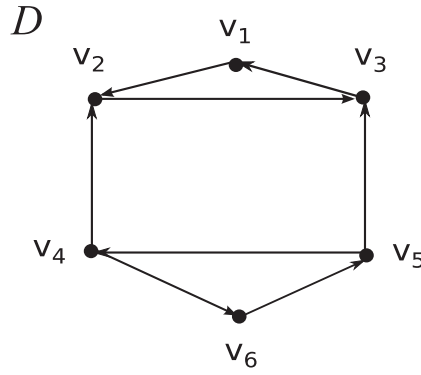


Fig. 2 A directed graph $D=(V, A)$.

Given a graph $G=(V, E)$ (a directed graph $D=(V, A)$, respectively), a *path* Π in G (in D , respectively) is a sequence of distinct vertices $v_{\Pi(0)}, v_{\Pi(1)}, \dots, v_{\Pi(h)}$, such that $\{v_{\Pi(j)}, v_{\Pi(j+1)}\} \in E$, for each j with $0 \leq j \leq h-1$ ($\{v_{\Pi(j)}, v_{\Pi(j+1)}\} \in A$, for each j with $0 \leq j \leq h-1$, respectively). The length of a path $\Pi=(v_{\Pi(0)}, v_{\Pi(1)}, \dots, v_{\Pi(h)})$ is h , that is the number of vertices that belong to the path minus one.

A *walk* in a graph $G=(V, E)$ (in a directed graph $D=(V, A)$, respectively) is a sequence Π of vertices $v_{\Pi(0)}, v_{\Pi(1)}, \dots, v_{\Pi(h)}$, such that $\{v_{\Pi(j)}, v_{\Pi(j+1)}\} \in E$, for each j with $0 \leq j \leq h-1$ ($\{v_{\Pi(j)}, v_{\Pi(j+1)}\} \in A$, for each j with $0 \leq j \leq h-1$, respectively). As for paths, the length of a walk Π' is the number of vertices in Π' that belong to the walk minus one.

Notice that the vertices in a path Π must be distinct, while vertices that belong to a walk are not necessarily distinct. Consider graph G in Fig. 1; $\Pi=v_1, v_2, v_4, v_6$ is a path between v_1 and v_6 of length 3 in G . $\Pi'=v_1, v_2, v_4, v_5, v_6, v_4$ is a walk in G from v_1 to v_4 of length 5, but is not a path, since vertex v_4 appears twice in Π' .

Notice that $\Pi=v_1, v_2, v_4, v_6$ is not a path in D of Fig. 2, since (v_1, v_4) and (v_4, v_6) are arcs of D , while (v_2, v_4) is not an arc of D (notice that (v_4, v_2) is an arc of D).

A walk $\Pi=v_{\Pi(0)}, v_{\Pi(1)}, \dots, v_{\Pi(h)}$ in a graph G (a digraph D , respectively) is called a *cycle* if each vertex in Π is distinct except for the first and the last vertex, that is $v_{\Pi(0)}=v_{\Pi(h)}$. A walk $\Pi=v_{\Pi(0)}, v_{\Pi(1)}, \dots, v_{\Pi(h)}$ is called a *circuit* if the first and the last vertex are identical, that is $v_{\Pi(0)}=v_{\Pi(h)}$. The length of a cycle and of a circuit are defined as for the length of a walk and of a circuit.

Similarly to the difference between path and walk, notice that in a cycle all the vertices except for the first and the last are distinct, while this may not be the case for a circuit. Consider graph G in Fig. 1, $C=v_1, v_2, v_3, v_1$ is a cycle (and also a circuit) in G having length 3. Consider $C'=v_1, v_2, v_4, v_5, v_2, v_3, v_1$; C' is a circuit in G having length 6, but is not a cycle, since vertex v_2 has two occurrences in C' .

Given two vertices u, v of a graph $G=(V, E)$ (of a directed graph $D=(V, A)$), the *distance* $d_G(u, v)$ between u and v in G (the distance $d_D(u, v)$ between u and v in D , respectively), is the length of the shortest path between u and v . Vertices v_1 and v_2 in graph G of Fig. 1 have distance 1 since they are adjacent; vertices v_1 and v_6 have distance 2, since $\Pi=v_1, v_5, v_6$ is the unique shortest path between v_1 and v_6 in G and Π has length 2. Notice that several shortest paths may exist between two vertices and that while d_G is symmetric, that is $d_G(u, v)=d_G(v, u)$, this is not the case for d_D , that is $d_D(u, v)$ may be different from $d_D(v, u)$. For example in the directed graph D of Fig. 2, $d_D(v_1, v_2)=1$, while $d_D(v_2, v_1)=2$. When there is no path from u to v , the distance usually has value $+\infty$.

A measure that has been widely used to study graph and network is the *diameter*. The diameter of an undirected graph $G=(V, E)$ (a directed graph D) is the maximum distance between any pair of vertices of G (of $D=(V, A)$, respectively). Formally,

$$\text{diam}(G) = \max_{u, v \in V} d_G(u, v)$$

$$\text{diam}(D) = \max_{u, v \in V} d_D(u, v)$$

Graph G in Fig. 1 has diameter 2, since $d_G(v_1, v_6)=2$ and the distance between any other pair of vertices in G is at most 2.

An undirected graph $G=(V, E)$ is called a *clique* (or a complete graph), when for each $u, v \in V$, $\{u, v\} \in E$, that is each pair of vertices in V is connected by an edge. A subgraph $G[V']$ of $G=(V, E)$, where $V' \subseteq V$, is called a clique of G when for each $u, v \in V'$, $\{u, v\} \in E$. When $G[V']$ is a clique, we say that V' induces a clique in G . The size of a clique $G[V']$ is $|V'|$.

A clique $G[V']$ of $G=(V, E)$ is called *maximal* if there is no vertex $u \in V/V'$ such that $\{u, v\} \in E$ for each vertex $v \in V'$. Intuitively, a clique $G[V']$ is maximal if it cannot be extended by adding some vertex to V' , that is there is no vertex of $v \in V/V'$ such that $\{v, u\} \in E$ for each $u \in V'$. A *maximum clique* $G[V']$ of $G=(V, E)$, with $V' \subseteq V$, is a clique of G having maximum size, that is there is no set W , with $W \subseteq V$ and $W \neq V'$, such that $G[W]$ is a clique and $|W| > |V'|$. Notice that a maximum is also a maximal clique, while a maximal clique may not be a maximum one. A concept related to a clique is that of *independent set*. An undirected graph $G=(V, E)$ is an independent set if $E=\emptyset$. A subgraph $G[V']$ of $G=(V, E)$, with $V' \subseteq V$, is an independent set if $\{u, v\} \notin E$, for each $u, v \in V'$. Notice that $G=(V, E)$ is an independent set if and only if the complement of G , $\bar{G}=(V, \bar{E})$, is a clique, where $\bar{E}=\{\{u, v\} : \{u, v\} \notin E\}$. As for cliques, we can define maximal and maximum independent set.

Consider a graph G in Fig. 1. The subgraphs induced by $\{v_1, v_2, v_3\}$, $\{v_1, v_2, v_3, v_5\}$, $\{v_4, v_5, v_6\}$ are all cliques of G . However notice that $\{v_1, v_2, v_3\}$ does not induce a maximal clique, since $\{v_1, v_2, v_3\} \subset \{v_1, v_2, v_3, v_5\}$. $G[\{v_4, v_5, v_6\}]$ is a maximal clique, but not a maximum clique, since it contains three vertices, while $G[\{v_1, v_2, v_3, v_5\}]$ contains four vertices; $G[\{v_1, v_2, v_3, v_5\}]$ is a maximum clique in G , as there is no clique of size 5 in G .

The sets $\{v_1, v_6\}$ and $\{v_3, v_4\}$ are two independent sets in G , since $\{v_1, v_6\} \cap E = \emptyset$ and $\{v_3, v_4\} \cap E = \emptyset$. They are both maximum independent sets of G .

A fundamental concept in graph theory is that of *connected component* of an undirected graph.

Definition 2.1: Given an undirected graph $G=(V, E)$, a *connected component* of G is a graph $G[V']$ induced by a maximal set $V' \subseteq V$ such that for each $v_1, v_2 \in V'$, there exists a path in G' from v_1 to v_2 .

Consider the graph G in Fig. 3. G contains two connected components: $G[\{v_1, v_2, v_3, v_4, v_5, v_6\}]$ and $G[\{v_7, v_8, v_9\}]$.

In directed graphs, the orientation of arcs leads to a different concept, that of *strongly connected component*.

Definition 2.2: Given a directed graph $D=(V, A)$, a *strongly connected component* is a graph $D[V']$ induced by a maximal set $V' \subseteq V$ such that for each pair (v_1, v_2) of vertices in $D[V']$, there exists a directed path $\Pi(v_1, v_2)$ in $D[V']$.

Consider the graph D in Fig. 4. D contains two strongly connected components: the strongly connected component induced by $\{v_1, v_2, v_3\}$ and the strongly connected component induced by $\{v_4, v_5, v_6\}$. Notice that if we ignore arc orientation, D is connected.

An interesting properties of directed graphs is that a directed graph $D=(V, A)$ can be decomposed in a second directed graph called *component directed graph* $D_{SCC}(D)=(V_{SCC}, A_{SCC})$, where the vertices in V_{SCC} correspond to strongly connected components of D and there is an arc (v_a, v_b) in A_{SCC} if and only if there is an arc in D from a vertex in the strong connected component associated with v_a to a vertex in the strong connected component associated with v_b . Notice that D_{SCC} is a directed acyclic graph.

Consider the directed graph $D=(V, A)$ in Fig. 4. The directed graph contains three directed components induced by vertices $\{v_1, v_2, v_3\}$, $\{v_4, v_5, v_6\}$ and $\{v_7\}$, respectively. The corresponding component directed graph D_{SCC} contains three vertices, corresponding to the three strongly connected components of D , and two arcs.

A problem widely investigated in computer science and graph theory, with application in the analysis of biological networks, is *graph isomorphism*. Graph isomorphism defines a bijection between the vertices of two given graphs (a similar definition can be given for isomorphism between directed graphs).

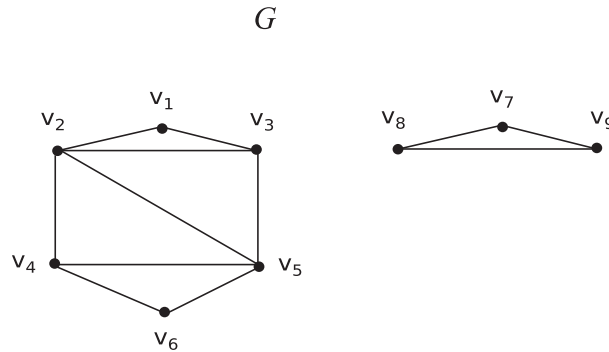


Fig. 3 A graph $G=(V, E)$ with two connected components.

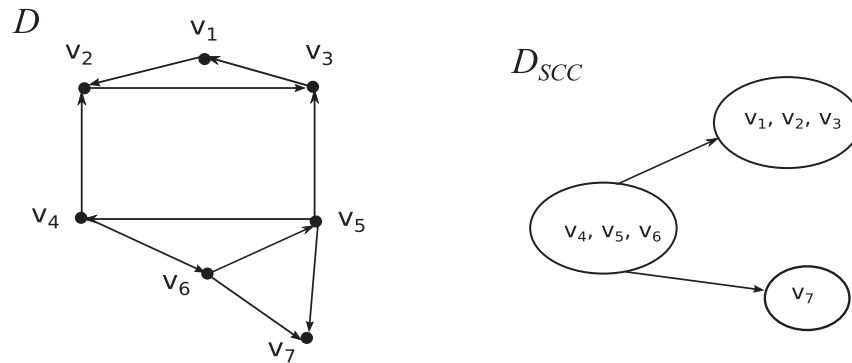


Fig. 4 A directed graph $D=(V, A)$ with three connected components (left) and the corresponding component directed graph D_{SCC} (right). Notice that the vertices of D_{SCC} are named with the vertices in the corresponding strongly connected components of D .

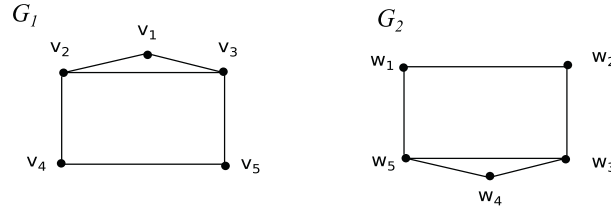


Fig. 5 Two isomorphic graphs G_1 and G_2 .

Definition 2.3: Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a graph isomorphism is a bijective function $f: V_1 \rightarrow V_2$, such that $\{u, v\} \in E_1$ if and only if $\{f(u), f(v)\} \in E_2$.

Consider the two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ represented in Fig. 5. The two graphs are isomorphic, since it is possible to define a bijection $f: V_1 \rightarrow V_2$ defined as follows: $f(v_1) = w_4$, $f(v_2) = w_3$, $f(v_3) = w_5$, $f(v_4) = w_2$, $f(v_5) = w_1$. Consider for example $\{v_1, v_2\} \in E_1$; it holds that $\{f(v_1), f(v_2)\} = \{w_4, w_3\} \in E_2$. It can be checked by direct inspection that indeed f defines an isomorphism between G_1 and G_2 .

Methodologies

In this section, we consider three main topics related to the analysis of a graph: network topology measures (applied to study to the global structure of a graph/network), centrality measures (applied to study the relevance of a specific vertex inside a network) and detection of network motifs (identification of relevant part of a graph/network).

Network Topology Measures

Several measures have been introduced to understand topological properties of a network. Here we introduce three well-known and applied measures: average path length, cluster coefficient, degree distribution.

The *average path length* of a graph has been introduced to measure the average distance between vertices of the graph, to characterize, for example, the information flow inside a network. Given an undirected graph $G = (V, E)$, the *average path length* $av(G)$ is defined as

$$av(G) = \frac{1}{|V|(|V| - 1)} \sum_{u, v \in V, u \neq v} d_G(u, v)$$

where we assume that $d_G(u, v) = 0$ when there is no path from u to v . A similar definition can be given for directed graphs.

Consider graph G in Fig. 1. Then

$$av(G) = \frac{40}{30}$$

The value of the *average path length* of a graph is related to the *small-world phenomenon*. In several analyzed graphs, like the World Wide Web, the reaction graph of Escherichia coli or the network representing the neural structure of Caenorhabditis elegans, the average path length is bounded with respect to the size of the graph (Albert and Barabási, 2002). Given a graph having n vertices, a graph that follows the small-world phenomenon has an average path length of value $O(\log n)$ (Watts and Strogatz, 1998).

Cluster coefficient has been introduced to measure the degree of transitivity in graph edges. Two measures have been introduced in this context: *global cluster coefficient* and *local cluster coefficient*. Given a graph $G = (V, E)$, define a triplet in G as a connected graph induced by three vertices. Denote by $\#_{Tr}$ the number of triplets in G , and by $\#_{CompTr}$ the number of triplets in G that are cliques (of size 3). The *global cluster coefficient* C_{Global} is defined as:

$$C_{Global} = \frac{3\#_{CompTr}}{\#_{Tr}}$$

Notice that $\#_{CompTr}$ is multiplied by three, as each clique of size three induces three triplets.

The *local cluster coefficient* $C_{local}(v)$ of a vertex $v \in V$ is defined as:

$$C_{local} = \frac{2|\{\{u, w\} : u, v \in N(v)\}|}{|N(v)|(|N(v)| - 1)}$$

The global cluster coefficient has been introduced to measure the degree of transitivity of the whole given graph, while the local degree clustering measure the degree of transitivity only in the neighborhood of a specific vertex v .

Degree distribution measures how many vertices of a graph have a given degree $k \geq 0$. In directed graphs, this measures is usually applied to the in-degree of the graph. Degree distribution has been introduced to understand the structure of networks and it is used for example to detect whether there are vertices that are particularly relevant for a network, like hubs. Hubs are vertices of a network that have a huge degree, when compared to the size of the network. In random-graph models degree distribution follows

a Gaussian law, and no hub is present in networks built according to these models. In scale-free network models degree distribution follows a power-law, and hubs are present in networks built according to these models. Several biological networks, for example metabolic networks, have been explained using the scale-free model (Jeong *et al.*, 2000).

Centrality Measures

Centrality measures have been introduced to measure the relevance of a specific vertex of a graph. Several measures have been introduced, based on different properties. We review some of the most relevant centrality measures:

- Degree centrality
- Closeness centrality
- Betweenness centrality
- Eigenvector centrality

Degree centrality of a vertex v is equal to the degree of v . The degree centrality measures the capability of establishing relationships of a given vertex. Consider the graph G in Fig. 1; then the vertex having maximum degree centrality is v_5 , since it has degree centrality 5, while v_2 has degree centrality 4 and v_1, v_3 have degree centrality 3.

Closeness centrality of a vertex v , in a connected graph, is defined as the average distance (length of a shortest path) between v and all other vertices of the graph. Central vertices are those closer to other vertices of the network. Consider the graph G in Fig. 1; then the vertex having maximum closeness centrality is v_5 , since v_5 has closeness centrality 5; v_5 is indeed adjacent to any other vertex of G . Vertex v_2 has closeness centrality 6, since v_2 has distance 1 from v_1, v_3, v_4, v_5 and has distance 2 from v_6 .

Betweenness centrality of a node v is defined as the number of shortest paths between other vertices than v that contain v . Betweenness centrality is used to measure the relevance of a vertex in the connecting vertices of a graph. Consider the graph G in Fig. 1. Then the vertex having maximum betweenness centrality is v_5 , since v_5 has degree centrality 4: a shortest path between v_6 and v_1, v_6 and v_2, v_6 and v_3 passes through v_5 ; a shortest path between v_3 and v_4 passes through v_5 . The betweenness centrality of v_1, v_3 and v_6 is 0, as no shortest path passes through any of these vertices. The betweenness centrality of v_2 is 2, since a shortest path between v_1 (v_3 , respectively) and v_4 passes through v_2 . Finally, the betweenness centrality of v_4 is 1, since a shortest path between v_2 and v_6 passes through v_4 .

Eigenvector centrality measures the influence of a vertex in a graph. Eigenvector centrality of a vertex v is proportional to eigenvector centrality of the vertices in $N(v)$. Formally

$$Mc = \lambda c$$

where M is adjacency matrix of graph G , c is an eigenvector of matrix M and λ is a constant. Informally, the eigenvector centrality value of a vertex v depends on the eigenvector centrality values of vertices in $N(v)$, while the eigenvector centrality value of v influences that of vertices in $N(v)$. A variant of eigenvector centrality is page rank, a centrality measure well-known due to its application in Web search (for more details on page rank see Langville and Meyer (2003)).

Motif Detection

The identification of motifs inside a network is a fundamental task in several contexts, from biological to social networks. A *motif* is a graph (also called a pattern). An *occurrence* of a motif G_1 in a graph $G=(V, E)$ is a subgraph $G[V']$, for some $V' \subseteq V$, which is isomorphic to G_1 . Notice that finding an occurrence of a motif in a graph is the *subgraph isomorphism problem*. Usually, methods look for motifs that have a statistically significant number of occurrences in a graph. Identifying network motifs is a relevant problem, as the presence of a motif is considered to be related to functional properties of biological systems.

Another approach similar to graph motif is based on the identification of highly cohesive subgraphs. The most known problem is that computing clique of maximum size. However, other approaches have been recently proposed based on the relaxation of some constraints of the clique definition, leading to the general concept of *clique-relaxation* (Komusiewicz, 2016). Clique-relaxation includes degree relaxation, connectivity relaxation, distance relaxation. For example, distance relaxation leads to the definition of s -club: an s -club, for $s \geq 1$, is a set of vertices having at distance at most s . When $s=1$, a 1-club is a clique. However, for $s > 1$, an s -club is not necessarily a clique and the problems of computing a maximum size s -club exhibits different properties from that of computing a maximum size clique.

Analysis and Assessment

In this section we discuss the main complexity results related to graph isomorphism, subgraph isomorphism, maximum clique computation and variants thereof. Observe that a shortest path between two vertices can be computed in polynomial time.

Graph Isomorphism

The graph isomorphism problem is a fundamental problem in computer science with many applications in several fields. The computational complexity of graph isomorphism is a long standing open problem, as it is not known to be in P or not.

The NP-completeness of graph isomorphism would imply the collapsing of the polynomial hierarchy to the second level (Goldreich *et al.*, 1986). A quasipolynomial time algorithm of time complexity $\exp((\log|V|)^{O(1)})$ for the graph isomorphism problem has recently been given in Babai (2016).

Motif Detection

The subgraph isomorphism problem, that is computing whether a graph G_2 is a subgraph of a graph G_1 is an NP-complete problem (Cook, 1971). Computing a maximum clique is a well-known NP-hard problem (Karp, 1972). The approximation complexity of the maximum clique has been investigated, leading to prove that the problem is not approximable within factor $O(|V|^{1-\epsilon})$, assuming $P \neq NP$ (Zuckerman, 2007). As for the parameterized complexity, maximum clique is $W[1]$ -complete (Downey and Fellows, 1995a,b), hence it is unlikely that it admits algorithms of time complexity $O(f(k)|V|)$ for a given graph $G=(V, E)$, where k denotes the size of the clique.

The problem of computing a maximum size s -club of maximum size has been extensively studied, and it is known to be NP-hard (as Max-clique), but fixed-parameter tractable (unlike Max-clique) (Schäfer *et al.*, 2012; Chang *et al.*, 2013).

Illustrative Examples

There several problems related to the computation of paths in a graph, with application in computational biology.

A notable example is the detection of signaling pathways in protein interaction networks (Scott *et al.*, 2006). Protein interactions are represented with a graph, whose vertices represent proteins and whose edges represent interaction. A signal pathway is a chain of interacting proteins. An approach to compute signaling paths is based on the computation of a path of a given length k and exploits the color-coding technique (Alon *et al.*, 1995) for the design of a fixed-parameter tractable algorithm for this problem.

A second example can be found in the representation of the structure of a gene with respect to alternative splicing events. Graph approaches have been proposed to represent the structure of a gene with an acyclic graph, called *isoform graph* (Lacroix *et al.*, 2008) or splicing graph (Beretta *et al.*, 2014). In this directed graph, vertices represent exons (or substrings of exons) and a directed arc represents the fact that two exons are consecutive in a transcript. Then certain paths in these directed graphs are sought to identify transcript.

Motif detection is a fundamental problem in biological network analysis. Motifs related to regulatory functions have been found for example in *Escherichia coli* (Alon, 2007). Clique detection has also been applied in several network analysis problems. A notable example is in the context of protein-protein interaction network, where maximal cliques are identified to discover molecular modules (Spirin and Mirny, 2003).

A concept related to motifs in vertex colored-graphs has been introduced to study properties of metabolic networks (Lacroix *et al.*, 2006; Dondi *et al.*, 2013,2011). In this model a vertex-colored graph $G=(V, E)$ and a multiset M of colors are given. The problem, called colored-motif, asks for a colorful motif in G , that is a subgraph $G[V']$, with $V' \subseteq V$, whose vertices have colors that matches M . The problem is NP-complete, even in restricted cases (for example when the input graph is a tree of bounded degree), but is fixed-parameter tractable (Fellows *et al.*, 2011; Björklund *et al.*, 2016; Pinter *et al.*, 2016).

Closing Remarks

In this contribution we have reviewed some of the most relevant concepts and properties related to graph isomorphism and graph components. We have given definition and examples of the main concepts related to graph traversal and connectivity in a graph. We have considered problems related to subgraph isomorphism and motif detection (maximal clique and maximum clique, clique relaxations). Moreover, we have reviewed two measures that have been widely applied to study graphs: network topology measures (average path length, diameter, cluster coefficient, degree distribution), centralization measures (degree centrality, closeness centrality, betweenness centrality, eigenvector centrality). Finally, we have given some examples of how these concepts and methodologies are applied in computational biology.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Algorithms for Structure Comparison and Analysis: Docking. Natural Language Processing Approaches in Bioinformatics

References

- Albert, R., Barabási, A.L., 2002. Statistical mechanics of complex networks. *Review of Modern Physics* 74, 47–97. doi:10.1103/RevModPhys.74.47.
 Alon, N., Yuster, R., Zwick, U., 1995. Color-coding. *Journal of the ACM* 42 (4), 844–856.

- Alon, U., 2007. Network motifs: Theory and experimental approaches. *Nature Reviews Genetics* 8, 450–461. doi:10.1038/nrg2102.
- Babai, L., 2016. Graph isomorphism in quasipolynomial time (extended abstract). In: Wicks, D., Mansour, Y. (Eds.), *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, Cambridge, MA, USA, June 18–21, 2016, ACM. pp. 684–697. Available at: <http://doi.acm.org/10.1145/2897518.2897542>.
- Beretta, S., Bonizzoni, P., Della Vedova, G., Pirola, Y., Rizzi, R., 2014. Modeling alternative splicing variants from rna-seq data with isoform graphs. *Journal of Computational Biology* 21, 16–40. doi:10.1089/cmb.2013.0112.
- Björklund, A., Kaski, P., Kowalik, L., 2016. Constrained multilinear detection and generalized graph motifs. *Algorithmica* 74, 947–967. doi:10.1007/s00453-015-9981-1.
- Chang, M., Hung, L., Lin, C., Su, P., 2013. Finding large k-clubs in undirected graphs. *Computing* 95, 739–758. doi:10.1007/s00607-012-0263-3.
- Cook, S.A., 1971. The complexity of theorem-proving procedures. In: *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, May 3–5, 1971, pp. 151–158. Ohio, United States: Shaker Heights doi:10.1145/800157.805047.
- Dondi, R., Fertin, G., Viale, S., 2011. Complexity issues in vertex-colored graph pattern matching. *Journal of Discrete Algorithms* 9, 82–99. doi:10.1016/j.jda.2010.09.002.
- Dondi, R., Fertin, G., Viale, S., 2013. Finding approximate and constrained motifs in graphs. *Theoretical Computer Science* 483, 10–21. doi:10.1016/j.tcs.2012.08.023.
- Downey, R.G., Fellows, M.R., 1995a. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal of Computing* 24, 873–921. doi:10.1137/S0097539792228228.
- Downey, R.G., Fellows, M.R., 1995b. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science* 141, 109–131. doi:10.1016/0304-3975(94)00097-3.
- Fellows, M.R., Fertin, G., Hermelin, D., Viale, S., 2011. Upper and lower bounds for finding connected motifs in vertex-colored graphs. *J. Comput. Syst. Sci.* 77, 799–811. doi:10.1016/j.jcss.2010.07.003.
- Goldreich, O., Micali, S., Wigderson, A., 1986. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In: 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27–29 October 1986, IEEE Computer Society. pp. 174–187. Available at: <https://doi.org/10.1109/SFCS.1986.47>.
- Hellmuth, M., Wieseke, N., Lechner, M., et al., 2014. Phylogenomics with paralogs. *PNAS*.
- Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabási, A.L., 2000. The large-scale organization of metabolic networks. *Nature* 407, 651–654. doi:10.1038/35036627.
- Karp, R.M., 1972. Reducibility among combinatorial problems. In: *Proceedings of a symposium on the Complexity of Computer Computations*, held on March 20–22, 1972, at the IBM Thomas J. Watson Research Center, York-Town Heights, New York, pp. 85–103.
- Komusiewicz, C., 2016. Multivariate algorithmics for finding cohesive subnetworks. *Algorithms* 9, 21. doi:10.3390/a9010021.
- Lacroix, V., Fernandes, C.G., Sagot, M., 2006. Motif search in graphs: Application to metabolic networks. *IEEE/ACM Trans. Comput. Biology Bioinform* 3, 360–368. doi:10.1109/TCBB.2006.55.
- Lacroix, V., Sammeth, M., Guigo, R., Bergeron, A., 2008. Exact transcriptome reconstruction from short sequence reads. In: Crandall, K.A., Lagergren, J. (Eds.), *Algorithms in Bioinformatics, Proceedings of the 8th International Workshop, WABI 2008, Karlsruhe, Germany, September 15–19, 2008*, Springer. pp. 50–63. DOI:10.1007/978-3-540-87361-7_5.
- Lafond, M., Dondi, R., El-Mabrouk, N., 2016. The link between orthology relations and gene trees: A correction perspective. *Algorithms for Molecular Biology* 11, 4. doi:10.1186/s13015-016-0067-7.
- Langville, A.N., Meyer, C.D., 2003. Survey: Deeper inside pagerank. *Internet Mathematics* 1, 335–380. doi:10.1080/15427951.2004.10129091.
- Pinter, R.Y., Shachnai, H., Zehavi, M., 2016. Deterministic parameterized algorithms for the graph motif problem. *Discrete Applied Mathematics* 213, 162–178. doi:10.1016/j.dam.2016.04.026.
- Schäfer, A., Komusiewicz, C., Moser, H., Niedermeier, R., 2012. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters* 6, 883–891. doi:10.1007/s11590-011-0311-5.
- Scott, J., Ideker, T., Karp, R.M., Sharan, R., 2006. Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology* 13, 133–144. doi:10.1089/cmb.2006.13.133.
- Spirin, V., Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America* 100, 12123–12128. doi:10.1073/pnas.2032324100.
- Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. *Nature* 440–442. doi:10.1038/30918.
- Zuckerman, D., 2007. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing* 3, 103–128. doi:10.4086/toc.2007.v003a006.

Further Reading

- Bang-Jensen, J., Gutin, G., 2002. *Digraphs – Theory, Algorithms and Applications*. Springer.
- Barabási, A.L., Po'sfai, M., 2016. *Network Science*. Cambridge University Press.
- Diestel, R., 2005. *Graph Theory (Graduate Texts in Mathematics)*. Springer.
- Downey, R., Fellows, M., 2013. *Fundamentals of Parameterized Complexity*. Springer.
- Niedermeier, R., 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.

Relevant Websites

- <http://barabasi.com/andhttp://barabasi.com/networksciencebook/>
Barabási Network Science Book.
- <http://www.weizmann.ac.il/mcb/UriAlon/download/network-motif-software>
mfinder, a network motif detection tool.

Graph Algorithms

Riccardo Dondi, University of Bergamo, Bergamo, Italy

Giancarlo Mauri and Italo Zoppis, University of Milan-Biocca, Milan, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The design and analysis of algorithms on graphs is a relevant and active area in computer science and discrete mathematics, with several application in different fields, from transport network to social network analysis and computational biology. Since graphs are structure widely applied to model problems in different contexts, the research has focused on the design of efficient algorithms for some natural problems. First, we will consider the problem of traversing and exploring a graph. Two well-known approaches for traversing a graph are depth-first search and breadth-first search, and we will consider different properties of these two search algorithms.

A natural problem when dealing with graphs is the quest for shortest paths between a source vertex and a target vertex. The computation of a shortest path between two vertices of a graph has several applications, from transportation and networking to specific problems in computational biology. We will review three well-known algorithms for computing shortest paths: Dijkstra's algorithm, Bellman–Ford algorithm and Floyd–Warshall algorithm.

A third well-known problem that has been considered in this context is the computation of maximum flow that can be transferred from a source vertex to a target vertex of a graph. We will review the Ford–Fulkerson algorithm and the relation between maximum flow and minimum cut of a graph. The computation of maximum flow and minimum cut have several applications from scheduling to phylogenetics.

Another graph problem that has application in several fields, is the computation of a minimum spanning tree of a graph. The problem has been considered in different contexts, from example networking and clustering. Given a graph, the minimum spanning tree problem ask for a tree that covers all the vertices of a given graph. We will review two well-known algorithms to compute a minimum spanning tree: the Kruskal's algorithm and the Prim's algorithm.

Finally, we will consider a graph problem that has been extensively studied in the last fifty years, the traveling salesman person, and we illustrate an algorithmic technique, known as nearest neighbour, that has been applied to design a heuristic for the traveling salesman person.

First, we introduce in Section Background/Fundamentals some definitions that will be useful in the rest of the paper. Then, in Section Methodologies, we will describe the graph algorithms we consider and in Section Analysis and Assessment we will discuss their time complexity. Finally, in Section Illustrative Examples, we will present some biological applications of the graph algorithms described in Section Methodologies.

Background/Fundamentals

In what follows, we will consider both weighted and unweighted graphs. We will focus mainly on undirected graphs, but most of the algorithms we consider can be extended to directed graphs. Given a graph $G=(V, E)$, a *path* Π in G is a sequence of distinct vertices $s=v_1, v_2, \dots, v_h=t$, having a source vertex s and a target vertex t .

Given a graph $G=(V, E)$, a *spanning tree* of G is a tree $T=(V, E')$ such that $E' \subseteq E$. We recall that a tree is a graph that induces no cycle. A *minimum spanning tree* $T'=(V, E')$ of a weighted graph G is a spanning tree of G that has minimum weight, that is

$$\sum_{\{u,v\} \in E'} w(\{u,v\})$$

is minimum among the spanning tree of G . Here $w: E \rightarrow \mathbb{R}^+ \cup \{0\}$ is the weight function, that associates a non negative value with each edge of G .

A path Π in $G=(V, E)$ is called *hamiltonian* if each vertex of V belongs to Π , that is each vertex of V is visited exactly once by Π . This definition can be extended to cycles, thus leading to the definition of hamiltonian cycles. The path $\Pi = v_1, v_3, v_6, v_4, v_5, v_2$ in the graph G of Fig. 1(a) is a hamiltonian path; the cycle $C = v_1, v_3, v_6, v_4, v_5, v_2, v_1$ is a hamiltonian cycle in G .

Methodologies

In this section we review some of the most relevant and studied graph algorithms. We start with depth-first search and breadth-first search traversal of a graph.

Breadth-First Search and Depth-First Search

Breadth-First Search (BFS) and Depth-First Search (DFS) are two basic algorithms for traversing a graph. In both algorithms, we assume that we are given an unweighted and undirected graph $G=(V, E)$, although the algorithm can be generalized to weighted

and directed graphs. Moreover, we consider a vertex $s \in V$, called the *source*, and we want to traverse every edge of G . Vertices of the graph can be in two possible states during the execution of breadth-first search and depth-first search: *undiscovered*, before the visit finds the vertex, *discovered*, when the algorithm has visited the vertex but some of the edges incident in it have not been considered, and *explored* once the vertex and each edge incident in it have been visited. Notice that when the two algorithms start, each vertex of G is unexplored.

We start presenting breadth-first search. Breadth-first search starts from vertex s and marks s as discovered, and visit G by *level*: level L_0 contains s , level L_1 contains the vertices in $N(s)$, and iteratively at step i breadth-first search defines L_i as the undiscovered vertices of $N(L_{i-1})$.

We present an implementation of BFS that uses a queue L to store the vertices to be visited.

Algorithm 1: Breadth-First Search

Data: G, s

Result: a BFS G

```

1 Mark  $s$  as Discovered;
2  $L := \text{Enqueue}(s)$ ;
3 /*  $L$  is a queue that contains the vertices discovered that have to be
   explored */;
4 while  $L \neq \emptyset$  do
5    $v := \text{Dequeue}(L)$ ;
6   foreach  $u \in N(v)$  do
7      $\text{Enqueue}(L, u)$ ;
8     if  $u$  is marked as undiscovered then
9       Mark  $u$  as discovered;
10  Mark  $v$  as explored;
```

Consider the example of Fig. 1. The BFS starts from vertex v_1 (Fig. 1(a)); vertices v_2 and v_3 , which are adjacent to v_1 , are enqueued in L (we assume v_2 is the first enqueued vertex, v_3 the second); v_1 is marked as explored, v_2 and v_3 as discovered. In Fig. 1(b) vertex v_2 (the first vertex of the queue) is explored, and vertex v_4 and v_5 are enqueued and marked as discovered (we assume v_5 is the first enqueued vertex, v_4 the second). In Fig. 1(c) vertex v_3 (the first vertex of the queue) is explored and the vertices in $N(v_3)$, are enqueued in L and marked as discovered. In Fig. 1(d)–(f) vertices v_5 , v_4 and v_6 , respectively are explored.

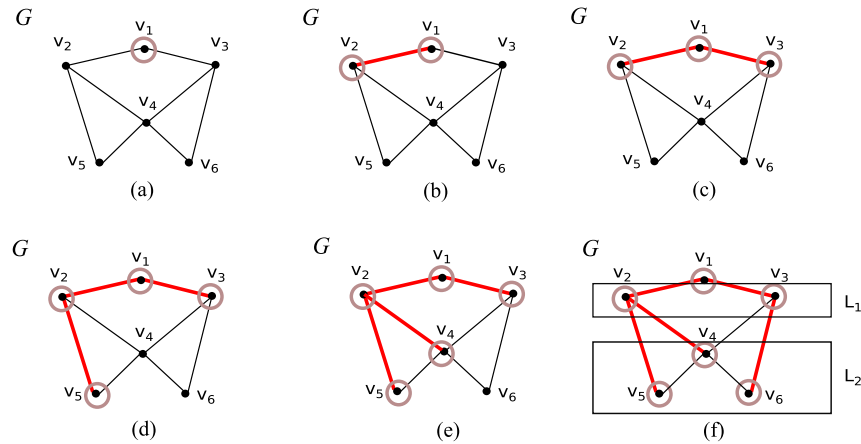


Fig. 1 An example of BFS. The vertices are discovered and explored starting from vertex v_1 . The discovered vertex at each step is circled, and the edges selected to discover vertices are highlighted in bold red. Notice that rectangles group the vertices at level L_1 (v_2 and v_3) and at level L_2 (v_4, v_5, v_6).

A relevant properties of BFS is that the vertices at level L_i , with $0 \leq i \leq V$, are those vertices at distance i in G from the source vertex s .

Depth-first search starts from vertex s , and explores the graph following branches, until it is possible to find unexplored vertices, thus leading to a different strategy. A recursive implementation for DFS that visit the connected component containing vertex s is given as follows:

Algorithm 2: Depth-First Search

Data: G, s

Result: a DFS G

```

1 Mark  $s$  as discovered;
2 foreach  $u \in N(v)$  do
3   if  $u$  is unexplored then
4     DFS( $G, u$ )
5  $s$  is marked as explored;
```

An iterative implementation of DFS can be given, using a stack S data structure.

Consider the example of Fig. 2. The DFS starts, as the DFS, from vertex v_1 (Fig. 2(a)); then vertices v_2 and v_3 , which are adjacent to v_1 , are considered (we assume that v_2 is considered first). In Fig. 2(b) vertex v_2 is marked as discovered and vertex v_4 and v_5 are visited (we assume that v_4 is considered first). In Fig. 2(c) vertex v_4 is marked as discovered and vertices in $N(v_4)$ are considered (we assume that v_5 is considered first). In Fig. 2(d) vertex v_5 is marked first as discovered and then as explored, since all vertex in $N(v_5)$ are already marked as discovered. In Step (e), vertex v_6 is discovered. In Step (f) vertex v_3 is discovered and Vertices v_3, v_6, v_4, v_2 and v_1 are finally marked as explored. Notice that the order in which vertices of G are discovered and explored by BFS and DFS is different. For example v_3 is the third vertex discovered by BFS and the last vertex discovered by DFS.

Breadth-first search and depth-first search have some relevant properties that we will illustrate next. First, both algorithms induce a tree (denoted as T_b for breadth-first search and as T_d for depth-first search). The vertices of T_b and T_d are those vertices of the connected component of G including the source vertex s , and the edges are those selected when marking a vertex as discovered. Consider Figs. 1 and 2; the edges of G that induce T_b and T_d , respectively, are highlighted in bold red.

Moreover, the following property for BFS holds.

Lemma 1: Let $G=(V, E)$ be a graph and let be T_b be a tree returned by the breadth-first search of G . Then, given an edge $\{u, v\}$ of G not included in T_b , it follows that u and v belong to levels L_i and L_j , with $i, j \geq 1$ such that $|i - j| \leq 1$.

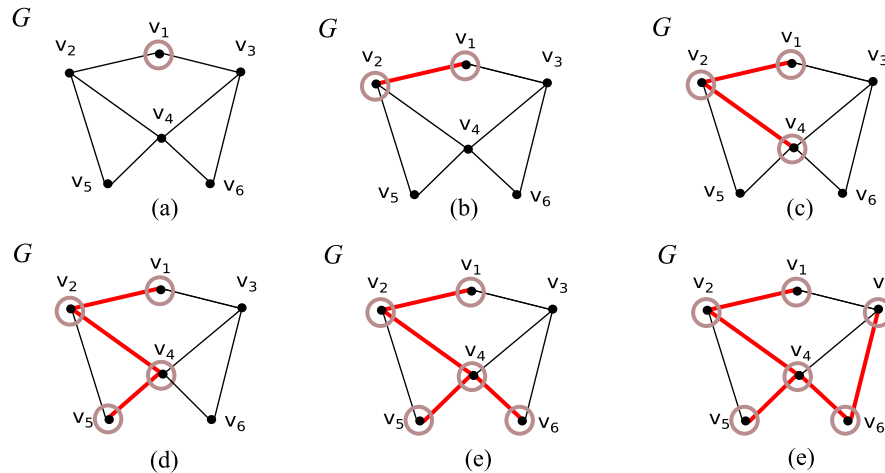


Fig. 2 An example of DFS. The vertices are visited starting from vertex v_1 . The discovered vertex at each step is circled, and the edges selected to discover vertices are highlighted in red.

Moreover, the path in T_b from s to a vertex v is a shortest path from s to v in G .
The following property for DFS holds.

Lemma 2: Let $G=(V, E)$ be a graph and let be T_d be a tree returned by the depth-first search of G . Then, given an edge $\{u, v\}$ of G not included in T_d , it holds that u and v are one the ancestor of the other in T_d .

Shortest Path Algorithms

The computation of shortest paths is a fundamental problem in graph theory, with several applications in transports and in several other fields. Here we present three algorithms to compute shortest paths: Dijkstra's algorithm, Bellman-Ford algorithm and Floyd-Warshall algorithm. We assume that s is the source vertex and t is the target vertex.

Dijkstra's algorithm

Dijkstra's algorithm (Dijkstra, 1959) computes the shortest path between a source vertex s and any other vertex of the graph in a weighted graph $G=(V, E)$, where each edge in E has a non negative weight.

Dijkstra's algorithm assigns a label l to the vertices of the graph, in order to compute a shortest path from s to these vertices. The label $l(v)$ associated with a vertex $v \in V$ contains the value of a path from s to v and can be of two kinds: temporary or definitive. When a label is marked as definitive, then the algorithm has found the length of a shortest path from s to v . Temporary labels are updated at each iteration of the algorithm until all the labels of the graphs are marked as definitive.

In the first step, the label of s is initialized to 0. Any other label is marked as temporary and as value $+\infty$. The algorithm iterative step considers the vertex v having a label with minimum value, and it marks the label of v as definitive. For each vertex $u \in N(v)$ with a temporary label, it is computed the value $l(v) + w(\{v, u\})$; if $l(u) \geq l(v) + w(\{v, u\})$, then $l(u)$ is updated to $l(v) + w(\{v, u\})$, else $l(u)$ is not modified.

Algorithm 3: Dijkstra's Shortest Path

Data: G, s

Result: shortest paths between s and any other vertex of G

```

1 foreach  $u \in V$  do
2    $l(u) := +\infty$ ;
3  $l(s) := 0$ ;
4  $l(s)$  is marked as definitive;
5  $S = \{s\}$ ;
6 /*  $S$  contains the vertices whose label is marked as definitive */;
7 foreach  $v \in N(s)$  do
8    $l(v) := w(\{s, v\})$ ;
9 while  $(V \setminus S) \neq \emptyset$  do
10  Consider  $v \in (V \setminus S)$  with minimum  $l(v)$ ;
11  foreach  $u \in N(v)$  do
12     $l(u) := \min(l(u), l(v) + w(\{u, v\}))$ ;
13     $l(v)$  is marked as definitive;
14     $S := S \cup \{v\}$ ;
```

Bellman-Ford algorithm

The Bellman Ford algorithm (Bellman, 1958; Floyd and Rivest, 1975) computes a shortest path from s to any other vertex of the graph $G=(V, E)$, also in the case that the weighted graph contains negative edge weights. The presence of negative cycles (cycles whose edge weights sum to a negative value) is also detected by Bellman-Ford algorithm. In this case, by traversing such a cycle, we decrease the total weight of the path arbitrarily, hence the algorithm returns the indication that no solution exists, otherwise it returns the values of a shortest path between s and a vertex v .

Bellman-Ford algorithm applies a strategy similar to that of Dijkstra's algorithm, that is it introduces a label $l(v)$ that is assigned to each vertex in $v \in V$. The label $l(v)$ is iteratively updated until it is equal to the length of a shortest path from s to v . While

Dijkstra's algorithm updates the labels using the outgoing edges of the last vertex whose label has been marked as definitive, Bellman-Ford algorithm iterates $|V| - 1$ updates, using for the update each edge in E .

Algorithm 4: Bellman-Ford Shortest Path

Data: G, s

Result: shortest paths between s and any other vertex of G ; if there exists a negative cycle, it outputs its existence

```

1 for  $u \in V$  do
2    $l(u) := +\infty$ ;
3  $l(s) := 0$ ;
4 for  $i = 1$  to  $|V| - 1$  do
5   foreach edge  $\{u, v\} \in E$  do
6      $l(v) := \min(l(u), l(v) + w(\{u, v\}))$ ;
7 foreach edge  $\{u, v\} \in E$  do
8   if  $l(u) > l(v) + w(\{u, v\})$  then
9     Return("Negative cycle");
```

Floyd-Warshall algorithm

Floyd-Warshall algorithm (Floyd, 1962; Warshall, 1962) is a dynamic programming algorithm to compute the shortest path between all pairs of vertices in a graph. The graph can be weighted and can have negative edge weights, but not negative cycles. First, we describe the dynamic programming recurrence of the Floyd-Warshall algorithm. Given a weighted graph $G = (V, E)$, let the vertices in V be $\{v_1, v_2, \dots, v_n\}$. Define a function $P[i, j, k]$ that returns the weight of a shortest path from v_i to v_j , that contains only the vertices $v_1, v_2, \dots, v_k$ as intermediate vertices. Floyd-Warshall algorithm exploits a dynamic programming strategy to compute function $P[i, j, k]$. $P[i, j, k]$ can be computed considering two possible cases: (1) vertex v_k belongs to a shortest path between v_i and v_j or (2) v_k does not belong to a shortest path between v_i and v_j . The dynamic programming recurrence to compute $P[i, j, k]$ is then:

$$P[i, j, k] = \min \begin{cases} P[i, j, k-1] \\ P[i, k, k-1] + P[k, j, k-1] \end{cases}$$

In the base case, that is when $k=0$, it holds $P[i, j, 0] = w(\{v_i, v_j\})$. We present here an iterative algorithms that implements the recurrence given above.

Algorithm 5: Floyd-Warshall Algorithm

Data: G

Result: Shortest paths between all pairs of vertices

```

1 for  $i = 1$  to  $|V|$  do
2   for  $j = 1$  to  $|V|$  do
3      $P[i, j, 0] := w(\{v_i, v_j\})$ ;
4     /* We assume that  $w(\{v_i, v_j\}) = +\infty$  if  $\{v_i, v_j\} \notin E$  */
5 for  $k = 1$  to  $|V|$  do
6   for  $i = 1$  to  $|V|$  do
7     for  $j = 1$  to  $|V|$  do
8        $P[i, j, k] := \min(P[i, j, k-1], P[i, k, k-1] + P[k, j, k-1])$ ;
```

The recurrence computes only the weight (or the length for unweighted graphs) of a shortest path between v_i and v_j . However, a shortest path can be computed for example by using a matrix $PA[i, j, k]$, where the entries are used to memorize the intermediate

vertices. The entries in matrix $PA[i, j, k]$ represent the predecessor of vertex v_j on a shortest path from v_i to v_j that contains only vertices in $v_1, v_2, \dots, v_k$. We present the recurrence to compute $PA[i, j, k]$, for $k \geq 1$:

$$PA[i, j, k] = \begin{cases} PA[i, j, k-1] & \text{if } P[i, j, k-1] \geq P[i, k, k-1] + P[k, j, k-1] \\ PA[k, j, k-1] & \text{if } P[i, j, k-1] > P[i, k, k-1] + P[k, j, k-1] \end{cases}$$

When $k=0$, $PA[i, j, 0] = v_i$. Notice that Floyd-Warshall algorithm computes a solution to the all-pairs shortest path problem, that is it computes a shortest path between any pair of vertices of the graph, while Dijkstra's algorithm and Bellman-Ford algorithm computes the shortest path from a given source vertex s .

Ford-Fulkerson Algorithm

The Ford-Fulkerson algorithm (Ford and Fulkerson, 1962) allows to compute the maximum flow in a graph. Consider a graph G , with two specific vertices s and t , called the source and the sink, respectively. Each edge of the graph is associated with a *capacity*, which denotes the maximum flow that is allowed on that edge.

Given a graph $G=(V, E, c)$ where c is a function that associates a capacity $c(\{u, v\}) \geq 0$ with each edge $\{u, v\} \in E$, a source $s \in V$ and a sink $t \in V$, the maximum flow problem asks for the maximum flow F that can be sent in G from s to t without violating the edge capacities.

Formally, let $F : E \rightarrow \mathbb{R}$ be a flow function. The flow F from s to t must satisfy the following constraints:

- $F(\{u, v\}) \leq c(\{u, v\})$ (capacity constraints)
- $F(\{u, v\}) = -F(\{v, u\})$ (skew symmetry)
- $\sum_{u \in V} F(\{v, u\}) = 0$, $\forall v \in V \setminus \{s, t\}$ (flow conservation).

Consider a flow F in G , we define the *residual graph* $G_F=(V, E, c_F)$, where the capacity c_F of each edge in E is defined as follows:

$$c_F(\{u, v\}) = c(\{u, v\}) - F(\{u, v\})$$

The Ford-Fulkerson algorithm computes the maximum flow in a graph. Given a flow F , the Ford-Fulkerson algorithm looks for an augmenting path in G_F , that is a path π from s to t in G_F such that the edges in the path have a positive capacity, thus allowing some flow to be sent from s to t . The algorithm is iterated until it is possible to find an augmenting path.

Algorithm 6: Ford-Fulkerson Algorithm

Data: $G=(V, E, c)$, $s, t \in V$

Result: A maximum flow from s to t in G

```

1 foreach  $\{u, v\} \in E$  do
2    $F(\{u, v\}) := 0$ ;
3    $F(\{v, u\}) := 0$ ;
4 while There exists an augmenting path  $\pi$  in  $G_F$  do
5   define  $c_F(\pi) = \min(c_F(\{u, v\}) : \{u, v\} \text{ belongs to } \pi)$ ;
6   foreach  $\{u, v\}$  in  $\pi$  do
7      $F(\{u, v\}) := F(\{u, v\}) + c_F(\pi)$ ;
8      $F(\{v, u\}) := -F(\{u, v\})$ ;
```

A cut in a graph G is a set of edges whose removal disconnect G . A *minimum cut* is a cut whose edges has minimum weight. A well-known result in graph theory is that the maximum flow of a graph is equal to the value of a minimum cut in a graph. Thus, Ford-Fulkerson algorithm can be used to compute a minimum cut of a graph.

Minimum Spanning Tree

In this section we present two algorithms to compute a minimum spanning tree: Kruskal's algorithm and Prim's algorithm. Both algorithms compute a minimum spanning tree of a weighted graph $G=(V, E)$ with a greedy strategy, but differ in the way vertices are added to the spanning tree.

Kruskal's algorithm

Kruskal's algorithm (Kruskal, 1956) starts from a forest consisting of an empty set of edges, and constructs a forest by greedily adding edges of minimum weight. Let $G=(V, E)$ be a graph and let $F=(V_F, E_F)$ be a forest, with $V_F=V$ and $E_F \subseteq E$. F is initialized as follows: $V_F=V$ and $E_F=\emptyset$. Kruskal's algorithm selects the edge $\{u, v\} \in E/E_F$ of minimum weight such that u and v belong to different trees of forest F ; $\{u, v\}$ is added to E_F and the trees containing u and v are merged.

Fig. 3 shows how the edges of a graph G are added to E_F . Notice that at each step, the selected edge is that of minimum weight that does not induce a cycle, independently from the fact that the selected edge shares an endpoint with other edges in E_F or not. Notice that in Fig. 3(a) F consists of six trees, each one containing of a single vertex. In Fig. 3(b), Kruskal's algorithm select edge $\{v_1, v_2\}$ of minimum weight and merges the trees containing v_1 and v_2 . Hence, F consists of five trees: the tree T_1 having vertices v_1, v_2 and edge $\{v_1, v_2\}$, and four trees each one containing a single vertex of v_3, v_4, v_5, v_6 . In Fig. 3(c), Kruskal's algorithm selects edge $\{v_3, v_6\}$ and merges the trees containing v_3 and v_6 . Hence, F consists of four trees: the tree T_1 , the tree T_2 having vertices v_3, v_6 and edge $\{v_3, v_6\}$, and two trees each one containing a single vertex of v_4, v_5 . In Fig. 3(d), Kruskal's algorithm selects edge $\{v_2, v_5\}$ and merges the tree T_1 and the tree containing v_5 . Hence, F consists of four trees: the tree T'_1 containing vertices v_1, v_2 and v_5 and edges $\{v_1, v_2\}$ and $\{v_2, v_5\}$, tree T_2 and one tree containing vertex v_4 . In Fig. 3(e), edge $\{v_2, v_3\}$ is selected and trees T'_1 and T_2 are merged, while in Fig. 3(f) $\{v_4, v_5\}$ is selected and a minimum spanning tree of G is finally computed.

Prim's algorithm

Prim's algorithm (Prim, 1957) starts from an empty tree, and constructs a subtree of the input graph by greedily adding edges of minimum weight. However, while Kruskal's algorithm constructs a forest, Prim's algorithm adds edges that induce a connected subtree at each step of the algorithm. Formally, let $G=(V, E)$ be a graph and let $T=(V_T, E_T)$ be a tree, with $V_T \subseteq V$ and $E_T \subseteq E$. T is initialized as follows: $V_T=\{r\}$, for some vertex $r \in V$ and $E_T=\emptyset$. Prim's algorithm selects the edge $\{u, v\} \in E/E_T$ of minimum weight such that exactly one of u, v , without loss of generality v , belongs to V_T ; then $\{u, v\}$ is added to E_T and u is added to V_T . Notice that, since for each selected edge $\{u, v\}$, $v \in V_T$ and $u \in V/V_T$, it follows that $\{u, v\}$ does not induce a cycle in G with edges of E_T .

Fig. 4 shows how the edges of a graph G are added to E_T by Prim's algorithm. At each step, the selected edge is that of minimum weight that contains a vertex in V_T and a vertex in V/V_T . Assume that vertex r is v_1 , that is $V_T=\{v_1\}$ in the first step. Then Prim's algorithm in Fig. 4(b) adds edge $\{v_1, v_2\}$ to E_T and adds vertex v_2 to V_T . In Fig. 4(c), Prim's algorithm adds edges $\{v_1, v_3\}$ to E_T and v_3 to V_T . Notice that $\{v_1, v_3\}$ is not the edge having minimum weight in E/E_T of Fig. 4(b), since the edge of minimum weight in E/E_T of Fig. 4(b) is $\{v_3, v_6\}$, but $v_3, v_6 \notin V_T$. In Fig. 4(d)–(f), Prim's algorithm selects edges $\{v_3, v_6\}$, $\{v_2, v_5\}$ and $\{v_5, v_4\}$, respectively, adds these edges to E_T and adds vertices v_6, v_5 , and v_4 , respectively, to V_T . In Fig. 4(f) a minimum spanning tree of G computed by Prim's algorithm is shown.

Nearest Neighbour Algorithm

One of the most studied problem in graph theory in the last fifty years is the *Traveling Salesman Problem* (TSP). Given a weighted graph $G=(V, E)$, and a vertex $s \in V$, TSP asks for a circuit of minimum weight that starts and ends in vertex s and that contains all the vertices of G .

Since the TSP is NP-hard, heuristics and approximation algorithms have been considered to solve the problem. These algorithms have polynomial-time complexity, but may returns suboptimal solutions. The nearest neighbour algorithm is a well-known heuristic for TSP. At each step, it considers a current vertex v and it considers the edges having minimum weight that connects v to an unvisited vertex u ; u is then considered as the current vertex. Notice that the heuristics may not be able to find a feasible solution

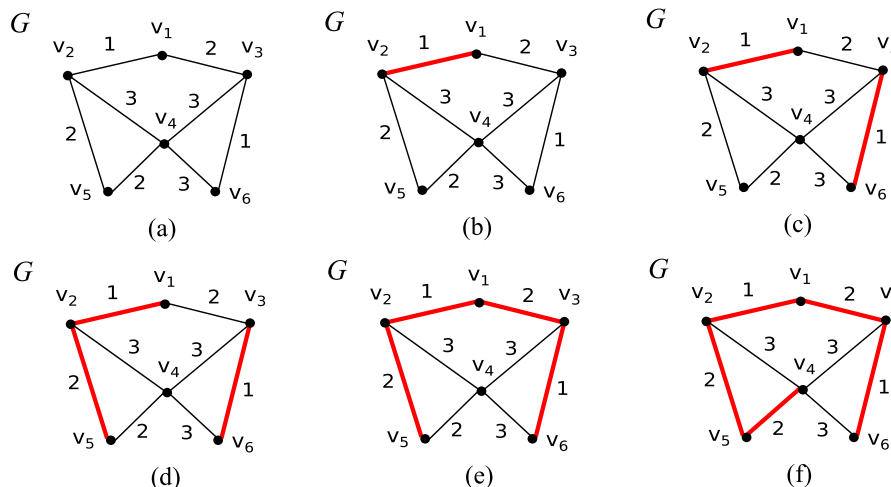


Fig. 3 An example of execution of Kruskal's algorithm. The bold red edges added to those added to E_F .

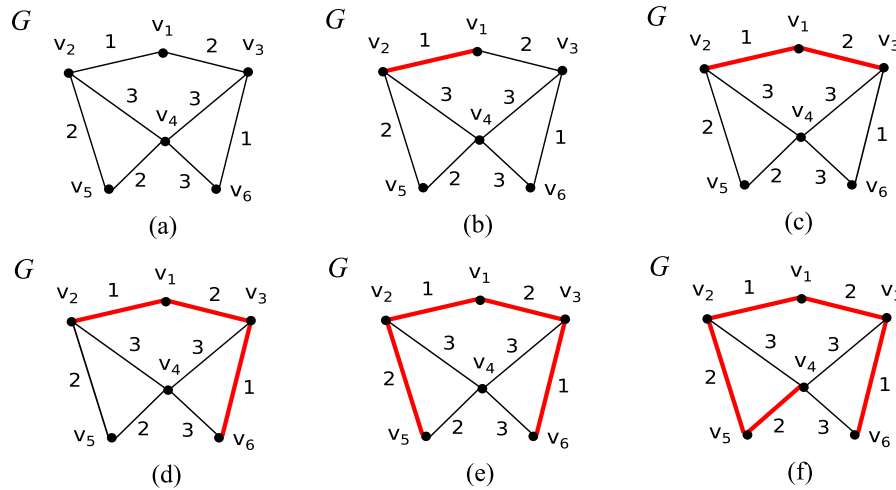


Fig. 4 An example of execution of Prim's algorithm. The bold red edges added to those added to E_T .

of TSP (that is a circuit that contains all the vertices of G), even if one exists. Moreover, even if it finds a feasible solution of TSP, the solution returned can be α times more than the optimal one, for each constant $\alpha > 0$, and may return the worst possible feasible solution (Gutin *et al.*, 2002).

Analysis and Assessment

In this section we discuss the time-complexity of the algorithms presented in Section Methodologies.

Breadth-First Search and Depth-First Search

Breadth-first search and depth-first search can be implemented so that their time-complexity is $O(|V| + |E|)$ (Cormen *et al.*, 2009). More specifically, implementation of breadth-first search uses a queue to store and retrieve the vertices to be visited, while depth-first search uses a stack to store and retrieve the vertices to be visited.

Shortest Path Algorithms

Dijkstra's algorithm using a Fibonacci heap in the implementation can achieve a time-complexity of $O(|V| \log |V|)$ (Cormen *et al.*, 2009). The Bellman-Ford algorithm has a time-complexity of $O(|V||E|)$ (Cormen *et al.*, 2009). Floyd-Warshall algorithm has a time-complexity of $O(|V|^3)$, due to the iteration of line- 5–8 of Algorithm 5 (Cormen *et al.*, 2009).

Ford-Fulkerson Algorithm

The time-complexity of the Ford-Fulkerson algorithm described in Section Methodologies depends on the algorithm used to find an augmenting path in the residual graph G_F . Notice that for some choice of the algorithm to compute the augmenting path, the Ford-Fulkerson may not even terminate. An augmenting path in the residual graph G_F can be computed efficiently using a BFS search. In this case the time-complexity of the algorithm is $O(|V||E|^2)$ (Cormen *et al.*, 2009).

Minimum Spanning Tree Algorithms

Kruskal's algorithm can implemented so that its running time is $O(|E| \log |E|)$, by using a disjoint-set data structure to store disjoint trees of forest, and applying Union-Find operations to merge two trees and find the tree containing a given vertex, respectively. In particular, the disjoint-set data structure can be implemented efficiently with the union-by-rank and path-compression heuristics.

Prim's algorithm can achieve a time-complexity $O(|E| \log |V|)$, using a binary heap in the implementation (Cormen *et al.*, 2009).

Traveling Salesman Person

The TSP problem is an NP-hard problem, and it is one of the of the classic NP-hard problems considered in Garey and Johnson (1995). The NP-hardness of TSP, even in the restricted case that each edge has weight equal to one, that is the graph in unweighted, follows from the NP-completeness of the problem of finding a Hamiltonian cycle in a graph (Karp, 1972). Indeed, in an

unweighted graph an optimal solution of TSP is a Hamiltonian cycle, thus from the NP-completeness of computing a Hamiltonian cycle follows the NP-hardness of TSP.

The time-complexity of nearest neighbour algorithm for TSP is $O(|V|)$, since it performs at most $|V|$ iteration and in each iteration it visits at most $|E|$ edges.

Illustrative Examples

In this section we present some examples where graph algorithms have been applied in computational biology.

Graph Traversal Algorithms

Depth-first search and breadth first search have many application in graph theory and in computational biology. Two notable examples are the computation of the connected components of a graph and strongly connected components of a directed graph, and the test of bipartiteness of a graph.

Depth-first search can be applied to compute the partition of a directed graph in strongly connected components. The decomposition of a directed graph in strongly connected components has been applied as a relevant step for the synthesis of signal transduction networks. The approach applied in [Albert et al. \(2007, 2008\)](#) aims to remove false positive of a signal transduction network by constructing a sparsest graph that maintains all reachability relationships of a given directed graph. This goal is reached by first decomposing the directed graph in strongly connected components, and then solving a second problem (called binary transitive reduction) on each strongly connected component independently.

Breadth-first search can be applied to test if a graph $G=(V,E)$ is bipartite, that is if the vertices of G can be partitioned in two disjoint sets V_1 and V_2 such that there is no edge between two vertices of V_1 (of V_2 , respectively). Testing bipartiteness is a fundamental in many practical problems, a notable example in computational biology is haplotyping (for a survey see [Bonizzoni et al., 2003](#)). One relevant problem in haplotyping is the single individual haplotype problem ([Lippert et al., 2002](#)). The single individual haplotype problem asks for a bipartition of a set of fragments coming from a haplotype, in order to be able to reconstruct the two copies of a haplotype from fragments. Fragments are usual represented as vertices of a graph, called conflict graph, and there exists an edge between two vertices if the corresponding fragments have different values in some positions. The reconstruction of the two copies of a haplotype corresponds to a bipartition of the conflict graph. Several variants of the problem have been proposed to correct errors in fragments ([Lippert et al., 2002](#)).

Ford-Fulkerson Algorithm

The computation of maximum flow and minimum cut has many applications in computational biology. The computation of a minimum cut has been applied for example to design heuristics for the supertree problem. Given a set S of trees, the goal of the supertree problem is to compute a supertree by combining different trees in S . Two natural heuristics for this problem apply iteratively a minimum cut on a graph that represents the relations among leaves ([Semple and Steel, 2000](#); [Page, 2002](#)). A similar strategy has been applied also in [Dondi et al. \(2017\)](#) to design an approximation algorithm for the problem of correcting a set of orthologous/paralogous relations between genes.

Spanning Trees

Spanning trees have been widely applied in the context of networking, but has found application also in computational biology. An approach for the clustering of microarray gene-expression data is based on the computation of a minimum spanning tree of a graph G that represents multi-dimensional gene expression data ([Xu et al., 2001](#)). An interesting property of this graph is that cluster corresponds to minimum spanning tree, thus a cluster of the expression data is computed by computing a partition of G in minimum spanning trees.

Traveling Salesman Person

The TSP problem has been applied to many problems in several contexts. One notable example in computational biology is the shortest superstring problem, a problem that has application in the context of genome assembly.

Given a set of strings that have to be assembled, a graph representation define a vertex for each string and an edge for each pair of overlapping strings and the weight of an edge is the length of the overlapping. Then a shortest superstring of the given strings corresponds to a solution of a variant of TSP in the associated graph ([Mucha, 2013](#)).

Closing Remarks

We have reviewed some of the main graph algorithms. First we have presented algorithms related to graph traversal (depth-first search and breadth-first search) and we have reviewed algorithms for the computation of shortest paths. We have considered the

Ford-Fulkerson algorithm for the computation of the maximum flow. Then we have reviewed the Kruskal's algorithm and the Prim's algorithm for the computation of the minimum spanning tree. Finally, we have presented the traveling salesman problem and a heuristic for its solution, the nearest neighbour algorithm. Moreover, we have presented some applications of these algorithms to problems in computational biology.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Natural Language Processing Approaches in Bioinformatics

References

- Albert, R., DasGupta, B., Dondi, R., *et al.*, 2007. A novel method for signal transduction network inference from indirect experimental evidence. *Journal of Computational Biology* 14, 927–949. doi:10.1089/cmb.2007.0015.
- Albert, R., DasGupta, B., Dondi, R., Sontag, E.D., 2008. Inferring (biological) signal transduction networks via transitive reductions of directed graphs. *Algorithmica* 51, 129–159. doi:10.1007/s00453-007-9055-0.
- Bellman, R., 1958. On a routing problem. *Quarterly of Applied Mathematics* 16, 87–90.
- Bonizzoni, P., Vedova, G.D., Dondi, R., Li, J., 2003. The haplotyping problem: An overview of computational models and solutions. *Journal of Computer Science and Technology* 18, 675–688. doi:10.1007/BF02945456.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. *Introduction to Algorithms*. MIT Press.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Nu-Merische Mathematik* 1, 269–271.
- Dondi, R., Lafond, M., El-Mabrouk, N., 2017. Approximating the correction of weighted and unweighted orthology and paralogy relations. *Algorithms for Molecular Biology* 12, 4. doi:10.1186/s13015-017-0096-x.
- Floyd, R.W., 1962. Algorithm 97: Shortest path. *Communication of the ACM* 5, 345. doi:10.1145/367766.368168.
- Floyd, R.W., Rivest, R.L., 1975. Expected time bounds for selection. *Communication of the ACM* 18, 165–172. doi:10.1145/360680.360691.
- Ford, L.R., Fulkerson, D.R., 1962. *Flows in Networks*. Princeton University Press.
- Garey, M.R., Johnson, D.S., 1995. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co.
- Gutin, G., Yeo, A., Zverovich, A., 2002. Traveling salesman should not be greedy: Domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics* 117, 81–86. doi:10.1016/S0166-218X(01)00195-0.
- Karp, R.M., 1972. Reducibility among combinatorial problems, In: *Proceedings of a symposium on the Complexity of Computer Computations*, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, York-town Heights, New York. pp. 85–103.
- Kruskal, J.B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. In: *Proceedings of the American Mathematical Society*, p. 7.
- Lippert, R., Schwartz, R., Lancia, G., Istrail, S., 2002. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in Bioinformatics* 3, 23–31. doi:10.1093/bib/3.1.23.
- Mucha, M., 2013. Lyndon words and short superstrings. In: Khanna, S. (Ed.), *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013*, New Orleans, Louisiana, USA, January 6–8, 2013, SIAM. pp. 958–972. DOI:10.1137/1.9781611973105.69.
- Page, R.D.M., 2002. Modified mincut supertrees. In: Guigo, R., Gusfield, D. (Eds.), *Proceedings of the Algorithms in Bioinformatics, Second International Workshop, WABI 2002*, Rome, Italy, September 17–21, 2002, Springer. pp. 537–552. DOI: 10.1007/3-540-45784-4_41.
- Prim, R.C., 1957. Shortest connection networks and some generalizations. *The Bell Systems Technical Journal* 36, 1389–1401.
- Seiple, C., Steel, M.A., 2000. A supertree method for rooted trees. *Discrete Applied Mathematics* 105, 147–158. doi:10.1016/S0166-218X(00)00202-X.
- Warshall, S., 1962. A theorem on boolean matrices. *Journal of the ACM* 9, 11–12. doi:10.1145/321105.321107.
- Xu, Y., Olman, V., Xu, D., 2001. Minimum spanning trees for gene expression data clustering. *Genome Informatics* 12, 24–33. doi:10.11234/gi1990.12.24.

Further Reading

- Gutin, G., Punnen, A., 2002. *The Traveling Salesman Problem and its Variations*. Combinatorial Optimization. US: Springer.
- Kleinberg, J., Tardos, E., 2013. *Algorithm Design*. Pearson.
- Schrijver, A., 2002. *Combinatorial optimization: Polyhedra and efficiency*. Algorithms and Combinatorics. Berlin Heidelberg: Springer.
- Skiena, S.S., 2008. *The Algorithm Design Manual*, 2nd ed. Springer.

Relevant Websites

- <http://www.math.uwaterloo.ca/tsp/>
History, application, and research on TSP.
- <http://visualgo.net/en>
Visualization of many graph algorithms.

Network Centralities and Node Ranking

Raffaele Giancarlo, Daniele Greco, Francesco Landolina, and Simona E Rombo, University of Palermo, Palermo, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Studying the significant role that single nodes play in a graph representing a network may provide important insights into the network structure and the functions that it implicitly codifies. Ranking or identifying the node importance is one of the first steps to understand the hierarchical organization of biological networks (Liu *et al.*, 2012). To this aim, the number of edges outcoming from a node, that is, its “degree,” is the simplest and intuitive indicator of how much important is that node. Traditionally, large degree nodes (also called *hubs*) are deemed as important nodes in a network. To cite another example of characterization of the importance of a node in a network, one can refer to its *H-index*, defined as the maximum integer h such that the considered node has at least h neighbours whose degrees are greater than h . Higher H-index indicates that the node has a number of neighbours with high degree. Compared with degree, H-index can better capture the spreading importance. A node with higher degree infects many neighbours at start times, while the spread process will cease quickly if its neighbours have low degrees.

In general, more sophisticated centrality indices have been proposed in the literature in order to provide a quantification of the importance, in terms of centrality, of nodes in a network. A plethora of different characterizations are available in literature, each of them scoring a different structural property for the network elements (Brandes and Erlebach, 2005).

Historically, centrality indices were introduced in the context of social network analysis to discover the most influential individuals in a community (Bonacich, 1972; Freeman, 1978; Leo, 1953). In biological sciences, they are employed to identify key players in biological processes modeled as biological networks such as gene regulatory, metabolic and protein-protein interaction networks (Koschützki and Schreiber, 2008; Pavlopoulos *et al.*, 2011). Furthermore, since *structure always affects function*, (Strogatz, 2001) centrality analysis aims at assessing the extent to which topological prominence predicts biological importance.

For example, Joeng *et al.* showed that high-degree nodes (hubs) in yeast protein interaction network are enriched in essential proteins, suggesting that essentiality is related to the central role that hubs play in mediating interactions among a large number of other, less connected, proteins (Jeong *et al.*, 2001). As a complementary approach, Gerstein *et al.* found that in regulatory networks, essentiality better correlates with betweenness-centrality -the number of shortest paths passing through a node- than degree-centrality (Yu *et al.*, 2007). Finally, Przytycka *et al.* provided a careful analysis of six different *S. cerevisiae* protein networks with different centrality measures, showing that non-essential protein-hubs are no more important than essential protein-hubs in maintaining the interactome connectivity, and that those centrality indices which better predicts nodes involved in topological network cohesiveness, are no better predictors for essential genes than degree-centrality (Zotenko *et al.*, 2008).

These examples clearly illustrate how the same input network can be valuable analyzed using different centrality measures, yielding complementary, contrasting, still informative biological perspectives on its constituents.

Ranking algorithms based on node centrality measures have been proposed and applied in the bioinformatics domain (Pavlopoulos *et al.*, 2011).

This manuscript aims at providing an overview of the main node centrality measures (Section Centrality Measures), by also showing some 5 we draw our conclusions. The next section is devoted to introduce some basic definitions needed for the full understanding of the manuscript.

Basics

We represent networks by *undirected, unweighted* graphs $G=(V,E)$, where V denotes the set of nodes and E the set of edges connecting them. G may also be represented in terms of its *adjacency matrix* $A=[a_{ij}]$, where $a_{ij}=1$ if $\{i,j\} \in E$ and 0 otherwise.

A *walk* from node u to node v is an alternating list $v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$ the edge e_i has endpoints v_{i-1} and v_i , with $v_0=u$ and $v_k=v$. A *trail* is a walk in which no edge is traversed more than once. A *path* is a trail in which no node is visited more than once.

The *length* of a walk is the number of edges traversed. We define the *distance* d_{uv} as the length of a *shortest* path connecting vertices u and v . Where necessary, directed graphs are introduced as well.

Centrality analysis is a type of network structural analysis (Ma and Gao, 2012). A centrality index induces a ranking of vertices (or edges) of a graph G by assigning them a real value based uniquely on its network structure.

Accordingly, a minimal requirement for a real-valued function $c: V \rightarrow \mathbb{R}$ to be a centrality index is that it can be computed having only access to a representation of G , e.g., its adjacency matrix.

Two graphs $G=(V_G, E_G)$ and $H=(V_H, E_H)$ are isomorphic ($G \simeq H$) if there exists a bijection $\phi: V_G \rightarrow V_H$ such that $(u,v) \in E_G$ if and only if $(\phi(u), \phi(v)) \in E_H$.

We recall from (Brandes and Erlebach, 2005) the definition of *structural index*.

Definition 1: (Structural Index) Let $G=(V,E)$ be a weighted, directed or undirected graph and let X represent the set of vertices or edges of G , respectively. A real-valued function s is called a structural index if and only if the following condition is satisfied: $\forall x \in X: G \simeq H \Rightarrow s_G(x) = s_H(\phi(x))$, where $s_G(x)$ denotes the value of $s(x)$ in G .

Every centrality index c discussed here is a structural index. In particular, we are only concerned in the induced ranking of vertices (or edges). Therefore, we consider that a vertex u is *more important* than another vertex v , with respect to c , if $c(u) > c(v)$.

It is worth noting that the above definition holds a fundamental limitation: real values assigned to vertices by a centrality index *do not* account for their *relative* importance. As pointed out in, (Brandes and Erlebach, 2005) the difference or ratio of two centrality values cannot be interpreted as how much more central one element is than the other. Indeed, other metrics and methods have to be used for this purpose (Lawyer, 2015).

Centrality indices serve only the purpose of producing a ranking which allows the identification of the most prominent nodes or edges in the network with respect to a given structural property.

Centrality Measures

Degree and Walk-Based Centralities

In this section, we present centrality measures based on the set of walks of G originating (or terminating) at a particular vertex, spanning from walks of length 1 (degree centrality) to length infinity (eigenvector centrality).

Degree centrality

Degree centrality $c_D(i)$ is the simplest measure of centrality, which assigns to each vertex i in G its degree:

$$c_D(i) = \sum_{j=1}^N A_{ij} \quad (1)$$

In case of a directed network, two separate centralities are defined: the *in-degree* $c_D^-(i)$ and *out-degree* $c_D^+(i)$ centralities, that is, the number of in-going and out-going edges of i , respectively.

Degree centrality is a local centrality index, ranking vertices of G according to their immediate neighborhood: the larger the number of interactions, the more important the node in the network. Furthermore, we can interpret every edge as a walk of length 1, and consider $c_D(i)$ as counting the number of paths of length 1 originating (or terminating) at node i .

High degree-centrality nodes are called *hubs*. In scale-free networks, the removal of hubs rapidly fragments the network, leading to different isolated connected components (Barabási and Pósfai, 2016). Under the assumption that an organism's function depends on the connectivity among various parts of its interactome, Jeong *et al.* (2001) correlate the degree of proteins in a *S. cerevisiae* protein interaction network to the lethality of their removal.

Furthermore, in graph evolution models high vertex degrees accounts not only for cohesiveness, but also for age. Indeed, in metabolic networks (Fell and Wagner, 2000) show that the metabolites with the highest connectivity are part of the oldest "core" metabolism of *E. coli*.

Katz centrality

In contrast with degree centrality, Katz status index (Leo, 1953) takes into account *all* walks originating (or terminating) at a given node i , and weights them according to their length. Intuitively, the greater the length the less influential a connection is.

Mathematically, Katz centrality $c_K(i)$ of vertex i is defined as:

$$c_K(i) = \sum_{k=1}^{\infty} \sum_{j=1}^N \alpha^k (A^k)_{ji} \quad (2)$$

The k -th power of the adjacency matrix A holds the total number of walks of length k between any two vertices. In order for the infinite sum to converge, the attenuation factor α has to be chosen in $[0, \frac{1}{\lambda_1}]$ where λ_1 is the largest eigenvalue of A (Ferrar, 1951). In this case, the following matrix form can be used to compute the centrality values:

$$\vec{c_K} = \left((I - \alpha A^T)^{-1} - I \right) \vec{1_N} \quad (3)$$

where $\vec{1_N}$ is the identity vector of size N (the number of nodes) with all entries set to 1, I is the $N \times N$ identity matrix, A^T denotes matrix transposition of A and $()^{-1}$ denotes matrix inversion.

Recently, Fletcher and Wennekers (2017) studying firing activity in neural networks, have shown Katz centrality to be the best predictor of firing rate with almost perfect correlation for all types of neural networks considered.

Eigenvector centrality

Bonacich's Eigenvector centrality (Bonacich, 1972) acknowledges the fact that not all the connections are equal. Recursively, the importance of a vertex depends both on the number and the importance of its neighbours, with high-scoring neighbours contributing more than low-scoring ones.

In the following, we will consider a graph G which is undirected, connected and unweighted. The eigenvector centrality x_i of node i , is proportional to the sum of the eigenvector centralities of its immediate neighbours:

$$x_i = \frac{1}{\lambda} \sum_{j \in G} A_{ij} x_j \quad (4)$$

where λ is a positive constant. In vector form, it can be rewritten as the eigenvector equation:

$$A \mathbf{x} = \lambda \mathbf{x} \quad (5)$$

In general, there are multiple eigenvalues λ for which a non-zero eigenvector satisfying Eq. (5) exists. Since G is undirected and connected, its adjacency matrix A results in a nonnegative, irreducible real matrix. Under these conditions, by the Perron-Frobenius theorem, (Gantmacher, 1960) there exists a unique largest eigenvalue $\lambda_1 = \lambda_{\max}$ which is real and positive. The corresponding eigenvector $\mathbf{x}^*$ is unique up to a common factor and contains entries which are both nonzero and of the same sign. Therefore, we take its absolute value and normalize it:

$$\mathbf{x}_{EG} = \frac{|\mathbf{x}^*|}{\|\mathbf{x}^*\|} \quad (6)$$

The eigenvector centrality $c_{EG}(i)$ of node i is defined as the i -th component of $\mathbf{x}_{EG}$.

As discussed in, (Borgatti and Everett, 2006) eigenvector centrality can be regarded as an elegant summary of Leo (1953), Hoede (1978) and Hubbell's measures (Hubbell, 1965). Notably, Google's Page Rank (Page et al., 1999) is a variant of eigenvector centrality.

Zotenko et al. (2008) used eigenvector centrality in comparison with other centrality indices, to identify network hubs in *S. cerevisiae* protein-protein interaction networks. Other applications include prediction of synthetic genetic interactions (Paladugu et al., 2008) and identification of putative gene-disease associations on a literature mined gene-interaction network (Özgür et al., 2008).

Subgraph centrality

The Subgraph centrality (Estrada and Rodriguez-Velazquez, 2005) of the vertex i is equal to the weighted sum of all closed walks that start and terminate at vertex i . As this is an infinite sum, convergence is guaranteed assigning walks of length k the weight $\frac{1}{k!}$. Thus, the subgraph centrality $c_{SG}(i)$ of node i is defined as:

$$c_{SG}(i) = \sum_{k=0}^{\infty} \frac{(A^k)_{ii}}{k!} \quad (7)$$

Closed walks are directly related to the subgraphs of the network (e.g., triangles). Furthermore, smaller subgraphs are given more weight than larger ones, based on the observation that real-world networks are motif-rich (Alon, 2007).

In biological networks, subgraph centrality has been used mostly for the identification of essential proteins in yeast protein networks (Estrada, 2006, 2010; Zotenko et al., 2008). According to Estrada (2006) subgraph centrality slightly outperform degree centrality in identifying essential proteins in yeast protein network and shows similar performance with eigenvector centrality.

Shortest-Paths Based Centralities

In this section, we present centralities based on the set of all shortest paths and distances within graphs. Unless otherwise stated, we will assume the input graph $G=(V,E)$ to be connected.

Eccentricity centrality

Let $ecc(u)$ be the maximum distance between u and any other vertex v in the network. The quantity $ecc(u)$ is the *eccentricity* of u . Eccentricity centrality (Hage and Harary, 1995) $c_E(u)$ is defined as the inverse of $ecc(u)$:

$$c_E(u) = \frac{1}{ecc(u)} = \frac{1}{\max\{d_{uv} : v \in V\}} \quad (8)$$

The greater $c_E(u)$ the more proximal the node u to any other node v in the network. In graph theory, (Harary, 1969) the set of nodes of G with minimum eccentricity $ecc(u)$, and thus maximum $c_E(u)$, is denoted as $C(G)$, the *center* of G .

In biological networks, the computation of high-eccentricity nodes in the metabolic network of *E. coli* allowed to predict some of the most central substrates involved in energetics, signalling, and also specific (Glycolysis, Citrate cycle) pathways of the cell (Wuchty and Stadler, 2003).

Closeness centrality

In contrast with eccentricity which considers only the longest shortest paths, closeness centrality $c_C(u)$ uses all the shortest paths between u and any other vertex v in the network.

Closeness-centrality $c_C(u)$ is defined as the reciprocal of *farness*, (Bavelas, 2017) the sum of distances between u and all other vertices:

$$c_C(u) = \frac{1}{\sum_{v \in V} d_{uv}} \quad (9)$$

Closeness centrality can be thought as the average distance between the node u and all other nodes in the graph. Therefore, the shorter the distance between u and any other vertex v , the more central the node is.

Closely related to $c_C(u)$, the *radiality* index $c_R(u)$ adapted from Valente and Foreman (1998) for undirected graphs, measures how well a node is integrated in the network. Let $diam(G)$ be the *diameter* of G , the maximum distance between any two vertices in the graph G .

The radiality $c_R(u)$ if node u is defined as:

$$c_R(u) = \frac{\sum_{v \in V} (diam(G) + 1 - d_{uv})}{N - 1} \quad (10)$$

where N denotes the number of vertices. The major difference with closeness centrality is that $c_R(u)$ reverses distances with respect to the diameter instead of taking the reciprocal.

Ma and Zeng (2003) used closeness centrality to identify the top central metabolites in *E. coli* metabolic network. In Mazurie et al. (2010) a normalized version of closeness centrality (Sabidussi, 1966) is used as a network structure descriptor in a cross-species comparison of metabolic networks.

Both closeness and radiality can also be applied to directed strongly connected networks.

Betweenness centrality

The betweenness of a vertex v as introduced in the context of social networks by Freeman (1978) refers to the extent to which a vertex v can control, or monitor, communication between any two other vertices in the network.

Under the assumption that information is transmitted along shortest paths, betweenness centrality measures how often v occurs on all shortest paths between any other two nodes in the network.

For a given pair of vertices s, t let σ_{st} be the number of all shortest-paths between s and t , and $\sigma_{st}(v)$ the number of shortest-paths between s and t containing v . We denote with $\delta_{st}(v)$:

$$\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (11)$$

the *fraction* of all shortest-paths between s and t containing v . The quantity $\delta_{st}(v)$ can be interpreted as the probability that a "message" from s to t goes through v . Iterating over all possible vertex pairs, the shortest-path betweenness centrality $c_{SPB}(v)$ of vertex v is defined as:

$$c_{SPB}(v) = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \delta_{st}(v) \quad (12)$$

Thus, the larger the overlap between the set all of shortest paths and the set of shortest paths containing v , the more central the node v .

Both Yu et al. (2007) and Joy et al. (2005) highlight proteins with high betweenness but low degree to be prominent in yeast PPI networks, suggesting they could support modular organization acting as key connector nodes between different, more densely connected, functional modules.

In mammalian transcriptional regulatory networks, Potapov et al. (2005) report high degree and high betweenness genes to be enriched with genes having tumor-suppressor or proto-oncogene properties.

Examples

In this section we propose examples in order to show how some of the centrality measures described in the previous sections can be calculated.

Let $G=(V,E)$ the graph shown in Fig. 1 together with its adjacent matrix. We focus on three measures, Degree, Betweenness and Closeness Centrality, respectively, that have been applied to G in order to highlight which nodes have a high value of the considered measure and what happens to the graph if one of these nodes is eliminated.

Before going on with we recall the notion of *all pair shortest path*, that concerns the search of all the shortest paths for each pair of nodes in an input graph. To do it, there are several techniques such as Floyd-Warshall (efficient for dense graphs) and Johnson (efficient for spread graphs) suitably adapted for the case of unweighted and undirected graphs (Aho et al., 1974).

Table 1 shows *all pair shortest paths* computed for the graph G in Fig. 2. It should be pointed out that the identified paths shown in the table have only been inserted once, since they can be run in both directions (undirected) as part of the symmetric array represented by the adjacency matrix of G .

Degree Centrality

We first consider the Degree Centrality, according to which a particular node is central to the network if it is one of the most active within it, i.e., if the node participates to a bigger number of associations/edges than the others nodes. An active node is a node that interacts frequently with the others by indicating which nodes are in direct contact with each other. Nodes with a high degree centrality could be interpreted as *crucial nodes* for the network considered, and even though it is easy to calculate, it provides an

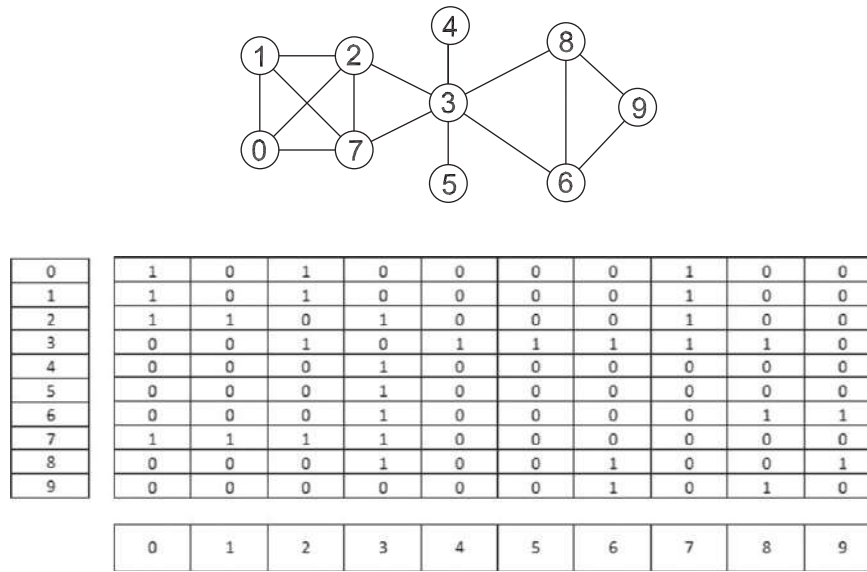


Fig. 1 An input graph $G=(V, E)$ and its adjacent matrix.

important information to identify the essential nodes in the graph. The index of a node can be defined as the number of nodes adjacent to it, or as the *cardinality of its neighborhood*. Consider the adjacent matrix of G shown in [Fig. 1](#). The degree centrality of a node can be computed by adding all non-zero values to the columns (or rows):

$$c_D(i) = \sum_{j=1}^N A_{ji} = \sum_{j=1}^N A_{ij} \quad (13)$$

Example: By calculating the index of node 3, we have to sum all the numbers of its row (column), obtaining an index of 6. [Table 2](#) shows the values of degree centrality for each node in G .

[Fig. 1](#) illustrates what happens if the node scoring the highest value of Degree Centrality is removed from the network. In particular, in that case G becomes a disconnected graph, including two singletons and two small sub-graphs of three and four nodes, respectively. This means that, more in general, by eliminating from the network those nodes scoring high values of Degree Centrality many knots could not be reached.

Closeness Centrality

Closeness Centrality is also based on the principle of the shortest path, as already specified in Section Centrality Measures. The intuition behind the closeness centrality measures is the following: increasing the length of the shortest path between a node and all the others, the centrality of the considered node should decrease since a node is less central when it is farther from the other knots. It is immediately noted that this type of centrality depends not only on direct arcs, but also on indirect ones, especially when two nodes are not adjacent to each other. The calculation of Closeness Centrality, previously quoted in [Formula \(9\)](#), is simply the inverse of the sum of the distances of all the shortest paths from the considered node v_i to all the other nodes accessible from it.

Example: To provide a concrete example, let us consider the node 0. By observing the values shortest paths shown in [Table 1](#), we consider those referring to paths that go from this node to all others (from 1 to 9) and then we calculate the length (e.g., $d_{0,1}=1$, $d_{0,5}=3$, etc.). By summing them we obtain the value 21 and, by its reverse, the value 0.047619 is obtained for the considered measure.

[Table 3](#) contains the values of Closeness Centrality for each node in G .

By removing the node scoring the highest value of Closeness Centrality, the network would drift, as in the previous case. More in general, by removing the best score nodes according to this measure, the graph would not always be disconnected since in this case the vertices with high centrality value have the meaning of nodes that allow the message to be spread quickly.

Table 1 All Pair Shortest Path for G

Source	Target	Path
0	1	0-1
0	2	0-2
0	3	0-2-3, 0-7-3
0	4	0-2-3-4, 0-7-3-4
0	5	0-2-3-5, 0-7-3-5
0	6	0-2-3-6, 0-7-3-6
0	7	0-7
0	8	0-2-3-8, 0-7-3-8
0	9	0-2-3-6-9, 0-7-3-6-9, 0-2-3-8-9, 0-7-3-8-9
1	2	1-2
1	3	1-7-3, 1-2-3
1	4	1-7-3-4, 1-2-3-4
1	5	1-7-3-5, 1-2-3-5
1	6	1-7-3-6, 1-2-3-6
1	7	1-7
1	8	1-7-3-8, 1-2-3-8
1	9	1-7-3-6-9, 1-2-3-6-9, 1-7-3-8-9, 1-2-3-8-9
2	3	2-3
2	4	2-3-4
2	5	2-3-5
2	6	2-3-6
2	7	2-7
2	8	2-3-8
2	9	2-3-6-9, 2-3-8-9
3	4	3-4
3	5	3-5
3	6	3-6
3	7	3-7
3	8	3-8
3	9	3-8-9, 3-6-9
4	5	4-3-5
4	6	4-3-6
4	7	4-3-7
4	8	4-3-8
4	9	4-3-8-9, 4-3-6-9
5	6	5-3-6
5	7	5-3-7
5	8	5-3-8
5	9	5-3-8-9, 5-3-6-9
6	7	6-3-7
6	8	6-8
6	9	6-9
7	8	7-3-8
7	9	7-3-6-9, 7-3-8-9
8	9	8-9

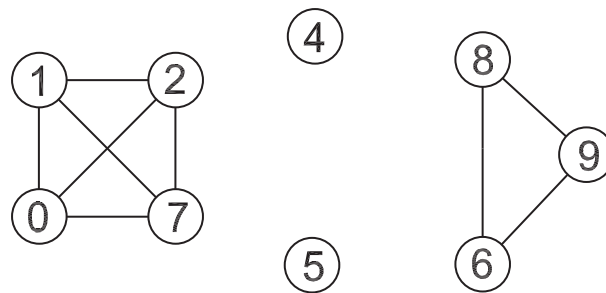
**Fig. 2** The graph G without the node scoring the highest value of Degree Centrality.

Table 2 Degree Centrality values for nodes in G

<i>Node</i>	<i>Index</i>
0	3
1	3
2	4
3	6
4	1
5	1
6	3
7	4
8	3
9	2

Table 3 Closeness Centrality values for nodes in G

<i>Node</i>	<i>Index</i>
0	0.048
1	0.048
2	0.067
3	0.083
4	0.05
5	0.05
6	0.059
7	0.067
8	0.058
9	0.042

Table 4 Betweenness Centrality values for nodes in G

<i>Node</i>	<i>Index</i>
3	27.0
2	6.0
7	6.0
6	3.5
8	3.5
0	0.0
1	0.0
4	0.0
5	0.0
9	0.0

Betweenness Centrality

Betweenness Centrality is based on the concept that a v_i node is central if it lies within several shortest paths, or when it takes a strategic role in the network acting as a link for several pairs of nodes. To calculate Betweenness Centrality we refer to the [Formula \(12\)](#), which consists of a simple summation depending on the number of *all pair shortest path* already recalled above.

The complexity of the method is in locating within the graph G all the possible shortest paths. Indeed, for each node of the graph we have to identify all the paths that hold it and sum it up (those paths that do not contain the node considered do not contribute to the summation).

Example: Consider node 6 in G and all the shortest paths between two nodes that contain it: in paths 0 – 9 and 1 – 9 it is present in two of the four possible paths, while in paths 2 – 9, 3 – 9, 4 – 9, 5 – 9 and 7 – 9 it is present in only one of the two possible paths. The result of the sum of the fractions [Formula \(11\)](#) is the value 3.5.

In [Table 4](#) the value of Betweenness Centrality for each node of the graph G is described in increasing order.

By removing from the network only the node with the highest Betweenness Centrality score, the graph would become disconnected as shown in [Fig. 2](#). Again, the node with highest centrality value is node 3.

More in general, by supposing that the communication between each pair of nodes is propagated through the shortest paths needed between them, in most cases if the highest score nodes are eliminated then the message will no longer propagate within the network.

Conclusion

In this manuscript we provided an overview of the most important centrality measures proposed in the literature and applied to biological networks analysis. We distinguished centrality measures based on walks computation from those based on shortest-paths computation. We also provided suitable examples showing how some of the considered measures can be calculated. More specifically, we considered the calculation of Degree, Closeness and Betweenness Centralities.

For all the considered measures, if the node scoring the highest value of centrality is eliminated from the network, then the network may be reduced to a disconnected graph, since that node is a *crucial* node for the links between the knots of the network. After deleting it, the values for each node will change, or most of the power values will be reduced.

However we note that, unlike the small example we proposed in which the node scoring the highest value of centrality is always the same, generally it may not happen and an interesting matter is the comparison of rankings obtained according to different centrality measures.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Quantification of Proteins from Proteomic Analysis

References

- Aho, A.V., Hopcroft, J.E., Ullman, J.D., 1974. The Design and Analysis of Computer Algorithms. Addison-Wesley Pub. Co.
- Alon, U., 2007. Network motifs: Theory and experimental approaches. *Nature Reviews Genetics* 8 (6), 450–461. [06].
- Barabási, A.-L., Pósfai, M., 2016. *Network Science*. Cambridge: Cambridge University Press.
- Bavelas, A., 2017. Communication patterns in taskoriented groups. *The Journal of the Acoustical Society of America* 22 (6), 725–730. [07/11/1950].
- Bonacich, P., 1972. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* 2 (1), 113–120.
- Borgatti, S.P., Everett, M.G., 2006. A graph-theoretic perspective on centrality. *Social Networks* 28 (4), 466–484.
- Brandes, U., Erlebach, T., 2005. *Network Analysis: Methodological Foundations* (Lecture Notes in Computer Science). New York: Springer-Verlag.
- Estrada, E., 2006. Virtual identification of essential proteins within the protein interaction network of yeast. *PROTEOMICS* 6 (1), 35–40.
- Estrada, E., 2010. Generalized walks-based centrality measures for complex biological networks. *Journal of Theoretical Biology* 263 (4), 556–565.
- Estrada, E., Rodríguez-Velázquez, J.A., 2005. Subgraph centrality in complex networks. *Physics Review E Stat. Nonlinear Soft Matter Physics* 71 (5 Pt 2), 05.
- Fell, D.A., Wagner, A., 2000. The small world of metabolism. *Nature Biotechnology* 18 (11), 1121–1122. [11].
- Ferrar, W.L., 1951. *Finite Matrices*. Clarendon Press.
- Fletcher, J.M., Wennekers, T., 2017. From structure to activity: Using centrality measures to predict neuronal activity. *International Journal of Neural Systems*. 1750013. [07/11/2016].
- Freeman, L.C., 1978. Centrality in social networks conceptual clarification. *Social Network* 1 (3), 215–239.
- Gantmacher, F.R., 1960. *The Theory of Matrices*, Number v. 1. Chelsea Pub. Co.
- Hage, P., Harary, F., 1995. Eccentricity and centrality in networks. *Social Networks* 17 (1), 57–63. [01].
- Harary, F., 1969. *Graph Theory*. Addison-Wesley Pub. Co.
- Hoede, C., 1978. A new status score for actors in a social network. Technical report, Twente University Department of Applied Mathematics.
- Hubbell, C.H., 1965. An Input-Output Approach to Clique Identification 28 (4), 377–399.
- Jeong, H., Mason, S.P., Barabási, A.L., Oltvai, Z.N., 2001. Lethality and centrality in protein networks. *Nature* 411 (6833), 41–42. [05].
- Joy, M.P., Brock, A., Ingber, D.E., Huang, S., 2005. High-betweenness proteins in the yeast protein interaction network. *Journal of Biomedicine Biotechnology* 2005 (2), 96–103.
- Koschützki, D., Schreiber, F., 2008. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene Regulation and Systems Biology* 2, 193–201.
- Lawyer, G., 2015. Understanding the influence of all nodes in a network. *Scientific Reports* 5, 8665.
- Leo, K., 1953. A new status index derived from sociometric analysis. *Psy-Chometrika* 18 (1), 39–43.
- Liu, Y.-Y., Slotine, J.-J., Barabási, A.-L., 2012. Control centrality and hierarchical structure in complex networks. *PLOS ONE* 7 (9), 59.
- Ma, X., Gao, L., 2012. Biological network analysis: Insights into structure and functions. *Briefings in Functional Genomics* 11 (6), 434.
- Ma, H.-W., Zeng, A.-P., 2003. The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics* 19 (11), 1423–1430. [07].
- Mazurie, A., Bonchev, D., Schwikowski, B., Buck, G.A., 2010. Evolution of metabolic network organization. *BMC Systems Biology* 4 (1), 59.
- Özgür, A., Vu, T., Erkan, G., Radev, D.R., 2008. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics* 24 (13), i277–i285. [07].
- Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The Pagerank Citation Ranking: Bringing Order to the Web.
- Paladugu, S.R., Zhao, S., Ray, A., Raval, A., 2008. Mining protein networks for synthetic genetic interactions. *BMC Bioinformatics* 9 (1), 426.
- Pavlopoulos, G.A., Secrier, M., Moschopoulos, C.N., *et al.*, 2011. Using graph theory to analyze biological networks. *BioData Mining* 4 (1), 10.
- Potapov, A.P., Voss, N., Sasse, N., Wingender, E., 2005. Topology of mammalian transcription networks. In: *International Conference on Genome Informatics* 16 (2), 270–278.
- Sabidussi, G., 1966. The centrality index of a graph. *Psychometrika* 31 (4), 581–603.
- Strogatz, S.H., 2001. Exploring complex networks. *Nature* 410 (6825), 268–276. [03].
- Valente, T.W., Foreman, R.K., 1998. Integration and radiality: Measuring the extent of an individual's connectedness and reachability in a network. *Social Networks*. 89–105. [01].
- Wuchty, S., Stadler, P.F., 2003. Centers of complex networks. *Journal of Theoretical Biology* 223 (1), 45–53.
- Yu, H., Kim, P.M., Sprecher, E., Trifonov, V., Gerstein, M., 2007. The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLOS Computational Biology* 3 (4), 1–8. [04].
- Zotenko, E., Mestre, J., O'Leary, D.P., Przytycka, T.M., 2008. Why do hubs in the yeast protein interaction network tend to be essential: Reexamining the connection between the network topology and essentiality. *PLOS Computational Biology* 4 (8), 1–16. [08].

Network Topology

Giuseppe Manco, Ettore Ritacco, and Massimo Guarascio, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The importance of networks lies in the simple model they represent, which applies to several real-world situations. Essentially, a network is a set of entities linked into a whole uniform structure. We can find examples of systems composed by individuals connected in some form in several situations. We can find (Newman, 2010, Section “Introduction”) lists the most common ones: in biology and bioinformatics for example, networks provide the mathematical representation of connections within biological systems (Barrat et al., 2008, ch.12). Examples of biological networks are: neural networks, connecting neurons in the brain (Bullmore and Sporns, 2009); ecological networks such as the “food web” (Dunne et al., 2002); protein interaction networks, representing interaction between proteins in a cell (Habibi et al., 2014); gene co-expression networks, connecting genes which show similar expression patterns across biological conditions (Weirauch, 2011); gene regulatory networks, describing interactions among molecular regulators (Davidson and Levin, 2005); metabolic networks, describing how the chemical compounds of a living cell are connected by biochemical reactions (Palsson, 2006).

Studying the characteristics of such networks is important because it can unveil their statistical and mathematical properties and ultimately help in making predictions about their structure and dynamics. The networks, we are considering, are complex systems, and hence understanding their topological organization allows to model their behavior.

We will focus on mathematical, statistical and computational tools for analyzing and modeling networks. We will split our dissertation into two articles: the current one will focus on defining fundamental quantitative properties characterizing and describing the underlying structure on a network; the next one is devoted to reviewing the mathematical models which explain such properties and the observations in the real world.

The content of this survey is limited due to the wide nature of the subject. The interested reader can refer to survey article by Strogatz (2001), Albert and Işıl Barabási (2002), Newman (2003) and books by Watts (1999), Dorogovtsev and Mendes (2003), Barrat et al. (2008), Kolaczyk (2009), Newman (2010), Van Mieghem (2014).

Background

We will start by fixing the mathematical notation. Networks are represented by a graph $G = \langle V, E \rangle$, defined by a set V of $N = |V|$ vertices/nodes and a set E of $L = |E|$ edges/links. The graph can be also represented through a $N \times N$ adjacency matrix A , with elements $a_{i,j}$ taking value 1 when there is an edge starting from i to j , and 0 otherwise. When $a_{i,j} = 1$, then j is denoted as an *adjacent* node of i . A *subgraph* G' of G is a graph devised by a subset of the nodes in G : formally, $G' = \langle V', E' \rangle$ where $V' \subseteq V$ and $E' \subseteq E$.

We can devise some properties which characterize a network, as exemplified in Fig. 1:

- Graphs can be *undirected*, if A is symmetrical, i.e., $a_{i,j} = a_{j,i}$ for each i, j . By converse, *directed* graphs assume that $A \neq A^T$.
- Nodes and edges can be associated with properties/attributes, such as e.g., weights. For the edges, weighted links assume the existence of a matrix $W \in \mathbb{R}^{N \times N}$ such that $w_{i,j}$ represents the weight associated with the edge from node i to j .
- Some graphs can admit *self loops*, i.e., the possibility that $a_{i,i} \neq 0$. Also, *multigraphs* admit multiple edges between the same pair of nodes. Graphs without self loops and multiple edges are called *simple graphs*.
- Finally, *hypergraphs* are special graphs where edges can involve multiple nodes. In such case, the adjacency graph can be alternatively represented as an incidence matrix between nodes and edges.

Given a node v , the *degree* K_v of the node is defined as the number of edges reaching i . For directed graphs, we can distinguish K_v^{in} and K_v^{out} , counting the edges where v is the destination or the source, respectively. Within Fig. 1(a), $K_2^{\text{in}} = 1$ and $K_2^{\text{out}} = 2$, whereas $K_2 = 3$ in Fig. 1(b). It turns out that $L = \sum_{v \in V} K_v^{\text{in}} = \sum_{v \in V} K_v^{\text{out}}$ for directed graphs, and $L = \frac{1}{2} \sum_{v \in V} K_v$ for undirected ones. When $K_v = N-1$ for each v , the graph is *complete* and $L = L_{\text{max}} = N(N-1)/2$. From an algebraic point of view, $K_v^{\text{in}} = \sum_{u \in V} A_{u,v}$ and $K_v^{\text{out}} = \sum_{u \in V} A_{v,u}$ (and $K_v = (\sum_{u \in V} A_{u,v}) = (\sum_{u \in V} A_{v,u})$ for undirected graphs). Further, $L = (\sum_{u,v \in V} A_{u,v})$ for directed graphs, and $L = \frac{1}{2} \sum_{u,v \in V} A_{u,v}$ for undirected graphs.

A *path* between u and v is a sequence of adjacent nodes $n_1 \rightarrow n_2 \dots \rightarrow n_k$ such that $n_1 = u$, $n_k = v$ and $n_i \neq n_j$ for each $i \neq j$. If such a path exists, then we say that v is *reachable* from u . Within Fig. 1(a), there's a path between 3 and 4 composed by the sequence $3 \rightarrow 1 \rightarrow 2 \rightarrow 4$ and hence 4 is reachable by 3. The converse does not hold, but it holds for the graph in Fig. 1(b). The *length* of a path is the number of edges traversed by the path. There is an interesting relationship between reachability, length and the powers of A . Clearly, adjacent nodes represent paths of length 1, and the adjacency matrix represents all those paths. It is straightforward to

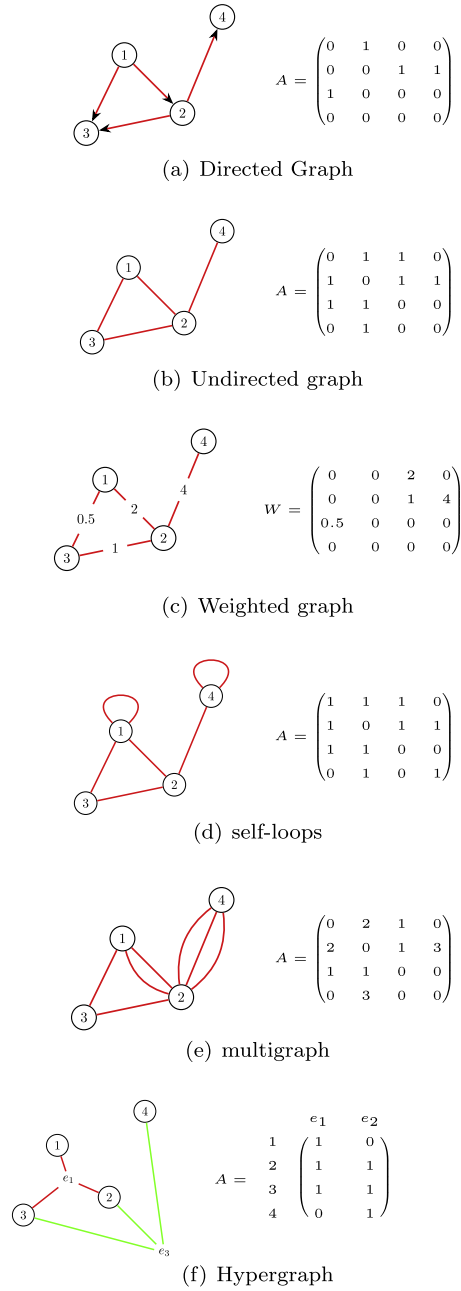


Fig. 1 Example graphs.

notice that, whenever $A_{u,w} A_{w,v} = 1$, then the path $u \rightarrow w \rightarrow v$ is observable in the network. Thus, for the graph in [Fig. 1\(a\)](#), we can observe that

$$A^2 = A \times A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Within A^2 , the cell (u, v) denotes the number of paths of length 2. In particular, node 1 can reach node 3 through the path $1 \rightarrow 2 \rightarrow 3$ and node 3 can reach node 2 through $3 \rightarrow 1 \rightarrow 2$.

Similarly, as the product $A_{u,w} A_{w,s} A_{s,v}$ denotes the existence of path $u \rightarrow w \rightarrow s \rightarrow v$, A^3 is representative of all the reachability relationships with paths of length 3:

$$A^3 = A^2 \times A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where we can observe that there are 4 paths of size 3, respectively $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$, $2 \rightarrow 3 \rightarrow 1 \rightarrow 2$, $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$, $3 \rightarrow 1 \rightarrow 2 \rightarrow 3$ and $3 \rightarrow 1 \rightarrow 2 \rightarrow 4$. In general, the cell (u, v) of A^k denotes the number of paths from u to v traversing k edges.

The distance $d_{u,v}$ between two vertices u and v is the length of the shortest path (also called *geodesic path*) between them.

Basic Tools and Methodology

The study of networks is essentially approached through graph theory (Bondy and Murty, 2008). The advent of large-scale networks has posed some problems which make the traditional approaches to graph theory inadequate to model the nature and complexity of these networks. This has shifted the focus on the analysis of large-scale statistical properties which are better suited to capture in quantitative terms the underlying organizing principles which justify the topology of the network.

When studying the topology of a network, we are mainly interested in investigating the structural properties of the underlying graph. We can summarize these properties in four essential components (Kolaczyk, 2009):

- connection characteristics among vertices;
- routes for movement of information;
- importance of vertices;
- regularities within groups of vertices.

We shall use two real networks as running examples. The first one is a relatively small transcriptional regulatory network of *E. coli* (Shen-Orr et al., 2002), consisting of 423 nodes and 518 edges. The network is represented in Fig. 2(a) and represents direct transcriptional interactions between transcription factors and the operons they regulate (an operon is a group of contiguous genes that are transcribed into a single mRNA molecule). The second dataset represent a network of yeast protein to protein interaction Sun et al. (2003) and it contains 2361 nodes (proteins) and 6646 edges (interactions), Fig. 2(b).

The literature on network science has developed some measures for quantifying network properties. These metrics summarize either local or global topological structures, and characterize either nodes and edges, or network cohesion. Simple global measures include the following:

- The sparsity coefficient

$$\rho = L/L_{max} = \frac{\kappa L}{N(N-1)}$$

where $\kappa=1$ for directed graphs, and $\kappa=2$ for undirected graphs. For the networks of Fig. 2 we have $\rho^{(ecoli)} = 5.8 \cdot 10^{-3}$ and $\rho^{(yeast)} = 2.3 \cdot 10^{-3}$.

- The average degree $\langle k \rangle$,

$$\langle k \rangle = \frac{1}{N} \sum_{v \in V} K_v$$

For directed graphs, the definition adapts to whether we are considering the node as a source or as a destination: $\langle k^{in} \rangle = \frac{1}{N} \sum_{v \in V} K_v^{in}$ and $\langle k^{out} \rangle = \frac{1}{N} \sum_{v \in V} K_v^{out}$. We can easily observe that $\langle k \rangle = \kappa L/N$. It is also worth observing (Newman, 2003) that the definition can be adapted to consider situations where there are nodes which are not connected. For the example networks, we have $\langle k \rangle^{(ecoli)} = 2.45$ and $\langle k \rangle^{(yeast)} = 5.63$.

- The average distance ℓ ,

$$\ell = \frac{1}{N(N-1)} \sum_{\substack{u, v \in V \\ u \neq v}} d_{u,v}$$

The contribution of unconnected nodes is not considered here. For the example networks, we have $\ell^{(ecoli)} = 4.82$ and $\ell^{(yeast)} = 4.38$.

- The size $d = \max_{u, v \in V} d_{u,v}$ of the longest distance in G , denoted as the *diameter* of G .

Among the basic statistics, we can also devise a census of some specific components which characterize the graph. Typical components are *dyads* (pairs of nodes) and *triads*. Fig. 3 shows all the possible dyads and triads which can occur in a network. For example, the dyad census on the *ecoli* network comprises 518 mutual dyads and 88,735 null dyads.

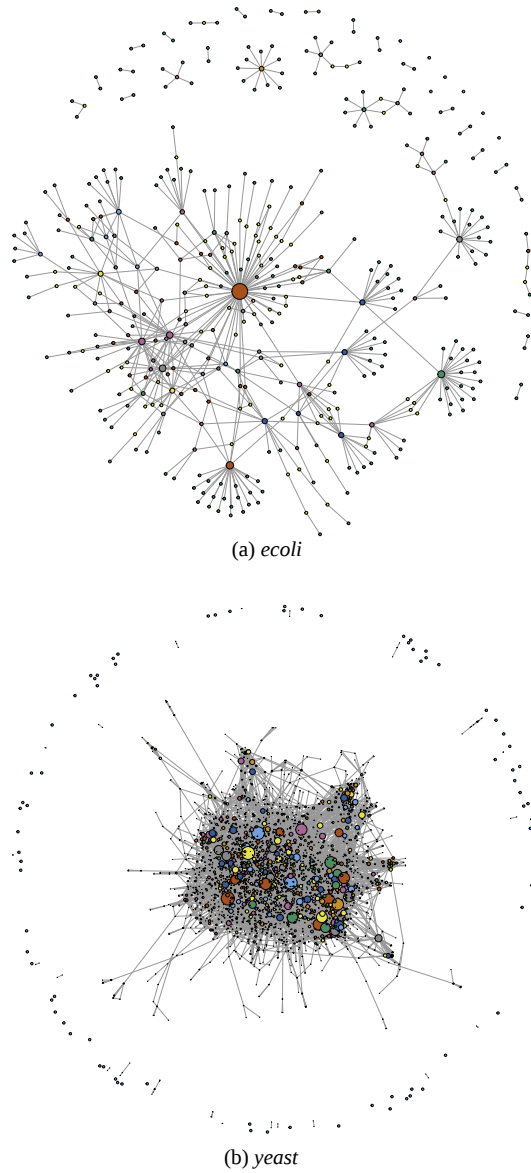


Fig. 2 Example biological networks (node size and color is displayed according to node degree).

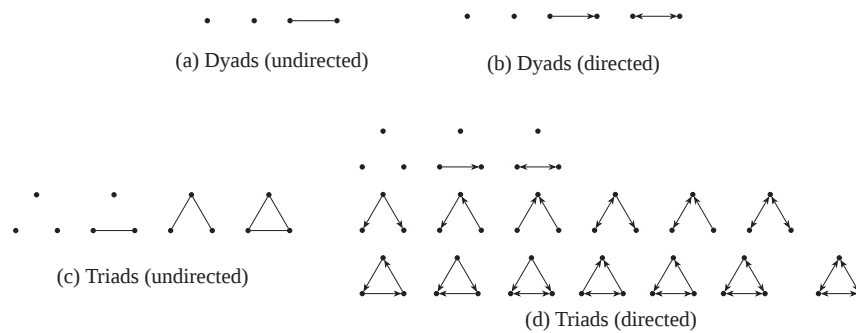


Fig. 3 Dyad and Triad census.

An essential tool in the analysis of the topology of a network is the concept of *centrality*, as a measure of the structural importance of an element in the network (Koschutzki *et al.*, 2005). The intuition about a centrality measure is to denote a rank about the importance of vertices or edges of a graph by assigning real values to them.

The simplest form of centrality is *degree centrality* $C_d(v)$, representing the potential of communication among nodes:

$$C_d(v) = \frac{K_v}{L-1}$$

The degree centrality is a local measure, because the centrality value of a node is only determined by the number of its neighbors. In directed networks, two variants can be devised, depending on whether we consider incoming or outgoing nodes.

A natural extension of degree centrality is the *eigenvector centrality* $C_e(v)$, which in a sense weights the importance of connections. It can be defined recursively as the sum of the eigenvector centrality of v 's neighbors:

$$C_e(v) = \frac{1}{\lambda} \sum_{u \in V} A_{u,v} C_e(u)$$

where λ is a normalizing constant. Written in matrix notation, C_e is the solution of the equation $AX = \lambda X$, which suggests that x is an eigenvector of A and λ is the corresponding eigenvalue. Typically, λ is chosen to be the largest eigenvalue of A (Bonacich, 1987). There are several variants of eigenvector, for instance PageRank (Brin and Page, 1998) or hub/authority (Kleinberg, 1999).

Other measures focus on on shortest paths, measuring, e.g., the average distance from other vertices, or the ratio of shortest paths a vertex lies on. These measures give information about the global network structure. *Closeness* centrality represents the potential of independent communication:

$$C_c(v) = \frac{1}{\sum_{\substack{u \in V \\ u \neq v}} d_{u,v}}$$

Nodes with high closeness are considered structurally important, since they can easily reach (or be reached) by other nodes.

An alternative concept is based on the idea that a node is capable of controlling the connections. *Betweenness* centrality measures the extent to which a node is crucial for the connection between any two nodes.

$$C_b(v) = \sum_{\substack{s, t \in V \\ s \neq t \neq v}} \frac{N_{s,t|v}}{N_{s,t}}$$

where $N_{s,t}$ represents the number of shortest paths between s and t , and $N_{s,t|v}$ the number of shortest paths passing through v . The ratio can be interpreted as the probability v is involved into any communication between s and t .

The so far described centrality measures on our two running examples are shown in Fig. 4.

Closeness and betweenness are particularly sensitive to large-scale networks. Their computation is tractable in the conventional sense, i.e., there exist polynomial time and space algorithms able to compute their exact values. Nevertheless, the exact centrality index computation is not practical for graphs with a large number of nodes: real scenarios can reach thousands, millions, even billions of nodes. Recently, approximation algorithms have been investigated (Riondato and Kornaropoulos, 2016; Brandes and Pich, 2007) that trade off accuracy for speed and efficiently compute high-quality approximations of the centrality values.

Assessment of Structural Properties

In this section we consider some basic properties of networks. There are essentially three concepts which characterize complex networks.

Degree Distribution

Recalling that the degree K_v of a node v in a network $G = \langle V, E \rangle$ is the number of nodes incident upon v in E , it is useful to consider the whole set $\{K_v\}_{v \in V}$ and investigate its properties. We can denote by $p_k = \Pr(K_v = k | v \in V)$ the probability that a randomly chosen node in V exhibits degree k . Fig. 5 shows the frequency histograms of node degrees for both *ecoli* and *yeast*.

The degree distribution provides a natural summary of the connectivity within the graph, and it becomes an interesting metric especially for large graphs. Typically, in big networks, the degree distribution of the vertices is right skewed: that is it has a long tail on the values above the mean. Fig. 6 shows the degree distribution for the two datasets, in log-log scale. For the *yeast* dataset, the node degrees range between 0 and 64, and the tail of the distribution is notably noisy, due to the skews on large degrees. We can also observe that the majority of vertices have low degree, and the density tends to decrease exponentially as long as the degree increases.

Such an exponential behavior is typical in large scale networks, and typically the distribution can be characterized as a *power-law* distribution:

$$p_k \propto k^{-\alpha}$$

The log-log plotting has the nice property of highlighting the linear relationship $\log p_k \propto -\alpha \log k$ which derives directly from the functional form of the distribution. Power law distribution has the property that their cumulative counterpart $P_k = \sum_{x=k}^{\infty} p_k$ is

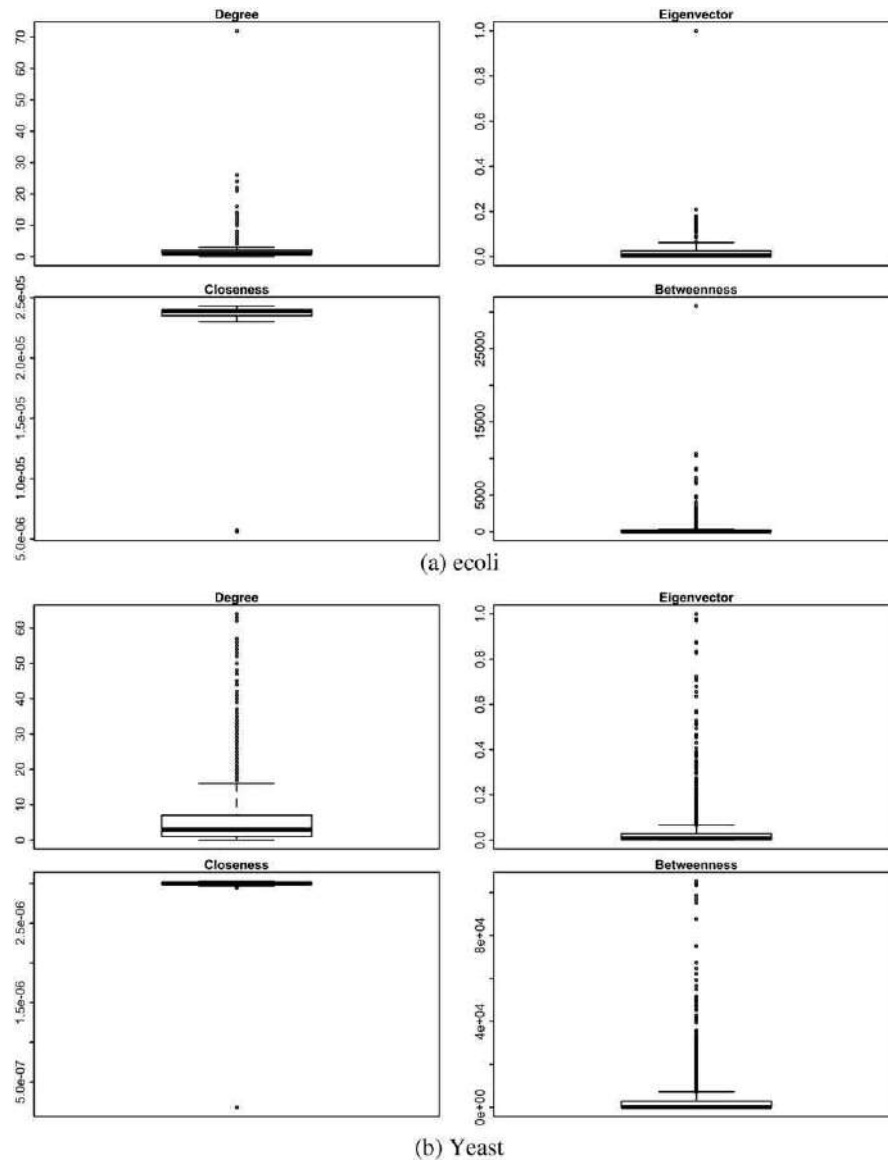


Fig. 4 Distribution of centrality measures.

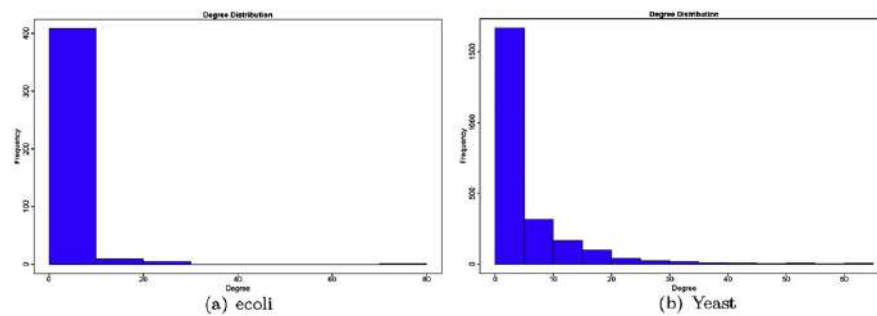


Fig. 5 Frequency histogram for K in both *ecoli* and *yeast*.

also a power law, with functional form $P_k \propto k^{-\alpha+1}$. Fig. 7(a) shows the fitting of the cumulative distribution for the *coli* network. The fitting holds for $\alpha = 2.39$.

Networks with power-law degree distributions are called *scale-free* networks and have been extensively studied (see e.g., Newman (2005)). There are, however, other common long-tailed functional forms for the degree distributions, such as exponential

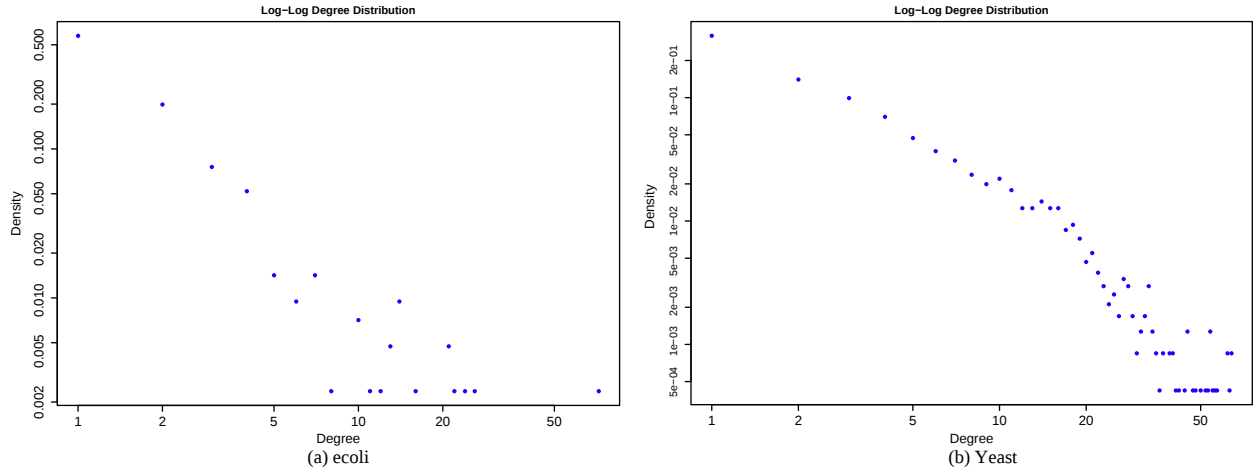


Fig. 6 Degree distribution (in log-log scale).

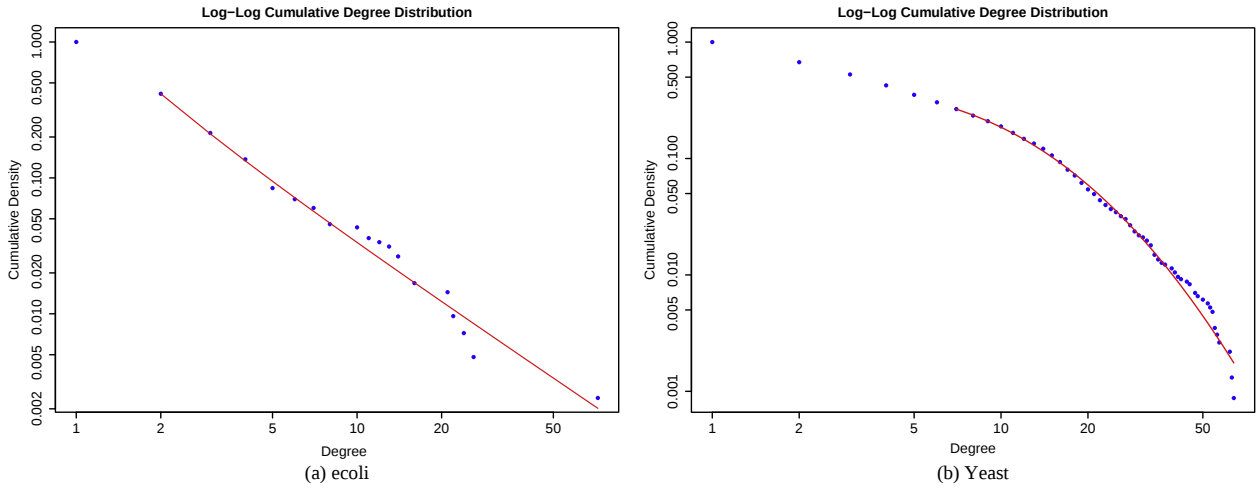


Fig. 7 Cumulative degree distributions and their fits.

or lognormal. **Fig. 7(b)** shows that the *yeast* dataset does not fit a power law: within the graph, the red line shows a lognormal fitting. More complicated models can even exhibit combined distributions, such as mixtures of power-laws ($p_k \propto \sum_k \beta_k k^{z_k}$) or power-laws with exponential truncation ($p_k \propto k^z e^{-\beta k}$).

Small World

The connectivity of a network is usually described by means of paths and connected components. A *connected component* is a maximal subset $S \subseteq V$ where there is a path between any pair of nodes in S . The connected components represent the subgroups.

There is a typical situation which can be observed in the topology of large-scale networks. Notably, a network is characterized by a component which includes a large amount of nodes (the *giant component*), while the rest of the network is made of small components disconnected from each other. This is clearly visible in the networks of **Fig. 2**.

Another feature which characterizes large-scale networks is the small-world phenomenon, i.e., the fact that distances are typically small and consequently ℓ is small as well. For the graphs in **Fig. 2**, we can observe $\ell_{\text{ecoli}} \approx 5$ and $\ell_{\text{yeast}} \approx 4$. Also, **Fig. 8** shows the frequency distribution of the distances in the two graphs. As we shall formally see in a next article, this phenomenon can be mathematically characterized as $\ell \leq C \log N$ for some constant C , as a consequence from the fact that the number of connected nodes grows exponentially as long as the distance grows.

There is a connection among the heavy-tailed degree distribution, centrality of nodes and the small world effect. The fact that there exist few nodes with high connectivity eases the walk-through the network, thus making the geodesic paths shorter. In fact, central nodes are more likely to appear in shortest paths within the network than less important nodes.

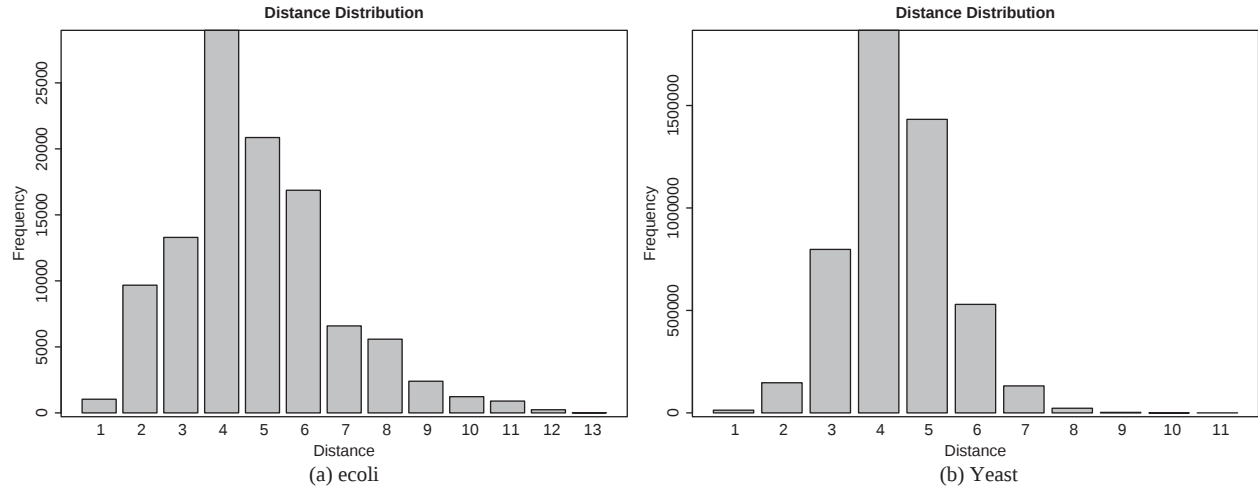


Fig. 8 Frequencies of all distances.

Network Cohesion

Another element which characterizes the topology of a network is the cohesion, representing the characterization of the local neighborhood. We have already seen that large-scale networks tend to have a relatively low degree. The question that raises is then: do nodes belong to cliques? A *clique* is a complete subgraph.

There are many ways to quantify network cohesion. However, a common measure is transitivity (or *clustering coefficient*), i.e., the relative number of closed triangles (triads). That is, if vertex u is connected to vertex v , and v is connected to w , then u is also connected to w . Measuring the extent at which this relationship holds provides as with a way of measuring cohesion. Thus, formally the clustering coefficient of a node v can be defined as:

$$C(v) = \frac{2e_v}{K_v(K_v - 1)}$$

where e_v is the number of edges connecting the neighbors of v (for example, $e_v = \frac{1}{2} \sum_{u,w} A_{v,u} A_{v,w} A_{u,w}$ on an undirected graph). The overall clustering coefficient C can hence be devised as the average of $C(v)$ for all v .

Most, large scale networks exhibit a relatively high clustering coefficient: a clear sign that transitive closure holds on a relevant part of the graph. For example, for the networks in Fig. 2, we have $C^{(ecoli)} = 0.02$ and $C^{(yeast)} = 0.10$. To see why these numbers are high, we can compare them to the totally random situation where the number of neighbors is constant and these neighbors are randomly chosen. Then, given two neighbors u, w of v , the probability of them being incident would be proportional to N^{-1} . In the *yeast* dataset, this would correspond to ~ 0.0003 , which is way lower than the actual $C^{(yeast)}$.

Assortativity and Degree Correlation

A further direction worth investigating when analyzing the topology of a large scale network, is to check for *assortativity*, i.e., the tendency of nodes to exhibit a kind of homophily according to a given property. In other words, assortativity measures a non-trivial level of affinity, projected on some features, of connected nodes. For example, in social networks it is common to associate with people who share the same interests or come from the same local community. A high level of assortativity shows a positive correlation between the shared features and the nodes connections (phenomenon called *assortative mixing*), while a low level indicates a negative correlation, i.e., connections are built by nodes which exhibit strong differences in terms of the chosen features (triggering a *dissortative mixing*). Building an *assortativity coefficient* requires several steps (Newman, 2010). First of all, we need to count the edges whose nodes share a feature f :

$$\sum_{edges(u,v)} sim_f(u,v) = \frac{1}{2} \sum_{u,v} A_{u,v} sim_f(uv) \quad (1)$$

where $\frac{1}{2}$ normalizes the double nature of each edge and $sim(u,v) \in [0,1]$ is a similarity function which measures the closeness of nodes u and v according to the feature f . For example, sim can be the Kronecker delta (for discrete features) or the normalized difference (for scalar values).

This quantity does not avoid trivial phenomena, for instance if all the nodes share only one feature its value is maximized, but there is no useful information in it. To balance this issue, we can remove trivial connections generated by a random mechanism, keeping only those connections that are beyond our expectation. The expected number of trivial random affinities is:

$$\frac{1}{2} \sum_{u,v} \frac{K_u K_v}{2L} \text{sim}_f(u, v) \quad (2)$$

where $2L$ is the number of involved nodes for all the edges and $\frac{K_u K_v}{2L}$ is the expected number of edges between nodes u and v . Now, we can define the *modularity* Q of a network as the difference of Eqs. (1) and (2):

$$Q = \frac{1}{2L} \sum_{u,v} \left(A_{u,v} - \frac{K_u K_v}{2L} \right) \text{sim}_f(u, v) \quad (3)$$

The last step is the normalization of the modularity in order to obtain the assortativity coefficient P . The normalization depends on the maximum value of Q , that is reached with a perfect assortative mixed network, where $\text{sim}_f(u, v) = 1$ when $A_{u,v} = 1$:

$$\begin{aligned} P = \frac{Q}{Q_{\max}} &= \frac{\frac{1}{2L} \sum_{u,v} \left(A_{u,v} - \frac{K_u K_v}{2L} \right) \text{sim}_f(u, v)}{\frac{1}{2L} \left(2L - \sum_{u,v} \frac{K_u K_v}{2L} \right) \text{sim}_f(u, v)} \\ &= \frac{\sum_{u,v} \left(A_{u,v} - \frac{K_u K_v}{2L} \right) \text{sim}_f(u, v)}{2L - \sum_{u,v} \frac{K_u K_v}{2L} \text{sim}_f(u, v)} \end{aligned} \quad (4)$$

The so defined assortativity is able to express if connected nodes exhibit a certain level of interest for the same features; an interesting evolution of this concept involves the possibility of measuring if different feature values are correlated among the nodes. Under this perspective nodes with some features try to connect to other nodes which are complementary or simply different: as instance let's think about sentimental relationships among users of a social network, an individual is searching for her better half that may complete her.

Considering assortative mixing characterized by scalar features, we can exploit the covariance concept to formalize the multi-value assortativity:

$$\begin{aligned} \text{cov}(x_u, x_v) &= \frac{\sum_{u,v} A_{u,v} (x_u - \mu)(x_v - \mu)}{\sum_{u,v} A_{u,v}} \\ &= \frac{1}{2L} \sum_{u,v} \left(A_{u,v} - \frac{K_u K_v}{2L} \right) x_u x_v \end{aligned} \quad (5)$$

where x_u (resp. x_v) is the value of the node u (resp. v) for the target feature, and μ the expected value computed as:

$$\mu = \frac{\sum_{u,v} A_{u,v} x_u}{\sum_{u,v} A_{u,v}} = \frac{1}{2L} \sum_u K_u x_u$$

It's simple to notice that Eq. (5) is very similar to Eq. (3) with the product $x_u x_v$ replacing $\text{sim}_f(u, v)$. Hence, its normalization, with the perfect mixing assumption, produces a multi-value assortativity coefficient:

$$P = \frac{\sum_{u,v} \left(A_{u,v} - \frac{K_u K_v}{2L} \right) x_u x_v}{\sum_{u,v} \left(K_u \delta_{u,v} - \frac{K_u K_v}{2L} \right) x_u x_v} \quad (6)$$

where $\delta_{u,v}$ is the Kronecker delta, which is 1 if $u=v$ and 0 otherwise.

A special case of this coefficient is the *mixing by degree*, for studying the *correlation degree*. In a network, the high-degree (resp. low-degree) nodes represent a strongly (resp. poorly) connected component. Since degree is a scalar feature, the mixing by degree multi-value assortativity coefficient can be computed as:

$$P = \frac{\sum_{u,v} \left(A_{u,v} - \frac{K_u K_v}{2L} \right) K_u K_v}{\sum_{u,v} \left(K_u \delta_{u,v} - \frac{K_u K_v}{2L} \right) K_u K_v} \quad (7)$$

where $K_u K_v$ replaced $x_u x_v$ in Eq. (6).

One strong advantage of this formulation is that it requires only to observe the structure of the network and no other information about nodes and edges is needed.

Groups of Vertices

In many real contexts, networks build up in a bottom-up process: single elements (the nodes) group together forming links (edges) according to domain-dependent dynamics and constraints. These groups, actually subgraphs, clusters or communities of nodes, in turn, can join into macro groups in a recursive process that leads to the building of the whole network. Groups can be classified into two categories:

- *Hard clustering*. Nodes are acknowledged to belong to only one group and the group set represents a partition of the node space.
- *Soft clustering*. In this case there is no observable strict partition of the graph, but we can distinguish two scenarios:
 - *Probabilistic clustering*. Nodes are characterized by a probability distribution of belonging to the groups of a finite group set. There is no evidence of a observable membership of the target node to a specific group: this membership is implicit.

- *Fuzzy clustering*. Nodes can simultaneously belong to different groups, i.e., groups can be overlapping. Typically the membership is equipped with a fuzziness score that represents the node's affiliation strength to the group.

In the network analysis research field the discovery of these groups is called *community detection*, which consists in the definition of methodologies and techniques able to locate dense connected (potentially overlapping) components that have low connection with the rest of the network. Detecting communities is a challenging and hard task, since it's difficult to understand what *intra-cluster high connection* and *inter-cluster low connection* mean, and since the number of communities is typically a priori unknown.

In the case of the hard clustering, an interesting approach, that addresses these problems, is to partition the graph where *the observed connection level is lower than the expected one*, in order to find non-trivial separations. This intuition is strictly linked with the *modularity* described in Section Assortativity and Degree Correlation. In fact, modularity measures how many edges fall beyond the expectation between vertices of the same type, see Eq. (3). If we consider as “type” the membership to a community, then connected portions of the network, exhibiting a high modularity degree, actually represent clusters, and a way to detect communities is to look for the partition that generates the highest modularity scores in the network. Unfortunately, modularity maximization is a NP-hard problem (Thai, 2008), therefore, there is space, in this research field, for the definition of heuristic algorithms able to compute a reasonably good approximation of the maximum modularity. A very simple greedy heuristic can be a variant of the (Kernighan and Lin, 1970) algorithm. The technique initially randomly separates the network into two communities, then moves each node from one cluster to the other trying to gain in terms of modularity above a certain threshold. When the algorithm reaches the convergence on the current partition, recursively will focus on the two formed communities. This process will be repeated until no other community can be split.

Closing Remarks

In this article we suggested that the network model can conveniently describe many real world processes. We defined the network model as a graph and provide different metrics to understand and measure features and topology of large-scale graphs. In particular we defined: (i) the centrality measures of the nodes as a relevance score; (ii) the node degree distribution as a tool able to catch the connectivity behavior of the networks; (iii) the node distance, i.e., the capability of information transmission; (iv) the assortativity providing information about nodes' homophily; and (v) communities, groups of strictly connected nodes.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Quantification of Proteins from Proteomic Analysis

References

- Albert, R., Iszl Barabasi, A., 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 47–97.
- Barrat, A., Barthelemy, M., Vespignani, A., 2008. *Dynamical Processes on Complex Networks*. Cambridge: Cambridge University Press.
- Bonacich, P.F., 1987. Power and centrality: A family of measure. *American Journal of Sociology* 92, 1170–1182.
- Bondy, A., Murty, M.R., 2008. *Graph Theory*. London: Springer-Verlag.
- Brandes, U., Pich, C., 2007. Centrality estimation in large networks. *International Journal of Bifurcation Chaos* 2303.
- Brin, S., Page, L., 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks* 30, 107–117.
- Bullmore, E., Sporns, O., 2009. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience* 10, 186–198.
- Davidson, E., Levin, M., 2005. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America* 102, 4935. Available at: <http://www.pnas.org/content/102/14/4935.full.pdf>.
- Dorogovtsev, S.N., Mendes, J.F.F., 2003. *Evolution of Networks, From Biological Nets to the Internet and WWW*. Oxford: Oxford University Press.
- Dunne, J.A., Williams, R.J., Martinez, N.D., 2002. Network structure and biodiversity loss in food webs: Robustness increases with connectance. *Ecology Letters* 5, 558–567.
- Habibi, I., Emami, E.S., Abdi, A., 2014. Quantitative analysis of intracellular communication and signaling errors in signaling networks. *BMC Systems Biology* 8, 89.
- Kernighan, B., Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. *The Bell Systems Technical Journal* 49, 291–307.
- Kleinberg, J., 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46, 604–632.
- Kolaczyk, E.D., 2009. *Statistical Analysis of Network Data*. Berlin: Springer.
- Koschutski, D., et al., 2005. Centrality indices. In: Brandes, U., Erlebach, T. (Eds.), *Network Analysis*. Berlin Heidelberg: Springer, pp. 16–61. (vol. 3418 of *Lecture Notes in Computer Science*).
- Newman, M., 2005. Power laws, pareto distributions and zipf's law. *Contemporary Physics* 46, 323–351.
- Newman, M.E.J., 2003. The structure and function of complex networks. *SIAM Review* 45, 167–256.
- Newman, M.E.J., 2010. *Networks: An Introduction*. Oxford: Oxford University Press.
- Palsson, B.O., 2006. *Systems Biology, Properties of Reconstructed Networks*. Cambridge University Press.
- Riondato, M., Kornaropoulos, E.M., 2016. Fast approximation of betweenness centrality through sampling. *Data Mining and Knowledge Discovery* 30, 438–475.
- Shen-Orr, S., Milo, R., Mangan, S., Alon, U., 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics* 31, 64–68.
- Strogatz, S.H., 2001. Exploring complex networks. *Nature* 410, 268–276.
- Sun, S., et al., 2003. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research* 31.
- Thai, M.T., 2008. Modularity maximization in complex networks. *Encyclopedia of Algorithms* 1–4.
- Van Mieghem, P., 2014. *Performance Analysis of Complex Networks and Systems*. Cambridge University Press.
- Watts, D.J., 1999. *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press.
- Weirauch, M.T., 2011. Gene Coexpression Networks for the Analysis of DNA Microarray Data. *Wiley-VCH Verlag GmbH & Co. KGaA*. pp. 215–250. (chapter 11).

Network Models

Massimo Guarascio, Giuseppe Manco, and Ettore Ritacco, ICAR-CNR, Rende, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Networks Science represents a range of techniques and methods for studying large-scale graph-based structures, in terms of the connections established among their entities. These methods span in several application scenarios and allow to understand, explain, describe, and predict the network characteristics. Models for link prediction between unconnected nodes, community/sub-group detection and discovering of interesting structural properties of the network are just some examples.

Understanding the formation and evolution of these networks is a relevant topic for several application scenarios (e.g., *sociology, security, marketing, and human-computer interaction*). In bioinformatics, Network Science is used to study complex biological systems. These systems are dynamic and tend to a quantitative balance between components which often determines biological outcome. Thus, the study of the structural properties can help to understand the cellular behavior from a systems perspective (Eisenberg *et al.*, 2000). In a previous article (Manco *et al.*, 2018), we already listed example biological networks that we recall here: Neural networks (Bullmore and Sporns, 2009); ecological networks (Dunne *et al.*, 2002); protein interaction networks (Habibi *et al.*, 2014); gene co-expression networks (Weirauch, 2011); gene regulatory networks (Davidson and Levin, 2005); metabolic networks (Palsson, 2006).

We also discussed some measurements which characterize essential structural properties of large networks, such as degree distribution, node centrality, network diameter and clustering coefficient. It is natural to ask whether these structural properties are the outcome of the compliance of a network to some mathematical model. Following Kolaczyk (2009), we can roughly describe a network model as a parameter set θ that allows to express the likelihood $\Pr(G|\theta)$ of observing the characteristics of the network G . The choice, of which aspects to model through $\Pr(\cdot)$, expresses the main differences among the various models. For example, the modeling can focus on the static network structures or alternatively on the dynamic processes underlying the network. Besides the justification of static or dynamic properties, network models can also be used for testing purposes. For example, for detecting whether a network significantly differs from a “null” model where it’s not expected to observe specific structural features.

Historically, a reference model was introduced in 1959 by Paul Erdős and Alfred Rényi, where all graphs on a fixed vertex set with a fixed number of edges are equally likely. Starting from there, several forthcoming studies assessed the properties of real networks into formal mathematical models capable of explaining and justifying e.g., “small-world” and scale-free paradigms.

The notion of network model can also be considered in a broader sense: That is, we are also willing to consider models which, although not capable of expressing global properties of the underlying network in a unified way, are nevertheless capable of devising predictive capabilities for specific aspects, such as the formation of a link or the semantic characterization of a node. These models represent the realization of predictive modeling typical of machine learning, in the scenario of network science.

This article is aimed at progressing from the description in Manco *et al.* (2018) by introducing some important reference network models. We classify the models as either descriptive or predictive. The former are aimed at providing a detailed description of the graph or some of its characteristics. For example, models which justify structural features such as degree distribution or the community structure of the network. By contrast, predictive models are aimed at injecting typical predictive modeling tasks such as classification or regression in the context of network science. Thus, the focus in this respect is on models for link prediction or graph classification.

Once again, there is a wide amount of literature in the subject, and the scope of this survey is limited to highlight the key concepts and guide the interested reader to further deepening on the subject. Some reference articles (Albert and Barabasi, 2002; Newman, 2003; Loscalzo and Yu, 2008) and books (Dehmer, 2010; Watts, 1999; Dorogovtsev and Mendes, 2003; Barrat *et al.*, 2008; Kolaczyk, 2009; Newman, 2010; Van Mieghem, 2014) can help in this context.

In the following we adopt the same notation introduced by Manco *et al.* (2018). Just as reminder, we highlight that a network can be represented by a graph $G=(V, E)$, defined by a set V of $N=|V|$ vertices/nodes and a set E of $L=|E|$ edges/links. The graph can be also represented through a $N \times N$ adjacency matrix A , with elements $a_{i,j}$ taking value 1 when there is an edge starting from i to j , and 0 otherwise. When $a_{i,j}=1$, then j is denoted as an *adjacent* node of i . Given a node v , we denote by K_v its degree, and by $Neighbors(v)$ the set of its neighbors.

Descriptive Models

In this section we analyze mathematical models for explaining *structural properties* and *community or subgroup structures*. Structural Analysis of the network aims at understanding how complex networks take place and evolve over time. Typically, the analysis concerns three main aspects: topology of networks, network growth and clustering and partitioning.

Network Models

The simplest mathematical model that we can expect for a network is the the Erdős-Rényi *Random Graph* (Erdős and Rényi, 1959). In this model, the basic assumption is that, by fixing the number N of nodes and the L of edges, edges are randomly placed

between nodes. Specifically, $G(N, L)$, defines an undirected graph involving N nodes and a fixed number of edges, L , chosen randomly from the $\binom{N}{2}$ possible edges in the graph.

An alternative characterization $G(N, p)$ of a random graph with N nodes considers a fixed probability p of observing a link between two nodes, and models the formation of a network as a stochastic process where edges are sampled from all possible pairs according to a Bernoulli trial governed by p . Thus, $G(N, p)$ represents a random graph governed by p where the probability of observing a graph G with L edges is given by a binomial distribution:

$$\Pr(G(N, p) \text{ has } L \text{ edges}) = \binom{\binom{N}{2}}{L} p^L (1-p)^{\binom{N}{2}-L}.$$

The term $p^L (1-p)^{\binom{N}{2}-L}$ represents the probability of observing a specific graph with L nodes. Alternatively, this probability can be specified in terms of the adjacency matrix A :

$$\Pr(A|p) = \prod_{i \neq j} p^{a_{ij}} (1-p)^{1-a_{ij}}$$

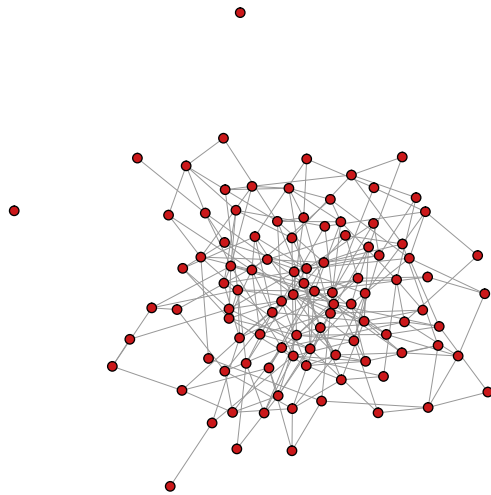
The $\binom{\binom{N}{2}}{L}$ coefficient generalizes over all possible configurations with L nodes. **Fig. 1** shows an example random graph and the underlying probability distribution. Notice that the two representations $G(N, L)$ and $G(N, p)$ are equivalent since, within a graph G with exactly L randomly and uniformly chosen edges, the probability of observing an edge is given by $L / \binom{N}{2}$, whereas a random graph with bernoulli parameter p has an expected number of edges given by $L = p \binom{N}{2}$. The latter comes directly from the properties of the binomial distribution.

The important consequences of this extremely simple model is that the most important structural properties can be devised and characterized mathematically.

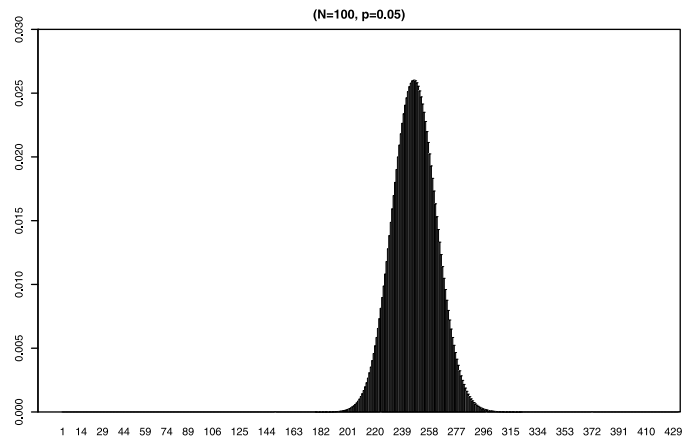
- The degree distribution is still characterized as a binomial distribution parameterized by p . In fact, a node has the same probability p to connect to any of the other $N-1$ nodes in the network. Thus, the probability $P(k)$ of observing a node with degree k is given by

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

As a consequence, the mean degree $\langle k \rangle$ corresponds to the mean value of the binomial distribution: $\langle k \rangle = p(N-1)$. Alternatively, we can assume the mean degree as a constant and devise p from that: $p = \langle k \rangle / (N-1)$.



(a) An example graph with 243 edges.



(b) Graph Distribution. The distribution is centered on $pN(N-1)/2 \approx 247$.

Fig. 1 Random Graphs with $N=100$ and $p=0.05$.

- The clustering coefficient $C(v)$, for a node v , can be estimated by resorting to the underlying distribution. Recall that we have that $C(v) = 2e_v / (K_v(K_v - 1))$ where e_v is the number of connections between the nodes in $\text{Neighbors}(v)$ (Manco et al., 2018). However, in a random graph the expected value for e_v by $E[e_v] = pK_v(K_v - 1)/2$. As a consequence, the expected clustering coefficient is inversely proportional to the number of nodes:

$$C(v) = p = \frac{\langle k \rangle}{N - 1}$$

- It is possible to correlate the presence of the *giant connected component* to the mean degree $\langle k \rangle$. In particular, when $\langle k \rangle < 1$ the graph does not exhibit a giant component, which by the converse can be measured when $\langle k \rangle \geq 1$.
- Since each node can reach on average $\langle k \rangle$ nodes in one step, the average number of nodes that can be reached in d step is given by the geometric series $\sum_{i=0}^d \langle k \rangle^i \approx \langle k \rangle^d$. Hence, a node can reach all the other nodes when $N \approx \langle k \rangle^d$, or equivalently, when $d \approx \frac{\log N}{\log \langle k \rangle}$. This allows to estimate the diameter in terms of the number of nodes.

The shortcomings of the random graph model should be clear so far. Despite the fact that the diameter can be expressed in logarithmic scale of the number of nodes, neither the clustering coefficient nor the degree distribution are realistic when compared to real networks.

Concerning the clustering coefficient, the main problem is the inverse proportion to the number of edges. For example, with reference to the *yeast* network described in Manco et al. (2018), we can observe that $C^{\text{yeast}} = 0.10$ whereas the random graph models would predict $C^{\text{yeast}} = 0.002$. In general, since the mean degree $\langle k \rangle$ can be considered a constant, a random graph generates very small and unrealistic clustering coefficients. This is somehow the effect of mathematical model underlying the probability of a link, which on one side allows for the exponential expansion of the random graph (and as a consequence enables a low diameter), but at the same time does not express any preference for local nodes: Each link is equally likely, whereas instead a small-world model would require that links within a neighborhood should be more probable.

The *Small-World* model (Watts and Strogatz, 1998) is an attempt to cope with this issue, by interpolating between an ordered finite-dimensional lattice and a random graph in order to produce local clustering and triadic closures. The basic intuition is given by the fact that a lattice structure, like the one depicted in Fig. 2(a), exhibits the exact opposite property of a random graph. In fact, in a ring we can establish a priori the number of connections along the ring. The figure shows a ring where each node v has $K_v = 6$. The clustering coefficient turns out to be a constant in this model, since each node exhibits the exact number of neighbors and the number of shared neighbors is fixed as well. For example, for the network in figure we can observe $C(v) = 0.6$ for each v . By converse, average distance ($\ell = 5$ in this case) is affected by the fact that reaching a node requires traversing the ring with a number of hops which is linear in the number N of nodes.

It turns out (Bollobás and Chung, 1988) that adding random edges to a ring can drastically reduces the diameter of the network. In fact, random edges reduce the distances by enabling shortcuts in the paths in the graph. The idea in Small World is to randomly rewiring edges according to a given probability p . The latter can counterbalance the two properties: the lower the value of p , the higher is the clustering coefficient. As a side effect, however, the diameter will result prohibitively high. As p goes from 0 to 1, the construction moves toward a random graph, since the number of edges which are randomly rewired increases. Fig. 2(b),(c) and (d) show how the ring of Fig. 2(a) evolves for increasing values of p . For large graphs, (Watts and Strogatz, 1998) found out that the interval $0.001 < p < 0.01$ guarantees graphs with realistic clustering coefficients and diameters.

The small world model admits several variations: for example in multiple dimensions representing, e.g., geographic distances (Kleinberg, 2000). Still, the degree distribution in this model resembles the binomial distribution of the random graph. The point is that both random graph and small world models assume that the degrees are likely to approach a mean value, and any deviation from this value happens with lower probability. By contrast, real networks exhibit a scale-free degree distribution: Several nodes have low degrees, and higher degrees occur with decreasing probability. The result is that the network exhibit few “hubs”, i.e., node collecting several connections, and the majority of the nodes is instead linked to these hubs. This characterization allows for a viable explanation of the low diameter, since hubs allow shorter connections of peripheral nodes.

A characterization of these properties was given by Barabasi and Albert (1999) in a model which describes network evolution and is called *Preferential Attachment*. The idea behind preferential attachment, originally proposed by Price (1976) is that, whenever a new node is generated in the network, it links to another node with probability that is proportional to its in-degree. The model was proposed for citation networks, where a new node represents a new article and links represents citations. Within the preferential attachment, the citations provided by a new article point to articles which already exhibit a high number of citations.

Mathematically, we can devise a generative process which, starting from an initial graph $G^{(0)}$ exhibiting $N^{(0)}$ nodes and $L^{(0)}$ edges, progressively adds a node v_t for $t \in \{0, 1, 2, \dots\}$. At each subsequent time step t , the node v_t is connected to m nodes already present in $G^{(t)}$, to generate a new graph $G^{(t+1)}$ comprising v_t and the m edges. The probability that the new node is connected to an existing node is proportional to the degree of the latter. In other words, the new node picks m nodes out of the existing network according to the multinomial distribution:

$$p_u = \frac{K_u}{\sum_u K_u}$$

Barabasi and Albert (1999) show that this simple model produces a network where the degree distribution follows a power-law with exponent $\alpha \approx 2.9$. Fig. 3 shows an example network generated according to the above scheme. We can devise, as expected, a

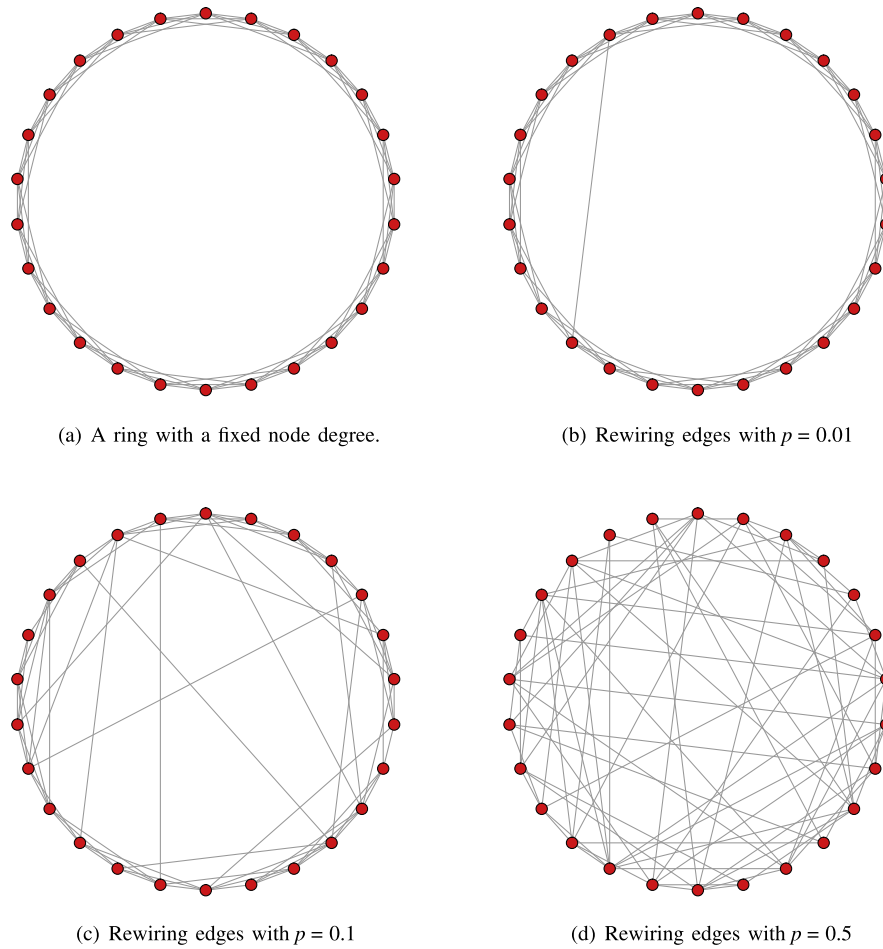


Fig. 2 Small-world graphs and their dependency on the rewiring probability p .

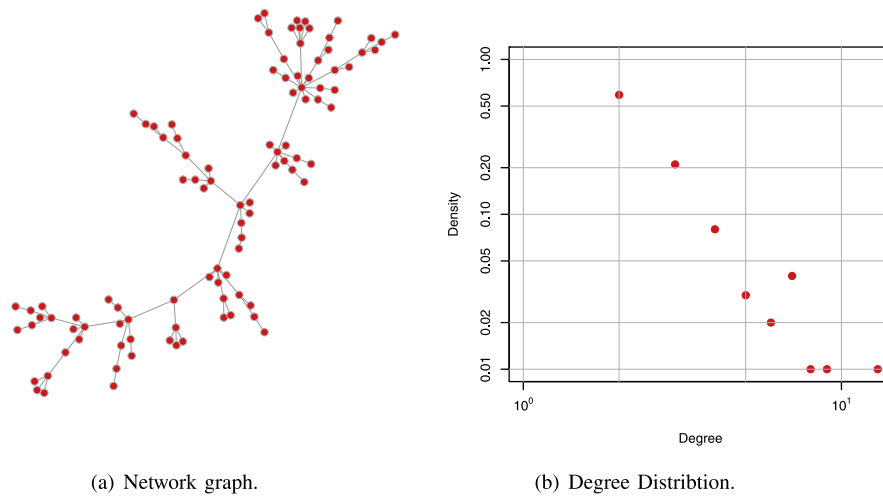


Fig. 3 Barabasi-Albert graph with $N=100$ and $m=1$.

low number of nodes with high degrees, whereas the majority of nodes exhibit $K=1$. This is also witnessed by the degree distribution plotted in the right part of the figure.

There have been several extensions of the basic model illustrated here, which vary from the details in the basic scheme (Albert and Barabasi, 2002). For example, by varying the importance of nodes, by assuming that links can disappear along time, or by adopting a different formulation of the probability to choose an attachment. Also, there has been criticism on the basic idea that

preferential attachment is the main model underlying link formation. For example, one can think of situations, common in biological systems (Vazquez et al., 2003; Solé et al., 2002), where the network grows as a result of node duplication and mutation. The *copying model* assumes a process similar to the one above described for the preferential attachment. However, at time t the connections of v_i are governed by a probability distribution p : With probability p the connections are uniformly chosen among the nodes in $G^{(t)}$, whereas with probability $1-p$ a node s in $G^{(t)}$ is randomly chosen, and a subset of the connections of s are copied as connections of v_i . Notably, this mechanism still produces a scale-free degree distribution and it is more realistic than the preferential attachment in the scenarios of interest for this article.

Besides these classical models, graph modeling can also be considered from a different perspective. We can consider a configuration $C = \{x_1, \dots, x_n\}$ of properties of interest (such as number of edges, centrality measures, etc.) and parameterize the probability of observing such properties in an exponential model. Thus, given a graph G , the probability of observing G with respect to the configuration C can be formalized as

$$Pr(G|C, \Theta) = \frac{1}{Z} \exp \left\{ \sum_i \theta_i x_i(G) \right\}$$

where Z is a normalizing constant, and $\Theta = \{\theta_1, \dots, \theta_n\}$ is the set of parameters which govern the probability of observing the given graph configuration. This general scheme is called *Exponential Random Graph model* (Newman, 2010; Kolaczyk, 2009), and it can be adopted in the traditional statistical setting where the purpose is to fit a parametric model to an ensemble of graphs describing specific properties of interest, rather than the graph as a whole.

Community Structure in Networks

A natural aspect, that several and heterogeneous networks exhibit, is the tendency of forming aggregations of nodes. These aggregations, commonly called clusters, groups or communities, are characterized by some sort of affinity among the nodes forming them and by a refractoriness to familiarize with external nodes. In some cases, communities can be grouped in super clusters in a recursive process that can generate a cluster hierarchy (Flake et al., 2002; Girvan and Newman, 2002). Networks with this behavior are said *modular* or *Small World Networks*.

In the literature, there is no standard definition of community since the concept of “affinity” strongly depends on the target scenario to be analyzed. Node affinity can be translated into similarity (of features, behavior or network topology, e.g., hive swarms), complementarity (e.g., some protein interactions) or shared elements (neighbors, e.g., social networks). In this article we formalize community models based on network topological aspects. An example is shown in Fig. 4, where a graph is partitioned into 4 communities, namely C1, C2, C3 and C4, distinguishable by different colors. As we can see, each community has a high inner level of connectivity, while the connectivity degree with other nodes is poor.

In general we can state that a community defines a group of nodes that exhibit a large number interactions among them with respect the other nodes of the network. This statement can be generalized as follows. Let $G = \langle V, E \rangle$ be a graph and $\delta(V, G)$ be a generic affinity function measuring a score of affinity of nodes in V according to the topology in G , then we define a community as a subgraph $G' = \langle V', E' \rangle$ such that:

$$V' \subseteq V, E' \subseteq E \text{ and } \delta(V', G') > \alpha \cdot \delta(V', G/G'),$$

where G/G' represent the complement of G' with respect to G and $\alpha \geq 1$ is a threshold value expressing how much the affinity within the community must be greater than the affinity with the rest of the graph. Example instantiations of δ are the following.

- Luce and Perry (1949) state that a community is a clique, i.e., a maximal complete subgraph of the network of at least three nodes. Luce (1950) further evolves the previous concept by defining a community as a relaxed variant of *clique*, namely n -

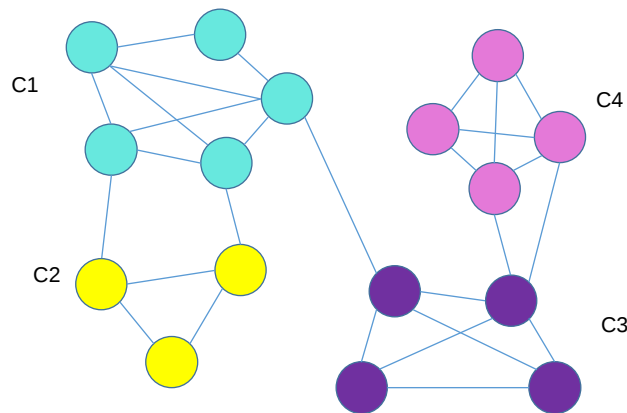


Fig. 4 Communities in a network.

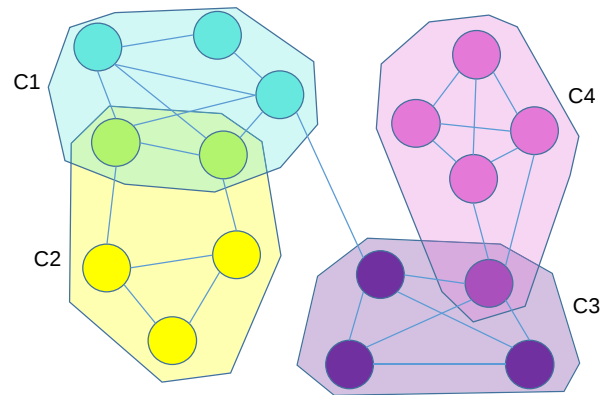


Fig. 5 An example partition into overlapping communities.

clique. A n -clique is a set of nodes in which at least two distinct n -chains exist. A n -chain is a path through the edges of the clique that involves all its n nodes, starting from and finishing on the same node.

- Seidman and Foster (1978) define a community as a maximal k -plex, which, given a positive integer k , is a subgraph of n nodes whose minimal degree is at least $n - k$ (notice that a complete graph is a 1-plex).
- In (Luccio and Sami, 1969) a graph G' is a community in G if, for each subgraph $G'' \subset G'$, such that $\text{terminals}(G'') > \text{terminals}(G')$ where the function $\text{terminals}(H)$ counts the number of paths connecting a generic group H to the nodes not in H . Borgatti et al. (1990) generalize the previous assumption with the property that nodes within a community show a greater edge connectivity with other community members than with external nodes.

Communities have been widely studied in social sciences, but it was in 2002 that the seminal paper by Girvan and Newman (2002) triggered a lot of interest on the problem of community detection. Since then, the problem has been extensively studied, mainly in the physics and in the computer science literature (see e.g., Fortunato, 2010; Xie et al., 2013; Harenberg et al., 2014; Schaub et al., 2017 for an extensive survey). It is important to categorize the approaches to community detection into two classes:

- Approaches which are able to detect non-overlapping communities. These assume that the nodes of the network can belong to a single group, the relation between the nodes is symmetric and no additional information is considered besides the graph structure (Raghavan et al., 2007; Newman, 2004, 2006; Clauset et al., 2004; Zhou, 2003; Richardson et al., 2009). Fig. 4 represent an example of partition into non-overlapping communities.
- By contrast, more recent approaches (Kumpula et al., 2008; Lancichinetti et al., 2008, 2011; Gregory, 2010; Xie and Szymanski, 2012; Chen et al., 2010; Padrol-Sureda et al., 2010) assume that communities can overlap, i.e., that nodes in a network can belong to different communities with different membership levels. Fig. 5 shows an example graph with overlapping communities.

Important examples in biology, where overlapping communities can be extremely useful, are Protein-protein interaction (PPI) networks (Adamcsek et al., 2006; Becker et al., 2012; Mahmoud et al., 2015). Proteins may have different functions in an organism, and typically they participate in a collective action to make a biological process start. A PPI network can be represented by a graph whose nodes are the proteins while an edge indicates a protein collaboration in a same biological process. Within such a scenario, communities represent different biological functions related to overlapping sets of proteins.

Predictive Models

In the previous section, we introduced models of the network characteristics which are capable of explain the real-world observations. The focus in this section is somewhat different. We assume a partial representation of the network graph $G = (V, E)$, and the purpose is to investigate prediction models which are capable of inferring the part of the network which is unobserved. More formally, we assume that each component i (either nodes or subsets of nodes) of the network is characterized by a property X_i , which can be either static or dynamic (i.e., depending from a temporal variable t so that $X_i(t)$ can change through time). For example, a property can be the possibility that an edge exists between a pair of nodes, or the susceptibility of an individual to a given infection. We shall not cover the topic of epidemics in this article (the interested reader can refer to Nowzari et al. (2016); Chen et al. (2013)). Instead, we focus on the topics of link prediction and graph classification.

Link Prediction

Link prediction (Martínez et al., 2016) is a discipline whose aim is to predict the evolution of a network. Specifically, we are interested in predicting which links will form or delete between nodes, by exploiting the network historical data. Within Fig. 6 we can observe how, given a fixed number of nodes, new edges will generate as time goes by: initially, the graph has two connected

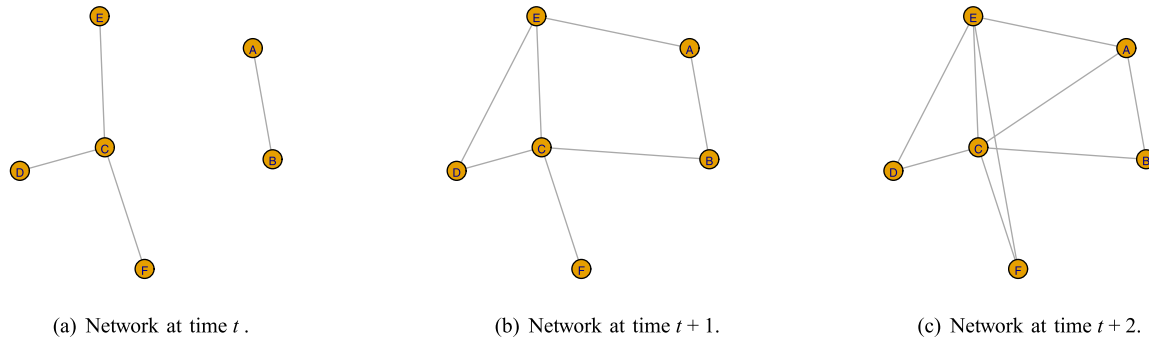


Fig. 6 Network evolution. Links among between nodes are continuously generated as time evolves.

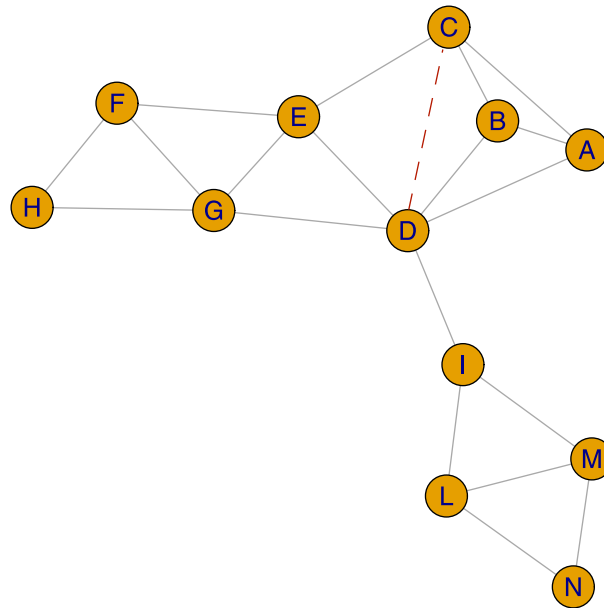


Fig. 7 Link prediction: The (C,D) edge is likely to belong to the networks, since all the neighbors of C are connected to D .

components with low global connectivity (**Fig. 6(a)**), then these components connected each other (**Fig. 6(b)**), finally another edge is added incrementing the connectivity (**Fig. 6(c)**).

Biology, biomedicine and bioinformatics have a wide and robust literature in this research area: for instance, [Martínez et al. \(2014\)](#) and [Schwikowski et al. \(2000\)](#) define an automatic technique for predicting interactions. **Fig. 7** shows an example situation where incomplete information is likely to occur, relative to the links in the network. Node C is surrounded by nodes which are all connected to D , and hence it is unlikely that a connection (C, D) won't occur.

To formalize the link prediction problem, we assume that $G=(V, E)$ is the current snapshot of the network. Then, for each pair $x, y \in V$, a *Link Prediction model* can be devised as a function $score(x, y)$, which represents the likelihood that the edge (x,y) should belong to E . Based on the function, G can be updated accordingly, by adding edges with high likelihood and/or removing edges with low likelihood. The specification of $score(x, y)$ falls roughly into one of the following categories: *Deterministic modeling*, *Probabilistic modeling* and *modeling based on supervised/unsupervised learning approaches*.

The idea within deterministic modeling consists in specifying $score(x, y)$ as a similarity score: it is likely that a connection is established between two nodes that are similar, i.e., that share relevant features, interests or objectives, or that exhibit a common behavior. Again, the similarity can be measured *locally* or *globally*, relative to the topology of the network.

- Local approaches define their similarity function observing only the features and the local network topology that are proper of the pair of nodes to rank; no other information is needed. The simplest definition exploits *Common Neighbors* ([Liben-Nowell and Kleinberg, 2007](#)),

$$score(x, y) = |Neighbors(x) \cap Neighbors(y)|.$$

Other variations of this definition are based on *Jaccard Index*, *Mutual Information* ([Tan et al., 2014](#)). The *Adamic-Adar Index*

(Adamic and Adar, 2001) is based on a slightly different definition, which somehow weights the importance of common neighbors:

$$score(x, y) = \sum_{z \in \text{Neighbors}(x) \cap \text{Neighbors}(y)} \frac{1}{\log|\text{Neighbors}(x)|}.$$

Other variations are based on the notions of *Resource Allocation* (Zhou et al., 2009) or *Preferential Attachment* Barabasi and Albert (1999).

- Global approaches exploit the whole network information. In its simplest form, a score based on global information can relate the similarity to the network distance and its variations (Liben-Nowell, 2005): $score(x, y) = d_{x,y}$. More refined proposal, such as the *The Katz Index* (Katz, 1953), weight the length and number of occurrences of a paths:

$$score(x, y) = \sum_l \beta^l \cdot |\text{paths}^{(l)}(x, y)|.$$

Here, β is a constant value and $\text{paths}^{(l)}(x, y)$ is the set of all paths of length l between x and y . *Random Walk models* (Liu and Lii, 2010) are based on the assumptions that a random walk in the network will eventually connect x and y . This entails a recursive definition, where the score depends on the score of the neighbors:

$$score(x, y) = \begin{cases} 1 & \text{if } x = y \\ \gamma \sum_{u \in \text{Neighbors}(x)} \sum_{v \in \text{Neighbors}(y)} score(u, v) & \text{otherwise} \end{cases}$$

where γ represents a normalization constant.

Probabilistic modeling maps the scoring function to the the likelihood of observing the link of interest in a probabilistic setting. In practice, the assumption is that the network is generated through a stochastic model governed by a parameter set θ , to be estimated. In this context, $score(x, y)$ represents the probability $\Pr(x, y | \theta)$ to observe the link (x, y) .

In a sense, the models illustrated in Section “Network Models” propose a way of encoding the above probability. For example, the random graph assumes a constant probability $score(x, y) = p$ for each pair of nodes. Specific instantiations of the exponential random graph model are particularly well-suited to model link prediction, by assuming that the properties of interest are indeed the prospective links. Also, communities can explain link formation by decomposing the probability of a link as the probability that the nodes belong to a given community, multiplied by the probability of observing a link within that community.

More in general, $\Pr(x, y | \theta)$ can be expressed by mapping a set of features $w_{x,y}$ relative to both x and y to a function $g(w_{x,y})$, representing the likelihood to observe these features when x and y are connected. Under this perspective, models based on probabilistic relational modeling (Getoor et al., 2003; Friedman et al., 1999; Heckerman and Meek, 2004; Yu et al., 2006) were also proven suitable for the task of link prediction. Probabilistic Relational Models extend the standard framework of Bayesian Networks by modeling properties and relations between data objects. The framework can be exploited to infer the probabilistic dependencies between properties of nodes and edges connecting such nodes, thus allowing for an effective formulation of the scoring function.

One can also think of specifying $score(x, y)$ as a binary function, and model link prediction as a supervised learning task. In this case, the objective is to specify a classification task which, based on the features $w_{x,y}$ of the prospective (x, y) link, is capable of predicting whether the link occurs. Features can include topological properties and node properties (such as type information, or information content associated with the node). Classification approaches can span from logistic regression to decision trees, SVMs and ensembles (Martínez et al., 2016; Lü and Zhou, 2011; Liben-Nowell and Kleinberg, 2007).

Graph Classification

We conclude this section by briefly mentioning other approaches to supervised learning on graphs. Graph classification can refer two different tasks. A first, intuitive tasks regards the prediction of specific attribute values relative to single nodes of the network. The modeling here resembles the idea of link prediction, where however the prediction of unobserved node attributes is the only goal. Once, again, one can think of formulating a dependency structure among the attributes and with respect of the topology of the network, or include side information which characterize the unknown nodes properties, to be inferred, in terms of the known properties. The probabilistic modeling of Getoor et al. (2003), or even simpler approaches based on standard classification algorithms, can be suitable for these tasks. The challenge here is to encode the information coming from the structure of the neighborhood of the node.

A different task (Tsuda and Saigo, 2010; Rehman et al., 2012; Kong and Yu, 2014) consists in training a model able to predict a class label for a whole graph (Motoda, 2006). This process exploits supervised/semi supervised learning techniques. However, the main issue is how to adapt classical machine learning models to exploit structural and topological information. Essentially, this corresponds to define a mapping function ϕ which encodes a graph G into a relational structure $\phi(G)$, representing a set of features of interest. The features can represent simple properties (Deshpande et al., 2005; Zhu et al., 2012), subgraph patterns (Saigo et al., 2009; Fei and Huan, 2010), or even high-dimensional representations (Gärtner, 2003; Riesen and Bunke, 2009; Schlkopf et al., 2004).

Closing Remarks

In this article we provided an overview of the main network models defined in the field of Network Science for describing and predicting the behavior of large real networks. In particular, we focused on: (i) Mathematical models for explaining structural properties of the network, (ii) descriptive models for detecting relevant groups within the network (communities), (iii) models for predicting properties of entities within a graph, such as unobserved connections/relationships between nodes or node/graph attributes.

Each of these topics exhibits a wide literature and several aspects that, for space reasons, we were not able to cover in this survey. Indeed, the article is meant to provide a basic reference summary for further deepening into the subject. The interested reader is strongly advised to go through the bibliographic references and explore the topics of her interest.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Quantification of Proteins from Proteomic Analysis

References

- Adamcsek, B., Palla, G., Farkas, I., Derényi, I., Vicsek, T., 2006. Cfinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* 22, 1021–1023.
- Adamic, L.A., Adar, E., 2001. Friends and neighbors on the web. *Social Networks* 25, 211–230.
- Albert, R., Barabási, A., 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 47–97.
- Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. *Science* 286, 509–512.
- Barrat, A., Barthelemy, M., Vespignani, A., 2008. *Dynamical Processes on Complex Networks*. Cambridge: Cambridge University Press.
- Becker, E., Robison, B., Chapple, C.E., Gunoch, A., Brun, C., 2012. Multifunctional proteins revealed by overlapping clustering in protein interaction network. *Bioinformatics* 28, 84–90.
- Bollobás, B., Chung, F.R.K., 1988. The diameter of a cycle plus a random matching. *SIAM Journal on Discrete Mathematics* 1, 328–333.
- Borgatti, S.P., Everett, M.G., Shirey, P.R., 1990. Ls sets, lambda sets and other cohesive subsets. *Social Networks* 12, 337–357.
- Bullmore, E., Sporns, O., 2009. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience* 10, 186–198.
- Chen, W., Lakshmanan, L., Castillo, C., 2013. Information and influence propagation in social networks. In: *Proceedings of the Synthesis Lectures on Data Management*, Morgan & Claypool Publishers.
- Chen, W., Liu, Z., Sun, X., Wang, Y., 2010. A game-theoretic framework to identify overlapping communities in social networks. *Data Mining and Knowledge Discovery* 21, 224–240.
- Clauset, A., Newman, M.E.J., Moore, C., 2004. Finding community structure in very large networks. *Physical Review E* 70.
- Davidson, E., Levin, M., 2005. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America* 102, 4935.
- Dehmer, M., 2010. *Structural analysis of complex networks*. Birkhäuser Basel.
- Deshpande, M., Kuramochi, M., Wale, N., Karypis, G., 2005. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering* 17, 1036–1050.
- Dorogovtsev, S.N., Mendes, J.F.F., 2003. *Evolution of Networks, From Biological Nets to the Internet and WWW*. Oxford: Oxford University Press.
- Dunne, J.A., Williams, R.J., Martinez, N.D., 2002. Network structure and biodiversity loss in food webs: Robustness increases with connectance. *Ecology Letters* 5, 558–567.
- Eisenberg, D., Marcotte, E.M., Xenarios, I., Yeates, T.O., 2000. Protein function in the post-genomic era. *Nature* 405, 823–836.
- Erdős, P., Rényi, A., 1959. On random graphs, I. *Publicationes Mathematicae (Debrecen)* 6, 290–297.
- Fei, H., Huan, J., 2010. Boosting with structure information in the functional space: An application to graph classification. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 643–652.
- Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M., 2002. Self-organization and identification of web communities. *Computer* 35, 66–71.
- Fortunato, S., 2010. Community detection in graphs. *Physics Reports* 486, 75–174.
- Friedman, N., Getoor, L., Koller, D., Pfeffer, A., 1999. Learning probabilistic relational models. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 1300–1309.
- Gärtner, T., 2003. A survey of kernels for structured data. *SIGKDD Exploration Newsletter* 5, 49–58.
- Getoor, L., Friedman, N., Koller, D., Taskar, B., 2003. Learning probabilistic models of link structure. *Journal of Machine Learning Research* 3, 679–697.
- Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America* 99, 7821–7826.
- Gregory, S., 2010. Finding overlapping communities in networks by label propagation. *New Journal of Physics* 12, 103018.
- Habibi, I., Emami, E.S., Abdi, A., 2014. Quantitative analysis of intracellular communication and signaling errors in signaling networks. *BMC Systems Biology* 8, 89.
- Harenberg, S., Bello, G., Gjeltner, L., et al., 2014. Community detection in large-scale networks: A survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics* 6, 426–439.
- Heckerman, D., Meek, C., 2004. Probabilistic entity-relationship models, prms, and plate models. In: *Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields*, pp. 55–60.
- Katz, L., 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 39–43.
- Kleinberg, J., 2000. Navigation in a small world. *Nature* 406, 845.
- Kolaczyk, E.D., 2009. *Statistical Analysis of Network Data*. Berlin: Springer.
- Kong, X., Yu, P., 2014. Graph classification in heterogeneous networks. In: Alhajj, R., Rokne, J. (Eds.), *Encyclopedia of Social Network Analysis and Mining*. Springer, p. 641.
- Kumpula, J.M., Kivela, M., Kaski, K., Saramaki, J., 2008. A sequential algorithm for fast clique percolation. *Physical Review E* 78, 026109.
- Lancichinetti, A., Fortunato, S., Kertész, J., 2008. Detecting the overlapping and hierarchical community structure of complex networks. *ArXiv e-prints*.
- Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S., 2011. Finding statistically significant communities in networks. *PLOS ONE* 6, e18961.
- Liben-Nowell, D., 2005. *An algorithmic approach to social networks*. PhD thesis, Massachusetts Institute of Technology. Cambridge, MA, USA.
- Liben-Nowell, D., Kleinberg, J., 2007. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology* 58, 1019–1031.

- Liu, W., Lii, L., 2010. Link prediction based on local random walk. *EPL (Europhysics Letters)* 89, 58007.
- Loscalzo, S., Yu, L., 2008. Social network analysis: Tasks and tools. *Social Computing, Behavioral Modeling, and Prediction*. 151–159.
- Luccio, F., Sami, M., 1969. On the decomposition of networks in minimally interconnected subnetworks. *IEEE Transactions on Circuit Theory* 16, 184188.
- Luce, R.D., 1950. Connectivity and generalized cliques in sociometric group structure. *Psychometrika* 15, 169–190.
- Luce, R.D., Perry, A.D., 1949. A method of matrix analysis of group structure. *Psychometrika* 14, 95–116.
- Lü, L., Zhou, T., 2011. Link prediction in complex networks: A survey. *Physica A Statistical Mechanics and its Applications* 390, 1150–1170.
- Mahmoud, H., Masulli, F., Rovetta, S., Russo, G., 2015. Detecting overlapping protein communities in disease networks. In: *Proceedings of the Computational Intelligence Methods for Bioinformatics and Biostatistics*, pp. 109–120.
- Manco, G., Ritacco, E., Guarascio, M., 2018. Network topology. In: *Proceedings of the Encyclopedia of Bioinformatics and Computational Biology*, Elsevier. This volume.
- Martínez, V., Berzal, F., Cubero, J., 2016. A survey of link prediction in complex networks. *ACM Computing Surveys* 49 (69), 1–33.
- Martínez, V., Cano, C., Blanco, A., 2014. ProphNet: A generic prioritization method through propagation of information. *BMC Bioinformatics* 15.
- Motoda, H., 2006. What can we do with graph-structured data? – A data mining perspective. In: *Proceedings of the AI 2006: Advances in Artificial Intelligence*, pp. 1–2.
- Newman, M.E., 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* 103, 8577–8582.
- Newman, M.E.J., 2003. The structure and function of complex networks. *SIAM Review* 45, 167–256.
- Newman, M.E.J., 2004. Fast algorithm for detecting community structure in networks. *Physical Review E* 69, 066133.
- Newman, M.E.J., 2010. *Networks: An Introduction*. Oxford: Oxford University Press.
- Nowzari, C., Preciado, V.M., Pappas, G.J., 2016. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems* 36, 26–46.
- Padrol-Sureda, A., Perarnau-Llobet, G., Pfeifle, J., Munts-Mulero, V., 2010. Overlapping community search for social networks. In: *Proceedings of the 26th IEEE Conference on Data Engineering (ICDE 2010)*, pp. 992–995.
- Palsson, B.O., 2006. *Systems biology*. In: *Proceedings of the Properties of Reconstructed Networks*, Cambridge University Press.
- Price, D., 1976. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science* 27, 292306.
- Raghavan, U., Albert, R., Kumara, S., 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 036106.
- Rehman, S.U., Khan, A.U., Fong, S., 2012. Graph mining: A survey of graph mining techniques. In: *Proceedings of the Seventh International Conference on Digital Information Management (ICDIM 2012)*, pp. 88–92.
- Richardson, T., Mucha, P.J., Porter, M.A., 2009. Spectral tripartitioning of networks. *Physical Review E* 80, 036111.
- Riesen, K., Bunke, H., 2009. Graph classification by means of lipschitz embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 1472–1483.
- Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., Tsuda, K., 2009. gboost: A mathematical programming approach to graph classification and regression. *Machine Learning* 75, 69–89.
- Schaub, M.T., Delvenne, J., Rosvall, M., Lambiotte, R., 2017. The many facets of community detection in complex networks. *Applied Network Science* 2.
- Schlkopf, B., Tsuda, K., Vert, J., 2004. *Kernel Methods in Computational Biology*. MIT Press.
- Schwikowski, B., Uetz, P., Fields, S., 2000. A network of protein–protein interactions in yeast. *Nature Biotechnology* 18, 1257–1261.
- Seidman, S.B., Foster, B.L., 1978. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology* 6, 139–154.
- Solé, R., Pastor-Satorras, R., Smith, E., Kepler, T., 2002. A model of large-scale proteome evolution. *Advances in Complex Systems* 5, 43–54.
- Tan, F., Xia, Y., Zhu, B., 2014. Link prediction in complex networks: A mutual information perspective. *PLOS ONE* 9 (9), e107056. (CoRR abs/1405.4341).
- Tsuda, K., Saigo, H., 2010. Graph classification. In: Aggarwal, C.C., Wang, H. (Eds.), *Managing and Mining Graph Data*, pp. 337–363.
- Van Mieghem, P., 2014. *Performance Analysis of Complex Networks and Systems*. Cambridge University Press.
- Vazquez, A., Flammini, A., Maritan, A., Vespignani, A., 2003. Modeling of protein interaction networks. *ComPlexUs* 1, 38–44.
- Watts, D.J., 1999. *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press.
- Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of ‘small-world’ networks. *Nature* 393, 440–442.
- Weirauch, M.T., 2011. Gene coexpression networks for the analysis of DNA microarray data. pp. 215–250.(Chapter 11)
- Xie, J., Kelley, S., Szymanski, B.K., 2013. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys* 45 (4), 1–43.
- Xie, J., Szymanski, B.K., 2012. Towards linear time overlapping community detection in social networks. In: *Proceedings of the Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD-2012)*, pp. 25–36. Berlin, Heidelberg: Springer.
- Yu, K., Chu, W., Yu, S., Tresp, V., Xu, Z., 2006. Stochastic relational models for discriminative link prediction. In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pp. 1553–1560.
- Zhou, H., 2003. Distance, dissimilarity index, and network community structure. *Physical Review E* 67.
- Zhou, T., Lü, L., Zhang, Y.C., 2009. Predicting missing links via local information. *The European Physical Journal B*. 623–630.
- Zhu, Y., Yu, J.X., Cheng, H., Qin, L., 2012. Graph classification: A diversified discriminative feature selection approach. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 205–214.

Community Detection in Biological Networks

Marco Pellegrini, Consiglio Nazionale delle Ricerche, Istituto di Informatica e Telematica, Pisa, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A strong thesis underlying the *systems biology* approach to challenging questions in biology is that such challenges are more likely to be met when framed in terms of (static or dynamic) features of molecular networks. Biological systems and in particular molecular networks allow to uncover emerging phenomena, not evident when interactions among cellular molecular components are considered in isolation. Alternatively, systems and networks can be seen as a suitable tool to model, describe and predict *mesoscale* phenomena, thus providing an essential bridge between the observable phenotypes (e.g. complex traits, diseases and tumors) and molecular “omic” characterizations (e.g. gene mutations, expressed proteins, epigenetic signals). Biological networks describing molecular interactions play a key role in this scenario. We can distinguish two main challenges. The first one is building reliable biological networks from inherently noisy data coming either from direct measurement of interactions, or inferred from indirect evidence. The second challenge is that of using such models (often obtained at a considerable cost) to do meaningful network data analysis leading to the prediction of putative underlying molecular mechanisms. Although several types of networks have been defined and studied in literature, protein-protein interaction networks (PPINs) have emerged at the forefront. This is due to several converging reasons. Firstly, most cell functions are mediated by proteins that cooperate to attain a certain functional effect. Secondly, protein interactions are amenable to cost-effective high throughput assays, thus giving the possibility of building global system-level snapshots of a complete system (often at a cellular level, and beyond). Thirdly, PPINs exhibit strong modularity in a topological sense, that give strong hints of the organization at a functional level. Due to their large size, and importance, PPINs have become the archetypal test case for developing efficient network analysis algorithm, as direct manual inspection of the wealth of accumulated data is next to impossible. Although our survey focusses on community detection for PPI networks, one should bear in mind that several different types of biological networks have been proposed, for which the algorithms included in this survey (or variants thereof) can be applied (Vidal et al., 2011; Gligorić et al., 2015).

Network science is a rich and steady trend of modern research, that draws techniques from fields like mathematics, graph theory, physics, statistics, and computer science (just to name the main sources), and applies them in a variety of fields, including notably the analysis of social, biological and technological networks. Thus, the algorithms described in this survey for community detection in biological networks are built upon a mathematically fertile ground cross fertilized by a large variety of approaches and inspirations.

In the background section, we will mainly recall basic definitions from graph theory underpinning many of the algorithms and methods introduced in the methodologies section. Then, we described the main measures used to assess the quality of the community prediction algorithms. We recall a few notable applications of these methods to large scale projects. Finally, we mention several variations on the theme as well as promising areas of future research, an open questions.

Background

Let $G=(V, E \subseteq V \times V)$ be a simple (undirected) graph (no self-loops, no multiple edges). To be concrete one could think of the nodes in V as molecular species (i.e. proteins, genes, metabolomes) and of the edges in E as evidence of their mutual interaction in the biological process under study. A subset of the nodes $Q \subset V$ induces a subgraph $H_Q=(Q, E_Q)$, where $E_Q=\{(a, b) \in E \mid a \in Q \wedge b \in Q\}$. For a (sub)graph G its *average degree* is:

$$av(G) = \frac{2|E|}{|V|}$$

The *density* of a graph $D(G)$ is the following ratio:

$$D(G) = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)}$$

which gives the ratio of the number of edges in G to the maximum possible number of edges in a complete graph with the same number of nodes.

A *clique* is a (sub)graph with density 1 (maximum possible value). A k -*clique* is a clique of k nodes. A clique C is *maximal* in G if, $C=V$, or, for any $v \in V \setminus C$, $C \cup \{v\}$ is not a clique in G . Given a parameter $\gamma \in [0..1]$, a γ -*quasi clique* is a graph $G=(V, E)$ such that:

$$\forall v \in V \quad |N_G(v)| \geq \gamma(|V|-1)$$

where $N_G(v) = \{u \in V \mid (u, v) \in E\}$ is the set of immediate neighbors of v in G . Note that v does not belong to $N_G(v)$.

The density of the subgraph induced by $N_G(v)$ is called the *clustering coefficient* of v . A k -core of a graph G is a maximal connected subgraph of G in which all vertices have degree at least k . A vertex v has *core number* (also called *core index*) k if it belongs to a k -core but not to any $(k+1)$ -core. A *core decomposition* of a graph is the partition of the vertices of a graph induced by their core numbers (Seidman, 1983). The *core-clustering coefficient* of a vertex v is the density of the highest k -core in the subgraph induced by $N_G(v) \cup \{v\}$.

Methodologies

Communities in graphs are often defined as relatively dense sub-graphs, thus, many methods for community detection are built upon finding local optima for some density-related function. Here, we list in chronological order important methods built upon this characterization, focussing on the most important steps of each method.

MCode

MCode (Bader and Hogue, 2003) is one of the earliest algorithms explicitly designed to cluster PPI networks. MCode is organized in four phases. In the first phase, each node v is given a weight that is the product of the core-clustering coefficient of v and the maximum value k for which there is a k -core in $N_G(v)$. The nodes are ranked by decreasing values of their weight and used as seeds in this order for the next phase. In the second phase, starting from the unused node highest in rank, MCode adds iteratively neighbors to the cluster being built, provided their weight is above a fixed fraction of the seed's weight. The added nodes are marked as used. In the third phase, clusters that do not include a 2-core are discarded and it is possible to augment the cluster with neighbouring nodes above a certain fixed fraction of the seed's weight. These additional nodes are not marked as used, thus they can be shared among several communities. The final stage computes the largest connected 2-core in each candidate cluster.

C-finder

CFinder (Adamcsek et al., 2006) is a fast program that implements the CPM (Clique Percolation Method) for graph clustering (Palla et al., 2005). The algorithm works in two main phases. The first phase finds all k -cliques in the input network, then CFinder associates two k -cliques if they share a common $(k-1)$ -clique. The clusters produced by CFinder are the equivalence classes of this association relation. Note that the produced distinct clusters may share nodes, and the intersection of two clusters may contain $(k-2)$ -cliques but never a $(k-1)$ -clique. CFinder was one of the first clustering algorithms that could control tightly the overlapping structure of the family of clusters it produces. The parameter k is critical and for biological applications a value in the range $k=4,5,6$ is suggested in the original publication.

CMC

The CMC algorithm (Clustering Based on Maximal Cliques) (Liu et al., 2009) works with edge-weighted graphs. CMC begins by listing all maximal cliques in a graph G by using the algorithm in Tomita et al. (2006), a fast variant of the Bron-Kerbosch method (Bron & Kerbosch, 1973). In the next phase, each maximal clique is given a score that is its average edge weight, and these clusters are ranked by decreasing score value. In the next phase, for each cluster C_i the algorithm considers in turns the clusters C_j of lower score and decides whether to merge C_i and C_j (or remove C_j) based on the relative size of their overlap, and on an weighted inter-cluster score function.

Coach

COACH (Wu et al., 2009) is based on the concept of the *core graph* of a graph $H=(V, E)$. Formally a *core graph* $CG(H)$ is defined as follows. Let's take any vertex $v \in V$ and a vertex u in its neighborhood $u \in N_H(v)$. If the degree of u in the subgraph induced by $N_H(v)$ is larger than the average degree of $G[N_H(v)]$, then u is labeled as a *core vertex*. The core graph of H , $CG(H)$ is the subgraph induced in H by the collection of all the *core vertices* of H for some vertex v (Note that, even if the name is similar, the notion of a *core vertex* in (Wu et al., 2009) is quite different from the notion of core number/core decomposition of (Seidman, 1983) which is based instead on the *minimum degree* of an induced subgraph.)

Given an input PPIN graph G , the algorithm COACH produces first a set of 'preliminary cores'. For each vertex $v \in G$, COACH considers the subgraph induced by $N_G(v) \cup \{v\}$ and computes the core graph of this restricted subgraph. If this core graph is sufficiently dense it is directly the preliminary core of v . Otherwise a greedy procedure is applied to extract from the core graph a sufficiently dense subgraph. After computing a preliminary core for each vertex in G , quasi-duplicates are removed. In the second phase, we add to each preliminary core any node that is linked to more than 50% of the nodes in the preliminary core. Note that both the selected preliminary cores and the preliminary cores augmented with the peripheral nodes may overlap.

SPICi and ClusterOne

Given a PPIN G with edge weights, SPICi (Jiang & Singh, 2010) uses a greedy approach in which nodes of the network are ranked by their weighted degree, and are considered in that order as candidate cluster seeds. A cluster is then expanded starting from a seed by adding the adjacent node with highest connection (by a weighted measure) to the current cluster, while keeping the weighted density above a user defined threshold. The use of advanced data structures to support dynamic graph operations ensures that this method is efficient even for very large graphs.

ClusterONE (Nepusz *et al.*, 2012) is a greedy method similar to SPICi but with significant differences. One difference is that in the cluster construction phase the algorithm uses both expansion and contraction operations. The computed clusters optimize a relative measure of cohesiveness that is essentially a ratio of the internal total weight of a cluster over the total weight of the edges that connect it to the complement graph. More formally, the algorithm evaluates for a set of nodes $V_0 \subseteq V$ the following function:

$$f(V_0) = \frac{w^{in}(V_0)}{w^{in}(V_0) + w^{out}(V_0) + p|V_0|}$$

where $w^{in}(V_0)$ is the cumulative weight of edges within the induced subgraph $G[V_0]$, $w^{out}(V_0)$ is the cumulative weight of edges connecting nodes in V_0 with nodes in $V \setminus V_0$, and $p|V_0|$ is a penalty terms that captures the possibly missing edges in G . The above process is then iterated on the unused node of highest rank elected as the next seed. Finally, in a post-processing phase, highly overlapping clusters are merged, and the candidate list of clusters is returned.

ProRank +

For a given PPIN G ProRank + (Zaki *et al.*, 2012; Hanna & Zaki, 2014) uses several edge pruning and filtering steps before applying a ranking step inspired by the well known Page-Rank algorithm in order to highlight candidate protein seeds, which is the eponymous step of the method.

In detail, the first phase of the algorithm uses the AdjustCD method (Chua *et al.*, 2006) to prune unreliable PPI from the graph G . Moreover three protein types based on local topology (named bridge, fjord, and shore proteins) are identified and removed, since they do not contribute to defining complexes. In the second phase for each pair of nodes the sequence similarity of the corresponding protein primary structure is computed via standard Smith-Waterman scoring. In the third phase a pager-rank-like iterative process that uses the network topology as well as the sequence similarity is applied so to obtain a non-local assessment of the importance of each node.

In the fourth phase, in order of page-rank value, each protein is taken as a seed and considered along with all its neighbors, as a candidate community. These nodes are marked as used and the process iterated with the next highest unused node.

The candidate communities are then filtered by removing almost duplicate communities. Pairs of candidate communities passing a cohesiveness test (similar to the scoring function in Nepusz *et al.* (2012)) are merged together.

Core&Peel

The Algorithm Core&Peel (Pellegrini *et al.*, 2016) works in four main phases. In the first phase, the *Core Decomposition* of the input graph G (denoted with $CD(G)$) is computed, using the linear time algorithm in Batagelj & Zaversnik (2003), giving us the core number $C(v)$ for each node $v \in V$. Moreover, for each vertex v in G the algorithms computes the *Core Count* of v , denoted with $CC(v)$, defined as the number of neighbors of v having core number at least as large as $C(v)$. Next, the vertices of V are sorted in decreasing lexicographic order of their core values $C(v)$ and core count value $CC(v)$.

In the second phase, each node v is considered in turn, in the order given by the first phase. For each v , we construct the set of nodes $N_{C(v)}(v)$ of neighbors of v in G having core number greater than or equal to $C(v)$, and the induced subgraph $G[N_{C(v)}(v) \cup \{v\}]$. We apply some filters based on simple node/edge counts in order to decide whether this subgraph should be processed in the third phase.

In the third phase, we take v and the induced subgraph $G[N_{C(v)}(v) \cup \{v\}]$ and we apply a variant of the peeling procedure described in Charikar (2000) that iteratively removes nodes of minimum degree in the graph. The peeling procedure stops (and reports failure) when the number of nodes drops below the user-defined size threshold q . The peeling procedure stops (and reports success) when the density of the resulting subgraph is above or equal to the user-defined density threshold δ . The set of nodes returned by the successful peeling procedure is added to the set of candidate communities.

In the fourth phase, the algorithm eliminates duplicates and communities completely enclosed in other communities. Also, for pairs of communities very similar to each other as measured by the Jaccard coefficient, the smaller of the two is discarded.

Assessment

In parallel with the development of new methods for community detection, the methodologies for assessing quantitatively the quality of the output have been evolving over time. In particular, as no single score function can be trusted to capture all the aspects of this multifarious problem, often such measures are aggregated together to provide a robust ranking.

Evaluation Measures for Protein Complex Prediction

Li et al., 2010

F-measure: Li et al. (2010) adopted the following f-measure computation to estimate the degree of matching between the found cluster and the gold standard complex. Let P be the collection of discovered clusters and let B be the collection of the gold standard complexes. For a pair of sets $p \in P$ and $b \in B$, the *precision-recall product score* is defined as $PR(p, b) = \frac{|p \cap b|^2}{|p| \times |b|}$. Only the clusters and complexes that pass a $PR(p, b)$ threshold ω (step function) are then used to compute precision and recall measures. Namely we define the matching sets: $N_p = |\{p \in P, \exists b \in B, PR(p, b) \geq \omega\}|$, and $N_b = |\{b \in B, \exists p \in P, PR(p, b) \geq \omega\}|$. Afterwards: $Precision = \frac{N_p}{|P|}$, $Recall = \frac{N_b}{|B|}$, and the *F-measure* is the harmonic mean of precision and recall. (Li et al., 2010) and other authors use $\omega = 0.2$ (or $\omega = 0.25$). Experiments in Peng et al. (2014) indicate that the relative ranking of methods is robust against variations of the value of ω .

Song & Singh 2009

Song and Singh (2009) propose three measures to evaluate the overlap between complexes and predicted clusters: the *Jaccard measure*, the *precision-recall measure* and the *semantic similarity measure*.

Jaccard measure: Let the sets P and B be as above, for a pair of sets $p \in P$ and $b \in B$, their Jaccard coefficient is $Jac(p, b) = \frac{|p \cap b|}{|p \cup b|}$. For each cluster p it is defined $Jac(p) = \max_{b \in B} Jac(p, b)$, and for each complex b it is defined $Jac(b) = \max_{p \in P} Jac(p, b)$. Next, compute the weighted average Jaccard measures using, respectively, the cluster and complex sizes: $Jaccard(P) = \frac{\sum_{p \in P} |p| Jac(p)}{\sum_{p \in P} |p|}$, and

$Jaccard(B) = \frac{\sum_{b \in B} |b| Jac(b)}{\sum_{b \in B} |b|}$. Finally, the *Jaccard measure* is the harmonic mean of *Jaccard* (P) and *Jaccard* (B).

Precision recall product: This measure is computed using exactly the same work flow as *Jaccard*, except that the Jaccard coefficient is replaced with the precision-recall product score used also in Li et al. (2010).

Semantic similarity measure: Let the sets P and B be as above, for a protein x , we define $P(x)$ as the set of predicted clusters that contain x : $P(x) = \{p \in P | x \in p\}$, and $B(x)$ as the set of golden complexes that contain x : $B(x) = \{b \in B | x \in b\}$. Denote with $I(\cdot)$ the indicator function of a set that is 0 for the empty set and 1 for any other set. Let $Bin(\cdot)$ denote the set of unordered pairs of distinct elements of a set. The semantic similarity of p in B is: $Den(p, B) = \frac{\sum_{(x, y) \in Bin(p)} I(B(x) \cap B(y))}{|Bin(p)|}$. Analogously the semantic similarity of b in P is: $Den(b, P) = \frac{\sum_{(x, y) \in Bin(b)} I(P(x) \cap P(y))}{|Bin(b)|}$. Next, we compute the weighted average semantic similarity weighted respectively by cluster and complex size: $Density(P) = \frac{\sum_{p \in P} |p| Den(p, B)}{\sum_{p \in P} |p|}$, and $Density(B) = \frac{\sum_{b \in B} |b| Den(b, P)}{\sum_{b \in B} |b|}$. Finally, the *Semantic Similarity Measure* is computed as the harmonic mean of *Density* (P) and *Density* (B). The semantic similarity measure is the only measure, among those considered in this survey, that puts a premium on correctly guessing the overlap among protein complexes in the golden standard.

Geometric accuracy, sensitivity, and positive predicted value (PPV)

Consider the predicted complexes P and the benchmark complexes B and define the weight $w(p, b) = |p \cap b|$ with $p \in P$ and $b \in B$.

The *Sensitivity* is:

$$S_n = \frac{\sum_{b \in B} \max_{p \in P} w(p, b)}{\sum_{b \in B} |b|}$$

The *Positive Predictive Value* (PPV) is:

$$PPV = \frac{\sum_{p \in P} \max_{b \in B} w(p, b)}{\sum_{p \in P} \sum_{b \in B} w(p, b)}$$

The *geometric accuracy* (Acc) is the geometric mean of S_n and PPV.

$$Acc = \sqrt{S_n \cdot PPV}$$

These three measures are widely used, however it has been noticed that they may produce unreliable scores, when applied to gold standards with highly overlapping communities (see Suppl. Materials in Song & Singh (2009)).

Maximum matching ratio

It is a desirable feature of predicted communities that each community in the reference gold standard is predicted only once. As many quality measures do not reward this property Nepusz et al. (2012) propose a new quality measure: the *maximum matching ratio* (MMR). Consider the predicted complexes P and the benchmark complexes B as nodes in a bipartite graph, where an edge (p, b) with $p \in P$ and $b \in B$ has a weight given by the *precision-recall product score* defined above. A *maximum weighted matching* (Kuhn, 1955) is computed in this bipartite weighted graph that associates at most one prediction to each known complex, while maximizing the weight of the global matching. The MMR is the ratio of the value of the maximum weighted match over the number of complexes in the benchmark (i.e. $|B|$).

Evaluation Measure for Gene Ontology Coherence

Gene Ontology is a well known tool for the functional characterization for proteins. As one expects that predicted clusters exhibit as strong functional coherence, measuring the enrichment of a Gene Ontology class over the predicted clusters is an additional important quality measure.

Formally, let G be a collection of gene ontology annotations, and g one GO class in G , and let M be the set of all proteins. For a predicted cluster p , the *hypergeometric* p-value $H(M, p, g)$ of the association of p to g , when $g \cap p \neq \emptyset$ is:

$$H(M, p, g) = \sum_{i=|p \cap g|}^{\min(|p|, |g|)} \frac{\binom{|M| - |g|}{|p| - i} \binom{|g|}{i}}{\binom{|M|}{|p|}}$$

which represents the probability that a subset of M of size $|p|$ chosen uniformly at random has with g an intersection of size larger than or equal to $|p \cap g|$. As, in general, p will have an hypergeometric score for many gene ontology classes, (Nepusz *et al.*, 2012; Zhang *et al.*, 2008) associate to each cluster p the intersecting gene ontology class of lower p-value. In order to correct for multiple comparisons any version of the Benjamini-Hochberg False Discovery Rate estimation method can be used (Storey & Tibshirani, 2003).

Case Studies

Comparative evaluation of several methods are often found in papers reporting on any new proposed algorithm. However, usually one should pay more attention to third party qualitative or quantitative evaluations, such a those found in Lin *et al.* (2007, 2010), Wange *et al.* (2010), Pizzuti *et al.* (2012), Srihari & Leong (2013), Chen *et al.* (2014), Clancy & Hovig (2014), Pizzuti & Rombo (2014), Srihari *et al.* (2015).

Applications of Community Finding Techniques

Since the turn of the last century several studies have produced comprehensive PPI networks using high throughput proteomic techniques (Gavin *et al.*, 2002; Ho *et al.*, 2002; Von Mering *et al.*, 2002), however in this early stage the extraction of protein complexes, functional modules and pathways from the data was based on direct observation and guided by previous accrued knowledge. Soon after the benefit of a more systematic algorithmic approach to detection/prediction became clear. V. Spirin and L. A. Mirny in their seminal paper (Spirin & Mirny, 2003) study PPI network modularity using three complementary methods. The first method is an exhaustive enumeration of all maximal cliques. The second is application of an early clustering algorithm (Blatt *et al.*, 1996) inspired by a model in statistical physics. The third is a Monte Carlo (MC) simulation aimed at uncovering dense subgraphs of a predetermined size. Gavin *et al.* (2006) use socio-affinity indices to define the weight of edges in the PPIN and then apply iteratively a single linkage hierarchical clustering method to cluster the graph into modules. Both Krogan *et al.* (2006) and by Hart *et al.* (2007) use MCL as main tool to cluster a high quality PPIN graph. Wang *et al.* (2009) develop the algorithm HACO, a variation of the (HAC) Hierarchical Agglomerative Clustering approach, that applies a biologically justified notion of divergence between two clusters. More recently, as richer sets of PPIN were generated, *community detection* methods have been increasingly applied within large scale proteomic studies: MCODE has been used in Li *et al.* (2014) to uncover blood transcription modules (BTM) in a large master network comprising 17,397 genes and 604,363 interactions; and Havugimana *et al.* (2012) use ClusterOne to uncover 622 putative protein complex from a network of high quality of human PPI.

Results and Discussion

Biological Networks

Vidal *et al.* (2011) survey several classes of biological networks: direct interaction networks (metabolic, PPI, gene regulatory) and functional interaction networks (transcriptional profiling, phenotypic profiling, genetic). Many other types of network obtained by heterogeneous data fusion are listed in Gligorić *et al.* (2015). As PPIN constitute the class of biological networks upon which a larger body of knowledge has been built over time, we discuss several trends and problems related to PPIN, although similar issue may arise also for other types of biological networks.

Functional Modules

The notion of a 'functional module' is somewhat general and encompasses a variety of possible schemes and mechanism. For example, a biological pathway (bp), meaning a series of interactions among molecules in a cell that leads to a certain product or a change of status in a cell, as exemplified by, say, the KEGG database (Kanehisa *et al.*, 2011), is a form of functional module where the function is attained by successive protein interactions. On the other side of the spectrum, a protein complex (pc) is also a form

of functional module where the function is attained by a simultaneous interaction of several proteins in an assembled form. Between these two extremes several intermediate configurations can be conceived. In PPIN, a pc may appear as dense subgraphs (for different notions of density), while bp would appear as paths from a trigger protein to one or several target proteins (Pržulj *et al.*, 2004), where in both cases often temporal effects are not encoded in the network. Thus, a notion of functional module based on density may be appropriate for pc detection but may miss pathways or other intermediate notions of functional module, that do not appear to be dense in the chosen graph representation.

Interestingly, several authors have advanced definitions for functional module characterization that use density in a more subtle way. Colak *et al.* (2010) augment the graph $G=(V, E)$, based on PPIN data, with a function $F(\cdot)$ that associates to each vertex $v \in V$ (protein/gene), a gene expression vector $F(v)$ that reflects the expression of v under a variety of conditions and in several tissues, as measured typically in transcriptomic assays. Since F has the structure of a vector space we can define a second graph G_F on the set of vertices V , where two nodes are connected by an edge if the two corresponding vectors are at a distance below a user defined-threshold, under a restricted L_∞ metric. Starting from the intuition that the genes participating in a functional module should have also similar gene expression profiles, the authors aim at finding dense connected bi-clusters, that is subsets $V' \subseteq V$ that induce dense connected subgraphs both in G and in G_F .

Alternatively, Wang & Qian (2013) model functional modules as dense bipartite subgraphs of the master PPIN graph $G=(V, E)$. While a dense bipartite subgraph Q may not be dense in G , in the standard sense, it forms a high weight dense subgraph in the weighted product graph G^2 , where we associate to a pair of vertices $u, v \in V$ the number of 2-paths connecting u and v in G . The authors propose then the algorithm LCP^2 that is an adaptation of the Low Conductance partitioning method in Voevodski *et al.* (2009) when applied to the weighted product graph.

Future Directions

Clustering Versus Community Finding

In this Chapter we have surveyed the basics of community finding in biological networks. Often both clustering techniques (described in Chapter *Cluster analysis of biological networks*) and community finding can be used for the same purpose (e.g. predicting pc in PPIN) and many methods may be considered hybrid approaches, sharing features of both. At a high level clustering techniques are characterized by the minimization of either global objective functions over partitions (e.g. the modularity function) or relative objective functions (e.g. more links inside than outside). At a first approximation, in clustering there is a more complex and subtle competition between modules in order to define their relative boundaries. Community finding instead often try to optimize a simpler local condition (e.g. density above a threshold) and each community is found rather independently from the others. In general, because of this flexibility in the computation, community finding techniques model better overlapping communities. In a practical situation both approaches should be considered and assessed relative to the data and the problem to be solved.

PPIN and Static Data Integration

As PPI networks and the associated analytic tools have been quite successful, there has been much research in methods that retain PPIN as a general scaffold onto which additional biological knowledge is added.

For example, the conservation of protein complexes across species as an additional constraint is studied in Nguyen *et al.* (2013). Jung *et al.* (2010) encode in PPIN the information on mutually exclusive interactions. Proteins in PPIN can also be marked with cellular localization annotations, GO annotations, gene expression, and several types of quality score (Veres *et al.*, 2015). All these aspects are important, and they are possible refinements of the graph model compatible with the majority of the algorithms for finding communities, involving the modelling of additional knowledge in the PPIN framework (see Xu *et al.*, 2013).

This line of research is quite fruitful and still productive, however, it has been pursued often with ad hoc techniques that are difficult to generalize to more than two or three types of data sources simultaneously. As the field of 'data integration' has been attracting attention from different areas, including computer science and computer engineering, it is likely that more robust approaches able to integrate a variety of data sources, could be based on general data integration principles (e.g. that in Žitnik & Zupan (2015)) based on matrix factorization, or on Bayesian inference (Jansen *et al.*, 2003). See also (Gligorićević *et al.*, 2015) for an overview.

PPIN and Dynamic Data Integration

The augmentation of PPIN with dynamic and temporal data deserves a separate treatment from static data integration. While several methods for modelling dynamic behaviour into PPIN have been proposed, the main obstacle to pursuing this line of research appears to be the lack of reliable data. Most high throughput genomic, transcriptomic and proteomic techniques are able to give a snapshot of the status of a cell (or a biological system) at a certain point in time, in a cost effective way. As of now, tracing the evolution over time of the molecular components of a biological system is either confined to few molecules in each experiment, or it is particularly laborious and expensive. Thus, in this area progress may come more readily from new measurement techniques, rather than from a purely "in silico" approach. In particular detecting experimentally the concentrations and

rates of expression over short time ranges is hard and often one has to use simulations instead (Clancy & Hovig, 2014). A line of research has attempted to encode in graph terms timing information modelled in explicit form as time sequences (Mucha *et al.*, 2010; Tang *et al.*, 2011).

Multispecies PPIN

Although PPINs are often built for a single species, data accumulated over time in repositories allows to build multi-species networks (Park *et al.*, 2011; Franceschini *et al.*, 2013), where one connects homologous proteins across different species to form layered graphs. Multi-species PPIN coupled with functional module and pc detection may uncover both conserved module/complexes, but also indicate possible missing edges, and suggest alternative undiscovered control paths by inter-species homology. The issue of evolutionary conservation of features across PPIN is surveyed in Jancura & Marchiori (2012).

PPIN Specialization

The early PPINs were built from one or several biological samples and then all interaction found were merged in a single data structure (PPI integration) thus usually forming a static integrated picture of protein activities in the cell. A recent trend instead aims at building (and analyzing) specialized PPIN that reflect specific conditions. For example tissue and tumor specific PPIN are described in Micale *et al.* (2015). A second line of research tries to take into account internal cellular architecture (e.g. lipid compartment) to refine the spatial resolution of the PPIN (Clancy & Hovig, 2014). In Immunological studies, PPIN representing the immune system response to external agents have been produced (Li *et al.*, 2014). In general this operation can be defined a *PPIN differentiation* as it tries to untangle and control the PPIN response within a specific restricted context (Ideker & Krogan, 2012).

Special Types of Communities

Most community prediction methods in PPIN fail when the protein complex (or the functional module) has features at odds with the assumption of the methods (in particular density and connectivity), thus specialized approaches for small pc and very sparse functional modules (e.g pathways) are needed (Srihari *et al.*, 2015). Although established community finding algorithms handle overlapping modules in a natural way, still finding biologically meaningful overlaps is challenging for more recent approaches (Pizzuti & Rombo, 2014).

The implications of the overlapping of modules on their functionality (competition, cooperation) has not been researched in depth so far. Some preliminary results on sub-complexes and on the characterization of the overall ‘complexome’ are giving some first glimpses on this issue (Zaki & Mora, 2014; Ahnert *et al.*, 2015).

The Complexome

A PPI network is a collection pairwise protein interactions in which communities (complexes, functional units) appear as dense sub-networks. Going one step further several authors (Lee *et al.*, 2011; Clancy & Hovig, 2014) propose the notion of the “complexome” as the study of the organizations of the functional units and complexes within the proteome. In this new scenario, a complexome network is a collection of pairwise functional interactions between pairs of functional units (aka complex-complex networks - CCIN). In this context, it is easier to frame questions relative to transient inter-complex interactions, and take advantage of inter-species evolutionary conservation of complexes functional characterization, even when the proteins involved might not be homologous.

Benchmarking and Resource Portals

PPI detection technologies, database and algorithms have matured in the last few years (Cannataro *et al.*, 2010) and on the side of data several state-of-the-art repositories and databases are available. However, on the side of algorithms the situation is somewhat fragmented. Many sw tools are available for downloading as stand alone applications, some are available via web interface, but many are no longer maintained and hard to find and install. The integrated environment Cytoscape has a number of algorithms available as plugins (Morris *et al.*, 2011), and other methods are available in social network analysis tools (e.g. NetworkX (Hagberg *et al.*, 2008)), and SNAP (Leskovec and Sosis, 2016). Clearly more could be done to offer in an integrated platform a wealth of algorithmic solutions to the problem of community finding in biological networks of several types. There is strong need for standardization in many steps of the analysis pipeline (Goh & Wong, 2016).

Closing Remarks

Searching for communities in biological networks is a fascinating subject of research, in which many different approaches and disciplines contribute original insights. The concept of *community* in a biological network is quite general and flexible but needs to be adapted to the specific biological question to be answered within the system's biology paradigm of network modelling. After

more than a decade of progress, we are probably quite ready for a step forward both in the type and quality of the new (mainly dynamic) of data that can be integrated in the network model, and in the quantitative and qualitative performance of new and improved community finding approaches and methodologies. Here, both the technological advances in high throughput assays and new insights of algorithmic nature will play a leading role.

Acknowledgment

Work supported by Italian Ministry of Education, Universities and Research (MIUR) and by the National Research Council of Italy (CNR) within the Flagship Project InterOmics PB.P05. (CUIP B91J12000270001).

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Algorithms for Structure Comparison and Analysis: Docking. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Quantification of Proteins from Proteomic Analysis

References

- Adamcsek, B., Palla, G., Farkas, I.J., Derenyi, I., Vicsek, T., 2006. Cfinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* 22 (8), 1021–1023. Available at: <https://doi.org/10.1093/bioinformatics/btl039>.
- Ahnert, S.E., Marsh, J.A., Hernández, H., Robinson, C.V., Teichmann, S.A., 2015. Principles of assembly reveal a periodic table of protein complexes. *Science* 350 (6266), aaa2245.
- Bader, G., Hogue, C., 2003. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4 (1), 2. Available at: <http://www.biomedcentral.com/1471-2105/4/2>.
- Batagelj, V., Zaversnik, M., 2003. An $O(m)$ algorithm for cores decomposition of networks, CoRR cs.DS/0310049.
- Blatt, M., Wiseman, S., Domany, E., 1996. Superparamagnetic clustering of data. *Physical review letters* 76 (18), 3251.
- Bron, C., Kerbosch, J., 1973. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM* 16 (9), 575–577.
- Cannataro, M., Guzzi, P.H., Veltri, P., 2010. Protein-to-protein interactions: Technologies, databases, and algorithms. *ACM Computing Surveys (CSUR)* 43 (1), 1.
- Charikar, M., 2000. Greedy approximation algorithms for finding dense components in a graph. In: Jansen, K., Khuller, S. (Eds.), *APPROX, Lecture Notes in Computer Science* 1913. New York, NY: Springer, pp. 84–95.
- Chen, B., Fan, W., Liu, J., Wu, F.-X., 2014. Identifying protein complexes and functional modules: From static ppi networks to dynamic ppi networks. *Briefings in bioinformatics* 15 (2), 177–194.
- Chua, H.N., Sung, W.-K., Wong, L., 2006. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics* 22 (13), 1623–1630.
- Clancy, T., Hovig, E., 2014. From proteomes to complexomes in the era of systems biology. *Proteomics* 14 (1), 24–41.
- Colak, R., Moser, F., Chu, J.S.-C., et al., 2010. Module discovery by exhaustive search for densely connected, co-expressed regions in biomolecular interaction networks. *PIOS One* 5 (10), e13348.
- Franceschini, A., Szklarczyk, D., Frankild, S., et al., 2013. String v9.1: Protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research* 41 (D1), D808–D815. Available at: <http://nar.oxfordjournals.org/content/41/D1/D808.abstract>.
- Gavin, A.-C., Aloy, P., Grandi, P., et al., 2006. Proteome survey reveals modularity of the yeast cell machinery. *Nature* 440 (7084), 631.
- Gavin, A.-C., Bösch, M., Krause, R., et al., 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415 (6868), 141–147.
- Glorigorjević, V., Pržulj, N., 2015. Methods for biological data integration: Perspectives and challenges. *Journal of the Royal Society Interface* 12 (112), 20150571.
- Goh, W.W.B., Wong, L., 2016. Integrating networks and proteomics: Moving forward. *Trends in Biotechnology* 34 (12), 951–959.
- Hagberg, A.A., Schult, D.A., Swart, P.J., 2008. Exploring network structure, dynamics, and function using NetworkX. In: *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA, pp. 11–15.
- Hanna, E., Zaki, N., 2014. Detecting protein complexes in protein interaction networks using a ranking algorithm with a refined merging procedure. *BMC Bioinformatics* 15 (1), 204. Available at: <http://www.biomedcentral.com/1471-2105/15/204>.
- Hart, G.T., Lee, I., Marcotte, E.M., 2007. A high-accuracy consensus map of yeast protein complexes reveals modular nature of gene essentiality. *BMC Bioinformatics* 8 (1), 236.
- Havugimana, P.C., Hart, G.T., Nepusz, T., et al., 2012. A census of human soluble protein complexes. *Cell* 150 (5), 1068–1081.
- Ho, Y., Gruhler, A., Heilbut, A., et al., 2002. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* 415 (6868), 180–183.
- Ideker, T., Krogan, N.J., 2012. Differential network biology. *Molecular Systems Biology* 8 (1), 565.
- Jancura, P., Marchiori, E., 2012. A survey on evolutionary analysis in ppi networks. *Protein-Protein Interactions-Computational and Experimental Tools*. InTech.
- Jansen, R., Yu, H., Greenbaum, D., et al., 2003. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* 302 (5644), 449–453. Available at: <http://www.sciencemag.org/content/302/5644/449.abstract>.
- Jiang, P., Singh, M., 2010. Spici: A fast clustering algorithm for large biological networks. *Bioinformatics* 26 (8), 1105–1111. Available at: <http://bioinformatics.oxfordjournals.org/content/26/8/1105.abstract>.
- Jung, S.H., Hyun, B., Jang, W.-H., Hur, H.-Y., Han, D.-S., 2010. Protein complex prediction based on simultaneous protein interaction network. *Bioinformatics* 26 (3), 385–391. Available at: <http://bioinformatics.oxfordjournals.org/content/26/3/385.abstract>.
- Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., Tanabe, M., 2011. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research* 40 (D1), D109–D114.
- Krogan, N.J., Cagney, G., Yu, H., et al., 2006. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature* 440 (7084), 637.
- Kuhn, H.W., 1955. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)* 2 (1-2), 83–97.
- Lee, S.H., Kim, P.-J., Jeong, H., 2011. Global organization of protein complexome in the yeast *Saccharomyces cerevisiae*. *BMC Systems Biology* 5 (1), 126.
- Leskovec, J., Soscic, R., 2016. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8 (1), 1.

- Li, S., Rouphael, N., Duraisingham, S., *et al.*, 2014. Molecular signatures of antibody responses derived from a systems biological study of 5 human vaccines. *Nature Immunology* 15 (2), 195–204.
- Li, X., Wu, M., Kwok, C.-K., Ng, S.-K., 2010. Computational approaches for detecting protein complexes from protein interaction networks: A survey. *BMC Genomics* 11 (Suppl 1), S3. Available at: <http://www.biomedcentral.com/1471-2164/11/S1/S3>.
- Lin, C., Cho, Y.-R., Hwang, W.-C., Pei, P., Zhang, A., 2007. *Clustering Methods in a Protein Protein Interaction Network*. Hoboken, NJ: John Wiley & Sons, Inc., pp. 319–355.
- Liu, G., Wong, L., Chua, H.N., 2009. Complex discovery from weighted ppi networks. *Bioinformatics* 25 (15), 1891–1897. Available at: <http://bioinformatics.oxfordjournals.org/content/25/15/1891.abstract>.
- Micale, G., Ferro, A., Pulvirenti, A., Giugno, R., 2015. Spectra: An integrated knowledge base for comparing tissue and tumor-specific ppi networks in human. *Frontiers in Bioengineering and Biotechnology* 3.
- Morris, J.H., Apeltin, L., Newman, A.M., *et al.*, 2011. clustermaker: A multi-algorithm clustering plugin for cytoscape. *BMC Bioinformatics* 12 (1), 436.
- Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.-P., 2010. Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328 (5980), 876–878.
- Nepusz, T., Yu, H., Paccanaro, A., 2012. Detecting overlapping protein complexes in protein protein interaction networks. *Nature Methods* 9, 471–472.
- Nguyen, P.-V., Srihari, S., Leong, H.W., 2013. Identifying conserved protein complexes between species by constructing interolog networks. *BMC Bioinformatics* 14 (S-16), S8.
- Palla, G., Derenyi, I., Farkas, I., Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435 (7043), 814–818.
- Park, D., Singh, R., Baym, M., Liao, C.-S., Berger, B., 2011. Isobase: A database of functionally related proteins across ppi networks. *Nucleic Acids Research* 39 (suppl 1), D295–D300.
- Pellegrini, M., Baglioni, M., Geraci, F., 2016. Protein complex prediction for large protein protein interaction networks with the Core&Peel method. *BMC Bioinformatics* 17 (Suppl 12), 37–58.
- Peng, W., Wang, J., Zhao, B., Wang, L., 2014. Identification of protein complexes using weighted pagerank-nibble algorithm and core-attachment structure. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12 (1), 179–192.
- Pržulj, N., Wigle, D.A., Jurisica, I., 2004. Functional topology in a network of protein interactions. *Bioinformatics* 20 (3), 340–348.
- Pizzuti, C., Rombo, S.E., 2014. Algorithms and tools for protein-protein interaction networks clustering, with a special focus on population-based stochastic methods. *Bioinformatics* 30 (10), 1343–1352.
- Pizzuti, C., Rombo, S., Marchiori, E., 2012. Complex detection in protein-protein interaction networks: A compact overview for researchers and practitioners. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. 211–223.
- Seidman, S.B., 1983. Network structure and minimum degree. *Social Networks* 5 (3), 269–287. Available at: <http://www.sciencedirect.com/science/article/pii/037887338390028X>.
- Song, J., Singh, M., 2009. How and when should interactome-derived clusters be used to predict functional modules and protein function? *Bioinformatics* 25 (23), 3143–3150. Available at: <http://bioinformatics.oxfordjournals.org/content/25/23/3143.abstract>.
- Spirin, V., Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences* 100(21), 12123–12128.
- Srihari, S., Leong, H.W., 2013. A survey of computational methods for protein complex prediction from protein interaction networks. *Journal of Bioinformatics and Computational Biology* 11 (2).
- Srihari, S., Yong, C.H., Patil, A., Wong, L., 2015. Methods for protein complex prediction and their contributions towards understanding the organisation, function and dynamics of complexes. *FEBS Letters* 589 (19), 2590–2602.
- Storey, J.D., Tibshirani, R., 2003. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences* 100 (16), 9440–9445. Available at: <http://www.pnas.org/content/100/16/9440.abstract>.
- Tang, X., Wang, J., Liu, B., *et al.*, 2011. A comparison of the functional modules identified from time course and static ppi network data. *BMC Bioinformatics* 12 (1), 339.
- Tomita, E., Tanaka, A., Takahashi, H., 2006. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science* 363 (1), 28–42.
- Veres, D.V., Gyurko, D.M., Thaler, B., *et al.*, 2015. Compipi: A cellular compartment-specific database for protein-protein interaction network analysis. *Nucleic Acids Research* 43 (Database issue (2015)), D485–D493.
- Vidal, M., Cusick, M.E., Barabasi, A.-L., 2011. Interactome networks and human disease. *Cell* 144 (6), 986–998.
- Voevodski, K., Teng, S.-H., Xia, Y., 2009. Finding local communities in protein networks. *BMC Bioinformatics* 10 (1), 297.
- Von Mering, C., Krause, R., Snel, B., *et al.*, 2002. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* 417 (6887), 399–403.
- Wang, H., Kakaradov, B., Collins, S.R., *et al.*, 2009. A complex-based reconstruction of the *Saccharomyces cerevisiae* interactome. *Molecular & Cellular Proteomics* 8 (6), 1361–1381.
- Wang, J., Li, M., Deng, Y., Pan, Y., 2010. Recent advances in clustering methods for protein interaction networks. *BMC Genomics* 11 (Suppl 3), S10. Available at: <http://www.biomedcentral.com/1471-2164/11/S3/S10>.
- Wang, Y., Qian, X., 2013. Functional module identification in protein interaction networks by interaction patterns. *Bioinformatics* 30 (1), 81–93.
- Wu, M., Li, X., Kwok, C.K., Ng, S.-K., 2009. A core-attachment based method to detect protein complexes in ppi networks. *BMC Bioinformatics* 10.
- Xu, B., Lin, H., Chen, Y., Yang, Z., Liu, H., 2013. Protein complex identification by integrating protein-protein interaction evidence from multiple sources. *PLOS ONE* 8 (12), e83841.
- Zaki, N., Berengueres, J., Efimov, D., 2012. Detection of protein complexes using a protein ranking algorithm. *Proteins: Structure, Function, and Bioinformatics* 80 (10), 2459–2468. Available at: <https://doi.org/10.1002/prot.24130>.
- Zaki, N., Mora, A., 2014. A comparative analysis of computational approaches and algorithms for protein subcomplex identification. *Scientific Reports* 4.
- Zhang, B., Park, B.-H., Karpinet, T., Samatova, N.F., 2008. From pull-down data to protein interaction networks and complexes with biological relevance. *Bioinformatics* 24 (7), 979–986. Available at: <http://bioinformatics.oxfordjournals.org/content/24/7/979.abstract>.
- Žitnik, M., Zupan, B., 2015. Data fusion by matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (1), 41–53.

Further Reading

- Cannataro, M., Guzzi, P.H., Veltri, P., 2010. Protein-to-protein interactions: Technologies, databases, and algorithms. *ACM Computing Surveys (CSUR)* 43 (1), 1.
- Chen, B., Fan, W., Liu, J., Wu, F.-X., 2014. Identifying protein complexes and functional modules: From static ppi networks to dynamic ppi networks. *Briefings in bioinformatics* 15 (2), 177–194.
- Clancy, T., Hovig, E., 2014. From proteomes to complexomes in the era of systems biology. *Proteomics* 14 (1), 24–41.
- Gligorijević, V., Pržulj, N., 2015. Methods for biological data integration: Perspectives and challenges. *Journal of the Royal Society Interface* 12 (112), 20150571.
- Goh, W.W.B., Wong, L., 2016. Integrating networks and proteomics: Moving forward. *Trends in Biotechnology* 34 (12), 951–959.
- Lee, S.H., Kim, P.-J., Jeong, H., 2011. Global organization of protein complexome in the yeast *Saccharomyces cerevisiae*. *BMC Systems Biology* 5 (1), 126.
- Li, X., Wu, M., Kwok, C.-K., Ng, S.-K., 2010. Computational approaches for detecting protein complexes from protein interaction networks: A survey. *BMC Genomics* 11 (Suppl 1), S3. Available at: <http://www.biomedcentral.com/1471-2164/11/S1/S3>.
- Lin, C., Cho, Y.-R., Hwang, W.-C., Pei, P., Zhang, A., 2007. *Clustering Methods in a Protein Protein Interaction Network*. Hoboken, NJ: John Wiley & Sons, Inc., pp. 319–355.
- Pizzuti, C., Rombo, S.E., 2014. Algorithms and tools for protein-protein interaction networks clustering, with a special focus on population-based stochastic methods. *Bioinformatics* 30 (10), 1343–1352.

- Pizzuti, C., Rombo, S., Marchiori, E., 2012. Complex detection in protein-protein interaction networks: A compact overview for researchers and practitioners. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. 211–223.
- Srihari, S., Leong, H.W., 2013. A survey of computational methods for protein complex prediction from protein interaction networks. *Journal of Bioinformatics and Computational Biology* 11 (2).
- Srihari, S., Yong, C.H., Patil, A., Wong, L., 2015. Methods for protein complex prediction and their contributions towards understanding the organisation, function and dynamics of complexes. *FEBS Letters* 589 (19), 2590–2602.
- Srihari, S., Yong, C.H., Wong, L., 2017. Computational Prediction of Protein Complexes from Protein Interaction Networks. Morgan & Claypool.
- Vidal, M., Cusick, M.E., Barabasi, A.-L., 2011. Interactome networks and human disease. *Cell* 144 (6), 986–998.
- Wang, J., Li, M., Deng, Y., Pan, Y., 2010. Recent advances in clustering methods for protein interaction networks. *BMC Genomics* 11 (Suppl 3), S10. Available at: <http://www.biomedcentral.com/1471-2164/11/S3/S10>.

Protein–Protein Interaction Databases

Max Kotlyar, University Health Network, Toronto, ON, Canada

Chiara Pastrello and Andrea EM Rossos, Krembil Research Institute, Toronto, ON, Canada

Igor Jurisica, University of Toronto, ON, Canada and Slovak Academy of Sciences, Bratislava, Slovakia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Protein-protein interaction (PPI) databases are key resources for life sciences research, alongside genomic, transcriptomic, and proteomic databases. The importance of PPI databases reflects the fundamental role of protein interactions in cellular processes; almost all cellular activities, such as metabolism, growth, and repair, occur through interactions between proteins. Knowledge of PPIs is essential for understanding phenotypes at a molecular level, including the mechanisms underlying diseases and treatments. Consequently, PPI datasets are used in diverse analytic workflows throughout health science research, including but not limited to discovery of disease genes (Wang *et al.*, 2011), drug discovery (De Las Rivas and Prieto, 2012), and prediction of gene function (Mostafavi and Morris, 2012). Interaction databases facilitate these applications by integrating PPI data from numerous sources, assessing data reliability, providing extensive query capabilities, and returning results in a variety of formats.

Although PPI databases are used in diverse projects, users generally have the same basic requirements: obtaining a high-quality, context-specific PPI network, as easily as possible. A high-quality network should have few false-negatives (i.e., missing interactions) and false-positives (i.e., false interactions), and ideally, should be specific to the conditions being studied (e.g., tissue, disease). The ease of obtaining a network can include factors such as the types of protein (gene) identifiers that can be used for querying the database, available options for filtering results (e.g., by reliability, tissue), and available download formats.

This article reviews how PPI databases meet these requirements, including their approaches for compiling high-quality, context-specific PPI networks and methods for facilitating access to these networks. Strategies for compiling networks include data integration to reduce missing interactions, scoring interaction evidence to reduce false positives, and overlaying networks with gene and protein expression data to provide context. Approaches to facilitate access include multiple options for query IDs, data filters, and download formats. This article provides an overview of these topics, including recent progress and remaining challenges.

Data Integration

Protein-protein interaction data comes from thousands of different studies employing small-scale experiments, high-throughput (HT) screens, and computational predictions to identify interactions. Initially, PPIs were identified in small-scale experiments; studies detected small numbers of interactions and often performed validation by several different bioassays to ensure reliability. Such studies continue to be important and trusted sources of PPI data. Over 38,000 small-scale studies of human PPIs have been published, providing more than 73,000 interactions (Kotlyar *et al.*, 2016).

In the 1990s, high-throughput (HT) screens were developed for detecting many more interactions than small-scale experiments, typically from a few hundred to several thousand. Approximately 400 human HT studies have reported over 210,000 interactions (Kotlyar *et al.*, 2016).

Computational methods began to be used in the late 1990s to predict PPIs. Numerous published prediction methods have reported over 650,000 high-confidence human predicted PPIs (Kotlyar *et al.*, 2016). However, predicted PPIs have not been widely used in health sciences research, despite increasing evidence for their reliability (Elefsinioti *et al.*, 2011; Zhang *et al.*, 2012; Kotlyar *et al.*, 2015).

The creation of PPI databases was driven by the need to integrate numerous disparate sources of PPI data into comprehensive, easily usable PPI networks, a task that is exceedingly difficult for individual researchers working with original PPI publications. Integration is difficult due to several main factors: PPIs are reported in thousands of articles appearing in diverse journals, PPIs and experimental conditions are typically reported as unstructured text, and PPIs may be reported using many different protein and gene identifiers. Overcoming these integration challenges required accomplishing several key tasks: (1) defining a standard data format and a controlled vocabulary for describing PPI detection experiments, (2) establishing accurate mapping between protein and gene identifiers, and (3) curating thousands of journal articles.

Curation is carried out by a number of major databases including BioGRID (Chatr-aryamontri *et al.*, 2017), DIP (Salwinski *et al.*, 2004), IntAct (Orchard *et al.*, 2014), and MINT (Ceol *et al.*, 2010). Many of these databases have been jointly addressing the above challenges, and have formed the International Molecular Exchange (IMEx) consortium (Orchard *et al.*, 2012) to co-ordinate their efforts. Most of these databases obtain all of their data by curating peer-reviewed publications, and are referred to as primary databases (see Table 1). These databases are commonly used sources of data for health sciences research.

Standardizing Data Format and Vocabulary

Data formats and vocabulary for PPI detection experiments were established by the Molecular Interaction (MI) workgroup of the Human Proteome Organization –Proteomics Standards Initiative (HUPO-PSI) (Kerrien *et al.*, 2007). The formats included both

Table 1 PPI databases that include human interactions

Resource type	Resource	URLs	Number of PPIs for <i>Homo sapiens</i>	Inputs	Output – download	PPI annotations
Primary databases	BioGrid (Stark <i>et al.</i> , 2006)	https://thebiogrid.org	312,107	51 Different IDs	PSI-MI, Osprey, delimited text	Score, evidences, modification, phenotype, database
	DIP (Xenarios <i>et al.</i> , 2000)	http://dip.doe-mbi.ucla.edu/dip/Main.cgi	9143	PIR, Uniprot, RefSeq, gene symbol, protein sequence, motif, article	NA	None
	HPRD (Keshava Prasad <i>et al.</i> , 2009)	www.hprd.org	41,327	16 different IDs or classes of proteins	NA	Evidences
	IntAct (Orchard <i>et al.</i> , 2014)	http://www.ebi.ac.uk/intact/	~373,400 (not clearly listed)	Gene symbol, UniProt, ChEBI, RNACentral, IMEx	PSI-MI, XGMML, Biopax, RDF	Evidences, database
	MIPS (Pagel <i>et al.</i> , 2005)	http://mips.helmholtz-muenchen.de/proj/ppi/	Not listed	Gene symbol, X-Ref	NA	Evidences
	MatrixDB (Chautard <i>et al.</i> , 2009)	http://matrixdb.univ-lyon1.fr/	789	Gene symbol, UniProt, EBI	NA	Evidences, extracellular matrix
Secondary databases	UniHI (Kalathur <i>et al.</i> , 2014)	http://www.unihl.org/	573,995	Gene symbol	Delimited text	Database
	HIPPIE (Schaefer <i>et al.</i> , 2012)	http://cbdm-01.zdv.uni-mainz.de/~mschaefer/hippie/index.php	Not listed	Gene symbol, UniProt, Entrez	Delimited text	Score
	HiTPredict (Schwikowski <i>et al.</i> , 2000)	http://hintdb.hgc.jp/http/	262,933	Keyword, UniProt, Entrez, Ensembl	Delimited text, MITAB-2.5	3 Single scores, Confidence, evidences
	InnateDB (Breuer <i>et al.</i> , 2013)	http://www.innatedb.ca	22,882 predicted 318,107 validated	15 Different IDs	Delimited text, Image, PSI-MI, SIF, XGMML	Cell type, tissue, database, evidences
	IID (Kotlyar <i>et al.</i> , 2016)	http://ophid.utoronto.ca/iid/	280,844 Experimental 59,048 Orthologous 664,643 Predicted	Gene symbol, Ensembl, Entrez, UniProt	Delimited text	Evidence, tissue
	HAPPI-2 (Chen <i>et al.</i> , 2017)	http://discovery.informatics.uab.edu/HAPPI/	342,599 Reliable interaction – score ≥ 0.75	Gene symbol, keywords	NA	Evidence, quality, effect, directionality
	PrePPI (Zhang <i>et al.</i> , 2012)	http://bhapp.c2b2.columbia.edu/PrePPI/	1.35 million	Gene symbol, UniProt	NA	9 Predictions scores, 3 experimental scores, 1 total score
	STRING (Szklarczyk <i>et al.</i> , 2017)	https://string-db.org/	Not listed	Gene symbol, protein ID (not specified), sequence	delimited text, XML, graphics formats	score, evidences
	APID (Alonso-López <i>et al.</i> , 2016)	apid.dep.usal.es	408,610	Gene symbol, UniProt	delimited text, PDF	Evidences
						(Continued)

Table 1 Continued

Resource type	Resource	URLs	Number of PPIs for <i>Homo sapiens</i>	Inputs	Output – download	PPI annotations
Prediction databases	iRefWeb (Turner <i>et al.</i> , 2010)	http://wiodaklab.org/iRefWeb/	222,098	Gene symbol, UniProt, Entrez	iRef Interaction IDs, MITAB, MINI-TAB	Evidences, score, percentile
	GPS-Prot (Fahey <i>et al.</i> , 2011)	gpsprot.org	284,864	Gene symbol, UniProt	NA	
	MyProteinNet (Basha <i>et al.</i> , 2015)	http://netbio.bgu.ac.il/myproteinnet/	Not listed	NA	Delimited text	Evidence, score, directionality
	CompPPI (Veres <i>et al.</i> , 2015)	compipi.linkgroup.hu	385,481	Gene symbol	Delimited text	Evidences, Score, Database, Subcellular localization
	TissueNet (Barshir <i>et al.</i> , 2013)	http://netbio.bgu.ac.il/tissuenet/#/	~240,000	Gene symbol, Entrez, Ensemble	NA	Number of tissues, Expression
	SPECTRA (Micale <i>et al.</i> , 2015)	https://alpha.dmi.unict.it/spectra/	Not listed	Gene symbol, Entrez, Ensemble, UniProt, probeset ID	Delimited text	Tissue, Disease
	FpClass (Kotlyar <i>et al.</i> , 2015)	http://ophid.utoronto.ca/fpclass/	250,498 high-conf	Gene symbol, Ensembl, Entrez, UniProt, RefSeq	Delimited text	1 Total score, 11 single scores
	hPRINT (Elefsinioti <i>et al.</i> , 2011)	http://print-db.org	117,360 rf.phys > 0.7	Gene symbol, Ensembl, Entrez	Delimited text	3 Single scores, 32 features
	PIPs (Scott and Barton, 2007)	http://www.compbio.dundee.ac.uk/www-pips/	37,606 Score ≥ 1	IPI, RefSeq, UniProt	NA	Score, evidences, database
	Struct2Net (Singh <i>et al.</i> , 2010)	http://cb.csail.mit.edu/cb/struct2net/webserver/	Dynamic query	Gene symbol, Ensembl, EntrezGene, RefSeq, UniProtKB, FlyBase, SGD, protein sequence	PSI-MI, delimited text	Score
	PRISM (Baspinar <i>et al.</i> , 2014)	http://cosbi.ku.edu.tr/prism/	33,853	PDB	NA	Energy
	iLoops (Planas-Iglesias <i>et al.</i> , 2013)	http://sbi.imim.es/iLoopsServer/	Dynamic query	Protein sequence	XML	Score, precision
	BIPS (Garcia-Garcia <i>et al.</i> , 2012)	http://sbi.imim.es/web/index.php/research/servers/bips	Dynamic query	gene symbol, Tair, UniProt, protein sequence	delimited text	Evidences

Evidences = experiments, methods, publications.

XML (PSI-MI XML) and tab-delimited text (MITAB) (Kerrien *et al.*, 2007). The controlled vocabulary provided an unambiguous way to describe all aspects of a detection experiment, such as the type of detection method and the type of molecular interaction identified (e.g., direct interaction, association) (Kerrien *et al.*, 2007). All major primary databases now use these formats and vocabulary (Orchard, 2012).

Mapping to Common Identifiers

Most current PPI databases uniquely identify interacting proteins using UniProtKB (Consortium, 2017) identifiers. These identifiers can represent not only all full length proteins, but also peptides, fusion proteins, specific isoforms, and proteins whose isoforms are not specified. However, PPI studies and resources sometimes use other identifiers such as gene and protein names, Ensembl IDs (Aken *et al.*, 2016), Entrez IDs (Maglott *et al.*, 2011), RefSeq IDs (Pruitt *et al.*, 2012), and species-specific IDs. Several tools help provide mappings between identifiers (Schmidt and Frishman, 2006; Côté *et al.*, 2007; Huang *et al.*, 2011), but in some cases unambiguous mappings are not possible.

Article Curation

Curation of PPI articles is the most labor-intensive step in PPI data integration. Curation requires extracting information about PPI experiments from unstructured text, and representing this information using a standard vocabulary and format. Several approaches have been considered for performing this task: manual curation by trained biocurators, automated curation aided by text mining, and curation by original authors (i.e., authors of PPI studies curate their own experiments and data). Curation by professional biocurators has provided the best results; automated methods could not achieve the same reliability or level of curation detail (Krallinger *et al.*, 2008; Wang *et al.*, 2016), while curation by authors was unreliable (Leitner *et al.*, 2010), and was not strongly supported by authors (Ceol *et al.*, 2008; Orchard and Hermjakob, 2015). However, manual curation, even by professionals, can be an error-prone (Cusick *et al.*, 2009), difficult, and slow process. Several strategies have been used to make curation easier and more reliable. The IMEx consortium created guidelines for publishing results of PPI experiments (Orchard *et al.*, 2007), to ensure that papers provide necessary, unambiguous information about PPI experiments – thereby improving PPI data and simplifying curation. The IMEx consortium also created guidelines for curation, to clarify the process and encourage consistency. Text-mining methods have been developed to assist curation (Kim *et al.*, 2016). An automated system, the PSI validator (Montecchi-Palazzi *et al.*, 2009), was implemented to check the syntax and semantics of IMEx-compliant curation results. To further ensure reliability, IMEx databases routinely select papers for review by multiple curators (Orchard *et al.*, 2012). These strategies have made curation more reliable, and sometimes easier, but it remains a complex, labor-intensive task. Curating all the published literature with PPI information, and keeping up with new publications presents a daunting challenge. To partially address this problem, IMEx participants have divided the curation workload, each curating a different set of journals (Orchard *et al.*, 2012).

Integrating Multiple PPI Databases and Predicted PPIs

Each major primary database (i.e., whose content is from curation) contains interactions not found in other primary databases. Consequently, obtaining a comprehensive PPI dataset requires content from multiple primary databases. Such comprehensive data can be easily obtained in two ways: from secondary databases (or meta-databases) and from the PSICQUIC service (Aranda *et al.*, 2011). Secondary databases such as APID (Prieto and De Las Rivas, 2006) and PINA (Wu *et al.*, 2009) integrate the contents of multiple primary databases. Some secondary databases such as IID (Kotlyar *et al.*, 2016) and STRING (Szklarczyk *et al.*, 2017) also include predicted PPIs, to provide datasets with more comprehensive proteome coverage. The PSICQUIC interface allows searching multiple PSI-MI compliant databases and returns up-to-date, combined results. Fig. 1 shows an overview of PPI database types and Table 1 provides a comprehensive listing of currently available PPI databases.

Errors in PPI Databases

Sources and Rates of Errors

Errors in a PPI database can be quantified as the false discovery rate (FDR): the fraction of false interactions in the database (i.e., false positives/(true positives + false positives)). False positives in PPI databases (i.e., non-interacting protein pairs) may come from errors in PPI experimental studies or in curation. Quantifying these error rates has been difficult. Error rates of PPI studies depend on how they are estimated – one approach is to retest a subset of detected PPIs by a different assay, but results will depend on the type of assay. Another approach is to compare detected PPIs against previously published PPI datasets, but results will depend on the choice of datasets. Hart *et al.* (2007) calculated FDRs for 7 HT studies as the averages of multiple estimates: resulting FDRs ranged from 35% to 83%. Such different rates can be due to multiple factors including the type of bioassay, the experimental conditions, and the types of proteins being studied. The error rate in curation was estimated at 45% by Cusick *et al.* (2009), based on re-curating some of the interactions in major primary databases. However, the error rates of PPI studies and of curation have

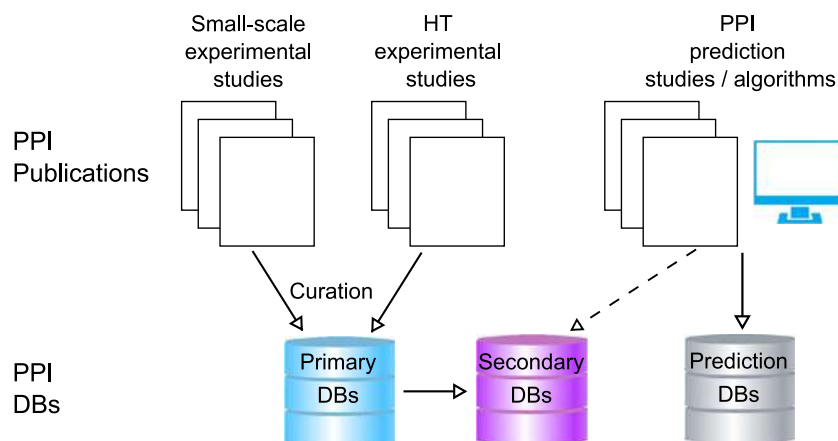


Fig. 1 Types of PPI databases and their sources of data.

likely decreased since the publication of the studies by [Hart et al. \(2006\)](#) and [Cusick et al. \(2009\)](#), due to the introduction of multiple quality control measures.

Improving Data Reliability

To improve reliability, PPI databases have to control errors from two sources – curation and PPI studies. Curation errors have been reduced through quality control measures such as automated syntactic and semantic checking ([Montecchi-Palazzi et al., 2009](#)), and cross-curation ([Orchard et al., 2012](#)), as mentioned in Section Article Curation. The error rate due to PPI studies, as well as curation, can be controlled by assigning confidence scores to interactions and filtering by these scores, although this also reduces the number of returned interactions.

Currently, there is no generally accepted method of calculating confidence scores. Different PPI databases, such as IntAct ([Orchard et al., 2014](#)), iRefWeb ([Turinsky et al., 2014](#)), and HAPPI-2 ([Chen et al., 2017](#)), have different methods of calculating scores. IntAct ([Orchard et al., 2014](#)) scores are weighted sums comprising four types of information: the number of publications reporting the interaction, the number of times the interaction was detected, the types of detection methods (e.g., protein complementation assay, imaging technique), and the types of interactions these methods detect (e.g., direct association, co-localization). Weights assigned to detection methods reflect reliability, while weights assigned to interaction types reflect similarity to direct interaction (i.e., ‘direct association’ receives the highest weight, while ‘colocalization’ receives the lowest). Scores from iRefWeb ([Turinsky et al., 2014](#)), based on the ones proposed by MINT ([Ceol et al., 2010](#)), combine 3 types of information: the number of publications reporting the interaction, the reliability of detection evidence, and the reliability of detection for all orthologous protein pairs. The reliability of detection is a weighted sum across all detections of the interaction; each experiment detecting the interaction has a maximum weight of 1, which is reduced by half if the interaction type was not direct, and further reduced by half if the detection method was a HT screen (i.e., a minimum score of 0.25 is assigned to non-direct interactions, detected by HT screens). HAPPI-2 ([Chen et al., 2017](#)) calculates confidence scores as a product of detection evidence:

$$P_{II} = 1 - \prod_i^N (1 - S_i)$$

where N is the number of different platforms that detected the interaction and S_i is a score between 0 and 1, reflecting the reliability of the platform for detecting direct interactions. For PPIs appearing in a previous version of the database, scores are calculated as:

$$P = 1 - (1 - P_I) \times (1 - P_{II})$$

where P_I is the score from the previous database version, calculated in the same way as P_{II} .

PPI Context

The context of an interaction is the set of conditions in which it usually takes place, including location (e.g., tissue, cell type, sub-cellular localization), time (e.g., developmental stage), and phenotype (e.g., disease state). The set of interactions in a particular context can be very different compared to the full set of PPIs for the organism – for example, some human tissues express less than half of all human genes ([Emig et al., 2011](#)) and additional conditions such as cell type, may further decrease this fraction. Consequently, context-specific PPI networks can have greater efficacy for applications such as identification of disease genes ([Magger et al., 2012](#)).

PPI experiments typically provide limited context information because experimental conditions are often different from an interaction's natural setting (e.g., PPI experiments often use cell lines or yeast cells, very different from a human tissue). However, some aspects of context can be estimated from gene or protein expression, and from protein annotations of disease, function, process, and sub-cellular localization. Several databases including HIPPIE (Alanis-Lobato *et al.*, 2017), IID (Kotlyar *et al.*, 2016), MyProteinNet (Basha *et al.*, 2015), and TissueNet (Barshir *et al.*, 2013), provide tissue-specific interactions by integrating PPIs from primary databases, and identifying a subset of PPIs whose proteins or encoding genes are expressed in the tissue. HIPPIE (Alanis-Lobato *et al.*, 2017) and MyProteinNet (Basha *et al.*, 2015) associate PPIs with sub-cellular localizations based on annotations of interacting protein pairs. HIPPIE (Alanis-Lobato *et al.*, 2017) also associates PPIs with diseases using a similar approach.

Database Functionality

In addition to preparing interaction data, PPI databases also provide extensive functionality for querying, and in some cases analyzing and visualizing interactions. The overall aim is enabling easy retrieval of relevant interactions. A typical use case could be a researcher seeking interactions for a set of mutated genes, and requiring some of the following functionality: querying the database using Ensembl gene IDs, retrieving all detected and predicted partners of the corresponding proteins, filtering retrieved interactions by reliability and tissue, downloading results in an Excel-compatible format, and visualizing the resulting network. This example highlights important types of functionality that users may need: queries by various gene and protein IDs, retrieval of comprehensive networks, filtering of interactions (e.g., by reliability or context), and export of results in convenient formats for further analysis and visualization. The sections below describe support for this functionality among PPI databases (Table 1).

Most PPI databases accept queries by UniProt IDs and many databases accept a wide variety of other identifiers. However, submitting a large set of query IDs other than UniProt can be difficult. Some databases do not support queries by multiple IDs, but a more common problem is that identifiers may not have 1:1 mappings to proteins. If an identifier maps to multiple proteins, users may be asked to select their intended mapping, which can be time consuming if there is a large number of query IDs.

Comprehensiveness of retrieved networks varies greatly between databases; for example, the number of human interactions in major PPI databases ranges from a few thousand to more than two million. Primary databases tend to have fewer interactions than secondary databases, and the largest networks are provided by databases that include detected and predicted PPIs. However, predicted PPIs such as those in STRING (Szklarczyk *et al.*, 2017) may be mostly functional interactions rather than direct or physical associations. Most PPI studies predicting direct or physical associations have not yet been integrated into any secondary databases.

Interactions may need filtering by reliability, context (e.g., tissue, disease state), interaction type, and other criteria. Some primary and many secondary databases enable filtering by confidence scores. Filtering by location is supported by several secondary databases including HIPPIE (Alanis-Lobato *et al.*, 2017), IID (Kotlyar *et al.*, 2016), MyProteinNet (Basha *et al.*, 2015), and TissueNet (Barshir *et al.*, 2013). Filtering by interaction type and other criteria is supported by BioGRID (Chatr-aryamontri *et al.*, 2017), HIPPIE (Alanis-Lobato *et al.*, 2017), and iRefWeb (Turinsky *et al.*, 2014).

Retrieved interactions can usually be downloaded as delimited text files, such as the MITAB files supplied by IMEx databases. IMEx databases and STRING also provide PSI-MI XML files. Several databases including BioGRID, HIPPIE, IntAct, and STRING also provide downloadable formats for network visualization.

Future Directions

In the near future, PPI databases may need to accommodate vast amounts of information related to isoform-specific, mutant-specific, and context-specific interactions. Most PPI studies have not considered how (or whether) isoforms, mutations, or context, affect interactions. However, recent studies (Yates and Sternberg, 2013; Petschnigg *et al.*, 2014; Yang *et al.*, 2016; Naro *et al.*, 2017) suggest that these factors have profound effects on interactomes. Non-synonymous single nucleotide polymorphisms have been shown to alter PPIs (Yates and Sternberg, 2013) and Petschnigg *et al.* showed that mutations of EGFR could substantially alter its interactions, including gains and losses of interaction partners (Petschnigg *et al.*, 2014). Recently, Yang *et al.* showed that pairs of protein isoforms shared less than 50% of their interactions (Yang *et al.*, 2016).

Given the ubiquity of polymorphisms, mutations, and alternative splicing in eukaryotic proteomes, these factors could profoundly impact our understanding of interactomes – not only greatly increasing the number of interactions but also highlighting the importance of context-specific interactions. These factors suggest that individuals, diseases, tissues, and other contexts are characterized by important interactome differences.

PPI databases may soon need to accommodate a huge increase in interactome complexity. This will include supporting queries by specific polymorphisms, mutations, and isoforms, and linking these variants to diseases, tissues and other contexts. Interactions currently stored in PPI databases will benefit from being described in terms of these properties (e.g., detected protein pairs mapped to specific isoforms, PPIs increased/decreased by specific mutations, etc.).

Closing Remarks

PPI databases have become major resources for molecular and systems biology, alongside genomic and proteomic databases. However, PPI data, perhaps more than other data types in systems biology, is scattered across a very large number of studies and is investigated by a great variety of experimental and computational methods. Consequently, PPI databases have undertaken large scale integration efforts including curating thousands of journal papers, creating a controlled vocabulary for describing PPI experiments, and defining common formats for PPI data. At the same time they have introduced quality control measures for curation, methods for scoring interactions, and approaches for associating interactions with context. Through these efforts, PPI networks have become a key resource for tasks such as prediction of gene function, identification of disease genes, and drug discovery.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Biological Database Searching. Graphlets and Motifs in Biological Networks. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases. Prediction of Protein Localization. Prediction of Protein–Protein Interactions: Looking Through the Kaleidoscope. Protein–DNA Interactions. Protein–Peptide Interactions in Regulatory Events. Protein–Protein Interaction Databases. Protein–Protein Interactions: An Overview. Transcription Factor Databases

References

- Aken, B.L., *et al.*, 2016. The Ensembl gene annotation system. Database: The Journal of Biological Databases and Curation 2016, baw093. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27337980> (accessed 09.08.16).
- Alanis-Lobato, G., Andrade-Navarro, M.A., Schaefer, M.H., 2017. HIPPIE v2.0: Enhancing meaningfulness and reliability of protein–protein interaction networks. Nucleic Acids Research 45 (D1), D408–D414. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27794551> (accessed 02.08.17).
- Alonso-López, D., *et al.*, 2016. APID interactomes: Providing proteome-based interactomes with controlled quality for multiple species and derived networks. Nucleic Acids Research 44 (W1), W529–W535.
- Aranda, B., *et al.*, 2011. PSICQUIC and PSIScore: Accessing and scoring molecular interactions. Nature Methods 8, 528–529.
- Barshir, R., *et al.*, 2013a. The TissueNet database of human tissue protein–protein interactions. Nucleic Acids Research 41 (Database Issue), D841–D844. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23193266>.
- Basha, O., *et al.*, 2015. MyProteinNet: Build up-to-date protein interaction networks for organisms, tissues and user-defined contexts. Nucleic Acids Research 43 (W1), W258–W263. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25990735> (accessed 02.08.17).
- Baspinar, A., *et al.*, 2014. PRISM: A web server and repository for prediction of protein–protein interactions and modeling their 3D complexes. Nucleic Acids Research 42 (W1), W285–W289. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24829450> (accessed 30.06.17).
- Breuer, K., *et al.*, 2013. InnateDB: Systems biology of innate immunity and beyond – Recent updates and continuing curation. Nucleic Acids Research 41 (Database Issue), D1228–D1233. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3531080&tool=pmcentrez&rendertype=abstract> (accessed 08.01.15).
- Ceol, A., *et al.*, 2008. Linking entries in protein interaction database to structured text: The FEBS Letters experiment. FEBS Letters 582 (8), 1171–1177. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18328820> (accessed 17.08.17).
- Ceol, A., *et al.*, 2010. MINT, the molecular interaction database: 2009 Update. Nucleic Acids Res 38 (Database Issue), D532–D539.
- Chatr-aryamontri, A., *et al.*, 2017. The BioGRID interaction database: 2017 Update. Nucleic Acids Research 45 (D1), D369–D379. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw1102> (accessed 28.03.17).
- Chautard, E., *et al.*, 2009. MatrixDB, a database focused on extracellular protein–protein and protein–carbohydrate interactions. Bioinformatics 25 (5), 690–691.
- Chen, J.Y., Pandey, R., Nguyen, T.M., 2017. HAPPI-2: A comprehensive and high-quality map of human annotated and predicted protein interactions. BMC Genomics 18 (1), 182. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/28212602> (accessed 30.06.17).
- Consortium, U., 2017. UniProt: The universal protein knowledgebase. Nucleic Acids Research 45 (D1), D158–D169. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw1099> (accessed 27.07.17).
- Côté, R.G., *et al.*, 2007. The Protein Identifier Cross-Referencing (PICR) service: Reconciling protein identifiers across multiple source databases. BMC Bioinformatics 8, 401. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17945017> (accessed 27.06.17).
- Cusick, M.E., *et al.*, 2009. Literature-curated protein interaction datasets. Nature Methods 6 (1), 39–46.
- De Las Rivas, J., Prieto, C., 2012. Protein interactions: Mapping interactome networks to support drug target discovery and selection. Methods in Molecular Biology (Clifton, NJ) 910, 279–296. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/22821600> (accessed 20.05.15).
- Elefsinioti, A., *et al.*, 2011. Large-scale de novo prediction of physical protein–protein association. Molecular and Cellular Proteomics 10 (11), Available at: <http://www.ncbi.nlm.nih.gov/pubmed/21836163>.
- Emig, D., Kacprowski, T., Albrecht, M., 2011. Measuring and analyzing tissue specificity of human genes and protein complexes. EURASIP Journal on Bioinformatics & Systems Biology 2011 (1), 5. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/21970702> (accessed 02.08.17).
- Fahey, M.E., *et al.*, 2011. GPS-Prot: A web-based visualization platform for integrating host–pathogen interaction data. BMC Bioinformatics 12 (1), 298.
- García-García, J., *et al.*, 2012. BIPS: Biana Interolog Prediction Server. A tool for protein–protein interaction inference. Nucleic Acids Research 40 (Web Server Issue), W147–W151. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/22689642> (accessed 08.07.17).
- Hart, G.T., Lee, I., Marcotte, E.R., 2007. A high-accuracy consensus map of yeast protein complexes reveals modular nature of gene essentiality. BMC Bioinformatics 8, 236.
- Hart, G.T., Ramani, A.K., Marcotte, E.M., 2006. How complete are current yeast and human protein–interaction networks? Genome Biology 7 (11), 120.
- Huang, H., *et al.*, 2011. A comprehensive protein-centric ID mapping service for molecular data integration. Bioinformatics (Oxford, England) 27 (8), 1190–1191. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/21478197> (accessed 27.07.17).
- Kalathur, R.K.R., *et al.*, 2014. UniHI 7: An enhanced database for retrieval and interactive analysis of human molecular interaction networks. Nucleic Acids Research 42 (Database Issue), D408–D414.
- Kerrien, S., *et al.*, 2007. Broadening the horizon – Level 2.5 of the HUPO-PSI format for molecular interactions. BMC Biology 5, 44.
- Keshava Prasad, T.S., *et al.*, 2009. Human protein reference database – 2009 Update. Nucleic Acids Research 37 (Database Issue), D767–D772.

- Kim, S., *et al.*, 2016. BioCreative V BioC track overview: Collaborative biocurator assistant task for BioGRID. Database 2016. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27589962> (accessed 27.07.17).
- Kotlyar, M., *et al.*, 2015. In silico prediction of physical protein interactions and characterization of interactome orphans. *Nature Methods* 12 (1), 79–84. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25402006> (accessed 12.08.15).
- Kotlyar, M., *et al.*, 2016. Integrated interactions database: Tissue-specific view of the human and model organism interactomes. *Nucleic Acids Research* 44 (D1), D536–D541. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4702811&tool=pmcentrez&rendertype=abstract> (accessed 02.04.16).
- Krallinger, M., *et al.*, 2008. Overview of the protein–protein interaction annotation extraction task of BioCreative II. *Genome Biology* 9 (Suppl. 2), S4. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18834495> (accessed 27.07.17).
- Leitner, F., *et al.*, 2010. The FEBS Letters/BioCreative II.5 experiment: Making biological information accessible. *Nature Biotechnology* 28 (9), 897–899. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20829821> (accessed 17.08.17).
- Magger, O., *et al.*, 2012. Enhancing the prioritization of disease-causing genes through tissue specific protein interaction networks. *PLOS Computational Biology* 8 (9), e1002690. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23028288> (accessed 02.08.17).
- Maglott, D., *et al.*, 2011. Entrez gene: Gene-centered information at NCBI. *Nucleic Acids Research* 39.
- Micale, G., *et al.*, 2015. SPECTRA: An integrated knowledge base for comparing tissue and tumor-specific PPI networks in human. *Frontiers in Bioengineering and Biotechnology* 3, 58.
- Montecchi-Palazzi, L., *et al.*, 2009. The PSI semantic validator: A framework to check MIAPE compliance of proteomics data. *Proteomics* 9, 5112–5119.
- Mostafavi, S., Morris, Q., 2012. Combining many interaction networks to predict gene function and analyze gene lists. *Proteomics* 12 (10), 1687–1696. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/22589215> (accessed 10.08.15).
- Naro, C., *et al.*, 2017. An orchestrated intron retention program in meiosis controls timely usage of transcripts during germ cell differentiation. *Developmental Cell* 41 (1), 82–93. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/28366282> (accessed 04.08.17).
- Orchard, S., 2012. Molecular interaction databases. *Proteomics* 12, 1656–1662.
- Orchard, S., *et al.*, 2007. The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nature Biotechnology* 25, 894–898.
- Orchard, S., *et al.*, 2012. Protein interaction data curation: The International Molecular Exchange (IMEx) consortium. *Nature Methods* 9 (4), 345–350.
- Orchard, S., Ammari, M., Aranda, B., *et al.*, 2014a. The MIntAct project – IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research* 42 (Database Issue), D358–D363. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkt1115> (accessed 28.03.17).
- Orchard, S., Hermjakob, H., 2015. Shared resources, shared costs – Leveraging biocuration resources. Database: The Journal of Biological Databases and Curation 2015. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25776020> (accessed 17.08.17).
- Pagel, P., *et al.*, 2005. The MIPS mammalian protein–protein interaction database. *Bioinformatics* 21 (6), 832–834.
- Petschnigg, J., *et al.*, 2014. The mammalian-membrane two-hybrid assay (MaMTH) for probing membrane-protein interactions in human cells. *Nature Methods* 11 (5), 585–592. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24658140> (accessed 07.03.15).
- Planas-Iglesias, J., *et al.*, 2013. iLoops: A protein–protein interaction prediction server based on structural features. *Bioinformatics* 29 (18), 2360–2362. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23842807> (accessed 12.07.17).
- Prieto, C., De Las Rivas, J., 2006. APID: Agile Protein Interaction Data Analyzer. *Nucleic Acids Research* 34 (Web Server Issue), W298–W302. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/16845013> (accessed 28.07.17).
- Pruitt, K.D., *et al.*, 2012. NCBI Reference Sequences (RefSeq): Current status, new features and genome annotation policy. *Nucleic Acids Research* 40.
- Salwinski, L., *et al.*, 2004. The database of interacting proteins: 2004 Update. *Nucleic Acids Research* 32 (Database Issue), D449–D451. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=308820&tool=pmcentrez&rendertype=abstract> (accessed 28.03.15).
- Schaefer, M.H., *et al.*, 2012. HIPPIE: Integrating protein interaction networks with experiment based quality scores. (Deane, C.M. (Ed.)) *PLOS ONE* 7 (2), e31826.
- Schmidt, T., Frishman, D., 2006. PROMPT: A protein mapping and comparison tool. *BMC Bioinformatics* 7 (1), 331. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/16817977> (accessed 27.07.17).
- Schwikowski, B., *et al.*, 2000. Filtering high-throughput protein–protein interaction data using a combination of genomic features. *Nature Biotechnology* 18 (12), 1257–1261.
- Scott, M.S., Barton, G.J., 2007. Probabilistic prediction and ranking of human protein–protein interactions. *BMC Bioinformatics* 8, 239.
- Singh, R., *et al.*, 2010. Struct2Net: A web service to predict protein–protein interactions using a structure-based approach. *Nucleic Acids Research* 38 (Web Server Issue), W508–W515. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20513650> (accessed 12.07.17).
- Stark, C., *et al.*, 2006. BioGRID: A general repository for interaction datasets. *Nucleic Acids Research* 34 (Database Issue), D535–D539.
- Szklarczyk, D., *et al.*, 2017. The STRING database in 2017: Quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research* 45 (D1), D362–D368. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw937> (accessed 30.06.17).
- Turinsky, A.L., *et al.*, 2014. Navigating the global protein–protein interaction landscape using iRefWeb. *Methods in Molecular Biology* (Clifton, NJ) 1091, 315–331. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24203342> (accessed 11.08.15).
- Turner, B., *et al.*, 2010. iRefWeb: Interactive analysis of consolidated protein interaction data and their supporting evidence. Database 2010 (0), baq023.
- Veres, D.V., *et al.*, 2015. ComPPI: A cellular compartment-specific database for protein–protein interaction network analysis. *Nucleic Acids Research* 43 (Database Issue), D485–D493. Available at: <http://nar.oxfordjournals.org/content/early/2014/10/27/nar.gku1007.full> (accessed 11.08.15).
- Wang, X., Gulbahce, N., Yu, H., 2011. Network-based methods for human disease gene prediction. *Brief Funct Genomics* 10 (5), 280–293.
- Wang, Q., *et al.*, 2016. Overview of the interactive task in BioCreative V. Database 2016. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27589961> (accessed 27.07.17).
- Wu, J., *et al.*, 2009. Integrated network analysis platform for protein–protein interactions. *Nature Methods* 6 (1), 75–77. Available at: <http://www.nature.com/doi/10.1038/nmeth.1282> (accessed 28.07.17).
- Xenarios, I., *et al.*, 2000. DIP: The database of interacting proteins. *Nucleic Acids Research* 28 (1), 289–291.
- Yang, X., *et al.*, 2016. Widespread expansion of protein interaction capabilities by alternative splicing. *Cell* 164 (4), 805–817.
- Yates, C.M., Sternberg, M.J.E., 2013. The effects of non-synonymous single nucleotide polymorphisms (nsSNPs) on protein–protein interactions. *Journal of Molecular Biology* 425 (21), 3949–3963. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23867278> (accessed 04.08.17).
- Zhang, Q.C., *et al.*, 2012. Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature* 490 (7421), 556–560.

Further Reading

- Baker, M., 2012. Proteomics: The interaction map. *Nature* 484 (7393), 271–275.
- Braun, P., *et al.*, 2009. An experimentally derived confidence score for binary protein–protein interactions. *Nature Methods* 6 (1), 91–97.
- Hien, M.Y., *et al.*, 2015. A human interactome in three quantitative dimensions organized by stoichiometries and abundances. *Cell* 163 (3), 712–723.
- Snider, J., *et al.*, 2015. Fundamentals of protein interaction network mapping. *Molecular Systems Biology* 11 (12), 848.

Relevant Website

<https://www.ebi.ac.uk/training/online/course/protein-interactions-and-their-importance/interaction-databases>
EMBL-EBI.

Biographical Sketch

Max Kotlyar, PhD Max Kotlyar is a research associate at University Health Network, specializing in computational biology. He completed his PhD at the University of Toronto, in the department of Medical Biophysics. His research focuses on computational approaches for identifying protein-protein interactions, disease biomarkers, and drug or alternative therapies from high-throughput biological datasets. His interests include data-mining, machine learning, and network-based methods for analyzing high-dimensional datasets, especially in systems biology.

Chiara Pastrello, PhD Chiara Pastrello is a research associate at University Health Network, Toronto. She is interested in cancer genetics and epigenetics, that she formalised in her medical genetics residency – focusing on the prediction of pathogenicity of unclassified variants in mismatch repair genes. She is further interested in cancer systems biology, specializing in network analysis, and the curation of expression data for multiple databases maintained at Prof. Jurisica lab. Her current research interests include the study of microRNAs in cancer signalling to distant metastasis, as well as the application of network analyses to study molecules and pathways playing a central role in cancer.

Andrea E.M. Rossos is a third-year undergraduate student in a Bachelor of Science program in Honours Biology at McMaster University. She was a Princess Margaret Summer Student who has worked in the Jurisica Lab for summer 2016 and summer 2017 as a recipient of the Ian Lawson Van Toch Internship Award in Cancer Informatics. Andrea has been involved with data curation and development for projects such as microRNA Data Integration Portal (mirDIP), Cancer Data Integration Portal (CDIP) and Arthritis Data Integration Portal (ADIP). She has also been involved with projects looking at the properties and diversity of biomolecule (miRNA, protein, etc.) databases. Her's current research interests are the prognostic and diagnostic roles of microRNAs and other biomolecules in cancer and disease.

Igor Jurisica, PhD I. Jurisica is Tier I Canada Research Chair in Integrative Cancer Informatics, Senior Scientist at Krembil Research Institute, Professor at University of Toronto and Visiting Scientist at IBM CAS. He is also an adjunct professor at the School of Computing, Pathology and Molecular Medicine at Queen's University, Computer Science at York University, affiliate scientist at the Institute of Neuroimmunology, Slovak Academy of Sciences, and an Honorary Professor at Shanghai Jiao Tong University. Since 2015, he has also served as Chief Scientist at the Creative Destruction Lab, Rotman School of Management. He has published extensively on data mining, visualization and cancer informatics, including multiple papers in *Science*, *Nature*, *Nature Medicine*, *Nature Methods*, *J Clinical Oncology*, and has over 11,167 citations over the last 5 years. He has been included in Thomson Reuters 2016, 2015, 2014 list of Highly Cited researchers, and The World's Most Influential Scientific Minds: 2015, 2014 Reports.

Alignment of Protein-Protein Interaction Networks

Swarup Roy, Sikkim University, Gangtok, India and North-Eastern Hill University, Shillong, India

Hazel N Manners, North-Eastern Hill University, Shillong, India

Ahed Elmsallati, McKendree University, Lebanon, IL, United States

Jugal K Kalita, University of Colorado, Boulder, CO, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Protein molecules are the workhorses of the cell, performing and controlling almost all activities in an organism. Although some proteins may work alone, proteins usually collaborate with others to achieve their intended tasks. When proteins work together, the influences and interactions among them can be shown in terms of a graph. The human cell produces potentially 100,000 different proteins, where a gene may produce more than one protein. The interactions among these proteins are responsible for many physiological activities in the body. Organisms vary in the number of their proteins and the number of interactions. Proteins and protein interactions of a large number of organisms are being determined and are recorded in databases. According to one study (Stumpf *et al.*, 2008), humans have 10 times more protein interactions than the fruit fly and 20 times more than the single-celled yeast. A simple but useful approach to viewing such a complex biological system is to represent it as a network of interplay among the different molecules.

Thus, complex systems like Protein-Protein Interactions (PPI) are usually studied computationally from a graph theoretic perspective. Studies suggest that PPI networks (PIN) are conserved through evolution (Sharan *et al.*, 2005). Highly connected proteins within a network are vital molecules and have been found to be more essential for survival than proteins with lower connectivity (Jeong *et al.*, 2001). As a result, the interactions between protein pairs, as well as the overall composition of the network, are important for the overall functioning of an organism. Understanding conserved substructures through comparative analysis of these networks can provide insights into a variety of biochemical processes. The ultimate goal of network alignment is to transfer knowledge of protein function from one species to another. Since sequence similarity metrics such as BLAST bit scores (Altschul *et al.*, 1990) are not conclusive evidence of similar function, the purpose of aligning two PPI networks is to supplement sequence similarity with topological information so as to identify orthologs as accurately as possible.

PIN alignment is a relatively young research area, and successes of PIN network alignment, so far, include uncovering large shared sub-networks between species as diverse as *S. cerevisiae* and *H. sapiens*, and reconstructing phylogenetic relationships among species based solely on the amount of overlap discovered between their PPI networks (Kuchaiev *et al.*, 2010; Kuchaiev and Pržulj, 2011). Comparing two or more biological networks is a particularly challenging problem, since many interesting questions we might ask of these networks are computationally intractable to answer (Atias and Sharan, 2012). Assuming we have k graphs and each graph has n nodes (in reality, it is likely that the graphs have unequal number of nodes), each match tuple a_i contains k nodes, one from each graph. Ideally, we should have n such tuples in the overall alignment A with the stipulation that an alignment metric value is optimized. This is a very complex optimization problem which is NP-complete, and thus, all approaches have to be heuristic. Most papers in the literature report promising results in creating alignments that show either large regions of biological, or of topological similarity between the PPI networks of various species, but few do both well.

We introduce the PIN alignment problem followed by a brief discussion of various aligners proposed to address the issues of pairwise alignment as well as multiple alignment of PPI networks. The quality of alignments needs to be assessed using validation measures. We discuss some of the assessment matrices used for evaluating alignment results.

Protein-Protein Interaction (PPI) Network

The physical interaction between a pair of protein molecules takes place because of biochemical activities within a cell. The synthesis of a protein may also be regulated by another protein. When proteins work together, the influences and interactions among them can be shown in terms of a graph (Braun and Gingras, 2012). A graph showing interactions among proteins in a single species is called a Protein-Protein Interaction network (or map). In a PPI network, the proteins are nodes and molecular interactions between them are edges.

Definition 1: (*Protein-Protein-Interaction Network*). A set of proteins $\mathcal{P}$ forms a PPI network $\mathcal{G} = \{\mathcal{P}, \mathcal{T}\}$ by virtue of the physical interactions among the protein molecules in $\mathcal{P}$ due to various biochemical events. $\mathcal{T}$ represents the set of interactions between a pair of proteins P_i and $P_j \in \mathcal{P}$.

The interaction between a pair of protein molecules is measured using a variety of assays, such as immunoprecipitation, tandem affinity purification, and the yeast two-hybrid approach. Recently, these techniques have been scaled up to measure

Table 1 A summary of few PPI databases

Sl. No.	Database	No. of species	No. of interactors	No. of interactions	Link
1.	MINT	611	25,530	125,464	http://mint.bio.uniroma2.it/
2.	HPRD	1	30,047	41,327	http://www.hprd.org/
3.	BioGRID	62	65,617	1423,105	http://thebiogrid.org/
4.	MatrixDB	1	14,533	26,954	http://matrixdb.univ-lyon1.fr/
5.	IntACT	275	98,289	720,711	http://www.ebi.ac.uk/intact/
6.	InnateDB	4	25,110	367,496	http://www.innatedb.com/
7.	DIP	372	95,742	720,711	http://dip.doe-mbi.ucla.edu/dip/Main.cgi
8.	STRING	2031	964,376	1380,838,440	http://string-db.org/
9.	Mentha	8	88,130	648,080	http://mentha.uniroma2.it/index.php
10.	HAPPI	1	23,060	2922,202	http://discovery.informatics.uab.edu/HAPPI/
11.	IsoBase	5	87,737	114,897	http://cb.csail.mit.edu/cb/mna/isobase/
12.	APID	8	93,912	754,895	http://cicblade.dep.usal.es:8080/APIID/init.action

interactions on a genome-wide level (Uetz *et al.*, 2000). High-throughput techniques have also been developed to systematically identify protein complexes using affinity purification techniques, followed by mass spectrometry (MS/MS) to sequence the protein (Ho *et al.*, 2002; Gavin *et al.*, 2002).

PPI networks have been built for many different species based on interactions discovered and published by researchers. For example, the BioGRID database currently includes interactions, chemical associations, and other details from 58,254 publications. The total number of non-redundant interactions in the database is 1,110,310, covering more than 60 species. A number of available PPI database sources are shown in Table 1.

Assessment Methods

As the optimal solution is unknown for alignment of real PPI networks, there should be a method to evaluate various results extracted using different aligners. Two collections of evaluation metrics are used to evaluate the quality of alignments. The first set assesses the quality of the topological characteristics of the alignment. The second set measures the essence of the alignment considering biological fit. In this section, we review the evaluation metrics used for PPI network alignment, considering their strengths and deficiencies.

Topological Assessment

Edge correctness (EC)

Edge Correctness (EC) is the most popular metric used to measure the topological quality of network alignments. EC calculates the percentage of edges in the first network that are mapped to edges in the second network (Kuchaiev *et al.*, 2010). EC can be formally defined as

$$EC(G_1, G_2, f) = \frac{|f(E_1) \cap E_2|}{|E_1|} \quad (1)$$

where G_1 and G_2 are the two networks to be matched, and f is the alignment function (Kuchaiev *et al.*, 2010).

It is crucial to realize that an EC of 1 does not necessarily reflect a perfect alignment. In some cases, two different possible network alignments may have the same EC, but one of the alignments is clearly better than the other (Patro and Kingsford, 2012; Milenković *et al.*, 2013). For example, any sparse section in the first network can be easily mapped to a dense part of the second network, although this alignment is not the best match that can be extracted.

Induced conserved structure (ICS)

Recently, to overcome EC's weaknesses, the Induced Conserved Structure (ICS) score has been introduced (Patro and Kingsford, 2012). To circumvent aligning nodes from the first network to dense regions of the second network where opportunities for aligning in different ways are high the ICS score penalizes such an alignment whereas EC does not. ICS is formally defined as

$$ICS(G_1, G_2, f) = \frac{|f(E_1) \cap E_2|}{|E_{G_2[f(v_1)]}|} \quad (2)$$

where $|E_{G_2[f(v_1)]}|$ denotes the total number of edges in the induced sub-network of G_2 that includes all nodes that have been mapped to by f . Therefore, if the number of edges in the subgraph of G_2 induced by the nodes mapped from G_1 , is larger than the number of edges in G_1 , the ICS of the alignment will be lower.

Symmetric substructure score

While ICS addresses an essential weakness in EC by penalizing sparse-to-dense alignments, it does not similarly punish dense-to-sparse alignments. To improve this, the Symmetric Substructure Score, or S^3 was introduced (Saraph and Milenkovic, 2014), to penalize sparse-to-dense and dense-to-sparse alignments equally. In metaheuristic aligners, this metric has been shown to be a more reliable predictor of correct alignment than EC and ICS; it has been found that metaheuristic aligners can increase ICS by minimizing the denominator without increasing the numerator (Saraph and Milenkovic, 2014). S^3 addresses this problem. S^3 is defined as

$$S^3(G_1, G_2, f) = \frac{|f(E_1) \cap E_2|}{|E_1| + |E_{G_2[f(V_1)]}| - |f(E_1) \cap E_2|}. \quad (3)$$

The denominator of S^3 can be thought of as the number of unique edges in the composite graph resulting from overlapping the graphs G_1 and G_2 according to the alignment f .

Largest common connected sub-component

The size of the Largest Common Connected Sub-component (LCCS) considers large connected sub-graphs to be desirable (Kuchaiev et al., 2010). High EC, ICS, or S^3 scores do not necessarily reflect that the aligned nodes form a connected substructure.

Node correctness and interaction correctness

Two other evaluation metrics that have been used require that the true alignment of two PPI networks is known in order to estimate the quality of the extracted alignment. If the true alignment is g , then *Node Correctness* (NC) is the percentage of nodes aligned correctly. Formally, NC is defined as

$$NC(G_1, G_2, f) = \frac{|u_i : f(u_i) = g(u_i)|}{|V|} \times 100 \quad (4)$$

where $u_i \in V_1$, $f(u_i)$ is the mapping of an aligned node using the proposed alignment, and $g(u_i)$ is the correct match. Interaction Correctness (IC) is defined similarly as the percentage of interactions (i.e., edges) mapped correctly (Milenkovic et al., 2010).

As discussed above, NC and IC are measures that require advance knowledge of the true alignment of the networks, which is often unknown, especially for real biological networks. These two tools have been used on both synthetic networks, created for network alignment benchmarking (Sahraeian and Yoon, 2012b), and in noisy self-alignment experiments, where a network is mapped to a version itself that has been corrupted with extra edges (Kuchaiev et al., 2010).

Biological Assessment

Assessing alignments based on topological quality is not always sufficient, especially given the limitations of the topological measures surveyed above. Evaluating alignments based only on topological measures does not mean that the quality of the alignment is certainly good. Preferably, the alignment should be evaluated so as to confirm that the aligned proteins correspond to evolutionarily conserved functional modules. Biological measurements evaluate the alignment's quality with respect to the functional similarities between the two nodes (proteins) that are mapped to each other.

Gene ontology

Alignment papers typically use Gene Ontology (GO) annotations (Ashburner et al., 2000) to judge the functional similarity of aligned nodes. Gene ontology annotations for each protein are collected, and compared to see if they represent similar functions.

It is important to notice that most GO-derived metrics ignore the hierarchical structure of the Gene Ontology (Patro and Kingsford, 2012). Furthermore, many GO terms are assigned based on sequence homology (Pal, 2006). These two factors may generate misleading results if not accounted for (Patro and Kingsford, 2012). It is necessary to keep these two concerns in mind when considering GO-based evaluation metrics.

The naive way for evaluating GO term overlap, which is used by several studies, is to directly report the percentage of aligned pairs of proteins in the alignment that share more than k GO annotations (Kuchaiev and Pržulj, 2011; Memišević and Pržulj, 2012; Neyshabur et al., 2013). This evaluation is very straightforward and easy to interpret, but more sophisticated evaluation approaches have been proposed, as well.

Resnik's semantic similarity

The hierarchical structure of the GO ontology has been considered in computing similarity (Patro and Kingsford, 2012). Resnik's semantic similarity measure (Resnik, 1995; Pesquita et al., 2009), one of the most common ontological similarity metrics that considers hierarchical structure, has been used to design a biological metric for evaluating biological network alignments (Patro and Kingsford, 2012). If the Resnik ontological similarity of two proteins, u and v , is given by $Sim_{Resnik}(u, v)$, the Resnik similarity of aligned networks G_1 and G_2 , using the scoring function f is defined formally as

$$S_{Resnik}(G_1, G_2, f) = \frac{1}{|V_1|} \sum_{u \in V_1} Sim_{Resnik}(u, f(u)). \quad (5)$$

Gene ontology consistency

Another biological metric is Gene Ontology Consistency (GOC) (Aladağ and Erten, 2013), which uses GO annotations to assess of the alignment quality. GOC is a Jaccard index on sets of GO annotations, defined formally as

$$GOC(G_1, G_2, f) = \sum_{(u_i, v_j) \in a} \frac{|GO(u_i) \cap GO(v_j)|}{|GO(u_i) \cup GO(v_j)|} \quad (6)$$

where u_i, v_j are two nodes matched in the alignment of G_1 and G_2 , where a and $u_i \in V_1, v_j \in V_2$. The set of GO annotations of a protein u is expressed by $GO(u)$. This is typically limited to GO terms at depth 5 of the ontology, to avoid issues emerging from annotations of differing specificity.

Normalized gene ontology consistency

NGOC (Elmsallati et al., 2016) improves the evaluation of PPI network alignment for biological fit by taking into account the total number of proteins aligned. As some aligners fail to map all nodes, it is not appropriate to measure the quality of the biological similarity of the resulting alignments using the evaluation metrics mentioned above, especially GOC. As all biological evaluation metrics sum the score of the aligned nodes, an alignment may be better by simply mapping more proteins than in other alignments.

For example, say that aligner A aligns 4000 pairs of nodes and has a GOC score of 234.00, and aligner B aligns 2500 pairs of nodes and has a GOC score of 197.00. In this case, it is not fair to make a decision, based only on the GOC score, stating that aligner A is better than aligner B . Using NGOC, the biological similarity score is divided by the total number of nodes to make a more balanced comparison. According to our example, aligner B is better than aligner A in this case as aligner B scores 0.0788 while aligner A scores only 0.0585.

The NGOC is formally defined as:

$$NGOC(G_1, G_2, f) = \frac{1}{n} \left(\sum_{(u_i, v_j) \in a} \frac{|GO(u_i) \cap GO(v_j)|}{|GO(u_i) \cup GO(v_j)|} \right) \quad (7)$$

where a is the alignment to be evaluated and n is the total number of proteins aligned.

Aligning PPI Networks

PPI network alignment is a relatively new research area. The goal is to uncover large shared subnetworks between species, and derive phylogenetic relationships among species by examining the extent of overlap exhibited by different PPI networks (Vijayan et al., 2015; Milenković and Pržulj, 2012). The abundance of large PPI networks in various publicly available databases presents new opportunities and challenges. Discovering phylogenetic relationships across species by taking into account protein interactions during evolution is a fundamental task in comparative systems biology (Tatusov et al., 1997).

Comparative analysis of PPI networks across species helps identify evolutionary conserved subnetworks. It is often considered a graph alignment problem where proteins from different species are superimposed on one another based on evidence accrued from high protein sequence similarity as well as high topological similarity.

Definition 2: (PPI Network Alignment). Given k distinct PPI networks $\mathcal{G}_1 = \langle \mathcal{P}_1, \mathcal{J}_1 \rangle, \mathcal{G}_2 = \langle \mathcal{P}_2, \mathcal{J}_2 \rangle, \dots, \mathcal{G}_k = \langle \mathcal{P}_k, \mathcal{J}_k \rangle$ from k different species, the PPI alignment problem is to find a conserved subnetwork, A , within each of the k graphs. The alignment graph $\mathcal{A}$ is a subgraph consisting of nodes representing k similar proteins (one per species), and edges representing conserved interactions among the species. Each alignment $\mathcal{A}_i$ can be represented as:

$$\mathcal{A}_i = \{ \langle p_{1i}, p_{2i}, \dots, p_{ki} \rangle | p_{ji} \in \mathcal{P}_j \}. \quad (8)$$

The alignment problem looks for optimal set $\mathcal{A}(\mathcal{G}_1, \dots, \mathcal{G}_k) = \{ \mathcal{A}_i \}_{i=1 \dots k}$ alignments by maximizing the coverage of vertices from all the participating networks.

Comparing biological networks is a challenging problem, since conserved subnetworks are expected to possess both topological and biological similarities. Alignment of a pair of candidate proteins (or a set of k proteins) needs to satisfy certain similarity criteria based on functional and topological mapping as described below.

Functional Mapping

PPI alignment methods often align networks with the help of extra biological information such as the amino acid sequences of proteins, expression levels of corresponding genes, and Gene Ontology terms describing gene relationships and functional annotations. This biological information is used by the aligner to discover more biologically meaningful conserved subnetworks. However, studies have shown that high sequence similarity, determined by programs such as BLAST (Altschul et al., 1990), do not always result in similar protein functions (Elmsallati et al., 2017).

Topological Mapping

Some aligners use only network topology or structure of the network to align the PPI networks. Similarity between the nodes or proteins is calculated using a topological similarity score based on the structure of the PPI networks. Network structure related characteristics such as graphlet degree signatures, and the presence of and the number of cliques, are used as measures of topological similarity. Topological information is preferred over sequence information (Davis *et al.*, 2015; Malod-Dognin and Pržulj, 2015). Most aligners of the GRAAL family, GRAAL, H-GRAAL, C-GRAAL and L-GRAAL are purely topological and identify statistically significant subnetworks or complexes based on only a few topological evaluation measures. These aligners have wider application as they can be applied to any type of networks, and are not limited to only biological networks. Some other topologically based aligners are NetCoffee, GHOST and MAGNA.

Existing aligners either produce promising results in identifying biologically or functionally significant alignments, or are effective in producing topological fit, but few achieve both well.

Pairwise Alignments

A pairwise PPI network alignment is an alignment limited to only two species, i.e., $k=2$. Given two networks, it finds an optimal matching of nodes between the two. The main objective of aligning two PPI networks from two species is to find conserved proteins in the two species (Park *et al.*, 2011; Tatusov *et al.*, 1997). Most published work on network alignment focuses on pairwise alignment. Given two or more PPI networks, one can perform two types of alignments: global alignment or local alignment.

In global alignment, each node of one network is aligned to one and only one node in the other, producing a single overall match of two or more networks. In global network alignment, each node in the smaller network is uniquely aligned to only one best matching node in the larger network. Such an alignment detects a maximal subnetwork, which is useful in functional orthology prediction.

In contrast, local alignment attempts to find the best alignment of subnetworks of one network with subnetworks of another, without worrying about how other nodes line up. Local alignment methods may produce one-to-many node alignments, where a node from one network may be aligned to more than one node from the other network. In local alignment, small regions of similarity are matched independently, and several such regions may overlap in conflicting ways. Computationally, local alignment is less expensive than global as global alignment aims to find a single large consistent match.

A simple example showing the difference between global vs local network alignments is illustrated in Fig. 1.

Global Aligners

In this section, we review state-of-the-art global network aligners. We categorize them into four types: IsoRank aligners, GRAAL family aligners, Index-based aligners, and other aligners that have no particular features in common. Most of the aligners surveyed work in two main stages. First, a similarity matrix is calculated to indicate the similarity of a node in one network to a node in the other network. The similarity matrix is computed based on various standards depending on the aligner. In the next step, the similarity matrix is provided as an input to a matching algorithm to extract an alignment. These matching algorithms are usually approximate to reduce their running time, with the *seed-and-extend* approach being particularly common.

IsoRank Aligners

The IsoRank algorithms (Singh *et al.*, 2008; Liao *et al.*, 2009), are the earliest developed global aligners. The basic IsoRank algorithm has two major versions. While the first version deals with global pairwise network alignment, the second version of IsoRank solves the problem of global multiple network alignment. The main idea of the IsoRank algorithm is to use an algorithm similar to Google's *PageRank* (Page *et al.*, 1999) to estimate node similarity, by recursively defining node similarity in terms of the similarity of a nodes' neighbors.

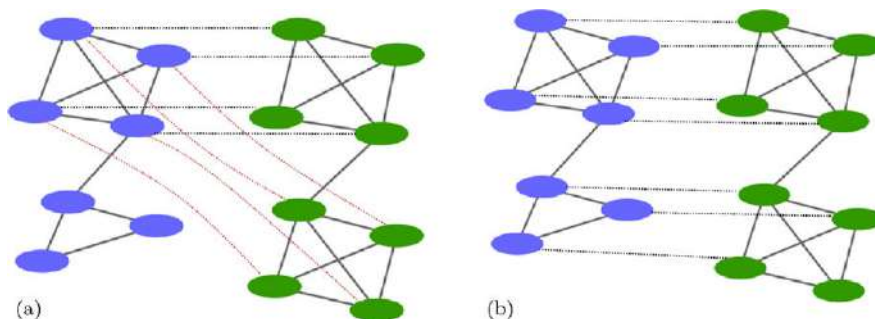


Fig. 1 Comparison of local and global network alignment. Dotted lines connecting nodes from different networks (of different colors) depicts an alignment of the node pair. (a) Local network alignment (one-to-many alignment); (b) Global network alignment (one-to-one alignment).

IsoRank

IsoRank (Singh *et al.*, 2008, 2007) tries to maximize the total number of nodes matched among the aligned networks. To accomplish this goal, a node from the first network is assumed similar to a node from the second network only if they both have similar neighborhoods in their respective networks. This intuition can be formalized as an eigenvalue problem for the aligned networks. It starts by calculating the scores R_{ij} , known as the quality of match, between nodes i and j ,

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv} \quad (9)$$

where $N(x)$ represents the set of x 's neighbors. The R_{ij} scores are used to write an eigenvalue equation $R=AR$, where R is the vector of all R_{ij} s, A is a doubly-indexed $|V_1||V_2| \times |V_1||V_2|$ matrix defined as

$$A[i, j][u, v] = \frac{1}{|N(u)||N(v)|} \quad (10)$$

for $(i, j) \in E_1$ and $(u, v) \in E_2$ and 0 otherwise. It is remarkable that A and R are vast in terms of size, but both are sparse matrices and can, therefore, be efficiently computed by matrix calculation techniques such as the power method (Golub and Van Loan, 2012). This strategy is very similar to Leordeanu and Hebert's (Leordeanu and Hebert, 2005) spectral technique for correspondence problems, which was originally used for graph alignment in computer vision contexts.

To estimate biological similarity, IsoRank uses BLAST bit scores (Altschul *et al.*, 1990) to measure the similarity between any two proteins. Biological similarity can be combined with topological similarity and weighted using a parameter α . The eigenvalue equation can be adjusted to obtain this balance and is formulated as

$$R = \alpha AR + (1 - \alpha)E, \quad \alpha \in [0, 1] \quad (11)$$

where E is the normalized vector of sequence similarity scores. To assign more importance to topological similarity, α is set to a value close to 1.0. On the other hand, if α has a value close to 0.0, it indicates that topological similarity is not considered significant and sequence similarity is more important. In the original experiments in the paper in which IsoRank is introduced, the best value of α is determined to be 0.6.

Extracting the final alignment follows finding the node similarity matrix for IsoRank. The alignment can be obtained by solving the problem of maximum weight bipartite graph matching (Singh *et al.*, 2008), and many methods can solve it. IsoRank uses a greedy matching technique that first aligns the nodes with the highest similarity, and then matches the remaining nodes with the next-highest similarity, and so forth. This is a *half* approximation of the maximum weight matching problem (Preis, 1999).

IsoRank works in two stages. The first step is to compute the similarity matrix, and then the next step obtains the final alignment. *Gap* alignment is defined by IsoRank as a node that does not have a valid match in the network. Due to its significance as the earliest global aligner developed, nearly all subsequent aligners compare their results with those generated by IsoRank.

Improvements of IsoRank

Significant enhancements to IsoRank have been presented (Kollias *et al.*, 2012, 2013). *Network Similarity Decomposition* (NSD) (Kollias *et al.*, 2012) is one approach that makes a notable improvement by speeding up the calculation of the node similarity matrix R_{ij} . The principal concept in NSD is to uncouple the two input graphs, processing them independently, and to decompose the computation of R_{ij} itself. In addition to having better time complexity, this approach is also suitable to large-scale parallelization. According to the authors, extracting the alignment this way speeds up the process three fold in some cases.

Another augmentation that updates the second phase of IsoRank was introduced in Kollias *et al.* (2013). The same IsoRank methodology is implemented in two stages with different matching estimation algorithms. This implementation of IsoRank is roughly 30 times faster than the original algorithm. The same approach of calculating the similarity matrix and combining it with biological similarity, as in Eq. (11) is used in the opening phase. In the second step, two different bipartite matching heuristics are applied to improve the running time of the original algorithm. A matrix based matching algorithm called *half-approximation* (Preis, 1999), with linear time complexity, is pipelined with an auction-based algorithm (Bertsekas and Castanon, 1992) to find the best-weighted matching and obtain the final alignment faster.

IsoRankN

IsoRankN (Liao *et al.*, 2009) or *IsoRank-Nibble* is an enhancement to the original IsoRank, presented earlier. It solves the problem of global multiple network alignment and can be used as a global pairwise aligner when only two networks are given as input. IsoRankN also works in two stages as in the original IsoRank algorithm. In the initial step, a similarity matrix is calculated for each pair of input networks. This step can be accomplished by running the original IsoRank algorithm on each pair of networks without performing the second round of IsoRank, where the final alignment is produced. The final alignment is generated in the second phase of IsoRankN by using an iterative spectral clustering algorithm (Andersen *et al.*, 2006). The standard output format for IsoRankN and all global multiple alignment programs is two sets of nodes, one from each network, that are aligned to each other. This format makes it difficult to evaluate the alignments according to the evaluation metrics (Neyshabur *et al.*, 2013). We note that the original implementation of IsoRank can align at most 5 networks of various species, due to its exponential time complexity the time for aligning more than six networks with IsoRankN is extremely high and therefore impractical (Hu *et al.*, 2013). Moreover,

the iterative nature of IsoRankN and the need to update large matrices at each iteration makes IsoRankN, worthless in aligning large networks, particularly those that have more than 10,000 nodes (Shih and Parthasarathy, 2012).

GRAAL (Graph Alignment) Family

The GRAAL series is a collection of network aligners that use topological similarity, as estimated by graphlet degree signatures (Pržulj 2007), to align one network to another. The intuition behind this methodology is that the local topology around the proteins in their respective protein networks is exhibited in the different ways proteins interact. Hence, homology information is deemed to be encoded in the network topology in addition to sequence similarity (Kuchaiev et al., 2010). The GRAAL family consists of four different algorithms, namely GRAAL (Kuchaiev et al., 2010), H-GRAAL (Milenkovic et al., 2010), MI-GRAAL (Kuchaiev and Pržulj, 2011), C-GRAAL (Memisevic and Pržulj, 2012) and L-GRAAL (Malod-Dognin and Pržulj, 2015).

GRAAL

GRAAL (Kuchaiev et al., 2010) is acknowledged to be the first global network alignment algorithm that employs *only* topological similarity to align networks. Matching networks based on their topological similarity alone makes GRAAL suitable not only for biological systems but also for other networks, such as social networks.

The main concept that all GRAAL algorithms use is that of graphlet degree signatures. Graphlets are small induced subgraphs that are obtained from a larger network. The idea of graphlet degree is a generalization of a node degree, and it counts the number of graphlets a node reaches (Pržulj 2007). The graphlet degree signature is a vector describing the number of graphlets of each type a node touches. Graphlets consist of between 2 and 5 proteins and can be expressed as shown in Fig. 2 in terms of 30 different tiny graphs. A node can reach these graphlets in 73 different ways, based on the automorphism orbits of the graphlets in question. The main reason behind using graphlets up to size 5 is that network topology similarity of many real-world networks can be computed easily using these sizes. Furthermore, increasing the size of graphlets leads to an exponential rise in time complexity (Milenkovic et al., 2010).

For each node in the network, a vector that counts the frequencies of all the ways that a node reaches the 73 graphlets is built. When two nodes are aligned, the match depends on the similarity of their graphlet degree signatures. The distance between any two nodes is first estimated as a log-ratio, and its value is in $[0,1]$, where 0 denotes topologically identical nodes and 1 implies that

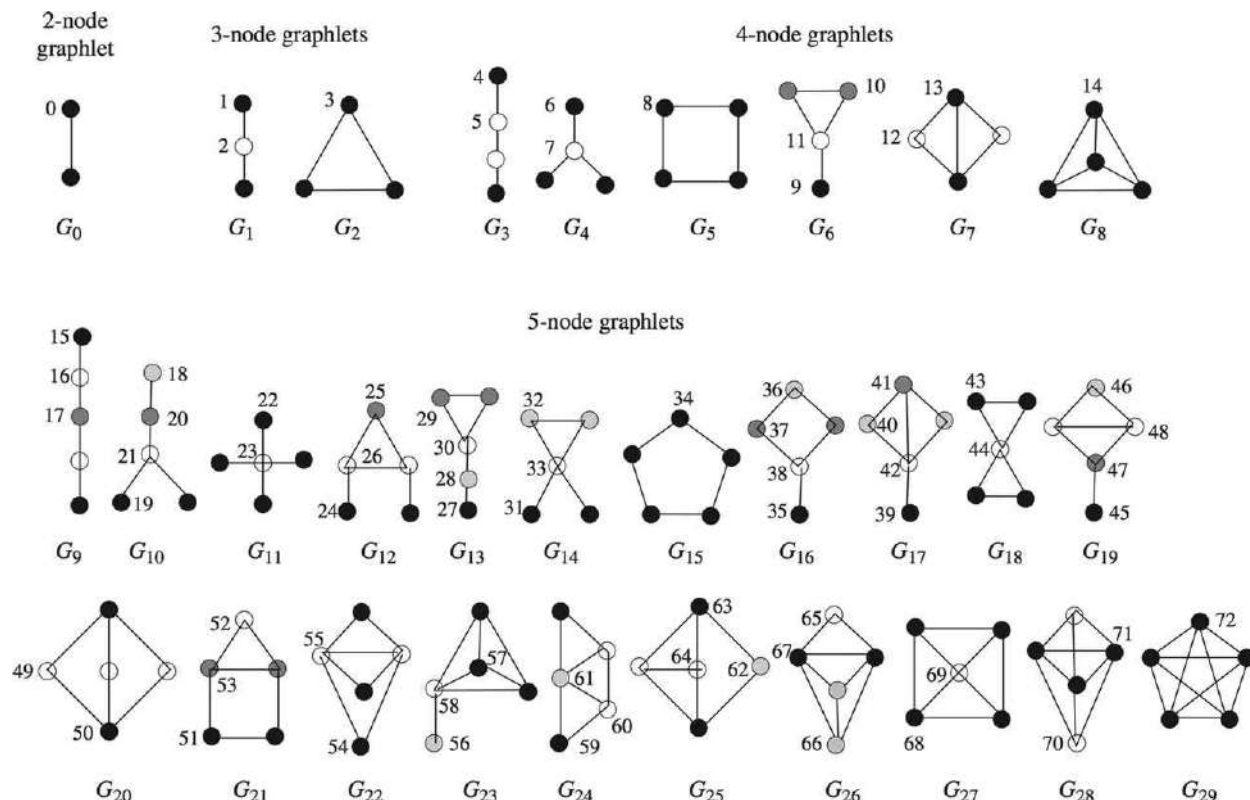


Fig. 2 Graphlets of up to size 5 nodes. Graphlets are induced sub-graphs that are extracted from a larger network. All possible graphlets that contain 2 to 5 nodes are labeled from G_0 to G_{29} . The figure shows all ways that a node touches a graphlet, shaded differently. These ways, known as *orbits* are labeled from 0 to 73. This figure is adapted from Pržulj, N., 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics* 23 (2), e177–e183.

the two nodes are completely topologically different. If the signature similarity between any two nodes u and v is expressed as $S(u,v)$, and α is a parameter that balances the usage of node signature similarity, the cost, $C(u,v)$, of matching node u with node v can be presented formally as

$$C(u,v) = 2 - \left(\frac{(1-\alpha)(\deg(u) + \deg(v))}{\max d(G_1) + \max d(G_2)} + \alpha S(u,v) \right) \quad (12)$$

where $\deg(x)$ is the degree of node x and $\max d(G)$ is the maximum degree of any node in graph G .

After determining the cost of aligning all pairs of nodes, GRAAL employs a *seed-and-extend* matching algorithm to obtain the final alignment. It begins by aligning nodes that have the least cost and expands the alignment to the neighbors of these nodes within a specified scope, usually up to 4 neighbors in depth. Dense regions of the networks are prioritized to be matched first since this enables larger common subnetworks to be decided.

H-GRAAL

H-GRAAL (Milenkovic et al., 2010) (*Hungarian-algorithm-based GRAPh ALigner*) is the second pairwise global network aligner based on graphlet degree similarity. It can also be used to align any other type of network. The H-GRAAL matching phase uses the *Hungarian algorithm* (Korsah et al., 2007) to find a better topological alignment than the one obtained using GRAAL, at the cost of higher runtime. H-GRAAL uses the same methodology used in GRAAL, where graphlet degree signature distances between all nodes are computed and used with the same cost function Eq. (12). The only difference between GRAAL and H-GRAAL is the use of the *Hungarian algorithm* to find the minimum weight bipartite match instead of the greedy *seed-and-extend algorithm*. There is clear tradeoff between getting better results and the time required to get such an alignment. The time complexity of the Hungarian algorithm is $O(n^3)$. Applying such algorithm in large biological networks is expensive, even though the results are better than GRAAL, when graphlet degree signatures are used for computing similarity.

MI-GRAAL

MI-GRAAL (Kuchaiev and Pržulj, 2011) (*Matching-based Integrative GRAPh ALigner*) integrates several similarity metrics that can be used to align two nodes. These metrics are not limited to topological similarity, but can also include any biological similarity metric, for instance by allowing a similarity matrix (typically of BLAST bit scores). MI-GRAAL integrates *five* different metrics, four metrics for topological similarity and one metric for biological similarity. The topological metrics used are: (1) *graphlet degree signature distance*, (2) *degree difference*, (3) *clustering coefficient difference*, which measures the extent of a node's neighbors' connectivity to each other, and (4) *eccentricity difference*, the maximum graph distance between a node u and any node in the graph. The fifth metric and the only biological metric used in the implementation is the BLAST bit score (Altschul et al., 1990), which is used to measure sequence similarity between proteins.

The cost of aligning any two nodes is calculated separately for each metric in independent matrices. The five matrices are then combined into one *confidence matrix*, C , that contains, as a fraction, the confidence of aligning node u of the first network and node v of the second network, based on the scores from the five matrices. Ties are broken randomly. Contradictions can arise when two nodes should be matched according to one metric, but not according to the others. The greedy *seed-and-extend* algorithm is used to find the final alignment after constructing a node-pairs priority queue in decreasing order simultaneously with the calculation of the confidence score matrix. If nodes u and v are aligned, their neighbors are aligned using maximum weight bipartite matching. MI-GRAAL continues in this manner until there are no more nodes in the first network to be matched.

It is important to note that using the *seed-and-extend* approach in the MI-GRAAL methodology as a second step, to find the final alignment, instead of the Hungarian algorithm, actually enhances the quality of MI-GRAAL's alignments, despite the fact that the Hungarian algorithm finds optimal matches according to the given similarity matrix. This appears to be the case because the similarity matrix given is only an estimate. By using a *seed-and-extend* algorithm that is, itself, biased towards edge conservation, better alignments can be constructed in less time than H-GRAAL.

C-GRAAL

C-GRAAL (Memisevic and Pržulj, 2012) (*Common-neighbors-based GRAPh ALigner*) is another algorithm in the GRAAL series. C-GRAAL uses graphlet degree signatures along with another topological metric to heuristically maximize the total number of edges aligned in the two networks. C-GRAAL also optionally can take bit scores as input. Topology-based alignment makes C-GRAAL applicable to any type of network. In addition to using graphlet degree signatures, C-GRAAL also uses the presence of *similar neighborhoods* to inform its *seed-and-extend* heuristic. The idea behind using this property is that if any two nodes share the same set of common neighbors, these two nodes must be aligned along with their neighbors. This can be done by exploiting a node's *neighborhood density*, the sum of its node degree and all of its neighbors' degrees, denoted as $nd(v)$. Neighborhood density of each node in both networks is calculated and then the combined neighborhood density, $cnd(u,v)$, is used to measure the similarity of any two nodes,

$$cnd(u,v) = \frac{nd(u) + nd(v)}{\max_{u_i \in G_1} (nd(u_i)) + \max_{v_j \in G_2} (nd(v_j))} \quad (13)$$

where G_1 and G_2 are the two networks to be aligned. The neighborhood density is used to determine the starting nodes for the *seed-and-extend* stage. C-GRAAL starts with (1) aligning the pair with the maximum cnd , and (2) extends the alignment to the

neighborhood of this pair by aligning the two neighbors with the highest number of common neighbors. Step 2 is repeated until there are no nodes in the neighborhood to be aligned, before going back to the first step. After aligning all nodes sharing a common neighborhood, C-GRAAL greedily aligns any remaining unaligned nodes to get the final alignment, using a node similarity measure combining graphlet degree signature with BLAST bit score. Note, that if no measure of node similarity is used, C-GRAAL breaks ties randomly.

L-GRAAL

L-GRAAL (Malod-Dognin and Pržulj, 2015) (*Lagrangian Graphlet-based ALigner*) is the latest algorithm published in the GRAAL series. L-GRAAL integrates the topological similarity used in all of the GRAAL family, the graphlet degree signature, with a *seed-and-extend* strategy based on Lagrangian relaxation, which was introduced in NATALIE (El-Kebir *et al.*, 2011). A unique feature of L-GRAAL is the use of interactions in the mapped edges based on their graphlet degree signature. Regarding its performance, the L-GRAAL implementation takes into account only graphlets of size 2 to 4. According to the authors, the quality of the resulting alignment does not improve by considering graphlets of up to size five. L-GRAAL balances biological and topological similarities using the α parameter, as in Eq. (11).

We should note that there are two important points that make L-GRAAL different from the rest of the GRAAL family, where graphlets are employed, and NATALIE, where integer programming and Lagrangian relaxation are used. Mapped interactions are not considered in the rest of the GRAAL series. L-GRAAL differs from NATALIE in that NATALIE uses a naïve scoring function to map proteins while L-GRAAL maps proteins, based on their BLAST similarity score as follows:

$$\text{sim}(u, v) = \frac{\text{SeqSim}(u, v)}{\max_{i,j}(\text{SeqSim}(i, j))} \quad (14)$$

where u and v are nodes to be aligned, and *SeqSim* is the BLAST bit score (Altschul *et al.*, 1990). L-GRAAL compares favorably with many aligners. Topologically, L-GRAAL does better than most state-of-the-art aligners such as MI-GRAAL, MAGNA, SPINAL and NATALIE in finding the largest common connected subgraphs. When Edge Correctness (EC) is used, L-GRAAL is better than all the aligners to which it has been compared.

Index-Based Aligners

Most aligners have poor performance in terms of time complexity. Hours to days are the average running times for the majority of global network aligners. To speed up the process of extracting alignments, some use parallel processing, such as the NSD approach (See Section "IsoRank"). The other possible way to improve performance is to use indices. In this section, we review two aligners that are index-based.

IBNAL

IBNAL (Elmsallati *et al.*, 2016) uses a unique methodology to extract a global alignment. The main idea in IBNAL is to divide the PPI network into two different sub-sets of proteins. For each node in the subset of subordinate nodes, a vector is built showing the number of cliques that this subordinate node connects. All subordinate nodes and cliques are indexed and stored in advance to expedite the process of extracting alignment later.

In the step of extracting the alignment, a similarity matrix that measures the distance between every two subordinate node vectors is calculated. The similarity is computed as the Euclidean distance between two subordinate nodes, where a distance of zero means that these two subordinate nodes touch the same number of cliques, and therefore are likely to be a match. A priority queue is built for all pairs of subordinate nodes, and ordered based on their similarity scores. Pairs of subordinate nodes are then popped from the queue and aligned along with the cliques that they touch. Cliques of the same size are matched to each other. This process continues until the queue is empty. According to the published results, IBNAL shows high topological scores with the S^3 (Symmetric Substructure Score) metric and comparable results to other state-of-the-art aligners with respect to GOC (Gene Ontology Consistency). The performance of IBNAL is excellent compared to other aligners in terms of the time needed to align. It takes IBNAL only a few seconds to extract the final alignment.

SSAlign

SSAlign (Elmsallati *et al.*, 2017) is another global network pairwise aligner that takes advantage of indexes and functional annotations. The main idea in SSAlign is to index all maximal substructures of up to a specified size. Larger substructures are excluded from consideration, as the cost of extracting and indexing such substructures is very expensive in time. The maximum size of the substructures that SSAlign uses in its implementation is 5. Extracting and indexing all substructures is an offline operation. SSAlign starts by building a priority queue for each type of symmetric substructure based on the Gene Ontology Consistency (GOC) score. Substructures with higher similarity scores are popped first from the queue, and are likely to be matched. After building all queues, SSAlign starts the first step of matching by aligning symmetric substructures with the highest GOC scores. During this step, all substructures that are partially matched are placed in another queue for the next step. The second phase of aligning takes care of all substructures that were queued in the first step. All nodes that are not aligned in the partially aligned substructures are matched to their corresponding nodes. SSAlign finalizes the alignment by aligning all neighbors of each node in

the alignment set. SSAlign is compared to a set of state-of-the-art aligners. According to the authors, SSAlign has been the only aligner that obtains topological and biological scores at the same time with respect to GOC and S^3 respectively.

Swap-Based Aligners

PISwap (Chindelevitch *et al.*, 2013), MAGNA (Saraph and Milenkovic, 2014), and Optnetalign (Clark and Kalita, 2015) are examples of aligners that use a swap methodology to optimize the alignment. While the PISwap optimizes the BLAST bit score, Optnetalign can optimize multiple objectives. MAGNA is unique in that it can start with an alignment produced by another aligner and further optimize it.

PISwap

PISwap (Chindelevitch *et al.*, 2013) is a heuristic aligner that uses a swap-based approach to iteratively improve an alignment. PISwap begins by creating a maximum weight bipartite match between the two networks using the Hungarian algorithm, based on optimizing BLAST bit scores. Once this mapping has been completed, PISwap attempts to improve the alignment topologically by performing a local search using the 3-opt heuristic, interchanging the assignment of 3 nodes at a time. Only 3-opt moves that do not decrease the bit score matching of the alignment, but improve the number of matched edges, are allowed.

PISwap has been evaluated considering its ability to create new alignments, as well as to refine alignments created by existing aligners. PISwap's authors (Chindelevitch *et al.*, 2013) find that PISwap compares favorably with IsoRank, GRAAL, and PATH (Zaslavskiy *et al.*, 2009), when evaluated on the IsoBase data set. They also claim it to be more robust to noise than earlier methods.

MAGNA

MAGNA (Saraph and Milenkovic, 2014) is a recently-published genetic algorithm for performing pairwise global network alignment. While the published version of the algorithm optimizes topological fit, as measured by S^3 , updated versions have added the ability to optimize biological similarity as well, using the standard method of linearly combining topological and biological fit into a combined objective as in Eq. (11).

Genetic algorithms work by maintaining a population of candidate solutions. These solutions are ranked by their quality, and the best solutions are recombined (crossed over) with each other to produce new candidate solutions, which are then slightly perturbed (mutated) and added to the new population. The genetic algorithm technique has been applied to a huge variety of optimization problems with excellent results. MAGNA either begins with a population of randomly-generated alignments, or with alignments created by other aligners. It then scores these alignments by either their EC (Edge Conservation), ICS (Induced Conserved Substructure), or S^3 score, depending on the user's choice, and then crosses over the best alignments to create a new population. This process repeats until no further improvements in the population are found, or until a user-specified limit is reached. The crossover operation MAGNA uses is motivated by abstract algebra, and essentially finds an alignment that is "between" the two given alignments, where the only operation allowed to transform an alignment is to swap the assignment of two nodes. Interestingly, unlike most genetic algorithms, no mutation operator is used.

MAGNA is able to both improve alignments created by other aligners, as well as create high quality new alignments, as compared to IsoRank, MI-GRAAL, and GHOST (Patro and Kingsford, 2012). For a noisy yeast self-alignment problem, MAGNA is able to correctly align a higher number of nodes than these other aligners, even in the presence of significant noise. It is also able to find non-trivial overlaps in GO annotations between aligned pairs. MAGNA's running time is comparable to several other recent aligners, such as GHOST. A parallel version of MAGNA, named as MAGNA++ (Vijayan *et al.* (2015)) has also been proposed by the same authors with improved alignment quality and graphical support for the user.

Optnetalign

Optnetalign (Clark and Kalita, 2015) approaches the problem of optimizing multiple objectives differently from existing aligners. As a multiobjective memetic algorithm, Optnetalign is a generalization of a genetic algorithm that uses the concept of *Pareto dominance* to compare alignments in its population and decide which ones to improve. We say that alignment x Pareto dominates alignment y if x is strictly better than y , with respect to one objective, and no worse than y with respect to all other objectives. For example, if x has an S^3 of 0.3 and a GOC of 300, and y has an S^3 of 0.29 and a GOC of 300, we say that x Pareto dominates y because it has a higher S^3 while having the same GOC. Importantly, this definition of Pareto dominance implies that a set of alignments may all be non-dominated with respect to one another. For example, if x has an S^3 of 0.5 and a GOC of 200, and y has an S^3 of 0.3 and a GOC of 300, then x does not Pareto dominate y and y does not Pareto dominate x . Since there exists a tradeoff between optimizing topological and biological fit, Optnetalign attempts to find a wide variety of non-dominated alignments in one run of the algorithm, and can produce hundreds of alignments at once.

Optnetalign starts by creating a set of random alignments, and then iteratively improves them, maintaining a population of non-dominated alignments. This task is highly parallelizable, and each thread works by selecting a random existing member of the population and then attempting to improve it using swap-based crossover, mutation, and local search operations. Once a thread has finished performing these operations, it checks whether the new alignment is non-dominated in the population. If so, it inserts

the new alignment into the population, and removes from the population any alignments that are Pareto dominated by the new alignment. This algorithm is highly scalable, and can scale to many cores for improved performance. After a time limit is reached, Optnetalign outputs all the non-dominated alignments it has found.

Optnetalign was extensively compared to most of the alignment algorithms discussed above, and was able to find a much wider variety of alignments at a wide range of tradeoffs between topological and biological fit. Additionally, on many problems in the IsoBase data set, Optnetalign was able to find better solutions than other aligners. Because of the large number of alignments created in a single run, the authors suggest that Optnetalign's output can be used to evaluate the relationships between different alignment objectives, and potentially to experiment with new alignment objectives.

Other Aligners

In this section, we survey a set of aligners that use disparate methodologies to match PPI networks. Pure topological mapping, spectral properties of graphs, and a mixture of biological and topological fits are the main concepts that these algorithms use.

GHOST

Spectral graph theory provides many properties that can be used to improve the computational performance of programs dealing with graphs. GHOST (Patro and Kingsford, 2012) takes advantage of such properties to build a similarity metric that is used as a cost function to align the input networks. The spectral signatures of subgraphs of different sizes around a specific node are calculated using the normalized Laplacian, and combined with a greedy *seed-and-extend* algorithm to find the alignment. The name of this algorithm is inspired by the multiscale spectral signature, which is the topological metric used.

First, for each node, the spectrum of the induced subgraphs of the node, at distances up to 4 hops is computed. The spectra are calculated as a set of eigenvalues of the Laplacian matrix. Because spectra are not commensurate with graph size, these spectra are transformed into spectral densities, which condense spectral information into commensurable quantities. The distance between any two nodes' signature is calculated as a structural distance defined in Banerjee (2012). Biological similarity is integrated with topological similarity in GHOST using the same equation that IsoRank uses (see Eq. (11)).

The alignment is computed in two steps. The first phase finds an initial alignment, which is enhanced in the second step. The initial alignment is matched by exploiting the *seed-and-extend* methodology, matching the most similar pair of nodes, and then extending the alignment to the neighborhood of the matched pairs by approximating a solution to a quadratic assignment problem (Garey and Johnson, 1979). After aligning all neighborhoods of the selected pair, the seed-and-extend strategy chooses the pair that has the next highest score as the next seed and the algorithm continues in this manner until all nodes are aligned. Thereafter, the second phase tries to improve the initial alignment using a local optimization strategy similar to *PLSwap* (Chindelevitch et al., 2010), but with a slight difference in rules and evaluation. The enhancement improves the topological quality of the final alignment. All possible realignments for single nodes are given a score, and the one with a better matching score is used, thereby improving the alignment.

NATALIE

NATALIE 2.0 (El-Kebir et al., 2011) is a pairwise global network aligner that combines topological and biological similarity to create alignments by solving an integer linear programming problem. NATALIE allows a *gap* alignment if a node does not have a good match in the second network, sharing this feature with IsoRank (see Section "IsoRank"). NATALIE balances sequence similarity and topological similarity using the approach used in Eq. (11). NATALIE 2.0 also allows the user to configure a minimum sequence similarity threshold, below which two nodes will never be matched.

NATALIE 2.0 formulates the alignment problem as an integer linear programming problem, generalized from a quadratic assignment problem (Garey and Johnson, 1979). The solution to the quadratic assignment problem, can be extracted by combining a generalization of the dual descent method (Adams and Johnson, 1994) for the quadratic assignment problem with the sub-gradient method (Held and Karp, 1971) for optimization. NATALIE 2.0 is an enhanced version of the work presented in Klau (2009), which used the method of Lagrangian relaxation for optimizing the common maximum substructure. According to the results presented in El-Kebir et al. (2011), NATALIE 2.0 has a very fast running time compared to other aligners, especially IsoRank and GRAAL, which it has been compared.

NETAL

NETAL (Neyshabur et al., 2013) (*NETwork Aligner*) is a pairwise two-phase network aligner. NETAL uses both topological and biological similarity to build the similarity matrix as the first phase. Then, a greedy search method is used to construct the final global alignment of the two networks.

NETAL starts by iteratively constructing a topological similarity matrix based on the neighborhood of each node. Another similarity matrix, for biological scores, is constructed based on two conditions: (1) The two proteins matched have to be biologically similar, and (2) The neighbors of two matched nodes have to be biologically similar. The two similarity matrices are then combined and balanced using an α parameter similar to the one used in Eq. (11) (see Section "IsoRank"). After the similarity score matrix is constructed, the interaction score matrix is built to indicate an approximation of how many interactions will be conserved if a given pair of proteins are aligned. Finally, using the similarity score matrix and the interaction

score matrix, the final alignment is constructed using a greedy search method. The search method starts by aligning nodes that have the maximum similarity score and updating the interaction score matrix. This process continues until the two PPI networks are aligned. The implementation of NETAL is available as an online tool, but we note that the current online implementation supports *only* topological similarity. Even though NETAL's computations are similar to those of IsoRank, there is a significant improvement in time complexity. The matrices of topological and biological similarity are not changed once constructed. However, the computation of these matrices can be performed offline making NETAL's performance better than other aligners that it has been compared to, such as GRAAL and MI-GRAAL. NETAL shows good results with noisy networks, and it outperforms MI-GRAAL in dealing with such networks (see Section "MI-GRAAL"). While the current version of NETAL does not support biological similarity inputs, NETAL is an extremely fast aligner and produces alignments with high levels of topological fit (Clark and Kalita, 2014).

SPINAL

SPINAL (Aladağ and Erten, 2013) (*Scalable Protein Interaction Network ALignment*) is another global pairwise aligner available publicly as a tool. SPINAL was developed to overcome the lack of scalability of previous approaches, such as IsoRank and MI-GRAAL. SPINAL, like most aligners, first computes a similarity matrix, and then creates an alignment based on it. In the first *coarse-grained* phase, the similarity matrix P is calculated iteratively based on a set of contributors C and sequence similarity. The set of contributors is calculated by finding the maximum weight matching for the bipartite graph of neighbors of the pair of nodes being matched. The confidence matrix P is defined formally as

$$P(u_i, v_j) = \frac{\sum_{(a_i, b_j) \in C} \frac{P(a_i, b_j)}{\deg(a_i) \times \deg(b_j)}}{\sqrt{|C|}} \quad (15)$$

where C is the set of contributors and $\deg(x)$ is the degree of a node x . We should note that a_i is a protein belonging to the first network and b_j is a protein in the second network. The similarity matrix P is then integrated and balanced with biological information using an α parameter in the same way used in IsoRank (see Eq. (11)). In the second *fine-grained* phase, a greedy *seed-and-extend* approach is combined with local improvement and swaps to compute the final alignment from the confidence matrix P .

SPINAL, according to the authors, shows good results when compared with two other aligners, MI-GRAAL and IsoRank. SPINAL runs five times faster than MI-GRAAL, which is a significant improvement. It also finds more conserved interactions than MI-GRAAL and IsoRank. Later work, comparing more aligners, showed that SPINAL is also competitive with or outperforms several other aligners as well.

PINALOG

PINALOG (Phan and Sternberg, 2012) (*Protein Interaction Network Alignment*) is a global pairwise aligner that uses both topological and biological information to align PPI networks. Unlike most aligners, PINALOG works in a three-stage process that aligns the networks in successively more fine-grained steps.

PINALOG starts by identifying a set of *communities* in a step known as *community detection*. Communities are dense sub-networks extracted from the original PPI network. The idea behind extracting such communities is that these sub-networks usually represent functional groupings of proteins within one PPI network. These communities are matched in the second phase, based on their sequence similarity scores. The second phase is known as *community mapping*, and it is done in two sub-steps. The first sub-step determines the similarity scores of all extracted communities in both networks and represents them as a matrix. Then, this matrix is used to find the best possible matching of communities by optimizing the scores. We should note here that the similarity between any two proteins is formally defined as

$$\text{SeqSim}(u_i, v_j) = \frac{\text{BLAST}(u_i, v_j)}{\sqrt{\text{BLAST}(u_i, u_i) \times \text{BLAST}(v_j, v_j)}} \quad (16)$$

where $\text{BLAST}(u_i, v_j)$ is the BLAST bit score (Altschul et al., 1990) obtained by aligning the sequence of the two proteins u_i and v_j .

In the third phase, PINALOG extracts the final alignment by aligning the communities that have the best scores, and adding aligned communities until no more pairs can be added to the final alignment. The third phase of PINALOG is known as *extension mapping*, which matches the individual proteins within matched communities.

PINALOG has been compared against MI-GRAAL and IsoRank. PINALOG surpasses IsoRank when the LCCS measure is used. Even though the LCCS of alignments produced by MI-GRAAL is larger when compared with those produced by PINALOG, PINALOG alignments have denser structures and its alignments are better in terms of homology than MI-GRAAL (Phan and Sternberg, 2012).

PROPER

PROPER (PROtein-protein interaction network alignment based on PERcolation algorithm) (Kazemi et al., 2016), method uses the *percolation graph matching* (PGM) concept, which assumes that some extra information, apart from the input networks, is available in the form of a set of seed nodes or proteins from which the iterative procedure can initiate and incrementally extend outward to other proteins to detect conserved protein complexes. The PROPER algorithm takes BLAST bit scores as input, along

with the PPI networks. It works in two steps, firstly the BLAST bit-score similarities are used to find a set of node pairs that will be used as seeds for the PGM algorithm in the second step. The selection of the seed pairs is an important step. In the second step, the percolation graph matching algorithm is executed. The pairs of nodes with BLAST similarity scores above a threshold are selected in descending order of their scores, and are matched with the seed pairs to extend incrementally. Pairs that already have been matched are discarded. Several measures, such as edge correctness, node correctness, number of conserved interactions, symmetric substructure score, induced conserved-structure score, largest connected shared-component, gene-ontology consistency and average normalized bit score have been used to show the significance of the algorithm.

Local Aligners

We present, below, some well-known local alignment methods.

PathBLAST

PathBLAST (Kelley *et al.*, 2004) is a local protein-protein aligner, and is one of the earliest PPI alignment methods. Just as BLAST is used to align protein sequences, PathBLAST aligns protein networks. Interacting pathways, which are chains of proteins are aligned with a pathway from the second network, in which the proteins occur in the same order. In this way PathBLAST restricts itself to linear chains of interacting proteins, simplifying the problem of network protein alignment. The alignments in PathBLAST are scored based on the similarity of the protein sequences at each pathway position, the quality of the protein interactions. PathBLAST allows gaps in the other network pathway alignments, i.e., where interacting proteins in one path can be aligned with a path in the other network where the proteins are not directly connected but instead are connected by an intermediate protein, i.e., at a distance of two. A web-based tool for PathBLAST is available (see Relevant Websites section) for seven species, where protein ID or protein sequences are provided by the user. PathBLAST reports high scoring alignments of the query network and the network on one of the available species.

NetworkBLAST

NetworkBLAST (Kalaev *et al.*, 2008) is a local protein-protein interaction network aligner. Unlike PathBLAST which detects paths, NetworkBLAST detects complexes that are conserved through evolution, between two species. It can also find complexes in a single network or species. A web-based tool (see Relevant Websites section) is available for NetworkBLAST, which takes two PPI networks as input, and the BLAST E-value of pairs of proteins. In case of a single species, only the PPI network is required as input. NetworkBLAST predicts complexes based on the density of the subnetworks, using a log likelihood ratio score, and nodes with high scores in the subnetworks are identified and used as seeds in a greedy expansion procedure.

NetAligner

NetAligner (Pache *et al.*, 2012) is another local network aligner, which finds conserved interactions of proteins based on evolutionary distances. From two input PPI networks, NetAligner creates a complex protein network based on the proteins that are present in the networks, and the interactions from the interactomes of the species. Seeds are identified from the connected components detected from the initial alignment and then are extended to other seed vertices using gaps or mismatch edges. In this extended alignment graph, the connected components are the final alignment. NetAligner is available online (see Relevant Websites section), where the user can select from three categories: Complex to Interactome, Pathway to Interactome, or Interactome to Interactome. In the complex to interactome mode, the query is a complex which is mapped to an interactome of the same or a different species. In the pathway to interactome mode, the query is a pathway from a species, which is mapped to the interactome of a different or the same species. This helps detect pathways that are conserved across species. In the interactome to interactome mode, whole networks or interactomes are mapped to each other to find conserved complexes between the species.

MaWISh

MaWISh is an early PPI aligner (Koyutürk *et al.*, 2006). It is a local PPI aligner that predicts conserved complexes, which are defined as locally maximal heavy subgraphs. In an alignment graph, conserved complexes that have similar functions are highly connected and dense, with fewer interactions with other complexes. MaWISh uses a greedy approach, which finds conserved hub nodes and extends them to nodes which have edges in both networks. MaWISh uses the min-cut algorithm to find a minimum number of edges in which to partition the graph, maximizing the weight of internal nodes to extract subgraphs. In MaWISh, balance is not considered, and the weight of only a part of the graph is considered. This method iteratively produces new heavy subgraphs until the subgraph becomes redundant (when $r\%$ of nodes are present in subgraph). In this way locally maximal heavy subgraphs, corresponding to or conserved complexes, are extracted from the PPI networks.

Multiple Alignments

Optimally one PPI network to another is NP complete, and that is why all algorithms for 2-network mapping are heuristic. If we have k PPI networks (say $k=50$ through a few thousand), and are tasked with finding an alignment of all the networks, an already

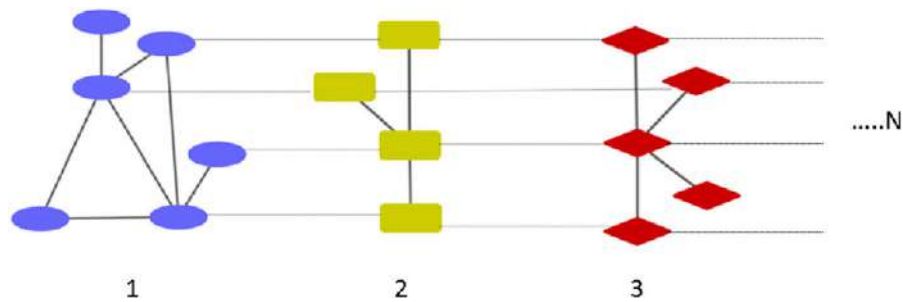


Fig. 3 An example of multiple network alignments for N different PPI networks. Nodes are shown in varying shapes to represent diverseness among the protein molecules. Edges among the inter-network nodes indicate possible alignments with respect to certain alignment parameters such as ontology or topological similarity.

difficult problem becomes more difficult. Aligning multiple PPI networks may enable better understanding of protein functions and protein interactions, leading to better understanding of the evolution of species. Even though multiple network aligners have been designed and implemented to deal with more than two input networks, they can still be used as pairwise aligners when the number of PPI networks is only two. Graphically, we illustrate possible multiple alignments of diverse PPI networks in Fig 3. Below, we provide a brief sketch of some of the currently available multiple aligners.

Græmlin

Græmlin (Flannick *et al.*, 2008) stands for *General and Robust Alignment of Multiple Large Interaction Networks*. It is one of several algorithms that align multiple networks, and is unique in requiring phylogenetic information as input, in addition to network data. There are two versions of this algorithm. Whereas the first algorithm, known as Græmlin 1.0 (Flannick *et al.*, 2006), performs local alignment, the second version of Græmlin is a global aligner, known as Græmlin 2.0 (Flannick *et al.*, 2008). Græmlin is one of the earliest aligners and works in two phases. In the first phase, a feature-based pairwise scoring function is calculated as a vector of pairwise node and edge features. The node features take into account four evolutionary events, namely, (1) *Protein deletion*, (2) *Protein duplication*, (3) *Protein mutation*, differences in the sequences of two proteins from different species, and (4) *Paralog mutation*, differences in the sequences of two proteins from the same species. The edge feature is built based on two events (1) *Edge deletion*, loss of an interaction between two proteins from different species, and *Paralog edge deletion*, loss of an interaction between two proteins in the same species. Then, using a training dataset, Græmlin learns a scoring function that optimizes the vector of features. In the second phase, Græmlin updates the initial alignment results from the first step and continues to iteratively update the alignment till the final alignment is found. At each iteration, a node is processed according to four possible actions: (1) leaving the node, (2) creating a new class containing the processed node, (3) moving the processed node with another existing class, or (4) merging the current class of a processed node to another existing class. Iterations continue until the final result is obtained (Flannick *et al.*, 2008).

Græmlin is similar to other aligners in the number of phases required to get the final result. However, unlike most aligners, the inputs to Græmlin differ from those of other aligners. Græmlin takes as input the networks to be aligned and the phylogenetic tree relating the species under consideration. On the other hand, most aligners, pairwise or multiple, take as input *only* the networks to be aligned. Hence, most aligners developed after Græmlin do not compare their results with Græmlin, even though Græmlin works in linear time (Neyshabur *et al.*, 2013). Note that Græmlin learns its scoring function. However, it requires a set of true alignments among some species for this purpose. Hence the results produced by Græmlin depend also on the set of true alignments used to train its scoring function (Shih and Parthasarathy, 2012; Hu *et al.*, 2013).

SMETANA

SMETANA (Sahraeian and Yoon, 2013) is a global multiple network aligner. SMETANA stands for *Semi-Markov random walk scores Enhanced by consistency Transformation for Accurate Network Alignment*. The goal of SMETANA is to effectively find the maximum expected alignment for large networks. As with most aligners, SMETANA works also uses two phases. In the first phase, the similarity score matrix, which shows the likelihood of aligning two nodes, based in the semi-Markov random walk model estimated using the *semi-Markov random walk model* (Sahraeian and Yoon, 2012a). A similarity matrix is calculated for all pairs of networks to be aligned. The similarity matrices are then integrated and balanced with biological information using the α parameter as given in Eq. (11). Next, two probabilistic consistency transformations are used to improve the estimated probabilities of aligning the pairs of nodes that were calculated in the first step. The first transformation is called *Intra-network Probabilistic Consistency*, where each node's neighbor information is used to update the original scores. The intuition behind using such a consistency transformation is similar to that used in IsoRank (see Section "IsoRank"), where the probability of aligning any two nodes increases as the neighbors of these two nodes are aligned. The second transformation is known as *Cross-network Probabilistic Consistency*, where the information about other networks is integrated to enhance the alignment of nodes across all aligned networks.

The second phase of SMETANA finds the final alignment using the scoring matrices constructed and enhanced in the first phase. The greedy seed-and-extend method is used to iteratively construct the final alignment. Starting at nodes that have the highest probability, SMETANA adds nodes to the final alignment with two constraints: (1) The nodes of a pair (u,v) to be added are not already in the current alignment, and (2) The maximum number of nodes in either network is not reached.

SMETANA is one of the only algorithms that has been extensively compared to many other aligners. It has a faster running time compared to IsoRankN and Græmlin. SMETANA also shows good performance as the number of aligned networks increases. Unlike Græmlin, SMETANA does not require training datasets to learn its scoring function. When used as a global pairwise aligner, SMETANA also has running time comparable to MI-GRAAL and C-GRAAL (Sahraeian and Yoon, 2013). On the other hand, SMETANA does not perform well when used to align more than six large-scale networks (Hu et al., 2013).

NetCoffee

NetCoffee (Hu et al., 2013) is another recently published global multiple aligner. It is an aligner that can be easily adjusted to align any kind of networks, not only PPI networks. NetCoffee is also one of the few network aligners available as an online tool. NetCoffee is unique among aligners in that it is a four-phase aligner. Another important feature that makes NetCoffee unique is that it combines biological and topological similarities to match two or more PPI networks.

NetCoffee is the extension of a prior method, called *T-Coffee* (Notredame et al., 2000), which is used for multiple sequence alignment. For each pair of networks to be aligned, a set of all bipartite graphs is built, known as a *Bipartite Graph Library*. Then NetCoffee assigns weights to all edges using an integrated approach similar to *T-Coffee*. Sequence similarity is calculated as a log-ratio and integrated with topological similarity using the same α parameter method as in Eq. (11). The third step of NetCoffee is to build the search space by selecting all candidate edges from all bipartite graphs. The aim of this step is to reduce the time needed to produce the final alignment. As a final step, NetCoffee finds the final alignment from the candidate edges collected in the third step. The final alignment is defined as an optimization problem, which is solved by simulated annealing (SA) (Kirkpatrick et al., 1983). NetCoffee also allows the user to specify a bit score threshold, above which two proteins are forced to be paired.

NetCoffee has been tested comprehensively on three datasets and its results have been compared with IsoRankN, Græmlin and SMETANA. According to the authors' results, NetCoffee has the ability to deal with large datasets and produce a final alignment in a reasonable running time compared to the aligners mentioned above. NetCoffee is at least 2 times faster than the aligners it was compared with. As one of the most recent aligners published, NetCoffee provides a solution to running time limitations that other multiple aligners and it can work with large input networks.

Table 2 Comparison of different PPI Alignment techniques

Alignment	Technique	Year	Pairwise alignment	Multiple alignment	Topological approach	Functional approach
Local Alignment	Græmlin	2006	✓	×	✓	✓
	Græmlin 2.0	2009	×	✓	✓	✓
	PathBLAST	2004	✓	×	✓	✓
	NetworkBLAST	2008	✓	×	✓	✓
	NetAligner	2012	✓	×	✓	×
	MaWISh	2006	✓	×	✓	✓
Global Alignment	IsoRank	2008	×	✓	✓	✓
	IsoRankN	2009	×	✓	✓	✓
	GRAAL	2009	✓	×	✓	×
	H-GRAAL	2010	✓	×	✓	×
	MI-GRAAL	2011	✓	×	✓	✓
	C-GRAAL	2012	✓	×	✓	×
	L-GRAAL	2015	✓	×	✓	×
	IBNAL	2016	✓	×	✓	✓
	SSAlign	2017	✓	×	✓	✓
	SMETANA	2013	×	✓	✓	✓
	NetCoffee	2014	×	✓	✓	✓
	BEAMS	2014	×	✓	✓	✓
	PISwap	2013	✓	×	✓	✓
	GHOST	2012	✓	×	✓	×
	MAGNA	2014	✓	×	✓	×
	Optnetalign	2015	✓	×	✓	✓
	NATALIE 2.0	2015	✓	×	✓	✓
	NETAL	2013	✓	×	✓	✓
	SPINAL	2013	✓	×	✓	✓
	PINALOG	2012	✓	×	✓	✓
	PROPER	2016	✓	×	✓	✓

BEAMS

BEAMS (Alkan and Erten, 2013) (*Backbone Extraction And Merge Strategy*) is a global multiple network aligner. BEAMS combines both topological and biological similarity to calculate a global multiple alignment. Biological similarity between proteins is calculated based on BLAST bit scores and balanced with topological similarity using the α parameter in the same way as in Eq. (11). BEAMS starts by constructing a k -partite similarity graph, where k is the number of input networks. The constructed k -partite weighted graph works as a superset from which the final alignment is extracted. The final alignment is a subset, a *filtered version*, of the complete k -partite graph of the input networks, and contains only edges that have maximum weights. The step of constructing the similarity graph can be considered an off-line step, as it needs to be done only once in advance, and it does not affect the overall running time.

Like most of the surveyed aligners, BEAMS takes a two-phase approach. In the first step, known as *Backbone Extraction*, all possible k -cliques of orthologous proteins, *backbones*, are extracted from the similarity graph and the input PPI networks, in a greedy manner, where the maximum score k -cliques are extracted first. These backbones are clustered and merged to form the final alignment, in a step known as *Merge Strategy*. We should note that these two steps are heuristic; the goal is to maximize the score of the clustered proteins. The merging step uses the well known *seed-and-extend* strategy, where the *backbones* that have the maximum scores are seeded first.

BEAMS has been compared only against two other multiple aligners, SMETANA and IsoRankN. Using only one dataset, IsoBase (Park et al., 2011), BEAMS shows better results in terms of biological metrics. In general, the authors' results show that BEAMS and SMETANA produce very similar results, and these methods are mutually more different from IsoRankN.

A comparative summary of different aligners are given in Table 2.

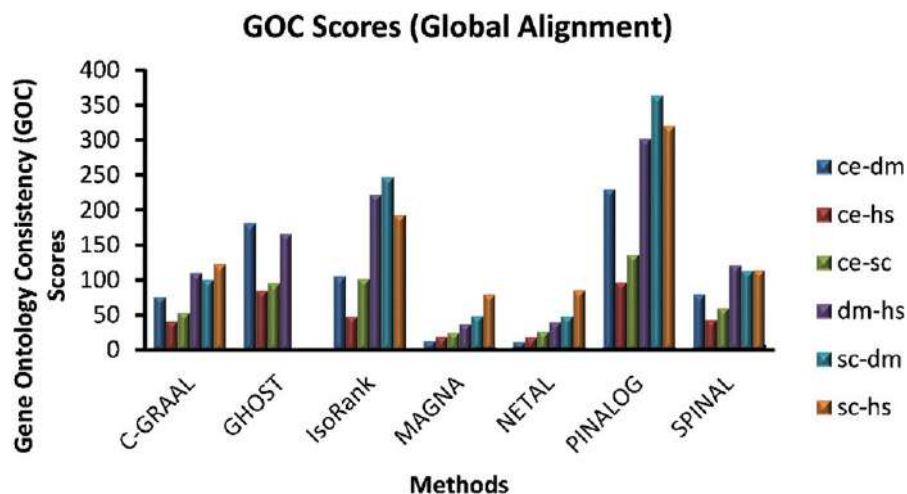


Fig. 4 Comparison of different global PPI Alignment techniques based on GOC score.

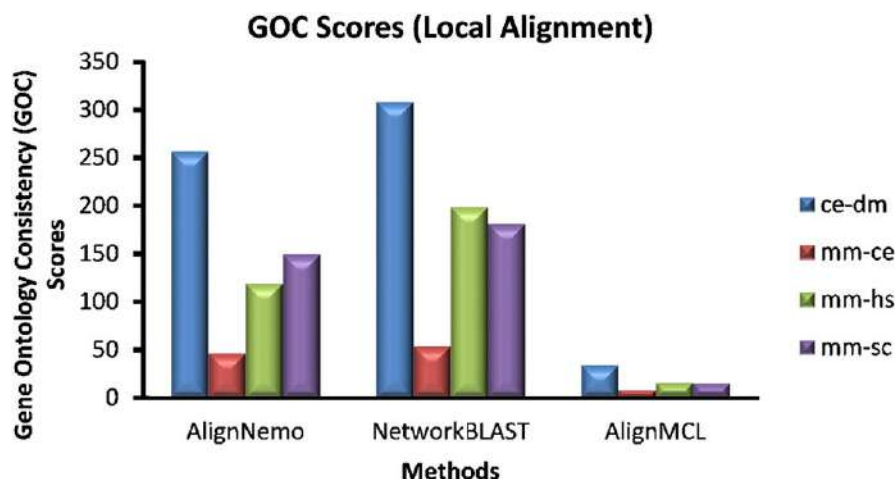


Fig. 5 Comparison of different local PPI Alignment techniques based on GOC score.

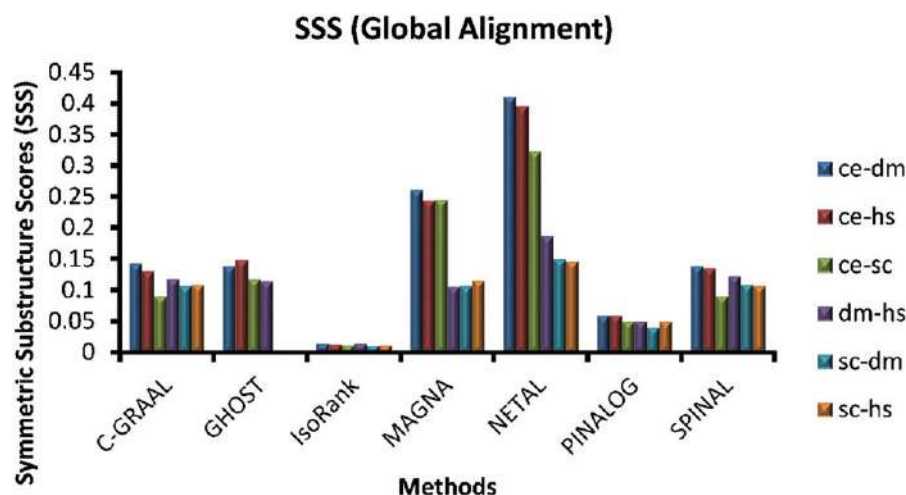


Fig. 6 Comparison of different global PPI Alignment techniques based on Symmetric Substructure Score (S^3).

Table 3 Comparison of few multiple PPI alignment techniques

Methods	No. of conserved interactions (CI)	Total no. of interactions between clusters	CI/Total no. of interactions between clusters (%)	Alignment score (AS)
BEAMS	7,060	114,889	6.15%	0.5261
IsoRankN	5,978	109,364	5.47%	0.3970
SMETANA	13,498	122,450	11.02%	0.4766

Performance Comparison

A comparison of some of the global and local aligners based on Isobase database of *Caenorhabditis elegans*, *Mus musculus*, *Homo sapiens*, *Drosophila melanogaster* and *Saccharomyces cerevisiae*, is shown below. GOC scores produced by different global and local aligners are shown in **Figs. 4** and **5**. In terms of GOC, local aligner, AlignNemo and NetworkBLAST perform well compared to AlignMCL. **Fig. 6** shows the comparison of global aligners with respect to the topological merit function S^3 . However, due to the presence of many-to-many links in local alignments we could not calculate any S^3 scores for local alignments. From the results, it is clear that PINALOG is a good global aligner in terms of functional significance (**Fig 4**), whereas NETAL and MAGNA produces relatively better topological matches (**Fig 6**).

We also report a comparison between multiple aligners in **Table 3** which is reproduced from (Sahraeian and Yoon, 2013). For a detailed description of the comparison attributes, one can refer to (Sahraeian and Yoon, 2013). As seen in the results, BEAMS algorithm performs better in terms of conserved interactions (CI) as compared to IsoRankN, but SMETANA outperforms them (**Table 3**).

Conclusion

Comparative analysis of PPINs across different species using network alignment helps in discovering sub-network motif and pathways conserved during evolution. Aligning two PPI networks supplements sequence similarity with topological information so as to identify orthologs as accurately as possible. They also provide insight into the prediction of protein functions, phylogenetic tree construction and drug design. Both pairwise and multiple alignment methods are available for PPINs. Both local and global alignment methods have been proposed. Local aligners produce multiple small conserved regions of matches with many-to-many node alignment. On the other hand, global alignments look for a single large region with a one-to-one match. Local alignments are less computationally expensive than global alignments. We discussed different global and local alignment methods. Network topology, sequence and functional conservation scores are the most commonly used metrics for network alignment. Methods generally either score well in topological matching, or in functional matching but not in both.

Alignment of multiple PPI networks involves searching over extremely large spaces, and therefore the algorithms require efficient heuristics, but still produce high quality results. The challenges that need to be overcome are only starting to be addressed.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Algorithms for Structure Comparison and Analysis: Docking. Natural Language Processing Approaches in Bioinformatics. Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases. Prediction of Protein Localization. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Quantification of Proteins from Proteomic Analysis

References

- Adams, W.P., Johnson, T.A., 1994. Improved linear programming-based lower bounds for the quadratic assignment problem. DIMACS Series in Discrete Mathematics and Theoretical Computer Science 16, 43–75.
- Aladağ, A.E., Erten, C., 2013. SPINAL: Scalable protein interaction network alignment. *Bioinformatics* 29 (7), 917–924.
- Alkan, F., Erten, C., 2013. BEAMS: Backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks. *Bioinformatics*. btt713.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3), 403–410.
- Andersen, R., Chung, F., Lang, K., 2006. Local graph partitioning using PageRank vectors. In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006. FOCS06 IEEE, pp. 475–486.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics* 25 (1), 25–29.
- Atias, N., Sharan, R., 2012. Comparative analysis of protein networks: Hard problems, practical solutions. *Communications of the ACM* 55 (5), 88–97.
- Banerjee, A., 2012. Structural distance and evolutionary relationship of networks. *Biosystems* 107 (3), 186–196.
- Bertsekas, D.P., Castanon, D.A., 1992. A forward/reverse auction algorithm for asymmetric assignment problems. *Computational Optimization and Applications* 1 (3), 277–297.
- Braun, P., Gingras, A.-C., 2012. History of protein-protein interactions: From egg-white to complex networks. *Proteomics* 12 (10), 1478–1498.
- Chindelevitch, L., Liao, C.-S., Berger, B., 2010. Local optimization for global alignment of protein interaction networks. In: *Pacific Symposium on Biocomputing*, vol. 15, pp. 123–132. Singapore: World Scientific.
- Chindelevitch, L., Ma, C.-Y., Liao, C.-S., Berger, B., 2013. Optimizing a global alignment of protein interaction networks. *Bioinformatics*. btt486.
- Clark, C., Kalita, J., 2014. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics* 30, 2351–2359.
- Clark, C., Kalita, J., 2015. A multiobjective memetic algorithm for PPI network alignment. *Bioinformatics* 31, 1988–1998.
- Davis, D., Yaveroglu, Ö.N., Malod-Dognin, N., Stojmirovic, A., Pržulj, N., 2015. Topology-function conservation in protein-protein interaction networks. *Bioinformatics*. btv026.
- El-Kebir, M., Heringa, J., Klau, G.W., 2011. Lagrangian relaxation applied to sparse global network alignment. *Pattern Recognition in Bioinformatics*. Springer. pp. 225–236.
- Elmsallati, A., Msalati, A., Kalita, J., 2016. Index-based network aligner of protein-protein interaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- Elmsallati, A., Roy, S., Kalita, J.K., 2017. Exploring symmetric substructures in protein interaction networks for pairwise alignment. In: *International Conference on Bioinformatics and Biomedical Engineering*, pp. 173–184. Berlin: Springer.
- Flannick, J., Novak, A., Do, C.B., Srinivasan, B.S., Batzoglou, S., 2008. Automatic parameter learning for multiple network alignment. *Research in Computational Molecular Biology*. Springer. pp. 214–231.
- Flannick, J., Novak, A., Srinivasan, B.S., McAdams, H.H., Batzoglou, S., 2006. Graemlin: General and robust alignment of multiple large interaction networks. *Genome Research* 16 (9), 1169–1181.
- Garey, M.R., Johnson, D.S., 1979. *Computers and Intractability*, 174. San Francisco, CA: Freeman.
- Gavin, A.-C., Bösch, M., Krause, R., *et al.*, 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415 (6868), 141–147.
- Golub, G.H., Van Loan, C.F., 2012. *Matrix Computations*, 3. Johns Hopkins University Press.
- Held, M., Karp, R.M., 1971. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming* 1 (1), 6–25.
- Ho, Y., Gruhler, A., Heilbut, A., *et al.*, 2002. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* 415 (6868), 180–183.
- Hu, J., Kehr, B., Reinert, K., 2013. NetCoffee: A fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics*. btt715.
- Jeong, H., Mason, S.P., Barabási, A.-L., Oltvai, Z.N., 2001. Lethality and centrality in protein networks. *Nature* 411 (6833), 41–42.
- Kalaei, M., Smoot, M., Ideker, T., Sharan, R., 2008. NetworkBLAST: Comparative analysis of Protein Networks. *Bioinformatics* 24 (4), 594–596.
- Kazemi, E., Hassani, H., Grossglauer, M., Modarres, H.P., 2016. Proper: Global protein interaction network alignment through percolation matching. *BMC Bioinformatics* 17 (1), 527.
- Kelley, B.P., Yuan, B., Lewitter, F., *et al.*, 2004. PathBLAST: A tool for alignment of protein interaction networks. *Nucleic Acids Research* 32 (suppl_2), W83–W88.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., *et al.*, 1983. Optimization by simulated annealing. *Science* 220 (4598), 671–680.
- Klau, G.W., 2009. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics* 10 (Suppl 1), S59.
- Kollias, G., Mohammadi, S., Grama, A., 2012. Network similarity decomposition (NSD): A fast and scalable approach to network alignment. *IEEE Transactions on Knowledge and Data Engineering* 24 (12), 2232–2243.
- Kollias, G., Sathe, M., Mohammadi, S., Grama, A., 2013. A fast approach to global alignment of protein-protein interaction networks. *BMC Research Notes* 6 (1), 35.
- Korsah, G.A., Stentz, A.T., Dias, M.B., 2007. The dynamic Hungarian algorithm for the assignment problem with changing costs, Technical Report. Pittsburgh, PA: Robotics Institute (CMU-RI-TR-07-27).
- Koyutürk, M., Kim, Y., Topkara, U., Szpankowski, S., Grama, A., 2006. Pairwise alignment of protein interaction networks. *Journal of Computational Biology* 13 (2), 182–199.
- Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., Pržulj, N., 2010. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*. [rsif2010063].
- Kuchaiev, O., Pržulj, N., 2011. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics* 27 (10), 1390–1396.
- Leordeanu, M., Hebert, M., 2005. A spectral technique for correspondence problems using pairwise constraints. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2005. ICCV 2005. vol. 2, pp. 1482–1489. IEEE.
- Liao, C.-S., Lu, K., Baym, M., Singh, R., Berger, B., 2009. IsorankN: Spectral methods for global alignment of multiple protein networks. *Bioinformatics* 25 (12), i253–i258.
- Malod-Dognin, N., Pržulj, N., 2015. L-graal: Lagrangian graphlet-based network aligner. *Bioinformatics*. btv130.
- Memišević, V., Pržulj, N., 2012. C-graal: Common-neighbors-based global graph alignment of biological networks. *Integrative Biology* 4 (7), 734–743.
- Milenković, T., Ng, W.L., Hayes, W., Pržulj, N., 2010. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*. 8).
- Milenković, T., Pržulj, N., 2012. Topological characteristics of molecular networks. *Functional Coherence of Molecular Networks in Bioinformatics*. New York: Springer, pp. 15–48.
- Milenković, T., Zhao, H., Faisal, F.E., 2013. Global network alignment in the context of aging. In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, p. 23. ACM.

- Neyshabur, B., Khadem, A., Hashemifar, S., Arab, S.S., 2013. NETAL: A new graph-based method for global alignment of protein-protein interaction networks. *Bioinformatics* 29 (13), 1654–1662.
- Notredame, C., Higgins, D.G., Heringa, J., 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302 (1), 205–217.
- Pache, R.A., Ceol, A., Aloy, P., 2012. NetAligner-a network alignment server to compare complexes, pathways and whole interactomes. *Nucleic Acids Research* 40 (W1), W157–W161.
- Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, previous number = SIDL-WP-1999-0120. Available at: <http://ilpubs.stanford.edu:8090/422/>.
- Pal, D., 2006. On Gene Ontology and function annotation. *Bioinformation* 1 (3), 97.
- Park, D., Singh, R., Baym, M., Liao, C.-S., Berger, B., 2011. Isobase: A database of functionally related proteins across PPI networks. *Nucleic Acids Research* 39 (suppl 1), D295–D300.
- Patro, R., Kingsford, C., 2012. Global network alignment using multiscale spectral signatures. *Bioinformatics* 28 (23), 3105–3114.
- Pesquita, C., Faria, D., Falcao, A.O., Lord, P., Couto, F.M., 2009. Semantic similarity in biomedical ontologies. *PLoS Computational Biology* 5 (7), e1000443.
- Phan, H.T., Sternberg, M.J., 2012. PINALOG: A novel approach to align protein interaction networks – implications for complex detection and function prediction. *Bioinformatics* 28 (9), 1239–1245.
- Preis, R., 1999. Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs. In: *Symposium on Theoretical Aspects of Computer Science* 99, pp. 259–269. Berlin: Springer.
- Pržulj, N., 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics* 23 (2), e177–e183.
- Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.
- Sahraeian, S.M., Yoon, B.-J., 2012a. Resque: Network reduction using semi-Markov random walk scores for efficient querying of biological networks. *Bioinformatics* 28 (16), 2129–2136.
- Sahraeian, S.M.E., Yoon, B.-J., 2012b. A network synthesis model for generating protein interaction network families. *PLoS ONE* 7 (8), e41474.
- Sahraeian, S.M.E., Yoon, B.-J., 2013. SMETANA: Accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS One* 8 (7), e67995.
- Saraph, V., Milenkovic, T., 2014. MAGNA: Maximizing accuracy in global network alignment. *Bioinformatics* 30 (20), 2931–2940.
- Sharan, R., Suthram, S., Kelley, R.M., *et al.*, 2005. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America* 102 (6), 1974–1979.
- Shih, Y.-K., Parthasarathy, S., 2012. Scalable global alignment for multiple biological networks. *BMC Bioinformatics* 13 (Suppl 3), S11.
- Singh, R., Xu, J., Berger, B., 2007. Pairwise global alignment of protein interaction networks by matching neighborhood topology. *Research in Computational Molecular Biology*. Springer. pp. 16–31.
- Singh, R., Xu, J., Berger, B., 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences* 105 (35), 12763–12768.
- Stumpf, M.P., Thorne, T., de Silva, E., *et al.*, 2008. Estimating the size of the human interactome. *Proceedings of the National Academy of Sciences* 105 (19), 6959–6964.
- Tatusov, R.L., Koonin, E.V., Lipman, D.J., 1997. A genomic perspective on protein families. *Science* 278 (5338), 631–637.
- Uetz, P., Giot, L., Cagney, G., *et al.*, 2000. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature* 403 (6770), 623–627.
- Vijayan, V., Saraph, V., Milenkovic, T., 2015. MAGNA++: Maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*. bttv161.
- Zaslavskiy, M., Bach, F., Vert, J.-P., 2009. Global alignment of protein-protein interaction networks by graph matching methods. *Bioinformatics* 25 (12), i259–i267.

Relevant Websites

<http://netaligner.irbbarcelona.org/NetAligner>.
<http://www.cs.tau.ac.il/~bnet/networkblast.htm>
 NetworkBLAST.
<http://pathblast.org/>
 PathBLAST.

Visualization of Biomedical Networks

Anne-Christin Hauschild, Krembil Research Institute, Toronto, ON, Canada

Chiara Pastrello and Andrea EM Rossos, Krembil Research Institute, Toronto, ON, Canada

Igor Jurisica, University of Toronto, ON, Canada and Slovak Academy of Sciences, Bratislava, Slovakia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Motivation

Biomedical research produces and needs to manage a large variety of information, leading to a diversity of computational challenges. Different high-throughput techniques produce the largest fraction of these data, contributing to improving our understanding of biological mechanisms in healthy and disease states. Hence the necessity for agile software solutions to adequately handle and analyze this information has become continuously more evident (Otasek *et al.*, 2012). Systems biology is a large part of computational biology that focuses on the comprehensive analysis of the relations between different biological factors, which leads to complex biomedical networks, such as protein-protein interactions (PPI), gene regulatory networks, or patient similarity networks. An important task within this framework is network visualization, analysis and interpretation, which is essential for interpretation and understanding of biomedical data. As data complexity increases with time, gaining insight from data becomes challenging without effective and scalable visualization methods. Individual network visualization tools differ greatly with respect to the supported features, scalability, and the analyses they enable. Further, it is important to consider the variety of potential users, who have a broad range of skills, expectations, and backgrounds, ranging from medicine and biology to computer science. Therefore, packages for network visualization must satisfy a diverse list of requirements (Agapito *et al.*, 2013):

- Fast and clear rendering of huge network structures and substructures;
- Interoperability with different databases that store biological network data, such as pathway or protein interaction databases;
- Integration of various standard annotation data sources, e.g., molecular function or localization from biological ontologies;
- Large range of functionalities and layouts, as well as feature expandability;
- User friendly querying and selecting of sub-networks (highlighting, zoom and focus);
- Compatibility with standard data input and output formats used for biological networks.

In this article, we provide a short introduction to various kinds of biological networks, their features and special visualization requirements, followed by a detailed description of the different layouts, their strengths and weaknesses, and discuss possible use cases. Furthermore, we elaborate how these networks can be annotated with additional information such as cellular location, function, or with “omics” data. Finally, the most commonly used software packages will be presented, their features compared, and their performance in terms of the defined requirements discussed, see [Table 1](#).

Biological Networks

The majority of biomedical networks incorporate data generated by various analytical technologies, such as metabolomics, transcriptomics or genomics. These “omics” data provide information about various states, molecules, or conditions across diverse biological mechanisms. Organisms’ phenotypes are the result of the combination of their genotype and the environment they live in. Research aimed at understanding any molecular mechanism underlying such phenotypes must take into account all available data, and not analyze single experiments and single condition data separately. Over the last several years, large proportions of the available data have been organized in public databases, which provide easy and centralized access via manual and automated queries. Modern tools dedicated to analyzing and visualizing biomedical networks can directly query these sources to answer specific research questions, integrate and annotate the retrieved data, and generate the corresponding visualizations and solutions needed to aid in data modeling and interpretation. [Fig. 1](#) gives an overview of such visualization and analytics workflow. In the following paragraphs we will briefly introduce the most common types of biomedical networks.

Epigenetic and Gene Regulatory Networks

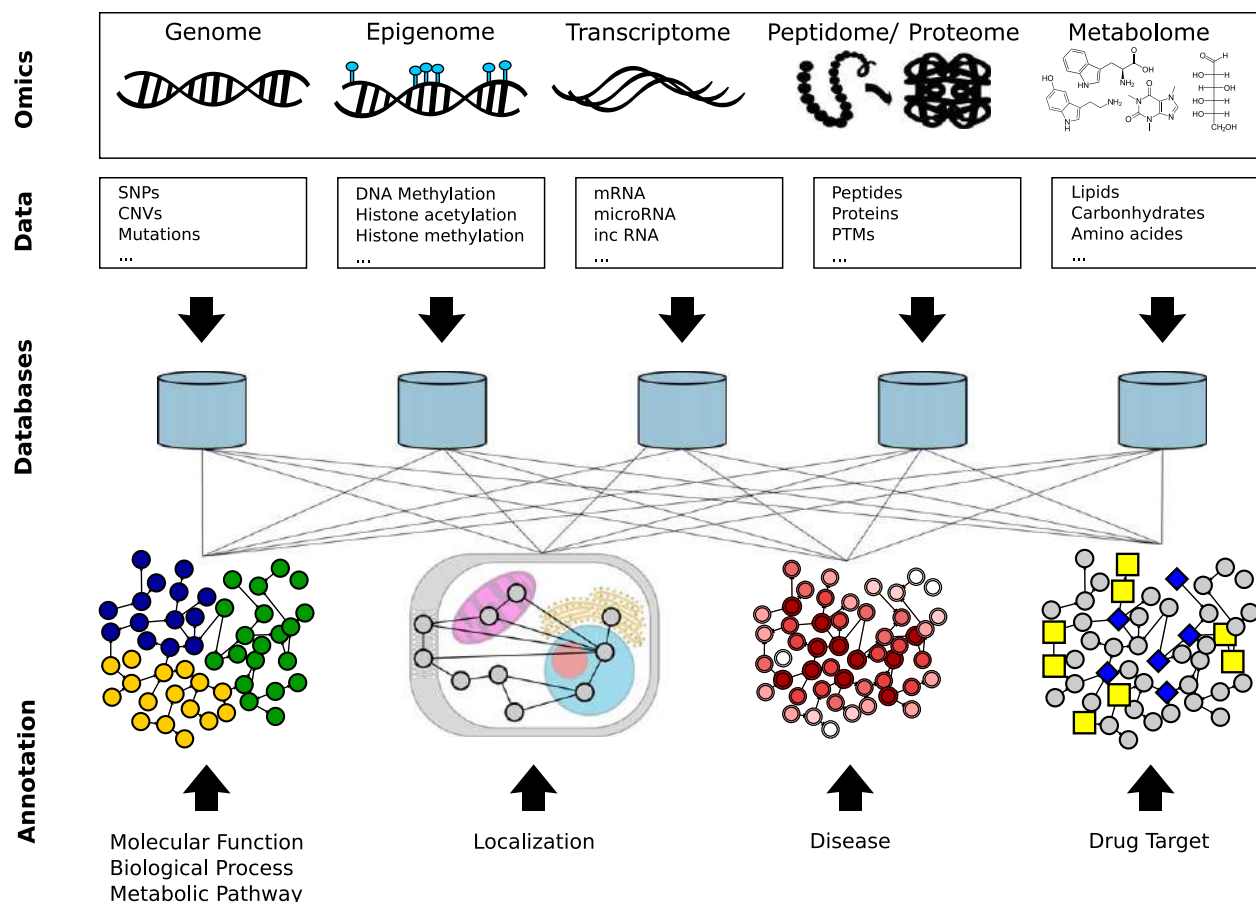
Complex regulatory genetic and epigenetic systems mediate the expression of thousands of genes at any given developmental, adult, or disease stage of an individual (Davidson and Levin, 2005). The functional relationship between these particular genes, and their associated regulatory modules, define gene regulatory networks (GRN).

Gene Coexpression Networks

Coexpression networks are based on large scale experiments evaluating the concurrent transcription of coding or non-coding genome sequences, which can include microarray technologies as well as RNAseq data (Okamura *et al.*, 2014). Large-scale analysis

Table 1 Software packages and web-applications for network visualization

Tool	URL	First release (latest version, year) ^a	Reference
Software			
BiNA	www.bina.unipax.info/	2007 (2.4.1.1, 2013)	(Küntzer <i>et al.</i> , 2007)
Cytoscape	www.cytoscape.org/	2003 (3.5.1, 2017)	(Shannon <i>et al.</i> , 2003)
mDraw	www.weizmann.ac.il/mcb/UriAlon/download/network-motif-software	NA (1.0, NA)	NA
Medusa	sites.google.com/site/medusa3visualization/	2005 (3, 2011)	(Hooper and Bork, 2005)
NaviCluster	navicluster.cb.k.u-tokyo.ac.jp/	2011 (2.0, 2011)	(Praneenarat <i>et al.</i> , 2011)
NAVIGATOR	ophid.utoronto.ca/navigator/	2005 (2.9.14, 2017)	(Brown <i>et al.</i> , 2009)
Pajek	pajek.imfm.si/	1997 (5.01, 2017)	(Batagelj and Mrvar, 2001)
VANESA	agbi.techfak.uni-bielefeld.de/vanesa/	2012 (0.3, 2017)	(Proß <i>et al.</i> , 2012)
VANTED	sourceforge.net/projects/vanted/	2006 (2.6.3, 2016)	(Junker <i>et al.</i> , 2006)
Vis ANT	visant.bu.edu	2003 (5.51, 2016)	(Hu <i>et al.</i> , 2004)
Web-application			
30 mics	3omics.cmdm.tw/	2013 (NA)	(Kuo <i>et al.</i> , 2013)
Cell Maps	cellmaps.babelomics.org/	2015 (1.0.4, 2015)	NA
esy N	www.esyn.org/	2014 (2.1, 2015)	(Bean <i>et al.</i> , 2014)
Net Gestalt	www.netgestalt.org/	2013 (1.2, NA)	(Shi <i>et al.</i> , 2013)
PINV	biosual.cbio.uct.ac.za/pinv.html	2014 (NA)	(Salazar <i>et al.</i> , 2014)

^aLatest version as available in August 2017.**Fig. 1** General overview of biological mechanisms and the corresponding types of omics data, which are subsequently stored in a number of network databases. The data can be utilized to create and visualize various types of biomedical networks that are finally, annotated with labels such as cellular location or function.

of coexpression networks based on similarity measures such as correlation, is a powerful tool to elucidate putative functional modules in the genome.

Protein-Protein Interaction Networks

Proteins perform the majority of cellular activities, including cell replication or growth, and are building blocks for most cellular structures. Many of these tasks are accomplished through stable and transient protein-protein interactions (PPIs). The entirety of all PPIs in a specific organism is commonly called the “interactome” (Cusick *et al.*, 2005). In order to computationally evaluate the interactome, it is typically represented as a graph or network.

Metabolic Networks

The entirety of all life-sustaining chemical transformations within a cell or microorganism is called metabolism (Hatzimanikatis *et al.*, 2004). Biochemical reactions are typically represented by edges connecting substrate and product metabolites, which are represented by product nodes, and create complex metabolic networks. The network structures further incorporate the different aspects of enzyme chemistry and structure. For instance, enzymes mediate, accelerate or decelerate metabolic processes and are a crucial element of the metabolic network.

Signaling Networks

Cellular signaling is the process of communication that mediates the basic activities and coordination of cells. The individual entities perceive and correctly respond to their immediate micro-environment. This cell communication is the basis of tissue growth, development, and repair, as well as the cellular immune response (Smith *et al.*, 2015). In a signaling network, intracellular or cell-surface receptors are activated by specific ligands, such as hormones or neurotransmitters, and the signal is carried through different cell compartments to reach the desired phenotype (Dinasarapu *et al.*, 2011).

Phylogenetic Networks

Phylogenetic trees are used to represent the processes and patterns of evolution. Visually representing the similarities among different phylogenetic trees supported by different genes is a powerful way for investigators to gain a better perspective on their evolutionary questions. In a phylogenetic network, edges represent evolutionary relationships between organisms (Wilgenbusch *et al.*, 2017).

Ecological Networks

Ecological systems can be represented by networks that comprise all interactions among the organisms within it (Ings *et al.*, 2009). The aim of ecological network analysis is to identify and understand the mechanisms on which the ecosystem is built, or that affect the ecosystem's stability. They can be subdivided into three broad types: food webs, mutualistic networks (including mutualistic interactions among plants and animals), and host-parasites networks. Visualizing the interactions and relations of different species enables a better understanding of the key principles and pillars of an ecosystem, and identifies crucial weak points.

Biological Networks Databases

A number of large-scale databases are available, providing easy and fast access to various network data. Prominent examples are BioCyc (Caspi *et al.*, 2014), KEGG (Kanehisa and Goto, 2000) and Reactome (Croft *et al.*, 2013) pathway databases for metabolic networks and signaling pathways, or BioGRID (Breitkreutz *et al.*, 2008), the IID (Kotlyar *et al.*, 2016), and STRING (Szklarczyk *et al.*, 2016) for PPI networks. For successful network analysis and visualization tools, it is useful to support either automated or easy manual data retrieval from these databases (see Section Network Visualization Software).

Layouts

An appropriate mathematical representation of a biomedical network is a graph. Biological objects such as proteins, genes, or species can be represented by nodes, while their interactions, for instance, biological reactions, gene regulation, or the genetic similarity of species represents the edges. This representation allows one to analyze the structural features of a network and reveals key components, for example, genes connected to certain cellular or disease mechanisms. It also allows one to apply a number of visualization techniques that enable the user to visually explore the data. The main goal of these techniques is to gain information by optimally presenting all objects and relations of the graph to the user (Agapito *et al.*, 2013). These so called layout algorithms, have been developed and used in mathematics and computer science for a long time. They aim to find an optimal set of coordinates for the nodes in the network in order to highlight underlying structures. To further improve network visualizations, both nodes and edges can be annotated with additional information, which can be highlighted by visual properties (e.g., color or shape), affecting the requirements and scalability of the layout algorithms. These additional visual features are described in Section Biological Network

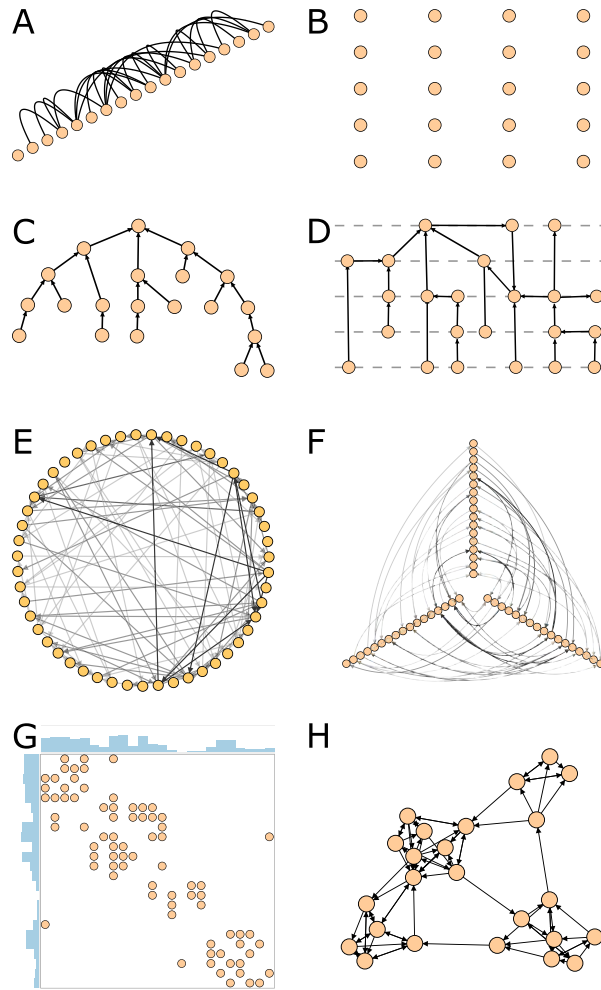


Fig. 2 Network Layouts: (A) planar layout; (B) grid layout; (C) tree layout; (D) hierarchical layout; (E) circular layout; (F) hive plot; (G) heat map; (H) clustering layout. Networks created with the aid of NAViGaTOR 3.0 (Brown *et al.*, 2009). Available at: <http://ophid.utoronto.ca/navigator/>, exported in SVG, and finalized using Inkscape. Available at: <https://inkscape.org>.

Annotation about “Network Annotation.” Different criteria have been proposed, for example, to satisfy certain well-defined aesthetic principles, or other constraints, such as, flexibility, minimal node overlap, edge crossing, or the size of the drawing area. A brief description of the most common layout algorithms and their features is presented below and illustrated in Fig. 2.

Random Layout

The most simple, and often the initial, representation of a network is the random layout. The nodes and edges of a graph are arranged in a random way in the specified area. The main advantage of this technique is its simplicity and performance for both implementation and use. However, with increasing numbers of nodes and edges, the graph results in a “hair-ball,” which cannot be interpreted due to chaotic edge-crossings and node overlap. This type of layout is often applied to idealize architectures for dynamical models of genetic networks (Li and Kurata, 2005; Agapito *et al.*, 2013).

Planar Layout

Planar graph layouts are node centered and aim to reduce edge crossing by placing the nodes of the network along a straight line in the Euclidean plane, see Fig. 2(A). The main advantage is that individual nodes can be arranged with respect to any ordered node attribute, such as, node name, or biological features such as gene ontology and biological function. The edges are drawn as convex curves in one of the two half-planes separated by the line. A special instance of a line layout is the arc diagram, where each of the edges is drawn as a semicircle without crossing edges (Masuda *et al.*, 1990). The planar layout is simple and easy to visualize for small and sparse networks. However, for large graphs, screen space is not optimally used and an overview requires the use of a very small scale. Additionally, for highly connected networks, minimizing the number of edge crossings is very difficult. Frequently, planar layouts are combined into more complex layouts, for instance parallel lines or the hive layout, see Section Hive Plot Layout.

Grid Layout

The grid layout was developed with the aim of treating the network as a system of interacting particles, where the nodes are placed on a 2-dimensional, Cartesian grid (Li and Kurata, 2005), see Fig. 2(B). The nodes attract and repel each other according to a predefined energy function based on the network topological structure. Each specific configuration and corresponding layout, therefore, is rated by a certain cost based on this function. Similar to a molecular system, a low cost results in low energy levels and corresponds to a stable configuration. This layout is a systematic way of representing simple, locally connected but globally sparse, networks. A popular variation of the grid layout is the orthogonal layout, where edges are preferentially aligned with the grid and nodes are spread upon a larger grid, in order to minimize edge crossings, and to preserve local network topologies. Visualizing metabolic pathways according to conventional, grid ordered, textbook-like drawings is a difficult task for standard graph drawing techniques. These techniques mostly focus on a few linear or circular main structures or backbones and arrange other components relative to them (Li and Kurata, 2005). However, these techniques are generally not applicable to other networks, such as gene regulatory networks or signal transduction pathways, since it is difficult to find any mostly isolated paths to use as a backbone.

Tree Layout

This layout simply draws the nodes of the graph in a tree-like structure, often with a hierarchical organization, which creates clear and easy to understand graphs, see Fig. 2(C). It provides a natural way to visualize the global structure of a complex pathway (Di Battista *et al.*, 1994). A major advantage is that most implementations show a low complexity and are therefore scalable and highly efficient for visualizing large networks. However, problems occur when arranging a huge number of nodes in limited space. There are different types of tree layouts, including unrooted or rooted trees. The later is commonly drawn with straight or orthogonal polyline edges, while nodes are placed along horizontal lines according to their level. This type of layout is frequently applied to represent hierarchical structures like phylogenetic or family trees, and search trees. The T-REX web server, for instance, specializes in the analysis and visualization of phylogenetic trees and networks (see Relevant Website section) (Boc *et al.*, 2012). Nevertheless, this representation is generally not suited for non-tree structures that contain loops, such as those found in other biomedical networks, such as PPI or gene regulatory networks. In contrast, unrooted trees, also called free trees, do not have a specific root, and therefore do not follow a top to bottom hierarchy. Nodes are commonly arranged in concentric circles, according to their level, around the graph-theoretic center of the tree (Di Battista *et al.*, 1994).

Hierarchical Layout

In a hierarchical layout or layered drawing, nodes are constrained to lie on a set of equally spaced horizontal lines, called layers, with the goal of reducing the number of edge crossings (Di Battista *et al.*, 1994), see Fig. 2(D). This layout has efficient and scalable implementations and creates nice visualizations for sparse networks, even if the number of nodes is large (Agapito *et al.*, 2013). However, the minimization of the number of overlapping edges is an NP-complete problem, which prevents its application for larger graphs (> 1000 objects). It can be used effectively for applications that provide a hierarchical assignment for the nodes, such as signaling pathways, pharmacokinetic models or the food hierarchy of a food web in an ecosystem. The layout can also be used in directed networks, where it is customary to order the graph such that all edges flow in the same direction, e.g., from top to bottom, or from left to right.

Circular Layout

The nodes in the circular layout are placed equidistant from the graph center (Mäkinen, 1988), aiming to minimize overlapping nodes, see Fig. 2(E). Despite its simplicity, the technique is widely used to visualize biological networks such as protein complexes, drug-protein, miRNA-gene and phylogenetic networks (Agapito *et al.*, 2013). A popular variation of the circular layout is the circos plot, which is generally used to visualize genomic or transcriptomic information (Krzywinski *et al.*, 2009). While the genomic region, a chromosome, or a specific set of genes is drawn as a large circle, edges can depict the gene regulatory interactions between the loci of distal genes, for instance, in cancer (Sumazin *et al.*, 2011).

Hive Plot Layout

More recently, a novel layout called the hive plot has gained popularity. Similar to other approaches, its objective is to create rational, informative and reproducible layouts. The nodes are placed on multiple radially arranged straight lines originating in the center, representing linear axes with respect to a well-defined coordinate system, see Fig. 2(F). This creates a compact and reproducible layout with perceptual uniformity, where connections are easy to follow (Krzywinski *et al.*, 2011). A common application of hive plots is the comparison of transcript splicing ratios between individuals or genotypic groups (Wu *et al.*, 2013). While each of the linear axes corresponds to a splice junction, individual events are represented by curved edges that circle around the center, connecting all axes.

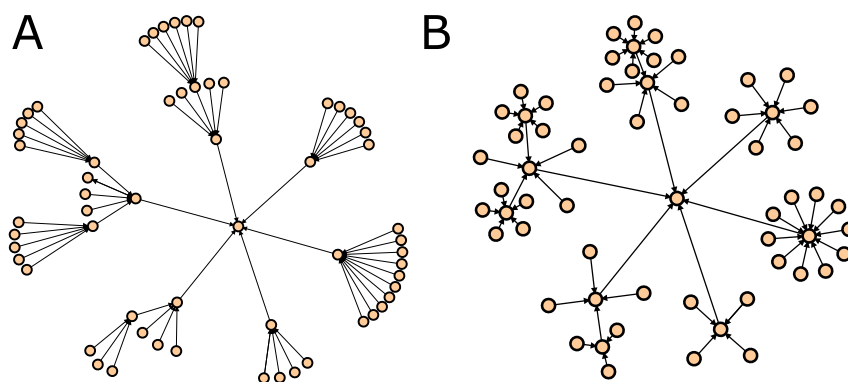


Fig. 3 Radial network layouts: (A) radial view; (B) balloon view. Networks created with the aid of NAViGaTOR 3.0 (Brown *et al.*, 2009). Available at: <http://ophid.utoronto.ca/navigator/>, exported in SVG, and finalized using Inkscape. Available at: <https://inkscape.org>.

Edge-Matrix Layout

Edge-matrix or heatmap visualizations of networks, can be utilized to depict the edges and edge weights between the vertices in a network. Directed networks result in a non-symmetric matrix. Weighed edges can be plotted via a range of colors, similar to the commonly used similarity matrix. This visualization can highlight densely connected subnetworks, by clustering the nodes according to a topological or content specific similarity (Gove *et al.*, 2011), see Section Graph Partitioning Algorithms for more details.

Cluster Layout

The aim of the cluster layout is to reduce the complexity of the network, by grouping similar nodes into a smaller number of visible representatives. This improves the clarity of the graph and simultaneously substantially increases the performance of layout rendering. In order to cluster the nodes, a similarity or distance metric is required, which can either be content or topology based. These could be, for instance, sequence similarity or based on topological cliques (see Section Graph Partitioning Algorithms) (Agapito *et al.*, 2013). A special variation of this layout, called a clustered circular layout, was developed as an informative technique to visualize PPI networks. The proteins can be clustered or annotated by their function, e.g., Gene Ontology (Fung *et al.*, 2010).

Radial Layout

The radial layout or balloon layout are special tree-based visualizations incorporating the idea of circular layouts (Herman *et al.*, 2000). The radial approach places the nodes onto concentric circles relative to their depth in the tree. Each subtree is subsequently plotted over a sector of the circle, while the algorithm guarantees that there is no overlap between adjacent sectors, see Fig. 3(A). The balloon view of a tree projects all sibling subtrees onto circles attached to the father node, see Fig. 3(B). An example for the radial network is presented by Djebbari *et al.*, where the center node. represents the biological complex ribonucleotide reductase. Proteins interacting with the components of a protein complex are connected via edges to a composite vertex, that represents the entire complex. Visualizations like this can show higher-levels of organization while maintaining the hierarchical structure of the biological complex (Djebbari *et al.*, 2011).

3D Layout

While analytical technologies moved from single experiments to high-throughput methods, biological networks started growing tremendously in the number of nodes as well as number of connections that they contain, creating the so called hairball effect. These huge numbers of nodes, densely connected with each other, result in various challenges for the performance of network analyses, but especially for visualization (Pavlopoulos *et al.*, 2015). A potential solution is to shift from two-dimensional (2D) to three-dimensional (3D) representations. A few software packages, such as Pajek, a previous version of Navigator, or the Cytoscape plugin 3DScape (Wang *et al.*, 2013), utilize the 3D space to show data in a virtual 3D universe.

Besides simply applying the same techniques in 3D, there are different ways to utilize the third dimension; for instance, by systematically implementing the concept of multiple layers for hierarchies as provided by Arena3D (Pavlopoulos *et al.*, 2008a).

In general, a 2D visualization provides the user with immediate visual feedback, while, in contrast, 3D rendering requires more interactivity between user and software with respect to the data but facilitates the potential discovery of relevant features that remain hidden in 2D. The relative advantages of 3D versus 2D visualization, depend on the task at hand, the data, and user skills; therefore, hardware acceleration and performance versus visual benefits have to be carefully considered when deciding between 2D and 3D visualizations (Pavlopoulos *et al.*, 2015).

Layout Algorithms

Automated network layouts, such as force-directed layouts, can be considered to be data-driven, unsupervised methods. They aim for a node organization based on network topology or connectivity, and are *de facto* multi-dimensionality reduction techniques (Baryshnikova, 2016).

Force-Based Algorithms

In general, force-based algorithms can be seen as a variant of multidimensional scaling (MDS) applied to graph-theoretic distances. It is a state-of-the-art group of algorithms that aims at estimating layouts for directed and undirected graphs on the basis of stress minimization. The elements of the graph are considered as objects with physical properties that are attracted and repelled by, e.g., electrical forces.

Spring embedded algorithms

Among the first force-directed approaches are the so called spring embedded algorithms, such as the Fruchterman & Reingold (Fruchterman and Reingold, 1991) and Eades algorithms (Agapito *et al.*, 2013). In these approaches, each node is represented as an electrically charged object, which is linked by springs representing an edge. In such a system, objects of equal charge repel each other and edges represent resistance, while two nodes attract each other when they have opposite charges. The system converges to an equilibrium of attracting and repelling forces by iteratively updating the positions of each node. These algorithms are particularly suited for weighted networks, where each edge is assigned a certain value representing, for instance, sequence similarity. The complexity of the algorithm is relative to the size (nodes and edges) of a graph, resulting in very time-consuming layout processes for large graphs. Due to their ability to effectively represent complex biomedical networks, force-directed layouts are implemented by the majority of network visualizations tools.

Kamada-Kawai algorithm

In contrast to the previous algorithm, this algorithm is more complex to encode, but often provides excellent results (Agapito *et al.*, 2013). Kamada-Kawai algorithm is based on the concept of graph theoretic distances between the objects (Kamada and Kawai, 1989). The distances between the different nodes can be determined by the lengths of shortest paths between each pair of nodes. These distances can be represented as spring forces that are directly proportional to the graph theoretic distances.

Graph drawing with intelligent placement

Graph dRrawing with Intelligent Placement – GRIP – has been designed to effectively visualize very large graphs (Gajer and Kobourov, 2002). It applies a multi-dimensional force-directed method in combination with a fast energy function minimization. A simple recursive coarsening scheme is employed, where vertices are placed intelligently, rather than randomly. Additionally, sets of nodes are placed simultaneously, at positions in close proximity to their optimal location. Therefore, the running time and space complexity of a network are near linear in the number of vertices.

Graph Partitioning Algorithms

While most layout algorithms try to optimize the general overview and aesthetics of a network, others such as graph partitioning algorithms, intend to infer topological meaning. Further, it is often advantageous to utilize such partitions for abstraction or complexity reduction, in order to reduce the visual complexity of the network (Herman *et al.*, 2000). Various partitioning algorithms are available and aim to identify local, topological or content based groupings. Methods that focus on structural information about the graph are often called structure-based partitioning methods; popular examples include community, motif, clique search, and structure-based clustering. Various software packages have been dedicated to solve these tasks, such as GraphCrunch (Kuchaiev *et al.*, 2011) or Mfinder (Milo *et al.*, 2002). In contrast, content-based approaches use semantic data associated with the graph elements to perform clustering. In a plane graph (see Fig. 4(A)), partitioning information can be utilized to visually indicate the groups within a network, by color or shape (see Fig. 4(B)), to reorganize the graph according to the grouping (see Fig. 4(C)), or to reduce the visual complexity of the network by replacing the elements of each group by a new representative object and applying mechanisms such as edge-bundling or node collapsing (see Fig. 4(D)). Decreasing the number of objects to be plotted improves both the clarity and performance of layout and rendering (Herman *et al.*, 2000). Furthermore, several tools enable stepwise moving up and down between different levels of detail as needed in order to evaluate biological meanings.

Probabilistic Graph Layout

The most recent combination of force-based and partitioning algorithms is called probabilistic graph layout, and aims to visualize uncertain networks using the underlying probability distributions of nodes and edges (Schulz *et al.*, 2017). Furthermore, the algorithm maps the probability distributions of edges to visually perceivable colored areas and shapes. Schulz *et al.* explored many aspects of filtering and depicting sets of overlapping partitions and points while maintaining the visibility of the underlying network topology, and thereby addressed various challenges raised by both force-based and partitioning approaches.

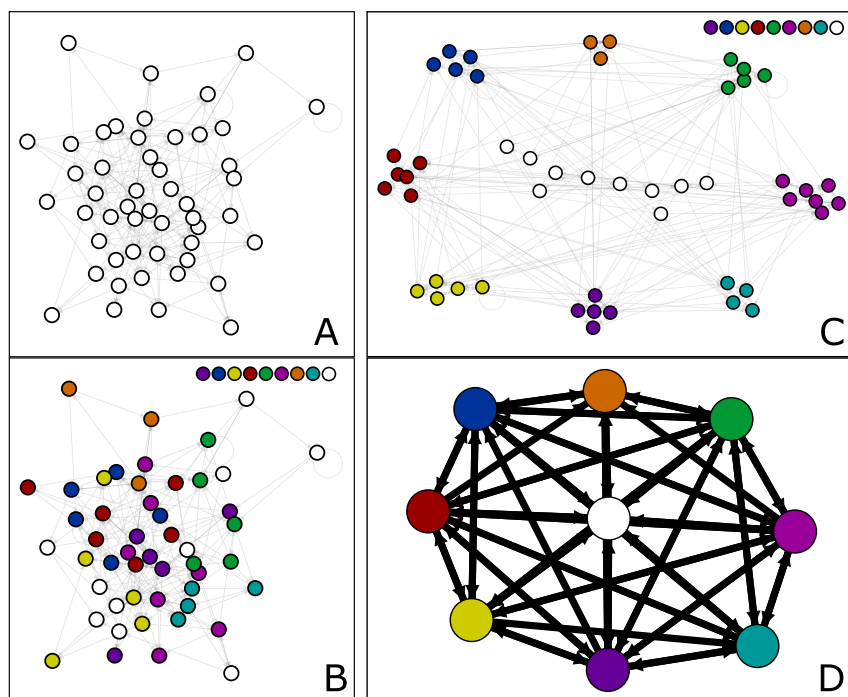


Fig. 4 Graph Partitioning Layout. A randomly generated graph visualized with the GRIP layout (A), application of community search by Navigator 3.0, and coloring by community (B), ordering according to communities each visualized using GRID (C), complexity reduction via merging community nodes and edges (D). Networks created in NAViGaTOR 3.0 (Brown *et al.*, 2009). Available at: <http://ophid.utoronto.ca/navigator/>, exported in SVG, and finalized using Inkscape. Available at: <https://inkscape.org>.

Biological Network Annotation

The visualization of large biological networks often results in an incomprehensible hairball, which is not directly interpretable. Although visual graph layouts hold great potential for uncovering hidden patterns within these networks (Baryshnikova, 2016), they are typically optimized to generate esthetically pleasing visualizations and rarely focus on systematic analysis. Therefore, it may be difficult to extract useful information with respect to biological meaning.

The integration of additional information about the biological entities represented in the network, for instance genes or proteins, may facilitate the organization and the analysis of the networks, and therefore clarify and highlight key attributes (Koh *et al.*, 2012), see Fig. 1. These node or edge annotations can be included in the original data, such as network files, or imported from biological databases. Incorporating this annotation information in the network visualization, by modifying the appearance of nodes and edges, can greatly increase the interpretability of the network, see Section Visualization of Annotation Visual Annotation for examples. In the next paragraphs, we will give an overview of examples of annotation resources that provide valuable information that can be used to gain further insight into the molecular mechanisms the network or a sub-network represents. One can categorize annotation information either according to type of data (quantitative vs. qualitative) or data context (disease relevance, tissue, or cellular localization for genes or proteins, as well as habitats for species networks). In the following section we will discuss the value of quantitative and qualitative annotation for biological networks and will give examples for both.

Qualitative Annotations

Biological networks can be annotated on different conceptual and functional levels. For instance, elements in gene regulatory networks or PPI networks can be associated with pathways, tissues, sub-cellular localizations or diseases. These qualitative annotations can be visualized through qualitative visual attributes, such as color, size, outline, and shape.

Widely used sets of annotations are available from public biomedical databases and ontologies, often initiated by international consortia, aiming to provide consistent descriptions of biomedical entities, such as proteins and other gene products (Koh *et al.*, 2012). An ontology formally names and defines the terms, properties, and relationships among the entities that exist in a particular domain. In the following subsections we will discuss some of the most relevant ontologies for the annotation of biological networks. Moreover, we will introduce a few of the most popular types of annotation. The majority of biomedical network research tackles the systematic analysis of genes, their products, their interactions, or their involvement in pathways; therefore, the focus of this section is set accordingly. Finally, we will also present relevant examples for other biological or medical networks.

Biological function

Various databases and ontologies provide information about the biological functions of proteins, for instance. The Gene Ontology (GO) initiative was created to provide consistent descriptions for large sets of gene products across databases, and has become a widely used source of information (Ashburner *et al.*, 2000). It provides three controlled and hierarchically organized vocabularies, “biological process,” “molecular function,” “cellular location,” each corresponding to an independent biological domain. The GO project aims to facilitate manual and automated access to information in the most effective (easy and efficient) way, and is therefore suitable for large scale annotation of PPI networks, and enrichment analysis (Smith *et al.*, 2007). It enables the identification of significantly over- or under-represented annotations in a given protein list or a PPI sub-network. The most commonly used ontology for this purpose is the biological process ontology. It is defined as a series of individual processes accomplished by one or more ordered assemblies of molecular functions (for instance, glycolysis or apoptosis). In contrast, the molecular function ontology encodes the specific role or roles a certain protein can possess and fulfill at a molecular level, for instance, catalytic activity or transmembrane transport. A more detailed description of GO definitions and policies can be found on the GO Web site (see Relevant Websites section).

Another essential ontological representation of protein-related entities is the PRotein Ontology (PRO) (Smith *et al.*, 2007). It formally describes taxon-specific and taxon-neutral protein entities of three major areas: evolutionary relationships of proteins, splice variants encoded by the same gene, and information on protein complexes.

In summary, modules of proteins that are part of the same process can be identified and can provide a functional interpretation of the network (Koh *et al.*, 2012). The same holds for proteins that are evolutionarily related or are part of the same protein complex. Changes in splicing variants, in contrast, can indicate a change in cell function, differentiation, or disease.

Subcellular localization

Proteins perform their functions in one or more cellular compartments. Compartmentalization is one mechanism of molecular regulation, as only proteins located in the same location can interact. As mentioned before, GO also provides an ontology for cellular components, indicating the primary location of a protein within the cell, for instance, the nucleus or mitochondrion. Using such annotations, visualization tools like Cell Where enable the user to depict proteins in a network according to their localization in a cell, as highlighted in Fig. 5.

Unfortunately, subcellular localization data are incomplete, redundant and often poorly structured, and a big part of the proteome has only computationally predicted localization (Veres *et al.*, 2015). Yet, it is fundamental to annotate PPIs with their subcellular localization, especially when analyzing molecular processes that span compartments.

Tissue annotation

When focusing on tissue-specific data, only the respective part of the whole data is going to be included in the network (Lopes *et al.*, 2011). The BRENDA Tissue Ontology (BTO) represents a comprehensive structured vocabulary of tissue terms for the source of an

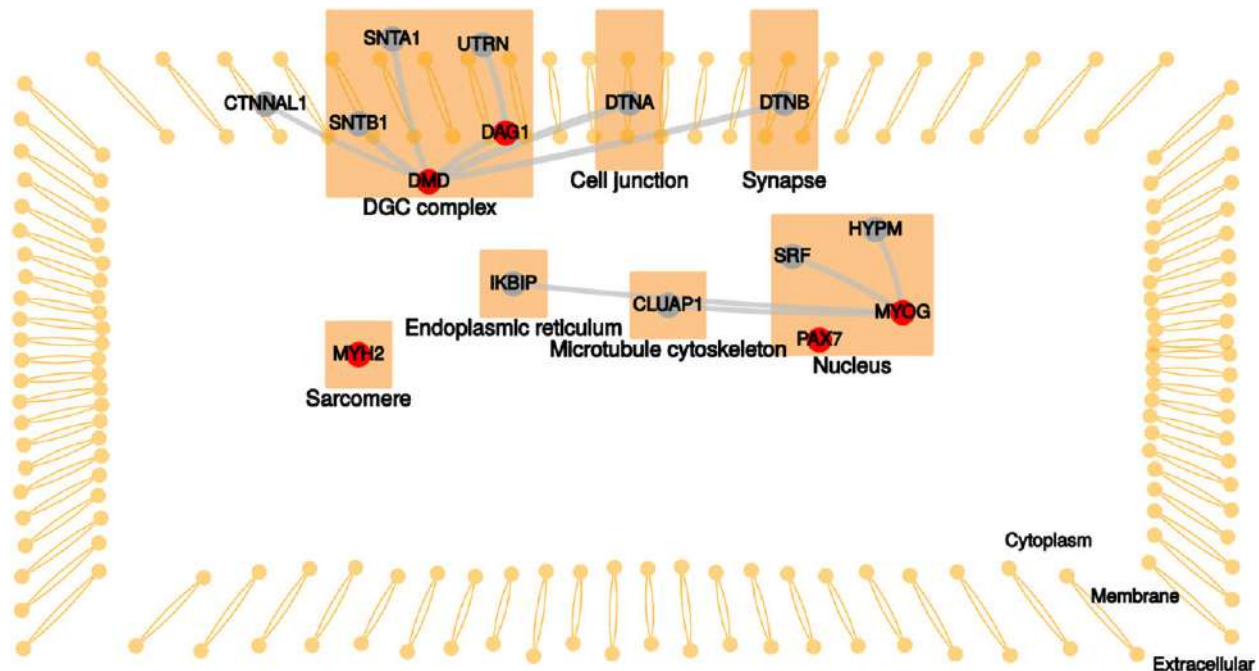


Fig. 5 Location annotation of a set of genes. The figure was created in Cell Where (Zhu *et al.*, 2015), using standard parameters that create an example network. Cell Where is available at: cellwhere-myo.rhcloud.com.

enzyme. While focusing on enzymes, it comprises annotations for tissues, cell lines, cell types, and cell cultures from uni- and multi-cellular organisms (Gremse *et al.*, 2010). Another valuable resource is the TissueNet database, which annotates PPIs with information about tissue coexpression across more than 40 human tissues. The database associates experimentally-identified PPIs with human tissues if both genes have been shown to be co-expressed in the same tissue. This bridges the lack of information in interactomes, as many PPIs have no tissue-contexts (Basha *et al.*, 2017). Similarly, the PPI database IID, provides tissue information based on protein expression data and RNA sequencing as automated annotation when querying for interactions (Kotlyar *et al.*, 2016).

Studies have shown that tissue-specific networks evolve tremendously during growth and differentiation and, subsequently, functionally similar tissues tend to share a large number of PPIs. This highlights the necessity for properly annotated networks, especially when studying tissue-specific effects or developmental stages (Liu *et al.*, 2014).

Disease

Diseases interfere with an organism on many molecular levels. For instance, genetic or epigenetic modifications can lead to changes of gene expression, protein de-regulation, alternative post-translational modification, and a large set of consequences for survival (Zhong *et al.*, 2009). Many diseases are caused by or result in a sequence of these changes, altering PPI or metabolic networks. Therefore, prioritizing gene and protein prognostic markers, or identifying the most useful drug target candidates benefits from comparing PPI network differences between healthy and disease conditions.

The aim of the Human Disease Ontology (DO) is to establish an open source ontology to facilitate the connections between biomedical data that is relevant to human diseases (Kibbe *et al.*, 2014). The system incorporates various sources and types of biomedical data, such as, genetic data, clinical data, and symptoms, linked to human disease. Other ontologies focus on specific diseases or disease groups, for instance, Infectious diseases, Cardiovascular Disease Ontology (CVDO) or the Mental Disease Ontology (MFO). Other resources are provided by public databases focusing on human diseases and PPI networks, such as, HIPPIE or SPECTRA. HIPPIE provides context-annotated PPI networks incorporating information, such as gene expression, gene ontology (GO) and MeSH disease headings (Alanis-Lobato *et al.*, 2016). SPECTRA is an integrated database associating tissue and tumor-specific protein interactions in humans (Micale *et al.*, 2016).

Quantitative Annotations

In contrast to qualitative information, quantitative annotations rely on continuous measures, that are determined analytically, calculated, or estimated by different techniques. For PPI, experimental results might include differential gene expression, percentage of promoter methylation, number of post-translational modifications, mutation status, or protein expression and abundance. These are all measures that can be used and visualized in a network, as well as any numeric attribute of an entity - such as weight, size, or speed of a species in a species network, or disease specificity in a drug network that is of relevance to the particular research question. Many databases collect quantitative data that can be used to annotate nodes, e.g., GEO (see Relevant Websites section), (Edgar *et al.*, 2002)), ArrayExpress, (Kolesnikov *et al.*, 2014)), The Protein Atlas (Uhlén *et al.*, 2015)), while very few databases include data that can be used to annotate edges - one example being COEXPRESdb (Okamura *et al.*, 2014)). Quantitative measures can be visualized through quantitative visual attributes such as size (node type and outline), transparency, or shades of colors (Fortney *et al.*, 2010; Li *et al.*, 2016; Zhang *et al.*, 2017; Zeng *et al.*, 2016), for examples see Fig. 6.

Visualization of Annotation

Recent software tools for network visualization support a large variety ways to indicate the pre-430 viously described quantitative or qualitative annotations. Quantitative differences or categories of entities can be depicted as vertices of different color, shape, or highlights. In contrast, qualitative measures are often indicated by transparency, color scales, or node sizes. Similar strategies hold for edges, utilizing thickness, color and dashed lines. Fig. 6 depicts the most commonly used techniques for network annotations.

A – Colored categories, qualitative: This type of visualization is often used to indicate different categories, such as the ones obtained from biological process or molecular function Gene Ontologies. Another example is metabolic networks, that comprise elements that can have different roles, such as reaction mediator, substrates, or products, which can be visualized with different colors or shapes.

B – Node highlighting, qualitative: Nodes that have special meaning or indications, such as drug targets or network hubs can be emphasized by highlights.

C – Shaped categories, qualitative: Similar to color, different shapes can indicate different groups or communities. In gene-regulatory or PPI networks, frequently a combination of both is used, where green and red, and up- and down-pointing arrows indicate up- and down-regulated genes, respectively.

D – Node transparency, quantitative: This technique can be used for instance to put emphasis on nodes that show a high range in a quantitative measure, such as expression, statistical significance, centrality, etc.

E – Node size, quantitative: Similar to the previous, it can be used to represent the importance of vertices with respect to a continuous feature. Both transparency and size can significantly increase the interpretability of a network by fading out or reducing unimportant information.

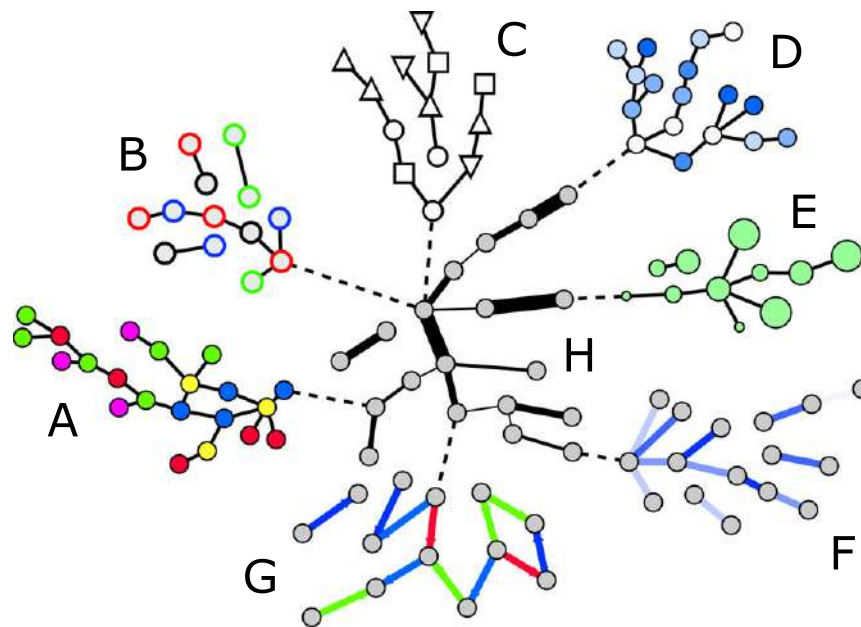


Fig. 6 The annotation of biological networks using different visualization techniques. Networks created in NAViGaTOR 3.0 (Brown *et al.*, 2009). Available at: <http://ophid.utoronto.ca/navigator/>, exported in SVG, and finalized using Inkscape. Available at: <https://inkscape.org>.

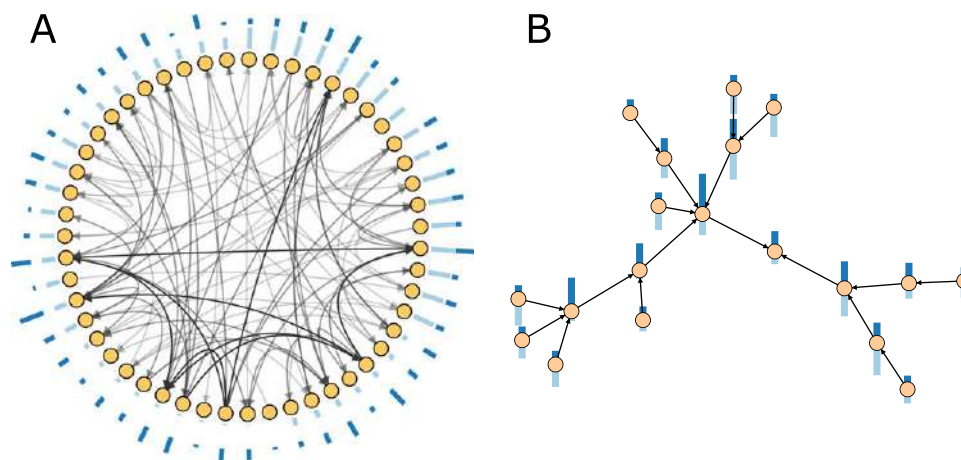


Fig. 7 The visualization of network annotations using the spokes techniques. Networks created in NAViGaTOR 3.0 (Brown *et al.*, 2009). Available at: <http://ophid.utoronto.ca/navigator/>, exported in SVG, and merged using Inkscape. Available at: <https://inkscape.org>.

F – Edge transparency, quantitative: Transparency can be used to indicate the strength of an interaction. In gene-regulatory networks it can be utilized to visualize the probability that a regulatory factor controls a certain gene.

G – Edge colors, qualitative: Different colors can indicate different categories of network relations, for instance, in PPI networks, the difference between protein complexes and regular interaction, levels of positive/negative co-expression, or PPI deregulation in normal versus cancer tissues.

H – Edge thickness, quantitative: Similar to color and shade, the thickness of an edge can indicate different strength or type of relations. This can be beneficial to visualize, for instance, gene co-expression in a PPI or gene-regulatory network.

In addition to the previously described visual annotations, it is possible to add bar plots, so-called spokes, representing quantitative attributes, to each node. **Fig. 7** depicts two examples of the spokes visualization: (A) is a circular plot; similar in design to a circos plot, it shows two stacked levels of annotation attached to the outer side of each node, while the edges are centered in the middle. (B) represents a typical GRIP – layout tree; again each node is annotated by two different attributes in two opposite directions. In theory an arbitrary number of features can be stacked on top of or next to each other. This kind of visual annotation often presents the information in a much clearer manner, especially in combination with circular plots.

Network Data Formats

A major challenge for most network software tools is the complexity of biomedical input data and its many file formats (Pavlopoulos *et al.*, 2008b). Although, there is no common standard file format that is shared by all visualization tools (not even the selection presented in this article); all support the import of plain text files, but the structure requirements of each tool vary at least slightly. Until now, most of the current packages for biological network visualization provide their own input and output file formats to load and store the networks. Naturally, one would like to take advantage of complementary strengths of different tools; however, applying different algorithms and tools to the same dataset requires additional effort to reformat files according to each specific format and strongly limits feasibility. This is one of the of the crucial bottlenecks in exploring the variety of available network analysis and visualization techniques.

Nevertheless, a number of common file formats and standard languages for storing biological information have been introduced in order to increase incompatibility between different software packages. Many current open-source standards are XML-based languages, such as BioPAX, SBML and PSI-MI. They employ hierarchical structures, and therefore allow software or user-specified levels of complexity and specificity. In the following, we will briefly introduce the most widely used data formats for used biological networks.

SBML – The Systems Biology Markup Language is intended to describe qualitative and quantitative computational models in a declarative form that can be exchanged between different software packages (Hucka *et al.*, 2015). Since it was one of the first standards for biological network models, a large number of libraries and tools for editing, parsing, and converting SBML into other formats are available. Even though it was initiated as a language to describe biochemical reactions, SBML is now a widely accepted and supported format among biological network visualization tools (Pavlopoulos *et al.*, 2008b).

BioPax – This “pathway language” is a collaborative effort to create a standard format that allows the distribution, sharing, and exchange of information between pathway databases. Therefore, it is most suitable for the description of PPIs, genetic interactions, gene regulatory, metabolic, and signaling pathways (Demir *et al.*, 2010).

GML – The Graph Modeling Language, initially proposed by M. Himsolt in 1997, is a non-XML, text file-based format for storing network data, mainly focusing on portability, simple syntax, extensibility, and flexibility (Himsolt, 1997). Its simple structure consists of a hierarchical key-value list encoding for graphs that can be annotated with arbitrary information and data structures.

KGML – The KEGG PATHWAY database (Kanehisa and Goto, 2000), was created in 1995 and contains manually curated maps for metabolic and regulatory pathways (Wrzodek *et al.*, 2011). The KEGG Markup Language (KGML) was designed, to store these KEGG graph objects in order to enable automatic visualizations of KEGG pathways. It aims to facilitate computational analysis and modeling of PPI and metabolic networks, but the usefulness of this format is often limited by the mapping of gene names or single KEGG identifier. However, current software tools such as KEGGtranslator enable easy conversion to other data formats, such as SBML (Wrzodek *et al.*, 2011).

PSI-MI – Proteomics Standards Initiative – Molecular Interaction, abbreviated as PSI-MI, aims to support the exchange, comparison and verification of PPIs (Hermjakob *et al.*, 2004).

SIF – The Simple Interaction Format is the standard data format utilized by Cytoscape and is intended to allow graphs to be easily build from a list of interactions, simply specified by source node, type of interaction, and target node. This format makes operations on networks extremely simple, for example, merging networks is achieved by simply combining lists. However, SIF does not include any layout information, and forces the visualization tool to recompute it each time the network is loaded. herefore, it is not frequently used by tools that aim for high performance on large networks.

XGMML – The extensible Graph Markup and Modeling Language, is the XML equivalent of GML, and used for graph description (Punin and Krishnamoorthy, 2001). While GML is not related to XML, XGMML is specified so as to ensure a 1:1 relationship with GML for trivial conversion between the two formats.

SBML, BioPAX, GML, KGML, and PSI-MI are the languages most compatible among the described software packages for biological network data. A detailed overview of compatibility of tools and data formats is depicted in Fig. 8. Since BioPAX and XGMML support the richest hierarchy, and have the most general approach, they can span a broad range of biomedical network data including, for instance, genetic interactions, PPI networks, and small molecules interactions, as well as gulatory and metabolic pathways (Pavlopoulos *et al.*, 2008b). In contrast, experimental data such as molecular interactions and interaction networks are most efficiently handled by PSI-MI. SBML, is best suited for its intended purpose, the description of relationships and simulations.

Network Visualization Software

With the growth of omics data and approaches, a plethora of visualization tools have been created (see, for instance, OMICTOOLS, see Relevant Websites section). We compared the functionality and applicability of 15 freely available tools from the OMICTOOLS website that can be used without programming knowledge. They are listed in Table 1. Software tools have been tested on a 64-bit PC with 8 GB of RAM running Windows 7; web applications have been tested using Mozilla Firefox Version 55.0.2.

The volume of data that needs to be visualized and analyzed in systems biology can be daunting, as the integration of different datasets as well as different types of information can lead to networks including millions of nodes and edges – a size not many

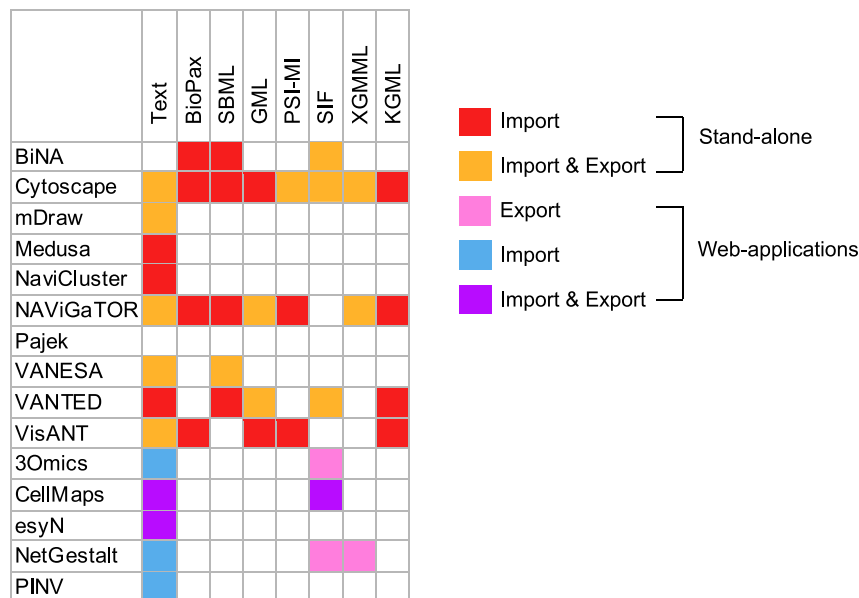


Fig. 8 Comparison of compatibility of the discussed software packages and data formats.

tools can handle. Among the list of tools tested, only three of the 15 analyzed software packages, Cytoscape, NAViGaTOR and Pajek, were able to load a network of 120,000 edges. In particular, the basic version of Pajek is designed to visualize networks composed of up to 100 million nodes/edges, but there are two other versions (Pajek-XXL and Pajek-3XL) that can handle maximum network sizes of 2 billion and 10 billion elements, respectively, making it the most scalable tool (Pavlopoulos *et al.*, 2017). For other tools, a possible workaround is not showing the entire data set. For example, querying, filtering, or zooming into the network can help focus on a specific area or part of the network, reducing the rendering load and increasing the performance of the visualization tool. Navigation and zooming is quite common, but its drawback is the difficulty in relating the area that is currently visible to the overall context. Querying and filtering are better suited to see patterns spanning the entire network, in case part of the available information is not relevant to the analysis and can consequently be hidden. Most of the tested tools use at least one technique to focus on a smaller subset of the data, as shown in Fig. 9. Abstraction is an alternative and powerful approach, where irrelevant detailed information is removed but important general structures are maintained (Zoubarev, 2009). It is achieved by creating single representatives for groups of similar elements, i.e., a meta-node representing a set of nodes, a meta-edge representing a set of edges, and a meta-network representing a subnetwork. Only five tools, NAViGaTOR, Pajek, VisANT, esyN, and NaviCluster use abstraction to reduce network complexity.

Even when working with smaller networks – of the order of thousands of elements – the first and most important task of visualization tools is to dissect the data and highlight components that are more important, for instance, peripheral or central to the analysis being performed. To this aim, most visualization tools allow the user to (1) interactively change the network layout, (2) append quantitative and qualitative network annotations – sometimes allowing users to query directly fundamental biological databases – and (3) perform network analyses that can be mapped to the network visualization, and are necessary to emphasize biologically significant nodes or areas of a network.

As system biology networks are rarely small enough to be meaningfully visualized upon being uploaded in any visualization tool, layouts are the first and most direct step to unraveling the infamous hairballs. As described in Section Layouts, many layouts and layout algorithms have been suggested, but each visualization tool only provides a selected subset, as shown in Fig. 9. Random, hierarchical and circular are the layouts most frequently present, while all the tools use at least one type of force-based algorithm, varying between spring embedded, Kamada-Kawai, or GRIP. CellMaps and esyN are the two web applications with the most versatile layout tools, while Cytoscape, Medusa, NAViGaTOR, Pajek, VANTED and VisANT offer the best range of layouts for tackling the challenge of network visualization. Among these, NAViGaTOR is the only one (at the time of writing this article) that includes hive-plots, a more recent type of layout (Krzywinski *et al.*, 2011).

As discussed in Section Biological Network Annotation, a variety of data is available in system biology that can be used to annotate biological networks, and consequently visualized to achieve a more meaningful representation. Most public data used by researchers is available through open access databases, additionally users might intend to annotate their network with data derived from their own experiments. For this reason it is important that visualization tools offer both the possibility to query public databases as well as to import the researcher's own data. As shown in Fig. 9, almost all visualization tools allow users to upload their own network and annotation data, and a majority of the tools have the ability to query one or more publicly available databases. Importantly, once any type of annotation data is included in the network, the annotation needs to be visualized in the network, and in the following we will refer to visual annotation as the visual representation (color, shape, etc.) of the annotation

	Scalability			Import		Export		Annotation			Analysis				Layouts										Visual annotation					
	Whole network	Query/filter/zoom	Abstraction	Data	Network	Data	Network	Image	Expandable plugins	Query DB	User's data	Automated visual mapping	Topology	Graph partitioning	Multiple networks	Data	Random	Planar	Grid	Tree	Hierarchical	Circular	Heatmap/Matrix	Radial/Concentric	Hive plot	3D	Colors	Size	Transparency	Shape (node)
BiNA																														
Cytoscape																#														
mDraw																											E			
Medusa																											N			
NaviCluster																														
NAViGaTOR																														
Pajek																														
VANESA										*																	N			
VANTED																														
VisANT																														
3Omics																														
CellMaps																														
esyN																														
NetGestalt																														
PINV																											N			

Fig. 9 Summary of the features of free visualization tools. Red squares are features present in stand-alone applications, while blue squares indicate features of web-applications: * = VANESA queries databases to build the network but not to annotate it. # = Cytoscape doesn't perform data analysis in its basic version, but many of its plug-ins do. E = edges; N = nodes.

values. While most of the tools offer automatic mapping of annotations to the visualization, only half of the investigated tools enable the user to choose and manage visual annotations. Among the possible visual annotations, color is the most frequently available, while only 3 tools, Cytoscape, NAViGaTOR and CellMaps (also the three tools with most visual annotation features), offer the possibility of modifying transparency.

Network analyses add valuable information that is fundamental to meaningfully visualizing biological networks, especially those of larger size. Even if network analyses can be run separately using programs like R, Python, or dedicated tools such as GraphCrunch (Kuchaiev *et al.*, 2011) or Mfinder (Milo *et al.*, 2002), and then uploaded as annotation data, the possibility of running the analyses directly in the visualization software or web application makes the process more interactive. As highlighted in Fig. 9, most visualization tools support the application of topology or graph partitioning algorithms to the network. Furthermore, we evaluated which tools support the analysis of structural features. Shortest paths are supported by BiNA, Cytoscape, NAViGaTOR, VANESA, VisANT, esyN. Additionally, NAViGaTOR performs flow analysis and cliques search. Only three tools, Cytoscape, Pajek and Medusa, are able to perform analyses across multiple networks.

In our software evaluation, we further considered the possibilities of performing analyses using network annotation data – for example, pathway enrichment or co-expression analysis. NAViGaTOR, VisANT, VANTED, CellMaps, 3Omics, and NetGestalt support some of these analyses. Cytoscape does not perform data analysis in its basic version (the one we used for comparison in this section), but a collection of 326 plugins (at the time of writing) includes a module that enables direct control of network visualization and annotation through R, makes Cytoscape a very versatile software for network analysis in system biology (Saito *et al.*, 2012). Similarly, BiNA and Pajek provide an interface to R and its network analyses functionality. Note that, often, the proper usage of layouts, layout algorithms and visual annotations, have a larger influence on information gain than the selected visualization tool. Moreover, frequently, combinations and variations of layouts are needed to achieve optimal visualization in order to gain insights into biological networks and their mechanisms.

Fig. 10 visualizes a PPI network with different layouts. While no information can be gained from the grid layout visualization, the orthogonal layout shows the presence of different groups in the network and the visual annotation of nodes with large degree (Fig. 10C) enables the identification of key proteins. Fig. 11 visualizes a phylogeny with different layouts. The circular visualization hardly indicates the relations between the different taxa. The GRIP layout spreads the different branches of the phylogeny, such that different subgroups can be identified and further visually annotated.

The comparison across both standalone and web applications revealed a number of key features. Notably, more recent visualization tools are being developed as web applications rather than standalone programs. Web implementations have greater visibility and often provide a more intuitive user interface design that can make advanced visualization more accessible to non-expert users. Moreover, web-based applications are independent of the hardware being used. Stand-alone applications, on the other hand, have some software and hardware requirements that vary across tools, but they can work without an on-line

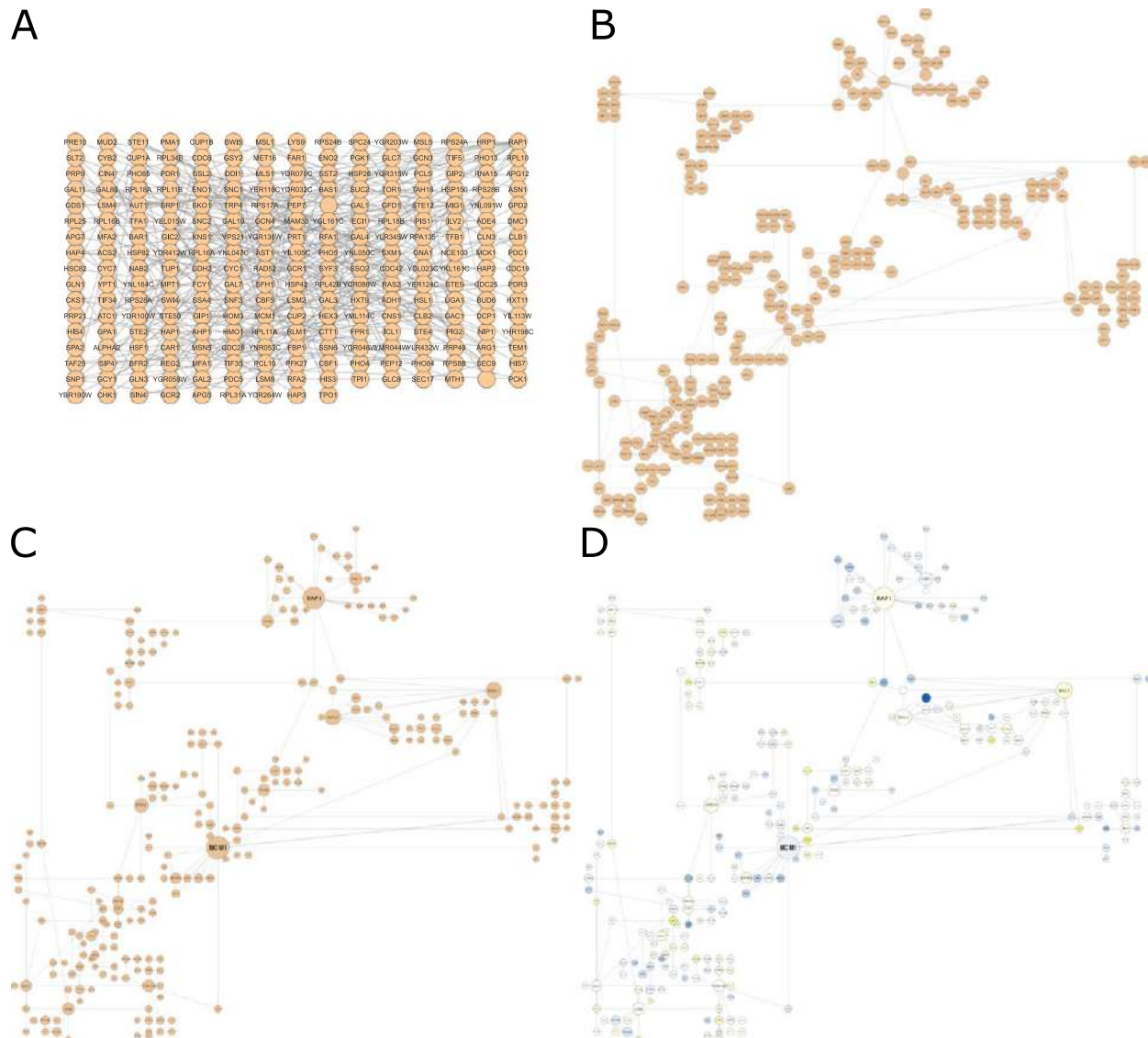


Fig. 10 Example of a PPI network, visualized in different layouts and annotations: (A) grid layout, (B) orthogonal layout, (C) node size annotation and (D) node color annotation. This figure was created by Cytoscape 3.6.0 (Shannon *et al.*, 2003) using the largest connected component of the sample yeast network provided by the tool Cytoscape. Available at: www.cytoscape.org/.

connection, do not rely on continuous support, in contrast to a web-server, and are more robust and scalable. Moreover, they ensure confidentiality, which is indispensable for delicate biomedical or clinical data.

Future Direction

Visualization is a powerful method to summarize and highlight key messages hidden in the huge amount of molecular data constantly being produced. In particular, supporting interactive visualization enables combining analysis with human insight – leading to “visual data mining.” While current tools are already more advanced and powerful than those of a decade ago, more interesting features and powerful developments are in sight for the next generation of tools. Handling of larger networks and more powerful analyses are possible with multiple threads running in parallel, including the use of multiple CPU cores and GPU-based computing that perform much faster than traditional single-CPU algorithms (Goddard *et al.*, 2017; Brown *et al.*, 2009). Even laboratories that lack proper computational facilities, the availability of cloud computing solves the problem of large network analyses. Faster rendering will be achieved with the development of improved graphic cards.

Simultaneously, a potential solution for the visualization of large scale networks is the transition from two-dimensional to three-dimensional representations. Over the last decade a number of software tools have developed modules that depict networks in a 3D universe, for instance, Pajek, a previous version of Navigator, Arena3D (Pavlopoulos *et al.*, 2008a), and the Cytoscape

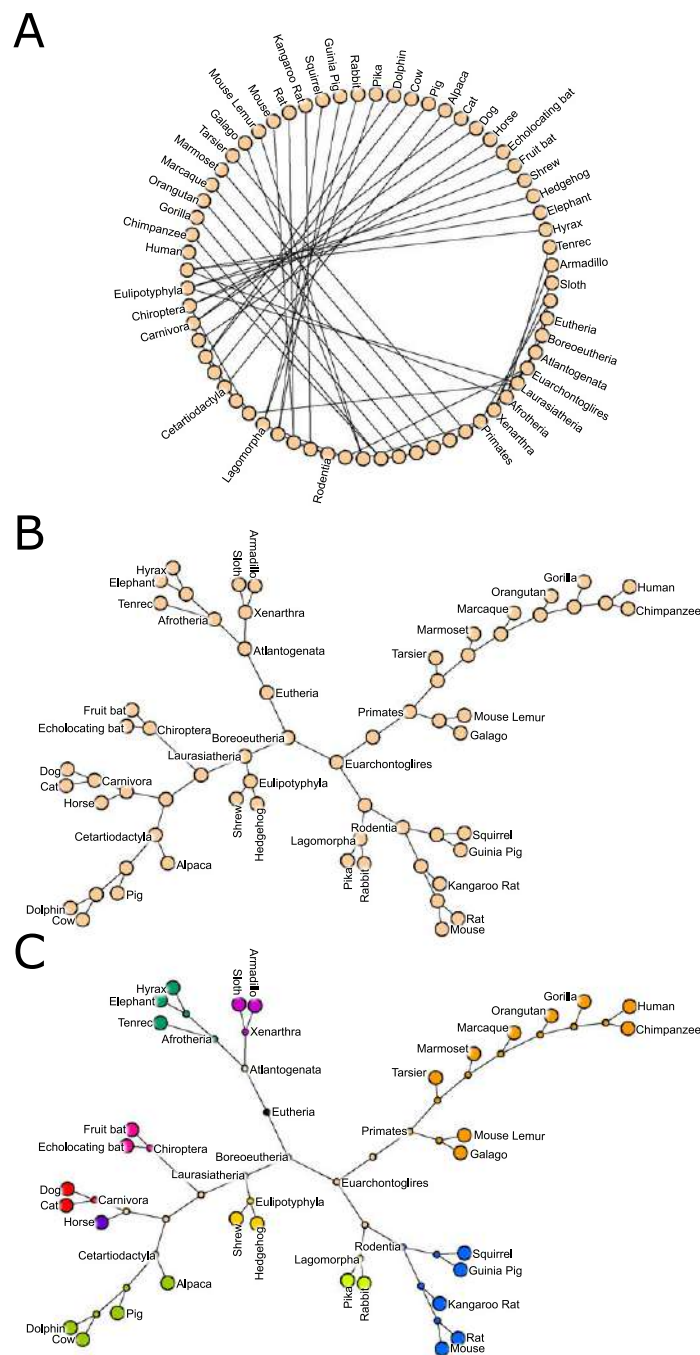


Fig. 11 Example of a phylogeny, visualized in different layouts and annotations: (A) circular layout, (B) tree layout optimized by GRIP, (C) node size and color annotation. This figure was created by NAViGaTOR (Brown *et al.*, 2009). Available at: <http://ophid.utoronto.ca/navigator/>, exported in SVG, and merged in Inkscape. Available at: <https://inkscape.org>. The corresponding phylogeny was taken from Song *et al.* (2012).

plugin 3DScape (Wang *et al.*, 2013). However, the presentation of a three-dimensional graph on a regular two-dimensional screen, does enable better visual exploration, but does not exploit its full potential. In contrast, different visualization technologies – including virtual reality – could change completely the way we approach network visualization. For example, stereoscopy (in which separate images are projected to each eye, to mimic 3D stereoscopic vision) can be helpful for exploring large, complex networks, especially in combination with rotation or user motion cues; Similar comments apply to immersive visualization environments (where the user is “immersed” inside the image). While 3D and immersive technology has been available for some time, only recently implementations for biomedical network analysis, like iCAVE (Liluashvili *et al.*, 2017) or VANTED (Sommer and Schreiber, 2017), are starting to appear, and with the advent of more affordable virtual reality devices the future of network visualization looks very promising (and three dimensional).

At the present time, most of the visualization tools focus on single static networks, where knowledge collected from different conditions is gathered in a single graph (Blonder *et al.*, 2012). However, biological entities go through topological and dynamic changes, and many interactions exist only in a specific space or time. For example, specific interactions that occur in different tissues, at different times of the day, and at different stages of life, are topological changes to the network, where as propagation of a signal or spread of a disease are flow changes. Both types of dynamic changes can be captured with respect to time, for instance, following the distribution of shortest paths in a progressing sequence of networks can uncover potential molecular mechanisms hidden in the differences among such networks (Wong *et al.*, 2015). Future visualization tools will have to be able to capture, analyze and visualize these temporal network changes.

Closing Remarks

Although many network visualization tools are available today, we still face many challenges. The continuously increasing amounts of data in health sciences still pose a bottleneck for the visualization and manipulation of the resulting large-scale networks, which may comprise millions of nodes and edges (Pavlopoulos *et al.*, 2017). The computer science community has implemented a variety of efficient and effective algorithms and libraries for graph analysis and static visualization of large networks, from programing environments and libraries, such as R or Python, to more dedicated software packages and tools, such as, the Stanford Network Analysis Project (SNAP) and GraphViz. An important characteristic of this family of packages and tools is the ease with which we can combine and integrate various algorithms into new workflows and packages.

The development of flexible software tools with user-friendly and interactive graphical interfaces, that are able to handle large graphs, still remains a challenge when applied to biomedical research. The direct comparison of different network visualization tools shows that each has specific strengths and weaknesses, evolving from varying objectives and intended purposes. For instance, alternative network visualizations approaches such as Circos (Sumazin *et al.*, 2011) and hive plots (Krzywinski *et al.*, 2011) have been designed to solve the hairball effect that limits the understandability and interpretability of biological networks, but are very specific in their intended usage. Therefore, a good approach would combine the complementary advantages of different tools and algorithms.

However, in contrast to environments and libraries, like R and SNAP, the combination of the strengths provided by different visualization tools is strongly limited by the lack of compatibilities of data formats. Although a number of standardized formats for network data have been formalized, our experience showed that none have a general compatibility for a majority of tools. Even in the best cases, where import/export functionalities are implemented, node and edge attributes often cannot be transferred. A good future strategy for the community developing tools for biomedical networks would be to strongly facilitate the interchangeable usage of these tools by the broader support of standard file formats and accurate conversions between them, while simultaneously preserving a maximal amount of annotation and layout information.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein Localization. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Quantification of Proteins from Proteomic Analysis. Two Decades of Biological Pathway Databases: Results and Challenges. Visualization of Biological Pathways

References

- Agapito, G., Guzzi, P.H., Cannataro, M., 2013. Visualization of protein interaction networks: Problems and solutions. *BMC Bioinform.* 14 (1), S1.
- Alanis-Lobato, G., Andrade-Navarro, M.A., Schaefer, M.H., 2016. Hippie v2. 0: Enhancing meaningfulness and reliability of protein-protein interaction networks. *Nucleic Acids Res.* 45 (D1), D408–D414.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25 (1), 25–29.
- Baryshnikova, A., 2016. Systematic functional annotation and visualization of biological networks. *Cell Syst.* 2 (6), 412–421.
- Basha, O., Barshir, R., Sharon, M., *et al.*, 2017. The tissuenet v. 2 database: A quantitative view of protein-protein interactions across human tissues. *Nucleic Acids Res.* 45 (D1), D427–D431.
- Batagelj, V., Mrvar, A., 2001. Pajekanalysis and visualization of large networks. In: *International Symposium on Graph Drawing*. Springer, pp. 477–478.
- Bean, D.M., Heimbach, J., Ficorella, L., *et al.*, 2014. esyn: Network building, sharing and publishing. *PLOS ONE* 9 (9), 1–5.
- Blonder, B., Wey, T.W., Dornhaus, A., James, R., Sih, A., 2012. Temporal dynamics and network analysis. *Methods Ecol. Evol.* 3 (6), 958–972.
- Boc, A., Diallo, A.B., Makarenkov, V., 2012. T-REX: A web server for inferring, validating and visualizing phylogenetic trees and networks. *Nucleic Acids Res.* 40 (W1), W573–W579.
- Breitkreutz, B.J., Stark, C., Reguly, T., *et al.*, 2008. The BioGRID interaction database: 2008 update. *Nucleic Acids Res.* 36 (Database issue), D637–D640.
- Brown, K.R., Otasek, D., Ali, M., *et al.*, 2009. Navigator: Network analysis, visualization and graphing toronto. *Bioinformatics* 25 (24), 3327–3329.
- Caspi, R., Altman, T., Billington, R., *et al.*, 2014. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Res.* 42 (D1), D459–D471.
- Croft, D., Mundo, A.F., Haw, R., *et al.*, 2013. The reactome pathway knowledgebase. *Nucleic Acids Res.* 42 (D1), D472–D477.
- Cusick, M.E., Klotz, N., Vidal, M., Hill, D.E., 2005. Interactome: Gateway into systems biology. *Hum. Mol. Genet.* 14 (Suppl. 2), R171–R181.
- Davidson, E., Levin, M., 2005. Gene regulatory networks.

- Demir, E., Cary, M.P., Paley, S., *et al.*, 2010. The biopax community standard for pathway data sharing. *Nat. Biotechnol.* 28 (9), 935–942.
- Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G., 1994. Algorithms for drawing graphs: An annotated bibliography. *Comput. Geom.* 4 (5), 235–282.
- Dinasarapu, A.R., Saunders, B., Ozerlat, I., Azam, K., Subramaniam, S., 2011. Signaling gateway molecule pages a data model perspective. *Bioinformatics* 27 (12), 1736–1738.
- Djebbari, A., Ali, M., Otasek, D., *et al.*, 2011. Navigator: Large scalable and interactive navigation and analysis of large graphs. *Internet Math.* 7 (4), 314–347.
- Edgar, R., Domrachev, M., Lash, A.E., 2002. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Res.* 30 (1), 207–210.
- Fortney, K., Kotlyar, M., Jurisica, I., 2010. Inferring the functions of longevity genes with modular subnetwork biomarkers of *Caenorhabditis elegans* aging. *Genome Biol.* 11 (2), R13.
- Fruchterman, T.M.J., Reingold, E.M., 1991. Graph drawing by force-directed placement. *Softw.: Pract. Exp.* 21 (11), 1129–1164.
- Fung, D.C., Wilkins, M.R., Hart, D., Hong, S.-H., 2010. Using the clustered circular layout as an informative method for visualizing protein-protein interaction networks. *Proteomics* 10 (14), 2723–2727.
- Gajer, P., Kobourov, S.G., 2002. Grip: Graph drawing with intelligent placement. *J. Graph Algorithms Appl.* 6 (3), 203–224.
- Goddard, T.D., Huang, C.C., Meng, E.C., *et al.*, 2017. Ucsf chimerax: Meeting modern challenges in visualization and analysis. *Protein Sci.* doi:10.1002/pro.3235.
- Gove, R., Gramsky, N., Kirby, R., *et al.*, 2011. Netvisia: Heat map & matrix visualization of dynamic social network statistics & content. In: *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Proceedings of the Third International Conference on Social Computing (SocialCom)*, pp. 19–26.
- Gremse, M., Chang, A., Schomburg, I., *et al.*, 2010. The brenda tissue ontology (bto): The first all-integrating ontology of all organisms for enzyme sources. *Nucleic Acids Res.* 39 (Suppl. 1), D507–D513.
- Hatzimanikatis, V., Li, C., Ionita, J.A., Broadbelt, L.J., 2004. Metabolic networks: Enzyme function and metabolite structure. *Curr. Opin. Struct. Biol.* 14 (3), 300–306.
- Herman, I., Melancon, G., Marshall, M.S., 2000. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. Vis. Comput. Gr.* 6 (1), 24–43.
- Hermjakob, H., Montecchi-Palazzi, L., Bader, G., *et al.*, 2004. The hupo psi's molecular interaction format: community standard for the representation of protein interaction data. *Nat. Biotechnol.* 22 (2), 177–183.
- Himsolt, M., 1997. Gml: A portable graph file format. Available at: <http://www.fmi.uni-passau.de/graphlet/gml/gml-tr.html>, Universitat Passau.
- Hooper, S.D., Bork, P., 2005. Medusa: A simple tool for interaction graph analysis. *Bioinformatics* 21 (24), 4432–4433.
- Hu, Z., Mellor, J., Wu, J., DeLisi, C., 2004. Visant: An online visualization and analysis tool for biological interaction data. *BMC Bioinform.* 5 (1), 17.
- Hucka, M., Bergmann, F.T., Dräger, A., *et al.*, 2015. Systems biology markup language (sbml) level 2 version 5: Structures and facilities for model definitions. *J. Integr. Bioinform.* 12 (2), 731–901.
- Ings, T.C., Montoya, J.M., Bascompte, J., *et al.*, 2009. Ecological networks-beyond food webs. *J. Anim. Ecol.* 78 (1), 253–269.
- Junker, B.H., Klukas, C., Schreiber, F., 2006. Vanted: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinform.* 7 (1), 109.
- Kamada, T., Kawai, S., 1989. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* 31 (1), 7–15.
- Kanehisa, M., Goto, S., 2000. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 28 (1), 27–30.
- Kibbe, W.A., Arze, C., Felix, V., *et al.*, 2014. Disease ontology 2015 update: An expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic Acids Res.* 43 (D1), D1071–D1078.
- Koh, G.C., Porras, P., Aranda, B., Hermjakob, H., Orchard, S.E., 2012. Analyzing protein-protein interaction networks. *J. Proteome Res.* 11 (4), 2014–2031.
- Kolesnikov, N., Hastings, E., Keays, M., *et al.*, 2014. Arrayexpress update simplifying data submissions. *Nucleic Acids Res.* 43 (D1), D1113–D1116.
- Kotlyar, M., Pastrello, C., Sheahan, N., Jurisica, I., 2016. Integrated interactions database: Tissue-specific view of the human and model organism interactomes. *Nucleic Acids Res.* 44 (D1), D536.
- Krzywinski, M., Birol, I., Jones, S.J., Marra, M.A., 2011. Hive plots – Rational approach to visualizing networks. *Briefings Bioinform.* 13 (5), 627–644.
- Krzywinski, M., Schein, J., Birol, I., *et al.*, 2009. Circos: An information aesthetic for comparative genomics. *Genome Res.* 19 (9), 1639–1645.
- Kuchaiev, O., Stevanović, A., Hayes, W., Pržulj, N., 2011. GraphCrunch 2: Software tool for network modeling, alignment and clustering. *BMC Bioinform.* 12 (1), 24.
- Küntzer, J., Backes, C., Blum, T., *et al.*, 2007. Bndb – The biochemical network database. *BMC Bioinform.* 8 (1), 367.
- Kuo, T.-C., Tian, T.-F., Tseng, Y.J., 2013. 3omics: A web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data. *BMC Syst. Biol.* 7 (1), 64.
- Li, W., Kurata, H., 2005. A grid layout algorithm for automatic drawing of biochemical networks. *Bioinformatics* 21 (9), 2036–2042.
- Li, Y.-H., Tavallaei, G., Tokar, T., *et al.*, 2016. Identification of synovial fluid microRNA signature in knee osteoarthritis: Differentiating early- and late-stage knee osteoarthritis. *Osteoarthritis Cartilage* 24 (9), 1577–1586.
- Lilushvili, V., Kalayci, S., Fluder, E., *et al.*, 2017. icave: An open source tool for visualizing biomolecular networks in 3d, stereoscopic 3d and immersive 3d. *Giga Sci.* 6 (8), 1–13.
- Liu, W., Wang, J., Wang, T., Xie, H., 2014. Construction and analyses of human large-scale tissue specific networks. *PLOS ONE* 9 (12), e115074.
- Lopes, T.J., Schaefer, M., Shoemaker, J., *et al.*, 2011. Tissue-specific subnetworks and characteristics of publicly available human protein interaction databases. *Bioinformatics* 27 (17), 2414–2421.
- Mäkinen, E., 1988. On circular layouts. *Int. J. Comput. Math.* 24 (1), 29–37.
- Masuda, S., Nakajima, K., Kashiwabara, T., Fujisawa, T., 1990. Crossing minimization in linear embeddings of graphs. *IEEE Trans. Comput.* 39 (1), 124–127.
- Micale, G., Ferro, A., Pulvirenti, A., Giugno, R., 2016. Spectra: An integrated knowledge base for comparing tissue and tumor-specific ppi networks in human. *Front Bioeng. Biotechnol.* 59.
- Milo, R., Shen-Orr, S., Itzkovitz, S., *et al.*, 2002. Network motifs: Simple building blocks of complex networks. *Science* 298 (5594), 824–827.
- Okamura, Y., Aoki, Y., Obayashi, T., *et al.*, 2014. Coexpresdb in 2015: Coexpression database for animal species by dna-microarray and rnaseq-based expression data with multiple quality assessment systems. *Nucleic Acids Res.* 43 (D1), D82–D86.
- Otasek, D., Pastrello, C., Jurisica, I., 2012. Scalable, integrative analysis and visualization of protein interactions. In: Cai, W., Hong, H. (Eds.), *Protein-Protein Interactions-Computational and Experimental Tools*. InTech.
- Pavlopoulos, G.A., Malliarakis, D., Papanikolaou, N., *et al.*, 2015. Visualizing genome and systems biology: Technologies, tools, implementation techniques and trends, past, present and future. *GigaScience* 4 (1), 38.
- Pavlopoulos, G.A., O'Donoghue, S.I., Satagopam, V.P., *et al.*, 2008a. Arena3D: Visualization of biological networks in 3D. *BMC Syst. Biol.* 2 (1), 104.
- Pavlopoulos, G.A., Paez-Espino, D., Kyripides, N.C., Iliopoulos, I., 2017. Empirical comparison of visualization tools for larger-scale network analysis. *Adv. Bioinform.* 2017.
- Pavlopoulos, G.A., Wegener, A.-L., Schneider, R., 2008b. A survey of visualization tools for biological network analysis. *Biodata Mining* 1 (1), 12.
- Praneenararat, T., Takagi, T., Iwasaki, W., 2011. Interactive, multiscale navigation of large and complicated biological networks. *Bioinformatics* 27 (8), 1121–1127.
- Proß, S., Bachmann, B., Janowski, S.J., Hofstadt, R., 2012. A new object-oriented petri net simulation environment based on modelica. In: *Proceedings of the Winter Simulation Conference*, p. 300.
- Punin, J., Krishnamoorthy, M., 2001. Xgmml (extensible graph markup and modeling language) 1.0 draft specification.
- Saito, R., Smoot, M.E., Ono, K., *et al.*, 2012. A travel guide to cytoscape plugins. *Nat. Methods* 9 (11), 1069–1076.
- Salazar, G.A., Meintjes, A., Mazandu, G.K., *et al.*, 2014. A web-based protein interaction network visualizer. *BMC Bioinform.* 15 (1), 129.
- Schulz, C., Nocaj, A., Goertler, J., *et al.*, 2017. Probabilistic graph layout for uncertain network visualization. *IEEE Trans. Vis. Comput. Graphics* 23 (1), 531–540.
- Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13 (11), 2498–2504.
- Shi, Z., Wang, J., Zhang, B., 2013. Netgestalt: Integrating multidimensional omics data over biological networks. *Nat. Methods* 10 (7), 597–598.
- Smith, B., Ashburner, M., Rosse, C., *et al.*, 2007. The obo foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* 25 (11), 1251.

- Smith, R.J., Koobatian, M.T., Shahini, A., Swartz, D.D., Andreadis, S.T., 2015. Capture of endothelial cells under flow using immobilized vascular endothelial growth factor. *Biomaterials* 51, 303–312.
- Sommer, B., Schreiber, F., 2017. Integration and virtual reality exploration of biomedical data with cmpi and vanted. *Inf. Technol.* 59 (4), 181–190.
- Song, S., Liu, L., Edwards, S.V., Wu, S., 2012. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proc. Natl. Acad. Sci.* 109 (37), 14942–14947.
- Sumazin, P., Yang, X., Chiu, H.-S., *et al.*, 2011. An extensive microRNA-mediated network of rna-rna interactions regulates established oncogenic pathways in glioblastoma. *Cell* 147 (2), 370–381.
- Szklarczyk, D., Morris, J.H., Cook, H., *et al.*, 2016. The string database in 2017: Quality-controlled protein-880 protein association networks, made broadly accessible. *Nucleic Acids Res.* 45 (D1), D362–D368.
- Uhlén, M., Fagerberg, L., Hallström, B.M., *et al.*, 2015. Tissue-based map of the human proteome. *Science* 347 (6220), 1260419.
- Veres, D.V., Gyurkó, D.M., Thaler, B., *et al.*, 2015. Compbi: A cellular compartment-specific database for protein-protein interaction network analysis. *Nucleic Acids Res.* 43 (D1), D485–D493.
- Wang, Q., Tang, B., Song, L., *et al.*, 2013. 3DScapeCS: Application of three dimensional, parallel, dynamic network visualization in Cytoscape. *BMC Bioinform.* 14 (1), 322.
- Wilgenbusch, J.C., Huang, W., Gallivan, K.A., 2017. Visualizing phylogenetic tree landscapes. *BMC Bioinform.* 18 (1), 85.
- Wong, S.W., Cercone, N., Jurisica, I., 2015. Comparative network analysis via differential graphlet communities. *Proteomics* 15 (2-3), 608–617.
- Wrzodek, C., Driäger, A., Zell, A., 2011. Keggtranslator: Visualizing and converting the kegg pathway database to various formats. *Bioinformatics* 27 (16), 2314–2315.
- Wu, E., Nance, T., Montgomery, S.B., 2013. Spliceplot: A utility for visualizing splicing quantitative trait loci. *Bioinformatics* 30 (7), 1025–1026.
- Zeng, Y., Zhang, L., Zhu, W., *et al.*, 2016. Quantitative proteomics and integrative network analysis identified novel genes and pathways related to osteoporosis. *J. Proteomics* 142, 45–52.
- Zhang, Y., Xie, J., Yang, J., *et al.*, 2017. Qubic: A bioconductor package for qualitative biclustering analysis of gene co-expression data. *Bioinformatics* 33 (3), 450–452.
- Zhong, Q., Simonis, N., Li, Q.-R., *et al.*, 2009. Edgetic perturbation models of human inherited disorders. *Mol. Syst. Biol.* 5 (1), 321.
- Zhu, L., Malatras, A., Thorley, M., *et al.*, 2015. Cellwhere: Graphical display of interaction networks organized on subcellular localizations. *Nucleic Acids Res.* 43 (W1), W571–W575.
- Zoubarev, A., 2009. Tools for visual analysis of biological networks.

Further Reading

- Fruchterman, M.J.T., Reingold, M.E., 1991. Graph drawing by force directed placement. *Softw. Pract. Exper.* 21, 1129–1164.
- Peter, E., 1984. A heuristic for graph drawing. *Congressus Numerantium* 42, 149–160.

Relevant Websites

- <https://www.ebi.ac.uk/arrayexpress/>
ArrayExpress.
- <http://coexpresdb.jp/>
COXPRESdb.
- <http://geneontology.org>
Gene Ontology Consortium.
- bioinformatics.med.uoc.gr:3838/NAP/
NAP.
- <https://www.ncbi.nlm.nih.gov/geo/>
NCBI.
- omictools.com/network-visualization-category
OMICTOOLS.
- <http://www.proteinatlas.org/>
The Human Protein Atlas.
- trex.uqam.ca
UQAM.

Biographical Sketch

Anne-Christin Hauschild is a postdoctoral researcher supervised by Igor Jurisica at the Princes Margaret Cancer Center and University of Toronto. She finished her PhD in 2016 on Computational Methods for Breath Metabolomics in Clinical Diagnostics, which was funded by a Scholarship of the International Max-Planck Research School. She is involved in the Mapping Cancer Markers (MCM) project to systematically survey the landscape of potential gene signature patterns for multiple cancers, such as lung and ovarian. Her current research interests further include translational medicine particularly in arthritis, focusing on pre - versus - post surgical analysis of clinical as well as genetic data on a large number of osteoarthritis patients. During her scientific career she published 6 first and 10 co-author papers in international journals and was invited 7 times to give oral presentations at major bioinformatics conferences such as GCB, ECCB, GLBIO.

Chiara Pastrello is a research associate at University Health Network, Toronto. She is interested in cancer genetics and epigenetics, that she formalized in her medical genetics residency – focusing on the prediction of pathogenicity of unclassified variants in mismatch repair genes. She is further interested in cancer systems biology, specializing in network analysis, and the curation of expression data for multiple databases maintained at Prof. Jurisica lab. Her current research interests include the study of microRNAs in cancer signaling to distant metastasis, as well as the application of network analyses to study molecules and pathways playing a central role in cancer.

Andrea E.M. Rossos is a third-year undergraduate student in a Bachelor of Science program in Honours Biology at McMaster University. She was a Princess Margaret Summer Student who has worked in the Jurisica Lab for summer 2016 and summer 2017 as a recipient of the Ian Lawson Van Toch Internship Award

in Cancer Informatics. Andrea has been involved with data curation and development for projects such as microRNA Data Integration Portal (mirDIP), Cancer Data Integration Portal (CDIP) and Arthritis Data Integration Portal (ADIP). Andrea has also been involved with projects looking at the properties and diversity of biomolecule (miRNA, protein, etc.) databases. Andreas current research interests are the prognostic and diagnostic roles of microRNAs and other biomolecules in cancer and other diseases.

I. Jurisica is Tier I Canada Research Chair in Integrative Cancer Informatics, Senior Scientist at Princess Margaret Cancer Center, Professor at University of Toronto and Visiting Scientist at IBM CAS. He is also an Adjunct Professor at the School of Computing, Pathology and Molecular Medicine at Queen's University, Computer Science at York University, affiliate scientist at the Institute of Neuroimmunology, Slovak Academy of Sciences, and an Honorary Professor at Shanghai Jiao Tong University. Since 2015, he has also served as Chief Scientist at the Creative Destruction Lab, Rotman School of Management. He has published extensively on data mining, visualization and cancer informatics, including multiple papers in Science, Nature, Nature Medicine, Nature Methods, J Clinical Oncology, and has over 11,167 citations over the last 5 years. He has been included in Thomson Reuters 2016, 2015 & 2014 list of Highly Cited researchers, and The World's Most Influential Scientific Minds: 2015 & 2014 Reports.

Cluster Analysis of Biological Networks

Asuda Sharma, Hesham Ali, and Dario Ghersi, University of Nebraska at Omaha, Omaha, NE, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological data is often represented as networks, as in the case of protein-protein interactions and metabolic pathways. Modeling, analyzing, and visualizing networks can help make sense of large volumes of data generated by high-throughput experiments. However, due to their size and complex structure, biological networks can be difficult to interpret without further processing. Cluster analysis is a widely used approach to extract meaningful information from biological networks, and works by grouping objects such that the similarity within a group is higher than the similarity between groups (Tan, 2006). Data clustering makes it easier to analyze and interpret the information contained in data (Xu and Wunsch, 2010). For example, clustering biological networks can help identify homologous proteins in different organisms or co-expressed genes.

Cluster analysis is based on two principles: (1) *homogeneity* (i.e., elements within the same cluster are similar to each other); and (2) *separation* (i.e., elements in different clusters are different from each other) (Jain et al., 1999). Clustering is often the first step in a data analytical pipeline, and is a well-known NP-hard problem. However, over the years a lot of heuristic algorithms have been developed which converge to a local optimum. Clustering algorithms are methods used to cluster the data into different groups based on a similarity measure, and as such they are not data-type specific. For example, a clustering algorithm that is used for clustering protein-protein interaction data can also be used to cluster social networking data. The selection of a clustering algorithm depends on the data characteristics along with the intended use of the results. It is a technique which must be run on a trial and error basis to find out the algorithm that best facilitates data interpretation. Data preprocessing and parameter optimization play a big role in the selection of a clustering algorithm.

Classification and *clustering* are often confused with each other but these terms have different working principles. Data classification (Fig. 1) is part of supervised learning where a collection of labeled data known as training data consisting of pairs of input cases and the desired outputs are used to construct a model for prediction of target output where the output values are unknown (Inza et al., 2010). In supervised learning, inputs are assumed to be the cause of outputs (Valpola, 2000).

In contrast, clustering (Fig. 2) is unsupervised learning and its goal is to partition the data samples into groups. Data labels are not available for unsupervised clustering (Inza et al., 2010). In unsupervised learning all the observations are assumed to be the result of latent variables (Valpola, 2000).

Some fields like community ecology use the terms classification and clustering interchangeably. However, in the field of life sciences, these two terms represent different concepts; classification of a biological network is different from clustering of a network.

Why So Many Clustering Algorithms?

There is no single definition of what a cluster is. Over the years, this has led to the development of many different clustering algorithms, creating a pressing need for objective benchmarks. Why are there so many different clustering algorithms and why is it

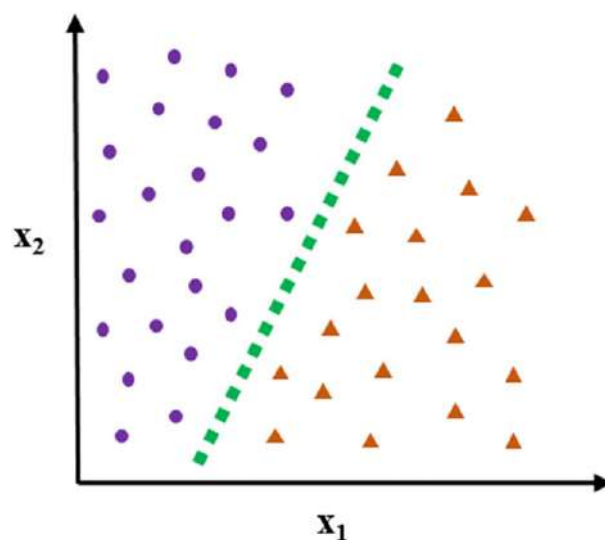


Fig. 1 Data classification.

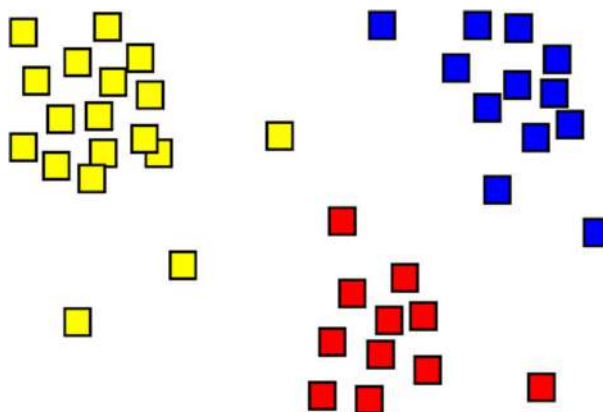


Fig. 2 An example of clustering in two dimensions (https://en.wikipedia.org/wiki/Cluster_analysis#/media/File:Cluster-2.svg).

difficult to develop comparison benchmarks? The answer lies in the diversity of models and the induction principles underlying the different algorithms. Inductive principles are formal definitions of a cluster (Estivill-Castro, 2002). As the definition changes, the inductive principle changes, which in turn leads to new clustering algorithms. Further, for a single model and inductive principle, there are many algorithms that can be used to generate the optimized results. The differences in inductive principles are often the results of assumptions made by the researchers and the specific problem at hand. Given any data set, the first step is a hypothesis on how to group the data. This hypothesis leads to the development of a model which is then used for clustering the instances of data which are not seen. The mathematical formulation of the inductive principles is used to discriminate different hypotheses. It is also known as the clustering criterion. Therefore, models are the structures used to represent different clusters and the inductive principle is used to select the model that best fits a given dataset (Estivill-Castro, 2002).

Various clustering algorithms such as K-means, affinity propagation, spectral clustering, and hierarchical clustering have been developed to cluster different types of data generated from different sources. For the same data set, different clustering methods will generally give different results. There is no one single tool which can give the best clusters for a given data set. Clustering methods were at first divided into two main groups: hierarchical and partitioning methods (Fraley and Raftery, 1998).

Drawbacks of Traditional Clustering Algorithms

High throughput technologies along with advancements in computational prediction tools have resulted in many large-scale biological networks. Traditional clustering algorithms such as hierarchical, centroid-based, and density-based methods work well on biological networks of moderate size data, but these algorithms in some cases take too long to produce the clusters or fail altogether. This is particularly prevalent when the input network size is very large, such as functional networks for higher organisms (Jiang and Singh, 2010). In recent years, with the notion of biological networks already established, the trend has shifted from simply constructing biological networks to analyzing existing networks and finding out how the objects in the networks interact with each other. Clustering is the most common approach for network analysis and is used by many researchers to study functional modules and protein complexes and their functions.

The main problem with traditional clustering algorithms is that they were developed long before the recent advances in bioinformatics and they are for the most part domain agnostic. Hence, in many cases, they get confounded by some of the characteristics specifically associated with biological data. For example, in the K-means clustering approach, which is one of the oldest and most widely-used methods for data clustering, it is required to identify the number of clusters “K” in advance, which is not always desirable in dealing with biological networks. Another common clustering approach is hierarchical clustering which does not leave any data element (such as gene or protein) unclustered, which is an unnecessary restriction in the case of biological networks. Also, the outcome of almost all traditional clustering methods are strongly driven by a rather small number of features in the input elements, which is again not the case in dealing with biological data with many attributes (Huttenhower *et al.*, 2007).

Many clustering algorithms, especially graph-based and biclustering algorithms have been developed to overcome the limitations of traditional methods. Some of these algorithms, such as SAMBA, CLICK, and spectral clustering are developed primarily to deal with biological data. The next section describes the traditional and the newly-developed clustering approaches, highlighting some of the most widely used clustering methods for each category.

Types of Clustering Methods

The majority of the existing clustering algorithms can be classified into six categories based on the working principle of each cluster model.

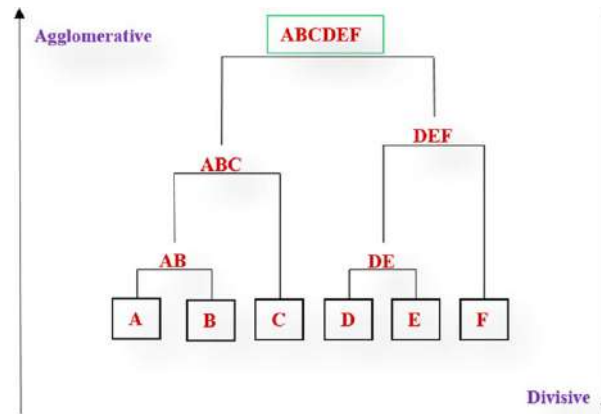


Fig. 3 Traditional representation of hierarchical clustering.

Connectivity-Based Clustering

More commonly known as hierarchical clustering, connectivity-based clustering is based on the idea that nearby objects are more closely related than objects that are farther away from each other. Hierarchical clustering comprises several algorithms that form clusters by connecting objects based on their distance from each other. Different clusters are formed in a nested (or hierarchical) fashion at different distances, and the clusters are visualized as a dendrogram. These algorithms build a hierarchy of clusters which merge with each other at different distance levels (Fig. 3).

Along with a suitable distance measure between data points, the user must select a linkage criterion to define the distance between sets of data points. Some examples of linkage criteria include Unweighted Pair Group Method with Arithmetic Mean (UPGMA), single-linkage (which uses the minimum distance between objects), and complete-linkage (which uses the maximum distance between objects).

Hierarchical clustering algorithms may either operate in an agglomerative or divisive fashion. Agglomerative clustering involves grouping single elements into clusters, whereas divisive clustering involves dividing a dataset into different partitions or groups (Zhong, 2013). Hierarchical clustering algorithms do not return “ready-to-use” clusters. Instead, the user must cut the dendrogram at a chosen distance threshold in order to obtain non-overlapping clusters.

Centroid-Based Clustering

Clusters in centroid-based algorithms are based on the concept of a central vector, which may or may not be part of the dataset. These algorithms are also based on the distance between objects. Centroid-based clustering optimizations are computationally intensive; therefore, most centroid-based algorithms return approximate solutions, generated by local optimization. The most famous example of this class of algorithms is *k-means* (Fig. 4). It is one of the most widely used iterative algorithms and its appeal is ease of implementation and good convergence speed. The *k-means* algorithm seeks to minimize the squared distances between objects within a cluster, in an iterative manner. One drawback of *k-means* algorithm is that the number of clusters must be specified in advance by the user.

The general principle behind the *k-means* algorithm can be summarized by the following steps (Charrad *et al.*, 2014):

1. Select ‘*k*’ (number of desired clusters), then randomly assign *k* data points as cluster centers.
2. Temporary clusters are created by associating each observation with the nearest center.
3. New cluster centers are formed by calculating the gravity centers of the temporary clusters formed in step 2.
4. The observations are then reallocated to the clusters with the closest centers.
5. The procedure is iterated until the algorithm converges.

Distribution-Based Clustering

Distribution-based clustering methods are based on probability distribution models. Clusters are defined based on ‘same distribution’. The advantage of this type of algorithms is that objects are randomly sampled from a distribution. These methods are the perfect clustering approaches in theory but they suffer from overfitting, where a noise or error is modeled rather than the true relationship. The problem of overfitting can be corrected by using constraints, adding to the model complexity.

Gaussian mixture models that use the Expectation-maximization (EM) algorithm (Fig. 5) are one of the most widely used distribution-based clustering methods. A fixed number of randomly initialized Gaussian distributions are used to model the

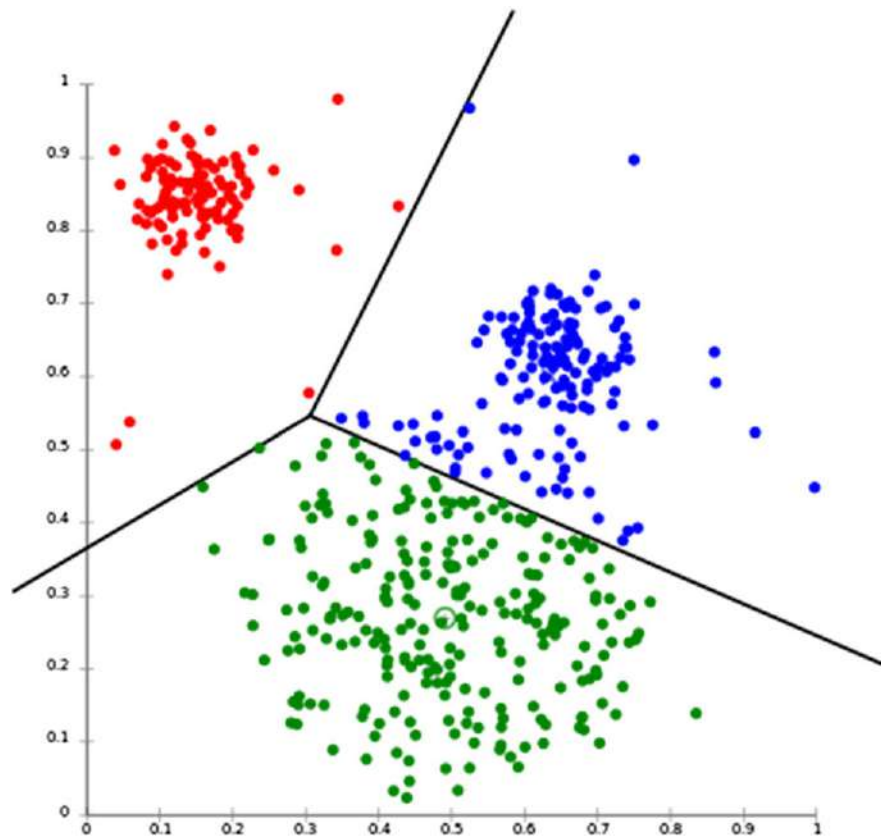


Fig. 4 An example of k-means clustering (https://en.wikipedia.org/wiki/Cluster_analysis#/media/File:Cluster-2.svg).

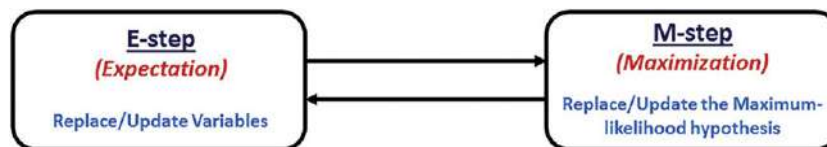


Fig. 5 The EM technique.

dataset. The goal of the EM algorithm is to estimate the mean and the standard deviation of the clusters and to maximize the likelihood of the distribution of the data (Fraley and Raftery, 1998).

Two steps are performed iteratively until the model converges to calculate the maximum likelihood, even in the presence of missing data (Fraley and Raftery, 1998):

- i) E-Step (Expectation step): The value of each of the missing variables is calculated using the existing data, the current estimates and the current hypothesis. The old values are then replaced with these new values.
- ii) M-Step (Maximization step): The new values calculated in the E-Step are assumed to be the correct values. Maximum-likelihood hypothesis is calculated based on these new values and the old hypothesis is replaced with this new hypothesis.

The two steps are performed repeatedly until the model converges and each observation belongs to a certain cluster with a definite probability.

Density-Based Clustering

The assumption for these types of methods is that each point specific to a cluster belongs to a probability distribution (Banfield and Raftery, 1993). The main goal of the density-based models is the identification of clusters along with the distribution parameters. The clusters are recognized because of their high density as compared to the points outside the cluster (Ester, Kriegel *et al.*, 1996). Objects lying in the 'noisy areas' have lower density as compared to the cluster density. DBSCAN or Density-based spatial clustering of applications with noise is the most commonly used density-based clustering algorithm. It is based on the

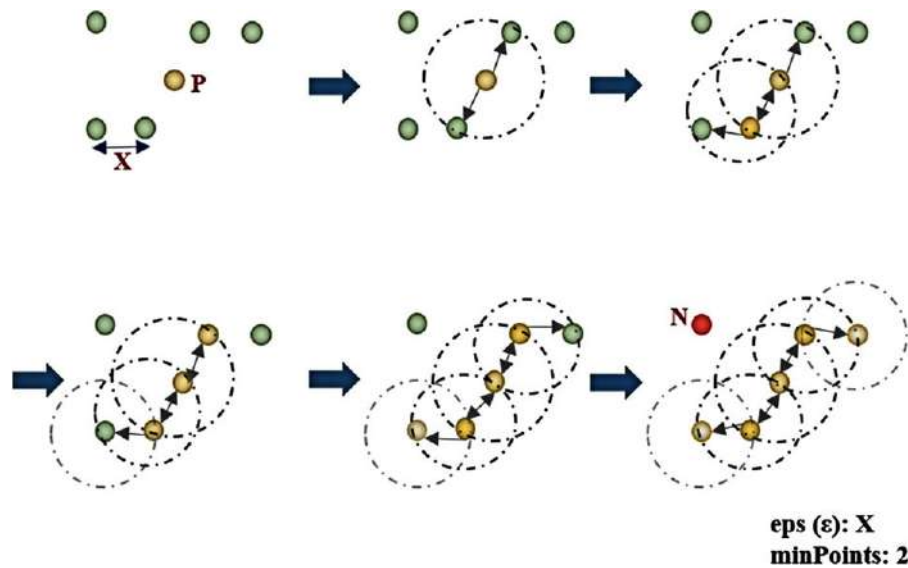


Fig. 6 Density-based spatial clustering of applications with noise (DBSCAN).

“density-reachability” and “density connectivity” cluster model. DBSCAN shares the same principle as linkage based clustering; points are connected to each other based on certain distance threshold values. The difference between linkage and density clustering is that the density clustering algorithm connects only those points that satisfy a given ‘density criterion’ within a certain radius. For example, point “p” is density-reachable from point “q” if p is in the epsilon (ϵ) neighborhood of q and q is a core point. Points “p” and “q” are density connected if point “r” exists in such a way that both p and q are within the epsilon distance along with a number of points in its neighborhood. DBSCAN has low complexity as compared to other clustering algorithms (Moreira *et al.*, 2005).

Algorithm (Fig. 6):

1. A point p is selected.
2. For each core-point p create an edge to every point that is density-reachable from p in the ϵ -neighborhood of p .
3. All points are assigned to a cluster.
4. If p is a border point no points are density-reachable and DBSCAN visits the next point.
5. The process is continued until all the points have been processed.

Subspace Clustering (Biclustering)

Clustering algorithms can help in extracting group of genes with similar variations in expression level under given experimental conditions. Clustering has been identified as one of the most common approaches used for analyzing gene expression data and has been successful in identifying gene pathways and predicting gene function in several studies (Liu *et al.*, 2010). However, clustering algorithms have a major drawback when it comes to gene expression data, as gene expression may have different patterns in different sets of samples (Pontes *et al.*, 2015). To better characterize gene expression, clustering should be performed simultaneously on both dimensions: genes and samples. Another problem is that genes are often involved in different biological processes and as such they cannot be clustered in just one group (Gasch and Eisen, 2002). These drawbacks of the clustering methods are solved by biclustering methods, which have been successfully employed to study gene expression data.

Biclustering, also known as co-clustering, is a data mining technique that can allow the simultaneous clustering of columns and rows of a matrix. J.A. Hartigan introduced biclustering in 1972 (Hartigan, 1972), but the algorithm was generalized by Y. Cheng and G.M. Church in 2000, when they proposed a biclustering algorithm for biological gene expression data that was based on data variance (Cheng and Church, 2000). Biclustering has become a very important technique to study gene expression data and has helped in understanding functionally related gene sets under different conditions (Pontes *et al.*, 2015).

A bicluster is defined as a subset of genes that exhibit consistent patterns over a given subset of conditions (Tanay *et al.*, 2002). Given a gene expression matrix, the biological phenomenon it embodies can be characterized by a collection of biclusters, where each bicluster represents different gene behavior based on the experimental condition. Biclusters have no constraints on them; this often leads to the problem of overfitting. The algorithm is accompanied by a heuristic scoring method or a statistical model that define the significant biological behavior, to guarantee that the biclusters are meaningful (Tanay *et al.*, 2002).

Several biclustering algorithms have been developed to find significant biclusters in an expression matrix. Most of these algorithms use a cost function that helps determine the quality of biclusters (Pontes *et al.*, 2015). Statistical-Algorithmic Method for Bicluster Analysis (SAMBA) is one of the most widely used biclustering methods based on graph theory (Tanay *et al.*, 2002). SAMBA works in three phases:

Phase I

- 1) Modeling of the input gene expression data as a bipartite graph where the two sides of the graph nodes correspond to genes and conditions and the edges represent the significant changes in the expression.
- 2) Weights to the vertex pairs are assigned according to a probabilistic method and the genes (upregulated or downregulated) are considered in a condition if the standardization levels are above 1 or below -1 .
- 3) The choice of signed or unsigned graphs depends on the data.

Phase II

- 1) Hashing technique is applied to find out the heaviest bidiques in the graph; the algorithm looks for the 'k' best bidiques that intersects every gene or set condition.

Phase III

- 1) A local improvement procedure on the biclusters in each of the heaps is performed by the algorithm, then the best modification is applied to the biclusters until no score improvement is possible.
- 2) To avoid similar biclusters with only a slight difference in vertex sets, the algorithm greedily filters the biclusters whose intersection with a solution is above a set L%.

Graph-Based Clustering

Graph-based clustering is a collection of clustering algorithms that are based on various concepts of graph theory. In a graph, the objects are represented by nodes and the nodes are connected by edges which are assigned a weight, defined as the distance between two nodes. In graph clustering, nodes are grouped into different clusters by considering the weights on the edges of the graph and their weights, if weighted graphs are used to model the network. The clusters are formed based on homogeneity and separation such that clusters have high degree of edge density (or have edges of high weights in the case of weighted graphs) while few edges exist between clusters (or number of edges within each cluster is more than the number of edges between different clusters). Graphs can have weighted and unweighted edges depending on the data. Graph-based algorithms are widely used for clustering biological networks (Fig. 7). The past two decades have seen an increase in research of the continuity clustering algorithms as opposed to the classical methods such as k-means and hierarchical clustering (Schaeffer, 2007).

Markov Clustering (MCL), Spectral Clustering (continuity clustering algorithm) and Speed and Performance In Clustering (SPICi) are some of the most widely-used algorithms that were specifically designed to cluster protein-protein interaction networks, gene-expression data, and other biological data.

MCL

Markov cluster algorithms are based on the concept of random walks, which are calculated using Markov chains. Markov chains are a sequence of variables where the present state of the variable is independent of the past or future state (Van Dongen, 2000).

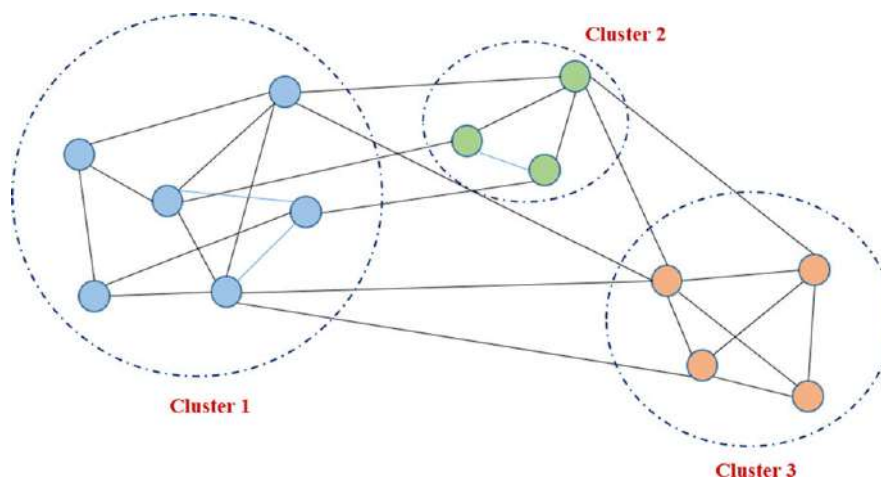


Fig. 7 Schematic representation of a graph network.

The algorithm simulates random walks and identifies densely connected areas within a graph. MCL algorithm is based on flow simulation within a graph; the flow is promoted where the current is strong and it is demoted where the current is weak. This process of flow simulations is made possible by transforming the graph into a Markov graph: a graph where the sum of weights of all the nodes arcs to one. The Markov process is used to compute the powers of the associated Markov matrix (flow expansion).

MCL is performed by alternating an expansion step, which allows simulated walks of long distance, with an inflation step that sharpens the structure of the local clusters. Expansion and inflation are operators that convert one stochastic matrix (elements of the matrix correspond to the probability values) into another. Expansion is a part of linear algebra and takes the power of a stochastic matrix by using normal matrix multiplication. Inflation, on the other hand, is a highly non-linear operator that takes the Hadamard power (powers are taken entry-wise) of the matrix and is followed by a scaling step such that the output is again a stochastic matrix (Lin *et al.*, 2007). The number of clusters is strongly influenced by the value of the inflation parameter.

Algorithm:

1. Input: An undirected graph, with power parameter 'e' and inflation parameter 'r'.
2. Creation of the associated matrix.
3. Normalization of the matrix.
4. Expansion by taking the 'e'th power of the matrix.
5. Inflation by taking the inflation of the resulting matrix with parameter 'r'.
6. Repeat steps 4-5 until a steady state is reached.
7. Analysis and interpretation of the resulting matrix to discover the clusters.

Bootstrapping is an important asset of MCL algorithm. It helps in retrieving cluster structures by using the imprint on the flow process. Some of the benefits of the MCL algorithm are: (1) it is a fast and scalable method; (2) it influences cluster granularity; (3) it is not misled by the edges that link different clusters; and (4) analysis of the algorithm has proven that there is an intrinsic relationship between the flow simulation process and the structures of the clusters in the input graph.

Spectral clustering

Continuity algorithms, as the name suggests seek the continuity of the data rather than looking for a centroid like classical clustering algorithms (Menendez *et al.*, 2014). Spectral clustering (SC) is a very popular continuity clustering method which is based on finding the best edge cut in the input graph. Spectral clustering uses the concept of similarity graphs and the information generated from the eigenvectors so the eigenvalues of the matrices can be generated and analyzed to find the best cut. Similarity graphs can be used to reformulate the clustering. Depending on the local neighborhood relationship between data points, there are different types of similarity graphs (Von Luxburg, 2007).

1. The " ϵ -neighborhood" graph: unweighted graphs where points with a pairwise distance smaller than the ϵ value.
2. The "k-nearest neighborhood" graph: two vertices v_i and v_j are connected if the later lies in the neighborhood of the vertex v_i .
3. The "Fully Connected" Graph: All the points with positive similarity are clustered together and edges are weighted by the similarity value s_{ij} .

The aim is to search for partitions in the data such that edges with high weights connects elements within each group while edges with low weights left outside the clusters connecting elements that belong to different groups (Von Luxburg, 2007). Parameter selection is critical to the quality of the clusters that are formed using these algorithms. The Laplacian matrix measures the difference in the value of a vertex from its neighbors in a graph. Laplacian matrices (also known as Kirchhoff matrices) are the most important part of spectral clustering algorithms. There are two different variants of Laplacian matrices: normalized and un-normalized that are used in the different spectral clustering algorithms. Given a graph G , with n vertices, its Laplacian matrix is defined by the $n \times n$ matrix $L(G) = D(G) - A(G)$, where $D(G) \rightarrow$ Diagonal matrix of vertex degrees and $A(G)$ is the adjacency matrix (Merris, 1994). The un-normalized Laplacian graph is defined as: $L = D - W$ where Matrix L satisfies several properties, such as L is symmetric and semi-definite (positive).

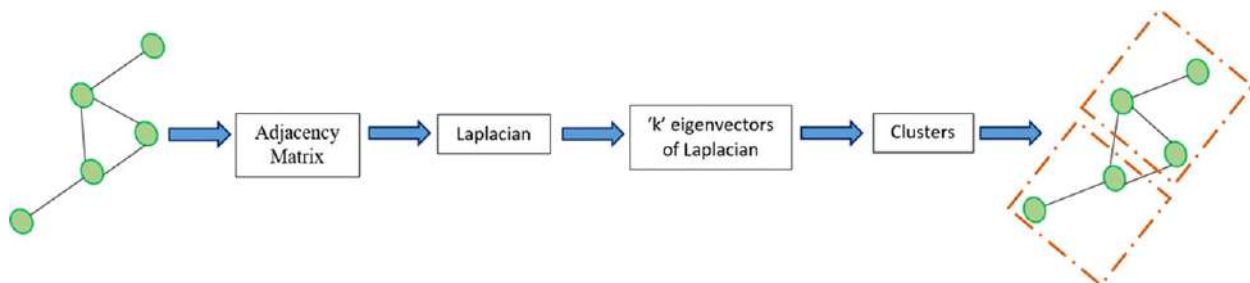


Fig. 8 Schematic diagram representing spectral clustering algorithm.

As spectral clustering seeks data continuity, it gives the best results when used for pattern recognition purpose. The use of the eigenvectors makes it a valuable tool for image processing (Fig. 8). The majority of spectral clustering algorithms minimize the interaction between dissimilar nodes to balance the size of the clusters (White and Smyth, 2005). All spectral clustering algorithms fall under three different categories (Verma and Meila, 2003):

1. Recursive Spectral Algorithms: The Recursive algorithms split the data into two halves based on an eigenvector and then these are recursively used to generate more partitions.
2. Multiway Spectral Algorithms: Multiple eigenvectors are used to do a multiway partition of the data.
3. Non-Spectral Algorithms: These are simple grouping algorithms used to cluster data in a short amount of time.

There are four widely used algorithms: (1) The Shi-Malik (SM) Algorithm (Shi and Malik, 2000); (2) Kannan, Vempla and Vetta (KVV) Algorithm (Kannan *et al.*, 2004); (3) Ng, Jordon and Weiss (NJW) Algorithm (Ng *et al.*, 2002); and (4) MeilPa and Shi Multicut Algorithm (MeilPa and Shi, 2001).

Speed and performance in clustering (SPICi)

SPICi was developed to analyze large networks (Jiang and Singh, 2010). It is a heuristic method that uses a greedy approach and builds clusters by starting from local seeds with a high weighted degree and adds nodes to maintain the density of the cluster. SPICi is based on a simple cluster expansion approach and uses various seed selection criteria along with different interaction confidence values. It is between 4 and 1000 times faster than other clustering approaches on biological networks and is one of the algorithms that is able to cluster large networks within a reasonable amount of time (Jiang and Singh, 2010).

The main goal of SPICi is to output a “set of disjoint dense subgraphs”. It uses the heuristic based greedy approach to build clusters which are expanded from the original seed pair of proteins. The density of a cluster is maintained over a set threshold value by adding only those unclustered nodes to the cluster that have a high support value. If the support value is not sufficiently high, nodes are removed from and the cluster.

Algorithm:

The algorithm is based on two parameters; (1) T_s : the support threshold and (2) T_d : the density threshold and consists of two main steps:

1. **Search**
 - i. Initialize a degree = V
 - ii. u extracted from DegreeQ with largest weighted degree
 - iii. If u has adjacent vertices then the second seed protein v is extracted from the vertices.
2. **Expand (u, v)**
 - i. Initialize a cluster $S = \{u, v\}$
 - ii. Initialize a CandidateQ that contains vertices neighboring u or v
 - iii. t is extracted from Candidate with highest support (t, S)
 - iv. Condition: Support (t, S) $\geq T_s * |S| * \text{density}(S)$ and
 $\text{Density}(S \cup \{t\}) \geq T_d$
Then, $S = S + \{t\}$

DegreeQ is a data structure and part of the priority queue which picks the seed proteins from which the clusters are built. CandidateQ are also part of the priority queue and are used to expand clusters.

Validation Criteria

Clustering algorithms depend on the features of the data and the input parameters. Clustering validity indices are used to determine the input parameters and evaluate the clusters. Validity indices are defined by: (1) compactness; and (2) separation (Liu *et al.*, 2010). Compactness measures how closely related objects in a cluster are. Compactness can be measured based on variance or distance. Some methods use variance as a variable to calculate the compactness of clusters, with lower variance indicating better compactness. Some methods use distance as a measure for compactness (e.g., maximum pairwise distance). Separation measures how distinct one cluster is from another. The minimum pairwise distances between objects in different clusters are widely used measures for measuring separation.

Validity measures are divided into two major categories (Rendón *et al.*, 2011): internal and external validity measures. The internal validity measures rely only on information in the data. They are used to evaluate the quality of the clusters without respect to any external information. External validity measures, on the other hand, rely on the external information and they know the “true” cluster number in advance, making them the best method for choosing the optional clustering algorithm for a specific dataset. The only setback is the requirement of additional information. Internal clustering validation can be used to choose the optimal clustering method along with the optimal cluster number without any additional information. In practice, it is difficult to find external information such as class labels, making external validation more difficult than internal validation.

Table 1 Widely used internal validity measures

Measure	Definition
1. Ball-Hall index Ball and Hall (1965)	$\mathcal{C} = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} \sum_{i \in I_k} \ \mathbf{M}_i^{(k)} - \mathbf{G}^{(k)} \ ^2$ where, $\mathbf{M}_i^{(k)}$ = Observations in a cluster and $\mathbf{G}^{(k)}$ = Barycenter of the cluster.
2. Dunn Index (D) Dunn (1974)	$\mathcal{C} = \frac{\min_{1 \leq i \leq m} \delta(\mathbf{C}_i, \mathbf{C}_j)}{\max_{1 \leq k \leq m} \Delta_k}$ where, $\delta(\mathbf{C}_i, \mathbf{C}_j)$ is the intercluster distance between the clusters ($\mathbf{C}_i$ and $\mathbf{C}_j$)
3. Silhouette (S) index Rousseeuw (1987)	$\mathcal{C} = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} \sum_{i \in I_k} \frac{\min_{k' \neq k} \frac{1}{n_{k'}} d(\mathbf{M}_i, \mathbf{M}_{k'}) - \frac{1}{n_k} \sum_{i' \in I_k} d(\mathbf{M}_i, \mathbf{M}_{i'})}{\max \left(\frac{1}{n_k} \sum_{i' \in I_k} d(\mathbf{M}_i, \mathbf{M}_{i'}), \min_{k' \neq k} \frac{1}{n_{k'}} d(\mathbf{M}_i, \mathbf{M}_{k'}) \right)}$ Where, $d(\mathbf{M}_i, \mathbf{M}_{i'})$ = distance between the observations
4. Davies-Bouldin index (DB) Davies and Bouldin (1979)	$\mathcal{C} = \frac{1}{K} \sum_{k=1}^K \max_{k' \neq k} \left(\frac{\delta_k + \delta_{k'}}{2} \right)$ where, δ_k = mean distance of the points belonging to the cluster $\mathbf{C}_k$ to the barycenter $\mathbf{G}^{(k)}$
5. Calinski-Harabasz index (CH) Caliński and Harabasz (1974)	$\mathcal{C} = \frac{N-K}{K-1} \frac{BGSS}{WGSS}$ where, $WGSS = \sum_{k=1}^K \sum_{i \in I_k} \ \mathbf{M}_i^{(k)} - \mathbf{G}^{(k)} \ ^2$ and $BGSS = \sum_{k=1}^K n_k \ \mathbf{G}^{(k)} - \mathbf{G} \ ^2$
6. S_Dbw index Halkidi and Vazirgiannis (2001)	$\mathcal{C} = \frac{\frac{1}{K} \sum_{k=1}^K \ (\text{Var}(\mathbf{V}_1^{(k)}), \dots, \text{Var}(\mathbf{V}_p^{(k)})) \ }{\ (\text{Var}(\mathbf{V}_1), \dots, \text{Var}(\mathbf{V}_p)) \ } + \frac{2}{K(K-1)} \sum_{k < k'} \frac{\gamma_{kk'}(\mathbf{H}_{kk'})}{\max(\gamma_{kk'}(\mathbf{G}^{(k)}), \gamma_{kk'}(\mathbf{G}^{(k')}))}$ Where, $\gamma_{kk'}$ = density for a given point and $\mathbf{H}_{kk'}$ = midpoints of the clusters and $\mathbf{G}^{(k)}$ = Barycenter of the cluster
7. SD index Halkidi et al. (2001)	$\mathcal{C} = \alpha \frac{\frac{1}{K} \sum_{k=1}^K \ (\text{Var}(\mathbf{V}_1^{(k)}), \dots, \text{Var}(\mathbf{V}_p^{(k)})) \ }{\ (\text{Var}(\mathbf{V}_1), \dots, \text{Var}(\mathbf{V}_p)) \ } + \frac{\max_{k \neq k'} \ \mathbf{G}^{(k)} - \mathbf{G}^{(k')} \ }{\min_{k \neq k'} \ \mathbf{G}^{(k)} - \mathbf{G}^{(k')} \ } \sum_{k=1}^K \frac{1}{\sum_{k'=1, k' \neq k}^K \ \mathbf{G}^{(k)} - \mathbf{G}^{(k')} \ }$ Where, $\mathbf{V}_1^{(k)}$ = variance vectors and $\mathbf{G}^{(k)} - \mathbf{G}^{(k')}$ = distance between barycenters of the clusters.
8. Root mean square std dev (RMSSTD) Sharma (1995)	$\mathcal{C} = \left\{ \frac{\sum_i \sum_{x \in C_i} \ \mathbf{x} - \mathbf{c}_i \ ^2}{[P \sum_i (n_i - 1)]} \right\}^{\frac{1}{2}}$ where, $\mathbf{c}_i$ = center of the i -th center $\mathbf{C}_i$ and n_i = number of objects in $\mathbf{C}_i$ and P = attributes number of datasets D
9. Baker-Hubert Gamma index Baker and Hubert (1975)	$\mathcal{C} = \frac{s^+ - s^-}{s^+ + s^-}$ where, $s^+ = \sum_{(r,s) \in I_B} \sum_{(u,v) \in I_W} 1_{\{d_{uv} < d_{rs}\}}$ and $s^- = \sum_{(r,s) \in I_B} \sum_{(u,v) \in I_W} 1_{\{d_{uv} > d_{rs}\}}$

Source: Reproduced from Desgraupes, B., 2013. Clustering indices. Technical Report, University of Paris Ouest – Lab Modal'X, p. 34.

Internal Validation

Several validation measures have been proposed over the years. The internal validation measures have certain limitations. For example, noise in the data can affect the pairwise distance calculated between objects, which in turn affects the performance score of the clustering algorithm. The performance of the existing internal validation measures in different situations remains unknown. The validation measures depend on the type of data and its characteristics. There are several internal validity indices (Table 1), which are widely used to validate the clustering results.

Internal validation measures follow four steps to determine the optimal cluster number and the best partition (Liu et al., 2010):

- 1) Initializing the list of clustering algorithms that are applied to the dataset.
- 2) Using different combinations of parameters to get different clustering results for each of the clustering algorithms.
- 3) The internal validation index of each of the partitions obtained in the previous step is calculated.
- 4) The optimal cluster number and the best partition are selected according to the criteria.

External Validation

The external validation criteria are also based on statistical methods. User specific intuition is required while evaluating the clusters using the external validation criteria (Kovács et al., 2005). External criteria evaluate the cluster results based on a set of pre-specified

Table 2 Widely used external validity measures

	Measure	Definition
1.	F-measure Rendón et al. (2011)	$F(i, j) = \frac{2 \text{Recall}(i, j) \text{Precision}(i, j)}{\text{Precision}(i, j) + \text{Recall}(i, j)}$ where, $\text{Recall}(i, j) = \frac{n_{ij}}{n_i}$ and $\text{Precision}(i, j) = \frac{n_{ij}}{n_j}$
2.	The Folkes–Mallows index Fowlkes and Mallows (1983)	$C = \sqrt{\text{Precision} * \text{Recall}}$ where, $\text{Recall}(i, j) = \frac{n_{ij}}{n_i}$ and $\text{Precision}(i, j) = \frac{n_{ij}}{n_j}$
3.	Entropy Rendón et al. (2011)	$E = \sum_{j=1}^m \frac{n_j}{n} \sum_i p_{ij} \log p_{ij}$ where, p_{ij} =purity
4.	The Kulczynski index Kulczyński (1928)	$C = \frac{1}{2} (\text{Precision} + \text{Recall})$ where, $\text{Recall}(i, j) = \frac{n_{ij}}{n_i}$ and $\text{Precision}(i, j) = \frac{n_{ij}}{n_j}$
5.	Purity Rendón et al. (2011)	$\text{Purity} = \sum_{j=1}^m \frac{n_j}{n} * \frac{1}{n_j} \max_i (n_{ij}^i)$, where n_j =size of cluster 'j' and m is the number of clusters.

Source: Reproduced from Desgraupes, B., 2013. Clustering indices. Technical Report, University of Paris Ouest – Lab Modal'X, p. 34.

parameters that reflect the user's intuition about the clustering ([Halkidi et al., 2001](#)). Some of the most widely used external validity indices are listed in [Table 2](#).

Conclusions

With the remarkable continuous advancements in instrumentations supporting biomedical research, massive amount of various types data is generated at a rapid rate. It can be argued that progress in biosciences and healthcare will be measured for the most part by how well researchers will be able to take advantage of the available data. Analytical tools will play a significant role in that regard. With biological networks emerging as a natural way to model and arrange the available data, clustering techniques are bound to represent a major tool in analyzing biological data and in extracting new knowledge desperately needed for reaching new medical discoveries. In this study, we show that although many clustering algorithms are already available to bioinformatics researchers, the research question of how to cluster biological data is far from having a clear answer. No current approach is consistently superior for addressing the clustering problem for all types of biological data. Different methods perform better than others in clustering certain data types while optimizing specific criteria. Which clustering approach should be used to cluster a given set of biological data to extract specific pieces of knowledge remains very much a current and relevant research question in bioinformatics. In addition, the issue of how to assess the quality of the clusters obtained by clustering algorithms and validate the biological value of the knowledge they provide represents an even more challenging research question. Although our study attempts to identify the advantages and disadvantages of each clustering approach, more research is needed to answer these two questions.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Quantification of Proteins from Proteomic Analysis

References

- Baker, F.B., Hubert, L.J., 1975. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association* 70 (349), 31–38.
- Ball, G.H., Hall, D.J., 1965. ISODATA, a novel method of data analysis and pattern classification. Menlo Park, CA: Stanford Research Inst.
- Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. *Biometrics* 803–821.
- Calinski, T., Harabasz, J., 1974. A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods* 3 (1), 1–27.
- Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A., Charrad, M.M., 2014. Package 'NbClust'. *Journal of Statistical Software* 165, 1–36.
- Cheng, Y., Church, G.M., 2000. Bicustering of expression data. *ISMB* 8 (2000), 93–103.
- Davies, D.L., Bouldin, D.W., 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 224–227.
- Dongen, S.V., 2000. Graph clustering by flow simulation. PhD Thesis, University of Utrecht, Utrecht.
- Dunn, J.C., 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics* 4 (1), 95–104.

- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* 96 (34), 226–231.
- Estivill-Castro, V., 2002. Why so many clustering algorithms: A position paper. *ACM SIGKDD Explorations Newsletter* 4 (1), 65–75.
- Fowlkes, E.B., Mallows, C.L., 1983. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 78 (383), 553–569.
- Fraley, C., Raftery, A.E., 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal* 41 (8), 578–588.
- Gasch, A.P., Eisen, M.B., 2002. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology* 3 (11), doi:10.1186/gb-2002-3-11-research0059.
- Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. *Journal of Intelligent Information Systems* 17 (2), 107–145.
- Halkidi, M., Vazirgiannis, M., 2001. Clustering validity assessment: Finding the optimal partitioning of a data set. In: *Proceedings IEEE International Conference on Data Mining*, 2001. *ICDM 2001*, IEEE, pp. 187–194.
- Hartigan, J.A., 1972. Direct clustering of a data matrix. *Journal of the American Statistical Association* 67 (337), 123–129.
- Huttenhower, C., Flamholz, A.I., Landis, J.N., *et al.*, 2007. Nearest neighbor networks: Clustering expression data based on gene neighborhoods. *BMC Bioinformatics* 8 (1), 250.
- Inza, I., Calvo, B., Armañanzas, R., *et al.*, 2010. Machine learning: An indispensable tool in bioinformatics. *Bioinformatics Methods in Clinical Research*. 25–48.
- Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. *ACM Computing Surveys (CSUR)* 31 (3), 264–323.
- Jiang, P., Singh, M., 2010. SPICi: A fast clustering algorithm for large biological networks. *Bioinformatics* 26 (8), 1105–1111.
- Kannan, R., Vempala, S., Vetta, A., 2004. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)* 51 (3), 497–515.
- Kovács, F., Legány, C., Babos, A., 2005. Cluster validity measurement techniques. In: *Proceedings of the 6th International symposium of hungarian researchers on computational intelligence*.
- Kulczyński, S., 1928. *Die pflanzenassoziationen der pieninen*. Imprimerie de l'Université.
- Lin, C., Cho, Y.R., Hwang, W.C., Pei, P., Zhang, A., 2007. Clustering methods in protein–protein interaction network. *Knowledge Discovery in Bioinformatics: Techniques, Methods and Application*, 1–35.
- Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., 2010. Understanding of internal clustering validation measures. In: *2010 IEEE Proceedings of the 10th International Conference on Data Mining (ICDM)*, IEEE, pp. 911–916.
- MeilPa, M., Shi, J., 2001. Learning segmentation by random walks. *Neural Information Processing Systems*.
- Menendez, H.D., Barrero, D.F., Camacho, D., 2014. A genetic graph-based approach for partitionial clustering. *International Journal of Neural Systems* 24 (03), 1430008.
- Merris, R., 1994. Laplacian matrices of graphs: A survey. *Linear Algebra and Its Applications* 197, 143–176.
- Moreira, A., Santos, M.Y., Carneiro, S., 2005. *Density-Based Clustering Algorithms – DBSCAN and SNN*. Portugal: University of Minho.
- Ng, A.Y., Jordan, M.I., Weiss, Y., 2002. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems* 2, 849–856.
- Pontes, B., Giraldez, R., Aguilar-Ruiz, J.S., 2015. Biclustering on expression data: A review. *Journal of Biomedical Informatics* 57, 163–180.
- Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M., 2011. Internal versus external cluster validation indexes. *International Journal of Computers and Communications* 5 (1), 27–34.
- Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20, 53–65.
- Schaeffer, S.E., 2007. Graph clustering. *Computer Science Review* 1 (1), 27–64.
- Sharma, S., 1995. *Applied multivariate techniques*. John Wiley & Sons, Inc.
- Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions* 22 (8), 888–905.
- Tan, P.N., 2006. *Introduction to Data Mining*. India: Pearson Education.
- Tanay, A., Sharan, R., Shamir, R., 2002. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* 18 (Suppl. 1), S136–S144.
- Valpola, H., 2000. *Bayesian ensemble learning for nonlinear factor analysis*. PhD Thesis, Helsinki University of Technology, Espoo, Finland. Published in *Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 108*, 54pp.
- Verma, D., Meila, M., 2003. A comparison of spectral clustering algorithms. *University of Washington Tech Rep UWCSE030501*, vol. 1, pp. 1–18.
- Von Luxburg, U., 2007. A tutorial on spectral clustering. *Statistics and Computing* 17 (4), 395–416.
- White, S., Smyth, P., 2005. A spectral clustering approach to finding communities in graph. *SDM* 5, 76–84.
- Xu, R., Wunsch, D.C., 2010. Clustering algorithms in biomedical research: A review. *Biomedical Engineering, IEEE Reviews* 3, 120–154.
- Zhong, C., 2013. *Improvements on Graph-based Clustering Methods*. Issue 114 of *Dissertations in Forestry and Natural Sciences*. ISBN: 9526111818; 9789526111810.

Biological Pathways

Giuseppe Agapito, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological pathways are an abstract representation methodology that is used to represent a certain number of biological events reactions, phosphorylation, catalysis, inhibition, deactivation, etc. that govern the normal or abnormal life cycle of the cellular machinery. Effectively, biological pathways can trigger a series of interactions among molecules in a cell that leads to the production of new molecules (such as fat or protein) or to change in the cell (turning genes on and off or spurring a cell to move). The activation of protein assembly is regulated by an intricate network of interactions among various molecules that lie into the cell, and with the outwards too (Ridley, 2011). Moreover, cells constantly receive signals from both inside and outside the body, which are prompted by such events as injury, infection, sun light, or even food. In order to fit and react to these signals, cells organize their actions sending and receiving signals through biological pathways. The molecules that comprise pathway interact together by means of signals to carry out their assigned tasks. The range of action of a biological pathway is not limited to the inside of the cell, in fact, a pathway can act with other pathways into its neighbourhood (signalling the start of a repair activity or meiosis) or on long distance (pathway product is a hormone that needs to move for long distances into the blood in order to reach the receptor target) (Hla, 2003). A vast number of pathways do not begin at the point i and finish at the point j . Indeed, several pathways have no real limits, and pathways often work in synergy to accomplish multiple tasks. When many biological pathways interact with each other, they produce a very intricate biological network as a result. Thus, biological pathways regulate the total activity of an organism. Therefore, it is important to emphasize that *biological pathways sometimes may function improperly*. When something goes improperly in a pathway, the result can be a disease such as: *cancer*, *diabetes* or *Alzheimer* (Antoni et al., 2006). Thanks to high-throughput analysis that allowed the investigation of global structures instead of the behaviour of individual molecules, it became possible to highlight and understand in a remarkable way, how these complex pathways networks work. Understanding how pathways work is very important, since, it is now becoming increasingly evident that the connection between pathways and diseases are fashioned at multiple interconnected levels. Biological pathways information are available through a significant number of databases ranging from databases manually created by human curators to large databases, comprising a vast number of pathways, created through biological experiments, or inferred electronically by using Natural Language Processing (NLP) techniques, or text/data mining from data coming from the scientific literature. Because of the differences in size, quality, and property, it is essential to use the right database that better fits the user's purpose. Reason for which at nowadays exist public and private biological pathway databases serving as repositories of the current knowledge on pathways (Cerami et al., 2010; Rahmati et al., 2017; Croft et al., 2010; Schaefer et al., 2008; Kanehisa and Goto, 2000; Goto et al., 2002; Bader et al., 2006; Mi and Thomas, 2009). Coordinated efforts are now being made to provide tools to consolidate, visualize, standardize and handle with these large amount of available data. This should make the data easily available to further analysis such as to attempt the decoding of other unknown principles of known and unknown biological pathways as well as new types of biological pathways. The complexity of biological systems are orchestrated by several interacting molecular types, comprising, DNA, RNA, proteins and other small molecules, yielding as result a network. The main goal of system biology is to map these networks in a way that could provide new insight of cell biology at the molecular level. Even though high-throughput methodologies such as Mass-Spectrometry, Next Generation Sequencing (NGS), DNA-microarray, SNP-microarray, Genome Wide Association Studies (GWAS) and so on, can provide huge amount of data per each single experiment, by simultaneous monitoring hundred thousand of gene, proteins or molecules per time, uncover the underlying network of physical interaction remain a tough challenge. The most advanced high-throughput technology such as microarrays lets researchers observe mRNA concentrations in a cell population on a genome wide scale. Other high-throughput techniques, such as transcription factor binding microarrays (ChIP-chip), compound library screening, protein and metabolite profiles, and genome-wide identification of protein-protein interactions and post translational modifications, are less developed but growing fast. Researchers can also employ reliable quantitative data from lower throughput methodologies such as functional reporter-gene assays. Despite the low data produced per single experiment, those methodologies may be applied even to the analysis of large biological networks, since these smaller but better-understood data can provide critical clue on the mechanisms involved into the phenomena under investigation. Other significant evidence comes from evolutionary conservation; a property often exploited in other areas of biological research, where known preservation of interaction in some species may be used to predict its existence in another species. Most pathway inference approaches to address a particular layer of biochemical interactions, such as transcriptional (protein-DNA), signalling protein-protein, protein-DNA, protein complex (protein-protein), or metabolic (protein-metabolite) interactions. Different layers can have a different network character of gene expression networks. For example, metabolic networks tend to be resistant to change, they are constrained by a few co-factors that influence myriad of reactions, and the corresponding graphs have many branches and loops. Graphs are a mathematical formalism that makes it possible to represent the interactions among objects. Biological networks are often represented as graphs, with "edges" connecting "nodes." (Pathway representation by means of graph is described in the next sections). Researchers are using several different algorithms to try to infer the possible structure of very different biological and artificial networks, and define some metrics with which to evaluate the reliability of the produced models represented by means of a graph. For example, one of the

most common problems that many molecular and cell biologists face in their research is: *I have a protein of interest but don't know its structure/function* (Roy *et al.*, 2010; Rost and Sander, 1993). Even though researchers have proposed several algorithms for network prediction, reconstruction, integration, consolidation, comparing are quite difficult tasks from point of view of the graph algorithm performance. Some of these difficulties arise due to these reasons including:

- heterogeneous source of data, i.e., high-throughput data with high noise or poorly validate and poor accurate data;
- it is difficult to explore biological networks within a specific cellular/molecular context, because network topology and features change changing the context;
- assessment requires the developing of errors balancing methodologies in predicting interacting that are not known; and
- common problem belonging to graph theory are known to belong to the NP-complete class of problems. Such as, finding the largest complete subgraph in a given graph. Algorithm used to figure out evidence comes from evolutionary conservation.

A biological pathway is a series of interactions among several molecules in a cell, i.e., DNA, RNA, proteins, and metabolites that lead to a certain product g.e. a change in a cell, by triggering the synthesis of new molecules, or turning genes on and off. Thus a rough classification of biological pathways could produce three classes of pathways involved in metabolism, in the regulation of gene expression and in the signals transduction.

Biological Pathway Classification

Advances in high throughput technologies have enabled the discovery of many pathways, however, many remain unknown. In fact, it is still a great challenge to identify and classify the missing pathways. Since pathways control almost the totality of biological events of an organism, it is better to describe pathways as a class rather than list them individually. Pathways can be divided into these main classes: *Metabolic*, *Gene Regulation* and *Signal Transduction*.

- *Metabolic Pathways*: refers to pathways that regulate the chemical reactions carried out by a cell in order to transform food into energy. Carbohydrates, lipids, and proteins are the major constituents of foods and serve as fuel molecules for the human body. The digestion (breaking down into smaller pieces) of these nutrients in the alimentary tract and the subsequent absorption (entry into the bloodstream) of the digestive end products make it possible for tissues and cells to transform the potential chemical energy of food into useful work.
- *Gene Regulation Pathways*: consist of pathways responsible for the activation (or deactivation) of genes (on, off). Gene regulation is vital for every organism, because genes contain the key information for all the processes in the cell, including protein production. Moreover, proteins are key components needed to carry out nearly every task in each living organism, including the development of muscles, organs etc.
- *Signal Transduction Pathways*: are pathways capable of delivering a signal from a source to a destination. In more details, they move a signal from a cell to the target receptor located on the cell membrane. Every receptor is compatible only with specific signals. The interaction between receptor and signal, establishes a communication channel allowing the signal to travel through the cell by means of specialized proteins to trigger a specific action in the cell.

The classification can be useful in the process of discovering new pathways to predict functions of pathway. The recognition of a pathway is based on the identification of constituents involved in it, that can provide clues about the pathway working principles and thus, the pathway type. Therefore, highlighting out what pathway is involved in a disease, and detecting which components of the pathway exhibit a wrong behaviour, may lead to more personalized strategies for diagnosing, treating and preventing disease.

The approach of personalized treatments is useful to tackle complex diseases like cancer, where: *urgency is an important issue*. The ability to define an efficient treatment, without try it, allows to get better results for the patient, than trying some drugs and discovering only after some time whether the treatment worked or not, with the possibility that during this time the patient would die.

Model for Biological Pathways

Graphs are a mathematical formalism that makes it possible to represent the interactions among objects. Biological networks are often represented as graphs, with "edges" connecting "nodes." Graphs are quite simple, and it is straightforward to describe and analyze interacting objects, for example, the friendship between people in a social network. Although its simplicity graphs need to be extended to be the ideal tool to represent biological pathways. An extension of graph should take in to account the kind of biological network that should be represented. For example, to represent Protein Protein Interaction Network (PPI), the simple graph is the ideal formalism to convey interaction among proteins, and the edges have a more obvious meaning, indicating for instance that two proteins have been shown to form a complex in laboratory assays. For example, in molecular interaction graphs, the nodes represent genes or gene products and the edges represent specific interactions. Whereas, in a protein-DNA network, an edge may account for the binding of a transcription factor to a promoter region, while in a PPI network it might represent a documented evidence of co-immunoprecipitation or a two-hybrid interaction. A more robust formalism to

graphically express metabolic pathways arises, making necessary extend the graph in order to introduce different types of edges able to deal with several specific situations. In detail, regulatory pathway should be represented as a directed graph or digraph, where the vertices represent proteins and the arcs represent the relations between the corresponding proteins. The relation between two proteins has a direction. Because protein p_1 , can regulate protein p_2 , while p_2 cannot always control p_1 . A digraph represents nodes and edges with optional data, even known as attributes. In a digraph, self-loops are allowed, but multiple edges between the same two nodes are not, i.e., n_1 and n_2 there can be only one edge. Metabolic pathways could be represented by using bipartite graphs. In a bipartite graph, there are two types of nodes, one for compounds and one for reactions, with the edges representing the input-output relationships between reactions and compounds. In a bipartite graph, edges connect compounds to reactions (for substrates) or reactions to compounds (for products), but there are no edges between two reactions or between two compounds. Signalling pathways are conveyed as multi-graphs, where nodes represent proteins and edges represent pairwise interactions between proteins. In particular, in a directed graph, an undirected edge between nodes n_1 and n_2 is replaced by two directed edges (n_1, n_2) and (n_2, n_1). The edges are regularly directed in signalling pathways, such as when a kinase phosphorylates a substrate.

Pathways Data

The number of pathway databases is growing quickly in recent years. This is advantageous because biologists often need to use information from many sources to support their research. But on the other hand, each database has its own representation conventions and data access methods, making data integration from multiple databases a huge challenge. This calls for a need to define an unique file format that makes possible to integrate together data from various databases. Pathway data are encoded using different type of file formats such as: structured file as: XML, OWL, RDF and XML-based, psimi-xml; or unstructured text files. The most used structured file format are: *BIOPAX – LEVEL 1,2,3* (Cite Biological and Medical Ontologies: Biological Pathways Exchange (BioPAX) in Encyclopedia of Bioinformatics and Computational Biology – BIOPAX 2017), *PSI-MI*, *BIOPAX*, *SBML* (Cite Standards and Models for Biological Data: SBML in Encyclopedia of Bioinformatics and Computational Biology 2017), *PSI-MI*, *SBML* and *CELLML*. Whereas, the most used unstructured file format are: MITAB, CSV (Comma Separated Values), psimi-tab tab delimited. With the following peculiarity: sometime the same information can be expressed in different ways, despite is used the same file format to encode the data, making the analysis process hard to achieve. The high level of heterogeneity among available pathway data, making automatic data analysis ineffective. An attempt to reduce the high level of heterogeneity is adopted by the BIOPAX (Demir *et al.*, 2010) consortium. BIOPAX is trying to impose itself as standard to represent biologicals pathways data. *BIOPAX -Biological Pathway Exchange-*, is a meta language defined in OWL DL and is represented in the RDF/XML format. BioPAX aims is, to facilitate the integration and exchange of data maintained in biological pathway databases, by means of a unique way to represent data. Traditionally, integration and comparison of different data from a syntactic and semantic point of view, has always been a challenge in the field of Bioinformatics. The latest version is BioPAX Level 3 and it is the recommended version. Since that, the BioPAX Level 3 supports metabolic pathways, signalling pathways (including states of molecules and generic molecules), gene regulatory networks, molecular interactions, genetic interactions. It is not fully backward compatible with Level 1 and 2. At now, BIOPAX is used to represent biological pathway data, on the correct way to became standard.

Pathways consist of a large number of elements that interact amongst themselves to carry out a collaborative biological action. This huge interaction network can be represented using the graph formalism, where every biological element such as protein, complexes, DNA molecule, RNA molecule etc., can be represented as vertices, and catalysis can be represented as edges (interactions). Currently, a standard graphical representation of a single biologicals elements does not exist, and there is a lack in uniformity in graphical form. Therefore, the same element can be displayed in different ways by different visualization tools. A good solution could be to define a uniform graphical representation of biological elements since, there is need to define an unique way to represent visually a biological element, if it is compared with computational analysis. The graphical representation can be achieved by, through the definition of unique graphical constraints able to convey graphically the biological meaning (Agapito *et al.*, 2013; Pastrello *et al.*, 2013). Involving domain experts – biologist, oncologist – asking what is the best way to convey the biological meaning graphically of each kind of element.

The visualization of biological elements is not a simple task as it might seem, because each element has a specific meaning related with the biological context. Despite the limited number of biological element known, the high content of information related with each one, makes data analysis very hard to face (Cite Visualization of Biological Pathways in Encyclopedia of Bioinformatics and Computational Biology 2017). Due to these problems a further effort it is required in order to develop algorithms (*parsers*) able to exploit biological information to represent opportunely every biological elements. A protein can have more than one meaning in relation with the biological context. Thus, there is need of algorithms able to correctly infer what a specific element means, in order to be able to represent it. The greatest effort in the representation of biological element is related with the management of input data, due the high semantics and syntax complexity that characterizes the data. Thus, understanding the meaning of the data to analyze, is essential in order to properly represent the data. It is coming clear that, before to be able to represent graphically the information, it is necessary a huge effort to parse correctly the information into the input data, in order to create the opportune data structures necessary to store the normalized data that will guarantee an uniformity in the visualization step. Successively, the graphical representation -how to arrange the data-, is accomplished by mean of layout algorithms, with the goal to represent graphically: biological interaction networks (Agapito *et al.*, 2013).

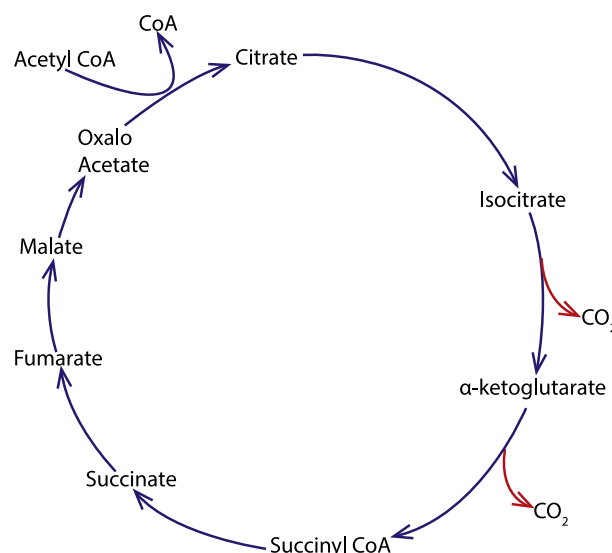


Fig. 1 The citric acid cycle is an example of metabolic pathway.

Metabolic Pathways

Metabolic pathways described the context of the whole cell metabolism. A cellular metabolic reaction network is a collection of enzymatic reactions and transport processes that serve to refill and reduce the relative amounts of individual metabolites. Making it possible the analysis of the structural design and capabilities of the metabolic network. Metabolism is defined as a complex system of chemical and physical processes involved in the maintenance of life. Metabolism comprises a vast collection of enzymatic reactions and transport processes able to convert organic compounds into the several of molecules essential to support cellular life. An example of metabolic pathways is presented in [Fig. 1](#).

The set of reactions and consequently the pathways that a metabolic network holds determine the architecture and topology of the network. The whole set of reactions is controlled by using sophisticated biological control schemes, making it possible to achieve the metabolic objectives throughout the cell's metabolic network. Cells to take advantage of the production capabilities of the metabolic networks, cells have to figure out a method to control the arrangement of the involved molecules and the pathways that determine these capabilities. The functional unit in metabolic networks is the enzyme or gene product, performing a particular chemical reaction or promoting a moving process. Control of metabolism includes regulation of individual reactions at several levels from enzymatic (activation/inhibition) until to the regulation of the genes transcription functions. The cells by physically controlling the single functional units involved in the metabolic network, they can control their metabolic pathways. Thus, trying to understand the regulatory mechanisms achieved by the cell to control the metabolic networks, will allow the analysis and understanding of metabolic pathways.

Due to the complexity of metabolic networks, the main difficulty to represent graphically metabolic pathways concerns with the identification of the whole model comprising the appropriate compounds (metabolites). The major obstacle to represent metabolic pathways is that multiple metabolic pathways could be available, that can link the substrates to the target metabolites due to the large number of potential routes. In addition, without a specific goal and automated ranking function, finding the most efficient pathway could be quite difficult, or could require the assembly and screening of an unrealistically large number of constructs. For this purpose, several computational tools have been developed, that consider the properties of enzymes, intermediate metabolites, metabolic flux and the host cell to support researcher in the assembly and visualization of the metabolic pathways. Thus, the use of computational tools can speed up the analysis of metabolic networks. Highlighting in an easy way all the possible pathways that can lead to the production of a given compound. Thus, the identification of the best variants of candidate compounds, i.e., enzymes, small molecules, may lead to an increase in pathway efficiency.

Computational resources not only help to retrieve information about metabolic pathways but also provide tools for exploring metabolic pathways to highlight the most relevant information and at the same time producing a networks representation able to simplify the visual analysis.

Metabolic Pathway Visualization and Analysis Tools

In this section are listed some of the most known and used tool in the analysis and visualization of Metabolic Pathway. Further information about other tools (not listed in this section) can be found at the follows web address <https://omictools.com/search?q=metabolic+pathway>.

- **BioNetGen** (Faeder *et al.*, 2009): Is an open source package for the specification and simulation of rule-based models of biochemical systems, comprising signal transduction, metabolic, and genetic regulatory networks. The software and the documentation are available at <http://bionetgen.org> web address. BioNetGen is easy to install and runs on Linux, Mac OS X, and Windows operating systems. Moreover, an online version of BioNetGen is available at the following web address: <http://vcell.org/bionetgen>. BioNetGen has been used to model a variety of biological processes. Models are written in BNGL (BioNetGen language), a text-based modeling language. BioNetGen can export model in SBML or MATLAB formats.
- **iPAVS** (Sreenivasaiah *et al.*, 2012): is an integrated biological pathway database designed to support pathway discovery in the fields of omics sciences and systems biology, providing a selection of highly-structured manually curated human pathway data for biologists. In the current version, iPAVS integrates several human pathways including metabolic, signaling, disease-related and drug-action pathways. Furthermore, iPAVS come with a very easy to use web interface allowing biologists to browse and search pathway resources and provides tools for data import, management, visualization, and analysis to support the interpretation of biological data in light of cellular processes. iPAVS web interface implemented advanced functionalities to make complex pathway analysis tasks and visualize the results. As well as, the simple and intuitive design make it possible to use all the advanced iPAVS features even too by everyone. iPAVS web interface and the relative user documentation are available at the following web address <http://ipavs.cidms.org>. Users can export results by using the Systems Biology Graphical Notations (SBGN) and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway notations as well as are used for the visual display of pathway information. Whereas, networks visualization is obtained through Scalable Vector Graphics (SVG).
- **iPath** (Letunic *et al.*, 2008; Yamada *et al.*, 2011): interactive Pathways Explorer (iPath) is a web-based tool for the visualization, analysis and customization of the various pathways maps. Current version provides three different global overview maps: i) metabolic pathways constructed using 1464 KEGG pathways, and gives an overview of the complete metabolism in biological systems; ii) regulatory pathways: 22 KEGG regulatory pathways; and iii) biosynthesis of secondary metabolites: contains 58 KEGG pathways involved in biosynthesis of secondary metabolites. In order to access iPATH web interface, users have to use a web-browser compatible with Adobe Flash plug-ing. The iPATH's user interface present a layout based on tabs between which the users can interact and analyze the visualized data (called Map). In iPath, maps (results) can be exported in various output formats among which: Portable Network Graphics (png), Encapsulated Postscript (eps), Postscript (ps), Portable Document Format (pdf), and Scalable Vector Graphics (svg).
- **Arcadia** (Villger *et al.*, 2010): is a software tool, offering multiple perspectives on the same data, with a focus on navigation. Enabling the interactive exploration on various kind of pathways, visualisation software provides considerable assistance in making sense of complex networks. Arcadia has been designed specifically to provide relevant visualisation options for metabolic pathways, avoiding user to do adjustments on the layouts computed. Arcadia is written in C++ based on the following open-source libraries: i) "LibSBML" an interface to SBML pathway files; ii) "Boost Graph Library" an efficient graph structure library; iii) GraphViz classic layout algorithms; iv) "libavoid" dynamic edge routing library; and v) "Qt" library for the developing of the Graphical User Interface. It is Cross-platform, the code can be compiled to run on Mac, UNIX or Windows systems. Arcadia can be downloaded at the following web address <https://sourceforge.net/projects/arcadiapathways/>. Results can be exported in SBML and SBGN formats, or as vector image in the following formats SVG, PDF, PS.
- **UBViz** (Second, 2005): is a software tool to explore matabolic pathways in 3D space. UBViz is available for the download after free registration at the follow web address <https://www.complex.iastate.edu/download/UBViz/download.html>. UBViz provides and performs the 3D layout algorithm locally on metabolic pathway based on the provided logical descriptions of the pathway. UBViz is a standalone application written by using C++, and it is easily installable by end-users. UBViz graphical interface developed using wxWidgets (see Relevant Websites section), an open-source multiplatform graphical user interface (GUI) toolkit. Node and edge 3D layout is obtained by using a GraphViz module based. Finally, results are rendered on the screen by OpenGL (see Relevant Websites section). Data can be directly downloaded from KEGG by exploiting the FTP functionalities of UBViz or loaded local data stored in the local disk.
- **COPASI** (Hoops *et al.*, 2006; Sahle *et al.*, 2006): is a software application for simulation and analysis of biochemical networks and their dynamics. COPASI can be download at the following address <http://copasi.org/Download/> and it is provided under the Artistic License 2.0. COPASI is a stand-alone program that supports models (reactions) in the SBML standard and can simulate their behavior using ODEs or Gillespie's stochastic simulation algorithm. COPASI's GUI interface is developed to visualize models, with tables for reactions, species, compartments, etc.; several number of network diagrams; the full set of differential equations; these can be exported in Latex or MathML formats. COPASI can perform several simulation either with stochastic kinetics or with differential equations, by implementing several algorithms. Furthermore, models can be analyzed and modified with a large set of available methods. COPASI can import and export models in the SBML format (levels 1 to 3). Models can also be exported in XPP format, Berkeley Madonna format, and as C code (in addition to MathML and Latex).

Table 1 conveys a graphically overview of the features available in each tool listed in this section.

Metabolic Pathways Databases

Comprehensive representations of metabolic pathways have been developed, which requires huge effort and would benefit from information stored in databases and from automatic retrieval and integration methods. Thus, the systematic collection of pathway

Table 1 Metabolic Pathways Tools features comparison. Unsupported features are conveyed by black colour, whereas colours convey the supported features for each tool

Name	OSC			FFS					CSk		RU		AA		V		A				
	Windows	Linux	OSX	BIOPAX	SBML	SIF	XML	TEXT	Other	Basic	Advanced	AcadOnly	Everyone	AfterRegist	Plugin	Web	standalone	2D	3D	Simple	Advanced
BioNetGen																					
iPAVS																					
iPath																					
Arcadia																					
UBViz																					
COPASI																					

In the table OSC, Operating System Compatibility indicates in which operating system the tool is supported; FFS, File Format Supported indicates the approved file format for loading/exporting; CSk, Computer Skills refers to the know-how necessary to use the tool; RU, Restriction To Use specifies the scope where is allowed to use the tool; AA Available Applications indicates the types of available tools; V, Visualization specifies the type of support data visualization; finally A, Analysis conveys the type of analysis capability offered by the tool.

information in the form of pathway databases is crucial. Here is reported a succinct list of the most known metabolic pathways databases.

- **BioPath** (Schreiber, 2002): is a database of biochemical pathways that provides access to metabolic transformations and cellular regulations derived from the Roche Applied Science "Biochemical Pathways" wall chart. In the current version BioPath contains about 14,000 chemical structures, 3900 biochemical transformations (reactions), from the following organisms: prokaryotes, plants, yeasts and animals. Provides the following features, structures available as connection tables, atoms of substrates and products annotated by atom-atom mapping numbers, 3D coordinates of small molecules generated by CORINA Classic and so on. BioPath is freely available for academic users, but requires a license for commercial use, at the follows web address <https://www.mn-am.com/databases/biopath>.
- **HumanCyc** (Romero *et al.*, 2004): is a bioinformatics database that describes the human metabolic pathways and the human genome. By presenting metabolic pathways as an organizing framework for the human genome, HumanCyc provides the user with an extended dimension for functional analysis of Homo sapiens at the genomic level. HumanCyc encoded data by using the BioPAX and SBML data formats. HumanCyc is available at <https://humancyc.org> only after a valid license has been purchased.
- **KEGG** (Kanehisa and Goto, 2000): is a database of metabolic pathways from multiple organisms. KEGG is accessible through web browser at <http://www.kegg.jp/kegg/>. AS well as KEGG provide API (application programming interface) allows customization of KEGG-based analysis. KEGG API is provided for academic use by academic users belonging to academic institutions. KEGG data are encoded by using xml data format.
- **MetaCyc** (Caspi *et al.*, 2007): is a metabolic pathway database contains pathways from over 2816 different organisms. MetaCyc describes metabolic pathways, reactions, enzymes, and substrate compounds. The MetaCyc data were gathered from a variety of literature and on-line sources, and contain citations to the source of each pathway. MetaCyc allows user to download in the following formats: BioPAX, Pathway Tools attribute-value, Pathway Tools tabular, and SBML formats respectively. MetaCyc is free only for academic users at the follows address <https://metacyc.org>.
- **SABIO-RK** (Wittig *et al.*, 2011): SABIO-RK is a curated database that contains information about biochemical reactions, their kinetic rate equations with parameters and experimental conditions. SABIO-RK web interface can be accessed via web-base user interfaces at the following web address <http://sabio.villa-bosch.de>. SABIO-RK can be automatically accessed via RESTful Web Services. The Web Services allow the direct SABIO-RK data access by other tools or systems biology modelling platforms. The output of this Web Services search is available in different formats (e.g., SBML, XML). The user guide where it is explained how to use RESTful API and how to create request are available at the following web address <http://sabio.villa-bosch.de/layouts/content/docuRESTfulWeb/manual.gsp>. data can be exported in spreadsheet, SBML or BioPAX format. Moreover, spreadsheet allows to export the data in a table format (xls or tsv).

Table 2 conveys a graphically overview of the features available in each database listed in this section.

Gene Regulation Pathways

Gene Regulation Pathway explicitly represent the causality of developmental processes. They regulate the expression of thousands of genes in any given developmental process. By understanding the interactions in these networks, can be highlighted in a

Table 2 Metabolic Pathways Databases features comparison. Unsupported features are conveyed by black colour, whereas colours convey the supported features for each tool

Name	FFS						O	RU		
	BIOPAX	SBML	SIF	XML	TEXT	Other	HomoSap	Other	AccadOnly	Everyone
HumanCyc										
KEGG										
MetaCyc										
SABIORK										

In the table FFS, File Format Supported indicates the approved file format for loading/exporting; O, Organism refers to the indicates the organism where data come from; RU, Restriction To Use specifies the scope where is allowed to use the tool.

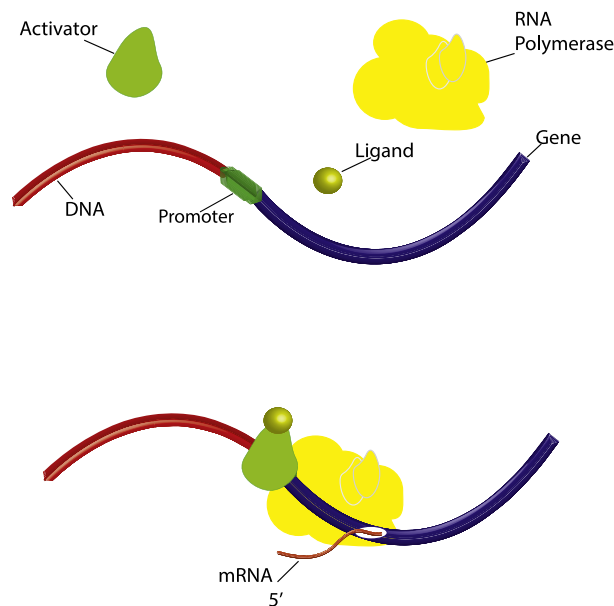


Fig. 2 The activator with bound ligand stimulates transcription, as an example of gene regulation pathway.

remarkable way the mechanisms of diseases that occur when these cellular processes are or not working properly. An example of regulation pathway is depicted in [Fig. 2](#).

They are essentially hardwired genomic regulatory codes, the role of which is to specify the sets of genes that must be expressed in specific spatial and temporal patterns ([Davidson and Levin, 2005](#)). In the last decade, gene regulatory network/pathway have brought a lot of interest, thanks to the recent advancements in genomics and computational biology because are enabling the construction of large-scale models of gene regulatory networks. High-throughput methodologies such as automated sequencing, gene-expression arrays, and yeast two-hybrid assays, examining different features of the gene-regulatory system through genome-wide datasets, producing a vast quantity of data for each single experiment, bringing to light the mechanisms that underlie the gene regulatory pathways and, thus the necessity of automatic methods able to deal with this enormous amount of data, to infer the structure and interactions of gene regulatory pathways arise. At this regard, several statistical and machine-learning methods have been proposed to infer gene regulatory pathway from gene expression data, as evidenced by the following papers ([Liang et al., 1998](#); [Butte et al., 2000](#); [Basso et al., 2005](#); [Pache and Aloy, 2012](#)). In particular, in the work ([Emmert-Streib et al., 2014](#)) authors state that: gene regulatory pathways because are inferred from mRNAs data provide information about regulatory interactions between regulators and their potential targets; gene- gene interactions, and potential protein-protein interactions (e.g., in a complex) ([de Matos Simoes et al., 2013](#)). Because a gene regulatory pathway can potentially describe many kinds of physical, biochemical interactions among genes and gene products ([de Matos Simoes et al., 2013](#)), the assessment of inferred networks is an

essential and complex topic. Networks are complex structures able to model different kinds of features of biological systems. Because biological networks are inferred, the quality of inferred biological networks has to be assessed. The main problems in the assessing of biological inferred network are the definition of a methodology able to detect the *true* interactions also known as gold standard. Gold standard search is addressed by validating interactions functional databases or previous literature (Kanehisa and Goto, 2000; Kotlyar *et al.*, 2016). The main drawback of this approach is that, although the set of known interactions might be large, many of them might not be relevant to the biological conditions under investigation. For these reasons, define a methods and algorithms able to qualitatively assess the accuracy of the obtained network by using the gold standard. Various modelling methodology have been developed for gene regulatory network. The available methodologies used to represent gene regulatory pathways or networks can be classified into three main class: The first-class contains the logical models. Logical models describe regulatory networks qualitatively, providing to the user a basic qualitative idea on how several functionalities of a given system work in an orchestration fashion to accomplish some biological tasks. The second-class contains the continuous models; continuous models are mathematical formalism able to deal with continuous data, i.e., exact molecular concentrations at a particular instant time. For example, to simulate the effects of a particular drug on cancer cells under different active ingredient concentrations, users must use continuous models with a fine-grain resolution. The third class comprises single molecule level models. The single molecule level model describes components (genes) by using stochastic representation, and consequently, allowing to user to get an idea of why should be present different behaviors, even starting from the same initial conditions. The fourth class includes hybrid models. Hybrid models have been promoted to describe both, discrete and continuous aspects in one model. Because in the real world both continuous and discrete elements are present at the same time. For example, gene expressions are represented as continuous values, whereas the binding of a molecule with a receptor is expressed as a discrete event (bound or unbound). The modeling techniques used to represent gene regulatory pathway comprises the following models. *Logical models* are based on a discrete logic modeling methodology as described in Glass and Kauffman (1973), Thomas (1973). Genes, proteins and small molecules in logical models, are represented by using local state information, as discrete information at each instant. Temporal system evolutions happen in a synchronized way and are described as discrete time steps. Boolean networks are described by means an oriented graph G . A graph G is a mathematical formalism able to describe interaction among object (Fauré *et al.*, 2006). A graph G is defined as a couple $G=(V, E)$ where V is the set of boolean vertex vi . Whereas, E is the set of boolean edges that is if exist an edge between the node vi and its parent nodes in G . And at a given time the state of all the nodes represent the state of the network. Probabilistic boolean networks are an extension of the probabilistic network able to deal with uncertainty in the synchronization of regulatory events (Shmulevich *et al.*, 2002a,c,b). Bayesian networks are represented as an annotated directed acyclic graph $G(V, E)$, where the nodes, $\{vi \in V\}$, are random variables denoting genes' expressions, and the edges indicate the reliances among nodes. The random variables are obtained from conditional probability distributions $P(vi|Pa(vi))$, where $Pa(vi)$ is the parents set of each node (Kim *et al.*, 2004; Zou and Conzen, 2004). Continuous models, using real value parameters over an ongoing time scale, allow a straightforward comparison of the global state and experimental data. Linear models allow considering a biological system as a state machine, where the change in internal state of the biological system depends on the current internal state plus any external conditions (De Jong *et al.*, 2004; Hecker *et al.*, 2009). Allowing a high level of abstraction and efficient inference of network structure and regulation functions. For example, linear models are the perfect tool to describe the concentration of mRNA – representing the internal state of a cell. The simplest form of a linear model is reported in Eq. (1).

$$X_i(t + \Delta t) = \sum_k Y_{ik} X_k(t) \quad (1)$$

Where in Eq. (1) $X_i(t + \Delta t)$ is the expression level of the i – th gene at time $(t + \Delta t)$, and Y_{ik} represents the influence of the j – th gene on the i – th gene.

Gene regulatory networks can be encoded by using differential equations systems (Sakamoto and Iba, 2001; Chen *et al.*, 1999). Differential equations allow to better describe the system dynamics, by explicitly modeling the changing of the states (molecules) over the time. The simplest form of a equations system is conveyed in Eq. (2).

$$\begin{aligned} g_1(t + \Delta t) - g_1(t) &= (w_{11}g_1(t) + \dots + w_{1n}g_n(t))\Delta t \\ &\vdots \\ g_n(t + \Delta t) - g_n(t) &= (w_{n1}g_1(t) + \dots + w_{nn}g_n(t))\Delta t \end{aligned} \quad (2)$$

Where in Eq. (2) $g_i(t + \Delta t)$ is the expression level of i – th gene at time $(t + \Delta t)$, and w_{ij} is the weight indicating how much the level of the i – th gene is influenced by j – th gene, with $(i, j = 1, \dots, n)$.

Single molecule level models allow representing regulatory networks by using stochastic modeling, because the amount of regulatory molecules is often low (Ross *et al.*, 1994; Elf *et al.*, 2007; Schütz *et al.*, 1997). Experimental assays demonstrated the stochastic behavior of the processes of transcription and translation (Shea and Ackers, 1985).

Hybrid models have been developed to describe discrete and continuous aspects into a single model. Examples of hybrid models applied to gene regulatory pathways have been presented in Matsuno *et al.* (2000), Xu *et al.* (2007).

For further information on the representation of the methodologies, regulatory pathway gene can be found in the following papers (Karlebach and Shamir, 2008; Vijesh *et al.*, 2013).

Gene Regulatory Pathways Visualization and Analysis Tools

In this section are listed some of the most known and used tool in the analysis and visualization of gene regulatory pathway. Further information about other tools (not listed in this section) can be found at the follows web address <https://omictools.com/gene-regulatory-networks-category>.

- **Sig2GRN** (Zhang *et al.*, 2016): is a software tool linking signaling pathway with gene regulatory network for dynamic simulation. As a software tool for modeling cellular dynamics, Sig2GRN can facilitate studies in systems biology by hypotheses generation and wet-lab experimental design. Sig2GRN is wrote using the Java programming language, and is available as Cytoscape Plugin. Feature that makes Sig2GRN compatible with Linux, Windows and MacOSX operating systems. The plugin is freely available at the following web address Availability: <http://histone.scse.ntu.edu.sg/Sig2GRN/>. The package contains the executable components to be copied into the Cytoscape plugins directory, and a test network with which try the software.
- **ARACNe** (Margolin *et al.*, 2006): Algorithm for the Reconstruction of Accurate Cellular Networks - ARACNe is available as a Cytoscape app that is integrated in the Cyni Toolbox panel. ARACNe plugin does not require to be installed but should just copied into the Cytoscape plugins folder. ARACNe documentation and support are available at the following web address <http://califano.c2b2.columbia.edu/aracne>. ARACNe is available under non-commercial license agreement for the download at the follows address <http://califano.c2b2.columbia.edu/aracne-license> after free registration. ARACNe is available to download as software compatible with all the current operating systems such as Linux, Mac OSX and Windows.
- **Discrete Dynamics Lab (DDLab)** (Wuensche, 2010): is an interactive graphics software for creating, visualizing, and analyzing many aspects of Cellular Automata (CA), Random Boolean Networks (RBNs), and Discrete Dynamical Networks (DDNs) in general, and studying their behavior, both from the time-series perspective - space- time patterns, and from the state-space perspective Attractor Basins. DDLab is free (open source) software under the GNU General Public License. However, institutional users (commercial or educational) are required to register and pay a registration fee. Personal users are also encouraged to register. DDLab is available as web-based application at <http://www.ddlab.com>. The user guide and manuals are available at the following address http://www.ddlab.org/download/dd_manual_2016/.
- **FERN** (Erhard *et al.*, 2008): Framework for Evaluation of Reaction Networks - FERN - is a fast, extensible and comprehensive framework for the stochastic simulation and analysis of chemical reaction networks written in Java. It includes state of the art algorithms for stochastic simulation and a powerful visualization system based on "gnuplot" and Cytoscape. FERN is freely available for download at the follows web address <https://drupal.bio.ifi.lmu.de/FERN/> section download and requirements for download additional features. At the same address can be found user guides, example and API to extend FERN.
- **GRENDL** (Haynes and Brent, 2009): Gene REgulatory Network Decoding Evaluations tool is an open and extensible software toolkit that generates random gene regulatory pathways according to user-defined constraints on the network topology and kinetics. GRENDL in the current release provides Kinetic regulatory functions such as Hill Kinetics and Michaelis-Menten with combinatoric interactions between regulators. Experimental design types supported: genetically diverse population samples, knockouts, overexpression and time course with external stimuli perturbations. The current version of GRENDL is wrote in C++ and is freely available for download at the follows web address <http://mblab.wustl.edu/software.html> section GRENDL.
- **FastNCA** (Chang *et al.*, 2008): Fast network component analysis -FastNCA- for gene regulatory network reconstruction from microarray data. FastNCA algorithm is coded by means of MATLAB, and it is available as MATLAB plugin, freely available for download at the following web address <http://www.ee.ucla.edu/~riccardo/NCA/nca.html>.
- **BNFinder2** (Cooper and Herskovits, 1992): is a software tool for learning Bayesian networks, allowing users, to build Bayesian network starting from experimental data. The current version of BNFinder2 is available for download from Python Package <https://pypi.python.org/pypi/BNfinder>. User's manuals and tutorial are available on the follows web site <http://bioputer.mimuw.edu.pl/software/bnf/>. BNFinder2 allows user to export data in the following formats sif (Simple Interaction File), and bif (Bayesian Interchange Format).

Table 3 conveys a graphically overview of the features available in each tool listed in this section.

Gene Regulatory Pathways Databases

Comprehensive representations of gene regulatory pathways have been developed, which requires huge effort and would benefit from information stored in databases and from automatic retrieval and integration methods. Thus, the systematic collection of pathway information in the form of pathway databases is crucial. Here is reported a succinct list of the most known gene regulatory pathways databases.

- **HRGRN** (Dai *et al.*, 2015): is a graph search-empowered integrative database of Arabidopsis signal transduction, metabolism, and gene regulatory networks. HRGRN employs the highly scalable graph database, Neo4j, to host large-scale of biological interactions among genes, proteins, and small RNAs that were either validated experimentally or predicted computationally. Besides, HRGRN comes with a series of graph path search algorithms to discover unknown relationships among genes and small molecules. HRGRN gives the users the capabilities to build sub-networks based on known interactions. The outcomes are visualized with both rich text, figures and interactive network graphs on web pages. HRGRN is available at <http://plantgm.noble.org/hrgn/>.

Table 3 Gene Regulatory Pathways Tools features comparison. Unsupported features are conveyed by black colour, whereas colours convey the supported features for each tool

Name	OSC			FFS					CSk		RU		AA		V		A				
	Windows	Linux	OSX	BIOPAX	SBML	SIF	XML	TEXT	Other	Basic	Advanced	AccadOnly	Everyone	AfterRegist	Plugin	Web	standalone	2D	3D	Simple	Advanced
Sig2GRN																					
ARACNe																					
DDLab																					
FERN																					
GRENDEL																					
FastNCA																					
BNFinder2																					

In the table OSC, Operating System Compatibility indicates in which operating system the tool is supported; FFS, File Format Supported indicates the approved file format for loading/exporting; CSk, Computer Skills refers to the know-how necessary to use the tool; RU, Restriction To Use specifies the scope where is allowed to use the tool; AA Available Applications indicates the types of available tools; V, Visualization specifies the type of support data visualization; finally A, Analysis conveys the type of analysis capability offered by the tool.

- **myGRV** (Bacha *et al.*, 2009): is a database storing information about regulatory links in genetic regulatory pathways. myGRV also allows rapid visualisation of these networks and lets you export network data to a variety of tools. myGRV is available after free registration to use through web browser at the following address <http://www.nottingham.ac.uk/~plzloose/mygm/>.
- **RegNetwork** (Liu *et al.*, 2015): is a data repository of five type transcriptional and post-transcriptional regulatory relationships for human and mouse. RegNetwork aims to develop a platform for collecting the experimentally validated regulations and the predicted regulations. By using RegNetwork, users can query and identify the combinatorial and synergic regulatory relationships among TFs, miRNAs, and genes. As well as providing the statistics of the built regulatory networks by means of queries. RegNetwork is freely available at <http://www.regnetworkweb.org/search.jsp>.
- **TRED** (Zhang, 2008): is an automated pipeline to extract known promoters from databases such as Genbank, EPD and DBTSS, and employ promoter finding program FirstEF combined with mRNA/EST information and cross-species comparisons. Curation has been done to assess computational prediction and ensure data accuracy. As well as, curation has also been provided for transcriptional regulation information, including transcription factor binding motifs and experimental evidence. TRED is available at the follows web address <https://cb.utdallas.edu/cgi-bin/TRED/tred.cgi?process=home>.
- **TRRUST** (version 2) (Han *et al.*, 2015): is a manually curated database of human and mouse transcriptional regulatory networks. The current version of TRRUST contains 8, 908 and 7, 382 TF-target regulatory relationships of 821 human TFs and 859 mouse TFs, respectively. TRRUST database also provides information of mode of regulation (activation or re-pression). Currently 9, 762 (59.9%) of regulatory relationships are known for mode of regulation. TRRUST is available at the following web address <http://www.grnpedia.org/trrust/>. Query results are conveyed in a tabular format and visualized as network by using HumanNet (Lee *et al.*, 2011). HumanNet is a probabilistic functional gene network of 18, 714 validated protein-encoding genes of Homo sapiens. Data can be download in TSV and BioC format respectively.

Table 4 conveys a graphically overview of the features available in each database listed in this section.

Signal Transduction Pathways

All the chemical or physical signals transmitted from a cell to another i.e., a change in conditions between cells is obtained through hormone exchange. As well as, from a cell to the external environment and vice versa, i.e., molecules in food communicate taste and smell through the interaction with specialized sensor cells. This process is known as *signal transduction*, making possible for the cells to interact with the external environment, by detecting molecules outside the plasma membrane and dispatch the information to the inside of the cell. Thanks to continuous advances in experimental research methodologies, it is even clearer that all these chemical or physical signals govern various biological process i.e., control the cell fates, response to blue light and up regulate or down regulate certain genes, form complex signal transduction networks that go under the name of signal transduction pathways. In particular, the proteins responsible for intercepting the stimuli are known as *receptors*, when a *ligand* binding with a receptor, triggers a series of signalling cascade a chain of biochemical events. Signalling pathways interact among them, forming a network allowing to coordinate cellular responses, by combining signalling events among them. Thus, with the term signal transduction pathways, researchers refer to the totality of biochemical mechanisms able to regulate the cellular machinery (Ray, 1999). Signal transduction pathways at

Table 4 Gene Regulatory Pathways Databases features comparison. Unsupported features are conveyed by black colour, whereas colours convey the supported features for each tool

Name	FFS						O	RU			
	BIOPAX	SBML	SIF	XML	TEXT	Other	HomoSap	Other	AccadOnly	Everyone	AfterRegist
HRGRN											
myGRV											
RegNetwork											
TRED											
RegNetwork											
TRRUST											

In the table FFS, File Format Supported indicates the approved file format for loading/exporting; O, Organism refers to the indicates the organism where data come from; RU, Restriction To Use specifies the scope where is allowed to use the tool.

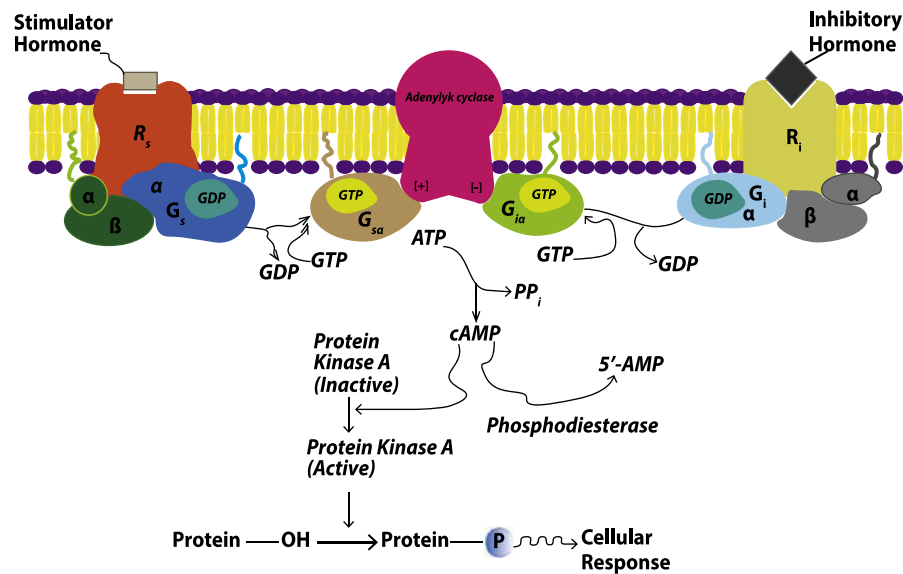


Fig. 3 The adenylyl cyclase signalling pathway, is an example of gene regulation pathway.

the molecular level responses includes changes in the genes transcription or translation tasks, as well as conformational changes in proteins. These molecular events are the core mechanisms controlling cell growth, proliferation, metabolism and many other biological processes. Each component of a signalling transduction pathway is classified according to the role it plays concerning the primary stimulus. An graphical representation of signal transduction pathway is shown in Fig. 3.

A ligand binds to its specific receptor on the cell-surface. As a result, the interaction between ligand and receptor produces a signal that is transferred through a membrane protein-transducer to a membrane-bound effector enzyme. The activity of the effector enzyme produces a small molecule or ion known as the second-messenger. Finally, the second-messenger carries the signal to its ultimate destination which may be in the nucleus, an intracellular compartment, or the cytosol. A few non-polar signal molecules such as estrogens and steroid hormones can spread through the cell membranes and, hence, enter in the cell. Once crossed the membrane, these molecules can bind to proteins that interact with the DNA and modulate gene transcription, altering the gene-expression patterns. A vast diversity of ligands, receptors, and transducers exists, but only a few second-messengers and types of effector enzymes are known. Amplification is an essential feature of signaling pathways. A single ligand receptor complex can interact with some transducer molecules, activating many molecules of effector enzyme. Thus, the production of many second-messenger can activate many kinase molecules that catalyze the phosphorylation of many target proteins. These series of

amplification events is known as a cascade. The cascade mechanism means that small amounts of an extracellular compound can affect large numbers of intracellular enzymes without crossing the plasma membrane or binding to each target protein. There are many types of receptors and various transducers, where bacterial transducers are different than eukaryotic ones. Receptors can be split into two broad classes internal and cell-surface receptors. Internal receptors even known as intracellular, are located in the cytoplasm of the cell, responding to the hydrophobic ligand molecules that can travel across the plasma membrane (Vanderbilt *et al.*, 1987). Once inside the cell, a significant number of these molecules tie to proteins that work as controllers of mRNA synthesis to regulate gene expression. Gene expression is the cell process of transforming DNA into a sequence of amino acids that finally becomes a protein. At the point when the ligand binds to the internal receptor, a conformational change uncovered a DNA-restricting site on the protein. The ligand-receptor complex moves into the nucleus ties to specific regulatory regions of the chromosomal DNA, promoting the start of transcription. Internal receptors can straightforwardly impact the gene expression without passing the signal to different receptors or second messenger. Cell-surface receptors are known as transmembrane receptors (Deller and Jones, 2000; Spear *et al.*, 2000). Transmembrane receptors are located on the cell surface and attached to the membrane, binding to the external specific ligand molecules. These receptors traverse the plasma membrane and perform signal transduction, converting an external cellular signal into an intracellular signal. Ligands that combine with cell-surface receptors don't need to enter into the cell nuclei to influence. A general cell-surface receptor comprises an extracellular ligand-binding domain, a hydrophobic membrane-spanning region, and a domain inside the cell. Cell-surface receptors affect most of the signalling processes in multi-cellular organisms. There are three general categories of cell-surface receptors: ion channel-linked receptors (Ackerman and Clapham, 1997), G-protein-linked receptors (Milligan, 1993; Watson and Arkinstall, 1994), and enzyme-linked receptors (Coussens, 1985). G proteins linked receptors play as transducers transmitting the external stimuli to effector enzymes. G proteins are observed in dozens of signalling pathways including the adenylyl cyclase and the inositol-phospholipid pathways. Ion channel-linked receptors binding a ligand change conformation to open a channel through the membrane that allows specific ions to pass through, i.e., hydrogen, sodium, calcium and so on. Ion channel-linked receptors are observed to act as a second messenger initiating signal transduction cascades and altering the physiology of the responding cell pathway. Finally, the enzyme-linked receptors are cell-surface receptors with intracellular domains associated with an enzyme. When a ligand ties to the extracellular receptor, a signal is conveyed through the membrane activating the enzyme, which starts a chain of events inside the cell that eventually lead to a response. The enzyme-linked receptors operate by a signalling pathway that includes a multifunctional transmembrane protein called a tyrosine kinase receptor. Signal transduction pathways are responsible for the correct cellular function in each tissue and environment (Gumbiner, 1996). Thus a methodology able to structurally represent the signal transduction pathway by using a standard form that can be readily stored, processed, analyzed and visualized by computers would be of great value, as well as visual representation of information make quickly for researcher understand better how signal transduction pathways work.

Signal Transduction Pathways Visualization and Analysis Tools

In this section are summarized some of the most known and used tool in the analysis and visualization of signal transduction pathways. Further information about other tools (not listed in this section) can be found at the follows web address <https://omictools.com/signaling-pathways-category>.

- **Cell Illustrator** (Nagasaki *et al.*, 2010): is a software tool that enables biologists to draw, model, elucidating and simulating complex biological processes and systems. Cell Illustrator allows researchers to model metabolic pathways, signal transduction cascades, gene regulatory pathways and dynamic interactions of various biological entities such as genomic DNA, mRNA and proteins. Cell Illustrator is written by using the Java program language, making it compatible with all the operating system compatible with Java. Cell Illustrator is available for the download at the following web address <http://www.cellillustrator.com/home>. Results can be exported in the CIL, and CSV File format.
- **ChemCell** (Joo *et al.*, 2007): is a particle-based reaction/diffusion simulator designed to model signalling, regulatory, or metabolic networks in biological cells. ChemCell is written in C++ and no additional software is needed to run ChemCell. Furthermore, ChemCell will run on any parallel machine that compiles C++ and supports the MPI message-passing library. ChemCell is an open-source code, and it is freely available for download at <http://www.sandia.gov/~sjplimp/download.html> under the terms of the GNU Public License (GPL).
- **Jarnac** (Sauro, 2000): is an interactive and interpretive language for describing and manipulating cellular system models and can be used to describe metabolic, signal transduction and gene networks, or in any physical system which can be described in terms of a network and associated flows. Jarnac works as a script engine for model building, simulation, and analysis and a simulation engine for SBML files. Performing continuous and stochastic simulations. Jarnac allows users to Export/Import models in SBML XML standard format. Jarnac is currently only supported on Windows and is freely available at <http://sbw.kgi.edu/sbwWiki/sysbio/jarnac>.
- **SBML-SAT** (Zi *et al.*, 2008): is a Systems Biology Markup Language (SBML) based Sensitivity Analysis Tool (SAT). SBML-SAT is designed to run simulation, steady state analysis, robustness analysis, as well as local and global sensitivity analysis for ordinary differential equations (ODE) based biological models. SBML-SAT provides four different global sensitivity analysis methods, including multi-parametric sensitivity analysis, partial rank correlation coefficient analysis, SOBOL's method and weighted average of local sensitivities. SBML-SAT is written by using the Matlab Programming language. Results can be exported in SBML

Table 5 Signal Transduction Pathways Tools features comparison. Unsupported features are conveyed by black colour, whereas colours convey the supported features for each tool

Name	OSC			FFS					CSk		RU	AA			V		A				
	Windows	Linux	OSX	BIOPAX	SBML	SIF	XML	TEXT	Other	Basic	Advanced	AccadOnly	Everyone	AfterRegist	Plugin	Web	standalone	2D	3D	Simple	Advanced
CellIllustrator	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ChemCell	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Jarnac	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SBML-SAT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SimBoolNet	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TENET	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

In the table OSC, Operating System Compatibility indicates in which operating system the tool is supported; FFS, File Format Supported indicates the approved file format for loading/exporting; CSk, Computer Skills refers to the know-how necessary to use the tool; RU, Restriction To Use specifies the scope where is allowed to use the tool; AA Available Applications indicates the types of available tools; V, Visualization specifies the type of support data visualization; finally A, Analysis conveys the type of analysis capability offered by the tool.

Level 1 Version 1 and 2, and SBML Level 2 Versions 1, 2 and 3. SBML-SAT is available for Windows, Mac, and Linux can be freely downloaded from its website <https://www2.hu-berlin.de/biologie/theorybp/?goto=tools>.

- **SimBoolNet** (Zheng *et al.*, 2009): is an software tool for simulating the dynamics of signalling transduction by using boolean networks. Users by specifying a level of stimulation to signal receptors, SimBoolNet simulates the response of downstream molecules and visualizes with animation and records the dynamic changes of the network. SimBoolNet is available as a Java plug-in for Cytoscape, allowing users to take advantage of the functionalities of Cytoscape. SimBoolNet can be freely downloaded from the following website <https://www.ncbi.nlm.nih.gov/CBBresearch/Przytycka/index.cgi#simboolnet>. SimBoolNet import networks encoded by using the delimited text and Microsoft Excel formats. Image files can be saved in the following formats including PNG, JPEG, and vector graphics file formats (including PDF, EPS and SVG, while statistical results can be saved in plain text file format.
- **TENET** (Chua *et al.*, 2015): is a network-based methodology that characterizes known targets in signaling networks by means of topological features. TENET starts computing a set of topological features and then leverages a SVM-based approach to identify predictive topological features that mark known targets. As a result, a computational model is generated and it specifies which topological features are necessary for discriminating the targets and how these elements should be combined to quantify the plausibility of a node being a target. TNET is written in Java feature that makes TNET compatible with Windows, Linux and Mac OSX, and it is freely available to the download at the following web site <https://sites.google.com/site/cosbyntu/software/tenet>.

Table 5 conveys a graphically overview of the features available in each tool listed in this section.

Signal Transduction Pathways Databases

Comprehensive representations of signal transduction pathways have been developed, which requires huge effort and would benefit from information stored in databases and from automatic retrieval and integration methods. Thus, the systematic collection of pathway information in the form of pathway databases is crucial. Here is reported a succinct list of the most known signal transduction pathways databases.

- **NetPath** (Kandasamy *et al.*, 2010): is a manually curated database of signal transduction pathways in humans. NetPath allow users annotation of transcriptionally regulated genes, provides manually curated textual descriptions of each pathway reaction, and data stored in NetPath can be searched by using SPARQL. All pathways stored in NetPath are freely available for download under an adaptive Creative Commons License in BioPAX level 3.0, PSI-MI version 2.5 and SBML version 2.1, Excel, and Tab-delimited formats. NetPath is available at this web site <http://www.netpath.org>.
- **OmniPath** (Türei *et al.*, 2016): is a comprehensive collection of literature curated human signaling pathways. OmniPath main goal is related with the development of an uniform representation of the pathways data, by integrating additional information on the structure and mechanism of the interactions, drug targets, functional annotation. OmniPath aims to provide a ready-to-use web database, as well as an open-source Python module called pypath, which makes pathway analysis easy. OmniPath is freely available at <http://omnipathdb.org>.
- **Reactome** (Croft *et al.*, 2010): is a curated, peer-reviewed resource of human biological processes. Reactome's rationale is to convey the expert consensus view supported by information contained in the literature in a reusable, computationally accessible format. Reactome covers several areas of biology including signaling, immune function, transcriptional regulation,

translation and so on. The Reactome website is reachable at the following web address www.reactome.org. The web site includes a pathway browser and a suite of data analysis tools to support pathway-based analysis of experimental data. Pathways are represented graphically using a Systems Biology Graphical Notation (SBGN)-based representation. Reactome software and data are freely available for download in multiple formats including MySQL, BioPAX, SBML, and PSI-MITAB. Data can also be accessed through the provided Web Services APIs.

- **Signaling Gateway** (Dinasarapu *et al.*, 2011): The UCSD-Nature Signaling Gateway is a comprehensive database of signal transduction pathways. This Gateway represents a collaboration between academia and scientific publishing and is designed to facilitate navigation of the complex world of research into cellular signaling. Signaling Gateway combines expert authored reviews with curated, highly-structured data (e.g., protein interactions) and automatic annotation from publicly available data sources (e.g., UniProt and Genbank). The information and data presented here are freely available to all users. Data can be downloaded in BIOPAX and SBML format. The Signaling Gateway database and related tools are freely available at <http://www.signaling-gateway.org/molecule/>.
- **PANTHER** (Mi *et al.*, 2005): Protein ANALYSIS THrough Evolutionary Relationships is a database of protein families, PANTHER also contains a number of signaling pathways associated with proteins in the database. PANTHER is freely available at the following web address <http://www.pantherdb.org>. PANTHER allows users to download data in SBML and BIOPAX formats.
- **SPIKE** (Paz *et al.*, 2010): Signaling Pathway Integrated Knowledge Engine is a database of highly curated human signaling pathways with an associated interactive software tool. SPIKE interactively graphically displays biological signaling networks, allows dynamic layout and navigation through these networks, and enables the superposition of DNA microarray and other functional genomics data on interaction maps. SPIKE save networks in XGMML format, and allows users to download data in sif and BIOPAX formats. SPIKE is freely available for academic purpose at the following web address <http://www.cs.tau.ac.il/~spike/>.
- **Signalink2** (Fazekas *et al.*, 2013): is an integrated resource to analyze signaling pathway cross-talks, transcription factors, miRNAs and regulatory enzymes. Signalink2 stores data in a MySQL database, where each record in the protein or miRNA tables represents a real protein or miRNA, respectively, and a real entity is represented only by one single record. Proteins could belong to pathways and have topological attributes are represented by means of layers. Allowing users to browse signalling pathways, cross-talks and multi-pathway proteins. Signalink2 data are freely available for download in multiple formats including CSV, BioPAX, SBML, cytoscape, PSI-XML, and PSI-MITAB. Signalink2 is freely available at the following web-site <http://signalink.org>.
- **SIGNOR** (Perfetto *et al.*, 2016): The SIGnaling Network Open Resource (SIGNOR) organizes and stores in a structured format signaling information published in the scientific literature. At the moment of writing, SIGNOR database contains more than 11000 manually-annotated causal relationships between proteins that participate in signal transduction. The entire network can be freely downloaded and used to support logic modeling or to interpret high content datasets. Furthermore, each relationship is connected to the literature reporting the experimental evidence. SIGNOR database is freely available at the follows web address <http://signor.uniroma2.it>. Data can be download in CSV, XLS and casualTAB format.

Table 6 conveys a graphically overview of the features available in each database listed in this section.

Table 6 Signal Transduction Pathways Databases comparison.

Unsupported features are conveyed by black colour, whereas colours convey the supported features for each tool

Name	FFS						O	RU			
	BIOPAX	SBML	SIF	XML	TEXT	Other	HomoSap	Other	AccadOnly	Everyone	AfterRegist
NetPath											
OmniPath											
Reactome											
Signaling Gateway											
PANTHER											
SPIKE											
Signalink2											
SIGNOR											

In the table FFS, File Format Supported indicates the approved file format for loading/exporting; O, Organism refers to the indicates the organism where data come from; RU, Restriction To Use specifies the scope where is allowed to use the tool.

See also: Computational Systems Biology. Computational Systems Biology. Disease Biomarker Discovery. Investigating Metabolic Pathways and Networks. Natural Language Processing Approaches in Bioinformatics. Pathway Informatics

References

- Ackerman, M.J., Clapham, D.E., 1997. Ion channels – Basic science and clinical disease. *New England Journal of Medicine* 336 (22), 1575–1586.
- Agapito, G., Guzzi, P.H., Cannataro, M., 2013. Visualization of protein interaction networks: Problems and solutions. *BMC Bioinformatics* 14 (1), S1.
- Antoni, M.H., Lutgendorf, S.K., Cole, S.W., *et al.*, 2006. The influence of bio-behavioural factors on tumour biology: Pathways and mechanisms. *Nature Reviews Cancer* 6 (3), 240.
- Bacha, J., Brodie, J.S., Loose, M.W., 2009. mygrn: A database and visualisation system for the storage and analysis of developmental genetic regulatory networks. *BMC Developmental Biology* 9 (1), 33.
- Bader, G.D., Cary, M.P., Sander, C., 2006. Pathguide: A pathway resource list. *Nucleic Acids Research* 34 (Suppl. 1), D504–D506.
- Basso, K., Margolin, A.A., Stolovitzky, G., *et al.*, 2005. Reverse engineering of regulatory networks in human b cells. *Nature Genetics* 37 (4), 382.
- Butte, A.J., Tamayo, P., Slonim, D., Golub, T.R., Kohane, I.S., 2000. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences* 97 (22), 12182–12186.
- Caspi, R., Foerster, H., Fulcher, C.A., *et al.*, 2007. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research* 36 (Suppl. 1), D623–D631.
- Cerami, e.g., Gross, B.E., Demir, E., *et al.*, 2010. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Research* 39 (Suppl. 1), D685–D690.
- Chang, C., Ding, Z., Hung, Y.S., Fung, P.C.W., 2008. Fast network component analysis (fastnca) for gene regulatory network reconstruction from microarray data. *Bioinformatics* 24 (11), 1349–1358. Available at: <https://doi.org/10.1093/bioinformatics/btn131>.
- Chen, T., He, H.L., Church, G.M., *et al.*, 1999. Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing* 4, 40.
- Chua, H.E., Bhowmick, S.S., Tucker-Kellogg, L., Dewey Jr, C.F., 2015. Tenet: Topological feature-based target characterization in signalling networks. *Bioinformatics* 31 (20), 3306–3314.
- Cooper, G.F., Herskovits, E., 1992. A bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9 (4), 309–347.
- Coussens, L., 1985. Tyrosine kinase receptor with extensive homology to egf receptor shares chromosomal location with neu oncogene. *Science* 23, 1132–1140.
- Croft, D., O’Kelly, G., Wu, G., *et al.*, 2010. Reactome: A database of reactions, pathways and biological processes. *Nucleic Acids Research* 39 (Suppl. 1), D691–D697.
- Dai, X., Li, J., Liu, T., Zhao, P.X., 2015. Hrgn: A graph search-empowered integrative database of arabidopsis signaling transduction, metabolism and gene regulation networks. *Plant and Cell Physiology* 57 (1), e12.
- Davidson, E., Levin, M., 2005. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America* 102 (14), 4935. Available at: <http://www.pnas.org/content/102/14/4935.short>.
- De Jong, H., Gouz’e, J.-L., Hernandez, C., *et al.*, 2004. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology* 66 (2), 301–340.
- Deller, M.C., Jones, E.Y., 2000. Cell surface receptors. *Current Opinion in Structural Biology* 10 (2), 213–219.
- de Matos Simoes, R., Dehmer, M., Emmert-Streib, F., 2013. Interfacing cellular networks of *S. cerevisiae* and *E. coli*: Connecting dynamic and genetic information. *BMC Genomics* 14 (1), 324.
- Demir, E., Cary, M.P., Paley, S., *et al.*, 2010. The biopax community standard for pathway data sharing. *Nature Biotechnology* 28 (9), 935–942.
- Dinasarapu, A.R., Saunders, B., Ozerlat, I., Azam, K., Subramaniam, S., 2011. Signaling gateway molecule pages a data model perspective. *Bioinformatics* 27 (12), 1736–1738.
- Elf, J., Li, G.-W., Xie, X.S., 2007. Probing transcription factor dynamics at the single-molecule level in a living cell. *Science* 316 (5828), 1191–1194.
- Emmert-Streib, F., Dehmer, M., Haibe-Kains, B., 2014. Gene regulatory networks and their applications: Understanding biological and medical problems in terms of networks. *Frontiers in Cell and Developmental Biology* 2.
- Erhard, F., Friedel, C.C., Zimmer, R., 2008. Fern – A java framework for stochastic simulation and evaluation of reaction networks. *BMC Bioinformatics* 9 (1), 356.
- Faeder, J.R., Blinov, M.L., Hlavacek, W.S., 2009. Rule-Based Modeling of Biochemical Systems with BioNetGen. Totowa, NJ: Humana Press, pp. 113–167. Available at: https://doi.org/10.1007/978-1-59745-525-1_5.
- Faur’e, A., Naldi, A., Chaouiya, C., Thieffry, D., 2006. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22 (14), e124–e131.
- Fazekas, D., Koltai, M., Türei, D., *et al.*, 2013. Signaling 2 – A signaling pathway resource with multi-layered regulatory networks. *BMC Systems Biology* 7 (1), 7. Available at: <https://doi.org/10.1186/1752-0509-7-7>.
- Glass, L., Kauffman, S.A., 1973. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology* 39 (1), 103–129.
- Goto, S., Okuno, Y., Hattori, M., Nishioka, T., Kanehisa, M., 2002. Ligand: Database of chemical compounds and reactions in biological pathways. *Nucleic Acids Research* 30 (1), 402–404.
- Gumbiner, B.M., 1996. Cell adhesion: The molecular basis of tissue architecture and morphogenesis. *Cell* 84 (3), 345–357.
- Han, H., Shim, H., Shin, D., *et al.*, 2015. Trnst: A reference database of human transcriptional regulatory interactions. *Scientific Reports* 5, 11432.
- Haynes, B.C., Brent, M.R., 2009. Benchmarking regulatory network reconstruction with grendel. *Bioinformatics* 25 (6), 801–807.
- Hecker, M., Lambeck, S., Toepper, S., Van Someren, E., Guthke, R., 2009. Gene regulatory network inference: Data integration in dynamic models a review. *Biosystems* 96 (1), 86–103.
- Hla, T., 2003. Signaling and biological actions of sphingosine 1-phosphate. *Pharmacological Research* 47 (5), 401–407.
- Hoops, S., Sahle, S., Gauges, R., *et al.*, 2006. Copasi a complex pathway simulator. *Bioinformatics* 22 (24), 3067–3074.
- Joo, J., Plimpton, S., Martin, S., Swiler, L., Faulon, J.-L., 2007. Sensitivity analysis of a computational model of the ikk-nf- κ b- $\text{I}\kappa$ B- $\text{a}20$ signal transduction network. *Annals of the New York Academy of Sciences* 1115 (1), 221–239.
- Kandasamy, K., Mohan, S.S., Raju, R., *et al.*, 2010. Netpath: A public resource of curated signal transduction pathways. *Genome Biology* 11 (1), R3. Available at: <https://doi.org/10.1186/gb-2010-11-1-r3>.
- Kanehisa, M., Goto, S., 2000. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Karlebach, G., Shamir, R., 2008. Modelling and analysis of gene regulatory networks. *Nature Reviews. Molecular Cell Biology* 9 (10), 770.
- Kim, S., Imoto, S., Miyano, S., 2004. Dynamic bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems* 75 (1), 57–65.
- Kotlyar, M., Pastrello, C., Sheahan, N., Jurisica, I., 2016. Integrated interactions database: Tissue-specific view of the human and model organism interactomes. *Nucleic Acids Research* 44 (D1), D536–D541.
- Lee, I., Blom, U.M., Wang, P.I., Shim, J.E., Marcotte, E.M., 2011. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Research* 21 (7), 1109–1121.
- Leticun, I., Yamada, T., Kanehisa, M., Bork, P., 2008. iPath: Interactive exploration of biochemical pathways and networks. *Trends in Biochemical Sciences* 33 (3), 101–103.

- Liang, S., Fuhrman, S., Somogyi, R., 1998. Reveal, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures.
- Liu, Z.-P., Wu, C., Miao, H., Wu, H., 2015. Regnetwork: An integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. Database, bav095.
- Margolin, A.A., Nemenman, I., Basso, K., *et al.*, 2006. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7 (1), S7.
- Matsuno, H., Doi, A., Nagasaki, M., Miyano, S., 2000. Hybrid petri net representation of gene regulatory network. Pacific Symposium on Biocomputing, 5. Singapore: World Scientific Press, p. 87.
- Mi, H., Lazareva-Ulitsky, B., Loo, R., *et al.*, 2005. The panther database of protein families, subfamilies, functions and pathways. Nucleic Acids Research 33 (Suppl. 1), D284–D288.
- Mi, H., Thomas, P., 2009. Panther pathway: An ontology-based pathway database coupled with data analysis tools. Protein Networks and Pathway Analysis. 123–140.
- Milligan, G., 1993. Mechanisms of multifunctional signalling by G protein-linked receptors. Trends in Pharmacological Sciences 14 (6), 239–244.
- Nagasaki, M., Saito, A., Jeong, E., *et al.*, 2010. Cell illustrator 4.0: A computational platform for systems biology. Silico Biology 10 (1, 2), 5–26.
- Pache, R.A., Aloy, P., 2012. A novel framework for the comparative analysis of biological networks. PLoS One 7 (2), e31220.
- Pastrello, C., Otasek, D., Fortney, K., *et al.*, 2013. Visual data mining of biological networks: One size does not fit all. PLoS Computational Biology 9 (1), e1002833.
- Paz, A., Brownstein, Z., Ber, Y., *et al.*, 2010. Spike: A database of highly curated human signaling pathways. Nucleic Acids Research 39 (Suppl. 1), D793–D799.
- Perfetto, L., Briganti, L., Calderone, A., *et al.*, 2016. Signor: A database of causal relationships between biological entities. Nucleic Acids Research 44 (D1), D548–D554.
- Rahmati, S., Abovsky, M., Pastrello, C., Jurisica, I., 2017. Pathdip: An annotated resource for known and predicted human gene-pathway associations and pathway enrichment analysis. Nucleic Acids Research 45 (D1), D419–D426.
- Ray, L.B., 1999. The science of signal transduction. Science 284 (5415), 755–756.
- Ridley, A.J., 2011. Life at the leading edge. Cell 145 (7), 1012–1022.
- Romero, P., Wagg, J., Green, M.L., *et al.*, 2004. Computational prediction of human metabolic pathways from the complete human genome. Genome Biology 6 (1), R2.
- Ross, I.L., Browne, C.M., Hume, D.A., 1994. Transcription of individual genes in eukaryotic cells occurs randomly and infrequently. Immunology & Cell Biology 72 (2), 101–109.
- Rost, B., Sander, C., 1993. Prediction of protein secondary structure at better than 70% accuracy. Journal of Molecular Biology 232 (2), 584–599.
- Roy, A., Kucukural, A., Zhang, Y., 2010. I-TASSER: A unified platform for automated protein structure and function prediction. Nature Protocols 5 (4), 725.
- Sahle, S., Gauges, R., Pahle, J., *et al.*, 2006. Simulation of biochemical networks using copasi: A complex pathway simulator. In: Proceedings of the 38th conference on Winter simulation. Winter Simulation Conference, pp. 1698–1706.
- Sakamoto, E., Iba, H., 2001. Inferring a system of differential equations for a gene regulatory network by using genetic programming. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1. IEEE, pp. 720–726.
- Sauro, H., 2000. Jarnac: An interactive metabolic systems language in computation in cells. In: Proceedings of an EPSRC emerging computing paradigms workshop. Dept. of Computer Science Technical Report No. 345, University of Hertfordshire.
- Schaefer, C.F., Anthony, K., Krupa, S., *et al.*, 2008. Pid: The pathway interaction database. Nucleic Acids Research 37 (Suppl. 1), D674–D679.
- Schreiber, F., 2002. High quality visualization of biochemical pathways in biopath. Silico Biology 2 (2), 59–73.
- Schütz, G.J., Schindler, H., Schmidt, T., 1997. Single-molecule microscopy on model membranes reveals anomalous diffusion. Biophysical Journal 73 (2), 1073–1080.
- Second, U., 2005. Uviz: A software tool for exploring metabolic pathways in 3-d space. BioTechniques 38 (4), 540–542.
- Shea, M.A., Ackers, G.K., 1985. The λ control system of bacteriophage lambda: A physical-chemical model for gene regulation. Journal of Molecular Biology 181 (2), 211–230.
- Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W., 2002a. Probabilistic boolean networks. Bioinformatics 18 (2), 261–274.
- Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W., 2002b. Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks. Bioinformatics 18 (2), 261–274.
- Shmulevich, I., Dougherty, E.R., Zhang, W., 2002c. From boolean to probabilistic boolean networks as models of genetic regulatory networks. Proceedings of the IEEE 90 (11), 1778–1792.
- Spear, P.G., Eisenberg, R.J., Cohen, G.H., 2000. Three classes of cell surface receptors for alphaherpesvirus entry. Virology 275 (1), 1–8.
- Sreenivasiah, P.K., Rani, S., Cayetano, J., Arul, N., Kim, D.H., 2012. Ipav: Integrated pathway resources, analysis and visualization system. Nucleic Acids Research 40 (D1), D803–D808. Available at: <https://doi.org/10.1093/nar/gkr1208>.
- Thomas, R., 1973. Boolean formalization of genetic control circuits. Journal of Theoretical Biology 42 (3), 563–585.
- Türei, D., Korcsmaros, T., Saez-Rodriguez, J., 2016. Omnipath: Guidelines and gateway for literature-curated signaling pathway resources. Nature Methods 13 (12), 966–967.
- Vanderbilt, J.N., Miesfeld, R., Maler, B.A., Yamamoto, K.R., 1987. Intracellular receptor concentration limits glucocorticoid-dependent enhancer activity. Molecular Endocrinology 1 (1), 68–74.
- Vijesh, N., Chakrabarti, S.K., Sreekumar, J., 2013. Modeling of gene regulatory networks: A review. Journal of Biomedical Science and Engineering 6 (02), 223.
- Villger, A.C., Pettifer, S.R., Kell, D.B., 2010. Arcadia: A visualization tool for metabolic pathways. Bioinformatics 26 (11), 1470–1471. Available at: <https://doi.org/10.1093/bioinformatics/btq154>.
- Watson, S.P., Arkinstall, S., 1994. G-Protein Linked Receptor Factsbook. Academic Press.
- Wittig, U., Kania, R., Golebiewski, M., *et al.*, 2011. Sabio-rk-database for biochemical reaction kinetics. Nucleic Acids Research 40 (D1), D790–D796.
- Wuensche, A., 2010. Ddlab—Discrete Dynamics Lab.
- Xu, R., Venayagamoorthy, G.K., Wunsch, D.C., 2007. Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. Neural Networks 20 (8), 917–927.
- Yamada, T., Letunic, I., Okuda, S., Kanehisa, M., Bork, P., 2011. iPath2.0: Interactive pathway explorer. Nucleic Acids Research 39 (suppl_2), W412–W415.
- Zhang, F., Liu, R., Zheng, J., 2016. Sig2grn: A software tool linking signaling pathway with gene regulatory network for dynamic simulation. BMC Systems Biology 10 (4), 123. Available at: <https://doi.org/10.1186/s12918-016-0365-1>.
- Zhang, M., 2008. Transcriptional Regulatory Element Database.
- Zheng, J., Zhang, D., Przytycki, P.F., *et al.*, 2009. Simboolnet a cytoscape plugin for dynamic simulation of signaling networks. Bioinformatics 26 (1), 141–142.
- Zi, Z., Zheng, Y., Rundell, A.E., Klipp, E., 2008. Sbml-sat: A systems biology markup language (sbml) based sensitivity analysis tool. BMC Bioinformatics 9 (1), 342.
- Zou, M., Conzen, S.D., 2004. A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics 21 (1), 71–79.

Relevant Websites

www.opengl.org
Open GL.
www.wxwidgets.org
Wx Widgets.

Biological Pathway Data Formats and Standards

Ramakanth C Venkata and Dario Gherzi, University of Nebraska at Omaha, Omaha, NE, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

The rapid increase, over the last two decades, in pathway data availability can be partly attributed to high-throughput technologies that are capable of rapidly generating massive amounts of data. The costs associated with these technologies have decreased dramatically in the post human genome project era, thereby enabling and accelerating their widespread use (Marx, 2013). As a result, the number of resources for pathway data representation has also increased. In 2013, the number of online resources for pathways and molecular interactions was 547 (pathguide.org), compared to 325 in the year 2010.

Inconsistent semantics and lack of standardization for representing data make it difficult to combine these resources, resulting in costly generation of possibly redundant data, complicating data integration, and causing time delays and lost opportunities. Developing and maintaining consensus regarding semantics and methods for data representation helps to minimize duplication of efforts. Therefore, well defined, shared data languages are critical for data integration and for facilitating knowledge sharing.

Developing one universal pathway data standard might not be feasible due to the inherent diversity associated with the data types and their uses in different branches of biology. For example, pathway data can include among other things information about metabolic pathways, gene regulation, protein interactions, signaling, and gene interactions. A universal data standard would need to consider and incorporate semantics and terminologies used to represent the abstractions in all these contexts (Cary *et al.*, 2005).

However, standardization is necessary to improve consistency and data integration. To this end, community-wide efforts have resulted in a few data formats and standards that have become popular, and are associated with widely used pathway resources, such as Reactome, Panther, KEGG, and Pathway Commons. This article offers a short description of some of these formats, highlighting similarities and differences between them.

Data Formats

BioPAX

BioPAX is a language designed to systematize, standardize, and facilitate the process of pathway mapping, and is the work of several groups. BioPAX is focused on the representation of cellular and molecular pathways and networks, and it facilitates the organization, analysis, and visualization of biological networks and their associated pathways (Strömbläck and Lambrix, 2005).

BioPAX offers a controlled vocabulary (an ontology) for representing aspects of biological pathways related to molecules (including small molecules), genes, gene interactions, events and states (e.g., post-translational modifications such as phosphorylation).

To facilitate data exchange, BioPAX uses OWL (Web Ontology Language) (Shi, 2007). The ontology consists of three classes that include: (1) **Pathway**; (2) **Interaction**; and (3) **Gene and PhysicalEntity**. Depending on the context, the classes can be further subdivided. For instance, BioPax makes available interaction subclasses (e.g., **GeneInteraction**, **MolecularInteractions**) to unambiguously represent the relationships or associations among gene or physical entity classes.

BioPAX uses a validator with a predefined set of rules to look for errors and inconsistencies (Rodchenkov *et al.*, 2013), and it enables the user to include cross-references to other databases, as well as to use standard biomedical vocabularies such as Gene Ontology (Ashburner *et al.*, 2000) and Proteomics Standard Initiative (PSI-MI, discussed below). Apart from representing functional and biological information, the ontology classes are also used to represent references to other sources and annotations (Demir *et al.*, 2010).

BioPAX level 3 is the current version of the biological pathway exchange format and is intended to replace older BioPAX levels. Level 3 makes substantial changes to the level 2 format and semantics. It adds elements such as physical entities, gene regulation, and genetic interactions to broaden the scope of the language. These additional elements allow more explicit representation of complex biological processes. For example, a newly included class in Level 3 called "Degradation" supports the representation of degradation or conversion from one form to another of **PhysicalEntity** (e.g., proteins).

BioPAX is supported by many software programs, and can be used for various purposes, including developing new tools. For example, it has been used for the development of classifiers and simulation models (Fanizzi *et al.*, 2008; Haydarlou *et al.*, 2016). BioPAX-enabled open source programs can be modified or customized to add new features. The Java library, Paxtools (Demir *et al.*, 2013), provides a Domain Object Model (DOM) that enables the use of BioPAX data elements by software developers for customized development of tools. It allows the conversion of data objects and pathway models to different data formats (including different BioPAX levels). Paxtools also internally validates these customized models (Demir *et al.*, 2013). The BioPAX validator allows customization of rules to perform quality checks on more than 100 classes. The report of the validation checks is provided in a human readable format (HTML or XML). The validator looks for common syntactic and semantic inconsistencies and assists in automatic correction of some of these errors. It also normalizes the BioPAX models to facilitate data integration (Rodchenkov *et al.*, 2013), by replacing non-standard identifiers with Uniform Resource Identifiers to ensure data consistency.

Further, BioPAX is used for other purposes such as customized ontology development with tools such as Protégé (Musen, 2015), development and maintenance of central pathway data repositories, such as Pathway Commons (Cerami *et al.*, 2011), and visualization of pathway maps in Cytoscape (Shannon *et al.*, 2003).

Proteome Standards Initiative-Molecular Interactions (PSI-MI) XML

PSI-MI XML 1.0 was proposed and developed by the Human Proteome Organization (HUPO) Proteomics Standard Initiative in 2004 as a solution for the inconsistent data representation practices followed by different databases (Orchard *et al.*, 2012). Although the first version focused only on protein interactions, and did not provide support for numerical parameters, these features were incorporated into later versions. It is currently updated and maintained by the Molecular Interaction workgroup of HUPO PSI. Around 65 interaction data resources support the PSI-MI format (see Relevant Website section).

PSI-MI data is also available in MITAB format (Kerrien *et al.*, 2007) (part of the PSI-MI standard), which allows convenient access to interaction data in tabular format, for use with spreadsheet software. The MITAB format represents binary interactions, where each row contains one pair of interactors and additional columns with other relevant aspects of the interaction (Kerrien *et al.*, 2007).

The root element of the molecular interaction format schema of PSI-MI XML 2.5 is **entrySet**, which can in turn contain one or more entry elements. The entry is the core element that describes one or more interactions, along with the related data as a self-contained unit. An entry element consists of: (1) a **source** element, which describes the source or origin of the data present in the entry; (2) an **availabilityList** element, which contains data availability statements (e.g., copyright information); (3) **experimentList**, which describes the type of experiments by which interactions of this entry were determined; (4) **interactorList**, which lists all the interactors that occur in the entry; (5) **interactionList**, which includes all the interactions belonging to the entry; and (6) **attributeList**, a semi-structured container for other data pertaining to the entry, such as attribute name or reference to external, controlled vocabularies (Kerrien *et al.*, 2007). It is important to note that the format accommodates the representation of nucleic acid interactions and other types of interactions, and both binary interactions and complexes can be represented in the PSI-MI XML format.

PSI-MI uses controlled vocabularies for consistency and to avoid ambiguity. The Open Biomedical Ontologies Foundry (OBO) (Smith *et al.*, 2007) maintains PSI-MI's controlled vocabulary with the namespace prefix "MI". The vocabulary is updated regularly to keep up with evolving technologies.

Efforts in the systems biology community have led to the development of several tools that support the PSI-MI format, including XML parsers for incorporating the format in other tools, format converters to covert PSI-MI XML into BioPAX or HTML, and semantic validators to tackle the problem of data consistency. Cytoscape (Shannon *et al.*, 2003) supports the PSI-MI XML format for visualizing molecular interaction data without requiring additional plug-ins. Examples of data providers that use the PSI-MI XML and MITAB formats can be found in the International Molecular Exchange Consortium (IMEx) (Orchard *et al.*, 2012), and the BioGRID interaction data repository (Stark, 2006).

Systems Biology Markup Language (SBML)

SBML was developed with the intent of providing support for the mathematical modeling of biological systems. SBML models describe the behavior of biological entities and the underlying biochemical processes by taking into account their temporal aspects (Hucka *et al.*, 2015). SBML does not attempt to be a universal language. Rather, its goal is to efficiently represent biological models in a machine-readable format that can facilitate their analysis and dissemination, thus mitigating the issue of incompatibility between different software.

The development of SBML is an ongoing process; different editions are termed levels. Although older level models can be converted into more recent levels through translation, the reverse is not always possible. The latest SBML specification is level 3, and it consists of set of core features that can be extended to incorporate additional features through optional packages.

Representing biological models is inherently challenging, as models often need to incorporate molecular features, concentrations, temporal variables, interactions, and context-dependent events. SBML provides ways to represent these aspects of quantitative modeling. Data objects are used to represent each type of component of a model, along with related information. An overview of SBML core model components is provided below:

1. **Function definitions** (user-defined functions).
2. **Unit definitions** (measurement units).
3. **Compartment** (enclosed space where species are located).
4. **Species** (group of entities sharing similar properties).
5. **Parameters** (used to define symbols and their associated values).
6. **Initial assignments** (initial values for model entities).
7. **Rules** (mechanisms to define the values of variables in a model and to determine their dynamic behavior).
8. **Constraints** (statements that define allowed values for quantities).
9. **Reactions** (processes that can alter the quantities of one or more species).
10. **Events** (state changes in the model that can instantaneously alter quantities in a context-dependent fashion).

Annotation of SBML objects should conform to MIRIAM (Minimum Information Required In the Annotation of Models) guidelines (Novère *et al.*, 2005). URIs (Uniform Resource Identifiers) are used to identify external resources and data used for annotating SBML models and elements. SBML uses XML to encode the models, whereas previous levels of the language used UML to represent data objects (Hucka *et al.*, 2010). In terms of model sharing, the European Bioinformatics Institute (EBI) hosts the BioModels Database (Novère, 2006), a repository with an interactive web interface for storing, retrieving, and exchanging SBML models.

Systems Biology Graphical Notation (SBGN)

SBGN is a collaborative community effort to develop a standard graphical notation to assist in the unambiguous representation of biological or biochemical interactions (Novère *et al.*, 2009). The SBGN language is stratified into levels. A level is organized in such a way as to provide a particular set of functionality to address a specific set of tasks. SBGN's graphical notation schema includes a defined controlled vocabulary, but it also allows the user to take advantage of other controlled vocabularies. However, users are required to include the controlled vocabulary term definitions in the text or figure legend.

SBGN consists of three languages that complement each other: (1) process; (2) entity-relationship; and (3) activity flow diagrams. The rationale behind this arrangement is to separate different properties of biological entities, to reduce complexity, and to improve abstraction. A short description of the three languages making SBGN follows below:

- Process Description Diagram is the representation of all molecular or biochemical interactions and processes involving biological entities. This type of diagram also captures information regarding temporal influences on the processes. This diagram includes two types of nodes: one for entities such as proteins and genes (Entity nodes), and the other for processes like associations and reactions (Process nodes) (Moodie *et al.*, 2015).
- Entity-Relationship Diagram describes the influence of entities on each other and thus each entity is only represented once. The edges represent the positive and negative influences and directionality among entities. Unlike process diagrams, entity-relationship diagrams can better capture the details of the state (e.g., active state) of the entities, but they lack the representation of temporal influences on the states (Sorokin *et al.*, 2015).
- Activity Flow Diagrams are similar to the diagrams observed in tertiary literature (e.g., textbooks and handbooks). Contrary to previous versions, in level 1 version 1.2 the activity flow language uses edges to provide information regarding the positive or negative influence among the entities in the pathways (Mi *et al.*, 2015).

KEGG Markup Language (KGML)

KGML is a standard language used for the KEGG databases (Ogata *et al.*, 1999) and is also the default exchange format used to represent KEGG graph objects (Kanehisa *et al.*, 2004). KGML aids in manual and automatic illustrations of biochemical pathways, and it also enables network modeling and supports computational pathway analysis. Manually drawn representations are used for reference pathways, whereas organism specific pathways are computationally derived from reference pathways. Reference pathways are manually drawn using available scientific evidence, and this pathway information is then computationally assigned to other organisms that share the same pathway. KGML files contain graphical objects that are used to represent entities and their associations (e.g., biochemical reactions) in the specific KEGG pathways. The files also include information on orthologs. The KGML pathway element defines graph objects, where the entry element is a node and the reaction or relation elements are edges. The relation elements are the connections between the gene products and are observed in protein networks, whereas the reaction elements are the connections between the biochemical entities as seen in chemical networks. Metabolic pathways are made up of either protein, or chemical, or both kinds of networks, and regulatory pathways mostly involve protein networks.

Other Languages

Other markup languages for biological pathways include BCML (Biological Connection Markup Language), an XML language based on SBGN standards that attempts to dynamically represent signaling pathways (Beltrame *et al.*, 2011). CellML (Beard *et al.*, 2009), another XML markup language, shares many similarities with SBML but differs in its scope and semantics. CellML places more emphasis on the mathematical modeling and simulation of cellular models, compared to SBML.

Closing Remarks

This article has given a brief overview of a few widely used languages for representing biological pathways, and it is by no means a comprehensive list of all standards and formats that have been developed over the years. The choice of a particular language is dependent on the problem at hand, as specific languages are better suited to particular domains than others. For example, PSI-MI is the language of choice for representing protein-protein interactions, whereas SBML is specifically designed to allow the unambiguous representation of mathematical models.

Acknowledgments

This work was partly supported by a Nebraska Research Initiative grant to DG.

See also: Biological Pathways. Computational Systems Biology. Natural Language Processing Approaches in Bioinformatics. Pathway Informatics

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The [G]ene [O]ntology consortium. *Nature Genetics* 25 (1), 25–29.
- Beard, D.A., Britten, R., Cooling, M.T., *et al.*, 2009. Cellml metadata standards, associated tools and repositories. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 367 (1895), 1845–1867.
- Beltrame, L., Calura, E., Popovici, R.R., *et al.*, 2011. The biological connection markup language: A SBGN-compliant format for visualization, filtering and analysis of biological pathways. *Bioinformatics* 27 (15), 2127–2133. doi:10.1093/bioinformatics/btr339. (ISSN 13674803).
- Cary, M.P., Bader, G.D., Sander, C., 2005. Pathway information for systems biology. *FEBS Letters* 579 (8), 1815–1820. doi:10.1016/j.febslet.2005.02.005. (ISSN 00145793).
- Cerami, E.G., Gross, B.E., Demir, E., *et al.*, 2011. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Research* 39 (SUPPL. 1), doi:10.1093/nar/gkq1039. (ISSN 03051048).
- Demir, E., Cary, M.P., Paley, S., *et al.*, 2010. The BioPAX community standard for pathway data sharing. *Nature Biotechnology* 28 (12), 1308. doi:10.1038/nbt1210-1308c. [1308].
- Demir, E., Babur, Ö., Rodchenkov, I., *et al.*, 2013. Using biological pathway data with paxtools. *PLoS Computational Biology* 9 (9), doi:10.1371/journal.pcbi.1003194. (ISSN 1553734X).
- Fanizzi, N., D'Amato, C., Esposito, F., 2008. Induction of classifiers through non-parametric methods for approximate classification and retrieval with ontologies. *International Journal of Semantic Computing* 2 (3), 403. doi:10.1142/S1793351X0800049X. (ISSN 1793-351X).
- Haydarlou, R., Jacobsen, A., Bonzanni, N., *et al.*, 2016. BioASF: A framework for automatically generating executable pathway models specified in BioPAX. *Bioinformatics* 32 (12), i60–i69. doi:10.1093/bioinformatics/btw250. (ISSN 14602059).
- Hucka, M., Bergmann, F.T., Hoops, S., *et al.*, 2010. The systems biology markup language (sbml): Language specification for level 3 version 1 core. *Nature Precedings*. Available at: <https://doi.org/10.1038/npre.2010.4123.1a>.
- Hucka, M., Bergmann, F.T., Dräger, A., *et al.*, 2015. Systems biology markup language (SBML) level 2 version 5: Structures and facilities for model definitions. *Journal of Integrative Bioinformatics* 12 (2), doi:10.2390/biecoll-jib-2015-271.
- Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., Hattori, M., 2004. The kegg resource for deciphering the genome. *Nucleic Acids Research* 32 (suppl_1), D277–D280.
- Kerrien, S., Orchard, S., Montecchi-Palazzi, L., *et al.*, 2007. Broadening the horizon level 2.5 of the HUP0-PSI format for molecular interactions. *BMC Biology* 5 (1), 44. doi:10.1186/1741-7007-5-44. (ISSN 1741-7007).
- Marx, V., 2013. Biology: The big challenges of big data. *Nature* 498 (7453), 255–260. doi:10.1038/498255a. (ISSN 0028-0836).
- Mi, H., Schreiber, F., Moodie, S., *et al.*, 2015. Systems biology graphical notation: Activity flow language level 1 version 1.2. *Journal of Integrative Bioinformatics* 12 (2), 265. doi:10.2390/biecoll-jib-2015-265. (ISSN 1613-4516 (Electronic)).
- Moodie, S., Le Novère, N., Demir, E., Mi, H., Villéger, A., 2015. Systems biology graphical notation: Process description language level 1 version 1.3. *Journal of Integrative Bioinformatics* (JIB) 12 (2), 213–280. doi:10.2390/BIECOLL-JIB-2015-263. (ISSN 16134516).
- Musen, M.A., 2015. The Protégé project: A look back and a look forward. *AI Matters* 1 (4), 4–12. doi:10.1145/2757001.2757003. [ISSN 2372-3483 (Electronic)].
- Novère, N.L., 2006. BioModels database, a curated resource of annotated published models. *Bioinformatics* 691, 2006.
- Novère, N.L., Finney, A., Hucka, M., *et al.*, 2005. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature Biotechnology* 23 (12), 1509–1515. doi:10.1038/nbt1156. (ISSN 1087-0156).
- Novère, N.L., Hucka, M., Mi, H., *et al.*, 2009. The systems biology graphical notation: Abstract. *Nature Biotechnology*. *Nature biotechnology* 27 (8), 735–742. doi:10.1038/nbt1558.
- Ogata, H., Goto, S., Sato, K., *et al.*, 1999. KEGG: Kyoto encyclopedia of genes and genomes. *ISSN* 03051048.
- Orchard, S., Kerrien, S., Abbani, S., *et al.*, 2012. Protein interaction data curation –The International Molecular Exchange Consortium (IMEx). *Nature Methods* 161819 (94), 345–350. doi:10.1038/nmeth.1931. (ISSN 1548-7105).
- Rodchenkov, I., Demir, E., Sander, C., Bader, G.D., 2013. The BioPAX validator. *Bioinformatics* 29 (20), 2659–2660. doi:10.1093/bioinformatics/btt452. (ISSN 13674803).
- Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research* 13 (11), 2498–2504. doi:10.1101/gr.1239303. (ISSN 1088905).
- Shi, Y., 2007. In: *Computational Science-ICCS 2007: Proceedings of the 7th International Conference, Beijing, China, May 27–30, 2007, Proceedings, vol. 3. Springer Science & Business Media*.
- Smith, B., Ashburner, M., Rosse, C., *et al.*, 2007. The OBO foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25 (11), 1251–1255. doi:10.1038/nbt1346. (ISSN 1087-0156).
- Sorokin, A., Le Novère, N., Luna, A., *et al.*, 2015. Systems biology graphical notation: Entity relationship language level 1 version 2. *Journal of Integrative Bioinformatics* 12 (2), 264. doi:10.2390/biecoll-jib-2015-264. (ISSN 1613-4516 (Electronic)).
- Stark, C., 2006. BioGRID: A general repository for interaction datasets. *Nucleic Acids Research* 34 (90001), D535–D539. doi:10.1093/nar/gkj109. (ISSN 0305-1048).
- Striömbäck, L., Lambrix, P., 2005. Representations of molecular pathways: An evaluation of sbml, psi mi and biopax. *Bioinformatics* 21 (24), 4401–4407.

Relevant Website

www.pathguide.org
Pathguide Technologies.

Biographical Sketch

Ramakanth Chirravuri Venkata, Pharm.D. Ramakanth Chirravuri Venkata is currently a Biomedical Informatics Master's student at the University of Nebraska at Omaha. He completed his Doctor of Pharmacy degree from Osmania University, India in 2015. His research interests include computational cancer biology, cancer pharmacology and therapeutics.

Dario Ghersi, M.D., PhD Dario Ghersi received his M.D. in 2004 from the University of Genoa, Italy, with a thesis on a mathematical model of the immune system. He then pursued a PhD in Computational Biology at Mount Sinai/New York University. From 2010 to 2014 he trained as a Postdoctoral Fellow at Princeton University, developing network-based and structural approaches to probe protein function. In 2011, he was awarded an American-Italian Cancer Foundation fellowship to develop computational methods for the analysis of cancer mutations. He is currently on the faculty of the School of Interdisciplinary Informatics at the University of Nebraska at Omaha. His research interests focus on cancer informatics, immunoinformatics, network biology, and structural bioinformatics.

Biological Pathway Analysis

Ramakanth Chirravuri Venkata and Dario Gherzi, University of Nebraska at Omaha, Omaha, NE, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological pathways can be defined as a set of interactions between biomolecules associated with well-defined cellular changes. Biological pathways are involved in signaling, metabolism, gene expression, and other critical biological processes. Over the years, the systems biology community has made an effort to curate biological pathways using machine readable mark-up languages to improve information exchange, algorithm development, and reusability.

Pathways can be represented using the network formalism, as directed or undirected graphs. Representing pathways as networks is beneficial, as we can take advantage of the many graph-based approaches for data mining, which can extract network features with important biological correlates (Pavlopoulos *et al.*, 2011; You *et al.*, 2006). For example, readily available softwares for network visualization and analysis like Cytoscape (Shannon *et al.*, 2003), Gephi (Bastian *et al.*, 2009), as well as libraries like igraph (Csardi and Nepusz, 2006) can calculate basic network properties including degree centrality (number of edges of a node), betweenness centrality (number of all shortest paths going through a node, used to identify "bottlenecks" in a network), clustering coefficient (which measures the potential of a network to be divided into clusters), network motifs (recurrent patterns in a network), and other measures that are not only interesting from a graph-theoretical perspective but correlate with biological features.

The importance of network-based approaches for pathway analysis is also evident in the way biological pathways are stored. To the more traditional relational model the systems biology community has added graph databases, which allow the visualization, integration, and analysis of heterogeneous data corresponding to different data formats and types (Lysenko *et al.*, 2016). For example, the Neo4j powered graph databases offers support for querying complex disease maps (Yoon *et al.*, 2017).

Specific approaches to analyze biological pathways are discussed below.

Graph Mining Approaches

Graph pattern mining is a broad concept, wherein different measures (known as *interestingness* measures) or functions are considered for different analyses. Several algorithms are available for identifying recurring graph patterns, and for finding significant and discriminative features in the graphs. An important concern associated with graph mining is the combinatorial complexity of the algorithms and the related scalability issues (Cheng *et al.*, 2014).

An important task in biological pathway graph mining is the identification of densely connected subgraphs that are sparsely connected to rest of the network. Density measures such as edge density (average degree) or edge ratio can be used to identify dense sub-graphs. Choice of the algorithms is based on whether the graph is directed or undirected. If the graph is undirected, traditional algorithms such as Greedy or Goldberg (Charikar, 2000; Goldberg, 1984) may be suitable. For directed graphs, maximum-flow based algorithms have been used successfully in the past (Balalau *et al.*, 2015; Khuller and Saha, 2009).

Mining algorithms are formally classified as *a priori* based approaches and *pattern growth* approaches. *A priori* based approaches generally use a breadth-first strategy to identify frequent substructures. In contrast, pattern growth approaches recursively grow a frequent subgraph, using a depth-first exploration strategy, or a combination of depth first search and breadth-first search strategies (Kavitha *et al.*, 2011).

Clustering techniques (hierarchical, partitioning, or random walk based) are used to reduce the complexity of the networks, by grouping similar entities together. The clusters can be visualized with a variety of tools, such as Arena3D (Harel and Koren, 2001; Pavlopoulos *et al.*, 2008). However, if the clusters within the graph are uniform, the clusters produced by clustering algorithms may be arbitrary. For this reason, visual inspection of graphs for patterns and clusters is imperative (Schaeffer, 2007).

Classification of features can be done using either kernel based (e.g., Support Vector Machines) or graph embedding methods. Graph kernels can be used to analyze structural patterns and classification problems. Graph kernels use matrix algebra operations to allow linear representation of the data (Nabhan, 2014). Graph embedding methods are routinely employed to reduce dimensionality of the data, and to facilitate further analysis and representation. Commonly used graph embedding methods are Principal Component Analysis (PCA), which takes into account data variance, and Linear Discriminant Analysis (LDA). Compared to PCA, LDA can achieve better class separation, but it can be severely affected by outliers (Yan *et al.*, 2007). Newer techniques use combinations of two or more of the methods mentioned above (e.g., kernel PCA) (Schölkopf *et al.*, 1997).

Many types of graph mining algorithms are available, with no single algorithm being suitable for all domains. Before selecting an appropriate method for graph mining, we need to be mindful of some of the distinguishing features of biological data: (1) graphs in the biological domain tend to be much larger than graphs observed in other domains; (2) the directionality of the edges is important as it conveys vital information about associations among entities (Aggarwal and Wang, 2010); and (3) heterogeneity and dynamic/temporal features of biological data may complicate data integration (Ozsoyoglu *et al.*, 2006).

Pathway Alignment Approaches

Pathway/network alignment is analogous to sequence alignment, where the goal is to identify similar or conserved regions across different species. The rationale behind identifying conserved regions is that it can be used to uncover functional similarities across species. Traditionally, only protein-protein networks were used for network alignment, but due to the emergence of high-throughput technologies in other domains, most biological networks are now used for alignment and comparison (Bayati *et al.*, 2013).

Pathway alignment approaches focus on the topological similarities that exist between networks and that may provide valuable information on biological function. Network alignment can also complement sequence alignment in identifying orthologs (Singh *et al.*, 2008). As in the case of sequence alignment, network alignment can be global or local. Although global alignment approaches are useful in identifying conserved large subgraphs, they may fail to identify localized network regions that are topologically similar.

In addition, network alignment approaches can be classified as pairwise or multiple, based on the number of networks being compared. Pairwise alignment involves alignment and comparison between two networks, whereas multiple alignments involve more than two networks. These alignments can be further sub-classified based on the type of node mapping in the networks being compared (one-to-one, one-to-many and many-to-many) (Clark and Kalita, 2014; Faisal *et al.*, 2015).

Comparisons across networks can also be performed without alignment, by quantifying their topological properties (e.g., degree distribution). However, this quantification is not suitable for identifying conserved regions across networks (Faisal *et al.*, 2015).

Local Alignment Approaches

Table 1 shows a range of local pairwise and multiple aligners. Local aligners aim to align only part of a network, with some methods handling only pairwise alignment and others returning multiple alignments as well. NetworkBlast (Kalaev *et al.*, 2008) is an extension of an older local aligner called PathBlast, that uses a probabilistic model for comparing multiple alignments (Tian and Samatova, 2009). NetAligner is a web server to align and compare complexes, pathways, or interactomes against interactomes (Pache *et al.*, 2012). MaWISH (Koyutürk *et al.*, 2006) uses a scoring function based on evolutionary models to compare networks. AlignMCL (Mina and Guzzi, 2012) uses Markov models to identify functional modules in the aligned graphs (Meng *et al.*, 2016).

The performance of local alignment approaches can be assessed by quality measures such as semantic similarity score, sensitivity/specificity, and F-score (Faisal *et al.*, 2015).

Global Alignment Approaches

Global network aligners use different strategies and approaches to align and compare networks. In brief, traditional aligners examine the topological similarities using a two step approach involving: (1) the use of a node cost function to identify pairwise node similarities; and (2) an alignment strategy to yield an optimal alignment across networks. Some of the global aligners also consider sequence similarities between the node pairs (e.g., proteins) alongside topological similarities (Meng *et al.*, 2016). Contrary to mapping approaches used by local aligners, global pairwise aligners use one-to-one mapping approach, and global multiple aligners use many-to-many mapping approach, to identify large common regions among the networks being compared. Similar to local alignment, quality measures are used to assess the alignment quality but instead of using only biological measures, topological quality measures are also considered (Crawford *et al.*, 2015; Faisal *et al.*, 2015).

Quality Measures for Global Alignment Methods

Topological quality measures such as node correctness, symmetric substructure score, largest connected common subgraph and number of aligned clusters are employed to assess true node mapping and edge conservation (Faisal *et al.*, 2015; Milano *et al.*, 2017).

Table 1 Local network alignment approaches

Name	Strategy
Pairwise aligners	
NetAligner	Monte Carlo permutation test to assess significance
NetworkBLAST	Combination of a greedy method with dynamic programming
MaWISH	Alignment scoring based on edge weights and cost-function
AlignMCL	Markov clustering algorithm for mining
Multiple aligners	
Graemlin	Progressive alignment
NetworkBLAST-M	Layered alignment

Biological quality measures such as exact protein ratio, normalized mean entropy, exact cluster ratio, Gene Ontology (GO) enrichment and GO semantic similarity are used to evaluate the functional similarities among aligned node pairs and functional consistency in the alignments (Faisal *et al.*, 2015; Panni and Rombo, 2013).

Pathway Analysis in Genome-Wide Association Studies

Genome Wide Association Studies (GWAS) use statistical models to investigate the association between variants (Single Nucleotide Polymorphisms) and phenotypes. Despite the debate regarding statistical significance vs. biological relevance in the scientific community, GWAS have been an important source for generating novel hypotheses in the field of genetics. GWAS tend to be suitable for detecting common variants associated with specific phenotypes (Teslovich *et al.*, 2010; Visscher *et al.*, 2012). However, in the case of complex diseases associating individual variants to specific phenotype can be challenging due to their small effect sizes. Another considerable issue with GWAS is that they use stringent p-value thresholds that are corrected for multiple hypothesis testing, in order to control false positives. This threshold may occasionally get rid of moderately significant genes that may represent biologically plausible candidates for genotype-phenotype associations (Jia *et al.*, 2011).

Pathway analysis may substantially alleviate these problems by examining the combined effects of related single variants, and also by aiding in the identification of related novel variants that are associated with the phenotypes. Pathway analysis examines association with the phenotype of interest using gene sets instead of individual genes (Kao *et al.*, 2016; Zhong *et al.*, 2010). Pathway analysis of GWAS studies generally consists of three steps: (1) determination of gene sets suitable for pathway analysis (e.g., Gene Ontology (Ashburner *et al.*, 2000) or KEGG (Ogata *et al.*, 1999)); (2) mapping variants to their respective genes; and (3) using statistical models to examine associations.

Conclusions

As it is the case for most bioinformatics approaches, biological pathway analysis has benefited from major advancements in high-throughput technologies and increased availability of data. Combined with advancements in computer hardware, high-performance computing, and algorithms in the fields of graph theory and machine learning, biological pathway analysis can now be used to assist in hypothesis generation, to summarize the results of large high-throughput experiments, and to computationally validate existing hypotheses.

Acknowledgments

This work was partly supported by a Nebraska Research Initiative grant to DG.

See also: Biological Pathways. Community Detection in Biological Networks. Computational Systems Biology. Graph Algorithms. Natural Language Processing Approaches in Bioinformatics. Network Centralities and Node Ranking. Network Models. Network Topology. Pathway Informatics

References

- Aggarwal, C.C., Wang, H., 2010. Graph data management and mining: A survey of algorithms and applications. In: *Managing and Mining Graph Data*, pp. 13–68. Springer.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* 25, 25–29. doi:10.1038/75556. ISSN 1061-4036.
- Balalau, O.D., Bonchi, F., Chan, T., Gullo, F., Sozio, M., 2015. Finding subgraphs with maximum total density and limited overlap. In: *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pp. 379–388. ACM.
- Bastian, M., Heymann, S., Jacomy, M., 2009. Gephi: An open source software for exploring and manipulating networks. In: *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*, pp. 361–362. doi: 10.1136/qshc.2004.010033. ISSN 14753898. Available at: <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154%5Cnpapers2://publication/uuid/CCEBC821>.
- Bayati, M., Gleich, D.F., Saberi, A., Wang, Y., 2013. Message-passing algorithms for sparse network alignment. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 7 (1), 3.
- Charikar, M., 2000. Greedy approximation algorithms for finding dense components in a graph. In: *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pp. 84–95. Springer.
- Cheng, H., Yan, X., Han, J., 2014. Mining graph patterns. In: *Frequent Pattern Mining*, pp. 307–338. Springer.
- Clark, C., Kalita, J., 2014. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics* 30 (16), 2351–2359.
- Crawford, J., Sun, Y., Milenković, T., 2015. Fair evaluation of global network aligners. *Algorithms for Molecular Biology* 10 (1), 19.
- Csardi, G., Nepusz, T., 2006. The igraph software package for complex network research. *InterJournal Complex Systems* 1695. Available at: <http://igraph.sf.net>.
- Faisal, F.E., Meng, L., Crawford, J., Milenković, T., 2015. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology* 2015 (1), 3.
- Goldberg, A.V., 1984. *Finding a Maximum Density Subgraph*. Berkeley, CA: University of California.

- Harel, D., Koren, Y., 2001. On clustering using random walks. In: FSTTCS, pp. 18–41. Springer.
- Jia, P., Wang, L., Meltzer, H.Y., Zhao, Z., 2011. Pathway-based analysis of gwas datasets: Effective but caution required. *International Journal of Neuropsychopharmacology* 14 (4), 567–572.
- Kalaei, M., Smoot, M., Ideker, T., Sharan, R., 2008. NetworkBLAST: Comparative analysis of protein networks. *Bioinformatics (Oxford, England)* 24 (4), 594–596. doi:10.1093/bioinformatics/btm630. ISSN 1367-4811.
- Kao, P.Y., Leung, K.H., Chan, L.W., Yip, S.P., Yap, M.K., 2016. Pathway analysis of complex diseases for gwas, extending to consider rare variants, multi-omics and interactions. *Biochimica et Biophysica Acta (BBA)-General Subjects*.
- Kavitha, D., Rao, B.M., Babu, V.K., 2011. A survey on assorted approaches to graph data mining. *International Journal of Computer Applications* 14 (1), 43–46.
- Khuller, S., Saha, B., 2009. On finding dense subgraphs. *Automata, Languages and Programming* 597–608.
- Koyutürk, M., Kim, Y., Topkara, U., *et al.*, 2006. Pairwise alignment of protein interaction networks. *Journal of Computational Biology* 13 (2), 182–199.
- Lysenko, A., Roznovát, I.A., Saqi, M., *et al.*, 2016. Representing and querying disease networks using graph databases. *Bio Data Mining* 9 (1), 23.
- Meng, L., Striegel, A., Milenković, T., 2016. Local versus global biological network alignment. *Bioinformatics* 32 (20), 3155–3164.
- Milano, M., Guzzi, P.H., Tymofieva, O., *et al.*, 2017. An extensive assessment of network alignment algorithms for comparison of brain connectomes. *BMC Bioinformatics* 18 (6), 235.
- Mina, M., Guzzi, P.H., 2012. Alignmcl: Comparative analysis of protein interaction networks through markov clustering. In: *IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, 2012, pp. 174–181. IEEE.
- Nabhan, A.R., 2014. Graph pattern mining techniques to identify potential model organisms. The University of Vermont and State Agricultural College.
- Ogata, H., Goto, S., Sato, K., *et al.*, 1999. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 27, 29–34. doi:10.1093/nar/27.1.29. ISSN 03051048.
- Ozsoyoglu, Z., Ozsoyoglu, G., Nadeau, J., 2006. Genomic pathways database and biological data management. *Animal Genetics* 37 (s1), 41–47.
- Pache, R.A., Céol, A., Aloy, P., 2012. Netaaligner network alignment server to compare complexes, pathways and whole interactomes. *Nucleic Acids Research* 40 (W1), W157–W161.
- Panni, S., Rombo, S.E., 2013. Searching for repetitions in biological networks: Methods, resources and tools. *Briefings in Bioinformatics* 16 (1), 118–136.
- Pavlopoulos, G.A., O'Donoghue, S.I., Satagopam, V.P., *et al.*, 2008. Arena3d: Visualization of biological networks in 3d. *BMC Systems Biology* 2 (1), 104.
- Pavlopoulos, G.A., Secrier, M., Moschopoulos, C.N., *et al.*, 2011. Using graph theory to analyze biological networks. *Bio Data Mining* 4 (1), 10.
- Schaeffer, S.E., 2007. Graph clustering. *Computer Science Review* 1 (1), 27–64.
- Schölkopf, B., Smola, A., Müller, K.-R., 1997. Kernel principal component analysis. In: *International Conference on Artificial Neural Networks*, pp. 583–588. Springer.
- Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software Environment for integrated models of biomolecular interaction networks. *Genome Research* 13 (11), 2498–2504. doi:10.1101/gr.1239303. ISSN 10889051.
- Singh, R., Xu, J., Berger, B., 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences* 105 (35), 12763–12768.
- Teslovich, T.M., Musunuru, K., Smith, A.V., *et al.*, 2010. Biological, clinical, and population relevance of 95 loci for blood lipids. *Nature* 466 (7307), 707.
- Tian, W., Samatova, N.F., 2009. Pairwise alignment of interaction networks by fast identification of maximal conserved patterns. *Pacific Symposium on Biocomputing* 14, 99–110.
- Visscher, P.M., Brown, M.A., McCarthy, M.I., Yang, J., 2012. Five years of gwas discovery. *The American Journal of Human Genetics* 90 (1), 7–24.
- Yan, S., Xu, D., Zhang, B., *et al.*, 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (1), 40–51.
- Yoon, B.-H., Kim, S.-K., Kim, S.-Y., 2017. Use of graph database for the integration of heterogeneous biological data. *Genomics & Informatics* 15 (1), 19–27.
- You, C.H., Holder, L.B., Cook, D.J., 2006. Application of graph-based data mining to metabolic pathways. In: *Data Mining Workshops, 2006. ICDM Workshops 2006. International Conference on Proceedings of the Sixth IEEE*, pp. 169–173. IEEE.
- Zhong, H., Yang, X., Kaplan, L.M., Molony, C., Schadt, E.E., 2010. Integrating pathway analysis and genetics of gene expression for genome-wide association studies. *The American Journal of Human Genetics* 86 (4), 581–591.

Biographical Sketch

Ramakanth Chirravuri Venkata, PharmD, is currently a Biomedical Informatics Master's student at the University of Nebraska at Omaha. He completed his Doctor of Pharmacy degree from Osmania University, India in 2015. His research interests include computational cancer biology, cancer pharmacology and therapeutics.

Dario Gheri, MD, PhD, received his MD in 2004 from the University of Genoa, Italy, with a thesis on a mathematical model of the immune system. He then pursued a PhD in Computational Biology at Mount Sinai/New York University. From 2010–2014 he trained as a Postdoctoral Fellow at Princeton University, developing network-based and structural approaches to probe protein function. In 2011, he was awarded an American-Italian Cancer Foundation fellowship to develop computational methods for the analysis of cancer mutations. He is currently on the faculty of the School of Interdisciplinary Informatics at the University of Nebraska at Omaha. His research interests focus on cancer informatics, immunoinformatics, network biology, and structural bioinformatics.

Two Decades of Biological Pathway Databases: Results and Challenges

Sara Rahmati, University of Toronto, Toronto, ON, Canada and Krembil Research Institute, Toronto, ON, Canada

Chiara Pastrello and Andrea EM Rossos, Krembil Research Institute, Toronto, ON, Canada

Igor Jurisica, University of Toronto, ON, Canada and Slovak Academy of Sciences, Bratislava, Slovakia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biomolecular pathway databases are essential components of bioinformatics workflows, and are integral to understanding disease and health processes. Pathways are usually defined as sets of biomolecules that cooperatively accomplish a specific cellular outcome. They have been defined at different levels of organization, for example, gene sets ([Liberzon et al., 2015](#)), networks of process-specific interactions ([Glaab et al., 2012](#); [Fazekas et al., 2013](#); [Rahmati et al., 2017](#)), or detailed kinetics of molecular events that cause a specific phenotype ([Kanehisa, 2000](#); [Nishimura, 2001](#); [Fabregat et al., 2016](#)). However, the underlying concept across all of these definitions is functional dependency of a set of biomolecules (including proteins) through molecular networks.

Description of these networks is usually defined across different papers, which focus on network components, dynamics, or diverse phenotype-specific differences. They are collected by data curators, organized in pathway maps by experts, and stored in online databases.

Although concept of a pathway as a set of proteins that co-function to accomplish a task (or result in a phenotype) has been around since early 1950s ([Komai, 1950](#); [Barratt et al., 1954](#)), their first formal representation in a knowledge base, EcoCyc (a branch of the current BioCyc group), was introduced in 1993 ([Karp and Paley, 1994](#)). It did not take long for researchers to recognize the significance of these databases in molecular biology research. On December 1, 1995, KEGG ([Kanehisa, 1997](#)) released its first version, Release 0.1, which contained gene classification tables. The first official version (Release 1.0) was introduced on October 24, 1996, and covered five organisms. By incorporating molecular interactions into metabolic gene sets, KEGG introduced the first web hosted pathway diagrams. In July 1997, it added signal transduction and gene regulatory pathways to its metabolic pathways, encompassing the three major categories of pathways for the first time ([Kanehisa, 1997](#)).

Pathway databases and analysis tools are valuable during gene set annotation, especially the sets, which are obtained from high-throughput (HTP) studies. Predominantly, the goal is to understand which genes play the most central, functionally important role in a specific phenotype such as a disease state or response to a drug.

Over the past two decades, the need for comprehensive and accurate pathway maps has resulted in creation of multiple broad and specific pathway databases. We review accomplishments and highlight remaining challenges.

Pathways in Systems Biology

Pathways are essential models in systems biology research and personalized medicine. High-dimensional data generated by HTP platforms, such as gene and protein arrays, sequencing, and ChIP-seq technologies, make integrating and understanding the information hidden in the HTP data challenging. Almost all HTP data analysis workflows use pathway and functional annotation and analysis to aid data interpretation and hypothesis generation.

Since almost every process or event in living organisms is a result of collaboration of one or more proteins, it is advantageous to summarize complex HTP data as less convoluted sets of complexes and signaling cascades. Pathways provide the information necessary to find and interpret these sets by grouping proteins whose coactivity is known to be important in cellular events or phenotypes. One common application of HTP technologies is in understanding the differential effect of treatments or diseases on genes, proteins, and metabolites. Pathway analysis of this data helps to fathom the underlying mechanisms of cells that are relevant to those conditions. Understanding the relevance of genotype and phenotype ([Carter et al., 2013](#)), differentiating subtypes of diseases ([Chuang et al., 2007](#)), and developing targeted-therapies ([Yauch and Settleman, 2012](#)) are a few examples of topics in which pathway analysis plays a significant role in current molecular biology and personalized medicine research.

Pathway Databases

As of August 2017, over five hundred pathway annotation, analysis, and visualization portals and tools are listed at [pathguide.org](#) portal ([Bader et al., 2006](#)). We review the most frequently used resources by considering different aspects such as data access, proteome coverage, level of domain-specificity, level of annotation provided, analytic services offered, format of input and output supported, etc. [Table 1](#) summarizes features of the most widely used databases of different pathway groups. We restrict the scope of this review to databases that are fully publicly available and include pathways for human. KEGG and BioCarta are the only two exceptions to these criteria, as KEGG was in the public domain until 2011, and BioCarta became free after its first few years as a commercial product.

Table 1 A list of major pathway databases and their features

Pathway Database	Status	First Release Listed/ Publication	Latest Release ^a (date/version)	Number of Annotated Genes/proteins (Unique)	Resource type	Level of Domain Specificity ^b	Services	Data Accessibility	File formats	Link	Reference
BioCarta	Discontinued	2001	2006	1369	Primary	1	NA	NA	NA	biocarta.com	doi:10.1089/152791601750294344 https://doi.org/10.1093/nar/gkn698
ConsensusPathDB	Maintained	2008	2017 / v.32	12,297	Integrated	1	Visualization	Download full lists	text	cpdb.molgen.mpg.de/CPDB	doi:10.1093/database/bar052
INOH	Discontinued	2011	NA	1554	Primary	1	NA	NA	NA	NA	doi:10.1093/database/bar052
IPANS	Maintained	2011	2015	1268	Hybrid	1	Visualization, enrichment	Download full lists	BioPax, JPG, SBML, SIF, XGML	ipavs.cidrns.org	doi:10.1093/nar/gkv1208
KEGG	Maintained	1995	2017 / v.83.1	6886	Primary	1	NA	Download images	Diagram	genome.jp/kegg/pathway.html	doi:10.1093/nar/gkv1070
PathCards	Maintained	2015	2017 / v.4.5.0.19	NA	Integrated	1	Visualization	Browse annotations	NA	pathcards.genecards.org	doi:10.1093/database/bar006
PathDIP	Maintained	2016	2017	18,144 (12,742 core)	Integrated/extended	1	Extended, enrichment	Download full lists and enrichment	text	ophid.utoronto.ca/pathdip	doi:10.1093/nar/gkv1082
PathExpand	Outdated	2010	NU	NA	Extended	1	Extended, enrichment (pre-computed)	Download enrichment analysis	Cytoscape	ico2s.org/servers/pathexpand.html	https://doi.org/10.1186/1471-2105-11-597
Pathway Commons	Maintained	2010	2017 / v.9	11,536	Integrated	1	Visualization	Download full lists	BioPax, SIF, gmt	pathwaycommons.org	doi:10.1093/nar/gkv1039
Reactome	Maintained	2005	2017 / v.61	10,374	Primary	1	Visualization, enrichment	Download full lists and analysis	BioPAX, PSI-MI, SBML, SBGN	reactome.org	doi:10.1093/nar/gkv1102
WikiPathways	Maintained	2008	2017	4823	Primary/de novo	1	Visualization, edit	Download full lists and images	BioPax, GPML, SVG, PDF, text, ovl, PNG	wikipathways.org/index.php/WikiPathways	doi:10.1093/nar/gkv1074
My Cancer Genome	Maintained	2016	2016	213	Primary	2 (cancer)	Visualization	NA	NA	mycancergenome.org/content/molecular-medicine/pathways	https://www.mycancergenome.org/content/molecular-medicine/pathways/doi:10.1038/cpt.2012.96
PharmGKB	Maintained	2000	2017	839	Primary/de novo	2 (drug)	Visualization	Download full lists and images	GPML, BioPax, text, diagram	pharmgkb.org/view/pathways.do	doi:10.1038/msb4100177
EHMN	Discontinued	2007	NA	1088	Primary/de novo	2 (metabolic)	NA	NA	NA	ehmn.bioinformatics.ed.ac.uk/	doi:10.1038/nar/gkv1164
HumanCyc	Maintained	2003	2016 / v.20.5	1159	de novo	2 (metabolic)	Visualization	Download full lists	BioPax, text	humancyc.org	doi:10.1093/nar/gkv1067
SMPDB	Maintained	2009	2015 / v.2.0	1059	Primary	2 (small molecules)	Visualization	Download full lists	PWML, SVG, SBGN, text, SBML, BioPax	smpdb.ca	doi:10.1093/nar/gkv1067
UniProt Pathways	Maintained	NA	2017	1146	Primary	2 (metabolic)	annotation	Download full lists	text	uniprot.org/help/pathway	UniProt Pathway
NetPath	Outdated	2010	NA	1372	Primary	2 (signalling)	Visualization, annotation	Download full lists	BioPax, PSI-MI, SBML	netpath.org	doi:10.1186/gb-2010-11-1-r3
Panther Pathways	Maintained	2006	2017 / v.12.0	2629	Primary	2 (signalling)	Visualization, annotation	Download full lists	SBML, SBGN	pantherdb.org/pathway	https://doi.org/10.1007/978-1-60761-175-2_7
PID	Discontinued	2008	2012	2569	Primary	2 (signalling)	NA	NA	NA	wiki.nci.nih.gov/pages/viewpage.action?pagelid=315491760	doi:10.1093/nar/gkn653

Signalink2.0	Outdated	2010	2011	528	integrated	2 (signalling)	Visualization	Download full lists	txt, BioPax, PSI-MI, SBML, Cytoscape	signalink.org	doi:10.1186/1752-0509-7-7
SIGNOR	Maintained	2015	2017	442	Primary	2 (signalling)	Visualization	Download full lists	text, PSI-Causal, Interaction	signor.uniroma2.it	doi:10.1093/nar/gkv1048
Spike	Outdated	2008	2012	1010	Primary	2 (signalling)	Visualization	Download full lists	BioPax, SIF, XML	cs.tau.ac.il/~spike	doi:10.1186/1471-2105-9-110
STKE	Discontinued	2002	2012	831	Primary	2 (signaling)	NA	Download full lists	SMBL	stke.sciencemag.org/about/help/cm	Connections-Map-XML - Schema-version - 7.0
OntoCancro	Outdated	2009	2014	1229	Integrated	3 (Genome Maintenance Mechanisms)	NA	NA	NA	ontocancro.inf.ufsm.br/previous/ontocancroPathways.php	doi:10.1039/c2mb25242b
RB-Pathways	Outdated	2008	NU	287	Primary	3 (retinoblastoma)	Visualization	Download full lists	BioPax, Cytoscape, CellDesigner	doi:10.1038/msb.2008.7	

NA – not available; NU – no updates; outdated – not updated since 2014.

^aObserved August 2017.

^b1, General; 2, Intermediate (e.g., Metabolism, signalling, disease, drugs, etc.); 3, Context-specific.

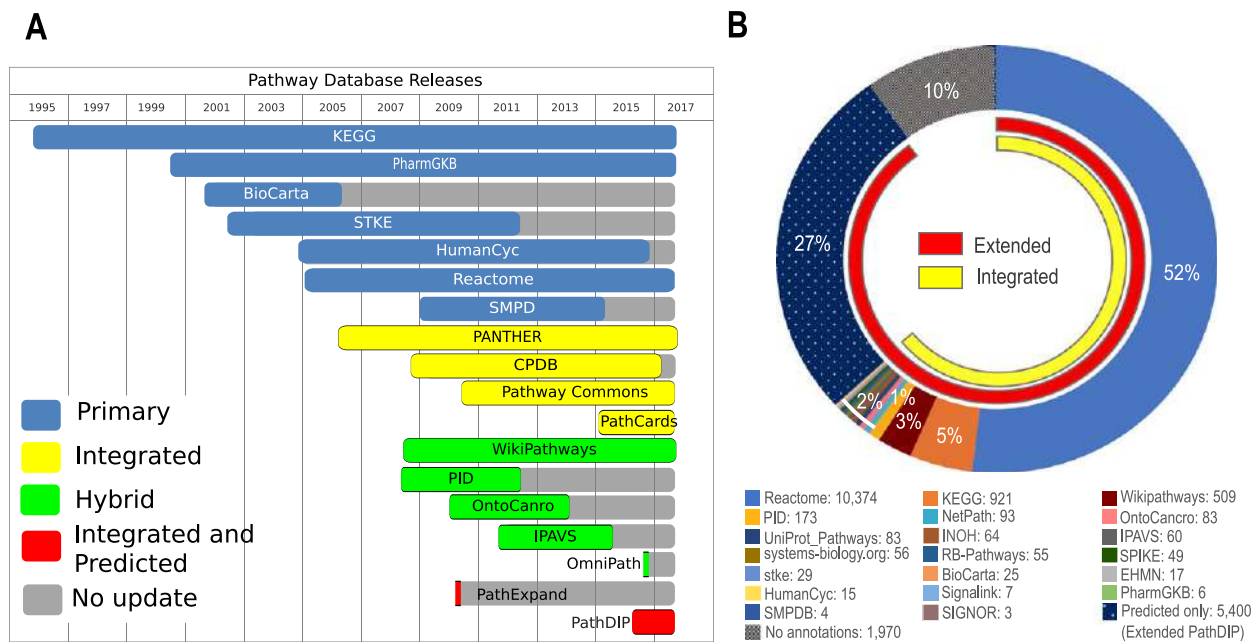


Fig. 1 (A) Timeline of development and update of major pathway databases grouped based on sources of their data: curated, integrated, hybrid, and predicted. (B) Comparison of human protein-coding gene coverage across different publicly available pathway databases. Starting from Reactome as the largest primary pathway database, this diagram shows incremental coverage for genes after adding individual databases. We have sorted primary and hybrid databases by the number of genes that each one adds. Approximating size of human genome to 20,000 genes, 27% of genome is annotated only through predictions (extended pathways) and 10% of genome remains void of pathway annotations.

Pathway Annotation Classes Based on Their Data Source

Pathway databases can be clustered into five major classes: primary, integrated, hybrid, extended, and *de novo*. They each offer different types of data, and they vary in proteome coverage and the detail they provide. Primary databases were the first category of pathway databases to be developed. Next, integrated and hybrid databases, which are the results of integration of primary pathway databases, appeared. Finally, extended pathway databases were developed to increase coverage and reduce bias present in the other three categories of databases. **Fig. 1(A)** depicts the timeline of development and update of the main representatives of the first four classes of pathway databases.

Primary pathway databases

Primary pathway databases, such as KEGG (Kanehisa, 1997), STKE (Gough, 2002), BioCarta (Nishimura, 2001), and Reactome (Fabregat *et al.*, 2016), were originally developed by expert collection of experimental findings that were spread across published papers. Biologists usually consider these databases to provide the highest quality pathway data. However, the drawbacks of primary databases include slow growth, partly due to limitations in technologies for experimental pathway discovery, and partly due to their reliance on manual human expert curation, which results in their limited proteome coverage.

To improve the speed and comprehensiveness of pathway curation, automatic literature mining tools have been developed (Kemper *et al.*, 2010; Ravikumar *et al.*, 2014a,b). However, these tools have several limitations and challenges such as the trade-off between the level of detail in their output and the complexity of their design, and the need for education of users to teach them how to define parameters, such as confidence level, to customize searches. Longer and more detailed lists of such challenges are reviewed in Ravikumar *et al.* (2014a,b); Huang and Lu (2016). Moreover, challenges for primary pathway databases include data heterogeneity and diversity in the nomenclature used, both of which will be further discussed in the following sections.

Integrated and hybrid pathway databases

Integrated pathway databases were developed by combining data from multiple primary databases into one consolidated database. Examples include Panther (Mi *et al.*, 2016), PathwayCommons (Cerami *et al.*, 2011), ConsensusPathDB (Herwig *et al.*, 2016), and PathCards (Belinky *et al.*, 2015). While they provide significantly increased coverage compared to primary databases, the challenge is the heterogeneity of the consolidated data, since each primary database may provide a different quality of curated information and level of detail.

The third class of pathway databases, which were developed along with integrated pathway databases, is *hybrid* databases (Liberzon *et al.*, 2015). They combine data from primary databases with new data curated from papers. Examples of these databases include PID (Schaefer *et al.*, 2009), iPavs (Sreenivasiah *et al.*, 2012), and WikiPathways (Kutmon *et al.*, 2016). Similarly to integrated databases, coverage is increased, but the information provided may be of inconsistent quality.

Data integration improved the proteome coverage of the primary databases, and quickly became popular among bioinformaticians and systems biology data analysts. However, integrated pathway databases still do not annotate a large portion of human protein-coding genes, limiting analysis and interpretation of HTP data. It has been shown that the twenty major pathway databases, combined, cover less than 65% of the human genome, leaving the rest of the genes as so called “pathway orphans” (Rahmati *et al.*, 2017).

Furthermore, integrating pathway data and keeping it up-to-date is a non-trivial task due to several challenges such as ID heterogeneity, ambiguous gene naming, open vocabulary, and lack of standard protocols for paper curation. Details about these challenges will be discussed in the following sections.

Extended pathway databases

Discovering and annotating pathways using experimental methods is expensive and time consuming, and may provide limited or biased coverage. In order to facilitate comprehensive and effective discovery of functional dependencies, several groups have suggested computational methods to integrate different types of biological data, such as sequence, expression, and interaction networks to predict novel protein-pathway associations.

Comparative genome analyses (Osterman and Overbeek, 2003; Moriya *et al.*, 2007; Port *et al.*, 2013), genome mapping (Date and Marcotte, 2003), and network integration (Glaab *et al.*, 2010, 2012; Herwig *et al.*, 2016; Rahmati *et al.*, 2017) are some of the methods that have been successfully applied to predict functional dependencies of proteins, and assign them to pathways. While the first two categories of methods are effective in capturing conserved functions of proteins, the advantage of network integration methods includes the capability of predicting organism-specific protein functions.

ConsensusPathDB (Herwig *et al.*, 2016), EnrichNet (Glaab *et al.*, 2012), PathDIP (Rahmati *et al.*, 2017), and PathExpand (Glaab *et al.*, 2010) are four major pathway data portals that integrate physical protein interactions with literature-based pathway information, to predict novel functional dependencies among proteins and pathways. Major differences among these databases, beyond the background data that they use, include the prediction algorithms used and the services provided.

ConsensusPathDB integrates a comprehensive list of pathways from twelve source databases. It can perform pathway enrichment analysis and gene annotation with pathways. However, it does not provide pathway annotation for genes that are not covered by its twelve pathway sources. It also performs neighborhood-based entity set (NEST) searches to predict gene sets that are closely related to the query list.

In EnrichNet, users first enter a list of query genes, and select a protein-protein interaction (PPI) network on which to base the analysis – either choosing from a list of four PPI networks provided by developers, or using the user's PPI network – and then customize their analysis by selecting an annotation database from the eight available databases, five of which are pathway databases. The analysis can be made condition-specific by choosing a gene expression dataset to be integrated with the selected PPI network. Although EnrichNet provides novel predicted protein-pathway associations for protein groups through a random walk between the target list and pathways, it cannot provide annotations for single proteins (the query list must contain at least ten identifiers) (Glaab *et al.*, 2012).

PathExpand has used three pathway databases, Biocarta, KEGG, and Reactome to provide proteins that act as pathway seeds on the PPI network. For its background network, it combines data from five publicly available PPI databases. Next, it defines three criteria for the relative connection of each protein in the PPI network to each pathway, in order to associate each protein to the seed pathway. Users can customize their search by selecting one of 1859 pathways provided by PathExpand and setting up expansion parameters. While its output includes known and predicted pathway members and their interactions as a gene list, an interactive graph layout, and a Cytoscape readable network, PathExpand does not support gene annotation or pathway enrichment analysis.

PathDIP integrates physical PPIs and known pathways from twenty primary and hybrid pathway databases to calculate a physical dependency score (association score) for proteins and pathways. Users can customize their search and analysis by choosing any subset of the twenty pathway databases. They can also decide on the type of protein-pathway association sets to include, i.e., *core* (curated) or *extended* (curated and predicted). Importantly, annotations for each gene are independent of the full input list. PathDIP also provides pathway and term over-representation analysis services. Results are available in different views, for example, table, matrix, or text files with word frequencies from enriched pathway titles.

Although extended pathway databases improve data coverage and decrease data biases (Fig. 1(B)), they cannot fully resolve these issues. Extended databases combine different types of biological data to extract hidden information in them. However, they are prone to incompleteness and noise derived from their sources, which results in presence of false positives and false negatives in their predictions (Rahmati *et al.*, 2017).

De novo pathway discovery

The last group of pathway databases includes *de novo* pathway prediction tools. *De novo* pathway discovery methods typically use a background network, and hypothesize that tightly connected gene or protein subnetworks are functionally related. The background network can be global, such as a full PPI network (Oliver, 2000; Sun *et al.*, 2012), or it can be constructed from condition-specific data such as from gene expression or genome-wide association studies (Chuang *et al.*, 2007; Brorsson *et al.*, 2009; Hung *et al.*, 2010; Jia *et al.*, 2011; Alcaraz *et al.*, 2017; Zhang and Zhang, 2017). Since these methods are not limited to prior knowledge about protein functions, they can discover novel pathways. Considering the availability and rapid growth of HTP biological data,

these methods are essential to annotate less-studied proteins and functions, which may be conserved across different conditions, or may be specific to a certain condition.

However, this category of pathway detection methods is not free of disadvantages. One of the biggest challenges is evaluating the quality of the discovered pathways. One approach to evaluating the discovered pathways is to use known data available in primary, integrated, and extended pathway databases. However, using previous studies to evaluate novel pathways can lead to biased conclusions, due to overestimation of the quality of results for the known data, while underestimating the quality of predictions that discover novel pathways. In order to address this problem, *Batra et al.* have suggested a framework for creating artificial networks and pathways (*Batra et al., 2017*). They have compared the performance of seven publicly available *de novo* pathway discovery software and applications, and concluded that none of them is consistently superior to other ones. Therefore, they have suggested guidelines on how to select a method that fits a specific study. Their comparison method is the first attempt to quantitatively compare *de novo* pathway evaluation methods (*Batra et al., 2017*).

Domain-Specificity of Pathways

The level of domain-specificity is another factor based on which we can categorize pathway databases. While some pathway databases focus on very specific classes of signaling cascades, such as autophagy (*Xie and Klionsky, 2007*), or apoptosis (*Doctor et al., 2003*), other databases cover broader areas, such as signal transduction (*Kandasamy et al., 2010; Gao et al., 2013*), metabolism (*Romero et al., 2005; Ma et al., 2007; Glaab et al., 2012; Wishart et al., 2013*), gene regulation (*Han et al., 2015; Marbach et al., 2016*), tissue and disease signaling (*Huang et al., 2008; Schaefer et al., 2009; Simão et al., 2010*), or drug response pathways (*Whirl-Carrillo et al., 2012*); or provide even more general and comprehensive information about pathways, such as KEGG, Reactome, WikiPathways, or PathDIP (*Kutmon, et al., 2016*).

Level of Pathway Details

Pathway databases also provide varying levels of molecular detail and annotation. Some of them, such as MSigDB (*Liberzon et al., 2015*), and ConsensusPathDB (*Herwig et al., 2016*), only include gene sets. Other databases such as PhosphositePlus (*Hornbeck et al., 2015*) and SIGNOR (*Perfetto et al., 2016*) provide process-specific interaction networks, while SPIKE, WikiPathways, Reactome and KEGG (*Paz et al., 2011; Kutmon, et al., 2016; Fabregat et al., 2016; Kanehisa et al., 2016*) describe details about the dynamics of interactions, including the sequence of events and detailed reactions (*Fig. 2*).

Databases of gene sets (*Fig. 2(A)*) and process-specific interaction networks (*Fig. 2(B)*) are sufficient for pathway and network enrichment analyses. However, databases that provide details about the dynamics of pathways are required for analyses that aim to model and interpret the sequence of reactions and events that comprise a biological process (*Fig. 2(C)*).

Design, Services, and Data Sharing

Pathway databases containing process-specific interaction networks usually provide pathway enrichment and network analysis services. Their data and analysis results can be stored and downloaded as text in a variety of simple file formats such as GMX (Gene Matrix format), GMT (Gene Matrix Transposed format), and PSI-MITAB (Proteomics Standards Initiative – Molecular Interactions). Some of the database portals provide visualization tools (e.g., EnrichNet, ConsensusPathDB), such as bar-plots, word-clouds, or networks (*Glaab et al., 2012; Herwig et al., 2016*).

However, these file formats and visualization tools are not detailed enough to represent the integrated dynamic reactions that form pathways. For example, they do not support ordering of events, or hierarchies of reactions and pathways. This type of data needs to be represented as diagrams in rich, machine-readable file formats such as BioPAX (Biological Pathways Exchange), SBML (Systems Biology Markup Language), and SBGN (Systems Biology Graphical Notation). Understanding the contents of these files usually requires special parsers, such as Paxtools (*Demir et al., 2013*). Some of these databases also provide visual diagrams of pathways in PDF, or image file formats. Databases such as Reactome (*Fabregat et al., 2016*), SPIKE (*Paz et al., 2011*), and ACSN (Atlas for Cancer Signaling Networks) (*Kuperstein et al., 2015*) offer interactive pathway maps through which users can zoom in and out, select a particular part of map to see more details, and hide/show specific types of molecules or reactions.

Reactome and WikiPathways make available data and services at different levels of detail, and in several different formats. For example, Reactome not only provides pathway data, but also offers services such as enrichment and network analysis. Its data and search results can be downloaded as gene sets in GMT, interaction sets as MITAB, and full dynamic pathways in BioPAX, SBML, and SBGN formats. The full Reactome database is also provided as a textbook in PDF and RTF formats. Similarly, WikiPathways provides gene sets in TXT, dynamic pathways in BioPAX, and diagrams in SVG, PDF, and PNG formats. Therefore, familiarity with pathway data sharing protocols and different types of pathway data representation can be useful when selecting the download format when using their data and services.

Data accessibility and pathway data sharing are other factors that distinguish individual databases. While many pathway databases support easy download of their full data (*Kutmon, et al., 2016; Fabregat et al., 2016; Herwig et al., 2016*), others restrict users' access at different levels; some only provide restricted download, for example, HPCIN and KEGG (*Huang et al., 2008; Kanehisa et al., 2016*). A few publicly available databases, such as PathCards (*Belinky et al., 2015*), let users browse the database

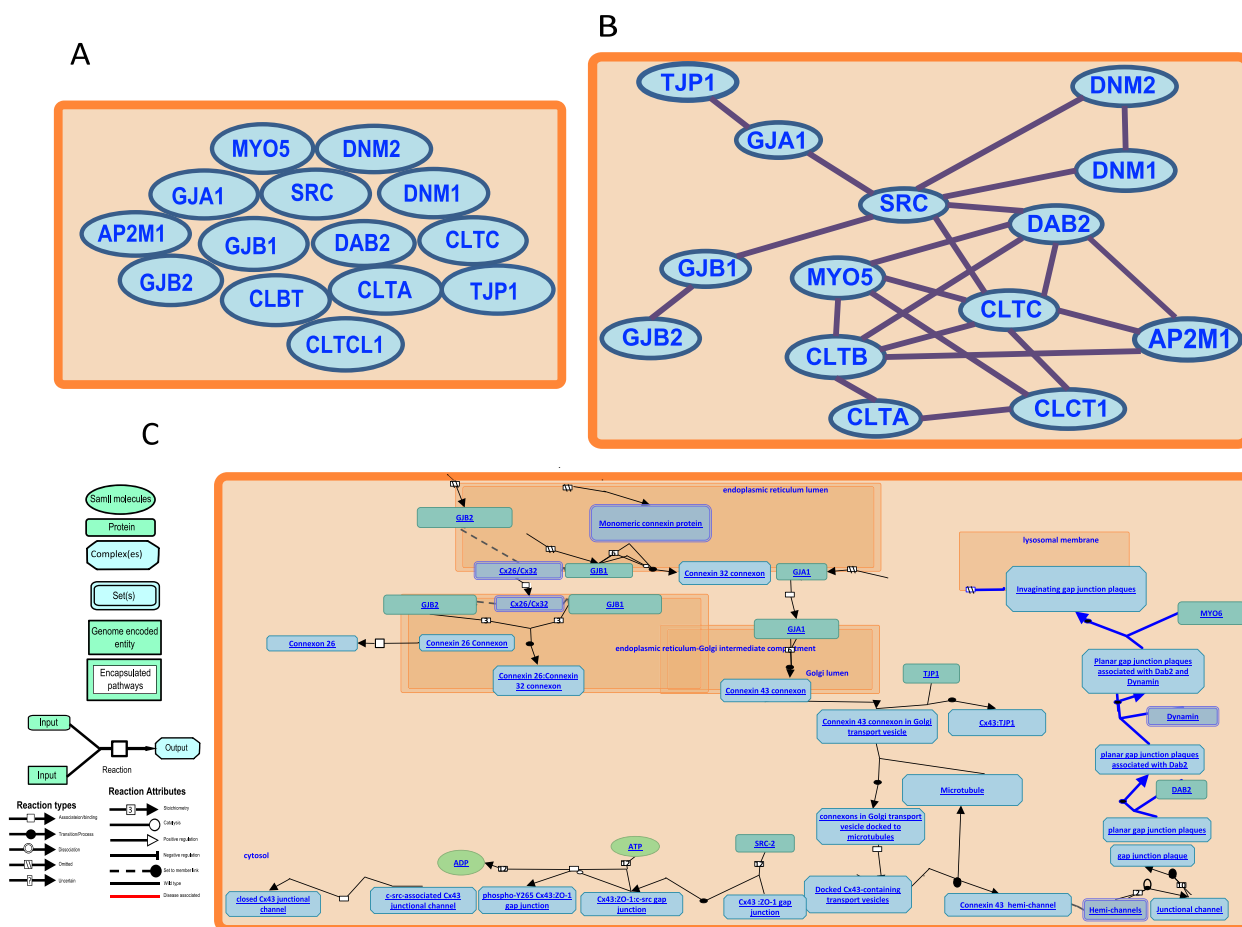


Fig. 2 Three major groups of pathway definitions based on level of detail. (A) *Gene sets*: genes grouped based on their functions. Usually used for over-representation and enrichment analysis, or for extracting functional gene annotations. (B) *Process-specific interaction networks*: genes inside each pathway are connected through functional links, providing high level molecular network topology. (C) *Dynamics in pathways*: Details about order of events, type of signaling, small molecules necessary to facilitate reactions, types of reactions, etc. are included in pathway maps and represented as images, interactive maps, or rich xml-based file formats (An example from Reactome (R-HSA-157858: Gap junction trafficking and regulation)).

but do not allow download. The last group, which are out of scope of this review, are fully commercial databases for which users need to purchase a license. Examples include Ingenuity Pathway Analysis (see Relevant Websites section), Tocris Bioscience Signaling Pathways (see Relevant Websites section), and Sino Biologicals (see Relevant Websites section).

Platforms for Pathway Reconstruction

In order to facilitate the design and development of pathway descriptions, several biomolecular reaction and pathway editors have been developed (Table 2). CellDesigner (Funahashi *et al.*, 2008), PathVisio (Kutmon *et al.*, 2015), and PathwayMapper (Bahceci *et al.*, 2017) are three widely used, publicly available tools, while Cell Illustrator and PathwayBuilder are two commercial products for designing and editing pathways. Payao (Matsuoka *et al.*, 2010) is a platform that uses CellDesigner to provide an online editor specific to pathway model curation, exchanging comments, and recording the discussions among collaborators.

Challenges

Pathways are models used by humans to describe, share, and discuss knowledge about cellular events. High demand for these models has resulted in the development of many pathway databases and analysis services. Despite all the efforts and achievements in pathway databases in the last twenty-four years, many challenges remain. While our incomplete knowledge of cellular processes is one major issue, lack of consensus on definitions, curation and annotation protocols, structured vocabulary, and context of pathways, limits the utility of all available information.

Table 2 List of active interactive pathway editors and visualizers

Visual Pathway Editor	Type	Availability	Services	Web Address	PMID
PathwayMapper	Portal	Free	view; edit; Collaborate	http://pathwaymapper.org	28334343
ccNetViz	Software	Free/ Open source	View	http://helikarlab.github.io/ccNetViz/	NA
VitaPad	Application	Free/ Open source	View; edit	https://sourceforge.net/projects/vitapad/	15564306
PathWhiz	Portal	Free	View	http://smpdb.ca/pathwhiz	25934797
BioPAX translator	Plugin	Free	View	ftp://ftp.pantherdb.org/CellDesigner/plugins/BioPAX/	22021903
BioPAXViz	Plugin	Free	View	https://github.com/CGU-CERTH/BioPAX.Viz	28453679
Arcadia	Software	Free	View	http://arcadiapathways.sourceforge.net/	20453003
Escher	Portal	Free	View	https://escher.github.io	26313928
PathwayMatrix	Software	Free	View	https://github.com/CreativeCodingLab/PathwayMatrix	26361499
CellDesigner	Software	Free	View; edit	http://www.celldesigner.org	24927840
PathVisio	Software	Free/ Open source	View; edit	http://www.pathvisio.org	25706687
Cell Illustrator	Software	License	View	http://www.cellillustrator.com	21685571
PathwayBuilder	Portal	License	View; edit	http://www.proteinlounge.com/PathwayBuilder.aspx	NA
PathwayTools	Software	Free	View; edit	http://brg.ai.sri.com/ptools/	NA
Payao	Portal	Free	View; edit; Collaborate	http://celldesigner.org/payao/payaopreview.html	20371497
ipath2	Portal	Free	View	https://pathways.embl.de/	18276143

Human Proteome Coverage

Currently, Reactome (V60) and KEGG (release 81), the two largest primary pathway databases, annotate 10,347 and 6886 unique protein-coding genes, respectively. Despite their size, they only overlap by 53%, and thus cover only about half of the human proteins, leaving over 9000 proteins with no pathway annotation (Rahmati *et al.*, 2017). Incompleteness of proteome coverage results in several challenges. First, any enrichment analysis using any one of the largest single primary database may not consider ~50% of the input proteins. Second, due to the small overlap of the primary databases, using different databases for enrichment analysis will provide inconsistent results. Although development of integrated databases has considerably reduced both problems, it has been shown that about 37% of the human protein-coding genes still lack pathway annotation, even after the 20 largest and most widely used pathway databases were integrated (Rahmati *et al.*, 2017).

Extended pathway databases were developed to provide improved proteome coverage. For example, pathDIP (version 3) makes available pathway annotation for over 90% of human protein-coding genes, 12,630 genes covered by integrated primary databases and 5400 additional genes covered by predicted pathway associations. While extended databases provide evidence for the physical association of proteins with pathway literature-based members, they still leave about 10% of protein coding genes devoid of annotations, mainly due to incompleteness and bias in their resources. Importantly, due to the integration of diverse resources, level of detail and quality of protein annotation is highly dependent on the quality of data available in those sources.

ID Conversion and Naming

An overarching challenge in pathway data integration is the diversity of the type and version of IDs, and ambiguity in the gene symbols used by different databases. For example, while SIGNOR accepts official gene symbols and UniProt IDs, it does not support Entrez Gene IDs; in contrast, SPIKE supports gene symbols and Entrez Gene IDs, but not UniProt IDs. While many major pathway and general bioinformatics resources have built-in ID conversion tables, and allow users to input different ID types, converting IDs is not trivial since most of the IDs are not fully stable, and some conversions are one-to-many. It can cause ambiguity (and imprecision) when using less frequently updated databases, and when curating older papers. This problem is particularly noticeable when using databases that are no longer updated such as BioCarta, SPIKE, INOH (Yamamoto *et al.*, 2011), and STKE.

Ambiguity in gene symbols (and their synonyms) is an even more challenging problem, as gene symbols are not unique identifiers. For example, PKB is a synonym for both AKT1 and PTK2B, and MCP-1 refers to both Metaphase Chromosome Protein and Monocyte Chemo attractant Protein 1. This can cause errors and inconsistency in curating papers, and in study results.

Although initiatives such as HGNC (Gray *et al.*, 2015) approach this problem by maintaining up-to-date ID-conversion tables and lists of gene names, the problem remains for other reasons, such as software errors in reading and representing gene names.

Ziemann *et al.* have systematically scanned 35,174 supplemental excel files attached to 3597 papers published between 2005 and 2015 in eighteen high-impact journals (Ziemann *et al.*, 2016). Based on their estimates, almost one fifth of the gene names in papers and their supplemental material are erroneous due to MS EXCEL converting gene names to dates and floating-point numbers. Therefore, structured curation, proper data sharing workflows, and support for data provenance, are essential to reduce these challenges.

Heterogeneity In Pathway Definition

Pathway data integration and interpretation is challenging due to the diversity of pathway definitions. Since pathways are models created by human experts, the same pathways, as represented in different databases, may contain different proteins and reactions, and likely will have different names. For example, while Hedgehog signaling is a single pathway in KEGG, Signalink2.0 (Fazekas *et al.*, 2013) has divided it into Hedgehog-core and Hedgehog-nonCore pathways. These three pathways only partially overlap. Due to this heterogeneity, most integrated pathway databases store pathways from different sources as separate pathways. One consequence of this approach is redundancy of data that usually leads to long lists of pathways as annotation and enrichment analysis results. This makes interpretation of the results challenging. Word-clouds (Herwig *et al.*, 2016) and term enrichment analysis (Rahmati *et al.*, 2017) are two effective strategies that partially solve this problem; however, neither of them can fully resolve it.

PathCards (Belinky *et al.*, 2015) has tried to tackle redundancy and heterogeneity in pathways content and names by consolidating pathways of related processes in super-pathways (using genes). An “onion” model, introduced by Signalink2.0, defines core and non-core pathways, and provides one potential solution to this problem. PathCards also uses a modified version of this strategy to define super-pathways and sub-pathways in a multilayer system by scoring membership of each gene in each super-pathway based on the frequency of appearance of that gene in source pathways under that super-pathway. PathCards first clusters pathways by computing a pairwise Jaccard Similarity, and applying a two-tier clustering approach that combines hierarchical clustering and nearest neighbor joining. In order to name each super-pathway, they use name of the member of each cluster (super-pathway) that is most highly connected to other members of the cluster. Using this approach, they reduce the number of pathways from 3215 across 12 different sources to 1073 super-pathways in PathCards. The frequency-based score is next used to assign the strength of membership of each gene in a super-pathway, i.e., forming the “onion” model. However, the authors have highlighted that most super-pathways comprise pathways from two sources, with a maximum of five sources in each super-pathway, which is attributed to the inherent biases in content of single pathway databases (Belinky *et al.*, 2015).

Our analysis confirms that pairwise pathway overlap alone is not sufficient as a similarity measure to accurately consolidate pathways. Fig. 3 shows hierarchical clustering of 4976 core pathways integrated in PathDIP from twenty major pathway databases. Using the Jaccard Similarity of pathway pairs according to their protein overlap as a similarity measure of the processes that they describe, we investigated members of different clusters, as well as the relative positions of the same processes from different source databases, on the clustering tree. Interestingly, pathways from the same database are frequently clustered closer to each other than the same processes from different databases. Consistent with our earlier examples in this article, we focused on the Hedgehog pathway. After cutting the clustering tree at height 1.5 (maximum height of tree is 8.65) we found 1378 clusters out of which 7 contained at least one Hedgehog pathway (clusters number 9, 25, 75, 520, 1312, 1365, and 1366). In Fig. 3(B-D) we highlight clusters #9, #520, and #1366. While in cluster #1366 four Hedgehog pathways from four different sources are grouped together (Hedgehog Signaling from Signalink, SIGNOR, and NetPath, and “Signaling events mediated by Hedgehog family” from PID) (Fig. 3(B)), cluster #512 has only one member which is “Hedgehog signaling events mediated by Gli proteins” from PID (Fig. 3(C)), and cluster #1366 includes only one Hedgehog pathway from Reactome (“Hedgehog ligand biogenesis”), which is grouped with 31 other pathways, out of which only one pathway is not from Reactome. Non-canonical NFκB signaling, protein degradation/proteasome, regulation of apoptosis, and p53-related pathways form four groups of pathways in this cluster, all of which have Reactome as their source database (Fig. 3(D)). This divergence in clustering of similar processes from different databases further highlights the heterogeneity of the contents of pathway databases, and the necessity to develop effective pathway annotation, integration, and consolidation methods.

Another major challenge in pathway consolidation is the ambiguous terminology that pathway databases use. For example, “mRNA splicing” and “Spliceosome” have been used for the same process in Reactome and KEGG, respectively. Similarly, “Hedgehog Signaling” in Wikipathways is referred to as “Signaling events mediated by Hedgehog family” in PID. Defining a consensus controlled vocabulary, or pathway ontology can help address this problem.

Pathway Curation

Manual pathway mining

Primary pathway databases are created and fully depend on manual curation of papers by experts. Since the beginning of pathway databases in 1995, these databases have been enhanced in many respects, and proteome coverage has been increased by data integration. Currently, pathDIP (version 3) includes 4976 pathways and represents 212,155 pairs of literature curated protein-pathways associations among 12,742 unique human protein-coding genes. Although several computational techniques for *de novo*

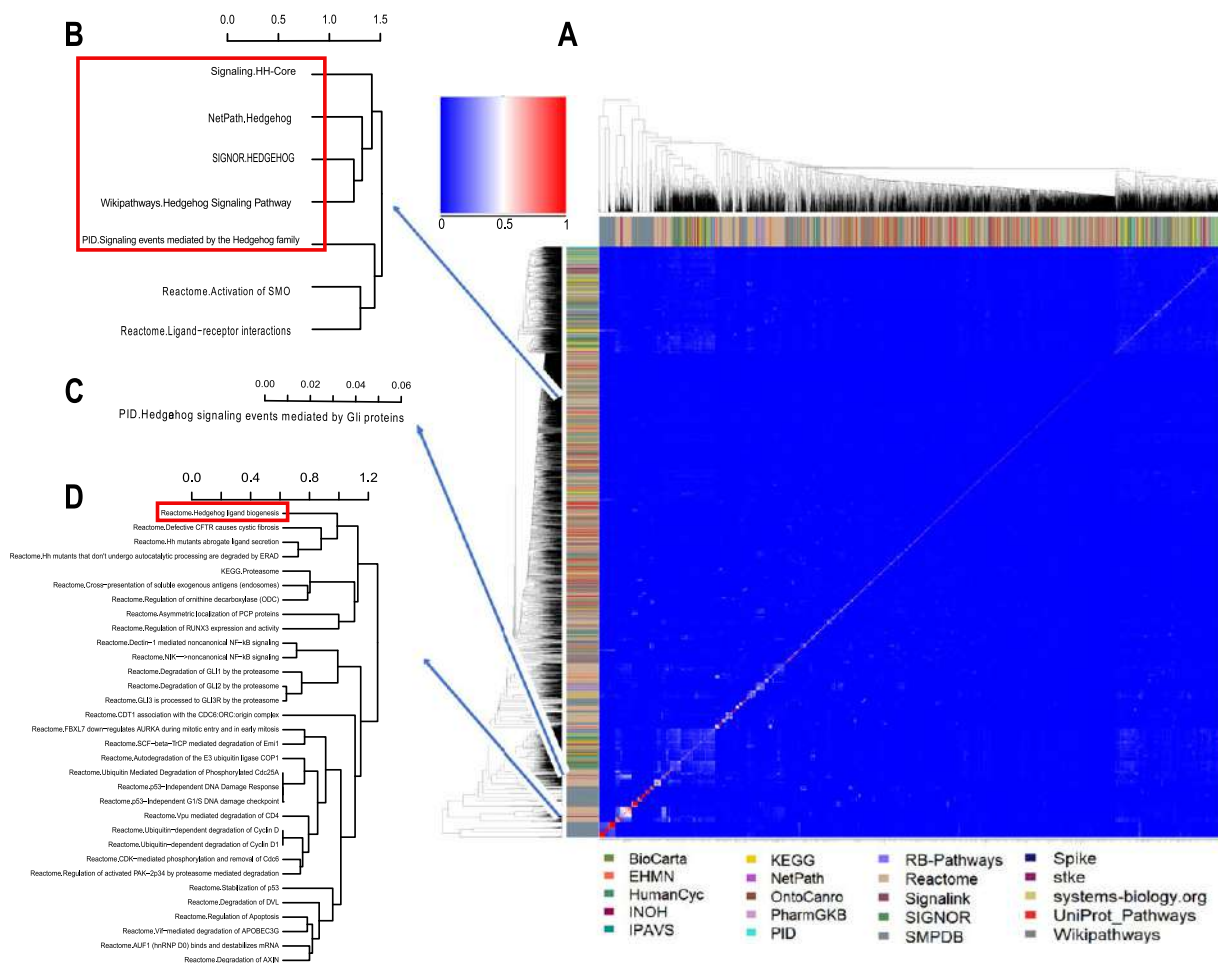


Fig. 3 (A) Clustering of 4976 pathways from 20 pathway databases (obtained from PathDIP v.3) based on their gene content overlaps is not sufficient for consolidating pathways effectively. (Color coding shows Jaccard Similarity of gene content of pathways.) In order to illustrate the divergence in pathways that are grouped in close clusters, we cut the clustering tree at height 1.5 (max 8.65) and found 1378 clusters. At this cut-off level, cluster sizes are small enough to plot them. Surprisingly, cluster members are highly divergent, even though the cut-off distance is very small. Using Hedgehog pathway as an example, we found seven clusters had at least one hedgehog pathway (#9, 25, 75, 520, 1312, 1365, and 1366). In this example, we have located two of these furthest clusters and one cluster in the middle, i.e., (B) cluster #9, (C) cluster #520, and (D) cluster #1366.

pathway prediction (or protein-pathway association prediction) have been developed, curated pathways remain the main source of high quality pathway definitions (with documented provenance, accuracy, and details).

However, pathway curation is slow and prone to errors and inconsistencies due to the above-mentioned ambiguities in protein and pathway naming. Furthermore, depending on the purpose of a study, the focus of a database, and the expertise of individual curators, different information can be extracted from the same paper and organized into different ontologies. For example, Reactome (V60) includes two sets of pathways, “PKB mediated events”, and “AKT phosphorylates targets in the cytosol”. Both sets include pathways and reactions related to the AKT1 gene, but while the second set includes the correct gene name in the pathway title, the first set includes the old nomenclature of AKT1 (PKB). Not surprisingly, these two pathway sets were curated and reviewed by different curators, in 2006 and 2015, respectively. It is easy to imagine how challenging it can be to integrate pathways with such nomenclature variability, and how difficult it can be to identify overlapping reactions from the two sets of pathways if the main gene is named differently (for example, the reaction describing AKT phosphorylation of TSC is called “AKT phosphorylates TSC2, inhibiting it” in one set and “Phosphorylation of TSC2 by PKB” in the other set).

The level of confidence in the experimental methods that have been used in source papers is another issue that usually is not recorded by pathway curators. Hence, the reliability of different modules of a pathway that have been extracted from different papers, can be inconsistent. Moreover, manual curation is prone to human errors. Schnoes *et al.* have reported a high rate of molecular function mis-annotation in six superfamilies of enzymes in KEGG (average ranging from 22% to 63% for each of the six enzyme clusters) that they attribute to misinterpretation by human curators (Schnoes *et al.*, 2009). Over 80% of these errors were identified as over-annotation, i.e., specificity levels higher than what the source paper implied. Such errors easily propagate into pathway databases and influence the interpretation of experiments.

Automatic pathway mining

Over the last decade several automatic literature mining approaches and expert pathway search engines have been implemented to accelerate biomedical literature curation (including, but not limited to, pathways) (Luo *et al.*, 2014; Subramani *et al.*, 2015; Ravikumar *et al.*, 2014a,b). However, automatic pathway curation is a challenging task that has several steps, including concept detection, document classification, information extraction, and ontology construction (Ravikumar *et al.*, 2014a,b), each of which is a research field in computer science. We refer the reader to (Ravikumar *et al.*, 2014a,b; Huang and Lu, 2016) for a detailed discussion of challenges in biomedical and pathway text mining systems.

Pathway Context

Pathway consolidation and pathway enrichment analysis are strongly affected by the context in which processes and signaling in a given pathway are defined. For example, most genes are part of several pathways, and their activity and relationship with those pathways can vary across different conditions/contexts. Annotating pathways with context information is challenging as it requires details about which part of the pathway is invariant to existing conditions, and which part may be context-specific; it is especially difficult since, in most cases, such information is not available.

Due to the importance of context in using and understanding pathways, more recently-developed pathway databases and tools have included predicted context to their analysis (Moulos *et al.*, 2013; Pinto *et al.*, 2014; Kuperstein *et al.*, 2015). Furthermore, context-specific pathway databases such as NetPath (Kandasamy *et al.*, 2010), PID (Schaefer *et al.*, 2009), RB-pathways (Calzone *et al.*, 2008), and PharmGKB (Whirl-Carrillo *et al.*, 2012) have been developed, focusing on phenotype-specific pathways (e.g., cancer) or condition-specific pathways (e.g., effect of a drug on a signaling cascade). Similarly, general-domain pathway databases, including KEGG and Reactome, have developed pathway categories, for example drug pathways or disease-specific pathways.

Despite these recent efforts, recording context during manual curation and considering it in designing pathway ontologies has not been traditionally part of the pathway annotation and sharing process. Although the importance of context is becoming more evident, the type and level of detail that is included as context differ across databases. Therefore, depending on the workflow, specific databases or even the original papers may be the best and the most accurate source of information. To further aid in this effort, expert data mining systems have been developed (Kemper *et al.*, 2010; Miwa *et al.*, 2013). The common concept behind these systems is to provide search engines with which users can find literature evidence for a process, pathway, or a part of it. Through these efforts, existing primary and integrated (hybrid) databases may be further expanded to provide both broad coverage and an increased, more homogeneous, level of detail across proteins and pathways.

Summary and Concluding Remarks

Over the past two decades, pathway databases have expanded in numbers, coverage, and richness of annotation they provide. They are an integral part of most of bioinformatics workflows used to understand system-level effects, for example immune response to stress or therapy (Brodin *et al.*, 2013), using previous knowledge. They are an essential component of pathway-targeted cancer-drug therapy methods (reviewed in Yauch and Settleman (2012)); offer insight in HTP data analyses such as gene and protein expression, and genome assembly and annotation (Braun, 2014) through enrichment analysis; and help establish the relevance of genotypes and phenotypes (Carter *et al.*, 2013; Chang *et al.*, 2015). Currently, several hundred pathway databases are freely available, each of which groups and annotates genes and their relationships at different levels of detail and context.

Data sharing and access support also vary across databases, with many providing data in diverse file formats and offering different analysis techniques and algorithms (reviewed in García-Campos *et al.* (2015)). To improve the consistency and speed of curation, diverse software for editing and visualizing pathways and their analyses results have been produced. (Table 1) summarizes these specifications of major pathway databases.

Despite all these achievements, several challenges remain. Although each of these challenges has its specific steps to be resolved, one common cause behind most of them is the lack of standard protocols for paper curation, and the lack of a common pathway ontology. Defining standards for pathway extraction, annotation, organization, and sharing of biological data has successful examples. IMEx (International Molecular Exchange Consortium) for example, is an international effort to define standards for protein interaction data providers to share their curation efforts and decrease redundancy in protein interactions (Orchard *et al.*, 2012). Similarly, BioPAX is an international initiative to facilitate pathway data exchange among pathway data providers (Demir *et al.*, 2010). While BioPAX is implemented as an ontology for pathway data sharing, it only provides a framework in which different ontologies can be defined. For example, KEGG and Reactome have defined their ontologies differently, both of which are deliverable through BioPAX. However, the current version of BioPAX (v3) does not define a standard controlled language (ontology) for sharing pathways. Although efforts such as INOH (Yamamoto *et al.*, 2011) and the Bioportal Pathway Ontology (Petri *et al.*, 2014) are focused on defining ontologies for pathway data, no consensus ontology has been accepted. Defining such a structured vocabulary, along with standard protocols for pathway curation, and formats for storing and sharing data (such as BioPAX) can overcome most of the current major issues in curation, sharing, and integrating pathway data, and increase consistency across different databases. Such improvements would also facilitate development of more efficient automatic pathway curation systems.

Despite their incompleteness and heterogeneity, pathway databases are crucial resources in current systems and molecular biology. While there are certain steps that need to be taken to improve quality of pathway data and their consistency, data mining, data annotation and integration, and *de novo* pathway prediction techniques are among the most promising directions for further improvement of pathway databases. They will improve the coverage, consistency, and depth of pathway data, enabling diverse bioinformatics workflows, and aiding systematic and comprehensive exploration of signaling processes in diverse biomedical contexts.

See also: Biological Database Searching. Biological Pathways. Cluster Analysis of Biological Networks. Computational Systems Biology. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Pathway Informatics. Protein–Protein Interaction Databases. Visualization of Biomedical Networks

References

- Alcaraz, N., List, M., Batra, R., *et al.*, 2017. De novo pathway-based biomarker identification. *Nucleic Acids Research*. e-151. doi:10.1093/nar/gkx642.
- Bader, G.D., Cary, M.P., Sander, C., 2006. Pathguide: A pathway resource list. *Nucleic Acids Research* 34 (Database issue), D504–6. doi:10.1093/nar/gkj126.
- Bahceci, I., Dogrusoz, U., La, K.C., *et al.*, 2017. PathwayMapper: A collaborative visual web editor for cancer pathways and genomic data. *Bioinformatics*. 1–3. doi:10.1093/bioinformatics/btx149.
- Barratt, R.W., Newmeyer, D., Perkins, D.D., Garnjobst, L., 1954. Map Construction in *Neurospora Crassa*. *Advances in Genetics* 6 (C), 1–93. doi:10.1016/S0065-2660(08)60127-3.
- Batra, R., Alcaraz, N., Gitzhofer, K., *et al.*, 2017. On the performance of de novo pathway enrichment. *NPJ Systems Biology and Applications* 3 (1), 6. doi:10.1038/s41540-017-0007-2.
- Belinky, F., Nativ, N., Stelzer, G., *et al.*, 2015. PathCards: Multi-source consolidation of human biological pathways. *Database* 2015 (0), doi:10.1093/database/bav006. [bav006-bav006].
- Braun, R., 2014. Systems analysis of high-throughput data. *Advances in Experimental Medicine and Biology* 844, 153–187. doi:10.1007/978-1-4939-2095-2_8.
- Brodin, P., Valentini, D., Uhlin, M., *et al.*, 2013. Systems level immune response analysis and personalized medicine. *Expert Review of Clinical Immunology* 9 (4), 307–317. doi:10.1586/eci.13.9.
- Borsson, C., Hansen, N.T., Lage, K., *et al.*, 2009. Identification of T1D susceptibility genes within the MHC region by combining protein interaction networks and SNP genotyping data. *Diabetes, Obesity and Metabolism* 11 (SUPPL. 1), 60–66. doi:10.1111/j.1463-1326.2008.01004.x.
- Calzone, L., Gelay, A., Zinovyev, A., Radvanyi, F., Barillot, E., 2008. A comprehensive modular map of molecular interactions in RB/E2F pathway. *Molecular Systems Biology* 4. doi:10.1038/msb.2008.7.
- Carter, H., Hofree, M., Ideker, T., 2013. Genotype to phenotype via network analysis. *Current Opinion in Genetics and Development* 23 (6), 611–621. doi:10.1016/j.gde.2013.10.003.
- Cerami, E.G., Gross, B.E., Demir, E., *et al.*, 2011. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Research* 39 (Database), D685–D690. doi:10.1093/nar/gkq1039.
- Chang, J., Gilman, S.R., Chiang, A.H., Sanders, S.J., Vitkup, D., 2015. Genotype to phenotype relationships in autism spectrum disorders. *Nature Neuroscience* 18 (2), 191–198. doi:10.1038/nn.3907.
- Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., Ideker, T., 2007. Network-based classification of breast cancer metastasis. *Molecular Systems Biology* 3. doi:10.1038/msb4100180.
- Date, S.V., Marcotte, E.M., 2003. Discovery of uncharacterized cellular systems by genome-wide analysis of functional linkages. *Nature Biotechnology* 21 (9), 1055–1062. doi:10.1038/nbt1861.
- Demir, E., Babur, Ö., Rodchenkov, I., *et al.*, 2013. Using Biological Pathway Data with Paxtools. *PLOS Computational Biology* 9 (9), doi:10.1371/journal.pcbi.1003194.
- Demir, E., Cary, M.P., Paley, S., *et al.*, 2010. The BioPAX community standard for pathway data sharing. *Nature Biotechnology* 28 (9), 935–942. doi:10.1038/nbt.1666.
- Doctor, K.S., Reed, J.C., Godzik, A., Bourne, P.E., 2003. The apoptosis database. *Cell Death and Differentiation* 10 (6), 621–633. doi:10.1038/sj.cdd.4401230.
- Fabregat, A., Sidiropoulos, K., Garapati, P., *et al.*, 2016. The reactome pathway knowledgebase. *Nucleic Acids Research* 44 (D1), D481–D487. doi:10.1093/nar/gkv1351.
- Fazekas, D., Koltai, M., Túrei, D., *et al.*, 2013. Signalink 2 – A signaling pathway resource with multi-layered regulatory networks. *BMC Systems Biology* 7 (1), 7. doi:10.1186/1752-0509-7-7.
- Funahashi, A., Matsuoka, Y., Jouraku, A., *et al.*, 2008. CellDesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the IEEE* 96 (8), 1254–1265. doi:10.1109/JPROC.2008.925458.
- Gao, J., Aksoy, B.A., Dogrusoz, U., *et al.*, 2013. Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal. *Science Signaling* 6 (269), doi:10.1126/scisignal.2004088. [pl1-pl1].
- García-Campos, M.A., Espinal-Enríquez, Jesús, Hernández-Lemus, E., 2015. Pathway analysis: State of the art. *Frontiers in Physiology* 6. doi:10.3389/fphys.2015.00383.
- Glaab, E., Baudot, A., Krasnogor, N., Schneider, R., Valencia, A., 2012. EnrichNet: Network-based gene set enrichment analysis. *Bioinformatics* 28 (18), i451–i457. doi:10.1093/bioinformatics/bts389.
- Glaab, E., Baudot, A., Krasnogor, N., Valencia, A., 2010. Extending pathways and processes using molecular interaction networks to analyse cancer genome data. *BMC Bioinformatics* 11, 597. doi:10.1186/1471-2105-11-597.
- Gough, N.R., 2002. Science's Signal Transduction Knowledge Environment. *Annals of the New York Academy of Sciences* 971 (1), 585–587. doi:10.1111/j.1749-6632.2002.tb04532.x.
- Gray, K.A., Yates, B., Seal, R.L., Wright, M.W., Bruford, E.A., 2015. Genenames.org: The HGNC resources in 2015. *Nucleic Acids Research* 43 (Database issue), D1079–D1085. doi:10.1093/nar/gku1071.
- Han, H., Shim, H., Shin, D., *et al.*, 2015. TRRUST: A reference database of human transcriptional regulatory interactions. *Scientific Reports* 5. doi:10.1038/srep11432.
- Herwig, R., Hardt, C., Lienhard, M., Kamburov, A., 2016. Analyzing and interpreting genome data at the network level with ConsensusPathDB. *Nature Protocols* 11 (10), 1889–1907. doi:10.1038/nprot.2016.117.
- Hornbeck, P.V., Zhang, B., Murray, B., *et al.*, 2015. PhosphoSitePlus, 2014: Mutations, PTMs and recalibrations. *Nucleic Acids Research* 43 (D1), D512–D520. doi:10.1093/nar/gku1267.
- Huang, Y.J., Hang, D., Lu, L.J., *et al.*, 2008. Targeting the human cancer pathway protein interaction network by structural genomics. *Mol Cell Proteomics* 7 (10), 2048–2060. doi:10.1074/mcp.M700550-MCP200.
- Huang, C.C., Lu, Z., 2016. Community challenges in biomedical text mining over 10 years: Success, failure and the future. *Briefings in Bioinformatics* 17 (1), 132–144. doi:10.1093/bib/bbv024.

- Hung, J.-H., Whitfield, T.W., Yang, T.-H., *et al.*, 2010. Identification of functional modules that correlate with phenotypic difference: the influence of network topology. *Genome Biology* 11 (2), R23. doi:10.1186/gb-2010-11-2-r23.
- Jia, P., Zheng, S., Long, J., Zheng, W., Zhao, Z., 2011. dmGWAS: Dense module searching for genome-wide association studies in protein-protein interaction networks. *Bioinformatics* 27 (1), 95–102. doi:10.1093/bioinformatics/btq615.
- Kandasamy, K., Mohan, S., Raju, R., *et al.*, 2010. NetPath: A public resource of curated signal transduction pathways. *Genome Biology* 11 (1), R3. doi:10.1186/gb-2010-11-1-r3.
- Kanehisa, M., 1997. A database for post-genome analysis. *Trends in Genetics* 13 (9), 375–376. doi:10.1016/S0168-9525(97)01223-7.
- Kanehisa, M., 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 28 (1), 27–30. doi:10.1093/nar/28.1.27.
- Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M., 2016. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research* 44 (D1), D457–D462. doi:10.1093/nar/gkv1070.
- Karp, P.D., Paley, S.M., 1994. Representations of metabolic knowledge: pathways. In: *Proceedings of the International Conference on Intelligent Systems for Molecular Biology; ISMB*, 2, pp. 203–11. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/7584392>.
- Kemper, B., Matsuzaki, T., Matsuoka, Y., *et al.*, 2010. PathText: A text mining integrator for biological pathway visualizations. *Bioinformatics* 26 (12), i374–i381. doi:10.1093/bioinformatics/btq221.
- Komai, T., 1950. Semi-Allelic Genes. *American Naturalist* 84 (818), 381–392. doi:10.1086/281636.
- Kuperstein, I., Bonnet, E., Nguyen, H.-A., *et al.*, 2015. Atlas of Cancer Signalling Network: A systems biology resource for integrative analysis of cancer data with Google Maps. *Oncogenesis* 4 (7), e160. doi:10.1038/oncsis.2015.19.
- Kutmon, M., Riutta, A., Nunes, N., *et al.*, 2016. WikiPathways: Capturing the full diversity of pathway knowledge. *Nucleic Acids Research* 44 (D1), D488–D494. doi:10.1093/nar/gkv1024.
- Kutmon, M., van Iersel, M.P., Bohler, A., *et al.*, 2015. PathVisio 3: An Extendable Pathway Analysis Toolbox. *PLoS Computational Biology* 11 (2), doi:10.1371/journal.pcbi.1004085.
- Liberzon, A., Birger, C., Thorvaldsdóttir, H., *et al.*, 2015. The Molecular Signatures Database Hallmark Gene Set Collection. *Cell Systems* 1 (6), 417–425. doi:10.1016/j.cels.2015.12.004.
- Luo, Y., Riedlinger, G., Szolovits, P., 2014. Text mining in cancer gene and pathway prioritization. *Cancer Informatics* 13 (Suppl 1), 69–79. doi:10.4137/CIN.S13874.
- Ma, H., Sorokin, A., Mazein, A., *et al.*, 2007. The Edinburgh human metabolic network reconstruction and its functional analysis. *Molecular Systems Biology*. 3. doi:10.1038/msb4100177.
- Marbach, D., Lamparter, D., Quon, G., *et al.*, 2016. Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases. *Nature methods*. 1–44. doi:10.1038/nmeth.3799.
- Matsuoka, Y., Ghosh, S., Kikuchi, N., Kitano, H., 2010. Payao: A community platform for SBML pathway model curation. *Bioinformatics* 26 (10), 1381–1383. doi:10.1093/bioinformatics/btq143.
- Mi, H., Poudel, S., Muruganujan, A., Casagrande, J.T., Thomas, P.D., 2016. PANTHER version 10: Expanded protein families and functions, and analysis tools. *Nucleic Acids Research* 44 (D1), D336–D342. doi:10.1093/nar/gkv1194.
- Miwa, M., Ohta, T., Rak, R., *et al.*, 2013. A method for integrating and ranking the evidence for biochemical pathways by mining reactions from text. *Bioinformatics* 29 (13), i44–i52. doi:10.1093/bioinformatics/btt227.
- Moriya, Y., Itoh, M., Okuda, S., Yoshizawa, A.C., Kanehisa, M., 2007. KAAS: An automatic genome annotation and pathway reconstruction server. *Nucleic Acids Research* 35 (SUPPL.2), W182–W185. doi:10.1093/nar/gkm321.
- Moulos, P., Klein, J., Jupp, S., *et al.*, 2013. The KUPNetViz: A biological network viewer for multiple-omics datasets in kidney diseases. *BMC Bioinformatics* 14 (1), 1–12. doi:10.1186/1471-2105-14-235.
- Nishimura, D., 2001. BioCarta. *Biotech Software & Internet Report* 2 (3), 117–120. doi:10.1089/152791601750294344.
- Oliver, S., 2000. Guilt-by-association goes global. *Nature* 403 (6770), 601–603. doi:10.1038/35001165.
- Orchard, S., Kerrien, S., Abbani, S., *et al.*, 2012. Protein interaction data curation: The International Molecular Exchange (IMEx) consortium. *Nature Methods* 9 (6), 626. doi:10.1038/nmeth0612-626a.
- Osterman, A., Overbeek, R., 2003. Missing genes in metabolic pathways: A comparative genomics approach. *Current Opinion in Chemical Biology* 7 (2), 238–251. doi:10.1016/S1367-5931(03)00027-9.
- Paz, A., Brownstein, Z., Ber, Y., *et al.*, 2011. SPIKE: A database of highly curated human signaling pathways. *Nucleic Acids Research* 39 (SUPPL. 1), D793–D799. doi:10.1093/nar/gkq1167.
- Perfetto, L., Briganti, L., Calderone, A., *et al.*, 2016. 'SIGNOR: A database of causal relationships between biological entities. *Nucleic Acids Research* 44 (D1), D548–D554. doi:10.1093/nar/gkv1048.
- Petri, V., Jayaraman, P., Tutaj, M., *et al.*, 2014. The pathway ontology – updates and applications. *Journal of Biomedical Semantics* 5 (1), 7. doi:10.1186/2041-1480-5-7.
- Pinto, J.P., Reddy Kalathur, R.K., Machado, R.S.R., *et al.*, 2014. StemCellNet: An interactive platform for network-oriented investigations in stem cell biology. *Nucleic Acids Research* 42 (W1), W154–W160. doi:10.1093/nar/gku455.
- Port, J.A., Parker, M.S., Kodner, R.B., *et al.*, 2013. Identification of G protein-coupled receptor signaling pathway proteins in marine diatoms using comparative genomics. *BMC genomics* 14 (503), 1–15. doi:10.1186/1471-2164-14-503.
- Rahmati, S., Abovsky, M., Pastrello, C., Jurisica, I., 2017. pathDIP: An annotated resource for known and predicted human gene-pathway associations and pathway enrichment analysis. *Nucleic Acids Research* 45 (Database issue), D419–D426. doi:10.1093/nar/gkw1082.
- Ravikumar, K.E., Waghlikar, K.B., Liu, H., 2014a. Challenges in adapting text mining for full text articles to assist pathway curation. *ACM BCB 2014 – Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 551–558. doi:10.1145/2649387.2649444.
- Ravikumar, K.E., Waghlikar, K.B., Liu, H., 2014b. Towards pathway curation through literature mining – a case study using pharngkb. *Pac Symp Biocomput* 19, 352–363. doi:10.1142/9789814583220_0034.
- Romero, P., Wagg, J., Green, M.L., *et al.*, 2005. Computational prediction of human metabolic pathways from the complete human genome. *Genome Biology* 6 (1), R2. doi:10.1186/gb-2004-6-1-r2.
- Schaefer, C.F., Anthony, K., Krupa, S., *et al.*, 2009. PID: The Pathway Interaction Database. *Nucleic Acids Research* 37 (Database), D674–D679. doi:10.1093/nar/gkn653.
- Schnoes, A.M., Brown, S.D., Dodevski, I., Babbitt, P.C., 2009. Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. *PLoS Computational Biology* 5 (12), doi:10.1371/journal.pcbi.1000605.
- Simão, É.M., Cabral, H.B., Castro, M.A.A., *et al.*, 2010. Modeling the human genome maintenance network. *Physica A: Statistical Mechanics and its Applications* 389 (19), 4188–4194. doi:10.1016/j.physa.2010.05.051.
- Sreenivasiah, P.K., Rani, S., Cayetano, J., Arul, N., Kim, D.H., 2012. IPAVS: Integrated pathway resources, analysis and visualization system. *Nucleic Acids Research* 40 (D1), D803–D808. doi:10.1093/nar/gkr1208.
- Subramani, S., Kalpana, R., Monickaraj, P.M., Natarajan, J., 2015. HPIminer: A text mining system for building and visualizing human protein interaction networks and pathways. *Journal of Biomedical Informatics* 54, 121–131. doi:10.1016/j.jbi.2015.01.006.
- Sun, S., Dong, X., Fu, Y., Tian, W., 2012. An iterative network partition algorithm for accurate identification of dense network modules. *Nucleic Acids Research* 40 (3), e18. doi:10.1093/nar/gkr1103.
- Whirl-Carrillo, M., McDonagh, E.M., Hebert, J.M., *et al.*, 2012. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics* 92 (4), 414–417. doi:10.1038/clpt.2012.96.

- Wishart, D.S., Jewison, T., Guo, A.C., *et al.*, 2013. HMDB 3.0 – The Human Metabolome Database in 2013. *Nucleic Acids Research* 41 (D1), D801–D807. doi:10.1093/nar/gks1065.
- Xie, Z., Klionsky, D.J., 2007. Autophagosome formation: Core machinery and adaptations. *Nature cell biology* 9 (10), 1102–1109. doi:10.1038/ncb1007-1102.
- Yamamoto, S., Sakai, N., Nakamura, H., *et al.*, 2011. INOH: Ontology-based highly structured database of signal transduction pathways. *Database*. 2011. doi:10.1093/database/bar052.
- Yauch, R.L., Settleman, J., 2012. Recent advances in pathway-targeted cancer drug therapies emerging from cancer genome analysis. *Current Opinion in Genetics & Development* 22 (1), 45–49. doi:10.1016/j.gde.2012.01.003.
- Zhang, J., Zhang, S., 2017. Discovery of cancer common and specific driver gene sets. *Nucleic Acids Research* 45 (10), e86. doi:10.1093/nar/gkx089.
- Ziemann, M., Eren, Y., El-Osta, A., *et al.*, 2016. Gene name errors are widespread in the scientific literature. *Genome Biology* 17 (1), 177. doi:10.1186/s13059-016-1044-7.

Further Reading

- Chen, Q., Zhang, C., Goebel, R., 2014. In: Goebel, Randy, Tanaka, Yuzuru, Wahlster, Wolfgang (Eds.), *Intelligent strategies for pathway mining: models and pattern identification* (Lecture notes in Artificial Intelligence). Springer.
- Dong, W., Keibler, MA, Stephanopoulos, G., 2017. Review of metabolic pathways activated in cancer cells as determined through isotopic labeling and network analysis. *Metabolic Engineering*. doi:10.1016/j.ymben.2017.02.002.
- Emmert-Streib, F., Dehmer, M., Haibe-Kains, B., 2014. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Front. Cell Dev. Biol.* 38 (2), doi:10.3389/fcell.2014.00038.
- Saikat, Ch, Sarkar, R.R., 2015. Comparison of human cell signaling pathway databases – evolution, drawbacks and challenges. *Database (Oxford)*, 1–25. doi:10.1093/database/bau126.
- Stobbe, M.D., Houten, S.M., Jansen, G.A., *et al.*, 2011. Critical assessment of human metabolic pathway databases: A stepping stone for future integration. *BMC Systems Biology* 5, 165.

Relevant Websites

- <https://www.qiagenbioinformatics.com/products/ingenuity-pathway-analysis/>
QIAGEN.
- <http://www.sinobiological.com/signalingPathways.html>
Sino Biological.
- <http://www.tocris.com/signaling-pathways>
TOCRIS.

Biographical Sketch

Sara Rahmati is a PhD candidate in Medical Biophysics, University of Toronto, Canada. She is interested in developing algorithms to extract and analyze information from high-throughput biological data and sizable cellular networks. Her PhD focus is on mining different types of biological data and integrating them through statistical, machine learning, and big data visualization techniques. She applies these techniques to predict functions and context of activity of protein interactions and cellular pathways. She is also experienced in developing computational methods in predictive structural biology and working with open-source protein docking software. Sara finished her Master's degree in the School of Computing, Queen's University, Canada and her Bachelor of Science in Computer Engineering, Isfahan University of Technology, Iran.

Chiara Pastrello is a research associate at University Health Network, Toronto. She is interested in cancer genetics and epigenetics, that she formalized in her medical genetics residency – focusing on the prediction of pathogenicity of unclassified variants in mismatch repair genes. She is further interested in cancer systems biology, specializing in network analysis, and the curation of expression data for multiple databases maintained at Prof. Jurisica lab. Her current research interests include the study of microRNAs in cancer signaling to distant metastasis, as well as the application of network analyses to study molecules and pathways playing a central role in cancer.

Andrea E.M. Rossos is a third-year undergraduate student in a Bachelor of Science program in Honors Biology at McMaster University, Canada. She was a Princess Margaret Summer Student who has worked in the Jurisica Lab for summer 2016 and summer 2017 as a recipient of the Ian Lawson Van Toch Internship Award in Cancer Informatics. Andrea has been involved with data curation and development for projects such as mirDIP (microRNA Data Integration Portal), CDIP (Cancer Data Integration Portal) and ADIP (Arthritis Data Integration Portal). Andrea has also been involved with projects looking at the properties and diversity of biomolecule (miRNA, protein, etc.) databases. Andrea's current research interests are the prognostic and diagnostic roles of microRNAs and other biomolecules in cancer and disease.

I. Jurisica is Tier I Canada Research Chair in Integrative Cancer Informatics, Senior Scientist at Princess Margaret Cancer Centre, Professor at University of Toronto and Visiting Scientist at IBM CAS. He is also an Adjunct Professor at the School of Computing, Pathology and Molecular Medicine at Queen's University, Computer Science at York University, affiliate scientist at the Institute of Neuroimmunology, Slovak Academy of Sciences, and an Honorary Professor at Shanghai Jiao Tong University. Since 2015, he has also served as Chief Scientist at the Creative Destruction Lab, Rotman School of Management. He has published extensively on data mining, visualization and cancer informatics, including multiple papers in *Science*, *Nature*, *Nature Medicine*, *Nature Methods*, *J Clinical Oncology*, and has over 11,167 citations over the last 5 years. He has been included in Thomson Reuters 2016, 2015 & 2014 list of Highly Cited researchers, and The World's Most Influential Scientific Minds: 2015 & 2014 Reports.

Visualization of Biological Pathways

Giuseppe Agapito, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

This article deals with theories and algorithms for drawing and displaying networks, with particular emphasis on biological Pathways. Drawings are an attractive and effective way of conveying information. Networks drawing includes all aspects of visualizing structural relations between objects, not easily highlighting using other ways of representation. But, how to determine which automatic network generation produces a good or wrong visualization of the information? The best drawing for networks may not exist. Human perception of the same drawing changes from individual to individual and different application domains require different kinds of drawing. Thus, it is needed to define some constraints to take into account, in order to produce good results.

Biological Pathways Representations

Networks in general, as well as biological pathways, can be represented using a mathematical formalism called *Graph*. Graphs are abstract objects, with the ability to describe relations between objects. Formally, a graph G consists of a finite set of vertices (*objects*) and a finite set of edges (*relations*). The vertices of a graph are sometimes called nodes or dot; edges are sometimes called links, lines, arcs or connections. The set of vertices of G is denoted by $V(G)$ and the set of edges is denoted by $E(G)$. In a short form a graph can be write as: $G=(V, E)$. An graphical example of graph is depicted in Fig. 1.

The number of vertices of a graph G is its *order*, written as $|G|$; instead, its number of edges is denoted by $|G|$. A graph of order 0 or 1 is called trivial. A vertex v and an edge e are incident if $v \in e$; thus e is an edge of v . Two vertices are with an edge and are called *endvertices* or *ends*, and the edge is known as ends join. An edge (v, w) is usually represented as vw or (uv) . Two vertices x and y of G are adjacent, or neighbours, if (x, y) is an edge of G . Two edges $e \neq f$ are adjacent if they have an end in common. A Graph G is complete if and only if, all its vertices are pairwise adjacent. The *degree (or valency)* $dG(v)=d(v)$ of a vertex v is the number $|E(v)|$ of edges at v ; this is equal to the number of neighbours of v . A vertex of degree 0 is isolated. A *path* is a non-empty graph $P=(V, E)$ of the form $V=\{x_0, x_1, \dots, x_k\}$, $E=\{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\}$, where the x_i are all distinct. The vertices x_0 and x_k are linked by P and are called its ends; the vertices $x_1, \dots, x_{k-1}$ are the inner vertices of P . The number of edges of a path is its *length*.

If every edge has none direction (joins an unordered pair of vertices) the graph is called *undirected*, this means that, it is possible walk through the edge in both direction; the edge (x, y) is the same of (y, x) . Otherwise, if the edge has a direction associated the

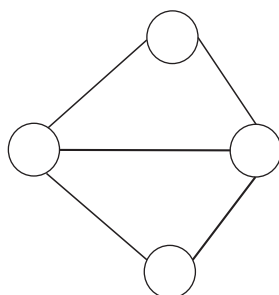


Fig. 1 Graph example.

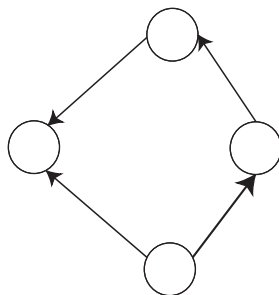


Fig. 2 Directed acyclic graph.

graph is called *directed graph* or *digraph*. Directed graphs are employed to model *signal transduction pathways* and *gene regulation networks*. Furthermore, if the digraph does not contain cycles is called **Directed Acyclic Graph (DAG)**. An example of DAG is shown in Fig. 2.

A cycle is an edge that joins a vertex to itself i.e. an edge (x, γ) with $x=\gamma$. Sometimes, it is needed to draw graphs where it is possible to have both edges with or without orientation. A graph extension able to represent and to manage correctly the two different types of edges is called **Mixed Graph**. A mixed graph G consists of three finite sets: the set of vertices $V(G)$, the set of undirected edges $E(G)$ and the set of directed edges $\vec{E}(G)$, and in short the graph can be written by means of the following triple: $G = (V, E, \vec{E})$. An example of mixed graph is shown in Fig. 3.

Cell signaling pathways are commonly represented using mixed graphs (Ma'ayan, 2009) in which some edges represent activation or inhibition relations, whereas other types of edges represent physical protein-protein interactions without a clear-cut directionality such as binding to anchors and scaffolds (Ma'ayan et al., 2005). The kinetics of a biochemical reaction is the rate that measures the effects of varying the conditions of the reactions under investigation. Kinetics of biochemical reactions can be represented through **Weighted Graph** (Bhalla and Iyengar, 1999). Thus, information may be attached to the nodes or arcs of a graph or both, where: the nodes are called labeled, and the arcs are called weighted. An example of a weighted graph is depicted in Fig. 4.

A weighted graph is, therefore, a particular type of labeled graph in which each edge is given a numerical weight (usually not negative). For example, in cell signaling: graphs can represent other properties of edges such as interaction weights by means of labels (Ma'ayan, 2009). When a graph contains only binary relations between the vertices, the graph is called: *Ordinary Graph* or *Simple Graph*, otherwise we have a *Hypergraph*. In Fig. 5 is shown an example of hyper graph.

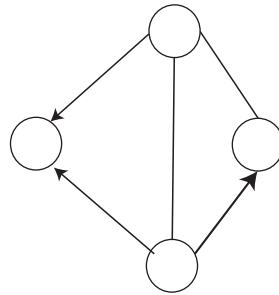


Fig. 3 Mixed graph.

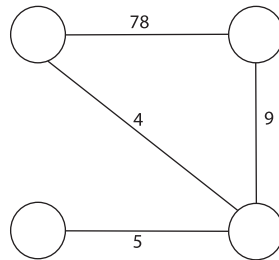


Fig. 4 Weighted Graph.

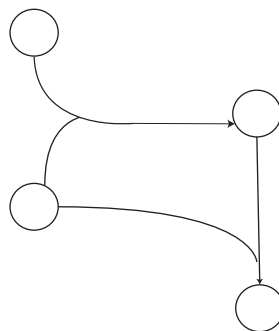


Fig. 5 Hyper graph.

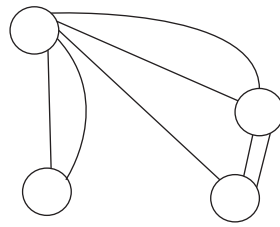


Fig. 6 Multi graph.

Hypergraphs offer a formalism that helps to overcome such conceptual limitations. As the name indicates, hypergraphs generalize graphs by allowing edges to connect more than two nodes, which may facilitate a more precise representation of biological knowledge (Klamt *et al.*, 2009). For example, in biology, the nodes typically describe proteins, metabolites, genes, or other biological entities, whereas the edges represent functional relationships or interactions between the nodes such as “binds to”, “catalyzes”, or “is converted to”. Many biological processes, however, are characterized by more than two participating partners and are thus not bilateral. A metabolic reaction such as $X + Y \rightarrow Z + T$ (involving four species), or a complex protein consisting of more than two proteins, are typical examples. As mentioned above, between a pair of nodes can exist more than one interaction (edge) with a different function and meaning. In this case, a pair of nodes are related in several different ways, from *multiple* or *parallel edges*, and in order to correctly represent this occurrence it is needed to use the *Multigraph* formalism.

A possible representation of the multi graph is depicted in Fig. 6.

Unlike a simple graph, a multigraph gives the formalism necessary to overcome the issues related to the management and representation of multiple edges connecting a pair of vertices. The close interplay between the p53/MDM2 pathways (Larsson and Girnita, 2005) is an interesting example where multigraph can help to describe and represent the interaction correctly among the nodes of the graph, with the advantage to simplify the phenomena to convey. Furthermore, if all multiple edges have a direction, the graph is called *Quiver* or *Multidigraph*. From the description above of the different types of existing graphs and their properties, it is possible to say that: graphs represent a very powerful tool to use in order to simplify the representation and the analysis of biological networks.

Paradigms for Graph Drawing

Depending on the context, the essential features of a graph should be drawn in different ways. Nodes may be drawn as dots, circles, boxes, or represented implicitly by their name labels. Edges may be drawn as straight lines, dotted lines, or bends. The information corresponding to the nodes and edges can be visualized using text labels at various positions in or next to a graph object, different colors, or other visual elements such as the thickness of lines, the size of boxes, etc. Moreover, a graph may be drawn in the 2D plane or three dimensions 3D. Once decided upon the representation, the question of how to convey the nodes and edges of a graph must be faced. A drawing, is a mapping of nodes and edges. But what distinguishes a good representation from a bad one? For example, drawing a *Entity Relation Diagram* with an algorithm designed to draw *Protein Protein Interaction Networks* will most likely produce a highly unsatisfactory picture. That is because an algorithm thought to draw a graph related to the protein-protein interactions network also uses some general knowledge on protein interactions. Also, for example, users would be highly confused if entities or relations were drawn in locations that highlight network topology features as hub or cliques, instead of locations that highlight in a remarkable way the connections among tables into the database. On the other hand, biologists would expect that the drawing reflects the results of the interaction; in order that, proteins involved in the same process are drawn next to each other for example. So for every particular graph drawing problem, we need especially customized algorithms which look beyond the structural properties of the given graphs and also use additional knowledge about the semantics behind the graph structure. This seems to imply the need to have a high number of different graph drawing algorithms. Fortunately, this is not true. There are a few general concepts and techniques which are powerful enough to cover a broad range of graph drawing applications. Thus, it is needed to specify some concepts able to describe the requirements of a nice drawing. First of all, it is necessary to define a drawing convention. Drawing convention defines the basic rules that the drawing must satisfy in order to be *admissible*. For example, in drawing biological pathway networks for a software application, it is possible to adopt the conventions where proteins are represented as a circle, whereas biochemical reactions are represented as a triangle, etc. Furthermore, there is the need to introduce a different type of edges able to describe every interaction among the different element that composes the pathway. From these examples, it is possible to note that the drawing conventions of a real-life application are very complex and involve the analysis of many details to take into account.

A short list of the most used drawing convention is the following.

- *polyline drawing*: each edge is drawn as polygonal chain (see Fig. 7);
- *straight line drawing*: each line is drawn as a straight line (see Fig. 8);
- *orthogonal drawing*: each edge is drawn as polygonal chain of alternating vertical and horizontal line (see Fig. 9);
- *planar drawing*: drawing without crossing edges (see Fig. 10).

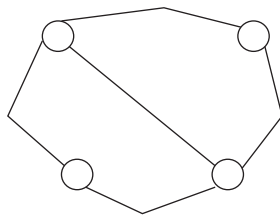


Fig. 7 Polyline drawing.

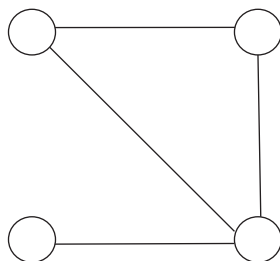


Fig. 8 Straight line drawing.

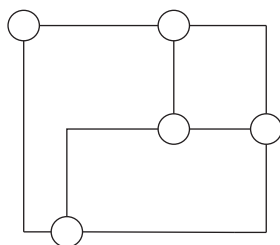


Fig. 9 Orthogonal drawing.

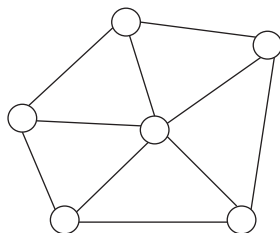


Fig. 10 Planar drawing.

The second criteria to take into account is to specify some properties of the drawing that would be applied as much as possible to improve readability. Finding a good arrangement is thus reduced to optimizing one or a few criteria. In algorithms where the primary goal is to produce layouts for human these criteria are called aesthetics criteria for a readable drawing; instead, technical criteria are on technological constraints such as: avoid crossing edge in drawing schema of electrical circuits. A short list of the most used aesthetics criteria for their practical importance is: *Crossing edge minimization*. Minimization of the total number of overlapping edges. If too many edges cross each other, the human eye cannot easily find out which nodes are connected by an edge. If a graph may be drawn without edge crossings (such graphs are called planar), this is very often preferable to a drawing with edge crossings. *Bend minimization*. Minimization of the total number of bends along the edges. This is an important aesthetics criterion for orthogonal layouts because the human eye can much more easily follow an edge with none or only a few bends than an edge wildly zig-zagging through the picture. *Maximum bends minimization*. Minimization of a maximum number of bends on edge. *Uniform bends minimization*. Minimization of the variance of the number of bends on edge. *Area minimization*. Minimization of the total area required for the draw. The ability to construct area efficient drawings is essential in practical visualization applications,

where saving space is a fundamental property. Moreover, a draw looks much better if the nodes and edges fill the space with homogenous density. *Symmetries*. If a graph contains symmetrical information, then it is important to reflect this symmetry in its arrangement. Symmetries can be further formalized introducing a mathematical model of symmetries in graph and drawing, but unfortunately, displaying symmetries is not an easy task. *Clustering*. When drawing huge networks such as biological or social networks gathering together similar nodes, can help to create a filter or reveal some of the graph's structure, and at the same time may increase the graph readability. *Total edge length minimization*. Minimization of the sum of the lengths of the edges. This criterion is meaningful if and only if the drawing strategy adopted prevent from being arbitrarily scaled down. *Maximum edge length minimization*. Minimization of the maximum length of an edge. This criterion is meaningful if and only if the drawing strategy adopted prevent from being arbitrarily scaled down. *Uniform edge length minimization*. Minimization of the variance of the lengths of the edges. *Angular resolution maximization*. Maximization of the smallest angle between two edges incident on the same vertex. This constraint is very significant in straight line drawing. The third criteria of a graph drawing algorithm is related to the computational efficiency. This is a crucial aspect of graphs drawing algorithms, that can ensure good performance (real time response), even for large graphs. This feature becomes a critical issue when graph analysis is in progress and users can navigate without delay the graph, as well as to guarantee real-time response. The aesthetics criteria are computationally hard, and on the other hand are in contrast with the other criteria, and it is very complicated to deal with all them at the same time. The reason for which there is the need to define approximation strategies and heuristics to get a compromise among the three criteria described before, in order to produce readable graphs.

Layout Algorithms

The purpose of data presentation is to reveal information buried in a set of data either to the exploration or just in its communication. A display is said to be effective if it conveys the real information in an easily comprehensible, yet precise, way. Graphs are used as abstractions in numerous fields of application. Consequently, the same structure can mean quite different things depending on the context. To effectively visualize a graph, the specific type of information it represents must govern the graphical design. The reason for which, graphs represent an important way to convey any information about connections, like computers, social, chemical or biological networks, which can be modeled as objects and relationship between these objects. These relationships are critical because they allow studying the problem in a graphic way, highlighting some characteristics not easily remarkable in other ways.

Visualization is an important way to analyze graphs. To provide visualization tools able to deal with networks their core has to comprises a layout algorithm, i.e. an algorithm able to decide how to present each object of the graph to the user. For this purpose, some layout algorithms were developed, in order to satisfy the drawing convention, some aesthetic criteria (the most useful related with the data to draw), in order to guarantee an excellent computational efficiency.

A concise description of the most known layout algorithms and their features (not go too far into the technical details) is presented below:

- **Random Layout Algorithm:** is very simple, because it arranges the graph nodes and edges in a random way on the screen. The advantage of this algorithm is its simplicity for both implementation and use, but on the other hand, it presents some disadvantages as a high number of cross-edges, a not optimum use of available space with large dimension graphs. Furthermore, the random layout is used to idealize architectures for dynamical models of genetic networks S.A. and [Kauffman \(1969\)](#).
- **Circular Layout Algorithm:** how the name suggests, the algorithm places the graph nodes in succession, one after the other follow in a circular arrangement. Thus, all nodes are at the same distance from the center of the screen, attempting to minimize overlapping vertices and cross edges (also known as overlapping edges). Although this algorithm can appear common, it is widely used to visualize complexes and pathways ([Tsay and Wu BL, 2009](#)).
- **Hierarchical Layout Algorithm (HLA):** has been developed by Sugiyama ([Sugiyama et al., 1981](#)) to reduce the number of cross edge, by arranging the graph nodes in different hierarchical groups. The HLA is scalable, efficient, and yield pleasant layouts for sparse graphs, even if the number of nodes is huge. A drawback of HLA is related to the computational efforts to minimize the production of overlapping edges. These kinds of problems belong to the class of NP-complete problems, that prevents the use of HLA for large graphs (with about a thousand of nodes and edges respectively). In addition, the hierarchical layout is used to dynamically visualize the complex data related to pathways ([Becker and Rojas, 2001](#)).
- **Fruchterman&Reingold and Eades&Peter Algorithms.** Fruchterman&Reingold algorithm ([Fruchterman and Reingold, 1991](#)) and Eades&Peter algorithm ([Eades, 1984](#)), also known as Spring Embedded Algorithm, belong to the class of Force-Directed Algorithms (FDA). FDA re-represent each node as an electrically charged item and each edge as a spring connecting two nodes. Thus, nodes with the same charge repel each other, while opposite nodes attracting each other, with an attraction/repulsing force due to the springs. These algorithms iteratively compute a distribution for each node determinate by the forces until the equilibrium is obtained. The computational time required to get the convergence to equilibrium grows rapidly with the size of the graph (that is, the number of nodes and edges) and the layout process becomes time-consuming for large graphs. The force-directed layout is commonly included in the majority of visualizations tools of biological networks ([Kojima et al., 2007](#)).
- **Kamada-Kawai Algorithm.** The algorithm developed by T. Kamada and S. Kawai is derived from the concept of theoretic distance between the nodes ([Kamada and Kawai, 1989](#)), i.e. the length of the shortest path among nodes. In this algorithm, the

forces among the nodes can be computed on their graph theoretic distances, determined by the lengths of shortest paths between each couple of nodes. The forces in the Kamada-Kawai algorithm, are represented as spring which force is directly proportional to the graph-theoretic distances. The Kamada-Kawai algorithm is more complex to encode on Fruchterman&Reingold or Eades&Peter algorithms but on the other hand provides excellent results.

- **Tree Layout Algorithm:** disposes graph as a tree without cycles, through a hierarchical organization of the nodes, producing clear and easy to understand visualization. The main advantage of the Tree Layout algorithm is its low complexity (generally, it presents linear performance in the number of nodes both for binary and *n*-ary trees) how explained in [Reingold and Tilford \(1981\)](#), making it efficient and scalable, even with large graphs. On the other hand, Tree Layout algorithm presents problems arranging a huge number of nodes in limited areas as the monitor. Finally, the Tree Layout algorithm provides a natural way to visualize the global structure of a complex pathways ([Tsay and Wu BL, 2009](#)).
- **Simulated Annealing Algorithm.** The name and inspiration come from the annealing in metallurgy, a technique involving heating and controlled cooling of material to increase the size of its crystals and reduce their defects. In fact, the structural properties of a solid depend on the rate of cooling (big crystals are obtained if the rate of cooling is slowly enough). The Simulated Annealing (SA) algorithm represents the space of the visualization problem as a set of states each one with associated energy, and it tries to find the state with minimum energy (i.e. below a threshold) that represents a potential solution. In particular, the SA may be applied to optimize different objective functions, for instance, a common goal could be minimizing the number of overlapping edges. Other works ([Zhang et al., 2010](#)) use SA to highlight the biological modules of the protein interaction networks by maximizing some quantitative measures such the modularity density, that is a measure for describing the modular organization of a network. The simulated annealing (SA) algorithm includes the following main steps:
 1. Random selection of a starting point among all possible solutions, although more sophisticated techniques can be used.
 2. Next solution selection and assessment, SA iteratively selects a point nearby of the current solution and determines whether the new point is better or worse than the current point. If the new point is better than the current one it becomes the next point otherwise, if the new point is worse than the current one, the algorithm can still choose the next point to explore, escaping from the local minimum (or maximum). This evaluation is done by an opportune assessment function.
 3. Stop criterion determines the end of the algorithm. The principal stop criteria are: the absence of changing of the value of the objective function, the average change in the value of the objective function is down respect to a defined threshold, or the number of iterations overcome the maximum number of iterations.

Simulated annealing has been successfully applied to the layout of general undirected graphs. The main drawback of SA algorithm is that the search space grows with the number of the nodes, becoming time-consuming with large graphs.

- **Cluster Layout Algorithm.** The Cluster Layout algorithm decrease the number of visible elements, reducing the visual complexity of the graph. Reducing the number of visible nodes and edges allow to improve the readability increasing in the same time the performance of layout and rendering ([Clemencon et al., 2011](#)). To get good results, clustering requires the definition of an appropriate metric that able to evaluate the similarity between nodes (or the distance). The metric can be *content-based* if it uses information associated with the content of the node, or *structured-based*, if it uses information on the structure of the graph. It is worthing to note that the Cluster Layout algorithm can be used to develop efficient functions such as filtering and search.
- **3D Layout Algorithm.** It is a popular technique to display graphs in 3D instead of 2D ([Ware et al., 1997](#)). The third dimension adds more available space, making easier the visualizzation and the navigation of large networks as well as biological pathways. As a consequence, 3D Layout algorithms have to enclose new features not present in 2D Layout algorithms, such as *transparency*, *depth*, *perspective*, navigation and so on. Furthermore, 3D Layout algorithms must allow to the users to interactively navigate the graph, modifying in real time the graph view based on the moving around the nodes. This improve the user navigation experience but in the same time, it adds further difficulty in handling the graph perspective.
- **Grid Layout Algorithm.** Grid Layout algorithm disposes the graph nodes on a 2-dimensional squared grid. Into the grid, the nodes are modeled as particles interacting together. Grid Layout algorithm model the graph as a system of interacting particles on a grid. The particles (nodes) interact according to a cost function which is designed based on the topological structure of the network. In such a system, closely related nodes attract each other, and remotely related nodes repulse each other. In [Li and Kurata \(2005\)](#), a Grid Layout algorithm to draw complex biochemical networks is proposed, helping researchers to gain insight into the complex network data.

See also: Alignment of Protein-Protein Interaction Networks. Biological Pathways. Computational Systems Biology. Natural Language Processing Approaches in Bioinformatics. Pathway Informatics. Visualization of Biomedical Networks

References

- Becker, M.Y., Rojas, I., 2001. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics* 17 (5), 461–467. Available at: <http://bioinformatics.oxfordjournals.org/content/17/5/461>.
- Bhalla, U.S., Iyengar, R., 1999. Emergent properties of networks of biological signaling pathways. *Science* 283 (5400), 381–387. Available at: <https://doi.org/10.1126/science.283.5400.381>.

- Clemençon, S., De Arazoza, H., Rossi, F., Tran, V.-C., 2011. Hierarchical clustering for graph visualization. In: Proceedings of 18th European Symposium on Artificial Neural Networks (ESANN 2011). Bruges, Belgique, pp. 227–232.
- Eades, P., 1984. A heuristic for graph drawing. *Congressus Numerantium* 42, 149–160.
- Fruchterman, T.M.J., Reingold, E.M., 1991. Graph drawing by force-directed placement. *Software: Practice Experience* 21, 1129–1164. Available at: <http://portal.acm.org/citation.cfm?id=137556.137557>.
- Kamada, T., Kawai, S., 1989. An algorithm for drawing general undirected graphs. *Information Processing Letters* 31, 7–15. Available at: [https://doi.org/10.1016/0020-0190\(89\)90102-6](https://doi.org/10.1016/0020-0190(89)90102-6).
- Kauffman, S.A., 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* 22 (3), 437–467. Available at: <http://www.sciencedirect.com/science/article/pii/0022519369900150>.
- Klamt, S., Haus, U.-U., Theis, F., 2009. Hypergraphs and cellular networks. *PLOS Computational Biology* 5 (5), e1000385. Available at: <https://doi.org/10.1371/journal.pcbi.1000385>.
- Kojima, K., Nagasaki, M., J. E., 2007. An efficient grid layout algorithm for biological networks utilizing various biological attributes. *BMC Bioinformatics* 6 (8), 76.
- Larsson, O., Girnita, A.G.L., 2005. Role of insulin-like growth factor 1 receptor signalling in cancer. *British Journal of Cancer*.
- Li, W., Kurata, H., 2005. A grid layout algorithm for automatic drawing of biochemical networks. *Bioinformatics* 21 (9), 2036–2042. Available at: <http://bioinformatics.oxfordjournals.org/content/21/9/2036>.
- Ma'ayan, A., 2009. Insights into the organization of biochemical regulatory networks using graph theory analyses. *The Journal of Biological Chemistry* 284 (9), 5451–5455. Available at: <https://doi.org/10.1074/jbc.r800056200>.
- Ma'ayan, A., Jenkins, S.L., Neves, S., *et al.*, 2005. Formation of regulatory patterns during signal propagation in a Mammalian cellular network. *Science (New York, N. Y.)* 309 (5737), 1078–1083. Available at: <https://doi.org/10.1126/science.1108876>.
- Reingold, E.M., Tilford, J.S., 1981. Tidier drawing of trees. *IEEE Transaction on Software Engineering SE-7* (2), 223–228.
- Sugiyama, K., Tagawa, S., Toda, M., 1981. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics* 11 (2), 109–12P5.
- Tsay, J.J., Wu BL, J.Y., 2009. Hierarchically organized layout for visualization of biochemical pathways. *Artificial Intelligence in Medicine* 48, 07–17.
- Ware, C., Franck, G., Parkhi, M., Dudley, T., 1997. Layout for visualizing large software structures in 3d. In: Proceedings of VISUAL '97. Springer Verlag, pp. 215–223.
- Zhang, S., Ning, X.-M., Ding, C., Zhang, X.-S., 2010. Determining modular organization of protein interaction networks by maximizing modularity density. *BMC Systems Biology* 4 (Suppl 2), S10. Available at: <http://www.biomedcentral.com/1752-0509/4?issue=S2/S10>.

Introduction

The notable improvements of bio-nano-technologies have allowed the fast production at affordable costs of increasing amounts of reliable data of different types. They comprise mainly biomolecular data which describe the observed biological processes, either physiological or pathological, from different perspectives; thus, considering them comprehensively can allow highlighting new insights on complex patho-physiological phenomena.

On the other hand, the development of information and communication technologies (ICT), as well as of data mining and machine learning techniques, makes easier processing the many generated data to automatically annotate them, and create repositories where to store them and make them publically accessible. In particular, the progress of sequencing techniques and the development of new technologies for automatic high-throughput analysis and annotation are producing a huge number of biomolecular data and knowledge.

Such multiplicity of very valuable data and information is sparsely stored in many heterogeneous and distributed databases that continuously increase in number, size and coverage (Galperin *et al.*, 2017). This creates a pressing need for integrative structures that are able to simplify the accessibility to the data and to decrease – if not to remove – both the variety of semantic representations of the data, and the data redundancy, which is unfortunately often present both in a single and among different biomedical-molecular databases (Louie *et al.*, 2007; Goble and Stevens, 2008). Integration of heterogeneous data is a challenging task, particularly in biology where the amount of data sources and data types increase rapidly, which involves issues such as data integrity, consistency, redundancy, connectivity, expressiveness and updatability. Integrative bioinformatics addresses these aspects by providing approaches and methods for the unified access to life science data, supporting searching and analysis of these data for the extraction of new knowledge from all the available data (Chen and Hofestädt, 2014).

After this introduction, this article first describes and discusses the main data integration approaches that have been proposed and used in bioinformatics. Then, it focuses on more advanced approaches for data integration, i.e., ontology-based, statistical and network-based, which allow for a semantic and biologically-driven integration of the data considered in bioinformatics and computational biology.

Data Integration Approaches

Many diverse approaches and implementations have been suggested to integrate distributed heterogeneous data, with notable applications in bioinformatics; they comprise *information linkage* (e.g., SRS (Etzold *et al.*, 1996), NCBI Entrez (Tatusova *et al.*, 1999)), *federated databases* (e.g., BioKleisli (Davidson *et al.*, 1997), BioMart (Smedley *et al.*, 2009)), *multi-databases* (e.g., TAMBIS (Stevens *et al.*, 2000)), *mediator-based solutions* (e.g., BioDataServer (Freier *et al.*, 2002)), Biomediator (Donelson *et al.*, 2004)) and *data warehousing* (e.g., EnsMart (Kasprzyk *et al.*, 2004), BioWarehouse (Lee *et al.*, 2006), Biozon (Birkland and Yona, 2006), GenoQuery (Lemoine *et al.*, 2008)). Despite many efforts have been performed, the problems regarding the creation, maintenance and updating of a public large and high-quality integration of originally distributed experimental and / or annotation data, which can aid discovering new biomedical knowledge via the reliable identification of missing annotations, keep being not fully solved yet.

Information Linkage

The *information linkage* approach for data integration aims to link together different information usually located in diverse resources without actually integrate them in a single place, but storing in a single place the references (i.e., links) to them in the location where they originally are; in this way, by following the integrated links it is possible to access the distributed data. It is one of the initial approaches used to integrate bioinformatics data. Its advantages are that it is quick to set up, easy to understand, and achieves a basic level of integration with minimal effort; conversely, its main disadvantages are that it does not include a cleaning and normalization of the data, does not have a way to directly integrate data from relational databases, and makes difficult to browse and mine the integrated data.

In the bioinformatics domain, some notable examples of implemented systems using this approach are the *Sequence Retrieval System* (SRS) and the *Entrez* system, both developed at the end of the last century.

SRS (Etzold *et al.*, 1996) is an information indexing and retrieval system that operates on databanks in a flat file or text format, such as some of those at the European Molecular Biology Laboratory (EMBL), supporting their data structure by providing special indices for implementing lists of sub-entities or hierarchically structured data fields. It provides a homogeneous interface to the entire flat file data within several biological databanks, retaining their original format; such interface enables users to access and query the contents of the databanks registered in the system, in order to navigate among them. Many elements are combined into the SRS system in order to

extend the power of normal retrieval systems, and to compete with the power of real databases, e.g., using a relational system, without compromising the rapidity. These components include languages for databank and syntax definition, a programmable parser, an indexing system, support for subentries, a system for taking advantage of links among databanks, and a query language.

The *Entrez* system (Tatusova *et al.*, 1999) was developed and implemented at the US governmental National Center for Biotechnology Information (NCBI) to manage large amounts of genome sequence data, and make them publicly available through the Web for data explorations beyond the flat file views that were traditional in those times. A highly enhanced and extended version of the *Entrez* system is still at present in use there. It is one of the most widely used interfaces to retrieve information from biological databases through simple text-based searches, linking and providing public access to the multiple and ample databases of biomolecular sequences, data and knowledge currently managed by the NCBI (Gibney and Baxevis, 2011). *Entrez* takes advantage of pre-existing logical relationships (expressed through Web-based links) between the individual entries in many public databases, which make sources interconnected so that any entry returned from one of the integrated sources also have related links to the other sources. These natural connections, mainly of biological nature, provide the ground for the method implemented in *Entrez*, through which all the information regarding a biological entity can be found avoiding sequential visits and queries to multiple distinct databases.

Federated Databases

The *federated database* approach is based on a meta-database management system (DBMS) able to map, transparently to the user, multiple independent databases into a single federated database. The individual databases composing the federation are interconnected by means of a computer network, and can be geographically distributed. In this approach each component database stays autonomous; thus, a federated database system is an alternative to merging the many single different databases composing the federation, which usually is an overwhelming job.

A federated, or virtual, database is hence the composition of all the component databases in the federated system; no integration of the data in the different constituent databases actually results from the data federation. Conversely, thanks to suitable data abstractions, a federated database system provides a homogeneous user interface that enables users to retrieve data from the multiple federated databases with a single query, despite the federated databases can be heterogeneous. To transparently support such functionality, a federated database system must be able to automatically subdivide a single user's query in sub-queries according to where the data addressed by the user's query are stored, submit each sub-query to the related relevant database of the federation, retrieve the answers of the individual databases, and compose them to provide an adequate result set answering the global user's query. Furthermore, since the federated databases can use different DBMS and query languages, the meta-DBMS must be able to apply wrappers to the sub-queries in order to translate each of them to the required and potentially different query languages.

Some notable examples of systems implemented with this approach in the bioinformatics domain are *BioKleisli* and *BioMart*.

BioKleisli (Davidson *et al.*, 1997) allows accessing various data sources important for the Human Genome Project (HGP) through the use of a framework that provides read access to multiple data sources containing biomedical complex structured data. Such framework includes three primary components: i) a powerful language to query and transform complex structured data, ii) an extensible architecture to implement the query primitives, and iii) optimization techniques that extend several known techniques to such more complex data types. The main limitation of *BioKleisli* is the lack of a global schema of the data in all the federated databases; thus, to express a query the user has to have a perfect knowledge of the data schema and structure of the federated sources integrated.

BioMart (Smedley *et al.*, 2009; Kasprzyk, 2011) is an open source data federation system originated at the European Bioinformatics Institute (EBI); it enables scientists to perform advanced integrated querying of biological data sources through a single Web interface regardless their physical location. Furthermore, *BioMart*'s capabilities are broadened through the integration with a set of widely used software packages, including DAS (Dowell *et al.*, 2001), Galaxy (Goecks *et al.*, 2010), Taverna (Hull *et al.*, 2006), BioConductor (Gentleman *et al.*, 2004), and Cytoscape (Shannon *et al.*, 2003). This makes possible using *BioMart* to address classical biological use cases, such as selection of single nucleotide polymorphism (SNP) for candidate gene screening, or annotation of microarray gene expression results. Being an easy to use, generic and scalable system it has been incorporated in well-known large data resources, such as Ensembl (Aken *et al.*, 2017), UniProt (The UniProt Consortium, 2017), COSMIC (Forbes *et al.*, 2017), and Reactome (Fabregat *et al.*, 2016).

Multi-Databases

A *multi-database* system is an integrated data system composed of a collection of autonomous database systems, typically heterogeneous, which contains a global descriptor for all the data in the component databases. It provides an integrated view through which the user can demand data management services on all the data in the integrated databases. Such integration requires the system to provide a mechanism for describing heterogeneous component databases, and to maintain an integrated description of all of them such that a user can view the multi-database as a single database.

The advantages of this approach, as well as of the federated database one, are that it is quick to be configured, its architecture is easy to understand (no knowledge of the domain is necessary), it achieves a basic level of integration with minimal effort, and it can wrap and plug in new data sources as they become available. On the other hand, it shows also several difficulties. In fact, it imposes constraints on original data sources, e.g., requiring that naming conventions across the integrated systems must be adhered to avoid inaccuracies in query results; it prevents performing query optimization across the multiple systems; if a source

system fails, the entire integrated application may fail; it is not readily suitable for data mining, or generic visualization tools; it relies on middleware technology, which often has performance (and reliability) problems.

One of the most renowned examples of multi-database systems in bioinformatics is *TAMBIS* (*Transparent Access to Multiple Bioinformatics Information Sources*) (Stevens *et al.*, 2000). Based on a knowledge model of the concepts and their relationships in molecular biology and bioinformatics, it allows biologists to ask rich and complex questions over a range of multiple bioinformatics information resources, such as databases and analysis tools, with a single interactive user interface. The used model of knowledge consists of a simple notion of the “things” that exist in bioinformatics and molecular biology, and how they are related to each other; this allows a biologist to browse it in order to see what can be expressed about “things” or concepts in molecular biology, and to join concepts together by their relationships so as to build new concepts.

Mediator-Based Solutions

The *mediator-based* approach defines a data integration architecture with two main components: wrappers and mediators (Fig. 1). The former ones are devoted to export information about data schemas, data and query processing capabilities of the data sources; the latter ones have the role of coordinating wrappers, by offering a unified view of all available data stored in a global schema, decomposing queries into smaller sub-queries executable by the wrappers, and lastly collecting the partial results and computing the final answer.

A mediator among the different data sources to be integrated is usually defined as the schema of a virtual database where the data sources are integrated; the mapping between each data source and the mediator (i.e., the global schema) must be defined. This involves several aspects to be addressed, including how to model the global schema, the sources, and the mappings between the two, and then how to verify the quality of the modeling process; how to construct the global schema, perform automatic wrapping of the sources (also considering existing limitations in the mechanisms to access the sources), and discover mappings between sources and global schema; how to process updates of the global schema and/or the sources; and how to answer queries expressed on the global schema and optimize query answering.

In the *mediator-based* approach, data extraction, cleaning, and reconciliation is at query run time, when a query expressed on the global schema must be automatically reformulated as a set of queries over the sources; the computed sub-queries are sent to the sources, and results are collected and assembled into the final answer. Such process must guarantee completeness of obtained answers, and efficiency in accessing the sources and computing the answers, which greatly depends on the adopted modeling of the data integration system; particularly, specification of the mapping between a source and the global schema can be source-centric, named *local-as-view* (LAV), when the sources are defined in terms of the global schema, or global-schema-centric, named *global-as-view* (GAV), when the global schema is defined in terms of the sources.

In the LAV method, the mapping gives a uniform query interface over the mediated global schema, directly supporting the query-answering by transforming a query over the global schema into specialized queries on the original data sources, which are modeled as views over the (nonexistent) mediated data schema. The burden of how to retrieve data from the sources is left to the query processor, but the benefit is that new sources can be added with much less effort than in a GAV system. Thus, the LAV method should be preferred when the mediated schema is more likely to change.

Conversely, in the GAV method the global schema is modeled as a set of views over the data sources where each element of the global schema is associated with a query over the sources; this simplifies answering queries over the mediated schema, leaving the complexity on implementing the mediator part regarding the retrieval of the data from the integrated sources. On the other hand, when a new source is integrated and/or an existing source changes its schema, the views for the mediated schema must be rewritten; thus, GAV is preferable when the sources unlikely change.

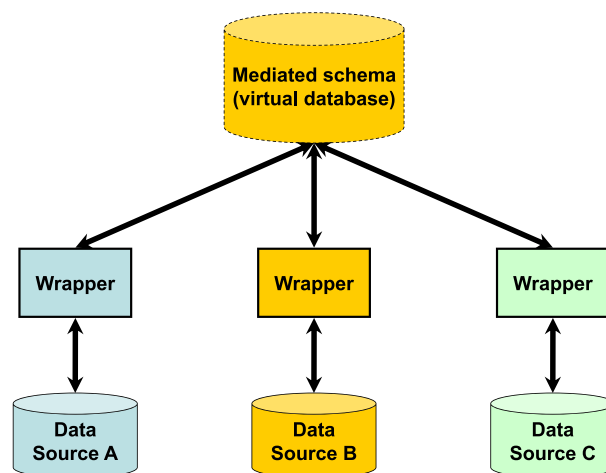


Fig. 1 Mediator-based data integration.

BioDataServer (Freier *et al.*, 2002) is an application example of the *mediator-based* approach in the bioinformatics domain; it proposes a data integration architecture implemented as a client-server system, which maps the user queries to the diverse and often limited query capabilities of the individual data sources. Given the very different query facilities, a common mechanism to process queries on the comprehensive database system is required; thus, the remote database access and integration process is reduced to a common set of simple operations that are executable by every wrapper.

Biomediator (Donelson *et al.*, 2004) is a similar system that performs data integration over multiple structured and semi-structured biological data sources. It uses a modular design and includes multiple components; one of them, the source knowledge base (SKB), contains i) all entities, attributes, and relationships in the mediated schema, ii) the catalog of all integrated data sources and the mediated schema elements mapping them, and iii) the mapping rules for bidirectional semantic translation of queries and source data flows. Notably, to describe the set of all integrated biological data and their interconnections the system uses an annotated mediated schema, and each user can define a completely customized mediated schema to specify his/her view of the set and express queries against it.

Data Warehousing

A usual approach in bioinformatics data integration is *data warehousing*, which consists of a priori integration and reconciliation of data extracted from multiple sources (Fig. 2). Typically, its construction is based on the extraction, transformation and load (ETL) process. During ETL, the initially dispersed data are first extracted from each of their original distinct source systems and temporarily stored in a staging database; then, they are transformed in order to both clean them (by detecting incomplete, incorrect, inaccurate, or irrelevant parts and correcting, replacing, or deleting them), and homogenize them for their integration. Finally, the cleaned and homogenized data are integrated in a data warehouse database, where they are arranged into hierarchical groups (called dimensions), facts, and aggregated facts, with a combination of facts and dimensions which is usually done according to a star data schema. To help users retrieving data from a usually complex data warehouse, on top of it data marts are typically created; they define subsets of the data in a comprehensive warehouse that regard specific topics or users.

Differently from other data integration approaches, in data warehousing the integration occurs at the lowest level, avoiding the need for the combination of queries and query results. All syntactic and semantic cleaning and harmonization of the data is performed a priori, during the data staging, allowing optimization of queries across data originally from multiple systems and faster query answering, independently on the original data sources, which remain untouched, and on their functioning; this is not possible in the other data integration approaches described. Furthermore, the data in a warehouse are readily suitable for data mining operations and generic visualization tools.

Besides these main advantages, data warehousing has also some issues, mainly regarding the time needed to perform the ETL process (which however occurs offline), and the domain knowledge required to correctly represent relationships among the integrated data. Furthermore, when original data sources are updated frequently, as it usually occurs for bioinformatics sources, the data in the warehouse become quickly obsolete; this demands to have in place suitable automatic procedures to regularly update easily the data in the warehouse. Additionally, due to the generally complex global data schema used, warehousing extension with the inclusion of new sources, or even maintenance against the evolution of the integrated sources, might be overwhelming, unless a modular and flexible data schema is adopted (Masseroli *et al.*, 2016).

Multiple bioinformatics systems based on the *data warehousing* approach have been and are being developed and actively used; some prominent examples are *EnsMart*, *Atlas*, *BioWarehouse*, *Biozon*, *GenoQuery*, and *GPKB*.

EnsMart (Kasprzyk *et al.*, 2004) is a generic data warehouse integrating and organizing data from individual databases into one query-optimized system for rapid and flexible querying of big biological data. It is based on the principle of creating a generic system from specific data sources, where disparate data can be integrated and interrogated in a flexible, efficient, unified, and domain-independent manner.

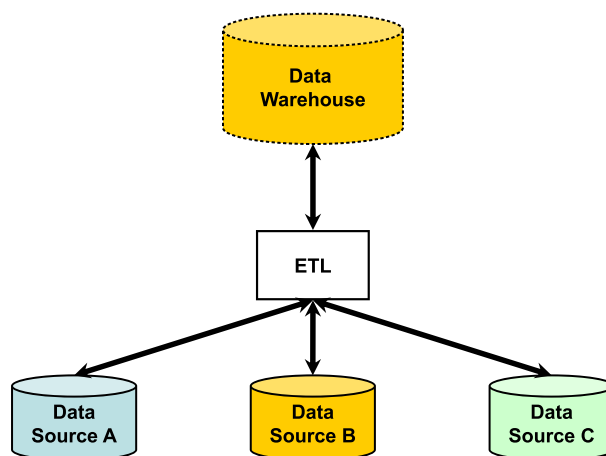


Fig. 2 Data warehousing data integration.

Atlas (Shah *et al.*, 2005) is a biological data warehouse that locally stores and integrates heterogeneous data (i.e., biological sequences, molecular interactions, homology information, functional annotations of genes, and biological ontologies) from multiple sources. It is based on individual relational data models for each of the integrated source data types, with data managed through Structured Query Language (SQL) calls implemented in a set of Application Programming Interfaces (APIs) for C++, Java, and Perl programming languages. These APIs include methods to parse and load source datasets into the Atlas database, and to facilitate data retrieval by offering end-users flexible, easy, integrated access to the data in the Atlas.

BioWarehouse (Lee *et al.*, 2006) is an evolving open source toolkit for constructing data warehouses; it combines different collections of bioinformatics databases within a single physical database management system (DBMS), in order to ease queries that span multiple databases. BioWarehouse aims to be a toolkit capable of supporting multiple alternative warehouses; its objective is to offer to every scientist the possibility to create a personal warehouse instance by combining different collections of databases. It also allows integrating local data with public bioinformatics sources in order to share experimental data with collaborators, or publish data online for the scientific community.

Biozon (Birkland and Yona, 2006) integrates multiple biological databases comprising heterogeneous types of biological data (e.g., DNA sequences, proteins, interactions and cellular pathways). Differently from other previous systems, besides warehousing existing data, it employs a single connected graph schema combined with a hierarchical ontology of data instances and relations; this allows computing and storing novel inferred data regarding similarity relationships and functional predictions of biological entities.

GenoQuery (Lemoine *et al.*, 2008) is a relational genomic warehouse adopting an original multi-tier architecture including a database layer and an entity layer; the latter one allows defining queries for searching instances of biological entities and their features in the distinct databases integrated in the warehouse, with no need of specifying in which original database search them.

The *Genomic and Proteomic Knowledge Base (GPKB)* (Masseroli *et al.*, 2016) employs a data warehouse approach to integrate a very high number of genomic and proteomic semantic annotations originally dispersed in many relevant sources (e.g., Entrez Gene, UniProt, IntAct, ExPASy Enzyme, GO, GOA, BioCyc, KEGG, Reactome, and OMIM). It innovatively adopts a flexible, modular, and multilevel global data schema founded on abstraction and generalization of the features of the data integrated; additionally, it implements several automatic processes supporting data integration and its maintenance, even when the integrated data sources evolve in number and structure besides content. These procedures also implement many techniques of data quality control to check the integrated data against a variety of possible errors and inconsistencies (Chisalberti *et al.*, 2010); thus, they avoid error propagation during their automatic processing, and guarantee consistency and quality of all integrated data, as well as their provenance tracking, which is a key factor for their subsequent proper processing and the interpretation of processing results. Furthermore, taking advantage of some defined logic rules and of the transitive relationships expressed by the biomolecular annotations integrated in the GPKB, novel annotations not existing in the integrated sources are automatically and reliably inferred and stored. Queries, although complex, can be easily expressed graphically on the knowledge base thanks to an intuitive Web interface, which also supports semantic query expansion and comprehensive explorative search of the integrated data, well supporting biomedical knowledge extraction.

The GPKB share with the TAMBS system the use of a model of knowledge in the world of bioinformatics and molecular biology; conversely, with the apparently similar BioWarehouse it shows several differences, mainly in the methodological choices about the global data schema adopted and the implemented data loading procedures. The GPKB data schema supports full data provenance tracking and is more flexible and easy to be expanded according the needs stemming from structural modifications of data sources previously integrated, or from the integration of new ones, which the implemented loading and updating procedures make easy.

Ontology-Based Data Integration

Proper integration of different data from multiple resources requires dealing with data heterogeneity, in order to effectively reconcile differences and recognize as equal, or referring to the same entity, data expressed in different ways. This is a complex task which involves recognizing the multiple different types of heterogeneities that can be present in the data and among their sources, and properly manage them.

While *system heterogeneity* refers to the use of different operating system and hardware platforms in distinct data sources, *structural heterogeneity* addresses the different structures where the data are stored in distinct data sources; in the case of using databases for data storing, the latter one refers to the different database models and/or schemas adopted. These heterogeneities are specifically dealt within each different data integration approach. On the other hand, *syntactic heterogeneity* concerns differences in the representation or format of the data, involving singular/plural and/or other lexical variations, which can be tackled using automatic natural language processing (NLP) techniques. Conversely, *semantic heterogeneity* involves differences in the interpretation of the meaning of the data, i.e., of the attributes of the data schema when data are stored in a database. Correctly managing and solving such heterogeneity requires the use of ontologies (i.e., formal models of representation with explicitly defined and named concepts and relationships linking them) able to consistently cover all the different concepts and the ways in which they are expressed in the integrated data (Lenzerini, 2002; Wache *et al.*, 2001).

In bioinformatics and computational biology, several relevant ontologies with high consistency and expressivity have been, or are being, developed, and they are widely adopted to express very numerous and publicly available annotations of multiple biological entities; this provides the ground to take advantage of such ontologies for ontology-based semantic integration of biological data and information which are usually sparsely stored. The most relevant of such ontologies are described in the following articles of this Encyclopedia.

Statistical and Network-Based Data Integration

When no logical, syntactic, lexical or semantic integration can be performed, advanced mathematical computational methods might be applied, particularly for the integration of numerical experimental results and of the information derived from them. Towards this goal, a variety of statistical univariate and multivariate techniques exists to identify variations or uniformities in datasets. However, traditional statistical methods have the limit of highlighting relationships among data only based on mathematical criteria (e.g., maximization of variance, or correlation), without taking into account correlations from biological origin. This suggests the combined use of different statistical and data mining techniques; the latter ones are available to discriminate groups of related data. They search for patterns in the data, usually returning several patterns which however include only a few significant ones. To automatically select them, novel pattern analysis methods are incorporated in existing data mining techniques, which are able to return a few results with a high likelihood of relevance.

Specific bioinformatics computational procedures, e.g., enrichment and pathway analyses, can be also used. They allow identifying biological entities that significantly share, or are related by, particular biological functions, on the basis of the information contained in public databases such as the Gene Ontology (GO) (Ashburner *et al.*, 2000) or the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Ogata *et al.*, 1999). Pathway analyses can also help in capturing the interactions among biological entities that the data to be integrated describe. When this occurs, network-based analyses and visualizations can be adopted to support better biologically-driven data integration. In particular, depending on the type of interactions described, a biological network can be represented with different types of graphs (i.e., mathematical structures), including directed or undirected graphs, directed acyclic graphs (DAGs), trees, forests, minimum spanning trees, Boolean networks, and Steiner trees (Ma'ayan, 2011); these can be visualized using a variety of strategies to make graph layouts more informative.

Conclusion

To answer usually complex biomedical-molecular questions, which typically require to evaluate data of multiple types, integrative bioinformatics provides multiple approaches to integrate heterogeneous biomolecular and biomedical data that are increasingly produced and available in numerous different sources. Among the several different methods and implementations proposed towards this goal, the most used are *mediator-based* and *data warehousing* solutions. The former ones typically provide online integration of the distributed data needed to answer a given request, requiring only limited overhead efforts; thus, they are particularly useful when data to be integrated are small (although extracted from individual big datasets) and vary frequently, with requests that are variable and require the most updated data to be answered. Conversely, the latter ones require previous static integration of the originally distributed heterogeneous data, which are then comprehensively queried; such initial overhead, which in addition makes the integrated data to become potentially obsolete with respect to those in the original data sources, pays off when the integrated data must be comprehensively evaluated, particularly when such evaluation requires complex and time consuming processing. Other advanced integrative bioinformatics approaches more recently used include *ontology-based*, *statistical* and *network-based* data integration, which proved effective for better biologically-driven data integration.

See also: Biological Database Searching. Biological Databases. Characterizing and Functional Assignment of Noncoding RNAs. Ecosystem Monitoring Through Predictive Modeling. Electronic Health Record Integration. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Binding Sites in DNA Sequences. Translational and Disease Bioinformatics. Translational Bioinformatics Databases

References

- Aken, B.L., Achuthan, P., Akanni, W., *et al.*, 2017. Ensembl 2017. *Nucleic Acids Research* 45 (D1), D635–D642.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *The Gene Ontology Consortium. Nature Genetics* 25 (1), 25–29.
- Birkland, A., Yona, G., 2006. BIOZON: A system for unification, management and analysis of heterogeneous biological data. *BMC Bioinformatics* 7 (70), 1–24.
- Chen, M., Hofestädt, R., 2014. Integrative Bioinformatics. In: Chen, M., Hofestädt, R. (Eds.), *Approaches in Integrative Bioinformatics*. Berlin, Heidelberg: Springer. Available at: http://dx.doi.org/10.1007/978-3-642-41281-3_1.
- Davidson, S.B., Overton, C., Tanen, V., Wong, L., 1997. BioKleisli: A digital library for biomedical researchers. *International Journal of Digital Library* 1, 36–53.
- Donelson, L., Tarczy-Hornoch, P., Mork, P., *et al.*, 2004. The BioMediator system as a data integration tool to answer diverse biologic queries. *Studies in Health Technology and Informatics* 107 (Pt 2), 768–772.
- Dowell, R.D., Jokerst, R.M., Day, A., Eddy, S.R., Stein, L., 2001. The distributed annotation system. *BMC Bioinformatics* 2, 7.
- Etzold, T., Ulyanov, A., Argos, P., 1996. SRS: Information Retrieval System for molecular biology data banks. *Methods in Enzymology* 266, 114–128.
- Fabregat, A., Sidiropoulos, K., Garapati, *et al.*, 2016. The Reactome pathway Knowledgebase. *Nucleic Acids Research* 44 (D1), D481–D487.
- Forbes, S.A., Beare, D., Boutselakis, H., *et al.*, 2017. COSMIC: Somatic cancer genetics at high-resolution. *Nucleic Acids Research* 45 (D1), D777–D783.
- Freier, A., Hofestädt, R., Lange, M., Scholz, U., Stephanik, A., 2002. BioDataServer: A SQL-based service for the online integration of life science data. *In Silico Biology* 2 (2), 37–57.
- Galperin, M.Y., Fernández-Suárez, X.M., Rigden, D.J., 2017. The 24th annual Nucleic Acids Research database issue: A look back and upcoming changes. *Nucleic Acids Research* 45 (D1), D1–D11.
- Gentleman, R.C., Carey, V.J., Bates, D.M., *et al.*, 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology* 5 (10), R80.
- Gibney, G., Baxevas, A.D., 2011. Searching NCBI databases using Entrez. *Current Protocols in Bioinformatics*. (Chapter 1, Unit 1.3).
- Ghisalberti, G., Masseroli, M., Tettamanzi, L., 2010. Quality controls in integrative approaches to detect errors and inconsistencies in biological databases. *Journal of Integrative Bioinformatics* 7 (3), 119.

- Goble, C., Stevens, R., 2008. State of the nation in data integration for bioinformatics. *Journal of Biomedical Informatics* 41, 687–693.
- Goecks, J., Nekutenko, A., Taylor, J., 2010. Galaxy Team. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology* 11 (8), R86.
- Hull, D., Wolstencroft, K., Stevens, R., *et al.*, 2006. Taverna: A tool for building and running workflows of services. *Nucleic Acids Research* 34 (Web Server issue), W729–W732.
- Kasprzyk, A., 2011. BioMart: Driving a paradigm change in biological data management. *Database* 2011, bar049.
- Kasprzyk, A., Keefe, D., Smedley, D., *et al.*, 2004. EnsMart: A generic system for fast and flexible access to biological data. *Genome Research* 14 (1), 160–169.
- Lee, T.J., Pouliot, Y., Wagner, V., *et al.*, 2006. BioWarehouse: A bioinformatics database warehouse toolkit. *BMC Bioinformatics* 7 (170), 1–14.
- Lemoine, F., Labedan, B., Froidevaux, C., 2008. GenoQuery: A new querying module for functional annotation in a genomic warehouse. *Bioinformatics* 24 (13), i322–i329.
- Lenzerini, M., 2002. Data integration: A theoretical perspective. In: *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 233–246. Madison, WI.
- Louie, B., Mork, P., Martin-Sanchez, F., Halevy, A., Tarczy-Hornoch, P., 2007. Data integration and genomic medicine. *Journal of Biomedical Informatics* 40 (1), 5–16.
- Ma'ayan, A., 2011. Introduction to network analysis in systems biology. *Science Signaling* 4 (190), tr5.
- Masseroli, M., Canakoglu, A., Ceri, S., 2016. Integration and querying of genomic and proteomic semantic annotations for biomedical knowledge extraction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13 (2), 209–219.
- Ogata, H., Goto, S., Sato, K., *et al.*, 1999. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 27 (1), 29–34.
- Shah, S.P., Huang, Y., Xu, T., Yuen, M.N., Ling, J., Ouellette, B.F., 2005. Atlas – A data warehouse for integrative bioinformatics. *BMC Bioinformatics* 6, 34.
- Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research* 13 (11), 2498–2504.
- Smedley, D., Haider, S., Ballester, B., *et al.*, 2009. BioMart – Biological queries made easy. *BMC Genomics* 10 (22), 1–12.
- Stevens, R., Baker, P., Bechhofer, S., 2000. TAMBIS: Transparent access to multiple bioinformatics information sources. *Bioinformatics* 16, 184–185.
- Tatusova, T.A., Karsch-Mizrachi, I., Ostell, J.A., 1999. Complete genomes in WWW Entrez: Data representation and analysis. *Bioinformatics* 15, 536–543.
- The UniProt Consortium, 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Wache, H., Vögele, T., Visser, U., *et al.*, 2001. Ontology-based integration of information – A survey of existing approaches. In: *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, pp. 108–117. Seattle, WA.

Further Reading

- Doan, A., Halevy, A., Ives, Z., 2012. *Principles of data integration*, first ed. Morgan Kaufmann. (ISBN: 9780124160446).
- Goble, C., Wolstencroft, K., 2014. Data integration. In: Hancock, J.M., Zveibil, M.J. (Eds.), *Dictionary of Bioinformatics and Computational Biology*. John Wiley and Sons, Inc.
- Kimball, R., Caserta, J., 2004. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering data*, first ed. Wiley Publishing. (ISBN: 978-0764567575).
- Zhang, Z., Bajic, V.B., Yu, J., Cheung, K.-H., Townsend, J.P., 2011. Data integration in bioinformatics: Current efforts and challenges. In: Mahdavi, M.A. (Ed.), *Bioinformatics – Trends and Methodologies*. InTech. ISBN 978-953-307-282-1 (Chapter 2).

Relevant Websites

- <http://www.biomart.org/>
BioMart.
- <http://www.bioinformatics.deib.polimi.it/GPKB/>
Genomic and Proteomic Knowledge Base (GPKB).
- <http://www.ncbi.nlm.nih.gov/Entrez/>
The Entrez system.

Biographical Sketch



Marco Masseroli received the Laurea degree in Electronic Engineering in 1990 from Politecnico di Milano, Italy, and the PhD degree in Biomedical Engineering in 1996, from the Universidad de Granada, Spain. He is associate professor in the Dipartimento di Elettronica, Informazione e Bioingegneria at Politecnico di Milano, and lecturer of Bioinformatics and Biomedical Informatics. He carried out research activity in the application of Information Technology to the medical and biological sciences in several Italian and international research centers. He has also been visiting professor in the Departamento de Anatomia Patologica, Facultad de Medicina at the Universidad de Granada, Spain, and visiting faculty at the Cognitive Science Branch of the National Library of Medicine, National Institute of Health, Bethesda. His research interests are in the area of bioinformatics and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughput genetic data. He is the author of more than 170 scientific articles, which have appeared in international journals, books, and conference proceedings.

Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines

Maria T Di Martino and Pietro H Guzzi, University “Magna Graecia” of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Background

Recent findings clearly support the concept that interconnecting miRNAs with TF genes and other upstream regulators (URs) may be of major help for the elucidation of different biologic events including oncogenetic and disease progression (Afshar *et al.*, 2014). Accordingly, several novel approaches have been developed for the investigation of relations among miRNAs, expression of target genes and transcription factors (van Iterson *et al.*, 2013; Di Martino *et al.*, 2015). To this aim, traditional gene-set-enrichment tools (Huang *da et al.*, 2009) have been partially substituted by more sophisticated models based on a system-level or semantic-based analysis (Guzzi *et al.*, 2012; Schulz *et al.*, 2011), and data interpretation by the integration with prior biological knowledge, into the workflow of analysis, for example, the use of causal networks such as networks that explicitate the causal relationships (Felciano *et al.*, 2013; Catlett *et al.*, 2013). Based on the ability to retrieve the direction of interactions among genes (including mutual feed-forward loop), causal analysis is more efficient than classic enrichment methods (Guzzi *et al.*, 2015). Here, we discuss general concepts related to production of omic data, specifically of gene and miRNA expression data, then, we present main available platforms of analysis and discuss future improvements and challenges of databases integration.

mRNA, miRNA, and Transcription Factor Interactions

As stated by the central dogma of molecular biology, genes guide protein synthesis through mRNA molecules. Since the information contained in genes cannot be directly translated into proteins, information is at first transcribed into mRNA molecules. Each molecule of mRNA encodes the information for one protein. The mRNA molecules migrate through the nuclear envelope to the cytoplasm, where they are translated in proteins through the mechanism of the protein synthesis which includes other cellular RNAs such as rRNA of ribosomes and tRNA. Finally, each mRNA is translated into a polymer of amino acids: a protein. In an ideal case, the quantity of mRNA molecules should be directly related to the quantity of the related protein. In such a way, the investigation of the quantity of mRNA through for example microarray technology, should support the investigation of the quantity of produced proteins. Unfortunately, as suggested by experimental evidences, this process is made complex by the presence of regulatory mechanisms that directly influence the level of proteins. In particular, recent findings have elucidated the role of two main classes of molecules that influence positively and negatively the protein synthesis: miRNA and TF (Burgess, 2015).

miRNA refers to a set of small RNA molecules composed of 21–23 nucleotides that do not encode any protein but participate as regulators in protein formation (Bartel, 2004). Recent studies demonstrated that miRNAs play an essential role in carcinogenesis because the disgregation of their activity may cause the development of tumor invasion and migration (Calin and Croce, 2006). miRNAs also act as a possible new target for molecular target therapy of various cancers (Gallo Cantafio *et al.*, 2016; Di Martino *et al.*, 2016; Gulla *et al.*, 2016; Di Martino *et al.*, 2014, 2013) or as drugs (Misso *et al.*, 2014; Di Martino *et al.*, 2012; Morelli *et al.*, 2015). Thus, there is an increasing interest for miRNA studies in clinical applications such as in serological diagnosis and molecular-targeted therapeutics (Di Martino *et al.*, 2016; Tagliaferri *et al.*, 2012). On the other side, TFs are protein modulators that regulate gene transcription through binding to the promoter region of target genes by their DNA-binding domains. In such a way, TFs may increase gene expression and consequently protein levels. Recent evidences support the idea that interconnecting miRNAs with TF genes may better elucidate different biological events. Consequently, several novel approaches have been devoted to the investigation of relations among miRNAs, expression of target genes and TFs (van Iterson *et al.*, 2013). To this aim, traditional gene-set-enrichment tools (Huang *da et al.*, 2009). have been partially substituted by more sophisticated models based on a system-level or semantic-based analysis (Guzzi *et al.*, 2012).

Interaction Databases

The interaction databases here reported fall into three main classes:

- Databases storing associations among miRNA and genes, i.e., storing which genes are targeted by miRNAs.
- Databases storing the associations among TF and genes, i.e., storing which genes are targeted by TFs.
- Databases storing the associations among TF and miRNA, i.e., storing which TFs are targeted by miRNAs.

All of these databases may store both confirmed associations, associations supported by experimental evidences, and predicted associations, or associations that are predicted by computational methods. The current scenario presents some main characteristics: (i) the number of confirmed associations is in general less than that of predicted ones, (ii) the number of false positives (i.e., not real associations) is considerable, and (iii) the level of overlap among databases is low. Consequently, all the approaches consider different data sources and integrate them in order to enhance the quality of considered associations.

The association among miRNAs and their target genes, for instance, genes up- or down-regulated, is currently an increasing research area. Currently, there exist different prediction software, that can predict possible genes regulated by a miRNA through machine learning approaches, and different technological platforms that are able to confirm these results in wet lab experiment (Rajewsky, 2006). As a result of the joint effort between both in silico and wet lab experiments, are now available several databases that store the association among miRNAs and mRNAs, such as Microcosm (Griffiths-Jones *et al.*, 2008), microRNA.org (Betel *et al.*, 2008), DIANA-microT (Maragkakis *et al.*, 2009), miRDB (Wang, 2008), PicTar (Krek *et al.*, 2005), PITA (Kertesz *et al.*, 2007), RNA22 (Miranda *et al.*, 2006), and TargetScan (Grimson *et al.*, 2007).

Similar to miRNAs, the complete enumeration of all the interactions among TF and genes is far to be complete. Thus, information stored into databases is quite incomplete. Main experiments used for discovering TF-gene relations are immunoprecipitations (ChIP) followed by sequencing (ChIP-seq) or by microarray hybridization (ChIP-chip) (Buck and Lieb, 2004). Both techniques enable a high-throughput discovery of relations, but usually, they also generate a large number of false positives (Qin *et al.*, 2011). In parallel to these techniques, we should recall main computational approaches for predicting TF and for retrieving resulting information from databases. For instance, the TRANSFAC database (Wingender, 2008) is one of the main resources of experimentally verified TF targets from publications or databases. Similarly, ChEA (Lachmann *et al.*, 2010), stores ChIP-seq and ChIP-chip data related to TF targets generated by different projects. The availability of different data sources with different reliabilities causes the need of integration of several of these methods and data to obtain comprehensive and accurate TF targets (Chen and Rajewsky, 2007).

The third main knowledge source is represented by databases storing information related to the regulation of miRNAs by TFs. The number of TF-miRNA regulation databases is lower than the number of the other two kinds of databases, because this approach is the youngest area of research. Examples of databases are TransmiR (Wang *et al.*, 2010), TransFac (Lenhard and Wasserman, 2002), TargetScan (Agarwal *et al.*, 2015), and PicTar (Krek *et al.*, 2005).

Tools and Pipelines

Here we report current available tools for integrative analysis approaches.

dChip-GemiNi (Gene and miRNA Network-Based Integration)

dChip-GemiNi (Gene and miRNA Network-based Integration) is a web server freely available for academic users which is able to integrate and analyze paired miRNA-mRNA expression data. The server side is written using the R programming language. Users may also download the source code for running it in a local environment. The ability of dChip-GemiNi has been tested by using some paired miRNA-mRNA datasets of solid cancers (liver, kidney, prostate, lung, and germ cell), and results are discussed in the paper of Yan *et al.* (2012).

The workflow of analysis that has been used to build dChip-GemiNi contains four steps:

1. First, publicly available databases as TargetScan for miRNA-mRNA association and TRANSFAC for TF binding sites) have been used to construct TF-miRNA-gene networks. Specifically, networks in which nodes are miRNA, genes, and TF, and edges represent the regulates relationship among them (e.g., a miRNA is connected to the target genes and a TF is connected to the target genes).
2. Then, experimental data including gene and miRNA expression profiles were collected from publicly databases (e.g., GEO).
3. Resulting networks (obtained in steps 1 and 2) were mined to extract significant motifs referred to as FFL motifs by small connected graphs in which there exist three different nodes (TF, miRNA, and mRNA).
4. Data managed in step 1 were used to further validate the statistical relevance of results through an ad hoc defined network motif score (NMS). The NMS is a function of multiple scores, including TF and miRNA binding scores to their target sequences, differential expression *P* values of the FFL components between normal and cancer tissues, and TF and miRNA's target enrichment in differentially expressed genes and miRNAs.

The workflow of analysis start from collection of experimental data, then two vectors of expression levels (one for mRNA and one for miRNA) are obtained from wet lab by analyzing two conditions, for instance, normal and cancer. Data may be paired as for sample, where were collected both mRNA and miRNA data for each sample, or unpaired such as for data belong to the same class but not to the same sample. Data are uploaded into the web server (see Relevant Websites section) so that an output of a list with significant FFLs, that are altered with respect to those used as the null model, will obtained. dChip-GemiNi is also able to individuate FFLs consisting of TFs (i.e., genes that are able to regulate the expression of other genes), miRNAs, and their common target genes. In such a way, new knowledge can be provided that is challenging to be discovered by the classical analysis. Experimental data are compared with respect to known associations among miRNAs, mRNAs, and TFs obtained from the literature and stored into the web server. TFs derived from literature are used as a null model to statistically rank predicted FFLs from the experimental data.

MAGIA2

MAGIA2 (Bisognin *et al.*, 2012), is the evolution of the MAGIA web tool for the integrated analysis of both genes and microRNA. MAGIA2 is deployed as a freely available web server (see Relevant Websites section). To build association networks, MAGIA2 uses

eight different databases of miRNA/mRNA associations: Microcosm (Griffiths-Jones *et al.*, 2008), mirna.org (Betel *et al.*, 2008), DIANA-microT (Maragkakis *et al.*, 2009), miRDB (Bisognin *et al.*, 2012), PicTar (Miranda *et al.*, 2006), PITA (Kertesz *et al.*, 2007), RNA22 (Miranda *et al.*, 2006), and TargetScan (Grimson *et al.*, 2007). Such predictors are used to build the null models, i.e., associations that are known by literature. Regarding TFs, MAGIA2 uses experimentally validated TF-miRNA interactions reported in mirGen2 (Alexiou *et al.*, 2010) and TransmiR (Wang *et al.*, 2010), whereas TF-gene interactions are obtained from ECRbase database (Loots and Ovcharenko, 2007).

The analysis through the MAGIA2 web server starts by uploading data, usually by the use of matrix for gene/transcripts and one for miRNA expression data. Data may belong to time series experiments in which for each sample there exists a pair miRNA/mRNA experiment (referred to as matched data), or a two-class experiment (referred to as un-matched data). Then, users select an association measure among mRNA and miRNA, i.e., a measure of relatedness among expression values. For matched experiments, MAGIA2 offers the following measures: Pearson linear correlation, Spearman rank-based correlation, and an association measure based on information theory for time series experiments (referred to as matched). Diversely, for un-matched design, only a meta-analysis is possible.

The choice among measures is strictly dependent on the characteristics of data: for non-normally distributed data and/or small sample size experiments (3–5), it is suggested to use Spearman correlation, which is a non-parametric rank-based linear measure, whereas for normally distributed data and medium-large sample size (more than 5), authors suggest the use of the Pearson linear correlation measure; finally, for large sample size (more than 20), it is suggested to use mutual information that is an information measure quantifying the mutual dependence of variables.

Conversely, for un-matched experiments, that are experiments in which samples are subdivided into two classes, the web server offers the meta-analysis approach that is based on the combination of *P values* of differential expression, distinctly for genes and miRNAs, across sample classes. The user may also choose which databases are used to extract associations from those explained so far. In case of choice of multiple databases, search results may contain their union or intersection. Finally, experimentally derived associations are compared to those contained in the databases.

mirConnX

mirConnX (Huang *et al.*, 2011) is based on a broader perspective with respect to the previous approaches since it uses a genome-wide approach. Unfortunately, it enables only the analysis of data of two organisms: human and mouse. The workflow of analysis is based on the comparison of two networks of associations among genes, TFs, and mRNAs.

The first network is used as a null model and is derived from the analysis of databases and literature. In this network, nodes are miRNAs, TFs, and genes, and edge connects two nodes when an association has been found. Examples of associations are a miRNA that regulates a gene or a TF, or a TF that regulates a gene. Edges are weighted, and the weight reflects the strength of the association. miRNA targets are derived by integrating results stored in PITA (Kertesz *et al.*, 2007), miRANDA (Enright *et al.*, 2003), TargetScan (Grimson *et al.*, 2007), RNAhybrid (Kruger and Rehmsmeier, 2006), Pictar (Krek *et al.*, 2005), TarBase (Papadopoulos *et al.*, 2009) and miRecords (Xiao *et al.*, 2009) databases. Similarly, associations among TF and genes are derived by integrating predictions stored in JASPAR (Xie *et al.*, 2005) and TRANSFAC (Wingender, 2008). The integration step is based on a mathematical model which is able to derive a value of confidence for each prediction that is used as a weight for the resulting edge. The network built from experimental data is obtained by analyzing all the possible pairwise interactions between TFs, miRNAs, and genes across the samples/replicates. The user may choose different measures of associations, both parametric and non-parametric, such as Pearson, Spearman, and Kendall.

Finally, the software integrates the two networks via a simple weighted sum function (*S*) producing a novel network in which edges, which are found in both networks, have a greater weight. Results are finally visualized by using a Cytoscape-based interface (Smoot *et al.*, 2011) and all feed-forward loops, and their neighbors are evidenced. In addition, other simple analyses can be executed including the ontology-based analysis.

IntegraMiR

IntegraMiR (Afshar *et al.*, 2014) is a novel approach of integration of data that is based on the workflow. It receives as input mRNA and miRNA expression data, obtained from samples that are subdivided into two classes (e.g., controls vs. cases). It starts by searching for differentially expressed genes and mRNAs between two conditions by using the Bioconductor package LIMMA (Smyth, 2005). This step produces two lists, one for differentially expressed genes and one for differentially expressed miRNAs. Moreover, IntegraMiR uses LIMMA package to perform gene set enrichment analysis (GSEA), taking into account known biological knowledge about these transcripts with the aim to derive biological significance of both changed and unchanged transcripts expression. Then, associations among mRNA and miRNA are derived considering their individual expression levels (i.e., considering pairs of mRNA-miRNA whose regulation is inversely correlated) or through their target interactions – via functional analysis through literature and databases. At the end of this step, IntegraMiR uses the TRANSFAC database (Wingender, 2008) to derive associations among TFs and mRNAs and the TransmiR database (Papadopoulos *et al.*, 2009) to derive associations among TFs and miRNAs. In particular, TransmiR focuses only on differentially expressed miRNA and mRNA. Thus, it can reconstruct FFLs whose members are differentially expressed. These FFLs are then organized considering the kind of deregulation and ranked by

using a statistical approach and visualized to the user (see the original publication for a complete list of results). The software is available for download (Afshar *et al.*, 2014).

Conclusions

As evidenced before, the TF-miRNA-mRNA association represents undoubtedly a main resource for elucidating gene expression regulation at a systems level. The complete determination of miRNA and TF targets will enable a more powerful and reliable analysis. Consequently, from a computational point of view, the integration of more data sources may improve the quality of analysis, since computational TF-miRNA regulatory networks are available for some genomes and diseases. Moreover, integrating TF-miRNA regulatory networks with other networks will aid in explaining how these networks regulate the biological processes and diseases at the systems level.

Acknowledgements

This article has been supported by the Italian Association for Cancer Research (AIRC), PI: PT. “Special Program Molecular Clinical Oncology – 5 per mille” n. 9980, 2010/15.

See also: Biological Database Searching. Biological Databases. Characterizing and Functional Assignment of Noncoding RNAs. Computational Pipelines and Workflows in Bioinformatics. Constructing Computational Pipelines. Ecosystem Monitoring Through Predictive Modeling. Electronic Health Record Integration. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Binding Sites in DNA Sequences. Profiling the Gut Microbiome: Practice and Potential. Translational and Disease Bioinformatics. Translational Bioinformatics Databases

References

- Afshar, A.S., Xu, J., Goutsias, J., 2014. Integrative identification of deregulated miRNA/TF-mediated gene regulatory loops and networks in prostate cancer. *PLOS ONE* 9 (6), e100806.
- Agarwal, V., *et al.*, 2015. Predicting effective microRNA target sites in mammalian mRNAs. *Elife* 4.
- Alexiou, P., *et al.*, 2010. miRGen 2.0: A database of microRNA genomic information and regulation. *Nucleic Acids Res* 38 (Database Issue), D137–D141.
- Bartel, D.P., 2004. MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell* 116 (2), 281–297.
- Betel, D., *et al.*, 2008. The microRNA.org resource: Targets and expression. *Nucleic Acids Res* 36 (Database Issue), D149–D153.
- Bisognin, A., *et al.*, 2012. MAGIA(2): From miRNA and genes expression data integrative analysis to microRNA-transcription factor mixed regulatory circuits (2012 update). *Nucleic Acids Res* 40 (Web Server Issue), W13–W21.
- Buck, M.J., Lieb, J.D., 2004. ChIP-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics* 83 (3), 349–360.
- Burgess, D.J., 2015. Molecular evolution: Decoupled transcription factor output? *Nat Rev Genet* 16 (1), 4.
- Calin, G.A., Croce, C.M., 2006. MicroRNA signatures in human cancers. *Nat Rev Cancer* 6 (11), 857–866.
- Catlett, N.L., *et al.*, 2013. Reverse causal reasoning: Applying qualitative causal knowledge to the interpretation of high-throughput data. *BMC Bioinformatics* 14, 340.
- Chen, K., Rajewsky, N., 2007. The evolution of gene regulation by transcription factors and microRNAs. *Nat Rev Genet* 8 (2), 93–103.
- Di Martino, M.T., *et al.*, 2012. Synthetic miR-34a mimics as a novel therapeutic agent for multiple myeloma: In vitro and in vivo evidence. *Clin Cancer Res* 18 (22), 6260–6270.
- Di Martino, M.T., *et al.*, 2013. In vitro and in vivo anti-tumor activity of miR-221/222 inhibitors in multiple myeloma. *Oncotarget* 4 (2), 242–255.
- Di Martino, M.T., *et al.*, 2014. In vitro and in vivo activity of a novel locked nucleic acid (LNA)-inhibitor-miR-221 against multiple myeloma cells. *PLOS ONE* 9 (2), e89659.
- Di Martino, M.T., *et al.*, 2015. Integrated analysis of microRNAs, transcription factors and target genes expression discloses a specific molecular architecture of hyperdiploid multiple myeloma. *Oncotarget* 6 (22), 19132–19147.
- Di Martino, M.T., *et al.*, 2016. Mir-221/222 are promising targets for innovative anticancer therapy. *Expert Opin Ther Targets* 20 (9), 1099–1108.
- Enright, A.J., *et al.*, 2003. MicroRNA targets in *Drosophila*. *Genome Biol* 5 (1), R1.
- Felciano, R.M., *et al.*, 2013. Predictive systems biology approach to broad-spectrum, host-directed drug target discovery in infectious diseases. *Pac Symp Biocomput.* 17–28.
- Gallo Cantafio, M.E., *et al.*, 2016. Pharmacokinetics and pharmacodynamics of a 13-mer LNA-inhibitor-miR-221 in mice and non-human primates. *Mol Ther Nucleic Acids* 5 (6).
- Griffiths-Jones, S., *et al.*, 2008. miRBase: Tools for microRNA genomics. *Nucleic Acids Res* 36 (Database Issue), D154–D158.
- Grimson, A., *et al.*, 2007. MicroRNA targeting specificity in mammals: Determinants beyond seed pairing. *Mol Cell* 27 (1), 91–105.
- Gulla, A., *et al.*, 2016. A 13 mer LNA-i-miR-221 inhibitor restores drug sensitivity in melphalan-refractory multiple myeloma cells. *Clin Cancer Res* 22 (5), 1222–1233.
- Guzzi, P.H., *et al.*, 2012. Semantic similarity analysis of protein data: Assessment with biological features and issues. *Brief Bioinform* 13 (5), 569–585.
- Guzzi, P.H., *et al.*, 2015. Analysis of miRNA, mRNA, and TF interactions through network-based methods. *EURASIP J Bioinform Syst Biol* 2015, 4.
- Huang, G.T., Athanassiou, C., Benos, P.V., 2011. mirConnX: Condition-specific mRNA-microRNA network integrator. *Nucleic Acids Res* 39 (Web Server Issue), W416–W423.
- Huang da, W., Sherman, B.T., Lempicki, R.A., 2009. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res* 37 (1), 1–13.
- Kertesz, M., *et al.*, 2007. The role of site accessibility in microRNA target recognition. *Nat Genet* 39 (10), 1278–1284.
- Krek, A., *et al.*, 2005. Combinatorial microRNA target predictions. *Nat Genet* 37 (5), 495–500.
- Kruger, J., Rehmsmeier, M., 2006. RNAhybrid: MicroRNA target prediction easy, fast and flexible. *Nucleic Acids Res* 34 (Web Server Issue), W451–W454.
- Lachmann, A., *et al.*, 2010. ChEA: Transcription factor regulation inferred from integrating genome-wide ChIP-X experiments. *Bioinformatics* 26 (19), 2438–2444.

- Lenhard, B., Wasserman, W.W., 2002. TFBS: Computational framework for transcription factor binding site analysis. *Bioinformatics* 18 (8), 1135–1136.
- Loots, G., Ovcharenko, I., 2007. ECRbase: Database of evolutionary conserved regions, promoters, and transcription factor binding sites in vertebrate genomes. *Bioinformatics* 23 (1), 122–124.
- Maragkakis, M., *et al.*, 2009. DIANA-microT web server: Elucidating microRNA functions through target prediction. *Nucleic Acids Res* 37 (Web Server Issue), W273–W276.
- Miranda, K.C., *et al.*, 2006. A pattern-based method for the identification of MicroRNA binding sites and their corresponding heteroduplexes. *Cell* 126 (6), 1203–1217.
- Misso, G., *et al.*, 2014. Mir-34: A new weapon against cancer? *Mol Ther Nucleic Acids* 3, e194.
- Morelli, E., *et al.*, 2015. Selective targeting of IRF4 by synthetic microRNA-125b-5p mimics induces anti-multiple myeloma activity in vitro and in vivo. *Leukemia* 29 (11), 2173–2183.
- Papadopoulos, G.L., *et al.*, 2009. The database of experimentally supported targets: A functional update of TarBase. *Nucleic Acids Res* 37 (Database Issue), D155–D158.
- Qin, J., *et al.*, 2011. ChIP-Array: Combinatory analysis of ChIP-seq/chip and microarray gene expression data to discover direct/indirect targets of a transcription factor. *Nucleic Acids Res* 39 (Web Server Issue), W430–W436.
- Rajewsky, N., 2006. microRNA target predictions in animals. *Nat Genet* 38 (Suppl.), S8–S13.
- Schulz, M., *et al.*, 2011. Retrieval, alignment, and clustering of computational models based on semantic annotations. *Mol Syst Biol* 7, 512.
- Smoot, M.E., *et al.*, 2011. Cytoscape 2.8: New features for data integration and network visualization. *Bioinformatics* 27 (3), 431–432.
- Smyth, G., 2005. Limma: Linear models for microarray data. In: Gentleman, R., Huber, W., Irizarry, R., Dudoit, S. (Eds.), *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York, NY: Springer, pp. 397–420.
- Tagliaferri, P., *et al.*, 2012. Promises and challenges of MicroRNA-based treatment of multiple myeloma. *Curr Cancer Drug Targets* 12 (7), 838–846.
- van Iterson, M., *et al.*, 2013. Integrated analysis of microRNA and mRNA expression: Adding biological significance to microRNA target predictions. *Nucleic Acids Res* 41 (15), e146.
- Wang, J., *et al.*, 2010. TransmiR: A transcription factor-microRNA regulation database. *Nucleic Acids Res* 38 (Database Issue), D119–D122.
- Wang, X., 2008. miRDB: A microRNA target prediction and functional annotation database with a wiki interface. *RNA* 14 (6), 1012–1017.
- Wingender, E., 2008. The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation. *Brief Bioinform* 9 (4), 326–332.
- Xiao, F., *et al.*, 2009. miRecords: An integrated resource for microRNA-target interactions. *Nucleic Acids Res* 37 (Database Issue), D105–D110.
- Xie, X., *et al.*, 2005. Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals. *Nature* 434 (7031), 338–345.
- Yan, Z., *et al.*, 2012. Integrative analysis of gene and miRNA expression profiles with transcription factor-miRNA feed-forward loops identifies regulators in human cancers. *Nucleic Acids Res* 40 (17), e135.

Relevant Websites

<http://www.canevolve.org/dChip-GemiNi/usergemi.php>
dChip-GemiNi Server.

<http://gencomp.bio.unipd.it/magia2/start/>
Magia2.

Biographical Sketch



Maria Teresa Di Martino, is lab assistant professor in modern technologies development at Magna Graecia University and Biologist at Medical Oncology and Translational Medical Oncology Units of Catanzaro Medical School. She had degree in Biological Science at University of Catania, Italy and the Speciality degree in clinical biochemistry and molecular biology at Medical School of Catanzaro. She worked as researcher at Biotechnology section of Agriculture Industrial Development, then moved at Catanzaro University, Clinical Chemistry Unit and Neonatal Screening, as diagnostic-clinical staff. After, the experience of guest researcher at Jerome Lipper Multiple Myeloma Center, Dana-Farber Cancer Institute and Harvard Medical School, Boston, She works at Magna Graecia University as diagnostic-clinical staff of Medical Oncology Units and as translational oncology investigator of Laboratory of Molecular Profiling and modern technologies development with main focus in innovative therapeutics. To this aim is actively pursued the application of computational strategies to analyze and integrate omic data to gain novel information in the molecular architecture of the cancer disease and derive novel biomarkers and new technological tools clinical translation. Recent researches focus on the development of preclinical models of multiple myeloma for the study of therapies based on miRNA mimics or inhibitors. Lastly, she develops experimental platform for research in pharmacogenomic profiling in order to test chemiotherapeutic toxicity and to identify biomarkers for toxicity prevention in different human cancers and in rare diseases.

Information Retrieval in Life Sciences

Pietro Cinaglia and Domenico Mirarchi, Magna Graecia University of Catanzaro, Catanzaro, Italy

Pierangelo Veltri, University "Magna Graecia" of Catanzaro, Catanzaro, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

In recent years, researchers are increasing their interest in applying data management techniques for information extraction from life related data sources. Information extraction is referred to the analysis of knowledge from several sources in order to understand the dynamics in life related processes, such as biological functions, environmental information, humans-environment relations. Usually, data are in raw format with heterogeneous structures; thus, a preprocessing phase is often performed to make the data usable to algorithms and tools (Belkin and Ingwersen, 2008), as well as to make them context-specific for the field of interest (Quan, 2007). This allows to extract knowledge from raw information (Bawden and Robinson, 2009), as well as to design data structures able to host biological related data (Nelson, 1994). Data can be collected in an Information Retrieval (IR) system; it is able to offer solutions for data integration of heterogeneous collections of data (Christopher *et al.*, 2008). Generally, it implements three main modules:

- indexing: that collects sets of concepts to generate a summary related to stored information,
- classification: used to relate information to indexes mechanisms,
- querying: used for information extraction through filtering, as well as keywords searching.

The indexation and the querying are often expressed using a syntax consisting of keywords through a key-value association. IR systems allow also to integrate data coming from different data sources, and thus heterogeneous both in format and structure (Stevens *et al.*, 2000). Data-Integration techniques are used to handle heterogeneous resources (e.g., physiology, pharmacology, and clinical data). IR can be enriched by ontologies; i.e., in Livingston *et al.* (2015) data is used for instance to correlate terms and concepts in life sciences. Ontologies can be used to enrich an existing set of data in order to build a relational model that allows an user to retrieve information of interest using query lang; e.g., Wild *et al.* (2012) uses SPARQL, a query language for graph based data. The semantics analysis of biomedical data is required to integrate information from sources with different schemes; an example may be related to the ontological databases, such as: SNOMED CT (Clinical Terms), International Classification of Disease (ICD) 9 and ICD-10, and Human Phenotype Ontology (HPO).

This article presents an overview of solutions used in life science to retrieve knowledge, as well as methods and algorithms for data analysis and integration based on machine-learning and data-mining techniques.

Data in Life Science

In life science, as well as in genomic and more generally in biomedicine, the data heterogeneity is growing in conjunction with analysis technologies that consequently produce output greatly diversified; think to the different technologies used to acquire data in high-throughput or micro-array experiments, such as: gene expression, single nucleotide polymorphisms (SNPs), copy number variation (CNV), proteomic and protein-protein interactions. All data need to be store using a model specific to the type of information that must be contained in it, and represented by a specific file format. An algorithm takes the advantage of each format to better process the input data and analyze them; generally, in life sciences the models are based on a key-value form. Several formats, with the related data models, are available for storage information in efficient way. Biomedical resources are often stored using flat file formats, such as: OWL, OBO, RDF, and CSV. These models stores the information using plain text; for instance, (i) in a CSV it consisting in tabular data where each line is a record that contains one or more values, (ii) in a OWL it consisting in classes of objects that describe taxonomies and classification networks. An ontology is a formal representation that defines a set of types, properties and interrelationships between the entities in a defined domain; furthermore, this terminology concerns a set of concepts that may be used to describe knowledge. An ontology may be of 'high level' or 'specific domain': the former is constituted by concepts that organize the upper parts of a knowledge base; instead, the latter may range in all the fields where syntax and semantics are permitted. Currently, most of the important biological ontologies use the OBO format representation. In Tirmizi *et al.* (2011) a methodology for translating OBO ontologies to OWL using the organization of the Semantic Web is presented. This solution produces an OWL Description Logics (DL) subset in order to improve the computational properties for efficiency and correctness. OWL is a Semantic Web language that supports query languages, and distributed infrastructure for information interchange; it is maintained by the World Wide Web consortium (W3C). In this scenario, the data integration techniques and ontologies can be used to perform a correlation between concepts and heterogeneous resources. Gene Ontology (GO) (Ashburner *et al.*, 2000) is a biological ontology used to retrieve information represented by terms consisting of: molecular functions, biological processes, and subcellular localizations. Analogously, Disease Ontology (DO) (DO Official WebSite, 2016) correlates human diseases through the cross-mapping of medical vocabularies (e.g., MeSH, ICD, NCI Thesaurus, SNOMED and OMIM); it contains several information consisting of genetic variations, phenotypes, proteins, and drugs. Generally, an ontology is used to retrieve additional information related to its

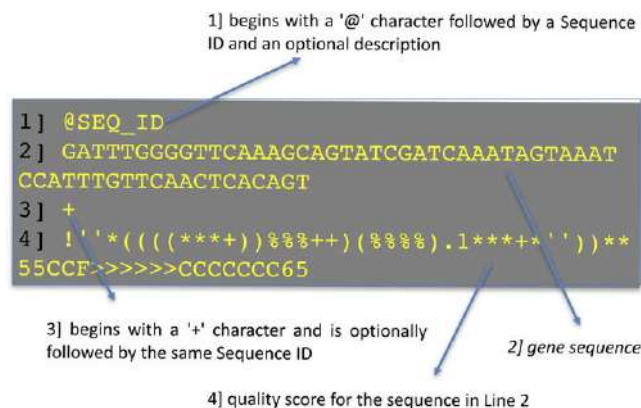


Fig. 1 FASTQ format.

content; it is modeled using taxonomies that represent, for instance, a genes list with the relations for each one (e.g., in Gene Ontology), or the definitions for a disease (e.g., Disease Ontology). Using data-integration methods a scientist can join this information in order to retrieve the molecular functions and its relations in biological processes. Furthermore, bioinformatics focuses its attention on life sciences related data in order to analyze the role of proteins, genes biological processes, and their functions for preventing and treating of diseases. Generally, in genomics the information (e.g., genes, nucleotide variations, biological sequences) are acquired using Microarray and Next Generation Sequencing (NGS) technologies, and subsequently processed by bioinformatics algorithms. A common file format is FASTQ, that is generally used as input for the alignment processes. FASTQ is composed of textual strings representing the gene sequences; it is composed by three main parts: (i) the first line begins with a '@' character followed by a Sequence ID and an optional description; (ii) the second is a long string characterized by symbols which denote the gene sequence; (iii) the latter represents the quality score for the sequence, its value is encoded using the American Standard Code for Information Interchange (ASCII). The sequence and its related quality score have the same length. Generally, between the second and third line a '+' char (optionally followed by the same Sequence ID) is also added to break the gene sequence content from the score; **Fig. 1** shows a summary model for the FASTQ format.

Techniques and Methods

A case study in data analysis for life sciences can be represented by a workflow which includes the following steps: (i) data retrieval from remote sources; (ii) heterogeneous data storage for local analysis in order to improve the performance and time consuming; (iii) heterogeneous resource modeling and indexation; (iv) knowledge extraction; optionally, (v) querying and (vi) statistical analysis. **Fig. 2** shown a diagram for the workflow.

Assuming that we want to integrate several 'omics information in order to combine large-scale biological data or to retrieve important biological insights, an algorithm for the multivariate analysis (Rohart *et al.*, 2017) of biological datasets could be implemented. Other approaches concern the ontology-based information systems for biomedicine which using semantic interoperability of biomedical resources in order to allow the storage and the querying of medical data. For instance, we can use the ontologies for knowledge management linking the terms related to: concepts, processes, and methods; similarly, in Jurisica *et al.* (2004) this approach was implemented using Data-integration and Semantic Similarity techniques in order to solve conflicts in heterogeneous data, as well as to map information on a formal schema. Furthermore, a Decision Support System (DSS) can be implemented to extract and analyze information from a large amount of data stored in a database; the latter could be based on a relation model using a Relational Database Management System (RDBMS). Assuming that we want to increase the effectiveness for an analysis a DSS could be used to provide a plan during a management activity through a Strategic Decisions-Making process. For instance, a similar approach is used in Cinaglia *et al.* (2015) to get an immediate online feedback from expert specialists, as well as to allow remote analysis of electrocardiogram (ECG). Generally, in clinical applications a DSS is implemented using Machine Learning (ML) techniques to support the identification of complex patterns through a multivariate statistical analysis, as well as to predict information using the knowledge extracted from larger dataset (Crippa *et al.*, 2017). In this scenario, a pattern can be represented as a sequence of items that can be modeled using a mathematical function. In a biological image a pattern is represented from a subset of numbers organized in a matrix; for instance if you're going to do a study in neurosciences, a pattern could be identified through an algorithm for pattern-recognition which integrates data across multiple variables in order to detect the alterations in brain's functionalities, as well as to investigate the regularities in the data using ML techniques. An ML algorithm can be performed using a training-set of known patterns (supervised learning), or it can be used to analyze directly an input dataset when no label was previously set (unsupervised learning) in order to discover unknown patterns during the prediction. Assuming that a physician needs to integrate several neuroimaging datasets for knowledge extraction, a virtual data-integrator could be implemented. Assuming that a study on the schizophrenia is being conducted, it could be used to retrieve data from remote sources (e.g., the datasets in input could

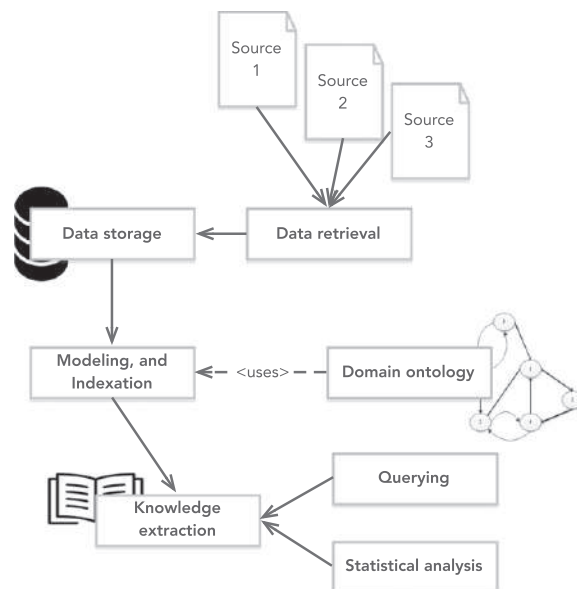


Fig. 2 Data in life sciences: Summary workflow.

be: HID, MRN's Collaborative Imaging and Neuroinformatics System, and XNAT) for a local analysis that integrates the information; this approach is used in SchizConnect (Ambite *et al.*, 2015; Wang *et al.*, 2016) where a query engine is also implemented, using an OGSA-DAI/DQP architecture (OGSA: Open Grid Services Architecture; DAI: Distributed Access and Integration; DQP: Distributed Query Processing), to offer a streaming data-flow evaluation engine and a distributed query engine.

Algorithms

In bioinformatics, data-mining and machine-learning solutions allow a scientist to obtain an interdisciplinary view from heterogeneous resources. Assuming that we want acquire information from ontological resources, an algorithm can be implemented to perform biological data analysis and integration. Briefly, it could be used for the management and integration of heterogeneous data acquired from ontological resources, such as: Disease Ontology, Gene Ontology, and DisGenet. For instance, we can imagine that a physician needs to retrieve additional informations for a disease using the ontologies. Thus, during the data-integration analysis the algorithm performs a pipeline composed from two phases. The first concerns the preprocessing: the data are integrated and organized in a single structure retrieving the information from the ontological datasets. Using this approach, the algorithm models a local ontology where novel relations among the concepts are established.

In Fig. 3 a diagram for the first phase is reported: Gene Ontology and Disease Ontology (OBO File Format) can be described as a Graph, where each Gene/Disease is a node that contains metadata (biological and descriptive information); DisGeNet (Pinero *et al.*, 2015) (TSV File Format) is described using a Tabular Structure (Cinaglia *et al.*, 2017).

The second phase consists in the analysis: when an user performs a request, the local data are investigated in order to return the information of interest. The Fig. 4 shows a possible output that concerns the genes involved for the disease, as well as the meta-data for each gene founded (e.g., description, biological function, biological processes and relations with other genes).

Similarly, in Cinaglia *et al.* (2017) an ontology-based system for clinical data integration and analysis is implemented. The literature presents several solutions to perform data analysis using the correlation between clinical ontologies and biological information.

Assuming that we want implement an algorithm for Hierarchical Hyperbolic SOM (H2SOM) clustering we needs to combine more features related to principles of machine learning and scientific-information in order to analyze the aspects of Toponome Imaging System (TIS) images in terms of space and collocation. This solution is able to resolve non-linear features (e.g., using dynamic interactive manipulation of the colors) and to organize clusters in a hierarchical structure; a similar approach is developer in Kolling *et al.* (2012). In a general-purpose case a toolkit can be used to speed up the algorithm implementation; for instance, Scikit-Learn (Pedregosa *et al.*, 2011) is a general purpose machine learning toolkit (based on Numpy and SciPy) to solve complex data analysis tasks in order to simplify the implementation of tool in Bioinformatics. Using NumPy and SciPy, a bioinformatics scientist is able to implement an algorithm for storage and manipulation of multi-dimensional array, as well as for the implementation of data structure and mathematical functions. During the implementation several Machine Learning approaches can be combined to improve the data analysis; for instance, in Abraham *et al.* (2014) supervised and unsupervised learning approaches are used to extract knowledge during neurological analysis.

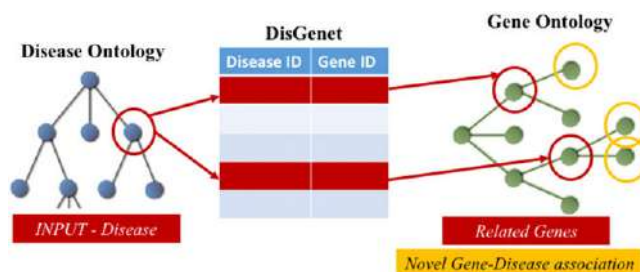


Fig. 3 Ontology relations for data-integration.

```
{
  "id": "6039",
  "name": "uveal melanoma",
  "def": "A uveal cancer that has_material_basis_in uvea pigment cells. [url:http\\://
cancergenome.nih.gov/cancersselected/UvealMelanoma, url:http\\://en.wikipedia.org/
wiki/Uveal_melanoma]",
  "synonym": ["\"melanoma of Uvea\" EXACT [NCI2004_11_17:C7712]", "\"melanoma
of Uvea\" EXACT [NCI2004_11_17:C7712]"],
  "umls": ["C0220633", "C0220633"],
  "is_a": "DOID:3479 ! uveal cancer",
  "genes": [{
    "geneid": "1964",
    "diseaseid": "C0220633",
    "score": 0.12081432,
    "genename": "EIF1AX",
    "description": "eukaryotic translation initiation factor 1A, X-linked",
    "diseasename": "Uveal melanoma",
    "sourceid": "CTD_human",
    "NofPmids": "3",
    "NofSnps": "0",
    "goterms": [{
      "genename": "EIF1AX",
      "go": "GO:0003743",
      "go_ref": "GO_REF:0000037"
    }, {
      "genename": "EIF1AX",
      "go": "GO:0005515",
      "go_ref": "PMID:21139563"
    }, {
      "genename": "EIF1AX",
      "go": "GO:0005515",
      "go_ref": "PMID:23435562"
    }
  ]
},
[...continue...]
```

Fig. 4 Data-integration analysis: Output.

The handling and manipulation of bioimaging data is required during the identification of patterns and features in order to check the presences of tumors or pathologies. A popular tool for medical image analysis is 3DSlicer (Kikinis *et al.*, 2014) (supported by the National Institutes of Health); it is used in research community to perform segmentation algorithms, ranging in complexity and automation.

Conclusions and Additional Reading

In life science the new analysis technologies produce a high amount of data that needs to be analyzed in order to extract knowledge; in this context, the bioinformatics helps to overcome compatibility issues due to the greatly diversification of the data models and types in order to obtain an interdisciplinary view. Data integration techniques, Machine-learning algorithms, as well as the data-mining approaches can be used to perform a correlation between the concepts within heterogeneous resources. An

overview for the techniques and methods for data retrieval in life sciences is presented, as well as the bioinformatics algorithms and tools for knowledge extractions with references to the data models and their structures.

See also: Biological Database Searching. Biological Databases. Characterizing and Functional Assignment of Noncoding RNAs. Ecosystem Monitoring Through Predictive Modeling. Electronic Health Record Integration. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Binding Sites in DNA Sequences. Profiling the Gut Microbiome: Practice and Potential. Translational and Disease Bioinformatics. Translational Bioinformatics Databases

References

- Abraham, A., Pedregosa, F., Eickenberg, M., *et al.*, 2014. Machine learning for neuroimaging with scikit-learn. *Front. Neuroinform.* 8, 14.
- Ambite, J.L., Tallis, M., Alpert, K., *et al.*, 2015. SchizConnect: Virtual data integration in neuroimaging. *Data Integr. Life Sci.* 9162, 37–51.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *The Gene Ontology Consortium. Nat. Genet.* 25 (1), 25–29.
- Bawden, D., Robinson, L., 2009. The dark side of information: Overload, anxiety and other paradoxes and pathologies. *J. Inf. Sci.* 35 (2), 180–191.
- Belkin, N., Ingwersen, P., Pejtersen, A.M., 2008. Introduction to information retrieval. In: *Proceeding of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1(1), pp. 11–15. Copenhagen, Denmark: ACM.
- Christopher, D.M., Prabhakar, R., Hinrich, S., 2008. *Introduction to Information Retrieval*, 1. Cambridge University Press. pp. 11–15.
- Cinaglia, P., Tradigo, G., Guzzi, P.H., Veltri, P., 2015. Design and implementation of a telecardiology system for mobile devices. *Interdiscip. Sci.* 7 (3), 266–274.
- Cinaglia, P., Veltri, P., Cannataro, M., 2017. eMiRo: An ontology-based system for clinical data integration and analysis. In: *Proceedings of the Italian Symposium on Advanced Database Systems. SEBD*.
- Crippa, A., Salvatore, C., Molteni, E., *et al.*, 2017. The utility of a computerized algorithm based on a multi-domain profile of measures for the diagnosis of attention deficit/hyperactivity disorder. *Front. Psychiatry* 8, 189.
- DO Official WebSite, 2016. Disease Ontology.
- Jurisdica, I., Mylopoulos, J., Yu, E., 2004. Ontologies for knowledge management: An information systems perspective. *Knowl. Inf. Syst.* 6, 380401.
- Kikinis, R., Pieper, S.D., Vosburgh, K.G., 2014. 3D slicer: A platform for subject-specific image analysis, visualization, and clinical support. In: *Intraoperative Imaging and Image-Guided Therapy*, pp. 277–289.
- Kolling, J., Langenkamper, D., Abouna, S., Khan, M., Nattkemper, T.W., 2012. WHIDE-a web tool for visual data mining colocation patterns in multivariate bioimages. *Bioinformatics* 28 (8), 1143–1150.
- Livingston, K.M., Bada, M., Baumgartner, W.A., Hunter, L.E., 2015. KaBOB: Ontology-based semantic integration of biomedical databases. *BMC Bioinform.* 16, 126.
- Nelson, M.R., 1994. We have the information you want, but getting it will cost you!: Held hostage by information overload. *Crossroads – Special Issue on the Internet*, 1(1), pp. 11–15.
- Pedregosa, F., Varoquaux, G., Gramfort, A., *et al.*, 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Pinero, J., Queralt-Rosinach, N., Bravo, A., *et al.*, 2015. DisGeNET: A discovery platform for the dynamical exploration of human diseases and their genes. *Database (Oxford)*.
- Quan, D., 2007. Improving life sciences information retrieval using semantic web technology. *Brief. Bioinform.* 8 (3), 172–182.
- Rohart, F., Gautier, B., Singh, A., Le Cao, K.A., 2017. mixOmics: An R package for 'omics feature selection and multiple data integration. *PLOS Comput. Biol.* 13 (11), e1005752.
- Stevens, R., Baker, P., Bechhofer, S., *et al.*, 2000. TAMBIS: Transparent access to multiple bioinformatics information sources. *Bioinformatics* 16 (2), 184–185.
- Tirmizi, S.H., Aitken, S., Moreira, D.A., *et al.*, 2011. Mapping between the OBO and OWL ontology languages. *J. Biomed. Semant.* 2 (Suppl. 1), S3.
- Wang, L., Alpert, K.I., Calhoun, V.D., *et al.*, 2016. SchizConnect: Mediating neuroimaging databases on schizophrenia and related disorders for large-scale integration. *Neuroimage* 124 (Pt B), 1155–1167.
- Wild, D.J., Ding, Y., Sheth, A.P., *et al.*, 2012. Systems chemical biology and the Semantic Web: What they mean for the future of drug discovery research. *Drug Discov. Today* 17 (9–10), 469–474.

Biographical Sketch

Pietro Cinaglia is PhD Student in Bioinformatics applied to Neurosciences at University of Catanzaro, Italy. His research activity concerns the study of Bioimaging and Bioinformatics solutions in Neurosciences, as well as the development and implementation of algorithms for biomedical data analysis and integration. He received his master degree cum laude in Biomedical Engineering at University of Catanzaro. From 2014–2016 he worked as software engineer at Magna Graecia of Catanzaro in several research projects.

Domenico Mirarchi is PhD Student of Magna Graecia University of Catanzaro since 2015. In 2014 worked as a research fellow at Magna Graecia University of Catanzaro on a “smart cities” project. During this period he has been involved in the design and implementation of algorithms. He received his master degree in 2011 in Biomedical Engineer.

Pierangelo Veltri is an associate professor in Computer Science at the Magna Graecia University of Catanzaro, Italy. He obtained his PhD in Computer Science from the University of Orsay (Paris XI), where he was researcher until 2002. He was researcher at INRIA, Pars from 1998 to 2002. From 2002–2012 was Assistant Professor in computer science and bioinformatics, and from 2012 Associate Professor.

**ENCYCLOPEDIA OF
BIOINFORMATICS AND
COMPUTATIONAL BIOLOGY**

ENCYCLOPEDIA OF BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

EDITORS IN CHIEF

Shoba Ranganathan

Macquarie University, Sydney, NSW, Australia

Michael Gribskov

Purdue University, West Lafayette, IN, United States

Kenta Nakai

The University of Tokyo, Tokyo, Japan

Christian Schönbach

*Nazarbayev University, School of Science and Technology, Department of Biology,
Astana, Kazakhstan*

VOLUME 2

Topics

Bruno Gaeta

University of New South Wales, Sydney, Australia



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge MA 02139, United States

Copyright © 2019 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers may always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-12-811414-8

For information on all publications visit our website at <http://store.elsevier.com>

		Working together to grow libraries in developing countries
www.elsevier.com • www.bookaid.org		

Publisher: Oliver Walter
Acquisition Editor: Sam Crowe
Content Project Manager: Paula Davies
Associate Content Project Manager: Ebin Clinton Rozario
Designer: Greg Harris

Printed and bound in the United States

EDITORS IN CHIEF



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. She hosted the Macquarie Node of the ARC Centre of Excellence in Bioinformatics (2008–2013). She was elected the first Australian Board Director of the International Society for Computational Biology (ISCB; 2003–2005); President of Asia-Pacific Bioinformatics Network (2005–2016) and Steering Committee Member (2007–2012) of Bioinformatics Australia. She initiated the Workshops on Education in Bioinformatics (WEB) as an ISMB2001 Special Interest Group meeting and also served as Chair of ICSB's Education Committee. Shoba currently serves as Co-Chair of the Computational Mass Spectrometry (CompMS) initiative of the Human Proteome Organization (HuPO), ISCB and Metabolomics Society and as Board Director, APBioNet Ltd.

Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books as well as articles for the 2013 Encyclopedia of Systems Biology. She is currently an Editor-in-Chief of the Encyclopedia of Bioinformatics and

Computational Biology and the Bioinformatics Section Editor of the Reference Module in Life Science as well as an editorial board member of several bioinformatics journals.



Dr. Gribskov graduated from Oregon State University in 1979 with a Bachelors of Science degree (with Honors) in Biochemistry and Biophysics. He then moved to the University of Wisconsin-Madison for graduate studies focused on the structure and function of the sigma subunit of *E. coli* RNA polymerase, receiving his Ph.D. in 1985. Dr. Gribskov studied X-ray crystallography as an American Cancer Society post-doctoral fellow at UCLA in the laboratory of David Eisenberg, and followed this with both crystallographic and computational studies at the National Cancer Institute. In 1992, Dr. Gribskov moved to the San Diego Supercomputer Center at the University of California, San Diego where he was lead scientist in the area of computational biology and an adjunct associate professor in the department of Biology. From 2003 to 2007, Dr. Gribskov was the president of the International Society for Computational Biology, the largest professional society devoted to bioinformatics and computational biology. In 2004, Dr. Gribskov moved to Purdue University where he holds an appointment as a full professor in the Biological Sciences and Computer Science departments (by courtesy). Dr. Gribskov's interests include genomic and transcriptomic analysis of model and non-model

organisms, the application of pattern recognition and machine learning techniques to biomolecules, the design and implementation of biological databases to support molecular and systems biology, development of methods to study RNA structural patterns, and systems biology studies of human disease.



Kenta Nakai received the PhD degree on the prediction of subcellular localization sites of proteins from Kyoto University in 1992. From 1989, he has worked at Kyoto University, National Institute of Basic Biology, and Osaka University. From 1999 to 2003, he was an Associate Professor at the Human Genome Center, the Institute of Medical Science, the University of Tokyo, Japan. Since 2003, he has been a full Professor at the same institute. His main research interest is to develop computational ways for interpreting biological information, especially that of transcriptional regulation, from genome sequence data. He has published more than 150 papers, some of which have been cited more than 1,000 times.



Christian Schönbach is currently Department Chair and Professor at Department of Biology, School of Science and Technology, Nazarbayev University, Kazakhstan and Visiting Professor at International Research Center for Medical Sciences at Kumamoto University, Japan. He is a bioinformatics practitioner interfacing genetics, immunology and informatics conducting research on major histocompatibility complex, immune responses following virus infection, biomedical knowledge discovery, peroxisomal diseases, and autism spectrum disorder that resulted in more than 80 publications. His previous academic appointments included Professor at Kumamoto University (2016–2017), Nazarbayev University (2013–2016), Kazakhstan, Kyushu Institute of Technology (2009–2013) Japan, Associate Professor at Nanyang Technological University (2006–2009), Singapore, and Team Leader at RIKEN Genomic Sciences Center (2002–2006), Japan. Other prior positions included Principal Investigator at Kent Ridge Digital Labs, Singapore and Research Scientist at Chugai Institute for Molecular Medicine, Inc., Japan. In 2018 he became a member of International Society for Computational Biology (ISCB) Board of Directors. Since 2010 he is serving Asia-Pacific Bioinformatics Network (APBioNet) as Vice-President (Conferences 2010–2016) and President (2016–2018).

VOLUME EDITORS



Mario Cannataro is a Full Professor of Computer Engineering and Bioinformatics at University “Magna Graecia” of Catanzaro, Italy. He is the director of the Data Analytics research center and the chair of the Bioinformatics Laboratory at University “Magna Graecia” of Catanzaro. His current research interests include bioinformatics, medical informatics, data analytics, parallel and distributed computing. He is a Member of the editorial boards of Briefings in Bioinformatics, High-Throughput, Encyclopedia of Bioinformatics and Computational Biology, Encyclopedia of Systems Biology. He was guest editor of several special issues on bioinformatics and he is serving as a program committee member of several conferences. He published three books and more than 200 papers in international journals and conference proceedings. Prof. Cannataro is a Senior Member of IEEE, ACM and BITS, and a member of the Board of Directors for ACM SIGBIO.



Bruno Gaeta is Senior Lecturer and Director of Studies in Bioinformatics in the School of Computer Science and Engineering at UNSW Australia. His research interests cover multiple areas of bioinformatics including gene regulation and protein structure, currently with a focus on the immune system, antibody genes and the generation of antibody diversity. He is a pioneer of bioinformatics education and has trained thousands of biologists and trainee bioinformaticians in the use of computational tools for biological research through courses, workshops as well as a book series. He has worked both in academia and in the bioinformatics industry, and currently coordinates the largest bioinformatics undergraduate program in Australia.



Mohammad Asif Khan, PhD, is an associate professor and the Dean of the School of Data Sciences, as well as the Director of the Centre for Bioinformatics at Perdana University, Malaysia. He is also a visiting scientist at the Department of Pharmacology and Molecular Sciences, Johns Hopkins University School of Medicine (JHUSOM), USA. His research interests are in the area of biological data warehousing and applications of bioinformatics to the study of immune responses, vaccines, inhibitory drugs, venom toxins, and disease biomarkers. He has published in these areas, been involved in the development of several novel bioinformatics methodologies, tools, and specialized databases, and currently has three patent applications granted. He has also led the curriculum development of a Postgraduate Diploma in Bioinformatics programme and an MSc (Bioinformatics) programme at Perdana University. He is an elected ExCo member of the Asia-Pacific Bioinformatics Network (APBioNET) since 2010 and is currently the President of Association for Medical and Bio-Informatics, Singapore (AMBIS). He has donned various important roles in the organization of many local and international bioinformatics conferences, meetings and workshops.

CONTENTS OF VOLUME 2

Editors in Chief	v
Volume Editors	vii
List of Contributors for Volume 2	xv
Preface	xxiii

VOLUME 2

The Gene Ontology <i>Pascale Gaudet</i>	1
Protein Functional Annotation <i>Pier Luigi Martelli, Giuseppe Profiti, and Rita Casadio</i>	8
Protein Post-Translational Modification Prediction <i>Chi NI Pang and Marc R Wilkins</i>	15
Protein Properties <i>Kentaro Tomii</i>	28
The Evolution of Protein Family Databases <i>Teresa K Attwood and Alex L Mitchell</i>	34
Transmembrane Domain Prediction <i>Pier Luigi Martelli, Castrense Savojardo, Giuseppe Profiti, and Rita Casadio</i>	46
Prediction of Protein Localization <i>Kenta Nakai and Kenichiro Imai</i>	53
Proteome Informatics <i>Frédérique Lisacek</i>	60
Proteomics Data Representation and Databases <i>Thibault Robin</i>	76
Proteomics Mass Spectrometry Data Analysis Tools <i>Aivett Bilbao</i>	84
Biological Databases <i>Sarinder K Dhillon</i>	96
Functional Genomics <i>Shalini Kaushik, Sandeep Kaushik, and Deepak Sharma</i>	118
Transcription Factor Databases <i>Edgar Wingender, Alexander Kel, and Mathias Krull</i>	134
Protein-DNA Interactions <i>Preeti Pandey, Sabeeha Hasnain, and Shandar Ahmad</i>	142
Gene Regulatory Network Review <i>Enze Liu, Lang Li, and Lijun Cheng</i>	155
MicroRNA and lncRNA Databases and Analysis <i>Xiangying Sun and Michael Gribskov</i>	165

Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine <i>Molly A Hall, Brian S Cole, and Jason H Moore</i>	171
Genome Informatics <i>Anil K Kesarwani, Ankit Malhotra, Anuj Srivastava, Guruprasad Ananda, Haitham Ashoor, Parveen Kumar, Rupesh K Kesharwani, Vishal K Sarsani, Yi Li, Joshy George, and R Krishna Murty Karuturi</i>	178
Genome Annotation <i>Josep F Abril and Sergi Castellano</i>	195
Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses? <i>Emmanuelle Lerat</i>	210
Bioinformatics Approaches for Studying Alternative Splicing <i>Prasoon K Thakur, Hukam C Rawal, Mina Obuca, and Sandeep Kaushik</i>	221
Genome-Wide Association Studies <i>William S Bush</i>	235
Gene Mapping <i>Kelly L Williams</i>	242
Genome Databases and Browsers <i>Dean Southwood and Shoba Ranganathan</i>	251
Comparative and Evolutionary Genomics <i>Takeshi Kawashima</i>	257
Genome Alignment <i>Tetsushi Yada</i>	268
Phylogenetic Footprinting <i>HiroYuki Toh</i>	284
Chromatin: A Semi-Structured Polymer <i>Wayne K Dawson, Michal Lazniewski, and Dariusz Plewczynski</i>	288
Nucleosome Positioning <i>Vladimir B Teif and Christopher T Clarkson</i>	308
Learning Chromatin Interaction Using Hi-C Datasets <i>Wing-Kin Sung</i>	318
Transcriptome Informatics <i>Liang Chen and Garry Wong</i>	324
Transcriptomic Databases <i>Askhat Molkenov, Altyn Zhelambayeva, Akbar Yermekov, Saule Mussurova, Aliya Sarkytbayeva, Yerbol Kalykhbergenov, Zhaxybay Zhumadilov, and Ulykbek Kairov</i>	341
Next Generation Sequence Analysis <i>Christian Rockmann, Christoph Endrullat, Marcus Frohme, and Heike Pospisil</i>	352
Transcriptomic Data Normalization <i>Fatemeh Vafaei, Hamed Dashti, and Hamid Alinejad-Rokny</i>	364
Differential Expression From Microarray and RNA-seq Experiments <i>Marc Delord</i>	372
Expression Clustering <i>Xiaoxin Ye and Joshua WK Ho</i>	388
Metabolome Analysis <i>Saravanan Dayalan, Jianguo Xia, Rachel A Spicer, Reza Salek, and Ute Roessner</i>	396

Mass Spectrometry-Based Metabolomic Analysis <i>Russell Pickford</i>	410
Metabolic Profiling <i>Joram M Posma</i>	426
Metabolic Models <i>Jean-Marc Schwartz and Zita Soons</i>	438
Protein Structural Bioinformatics: An Overview <i>M Michael Gromiha, Raju Nagarajan, and Samuel Selvaraj</i>	445
Protein Structure Databases <i>David R Armstrong, John M Berrisford, Matthew J Conroy, Alice R Clark, Deepti Gupta, and Abhik Mukhopadhyay</i>	460
Protein Structure Classification <i>Natalie L Dawson, Sayoni Das, Jonathan G Lees, and Christine Orengo</i>	472
Secondary Structure Prediction <i>Jonas Reeb and Burkhard Rost</i>	488
Protein Three-Dimensional Structure Prediction <i>Sanne Abeln, Klaas Anton Feenstra, and Jaap Heringa</i>	497
Protein Structure Analysis and Validation <i>Tsuyoshi Shirai</i>	512
Protein Structure Visualization <i>Sandeep Kaushik and Soumya Lipsa Rath</i>	520
Computational Tools for Structural Analysis of Proteins <i>Luciano A Abriata</i>	539
Molecular Dynamics and Simulation <i>Liangzhen Zheng, Amr A Alhossary, Chee-Keong Kwoh, and Yuguang Mu</i>	550
Nucleic-Acid Structure Database <i>Airy Sanjeev, Venkata SK Mattaparthi, and Sandeep Kaushik</i>	567
RNA Structure Prediction <i>Junichi Iwakiri and Kiyoshi Asai</i>	575
Rational Structure-Based Drug Design <i>Varun Khanna, Shoba Ranganathan, and Nikolai Petrovsky</i>	585
Chemoinformatics: From Chemical Art to Chemistry in Silico <i>Jaroslav Polanski</i>	601
Fingerprints and Pharmacophores <i>Daniela Schuster</i>	619
Public Chemical Databases <i>Sunghwan Kim</i>	628
Chemical Similarity and Substructure Searches <i>Nils M Kriege, Lina Humbeck, and Oliver Koch</i>	640
Binding Site Comparison – Software and Applications <i>Christiane Ehrhart, Tobias Brinkjost, and Oliver Koch</i>	650
Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications <i>Swathik Clarancia Peter, Jaspreet Kaur Dhanja, Vidhi Malik, Navaneethan Radhakrishnan, Mannu Jayakanthan, and Durai Sundar</i>	661
Pharmacophore Development <i>Balakumar Chandrasekaran, Nikhil Agrawal, and Sandeep Kaushik</i>	677

Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer <i>Muhammad Yusuf, Ari Hardianto, Muchtaridi Muchtaridi, Rina F Nuwarda, and Toto Subroto</i>	688
Molecular Phylogenetics <i>Michael A Charleston</i>	700
Evolutionary Models <i>David A Liberles and Barbara R Holland</i>	712
Molecular Clock <i>Cara Van Der Wal and Simon YW Ho</i>	719
Phylogenetic Tree Rooting <i>Richard J Edwards</i>	727
Tree Evaluation and Robustness Testing <i>Mahendra Mariadassou, Avner Bar-Hen, and Hirohisa Kishino</i>	736
Applications of the Coalescent for the Evolutionary Analysis of Genetic Data <i>Miguel Arenas</i>	746
Population Genetics <i>Conrad J Burden</i>	759
Computational Systems Biology <i>Sucheendra K Palaniappan, Ayako Yachie-Kinoshita, and Samik Ghosh</i>	789
Pathway Informatics <i>Sarita Poonia, Smriti Chawla, Sandeep Kaushik, and Debarka Sengupta</i>	796
Network Inference and Reconstruction in Bioinformatics <i>Paolo Tieri, Lorenzo Farina, Manuela Petti, Laura Astolfi, Paola Paci, and Filippo Castiglione</i>	805
Graphlets and Motifs in Biological Networks <i>Laurentino Quiroga Moreno</i>	814
Protein-Protein Interactions: An Overview <i>Edson L Folador, Sandeep Tiwari, Camila E Da Paz Barbosa, Syed B Jamal, Marco Da Costa Schulze, Debmalya Barh, and Vasco Azevedo</i>	821
Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope <i>Anna Laddach, Sun Sook Chung, and Franca Fraternali</i>	834
Protein-Protein Interaction Databases <i>Ashwini Patil</i>	849
Computational Approaches for Modeling Signal Transduction Networks <i>Zhongming Zhao and Junfeng Xia</i>	856
Cell Modeling and Simulation <i>Ayako Yachie-Kinoshita and Kazunari Kaizu</i>	864
Quantitative Modelling Approaches <i>Filippo Castiglione, Emiliano Mancini, Marco Pedicini, and Abdul Salam Jarrah</i>	874
Data Formats for Systems Biology and Quantitative Modeling <i>Martin Golebiewski</i>	884
Computational Lipidomics <i>Josch K Pauling</i>	894
Multi-Scale Modelling in Biology <i>Socrates Dokos</i>	900

Computational Immunogenetics <i>Marta Gómez Perosanz, Giulia Russo, Jose Luis Sanchez-Trincado Lopez, Marzio Pennisi, Pedro A Reche, Adrian Shepherd, and Francesco Pappalardo</i>	906
Immunoinformatics Databases <i>Christine LP Eng, Tin Wee Tan, and Joo Chuan Tong</i>	931
Study of Human Antibody Responses From Analysis of Immunoglobulin Gene Sequences <i>Katherine JL Jackson</i>	940
Epitope Predictions <i>Roman Kogay and Christian Schönbach</i>	952
Immunoglobulin Clonotype and Ontogeny Inference <i>Pazit Polak, Ramit Mehr, and Gur Yaari</i>	972
Quantitative Immunology by Data Analysis Using Mathematical Models <i>Shoya Iwanami and Shingo Iwami</i>	984
Bioimage Informatics <i>Sorayya Malek, Mogeheb Mosleh, Sarinder K Dhillon, and Pozi Milow</i>	993
Bioimage Databases <i>Arpah Abu and Sarinder K Dhillon</i>	1011
Segmentation Techniques for Bioimages <i>Saowaluck Kaewkamnerd, Apichart Intarapanich, and Sissades Tongsim</i>	1028
Translational and Disease Bioinformatics <i>Sandeep Kaushik, Chakrawarti Prasun, and Deepak Sharma</i>	1046
Translational Bioinformatics Databases <i>Onkar Singh, Nai-Wen Chang, Hong-Jie Dai, and Jitendra Jonnagaddala</i>	1058
Electronic Health Record Integration <i>Hong Yung Yip, Nur A Taib, Haris A Khan, and Sarinder K Dhillon</i>	1063
Genome Databases and Browsers for Cancer <i>Madhu Goyal</i>	1077
Text Mining Resources for Bioinformatics <i>Sandeep Kaushik, Priyanka Baloni, and Charu K Midha</i>	1083
Preclinical: Drug Target Identification and Validation in Human <i>Meena K Sakharkar, Karthic Rajamanickam, Chidambaram S Babu, Jitender Madan, Ramesh Chandra, and Jian Yang</i>	1093
Biomedical Text Mining <i>Hagit Shatkay</i>	1099
Biodiversity Databases and Tools <i>Lina AM Zalani, Kho S Jye, Shaarmini Balakrishnan, and Sarinder K Dhillon</i>	1110
Challenges in Creating Online Biodiversity Repositories With Taxonomic Classification <i>Mohd NM Ali, Amy Y Then Hui, and Sarinder K Dhillon</i>	1124
Ecological Networks <i>Kazuhiro Takemoto and Midori Iida</i>	1131
Dedicated Bioinformatics Analysis Hardware <i>Bertil Schmidt and Andreas Hildebrandt</i>	1142
Computational Pipelines and Workflows in Bioinformatics <i>Jeremy Leipzig</i>	1151

LIST OF CONTRIBUTORS FOR VOLUME 2

Sanne Abeln

*Center for Integrative Bioinformatics VU (IBIVU),
The Netherlands*

Luciano A Abriata

*Swiss Federal Institute of Technology in Lausanne,
Lausanne, Switzerland; and Swiss Institute of
Bioinformatics, Lausanne, Switzerland*

Josep F. Abril

*University of Barcelona (UB), Barcelona, Spain;
and Institute of Biomedicine of the University of
Barcelona (IBUB), Barcelona, Spain*

Arpah Abu

University of Malaya, Kuala Lumpur, Malaysia

Nikhil Agrawal

University of KwaZulu-Natal, Durban, South Africa

Shandar Ahmad

Jawaharlal Nehru University, New Delhi, India

Amr A. Alhossary

Nanyang Technological University, Singapore

Mohd N.M. Ali

University of Malaya, Kuala Lumpur, Malaysia

Hamid Alinejad-Rokny

University of New South Wales, Sydney, NSW, Australia

Guruprasad Ananda

*The Jackson Laboratory, Farmington, CT,
United States*

Miguel Arenas

University of Vigo, Vigo, Spain

David R. Armstrong

*Protein Data Bank in Europe, EMBL-EBI, Hinxton,
United Kingdom*

Kiyoshi Asai

*University of Tokyo, Kashiwa, Japan; and Artificial
Intelligence, Research Center, Koto-ku, Japan*

Haitham Ashoor

*The Jackson Laboratory, Farmington, CT,
United States*

Laura Astolfi

*Department of Computer, Control and Management
Engineering "A. Ruberti", Sapienza University of Rome,
Italy; and Neuroelectrical Imaging and BCI Lab,
Fondazione Santa Lucia IRCSS, Rome, Italy*

Teresa K. Attwood

*The University of Manchester, Manchester, United
Kingdom*

Vasco Azevedo

*Federal University of Minas Gerais, Belo Horizonte,
Brazil*

Chidambaram S. Babu

JSS University, Mysuru, India

Shaarmini Balakrishnan

*Manipal International University, Nilai, Negeri
Sembilan, Malaysia*

Priyanka Baloni

Institute for Systems Biology, Seattle, WA, United States

Debmalya Barh

*Institute of Integrative Omics and Applied Biotechnology,
Purba Medinipur, India*

Avner Bar-Hen

CNAM, Paris, France

John M. Berrisford

*Protein Data Bank in Europe, EMBL-EBI, Hinxton,
United Kingdom*

Aivett Bilbao

*Pacific Northwest National Laboratory, Richland, WA,
United States*

Tobias Brinkjost

TU Dortmund University, Dortmund, Germany

Conrad J. Burden

*Australian National University, Canberra, ACT,
Australia*

William S. Bush

*Case Western Reserve University, Cleveland, OH,
United States*

Rita Casadio

University of Bologna, Bologna, Italy

Sergi Castellano

*Great Ormond Street Institute of Child Health (ICH),
University College London (UCL), London, United
Kingdom; and UCL Genomics, University College
London (UCL), London, United Kingdom*

Filippo Castiglione

National Research Council of Italy, Rome, Italy

- Ramesh Chandra
University of Delhi, Delhi, India
- Balakumar Chandrasekaran
University of KwaZulu-Natal, Durban, South Africa
- Nai-Wen Chang
National Taiwan University, Taipei city, Taiwan, Republic of China; and National Yang-Ming University, Taipei, Taiwan, Republic of China
- Michael A. Charleston
University of Tasmania, Hobart, TAS, Australia
- Smriti Chawla
Indraprastha Institute of Information Technology, Delhi, India
- Liang Chen
University of Macau, Macau, China
- Lijun Cheng
The Ohio State University, Columbus, OH, United States
- Sun Sook Chung
King's College, London, UK
- Alice R. Clark
Protein Data Bank in Europe, EMBL-EBI, Hinxton, United Kingdom
- Christopher T. Clarkson
University of Essex, Colchester, United Kingdom
- Brian S. Cole
University of Pennsylvania, Philadelphia, PA, United States
- Matthew J. Conroy
Protein Data Bank in Europe, EMBL-EBI, Hinxton, United Kingdom
- Marco Da Costa Schulze
Federal University of Paraíba, João Pessoa, Brazil
- Camila E. Da Paz Barbosa
Federal University of Paraíba, João Pessoa, Brazil
- Hong-Jie Dai
National Taitung University, Taipei, Taiwan, Republic of China
- Sayoni Das
Institute of Structural and Molecular Biology, University College London, London, United Kingdom
- Hamed Dashti
Sharif University of Technology, Tehran, Iran
- Wayne K. Dawson
CeNT, University of Warsaw, Warsaw, Poland; and BiLab, The University of Tokyo, Tokyo, Japan
- Natalie L. Dawson
Institute of Structural and Molecular Biology, University College London, London, United Kingdom
- Saravanan Dayalan
The University of Melbourne, Parkville, VIC, Australia
- Marc Delord
Université Paris Diderot-Paris 7, Sorbone Paris Cité, Paris, France
- Jaspreet Kaur Dhanja
Indian Institute of Technology Delhi, New Delhi, India
- Sarinder K. Dhillon
University of Malaya, Kuala Lumpur, Malaysia
- Socrates Dokos
University of New South Wales, Sydney, NSW, Australia
- Richard J. Edwards
University of New South Wales, Sydney, NSW, Australia
- Christiane Eehrt
TU Dortmund University, Dortmund, Germany
- Christoph Endrullat
University of Applied Sciences, Wildau, Germany
- Christine L.P. Eng
National University of Singapore, Singapore
- Kiyoshi Ezawa
Kyushu Institute of Technology, Fukuoka, Japan
- Lorenzo Farina
Department of Computer, Control and Management Engineering "A. Ruberti", Sapienza University of Rome, Italy; and Neuroelectrical Imaging and BCI Lab, Fondazione Santa Lucia IRCSS, Rome, Italy
- Klaas Anton Feenstra
Center for Integrative Bioinformatics VU (IBIVU), The Netherlands
- Edson L. Folador
Federal University of Paraíba, João Pessoa, Brazil
- Franca Fraternali
King's College, London, UK
- Marcus Frohme
University of Applied Sciences, Wildau, Germany
- Marta Gómez Perosanz
Complutense University of Madrid, Madrid, Spain
- Pascale Gaudet
SIB Swiss Institute of Bioinformatics, Geneva, Switzerland

Joshy George
The Jackson Laboratory, Farmington, CT, United States

Samik Ghosh
The Systems Biology Institute, Tokyo, Japan

Martin Golebiewski
Heidelberg Institute for Theoretical Studies (HITS gGmbH), Heidelberg, Germany

Madhu Goyal
University of Technology Sydney, Sydney, NSW, Australia

Michael Gribskov
Purdue University, West Lafayette, IN, United States

M. Michael Gromiha
Tokyo Institute of Technology, Tokyo, Japan

Deepti Gupta
Protein Data Bank in Europe, EMBL-EBI, Hinxton, United Kingdom

Molly A. Hall
The Pennsylvania State University, University Park, PA, United States

Ari Hardianto
Universitas Padjadjaran, Jatinangor, Indonesia

Sabeeha Hasnain
Jawaharlal Nehru University, New Delhi, India

Jaap Heringa
Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Andreas Hildebrandt
Institut für Informatik, Mainz, Germany

Joshua W.K. Ho
Victor Chang Cardiac Research Institute, Darlinghurst, NSW, Australia; and University of New South Wales, Sydney, NSW, Australia

Simon Y.W. Ho
University of Sydney, Sydney NSW, Australia

Barbara R. Holland
University of Tasmania, Hobart, TAS, Australia

Amy Y. Then Hui
University of Malaya, Kuala Lumpur, Malaysia

Lina Humbeck
TU, Dortmund, Germany

Midori Iida
Kyushu Institute of Technology, Fukuoka, Japan

Kenichiro Imai
Advanced Industrial Science and Technology, Tokyo, Japan

Apichart Intarapanich
National Electronics and Computer Technology Center, Pathum Thani, Thailand

Junichi Iwakiri
University of Tokyo, Kashiwa, Japan

Shingo Iwami
Kyushu University, Fukuoka, Japan; and Japan Science and Technology Agency, Kawaguchi, Japan

Shoya Iwanami
Kyushu University, Fukuoka, Japan

Katherine J.L. Jackson
Garvan Institute of Medical Research, Darlinghurst, NSW, Australia

Syed B. Jamal
Federal University of Minas Gerais, Belo Horizonte, Brazil

Abdul Salam Jarrah
American University of Sharjah, Sharjah, UAE

Mannu Jayakanthan
Tamil Nadu Agricultural University, Coimbatore, India

Jitendra Jonnagaddala
University of New South Wales, Sydney, NSW, Australia

Kho S. Jye
University of Malaya, Kuala Lumpur, Malaysia

Saowaluck Kaewkamnerd
National Electronics and Computer Technology Center, Pathum Thani, Thailand

Ulykbek Kairov
Nazarbayev University, Astana, Kazakhstan

Kazunari Kaizu
RIKEN Quantitative Biology Center (QBiC), Osaka, Japan

Yerbol Kalykhbergenov
Nazarbayev University, Astana, Kazakhstan

R. Krishna Murty Karuturi
The Jackson Laboratory, Farmington, CT, United States

Shalini Kaushik
Indian Institute of Technology Roorkee, India

Sandeep Kaushik
European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal; and University of Minho, Braga, Portugal

Takeshi Kawashima
National Institute of Genetics, Mishima, Japan

Alexander Kel
geneXplain GmbH, Wolfenbüttel, Germany

Anil K. Kesarwani
The Jackson Laboratory, Farmington, CT, United States

Rupesh K. Kesharwani
The Jackson Laboratory, Farmington, CT, United States

Haris A. Khan
University of Malaya, Kuala Lumpur, Malaysia

Varun Khanna
Flinders University, Adelaide, SA, Australia; and Vaxine Pty Ltd., Adelaide, SA, Australia

Sunghwan Kim
National Institutes of Health, Bethesda, MD, United States

Hirohisa Kishino
University of Tokyo, Tokyo, Japan

Oliver Koch
Technical University of, Dortmund, Germany

Roman Kogay
Nazarbayev University, Astana, Republic of Kazakhstan

Nils M. Kriege
TU, Dortmund, Germany

Mathias Krull
geneXplain GmbH, Wolfenbüttel, Germany

Parveen Kumar
The Jackson Laboratory, Farmington, CT, United States

Subhas C Kundu
Headquarters of the European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal

Chee-Keong Kwoh
Nanyang Technological University, Singapore

Anna Laddach
King's College, London, UK

Michal Lazniewski
CeNT, University of Warsaw, Warsaw, Poland; and Medical University of Warsaw, Warsaw, Poland

Jonathan G. Lees
Institute of Structural and Molecular Biology, University College London, London, United Kingdom

Jeremy Leipzig
Drexel University, Philadelphia, PA, United States

Emmanuelle Lerat
Université Lyon, CNRS, Villeurbanne, France

Lang Li
The Ohio State University, Columbus, OH, United States

Yi Li
The Jackson Laboratory, Farmington, CT, United States

David A. Liberles
Temple University, Philadelphia, PA, United States

Frédérique Lisacek
SIB Swiss Institute of Bioinformatics, Geneva, Switzerland; and University of Geneva, Geneva, Switzerland

Enze Liu
Indiana University School of Medicine, Indianapolis, IN, United States; and Indiana University, Indianapolis, IN, United States

Pier Luigi Martelli
University of Bologna, Bologna, Italy

Jitender Madan
Chandigarh College of Pharmacy Landran, Mohali, India

Sorayya Malek
University of Malaya, Kuala Lumpur, Malaysia

Ankit Malhotra
The Jackson Laboratory, Farmington, CT, United States

Vidhi Malik
Indian Institute of Technology Delhi, New Delhi, India

Emiliano Mancini
University of Amsterdam, Amsterdam, Netherlands

Mahendra Mariadassou
INRA, Paris, France

Venkata S.K. Mattaparthi
Tezpur University, Tezpur, India

Ramit Mehr
Bar-Ilan University, Ramt Gan, Israel

Charu K. Midha
Institute for Systems Biology, Seattle, WA, United States

Pozi Milow
University of Malaya, Kuala Lumpur, Malaysia

Alex L. Mitchell
EMBL-EBI, Hinxton, United Kingdom

Askhat Molkenov
Nazarbayev University, Astana, Kazakhstan

Jason H. Moore
University of Pennsylvania, Philadelphia, PA, United States

Laurentino Quiroga Moreno
Birkbeck College, University of London, United Kingdom

Mogeeb Mosleh
University of Taiz, Taiz, Yemen

Yuguang Mu
Nanyang Technological University, Singapore

Muchtaridi Muchtaridi
Universitas Padjadjaran, Jatinangor, Indonesia

Abhik Mukhopadhyay
Protein Data Bank in Europe, EMBL-EBI, Hinxton, United Kingdom

Saule Mussurova
Nazarbayev University, Astana, Kazakhstan

Raju Nagarajan
Indian Institute of Technology Madras, Chennai, India; and Vanderbilt University Medical Center, Nashville, TN, United States

Kenta Nakai
The University of Tokyo, Tokyo, Japan

Rina F. Nuwarda
Universitas Padjadjaran, Jatinangor, Indonesia

Mina Obuca
Institute of Molecular Genetics of the ASCR, Prague, Czech Republic

Christine Orengo
Institute of Structural and Molecular Biology, University College London, London, United Kingdom

Paola Paci
CNR National Research Council, IASI Institute of Systems Analysis and Computer Science "Antonio Ruberti", Rome, Italy

Sucheendra K. Palaniappan
The Systems Biology Institute, Tokyo, Japan

Preeti Pandey
Jawaharlal Nehru University, New Delhi, India

Chi N.I. Pang
The University of New South Wales, Sydney, NSW, Australia

Francesco Pappalardo
University of Catania, Catania, Italy

Ashwini Patil
The University of Tokyo, Tokyo, Japan

Josch K. Pauling
Humboldt University of Berlin, Berlin, Germany

Marco Pedicini
Roma Tre University, Rome, Italy

Marzio Pennisi
University of Catania, Catania, Italy

Swathik Clarancia Peter
Tamil Nadu Agricultural University, Coimbatore, India

Nikolai Petrovsky
Flinders University, Adelaide, SA, Australia; and Vaxine Pty Ltd., Adelaide, SA, Australia

Manuela Petti
Department of Computer, Control and Management Engineering "A. Ruberti", Sapienza University of Rome, Italy; and Neuroelectrical Imaging and BCI Lab, Fondazione Santa Lucia IRCSS, Rome, Italy

Russell Pickford
University of New South Wales, Kensington, NSW, Australia

Dariusz Plewczynski
CeNT, University of Warsaw, Warsaw, Poland; and MINI, Warsaw University of Technology, Warsaw, Poland

Pazit Polak
Bar-Ilan University, Ramt Gan, Israel

Jaroslav Polanski
University of Silesia Institute of Chemistry, Katowice, Poland

Sarita Poonia
Indraprastha Institute of Information Technology, Delhi, India

Joram M. Posma
Imperial College London, London, United Kingdom

Heike Pospisil
University of Applied Sciences, Wildau, Germany

Chakrawarti Prasun
Indian Institute of Technology Roorkee, India

Giuseppe Profiti
University of Bologna, Bologna, Italy

Navaneethan Radhakrishnan
Indian Institute of Technology Delhi, New Delhi, India

Karthic Rajamanickam
University of Saskatchewan, Saskatoon, SK, Canada

Shoba Ranganathan
Macquarie University, Sydney, NSW, Australia

Soumya Lipsa Rath
Nagoya University, Nagoya, Japan

Hukam C. Rawal
ICAR – National Research Centre on Plant Biotechnology, New Delhi, India

Pedro A. Reche
Complutense University of Madrid, Madrid, Spain

Jonas Reeb
*Technical University of Munich (TUM), Garching/
Munich, Germany*

Rui L Reis
*Headquarters of the European Institute of Excellence on
Tissue Engineering and Regenerative Medicine,
Guimaraes, Portugal*

Thibault Robin
University of Geneva, CUI, Carouge, Switzerland

Christian Rockmann
University of Applied Sciences, Wildau, Germany

Ute Roessner
The University of Melbourne, Parkville, VIC, Australia

Burkhard Rost
*Technical University of Munich (TUM), Garching/
Munich, Germany*

Giulia Russo
University of Catania, Catania, Italy

Meena K. Sakharkar
University of Saskatchewan, Saskatoon, SK, Canada

Reza Salek
*European Bioinformatics Institute (EMBL-EBI),
Hinxton, United Kingdom*

Jose Luis Sanchez-Trincado Lopez
Complutense University of Madrid, Madrid, Spain

Airy Sanjeev
Tezpur University, Tezpur, Assam, India

Aliya Sarkytbayeva
Nazarbayev University, Astana, Kazakhstan

Vishal K. Sarsani
The Jackson Laboratory, Farmington, CT, United States

Castrense Savojardo
University of Bologna, Bologna, Italy

Christian Schönbach
Nazarbayev University, Astana, Kazakhstan

Bertil Schmidt
Institut für Informatik, Mainz, Germany

Daniela Schuster
*Paracelsus Medical University of Salzburg, Salzburg,
Austria*

Jean-Marc Schwartz
University of Manchester, Manchester, United Kingdom

Samuel Selvaraj
Bharathidasan University, Tiruchirappalli, India

Debarka Sengupta
*Indraprastha Institute of Information Technology, Delhi,
India*

Deepak Sharma
Indian Institute of Technology Roorkee, India

Hagit Shatkay
University of Delaware, Newark, DE, United States

Adrian Shepherd
University of London, London, United Kingdom

Tsuyoshi Shirai
*Nagahama Institute of Bio-Science and Technology,
Nagahama, Japan*

Onkar Singh
*Institute of Information Science, Academia Sinica,
Taipei, Taiwan, Republic of China; and National Yang-
Ming University, Taipei, Taiwan, Republic of China*

Zita Soons
Maastricht University, Maastricht, The Netherlands

Dean Southwood
Macquarie University, Sydney, NSW, Australia

Rachel A. Spicer
*European Bioinformatics Institute (EMBL-EBI),
Hinxton, United Kingdom*

Anuj Srivastava
The Jackson Laboratory, Farmington, CT, United States

Toto Subroto
Universitas Padjadjaran, Bandung, West Java, Indonesia

Xiangying Sun
Purdue University, West Lafayette, IN, United States

Durai Sundar
Indian Institute of Technology Delhi, New Delhi, India

Wing-Kin Sung
*National University of Singapore, Singapore; and
Genome Institute of Singapore, Singapore*

Nur A. Taib
University of Malaya, Kuala Lumpur, Malaysia

Kazuhiro Takemoto
Kyushu Institute of Technology, Fukuoka, Japan

Tin Wee Tan
National University of Singapore, Singapore

Vladimir B. Teif
University of Essex, Colchester, United Kingdom

Prasoon K. Thakur
*Institute of Molecular Genetics of the ASCR, Prague,
Czech Republic*

Paolo Tieri
*CNR National Research Council, IAC Institute for
Applied Computing "Mauro Picone", Rome, Italy*

Sandeep Tiwari
*Federal University of Minas Gerais, Belo Horizonte,
Brazil*

Hiroyuki Toh
Kwansei Gakuin University, Sanda, Japan

Kentaro Tomii
*Artificial Intelligent Research Center (AIRC), National
Institute of Advanced Industrial Science and Technology
(AIST), Tokyo, Japan*

Joo Chuan Tong
*National University of Singapore, Singapore; and
Institute of High Performance Computing, Singapore*

Sissades Tongsima
*National Center for Genetic Engineering and
Biotechnology, Pathum Thani, Thailand*

Fatemeh Vafaei
*University of New South Wales, Sydney, NSW,
Australia*

Cara Van Der Wal
*University of Sydney, Sydney, NSW, Australia; and
Australian Museum Research Institute, Australian
Museum, Sydney, NSW, Australia*

Marc R. Wilkins
*The University of New South Wales, Sydney, NSW,
Australia*

Kelly L. Williams
Macquarie University, Sydney, NSW, Australia

Edgar Wingender
*geneXplain GmbH, Wolfenbüttel, Germany; and
University Medical Center Göttingen, Göttingen,
Germany*

Garry Wong
University of Macau, Macau, China

Jianguo Xia
*McGill University, Sainte Anne de Bellevue, QC,
Canada; and McGill University, Montreal, QC, Canada*

Junfeng Xia
Anhui University, Hefei, China

Gur Yaari
Bar-Ilan University, Ramt Gan, Israel

Ayako Yachie-Kinoshita
*The Systems Biology Institute, Tokyo, Japan; and
University of Toronto, Toronto, ON, Canada*

Tetsushi Yada
Kyushu Institute of Technology, Fukuoka, Japan

Jian Yang
University of Saskatchewan, Saskatoon, SK, Canada

Xiaoxin Ye
*Victor Chang Cardiac Research Institute, Darlinghurst,
NSW, Australia*

Akbar Yermekov
Nazarbayev University, Astana, Kazakhstan

Hong Yung Yip
University of Malaya, Kuala Lumpur, Malaysia

Muhammad Yusuf
Universitas Padjadjaran, Jatinangor, Indonesia

Lina A.M. Zalani
University of Malaya, Kuala Lumpur, Malaysia

Zhongming Zhao
*The University of Texas Health Science Center at
Houston, Houston, TX, United States*

Altyn Zhelambayeva
Nazarbayev University, Astana, Kazakhstan

Liangzhen Zheng
Nanyang Technological University, Singapore

Zhaxybay Zhumadilov
Nazarbayev University, Astana, Kazakhstan

PREFACE

Bioinformatics and Computational Biology (BCB) combine elements of computer science, information technology, mathematics, statistics, and biotechnology, providing the methodology and *in silico* solutions to mine biological data and processes, for knowledge discovery. In the era of molecular diagnostics, targeted drug design and Big Data for personalized or even precision medicine, computational methods for data analysis are essential for biochemistry, biology, biotechnology, pharmacology, biomedical science, and mathematics and statistics. Bioinformatics and Computational Biology are essential for making sense of the molecular data from many modern high-throughput studies of mice and men, as well as key model organisms and pathogens. This Encyclopedia spans basics to cutting-edge methodologies, authored by leaders in the field, providing an invaluable resource to students as well as scientists, in academia and research institutes as well as biotechnology, biomedical and pharmaceutical industries.

Navigating the maze of confusing and often contradictory jargon combined with a plethora of software tools is often confusing for students and researchers alike. This comprehensive and unique resource provides up-to-date theory and application content to address molecular data analysis requirements, with precise definition of terminology, and lucid explanations by experts.

No single authoritative entity exists in this area, providing a comprehensive definition of the myriad of computer science, information technology, mathematics, statistics, and biotechnology terms used by scientists working in bioinformatics and computational biology. Current books available in this area as well as existing publications address parts of a problem or provide chapters on the topic, essentially addressing practicing bioinformaticists or computational biologists. Newcomers to this area depend on Google searches leading to published literature as well as several textbooks, to collect the relevant information.

Although curricula have been developed for Bioinformatics education for two decades now (Altman, 1998), offering education in bioinformatics continues to remain challenging from the multidisciplinary perspective, and is perhaps an NP-hard problem (Ranganathan, 2005). A minimum Bioinformatics skill set for university graduates has been suggested (Tan *et al.*, 2009). The Bioinformatics section of the Reference Module in Life Sciences (Ranganathan, 2017) commenced by addressing the paucity of a comprehensive reference book, leading to the development of this Encyclopedia. This compilation aims to fill the “gap” for readers with succinct and authoritative descriptions of current and cutting-edge bioinformatics areas, supplemented with the theoretical concepts underpinning these topics.

This Encyclopedia comprises three sections, covering Methods, Topics and Applications. The theoretical methodology underpinning BCB are described in the Methods section, with Topics covering traditional areas such as phylogeny, as well as more recent areas such as translational bioinformatics, cheminformatics and computational systems biology. Additionally, Applications will provide guidance for commonly asked “how to” questions on scientific areas described in the Topics section, using the methodology set out in the Methods section. Throughout this Encyclopedia, we have endeavored to keep the content as lucid as possible, making the text “... as simple as possible, but not simpler,” attributed to Albert Einstein. Comprehensive chapters provide overviews while details are provided by shorter, encyclopedic chapters.

During the planning phase of this Encyclopedia, the encouragement of Elsevier’s Priscilla Braglia and the constructive comments from no less than ten reviewers lead our small preliminary editorial team (Christian Schönbach, Kenta Nakai and myself) to embark on this massive project. We then welcomed one more Editor-in-Chief, Michael Gribskov and three section editors, Mario Cannataro, Bruno Gaeta and Asif Khan, whose toils have results in gathering most of the current content, with all editors reviewing the submissions. Throughout the production phase, we have received invaluable support and guidance as well as milestone reminders from Paula Davies, for which we remain extremely grateful.

Finally we would like to acknowledge all our authors, from around the world, who dedicated their valuable time to share their knowledge and expertise to provide educational guidance for our readers, as well as leave a lasting legacy of their work.

We hope the readers will enjoy this Encyclopedia as much as the editorial team have, in compiling this as an ABC of bioinformatics, suitable for naïve as well as experienced scientists and as an essential reference and invaluable teaching guide for students, post-doctoral scientists, senior scientists, academics in universities and research institutes as well as pharmaceutical, biomedical and biotechnological industries. Nobel laureate Walter Gilbert predicted in 1990 that “In the year 2020 you will be able to go into the drug store, have your DNA sequence read in an hour or so, and given back to you on a compact disk so you can analyze it.” While technology may have already arrived at this milestone, we are confident one of the readers of this Encyclopedia will be ready to extract valuable biological data by computational analysis, resulting in biomedical and therapeutic solutions, using bioinformatics to “measure” health for early diagnosis of “disease.”

References

- Altman, R.B., 1998. A curriculum for bioinformatics: the time is ripe. *Bioinformatics*. 14 (7), 549–550.
Ranganathan, S., 2005. Bioinformatics education—perspectives and challenges. *PLoS Comput Biol* 1 (6), e52.
Tan, T.W., Lim, S.J., Khan, A.M., Ranganathan, S., 2009. A proposed minimum skill set for university graduates to meet the informatics needs and challenges of the “-omics” era. *BMC Genomics*. 10 (Suppl 3), S36.
Ranganathan, S., 2017. *Bioinformatics. Reference Module in Life Sciences*. Oxford: Elsevier.

Shoba Ranganathan

The Gene Ontology

Pascale Gaudet, SIB Swiss Institute of Bioinformatics, Geneva, Switzerland

© 2019 Elsevier Inc. All rights reserved.

Background and Context

Up until the 1990s, data generated in biological and biomedical research were mostly shared via research articles. The development and democratisation of computers, on one hand, as well as the creation of network of this informatics infrastructure, on the other, opened a completely new way to share information and provided new possibilities to structure and integrate data that free text research articles cannot achieve. Concomitantly, high throughput technologies with applications for the biomedical sciences started to emerge, which allowed to analyze large numbers of biological samples (for example, microarrays) and to sequence ever larger and longer nucleic acid sequences (DNA and RNA). This gave rise to the new fields of proteomics, genomics, and transcriptomics. Repositories storing this data have been growing at exponential rates. One of the most important sequence repositories, GenBank, contains nucleotide sequences for 370,000 different species, with yearly rates of increase in certain data types such as transcriptomics of nearly 50% (Benson *et al.*, 2017). Biological research had entered a new era, in which data processing and analysis require informatics tools, leading to the emergence of the field of bioinformatics.

Gene function analysis in the genomics era. The Gene Ontology project was started at the beginning of the genomics era, when the rapidly increasing number of sequenced genomes made it necessary to have a way to describe gene products' functions that would be applicable to all realms of life. Up until that point, in each species the vast majority of the genes had not been identified nor studied. The relatively few genes that had been identified had some amount of knowledge associated with it (even if very small): Either a mutant and its associated phenotype, an enzymatic activity, a genetic interaction, etc. The availability of entire genome sequences, and moreover, the sequences of genomes of multiple species, increased by many orders of magnitude the number of different genes for which we at least knew the existence. However, there was no systematic way to describe gene functions, with different biology communities having developed their own, often organism-specific nomenclatures. For example, the term 'budding' has different meaning when describing plant propagation, tooth formation initiation, or asexual reproduction that occurs for example in *Saccharomyces cerevisiae*. The Gene Ontology was created to meet that need. The specific question that got the project started was the need, formulated by Prof Michael Ashburner, to classify all genes from *Drosophila*, the model system he was studying and whose genome had just been sequenced, and have the same classification scheme for other model organisms so that gene functions (and hence, genomes), could be compared (Lewis, 2017). The ability to compare homologous and orthologous genes allows formulating research hypotheses for uncharacterized genes, or when certain biological modules are well conserved, providing a model of a gene's function.

Why an Ontology?

In information sciences, an ontology is a computational structure that describe entities and relationships of a domain of interest in a structured, and hence computable format. Ontologies are composed of a set of entities, also called classes, arranged into a hierarchy from the general to the specific (Hastings, 2017). This structure is powerful because it is amenable to computational analysis using logical reasoning. Moreover, linking ontology classes with biological entities provides a structured classification of gene function which itself can be further used for analysis of experimental data.

The Gene Ontology

Organisation of GO. The overarching goal of the GO project is to provide life science researchers with current functional information concerning gene functions and the cellular context under which these occur. GO is a structured, controlled vocabulary that formally represents biological knowledge. In the GO system, the terms are subdivided in three non-overlapping ontologies (also known as aspects): Molecular Function (MF), which describe the molecular activities of gene products; Biological Process (BP), the pathways and larger processes made up of the activities of multiple gene products, and Cellular Component (CC), that illustrate where gene products are active (Ashburner *et al.*, 2000). These aspects are non-redundant and share a common space of identifiers and a well-specified syntax.

Classes and definitions. In GO, each term defines a *class*. The class itself is defined in two different ways: Textually, with a human-readable definition, and logically by relationships to other terms in the ontology and which can be computed over. The most common relation is *is_a*, which represents the ontological definition of a class (*i.e.*, it describes what the term *is* relative to others named classes in the ontology). Other commonly used relations in GO include *part_of*, representing partonomy between classes. For example, the brain is *part_of* the body.

Molecular functions are generally described as *types* of other functions, using the *is_a* relationship. For example, *DNA helicase activity* and *lyase activity* are both types of catalytic activity, and hence are related to *catalytic activity* by *is_a* relationships.

Biological process has three main types of relations, *is_a*, *part_of*, and *regulates*. The *is_a* relation represents types of a process, for example *mitotic DNA replication* (GO:1902969) is a particular type of *nuclear DNA replication* (GO:0033260). On the other hand, sub-processes are represented by *part_of* relations: *mitotic DNA replication* is a step of the *cell cycle* (GO:0000278). Biological regulation is a type of Biological Process, represented by the relation *regulates* and two more specific relations, *negatively regulates* and *positively regulates*.

Each class must be defined as a *is_a* class of another class, to ensure that the ontology is completely representing all classes logically. In addition, each class may have multiple relationships to other classes' terms. This allows for more expressivity than a simple hierarchy. This structure makes GO a directed graph with terms as nodes and relationships as edges. Fig. 1 graphically illustrates the structure of GO.

An increasingly larger number of terms are being defined with equivalence axioms or logical definitions (Gene Ontology Consortium, 2015). These use explicitly defined relations from the Relations Ontology to produce a description for terms that is amenable to logical reasoning. An example is shown in Fig. 2. The equivalence axiom makes the class unique relative to all other classes in the ontology.

As of November 2017, the full ontology contained 49,968 terms: 4146 cellular components, 29,669 biological processes and 11,140 molecular functions. The ontology also contains 73,776 explicitly encoded *is_a* relationships, 30,879 explicitly encoded *part_of* relationships, and 20,564 explicitly encoded *regulates*, *negatively regulates* or *positively regulates* relationships.

GO uses external ontologies extensively to represent other types of terms such as chemicals (ChEBI ontology (de Matos et al., 2010)), vertebrate anatomical parts (Uberon (Mungall et al., 2012)), plant parts (Cooper et al., 2013), cell types (Bard et al., 2005), and sequence features (Eilbeck et al., 2005).

Managing changes in the ontology. GO undergoes frequent revisions: changes of relations between terms, addition of new terms, or term removal. Terms are never deleted from the ontology, but their status changes to "obsolete" and all relationships to other terms are removed (Rhee et al., 2008). Term obsolescence may lead to loss of annotations if the responsible group does not re-house the annotations under a more appropriate term. Other types of changes such as changes to the relationships between terms and term merges do not impact annotations, because annotations always refer to specific terms, not their position within the GO. However, these changes can affect the analyses done using the ontology.

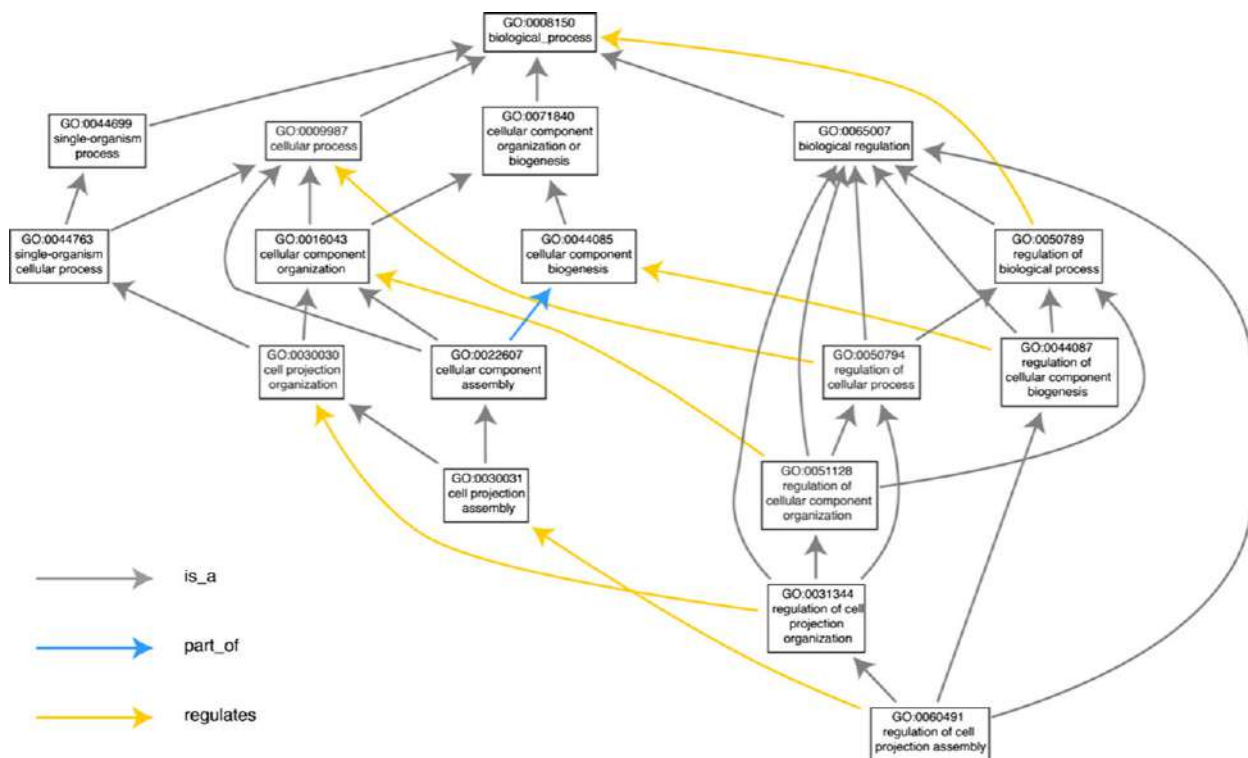


Fig. 1 Illustration of the Gene Ontology structure. Paths of the term *regulation of cell projection assembly*, GO:0060491 to its root term. GO is a directed graph with terms as nodes and relationships as edges; these relationships are either *is_a*, *part_of*, *has_part*, or *regulates*. In its basic representation, there should be no cycles in this graph, and we can therefore establish parent (more general) and child (more specific) terms. Reproduced from PMID:27812933.


```

name: L-glutamate import across plasma membrane
equivalentTo:
  transport that ('has target start location' some 'extracellular region')
  and ('has target end location' some cytosol)
  and (imports some 'L-glutamate')
  and ('results in transport across' some 'plasma membrane')
inferred classifications:
  'import across plasma membrane'
  'L-alpha-amino acid transmembrane transport'
  'L-glutamate import into cell'

```

Fig. 2 Logical definition for the class L-glutamate import across plasma membrane.

GO Annotations

Annotation objects. Ideally, gene ontology terms are associated to the biological entity having the function being captured, i.e. the gene product. In practice, database entries for *genes* are often used as a proxy for gene products. Gene products can be either a protein, a non-protein-coding RNA, or any other gene product, including isoforms and post-translationally-modified proteins, which can be captured using the Protein Ontology (Natale *et al.*, 2017). Moreover, GO allows capturing isoform-specific data when appropriate. Identifiers are for example UniProtKB isoform identifiers, such as P12345-2.

Semantics of a GO annotation. The association of a GO term and a gene product means that the gene product inherently possesses an activity (or, more precisely ontologically, the potential for that activity) or a molecular role (MF term), directly participates in a process (BP), and that its function occurs in a specific cellular localization (CC) (Ashburner *et al.*, 2000). Therefore, transient localizations such as endoplasmic reticulum and Golgi apparatus for secreted proteins are not in the scope of GO. Dynamic localizations such as the movement to the mitotic spindle during cytokinesis should be captured, as these correspond to locations at which the gene product is active. A thorough overview of the meaning of biological function in GO has been written by Thomas (2017).

A gene product can be annotated using as many GO terms as necessary to completely describe its function, and the GO terms can be at varying levels in the hierarchy, depending on the state of knowledge when the annotation was made. If a gene product is annotated to any particular term, then the annotation also holds for all the is-a and part-of parent terms. Annotations to more granular terms carry more information.

Certain annotations additionally contain *qualifiers*: *contributes_to*, *colocalizes_with* and *NOT*. The qualifier can profoundly change the meaning of an annotation, in particular the *NOT* qualifier, which means that a gene protein does not have a particular function, or does not participate in a given process, or is not localized to the a given cellular component. *Contributes_to* is used for molecular function annotations, and means that while the gene product has some role in the molecular function, on its own it does not possess the function. This usually applies to a protein complex: for example, each RNA polymerase subunit contributes to RNA polymerase activity, but none of the individual subunits has that function on its own. *Colocalizes_with* is used to capture a localization near an organelle, but when the gene product is not an intrinsic part of the organelle. Details about the qualifiers used in GO and their definitions can be found on the GO website, (see “Relevant Websites” section).

Data provenance. The supporting evidence for each annotation is captured using evidence codes and references. Evidence Codes represent the type of information from which the annotation is derived, and are grouped in five broad categories: experimental evidence codes, computational analysis evidence codes, author statement evidence codes, curatorial statement evidence codes, and automatically assigned evidence codes (Fig. 3). Experimental evidence codes represent data captured from experimental data. The data can be derived from a direct assay (*IDA*), from a mutant (*IMP* and *IGI*), from a physical interaction (*IPI*), or from the expression pattern of a gene product (*IEP*). The GOC has recently introduced new evidence codes to label data derived from high throughput experiments: *High throughput direct assay evidence* (*HDA*), *high throughput mutant phenotype evidence* (*HMP*), *high throughput genetic interaction evidence* (*HGI*), and *high throughput expression pattern evidence* (*HEP*). The largest fraction of annotations is attributed automatically, e.g., annotations via the InterPro resource (Burge *et al.*, 2012).

Manual annotation of Panther protein families (Mi *et al.*, 2017) based on the phylogenetic relationships between the members of these protein families (PMID:21873635) are assigned the evidence code *Inferred from Biological Aspect of Ancestor* (*IBA*). Curators also manually assign annotations derived from the known functions of similar sequences (*ISS* and related methods *ISA*, *ISM* and *ISO*). Curators can also capture annotations based on statements that can be traced to an original publication (*TAS*) and they can make statements based on well-known biological knowledge using the *Inferred from Curator* (*IC*) code (for example, a protein

Experimental annotations	Automatically assigned annotations
IDA: Inferred from Direct Assay	IEA: Inferred from Electronic Annotation
IPI: Inferred from Physical Interaction	
IMP: Inferred from Mutant Phenotype	
IGI: Inferred from Genetic Interaction	
IEP: Inferred from Expression Pattern	
High throughput experimental annotations	Curated non-experimental annotations
HTP: High ThroughPut evidence	IBA: Inferred from Biological Aspect of Ancestor
HDA: High throughput Direct assay evidence	ISS: Inferred from Sequence or Structural similarity
HMP: High throughput Mutant Phenotype evidence	- ISO: Inferred from Sequence Orthology
HGI: High throughput Genetic Interaction evidence	- ISA: Inferred from Sequence Alignment
HEP: High throughput Expression Pattern evidence	- ISM: Inferred from Sequence Model
	TAS: Traceable Author statement
	IC: Inferred by Curator
	ND: No biological Data available

Fig. 3 The main evidence codes used by the GOC project.

with transcription factor activity is expected to be active in the nucleus of eukaryotic cells). Finally, when no data are available for an aspect of GO for a given protein, a curator can annotate the root term of the ontology (*molecular_function*, *biological_process*, or *cellular_component*), and use the evidence code *ND*. More details about the use of evidence codes and how they are used in GO can be found in [Gaudet et al. \(2017\)](#); [Chibucos et al. \(2017\)](#).

The Reference represents the source of the annotation. For annotations based on experimental evidence, the Reference contains the PubMed (PMID or PMC) accession ID. When the evidence code denotes an automatically assigned annotation, the reference will contain one of the GO_REF codes maintained by the GO Consortium (see “Relevant Websites” section) that details the automated assignment methods.

Contributing groups. Various contribute annotations, including UniProt ([UniProt Consortium, 2014](#)), Mouse Genome Informatics ([Drabkin et al., 2012](#)), Saccharomyces Genome Database ([Hong et al., 2008](#)), Wormbase ([Harris et al., 2010](#)), Flybase ([dos Santos et al., 2015](#)), dictyBase ([Gaudet et al., 2011](#)), EcoCyc ([Keseler et al., 2009](#)) and the Functional Gene Annotation group at University College London ([Buchan et al., 2010](#)). While most funding for these efforts come from the US National Institutes of Health, many smaller funding streams have enabled a wide consortium of groups to contribute to GO resources.

In addition to the above-mentioned groups who have dedicated resources to produce GO annotations, the GOC is fortunate to have fostered collaborations with research groups that can improve the ontology, the annotations, or both. Major revisions or expansions of a specific GO domain are often undertaken in consultation with domain experts. Notable successful collaborative projects include that of the immune system ([Diehl et al., 2007](#)), heart development ([Khodiyar et al., 2011](#)), kidney development ([Alam-Faruque et al., 2014](#)), fetal development ([Wick et al., 2014](#)), muscle processes and cellular components ([Feltrin et al., 2009](#)), transcription ([Tripathi et al., 2013](#)), and cilium biology ([van Dam et al., 2013](#)). Some model organism databases invite direct contributions from researchers, such as PomBase for *Schizosaccharomyces pombe*, who has developed curation tool, CANTO ([Rutherford et al., 2014](#)).

Members of the broader community are welcome to suggest improvements to the ontology and annotations. The GO project is now managed via GitHub (see “Relevant Websites” section). Users are welcome to submit their feedback via the tracker. Details about contributing to GO are available on the GOC website (see “Relevant Websites” section).

Main Applications of the Gene Ontology

One of the main uses of GO is to analyze the results of high-throughput experiments. Two types of analyses can be done to interpret this type of data. One is to group the location or function of genes that are over- or under-expressed (enrichment

analysis). In functional profiling, GO is used to determine which processes are different between sets of genes. This is done by using a likelihood-ratio test to determine if GO terms are represented differently between the gene sets (Mi *et al.*, 2013).

GO can also be used to infer the function of unannotated genes. Gene prediction with significant similarity to annotated genes can be assigned one or several of the functions of the characterized genes (Conesa and Götz, 2008; Meehan *et al.*, 2013). Other methods such as the presence of specific protein domains can also be used to assign GO terms (Burge *et al.*, 2012; Pedruzzi *et al.*, 2015).

Another usage of GO is as a dictionary to support text mining of the biomedical literature. In this case domain-specific concepts derived from GO are used to classify the literature or attach concepts to documents (Ruch, 2017).

Accessing Gene Ontology Project Data

Ontology: There are three different editions of the GO, in increasing order of complexity: *Go-basic*, *go*, and *go-plus*.

Go-basic: The basic edition of the GO, filtered such that annotations can be propagated up the graph (*i.e.*, it does not contain any cycles). The relations included are *is a*, *part_of*, *regulates*, *negatively_regulates*, and *positively_regulates*. Moreover, *go-basic* excludes relationships that cross different aspect of the ontology. This version is available in Open Biological Ontology (OBO) format only.

Go: This core edition of the GO includes additional relationship types, including some that span the three GO aspects, such as *has_part* and *occurs_in*, connecting the otherwise disjoint hierarchies found in *go-basic*. This version of the GO ontology is available in OBO format and in OWL-RDF/XML.

Go-plus: This is the most expressive edition of the GO; it includes more relationships than the *go* file, as well as connections to external ontologies, including the Chemical Entities of Biological Interest ontology (ChEBI; (Hastings *et al.*, 2013)), the Uberon anatomy ontology (Mungall *et al.*, 2012), and the Plant Ontology for plant structure/stage (PO (Cooper and Jaiswal, 2016)). It also includes import modules that are minimal subsets of those ontologies. This version of the GO ontology is available in OWL-RDF/XML.

See “Relevant Websites” section for instructions to download those files.

Ontology subsets. Gene Ontology subsets (also known as *slims*) are reduced versions of the ontology containing fewer terms and usually designed to summarize and aggregate the available data for specific application; for *e.g.*, species-specific subsets or more generic subsets with informative terms in various categories). Subsets are particularly useful for giving a high level summary of GO annotations. Further information is available from the GO website (see “Relevant Websites” section).

Annotations. Annotations are available in Gene Association File format (GAF) (see Relevant Websites Section). A newer format, GPAD, is designed to be more normalized than GAF, and is intended to work in conjunction with a separate format for exchanging gene product information (GPI). Details about the contents of the GAF format can be found in Gaudet *et al.* (2017).

Web interfaces. Association of GO terms with gene products is made by different groups who contribute the data to a central database. That data is disseminated by the GO consortium via the AmiGO platform (see “Relevant Websites” section), the official web-based open-source tool for querying, browsing, and visualizing the Gene Ontology and annotations. AmiGO supports basic searching, browsing, the ability to download custom data sets, and a common question wizard interface.

Best practices for using GO data. GO and its annotations are updated and released daily. Therefore, for reproducibility and correct reporting of methodology, researchers should provide the data at which files used for a particular analysis were downloaded (Blake, 2013).

Future Directions

The expressivity of GO annotations is limited by the fact that the current paradigm associates a gene product with a GO class, but there are no links between independent annotations. One example is a steroid hormone receptor, which act as a hormone receptor when localized in the cytosol, and upon binding to its ligand, translocates to the nucleus where it acts as a transcription factor. In GO, the gene encoding that receptor would be annotated to both molecular functions (transcription factor activity and receptor activity) and two cellular components (cytosol and nucleus), but the connections between these annotations are unfortunately not part of the data model. The GO Consortium is working towards the elaboration of a new data model, the GO-Causal Activity Model (GO-CAM), which links different annotations to each other as appropriate, to produce a more complete model of biology. Improving the expressivity of GO annotations will enable researchers working on logic-based models in systems biology to use the rich GO data to enhance their models.

Closing Remarks

GO is among the most significant informatics resources in the biomedical research community, providing scientists a way to interpret genomic, proteomic and transcriptomic data, and discover gene function and build genomic profiles. Understanding the design and data production in GO is essential for researchers to make the best use of this extensive resource.

Acknowledgements

The GO project is funded by the NIH HG002273–17 to the multiple PI team of Judith A. Blake, J. M. Cherry, Suzanna E. Lewis, Paul Sternberg, and Paul D. Thomas.

See also: Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: Introduction. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Ontology: Introduction

References

- Alam-Farouque, Y., Hill, D.P., Dimmer, E.C., *et al.*, 2014. Representing kidney development using the gene ontology. *PLOS ONE* 9, e99864.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *The gene ontology consortium. Nat. Genet.* 25, 25–29.
- Bard, J., Rhee, S.Y., Ashburner, M., 2005. An ontology for cell types. *Genome Biol.* 6, R21.
- Benson, D.A., Cavanaugh, M., Clark, K., *et al.*, 2017. GenBank. *Nucleic Acids Res.* 45, D37–D42.
- Blake, J.A., 2013. Ten quick tips for using the gene ontology. *PLOS Comput. Biol.* 9, e1003343.
- Buchan, D.W.A., Ward, S.M., Lobley, A.E., *et al.*, 2010. Protein annotation and modelling servers at University College London. *Nucleic Acids Res.* 38, W563–W568.
- Burge, S., Kelly, E., Lonsdale, D., *et al.*, 2012. Manual GO annotation of predictive protein signatures: The InterPro approach to GO curation. *Database J. Biol. Databases Curation* 2012, bar068.
- Chibucos, M.C., Siegle, D.A., Hu, J.C., Giglio, M., 2017. The evidence and conclusion ontology (eco): Supporting go annotations. *Methods Mol. Biol. Clifton NJ* 1446, 245–259.
- Conesa, A., Götz, S., 2008. Blast2GO: A comprehensive suite for functional analysis in plant genomics. *Int. J. Plant Genomics* 2008, 619832.
- Cooper, L., Jaiswal, P., 2016. The plant ontology: A tool for plant genomics. *Methods Mol. Biol. Clifton NJ* 1374, 89–114.
- Cooper, L., Walls, R.L., Elser, J., *et al.*, 2013. The plant ontology as a tool for comparative plant anatomy and genomic analyses. *Plant Cell Physiol.* 54, e1.
- de Matos, P., Alcántara, R., Dekker, A., *et al.*, 2010. Chemical Entities of Biological Interest: An update. *Nucleic Acids Res.* 38, D249–D254.
- Diehl, A.D., Lee, J.A., Scheuermann, R.H., Blake, J.A., 2007. Ontology development for biological systems: Immunology. *Bioinform. Oxf. Engl.* 23, 913–915.
- dos Santos, G., Schroeder, A.J., Goodman, J.L., *et al.*, 2015. FlyBase: Introduction of the *Drosophila melanogaster* release 6 reference genome assembly and large-scale migration of genome annotations. *Nucleic Acids Res.* 43, D690–D697.
- Drabkin, H.J., Blake, J.A., Mouse Genome Informatics Database, 2012. Manual Gene Ontology annotation workflow at the Mouse Genome Informatics Database. *Database J. Biol. Databases Curation* 2012, bas045.
- Eilbeck, K., Lewis, S.E., Mungall, C.J., *et al.*, 2005. The sequence ontology: A tool for the unification of genome annotations. *Genome Biol.* 6, R44.
- Feltrin, E., Campanaro, S., Diehl, A.D., *et al.*, 2009. Muscle research and gene ontology: New standards for improved data integration. *BMC Med. Genomics* 2, 6.
- Gaudet, P., Fey, P., Basu, S., *et al.*, 2011. DictyBase update 2011: Web 2.0 functionality and the initial steps towards a genome portal for the Amoebozoa. *Nucleic Acids Res.* 39, D620–D624.
- Gaudet, P., Škunca, N., Hu, J.C., Dessimoz, C., 2017. Primer on the gene ontology. *Methods Mol. Biol. Clifton NJ* 1446, 25–37.
- Gene Ontology Consortium, 2015. Gene ontology consortium: Going forward. *Nucleic Acids Res.* 43, D1049–D1056.
- Harris, T.W., Antoshechkin, I., Bieri, T., *et al.*, 2010. WormBase: A comprehensive resource for nematode research. *Nucleic Acids Res.* 38, D463–D467.
- Hastings, J., 2017. Primer on Ontologies. *Methods Mol. Biol. Clifton NJ* 1446, 3–13.
- Hastings, J., de Matos, P., Dekker, A., *et al.*, 2013. The ChEBI reference database and ontology for biologically relevant chemistry: Enhancements for 2013. *Nucleic Acids Res.* 41, D456–D463.
- Hong, E.L., Balakrishnan, R., Dong, Q., *et al.*, 2008. Gene Ontology annotations at SGD: New data sources and annotation methods. *Nucleic Acids Res.* 36, D577–D581.
- Keseler, I.M., Bonavides-Martínez, C., Collado-Vides, J., *et al.*, 2009. EcoCyc: A comprehensive view of *Escherichia coli* biology. *Nucleic Acids Res.* 37, D464–D470.
- Khodiyar, V.K., Hill, D.P., Howe, D., *et al.*, 2011. The representation of heart development in the gene ontology. *Dev. Biol.* 354, 9–17.
- Lewis, S.E., 2017. The vision and challenges of the gene ontology. *Methods Mol. Biol. Clifton NJ* 1446, 291–302.
- Meehan, T.F., Vasilevsky, N.A., Mungall, C.J., *et al.*, 2013. Ontology based molecular signatures for immune cell types via gene expression analysis. *BMC Bioinform.* 14, 263.
- Mi, H., Huang, X., Muruganujan, A., *et al.*, 2017. PANTHER version 11: Expanded annotation data from gene ontology and reactome pathways, and data analysis tool enhancements. *Nucleic Acids Res.* 45, D183–D189.
- Mi, H., Muruganujan, A., Casagrande, J.T., Thomas, P.D., 2013. Large-scale gene function analysis with the PANTHER classification system. *Nat. Protoc.* 8, 1551–1566.
- Mungall, C.J., Tornai, C., Gkoutos, G.V., Lewis, S.E., Haendel, M.A., 2012. Uberon, an integrative multi-species anatomy ontology. *Genome Biol.* 13, R5.
- Natale, D.A., Arighi, C.N., Blake, J.A., *et al.*, 2017. Protein Ontology (PRO): Enhancing and scaling up the representation of protein entities. *Nucleic Acids Res.* 45, D339–D346.
- Pedruzzi, I., Rivoire, C., Auchincloss, A.H., *et al.*, 2015. HAMAP in 2015: Updates to the protein family classification and annotation system. *Nucleic Acids Res.* 43, D1064–D1070.
- Rhee, S.Y., Wood, V., Dolinski, K., Draghici, S., 2008. Use and misuse of the gene ontology annotations. *Nat. Rev. Genet.* 9, 509–515.
- Ruch, P., 2017. Text mining to support gene ontology curation and vice versa. *Methods Mol. Biol. Clifton NJ* 1446, 69–84.
- Rutherford, K.M., Harris, M.A., Lock, A., Oliver, S.G., Wood, V., 2014. Canto: An online tool for community literature curation. *Bioinform. Oxf. Engl.* 30, 1791–1792.
- Thomas, P.D., 2017. The gene ontology and the meaning of biological function. *Methods Mol. Biol. Clifton NJ* 1446, 15–24.
- Tripathi, S., Christie, K.R., Balakrishnan, R., *et al.*, 2013. Gene ontology annotation of sequence-specific DNA binding transcription factors: Setting the stage for a large-scale curation effort. *Database J. Biol. Databases Curation* 2013, bat062.
- UniProt Consortium, 2014. Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Res.* 42, D191–D198.
- van Dam, T.J., Wheway, G., Slaats, G.G., SYSCILIA Study Group/Huynen, M.A., Giles, R.H., 2013. The SYSCILIA gold standard (SCGSv1) of known ciliary components and its applications within a systems biology consortium. *Cilia* 2, 7.
- Wick, H.C., Drabkin, H., Ngu, H., *et al.*, 2014. DFLAT: Functional annotation for human development. *BMC Bioinform.* 15, 45.

Further Reading

Gaudet, P., Dessimoz, C., 2017. Gene ontology: Pitfalls, biases, and remedies. In: *Proceedings of the Gene Ontology Handbook*, pp. 189–206. *Methods in Molecular Biology*, ISSN 1940–6029.

- Gaudet, P., Škunca, N., Hu, J., Dessimoz, C., 2017. Primer on the gene ontology. In: Proceedings of the Gene Ontology Handbook, pp. 25–40. Methods in Molecular Biology, ISSN 1940–6029.
- Hastings, J., 2017. Primer on ontologies. In: Proceedings of the Gene Ontology Handbook, pp. 3–14. Methods in Molecular Biology, ISSN 1940–6029.
- Lewis, S.E., 2017. The vision and challenges of the gene ontology. In: Proceedings of the Gene Ontology Handbook, pp. 291–302. Methods in Molecular Biology, ISSN 1940–6029.
- Lovering, R.C., 2017. How does the scientific community contribute to Gene Ontology? In: Proceedings of the Gene Ontology Handbook, pp. 85–96. ISSN 1940–6029.
- Monoz-Torres, M., Carbon, S., 2017. Get GO! retrieving GO data using AmiGO, QuickGO, API, files, and tools. In: Proceedings of the Gene Ontology Handbook, pp. 149–160. Methods in Molecular Biology, ISSN 1940–6029.
- Poux, S., Gaudet, P., 2017. Best practices. In: Proceedings of the Manual Annotation With the Gene Ontology pp. 41–54. Methods in Molecular Biology, ISSN 1940–6029.
- Thomas, P.D., 2017. The gene ontology and the meaning of biological function. In: Proceedings of the Gene Ontology Handbook, pp. 15–24. ISSN 1940–6029.

Relevant Websites

- <http://amigo.geneontology.org>
AmiGO, the Gene Ontology Consortium's web-based set of tools for searching and browsing the Gene Ontology database.
- <http://geneontology.org/page/contributing-go>
Contributing to GO.
- <http://geneontology.org/page/download-annotations>
Download GO annotations.
- <http://geneontology.org/page/download-ontology>
Download Ontology.
- www.geneontology.org
Gene Ontology Consortium.
- <http://www.geneontology.org/cgi-bin/references.cgi>
GO Reference Collection.
- <http://geneontology.org/page/go-slim-and-subset-guide>
GO Slim and Subset Guide.
- <https://www.ebi.ac.uk/QuickGO/>
QuickGO - EMBL-EBI.
- <https://github.com/geneontology/>
The Gene Ontology Consortium software and issue tracker on GitHub.
- https://en.wikipedia.org/wiki/Web_Ontology_Language
Web Ontology Language.

Biographical Sketch

Pascale Gaudet works in the field of biocuration since 2003, working on various projects, including the Model Organism Database dictyBase, the neXtProt database on human proteins of the Swiss Institute of Bioinformatics (SIB), and the Gene Ontology project. Since 2017 Pascale Gaudet is the Project Manager of the GO project. Additionally, Pascale Gaudet is one of the founding members and the first chairperson of the International Society for Biocuration (ISB). The ISB seeks to connect biocurators, developers, and researchers with an interest in biocuration, both amongst themselves and with the users and funders of biological informatics resources.

Protein Functional Annotation

Pier Luigi Martelli, Giuseppe Profiti, and Rita Casadio, University of Bologna, Bologna, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Functional annotation is one of the most crucial activities of a Bioinformatician. As a matter of fact, as long as a sequence of characters, no matter of nucleic acids or protein residues, is a string, biologists can make little usage of it. It is therefore mandatory to try to endow with structural and functional features any string of biological origin. In the following, in spite of being deeply aware of the relevance of the annotation of the whole genome in different species, we will focus into the problem of protein sequence annotation that starts immediately after the recognition of a coding region in the DNA sequence. Functional annotation is a prerequisite for the inclusion of the sequence in the public archive/database.

UniProt

Today, the largest public repository of protein sequences is the Universal Protein Resource (UniProt) (<http://www.uniprot.org/help/about>). UniProtKB (Uniprot Knowledgebase, UniProtKB/Swiss-Prot 2017_02) comprises two databases, TrEMBL and SwissProt. The difference consists in the quality of the annotation. In SwissProt (including 553,655 sequences), protein annotation is manually curated with the aid of literature and computational analysis; in TrEMBL (including 77,483,538 sequences) annotation is automatically computed and not reviewed. More than 95% of the protein sequences come from the translations of coding sequences (CDS) submitted to the EMBL-Bank/GenBank/DBJ nucleotide sequence resources of the International Nucleotide Sequence Database Collaboration (INSDC). The problem of protein annotation starts here and it is extremely relevant, since all our molecular knowledge is based on what the database contains.

How many proteins do we know? For each record, Uniprot includes indications relative to the protein existence. The protein existence can derive from wet experiments (e.g., 3D structure, biochemical characterization, protein–protein interaction). In this case, the record lists “Evidence at protein level.” Alternatively, the protein existence may be associated with the existence of a transcript (“Evidence at transcript level”). It is quite evident from [Table 1](#) that proteins with experimental evidence and evidence at the transcript level are only 1.5% of the total database. About 23% of the proteins are inferred by sequence similarity with existing orthologues in evolutionarily related species. The remaining 75% are “predicted proteins,” without any evidence of the above qualifiers (http://www.uniprot.org/docs/pe_criteria). The percentage of predicted proteins is much less in the SwissProt database (2.5%). Interestingly, SwissProt contains some 0.3% of “Uncertain” proteins, so to say proteins that may be products of pseudogenes, dubious Coding DNA Sequences and genes, but that have been manually curated. Protein allocation in TrEMBL or SwissProt is indeed entirely dependent on the manual curation or automatic procedure of the annotation (<http://www.uniprot.org/help/about>). From the statistics, one may conclude that we know with some experimental evidence only 1.7% of the proteins listed in Uniprot KB (78,037,193).

The Gene Ontology

The Gene Ontology project started in 1998 with the goal of providing a controlled vocabulary of terms for describing gene product functional characteristics in a computationally feasible way, by associating words, actions and sentences suited to characterize the complete function of a gene with numbered terms (GO terms). GO terms have three major routes: biological process (BP), molecular function (MF) and cellular components (CC). Each route is hierarchically organized from more general (less informative) to more specific terms (leaves). Since its foundations, the Gene Ontology database has developed formal ontologies and constantly has been revised to include new discoveries. Information was derived from experiments described in some 100,000 peer-reviewed scientific papers (<http://www.geneontology.org/>). AmiGO 2 (<http://amigo.geneontology.org/amigo>) is a tool that allows the gene-GO term association search in the database. The release (March 2017) of GO includes 29,385 BP, 10,543 MF and

Table 1 Some statistics of the latest release of UniProtKB (release 2017_02).

<i>Protein existence</i>	<i>#UNIPROT/TrEMBL entries</i>	<i>#UNIPROT/SwissProt entries</i>
Evidence at protein level	125,718	94,730
Evidence at transcript level	1,066,148	57,857
Inferred from homology	17,797,259	385,421
Predicted	58,494,413	13,697
Uncertain	0	1950

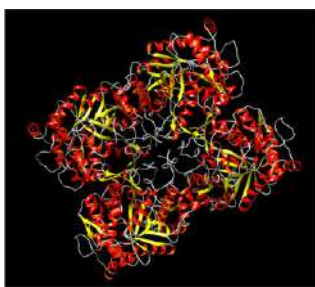
4045 CC terms. A GO term has experimental origins when the gene/gene product has a physical characterization in a paper that describes the corresponding association. Experimental Evidence codes, as well as Computational Analysis and Author Statement evidence codes are listed at <http://www.geneontology.org/page/guide-go-evidence-codes>

Foundations of Functional Annotations

Our information on the biological complexity can be dissected into three major levels of knowledge: sequence, structure, and function. Accordingly, we may think of the relationships among different spaces (universes). **Fig. 1** shows the relationships among the sequence and the structure space. Information is maximal when we know the structure of the protein, stored in the Protein Data Bank (PDB, <http://www.rcsb.org/pdb/home/home.do>). The structure of the protein is mainly derived from its electron density determination, and this allows us to determine even its function based on quantum molecular computations. Problem is that we know the structure of some 124,114 proteins (PDB, March 2017); however, only 40,463 UniProt entries are linked to nearly all the PDB entries. This is mostly due to redundancy of structures with respect to the same reference sequence. In conclusions, we have information based on the structure for only 0.05% of the total number of sequences. These proteins are associated with about 23,000 experimental GO terms (**Table 2**). Ideally, we can then connect the space of functions with that of structures and sequences. The question poses as to whether it is possible to infer functions from sequences, taking advantage of what it can be derived in terms of basic association rules from known structures and their associated sequences and functions. Evolutionary relationships can help: we know from biological observations that functions are conserved through species, that orthologous proteins conserve functions, have similar structures and similar sequences. We also know that even when alignment methods fail in retrieving significant scoring below 30% sequence identity, proteins may conserve structures and functions, being included in what we call remote homologs. All this biological information, related to the general concept of evolution, has been promoting in the last decades an enormous effort in developing computational methods. Although different and based on different strategy, different methods try to address the problem of how is it possible to fill the gap between the deluge of protein sequences and their structural and functional annotation.

```
>M34696.1 S.solfataricus beta-D-galactosidase (lacS) gene,
complete cds. : Location:1..1000
AAGGAGAAACTTGGCAGTTTATACTTGACAGTAGGTTGTGGAGTGATGACTGGATCAAT
ACTAGGAGGAGTAGCATATATAATTACGTTACACAATTTTATAACCAATATTTCAATAGA
CCTTATGCTTATCCTATCCTCTATCTAAGATTCTCGGTATCTCCCTATCTTTGACCAT
AAAAGATACCTCGCTCAAAGCTTAAATAATATTAATCATAAATAAAGTCATGTACTCATTT
CCAAATAGCTTTTAGGTTTGGTTGGTCCCGGCGGATTTCAATCAGAAATGGGAACACCA
GGGTCAGAAAGATCCAAATACTGACTGGTATAAATGGGTTTCATGATCCAGAAAACATGGCA
GCGGATTAGTAAGTGGAGATCTACCAGAAAATGGGCCAGGCTACTGGGGAACTATAAG
ACATTTACGATAATGCACAAAAATGGGATTAAAAATAGCTAGACTAAATGTGGAATGG
TCTAGGATATTTCTTATCCATTACCAAGGCCACAAAACCTTGATGAATCAAAACAAGAT
GTGACAGAGGTTGAGATAAACGAAACAGGTTAAAGAGACTTGACGAGTACGCTAATAAAA
GACGATTAAACCATACAGGGAATATTCAGGATCTTAAAGTAGAGGACTTTACTTTT
ATACTAAACATGTATCATTTGGCCATTACCTCTATGTTACACGACCCAATAAGAGTAAGA
AGAGGAGATTTTACTGGACCAAGTGGTTGGCTAAGTACTAGAACAGTTTACGAATTCGCT
AGATTCTCAGTTATATAGCTTGGAAATTCGATGATCTAGTGGATGAGTACTCAACAATG
AATGAACCTAACGTTGTTGGAGGTTTAGGATACGTTGGTGTAAAGTCCGGTTTCCCCCA
GGATACCTAAGCTTTGAACCTTCCCGTAGGCATATGTATAACATCATTCAGCTCACGCA
AGACGCTATGATGGGATAAAGAGTGTCTTCTAAAAACCAAG
```

```
>BGAL_SULSO Beta-galactosidase
MYSFNSFRFGWSQAGFQSEMGTPGSEDPNTDWYKWHDPENMAAGLVSGDLPENGPYW
GNYKTFHDNAQKMLKIALRLNVEWSRIFPNPLRPQNFDESKQDVTEVEINENELKRLDE
YANKDALNHYREIKDLKSRGLYFILNMYHWLPLWLHDPIDRVRRGDFTPGSGWLSRTV
YEFARFSAYIAWKFDDLVDEYSTMNEPMVVGGLGVGVKSGFPFPGYLSFELSRAMYNII
QAHARAYDGIKSVSKPVGIIYANSSFPQLTDKDMEAHEMAENDNRWFFDAIIRGEITR
GNEKIVRDDLKGRLDWIGVNYITRTVVKRTEKGYVSLGGYGHGCERNVSLAGLPTSDFG
WEFFPEGLYDLVTKYWNRYHLYMYVTENGIAADDADYQRPYLVSHVYQVHRAINSADVR
GYLHWSLADNYESWGSFMRFGLLKVDYNTKRLYWRPSALVYREIATNGAITDEIEHLNS
VPPVKPLRH
```



1GOW

GenBank

UniProt/SwissProt

PDB

I
N
F
O
R
M
A
T
I
O
N

Fig. 1 Information content increases from genes to protein structures.

Table 2 Links between sequence, structure and function in SwissProt

<i>Protein existence</i>	<i>#UNIPROT/SwissProt entries</i>	<i>#Distinct Protein Data Banks (PDBs)</i>	<i>#Distinct experimental GO terms^a</i>	<i>#Distinct nonexperimental GO terms^a</i>
Evidence at protein level	94730	93843 ^a	22861	20135
With PDB and exp GO	13056	61214	15018	12802
With PDB and no exp GO	11227	35535	0	6546
With exp GO and no PDB	34513	0	19676	15885
No PDB and no exp GO	35934	0	0	11532
Evidence at transcript level	57857	77	9506	14534
With PDB and exp GO	18	22	121	203
With PDB and no exp GO	53	55	0	205
With exp GO and no PDB	11029	0	9492	9084
No PDB and no exp GO	46757	0	0	12949
Inferred from homology	385421	92	3376	10895
With PDB and exp GO	6	6	14	17
With PDB and no exp GO	80	86	0	233
With exp GO and no PDB	4619	0	3372	4384
No PDB and no exp GO	380716	0	0	10293
Predicted	13697	5	679	1556
With PDB and no exp GO	4	5	0	8
With exp GO and no PDB	906	0	679	698
No PDB and no exp GO	12787	0	0	1228
Uncertain	1950	1	71	849
With PDB and no exp GO	1	1	0	4
With exp GO and no PDB	72	0	71	177
No PDB and no exp GO	1877	0	0	744

^aCounting is performed considering experimental and electronic labels, as detailed at <http://www.geneontology.org/page/guide-go-evidence-codes>

The Annotation Process in UniProt

UniProt performs both expertly curated manual and automatic annotation (a flow diagram is reported at <http://www.uniprot.org/help/biocuration>) and highlights also the interplay between manual and automatic annotation.

Manual Annotation

Manual annotation in UniProt is a multistep-curated process that includes similarity search, feature prediction, and validation (http://www.uniprot.org/help/manual_curation). Details are described in the standard operating procedure (SOP) for UniProt manual curation (<http://www.uniprot.org/help/biocuration>).

Briefly, a sequence is aligned with other well-known sequences with alignment tools such as Blast, and information for its annotation derives from a variety of resources, including scientific literature, sequence analysis tools, phylogenetic and comparative genomics databases (Ensembl Compara, <http://www.ensembl.org/info/genome/compara/index.html>, and other species-specific collections). A protein is practically annotated with InterPro (<https://www.ebi.ac.uk/interpro/about.html>), a cluster of different resources that classifies sequences at superfamily, family and subfamily levels and comprises predictive models of protein functions from several databases. High-quality automated and manual annotation of proteins (HAMAP, <http://hamap.expasy.org/>) is part of the InterPro cluster for family classification. For association of functional GO terms to structural and functional features, InterPro integrates InterPro2GO that maps GO term to InterPro entries. The core of the classification scheme is the software package InterProScan that allows scanning sequences against InterPro features (<https://www.ebi.ac.uk/interpro/interproscan.html>) and integrating prediction of functional domains and important sites. For inclusion of distinguished features such as transmembrane regions and signal peptides, manual annotation follows a set of expert rules (<http://www.uniprot.org/help/transmem>; <http://www.uniprot.org/help/signal>).

Automatic Annotation

The automatic annotation process of UniProt is a multistep process performed by a pipeline that is based again on the protein classification schemes of InterPro. Once the structural and functional features are expertly determined, two prediction systems are active for the automatic annotation: UniRule and the Statistical Automatic Annotation System (SAAS). Both are essentially based on the application of rules (if conditions → then annotation), differently extracted. UniRule collects rules (some 5557 Identifiers (ID)) derived by curators from manually annotated entries and templates, and routinely annotates properties such as the protein name, functions (GO terms), catalytic activity, pathway membership, and subcellular location, along with sequence specific

information, such as the positions of posttranslational modifications and active sites. Similarly, SAAS generates automatic rules for functional annotation from expertly annotated entries in UniProtKB. The algorithm uses machine learning to find the most concise rule for an annotation based on the properties of sequence length, InterPro group membership and taxonomy. SAAS includes 13,674 IDs and can annotate protein properties such as function (GO term), catalytic activity, pathway membership, and sub-cellular location, with the exclusion of protein names and feature predictions. A sequence analysis method (SAM) is adopted to enrich records with extra sequence-specific information. Predictions of sequence features such as Signal, Transmembrane and Coil regions are computed by other prediction methods, including THMM, Phobius, and SignalP (<http://www.uniprot.org/help/sam>).

Our Present Knowledge

When addressing the problem of functional annotations, we should cope with what we know. In SwissProt, some 94,730 reviewed sequences with evidence at the protein level have links with some 93,843 PDB files and only 22,861 GO terms with experimental labels (Table 2). Some other 20,135 GO terms have only electronic indexes. The number of protein sequences with experimentally detected functions increases when we include proteins with evidence at the transcript level. Therefore, the actual situation is not particularly sound in terms of putative subsets to be adopted for training/testing specific computational methods to predict function from sequence. In the following, we will briefly review the actual state of the art methods (some of them have been already mentioned, being part of the InterPro consortium).

Computational methods have been implemented to associate uncharacterized proteins (targets) to terms describing the function, mostly complying with the Gene Ontology. In general, these methods compare a target to a set of annotated proteins in order to extract and transfer the functional terms. Different approaches are available, in relation to the information available for the target protein at the sequence and/or structure levels and to the similarity with the functionally characterized proteins (Chothia and Lesk, 1986). In the following methods are clustered according to their basic principles of implementation (Table 3).

Methods Based on Structure Similarity

When the target protein is endowed with experimental 3D structure, methods based on structure superimposition can be applied, relying on the observation that a high level of structural similarity among proteins generally implies functional similarity. This approach can be extended to proteins with modeled 3D structure, with limitations that strongly depend on the reliability of the model.

Methods for large-scale comparison of protein structures are therefore adopted to scan the target protein against the Protein Data Bank to retrieve annotation terms. Among them, CE (Shindyalov and Bourne, 2001), PDB-eFOLD (Krissinel and Henrick, 2004), DALI (Holm and Rosenström, 2010), VAST (Madej et al., 2014). They adopt different strategies to compare structures at the level of the backbone atoms. However, the conservation of the overall fold is not sufficient to ensure function similarity. A comprehensive structural classification of proteins domains is available at CATH (<http://www.cathdb.info/>, Sillitoe et al., 2015) and it clearly shows a high heterogeneity of functional terms for some of the folds, including TIM barrel fold, ferredoxin fold, and Rossmann fold.

Table 3 Methods for function prediction

<i>Name</i>	<i>Input</i>	<i>WEB address</i>
CE	Structure	http://source.rcsb.org/jfatcatserver/
DALIStructure	Structure	http://ekhidna.biocenter.helsinki.fi/dali_server/
PDB-eFOLD	Structure	http://www.ebi.ac.uk/msd-srv/ssm/
VAST	Structure	https://structure.ncbi.nlm.nih.gov/Structure/VAST
ProFunc	Structure	http://www.ebi.ac.uk/thornton-srv/databases/ProFunc/
ProBis	Structure	https://probis.nih.gov/
Blast2GO	Sequence	https://www.blast2go.com/
Argot2	Sequence	http://www.medcomp.medicina.unipd.it/Argot2/
SIFTER	Sequence	http://sifter.berkeley.edu/
BAR3	Sequence	http://bar.biocomp.unibo.it/
COG	Sequence	https://www.ncbi.nlm.nih.gov/COG/
SMART	Sequence	http://smart.embl-heidelberg.de/
PFAM	Sequence	http://pfam.xfam.org/
FunFams	Sequence	http://www.cathdb.info/
CCD-SPARCLE	Sequence	https://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml
PROSITE	Sequence	http://prosite.expasy.org/
PRINTS	Sequence	http://130.88.97.239/PRINTS/index.php
INGA	Sequence	http://protein.bio.unipd.it/inga
FFPred	Sequence	http://bioinf.cs.ucl.ac.uk/web_servers/ffpred/ffpred_help/
CombFunc	Sequence	http://www.sbg.bio.ic.ac.uk/~mwass/combfunc/

The reliable association between structure and function requires the detailed conservation of the protein sites important for catalytic activity, substrate binding, and intermolecular interaction. Repertoires of 3D templates of enzyme catalytic sites are available at Catalytic Site Atlas (<http://www.ebi.ac.uk/thornton-srv/databases/CSA/>, Furnham *et al.*, 2014.) and Pocketome (<http://www.pocketome.org/>, Kufareva *et al.*, 2012).

When global structural alignments of the target do not retrieve significant results in the PDB, the recognition of structural motifs can assist the functional annotation. Methods implementing local structure alignment, such as ProFunc (Laskowski *et al.*, 2005) can identify local regions, functionally relevant and conserved across several globally divergent structures. Putative active site pockets on the protein surface can be determined with geometric-based approaches like Ligsite (Huang and Schroeder, 2006) and ProBIS (Konc *et al.*, 2015).

Methods Based on Sequence Similarity

When the 3D structure of the target proteins is not available, protein annotation is routinely performed by aligning the residue sequence against the annotated proteins available in sequence databases. The inference of functional similarity requires a higher sequence identity with respect to structural similarity. Generally, the more specific the function to be transferred, the higher the level of identity. Rost (2002) showed that less than 30% of the enzyme sequences sharing more than 50% identity have entirely identical EC numbers and enzymes with different specific functions can align with BLAST E-values below 10^{-50} . Moreover, multidomain proteins increase the complexity of the protein sequence–structure–function relationship and require a careful check of the extent of alignment length with respect to the target sequence (coverage).

BLAST (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>) is nowadays considered as the baseline method for transferring function annotations upon similarity search against a database of annotated sequence. Due to the complex patterns of association between sequences and function, statistical procedures are routinely applied on the set of retrieved terms to strengthen the reliability of the transferred annotations. For example, the widely adopted Blast2GO (Götz *et al.*, 2008) weights each term by considering the number of retrieved sequences it is associated to and the evidence codes linked to the annotations in the database. Argot2 (Falda *et al.*, 2012) weights GO terms according to their semantic similarity relations and the alignment scores of the retrieved sequences.

Phylogenetic analysis has been also adopted to support the transfer of annotation (e.g., SIFTER, Sahraeian *et al.*, 2015), under the assumption that function evolves parsimoniously within a phylogeny, mainly by duplication events; the position of the query sequence in a phylogenetic tree is more informative than the mere sequence similarity. An alternative strategy to perform a reliable transfer of annotation is based on the preliminary similarity-based clustering of the protein universe into groups that can be functionally characterized on the basis of the existing annotations. COG (Galperin *et al.*, 2015) contains orthology-based clusters of microbial sequences and endows them with manually curated annotation. A complete nonhierarchical clustering of UniProt is at the basis of BAR3 (Profitti *et al.*, 2017): pairs of sequences sharing identity > 40% with an alignment coverage > 90% are linked to form a graph, whose connected components are extracted to form the clusters. Functional terms associated to clustered proteins are statistically tested for overrepresentation and significant annotation are assigned to the whole cluster. A new sequence falling in the cluster upon alignment inherits the validated terms.

Methods Based on Sequence Profiles and Motifs

Sequences sharing low level of identity with already characterized proteins present the hardest challenges to the process of annotation. Procedures based on sequence profiles are able to extend the range of application of annotation transfer. Sequence profiles are built starting from the multiple alignment of sequences or structures of protein domains already characterized at the functional level. Profiles are routinely represented with either position specific scoring matrices (PSSMs) or Hidden Markov Models (HMMs). Uncharacterized sequences can be aligned against a collection of profiles to retrieve remote similarities and transfer annotation. Profile representation allows increasing the sensitivity of searches by scoring in a different and family-dependent way substitutions and indels in each position of the alignment. The main difference among methods lies in the strategy adopted to build the collection.

Popular repositories of HMMs obtained after manual curation of multiple sequence alignments are PFAM (Finn *et al.*, 2016) and SMART (Letunic *et al.*, 2015), containing models for more than 16,000 and 1200 protein domains, respectively.

The NCBI Conserved Domain Database (CDD) collects PSSMs for curated domains, integrating information from 3D-structure, SMART, PFAM, COG and other external source databases (Marchler-Bauer *et al.*, 2017). On top of CDD, SPARCLE improves the annotation by analyzing the sequential order of the different domains associated to a protein (domain architecture) (Marchler-Bauer *et al.*, 2017).

CATH-Gene3D Functional Families (FunFams) is a compilation of 92,882 HMMs obtained starting from multiple structural alignments of domains belonging to the same CATH family and sharing the same function (Sillitoe *et al.*, 2013).

When also remote similarity search with sequence profiles fails, function annotation can exploit the presence of sequence motifs or fingerprints. They generally represent fragments of the sequence that, in the folded protein, are part of relevant active or functional sites, as highlighted in multiple sequence alignments. Motifs and fingerprints are usually represented with regular expressions or small sequence profiles. PROSITE (Sigrist *et al.*, 2013) and PRINTS (Attwood *et al.*, 2012) are two of the most popular resources, that are also included in the InterPro annotation pipeline.

Integrative and Machine Learning Based Methods

Function annotation of sequences sharing no detectable similarity with characterized proteins requires the development of integrative methods that try to bridge the gap by exploiting different sources of information, including the prediction of relevant features and the mining of interactomics and transcriptomics data. The information is integrated with different approaches, including the expert-driven definition of decision trees, the application of simple regression rules, and the implementation of machine-learning algorithms. The rules of associations between features and annotation are extracted with manual or automatic procedures by analyzing sets of characterized proteins available in datasets (training set). The unbiased estimation of the prediction performance on uncharacterized sequences requires to collect sets of annotated proteins (testing sets), separated and as different as possible from the training sets. This operation is sometimes difficult, in particular when different tools, trained on different training sets, are integrated. For that reason, critical assessment experiments are a useful way to estimate, compare and interpret the methods performance (see next session).

It is in the set of considered features that lies the major difference among the available tools.

FFPred (Minneci *et al.*, 2013) combines with support vector machines (SVM) information on residue composition and presence of low complexity regions and predictions of the following features: secondary structure, signal peptide, transmembrane regions, coiled-coil or disordered segments, phosphorylation and glycosylation sites.

The availability of experimental data on transcriptomics and interactomics leads to the so called “guilty by association” approach: coexpressed genes and proteins in physical or functional relation reliably share common functional features. For example, INGA (Piovesan *et al.*, 2015) combines information on sequence similarity, domain architecture and protein–protein interactions as derived from STRING. In MS-kNN (Lan *et al.*, 2013) a k-Nearest-Neighbor (kNN) system analyzes similar information, integrated with experimental data on gene expression. CombFunc (Wass *et al.*, 2012), combines with SVMs prediction of structural domains and ligand binding sites, sequence motifs, sequence similarity and conservation, protein–protein interaction data from STRING and MINT, coexpression data from CoexpressDB.

An alternative source of information to be integrated is the literature published in scientific papers: EVEX (Van Landeghem *et al.*, 2012.) analyzes the PubMed abstracts to mine information about relevant biomolecular events such as phosphorylation, regulation targets, binding partners, and several other, assigning confidence values to these events.

Critical Assessment of Function Annotation

As we mentioned in the previous section, the evaluation of the performance of methods for functional annotation requires the availability of testing dataset of annotated proteins not used, in any way, during the training phase of the method. Moreover, the comparison of different methods requires the adoption of uniform testing sets. In order to provide an independent and unbiased assessment of computational methods and to provide insights on the state-of-the-art of function annotation, in 2010 the first critical assessment of function annotation (CAFA, <http://biofunctionprediction.org/cafa/>), experiment was run, followed by two more editions in 2013 and 2016. CAFA was inspired by critical assessment of structure prediction (CASP) and critical assessment of prediction of interactions (CAPRI), devoted to the blind evaluation of computational methods for the prediction of the structure of proteins and of protein complexes, respectively. In CAFA experiments, organizers provide for the community a large set of sequences with unknown or incomplete function. The task consists in predicting Gene Ontology annotations for each of these sequences, before a fixed deadline. After the prediction deadline, the experiment enters the “annotation growth” period in which protein functions are expected to accumulate in public databases (6–12 months). The predictions of different methods are afterwards compared with the experimental annotation deposited in UniProt during the “annotation growth” phase.

The most updated assessment is the CAFA2, i.e., the 2013 edition (CAFA2, Jiang *et al.*, 2016). The benchmark dataset comprised 3681 sequences and 126 different prediction models were evaluated. Results show that the predictive performance is increasing with respect to the CAFA1 edition. Top-performing algorithms are specific for sub-ontology (molecular function, biological process, cellular component) and biological domain (prokaryota, eukaryota). The main index of evaluation is the F1-score, which is the harmonic mean of the precision ($\#$ of correctly predicted annotations / $\#$ of predicted annotations) and the recall ($\#$ of correctly predicted annotations / $\#$ annotations of the benchmark dataset). The best F1-scores reached on eukaryotic proteins are about 0.6, 0.4, and 0.5 for MF, BP, and CC, respectively. In the case of prokaryotes, F1-scores are higher for BP and CC (0.5 and 0.7, respectively). These results, although encouraging, show that there is plenty of room for improvement of the methods for automatic function annotation.

Acknowledgments

This work was partially supported by the Italian Ministry of Education, University and Research [PRIN 2010–2011 project 20108XYHJS] to P.L.M., [PON projects PON01_02249 and PAN Lab PONA3_00166] to R.C. and P.L.M; European Union RTD Framework Program [COST BMBS Action TD1101, Action BM1405] to R.C and University of Bologna [FARB 2012] to R.C. G.P. would like to thank also ELIXIR-IIB and ELIXIR Europe for the support.

See also: Functional Enrichment Analysis Methods. Natural Language Processing Approaches in Bioinformatics. Ontology-Based Annotation Methods. Semantic Similarity Functions and Measures. Tools for Semantic Analysis Based on Semantic Similarity

References

- Attwood, T.K., Coletta, A., Muirhead, G., *et al.*, 2012. The PRINTS database: A fine-grained protein sequence annotation and analysis resource – Its status in 2012. *Database* 2012. doi:10.1093/database/bas019.
- Chothia, C., Lesk, A.M., 1986. The relation between the divergence of sequence and structure in proteins. *EMBO Journal* 5, 823–826.
- Falda, M., Toppo, S., Pescarolo, A., *et al.*, 2012. Argot2: A large scale function prediction tool relying on semantic similarity of weighted gene ontology terms. *BMC Bioinformatics* 13, S14.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research* 44, D279–D285.
- Furnham, N., Holliday, G.L., de Beer, T.A., *et al.*, 2014. The catalytic site atlas 2.0: Cataloging catalytic sites and residues identified in enzymes. *Nucleic Acids Research* 42, D485–D489.
- Galperin, M.Y., Makarova, K.S., Wolf, Y.I., Koonin, E.V., 2015. Expanded microbial genome coverage and improved protein family annotation in the COG database. *Nucleic Acids Research* 43, D261–D269.
- Götz, S., García-Gómez, J.M., Terol, J., *et al.*, 2008. High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic Acids Research* 36, 3420–3435.
- Holm, L., Rosenström, P., 2010. Dali server: Conservation mapping in 3D. *Nucleic Acids Research* 38, W545–W549.
- Huang, B., Schroeder, M., 2006. LIGSITEcs: Predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Structural Biology* 6, 19.
- Jiang, Y., Oron, T.R., Clark, W.T., *et al.*, 2016. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology* 17, 184.
- Konc, J., Miller, B.T., Štular, T., *et al.*, 2015. ProBiS-CHARMMing: Web interface for prediction and optimization of ligands in protein binding sites. *Journal of Chemical Information and Modeling* 55, 2308–2314.
- Krissinel, E., Henrick, K., 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica D* 60, 2256–2268.
- Kufareva, I., Ilatovskiy, A.V., Abagyan, R., 2012. Pocketome: An encyclopedia of small-molecule binding sites in 4D. *Nucleic Acids Research* 40, D535–D540.
- Lan, L., Djuric, N., Guo, Y., Vucetic, S., 2013. MS-kNN: Protein function prediction by integrating multiple data sources. *BMC Bioinformatics* 14, S8.
- Laskowski, R.A., Watson, J.D., Thornton, J.M., 2005. ProFunc: A server for predicting protein function from 3D structure. *Nucleic Acids Research* 33, W89–W93.
- Letunic, I., Doerks, T., Bork, P., 2015. SMART: Recent updates, new developments and status in 2015. *Nucleic Acids Research* 43, D257–D260.
- Madej, T., Lanczycki, C.J., Zhang, D., *et al.*, 2014. MMDB and VAST + : Tracking structural similarities between macromolecular complexes. *Nucleic Acids Research* 42, D297–D303.
- Marchler-Bauer, A., Bo, Y., Han, L., *et al.*, 2017. CDD/SPARCLE: Functional classification of proteins via subfamily domain architectures. *Nucleic Acids Research* 45, D200–D203.
- Minneci, F., Piovesan, D., Cozzetto, D., Jones, D.T., 2013. FFPred 2.0: Improved homology-independent prediction of gene ontology terms for eukaryotic protein sequences. *PLOS ONE* 8, e63754.
- Piovesan, D., Giollo, M., Leonardi, E., *et al.*, 2015. INGA: Protein function prediction combining interaction networks, domain assignments and sequence similarity. *Nucleic Acids Research* 43, W134–W140.
- Profitti, G., Martelli, P.L., Casadio, R., 2017. The Bologna Annotation Resource (BAR 3.0): Improving protein functional annotation. *Nucleic Acids Research* 45, W285–W290.
- Rost, B., 2002. Enzyme function less conserved than anticipated. *Journal of Molecular Biology* 318, 595–608.
- Sahraeian, S.M., Luo, K.R., Brenner, S.E., 2015. SIFTER search: A web server for accurate phylogeny-based protein function prediction. *Nucleic Acids Research* 43, W141–W147.
- Shindyalov, I.N., Bourne, P.E., 2001. A database and tools for 3-D protein structure comparison and alignment using the combinatorial extension (CE) algorithm. *Nucleic Acids Research* 29, 228–229.
- Sigrist, C.J., de Castro, E., Cerutti, L., *et al.*, 2013. New and continuing developments at PROSITE. *Nucleic Acids Research* 41, D344–D347.
- Sillitoe, I., Cuff, A.L., Dessailly, B.H., *et al.*, 2013. New functional families (FunFams) in CATH to improve the mapping of conserved functional sites to 3D structures. *Nucleic Acids Research* 41, D490–D498.
- Sillitoe, I., Lewis, T.E., Cuff, A., *et al.*, 2015. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research* 43, D376–D381.
- Van Landeghem, S., Hakala, K., Rönqvist, S., *et al.*, 2012. Exploring biomolecular literature with EVEX: Connecting genes through events, homology, and indirect associations. *Advanced Bioinformatics* 2012, 582765.
- Wass, M.N., Barton, G., Sternberg, M.J., 2012. CombFunc: Predicting protein function using heterogeneous data sources. *Nucleic Acids Research* 40, W466–W470.

Protein Post-Translational Modification Prediction

Chi NI Pang and Marc R Wilkins, The University of New South Wales, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Post-translational modifications (PTMs) of proteins involve the covalent modification of amino acid side chains or of the protein's N- and C-termini. There are two main types of PTMs. The first involve the enzymatic covalent addition of small molecules (e.g., phosphoryl group for phosphorylation) or polypeptide (e.g., ubiquitin for ubiquitination) to the amino acids, whereas the second involve the proteolytic cleavage of the polypeptide chain. The most common types of PTM additions include phosphorylation, acetylation, methylation, ubiquitination, and glycosylation (Fig. 1), of the more than 450 different types of PTM documented in the Uniprot Database (Bateman *et al.*, 2017).

There are many biological functions for post-translational modifications. One way in which the addition of PTM regulates the activity of a protein is through changing the conformation of the target protein, a function well-documented for phosphorylation (Lin *et al.*, 1996; Cohen, 2000). Other post-translation modifications act as a subcellular localization signal (e.g., phosphorylation, methylation, ubiquitination) (Hunter, 2007; Scott and Pawson, 2009), while fatty-acid modifications can be used to anchor a protein to membranes. Post-translational modifications can also affect the half-life of a protein. For example, ubiquitination of a protein can mark it for degradation via the ubiquitin-proteasome pathway (Ciechanover, 1994), while N-terminal modifications are well known to regulate protein stability (Varshavsky, 2011). The presence of PTMs can also affect protein folding and solubility (Tokmakov *et al.*, 2012).

In addition to the above functional roles, PTMs are known to modulate protein-protein interactions. Enzymes can act as 'writers' or 'erasers' of modifications (Beltrao *et al.*, 2013), often through transient binding with the target (Stein *et al.*, 2009). Protein interaction domains, a number of which are known, can mediate direct binding to the target proteins and act as a 'reader' for specific

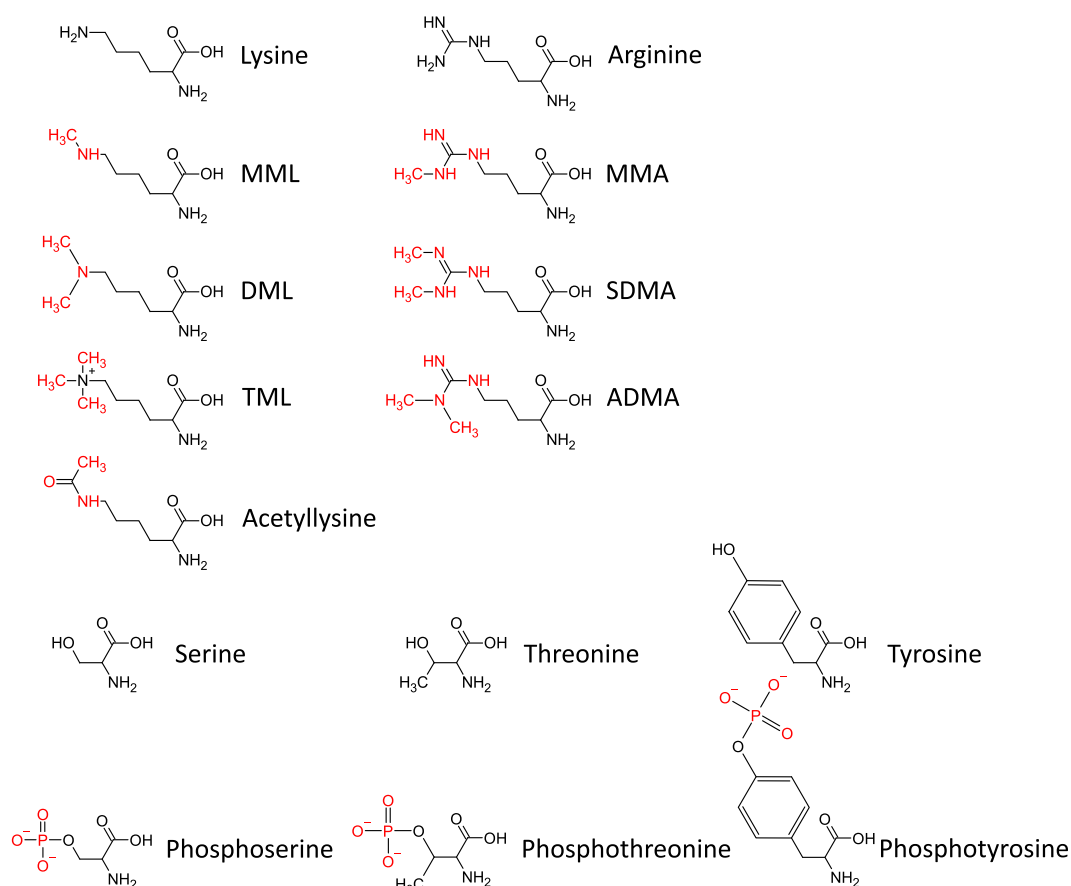


Fig. 1 Chemical structures of several common types of modified amino acids. This figure shows the chemical structures of amino acids (black) and their chemical modifications (red). These include acetylated lysine and phosphorylated serine, threonine, and tyrosine, and different types of methylated residues, including mono-methyllysine (MML), di-methyllysine (DML), tri-methyllysine (TML), mono-methylarginine (MMA), symmetrical di-methylarginine (SDMA), and asymmetrical di-methylarginine (ADMA).

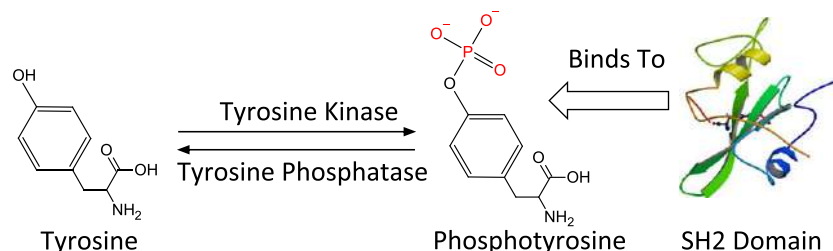


Fig. 2 Reversible tyrosine phosphorylation regulates the binding of SH2 domains to tyrosine phosphorylated peptides. A tyrosine kinase catalyzes the transfer of a phosphoryl moiety to a tyrosine residue, resulting in a phosphotyrosine residue. This reaction can be reversed by a tyrosine phosphatase. A phosphotyrosine residue in the peptide can be recognized and bound by SH2 domains (PDB: 1LKK; Tong *et al.*, 1996). The SH2 domain is shown binding to a tyrosine-phosphorylated peptide.

PTMs (reviewed in Seet *et al.*, 2006). For modifications which are reversible through the action of ‘writer’ and ‘eraser’ enzymes, the binding of the ‘reader’ to the modified protein can be dynamically regulated in response to signals upstream of the enzymes. An example of this is the binding of SH2 domains to phosphotyrosine residues, which can be reversed by a competing tyrosine phosphatase (Fig. 2; Seet *et al.*, 2006). In other cases, PTMs can modulate protein-protein interactions by changing the physiochemical properties (e.g., hydrophobicity and charge) of a binding interface (Winter *et al.*, 2014) and/or causing conformational changes that exposes the interaction surface (Venne *et al.*, 2014). Recently, systematic analysis of protein interaction networks has identified that proteins with a high number of interaction partners, also called network ‘hubs’, are often enriched for PTMs. This observation is further evidence that PTMs have important roles in regulating protein-protein interactions (Duan and Walther, 2015).

The ‘histone code’ is a well-known process in which PTMs regulate the recruitment of transcription factors and their regulatory factors to active chromatin regions (Bannister and Kouzarides, 2011). The histone code involves multiple types of PTMs that are located at different sites along the tail of histone proteins (Bannister and Kouzarides, 2011). Different combinations of PTMs, involving ‘PTM crosstalk’, lead to the binding of different transcription factors to the histones. Recently, it has been suggested that histones may not be unique in having a ‘modification code’ and other proteins, including p53 (Gu and Zhu, 2012), RNA polymerase II (Eick and Geyer, 2013) and FoxO (Calnan and Brunet, 2008) were reported to have diverse PTMs which regulate their binding with interaction partners. This led to the general hypothesis that ‘interaction codes’ may be widespread in the proteome (Winter *et al.*, 2014). Interestingly, modification enzymes may also be subjected to regulation by PTMs, which adds further complexity to the analysis of the PTM code (Venne *et al.*, 2014). Deciphering this code would require the analysis of multiple types of PTMs, knowing the specificity of modifying enzymes, and knowing the role of PTMs in modulating protein folding, stability, localization and protein-protein interactions (Prabakaran *et al.*, 2012; Venne *et al.*, 2014).

Large-scale PTM datasets (Dinkel *et al.*, 2011; Zanzoni *et al.*, 2011; Huang *et al.*, 2016; Bateman *et al.*, 2017) have been generated mainly from proteomic studies (Hornbeck *et al.*, 2015; Mínguez *et al.*, 2015). Such datasets have accelerated the development of PTM prediction tools. As for all software, vital steps for the development of these tools include defining the needs of the users, incorporating features that will address the needs, and ensuring that the tool is robust and achieves the desired goals. Recently, a comprehensive review of PTM prediction tools for 10 types of PTMs has been published elsewhere (Audagnotto and Dal Peraro, 2017). Accordingly, this article will not explore what all the existing PTM prediction tools are, but will explore the aims and hypothesis that are widely applicable to PTM predictions tools and will highlight a series of challenges involved in designing, engineering, and testing these prediction tools to achieve these aims.

What are the Criteria for Evaluating and Selecting a PTM Prediction Tool?

When evaluating PTM prediction tools, it is important to consider the number of steps that is involved in building and benchmarking the performance of the tool (Fig. 3), any of which could affect whether the tool is suitable for the need of the research project. The main steps involve the training dataset, the predictive of selection of features, the testing dataset, the machine learning algorithm, the evaluation dataset, the benchmarking approach and the performance metrics. Below, we illustrate the critical factors that influence the performance of PTM prediction tools through a number of case studies.

Case Studies on PTM Prediction Tools

Prediction of Phosphorylation Sites

PhosphoPredict

PhosphoPredict is a human kinase-specific phosphorylation site prediction tool developed by Song *et al.* (2017) (Table 1). This tool is of particular interest as it uses a combination of feature selection and protein-protein interaction to predict phosphorylation sites. The tool predicts phosphorylation sites for 12 human kinase families, including ATM, CDKs, GSK-3, MAPKs, PKA, PKB, PKC, and SRC. The

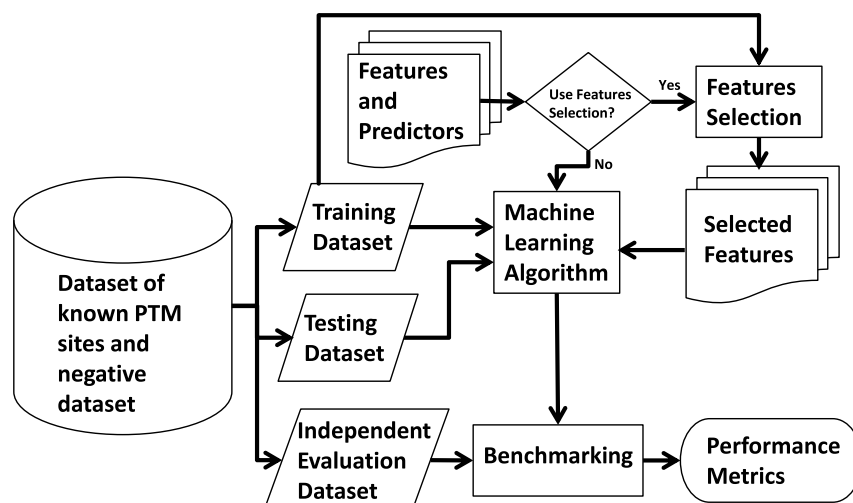


Fig. 3 Flow chart illustrating the process for developing a PTM prediction tool. Known PTM sites and sites known not be modified by the same type of PTM are collected. Datasets can be split into three, to generate a training dataset, the testing dataset, and an independent evaluation dataset. The features (also called predictors) of the model are defined (e.g., predicted surface accessibility and predicted intrinsically disordered regions). These features can be used directly for machine learning. Optionally, a subset of features which are least redundant and best describe the training dataset can be chosen using a feature selection algorithm. The training dataset, the testing dataset and the selected features can be used to train the machine learning algorithm and to develop the PTM prediction tool. The performance of the machine learning algorithm is benchmarked using the independent evaluation dataset and the results are then reported using various performance metrics (e.g., sensitivity, specificity, and accuracy).

Table 1 Case studies on post-translational modifications prediction tools

Name of tool and reference	Species specific?	Enzyme specific?	Dataset	Features ^a	Machine learning algorithm
Phosphorylation					
PhosphoPredict (Song <i>et al.</i> , 2017)	Human only	Yes, 12 kinase families	Phospho.ELM	Amino Acid Type, PSS, PSA, PIDR, Gene Ontology, Protein Domains, KEGG Pathways, Protein-Protein Interactions from STRING. Maximum Relevance Minimum Redundancy features selection	Random forest
ELM (Dinkel <i>et al.</i> , 2016)	Many organisms	Yes	Eukaryotic Linear Motif database and Phospho.ELM	Regular Expression Patterns, PSS, PSA, PIDR, and Protein Domains	Regular expression matching
Acetylation					
GPS-PAIL (Deng <i>et al.</i> , 2016)	Seven eukaryotes including Human and Yeast	Yes, seven histone acetyl-transferase (HAT)	702 non-redundant HAT-specific acetyllysine sites for 205 proteins	Sequence + / - 30 amino acids surrounding acetylation sites included	Simulated annealing to train scoring matrix and position specific weights. Optimization of motif length.
Methylation					
GPS-MSP (Deng <i>et al.</i> , 2017)	Many organisms	No, but specific to type of methylation	1521 meth-K from 962 proteins, and 3900 meth-R sites from 1751 protein	Sequence + / - 30 amino acids surrounding methylation sites included	Simulated annealing to train scoring matrix and position specific weights. Optimization of motif length.
Ubiquitination					
Nguyen <i>et al.</i> (2016)	Many organisms	No, but use clustering to predict putative motifs	2566 ubiquitination sites	Amino acid composition, amino acid pairwise composition, and PSSM	Support Vector Machine, Maximal Dependence Decomposition

^aPSS: Predicted Secondary Structure; PSA: Predicted Solvent Accessibility; PIDR: Predicted Intrinsically-Disordered Region; PSSM: Position-specific scoring matrix.

human phosphorylation sites and their kinase-specificity were obtained from Phospho.ELM (Dinkel *et al.*, 2011). A wide-range of predictive features are utilized, which includes those that are sequence-based, including amino acid type, predicted secondary structure (Wagner *et al.*, 2005), predicted solvent accessibility (Wagner *et al.*, 2005) and predicted intrinsic-disorder (Ward *et al.*, 2004). In addition, functional features were also used for predictions, including GO annotations (Carbon *et al.*, 2017) and protein domains of the query substrate protein (Finn *et al.*, 2016, 2017), KEGG biochemical pathways (Kanehisa *et al.*, 2016) and protein-protein Interactions (Szkarczyk *et al.*, 2017). These functional features are represented as categorical labels with one feature for each category.

Song *et al.* (2017) used the feature selection tool called maximum Relevance Minimum Redundancy (mRMR) (Peng *et al.*, 2005) to identify the most important subset of features for each kinase, which led to increased prediction performance. These features were used in the Random Forest (Breiman, 2001; Liaw and Wiener, 2002) machine learning algorithm, trained using five-fold cross-validation. Independent datasets from PhosphoSitePlus (Hornbeck *et al.*, 2015) and Uniprot (Bateman *et al.*, 2017) were used to verify that the tools were not affected by overfitting. The benchmarking resulted in high specificity (88.3%–96.7%) but variable sensitivity (29.2%–80.5%) for different kinases. PhosphoPredict was found to have better or similar performance as compared to other tools, including KinasePhos (Huang *et al.*, 2005), PPSP (Xue *et al.*, 2006), GPS (Xue *et al.*, 2006), and Musite (Xue *et al.*, 2006). Together, this tool highlights how a careful and well-informed design, that considers many aspects of PTMs, can lead to high quality predictions.

Eukaryotic linear motif (ELM)

The ELM database contains kinase-specific phosphorylation motifs represented as regular expression patterns, which enable users to predict the presence of kinase-specific phosphorylation sites within query protein sequences (Table 1) (Gouw *et al.*, 2017). The ELM database currently has 20 kinase-specific phosphorylation motifs and 12 kinase binding motifs and has recently been updated to include new motifs for CDKs, MAPKs and Plks (Gouw *et al.*, 2017). For a user-defined sequence, the tool can identify matches to entries in the Phospho.ELM database (Dinkel *et al.*, 2011), including experimentally identified phosphorylation sites and their kinase specificity if known. Users can also search the companion Switches.ELM database, which contain a list of sites in which the presence or absence of phosphorylation is known to affect protein-protein interaction (Van Roey *et al.*, 2013). In addition to kinase-specific phosphorylation sites, other types of modifications with known conserved motifs in ELM includes N-glycosylation, N-myristoylation, O-fucose modification, O-glucose modification, S-palmitoylation, sumoylation, and enzyme-specific peptide cleavage sites. The ELM database also contains annotations for conserved motif regions, also called Short Linear Motifs (SLiMs). SLiMs are compact linear sequences on average six amino acids in length, preferentially found in intrinsically disordered regions and involved in PTMs, molecular targeting or protein-protein interaction interfaces (Davey *et al.*, 2012).

In ELM, users can apply filters to increase the specificity of their motif searches. These include filters for motifs present in specific taxonomic groups or cell compartments. The probability filter is based on a score calculated from multiplying the probabilities of occurrence of each amino acid within the regular expression pattern. This filter can be used to reduce matches to degenerate motifs. The globular domain filters identify structurally ordered domains, which include predicted protein domains from Pfam (Finn *et al.*, 2016) and SMART (Letunic and Bork, 2017) and predicted globular regions from GlobPlot (Letunic and Bork, 2017). Via *et al.* (2009) showed the utility of structural filters for phosphorylation, which can be used to prioritize matches to disordered regions. Predicted secondary structures from SMART are used as a proxy for surface accessible loops. The disordered region filters include GlobPlot and IUPred (Linding *et al.*, 2003). Unlike many other PTM prediction tools, which rely on machine learning algorithms whose predictive mechanisms are opaque, the ELM database provides clear information on why a site was predicted. The predictions made by ELM are based on known kinase-specific phosphorylation motifs and sequence and structural features. These can be visualized as sequence tracks, which enables users to inspect the structural environment, motifs or other PTMs surrounding the phosphorylation site, thus providing a rich functional and structural context.

Prediction of Acetylation Sites

GPS-PAIL

The GPS-PAIL tool (Deng *et al.*, 2016) predicts acetylation sites that are specific to seven histone acetyltransferases (HATs): CREBBP, EP300, HAT1, KAT2A, KAT2B, KAT5 and KAT8 (Table 1). Few tools can predict acetylation sites specific to single acetyltransferases (Li *et al.*, 2012; Wang *et al.*, 2012; Deng *et al.*, 2016) and thus GPS-PAIL is of particular interest. Since not all acetyltransferases are present ubiquitously, Deng *et al.* (2016) also checked for the presence of each acetyltransferase in the proteomes of seven eukaryotic species. The training dataset included 702 experimentally determined acetyltransferase-specific acetylation sites for 205 proteins from seven eukaryotes, with 544 sites for 160 human proteins. The method uses the BLOSUM62 amino acid substitution matrix to compare a query site with known enzyme-specific acetylation sites; predicted acetylation sites have a high score. Simulated annealing was used to train position-specific weights and to mutate the BLOSUM62 matrix to achieve better predictions. This is a complementary method to other commonly used machine learning tools, requiring a problem and domain-specific ‘coding scheme’ to describe the search space (Gouw *et al.*, 2017); in this case it was the BLOSUM62 matrix and the position specific weights. The use of simulated annealing is notable as it is an active research area. Optimal motif length in GPS-PAIL was identified through an exhaustive search. Analysis of the sequence motifs surrounding acetylation sites of each acetyltransferase highlighted their sequence specificity, with K, R, G and A residues most often enriched in these motifs.

Performance evaluation of GPS-PAIL was carried out through leave-one-out cross-validation and N-fold cross-validation (Deng *et al.*, 2016). Among the seven acetyltransferases, the tool achieved a high specificity of 80%–95% but considerably lower

sensitivity of 38%–57.5%. Of the 10,505 experimentally determined acetylation sites from 4488 human proteins, the tool predicted the acetyltransferase responsible for 1492 sites (14.2%) and 913 proteins (20.34%), consistent with low sensitivity. GPS-PAIL was found to be more efficient for predicting acetyltransferase-specific sites for mammalian and plants as compared to other proteomes. The majority of acetylation sites were found to be targeted by one acetyltransferase only, while no acetylation sites were the target of all seven acetyltransferases, which suggests acetylation sites tend to be acetyltransferase-specific.

Prediction of Methylation Sites

GPS-MSP

Methyltransferase-specific protein methylation motifs have been described (e.g., [Hamey et al., 2016](#)) however, to the best of our knowledge, there is currently a lack of prediction tools for methyltransferase-specific protein methylation sites. Thus the tools predict sites, in a general sense, but not the possible methyltransferases that may be responsible. [Deng et al. \(2017\)](#) developed a suite of prediction tools (GPS-MSP) specific for different types of protein methylation, including mono-, di-, and tri-methyllysine, and mono-, symmetric di-, and asymmetric di-methylarginine. Separate prediction models for each type of methylation were shown to improve prediction accuracy. The same algorithm as GPS-PAIL (above) was used for GPS-MSP. The training dataset contained 1521 methyllysines from 962 proteins, and 3900 methylarginine sites from 1751 proteins. For different types of methylation sites, leave-one-out cross-validation and receiver operating characteristic analyses showed that the sensitivity ranged from ~35% to 50% at a fixed specificity of 90%. Analysis of the sequence motif surrounding the arginine methylation sites identified the 'RGG' motif, confirming previous reports ([Lischwe et al., 1985](#); [Kim et al., 1997](#)), while analysis of lysine methylation sites identified putative new sequence motifs, including the overrepresentation of leucine at the –1 position of mono-methyllysine. This method shows how in the absence of enzyme-specific data, different types of methylation could be used as a proxy for enzyme-specificity to enhance prediction performance.

Prediction of ubiquitination sites

Protein ubiquitination is catalyzed by the E1 activating enzyme, the E2 conjugating enzyme and the E3 ubiquitin ligase that mediates substrate specificity ([Amoutzias et al., 2012](#)). There are > 600 E3 ligases in human yet there is currently little information on the specificity of each E3 ligase ([Amoutzias et al., 2012](#)). Therefore, current prediction tools are based on global prediction of ubiquitination sites, as reviewed by [Chen et al. \(2014\)](#). [Nguyen et al. \(2015\)](#) used unsupervised clustering of the ubiquitination sites and developed a prediction model for each cluster. Analysis of sequence motifs for each of the clusters showed that each have varying sequence specificity, suggesting that clustering and motif analysis could be used to identify putative E3 ligase recognition motifs ([Nguyen et al., 2015](#)). [Nguyen et al. \(2017\)](#) later improved their prediction model by incorporating additional features, including amino acid composition, amino acid pairwise composition and a position-specific scoring matrix, into support vector machines (SVM). The SVM model for each cluster was combined into an ensemble model. This resulted in sensitivity of 67.6% and specificity of 58.8% from five-fold cross validation. Similar to protein methyltransferases and methylation, the E3 ligase-specificity for most ubiquitination sites are poorly defined. [Nguyen et al. \(2017\)](#) demonstrated that sequence-based clustering of ubiquitination sites, prior to training, could act as a viable proxy in the absence of enzyme-specific motifs. It is interesting to note that the same research group has also developed the UbiNet database of protein-protein interaction networks focusing on the interaction partners of the E1, E2 and E3 enzymes ([Nguyen et al., 2016](#)). This ubiquitination enzyme protein-protein interaction database, in conjunction with their prediction algorithm, could be used to explore the relationship between specific E3 ligases and their potential substrates.

What are the Biological Questions Explored Through the Prediction of PTMs?

As noted above, prediction of post-translational modifications involves the use of simple motif-driven methods, through to machine learning and/or artificial intelligence approaches, to identify and also characterize PTMs in the proteome. Whilst it is possible to generate a list of predicted PTMs, a major challenge is to analyze the list of predicted modification sites to generate valuable biological insights. Here, we will highlight three main biological questions that potential users of PTM prediction tools may like to address and provide rationale on why these questions are important. (1) Where are the modification site(s) and how do these relate to the structure and function of the protein? (2) What are the modification enzyme(s) responsible for writing or erasing a PTM at a specific amino acid within a protein? (3) What is the functional role of the PTM in the proteome? These three questions are complementary and depend on whether the analysis is focused on the modification enzymes, the protein substrates, or the specific type of modification.

Where are the Modification Site(s) and How do These Relate to the Structure and Function of the Protein?

After obtaining a list of modification sites from a PTM prediction tool, one subsequent step can involve mapping modifications to the structural features of the target protein. This helps understand how the modifications on a specific protein may affect its function. These analyses often involve the use of information from three-dimensional protein structures, along with predicted structural properties (e.g., domains, disordered regions) from protein sequence analysis tools.

When the three-dimensional structure of a modified protein is available, the modification site can be mapped onto structure. This can provide accurate information on surface accessibility of the modification site (Pang *et al.*, 2007) and, for enzymes, whether the modification is in the vicinity of active sites and may affect catalytic activity (Zanzoni *et al.*, 2011; Craveur *et al.*, 2014). Molecular dynamic simulations of modified proteins can be used to predict whether the modifications may cause conformational change (reviewed in Audagnotto and Dal Peraro, 2017).

Structures of protein complexes can provide information on whether modification sites are in the vicinity of protein-DNA, protein-RNA and protein-protein interaction interfaces and therefore have the potential to modulate interactions (Beltrao *et al.*, 2012). There are large variety of known protein-protein interfaces (Winter, 2006; Zhao *et al.*, 2011; Garma *et al.*, 2012) and these include two main types, domain-domain and domain-motif interactions, which have been well characterized in the literature (Stein and Aloy, 2010; Mosca *et al.*, 2014). Domain-domain interactions are often more stable interactions while domain-motif interactions tend to mediate transient interactions (Mosca *et al.*, 2014). The 3DID database systematically curates domain-domain or domain-motif interaction interfaces from known 3-D structures of protein complexes (Mosca *et al.*, 2014). Using the 3DID database, it is possible to determine whether PTM sites are present at or adjacent to the interface residues of an interaction domain or motif. This can help understand whether a PTM could possibly mediate or block a protein-protein interaction, and thus act as a protein interaction switch (Akiva *et al.*, 2012; Beltrao *et al.*, 2012; Van Roey *et al.*, 2012).

Modifications on critical sites of a protein can have important effects on signaling. A well-known example involves receptor tyrosine kinases where, upon ligand binding, there is autophosphorylation which leads to conformational changes and homo-dimerization (Ullrich and Schlessinger, 1990). These autophosphorylation sites have a variety of known functions, which include maintaining kinase activity in the absence of bound ligand by disrupting autoinhibitory interaction (Rosen *et al.*, 1983; Lemmon and Schlessinger, 2010) and modulating interaction with specific downstream substrates as part of signal transduction (Kazlauskas and Cooper, 1989; Arvidsson *et al.*, 1994; Hanke and Mann, 2009). Analysis of signaling proteins, such as receptor tyrosine kinases, may involve a multi-faceted and integrative approach, including but not limited to the application of tools and concepts discussed above.

What are the Enzyme(s) Responsible for Writing or Erasing a PTM on a Specific Amino Acid Within a Protein?

This question is applicable to PTMs that can be catalyzed by multiple enzymes. This enquiry is suited to researchers who are interested in the regulatory role of modifying enzymes and their effect on one target protein of interest. Knowing the specific upstream modifying enzymes would then, in turn, enable the researcher to perform further investigative experiments. This may involve verifying the activity of an enzyme through the absence or decreased abundance of the modification site upon perturbation of the enzyme through knockout (CRISPR) (Hamey *et al.*, 2017), knockdown (RNA inhibition) (Weiss *et al.*, 2007), or targeting with an inhibitor (Weiss *et al.*, 2007). Identifying the enzyme responsible for the modification site also enables us to contextualize the modified protein within an enzyme-substrate interaction network (Linding *et al.*, 2008; Erce *et al.*, 2012). This enables researchers to analyze the function of the modification enzyme, and one or more substrates, as a group, including identifying enriched canonical pathways, biological functions, sub-cellular localization and coordinated changes in PTM abundance under different cellular contexts (Linding *et al.*, 2008; Yang *et al.*, 2015).

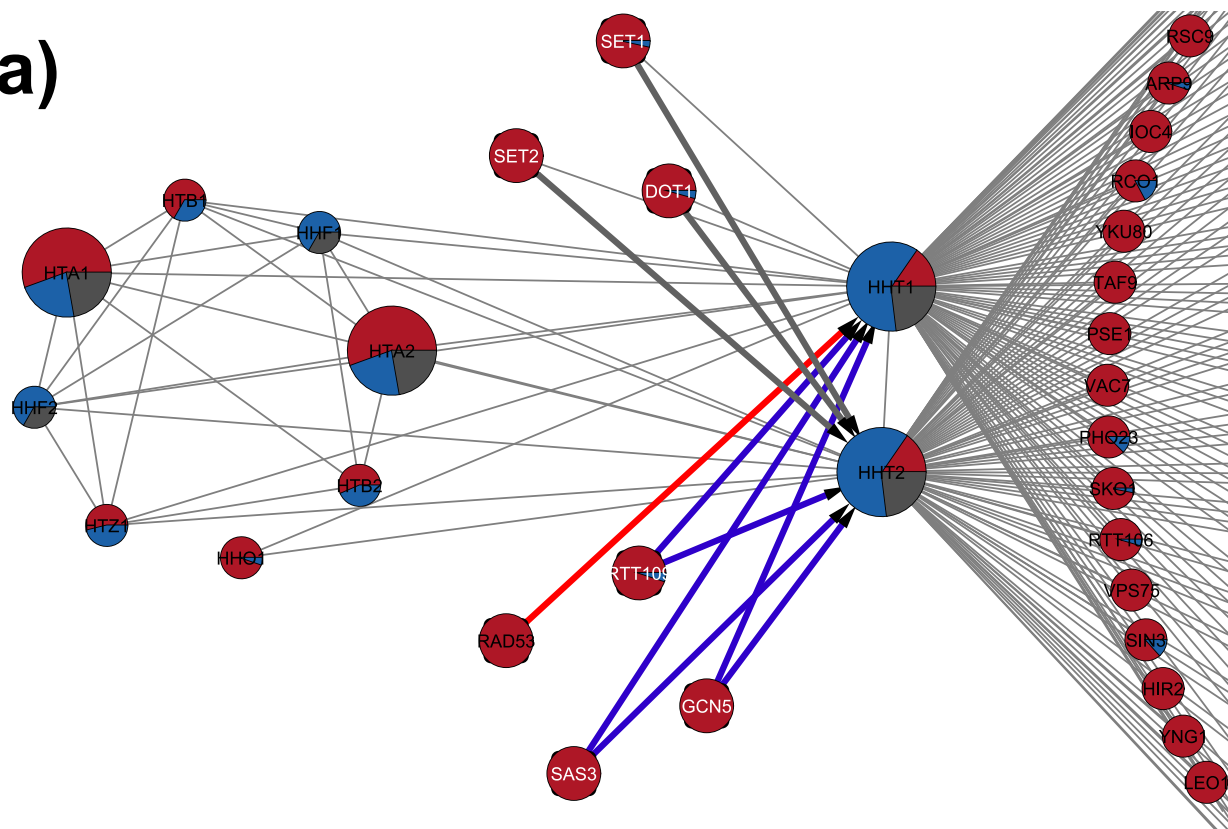
What is the Functional Role of a PTM in the Proteome?

It is important to gain a general perspective on the function of each type of PTM in the proteome. Functional enrichment analysis of proteins which contain a specific PTM type might help to highlight the role of a PTM within the cell, including the biological pathways that are overrepresented in the modified proteins. These analyses could be performed using tools for Gene Ontology (GO) enrichment (Maere *et al.*, 2005; Grossmann *et al.*, 2007; Bindea *et al.*, 2009; Eden *et al.*, 2009; Reimand *et al.*, 2016; Mi *et al.*, 2017) and Pathway Analysis (Krämer *et al.*, 2014; Fabregat *et al.*, 2016). Different types of PTMs are associated with diverse biological functions. The best known examples include serine, threonine and tyrosine phosphorylation, which are often involved in signaling (Linding *et al.*, 2008). By contrast, lysine acetylation is frequently observed on proteins involved in energy metabolism (Choudhary *et al.*, 2014), arginine methylation is associated with RNA process and transport (Bedford and Clarke, 2009; Blanc and phane Richard, 2017), lysine methylation is enriched among protein translation machineries (Lanouette *et al.*, 2014) and fatty acid modifications are frequently present on membrane-bound proteins and receptors (Schwenk *et al.*, 2010). Among eukaryotes, protein glycosylation is associated with secreted proteins, proteins with extracellular domains and protein quality control (Benham, 2012; Xu and Ng, 2015). Interestingly, recent studies have also shown that glycosylation is also present among bacterial species (Szymanski and Wren, 2005; Lu *et al.*, 2015). This bacterial glycosylation has been suggested to modulate the function of virulence factors or factors involved in hijacking the host cellular machinery (Szymanski and Wren, 2005; Lu *et al.*, 2015).

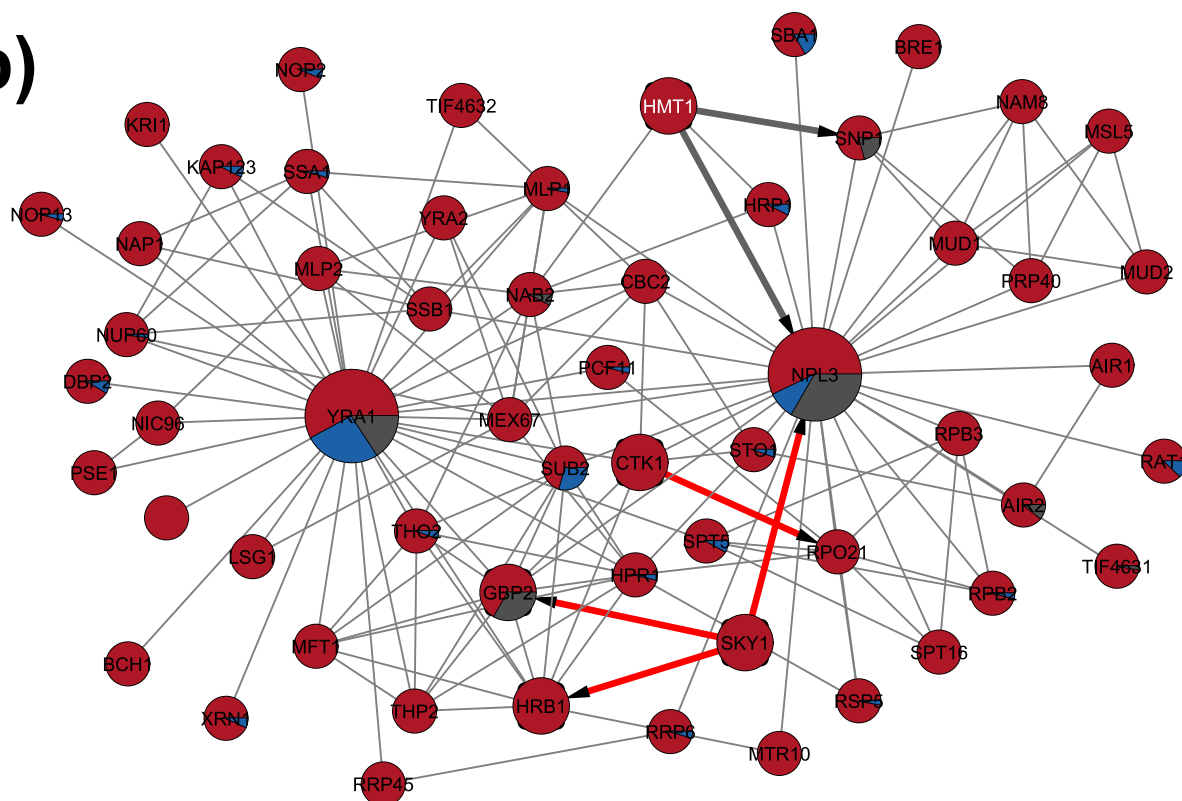
Results and Discussion

Machine learning tools have a capacity to provide accurate predictions if the approach used to train, test and evaluate the tools are well-designed and the datasets are of high quality. However it must be acknowledged that PTM data remains sparse, will contain

a)



b)



false positives and also false negatives. Accordingly, it is important to highlight the challenges involved in dataset curation and how these could affect the performance of the prediction tools. Major factors to consider include the enzymes responsible for a modification site, and the size, quality, composition of the modification datasets.

Incorporating the site specificity of PTM enzymes is likely to improve the performance of predictive tools. Modifications known to arise from specific enzymes can be analyzed for particular sequence motifs using tools such as pLogo (O'Shea *et al.*, 2013). The existence of a sequence motif is a good indication that enzyme-specific prediction would improve prediction accuracy, and conversely that using all available data without considering enzyme-specificity in such cases would lead to worse prediction performance. As described above, enzyme-specific prediction improved predictions of phosphorylation (Song *et al.*, 2017) and acetylation (Deng *et al.*, 2016) sites. A downstream benefit of enzyme-specific prediction is the ability to better understand the specific function of each PTM-catalyzing enzyme. For example, GO enrichment analysis of kinase-specific substrates predicted by PhosphoPredict show that while all kinase families were significantly enriched among diverse biological pathways, channel activity and ion channel activity were significantly overrepresented among MAPK only.

The taxonomic relevance of each enzyme should be considered when using enzyme-specific PTM prediction. For example, *S. cerevisiae* has three relevant acetyltransferases (HAT1, KAT2A and KAT6), and therefore predictions made by GPS-PAIL for other acetyltransferases should not be used for yeast (Deng *et al.*, 2016). However, in cases where a PTM is highly conserved across cellular organisms, such as N-glycosylation (Deshpande *et al.*, 2008; Aebi, 2013; Jarrell *et al.*, 2014; Pinho and Reis, 2015; Strasser, 2016), many enzymes may be conserved and taxonomic filters may be less useful (Beltrao *et al.*, 2013). Most enzyme-specific PTM data are focused on human and a few model organisms such as yeast. Databases that specialize in one type of PTM often include annotation for enzyme specificity. For example, both Phospho.ELM (Dinkel *et al.*, 2011) and PhosphoSitePlus (Hornbeck *et al.*, 2015) curates phosphorylation sites and include annotation for kinase-specific sites.

An adequate number of PTM sites in the training datasets are necessary for the machine learning algorithms to detect reliable patterns from the data and to use these patterns for predictions. However, it is difficult to know what actually represents an adequate number. It is important for tools to report the number of positive modification sites and negative sites considered, such that users can evaluate tools based on the size of the datasets. However there remains a strong possibility of false negatives being present and used. False negatives could be present due to a number of reasons, including difficulties in detecting low abundance proteins (Picotti *et al.*, 2009) or low stoichiometry modification sites (Amoutzias *et al.*, 2012) and challenges in the analysis of some membrane proteins, which are difficult to solubilize for analysis with mass spectrometry (Barrera and Robinson, 2011). To increase the coverage of the training datasets and to mitigate the effect of false negatives, data can be combined from multiple sources, including publically available PTM databases and those curated from literature. There are a number of publically available PTM databases, including Uniprot (Bateman *et al.*, 2017) and dbPTM (Huang *et al.*, 2016), which could be used for building collections of modification sites. However, there will typically be inconsistencies between different large-scale PTM resources which are difficult to reconcile.

Similar to false negatives, there are likely to be false positive modifications in experimental data and, when used for a tool's training set, these could lead to decreased prediction accuracy and specificity. This problem could be mitigated by filtering experimental data with a stringent false discovery rate (FDR), or selecting only modification sites which are verified by two or more independent experiments (Amoutzias *et al.*, 2012). Recently, Hart-Smith *et al.* (2016) showed that large-scale mass spectrometry analysis of protein methylation sites are subjected to high FDR. The reason is that many types of amino acid substitutions are isobaric to the mass change associated with a methyl group, and in many cases, it is not possible to distinguish between them. The study also highlighted that a target-decoy approach is ineffective for controlling for the FDR of protein methylation sites. Heavy methyl-group labelling through Stable Isotope Labelling with Amino Acid in Cell Culture (SILAC) was suggested as the validation criteria required for methylation sites identification. The MethylQuant tool (Tay *et al.*, 2017a,b) was recently developed to support heavy-methyl SILAC analyses. The above highlights the importance of using high quality experimental datasets, such as those that uses SILAC for methylation site identification, to avoid inclusion of false positive sites.

The ideal training and testing datasets should contain an equal proportion of positive and negative PTM site data. Yet there are often more unmodified than modified sites. One common solution is to train multiple machine learning models, e.g. an ensemble approach, each time using a random subset of the negative sites and averaging results from many models to increase prediction performance (Yang *et al.*, 2015). A further problem with modification datasets is the presence of inspection bias. For example,

Fig. 4 Visualization of modified proteins, protein-protein interactions, and enzyme-substrate interactions in *S. cerevisiae* using the PTMOracle app in Cytoscape. Each protein is represented as a node in the network. The pie chart for each protein represents the proportion of known modification sites that are phosphorylation (red), acetylation (blue) and methylation (grey). Protein-protein interactions are represented by thin grey lines, kinase-substrate interactions are represented by thick red arrows, acetyltransferase-substrate interactions are represented by thick blue arrows and methyltransferase-substrate interactions are represented by thick grey arrows. All enzyme-substrate interactions are directional, with the arrow indicating the substrate. (a) The interactions between histone proteins are shown in the network. The network focuses on the enzyme-substrate interactions that target histone proteins Hht1p and Hht2p (large nodes), and a subset of non-histone interaction partners for these two histone proteins are also shown. Hht1p is phosphorylated, acetylated and methylated, and the enzymes which catalyze these modifications are known and shown. Unlike Hht1p, the enzymes which are responsible for the phosphorylation, acetylation and methylation of Hht2p have not yet been experimentally identified. The methyltransferases Set1p, Set2p, and Dot1p form protein-protein interactions with Hht1p, predicting that these methyltransferases catalyze Hht1p methylation. (b) The protein-protein interaction network focusing on Npl3p and Yra1p (large nodes). The enzymes responsible for Npl3p methylation and phosphorylation are known and shown (arrows). The methyltransferase Hmt1p interacts with the methylated protein Nab2p, suggesting that it is responsible for Nab2p methylation. In contrast, the methyltransferase responsible for the methylation of Yra1p has not been confirmed, and Yra1p is not known to bind any existing protein modifying enzymes.

histone protein is the most extensively analyzed protein for protein methylation sites (Lothrop *et al.*, 2013) and there will likely be a high number of histone proteins in any methylation sites training set. This may potentially lead to bias. This could be resolved by identifying redundant protein sequences and redundant modification sites from the dataset, using tools such as CD-HIT (Huang *et al.*, 2010) and keeping only one representative example from each group of homologous sites (for example, Deng *et al.*, 2017; Nguyen *et al.*, 2017).

Future Directions

There remains scope for the improvement of PTM site prediction. Since functionally important modification sites are more likely to be evolutionarily conserved, others have used this as a means to rank and prioritize predicted PTM sites. Beltrao *et al.* (2012) developed the PTMfunc database which predicts the functional relevance of phosphorylation, acetylation and ubiquitination sites for 11 eukaryotic organisms. The PTM sites that are evolutionarily conserved are more likely to be biologically important. Users of PTM prediction tools could query the PTMcode database developed by Minguéz *et al.* (2012) to identify single or pairs of co-evolving sites that are within the same domain or motifs sequences, thus highlighting putative important residues within domains or motifs, or co-evolving sites between a interacting proteins. This would help predict putative sites of greatest functional importance.

Downstream, it is important to decipher how PTMs regulate the dynamics of biological networks and participate in protein ‘interaction codes’ (Winter *et al.*, 2014). Bioinformatic tools are emerging that enable users to investigate cross-talk between PTMs, and explore enzyme-substrate relationships in the context of the interactome. Tay *et al.* (2017a,b) developed the PTMOracle app for the Cytoscape network software platform. This tool enables users to generate pie charts for each node in the network to represent the proportion of different types of PTMs on the protein (Fig. 4). More importantly, the tool can co-map and co-analyze PTMs, enzyme-substrate relationships along with other protein-protein interactions, and to generate and visualize a series of complex queries involving protein structural data, domains, motifs and interactions. Whilst the PTMOracle tool currently only uses known PTM sites in its underlying databases, predicted PTMs could also be used to enrich the networks generated and to help develop new hypotheses about PTM and protein function.

Closing Remarks

In conclusion, this article has highlighted several critical factors for developing robust and reliable PTM prediction tools. These factors include careful curation of PTM datasets, the use of enzyme-specific modification sites for tool training datasets, and the use of feature selection tools to identify non-redundant sets of features. We also discussed the biological questions that could be explored when analyzing a list of predicted PTMs. These questions include how a PTM relates to the structure and functions of the protein, the effect of PTMs on protein-protein interactions, the binding specificity of the modification enzymes and the biological processes that they regulate, and the function of different types of PTMs. It is hoped that these factors, together, will inspire the development of novel bioinformatics tools, to decipher the function of PTMs and to illuminate the roles of PTMs in protein-protein ‘interaction codes’.

Acknowledgements

MRW acknowledges support from NCRIS, administered by Bioplatforms Australia, and from the New South Wales State Government RAAP Scheme. We thank Aidan Tay for kindly providing the figure that illustrates PTMOracle and Daniel Winter for kindly providing the figures that illustrate the chemical structures of modified amino acids.

See also: Algorithms for Strings and Sequences: Searching Motifs. Algorithms for Structure Comparison and Analysis: Docking. Computational Approaches for Modeling Signal Transduction Networks. Data Mining: Classification and Prediction. Data Mining: Prediction Methods. Functional Genomics. Gene Prioritization Tools. Identification of Sequence Patterns, Motifs and Domains. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein Localization. Proteomics Mass Spectrometry Data Analysis Tools. The Evolution of Protein Family Databases. The Evolution of Protein Family Databases

References

- Aebi, M., 2013. N-linked protein glycosylation in the ER. *Biochimica et Biophysica Acta (BBA) – Molecular Cell Research* 1833 (11), 2430–2437. doi:10.1016/j.bbamcr.2013.04.001.
- Akiva, E., *et al.*, 2012. A dynamic view of domain-motif interactions. *PLOS Computational Biology* 8 (1), doi:10.1371/journal.pcbi.1002341.
- Amoutzias, G.D., *et al.*, 2012. Evaluation and properties of the budding yeast phosphoproteome. *Molecular & Cellular Proteomics: MCP* 11 (6), doi:10.1074/mcp.M111.009555.
- Arvidsson, A.K., *et al.*, 1994. Tyr-716 in the platelet-derived growth factor beta-receptor kinase insert is involved in GRB2 binding and Ras activation. *Molecular and Cellular Biology* 14 (10), 6715–6726. <https://doi.org/10.1128/MCB.14.10.6715>.
- Audagnotto, M., Dal Peraro, M., 2017. Protein post-translational modifications: In silico prediction tools and molecular modeling. *Computational and Structural Biotechnology Journal* 15, 307–319. doi:10.1016/j.csbj.2017.03.004.

- Bannister, A.J., Kouzarides, T., 2011. Regulation of chromatin by histone modifications. *Cell Research* 21 (3), 381–395. doi:10.1038/cr.2011.22.
- Barrera, N.P., Robinson, C.V., 2011. Advances in the mass spectrometry of membrane proteins: From individual proteins to intact complexes. *Annual Review of Biochemistry* 80, 247–271. doi:10.1146/annurev-biochem-062309-093307.
- Bateman, A., et al., 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169. doi:10.1093/nar/gkw1099.
- Bedford, M.T., Clarke, S.G., 2009. Protein arginine methylation in mammals: Who, what, and why. *Molecular Cell* 33 (1), 1–13. https://doi.org/10.1016/j.molcel.2008.12.013.
- Beltrao, P., et al., 2012. Systematic functional prioritization of protein posttranslational modifications. *Cell* 150 (2), 413–425. doi:10.1016/j.cell.2012.05.036.
- Beltrao, P., et al., 2013. Evolution and functional cross-talk of protein post-translational modifications. *Molecular Systems Biology* 9 (1), 1–13. doi:10.1002/msb.201304521.
- Benham, A.M., 2012. Protein secretion and the endoplasmic reticulum. *Cold Spring Harbor Perspectives in Biology* 4 (8), a012872. doi:10.1101/cshperspect.a012872.
- Bindea, G., et al., 2009. ClueGO: A cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics* 25 (8), 1091–1093. doi:10.1093/bioinformatics/btp101.
- Blanc, R.S., phane Richard, S., 2017. Arginine methylation: The coming of age. *Molecular Cell* 65, 8–24. doi:10.1016/j.molcel.2016.11.003.
- Breiman, L., 2001. Random forests. *Machine Learning* 45 (1), 5–32. doi:10.1023/A:1010933404324.
- Calnan, D.R., Brunet, A., 2008. The FoxO code. *Oncogene* 27 (16), 2276–2288. doi:10.1038/onc.2008.21.
- Carbon, S., et al., 2017. Expansion of the gene ontology knowledgebase and resources: The gene ontology consortium. *Nucleic Acids Research* 45 (D1), D331–D338. doi:10.1093/nar/gkw1108.
- Chen, Z., et al., 2014. Towards more accurate prediction of ubiquitination sites: A comprehensive review of current methods, tools and features. *Briefings in Bioinformatics* 16 (4), 640–657. doi:10.1093/bib/bbu031.
- Choudhary, C., et al., 2014. The growing landscape of lysine acetylation links metabolism and cell signalling. *Nature Reviews Molecular Cell Biology* 15 (8), 536–550. doi:10.1038/nrm3841.
- Ciechanover, A., 1994. The ubiquitin-proteasome proteolytic pathway. *Cell* 79 (1), 13–21. doi:10.1016/0092-8674(94)90396-4.
- Cohen, P., 2000. The regulation of protein function by multisite phosphorylation – A 25 year update. *Trends in Biochemical Sciences* 25 (12), 596–601. doi:10.1016/S0968-0004(00)01712-6.
- Craveur, P., Rebehmed, J., De Brevern, A.G., 2014. PTM-SD: A database of structurally resolved and annotated posttranslational modifications in proteins. *Database* 2014, 1–9. doi:10.1093/database/bau041.
- Davey, N.E., et al., 2012. Attributes of short linear motifs. *Molecular BioSystems* 8 (1), 268–281. doi:10.1039/C1MB05231D.
- Deng, W., et al., 2016. GPS-PAIL: Prediction of lysine acetyltransferase-specific modification sites from protein sequences. *Scientific Reports* 6 (1), 39787. doi:10.1038/srep39787.
- Deng, W., et al., 2017. Computational prediction of methylation types of covalently modified lysine and arginine residues in proteins. *Briefings in Bioinformatics* 18 (4), 647–658. doi:10.1093/bib/bbw041.
- Deshpande, N., et al., 2008. Protein glycosylation pathways in filamentous fungi. *Glycobiology* 18 (8), 626–637. doi:10.1093/glycob/cwn044.
- Dinkel, H., et al., 2011. Phospho.ELM: A database of phosphorylation sites-update 2011. *Nucleic Acids Research* 39 (Database issue), D261–D267. doi:10.1093/nar/gkq1104.
- Dinkel, H., et al., 2016. ELM 2016–data update and new functionality of the eukaryotic linear motif resource. *Nucleic Acids Research* 44 (D1), D294–300. https://doi.org/10.1093/nar/gkv1291.
- Duan, G., Walther, D., 2015. The roles of post-translational modifications in the context of protein interaction networks. *PLOS Computational Biology* 11 (2), 1–23. doi:10.1371/journal.pcbi.1004049.
- Eden, E., et al., 2009. GOrilla: A tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics* 10 (1), 48. doi:10.1186/1471-2105-10-48.
- Eick, D., Geyer, M., 2013. The RNA polymerase II carboxy-terminal domain (CTD) code. *Chemical Reviews* 113 (11), 8456–8490. doi:10.1021/cr400071f.
- Erce, M.A., et al., 2012. The methylproteome and the intracellular methylation network. *Proteomics* 12 (4–5), 564–586. doi:10.1002/pmic.201100397.
- Fabregat, A., et al., 2016. The reactome pathway knowledgebase. *Nucleic Acids Research* 44 (D1), D481–D487. doi:10.1093/nar/gkv1351.
- Finn, R.D., et al., 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research* 44 (D1), D279–D285. doi:10.1093/nar/gkv1344.
- Finn, R.D., et al., 2017. InterPro in 2017 – Beyond protein family and domain annotations. *Nucleic Acids Research* 45, D190–D199. doi:10.1093/nar/gkw1107.
- Gamma, L., et al., 2012. How many protein-protein interactions types exist in nature? *PLOS ONE* 7 (6), e38913. doi:10.1371/journal.pone.0038913.
- Gouw, M., et al., 2017. The eukaryotic linear motif resource – 2018 update. *Nucleic Acids Research*. doi:10.1093/nar/gkx1077.
- Grossmann, S., et al., 2007. Improved detection of overrepresentation of gene ontology annotations with parent-child analysis. *Bioinformatics* 23 (22), 3024–3031. doi:10.1093/bioinformatics/btm440.
- Gu, B., Zhu, W.G., 2012. Surf the post-translational modification network of p53 regulation. *International Journal of Biological Sciences* 8 (5), 672–684. doi:10.7150/ijbs.4283.
- Hamey, J.J., et al., 2016. The activity of a yeast Family 16 methyltransferase, Efm2, is affected by a conserved tryptophan and its N-terminal region. *FEBS Open Bio* 6 (12), 1320–1330. doi:10.1002/2211-5463.12153.
- Hamey, J.J., et al., 2017. METTL21B is a novel human lysine methyltransferase of translation elongation factor 1A: Discovery by CRISPR/Cas9 knock out. *Molecular & Cellular Proteomics: MCP*. doi:10.1074/mcp.M116.066308. pii, p. mcp.M116.066308.
- Hanke, S., Mann, M., 2009. The phosphotyrosine interactome of the insulin receptor family and its substrates IRS-1 and IRS-2. *Molecular & Cellular Proteomics* 8 (3), 519–534. doi:10.1074/mcp.M800407-MCP200.
- Hart-Smith, G., et al., 2016. Large scale mass spectrometry-based identifications of enzyme-mediated protein methylation are subject to high false discovery rates. *Molecular & Cellular Proteomics* 15 (3), 989–1006. doi:10.1074/mcp.M115.055384.
- Hornbeck, P.V., et al., 2015. PhosphoSitePlus, 2014: Mutations, PTMs and recalibrations. *Nucleic Acids Research* 43 (D1), D512–D520. doi:10.1093/nar/gku1267.
- Huang, H.D., et al., 2005. KinasePhos: A web tool for identifying protein kinase-specific phosphorylation sites. *Nucleic Acids Research* 33 (Web Server issue), W226–W229. doi:10.1093/nar/gki471.
- Huang, K.Y., et al., 2016. dbPTM 2016: 10-year anniversary of a resource for post-translational modification of proteins. *Nucleic Acids Research* 44 (D1), D435–D446. doi:10.1093/nar/gkv1240.
- Huang, Y., et al., 2010. CD-HIT suite: A web server for clustering and comparing biological sequences. *Bioinformatics* 26 (5), 680–682. doi:10.1093/bioinformatics/btq003.
- Hunter, T., 2007. The age of crosstalk: Phosphorylation, ubiquitination, and beyond. *Molecular Cell* 28 (5), 730–738. doi:10.1016/j.molcel.2007.11.019.
- Jarrell, K.F., et al., 2014. N-linked glycosylation in archaea: A structural, functional, and genetic analysis. *Microbiology and Molecular Biology Reviews* 78 (2), 304–341. doi:10.1128/MMBR.00052-13.
- Kanehisa, M., et al., 2016. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research* 44 (D1), D457–D462. doi:10.1093/nar/gkv1070.
- Kazlauskas, A., Cooper, J.A., 1989. Autophosphorylation of the PDGF receptor in the kinase insert region regulates interactions with cell proteins. *Cell* 58 (6), 1121–1133. https://doi.org/10.1016/0092-8674(89)90510-2.
- Kim, S., et al., 1997. Identification of N(G)-methylarginine residues in human heterogeneous RNP protein A1: Phe/Gly-Gly-Gly-Arg-Gly-Gly-Gly/Phe is a preferred recognition motif. *Biochemistry* 36 (17), 5185–5192. doi:10.1021/bi9625509.
- Krämer, A., et al., 2014. Causal analysis approaches in ingenuity pathway analysis. *Bioinformatics* 30 (4), 523–530. doi:10.1093/bioinformatics/btt703.
- Lanoue, S., et al., 2014. The functional diversity of protein lysine methylation. *Molecular Systems Biology*. 724. doi:10.1002/msb.134974.
- Lemmon, M.A., Schlessinger, J., 2010. Cell signaling by receptor tyrosine kinases. *Cell* 141 (7), 1117–1134. doi:10.1016/j.cell.2010.06.011.
- Letunic, I., Bork, P., 2017. 20 years of the SMART protein domain annotation resource. *Nucleic Acids Research*. gkx922. doi:10.1093/nar/gkx922.
- Liaw, A., Wiener, M., 2002. Classification and regression by random forest. *R News* 2 (3), 18–22. (Available at: <http://cran.r-project.org/doc/Rnews/>).
- Lin, K., et al., 1996. A protein phosphorylation switch at the conserved allosteric site in GP. *Science* 273 (5281), 1539–1542. doi:10.1126/science.273.5281.1539.
- Linding, R., et al., 2003. GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Research* 31 (13), 3701–3708. https://doi.org/10.1093/nar/gkg519.

- Linding, R., *et al.*, 2008. NetworkKIN: A resource for exploring cellular phosphorylation networks. *Nucleic Acids Research* 36 (Database issue), D695–D699. doi:10.1093/nar/gkm902.
- Lischwe, M.A., *et al.*, 1985. Clustering of glycine and NG,NG-dimethylarginine in nucleolar protein C23. *Biochemistry* 24 (22), 6025–6028. <https://doi.org/10.1021/bi00343a001>.
- Li, T., *et al.*, 2012. Characterization and prediction of lysine (K)-acetyl-transferase specific acetylation sites. *Molecular & Cellular proteomics: MCP* 11 (1), doi:10.1074/mcp.M111.011080.
- Lothrop, A.P., Torres, M.P., Fuchs, S.M., 2013. Deciphering post-translational modification codes. *FEBS Letters* 587 (8), 1247–1257. doi:10.1016/j.febslet.2013.01.047.
- Lu, Q., Li, S., Shao, F., 2015. Sweet Talk: Protein glycosylation in bacterial interaction with the host. *Trends in Microbiology* 23 (10), 630–641. doi:10.1016/j.tim.2015.07.003.
- Maere, S., Heymans, K., Kuiper, M., 2005. BiNGO: A cytoscape plugin to assess overrepresentation of Gene Ontology categories in biological networks. *Bioinformatics* 21 (16), 3448–3449. doi:10.1093/bioinformatics/bti551.
- Mi, H., *et al.*, 2017. PANTHER version 11: Expanded annotation data from Gene Ontology and Reactome pathways, and data analysis tool enhancements. *Nucleic Acids Research* 45 (D1), D183–D189. doi:10.1093/nar/gkw1138.
- Minguez, P., *et al.*, 2012. Deciphering a global network of functionally associated post-translational modifications. *Molecular Systems Biology* 8, 599. doi:10.1038/msb.2012.31.
- Minguez, P., *et al.*, 2015. PTMcode v2: A resource for functional associations of post-translational modifications within and between proteins. *Nucleic Acids Research* 43 (D1), D494–D502. doi:10.1093/nar/gku1081.
- Mosca, R., *et al.*, 2014. 3did: A catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Research* 42 (Database issue), D374–D379. doi:10.1093/nar/gkt887.
- Nguyen, V.-N., *et al.*, 2015. Characterization and identification of ubiquitin conjugation sites with E3 ligase recognition specificities. *BMC Bioinformatics. BioMed Central* 16 (Suppl. 1), S1. doi:10.1186/1471-2105-16-S1-S1.
- Nguyen, V.N., *et al.*, 2016. UbiNet: An online resource for exploring the functional associations and regulatory networks of protein ubiquitylation. *Database* 2016, 1–14. doi:10.1093/database/baw054.
- Nguyen, V.N., *et al.*, 2017. A new scheme to characterize and identify protein ubiquitination sites. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 14 (2), 393–403. doi:10.1109/TCBB.2016.2520939.
- O'Shea, J.P., *et al.*, 2013. pLogo: A probabilistic approach to visualizing sequence motifs. *Nature Methods* 10 (12), 1211–1212. doi:10.1038/nmeth.2646.
- Pang, C.N.I., Hayen, A., Wilkins, M.R., 2007. Surface accessibility of protein post-translational modifications. *Journal of Proteome Research* 6 (5), 1833–1845. doi:10.1021/pr060674u.
- Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (8), 1226–1238. doi:10.1109/TPAMI.2005.159.
- Picotti, P., *et al.*, 2009. Full dynamic range proteome analysis of *S. cerevisiae* by targeted proteomics. *Cell* 138 (4), 795–806. <https://doi.org/10.1016/j.cell.2009.05.051>.
- Pinho, S.S., Reis, C.A., 2015. Glycosylation in cancer: Mechanisms and clinical implications. *Nature Reviews Cancer* 15 (9), 540–555. doi:10.1038/nrc3982.
- Prabakaran, S., *et al.*, 2012. Post-translational modification: Nature's escape from genetic imprisonment and the basis for dynamic information encoding. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 4 (6), 565–583. doi:10.1002/wsbm.1185.
- Reimand, J., *et al.*, 2016. g:Profiler – A web server for functional interpretation of gene lists (2016 update). *Nucleic Acids Research* 44 (W1), W83–W89. doi:10.1093/nar/gkw199.
- Rosen, O.M., *et al.*, 1983. Phosphorylation activates the insulin receptor tyrosine protein kinase. *Proceedings of the National Academy of Sciences of the United States of America* 80 (11), 3237–3240. <https://doi.org/10.1073/pnas.80.11.3237>.
- Schwenk, R.W., *et al.*, 2010. Fatty acid transport across the cell membrane: Regulation by fatty acid transporters. *Prostaglandins Leukotrienes and Essential Fatty Acids* 82 (4–6), 149–154. doi:10.1016/j.plefa.2010.02.029.
- Scott, J.D., Pawson, T., 2009. Cell signaling in space and time: Where proteins come together and when they're apart. *Science* 326 (5957), 1220–1224. doi:10.1126/science.1175668.
- Seet, B.T., *et al.*, 2006. Reading protein modifications with interaction domains. *Nature Reviews Molecular Cell Biology* 7 (7), 473–483. doi:10.1038/nrm1960.
- Song, J., *et al.*, 2017. PhosphoPredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection. *Scientific Reports* 7 (1), 6862. doi:10.1038/s41598-017-07199-4.
- Stein, A., *et al.*, 2009. Dynamic interactions of proteins in complex networks: A more structured view. *The FEBS Journal* 276 (19), 5390–5405. doi:10.1111/j.1742-4658.2009.07251.x. 2009/08/29.
- Stein, A., Aloy, P., 2010. Novel peptide-mediated interactions derived from high-resolution 3-dimensional structures. *PLOS Computational Biology* 6 (5), doi:10.1371/journal.pcbi.1000789.
- Strasser, R., 2016. Plant protein glycosylation. *Glycobiology* 26 (9), 926–939. doi:10.1093/glycob/cww023.
- Szklarczyk, D., *et al.*, 2017. The STRING database in 2017: Quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Research* 45 (D1), D362–D368. doi:10.1093/nar/gkw937.
- Szymanski, C.M., Wren, B.W., 2005. Protein glycosylation in bacterial mucosal pathogens. *Nature Reviews Microbiology* 3 (3), 225–237. doi:10.1038/nrmicro1100.
- Tay, A.P., *et al.*, 2017a. MethylQuant: A tool for sensitive validation of enzyme-mediated protein methylation sites from heavy-methyl SILAC data. *Journal of Proteome Research*. doi:10.1021/acs.jproteome.7b00601.
- Tay, A.P., *et al.*, 2017b. PTMOracle: A Cytoscape app for covisualizing and coanalyzing post-translational modifications in protein interaction networks. *Journal of Proteome Research* 16 (5), 1988–2003. doi:10.1021/acs.jproteome.6b01052.
- Tokmakov, A.A., *et al.*, 2012. Multiple post-translational modifications affect heterologous protein synthesis. *The Journal of Biological Chemistry* 287 (32), 27106–27116. doi:10.1074/jbc.M112.366351.
- Tong, L., *et al.*, 1996. Crystal structures of the human p56lckSH2 domain in complex with two short phosphotyrosyl peptides at 1.0 Å and 1.8 Å resolution. *Journal of Molecular Biology* 256 (3), 601–610. doi:10.1006/jmbi.1996.0112.
- Ullrich, A., Schlessinger, J., 1990. Signal transduction by receptors with tyrosine kinase activity. *Cell* 61 (2), 203–212. doi:10.1007/s00497-011-0177-9.
- Van Roey, K., *et al.*, 2013. The switches.ELM Resource: A compendium of conditional regulatory interaction interfaces. *Science Signaling* 6 (269), <https://doi.org/10.1126/scisignal.2003345>.
- Van Roey, K., Gibson, T.J., Davey, N.E., 2012. Motif switches: Decision-making in cell regulation. *Current Opinion in Structural Biology* 22 (3), 378–385. doi:10.1016/j.sbi.2012.03.004.
- Varshavsky, A., 2011. The N-end rule pathway and regulation by proteolysis. *Protein Science* 20 (8), 1298–1345. doi:10.1002/pro.666.
- Venne, A.S., Kollipara, L., Zahedi, R.P., 2014. The next level of complexity: Crosstalk of posttranslational modifications. *Proteomics* 14 (4–5), 513–524. doi:10.1002/pmic.201300344.
- Via, A., *et al.*, 2009. A structure filter for the eukaryotic linear motif resource. *BMC Bioinformatics* 10, 351. <https://doi.org/10.1186/1471-2105-10-351>.
- Wagner, M., *et al.*, 2005. Linear regression models for solvent accessibility prediction in proteins. *Journal of Computational Biology* 12 (3), 355–369. doi:10.1089/cmb.2005.12.355.
- Wang, L., *et al.*, 2012. ASEB: A web server for KAT-specific acetylation site prediction. *Nucleic Acids Research* 40 (Web Server issue), W376–W379. doi:10.1093/nar/gks437.
- Ward, J.J., *et al.*, 2004. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *Journal of Molecular Biology* 337 (3), 635–645. doi:10.1016/j.jmb.2004.02.002.
- Weiss, W.A., Taylor, S.S., Shokat, K.M., 2007. Recognizing and exploiting differences between RNAi and small-molecule inhibitors. *Nature Chemical Biology* 3 (12), 739–744. doi:10.1038/nchembio1207-739.
- Winter, C., 2006. SCOPPI: A structural classification of protein-protein interfaces. *Nucleic Acids Research* 34 (90001), D310–D314. doi:10.1093/nar/gkj099.
- Winter, D.L., Erce, M.A., Wilkins, M.R., 2014. A web of possibilities: Network-based discovery of protein interaction codes. *Journal of Proteome Research* 13 (12), 5333–5338. doi:10.1021/pr500585p.
- Xu, C., Ng, D.T.W., 2015. Glycosylation-directed quality control of protein folding. *Nature Reviews Molecular Cell Biology* 16 (12), 742–752. doi:10.1038/nrm4073.
- Xue, Y., *et al.*, 2006. PPSP: Prediction of PK-specific phosphorylation site with Bayesian decision theory. *BMC Bioinformatics* 7, 163. doi:10.1186/1471-2105-7-163.
- Yang, P., *et al.*, 2015. Positive-unlabeled ensemble learning for kinase substrate prediction from dynamic phosphoproteomics data. *Bioinformatics* 32 (2), bttv550. doi:10.1093/bioinformatics/bttv550.
- Zanzoni, A., *et al.*, 2011. Phospho3D 2.0: An enhanced database of three-dimensional structures of phosphorylation sites. *Nucleic Acids Research* 39 (Suppl. 1), doi:10.1093/nar/gkq936.
- Zhao, N., *et al.*, 2011. Structural similarity and classification of protein interaction interfaces. *PLOS ONE* 6 (5), e19554. doi:10.1371/journal.pone.0019554.

Further Readings

- Basu, A., Rose, K.L., Zhang, J., *et al.*, 2009. Proteome-wide prediction of acetylation substrates. *Proceedings of the National Academy of Sciences of the United States of America* 106, 13785–13790.
- Betts, M.J., Wichmann, O., Utz, M., *et al.*, 2017. Systematic identification of phosphorylation-mediated protein interaction switches. *PLOS Computational Biology* 13, e1005462.
- Chaudhuri, R., Yang, J., 2017. Cross-species PTM mapping from phosphoproteomic data. In: Wu, C.H., A., C.N., Ross, Karen E. (Eds.), *Protein Bioinformatics: From Protein Modifications and Networks to Proteomics*. New York: Humana Press, pp. 459–469.
- Chauhan, J.S., Bhat, A.H., Raghava, G.P.S., Rao, A., 2012. GlycoPP: A webserver for prediction of N- and O-glycosites in prokaryotic protein sequences. *PLOS ONE* 7, e40155.
- Chen, X., Shi, S.-P., Xu, H.-D., Suo, S.-B., Qiu, J.-D., 2016. A homology-based pipeline for global prediction of post-translational modification sites. *Scientific Reports* 6, 25801.
- Chuang, G.-Y., Boyington, J.C., Joyce, M.G., *et al.*, 2012. Computational prediction of N-linked glycosylation incorporating structural properties and patterns. *Bioinformatics* 28, 2249–2255.
- Horn, H., Schoof, E.M., Kim, J., *et al.*, 2014. KinomeXplorer: An integrated platform for kinome biology studies. *Nature Methods* 11, 603–604.
- Huang, Y., Xu, B., Zhou, X., *et al.*, 2015. Systematic characterization and prediction of post-translational modification cross-talk. *Molecular & Cellular Proteomics* 14, 761–770.
- Jia, J., Zhang, L., Liu, Z., Xiao, X., Chou, K.-C., 2016. pSumo-CD: Predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC. *Bioinformatics* 32, 3133–3141.
- Maurer-Stroh, S., Eisenhaber, B., Eisenhaber, F., 2002. N-terminal N-myristoylation of proteins: Prediction of substrate proteins from amino acid sequence. *Journal of Molecular Biology* 317, 541–557.
- Maurer-Stroh, S., Eisenhaber, F., 2005. Refinement and prediction of protein prenylation motifs. *Genome Biology* 6, R55.
- Shi, Y., Guo, Y., Hu, Y., Li, M., 2015. Position-specific prediction of methylation sites from sequence conservation based on information theory. *Scientific Reports* 5, 12403.
- Wang, X.-B., Wu, L.-Y., Wang, Y.-C., Deng, N.-Y., 2009. Prediction of palmitoylation sites using the composition of k-spaced amino acid pairs. *Protein Engineering, Design & Selection* 22, 707–712.
- Woodsmith, J., Kamburov, A., Stelzl, U., 2013. Dual coordination of post translational modifications in human protein networks. *PLOS Computational Biology* 9, e1002933.
- Xu, Y., Ding, J., Wu, L.-Y., Chou, K.-C., 2013. iSNO-PseAAC: Predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition. *PIOS ONE* 8, e55844.

Relevant Websites

- msp.biocuckoo.org
GPS-MSP – Methyl-group Specific Predictor 1.0.
- pail.biocuckoo.org
GPS-PAIL 2.0 – Prediction of Acetylation on Internal Lysines.
- github.com/PengyiYang/KSP-PUEL
KSP-PUEL – Positive-unlabeled ensemble learning for kinase-substrate prediction from dynamic phosphoproteomics data.
- PhosphoPredictphosphopredict.erc.monash.edu
PhosphoPredict – Prediction of kinase-specific phosphorylation sites.
- ptmcode.embl.de
PTMCode 2 – Known and predicted PTM functional associations.
- ptmfunc.com
PTMfunc – Repository of functional predictions for protein PTMs.
- apps.cytoscape.org/apps/ptmoracle
PTMOracle – Co-visualisation and co-analysis of PTM and PPI data.
- elm.eu.org
The Eukaryotic Linear Motif (ELM) resource for Functional Sites in Proteins.
- csb.cse.yzu.edu.tw/UbiNet
UbiNet – A web resource for exploring ubiquitination networks.

Biographical Sketch



Dr. Chi Nam Ignatius Pang is a research associate currently working with Prof Wilkins. He completed his PhD in 2010 on the topic "The Dynamics of Protein Interaction Networks." In his thesis, he explored different mechanisms that govern the dynamics of protein-protein interactions – such as how changes in domain-domain interactions, post-translational modifications, and protein abundance can dynamically switch on/off protein-protein interactions. He also performed a large-scale screen of arginine and lysine methylation in the yeast proteome, and found that there are more methylated proteins than previously thought. In collaboration with Dr. Gene Hart-Smith, he is currently investigating how protein methylation affects protein-protein interactions. This involve identifying changes in protein complexes between wild type yeasts and methyltransferase knockout mutants using protein correlation profiling. Prior to his postdoc he had 2 years of experience in finance involving data analytics and fraud detection.



Prof Marc Wilkins is the Director of the Initiative. Professor Wilkins holds the chair of Systems Biology in the School of Biotechnology and Biomolecular Sciences. His research interests involve the proteomics of protein-protein interactions and systems biology. Specifically, Professor Wilkins' research focuses on the dynamics of protein-protein interaction networks, and the role that gene expression and protein post-translational modifications play in the control of protein interactions and thus delivery of cellular function in yeast and human cells.

Protein Properties

Kentaro Tomii, Artificial Intelligent Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Protein molecules, the main constituents of living things, are organic compounds consisting mainly of carbon, nitrogen, oxygen, hydrogen, and sulfur. Protein behavior such as dynamics and interactions depends on the protein's physicochemical properties, even when a protein performs complex and exquisite roles *in vivo*. Therefore, knowing protein properties is expected to be useful for analyzing and understanding proteins.

A protein possesses its own unique properties to perform its function. Proteins are classifiable roughly into four types: globular proteins, membrane proteins, fibrous proteins, and intrinsically disordered proteins. Protein properties differ depending on the protein type. For instance, most membrane proteins show higher hydrophobicity than proteins of other types. This nature is strongly associated with their roles in a cell, i.e., the manner in which they localize and work at membranes. This characteristic, which is related to the protein solubility and other properties, is also useful to classify them using computational methods as introduced in the section below, although proteins of different types can be fused. For instance, a protein can have soluble domain(s) and transmembrane domain(s), or can have soluble domain(s) and intrinsically disordered regions. The function of the entire protein is based on the properties of individual domains.

Various properties of proteins or protein domains can be measured using physicochemical and biochemical experiments. However, indeed, precise measurements cannot be obtained easily. Therefore, conducting research on an unknown protein is expected to be effective to ascertain the approximate values of various protein properties using computational methods. This article briefly introduces widely various *in silico* methods and tools used for predicting and calculating protein properties such as molecular weight, isoelectric point, solubility, crystallizability, and hydrophobicity.

Amino Acids

A protein is a string of amino acids. The arrangement of that string (primary structure) determines a protein's structure and function. That is to say, amino acid properties are the origin of the structure, function (and also evolution) of proteins. In general, naturally occurring proteins comprise the 20 most common amino acids with different side-chains. Each type of amino acid possesses unique properties that affect the structure and function of proteins, depending on its side-chains. Based on the chemical properties of side-chains, the 20 amino acids are classified roughly into four groups: hydrophobic, polar, and charged residues, and Gly which only has H (hydrogen) as its side-chain. Already, numerous physicochemical and biochemical properties of amino acids have been measured and published.

A compendium of such properties is available. Based on the 188 amino acid properties collected and analyzed by Kidera and his colleagues (Kidera *et al.*, 1985), Nakai *et al.* (1988) launched the so-called AAindex database, which includes 222 amino acid indices: sets of 20 numerical values representing their various properties. Since then, the database has been developed and expanded. In 1996, Tomii and Kanehisa (1996) reformatted the database and released a collection of 402 amino acid indices, with a list of similar indices having correlation coefficients of 0.8 (– 0.8) or more (less), and 42 amino acid substitution matrices representing similarity between amino acids. In a substitution matrix, higher (better) scores are assigned to pairs of residues with more similar properties. They are used for protein sequence analysis such as similarity search and phylogenetic inference. The ever-growing database currently (AAindex Release 9.2) contains 566 indices, 94 matrices, and 47 contact potentials (Kawashima *et al.*, 2008). The AAindex database is available at “see Relevant Websites section”. According to the result of cluster analysis, most of the collected indices are roughly classified into several classes according to characteristics such as hydrophobicity, propensities related to secondary structures, volumes of amino acids, and amino acid composition.

ExPASy offers a tool called ProtScale to visualize a profile as a two-dimensional plot for a protein using 57 indices at “see Relevant Websites section” (Gasteiger *et al.*, 2005). EMBOSS also provides a program called pepwindow via the web browser interface (see “Relevant Websites section”) and Web Services, for drawing a hydropathy plot for a protein based on the hydropathy index (Fig. 1) proposed in an earlier report (Kyte and Doolittle, 1982). It can also be used to draw various plots using various indices by substituting the index in ‘datafile’.

Amino Acid Composition and Molecular Weight

As described above, naturally occurring proteins consist mainly of 20 types of amino acids. The 20 most common amino acids are known to be used unevenly. The frequency of use of individual amino acids is biased. On average, frequently occurring residues such as Ala and Leu account for approximately 9%, although rare residues such as Cys and Trp account for about 1%. Proteins are roughly classified into four types: water-soluble globular proteins, membrane proteins, fibrous proteins, and intrinsically disordered proteins. Amino acid compositions for those four types of proteins differ (see below). As one might expect, they differ for individual proteins. Especially, fibrous proteins such as collagen consist of a characteristic repeat such as Gly Pro-X and Gly X-hydroxyproline, which is a modified proline residue.

```

H KYTJ820101
D Hydropathy index (Kyte-Doolittle, 1982)
R PMID:7108955
A Kyte, J. and Doolittle, R.F.
T A simple method for displaying the hydropathic character of a protein
J J. Mol. Biol. 157, 105-132 (1982)
C JURD980101    0.996  CHOC760103    0.964  OLSK800101    0.942
  JANJ780102    0.922  NADH010102    0.920  NADH010101    0.918
  DESM900102    0.898  EISD860103    0.897  CHOC760104    0.889
  NADH010103    0.885  WOLR810101    0.885  RADA880101    0.884
  MANP780101    0.881  EISD840101    0.878  PONP800103    0.870
  WOLR790101    0.869  NAKH920108    0.868  JANJ790101    0.867
  JANJ790102    0.866  BASU050103    0.863  PONP800102    0.861
  MEIH800103    0.856  NADH010104    0.856  PONP800101    0.851
  PONP800108    0.850  CORJ870101    0.848  WARP780101    0.845
  COWR900101    0.845  PONP930101    0.844  RADA880108    0.842
  ROSG850102    0.841  DESM900101    0.837  BLAS910101    0.836
  BIOV880101    0.829  RADA880107    0.828  BASU050101    0.826
  KANM800104    0.824  LIFS790102    0.824  CIDH920104    0.824
  MIYS850101    0.821  RADA880104    0.819  NAKH900111    0.817
  CORJ870104    0.812  NISK800101    0.812  FAUJ830101    0.811
  ROSM880105    0.806  ARGP820103    0.806  CORJ870103    0.806
  NADH010105    0.804  NAKH920105    0.803  ARGP820102    0.803
  CORJ870107    0.801  MIYS990104    -0.800  CORJ870108    -0.802
  KRIW790101    -0.805  MIYS990105    -0.818  MIYS990103    -0.833
  CHOC760102    -0.838  MIYS990101    -0.840  MIYS990102    -0.840
  MONM990101    -0.842  GUYH850101    -0.843  FASG890101    -0.844
  RACS770102    -0.844  ROSM880101    -0.845  JANJ780103    -0.845
  ENGD860101    -0.850  PRAM900101    -0.850  JANJ780101    -0.852
  GRAR740102    -0.859  PUNT030102    -0.862  GUYH850104    -0.869
  MEIH800102    -0.871  PUNT030101    -0.872  ROSM880102    -0.878
  KUHL950101    -0.883  GUYH850105    -0.883  OOBM770101    -0.899

I   A/L   R/K   N/M   D/F   C/P   Q/S   E/T   G/W   H/Y   I/V
    1.8   -4.5  -3.5  -3.5   2.5  -3.5  -3.5  -0.4  -3.2   4.5
    3.8   -3.9   1.9   2.8  -1.6  -0.8  -0.7  -0.9  -1.3   4.2
//

```

Fig. 1 The AAindex entry of the hydropathy index proposed by Kyte and Doolittle. Each entry in AAindex includes its accession number (denoted as 'H'), the name or short description of the index ('D'), the PubMed ID of literature which publishes the index ('R'), the author name(s) ('A'), the title of the literature ('T'), the bibliographic information of the literature ('J'), the list of similar indices and their correlation coefficients with the index ('C'), the index value for each amino acid ('I'), and the termination symbol ('//').

It is often argued that amino acid composition is associated with structural and biological characteristics of proteins. Actually, chain-length (=domain size) dependence of amino acid composition is also known to exist (White, 1992; Carugo, 2008). Therefore, many computational prediction methods have been proposed, such as the molar extinction coefficient, solubility and crystallizability of proteins, for folding types, structural classes, secondary structure contents, quaternary structural types, and subcellular localization, and for membrane proteins, thermophilic proteins, disordered proteins, based on amino acid and/or dipeptide composition.

A protein's molecular weight is the total of the molecular weights of the protein's constituent amino acids, in addition to the weights originated from posttranslational modifications. Knowing the molecular weight of a protein is helpful for setting and selecting methods for expression and purification systems of it. Although the molecular weight of a protein can be ascertained using analytical biochemistry, exact measurements are quite laborious. At present, as the genome sequences are accumulated, the molecular weights of constituent amino acids are often used to calculate the total by converting the decoded nucleotide sequence into an amino acid sequence. In doing so, it is noteworthy that the molecular weight of the water molecules desorbed along with peptide bond formation is subtracted from the sum of the constituent amino acid molecular weights of the proteins.

Isoelectric Point

Proteins contain ionizable residues including both acidic (aspartic acid and glutamic acid) and basic (arginine, lysine, and histidine) residues. "The isoelectric point (pI) is defined as the level of pH at which the protein/peptide has a net charge of zero

(Smoluch *et al.*, 2016)". Proteins show different pI depending on the type and number of constituent amino acids. For instance, pI of a protein such as aldolase is around 10, although the pI of a protein such as ornithine decarboxylase is around 4 (Dice and Goldberg, 1975). In proteomes, the distribution of pI values of proteins has been known to show a generally bimodal distribution. It has a low fraction of proteins with pI close to 7.4, which closely approximates the pH of the major compartments inside of most cells. A report of one study describes the relation between the length of proteins and their pI values (Kiraga *et al.*, 2007). It might have different charge states depending primarily on the surface charge and solution state. At the pI of a protein, as attractions between protein molecules are maximized, its solubility is minimized. Using such a difference in pI, proteins are separable. Therefore, it is important to calculate pI for each protein. Actually, the pI of a protein is calculable approximately from the protein's amino acid composition. However, generally, it is difficult to obtain the exact value of pI without experimental research.

Generally speaking, although accurate calculation is not easy, various approximate calculation methods have been proposed to estimate the pI of a protein. In brief, the pI of a protein is calculated based on pKa, the inverse logarithm of an acid dissociation constant, of charged residues. Many sets of pKa values for the ionizable groups of seven charged residues (Cys, Asp, Glu, His, Lys, Arg, and Tyr) have been published along with the terminal groups: amino and carboxyl groups. Recently, an isoelectric point calculator, called IPC (see "Relevant Websites section"), has been proposed based on the optimized sets of pKa values using a basin-hopping procedure (Kozłowski, 2016). To develop accurate methods for structure-based calculation of pKa values, a blind prediction challenge was organized: The pKa Cooperative (see "Relevant Websites section"). For this challenge, invited research groups have attempted to predict an unpublished set of experimental pKa values, especially for interior residues in proteins (Nielsen *et al.*, 2011). Actually, pKa calculations for those residues can be difficult.

The pI values of proteins can provide useful information to advance protein science. One report has described that "a relationship exists between a protein's pI and the pH under which it will crystallize" (Kirkwood *et al.*, 2015). One report describes that mammalian complexes of three or more unique proteins have greater homogeneity than the expected values in terms of predicted pI values (Wong *et al.*, 2008).

Solubility and Crystallizability

The solubility of proteins in aqueous solution is regarded as dependent mainly on the surface structure of proteins and the types of amino acids on their surfaces. Generally, proteins with more charged and polar residues on their accessible surface with water molecules show higher solubility. Therefore, the optimum pH differs for each protein. The solubility of a protein also depends on the salt concentration of the solution and the type of salt. Proteins such as membrane proteins, with many hydrophobic residues on their surface, are insoluble in most cases, although they can be solubilized in some cases using surfactants.

Early computational methods for protein solubility prediction from its sequence were simple and were based on the rate of charged residues contained in a protein. In recent years, with the progress of the structural genomics projects worldwide, large amounts of data related to the expression system of many proteins and their results have been accumulated. Therefore, most current prediction methods use such huge datasets as learning datasets. Furthermore, they are based on machine learning methods, which makes it possible to use not only the rate of charged residues but also various physicochemical properties such as those in AAindex for solubility prediction. The prediction accuracy has improved as a consequence. A useful interpretation and list of solubility prediction methods for protein production in *Escherichia coli* are available in a recent review (Chang *et al.*, 2014).

X-ray crystallographic analyses still play a major role in protein tertiary structural studies. About 90% of entries containing proteins in PDB were determined using X-ray crystallography as of December 2017. In X-ray crystallography, obtaining diffraction quality crystals of proteins is a central issue. As is the case with solubility prediction, prediction methods for protein crystallization have been developed for and along with the development of structural genomics projects. For advancing structural genomics projects, it is indispensable to select good target proteins and to search optimal crystallization conditions. Like solubility prediction methods, as structural genomics projects progress, large amounts of data related to protein crystallization are becoming available, leading to a vigorous construction of prediction methods using such huge data and machine learning methods. For these reasons, crystallization prediction methods have progressed in recent years by consideration of various protein characteristics related to their crystallization, although it remains controversial, which is the influential factor for protein crystallizability. An exhaustive and practical review of prediction for protein crystallizability has been published (Smiałowski and Wong, 2016).

Hydrophobicity

Proteins usually include hydrophobic residues as well as charged and polar residues. The proportion of hydrophobic residues in a protein is useful for classifying proteins because membrane proteins tend to have more hydrophobic residues in their trans-membrane region(s) than other proteins do. For instance, a GRAVY (GRand AVerage of hydropathY) score, which is calculated by summing the hydropathy values of all the amino acids and by dividing by the number of residues in a protein, has been developed and used for predicting membrane proteins (Kyte and Doolittle, 1982).

Because hydrophilicity and hydrophobicity are two sides of the same coin, Kyte and Doolittle used the term 'hydropathy', which means "strong feelings about water." They originally proposed their hydropathy index based on both the water-vapor transfer free energies and the interior–exterior distribution of the side-chains. Recently, an engineering measurement, contact angles of a water nanodroplet on a(n artificial) planar surface (of peptides), has been used for characterizing hydrophobicity of amino acids

(Zhu *et al.*, 2016). Including these, many different scales for representing hydrophobicity of amino acids have been published to date (Simm *et al.*, 2016). The variation of hydrophobicity scales can be attributed to their underlying formulations. A prominent difference is the way of dealing with amino acids. For instance, Cys residues can be hydrophobic when we consider their locations in proteins, although their hydrophilic nature emerges when we consider individual free amino acids.

Disordered Proteins

Soluble globular proteins, membrane proteins, and fibrous proteins must possess unique tertiary structures under physiological conditions to perform their functions. However, it is considered that intrinsically disordered proteins (IDPs) do not possess unique tertiary structures but instead exist as dynamic structural ensembles, although they have specific structures when they bind to other molecules under physiological conditions. Most IDPs are involved in important roles in cellular signalling and regulation. Many nucleic proteins are known to contain IDPs. Actually, it is considered that about 20%–30% of eukaryotic proteins are IDPs. Intrinsically disordered proteins (or regions) tend to have more abundant Gly, Ser, and Pro residues, and less hydrophobic residues than other proteins (or regions). Elucidating intrinsically disordered regions in a protein is helpful to ascertain the boundaries of protein domains suitable for X-ray crystallographic analysis. Numerous IDP (or region) prediction methods using this tendency and other features have been reported. A comprehensive summary of methods and databases for disorder prediction is available (Lieutaud *et al.*, 2016). Because IDP prediction can reflect difficulties of discrimination between ordered and disordered proteins (or regions) and because more information about IDPs has been accumulated than ever, machine learning methods such as support vector machine (SVM) and artificial neural network (ANN) have been applied for this purpose.

Quaternary Structure

A complex composed of multiple protein chains (subunits) is defined as a quaternary structure, also called a multimer, or biological assembly. Many proteins form their quaternary structures when they function *in vivo*. The quaternary structure of a protein can be ascertained by physicochemical and biochemical studies. However, accurate measurements are generally elaborate, and exhaustive measurements of protein quaternary structures are not easy. Actually, the estimated state(s) of protein quaternary structure measured in solution, those described by the author(s) in PDB, and the states estimated based on analysis of contacts between protein chains in the crystal in PDB are often different. Under these circumstances, development of an accurate method for estimating the quaternary structure of proteins is expected. However, even among homologous proteins, the quaternary structure of proteins is often not conserved as much as its three-dimensional structure is, and in not as many cases. However, even between distantly related proteins, predictions using the quaternary structure of homologous proteins can be effective in some cases (Nakamura *et al.*, 2017).

Useful Resources

Below, we introduce and summarize useful sites and tools other than those introduced into the papers and reviews described above.

	Name	URL	Descriptions
Amino acids	Amino acid explorer	https://www.ncbi.nlm.nih.gov/Class/Structure/aa/aa_explorer.cgi	Starting points for learning biochemical and physicochemical properties of amino acids
	Amino acid information finder	http://www.currentprotocols.com/WileyCDA/CurPro3Tool/toolId-1.html	
Molecular weight	DNA/RNA/Protein molecular weight calculator	http://www.currentprotocols.com/WileyCDA/CurPro3Tool/toolId-8.html	Molecular weight calculator for both nucleotide and amino acid sequences
Isoelectric point	Proteome- <i>pI</i>	http://isoelectricpointdb.org	Database of isoelectric points pre-calculated with 18 methods for 21,721,250 protein sequences from 5029 organisms (Kozłowski, 2017)
Solubility	Recombinant protein solubility prediction	http://www.biotech.ou.edu/	Prediction method for proteins overexpressed in <i>E. coli</i> using logistic regression (Diaz <i>et al.</i> , 2010)
	CamSol	http://www-mvsoftware.ch.cam.ac.uk/index.php/camsolhome	Prediction method for protein solubility and aggregation propensity (Sormanni <i>et al.</i> , 2015)
	Periscope	http://lightning.med.monash.edu/periscope/	Sequence-based predictor for soluble protein expression in the periplasm of <i>E. coli</i> (Chang <i>et al.</i> , 2016)

	Protein-Sol	https://protein-sol.manchester.ac.uk/	Sequence-based method for predicting protein solubility and lysine and arginine content for solubility design (Hebditch <i>et al.</i> , 2017)
Hydrophobicity	GRAVY calculator	http://www.gravy-calculator.de/	Tools for calculating the GRAVY score
	Protein GRAVY	http://www.bioinformatics.org/sms2/protein_gravy.html	
Disordered proteins/ regions List of approx. 40 methods for prediction of intrinsically disordered proteins (regions)	List of protein disorder predictors	http://iimcb.genesilico.pl/metadisorder/	list_of_protein_disorder_tools_programs.html
Quaternary structure	Online analysis tools – Protein quaternary structure	https://molbiol-tools.ca/Protein_quaternary_structure.htm	List of tools for predicting and analyzing quaternary structure(s) of proteins
Various properties	ProtParam	https://web.expasy.org/protparam/	Tool for computing various physicochemical properties, including amino acid composition, molecular weight, <i>pI</i> , GRAVY, and more (Gasteiger <i>et al.</i> , 2005)
	EMBOSS Pepstats	https://www.ebi.ac.uk/Tools/seqstats/emboss_pepstats	Tool for computing various protein properties such as molecular weight, <i>pI</i> , and improbability of expression in inclusion bodies
	Current protocols tools and calculators	http://www.currentprotocols.com/WileyCDA/Section/id-810245.html	Large list of tools and calculators for life scientists
	ExPASy bioinformatics resource portal	https://www.expasy.org/	Extensive collection of databases and tools in widely diverse life sciences (Artimo <i>et al.</i> , 2012)
	Online analysis tools	https://molbiol-tools.ca	Well-organized list of computational tools for molecular biologist by Dr. Kropinski
	OMICtools	https://omictools.com/proteomics-category	Comprehensive collection of tools and databases in proteomics

Closing Remarks

Elucidation of protein properties is important and helpful for advancing protein research. Many computational methods for the broad range of protein properties have been developed and have been made available, especially along with the development of structural genomics projects as described above. These methods are expected to become more helpful and reliable for our research, according to their improvement and the accumulation of related data.

Acknowledgement

We thank Professor Kenta Nakai for the invitation to contribute this article.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Strings and Sequences: Searching Motifs. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Epitope Predictions. Functional Genomics. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein Localization. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Functional Annotation. Protein Post-Translational Modification Prediction. Protein Post-Translational Modification Prediction. Proteomics Data Representation and Databases. Proteomics Mass Spectrometry Data Analysis Tools. Secondary Structure Prediction. Sequence Analysis. Sequence Composition. The Evolution of Protein Family Databases. Transcription Factor Databases. Transmembrane Domain Prediction

References

- Artimo, P., Jonnalagedda, M., Arnold, K., *et al.*, 2012. ExPASy: SIB bioinformatics resource portal. *Nucleic Acids Research* 40, W597–W603.
 Carugo, O., 2008. Amino acid composition and protein dimension. *Protein Science* 17, 2187–2191.

- Chang, C.C., Li, C., Webb, G.I., *et al.*, 2016. Periscope: Quantitative prediction of soluble protein expression in the periplasm of *Escherichia coli*. *Scientific Reports* 6, 21844.
- Chang, C.C., Song, J., Tey, B.T., Ramanan, R.N., 2014. Bioinformatics approaches for improved recombinant protein production in *Escherichia coli*: Protein solubility prediction. *Briefings in Bioinformatics* 15, 953–962.
- Diaz, A.A., Tomba, E., Lennarson, R., *et al.*, 2010. Prediction of protein solubility in *Escherichia coli* using logistic regression. *Biotechnology and Bioengineering* 105, 374–383.
- Dice, J.F., Goldberg, A.L., 1975. Relationship between *in vivo* degradative rates and isoelectric points of proteins. *Proceedings of the National Academy of Sciences of the United States of America* 72, 3893–3897.
- Gasteiger, E., Hoogland, C., Gattiker, A., 2005. Protein identification and analysis tools on the ExPASy server. In: Walker, J.M. (Ed.), *The Proteomics Protocols Handbook*. Humana Press, pp. 571–607.
- Hebdtich, M., Carballo-Amador, M.A., Charonis, S., Curtis, R., Warwicker, J., 2017. Protein-Sol: A web tool for predicting protein solubility from sequence. *Bioinformatics* 33, 3098–3100.
- Kawashima, S., Pokarowski, P., Pokarowska, M., *et al.*, 2008. AAindex: Amino acid index database, progress report 2008. *Nucleic Acids Research* 36, D202–D205.
- Kidera, A., Konishi, Y., Oka, M., Ooi, T., Scheraga, H.A., 1985. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *Journal of Protein Chemistry* 4, 23–55.
- Kiraga, J., Mackiewicz, P., Mackiewicz, D., *et al.*, 2007. The relationships between the isoelectric point and length of proteins, taxonomy and ecology of organisms. *BMC Genomics* 8, 163.
- Kirkwood, J., Hargreaves, D., O'Keefe, S., Wilson, J., 2015. Using isoelectric point to determine the pH for initial protein crystallization trials. *Bioinformatics* 31, 1444–1451.
- Kozlowski, L.P., 2016. IPC – Isoelectric point calculator. *Biology Direct* 11, 55.
- Kozlowski, L.P., 2017. Proteome-*pt*: Proteome isoelectric point database. *Nucleic Acids Research* 45, D1112–D1116.
- Kyte, J., Doolittle, R.F., 1982. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology* 157, 105–132.
- Lieutaud, P., Ferron, F., Uversky, A.V., *et al.*, 2016. How disordered is my protein and what is its disorder for? A guide through the “dark side” of the protein universe. *Intrinsically Disordered Proteins* 4, e1259708.
- Nakai, K., Kidera, A., Kanehisa, M., 1988. Cluster analysis of amino acid indices for prediction of protein structure and function. *Protein Engineering* 2, 93–100.
- Nakamura, T., Oda, T., Fukasawa, Y., Tomii, K., 2017. Template-based quaternary structure prediction of proteins using enhanced profile-profile alignments. *Proteins*.
- Nielsen, J.E., Gunner, M.R., Garcia-Moreno, B.E., 2011. The pKa Cooperative: A collaborative effort to advance structure-based calculations of pKa values and electrostatic effects in proteins. *Proteins* 79, 3249–3259.
- Simm, S., Einloft, J., Mirus, O., Schleiff, E., 2016. 50 years of amino acid hydrophobicity scales: Revisiting the capacity for peptide classification. *Biological Research* 49, 31.
- Smialowski, P., Wong, P., 2016. Protein crystallizability. *Methods in Molecular Biology* 1415, 341–370.
- Smoluch, M., Mielczarek, P., Drabik, A., Silberring, J., 2016. Online and offline sample fractionation. In: Ciborowski, P., Silberring, J. (Eds.), *Proteomic Profiling and Analytical Chemistry*, second ed. Elsevier, pp. 63–99.
- Sormanni, P., Aprile, F.A., Vendruscolo, M., 2015. The CamSol method of rational design of protein mutants with enhanced solubility. *Journal of Molecular Biology* 427, 478–490.
- Tomii, K., Kanehisa, M., 1996. Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Engineering* 9, 27–36.
- White, S.H., 1992. Amino acid preferences of small proteins. Implications for protein stability and evolution. *Journal of Molecular Biology* 227, 991–995.
- Wong, P., Althammer, S., Hildebrand, A., *et al.*, 2008. An evolutionary and structural characterization of mammalian protein complex organization. *BMC Genomics* 9, 629.
- Zhu, C., Gao, Y.A., Li, H., *et al.*, 2016. Characterizing hydrophobicity of amino acid side chains in a protein environment via measuring contact angle of a water nanodroplet on planar peptide network. *Proceedings of the National Academy of Sciences of the United States of America* 113, 12946–12951.

Relevant Websites

<http://www.genome.jp/aaindex/>
AAindex.

https://www.ebi.ac.uk/Tools/seqstats/emboss_pepwindow/
EMBOSS Pepwindow.

<http://web.expasy.org/protscale/>
ExPASy
ProtScale.

<http://isoelectric.ovh.org>
IPC.

<http://www.pkacoop.org>
pKa Cooperative.

The Evolution of Protein Family Databases

Teresa K Attwood, The University of Manchester, Manchester, United Kingdom

Alex L Mitchell, EMBL-EBI, Hinxton, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

Elucidation of protein function forms a major part of understanding fundamental natural processes, including the basis of diseases and the interaction of species with their environments. Characterising protein function using laboratory-based experiments is a high-cost, low-throughput process that cannot keep pace with the volumes of data being generated by genomic and environmental sequencing projects. The problem has become particularly acute in recent years, as sequencing technology has developed to the point where it is possible to sequence entire genomes, or to generate hundreds of millions of DNA sequences from environmental samples, in a single experiment. The pace of data generation now outstrips the rate of experimental characterisation by many orders of magnitude, and the disparity is likely to get worse as sequencing technologies improve. For the custodians of the data, charged with the task of assigning functions to the mass of raw sequences, the deluge has created almost unimaginable annotation burdens. To give an example, during the last 30 years, dedicated teams of curators have manually annotated ~555,000 UniProtKB/Swiss-Prot sequences; this figure is roughly equal to the number of sequences now entering UniProtKB per week. To put these figures in context, the European Bioinformatics Institute Metagenomics peptide database now contains almost 50 million sequences, derived from 400 data-sets ([Mitchell et al., 2017](#)). More than 15 million of these are predicted to be full length, but only ~1 million currently have counterparts in UniProtKB; and the database is expected to grow to 100s of millions or billions of sequences as proteins continue to be assimilated. The future implications for resources like UniProt ([UniProt Consortium, 2017](#)) are staggering. Inevitably, this impossible situation has mandated the use of computational approaches to complement and mitigate curators' Herculean annotation tasks.

Before the situation became as acute as it is today, an innovative strategy to try to address the problem focused on the use of protein family databases. Such databases have been part of the ecosystem of bioinformatics tools and resources for almost 30 years. Underpinned by information derived from multiple sequence alignments, they offer a more sensitive and *scalable* alternative to pairwise sequence analysis approaches (e.g., such as BLAST ([Altschul et al., 1990](#))) that compare a query sequence against all proteins in a target database (consider, for example, that Tara Oceans scientists have estimated that a BLASTx comparison of ~120 million of their eukaryotic sequences against UniRef90 (containing ~26 million sequences) and the many millions of sequences in the Marine Microbial Eukaryote Transcriptome Sequencing Project (MMETSP; [Keeling et al., 2014](#)) would require ~9 million CPU hours on a high-performance computing facility). Nevertheless, the pivotal role that protein family databases now play in sequence analysis is probably under-appreciated: today, much of their work is conducted behind the scenes in support of the principal players – the primary sequence repositories, like UniProtKB.

The origins of protein family databases can be traced to a description – in *Of URFs and ORFs* ([Doolittle, 1986](#)) – of short amino acid 'patterns' that often characterise specific functional sites. In his book, Doolittle asserted that such patterns could be grouped and potentially used to search for similar sites in query sequences. Reading this inspired Bairoch (then a postgraduate student in Geneva working on developing the Swiss-Prot database) to write a program to do just this – he called the program PROSITE ([Bairoch, 2000](#)). At the same time, he scoured the literature for other sequence patterns, but was dismayed to discover that there were few of them; and those he did find weren't sufficiently diagnostic to be useful ([Bairoch, 2000](#)). He therefore decided to create some patterns of his own. In the sections that follow, we explore how Bairoch's pioneering work spawned the creation of numerous protein family databases, and how these became such critical tools for sequence annotation.

Protein Family Classification – Different Approaches

PROSITE

Seeking to create a collection of his own diagnostic 'signatures', Bairoch began by developing short patterns for various binding sites, active sites, sites of post-translational modification, etc. and/or particular family groupings. The process involved creating multiple alignments of representative sequences, inspecting the alignments for conserved residues, or residue groups, and then encoding the observed pattern of conservation in a regular expression-like syntax that would, ideally, provide a consensus for all sequences in the family or sharing the functional site. Importantly, for each, he provided a detailed abstract describing the biological entity (site, domain, family, etc.) encompassed by the expression. The first collection of these patterns – 58 in all – was bundled with the PROSITE search tool and released in March 1988 as part of the PC/Gene software package, which was then being commercialised by IntelliGenetics.

At the time, PROSITE was very useful for diagnosing potential functional sites in individual sequences and/or for suggesting their likely family relationships; but it was also becoming clear that the resource had a much larger role in characterising entire sequence collections, such as those represented by Swiss-Prot ([Bairoch and Boeckmann, 1991](#)). Hence, before long, Bairoch was

Table 1 Sample of some of the first sequence patterns released in PROSITE

Pattern and documentation IDs	Functional site	Sequence Pattern ^a	Pattern length
PS00001; PDOC00001	*N-glycosylation site	N-[P]-[ST]-[P]	4
PS00004; PDOC00004	*cAMP- and cGMP-dependent protein kinase phosphorylation site	[RK](2)-x-[ST]	4
PS00005; PDOC00005	*Protein kinase C phosphorylation site	[ST]-x-[RK]	3
PS00006; PDOC00006	*Casein kinase II phosphorylation site	[ST]-x(2)-[DE]	4
PS00007; PDOC00007	*Tyrosine kinase phosphorylation site	[RK]-x(2,3)-[DE]-x(3)-Y	8–9
PS00008; PDOC00008	*N-myristoylation site	G-[EDRKHPFYW]-x(2)-[STAGCN]-[P]	6
PS00009; PDOC00009	*Amidation site	x-G-[RK]-[RK]	4
PS00010; PDOC00010	Aspartic acid and asparagine hydroxylation site	C-x-[DN]-x(4)-[FY]-x-C-x-C	12
PS00011; PDOC00011	Vitamin K-dependent carboxylation domain	E-x(2)-[ERK]-E-x-C-x(6)-[EDR]-x(10,11)-[FYA]-[YW]	26–27
PS00012; PDOC00012	Phosphopantetheine attachment site	[DEQGSTALMKRH]-[LIVMFYSTAC]-[GNQ]-[LIVMFYAG]- [DNEKHS]-S-[LIVMST]-[PCFY]-[STAGCPQLIVMF]- [LIVMATN]-[DENQGTAKRHLM]-[LIVMWSTA]- [LIVGSTACR]-[LPIY]-[IVY]-[LIVMFA]	16

^aWithin these expressions, amino acid residues (denoted using the IUPAC single-letter notation) in square brackets are allowed at that position; those in curly brackets are forbidden at that position; x is a wild-card, denoting any amino acid; numbers in parentheses indicate how many times a residue, or residue group, can occur; and a residue on its own is strictly conserved.

Note: Asterisks denote patterns that have been shown to exhibit poor diagnostic performance.

under pressure to liberate PROSITE from its commercial shackles. Accordingly, in October 1989, he announced that he was making the resource publicly available, and released a new version (4.0) the following month, with 202 entries. **Table 1** shows some of the first patterns collected in PROSITE (Bairoch, 1991).

The first thing to notice in **Table 1** is that most of the expressions are very short and, in a sense, quite ‘primitive’ – some are just 3 or 4 residues long. In the late 1980s, when sequence databases were still relatively small (by comparison with the vast, genome-scale repositories of today), these and other PROSITE patterns were nevertheless quite useful indicators of potential functional sites or family relationships. However, the shorter a pattern and the more variable it is (i.e., the larger the allowed residue groups and the more wild-cards used), the poorer its diagnostic performance. Inevitably, as target sequence databases grew larger, this was to become increasingly problematic.

In 1994, the diagnostic performance of then current PROSITE patterns was analysed statistically with respect to Swiss-Prot 30.0 and to a randomised database derived from it by regional shuffling. The frequency of occurrence of the patterns depicted in **Table 1** is shown in **Table 2**. As can be seen, the most promiscuous patterns are those that are shortest and most variable. In fact, patterns PS00001 to PS00009 are so noisy that they’re flagged as ‘patterns with a high probability of occurrence’ (this flag allows such promiscuous matchers to be ignored by modern PROSITE-scanning tools).

Interestingly, in **Table 1**, pattern identifiers (IDs) PS00002 and PS00003 are missing; the latter is also missing from **Table 2**. PS00002 encoded a glycosaminoglycan attachment site (S-G-x-G); this expression was qualified by an additional condition that there should be at least two acidic amino acids from –2 to –4 relative to the serine attachment site. Similarly, PS00003 encoded a tyrosine sulphation site, the consensus features of which were described in terms of the acidic, hydrophobic and polar residues within 1 and 7 residues N- and C-terminally to the tyrosine (specifically, E or D within two residues of the Y (typically at –1); at least three acidic residues from –5 to +5; no more than 1 basic residue and 3 hydrophobic from –5 to +5; at least one P or G from –7 to –2 and from +1 to +7; or at least two or three D, S or N from –7 to +7; absence of disulphide-bonded C residues from –7 to +7; and absence of N-linked glycans near the Y).

In early PROSITE releases, PS00002 and PS00003 were termed ‘rules’. Rules were logical natural-language assertions used to complement or replace patterns that couldn’t be described effectively by the formal consensus-expression syntax. Being written in free-text English, the only way to scan rules against a sequence database was to encode them directly inside the search program; many of the earliest PROSITE-scanning tools were unable to do this (Gattiker *et al.*, 2002).

As the mass of accumulating protein sequences grew, and their diversity widened, the diagnostic power of several PROSITE entries began to erode. Some of these were revised and updated in light of knowledge derived from new family members in Swiss-Prot (see **Table 3**); PS00002, PS00003 and other poorly performing rules and patterns were deleted. However, from the outset, the considerable divergence of some protein families (e.g., globins, immunoglobulins, heat-shock proteins), functional sites and domains (SH2, SH3, PH, Kringle domains, etc.) couldn’t be encapsulated efficiently using sequence patterns. Part of the problem is

Table 2 Pattern frequencies based on Swiss-Prot 30.0 and the corresponding PROSITE release. Patterns PS00001 to PS00009, which are the shortest and most variable, are the most promiscuous

Pattern		Swiss-Prot 30.0		Randomised		Exp. hits	Quota
ID	Length	Matches	Sequences	Matches	Sequences	/1000 res.	%
PS00001	4	79,458	25,162	74,272	25,707	5.25E + 00	93.47
PS00002	4	6872	5357	6155	4895	4.35E – 01	89.57
PS00004	4	23,439	14,466	23,128	14,653	1.63E + 00	98.67
PS00005	3	185,672	35,425	190,343	35,449	1.35E + 01	102.52
PS00006	4	205,059	34,946	194,184	34,658	1.37E + 01	94.70
PS00007	8–9	21,117	13,635	20,903	13,618	1.48E + 00	98.99
PS00008	6	179,880	34,404	177,861	34,284	1.26E + 01	98.88
PS00009	4	13,478	10,339	12,080	9360	8.54E – 01	89.63
PS00010	12	299	67	4	4	2.83E – 04	1.34
PS00011	26–27	38	38	6	6	4.24E – 04	15.79
PS00012	16	46	46	9	9	6.36E – 04	19.57

Table 3 Evolution of the pattern for rhodopsin-like G protein-coupled receptors (GPCRs) since February 1991

Vsn., date	Pattern PS00237	FP	FN
6.1, Feb. 91	[LMRK]-[RKHQNSL]-x(3)-[NTHR]-[LIVMFYW](2)-[LIVM]-x-[SNH]-[LIV]-x(3)-[DEG]-[LIVMFYWA]	2	4
7.1, Jun. 91	[GSTALIVMC]-[GSTA]-[GSTALIVMF]-x(2)-[LIVMN]-x(2)-[LIVMFT]-[GSTAN]-[LIVMFYWAS]-[DEN]- R -[FYWCH]-x(2)-[LIVM]	0	2
8.0, Dec. 91	[GSTALIVMC]-[GSTAPDE]-[EDPKRH]-x(2)-[LIVMNG]-x(2)-[LIVMFT]-[GSTAN]-[LIVMFYWAS]-[DEN]- R -[FYWCH]-x(2)-[LIVM]	1	0
12.0, Jun. 94	[GSTALIVMYWC]-[GSTANCPDE]-[EDPKRH]-x(2)-[LIVMNGA]-x(2)-[LIVMFT]-[GSTANC]-[LIVMFYWSTAC]-[DENH]- R -[FYWCSH]-x(2)-[LIVM]	20	8
15.0, Jul. 98	[GSTALIVMFYWC]-[GSTANCPDE]-[EDPKRH]-x(2)-[LIVMNQGA]-x(2)-[LIVMFT]-[GSTANC]-[LIVMFYWSTAC]-[DENH]- R -[FYWCSH]-x(2)-[LIVM]	44	66
19.0, Apr. 05	[GSTALIVMFYWC]-[GSTANCPDE]-[EDPKRH]-x-[PQ]-[LIVMNQGA]-x(2)-[LIVMFT]-[GSTANC]-[LIVMFYWSTAC]-[DENH]- R -[FYWCSH]-[PE]-x-[LIVM]	76	282
20.0, Nov. 06	[GSTALIVMFYWC]-[GSTANCPDE]-[EDPKRH]-x-[PQ]-[LIVMNQGA]-[RK]-[RK]-[LIVMFT]-[GSTANC]-[LIVMFYWSTAC]-[DENH]- R -[FYWCSH]-[PE]-x-[LIVM]	74	387
2017_08 Aug, 17	[GSTALIVMFYWC]-[GSTANCPDE]-[EDPKRH]-x-[PQ]-[LIVMNQGA]-[RK]-[RK]-[LIVMFT]-[GSTANC]-[LIVMFYWSTAC]-[DENH]- R -[FYWCSH]-[PE]-x-[LIVM]	153	454

Note: To keep pace with the growth of Swiss-Prot, successive updates aimed to maximise the number of true-positive matches, and minimise the number of false-positives (FP) and missed true sequences (FN). Inevitably, attempts to accommodate the family's sequence diversity led the pattern to become less selective over time. The Arg residue, implicated in G protein coupling, is highlighted.

that matching a consensus expression is a binary event: i.e., a sequence either matches exactly, or not at all. Diagnostically, this approach is very brittle. Efforts to capture highly divergent sequence groups in this way tend to result in patterns that are either too permissive (i.e., make significant numbers of erroneous (false-positive) matches), or too selective (i.e., fail to match significant numbers of genuine family members – false negatives). Attempting to address this problem, two parallel initiatives emerged, complementing some of the diagnostic limitations of patterns and rules with more sophisticated pattern-recognition techniques: these were 'profiles' (Bairoch and Bucher, 1994) and protein 'fingerprints' (Attwood and Beck, 1994).

PROSITE's Profile Library

Profiles (or position-specific weight matrices) are tables of amino acid weights and gap penalties used to compute a similarity score, relative to a pre-defined cut-off value, when aligned with a protein sequence. They were introduced into PROSITE 12.0, in June 1994. The data structure used was a generalisation of that originally described by Gribskov and colleagues (Gribskov *et al.*, 1987, 1990), and modified by Lüthy and co-workers (Lüthy *et al.*, 1994).

Like patterns, profiles are built from sequence alignments; however, they are more tolerant of amino acid changes and of sequence length differences arising from insertions or deletions. In principle, this allows relationships between sequences to be modelled more 'realistically'. One advantage of this is that, by contrast with the specificity of patterns (which focus on short, highly similar regions), profiles can be used to characterise the full length of sequence alignments, whether they encode family-specific groups or family-agnostic domains. Consequently, by design, profiles can encapsulate both conserved and divergent regions. This gives rise to the possibility of unrelated sequences achieving significant scores with partially incorrect alignments. This situation is

mitigated to some extent by the requirement that, in order to be accepted as a true match, a sequence must align correctly with residues known to have particular structural or functional properties (metal-ion binding, catalysis, protein-protein interaction, etc.) as verified by experimental data (Bairoch and Bucher, 1994). Part of a profile is shown in Fig. 1, illustrating the complexity of the scoring system relative to the simplicity of the consensus expressions in Table 1.

Broadly speaking, a profile is a scoring 'matrix' (the elements of which are denoted MA in Fig. 1) containing an alternating sequence of position-specific scores. When a sequence is aligned with a profile, its overall similarity score is derived by summing the scorable components, or 'states': specifically, the 'beginning' (B), 'match' (M), 'insert' (I), 'delete' (D) and 'end' (E) states. In addition, so-called 'state-transition' steps between consecutive positions are also scored: 16 state-transition scores, including transitions between identical states, are defined for all possible transitions between the set of states [B,M,I,D] and [M,I,D,E]: BI, BD, MI, MD, IM, etc. (state-transition scores are analogous to gap-opening penalties used in alignment algorithms).

```
ID G PROTEIN_RECEP_F1_2; MATRIX.
AC PS50262;
DT 01-DEC-2001 CREATED; 01-OCT-2013 DATA UPDATE; 30-AUG-2017 INFO UPDATE.
DE G-protein coupled receptors family 1 profile.
MA /GENERAL SPEC: ALPHABET='ABCDEFGHIJKLMNPQRSTVWYZ'; LENGTH=259;
MA /DISJOINT: DEFINITION=PROTECT; N1=6; N2=254;
MA /NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=1.9359; R2=0.02006056; TEXT='NScore';
MA /CUT OFF: LEVEL=0; SCORE=328; N SCORE=8.5; MODE=1; TEXT='!';
MA /CUT OFF: LEVEL=-1; SCORE=228; N SCORE=6.5; MODE=1; TEXT='?';
MA /DEFAULT: D=-20; I=-20; B1=-100; E1=-100; MI=-105; MD=-105; IM=-105; DM=-105; MM=1; M0=-10;
MA /I: B1=0; BI=-105; BD=-105;
MA /M: SY='G': M=-1,-11,-24,-13,-15,-19,30,-19,-20,-18,-15,-11,-5,-19,-16,-18,-2,-11,-15,-22,-20,-16;
MA /M: SY='N': M=-9,33,-19,15,-2,-18,-2,8,-18,-2,-26,-18,51,-20,-1,-2,9,1,-26,-38,-17,-2;
MA /M: SY='I': M=-1,-21,-16,-26,-21,0,-16,-22,10,-22,8,4,-17,-22,-19,-20,-8,-3,10,-21,-7,-20;
MA /M: SY='L': M=-6,-24,-17,-28,-21,8,-25,-20,14,-24,23,12,-21,-25,-20,-19,-17,-5,11,-17,0,-20;
MA /M: SY='V': M=-1,-19,-16,-23,-21,-2,-24,-14,15,-18,6,7,-16,-24,-18,-17,-7,-1,21,-26,-5,-20;
MA /M: SY='I': M=-8,-28,-16,-33,-25,6,-31,-24,26,-26,24,15,-24,-26,-21,-23,-20,-8,20,-20,-1,-24;
MA /M: SY='V': M=-6,-23,-19,-27,-21,5,-24,-16,8,-19,8,5,-20,-23,-17,-17,-15,-6,5,-2,7,-19;
MA /M: SY='V': M=-4,-22,-14,-26,-21,-5,-23,-23,16,-18,7,6,-19,-22,-18,-19,-7,-1,20,-24,-8,-21;
MA /M: SY='I': M=-8,-25,-19,-30,-24,14,-29,-20,19,-24,19,11,-21,-25,-21,-20,-17,-4,16,-16,5,-23;
MA /M: SY='F': M=-3,-18,-12,-23,-18,4,-20,-16,2,-17,3,1,-14,-22,-16,-14,-7,0,3,-16,0,-17;
MA /M: SY='R': M=-9,-10,-22,-12,-6,-10,-18,-7,-15,7,-11,-5,-5,-17,-1,14,-6,-4,-11,-18,-5,-5;
MA /M: SY='K': M=-8,1,-21,-1,0,-14,-16,-1,-18,4,-17,-10,3,-12,-1,3,-1,-1,-15,-23,-5,-1;
MA /M: SY='I': M=-8,-7,-25,-7,0,-19,-16,-6,-19,11,-18,-7,-3,-5,2,13,-3,-4,-15,-23,-10,0;
MA /M: SY='R': M=-8,-7,-21,-9,-3,-16,-16,-4,-14,7,-12,-1,-3,-14,3,11,-4,-4,-11,-23,-8,-1;
MA /I: I=-4; MD=-23;
MA /M: SY='L': M=-7,-17,-20,-19,-11,-2,-20,-11,1,-8,11,8,-14,-19,-8,-1,-14,-7,0,-18,-3,-10; D=-4;
MA /I: I=-4; MI=0; MD=-23; IM=0; DM=-23;
MA /M: SY='R': M=-11,-5,-24,-7,-1,-18,-17,7,-20,8,-16,-6,1,-15,6,17,-5,-6,-17,-22,-6,0;
MA /M: SY='T': M=-2,3,-16,-3,-3,-17,-11,-7,-17,-1,-20,-13,9,-13,-1,0,15,16,-11,-31,-13,-2;
.
.
MA /M: SY='P': M=-9,-21,-32,-16,-7,-13,-21,-20,-14,-14,-20,-14,-20,57,-14,-20,-11,-8,-20,-22,-19,-14;
MA /M: SY='I': M=-10,-29,-22,-34,-27,18,-31,-24,23,-26,18,11,-24,-25,-25,-22,-20,-9,17,-7,7,-26;
MA /M: SY='I': M=-7,-28,-20,-34,-27,4,-33,-26,32,-27,22,15,-23,-24,-22,-24,-19,-7,25,-21,-2,-26;
MA /M: SY='Y': M=-17,-20,-21,-22,-21,27,-29,10,-1,-13,1,0,-18,-29,-14,-12,-18,-8,-7,18,59,-20;
MA /I: E1=0;
NR /RELEASE=2017_08,555426;
NR /TOTAL=2510(2508); /POSITIVE=2456(2454); /UNKNOWN=41(41);
NR /FALSE_POS=13(13); /FALSE_NEG=12; /PARTIAL=3;
CC /MATRIX_TYPE=protein domain;
CC /SCALING_DB=reversed;
CC /AUTHOR=K_Hofmann;
CC /TAXO-RANGE=?E?V; /MAX-REPEAT=2;
CC /VERSION=3;
DR Q42385 , 5H1AA_TAKRU, T; Q42384 , 5H1AB_TAKRU, T;
DR Q6XXX9 , 5HT1A_CANLF, T; Q9N297 , 5HT1A_GORGO, T;
DR Q0EAB6 , 5HT1A_HORSE, T; P08908 , 5HT1A_HUMAN, T;
DR Q64264 , 5HT1A_MOUSE, T; Q9N298 , 5HT1A_PANTR, T;
DR Q9N296 , 5HT1A_PONPY, T; P19327 , 5HT1A_RAT , T;
.
.
DR P59528 , TR123_MOUSE, ?; Q7M710 , TR125_MOUSE, ?;
DR Q67ET4 , TR125_RAT , ?; Q67ES0 , TR140_RAT , ?;
DR P16751 , UL78_HCMVA , ?;
DR A1WRP5 , COXX_VEREI , F; Q6UW6 , CRLC_DICDI , F;
DR Q54QV5 , CRLD_DICDI , F; Q54GG1 , CRLF_DICDI , F;
DR Q5UAW9 , GP157_HUMAN, F; Q8C206 , GP157_MOUSE, F;
DR Q5FVG1 , GP157_RAT , F; Q9VRN2 , MTH2_DROME, F;
DR Q09344 , SRABE_CAEEL, F; Q19992 , SRD1_CAEEL, F;
DR O01609 , SRD33_CAEEL, F; O17240 , SRD3_CAEEL, F;
DR P46564 , SRV1_CAEEL, F;
3D 1F88; 1GZM; 1HL1; 1HOF; 1HZX; 1JFP; 1L9H; 1LN6; 1U19; 2G87; 2HPY; 2I35;
3D 2I36; 2I37; 2J4Y; 2KS9; 2KSA; 2KSB; 2LNL; 2PED; 2R4R; 2R4S; 2RH1; 2VT4;
3D 2X72; 2Y00; 2Y01; 2Y02; 2Y03; 2Y04; 2YCW; 2YCX; 2YCY; 2YD0; 2YDV;
.
.
3D 4Z34; 4Z35; 4Z36; 4ZJ8; 4ZJC; 4ZUD; 4ZWJ; 5A8E; 5C1M; 5CXV; 5D5A; 5D5B;
3D 5D6L; 5DGY; 5DHG; 5DHH; 5DSG; 5DYS; 5EN0; 5F8U; 5G53; 5GLH; 5GLI; 5IU4;
3D 5IU7; 5IU8; 5IUA; 5IUB; 5JQH; 5K2A; 5K2B; 5K2C; 5K2D;
PR PRU0521;
DO PDOC00210;
//
```

Fig. 1 Example profile showing position-specific and state-transition scores for rhodopsin-like GPCRs. The 'ALPHABET' and 'LENGTH' parameters indicate that the profile spans a 259-residue portion of the protein. All true-positive (T) and partial (P) matches are listed in the database cross-references (DR); links to structures in the PDB (Rose *et al.*, 2017) are made in the 3D lines. The profile's diagnostic performance is shown in the numerical results (NR).

Fig. 1 shows excerpts from the PROSITE profile for rhodopsin-like GPCRs (G_PROTEIN_RECEP_F1_2, PS50262). Of particular interest is the profile's diagnostic power, encoded in the NR lines. These reveal that the profile statistics were derived from a search of UniProtKB/Swiss-Prot 2017_08, which contained 555,426 sequences. The profile made 2510 matches in 2508 sequences (hence, some matches were repeats); there were 13 false-positive matches and 12 false negatives; three partial matches were also made. With 99.47% precision and 99.51% recall, the diagnostic performance of this profile is near perfect.

Alongside this profile, PROSITE also contains a 17-residue pattern for the family, as seen in Table 3 (G_PROTEIN_RECEP_F1_1, PS00237). Against the same version of UniProtKB/Swiss-Prot, the pattern made 2142 matches in 2138 sequences: 153 of these were false positive; and 31 were partial matches. In addition, 454 sequences were noted as false negatives. Thus, with 92.86% precision and 81.42% recall, the pattern doesn't match the diagnostic performance of the profile. In general, where both patterns and profiles are used to describe a particular protein family or domain, the profile generally has the greater diagnostic power.

As profiles continued to outperform patterns, a growing profile library began to augment PROSITE: by August 2012, the number of profiles exceeded one thousand (Sigrist *et al.*, 2013); by August 2017, they comprised almost half the database. However, creating patterns and profiles, regularly assessing and refining their diagnostic behaviour in response to the changing size and composition of Swiss-Prot, and maintaining their annotation accordingly, is extremely labour intensive. Thus, while the philosophy of providing comprehensive manual annotation for every database entry adds significant value to the database, and still contributes to its popularity, the time-consuming nature of the task is one of the hardest challenges for curators, and has placed an inevitable brake on its growth. By August 2017, PROSITE provided descriptions of 1788 sites, domains and families, diagnosed by 2500 patterns and profiles.

PRINTS

The second pattern-recognition technique devised in response to the diagnostic limitations of some of the original PROSITE patterns was 'fingerprinting' (Attwood *et al.*, 1994). An early idea behind the development of PROSITE was that a pattern, defining a single conserved region, or motif, within a sequence alignment was sufficient to be able to diagnose a complete protein family. However, most alignments contain several motifs, often separated by regions that are poorly conserved and/or gapped. The concept of fingerprints thus stems from the observation that groups of motifs, and their unique inter-relationships, can together create distinctive signatures for particular protein families or domains. From a diagnostic perspective, this offers greater flexibility than patterns, as it affords the possibility to tolerate mis-matches at the level of individual motifs, without compromising the discriminating power of the fingerprint as a whole, as shown in Fig. 2.

The first fingerprint was created in an attempt to capture all known rhodopsin-like GPCRs in Swiss-Prot (Attwood and Findlay, 1993, 1994). The distinguishing trait of GPCRs is the presence of seven conserved transmembrane (TM) domains; the fingerprint therefore encoded each of these motifs. The resulting discriminator proved to be more robust than the existing PROSITE pattern for this family, which was originally based on a conserved region within TM domain 2 (see Table 3). This led the pattern to be

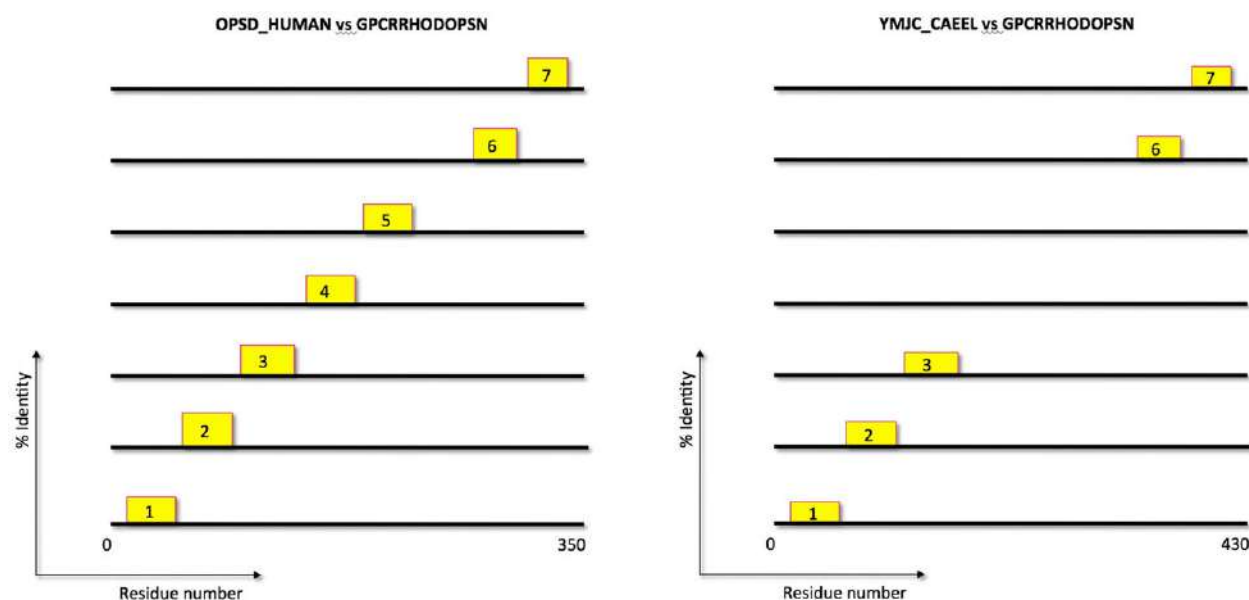


Fig. 2 Complete (human rhodopsin, OPSD_HUMAN) and partial (*C. elegans* putative GPCR, YMJC_CAEEL) matches with the rhodopsin-like GPCR fingerprint. The *C. elegans* protein has a clear relationship with this GPCR family, but lacks significant matches with the fourth and fifth TM domains.

revised (Attwood and Findlay, 1993, 1994), based instead on TM domain 3: this is more tightly conserved, and includes the G protein-interaction site at the N-terminal extremity of the second cytoplasmic loop; the revised pattern was also later complemented by a more potent profile.

In light of the success of the GPCR fingerprint, the technique was used to characterise a range of other protein families, several of which weren't yet available in PROSITE. The results were released in October 1991 in a new resource, initially known as the Features Database (Akrigg *et al.*, 1992). Inspired by PROSITE, the Features Database adhered to the underlying philosophy of adding value to its contents through hand-crafted annotation. Given their similarities, and their growing annotation burdens, there were obvious advantages to unifying their formats (Bairoch, 2000). Nevertheless, a proposal to the European Commission to unite their contents, to create the world's first integrated protein family resource, proved unsuccessful (Attwood *et al.*, 2011).

Despite this setback, both databases continued to evolve. The Features Database, which was growing at a steady rate of 200 fingerprints annually, was subsequently re-branded 'PRINTS' (Attwood and Beck, 1994; Attwood *et al.*, 1994). However, without a dedicated team of curators, this growth rate proved unsustainable. The last release of the PRINTS database was made available in 2012, with 2156 fingerprints encoding 12,444 motifs (Attwood *et al.*, 2012), and is still available for online searching.

Blocks

Following the development of PROSITE and PRINTS, similar attempts were made to characterise protein families, but avoiding heavy reliance on manual approaches to create the signatures and annotate their results. Amongst the first of these was a method that, like fingerprints, made use of multiple conserved motifs, or 'blocks', derived from sequence alignments (Henikoff and Henikoff, 1991). In this approach, the most conserved alignment regions were identified automatically using two independent motif-finding algorithms; only motifs identified by both algorithms were accepted in the final family-specific set of blocks, which were then deposited in the Blocks database.

Blocks' algorithmic approach meant that, for the same family of proteins, the identified blocks generally differed from motifs in the 'equivalent' fingerprint. Importantly also, the blocks-based approach didn't explicitly exploit the relationships between motifs, but treated each motif separately; the diagnostic behaviour of blocks and fingerprints designed to characterise the same families was therefore different. To afford users these different diagnostic perspectives, each release of Blocks was complemented with a copy of Blocks-format PRINTS, derived directly from the manually-selected motifs from the PRINTS database.

In the original versions of the Blocks database, blocks were derived from alignments of sequence families identified in PROSITE. However, as PROSITE's growth was curbed by its high level of manual processing, this was also limiting the growth of Blocks. By the late 1990s, information from a range of additional protein families had become available in complementary databases like PRINTS (Attwood and Beck, 1994), ProDom (Sonnhammer and Kahn, 1994), Pfam (Sonnhammer *et al.*, 1997) and DOMO (Gracy and Argos, 1998). A more comprehensive, non-redundant version of the Blocks database was therefore devised – termed Blocks+ – which included, in a stepwise fashion, i) all blocks derived from PROSITE, ii) blocks derived from PRINTS not present in PROSITE, iii) blocks derived from Pfam not present in PROSITE or PRINTS, and so on for ProDom and DOMO (Henikoff *et al.*, 1999). Eventually, this multiple-database pipeline was replaced by a more streamlined approach utilising protein families from the integrated database, InterPro (Apweiler *et al.*, 2001).

Even though Blocks releases were not reliant on manual annotation, database maintenance was still laborious, and production ceased in April 2007. The final release contained 29,068 blocks, representing 5900 sequence groups documented in InterPro 14.0. Although no longer maintained, Blocks and Blocks-format PRINTS are still available for searching online.

ProDom

A rather different analytical approach emerged from a study of protein modules. These are analogous to molecular building-blocks, discrete domains that have been combined in different ways during the course of evolution to create a range of multi-functional proteins. An automatic method was developed to recognise such modules and domains, and to cluster them into groups. The algorithm was applied to more than 21,000 full-length sequences in Swiss-Prot 21.0, and the resulting domain families were organised automatically into a database – this was ProDom (Sonnhammer and Kahn, 1994). Multiple alignments were also created for each group, from which consensus sequences were generated to facilitate database searches.

A particular challenge with this approach was that the clusters (and their alignments and consensus sequences) had to be recalculated for each release of Swiss-Prot. Not only was this onerous, but the resultant domain 'families' were not consistent between computations on successive Swiss-Prot releases. This lack of stability made indexing ProDom's contents problematic, and led to integration issues with early releases of InterPro (Apweiler *et al.*, 2001); moreover, as Swiss-Prot grew and TrEMBL (Bairoch and Apweiler, 1996) became part of the expanding target database (in 1996, TrEMBL 1.0 held ~105,000 sequences, almost twice the size of Swiss-Prot 34.0, with which it was released in parallel), this brought significant maintenance overheads. The last version of ProDom (ProDom 2012.1), based on more than 21 million full-length sequences from the December 2012 release of UniProtKB/Swiss-Prot and TrEMBL (UniProt Consortium, 2013), included more than 1.7 million domain families containing at least two sequences.

Pfam

During the decade following the release of the first 58 PROSITE patterns, protein family databases proliferated, and database maintenance issues began to take centre stage. The principal tension was between the need to provide users with high-quality, reliable information, and their desire for rapid access to the most comprehensive data. Databases like PROSITE and PRINTS focused on the former, using manual approaches to craft quality alignments and detailed annotation; resources like Blocks and ProDom, on the other hand, made concerted efforts to address the latter, as automation was the only practical way for databases to become more ‘complete’. A kind of hybrid approach was introduced with the advent of Pfam (Sonnhammer *et al.*, 1997), a Hidden Markov Model (HMM)-based database that included both annotated and automatically derived components (denoted Pfam A and Pfam B, respectively).

Conceptually, HMMs are similar to profiles, which model sequence alignments as strings of match, insert and delete states. The main difference is that, rather than using absolute scores, HMMs are probabilistic: i.e., match states instead contain amino acid probabilities within a given alignment column; and transition states denote transition probabilities to and from insert and delete states, reflecting the propensity to insert or skip a residue at a particular position. Pfam A (release 1.0) contained 175 of the most widely populated sequence clusters, seeded by family information from sources like PROSITE, ProDom, etc.; Pfam B contained 11,929 clusters, based on alignments constructed from the portion of Swiss-Prot not already represented in Pfam A (Sonnhammer *et al.*, 1997) – for Pfam 1.0, this was 81% of Swiss-Prot 33.0.

Although Pfam emerged almost ten years after PROSITE, the database nevertheless grew rapidly. In large part, this resulted from a shifting emphasis towards automation, as Pfam adapted to cope with the engorgement of TrEMBL (Finn *et al.*, 2016). Annotating the growing numbers of sequence clusters had become the most burdensome task; it was therefore decided to adopt the Wikipedia model, effectively outsourcing annotation to the community. This meant that, from release 25.0 onwards, the notes traditionally assigned to Pfam entries were to be largely replaced by Wikipedia articles. By March 2017, Pfam 29.0 contained 16,712 entries, and Pfam B, its automatically generated supplement, had been removed.

InterPro

The TrEMBL database, containing translations of all coding sequences in the EMBL data library (Emmert *et al.*, 1994), was released in 1996 as a computer-annotated supplement to Swiss-Prot (Bairoch and Apweiler, 1996). This provided rapid access to protein products of genomic data within a familiar Swiss-Prot-like framework, but in a relatively ‘raw’ state, with little annotation. In 1997, Bairoch, Attwood and Apweiler returned to the earlier discussion of the feasibility of uniting PROSITE and PRINTS. By then, however, the vision was more ambitious: the quest, to develop a resource to annotate TrEMBL entries, to add value to the deluge of uncharacterised genomic data pouring from sequencing projects, using signatures from several protein family resources, including ProDom and the newly announced Pfam database. This time, the case for integration was stronger, and the funding bid for the new project – termed InterPro – was successful. A beta release of InterPro was made in October 1999, characterising 2423 families and domains in Swiss-Prot 38.0 and TrEMBL 11.0: this ‘pilot’ release combined 1370 entries from PROSITE 16.0; 1157 fingerprints from PRINTS 23.1; 241 preliminary profiles from the Profile library; and 1465 entries from Pfam 4.0. It had thus taken seven years finally to realise the goal of an integrated protein family database (Apweiler *et al.*, 2001).

Although ProDom was part of the original InterPro proposal, it was nevertheless excluded from initial releases because the nature of its clusters fluctuated between different versions of Swiss-Prot. Before it could be meaningfully integrated, some means of assigning stable accession numbers to ProDom entries therefore had to be found. Another important factor

Table 4 Composition of InterPro release 64.0, July 2017

Partner database	Version	Entries	# Integrated	% Integrated
Pfam*	31.0	16,712	16,119	96
PANTHER*	11.1	91,538	6,224	7
TIGRFAMs*	15.0	4,488	4,448	99
PIRSF*	3.02	3,285	3,201	97
CDD	3.16	12,805	2,165	17
HAMAP	201.701.18	2,160	2,160	100
PRINTS	42.0	2,106	1,980	94
SUPERFAMILY*	1.75	2,019	1,479	73
ProDom	2006.1	1,894	1,308	69
PROSITE patterns	20.132	1,309	1,298	99
SMART*	7.1	1,312	1,264	96
CATH-Gene3D*	4.1.0	2,737	1,235	45
PROSITE profiles	20.132	1,174	1,142	97
SFLD*	2	480	146	30

Note: Asterisks denote partner databases that contribute HMM-based signatures to the resource.

tempered the early development of InterPro (and remains a challenge today): although conceptually straightforward, amalgamating the contents of PROSITE, PRINTS and Pfam was a major undertaking. The databases use different analysis methods to identify sequence groups; importantly, they also use very different terminologies to describe them - at the time, there was no common definition of what constituted a 'family', a 'superfamily', a 'domain family', and so on. Thus, trying sensibly to merge apparently equivalent database entries, whose terminologies and whose sets of identified sequences were different, was extremely difficult. To facilitate the integration process, initial work therefore focused on partner databases that offered some level of annotation. Hence, ProDom didn't appear in InterPro until release 1.2 in June 2000, to which it contributed 540 entries.

During the next decade, additional resources were considered as potential InterPro partners (Blocks was amongst these; however, by 2001 it was taking source data directly from InterPro, so its inclusion in InterPro would have added a kind of circularity), resulting in seven more databases joining the InterPro consortium (Hunter *et al.*, 2009). By 2017, with 14

Protein family membership

- F** G protein-coupled receptor, rhodopsin-like (IPR000276)
- F** Opsin (IPR001760)

Domains and repeats



Detailed signature matches

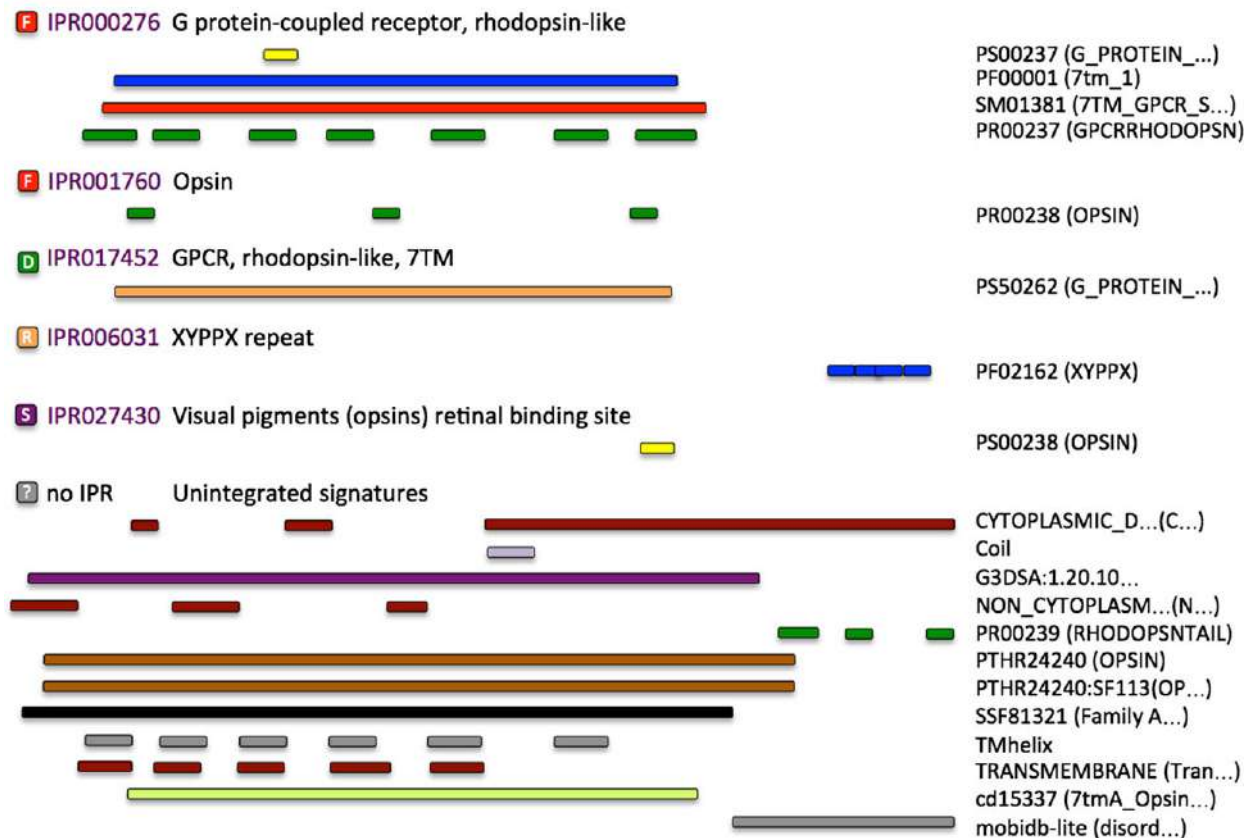


Fig. 3 Output from a search of InterPro with the sequence of rhodopsin from the Japanese flying squid (P31356, OPSPD_TODPA). Matches to signatures within each partner database are colour-coded: PROSITE (patterns), yellow, (profiles) orange; PRINTS, green; Pfam, blue; PANTHER, brown; Gene3D, purple; SUPERFAMILY, black. The 'anatomy' of the sequence is revealed in terms of the overarching superfamily and specific family to which it belongs, the functional sites (G protein-coupling, retinal-binding) and C-terminal repeats it contains, the location of its 7TM-domain and the positions of its constituent TM regions.

contributing data sources, the annotation challenges had become immense, and now lend InterPro a level a complexity that was inconceivable 25 years ago, when integrating protein family databases was first discussed. [Table 4](#) gives a hint of this complexity: in it are listed the versions of the partner databases contributing to the July 2017 release, their sizes and the numbers of their entries included in the resource.

While most of the contents of each partner database are incorporated, aberrant signatures (i.e., those that match unrelated sequences outside their intended scope), or differences in and between them, cause integration of some entries to be deferred. Other concerns include i) how to reconcile sequence-based families with structure-based families (e.g., such as defined in CATH-Gene3D ([Dawson et al., 2017](#)) and SUPERFAMILY ([Oates et al., 2015](#))), whose conceptual models of protein families differ considerably; and ii) how to manage and compare the contents of partner databases that contain vast amounts of (largely automatically generated) data relative to others (e.g., PANTHER ([Mi et al., 2017](#))), especially in cases where their signature collections change substantially from release to release.

The data complexity and semantic issues outlined above make efficient processing very difficult, and annotating the results a continual headache. In consequence, note the discrepancies between columns 3 and 4 of [Table 4](#) – while the integration level of the contents of some partner databases is 90% or more, for others it is less than 50%. Notwithstanding the challenges, with 30,876 entries in release 64.0, InterPro is the largest integrated protein family database, offering the most comprehensive opportunity for protein family characterisation in the world ([Finn et al., 2017](#)). Helping to fulfil its original rationale, more than 80% of TrEMBL entries can now achieve at least some level of annotation using the resource.

InterPro's considerable diagnostic power, with its multiple-database perspective, is illustrated in [Fig. 3](#), which shows the result of searching the database with the sequence of rhodopsin from the Japanese flying squid. The sequence is diagnosed as belonging to the rhodopsin-like GPCR superfamily, being specifically a member of the opsin family; it contains a G protein-coupling motif in TM region 3, and a retinal-binding site in TM region 7; the location of its 7TM-domain is identified, as are the positions of each of the constituent TM regions; and a repetitive region is confirmed to reside C-terminally to the 7TM-domain. No other single database can dissect, in such an extensive, fine-grained way, the 'anatomy' of a protein sequence.

As InterPro has evolved, its composition has changed markedly. Early releases provided a powerful balance of the diagnostic opportunities afforded by patterns, profiles, fingerprints and HMMs, yielding a resource whose diagnostic power was greater than the sum of its parts. Over the years, this balance has, to a large extent, been diluted as more HMM-based databases joined the consortium. [Table 4](#) reveals that the bulk of the contributions to InterPro are now HMM-based (Pfam, PANTHER, TIGRFAMs, PIRSF, etc.); a significant challenge for maintaining InterPro's health in future will therefore be to retain the broad scope and resilience of its original diagnostic diversity. Perhaps a more pressing and invidious threat concerns the financial viability of InterPro's partner resources. As mentioned earlier, several family databases have struggled to sustain regular release schedules; some have ceased production. Many of these resources began effectively as cottage industries, the 'pet' projects of enthusiastic individuals, shared publicly for the benefit of the community. However, without adequate funds, databases cannot survive. It is particularly ironic, then, that the value and importance of InterPro was recently recognised when it was designated an ELIXIR 'core resource'. InterPro's uniqueness derives from the contributions of its partner databases, yet many of these are neither part of ELIXIR, nor considered sufficiently financially viable to be eligible as core resources, presenting a kind of conundrum for InterPro's continuity (and a problem for its future diagnostic finesse).

Closing Remarks

Protein family databases have come a long way since Doolittle's suggestion, more than three decades ago, that sequence patterns could characterise the potential functions of unannotated sequences, and Bairoch created the first protein family database based on such patterns. Today, there are numerous publicly available databases that offer insights into protein functions, structures and family relationships, many more than have been integrated into InterPro. Nevertheless, InterPro stands apart, a union of many different partners, whose different philosophical and methodological approaches have been melded into a uniquely powerful whole, one that now provides the backbone of annotation for newly determined sequences entering TrEMBL, and one that continues to provide a versatile, high-quality diagnostic tool for researchers around the world.

Acknowledgment

We are extremely grateful to Amos Bairoch for the time he devoted to excavating his personal archives, retrieving 'ancient' pre-Web files and emails to verify and augment some of the PROSITE-related information presented here.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Strings and Sequences: Searching Motifs. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Data Storage and Representation. Functional Genomics. Hidden Markov Models. Hidden Markov Models. Homologous Protein Detection. Identification of Homologs. Identification of Sequence Patterns, Motifs and Domains. Information Retrieval in Life Sciences.

Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Protein Functional Annotation. Protein Structural Bioinformatics: An Overview. Protein Structure Classification. Standards and Models for Biological Data: FGED and HUPO. Supervised Learning: Classification. Transmembrane Domain Prediction. Transmembrane Domain Prediction

References

- Akrigg, D.A., Attwood, T.K., Bleasby, A.J., *et al.*, 1992. SERPENT – An information storage and analysis resource for protein sequences. *CABIOS* 8 (3), 295–296.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *J. Mol. Biol.* 215 (3), 403–410.
- Apweiler, R., Attwood, T.K., Bairoch, A., *et al.*, 2001. The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res.* 29 (1), 37–40.
- Attwood, T.K., Beck, M.E., 1994. PRINTS – A protein motif fingerprint database. *Protein Eng.* 7 (7), 841–848.
- Attwood, T.K., Beck, M.E., Bleasby, A.J., Parry-Smith, D.J., 1994. PRINTS – A database of protein motif fingerprints. *Nucleic Acids Res.* 22 (17), 3590–3596.
- Attwood, T.K., Coletta, A., Muirhead, G., *et al.*, 2012. The PRINTS database: A fine-grained protein sequence annotation and analysis resource – Its status in 2012. *Database*. doi:10.1093/database/bae019.
- Attwood, T.K., Findlay, J.B.C., 1993. Design of a discriminating fingerprint for G-protein-coupled receptors. *Protein Eng.* 6 (2), 167–176.
- Attwood, T.K., Findlay, J.B.C., 1994. Fingerprinting G-protein-coupled receptors. *Protein Eng.* 7 (2), 195–203.
- Attwood, T.K., Gisel, A., Eriksson, N.-E., Bongcam-Rudloff, E., 2011. Concepts, historical milestones and the central place of bioinformatics in modern biology: A European perspective. In: Mahdavi, M.A. (Ed.), *Bioinformatics – Trends and Methodologies*. Intech Online Publishers. ISBN 978-953-307-282-1.
- Bairoch, A., 1991. PROSITE: A dictionary of sites and patterns in proteins. *Nucleic Acids Res.* 19, 2241–2244.
- Bairoch, A., 2000. Serendipity in bioinformatics, the tribulations of a Swiss bioinformatician through exciting times!. *Bioinformatics* 16 (1), 48–64.
- Bairoch, A., Apweiler, R., 1996. The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Res.* 24 (1), 21–25.
- Bairoch, A., Boeckmann, B., 1991. The SWISS-PROT protein sequence data bank. *Nucleic Acids Res.* 19, 2247–2249.
- Bairoch, A., Bucher, P., 1994. PROSITE: Recent developments. *Nucleic Acids Res.* 22 (17), 3583–3589.
- Dawson, N.L., Lewis, T.E., Das, S., *et al.*, 2017. CATH: An expanded resource to predict protein function through structure and sequence. *Nucleic Acids Res.* 45 (D1), D289–D295. doi:10.1093/nar/gkw1098.
- Doolittle, R.F., 1986. *Of Urfs and Orfs: A Primer on How to Analyze Derived Amino Acid Sequences*. Mill Valley, CA: University Science Books.
- Emmert, D.B., Stoehr, P.J., Stoesser, G., Cameron, G.N., 1994. The European Bioinformatics Institute (EBI) databases. *Nucleic Acids Res.* 22 (17), 3445–3449.
- Finn, R.D., Attwood, T.K., Babbitt, P.C., *et al.*, 2017. InterPro in 2017 – Beyond protein family and domain annotations. *Nucleic Acids Res.* 45 (D1), D190–D199. doi:10.1093/nar/gkw1107.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Res.* 44 (D1), D279–D285.
- Gattiker, A., Gasteiger, E., Bairoch, A., 2002. ScanProsite: A reference implementation of a PROSITE scanning tool. *Appl. Bioinform.* 1 (2), 107–108.
- Gracy, J., Argos, P., 1998. DOMO: A new database of aligned protein domains. *Trends Biochem. Sci.* 23 (12), 495–497.
- Gribskov, M., Luethy, R., Eisenberg, D., 1990. Profile analysis. *Method Enzymol.* 183, 146–159.
- Gribskov, M., McLachlan, A.D., Eisenberg, D., 1987. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA* 84 (13), 4355–4358.
- Henikoff, S., Henikoff, J.G., 1991. Automated assembly of protein blocks for database searching. *Nucleic Acids Res.* 19 (23), 6565–6572.
- Henikoff, S., Henikoff, J.G., Pietroskovski, S., 1999. Blocks + : A non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics* 15 (6), 471–479.
- Hunter, S., Apweiler, R., Attwood, T.K., *et al.*, 2009. InterPro: The integrative protein signature database. *Nucleic Acids Res.* 37 (Database Issue), D211–D215.
- Keeling, P.J., Burki, F., Wilcox, H.M., *et al.*, 2014. The Marine Microbial Eukaryote Transcriptome Sequencing Project (MMETSP): Illuminating the functional diversity of eukaryotic life in the oceans through transcriptome sequencing. *PLoS Biology* 12 (6), e1001889. Available at: <https://doi.org/10.1371/journal.pbio.1001889>
- Lüthy, R., Xenarios, I., Bucher, P., 1994. Improving the sensitivity of the sequence profile method. *Protein Sci.* 3 (1), 139–146.
- Mi, H., Huang, X., Muruganujan, A., *et al.*, 2017. PANTHER version 11: Expanded annotation data from Gene Ontology and Reactome pathways, and data analysis tool enhancements. *Nucleic Acids Res.* 45 (D1), D183–D189. doi:10.1093/nar/gkw1138.
- Mitchell, A.L., Scheremetjew, M., Denise, H., *et al.*, 2017. EBI Metagenomics in 2017: Enriching the analysis of microbial communities, from sequence reads to assemblies. *Nucleic Acids Res.* doi:10.1093/nar/gkx967.
- Oates, M.E., Stahlhacke, J., Vavoulis, D.V., *et al.*, 2015. The SUPERFAMILY 1.75 database in 2014: A doubling of data. *Nucleic Acids Res.* 43 (D1), D227–D233. doi:10.1093/nar/gku1041.
- Rose, P.W., Prlić, A., Altunkaya, A., *et al.*, 2017. The RSCB protein data bank: Integrative view of protein, gene and 3D structural information. *Nucleic Acids Res.* 45 (D1), D158–D169. doi:10.1093/nar/gkw1099.
- Sigrist, C.J.A., de Castro, E., Cerutti, L., *et al.*, 2013. New and continuing developments at PROSITE. *Nucleic Acids Res.* 41 (D1), D344–D347.
- Sonnhammer, E.L., Eddy, S.R., Durbin, R., 1997. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins* 28 (3), 405–420.
- Sonnhammer, E.L., Kahn, D., 1994. Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci.* 3 (3), 482–492.
- The UniProt Consortium, 2013. Update on activities at the Universal Protein Resource (UniProt). *Nucleic Acids Res.* 41 (D1), D43–D47. doi:10.1093/nar/gks1068.
- The UniProt Consortium, 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Res.* 45 (D1), D158–D169. doi:10.1093/nar/gkw1099.

Further Reading

- Attwood, T.K., Miller, C.J., 2002. Progress in bioinformatics and the importance of being earnest. *Biotechnol. Annu. Rev.* 8, 1–54.
- Attwood, T.K., Pettifer, S.R., Thorne, D., 2016. *Bioinformatics Challenges at the Interface of Biology and Computer Science: Mind the Gap*. John Wiley and Sons. ISBN: 978-0-470-03548-1.
- Bork, P., Bairoch, A., 1996. Go hunting in sequence databases but watch out for the traps. *Trends Genet.* 12 (10), 425–427. Available at: [https://doi.org/10.1016/0168-9525\(96\)60040-7](https://doi.org/10.1016/0168-9525(96)60040-7).
- Faria, D., Schlicker, A., Pesquita, C., *et al.*, 2012. Mining GO annotations for improving annotation consistency. *PLoS ONE* 7 (7), e40519. <https://doi.org/10.1371/journal.pone.0040519>.
- Higgs, P., Attwood, T.K., 2005. *Bioinformatics and Molecular Evolution*. Blackwell Publishing. ISBN: 1-4051-0683-2.
- Holliday, G.L., Davidson, R., Akiva, E., Babbitt, P.C., 2017. Evaluating functional annotations of enzymes using the Gene Ontology. *Methods Mol. Biol.* 1446, 111–132.
- Ioannidis, J.P.A., 2005. Why most published research findings are false. *PLoS Med.* 2 (8), e124. Available at: <https://doi.org/10.1371/journal.pmed.0020124>.
- Mushegian, A., 2011. Grand challenges in bioinformatics and computational biology. *Front. Genet.* 2, 60. doi:10.3389/fgene.2011.00060.

- Poux, S., Magrane, M., Arighi, C.N., *et al.*, 2014. Expert curation in UniProtKB: A case study on dealing with conflicting and erroneous data. Database. Available at: <https://doi.org/10.1093/database/bau016>
- Sangrador-Vegas, A., Mitchell, A.L., Chang, H.-Y., Yong, S.-Y., Finn, R.D., 2016. GO annotation in InterPro: Why stability does not indicate accuracy in a sea of changing annotations. Database. Available at: <https://doi.org/10.1093/database/baw027>
- Schnoes, A.M., Brown, S.D., Dodevski, I., Babbitt, P.C., 2009. Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. PLOS Comput. Biol. 5 (12), e1000605. doi:10.1371/journal.pcbi.1000605.
- Smith, T.F., 1990. The history of the genetic sequence databases. Genomics 6 (4), 701–707. Available at: [https://doi.org/10.1016/0888-7543\(90\)90509-S](https://doi.org/10.1016/0888-7543(90)90509-S).
- Strasser, B.J., 2008. GenBank – Natural history in the 21st century? Science 322 (5901), 537–538. doi:10.1126/science.1163399.
- Wheeler, S.J., Boguski, M.S., 1998. Late-night thoughts on the sequence annotation problem. Genome Res. 8 (3), 168–169.
- Wong, W.-C., Maurer-Stroh, S., Eisenhaber, F., 2010. More than 1001 problems with protein domain databases: Transmembrane regions, signal peptides and the issue of sequence homology. PLOS Comput. Biol. 6 (7), e1000867. Available at: <https://doi.org/10.1371/journal.pcbi.1000867>.

Relevant Websites

http://blocks.fhcrc.org/blocks/blocks_search.html
Blocks.

<http://www.cathdb.info>
CATH/Gene3D.

<https://elixir-europe.org/>
ELIXIR.

<http://www.ebi.ac.uk/interpro/>
InterPro.

<http://pfam.xfam.org/>
Pfam.

<http://130.88.97.239/PRINTS/>
PRINTS.

<http://prodom.prabi.fr/prodom/current/html/home.php>
ProDom.

<http://prosite.expasy.org/>
PROSITE.

<https://www.embl.de/tara-oceans/>
Tara Oceans.

<https://www.uniprot.org/>
UniProt.

Biographical Sketch



Terri is professor of Bioinformatics at the University of Manchester. Her interests in protein sequence analysis led to the creation of various databases (e.g., PRINTS, InterPro) and software tools like Utopia Documents, which uses semantic technologies to integrate research data with scientific articles. She has written several bioinformatics textbooks and reference works. She currently Chairs the GOBLET Foundation, and leads the development of ELIXIR's Training e-Support System, TeSS.



Alex is coordinator for the InterPro and EMBL-EBI Metagenomics databases at the European Bioinformatics Institute. He obtained his DPhil in pharmacology from the University of Oxford, and was previously employed as a molecular biologist at the Institute of Psychiatry in London, before moving to the University of Manchester to work on protein family databases. He joined EMBL-EBI in 2011.

Transmembrane Domain Prediction

Pier Luigi Martelli and Castrense Savojardo, University of Bologna, Bologna, Italy
Giuseppe Profiti, University of Bologna, Bologna, Italy and National Research Council, Italy
Rita Casadio, University of Bologna, Bologna, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

A fraction ranging from 20% to 30% of all genes in any organism encodes for transmembrane (TM) proteins (Krogh *et al.*, 2001). This class is involved in a wide range of functions, enabling almost all communications of matter, energy and information across the biological membranes enclosing cells and organelles. Indeed, only hydrophobic and small molecules can diffuse through membranes and cross them without need of proteins. Owing to their crucial role in physiological processes, TM proteins are the privileged target of drugs: in humans, TM proteins represent around 50% of known pharmaceutical targets (Bakheet and Doig, 2009).

TM proteins consist of three regions: the TM region spans the hydrophobic phase of the lipid bilayer and separate two domains (called intra-cellular and extra-cellular domains, in the case of proteins inserted in the plasma membrane) that are in contact with the aqueous solvent. In multi-pass TM proteins (having more than one membrane-spanning segment), the three regions comprise discontinuous protein segments that fold to form structural domains responsible of protein functions and interaction with other proteins and molecules. Specifically, TM domains can serve, for example, as anchors, channels (gated or not, nonspecific or selective), signal transducers, modifiers of physical and chemical properties of the membrane.

Two major structural classes of transmembrane proteins have been discriminated so far: all- α and β -barrel, whose membrane-spanning segments are α -helices or β -strands, respectively (Fig. 1). Indeed, to stabilize the interaction between the protein and the membrane, the membrane-spanning segments adopt a regular secondary structure: this allows to avoid contacts among the polar groups of the backbone and the hydrophobic chains of the lipid bilayer.

All- α TM proteins known to date comprise a number of helices ranging from 1 to 24 (as in the case of the voltage-gated sodium channel NavPaS from *Periplaneta americana*, PDB code: 5X0M) and have been found in all membrane types with the exception of the outer membrane of Gram-negative bacteria where only β -barrel TM proteins are present. TM β -barrel domains result from the interaction of different β -strands, ranging from 8 to 60 (as in the case of Type II secretion system GspD from *Escherichia coli*, PDB code: 5WQ7) and often including different protein chains. β -barrel TM proteins are also present together with all- α TM proteins in the outer membrane of mitochondria and chloroplasts.

The crystallization of TM proteins and the possibility to resolve their structure by means of X-ray diffractometry is hampered by the need of mimicking with detergents the tripartite bi-phasic environment where TM proteins are stable. Recently, improvements of cryo-electron microscopy technique, not requiring crystallization, opened a new way to resolve the structure of TM protein (Fernandez-Leiro and Scheres, 2016). However, experimental characterization of TM protein structure is still challenging. Computational methods are required for formulating reliable and testable hypotheses on conformation and functional role of TM proteins, whose covalent structure can be deduced from the translation of genomes sequenced with modern Next Generation Sequencing technologies at ever increasing throughput and decreasing cost.

The major computational problems related to TM domains are:

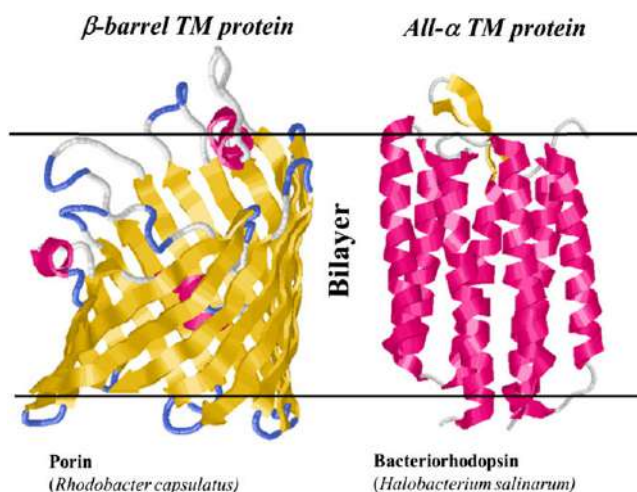


Fig. 1 Two structural classes of TM proteins.

Table 1 TM proteins in UniProtKB (Dec 2017)

Keyword	UniProtKB	With structure	SwissProt	With structure
Transmembrane	20,227,696	4,648	78,045	3,112
Transmembrane α -helix	20,079,319	4,496	76,687	2,990
Transmembrane β -strand	74,199	115	1,027	95

- The determination of the number of membrane-spanning segments and their position along the sequence.
- The determination of the orientation of the membrane protein with respect to the lipid bilayer (topology).
- The determination of the three-dimensional arrangement of the TM spanning segments.
- The determination of the function of TM proteins, including their subcellular localization.

Databases

UniProtKB

Table 1 shows the status of the 2017_12 release of UniProtKB ([The UniProt Consortium, 2017](#)): it contains 20,227,696 proteins labelled with the keyword “transmembrane” (KW-0812), out of 102,804,649 total sequences (19.7%). When restricting to the subset containing the manually curated entries of SwissProt, TM proteins represent 78,045 sequences out of 556,388 (14.0%). Only a 3112 SwissProt sequences (corresponding to 4.0%) are endowed with a three-dimensional structure, at least for one domain. The fraction is much lower when considering all UniProt (0.02%). Moreover, in most cases the experimentally resolved structure is partial and does not include the TM domain. For that reason, specialized datasets collecting proteins with structurally resolved TM domain are needed (see next section).

With few exceptions, TM proteins in UniProtKB are also marked with two more labels, identifying the type: “transmembrane helix” (KW-1133) and “transmembrane beta strand” (KW-1134). It is evident, from the figures in **Table 1**, that most of the known TM proteins belong to the all- α class.

Databases of Structurally Resolved TM Domains

The oldest collection of TM domains with well resolved structure is maintained at UC Irvine laboratory of Prof. White (see “Relevant Websites section”) and, when excluding monotopic proteins not spanning the membrane, it counts 2372 coordinate files relative to 735 unique proteins (release: Jan 2018). The resource collects and expertly classifies the information deposited at the PDB about the available TM domain structures.

One major problem in the analysis of TM protein structure is that neither the exact position of the phospholipids nor the orientation of the protein with respect to the membrane plane are known. Different procedures have been implemented to infer the interaction between membranes and TM proteins, analyzing the physical-chemical properties of the protein surface as determined from the crystallographic structure. Some of them are listed below.

- PDBTM (see “Relevant Websites section”, [Kozma et al., 2013](#)) uses a geometrical approach to locate the most likely position of the lipid bilayer: accessible surfaces of hydrophobic (F, G, I, L, M, V, W and Y) and hydrophilic residues (A, C, D, E, H, K, N, P, Q, R, S and T) are computed; the membrane, represented as two parallel planes more than 15 Å apart, is positioned to minimize the exposure of hydrophobic residues with some constraint on the protein backbone geometry. PDBTM (release: Jun 2017) contains data from 3227 PDB chains, 2848 all- α and 366 β -barrel.
- OPM (Orientations of Proteins in Membranes, see “Relevant Websites section”, [Lomize et al., 2006](#)) adopts an energy based approach in which each protein is modeled as a rigid body with flexible side chains, while the bilayer is represented as a planar hydrophobic slab of adjustable thickness ranging between 21.1 Å and 43.8 Å, depending on the type of biological membrane; transfer energy is calculated at an all-atom level using atomic solvation parameters determined for the water-decadiene system, neglecting explicit electrostatic interactions and contributions of pore facing atoms; most probable protein arrangement is computed by minimizing its transfer energy from water to the membrane. OPM contains (Dec 2017) data from 1783 non-redundant PDB files, 1554 all- α and 229 β -barrel.
- MemProtMD (see “Relevant Websites section”, [Stansfeld et al., 2015](#)) performs a coarse-grained molecular dynamics simulation: groups of atoms are treated collectively as large particles, allowing to increase timescales and system dimensions; the dynamics of the system comprising protein, membrane lipids and solvent is simulated for 1 μ s and the self-assembly of the system is followed. Different lipid species are used in different simulations. MemProtMD currently contains simulations for some 3000 proteins.

Computational Prediction of TM Segments and Topology

Computational characterization of TM protein sequences routinely starts with the prediction of the number of TM segments, of their location along the sequence and their orientation with respect to the lipid bilayer.

Different methods implement different strategies for detecting signals embodied in the sequence in relation to the interaction with the membrane phospholipids and the internal and external polar environments. Local sequence composition is the major source of signal, often complemented with the evolutionary information obtained from multiple sequence alignment of similar proteins. Depending on the TM protein type, the prediction must satisfy different constraints on the number of membrane-spanning segments and their length. In all- α TM proteins the number of membrane-spanning elements is not lower-bounded and their length ranges from 16 to more than 40 residues, depending on their tilt with respect to the membrane plane. Most helices have a clear hydrophobic character, but helices surrounding a pore or in reciprocal interaction routinely show a weaker hydrophobic signal and exhibit a more amphipathic quality. An important topogenic signal is the so-called “positive inside rule” stating that, owing to the unbalance of charge density between the two sides of plasma membranes, the loops facing the cytoplasmic phase (inner loops) are richer in positively charged residues than loops facing the extra-cellular environment (outer loops) (von Heijne, 1989). In eukaryotes, the rule can be applied also to proteins inserted in other membranes, but the definition of inner and outer loops must be redefined for each membrane type (Sharpe *et al.*, 2010).

The generation of the circular arrangement of β -barrels requires at least 8 segments and, so far, the largest resolved barrel consists of 60 strands. In some cases (e.g., TolC, α -hemolysin, Type II secretion system GspD), the functional protein is a multimeric complex in which strands from different subunits interact to form the complete barrel. The minimum number of strands per chain is equal to two. The minimum length of each strand is 6 residues, needed to span the membrane width with an extended conformation. The longest TM β -strand crystallized so far is 24 residue long. Since β -barrels usually delimitate water-filled pores, the strands consist of alternating hydrophobic and hydrophilic residues. Statistical analysis on available structures also reveals that in Gram-negative bacteria, loops exposed towards the extracellular space (outer loops) are, on average, longer than loops facing the periplasmic space (inner loops) (Martelli *et al.*, 2002).

Several confounding factors complicate the prediction of the correct topology. In particular, in the case of all- α membrane proteins the presence of a N-terminal signal peptide with a clear hydrophobic character can lead to the misprediction of an extra helix. Moreover, some all- α TM protein contain partially reentrant segments that span only a portion of the bilayer and strongly challenge the computational characterization of TM segments (Tsirigos *et al.*, 2018).

Scale-Based Methods

Early methods for the localization of TM segments, in particular α -helices, rely on the search of hydrophobic segments along the sequence. More than 20 different scales endowing each residue type with a hydrophobicity value have been proposed to this aim (see “Relevant Websites section”). Among them, the most adopted is the Kyte-Doolittle hydropathy scale (Kyte and Doolittle, 1982) that combines data on water-vapor transfer free energies of the different residues and their exposure distribution in a set of structurally resolved proteins.

Thermodynamic scales have been also generated on the basis of the free energy of transfer between water and hydrophobic liquid (octanol) and between water and phosphocholine interface (White and Wimley, 1999). More recently, a “biological” hydrophobicity scale has been compiled starting from the measurement of the propensity of each residue to be translocated in the endoplasmic reticulum membrane by translocon protein Sec61 (Hessa *et al.*, 2005). This scale has been proved to outperform the others because it probably better casts the physical-chemical constraints deriving from both the protein-membrane interaction and the insertion process. It has been implemented in the SCAMPI2 predictor (Peters *et al.*, 2016).

Hydrophobicity scales are routinely applied to the recognition of TM α -helices by averaging over a window and by searching for hydrophobic stretches at least 15 residue long. The presence of weakly hydrophobic or amphipathic helices strongly limits the performance of scale-based methods. An evolution of the scale-based approach is the first version of MEMSAT (Jones *et al.*, 1994) where 5 different scales (Helix inner end, Helix middle, Helix outer end, Inner loop, Outer loop) are derived from the analysis of 83 all- α TM proteins. During the prediction phase, the five signals computed along the sequence are integrated with a dynamic programming procedure casting the basic grammatical constraints of the topology of all- α TM proteins. Scale-based methods are of scarce relevance for the prediction of β -barrel membrane proteins.

Machine-Learning Based Methods

The best performing methods for the topology prediction of both all- α and β -barrel TM proteins are based on machine-learning approaches such as Neural Networks (NN), Support Vector Machines (SVM), Hidden Markov Models (HMM) and Conditional Random Fields (CRF). Machine learning methods require the availability of a possibly large dataset of proteins with known topology that is used to fix the trainable parameters of the different methods. If the training set is large and diverse enough, the training procedure allows to extract the general rules of the mapping between the sequence and the topology. After the training phase, the parameters can be applied to the prediction of uncharacterized protein sequences. The performance assessment of machine-learning based methods requires a careful validation on protein sets completely independent of the training set. The adoption of machine learning tools allows to analyze more complex input encoding than the simple sequence. In particular, it has been proved that the introduction of evolutionary information extracted from multiple sequence alignments of similar proteins and encoded with sequence profiles largely improves the prediction performance. This has been proved for the prediction of both all- α and β -barrel TM proteins using NNs (Rost *et al.*, 1995; Jacoboni *et al.*, 2001) and HMMs (Martelli *et al.*, 2002; Martelli *et al.*, 2003). While NNs and

SVMs are very efficient in analyzing the local context of a residue, graphical models such as HMMs and Grammatical Restrained Hidden CRFs cast in a more natural way the grammatical constraints that TM proteins must satisfy (schematically depicted in Fig. 2). The complementarity among methods makes it possible to improve the predictive performance by adopting ensemble methods (Martelli *et al.*, 2003; Bernsel *et al.*, 2009) or hybrid approaches (Savojardo *et al.*, 2013).

Table 2 lists some of the most adopted methods for the prediction of all- α TM proteins, specifying whether they are able to deal with reentrant regions (RR column) and with the possible presence of N-terminal signal peptides (SP columns).

Table 3 lists some of the most popular methods for the prediction of β -barrel TM proteins.

Three-Dimensional Reconstruction

The prediction of the TM protein topology is of great help for classifying new protein sequences, restricting the set of possible conformations they can assume and functions they can perform. In some cases, prediction of protein topology improves the search of a structural template in the PDB and assists the target-template alignment for comparative modelling, even at low sequence identity. However, due to the scarcity of structural information on TM proteins, methods allowing the template-independent reconstruction of the 3D structure of TM domains are needed. In most cases, these methods start from a topology prediction and predict the most probable contacts among TM-segments. On this basis, optimization procedures can be applied to search for the most favorable arrangement of the TM segments inside the lipid bilayer, after defining a score accounting for inter-segment and segment-membrane interactions.

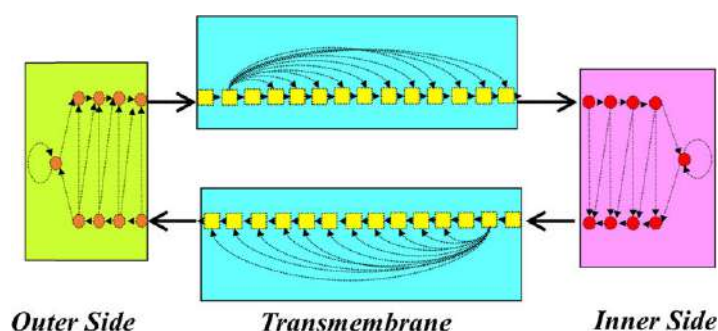


Fig. 2 Generic model of TM protein. State represent different positions along the sequence.

Table 2 Available methods for the topology prediction of all- α TM proteins

	Method ^a	Input ^b	SP ^c	RR ^d	Reference/site
ENSEMBLE	HMM + NN	Prof	No	No	Martelli <i>et al.</i> (2002) mu2py.biocomp.unibo.it/mempype
HMMTOP	HMM	Seq	No	No	Tusndy and Simon (1998) www.enzim.hu/hmmtop/
MEMSAT-SVM	SVM	Prof	Yes	Yes	Nugent and Jones (2009) http://bioinf.cs.ucl.ac.uk/psipred/
PHDhtm	NN	Prof	No	No	Rost <i>et al.</i> (1996) www.predictprotein.org
PHILIUS	DBN	No	Yes	No	Reynolds <i>et al.</i> (2008) www.yeastrc.org/philius/
PolyPHOBIUS	HMM	MSA	Yes	No	Kll <i>et al.</i> (2005) phobius.sbc.su.se/poly.html
SCAMPI2	Scales	Seq	No	No	Peters <i>et al.</i> (2016) scampi.bioinfo.se/
SPOCTOPUS	NN + HMM	Prof	Yes	Yes	Viklund <i>et al.</i> (2008) octopus.cbr.su.se
TMHMM	HMM	Seq	No	No	Sonnhammer <i>et al.</i> (1998) www.cbs.dtu.dk/services/TMHMM/
TOPCONS	Consensus	Prof	Yes	Yes	Tsirigos <i>et al.</i> (2015) topcons.cbr.su.se/

^aMethod: NN = Neural Networks; HMM = Hidden Markov Models; SVM = Support Vector Machines; DBN = Dynamic Bayesian Networks.

^bInput: Seq = Sequence; Prof = Sequence profile; MSA = Multiple Sequence Alignments.

^cSP = prediction of Signal peptide.

^dRR = prediction of reentrant regions.

Table 3 Available methods for the topology prediction of β -barrel TM proteins

	Method ^a	Input ^b	Reference/site
BETAWARE	NN + GRHCRF	Prof	Savojardo et al. (2013) betaware.biocomp.unibo.it
BOCTOPUS2	SVM + HMM	Prof	Hayat et al. (2016) boctopus.bioinfo.se/
PRED-TMBB	HMM	Seq	Bagos et al. (2004) bioinformatics.biol.uoa.gr/PRED-TMBB/
Pred β TM	SVM + rules	Seq	Roy Choudhury and Novič (2015) http://transpred.ki.si/
PROF-TMB	HMM	Prof	Bigelow et al. (2004) www.predictprotein.org

^aMethod: NN = Neural Networks; HMM = Hidden Markov Models; SVM = Support Vector Machines; GRHCRF = Grammatical Restrained Hidden Conditional Random Fields.

^bInput: Seq = Sequence; Prof = Sequence profile.

Available methods for contact predictions in all- α proteins include:

- MEMPACK (see “Relevant Websites section”, [Nugent and Jones, 2010](#)), a method to predict lipid exposure, residue-residue contacts, helix-helix interactions and ultimately overall protein helical packing. Individual classifiers for lipid exposure and residue contacts are based on SVMs and have been trained on data derived from the OPM and PDBTM as well as molecular dynamics data extracted from the Coarse Grained Database ([Chetwynd et al., 2008](#)) (for lipid exposure prediction).
- TMHcon (see “Relevant Websites section”, [Fuchs et al., 2009](#)), a method for predicting helix-helix contact in TM proteins based on NNs and different input features, including: evolutionary information, sequence distance and co-evolution measures of residue pairs as well as protein global features such as protein length and number of TM segments.
- FILM3 (see “Relevant Websites section”, [Nugent and Jones, 2013](#)), a method combining PSICOV ([Jones et al., 2012](#)), an advanced residue-residue contact predictor, with TM topology prediction (performed using MEMSAT-SVM) and fragment selection and assembly. The main novelty of FILM3 consists in the adoption of scoring energy functions (used for fragment selection and assembly) entirely based on distance constraint imposed by predicted TM segments and contacts predicted by PSICOV.
- MemBrain (see “Relevant Websites section”, [Yang and Shen, 2018](#)), a recent approach to predict inter-helix contacts based on deep learning methods.

Functional Characterization

Structural characterization of TM proteins helps in formulating hypotheses about their role in the complexity of biological functions. However, the three-dimensional reconstruction is not always feasible and, in some cases, similar structures are associated to different specific functions or subcellular localizations. For this reason, methods for inferring protein function and localization from sequence have been implemented.

To date, the only prediction method for protein localization specific for TM proteins is MemLoc ([Pierleoni et al., 2011a](#); see “Relevant Websites section”) that classifies protein sequences into three subcellular localizations (Plasma membrane, Organelle membranes, Internal membranes) with a SVM. The MemPype system ([Pierleoni et al., 2011b](#); see “Relevant Websites section”) integrates MemLoc with similarity-based inference methods and tools for the prediction of topology, presence of signal peptide and GPI-anchor.

General-purpose systems for inferring protein functions can be applied to TM proteins (see also [Martelli et al., 2017](#)). The basic idea of these methods is to extrapolate functional information present in databases in order to annotate sequences lacking experimental characterization. For example, the Bologna Annotation Resource (BAR3.0, see “Relevant Websites section”) is a prediction tool that exploits large-scale clustering of UniProtKB sequences resulting from the application of strict similarity constraints (more than 40% sequence identity on an alignment coverage larger than 90%; [Profitti et al., 2017](#)). Functional characterization of each cluster is derived from proteins carrying functional annotation (in terms of Gene Ontology, GO) in UniProtKB, after an analysis of statistical significance. This procedure allows to spread functional annotation to all the members of the clusters. For example, in the present version, BAR 3.0 contains 2294 protein sequences experimentally associated with the GO term “integral component of membrane” (GO:0016021) or its descendants, and the BAR3.0 structure allows spreading the annotation to 751,272 protein sequences.

See also: Algorithms for Strings and Sequences: Searching Motifs. Cell Modeling and Simulation. Data Mining: Classification and Prediction. Data Mining: Prediction Methods. Functional Genomics. Identification of Sequence Patterns, Motifs and Domains. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein Localization. Protein Functional Annotation. Protein Structural Bioinformatics: An Overview. Secondary Structure Prediction. Sequence Analysis. Sequence Composition. The Evolution of Protein Family Databases

References

- Bagos, P.G., Liakopoulos, T.D., Spyropoulos, I.C., Hamodrakas, S.J., 2004. PRED-TMBB: A web server for predicting the topology of beta-barrel outer membrane proteins. *Nucleic Acids Res.* 32, W400–W404.
- Bakheet, T.M., Doig, A.J., 2009. Properties and identification of human protein drug targets. *Bioinformatics* 25, 451–457.
- Bernsel, A., Viklund, H., Hennerdal, A., Elofsson, A., 2009. TOPCONS: Consensus prediction of membrane protein topology. *Nucleic Acids Res.* 37, W465–W468.
- Bigelow, H.R., Petrey, D.S., Liu, J., *et al.*, 2004. Predicting transmembrane beta-barrels in proteomes. *Nucleic Acids Res.* 32, 2566–2577.
- Chetwynd, A.P., Scott, K.A., Mokrab, Y., Sansom, M.S., 2008. CGDB: A database of membrane protein/lipid interactions by coarse-grained molecular dynamics simulations. *Mol. Membr. Biol.* 225, 662–669.
- Fernandez-Leiro, R., Scheres, S.H., 2016. Unravelling biological macromolecules with cryo-electron microscopy. *Nature* 537, 339–346.
- Fuchs, A., Kirschner, A., Frishman, D., 2009. Prediction of helix-helix contacts and interacting helices in polytopic membrane proteins using neural networks. *Proteins* 74, 857–871.
- Hayat, S., Peters, C., Shu, N., *et al.*, 2016. Inclusion of dyad-repeat pattern improves topology prediction of transmembrane β -barrel proteins. *Bioinformatics* 32, 1571–1573.
- Hessa, T., Kim, H., Bihlmaier, K., *et al.*, 2005. Recognition of transmembrane helices by the endoplasmic reticulum translocon. *Nature* 433, 377–381.
- Jacoboni, I., Martelli, P.L., Fariselli, P., *et al.*, 2001. Prediction of the transmembrane regions of beta-barrel membrane proteins with a neural network-based predictor. *Protein Sci.* 10, 779–787.
- Jones, D.T., Buchan, D.W., Cozzetto, D., Pontil, M., 2012. PSICOV: Precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* 28, 184–190.
- Jones, D.T., Taylor, W.R., Thornton, J.M., 1994. A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry* 33, 3038–3049.
- Käll, L., Krogh, A., Sonnhammer, E.L., 2005. An HMM posterior decoder for sequence feature prediction that includes homology information. *Bioinformatics* 21, i251–i257.
- Kozma, D., Simon, I., Tusnády, G.E., 2013. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. *Nucleic Acids Res.* 41, D524–D529.
- Krogh, A., Larsson, B., von Heijne, G., Sonnhammer, E.L., 2001. Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *J. Mol. Biol.* 305, 567–580.
- Kyte, J., Doolittle, R.F., 1982. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.* 157, 105–132.
- Lomize, M.A., Lomize, A.L., Pogozheva, I.D., Mosberg, H.I., 2006. OPM: Orientations of proteins in membranes database. *Bioinformatics* 22, 623–625.
- Martelli, P.L., Fariselli, P., Casadio, R., 2003. An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins. *Bioinformatics* 19, i205–i211.
- Martelli, P.L., Fariselli, P., Krogh, A., Casadio, R., 2002. A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins. *Bioinformatics* 18, S46–S53.
- Martelli, P.L., Profiti, G., Casadio, R., 2017. Protein function prediction. *Ref. Modul. Life Sci.*
- Nugent, T., Jones, D.T., 2009. Transmembrane protein topology prediction using support vector machines. *BMC Bioinform.* 10, 159.
- Nugent, T., Jones, D.T., 2010. Predicting transmembrane helix packing arrangements using residue contacts and a force-directed algorithm. *PLOS Comput. Biol.* 6, e1000714.
- Nugent, T., Jones, D.T., 2013. Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. *PNAS* 109, E1540–E1547.
- Peters, C., Tsigirgos, K.D., Shu, N., Elofsson, A., 2016. Improved topology prediction using the terminal hydrophobic helices rule. *Bioinformatics* 32, 1158–1162.
- Pierleoni, A., Martelli, P.L., Casadio, R., 2011a. MemLoc: Predicting subcellular localization of membrane proteins in eukaryotes. *Bioinformatics* 27, 1224–1230.
- Pierleoni, A., Indio, V., Savojardo, C., *et al.*, 2011b. MemPype: A pipeline for the annotation of eukaryotic membrane proteins. *Nucleic Acids Res.* 39, W375–W380.
- Profiti, G., Martelli, P.L., Casadio, R., 2017. The Bologna Annotation Resource (BAR 3.0): Improving protein functional annotation. *Nucleic Acids Res.* 45, W285–W290.
- Reynolds, S.M., Käll, L., Riffle, M.E., Bilmes, J.A., Noble, W.S., 2008. Transmembrane topology and signal peptide prediction using dynamic bayesian networks. *PLOS Comput. Biol.* 4, e1000213.
- Rost, B., Casadio, R., Fariselli, P., Sander, C., 1995. Transmembrane helices predicted at 95% accuracy. *Protein Sci.* 4, 521–533.
- Rost, B., Fariselli, P., Casadio, R., 1996. Topology prediction for helical transmembrane proteins at 86% accuracy. *Protein Sci.* 5, 1704–1718.
- Roy Choudhury, A., Novič, M., 2015. Pred β TM: A novel β -transmembrane region prediction algorithm. *PLOS One* 10, e0145564.
- Savojardo, C., Fariselli, P., Casadio, R., 2013. BETAWARE: A machine-learning tool to detect and predict transmembrane beta-barrel proteins in prokaryotes. *Bioinformatics* 29, 504–505.
- Sharpe, H.J., Stevens, T.J., Munro, S., 2010. A comprehensive comparison of transmembrane domains reveals organelle-specific properties. *Cell* 142, 158–169.
- Sonnhammer, E.L., von Heijne, G., Krogh, A., 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 6, 175–182.
- Stansfeld, P.J., Goose, J.E., Caffrey, M., *et al.*, 2015. MemProtMD: Automated insertion of membrane protein structures into explicit lipid membranes. *Structure* 23, 1350–1361.
- The UniProt Consortium, 2017. UniProt the universal protein knowledgebase. *Nucleic Acids Res.* 45, D158–D169.
- Tsigirgos, K.D., Govindarajan, S., Bassot, C., *et al.*, 2018. Topology of membrane proteins—predictions, limitations and variations. *Curr. Opin. Struct. Biol.* <http://dx.doi.org/10.1016/j.sbi.2017.10.00> (in press).
- Tsigirgos, K.D., Peters, C., Shu, N., *et al.*, 2015. The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. *Nucleic Acids Res.* 43, W401–W407.
- Tusnády, G.E., Simon, I., 1998. Principles governing amino acid composition of integral membrane proteins: Application to topology prediction. *J. Mol. Biol.* 283, 489–506.
- Viklund, H., Bernsel, A., Skwark, M., Elofsson, A., 2008. SPOCTOPUS: A combined predictor of signal peptides and membrane protein topology. *Bioinformatics* 24, 2928–2929.
- von Heijne, G., 1989. Control of topology and mode of assembly of a polytopic membrane protein by positively charged residues. *Nature* 341, 456–458.
- White, S.H., Wimley, W.C., 1999. Membrane protein folding and stability: Physical principles. *Annu. Rev. Biophys. Biomol. Struct.* 28, 319–365.
- Yang, J., Shen, H.B., 2018. MemBrain-contact 2.0: A new two-stage machine learning model for the prediction enhancement of transmembrane protein residue contacts in the full chain. *Bioinformatics*. <http://dx.doi.org/10.1093/bioinformatics/btx593> (in press).

Relevant Websites

<https://bar.biocomp.unibo.it>
 Biocomputing Group-University of Bologna.
www.csbio.sjtu.edu.cn/bioinf/MemBrain
 MemBrain.
<http://blanco.biomol.uci.edu/mpstruc/>
 Membrane Proteins of Known Structure.

<https://mu2py.biocomp.unibo.it/memloci>

MemLocI-Biocomputing Group.

<http://memprotmd.bioch.ox.ac.uk/>

MemProtMD.

<https://mu2py.biocomp.unibo.it/mempype>

MemPype-Biocomputing Group.

<http://opm.phar.umich.edu/>

Orientations of Proteins in Membranes.

<http://pdbtm.enzim.hu/>

Protein Data Bank of Transmembrane Proteins.

<http://web.expasy.org/protscale/>

ProtScale.

<http://bioinf.cs.ucl.ac.uk/psipred/>

PSIPRED Protein Sequence Analysis Workbench.

<http://webclu.bio.wzw.tum.de/tmhcon/>

TMHcon.

http://bioinf.cs.ucl.ac.uk/software_downloads/

UCL-CS Bioinformatics: Software & Downloads.

Prediction of Protein Localization

Kenta Nakai, The University of Tokyo, Tokyo, Japan

Kenichiro Imai, Advanced Industrial Science and Technology, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the body of multicellular organisms, there are many cell types, each of which contains a set of proteins, some of which are specific to it while some are ubiquitous to all cell types. With the development of single-cell transcriptomics technology, precise mapping of each protein to its expressed cells is ongoing. Such information will be undoubtedly useful in clarifying the cellular function of these proteins. Similarly, each protein is localized at a compartment(s) (organelle) within a cell (or excreted to outside of the cell). Such knowledge would be also useful in understanding the function of the protein because each compartment plays some specific roles within the cell. Especially, knowledge on whether a protein is secreted or not may have particular importance in pharmaceutical and medical sciences. And it should be noted that the information of protein subcellular localization is also important for unicellular organisms, such as bacteria. Attempts to systematically determine the subcellular localization within a cell have been performed in the field of proteomics. Nowadays, even the information of semi-quantitative localization frequency at multiple localization sites for a protein is accumulating.

It has been demonstrated that many (if not all) proteins are sorted to their final localization site after their synthesis in the cytosol according to the information that is encoded as part(s) of their own amino acid sequence (known as the signal hypothesis). Typically, such information is encoded as an N-terminal presequence and is called the targeting signal. In other words, some subcellular machinery recognizes and interprets such signals in the nascent proteins. Therefore, the prediction of protein subcellular localization from their amino acid sequence can have some extra biological rationale: If we can mimic the recognition process within the cell, we would be able to predict the localization, in principle. For practical purposes, however, additional information, such as the protein–protein interaction data, could be used. It should be also emphasized that the prediction problem has been used as a test bed for a variety of machine-learning algorithms probably because of its simplicity in scheme and its potential practical value. In this article, we will sketch a brief overview of this field, introducing some representative works. Finally, we will discuss its future directions.

Topics

Prediction of Subcellular Localization Sites

Basic prediction scheme

Since possible repertory of localization sites can be different between organisms, the basic input information is a set of a (precursor) amino acid sequence and its species origin. In eukaryotic cells, the category has been conventionally divided into animal cells and plant ones; unicellular eukaryotes, such as yeasts, can be treated as an animal or a special type of plant lacking chloroplasts. In many cases, the repertory for animal cells are: The cytosol, the nucleus, the mitochondrion, the endoplasmic reticulum (ER), the Golgi apparatus, the lysosome, the plasma membrane and the extracellular space (i.e., excreted). In plants, vacuoles are usually treated as equivalent with lysosomes in animals. In addition, plastids (typically chloroplasts) are added in plants. Sometimes, (integral) membrane proteins in compartments, such as mitochondria, are collectively treated as membrane proteins. It is known that some of the above compartments, such as mitochondria and plastids, have further precise structures because they are composed of multiple membranes. The nucleolus in the nucleus would be regarded as an independent site. However, attempts to include more precise predictions are rare mainly because of the difficulty in their data size: If a site contains only a small number of known proteins, its treatment in the machine-learning scheme is difficult. Similarly, in spite of their biological importance, peroxisomes are often not included in the repertory because of their relatively small sample size. Sometimes, cytoskeletal proteins are regarded as an independent site. The treatment of peripheral membrane proteins, that is, proteins that are weakly bound to the membrane mainly through protein–protein interaction, is quite difficult (except the cases when they are the lipid-anchored proteins). Theoretically, they can be treated as soluble proteins but both experimental data and practical expectation prefer them to be treated as membrane proteins. In bacterial predictions, the repertories have been classified into two, depending on the presence or the absence of the outer membrane: Gram-negative and Gram-positive bacteria. In Gram-positive bacteria, the repertory is the cytoplasm, the cell membrane, and the extracellular space while in Gram-negative ones, it is the cytoplasm, the inner membrane, the periplasm, the outer membrane, and the extracellular space. The cell membrane in Gram-positive bacteria is considered to be equivalent with the inner membrane in Gram-negative bacteria. Detailed attempts to predict archaeobacterial proteins are few and they seem to have been practically treated as a kind of Gram-positive bacterial protein.

Since the N-terminal part of a protein can affect its localization significantly, the input amino acid sequence must be the accurate precursor sequence. However, it is often the case that the translation site of a protein is not experimentally determined. Moreover, it is known that the translation site can be switched in various conditions because of alternative choices of

transcriptional start sites and/or alternative splicing. Thus, predicting the potential changes of localization sites depending on the choices of translational start sites would be a future direction in this field (see the concluding remarks).

In many cases, the most probable localization site or a list of possible sites with p values are provided in the prediction. Recently, there have also been many attempts to extend the scheme to treat with proteins that are localized at multiple sites. However, objective evaluation of their performances is difficult because both systematic and quantitative data of multiple localization have been quite rare, if any.

Conventionally, the prediction accuracy of a prediction method is assessed using the n -fold cross validation test, including the jack-knife test, where the number of testing data is one. As for the measure of prediction performance, standard measures in machine learning, such as the overall accuracy, the sensitivity/specificity (or precision/recall), and the Matthew's correlation coefficient are often used.

General prediction methods

Many prediction methods of subcellular localization sites have been developed and improved by combining biological or empirical sequence features correlated with subcellular localization with a variety of machine-learning algorithms, such as the k -nearest neighbor classifier, the support vector machine, and deep learning: PSORT (Nakai and Kanehisa, 1992), CELLO 2.5 (Yu *et al.*, 2006), WoLF PSORT (Horton *et al.*, 2007), MultiLoc2 (Blum *et al.*, 2009), SherLoc2 (Briesemeister *et al.*, 2009), YLoc (Briesemeister *et al.*, 2010), iLoc-Euk (Chou *et al.*, 2011), Loctree3 (Goldberg *et al.*, 2014), and DeepLoc (Almagro Armenteros *et al.*, 2017).

Prediction features generally used in the prediction of subcellular localization sites are roughly categorized into three types: Targeting signal features, sequence features, and annotation-based features. The information of targeting signals gives powerful prediction features and thus is detailed in the next section. The targeting signals are roughly classified into two classes: N-terminal targeting signals and non-N-terminal targeting signals. Representative N-terminal targeting signals are the signal sequence for the secretory pathway, the chloroplastic targeting signal, and the mitochondrial targeting signal. Whereas, the nuclear localization signal (NLS) and the nuclear export signal (NES) are usually located in the internal position of protein sequences, and peroxisomal targeting signal 1 (PTS1) is located at the C-terminus of proteins.

Sequence features are heavily used in localization prediction since some differences in the sequence features are empirically known to be correlated with different localization sites. Note that these sequence features may not be directly related to protein targeting itself. Representative sequence features are a series of expanded amino acid compositions of the entire sequence, such as the content of dipeptides, n -grams, k -mers, and the pseudo amino acid composition. Pseudo amino acid composition is reported to be more informative in terms of incorporating sequence-order information of a protein sequence (Chou, 2001). Functional motifs are also used in the prediction because some sequence motifs are directly related to the targeting signals and some functional information is closely related to the function of a localization site (e.g., the DNA-binding motif).

Typical annotation-based features are experimentally verified localization information found in UniProt or GO terms, in addition to the information of functional domains, protein-protein interaction and the text information from PubMed abstracts. Those annotation-based features are transferred from homologous proteins or the protein itself. In those features, the transfer of localization annotation from homologous proteins seems to be practically useful. Indeed, a simple homology-based inference outperforms methods based on machine learnings (Imai and Nakai, 2010) if a homologous protein with localization annotation is available.

Evaluation of prediction performance is a difficult issue in subcellular localization prediction. There is often some overlap between the training and the test sets of different methods since most methods use UniProt annotation for making these datasets. To evaluate the prediction performance with less bias, Salvatore *et al.* (2017) recently made a benchmark dataset from new data obtained by recent large-scale experimental studies in human cells and examined the performance of six state-of-the-art methods (CELLO 2.5, LocTree2, MultiLoc2, SherLoc2, WoLF PSORT and YLoc) (Salvatore *et al.*, 2017). In this assessment, CELLO 2.5, SherLoc2, LocTree2, and YLoc showed better performance (F_1 score=0.70), where F_1 score=2((precision $\times$ recall)/(precision + recall)). Also they developed an ensemble method, SubCons that combines four predictors (CELLO2.5, LocTree2, MultiLoc2 and SherLoc2) using a Random Forest classifier and it outperformed the six methods (F_1 score=0.79).

Targeting Signal Prediction

Prediction of signal sequence

Signal sequences (signal peptides) are the N-terminal sorting signal that targets the linked protein to the secretory pathway in eukaryotes and prokaryotes. About 10%–20% of eukaryotic proteome and 10% of bacterial proteome have been estimated to have the signal sequence (Kanapin *et al.*, 2003; Ivankov *et al.*, 2013). In eukaryotic cells, the signal recognition particle (SRP) cotranslationally recognizes the signal sequence upon its emergence from the ribosome and transfers it to the Sec61 translocon in the ER membrane (Nilsson *et al.*, 2015) via the SRP receptor. In bacterial cells, the SRP pathway is mainly used for targeting the membrane proteins with noncleaved signal sequence (signal anchors). Instead, secretory proteins containing the signal sequence are recognized by SecA, which drives the translocation through the SecYEG translocon (Kudva *et al.*, 2013). The signal peptidase cleaves signal sequences and the mature proteins are generated. Signal sequences do not share sequence similarity but share some characteristic features (von Heijne, 1990): Signal sequences are on average 16–30 amino acid residues in length and

comprise characteristic tripartite architecture: A positively charged *n*-region, a central hydrophobic *h*-region and a *c*-region with the cleavage site for signal peptidases. The cleavage site has a consensus pattern known as the -1,-3 rule: Amino acids with small side chains are preferred at the -1 position of signal sequences and no charged amino acid residues are preferred at their -3 position.

For predicting the signal sequence and its cleavage site, many prediction methods, such as SignalP (Bendtsen *et al.*, 2004; Petersen *et al.*, 2011), Signal-CF (Chou and Shen, 2007), and Signal-BLAST (Frank and Sippl, 2008), have been developed with the characteristic features. According to an assessment study in 2009 (Choo *et al.*, 2009), SignalP 3.0 achieves the best overall accuracy (0.872–0.914) in eukaryotic, Gram-positive, and Gram-negative bacterial data sets. Therefore, the discrimination between secretory and nonsecretory proteins (not including membrane proteins) based on the signal sequence prediction is the most successful in targeting signal predictions. However, those methods share a problem: Difficulty in the discrimination between the signal sequence and the transmembrane region. This problem will yield many false-positive predictions from N-terminal transmembrane regions when the methods are applied in proteomic analyses. Recently, SignalP (SignalP 4.0) was updated to tackle the problem using two types of neural network-based methods, SignalP-TM and SignalP-noTM: SignalP-TM has been trained with sequences containing transmembrane segments in the dataset while SignalP-noTM has been trained without those sequences (Petersen *et al.*, 2011). SignalP 4.0 shows better discrimination between signal peptides and transmembrane regions, and consequently achieves the best signal sequence prediction. On the other hand, there is still room for improvement on the cleavage site prediction: Precision and sensitivity of current methods hovers around ~66% and ~68%, respectively.

Prediction of mitochondrial targeting signal

Mitochondria have been estimated to host 1000–1500 distinct proteins, respectively (Meisinger *et al.*, 2008). The vast majority of mitochondrial proteins are encoded in the nuclear genome and imported by the translocases in the mitochondrial outer and inner membranes. About a half of mitochondrial proteins possess an N-terminal cleavable targeting signal (presequence), while the remaining proteins are thought to have a noncleavable internal targeting signal (Chacinska *et al.*, 2009). The mechanism of presequence-dependent targeting and the features of the presequence are well studied, however, the targeting mechanism with the internal targeting signal is still unclear. Thus, the main target of mitochondrial targeting signal prediction has been based on the presequences. Presequences-containing proteins are translocated by the outer and the inner membrane translocase, TOM and TIM23-PAM complexes (Schulz *et al.*, 2015; Wiedemann and Pfanner, 2017). Presequences have a length of 20–60 amino acid residues (Calvo *et al.*, 2017) and do not share sequence similarity. However, they exhibit a few characteristic features: A high composition of arginine and near absence of negatively charged residues (von Heijne, 1986; Schneider *et al.*, 1998). They are also often capable of forming a local amphiphilic α -helical structure with hydrophobic residues on one face and positively charged residues on the opposite face (Chacinska *et al.*, 2009; Fukasawa *et al.*, 2015). The presequence is cleaved in the matrix by the mitochondrial processing peptidase (MPP) and some of them are subsequently further cleaved by the intermediate peptidases, such as Oct1 and Icp55 (Mossmann *et al.*, 2012). The cleavage by MPP occurs at a position that is two amino acids C-terminal from an arginine (the R-2 motif). Icp55 and Oct1 subsequently cleave one amino acid and eight amino acids, respectively. Therefore, proteins cleaved by MPP + Icp55 have an arginine at position -3 (the R-3 motif) in the presequence while proteins cleaved by MPP + Oct1 have an arginine at position -10 (the R-10 motif).

Widely used presequence prediction methods, such as MitoProtII (Claros, 1995), TargetP (Emanuelsson *et al.*, 2000), Predotar (Small *et al.*, 2004), and MitoFates (Fukasawa *et al.*, 2015) were developed using machine-learning techniques with these properties of the presequences. These tools are also capable of predicting the presequence with its cleavage site. MitoProtII and MitoFates are specific predictors for the (mitochondrial) presequence, while TargetP and Predotar integrate other N-terminal targeting signal predictions, such as the secretory signal sequence and the chloroplastic targeting signal. Among those tools, Mitofates performs better in predicting both the presequence existence and the cleavage site: Precision and recall of 0.79 and 0.80 in the presequence prediction and the sensitivity of cleavage site detection is 71% (Fukasawa *et al.*, 2015). However, proteomic analyses of N-termini of mitochondrial proteins of mouse and yeast point out low accuracy (especially in the cleavage site prediction) of those presequence prediction tools (Vögtle *et al.*, 2009; Calvo *et al.*, 2017). A cluster analysis of yeast presequence data obtained from the proteomic analysis with features used in the presequence prediction suggests that presequences can be grouped into at least three clusters (Fukasawa *et al.*, 2015). The largest cluster (60% of presequences) shows typical presequence features such as the enrichment of arginines, almost no negatively charged residues, positively charged amphiphilicity, and the MPP cleavage pattern. On the other hand, the two remaining clusters differ from the classical view of presequences: Many of them have low net-charge and their cleavage sites did not match to any known patterns. Presequences belonging to these clusters should be the reason of the low prediction accuracy. To further improve the presequence prediction, it will be necessary to better characterize these untypical presequences.

Prediction of chloroplastic targeting signal

Approximately 3000 different proteins are estimated to be needed to develop a chloroplast. Similar to mitochondria, the vast majority of those chloroplast proteins are nuclear-encoded and imported into the chloroplast. Most chloroplast precursor proteins have a cleavable N-terminal extension, the chloroplastic targeting signal (transit peptide) (Li and Chiu, 2010; Paila *et al.*, 2015). Thus, existing chloroplastic targeting signal prediction tools deal with the cleavable N-terminal transit peptides. Precursor proteins containing the transit peptide are imported through the translocase in the outer and inner envelope membranes, the TOC and TIC complex, respectively. Similar to other N-terminal targeting signals, the transit peptide is cleaved off by the stroma processing peptidase during or after the translocation. Chloroplastic transit peptides exhibit a few characteristic features: A high content of

hydroxylated amino acids (e.g., serines), lack of negatively charged residues, and a propensity to form α -helical structures in hydrophobic environments (Bruce, 2001; Jarvis, 2008). In this respect, transit peptides are similar to mitochondrial presequences. Their cleavage sites are characterized by the higher frequency of Ala, Ile, Cys, and Val residues. A loosely conserved consensus motif ([V,I]X[A,C] ↓ A, where ↓ represents the cleavage site) was previously found from a limited set of 32 cleavage sites (Gavel and von Heijne, 1990). A more recent 198 cleavage site set gives three motifs, [V,I][R,A] ↓ [A,C]AAE, S[V,I][R,S,V] ↓ [C,A]A, and [A,V]N ↓ A [A,M]AG[E,D] (Savoijardo *et al.*, 2015).

A widely used prediction method for the transit peptide and its cleavage site is ChloroP (Emanuelsson *et al.*, 1999). However, predictors that are specialized for transit peptides are minor and the transit peptide prediction is usually integrated as a part of the prediction of N-terminal targeting signals: TargetP (Emanuelsson *et al.*, 2000), iPSORT (Bannai *et al.*, 2002), Predotar (Small *et al.*, 2004), and TPpred3 (Savoijardo *et al.*, 2015) are capable of predicting transit peptides. Indeed, TargetP includes ChloroP in its algorithm. Among those methods, TPpred3 achieves better performance for transit peptide prediction (46% precision and 64% recall) (Savoijardo *et al.*, 2015), although further improvement seems to be needed for practical purposes. Comparing with mitochondrial targeting signals, the data size of known transit peptides is smaller. Larger sets of established N-terminal proteomics data should be useful to further improve the prediction performance.

Prediction of nuclear localization signals and nuclear export signals

Nuclear proteins are carried into and out of the nuclei through the nuclear pore complex by nucleocytoplasmic transport receptors, that is, the importin (karyopherin)- β family. The human genome encodes 20 importin- β family proteins, of which 10 are nuclear import receptors (importin- β , transportin-1, -2, -SR, importin-4, -5 (RanBP5), -7, -8, -9, and -11), 7 are export receptors (exportin-1 (CRM1), -2 (CAS/CSE1L), -5, -6, -7, -t, and RanBP17), 2 are bidirectional receptors (importin-13 and exportin-4), and the function of RanBP6 is undetermined (Kimura and Imamoto, 2014). Those nucleocytoplasmic transport receptors are thought to recognize specific targeting signals on those cargo proteins. Nevertheless the targeting signals have been identified for only a few nucleocytoplasmic transport receptors, so far (Soniat and Chook, 2015): The classical nuclear localization signal (cNLS) for the importin- α adaptor, which links cargos and importin β (Lange *et al.*, 2007), PY-NLS for transportin-1 and -2 (Lee *et al.*, 2006), the NES for exportin-1/CRM1 (Hutten and Kehlenbach, 2007), and the SR-rich domain that binds to transportin-SR (Maertens *et al.*, 2014). Among those nuclear targeting signals, cNLS and NES are well characterized. Thus, recent prediction algorithms for NLSs and NESs mainly target the two types of the signals.

cNLSs include two types of signals, the monopartite and the bipartite NLSs: The monopartite NLS means a single cluster of basic amino acid residues (e.g., KR[K/R]R and K[K/R]RK) and the bipartite NLSs means two clusters of basic amino acids separated by a 10–12 amino acid linker (e.g., KRX_{10–12}K[K/R][K/R]) (Kosugi *et al.*, 2009). NucPred (Brameier Krings and MacCallum, 2007), cNLSmapper (Kosugi *et al.*, 2009), NLSstradamus (Ba *et al.*, 2009), NucImport (Mehdi *et al.*, 2011), and seqNLS (Lin and Hu, 2013) are widely used for the cNLS prediction. According to a recent assessment based on experimentally identified human NLSs (77% of them are classified as known cNLSs) (Lisitsyna *et al.*, 2017), the highest Matthews' correlation coefficient (~ 0.3) was obtained from NucPred, seqNLS, and NLSstradamus. However, those prediction methods correctly identified only $\sim 45\%$ of the human NLS data, suggesting the necessity for further improvement on the NLS prediction. Recent proteomic analysis for the identification of cargo proteins of 12 nucleocytoplasmic transport receptors (10 nuclear import receptors and 2 bidirectional receptors) (Kimura *et al.*, 2017) would be a valuable resource for further improvement of NLS prediction and/or for finding novel NLSs. Meanwhile, the proteomic analysis pointed out that about 30% of the identified cargos are shared by multiple receptors. Better characterization of such multiple recognitions may be fruitful for future improvements.

Nuclear export receptor exportin-1/CRM1 binds to 8–15 residue-long NESs in hundreds of distinct cargos. NES sequences are diverse and 11 patterns were defined by a peptide library-based study and structural analyses of exportin-1/CRM1-NES (Kosugi *et al.*, 2008; Fung *et al.*, 2015, 2017). Those NESs usually have 4–5 hydrophobic residues, which bind to the hydrophobic pocket of the receptors. The hydrophobic residues are arranged in various patterns: $\Phi X_3 \Phi X_2 \Phi X \Phi$, $\Phi X \Phi X_2 \Phi X \Phi$ and $\Phi X_2 \Phi X_3 \Phi X_2 \Phi$ (Φ represents Leu, Val, Ile, Phe, or Met). Several computational tools, such as NetNES (La Cour *et al.*, 2004), NESsential (Fu *et al.*, 2011), NESmapper (Kosugi *et al.*, 2014), LocNES (Xu *et al.*, 2014), and Wregex-based prediction (Prieto *et al.*, 2014) have been developed to predict NESs, based on the sequence profiles of NESs and some biophysical properties (e.g., disorder propensity and solvent accessibilities). However, the performance of those predictors is still not enough (precision is $\sim 50\%$ with 20% recall). The diverse NES patterns and the low performance of predictors suggest that there may be no fixed hydrophobic pattern in NESs. There may be a limitation to predict NESs from only sequence. Combination with structure-based predictions might help to find a breakthrough.

Other signals

Besides the above signals the predictors of which are relatively well developed, there are also known signals that may be used for the prediction. Here, we briefly sketch several of them.

As noted above, peroxisomes are often not included in the prediction repertory because known peroxisomal proteins are not so many. However, two kinds of signals, namely PTS1 and PTS2, are known to exist. PTS1 is characterized by the presence of the SKL motif (the more precise consensus is [SAC][KRH][LM]) in the C-terminus of proteins and is recognized by the receptor Pex5, which is a modular protein containing a C-terminal tetratricopeptide repeat domain (Gould *et al.*, 1987; Kim and Hettema, 2015). Of course, its computational recognition is easy. It is also possible to include weaker conserved patterns near PTS1. However, the number of proteins that are sorted with PTS1 in vivo is not enough to make the prediction of the entire peroxisomal proteins

reliable. The other signal, PTS2, is recognized by the receptor Pex7. The signal is characterized as an about nine-residue motif near the N-terminus; the consensus motif is R[LIVQ]XX[LIVQH][LSGA]X[HQ][LA] (Petriv *et al.*, 2004). But its reliable predictors have not been developed because of its small sample size, again.

It has been shown that some lysosomal proteins are recognized from their special modification with mannose-6-phosphate (M6P), allowing their recognition by M6P receptors in the Golgi complex and ensuing their transport to the endosomal/lysosomal system (Braulke and Bonifacio, 2009). However, their sequence determinant is still a mystery and thus there have not been effective predictors for lysosomal proteins, so far.

Similarly, some other posttranslational modifications can affect the localization of proteins significantly. Typical examples are lipid anchors, where attached lipid moiety is inserted into the membrane as an anchor. Three types of lipid anchors are known: N-myristoylation, glycosylphosphatidylinositol (GPI) attachment, and prenylation. In N-myristoylation, a 14-carbon fatty acid is attached to an N-terminal Gly residue (the consensus motif is MGXXX[ST]) (Johnson *et al.*, 1994). GPI is covalently attached to the C-terminus of proteins (Ferguson *et al.*, 2009). In prenylation, a 15-carbon (farnesylation), or a 20-carbon (geranylgeranylation) lipid is attached to a cysteine by farnesyltransferases or by protein geranylgeranyl transferases I, respectively (the consensus motif is CAAX at the C-terminus) (Fu and Casey, 1999). For each of them, prediction algorithms, such as NMT (Maurer-Stroh *et al.*, 2002), PredGPI (Pierleoni *et al.*, 2008), GPS-Lipid (Xie *et al.*, 2016), have been developed (Audagnotto and Dal Peraro, 2017). However, distinctions on which membrane the predicted lipid will be targeted might be necessary in the future when enough data for each membrane are accumulated.

For the prediction of the localization membrane for (integral) membrane proteins, the prediction of their membrane topology is also important because it is known that some important signals can exist on the so-called cytoplasmic tail (i.e., the terminal region that is protruded to the cytoplasmic side with a membrane) of membrane proteins. For the prediction of membrane topology itself, please refer to its specific article. Here, we would like to point out just two things: The prediction of membrane topology can be related to the prediction of the cleavage of N-terminal hydrophobic segment because N-terminal transmembrane segments are often wrongly predicted as cleavable signal peptides, as explained above. Second, the most important localization signal existing on the C-terminus of a protein, which also exists as a part of the cytoplasmic tail, is known as the KDEL signal, which works as a retention signal to the ER (Munro and Pelham, 1987). Again, its computational recognition is not so difficult, though there are some differences in the optimum motif between organisms, such as HDEL in yeasts. However, the distinction of ER membrane proteins from the others remains to be a very difficult prediction problem because the KDEL signal is not so universal.

For some of the proteins localized at relatively minor sites, such as the Golgi body and the cell wall (in yeasts and plants), there have been several reports on their potential signals (Banfield, 2011; Mao *et al.*, 2008) but their known examples are too few to be tested in the scheme of machine learning.

Concluding Remarks

When publishing a paper on a new prediction method, it is generally required that the method should outperform existing ones. Indeed, many recent papers in the field of protein localization prediction claim that their predictors are apparently quite reliable. However, as far as the amino acid sequence is used as its input, it is questionable if these predictors can be practically useful for totally new proteins. It seems to be a fundamental problem that we do not know how many nuclear DNA-encoded proteins are sorted with their own amino acid sequence signals. Thus, perhaps, a promising new direction in this field would be to use prediction tools to evaluate the possibility of whether each protein is sorted with a classical pathway or not. On the other hand, it is certain that the practical importance of subcellular localization prediction would become even less with the future accumulation of more systematic and precise proteomic data. Then, what new directions will be pursued? As mentioned above, one possibility is to develop a more precise predictor that can predict the semiquantitative distribution between multiple sites. To do this, more comprehensive data for localization are necessary. Although this is not the scope of this article, the prediction of subcellular localization for RNAs would also be an interesting frontier. Finally, we believe that the field should be extended to more general goals. Three possibilities in this direction would be (1) to explore the possibility of predicting differential localization of a protein between different cell types/conditions, incorporating the information of protein isoform expression between cells/conditions; (2) to explore the possibility of more general prediction of protein *in vivo* fates, including the predictions of more general post-translational modifications and degradations; and (3) to explore the possibility to develop predictors for artificial proteins, incorporating a vast amount of synthetic biological data. Such predictors will have some practical value for designing new drugs, for example, that are optimized to be targeted to a certain localization site.

See also: Algorithms for Strings and Sequences: Searching Motifs. Artificial Intelligence and Machine Learning in Bioinformatics. Artificial Intelligence. Cell Modeling and Simulation. Data Mining: Classification and Prediction. Data Mining: Prediction Methods. Epitope Predictions. Functional Genomics. Machine Learning in Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Protein Functional Annotation. Protein Post-Translational Modification Prediction. Protein Properties. Protein-Protein Interaction Databases. Sequence Analysis. Sequence Composition. Supervised Learning: Classification. The Evolution of Protein Family Databases. Transmembrane Domain Prediction

References

- Almagro Armenteros, J.J., Sønderby, C.K., Sønderby, S.K., Nielsen, H., Winther, O., 2017. DeepLoc: Prediction of protein subcellular localization using deep learning. *Bioinformatics* 33 (21), 3387–3395.
- Audagnotto, M., Dal Peraro, M., 2017. Protein post-translational modifications: In silico prediction tools and molecular modeling. *Computational and Structural Biotechnology Journal* 15, 307–319.
- Ba, A.N.N., Pogoutse, A., Provart, N., Moses, A.M., 2009. NLStradamus: A simple Hidden Markov Model for nuclear localization signal prediction. *BMC Bioinformatics* 10 (1), 202.
- Banfield, D.K., 2011. Mechanisms of protein retention in the Golgi. *Cold Spring Harbor Perspectives in Biology* 3 (8), a005264.
- Bannai, H., Tamada, Y., Maruyama, O., Nakai, K., Miyano, S., 2002. Extensive feature detection of N-terminal protein sorting signals. *Bioinformatics* 18 (2), 298–305.
- Bendtsen, J.D., Nielsen, H., von Heijne, G., Brunak, S., 2004. Improved prediction of signal peptides: SignalP 3.0. *Journal of Molecular Biology* 340 (4), 783–795.
- Blum, T., Briesemeister, S., Kohlbacher, O., 2009. MultiLoc2: Integrating phylogeny and Gene Ontology terms improves subcellular protein localization prediction. *BMC Bioinformatics* 10 (1), 274.
- Brameier, M., Krings, A., MacCallum, R.M., 2007. NucPred – Predicting nuclear localization of proteins. *Bioinformatics* 23 (9), 1159–1160.
- Braulke, T., Bonifacio, J.S., 2009. Sorting of lysosomal proteins. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1793 (4), 605–614.
- Briesemeister, S., Blum, T., Brady, S., *et al.*, 2009. SherLoc2: A high-accuracy hybrid method for predicting subcellular localization of proteins. *Journal of Proteome Research* 8 (11), 5363–5366.
- Briesemeister, S., Rahnenführer, J., Kohlbacher, O., 2010. YLoc – An interpretable web server for predicting subcellular localization. *Nucleic Acids Research* 38 (Suppl_2), W497–W502.
- Bruce, B.D., 2001. The paradox of plastid transit peptides: Conservation of function despite divergence in primary structure. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* 1541 (1–2), 2–21.
- Calvo, S.E., Julien, O., Clauser, K.R., *et al.*, 2017. Comparative analysis of mitochondrial N-termini from mouse, human, and yeast. *Molecular and Cellular Proteomics* 16 (4), 512–523.
- Chacinska, A., Koehler, C.M., Milenkovic, D., Lithgow, T., Pfanner, N., 2009. Importing mitochondrial proteins: Machineries and mechanisms. *Cell* 138 (4), 628–644.
- Choo, K.H., Tan, T.W., Ranganathan, S., 2009. A comprehensive assessment of N-terminal signal peptides prediction methods. *BMC Bioinformatics* 10 (Suppl. 5), S2.
- Chou, K.C., 2001. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics* 43 (3), 246–255.
- Chou, K.C., Shen, H.B., 2007. Signal CF: A subsite-coupled and window-fusing approach for predicting signal peptides. *Biochemical and Biophysical Research Communications* 357 (3), 633–640.
- Chou, K.C., Wu, Z.C., Xiao, X., 2011. iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PLOS ONE* 6 (3), e18258.
- Claros, M.G., 1995. MitoProt, a Macintosh application for studying mitochondrial proteins. *Bioinformatics* 11 (4), 441–447.
- Emanuelsson, O., Nielsen, H., Brunak, S., Von Heijne, G., 2000. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *Journal of Molecular Biology* 300 (4), 1005–1016.
- Emanuelsson, O., Nielsen, H., Von Heijne, G., 1999. ChloroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Science* 8 (5), 978–984.
- Ferguson, M.A., Kinoshita, T., Hart, G.W., 2009. *Glycosylphosphatidylinositol Anchors*. NY, United States: Cold Spring Harbor Laboratory Press.
- Frank, K., Sippl, M.J., 2008. High-performance signal peptide prediction based on sequence alignment techniques. *Bioinformatics* 24 (19), 2172–2176.
- Fu, H.W., Casey, P.J., 1999. Enzymology and biology of CaaX protein prenylation. *Recent Progress in Hormone Research* 54, 315–342.
- Fukasawa, Y., Tsuji, J., Fu, S.C., *et al.*, 2015. MitoFates: Improved prediction of mitochondrial targeting sequences and their cleavage sites. *Molecular & Cellular Proteomics* 14 (4), 1113–1126.
- Fung, H.Y.J., Fu, S.C., Brautigam, C.A., Chook, Y.M., 2015. Structural determinants of nuclear export signal orientation in binding to exportin CRM1. *eLife* 4, e10034.
- Fung, H.Y.J., Fu, S.C., Chook, Y.M., 2017. Nuclear export receptor CRM1 recognizes diverse conformations in nuclear export signals. *eLife* 6. doi:10.7554/eLife.23961.
- Fu, S.C., Imai, K., Horton, P., 2011. Prediction of leucine-rich nuclear export signal containing proteins with NESsential. *Nucleic Acids Research* 39 (16), e111.
- Gavel, Y., von Heijne, G., 1990. A conserved cleavage-site motif in chloroplast transit peptides. *FEBS Letters* 261 (2), 455–458.
- Goldberg, T., Hecht, M., Hamp, T., *et al.*, 2014. LocTree3 prediction of localization. *Nucleic Acids Research* 42 (W1), W350–W355.
- Gould, S.G., Keller, G.A., Subramani, S., 1987. Identification of a peroxisomal targeting signal at the carboxy terminus of firefly luciferase. *The Journal of cell biology* 105 (6), 2923–2931.
- Horton, P., Park, K.J., Obayashi, T., *et al.*, 2007. WoLF PSORT: Protein localization predictor. *Nucleic Acids Research* 35 (Suppl_2), W585–W587.
- Hutten, S., Kehlenbach, R.H., 2007. CRM1-mediated nuclear export: To the pore and beyond. *Trends in Cell Biology* 17 (4), 193–201.
- Imai, K., Nakai, K., 2010. Prediction of subcellular locations of proteins: Where to proceed? *Proteomics* 10 (22), 3970–3983.
- Ivanov, D.N., Payne, S.H., Galperin, M.Y., *et al.*, 2013. How many signal peptides are there in bacteria? *Environmental Microbiology* 15 (4), 983–990.
- Jarvis, P., 2008. Targeting of nucleus-encoded proteins to chloroplasts in plants. *New Phytologist* 179 (2), 257–285.
- Johnson, D.R., Bhattacharya, R.S., Knoll, L.J., Gordon, J.I., 1994. Genetic and biochemical studies of protein N-myristoylation. *Annual Review of Biochemistry* 63 (1), 869–914.
- Kanapin, A., Batalov, S., Davis, M.J., *et al.*, 2003. Mouse proteome analysis. *Genome Research* 13 (6b), 1335–1344.
- Kim, P.K., Hettema, E.H., 2015. Multiple pathways for protein transport to peroxisomes. *Journal of molecular biology* 427 (6), 1176–1190.
- Kimura, M., Imamoto, N., 2014. Biological significance of the importin- β family-dependent nucleocytoplasmic transport pathways. *Traffic* 15 (7), 727–748.
- Kimura, M., Morinaka, Y., Imai, K., *et al.*, 2017. Extensive cargo identification reveals distinct biological roles of the 12 importin pathways. *eLife* 6, e21184.
- Kosugi, S., Hasebe, M., Matsumura, N., *et al.*, 2009. Six classes of nuclear localization signals specific to different binding grooves of importin α . *Journal of Biological Chemistry* 284 (1), 478–485.
- Kosugi, S., Hasebe, M., Tomita, M., Yanagawa, H., 2008. Nuclear export signal consensus sequences defined using a localization-based yeast selection system. *Traffic* 9 (12), 2053–2062.
- Kosugi, S., Hasebe, M., Tomita, M., Yanagawa, H., 2009. Systematic identification of cell cycle-dependent yeast nucleocytoplasmic shuttling proteins by prediction of composite motifs. *Proceedings of the National Academy of Sciences* 106 (25), 10171–10176.
- Kosugi, S., Yanagawa, H., Terauchi, R., Tabata, S., 2014. NESmapper: Accurate prediction of leucine-rich nuclear export signals using activity-based profiles. *PLOS Computational Biology* 10 (9), e1003841.
- Kudva, R., Denks, K., Kuhn, P., *et al.*, 2013. Protein translocation across the inner membrane of Gram-negative bacteria: The Sec and Tat dependent protein transport pathways. *Research in Microbiology* 164 (6), 505–534.
- La Cour, T., Kierner, L., Mølgaard, A., *et al.*, 2004. Analysis and prediction of leucine-rich nuclear export signals. *Protein Engineering Design and Selection* 17 (6), 527–536.
- Lange, A., Mills, R.E., Lange, C.J., *et al.*, 2007. Classical nuclear localization signals: Definition, function, and interaction with importin α . *Journal of Biological Chemistry* 282 (8), 5101–5105.
- Lee, B.J., Cansizoglu, A.E., Süel, K.E., *et al.*, 2006. Rules for nuclear localization sequence recognition by karyopherin β 2. *Cell* 126 (3), 543–558.
- Lin, J.R., Hu, J., 2013. SeqNLS: Nuclear localization signal prediction based on frequent pattern mining and linear motif scoring. *PLOS ONE* 8 (10), e76864.

- Lisitsyna, O.M., Septyarskiy, V.B., Sheval, E.V., 2017. Comparative analysis of nuclear localization signal (NLS) prediction methods. *Biopolymers and Cell* 33, 147–154.
- Li, H.M., Chiu, C.C., 2010. Protein transport into chloroplasts. *Annual Review of Plant Biology* 61, 157–180.
- Maertens, G.N., Cook, N.J., Wang, W., *et al.*, 2014. Structural basis for nuclear import of splicing factors by human Transportin 3. *Proceedings of the National Academy of Sciences* 111 (7), 2728–2733.
- Mao, Y., Zhang, Z., Gast, C., Wong, B., 2008. C-terminal signals regulate targeting of glycosylphosphatidylinositol-anchored proteins to the cell wall or plasma membrane in *Candida albicans*. *Eukaryotic Cell* 7 (11), 1906–1915.
- Maurer-Stroh, S., Eisenhaber, B., Eisenhaber, F., 2002. N-terminal N-myristoylation of proteins: Prediction of substrate proteins from amino acid sequence1. *Journal of Molecular Biology* 317 (4), 541–557.
- Mehdi, A.M., Sehgal, M.S.B., Kobe, B., Bailey, T.L., Bodén, M., 2011. A probabilistic model of nuclear import of proteins. *Bioinformatics* 27 (9), 1239–1246.
- Meisinger, C., Sickmann, A., Pfanner, N., 2008. The mitochondrial proteome: From inventory to function. *Cell* 134 (1), 22–24.
- Mossmann, D., Meisinger, C., Vögtle, F.N., 2012. Processing of mitochondrial presequences. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms* 1819 (9), 1098–1106.
- Munro, S., Pelham, H.R., 1987. A C-terminal signal prevents secretion of luminal ER proteins. *Cell* 48 (5), 899–907.
- Nakai, K., Kanehisa, M., 1992. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics* 14 (4), 897–911.
- Nilsson, I., Lara, P., Hessa, T., *et al.*, 2015. The code for directing proteins for translocation across ER membrane: SRP cotranslationally recognizes specific features of a signal sequence. *Journal of Molecular Biology* 427 (6), 1191–1201.
- Paila, Y.D., Richardson, L.G., Schnell, D.J., 2015. New insights into the mechanism of chloroplast protein import and its integration with protein quality control, organelle biogenesis and development. *Journal of Molecular Biology* 427 (5), 1038–1060.
- Petersen, T.N., Brunak, S., von Heijne, G., Nielsen, H., 2011. SignalP 4.0: Discriminating signal peptides from transmembrane regions. *Nature Methods* 8 (10), 785.
- Petriv, I., Tang, L., Titorenko, V.I., *et al.*, 2004. A new definition for the consensus sequence of the peroxisome targeting signal type 2. *Journal of molecular biology* 341 (1), 119–134.
- Pierleoni, A., Martelli, P.L., Casadio, R., 2008. PredGPI: A GPI-anchor predictor. *BMC Bioinformatics* 9 (1), 392.
- Prieto, G., Fullaondo, A., Rodríguez, J.A., 2014. Prediction of nuclear export signals using weighted regular expressions (Wregex). *Bioinformatics* 30 (9), 1220–1227.
- Salvatore, M., Warholm, P., Shu, N., Basile, W., Elofsson, A., 2017. SubCons: A new ensemble method for improved human subcellular localization predictions. *Bioinformatics* 33 (16), 2464–2470.
- Savojardo, C., Martelli, P.L., Fariselli, P., Casadio, R., 2015. TPpred3 detects and discriminates mitochondrial and chloroplastic targeting peptides in eukaryotic proteins. *Bioinformatics* 31 (20), 3269–3275.
- Schneider, G., Sjöling, S., Wallin, E., *et al.*, 1998. Feature-extraction from endopeptidase cleavage sites in mitochondrial targeting peptides. *Proteins: Structure, Function, and Bioinformatics* 30 (1), 49–60.
- Schulz, C., Schendzielorz, A., Rehling, P., 2015. Unlocking the presequence import pathway. *Trends in Cell Biology* 25 (5), 265–275.
- Small, I., Peeters, N., Legaei, F., Lurin, C., 2004. Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences. *Proteomics* 4 (6), 1581–1590.
- Soniat, M., Chook, Y.M., 2015. Nuclear localization signals for four distinct karyopherin- β nuclear import systems. *Biochemical Journal* 468 (3), 353–362.
- Vögtle, F.N., Wortelkamp, S., Zahedi, R.P., *et al.*, 2009. Global analysis of the mitochondrial N-proteome identifies a processing peptidase critical for protein stability. *Cell* 139 (2), 428–439.
- von Heijne, G., 1986. Mitochondrial targeting sequences may form amphiphilic helices. *The EMBO Journal* 5 (6), 1335–1342.
- von Heijne, G., 1990. The signal peptide. *The Journal of Membrane Biology* 115 (3), 195–201.
- Wiedemann, N., Pfanner, N., 2017. Mitochondrial machineries for protein import and assembly. *Annual Review of Biochemistry* 86, 685–714.
- Xie, Y., Zheng, Y., Li, H., *et al.*, 2016. GPS-Lipid: A robust tool for the prediction of multiple lipid modification sites. *Scientific Reports* 6, 28249.
- Xu, D., Marquis, K., Pei, J., *et al.*, 2014. LocNES: A computational tool for locating classical NESs in CRM1 cargo proteins. *Bioinformatics* 31 (9), 1357–1365.
- Yu, C.S., Chen, Y.C., Lu, C.H., Hwang, J.K., 2006. Prediction of protein subcellular localization. *Proteins: Structure, Function, and Bioinformatics* 64 (3), 643–651.

Further Reading

- Nakai, K., 2000. Protein sorting signals and prediction of subcellular localization. *Advances in Protein Chemistry* 54, 277–344.
- Nielsen, H., 2017. Predicting subcellular localization of proteins by bioinformatic algorithms. *Current Topics in Microbiology and Immunology* 404, 129–158.

Introduction

Proteomics sets the ambitious challenge of the large-scale determination of gene and cellular function directly at the protein level. As suggested in a Nature Methods Commentary: “Sequencing the human genome was perhaps the easy part, and now making sense of the constantly moving and changing picture of the proteome will require a lot of time, effort and creativity” (Nilsson *et al.*, 2010). Proteomics has thus evolved as a multidisciplinary field relying on both traditional analytical methods and the development of new technologies. Considering that in complex mixtures proteins may span up to 12 orders of magnitude in relative abundance (Angel *et al.*, 2012), reliable profiling of protein expression is obviously not straightforward.

More generally, the elucidation of biological processes requires information on protein abundance, amino acid variations and modifications, as well as interacting partners. Proteomics large-scale studies undertaken to collect this information roughly span four main objectives: (1) identification of all proteins from a cell or tissue creating a catalogue of information; (2) analysis of differential protein expression associated with specific conditions, different cell states and sample treatments; (3) characterisation of proteins by discovering their function, cellular location, posttranslational modifications (PTMs), etc and (4) contribution to understanding protein interaction networks. When met, these are useful to design effective diagnostic techniques (Liu *et al.*, 2013) or means of monitoring plant growth (Baginsky, 2009).

The present article is introductory to Robin (2018) and Bilbao (2018) where details of the relevant proteome databases and tools are provided. This trilogy is intended as explanatory for the common terminology found in the dense and sometimes cryptic data analysis paragraphs of the Material and Methods section of published proteomics studies.

Proteomics Technical Issues Bearing on Data Processing

The major steps leading to the generation of proteomics data involve sample preparation, molecule separation and mass measurement. Despite its paramount importance in the outcome of a proteomics experiment sample preparation cannot be detailed or discussed here. This information is nonetheless accounted for in the “Minimum Information About a Proteomics Experiment” (MIAPE, (Taylor *et al.*, 2007)) standard with which any stored experimental dataset must comply in proteomics data repositories (see Robin, 2018). Techniques for molecule separation and mass measurement are now introduced briefly in relation to the questions raised by their automation.

Separation Techniques

Technical means to separate complex mixtures have been and still are a central concern in physics and chemistry. The physico-chemical properties of biological molecules account for the common use of electrophoresis and liquid chromatography in molecular biology.

The major difference in using one or the other techniques in the proteomics workflow is the position of the protein digestion step in the overall process. Proteins are separated by 2D-gel electrophoresis, excised from the gel and then digested so as to feed the mass spectrometer whereas proteins are digested prior to running the chromatographic separation that is coupled to the mass spectrometer.

2D-gel Electrophoresis and image processing

Under the influence of an electrical field, charged molecules migrate in the direction of the electrode bearing the opposite charge. Because of their varying charges and masses, different molecules will migrate at different speeds and will thus be separated into single fractions. Such a principle defines electrophoresis.

The electrophoretic mobility, which influences the speed of migration, is a significant and characteristic parameter of a charged molecule. The electrophoretic separation of samples is done in a buffer with a precise pH value and a constant ionic strength. Proteins are separated in two dimensions: mass and pI (isoelectric point). On gels they are not visible and various staining methods are used for detection. Proteins are revealed on a 2D-gel as spots of various sizes and intensity that correspond to protein abundance (darker spots for more abundant proteins).

An image capture of a 2D-gel can be analysed by software to assist spot picking by a robot and/or identify differentially expressed proteins. Gel suppliers often support 2D-gel analysis software most of which is licensed. The two commonly used tools are SameSpots (see “Relevant Websites section”) and ImageMaster/Melanie (see “Relevant Websites section”). They have been developed over decades. The first version of SameSpots was called Phoretix 2D (Mahon and Dupree, 2001) and Melanie was popularised in Appel *et al.* (1991). Despite their respective good performance, both tools still stumble on the inherent ambiguity of faded spots. Other algorithmic issues hindering 2D-gel qualitative and quantitative comparison are well explained in Dowsey *et al.* (2010).

Digitalised 2D-gels have been annotated and stored in on-line databases for comparative and benchmarking purposes. Many of these were created in the 1990s and are no longer maintained. The reference database remains World 2D-page (see "Relevant Websites section") and more recently the GelMap portal was initiated for plant species (see "Relevant Websites section").

Although many hundreds and even thousands of proteins can be separated on a 2D-gel, this technique remains less popular in high-throughput settings despite notable attempts such as Dowsey *et al.* (2008). Nonetheless, 2D-electrophoresis cannot be ignored as a useful approach for instance to assess the extent of protein forms in a given sample and tends to be considered as a complement to the popular Liquid Chromatography (LC)-based proteomics (Rogowska-Wrzesinska *et al.*, 2013).

Liquid chromatography and retention time prediction

Liquid Chromatography (LC) utilises a liquid mobile phase to separate the components of a peptide mixture. A liquid medium continuously flows through a chromatographic column and carries the analytes. The chromatographic separation process is based on the difference in the surface interactions of the analyte and eluent molecules. The velocity at which individual molecules in the mixtures move through the column is a function of the physical property of the analytes, the content of the column and the composition of the mobile phase. Chromatograms are output as plots of the retention time (duration from tip to bottom of the column) in the x-axis and compound abundance in the y-axis. The most popular technique in proteomics is the reverse-phase chromatography. Although reproducible, small variations are observable in these plots as a reflection of peptide conformational changes among others.

The hydrophobicity of peptides is highly correlated to retention time. Due to surface interactions, hydrophobic peptides tend to stay longer in the column. However the sole consideration of this property is not enough to accurately predict retention time from the peptide amino acid sequence. The long path to designing an efficient method is well described in Henneman and Palmblad (2013) where the many attempts with mixed results and performance are referred to in a timeline starting in 1951. Furthermore, the comparison of different methods for retention time prediction has often not led to a consensus in the metric to be used in the evaluation of the accuracy of prediction. As very nicely put by Henneman and Palmblad (2013), discussions centred on the (non)-linearity of the model and variations of the tolerance to error are pointless when the need for assigning a confidence interval to a predicted retention time is prevalent. Another excellent review of peptide retention time prediction methods can be found in Moruz and Käll (2017) where the contribution of prediction is shown (i) to improve peptide identification in simple mixtures by matching theoretical and observed masses and retention times (ii) to increase the throughput of proteomics experiments or (iii) to decrease the number of candidate peptides during database searching.

Interestingly, software designed for 2D-gel image analysis was transposed in the 1990s to LC-MS images defined as plots of mass versus retention time mainly for quantitative studies. Algorithmic aspects of the associated software are reviewed in Dowsey *et al.* (2010).

Ion mobility

Ion mobility (IM) is a fast and reproducible technique for separation of molecules based on their size and shape. Molecules having the same mass-to-charge ratio, but different conformational arrangements can be distinguished. IM has been and is used in combination with LC and mass spectrometry (see next section), providing an additional dimension of information with benefits for proteomics analyses (Baker *et al.*, 2015). The following list of software supports IM data processing: Synapter (Bond *et al.*, 2013), LC-IMS-MS FeatureFinder (Crowell *et al.*, 2013), ISOQuant (Distler *et al.*, 2014) and Skyline (Pino *et al.*, 2017).

While IM has shown great utility when coupled with MS for analysis of complex samples, methods for processing the complex data generated still lag behind. In fact, the incorporation of this extra separation dimension requires upgrades and optimisation of existing computational pipelines and the development of new algorithmic strategies to fully exploit the advantages of the technology.

Mass Spectrometry

A mass spectrometer produces charged particles (ions) from the molecules to be analysed, in the case of proteomics, peptide mixtures. Various means are implemented in the spectrometer to exert forces on the charged particles and measure the mass or molecular weight of the analysed substance.

From raw data to peak lists

All mass spectrometers carry out three distinct functions:

1. *Ionisation* for charging peptides; the major two processes are Electrospray Ionisation (ESI) and Matrix-Assisted Laser Desorption Ionisation (MALDI). These techniques are detailed in all basic mass spectrometry (MS) manuals.
2. *Ion analysis* for separating ions by mass; the different analysers in proteomics are Time-Of-Flight (TOF), Quadrupole, Ion Trap, Fourier Transform Ion Cyclotron Resonance (FTICR). These techniques are detailed in all basic MS manuals.
3. *Ion detection* that requires the collection of ions of different masses. The charged ion beam generated by the analyser is directed sequentially (point) or in parallel (array) onto a point/array collector. This causes an electric current the flow of which marks the arrival of successive ions and the magnitude of which marks their abundance. The analogue electrical signal is digitised and processed by a computer.

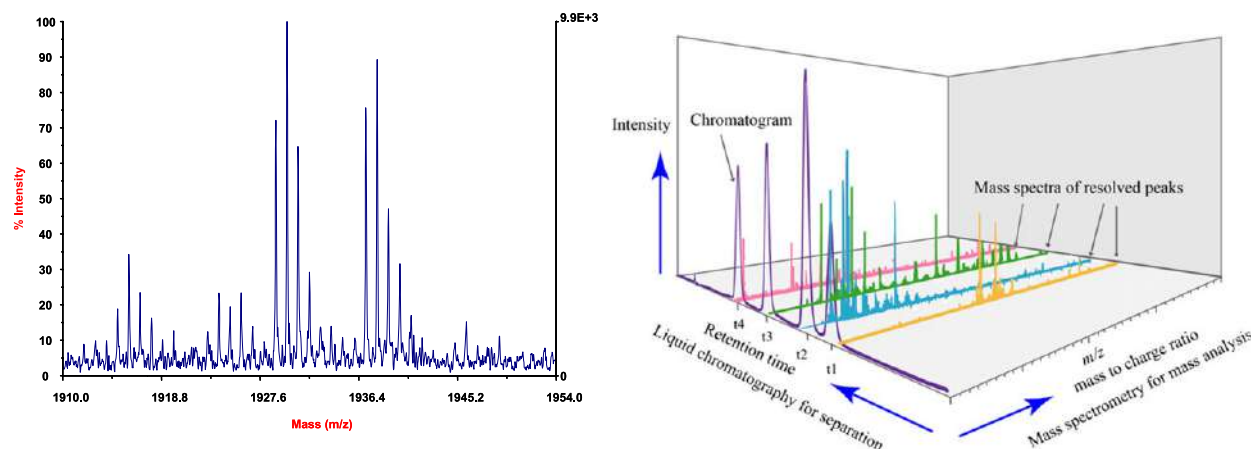


Fig. 1 (a) Typical MALDI-TOF spectrum, (b) Relationship between dimensions m/z , intensity, retention time. Source: https://en.wikipedia.org/wiki/File:Liquid_chromatography_MS_spectrum_3D_analysis.png.

A mass spectrometer outputs raw mass spectra featuring the mass-to-charge ratio (m/z) in the x-axis and the intensity in the y-axis (Fig. 1).

In theory, the mass of each peptide should correspond to a peak in the spectrum. In reality, the imperfection of the sample, its handling and processing, etc., introduce a background noise and cause variations in intensity. The selection of the relevant peaks, that is, those definitely corresponding to peptide masses, is not straightforward. The most intense peak in an isotopic group is not necessarily the monoisotopic mass of a peptide. The intensity ratio of all peaks in this group must be examined. Moreover, a spectrum may contain single- and double- or triple-charged species. Indeed, a population of variably charged ions is generated in the ionisation process. In a given population, peptides may have 1, 2 and 0 sites of protonation. In this case, the intensity of the peaks is a reflection of the population generated in the ionisation process. Multiple-charged ions are generally of low abundance in MALDI but frequent in ESI. At sufficient resolution, a single-charged ion will show isotopic peaks that differ by 1 mass unit, a double-charged ion will show peaks that differ by 0.5 mass units and so on.

Despite these objective difficulties, peak detection is automated. Many solutions are proprietary and integrated in software licensed with an instrument. OpenMS (see “Relevant Websites section”) detailed in Bilbao (2018) is a popular software library implementing a wide range of useful functions for MS data analysis. It includes for instance, peak detection which main task is centroiding. According to the most recent IUPAC recommendations (Murray *et al.*, 2013) centroiding spectra entails calculating the centroid based on the average m/z value weighed by the intensity, and assigning m/z values based on a calibration file (Internal calibration consists in running a well-known sample, detect the signals, and use a correction function to associate the known values to the experimentally measured values. It is a recalibration step that corrects the experimental data with the use of information in the same spectrum. Once the correction is made, the instrument can update its hardware parameters accordingly. Regular calibration of the m/z scale is necessary to maintain accuracy in the instrument.). Only the centroid m/z value and the peak magnitude are stored. This is the basis of peak lists that are input in the various software tools that identify proteins from mass spectra. Another popular open software library for performing basic mass spectra processing converting raw data to peak lists is ProteoWizard (see “Relevant Websites section”). As OpenMS, its major advantage is that it is instrument independent and provides tools to convert vendor proprietary formats to standard and open formats (Robin, 2018; Bilbao, 2018).

Inherent Role of Bioinformatics in Proteomics

In 1993, five independent groups published on a method for protein identification using mass spectrometry (MS) referred to as Peptide Mass Fingerprinting, shortened as PMF (Henzel *et al.*, 1993; James *et al.*, 1993; Mann *et al.*, 1993; Pappin *et al.*, 1993; Yates *et al.*, 1993). Each of these was integrating software to match a list of experimentally observed peptide masses with theoretical peptide masses computed from protein sequences stored in a selected database. This computation is fairly simple given the known monoisotopic mass of each of the 20 amino acids. By definition, the mass of a peptide is the sum of masses of the amino acids it is made of.

This protein identification technology relies heavily on the fast and accurate mapping of mass spectra to protein database entries, and gave birth to a bioinformatics discipline focusing on the development of software tools for the processing of MS data (proteome informatics). Several software tools implementing this *Peptide Mass Fingerprinting* (PMF) identification were developed in the following years such as Perkins *et al.* (1999) the seed of the most popular Mascot software (see Bilbao, 2018).

Soon after, the first publication on another MS based approach for identifying proteins emerged, tandem MS or MS/MS (Eng *et al.*, 1994). In this setup the mass spectrometer performs sequential MS analyses. In a first stage intact peptides are analysed and a mass spectrum is produced. Next, a selection of these peptides is fragmented at the peptide bond by bombarding them with fast-moving atoms. The mass and intensity of each peptide fragment is detected producing a MS/MS spectrum. Each MS/MS spectrum can be mapped to its originating peptide using a similar approach to PMF, where the assembly of peptide fragment masses making up the mass spectrum is screened against theoretical peptide fragment masses. Technically, the workflow can be

fully automated: digested peptide mixtures are run through LC, then upon elution from the chromatographic column the analytes are ionised and directly sprayed into a first mass spectrometer, where they are measured. The resulting spectrum provides the mass to charge (m/z) ratio of the intact analyte ions (precursor ions) that are coming off the LC. The ions of a precursor are selected, and fragment ions are created and measured in a second mass spectrometer to produce the MS/MS spectrum for the analyte. This approach was called *shotgun proteomics* (Wolters *et al.*, 2001) referring to earlier *shotgun genomics*, which led to sequencing the first human genome, as a means to boost the field to the high-throughput level.

Early automation: Shotgun proteomics

From the turn of the 21st century, mass spectrometry combined with liquid chromatography has quickly become a central analytical platform for the high-throughput investigation of complex protein samples (Aebersold and Mann, 2003). MS instrumentation with higher resolving power and mass accuracy, increased sensitivity, alternative fragmentation mechanisms and new data acquisition strategies has boosted the throughput, quality and depth of proteomics data. At the same time, the ability to quickly and consistently analyse the large amounts of experimental data produced in proteomics research has relied on software advances well reviewed in Nesvizhskii (2010) and illustrated in Bilbao (2018).

Over the years, bioinformatics has emerged as a crucial contributor with the development of algorithms, the connection of various tools into pipelines, as well as the deposition and dissemination of both software and data. Currently, the prominence of open source software and public data in reproducible research is emphasised and clearly details in both aspects of software in Bilbao (2018) and public data in Robin (2018).

Fragmentation techniques

Tandem mass spectrometry (abbreviated MS/MS) consists in determining the masses of selected ions. In MS/MS, a precursor or parent ion is mass-selected and fragmented and the masses of the resulting product or daughter ions are analysed. Some peptides are further analysed to determine their precise amino-acid content. The technique requires either two or more analysers in series or relies on the intrinsic characteristics of some spectrometers such as the MALDI-TOF. When several analysers are involved, precursor ions are separated according to their m/z value in the first analyser and selected for fragmentation. Fragments are analysed in a second analyser.

Different types of fragment ions observed in an MS/MS spectrum depend on the peptide primary sequence, the amount of internal energy, how the energy was introduced, the charge state, etc. Fragment ions are named according to an accepted nomenclature first proposed by Roepstorff and Fohlman (1984), and modified by Johnson *et al.* (1987). It is represented in the Fig. 2

The most common ions observed in peptide mass spectra are a , b , and γ . The a type ions occur at a lower frequency and abundance compared to the b and γ type ions. The a - b pairs are often observed in fragment MS/MS spectra. The presence of such pairs can then be used for the detection of b ions.

The most commonly used peptide fragmentation techniques are:

- Collision-Induced Dissociation (CID).
- Electron-Capture Dissociation (ECD).
- Electron-Transfer Dissociation (ETD).

In CID, molecular ions are fragmented in the gas phase (Mitchell Wells and McLuckey, 2005). In this case, the molecular ions are pushed to collide with neutral molecules such as helium, nitrogen or argon. Collision induces kinetic to internal energy transfer. The transfer of energy results in bond breakage so that ions are split into smaller fragments. CID most often leads to the generation of b and γ type ions. Note that Higher-energy Collisional Dissociation (HCD) often cited in the literature is a CID technique specific to a particular type of mass spectrometer, namely the Orbitrap which principle was introduced in Makarov (2000).

In ECD, molecular ions are fragmented by the direct introduction of low energy electrons to trapped gas phase ions (Zubarev *et al.*, 1998). During ECD, the parent ions capture single electrons and fragment simultaneously. In such case, the

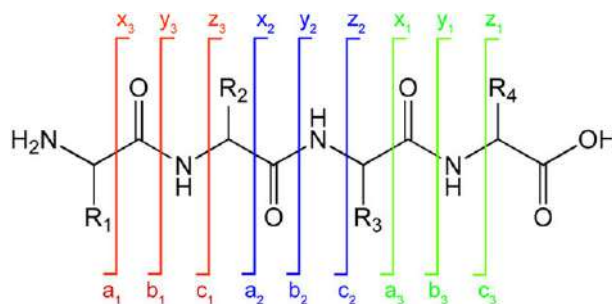


Fig. 2 The a, b, c ions extend from the N-terminus while x, y, z ions extend from the C-terminus. Source: https://commons.wikimedia.org/wiki/File:Peptide_fragmentation.gif

fragmentation occurs also in the positions that are not energetically favoured. This generates more complete ions series, most often *a*, *c* and *z* type ions.

ETD is another technique to fragment molecular ions in the gas phase (Syka *et al.*, 2004). Similar to ECD, ETD induces fragmentation of peptides by transferring electrons to them. ETD cleaves randomly along the peptide backbone while side chains are left intact. The technique works well for higher charge state ions ($> +2$) generating *c* and *z* type ions. ETD is advantageous for the fragmentation of longer peptides or even entire proteins.

Tandem MS provides structural information by establishing relationships between precursor ions and their fragmentation products.

Data Processing

A tandem (MS/MS or MS2) mass spectrum is composed of a precursor peptide and a combination of fragment peaks produced from the fragmentation the peptide sequence. The number of peaks may vary from ten to several hundreds. The peak of a MS/MS spectrum is characterised by the same two values as the MS1 spectrum: mass-to charge ratio (m/z) in its intensity (abundance).

Peptide identification relies on matching peaks from MS/MS spectra with either theoretical or reference experimental spectra. Software for performing this task is detailed in Bilbao (2018). Generally, it first entails the *in silico* digestion of protein sequences into peptides in accordance with the cleavage rules of the protease used in the sample preparation step of the experimental workflow. Then, for each peptide a theoretical spectrum is generated and the calculated peptide fragments are compared with the experimentally observed peaks. A peptide match score is derived which reflects the similarity between the MS/MS spectrum and the theoretical peptide spectrum. This aspect is detailed in the next section.

This type of analysis has been extensively reviewed in the literature where different algorithmic solutions are described in terms of finding the optimal Peptide Spectrum Matches (PSMs). The need to optimise arises from a variety of situations inherent to mass spectrometry. An MS/MS spectrum contains peaks that cannot be assigned to backbone fragments. Possible sources of these peaks include neutral losses, side chain fragments and ions due to multiple backbone fragmentations. Other challenges include ambiguous co-fragmented peptides and frequently missing fragment ions at the beginning or the end of a peptide sequence.

Manually or automatically annotated MS/MS spectra are commonly displaying the *ion ladders* reflecting the ion nomenclature as shown in the figure below where peaks matching an ion series are labelled (Fig. 3).

PSM Scoring and Validation of Protein Inference

A number of different scoring functions have been described in the literature including simple spectral correlation functions such as the dot-product (Liu *et al.*, 2007), more advanced cross correlation functions, scoring functions based on empirically observed rules or statistically derived fragmentation frequencies. The score reported for a given PSM can be on some arbitrary scale or converted to a statistical measure such as a p-value or an expectation value, E-value. The final list of identified peptides is compiled into a protein 'hit list', which is the output of a typical proteomics experiment

Several properties can contribute to scoring the similarity between an experimental spectrum and a reference. The precursor mass is mainly used to restrict the search space. The mass differences for both the matching fragments and the precursor are usually considered in the scoring. Other properties include the number and intensities of the experimental peaks, number of complementary ions, number of missed cleavages and number (and types) of modifications.

A great variety of scoring strategies have been proposed for peptide identification. Broadly speaking, they can be divided into heuristic and probabilistic schemes. Heuristic methods typically combine a set of scoring properties in a practical form reflecting expert knowledge. A product or a linear function are simple examples of how these properties can be combined into a scoring function. Probabilistic schemes compute the probability that a given sequence is the origin of the spectrum.

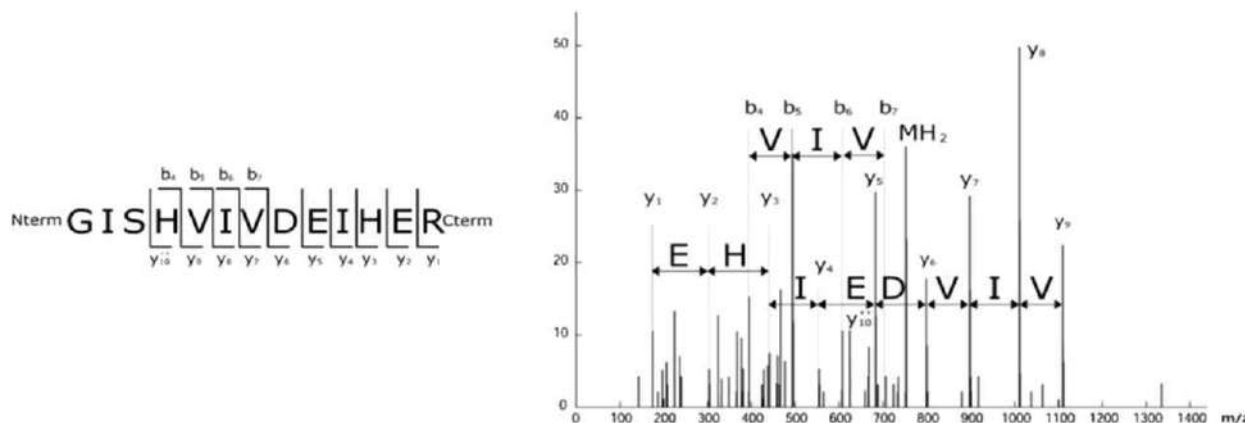


Fig. 3 Example of annotated spectrum that shows the ion ladders (*y* and *b* series) reconstituted for peptide GISHVIVDEIHER. Reproduced from Hernandez, P., *et al.*, 2006. Automated protein identification by tandem mass spectrometry: Issues and strategies. *Mass Spectrom. Rev.* 25 (2), 235–254.

The “target-decoy” approach is commonly used for estimating the false discovery rate (FDR) (Elias and Gygi, 2007). A target-decoy database consists of all sequences to be used in the identification process plus the same amount of decoy sequences. For each sequence entry in the database, a ‘matching’ decoy entry is produced by either reversing the original sequence or by shuffling the letter code. Seemingly counterintuitive, searching against a larger search space decreases the number of reliable identifications because of the higher probability of retrieving false positives.

De Novo Sequencing

De novo sequencing algorithms attempt to derive a peptide sequence directly from the MS/MS spectra by considering the mass difference between pairs of peaks and all possible amino acid combinations (longest ion ladders). The “peptide sequence tag” algorithm was a stepping-stone in the implementation of the identification process (Mann and Wilm, 1994). It is based on extracting short unambiguous amino acid sequences from the spectrum peak pattern by linking two peaks with a mass difference corresponding to the mass of an amino acid. In theory the full peptide sequence can be found using an algorithm based on tag-extraction by finding a tag spanning the full mass range of the spectrum. The approach seemed well suited to solving *de novo* sequencing. However, the accuracy of such algorithms hinges on very high quality spectral data and therefore remains limited to sequencing a small amount of peptides. In practise, sequence tags in combination with the spectrum precursor mass information have been used as a specific probe to identify the peptide origin of the MS/MS spectrum.

Many algorithms have been published whether dependent on dynamic programming, integer linear programming, machine learning and a few other modelling techniques. The coverage, the description and the evaluation of software solutions is provided in Muth and Renard (2017) where all methods that were published were tested. In the end, only a handful remains operational and usable with any type of fragmentation data. Only two of these are open (Frank and Pevzner, 2005; Ma, 2015). Interestingly, the most convincing application of this software so far is antibody sequencing (Tran *et al.*, 2016; Guthals *et al.*, 2017) which brings out sequence polymorphism independently of DNA sequencing. Such an alternative method is helpful since an individual antibody repertoire cannot be predicted from the genome alone.

Identification and Localisation of Post-Translational Modifications

At this point, it is probably important to note that only a minority of spectra are actually assigned to peptides when processing data. In fact, the vast majority, that is at least 60% of spectra produced by the mass spectrometer remain unassigned and this is sometimes called “proteomics dark matter” (Skinner and Kelleher, 2015). This substantial unidentifiable portion of the data is often thought to be attributable to our poor knowledge of Post-Translational Modifications (PTMs). PTMs are alteration of proteins during or after protein biosynthesis, resulting in changes of the protein physicochemical properties, which modulate and regulate cellular functions. Abnormal PTM abundances are involved in disease states.

By identifying the occurrence of regular mass shifts corresponding to the addition or removal of chemical groups of known masses in a peptide MS/MS spectrum, it is possible to identify a PTM and its position on the peptide sequence. When specified in the search parameters, search engines can consider relevant PTMs, such as phosphorylation. However, this number is usually limited because the search space is substantially expanded.

Exhaustive identification of PTMs in high-throughput MS-experiments has proven to be difficult for a number of reasons including the large number of possible PTMs, sub-stoichiometric amounts of modified proteins, and as peptides carrying certain PTMs display MS2 fragmentation patterns which can be difficult to interpret (Creasy and Cottrell, 2004). Therefore successful protein modification studies rely on carefully designed experimental setups applying extensive sample fractionation/enrichment protocols for the detection of low abundant protein species. MS data need to be accurate and contain information rich fragmentation patterns. Software for PTM automated detection is detailed in Bilbao (2018).

It was proposed to rename modified proteins into proteoforms (Smith *et al.*, 2013) or modforms (Prabakaran *et al.*, 2012) suggesting that protein-protein interactions should actually be described as *form-*form interactions as pointed out in Chichester *et al.* (2003).

Proteomics Workflows

In this section, the impact of the technical and experimental choices on data processing and analysis is reviewed.

Database Versus Spectral Library Search

As explained in Section “Proteomics Technical Issues Bearing on Data Processing”, the fragmentation pattern encoded by a tandem (or MS/MS) spectrum supports the identification of the corresponding peptide sequence. Search engines annotate the MS/MS spectra with peptide sequences using various algorithms and scoring strategies (Nesvizhskii, 2010). Even though database search engines remain very popular, a significant number of articles promote the use of direct spectral matching and the construction of spectral libraries to support reliable peptide identification and quantification especially in high-throughput settings (Lam *et al.*, 2007). Computationally, using spectral libraries is also much more efficient as it skips the construction of theoretical spectra. The

downside is of course, the potential incompleteness of the library and its subsequent limited coverage. Nonetheless, the constant improvement of instrument accuracy and sensitivity goes along with the potential for generating libraries with good coverage.

In database search, protein sequences are digested *in silico* into peptides. Considering knowledge about the type of fragmentation technique employed, theoretical spectra are generated from each peptide sequence. The search engine then scores the observed tandem mass spectra by matching against the predicted fragmentation, producing a list of peptide-to-spectrum matches (PSMs).

Spectral library searching offers a natural way of incorporating previous knowledge about observed peptides and their respective fragmentation patterns into a new search. The National Institute of Standards and Technology (NIST, see “Relevant Websites section”) and the Institute for Systems Biology (ISB, see “Relevant Websites section”) have invested effort in compiling high quality publicly available spectrum libraries for different organisms and instrument types including some specialised libraries rich in modified peptides. Spectral libraries generated from previously annotated spectra can be used as reference to further annotate new generated spectra, producing a list of spectrum-to-spectrum matches (SSMs).

To enable the exploration of these spectral libraries when analysing new experimental data, several tools have been developed (Griss, 2016). Firstly, peptide identification with these tools is fast as the experimental data is screened against small size databases and the time-consuming modelling of theoretical spectra can be skipped. Secondly, a library spectrum is rich in information (including fragmentation intensities of a wide range of fragment ions) relative to a theoretical peptide spectrum. Finally, library search tools typically employ simple and fast scoring algorithms and are expected to yield more discriminative match scores than sequence search tools. The main drawback remains the impossible identification of peptides, which are absent in the library. Note that decoy spectral library tools have been developed to evaluate the quality of spectral matching based on an FDR calculation. These are reviewed in Zhang *et al.* (2018).

A particular representation spectral data is in the form of spectral network in which spectral similarity is mapped (nodes are spectra and edges are similarity between spectra as measured by peak alignment). As noted in Guthals *et al.* (2012), the spectral network paradigm is founded on two core principles beyond mainstream approaches: (1) matching unidentified spectra to a spectral library or to other unidentified spectra is more efficient than to theoretical spectra based on reference sequences and (2) consensus interpretation of sets of related spectra is more reliable than identification of one spectrum at a time.

At this point in time, the consensus view when comparing the much more frequently used database search to spectral library matching is that each corresponding output complements each other. The combination of both strategies tends to increase the number of confidently identified peptides and proteins.

Data Dependent Acquisition (DDA) Versus Data Independent Acquisition (DIA)

In shotgun proteomics, the selection of precursor ions in mass spectra (MS1) for fragmentation that will produce tandem mass spectra (MS2) is defined upon intensity. In other words, the ten or twenty (parametered choice) most intense peaks designate which peptide will be fragmented. To make this choice less dependent on the data and somehow less arbitrary, a new approach called “Data Independent Acquisition” (DIA) was introduced contrasting to “Data Dependent Acquisition” (DDA) by means of co-selection and co-fragmentation. DIA is by design avoiding the detection and selection of individual precursor ions during LC-MS analysis.

An early DIA proof-of-principle was reported in Purvine *et al.* (2003). Then Venable *et al.* (2004) suggested the use of sequential isolation and fragmentation of several precursor ion windows (10u each) using a linear ion trap over a defined mass range, thereby introducing the term “data-independent acquisition”. DIA integrates qualitative and quantitative information. Recorded spectra are a continuous map of multiplexed chromatographic profiles from all detectable peptides, which are eluted, ionized and fragmented. As a consequence of this parallel peptide analysis, one of the main challenges to perform identification in DIA spectra is the lack of a direct relationship between each precursor ion and its fragment ions, as opposed to DDA. The most straightforward way to process DIA spectra is the direct submission to DDA search engines. This topic is well covered in Bilbao (2018).

Overall, DDA performance declines as sample complexity increases because the semi-stochastic selection of precursor ions aggravates known limitations for both identification and quantification: limited reproducibility and dynamic range, bias toward high abundance peptides and under-sampling to name a few. In turn, DIA solves these issues while generating highly convoluted, dense and redundant datasets that remain challenging to analyse.

Top Down Versus Bottom up MS

Shotgun proteomics and other strategies that emerged with the increasing resolution power of mass spectrometers are still facing with challenges of protein information loss resulting from the ambiguity of the peptide reassembly process into proteins. To begin with, identified peptides are often not specific to an individual protein and even less to a protein form, which may lead to piecing together chimeric proteins. Secondly, significant portions of a protein may not be identified, for instance due to the absence of lysine or arginine and the oversized peptides generated by tryptic digestion. Consequently, potential meaningful PTMs or sequence variants occurring in these regions cannot be detected.

So far in the present article, all approaches have been based on breaking proteins into pieces (peptides and peptide fragments). The generic term for qualifying these is “bottom-up” proteomics. In recent years, “top-down” proteomics has emerged as an answer to the shortcomings just listed. In this set-up, the intact protein is introduced into the mass spectrometer where both intact and fragment ions masses are measured. Taking advantage of high resolution and mass accuracy, Top Down proteomics studies

have largely been implemented using ESI coupled to either Fourier transform ion cyclotron resonance (FT-ICR) or Orbitrap mass analysers. These approaches are well reviewed in [Catherman et al. \(2014\)](#).

The most renowned software for the identification of intact proteins is ProSight which latest version is defined as light ([Fellers et al., 2015](#)). It uses the precursor mass and a mass tolerance window to generate a possible list of candidates from a larger annotated database. The theoretical fragment ions from the candidates are then compared to the experimentally determined fragment ions within a fragment mass tolerance. A P-score is calculated for each hit, representing the probability that a random sequence could account for the matching ions. ProSight versions were gradually improved to include fixed (e.g., alkylation of cysteine residues) and terminal (e.g., N-terminal acetylation) modifications. An alternative to ProSight is mainly BIG Mascot or MascotTD, based on the popular Bottom Up software platform Mascot and extends the precursor mass cutoff from 16 kDa to 110 kDa (see “Relevant Websites section”).

To be complete, the term “middle-down” proteomics should also be defined in this section. This approach generates larger size peptides than the regular bottom-up. The major challenge is often the selection of the enzyme that will digest proteins into peptides of the appropriate size.

Label-Free Versus Labelled Quantification

Broadly speaking, absolute quantification estimates intracellular protein concentrations on individual samples, and relative quantification compares the measured protein abundances in one sample relative to another. In terms of sample preparation, quantification methods can include stable isotope labels or label-free based.

Adding stable isotope labels significantly reduces experimental bias and improves accuracy, which requires specific but typically straightforward data processing approaches. Relative quantification relies on three main types of techniques in proteomics: metabolic, chemical or enzymatic. The metabolic approach is characterised by the Stable Isotope Labelling with Amino acids in Cell culture (SILAC) method that uses non-radioactive isotopic labelling ([Ong et al., 2002](#)). The chemical methods include isotope-coded affinity tags (ICAT) ([Gygi et al., 1999](#)), isobaric labelling tandem mass tags (TMT) ([Thompson et al., 2003](#)) and isobaric tags for relative and absolute quantification (iTRAQ) ([Ross et al., 2004](#)). The enzymatic approach is illustrated with the $^{16}\text{O}/^{18}\text{O}$ where the enzyme-catalyzed incorporation of oxygen is performed in the C-terminal carboxylic acids of peptides during the digestion procedure ([Yao et al., 2001](#)). In the workflows using of these different methods, samples corresponding to distinct experimental conditions are labelled and combined, and all samples are compared in the same LC-MS analysis of the pooled samples. Labelling is performed on proteins (pre-digestion) or on peptides (post digestion). There is no computational challenge as each mass change caused by labels is known and only mass differences are sought. In the end, label-based methods increase costs (reagents are expensive) and sample preparation time, while limiting the number of different samples that can be compared.

In contrast, in label-free methods different samples are prepared and analysed individually, requiring computational methods to align retention time across multiple samples and to normalise intensity. As ([Bilbao, 2018](#)) is fairly detailed on software available for supporting peptide and protein quantification, this section will only mention image-based analysis. For example, MS1 “2D-images” plotting m/z and retention time (rt) values of peptides have been used and processed in much the same way 2D-gels images are scanned and compared for differential analysis. In this representation, retention time alignment is approached by superimposing 2D (m/z , rt) maps. Solutions to these image processing issues overlapping those raised by 2D-gel analysis are reviewed in ([Dowsey et al., 2010](#)). Furthermore, the technique known as MALDI-imaging mass spectrometry offers yet another label-free technology. A tissue section is coated with matrix and irradiated with a laser. Molecules are then desorbed and ionised according to the MALDI technique. Spectra are acquired from each location (pixel) over the surface of the tissue and 2D ion density images are reconstructed from the spectra ([Seeley and Caprioli, 2008](#)). Image analysis software is listed on the website of the active MALDI-imaging community: <https://ms-imaging.org>.

A comprehensive coverage of quantification should include a mention of targeted proteomics usually opposed to discovery (shotgun) proteomics. Selected reaction monitoring (SRM) has become the MS method of choice to accurately and consistently quantify a selection of proteins across samples. It is typically used as a means of validating the expression of a protein or a proteotypic (i.e., uniquely characterising a protein ([Mallick et al., 2007](#))) peptide. Note that so-called “directed proteomics” is an intermediary strategy that entails using a list of predetermined precursor ions, otherwise called “inclusion list” in order to focus on a particular segment of a global proteome and increase reproducibility. To conclude this section on quantification approaches, [Fig. 4](#) (inspired from [Fig. 1](#) in ([Bensimon et al., 2012](#))) highlights the requirements of the different label-free strategies and the corresponding depth of analysis achieved.

Some Omics Bridging Issues

Needless to say, proteomics as part of the omics landscape is integrated with many other omics. It is virtually impossible to cover all existing options and the following highlights a selection.

Proteogenomics

Genomic sequencing data can be used to improve protein identification through refining the construction of a protein sequence database. Proteomic data can in turn support large scale RNA and DNA sequencing initiatives by providing evidence of coding sequence variants, novel coding transcripts or refined gene boundaries. In fact support to complete genome annotation was suggested

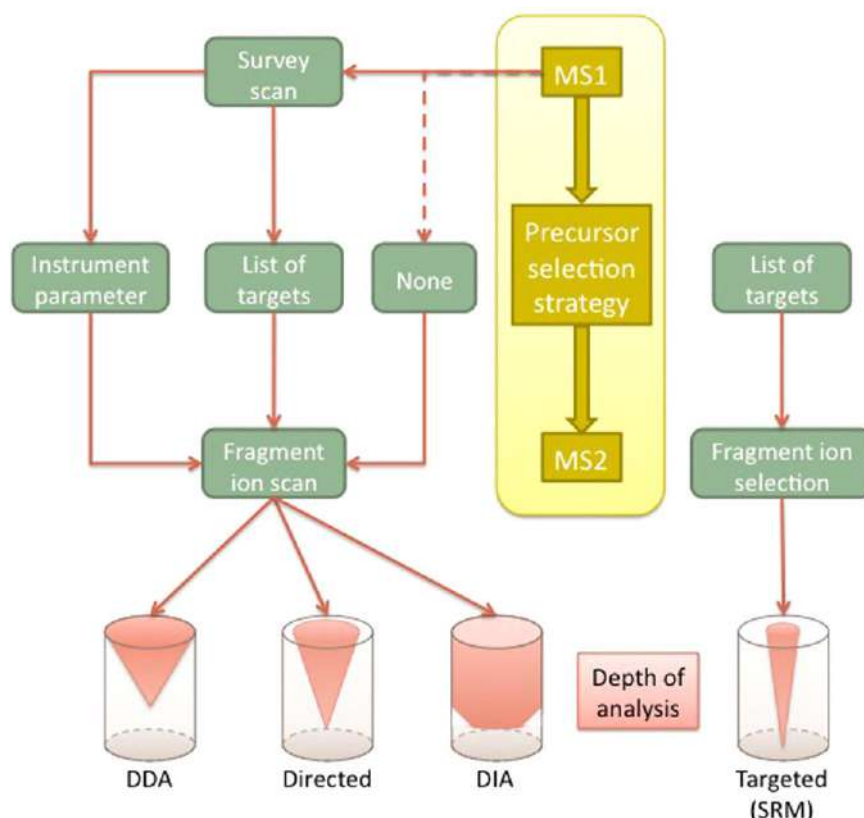


Fig. 4 The various protein identification and quantification strategies and corresponding options at the MS1 level and subsequent MS2 level following precursor ion selection (full red line, required; dashed red line, optional). DDA=Data-Dependent Acquisition (implemented in shotgun proteomics); DIA=Data-Independent Acquisition; “Directed” and “Targeted” are defined in the text. Results are represented in terms of coverage in red volumes to illustrate the depth of analysis.

very early on in proteomics with the prospect of revealing information not accessible in genomics (Küster *et al.*, 2001; Choudhary *et al.*, 2001). But it took longer than expected to design systematic approaches such as Branca *et al.* (2014) and the work that follows to shape the field of proteogenomics. In the meanwhile, isolated attempts led to contribute to plant gene/protein characterisation, e.g., identification of paralogue genes from multigenic families (Delalande *et al.*, 2005) as well as refining bacterial genome annotation, e.g., identification of precise translation start in proteins of *Mycobacterium tuberculosis* (Rison *et al.*, 2007).

Since 2013, efforts in proteogenomics are more structured. For instance it is playing an important role in the Chromosome-Centric Human Proteome Project (C-HPP) developed within the Human Proteome Organisation (HUPO). C-HPP aims at characterising all protein coding genes of the human proteome and the information is collected and centralised in the neXtprot database (Gaudet *et al.*, 2013). Proteogenomics is also gaining attention in bacterial genome sequencing and annotation. Beyond the resolution of ambiguous cases of translations start sites, generic solutions are now available (Omasits *et al.*, 2017) and make a difference in terms of coding sequence annotation in complete genomes.

Another aspect of proteogenomics is shown in the range of software was developed for the identification of sequence variants through building customised databases of protein sequences of known variant peptides. There are basically three options for the search engines used in this context: mass spectra are searched (i) once against a database of mixed of wild-type and variant peptides, (ii) twice against a first database of wild-type and a second database of variant peptides (iii) once against a database of variant peptides. These approaches are benchmarked on two large human and bacterial datasets with two different search engines in Ivanov *et al.* (2017). Despite a preference for the first strategy, results with others are close and possible gaps are influenced by the search database size or the size of the training data set used for validation.

The existence of small open reading frames (less than 300 nucleotides) undetected by gene prediction software (especially in the human and mouse genomes) has been debated prior the finding experimental means to produce evidence. A solution was brought by ribosome profiling coupled to MS-based protein and peptide identification (Koch *et al.*, 2014). This also led to the creation of a database of small ORFs (see “Relevant Websites section”).

Ultimately, proteogenomics has unique information to contribute in the context of other omics. For example, proteogenomic techniques can help assess protein-level expression in viral infections as they alter gene expression and protein production when the virus hijacks cellular processes that allow for self-replication (Ruggles *et al.*, 2017).

Interactomics

The rising of network biology (Barabási and Oltvai, 2004) has significantly modified knowledge representation in molecular biology. In protein science, experimental techniques for studying protein-protein interactions and protein expression in cells and tissues have evolved in parallel for a while and finally met to benefit each other. (Bensimon *et al.*, 2012) provide a very clear account of this mutual influence and resulting integration mostly in the context of systems biology. In fact, proteomics has been used for solving protein-protein interactions in complexes since the publication of the Tandem Affinity Purification (TAP) method (Rigaut *et al.*, 1999). Ten years later, a full pipeline known as ProHits processes raw MS data files to produce fully annotated protein-protein interaction datasets and considered a reference (Liu *et al.*, 2010). The noisy nature of interactomics data led the authors to also release a contaminant database known as the Crapome (<http://www.crapome.org>).

Substantial progress in generating reliable MS-based interactome data is expected to also arise from work that has been and still is invested in detecting protein interactions by cross-link experiments (shortened as XL-MS). The identification of cross-linked peptides from mass spectra and the mapping of identified cross-links in a structural context naturally relies on efficient computational methods, as reviewed in (Yilmaz *et al.*, 2018). Cryo-EM (Electron Microscopy) data was shown to complement XL-MS (Snijder *et al.*, 2017) and such type of “hybrid” MS strategies (i.e., combined with other technologies) is very likely to significantly expand in the years to come.

PTM Detection and Mapping

In this section, two examples of the most frequent PTMs are briefly covered to highlight the productive contribution of bioinformatics in the past decade. Other related issues are discussed later in the open questions.

Phosphoproteomics

O-Phosphorylation, which corresponds to the addition of a phosphate residue on a serine, a threonine or a tyrosine, is a simple modification that plays a central role in signalling processes. A rough estimate of 60% phospho-proteins is given for the content in the human genome (Vlastaridis *et al.*, 2017).

Phosphoproteomics is massively contributing to populating databases such as PhosphoSite (see “Relevant Websites section”). Similar experimental techniques are commonly applied. In particular, metal-based phosphopeptide enrichment strategies methods including immobilized metal affinity chromatography (IMAC) and metal oxide affinity chromatography (MOAC) are prevalent in the field. Phosphopeptide search is often included in database search engines though the remaining challenge remains the positioning of the PTM. Many peptides contain several potential sites and scoring is necessary (Savitski *et al.*, 2011)

As pointed out in Riley and Coon (2016) phosphoproteomics studies tend to focus on O-phosphorylation (serine, threonine, and tyrosine) while alternative N-phosphorylation (lysine, arginine, and histidine) and S-phosphorylation (cysteine) are known to regulate/modulate signalling in bacterial systems, and possibly in eukaryotes as well Kee and Muir (2012). Conventional LC-MS/MS approaches need to be revised to accommodate the specificity of these modifications in higher throughput.

Glycoproteomics

Glycosylation entails the attachment of glycan molecules (oligosaccharides) on proteins. At least 60% of proteins are expected to be glycosylated in mammalian proteomes and many thousands of fully defined glycan structures have been published. Mass spectrometry is the most popular technique to characterise the structure of glycans and glycopeptides; NMR is a common alternative to MS for solving structures with a lesser throughput and higher amounts needed of sample material. A repository of glycan structures each one given a unique identifier was released in an effort to channel the diverse contributions of the glycoscience community (Tiemeyer *et al.*, 2017). In parallel, curated databases are also developed as summarised in Lisacek *et al.* (2017) Fig. 5 illustrates the variety of glycan molecules attached to a specific site of human elastase.

To date, many studies are based on MS1 data and limited to identifying glycopeptides associated with glycan monosaccharide compositions as opposed to fully defined structures, e.g., Rombouts *et al.* (2016). The fragmentation pattern of glycans was analysed and formalised by Domon and Costello (1988). Multiple fragmentation energies may be required to gain enough information on a glycan structure. Then, standard CID fragmentation of glycopeptides tends to reveals mostly losses of glycans via glycosidic bond cleavage, and relatively little peptide bond cleavage. This lack of information means that determining the peptide identity will be difficult.

Technical steps of current glycoproteomics workflows are not evenly mature or optimised. Advanced methods are the result of applicable or transposable proteomics principles while open questions reflect issues specific to the field of glycosciences and still in need to be tackled. Throughput is increasing and in much the same way lessons learnt in proteomics techniques have benefitted those in glycoproteomics, ideas implemented in proteome informatics also deeply influence glycoproteomics data analysis software. For example, data pre-processing approaches are considered operational (handling of raw files, peak picking, spectral matching or deconvolution). In contrast, open questions such as site-specific localisation and profiling of glycosylation, integration of candidate peptide and glycan scores, target-decoy models and FDR calculation still require time to reach a consensus stage. Reviews such as Dallas *et al.* (2013), Hu *et al.* (2016) and Gaunitz *et al.* (2017) provide rich overviews of glycoproteome informatics with detailed analyses of the obstacles to full automation. Interestingly, undertaking the reanalysis of a dataset initially

← ★ Elastase iib

UniProt: [P08861](#)

neXtProt: [NX P08861](#)

GeneCards: [CELA3B](#)

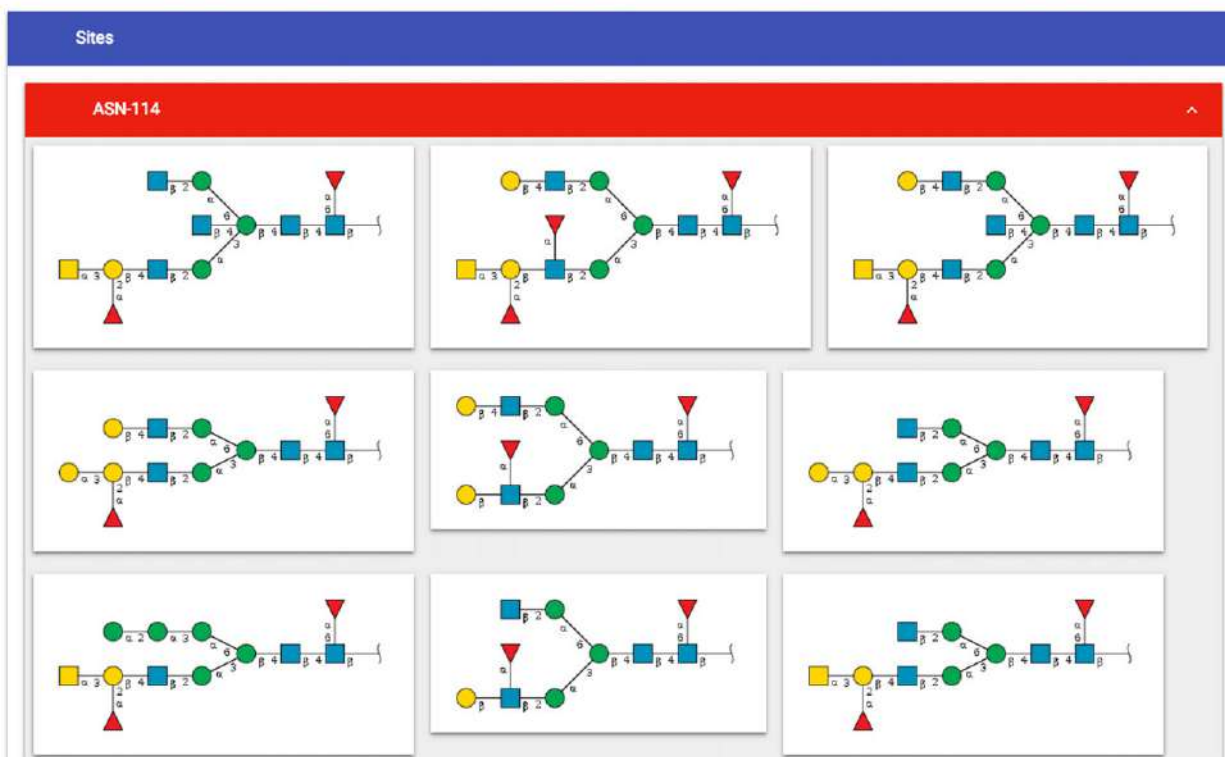


Fig. 5 Example of glycan molecules attached to the asparagine residue (N-linked) at position 114 of human elastase. The cartoon code is detailed in Varki, A., Cummings, R.D., Aebi, M., *et al.*, 2015. Symbol nomenclature for graphical representations of glycans. *Glycobiology* 25, pp. 1323–1324. Available at: <https://doi.org/10.1093/glycob/cwv091>. This information was found at <https://glyconnect.expasy.org/>.

released as global proteomic and phosphoproteomic analyses led to the identification of a large number of glycopeptides (Hu *et al.*, 2017) thereby demonstrating both the value of reanalysis and the wealth of information hidden in MS data.

Open Questions and Future Directions

Standards and Coverage

To link with the discussion in [Robin \(2018\)](#), the importance of complying with standards in order to support comparability and reproducibility of data is also stressed here. More effort is needed in that direction and not surprisingly new standards are emerging in proteogenomics (proBAM [Menschaert et al., 2018](#)) or in glycoproteomics (MIRAGE [Kolarich et al., 2013](#))

The proBAM format is based on the BAM format originally designed to store alignments of short DNA or RNA reads to a reference genome. The main advantages of proBAM are (i) the connection of proteomics data to a range of bioinformatics tools that process BAM files, (ii) the option to import information about peptide identification to a genome browser, (iii) exchange and re-use of MS-based proteomics data made easier. The production of this format directly stems from recent progress in proteogenomics.

It can be safely assumed that further effort will be invested in developing the necessary tools in particular with the goal of increasing the peptide coverage in protein identification software and mapping the various sources of variations ranging from SNPs (Single Nucleotide Polymorphism) to PTMs.

Method Improvement and Further Integration With Other – Omics

As mentioned throughout this article, technological development plays a key role in the evolution of proteomics. MS-based proteomics methodologies have evolved rapidly over recent years along with a remarkable progress on specialised software. In fact,

proteome informatics has to keep up with constant innovation such as moving from DDA to DIA technologies or tackling the challenges of top down proteomics. Furthermore, new approaches are nowadays used in combination with other omics such as genomics, transcriptomics and metabolomics (see related discussion [Bilbao, 2018](#)).

This increasing throughput and large-scale context is grounded in the big data era and requires robust and efficient MS software as well as advanced MS hardware for comprehensively exploiting the information and generating knowledge. In this sense, progress in instrumentation and analytical methods are expected, and importantly, development of new computational methods to cope with these advances will continue to be an active area of research. An increasing number of available tools including proprietary and open source can be expected.

It should also be noted that both in transcriptomics and proteomics, the unavailability of sequenced or well-annotated genomes and therefore the lack of reference templates limit the interpretation of data. This is particularly acute in plant biology (see for example Special Issue of *Biochimica et Biophysica Acta (BBA) – Proteins and Proteomics* Edited by Hans-Peter Mock Plant Proteomics: a bridge between fundamental processes and crop production, Volume 1864, Issue 8, Pages 881–1060, 2016) and with animals that are not considered as model organisms (see for example [De Almeida et al., 2018](#)). This gap is likely to be filled as genome-sequencing technology advances.

Definite Potential for Elucidating PTM Cross-Talk

As pointed in [Venne et al. \(2014\)](#), PTMs can be mutually exclusive such as the phosphorylation and the O-GlcNAcylation of serine and threonine of signalling proteins presented very early on as the “yin-yang hypothesis” ([Hart et al., 1995](#)) but they can also be cooperative as in the well-studied case of histones ([Schwämmle et al., 2014](#)). However, crosstalk appears to be species-dependent ([Beltrao et al., 2012](#)) making the concept of “PTM code” still difficult to define and circumscribe despite being referred to in many articles (e.g., in relation to chaperones ([Cloutier and Coulombe, 2013](#)) or tubulins ([Janke, 2014](#))). To grasp an understanding of the scenario of protein modifications and better apprehend interplay, work should however not be limited to describing the modified protein “landscape” of a cell or a tissue. As a matter of fact, understanding also means entangling a tight bundle of interactions, which encompasses ascertaining the role, the origin and the purpose of enzymes that carry out PTMs. This task is likely to take a while.

The generalised use of mass spectrometry (MS) following top down, middle down or bottom-up strategies mentioned in Section “Proteomics Workflows”, as well as the refinement of dedicated approaches described in Section “PTM Detection and Mapping”, explain the rather recent accumulation of detected PTMs. The role of PTMs has an obvious bearing on the overall system of interactions. In fact, examples of protein-protein interaction networks integrating PTM knowledge are rare and devoted to mapping data collected in eukaryotes such the yeast methylome ([Erce et al., 2012](#)) or the phospho-tyrosine interaction network in human ([Grossmann et al., 2015](#)). Bioinformatics resources mapping modified protein interactions are still scarce. For example, with a focus on human and mouse data, TOPFind attempts to address this question with the PathFinder tool ([Fortelny et al., 2015](#)) mapping and modelling a protease interaction network.

Conclusion

This overview spans over two decades of bioinformatics development for the reliable identification of proteins in complex mixtures. Many more points and applications could be raised and the reader will find finer details regarding a broad range of topics in cited references. Complexity is unlikely to be reduced in future large-scale studies as hinted in Section “Open Questions and Future Directions” and confirmed for instance by recent effort to launch metaproteomics applications (proteomics of microbiota), which in much the same metagenomics was initiated, cannot be tackled without powerful software ([Muth et al., 2015](#); [Cheng et al., 2017](#)). Proteome informatics can still prosper.

See also: Clinical Proteomics. Data-Information-Concept Continuum From a Text Mining Perspective. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Identification of Proteins from Proteomic Analysis. Identification of Sequence Patterns, Motifs and Domains. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein Localization. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Post-Translational Modification Prediction. Protein Properties. Protein-Protein Interaction Databases. Proteomics Data Representation and Databases. Proteomics Mass Spectrometry Data Analysis Tools. Quantification of Proteins from Proteomic Analysis. Text Mining for Bioinformatics Using Biomedical Literature. The Evolution of Protein Family Databases. Transmembrane Domain Prediction. Utilising IPG-IEF to Identify Differentially-Expressed Proteins

References

- Aebersold, R., Mann, M., 2003. Mass spectrometry-based proteomics. *Nature* 422, 198–207. Available at: <https://doi.org/10.1038/nature01511>.
 De Almeida, A.M., Eckersall, D., Miller, I., 2018. *Proteomics in Domestic Animals: From Farm to Systems Biology*. Springer.

- Angel, T.E., Aryal, U.K., Hengel, S.M., *et al.*, 2012. Mass spectrometry-based proteomics: Existing capabilities and future directions. *Chem. Soc. Rev.* 41, 3912. Available at: <https://doi.org/10.1039/c2cs15331a>.
- Appel, R.D., Hochstrasser, D.F., Funk, M., *et al.*, 1991. The MELANIE project: From a biopsy to automatic protein map interpretation by computer. *Electrophoresis* 12, 722–735. Available at: <https://doi.org/10.1002/elps.1150121006>.
- Baginsky, S., 2009. Plant proteomics: Concepts, applications, and novel strategies for data interpretation. *Mass Spectrom. Rev.* 28, 93–120. Available at: <https://doi.org/10.1002/mas.20183>.
- Baker, E.S., Burnum-Johnson, K.E., Ibrahim, Y.M., *et al.*, 2015. Enhancing bottom-up and top-down proteomic measurements with ion mobility separations. *Proteomics* 15, 2766–2776. Available at: <https://doi.org/10.1002/pmic.201500048>.
- Barabási, A.-L., Oltvai, Z.N., 2004. Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.* 5, 101–113. Available at: <https://doi.org/10.1038/nrg1272>.
- Beltrao, P., Albanese, V., Kenner, L.R., *et al.*, 2012. Systematic functional prioritization of protein posttranslational modifications. *Cell* 150, 413–425. Available at: <https://doi.org/10.1016/j.cell.2012.05.036>.
- Bensimon, A., Heck, A.J.R., Aebersold, R., 2012. Mass spectrometry-based proteomics and network biology. *Annu. Rev. Biochem.* 81, 379–405. Available at: <https://doi.org/10.1146/annurev-biochem-072909-100424>.
- Bilbao, A., 2018. encyclopedia of bioinformatics and computational biology: Proteomics mass spectrometry data analysis tools. In: *Reference Module in Life Sciences*. Elsevier. <https://doi.org/10.1016/B978-0-12-809633-8.20274-4>.
- Bond, N.J., Shliha, P.V., Lilley, K.S., Gatto, L., 2013. Improving qualitative and quantitative performance for MS^E-based Label-free Proteomics. *J. Proteome Res.* 12, 2340–2353. Available at: <https://doi.org/10.1021/pr300776t>.
- Branca, R.M.M., Orre, L.M., Johansson, H.J., *et al.*, 2014. HiRIEF LC-MS enables deep proteome coverage and unbiased proteogenomics. *Nat. Methods* 11, 59–62. Available at: <https://doi.org/10.1038/nmeth.2732>.
- Catherman, A.D., Skinner, O.S., Kelleher, N.L., 2014. Top down proteomics: Facts and perspectives. *Biochem. Biophys. Res. Commun.* 445, 683–693. Available at: <https://doi.org/10.1016/j.bbrc.2014.02.041>.
- Cheng, K., Ning, Z., Zhang, X., *et al.*, 2017. MetaLab: An automated pipeline for metaproteomic data analysis. *Microbiome* 5. Available at: <https://doi.org/10.1186/s40168-017-0375-2>.
- Chichester, C., Nikitin, F., Ravarini, J.-C., Lisacek, F., 2003. Consistency checks for characterizing protein forms. *Comput. Biol. Chem.* 27, 29–35.
- Choudhary, J.S., Blackstock, W.P., Creasy, D.M., Cottrell, J.S., 2001. Interrogating the human genome using uninterpreted mass spectrometry data. *Proteomics* 1, 651–667. Available at: [https://doi.org/10.1002/1615-9861\(200104\)1:5651::AID-PROT6513.0.CO;2-N](https://doi.org/10.1002/1615-9861(200104)1:5651::AID-PROT6513.0.CO;2-N).
- Cloutier, P., Coulombe, B., 2013. Regulation of molecular chaperones through post-translational modifications: Decrypting the chaperone code. *Biochim. Biophys. Acta* 1829, 443–454. Available at: <https://doi.org/10.1016/j.bbagr.2013.02.010>.
- Creasy, D.M., Cottrell, J.S., 2004. Unimod: Protein modifications for mass spectrometry. *Proteomics* 4, 1534–1536. Available at: <https://doi.org/10.1002/pmic.200300744>.
- Crowell, K.L., Slys, G.W., Baker, E.S., *et al.*, 2013. LC-IMS-MS Feature Finder: Detecting multidimensional liquid chromatography, ion mobility and mass spectrometry features in complex datasets. *Bioinformatics* 29, 2804–2805. Available at: <https://doi.org/10.1093/bioinformatics/btt465>.
- Dallas, D.C., Martin, W.F., Hua, S., German, J.B., 2013. Automated glycopeptide analysis-review of current state and future directions. *Brief. Bioinform.* 14, 361–374. Available at: <https://doi.org/10.1093/bib/bbs045>.
- Delalande, F., Carapito, C., Brizard, J.-P., Brugidou, C., Van Dorsselaer, A., 2005. Multigenic families and proteomics: Extended protein characterization as a tool for paralog gene identification. *Proteomics* 5, 450–460. Available at: <https://doi.org/10.1002/pmic.200400954>.
- Distler, U., Kuharev, J., Navarro, P., *et al.*, 2014. Drift time-specific collision energies enable deep-coverage data-independent acquisition proteomics. *Nat. Methods* 11, 167–170. Available at: <https://doi.org/10.1038/nmeth.2767>.
- Domon, B., Costello, C.E., 1988. A systematic nomenclature for carbohydrate fragmentations in FAB-MS/MS spectra of glycoconjugates. *Glycoconj. J.* 5, 397–409. Available at: <https://doi.org/10.1007/BF01049915>.
- Dowsey, A.W., Dunn, M.J., Yang, G.-Z., 2008. Automated image alignment for 2D gel electrophoresis in a high-throughput proteomics pipeline. *Bioinformatics* 24, 950–957. Available at: <https://doi.org/10.1093/bioinformatics/btn059>.
- Dowsey, A.W., English, J.A., Lisacek, F., *et al.*, 2010. Image analysis tools and emerging algorithms for expression proteomics. *Proteomics* 10, 4226–4257. Available at: <https://doi.org/10.1002/pmic.200900635>.
- Elias, J.E., Gygi, S.P., 2007. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods* 4, 207–214. Available at: <https://doi.org/10.1038/nmeth1019>.
- Eng, J.K., McCormack, A.L., Yates, J.R., 1994. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* 5, 976–989. Available at: [https://doi.org/10.1016/1044-0305\(94\)80016-2](https://doi.org/10.1016/1044-0305(94)80016-2).
- Erce, M.A., Pang, C.N.I., Hart-Smith, G., Wilkins, M.R., 2012. The methylproteome and the intracellular methylation network. *Proteomics* 12, 564–586. Available at: <https://doi.org/10.1002/pmic.201100397>.
- Fellers, R.T., Greer, J.B., Early, B.P., *et al.*, 2015. ProSight Lite: Graphical software to analyze top-down mass spectrometry data. *Proteomics* 15, 1235–1238. Available at: <https://doi.org/10.1002/pmic.201400313>.
- Fortelny, N., Yang, S., Pavlidis, P., Lange, P.F., Overall, C.M., 2015. Proteome TopFIND 3.0 with TopFINDER and PathFINDER: Database and analysis tools for the association of protein termini to pre- and post-translational events. *Nucleic Acids Res.* 43, D290–D297. Available at: <https://doi.org/10.1093/nar/gku1012>.
- Frank, A., Pevzner, P., 2005. PepNovo: De novo peptide sequencing via probabilistic network modeling. *Anal. Chem.* 77, 964–973. Available at: <https://doi.org/10.1021/ac048788h>.
- Gaudet, P., Argoud-Puy, G., Cusin, I., *et al.*, 2013. neXtProt: Organizing protein knowledge in the context of human proteome projects. *J. Proteome Res.* 12, 293–298. Available at: <https://doi.org/10.1021/pr300830v>.
- Gaunitz, S., Nagy, G., Pohl, N.L.B., Novotny, M.V., 2017. Recent advances in the analysis of complex glycoproteins. *Anal. Chem.* 89, 389–413. Available at: <https://doi.org/10.1021/acs.analchem.6b04343>.
- Griss, J., 2016. Spectral library searching in proteomics. *Proteomics* 16, 729–740. Available at: <https://doi.org/10.1002/pmic.201500296>.
- Grossmann, A., Benlasfer, N., Birth, P., *et al.*, 2015. Phospho-tyrosine dependent protein-protein interaction network. *Mol. Syst. Biol.* 11 (3), 794. Available at: <https://doi.org/10.15252/msb.20145968>.
- Guthals, A., Gan, Y., Murray, L., *et al.*, 2017. De novo MS/MS sequencing of native human antibodies. *J. Proteome Res.* 16, 45–54. Available at: <https://doi.org/10.1021/acs.jproteome.6b00608>.
- Guthals, A., Watrous, J.D., Dorrestein, P.C., Bandeira, N., 2012. The spectral networks paradigm in high throughput mass spectrometry. *Mol. Biosyst.* 8 (10), 2535–2544. Available at: <https://doi.org/10.1039/c2mb25085c>.
- Gygi, S.P., Rist, B., Gerber, S.A., *et al.*, 1999. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat. Biotechnol.* 17, 994–999. Available at: <https://doi.org/10.1038/13690>.
- Hart, G.W., Greis, K.D., Dong, L.Y., *et al.*, 1995. O-linked N-acetylglucosamine: The “yin-yang” of Ser/Thr phosphorylation? Nuclear and cytoplasmic glycosylation. *Adv. Exp. Med. Biol.* 376, 115–123.
- Henneman, A.A., Palmblad, M., 2013. Retention time prediction and protein identification. In: *Matthiesen, R. (Ed.), Mass Spectrometry Data Analysis in Proteomics*. Totowa, NJ: Humana Press, pp. 101–118. Available at: https://doi.org/10.1007/978-1-62703-392-3_4.
- Henzel, W.J., Billeci, T.M., Stults, J.T., *et al.*, 1993. Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proc. Natl. Acad. Sci. USA* 90, 5011–5015.

- Hu, H., Khatri, K., Zaia, J., 2017. Algorithms and design strategies towards automated glycoproteomics analysis: Algorithms and design strategies. *Mass Spectrom. Rev.* 36 (4), 475–498. Available at: <https://doi.org/10.1002/mas.21487>.
- Hu, Y., Shah, P., Clark, D.J., Ao, M., Zhang, H., 2017. Reanalysis of global proteomic and phosphoproteomic data identified a large number of glycopeptides. Available at: <https://doi.org/10.1101/233247>.
- Ivanov, M.V., Lobas, A.A., Karpov, D.S., Moshkovskii, S.A., Gorshkov, M.V., 2017. Comparison of false discovery rate control strategies for variant peptide identifications in shotgun proteogenomics. *J. Proteome Res.* 16, 1936–1943. Available at: <https://doi.org/10.1021/acs.jproteome.6b01014>.
- James, P., Quadroni, M., Carafoli, E., Gonnet, G., 1993. Protein identification by mass profile fingerprinting. *Biochem. Biophys. Res. Commun.* 195, 58–64. Available at: <https://doi.org/10.1006/bbrc.1993.2009>.
- Janke, C., 2014. The tubulin code: Molecular components, readout mechanisms, and functions. *J. Cell Biol.* 206, 461–472. Available at: <https://doi.org/10.1083/jcb.201406055>.
- Johnson, R.S., Martin, S.A., Biemann, K., Stults, J.T., Watson, J.T., 1987. Novel fragmentation process of peptides by collision-induced decomposition in a tandem mass spectrometer: Differentiation of leucine and isoleucine. *Anal. Chem.* 59, 2621–2625. Available at: <https://doi.org/10.1021/ac00148a019>.
- Kee, J.-M., Muir, T.W., 2012. Chasing phosphohistidine, an elusive sibling in the phosphoamino acid family. *ACS Chem. Biol.* 7, 44–51. Available at: <https://doi.org/10.1021/cb200445w>.
- Koch, A., Gawron, D., Steyaert, S., *et al.*, 2014. A proteogenomics approach integrating proteomics and ribosome profiling increases the efficiency of protein identification and enables the discovery of alternative translation start sites. *Proteomics* 14, 2688–2698. Available at: <https://doi.org/10.1002/pmic.201400180>.
- Kolarich, D., Rapp, E., Struwe, W.B., *et al.*, 2013. The minimum information required for a glycomics experiment (MIRAGE) project: Improving the Standards for reporting mass-spectrometry-based glycoanalytic data. *Mol. Cell. Proteomics* 12, 991–995. Available at: <https://doi.org/10.1074/mcp.0112.026492>.
- Küster, B., Mortensen, P., Andersen, J.S., Mann, M., 2001. Mass spectrometry allows direct identification of proteins in large genomes. *Proteomics* 1, 641–650. Available at: [https://doi.org/10.1002/1615-9861\(200104\)1:5<641::AID-PROT6413.0.CO;2-R](https://doi.org/10.1002/1615-9861(200104)1:5<641::AID-PROT6413.0.CO;2-R).
- Lam, H., Deutsch, E.W., Eddes, J.S., *et al.*, 2007. Development and validation of a spectral library searching method for peptide identification from MS/MS. *Proteomics* 7, 655–667. Available at: <https://doi.org/10.1002/pmic.200600625>.
- Lisacek, F., Mariethoz, J., Alocci, D., *et al.*, 2017. Databases and associated tools for glycomics and glycoproteomics. In: Lauc, G., Wührer, M. (Eds.), *High-Throughput Glycomics and Glycoproteomics*. New York, NY: Springer New York, pp. 235–264. https://doi.org/10.1007/978-1-4939-6493-2_18.
- Liu, J., Bell, A.W., Bergeron, J.J.M., *et al.*, 2007. Methods for peptide identification by spectral comparison. *Proteome Sci.* 5, 3. Available at: <https://doi.org/10.1186/1477-5956-5-3>.
- Liu, Y., Hüttenhain, R., Collins, B., Aebersold, R., 2013. Mass spectrometric protein maps for biomarker discovery and clinical research. *Expert Rev. Mol. Diagn.* 13, 811–825. Available at: <https://doi.org/10.1586/14737159.2013.845089>.
- Liu, G., Zhang, J., Larsen, B., *et al.*, 2010. ProHits: Integrated software for mass spectrometry-based interaction proteomics. *Nat. Biotechnol.* 28, 1015–1017. Available at: <https://doi.org/10.1038/nbt1010-1015>.
- Ma, B., 2015. Novor: Real-Time peptide de novo sequencing software. *J. Am. Soc. Mass Spectrom.* 26, 1885–1894. Available at: <https://doi.org/10.1007/s13361-015-1204-0>.
- Mahon, P., Dupree, P., 2001. Quantitative and reproducible two-dimensional gel analysis using Phoretix 2D Full. *Electrophoresis* 22, 2075–2085. Available at: [https://doi.org/10.1002/1522-2683\(200106\)22:12<2075::AID-ELPS20753.0.CO;2-C](https://doi.org/10.1002/1522-2683(200106)22:12<2075::AID-ELPS20753.0.CO;2-C).
- Makarov, A., 2000. Electrostatic axially harmonic orbital trapping: A high-performance technique of mass analysis. *Anal. Chem.* 72, 1156–1162. Available at: <https://doi.org/10.1021/ac991131p>.
- Mann, M., Højrup, P., Roepstorff, P., 1993. Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biol. Mass Spectrom.* 22, 338–345. Available at: <https://doi.org/10.1002/bms.1200220605>.
- Mallick, P., Schirle, M., Chen, S.S., *et al.*, 2007. Computational prediction of proteotypic peptides for quantitative proteomics. *Nat. Biotechnol.* 25, 125–131. Available at: <https://doi.org/10.1038/nbt1275>.
- Mann, M., Wilm, M., 1994. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Anal. Chem.* 66, 4390–4399. Available at: <https://doi.org/10.1021/ac00096a002>.
- Menschaert, G., Wang, X., Jones, A.R., *et al.*, 2018. The proBAM and proBed standard formats: Enabling a seamless integration of genomics and proteomics data. *Genome Biol.* 19. Available at: <https://doi.org/10.1186/s13059-017-1377-x>.
- Mitchell Wells, J., McLuckey, S.A., 2005. Collision-induced dissociation (CID) of peptides and proteins. In: Burlingame, A.L. (Ed.), *Methods in Enzymology* 402. Elsevier, pp. 148–185. [https://doi.org/10.1016/S0076-6879\(05\)02005-7](https://doi.org/10.1016/S0076-6879(05)02005-7).
- Moruz, L., Käll, L., 2017. Peptide retention time prediction. *Mass Spectrom. Rev.* 36, 615–623. Available at: <https://doi.org/10.1002/mas.21488>.
- Murray, K.K., Boyd, R.K., Eberlin, M.N., *et al.*, 2013. Definitions of terms relating to mass spectrometry (IUPAC recommendations 2013). *Pure Appl. Chem.* 85, 1515–1609. Available at: <https://doi.org/10.1351/PAC-REC-06-04-06>.
- Muth, T., Behne, A., Heyer, R., *et al.*, 2015. The MetaProteomeAnalyzer: A powerful open-source software suite for metaproteomics data analysis and interpretation. *J. Proteome Res.* 14, 1557–1565. Available at: <https://doi.org/10.1021/pr501246w>.
- Muth, T., Renard, B.Y., 2017. Evaluating de novo sequencing in proteomics: Already an accurate alternative to database-driven peptide identification? *Brief. Bioinform.* Available at: <https://doi.org/10.1093/bib/bbx033>.
- Nesvizhskii, A.I., 2010. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *J. Proteomics* 73, 2092–2123. Available at: <https://doi.org/10.1016/j.jprot.2010.08.009>.
- Nilsson, T., Mann, M., Aebersold, R., *et al.*, 2010. Mass spectrometry in high-throughput proteomics: Ready for the big time. *Nat. Methods* 7, 681–685. Available at: <https://doi.org/10.1038/nmeth0910-681>.
- Omasits, U., Varadarajan, A.R., Schmid, M., *et al.*, 2017. An integrative strategy to identify the entire protein coding potential of prokaryotic genomes by proteogenomics. *Genome Res.* 27, 2083–2095. Available at: <https://doi.org/10.1101/gr.218255.116>.
- Ong, S.-E., Blagoev, B., Kratchmarova, I., *et al.*, 2002. Stable isotope labeling by amino acids in cell culture, silac, as a simple and accurate approach to expression proteomics. *Mol. Cell. Proteomics* 1, 376–386. Available at: <https://doi.org/10.1074/mcp.M200025-MCP200>.
- Pappin, D.J., Højrup, P., Bleasby, A.J., 1993. Rapid identification of proteins by peptide-mass fingerprinting. *Curr. Biol.* 3, 327–332.
- Perkins, D.N., Pappin, D.J.C., Creasy, D.M., Cottrell, J.S., 1999. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 20, 3551–3567. Available at: [https://doi.org/10.1002/\(SICI\)1522-2683\(19991201\)20:183551::AID-ELPS35513.0.CO;2-2](https://doi.org/10.1002/(SICI)1522-2683(19991201)20:183551::AID-ELPS35513.0.CO;2-2).
- Pino, L.K., Searle, B.C., Bollinger, J.G., *et al.*, 2017. The skyline ecosystem: Informatics for quantitative mass spectrometry proteomics. *Mass Spectrom. Rev.* Available at: <https://doi.org/10.1002/mas.21540>.
- Prabakaran, S., Lippens, G., Steen, H., Gunawardena, J., 2012. Post-translational modification: Nature's escape from genetic imprisonment and the basis for dynamic information encoding. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 4, 565–583. Available at: <https://doi.org/10.1002/wsbm.1185>.
- Purvine, S., Eppel*, J.-T., Yi, E.C., Goodlett, D.R., 2003. Shotgun collision-induced dissociation of peptides using a time of flight mass analyzer. *Proteomics* 3, 847–850. Available at: <https://doi.org/10.1002/pmic.200300362>.
- Rigaut, G., Shevchenko, A., Rutz, B., *et al.*, 1999. A generic protein purification method for protein complex characterization and proteome exploration. *Nat. Biotechnol.* 17, 1030–1032. Available at: <https://doi.org/10.1038/13732>.
- Riley, N.M., Coon, J.J., 2016. Phosphoproteomics in the age of rapid and deep proteome profiling. *Anal. Chem.* 88, 74–94. Available at: <https://doi.org/10.1021/acs.analchem.5b04123>.
- Rison, S.C.G., Mattow, J., Jungblut, P.R., Stoker, N.G., 2007. Experimental determination of translational starts using peptide mass mapping and tandem mass spectrometry within the proteome of *Mycobacterium tuberculosis*. *Microbiology* 153, 521–528. Available at: <https://doi.org/10.1099/mic.0.2006/001537-0>.

- Robin, T., 2018. Proteomics Data Representation and Databases. In: Reference Module in Life Sciences. Elsevier. Available at: <https://doi.org/10.1016/B978-0-12-809633-8.20273-2>.
- Roepstorff, P., Fohlman, J., 1984. Letter to the editors. *Biol. Mass Spectrom.* 11 (11), 601. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1002/bms.1200111109>.
- Rogowska-Wrzęsinska, A., Le Bihan, M.-C., Thaysen-Andersen, M., Roepstorff, P., 2013. 2D gels still have a niche in proteomics. *J. Proteomics* 88, 4–13. Available at: <https://doi.org/10.1016/j.jprot.2013.01.010>.
- Rombouts, Y., Jónasdóttir, H.S., Hipgrave Ederveen, A.L., *et al.*, 2016. Acute phase inflammation is characterized by rapid changes in plasma/peritoneal fluid N-glycosylation in mice. *Glycoconj. J.* 33, 457–470. Available at: <https://doi.org/10.1007/s10719-015-9648-9>.
- Ross, P.L., Huang, Y.N., Marchese, J.N., *et al.*, 2004. Multiplexed protein quantitation in *Saccharomyces cerevisiae* using amine-reactive isobaric tagging reagents. *Mol. Cell. Proteomics* 3, 1154–1169. Available at: <https://doi.org/10.1074/mcp.M400129-MCP200>.
- Ruggles, K.V., Krug, K., Wang, X., *et al.*, 2017. Methods, tools and current perspectives in proteogenomics. *Mol. Cell. Proteomics* 16, 959–981. Available at: <https://doi.org/10.1074/mcp.MR117.000024>.
- Savitski, M.M., Lemeer, S., Boesche, M., *et al.*, 2011. Confident phosphorylation site localization using the mascot delta score. *Mol. Cell. Proteomics* 10. Available at: <https://doi.org/10.1074/mcp.M110.003830>.
- Schwämmle, V., Aspöf, C.-M., Sidoli, S., Jensen, O.N., 2014. Large scale analysis of co-existing post-translational modifications in histone tails reveals global fine structure of cross-talk. *Mol. Cell. Proteomics* 13, 1855–1865. Available at: <https://doi.org/10.1074/mcp.O113.036335>.
- Seeley, E.H., Caprioli, R.M., 2008. Molecular imaging of proteins in tissues by mass spectrometry. *Proc. Natl. Acad. Sci. USA* 105, 18126–18131. Available at: <https://doi.org/10.1073/pnas.0801374105>.
- Skinner, O.S., Kelleher, N.L., 2015. Illuminating the dark matter of shotgun proteomics. *Nat. Biotechnol.* 33, 717–718. Available at: <https://doi.org/10.1038/nbt.3287>.
- Smith, L.M., Kelleher, N.L., 2013. Proteoform: A single term describing protein complexity. *Nat. Methods* 10, 186–187. Available at: <https://doi.org/10.1038/nmeth.2369>.
- Snijder, J., Schuller, J.M., Wiegand, A., *et al.*, 2017. Structures of the cyanobacterial circadian oscillator frozen in a fully assembled state. *Science* 355, 1181–1184. Available at: <https://doi.org/10.1126/science.aag3218>.
- Syka, J.E.P., Coon, J.J., Schroeder, M.J., Shabanowitz, J., Hunt, D.F., 2004. Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. *Proc. Natl. Acad. Sci.* 101, 9528–9533. Available at: <https://doi.org/10.1073/pnas.0402700101>.
- Taylor, C.F., Paton, N.W., Lilley, K.S., *et al.*, 2007. The minimum information about a proteomics experiment (MIAPE). *Nat. Biotechnol.* 25, 887–893. Available at: <https://doi.org/10.1038/nbt1329>.
- Thompson, A., Schäfer, J., Kuhn, K., *et al.*, 2003. Tandem mass tags: A novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Anal. Chem.* 75, 1895–1904. Available at: <https://doi.org/10.1021/ac0262560>.
- Tiemeyer, M., Aoki, K., Paulson, J., *et al.*, 2017. GlycoCan: An accessible glycan structure repository. *Glycobiology* 1–5. Available at: <https://doi.org/10.1093/glycob/cwx066>.
- Tran, N.H., Rahman, M.Z., He, L., *et al.*, 2016. Complete de novo assembly of monoclonal antibody sequences. *Sci. Rep.* 6. Available at: <https://doi.org/10.1038/srep31730>.
- Venable, J.D., Dong, M.-Q., Wohlschlegel, J., Dillin, A., Yates, J.R., 2004. Automated approach for quantitative analysis of complex peptide mixtures from tandem mass spectra. *Nat. Methods* 1, 39–45. Available at: <https://doi.org/10.1038/nmeth705>.
- Venne, A.S., Kollipara, L., Zahedi, R.P., 2014. The next level of complexity: Crosstalk of posttranslational modifications. *Proteomics* 14, 513–524. Available at: <https://doi.org/10.1002/pmic.201300344>.
- Vlastaridis, P., Kyriakidou, P., Chaliotis, A., *et al.*, 2017. Estimating the total number of phosphoproteins and phosphorylation sites in eukaryotic proteomes. *GigaScience* 6, 1–11. Available at: <https://doi.org/10.1093/gigascience/giw015>.
- Wolters, D.A., Washburn, M.P., Yates, J.R., 2001. An automated multidimensional protein identification technology for shotgun proteomics. *Anal. Chem.* 73, 5683–5690. Available at: <https://doi.org/10.1021/ac010617e>.
- Yao, X., Freas, A., Ramirez, J., Demirev, P.A., Fenselau, C., 2001. Proteolytic ¹⁸O labeling for comparative proteomics: Model studies with two serotypes of adenovirus. *Anal. Chem.* 73, 2836–2842. Available at: <https://doi.org/10.1021/ac001404c>.
- Yates, J.R., Speicher, S., Griffin, P.R., Hunkapiller, T., 1993. Peptide mass maps: A highly informative approach to protein identification. *Anal. Biochem.* 214, 397–408. Available at: <https://doi.org/10.1006/abio.1993.1514>.
- Yilmaz, S., Shiferaw, G.A., Rayo, J., *et al.*, 2018. Cross-linked peptide identification: A computational forest of algorithms. *Mass Spectrom. Rev.* Available at: <https://doi.org/10.1002/mas.21559>.
- Zhang, Z., Burke, M., Mirokhin, Y.A., *et al.*, 2018. Reverse and random decoy methods for false discovery rate estimation in high mass accuracy peptide spectral library searches. *J. Proteome Res.* 17, 846–857. Available at: <https://doi.org/10.1021/acs.jproteome.7b00614>.
- Zubarev, R.A., Kelleher, N.L., McLafferty, F.W., 1998. Electron capture dissociation of multiply charged protein cations. A nonergodic process. *J. Am. Chem. Soc.* 120, 3265–3266. Available at: <https://doi.org/10.1021/ja973478k>.

Relevant Websites

- <http://totalab.com/2d-products>
2D Products - TotalLab - Supporting your research.
- <http://www.crapome.org>
Crapome.org.
- <https://www.gelmap.de>
Gelmap. Spot visualization by LUH.
- http://www.matrixscience.com/help/top_down_help.html
Mascot database search: Top-down searches - Matrix Science.
- <http://2d-gel-analysis.com>
Melanie 2D gel analysis software for protein expression profiling.
- <https://ms-imaging.org>
MS Imaging—Home of Mass Spectrometry Imaging.
- <https://www.openms.de>
OpenMS.
- <http://peptide.nist.gov/>
Peptide Mass Spectral Libraries.
- <https://www.phosphosite.org>
PhosphoSitePlus.
- <http://proteowizard.sourceforge.net>
ProteoWizard.

<http://www.peptideatlas.org/speclib/>

Spectrum Library Central at PeptideAtlas.

<http://www.sorfs.org>

sORFs.org: repository of small ORFs identified by ribosome profiling.

<https://world-2dpage.expasy.org>

The World-2DPAGE Constellation - ExPASy.

Proteomics Data Representation and Databases

Thibault Robin, University of Geneva, CUI, Carouge, Switzerland

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteomics distinguishes itself from other life sciences by the variety and the complexity of the data generated. Compared to genomics, many biological events such as post-translational modifications, protein isoforms, and protein degradation add challenges to data analysis and interpretation (Altelaar *et al.*, 2012). Furthermore, proteins rarely act independently but rather interact with each other in large intricate, dynamic and plastic networks (Baker, 2012; Altelaar *et al.*, 2012).

In recent years, significant breakthroughs were achieved in proteomics technologies (Altelaar *et al.*, 2012). Mass spectrometry (MS) in particular has emerged as the reference approach for the high-throughput identification of peptides and proteins. This progress was made possible through improvements in sample preparation, instrumentation and computational tools (Altelaar *et al.*, 2012). The widespread adoption of next-generation mass spectrometers by the proteomics community yielded to a massive increase in the amount of data generated, raising concerns about their storage and sharing policies. Despite some initial reluctance, the need for open access data in proteomics is nowadays fully embraced by the community. Although data sharing was introduced a long time ago in other omics fields, it was not effortless in proteomics and required a collective input over the years to convince both researchers and instrument manufacturers (Prince *et al.*, 2004; Verheggen and Martens, 2015; Deutsch *et al.*, 2017).

To meet the demand, a large panel of databases with their respective underlying purpose and content specificities was developed over time. Despite the undeniable biological value provided, the resulting diversity can prove to be troublesome for users who have to grasp the extent of the various data formats for both input and output files. Another consequence of such fragmented information is that long-term database maintenance is not guaranteed. Some databases may be financially impacted when the only source of income lies in research grants and donations. A recent setback was the successive discontinuation of the Peptidome (2011) (Slotta *et al.*, 2009; Csordas *et al.*, 2013) and Tranche (2013) (Smith *et al.*, 2011; Science Signaling, 2013) databases due to the lack of funding, resulting in the partial loss of each dataset (Deutsch *et al.*, 2017). These incidents led the ProteomeXchange consortium to make data sustainability one of its priorities (Deutsch *et al.*, 2017).

This article will describe the proteomics field in regard to the available databases and the data they contain. First, the principal data types are presented along with the corresponding standard formats that were established through the years. Next, the main categories of databases are detailed and illustrated, with representative examples that have the greatest impact on the field today. Then, the ProteomeXchange consortium is detailed to present the significant contribution it brought to the proteomics field. Finally, the prospects of proteomics data and databases are examined.

Data Types and Standard Formats

The data generated in most MS-based proteomics experiments can be divided into two main categories depending on their origin: raw data and processed data. Metadata is a separate though significant category, present throughout the analysis process. Its purpose is to store and track a wide range of technical and biological information, essential to proper data handling and interpretation. Metadata information is especially essential in the proteomics field, since a broad range of approaches and technologies are being used (Table 1).

Raw Data

In its primal form, proteomics raw data is mass spectra. A mass spectrum consists at its core in a collection of mass-to-charge ratios (m/z) plotted against their corresponding intensities (Aebersold and Mann, 2003). Supplementary information is usually provided, such as the precursor ion m/z value and charge along with the retention time when a chromatographic separation step was performed.

Proprietary formats

Mass spectrometers usually produce raw data in the form of proprietary binary files presenting various levels of compression (Martens *et al.*, 2005b). Most of the time, the exact specifications of the format encoding-schemes are not publicly available. Proprietary binary files cannot consequently be accessed by users without software supplied by the instrument vendor. Likewise, third-party software, such as MSConvert from the ProteoWizard suite (Kessner *et al.*, 2008), will require the implementation of specific proprietary libraries to decode the information contained in the data files. The data formats also change drastically from one vendor to another, or even sometimes between instruments, raising important standardization concerns. Despite all the restrictions, proprietary formats contain the greatest amount of information and present overall good performances when being processed by compatible software.

Table 1 Summary of the main open data formats in the proteomics field

Name	Supported Data	Organization	URL
mzData	Raw mass spectra	Proteomics Standards Initiative, Human Proteome Organization	http://www.psdev.info/mzdata
mzXML	Raw mass spectra	Seattle Proteome Center, Institute for Systems Biology	http://tools.proteomecenter.org/wiki/index.php?title=Formats:mzXML
mzML	Raw mass spectra	Proteomics Standards Initiative, Human Proteome Organization	http://www.psdev.info/mzml
mz5	Raw mass spectra	Proteomics Center, Boston Children's Hospital	http://software.steenlab.org/mz5
TraML	SRM transition lists	Proteomics Standards Initiative, Human Proteome Organization	http://www.psdev.info/traml
mzIdentML	Identification data	Proteomics Standards Initiative, Human Proteome Organization	http://www.psdev.info/mzidentml
mzQuantML	Quantification data	Proteomics Standards Initiative, Human Proteome Organization	http://www.psdev.info/mzquantml
pepXML	Identification and quantification data	Seattle Proteome Center, Institute for Systems Biology	http://tools.proteomecenter.org/wiki/index.php?title=Formats:pepXML
protXML	Identification and quantification data	Seattle Proteome Center, Institute for Systems Biology	http://tools.proteomecenter.org/wiki/index.php?title=Formats:protXML
mzTab	Identification and quantification data	Proteomics Standards Initiative, Human Proteome Organization	http://www.psdev.info/mztab

Peak list formats

The limitations of the proprietary formats favored the common practice of extracting the mass spectra from the binary files as peak lists stored in text files at the cost of an overall reduced amount of information. In addition to the almost complete loss of metadata information, the mass spectra usually go through denoising, deisotoping, centroiding and charge deconvolution steps during the format conversion (Martens *et al.*, 2005b). The resulting peak list files are therefore significantly more compact while becoming easily manageable for users. The lack of metadata represents the main shortcoming of this format, making it poorly adapted to data storing and sharing.

Many peak list formats were developed through the years, such as the Mascot generic format (MGF), the Micromass peak lists (PKL) and the SEQUEST files (DTA). Despite slight variations in the amount of information they contain, these formats are easily inter-convertible without significant data loss using publicly available software (Martens *et al.*, 2005b).

mzData and mzXML formats

In order to determine a middle solution between proprietary binary files and basic peak lists, new open data formats were proposed at the beginning of the 2000s owing to the expansion of XML (Extensible Markup Language) in bioinformatics. For several years, the mzData (HUPO-PSI, 2006) and mzXML (Pedrioli *et al.*, 2004) formats used to be the most common open data formats to store and share raw data information (Martens *et al.*, 2011). Although sharing the same XML architecture, the two formats mainly differed in regard to their ontology policies (Martens *et al.*, 2011).

On the one hand, the mzData format was developed by the HUPO Proteomics Standards Initiative (PSI) as an open format standard to share and archive data. Its metadata annotation was based on a controlled vocabulary that could be regularly updated (Martens *et al.*, 2011). On the other hand, the mzXML format was developed at the Institute for Systems Biology (ISB) as an open data format accompanied by a suite of tools (Pedrioli *et al.*, 2004). Its metadata annotation was based on a fixed schema that required modification along with the corresponding software to incorporate new annotations (Martens *et al.*, 2011). Although nowadays deprecated, the mzData and mzXML formats are still compatible with most current software and are being used at a smaller scale by the proteomics community.

mzML format

With the purpose of format unification, the mzML format was developed as part of a collective effort under the guidance of the HUPO-PSI to replace both the mzData and mzXML formats (Martens *et al.*, 2011). The coexistence of two open formats having similar functionality was perceived at the time as confusing for users. The goal was then to design a unique XML-based format supporting all past features while providing a consensus way to encode the information (Martens *et al.*, 2011). In addition to instrument support improvement, the mzML format features notably metadata annotation following a flexible controlled vocabulary that allows the inclusion of custom new terms by users without requiring the modification of the XML schema (Martens *et al.*, 2011). The mzML format quickly became compatible with most tools available in proteomics, strengthening its format unification goal.

mz5 format

More recently, the mz5 open data format was developed at the Proteomics Center of the Boston Children's Hospital with the aim of providing a more speedy and storage efficient alternative to mzML while preserving the same ontology (Wilhelm *et al.*, 2012).

Distinguishing itself from its predecessors, the mz5 format is designed based on the Hierarchical Data Format (HDF5) instead of the traditional XML architecture. Despite the partial base-64 encoding being used in mzML, XML was designed to remain easily manipulable by users and consequently struggles to handle massive data files. With data benchmarks, the mz5 format was shown to increase 3–4 times read and write performances while halving data file size in comparison to its XML-based counterparts (Wilhelm *et al.*, 2012). Despite its obvious benefits, the low amount of compatible software prevented the widespread adoption of the mz5 format by the proteomics community.

Processed Data

In most high-throughput proteomics experiments, peptide and protein identification is achieved through the database search approach (Nesvizhskii, 2006). In this method, the experimental spectra are aligned against theoretical spectra inferred from the fragmentation patterns of the peptides contained in a reference protein sequence database. The tools performing the search, referred to as search engines, additionally provide numerous statistical scores assessing the quality of the resulting peptide-spectrum matches (PSMs), allowing to rank the peptides matching a given experimental spectrum (Nesvizhskii, 2006). In a further step, the identified peptides are used to determine the proteins from which they originated. Protein inference can however prove to be challenging, especially in eukaryote species, because peptides can belong to several distinct proteins and multiple protein isoforms may coexist (Nesvizhskii, 2005). The notion of proteotypic peptide was then introduced to single out those peptides that are unique to a protein.

Protein identification is the main task in most MS data processing workflows. Nonetheless, protein quantification is also frequently performed. Quantification experiments usually have the aim of comparing protein expression across multiple samples prepared in distinct conditions, although absolute quantification of proteins can also be achieved. Two mutually exclusive strategies have been established through time for quantification in proteomics: stable isotope labeling and label-free quantification (Bantscheff *et al.*, 2007). In stable isotope labeling approaches such as SILAC (Ong *et al.*, 2002), a mass tag is inserted in proteins through a wide range of experimental protocols. The induced mass shift between the light and heavy forms of peptides is detectable by mass spectrometers, leading to the relative quantification of the proteins of interest through the comparison of the signal intensities (Bantscheff *et al.*, 2007). The label-free approaches expanded more recently with the emergence of more accurate instruments. Being usually based either on spectral counting or on precursor ion intensity, label-free approaches do not include any isotope-tagging step and consequently reduce the sample preparation complexity (Bantscheff *et al.*, 2007).

PepXML and protXML formats

The pepXML and protXML formats (Keller *et al.*, 2005) were originally developed with the purpose of improving and facilitating data transfer between the various tools being part of the Trans-Proteomic Pipeline (TPP) (Keller *et al.*, 2005). The pepXML format stores and shares peptide identification and quantification data while the protXML format focuses solely on the corresponding protein data (Deutsch *et al.*, 2015). Some search engines can directly output their results into files in the pepXML format to be further processed by the TPP. Converter tools are also available in the TPP for the other formats. Although the pepXML and protXML formats were not defined with the underlying aim of becoming official standard formats, many tools outside the TPP adopted them over the years (Deutsch *et al.*, 2015).

mzIdentML and mzQuantML formats

To provide a reliable open standard format for the distribution of peptide and protein identification data, mzIdentML was developed by the HUPO-PSI alongside with a suite of converters for most open and proprietary formats available (Jones *et al.*, 2012). As in the case of pepXML and protXML, search engines that evolve over the years offer the option to export directly their results in the mzIdentML format (Jones *et al.*, 2012).

Based on the success of mzIdentML, the mzQuantML format was launched shortly thereafter under the direction of the HUPO-PSI as a quantitative-focused counterpart (Walzer *et al.*, 2013). It was developed to answer the pressing need for a standard format in quantitative proteomics, especially since the techniques and approaches may diverge significantly between individual experiments (Walzer *et al.*, 2013). The mzQuantML format notably features the ability to cross-reference other XML-based open formats such as mzML and mzIdentML. These cross-links help to keep track of the numerous data and metadata produced throughout the analysis workflow (Walzer *et al.*, 2013). Both formats are based on the same controlled vocabulary that was introduced by the HUPO-PSI for mzML (Jones *et al.*, 2012; Walzer *et al.*, 2013), thus providing a robust yet flexible way to annotate metadata information.

mzTab format

The mzTab format was recently launched by the HUPO-PSI as a complement to both the mzIdentML and mzQuantML formats (Griss *et al.*, 2014). The underlying purpose was to design a simple tabular format summarizing identification and quantification data from proteomics and metabolomics experiments. As the strength of the format resides in the easy report of experimental results, it usually contains an overall lower amount of information compared to most XML-based formats (Griss *et al.*, 2014). Files in the mzTab format do not require specific parsing software since no special encoding is involved, which significantly simplifies their handling by users.

TraML format

Through the years, more specialized open formats also made their apparition in the proteomics field. The TraML (Deutsch *et al.*, 2012) standard format developed by the HUPO-PSI is such an example, capturing solely the transition lists resulting from selected reaction monitoring (SRM) experiments. In SRM approaches, a transition represents the pair of m/z values of the targeted peptide precursor ion and of one of its corresponding fragment ions. TraML is designed around the same principles that were established for the mzML and mzIdentML formats, using an XML-based architecture along with a controlled vocabulary encoding the metadata information (Deutsch *et al.*, 2012).

Databases and Repositories

Databases in the proteomics field differ strongly in terms of both their purpose and the data they contain. While some solely serve as repositories for the dissemination of experimental datasets, others have been developed with more advanced functions such as sequence annotation or targeted research goals.

Proteomics databases and repositories are still under substantial development in comparison to other life science fields. This delay can partially be explained by the initial skepticism of a part of the proteomics community in regard to sharing publicly experimental data due to data submission imposed by publishers. One of the common concerns was the possible third-party reanalysis of datasets prior to result release in a publication of the original submitter. To address this issue, most databases feature nowadays the ability to keep dataset submission private for the duration of the reviewing process (Vizcaíno *et al.*, 2014).

Needless to say, UniProt (The UniProt Consortium, 2017) remains arguably the most popular and internationally renowned resource for proteins, playing a central role in the proteomics database landscape as the ever-present reference for protein annotation. As UniProt is however a knowledge database that contains almost to no MS data, it will not be presented in details here (Table 2). Likewise, the species-specific resources will also not be discussed in this article.

GPMdb

The Global Proteome Machine database (Craig *et al.*, 2004) (GPMdb) was initially designed by Beavis Informatics for the purpose of storing the reprocessed data produced by the GPM data analysis servers. In this automated pipeline, the MS raw data submitted by users or contained in other databases are reanalyzed using the online version of the X!Tandem (Craig and Beavis, 2004) search engine. The input files submitted to the GPMdb have to be formatted either as peak list (DTA, PKL or MGF) or in a compatible open standard format (mzXML or mzData). In addition to a large panel of search parameters, the users can choose prior to the database search if they authorize the inclusion of the dataset to the GPMdb. With the accumulation of a large amount of data over the years, the GPMdb has diversified its functions and is today commonly used by researchers to compile spectral libraries and develop related search tools. This notably led to the inception of the X!Hunter (Craig *et al.*, 2006) online search engine, which can perform the search of experimental mass spectra against consensus spectral libraries built from the GPMdb.

PeptideAtlas

The PeptideAtlas database (Desiere *et al.*, 2006) was originally developed in 2004 at the Institute for Systems Biology (ISB) as a resource providing genome annotations from the automatic reprocessing of MS raw data by the TPP pipeline. PeptideAtlas presents the peculiarity of being composed of distinct builds according to the organism or tissue of interest (Deutsch *et al.*, 2008). Another noteworthy feature of PeptideAtlas consists in the computation of an observability score for each proteotypic peptide represented in the database (Deutsch *et al.*, 2008). This offers the option of determining the peptides that are the most likely to be detected in a MS experiment for a given protein. As in the case of the GPMdb, the PeptideAtlas database surpassed its initial annotation purpose and is nowadays frequently used as a research database for the compilation of spectral libraries.

Table 2 Summary of the main public databases and repositories in the proteomics field

Name	Supported Data	Organization	URL
PRIDE Archive	Raw and processed data	European Molecular Biology Laboratory, European Bioinformatics Institute	https://www.ebi.ac.uk/pride/archive
MassIVE	Raw and processed data	Center for Computational Mass Spectrometry, University of California San Diego	https://massive.ucsd.edu
jPOSTrepo	Raw and processed data	National Bioscience Data Center, Japan Science and Technology Agency	http://jpostdb.org
PASSEL	SRM raw data	Seattle Proteome Center, Institute for Systems Biology	http://www.peptideatlas.org/passel
GPMdb	Processed data	Beavis Informatics	http://gpmdb.thegpm.org
PeptideAtlas	Processed data	Seattle Proteome Center, Institute for Systems Biology	http://www.peptideatlas.org

PASSEL

The PeptideAtlas SRM Experiment Library (PASSEL) was developed in the framework of the PeptideAtlas project (Farrah *et al.*, 2012). It was designed as a receiving repository for data produced in SRM experiments. In addition, the raw data submitted to PASSEL are automatically reprocessed by specific tools such as mQuest from the mProphet suite (Reiter *et al.*, 2011). The results of this workflow are stored in a separate database that can be accessed through the PASSEL web interface. Among all the obtained transitions, those considered to be the best based on quality metrics are included in the SRMATlas database (Picotti *et al.*, 2008).

PRIDE/PRIDE Archive

The Proteomics Identifications (PRIDE) database (Martens *et al.*, 2005a) was established in 2004 at the European Bioinformatics Institute (EBI) as a structured repository designed to propagate experimental proteomics data. The underlying intent was to provide a resource that would keep the submitted data intact, preserving it from any kind of control or alteration. PRIDE contains both the raw MS data and the resulting peptide and protein identifications, along with the corresponding biological and experimental metadata (Martens *et al.*, 2005a). As of 2014, the PRIDE database was entirely redesigned and replaced by PRIDE Archive (Vizcaíno *et al.*, 2016). The most significant structural changes brought by PRIDE Archive lie in a new data storage architecture and submission system along with a reworked web interface (Vizcaíno *et al.*, 2016).

The Peptidome database (Slotta *et al.*, 2009) was a public data repository developed in 2009 at the National Center for Biotechnology Information (NCBI) that aimed to allow the exchange of experimental proteomics data. Peptidome was thus closely related to PRIDE in terms of both its aim and features. After the discontinuation of Peptidome in 2011 due to lack of funds, a coordinated effort of both teams led to transfer most of its content to the PRIDE repository (Csordas *et al.*, 2013).

MassIVE

The Mass spectrometry Interactive Virtual Environment (MassIVE) repository was developed at the Center for Computational Mass Spectrometry (CCMS) as a public data repository for datasets produced in MS-based proteomics experiments. Compared to other repositories, MassIVE aims to enhance the interactivity of its content. A noteworthy feature is the ability to add comments and reanalyzes for a given submitted dataset. MassIVE also provides the possibility to directly reprocess the raw data that were submitted using several online workflows.

Tranche (Smith *et al.*, 2011) was a public data repository developed in 2005 as part of the Proteome Commons network that aimed to transfer proteomics raw datasets at a large scale. Since the repository had to handle a high data traffic in relation with the large file size, a special effort was made in the design of an efficient peer-to-peer infrastructure coupled with a reliable client-server architecture (Smith *et al.*, 2011). After the funding shortfall that led to the discontinuation of Proteome Commons and Tranche in 2013 (Science Signaling, 2013), as many data sets as possible were recovered and transferred to MassIVE (Deutsch *et al.*, 2017).

jPOST

The Japan Proteome Standard (jPOST) is an emerging resource that ultimately aims to provide both a repository and a database for the worldwide proteomics community. On the one hand, the jPOST repository (jPOSTrepo) (Okuda *et al.*, 2017) was launched at the National Bioscience Data Center (NBDC) in 2015 as a platform for the global exchange of datasets produced in MS-based proteomics experiments. The underlying aim was to offer an alternative for the Asia and Oceania regions to the existing proteomics repositories. Indeed, most public MS data resources are either hosted in Europe (PRIDE Archive) or in the United States (PASSEL, MassIVE). This can prove to be problematic in terms of data transfer speed for researchers who agree to share their experimental datasets but are located far away from these regions. A special effort was thus additionally undertaken to deploy an efficient file upload system, further facilitating data exchange (Okuda *et al.*, 2017).

On the other hand, the jPOST database (jPOSTdb) is still under development and has not been made publicly available yet. Eventually, jPOSTdb will include the automatic reprocessing of the MS raw data submitted to jPOSTrepo through a customized workflow using a combination of several search engines.

ProteomeXchange Consortium

The ProteomeXchange (PX) consortium (Vizcaíno *et al.*, 2014) was originally formed in 2006 and officially launched in 2011 as a collective effort to coordinate data exchange and submission between MS-based proteomics databases and repositories. Since the partner resources had been developed independently at distinct research institutions, it resulted in limited coordination in regard to their respective data submission guidelines and policies (Vizcaíno *et al.*, 2014). This proved to be troublesome for researchers willing to share their experimental data, struggling to determine where and in which form datasets should be submitted. Likewise, data comparison through the use of several sources could turn out to be challenging. The purpose of the PX consortium was to remove these obstacles by providing a regulated infrastructure and framework (Vizcaíno *et al.*, 2014). Such an initiative had already been successfully implemented in other omics fields, as for instance with the development in genomics of the International Nucleotide Sequence Database Collaboration (INSDC) (Cochrane *et al.*, 2016) to regulate the dissemination of DNA sequencing data.

In its first inception, the PX consortium had only two core members: PRIDE and PeptideAtlas (along with its PASSEL resource) (Vizcaíno *et al.*, 2014). PRIDE was used at the time as the unique entry point for data produced in shotgun proteomics experiments, whereas SRM data were redirected to PASSEL. In the pursuit of its unification role, more receiving repositories progressively joined the PX consortium in recent years. This led notably to the inclusion of the MassIVE repository in 2014, followed shortly by the addition of jPOST in 2016 (Deutsch *et al.*, 2017). More emerging resources are also expected to become members of the PX consortium in the future, further improving the state of data sharing in the proteomics field.

Data submission in the PX consortium can be subdivided in two distinct workflows: complete and partial (Deutsch *et al.*, 2017). In a complete submission, the format of all the submitted files can be recognized and parsed by the receiving repository. It is the recommended submission process in most cases, as it allows searching directly through the results with precise queries. A partial submission contains files that are in a non-compatible format, thus preventing the receiving repository to read them. As with the complete submissions, the files can however still be searched using the associated submission metadata. Despite its limitations, the partial submission process remains essential to support the inclusion of datasets coming from a broad range of proteomics experiments, including marginal or emerging techniques (Deutsch *et al.*, 2017).

Both complete and partial workflows require the submission of the raw data, the processed data and the corresponding biological and technical metadata information. The latter is encoded in an XML format that was specifically developed for the PX consortium (PX XML) (Vizcaíno *et al.*, 2014). After submission, each dataset is assigned a unique PX identifier. A receiving repository specific identifier is additionally assigned, except for PRIDE (Deutsch *et al.*, 2017). If a dataset was submitted as complete, a digital object identifier (DOI) is also created to allow proper referencing in case of reanalysis in another project (Vizcaíno *et al.*, 2014). Since data reuse is also one of the main focuses of the PX consortium, distinct identifiers are specifically generated to track reprocessed datasets and store corresponding results (Deutsch *et al.*, 2017).

The publicly available datasets stored in the resources of the PX consortium are accessed and queried through the ProteomeCentral online portal, which was deployed from the first version. ProteomeCentral notably features the ability to filter the datasets using metadata keywords, which can be for instance used to retrieve solely the set of results matching a given organism, tissue or MS instrument (Deutsch *et al.*, 2017). A RSS feed was also set up to push updates about new publicly available datasets that are released in the PX consortium, providing links to both the ProteomeCentral entry and the corresponding PX XML file storing the submission metadata (Vizcaíno *et al.*, 2014).

Despite the essential contribution brought by PX consortium to the proteomics field, there is still room for improvement. Some work remains for example to be done regarding the compatibility of complete submission with data produced by increasingly popular experimental approaches, such as the data independent acquisition, top-down proteomics or MS-based imaging techniques (Deutsch *et al.*, 2017). The MassIVE repository already started this process by becoming compatible with some DIA-based workflows (Deutsch *et al.*, 2017). The PX consortium is arguably the most impactful initiative that occurred in recent years for databases and repositories in proteomics, bringing together the different resources in the field and standardizing data submission, processing and dissemination.

Conclusion and Prospects

The successful outcome of collective efforts such as the ProteomeXchange initiative has significantly boosted data sharing in the proteomics field. An obvious consequence is the increasing number of journal editors who nowadays mandate the deposition of the original raw datasets to enhance the publication of scientific articles and allow for evidence checks. Further into the future, more integration of the different omics fields is to be expected. Progress has already been made in this direction with the recent development of the Omics Discovery Index (OmicsDI) portal (Perez-Riverol *et al.*, 2016). OmicsDI aims to provide an online platform for the dissemination and access of datasets from genomics, transcriptomics, metabolomics and proteomics resources. Tighter collaboration between proteomics and metabolomics is also likely, based in particular on the fact that both fields rely predominantly on MS-based approaches and techniques to produce their data. Some open standard formats such as mzTab have already started to become compatible with metabolomics workflows.

Acknowledgement

The author would like to thank the Dr. Frédérique Lisacek and Emma Ricart Altimiras for their support.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Clinical Proteomics. Data Storage and Representation. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Identification of Proteins from Proteomic Analysis. Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Protein Structure Databases. Proteomics Mass Spectrometry Data Analysis Tools. Quantification of Proteins from Proteomic Analysis. Standards and Models for Biological Data: FGED and HUPO. Supervised Learning: Classification. The Evolution of Protein Family Databases

References

- Aebersold, R., Mann, M., 2003. Mass spectrometry-based proteomics. *Nature* 422 (6928), 198–207.
- Altelaar, A.F.M., Munoz, J., Heck, A.J.R., 2012. Next-generation proteomics: Towards an integrative view of proteome dynamics. *Nature Reviews Genetics* 14 (1), 35–48.
- Baker, M., 2012. Proteomics: The interaction map. *Nature* 484 (7393), 271–275.
- Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., Kuster, B., 2007. Quantitative mass spectrometry in proteomics: A critical review. *Analytical and Bioanalytical Chemistry* 389 (4), 1017–1031.
- Cochrane, G., Karsch-Mizrachi, I., Takagi, T., International Nucleotide Sequence Database Collaboration, 2016. The international nucleotide sequence database collaboration. *Nucleic Acids Research* 44 (D1), D48–D50.
- Craig, R., Beavis, R.C., 2004. TANDEM: Matching proteins with tandem mass spectra. *Bioinformatics* 20 (9), 1466–1467.
- Craig, R., Cortens, J.P., Beavis, R.C., 2004. Open source system for analyzing, validating, and storing protein identification data. *Journal of Proteome Research* 3 (6), 1234–1242.
- Craig, R., Cortens, J.C., Fenyo, D., Beavis, R.C., 2006. Using annotated peptide mass spectrum libraries for protein identification. *Journal of Proteome Research* 5 (8), 1843–1849.
- Csordas, A., Wang, R., Ríos, D., *et al.*, 2013. From peptidome to PRIDE: Public proteomics data migration at a large scale. *Proteomics* 13 (10–11), 1692–1695.
- Desiere, F., Deutsch, E.W., King, N.L., *et al.*, 2006. The PeptideAtlas project. *Nucleic Acids Research* 34 (Database issue), D655–D658.
- Deutsch, E.W., Chambers, M., Neumann, S., *et al.*, 2012. TraML – A standard format for exchange of selected reaction monitoring transition lists. *Molecular & Cellular Proteomics* 11 (4), (R111.015040–R111.015040).
- Deutsch, E.W., Csordas, A., Sun, Z., *et al.*, 2017. The ProteomeXchange consortium in 2017: Supporting the cultural change in proteomics public data deposition. *Nucleic Acids Research* 45 (D1), D1100–D1106.
- Deutsch, E.W., Lam, H., Aebersold, R., 2008. PeptideAtlas: A resource for target selection for emerging targeted proteomics workflows. *EMBO Reports* 9 (5), 429–434.
- Deutsch, E.W., Mendoza, L., Shteynberg, D., *et al.*, 2015. Trans-Proteomic Pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics. *PROTEOMICS – Clinical Applications* 9 (7–8), 745–754.
- Farrah, T., Deutsch, E.W., Kreisberg, R., *et al.*, 2012. PASSEL: The PeptideAtlas SRM experiment library. *Proteomics* 12 (8), 1170–1175.
- Griss, J., Jones, A.R., Sachsenberg, T., *et al.*, 2014. The mzTab data exchange format: Communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience. *Molecular & Cellular Proteomics* 13 (10), 2765–2775.
- HUPO-PSI, 2006. The mzData standard. Available at: <http://www.psdev.info/mass-spectrometry#mzdata> (accessed 27.03.17).
- Jones, A.R., Eisenacher, M., Mayer, G., *et al.*, 2012. The mzIdentML data standard for mass spectrometry-based proteomics results. *Molecular & Cellular Proteomics* 11 (7), (M111.014381–M111.014381).
- Keller, A., Eng, J., Zhang, N., Li, X.K., Aebersold, R., 2005. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Molecular Systems Biology* 1 (1), E1–E8.
- Kessner, D., Chambers, M., Burke, R., Agus, D., Mallick, P., 2008. ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* 24 (21), 2534–2536.
- Martens, L., Chambers, M., Sturm, M., *et al.*, 2011. mzML – A community standard for mass spectrometry data. *Molecular & Cellular Proteomics* 10 (1), doi:10.1074/mcp.R110.000133.
- Martens, L., Hermjakob, H., Jones, P., *et al.*, 2005a. PRIDE: The proteomics identifications database. *Proteomics* 5 (13), 3537–3545.
- Martens, L., Nesvizhskii, A.I., Hermjakob, H., *et al.*, 2005b. Do we want our data raw? Including binary mass spectrometry data in public proteomics data repositories. *Proteomics* 5 (13), 3501–3505.
- Nesvizhskii, A.I., 2005. Interpretation of shotgun proteomic data: The protein inference problem. *Molecular & Cellular Proteomics* 4 (10), 1419–1440.
- Nesvizhskii, A.I., 2006. Protein identification by tandem mass spectrometry and sequence database searching. *Mass Spectrometry Data Analysis in Proteomics*. New Jersey, NJ: Humana Press, pp. 87–120.
- Okuda, S., Watanabe, Y., Moriya, Y., *et al.*, 2017. jPOSTrepo: An international standard data repository for proteomes. *Nucleic Acids Research* 45 (D1), D1107–D1111.
- Ong, S.-E., Blagoev, B., Kratchmarova, I., *et al.*, 2002. Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. *Molecular & Cellular Proteomics* 1 (5), 376–386.
- Pedrioli, P.G.A., Eng, J.K., Hubley, R., *et al.*, 2004. A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology* 22 (11), 1459–1466.
- Perez-Riverol, Y., Bai, M., Leprevost, F., Squizzato, S., *et al.*, 2016. Omics Discovery Index—Discovering and Linking Public Omics Datasets.
- Picotti, P., Lam, H., Campbell, D., *et al.*, 2008. A database of mass spectrometric assays for the yeast proteome. *Nature Methods* 5 (11), 913–914.
- Prince, J.T., Carlson, M.W., Wang, R., Lu, P., Marcotte, E.M., 2004. The need for a public proteomics repository. *Nature Biotechnology* 22 (4), 471–472.
- Reiter, L., Rinner, O., Picotti, P., *et al.*, 2011. mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nature Methods* 8 (5), 430–435.
- Science Signaling, 2013. ST NetWatch: Protein Databases. Available at: <http://stke.sciencemag.org/resources/st-netwatch-archive/st-netwatch-protein-databases> (accessed 23.03.17).
- Slotta, D.J., Barrett, T., Edgar, R., 2009. NCBI peptidome: A new public repository for mass spectrometry peptide identifications. *Nature Biotechnology* 27 (7), 600–601.
- Smith, B.E., Hill, J.A., Gjukich, M.A., Andrews, P.C., 2011. Tranche distributed repository and ProteomeCommons.org. In: Hamacher, M., Eisenacher, M., Stephan, C (Eds.), *Data Mining in Proteomics*. Totowa, NJ: Humana Press, pp. 123–145.
- The UniProt Consortium, 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Verheggen, K., Martens, L., 2015. Ten years of public proteomics data: How things have evolved, and where the next ten years should lead us. *EuPA Open Proteomics* 8, 28–35.
- Vizcaino, J.A., Csordas, A., del-Toro, N., *et al.*, 2016. 2016 update of the PRIDE database and its related tools. *Nucleic Acids Research* 44 (D1), D447–D456.
- Vizcaino, J.A., Deutsch, E.W., Wang, R., *et al.*, 2014. ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nature Biotechnology* 32 (3), 223–226.
- Walzer, M., Qi, D., Mayer, G., *et al.*, 2013. The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics. *Molecular & Cellular Proteomics* 12 (8), 2332–2340.
- Wilhelm, M., Kirchner, M., Steen, J.A.J., Steen, H., 2012. mz5: Space- and time-efficient storage of mass spectrometry data sets. *Molecular & Cellular Proteomics* 11 (1), [O111.011379–O111.011379].

Further Reading

- Deutsch, E.W., 2012. File formats commonly used in mass spectrometry proteomics. *Molecular & Cellular Proteomics* 11 (12), 1612–1621.
- Martens, L., 2011. Proteomics databases and repositories. In: Wu, C.H., Chen, C (Eds.), *Bioinformatics for Comparative Proteomics*. Totowa, NJ: Humana Press, pp. 213–227.

- Martens, L., Vizcaino, J.A., 2017. A golden age for working with public proteomics data. *Trends in Biochemical Sciences* 42 (5), 333–341.
- Mayer, G., Montecchi-Palazzi, L., Ovelheiro, D., *et al.*, 2013. The HUPO proteomics standards initiative-mass spectrometry controlled vocabulary. *Database* 2013 (0), bat009.
- Perez-Riverol, Y., Alpi, E., Wang, R., *et al.*, 2015. Making proteomics data accessible and reusable: Current state of proteomics databases and repositories. *Proteomics* 15 (5–6), 930–950.
- Riffle, M., Eng, J.K., 2009. Proteomics data repositories. *Proteomics* 9 (20), 4653–4663.
- Taylor, C.F., Paton, N.W., Lilley, K.S., *et al.*, 2007. The minimum information about a proteomics experiment (MIAPE). *Nature Biotechnology* 25 (8), 887–893.
- Vaudel, M., Verheggen, K., Csordas, A., *et al.*, 2016. Exploring the potential of public proteomics data. *Proteomics* 16 (2), 214–225.
- Verheggen, K., Martens, L., 2015. Ten years of public proteomics data: How things have evolved, and where the next ten years should lead us. *EuPA Open Proteomics* 8, 28–35.
- Vizcaino, J.A., Foster, J.M., Martens, L., 2010. Proteomics data repositories: Providing a safe haven for your data and acting as a springboard for further research. *Journal of Proteomics* 73 (11), 2136–2146.

Biographical Sketch



Thibault Robin is a Ph.D. student in bioinformatics at the Proteome Informatics Group (PIG) of the Swiss Institute of Bioinformatics (SIB) affiliated with the University of Geneva. After obtaining a Bachelor degree in biology in 2013, he developed a strong interest for informatics and programming. This led to his graduation with a Master degree in bioinformatics in 2015. He is currently working on the development of new proteomics tools, focusing in particular on post-translational modifications and sequence variants.

Proteomics Mass Spectrometry Data Analysis Tools

Aivett Bilbao, Pacific Northwest National Laboratory, Richland, WA, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

The overall purpose of proteomics is the large-scale study of genes and cellular functions at the phenotype level. Information on protein abundances, their variations and modifications, and their interacting networks are critical for understanding complex biological processes, which in turn, are useful to accomplish scientific developments such as effective diagnostic techniques and disease treatments (Aebersold and Mann, 2016; Szabo and Janaky, 2015). The ability to quickly and consistently analyze the large amounts of experimental data produced in proteomics research relies on software advances (Nesvizhskii, 2010; Yates, 2015). Bioinformatics have emerged as a crucial contributor with the development of novel algorithms, the connection of various tools into complex pipelines, as well as the deposition and dissemination of both software and data. In particular, the prominence of open source software and public data in reproducible research has been recently highlighted in several publications (Deutsch *et al.*, 2017; Jiménez *et al.*, 2017; Martens and Vizcaíno, 2017; da Veiga Leprevost *et al.*, 2017; Wilkinson *et al.*, 2016).

As a specialized section of “Proteome Informatics” (Lisacek), this article provides an up-to-date and focused literature review of the software for MS-based proteomics. In particular, freely available and open source tools are cited and discussed, providing a solid starting point with relevant notions for both non-specialized end-users and bioinformaticians.

Since understanding of the data structure is critical to comprehend the processing software methods and obtaining accurate results, the first section summarizes the MS data acquisition process and spectral types currently employed in discovery or untargeted proteomics. Next, the general workflows are discussed and the community efforts to facilitate reproducible research are highlighted. In the remaining sections, algorithmic concepts related to identification, statistical assessment of error rates and quantification tools are compared with an emphasis on the most popular free and open source tools. The review concludes with a discussion of pertinent guidelines and future directions in the field.

Understanding Mass Spectrometry Data From Proteomics Analyses

Discovery of the proteins present in a sample is typically achieved by liquid-chromatography (LC) coupled to mass spectrometry (MS). In the so-called shotgun or bottom-up approach, complex protein samples are digested into peptides. The most popular enzyme used for protein digestion is trypsin, which leads to peptides with C-terminally protonated amino acids, providing an advantage in subsequent MS-based peptide sequencing (Aebersold and Mann, 2016; Aebersold, 2003). As peptides elute from the LC column, they are ionized by electrospray (ESI) and subsequent ions are analyzed by the mass spectrometer. In the traditional data-dependent acquisition (DDA) mode, the MS instrument is operated to maximize the number of selected and fragmented peptides, as they elute from the LC column. Within each DDA cycle, ion signals are recorded in a MS1 or survey scan (precursor ion signals) and the top-*n* most abundant ions are then selected and serially isolated for fragmentation (MS/MS, MS2 or tandem MS) to generate structural information. Alternatively, the mass spectrometer can be operated in data-independent acquisition mode (DIA), where all ions from peptides eluting during LC-MS analysis are fragmented and analyzed in parallel, regardless of their intensities. The advantages and disadvantages as well as specific flavors of DIA with a focus on processing software strategies are discussed in Bilbao *et al.* (2015a).

The different type of generated spectra is illustrated in Fig. 1. Data analysis tools pertinent for each spectral type are discussed in the following sections.

The Data Processing Workflow in MS-Based Proteomics

A wide range of informatics methods comprises signal processing of raw MS data, identification and quantification approaches for protein characterization, assessment and control of error rates, as well as application of statistical and machine learning techniques to unravel relevant protein targets and interacting networks in complex processes. Development and software implementation of algorithms for every step is a laborious process that requires substantial amounts of time and effort. As a result, the complete proteomics data analysis is a composite procedure involving several specialized tasks which are interconnected as a software workflow or pipeline (Fig. 2). These specialized tools have been developed and improved over the years in a stepwise fashion (Noble and MacCoss, 2012), which has greatly promoted the field but also raised major challenges when moving from single and individual tools to complex and integrated workflows: software availability and reproducible results. The challenges have historically rendered bioinformatics as the most difficult area in MS-based proteomics research (Aebersold, 2009). Multiple tools normally require substantial effort and informatics skills for correct installation and configuration (da Veiga

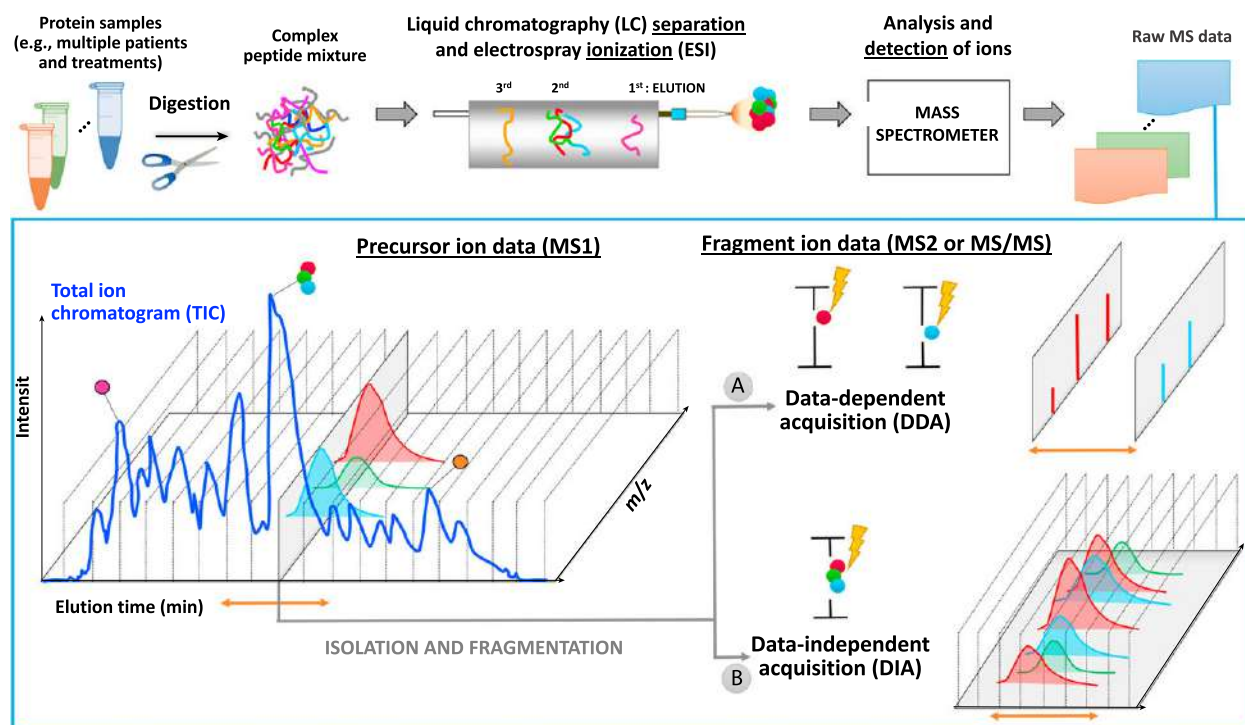


Fig. 1 Sample analysis and mass spectrometry data in discovery proteomics. Following sample preparation and enzymatic digestion, complex peptide mixtures are separated by LC and peptide ions are generated by ESI. The mass spectrometer analyses and detects the peptide ions as they elute. Files with raw MS data are generated from each sample. A file usually contains precursor (MS1) and fragment (MS2) ion spectra. Consecutive spectra in the analysis run are represented by dashed lines. A first representation of the data is the MS1 total ion chromatogram (TIC, blue trace), which is the sum of all ions detected at each time point or spectrum. Examples of MS2 spectra acquired for a specific time range are shown, depending on the MS operation mode used in the analysis run. In DDA mode, the mass spectrometer generates MS2 spectra after isolation and fragmentation of selected precursor ions. The top-2 most intense ions (red and blue LC-MS peaks) are selected to collect individual MS2 spectra at discrete time points and the low abundance one (green LC-MS peak) is undetected. Ions can be selected or not at any time point across the LC peak. In DIA mode, the mass spectrometer generates MS2 spectra of multiple co-eluting ions which are fragmented all together. Continuous fragment ion traces are recorded in convoluted or multiplexed MS2 spectra. Multiple LC-MS/MS peaks from the 3 fragmented peptide ions are detected, including the low abundance one (green LC-MS/MS peaks).

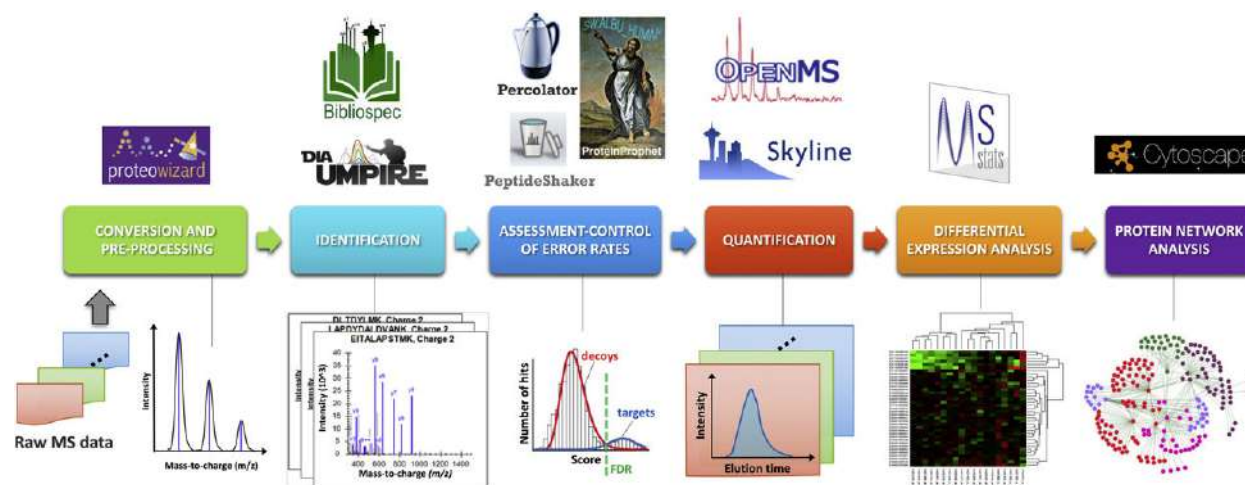


Fig. 2 General proteomics data processing workflow. Processing steps are shown as interconnected boxes. Above and below each step, available logos from example tools and visual representations are included, respectively. Some tools include a graphical user interface (GUI) or only command-line access. In the first step, raw MS data in proprietary formats from different instrument vendors are typically converted to open data formats. Pre-processing can involve centroid data generation, where multiple measurements per peak (black traces) are reduced to single centroid peaks (blue sticks). Next, identification followed by assessment and control of error rates are performed. Quantification information of confident identifications across multiple samples are used as input for downstream analyses and biological interpretations. For label-free data, quantification include a multi-sample alignment step.

Leprevost *et al.*, 2017), lack appropriate documentation and have poor or none graphical user interface (GUI) (Cappadona *et al.*, 2012).

Fortunately, software availability challenges are being recently addressed with the incorporation of advances in software engineering and the contributions from open-source and community-driven efforts.

The BioContainers framework (da Veiga Leprevost *et al.*, 2017) (see Relevant Websites Section) adopted container based technologies such as Docker or rkt to provide complete pipelines as easily installable, deployable and executable packages.

The ELIXIR Tools and Data Services Registry (see Relevant Websites Section) (Ison *et al.*, 2016), is a catalogue of bioinformatics software and resources with extensive metadata descriptions.

A useful list of free software for analysis of mass spectrometry data can be found in <http://www.ms-utils.org>.

Other efforts to provide well-documented software parameters and dependencies include workflow systems such as KNIME (Berthold *et al.*, 2009) and the Galaxy platform (Afgan *et al.*, 2016). Examples for proteomics are the OpenMS 2 tools (Aiche *et al.*, 2015) delivered as KNIME (see Relevant Websites Section), and proteomics workflows in Galaxy (see Relevant Websites Section) for OpenMS, the Trans-Proteomic Pipeline (TPP) and MaxQuant.

The other component for reproducible research, besides the software, is the data itself. Along the same lines of efforts, current requirements from scientific journals and funding agencies have promoted a change of mentality: public data sharing is a common acceptance as good scientific practice in the proteomics field, as embraced earlier in genomics and transcriptomics (Martens and Vizcaíno, 2017).

The set of principles referred to as the FAIR Data Principles that requires data to be Findable, Accessible, Interoperable, and Reusable in the long-term has been recently published as a guideline to support and enhance the reusability in contemporary e-Science (Wilkinson *et al.*, 2016).

The ProteomeXchange (PX) Consortium (see Relevant Websites Section) standardizes submission and dissemination of mass spectrometry proteomics data worldwide (Deutsch *et al.*, 2017).

Recent and detailed information about proteomics data representation and databases can be found in the complementing article “Proteomics data representation and databases”.

As a remarkable outcome of the mentioned efforts, the majority of the software tools and related data discussed in the following sections are freely available to the community. All readers, from end-users to software developers, are invited to explore and download content from the cited resources for gaining a better understanding of the tools, familiarize with the data and optimizing parameters for better results.

It is important to note that there is a myriad of tools and that new ones regularly come out. Evaluating a tool can be time-consuming and demands patience. It is common to find many tools stating to have the needed functionality. However, a deep examination is often required to determine whether the statements are true or whether the tool fits the specific user needs. It is recommended to select a tool taking into account the input and output file formats, available documentation, tutorials and test data, and related publications including application studies. It is equally important to check if the tool is actively maintained and updated, which informatics skills are needed and how scalable it is. Regarding the later point, a few files should be tried first to become familiar with the tool, adjust parameters and evaluate results, and then more files could be processed according to a given study to evaluate if the tool is likely or not to handle all the data.

Preprocessing MS Data

After the mass spectra acquisition, the first step in the informatics pipeline involves several preprocessing methods. Since current MS instruments acquire high resolution data in profile mode, centroid data is computationally generated either during or post-acquisition (Fig. 2, conversion and pre-processing). Preprocessing methods include mass calibration, noise removal, peak picking and grouping of peaks into isotopic profiles with deconvolution and charge-state determination. Preprocessing allows significant data reduction and enhancement for further analysis. Format of raw MS files depends on the instrument vendor, as each of the major vendors uses its own proprietary binary format and continuously update it to support new features of their instruments. Therefore, open formats such as mzXML (Pedrioli *et al.*, 2004) and mzML (Martens *et al.*, 2011) have been created by the proteomics community and developers of analysis software to enable the exchange of data between tools. An overview of the most common file formats used in MS-based proteomics can be found in Deutsch (2012).

Raw MS data processing can be performed using proprietary software provided by the instrument vendors, which may produce better results because methods are customized to pair the characteristics of the data generated by the specific instrument. Alternatively, raw MS data can be parsed from the vendor format to open file formats – with or without preprocessing – to continue the processing pipeline using other software tools. The freely available ProteoWizard toolkit (Chambers *et al.*, 2012) provides software for converting proprietary files from most vendors to open file formats and includes several preprocessing algorithms both open source from the community and non-open source from the vendors. Other software platforms including preprocessing tools are OpenMS (Röst *et al.*, 2016; Sturm *et al.*, 2008), MaxQuant (Cox and Mann, 2008; Tyanova *et al.*, 2016) and MzJava (Horlacher *et al.*, 2015b).

A special case is the Skyline software tool for quantification (MacLean *et al.*, 2010; Pino *et al.*, 2017), where MS data from many instrument vendor formats can be imported directly, i.e., a conversion to open file formats is usually not required.

Identification by Tandem Mass Spectrometry

The fragmentation pattern encoded by a tandem or MS/MS spectrum yield identification of the corresponding peptide sequence. The so-called search engines (Fig. 2, identification) annotate the MS/MS spectra with peptide sequences using various algorithms and scoring strategies (Nesvizhskii, 2010). Peptide identification strategies, discussed in detail in (Lisacek), can be roughly classified into four categories: database search, spectral library search, de-novo sequencing and hybrid approaches such as sequence tag-assisted database search.

SEQUEST (Eng *et al.*, 1994) and MASCOT (Perkins *et al.*, 1999) proprietary database search engines have been traditionally and predominantly used. Nowadays, open source tools are becoming more popular, with increased robustness, usability and flexibility for customization or addition of new features. For instance, Comet is an academic version of SEQUEST. Table 1 summarizes examples of free and/or open source peptide identification software currently available.

As Table 1 shows, most of the identification tools do not have a GUI. Andromeda is available as standalone or integrated into MaxQuant. The SearchGUI (Vaudel *et al.*, 2011) (see Relevant Websites Section) provides an open source, user friendly framework to operate multiple command line search engines.

In terms of scoring, several properties can be taken into account to measure the similarity between an experimental spectrum and a reference. The precursor mass is mainly used to restrict the search space. However, the mass differences for both the matching fragments and precursor can be considered into the scoring. Other properties include the number and intensities of the experimental peaks, number of complementary ions, number of missed cleavages and number (and types) of modifications.

A great variety of scoring strategies have been proposed for peptide identification. Broadly speaking, they can be divided into heuristic and probabilistic schemes. Heuristic methods typically combine a set of scoring properties in a practical form reflecting expert knowledge. A product or a linear function are simple examples of how these properties can be combined into a scoring function. Probabilistic schemes compute the probability that a given sequence is the origin of the spectrum.

With the introduction of machine learning, algorithms that can automatically learn from training data have been applied to many fields, including proteomics. Specifically, MSGF+ is able to automatically derive scoring parameters from annotated spectra. More recently, DeepNovo implements a deep neural network model for de novo peptide sequencing.

Finally, the actual software implementation of the algorithm makes a great difference in terms of computing resources required by the tool. By using a fragment-ion indexing method, MSFragger enables a significant improvement in speed over most existing proteome database search tools.

Post-Translational Modifications

Post-translational modifications (PTMs) are alteration of proteins during or after protein biosynthesis, resulting in changes of the protein physicochemical properties which modulate and regulate cellular functions. Abnormal PTM abundances are involved in disease states.

Table 1 Selected list of peptide identification software freely available

Software	Programming language	Open source	GUI	Identification type	Scoring and modelling strategies	Reference and website
Comet	C++	Yes	No	Sequence DB	Heuristic. Cross-correlation	(Eng <i>et al.</i> , 2015, 2013) http://comet-ms.sourceforge.net
MSGF+	Java	Yes	No	Sequence DB	Probabilistic. Directed acyclic graph, dot product, generating function	(Kim and Pevzner, 2014) https://omics.pnl.gov
X!Tandem	C++	Yes	No ^a	Sequence DB	Heuristic. Hyperscore (hypergeometric distribution-based)	(Craig and Beavis, 2004) http://www.thegpm.org
Andromeda	C#	No	Yes	Sequence DB	Probabilistic. Binominal distribution-based	(Cox <i>et al.</i> , 2011) http://www.maxquant.org
MSFragger	Java	Yes	Yes	Sequence DB	Heuristic. Hyperscore (similar to X!Tandem)	(Kong <i>et al.</i> , 2017) http://www.nesvilab.org
SpectraST	C++	Yes	No ^a	Spectral library	Heuristic. Dot product	(Lam <i>et al.</i> , 2007) http://tools.proteomecenter.org
BiblioSpec	C++	Yes	No	Spectral library	Heuristic. Dot product	(Frewen <i>et al.</i> , 2006) https://skyline.ms
PepNovo+	C++	Yes	No	De-novo	Probabilistic. Directed acyclic graph	(Frank <i>et al.</i> , 2007) http://proteomics.ucsd.edu
DeepNovo	Python	Yes	No	De-novo	Probabilistic. Deep learning neural network	(Tran <i>et al.</i> , 2017) https://github.com/nh2tran/DeepNovo

^aAn alternative version is available through TPP GUI.

By identifying the occurrence of regular mass shifts corresponding to the addition or removal of chemical groups of known masses in a peptide MS/MS spectrum, it is possible to identify a PTM and its position on the peptide sequence. When specified in the search parameters, search engines can consider relevant PTMs, such as phosphorylation. However, this number is usually limited because the search space is substantially expanded. Consequently, an important fraction of MS/MS spectra usually remains unidentified, mainly due to the diversity of PTMs and other chemical modifications that are unaccounted. An attractive exception is the automatic PTMs detection algorithm in the proprietary search engine ProteinPilot (Shilov *et al.*, 2007), where modifications, substitutions, and cleavage events are modeled with probabilities rather than by discrete user-controlled settings. This alleviates the user with the tedious task of optimizing parameters.

A range of tools have been developed to perform the so-called “open modification search”, i.e., searching with no a priori PTM information (Ahrné *et al.*, 2010). Approaches aiming to automatically discover unexpected modifications in high throughput proteome data include: MODa (Na *et al.*, 2012), MzMod (Horlacher *et al.*, 2015a) and MSFragger (Kong *et al.*, 2017).

In addition, quantification of PTMs in large-scale studies using DIA has been reported with the tool called inference of peptidofoms (IPF), a fully automated algorithm that uses spectral libraries to query, validate and quantify peptidofoms in DIA datasets (Rosenberger *et al.*, 2017b).

Recent advances in instrumentation and development of dedicated software are making possible the characterization of proteomic samples by limiting or circumventing the enzymatic protein digestion. These approaches, referred to as middle-down and top-down, provide more complete sequence data, as oppose to bottom-up proteomics. Middle-down proteomics are hybrid methods, employing limited digestion to create large peptides which retain a richer sequence and PTM information and can be analyzed relatively easily by MS (Sidoli *et al.*, 2014). In top-down proteomics, high resolution MS/MS data is obtained directly from fragmentation of the intact protein (Smith and Kelleher, 2013; Zhang and Ge, 2011). Various software tools are available to analyze this type of complex spectra, adapted to the identification of proteofoms and PTMs, such as MS-Align + (Liu *et al.*, 2012), ProSight Lite (Fellers *et al.*, 2015), TopPIC (Kou *et al.*, 2016), MASH Suite (Cai *et al.*, 2016) and Informed-Proteomics (Park *et al.*, 2017).

Assessment of Error Rates in Identification Results and Protein Inference

Since only a fraction of all annotations generated by search engines are correct, proper assessment and control of error rates in identification results is a critical step in the informatics pipeline to distinguish correct from incorrect identifications. In proteomics, the most commonly used and accepted statistical confidence measure is the false discovery rate (FDR) (Benjamini and Hochberg, 1995) initially adapted by Elias and Gygi (2007) and also known as the “target-decoy approach” (TDA). In this strategy, a decoy database is first generated by reversing or shuffling the amino acids in the sequences of the reference or target database. All MS/MS spectra are searched against a composite target plus decoy protein sequence database, the best peptide match for each spectrum is selected for further analysis and the number of decoy peptide matches are used to estimate the fraction of incorrect annotations at given thresholds (Fig. 2, assessment and control of error rates).

The FDR analysis can be performed by the identification software or externally using stand-alone tools, represented in Table 2.

Table 2 Selected list of open source tools freely available for error rate assessment in identification results and protein inference

Software	Programming language	Features	Base modelling	Reference and website
PeptideProphet ^a	C++	Peptide FDR analysis	Target-decoy, mixture model, expectation-maximization	(Choi and Nesvizhskii, 2007; Keller <i>et al.</i> , 2002) http://tools.proteomecenter.org
ProteinProphet ^a	C++	Protein inference and FDR analysis	Mixture model, expectation-maximization	(Nesvizhskii <i>et al.</i> , 2003) http://tools.proteomecenter.org
iProphet ^a	C++	Peptide FDR analysis. Combination of results from multiple search engines	Target-decoy, mixture model, expectation-maximization	(Shteynberg <i>et al.</i> , 2011) http://tools.proteomecenter.org
Percolator	C++	Peptide and protein FDR analysis. Protein inference	Target-decoy, support vector machine	(Käll <i>et al.</i> , 2007; The <i>et al.</i> , 2016) http://percolator.ms
MAYU	Perl	Peptide and protein FDR analysis. Protein inference	Target-decoy, hypergeometric model	(Reiter <i>et al.</i> , 2009) http://proteomics.ethz.ch
PeptideShaker	Java	Peptide and protein FDR analysis. Combination of results from multiple search engines.	Target-decoy	(Vaudel <i>et al.</i> , 2015) http://compomics.github.io/projects/peptide-shaker.html
mProphet	Perl and R	Protein inference Peptide FDR for SRM, adapted later for DIA	Target-decoy, linear discriminant analysis	(Reiter <i>et al.</i> , 2011) http://www.mprophet.org/

^aAvailable through TPP.

The vast majority of the peptide FDR analysis tools utilize the target-decoy strategy. In the case of PeptideProphet, the initial parametric and unsupervised mixture model approach for posterior probability calculation was later combined with the target-decoy strategy within a single semisupervised framework (Choi and Nesvizhskii, 2007; Keller *et al.*, 2002).

Tools such as Percolator, MAYU and PepideShaker, also include the protein inference step to roll up identified peptide sequences into the corresponding proteins, and then calculate a statistical confidence score for each protein identification.

Inferring protein identities given a set of identified peptides is a difficult task due to several factors such as the presence of distinct proteins having a high degree of sequence homology and alternative splice forms of the same gene (Nesvizhskii and Aebersold, 2005).

It has been reported that the combination of several search engines further improves the results, by taking advantage of the specific strengths of each search engine to generate maximal results (Audain *et al.*, 2017; Kwon *et al.*, 2011; Shteynberg *et al.*, 2013). Tools like iProphet and PeptideShaker are of special considerations in the FDR analysis and necessary to properly combine multiple results while avoiding the accumulation of false positives from the different search engines.

Moreover, the FDR is a summary statistics for the entire identification results either at the PSM, peptide or protein level, however it does not indicate the identification confidence for a single hit at each level. There is a distinction between a global and local FDR. The global FDR is the fraction of incorrect hits among all hits that pass the threshold. The local FDR is the fraction of incorrect hits within a subset of hits that share the same score. In the data interpretation guidelines recently published by the Human proteome project (Deutsch *et al.*, 2016), detail reports of FDR values at each of the three levels are required for publication. The PSM-level FDR should be lower than the peptide-level FDR, which should be lower than the protein-level FDR. The larger the dataset, the more extreme these differences become. Extraordinary and novel detection claims require the evidence of two distinct peptides with length of nine amino acids or more.

Despite the FDR is a well-accepted concept, it has been reported that more advance statistical methods are needed in proteomics (Serang and Käll, 2015). An alternative to the target-decoy database approach was proposed and implemented in MSGF+, which uses E-values to evaluate statistical significance of individual PSMs through the generating function approach (Kim and Pevzner, 2014; Kim *et al.*, 2008).

The mProphet algorithms was originally implemented to automate data processing and statistical validation of large-scale selected reaction monitoring (SRM) experiments by scoring decoy transitions. Due to the similarity of the validation procedures, it has been extended and re-implemented in several tools for DIA.

Software to generate decoy spectral libraries are discussed in (Lisacek).

Quantification

Bioinformatics has played an essential role on the large-scale characterization of complex proteomes, traditionally by cataloguing proteins, and more recently, towards quantifying their expression profiles at different cellular states.

Due to differences in ionization efficiency, the signal detected by MS is not directly proportional to the protein concentration. Several combinations of analytical and computational methods have been developed and applied to quantify proteins by MS. Examples of studies evaluating quantification strategies include (Ahrné *et al.*, 2013; Blein-Nicolas and Zivy, 2016; Dowle *et al.*, 2016; Nahnsen *et al.*, 2013; Neilson *et al.*, 2011). **Table 3** shows a list of selected quantification software currently available.

Quantification methods can be label-free based or include stable isotope labels. Labeling methods are discussed in (Lisacek).

In label-free methods different samples are analyzed by LC-MS individually (**Fig. 2**, quantification), requiring reproducible chromatography and advanced informatics algorithms for retention time alignment across multiple samples and intensity normalization to minimize systematic biases. The so-called spectral counting technique has been conventionally used as a rapid and semi-quantitative measure of protein abundance based on the number of MS/MS spectra identified per peptide (Liu *et al.*, 2004). However, biological samples usually comprise a great variety of proteins across a wide dynamic range of abundances, where spectral counting has been found to often give irreproducible results, more notably for low abundance proteins (Cappadona *et al.*, 2012; Ahrné *et al.*, 2013; Blein-Nicolas and Zivy, 2016). In contrast, label-free methods based on the area under the peak or ion chromatogram provide a level of accuracy comparable to labeling approaches (Neilson *et al.*, 2011).

Multiple software packages are available to analyze the integrated peak areas and determine candidate proteins differentially expressed (**Fig. 2**, differential expression analysis). For instance, the R statistical package MSstats (Choi *et al.*, 2014) is a popular choice for analyzing DDA, DIA and SRM quantification results.

Data-Dependent Acquisition and MS1 Quantification

In shotgun proteomics, confidently identified peptides can be quantified using the continuous precursor ion signals from MS1 data or survey scans. MS1 quantification can be performed in two ways, differing in the approach used for chromatogram processing. One strategy, referred here as LC-MS feature-based, consists in first detecting and characterizing every LC-MS peak (often called LC-MS feature) in the MS1 data and then annotating them using the peptide identifications from the MS2 data. Among the first open source proteomics software implementing this approach are SuperHirn (Mueller *et al.*, 2007) and VIPER (Monroe *et al.*, 2007), and more recently MzMine 2 (Pluskal *et al.*, 2010). In contrast to attempt finding every MS1 peak, the second strategy is guided or targeted, i.e., extracted ion chromatograms (XIC or EIC) are constructed given a list of

Table 3 Selected software frameworks freely available for protein quantification

Software	Programming language	Open source	Identification	Chromatogram processing	Labelling	DIA	Reference and website
OpenMS	C++, Python	Yes	X!Tandem, MSGF+, OMSSA, MyriMatch, PepNovo	LC-MS feature-based, XIC-based	SILAC, iTRAC, TMT	Yes ^a	(Röst <i>et al.</i> , 2016; Sturm <i>et al.</i> , 2008; Pfeuffer <i>et al.</i> , 2017) http://www.openms.de/
TPP	Perl, C++, C, Java	Yes	X!Tandem, SpectraST, Comet	–	ICAT, SILAC, iTRAC, TMT	No	(Deutsch <i>et al.</i> , 2010, 2015) http://tools.proteomecenter.org
MaxQuant	C#	No	Andromeda	LC-MS feature-based	ICAT, SILAC, iTRAC, TMT	No	(Cox and Mann, 2008; Tyanova <i>et al.</i> , 2016) http://www.maxquant.org
Skyline	C#	Yes	BiblioSpec. Results import from a variety of search engines	XIC-based Advanced processing for SRM and PRM	Customizable heavy labeled stable-isotope modifications	Yes	(MacLean <i>et al.</i> , 2010; Pino <i>et al.</i> , 2017) https://skyline.ms

^aOpenSWATH.

mass-to-charge targets. Referred here as XIC-based, information from confidently identified peptides are used to extract the precursor ion signals of each peptide from the MS1 data. This is the case of Skyline, which compute the area under the peak from the XIC corresponding to each targeted peptide.

Data-Independent Acquisition and MS2 Quantification

DDA-based methods have been widely used to identify and quantify proteins. However, the semi-stochastic sampling in DDA leads to irreproducible results, particularly for highly complex proteomic samples. With recent developments in MS instrumentation and data analysis software, it has become practical the systematic and parallel fragmentation of all ions from peptides eluting during LC-MS analysis by DIA. These methods avoid the selection of individual peptide ions (Bilbao *et al.*, 2015a; Chapman *et al.*, 2014; Hu *et al.*, 2016). The label-free quantification performance of DIA methods has seen a growing popularity and adoption, providing a valid alternative to isotope-labeling-based methods (Navarro *et al.*, 2016).

DIA requires more sophisticated post-acquisition data analysis compared to DDA and several informatics tools have been developed to effectively process these complex datasets. The initial SWATH strategy (Gillet *et al.*, 2012) describing a targeted data extraction method using information from spectral libraries has been automated and improved in several tools: Skyline, DIANA (Teleman *et al.*, 2014), OpenSWATH (Röst *et al.*, 2014), SWATHProphet (Keller *et al.*, 2015). More recently, alternative workflows that circumvent the need of DDA for creating a spectral library have been developed: DIA-Umpire (Tsou *et al.*, 2015, 2016), GroupDIA (Li *et al.*, 2015), MSPLIT-DIA (Wang *et al.*, 2015) and Pecan (Ting *et al.*, 2017). A discussion about the challenges associated with error-rate control in the analysis of DIA data has been recently reported (Rosenberger *et al.*, 2017a).

Finally, a drawback of DIA is the increased likelihood of interference due to the overlap of fragment ions from co-fragmented precursors and several computational strategies can tackle this issue to further expand the benefits of DIA (Bilbao *et al.*, 2015b, 2016; Zhang *et al.*, 2015).

Selected Reaction Monitoring

Also referred to as multiple reaction monitoring (MRM), SRM is the MS method of choice to accurately and consistently quantify proteins across hundreds of complex samples. However, it is applicable when only a limited number of predefined proteins are compared. Therefore, it is typically used as the validation method, to confirm a list of target proteins obtained in a previous global or discovery study. For SRM data acquisition, the two mass filters of a triple quadrupole MS instrument select and monitor, over chromatographic elution, the signal of a series of transitions (precursor/fragment ion pairs and expected times) in a predefined list of target peptides (Lange *et al.*, 2008; Picotti and Aebersold, 2012).

Several informatics tools are available to assist the time-consuming process of defining the acquisition method. Examples of software to select proteins, proteotypic peptides, transitions and best acquisition settings include SRMcolider (Röst *et al.*, 2012), MRMOptimizer (Alghanem *et al.*, 2017), PREGO (Searle *et al.*, 2015) and the PNNL Biodiversity Plugin as one of the external tools available in Skyline (Degan *et al.*, 2016).

Since SRM data consist of a set of chromatographic traces for each monitored peptide, data analysis is simpler, compared to shotgun methods. The Skyline software tool is widely used for the integration of the chromatographic peaks for each transition, allowing the relative or, if heavy isotope-labeled reference standards are used, absolute quantification of the targeted peptides, which are used as a surrogate measure of the proteins of interest.

Alternatively, targeted MS methods such as parallel reaction monitoring (PRM) (Peterson *et al.*, 2012), use hybrid instruments where the third quadrupole is substituted with a high resolution and accurate mass analyzer, acquiring high-resolution full MS/MS spectra for each target peptide. Consequently, in PRM, the best fragment ions can be computationally selected post-acquisition. A web-based application for assay development was recently reported for targeted and label-free phosphoproteome analysis by PRM (Lawrence *et al.*, 2016).

Discussions regarding the best practices for targeted MS measurements in biology and medicine, including the computational and statistical tools, can be found in Carr *et al.* (2014).

How it all Fits Together

MS-based proteomics studies are supported by publicly accessible bioinformatics tools for processing and interpreting the large amounts of data that are generated in complex projects (Aebersold and Mann, 2016). To illustrate how various software tools are connected into processing pipelines, two recent representative applications are described.

The first study was based on MS1 quantification. Rieckmann and colleagues characterized 28 primary human hematopoietic cell populations in either steady or activated state (Rieckmann *et al.*, 2017). Measurements from three to four donors generated a total of 175 immune cell proteomes. Samples were analyzed by LC-MS with DDA and selection of the top-10 most intense precursor ions for fragmentation. Raw MS files were analyzed with MaxQuant. Identification was performed by the Andromeda search engine. The Label Free Quantification algorithm (MaxLFQ) was applied. Data analysis and visualization was performed using Perseus and R. More than 10,000 proteins were identified in total.

The second study was based on the SWATH strategy. Williams *et al.* (2016) analyzed phenotypes associated with liver mitochondrial metabolism. The authors profiled 386 individuals from 80 cohorts of the BXD mouse genetic reference population across two environmental states. Measurements in the livers of the entire population involved: transcriptome, proteome and metabolome. In particular for proteomics, a spectral library was built from a pooled sample analyzed by LC-MS with DDA and selection of the top-20 most intense precursor ions for fragmentation. Raw files were converted to mzML using AB Sciex Data Converter, applying the algorithm for centroid mode. Files were processed through TPP for identification. Search results were combined using iProphet and consensus spectra for a total of 22,208 peptides were constructed using SpectraST. Retention time was transformed to indexed retention time (iRT) (Escher *et al.*, 2012). The top-5 most abundant *b* and *y* fragment ions of each peptide were selected to generate the assays for the targeted data extraction. The DIA LC-MS analysis with SWATH was performed using a set of 32 overlapping isolation windows (width of 26 Da), covering the 400 to 1200 *m/z* precursor range. Raw files were converted to profile mzXML files using ProteoWizard. Targeted data extraction was performed using the OpenSWATH workflow. Identified peptides were analyzed with ProteinProphet. Peptide quantities and network graphs were calculated in R using the custom package imsbInfer (see Relevant Websites Section). A total of 2622 proteins were quantified in all 80 cohorts.

As it can be perceived, the processing and interpretation of hundreds of MS files and more than a million of spectra generated from these projects would have been impossible without bioinformatics tools properly assembled into workflows.

Related to network biology for MS-based proteomics (Fig. 2, protein network analysis), more information can be found in Bensimon *et al.* (2012). Examples of other available software include STRING (see Relevant Websites Section) as an online resource for querying and visualizing interaction networks and Cytoscape (Saito *et al.*, 2012; Shannon *et al.*, 2003) as one of the most popular open-source software for integration, visualization and analysis.

Closing Remarks

MS-based proteomics methodologies have evolved rapidly over recent years along with a remarkable progress on specialized software. These approaches are nowadays used in combination with other omics disciplines such as genomics, transcriptomics and metabolomics (Nesvizhskii, 2014; White *et al.*, 2017a,b). With the increasingly high throughput large scale studies in the big data era, robust and efficient MS software is as important as advanced MS hardware for comprehensively exploiting the information and generating knowledge. In this sense, progress in instrumentation and analytical methods are expected, and importantly, development of new computational methods to cope with these advances will continue to be an active area of research. Therefore, an increasing number of available tools including proprietary and open source can be expected.

The data processing workflow in MS-based proteomics is a set of consecutive complex tasks. It has been previously stated that this stepwise approach should be replaced by a joint model in which all relevant aspects of the experiment are taken into account (Noble and MacCoss, 2012). However, it is not possible tackling the complexity and variety of applications with a single joint model. Software modularity is thus required, yet, with high transparency and automation to support efficient integration of all available information. Open source software plays a key role towards this goal. Software modules with well-defined interfaces and input/output formats facilitate integration at different levels, avoiding information loss at every processing step and opens the

potential to integrate state-of-the-art machine learning to model dependencies among variables at the spectral, peptide, and protein level that are currently decoupled, as well as other relevant information such as clinical or environmental data. This is enhancing our understanding of the state of the proteome and its response to perturbations.

The diversity of questions possibly addressed with MS and the evolving technologies require software customization and/or extension. Robust open source software accelerates adaptations for technological innovations. For instance, the additional ion mobility (IM) dimension currently available in commercial instruments provides reproducible separation and structural information with numerous benefits in proteomic analyses (Baker *et al.*, 2015). However, this brings challenges for data processing because existing software tools for LC-MS data do not work. Skyline was updated to support LC-IM-MS and optimization of IM informatics is in progress for analysis of these large and multi-dimensional datasets (Pino *et al.*, 2017).

In this regard, several useful comments can be summarized about scientific software development. First, bioinformaticians are strongly encouraged to embrace practices that ensure software quality, sustainability and reproducible research (Jiménez *et al.*, 2017; Perez-Riverol *et al.*, 2016). Development of new algorithms and data analysis tools is challenging, it is a common practice developing prototypes first, with at least some basic design and testing. However, as soon as a software tool proves its usefulness, and before it is “too late” because it is too extensive and complex, time and effort must be dedicated for more testing, re-factoring and documentation. And this implies financial support.

Similarly, bioinformaticians are encouraged to join community efforts in order to avoid reinventing the wheel. Improving upon something is typically much easier than starting from scratch and allows for incremental progress, as one can see further by standing on the shoulders of giants. Crucially, any previous related work should be properly cited and acknowledged to avoid the term “research parasites”. This term has been recently and unjustifiably used for people reusing data (Martens and Vizcaíno, 2017), and should be also avoided for people reusing code.

Second, a software tool should have both command line functionality and GUI, either as stand-alone tool or integrated into current pipelines. Command line interfaces facilitate integration with other tools and are necessary for automation in large scale studies. On the other hand, a GUI expedites a quick execution in a few clicks, potentially increasing dissemination and number of users. In addition, the GUI facilitates testing, quality control, optimization of parameters and re-execution from any step to the end. Related to this point, visualization, at least as static figures in output files, is critical to support quick evaluation of results.

Third, a person other than the developer should try to follow the documentation (e.g., user guide, tutorial) and test the tool. The provided feedback will increase the robustness and usability. For instance, in a recent study of benchmarking DIA label-free quantification tools (Navarro *et al.*, 2016), feedback enabled developers to improve their software tools and after optimization, all tools provided highly convergent identification and reliable quantification performance. Documentation should include installation steps, software dependencies with versions, file formats and test data. Testing should be performed also with real data. For example, a simple approach producing acceptable results for a simple protein mixture, may produce poor results when processing a complex one such as a cell lysate.

Fourth, to ensure reproducibility, scientific articles related to both benchmarking and application studies should properly document the usage of any software, including details about the parameters and software version used, and data description with experimental and technical metadata related with the type of experiment. The metadata requirements for proteomics are in general much less comprehensive than those from more mature omics fields (Martens and Vizcaíno, 2017).

Finally, multidisciplinary collaborations will facilitate the design, implementation and validation of more advanced computational methods for proteomics and integration with other omics. Development of more powerful and user-friendly MS tools that can be used by researchers non-specialized in mass spectrometry or informatics is encouraged.

Acknowledgements

The author would like to acknowledge the following researchers for various insightful discussions: Frédérique Lisacek (software tools, text editions and article organization), Samuel H. Payne (scoring functions and MSGF+), John Fjeldsted (software workflows and automation) and Richard Allen White III (open source software).

See also: Clinical Proteomics. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Identification of Proteins from Proteomic Analysis. Natural Language Processing Approaches in Bioinformatics. Protein Post-Translational Modification Prediction. Protein Properties. Proteomics Data Representation and Databases. Quantification of Proteins from Proteomic Analysis. Transmembrane Domain Prediction

References

- Aebersold, R., 2003. Mass spectrometry-based proteomics. *Nature* 422, 198–207.
- Aebersold, R., 2009. A stress test for mass spectrometry-based proteomics. *Nature Methods* 6 (6), 411–412.
- Aebersold, R., Mann, M., 2016. Mass-spectrometric exploration of proteome structure and function. *Nature* 537 (7620), 347–355.

- Aïgan, E., Baker, D., Van den Beek, M., *et al.*, 2016. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research* 44 (W1), W3–W10.
- Ahrné, E., Molzahn, L., Glatter, T., Schmidt, A., 2013. Critical assessment of proteome-wide label-free absolute abundance estimation strategies. *Proteomics* 13 (17), 2567–2578.
- Ahrné, E., Müller, M., Lisacek, F., 2010. Unrestricted identification of modified proteins using MS/MS. *Proteomics* 10 (4), 671–686.
- Aiche, S., Sachsenberg, T., Kenar, E., *et al.*, 2015. Workflows for automated downstream data analysis and visualization in large-scale computational mass spectrometry. *Proteomics* 15 (8), 1443–1447.
- Alghanem, B., Nikitin, F., Stricker, T., *et al.*, 2017. Optimization by infusion of multiple reaction monitoring transitions for sensitive peptides LC-MS quantification. *Rapid Communications in Mass Spectrometry*.
- Audain, E., Uszkoreit, J., Sachsenberg, T., *et al.*, 2017. In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics. *Journal of Proteomics* 150 (Suppl. C), 170–182.
- Baker, E.S., Burnum-Johnson, K.E., Ibrahim, Y.M., *et al.*, 2015. Enhancing bottom-up and top-down proteomic measurements with ion mobility separations. *Proteomics* 15 (16), 2766–2776.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)* 57 (1), 289–300.
- Bensimon, A., Heck, A.J., Aebersold, R., 2012. Mass spectrometry-based proteomics and network biology. *Annual Review of Biochemistry* 81, 379–405.
- Berthold, M.R., Cebon, N., Dill, F., *et al.*, 2009. KNIME—the Konstanz information miner: Version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter* 11 (1), 26–31.
- Bilbao, A., Lisacek, F., Hopfgartner, G., 2016. Dedicated software enhancing data-independent acquisition methods in mass spectrometry. *Chimia International Journal for Chemistry* 70 (4), 293.
- Bilbao, A., Varesio, E., Luban, J., *et al.*, 2015a. Processing strategies and software solutions for data-independent acquisition in mass spectrometry. *Proteomics* 15 (5–6), 964–980.
- Bilbao, A., Zhang, Y., Varesio, E., *et al.*, 2015b. Ranking fragment ions based on outlier detection for improved label-free quantification in data-independent acquisition LC-MS/MS. *Journal of Proteome Research* 14, 4581–4593.
- Blein-Nicolas, M., Zivy, M., 2016. Thousand and one ways to quantify and compare protein abundances in label-free bottom-up proteomics. *Biochimica et Biophysica Acta (BBA) – Proteins and Proteomics* 1864 (8), 883–895.
- Cai, W., Guner, H., Gregorich, Z.R., *et al.*, 2016. MASH Suite Pro: A comprehensive software tool for top-down proteomics. *Molecular & Cellular Proteomics* 15 (2), 703–714.
- Cappadona, S., Baker, P.R., Cutillas, P.R., Heck, A.J., Breukelen, B.van., 2012. Current challenges in software solutions for mass spectrometry-based quantitative proteomics. *Amino Acids* 43 (3), 1087–1108.
- Carr, S.A., Abbatiello, S.E., Ackermann, B.L., *et al.*, 2014. Targeted peptide measurements in biology and medicine: Best practices for mass spectrometry-based assay development using a fit-for-purpose approach. *Molecular & Cellular Proteomics* 13 (3), 907–917.
- Chambers, M.C., Maclean, B., Burke, R., *et al.*, 2012. A cross-platform toolkit for mass spectrometry and proteomics. *Nature Biotechnology* 30 (10), 918–920.
- Chapman, J.D., Goodlett, D.R., Masselon, C.D., 2014. Multiplexed and data-independent tandem mass spectrometry for global proteome profiling. *Mass Spectrometry Reviews* 33, 452–470.
- Choi, H., Nesvizhskii, A.I., 2007. Semisupervised model-based validation of peptide identifications in mass spectrometry-based proteomics. *Journal of Proteome Research* 7 (01), 254–265.
- Choi, M., Chang, C.-Y., Clough, T., *et al.*, 2014. MSstats: An R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments. *Bioinformatics* 30 (17), 2524–2526.
- Cox, J., Mann, M., 2008. MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology* 26 (12), 1367–1372.
- Cox, J., Neuhauser, N., Michalski, A., *et al.*, 2011. Andromeda: A peptide search engine integrated into the MaxQuant environment. *Journal of Proteome Research* 10 (4), 1794–1805.
- Craig, R., Beavis, R.C., 2004. TANDEM: Matching proteins with tandem mass spectra. *Bioinformatics* 20 (9), 1466–1467.
- da Veiga Leprevost, F., Grüning, B.A., Alves Aflitos, S., *et al.*, 2017. BioContainers: An open-source and community-driven framework for software standardization. *Bioinformatics*. btx192.
- Degan, M.G., Ryadinskiy, L., Fujimoto, G.M., *et al.*, 2016. A skyline plugin for pathway-centric data browsing. *Journal of The American Society for Mass Spectrometry* 27 (11), 1752–1757.
- Deutsch, E.W., 2012. File formats commonly used in mass spectrometry proteomics. *Molecular & Cellular Proteomics* 11 (12), 1612–1621.
- Deutsch, E.W., Csordas, A., Sun, Z., *et al.*, 2017. The ProteomeXchange consortium in 2017: Supporting the cultural change in proteomics public data deposition. *Nucleic Acids Research* 45 (D1), D1100–D1106.
- Deutsch, E.W., Mendoza, L., Shteynberg, D., *et al.*, 2010. A guided tour of the trans-proteomic pipeline. *Proteomics* 10 (6), 1150–1159.
- Deutsch, E.W., Mendoza, L., Shteynberg, D., *et al.*, 2015. Trans-proteomic pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics. *Proteomics – Clinical Applications* 9 (7–8), 745–754.
- Deutsch, E.W., Overall, C.M., Van Eyk, J.E., *et al.*, 2016. Human proteome project mass spectrometry data interpretation guidelines 2.1. *Journal of Proteome Research* 15 (11), 3961–3970.
- Dowle, A.A., Wilson, J., Thomas, J.R., 2016. Comparing the diagnostic classification accuracy of iTRAQ, peak-area, spectral-counting, and emPAI methods for relative quantification in expression proteomics. *Journal of Proteome Research* 15 (10), 3550–3562.
- Elias, J.E., Gygi, S.P., 2007. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods* 4 (3), 207–214.
- Eng, J.K., Hoopmann, M.R., Jahan, T.A., *et al.*, 2015. A deeper look into Comet—implementation and features. *Journal of the American Society for Mass Spectrometry* 26 (11), 1865–1874.
- Eng, J.K., Jahan, T.A., Hoopmann, M.R., 2013. Comet: An open-source MS/MS sequence database search tool. *Proteomics* 13 (1), 22–24.
- Eng, J.K., McCormack, A.L., Yates, J.R., 1994. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry* 5 (11), 976–989.
- Escher, C., Reiter, L., MacLean, B., *et al.*, 2012. Using iRT, a normalized retention time for more targeted measurement of peptides. *Proteomics* 12 (8), 1111–1121.
- Fellers, R.T., Greer, J.B., Early, B.P., *et al.*, 2015. ProSight lite: Graphical software to analyze top-down mass spectrometry data. *Proteomics* 15 (7), 1235–1238.
- Frank, A.M., Savitski, M.M., Nielsen, M.L., Zubarev, R.A., Pevzner, P.A., 2007. De novo peptide sequencing and identification with precision mass spectrometry. *Journal of Proteome Research* 6 (1), 114–123.
- Frewen, B.E., Merrihew, G.E., Wu, C.C., Noble, W.S., MacCoss, M.J., 2006. Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries. *Analytical Chemistry* 78 (16), 5678–5684.
- Gillet, L.C., Navarro, P., Tate, S., *et al.*, 2012. Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis. *Molecular & Cellular Proteomics*. 11.
- Horlacher, O., Lisacek, F., Müller, M., 2015a. Mining large scale tandem mass spectrometry data for protein modifications using spectral libraries. *Journal of Proteome Research* 15 (3), 721–731.
- Horlacher, O., Nikitin, F., Alocci, D., *et al.*, 2015b. MzJava: An open source library for mass spectrometry data processing. *Journal of Proteomics* 129, 63–70.

- Hu, A., Noble, W.S., Wolf-Yadlin, A., 2016. Technical advances in proteomics: New developments in data-independent acquisition. *F1000Research* 5.
- Ison, J., Rapacki, K., Ménager, H., *et al.*, 2016. Tools and data services registry: A community effort to document bioinformatics resources. *Nucleic Acids Research* 44 (D1), D38–D47.
- Jiménez, R., Kuzak, M., Alhamdoosh, M., *et al.*, 2017. Four simple recommendations to encourage best practices in research software. *F1000Research* 6, 876.
- Käll, L., Canterbury, J.D., Weston, J., Noble, W.S., MacCoss, M.J., 2007. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods* 4 (11), 923–925.
- Keller, A., Bader, S.L., Shteynberg, D., Hood, L., Moritz, R.L., 2015. Automated validation of results and removal of fragment ion interferences in targeted analysis of data independent acquisition MS using SWATHProphet. *Molecular & Cellular Proteomics*. (mcp-0114).
- Keller, A., Nesvizhskii, A.I., Kolker, E., Aebersold, R., 2002. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical Chemistry* 74 (20), 5383–5392.
- Kim, S., Gupta, N., Pevzner, P.A., 2008. Spectral probabilities and generating functions of tandem mass spectra: A strike against decoy databases. *Journal of Proteome Research* 7 (8), 3354–3363.
- Kim, S., Pevzner, P.A., 2014. MS-GFmathplus makes progress towards a universal database search tool for proteomics. *Nature Communications* 5, 5277.
- Kong, A.T., Leprevost, F.V., Avtonomov, D.M., Mellacheruvu, D., Nesvizhskii, A.I., 2017. MSFragger: Ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nature Methods* 14 (5), 513–520.
- Kou, Q., Xun, L., Liu, X., 2016. TopPIC: A software tool for top-down mass spectrometry-based proteoform identification and characterization. *Bioinformatics* 32 (22), 3495–3497.
- Kwon, T., Choi, H., Vogel, C., Nesvizhskii, A.I., Marcotte, E.M., 2011. MSblender: A probabilistic approach for integrating peptide identifications from multiple database search engines. *Journal of Proteome Research* 10 (7), 2949–2958.
- Lam, H., Deutsch, E.W., Edes, J.S., *et al.*, 2007. Development and validation of a spectral library searching method for peptide identification from MS/MS. *Proteomics* 7 (5), 655–667.
- Lange, V., Picotti, P., Domon, B., Aebersold, R., 2008. Selected reaction monitoring for quantitative proteomics: A tutorial. *Molecular Systems Biology* 4 (1), 1–14.
- Lawrence, R.T., Searle, B.C., Llovet, A., Villén, J., 2016. Plug-and-play analysis of the human phosphoproteome by targeted high-resolution mass spectrometry. *Nature Methods*.
- Liu, H., Sadygov, R.G., Yates III, J.R., 2004. A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Analytical Chemistry* 76 (14), 4193–4201.
- Liu, X., Sirotkin, Y., Shen, Y., *et al.*, 2012. Protein identification using top-down spectra. *Molecular & Cellular Proteomics* 11 (6), 111–8524.
- Li, Y., Zhong, C.-Q., Xu, X., *et al.*, 2015. Group-DIA: Analyzing multiple data-independent acquisition mass spectrometry data files. *Nature Methods*.
- MacLean, B., Tomazela, D.M., Shulman, N., *et al.*, 2010. Skyline: An open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* 26 (7), 966–968.
- Martens, L., Chambers, M., Sturm, M., *et al.*, 2011. mzML – A community standard for mass spectrometry data. *Molecular & Cellular Proteomics* 10 (1), 110–133.
- Martens, L., Vizcaino, J.A., 2017. A golden age for working with public proteomics data. *Trends in Biochemical Sciences*.
- Monroe, M.E., Tolić, N., Jaitly, N., *et al.*, 2007. VIPER: An advanced software package to support high-throughput LC-MS peptide identification. *Bioinformatics* 23 (15), 2021.
- Mueller, L.N., Rinner, O., Schmidt, A., *et al.*, 2007. SuperHirn – A novel tool for high resolution LC-MS-based peptide/protein profiling. *Proteomics* 7 (19), 3470–3480.
- Nahnsen, S., Bielow, C., Reinert, K., Kohlbacher, O., 2013. Tools for label-free peptide quantification. *Molecular & Cellular Proteomics* 12 (3), 549–556.
- Na, S., Bandeira, N., Paek, E., 2012. Fast multi-blind modification search through tandem mass spectrometry. *Molecular & Cellular Proteomics* 11 (4), 111–10199.
- Navarro, P., Kuharev, J., Gillet, L.C., *et al.*, 2016. A multicenter study benchmarks software tools for label-free proteome quantification. *Nature Biotechnology*.
- Neilson, K.A., Ali, N.A., Muralidharan, S., *et al.*, 2011. Less label, more free: Approaches in label-free quantitative mass spectrometry. *Proteomics* 11 (4), 535–553.
- Nesvizhskii, A.I., 2010. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics* 73 (11), 2092–2123.
- Nesvizhskii, A.I., 2014. Proteogenomics: Concepts, applications and computational strategies. *Nature Methods* 11 (11), 1114–1125.
- Nesvizhskii, A.I., Aebersold, R., 2005. Interpretation of shotgun proteomic data: The protein inference problem. *Molecular & Cellular Proteomics* 4 (10), 1419–1440.
- Nesvizhskii, A.I., Keller, A., Kolker, E., Aebersold, R., 2003. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry* 75 (17), 4646–4658.
- Noble, W.S., MacCoss, M.J., 2012. Computational and statistical analysis of protein mass spectrometry data. *PLOS Computational Biology* 8 (1), 1–6.
- Park, J., Piehowski, P.D., Wilkins, C., *et al.*, 2017. Informed-Proteomics: Open-source software package for top-down proteomics. *Nature Methods*.
- Pedrioli, P.G., Eng, J.K., Hubley, R., *et al.*, 2004. A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology* 22 (11), 1459–1466.
- Perez-Riverol, Y., Gatto, L., Wang, R., *et al.*, 2016. Ten simple rules for taking advantage of Git and GitHub. *PLOS Computational Biology: Public Library of Science* 12 (7), e1004947.
- Perkins, D.N., Pappin, D.J., Creasy, D.M., Cottrell, J.S., 1999. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 20 (18), 3551–3567.
- Peterson, A.C., Russell, J.D., Bailey, D.J., Westphall, M.S., Coon, J.J., 2012. Parallel reaction monitoring for high resolution and high mass accuracy quantitative, targeted proteomics. *Molecular & Cellular Proteomics* 11, 1475–1478.
- Pfeuffer, J., Sachsenberg, T., Alka, O., *et al.*, 2017. OpenMS – A platform for reproducible analysis of mass spectrometry data. *Journal of Biotechnology* 261 (Suppl. C), 142–148.
- Picotti, P., Aebersold, R., 2012. Selected reaction monitoring-based proteomics: Workflows, potential, pitfalls and future directions. *Nature Methods* 9 (6), 555–566.
- Pino, L.K., Searle, B.C., Bollinger, J.G., *et al.*, 2017. The skyline ecosystem: Informatics for quantitative mass spectrometry proteomics. *Mass Spectrometry Reviews*.
- Pluskal, T., Castillo, S., Villar-Briones, A., Orešić, M., 2010. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics* 11 (1), 395.
- Reiter, L., Claassen, M., Schrimpf, S.P., *et al.*, 2009. Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry. *Molecular & Cellular Proteomics* 8 (11), 2405–2417.
- Reiter, L., Rinner, O., Picotti, P., *et al.*, 2011. mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nature Methods* 8 (5), 430–435.
- Rieckmann, J.C., Geiger, R., Hornburg, D., *et al.*, 2017. Social network architecture of human immune cells unveiled by quantitative proteomics. *Nature Immunology* 18 (5), 583–593.
- Rosenberger, G., Bludau, I., Schmitt, U., *et al.*, 2017a. Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses. *Nature Methods* 14 (9), 921–927.
- Rosenberger, G., Liu, Y., Rost, H.L., *et al.*, 2017b. Inference and quantification of peptidoforms in large sample cohorts by SWATH-MS. *Nature Biotechnology* 35 (8), 781–788.
- Röst, H., Malmström, L., Aebersold, R., 2012. A computational tool to detect and avoid redundancy in selected reaction monitoring. *Molecular & Cellular Proteomics* 11 (8), 540–549.
- Röst, H.L., Rosenberger, G., Navarro, P., *et al.*, 2014. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nature Biotechnology* 32 (3), 219–223.
- Röst, H.L., Sachsenberg, T., Aiche, S., *et al.*, 2016. OpenMS: A flexible open-source software platform for mass spectrometry data analysis. *Nature Methods* 13 (9), 741–748.
- Saito, R., Smoot, M.E., Ono, K., *et al.*, 2012. A travel guide to cytoscape plugins. *Nature Methods* 9 (11), 1069–1076.

- Searle, B.C., Egertson, J.D., Bollinger, J.G., Stergachis, A.B., MacCoss, M.J., 2015. Using data independent acquisition (DIA) to model high-responding peptides for targeted proteomics experiments. *Molecular & Cellular Proteomics* 14 (9), 2331–2340.
- Serang, O., Käll, L., 2015. Solution to statistical challenges in proteomics is more statistics, not less. *Journal of Proteome Research*.
- Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research* 13 (11), 2498–2504.
- Shilov, I.V., Seymour, S.L., Patel, A.A., *et al.*, 2007. The paragon algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Molecular & Cellular Proteomics* 6, 1638.
- Shteynberg, D., Deutsch, E.W., Lam, H., *et al.*, 2011. iProphet: Multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates. *Molecular & Cellular Proteomics* 10 (12), 111–7690.
- Shteynberg, D., Nesvizhskii, A.I., Moritz, R.L., Deutsch, E.W., 2013. Combining results of multiple search engines in proteomics. *Molecular & Cellular Proteomics* 12 (9), 2383–2393.
- Sidoli, S., Schwämmle, V., Ruminowicz, C., *et al.*, 2014. Middle-down hybrid chromatography/tandem mass spectrometry workflow for characterization of combinatorial post-translational modifications in histones. *Proteomics* 14 (19), 2200–2211.
- Smith, L.M., Kelleher, N.L., 2013. Proteoform: A single term describing protein complexity. *Nature Methods* 10 (3), 186–187.
- Sturm, M., Bertsch, A., Gröpl, C., *et al.*, 2008. OpenMS – An open-source software framework for mass spectrometry. *BMC Bioinformatics* 9 (1), 163.
- Szabo, Z., Janaky, T., 2015. Challenges and developments in protein identification using mass spectrometry. *TrAC Trends in Analytical Chemistry*. 76–87.
- Teleman, J., Röst, H., Rosenberger, G., *et al.*, 2014. DIANA-algorithmic improvements for analysis of data-independent acquisition MS data. *Bioinformatics*. btu686.
- The, M., MacCoss, M.J., Noble, W.S., Käll, L., 2016. Fast and accurate protein false discovery rates on large-scale proteomics data sets with percolator 3.0. *Journal of the American Society for Mass Spectrometry* 27 (11), 1719–1727.
- Ting, Y.S., Egertson, J.D., Bollinger, J.G., *et al.*, 2017. PECAN: Library-free peptide detection for data-independent acquisition tandem mass spectrometry data. *Nature Methods* 14 (9), 903–908.
- Tran, N.H., Zhang, X., Xin, L., Shan, B., Li, M., 2017. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences* 114 (31), 8247–8252.
- Tsou, C.-C., Avtonomov, D., Larsen, B., *et al.*, 2015. DIA-Umpire: Comprehensive computational framework for data independent acquisition proteomics. *Nature Methods* 12 (3), 258–264.
- Tsou, C.-C., Tsai, C.-F., Teo, G., Chen, Y.-J., Nesvizhskii, A.I., 2016. Untargeted, spectral library-free analysis of data independent acquisition proteomics data generated using Orbitrap mass spectrometers. *Proteomics* 16, 2257–2271.
- Tyanova, S., Temu, T., Cox, J., 2016. The MaxQuant computational platform for mass spectrometry-based shotgun proteomics. *Nature Protocols* 11 (12), 2301–2319.
- Vaudel, M., Barsnes, H., Berven, F.S., Sickmann, A., Martens, L., 2011. SearchGUI: An open-source graphical user interface for simultaneous OMSSA and X! Tandem searches. *Proteomics* 11 (5), 996–999.
- Vaudel, M., Burkhardt, J.M., Zahedi, R.P., *et al.*, 2015. PeptideShaker enables reanalysis of MS-derived proteomics data sets. *Nature Biotechnology* 33 (1), 22–24.
- Wang, J., Tucholska, M., Knight, J.D., *et al.*, 2015. MSPLIT-DIA: Sensitive peptide identification for data-independent acquisition. *Nature Methods*.
- White, R.A., Borkum, M.I., Rivas-Ubach, A., *et al.*, 2017a. From data to knowledge: The future of multi-omics data analysis for the rhizosphere. *Rhizosphere* 3 (Part 2), 222–229.
- White, R.A., Rivas-Ubach, A., Borkum, M.I., *et al.*, 2017b. The state of rhizospheric science in the era of multi-omics: A practical guide to omics technologies. *Rhizosphere* 3 (Part 2), 212–221.
- Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., *et al.*, 2016. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data* 3, 160018.
- Williams, E.G., Wu, Y., Jha, P., *et al.*, 2016. Systems proteomics of liver mitochondria function. *Science* 352 (6291), aad0189.
- Yates III, J.R., 2015. Pivotal role of computers and software in mass spectrometry – SEQUEST and 20 years of tandem MS database searching. *Journal of The American Society for Mass Spectrometry* 26 (11), 1804–1813.
- Zhang, H., Ge, Y., 2011. Comprehensive analysis of protein modifications by top-down mass spectrometry. *Circulation: Cardiovascular Genetics* 4 (6), 711–724.
- Zhang, Y., Bilbao, A., Bruderer, T., *et al.*, 2015. The use of variable Q1 isolation windows improves selectivity in LC–SWATH–MS acquisition. *Journal of Proteome Research* 14, 4359–4371.

Relevant Websites

<http://biocontainers.pro/>
BioContainers.

<https://bio.tools/>
BioTools.

<http://compomics.github.io/projects/searchgui.html>
Compomics Docs.

<https://www.knime.com/community/bioinf/openms>
OpenMS Nodes for KNIME.

<https://galaxyproject.org/proteomics/use-cases>
Proteomics Use Cases.

<http://www.proteomexchange.org>
ProteomeXchange.

<http://string-db.org>
STRING.

<https://github.com/wolski/imsblnfer>
Wolski/imsblnfer.

Biological Databases

Sarinder K Dhillon, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Database is one of the most important innovation in computer science and information technology. Since its birth in 1960s, it has evolved many folds and is still evolving until today. Previously file based systems were used but were limited in its use due to issues such as data redundancy and inconsistency, difficulty in accessing data, data integrity and security, data isolation and many more. In order to resolve these concerns and limitations, the database approach was introduced and it was received well by the community. This is because database approach is more advanced, as the data is structured in a more formalised manner while the relationships between the data items are meaningful and consistent. Additionally, the database redundancy problem is very much minimised in databases compared to file based systems. In this way, the data can be used more efficiently either by automated or even manual systems. Database system applications are widely used in enterprises, banking and finance, universities, airlines, telecommunication and science due to the need for a computerised management of data which includes manipulation of data. Database systems promotes the use of advanced techniques such as data visualisation, data mining and data integration.

Currently in the era of data explosion, particularly in biology, the use and application of database technology is seen to be rising. Many areas such as bioinformatics, biodiversity and medical are applying the database technology to store, retrieve, manipulate and analyse data. The trend of big data has taken over conventional approaches in managing experimental data in biology. It is envisaged that a marriage between biology and database will soon churn out new trends, applications and discoveries.

This article presents the database fundamentals which includes different types of databases, the evolvement of biological databases, the approaches used in developing biological databases, some case studies using the approaches described, the results and discussion and finally the future direction of biological databases.

Database Fundamentals

Databases

In the simplest form, a database can be defined as an organised and formal collection of information stored in a computer readable format. In this section, some of the common approaches used in developing databases are presented.

Relational databases

The relational database has evolved over the years since its first introduction in 1970 and was largely adopted by industries across the world. Data in the relational model is usually represented by a structured schema ([Fig. 1](#)) in order to provide the data dictionary or metadata concerning the actual data ([Elmasri and Navathe, 2011](#)). Data which are represented by attributes are clustered based on the similarity, type and format in the form of columns of a table. The row on the other hand, represents the records in a database. A primary key is used to link data between multiple tables, whereas a foreign key is a reference to a primary key in another table. The data in a database is organised based on the relationships among the entities (and their corresponding attributes) while redundancy is minimized by performing data normalization.

Relational database comes with a default declarative language named Structured Query Language (SQL) developed at IBM by Donald D. Chamberlin and Raymond F. Boyce in the early 1970s. It provides interaction with structure data via CRUD (Create, Read, Update and Delete) statements and allows combination with multiple statement to perform more complex actions ([Cancelo, 2014](#)) ([Fig. 2](#)).

One of the main drawback of the relational model is that searching is burdened whereby the value being searched for is compared against every row in the table, from the first row to the last. This process is time consuming and has higher computational burden especially for large databases unless it uses other methods for searching such as indexes ([Segaran and Hammerbacher, 2009a](#)).

Multidimensional databases (OLAP)

In principle, a multidimensional database (MDD) is created using inputs from a relational database. In a MDD, data is stored in a cube, or a multidimensional array. When talking about MDDs, the term OLAP is prominent. The On-Line Analytical Processing (OLAP) is a term introduced by Dr. E.F. Codd ([Mallach, 2000](#)) which encompasses relational databases. The rule based OLAP system described by Dr. Codd are explained by [Pendse \(2008\)](#) formulating them into five simple definitions; Fast, Analysis, Shared, Multidimensional and Information. The OLAP is typically used in the business domain where it meets the needs of business analysts in business management, generating reports in addition to data mining. OLAP systems has also been used in

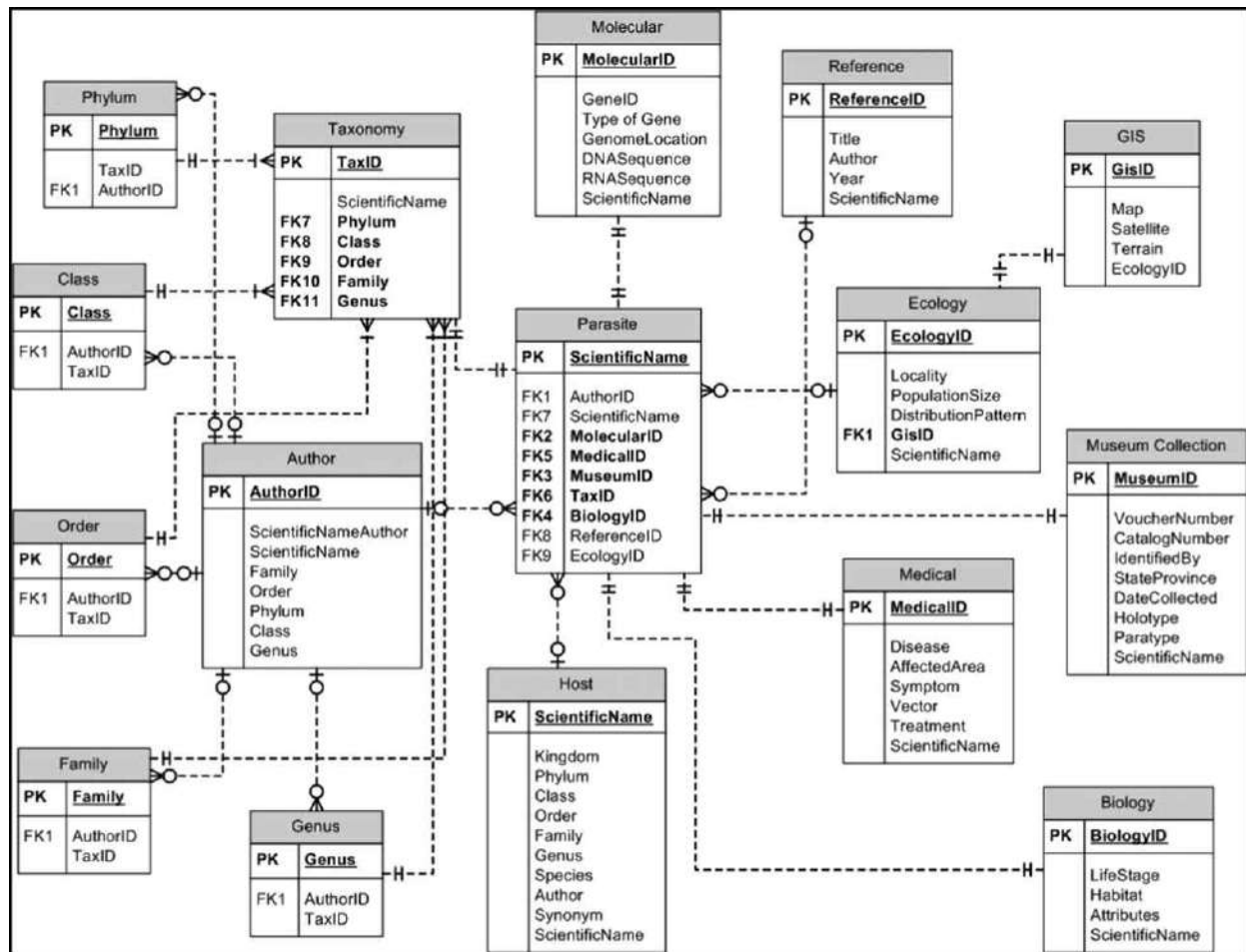


Fig. 1 An example of Entity Relationship Diagram using a relation schema.

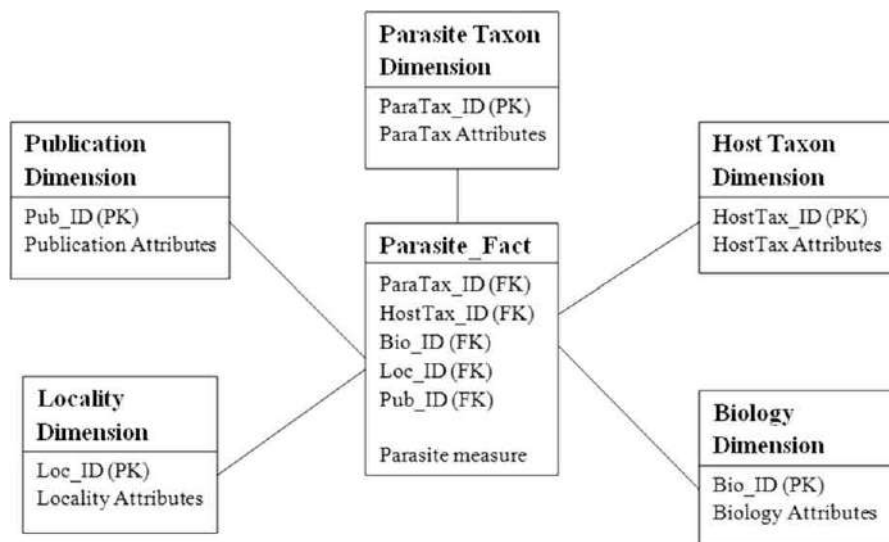


Fig. 2 Example of an OLAP Model where data is represented in a cube dimensional model.

Biology, but are vaguely reported. Information stored in a MDD database are often used for effective decision making that usually deals with complex relationships with a short response time.

Graph and ontology based databases (NoSQL databases)

Ontology based databases are in principle made of the object oriented approach, where the datasets are treated as objects, with values and properties, similar to objects represented by object oriented programming like C++ and Java. Ontologies can be defined as formal representation of the types, properties, and interrelationships of the entities on a specific domain and form the basic component of the semantic web architecture. These entities, being linked with other entities by the means of relationships, lay the underlying foundation of the semantic web (Guarino, 1995). The ontologies, when linked with other ontologies, generate a network of nodes and relationships, laying the foundations of linked data (Heath and Bizer, 2011).

The ontology development in a particular domain is facilitated by standardization of vocabulary that is accepted and used by domain experts. These vocabularies are shared for reuse, monitored by a body of experts to oversee the development of such vocabularies. An ontology can enhance information providing capability for a database by the use of metadata to discover and gather new information from other databases, or by linking them to create a better information network. An ontology can also be used as an underlying framework for building a database (Abu et al., 2013a,b; Ali et al., 2017).

NoSQL is a term used to refer to schemaless databases, which do not fit within the traditional relational paradigm. It describes a new category of non-relational databases, which are also known as distributed data stores that are capable of scaling to large datasets spanning to multiple data centres (Ercan, 2014). Recently, NoSQL database systems started to receive much attention as the demand increases for high speed data access over large volume of data without much effort in scaling and tuning (Ferreira et al., 2013). NoSQL architecture is designed to overcome the limited scalability, flexibility, performance, availability, and infrastructure cost issues associated with relational databases (Stonebraker, 2009). They depend on horizontal scalability which scales performance and capacity by increasing the number of nodes, instead of increasing the computer power of a single node as seen in relational databases (Abramova and Bernardino, 2013). An example of a no SQL database is a graph database Fig. 3. Data are stored as graph in the form of nodes (objects) and edges (relationships). The primary goal is to visualize the relationships between nodes. There are now many tools available to design and develop moSQL databases, such as Neo4j (2018) and MongoDB (2018).

Graph databases, which forms the base for noSQL databases use a graph model for semantic queries with nodes, edges and properties to represent and store data. Unlike the relational structures, in graph dabases, millions of connections can traversed

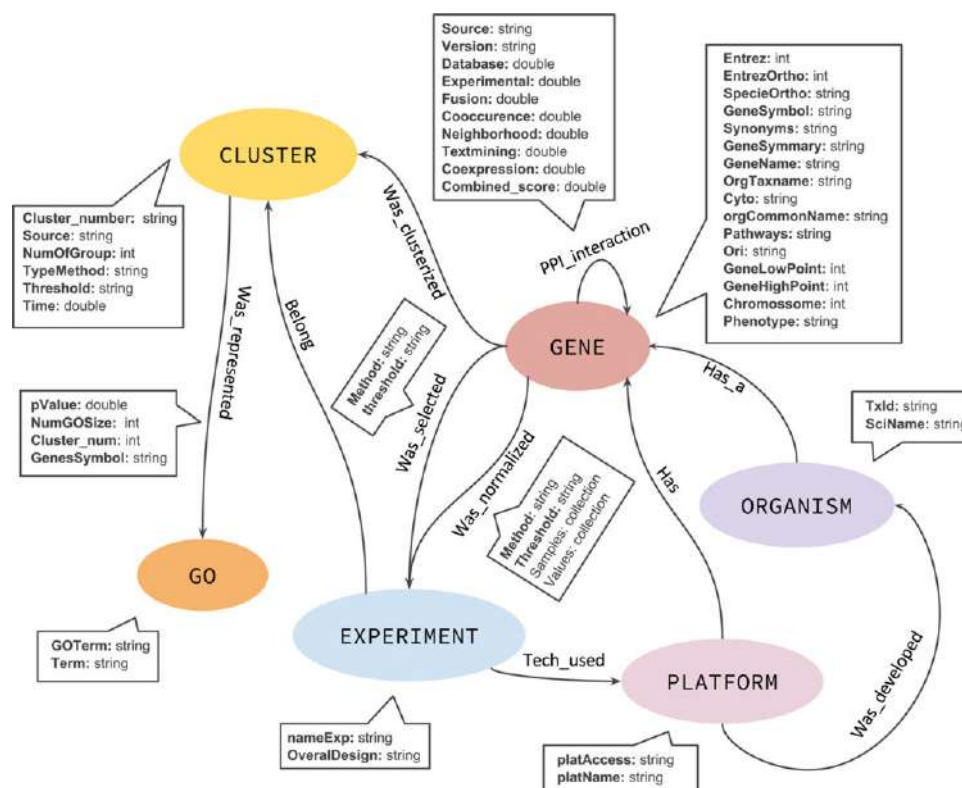


Fig. 3 Graph data model representing the nodes (in oval), relationships (arrows). The descriptive boxes showing the mainly properties in nodes and edges. Source: Costa, R.L., Gadelha, L., Ribeiro-Alves, M., Porto, F., 2017. GenNet: An integrated platform for unifying scientific workflows and graph databases for transcriptome data analysis. PeerJ, 5, p. e3509.

per second per core (Neo4j, 2018). Having a conceptual design, similar to network-model databases in 1970s, graph databases excel at managing linked data and complex queries, independent of the total size of the dataset (Angles and Gutierrez, 2008). Graph databases explore the larger neighborhood around the initial starting points with a pattern and a set of starting points by collecting and aggregating information from thousands or millions of nodes and relationships (Neo4j, 2018). Contrary to relational databases that operate with SQL language, in graph databases, there is no single query language that operates in same fashion as SQL. Query languages such as Gremlin, SPARQL, and Cypher can be used to access graph databases using application programming interfaces (APIs) (Wood, 2012).

Spatial databases

Data in a geometric space contains objects such as coordinates, points, lines, polygons and regions are stored in a spatial database as in objects, rather than pictures or images of a space. It is important to note that spatial databases are different from image databases. The data model encompasses the spatial data types (SDTs) which is used as underlying framework to construct Geographic Information Systems (GIS). Typically, spatial databases uses the relational schema discussed in Section “Relational databases”.

Database System Architectures

Distributed databases

In a typical distributed databases environment, the databases are stored on several computers, which are interconnected via a communication media such as a network connection. These databases are loosely coupled in terms of physical specifications and hardware however in terms of the design schema they might have both a local schema and a global schema. A local schema is unique to a particular database whereas a global schema is used to connect all the databases to achieve a specific goal. Distributed databases are also used in situations where there is an attempt to link or integrate a few existing databases, which are built upon different formats or using different architectures. The reason behind a potential integration could be to share the data stored in these distributed databases.

A general structure of distributed database is shown in Fig. 4.

Federated databases

In a federated database environment, several homogeneous or heterogeneous databases are harmonised to function as a single entity in a unified view, through a common structured metadata. Each of the constituent database serves as a distinct, self-sustainable and autonomous component in the federated system. These components acts as nodes in a computer network and may be geographically decentralized. Federated database is one of most common mainstream model in life sciences data integration. An example of federated database architecture is shown in Fig. 5.

Data warehouse

A data warehouse is a repository for storing data which may have been gathered from a source or multiple sources, manually or automatically, via an integration layer that transforms data to meet the criteria of the warehouse. Data warehouse can be

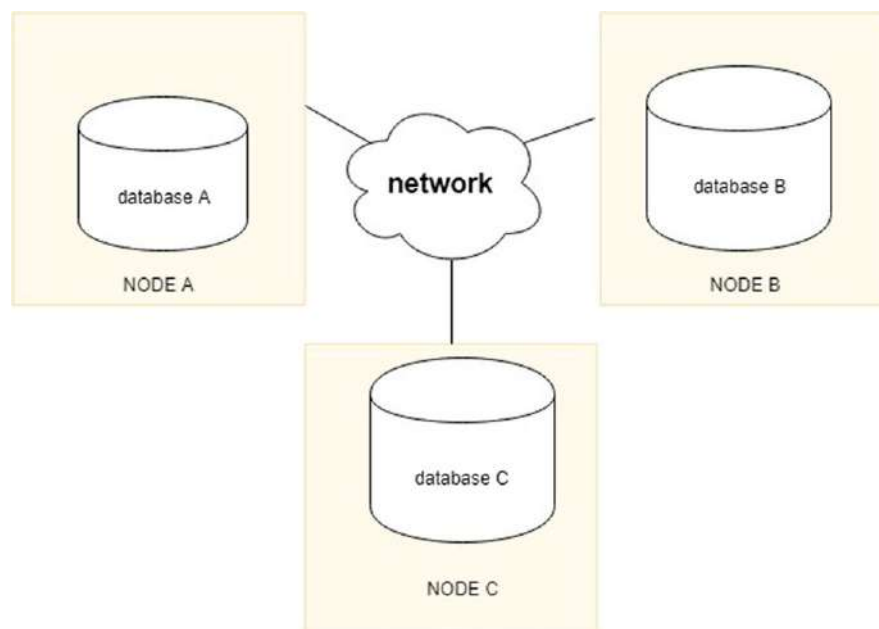


Fig. 4 Distributed Database Structure. The nodes (computers) storing database A, B, and C are distributed and connected via the network.

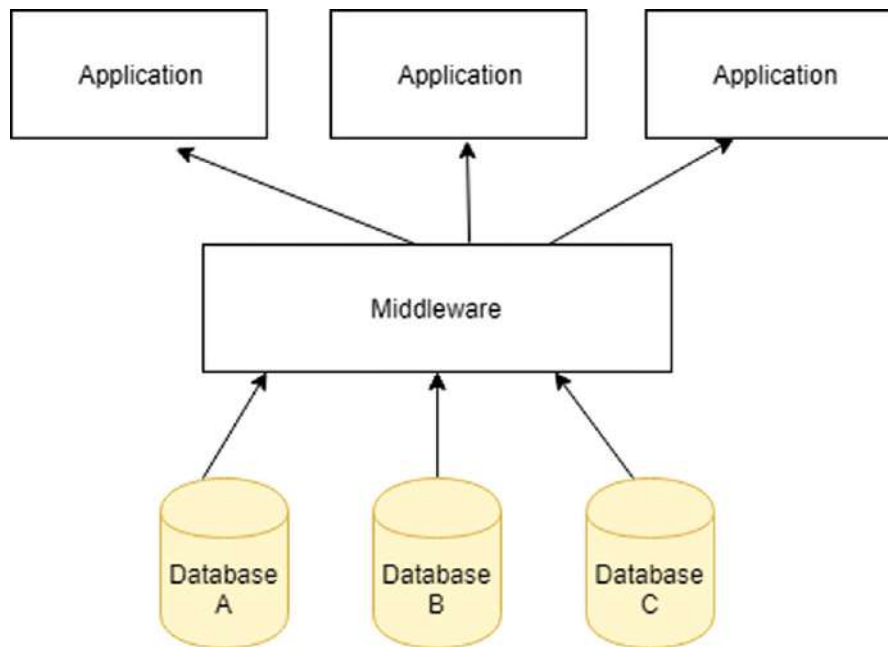


Fig. 5 Federated Database Architecture.

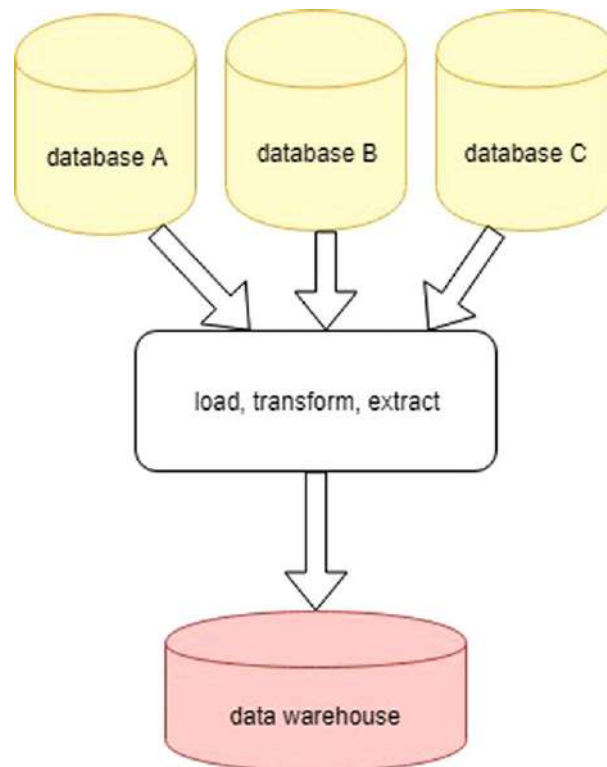


Fig. 6 Data Warehouse Architecture.

conceptualised as a one stop information center large volume of data which is designed under a common framework. In a centralized data warehouse, since data is unified and in homogeneous format, queries are easier to write and gives user better performance than accessing multiple distributed data warehouses. Integration of data in bioinformatics data warehouse and the issues affecting the integration has been presented by [Kormeier \(2014\)](#). An example of data warehouse architecture is shown in [Fig. 6](#).

Uses of Databases

The role of database has evolved from a stand alone specialised system into an integral component of an advanced and sophisticated computing environment. Conventionally, databases were constructed for storage, retrieval and manipulation of data. Today, the role has gone far beyond to knowledge extraction, data mining and visualisation.

Knowledge extraction

Databases are built to support front end applications which ranges from simple query based systems to sophisticated artificial intelligence enabled systems. With the concept of big data that has taken Information Technology into a new era altogether, databases are extensively used for knowledge extraction using algorithms that are run through large volumes of data. Thus, databases now have bigger role to play in many data centric domains to function as platforms for machine readable and machine interpretable format of information. Thiele *et al.* (2010) presented an interesting tool, the Protein Information and Knowledge Extractor (PIKE) which allows researchers to extract information from genomics and proteomics databases. Another good example is Lynx (see “Relevant Websites section”), a database and knowledge extraction engine for integrative medicine (Sulakhe *et al.*, 2016). Thessen and Parr (2014) presented a method on knowledge extraction and semantic annotation of text from the Encyclopedia of Life (Parr *et al.*, 2014).

Data mining

Databases are very useful for storing massive and ever growing amount of data, which can be utilised for discovering patterns and rules, automatically or even semi-automatically. This method is widely known as data mining and is conceived based on logics in database systems (Paramasivam *et al.*, 2014). A clean set of integrated data is usually stored in a data warehouse before the data mining techniques are applied. Some of the key techniques commonly used in data mining include classification, clustering, prediction, association, text mining, link analysis and regression. Data mining is seen as an important concept in databases as the process involves collecting data, creating database and management, analyzing data and finally interpreting data (Han *et al.*, 2000).

Visualisation

Visualizations can be used to interpret meaning from data through comprehension and exploration as large part of human nervous system has evolved to process visual information (Segaran and Hammerbacher, 2009b). An effective visual design aids decision making and interpretation to generate knowledge. One of the main component of data visualisation is the primary data, which conceived in a data warehouse. A data visualisation software or a script can be used to find pattern, correlations or trends to generate knowledge, which could not be seen in textual or numerical content in a database. One good example of data visualisation is using graph based databases whereby the data is retrieved in a visual context.

Biological Databases

Since its birth, database has become a core element of any discipline that produces large amounts of data. In Biology (particularly in genomics), massive amounts of data are being accumulated on a daily basis via experimental work, field work, software analysis, illustrations, microscopic photographs and image acquisition. Stephens *et al.* (2015) compared other Big Data Domains such as, astronomy, youtube and twitter in their recent study on Genomics perceiving to be Big Data science (Fig. 7). They estimated that genomics is leading or at par with the other big domains in terms of terms of data acquisition, storage, distribution, and analysis. Hence, the need to accommodate the data surge in biology is a challenge. This scenario has lead to the construction of not only single databases but also the consortiums managing large amounts of biological data.

Data Phase	Astronomy	Twitter	YouTube	Genomics
Acquisition	25 zetta-bytes/year	0.5–15 billion tweets/year	500–900 million hours/year	1 zetta-bases/year
Storage	1 EB/year	1–17 PB/year	1–2 EB/year	2–40 EB/year
Analysis	In situ data reduction	Topic and sentiment mining	Limited requirements	Heterogeneous data and analysis
	Real-time processing	Metadata analysis		Variant calling, ~2 trillion central processing unit (CPU) hours
	Massive volumes			All-pairs genome alignments, ~10,000 trillion CPU hours
Distribution	Dedicated lines from antennae to server (600 TB/s)	Small units of distribution	Major component of modern user's bandwidth (10 MB/s)	Many small (10 MB/s) and fewer massive (10 TB/s) data movement

doi:10.1371/journal.pbio.1002195.t001

Fig. 7 Four domains of Big Data in 2025. Source: Stephens, Z.D., Lee, S.Y., Faghri, F., *et al.*, 2015. Big data: Astronomical or Genomical? PLOS Biology 13(7), e1002195. Available at: <https://doi.org/10.1371/journal.pbio.1002195>.

The most current online biological databases can be found in the yearly issue of the journal *Nucleic Acids Research* (NAR: see “Relevant Websites section”) and *The Journal of Biological Database and Curation* (DATABASE: see “Relevant Websites section”). Both these journals are freely published by the Oxford Academic Journals (see “Relevant Websites section”). Another important source of biological databases is the BMC Bioinformatics (see “Relevant Websites section”).

In this section, a range of biological databases is presented. Biological databases are defined under three broad categories which are primary databases, secondary databases and specialised databases. Types of biological data includes (but not limited to): nucleotide and protein sequences (b) protein structure (c) microarray and gene expression (d) metabolic pathways (e) species profile (taxonomy) (f) animal model (g) human disease (h) clinical database and others.

Primary Databases

Nucleotide sequence databases were propelled by the necessity to store primary experimental data for further analysis in Bioinformatics. Major sequence databases include the European Nucleotide Archive (ENA) (see “Relevant Websites section”) (Rasko *et al.*, 2011), GenBank (see “Relevant Websites section”) (Benson *et al.*, 2008) and the DNA Data Bank of Japan (DDBJ) (see “Relevant Websites section”) (Jun *et al.*, 2017) and EMBL Nucleotide Sequence Database (see “Relevant Websites section”) (Kanz *et al.*, 2005). These databases are matured sequence databases with search functionality and basic Bioinformatics tools, which are useful to the Bioinformatics community.

A variety of primary sequence databases exist and are widely used in bioinformatics. A significant contribution was made by UniProt (see “Relevant Websites section”) to combine Swiss-Prot, TrEMBL and PIR-PSD protein sequence databases into a unified view. The widespread protein aid (uniprot) is a complete resource for protein sequence and annotation statistics. The uniprot databases consists of uniprot knowledgebase (uniprotkb), the uniprot reference clusters (uniref), and the uniprot archive (uniparc) (Uniport, 2017). Protein Data Bank (PDB: see “Relevant Websites section”) is one of the oldest database that contains database containing experimentally determined three-dimensional structures of proteins, nucleic acids and other biological macromolecules (Burley *et al.*, 2017).

Microarray database contain microarray gene expression data, which can be further analysed, and interpreted. Recently microarray databases have become very popular with the emergence of high-throughput technology, enabling the expression for thousands of genes simultaneously. Example of very common microarray databases are ArrayExpress (see “Relevant Websites section”) from EBI (Brazma *et al.*, 2003) and GEO (Gene Expression Omnibus) (see “Relevant Websites section”) from NCBI (Barrett *et al.*, 2006) and the Stanford Microarray Database (see “Relevant Websites section”) (Sherlock *et al.*, 2001).

As an important and one of the most studied branch of Biology, biochemistry related databases have emerged, such as the metabolic pathway databases. A few good examples are BRAunschweig ENzyme Database, BRENDA (see “Relevant Websites section”) (Schomburg *et al.*, 2002), KEGG PATHWAY Database (see “Relevant Websites section”) (Kanehisa and Goto, 2000) and Metacyc (Caspi *et al.*, 2007).

Phylogenetics is the study the evolutionary history and relationships among individuals or groups of organisms. Despite being one of the important aspect of Biology, there are not many databases available online for phylogenetics. There are two databases worth mentioning which are Phylomedb (see “Relevant Websites section”) (Huerta-Cepas *et al.*, 2010), a public database for complete catalogs of gene phylogenies (phylomes) and treebase (see “Relevant Websites section”) (Sanderson *et al.*, 1994).

Secondary Databases

Secondary databases are more focused as these databases contain derived data (by analyzing primary data) to address specific requirements. This section presents significant and popular secondary biological databases which are well represented in their domain.

The Eukaryotic Promoter Database (EPD: see “Relevant Websites section”) comprehensive organisms-specific transcription start site (TSS) collections automatically derived from next generation sequencing (NGS) data (Dreos *et al.*, 2016).

NCBI Reference Sequence Database (RefSeq (see “Relevant Websites section”)) is a comprehensive, integrated, non-redundant, well-annotated set of reference sequences including genomic, transcript, and protein (Pruitt, 2006).

Pfam (see “Relevant Websites section”) is a database of protein families that consists of their annotations and a couple of series alignments generated using fashions. The most current version, Pfam 31.0, was launched in March 2017 and incorporates 16,712 families (Finn *et al.*, 2017a,b). Pfam has moreover been used inside the advent of different assets which includes ipfam, which catalogs region-area interactions inside and between proteins, based totally on information in structure databases and mapping of Pfam domain names onto those structures (Xu and Dunbrack, 2012).

Prosite (see “Relevant Websites section”) is a protein database, which includes entries describing the protein families, domains and practical websites in addition to amino acid styles and profiles in them, manually curated via a team of the Swiss Institute of Bioinformatics and tightly included into Swiss-Prot protein annotation. Prosite was developed in 1988 by Amos Bairoch, who also directed the institution for more than two decades (Prosite introduction, 2017).

The PRIDE PRoteomics IDentifications (PRIDE) (see “Relevant Websites section”) is a data warehouse on proteomics data, including protein and peptide identifications, post-translational modifications and supporting spectral evidence.

SCOP2 (see “Relevant Websites section”) is a successor of Structural classification of proteins (SCOP) focusing on proteins that are structurally characterized and deposited in the PDB. These proteins are organised according to their structural and evolutionary relationships in a complex graph network. Each node represents a relationship of a particular type and is exemplified by a region of protein structure and sequence (Andreeva *et al.*, 2013).

CATH (Class, Architecture, Topology, Homology) (see “Relevant Websites section”) is a protein structure classification database containing annotations for over 235,000 protein domain structures and includes 25 million domain predictions (Sillitoe *et al.*, 2014).

Protein-protein interaction database are useful source of secondary databases. Two major works in this area include BioGRID and STRING. The Biological General Repository for Interaction Datasets (BioGRID: see “Relevant Websites section”) is a public database of genetic and protein interactions curated from the primary biomedical literature for model organism species and humans (Chatr-Aryamontri *et al.*, 2014) whereas the Search Tool for the Retrieval of Interacting Genes (STRING: see “Relevant Websites section”) contains the Predicted functional associations between proteins (Szklarczyk *et al.*, 2014).

Specialised Databases

Other than the above mentioned categories, there are specialised biological databases available for the scientific community. A few examples are Phylogenomic Database for Plant Comparative Genomics (GreenPhylDB: see “Relevant Websites section”), the Plant Secretome and Subcellular Proteome Knowledgebase (PlantSecKB: see “Relevant Websites section”), TreeBASE (see “Relevant Websites section”), an open-access database of phylogenetic trees and associated data, the Plant Alternative Splicing database (PASD: see “Relevant Websites section”), Comparative Toxicogenomics Database (CTD: see “Relevant Websites section”), Drosophila Genes & Genomes (FlyBase: see “Relevant Websites section”), WormBase (see “Relevant Websites section”), AceDB (see “Relevant Websites section”), the Arabidopsis Information Resource (TAIR: see “Relevant Websites section”), Online Mendelian Inheritance in Man (OMIM: see “Relevant Websites section”) which is an online catalogue of human genes and genetic disorders and the *Saccharomyces* Genome Database (SGD: see “Relevant Websites section”).

Integrated Databases

Integrated databases are databases that are unified under a platform, with the aim to offer a suite of applications for secondary uses of the data. These days integrated databases are preferred by researchers due to their usability as one stop centres for knowledge extraction. A large number of biological databases are produced or are managed by the National Center for Biotechnology Information (Coordinators, 2016), which contains a myriad of biological content. Other initiatives are the Kyoto Encyclopedia of Genes and Genomes (KEGG: see “Relevant Websites section”), GeneCards (see “Relevant Websites section”), the InterPro Consortium (Finn *et al.*, 2017a,b), the European Molecular Biology Laboratory (EMBL) Nucleotide Sequence Database (see “Relevant Websites section”) and the Global Biodiversity Information Facility (GBIF: see “Relevant Websites section”). These databases are more like consortiums managing and integrating sources of information to provide a unified access to users.

Other Biological Databases

Biodiversity (taxonomy) databases

Taxonomy databases, also referred as biodiversity databases have been developed much before sequence databases are introduced. Taxonomy, being a very old and matured branch of Biology, has produced abundant of data, particularly on species profiling. Almost every country listed as a megabiodiversity country has produced a database of their very own indigenous organisms. Some of the most important biodiversity initiatives are started by GBIF (see “Relevant Website section”), Taxonomic Data Working Group (TDWG: see “Relevant Websites section”) and Encyclopedia of Life (EoL: see “Relevant Website section”). These global initiatives encourage sharing of biodiversity data which was previously stored in data silos managed by smaller groups. The unprecedented volume of biodiversity data has caused a momentous increase in the number of online biodiversity databases hosted in separate infrastructure which are managed by research groups. Example of some very popular databases include the Catalogue of Life (see “Relevant Websites section”), Integrated Taxonomic Information System (ITIS: see “Relevant Websites section”) and FishBase (see “Relevant Websites section”).

Animal model database

Animal models are mostly used in biomedical research which is an important research area to demonstrate biological significance in experiments. Due to this, animal model, particularly mouse model databases are adevent these days. A few examples are mentioned here to demonstrate the kind of databases produced in this domain.

The MUGEN mouse database (MMdb: see “Relevant Websites section”) is a repository of murine models of immune processes and immunological diseases.

Mouse Genome Informatics (MGI: see “Relevant Websites section”) MGI is the international database resource for the laboratory mouse, providing integrated genetic, genomic, and biological data to facilitate the study of human health and disease.

The Mouse Phenome Database (MPD: see “Relevant Websites section”) is one of the most widely used resource for primary experimental trait data and genotypic variation.

International Mouse Strains Resource (IMSR: see “Relevant Websites section”) is a database on mouse strains, stocks, and mutant ES cell lines available worldwide, including inbred, mutant, and genetically engineered strains.

Other mouse model databases are The European Mouse Mutant Archive (see “Relevant Websites section”) and the Rat Resource and Research Center (see “Relevant Websites section”).

Besides mouse, other animals database on Zebra, Zebrafish and Drosophila have been published. One good example is the Zebrafish models (ZF-Models: see “Relevant Websites section”), which is an information system on zebrafish that can be referred for producing and identifying disease models, drug targets and insight into pathways of gene regulation applicable to human development and disease. Other examples are Zebra International Resource Center (see “Relevant Websites section”) and Bloomington Drosophila Stock Center (BDSC: see “Relevant Websites section”).

Clinical/health database

Clinical and Health databases must not be sidelined in the discussion about biological databases. While we will not be able to mention all the works done in this area, it is crucial to highlight a few significant examples to demonstrate the kind of databases produced for this purpose. The National Center of Biotechnology Information (NCBI), as part of their database initiative, has produced two important clinical and health databases which are ClinVar (see “Relevant Websites section”), a database on the reports of the relationships among human variations and phenotypes; and MedGen (see “Relevant Websites section”) a database containing information related to human medical genetics, such as attributes of conditions with a genetic contribution. Database of Genotypes and Phenotypes (dbGaP: see “Relevant Websites section”) is an archive of data and results from studies that have investigated the interaction of genotype and phenotype in humans. PubMed Health (see “Relevant Websites section”) is the world’s largest digital medical library. MEDLINE (see “Relevant Websites section”) contains journal citations and abstracts for biomedical literature from around the world while PubMed (see “Relevant Websites section”) is an archive of more than 28 million citations (links to full text content from PubMed and publisher websites) for biomedical literature from MEDLINE, life science journals, and online books.

Clinical and Health databases focusing on diseases are also useful resource for biologists and medical scientists. There are some prominent disease databases published online such as MalaCrads (see “Relevant Websites section”), the Autoimmune Disease Database (see “Relevant Websites section”), Inflammatory Bowel Diseases (IBD: see “Relevant Websites section”), KEGG Disease Database (see “Relevant Websites section”) and LiverWiki (see “Relevant Websites section”), a wiki-based database for human liver. The Diseases Database ver 2.0 contain a whole range of database in their portal, which is available at the website provided in “Relevant Websites section”. Finally, a comprehensive collection of human related biological databases is presented by [Zou et al. \(2015\)](#).

Biological Databases Case Studies: Genecards and Cancer Genome Atlas

Genecards (see “Relevant Websites section”)

In the early 80s, storing sequence data was not easy due to cost and infrastructure limitations. Nevertheless, many institutions have started to store their data in primary databases which are available in isolated platforms and are focused on specific subject of study. In order to gather these scattered data, The Weizmann Institute of Science Crown Human Genome Center (<http://www.weizmann.ac.il>) developed a database called *GeneCards* in 1997. In the early stage, this database was dealing mostly with human genome information, human genes, the encoded protein’s function and related diseases.

Later, it incorporated automatically-mined genomic, proteomic and transcriptomic data, as well as orthologies, disorder relationships, snps, gene expression, gene characteristic, and hyperlinks for ordering assays and antibodies ([Genecards, 2018](#)). Currently it serves as a complete, authoritative compendium of annotative information about human genes that has been broadly used for almost 15 years ([Safran et al., 2010](#)). Genecards has blossomed into a collection of equipment (along with Genedecks, Genealacart, Geneloc, Genenote and Geneannot) for a spread of analyses of each single human genes and units due to its viewing information and deep links associated with unique genes and constant improvements, over the years. ([Stelzer et al., 2011](#)) (**Figs. 8–11**).

GeneCards versions

GeneCards Version 1.xx- contained gene-centric information, automatically mined and integrated from a myriad of databases resulting in a web-based *card* for each of human gene entry ([GeneCards, 2018](#)).

GeneCards Version 2.xx- the information is stored in flat files, one file per gene, all indexed to enable full-text searches ([GeneCards, 2018](#)).

GeneCards Version 3.xx- introduced a persistent object/relational model of all the data entities and relationships displaying diverse functions of single genes by extracting various slices of attributes of the genes ([GeneCards, 2018](#)). It also hosts the new seek facility by offering hyperlinks to a sample gene and its diverse sections ([Safran et al., 2010](#)).

GeneCards Version 4.xx- the hybrid system is eliminated, moving to a robust infrastructure based entirely on a new and well defined relational database ([GeneCards, 2018](#)).

GeneCards applications

GeneDecks exploits Genecards unique wealth of combinatorial annotations to discover similar (associate) genes, and to perform quantitative descriptor enrichment analyses for figuring out set-shared annotations. Some of these functions are available online, through the Genecards website, while others independent studies. Some of the Genecards applications are described below.

SYNLET – Regulatory control networks of synthetic lethality via Genedecks mainly addresses robustness of phenotypic characteristic on the basis of artificial lethality, as proposed for novel most cancers remedy regimes. It derives novel standards, methodologies and algorithms for annotation and evaluation of regulatory networks, with attention on tumorigenesis and drug resistance. It aims

The screenshot shows the GeneCards website interface. At the top, there is a navigation bar with links to GeneCards, MalaCards, LifeMap Discovery, PathCards, TGex, VarDict, GeneAnalytics, GeneAlaCart, and GenesLikeMe. Below this is a search bar with a 'Keywords' dropdown and a 'Search Term' input field. The main content area features the GeneCards logo and a description: 'GeneCards is a searchable, integrative database that provides comprehensive, user-friendly information on all annotated and predicted human genes. It automatically integrates gene-centric data from ~125 web sources, including genomic, transcriptomic, proteomic, genetic, clinical and functional information.' A large orange circular icon with a DNA helix is also present. Below the description is a section titled 'Explore a Gene' with a search bar containing 'ADAM10' and a 'GO' button. Underneath, there are links to 'Jump to section for this gene:' including Aliases, Disorders, Domains, Drugs, Expression, Function, Genomics, Localization, Orthologs, Paralogs, Pathways, Products, Proteins, Publications, Sources, Summaries, Transcripts, and Variants. On the right side, there are sections for 'NGS Analysis Tools' (TGex, VarDict), 'Affiliated Databases' (MalaCards, LifeMap, PathCards, GeneLogic), and 'Analysis Tools' (GeneAnalytics, GeneAlaCart, GenesLikeMe, GeneHancer). At the bottom, there is a 'GeneCards Database Statistics' table.

GeneCards Database Statistics				
		Category	# of Genes	Example Genes
Total GeneCards genes	147,013	Protein-coding	21,360	PDGFRA, ERBB2, ERBB4, ESR1, FGFR1, FGFR2, FGFR3, MET, PDGFRB, CDK4
HGN Approved genes	41,298	RNA genes	98,609	HCP5, DSCR4, C9orf31, MEG3, PRNT, RUSC1-AS1, TERC, DLEU1, C10orf91, C13orf54
Disease genes	11,483	Pseudogenes	16,337	GNRHR2, GUCY1B2, LPAL2, ABCG13, PPP4R1L, PRKY, HLA-H, KIR3DX1, MMP23A, SERHL
Hot genes	500	Genetic loci	1,819	DYX1, ST2, AAVS1, IFNR, NM, AZF1, BCL5, FRAXA, ST3, ST8
		Gene Clusters	18	PCDHA@, IGLV@, RNSS1@, PCDHG@, IGKV@, PCDHB@, SNORD116@, HCCA@, HOXD@, HOXS@
		Uncategorized	9,770	C20orf181, ERYMR1-1, TRAV35, TRBV11-2, ENSG00000235978, TRAJ15, WBSGF2, TRAJ8, PCDHACT, ENSG00000254859

Fig. 8 The screenshot image of the domain GeneCards, human genome database. Retrieved from: <https://www.genecards.org/>.

to explore key proteins of cellular escape mechanisms that conquer lethality of medicine and find approaches to block them, utilizing siRNA. Genedecks became one of the major equipment which served the consortium to choose candidates for the siRNA experiments. The set distiller mode of Genedecks permits the identification of annotation descriptors shared via experimentally-received gene sets, permitting the evaluation in their belonging to precise practical lessons, for this reason a really appropriate choice of inactivation targets (Safran *et al.*, 2010).

SysKid – Systems biology towards novel chronic kidney disease diagnosis and treatment explores chronic kidney ailment, with diabetes mellitus and high blood pressure being the maximum widely wide-spread causative conditions. It assumes that despite it being numerous in etiology, the underlying molecular pathophysiology of different manifestations of this condition can be similar. The venture goals at acquiring a unified view on the disease within the scope of systems biology, primarily based on high-throughput omics analysis. The Genecards version 3.xx database and superior seek engine are instrumental for the combination procedure and in figuring out the maximum promising sickness markers (Stelzer *et al.*, 2011). For instance, Genecards might be used to perceive applicable genes for metabolomics hints. In this framework, a 'Genecards 100k' attempt is presently beneath way, aiming at increasing the range of gene entries closer to one hundred 000. This will be finished mainly through a big growth of Genecards' scope within the realm of non-protein-coding RNA genes, and resolving a massive wide variety of genes currently entitled 'uncategorized'. In parallel, a challenge-particular database (Genekid) could be created, to residence incoming omics facts in Genecards-well suited tables, for that reason facilitating the systems biology analyses (Safran *et al.*, 2010).

Future of genecards

The success of applications described above will further upgrade GeneCards core features. An exciting destiny undertaking is to plot a set of rules for unifying disorder names/descriptions and permit those tables to be merged. An increase in pathway repertoire and inclusion of public area (e.g., reactome) and/or additional industrial pathways are expected. In the future, the 'feature' phase will also consist of animal models from species other than mouse (Safran *et al.*, 2010).

The Cancer Genome Atlas (TCGA) (see "Relevant Websites section")

The National Institutes of Health (NIH) mounted The cancer Genome Atlas (TCGA) to generate complete, multi-dimensional maps of the important thing genomic changes in important types and subtypes of cancer (National Institutes of Health, 2017).



SYNLET home
consortium
disseminations
meetings
internal

SYNLET Project

Project Summary

This project addresses robustness of **phenotypic function** on the basis of highly interlinked **network topologies** on a very practical level: **Synthetic Lethality** – as proposed for novel cancer treatment regimes. Function in this context is the property of transformed cells to continuously divide, and robustness is encoded by regulatory elements on the genomic and proteomic level triggering drug treatment escape as an emergent property: chemotherapy resistance. This project will derive novel concepts, methodologies, and their algorithmic implementation for annotation and analysis of general regulatory networks - with particular focus on network robustness and escape routes toward maintaining function. In parallel, computational genomics will be applied on a gene expression data set derived from a unique, systematic collection of chemotherapy resistant cancer cell lines. These real world data encode a set of regulatory escape mechanisms to overcome the lethality imposed by a specific drug. The general concepts and approaches – focusing on dynamical levels – will subsequently be calibrated and merged with results from computational genomics – focusing on a static system representation – to gain insight into the mechanisms of cancer cell robustness, but in particular to identify key proteins of cellular escape mechanisms to overcome lethality of drugs. Blocking these proteins may result in synthetic lethality including re-sensitization of chemoresistant cancer cells to cytotoxic drugs. To prove the validity of our Complex Systems Analysis approach we will test generated hypotheses on synthetically lethal hubs in the cellular control network via experimental knock down experiments utilizing siRNA.

EU foundation

Proposal Full Title	Regulatory Control Networks of Synthetic Lethality
Proposal Acronym	SYNLET
Date of Preparation	February 15, 2006
Version No.	2.0
Type of Instrument	Specific Targeted Research Project
Submission Stage	Full Proposal
Activity Code Addresses	NEST-2005-Path-COM

Coordinator:
Dr. Michaelis Martin
e-mail: martin.michaelis@blue-drugs.com
Fax: +49 69 67 86 65-9172

Frankfurter Stiftung für Krebskranke Kinder

<http://www.kinderkrebsforschung-frankfurt.de>

Hilfe für krebskranke Kinder e.V.

<http://www.hilfe-fuer-krebskranke-kinder.de>

Fig. 9 The screenshot image of the domains SYNLET. Retrieved from: <http://synlet.izbi.uni-leipzig.de/>.

The Cancer Genome Atlas (TCGA) is a venture began in 2005, to catalogue genetic mutations answerable for cancer, the usage of genome sequencing and bioinformatics. TCGA applies excessive-throughput genome analysis strategies to enhance the ability to diagnose and treat cancer via a detailed investigation of the genetic basis of this ailment (Tomczak *et al.*, 2015; The National Genome Atlas, 2017).

In 2006, TCGA targeted on characterization of 3 varieties of human cancers: Glioblastoma Multiforme, lung, and ovarian cancer. This attempt initiated by NCI and NHGRI in 2006 developed the rules, production pipeline, collaborative studies network, databases and analytical equipment important for TCGA's big-scale analysis of cancer genomics. The pilot's characterizations of brain tumours and ovarian cancer have proven that this systematic, excessive-quantity approach can generate excellent novel discoveries, useful for scientific researchers around the globe (The National Genome Atlas, 2017).

In 2009, TCGA accelerated into Phase II, which planned to complete the genomic characterization and collection evaluation of 20–25 special tumour types through 2014. TCGA passed that aim, characterizing 33 cancer types together with 10 rare cancers. The project was split between genome characterization centres (GCCs), which carry out the sequencing, and genome records analysis facilities (GDACs), which carry out the bioinformatics analyses (The National Genome Atlas, 2017). With its pipeline in area, TCGA correctly acquired samples of tumours and coupled ordinary tissue, rapidly sequenced the DNA and RNA of the samples, and comprehensively characterized the genomes the usage of seven genomic platforms (The National Genome Atlas, 2017).

In December 2013, while TCGA finalized its collection of matched tumour and everyday samples, sufficient tissue have been accumulated to map the genomic traits of extra than 30 cancer types. TCGA evaluation operating agencies, groups of scientists from across the United States of America, have now comprehensively studied 33 cancer types and subtypes, along with 10 rare cancers (The National Genome Atlas, 2017).

This pioneering effort to map and analyse most cancers genomes in a huge-scale, systematic way has modified the way most cancers are studied, and could ultimately rework the way most cancers are dealt with because TCGA information is loose to use, it significantly reduces the costs and expedites the manner of drug development, allowing both public and private quarter researchers

NIH THE CANCER GENOME ATLAS
National Cancer Institute
National Human Genome Research Institute

Launch Data Portal | Contact Us | For the Media

Search

Home About Cancer Genomics Cancers Selected for Study Research Highlights Publications News and Events About TCGA

TCGA's Study of Soft Tissue Sarcoma

TCGA's analysis of six major types of adult soft tissue sarcomas reveals frequent copy number alterations, low mutational loads, and a diverse array of underlying molecular mechanisms. Certain sarcomas may benefit from immunotherapy.

Learn More

TCGA's Pan-Cancer Atlas

TCGA's Sarcoma Study

Expanded Bladder Cancer Study

Cancers Selected for Study

Launch Data Portal

The Genomic Data Commons (GDC) Data Portal is an interactive data system for researchers to search, download, upload, and analyze harmonized cancer genomic data sets, including TCGA.

Questions About Cancer

Visit www.cancer.gov

Call 1-800-4-CANCER

Use LiveHelp Online Chat

Multimedia Library

- Images
- Videos and Animations
- Podcasts
- Interactive

Stay Connected

- Sign up for email updates
- RSS newsfeeds
- Twitter
- YouTube

TCGA in Action

December 2016
CASE STUDY: A Researcher Mined TCGA Data to Study Her Own Ovarian Cancer
Faced the bleak prognosis associated with stage IIIC ovarian cancer, Shirley Pepke, Ph.D., used her computational biology background to study her own cancer in comparison to TCGA data. Pepke has now been in remission for a year. Image Credit: Gus Ruelas/USC

September 2016
CASE STUDY: TCGA Data Leveraged for Developing FDA-Designated Breakthrough Therapy
Pharmaceutical company Loxo Oncology used TCGA data in the research that led them to develop FDA-designated Breakthrough Therapy LOXO-101, a promising new targeted therapy.

[More Stories](#)

News and Announcements

May 01, 2018
TCGA Releases The Pan-Cancer Atlas
TCGA's Pan-Cancer Atlas of insights and overarching themes on cancer culminates the 10+ year project.

November 02, 2017
TCGA Soft Tissue Sarcoma Study Charts Diverse Landscape of Molecular Changes
TCGA's analysis of six major types of adult soft tissue sarcomas reveals frequent copy number alterations, low mutational loads, and a diverse array of underlying molecular mechanisms. Certain sarcomas may benefit from immunotherapy.

[View All](#)

Research Briefs

June 2017
TCGA Data Used to Study Ties Between Tumor Genetics and Evolutionary History
A study using TCGA data investigated the theory that cancer cells revert to ancient cellular processes. The researchers found repression of newer cellular processes and activation of older genes, providing insight into the hallmarks of cancer common across different types.

[View All](#)

Perspectives

October 2015
Genomic Contributions to Breast Cancer Disparities
Dr. Tanya Keenan shares her study's findings on the genomic contributions to breast cancer disparities and advocates for inclusion in breast cancer research.

[View All](#)

Home | Contact Us | Web Site Policies | Accessibility | FOIA | RSS

U.S. Department of Health and Human Services | National Institutes of Health | National Cancer Institute | National Human Genome Research Institute | USA.gov

NIH...Turning Discovery Into Health®

Fig. 10 The screenshot image of The Cancer Genome Atlas (TCGA). Available from: <https://cancergenome.nih.gov/>.

to pursue centred cures geared toward the precise pathways in a positive most cancers kind or subtype. This new level of insight, supplied by means of TCGA will chart a brand new path for most cancers research and pioneer the manner to extra effective, individualized techniques for helping every affected person with most cancers (The National Genome Atlas, 2017).

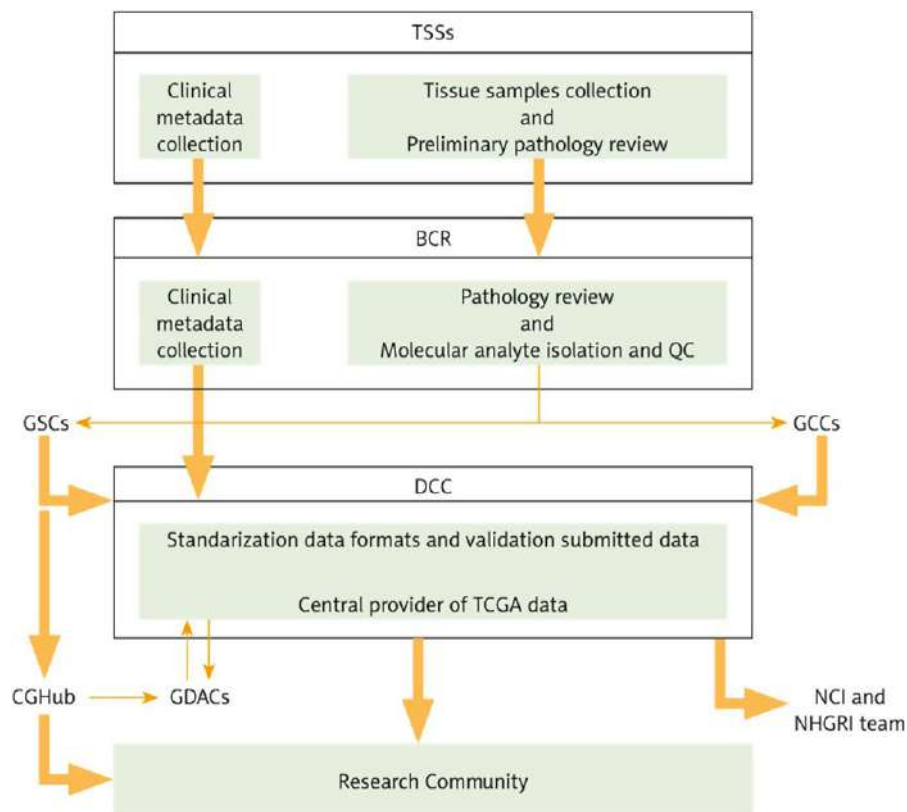


Fig. 11 The Cancer Genome Atlas (TCGA) Research Network Centres flowchart. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4322527/figure/F0001/>.

The TCGA initiative scheduled 500 affected person samples and used distinctive strategies to investigate the patient samples. Techniques include gene expression profiling, reproduction quantity version profiling, SNP genotyping, genome huge DNA methylation profiling, microRNA profiling, and exon sequencing. TCGA is sequencing the whole genomes of some tumours, which include at least 6000 candidate genes and microRNA sequences. (The National Genome Atlas, 2017).

This catalogue serves as an effective useful resource for a brand new generation of research aimed towards developing higher strategies for diagnosing, treating and preventing most cancers. TCGA also presents a version for plenty different genome mapping projects (National Cancer Institute, 2017).

Mutually led via the countrywide cancer Institute (NCI) and the country wide Human Genome Research Institute (NHGRI), TCGA is changing the manner genetic most cancers studies are completed. Following the instance set with the aid of the Human Genome undertaking, TCGA's collaborative studies network has added together researchers from diverse disciplines and more than one establishment to provide precious statistics units to be used by using the worldwide studies community (The National Genome Atlas, 2017).

The shape of TCGA is properly organised and includes several cooperating centres liable for collection and sampleprocessing, observed through high-throughput sequencing and sophisticated bioinformatics data analyses.

First, special tissue supply websites (TSSs) accumulate the desired bio specimens (blood, tissue) from eligible cancers patients and deliver them to the Biospecimen Core Resource (BCR). Subsequently, the BCR catalogue validates and then submit the scientific data and metadata to the Data Coordinating Center (DCC) and provide molecular analyses for the Genome Characterization Centres (GCCs) and Genome Sequencing Facilities (GSCs) for similar characterization and excessive-throughput sequencing. Then, series-associated facts are deposited inside the DCC. The Genome Characterization facilities also post hint documents, sequences, and alignment mappings to NCI's cancer Genomics Hub (CGHub) secure repository. The generated genomic records is made available to the research community and Genome Data Analysis Centers (GDACs). The GDACs provide new facts-processing, evaluation, and visualisation gear to the whole studies community to facilitate broader use of TCGA data. Furthermore, the statistics generated by means of the TCGA studies network is centrally managed on the DCC and deposited into public databases (TCGA Portal, NCBI's hint Archive, CGHub), allowing scientists to obtain cancer datasets.

TCGA applications

The Cancers Imaging Archive, TCIA (see "Relevant Websites section") TCIA is a service created through the NCI to host a huge range of medical images (radiological imaging statistics), from TCGA instances, therefore it supports imaging phenotype-genotype research. The data is de-identified and available for public download.

Berkeley Morphometric Visualisation and Quantification from H&E sections (see “Relevant Websites section”) is a data repository of computed histology-primarily based snap shots of various tumour samples from TCGA cases, and is backed via the Lawrence Berkeley National Laboratory.

The Cancer Digital Slide Archive (CDSA: see “Relevant Websites section”), is an online interactive device for viewing and annotating diagnostic and tissue slide photographs of different tumour kinds from TCGA assignment.

The Broad GDAC Firehose (see “Relevant Websites section”) is an analytical infrastructure created to coordinate the flow of cancers datasets, using various quantitative algorithms along with GISTIC, MutSig, Clustering, and Correlation.

The MD Anderson GDAC’s MBatch (see “Relevant Websites section”) is a website that permits scientists to discover and quantify the batch effects accompanying TCGA facts set, currently in line with hierarchical clustering and more desirable PCA plots.

Cancers Genome Workbench, (CGWB: see “Relevant Websites section”), is a utility evolved with the aid of the National Cancer Institute (NCI) to combine and display pattern-level genomic and transcription changes in numerous cancers, by obtaining information from several cancers projects, including TCGA. The major visitors in CGWB are incorporated tracks view, heat map view, and an alignment viewer known as Bambino.

Future perspectives of TCGA

Systematic advances in cancer genomics provided via TCGA have revealed a brand new approach of molecular biology in cancer. The application of high-throughput technology, boosted with bioinformatics analytics has been successful in highlighting the similarities and differences in the genomic structure of every cancer and across more than one kind. The TCGA has furnished a large amount of publicly available data on most cancers genetic and epigenetic profiles, highlighting candidate cancer biomarkers and drug objectives. Moreover, translation of cancer genomics into therapeutics and diagnostics will offer potential to increase personalized cancer medicinal drug. The ultimate goal for scientists is to develop very high end bioinformatics tools to get rid of noise in data and enhance the decision of the analysis, for cutting edge discoveries. In the future, all novel findings will facilitate diagnosis, treatment, and cancer prevention. Lately, researchers have new discoveries in cancer research by trying to “teach” a device – an artificially clever laptop, called Watson – to aid doctors in diagnosing patients. In time to come, these rapid advances can hopefully be incorporated into clinics (Tomczak *et al.*, 2015).

Results and Discussion

Results are presented in terms of database models, tools and architecture employed in developing biological databases.

Database Models

Predominantly, traditional relational model has been used in the design and development of biological databases. Some prominent example of relational databases are GenBank, EMBL, SWISS-PROT and Protein Information Resource(PIR) and CBS Genome Atlas Database.

However, with the recent development in the semantic web technologies, many biological ontologies have been created that uses object based database models using ontologies and nosql database tools. One good example is BioPortal (see “Relevant Websites section”) which is a repository for biomedical ontologies. It currently contains 704 ontologies. Other prominent examples are BioPAX (see “Relevant Websites section”), Cell Cycle Ontology (CCO: see “Relevant Websites section”), Gene Expression Knowledgebase (GexKB: see “Relevant Websites section”), Disease Ontology (see “Relevant Websites section”), Gene Ontology Consortium (see “Relevant Websites section”), Sequence Ontology (see “Relevant Websites section”) and SNOMED CT (see “Relevant Websites section”). A good example of ontology based database is the Fish Ontology (FISHO: see “Relevant Websites section”) which is shown in Fig. 12.

The Human Protein Reference Database (HPRD: see “Relevant Websites section”) (Prasad *et al.*, 2009), on the contrary, was developed using the traditional object based data model.

A summary of database models and development tools employed in biological databases are presented in Table 1.

Database Architecture

In terms of database architecture, all three approaches: Distributed, Federated and Data Warehouse have been utilised in biological databases. Example of distributed databases include Ensembl, WormBase, and the Berkeley Drosophila Genome Project whereas examples of federated databases are TwinNET (Muilu *et al.*, 2007), ENCODE (Blankenberg *et al.*, 2007), EBI search (Park *et al.*, 2017), SPINE2 (Goh *et al.*, 2003), Cancer Biomedical Informatics Grid (Saltz *et al.*, 2006), NIF (Gardner *et al.*, 2008), Biomedical Informatics Research Network (Ashish *et al.*, 2010), Biomedical Investigations (Taylor *et al.*, 2008), EdgeExpressDB (Severin *et al.*, 2009) and Minimal Information About Neural Electromagnetic Ontologies (Frishkoff *et al.*, 2011). Examples of biological databases that are built on data warehouse concept are Pathway Commons (Cerami *et al.*, 2010), String (Szklarczyk *et al.*, 2010), CBS Genome Atlas Database (Hallin and Ussery, 2004), BioMart (Haider *et al.*, 2009), BrainMap (see “Relevant Websites section”), PubBrain (see “Relevant Websites section”).

Table 1 Summary of data modeling techniques in biological databases

Database	Data modeling	Function	References (Harvard Style)
European Nucleotide Archive (ENA) (https://www.ebi.ac.uk/ena)	noSQL	Database of raw sequencing data, sequence assembly information and functional annotation	Nicole Silvester, Blaise Alako, Clara Amid, Ana Cerdeño-Tarraga, Laura Clarke, Iain Cleland, Peter W Harrison, Suran Jayatilaka, Simon Kay, Thomas Keane, Rasko Leinonen, Xin Liu, Josué Martínez-Villacorta, Manuela Menchi, Kethi Reddy, Nima Pakseresht, Jeena Rajan, Marc Rossello, Dmitriy Smirnov, Ana L Toribio, Daniel Vaughan, Vadim Zalunin, Guy Cochrane; The European Nucleotide Archive in 2017, <i>Nucleic Acids Research</i> , Volume 46, Issue D1, 4 January 2018, Pages D36–D40. https://doi.org/10.1093/nar/gkx1125
GenBank (https://www.ncbi.nlm.nih.gov/genbank/)	Flat file	An annotated collection of all publicly available DNA sequences	NCBI GenBank. https://www.ncbi.nlm.nih.gov/genbank/
Protein Data Bank (PDB) (https://www.wwpdb.org/)	Flat file (XML)	Expertise in the techniques of X-ray crystal structure determination, NMR, cryoelectron microscopy and theoretical modeling.	Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov I. N. & Bourne, P. E. (2000). The Protein Data Bank. <i>Nucleic Acids Research</i> , 28(1), 235–242.
MEDLINE (https://www.nlm.nih.gov/bsd/pmresources.html)	Flat file (XML)	Database on journal citations and abstracts for biomedical literature from around the world	
ArrayExpress (https://www.ebi.ac.uk/arrayexpress/)	Flat file (XML)	Database of microarray gene expression data	Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G. G., Oezcimen, A., Rocca-Serra, P. & Sansone, S.-A. (2003). ArrayExpress – A public repository for microarray gene expression data at the EBI. <i>Nucleic Acids Research</i> , 31(1), 68–71.
GEO (https://www.ncbi.nlm.nih.gov/geo/)	Flat file and relational	Database of high-throughput gene expression data	Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., Edgar, R. (2009). NCBI GEO: archive for high-throughput functional genomic data. <i>Nucleic Acids Res.</i> , 37 (Database issue), D885–D890. https://doi.org/10.1093/nar/gkn764
EdgeExpressDB (http://fantom.gsc.riken.jp/4/edgeexpress/about/)	Relational (snow flake schema design)	A database for interpreting biological networks and comparing large high-throughput expression datasets	Severin, J., Waterhouse, A.M., Kawaji, H., Lassmann, T., van Nimwegen, E., Balwierz, P.J., Hoon, M.J., Hume, D.A., Carninci, P., Hayashizaki, Y., Suzuki, H., Daub, C.O. & Forrest, A.R. (2009). FANTOM4 EdgeExpressDB: An integrated database of promoters, genes, microRNAs, expression dynamics and regulatory interactions. <i>Genome Biology</i> , 10(4), R39. https://doi.org/10.1186/gb-2009-10-4-r39
BRENDA (https://www.brenda-enzymes.org/)	Relational	Database on identified enzymes	Barthelme, J., Ebeling, C., Chang, A., Schomburg, I., & Schomburg, D. (2007). BRENDA, AMENDA and FRENDA: The enzyme information system in 2007. <i>Nucleic Acids Research</i> , 35(Database issue), D511–D514. https://doi.org/10.1093/nar/gkl972
KEGG PATHWAY (http://www.genome.jp/kegg/pathway.html)	Flat file (XML)	A database resource for network of protein products, including functional RNAs.	Kanehisa and Goto (2000) KEGG: kyoto encyclopedia of genes and genomes. <i>Nucleic acids research</i> , 28(1), pp.27–30.
Metacyc (https://metacyc.org/)	Flat file	Database describing metabolic pathways and enzymes from all domains of life.	Caspi <i>et al.</i> , 2016, “The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases,” <i>Nucleic Acids Research</i> 44:D471–D480
Human metabolic atlas (http://www.metabolicatlas.org)	Relational model	Resource for human metabolism	Natapol Pornputtapong, Intawat Nookaew, Jens Nielsen; Human metabolic atlas: An online resource for human metabolism, <i>Database</i> , Volume 2015, 1 January 2015, bav068. https://doi.org/10.1093/database/bav068
Encyclopedia of Life (eol.org)	Flat File (JSON)	Database containing information about life on Earth	Parr, C.S., Wilson, N., Leary, P., Schulz, K.S., Lans, K., Walley, L., ... Corrigan, Jr., R.J. (2014). The Encyclopedia of Life v2: Providing Global Access to Knowledge About Life on Earth. <i>Biodiversity Data Journal</i> , (2), e1079. Advance online publication. https://doi.org/10.3897/BDJ.2.e1079
Pfam (https://pfam.xfam.org/)	Relational	A collection of protein multiple sequence alignments	<i>The Pfam protein families database: towards a more sustainable future</i> : R.D. Finn, P. Coghill, R.Y. Eberhardt, S.R. Eddy, J. Mistry, A. L. Mitchell, S.C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G.A. Salazar, J. Tate, A. Bateman <i>Nucleic Acids Research</i> (2016) Database Issue 44:D279–D285

(Continued)

Table 1 Continued

Database	Data modeling	Function	References (Harvard Style)
SCOP2 (http://scop2.mrc-lmb.cam.ac.uk/)	Relational model Graph Data Model	Structural classification of proteins	Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., & Murzin, A.G. (2014). SCOP2 prototype: A new approach to protein structure mining. <i>Nucleic Acids Res.</i> , 42(Database issue), D310–D314. https://doi.org/10.1093/nar/gkt1242
BioGRID (https://thebiogrid.org/)	Relational model	An interaction repository with data compiled through comprehensive curation efforts.	Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A. and Tyers, M., 2006. BioGRID: a general repository for interaction datasets. <i>Nucleic acids research</i> , 34(suppl_1), pp. D535–D539.
STRING (https://string-db.org/)	Relational Model	A database of known and predicted protein-protein interactions.	Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, Berend Snel; STRING: a database of predicted functional associations between proteins, <i>Nucleic Acids Research</i> , Volume 31, Issue 1, 1 January 2003, Pages 258–261. https://doi.org/10.1093/nar/gkg034
GreenPhylDB (http://greenphyl.cirad.fr)	Relational model	Platform for comparative functional genomics in <i>Oryza sativa</i> and <i>Arabidopsis thaliana</i> genomes	Conte, M. G., Gaillard, S., Lanau, N., Rouard, M., & Périn, C. (2008). GreenPhylDB: a database for plant comparative genomics. <i>Nucleic Acids Research</i> , 36(Database issue), D991–D998. https://doi.org/10.1093/nar/gkm934
TreeBASE (treebase.org)	Relational model	Database on phylogenetic relationships	Piel, W.H., Chan, L., Dominus, M.J., Ruan, J., Vos, R.A. and Tannen, V., 2009. TreeBASE v. 2: a database of phylogenetic knowledge.
FlyBase (Flybase.org)	Relational	Contain fundamental genomic and genetic data on the major genetic model <i>Drosophila melanogaster</i> and related species	The FlyBase database of the <i>Drosophila</i> genome projects and community literature. (2003). <i>Nucleic Acids Res.</i> , 31(1), 172–175.
WormBase (Wormbase.org)	Flat file (XML)	Database for <i>Caenorhabditis elegans</i> genome and its biology.	Stein, L., Sternberg, P., Durbin, R., Thierry-Mieg, J., & Spieth, J. (2001). WormBase: network access to the genome and biology of <i>Caenorhabditis elegans</i> . <i>Nucleic Acids Research</i> , 29(1), 82–86. https://doi.org/10.1093/nar/29.1.82
RefSeq (https://www.ncbi.nlm.nih.gov/refseq/)	Relational model	Contains sequences, including genomic DNA, transcripts, and proteins	O'Leary, N.A., Wright, M.W., Brister, J.R., Ciufu, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D. and Astashyn, A., 2015. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. <i>Nucleic acids research</i> , 44(D1), pp. D733–D745.
OMIM (omim.org)	Flat file	Database of human genes and genetic disorders	Amberger, J.S., Bocchini, C.A., Schiettecatte, F., Scott, A.F., & Hamosh, A. (2015). OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalogue of human genes and genetic disorders. <i>Nucleic Acids Research</i> , 43(Database issue), D789–D798. https://doi.org/10.1093/nar/gku1205
AceDB (http://www.sanger.ac.uk/science/tools/acedb)	Hierarchical object	A database for handling genome and bioinformatics data	Srinivasan, J., Otto, G.W., Kahlow, U., Geisler, R., & Sommer, R.J. (2004). AppaDB: An AcedB database for the nematode satellite organism <i>Pristionchus pacificus</i> . <i>Nucleic Acids Research</i> , 32(Database issue), D421–D422. https://doi.org/10.1093/nar/gkh057
PRoteomics IDentifications database	Relational model	Database on mass spectrometry (MS) – based proteomics data	Vizcaino, J. A., Csordas, A., del-Toro, N., Dianes, J. A., Griss, J., Lavidas, I., Hermjakob, H. (2016). 2016 update of the PRIDE database and its related tools. <i>Nucleic Acids Research</i> , 44(Database issue), D447–D456. https://doi.org/10.1093/nar/gkv1145

some fraudulent parties. Scientists who may not want to share unpublished data have their valuable data stored in personal computers which are most of the time untapped. These kind of speckled data, both textual and image data, will not allow discovery of new knowledge and its evident that correlated data can give rise to new discoveries in science.

Finally, the development of databases require data to be in the digital form and digitizing biological data is a daunting task. It requires extensive manpower, enough equipments and all of these requires funds. However, this can be overcome if world class biological data centres makes databasing lenient enough for underprivileged scientists.

Future Directions

In line with the advancement of the digital world and high performance computing technology, biological databases have been mushrooming and the number is expected grow consistently. These biological databases have become an integral part of research in many scientific areas, from data extraction, knowledge discovery and biological simulations to advanced analysis.

In line with the exponential increase in size and heterogeneity of biological data, new database models have been proposed, which are more scalable compared to traditional relational databases deployed on single computer system. Recent studies by [Valduriez \(2011\)](#), [Han et al. \(2011\)](#) and [Konishetty et al. \(2012\)](#) have explored the practical implementations of NoSQL databases to help in developing a viable and usable practice in information management. While new biological databases are being introduced every year, it is now important to relook at models and infrastructure deployed in construction of these databases. Older database models such as relational, which have been the core technology behind biological databases, are losing its relevance with regards to handling exponential increase of data. Therefore, traditional database concepts are evolving in order to meet and map the data management requirements of today's environment.

The growing data in genomics, metabolomics, proteomics and metagenomics can be regarded as Big Data if these data sources are harmonised. It is critical to explore back end data integration that are streamlined into front end resilient computer systems for performing seamless transactions, whether for simple search algorithms or high level data science. In order to achieve this vision, these data sources need to be mediated via ontologies and noSQL databases by adopting parallel computing technology and artificial intelligence. In order to facilitate the new paradigm, automated methods need to map relational models (SQL based) into noSQL models which are the essence of big data applications.

NoSQL graph model has recently become very popular as it presents a solution to many of today's challenges such as visualisation of data with complex connections and rich relationships.

Closing Remarks

This article presents a holistic overview of biological databases, with regards to modeling and the architecture of the databases. A comprehensive literature on biological databases is presented with a focus on two case studies which are GeneCards and Cancer Genome Atlas. This literature, however, by no means covers the whole range of biological databases that are currently available. The Nucleic Acids Research Journal features the latest biological databases with a very good coverage on the most recent research in this area (see "Relevant Websites section"). Finally, the future direction of biological databases is discussed focusing on the new concepts in the digital world such as Big Data, Data Science and NoSQL databases which holds the future of biological databases.

Acknowledgement

I would like to thank all my postgraduate students for assisting in my biological database research over the years. The research contribution made by these students ranging from relational models, semantic web, ontologies, image databases, noSQL and graph models as well as medical databases has been overwhelming.

See also: Biological Database Searching. Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics

References

- About uniprot, 2017. Uniprot (online). Available at: <http://www.uniprot.org/> (14 February 2018).
- Abramova, V., Bernardino, J., 2013. NoSQL databases. In: Proceedings of the International C* Conference on Computer Science and Software Engineering - C3S2E '13. ACM Press New York, New York, USA, pp. 14 – 22. Available at: <http://doi.org/10.1145/2494444.2494447>.
- Abu, A., Lim, S.L.H., Sidhu, A.S., Dhillon, S.K., 2013a. Biodiversity image retrieval framework for monogeneans. *Systematics and Biodiversity* 11 (1), 19–33.
- Abu, A., Susan, L.L.H., Sidhu, A.S., Dhillon, S.K., 2013b. Semantic representation of monogenean haptor Bar image annotation. *BMC Bioinformatics* 14 (1), 48.
- Ali, N.M., Khan, H.A., Amy, Y., et al., 2017. Fish Ontology framework for taxonomy-based fish recognition. *PeerJ* 5, e3811.
- Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., Murzin, A.G., 2013. SCOP2 prototype: A new approach to protein structure mining. *Nucleic Acids Research* 42 (D1), D310–D314.
- Angles, R., Gutierrez, C., 2008. Survey of graph database models. *ACM Computing Surveys* 40 (1), 1–39. Available at: <http://doi.org/10.1145/1322432.1322433>.
- Ashish, N., Ambite, J.L., Muslea, M., Turner, J.A., 2010. Neuroscience data integration through mediation: An (F) BIRN case study. *Frontiers in Neuroinformatics* 4, 118.
- Barrett, T., Troup, D.B., Wilhite, S.E., et al., 2006. NCBI GEO: Mining tens of millions of expression profiles – database and tools update. *Nucleic Acids Research* 35 (suppl_1), D760–D765.
- Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L., 2008. GenBank. *Nucleic Acids Research* 36 (Database issue), D25.
- Blankenberg, D., Taylor, J., Schenck, I., et al., 2007. A framework for collaborative analysis of ENCODE data: Making large-scale analyses biologist-friendly. *Genome Research* 17 (6), 960–964. doi:10.1101/gr.5578007.
- Brazma, A., Parkinson, H., Sarkans, U., et al., 2003. ArrayExpress – A public repository for microarray gene expression data at the EBI. *Nucleic Acids Research* 31 (1), 68–71.
- Burley, S.K., Berman, H.M., Kleywegt, G.J., et al., 2017. Protein Data Bank (PDB): The single global macromolecular structure archive. *Protein Crystallography: Methods and Protocols*. 627–641.
- Cancelo, N., 2014. Not Only SQL as a Alternative to Relational Database Systems. 2.
- Caspi, R., Foerster, H., Fulcher, C.A., et al., 2007. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research* 36 (Suppl_1), D623–D631.
- Cerami, E.G., Gross, B.E., Demir, E., et al., 2010. Pathway commons, a web resource for biological pathway data. *Nucleic acids research* 39 (Suppl_1), D685–D690.
- Chatr-Aryamontri, A., Breitkreutz, B.J., Oughtred, R., et al., 2014. The BioGRID interaction database: 2015 update. *Nucleic Acids Research* 43 (D1), D470–D478.

- Coordinators, N.R., 2016. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 44 (Database issue), D7.
- Dreos, R., Ambrosini, G., Groux, R., Cavin Périer, R., Bucher, P., 2016. The eukaryotic promoter database in its 30th year: Focus on non-vertebrate organisms. *Nucleic Acids Research* 45 (D1), D51–D55.
- Elmasri, R., Navathe, S., 2011. *Fundamentals of Database Systems*. Addison-Wesley.
- Ercan, M.Z., 2014. Evaluation of NoSQL databases for EHR systems. In: *Proceedings of the 25th Australasian Conference on Information Systems*, pp. 1–10. Auckland, New Zealand.
- Ferreira, G.D.S., Calil, A., Mello, R.D.S., 2013. On Providing Ddl Support for a Relational Layer over a Document Nosql Database, in: *Proceedings of International Conference on Information Integration and Web-based Applications & Services*. Vienna, Austria: ACM, 125–132.
- Finn, R.D., Attwood, T.K., Babbitt, P.C., 2017a. InterPro in 2017—beyond protein family and domain annotations. *Nucleic Acids Research* 45, D190–D199. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5210578/> (accessed 14.02.18).
- Finn, R.D., Coghill, P., Eberhardt, R.Y., 2017b. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research*. Available at: <http://pfam.xfam.org/> (accessed 14.02.18).
- Gardner, D., Akil, H., Ascoli, G.A., *et al.*, 2008. The neuroscience information framework: A data and knowledge environment or neuroscience. *Neuroinformatics* 6 (3), 49–160.
- Frishkoff, G., Sydes, J., Mueller, K., *et al.*, 2011. Minimal Information for Neural Electromagnetic Ontologies (MINEMO): A standards-compliant method for analysis and integration of event-related potentials (ERP) data. *Standards in Genomic Sciences* 5 (2), 211.
- Goh, C.S., Lan, N., Echols, N., *et al.*, 2003. SPINE 2: A system for collaborative structural proteomics within a federated database framework. *Nucleic Acids Research* 31 (11), 2833–2838.
- Guarino, N., 1995. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies* 43 (5–6), 625–640.
- Hallin, Peter F., Ussery, David W., 2004. CBS Genome Atlas Database: A dynamic storage for bioinformatic results and sequence data. *Bioinformatics* 20 (18), 3682–3686. Available at: <https://doi.org/10.1093/bioinformatics/bth423>.
- Haider, S., Ballester, B., Smedley, D., *et al.*, 2009. BioMart Central Portal—unified access to biological data. *Nucleic Acids Research* 37 (suppl_2), W23–W27.
- Han, J., Kamber, M., Pei, J., 2000. *Data mining: Concepts and Techniques* (the Morgan Kaufmann Series in data management systems). Morgan Kaufmann.
- Han, J., Haihong, E., Le, G., Du, J., 2011. Survey on NoSQL database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, IEEE, pp. 363–366.
- Heath, T., Bizer, C., 2011. *Linked Data: Evolving the Web Into a Global Data Space*. Morgan & Claypool Publishers.
- Huerta-Cepas, J., Capella-Gutierrez, S., Prysacz, L.P., *et al.*, 2010. PhylomeDB v3.0: An expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. *Nucleic Acids Research* 39 (suppl_1), D556–D560.
- Mashima, J., Kodama, Y., Fujisawa, T., *et al.*, 2017. DNA Data Bank of Japan. *Nucleic Acids Research* 45 (D1), D25–D31 <https://doi.org/10.1093/nar/gkw1001>.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Kanz, C., Aldebert, P., Althorpe, N., *et al.*, 2005. The EMBL nucleotide sequence database. *Nucleic Acids Research* 33 (suppl_1), D29–D33.
- Kormeier, B., 2014. Data Warehouses in Bioinformatics. In: Chen, M., Hofestädt, R. (Eds.), *Approaches in Integrative Bioinformatics*. Berlin, Heidelberg: Springer.
- Konishetty, V.K., Kumar, K.A., Voruganti, K., Rao, G.V., 2012. August. Implementation and evaluation of scalable data structure over HBase. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics, ACM*, 1010–1018.
- Mallach, E.G., 2000. *Decision Support and Data Warehouse Systems*. Irwin/McGraw-Hill: Pennsylvania State University.
- MongoDB, 2018. MongoDB. Retrieved from NoSQL Databases Explained: <https://www.mongodb.com/nosql-explained>.
- Muiliu, J., Peltonen, L., Liton, J.-E., 2007. The federated database – A basis for biobankbasedpost-Genome studies, integrating phenome and genome data from 600 000 twin pairs in Europe. *European Journal of Human Genetics*. 1–6.
- Neo4j, 2018. Neo4j. Retrieved from Neo4j: <https://neo4j.com/developer/graph-database/>.
- Paramasivam, V., Yee, T.S., Dhilon, S.K., Sidhu, A.S., 2014. A methodological review of data mining techniques in predictive medicine: An application in hemodynamic prediction for abdominal aortic aneurysm disease. *Biocybernetics and Biomedical Engineering* 34 (3), 139–145.
- Park, Y.M., Squizzato, S., Buso, N., Gur, T., Lopez, R., 2017. The EBI search engine: EBI search as a service – making biological data accessible for all. *Nucleic Acids Research* 2, 2017 <https://doi.org/10.1093/nar/gkx359>.
- Pendse, N., 2008. What is OLAP? An Analysis of What the Often Misused OLAP Term is Supposed to Mean. Retrieved February 8, 2013, from www.olapreport.com/fasmi.htm.
- Prasad, T.S.K., 2009. Human Protein Reference Database - 2009 Update. *Nucleic Acids Research* 37, D767–D772. [PubMed].
- Pruitt, K.D., Tatusova, T., Maglott, D.R., 2006. NCBI reference sequences (RefSeq): A curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research* 35 (suppl_1), D61–D65.
- Prosite introduction, 2017. PROSITE (online). Available at: <https://prosite.expasy.org/> (accessed 14.02.18).
- Ranganathan, S., Tammi, M., Gribskov, M., Tan, T.W., 2006. Establishing bioinformatics research in the Asia Pacific. *BMC Bioinformatics* 7 (Suppl 5), S1. Available at: <http://doi.org/10.1186/1471-2105-7-S5-S1>.
- Saltz, J., Oster, S., Hastings, S., *et al.*, 2006. caGrid: Design and implementation of the core architecture of the cancer biomedical informatics grid. *Bioinformatics* 22 (15), 1910–1916.
- Safran, M., Dalah, I., Alexander, J., *et al.*, 2010. GeneCards Version 3: The human gene integrator. Database 2010.
- Sanderson, M.J., Donoghue, M.J., Piel, W.H., Eriksson, T., 1994. TreeBASE: A prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. *American Journal of Botany* 81 (6), 183.
- Schomburg, I., Chang, A., Hofmann, O., *et al.*, 2002. BRENDA: A resource for enzyme data and metabolic information. *Trends Biochemical Sciences* 27 (1), 54–56.
- Segaran, T., Hammerbacher, J., 2009a. *Beautiful data*, first ed. Canada: O'Reilly, p. 112.
- Segaran, T., Hammerbacher, J., 2009b. *Beautiful data*, first ed. Canada: O'Reilly, p. 184.
- Severin, J., Waterhouse, A.M., Kawaji, H., *et al.*, 2009. FANTOM4 EdgeExpressDB: An integrated database of promoters, genes, microRNAs, expression dynamics and regulatory interactions. *Genome Biology* 10 (4), R39.
- Sherlock, G., Hernandez-Boussard, T., Kasarskis, A., *et al.*, 2001. The stanford microarray database. *Nucleic Acids Research* 29 (1), 152–155.
- Sillitoe, I., Lewis, T.E., Cuff, A., *et al.*, 2014. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research* 43 (D1), D376–D381.
- Stelzer, G., Dalah, I., Stein, T.I., *et al.*, 2011. In-silico human genomics with GeneCards. *Human Genomics* 5 (6), 709.
- Stephens, Z.D., Lee, S.Y., Faghri, F., *et al.*, 2015. Big data: Astronomical or genomic? *PLOS Biology* 13 (7), e1002195. Available at: <https://doi.org/10.1371/journal.pbio.1002195>.
- Stonebraker, M., 2009. SQL Databases vs NoSQL Databases. Retrieved November 15, 2016, from: <http://cacm.acm.org/blogs/blog-cacm/50678-the-nosql-discussion-has-nothing-to-do-with-sql/fulltext>.
- Sulakhe, D., Xie, B., Taylor, A., *et al.*, 2016. Lynx: A knowledge base and an analytical workbench for integrative medicine. *Nucleic Acids Research* 44 (Database issue), D882–D887. Available at: <http://doi.org/10.1093/nar/gkv1257>.
- Szklarczyk, D., Franceschini, A., Kuhn, M., *et al.*, 2010. The STRING database in 2011: Functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research* 39 (suppl_1), D561–D568.
- Szklarczyk, D., Franceschini, A., Wyder, S., *et al.*, 2014. STRING v10: Protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Research* 43 (D1), D447–D452.
- The National Genome Atlas, 2017. National Cancer Institute, Available at: <https://cancergenome.nih.gov/abouttcga> (accessed 26.02.18).

- National Cancer Institute, 2017. (online). Available at <https://www.cancer.gov/> [14 February 2018].
- Thessen, A.E., Parr, C.S., 2014. Knowledge extraction and semantic annotation of text from the encyclopedia of life. PLOS ONE 9 (3), e89550. Available at: <https://doi.org/10.1371/journal.pone.0089550>.
- Thiele, H., Glandorf, J., Hufnagel, P., 2010. Bioinformatics strategies in life sciences: From data processing and data warehousing to biological knowledge extraction. *Journal of Integrative Bioinformatics (JIB)* 7 (1), 32–40.
- Tomczak, K., Czerwińska, P., Wiznerowicz, M., 2015. The Cancer Genome Atlas (TCGA): An immeasurable source of knowledge. *Contemporary Oncology* 19 (1A), A68–A77. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4322527/#CIT0044> (accessed 26.02.18).
- Valduriez, P., 2011. Principles of Distributed Data Management in 2020? In: Hameurlain, A., Liddle, S.W., Schewe, K.D., Zhou, X. (Eds.), *Database and Expert Systems Applications. DEXA 2011. Lecture Notes in Computer Science*, vol 6860. Berlin, Heidelberg: Springer.
- Weizmann Institute of Science, 2018. GeneCards (online). Available at <https://www.genecards.org/> 14 February 2018.
- Wood, P.T., 2012. Query languages for graph databases. *ACM SIGMOD Record* 41 (1), 50. Available at: <http://doi.org/10.1145/2206869.2206879>.
- Xu, Q., Dunbrack, R.L., 2012. Assignment of protein sequences to existing domain and family classification systems: Pfam and the PDB. *Bioinformatics* 28 (21), 2763–2772. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3476341/> (accessed 14.02.18).
- Zou, D., Ma, L., Yu, J., Zhang, Z., 2015. Biological databases for human research. *Genomics, Proteomics & Bioinformatics* 13 (1), 55–63.

Further Reading

Bioinformatics: Converting Data to Knowledge: Workshop Summary, 2000. Washington (DC): National Academies Press (US), Available from: <https://www.ncbi.nlm.nih.gov/books/NBK44936/> (accessed 20.02. 2018).

Relevant Websites

- <https://www.infrafrontier.eu/resources-and-services/access-emma-mouse-resources>
Access to EMMA mouse resources - Infrafrontier.
- <https://www.acedb.org/>
AceDB.
- <https://www.ebi.ac.uk/arrayexpress>
ArrayExpress < EMBL-EBI.
- <http://tcga.lbl.gov/biosig/tcgadownload.do>
Berkeley Cancer Morphometric Data - TCGA - Lawrence Berkeley.
- <http://thebiogrid.org>
BioGRID.
- <http://www.biopax.org/>
BioPAX (on Github).
- <https://bdsc.indiana.edu/>
Bloomington Drosophila Stock Center.
- <https://bmcbioinformatics.biomedcentral.com/>
BMC Bioinformatics.
- www.brainmap.org
brainmap.org.
- <http://cancer.digitalslidearchive.net/>
Cancer Digital Slide Archive.
- <https://cgwb.nci.nih.gov>
Cancer Genome Workbench.
- <http://www.catalogueoflife.org/>
Catalogue of Life.
- <http://www.cathdb.info>
CATH: Protein Structure Classification Database at UCL.
- <https://bioportal.bioontology.org/ontologies/CCO>
Cell Cycle Ontology - Summary.
- <https://www.clinicalgenome.org/data-sharing/clinvar/>
ClinVar - ClinGen.
- <https://academic.oup.com/database>
Database - Oxford Academic.
- <http://www.ddbj.nig.ac.jp/>
DDBJ Center.
- <http://www.diseasesdatabase.com/>
Diseases Database Ver 2.0.
- <http://disease-ontology.org/>
Disease Ontology.
- <http://www.dnachip.org>
DNAchip.
- <https://www.brenda-enzymes.org/index.php>
Enzyme Database - BRENDA.
- <https://www.eol.org>
EOL.org.
- <http://epd.vital-it.ch>
EPD The Eukaryotic Promoter Database.

<http://www.ebi.ac.uk/embl/index.html>
European Nucleotide Archive.

<https://prosite.expasy.org/>
ExPASy - PROSITE.

<http://flybase.org/>
FlyBase Homepage.

<https://confluence.broadinstitute.org/display/GDAC/home>
FIREHOSE.

<http://www.fishbase.org/>
FishBase.

<https://bioportal.bioontology.org/ontologies/FISHO>
Fish Ontology - Summary.

<https://www.gbif.org>
GBIF.org.

<https://www.ncbi.nlm.nih.gov/genbank/>
GenBank Overview.

www.genecards.org
GeneCards - Human Genes.

<http://www.greenphyll.org/cgi-bin/index.cgi>
GreenPhyl v4.

<http://www.geneontology.org/>
Gene Ontology Consortium.

<https://sourceforge.net/projects/gexkb/>
Gene Expression Knowledge Base.

<https://www.ncbi.nlm.nih.gov/gap>
Home - dbGaP - NCBI - NIH.

<https://www.ncbi.nlm.nih.gov/geo>
Home - GEO - NCBI - NIH.

<https://www.ncbi.nlm.nih.gov/medgen/>
Home - MedGen - NCBI - NIH.

<https://www.ncbi.nlm.nih.gov/pubmed/>
Home - PubMed - NCBI - NIH.

<https://cancergenome.nih.gov/>
Home - The Cancer Genome Atlas - Cancer Genome - TCGA - NIH.

<https://www.hprd.org>
HPRD.org.

<https://www.itb.cnr.it/ibd/>
IBDsite - CNR-ITB.

<http://www.findmice.org/>
International Mouse Strain Resource.

<https://www.itis.gov/>
Integrated Taxonomic Information System.

<https://academic.oup.com/journals>
Journals - Oxford Academic.

<http://www.genome.jp/kegg/>
KEGG.

<http://www.genome.jp/kegg/disease/>
KEGG DISEASE Database - GenomeNet.

<http://www.kegg.jp/kegg/kegg2.html>
KEGG - Table of Contents.

<http://liverwiki.hupo.org.cn/>
LiverWiki.

<http://lynx.ci.uchicago.edu>
Lynx.

<https://www.malacards.org>
MalaCards - human disease database.

<https://www.nlm.nih.gov/bsd/pmresources.html>
MEDLINE/PubMed Resources Guide.

<http://www.informatics.jax.org/>
MGI-Mouse Genome Informatics.

<https://phenome.jax.org>
MPD: Mouse Phenome Database.

<http://bioit.fleming.gr/mugen/mde.jsp>
mugen database environment.

<https://bioportal.bioontology.org/>
NCBO BioPortal.

<https://www.ncbi.nlm.nih.gov/pubmedhealth>
NCBI - National Library of Medicine - PubMed Health - NIH.

<https://academic.oup.com/nar>
Nucleic Acids Research - Oxford Academic.

<https://www.omim.org/>
OMIM.

www.pubbrain.org
OpenfMRI.
<https://pfam.xfam.org/>
Pfam: Home page.
<http://phylomedb.org/>
PhylomeDB v4.
<http://proteomics.yzu.edu/altsplice/>
Plant Alternative Splicing Database.
<http://bioinformatics.yzu.edu/secretomes/plant/index.php>
PlantSecKB.
<https://www.ebi.ac.uk/pride/archive/>
PRIDE Archive.
<https://www.ncbi.nlm.nih.gov/refseq/>
RefSeq: NCBI Reference Sequence Database - NIH.
<http://www.rrrc.us/>
RRRC.
<https://www.yeastgenome.org/>
Saccharomyces Genome Database.
<http://scop2.mrc-lmb.cam.ac.uk/>
SCOP2.
<http://www.sequenceontology.org/>
Sequence Ontology.
<https://www.snomed.org/snomed-ct>
Snomed CT - SNOMED International.
<http://string.embl.de/>
STRING: functional protein association networks.
<https://www.arabidopsis.org/>
TAIR.
<http://bioinformatics.mdanderson.org/tcgabatcheffects>
TCGA Batch Effects Tool.
www.tdwg.org
TDWG.
<http://www.autoimmuneregistry.org/>
The Autoimmune Registry.
<http://www.cancerimagingarchive.internet>
The Cancer Imaging Archive (TCIA) - A growing archive of medical.
<http://ctdbase.org/>
The Comparative Toxicogenomics Database.
<https://treebase.org/>
TreeBASE Web.
www.uniprot.org
UniProt.Org.
<http://www.wormbase.org>
WormBase Home.
<https://www.wwpdb.org>
wwPDB: Worldwide Protein Data Bank.
<http://zebrafish.org>
Zebrafish International Resource Center.
<http://www.zf-health.org/zf-models/>
zf-models - zf-health.

Functional Genomics

Shalini Kaushik, Indian Institute of Technology Roorkee, India

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal and University of Minho, Braga, Portugal

Deepak Sharma, Indian Institute of Technology Roorkee, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Functional genomics is a branch that integrates molecular biology and cell biology studies, and deals with the whole structure, function and regulation of a gene in contrast to the gene-by-gene approach of classical molecular biology technique. It aims to relate the phenotype and genotype on genome level and includes processes such as transcription, translation, protein-protein interaction and epigenetic regulation. This involves comprehensive analysis to understand genes, their functional roles and variable levels of protein expression. The Human Genome Project (HGP) ([Collins et al., 2003](#); [Green et al., 2015](#)) is an integral part of functional genomics. It elucidated that the human genome contains 3164.7 million nucleotide bases with the total number of ~20,000 genes. The size of chromosomes is in between 47 and 250 Mb. Chromosome 1 has a maximum number of genes (5078) and the MT has fewest (37). Functional genomics is characterized by the following distinct research areas:

Genetic Interaction Mapping

Gene Interaction

Gene interactions occur when two or more allelic or non-allelic genes of same genotype influence the outcome of particular phenotypic characters. To understand the molecular basis of this complex biological phenomena, there is a need of genetic interaction mapping where the effects on one gene are modified by one or several other genes. Furthermore, systematic pairwise deletion of genes or inhibition of gene expression can be used to identify genes with related function, even if they do not interact physically. Genetic interaction is the set of functional association between genes. One such relationship is epistasis, which is the interaction of non-allelic genes where the effect of one gene is masked by another gene to result either in the suppression of the effect or they both combine to produce a new trait (character). Mapping genetic interactions by simultaneously perturbing pairs of genes, which report on how genes work together, provides a powerful tool for systematically defining gene function and pathways.

The complex etiology of most human diseases that cannot be explained by single nucleotide polymorphism (SNPs) is thought to be caused by combination of multiple genetic variations. By finding the (i) similarities in sequences, gene order and fusion events, (ii) common structural motifs, or (iii) resemblance in gene expression, the role of uncharacterized proteins can be deduced computationally. Furthermore in gene networks, based on the comparison of neighbors, functionality of related genes can be deciphered ([Schlitt et al., 2003](#)). The statistical epistasis (deviation from additivity in a mathematical model) enabled the identification of an array of different gene interactions that include aggravating or synergistic interactions ([Boucher and Jenna, 2013](#)).

Computational Tools for Gene-Gene Interactions

Till date, almost 100 computational tools are available for epistasis detection (see “Relevant Websites section”). Some of the prominent ones are: (i) BEAM (Bayesian Epistasis Association Mapping), which treats the disease-associated markers and their interactions via a bayesian partitioning model and computes the posterior probability that each marker set is related with the disease by employing Markov chain Monte Carlo ([Zhang and Liu, 2007](#)) (ii) TEAM (Tree-Based Epistasis Association Mapping), for fast detection of gene-gene interactions without ignoring any epistatic interaction in genome-wide association study (GWAS) of humans ([Zhang et al., 2010a](#)), (iii) BOOST (BOolean Operation-based Screening and Testing), a computationally and statistically powerful tool for the detection of unrevealed gene-gene interactions, which is the basis of many complex diseases ([Wan et al., 2010](#)), and (iv) TS-GSIS (Two Stage-Grouped Sure Independence Screening), for studying SNP-SNP interactions with/without marginal effects as well as making direct inference and easy interpretation on the biologically important genes ([Table 1](#); [Fang et al., 2017](#)).

Single Nucleotide Polymorphism

A single-nucleotide polymorphism (SNP) is a variation in a single nucleotide that occurs at a specific position in the genome. These genetic variations in different individuals are the cause of differences in human genomes and are at the forefront of many disease-gene association studies. Based on SNPs, there are many techniques [for instance, Restriction Fragment Length Polymorphism (RFLP)] used for determining suspected individuals of crime such as DNA fingerprinting. Microsatellites or simple sequence repeats (SSRs) and minisatellites are abundant in whole genome and show polymorphism at a high level. SSRs are

Table 1 Tools implicated in gene interaction mapping, SNP and gene regulation

Research areas	Website	Key feature(s)	CI ^a	Access ^b	URL
Gene interaction mapping	BEAM	Investigates disease-associated markers and their interactions	38.1	D	https://sites.fas.harvard.edu/~junliu/BEAM/
	BOOST	Discovers unknown gene-gene interactions	42.1	D	http://snpboost.sourceforge.net
	TEAM	Identifies genetic markers underlying phenotypic studies	17.1	W,D	http://csbio.unc.edu/epistasis/client-team2.php
	TS-GSIS	Detects SNP-SNP interactions and their whole genome effects	— ^c	D	https://cran.r-project.org/web/packages/TSGSIS/index.html
SNP	kSNP3.0	Identifies SNPs and phylogenetic analysis of genomes	26.4	D	https://sourceforge.net/projects/ksnp/files/
Gene regulation	SP2TFBS	Collects specific annotation of human SNPs	6.0	W,D	https://ccg.vital-it.ch/snp2tfbs/
	AlignACE	Finds conserved sequence elements and identifies motifs in a set of DNA sequences	66.9	W,D	http://arep.med.harvard.edu/mrnadata/mrnasoft.html
	Contra	Identifies TFBSs	6.3	W	http://bioit.dmbr.ugent.be/ConTra/index.php
	CpGProD	Predicts promoters associated with CpG islands in mammalian genomic sequences	12.2	W,D	http://pbil.univ-lyon1.fr/software/cpgprod.html
	DBTSS	Database of TSSs, majority of human adult and embryonic tissues	15.9	W,D	https://dbtss.hgc.jp/
	JASPAR	Repository of transcription binding factor profiles in eukaryotes	102.0	W	http://jaspar.genereg.net/
	MATCH	Weight matrix-based program used to predict TFBSs	73.9	W	http://www.gene-regulation.com/pub/programs.html#match
	MEME	Portal for discovery and analysis of motifs	380.0	W,D	http://meme-suite.org/tools/meme
	microTSS	Identifies TSSs	9.8	D	https://github.com/geo2mandos/microTSS
	MoPP	Motif discovery tool	1.9	W	http://compbio.iitr.ac.in/reganalyst/mopp/
	MyPatternFinder	Finds matches to a user-defined query motif	1.9	W	http://compbio.iitr.ac.in/reganalyst/mypatternfinder/
	oPOSSUM	Detects over-represented TFBSs in co-expressed genes	28.2	W	http://www.cisreg.ca/oPOSSUM/
	PRODORIC	Organizes information on prokaryotic gene expression and integrate it into regulatory networks	24.4	W,D	http://prodoric.tu-bs.de/
	SCPD	Catalogs TFBSs and TSSs in <i>S. cerevisiae</i>	30.0	W	http://rulai.cshl.edu/SCPD/
	TRANSCompel	Provides data on experimental-proven binding-sites, consensus binding sites and regulated genes	152.5	D	http://gene-regulation.com/pub/databases/transcompel/compelSM.html
	TRANSFAC	Repository of DNA binding sites with their transcription factors	44.1	W	http://genexplain.com/transfac/

^aCI, citation index (number of citations per year).^bW, Web based; D, downloadable; W,D, both Web based and downloadable.^c—, no citations.

distributed both in coding and non-coding DNA and are implicated in diverse biological functions. Protein coding SNPs can be further divided into synonymous and non-synonymous (nsSNPs); synonymous SNPs do not change the amino acid sequence while nsSNPs do.

Tools Implicated in Analysis of SNPs

Several computational tools for the analysis of SNPs have been developed (Table 1; for a more exhaustive list of tools see Karchin, 2009). The algorithm CHOISS (Choosing Optimal-Interval SNP Set) finds a set of SNP markers based on interval regularity (Lee and Kang, 2004). Subsequently, an informative SNP selection method for unphased genotype data based on multiple linear regression (MLR) was implemented in the software package MLR-tagging (He and Zelikovsky, 2006). A neural network-derived method, SNAP (screening for non-acceptable polymorphisms), has also been established that predicts functional effects of nsSNPs by incorporating evolutionary information (residue conservation within sequence families), aspects of protein structure (secondary structure, solvent accessibility) and other relevant information (Bromberg and Rost, 2007). kSNP3.0 is a program for SNP detection and phylogenetic analysis of genomes without genome alignment or reference genome (Gardner et al., 2015). Recently, a database SP2TFBS was developed that catalogs regulatory SNPs affecting predicted transcription factor binding site (TFBS) affinity (Kumar et al., 2017).

Gene Regulation

Promoter Analysis

There are two main promoters, one TATA-box enriched promoter, a well-defined binding site, and another CpG island promoter. Using the Cap Analysis of Gene Expression (CAGE) approach, it is possible to re-design promoter for differential gene expression. One can tag several thousand transcription start sites and quantitatively analyze promoter usage of different tissues with their differential expression in such sites. The identification and analysis of promoters is essential to understand the transcriptional regulatory networks. Several transcriptional start sites databases have been developed (**Table 1**), amongst which many are dedicated to eukaryotes and mammals for instance, DBTSS (Database of Transcription Start Sites), SCPD (*S. cerevisiae* Promoter Database) and microTSS (hosts miRNA transcription start sites). DBTSS covers the data of the majority of human adult and embryonic tissues, and contains 418 million TSS tag sequences from 28 different tissues/cell cultures (Wakaguri *et al.*, 2008). On the other hand, SCPD catalogs experimentally mapped TFBSs and TSSs in yeast *S. cerevisiae* (Zhu and Zhang, 1999). These databases not only identify and retrieve promoters but also provide information on genes with mapped regulatory regions.

Various Regulatory Regions

Different types of regulatory regions are present in our human genome sequence which is a huge source of data of functional genes (Collado-Vides *et al.*, 2009). Different expression of promoter depends on various factors like transcription factor, enhancer, activator, repressor and they define the regulatory region. In addition, there are conserved sequences in human genome that can be employed in phylogenetic footprinting; it is an area of bioinformatics used to identify TFBSs within a non-coding region of DNA by comparing it to the orthologous sequences in different species (Impey *et al.*, 2004; Lenhard *et al.*, 2003). JASPAR is a database for transcription binding factor profiles and it catalogs matrix-based nucleotide profiles describing the binding preference of transcription factors from multiple species (**Table 1**; Sandelin *et al.*, 2004). In a similar vein, TRANSFAC database maintains data on eukaryotic transcription factors, their experimentally proven binding sites, consensus binding sequences and regulated genes (Wingender *et al.*, 1996). Likewise, TRANSCOMP database also contains data on eukaryotic transcription factors experimentally proven to act together in a synergistic or antagonistic manner (Matys *et al.*, 2006). On the other hand, PRODORIC hosts data on environmental stimuli with *trans*-acting transcription factors, *cis*-acting promoter elements and regulon definition (Munch *et al.*, 2003). It includes graphical representations of operon, gene and promoter structures including regulator-binding sites, transactivational start sites and also provides supplemented data on regulatory proteins.

In the vast area of functional genomics, finding the TFBSs on a whole-genome level, is a formidable challenge for gene-regulation studies including, annotation of regulatory sites and interpreting gene regulatory networks. MATCH (Kel *et al.*, 2003), oPOSSUM (Ho Sui *et al.*, 2005) and ConTra (Broos *et al.*, 2011) are some of the notable tools used to find TFBSs.

Pattern Discovery

Pattern identification is another area of bioinformatics that analyzes expression pattern of genes as well as sequence patterns in DNA, RNA or proteins. Additionally, in bioinformatics analysis, feature selection technique (Saeys *et al.*, 2007), also plays an important role in extracting huge data from genomic sequences (Sandelin and Wasserman, 2004).

Bioinformatics Tools Used for Regulation of Genes

The availability of numerous genomic sequences and the huge genome-scale experimental data allows the use of computational techniques to investigate *cis*-acting sequences controlling transcriptional regulation. Various tools have been developed to find new sites for a given transcription factor based on a set of known sites (for instance, MyPatternFinder Sharma *et al.*, 2009) and Fuzznuc (emboss.sourceforge.net/apps/cvs/emboss/apps/fuzznuc.html) as well as to find unknown DNA binding motifs for unspecified transcription factors by searching the regions upstream of the translational start sites of a set of potentially coregulated genes (for instance, MEME Bailey *et al.*, 2009; Tavan *et al.*, 1990) and MoPP (Sharma *et al.*, 2009; **Table 1**). AlignACE is a computational tool that implements Gibbs sampling algorithm for identifying motifs that are present in a set of DNA sequences in *S. cerevisiae* (Hughes *et al.*, 2000).

Microarray Data Analysis

Implications of Microarrays in Functional Genomics

Rapid and systematic study of the expression of many genes can be analyzed by microarray technology. On the solid surface (silicon chip, nylon membrane or glass slide), thousands of tiny spots (oligonucleotides) are immobilized at defined positions. These known oligonucleotides act as probes and permit similarity analysis by binding with corresponding unknown samples via hybridization for the purpose of gene discovery and their function. Thus, microarrays facilitate the finding of absolutely novel and

unexpected functional roles of genes (Slonim and Yanai, 2009). Furthermore, it compares the expression pattern of genes in normal and diseased tissues. Identification of SNP in the related genes can be observed by using guide DNA. Such kind of detection helps in mutation analysis as genes may differ from each other by single nucleotide base. In addition, microarrays are also being employed in comparative genome hybridization that permits the detection of altered chromosomal copy number and these alterations can be measured by hybridizing fluorescently labeled DNA from a diseased specimen to the microarray plate.

Applications of Microarrays

DNA microarrays are useful in the identification of various diseases like cancer, in the field of pharmacogenomics, in the discovery of drugs or development of more effective drugs for specific treatment strategies. Comparative study of biochemical composition of the gene product that is responsible for the disease with the proteins formed by the normal genes, helps researchers to synthesize drugs that can minimize the effect of such defective proteins. DNA microarrays help in gene discovery, their expression profiling, their regulation, genetic disease screening, SNP discrimination, microbial detection and typing, and observe variation between transcript levels of genes under different conditions. It also helps in toxicological research by tracking the modifications in the genetic profiles of the cells when exposed to certain toxicants. Additionally, microarray technology is useful in clinical microbiology as it is a potential tool for bacterial recognition and for diagnosis of bloodstream infections caused by certain pathogens. Another application of microarray technique is the determination of antimicrobial drug resistance by detecting the mutations due to drug resistance in the genome of microbes (Miller and Tang, 2009).

Computational Prospective for Microarray Technology

Various bioinformatics tools and programs are available for microarray data analysis and mining (Selvaraj and Natarajan, 2011). Computational analysis of microarray data becomes more important to address the problems of data quality, to handle the large amount of microarray data and standardization of this technology. For this, data mining needs to be carried out which includes clustering and classification. Clustering, as name suggests, is to categorize data into groups of related genes (genes with similar characteristics falls into single group). It is an unsupervised learning approach of classifying data into clusters of related observations and is based on the algorithm of K-means clustering and Self-Organizing Map (SOM) (Tavan *et al.*, 1990). On the other hand, classification is a supervised learning approach which predicts the class of the unknown gene by observing the pre-classified examples and is based on the algorithm such as Artificial Neural Network (ANN), k-Nearest Neighbors (kNN) and Support Vector Machines (SVM). One of the main application of classification is the prediction of disease type. Prominent tools available for the clustering analysis include MAGIC Tools (Heyer *et al.*, 2005) and dChip (Amin *et al.*, 2011) and for classification purpose include LIBSVM (see "Relevant Websites section") (Table 2).

High-Throughput GoMiner provides outcomes across multiple microarrays one at a time, for the broad scale of applications such as various drug treatments, time-courses evaluation, collective gene comparison (Zeeberg *et al.*, 2005). It organizes the list of under- and over-expressed genes, particularly those genes that have some biological importance in Gene Ontology. iArray (Simon *et al.*, 2007), EDGE (Leek *et al.*, 2006) and Cyber-T (Kayala and Baldi, 2012) are some of the programs available for gene expression analysis. Such studies are important for finding potential drug targets and biomarkers.

SAGE (Serial Analysis of Gene Expression)

Unlike microarray technology, where hybridization is the principle for analysing expression of genes, SAGE depends on RNA sequencing for global analysis of gene expression in a cell. The main benefit of using this technique over others is that it does not require any prior understanding of transcripts and it provides a clear illustration of both qualitative study by comparing gene expression between samples and quantitative study by identifying novel transcripts. The method involves the isolation of short (9–11 base pairs) oligonucleotides sequence tag from discrete mRNAs. These short distinctive sequence tags (SAGE tags) are unique to each gene, allowed to concatenate for efficient sequencing and analysis of transcripts in a serial fashion. The resulting sequence data are analyzed to identify each gene expressed in cell and the level of gene expression. This information can further be used to test the differences in the level of expression of gene between cells.

SAGE Bioinformatics Packages

The role of SAGE tools is very clear as we need them for the isolation of the unique sequence tags from raw sequence files and their tabulation as well as comparison of SAGE tag abundance. Furthermore, SAGE software helps in matching tags to reference sequences in other databases.

Various bioinformatics methods have been developed for compiling and analysing the SAGE data (Table 2; for a detailed list of tools see Tuteja and Tuteja, 2004). In order to identify the SAGE tags, SAGE300 (see "Relevant Websites section") compiles the database of tags extracted from human sequences in Genbank. In contrast, POWER_SAGE software is a useful tool for planning SAGE experiments and it has the capability of handling large number of transcripts/cDNAs and different sample size combinations

Table 2 Tools implicated in microarray, SAGE and mutation analysis as well as other prominent tools used in functional genomics

Research areas	Website	Key feature(s)	CI ^a	Access ^b	URL
Microarray	Cyber-T	Analysis of high-throughput data particularly from microarrays, NGS and mass spectroscopy	19.0	D	http://cybert.microarray.ics.uci.edu/
	dChip	For unsupervised sample clustering and identification of novel cluster and corresponding genes	— ^c	D	https://sites.google.com/site/dchipsoft/
	EDGE	For differential gene expression analysis	19.6	D	http://www.genomine.org/edge/
	High-Throughput GoMiner	Organizes lists of under- and over-expressed genes for biological interpretation in context to Gene Ontology	22.1	W,D	https://discover.nci.nih.gov/gominer/htgm.jsp
	iArray	Use meta-analysis of multiple microarray datasets	—	D	http://zhoulab.usc.edu/iArrayAnalyzer.htm
	LIBSVM	For classifying groups based on Support Vector Machine	567.4	D	http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/
SAGE	MAGIC	Clustering tool to analyze all gene-expression data	3.6	D	http://www.bio.davidson.edu/MAGIC/
	SAGEmap	Public gene expression data repository	27.0	W	https://www.ncbi.nlm.nih.gov/projects/SAGE/
Mutation	SAGEnet	For analysing gene expression patterns	—	W	http://www.sagenet.org/
	Cas-OFFinder	Algorithm for searching potential off- target sites of cas9 RNA-guided endonuclease	69.3	D	http://www.rgenome.net/cas-offinder/
	COSMIC	Catalogue of somatic mutations in cancer	46.7	W	https://cancer.sanger.ac.uk/cosmic
	HGMD	Collates all known gene lesions responsible for human inherited diseases	364.9	W	http://www.hgmd.cf.ac.uk/ac/index.php
	Mudi	Mutation detecting tool in different organisms from whole genome sequence data	1.3	D	http://naoii.nig.ac.jp/mudi_top.html
	MUFFINN	For cancer gene discovery via network analysis of somatic mutation data	11.4	W,D	http://www.inetbio.org/muffinn/search.php
	PANTHER	Detects amino acid substitutions using Hidden Markov Model algorithm	205.8	W	http://pantherdb.org/
	Polyphen-2	Predicts the adverse effects of a missense mutation on protein stability	164.7	W	http://genetics.bwh.harvard.edu/pph2
	PROVEAN	A tool that predicts non-synonymous variations	91.5	W	http://provean.jcvi.org/index.php
	SIFT	Predicts the occurrence of non-synonymous polymorphisms or laboratory-induced missense mutations	207.9	W	http://sift.jcvi.org/
General tools	SNAP2	For identifying functional effects of mutation using neural network algorithm	49.6	W	https://www.rostlab.org/services/snap/
	COMPLEAT	Analysing high throughput data, particularly for analysis of network dynamics	42.1	D	https://fgr.hms.harvard.edu/compleat
	DAVID	For functional classification and annotation of genes	450.3	W,D	https://david.ncifcrf.gov/
	GSEA	For identifying over-expressed genes/proteins among large set of data	22.0	D	http://software.broadinstitute.org/gsea/index.jsp

^aCI, citation index (number of citations per year).^bW, Web based; D, downloadable; W,D, both Web based and downloadable.^c—, no citations.

(Man *et al.*, 2000). Expression Profile Viewer (ExProView) is a tool for the analysis of gene expression profiles derived from expressed sequence tags and SAGE (Larsson *et al.*, 2000). It visualizes the transcript data in a two-dimensional array of dots, in which each dot represents a known gene as specified in the transcript database. In addition, SAGENet (see “Relevant Websites section”) catalogs tags for colon cancer, pancreatic cancer and corresponding normal tissues while SAGEmap is a public database for gene expression (Lash *et al.*, 2000).

Mutations

A mutation is an alteration in the primary sequence of a gene. Mutations may lead to an alteration in the structural and/or functional properties of proteins and result in some of the deadly diseases. Mutations can be categorized into two types (i) spontaneous and (ii) induced mutation. The spontaneous mutations involve deamination, depurination, tautomerism and slipped strand mispairing (Drake *et al.*, 1998). On the other hand, induced mutations occur due to the effects induced by chemicals such as hydroxylamine,

oxidative radicals, DNA intercalating agents (e.g., ethidium bromide), alkylating agents, base analogs and nitrous acid. Some other physical factors that have been known to induce mutations include ultraviolet light, ionizing radiation and DNA crosslinkers. The scale of mutations may be small or large, and may alter function of some of the essential proteins. Mutations may be classified as (i) Insertions (addition of DNA bases); (ii) Deletions (deletion of DNA bases); (iii) Duplications (a DNA segment is copied one or more times); (iv) Frameshift mutations (addition/loss of bases that shifts the reading frame of the codons); (v) Nonsense mutations (a sense codon changes to a premature stop codon that results in shorter, unfinished proteins); and (vi) Missense mutations (nucleotide substitutions that result in a different codon which codes for different amino acid). Even small mutations could lead to a defective phenotype. However, such unexpected consequences due to mutations are normally kept in check by an error-prone repair system of our highly developed replication system.

Mutation studies have been applied in pharmacological investigations for carrying out translational work in functional genomics. Mutations in a specific gene are detected using TILLING and AMES tests. With the advent of advanced bioinformatics techniques, it has been possible to analyze the mutations in various diseases like colon cancer and thus indicating a positive role of microsatellites in mutagenesis (Brinkmann *et al.*, 1998). Hereditary nonpolyposis colorectal cancer (HNPCC) is thought to increase the risk of developing colon cancer. Bioinformatics approaches have been employed to investigate the role of nsSNPs in HNPCC genes, by identifying functional SNPs and by exploring phenotypic variation due to genetic mutations (Doss and Sethumadhavan, 2009). These bioinformatic investigations have led to the study of various HNPCC related genes, pathways, diseases and post-translational modifications.

Bioinformatics in Mutational Studies

In recent years, bioinformatics studies have provided a new direction in mutational studies. Monte Carlo simulation methods deal with stochastic models in biology that are interactable mathematically. These methods have been used in the incorporation of mutation and natural selection into Wright-Fisher gametic sampling model (Mode and Gallop, 2008). It helps to predict an infinite model of various mutations, gene conversions, migration among subpopulations and allows recombination. In addition, Human Gene Mutation Database (HGMD) is a collection of germline mutations associated with human inherited diseases that contains 141,000 different lesions detected in over 5700 different genes, with new mutation entries (Table 2; Stenson *et al.*, 2014).

In vitro mutational study is a difficult and time taking process. However, the structural and functional study of mutational genes are possible through computational tools. For instance, there are various software tools available for driver mutation prioritization (see "Relevant Websites section"). The most notable tools among them are mentioned in this section (Table 2). The effects of missense mutations on the structure of a protein product can be analyzed even if its 3D (3 dimensional) structure is not available. The 3D structure of a protein can be deduced by using the primary sequence through homology modeling or other approaches like threading. The analysis of the effects of mutations on the structural stability of a protein can also be carried out. Investigations of steric, anomeric carbon, chiral center and stereochemical consequences of substitutions, provide insights on the molecular fit using structural studies. ProDrg (van Aalten *et al.*, 1996) and PyMOL (Lill and Danielson, 2011) are tools used to observe the effects of mutations on the 3D structure and also assist in carrying out computer-aided drug designing. The effects of missense mutations have been analyzed on the structure and function of p53, a tumor suppressor gene (Kato *et al.*, 2003). CRISPR (Type II Clustered Regularly Interspaced Short Palindromic Repeats)/Cas system helps in adaptive immune response by protecting the host cells against pathogens. Cas-OFFinder algorithm has been developed to search for potential off-target sites in a given genome or user-defined sequences (Bae *et al.*, 2014).

Over 4 million coding mutations have been identified across all human cancer disease types. COSMIC (Catalogue Of Somatic Mutations In Cancer database) describes 10 million non-coding mutations, 1 million copy-number aberrations, 9 million gene-expression variants, and almost 8 million differentially methylated CpGs (Bamford *et al.*, 2004). Mutated protein sequence analysis can also lead to evolutionary relationships. Evolutionary trees are constructed for prediction of orthologs, xenologs and paralogs using Hidden Markov Model (HMM) algorithm. PANTHER (Protein ANALYSIS THrough Evolutionary Relationships) provides a single nucleotide polymorphism (SNP) scoring tool that uses multiple sequence alignments to identify amino acid substitutions (Mi *et al.*, 2016). Similarly, SIFT (Sorting Intolerant From Tolerant) predicts the occurrence of non-synonymous polymorphisms or laboratory-induced missense mutations based on PSI-BLAST (Ng and Henikoff, 2003). On the other hand, Mudi identifies the causative mutations in the whole-genome sequence data and performs mapping, annotation, prioritization and detection of variant alleles (Iida *et al.*, 2014).

Cancer is caused by DNA sequence abnormalities. A major challenge in cancer is to detect a defected gene via somatic mutation profiling. The Cancer Genome Atlas [TCGA] (Cancer Genome Atlas Research *et al.*, 2013) and the International Cancer Consortium [ICGC] (Zhang *et al.*, 2011) are two comprehensive and coordinated projects that deal with systematic profiling of genome alterations in many cancer types. MUFFINN (Mutations for Functional Impact on Network Neighbors) introduced an intuitive, frequency-based approach to quantify the significance of the mutation frequency of each gene or a genomic region compared with a background mutation rate [BMR] (Cho *et al.*, 2016). MUFFINN prioritizes genes based on four ways: (i) the mutation frequency of each gene as well as that of its neighbors in a functional network; (ii) mutation frequency of direct neighbor genes of maximum mutation frequency; (iii) mutation frequency of all direct neighbors with normalization by the number of their network neighbors; and (iv) mutational frequency of all genes of the entire network with diffused weight through the network. On the other hand, Polyphen-2 is a tool implemented in R language to predict the

adverse effects of a missense mutation on the protein stability. It uses eight sequence-based and three structure-based predictive features, which are selected automatically by an iterative greedy algorithm (Adzhubei *et al.*, 2013). The prediction pipeline is characterized by comparing two properties like wild-type allele (normal) corresponding to mutant allele (disease). Likewise, PROVEAN webserver also predicts functional effects of amino acid substitution using alignment-based approach (Choi and Chan, 2015).

miRNAs/Transposons

Genomics, Biogenesis and Mechanism of Regulatory RNA

MicroRNAs (miRNAs) are approximately 22 nucleotides long double stranded endogenous non-coding RNAs. miRNAs play an important role in the suppression of target mRNAs (Krol *et al.*, 2010). For instance, Lin-4 is the predominant regulatory RNA that controls the larval development in *C. elegans*. It is now recognized as the founding member of a large class of miRNAs (Bracht *et al.*, 2010). Two RNase III enzymes, named as DORSHA and DICER, cleave primary mRNA and hence, release a hairpin loop in the nucleus (Han *et al.*, 2004). This mRNA is then exported to cytoplasm and cleaved by DICER that results in mature miRNA duplex. The resulted mature miRNA duplex again gets processed by DICER protein and finally two strands act as guide RNA and passenger RNA. The function of guide RNA strand is to cleave the target mRNA specifically, in order to repress its function. On the other hand, passenger mRNA strand acts as an helper for guide RNA to specifically bind to the target mRNA. The pathway of short-interfering RNA (siRNA) or silencing RNA is similar to miRNA, except that the maturation of siRNA is in the cytosol rather than in the nucleus (Bartel, 2004; He and Hannon, 2004).

Computational Approaches for Studying miRNAs

Experimentally, the functional study of miRNA is a time-consuming process. Hence, various computational approaches have been developed for the study of orthologous and paralogous gene identification of miRNAs. Furthermore, several bioinformatics tools have been developed to manage the exponential growth in the generation of miRNA-related data. The main goals of currently available tools are (i) miRNA expression analysis, (ii) miRNA target prediction, (iii) miRNA detection, (iv) studying miRNA in the

Table 3 Various resources and tools for miRNAs/Transposons

Website	Key feature(s)	CI ^a	Access ^b	URL
BUFET	Boosting the unbiased miRNA functional enrichment analysis using bitsets	— ^c	D	https://github.com/diwis/BUFET/
DASHR	Database of small non-coding RNAs in humans	8.0	W,D	http://lisanwanglab.org/DASHR/smdb.php
deepBase	Identification, expression, evolution and function of small non-coding RNAs	25.7	D	http://rna.sysu.edu.cn/deepBase/
MatureBayes	Algorithm for identifying the mature miRNA within novel precursors	8.2	D	http://mirna.imbb.forth.gr/MatureBayes.html
microRNA.org	For miRNA target prediction as well as hosts expression profiles	187.7	W	www.microrna.org/
miRBase	Repository of miRNA sequences, targets and gene nomenclature	323.2	W	http://www.mirbase.org/
mirDB	For miRNA target prediction and functional annotation	112.7	W,D	http://mirdb.wustl.edu/
miRmine	Database of human miRNA expression profiles	19.2	W	http://guanlab.ccmb.med.umich.edu/mirmine/
miR-PREFeR	Prediction of plant miRNAs using small RNA-Seq data	9.0	D	https://github.com/hangelwen/miR-PREFeR
miRsig	For identification of pan-cancer miRNA-miRNA interaction signatures	3.9	W	http://bnet.egr.vcu.edu/miRsig/
Repbse	Database for repetitive DNA elements	85.0	W,D	http://www.girinst.org/repbse/
RetroSeq	For transposable element discovery from NGS data	16.3	W	https://github.com/tk2/RetroSeq
siDirect	Target-specific siRNA design software that avoids off-target silencing	11.1	W,D	http://sidirect2.nai.jp/
SRF	Identification of all types repetitive sequences, including transposons	10.2	W	http://compbio.iitr.ac.in/srf/
Target-align	Tool for miRNA target identification	10.1	W	http://www.leonxie.com/targetAlign.php
tasiRNAdb	Resource for known ta-siRNA regulatory pathways in plants	5.2	W	http://bioinfo.jit.edu.cn/tasiRNADatabase/
YM500	RNA sequencing database for small RNAs in human and mice	8.2	W	http://driverdb.tms.cmu.edu.tw/ym500v3/index.php

^aCI, citation index (number of citations per year).

^bW, Web based; D, downloadable; W,D, both Web based and downloadable.

^c—, no citations.

context of metabolic and signaling pathways, (v) analysis of interplay between miRNA and transcription factors, (vi) study miRNA in therapeutics, and (vii) identification of miRNA regulatory networks (Akhtar *et al.*, 2016). Here, we discuss some of the major bioinformatics resources and tools that deal with these aspects of miRNA research. High throughput sequencing of mRNA is possible through RNA sequencing. On the basis of small-RNAseq data and expression values, several databases such as miRbase, miRDB, microRNA.org, YM500, DASHR and deepBase have been developed (Table 3). These databases help to predict the functional annotations, target predictions and expression profiles of various miRNAs.

miRBase is an online repository of miRNA sequences and annotations (Griffiths-Jones *et al.*, 2008). It has more than 28,645 entries representing hairpin miRNAs that expressed 35,828 mature miRNA products in 223 different species (Moore *et al.*, 2015). The sequence data, nomenclature, annotations and target predictions have been mentioned systematically. In a similar vein, miRDB is a resource for miRNA target predictions and functional annotations, which are known to play a crucial role in understanding various physiological and disease processes (Wong and Wang, 2015). Till date, it has 2588 and 1912 functional miRNAs from humans and mice, respectively. For better prediction of targets, Support Vector Machine (SVM) has been incorporated as MirTarget algorithm in this resource. Likewise, microRNA.org predicts candidate targets using miRanda algorithm and hosts expression profiles to provide functional information. The current version of the database has target sites for 1100 human, 717 mouse, 387 rat, 186 *D. melanogaster*, and 233 *C. elegans* miRNAs. In comparison, YM500 database has about 11,000 cancer-related smRNA and 10,000 RNA-seq datasets (Cheng *et al.*, 2013). It allows visualization of expression data, enables search for existing evidence of the novel miRNAs, analyzes arm switching phenomenon, performs survival analysis of a specific small non-coding RNA (sncRNA) in different cancer types and finds the isomiRs of the known miRNAs in miRBase. DASHR (Database of small human non-coding RNAs) is another repository for human sncRNA genes, precursor and mature sncRNA annotations, sequence, expression levels (Leung *et al.*, 2016). Currently, it stores 187 small-RNA deep sequencing datasets from 42 tissues and cell types for more than 48,000 sncRNA loci, genes and mature sncRNA products. DASHR allows searches using sncRNA gene name or RNA sequence, location of sncRNA loci in a given interval of genomic coordinates, and visualization of expression of sncRNA genes and mature products.

miRmine is a human miRNA expression database developed using 304 high-quality microRNA sequencing (miRNA-seq) datasets from NCBI-SRA (Panwar *et al.*, 2017). The expression profiles for one or more miRNAs from a specific tissue or cell-line, either normal or with disease information can be retrieved easily. In addition, miRsig (miRNA signature) is a web tool developed for prediction and visualization of the common signatures or core sets of miRNA-miRNA interactions for different diseases (Nalluri *et al.*, 2017). It uses six network inference algorithms, generates scores using each of the algorithms and generates disease-specific miRNA-interaction networks using a consensus-based approach. Finally, it makes the graph intersection analysis on multiple diseases to find core interactions. Amongst siRNA (small interfering RNA) based research studies, siDirect has been developed for computing functional and off-target minimized siRNA design for mammalian RNA interference (RNAi). The server accepts an input sequence and returns siRNA candidates that can be used for systematic functional genomics and therapeutic gene silencing (Ui-Tei *et al.*, 2004). Another distinct tool MatureBayes has been established that uses a machine learning approach for the prediction of miRNA genes products (Gkirtzou *et al.*, 2010). It employs Naive Bayes classifier and uses sequence and secondary structural information of miRNA precursors to determine mature miRNA candidates. It can predict the start position of mature miRNA using information from miRNA precursors in humans and mouse. BUFET (Bitset-based Unbiased miRNA Functional Enrichment Tool) is a recent and novel computational approach that helps to discover potential biological functions that would be affected by the differential expression of a group of miRNAs (Zagganas *et al.*, 2017). It implements efficient data structures that lead to a significant reduction in the execution time of the unbiased enrichment analysis. In addition, this tool exploits parallel architecture of multi-core systems to increase its performance.

Plant miRNAs

The plant miRNAs are non-coding regulatory RNAs that perform important roles in the regulation of plant development and physiological processes like growth, development and stress responses. Plant miRNAs have also played a significant role in the phenotypic evolution, for instance, adaptation of plants to life on land (Taylor *et al.*, 2014). Till date, 20 unique *Arabidopsis* miRNAs have been reported; a few are closely related to each other representing 15 distinct miRNA families. Since some could be derived from multiple genomic loci, the 20 miRNAs could represent more than 40 *Arabidopsis* genes (Bartel, 2004).

In both plants and animals, miRNAs regulate the expression of the target mRNA, often in the context of regulating developmental events. However, in spite of similarities between them, the repression mechanism of target mRNAs varies between plants and animals. Repression of gene expression in animals is mediated by translational attenuation of 3' UTRs (untranslated regions) of target mRNAs that contains multiple miRNA binding sites. Whereas, in almost all plants, miRNAs regulate the gene expression of their target mRNAs by cleaving at a single site within the coding region (Millar and Waterhouse, 2005). Besides their roles in growth, development and maintenance of genome integrity, sncRNAs are also important components in plant stress responses (Khraiwesh *et al.*, 2012). The expression profiles of most miRNAs involved in plant growth and development are significantly altered during stress (Sunkar *et al.*, 2012). miRNAs in different plant species are modulated differently in abiotic and biotic stresses. Furthermore, miRNA398 (miR398) has been proposed to regulate plant stress responses due to oxidative stress, water deficit, salt stress, abscisic acid stress and ultraviolet stress (Zhu *et al.*, 2011).

A critical step in elucidating miRNA function is identifying potential miRNA target. Target-align is one such prediction tool which uses dynamic programming alignment techniques (Table 3; Xie and Zhang, 2010). Trans-acting siRNAs (ta-siRNAs) play a significant role in regulating gene networks involved in development, metabolism, DNA methylation and stress responses. TasiRNAdb is a

database developed for a comprehensive understanding of ta-siRNA regulatory pathways in plants (Zhang *et al.*, 2014). This database also provides a tool, TasExpAnalysis, for mapping user-submitted RNA and degradome libraries to a ta-siRNAs-producing locus (TAS). In addition, the users can perform TAS cleavage analysis and sRNA phasing analysis within tasiRNAdb. Concurrently, a distinct tool miR-PREFeR (miRNA PREdiction From small RNA-Seq data) has also been developed that uses the expression profiles of miRNAs from small RNA-Seq data samples, for the precise prediction of plant miRNAs (Lei and Sun, 2014).

miRNA in Therapeutics

By regulating gene expression, miRNAs play an important role in various biological processes like cell proliferation, apoptosis, biogenesis, transcription and signal transduction. It is not surprising that the deregulation of miRNA results in many human diseases. This can be illustrated by some examples: (i) miRNA expression pattern in heart disease shows that miR-1, miR-29, miR-30, miR-133, and miR-150 have often been found to be down-regulated while miR-21, miR-125, miR-195, miR-199, and miR-214 are up-regulated with hypertrophy (Shah *et al.*, 2009); (ii) miR-17-92 is overexpressed in lung cancer (Hayashita *et al.*, 2005), and (iii) let-7 has been found to be a suppressor miRNA in tumor cells (Johnson *et al.*, 2005).

The recent advances in application of antisense RNAi (RNA interference) and mRNAs has made it possible to manipulate miRNA regulations and hence show considerable promise as therapeutic agents (Vidal *et al.*, 2005). Decreased level of miR-122 in hepatitis B virus (Jopling *et al.*, 2008), miR-21 upregulation in hepatocellular carcinoma (Xu *et al.*, 2011) and high expression of miR-10b in glioblastoma (Gueussous *et al.*, 2013) are few signatures associated with their respective diseases. Dysregulated miRNAs have been observed in cancer therefore miRNAs have also been used as biomarkers for cancer diagnosis. miRNA studies have great potential in terms of drug discoveries due to their involvement in different biological pathways. Recently, clinical trials have been carried out for miRNAs- and siRNAs-targeted therapeutics (Rupaimoole and Slack, 2017). The *in silico* approaches and potential use of bioinformatics to predict miRNAs and their targets has seen a lot of development especially in the context of diseases (Tombol *et al.*, 2009; Banwait and Bastola, 2015).

Transposons in Functional Genomics

Transposable elements or jumping genes are moderately repeated mobile DNA sequences, interspersed throughout the genomes. These elements are of two types, retrotransposons (type I elements) and transposons (type II elements). They regulate the functioning of a gene and the evolution of genome. Transposase enzyme helps in the mobilization of transposons. Generally eukaryotes contain retrotransposons, but plants have transposons in their genome. Retrotransposons follow a copy and paste mechanism *i.e.*, first DNA is copied to a RNA molecule that is reverse transcribed again into a DNA using the reverse transcriptase enzyme. This reverse transcribed DNA sequence is then inserted at new position into the genome. In contrast, transposons follow cut and paste mechanism that do not involve any RNA intermediate. They can bind to target site in both specific or non-specific manner. Retrotransposons are of two types LTRs (long terminal repeats) and Non-LTRs (non-long terminal repeats). The most abundant Non-LTRs present in mammals are of two types, LINEs (long interspersed nuclear elements) and SINEs (short interspersed nuclear elements). Multiple copies of transgenes and transposons are often integrated within the genome and they are considered to be identical as endogenous sequences. Repetitive sequences and their transposition is very important in the context of therapeutic studies of diseases, like cancer. Several bioinformatics approaches have been developed to understand various aspects of these elements (Table 3). Repbase is a database for transposable elements (TEs) and other types of repeats from eukaryotic genomes (Bao *et al.*, 2015). The current version of the database contains 38,000 sequences of different families or subfamilies of repeats. In addition, RetroSeq is a tool that can be used for the prediction and genotyping of non-reference transposable element (TE) insertions from whole genome sequencing data (Keane *et al.*, 2013). On the other hand, SRF (Spectral Repeat Finder) is a widely used tool for the detection of all the types/classes of repeats, including transposons, within genomic DNA (Sharma *et al.*, 2004).

Epigenomics

Epigenomics (*i.e.*, genome science of epigenetics) provides the functional context of genome sequences and proposes that organization of genes in a particular order on a chromosome and arrangement of these chromosomes in a cell play an important role in deciphering the function of genes. Epigenetics encompasses heritable changes in the DNA structure without changing the genetic code, *i.e.*, change in phenotype without any change in genotype, resulting in chromatin remodeling and consequently affecting transcriptional potential and expression of genes. This is a natural process that takes place regularly in the cell/s of an organism. Epigenetics plays a crucial role in different cellular events, for instance, cell development and differentiation. Several factors, like environment, age, and disease states, may have disruptive effects on a gene's expression pattern. It has been shown that the epigenetic states such as gene silencing can be induced by processes like DNA methylation or by expression of non-coding RNA (Egger *et al.*, 2004; Schubeler, 2015).

Functional anatomy of genome, provided by epigenomics has disclosed various connections among genes and with the nearby non-genic DNA. One such example is beta-globin cluster of human hemoglobin, in which genes are distinctively expressed

throughout the development (Proudfoot *et al.*, 1980). During the development stages of cluster, the genes are arranged in order of their expressions. Another astonishing interaction is found between intra- and inter-chromosomes, mediated by chromatin proteins (van Steensel and Dekker, 2010). In fact, DNA loop structures were also found to be associated with functions that can be illustrated by multiple interleukin genes in Th2 cytokine locus of mouse (Cai *et al.*, 2006). An example of large-scale genomic organization mediated by chromatin is the link between long RNAs, heterochromatin modification and gene activity (Feinberg, 2010). Furthermore, the organization of genome is affected by Large Organized Chromatin K9-modifications (LOCKS) and these large regions provide a vital mechanism for functional genomic organization (Wen *et al.*, 2009).

Epigenetic diseases such as cancer, play an important role in epigenetic pathology. The combination of mutations, structural variations and epigenetic alterations differ between each tumor, making patient-specific diagnosis and treatment strategies necessary (Schweiger *et al.*, 2013). Epigenetic modifications play a critical role in the regulation of transcription, DNA repair and replication. The cause of a disease may be due to the environmentally driven variation in epigenetics, particularly as a substitute for mutational change (Jiang *et al.*, 2004). The risk of a disease might also be due to (i) chromatin remodeling or post transcriptional

Table 4 Tools and databases used in epigenetics

Website	Key feature(s)	CI ^a	Access ^b	URL
4D Genome	Resources of chromatin interaction for structure and function relationship of genome	24.0	W	https://4dgenome.research.chop.edu
Catalogue of Imprinted Genes	Catalogue of imprinted genes and parent-of-origin effects in humans and animals	7.31	W	http://igc.otago.ac.nz/home.html
ChromatinDB	Genome wide histone modifications in <i>S. cerevisiae</i>	3.4	W	http://www.chromdb.org/
CpGCluster	A distance-based algorithm for CpG islands detection	15.8	D	https://github.com/bioinfoUGR/cpgcluster
CpGProD	For identifying CpG islands associated with transcription start sites in mammalian genomes	12.2	W	http://pbil.univ-lyon1.fr/software/cpgprod.html
CR Cistrome	A chip-seq database for histone modification and chromatin regulation in human as well as mouse	20.0	W	http://cistrome.org/cr
DBCAT	Database of CpG islands and various analytical tools for comprehensive methylation profiles in cancer	2.7	W	http://dbcat.cgm.ntu.edu.tw/
DeepCpG	Accurate prediction of single-cell DNA methylation states using deep learning	10.0	W,D	http://deepcpg.readthedocs.io/en/latest/
DiseaseMeth	Repository for DNA methylation in human diseases	10.5	W	http://www.biobigdata.com/diseasemeth
EMDN algorithm	For integrative study of DNA methylation and gene expression	1.3	D	https://github.com/william0701/EMDN
EpiFactors	Provides knowledge about various epigenetic regulators, their complex, targets and products	19.0	W	http://epifactors.autosome.ru/
Gene Imprint	Portal for imprinted genomes of various species	— ^c	W	http://www.geneimprint.com/
HHMD	Comprehensive database for human histone modifications	6.8	W	http://bioinfo.hrbmu.edu.cn/hhmd
Histome	Database of human histone variants, different sites of post-translational modifications and histone modifying enzymes	14.1	W	http://www.actrec.gov.in/histome/
Histone DB	Repository of histone protein sequences, classified by histone types and variants	9.0	W	https://www.ncbi.nlm.nih.gov/projects/HistoneDB2.0
MethBank	Whole-genome single-base-resolution methylomes of gametes and early embryos	3.7	W	http://www.dnamethylome.org/
MethBase	Database of annotated methylomes from the public domain	18.3	W	http://smithlabresearch.org/software/methbase/
MethDB	Resource for DNA methylation data	5.4	W	http://www.methdb.de
MethSMRT	Database hosting 6 mA and 4 mC methylations	2.0	W	http://sysbio.sysu.edu.cn/methsmrt/
MethylomeDB	Genome-wide DNA methylation profiles of brain	9.3	W	http://www.neuroepigenomics.org/methylomedb/
NAB	Program to generate three-dimensional molecular structures	0.1	D	http://casegroup.rutgers.edu/casegr-sh-2.2.html
NGSmethDB	Repository for single-base whole-genome methylome maps for the best-assembled eukaryotic genomes	2.0	W	http://bioinfo2.ugr.es/NGSmethDB/gbrowse/
PathoYeasttract	Provides information about genomic transcriptional regulation in yeasts	6.8	W	http://pathoyeasttract.org/
QSEA	Modeling of genome-wide DNA methylation from sequencing enrichment experiments	2.0	W,D	http://www.bioconductor.org/packages/qsea
WAMIDEX	Database of murine genomic imprinting and differential expression	4.8	W	http://atlas.genetics.kcl.ac.uk/

^aCI, citation index (number of citations per year).

^bW, Web based; D, downloadable; W,D, both Web based and downloadable.

^c—, no citations.

modification of histones (e.g., acetylation, ubiquitinylation or methylation), (ii) binding factors, and (iii) DNA replication, repair and transcription based processes (Dawson and Kouzarides, 2012). The database PathoYeasttract (PATHOgenic YEAsT Search for Translational Regulators and Consensus Tracking) helps in the prediction of the regulatory associations in pathogenic yeasts *Candida albicans* and *C. glabrata* (Table 4; Monteiro *et al.*, 2017). High throughput sequencing technologies provide the comprehensive maps of nucleosome positioning and chromatin conformation (de Wit and de Laat, 2012). In addition, epigenomics is transforming the search for genetic causes of common human diseases on the basis of epigenome anatomy with relevant possible disease list and a new approach to their search (Feinberg, 2010).

During the cell division, DNA exists in highly condensed thread-like structures called chromosomes. The DNA in a eukaryotic chromosome is traditionally bound by two classes of proteins called histones and non-histone proteins. Histones (namely H1, H2A, H2B, H3, H4) are responsible for the most basic level of chromosome organization termed as nucleosomes. Nucleosome exists in a fully super condensed form with the help of histone proteins. However, the histone proteins can be chemically modified during the process of replication and transcription by acetylation, methylation, phosphorylation, sumoylation, ubiquitylation and ADP ribosylation process. These chemical modifications of histones lead to chromosomal rearrangement and consequently help in the regulation of gene expression. Such alterations in DNA organization do not change DNA sequences but still change the expression pattern of a gene and possibly the phenotype of an organism.

DNA Methylation, Genome Imprinting and Paramutation

In mammals, 70%–80% of the cytosines have to be methylated to form 5-methylcytosine within a gene, thereby altering the gene expression. Occasionally, some genes remain unmethylated, for instance, Toll-like receptors in dendritic cells, monocytes and natural killer cells. Various studies have been carried out to decipher how CpG islands remain methylation-free in a globally methylated genome (Deaton and Bird, 2011). Interestingly, in cancer cells, CpG islands are known to contribute to gene silencing. The method BsRADseq (bisulfite-converted restriction site associated DNA sequencing) has the ability to quantify the level of DNA methylation differentiation across multiple individuals (Trucchi *et al.*, 2016). The epigenetic phenomenon that causes genes to be expressed in a parent-of-origin specific manner in daughter cells is termed as ‘genome imprinting’. The expression or suppression of genes in progeny is determined by an epigenetic factor of the parental genome. Indeed, the epigenetic mechanisms that are involved in genomic imprinting and X chromosome inactivation (Barr body) share marked similarities. The imprinted genome of various species are listed in Gene Imprint database (See Relevant Websites Section) (Table 4). In comparison, the Catalogue of Imprinted Genes is a collation of genes and phenotypes for which parent-of-origin effects have been reported (Morison *et al.*, 2001). WAMIDEX (Web Atlas of Murine genomic Imprinting and Differential EXpression) is another useful tool that integrates murine imprinted genes with a genome browser that makes microarray data easily accessible in annotation rich genomic context (Schulz *et al.*, 2008).

Sometimes changes in one allele induce the heritable changes in other allele, for example, the color of corn kernels in maize. This phenomenon is called paramutation. Gene expression is regulated by two ways - *cis* or *trans*. In *cis*-interactions, the binding site is on the same molecule of DNA as the gene(s) to be transcribed, while for *trans*-interactions it is on the other DNA molecule or chromosome. In *trans* interactions, paramutation occurs between homologous DNA sequences on different chromosomes that lead to changes in gene expression that are transmitted to next generation through mitosis and meiosis. Paramutation has been observed mainly between homologous DNA sequences present in both allelic and non-allelic positions.

Computational Tools in Epigenetics

The Nucleic acid builder (NAB) is a molecular mechanics program which is used for generation of three-dimensional molecular structures, for instance, in the library in DNA wrapping study (Table 4; Cheatham *et al.*, 2001). Flexibility is one of the factors involved in DNA binding to nucleosomes, and the salt concentration and DNA sequence affects this flexibility. The NAB package provides an alternative to molecular dynamics in implicit solvent and can be used to produce a picture of the vibrational motions for small nucleic acids. DNA duplexes are usually long due to which they have different open and closed curves. The NAB library can be used to produce simple closed circles with or without supercoiled form, simple nucleosome core fragment model and duplex winding around any open curve. Furthermore, various tools and databases have been associated with methylation, CpG islands, histones and chromatin structures (Table 4).

DNA methylation databases

As discussed above, DNA methylation has an important role in epigenetics. The first database developed was MethDB which provides information about DNA methylation profiles and associated gene expression (Amoreira *et al.*, 2003). The database contains experimentally compiled data of a DNA, where cytosine is converted into 5-methylcytosine through methylation process and then regulates epigenetic process in different genes, cells and tissues. It mainly describes methylation profile, pattern and content. The methylation profile gives positional values of cytosine in a sequence of interests. Subsequently, NGSmethDB and MethBase that are based on DNA methylation in bisulfite sequencing data were developed. NGSmethDB retrieves methylation data derived from next-generation sequencing. Two cytosine methylation contexts (CpG and CAG/CTG) are considered. Through a browser interface, the user can search for methylation states and retrieve methylation values for a set of tissues in a given chromosomal region, or display the methylation of promoters among different tissues (Hackenberg *et al.*, 2011). Whereas, MethBase (a database of annotated

methyomes from the public domain) along with MethPipe (a pipeline for both low and high-level methylome analysis) enable researchers to extract interesting features from methylomes and compare them with those identified in public methylomes (Song *et al.*, 2013). MethylomeDB is the first comprehensive source methylation profile of brain that facilitates the comparative epigenomic investigation, as well as analysis of schizophrenia, Alzheimer disease and depression methylomes (Xin *et al.*, 2012). On the other hand, MethBank is another DNA methylome programming database dedicated to storing, browsing and visualizing the genome-wide single-base nucleotide methylomes of gametes and early embryos at different developmental stages in mouse and zebrafish (Zou *et al.*, 2015). In contrast to other methylation databases, MethSMRT is the first database hosting 6-mA and 4-mC methylations (Ye *et al.*, 2017). Additionally, Epigenetic Module based on Differential Networks (EMDN) algorithm was developed for simultaneous analysis of DNA methylation and gene expression data (Ma *et al.*, 2017). Quantitative Sequence Enrichment Analysis (QSEA) is a comprehensive workflow for predicting exact levels of DNA methylation from sequence enrichment experiments (Lienhard *et al.*, 2017). Since aberrant changes in DNA methylation leads to diseases, DiseaseMeth, a repository for DNA methylation in human diseases, was established (Xiong *et al.*, 2017).

CpG island related databases

DBCAT (database of CpG islands and analytical tools) is a database for investigating comprehensive methylation profiles of DNA in human cancers. The database contains information regarding the human genes and CpG islands while the analytical tools comprise of a CpG island finder, a genome query browser and a methylation microarray analytical tool. The analytical tool identifies genes with methylated regions, provides comparison across microarrays and also displays the information about probe sites, intensities, and transcription binding sites. It is the first methylation analytical tool investigating epigenetic regulation related to a human disease (Kuo *et al.*, 2011). CpGProD (CpG island Promoter Detection) is another tool for identifying CpG island associated with transcription start sites in mammalian genome sequences (Ponger and Mouchiroud, 2002). In contrast, CpGcluster is based on the physical distance between neighboring CpGs on the chromosome and is able to predict directly clusters of CpGs. By assigning a p-value to each of these clusters, the most statistically significant ones are predicted as CpG islands (Hackenberg *et al.*, 2006). Recently, machine learning methods have also been incorporated in this field. The neural networks based DeepCpG method has been formulated to predict methylation states in single cells (Angermueller *et al.*, 2017).

Histone related databases

HistoneDB is a resource for the interactive comparative analysis of histone protein sequences and their implications for chromatin function (Draizen *et al.*, 2016). Over the past decade, several enzymes that catalyze and modify histones have been discovered, enabling investigations of their roles in normal cellular processes and adverse pathological conditions. Histone (The Histone Infobase) is a browsable, manually curated and relational database of 55 human histone proteins, 106 distinct sites of their post-translational modifications and 152 histone-modifying enzymes (Khare *et al.*, 2012). In addition, Human Histone Modification Database (HHMD) focuses on storage of histone modification datasets that can be searched for functional annotations, gene ID, chromosome location and cancer name (Zhang *et al.*, 2010b). EpiFactors, is a curated information-based database to facilitate epigenetic regulators, their complexes, targets and products. EpiFactors contains information on 815 proteins, including 95 histones and protamines, as well as 69 protein complexes that are involved in epigenetic regulation (Medvedeva *et al.*, 2015).

Chromatin related databases

DNA is wrapped around histone proteins to form highly condensed chromatin structure in order to fit into the nucleus of a cell. Post-translational modifications of histone proteins play an important role in gene expression. ChromatinDB is a database for histone modification of *Saccharomyces cerevisiae* (O'Connor and Wyrick, 2007). On the other hand, 4DGenome is a distinct database that hosts chromatin interaction data from five organisms and covers all experimental/computational technologies for detecting chromatin interactions including 3C, 4C-Seq, 5C, ChIA-PET, Hi-C, Capture-C and IM-PET (Teng *et al.*, 2015). In addition, CR Cistrome is an online resource for the linkage of chromatin regulators with histone modifications and datasets collected from ChIP-sequencing (Wang *et al.*, 2014).

General Tools Implicated in Functional Genomics

The Database for Annotation, Visualization and Integrated Discovery (DAVID) is a widely used tool that allows functional classification of genes, discovers functionally related genes, enables conversion of gene/protein identifiers from one type to another as well as explores gene names in a batch (Huang *et al.*, 2007; Dennis *et al.*, 2003; Table 2). On the other hand, GSEA (Gene Set Enrichment Analysis) is Gene Ontology based functional enrichment analysis method that recognizes the over-expressed and statistically significant set of genes or proteins among large set of data (Hung *et al.*, 2012; Subramanian *et al.*, 2005). In addition, a protein Complex Enrichment Analysis Tool (COMPLEAT) has also been developed for analysing high-throughput data, particularly for analysis of network dynamics (Vinayagam *et al.*, 2013).

Conclusion

This article will not only enable researchers to get a comprehensive overview of tools available in the field of functional genomics but will also enable them to select the best tool(s) to carry out their task. Additionally, it will assist the development of new and better resources by highlighting the areas where no (or a few) resources exist.

See also: Cell Modeling and Simulation. Comparative Genomics Analysis. Computational Systems Biology. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Gene Prioritization Tools. Gene Prioritization Using Semantic Similarity. Gene Regulatory Network Review. Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine. Genome Analysis – Identification of Genes Involved in Host-Pathogen Protein-Protein Interaction Networks. Genome Annotation: Perspective From Bacterial Genomes. Genome-Wide Haplotype Association Study. Hidden Markov Models. Integrative Analysis of Multi-Omics Data. Metagenomic Analysis and its Applications. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequence Analysis. Next Generation Sequencing Data Analysis. Ontology in Bioinformatics. Pathway Informatics. Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases. Protein Functional Annotation. Regulation of Gene Expression. Sequence Composition. Single Nucleotide Polymorphism Typing. Standards and Models for Biological Data: FGED and HUPO. Whole Genome Sequencing Analysis

References

- Adzhubei, I., Jordan, D.M., Sunyaev, S.R., 2013. Predicting functional effect of human missense mutations using PolyPhen-2. *Curr. Protoc. Hum. Genet.* 7, 7–20.
- Akhtar, M.M., Micolucci, L., Islam, M.S., Olivieri, F., Procopio, A.D., 2016. Bioinformatic tools for microRNA dissection. *Nucleic Acids Res.* 44, 24–44.
- Amin, S.B., Shah, P.K., Yan, A., *et al.*, 2011. The dChip survival analysis module for microarray data. *BMC Bioinform.* 12, 72.
- Amoreira, C., Hindermann, W., Grunau, C., 2003. An improved version of the DNA Methylation database (MethDB). *Nucleic Acids Res.* 31, 75–77.
- Angermueller, C., Lee, H.J., Reik, W., Stegle, O., 2017. DeepCpG: Accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.* 18, 67.
- Bae, S., Park, J., Kim, J.S., 2014. Cas-Offinder: A fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases. *Bioinformatics* 30, 1473–1475.
- Bailey, T.L., Boden, M., Buske, F.A., *et al.*, 2009. MEME SUITE: Tools for motif discovery and searching. *Nucleic Acids Res.* 37, W202–W208.
- Bamford, S., Dawson, E., Forbes, S., *et al.*, 2004. The COSMIC (Catalogue of Somatic Mutations in Cancer) database and website. *Br. J. Cancer* 91, 355–358.
- Banwait, J.K., Bastola, D.R., 2015. Contribution of bioinformatics prediction in microRNA-based cancer therapeutics. *Adv. Drug Deliv. Rev.* 81, 94–103.
- Bao, W., Kojima, K.K., Kohany, O., 2015. Repbase Update, a database of repetitive elements in eukaryotic genomes. *Mob. DNA* 6, 11.
- Bartel, D.P., 2004. MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell* 116, 281–297.
- Boucher, B., Jenna, S., 2013. Genetic interaction networks: Better understand to better predict. *Front. Genet.* 4, 290.
- Bracht, J.R., V.A.N. Wynsberghe, P.M., Mondol, V., Pasquinelli, A.E., 2010. Regulation of lin-4 miRNA expression, organismal growth and development by a conserved RNA binding protein in *C. elegans*. *Dev. Biol.* 348, 210–221.
- Brinkmann, B., Klintschar, M., Neuhauser, F., Huhne, J., Rolf, B., 1998. Mutation rate in human microsatellites: Influence of the structure and length of the tandem repeat. *Am. J. Hum. Genet.* 62, 1408–1415.
- Bromberg, Y., Rost, B., 2007. SNAP: Predict effect of non-synonymous polymorphisms on function. *Nucleic Acids Res.* 35, 3823–3835.
- Broos, S., Hulpiau, P., Galle, J., *et al.*, 2011. ConTra v2: A tool to identify transcription factor binding sites across species, update 2011. *Nucleic Acids Res.* 39, W74–W78.
- Cai, S., Lee, C.C., Kohwi-Shigematsu, T., 2006. SATB1 packages densely looped, transcriptionally active chromatin for coordinated expression of cytokine genes. *Nat. Genet.* 38, 1278–1288.
- Cancer Genome Atlas Research, N., Weinstein, J.N., Collisson, E.A., *et al.*, 2013. The cancer genome atlas pan-cancer analysis project. *Nat. Genet.* 45, 1113–1120.
- Cheatham 3rd, T.E., Brooks, B.R., Kollman, P.A., 2001. Molecular modeling of nucleic acid structure. *Curr. Protoc. Nucleic Acid Chem.* 7, 7. 5.
- Cheng, W.C., Chung, I.F., Huang, T.S., *et al.*, 2013. YM500: A small RNA sequencing (smRNA-seq) database for microRNA research. *Nucleic Acids Res.* 41, D285–D294.
- Cho, A., Shim, J.E., Kim, E., *et al.*, 2016. MUFFINN: Cancer gene discovery via network analysis of somatic mutation data. *Genome Biol.* 17, 129.
- Choi, Y., Chan, A.P., 2015. PROVEAN web server: A tool to predict the functional effect of amino acid substitutions and indels. *Bioinformatics* 31, 2745–2747.
- Collado-Vides, J., Salgado, H., Morett, E., *et al.*, 2009. Bioinformatics resources for the study of gene regulation in bacteria. *J. Bacteriol.* 191, 23–31.
- Collins, F.S., Morgan, M., Patrinos, A., 2003. The Human Genome Project: Lessons from large-scale biology. *Science* 300, 286–290.
- Dawson, M.A., Kouzarides, T., 2012. Cancer epigenetics: From mechanism to therapy. *Cell* 150, 12–27.
- Deaton, A.M., Bird, A., 2011. CpG islands and the regulation of transcription. *Genes Dev.* 25, 1010–1022.
- Dennis Jr., G., Sherman, B.T., Hosack, D.A., *et al.*, 2003. DAVID: Database for annotation, visualization, and integrated discovery. *Genome Biol.* 4, P3.
- Doss, C.G., Sethumadhavan, R., 2009. Investigation on the role of nsSNPs in HNPCC genes – A bioinformatics approach. *J. Biomed. Sci.* 16, 42.
- Draizen, E.J., Shaytan, A.K., Marino-Ramirez, L., *et al.*, 2016. HistoneDB 2.0: A histone database with variants – An integrated resource to explore histones and their variants. *Database (Oxford)* 2016.
- Drake, J.W., Charlesworth, B., Charlesworth, D., Crow, J.F., 1998. Rates of spontaneous mutation. *Genetics* 148, 1667–1686.
- Egger, G., Liang, G., Aparicio, A., Jones, P.A., 2004. Epigenetics in human disease and prospects for epigenetic therapy. *Nature* 429, 457–463.
- Fang, Y.H., Wang, J.H., Hsiung, C.A., 2017. TSGSIS: A high-dimensional grouped variable selection approach for detection of whole-genome SNP-SNP interactions. *Bioinformatics* 33, 3595–3602.
- Feinberg, A.P., 2010. Epigenomics reveals a functional genome anatomy and a new approach to common disease. *Nat. Biotechnol.* 28, 1049–1052.
- Gardner, S.N., Slezak, T., Hall, B.G., 2015. kSNP3.0: SNP detection and phylogenetic analysis of genomes without genome alignment or reference genome. *Bioinformatics* 31, 2877–2878.
- Gkirtzou, K., Tsamardinos, I., Tsakalides, P., Poirazi, P., 2010. MatureBayes: A probabilistic algorithm for identifying the mature miRNA within novel precursors. *PLOS ONE* 5, e11843.
- Green, E.D., Watson, J.D., Collins, F.S., 2015. Human Genome Project: Twenty-five years of big biology. *Nature* 526, 29–31.
- Griffiths-Jones, S., Saini, H.K., Van Dongen, S., Enright, A.J., 2008. miRBase: Tools for microRNA genomics. *Nucleic Acids Res.* 36, D154–D158.
- Guessous, F., Alvarado-Velez, M., Marcinkiewicz, L., *et al.*, 2013. Oncogenic effects of miR-10b in glioblastoma stem cells. *J. Neurooncol.* 112, 153–163.
- Hackenberg, M., Previti, C., Luque-Escamilla, P.L., *et al.*, 2006. CpGcluster: A distance-based algorithm for CpG-island detection. *BMC Bioinform.* 7, 446.

- Hackenberg, M., Barturen, G., Oliver, J.L., 2011. NGSmethDB: A database for next-generation sequencing single-cytosine-resolution DNA methylation data. *Nucleic Acids Res.* 39, D75–D79.
- Han, J., Lee, Y., Yeom, K.H., *et al.*, 2004. The Drosha-DGCR8 complex in primary microRNA processing. *Genes Dev.* 18, 3016–3027.
- Hayashita, Y., Osada, H., Tatematsu, Y., *et al.*, 2005. A polycistronic microRNA cluster, miR-17-92, is overexpressed in human lung cancers and enhances cell proliferation. *Cancer Res.* 65, 9628–9632.
- He, J., Zelikovsky, A., 2006. MLR-tagging: Informative SNP selection for unphased genotypes based on multiple linear regression. *Bioinformatics* 22, 2558–2561.
- He, L., Hannon, G.J., 2004. MicroRNAs: Small RNAs with a big role in gene regulation. *Nat. Rev. Genet.* 5, 522–531.
- Heyer, L.J., Moskowitz, D.Z., Abele, J.A., *et al.*, 2005. MAGIC tool: Integrated microarray data analysis. *Bioinformatics* 21, 2114–2115.
- Ho Sui, S.J., Mortimer, J.R., Arenillas, D.J., *et al.*, 2005. oPOSSUM: Identification of over-represented transcription factor binding sites in co-expressed genes. *Nucleic Acids Res.* 33, 3154–3164.
- Huang, D.W., Sherman, B.T., Tan, Q., *et al.*, 2007. The DAVID Gene Functional Classification Tool: A novel biological module-centric algorithm to functionally analyze large gene lists. *Genome Biol.* 8, R183.
- Hughes, J.D., Estep, P.W., Tavazoie, S., Church, G.M., 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.* 296, 1205–1214.
- Hung, J.H., Yang, T.H., Hu, Z., Weng, Z., Delisi, C., 2012. Gene set enrichment analysis: Performance evaluation and usage guidelines. *Brief. Bioinform.* 13, 281–291.
- Iida, N., Yamao, F., Nakamura, Y., Iida, T., 2014. Mudi, a web tool for identifying mutations by bioinformatics analysis of whole-genome sequence. *Genes Cells* 19, 517–527.
- Impey, S., Mccorkle, S.R., Cha-Molstad, H., *et al.*, 2004. Defining the CREB regulon: A genome-wide analysis of transcription factor regulatory regions. *Cell* 119, 1041–1054.
- Jiang, Y.H., Bressler, J., Beaudet, A.L., 2004. Epigenetics and human disease. *Annu. Rev. Genom. Hum. Genet.* 5, 479–510.
- Johnson, S.M., Grosshans, H., Shingara, J., *et al.*, 2005. RAS is regulated by the let-7 microRNA family. *Cell* 120, 635–647.
- Jopling, C.L., Schutz, S., Sarnow, P., 2008. Position-dependent function for a tandem microRNA miR-122-binding site located in the hepatitis C virus RNA genome. *Cell Host Microbe* 4, 77–85.
- Karchin, R., 2009. Next generation tools for the annotation of human SNPs. *Brief. Bioinform.* 10, 35–52.
- Kato, S., Han, S.Y., Liu, W., *et al.*, 2003. Understanding the function-structure and function-mutation relationships of p53 tumor suppressor protein by high-resolution missense mutation analysis. *Proc. Natl. Acad. Sci. USA* 100, 8424–8429.
- Kayala, M.A., Baldi, P., 2012. Cyber-T web server: Differential analysis of high-throughput data. *Nucleic Acids Res.* 40, W553–W559.
- Kean, T.M., Wong, K., Adams, D.J., 2013. RetroSeq: Transposable element discovery from next-generation sequencing data. *Bioinformatics* 29, 389–390.
- Kel, A.E., Gossling, E., Reuter, I., *et al.*, 2003. MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.* 31, 3576–3579.
- Khare, S.P., Habib, F., Sharma, R., *et al.*, 2012. Histone – A relational knowledgebase of human histone proteins and histone modifying enzymes. *Nucleic Acids Res.* 40, D337–D342.
- Khraiwesh, B., Zhu, J.K., Zhu, J., 2012. Role of miRNAs and siRNAs in biotic and abiotic stress responses of plants. *Biochim. Biophys. Acta* 1819, 137–148.
- Krol, J., Loedige, I., Filipowicz, W., 2010. The widespread regulation of microRNA biogenesis, function and decay. *Nat. Rev. Genet.* 11, 597–610.
- Kumar, S., Ambrosini, G., Bucher, P., 2017. SNP2TFBS – A database of regulatory SNPs affecting predicted transcription factor binding site affinity. *Nucleic Acids Res.* 45, D139–D144.
- Kuo, H.C., Lin, P.Y., Chung, T.C., *et al.*, 2011. DBCAT: Database of CpG islands and analytical tools for identifying comprehensive methylation profiles in cancer cells. *J. Comput. Biol.* 18, 1013–1017.
- Larsson, M., Stahl, S., Uhlen, M., Wennberg, A., 2000. Expression profile viewer (ExProView): A software tool for transcriptome analysis. *Genomics* 63, 341–353.
- Lash, A.E., Tolstoshev, C.M., Wagner, L., *et al.*, 2000. SAGEmap: A public gene expression resource. *Genome Res.* 10, 1051–1060.
- Lee, S., Kang, C., 2004. CHOIS for selection of single nucleotide polymorphism markers on interval regularity. *Bioinformatics* 20, 581–582.
- Leek, J.T., Monsen, E., Dabney, A.R., Storey, J.D., 2006. EDGE: Extraction and analysis of differential gene expression. *Bioinformatics* 22, 507–508.
- Lei, J., Sun, Y., 2014. miR-PREFeR: An accurate, fast and easy-to-use plant miRNA prediction tool using small RNA-Seq data. *Bioinformatics* 30, 2837–2839.
- Lenhard, B., Sandelin, A., Mendoza, L., *et al.*, 2003. Identification of conserved regulatory elements by comparative genome analysis. *J. Biol.* 2, 13.
- Leung, Y.Y., Kuksa, P.P., Amlie-Wolf, A., *et al.*, 2016. DASHR: Database of small human noncoding RNAs. *Nucleic Acids Res.* 44, D216–D222.
- Lienhard, M., Grasse, S., Rolff, J., *et al.*, 2017. QSEA-modelling of genome-wide DNA methylation from sequencing enrichment experiments. *Nucleic Acids Res.* 45, e44.
- Lill, M.A., Danielson, M.L., 2011. Computer-aided drug design platform using PyMOL. *J. Comput. Aided Mol. Des.* 25, 13–19.
- Ma, X., Liu, Z., Zhang, Z., Huang, X., Tang, W., 2017. Multiple network algorithm for epigenetic modules via the integration of genome-wide DNA methylation and gene expression data. *BMC Bioinform.* 18, 72.
- Man, M.Z., Wang, X., Wang, Y., 2000. POWER_SAGE: Comparing statistical tests for SAGE experiments. *Bioinformatics* 16, 953–959.
- Matys, V., Kel-Margoulis, O.V., Fricke, E., *et al.*, 2006. TRANSFAC and its module TRANSCOMP: Transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.* 34, D108–D110.
- Medvedeva, Y.A., Lennartsson, A., Ehsani, R., *et al.*, 2015. EpiFactors: A comprehensive database of human epigenetic factors and complexes. *Database (Oxford)* 2015, bav067.
- Mi, H., Poudel, S., Muruganujan, A., Casagrande, J.T., Thomas, P.D., 2016. PANTHER version 10: Expanded protein families and functions, and analysis tools. *Nucleic Acids Res.* 44, D336–D342.
- Millar, A.A., Waterhouse, P.M., 2005. Plant and animal microRNAs: Similarities and differences. *Funct. Integr. Genom.* 5, 129–135.
- Miller, M.B., Tang, Y.W., 2009. Basic concepts of microarrays and potential applications in clinical microbiology. *Clin. Microbiol. Rev.* 22, 611–633.
- Mode, C.J., Gallop, R.J., 2008. A review on Monte Carlo simulation methods as they apply to mutation and selection as formulated in Wright-Fisher models of evolutionary genetics. *Math. Biosci.* 211, 205–225.
- Monteiro, P.T., Pais, P., Costa, C., *et al.*, 2017. The PathoYeast database: An information system for the analysis of gene and genomic transcription regulation in pathogenic yeasts. *Nucleic Acids Res.* 45, D597–D603.
- Moore, A.C., Winkler, J.S., Tseng, T.T., 2015. Bioinformatics resources for microRNA discovery. *Biomark. Insights* 10, 53–58.
- Morison, I.M., Paton, C.J., Cleverley, S.D., 2001. The imprinted gene and parent-of-origin effect database. *Nucleic Acids Res.* 29, 275–276.
- Munch, R., Hiller, K., Barg, H., *et al.*, 2003. PRODORIC: Prokaryotic database of gene regulation. *Nucleic Acids Res.* 31, 266–269.
- Nalluri, J.J., Barh, D., Azevedo, V., Ghosh, P., 2017. miRsig: A consensus-based network inference methodology to identify pan-cancer miRNA-miRNA interaction signatures. *Sci. Rep.* 7, 39684.
- Ng, P.C., Henikoff, S., 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Res.* 31, 3812–3814.
- O'Connor, T.R., Wyrick, J.J., 2007. ChromatinDB: A database of genome-wide histone modification patterns for *Saccharomyces cerevisiae*. *Bioinformatics* 23, 1828–1830.
- Panwar, B., Omenn, G.S., Guan, Y., 2017. miRmine: A database of human miRNA expression profiles. *Bioinformatics* 33, 1554–1560.
- Ponger, L., Mouchiroud, D., 2002. CpGProD: Identifying CpG islands associated with transcription start sites in large genomic mammalian sequences. *Bioinformatics* 18, 631–633.
- Proudfoot, N.J., Shander, M.H., Manley, J.L., Gefter, M.L., Maniatis, T., 1980. Structure and in vitro transcription of human globin genes. *Science* 209, 1329–1336.
- Rupaimoole, R., Slack, F.J., 2017. MicroRNA therapeutics: Towards a new era for the management of cancer and other diseases. *Nat. Rev. Drug Discov.* 16, 203–222.
- Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* 23, 2507–2517.
- Sandelin, A., Wasserman, W.W., 2004. Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics. *J. Mol. Biol.* 338, 207–215.

- Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W.W., Lenhard, B., 2004. JASPAR: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 32, D91–D94.
- Schlitt, T., Palin, K., Rung, J., *et al.*, 2003. From gene networks to gene function. *Genome Res.* 13, 2568–2576.
- Schubeler, D., 2015. Function and information content of DNA methylation. *Nature* 517, 321–326.
- Schulz, R., Woodfine, K., Menhenniott, T.R., *et al.*, 2008. WAMIDEX: A web atlas of murine genomic imprinting and differential expression. *Epigenetics* 3, 89–96.
- Schweiger, M.R., Barmeyer, C., Timmermann, B., 2013. Genomics and epigenomics: New promises of personalized medicine for cancer patients. *Brief. Funct. Genom.* 12, 411–421.
- Selvaraj, S., Natarajan, J., 2011. Microarray data analysis and mining tools. *Bioinformatics* 6, 95–99.
- Shah, P.P., Hutchinson, L.E., Kakar, S.S., 2009. Emerging role of microRNAs in diagnosis and treatment of various diseases including ovarian cancer. *J. Ovarian Res.* 2, 11.
- Sharma, D., Mohanty, D., Suroolia, A., 2009. RegAnalyst: A web interface for the analysis of regulatory motifs, networks and pathways. *Nucleic Acids Res.* 37, W193–W201.
- Sharma, D., Issac, B., Raghava, G.P., Ramaswamy, R., 2004. Spectral Repeat Finder (SRF): Identification of repetitive sequences using Fourier transformation. *Bioinformatics* 20, 1405–1412.
- Simon, R., Lam, A., Li, M.C., *et al.*, 2007. Analysis of gene expression data using BRB-ArrayTools. *Cancer Inform.* 3, 11–17.
- Slonim, D.K., Yanai, I., 2009. Getting started in gene expression microarray analysis. *PLOS Comput. Biol.* 5, e1000543.
- Song, Q., Decato, B., Hong, E.E., *et al.*, 2013. A reference methylome database and analysis pipeline to facilitate integrative and comparative epigenomics. *PLOS ONE* 8, e81148.
- Stenson, P.D., Mort, M., Ball, E.V., *et al.*, 2014. The Human Gene Mutation Database: Building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine. *Hum. Genet.* 133, 1–9.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* 102, 15545–15550.
- Sunkar, R., Li, Y.F., Jagadeeswaran, G., 2012. Functions of microRNAs in plant stress responses. *Trends Plant Sci.* 17, 196–203.
- Tavan, P., Grubmüller, H., Kuhnel, H., 1990. Self-organization of associative memory and pattern classification: Recurrent signal processing on topological feature maps. *Biol. Cybern.* 64, 95–105.
- Taylor, R.S., Tarver, J.E., Hiscock, S.J., Donoghue, P.C., 2014. Evolutionary history of plant microRNAs. *Trends Plant Sci.* 19, 175–182.
- Teng, L., He, B., Wang, J., Tan, K., 2015. 4DGenome: A comprehensive database of chromatin interactions. *Bioinformatics* 31, 2560–2564.
- Tombol, Z., Szabo, P.M., Molnar, V., *et al.*, 2009. Integrative molecular bioinformatics study of human adrenocortical tumors: MicroRNA, tissue-specific target prediction, and pathway analysis. *Endocr. Relat. Cancer* 16, 895–906.
- Trucchi, E., Mazzarella, A.B., Gilfillan, G.D., *et al.*, 2016. BsRADseq: Screening DNA methylation in natural populations of non-model species. *Mol. Ecol.* 25, 1697–1713.
- Tuteja, R., Tuteja, N., 2004. Serial analysis of gene expression (SAGE): Unraveling the bioinformatics tools. *Bioessays* 26, 916–922.
- Ui-Tei, K., Naito, Y., Takahashi, F., *et al.*, 2004. Guidelines for the selection of highly effective siRNA sequences for mammalian and chick RNA interference. *Nucleic Acids Res.* 32, 936–948.
- van Aalten, D.M., Bywater, R., Findlay, J.B., *et al.*, 1996. PRODRG, a program for generating molecular topologies and unique molecular descriptors from coordinates of small molecules. *J. Comput. Aided Mol. Des.* 10, 255–262.
- van Steensel, B., Dekker, J., 2010. Genomics tools for unraveling chromosome architecture. *Nat. Biotechnol.* 28, 1089–1095.
- Vidal, L., Blagden, S., Attard, G., De Bono, J., 2005. Making sense of antisense. *Eur. J. Cancer* 41, 2812–2818.
- Vinayagam, A., Hu, Y., Kulkarni, M., *et al.*, 2013. Protein complex-based analysis framework for high-throughput data sets. *Sci. Signal.* 6, rs5.
- Wakaguri, H., Yamashita, R., Suzuki, Y., Sugano, S., Nakai, K., 2008. DBTSS: Database of transcription start sites, progress report 2008. *Nucleic Acids Res.* 36, D97–D101.
- Wang, Q., Huang, J., Sun, H., *et al.*, 2014. CR Cistrome: A ChIP-Seq database for chromatin regulators and histone modification linkages in human and mouse. *Nucleic Acids Res.* 42, D450–8.
- Wan, X., Yang, C., Yang, Q., *et al.*, 2010. BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. *Am. J. Hum. Genet.* 87, 325–340.
- Wen, B., Wu, H., Shinkai, Y., Irizarry, R.A., Feinberg, A.P., 2009. Large histone H3 lysine 9 dimethylated chromatin blocks distinguish differentiated from embryonic stem cells. *Nat. Genet.* 41, 246–250.
- Wingender, E., Dietze, P., Karas, H., Knuppel, R., 1996. TRANSFAC: A database on transcription factors and their DNA binding sites. *Nucleic Acids Res.* 24, 238–241.
- de Wit, E., de Laat, W., 2012. A decade of 3C technologies: Insights into nuclear organization. *Genes Dev.* 26, 11–24.
- Wong, N., Wang, X., 2015. miRDB: An online resource for microRNA target prediction and functional annotations. *Nucleic Acids Res.* 43, D146–D152.
- Xie, F., Zhang, B., 2010. Target-align: A tool for plant microRNA target identification. *Bioinformatics* 26, 3002–3003.
- Xin, Y., Chanrion, B., O'donnell, A.H., *et al.*, 2012. MethylomeDB: A database of DNA methylation profiles of the brain. *Nucleic Acids Res.* 40, D1245–D1249.
- Xiong, Y., Wei, Y., Gu, Y., *et al.*, 2017. DiseaseMeth version 2.0: A major expansion and update of the human disease methylation database. *Nucleic Acids Res.* 45, D888–D895.
- Xu, J., Wu, C., Che, X., *et al.*, 2011. Circulating microRNAs, miR-21, miR-122, and miR-223, in patients with hepatocellular carcinoma or chronic hepatitis. *Mol. Carcinog.* 50, 136–142.
- Ye, P., Luan, Y., Chen, K., *et al.*, 2017. MethSMRT: An integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing. *Nucleic Acids Res.* 45, D85–D89.
- Zagganas, K., Vergoulis, T., Paraskevopoulou, M.D., *et al.*, 2017. BUFET: Boosting the unbiased miRNA functional enrichment analysis using bitsets. *BMC Bioinform.* 18, 399.
- Zeeberg, B.R., Qin, H., Narasimhan, S., *et al.*, 2005. High-Throughput GoMiner, an 'industrial-strength' integrative gene ontology tool for interpretation of multiple-microarray experiments, with application to studies of Common Variable Immune Deficiency (CVID). *BMC Bioinform.* 6, 168.
- Zhang, C., Li, G., Zhu, S., Zhang, S., Fang, J., 2014. tasiRNAdb: A database of ta-siRNA regulatory pathways. *Bioinformatics* 30, 1045–1046.
- Zhang, J., Baran, J., Cros, A., *et al.*, 2011. International Cancer Genome Consortium Data Portal – A one-stop shop for cancer genomics data. *Database (Oxford)* 2011, bar026.
- Zhang, X., Huang, S., Zou, F., Wang, W., 2010a. TEAM: Efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics* 26, i217–i227.
- Zhang, Y., Liu, J.S., 2007. Bayesian inference of epistatic interactions in case-control studies. *Nat. Genet.* 39, 1167–1173.
- Zhang, Y., Lv, J., Liu, H., *et al.*, 2010b. HHMD: The human histone modification database. *Nucleic Acids Res.* 38, D149–D154.
- Zhu, C., Ding, Y., Liu, H., 2011. MiR398 and plant stress responses. *Physiol. Plant* 143, 1–9.
- Zhu, J., Zhang, M.Q., 1999. SCPD: A promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics* 15, 607–611.
- Zou, D., Sun, S., Li, R., *et al.*, 2015. MethBank: A database integrating next-generation sequencing single-base-resolution DNA methylation programming data. *Nucleic Acids Res.* 43, D54–D58.

Relevant Websites

<http://omictools.com/epistasis-detection-category>
Epistasis detection software tools | GWAS analysis
OMICtools.

<http://www.geneimprint.com>
Geneimprint.

<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>
LIBSVM.

<https://omictools.com/driver-mutations-category>
OMICtools.

<http://www.sagenet.org>
SageNet.

Transcription Factor Databases

Edgar Wingender, geneXplain GmbH, Wolfenbüttel, Germany and University Medical Center Göttingen, Göttingen, Germany

Alexander Kel and Mathias Krull, geneXplain GmbH, Wolfenbüttel, Germany

© 2019 Elsevier Inc. All rights reserved.

Background

Transcription, i.e., the synthesis of an RNA strand that is complementary to a piece of genomic DNA of an organism, is an essential step of gene expression. To our present knowledge, it is the most tightly and subtly regulated mechanism of the whole process of making a gene product according to genomic instructions. Transcribing DNA into RNA is done by a small class of enzymes, the DNA-dependent RNA polymerases. To direct polymerase molecules to a specific place in the genome, the transcription start site (TSS), the assistance of additional factors is required, which are called transcription factors (TFs). In the beginning of the research on these factors, the focus was on a set of co-factors of the polymerase that we call today general transcription factors (GTFs). Best studied for bacteria are the sigma factors, for eukaryotic a set of GTFs is known such as TFIIA, TFIIB, TFIID-F, TFIIF. Some of these GTFs bind directly to the DNA with a more or less relaxed sequence-specificity, others are included through protein-protein interactions. They are involved in the assembly of the transcription complex at the core promoter, the region around the TATA-box at ~ -30 bp of the TSS and the TSS itself. A much larger set of DNA-binding TFs that bind to specific regulatory sequences, or cis-regulating elements, have been discovered later. In addition, there are factors involved in transcriptional regulation that do not bind directly to DNA by themselves, but are incorporated into DNA-bound complexes through protein-protein interactions. Depending on the TF definition applied, they may be included or some of them may be split off as separate class of, e.g., co-activators etc. In bacteria, we may have up to 300 TFs (Gama-Castro *et al.*, 2016), while the human genome may encode up to 2000 TFs (Wingender *et al.*, 2013), the exact number depending on the definition.

Aims of TF Databases

A TF database (TFDB) may first of all store information about TFs. In doing so, it should provide additional information rather than just serving as a specialized subsection of a comprehensive protein resource such as UniProt. Therefore, already the first compilation of TFs from 1988 that developed into the database TRANSFAC combined information about (eukaryotic) TFs with specifications of their DNA binding sites (Wingender, 1988). As an increasing amount of transcription factor binding sites (TFBSs) was gathered from the scientific literature, it was possible to complement the TFDB contents by binding sites models, for instance as consensus sequences, as positional weight matrices (PWMs) or as Hidden Markov Models (HMMs). Libraries of such models were compiled and are used now to computationally predict potential TFBSs in yet uncharacterized DNA sequences. Also on the side of the binding proteins, the increase of the number of known TFs and their characterization, in particular the mapping of their DNA-binding domains (DBDs), allowed to come up with corresponding domain models that became part of general protein motif collections like InterPro or of specialized TFDBs (see Fig. 1 for a simplified scheme). They are used to predict potential TFs in not yet annotated genomes, or to classify TFs according to their DBDs, e.g., in TFClass (Wingender *et al.*, 2013).

For the purpose of predicting corresponding entities (TFBSs or TFs), several of the TFDBs available today are accompanied with pieces of software that support these tasks.

Among the TFDBs that have been developed so far, some have a more comprehensive scope and try to cover the whole field of prokaryotic or eukaryotic TFs, while others are more focussed on certain taxa such as insects or plants. Some will be described in the following, others can be found in Table 1. Moreover, the individual TFDBs differ in the way their content has been curated, whether manually or by text mining approaches, or a mix of both. The exact method by which the contents have been gathered is not always clear.

TRANSFAC

The oldest still actively maintained database in the field is the TRANSFAC database. Its contents are basically structured in tables that provide information about the TFs (FACTOR table) or their genomic binding sites (SITE). All information in these tables has been manually extracted from the original publications. Semi-automatically retrieved are, however, data from high-throughput experiments such as ChIPseq data. Large numbers of genomic TF binding fragments can arise from these approaches, millions of them are stored in a separate table of the database (FRAGMENT).

On top of these entities, abstract models of DNA binding specificities as PWMs (MATRIX) or of DBDs (CLASS) have been implemented (Fig. 1), the contents of which are semi-manually curated. The entries of the MATRIX table are assigned to TFs of one of four taxonomic groups (vertebrates, insects, plants, and fungi). Many entries have been generated by TRANSFAC curators from information compiled in the SITE table. Other MATRIX entries were published in scientific reports and were included for documentary purposes. This dual purpose made it necessary to set up filters discriminating between those matrices that are suitable for predicting TFBSs and others, which were incorporated for archiving.

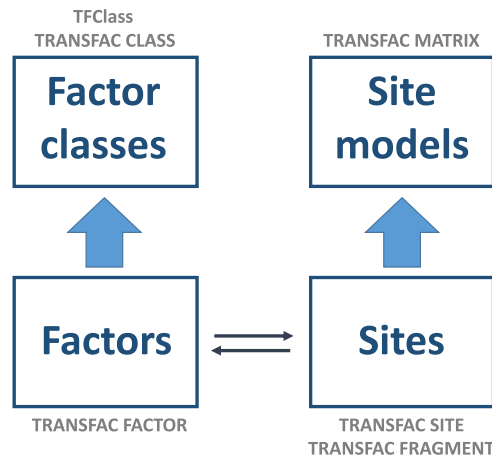


Fig. 1 Simplified scheme of content types of TFDBs. Individual databases usually focus on some of these content areas. As examples, the corresponding tables of the TRANSFAC database and associated resources have been indicated. “Factors” stands for information about transcription factors (TFs), “Sites” for TF binding sites (TFBS), either conventionally identified or obtained from high-throughput experiments such as ChIPseq, both pointing to each other. On top of these entities, abstract models of DNA-binding domains of TFs and of TFBSs are frequently provided for both descriptive and predictive purposes.

An accompanying resource is the database TRANSCompel, the concept of which has emerged from the COMPEL database on composite elements (CEs) (Kel *et al.*, 1995). These are binary or higher-order combinations of individual TFBSs, which together convey a qualitatively and/or quantitatively new regulatory feature to the associated gene. They serve as integrative modules, where two regulatory activities exert a synergistic effect; for instance, one TFBS of a CE may interact with a tissue-specific TF, while the partner site may convey the response to a certain signal transduction pathway. Another accompanying database is TRANSPro, a comprehensive compilation of the promoters of several genomes (human, mouse, rat, and the plant *Arabidopsis thaliana*). TRANSFAC is connected with TFClass, a classification system for TFs according to their DBDs, so far applied to mammalian TFs (Wingender *et al.*, 2018). Since 1999, another complementary information resource has been established that represents signaling pathways aiming at TFs (Heinemeyer *et al.*, 1999). This database, TRANSPATH, has been focussing on pathways that lead to the activation or inactivation of TFs, although its scope was expanded later on.

TRANSFAC strives to cover the whole area of eukaryotic TFs, although the coverage of different taxa may be significantly different.

First attempts to use TRANSFAC contents for the prediction of TFBSs made use of the information about the experimentally validated sites compiled in the SITE table. Soon after, partly based on TRANSFAC’s matrix library, several algorithms were developed that utilize PWMs for TF search, e.g., ConsInspector (Frech *et al.*, 1993), MATRIX SEARCH (Chen *et al.*, 1995), MatInspector (Quandt *et al.*, 1995), MATCH (Kel *et al.*, 2003a,b) and several others (see Relevant Websites Section). TRANSFAC itself is accompanied by MATCH (Kel *et al.*, 2003a,b). The program identifies TF binding sites at three different levels of stringency either minimizing the false positive or the false negative rate, or the rate of the sum of both error rates. The different algorithms mainly differ in the way how site scores are computed. In some of the earlier approaches some algorithms use matrix regularization (pseudo-counts) and forbidden nucleotides in order to account for specific constraints of the nucleotide distribution in the TF motifs. Similar to the MatInspector approach, MATCH applies specific weighting of PWM positions by information content of the nucleotide composition in the given position of the matrix. It was shown that this approach often increases precision of TF site recognition (Kondrakhin *et al.*, 2016). On top of MATCH results, the program F-Match identifies TF binding sites that are enriched in a set of regulatory sequences (e.g. a set of promoter sequences of co-regulated genes or a set of ChIP-seq peaks of active chromatin region). Combinations of predicted TF sites can be identified with a specific genetic algorithm implemented in the program CMA (Composite Module Analyst) (Kel *et al.*, 2006). CMA and other algorithms with similar scope were independently reviewed by Klepper *et al.* (2008).

The TRANSFAC database started in 1988 first as a private initiative and continued later as a public project. To secure long-term financial support, a company (BIOBASE) was established in 1999 to maintain and distribute the database commercially. An older version was kept in the public domain. Today, TRANSFAC is maintained, further developed and distributed by the company geneXplain. The FACTOR table of TRANSFAC has recently been made freely accessible.

Since its publication, the aims and ideas of the TRANSFAC database were adopted by a number of other databases, many of them also adding their own concepts.

JASPAR

JASPAR started as a carefully selected, non-redundant compilation of high-quality profiles, or position-specific scoring matrices that were derived from experimentally validated TF binding sites (Sandelin *et al.*, 2004). The database expanded over the years and more recently included third-generation binding models, which in contrast to PWMs or PSSMs consider position

Table 1 Overview of transcription factor databases, their scope and associated tools

Database name	URL	Species focus	Project start	Last updated	Brief summary	Tools	Country
3d-footprint	http://floresta.eead.csic.es/3dfootprint/	Eukaryotes		2017	Dedicated to structural data of DNA-binding proteins. Comprises clustering and classification of protein-DNA complexes.	Structure analysis, consensus literature miner, BLAST	Spain
AnimalTFDB2.0	http://bioinfo.life.hust.edu.cn/AnimalTFDB/	Vertebrates		2015?	General transcription factor (TF) database that focuses on curation and classification of genome-wide TFs, chromatin modifying proteins and transcription co-factors. The classification is based on DNA-binding domain characteristics.	TF prediction; BLAST	China
ArchaeaTF	http://bioinformatics.zj.cn/archaeatf/	Archaea	2006	2008	Database on (putative) transcription factors in Archaeal genomes. Due to annotated sequence characteristics, it can be useful for comparative genomics analysis of TFs in Archaea	Phylogenetic tree; BLAST	China
AthaMap	http://www.athamap.de/	Plants; Arabidopsis	2004	2016	Species-specific database on genome-wide putative transcription factor and small RNA binding sites in Arabidopsis thaliana. The sites are either experimentally verified or predicted by positional weight matrices representing TF binding specificities	TFBS and small/miRNA RNA target site search	Germany
ATRM (Arabidopsis Transcriptional Regulatory Map)	http://atrm.cbi.pku.edu.cn/	Plants; Arabidopsis	2014?	2017	Depicts the transcriptional regulatory network in Arabidopsis thaliana based on literature text mining and manual annotation		China
CIS-BP	http://cisbp.cabr.utoronto.ca/	Plants; Arabidopsis	2003?	2012?	Contains information on Arabidopsis thaliana transcription factors. The TF are classified as members of 50 families. The Catalog of Inferred Sequence Binding Preferences (CIS-BP). It documents a large number of TF binding motifs/matrices, also from a variety of other sources.	TF binding site prediction tools	US
CollectF	http://www.collectf.org/browse/home/	Eukaryotes	2014	2015	Database for TFBS in bacteria. All sites have been experimentally verified and manually curated.		Canada
CoryneRegNet	https://coryneregnet.compbio.sdu.dk/v6/index.html	Prokaryotes; Corynebacteria	2013?	2015	Reference database for TF in Corynebacteria and their regulatory networks. Comprises experimentally-verified as well as automatically predicted regulation data.	TFBS prediction tool	US
CTCFBSDB	http://insulatordb.uthsc.edu/	Eukaryotes		2012	Dedicated to CTCF binding sites and genome organization. Contains experimentally-verified as well as automatically predicted sites.	Site prediction tool	US
ctFbase	http://bioinformatics.zj.cn/ctFbase/Home.php	Cyanobacteria		2007	Database focused on putative transcription factors in the genomes of Cyanobacteria. Classification of TFs based on DBD characteristics.	BLAST; sequence alignment, phylogenetic tree	China
DBD	http://www.transcriptionfactor.org/index.cgi?Home	Eukaryotes	2008	2011	Genome-wide collection of putative TF with assignments to sequence specific DNA-binding domain families. The DBD database covers 930 sequenced genomes.		UK
DBTBS	http://dbtbs.hgc.jp/	Prokaryotes; Bacillus subtilis	1999	2008	Resource dedicated to experimentally validated TFBS in gene regulatory regions of Bacillus subtilis. All cis-elements (TFBS) in a promoter are depicted together with their positions.		Japan
iDMMPMM	http://autosome.ru/iDMMPMM/	Drosophila	2009	2009	Drosophila melanogaster collection of transcription factor DNA-binding motifs from several experimental sources	Genome Browser	Russia
DOOR ²	http://csbl.bmb.uga.edu/DOOR2/index.php#tabs-1	Prokaryotes	2009	2013	Comprehensive prokaryotic operon database with data from 2072 bacterial genomes. The operons are predicted based on essential genomic features. Contains transcription units from experiments and RNA-seq predictions.		US

ENCODE	https://www.encodeproject.org/	Human, mouse, worm, fly	2017	Collection of high-throughput data about functional DNA elements, including TF binding regions from ChIP-seq experiments conducted in different cell types.	US
Factorbook	http://www.factorbook.org/human/chipseq/tf/	Human, mouse	2012? 2017?	Factorbook summarizes the results of ENCODE TF ChIP-seq data, together with ChIP-seq of histone modification and nucleosome occupancy experiments	US
Fly Factor Survey	http://mccb.umassmed.edu/tfs/	Drosophila	2008? 2013?	Focuses on DNA-binding specificities of TF in Drosophila melanogaster based on results obtained from systematic application of the bacterial one-hybrid method.	USA
footprintDB	http://floresta.eead.csic.es/footprintdb/	Eukaryotes	2016 2017	Resource that collects TFs, TFBS and their DNA-binding specificity representing matrices from various sources.	Spain
Grassius	http://grassius.org/	Plants; Rice, Maize, Sorghum etc.	2008 2017	Collection of databases focused on gene regulation control in grass plants. Captures interactions of transcription factors with DNA-binding sites in promoter regions by literature curation and community contribution.	USA
GTRD	http://gtrd.biouml.org/	Human, mouse	2017	The Gene Transcription Regulation Database (GTRD) has a comprehensive collection of uniformly processed ChIP-seq data sets, allowing to deduce TFBS in human and mouse.	Russia
hmChIP	http://ijlab.biostat.jhsph.edu/database/cgi-bin/hmChIP.pl	Human, mouse	2011?	Resource to collect ChIP-chip and ChIP-seq data sets from experiments conducted in human and mouse cell lines or tissues.	USA
HOCOMOCO	http://hocomoco.autosome.ru/	Human, mouse	2017	The Homo sapiens Comprehensive Model Collection (HOCOMOCO) contains mono- and dinucleotide TF DNA-binding models for more than 1100 human and mouse factors.	Russia
HOMER	http://homer.ucsd.edu/homer/index.html	Eukaryotes	2010 2017	The HOMER website combines TF motif databases with a software for binding site prediction and next-generation sequencing analysis.	USA
HTPSELEX	http://ceg.vital-it.ch/htpselex/		2010	This database comprises sets of in vitro selected TFBS sequences obtained by SELEX and high-throughput SELEX experiments.	
HTRdb	http://www.lbbc.lbbc.unesp.br/htr/	Human	2009 2012?	Database on regulatory interactions between human transcription factors and their target genes. The interactions are experimentally verified and manually curated from the biomedical literature.	Switzerland
Islet Regulome Browser	http://gattaca.imppc.org/isletregulome/info			Visualization tool for maps of ChIP-seq derived TFBS in pancreatic islet and progenitor cells	
ITAK database	http://itak.feilab.net/cgi-bin/itak/db_home.cgi	Plants	2010 2016	Database for putative plant transcription factors and protein kinases predicted by the classifier software ITAK.	Brazil
JASPAR	http://jaspar.genereg.net/	Eukaryotes	2004 2017	Collection of non-redundant transcription factor DNA-binding models for multiple species. They are stored as Position Frequency Matrices (PFMs) and TF flexible models (TFFMs).	UK
MtbRegList	http://mtbreglist.genap.ca/MtbRegList/www/index.php	Prokaryotes; M. tuberculosis	2005 2005	Resource focused on the analysis of gene regulation mechanisms in Mycobacterium tuberculosis. Contains transcription start sites and TFBS as well as the respective transcription factors.	Canada
newPLACE	https://sogo.dna.affrc.go.jp/cgi-bin/sogo.cgi?lang=en&pj=640&action=page&page=newplace	Plants	1999 2007?	Database of motifs detected in plant cis-acting regulatory DNA elements, extracted from published literature.	Japan
NURSA	https://www.nursa.org/nursa/index.jsf	Human, mouse, rat	2003 2017	A website dedicated to accrue information on nuclear receptors and co-regulators and their physiological roles. The data includes NR ligands and transcriptional targets.	USA
P2TF	http://www.p2tf.org/	Prokaryotes	2012?	BLAST, Genome Browser	France (Continued)

Table 1 Continued

Database name	URL	Species focus	Project start	Last updated	Brief summary	Tools	Country
PAZAR	http://www.pazar.info/	Eukaryotes		will retire end of 2018	The Predicted Prokaryotic Transcription Factors (P2TF) database includes putative TFs of a large number of bacterial and archaeal genomes as well as a TF classification.		Canada
Plant Cistrome Database	http://neomorph.salk.edu/dap_web/pages/index.php	Plants; Arabidopsis; Maize		2016	Website hosting TF and regulatory sequence collections. It provides a software framework for regulatory sequence annotation tasks.	Genome Browser, Matrix comparison and clustering	USA
PlantPAN 2.0	http://plantpan2.https.ncu.edu.tw/index.html	Plants	2008	2015	Resource that provides TF binding profiles generated out of DAP-seq experiments in plants.	Network visualization, promoter analysis	Taiwan
PlantRegulome	http://plantregulome.org/	Plants; Arabidopsis	2013	2013	The Plant Promoter Analysis Navigator (PlantPAN) contains TFs, binding sites, and motifs for 76 plant species and allows to analyze promoters and to construct regulatory networks.	Regulatory network visualization	USA
PlantTFDB	http://planttfdb.cbi.pku.edu.cn/	Plants	2010	2017	Focuses on creating maps of gene regulatory information in various developmental stages, tissues and cell types of Arabidopsis thaliana.	TF prediction server; tools for regulation prediction and functional enrichment analyses	China
PlnTFDB	http://plntfdb.bio.uni-potsdam.de/v3.0/	Plants	2009		Database storing TF from 165 plant species as a result from an automated prediction pipeline. The TF are classified by DBD family assignments.	BLAST	Germany
PRODORIC	http://www.prodoric.de/	Prokaryotes	2001	2008	The resource aims to identify and classify all plant genes involved in transcriptional control. The TF family classification is based on characteristic domains described in the literature.	Genome Browser, network analysis and visualization	Germany
RegulonDB	http://regulondb.ccg.unam.mx/	Prokaryotes; E. coli	1998	2017	A database containing comprehensive gene regulation and expression information in prokaryotes. It includes experimentally validated and manually curated transcription factor binding sites. It's accompanied by bioinformatics tools for the prediction, analysis and visualization of gene regulatory networks.	Regulatory network visualization, genome browser, text mining, PWM clustering, classification browser	Mexico
ReMap 2018	http://tagc.univ-mrs.fr/remap/index.php	Human	2015	2017	Database dedicated to comprehensively model the transcriptional regulatory network in E. Coli K-12. It includes information on transcription units, operons and regulons. The data is manually curated from peer-reviewed literature.	Statistical search of TF binding peaks	France
Riken TFdb	http://genome.gsc.riken.jp/TFdb/	Mouse		2004?	A platform that collects TFBS ChIP-seq data sets from ENCODE, GEO and ArrayExpress. Peaks have been uniformly processed and are available for either in total or as non-redundant merged peaks.		Japan
ScerTF	http://stormo.wustl.edu/ScerTF	Yeast		2012?	A database containing a set of mouse non-redundant transcription factors and their related genes.	TFBS prediction	USA
STIFDB	http://caps.ncbs.res.in/stifdb2/	Plants; Arabidopsis, rice		2012	A resource that collects non-redundant position weight matrices (PWMs) for yeast transcription factors from literature and public database sources.	Blast	India
TcoF	http://www.cbric.kau.ac.sa/tcof/	Human	2010	2017	The Stress Responsive Transcription Factor Database (STIFDB) is a comprehensive set of biotic and abiotic stress responsive genes and their potential TFBS in respective promoter regions in Arabidopsis thaliana and Oryza sativa.		Saudi-Arabia
TEC	https://shigen.nig.ac.jp/ecoli/tec/top/	Prokaryotes; E. coli	2010?	2016?	A database focusing on human transcription co-factors and TF interacting proteins.		Japan

TFBSshape	http://rohsiab.cmb.usc.edu/TFBSshape/	Eukaryotes	2014?	TEC contains regulatory targets for TFs and sigma factors in <i>E. coli</i> K-12 determined by SELEX-chip screening. TFBSshape includes predicted DNA shape features for TFBS, characterizing DNA binding specificities of TFs from 23 species.	USA
TFcat	http://www.tfcat.ca/	Human, mouse	2009?	Catalog of mouse and human transcription factors (TF) based manual curation from the biomedical literature. In addition, TF encoding genes have been predicted based on sequence analysis. DNA-binding TF are assigned to a structural classification system.	Canada
TFcheckpoint	http://www.tfcheckpoint.org/	Human, mouse, rat	2014?	Manually curated database for human, mouse and rat TFs based on public database and literature sources.	Norway
Tfe	http://cisreg.cmmr.ubc.ca/cgi-bin/tfe/home.pl	Human, mouse, rat	2012	The Transcription factor encyclopedia (Tfe) aims to create encyclopedic articles for each well-studied TF.	Canada
TFindit	http://bioinfopen.uncc.edu/tfindit/	Eukaryotes	2012	Database and web service for structural information of transcription factor (TF)-DNA interactions. It contains pairs of bound and unbound transcription factor structures and homologous TF-DNA complex structures.	USA
TRANSFAC	http://genexplain.com/transfac/ Public: http://gene-regulation.com/pub/databases.html Factor search: http://factor.genexplain.com/cgi-bin/transfac_factor/search.cgi http://www.mgs.bionet.nsc.ru/mgs/gmw/trrd/	Eukaryotes	1988	TRANSFAC® is the database of eukaryotic transcription factors, their genomic binding sites and DNA-binding profiles. Its TFBS data is experimentally verified and manually curated from peer-reviewed literature.	Germany
TRRD	http://www.mgs.bionet.nsc.ru/mgs/gmw/trrd/	Eukaryotes	1995	The Transcription Regulatory Regions Database (TRRD) collects information on structural and functional organization of transcriptional regulatory regions of eukaryotic genes. All data in TRRD has been experimentally verified.	Russia
TRRUST	http://www.grnpedia.org/trust/	Human, mouse	2014	Manually curated database of human and mouse transcriptional regulatory networks. For a large part of the stored TF-target gene relations, the mode of regulation (activation or repression) is provided.	South Korea
UniProbe	http://the_brain.bwh.harvard.edu/uniprobe/	Eukaryotes	2008	The UniPROBE (Universal PBM Resource for Oligonucleotide Binding Evaluation) database contains data on in vitro DNA-binding specificities of proteins created by the universal protein binding microarray (PBM) experimental method.	USA
Yeastract	http://www.yeasttract.com/index.php	Yeast	2006	YEASTRACT (Yeast Search for Transcriptional Regulators And Consensus Tracking) is a curated database of regulatory interactions between TF and target genes in <i>Saccharomyces cerevisiae</i> .	Portugal
YefasCo	http://yefasco.ccbr.utoronto.ca/	Yeast	2012	The site provides a collection of TF specificities for <i>Saccharomyces cerevisiae</i> as position frequency matrices (PFM) or position weight matrices (PWM).	Canada

interdependencies within and variable lengths of the TF binding motifs. These Transcription Factor Flexible Models (TFFMs) are built with the help of hidden Markov models (HMMs) (Mathelier and Wasserman, 2013). The TFFMs of the provided library have been created from ChIP-seq data coming from the ENCODE project. Software tools for the application of these advanced binding models accompany the library. They are complemented by tools that support the creation of new TFFMs as well as to scan sequences with them (Mathelier et al., 2016). – As for the factor side, JASPAR is complemented by TFE, a transcription factor encyclopedia, which so far as compiled more than 800 expert-written articles about individual TFs.

Plant TFs

Two actively maintained resources for plant TFs have to be highlighted. One is **PlantTFDB** from Peking University, which covers properties of TFs from so far 165 plant species. Using an automated TF prediction pipeline, more than 320,000 TFs were identified to be encoded in plant genomes and can be browsed by (DBD) family assignments. Making use of a related resource of the same group, PlantRegMap, binding site and binding motif information has been made available as well (Jin et al., 2017). TFBSs have been partly taken from literature, where experimental evidence was given, or predicted by using the binding motif information. – Another program to identify TFs encoded in plant genomes is iTAK. It was trained with contents of PlantTFDB and others, and the results are provided online as database (Zheng et al., 2016). In addition, there are several databases that focus on individual species, mostly the best-studied plant organism *Arabidopsis thaliana*, such as AthaMap (see also Table 1) (Hehl et al., 2016).

Other Model Organisms

There are several other TFDBs that provide corresponding information with different emphasis on individual binding sites, high-throughput data on binding regions, or binding motifs for the factors of, e.g., *Drosophila* (Fly Factor Survey; Zhu et al., 2011) or yeast (Yeasttract; Teixeira et al., 2014).

TFDBs on Prokaryotic Transcription Factors

One of the most popular databases about entities of prokaryotic transcription is **RegulonDB** (Huerta et al., 1998). Structurally similar to TRANSFAC, the early database version provided in-depth information about *E. coli* TFs and their binding sites. As a unique feature, RegulonDB put the co-regulation of genes or operons in the focus. By time, the database broadened its scope even further by integrating information about regulatory interactions at genes encoding components of defined metabolic pathways with the signals triggering them into so-called Genetic Sensory-Response units (SENSOR units). – A similar resource focussing on *Bacillus subtilis*, **DBTBS**, has been released in 2000 by a team from Tokyo University (Ishii et al., 2001).

With regard to the range of organisms covered, the database **PRODORIC** had a broader approach since it aimed at the whole plethora of prokaryotic organisms. Otherwise, it has similar aims as RegulonDB and a structure resembling TRANSFAC (Münch et al., 2003). Later, the database was complemented with a rich set of software tools (Grote et al., 2009). Among them are a site scanning software, Virtual Footprint, supporting sensitive searches for single or composite TF sites in bacterial genomes. It also visualizes search results in the context of the whole bacterial genome.

Additional Collections of Binding Sites and Motifs

ENCODE (Encyclopedia of DNA Elements) is a consortial project that collects high-throughput data about functional DNA elements, among them TF binding regions from, e.g., ChIPseq studies (Davis et al., 2017). Its main focus is on the human genome, but several model organisms are covered as well. Binding motifs have been extracted from ENCODE datasets that refer to defined TFs, usually with the MEME-ChIP software suite (Machanick and Bailey, 2011), and have been compiled in Factorbook (Wang et al., 2012). At present, it shows binding motifs of 167 TFs derived from 837 experiments.

A principally similar approach, but using pre-processed high-throughput data provided by GTRD (Gene Transcription Regulation Database; Yevshin et al., 2017) and a novel motif discovery tool (ChIPMunk; Kulakovskiy et al., 2010), led to the generation of HOCOMOCO, a collection of mono- and dinucleotide matrices for human and mouse TFs (Kulakovskiy et al., 2018).

Based on a novel experimental approach, the UniPROBE (Universal PBM Resource for Oligonucleotide Binding Evaluation) database has been generated (Hume et al., 2015). The underlying data were obtained by using protein binding microarrays (PBMs) of synthetic oligonucleotides (decamers) to determine those sequences that are suitable to interact with a test protein or its DNA-binding domain in vitro (Mukherjee et al., 2004).

Acknowledgement

TRANSFAC and TRANSPATH are registered trademarks of QIAGEN.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Data Storage and Representation. Genome-Wide Scanning of Gene Expression. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Binding Sites in DNA Sequences. Protein-DNA Interactions. Regulation of Gene Expression

References

- Chen, Q.K., Hertz, G.Z., Stormo, G.D., 1995. MATRIX SEARCH 1.0: A computer program that scans DNA sequences for transcriptional elements using a database of weight matrices. *Comput. Appl. Biosci.* 11, 563–566.
- Davis, C.A., Hitz, B.C., Sloan, C.A., *et al.*, 2017. The encyclopedia of DNA elements (ENCODE): Data portal update. *Nucleic Acids Res.* 2017 (26), gkx1081 (ahead of print).
- Frech, K., Herrmann, G., Werner, T., 1993. Computer-assisted prediction, classification, and delimitation of protein binding sites in nucleic acids. *Nucleic Acids Res.* 21, 1655–1664.
- Gama-Castro, S., Salgado, H., Santos-Zavaleta, A., *et al.*, 2016. RegulonDB version 9.0: High-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res.* D133–D143.
- Grote, A., Klein, J., Retter, I., *et al.*, 2009. PRODORIC (release 2009): A database and tool platform for the analysis of gene regulation in prokaryotes. *Nucleic Acids Res.* 37, D61–D65.
- Hehl, R., Norval, L., Romanov, A., Bülow, L., 2016. Boosting AthaMap database content with data from protein binding microarrays. *Plant Cell Physiol.* 57, e4.
- Heinemeyer, T., Chen, X., Karas, H., *et al.*, 1999. Expanding the TRANSFAC database towards an expert system of regulatory molecular mechanisms. *Nucleic Acids Res.* 27, 318–322.
- Huerta, A.M., Salgado, H., Thieffry, D., Collado-Vides, J., 1998. RegulonDB: A database on transcriptional regulation in *Escherichia coli*. *Nucleic Acids Res.* 26, 55–59.
- Hume, M.A., Barrera, L.A., Gisselbrecht, S.S., Bulyk, M.L., 2015. UniPROBE, update 2015: New tools and content for the online database of protein-binding microarray data on protein-DNA interactions. *Nucleic Acids Res.* 43, D117–D122.
- Ishii, T., Yoshida, K., Terai, G., Fujita, Y., Nakai, K., 2001. DBTBS: A database of *Bacillus subtilis* promoters and transcription factors. *Nucleic Acids Res.* 29, 278–280.
- Jin, J.P., Tian, F., Yang, D.C., *et al.*, 2017. PlantTFDB 4.0: Toward a central hub for transcription factors and regulatory interactions in plants. *Nucleic Acids Res.* 45, D1040–D1045.
- Kel, A.E., Gößling, E., Reuter, I., *et al.*, 2003a. MATCH™: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.* 31, 3576–3579.
- Kel, A.E., Gössling, E., Reuter, I., *et al.*, 2003b. MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.* 31, 3576–3579.
- Kel, A., Kononova, T., Waleev, T., *et al.*, 2006. Composite module analyst: A fitness-based tool for identification of transcription factor binding site combinations. *Bioinformatics* 22, 1190–1197.
- Kel, O.V., Romaschenko, A.G., Kel, A.E., Wingender, E., Kolchanov, N.A., 1995. A compilation of composite regulatory elements affecting gene transcription in vertebrates. *Nucleic Acids Res.* 23, 4097–4103.
- Klepper, K., Sandve, G.K., Abul, O., Johansen, J., Drablos, F., 2008. Assessment of composite motif discovery methods. *BMC Bioinform.* 9, 123.
- Kondrakhin, Y., Valeev, T., Sharipov, R., *et al.*, 2016. Prediction of protein-DNA interactions of transcription factors linking proteomics and transcriptomics data. *EuPA Open Proteom.* 13, 14–23.
- Kulakovskiy, I.V., Boeva, V.A., Favorov, A.V., Makeev, V.J., 2010. Deep and wide digging for binding motifs in ChIP-Seq data. *Bioinformatics* 26, 2622–2623.
- Kulakovskiy, I.V., Vorontsov, I.E., Yevshin, I.S., *et al.*, 2018. HOCOMOCO: Towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Res.* 46, D252–D259.
- Machanic, P., Bailey, T.L., 2011. MEME-ChIP: Motif analysis of large DNA datasets. *Bioinformatics* 27, 1696–1697.
- Mathelier, A., Fornes, O., Arenillas, D.J., *et al.*, 2016. JASPAR 2016: A major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.* 44, D110–D115.
- Mathelier, A., Wasserman, W.W., 2013. The next generation of transcription factor binding site prediction. *PLOS Comput. Biol.* 9, e1003214.
- Mukherjee, S., Berger, M.F., Jona, G., *et al.*, 2004. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nat. Genet.* 36, 1331–1339.
- Münch, R., Hiller, K., Barg, H., *et al.*, 2003. PRODORIC: Prokaryotic database of gene regulation. *Nucleic Acids Res.* 31, 266–269.
- Quandt, K., Frech, K., Karas, H., Wingender, E., Werner, T., 1995. MatInd and MatInspector: New fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res.* 23, 4878–4884.
- Sandelin, A., Alkema, W., Engström, P., Wasserman, W.W., Lenhard, B., 2004. JASPAR: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 32, D91–D94.
- Teixeira, M.C., Monteiro, P.T., Guerreiro, J.F., *et al.*, 2014. The YEASTRACT database: An upgraded information system for the analysis of gene and genomic transcription regulation in *Saccharomyces cerevisiae*. *Nucleic Acids Res.* 42, D161–D166.
- Wang, J., Zhuang, J., Iyer, S., *et al.*, 2012. Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Res.* 22, 1798–1812.
- Wingender, E., 1988. Compilation of transcription regulating proteins. *Nucleic Acids Res.* 16, 1879–1902.
- Wingender, E., Schoeps, T., Dönitz, J., 2013. TFClass: An expandable hierarchical classification of human transcription factors. *Nucleic Acids Res.* 41, D165–D170.
- Wingender, E., Schoeps, T., Haubrock, M., Krull, M., Dönitz, J., 2018. TFClass: Expanding the classification of human transcription factors to their mammalian orthologs. *Nucleic Acids Res.* 46, D343–D347.
- Yevshin, I.S., Sharipov, R.N., Valeev, T.F., Kel, A.E., Kolpakov, F.A., 2017. GTRD: A database of transcription factor binding sites identified by ChIP-seq experiments. *Nucleic Acids Res.* 45, D61–D67.
- Zheng, Y., Jiao, C., Sun, H., *et al.*, 2016. iTAK: A program for genome-wide prediction and classification of plant transcription factors, transcriptional regulators, and protein kinases. *Mol. Plant* 9, 1667–1670.
- Zhu, L.J., Christensen, R.G., Kazemian, M., *et al.*, 2011. FlyFactorSurvey: A database of *Drosophila* transcription factor binding specificities determined using the bacterial one-hybrid system. *Nucleic Acids Res.* 39, D111–D117.

Relevant Websites

- <http://gene-regulation.com/>
Gene regulation.
- https://bip.weizmann.ac.il/toolbox/seq_analysis/promoters.html
Promoters.
- <https://omictools.com/transcription-factor-binding-site-prediction-category>
Transcription Factor Staining.

Protein-DNA Interactions

Preeti Pandey, Sabeeha Hasnain, and Shandar Ahmad, Jawaharlal Nehru University, New Delhi, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

The central dogma of molecular biology involves transcription and translation of genes leading to formation of proteins to carry out specific biological functions. Activation and suppression of this so-called “gene expression” by DNA binding proteins is a fundamental regulatory mechanism involving chromatin modification and transcription complexes to initiate RNA synthesis (Announcement, 1996). In human and other higher organisms, transcription involves binding of a specific group of proteins (known as transcription factors) to and unzipping of local regions in the double helical genomic DNA into two separate strands and synthesis of messenger RNA (mRNA) from one of them. Apart from transcription, DNA recognition by proteins is crucial in host response to certain pathogens (some pathogen DNAs can be recognized by “DNA sensors”), transfer of genetic information, cell differentiation, packaging, rearrangement, replication and repair (Wold, 1997; Dalrymple *et al.*, 2001; Wang *et al.*, 2000; Lieb *et al.*, 2001; Ofra *et al.*, 2007; Luscombe *et al.*, 2000; Walter *et al.*, 2009; Hashimoto *et al.*, 2003; Ptashne, 2005; Dillon and Dorman, 2010; Thanbichler *et al.*, 2005; Kow *et al.*, 2007; Kamashev, 2000). All such interactions proceed by the binding of proteins to (usually specific) DNA sequences and the formation of a protein-DNA complex (Luscombe *et al.*, 2000). The understanding of DNA-protein interactions is therefore extremely important to understand the fundamental processes of life.

The interaction between DNA-binding proteins and their target DNA is often accompanied by large conformational changes in either or both of them (Thompson and Landy, 1988; Paull *et al.*, 1993; Masse *et al.*, 2002; Andrabi *et al.*, 2014). These conformational changes facilitate and stabilize the formation of a protein-DNA complex, enabling the desired molecular events to take place. Different types of conformational changes have been confirmed by experimental data, e.g., an increase in superhelical density of DNA and bending and twisting of DNA are well documented (Beloin *et al.*, 2003; Tapias *et al.*, 2000; Noy *et al.*, 2016).

This review focuses on the understanding of the mechanism of protein-DNA interactions and their role in various biological processes, gives a broad overview of the nature of conformational changes in both protein and DNA in the formation of protein-DNA complex and the factors affecting its stability. We also aim to review the experimental and computational techniques to understand the structural and conformational changes in protein and DNA to form a stable complex. This review also investigates the current understanding of the role of DNA structure and dynamics in protein DNA interactions and highlights various unanswered questions on the subject.

Representative Systems of Protein-DNA Interactions

As stated above, the interactions between protein and DNA play a central role in a variety of biological processes such as transcription factors binding to DNA, DNA modifications, viral infection and chromosomal packing. Some of these interactions, such as transcription factor binding occur while the interacting sequence is still a part of a genomic DNA, which is millions of times larger, whereas other interactions may potentially be with a free and small DNA fragment in the cytoplasm (Beloin *et al.*, 2003). A protein on the other hand mostly acts as a complete unit, as a monomer or a complex formed by oligomerization on its own copies, with other co-factors or as a small number of subunits (e.g., histones) (Rouvière-Yaniv *et al.*, 1979; Luger *et al.*, 1997; Dillon and Dorman, 2010; Thanbichler *et al.*, 2005; Lia *et al.*, 2003). DNA-recognition in a genome towards gene expression requires rapid binding of modifying enzymes, polymerases, repressors and activators to their site of action on a large, packaged DNA, not always easily “visible” to the acting proteins (Gasser and Laemmli, 1987). The interaction between a protein and its “target” DNA in transcription factor binding promotes or suppresses specific genes and the corresponding proteins are called *activators* and *repressors* respectively. One of the most widely studied systems is the repressor protein responsible for switching on and off the *lac* and *lambda* phage genes of *Escherichia coli* (Matthews *et al.*, 1982; Kamashev *et al.*, 1995). The inactivation of *lac* and *lambda* repressor proteins leads to the expression of *lac* and *lambda* gene respectively. The process of binding of transcription factor is crucial, in general, for any such processes. The inappropriate binding of transcription factors results in alterations in gene expression, which has been found to be a cause of disorders (Jimenez-Sanchez *et al.*, 2001) and medical conditions in humans (Lee and Young, 2013; Farnham, 2009).

Another widely studied model system of protein-DNA interactions is related to the packing of chromosomes (Dillon and Dorman, 2010; Thanbichler *et al.*, 2005). The chromosomes consist of incredibly long DNA chains, which fold and compactify to protein-DNA complexes called chromatids (Fischle *et al.*, 2003). The DNA binding proteins involved in this process can be classified into histones and nonhistones chromosomal proteins. Histone proteins form nucleosomes which are attached by DNA string, forming a *bead-string* kind of structure at their first level of organization.

Role of protein-DNA recognition in host defense against invading pathogens is also of great significance so much so that in some cases pathogen DNA itself can act as a vaccine (Rice *et al.*, 1999; Modlin, 2000). Examples of protein-DNA interactions in host defense are CpG-rich viral DNA like simplex virus (HSV-1), HSV-2, and murine cytomegalovirus (MCMV), which are recognized by TLR9 resulting in activation of inflammatory cytokinesis and type 1 IFN secretion (Hochrein *et al.*, 2004; Krug *et al.*, 2004a,b; Lund *et al.*, 2003; Tabeta *et al.*, 2004; Akira *et al.*, 2006; Herzner *et al.*, 2015; Manders and Thomas, 2000; Dell’Oste *et al.*, 2015).

It may be noted that, although more common and widely investigated, it is not always the protein, which acts on DNA and hunts for its targets. As an example of the opposite, DNA fragments, binding to protein are known to stabilize proteins protecting them against misfolding and aggregation and thus leading to formation of a stable structure (Jolly and Morimoto, 2000; Morimoto, 1998; Wu, 1995).

DNA-Binding Protein Structures, Dynamics and Conformational Changes

Experimental techniques like X-ray crystallography and NMR have been employed to give high resolution structures of DNA-protein complexes in crystalline state and in solution respectively (Spolar and Record, 1994; Garvie and Wolberger, 2001). Many studies have shown large conformational changes in the structures of both the DNA and proteins during sequence-specific binding (Kalodimos, 2004; Kalodimos *et al.*, 2002; Spolar and Record, 1994). The conformational changes in DNA vary from smooth change in bending deformations to drastic changes resulting in disruption of base pair stacking, base flipping, sharp bends and kinks in the helical axis (Schultz *et al.*, 1991; McClarin *et al.*, 1986; Kim *et al.*, 1990; Honnappa *et al.*, 2005). Two key questions that have been investigated in details are:

"How a protein's binding nonspecifically to DNA, preceding the specific interaction, brings out a change in the structural and dynamic properties of DNA?" and "How does a complex undergo transitions from nonspecific to specific interactions".

To address these questions, Kalodimos (2004) studied *lac* repressor and categorized its DNA-binding domain on the basis of structure and dynamics in the free state and non-specifically bound state to DNA. In the study, the authors have reported the high-resolution structures of the dimeric *lac* repressor DNA-binding domain bound to an 18-base pair non-specific DNA fragment by imposing 2412 experimental restraints on multidimensional NMR spectroscopy. The authors did a comparison of the nonspecific and specific binding modes to determine the changes that occurred due to specific binding. The calculations for protein-DNA docking were performed (Brünger *et al.*, 1998) using HADDOCK setup and protocols, which are a handy set of tools for studying protein-DNA interactions *de novo* (Brünger *et al.*, 1998; Dominguez *et al.*, 2003). The protein in a non-specific complex is rotated by an additional ~ 25 degrees relative to DNA, when compared with the specific complex. Some of the residues (Tyr7, Gln18, and Arg22 in this complex), which play a key role in specific interaction, are found to undergo shifting and twisting to participate in hydrogen bonding and electrostatic interactions in the case of their corresponding non-specific complex. The interface of protein-DNA in case of non-specific complex in this system is found to be highly flexible in contrast to the rigid interface in the case of specific complex, thus implicating a role of conformational dynamics in specificity. In this system and in general, the atomic arrangements of protein-DNA complex are significantly altered in their nonspecific to specific transitions of interactions, which may be crucial in target search followed by complex formation with the genomic target DNA. In a recent study by Corona and Guo (2016), authors performed a comparative analysis of this protein-DNA complex to investigate the structural features that contributes to the binding specificity. Based on the binding specificity DNA-protein complex has been categorized as (1) Highly specific (HS) (2) multi-specific (MS) and (3) nonspecific (NS). From the analysis of conformational changes between bound and unbound states, authors found HS and MS to have larger conformational changes and flexibility in both bound and unbound states.

In another study by Velmurugu *et al.* (2016), dynamic conformational changes were studied for a DNA-repair protein radiation-sensitive 4 (Rad4; yeast XPC ortholog) recognition of the damaged site on DNA by using temperature-jump spectroscopy. One of the key finding of the study was the role of DNA deformability which is found to be the key factor that helps in a rapid search of the damaged site by DNA-repair protein. Specifically, the RAD4 recognizes the damaged site through twist-open mechanism in which twisting assists in inter-converting RAD4 from search mode to interrogation mode. The role of β -hairpin here is essential for the nucleotide flipping. The conformational changes compatible with rapid diffusion on DNA makes RAD4 stall preferentially at these sites offering time for DNA to open.

Insights into the conformational dynamics of protein-DNA interactions, not accessible to crystal structures determined by X-ray can be gleaned by NMR relaxation of backbone amides (Akke, 2002). Such studies have resulted in the understanding that most of the functional processes, such as docking, protein folding, allosteric transitions are associated with slow motion of *motor* domains (on the time scale of μ s to ms). Most of the residues that perform the motion at this time scale are found to be involved in both specific and nonspecific binding. Motion on this time scale and the flexibility of the complex facilitates the conformational transitions and the search of specific sites respectively.

To understand the effect of conformational dynamics in protein-DNA recognition Chu *et al.*, (2014) developed a two-basin structure based model to understand the dynamics of *Sulfolobus solfataricus* DNA Y-family polymerase IV (DPO4) when it binds to DNA. A two-basin structure based model (SBM) was developed with electrostatic interaction given by Debye-Huckel model. Thermodynamic and kinetic simulations were performed to investigate the conformational transitions of DPO4 during the process of binding to the target site on DNA. In the study, authors have found that DPO4 maintains multiple conformational equilibrium stages during the process of binding and the distribution of conformations vary at different stages of binding. By varying the strength of electrostatic interactions and the flexibility of the linkers, the authors showed the way DPO4 dynamically regulates the DNA recognition.

In yet another recent study using NMR spectroscopy, Desjardins *et al.* (2016) investigated the mechanism of binding of the Ets-1 transcription factor to DNA. Ets-1 belonging to ETS family of eukaryotic transcription factors is auto-inhibited by serine-rich region (SRR) and helical inhibitory module (IM). The structural and dynamic features of the Ets-1 were characterized for both specific and nonspecific binding of ETS-1 to 12 bp of oligonucleotides. On binding to both nonspecific and specific DNA, the N-terminal sequences are found to be predominantly unfolded, but still, sample ordered conformations. With specific DNA, the backbone undergoes more structural and dynamic changes and arginine/lysine side chains of the core ETS domains are much larger in comparison to non-specific binding. The study supports the general model that the Ets-1 form non-specific binding *via* electrostatic interactions and hydrogen bonding assist in the formation of well-ordered specific complexes with DNA.

Conformational changes in protein-DNA complexes are not always associated with target search as the physical cellular environments such as varying pH, temperature as well as site-specific chemical modification (phosphorylation and methylation) often lead to the formation and alteration of the oligomeric states of protein-DNA complexes (Andrabi *et al.*, 2014). Irrespective of the mechanisms involved in the conformational changes, two or more conformations of proteins in DNA-bound and unbound states may be compared to investigate the differences caused by complex formation. Based on this approach, we recently surveyed and analyzed the conformational changes in a large number of protein-DNA complex (Andrabi *et al.*, 2014). We addressed three issues of conformational changes in DNA-binding proteins: (a) types of conformational changes and their distribution among different proteins with respect to charge and dipole moments, (b) Intrinsic flexibility of unbound DNA and its role in conformational change (c) contributions of conformational change to the stability and specificity of DNA-protein complex. Six groups of conformational changes were identified and related to their biological implications. Proteins in one of these groups undergoing little conformational change “the rigid group” were found to form smaller number of contacts with DNA in comparison to proteins, which undergo large conformational change. We also reported that the degree of conformational change in protein-DNA complex increases the stability and specificity of the complex. Normal mode analysis was used to analyze the extent and direction of conformational changes in protein in some classes of DNA-binding proteins. The amount of conformational changes occurring in a protein was found to depend on the polarity of proteins. The proteins with positive charge interact mostly with the negatively charged backbone of DNA and thereby undergo least conformational change. However, for hydrophobic residues like Cys, Ala and Gly, proteins undergo large conformational changes at the interface to adjust for the binding.

Role of DNA Structure and Dynamics

Binding of proteins to its specific site of action on DNA is usually accompanied by diffusion and nonspecific binding to DNA (Berg and von Hippel, 1985). It has been proposed that the proteins undergoing three-dimensional diffusive motion in cytoplasm bind nonspecifically to DNA and then undergo one-dimensional sliding motion to reach its specific binding site on DNA (Shimamoto, 1999; Halford and Marko, 2004; Kalodimos, 2004; Esadze *et al.*, 2014). This mechanism is termed as facilitated diffusion mechanism (Berg *et al.*, 1981; Richter and Eigen, 1974) and has been proposed to justify the fast rate of binding of protein to genomic DNA (see Fig. 1 for the general process of TFs finding their targets in genomic DNA). This non-specific binding of protein and DNA is a result of electrostatic interactions between the negatively charged phosphate backbone and protein.

The specific binding of proteins to DNA involves two types of mechanisms: one is the hydrogen bond formation with specific sequence along the major groove and the other is sequence dependent deformation of DNA (Fuxreiter *et al.*, 2011; Rohs *et al.*, 2009b; Ahmad *et al.*, 2006). They are respectively termed as *sequence* and *shape* readout (earlier known as *direct* and *indirect* readout). In some protein-DNA complexes, the bending of DNA ends up with huge conformational changes in the structure of

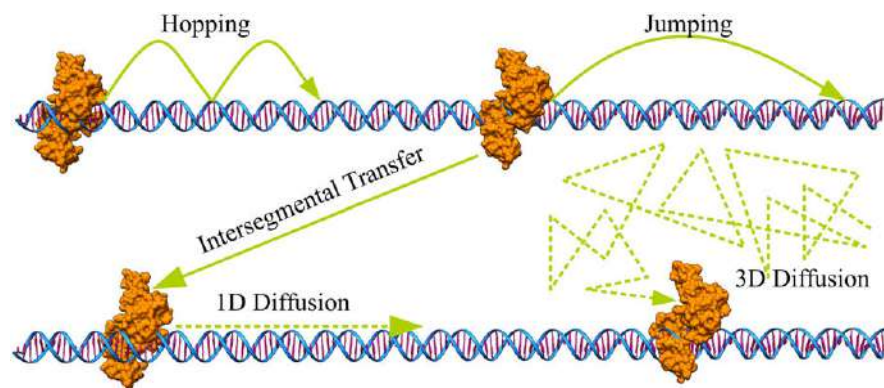


Fig. 1 Diffusion mechanisms of transcription factors trying to find their targets in genomic DNA (for simplicity genomic DNA is shown as a straight helix instead of chromatin folding): The TFs are reported to undergo electrostatically dominated non-specific interactions with DNA, which undergoes a non-specific to specific hydrogen-bond dominated interaction transition upon reaching the targets. Diffusions processes themselves during non-specific association between protein and DNA occurs through multiple events such as hopping, jumping, inter-segmental transfer and three-dimensional diffusion.

DNA and is found to deviate from B-form of double helix. The role of dynamics is not just limited to non-specific interactions but it also plays an important role in specific interactions between protein and DNA. As an example, Sox2 proteins cause a large DNA bend when they bind to their genomic targets requiring substantial interaction energy, which is often provided by partner protein-protein interactions with cofactors (Hou *et al.*, 2017; Kamachi *et al.*, 1999; Scaffidi and Bianchi, 2001). The flexible and positively charged tails of proteins assist in fine-tuning the specificity in case of transcription factors (Fuxreiter *et al.*, 2011; Rohs *et al.*, 2010).

Molecular dynamics (MD) is the principal technique to investigate the detailed molecular mechanism of protein-DNA interactions, partly because actual experimental procedures to observe the process of protein-DNA interactions are beyond the technological capabilities of researchers. There have been a number of successful efforts in applying MD to understand and interpret observed biological knowledge of protein-DNA interactions in the cellular contexts. For example, in a recent MD study of telomere repeat binding factors (TRF1 and TRF2), the dynamics of individual amino acids chains suggested that they could contribute to the recognition of more than one base pair (Etheve *et al.*, 2016; Garton and Laughton, 2013; Jolma *et al.*, 2010). The study helped to resolve conflicting experimental data (Luscombe *et al.*, 2000; Garton and Laughton, 2013). In a study by Tan *et al.* (2016) authors emphasized the dynamic aspects of nonspecific protein-DNA interactions. The dynamic coupling between protein binding to DNA and the change in the structure of DNA was investigated. Protein binding to DNA changes the DNA structure, which in turn modifies the mechanics of proteins along the DNA scaffold and thereby its interaction dynamics (Bhattacharjee and Levy, 2014a,b; Rohs *et al.*, 2009a,b; Stella *et al.*, 2010). Molecular dynamics simulations have been performed on Coarse-grained model of DNA-binding proteins to understand the dynamic couplings among protein-DNA binding, sliding of protein along DNA and bending of DNA. Simulations have been performed on bacterial architectural protein HU (Rouvière-Yaniv *et al.*, 1979) and 14 other DNA-binding proteins. At long time-scales, it has been found that the sliding motion of HU is associated with DNA bending different from *induced-fit* and *population-shift*. At shorter time scales, the HU is found to pause when the DNA is highly bent and starts to move when DNA returns to less bent structure. Thus the sliding motion is found to largely depend on the DNA bending dynamics.

Issues in Protein-DNA Interactions Predictions

The interplay between protein and DNA are ubiquitous in nature and governs several cellular processes (replication, transcription, translation, chromosomal packing, *etc.*) crucial for growth and survival of any living organism. Given its importance, the field of protein DNA interaction has been studied by many researchers with a focus on the development of tools and techniques for identification of DNA binding site in proteins, DNA binding proteins from sequence or structure and transcription factor binding site on genomic DNA. In the following sections, we review the progress made on these issues.

Prediction of DNA Binding Residues and DNA Binding Proteins

The binding of DNA to the protein is very specific where DNA binds to a particular region of the protein usually termed as the binding site, defined by the group of residues that are essential for the biological function and whose mutations can result in attenuation/loss of function. An accurate estimation of DNA-binding site and DNA-binding proteins thus is of utmost significance and can provide an insight into the various functions of the protein and also in designing novel therapeutics. Many experimental and computational methods have been developed for prediction of DNA binding site and DNA-binding proteins. Experimental methods for direct assessment of DNA-binding specificities include chromatin immunoprecipitation (ChIP) (Bardet *et al.*, 2013; Johnson *et al.*, 2007), MicroChIP (Luscombe *et al.*, 2001), Fast ChIP (Mandel-Gutfreund and Margalit, 1998), electrophoretic mobility shift assays (EMSAs) (Jones *et al.*, 1999, 2003), peptide nucleic acid (PNA)-assisted identification of RNA binding proteins (RBPs) (PAIR) (Olson *et al.*, 1998), X-ray crystallography (Orengo *et al.*, 1997) and nuclear magnetic resonance (NMR) spectroscopy (Ponting *et al.*, 1999). With the growing advancement in associated computational technologies and machine learning methods, several bioinformatics methods have also been developed for the prediction of DNA binding site complementing experimental procedures. The computational approaches for prediction of DNA binding site and DNA-binding proteins can be broadly classified into three categories: sequenced based approaches, structure-based approaches and homology modeling and threading (discussed in Section “Computational Methods for Investigating Protein-DNA Interactions”) (Si *et al.*, 2015).

Many sequence- and structure-based methods for prediction of DNA-binding site and DNA-binding proteins rely on the identification of homologous sequences/structures with known DNA-binding site information. In sequence-based methods, based on sequence alignment of the homologous sequence and query protein, DNA binding site is inferred from the homologous sequence. Structure-based methods are typically utilized when the crystal structure of the protein of interest is known. In this approach, the probable DNA binding site is inferred by comparing the binding site of the known DNA binding proteins and the query protein (Fig. 2). Homology based methods are effective when similar proteins with known binding sites are available, which is often not the case for many DNA-binding proteins. Several studies have utilized non-comparative sequence and structure-based methods for identification of DNA-binding site and DNA binding proteins. These methods try to develop an association between the *descriptors* or *features* of subsequences or substructures in a protein and implicitly model their propensity individually or cumulatively to binding DNA (Ahmad and Sarai, 2005; Yan *et al.*, 2006; Hwang *et al.*, 2007; Ofra *et al.*, 2007; Wu *et al.*, 2009; Carson *et al.*, 2010; Alibés *et al.*, 2010b; Xiong *et al.*, 2011; Li *et al.*, 2013, 2014a,b; Zhao *et al.*, 2014; Peled *et al.*, 2016). Various sequence-based features include amino acid sequence/residue type, sequence conservation, global composition of amino acids.

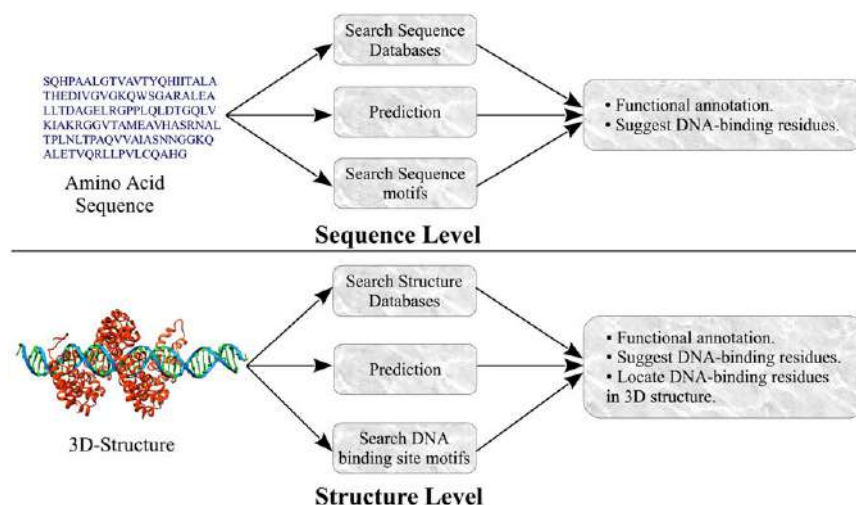


Fig. 2 Sequence and structure based predictions of protein-DNA interactions. Two groups of widely investigated problems are (a) identification of DNA-binding residues for known DBPs and (b) identifying novel candidate DBPs from a list of proteins.

Structure-based features include structural motifs, structural neighborhood, structural flexibility, secondary structure, accessible surface area (ASA), hydrophobicity, electrostatic potentials, net charge, and dipole and quadrupole moments (Si *et al.*, 2015, Ahmad and Sarai, 2004). Once the features have been identified, the next step is to develop a computational model that can find a quantitative predictability of DNA binding from these features. Among the earliest efforts, Ahmad *et al.* (2004) examined the protein DNA binding residues in terms of local amino acid composition, solvent accessible surface area, and secondary structure. Employing this knowledge a neural network model was developed to identify the DNA binding site and DNA binding proteins directly from sequence. Later on, the position-specific scoring matrix (PSSM) profiles were also incorporated into the model to improve the accuracy of the DNA-binding site prediction (Ahmad and Sarai, 2005). Utilizing the same dataset as by Ahmad *et al.*, Kuznetsov *et al.* (2006) used sequence and structure features to identify DNA binding site employing Support Vector Machine, a supervised pattern recognition method. Yan *et al.* (2006) utilized the Naive Bayes classifier and sequence based features (identity of central and neighboring amino acid residues) to predict the DNA binding residues in DNA binding proteins. Using amino acid residue properties *i.e.* molecular mass, hydrophobicity index and side chain *pka* values, Wang and Brown (2006) developed SVM classifier for identification of DNA binding site from primary sequence data. Hwang *et al.* (2007) employed machine learning techniques (kernel logistic regression, support vector machine and penalized logistic regression) for prediction of DNA binding sites from primary sequences with/without the use of PSSM profiles. Employing features derived from sequence and structure, Bhardwaj and Lu (2007) used the SVM classifier to predict DNA binding residues. Chu *et al.* (2009) used protein secondary structures for identification of DNA binding sites in transcription factors. More recently, Zhou *et al.* (2017) have utilized a novel residue encoding method termed as Position Specific Score Matrix (PSSM) Relation Transformation (PSSM-RT), to encode amino acid residues using the evolutionary information between residues and an ensemble learning classifier (EL_PSSM-RT) for prediction of DNA binding site. In a study by Yan and Kurgan (2017), authors have developed a novel method that accurately predicts DNA- and RNA-binding residues as well as DNA- and RNA-binding proteins using sequence-derived features (physicochemical and biochemical properties) and logistic regression model.

Several machine-learning approaches have also been developed for the related problem of identifying the DNA binding sites in proteins using evolutionary information. Similar to prediction of DNA binding residues, both sequence based (Cai and Lin, 2003; Yu *et al.*, 2006; Kumar *et al.*, 2007, 2009; Langlois and Lu, 2010; Huang *et al.*, 2011; Lin *et al.*, 2011) and structure-based methods (Stawiski *et al.*, 2003; Ahmad and Sarai, 2004; Bhardwaj and Lu, 2007; Szilágyi and Skolnick, 2006; Nimrod *et al.*, 2009, 2010; Zhou and Yan, 2011; Zhao *et al.*, 2014) have been developed for the identification of DNA binding proteins. Cai and Lin (2003) utilized pseudo amino acid composition and a group of nonlinear features derived from primary protein sequence to develop SVM for prediction of DNA binding proteins. Yu *et al.* (2006) utilizing sequence derived physicochemical properties and SVM proposed the binary classifications for rRNA-, RNA-, and DNA-binding proteins. Shao *et al.* (2009) developed a classifier based on SVM to differentiate between DNA/RNA binding proteins from non-nucleic acid binding proteins. Kumar *et al.* (2009) proposed a method, DNA-Prot, to determine DNA-binding proteins from protein sequence based on random forest. Later, Lin *et al.* (2011) developed a new method, iDNA-Prot, for annotating uncharacterized proteins as DNA-binding proteins or non-DNA-binding proteins. Later, in a study by Ma and Wu (2013), authors developed a new approach for identifying DNA-binding proteins using sequence-derived knowledge and support vector machine-sequential minimal optimization (SVM-SMO) algorithm. In a study by Peled *et al.* (2016), authors have utilized biophysical features and random forest to discover DNA and RNA binding proteins. In a recent study by Paz *et al.* (2016) authors have developed a non-homology based approach for prediction of DNA- and RNA-binding proteins. The method utilizes the structure-based features (electrostatic features and general properties of the protein), given the three-dimensional structure of the protein. Wei *et al.* (2017) have

developed a novel approach, Local-DPP, combining local Pse-PSSM (Pseudo Position-Specific Scoring Matrix) with random forest classifier to predict DNA-binding proteins.

Prediction of TF Binding Sites

In the preceding section, we have discussed the identification of DNA-binding site on protein; however, in protein-DNA interaction, DNA also contains specific sites, which are recognized by the protein. These transcription factors (TF), the proteins which bind at specific sites on genomic DNA regulate the process of transcription. The presence of transcription factor on the DNA will attract or impede RNA polymerase, thereby promoting or repressing gene expression, respectively. To gain insight into the overall picture of the functioning genome, it is crucial to determine the genomic positions to which transcription factors bind. Transcription Factor Binding Sites (TFBSs) prediction is one of the well-studied domains in the area of protein-DNA interaction and continues to progress rapidly, as the subject moves from the knowledge of only a few binding events to genome-wide data sets. Various experimental and computational methods have been developed to characterize the binding site of TFs (reviewed in [Xie *et al.*, 2011](#); [Bulyk, 2003](#)). One of the simplest computational approaches, used for the identification of TFBSs is searching for the over-represented nucleotide sequences using motif-detection algorithms. But, since the binding sites of transcription factors are short nucleotide sequences and are tolerant to sequence variations, plus this approach (motif identification) requires a large number of sequences for pattern identification, this approach is severely limited by low accuracy. Sequence-based computational methods for prediction of transcription factor binding sites (TFBSs) are often based on Position Weight Matrices (PWMs), which reflects the degeneracy observed TFBSs among the binding motifs corresponding to a particular transcription factor. PWMs have been utilized by various computational methods to identify the binding site of TFs (reviewed in [Bulyk, 2003](#)). In a recent study [Andrabi *et al.* \(2017\)](#) have shown that the information about the TF binding lies not only at specific binding positions but also extends as far as 200 nt away from the TFBSs.

Another successful approach employed for the prediction of TFBSs is structure-based prediction of TFBSs, which makes use of the available three-dimensional structure of protein-DNA complexes. Structure-based prediction of TFBSs has numerous advantages over sequence-based methods. For example, they can also describe the biophysics of interaction, not possible from a simple sequence-level analysis. However, there are several issues in structure-based prediction of TFBSs. One of the crucial issues is the choice of the scoring function used for estimating the binding energy or affinity of the protein-DNA complex. Two types of scoring functions are usually utilized, physics-based molecular mechanics force fields and the knowledge-based statistical potentials. Molecular mechanics energy function comprises of physicochemical interactions which include van der Waals (VDW) interaction, electrostatic interactions, solvation energy, *etc.* ([Liu and Bradley, 2012](#)). Molecular mechanics energy functions are based on approximations and assume fixed charges. They have been successfully utilized in investigating protein-DNA interactions ([Havranek *et al.*, 2004](#); [Morozov *et al.*, 2005](#); [Siggers and Honig, 2007](#); [Alibés *et al.*, 2010a](#)). Besides the electrostatic and VDW interactions, the physics-based energy function also takes into account the hydrogen bonds, π -cation, and π - π interactions, on the thought that these interactions play a significant role in the stability of the protein-ligand complex. Although the molecular mechanics energy functions can comprehensively illustrate the protein-DNA interactions, they are typically time-consuming. Structure-based methods only consider few snapshots for the prediction of TFBSs, which represents a very small fraction of possible conformations of protein-DNA complexes and because molecular mechanics energy functions are sensitive to conformational alterations, these may result in incorrect identification of TFBSs.

Knowledge-based statistical potentials are deduced from the statistical study of known protein-DNA complexes. They are frequently utilized because they are computationally inexpensive and are comparatively less susceptible to conformational transitions than physics-based molecular mechanics energy functions. Knowledge-based statistical potentials ranges in resolution from atom-based potentials ([Zhang *et al.*, 2005](#); [Donald *et al.*, 2007](#); [Robertson and Varani, 2007](#)) to residue-based potentials ([Mandel-Gutfreund and Margalit, 1998](#); [Aloy *et al.*, 1998](#); [Liu, 2005a,b](#); [Takeda *et al.*, 2013](#)). Recent studies have shown that residue-based statistical potentials also work well in the case of protein-DNA interaction ([Liu, 2005a,b](#); [Takeda *et al.*, 2013](#)). Knowledge-based statistical potentials also differ in terms of their applicability to distance ranges e.g., distance-independent ([Mandel-Gutfreund and Margalit, 1998](#); [Aloy *et al.*, 1998](#)) and distance dependent functions ([Liu, 2005a,b](#); [Zhang *et al.*, 2005](#); [Robertson and Varani, 2007](#); [Takeda *et al.*, 2013](#)). Knowledge-based statistical potentials are limited by two factors. First being the mean force makeup of the statistical potentials. For example, lysine and arginine make specific and nonspecific interactions with DNA in the form of hydrogen bonds and electrostatic interactions. Though knowledge-based statistical potentials could capture the hydrogen bonds implicitly, they are averaged out with the nonspecific interactions. The other originates due to low frequency problem. Recent studies have shown that the interaction between the DNA and aromatic amino acids are abundant; however, very limited knowledge is available about these specific interactions ([Wilson *et al.*, 2014](#); [Wilson and Wetmore, 2015](#)). [Corona and Guo \(2016\)](#), through comparative analysis have shown that the interaction between the DNA bases and amino acid residues Tyr and His are enriched in highly specific DNA-binding proteins and might contribute directly to the TF binding specificity ([Farrel *et al.*, 2016](#)). Some studies have also shown that combining knowledge-based potential with other interactions like hydrogen bond and π interaction improves the accuracy of the TFBSs prediction when compared with atomic-level and residue-level knowledge-based potentials.

Experimental Methods to Investigate Protein-DNA Interactions

In view of various essential roles played by protein-DNA interactions, various experimental methods have evolved over time to elucidate them. Apart from constantly improving technologies to analyze interactions in greater detail and accuracy, some method

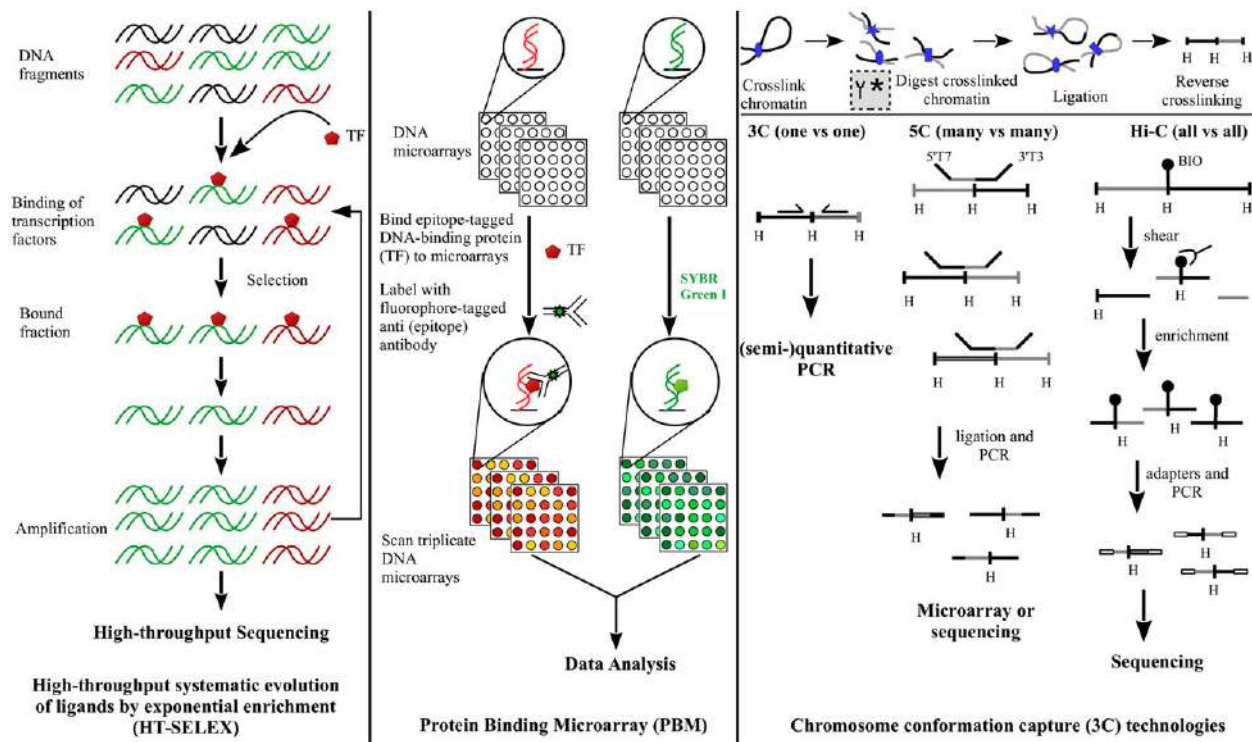


Fig. 3 Representative experimental techniques to study protein-DNA interactions: (a) HT-SELEX in which TF is allowed to bind a diverse set of DNA sequences, which are picked up using an antibody and then the sequencing of attached DNA is carried out to infer binding patterns (b) Protein-binding Microarray use the technique of hybridization with pre-selected DNA oligomers. A relatively long sequence can be used to interrogate interactions with many sub-sequences and Bioinformatics approaches help in designing an optimum chip and (c) Chromatin conformation capture (3C) and their variants which can reveal the larger-scale contact maps between different parts of genomic DNA as a result of nucleosome formation.

serve specific purposes and have their own trade-offs. In this section, we will discuss popularly used experimental methods to interrogate protein-DNA interactions. Overall, experimental techniques can be grouped into (a) *In vitro* binding experiments (b) NGS-based methods and (c) other methods. **Fig. 3** gives the schematic representation of the experimental techniques used in investigating protein-DNA interactions. They include protein-binding microarrays, high throughput systematic evolution of ligands by exponential enrichment (HT-SELEX) and Chromosome Conformational Capture (3C). They are briefly reviewed in the following:

***In Vitro* Binding Experiments**

Various *in vitro* techniques popularly used to assay protein-DNA interactions are Protein binding microarray (PBM), High-throughput systematic evolution of ligands by exponential enrichment (HT-SELEX) (Jolma *et al.*, 2010), High-throughput sequencing-fluorescent ligand interaction profiling (HiTS-FLIP), Mechanically induced trapping of molecular interactions (MITOMI), and DIP-chip and DIP-seq (DNA immunoprecipitation followed by microarray (DIP-chip) and DIP followed by high-throughput sequencing (DIP-seq)). All these methods allow high-throughput characterization of DNA-binding proteins. A brief description of these methods is provided below.

Protein binding microarray (PBM)

PBM is a DNA microarray-based technology to ascertain the binding preferences of a transcription factor towards DNA. In this technique, first, the DNA-binding protein is expressed, purified and then is allowed to bind directly to a DNA microarray. The array is then tagged with a fluorescent antibody, which is further utilized to measure the relative amount of protein bound to each probe. The probes in the DNA microarray often consist of large number of possible k-mers. Unique k-mers needed to represent much larger set of subsequences can be designed using *de Bruijn* algorithm (Vassallo and Ralston, 1992).

High-throughput systematic evolution of ligands by exponential enrichment (HT-SELEX)

In this technique, a double-stranded DNA mixture is incubated with a DNA-binding protein immobilized onto a well-plate. The mixture is then washed, and the bound oligonucleotides are retrieved, amplified by PCR and sequenced to identify the TF binding specificities (Ogawa and Biggin, 2012).

High-throughput sequencing–fluorescent ligand interaction profiling (HiTS–FLIP)

It is a high throughput technology to perform DNA sequencing and systematic protein-DNA binding experiment on the flowcell. This allows determining the DNA-binding specificities of a transcription factor (Nutiu *et al.*, 2011).

Mechanically induced trapping of molecular interactions (MITOMI)

It is a high-throughput microfluidic platform widely used to study protein-DNA interactions. In this method, a microfluidic equipment is aligned to a microarray containing programmed DNA, such that the TFs (located onto the surface) can bind to the DNA (Maerkl and Quake, 2007). The equipment allows mechanical trapping of the protein-DNA complexes, thus protecting them from being washed out. The device is then examined to quantify binding of TFs.

DNA immunoprecipitation followed by microarray (DIP–chip) and DIP followed by high-throughput sequencing (DIP–seq)

It is a high throughput method for identifying DNA regions that bind to DNA-binding proteins or transcription factors on a genome-wide scale. In this technique, DNA-binding protein is allowed to bind with genomic DNA, and the bound complexes are then segregated using immunoprecipitation or affinity purification (Gossett and Lieb, 2008; Liu, 2005a,b). The DNA fragments are then purified, amplified, fluorescently labeled and characterized either by hybridization to a DNA microarray or high-throughput sequencing.

NGS-Based Methods

ChIP-Seq and DNase-Seq

With the advancement in the next generation sequencing technologies, ChIP-Seq and DNase-Seq have become the preferred choice of *in vivo* techniques to elucidate the protein-DNA interactions (Meyer and Liu, 2014). Both these technologies estimate the binding occupancy of a DNA-binding protein to the nucleotide sequences.

In ChIP-Seq, the cells are first treated with chemical crosslinks which allows cross linking (covalent binding) of the proteins to each other and then to the DNA. Once the protein-DNA are crosslinked, the chromatin is extracted and DNA is sheared using sonication and the bound complex is isolated utilizing antibodies specific to the protein of interest. The cross-links are then reversed and purified DNA segments are characterized using sequencing.

DNase-seq is a more recent technology developed to obtain the genome-wide protein-DNA binding map. In DNase-seq, protein-DNA complexes are digested with DNase I which preferentially digests the DNA not bound to proteins. The DNA fragments are then extracted, purified and sequenced.

DNase-Seq can be used to interrogate binding sites of large number of TFs in a single experiment. However, the challenge is to assign the identified site to their corresponding TF and powerful Bioinformatics techniques have been developed to address this issue exclusively (Rajagopal *et al.*, 2016).

Conformation capture by 3C, 5C and HiC

The topological structures of the genomic DNA play a significant role in coordinating transcription and other DNA-dependent metabolic processes and thus are of great importance. Chromosome conformation capture (3C) technology developed by Dekker *et al.* (2002) are one of the recent technologies which allow study of chromatin structure in their native cellular environment with high resolution. All 3C and 3C-derived technologies (4C (chromosome conformation capture-on-chip), 5C (chromosome conformation capture carbon copy), HiC, ChIP-loop and ChIA-PET) follow a similar set of protocols to study the 3D organization of the genomic DNA. In 3C methods, first, the genomic DNA is fixed, utilizing fixative agents like formaldehyde (Dekker *et al.*, 2002). Then the chromatin is cut using restriction enzymes and the topologically proximal regions are re-ligated to allow intramolecular crosslinking. The fragments close to each other ligate, thus quantifying the proximity of fragments. The primary difference between 3C methods lies in the fragments of interest. In 3C, also referred as “one-vs-one”, the two specific regions are probed for interaction, which is usually known *a priori*. In 5C, also known as “many-vs-many”, all fragments in a given region are probed for interaction between them. Hi-C (all-vs-all) is an unbiased technique to identify interaction between all potential fragments.

Other Methods

Apart from the techniques discussed above, a plethora of other *in-vitro* and *in-vivo* techniques are also available to interrogate protein-DNA interactions. They include Electrophoretic mobility shift assay (EMSA) (Hellman and Fried, 2007), Yeast one-hybrid assay (Y1H) (Reece-Hoyes and Marian Walhout, 2012), Proximity ligation assay (PLA) (Gustafsdottir *et al.*, 2007), X-ChIP, Fast ChIP, Carrier ChIP, Native-ChIP (N-ChIP), Matrix ChIP, ChIP-Chip (Tong and Falk, 2009), ChIP display (Barski, 2004), other variations of ChIP such as Quick and Quantitative ChIP (Q² ChIP) and MicroChIP, ChIP-loop and ChIA-PET (Li *et al.*, 2014a,b; de Wit and de Laat, 2012), for which the readers are directed to relevant literature.

Computational Methods for Investigating Protein-DNA Interactions

As discussed above, protein-DNA interactions occur in complex and diverse cellular conditions, which are difficult to observe in their native form. Crystal structures do give us a closer look at the atomic level determinants of molecular interactions. However, on the one hand, many protein structures remain unsolved and on the other, it remains unknown if the crystal structures solved *in vitro* represent real cellular context. Some of our data suggests that the three dimensional structures of protein-DNA complexes are cell-type dependent (Andrabi *et al.* Bioarxiv 2015). To address the first of these two issues efforts are made to model protein structures using comparative modeling and predictive models and for the second first principle computations have been attempted. Some of the specific cases have been discussed above. From a technical perspective, the techniques used for modeling DNA-binding proteins are often based on homology modeling and threading (Morozov *et al.*, 2005). A high quality template is an essential requirement for homology modeling and in the absence of a similar protein structure available, data driven approach such as machine learning are a handy choice. For the second issue, *ab initio* physics based techniques can try to emulate the cellular contexts and efforts are going on to bring such techniques as close to the experimental conditions as possible. Homology modeling, Molecular dynamics (MD), Monte Carlo (MC) and machine learning (ML) principles applied to model protein-DNA interactions are not unique to them but are essentially an adaptation from other applications. The technical considerations behind the application of MD, MC and ML techniques have been frequently reviewed and the same can be referred to elsewhere (Tuckerman and Martyna, 2000; Ding *et al.*, 2010; Paquet and Viktor, 2015).

Summary and Conclusions

Protein-DNA interactions are essential components of all biological systems, fundamental to almost all biological processes. Tremendous efforts have been made to understand them employing (a) first principle science such as MD and MC (b) experimental techniques such as ChIP-Seq and HiC and (c) data driven approaches such as ML and statistics. It is not possible to predict DNA-binding sites from sequence with high confidence build homology based models for many novel proteins as well as direct query the nature of genome-wide interactions between TFs and their target DNAs. The studies have so far revealed the role of sequence and structure/shape of the protein and DNA sequences in the recognition process. The current trends are to look at the DNA topology in genome-wide recognition, cell-type specific structures and role for non-binding flanking regions of DNA playing an allosteric role in protein-DNA interactions. Data driven techniques are also going to be a crucial tool to interrogate such large-scale interactions, due to the availability of unprecedented amounts of data emerging from NGS-based techniques under many phenotypic contexts. Even though, no universal recognition code of DNA is yet described, more and more insights are emerging and controlling protein-DNA interactions will become easier with the growing knowhow.

See also: Chromatin: A Semi-Structured Polymer. Computational Prediction of Nucleic Acid Binding Residues from Sequence. Computational Systems Biology. Genome-Wide Scanning of Gene Expression. Natural Language Processing Approaches in Bioinformatics. Nucleic-Acid Structure Database. Prediction of Protein-Binding Sites in DNA Sequences. Protein-Protein Interaction Databases. Regulation of Gene Expression. Sequence Analysis. Transcription Factor Databases

References

- Ahmad, S., Gromiha, M.M., Sarai, A., 2004. Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics* 20, 477–486. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/14990443>.
- Ahmad, S., Kono, H., Araúzo-Bravo, M.J., Sarai, A., 2006. ReadOut: Structure-based calculation of direct and indirect readout energies and specificities for protein-DNA recognition. *Nucleic Acids Res.* 34, W124–W127. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/16844974>.
- Ahmad, S., Sarai, A., 2004. Moment-based prediction of DNA-binding proteins. *J. Mol. Biol.* 341, 65–71. Available at: <http://www.sciencedirect.com/science/article/pii/S0022283604006382>.
- Ahmad, S., Sarai, A., 2005. PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinformatics* 6, 33. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=550660&tool=pmcentrez&rendertype=abstract>.
- Akira, S., Uematsu, S., Takeuchi, O., 2006. Pathogen recognition and innate immunity. *Cell* 124, 783–801. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/16497588>.
- Akke, M., 2002. NMR methods for characterizing microsecond to millisecond dynamics in recognition and catalysis. *Curr. Opin. Struct. Biol.* 12, 642–647. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12464317>.
- Alibés, A., Nadra, A.D., De Masi, F., *et al.*, 2010a. Using protein design algorithms to understand the molecular basis of disease caused by protein-DNA interactions: The Pax6 example. *Nucleic Acids Res.* 38, 7422–7431. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20685816>.
- Alibés, A., Serrano, L., Nadra, A.D., 2010b. Structure-based DNA-binding prediction and design. In: Mackay, J.P., Segal, D.J. (Eds.), *Engineered zinc finger proteins: Methods and protocols*. Totowa, NJ: Humana Press, pp. 77–88. Available at: https://doi.org/10.1007/978-1-60761-753-2_4.
- Aloy, P., Moont, G., Gabb, H.A., *et al.*, 1998. Modelling repressor proteins docking to DNA. *Proteins* 33, 535–549.
- Andrabi, M., Hutchins, A.P., Miranda-Saavedra, D., *et al.*, 2017. Predicting conformational ensembles and genome-wide transcription factor binding sites from DNA sequences. *Sci. Rep.* 7, 4071. Available at: <http://www.nature.com/articles/s41598-017-03199-6>.
- Andrabi, M., Mizuguchi, K., Ahmad, S., 2014. Conformational changes in DNA-binding proteins: Relationships with precomplex features and contributions to specificity and stability. *Proteins Struct. Funct. Bioinform.* 82, 841–857. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24265157>.
- Announcement, 1996. *Mol. Cell. Biochem.* 159, 170.

- Bardet, A.F., Steinmann, J., Bafna, S., *et al.*, 2013. Identification of transcription factor binding sites from ChIP-seq data at high resolution. *Bioinformatics* 29, 2705–2713.
- Barski, A., 2004. ChIP Display: Novel method for identification of genomic targets of transcription factors. *Nucleic Acids Res.*, 32, e104–e104. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gnh097>.
- Beloin, C., Jeusset, J., Révet, B., *et al.*, 2003. Contribution of DNA conformation and topology in right-handed DNA wrapping by the bacillus subtilis LrpC protein. *J. Biol. Chem.* 278, 5333–5342. Available at: <http://www.jbc.org/lookup/doi/10.1074/jbc.M207489200>.
- Berg, O.G., von Hippel, P.H., 1985. Diffusion-controlled macromolecular interactions. *Annu. Rev. Biophys. Biophys. Chem.* 14, 131–160. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/3890878>.
- Berg, O.G., Winter, R.B., von Hippel, P.H., 1981. Diffusion-driven mechanisms of protein translocation on nucleic acids. 1. Models and theory. *Biochemistry* 20, 6929–6948. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/7317363>.
- Bhardwaj, N., Lu, H., 2007. Residue-level prediction of DNA-binding sites and its application on DNA-binding protein predictions. *FEBS Lett.* 581, 1058–1066. Available at: <http://www.sciencedirect.com/science/article/pii/S0014579307001263>.
- Bhattacharjee, A., Levy, Y., 2014a. Search by proteins for their DNA target site: 1. The effect of DNA conformation on protein sliding. *Nucleic Acids Res.* 42, 12404–12414. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25324308>.
- Bhattacharjee, A., Levy, Y., 2014b. Search by proteins for their DNA target site: 2. The effect of DNA conformation on the dynamics of multidomain proteins. *Nucleic Acids Res.* 42, 12415–12424. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gku933>.
- Brünger, A.T., Adams, P.D., Clore, G.M., *et al.*, 1998. Crystallography & NMR system: A new software suite for macromolecular structure determination. *Acta Crystallogr. D: Biol. Crystallogr.* 54, 905–921. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9757107>.
- Bulyk, M.L., 2003. Computational prediction of transcription-factor binding site locations. *Genome Biol.* 5, 201. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/14709165%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC395725>.
- Cai, Y., Lin, S.L., 2003. Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence. *Biochim. Biophys. Acta* 1648, 127–133. Available at: <http://www.sciencedirect.com/science/article/pii/S1570963903001122>.
- Carson, M.B., Langlois, R., Lu, H., 2010. NAPS: A residue-level nucleic acid-binding prediction server. *Nucleic Acids Res.* 38, W431–W435. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20478832>.
- Chu, W.-Y., Huang, Y.-F., Huang, C.-C., *et al.*, 2009. ProteDNA: A sequence-based predictor of sequence-specific DNA-binding residues in transcription factors. *Nucleic Acids Res.* 37, W396–W401. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/19483101>.
- Chu, X., Liu, F., Maxwell, B.A., *et al.*, 2014. Dynamic conformational change regulates the protein-DNA recognition: An investigation on binding of a Y-family polymerase to its target DNA. *PLOS Comput. Biol.* 10, e1003804. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25188490>.
- Corona, R.I., Guo, J.-T., 2016. Statistical analysis of structural determinants for protein-DNA-binding specificity. *Proteins* 84, 1147–1161. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24655651>.
- Dalrymple, B.P., Kongsuwan, K., Wijffels, G., Dixon, N.E., Jennings, P.A., 2001. A universal protein-protein interaction motif in the eubacterial DNA replication and repair systems. *Proc. Natl. Acad. Sci.* 98, 11627–11632. Available at: <http://www.pnas.org/cgi/doi/10.1073/pnas.191384398>.
- Dekker, J., Rippe, K., Dekker, M., Kleckner, N., 2002. Capturing chromosome conformation. *Science* 295, 1306–1311. Available at: <http://www.sciencemag.org/cgi/doi/10.1126/science.1067799>.
- Dell'Oste, V., Gatti, D., Giorgio, A.G., *et al.*, 2015. The interferon-inducible DNA-sensor protein IFI16: A key player in the antiviral response. *New Microbiol.* 38, 5–20. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25742143>.
- Desjardins, G., Okon, M., Graves, B.J., McIntosh, L.P., 2016. Conformational dynamics and the binding of specific and nonspecific DNA by the autoinhibited transcription factor Ets-1. *Biochemistry* 55, 4105–4118. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27362745>.
- de Wit, E., de Laat, W., 2012. A decade of 3C technologies: Insights into nuclear organization. *Genes Dev.* 26, 11–24. Available at: <http://genesdev.cshlp.org/cgi/doi/10.1101/gad.179804.111>.
- Dillon, S.C., Dorman, C.J., 2010. Bacterial nucleoid-associated proteins, nucleoid structure and gene expression. *Nat. Rev. Microbiol.* 8, 185–195. Available at: <http://www.nature.com/doi/10.1038/nrmicro2261>.
- Ding, X.-M., Pan, X.-Y., Xu, C., Shen, H.-B., 2010. Computational prediction of DNA-protein interactions: A review. *Curr. Comput. Aided-Drug Des.* 6, 197–206. Available at: <http://www.eurekaselect.com/openurl/content.php?genre=article&issn=1573-4099&volume=6&issue=3&spage=197>.
- Dominguez, C., Boelens, R., Bonvin, A.M.J.J., 2003. HADDOCK: A protein–protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* 125, 1731–1737. Available at: <http://pubs.acs.org/doi/abs/10.1021/ja026939x>.
- Donald, J.E., Chen, W.W., Shakhnovich, E.I., 2007. Energetics of protein–DNA interactions. *Nucleic Acids Res.* 35, 1039–1047. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkl1103>.
- Esadze, A., Kemme, C.A., Kolomeisky, A.B., Iwahara, J., 2014. Positive and negative impacts of nonspecific sites during target location by a sequence-specific DNA-binding protein: Origin of the optimal search at physiological ionic strength. *Nucleic Acids Res.* 42, 7039–7046. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24838572>.
- Etheve, L., Martin, J., Lavery, R., 2016. Protein–DNA interfaces: A molecular dynamics analysis of time-dependent recognition processes for three transcription factors. *Nucleic Acids Res.* gkw841. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw841>.
- Farnham, P.J., 2009. Insights from genomic profiling of transcription factors. *Nat. Rev. Genet.* 10, 605–616. Available at: <http://www.nature.com/doi/10.1038/nrg2636>.
- Farrel, A., Murphy, J., Guo, J., 2016. Structure-based prediction of transcription factor binding specificity using an integrative energy function. *Bioinformatics* 32, i306–i313. Available at: <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btw264>.
- Fischle, W., Wang, Y., Allis, C.D., 2003. Histone and chromatin cross-talk. *Curr. Opin. Cell Biol.* 15, 172–183. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0955067403000139>.
- Fuxreiter, M., Simon, I., Bondos, S., 2011. Dynamic protein-DNA recognition: Beyond what can be seen. *Trends Biochem. Sci.* 36, 415–423. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/21620710>.
- Garton, M., Laughton, C., 2013. A comprehensive model for the recognition of human telomeres by TRF1. *J. Mol. Biol.* 425, 2910–2921. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23702294>.
- Garvie, C.W., Wolberger, C., 2001. Recognition of specific DNA sequences. *Mol. Cell* 8, 937–946. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/8303294>.
- Gasser, S.M., Laemmli, U.K., 1987. A glimpse at chromosomal order. *Trends Genet.* 3, 16–22. Available at: [https://doi.org/10.1016/0168-9525\(87\)90156-9](https://doi.org/10.1016/0168-9525(87)90156-9).
- Gossett, A.J., Lieb, J.D., 2008. DNA Immunoprecipitation (DIP) for the determination of DNA-binding specificity. *Cold Spring Harb. Protoc.* 2008.pdb.prot4972–prot4972. Available at: <http://www.cshprotocols.org/cgi/doi/10.1101/pdb.prot4972>.
- Gustafsdottir, S.M., Schlingemann, J., Rada-Iglesias, A., *et al.*, 2007. In vitro analysis of DNA-protein interactions by proximity ligation. *Proc. Natl. Acad. Sci.* 104, 3067–3072. Available at: <http://www.pnas.org/cgi/doi/10.1073/pnas.0611229104>.
- Halford, S.E., Marko, J.F., 2004. How do site-specific DNA-binding proteins find their targets? *Nucleic Acids Res.* 32, 3040–3052. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10336412>.
- Hashimoto, M., Imhoff, B., Ali, M.M., Kow, Y.W., 2003. HU protein of Escherichia coli has a role in the repair of closely opposed lesions in DNA. *J. Biol. Chem.* 278, 28501–28507. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12748168>.
- Havranek, J.J., Duarte, C.M., Baker, D., 2004. A simple physical model for the prediction and design of protein-DNA interactions. *J. Mol. Biol.* 344, 59–70. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15504402>.

- Hellman, L.M., Fried, M.G., 2007. Electrophoretic mobility shift assay (EMSA) for detecting protein–nucleic acid interactions. *Nat. Protoc.* 2, 1849–1861. Available at: <http://www.nature.com/nprot/journal/v2/n8/abs/nprot.2007.249.html>.
- Herzner, A.-M., Hagmann, C.A., Goldeck, M., *et al.*, 2015. Sequence-specific activation of the DNA sensor cGAS by Y-form DNA structures as found in primary HIV-1 cDNA. *Nat. Immunol.* 16, 1025–1033. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/16497588>.
- Hochrein, H., Schlatter, B., O’Keeffe, M., *et al.*, 2004. Herpes simplex virus type-1 induces IFN- α production via Toll-like receptor 9-dependent and -independent pathways. *Proc. Natl. Acad. Sci. USA* 101, 11416–11421. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15272082>.
- Honnappa, S., John, C.M., Kostrewa, D., Winkler, F.K., Steinmetz, M.O., 2005. Structural insights into the EB1-APC interaction. *EMBO J.* 24, 261–269. Available at: <http://emboj.embopress.org/cgi/doi/10.1038/sj.emboj.7600529>.
- Hou, L., Srivastava, Y., Jauch, R., 2017. Molecular basis for the genome engagement by Sox proteins. *Semin. Cell Dev. Biol.* 63, 2–12. Available at: <https://doi.org/10.1016/j.semcdb.2016.08.005>.
- Huang, H.-L., Lin, I.-C., Liou, Y.-F., *et al.*, 2011. Predicting and analyzing DNA-binding domains using a systematic approach to identifying a set of informative physicochemical and biochemical properties. *BMC Bioinformatics* 12 (Suppl 1), S47. Available at: <http://www.biomedcentral.com/1471-2105/12/S1/S47>.
- Hwang, S., Gou, Z., Kuznetsov, I.B., 2007. DP-Bind: A web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics* 23, 634–636. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17237068>.
- Jimenez-Sanchez, G., Childs, B., Valle, D., 2001. Human disease genes. *Nature* 409, 853–855. Available at: <http://www.nature.com/doi/10.1038/35057050>.
- Johnson, D.S., Mortazavi, A., Myers, R.M., Wold, B., 2007. Genome-wide mapping of in vivo protein-DNA interactions. *Science* 316 (80-), 1497–1502. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17540862>.
- Jolly, C., Morimoto, R.I., 2000. Role of the heat shock response and molecular chaperones in oncogenesis and cell death. *J. Natl. Cancer Inst.* 92, 1564–1572. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11018092>.
- Jolma, A., Kivioja, T., Toivonen, J., *et al.*, 2010. Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome Res.* 20, 861–873.
- Jones, S., Barker, J.A., Nobeli, I., Thornton, J.M., 2003. Using structural motif templates to identify proteins with DNA binding function. *Nucleic Acids Res.* 31, 2811–2823. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12771208>.
- Jones, S., van Heyningen, P., Berman, H.M., Thornton, J.M., 1999. Protein-DNA interactions: A structural analysis. *J. Mol. Biol.* 287, 877–896. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10222198>.
- Kalodimos, C.G., 2004. Structure and flexibility adaptation in nonspecific and specific protein-DNA complexes. *Science* 305 (80-), 386–389. Available at: <http://www.sciencemag.org/cgi/doi/10.1126/science.1097064>.
- Kalodimos, C.G., Bonvin, A.M.J.J., Salinas, R.K., *et al.*, 2002. Plasticity in protein-DNA recognition: Lac repressor interacts with its natural operator O1 through alternative conformations of its DNA-binding domain. *EMBO J.* 21, 2866–2876. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12065400>.
- Kamachi, Y., Cheah, K.S., Kondoh, H., 1999. Mechanism of regulatory target selection by the SOX high-mobility-group domain proteins as revealed by comparison of SOX1/2/3 and SOX9. *Mol. Cell. Biol.* 19, 107–120. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9858536>.
- Karnashev, D., 2000. The histone-like protein HU binds specifically to DNA recombination and repair intermediates. *EMBO J.* 19, 6527–6535. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=305869&tool=pmcentrez&rendertype=abstract%5Chttp://emboj.embopress.org/content/19/23/6527.abstract>.
- Karnashev, D.E., Esipova, N.G., Ebralidze, K.K., Mirzabekov, A.D., 1995. Mechanism of Lac repressor switch-off: Orientation of the Lac repressor DNA-binding domain is reversed upon inducer binding. *FEBS Lett.* 375, 27–30. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/7498473>.
- Kim, Y.C., Grable, J.C., Love, R., Greene, P.J., Rosenberg, J.M., 1990. Refinement of Eco RI endonuclease crystal structure: A revised protein chain tracing. *Science* 249, 1307–1309. Available at: <http://www.sciencemag.org/cgi/doi/10.1126/science.2399465>.
- Kow, Y.W., Imhoff, B., Weiss, B., *et al.*, 2007. Escherichia coli HU protein has a role in the repair of abasic sites in DNA. *Nucleic Acids Res.* 35, 6672–6680. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17916578>.
- Krug, A., French, A.R., Barchet, W., *et al.*, 2004a. TLR9-dependent recognition of MCMV by IPC and DC generates coordinated cytokine responses that activate antiviral NK cell function. *Immunity* 21, 107–119. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15345224>.
- Krug, A., Luker, G.D., Barchet, W., *et al.*, 2004b. Herpes simplex virus type 1 activates murine natural interferon-producing cells through toll-like receptor 9. *Blood* 103, 1433–1437. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/14563635>.
- Kumar, K.K., Pugalenth, G., Suganthan, P.N., 2009. DNA-Prot: Identification of DNA binding proteins from protein sequence information using random forest. *J. Biomol. Struct. Dyn.* 26, 679–686. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/19385697>.
- Kumar, M., Gromiha, M.M., Raghava, G.P.S., 2007. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. *BMC Bioinformatics* 8, 463. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18042272>.
- Kuznetsov, I.B., Gou, Z., Li, R., Hwang, S., 2006. Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins* 64, 19–27. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17705269>.
- Langlois, R.E., Lu, H., 2010. Boosting the prediction and understanding of DNA-binding domains from sequence. *Nucleic Acids Res.* 38, 3149–3158. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20156993>.
- Lee, T.I., Young, R.A., 2013. Transcriptional regulation and its misregulation in disease. *Cell* 152, 1237–1251. Available at: <https://doi.org/10.1016/j.cell.2013.02.014>.
- Lia, G., Bensimon, D., Croquette, V., *et al.*, 2003. Supercoiling and denaturation in Gal repressor/heat unstable nucleoid protein (HU)-mediated DNA looping. *Proc. Natl. Acad. Sci.* 100, 11373–11377. Available at: <http://www.pnas.org/cgi/doi/10.1073/pnas.2034851100>.
- Li, B.-Q., Feng, K.-Y., Ding, J., Cai, Y.-D., 2014a. Predicting DNA-binding sites of proteins based on sequential and 3D structural information. *Mol. Genet. Genomics* 289, 489–499. Available at: <http://link.springer.com/10.1007/s00438-014-0812-x>.
- Lieb, J.D., Liu, X., Botstein, D., Brown, P.O., 2001. Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nat. Genet.* 28, 327–334. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11455386>.
- Li, G., Cai, L., Chang, H., *et al.*, 2014b. Chromatin interaction analysis with paired-end tag (ChIA-PET) sequencing technology and application. *BMC Genomics* 15.S11. Available at: <http://bmcbgenomics.biomedcentral.com/articles/10.1186/1471-2164-15-S12-S11>.
- Lin, W.-Z., Fang, J.-A., Xiao, X., Chou, K.-C., 2011. iDNA-Prot: Identification of DNA binding proteins using random forest with grey model. *PLoS One* 6, e24756. Available at: <http://dx.plos.org/10.1371/journal.pone.0024756>.
- Li, T., Li, Q.-Z., Liu, S., *et al.*, 2013. PreDNA: Accurate prediction of DNA-binding sites in proteins by integrating sequence and geometric structure information. *Bioinformatics* 29, 678–685. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23335013>.
- Liu, L.A., Bradley, P., 2012. Atomistic modeling of protein–DNA interaction specificity: Progress and applications. *Curr. Opin. Struct. Biol.* 22, 397–405. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0959440X12000991>.
- Liu, X., 2005a. DIP-chip: Rapid and accurate determination of DNA-binding specificity. *Genome Res.* 15, 421–427. Available at: <http://www.genome.org/cgi/doi/10.1101/gr.3256505>.
- Liu, Z., 2005b. Quantitative evaluation of protein-DNA interactions using an optimized knowledge-based potential. *Nucleic Acids Res.* 33, pp. 546–558. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gki204>.
- Luger, K., Mäder, A.W., Richmond, R.K., Sargent, D.F., Richmond, T.J., 1997. Crystal structure of the nucleosome core particle at 2.8 Å resolution. *Nature* 389, 251–260. Available at: <http://www.nature.com/doi/10.1038/38444>.
- Lund, J., Sato, A., Akira, S., Medzhitov, R., Iwasaki, A., 2003. Toll-like receptor 9-mediated recognition of Herpes simplex virus-2 by plasmacytoid dendritic cells. *J. Exp. Med.* 198, 513–520. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12900525>.

- Luscombe, N.M., Austin, S.E., Berman, H.M., Thornton, J.M., 2000. An overview of the structures of protein-DNA complexes. *Genome Biol.* 1:REVIEWS001 Available at: <http://genomebiology.com/2000/1/1/reviews/001.1%5Chttp://genomebiology.com/2000/1/1/reviews/001>.
- Luscombe, N.M., Laskowski, R.A., Thornton, J.M., 2001. Amino acid-base interactions: A three-dimensional analysis of protein-DNA interactions at an atomic level. *Nucleic Acids Res.* 29, . pp. 2860–2874. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/29.13.2860>.
- Maerkl, S.J., Quake, S.R., 2007. A systems approach to measuring the binding energy landscapes of transcription factors. *Science* 315 (80–), 233–237. Available at: <http://www.sciencemag.org/cgi/doi/10.1126/science.1131007>.
- Mandel-Gutfreund, Y., Margalit, H., 1998. Quantitative parameters for amino acid-base interaction: Implications for prediction of protein-DNA binding sites. *Nucleic Acids Res.* 26, 2306–2312. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9580679>.
- Manders, P., Thomas, R., 2000. Immunology of DNA vaccines: CPG motifs and antigen presentation. *Inflamm. Res.* 49, 199–205. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10893042>.
- Masse, J.E., Wong, B., Yen, Y.-M., *et al.*, 2002. The *S.cerevisiae* architectural HMGB Protein NHP6A Complexed with DNA: DNA and protein conformational changes upon binding. *J. Mol. Biol.* 323, 263–284. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0022283602009385>.
- Matthews, B.W., Ohlendorf, D.H., Anderson, W.F., Takeda, Y., 1982. Structure of the DNA-binding region of lac repressor inferred from its homology with CRO repressor. *Proc. Natl. Acad. Sci.* 79, 1428–1432. Available at: <http://www.pnas.org/content/79/5/1428.short%5Chttp://www.pnas.org/cgi/doi/10.1073/pnas.79.5.1428>.
- Ma, X., Wu, J., 2013. Identification of DNA-binding proteins using support vector machine with sequence information. *Comput. Math. Methods Med.* 2013. Available at: <http://downloads.hindawi.com/journals/cmmm/aip/524502.pdf>.
- McClarin, J.A., Frederick, C.A., Wang, B.C., *et al.*, 1986. Structure of the DNA-Eco RI endonuclease recognition complex at 3 Å resolution. *Science* 234, 1526–1541. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/3024321>.
- Meyer, C.A., Liu, X.S., 2014. Identifying and mitigating bias in next-generation sequencing methods for chromatin biology. *Nat. Rev. Genet.* 15, 709–721. Available at: <http://www.nature.com/doi/doi/10.1038/nrg3788>.
- Modlin, R.L., 2000. Immunology. A Toll for DNA vaccines. *Nature* 408, 659–660. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11130055>.
- Morimoto, R.I., 1998. Regulation of the heat shock transcriptional response: Cross talk between a family of heat shock factors, molecular chaperones, and negative regulators. *Genes Dev.* 12, 3788–3796. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9869631>.
- Morozov, A.V., Havranek, J.J., Baker, D., Siggia, E.D., 2005. Protein-DNA binding specificity predictions with structural models. *Nucleic Acids Res.* 33, 5781–5798.
- Nimrod, G., Schushan, M., Szilágyi, A., Leslie, C., Ben-Tal, N., 2010. iDBPs: A web server for the identification of DNA binding proteins. *Bioinformatics* 26, 692–693. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20089514>.
- Nimrod, G., Szilágyi, A., Leslie, C., Ben-Tal, N., 2009. Identification of DNA-binding proteins using structural, electrostatic and evolutionary features. *J. Mol. Biol.* 387, 1040–1053. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/19233205>.
- Noy, A., Suthitbutpong, T., A. Harris, S., 2016. Protein/DNA interactions in complex DNA topologies: Expect the unexpected. *Biophys. Rev.* 8, 233–243. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27738452>.
- Nutiu, R., Friedman, R.C., Luo, S., *et al.*, 2011. Direct measurement of DNA affinity landscapes on a high-throughput sequencing instrument. *Nature biotechnology* 29, 659–664.
- Ofran, Y., Mysore, V., Rost, B., 2007. Prediction of DNA-binding residues from sequence. *Bioinformatics* 23, i347–i353. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/1764>.
- Ogawa, N., Biggin, M.D., 2012. High-throughput SELEX determination of DNA sequences bound by transcription factors in vitro. *Methods Mol. Biol.* 786, 51–63.
- Olson, W.K., Gorin, A.A., Lu, X.J., Hock, L.M., Zhurkin, V.B., 1998. DNA sequence-dependent deformability deduced from protein-DNA crystal complexes. *Proc. Natl. Acad. Sci. USA.* 95, 11163–11168. Available at: <http://www.pnas.org/cgi/doi/10.1073/pnas.95.19.11163>.
- Orengo, C.A., Michie, A.D., Jones, S., *et al.*, 1997. CATH – A hierarchic classification of protein domain structures. *Structure* 5, 1093–1108. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9309224>.
- Paquet, E., Viktor, H.L., 2015. Molecular dynamics, monte carlo simulations, and Langevin dynamics: A computational review. *Biomed Res. Int.* 2015, 1–18. Available at: <http://www.hindawi.com/journals/bmri/2015/183918/>.
- Paull, T.T., Haykinson, M.J., Johnson, R.C., 1993. The nonspecific DNA-binding and -bending proteins HMG1 and HMG2 promote the assembly of complex nucleoprotein structures. *Genes Dev.* 7, 1521–1534. Available at: <http://www.genesdev.org/cgi/doi/10.1101/gad.7.8.1521>.
- Paz, I., Kligun, E., Bengad, B., Mandel-Gutfreund, Y., 2016. BindUP: A web server for non-homology-based prediction of DNA and RNA binding proteins. *Nucleic Acids Res.* 44, W568–W574. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27198220%5Chttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4987955>.
- Peled, S., Leiderman, O., Charar, R., *et al.*, 2016. De-novo protein function prediction using DNA binding and RNA binding proteins as a test case. *Nat. Commun.* 7, 13424. Available at: <http://www.nature.com/doi/doi/10.1038/ncomms13424>.
- Ponting, C.P., Schultz, J., Milpetz, F., Bork, P., 1999. SMART: Identification and annotation of domains from signalling and extracellular protein sequences. *Nucleic Acids Res.* 27, 229–232. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9847187>.
- Ptashne, M., 2005. Regulation of transcription: From lambda to eukaryotes. *Trends Biochem. Sci.* 30, 275–279. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15950866>.
- Rajagopal, N., Srinivasan, S., Kooshesh, K., *et al.*, 2016. High-throughput mapping of regulatory DNA. *Nat. Biotechnol.* 34, 167–174. Available at: <http://www.nature.com/doi/doi/10.1038/nbt.3468>.
- Reece-Hoyes, J.S., Marian Walhout, A.J., 2012. Yeast one-hybrid assays: A historical and technical perspective. *Methods*, 57, . pp. 441–447. Available at: <http://link.springer.com/10.1007/978-1-62703-673-3>.
- Rice, J., King, C.A., Spellerberg, M.B., Fairweather, N., Stevenson, F.K., 1999. Manipulation of pathogen-derived genes to influence antigen presentation via DNA vaccines. *Vaccine* 17, 3030–3038. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10462238>.
- Richter, P.H., Eigen, M., 1974. Diffusion controlled reaction rates in spheroidal geometry. Application to repressor–operator association and membrane bound enzymes. *Biophys. Chem* 2, 255–263. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/4474030>.
- Robertson, T.A., Varani, G., 2007. An all-atom, distance-dependent scoring function for the prediction of protein-DNA interactions from structure. *Proteins* 66, 359–374. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17078093>.
- Rohs, R., Jin, X., West, S.M., *et al.*, 2010. Origins of specificity in protein-DNA recognition. *Annu. Rev. Biochem.* 79, 233–269. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20334529>.
- Rohs, R., West, S.M., Liu, P., Honig, B., 2009a. Nuance in the double-helix and its role in protein–DNA recognition. *Curr. Opin. Struct. Biol.* 19, 171–177. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0959440X09000347>.
- Rohs, R., West, S.M., Sosinsky, A., *et al.*, 2009b. The role of DNA shape in protein–DNA recognition. *Nature* 461, 1248–1253. Available at: <http://www.nature.com/doi/doi/10.1038/nature08473>.
- Rouvière-Yaniv, J., Yaniv, M., Germond, J.E., 1979. E. coli DNA binding protein HU forms nucleosomal-like structure with circular double-stranded DNA. *Cell* 17, 265–274. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/222478>.
- Scaffidi, P., Bianchi, M.E., 2001. Spatially precise DNA bending is an essential activity of the sox2 transcription factor. *J. Biol. Chem.* 276, 47296–47302. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11584012>.
- Schultz, S.C., Shields, G.C., Steitz, T.A., 1991. Crystal structure of a CAP-DNA complex: The DNA is bent by 90 degrees. *Science* 253, 1001–1007. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/1653449>.
- Shao, X., Tian, Y., Wu, L., *et al.*, 2009. Predicting DNA- and RNA-binding proteins from sequences with kernel methods. *J. Theor. Biol.* 258, 289–293. Available at: <http://www.sciencedirect.com/science/article/pii/S0022519309000289>.

- Shimamoto, N., 1999. One-dimensional diffusion of proteins along DNA. Its biological and chemical significance revealed by single-molecule measurements. *J. Biol. Chem.* 274, 15293–15296. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10336412>.
- Siggers, T.W., Honig, B., 2007. Structure-based prediction of C₂H₂ zinc-finger binding specificity: Sensitivity to docking geometry. *Nucleic Acids Res.* 35, 1085–1097. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17264128>.
- Si, J., Zhao, R., Wu, R., 2015. An overview of the prediction of protein DNA-binding sites. *Int. J. Mol. Sci.* 16, 5194–5215.
- Spolar, R.S., Record, M.T., 1994. Coupling of local folding to site-specific binding of proteins to DNA. *Science* 263, 777–784. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/8303294>.
- Stawiski, E.W., Gregoret, L.M., Mandel-Gutfreund, Y., 2003. Annotating nucleic acid-binding function based on protein structure. *J. Mol. Biol.* 326, 1065–1079. Available at: <http://www.sciencedirect.com/science/article/pii/S0022283603000317>.
- Stella, S., Cascio, D., Johnson, R.C., 2010. The shape of the DNA minor groove directs binding by the DNA-bending protein Fis. *Genes Dev.* 24, 814–826. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20395367>.
- Szilágyi, A., Skolnick, J., 2006. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *J. Mol. Biol.* 358, 922–933. Available at: <http://www.sciencedirect.com/science/article/pii/S002228360600252X>.
- Tabeta, K., Georgel, P., Janssen, E., *et al.*, 2004. Toll-like receptors 9 and 3 as essential components of innate immune defense against mouse cytomegalovirus infection. *Proc. Natl. Acad. Sci. USA.* 101, 3516–3521. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/14993594>.
- Takeda, T., Corona, R.I., Guo, J.T., 2013. A knowledge-based orientation potential for transcription factor-DNA docking. *Bioinformatics* 29, 322–330.
- Tan, C., Terakawa, T., Takada, S., 2016. Dynamic coupling among protein binding, sliding, and DNA bending revealed by molecular dynamics. *J. Am. Chem. Soc.* 138, 8512–8522. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27309278>.
- Tapias, A., López, G., Ayora, S., 2000. Bacillus subtilis LrpC is a sequence-independent DNA-binding and DNA-bending protein which bridges DNA. *Nucleic Acids Res.* 28, 552–559. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10606655>.
- Thanbichler, M., Wang, S.C., Shapiro, L., 2005. The bacterial nucleoid: A highly organized and dynamic structure. *J. Cell. Biochem.*, 96. pp. 506–521. Available at: <http://doi.wiley.com/10.1002/jcb.20519>.
- Thompson, J.F., Landy, A., 1988. Empirical estimation of protein-induced DNA bending angles: Applications to lambda site-specific recombination complexes. *Nucleic Acids Res.* 16, 9687–9705. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/2972993>.
- Tong, Y., Falk, J., 2009. Genome-wide analysis for protein – DNA interaction: Chip-chip. *Methods in Mol. Biol. (Clifton, N. J.)* 235–251. United States Available at: http://link.springer.com/10.1007/978-1-60327-378-7_15.
- Tuckerman, M.E., Martyna, G.J., 2000. Understanding modern molecular dynamics: Techniques and applications. *J. Phys. Chem. B*, 104. pp. 159–178. Available at: <http://pubs.acs.org/doi/abs/10.1021/jp992433y>.
- Vassallo, M., Ralston, A., 1992. Algorithms for De Bruijn sequences – a case study in the empirical analysis of algorithms. *Comput. J.* 35. pp. 88–90. Available at: <https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/35.1.88>.
- Velmurugu, Y., Chen, X., Slogoff Sevilla, P., Min, J.-H., Ansari, A., 2016. Twist-open mechanism of DNA damage recognition by the Rad4/XPC nucleotide excision repair complex. *Proc. Natl. Acad. Sci. USA* 113, E2296–E2305. Available at: <http://www.pnas.org/content/pnas/early/2016/03/30/1514666113.abstract.html?collection>.
- Walter, M.C., Rattei, T., Arnold, R., *et al.*, 2009. PEDANT covers all complete RefSeq genomes. *Nucleic Acids Res.*, 37. pp. D408–D411. Available at: <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkn749>.
- Wang, L., Brown, S.J., 2006. BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences. *Nucleic Acids Res.* 34, W243–W248. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/16845003>.
- Wang, Y., Cortez, D., Yazdi, P., *et al.*, 2000. BASC, a super complex of BRCA1-associated proteins involved in the recognition and repair of aberrant DNA structures. *Genes Dev.* 14, 927–939. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10783165>.
- Wei, L., Tang, J., Zou, Q., 2017. Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information. *Inf. Sci. (NY)* 384, 135–144. Available at: <http://www.sciencedirect.com/science/article/pii/S0020025516304509>.
- Wilson, K.A., Kellie, J.L., Wetmore, S.D., 2014. DNA-protein π -interactions in nature: Abundance, structure, composition and strength of contacts between aromatic amino acids and DNA nucleobases or deoxyribose sugar. *Nucleic Acids Res.* 42, 6726–6741. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24744240>.
- Wilson, K.A., Wetmore, S.D., 2015. A survey of DNA-protein π -interactions: A comparison of natural occurrences and structures, and computationally predicted structures and strengths. In: Scheiner, S. (Ed.), *Noncovalent Forces*. Cham: Springer International Publishing, pp. 501–532. Available at: https://doi.org/10.1007/978-3-319-14163-3_17.
- Wold, M.S., 1997. REPLICATION PROTEIN A: A heterotrimeric, single-stranded dna-binding protein required for eukaryotic dna metabolism. *Annu. Rev. Biochem.* 66, 61–92. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/9242902>.
- Wu, C., 1995. Heat shock transcription factors: Structure and regulation. *Annu. Rev. Cell Dev. Biol.* 11, 441–469. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/8689565>.
- Wu, J., Liu, H., Duan, X., *et al.*, 2009. Prediction of DNA-binding residues in proteins from amino acid sequences using a random forest model with a hybrid feature. *Bioinformatics* 25, 30–35.
- Xie, Z., Hu, S., Qian, J., Blackshaw, S., Zhu, H., 2011. Systematic characterization of protein-DNA interactions. *Cell. Mol. Life Sci.* 68, 1657–1668. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/21207099>.
- Xiong, Y., Xia, J., Zhang, W., Liu, J., 2011. Exploiting a reduced set of weighted average features to improve prediction of DNA-binding residues from 3D structures. *PLOS One* 6, e28440. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/22174808>.
- Yan, C., Terribilini, M., Wu, F., *et al.*, 2006. Predicting DNA-binding sites of proteins from amino acid sequence. *BMC Bioinformatics* 7, 262. Available at: <http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-262>.
- Yan, J., Kurgan, L., 2017. DRNAPred, fast sequence-based method that accurately predicts and discriminates DNA-and RNA-binding residues. *Nucleic Acids Res.* 45.
- Yu, X., Cao, J., Cai, Y., Shi, T., Li, Y., 2006. Predicting rRNA-, RNA-, and DNA-binding proteins from primary structure with support vector machines. *J. Theor. Biol.* 240, 175–184. Available at: <http://www.sciencedirect.com/science/article/pii/S0022519305004212>.
- Zhang, C., Liu, S., Zhu, Q., Zhou, Y., 2005. A knowledge-based energy function for protein-ligand, protein-protein, and protein-DNA complexes. *J. Med. Chem.* 48, 2325–2335. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15801826>.
- Zhao, H., Wang, J., Zhou, Y., Yang, Y., 2014. Predicting DNA-binding proteins and binding residues by complex structure prediction and application to human proteome. *PLOS One* 9, 26–28.
- Zhou, J., Lu, Q., Xu, R., He, Y., Wang, H., 2017. EL _ PSSM-RT: DNA-binding residue prediction by integrating ensemble learning with PSSM. *Relation Transformation*. 1–16.
- Zhou, W., Yan, H., 2011. Prediction of DNA-binding protein based on statistical and geometric features and support vector machines. *Proteome Sci.* 9 (Suppl 1), S1. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3289070&tool=pmcentrez&rendertype=abstract>.

Gene Regulatory Network Review

Enze Liu, Indiana University School of Medicine, Indianapolis, IN, United States and Indiana University, Indianapolis, IN, United States

Lang Li, The Ohio State University, Columbus, OH, United States

Lijun Cheng, The Ohio State University, Columbus, OH, United States

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

BN	Bayesian network	MI	Mutual information
CMI	Conditional mutual information	PPI	Protein–protein interactions
DBN	Dynamic Bayesian Network	ODE	Ordinary differential equation
EST	Expressed sequence tags	PTM	Posttranslational modification
GRN	Gene regulatory networks	ROC	Receiver-operating characteristic
KNN	k-nearest neighbors	TF	Transcription factors

Introduction

Genes act as a secret key in governing the entire biological system's function. Genes' cooperation and interactions form a dynamic network that brings about our vivid biodiversity (Cheng *et al.*, 2013). If one gene is regulated, then it affects its targets, which further affect their next targets, causing a domino effect towards the entire cellular environment. A gene regulatory network (GRN) is the collection of molecular regulators (DNA segments, miRNAs, or histones) in a cell that interact with each other directly or indirectly (through their RNA and protein expression products) and with other substances in the cell (Vijesh *et al.*, 2013). By a comprehensive network perspective, we can mine the direct regulations among genes and gain deep insights into various biological processes, which could further facilitate medical related fields such as drug design or drug target discovery.

A GRN is composed of functional linkages between regulators and targets to which their products bind. *Cis*-regulatory element and *trans*-regulatory element are two main regulatory types (Peters, 2008). *Cis*-regulatory elements are present near the structural portion of the gene/protein as the gene they regulate, such as, the photosynthetic protein family, are expressed at the same time in development. Whereas *trans*-regulatory elements can distantly regulate genes from which they were transcribed. Enhancers and multiple *trans*-acting factors are essential for *trans*-control transcription initiations. Regulators contain DNA epigenetic modifications by methylation, miRNA, transcription factors (TFs), and posttranslational modification (PTM) that include histone proteins and other proteins, which are involved in methylation, phosphorylation, acetylation, ubiquitylation, and sumoylation (Fig. 1). Motif is a basic unit of GRN subgraph for these regulators and gene sets regulated with similar functions in networks. Normally, TFs coded by a gene can regulate gene expression by binding to specific motifs. MicroRNAs (miRNAs) regulate gene expression via RNA silencing or posttranscription regulations. Histones can alter the chromatin structure, which further controls the access of TFs and polymerases to genes thus resulting in an expression regulation (Dong and Weng, 2013). Methylation plays a crucial role in regulating gene expression by blocking the promoters that can activate TFs (Phillips, 2008). Large experimental evidence has been gathered to verify biology gene interactions, such as KEGG (see "Relevant Websites section"), Pathway Commons (see "Relevant Websites section"), MetaCyc Metabolic Pathway Database (see "Relevant Websites section"), JASPAR CORE (see "Relevant Websites section"), HistoneDB 2.0 (see "Relevant Websites section") and miRGator v3.0 (see "Relevant Websites section"), GeneHancer (see "Relevant Websites section"), etc. However, this is only the tip of the iceberg in terms of genome knowledge. Huge knowledge mining from genome is still in its infancy. Comprehensive understanding of GRNs is a major challenge in the field of systems biology. Recent advances in high-throughput techniques provide an opportunity to reconstruct regulatory networks, by offering a huge amount of binding data, like ChIP-Seq, miRNA-Seq, ATAC-Seq with expression data RNA-Seq (Cheng *et al.*, 2011). By performing various topological analyses, including its hierarchical organization and motif enrichment, a potential molecular mechanism would be detected. Gene expression microarrays and RNA-Seq provide us another chance to evaluate gene expression dependencies between TFs and its target genes systematically for understanding systematical biological activities (Madan Babu and Teichmann, 2003). Accurate reconstruction of the gene regulatory interactions, forming GRN, is one of the key tasks in systems biology.

The inference of a GRN is often accomplished through the use of gene expression data. So far, there are numerous computational methods and models developed for restoring GRNs in a real cellular environment. However, each of them have their own assumptions and methods, drawing different blueprints that the GRN described. There is still much confusion about the basic meaning of GRN, ways of assessment, and possible biomedical application. Typically, the relationships between genes are directional in nature and they can change over time or in response to external stimulus. Researchers are facing the choice of whether to include extra features such as causality and temporal behaviors when modeling gene networks or not. In this paper, we aim to clarify the meaning and usage conditions for GRNs and also provide a discussion on next steps to bring GRNs closer to clinical and medical application. In the section "Expression Data for Reconstructing GRN," we briefly introduced the expression data composition and two major platforms that generate expression data. In the section "Static Network Models and Inference

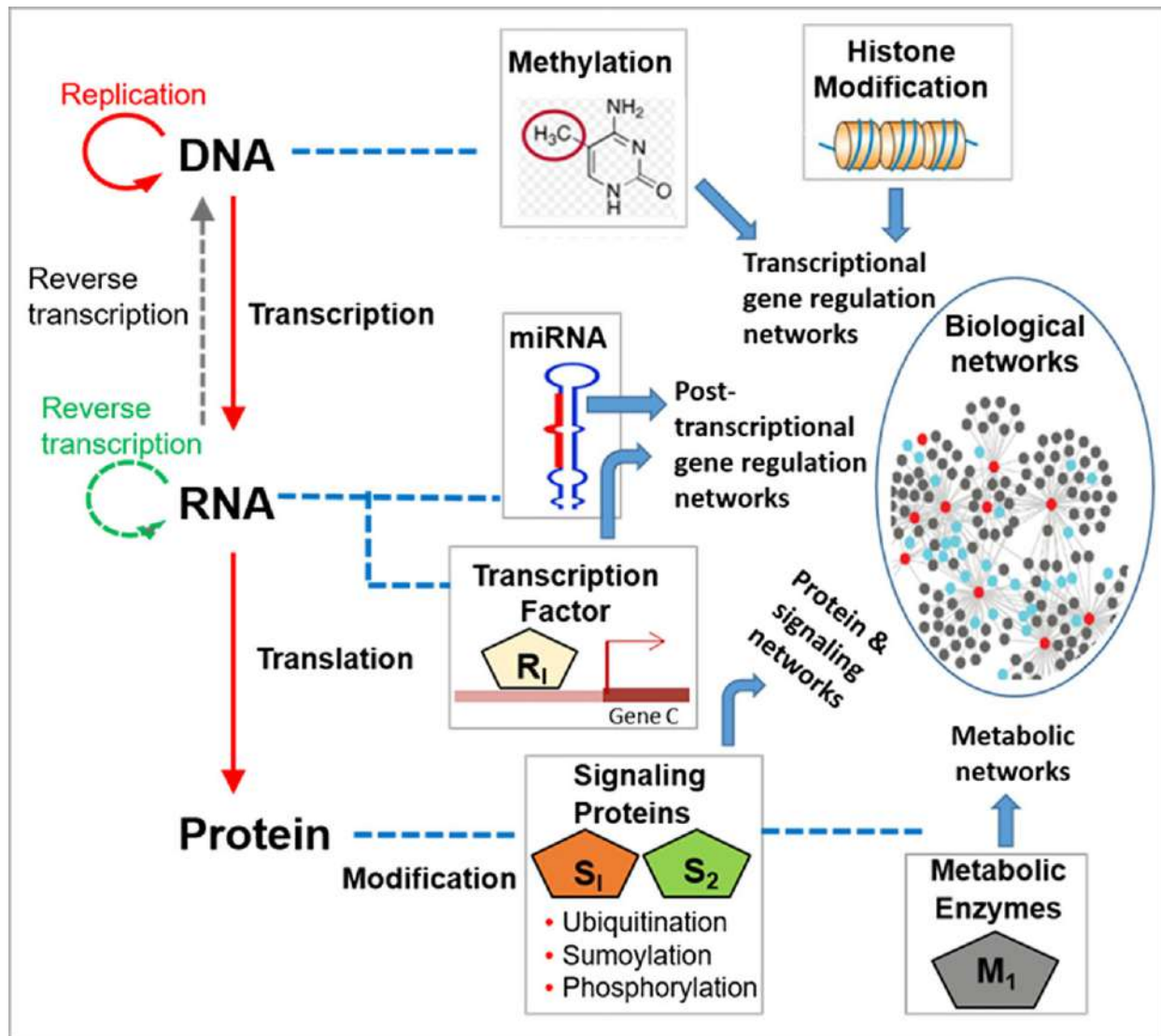


Fig. 1 Central dogma and main regulating elements for biological networks at each molecular level.

Methods,” we focus on network inference methods for reconstructing a static GRN. We highlighted methods themselves, the underlying assumptions and advantages/drawbacks by indicating their features and comparing them. For some representative methods we illustrated examples of how they had been applied in real study. In the section “Dynamic Network Models and Inference Methods,” we discussed inferring methods that simulate temporal changes of gene regulations over time in a dynamic GRN. In the section “Network Validation and Assessment of the Network Inference Methods,” we expanded to how to validate the inferred network and brought up several measurements for assessing the performance of different inference methods. In the section “Gene Regulatory Network Inference Development Trend,” we forecast the trend of further development of GRN inference methods, including building comprehensive GRNs by integrating other omics data besides expression data and methods updating trends for resolving current issues during GRN reconstruction.

Expression Data for Reconstructing Gene Regulatory Network

Nowadays, GRNs are mainly reconstructed from gene expression data, which is generated by high-throughput platforms such as microarray and RNA-seq. Microarray technology uses a set of short expressed sequence tags (ESTs) generated from the cDNA library to hybridize target RNAs and then reflect fluorescent intensity, which is used to measure the expression level of certain RNAs. RNA-seq utilizes next generation sequencing (NGS) technology to sequence the cDNA library generated from sample RNAs and then directly generates read (RNA) counts to represent expression level of RNAs. Although their principles and experiments differ, data preprocessing and normalization methods for microarray and RNA-seq vary drastically. Normalized gene expression data generated from either platform have the same form associated with a matrix with M rows of genes under N columns of

conditions (samples). Static network modeling and dynamic network modeling with time series are two typical reconstruction types for GRN. Five representative models related to the two types will be discussed in detail (Fig. 2).

Static Network Models and Inference Methods

Static models reflect interactions among genes at any given time point. These models are often used to describe network topologies and qualitative features of gene relationships (Wang and Huang, 2014).

CoExpression Networks and Relevance Networks

A GRN built with a correlation based method is called a coexpression network (Muhammad *et al.*, 2017). A coexpression network is constructed by calculating pairwise correlations for each gene pairs to form a fully connected graph, and then by applying a threshold scheme to remove edges between genes that are not significantly correlated (Fig. 2(a)). Pearson correlation, Spearman correlation and Euclidean distance are widely used for measuring linear correlations while mutual information (MI) and kernel correlations between two genes (Cheng *et al.*, 2013) are the representative measures for detecting nonlinear correlations. These measures are used to calculate association on a pair of genes to seek if they are coexpression. *P*-value, *Z*-score, clustering coefficient (Elo *et al.*, 2007), and random matrix theory (Luo *et al.*, 2007) are typical threshold schemes for filtering out low content coexpression.

Pearson correlation ρ is the most basic correlation that measures pairwise correlation between two continuous variables.

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (1)$$

Here X and Y denote expression levels of two genes. *Cov* is the covariance between gene X , Y and sigma σ indicates the individual standard deviation. Pearson correlation measures the collinearity between two genes with the assumption that both genes' expression level follow an approximately normal distribution, making it unsuitable for measuring genes with nonlinear relationships.

Rank based correlation describes correlations by comparing the rank of the variables (genes) instead of covariance. Take Spearman correlation as an example, which compares the monotonic relationship between two variables (gene). If two genes follow a normal expression pattern and have a clearly linear relationship, then Pearson correlation and Spearman correlation would be very similar. However, if two genes are monotonically correlated, not linearly correlated, only the Spearman correlation could measure the nonlinear relationships. In other words, the Spearman correlation is more general and robust compared to the Pearson correlation, even though information gets lost during the ranking transformation.

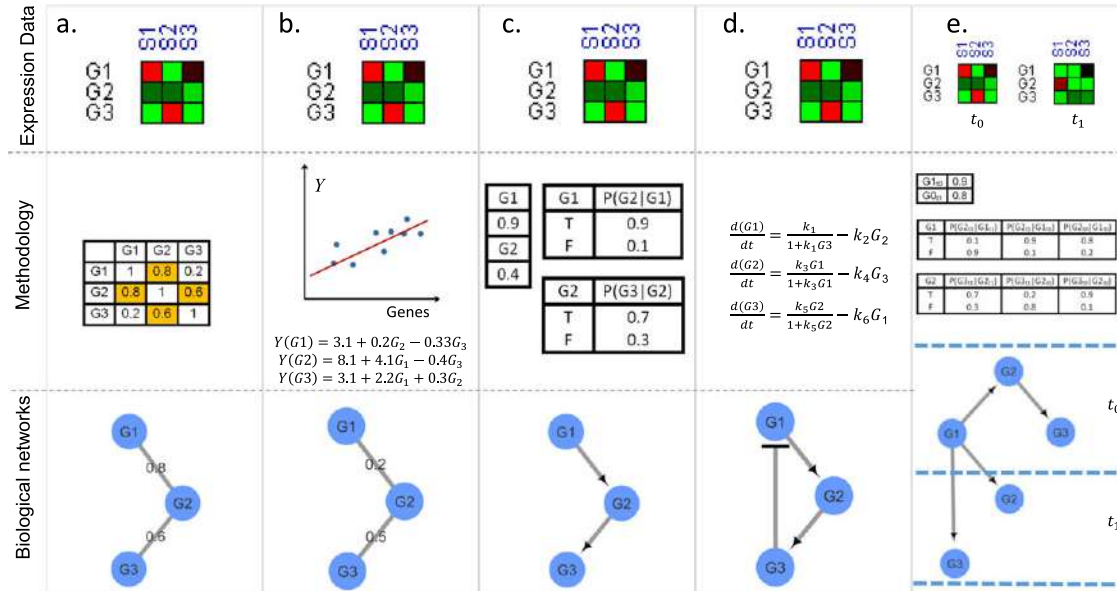


Fig. 2 Representative models and steps for reconstructing gene regulatory network. (a) Coexpression network (b) regression based regulatory network (c) Bayesian network (d) differential equation based regulatory network (e) dynamic Bayesian network.

Distance based correlation mainly refers to Euclidean distance, which is the geometric distance between two genes if two genes are assumed to be in the same p -dimensional Euclidean space. Euclidean distance is sensitive to transformations, whereas the other two are normally invariant to linear transformations.

Information theoretic entropy and mutual information concepts provide a new perspective to scale the correlation between variables. Entropy measures the quantity of information within a given system and two variables that contain more entropy tend to be more related. MI measures all the entropy within a system, including marginal entropy from variables and joint entropy between/among variables. A MI $I(X,Y)$ measures a system of two variables (genes) and can be defined as:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \quad (2)$$

where

$$H(X) = - \sum_{x=1}^m f(x) \log(f(x)) \quad (3)$$

is the marginal entropy and

$$H(X,Y) = - \sum_{x=1, y=1}^{m,n} f(x,y) \log(f(x,y)) \quad (4)$$

is the joint entropy for two continuous variables of gene X and Y .

A gene association network built with MI is called a relevance network (Butte and Kohane, 1999). ARACNe (Margolin *et al.*, 2006) is the most famous and representative algorithm for building a relevance network. ARACNe assumes genes following a normal distribution and performed a Gaussian Kernel estimator (Scott and Sheather, 1985) to estimate marginal distribution $f(x)$ and joint distribution $f(x,y)$ in formula 3 and 4. ARACNe defined its edge-pruning thresholds named data processing inequality (DPI) as: For all available triplets g_i, g_j , and g_k , if g_i interacts with g_k via g_j , then the edges (pairwise MI) among them should follow the following inequality: $I(E_i, E_j) \leq \min(I(E_i, E_k), I(E_k, E_j))$, meaning that the indirect interaction in the system should contain the smallest MI value. Any violated indirect interactions will be pruned.

A GRN based on correlation provides coexpression or cofunctionality from a network scale with no directionality, meaning that if no prior knowledge is provided (e.g., a gene is confirmed to be a transcription factor in a two-gene interaction system), we cannot identify the causations and results of genes in a GRN. MI detects underlying correlations that linear Pearson correlation cannot discover, especially for variables with nonlinear correlation.

Regression Based Approaches

Regression based approaches are a conventional and straightforward method that is widely used in inferring GRNs (Fig. 2(b)). In a regression model, the problem of constructing a GRN is converted to the problem of selecting regulators (e.g., TFs) that have a significant impact towards particular genes. Given an expression matrix with m genes and k conditions. A multiregression model for each gene can be formulated as:

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_k X_{ik} + \varepsilon_i$$

Where: y_i is i th gene of whose regulating relationship towards other genes is to be inferred. β_0 denotes the constant/intercept. X_{ik} indicates the expression values of gene i in condition k . ε_i is the noise/error terms within gene i . For each gene i , a series of beta indicating the contributing towards gene i expression will be predicted. Various feature selection methods can be applied during the inference such as forward, backward, stepwise, ridge, and LASSO. After this step, a p -value cutoff (< 0.05 or less) is usually used to determine whether there is a link between gene i and predictor gene j . If so, the associated beta indicates the weight and the correlation directions.

Regression based approaches not only estimate the underlying associations among genes in a network, but also infer the association intensities. More importantly, the correlations among predictors can easily be incorporated in the formula by adding the correlated predictors' term, making it capable of mimicking real coregulating scenarios in biological networks. However, the model assumes that all the predictors follow a multivariate normal distribution similar to a coexpression network, therefore regression does not infer the directionality of associations.

Bayesian Networks

A Bayesian network (BN) is a probabilistic network model that takes a group of random variables and measures their relationships as a directed acyclic graph (DAG) with all the conditional dependencies among them (Fig. 2(c)). Given a certain network structure G , a BN can be expressed as the following joint probability density function (PDF) p_B (a product of individual marginal PDFs and conditional PDFs):

$$p_B(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i | P_{a_i}^g, \theta_i)$$

Where $P_{a_i}^g$ is the set of parents of node x_i in a graph G , gene node $g \in G$, θ_i is the i th distribution mean, where all gene expression obeys a normal distribution $N(\theta_i, \delta_i)$. All conditional probabilities denote all the edges within the DAG. The product of them clearly indicate the local Markov property of BN: Every node (gene) is conditionally independent of its nondescendants given its parents.

Reconstructing a BN from the given expression data requires two types of inferences: (1) Parameter learning, which infers marginal and conditional distributions of all nodes, and (2) structure learning, which infers the optimal topology that has the biggest overall probability. In parameter learning, the marginal distributions can be either given by prior knowledge or inferred by various methods such as principle of maximum entropy (Clarke, 2007) while conditional distributions are often inferred with approaches such as maximum likelihood estimator or expectation maximization (Friedman *et al.*, 2000). Learning the optimal structure of a BN has been a great challenge for decades. One can always use a brute force approach to go through every possible structure and finally identify the best, which would cost superexponential time complexity to achieve. Thus, most of the existing algorithms for searching the optimal structures are heuristic and progressive. For instance, Monte Carlo Markov chain (MCMC) (Mukherjee and Speed, 2008) methods use approximate sampling approaches. Greedy equivalence search (GES) (Chickering, 2002) proposed a local optimal constraint and performed a stepwise searching scheme. As for the scoring scheme, usually a Bayesian information criterion (BIC) or Akaike's information criterion (AIC) is chosen (Liu, Malone and Yuan, 2012).

Compared to coexpression networks, BNs can clearly point out the directionality of each edge thus revealing a causation relationship among all genes, which is significantly advantageous. Large number experiments have shown that BN can offer a better accuracy and tolerance with noise expression input on small number of genes (Hill *et al.*, 2016). As more variables (genes) are incorporated into the model, the computational time grows superexponentially, which brings up the great challenge of building a whole genomic BN for higher organisms like humans (Hill *et al.*, 2016). Moreover, for one dataset, there might be more than one optimal BN structure that has the equivalent overall probabilities. These models (structures) are called equivalence class (Chickering, 2002), in which each model is probabilistically indistinguishable. Consider two BNs $A \rightarrow B \rightarrow C$ and $C \rightarrow B \rightarrow A$ are probabilistically equivalent since their overall probabilities are $P(A)P(B|A)P(C|B)$ and $P(C)P(B|C)P(A|B)$, which is the same. Causal network (CN) refers to a more strict form of BN, meaning not only conditional dependencies, but also Markov conditions among variables need to be followed in the DAG, which provide a clear causation of one variable over another. Hence, the above BN are not causally equivalent due to different Markov conditions: $A \perp\!\!\!\perp C|B$ (A is independent of C given B) versus $C \perp\!\!\!\perp A|B$ (C is independent of A given B), where in the first network, A causes C , in the second, C causes A . GES algorithm that is mentioned above is a searching algorithm designed for CN models.

Another significant drawback of the BN model lies in its assumption: A DAG doesn't allow the existence of feedback loops, which have been widely proven to exist in biological networks. This can be resolved by dynamic Bayesian network (DBN) (Yu and Liu, 2004; Murphy and Russell, 2002). Additionally, there is no way to observe the type of regulations (activation/inhibition) in a GRN generated with a BN model. Hence, prior knowledge or differential expression analysis is usually incorporated.

Applications

The applications of static models can be found in various fields such as cancer biology and biotechnology (Marbach *et al.*, 2012). Moreover, Chuang *et al.* (2007) constructed a relevance network that uses mutual information to measure protein-protein interactions with protein interaction information from databases and gene expression profiles from breast cancer patients. They were able to identify several important biomarkers that highly correlated to breast cancer metastasis. Zhang *et al.* (2017) utilized linear regression and Cox regression and created a semisupervised learning approach and built a comprehensive drug-disease-gene network for drugs repositioning on ovarian cancer. Yavari *et al.* (2008) clustered genes based on gene ontology (GO) and used a BN to build a regulatory network containing 84 yeast genes based on coclustered information.

Dynamic Network Models and Inference Methods

Dynamic network models capture the dynamic nature of real networks thus yielding quantitative predictions of gene behaviors (Wang and Huang, 2014). If you don't like static models, self-regulations of genes can be reflected in dynamic models.

Dynamic Bayesian Networks

At a given time slice, DBNs show exactly the forms and properties as BNs. However, as an extension of BN, DBNs incorporate time dimensions into the network reconstruction to detect features of dynamic GRNs with time series, including feedback loops and self-regulation that commonly occur in nature and are not able to be simulated by BNs (Fig. 2(e)). A DBN with N genes can be described as:

$$P(X_t|X_{t-1}) = \prod_{i=1}^N P(X_{i,t}|Pa_{Bt}(X_{i,t}))$$

Where $X_{i,t}$ is a given gene i at time slice t and Pa is its parent's set. In a DBN, each node (gene) has duplicates corresponding to the number of time slice t in order to model the cyclic interactions and dynamic behaviors of GRN (Aluru, 2005). Temporal relationships among genes are represented by interconnections between time slices (Chai *et al.*, 2014), which means even though a DBN generates DAGs similar to BN, a DBN could have both interslice connections and intraslice connections. The structural learning and parameter learning of a DBN is similar to a BN. However, to infer the future state, a DBN needs to be converted to hidden Markov models first and then applied with forward or backward algorithms (Murphy and Russell, 2002).

Boolean Network Models

In system biology, a Boolean network refers to a set of genes whose states are binary (activate or inactivate) (D'haeseleer *et al.*, 2000). At a given time slice t , each gene x will only be in one state $S(t)$ where $S(t) = [x_1(t), x_2(t), \dots, x_n(t)]$. States of each gene are determined by one Boolean logic function B , which takes a subset of genes as input and generates the current state for associated genes. Boolean networks use a series of discrete states of the network to represent the complete dynamics of the whole GRN. Current state of the whole network at time slice $t + 1$ is inferred from the network state at time slice t . Hence, to generate a GRN using a Boolean network model, the expression value of genes that are continuous needs to be discretized. The objective of the searching process is to identify a series of logic functions that optimally explain the expression data.

A Boolean network is a simple way to simulate the sophisticated dynamics of GRN. There are several studies that take advantage of Boolean network model and successfully capture some common patterns of regulator genes and the suppressed/activated genes (Shmulevich *et al.*, 2002). However, limitations of the Boolean network model are still obvious: a GRN is a real-time dynamic system and states of genes are more complicated than binary. Hence a Boolean network cannot accurately mimic the real dynamic interactions among genes since gene expression data normally contains a small number of time slices. Moreover, a Boolean network also masks important information during discretization. As gene expression patterns are generally more complex than having only two states, data discretization often leads to information loss.

Constructing GRN Using Ordinary Differential Equations

Unlike Boolean networks or DBNs, GRN modeling by ordinary differential equation uses a series of discrete states to simulate the complete dynamics of GRNs (Fig. 2(d)). Each gene expression variation uses continuous variables to describe the complete concentration change of RNA or protein with generation, degradation, interaction, regulations, and other events. Hence the concentration change of each gene expression is simulated by one or more independent gene variables and their derivatives with an ordinary differential equation (ODE). The following formula is a standard ODE, which denotes the dynamic concentration change of gene x_i expression under the influence from N other genes in the system (Cao *et al.*, 2012).

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_n, p, u)$$

Where x_i is the target gene, $x_1, x_2, \dots, x_n$ are n regulator genes that could affect the expression level of x_i . p is the parameter set and u is the external perturbation, all of which are included in a node function f_i .

A differential equation is said to be *linear* if f can be written as a linear combination of the derivatives of x in time t :

$$x^{(t)} = \sum_{i=1}^n p_i(x) x_i^{(t)} + u^{(t)}(x)$$

where $p_i(x)$ and $u(x)$ are continuous functions in x . For a linear system, regression methods can be used to identify the solutions. However, an optimal evolutionary algorithm is normally used to estimate the nonlinear system to obtain the steady-state parameter equation (Kimura *et al.*, 2004).

GRNs based on differential equations have several limitations: using differential equations assumes the concentration of RNA or protein depends solely on the concentration of other genes within the system. The influence from other molecules or mechanisms (e.g., TFs) is not directly taken into account (Vijesh *et al.*, 2013). Moreover, in reality, parameters in differential equations are usually difficult to estimate in an experiment.

Applications

Van Gerven *et al.* (2008) illustrated a DBN based model that is developed to provide prognostics of carcinoid patients. Lähdesmäki *et al.* (2003) presented an algorithm for searching optimal Boolean network models that best mimic yeast regulatory network from 15 time-series expression data. Geva-Zatorsky *et al.* (2006) proposed a differential equation based model for reconstructing the P53 dynamic system, which plays a key role in tumor suppression.

Network Validation and Assessment of the Network Inference Methods

To prove the correctness and fidelity of the inferred GRN, a network validation is essential after the inference. Normally, various microarray experiments *in silico* are conducted to record the expression change after gene perturbations (Yeang *et al.*, 2005), such as knockout, drug treatment, or overexpression. The regulation that exists between two genes can be directly validated by using variation analysis of gene expression and investigate associated biological functions. Moreover, the direction of regulation can be identified (Chen *et al.*, 2015).

Assessing the performance of inference methods is also an important aspect in GRN inference. Generally, for an undirected network, the inferred network model generated from validation data needs to be compared with the associated gold standard to see the consistency of the presence or absence of edges in gold standard. For a directed network, not only consistency of edge presence and absence, but also the consistency of edge directionality needs to be evaluated. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) edge counts should be calculated. And precision (specificity) and recall (sensitivity),

defined as $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$ respectively, can be calculated and used as performance assessments. The higher the P and R values are, the better gold standard network has been restored by the method. To measure the performance of methods under a series of changes, for instance, a parameter change from 0 to 1 with a 0.1 interval, or different inputs with 10 to 100 genes within. A receiver-operating characteristic (ROC) curve (Balakrishnan, 1992) denotes the performance consistency of inference method under different conditions. In a ROC graph, y and x axis indicate true positive rate (TPR) and false positive rate (FPR), which can be further calculated with the respective formula: $TPR = TP/(TP + FN)$, $FPR = FP/(FP + TN)$.

Gold standards that are based on real world data have been systematically built up in both machine learning and system biology fields. ALARM network (Beinlich *et al.*, 1989) is a popular tool that is firstly applied for comparison of reconstructed network in machine learning. DREAM project (Marbach *et al.*, 2009; Prill *et al.*, 2010; Marbach *et al.*, 2010) consists of a series of well-studied regulatory network of prokaryotes, eukaryotes, including *Escherichia coli* (Simmons *et al.*, 2008) and in silico microarray datasets of knock-down, knock-out and their associated transcriptome data are collected and published systematically. Causal molecular networks inference focus specifically on reconstructing the signaling downstream of receptor tyrosine kinases based on phosphoprotein data from cancer cell lines as well as in silico data. Linear regression and pair-wise correlation obtained the best-scoring model for the experimental data and in silico data respectively compared with prior network, Ensemble method, nonlinear regression, ODEs, and BNs. A similar result showed linear correlation outperforms over 30 network inference methods on *E. coli*, *Staphylococcus aureus*, *Saccharomyces cerevisiae*, and in silico microarray data (Marbach *et al.*, 2012). A recent large scale study (Saelens *et al.*, 2018) evaluated 42 methods for detecting coexpression networks on nine genome-wide gene expression datasets, including *E. coli*, *Saccharomyces cerevisiae*, human datasets, and two synthetic datasets. Benchmark results suggested that decomposition methods can detect local coexpression networks while other methods cannot. Topology inference of GRNs from high-throughput omics data is a long-standing open challenge in computational systems biology. Accuracy and speed become a bottleneck for optimum network reconstruction in such big data.

Existing gold standards of biological networks suffer from the following issues: most of the gold standards describe certain biological processes most likely in primitive organisms such as *E. coli* and yeast, which has a significant regulation mechanism compared with humans. Hence they cannot fully reflect the performance of certain methods that are specifically designed for humans. Secondly, many of these gold standards are the product of curated and repetitive studies among different labs, across different time periods, hence under different experiment conditions, which means they represent more general patterns of certain biological processes, which may not necessarily cover some extreme cases. Thus, using these gold standards to benchmark methods developed for extreme cases could result in bias. For instance, a method for inferring GRNs especially for cancer cells will show a biased benchmark result when using gold standards that mimic the GRN in a normal human cell.

Gene Regulatory Network Inference Development Trend

Multi-Omics Data Integration and Hybrid Model

Due the development of genomic, transcriptomic, and proteomic research, numerous novel experiments and computational technologies lead to the explosion of multiomics data curation. For instance, genomic database NCBI (Coordinators, 2016), transcriptomic database ExPASy (Gasteiger *et al.*, 2003), protein database Pfam (Bateman *et al.*, 2002), pathway database KEGG (Ogata *et al.*, 1999), expression data hub GEO (Edgar *et al.*, 2002), and ArrayExpress (Brazma *et al.*, 2003) store thousands of high-throughput datasets across all kinds of diseases. These databases have a considerable overlaps in terms of describing same biological processes or functions. These overlapped data actually measure the same thing however from various different angles. Hence, when building GRNs from expression data, if associated data (e.g., mutation or binding information) can be utilized, then the inference could be more solid, accurate, and comprehensive (Coordinators, 2016). Take Gong *et al.* (2015) as an example; in the article, they integrated data from RNA-seq, histone modification ChIP-seq, transcription factor, and gene perturbation experiments into a DBN model that resembles a temporal control of gene expression patterns during human cardiac differentiation. Moreover, Hore *et al.* (2016) created a comprehensive GRN by incorporating multiomics data into a Bayesian sparse tensor decomposition model. The common features across multiple tissues are discovered from one GRN. Lu *et al.* (2017) combined ChIP-seq data with expression data from RNA-seq to identify transcription factor binding sites and regulatory modules, which could be considered as another novel method of reconstructing a comprehensive GRN. Due to the rapid advancing single-cell techniques, mapping GRNs from single-cell multiomics data could provide opportunities to systematically understand cellular heterogeneity, which can greatly facilitate precision medicine studies (Fiers *et al.*, 2018).

As more types of omics data were incorporated into the model, the number of variables significantly increased, which could cause convergence difficulties of time consuming models such as BN and regression (De Smet and Marchal, 2010). To overcome this issue, novel methods have been developed under two ideas. (1) New models and learning algorithms that contain a higher learning efficiency. Take deep learning as an example; it has been extensively developed and applied during recent years and proven to be successful in many machine learning tasks mainly due to its multiple layers and hierarchical representation. "D-GEX" (Chen *et al.*, 2016), a deep learning method, has been developed for training GRNs from expression data and predicting expression of target genes in cancer. The new model has been proven to have a better outcome compared to linear regression and k-nearest neighbors (KNN) regression (Bentley, 1975). (2) Hybrid models and learning algorithms that incorporate existing approaches together to get a better performance. For instance, Liu *et al.* (2016) developed a hybrid method called a local , which can be summarized as follows: use pair-wise conditional mutual information (CMI) to build a skeleton; then apply a BN model for each

of the genes and their first-order neighbors to determine the direction of regulations; integrate subnetworks into candidate network; and perform CMI again to trim necessary edges so as to obtain comprehensive optimization. Integrating different models to solve the problem might be a feasible way to model expression data with high dimensions. Taken together, these approaches give us new thoughts on using newly developed methods to solve GRN related problems.

GRN Application for Precision Medicine

GRNs can be used to reveal numerous biological functions, which also include cancer initiation and progression. Through a GRN, one may be able to better understand how genes and proteins interact, which further leads to a better understanding of functions of oncogenes and related pathways, offering a better discovery towards drug target identification. A typical application is a GRN approach for precision medicine. Mounika Inavolu *et al.* (2017) developed a novel algorithm that integrates expression profiles with protein–protein interaction (PPI) networks to identify key deregulated subnetworks, which facilitated drug target discovery especially for breast cancer. Precision medicine target-drug selection (PMTDS) is another novel method for detecting individual interaction networks from multiomics data and recommending the optimum targets and associated drugs for precision medicine style (Vasudevaraja *et al.*, 2017).

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Cell Modeling and Simulation. Community Detection in Biological Networks. Community Detection in Biological Networks. Computational Systems Biology. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine. Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine. Graphlets and Motifs in Biological Networks. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Network Models. Network Properties. Network-Based Analysis for Biological Discovery. Networks in Biology. Pathway Informatics. Protein-DNA Interactions. Regulation of Gene Expression

References

- Aluru, S., 2005. Handbook of Computational Molecular Biology. CRC Press.
- Balakrishnan, N., 1992. Handbook of the Logistic Distribution. New York, Basel: Marcel Dekker. Inc.
- Bateman, A., Birney, E., Cerruti, L., *et al.*, 2002. The Pfam protein families database. *Nucleic Acids Research* 30 (1), 276–280.
- Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F., 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: *Proceedings of the Conference on Artificial Intelligence in Medical Care* 89, pp. 247–256. Berlin, Heidelberg: Springer.
- Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18 (9), 509–517.
- Brazma, A., Parkinson, H., Sarkans, U., *et al.*, 2003. ArrayExpress – A public repository for microarray gene expression data at the EBI. *Nucleic Acids Research* 31 (1), 68–71.
- Butte, A.J., Kohane, I.S., 1999. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing* 2000, 418–429.
- Cao, J., Qi, X., Zhao, H., 2012. Modeling gene regulation networks using ordinary differential equations. In: *Proceedings of the Next Generation Microarray Bioinformatics*, pp. 185–197. Humana Press.
- Chai, L.E., Loh, S.K., Low, S.T., *et al.*, 2014. A review on the computational approaches for gene regulatory network construction. *Computers in Biology and Medicine* 48, 55–65.
- Cheng, L., Khorasani, K., Ding, Y., Guo, X., 2013. Gene interaction networks based on kernel correlation metrics. *International Journal of Computational Biology and Drug Design* 6 (1–2), 72–92.
- Cheng, C., Yan, K.K., Hwang, W., *et al.*, 2011. Construction and analysis of an integrated regulatory network derived from high-throughput sequencing data. *PLOS Computational Biology* 7 (11), e1002190.
- Chen, Y., Li, Y., Narayan, R., Subramanian, A., Xie, X., 2016. Gene expression inference with deep learning. *Bioinformatics* 32 (12), 1832–1839.
- Chen, Y.H., Yang, C.D., Tseng, C.P., Huang, H.D., Ho, S.Y., 2015. GeNOSA: Inferring and experimentally supporting quantitative gene regulatory networks in prokaryotes. *Bioinformatics* 31 (13), 2151–2158.
- Chickering, D.M., 2002. Optimal structure identification with greedy search. *Journal of Machine Learning Research* 3 (2002), 507–554.
- Chuang, H.Y., Lee, E., Liu, Y.T., Lee, D., Ideker, T., 2007. Network-based classification of breast cancer metastasis. *Molecular Systems Biology* 3 (1), 140.
- Clarke, B., 2007. Information optimality and Bayesian modelling. *Journal of Econometrics* 138 (2), 405–429.
- Coordinators, N.R., 2016. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 44 (Database issue), D7.
- De Smet, R., Marchal, K., 2010. Advantages and limitations of current network inference methods. *Nature Reviews Microbiology* 8 (10), 717.
- Dong, X., Weng, Z., 2013. The correlation between histone modifications and gene expression. *Epigenomics* 5 (2), 113–116.
- D'haeseleer, P., Liang, S., Somogyi, R., 2000. Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics* 16 (8), 707–726.
- Edgar, R., Domrachev, M., Lash, A.E., 2002. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research* 30 (1), 207–210.
- Elo, L.L., Järvenpää, H., Orešić, M., Lahesmaa, R., Aittokallio, T., 2007. Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process. *Bioinformatics* 23 (16), 2096–2103.
- Fiers, M.W., Minnoye, L., Aibar, S., *et al.*, 2018. Mapping gene regulatory networks from single-cell omics data. *Briefings in Functional Genomics*.
- Friedman, N., Linial, M., Nachman, I., Pe'er, D., 2000. Using Bayesian networks to analyze expression data. *Journal of Computational Biology* 7 (3–4), 601–620.
- Gasteiger, E., Gattiker, A., Hoogland, C., *et al.*, 2003. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Research* 31 (13), 3784–3788.
- Geva-Zatorsky, N., Rosenfeld, N., Itzkovitz, S., *et al.*, 2006. Oscillations and variability in the p53 system. *Molecular Systems Biology* 2 (1).
- Gong, J., Liu, C., Liu, W., *et al.*, 2015. An update of miRNASNP database for better SNP selection by GWAS data, miRNA expression and online tools. *Database* 2015.
- Hill, S.M., Heiser, L.M., Cokelaer, T., *et al.*, 2016. Inferring causal molecular networks: Empirical assessment through a community-based effort. *Nature Methods* 13 (4), 310.
- Hore, V., Viñuela, A., Buil, A., *et al.*, 2016. Tensor decomposition for multiple-tissue gene expression experiments. *Nature Genetics* 48 (9), 1094.
- Kimura, S., Ide, K., Kashiwara, A., *et al.*, 2004. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics* 21 (7), 1154–1163.
- Lähdesmäki, H., Shmulevich, I., Yli-Harja, O., 2003. On learning gene regulatory networks under the Boolean network model. *Machine Learning* 52 (1–2), 147–167.
- Liu, Z., Malone, B., Yuan, C., 2012. Empirical evaluation of scoring functions for Bayesian network model selection. *BMC Bioinformatics* 13 (Suppl. 15), S14.
- Liu, F., Zhang, S.W., Guo, W.F., *et al.*, 2016. Inference of gene regulatory network based on local bayesian networks. *PLOS Computational Biology* 12 (8), e1005024.

- Lu, X.M., Batugedara, G., Lee, M., *et al.*, 2017. Nascent RNA sequencing reveals mechanisms of gene regulation in the human malaria parasite *Plasmodium falciparum*. *Nucleic Acids Research* 45 (13), 7825–7840.
- Luo, F., Yang, Y., Zhong, J., *et al.*, 2007. Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinformatics* 8 (1), 299.
- Madan Babu, M., Teichmann, S.A., 2003. Evolution of transcription factors and the gene regulatory network in *Escherichia coli*. *Nucleic Acids Research* 31 (4), 1234–1244.
- Marbach, D., Costello, J.C., Küffner, R., *et al.*, 2012. Wisdom of crowds for robust gene network inference. *Nature Methods* 9 (8), 796.
- Marbach, D., Prill, R.J., Schaffter, T., *et al.*, 2010. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences* 107 (14), 6286–6291.
- Marbach, D., Schaffter, T., Mattiussi, C., Floreano, D., 2009. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology* 16 (2), 229–239.
- Margolin, A.A., Nemenman, I., Basso, K., *et al.*, 2006. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 7 (Suppl. 1), S7.
- Mounika Inavolu, S., Renbarger, J., Radovich, M., *et al.*, 2017. IODNE: An integrated optimization method for identifying the deregulated subnetwork for precision medicine in cancer. *CPT: Pharmacometrics & Systems Pharmacology* 6 (3), 168–176.
- Muhammad, D., Schmittling, S., Williams, C., Long, T.A., 2017. More than meets the eye: Emergent properties of transcription factors networks in *Arabidopsis*. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms* 1860 (1), 64–74.
- Mukherjee, S., Speed, T.P., 2008. Network inference using informative priors. *Proceedings of the National Academy of Sciences* 105 (38), 14313–14318.
- Murphy, K.P., Russell, S., 2002. Dynamic bayesian networks: Representation, inference and learning.
- Ogata, H., Goto, S., Sato, K., *et al.*, 1999. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 27 (1), 29–34.
- Peters, A.D., 2008. A combination of cis and trans control can solve the hotspot conversion paradox. *Genetics* 178 (3), 1579–1593.
- Phillips, T., 2008. The role of methylation in gene expression. *Nature Education* 1 (1), 116.
- Prill, R.J., Marbach, D., Saez-Rodriguez, J., *et al.*, 2010. Correction: Towards a rigorous assessment of systems biology models: The DREAM3 challenges. *PLOS ONE* 5 (2), e9202.
- Saelens, W., Cannoodt, R., Saeys, Y., 2018. A comprehensive evaluation of module detection methods for gene expression data. *Nature Communications* 9 (1), 1090.
- Scott, D.W., Sheather, S.J., 1985. Kernel density estimation with binned data. *Communications in Statistics-Theory and Methods* 14 (6), 1353–1359.
- Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W., 2002. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18 (2), 261–274.
- Simmons, L.A., Foti, J.J., Cohen, S.E., Walker, G.C., 2008. The SOS regulatory network. *EcoSal Plus* 2008.
- Van Gerven, M.A., Taal, B.G., Lucas, P.J., 2008. Dynamic Bayesian networks as prognostic models for clinical patient management. *Journal of Biomedical Informatics* 41 (4), 515–529.
- Vasudevaraja, V., Renbarger, J., Shah, R.G., *et al.*, 2017. PMTDS: A computational method based on genetic interaction networks for Precision Medicine Target-Drug Selection in cancer. *Quantitative Biology* 5 (4), 380–394.
- Vijesh, N., Chakrabarti, S.K., Sreekumar, J., 2013. Modeling of gene regulatory networks: A review. *Journal of Biomedical Science and Engineering* 6 (02), 223.
- Wang, Y.R., Huang, H., 2014. Review on statistical methods for gene network reconstruction using expression data. *Journal of Theoretical Biology* 362, 53–61.
- Yavari, F., Towhidkhah, F., Gharibzadeh, S., 2008. Gene regulatory network modeling using Bayesian networks and cross correlation. In: *Proceedings of the Cairo International Biomedical Engineering Conference, CIBEC 2008*, pp. 1–5. IEEE.
- Yeang, C.H., Mak, H.C., McCuine, S., *et al.*, 2005. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biology* 6 (7), R62.
- Yu, L., Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research* 5 (2004), 1205–1224.
- Zhang, W., Chien, J., Yong, J., Kuang, R., 2017. Network-based machine learning and graph theory algorithms for precision oncology. *npj Precision Oncology* 1 (1), 25.

Relevant Websites

<https://metacyc.org/>
BIOCYC.

<http://www.genecards.org/>
GeneCards.

<http://www.ncbi.nlm.nih.gov/projects/HistoneDB2.0>
HistoneDB 2.0.

<http://jaspar.genereg.net/>
JASPAR.

www.genome.jp/kegg/
KEGG.

<http://mirgator.kobic.re.kr/index.html>
miRGator v3.0.

<http://www.pathwaycommons.org/>
Pathway Commons.

Biographical Sketch



Enze Liu, Ms. Enze Liu is currently enrolled as a PhD student at the Indiana University School of Informatics and Computing. Before that, he graduated from Royal Institute of Technology, Stockholm, Sweden in 2013 with a master's degree in computational and system biology. He then worked at the Science for Life Laboratory in Stockholm, Sweden and Biomarker Bio-tech corporations in Beijing, China. His research interests mainly lie in the computational biology domain, in cancer biology and personalized medicine.



Lang Li, PhD Dr. Li gained his PhD of Biostatistics from University of Michigan in 2000. He then served Indiana University School of Medicine for 11 years and became the chair of Computational Biology and Bioinformatics and associate director of the Indiana Institute of Personalized Medicine. He recently joined The Ohio State University College of Medicine as the new chair of Department of Biomedical Informatics. Dr. Li's research interests include numerous areas across system biology, drug interaction, and personalized medicine.



Lijun Cheng, PhD Dr. Cheng gained her PhD from Donghua University in China. She started her research career as a postdoctoral fellow at the Indiana University School of Medicine in 2013. Dr. Cheng currently works in the Department of Biomedical Informatics, College of Medicine, Ohio State University as an assistant professor. Her research interests include computational and systems biology mainly in the cancer biology and personalized medicine domains.

MicroRNA and lncRNA Databases and Analysis

Xiangying Sun and Michael Gribskov, Purdue University, West Lafayette, IN, United States

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

eIncRNA	Enhancer lncRNA	ncRNA	Non-coding RNA
lincRNA	Long intergenic non-coding RNA	PIN	Partially intronic RNA, a kind on lncRNA
lncRNA	Long non-coding RNA	piRNA	Piwi-interacting RNA
miR	miRNA, microRNA	plncRNA	Promoter lncRNAs
miRNA	microRNA	pri-miRNA	miRNA primary transcript
NAT	Natural antisense transcript	TIN	Totally intronic RNA, a kind of lncRNA

Introduction

Over half of the transcription in mammals originates from non-coding regions of the genome, producing noncoding RNA (ncRNA). The function of some non-coding RNA is known, for instance, the ribosomal and tRNA components of the translation system (rRNA and tRNA), small nuclear RNA (snRNA) components of the spliceosome, small nucleolar RNAs (snoRNA) which are mostly involved in modification of RNA bases, and others such as the telomerase guide RNA, tmRNA, which releases stalled ribosomes, and RNaseP RNA, the catalytic component of the tRNA modification enzyme RNaseP. But genome-wide analysis of transcription has revealed that a large fraction of transcripts are from neither known structural classes nor from mRNA. Because RNA synthesis is an energetically expensive process, it is unlikely that miRNAs and lncRNAs, which together make up as much as 65% of transcription ([Kapranov et al., 2007](#)), are accidental noise produced by random read-through from coding gene transcription. Indeed, in the last twenty years, a great deal has been learned about microRNAs (miRNA) and their important role in cellular regulation, but the function of the remaining ncRNAs, loosely referred to as long noncoding RNAs (lncRNA) is, except for a few examples, still obscure. Computational approaches have played an important role in identifying both miRNAs and lncRNAs, and methods are still undergoing rapid development.

MicroRNA

Biological Background

MicroRNAs are typically 22 base long RNAs (lengths can vary from 16–24 bases in different species) that play an important role in gene regulation in eukaryotic organisms, usually acting by targeting the mRNA for degradation or acting as a translation repressor (see ([Catalanotto et al., 2016](#)) for a review of nuclear functions). MicroRNA genes are common in eukaryotic genomes; usually there are thousands of miRNA genes, for instance in human, the ENCODE project reported 11,000 small RNA genes ([The Encode Project Consortium, 2012](#)), and about 2600 mature miRNAs are listed in miRBase ([Griffiths-Jones, 2006](#)). MiRNA genes are located in the introns of protein coding genes (sometimes called miRtrons), in UTRs of coding transcripts, or as completely separate transcripts. The primary transcript (pri-miRNA) is processed to produce a 60–100 base precursor RNA by the splicing process, in the case of intron encoded miRNAs, or by Drosha (DCL1 in plants) in the case of non-intron miRNAs. In either case, the result is a precursor RNA (pre-miRNA), with an extended base-paired hairpin structure. After export from the nucleus, the pre-miRNA is asymmetrically cleaved, near the loop of the hairpin structure, by the Dicer endonuclease to produce a mature miRNA. One strand of the mature miRNA, usually with a strong preference for the strand originating from the 5' end of the pre-miRNA, referred to as the 5p strand, is loaded into the RNA-induced silencing complex, RISC, and the other, the 3p or * strand is degraded. Within the RISC, the miRNA is bound by an Argonaute (Ago) protein, and is used to locate its complementary target mRNA, usually binding in the 3' UTR. The first 2–7 bases of the miRNA, the seed sequence, are particularly important in binding to the target, although extended complementarity or the miRNA and target mRNA is common. The seed region is also of interest because miRNAs with the same seed sequence are generally assigned to the same family. The mRNA is ultimately degraded by one of several pathways once it is bound to RISC. There are many exceptions and differences from the canonical process described above, but the general aspects are highly conserved. In addition to classical miRNA, there are additional classes of regulatory small RNAs including piwi-interacting RNA (piRNA), which interact with piwi proteins, a subtype of Argonaute proteins, and appear to act primarily to repress transcription of transposable elements (for a review see ([Tang, 2010](#))), and small interfering RNAs (siRNA) which are also produced from double stranded precursors by a Dicer-like system. One of the difficulties in predicting miRNAs is that exceptions to almost every aspect of the canonical process, described above, have been found, and while the process is very similar, there are significant differences between plants, animals, and fungi.

From a computational viewpoint, the focus is usually on 1) miR discovery, identifying the miRNA genes or pri-miRNA transcripts from genome or transcriptome sequence and predicting the mature miRNA sequence from the gene/precursor sequence, and 2) predicting the mRNA targets.

miR discovery

Mature miRNAs can be identified experimentally by isolating and sequencing small RNAs, typically 17–28 bases long. The sequences can then be mapped to the reference genome using standard short-read mapping programs such as BWA or Bowtie2 (Friedländer *et al.*, 2012; Hendrix *et al.*, 2010). Mismatches must be allowed since miRNAs may be the target of adenosine to inosine editing (Cai *et al.*, 2009). Pre-miRNAs, which are typically capped and polyadenylated can also be identified in typical RNA-Seq experiments. In this case, the two arms of the miRNA hairpin stem are often detected as separate reads (Kozomara and Griffiths-Jones, 2013). Crosslinking-immunoprecipitation has been used to identify miRNAs bound *in vivo* to Argonaute. This should, in principle, provide much better experimental datasets, but Agarwal *et al.* (2015) suggest that many of these putative sites may be non-functional.

Computational identification of novel miRNA genes in genomic sequence relies on a combination of sequence similarity to known mature miRNAs, typically based on Blast (Altschul *et al.*, 1997) searches, and secondary structure prediction using Unafold (Markham and Zuker, 2008), RNAfold (Mathews *et al.*, 2004), or the Vienna RNA package (Lorenz *et al.*, 2011). MiRNAs are often highly conserved between species, and the conservation of the mature miRNA should be nearly perfect, however due to “arm switching” (Griffiths-Jones *et al.*, 2011), the shift of the mature miRNA from the 5p to 3p side of the precursor hairpin, matching to just mature miRNAs can be problematic, and matching to the pre-miRNA is likely to be more reliable. The second approach to identification of miRNAs lies in detection of the long base-paired hairpin stem of the miRNA. Minimum free energy RNA secondary structure prediction methods are typically used to detect potential hairpin structures (see references, above), usually after pre-screening to restrict the analysis to only 3'UTRs, to identify sequences similar to known miRNA, or to remove protein coding sequences, structural RNAs and transposable elements. Because there are many sequences that are predicted to fold as an acceptable hairpin stem, for instance Bentwich *et al.* (2005) identified about 11 million in the human genome, this basic approach is typically augmented by additional *ad hoc* criteria. These so-called context criteria examine features such as the presence of a base paired region adjacent to the precursor hairpin with a typical 2 base overhang on the 3p arm, require fewer than 4 mismatches between the 5p and 3p arms in the mature miRNA region, absence of loops in the mature miRNA region, constraints on GC-content, minimum predicted folding free energy (Zhang *et al.*, 2006) or alignment score, structural “exposure” of the seed binding region in the mRNA, exclusion of perfect inverted repeats forming the putative pre-miRNA hairpin (possible transposable element) (see, for instance, (Lucas and Budak, 2012; Meyers *et al.*, 2008)), and continuous pairing in the precursor stem. RNA sequencing data is frequently included, requiring that sequences for both the 5p and 3p arms be detected with a minimum number of reads. Most of these features have been determined by inspection from specific sets of predicted or validated miRNAs and their power and generality are often unclear.

miRNA target prediction

There is general consensus that complementarity of the mRNA with the miRNA seed sequence, conservation of the miRNA and mRNA target sequence across species, the predicted stability of the miRNA-mRNA duplexes, presence of multiple sites (abundance), and accessibility of the mRNA target site are among the most important features in predicting mRNA target sites (Peterson, 2014). However many of the same *ad hoc* features listed above may be incorporated. There are literally dozens of predictive methods (for a recent comparison see (Fan and Kurgan, 2015)). The recall (fraction of known miRNA targets identified in known data) and precision (fraction of correctly predicted targets) vary widely, and offer the classic trade-off between high recall-low precision and low recall-low precision.

Many machine learning methods have been applied to miRNA discovery (see (Demirci *et al.*, 2017) for a review), most often trained using data from miRBase as training data. Negative datasets are usually obtained from coding regions (or equivalently, exons), or by random sampling from whole genomes. It has been suggested that an average decision tree is best, and generalizes across species. Recently, a number of groups have attempted to determine which of the many proposed features are most discriminative. Agarwal *et al.* (2015) found that 3'-UTR site abundance, predicted seed-and downstream pairing stability, the base at position 1 and 8 of the miRNA seed sequence, the base at position 8 of the target site, local UA content, predicted structural accessibility, distance from the miRNA site to the stop codon of polyA site, site conservation, ORF length, 3'UTR length, and the number of offset sites in the UTR are the most important features in identifying miRNA targets. Lopes *et al.* (2014) found that minimum predicted free energy index, ensemble free energy, normalized number of sequence variants, normalized Shannon entropy, and normalized base-pair distance were the most important features in a random forest approach. Tran *et al.* (2015) found, using a boosted support vector machine approach, that the most important features are folding free energy of the longest nonexact stem, maximum number of consecutive G's in the longest nonexact stem, percentage of CC and GA dinucleotides, maximum number of consecutive C's, maximum number of consecutive G's in the hairpin, percentage of G-U pairs, folding free energy normalized by hairpin size, percentage of paired U, average folding free energy, percentage of unpaired-unpaired A-paired triplets, and size of bulges. As just these three examples show, there is considerable disagreement, even today, over what features are most relevant and powerful for miRNA target identification.

Databases/Resources

There are many online resources related to miRNA (for a recent review see (Singh, 2017)). A large fraction of these have been created for a particular organism or purpose, and then not updated. Below, we give our recommendations for the most useful and reliable resources (in our opinion).

- MiRBase (miRBase, 2016; Kozomara and Griffiths-Jones, 2013) is the original miRNA resource and still hosts the miRBase registry which provides unique names for novel miRNA genes prior to publication. Release 21 of miRBase contains 28,645 entries from 223 species, including extensive annotation of functions, experimental evidence, and links to other databases.

- DIANA-TarBase ([Paraskevopoulou, 2016](#); [DIANA-TOOLS, 2016](#); [Vlachos et al., 2015](#)) focuses on experimentally validated miRNAs and includes more than half a million experimentally supported miRNA-mRNA interactions. In addition TarBase includes computational predictions made with the MicroT-CDS method.
- Plant Non-coding RNA Database ([PNRD, 2016](#); [Yi et al., 2014](#)) focuses on all types of non-coding RNAs in plants, not just miRNAs. Since miRNAs are structurally somewhat different in plants, a plant specific resource is sometimes useful. The earlier Plant MicroRNA Database (PMRD) appeared to be inactive at the time this article was written.
- Rfam ([Rfam, 2017](#)) contains a large amount of information about miR families, including sequences, species of occurrence, secondary structure (usually predicted), and matching motifs.

lncRNA

Biological Background

By definition, long noncoding RNA (lncRNA) collectively refers to the group of transcribed RNAs, with a length longer than 200 nucleotides and low coding potential. However, the 200 nucleotide threshold is an arbitrary threshold, which was selected based on a convenient biochemical cutoff in RNA isolation protocols. BC1 and snaR, for example, are examples of ncRNAs that are shorter than 200 nucleotides but still classified as lncRNAs. Therefore, in 2011, Amaral *et al.* refined this definition: lncRNAs are noncoding RNAs that may have a function as either primary or spliced transcripts, that do not encode proteins, and are neither members of structural RNA families (tRNAs, snoRNAs, spliceosomal RNAs, etc.), nor processed into known classes of small RNAs, such as microRNAs (miRNAs), piwi-interacting (piRNAs) and small nucleolar RNA (snoRNAs)^[1]. Clearly this is still somewhat unsatisfying as lncRNAs are primarily defined as those that do not belong to known classes.

Because of the generous definition, lncRNAs are diverse in their biogenesis, stability, sub-cellular localization, evolutionary conservation, structures and functions ([Ayupe et al., 2015](#); [Johnsson et al., 2014](#)). lncRNAs are typically capped, spliced, and polyadenylated. Compared with mRNAs, lncRNAs have relatively a lower expression level and lower stability. They are more tissue specific and cell-type specific, and are often expressed in a narrower time window. lncRNAs are mostly located in the nucleus, presumably to regulate gene expression, at the epigenetic level, but a minority of lncRNAs are present in the cytoplasm where they regulate translation. For example, Xist is a well-studied lncRNA that is involved in X inactivation in placental mammals. Xist is localized in the nucleus, and is highly expressed from the inactivated X chromosome at the onset of X chromosome inactivation. Xist binds at many locations in the inactivated X chromosome (by an, as yet, not well understood process) and recruits silencing factors such as the Polycomb repressive complex 2 (PRC2) to silence X chromosome genes ([Brown et al., 1992](#); [Clemson et al., 1996](#)).

Beyond primates, little sequence conservation is typically observed in lncRNAs. Unlike mRNAs, the lack of conservation in the sequences of lncRNAs does not indicate a lack of common functions; An increasing number of examples have shown that lncRNAs are conserved in structure rather than sequence, and the secondary (or higher) structure of lncRNAs constitutes the main functional unit. For example, HOTAIR is a trans-acting lncRNA whose sequence is poorly conserved in mammals beyond primates ([Bhan and Mandal, 2016](#)). Covariance analysis across 33 mammalian sequences of HOTAIR revealed a significant number of covarying positions and half-flips localized in all four domains of HOTAIR, which maintain a similar structure ([Somarowthu et al., 2015](#)).

According to NONCODE (v 5.0, a database of lncRNAs documented in the literature), 354,855 lncRNAs have been identified in 17 species. However, the functional roles of these lncRNAs remain mostly unknown. According to lncRNAdb (a database of eukaryotic lncRNA annotations), fewer than 300 have annotated functions confirmed by overexpression or knockdown experiments. lncRNAs, in general, can either repress or activate gene expression, which is associated with cell-fate programming ([Flynn and Chang, 2014](#)) and numerous human diseases ([Esteller, 2011](#)). The number of lncRNAs whose mechanisms are known in detail are even more limited, less than 20. But these examples have already shown that lncRNAs are involved in important biological processes, such as genomic imprinting, chromatin remodeling, post-transcriptional RNA processing, and regulation of translation. Based on our current knowledge, lncRNAs consummate their regulatory roles in 3 major ways: 1). Decoys: that is they bind to regulatory proteins and preclude their access to DNA; 2). Scaffolds: they recruit epigenetic complexes to regulate chromatin states; and 3). Guides: the lncRNA binds proteins and guides the ribonucleoprotein complex to a target ([Rinn and Chang, 2012](#)). PANDA is an example of a lncRNA decoy. It sequesters a transcription factor called NF-YA, and keeps NF-YA from binding to its target genes, and prevents p53-mediated apoptosis ([Hung et al., 2011](#)). HOTAIR, which is located in the HOXC cluster, is an example of a lncRNA scaffold. It can simultaneously bind PRC2 in its 5' domain and LSD1 in its 3' domain. PRC2 has the function of histone H3 lysine-27 trimethylation, and LSD1 is involved in demethylation of histone H3 at lysine 4. This combination of interactions ensures epigenetic silencing of multiple cancer related genes ([Hajjari and Salavaty, 2015](#)). As mentioned above, Xist is an example of a lncRNA guide. Xist recruits Polycomb 1 and 2 complexes and guides them to the inactivated X chromosome to establish and maintain its silencing. Because the mechanisms of so few lncRNAs are known in detail, it is likely that many other mechanisms will be uncovered, ultimately revealing a more complex picture of the role of lncRNAs in regulatory networks.

Classification of lncRNAs

In GENCODE ([GENCODE, 2017](#)), lncRNA is classified based on its genomic location with respect to nearby protein-coding genes. This is also one of the most commonly used methods to classify lncRNAs. Initially, lncRNAs are classified as either intergenic

lncRNAs or intragenic lncRNAs. The transcripts of Intergenic lncRNAs (lincRNAs) do not overlap protein coding transcripts, while intragenic lncRNAs are transcribed from regions that overlap protein coding genes and can be further classified into sense and antisense lncRNAs. Sense lncRNAs are transcribed from regions of protein-coding genes on the same strand as the mRNA. They can overlap with both introns and exons of protein-coding genes. Totally Intronic RNAs (TINs), are lncRNAs that are located entirely within intronic regions of protein-coding genes. Partially Intronic RNAs (PINs), (Nakaya *et al.*, 2007) are lncRNAs that partially or entirely cover the introns of protein-coding gene. Antisense lncRNAs, or Natural Antisense Transcripts (NATs), are lncRNAs transcribed from the opposite strand of protein-coding genes.

Another way to classify lncRNAs is to distinguish their roles in the regulation of gene expression, distinguishing cis-acting and trans-acting RNAs. Cis-acting lncRNAs regulate the expression of genes that are positioned at the same, or a nearby, genomic locus. They may function through transcriptional interference or chromatin modification. Promoters and enhancers are two natural targets of cis-regulatory lncRNAs, which can recruit transcription factors, or chromatin modification complexes which remodel the structure of adjacent protein coding genes, to increase transcription. Promoter lncRNAs (sometimes called bidirectional promoter lncRNAs), plncRNAs, are transcribed from regions near the transcription start site (usually within 1500 bp of the transcription start site) of protein-coding genes, whereas enhancer lncRNAs (elncRNAs) are located up to 1 Mbp upstream or downstream of the regulated gene. Trans-acting lncRNAs can control the expression of a gene at an independent loci, for example, genes on a different chromosomes.

Discovery and Exploration of lncRNAs

Determining the nature and possible biological functions of lncRNAs has become a focus of intense research. Expression profiling is often a first step in uncovering the function of a lncRNA. Identifying differentially expressed lncRNAs in developmental stages or conditions can imply their potential functions. Alternatively, an informatic method termed “Guilt by Association” identifies functions of lncRNAs by looking for protein-coding genes whose expression is significantly correlated with a lncRNA (Guttman *et al.*, 2009).

Even though some researchers have successfully identified lncRNAs using polyadenylated RNA sequencing (mRNA-Seq), total cellular RNA sequencing (total RNA-Seq) is the usually the method of choice for comprehensive expression profiling of lncRNAs. This is because some lncRNAs, particularly elncRNAs, may not be spliced or polyadenylated. By using total cellular RNA-Seq, both mRNAs and lncRNAs can be identified, regardless of whether they are polyadenylated.

In the following section, we describe a computational pipeline for the identification of lncRNAs using total RNA-Seq data.

1. First the RNA-Seq reads must be mapped, separately, to a reference.
 - a. If using a reference genome, reads are first mapped to the genome using an intron aware mapper (e.g., using Tophat2 (Kim *et al.*, 2013). Transcripts from different samples are merged (e.g., using Cufflinks (Trapnell *et al.*, 2012)).
 - b. If not using a reference genome, reads from all samples are combined to construct a *de novo* transcript assembly (e.g., using Trinity (Grabherr *et al.*, 2011)).
2. Reads from the individual samples are separately mapped to the reference.
3. Possible protein coding transcripts are excluded by multiple filtering steps, for example, removing
 - a. annotated protein coding transcripts,
 - b. transcripts with high coding potential (e.g., using the Coding Potential Calculator (Kong *et al.*, 2007)),
 - c. transcripts that match conserved known proteins or motifs (e.g., using Blastx (Altschul *et al.*, 1997)),
 - d. transcripts that have a high rate of synonymous versus nonsynonymous substitutions (e.g., using PhyloCSF (Lin *et al.*, 2011)).
4. ChIP-Seq (Chromatin Immunoprecipitation Sequencing) can be used to identify lncRNAs involved in gene activation involving transcription factors, or histone modification.

lncRNA Databases/Resources

Many online resources are available for lncRNAs. As with miRNAs, the spectrum of resources rapidly changes as many database are created for particular purposes, but not maintained over time. In Table 1 we list a few of the currently active resources. Online searches for “lncRNA database” or similar terms will typically provide an updated list of resources, and a list is also maintained on Wikipedia (see the source citation in Table 1).

Closing Remarks and Future Directions

MicroRNA and lncRNA play important roles in biological regulation. In the case of miRNA, a good deal is known about the identity of miRNAs and their targets. In spite of this depth of knowledge, there are a dismaying number of online resources that have been created and, apparently, never updated. This makes it particularly hard to make recommendations about electronic resources; miRBase, Diana-TarBase, and Rfam seem to have the best long-term availability.

Table 1 lncRNA databases and resources

Database	Species	Last update	URL http:// Description
lncRNAdb	69 species	23-Nov-15	www.lncrnadb.org Includes lnc RNAs shown to be functional by overexpression or knockdown experiments
RNAcentral	37 species	1-Apr-17	rnacentral.org Combines 25 well maintained ncRNA databases. Provides integrated text search, sequence similarity search, and programmatic data access (API)
NONCODE	17 species	6-Sep-17	www.noncode.org Includes lncRNAs from published literature, GenBank, and specialized Databases such as Ensembl, RefSeq, lncRNAdb and LNCipedia. Functions of lncRNA are predicted by lnc-GFP
LNCipedia	Human	4-May-17	lncipedia.org Includes 146,742 annotated human lncRNAs. Provides basic transcript information, predicted 2° structure, calculated protein coding potential, and predicted microRNA binding sites
GreenC	45 species	19-Sep-16	greenc.sciencedesigners.com Includes lncRNAs annotated in plants and algae that are identified by using self-developed pipelines. Provides information about sequence, genomic coordinates, coding potential, and predicted folding energy
PLAR2	17 vertebrates	Unknown	www.weizmann.ac.il/Biological_Regulation/IgorUlitsky/pipeline-lncrna-annotation-rna-seq-data-plar Includes lncRNAs identified using self-developed pipelines. 3P-seq information are included
lncRNADisease	human	26-Jul-17	www.cuilab.cn/lncrnadisease Includes experimentally supported lncRNA-disease association data and lncRNA interactions in various levels, including protein, RNA, miRNA, and DNA
lnc2Cancer	human	4-Jul-16	www.bio-bigdata.com/lnc2cancer A manually curated database that include 1488 entries of associations between 666 human lncRNAs and 97 human cancers

Source: Wikipedia (http://en.wikipedia.org/wiki/List_of_long_non-coding_RNA_databases).

The situation with lncRNAs is very different. Very few lncRNAs have experimentally determined functions, and many have been predicted only computationally. We expect that methods for identifying lncRNA and for assigning functions will develop rapidly over the next few years as more biological information about their function becomes available. In the meantime, it is likely that there will considerable flux in the availability of available websites and tools.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Cell Modeling and Simulation. Cell Modeling and Simulation. Characterizing and Functional Assignment of Noncoding RNAs. Computational Approaches for Modeling Signal Transduction Networks. Computational Systems Biology. Data Storage and Representation. Natural Language Processing Approaches in Bioinformatics. Nucleic-Acid Structure Database. Prediction of Coding and Non-Coding RNA. RNA Structure Prediction. Sequence Analysis

References

- Agarwal, V., Bell, G.W., Nam, J.-W., Bartel, D.P., 2015. Predicting effective microRNA target sites in mammalian mRNAs. *eLIFE* 4, e05005.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25, 3389–3402.
- Ayup, A.C., Tahira, A.C., Camargo, L., *et al.*, 2015. Global analysis of biogenesis, stability and sub-cellular localization of lncRNAs mapping to intragenic regions of the human genome. *RNA Biology* 12, 877–892.
- Bentwich, I., Avniel, A., Karov, Y., *et al.*, 2005. Identification of hundreds of conserved and nonconserved human microRNAs. *Nature Genetics* 37, 766–770.
- Bhan, A., Mandal, S.S., 2016. Estradiol-induced transcriptional regulation of long non-coding RNA, HOTAIR. *Methods in Molecular Biology* 1366, 395–412.
- Brown, C.J., Hendrich, B.D., Rupert, J.L., *et al.*, 1992. The human XIST gene: Analysis of a 17 kb inactive X-specific RNA that contains conserved repeats and is highly localized within the nucleus. *Cell* 71, 527–542.
- Cai, Y., Yu, X., Hu, S., Yu, J., 2009. A brief review on the mechanisms of miRNA regulation. *Genomics Proteomics Bioinformatics* 7, 147–154.
- Catalanotto, C., Cogoni, C., Zardo, G., 2016. MicroRNA in control of gene expression: MicroRNA in control of gene expression. *International Journal Molecular Sciences* 17, 1712.
- Clemson, C.M., McNeil, J.A., Willard, H.F., Lawrence, J.B., 1996. XIST RNA paints the inactive X chromosome at interphase: Evidence for a novel RNA involved in nuclear/chromosome structure. *Journal of Cell Biology* 132, 259–275.
- Demirci, M.D.S., Baumbach, J., Allmer, J., 2017. On the performance of pre-microRNA detection algorithms. *Nature Communications* 8, 330.
- DIANA-TOOLS, 2016. DIANA-TOOLS. Available at: diana.imis.athena-innovation.gr (accessed 21.10.17).

- Esteller, M., 2011. Non-coding RNAs in human disease. *Nature Reviews Genetics* 12, 861–874.
- Fan, X., Kurgan, L., 2015. Comprehensive overview and assessment of computational prediction of microRNA targets in animals. *Briefings in Bioinformatics* 16, 780–794.
- Flynn, R.A., Chang, H.Y., 2014. Long noncoding RNAs in cell-fate programming and reprogramming. *Cell Stem Cell* 14, 752–761.
- Friedländer, M.R., Mackowiak, S.D., Li, N., Chen, W.I., Rajewsky, N., 2012. miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. *Nucleic Acids Research* 40, 37–52.
- GENCODE, 2017. GENCODE. Available at: <https://www.gencodegenes.org/> (accessed 21.10.17).
- Grabherr, M.G., Haas, B.J., Yassour, M., *et al.*, 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* 29, 644–652.
- Griffiths-Jones, S., Hui, J.H., Marco, A., Ronshaugen, M., 2011. MicroRNA evolution by arm switching. *EMBO Reports* 12, 172–177.
- Griffiths-Jones, S., Grocock, R.J., van Dongen, S., Bateman, A., Enright, A.J., 2006. miRBase: MicroRNA sequences, targets and gene nomenclature. *Nucleic Acids Research* 34, D140–D144.
- Guttman, M., Amit, I., Garber, M., *et al.*, 2009. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature* 458, 233–237.
- Hajjari, M., Salavaty, A., 2015. HOTAIR: An oncogenic long non-coding RNA in different cancers. *Cancer Biology & Medicine* 12, 1–9.
- Hendrix, D., Levine, M., Shi, W., 2010. miRTRAP, a computational method for the systematic identification of miRNAs from high throughput sequencing data. *Genome Biology* 11, R39.
- Hung, T., Wang, Y., Lin, M.F., *et al.*, 2011. Extensive and coordinated transcription of noncoding RNAs within cell-cycle promoters. *Nature Genetics* 43, 621–629.
- Johnsson, P., Lipovich, L., Grandér, D., Morris, K.V., 2014. Evolutionary conservation of long non-coding RNAs; sequence, structure, function. *Biochimica Biophysica Acta* 1840, 1063–1071.
- Kapranov, P., Cheng, J., Dike, S., *et al.*, 2007. RNA maps reveal new RNA classes and a possible function for pervasive transcription. *Science* 316, 1484–1488.
- Kim, D., Pertea, G., Trapnell, C., *et al.*, 2013. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology* 14, R36.
- Kong, L., Zhang, Y., Ye, Z.Q., *et al.*, 2007. CPC: Assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Research* 35, W345–W349.
- Kozomara, A., Griffiths-Jones, S., 2013. miRBase: Annotating high confidence microRNAs using deep sequencing data. *Nucleic Acids Research* 42, D68–D73.
- Lin, M.F., Jungreis, I., Kellis, M., 2011. PhyloCSF: A comparative genomics method to distinguish protein coding and non-coding regions. *Bioinformatics* 27, i275–i282.
- Lopes, I.O.N., Schliep, A., de Carvalho, A.C.P., 2014. The discriminant power of RNA features for pre-miRNA recognition. *BMC Bioinformatics* 15, 124.
- Lorenz, R., Bernhart, S.H., Höner, Z., *et al.*, 2011. ViennaRNA Package 2.0. *Algorithms for Molecular Biology* 6, 26.
- Lucas, S.J., Budak, H., 2012. Sorting the wheat from the chaff: Identifying miRNAs in genomic survey sequences of *Triticum aestivum* chromosome 1AL. *PLOS One* 7, e40859.
- Markham, N.R., Zuker, M., 2008. UNAFold: Software for nucleic acid folding and hybridization. *Methods in Molecular Biology* 453, 3–31.
- Mathews, D.H., Disney, M.D., Childs, J.L., *et al.*, 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences, USA* 101, 7287–7292.
- Meyers, B.C., Axtell, M.J., Barte, I.B., *et al.*, 2008. Criteria for annotation of plant MicroRNAs. *Plant Cell* 20, 3186–3190.
- miRBase, 2016. miRBase. Available at: <http://www.mirbase.org/> (accessed 21.10.17).
- Nakaya, H.I., Amaral, P.P., Louro, R., *et al.*, 2007. Genome mapping and expression analyses of human intronic noncoding RNAs reveal tissue-specific patterns and enrichment in genes related to regulation of transcription. *Genome Biology* 8, R43.
- Paraskevopoulou, M.D., Vlachos, I.S., Hatzigeorgiou, A.G., 2016. DIANA-TarBase and DIANA suite tools: Studying experimentally supported microRNA targets. *Current Protocols in Bioinformatics* 55, 12.14.1–12.14.18.
- Peterson, S.M., Thompson, J.A., Ufkin, M.L., *et al.*, 2014. Common features of microRNA target prediction tools. *Frontiers in Genetics* 18, 5–23.
- PNRD, 2016. PNRD. Available at: <http://structuralbiology.cau.edu.cn/PNRD/>.
- Rfam, 2017. Rfam. Available at: <http://rfam.xfam.org/> (accessed 21.10.17).
- Rinn, J.L., Chang, H.Y., 2012. Genome regulation by long noncoding RNAs. *Annual Review of Biochemistry* 81, 145–166.
- Singh, N.K., 2017. MicroRNAs databases: Developmental methodologies, structural and functional annotations. *Interdisciplinary Science and Computational Life Science* 9, 357–377.
- Somarowthu, S., Legiewicz, M., Chillón, I., *et al.*, 2015. HOTAIR forms an intricate and modular secondary structure. *Molecular Cell* 58, 353–361.
- Tang, F., 2010. Small RNAs in mammalian germline: Tiny for immortal. *Differentiation* 79, 141–146.
- The Encode Project Consortium, 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 91–100.
- Tran, V.T., Tempel, S., Zerath, B., Zehraoui, F., Tahi, F., 2015. miRBoost: Boosting support vector machines for microRNA precursor classification. *RNA* 21, 775–785.
- Trapnell, C., Roberts, A., Goff, L., *et al.*, 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols* 7, 562–578.
- Vlachos, I.S., Paraskevopoulou, M.D., Karagkouni, D., *et al.*, 2015. DIANA-TarBase v7.0: Indexing more than half a million experimentally supported miRNA:mRNA interactions. *Nucleic Acids Research* 43, D153–D159.
- Yi, X., Zhang, Z., Ling, Y., Xu, W., Su, Z., 2014. PNRD: A plant non-coding RNA database. *Nucleic Acids Research* 43, D982–D989.
- Zhang, B.H., Pan, X.H., Cox, S.B., Anderson, T.A., 2006. Evidence that miRNAs are different from other RNAs. *Cellular and Molecular Life Sciences* 63, 246–254.

Relevant Website

http://en.wikipedia.org/wiki/List_of_long_non-coding_RNA_databases
Wikipedia.

Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine

Molly A Hall, The Pennsylvania State University, University Park, PA, United States

Brian S Cole and Jason H Moore, University of Pennsylvania, Philadelphia, PA, United States

© 2019 Elsevier Inc. All rights reserved.

Genetic Interaction Modeling is Needed to Better Understand and Predict Complex Diseases

Imagine you are prescribed a medication in your doctor's office and a warning pops up on the computer screen as your doctor enters the prescription into your electronic health record. This warning tells your doctor that you have a genetic variant that predisposes you to serious side effects from the medicine being prescribed. This scenario is the goal of precision medicine: that medications, therapies, disease risk assessment, and interventions are informed by each individual patient's genetic background as well as their environmental history. Presently, however, clinicians typically do not have access to patient genetic data or a comprehensive list of past and current environmental exposures. More importantly, even if warning systems like these were available in every clinic, this scenario would not be possible for every drug therapy and disease because, to date, there is simply not enough known about the complex mechanisms that lead to common diseases and adverse drug reactions.

For many rare, Mendelian disorders like cystic fibrosis and Huntington's disease, the genetic component has been largely explained. These diseases involve one or a small set of rare but highly penetrant mutations. For example, early linkage studies found mutations in *cystic fibrosis transmembrane conductance regulator* (*CFTR*) that lead to cystic fibrosis (Kerem *et al.*, 1989; Riordan *et al.*, 1989; Rommens *et al.*, 1989). Conversely, due to their complex etiologies, the genetic basis of common traits like type 2 diabetes, height, body mass index (BMI), and many cancers have remained elusive.

Genetic association studies are often used to study complex traits like these. In a genetic association test, allele frequencies for a locus are compared between individuals who have a disease of interest (e.g., type 2 diabetes cases) and those who do not (e.g., type 2 diabetes controls). Before the advent of low-cost and high-throughput genotyping technologies, genetic association tests were performed for genetic loci that were predicted to be involved with a disease (candidate genetic association studies). Alternatively, genome-wide association studies (GWAS) allow agnostic testing of a set of single nucleotide polymorphisms (SNPs) across the genome to identify novel genetic regions predictive of disease.

GWAS have been successful at identifying numerous loci across the genome that are associated with a diverse and wide range of phenotypes. However, as predicted by Wang *et al.* (2005), if the odds ratios of all SNP-phenotype associations curated in the NHGRI GWAS Catalog (Welter *et al.*, 2014) are considered, the vast majority of SNPs have a low effect on the traits with which they are associated, and relatively few SNPs have a high effect (Wang *et al.*, 2005; Hall *et al.*, 2016a,b). Manolio *et al.* (2009) suggested a 2-fold risk increase (odds ratio of 2) as indicative of a variant exhibiting a large effect (Manolio *et al.*, 2009). Yet, the majority of GWAS-identified associations demonstrate an odds ratio less than 1.5, which has led many in the field to look beyond GWAS (Manolio *et al.*, 2009; Maher, 2008; Hall *et al.*, 2016a,b; Moore *et al.*, 2010).

Incorporating gene-gene interactions is an essential component to modeling the complexity that exists in biology and the processes that lead to common diseases. Gene products interact physically with one another or within one or multiple pathways. Further, physical interactions between and within DNA, RNA, and proteins are a major part of gene regulation (Martinez, 2002) and translation (Gallie, 2002). Considering loci individually does not allow for the complexity known to exist in molecular and cell biology and in physiology. Embracing complexity by exploring gene-gene interactions is likely to explain much of the yet unknown etiology of common diseases, as these complex traits arise from vastly dynamic and interconnected systems in biology. Such analyses will also identify the genotype combinations that confer the greatest risk to adverse health outcomes. Discussed in this article is the history of epistasis, examples of genetic interactions in model systems and humans, major challenges to identifying gene-gene interactions and common methods for overcoming these challenges, as well as consideration of additional types of complexity that must be explored in combination with epistasis to further reveal the etiology of common diseases.

Historical Perspectives on Analysis and Interpretation of Gene-Gene Interactions

The analysis of gene-gene interactions long predated the elucidation of the structure of DNA. After the rediscovery of Mendel's pioneering experiments of pea crosses that spawned the field of genetics, an explosive period of genetic discovery, driven by experiments in model systems and mathematical analysis at the population level, dominated the first two decades of the twentieth century (Sturtevant, 2001). It was during this gilded age of genetics that pioneering analysis of model systems extended and refined Mendel's laws into a cohesive theory of genetics that formed the basis of our modern understanding. During this period, epistasis was discovered by William Bateson, the biologist who coined the term "genetics" to name the nascent field of the study of heritable variation (Bateson, 1909).

Bateson used the term "epistasis" to describe a cross between two strains in which the phenotypic distribution of the resulting offspring departs from the ratios expected by Mendel's laws (Cordell, 2002). Specifically, Bateson used the term epistasis to

describe one mutation blocking or masking the effects of another, hence the use of term “epistasis” which may be translated as “resting upon.” Bateson’s usage of the term epistasis described an interaction between two genetic variants in which one variant negates the effects of another (Phillips, 2008). This type of genetic interaction, sometimes called modification, was the first form of gene-gene interactions to be observed in experimental crosses. Working together with Reginald Punnett, Bateson developed a two-locus Punnett square to describe the phenotypic ratios of F_2 progeny from crosses of two strains of the flowering sweet pea *Lathyrus odoratus* which displayed a flower coloration trait only when two separate dominant alleles were present at separate loci (Sturtevant, 2001). This two-locus epistasis model extended Mendel’s original postulations of a two-locus model to incorporate an interaction between genetic variants, without which the phenotypic ratios of Bateson’s and Punnett’s sweet peas did not conform to Mendel’s laws.

Bateson and Punnett’s description of epistasis is the result of crosses between self-fertilizing strains of plants, which are essentially controlled for genetic background, allowing the analysis of the effects of one or a small number of genetic variants on visible phenotypes such as morphological traits. Natural populations of organisms, including humans and wild populations of other organisms which are used as model systems in experimental genetics, contain genetic variation across the genome, eliminating the ability to analyze the effects of one or a small number of genetic variants against a controlled background (Moore and Williams, 2005). The statistical geneticist R. A. Fisher extended the description of epistasis to populations which are not controlled for genomic background by defining “epistasis” as deviations from additivity in a linear model (Fisher, 1919). This definition of gene-gene interactions allows for the statistical detection of epistasis in a population which contains a large number of polymorphic sites in the genome by defining epistasis as a statistical deviation from additivity, a definition which incorporates the mean effect of two or more genetic variants in a given population of organisms (Doust *et al.*, 2014.). Importantly, Bateson’s definition of epistasis involved organisms which share almost all of their genome (inbred strains) and Fisher’s definition involved organisms which contain polymorphisms across the genome (wild populations). Modern scientists have synthesized these concepts into biological and statistical epistasis (Moore and Williams, 2005). Biological epistasis refers to experimental crosses in which the distribution of phenotypes in offspring deviate from Mendelian ratios (as described by Bateson and Punnett), and statistical epistasis indicates genetic effects which deviate significantly from additivity in highly polymorphic populations (as described by Fisher). As a hypothetical example to demonstrate statistical epistasis consider two loci: LocusA and LocusB. If the relationship between the two loci is additive, we would expect the combined effect of the two on a phenotype to be the addition of the main effect of LocusA and the main effect of LocusB. For example, if there is a 2-fold and 3-fold risk associated with the risk alleles for LocusA and LocusB, respectively, the additive result from both loci is a 5-fold increase risk. If the relationship is epistatic, however, the effect of the two loci together will significantly differ from the combined main effects of the two loci. In the scenario described, the presence of both risk alleles under an epistatic relationship could be a 15-fold risk increase; alternately, risk could decrease to 1.1-fold. In other words, when statistical epistasis occurs, there is a non-linear relationship between the effects of two or more loci when combined. While these two forms of epistasis are experimentally distinct, the underlying theory is identical: epistasis, defined broadly, is the interaction between distinct genetic variants (Phillips, 1998, 2008). This definition encompasses both statistical epistasis which might be detectable in population-scale studies and biological epistasis which might be observable in controlled crosses.

Epistasis research has continued to play a central role in genetics since the early work of Bateson, Punnett, Fisher, and others at the dawn of the twentieth century. An important application of epistasis to biological discovery came in the form of pathway ordering, in which multiple strains of a model organism are crossed together and phenotypes observed such that the ordering of a biological pathway becomes evident (Avery and Wasserman, 1992). This important genetic tool can be used to discover which gene products are upstream or downstream of other gene products, providing evidence of gene product function without molecular or biochemical analysis. This can be achieved by crossing together separate mutant strains of a model organism which display different phenotypes (Beadle, 1945). If a double mutant displays the same phenotype as one of the mutants does individually, one mutation likely occurs in a gene whose product functions downstream in a biochemical pathway. While this is certainly not always the case, epistasis as a tool for pathway ordering elucidated the ordering of mutations (and thereby their gene products, even if the molecular functions were only later established) in the biological pathways which control cell cycle in yeast, sex determination in *C. elegans*, embryonic development in *D. melanogaster*, and other pathways (Phillips, 2008).

As described by Moore (2003), epistasis is thought to be ubiquitous in biology (Moore, 2003; Templeton, 2000). Examples of epistasis have been observed in many model organisms (Mackay, 2014), including yeast (Wagner, 2000; Boone *et al.*, 2007; Tong *et al.*, 2004; Szappanos *et al.*, 2011; Moore, 2005; Baryshnikova *et al.*, 2013), *C. elegans* (Lehner *et al.*, 2006; Gaertner *et al.*, 2012; Byrne *et al.*, 2007), *D. melanogaster* (Horn *et al.*, 2011; Huang *et al.*, 2012; Lloyd *et al.*, 1998), *M. musculus* (Cheng *et al.*, 2011; Hanlon *et al.*, 2006; Gale *et al.*, 2009), and *A. thaliana* (Rowe *et al.*, 2008; Kroymann and Mitchell-Olds, 2005). While examples of epistasis have accrued in model organisms over the years, epistasis is not confined to Mendelian traits, for which a small set of highly penetrant mutations explain much of the variance in observed phenotypes. Epistatic interactions between genetic loci have been discovered in human traits ranging from blood types and eye coloration to complex, polygenic, and multifactorial traits such as disease susceptibility (Moore, 2005). Nelson *et al.* (2001) identified interactions between ApoB and ApoE in females as well as between the low-density lipoprotein receptor and the ApoAI/CIII/AIV complex in males for triglyceride levels (Nelson *et al.*, 2001). Interactions between SNPs in three estrogen metabolism genes, *COMT*, *CYP1B1*, and *CYP1A1*, were identified by Ritchie *et al.* (2001) that were predictive of sporadic breast cancer (Ritchie *et al.*, 2001). Further, a study by Hemani *et al.* (2014) identified and replicated a large number of genetic interactions involved in gene expression regulation (Hemani *et al.*, 2014).

Epistasis research was spawned shortly after the rediscovery of Mendel's foundational work that gave rise to the field of genetics and has found application in the understanding of how genetic variants interact to determine phenotypic outcomes. While genetic interactions have provided insight into traits which cannot be adequately explained by additive models or Mendelian ratios, epistasis research remains an active application of genetics in the modern scientific research enterprise. Particularly, detection of epistasis in studies of complex traits in humans presents methodological and computational challenges that remain an active area of development.

Major Challenges to Exploring Genetic Interactions and Common Solutions

Genome-wide association studies most commonly utilize regression to identify statistically significant genotype-phenotype associations and to quantify the certainty and effect size of the association. These regression analyses typically utilize multivariable linear or generalized linear models in which genotypes and covariates of interest such as age, gender, and Body Mass Index are represented as additive terms in a linear equation for which the parameters are estimated by regression. As described above, epistasis is defined as significant deviation from additivity. It is necessary to consider this definition when testing for epistasis using regression. Testing for independent locus main effects in regression, even at the genome-wide level with GWAS, typically involves a model such as $Phenotype \sim \beta_0 + \beta_1 SNP + covariates$. In this linear model formula, the main effect of a SNP (as represented by the parameter estimate of β_1 and its associated significance) on a phenotype (or outcome) of interest is tested. In GWAS, a model is tested for each SNP included in the data set. A likelihood ratio test is recommended when performing interaction tests using regression. The likelihood ratio test compares the difference between a full and reduced model, where the full model in this case is $Phenotype \sim \beta_0 + \beta_1 SNP + \beta_2 SNP2 + \beta_3 SNP1 \times SNP2 + covariates$ and the reduced model is $Phenotype \sim \beta_0 + \beta_1 SNP + \beta_2 SNP2 + covariates$. The difference between these two models is the interaction term, and thus, a significant likelihood ratio p-value indicates a significant effect of the interaction (β_3) above and beyond the additive effects from SNP1 (β_1) and SNP2 (β_2). If the result of interest is the effect of the interaction of the two SNPs when adjusting for the main effects of each SNP, it would be sufficient to consider the effect of the interaction term in the full model; however, when attempting to identify statistical epistasis, it is recommended to use the likelihood ratio p-value to identify significant SNP-SNP interactions (Hall *et al.*, 2017).

The interaction models above involve two SNPs, but how are the SNPs included in the model chosen? One option is to consider every pairwise combination of SNPs within a data set. This approach is not ideal, however, when dealing with large sets of genetic variants. The post-genome era has led to a consistent decline in sequencing cost and a commensurate explosion in high-throughput genotype data. Technologies such as exome sequencing and whole genome sequencing can identify millions to tens of millions of variants within a population of humans, lending the ability for researchers to study the associations of traits with genotypes at historically unprecedented depth and breadth. With this increase in genotyping power comes an increase in the number of potential interactions between genotypes, creating a steep multiple hypothesis testing burden (Eichler *et al.*, 2010; Maher, 2008; Manolio *et al.*, 2009). Because of this, exhaustive searches for significant interactions between multiple genetic loci within the context of trait association become incredibly difficult because associations of ever greater significance are needed to pass a multiple hypothesis testing correction with increasing numbers of genotyped loci. For instance, an exhaustive test of every two-way SNP-SNP combination from 500,000 SNPs will amount to 1×10^{11} pairwise SNP-SNP tests. If the Bonferroni correction at an alpha level of 0.05 is utilized as is common statistical practice, a model would need a p-value below 5×10^{-13} to be considered statistically significant when adjusting for the number of tests. With such a stringent significance threshold, it is likely that many true positives will go undiscovered with this approach. This problem is exponentially exacerbated when higher order genetic interactions are taken into account, for example the interaction between three loci instead of two (Greene *et al.*, 2010).

The rapid expanse of genotyping depth and breadth, accompanied by the exponentially increasing burden of multiple hypothesis testing, combine to create a situation in which exhaustive searches for significant interactions between genetic variants become untenable. Because of this, new techniques for searching for significant interactions only where they are likely to occur are motivated. These techniques fall into broad categories, including main effect filtering, knowledge-based searches, and machine learning approaches.

Logistically speaking, main effect filtering is a straightforward and simple method used to reduce the number of SNP-SNP models to test. It involves first testing SNPs for association with a trait of interest and choosing those exhibiting a main effect at a chosen significance threshold for subsequent SNP $\times$ SNP interaction. However, the major limitation of this method is that the SNPs chosen are only those that have a significant effect on the phenotype on their own, which precludes the possibility of identifying SNPs that only have an effect when combined with other variants (Ritchie, 2011).

Knowledge-based filtering offers an appealing alternative to main effect filtering, as the SNP-SNP pairs are chosen for interaction testing based on demonstrated biological relationships (Ma *et al.*, 2015; Ritchie, 2011, 2015; Sun *et al.*, 2014; Thomas *et al.*, 2009). Previous studies have shown success in applying prior knowledge to genetic interaction analyses for ovarian cancer (Kim *et al.*, 2014), multiple sclerosis (Bush *et al.*, 2011; Ritchie, 2009), HIV pharmacogenetics (Grady *et al.*, 2010), age-related cataract (Hall *et al.*, 2015; Pendergrass *et al.*, 2015), HDL cholesterol (Turner *et al.*, 2011; Ma *et al.*, 2015), and other lipid traits (De *et al.*, 2015). Knowledge-based searches for epistasis utilize external knowledge such as the organization of genetic variants into genes, biological pathways, and biophysical interaction networks to identify sets of genetic variants which are likely candidates for epistatic interactions without the need to evaluate or investigate all sets of genetic variants (Greene *et al.*, 2015; Moore *et al.*, 2012;

Table 1 Traditional encodings for genotypes in linear and multivariable linear regression analyses, including generalized linear models

<i>Genotype</i>	<i>AA</i>	<i>Aa</i>	<i>aa</i>
Additive encoding	0	1	2
Dominant encoding	0	1	1
Recessive encoding	0	0	1
Codominant encoding	0, 0	0, 1	1, 0

A biallelic locus with two possible alleles, here denoted as A and a, may be encoded in multiple manners intended to reflect the putative mechanism of action according to Mendel's laws. In this example, the non-reference allele (a) is considered dominant to the reference allele (A) for the purposes of defining dominance and recessivity. In the codominant encoding, two "dummy" or "one-hot" variables are used to denote the three genotype states.

Moore and White, 2006). As biological knowledge of the organization of genetic variants into genes, pathways, and networks increases, knowledge-guided design of epistasis search strategies is likely to improve. Importantly, advances in high-throughput biological interaction screens and functional genomics analyses are currently unlocking knowledge which can be expected to inform targeted searches for epistasis by discarding unlikely interactions, for example between genetic variants within genes that are not co-expressed. Conversely, interactions which are empirically likely based on biological knowledge may be prioritized for search, such as at protein-protein interaction domains between proteins which biophysically interact to transduce an intracellular signal.

In contrast to knowledge-based approaches, advances in machine learning techniques have led to development of algorithmic approaches to prioritize genetic variants which are likely to interact. Feature selection and feature construction methods aim to reduce the complexity of the search space, in this case representing combinations of genetic variants under study (Kira and Rendell, 1992a). The Multifactor Dimensionality Reduction (MDR) algorithm developed by Ritchie and Moore and modified by many others reduces the complexity of an epistasis search by constructing a new two-level feature from the distributions of the trait under study among the 9 possible two-locus genotype combinations between two biallelic variants (Ritchie *et al.*, 2001). Importantly, this new feature can be tested in linear and generalized linear models, as well as being appropriate for input into classifiers such as decision trees (Lou *et al.*, 2007; Moore and Andrews, 2015). MDR has been generalized to handle missing data, imbalanced classes, quantitative traits, permutation analysis, adjustment for covariates, and family pedigree information. Originally written in Java, MDR has recently been implemented in the scikit-learn framework for Python, a popular machine learning distribution (Pedregosa *et al.*, 2011).

In contrast to MDR, which is a feature construction method, feature selection methods can reduce the search space by identifying subsets of features themselves which are likely to interact without creating new features like that created by MDR (Kira and Rendell, 1992a). The Relief family of algorithms has seen application in epistasis discovery (Kira and Rendell, 1992b). Modifications of the Relief algorithm, including SURF, SURF*, multiSURF, and TURF can identify subsets of genetic variants which may be tested directly for epistasis (Greene *et al.*, 2009, 2010; Moore and White, 2007).

Beyond the issues relevant to reducing the multiple testing burden, a further challenge to identifying SNP models with epistatic action is the way SNPs are genetically encoded. Traditional encoding methods include additive, dominant, and recessive, and each make critical assumptions about a given SNP's biological action (Table 1). It is common for investigators to choose one encoding, most often additive. However, every SNP across the genome does not demonstrate the same genetic action. If a SNP is encoded under a different assumption than its true genetic action, this can lead to a reduction in power or inflation of false positive results. The traditional approach is inflexible and does not reflect the diversity known to exist in genetics. One approach to circumvent the bias in choosing a traditional encoding method is to use the codominant encoding when testing for interactions, which is a dummy encoding for the different genotype states. Still, further method development is needed to create encoding alternatives that are flexible to the diverse genetic action of SNPs.

Feature construction and feature selection methods are examples of machine learning methods which can provide information about likely interactions without incorporating expert knowledge, instead using the data itself. In contrast, knowledge-based methods leverage a growing base of orthogonal biological knowledge gleaned from increasingly high-throughput experimentation, including interaction screens. In the future, both machine learning methods and biological knowledge are likely to expand. Knowledge of pathways and biophysical interactions is steadily increasing, and software which can organize genetic variants under study into higher-order units such as genes, pathways, and networks will likely find increasing application in the search for epistasis. Commensurately, statistical advances in the form of novel association tests which can control type I and type II error and methods to assign appropriate genetic encodings offer to improve the quality of epistatic hypotheses generated in genome-wide studies, offering the power to quantify uncertainty in the search for gene-gene interactions (Wang *et al.*, 2016; Pattin *et al.*, 2009; Kooperberg and LeBlanc, 2008).

In sum, epistasis research faces a considerable challenge from the multiple hypothesis testing problem which has motivated the development and application of new approaches that aim to reduce the complexity of an exhaustive search through external knowledge of the search space or machine learning methods. It is reasonable to expect further developments in both areas in the future as biological knowledge accumulates. Simultaneously, advances in statistical and machine learning methodologies could lead to breakthroughs in epistasis discovery by utilizing patterns within data, including the most powerful ways in which to genetically encode SNPs. Equipped with both machine learning tools and external knowledge, the epistasis researcher of the future could circumvent the incredible complexity of all possible interactions between genotypes within a population and focus directly on discovery of biologically relevant interactions.

Beyond Epistasis: Investigating Additional Complexity in Precision Medicine

In this article, we have described the importance of exploring genetic interactions as well as the challenges surrounding these analyses and common methods to overcome those hurdles. For precision medicine to be a reality, though, investigating genetic interaction in combination with other types of complexity is a necessity. Here we will discuss the ways in which the environment, rare and structural variation, and multi-omics data are important features to consider for disease prediction in the context of epistasis.

The exposome, herein defined as all environmental exposures in an individual's life, has a major impact on health outcomes (Rappaport and Smith, 2010). A large portion of the complexities leading to common diseases are missed if the environment is not considered. Gene-gene-environment interaction models are especially difficult to test for the same multiple-testing reasons outlined previously. Currently, much of the work done in gene-environment interactions include candidate environmental exposures, but this prevents identification of gene interactions with novel exposures. Environment-wide association studies (EWAS) have successfully identified novel exposures associated with disease which can subsequently be used for gene-exposure interaction analysis (Patel *et al.*, 2010, 2013, 2014; Patel and Ioannidis, 2014; Tzoulaki *et al.*, 2012; Hall *et al.*, 2014, 2017). This approach decreases the search space to only include exposures identified through EWAS. Just as knowledge-based methods have been successful at identifying gene-gene interactions predictive of disease, database knowledge may be employed for integrating the exposome with epistatic models. The Human Metabolome Database (HMDB) (Wishart *et al.*, 2007), the Toxic Exposome Database (Wishart *et al.*, 2015), and the Comparative Toxicogenomics Database (CTD) (Davis *et al.*, 2017) are examples of databases with knowledge of gene-gene and gene-exposure relationships, which may be leveraged to build putative gene-gene-exposure models for testing. Understanding the connection between gene networks and the environment is an essential component to uncovering the etiology of disease.

Whether it be gene-gene or gene-environment interactions, the vast majority of studies have focused on common SNPs (Hall *et al.*, 2016a,b). However, rare variants and copy number variation (CNV) have demonstrated influence on many diseases, and methods that consider these types of data are therefore essential in discovering complex genetic and gene-environment interactions. Successful examples have been found for epistasis involving rare variation and between CNV and the environment (Kim *et al.*, 2017; Hall *et al.*, 2017; Albers *et al.*, 2012). Further, other -omics data such as from the transcriptome, proteome, and metabolome are important features in elucidating the biological impact of interactions at the gene level. There is need for development of methods to handle these types of data that have unique challenges such as limited power for detection and heterogeneity.

Finally, to investigate epistasis and gene-environment interactions with common and rare variants, CNVs, and multi-omics data, studies that generate multiple types of data (e.g., exposure, genotype, sequence, CNV, gene expression, metabolomic, proteomic) are critical. Costs associated with generating multiple data types on large sample sizes are limiting, however. Examples of studies with multiple data types available include the electronic MEDical Record and GENomics Network (eMERGE Network), the National Health and Nutrition Examination Surveys (NHANES), the Marshfield Personalized Medicine Research Project (PMRP), the Women's Health Initiative (WHI), and the Million Veterans Project (MVP).

As more data types become available for an increasing number of samples, the opportunity arises for improved understanding of biology and the complex processes at play in development of disease. Yet, further unmet methodological development needs will also emerge in finding the most appropriate ways to aggregate, quality control, integrate, and analyze multiple disparate data types. As such, there is immense opportunity to uncover the hidden etiology of disease as further data and methods become available.

Conclusions

To realize the goal of precision medicine, genetic interactions are an important part of modeling complexity that leads to common disease, and their consideration is essential. There are many examples of successful searches for epistasis in humans and non-human model systems. However, as data availability increases, so do the challenges of finding true genetic interactions amidst the noise. Many approaches offer promise in circumventing these hurdles. Still, to understand the mechanisms that lead to diseases, and leverage that knowledge in order make informed and individualized decisions about health care, epistasis must be considered in the context of the environment, rare variation, CNVs, and other types of inter-individual variation. For these reasons, multi-omics data must be incorporated in the search for epistasis. This is an exciting time for the field of biomedical informatics, as there is an abundance of available data and methods, and the further unmet needs for complex and integrative analyses using multiple data types leave ample room for exploration, innovation, and discovery.

See also: Computational Systems Biology. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Gene Regulatory Network Review. MicroRNA and lncRNA Databases and Analysis. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Networks in Biology. Pathway Informatics

References

- Albers, C.A., *et al.*, 2012. Compound inheritance of a low-frequency regulatory SNP and a rare null mutation in exon-junction complex subunit RBM8A causes TAR syndrome. *Nature Genetics* 44 (4), 435–439. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/22366785> (accessed 31.10.17).
- Avery, L., Wasserman, S., 1992. Ordering gene function: The interpretation of epistasis in regulatory hierarchies. *Trends in Genetics* 8 (9), 312–316.
- Baryshnikova, A., *et al.*, 2013. Genetic interaction networks: Toward an understanding of heritability. *Annual Review of Genomics and Human Genetics* 14, 111–133. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/23808365>.
- Bateson, W., 1909. *Mendel's Principles of Heredity*. Cambridge University Press.
- Beadle, G.W., 1945. Genetics and metabolism in neurospora. *Physiological Reviews* 25 (4).
- Boone, C., Bussey, H., Andrews, B.J., 2007. Exploring genetic interactions and networks with yeast. *Nature Reviews. Genetics* 8 (6), 437–449.
- Bush, W.S., *et al.*, 2011. A knowledge-driven interaction analysis reveals potential neurodegenerative mechanism of multiple sclerosis susceptibility. *Genes and Immunity* 12 (5), 335–340. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3136581&tool=pmcentrez&rendertype=abstract> (accessed 17.08.15).
- Byrne, A.B., *et al.*, 2007. A global analysis of genetic interactions in *Caenorhabditis elegans*. *Journal of Biology* 6 (3), 8.
- Cheng, Y., *et al.*, 2011. Mapping genetic loci that interact with myostatin to affect growth traits. *Heredity* 107 (6), 565–573.
- Cordell, H.J., 2002. Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics* 11 (20), 2463–2468.
- Davis, A.P., *et al.*, 2017. The Comparative Toxicogenomics Database: Update 2017. *Nucleic Acids Research* 45 (D1), D972–D978. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27651457> (accessed 31.10.17).
- De, R., *et al.*, 2015. Identifying gene-gene interactions that are highly associated with Body Mass Index using Quantitative Multifactor Dimensionality Reduction (QMDR). *BioData Mining* 8 (1), 41. Available at: <http://biodatamining.biomedcentral.com/articles/10.1186/s13040-015-0074-0> (accessed 09.04.16).
- Doust, A.N., *et al.*, 2014. Beyond the single gene: How epistasis and gene-by-environment effects influence crop domestication. *Proceedings of the National Academy of Sciences of the United States of America* 111, 6178–6183.
- Eichler, E.E., *et al.*, 2010. Missing heritability and strategies for finding the underlying causes of complex disease. *Nature Reviews Genetics* 11 (6), 446–450.
- Fisher, R.A., 1919. XV. The Correlation between relatives on the supposition of Mendelian inheritance. *Transactions of the Royal Society of Edinburgh* 52 (2), 399–433. Available at: http://www.journals.cambridge.org/abstract_S0080456800012163 (accessed 31.10.17).
- Gaertner, B.E., *et al.*, 2012. More than the sum of its parts: A complex epistatic network underlies natural variation in thermal preference behavior in *Caenorhabditis elegans*. *Genetics* 192 (4), 1533–1542.
- Gale, G.D., *et al.*, 2009. A genome-wide panel of congenic mice reveals widespread epistasis of behavior quantitative trait loci. *Molecular Psychiatry* 14 (6), 631–645.
- Gallie, D.R., 2002. Protein-protein interactions required during translation. *Plant Molecular Biology* 50 (6), 949–970.
- Grady, B.J., *et al.*, 2010. Finding unique filter sets in PLATO: A precursor to efficient interaction analysis in GWAS data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 315–26. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2903053&tool=pmcentrez&rendertype=abstract> (accessed 17.08.15).
- Greene, C.S., *et al.*, 2009. Spatially Uniform Relief (SURF) for computationally-efficient filtering of gene-gene interactions. *BioData Mining* 2, 5.
- Greene, C.S., *et al.*, 2015. Understanding multicellular function and disease with human tissue-specific networks. *Nature Publishing Group* 47 (6).
- Greene, C.S., *et al.*, 2010. Enabling personal genomics with an explicit test of epistasis. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp.327–36.
- Hall, M.A., *et al.*, 2014. Environment-wide association study (EWAS) for type 2 diabetes in the Marshfield Personalized Medicine Research Project Biobank. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 200–11. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4037237&tool=pmcentrez&rendertype=abstract> (accessed 08.07.15).
- Hall, M.A., *et al.*, 2015. Biology-driven gene-gene interaction analysis of age-related cataract in the eMERGE network. *Genetic Epidemiology* 39 (5), 376–384. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25982363> (accessed 23.07.15).
- Hall, M.A., *et al.*, 2017. PLATO software provides analytic framework for investigating complexity beyond genome-wide association studies. *Nature Communications* 8 (1), 1167. Available at: <http://www.nature.com/articles/s41467-017-00802-2> (accessed 31.10.17).
- Hall, M.A., Moore, J.H., Ritchie, M.D., 2016a. Embracing complex associations in common traits: Critical considerations for precision medicine. *Trends in Genetics* 32 (8), 470–484. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/27392675> (accessed 12.05.17).
- Hall, M.A., Moore, J.H., Ritchie, M.D., *et al.*, 2016b. Embracing complex associations in common traits: Critical considerations for precision medicine. *Trends in Genetics* 32 (8), 470–484. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0168952516300506> (accessed 05.09.16).
- Hanlon, P., *et al.*, 2006. Three-locus and four-locus QTL interactions influence mouse insulin-like growth factor-I. *Physiological Genomics* 26 (1), 46–54.
- Hemani, G., *et al.*, 2014. Detection and replication of epistasis influencing transcription in humans. *Nature* 508 (7495), 249–253. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24572353>.
- Horn, T., *et al.*, 2011. Mapping of signaling networks through synthetic genetic interaction analysis by RNAi. *Nature Methods* 8 (4), 341–346.
- Huang, W., *et al.*, 2012. Inaugural article: Epistasis dominates the genetic architecture of *Drosophila* quantitative traits. *Proceedings of the National Academy of Sciences* 109 (39), 15553–15559.
- Kerem, B., *et al.*, 1989. Identification of the cystic fibrosis gene: Genetic analysis. *Science (New York, NY)* 245 (4922), 1073–1080.
- Kim, D., *et al.*, 2014. Knowledge-driven genomic interactions: An application in ovarian cancer. *BioData Mining* 7, 20. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4161273&tool=pmcentrez&rendertype=abstract> (accessed 17.08.15).
- Kim, D., *et al.*, 2017. The joint effect of air pollution exposure and copy number variation on risk for autism. *Autism Research* 10 (9), 1470–1480. Available at: <http://doi.wiley.com/10.1002/aur.1799> (accessed 26.10.17).
- Kira, K., Rendell, L.A., 1992a. A practical approach to feature selection. In: *Proceedings of the Ninth International Workshop on Machine Learning*. pp. 249–256.
- Kira, K., Rendell, L.A., 1992b. The feature selection problem: Traditional methods and a new algorithm. *AAAI*. pp. 129–134.
- Kooperberg, C., LeBlanc, M., 2008. Increasing the power of identifying gene $\times$ gene interactions in genome-wide association studies. *Genetic Epidemiology*.
- Kroymann, J., Mitchell-Olds, T., 2005. Epistasis and balanced polymorphism influencing complex trait variation. *Nature* 435 (7038), 95–98.
- Lehner, B., *et al.*, 2006. Systematic mapping of genetic interactions in *Caenorhabditis elegans* identifies common modifiers of diverse signaling pathways. *Nature Genetics* 38 (8), 896–903.
- Lloyd, V., Ramaswami, M., Krämer, H., 1998. Not just pretty eyes: *Drosophila* eye-colour mutations and lysosomal delivery. *Trends in Cell Biology* 8 (7), 257–259.
- Lou, X.-Y., *et al.*, 2007. A generalized combinatorial approach for detecting gene-by-gene and gene-by-environment interactions with application to nicotine dependence. *The American Journal of Human Genetics* 80 (6), 1125–1137.

- Mackay, T.F.C., 2014. Epistasis and quantitative traits: Using model organisms to study gene–gene interactions. *Nature Reviews. Genetics* 15 (1), 22–33. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3918431&tool=pmcentrez&rendertype=abstract> (accessed 21.05.15).
- Maher, B., 2008. Personal genomes: The case of the missing heritability. *Nature* 456 (7218), 18–21. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18987709> (accessed 09.01.15).
- Ma, L., Keinan, A., Clark, A.G., 2015. Biological knowledge-driven analysis of epistasis in human GWAS with application to lipid traits. *Methods in Molecular Biology* (Clifton, NJ) 1253, 35–45. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25403526>.
- Manolio, T.A., et al., 2009. Finding the missing heritability of complex diseases. *Nature* 461 (7265), 747–753.
- Martinez, E., 2002. Multi-protein complexes in eukaryotic gene transcription. *Plant Molecular Biology* 50 (6), 925–947.
- Moore, C.B., et al., 2012. BioBin: A bioinformatics tool for automating the binning of rare variants using publicly available biological knowledge. *From Second Annual Translational Bioinformatics Conference BMC Medical Genomics* 6 (2), 13–16.
- Moore, J.H., 2003. The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Human Heredity*, 73–82.
- Moore, J.H., 2005. A global view of epistasis. *Nature Genetics* 37 (1), 13–14.
- Moore, J.H., Andrews, P.C., 2015. Epistasis analysis using multifactor dimensionality reduction. *Methods in Molecular Biology* (Clifton, NJ), 301–314.
- Moore, J.H., Asselbergs, F.W., Williams, S.M., 2010. Bioinformatics challenges for genome-wide association studies. *Bioinformatics* (Oxford, England) 26 (4), 445–455. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2820680&tool=pmcentrez&rendertype=abstract> (accessed 15.07.15).
- Moore, J.H., White, B.C., 2007. LNC5 4447 – Tuning Relief for Genome-Wide Genetic Analysis.
- Moore, J.H., White, B.C., 2006. Exploiting Expert Knowledge in Genetic Programming for Genome-Wide Genetic Analysis. Berlin, Heidelberg: Springer, pp. 969–977.
- Moore, J.H., Williams, S.M., 2005. Traversing the conceptual divide between biological and statistical epistasis: Systems biology and a more modern synthesis. *BioEssays* 27 (6), 637–646.
- Nelson, M.R., et al., 2001. A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation. *Genome Research* 11 (3), 458–470.
- Patel, C.J., et al., 2013. Systematic evaluation of environmental and behavioural factors associated with all-cause mortality in the United States national health and nutrition examination survey. *International Journal of Epidemiology* 42 (6), 1795–1810. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3887569&tool=pmcentrez&rendertype=abstract> (accessed 16.08.15).
- Patel, C.J., et al., 2014. Investigation of maternal environmental exposures in association with self-reported preterm birth. *Reproductive Toxicology* (Elmsford, NY) 45, 1–7. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4316205&tool=pmcentrez&rendertype=abstract> (accessed 16.08.15).
- Patel, C.J., Bhattacharya, J., Butte, A.J., 2010. An Environment-Wide Association Study (EWAS) on type 2 diabetes mellitus. *PLoS One* 5 (5), e10746. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2873978&tool=pmcentrez&rendertype=abstract> (accessed 29.06.15).
- Patel, C.J., Ioannidis, J.P.A., 2014. Studying the elusive environment in large scale. *Journal of American Medical Association* 311 (21), 2173–2174. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4110965&tool=pmcentrez&rendertype=abstract> (accessed 16.08.15).
- Pattin, K.A., et al., 2009. A computationally efficient hypothesis testing method for epistasis analysis using multifactor dimensionality reduction. *Genetic Epidemiology* 33 (1), 87–94.
- Pedregosa, Fabianpedregosa, F., et al., 2011. Scikit-learn: Machine learning in Python Gaël Varoquaux. *Journal of Machine Learning Research* 12, 2825–2830. Available at: <http://delivery.acm.org/10.1145/2080000/2078195/p2825-pedregosa.pdf>
- Pendergrass, S.A., et al., 2015. Next-generation analysis of cataracts: Determining knowledge driven gene-gene interactions using biofilter, and gene-environment interactions using the Phenx Toolkit*. *Pacific Symposium on Biocomputing*, Pacific Symposium on Biocomputing, pp. 495–505. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25741542> (accessed 17.08.15).
- Phillips, P.C., 1998. The language of gene interaction. *Genetics* 149 (3).
- Phillips, P.C., 2008. Epistasis the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics* 9 (11), 855–867.
- Rappaport, S.M., Smith, M.T., 2010. Epidemiology. Environment and disease risks. *Science* (New York, NY) 330 (6003), 460–461. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/20966241> (accessed 16.08.15).
- Riordan, J.R., et al., 1989. Identification of the cystic fibrosis gene: Cloning and characterization of complementary DNA. *Science* (New York, NY) 245 (4922), 1066–1073.
- Ritchie, M.D., et al., 2001. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics* 69 (1), 138–147.
- Ritchie, M.D., 2009. Using prior knowledge and genome-wide association to identify pathways involved in multiple sclerosis. *Genome Medicine* 1 (6), 65. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2703874&tool=pmcentrez&rendertype=abstract> (accessed 30.06.15).
- Ritchie, M.D., 2011. Using biological knowledge to uncover the mystery in the search for epistasis in genome-wide association studies. *Annals of Human Genetics* 75 (1), 172–182.
- Ritchie, M.D., 2015. Finding the epistasis needles in the genome-wide haystack. *Methods in Molecular Biology* (Clifton, NJ) 1253, 19–33. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25403525>.
- Rommens, J.M., et al., 1989. Identification of the cystic fibrosis gene: Chromosome walking and jumping. *Science* (New York, NY) 245 (4922), 1059–1065.
- Rowe, H.C., et al., 2008. Biochemical networks and epistasis shape the *Arabidopsis thaliana* metabolome. *The Plant Cell* 20 (5), 1199–1216.
- Sturtevant, A.H., 2001. A History of Genetics. CSHL Press.
- Sun, X., et al., 2014. Analysis pipeline for the epistasis search – Statistical versus biological filtering. *Frontiers in Genetics* 5 (APR).
- Szappanos, B., et al., 2011. An integrated approach to characterize genetic interaction networks in yeast metabolism. *Nature Genetics* 43 (7), 656–662.
- Templeton, 2000. Epistasis and Complex Traits. New York: Oxford University Press.
- Thomas, D.C., et al., 2009. Use of pathway information in molecular epidemiology. *Human Genomics* 4 (1), 21–42.
- Tong, A.H.Y., et al., 2004. Global mapping of the yeast genetic interaction network. *Science* (New York, NY) 303 (5659), 808–813.
- Turner, S.D., et al., 2011. Knowledge-driven multi-locus analysis reveals gene-gene interactions influencing HDL cholesterol level in two independent EMR-linked biobanks. *PLoS one* 6 (5), e19586. Available at: <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0019586> (accessed 17.08.15).
- Tzoulaki, I., et al., 2012. A nutrient-wide association study on blood pressure. *Circulation* 126 (21), 2456–2464. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4105584&tool=pmcentrez&rendertype=abstract> (accessed 16.08.15).
- Wagner, A., 2000. Robustness against mutations in genetic networks of yeast. *Nature Genetics* 24 (4), 355–361.
- Wang, M.H., et al., 2016. A fast and powerful W-test for pairwise epistasis testing. *Nucleic Acids Research* 44 (12), e115.
- Wang, W.Y.S., et al., 2005. Genome-wide association studies: Theoretical and practical concerns. *Nature Reviews Genetics* 6 (2), 109–118. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15716907> (accessed 12.05.15).
- Welter, D., et al., 2014. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research* 42 (D1),
- Wishart, D., et al., 2015. T3DB: The toxic exposome database. *Nucleic Acids Research* 43 (D1), D928–D934. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/25378312> (accessed 31.10.17).
- Wishart, D.S., et al., 2007. HMDB: The Human Metabolome Database. *Nucleic Acids Research* 35 (Database), D521–D526. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17202168> (accessed October 31.10.17).

Genome Informatics

Anil K Kesarwani, Ankit Malhotra, Anuj Srivastava, Guruprasad Ananda, Haitham Ashoor, Parveen Kumar, Rupesh K Kesharwani, Vishal K Sarsani, Yi Li, Joshy George, and R Krishna Murty Karuturi, The Jackson Laboratory, Farmington, CT, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Inherited characteristics of an organism are encoded in a molecule called deoxyribonucleic acid (DNA), represented as a string of four-element code (ACGT). This molecule encodes the information required for the synthesis of all the proteins required for the functioning of cell that ultimately determines the phenotype of the organism. Variations introduced in this code introduces the changes in the organisms for natural selection to work on. The variable fitness level of organisms results in certain variations to be selected over time resulting in the incredibly different variety of viable organisms we see around. The genetic information in the DNA is transcribed into RNA by an enzyme called RNA polymerase. The RNA is then decoded into a ribosome, outside the nucleus, to produce a specific amino acid chain, or polypeptide. The polypeptide later folds into an active protein and performs its functions in the cell which then determines the phenotype of the cell and the organism ([Felix, 2016](#)).

The particular genetic encoding for an organism is called its *genotype* and the resulting physical characteristics is called its *phenotype*. In sexually reproducing organisms including humans, sexual recombination as well as mutations introduce variation in the *genotype* resulting in the variations in its *phenotype* enabling natural selection to select organisms with better fitness in their current environment. Small changes in the genotype space can have significant differences in the phenotypes resulting in the enormous diversity of organisms we see around. Our understanding of the genotype-phenotype relationship has helped us to understand the genetic cause of several diseases and have helped us to treat them with better efficacy ([Lehner, 2007](#)).

The DNA of an organism is typically packaged into several chromosomes in eukaryotic cells, exist in discrete chromosome territories. There is compelling evidence that suggest that each chromosome is comprised of many distinct chromatin domains, referred to variably as topological domains or topologically associating domains (TADs), that are hundreds of kilobases to several million bases in length. These chromatin domains are stable for many cell divisions, cell-type specific and demonstrated to play important roles in transcriptional regulation, DNA replication, and recombination. Additionally, regulatory DNA elements such as enhancers, silencers and insulators are embedded in chromosomes and they control gene expression during development and maintenance ([Pope et al., 2014](#)).

A significant step in our understanding of our genetic makeup happened when the first release of the human genome was announced by the Human Genome Project (HGP) in 2001. The Genome Reference Consortium (GRC) – an international, collaborative research program whose goal was the complete mapping and understanding of all the genes of a genome – facilitate the curation of genome assemblies for human, mouse or zebrafish ([Hudson, 2002](#)).

Once the genome sequence was established, the next task is to identify the functional elements in the genome. The ENCODE (Encyclopedia of DNA Elements) Consortium is an international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI). The goal of ENCODE is to build a comprehensive parts list of functional elements in the human genome, including elements that act at the protein and RNA levels, and regulatory elements that control cells and circumstances in which a gene is active. The ENCODE consortium not only produces high-quality data, but also analyzes the data in an integrative fashion and provides tools to search and visualize them ([Consortium et al., 2007](#); [Consortium et al., 2012](#)).

The gene expression levels in a cell is also controlled using epigenetic mechanisms, defined as heritable alterations that are not encoded in DNA sequence. These modifications include mechanisms such as DNA methylation, histone modification, DNA accessibility and chromatin structure. The NIH Roadmap Epigenomics Mapping Consortium was launched with the goal of producing a public resource of human epigenomic data to catalyze basic biology and disease-oriented research. The Consortium leverages experimental pipelines built around next-generation sequencing technologies to map the epigenetic landscapes in primary ex vivo tissues selected to represent the normal counterparts of tissues and organ systems frequently involved in human disease ([Roadmap Epigenomics et al., 2015](#)).

The basic mechanisms involved in the translation of the molecular codes encoded in the genetic material to the proteins that do the functions are very similar in almost all organisms. Therefore, biologists can use other organisms to study the basic mechanisms to understand gene regulation. Organisms like mouse that are closer to humans in the evolutionary tree are used to model diseases in humans and have proved to be of significant advantage to our understanding of the genetic mechanisms of several diseases. The goal of the modENCODE consortium project is to provide the biological research community with a comprehensive encyclopedia of genomic functional elements in model organisms such as *C. elegans* and *D. melanogaster*, including the domains of gene structure, mRNA and ncRNA expression profiling, transcription factor binding sites, histone modifications and replacement, chromatin structure, DNA replication initiation and timing, and copy number variation ([Boley et al., 2014](#)).

The DNA sequence of any two individuals will be almost identical in the sense that the sequence will be matching in more than 99% of the genome. The 1% variations, however, may greatly affect an individual's disease risk. One of the significant DNA variation commonly studied is single nucleotide polymorphisms (SNPs). Sets of nearby SNPs on the same chromosome are inherited in blocks and is called a haplotype. The HapMap is a map of these haplotype blocks and the specific SNPs that identify

the haplotypes are called tag SNPs. The International HapMap Project is a partnership of scientists and funding agencies from Canada, China, Japan, Nigeria, the United Kingdom and the United States. This will make genome scan approaches to finding regions with genes that affect diseases much more efficient and comprehensive ([International HapMap, 2003](#)).

The high quality curated genomes made available by the GRC and the hapmap data from the consortium coupled with advanced sequencing technologies enable us to study the genomic variants associated with diseases or predisposition to certain diseases, develop or identify appropriate therapeutic strategies. Examples of such initiatives are the Alzheimer's Disease Sequencing Project (ADSP) to identify new genomic variants contributing to increased or reduced risk of developing Late-Onset Alzheimer's Disease (LOAD), the GoT2D consortia to understand the genetic architecture of type 2 diabetes, and The Cancer Genome Atlas (TCGA) – a collaboration between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) – has generated comprehensive, multi-dimensional maps of the key genomic changes in 33 types of cancer. Likewise, the International Cancer Genome consortium (ICGC) project generated a comprehensive description of genomic, transcriptomic and epigenomic changes in 50 different tumor types and/or subtypes which are of clinical and societal importance across the globe ([Devarakonda et al., 2013](#); [Cancer Genome Atlas Research et al., 2013](#); [Zhang et al., 2011](#)).

The Trans-Omics for Precision Medicine (TOPMed), is a program to generate scientific resources to enhance our understanding of fundamental biological processes that underlie heart, lung, blood and sleep disorders. The projects aim to integrate whole-genome sequencing data, metabolic profiles, protein levels and RNA expression data with molecular, behavioral, imaging, environmental, and clinical data to uncover factors that increase or decrease the risk of disease and develop more targeted and personalized treatments ([Auer et al., 2016](#)).

The identification of SNPs and other differences in an individual is complicated because of the difficulty in identifying the disease-causing alterations from the large number of SNPs observed in any individual. The 1000 Genomes Project was initiated to find most genetic variants with frequencies of at least 1% in the populations studied. Though the project started with the aim of profiling 1000 individuals from three populations, the final data set contains data for 2504 individuals from 26 populations. Low coverage and exome sequence data are present for all of these individuals, 24 individuals were also sequenced to high coverage for validation purposes. The variations present in apparently normal individuals helps to remove germline events and help to identify disease causing mutations ([Kuehn, 2008](#)).

Putting together, studying genome and variations within is crucial to understanding the evolution, diversity and diseases. Plethora of tools and techniques have been developed to elicit genomic structure, variations and their functional characterization. Hence, this article hi-lights the informatics of all important aspects of genomic study. However, before diving into genome informatics, we briefly discuss the evolution of genomic technologies in Section “Genomic Technologies in Genomic Studies”.

Genomic Technologies in Genomic Studies

The completion of the human genome project ([Lander et al., 2001](#)) (HGP) in 2004 heralded a new era in biomedical sciences and brought the world of genomics to the mainstream. The HGP costed about 500 million dollars and took about a decade to finish a human genome sequence. Over the next decade, the NIH championed new initiatives to foster technology development at a massive scale to be able to sequence the human genome at a fraction of the cost (\$1000) in a few hours. This provided a massive impetus that gave rise to whole slew of genomic technologies to not only generate the sequence of the genome but to also interrogate it for functional characterization.

Before the HGP, genomics was limited to site based analysis. No single technology existed that could provide a complete view of the genome. Light based microscopy techniques to detect signals from bio-markers that have been added to the genome have been around since the middle of the last century. However, these technologies either provided a very coarse view of the genome, or required the investigator to know precisely the location of the genome to query. In addition, these methods were tedious and expensive requiring hours of a highly skilled technician's time. These were followed by the advent of polymerase chain reaction (PCR) based techniques to interrogate the presence or absence of genomic segments. However once again these technologies were low throughput, time consuming and required prior knowledge of the genome of interest to design appropriate primers. To circumvent this problem, sequencing of the genome was adopted as a viable strategy to get de novo information on genomic regions of interest. The sequencing technologies evolved in 3 generations of advancements.

First generation sequencing technology: These methods were the first of the sequencing based methods – developed by Fred Sanger and colleagues in 1977 ([Sanger et al., 1977](#); [Prober et al., 1987](#)) that relied on incorporation of chain-terminating dideoxynucleotides during the DNA replication process. 'Sanger sequencing' was the mainstay of genomic interrogation technologies for close to 30 years and is even used currently for small projects and for validation experiments. The mean sequencing reads are ~700 bp in length. However, the throughput still remains severely inadequate compared to the more recent next generation sequencing methods described below.

With the successful completion of the HGP, most of which was achieved using Sanger sequencing methods, the genomic community focused on developing new platforms that would allow for cheaper, faster and less error prone DNA interrogation. Over the past decade, rapid research and development of genomic technologies, both microarray based and more critically sequencing based (“Next Generation Sequencing”) technologies have allowed us to get close to that aim, leading to the two generations of advanced hi-throughput sequencing technologies.

Second generation sequencing technologies: Next generation sequencing technologies were inspired by a new development called pyrosequencing ([Nyren et al., 1993](#)), pioneered by Pal Nyren and colleagues, and relied on luminescence produced and measured

during new DNA strand synthesis. Pyrosequencing is a sequencing by synthesis (SBS) based method that uses the DNA polymerase to construct nascent DNA strand using a template strand. Pyrosequencing was licensed by 454 Life Sciences (later acquired by Roche). They produced the first commercially available massively parallel sequencing technology (Ronaghi *et al.*, 1998). The 454 machines produced long contiguous reads in the range of 4–500 bps in millions of wells at the same time. However, the data from the 454 systems suffered from errors arising in nucleotide runs. In long nucleotide runs, the phospho-luminiscent based method had difficulties determining the precise number of nucleotides that were present (Ronaghi *et al.*, 1998).

At around the same time, Solexa and the SOLiD method of sequencing also gained momentum (Voelkerding *et al.*, 2009). Solexa (later acquired by Illumina) relied on the incorporation of dNTPs that are end terminated i.e., they can only incorporate one nucleotide at a time (Bentley *et al.*, 2008). Although this prevented the system from generating errors in the long nucleotide runs, it also limited the system to only generate about 35 basepairs at a time. Sequencing chemistry improvements and other developments allowed for read sizes of up to 250 basepairs to be sequenced (using the MiSeq system). In the most current state-of-the-art, the Illumina HiSeq method can produce about a 100 gigabases of nucleotide sequences in about 2 days for about a thousand dollars in price – there by realizing the goal of bringing low cost sequencing to the disease genomic studies and the clinic (Balasubramanian, 2011; Quail *et al.*, 2012).

Third generation sequencing technologies: There is considerable debate in the genomic community on how to set the various milestones that define a generational change between the continuous developments of sequencing technologies. We choose to define these milestones whenever there is a significant improvement in technology resulting in much better sequencing data.

One example of such a technology is the single molecule real time (SMRT) sequencing platform from Pacific Biosciences or PacBio (Quail *et al.*, 2012). The platform relies on polymerase based sequencing of single molecules of DNA in specially made wells known as zero-mode waveguides (Harris *et al.*, 2008). SMRT sequencing and its newest iterations (Sequel) produce read lengths in the range of 10–20 kb range (Rhoads and Au, 2015). The long reads (although plagued by a slightly higher error rate) provide much more sequence context than the short-read sequencing platforms and have enabled an unprecedented view of the genome (Nakano *et al.*, 2017). It allowed development of pipelines that can deliver de-novo assembled genomes of an individual in a few hours.

More recently nanopore based DNA sequencing technologies have come to the fore. While the idea of using channels (nanopores) to linearize DNA molecules and drive them through is not new, the Oxford Nanopore Minion machines have successfully commercialized the concept (Clarke *et al.*, 2009). The nanopore channel is capable of detecting the change in electric current as different nucleotides of the DNA molecule pass through the channel and thus can read the DNA sequence. The platform allows for potentially generating majority of DNA sequences in the same size range as the SMRT system, however the largest read lengths could top 100–200 kbp (Branton *et al.*, 2008).

There are several other genomic technologies that could be added to the gamut of third generation sequencing technologies, such as the BioNano Genomics' Irys platform and the Fluidigm C1 single cell system. Irys systems produced by Bio Nano Genomics are optical mapping based solution for interrogating large DNA molecules. It relies on DNA nicking enzymes that allow for a large DNA molecule to be tagged at certain known locations across the genome (Schwartz *et al.*, 1993). DNA molecule is then linearized and run through a sensitive system that can identify the nick sites. Identification of the genomic location requires prior knowledge of genome sequence and nicking sites.

The data and technologies greatly enhances our ability to understand the underlying genetics in the development and disease. However, we need to cross the hurdle of plethora of informatics problems. In the remainder of the article, we review the fundamental genome informatics problems and the respective methods based on Illumina sequencing data, long-read sequencing data and microarray data.

Genomic Structural Informatics

Eliciting genomic structure, identifying functional elements and variants that affect their function is critical to eliciting the genomic factors associated with cellular function, cellular state, cellular state transition, diversity, disease and ultimately devising appropriate treatment options. Plethora of informatics methods and procedures have been devised in conjunction with the technologies discussed in Section "Genomic Technologies in Genomic Studies" to elicit genomic structure and variation. This section outlines the informatics procedures to elicit linear as well as 3D organization of genome, identifying functional elements, and genomic variants that lead to diversity and disease.

Eliciting Linear Structure of Genome

A typical eukaryotic chromosome is at least millions of base-pair long while currently available sequencing technology can decode only hundreds to tens of thousands of base-pair in one stretch. Thus, in shotgun (short-read) sequencing approach, chromosomes are broken down in fragments and sequenced. The process of rejoining these fragments to build the linear structure of chromosomes is called assembly and the program that performs the merging of smaller fragments into larger contigs/scaffolds is called *assembler*. The assembly procedures are classified as (1) reference or template based assembly, and (2) *de novo assembly*. The reference based assembly uses genome assembly of a nearest species as starting point for assembly. It involves mapping the reads to the reference genome, resolving conflicts, restructuring reference genome based on the structural patterns observed in the genome to be assembled. On the other hand, *de novo* methods assemble the genome from scratch, discussed in detail below.

de novo assembly starts assembling the genome by connecting short sequence reads based on their similarity which gives rise to longer reads and eventually contigs and scaffolds. Long insert libraries and long reads, which can be generated using Illumina mate-pair and Pacific BioSciences (PacBio) technology (Shi *et al.*, 2016) respectively, have been used due to their capability to span larger repeats and provide better continuity to the assembly. *de novo* assembly algorithms can be divided primarily into 3 types: Greedy, Overlap-layout-consensus (OLC) and de Bruijn graph based algorithms (Miller *et al.*, 2010).

Greedy algorithms, which perform local optimization of an objective function, involve iteratively merging the reads based on sequence similarity until no more merging is feasible. The criterion for merging typically involves length and percentage identity of overlapping regions among the reads (Miller *et al.*, 2010). Greedy algorithms have been used effectively to assemble smaller genomes that are of repeat free. Popular greedy assemblers are Phrap (de la Bastide and McCombie, 2007), TIGR Assembler (Pop and Kosack, 2004), and CAP3 (Huang and Madan, 1999).

The OLC algorithms use a different overlap approach, which works in 3 steps: (1) The Overlap step involves all vs all pairwise overlap comparison among the reads and use it to create the overlap graph where nodes corresponds to reads and edges represents overlap between the reads; (2) The layout step involves analysis of graph to identify paths in which each node is being traversed just once (hamiltonian path); (3) The final step involves the consensus determination where reads overlapping the same base vote for the identity of that base (Miller *et al.*, 2010). Compared to the greedy algorithms, OLC approach can use long distance information and can be used for assembly of larger genomes with repeats. Some of the popular OLC assembler are ARACHNE (Batzoglou *et al.*, 2002), Celera Assembler (Myers *et al.*, 2000), Newbler (Margulies *et al.*, 2005) and Minimus (Sommer *et al.*, 2007).

Most popular category of *de novo* algorithms are *de Bruijn* based (k-mer graph) methods. They work in 2 steps: 1) reads are broken into k-mers (typically $k=25$) and de Bruijn graph is being constructed where each k-mer is a node and edges correspond to overlap of $k-1$ nucleotides between connected nodes. A simple de Bruijn graph of $k=3$ with 5 nodes is shown in the Fig. 1. For typical genomes, de Bruijn graph for carefully chosen k-mer size, most of neighboring nodes will be connected and distal connection (loop formation) will occur in the repeating region of the genomes only. The second step of the de Bruijn graph algorithm is to find an Eulerian path (a path in a graph where each edge is visited once) (Miller *et al.*, 2010). de Bruijn graph algorithms are fast and can be efficiently used for eukaryotic genomes' assemblies. Limitation includes sensitivity to parameter k and vulnerability to sequencing errors. Popular de Bruijn graph assemblers are Velvet (Zerbino and Birney, 2008), SOAPdenovo2 (Luo *et al.*, 2012), ABySS (Simpson *et al.*, 2009) and ALLPATHS-LG (Gnerre *et al.*, 2011).

The quality of *de novo* assembled genomes is tested using various metrics (Haiminen *et al.*, 2011): Contig N50/Scaffold N50, total number of known genes/transcripts found in the genome, the rate of discordant alignments (which quantifies mis-assembly) determined by mapping the paired end data back to the genome, completeness of the assembly detected by using CEGMA (Core Eukaryotic Genes Mapping Approach) (Parra *et al.*, 2007) which uses a set of 248 ultra-conserved CEGs conserved among *A. thaliana*, *C. elegans*, *D. melanogaster*, *H. sapiens*, *S. cerevisiae*, and *S. pombe*. One of the popular software to automatically detect the quality of the assembly is reapr (Recognition of Errors in Assemblies using Paired Reads) (Hunt *et al.*, 2013). It primarily measures the mis-assemblies and break incorrect scaffolds at mis-assembly points. More software packages for evaluation of quality of assembly can be found at "Relevant Websites section".

Putting together, a typical complete whole genome *de novo* assembly pipeline is shown in Fig. 2. It is based on sequencing of Illumina paired end, Illumina mate pair and low-coverage PacBio data. The steps of the assembly include QC of the raw data by NGS QC Toolkit (Patel and Jain, 2012), k-mer correction by corrector module of SOAPdenovo2, k-mer size estimation by KmerGenie (Chikhi and Medvedev, 2014), Assembly using SOAPdenovo2 (Luo *et al.*, 2012), gap closer using GapCloser module of SOAPdenovo2, evaluation & correction of assembly using reapr (Hunt *et al.*, 2013), and finally extending the contigs using low coverage ($\sim 5X$) data from PacBio using PBJelly (English *et al.*, 2012).

Inferring 3D Structure of Genome

DNA is tightly packaged in 3D space of the nucleus. Understanding the 3D genome architecture helps explain events such as promoter-enhancer interaction and disease development (Javierre *et al.*, 2016; Au-Yeung *et al.*, 2016; Martin *et al.*, 2015).

Identifying 3D genome interactions has been a popular approach to infer 3D structure of genome. Imaging technologies, such as microscopy, have enabled the identification of 3D genome interactions for specific predefined regions. On the other hand, chromatin conformation capture methods enabled a flexible platform to identify 3D genome interactions. For instance, 3C technology (Dekker *et al.*, 2002) enabled identification of the interactions between two predefined genomic loci. 4C (Zhao *et al.*, 2006) technology identifies 3D interactions between one locus and all other genomic loci. 5C (Dostie *et al.*, 2006) technology

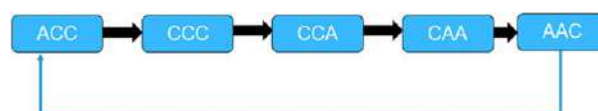


Fig. 1 An example of a de Bruijn graph with $k=3$. It has one repeating k-mer thus instead of six it has only 5 nodes; blue line is showing the formation of loop pointing to the repeating k-mer (<http://www.homolog.us/Tutorials/index.php?p=2.1&s=1>).

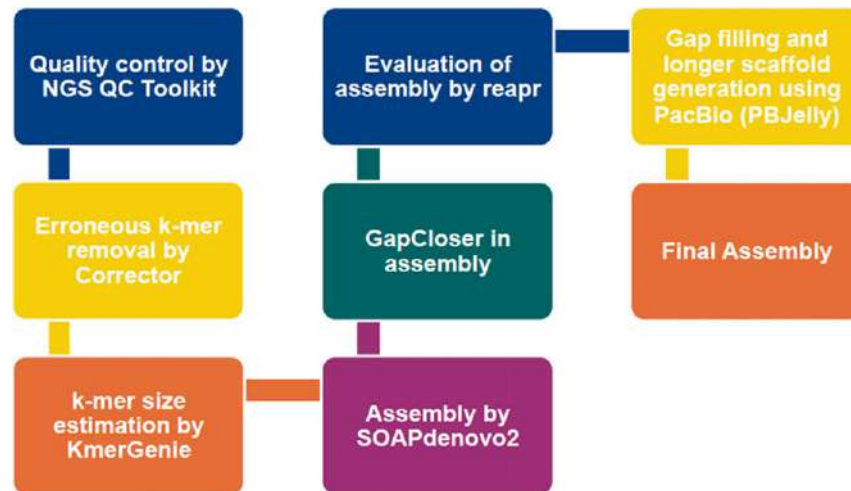


Fig. 2 A schematic of de novo assembly pipeline using Illumina paired end, Illumina mate pair and low coverage PacBio data.

identifies interaction between two sets of predefined loci. Finally, the development of HiC (Lieberman-Aiden *et al.*, 2009) technology enabled the identification of 3D genome interaction at genome scale.

HiC technology generates a massive amount of data which introduced new computational challenges including speed and scalability for data processing, data normalization, and data visualization. To address speed and scalability challenges in processing HiC data, many specialized high-performance tools were developed, e.g., HiCPro (Servant *et al.*, 2015), HiFive (Sauria *et al.*, 2015), HiCBench (Lazaris *et al.*, 2017) and Juicer (Durand *et al.*, 2016a,b). In general, the output of these tools is a contact matrix quantifying the strength of interactions for all pairs of genomic bins. The resolution of the contact matrix increases with increasing sequencing depth.

HiC data normalization is a crucial step before proceeding into downstream data analysis. Sequencing process may introduce several biases innate in the HiC data. HiC data normalization methods fall into two main categories: explicit and implicit normalization. For explicit normalization, HiC data is normalized for each possible covariate (e.g., GC-content bias and sequencing depth bias) independently (Yaffe and Tanay, 2011). HiCNorm (Hu *et al.*, 2012) is an example of explicit HiC data normalization method. For implicit normalization, the main assumption is that any given genomic bin has an equal chance to interact with another bin. Most implicit normalization methods implement normalization using matrix balancing concept. Several implicit approaches have been developed including KR normalization (Rao *et al.*, 2014) and iterative correction and eigenvector decomposition (ICE) (Imakaev *et al.*, 2012) normalization.

As a typical genome browsers are not suitable to visualize 3D genome data, several groups developed specialized tools to visualize 3D genome data. These tools provide convenient ways to visualize 3D genome data including heatmaps, circular plots, and arc tracks. Examples of tools to visualize 3D genome are HiCBrowse (Paulsen *et al.*, 2014a), Juicebox (Durand *et al.*, 2016b), my5C (Lajoie *et al.*, 2009), 3D genome browser (see “Relevant Websites section”), and Epigenome browser (Zhou *et al.*, 2013).

Although HiC data is providing useful information, its coarse resolution limits its ability to map HiC interactions between functional elements (enhancer, promoter, insulators, etc) and the genome. To overcome HiC limitations, many technologies have been developed that focus on interactions that are related to genomic functional elements. These technologies include: chromatin interaction analysis by paired-end tag sequencing (ChIA-PET) (Fullwood *et al.*, 2009), and captureHiC (Martin *et al.*, 2015). ChIA-PET technology reports chromatin interactions that are mediated by specific DNA binding proteins such as CTCF or Pol II. For instance, interactions mediated by Pol II are more likely to contain enhancer-promoter interactions. On the other hand, captureHiC enriches interactions that involve active promoter regions to generate interaction maps between active promoters and the rest of the genome.

The informatics methods developed for HiC data analysis are no longer applicable to process ChIA-PET and captureHiC data. Thus, several methods were developed to process ChIA-PET data: ChIA-PET tool (Li *et al.*, 2010), ChIASig (Paulsen *et al.*, 2014b), and Mango (Phanstiel *et al.*, 2015) are a few examples. CHiCAGO (Cairns *et al.*, 2016) pipeline was developed to process captureHiC data.

Identifying Functional Elements in Genome

Immediate follow-up to genome assembly and 3D architecture prediction is to identify the functional elements such as genes, promoters and distant regulatory elements (e.g., protein binding elements, enhancers and insulators) to make sense of the genome.

Gene prediction approaches are mainly classified into two categories: (a) evidence-based, and (b) *ab initio* based (Picardi and Pesole, 2010). In the Evidence-based (Similarity or Homology-based) algorithms, the sequence of interest is searched against extrinsic homologous (shared evolutionary history) expression-sequence tags (ESTs), mRNA and protein sequences for complete or partial matches using exact algorithms like Smith-Watermann (Smith and Waterman, 1981) or approximate procedures like BLAST (Altschul *et al.*, 1990) and BLAT (Kent, 2002). A high sequence-similarity, for example, with a protein-product suggests that

the target sequence constitutes a protein-coding gene. Latest technologies like RNA-sequencing and Chip-sequencing data can also be exploited as external evidence for accurate gene prediction and validation (Pombo *et al.*, 2017; Hoff *et al.*, 2016; Sikora-Wohlfeld *et al.*, 2013). The extensive dependence on mRNA or protein sequences makes this approach an expensive and limited to known genes. In addition, gene prediction may be hampered by low quality sequences, frameshift mutations or horizontal gene transfers (in prokaryotes). In contrast, *Ab initio* methods rely on the intrinsic sequence content and pre-defined signatures in protein-coding genes i.e., the protein-coding genes are comprised of open reading frames (ORFs) which starts with an initiation codon (ATG) and ends with a triplet stop codon (TAA, TAG or TGA). In a double stranded DNA sequence, there can be six ORFs, three from forward direction and three from reverse (complimentary strand) direction that are always read from 5' to 3' direction. The average gene length in *E. coli*, *S. cerevisiae* and *H. Sapiens* is 317, 483 and 450 codons respectively (Brown, 2002). In general, prokaryotes' genomes contain only 6%–14% of non-coding DNA (Rogozin *et al.*, 2002) and their genes mainly contain continuous ORFs with well characterized promoter sequences, making ORF finding an effective way to identify genes. In contrast to prokaryotes' genomes, eukaryotes' genomes such as humans may contain up to 98% non-coding DNA (Elgar and Vavouri, 2008) and the gene structure is complex due to the presence of introns. They also contain complex promoter or regulatory regions comprised of variable length elements like CpG islands, making ORF finding challenging in eukaryotes. Moreover, eukaryotes may encounter selection of different exons from the same gene, producing multiple gene isoforms, further hampering the gene prediction. However, exons and introns in a gene can be traced using their specific sequence. The upstream sequence (5' end) at an exon-intron boundary is evolutionarily conserved and is mainly 5'-AG↓GTAAGT-3' where arrow indicates the exact breakpoint. The downstream sequence (3' end) at an exon-intron boundary is less evolutionary conserved comparatively and is 5'-PPPPPPNCAG↓-3' where P stands for Pyrimidine nucleotide i.e., Thymidine or Cytosine and N can be any nucleotide.

Gene regulation is combinatorial in nature and is achieved by the interaction of proteins binding to gene regulatory elements on the genome such as promoters, enhancers, insulators and transcription factor binding sites. Among these gene regulatory elements, promoters are the easiest to locate as they are mainly present upstream (5' end) of transcription start sites (TSS) and can be classified into core and proximal promoters. Core promoter region lies 100 base pairs around the TSS and acts as a binding site for general transcription factors (e.g., Pol II). On the other hand, proximal promoter is located few hundred bases upstream of TSS region. Prokaryotes generally contain two short consensus sequences in promoter, TATAAT (also called as Pribnow Box) and TTGACA at 10 and 35 upstream of TSS. On the contrary, eukaryotes promoter regions are more complex and are not well characterized. Many eukaryotic promoters contain a highly conserved TATA box (TATAAA) sequence within first 50 bases of TSS upstream region. Alternatively, some eukaryotic genes may also contain another promoter element called initiator (Kugel and Goodrich, 2017). Unlike TATA box, an initiator is comprised of a highly degenerate sequence 5'-PPA⁺N[T/A]⁺3PPP-3' where P is Pyrimidine (C or T) nucleotide, N can be any nucleotide, A⁺ is the transcription starting point and T/A is T or A nucleotide downstream (+3) of TSS. In contrast, the gene regulatory elements such as enhancers are located far away from the promoter or TSS region, up to few kilo bases or even mega bases in either direction (upstream or downstream) of the gene. The genomic approaches to identify these elements use chromatin immunoprecipitation (ChIP) based techniques like ChIP-Chip or ChIP-seq (Chen *et al.*, 2012; Heintzman *et al.*, 2007). Here, target protein is first cross-linked to cell's chromatin and then protein-DNA complexes are precipitated using protein-specific antibodies. The identity of DNA sequences, present in the protein-DNA complexes, are revealed by mapping them back to the genome and identifying enriched signals by peak finding algorithms such as MACS (Zhang *et al.*, 2008) and QUEST (Valouev *et al.*, 2008).

In eukaryotes, DNA is wrapped around a structure called nucleosome. Nucleosome's length is about 185 bp (Zhou *et al.*, 2011) and it is built from a family of proteins called histones. The combination of DNA and nucleosome is called chromatin. Histone protein tails may go through epigenetic post-transcriptional chemical modifications (Zhou *et al.*, 2011). These modifications include methylation, acetylation, and phosphorylation. Similar to the gene regulatory elements, ChIP-chip and ChIP-seq technologies are used in conjunction with bioinformatics methods (Ashoor *et al.*, 2013; Xu *et al.*, 2010) to identify chromatin modifications.

Chemical modifications of histone tails may correlate with gene expression; certain modifications (e.g., H3K27ac) correlate with increase in gene expression, while others (e.g., H3K27me3) correlate with decrease in gene expression. Many studies (Ernst and Kellis, 2012; Hoffman *et al.*, 2012, 2013; Hon *et al.*, 2008) used combinations of these modifications to demarcate functional regions in human and drosophila genomes. These methods use semi-supervised approaches to identify different patterns of several chromatin modifications and then annotate those patterns based on expert knowledge. Patterns help demarcate the genome into euchromatin and hetero-chromatin regions, identify gene regulatory elements such as enhancers and promoters. Moreover, specific chromatin modifications like H3k27ac, are used to identify super-enhancers in the genome (Pott and Lieb, 2015). Super-enhancers are defined as a set of closely localized enhancers, which are usually located near cell identity genes (Pott and Lieb, 2015). Super-enhancer detection programs work on constructing longer stitches of enhancer marks like H3K27ac. It then scores such longer stitches based on the abundance of mark reads to separate normal enhancers from super-enhancers. Examples of these programs are ROSE (Whyte *et al.*, 2013) and LILY (Boeva *et al.*, 2017) which is specialized to detect super-enhancers for cancer data.

Identifying Genomic Variants

The population diversity and disease are driven by the genomic variations present in the population and disease tissues. These variations include single nucleotide variants (SNVs) and structural variants (SVs). Identifying these variants help understand a variety of population traits as well as disease. Hi-throughput genomic technologies in combination with a variety of informatics

methods made their identification feasible at genome scale. However, a good quality control, alignment of the sequencing reads to the reference genome, and downstream processing are important steps for effective use of these technologies. A typical SNV calling workflows is given in **Fig. 3** and informatics of these steps is described below.

Quality Control: Sequencing technologies are not perfect and reads generated from every technology suffers from technical errors. Sequencing by synthesis method from Illumina have problem of deteriorating quality toward the 3' end of reads due to phasing (incomplete removal of 3' terminators and sequences in the cluster) (Kircher *et al.*, 2011). With increasing read length, more error accumulates which is problematic in signal detection and base calling. Moreover, if fragment length is shorter than the read length, added adaptors during the library-prep step occasionally get sequenced with the reads (Bolger *et al.*, 2014). This is problematic in the alignment the sequenced adaptors will appear as mismatches (due to non-reference bases). Thus removal of poor quality bases/adaptors is essential. The basic diagnostic involves generating the quality report using fastqc (See Relevant Websites Section), performing trimming/filtering using FASTQ/A Trimmer (see "Relevant Websites section") & trimmomatic (Bolger *et al.*, 2014), and adaptor removal using cut-adapt (Martin, 2011) & trimmomatic. High quality reads passing the trimming & filtering criterion will be used for downstream analysis.

Alignment: Alignment of the reads to the genome of interest follows quality control. Read alignment is a "pattern matching" problem. Aligners account for sequencing errors and align millions of reads to the genome at reasonable speed. There are various aligners available and can be broadly classified into two categories (Lindner and Friedel, 2012):

- A) Hash table based aligners use the seed and extend approach. To quickly perform the seeding, these algorithms store reads or reference genome as a hash table (Lindner and Friedel, 2012). They attempt to reduce the sequence search space without discarding the correct alignment candidates. Popular aligners in this category are Mosaik (Lee *et al.*, 2014), BFAST (Homer *et al.*, 2009), Novoalign (see "Relevant Websites section"), PASS (Campagna *et al.*, 2009) and SHRiMP2 (David *et al.*, 2011).
- B) FM-index based aligners work in two steps: first identify the exact matches and then build in-exact alignment from the exact matches. Various representations of suffix/prefix trie namely suffix tree, FM-index and enhanced suffix array are used to find exact matches (Li and Homer, 2010). In contrast to hash table based approach, alignment is needed only once for identical copies of substring in reference genome in FM-index based aligners. Popular alignment algorithms in this category are BWA (Li and Durbin, 2010), BOWTIE 2 (Langmead and Salzberg, 2012) and SOAP2 (Li *et al.*, 2009a).

Post Alignment Processing is an important step to increase accuracy of alignment and variant calling:

- A) Duplicate removal: Reads aligning to the same position could represent true alignments or PCR artifacts. It is recommended to remove the PCR duplicates prior to variant calling as duplicates from PCR could skew variant allele frequency estimates (Bao *et al.*, 2014). Popular methods for removing PCR duplicates include Picard, MarkDuplicates (see "Relevant Websites section") and samtools (rmdup) (Li *et al.*, 2009b). These tools identify duplicates based on their 5' alignment position and orientation with

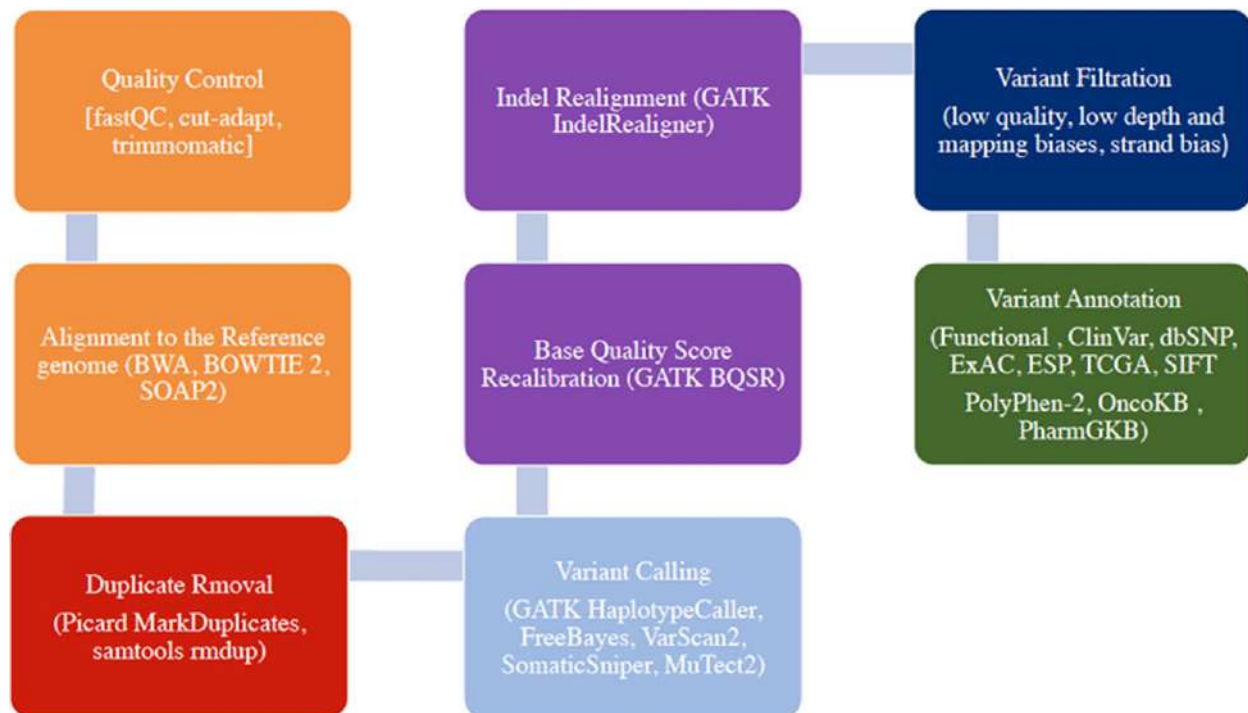


Fig. 3 A schematic depicting typical variant calling workflow.

respect to the genome. The 3' position is not taken into account as trimming of 3' end of reads may result in reads of varying lengths.

- B) Realignment around indels: Various loci in a genome have indels (insertions and deletions) in the sample's genome with respect to reference genome. Alignment of reads with indels is problematic and often leads to inaccurate placement of reads with respect to reference as Indels cause nearby regions shifted. To circumvent this problem, GATK IndelRealigner (McKenna *et al.*, 2010) creates the list of problematic regions and perform the local realignment across these regions to create a consensus indel and thus minimizing the false SNP calls.
- C) Base Quality Score Recalibration (BQSR): Per base quality score emitted by the sequencing machine play important role in variant calling. However, quality values emitted by the sequencer are affected by systematic technical error which leads to over/under assessment of base quality in the data. BQSR step is designed to apply machine learning to model these errors and adjust the quality score for increased accuracy in variant calling. It works in 2 steps: 1) builds a model of covariation based on the data and set of known variation, covariation is analyzed for various features of the base such as machine cycle, quality score and dinucleotide context; and, 2) it adjusts the quality score in the data based on this recalibration model (McKenna *et al.*, 2010).

Variant calling: The primary challenge in variant calling is identifying/separating the real mutation signature from the noise embedded in the sequence data. The current popular algorithm from GATK suite for variant calling is GATK HaplotypeCaller (McKenna *et al.*, 2010) which works in four steps: 1) Active Region search, identifies the regions of genome containing the evidence of significant variation; 2) Identification of haplotypes by performing the assembly of Active Regions; 3) Perform pairwise alignment of reads with haplotype using PairHMM algorithm to determine the haplotype likelihoods based on read data; and, finally, 4) assign genotype to sample based on Bayes' rule for each candidate variant site. Other popular variant callers are FreeBayes (Garrison and Marth, 2012), VarScan2 (Koboldt *et al.*, 2012), SomaticSniper (Larson *et al.*, 2012) and MuTect2 (Cibulskis *et al.*, 2013).

Variant filtration: Errors are introduced during sequencing and bioinformatics steps, which can lead to misleading results. It is therefore necessary to assess various quality metrics to identify real variants. A few common metrics output by most variant callers are coverage and allele frequency. Filters based on these metrics would help get rid of low coverage and low allele frequency variants which are typically false positives. Similarly, variants in low-complexity regions (such as homopolymers and tandem repeats) or in regions with known assembly issues are likely to be errors and would need to be filtered out. Additionally, tool-specific filters would be useful to deduce the authenticity of a variant – for instance, the GATK variant callers (McKenna *et al.*, 2010; De Summa *et al.*, 2017) output information on the mapping quality, genotype quality, evidence of strand bias, position of variant allele along reads – which could be effective in identification and removal of erroneous variant calls.

Structural Variant (SV) calling: In addition to SNVs and microIndels, the structural variants (SVs) such as copy number variations (CNVs), Deletions, Insertions, Duplications, Inversions and Translocations are significant determinants of human genetic diversity (Genomes Project *et al.*, 2010; Mills *et al.*, 2011; Sudmant *et al.*, 2015). SVs can span in size from a few base pairs to chromosome scale, can cause diverse phenotypes and genomic diseases such as cancer. Over the next decade, we expect accurate detection of SVs would become regular practice at the clinic.

Traditional approach to the detection of SVs relied on karyotyping, PCR, FISH and microarray based approaches. However, these legacy approaches permit only a very coarse view of the SVs in the genome (Conrad *et al.*, 2010; Zhao *et al.*, 2013). With the advent of next generation sequencing technologies there has been an unprecedented growth in technological and analytical approaches for the identification of SVs. The massive amounts of sequence data generated from the first, second and third generation sequencing technologies has enabled us to detect SVs and resolve them down to the base pair level.

Currently there are three basic approaches to SV detection: (1) read-depth (RD); (2) split-read (SR) and (3) paired end (PR) (Fig. 4) (Medvedev *et al.*, 2009). RD aligns sequences against a reference genome and counts the depth to estimate the copy number of that particular genomic segment. Using alignment information, SR uses reads that have multiple alignments from different parts of the same read – as these could potentially be overlapping SV breakpoints. PR relies on the paired end nature of the sequencing data from the Illumina sequencing platform to find read-pairs that align abnormally to the reference genome and could have spanned of SV breakpoints.

However, each individual method has its own biases that prevent it from detecting the full spectrum of structural variations in the human genome. Therefore, there is a need for new and improved methods of detecting structural variations. One such method is fusorSV (Published at: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-018-1404-6T>, see "Relevant Websites section") which takes an ensemble approach to the problem. fusorSV is an open source framework that includes a data fusion model built using the 1000 genomes project as the truth set. The model is then used to integrate the output from eight most popular SV calling methods to come up with unified and comprehensive SV call set that maximizes discovery while reducing false positives. Plewczynski *et al.* (2016) provide a comprehensive overview of recent approaches to SV detection using sequence data from whole genome and whole exome sequencing platforms.

Genomic Functional Informatics

Annotating the function of genomics elements and variants is a critical step for meaningful interpretation of data. In this section, we review the *in silico* approaches to functional annotation of genomic elements and variants.

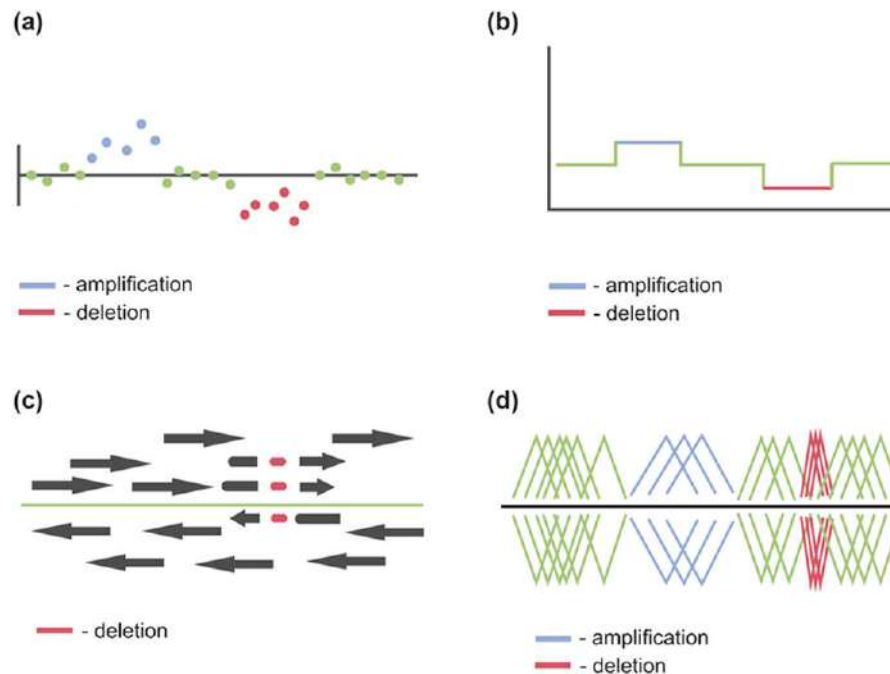


Fig. 4 Typical methods for SV detection (a) aCGH, (b) RD, (c) split read, and (d) paired-end. Adapted from Iskow, R.C., Gokcumen, O., Lee, C., 2012. Exploring the role of copy number variants in human adaptation. *Trends Genet.* 28 (6), 245–257.

Functional Analysis of Variants

The first step towards meaningful interpretation of variants and their functional impact on the genome is variant annotation. SNVs in coding regions can be characterized as transitions if they change a purine base to a purine base or a pyrimidine base to a pyrimidine base; or as transversions if they change a purine base to a pyrimidine base or vice-versa. Further, based on the type of amino-acid change, coding SNVs can be synonymous (no change in amino acid), missense (change in amino acid), or nonsense (introduction of a stop codon). Similarly, coding insertions and deletions (indels) can be classified as inframe if the size of the indel event is a multiple of three bases or frameshift if the size of the indel event is not a multiple of three bases and therefore shifts the reading frame.

The advent of projects such as 1000 Genomes Project (Kuehn, 2008), the Exome Aggregation Consortium (EXAC) (Lek *et al.*, 2016), Exome Sequencing Project (ESP) (See Relevant Websites Section), The Cancer Genome Atlas (TCGA) (Cancer Genome Atlas Research *et al.*, 2013) and others have provided rich catalogs of variants in a number of healthy and disease populations. Population level annotation helps in characterizing a variant as common or rare based on its frequency in different populations, which in turn is interpretation of the variant's impact on disease. In cancer studies, normal population allele frequencies are often used to characterize a variant as germline or somatic depending on whether its population allele frequency is above or below a certain threshold.

Variants can be further functionally characterized based on how conserved they are across different species, with the idea that functionally important variants are likely to be highly conserved. Conservation based scores provided by PhyloP (Pollard *et al.*, 2010), phastCons (Siepel *et al.*, 2005), GERP (Cooper *et al.*, 2005), SiPhy (Garber *et al.*, 2009), and others can be used to assess the evolutionary conservation of variants. Furthermore, prediction based scores such as SIFT (Ng and Henikoff, 2003), PolyPhen (Adzhubei *et al.*, 2013), FATHMM (Shihab *et al.*, 2013), MutationTaster (Schwarz *et al.*, 2010), etc can help evaluate the pathogenic nature of variants based on their the impact on the structure and/or function of the protein.

To understand the impact of variants on drug response and disease, they are annotated with information from clinical knowledge bases such as the JAX Clinical Knowledge Base (JAX CKB) (Patterson *et al.*, 2016), Clinical Interpretation of Variants in Cancer (CIVIC) (Griffith *et al.*, 2017), OncoKB (Chakravarty *et al.*, 2017), Pharmacogenomics Knowledgebase (PharmGKB) (Whirl-Carrillo *et al.*, 2012) amongst others. The clinical annotation of variants associates variants with therapies, diagnostic/prognostic information, and clinical trials, and thus helps assess the clinical relevance of variants.

Functional Analysis of Non-Coding Regulatory Elements

Functional analysis of non-coding regulatory elements is carried out by mapping their target genes and conducting functional analysis of the target genes using experimental or enrichment based *in silico* methods. As promoters were defined using a fixed short window around transcription start sites (TSSs) (−1 kb, +200 bp for example), mapping promoters to gene targets is

straightforward. In addition, recently, functional annotation of mammalian (FANTOM) consortium generated genome-wide and cell specific maps for promoters (Consortium *et al.*, 2014), transcribed enhancers (Andersson *et al.*, 2014) and other functional elements to genes including lncRNAs (Hon *et al.*, 2017) and miRNAs (de Rie *et al.*, 2017).

In the *in silico* approach, protein binding elements/sites and other distant functional elements are mapped to a nearest gene or all genes within a linear genomic window. The mapped genes are used to conduct enrichment analysis of Gene Ontologies, pathways, gene signature such as MSigDB. However, such approach has inherent bias due to non-uniform distribution of genes across the genome and widely varying gene lengths which leads to so called varying lengths of 'assignment domains' (Jair Zhou, 2012). The regulatory elements within an assignment domain of a gene are mapped to the gene. Assignment domain depends on the criterion used for mapping regulatory elements to genes. The bias is minimal if short windows (< 5 kb) are used with reference to TSS of genes. However, the bias was shown to be unacceptably high for other choice of reference (e.g., whole gene) or longer window sizes of the order of tens of kilobases. To circumvent this problem, GREAT (McLean *et al.*, 2010) and reFABS (Jair Zhou, 2012) were proposed for unbiased functional analysis. GREAT contrasts total length of assignment domains of the mapped genes to that of the genes in the functional category. Whereas, reFABS performs resampling based test to correct for the bias.

The target mapping can be more precise using experimental approaches that map interaction of the distant functional elements with promoters (e.g., enhancer-promoter interaction and TFBS-promoter interaction). 3D genome technologies such as HiC and ChIA-PET enable such precise mapping. Given the high resolution and targeted nature of ChIA-PET data, it is more suitable to discover promoter interactions compared to HiC data. Recently, more sophisticated methods were developed to identify enhancer-promoter interaction. For instance, promoter enhancer predictor (PEP) (Yang *et al.*, 2017) uses deep learning from sequence based features only to predict enhancer promoter interactions. Moreover, TargetFinder method (Whalen *et al.*, 2016), integrates information from 3D genome data, in particular HiC data, gene expression, epigenomic data (ChIP-seq and DNase-seq) to predict enhancer targets which can be used for the functional analysis of enhancers as described above.

Functional Analysis of Genes and Their Isoforms

Functional analysis of genes can involve determination of biochemical function, molecular regulation, interaction and expression of genes. Prediction of gene function mostly relies on sequence homology, domain annotation and interaction to functionally known proteins (Clark and Radivojac, 2011; Mostafavi *et al.*, 2008). The homology-based predictions rely on the hypothesis that significant sequence similarity of genes suggests the predicted functional similarity. Homology search for coding region of a gene against the non-redundant protein database, such as NCBI (National Center for Biotechnology Information), SwissProt, allows the initial determination of gene function. Existing methods such as, PSI-BLAST and HMMer (Bork *et al.*, 1998; Bork and Koonin, 1998; Koonin *et al.*, 1998) can determine subtle conservation of protein sequence, which can be used to assign function to an uncharacterized protein. Identifying common evolutionary origin of genes based on homology method can provide further insights into gene function. However, homology approach may have caveats in functional characterization of gene, as fewer amino acid differences can alter the protein conformation and so function, and mutations in genes although may retain the structure, but may confer distinct function (Sjolander, 2004). Gene ontology (Ashburner *et al.*, 2000) and UniProt Gene Ontology Annotation (UniProt-GOA) database (Barrell *et al.*, 2009) are the widely used resources to determine cellular component, molecular function and biological process.

Function prediction of RNA-isoforms: Alternative splicing (AS) is known to be the primary source of transcriptome and proteome diversity in eukaryotes. Functional characterization of specific isoforms and their relevance to cell growth, development and differentiation as well in disease has been reported. However, for most isoforms, the function annotations remain missing. Thus, the extent of functional diversity of alternate mRNA isoforms generated from AS is largely unknown.

Not all mRNA isoforms derived from AS are protein-coding. In addition, AS produces unstable mRNAs which are degraded by nonsense-mediated decay (NMD) machinery (Lykke-Andersen and Jensen, 2015). Transcriptome-wide analysis predicted that ~10% of mRNAs are NMD targets in human (Mendell *et al.*, 2004). Furthermore, some studies suggest that a large number of low abundant isoforms are the consequence of splicing noise, and therefore can be considered as non-functional isoforms (Melamud and Moul, 2009; Pickrell *et al.*, 2010). While, splicing error or noisy splicing is possible, the low abundance of the alternate isoforms can also be due to their high-turnover rate, for instance their recognition by NMD pathway and consequent degradation. Alternate isoforms can have tissue-specific expression and distinct functions (Flouriot *et al.*, 2000; Himeji *et al.*, 2002). Such variations in isoforms are difficult to capture based on sequence-based analysis.

Unlike annotating gene function, the isoform-specific functional annotation is limited by several constraints: (1) AS can lead to skipping of exons that encode protein domains (Liu and Altman, 2003; Resch *et al.*, 2004); (2) AS can cause disorder in amino acid sequence and structure, and thus functional change (Buljan *et al.*, 2012; Romero *et al.*, 2006); (3) AS resulting in a few amino acid changes in the open-reading frame can modulate the protein structure and function (Vogan *et al.*, 1996; Yan *et al.*, 2000); and, (4) mRNA isoforms can have opposite functions, for example, BCLX gene generate two isoforms, where one being pro-apoptotic and the other one being anti-apoptotic (Revil *et al.*, 2007). Most functional annotations are documented at gene levels (Barrell *et al.*, 2009) and genome-wide catalogue for isoform-specific function is still missing. However, methods are available that infer isoform functions based on domains and binding sites (Murvai *et al.*, 2001; Thibert *et al.*, 2005; Vacic *et al.*, 2010; Verspoor *et al.*, 2012). Several computational approaches have been applied to determine isoform-specific function (Eksi *et al.*, 2013; Jia *et al.*, 2010; Li *et al.*, 2014a,b; Tseng *et al.*, 2015). Another method modelled gene-isoform relationship as multiple-instance data

and developed a multiple instance-based label propagation method to predict functions. In addition, assessing isoform orthologues has been found to provide functional information (Jia *et al.*, 2010; Zambelli *et al.*, 2010). Further improvements in isoform function prediction method include integration of heterogeneous data: RNA-Seq, protein-sequence and structure, amino acid composition, protein isoform docking and post-translational modification (Li *et al.*, 2014a).

Functional Analysis of ncRNAs – miRNAs and lncRNAs

Computational functional analyses of microRNAs (miRNAs): Short RNAs are part of the broad family of non-coding RNAs, which perform many regulatory functions. In particular, miRNAs have major interest because of their activity in gene silencing through post-transcriptional repression (Bartel, 2009). Mature miRNAs are about 21–23 nucleotides (nt) in length (Winter *et al.*, 2009), single stranded molecules that bind to 3′ untranslated regions (3′ UTR) of their target mRNAs and regulate many biological processes, including cell differentiation (Bray *et al.*, 2009), organ development (Ambros, 2011). It has also been shown that miRNAs are involved in various diseases, such as cancer (Esquela-Kerscher and Slack, 2006; Bailey *et al.*, 2009) and cardiovascular disorders (Urbich *et al.*, 2008). The mRNA:miRNA interaction depends on a functional motif (2–7 nt) present in the 5′ end of the miRNA which is known as “seed” region (Lewis *et al.*, 2005). In most cases, these short sequences negatively influence the expression of miRNA target genes via translation repression or degradation.

Similar to distant functional elements, the function of miRNAs is also elicited by mapping miRNAs to their target genes, followed by functional analysis of target genes in an *in silico* approach. Numerous online and stand-alone bioinformatics tools have been developed to find miRNA target genes, see Table 1. These algorithms depend on so called *sequence to sequence base pairing*, i.e., complementarity of the base pairing (aka Watson-Crick base pairing) between the seed sequences of the miRNA and the 3′ UTR of the genes, to map miRNA target genes. Further improvement of the method is possible by investigating the nature of the evolutionary conservation of the miRNA and target sequences (Lewis *et al.*, 2005). Some methods predict secondary structure of the duplex to calculate thermodynamic stability of the mRNA:miRNA interaction (Rehmsmeier *et al.*, 2004; John *et al.*, 2004). Next, these predicted targets are used to determine biological processes (gene ontology (The Gene Ontology, 2017), disease ontology (Hillen *et al.*, 1991)) and pathways (Reactome (Fabregat *et al.*, 2016) and KEGG (Kanehisa *et al.*, 2017)) affected by the miRNAs and hence the function of miRNAs.

Computational functional analysis of long non-coding RNAs (lncRNAs): lncRNAs are RNA molecules with a length of more than 200 nucleotides (Perkel, 2013). lncRNAs typically do not encode proteins, certain lncRNAs were shown to have protein coding potential (Guo *et al.*, 2013) though. They are located throughout the genome (Derrien *et al.*, 2012) and exclusively localized in the nucleus (Vance and Ponting, 2014). However, a recent study also reported that it can be found in cytoplasm of the cell (Sun and Kraus, 2015). The lncRNAs are important epigenetic, post transcriptional and transcriptional regulators, and have a variety of functions in developmental, cellular, molecular and disease processes such as cancer (Sun and Kraus, 2015). Several bioinformatics resources are given in Table 2 to access the annotated and predicted lncRNAs with their function, disease association and expression analysis. Along with these resources, multiple tools have been developed to determine the functional properties of the lncRNAs. The tools use homology-based sequence similarity search such as BLASTn (Zhang *et al.*, 2008) and BLAT (Kent, 2002), alignment-based coding potential search (Kong *et al.*, 2007), alignment-free coding region search with CPAT (Wang *et al.*, 2013), motif-based analyses using HOMER (Heinz *et al.*, 2010) and MEME (Bailey *et al.*, 2009). Several R/ Bioconductor packages, for instance, ClusterProfiler (Yu *et al.*, 2012), GDCRNATools (Ruidong *et al.*, 2017) and WGCNA (Langfelder and Horvath, 2008) are also available to explore the gene enrichment, pathways and clustering-based network analyses for functional analysis of lncRNAs.

Population Genome Informatics

Over the last two decades, the availability of entire genomes of multiple species (Mouse Genome Sequencing *et al.*, 2002; International Chicken Genome Sequencing, 2004; International Human Genome Sequencing, 2004; Chimpanzee and Analysis, 2005; Rhesus Macaque Genome *et al.*, 2007) has enabled an in-depth investigation of different species at the molecular level.

Table 1 Online and standalone bioinformatics resources for miRNA target predictions

Tool	Website	Organisms	Features	Reference
TargetScan	www.targetscan.org/	Mammals, worm, fly, fish	Seed match, conservation	Lewis <i>et al.</i> (2005)
DIANA-microT-CDS	www.microna.gr/microT-CDS	Human, mouse, fly, worm	Seed match, conservation, free energy, target site accessibility	Paraskevopoulou <i>et al.</i> (2013)
miRanda	www.microna.org	Human, mouse, rat, fly, worm	Seed match, conservation, free energy	John <i>et al.</i> (2004)
RNAhybrid	www.bibiserv2.cebitec.uni-bielefeld.de/rnahybrid	Mouse, rat, fly, worm	Seed match, conservation, free energy	Rehmsmeier <i>et al.</i> (2004)

Note: Min, H., Yoon, S., 2010. Got target?: Computational methods for microRNA target prediction and their extension. *Exp. Mol. Med.* 42 (4), 233–244.

Table 2 Databases for lncRNAs

Database	Website (Reference)
RNAcentral	http://rnacentral.org (Bateman <i>et al.</i> , 2011)
lncRNAdb	http://www.lncrnadb.org (Quek <i>et al.</i> , 2015)
NONCODE	http://www.noncode.org (Zhao <i>et al.</i> , 2016)
Rfam	http://rfam.org (Kalvari <i>et al.</i> , 2017)
LncRNAWiki	http://lncrna.big.ac.cn/index.php (Ma <i>et al.</i> , 2015)
LNCipedia	https://lncipedia.org (Volders <i>et al.</i> , 2013)
lncRNAdb	http://www.ncrna.org/rnadb (Kin <i>et al.</i> , 2007)
Human Body Map lincRNAs	http://www.broadinstitute.org/genome_bio/human_lincrnas (Khalil <i>et al.</i> , 2009)
LNCediting	http://bioinfo.life.hust.edu.cn/LNCediting (Gong <i>et al.</i> , 2017)
Co-LncRNA	http://www.bio-bigdata.com/Co-LncRNA (Zhao <i>et al.</i> , 2015)
Lnc2Cancer	http://www.bio-bigdata.net/lnc2cancer (Ning <i>et al.</i> , 2016)
LncRNADisease	http://cmbi.bjmu.edu.cn/lncrnadisease (Chen <i>et al.</i> , 2013)

Note: Signal, B., Gloss, B.S., Dinger, M.E., 2016. Computational approaches for functional prediction and characterisation of long noncoding RNAs. *Trends Genet.* 32 (10), 620–637.

Along with this, the advent of sophisticated sequence alignment algorithms (Schwartz *et al.*, 2003; Blanchette *et al.*, 2004) and the field of comparative genomics (Miller *et al.*, 2004) have contributed tremendously to a better understanding of evolution and function of genomes. Comparative genomics involves comparison of genomic DNA among different species to understand various genomic differences and similarities as well as the evolutionary relationship between them. This approach has been used by many scientists to study the pattern and rate variation of mutations across multiple species and it has been clearly demonstrated that mutation rates not only vary between species and organisms but also between different regions of a genome (Chiaromonte *et al.*, 2001; Hardison *et al.*, 2003; Ananda *et al.*, 2011). In fact, regions with high and low neutral substitution rates in the human genome have been shown to contain different functional classes of genes – neutral substitution hot spots are generally biased towards surface receptor and immune response genes while cold spots are biased towards regulatory genes (Chuang and Li, 2004). The knowledge of neutral mutation rate variation can also be valuable to the discovery of non-coding functional elements – specifically, to distinguish between functional conservation and low background neutral mutation rates. Functional prediction algorithms incorporating background rates demonstrate improved separation of functional and neutral sites (Siepel *et al.*, 2005; Taylor *et al.*, 2006; Tyekucheva *et al.*, 2008). Understanding mutation rate variation is also critical for elucidating how biochemical processes (e.g., replication, repair, recombination etc.) and errors in them drive mutagenesis.

The next generation sequencing (NGS) technologies, which have facilitated cost-effective sequencing of entire genomes, are being efficiently employed by projects such as the 1000 Genomes Project (Kuehn, 2008) the Exome Aggregation Consortium (EXAC) (Lek *et al.*, 2016), Exome Sequencing Project (ESP) (See Relevant Websites Section) and others to generate large population-scale resources. These large-scale population genomics projects provide rich comprehensive catalogs of genetic variation of various types (SNVs, indels, structural variants etc.) across human populations of different ancestries (African, European, East Asian, South Asian, South American etc.). They have helped understand the nature and extent of intra- and inter-population differences in genetic variation and have facilitated population-based interpretation of clinical findings, which is important to curtail misdiagnosis (Husten, 2016). Additionally, population genomics also helped identify candidate genes for different diseases based on the distribution of functional variants in them. Scores such as Residual Variance Intolerance Score (RVIS) (Petrovski *et al.*, 2013) and probability of loss of function intolerance (pLI) (Lek *et al.*, 2016) compare the observed genetic variation in a population based on the expected variation to identify genes that are more or less tolerant to functional genetic variation. Thus, they help identify genes under different selective constraints – for instance, genes with the highest tolerance for genetic variation are likely under positive selection, and those with the lowest tolerance under purifying selection. This information is pivotal to identification of candidate genes in a disease population.

Statistical genetics, based on the understanding laid above, makes use of statistical methods to study evolutionary processes that shape genetic variations, and to infer the effect of genetic variants on normal phenotypes and diseases. Besides analyzing the evolutionary constraints and patterns that help understand the genomic function, statistical genetics has proven to be a highly successful approach to identify genetic markers associated with phenotypes.

The early efforts to map disease genes were through linkage analysis in familial data (Laird and Lange, 2011). A marker gene and a disease susceptibility gene are in linkage if they are inherited together during the meiosis phase of sexual reproduction, namely the recombination fraction between them is smaller than one half. Parametric linkage analysis (Abecasis *et al.*, 2002) need to specify appropriate disease inheritance mode, disease allele frequency and penetrance, which may not be always obtainable. Non-parametric linkage analysis (Kruglyak *et al.*, 1996), which is based on identity-by-descent (IBD) sharing (Keith *et al.*, 2008), does not have such requirement, hence is more general. However, linkage analysis is most successful for Mendelian diseases (monogene for a disease), and much less powerful for complex diseases where polygenes dictate a disease. Since family data are hard to collect, association studies in unrelated subjects prevail for complex diseases.

Association analysis is based on linkage disequilibrium (LD) (Pritchard and Przeworski, 2001) among genetic variants. Two alleles of two variants are in linkage disequilibrium if their co-occurrence probability in a haplotype is larger than that when they are independent. r^2 is a LD coefficient for measuring such allelic associations (Pritchard and Przeworski, 2001). Disease variants,

which may not be typed in a study, can be tagged by its LD proxy variants. To achieve the same detection power, the sample size required in an association study which analyzes marker variants is proportion to the reciprocal of r^2 of that when disease causing variant is directly genotyped.

In order to tag most of the common SNPs, the number of typed SNPs in genome-wide association studies (GWAS) has increased from several hundred thousand to five million. Majority of the typed SNPs are common (minor allele frequency > 5%), and non-coding in GWAS. To increase SNP coverage, LD-based imputation methods (Marchini *et al.*, 2007; Marchini and Howie, 2010; Browning and Browning, 2009) for untyped SNPs were developed, which also made the meta-analysis among GWASs on different typing platforms feasible.

Exome array (Huyghe *et al.*, 2013; Liu *et al.*, 2017) for low frequency coding SNPs and exome sequencing (Kiezun *et al.*, 2012; Liu *et al.*, 2014) for rare coding SNPs are alternative ways to investigate roles of coding genetic variants. Two main approaches have been developed for rare variant association analysis which typically has low power. One is burden test (AL *et al.*, 2010) which sums up the counts of alleles of all rare variants in a gene region. Burden test is powerful when the majority of rare variants are functional and have the same effect direction. The other is SKAT-O (L. *et al.*, 2012), which calculates the weighted sum of test statistics for all variants in a gene region. SKAT-O is powerful when a large portion of the variants are neutral and/or have different effect directions.

In GWAS, the most common confounding factor is population structure, where population subgroups are associated with both disease and variant allele frequency, inducing the crude odds ratio different from the stratified odds ratio. The confounding effect of population structure can be adjusted by several ways. For example λ_{GC} correction (Devlin and Roeder, 1999), principal component correction (Price *et al.*, 2006), and linear or generalized linear mixed model (Chen *et al.*, 2016; Zhou and Stephens, 2012) correction.

Discussion

Genome informatics is critical to eliciting genomic structure and its relationship to individual and population phenotypes, including disease. Multiple aspects of genome informatics play important role in a modern genomic project. Though plethora of informatics methods have been developed for each aspect of genome analysis, none of them serve as single go-to tool. Hence, a combination of methods may need to be used for accurate analysis. In addition, while the emerging technologies offer opportunities to better understand the role of genome by generating more precise data, they pose brand new informatics challenges including developing new tools and integrative analysis.

Author Contributions

Karuturi and George conceptualized and led the writing of the article. All authors contributed equally to the article.

See also: Bioinformatics Approaches for Studying Alternative Splicing. Chromatin: A Semi-Structured Polymer. Comparative and Evolutionary Genomics. Comparative Genomics Analysis. Computational Immunogenetics. Detecting and Annotating Rare Variants. Exome Sequencing Data Analysis. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Functional Enrichment Analysis. Functional Genomics. Gene Mapping. Genome Alignment. Genome Analysis – Identification of Genes Involved in Host-Pathogen Protein-Protein Interaction Networks. Genome Annotation. Genome Annotation: Perspective From Bacterial Genomes. Genome Databases and Browsers. Genome-Wide Association Studies. Genome-Wide Haplotype Association Study. Hidden Markov Models. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Learning Chromatin Interaction Using Hi-C Datasets. Linkage Disequilibrium. Metagenomic Analysis and its Applications. Natural Language Processing Approaches in Bioinformatics. Nucleosome Positioning. Phylogenetic Footprinting. Pipeline of High Throughput Sequencing. Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases. Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?. Sequence Analysis. Single Nucleotide Polymorphism Typing. Transcription Factor Databases. Transcriptomic Databases. Whole Genome Sequencing Analysis

References

- Abecasis, G.R., *et al.*, 2002. Merlin – Rapid analysis of dense genetic maps using sparse gene flow trees. *Nat. Genet.* 30, 97–101.
- Adzhubei, I., Jordan, D.M., Sunyaev, S.R., 2013. Predicting functional effect of human missense mutations using PolyPhen-2. *Curr. Protoc. Hum. Genet.* (Chapter 7: p. Unit7 20).
- Altschul, S.F., *et al.*, 1990. Basic local alignment search tool. *J. Mol. Biol.* 215 (3), 403–410.
- Price, A.L., *et al.*, 2010. Pooled association tests for rare variants in exon-resequencing studies. *Am. J. Hum. Genet.* 86, 832–838.
- Ambros, V., 2011. MicroRNAs and developmental timing. *Curr. Opin. Genet. Dev.* 21 (4), 511–517.
- Ananda, G., Chiaromonte, F., Makova, K.D., 2011. A genome-wide view of mutation rate co-variation using multivariate analyses. *Genome Biol.* 12 (3), R27.
- Andersson, R., *et al.*, 2014. An atlas of active enhancers across human cell types and tissues. *Nature* 507 (7493), 455–461.
- Ashburner, M., *et al.*, 2000. The Gene Ontology Consortium, Gene ontology: Tool for the unification of biology. *Nat. Genet.* 25 (1), 25–29.

- Ashoor, H., *et al.*, 2013. HMCAN: A method for detecting chromatin modifications in cancer samples using ChIP-seq data. *Bioinformatics* 29 (23), 2979–2986.
- Au-Yeung, G., *et al.*, 2016. Selective targeting of Cyclin E1 amplified high grade serous ovarian cancer by cyclin-dependent kinase 2 and AKT inhibition. *Clin. Cancer Res.*
- Auer, P.L., *et al.*, 2016. Guidelines for large-scale sequence-based complex trait association studies: Lessons learned from the NHLBI exome sequencing project. *Am. J. Hum. Genet.* 99 (4), 791–801.
- Bailey, T.L., *et al.*, 2009. MEME SUITE: Tools for motif discovery and searching. *Nucleic Acids Res.* 37 (Web Server issue), W202–W208.
- Balasubramanian, S., 2011. Sequencing nucleic acids: From chemistry to medicine. *Chem. Commun. (Camb.)* 47 (26), 7281–7286.
- Bao, R., *et al.*, 2014. Review of current methods, applications, and data management for the bioinformatics analysis of whole exome sequencing. *Cancer Inform.* 13 (Suppl. 2), 67–82.
- Barrell, D., *et al.*, 2009. The GOA database in 2009 – An integrated Gene Ontology Annotation resource. *Nucleic Acids Res.* 37 (Database issue), D396–D403.
- Bartel, D.P., 2009. MicroRNAs: Target recognition and regulatory functions. *Cell* 136 (2), 215–233.
- Bateman, A., *et al.*, 2011. RNACentral: A vision for an international database of RNA sequences. *RNA* 17 (11), 1941–1946.
- Batzoglou, S., *et al.*, 2002. ARACHNE: A whole-genome shotgun assembler. *Genome Res.* 12 (1), 177–189.
- Bentley, D.R., *et al.*, 2008. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* 456 (7218), 53–59.
- Blanchette, M., *et al.*, 2004. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.* 14 (4), 708–715.
- Boeva, V., *et al.*, 2017. Heterogeneity of neuroblastoma cell identity defined by transcriptional circuitries. *Nat. Genet.* 49 (9), 1408–1413.
- Boley, N., *et al.*, 2014. Navigating and mining modENCODE data. *Methods* 68 (1), 38–47.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30 (15), 2114–2120.
- Bork, P., *et al.*, 1998. Predicting function: From genes to genomes and back. *J. Mol. Biol.* 283 (4), 707–725.
- Bork, P., Koonin, E.V., 1998. Predicting functions from protein sequences—where are the bottlenecks? *Nat. Genet.* 18 (4), 313–318.
- Branton, D., *et al.*, 2008. The potential and challenges of nanopore sequencing. *Nat. Biotechnol.* 26 (10), 1146–1153.
- Bray, I., *et al.*, 2009. Widespread dysregulation of MiRNAs by MYCN amplification and chromosomal imbalances in neuroblastoma: Association of miRNA expression with survival. *PLOS ONE* 4 (11), e7850.
- Brown, T.A., 2002. *Genomes*, second ed. Oxford.
- Browning, B.L., Browning, S.R., 2009. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *Am. J. Hum. Genet.* 84, 210–223.
- Buljan, M., *et al.*, 2012. Tissue-specific splicing of disordered segments that embed binding motifs rewires protein interaction networks. *Mol. Cell* 46 (6), 871–883.
- Cairns, J., *et al.*, 2016. HiCAGO: Robust detection of DNA looping interactions in Capture Hi-C data. *Genome Biol.* 17 (1), 127.
- Campagna, D., *et al.*, 2009. PASS: A program to align short sequences. *Bioinformatics* 25 (7), 967–968.
- Cancer Genome Atlas Research, N., *et al.*, 2013. The cancer genome atlas pan-cancer analysis project. *Nat. Genet.* 45 (10), 1113–1120.
- Chakravarty, D., *et al.*, 2017. OncoKB: A precision oncology knowledge base. *JCO Precis. Oncol.* 2017.
- Chen, C.Y., Morris, O., Mitchell, J.A., 2012. Enhancer identification in mouse embryonic stem cells using integrative modeling of chromatin and genomic features. *BMC Genom.* 13, 152.
- Chen, G., *et al.*, 2013. LncRNADisease: A database for long-non-coding RNA-associated diseases. *Nucleic Acids Res.* 41 (Database issue), D983–D986.
- Chen, H., *et al.*, 2016. Control for population structure and relatedness for binary traits in genetic association studies via logistic mixed models. *Am. J. Hum. Genet.* 98 (4), 653–666.
- Chiaromonte, F., *et al.*, 2001. Association between divergence and interspersed repeats in mammalian noncoding genomic DNA. *Proc. Natl. Acad. Sci. USA* 98 (25), 14503–14508.
- Chikhi, R., Medvedev, P., 2014. Informed and automated k-mer size selection for genome assembly. *Bioinformatics* 30 (1), 31–37.
- Chimpanzee, S., Analysis, C., 2005. Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature* 437 (7055), 69–87.
- Chuang, J.H., Li, H., 2004. Functional bias and spatial organization of genes in mutational hot and cold regions in the human genome. *PLOS Biol.* 2 (2), E29.
- Cibulskis, K., *et al.*, 2013. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.* 31 (3), 213–219.
- Clarke, J., *et al.*, 2009. Continuous base identification for single-molecule nanopore DNA sequencing. *Nat. Nanotechnol.* 4 (4), 265–270.
- Clark, W.T., Radivojac, P., 2011. Analysis of protein function and its prediction from amino acid sequence. *Proteins* 79 (7), 2086–2096.
- Conrad, D.F., *et al.*, 2010. Origins and functional impact of copy number variation in the human genome. *Nature* 464 (7289), 704–712.
- Consortium, E.P., *et al.*, 2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 447 (7146), 799–816.
- Consortium, E.P., 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489 (7414), 57–74.
- Consortium, F., *et al.*, 2014. A promoter-level mammalian expression atlas. *Nature* 507 (7493), 462–470.
- Cooper, G.M., *et al.*, 2005. Distribution and intensity of constraint in mammalian genomic sequence. *Genome Res.* 15 (7), 901–913.
- David, M., *et al.*, 2011. SHRIMP2: Sensitive yet practical SHort Read Mapping. *Bioinformatics* 27 (7), 1011–1012.
- Dekker, J., *et al.*, 2002. Capturing chromosome conformation. *Science* 295 (5558), 1306–1311.
- de la Bastide, M., McCombie, W.R., 2007. Assembling genomic DNA sequences with PHRAP. *Curr. Protoc. Bioinform.* (Chapter 11: p. Unit11 4).
- de Rie, D., *et al.*, 2017. An integrated expression atlas of miRNAs and their promoters in human and mouse. *Nat. Biotechnol.* 35 (9), 872–878.
- De Summa, S., *et al.*, 2017. GATK hard filtering: Tunable parameters to improve variant calling for next generation sequencing targeted gene panel data. *BMC Bioinform.* 18 (Suppl. 5), 119.
- Derrien, T., *et al.*, 2012. The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. *Genome Res.* 22 (9), 1775–1789.
- Devarakonda, S., Morgensztern, D., Govindan, R., 2013. Clinical applications of The Cancer Genome Atlas project (TCGA) for squamous cell lung carcinoma. *Oncology (Williston Park)* 27 (9), 899–906.
- Devlin, B., Roeder, K., 1999. Genomic control for association studies. *Biometrics* 55, 997–1004.
- Dostie, J., *et al.*, 2006. Chromosome Conformation Capture Carbon Copy (5C): A massively parallel solution for mapping interactions between genomic elements. *Genome Res.* 16 (10), 1299–1309.
- Durand, N.C., *et al.*, 2016a. Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. *Cell Syst.* 3 (1), 95–98.
- Durand, N.C., *et al.*, 2016b. Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. *Cell Syst.* 3 (1), 99–101.
- Eksi, R., *et al.*, 2013. Systematically differentiating functions for alternatively spliced isoforms through integrating RNA-seq data. *PLOS Comput. Biol.* 9 (11), e1003314.
- Elgar, G., Vavouri, T., 2008. Tuning in to the signals: Noncoding sequence conservation in vertebrate genomes. *Trends Genet.* 24 (7), 344–352.
- English, A.C., *et al.*, 2012. Mind the gap: Upgrading genomes with Pacific Biosciences RS long-read sequencing technology. *PLOS ONE* 7 (11), e47768.
- Ernst, J., Kellis, M., 2012. ChromHMM: Automating chromatin-state discovery and characterization. *Nat. Methods* 9 (3), 215–216.
- Esquela-Kerscher, A., Slack, F.J., 2006. Oncomirs – MicroRNAs with a role in cancer. *Nat. Rev. Cancer* 6 (4), 259–269.
- Fabregat, A., *et al.*, 2016. The Reactome pathway Knowledgebase. *Nucleic Acids Res.* 44 (D1), D481–D487.
- Felix, M.A., 2016. Phenotypic evolution with and beyond genome evolution. *Curr. Top. Dev. Biol.* 119, 291–347.
- Flouriou, G., *et al.*, 2000. Identification of a new isoform of the human estrogen receptor-alpha (hER-alpha) that is encoded by distinct transcripts and that is able to repress hER-alpha activation function 1. *EMBO J.* 19 (17), 4688–4700.
- Fullwood, M.J., *et al.*, 2009. An oestrogen-receptor-alpha-bound human chromatin interactome. *Nature* 462 (7269), 58–64.
- Garber, M., *et al.*, 2009. Identifying novel constrained elements by exploiting biased substitution patterns. *Bioinformatics* 25 (12), i54–i62.
- Garrison, E., Marth, G., 2012. Haplotype-based variant detection from short-read sequencing. *arXiv*. (1207.3907).

- Genomes Project, C., *et al.*, 2010. A map of human genome variation from population-scale sequencing. *Nature* 467 (7319), 1061–1073.
- Gnerre, S., *et al.*, 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. USA* 108 (4), 1513–1518.
- Gong, J., *et al.*, 2017. LNCediting: A database for functional effects of RNA editing in lncRNAs. *Nucleic Acids Res.* 45 (D1), D79–D84.
- Griffith, M., *et al.*, 2017. CIViC is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer. *Nat. Genet.* 49 (2), 170–174.
- Guo, X., *et al.*, 2013. Long non-coding RNAs function annotation: A global prediction method based on bi-colored networks. *Nucleic Acids Res.* 41 (2), e35.
- Haiminen, N., *et al.*, 2011. Evaluation of methods for de novo genome assembly from high-throughput sequencing reads reveals dependencies that affect the quality of the results. *PLOS ONE* 6 (9), e24182.
- Hardison, R.C., *et al.*, 2003. Covariation in frequencies of substitution, deletion, transposition, and recombination during eutherian evolution. *Genome Res.* 13 (1), 13–26.
- Harris, T.D., *et al.*, 2008. Single-molecule DNA sequencing of a viral genome. *Science* 320 (5872), 106–109.
- Heintzman, N.D., *et al.*, 2007. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat. Genet.* 39 (3), 311–318.
- Heinz, S., *et al.*, 2010. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol. Cell* 38 (4), 576–589.
- Hillen, U., Weidenmeier, W., Haug, C., 1991. Ruptured aneurysm of an aberrant subclavian artery. *Dtsch. Med. Wochenschr.* 116 (48), 1832–1836.
- Himeji, D., *et al.*, 2002. Characterization of caspase-8L: A novel isoform of caspase-8 that behaves as an inhibitor of the caspase cascade. *Blood* 99 (11), 4070–4078.
- Hoff, K.J., *et al.*, 2016. BRAKER1: Unsupervised RNA-seq-based genome annotation with GeneMark-ET and AUGUSTUS. *Bioinformatics* 32 (5), 767–769.
- Hoffman, M.M., *et al.*, 2012. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nat. Methods* 9 (5), 473–476.
- Hoffman, M.M., *et al.*, 2013. Integrative annotation of chromatin elements from ENCODE data. *Nucleic Acids Res.* 41 (2), 827–841.
- Homer, N., Merriman, B., Nelson, S.F., 2009. BFAST: An alignment tool for large scale genome resequencing. *PLOS ONE* 4 (11), e7767.
- Hon, C.C., *et al.*, 2017. An atlas of human long non-coding RNAs with accurate 5' ends. *Nature* 543 (7644), 199–204.
- Hon, G., Ren, B., Wang, W., 2008. ChromaSig: A probabilistic approach to finding common chromatin signatures in the human genome. *PLOS Comput. Biol.* 4 (10), e1000201.
- Hu, M., *et al.*, 2012. HiCNorm: Removing biases in Hi-C data via Poisson regression. *Bioinformatics* 28 (23), 3131–3133.
- Huang, X., Madan, A., 1999. CAP3: A DNA sequence assembly program. *Genome Res.* 9 (9), 868–877.
- Hudson, K., 2002. The Human Genome Project: A public good. *Health Matrix Cleveland* 12 (2), 367–375.
- Hunt, M., *et al.*, 2013. REAPR: A universal tool for genome assembly evaluation. *Genome Biol.* 14 (5), R47.
- Husten, L., 2016. Imprecise medicine: Genetic tests lead to misdiagnosis.
- Huyghe, J.R., *et al.*, 2013. Exome array analysis identifies novel loci and low-frequency variants for insulin processing and secretion. *Nat. Genet.* 45 (2), 197–201.
- Imakaev, M., *et al.*, 2012. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat. Methods* 9 (10), 999–1003.
- International Chicken Genome Sequencing, C., C., 2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature* 432 (7018), 695–716.
- International HapMap, C., C., 2003. The International HapMap Project. *Nature* 426 (6968), 789–796.
- International Human Genome Sequencing, C., C., 2004. Finishing the euchromatic sequence of the human genome. *Nature* 431 (7011), 931–945.
- Iskrow, R.C., Gokumen, O., Lee, C., 2012. Exploring the role of copy number variants in human adaptation. *Trends Genet.* 28 (6), 245–257.
- Jair Zhou, H.R.L.A.R.K.M.K., 2012. Bias in genome scale functional analysis of transcription factors using binding site data. *J. Phys. Chem. Biophys.*
- Javierre, B.M., *et al.*, 2016. Lineage-specific genome architecture links enhancers and non-coding disease variants to target gene promoters. *Cell* 167 (5), 1369–1384. e19.
- Jia, Y., *et al.*, 2010. Refining orthologue groups at the transcript level. *BMC Genom.* 11 (Suppl. 4), S11.
- John, B., *et al.*, 2004. Human MicroRNA targets. *PLOS Biol.* 2 (11), e363.
- Kalvari, I., *et al.*, 2017. Rfam 13.0: Shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res.*
- Kanehisa, M., *et al.*, 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* 45 (D1), D353–D361.
- Keith, J.M., *et al.*, 2008. Calculation of IBD probabilities with dense SNP or sequence data. *Genet. Epidemiol.* 32, 513–519.
- Kent, W.J., 2002. BLAT – The BLAST-like alignment tool. *Genome Res.* 12 (4), 656–664.
- Khalil, A.M., *et al.*, 2009. Many human large intergenic noncoding RNAs associate with chromatin-modifying complexes and affect gene expression. *Proc. Natl. Acad. Sci. USA* 106 (28), 11667–11672.
- Kiezun, A., *et al.*, 2012. Exome sequencing and the genetic basis of complex traits. *Nat. Genet.* 44 (6), 623–630.
- Kin, T., *et al.*, 2007. tRNAdb: A platform for mining/annotating functional RNA candidates from non-coding RNA sequences. *Nucleic Acids Res.* 35 (Database issue), D145–D148.
- Kircher, M., Heyn, P., Kelso, J., 2011. Addressing challenges in the production and analysis of illumina sequencing data. *BMC Genom.* 12, 382.
- Koboldt, D.C., *et al.*, 2012. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.* 22 (3), 568–576.
- Kong, L., *et al.*, 2007. CPC: Assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res.* 35 (Web Server issue), W345–W349.
- Koonin, E.V., Tatusov, R.L., Galperin, M.Y., 1998. Beyond complete genomes: From sequence to structure and function. *Curr. Opin. Struct. Biol.* 8 (3), 355–363.
- Kruglyak, L., *et al.*, 1996. Parametric and nonparametric linkage analysis: A unified multipoint approach. *Am. J. Hum. Genet.* 58, 1347–1363.
- Kuehn, B.M., 2008. 1000 Genomes Project promises closer look at variation in human genome. *JAMA* 300 (23), 2715.
- Kugel, J.F., Goodrich, J.A., 2017. Finding the start site: Redefining the human initiator element. *Genes Dev.* 31 (1), 1–2.
- L., S., W., M.C., L., X., 2012. Optimal tests for rare variant effects in sequencing association studies. *Biostatistics* 13, 762–775.
- Laird, N., Lange, C., 2011. *The Fundamentals of Modern Statistical Genetics*. New York: Springer Science.
- Lajoie, B.R., *et al.*, 2009. My5C: Web tools for chromosome conformation capture studies. *Nat. Methods* 6 (10), 690–691.
- Lander, E.S., *et al.*, 2001. Initial sequencing and analysis of the human genome. *Nature* 409 (6822), 860–921.
- Langfelder, P., Horvath, S., 2008. WGCNA: An R package for weighted correlation network analysis. *BMC Bioinform.* 9, 559.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* 9 (4), 357–359.
- Larson, D.E., *et al.*, 2012. SomaticSniper: Identification of somatic point mutations in whole genome sequencing data. *Bioinformatics* 28 (3), 311–317.
- Lazaris, C., *et al.*, 2017. HiC-bench: Comprehensive and reproducible Hi-C data analysis designed for parameter exploration and benchmarking. *BMC Genom.* 18 (1), 22.
- Lee, W.P., *et al.*, 2014. MOSAIK: A hash-based algorithm for accurate next-generation sequencing short-read mapping. *PLOS ONE* 9 (3), e90581.
- Lehner, B., 2007. Modelling genotype-phenotype relationships and human disease with genetic interaction networks. *J. Exp. Biol.* 210 (Pt 9), 1559–1566.
- Lek, M., *et al.*, 2016. Analysis of protein-coding genetic variation in 60,706 humans. *Nature* 536 (7616), 285–291.
- Lewis, B.P., Burge, C.B., Bartel, D.P., 2005. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell* 120 (1), 15–20.
- Li, G., *et al.*, 2010. ChIA-PET tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. *Genome Biol.* 11 (2), R22.
- Li, H., Durbin, R., 2010. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* 26 (5), 589–595.
- Li, H., Homer, N., 2010. A survey of sequence alignment algorithms for next-generation sequencing. *Brief. Bioinform.* 11 (5), 473–483.
- Li, H., *et al.*, 2009a. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25 (16), 2078–2079.
- Li, H.D., *et al.*, 2014a. The emerging era of genomic data integration for analyzing splice isoform function. *Trends Genet.* 30 (8), 340–347.
- Li, R., *et al.*, 2009b. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics* 25 (15), 1966–1967.
- Li, W., *et al.*, 2014b. High-resolution functional annotation of human transcriptome: Predicting isoform functions by a novel multiple instance-based label propagation method. *Nucleic Acids Res.* 42 (6), e39.

- Lieberman-Aiden, E., *et al.*, 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 326 (5950), 289–293.
- Lindner, R., Friedel, C.C., 2012. A comprehensive evaluation of alignment algorithms in the context of RNA-seq. *PLOS ONE* 7 (12), e52403.
- Liu, S., Altman, R.B., 2003. Large scale study of protein domain distribution in the context of alternative splicing. *Nucleic Acids Res.* 31 (16), 4828–4835.
- Liu, H., *et al.*, 2017. Genome-wide analysis of protein-coding variants in leprosy. *J. Investig. Dermatol.* 137, 2544–2551.
- Liu, H., *et al.*, 2014. Genome-wide linkage, exome sequencing and functional analyses identify ABCB6 as the pathogenic gene of dyschromatosis universalis hereditaria. *PLOS ONE* 9, e87250. doi:10.1371/journal.pone.0087250.
- Luo, R., *et al.*, 2012. SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. *Gigascience* 1 (1), 18.
- Lykke-Andersen, S., Jensen, T.H., 2015. Nonsense-mediated mRNA decay: An intricate machinery that shapes transcriptomes. *Nat. Rev. Mol. Cell Biol.* 16 (11), 665–677.
- Ma, L., *et al.*, 2015. LncRNAWiki: Harnessing community knowledge in collaborative curation of human long non-coding RNAs. *Nucleic Acids Res.* 43 (Database issue), D187–D192.
- Marchini, J., *et al.*, 2007. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nat. Genet.* 39, 906–913.
- Marchini, J., Howie, B., 2010. Genotype imputation for genome-wide association studies. *Nat. Rev. Genet.* 11, 499–511.
- Margulies, M., *et al.*, 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437 (7057), 376–380.
- Martin, M., 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* 17 (1), 10–12.
- Martin, P., *et al.*, 2015. Capture Hi-C reveals novel candidate genes and complex long-range interactions with related autoimmune risk loci. *Nat. Commun.* 6, 10069.
- McKenna, A., *et al.*, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* 20 (9), 1297–1303.
- McLean, C.Y., *et al.*, 2010. GREAT improves functional interpretation of cis-regulatory regions. *Nat. Biotechnol.* 28 (5), 495–501.
- Medvedev, P., Stanciu, M., Brudno, M., 2009. Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods* 6 (Suppl. 1), S13–S20.
- Melamud, E., Moul, J., 2009. Stochastic noise in splicing machinery. *Nucleic Acids Res.* 37 (14), 4873–4886.
- Mendell, J.T., *et al.*, 2004. Nonsense surveillance regulates expression of diverse classes of mammalian transcripts and mutes genomic noise. *Nat. Genet.* 36 (10), 1073–1078.
- Miller, J.R., Koren, S., Sutton, G., 2010. Assembly algorithms for next-generation sequencing data. *Genomics* 95 (6), 315–327.
- Miller, W., *et al.*, 2004. Comparative genomics. *Annu. Rev. Genom. Hum. Genet.* 5, 15–56.
- Mills, R.E., *et al.*, 2011. Mapping copy number variation by population-scale genome sequencing. *Nature* 470 (7332), 59–65.
- Mostafaei, S., *et al.*, 2008. GeneMANIA: A real-time multiple association network integration algorithm for predicting gene function. *Genome Biol.* 9 (Suppl. 1), S4.
- Mouse Genome Sequencing, C., *et al.*, 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* 420 (6915), 520–562.
- Murvai, J., *et al.*, 2001. Prediction of protein functional domains from sequences using artificial neural networks. *Genome Res.* 11 (8), 1410–1417.
- Myers, E.W., *et al.*, 2000. A whole-genome assembly of *Drosophila*. *Science* 287 (5461), 2196–2204.
- Nakano, K., *et al.*, 2017. Advantages of genome sequencing by long-read sequencer using SMRT technology in medical area. *Hum. Cell* 30 (3), 149–161.
- Ng, P.C., Henikoff, S., 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Res.* 31 (13), 3812–3814.
- Ning, S., *et al.*, 2016. Lnc2Cancer: A manually curated database of experimentally supported lncRNAs associated with various human cancers. *Nucleic Acids Res.* 44 (D1), D980–D985.
- Nyren, P., Pettersson, B., Uhlen, M., 1993. Solid phase DNA minisequencing by an enzymatic luminometric inorganic pyrophosphate detection assay. *Anal. Biochem.* 208 (1), 171–175.
- Paraskevopoulou, M.D., *et al.*, 2013. DIANA-microT web server v5.0: Service integration into miRNA functional analysis workflows. *Nucleic Acids Res.* 41 (Web Server issue), W169–W173.
- Parra, G., Bradnam, K., Korf, I., 2007. CEGMA: A pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics* 23 (9), 1061–1067.
- Patel, R.K., Jain, M., 2012. NGS QC Toolkit: A toolkit for quality control of next generation sequencing data. *PLOS ONE* 7 (2), e30619.
- Patterson, S.E., *et al.*, 2016. The clinical trial landscape in oncology and connectivity of somatic mutational profiles to targeted therapies. *Hum. Genom.* 10, 4.
- Paulsen, J., *et al.*, 2014a. HiBrowse: Multi-purpose statistical analysis of genome-wide chromatin 3D organization. *Bioinformatics* 30 (11), 1620–1622.
- Paulsen, J., *et al.*, 2014b. A statistical model of ChIA-PET data for accurate detection of chromatin 3D interactions. *Nucleic Acids Res.* 42 (18), e143.
- Perkel, J.M., 2013. Visiting “noncodarnia”. *Biotechniques* 54 (6), 303–304.
- Petrovski, S., *et al.*, 2013. Genic intolerance to functional variation and the interpretation of personal genomes. *PLOS Genet.* 9 (8), e1003709.
- Phanstiel, D.H., *et al.*, 2015. Mango: A bias-correcting ChIA-PET analysis pipeline. *Bioinformatics* 31 (19), 3092–3098.
- Picardi, E., Pesole, G., 2010. Computational methods for ab initio and comparative gene finding. *Methods Mol. Biol.* 609, 269–284.
- Pickrell, J.K., *et al.*, 2010. Noisy splicing drives mRNA isoform diversity in human cells. *PLOS Genet.* 6 (12), e1001236.
- Plewczynski, D., *et al.*, 2016. Chapter 3 – Analysis of Structural Chromosome Variants by Next Generation Sequencing Methods, in *Clinical Applications for Next-Generation Sequencing*. Boston: Academic Press, pp. 39–61.
- Pollard, K.S., *et al.*, 2010. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.* 20 (1), 110–121.
- Pombo, M.A., *et al.*, 2017. Use of RNA-seq data to identify and validate RT-qPCR reference genes for studying the tomato-Pseudomonas pathosystem. *Sci. Rep.* 7, 44905.
- Pope, B.D., *et al.*, 2014. Topologically associating domains are stable units of replication-timing regulation. *Nature* 515 (7527), 402–405.
- Pop, M., Kosack, D., 2004. Using the TIGR assembler in shotgun sequencing projects. *Methods Mol. Biol.* 255, 279–294.
- Pott, S., Lieb, J.D., 2015. What are super-enhancers? *Nat. Genet.* 47 (1), 8–12.
- Price, A.L., *et al.*, 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.* 38, 904–909.
- Pritchard, J.K., Przeworski, M., 2001. Linkage disequilibrium in humans: Models and data. *Am. J. Hum. Genet.* 69, 1–14.
- Prober, J.M., *et al.*, 1987. A system for rapid DNA sequencing with fluorescent chain-terminating dideoxynucleotides. *Science* 238 (4825), 336–341.
- Quail, M.A., *et al.*, 2012. A tale of three next generation sequencing platforms: Comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genom.* 13, 341.
- Quek, X.C., *et al.*, 2015. lncRNAdb v2.0: Expanding the reference database for functional long noncoding RNAs. *Nucleic Acids Res.* 43 (Database issue), D168–D173.
- Rao, S.S., *et al.*, 2014. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 159 (7), 1665–1680.
- Rehmsmeier, M., *et al.*, 2004. Fast and effective prediction of microRNA/target duplexes. *RNA* 10 (10), 1507–1517.
- Resch, A., *et al.*, 2004. Assessing the impact of alternative splicing on domain interactions in the human proteome. *J. Proteome Res.* 3 (1), 76–83.
- Revil, T., *et al.*, 2007. Protein kinase C-dependent control of Bcl-x alternative splicing. *Mol. Cell Biol.* 27 (24), 8431–8441.
- Rhesus Macaque Genome, S., *et al.*, 2007. Evolutionary and biomedical insights from the rhesus macaque genome. *Science* 316 (5822), 222–234.
- Rhoads, A., Au, K.F., 2015. PacBio sequencing and its applications. *Genom. Proteom. Bioinform.* 13 (5), 278–289.
- Roadmap Epigenomics, C., *et al.*, 2015. Integrative analysis of 111 reference human epigenomes. *Nature* 518 (7539), 317–330.
- Rogozin, I.B., *et al.*, 2002. Congruent evolution of different classes of non-coding DNA in prokaryotic genomes. *Nucleic Acids Res.* 30 (19), 4264–4271.
- Romero, P.R., *et al.*, 2006. Alternative splicing in concert with protein intrinsic disorder enables increased functional diversity in multicellular organisms. *Proc. Natl. Acad. Sci. USA* 103 (22), 8390–8395.
- Ronaghi, M., Uhlen, M., Nyren, P., 1998. A sequencing method based on real-time pyrophosphate. *Science* 281 (5375), 363–365.
- Li, R., Qu, H., Wang, S., *et al.*, 2017. GDCRNTools: An R/Bioconductor package for integrative analysis of lncRNA, miRNA, and mRNA data in GDC. *bioRxiv*.
- Sanger, F., Nicklen, S., Coulson, A.R., 1977. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA* 74 (12), 5463–5467.
- Sauria, M.E., *et al.*, 2015. HiFive: A tool suite for easy and efficient HiC and 5C data analysis. *Genome Biol.* 16, 237.
- Schwartz, D.C., *et al.*, 1993. Ordered restriction maps of *Saccharomyces cerevisiae* chromosomes constructed by optical mapping. *Science* 262 (5130), 110–114.
- Schwartz, S., *et al.*, 2003. Human-mouse alignments with BLASTZ. *Genome Res.* 13 (1), 103–107.
- Schwarz, J.M., *et al.*, 2010. MutationTaster evaluates disease-causing potential of sequence alterations. *Nat. Methods* 7 (8), 575–576.
- Servant, N., *et al.*, 2015. HiC-Pro: An optimized and flexible pipeline for Hi-C data processing. *Genome Biol.* 16, 259.
- Shi, L., *et al.*, 2016. Long-read sequencing and de novo assembly of a Chinese genome. *Nat. Commun.* 7, 12065.

- Shihab, H.A., *et al.*, 2013. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Hum. Mutat.* 34 (1), 57–65.
- Siepel, A., *et al.*, 2005. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res.* 15 (8), 1034–1050.
- Sikora-Wohlfeld, W., *et al.*, 2013. Assessing computational methods for transcription factor target gene identification based on ChIP-seq data. *PLOS Comput. Biol.* 9 (11), e1003342.
- Simpson, J.T., *et al.*, 2009. ABySS: A parallel assembler for short read sequence data. *Genome Res.* 19 (6), 1117–1123.
- Sjolander, K., 2004. Phylogenomic inference of protein molecular function: Advances and challenges. *Bioinformatics* 20 (2), 170–179.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147 (1), 195–197.
- Sommer, D.D., *et al.*, 2007. Minimus: A fast, lightweight genome assembler. *BMC Bioinform.* 8, 64.
- Sudmant, P.H., *et al.*, 2015. An integrated map of structural variation in 2,504 human genomes. *Nature* 526 (7571), 75–81.
- Sun, M., Kraus, W.L., 2015. From discovery to function: The expanding roles of long noncoding RNAs in physiology and disease. *Endocr. Rev.* 36 (1), 25–64.
- Taylor, J., *et al.*, 2006. ESPERR: Learning strong and weak signals in genomic sequence alignments to identify functional elements. *Genome Res.* 16 (12), 1596–1604.
- The Gene Ontology, C., C., 2017. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res.* 45 (D1), D331–D338.
- Thibert, B., Bredesen, D.E., del Rio, G., 2005. Improved prediction of critical residues for protein function based on network and phylogenetic analyses. *BMC Bioinform.* 6, 213.
- Tseng, Y.T., *et al.*, 2015. IIDB: A database for isoform-isoform interactions and isoform network modules. *BMC Genom.* 16 (Suppl. 2), S10.
- Tyekucheva, S., *et al.*, 2008. Human-macaque comparisons illuminate variation in neutral substitution rates. *Genome Biol.* 9 (4), R76.
- Urbich, C., Kuehnbacher, A., Dimmeler, S., 2008. Role of microRNAs in vascular diseases, inflammation, and angiogenesis. *Cardiovasc. Res.* 79 (4), 581–588.
- Vacic, V., *et al.*, 2010. Graphlet kernels for prediction of functional residues in protein structures. *J. Comput. Biol.* 17 (1), 55–72.
- Valouev, A., *et al.*, 2008. Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nat. Methods* 5 (9), 829–834.
- Vance, K.W., Ponting, C.P., 2014. Transcriptional regulatory functions of nuclear long noncoding RNAs. *Trends Genet.* 30 (8), 348–355.
- Verspoor, K.M., *et al.*, 2012. Text mining improves prediction of protein functional sites. *PLOS ONE* 7 (2), e32171.
- Voelkerding, K.V., Dames, S.A., Durtschi, J.D., 2009. Next-generation sequencing: From basic research to diagnostics. *Clin. Chem.* 55 (4), 641–658.
- Vogan, K.J., Underhill, D.A., Gros, P., 1996. An alternative splicing event in the Pax-3 paired domain identifies the linker region as a key determinant of paired domain DNA-binding activity. *Mol. Cell Biol.* 16 (12), 6677–6686.
- Volders, P.J., *et al.*, 2013. LNCipedia: A database for annotated human lncRNA transcript sequences and structures. *Nucleic Acids Res.* 41 (Database issue), D246–D251.
- Wang, L., *et al.*, 2013. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. *Nucleic Acids Res.* 41 (6), e74.
- Whalen, S., Truty, R.M., Pollard, K.S., 2016. Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nat. Genet.* 48 (5), 488–496.
- Whirl-Carrillo, M., *et al.*, 2012. Pharmacogenomics knowledge for personalized medicine. *Clin. Pharmacol. Ther.* 92 (4), 414–417.
- Whyte, W.A., *et al.*, 2013. Master transcription factors and mediator establish super-enhancers at key cell identity genes. *Cell* 153 (2), 307–319.
- Winter, J., *et al.*, 2009. Many roads to maturity: MicroRNA biogenesis pathways and their regulation. *Nat. Cell Biol.* 11 (3), 228–234.
- Xu, H., *et al.*, 2010. A signal-noise model for significance analysis of ChIP-seq with negative control. *Bioinformatics* 26 (9), 1199–1204.
- Yaffe, E., Tanay, A., 2011. Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nat. Genet.* 43 (11), 1059–1065.
- Yang, Y., *et al.*, 2017. Exploiting sequence-based features for predicting enhancer-promoter interactions. *Bioinformatics* 33 (14), i252–i260.
- Yan, M., *et al.*, 2000. Two-amino acid molecular switch in an epithelial morphogen that regulates binding to two distinct receptors. *Science* 290 (5491), 523–527.
- Yu, G., *et al.*, 2012. clusterProfiler: An R package for comparing biological themes among gene clusters. *OMICS* 16 (5), 284–287.
- Zambelli, F., *et al.*, 2010. Assessment of orthologous splicing isoforms in human and mouse orthologous genes. *BMC Genom.* 11, 534.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 18 (5), 821–829.
- Zhang, J., *et al.*, 2011. International cancer genome consortium data portal – A one-stop shop for cancer genomics data. *Database (Oxford)* 2011, bar026.
- Zhang, Y., *et al.*, 2008. Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* 9 (9), R137.
- Zhao, Z., *et al.*, 2006. Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions. *Nat. Genet.* 38 (11), 1341–1347.
- Zhao, M., *et al.*, 2013. Computational tools for copy number variation (CNV) detection using next-generation sequencing data: Features and perspectives. *BMC Bioinform.* 14 (Suppl. 11), S1.
- Zhao, Y., *et al.*, 2016. NONCODE 2016: An informative and valuable data source of long non-coding RNAs. *Nucleic Acids Res.* 44 (D1), D203–D208.
- Zhao, Z., *et al.*, 2015. Co-LncRNA: Investigating the lncRNA combinatorial effects in GO annotations and KEGG pathways based on human RNA-Seq data. *Database (Oxford)* 2015.
- Zhou, X., *et al.*, 2013. Exploring long-range genome interactions using the WashU Epigenome Browser. *Nat. Methods* 10 (5), 375–376.
- Zhou, V.W., Goren, A., Bernstein, B.E., 2011. Charting histone modifications and the functional organization of mammalian genomes. *Nat. Rev. Genet.* 12 (1), 7–18.
- Zhou, X., Stephens, M., 2012. Genome-wide efficient mixed-model analysis for association studies. *Nat. Genet.* 44, 821–824.

Relevant Websites

<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
 Babraham Bioinformatics.
<http://evs.gs.washington.edu/EVS/>
 Exome Variant Server.
http://hannonlab.cshl.edu/fastx_toolkit/
 FASTX-Toolkit
 Hannon Lab.
<https://www.jax.org/research-and-faculty/tools/fusorsv>
 FusorSV
 The Jackson Laboratory.
<http://www.3dgenome.org>
 Genome3D.
<http://broadinstitute.github.io/picard/>
 GitHub Pages.
<http://www.novocraft.com>
 Novocraft.
<https://omictools.com/assembly-evaluation-category>
 OMICtools.

Genome Annotation

Josep F Abril, University of Barcelona (UB), Barcelona, Spain and Institute of Biomedicine of the University of Barcelona (IBUB), Barcelona, Spain

Sergi Castellano, Great Ormond Street Institute of Child Health (ICH), University College London (UCL), London, United Kingdom and UCL Genomics, University College London (UCL), London, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

"The genome sequence of an organism is an information resource unlike any that biologists have previously had access to. But the value of the genome is only as good as its annotation. It is the annotation that bridges the gap from the sequence to the biology of the organism." Lincoln Stein, 2001. *Nat. Rev. Genet.* 2(7), 493–503.

Introduction

The precise ordering of nucleotides and amino acids confer specific functional properties to biological sequences. Functional and structural features can then be understood as regions contained within those sequences, defined as segments delimited by an interval of positions, initial and terminal, on the coordinates axis provided by the sequence itself. In a wide sense, sequence annotation is defined by the procedures leading to determine the location and properties of these sequence features. Genomes are the organisms' blueprints, encoding for all the functional elements that lead to a fully developed life form. Therefore, genomes are complex and contain a mixture of features with different roles, which are interspersed with non-functional sequences under no evolutionary constraint. Genome annotation is the process of identifying any functional element along the DNA sequence of a genome, yet at initial stages often focuses on genes. The annotation gives meaning to the genome by providing the location and function of genes (protein coding or otherwise) and regulatory regions, which underlie the biology of living organisms. Completing the catalog of functional regions of the genome can facilitate further downstream analyses, as for instance the assessment of the effect of mutations caused by single nucleotide variants in individuals or populations. As it has been remarked since the start of the genomics era, raw genomes are thus of limited use to scientists and their annotation has become central to genome research.

Genome annotation itself has evolved over the years and it is possible to distinguish three, possibly four, major stages based on the techniques and data used. Each succeeding, yet overlapping, stage has refined previous annotations of the genome, characterized some of its previously unannotated parts, and provided novel insights into genome biology. The development in the mid 1990s of computational techniques to predict protein-coding genes, which span thousands of nucleotides in eukaryotes, marks the start of the efforts to provide genome-wide annotations. This first stage centered on the annotation of individual reference genomes (one per species), using a combination of the statistical properties of the sequences of the protein-coding genes themselves (to predict genes *ab initio*) and the similarity of mRNA and protein sequences to their genome of origin (to align known transcripts and proteins). Both approaches though rely on the existence of a set of already characterized genes, *a priori* knowledge based on experimental evidences, which will be used either as training set to build the models describing the genes species-specific signals and sequence content biases for subsequent computational predictions, or to be compared against other anonymous sequences to map their location based on similarity, thus annotating the genes in them (Fig. 1).

When the reference genomes of more species became available in the early and mid 2000s, they ushered in a comparative and second stage of genome annotation, where mRNA, protein and genome sequences from one species could be homologously aligned to annotate the genome of another. A third and regulatory stage of genome annotation was initiated in the mid and late 2000s, after genome-wide methods to profile DNA binding, conformation and alteration (*e.g.*, in their copy number) were introduced. Such genome-wide and multi-omics regulatory annotations have become standard in the last few years and short (in the tens or hundreds of nucleotides) regulatory elements are commonly annotated. Finally, genome annotation definitely moved away from one or few genomes per species in the late 2000s and early 2010s. This fourth and population stage of genome annotation, which uses hundreds if not thousands of genomes from the same species to annotate individual nucleotides, is where we find ourselves in the late 2010s. We discuss in some detail these different stages of genome annotation and how their increased resolution and inclusiveness of multi-omics approaches leads to more precise genome biology.

The *ab Initio* and Similarity Years

Ab initio gene prediction (also known as *de novo*) refers to the identification of protein-coding genes using the signals that define and characterize their gene structures along the genome. This is done without the aid of their encoded mRNAs and proteins. When the sequences of these mRNAs and proteins are sequenced they can be aligned to the genome to locate the genes that encode them. Thus, allowing the prediction of protein-coding genes by similarity. We discuss both approaches and the increased accuracy of similarity approaches over the *ab initio* ones.

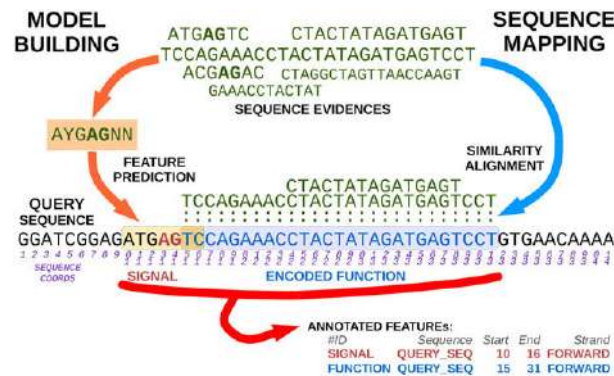


Fig. 1 In order to annotate an anonymous sequence (query) we have two main approaches. The first approach, depicted on the left, it is to build models that, based on known sequences, summarize some property or bias characteristic of the functional elements being annotated, and then apply an algorithm to score these elements along the sequence, taking the optimal prediction as the one to annotate. The second approach uses the alignment against the query of the set of known sequences, retrieved either from the same (similarity search) or other species (homology search). Again, the optimal alignment is taken as the one to annotate the functional element. The coordinates – relative to the original sequence – define the location for each of the functional elements found in the anonymous genome, also known as the annotation set.

Ab Initio Gene Prediction

The structure of prokaryotic and eukaryotic genes is such that start (AUG) and stop (UAA/UAG/UGA) codons signal the initiation and termination steps of protein translation. These codons, together with some nucleotide context around them, define the coding DNA sequence (CDS) of the exon or exons in a gene. In eukaryote genomes, in particular, codons in exons along the CDS are not necessarily adjacent but often interrupted by introns of hundreds or thousands of nucleotides in length (Sharp, 2005). These breaks are defined, in most cases, by the canonical donor [AG|GTRAGT, where R=A or G] and acceptor [YTTYYYYYYNCAG|G, where Y=C or T and N=A, C, T or G] splice sites in mammalian genomes (Burset *et al.*, 2000; Abril *et al.*, 2005). Albeit GT-AG and the surrounding context defines a highly conserved motif, splicing is a complex biological process that tolerates small nuclear RNA variants in the spliceosome (Kyriakopoulou *et al.*, 2006), the RNA-protein complex that mediates the splicing process, and even non-canonical splice sites, such as those defined by the donor and acceptor pair AT-AC (Burset *et al.*, 2000; Patel and Steitz, 2003). Together with transcription initiation and termination sites, these signals have allowed the computational definition of prokaryotic and eukaryotic genes.

In addition, the genetic code is degenerate and the same amino acid can be encoded by different codons. These codons are synonymous and the frequency of their occurrence is known as codon usage or bias. Codon usage varies between species and between genes in a genome due to mutational biases and differences in the strength of natural selection on translational optimization (Plotkin and Kudla, 2011). Predating the sequencing of genomes there were tools that used codon bias to annotate expressed sequence tags (ESTs), fragments of mRNAs, which were even capable to correct frameshifts from sequencing errors like ESTSCAN (Iseli *et al.*, 1999). Thus, codon usage biases within the signals that define prokaryotic and eukaryotic genes can be used to computationally identify the CDS of genes along a particular species genome. Some of the first *ab initio* programs to predict genes this way did so in the short genomes of prokaryotes. For example, the seminal GENEMARK (Lukashin and Borodovsky, 1998) and GLIMMER (Delcher *et al.*, 1999) programs made use of Markov chains – a stochastic model describing a sequence of nucleotide-emitting states in which the probability of the next state only depends on the previous one – to capture the usage and dependence between successive nucleotides and codon positions, while defining the start and end of genes with models of the nucleotides around these signals (their sequence context). Interestingly, Markov chains of order-five, in which the probability of the next nucleotide depends on the five previous ones, work best in gene prediction as they also capture dependencies between consecutive amino acids in proteins (Durbin *et al.*, 1998). These models are used to score potential genes along the genome. The final prediction is then obtained using algorithms that find the optimal gene structure, that is, the combination of predicted exons that maximizes the score of the predicted genes. GENEMARK, for example, uses the Viterbi algorithm (Durbin *et al.*, 1998). This is a dynamic programming algorithm (Eddy, 2004a) – an optimization technique based on breaking down a problem into smaller ones and using them recursively to find the optimal solution – which, in a Hidden Markov Model (HMM) (Eddy, 2004b) – a statistical model following a Markov Chain process with hidden states that either emit nucleotides for coding or noncoding sequences – finds the best scoring gene structures. More recent developments include Prodigal (Hyatt *et al.*, 2010), which extends dynamic programming with a special set of rules to cope with overlapping genes while looking for the optimal gene structure prediction along the genome.

Hyatt *et al.* also established a reference annotation set to evaluate the accuracy of available prokaryotic gene-finders; a difficult task due to the lack of an experimentally verified translation start in many genes that changes the average length of genes, as well as the huge variability in GC content across prokaryota. The accuracy of gene prediction in prokaryotic genomes is generally high, with more than 95% of true protein coding genes being detected in most species (Hyatt *et al.*, 2010). Though the accuracy predicting the start of the gene is lower, usually around 80%. As a result, the 5' end (upstream) of prokaryotic genes is less likely to

be correct. In addition, prokaryotic gene prediction may still slightly overpredict the number of genes in a genome, with the resulting false positive genes being often short. It is however possible that some of these genes are indeed real. In any case, prokaryotic gene prediction has become remarkably accurate in the last two decades.

The identification of eukaryotic genes is, compared to their prokaryotic counterparts, more demanding due to their length (often in the hundreds of thousands of bases) and their split structure. Still, similar approaches to those in prokaryotes can be used. Early eukaryotic gene predictors included GENEID (Guigo *et al.*, 1992) and GENSCAN (Burge and Karlin, 1997). Both of these programs use Markov Chains of order five to assess codon usage and score coding exons as a combination of their coding bias and their start, end and splice sites signals; either in a rule-based (the GENEID Gene Model; (Guigo *et al.*, 1992)) or a generalized HMM (GHMM) framework for GENSCAN (Burge and Karlin, 1997). To cope with the combinatorial explosion of putative exons and predict the optimal gene structure along the genome dynamic programming was used (Burge and Karlin, 1997; Guigo, 1998).

The accuracy of such predictions is lower than in prokaryotic genomes (Table 1). An influential work on the assessment of gene prediction accuracy in eukaryotes (Berset and Guigo, 1996) established the ground metrics at the nucleotide, exon and gene levels, and provided one of the first reference gene test sets on this field. To begin with, there is in gene prediction a compromise between sensitivity (S_n in Table 1), the proportion of true coding nucleotides correctly predicted as such, and specificity (S_p in Table 1), the proportion of predicted coding nucleotides actually coding, so increasing one often decreases the other. At the nucleotide level this is not much of an issue, as fragments of most genes are predicted; yet missing the right start and end of an exon reduces the sensitivity and specificity of gene prediction at the exon level (Table 1). The accuracy for the overall gene is even lower due the difficulty in determining the right combination of exons along the genome. On the other hand, short genes encoding small proteins (Su *et al.*, 2013) are usually missed (missing exons, ME, in Table 1) as most genome annotation protocols apply a minimum length threshold, *i.e.*, of 100 amino acids, to reduce the number of falsely predicted genes (wrong exons, WE, in Table 1). These measures show that classic *ab initio* gene prediction programs identified no more than 80% of the true coding nucleotides while predicting as coding a large fraction of noncoding ones (Table 1). Also, a substantial fraction of coding exons had their boundaries mispredicted. Furthermore, around 40% of the predicted exons are completely wrong, a rather unreasonable figure. Still, *ab initio* gene prediction provided at the time a first set of annotations that could be later refined using comparative and/or experimental approaches (see below). To cope with annotation changes and multiple transcripts in a gene further accuracy measures have been proposed such as the Annotation Edit Distance, which quantifies changes to a gene annotation, and the Splice Complexity, which quantifies the complexity of alternative splicing in a gene (Eilbeck *et al.*, 2009). The terms recall and precision were later adopted to measure gene prediction accuracy. Recall is equivalent to the sensitivity measure described above, whereas precision is equivalent to specificity. Importantly, specificity or precision in the gene prediction field avoid the estimation of true negatives from incomplete genomes. Nowadays gene predictions are also compared with respect to the area under the curve in a receiver operating characteristics plot (known as AUC-ROC plot) (Powers, 2011).

The first large-scale assessment on gene-finding accuracy was performed over 2.4 megabases of the *Drosophila melanogaster* genome, which contained the *alcohol dehydrogenase* gene (*Adh*), in a collaborative experiment known as the Genome Annotation Assessment Project (GASP) (Reese *et al.*, 2000). This region was first annotated by human curators, with the raw sequence and a subset of the reference fly annotations later provided to the participants performing computational gene predictions. Similar assessments have subsequently been done in the worm (nGASP) (Coghlan *et al.*, 2008) and human genomes (EGASP and RGASP, see Figs. 2 and 3) (ENCODE Consortium, 2012; Guigo *et al.*, 2006).

Human curators have been invaluable in producing reference annotations for the above assessment experiments and in integrating the experimental evidence that validates (or refutes) individual gene predictions. For example, *D. melanogaster* was the first eukaryotic genome assembled from shotgun sequences (where DNA is broken up randomly) (Adams *et al.*, 2000) and Celera, the company sequencing it, organized an annotation jamboree with bioinformatics experts, protein specialists, and fruit fly biologists to functionally annotate it (Pennisi, 2000). Similar community annotation projects have since then become a standard for other genomes, among those it is worth citing the VERtebrate Genome Annotation database (VEGA, <http://vega.sanger.ac.uk>) maintained by the human and vertebrate analysis and annotation (HAVANA) group at the Wellcome Trust Sanger Institute (Harrow *et al.*, 2012), which also has defined guidelines to integrate gene annotations (Madupu *et al.*, 2010; Loveland *et al.*, 2012). To assist curators in the visualization and editing of gene annotations different tools have been created. First, to produce static maps of annotations along a genome sequence with GFF2PS (Abril and Guigo, 2000), GFF2APLOT (Abril *et al.*, 2003), and more recently CIRCOS (Krzywinski *et al.*, 2009). Later, tools to manually review gene predictions and integrate experimental evidence, such as APOLLO (Lewis *et al.*, 2002), the Integrative Genomics

Table 1 Accuracy of gene prediction programs on human chromosome 22

Program	Nucleotide			Exon				
	S_n	S_p	CC	S_n	S_p	$\frac{S_n+S_p}{2}$	ME	WE
<i>“ab initio” gene finding</i>								
GENEID	0.73	0.67	0.70	0.65	0.55	0.60	0.21	0.33
GENSCAN	0.79	0.53	0.64	0.68	0.41	0.55	0.15	0.48
<i>Comparative genomics approach</i>								
SGP2	0.75	0.73	0.73	0.66	0.58	0.62	0.19	0.28
TWINSKAN	0.72	0.67	0.69	0.69	0.59	0.64	0.18	0.29

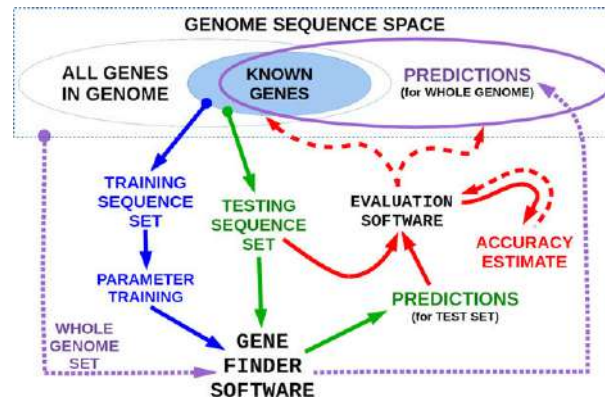


Fig. 2 The assessment of the accuracy of gene predictions (or other functional features) is usually done in three steps. First, a set of well characterized genes is split into the training and the testing sequence sets. Second, the training set is used to estimate the species-specific parameters, such as the frequencies of the different codons in coding exons. Finally, the test set is used to predict genes with the computational approach under assessment. The sequence coordinates of the predicted genes are then compared against those of the known genes. Accuracy metrics, calculated at the nucleotide, exon and gene levels, are used to assess accuracy of the method and the reliability of the predictions. This assessment can be used later on either to estimate overall genome predictions reliability or to iterate through the train/test sets again in order to improve the software parameters (dashed lines).

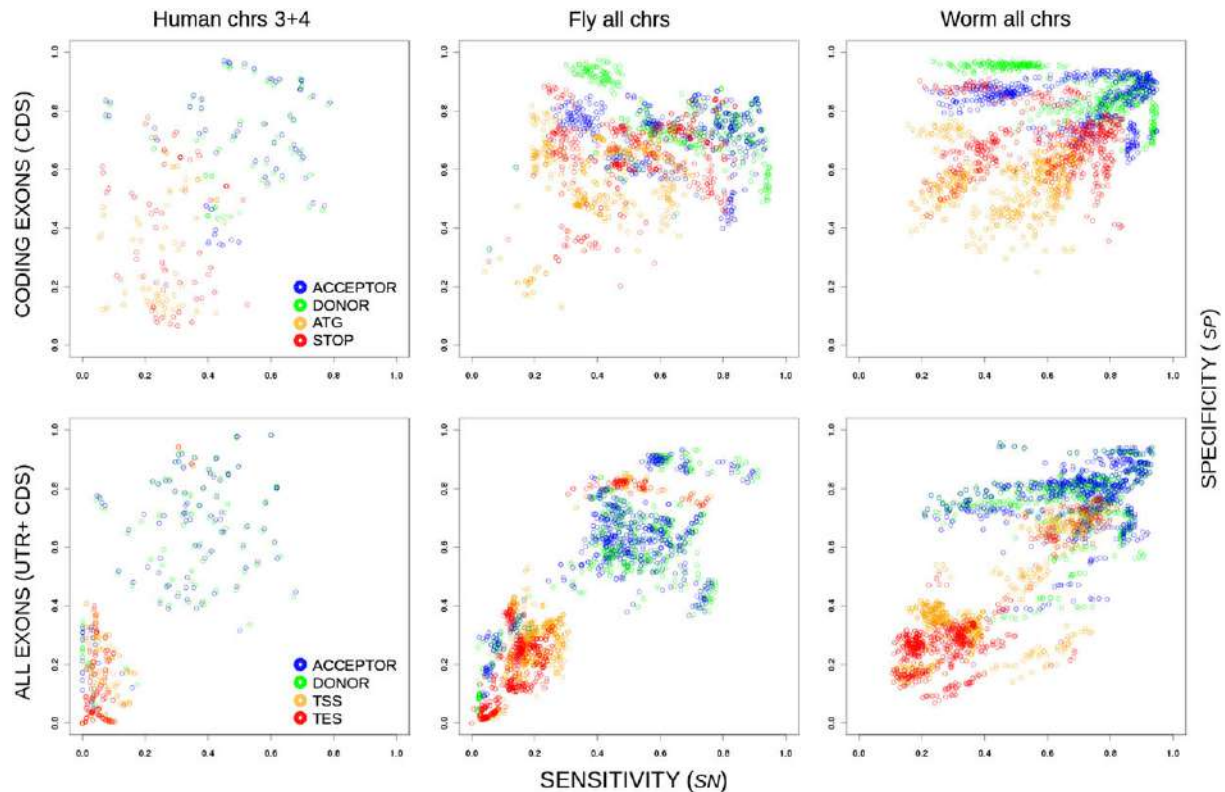


Fig. 3 Sensitivity versus specificity scatterplots for the prediction of a number of defining signals in coding exons (CDS), on top row, and full-length exons, on bottom row, for the submitted predictions to RGASP. The accuracy metrics were computed over the genome sequences from three species: the human chromosomes three and four, the *Drosophila melanogaster* (fly) genome and the *Caenorhabditis elegans* (worm) genome. Each dot corresponds to a set of predictions made by one of each RGASP participant over one of the species sequences. Note the lower accuracy of predictions in the larger human genome as well as the difficulty to predict the untranslated part of exons due to missing the correct transcription start (TSS) and end (TES) sites in the three species.

Viewer (IGV) (Thorvaldsdottir et al., 2013) and the GALAXY server (Giardine et al., 2005) grew in importance. Over time, genome browsers providing access to databases storing the analyses from genome annotation pipelines have become the standard. These include GBrowse (Stein et al., 2002), the UCSC Genome Browser (Kent et al., 2002) and ENSEMBL (Hubbard et al., 2002; Aken et al., 2017), which are widely used today (Fig. 4).



Fig. 4 UCSC Genome Browser view of the human chromosome 17, focused on the sequence annotation around the *ACADVL* gene. Topmost annotation tracks show several gene/transcript structures obtained by different methods; mid tracks summarize the conservation score and the human sequence aligned to different vertebrate genomes; bottom tracks include sequence variants (OMIM and common variants), histone marks from the ENCODE project and transcript gene expression. The gene annotation tracks on top displays predictions from GENCODE, GENSCAN, SGP2, and AUGUSTUS, which can be compared with the current curated annotations from two reference sets, GENCODE and NCBI-RefSeq. From the aforementioned gene-finding programs, only AUGUSTUS predicts the three alternative spliced transcripts as it integrates experimental evidence, although it fails to predict some conserved exons detected by the other programs. Note that downstream *DVL2* gene was also detected by the gene-finders, but the upstream *DLG4* gene was missed.

Using mRNAs and Proteins

The sequencing of fragments (ESTs) or full-length mRNAs and proteins from the species whose genome is being annotated has been the main approach to increase the accuracy of *ab initio* gene annotation. Instead of relying on the statistical properties of the CDS and the signals that define them, the direct alignment of transcribed or translated sequences from a genome provides

experimental evidence to a gene structure. The alignment of such sequences to a naked genome has been computationally addressed yet again as a dynamic programming algorithm that takes into account the splice sites and exon/intron structure of eukaryotic genes. In particular, ESTs or full-length transcripts can be properly aligned to the genome using variations of the Smith-Waterman (Smith and Waterman, 1981; Gotoh, 1982) and Needleman-Wunsch (Needleman and Wunsch, 1970) algorithms, the classic local and global sequence alignment algorithms, respectively. This is the approach used by the pioneering EST_GENOME program (Mott, 1997). For the sake of speed, other approaches use the heuristic BLAST algorithm (Altschul *et al.*, 1990) to produce the starting alignments of coding exons in the genome, which are later recursively chained together to obtain the best gene structure. SIM4 (Florea *et al.*, 1998) and SPIDEY (Wheeler *et al.*, 2001) were examples of the latter. A more recent tool to align transcripts to a genomic sequence is EXONERATE (Slater and Birney, 2005), which provides (slower) exhaustive or (faster) heuristic approaches to align transcripts using different dynamic programming algorithms. On the other hand, spliced-alignment algorithms with proteins require the conceptual translation of the genomic sequence and finding the optimal alignment of a multi-exon structure to a related protein. PROCUSTES (Gelfand *et al.*, 1996) and GENEWISE (Birney and Durbin, 1997, 2000) are well-known examples, with the HMM-based GENEWISE at the core of the ENSEMBL gene annotation system (Curwen *et al.*, 2004; Aken *et al.*, 2016). The alignment of proteins to predict genes in large scale genomes can become extremely time- and space-demanding. This investment, however, is usually at the benefit of prediction accuracy. In this regard, the faster EXONERATE can also align proteins to DNA sequences and is also heavily used in the ENSEMBL pipeline.

In the last few years, the widespread use of next-generation sequencing technologies for genomes and transcriptomes (RNA sequencing, RNA-seq) and high-throughput mass spectrometry approaches for proteins have put the approaches above at the forefront of genome annotation. For example, the alignment of transcripts from RNA-seq experiments has allowed the comprehensive annotation of alternative splice forms of protein-coding genes. Humans, for example, produce on average four different protein-coding sequences per gene (ENCODE Consortium, 2012). Pseudogenes, which derived from functional genes through retrotransposition or duplication but have lost the original functions of their parental genes, are sometimes (about 10%) also transcribed (ENCODE Consortium, 2012). The high degree of conservation and similarity to functional genes of recent pseudogenes are significant enough to confound conventional gene prediction approaches (Zheng *et al.*, 2007). Other non-standard types of genes that have benefited from experimental evidence are fused genes, which combine part or all of the exons from transcripts of two collinear genes. In this way, they may contribute to the increase the protein repertoire in eukaryotes (Parra *et al.*, 2006). Similarly, trans-splicing, where the starting exons from one transcript are merged with exons from another transcript, far downstream or even located in another chromosome sequence, can be annotated using sequenced transcripts. Remarkably, up to 70% of the worm *Caenorhabditis elegans* mRNAs begin with the spliced leader sequence (SL, 22bp), which is not associated with the gene (Hastings, 2005).

Annotation of Non-Standard Protein Coding Genes

The identification of eukaryotic selenoprotein genes has challenged most computational gene prediction methods. The reason for this is the alternative use of the UGA codon, typically a termination signal, to code for selenocysteine (Chambers *et al.*, 1986; Zinoni *et al.*, 1986). The recoding of UGA confounds the computational identification of selenoprotein genes, whose exons containing a selenocysteine codon are truncated or skipped in their prediction. Selenocysteine, the 21st amino acid in the genetic code, is incorporated into selenoprotein with the help of an mRNA element, the selenocysteine insertion sequence (SECIS), located in the 3' untranslated region of selenoprotein genes (48,49). This RNA structure directs the incorporation of selenocysteine through the interaction with several trans-acting factors. Selenocysteine mediates the biological functions of selenium, an essential micronutrient whose deficiency is associated to infertility, preterm birth and serious children's disorders of the bone and heart (Rayman, 2000).

Efforts to identify selenoproteins were first directed to the computational prediction of the SECIS element in Expressed Sequence Tags (ESTs, sequenced mRNA fragments). Known SECIS elements were used to derive deterministic models of their secondary structures (Fig. 5) which, in turn, were used to scan and identify two novel selenoprotein genes (Kryukov *et al.*, 1999; Lescure *et al.*, 1999). Efforts to identify additional selenoprotein genes in eukaryotic genomes soon followed. The inclusion of UGA as a selenocysteine codon in the prediction of optimal gene structures along the genome, however, led to the prediction of many erroneous selenoprotein genes (false positives). The low specificity of selenoprotein gene prediction was due to the difficulty of using the typical codon usage characteristic of proteins to extend them beyond an in-frame termination codon (Castellano *et al.*, 2001). A solution to this problem was to use the prediction of SECIS elements along the genome to limit the number of exons with a TGA in-frame that are incorporated into the dynamic programming recursion used to predict optimal genes. Several new selenoproteins in different species were identified this way (Castellano *et al.*, 2001; Kryukov *et al.*, 2003; Castellano *et al.*, 2005). Another example of stop codon recoding is pyrrolysine, which is encoded by UAG but may not require a specific stem loop structure in the 3' untranslated region for translation (Zhang *et al.*, 2005).

The Comparative and Homology Years

The Zoo Blot is an experimental technique which uses Southern blot analysis to test the ability of a nucleic acid probe from one species to hybridize with DNA of a range of species. Zoo Blots were already used in the 1980s to characterize the protein coding

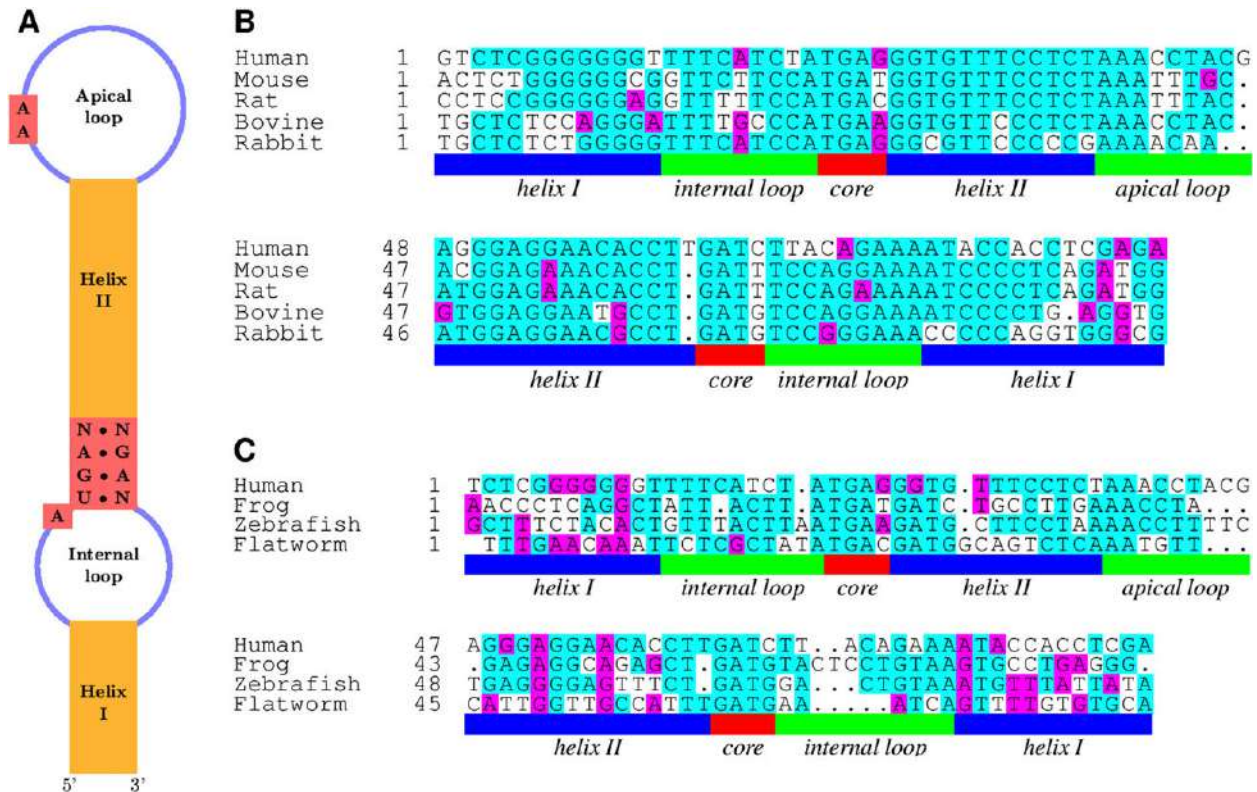


Fig. 5 (a) Schematic SECIS element (form 1) divided into structural units. Mostly invariant nucleotides are indicated. (b) Alignment of human and other mammalian glutathione peroxidase 1 SECIS sequences. Note the strong conservation among orthologous sequences. (c) Alignment of human and other non-mammalian glutathione peroxidase 1 SECIS sequences. Compared to the previous alignment, sequence conservation drops significantly.

regions of newly discovered genes (Monaco *et al.*, 1986; Katzav *et al.*, 1989). The rationale behind this approach was that protein coding regions will be conserved across species and would thus retain the capability to hybridize. Testing for hybridization of genomic probes to genomic libraries from many species will reveal the fragments likely to correspond to coding exons. As often happens, bioinformatics techniques based on sequence similarity replicate in the computer those experimental techniques based on the hybridization properties of nucleic acid molecules. In this regard, comparative gene prediction methods that rely on the fact that protein coding regions are generally more conserved during evolution than non-functional ones (Fig. 6), can be thought as a sort of electronic zoo blots. While phylogenetic footprinting – the phenomenon of the conservation of functional regions across species – has been used to discover, identify and characterize functional regions other than protein coding genes (Margulies *et al.*, 2003).

Essentially, there are two main classes of comparative approaches for gene identification: the comparison of the DNA query sequence with a target protein or mRNA sequence, or a database of such sequences (the computational equivalent of a Northern zoo blot), and the comparison of two or more genomic sequences. In both approaches query and target sequences may be from the same (as discussed above) or different species. If the latter, the selection of the target organism is not a negligible step; the organism should be chosen, when possible, at the phylogenetic distance maximizing the correlation between sequence conservation and coding function. Thus, allowing the inference of homology.

Using mRNAs and Proteins

The backbone of similarity-aided or similarity-based gene structure determination is constituted by those methods that rely on a comparison of the query sequence with protein, or mRNA sequences. Although mostly known as a database search program, BLASTX (Altschul *et al.*, 1990; Gish and States, 1993) illustrates the rationale behind this approach. With BLASTX a genomic query is translated into a set of amino acid sequences in the six possible frames and compared against a database of known protein sequences. The assumption is that those segments in the genomic query similar to database proteins are likely to correspond to homologous coding exons. A similar assumption is behind the comparison of the genomic query against a database of mRNA sequences (such as ESTs), using BLASTN (Altschul *et al.*, 1990), FASTA (Pearson and Lipman, 1988; Pearson, 2000), or similar programs. The National Center for Biotechnology Information GENBANK (Benson *et al.*, 2013) and UNIPROT (Bateman *et al.*, 2017), and their European and Asian counterparts, have been the main sources of nucleotide and protein sequences, respectively, for homology-based searches.

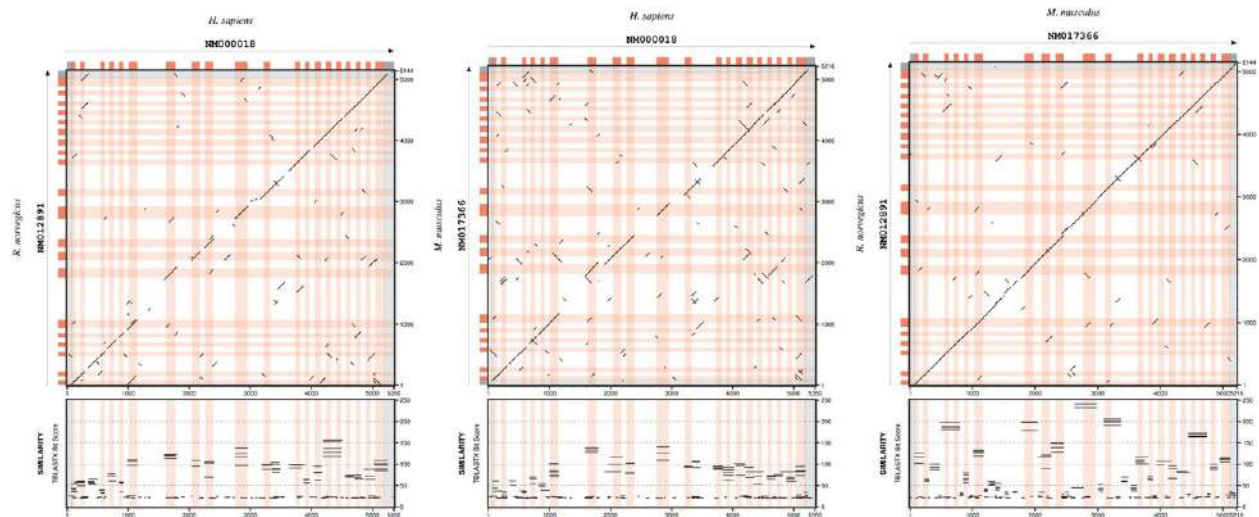


Fig. 6 Comparison of the orthologous genomic region of the acyl-Coenzyme A dehydrogenase very long chain (*ACADVL*) in human, mouse, and rat. The comparison was done at the amino acid level using the ungapped TBLASTX. Coding exons are shown as red boxes and non-coding ones (untranslated regions) are depicted in grey. Exons are projected to highlight their conservation (orange/grey ribbons). Note that introns are also conserved, albeit to a lesser extent than coding exons between human and any of the rodent genomes. The plot was obtained with GFF2APLOT.

Database search programs, however, are not dedicated gene prediction tools; they are not capable of automatically identifying start and stop codons or splice sites, the signals defining the exonic structure of the genes. Thus, after a database search and the identification of potential targets of homology, additional tools are required to define these structures. One approach is to use the top database match as target sequence and obtain a so-called spliced alignment between this and the genomic query. In such an alignment, large gaps – likely to correspond to introns – are only allowed at legal splice junctions. To calculate such spliced alignments between transcript and genomic sequences several programs we have already mentioned have been used: SIM4 (Florea *et al.*, 1998), EST_GENOME (Mott, 1997), SPIDEY (Wheelan *et al.*, 2001), and EXONERATE (Slater and Birney, 2005). Regarding the alignment of proteins of one species to another species genome, GENEWISE (Birney and Durbin, 1997, 2000) and EXONERATE (Slater and Birney, 2005) are used by ENSEMBL. In this direction, Meyer and Durbin (Meyer and Durbin, 2004) have developed the program PROJECTOR, which makes explicit use of the conservation of the exon-intron structure between two related genes. Because human and mouse orthologous genes show a remarkable conservation of their exonic structure, (Mouse Genome Sequencing Consortium, 2002), the PROJECTOR program outperforms GENEWISE predictions when human targets are used in mouse genomic sequences, and vice versa, in particular, when the conservation at the amino level is weak.

In an alternative approach, the results of a database search can be integrated *ad-hoc* into the framework of a typical *ab initio* gene prediction program. In essence, these methods promote candidate exons in the query sequence for which similar known coding sequences exist. Indeed, the score of the candidate exon – initially a function of the score of the splice (start, stop) sites and of the coding potential of the exon sequence – is increased as a function of the similarity between the candidate exon and the known coding sequences. In this way, candidate exons showing similarity to known coding sequences and thus likely orthologous, are promoted into the final gene prediction. In theory, this approach should produce predictions as accurate as pure *ab initio* programs when no similar target sequences exist, but more accurate ones (ideally, as accurate as those from splicing alignment tools) when such target sequences do exist. One example of this approach is the program GENOMESCAN (Yeh *et al.*, 2001), an extension of GENSCAN (Burge and Karlin, 1997) over which it reports increased accuracy. GRAILEXP (Xu *et al.*, 1997), CRASA (Chuang *et al.*, 2003) and AUGUSTUS (Stanke *et al.*, 2006a) are examples of methods using ESTs or mRNAs instead.

Using Whole-Genomes

With the increasing availability of the genome sequences for a wide range of eukaryotic organisms, whole genome sequence comparisons gained popularity as a mean of identifying protein coding genes. Under the assumption that regions conserved in genome sequence alignments will tend to correspond to coding exons from homologous genes, a number of programs were developed. The program EXOFISH (Crollius *et al.*, 2000) was one of the first such programs. Basically, it predicted human exons aided by the comparison of the human genome with sequences from *Tetraodon nigroviridis*, a puffer fish species (within Actinopterygii) that diverged about 400 Myr ago from the lineage later leading to humans. Latter developments followed notably different approaches.

In one such approach (Pedersen and Scharling, 2002; Blayo *et al.*, 2003), the problem was stated as a generalization of pairwise sequence alignment: given two genomic sequences coding for homologous genes, the goal was to obtain the predicted exonic structure in each sequence maximizing the score of the alignment of the resulting amino acid sequences. Both, (Blayo *et al.*, 2003)

and Pedersen and Scharling (Pedersen and Scharling, 2002) solved this problem through a complex extension of the classical dynamic programming algorithm for sequence alignment.

In a different approach, the programs SLAM (Alexandersson *et al.*, 2003) and DOUBLESCAN (Meyer and Durbin, 2002) combined sequence alignment pair HMMs (Durbin *et al.*, 1998), with gene prediction generalized HMMs (Burge and Karlin, 1997) into the so-called generalized pair HMMs. In these, gene prediction is not the result of the sequence alignment, as in the programs above, but both gene prediction and sequence alignment are obtained simultaneously.

A third class of programs adopted a more heuristic approach, and clearly separated gene prediction from sequence alignment. The programs ROSETTA (Batzoglou *et al.*, 2000), SGP1 (from Syntenic Gene Prediction, (Wiehe *et al.*, 2001)), and CEM (from the Conserved Exon Method, (Bafna and Huson, 2000)) are representative of this approach. All these programs start by aligning two syntenic sequences – understood here as sequences from different species, but containing homologous genes in a similar order –, and then predict gene structures in which the exons are compatible with the alignment.

Although similarity-based gene prediction with homologous genomic sequences produced high quality results (Miller, 2001), an obvious shortcoming was the need for two homologous sequences. Also, genes without a homologue in the partner sequence will escape detection. This is particularly problematic if species are compared at genomic regions where synteny is not preserved. Given only a single query sequence, it is therefore desirable to automatically search for homologous sequences or syntenic chromosome stretches in other species that are suited to similarity-based approaches. The programs TWINSCAN (Korf *et al.*, 2001) and SGP2 (Parra *et al.*, 2003) attempted to address this limitation. The approach in these programs was reminiscent of that used in GENOMESCAN (Yeh *et al.*, 2001). Essentially, the sequence from the query genome is compared against a collection of sequences from the target or informant genome – which can be a single homologous sequence to the query sequence, a whole assembled genome, or a collection of shotgun reads –, and the results of the comparison are used to modify accordingly the scores of the exons produced by *ab initio* gene predictors. In TWINSCAN, the query and target genome sequences are compared at the nucleotide level with BLASTN and the results serve to modify the underlying probability of the potential exons predicted by GENSCAN. In SGP2, the genome sequences are compared using TBLASTX (Altschul *et al.*, 1990), and the results used to modify the scores of the potential exons predicted by GENEID.

TWINSCAN, SGP2, and SLAM were successfully applied to the annotation of the mouse genome (Mouse Genome Sequencing, C, 2002), and helped to identify previously unconfirmed genes (Guigo *et al.*, 2003). Importantly, a number of studies consistently indicate that comparative gene finders improve over their *ab initio* counterparts (Parra *et al.*, 2003; Wu *et al.*, 2004). Indeed, in Table 1, we show that, when evaluated in human chromosome 22, TWINSCAN and SGP2 outperformed GENSCAN and GENEID, respectively. The exhaustive scrutiny to which the sequence of human chromosome 22 (Dunham *et al.*, 1999) was subjected through the Vertebrate Genome Annotation database project at the Wellcome Sanger Institute offers an excellent platform to obtain estimates of the accuracy of current gene finders.

Annotating a genome has required a complex craftsmanship from its inception, given the many different layers of annotations that have to be considered and therefore put together. Common to the tools and approaches described above is the first task of masking the repetitive sequences, often taxa-specific, that populate the genome; this step facilitates and speeds up the downstream analyses just by replacing the segments where the repeats are found by N's, a common nucleotide wildcard – meaning A or C or G or T-. RepeatMasker (Smit *et al.*, 2013-2015), has been the classical tool for this purpose, yet it has been described as sensitive but computationally costly when applied to large eukaryotic genomes (Bedell *et al.*, 2000). After repeat-masking, genome sequences are scanned for protein-coding and non-coding genes using different prediction tools while, simultaneously, homology-based searches and whole-genome alignments to other species are performed. The result are sets of annotations with varying degrees of supporting evidence that need to be integrated into one set of annotated genes, also known as the gene-build of a genome. Additional genome annotation layers result from high-throughput experimental approaches, such as RNA-seq and the transcriptional evidence it produces (which may be used on the gene-build itself), or the typing and resequencing experiments that sample sequence variation across populations (human or otherwise), and so on. Genes from the gene-build, and the proteins they encode, require these other annotation layers to assign them a molecular function, a process known as functional annotation. These functions rely on controlled vocabularies to describe them, for instance the Gene Ontology (GO) (Ashburner *et al.*, 2000; The Gene Ontology Consortium, 2017).

As the complexity of the annotation process increased due to the need to integrate new layers of information, guidelines and/or automatic or semiautomatic workflows to annotate novel genomes or to re-annotate existing ones (every time the assemblies are improved) were developed (Mudge and Harrow, 2016; Elsik *et al.*, 2014). For example, the NCBI prokaryotic (Tatusova *et al.*, 2016) and eukaryotic (Thibaud-Nissen *et al.*, 2016) pipelines, or EnSEMBL (Potter *et al.*, 2004), which rely on in-house protocols for large-scale genome annotation. In addition, gene-finding tools were wrapped into specialized genome annotation workflows. For instance, the Fgenesh (Salamov and Solovyev, 2000), SNAP+Exonerate, and GeneMark+ AUGUSTUS were integrated into the Fgenesh++ (Solovyev *et al.*, 2006), MAKER (Cantarel *et al.*, 2008), and BRAKER (Hoff *et al.*, 2016) annotation pipelines, respectively.

Comparative genomics can be also useful to assess the completeness of genome annotations. REFSEQ (Pruitt *et al.*, 2012) has been used as a reference annotation set for the annotation of novel gene-builds, but also to validate the completeness of genome assemblies. A set of 458 highly reliable core proteins, conserved across plants, fungi and mammals, was collected and integrated into a gene-finding validation pipeline known as CEGMA (Parra *et al.*, 2007). More recently, it has been superseded by a set of single-copy orthologs in the BUSCO pipeline (Simao *et al.*, 2015), which can be used to benchmark the completeness of both the annotation of genes and the genome assembly.

Annotation on Non-Standard Protein-Coding Genes

Comparative methods for gene prediction have been also useful to annotate selenoprotein genes. The fact that downstream of a recoded stop codon lies a real protein implies this region to have the typical pattern of sequence conservation among its protein homologues. At the least, to the extent of the conservation in the upstream protein sequence. Therefore, sequence conservation beyond a TGA codon in the genome strongly suggests selenocysteine coding function. It is then feasible to predict selenoprotein genes in two different species genomes, without information from their SECIS elements, and compare them to identify a pattern of symmetrical protein sequence conservation around a predicted selenocysteine codon. Application of this approach to the human and fugu genomes identified a new selenoprotein in this fish (Castellano *et al.*, 2004). More recently, selenoproteins have been annotated across eukaryotes, in a SECIS-independent manner, using alignment profiles derived from the known selenoprotein families (Mariotti and Guigo, 2010).

The RNA and Regulatory Years

The functional annotation of the regulatory and non-coding part of the genome has lagged behind the annotation of protein-coding genes. Systematic analysis of RNA genes and regulatory regions was made possible in the mid and late 2000s by a number of experimental approaches for the characterization of transcribed regions, transcription factor binding sites, DNA methylation sites and chromatin structure. The ENCODE project was an early adopter of these methods to systematically characterize the human genome, (2012 ENCODE Project Consortium). Among these, tiling arrays and RNA-seq have been important to understand the transcriptional landscape of the human genome, which is pervasively transcribed (Djebali *et al.*, 2012; Bertone *et al.*, 2004). This was perhaps not unexpected if one acknowledged the high initiation promiscuity of *Pol II*, the eukaryotic RNA polymerase complex (Struhl, 2007). In any case, transcription from regions where protein-coding genes or known RNA genes – such as ribosomal and transfer RNAs, small nuclear RNAs involved in splicing, microRNAs, short interfering RNAs, and a few others – have so far not been annotated, produced thousands of long noncoding RNAs with no obvious function. These unknown RNAs (Derrien *et al.*, 2012), which standard gene prediction programs cannot detect using codon bias measures, constitute a vast array of potential genes that do not encode for proteins (Kowalczyk *et al.*, 2012), albeit some may do (Ji *et al.*, 2015). While a few of these new RNAs have been shown to have functions (Nesterova *et al.*, 2001; Sleutels *et al.*, 2002), there is little functional evidence for the majority of them. Their transcription is often extremely low and they tend to have limited sequence or RNA structure conservation (Rivas *et al.*, 2017), raising the possibility that they are transcriptional noise. Indeed, between and within species analyses indicate that most of these RNA genes are not under strong evolutionary constraint (purifying selection) (Wiberg *et al.*, 2015). Alternatively, it could be their transcription itself rather than their sequence that is under selection. The functional annotation of RNA genes thus remains open and additional work is needed before this controversy is solved.

The high-throughput sequencing of transcriptomes and other experimental approaches has required the development of fast aligners like BWA (Li and Durbin, 2009) and Bowtie (Langmead and Salzberg, 2012), which can align millions of short sequences to a reference genome sequence. Specialized tools can integrate those alignments into predicted gene structures, such as AUGUSTUS (Stanke *et al.*, 2006b), mGENE (Schweikert *et al.*, 2009), or the Transomics pipeline (Solovyev *et al.*, 2006). Other tools, like SpliceGrapher (Rogers *et al.*, 2012), have been developed to refine the alignments of mapped sequences to predict “splicing graphs”, a representation of the splicing variants of a gene in which exons are shown as nodes that are connected by the intervening introns represented as edges. As discussed, obtaining simultaneously the genome and transcriptome of a species facilitates the task of gene annotation, by aligning the transcriptome sequences along the newly assembled genome, and provides an estimation of gene expression levels when translating the alignment coverage of read sequences into normalized counts (Mortazavi *et al.*, 2008). An extensive assessment of the sequence mappers (Engstrom *et al.*, 2013) and the gene-finders (Steijger *et al.*, 2013) was undertaken by the GENCODE consortium (Harrow *et al.*, 2012). Among the results obtained it is worth mentioning that 5'- and 3'-UTR exons were still difficult to predict (see Fig. 3) as well as the different exon combinations for splicing transcript isoforms, and that prediction accuracy correlated with the gene expression level, as highly expressed genes are often more abundant in the training sets. However, this is changing as single molecule sequencing methods, such as PacBio or ONT nanopore, produce long sequences recovering the full length splicing isoforms of a gene (Sharon *et al.*, 2013). Thus easing the task of protein-coding and non-coding mRNA genes annotation, even at single cell resolution (Byrne *et al.*, 2017).

Regulatory regions where proteins (*e.g.*, transcription factors) bind have been studied using chromatin immunoprecipitation followed by sequencing (ChIP-seq). Around 8% of the genome across different cell types is typically protein-bound with sequence-specific signals known to bind proteins being common in them. In addition, *DNase I* hypersensitivity has been used to identify those regions of the genome open to transcription. These regions can be bound by transcription factors and were found to be upstream of the predicted transcription start sites, coinciding with the regions defined by the ChIP-seq experiments. Methylation of cytosine, usually at CpG dinucleotides, is involved in the epigenetic regulation of gene expression, with promoter and gene methylation leading to repression and promotion of transcription, respectively. Levels of DNA methylation correlate with chromatin accessibility as defined by the *DNase I* hypersensitivity experiments. Interestingly, most variable regions in their methylation correspond to genes rather than their promoters.

One surprising result from the ENCODE project was the finding that the majority (about 80%) of the human genome participated in one or more of the experiments above (ENCODE Project Consortium, 2012), mostly in the RNA transcription

experiments discussed above. This was initially interpreted as providing evidence for a functional role to most of the human DNA sequence but low specificity of the assays used is a simpler explanation. These assays often do not distinguish between meaningful function – some molecular function that impacts fitness in the individual – and one that does not. As a result, having the majority of the genome performing important roles contrasts with the small fraction of it – probably 5%–10% – that has been shown to be under evolutionary constraint (Asthana *et al.*, 2007; Davydov *et al.*, 2010). It follows that some fraction of non-coding DNA is functional in the typical meaning of this word but that the majority of it – mostly transposable elements that make up about half of the human genome (Eddy, 2012) – is not (albeit being biochemically active in different experiments). Indeed, one of the common tasks in gene prediction is masking about half of the human genome for repeats and transposons using the venerable REPEATMASKER program.

The Population Years

The functional annotation of a species genome provides a reference to which additional genomes from the same species can be compared. Thousands of human exomes (protein-coding fraction of the genome) or genomes have now been sequenced (1k Genomes Project Consortium, 2012, 2015) and, in doing so, many differences have been found between them and their reference – *e.g.*, between four and five million nucleotide differences in each human genome –. These differences, which include single nucleotide variants, insertions and deletions, copy number variants and others, are often novel and unannotated, which each ensuing genome having more of them. These rare variants in humans, which are found in specific populations and at low frequency, are to a large the result of the recent growth of human populations (Keinan and Clark, 2012). Indeed, humans have accumulated in the last 5000–10,000 years an enormous number of rare variants, particularly in non-African populations. Protein-coding variants at low frequency are often slightly deleterious as measured by approaches that infer the probability of amino acid changes impacting the structure or function of proteins (Fu *et al.*, 2013). Programs like POLYPHEN (Adzhubei *et al.*, 2010) and SIFT (Kumar *et al.*, 2009), using amino acid conservation (within and between species) and different amino acid properties, annotate their deleteriousness across the genome. In addition, a few hundred variants disrupt the translation of proteins in a typical human genome. Non-coding variants in regulatory regions have also been annotated with regard to their deleteriousness using, for example, their conservation among species with PHASTCONS (Siepel *et al.*, 2005). It follows that the functional annotation of single nucleotide variants, particularly those that change (amino acid substitutions) or disrupt proteins (*e.g.*, changing splice sites or creating stop codons), are important to understand interindividual differences linked to diseases that are both rare and severe. Indeed, rare diseases, understood as those that occur in one out of 2000 individuals, have often a genetic basis and the annotation of the deleteriousness of the causal mutation(s) in single-gene disorders is the first step in their diagnosis and treatment.

Still, most of the genetic variation between individuals genomes of a species is shared among some (or most) of them. This common variation was described in a landmark study that provided not only a catalog of single nucleotide variants in humans but the genetic association among them (International HapMap Consortium, 2005). This association, known as linkage disequilibrium, reflects the coinheritance of sets of single nucleotide variants (haplotypes) and their functional annotation is needed to investigate, using genome-wide association studies, the hereditary factors involved in disease. This is important as each human genome has around 10,000 rare and common amino acid changes, a few hundred protein-disrupting ones and about 500,000 non-coding changes in regulatory regions (1k Genomes Project Consortium, 2015). Of these, about 2000 are linked through genome-wide studies with common disease whereas a few dozen are implicated with rare disease per genome. Today, thousands of mutations have been associated to common or rare disease in humans and their genome and functional annotation can be obtained from different databases, for example OMIM (<http://omim.org>), CLINVAR (Landrum *et al.*, 2016) and the NHGRI-EBI GWAS catalog (MacArthur *et al.*, 2017).

Conclusion

Genome annotation has come a long way since the first protein-coding genes were annotated using *ab initio* and later comparative approaches. Experimental evidence is now routinely integrated in the annotation of these genes and their multiple alternative splice forms. At the same time, promoters and smaller regulatory elements, as well as RNA genes (including controversial ones), are being annotated with a variety of experimental approaches genome-wide. Knowing the location of these coding and noncoding functional features has provided the framework to understand the role of sequence variants in health and disease, which remains largely undetermined. Still, since the early nineties, when the Human Genome Project was launched, our knowledge of genome biology has greatly improved. This is not only due to the ever increasing computational capabilities but also to novel discoveries in molecular biology from experimental methods with increased resolution and throughput, which has been harnessed in annotation consortiums like the ENCODE project. As a result, well annotated genomes like that of human, fly or mice, the result of years of intense human curation, as well as gold standards such REFSEQ or VEGA genes, provide reference annotations to other species.

This is important as the pace of sequencing new species genomes, transcriptomes, and environmental (metagenomics) samples is only accelerating given the ever decreasing cost of sequencing technologies. Thus, a shift towards more automated annotation pipelines is warranted to cope with the scale of the analyses in genome annotation. Automated annotation pipelines have room to

improve though, with gene-builds still having a significant impact in the downstream analyses. For example, on the variants found in a genome and their predicted functional consequences (Frankish *et al.*, 2015). This is due to the fact that most gene-finding tools return a single and often different set of optimal gene structures. This also poses a problem for overlapping genes, including those on different strands, nested genes, genes located within introns of other genes, non-standard gene structures, and genes with translation recoding like selenoprotein genes, as they are often not included in the optimal prediction of genes. Furthermore, most gene prediction tools do not consider frame shifts, RNA editing, and the impact of sequencing errors in the identification of gene structures. Still, some tools are capable to infer more than one alternative splicing isoform provided that there is some sort of experimental evidence, for example, from RNA-seq experiments. However, problems may appear when transcripts contain start or stop codons split by introns. Together with non-canonical splice-sites, these are some of the special cases that human curators solve in the final refinement steps of genome annotation when all sort of evidences are integrated. Many of aforementioned issues will be overcome by the use of single molecule sequencing technologies, which make possible to map the full length of individual protein-coding and non-coding transcript isoforms. In combination with single cell gene expression experiments, it will be finally feasible to produce gene-builds and their corresponding functional annotation at the single molecule and cell resolution, an astounding promise just two decades ago.

Acknowledgements

This study was partially funded (Sergi Castellano) by the NIHR HS&DR Programme (14/21/45) and supported by the NIHR GOSH BRC. The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.

See also: Algorithms for Strings and Sequences: Multiple Alignment. Algorithms for Strings and Sequences: Pairwise Alignment. Bioinformatics Approaches for Studying Alternative Splicing. Comparative and Evolutionary Genomics. Comparative Genomics Analysis. Data Mining: Classification and Prediction. Data Mining: Prediction Methods. Detecting and Annotating Rare Variants. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Functional Genomics. Gene Mapping. Genome Annotation: Perspective From Bacterial Genomes. Genome Databases and Browsers. Genome Informatics. Genome-Wide Haplotype Association Study. Hidden Markov Models. Integrative Analysis of Multi-Omics Data. Linkage Disequilibrium. Metagenomic Analysis and its Applications. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Ontology in Bioinformatics. Ontology-Based Annotation Methods. Phylogenetic Footprinting. Prediction of Coding and Non-Coding RNA. Protein Functional Annotation. Quantitative Immunology by Data Analysis Using Mathematical Models. Sequence Analysis. Sequence Composition. Single Nucleotide Polymorphism Typing. Whole Genome Sequencing Analysis

References

- Abril, J.F., Castelo, R., Guigo, R., 2005. Comparison of splice sites in mammals and chicken. *Genome Res.* 15 (1), 111–119.
- Abril, J.F., Guigo, R., 2000. gff2ps: Visualizing genomic annotations. *Bioinformatics* 16 (8), 743–744.
- Abril, J.F., Guigo, R., Wiehe, T., 2003. gff2aplot: Plotting sequence comparisons. *Bioinformatics* 19 (18), 2477–2479.
- Adams, M.D., *et al.*, 2000. The genome sequence of *Drosophila melanogaster*. *Science* 287 (5461), 2185–2195.
- Adzhubei, I.A., *et al.*, 2010. A method and server for predicting damaging missense mutations. *Nat. Methods* 7 (4), 248–249.
- Aken, B.L., *et al.*, 2016. The Ensembl gene annotation system. *Database* (Oxford) 2016.
- Aken, B.L., *et al.*, 2017. Ensembl 2017. *Nucleic Acids Res.* 45 (D1), D635–D642.
- Alexandersson, M., Cawley, S., Pachter, L., 2003. SLAM: Cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.* 13 (3), 496–502.
- Altschul, S.F., *et al.*, 1990. Basic local alignment search tool. *J. Mol. Biol.* 215 (3), 403–410.
- Ashburner, M., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25 (1), 25–29.
- Asthana, S., *et al.*, 2007. Widely distributed noncoding purifying selection in the human genome. *Proc. Natl. Acad. Sci. USA* 104 (30), 12410–12415.
- Bafna, V., Huson, D.H., 2000. The conserved exon method for gene finding. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 8, 3–12.
- Bateman, A., *et al.*, 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Res.* 45 (D1), D158–D169.
- Batzoglou, S., *et al.*, 2000. Human and mouse gene structure: Comparative analysis and application to exon prediction. *Genome Res.* 10 (7), 950–958.
- Bedell, J.A., Korf, I., Gish, W., 2000. MaskerAid: A performance enhancement to RepeatMasker. *Bioinformatics* 16 (11), 1040–1041.
- Benson, D.A., *et al.*, 2013. GenBank. *Nucleic Acids Res.* 41 (D1), D36–D42.
- Bertone, P., *et al.*, 2004. Global identification of human transcribed sequences with genome tiling arrays. *Science* 306 (5705), 2242–2246.
- Birney, E., Durbin, R., 1997. Dynamite: A flexible code generating language for dynamic programming methods used in sequence comparison. In: *Proceedings of Ismb-97 – Fifth International Conference on Intelligent Systems for Molecular Biology*, pp. 56–64.
- Birney, E., Durbin, R., 2000. Using GeneWise in the *Drosophila* annotation experiment. *Genome Res.* 10 (4), 547–548.
- Blayo, P., Rouze, P., Sagot, M.F., 2003. Orphan gene finding – An exon assembly approach. *Theor. Comput. Sci.* 290 (3), 1407–1431.
- Burge, C., Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 268 (1), 78–94.
- Burset, M., Guigo, R., 1996. Evaluation of gene structure prediction programs. *Genomics* 34 (3), 353–367.
- Burset, M., Seledtsov, I.A., Solovyev, V.V., 2000. Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Res.* 28 (21), 4364–4375.
- Byrne, A., *et al.*, 2017. Nanopore long-read RNAseq reveals widespread transcriptional variation among the surface receptors of individual B cells. *Nat. Commun.* 8, 16027.
- Cantarel, B.L., *et al.*, 2008. MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Res.* 18 (1), 188–196.

- Castellano, S., *et al.*, 2001. *In silico* identification of novel selenoproteins in the *Drosophila melanogaster* genome. EMBO Rep. 2 (8), 697–702.
- Castellano, S., *et al.*, 2004. Reconsidering the evolution of eukaryotic selenoproteins: A novel nonmammalian family with scattered phylogenetic distribution. EMBO Rep. 5 (1), 71–77.
- Castellano, S., *et al.*, 2005. Diversity and functional plasticity of eukaryotic selenoproteins: Identification and characterization of the SelJ family. Proc. Natl. Acad. Sci. USA 102 (45), 16188–16193.
- Chambers, I., *et al.*, 1986. The structure of the mouse glutathione peroxidase gene: The selenocysteine in the active site is encoded by the 'termination' codon, TGA. EMBO J. 5 (6), 1221–1227.
- Chuang, T.J., *et al.*, 2003. A complexity reduction algorithm for analysis and annotation of large genomic sequences. Genome Res. 13 (2), 313–322.
- Coghlan, A., *et al.*, 2008. nGASP – The nematode genome annotation assessment project. BMC Bioinform. 9, 549.
- E.PENCODE, Project Consortium, 2012. An integrated encyclopedia of DNA elements in the human genome. Nature 489 (7414), 57–74.
- Crollius, H.R., *et al.*, 2000. Estimate of human gene number provided by genome-wide analysis using *Tetraodon nigroviridis* DNA sequence. Nat. Genet. 25 (2), 235–238.
- Curwen, V., *et al.*, 2004. The Ensembl automatic gene annotation system. Genome Res. 14 (5), 942–950.
- Davydov, E.V., *et al.*, 2010. Identifying a high fraction of the human genome to be under selective constraint using GERP plus. PLOS Comput. Biol. 6 (12), 1–12.
- Delcher, A.L., *et al.*, 1999. Improved microbial gene identification with GLIMMER. Nucleic Acids Res. 27 (23), 4636–4641.
- Derrien, T., *et al.*, 2012. The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. Genome Res. 22 (9), 1775–1789.
- Djebali, S., *et al.*, 2012. Landscape of transcription in human cells. Nature 489 (7414), 101–108.
- Dunham, I., *et al.*, 1999. The DNA sequence of human chromosome 22. Nature 402 (6761), 489–495.
- Durbin, R., *et al.*, 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge, United Kingdom: Cambridge University Press, (xi, 356 pages; Illustrations; 26 cm).
- Eddy, S.R., 2004a. What is dynamic programming? Nat. Biotechnol. 22 (7), 909–910.
- Eddy, S.R., 2004b. What is a hidden Markov model? Nat. Biotechnol. 22 (10), 1315–1316.
- Eddy, S.R., 2012. The C-value paradox, junk DNA and ENCODE. Curr. Biol. 22 (21), R898–R899.
- Eilbeck, K., *et al.*, 2009. Quantitative measures for the management and comparison of annotated genomes. BMC Bioinform. 10, 67.
- Elsik, C.G., *et al.*, 2014. Finding the missing honey bee genes: Lessons learned from a genome upgrade. BMC Genomics 15, 86.
- Engstrom, P.G., *et al.*, 2013. Systematic evaluation of spliced alignment programs for RNA-seq data. Nat. Methods 10 (12), 1185–1191.
- Florea, L., *et al.*, 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. Genome Res. 8 (9), 967–974.
- Frankish, A., *et al.*, 2015. Comparison of GENCODE and RefSeq gene annotation and the impact of reference geneset on variant effect prediction. BMC Genom. 16 (Suppl. 8), S2.
- Fu, W.Q., *et al.*, 2013. Analysis of 6515 exomes reveals the recent origin of most human protein-coding variants. Nature 493 (7431), 216–220.
- Gelfand, M.S., Mironov, A.A., Pevzner, P.A., 1996. Gene recognition via spliced sequence alignment. Proc. Natl. Acad. Sci. USA 93 (17), 9061–9066.
- 1k Genomes Project Consortium, *et al.*, 2012. An integrated map of genetic variation from 1092 human genomes. Nature 491 (7422), 56–65.
- 1k Genomes Project Consortium, *et al.*, 2015. A global reference for human genetic variation. Nature 526 (7571), 68–74.
- Giardine, B., *et al.*, 2005. Galaxy: A platform for interactive large-scale genome analysis. Genome Res. 15 (10), 1451–1455.
- Gish, W., States, D.J., 1993. Identification of protein coding regions by database similarity search. Nat. Genet. 3 (3), 266–272.
- Gotoh, O., 1982. An improved algorithm for matching biological sequences. J. Mol. Biol. 162 (3), 705–708.
- Guigo, R., *et al.*, 1992. Prediction of gene structure. J. Mol. Biol. 226 (1), 141–157.
- Guigo, R., 1998. Assembling genes from predicted exons in linear time with dynamic programming. J. Comput. Biol. 5 (4), 681–702.
- Guigo, R., *et al.*, 2003. Comparison of mouse and human genomes followed by experimental verification yields an estimated 1019 additional genes. Proc. Natl. Acad. Sci. USA 100 (3), 1140–1145.
- Guigo, R., *et al.*, 2006. EGASP: The human ENCODE genome annotation assessment project. Genome Biol. 7 (Suppl. 1), S2 1–31.
- Harrow, J., *et al.*, 2012. GENCODE: The reference human genome annotation for The ENCODE project. Genome Res. 22 (9), 1760–1774.
- Hastings, K.E.M., 2005. SL trans-splicing: Easy come or easy go? Trends Genet. 21 (4), 240–247.
- Hoff, K.J., *et al.*, 2016. BRAKER1: Unsupervised RNA-seq-based genome annotation with GeneMark-ET and AUGUSTUS. Bioinformatics 32 (5), 767–769.
- Hubbard, T., *et al.*, 2002. The Ensembl genome database project. Nucleic Acids Res. 30 (1), 38–41.
- Hyatt, D., *et al.*, 2010. Prodigal: Prokaryotic gene recognition and translation initiation site identification. BMC Bioinform. 11, 119.
- International HapMap Consortium, C., 2005. A haplotype map of the human genome. Nature 437 (7063), 1299–1320.
- Iseli, C., Jongeneel, C.V., Bucher, P., 1999. ESTScan: A program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. Proc. Int. Conf. Intell. Syst. Mol. Biol. 138–148.
- Ji, Z., *et al.*, 2015. Many lncRNAs, 5' UTRs, and pseudogenes are translated and some are likely to express functional proteins. eLife 4.
- Katzav, S., Martin-Zanca, D., Barbacid, M., 1989. *vav*, a novel human oncogene derived from a locus ubiquitously expressed in hematopoietic cells. EMBO J. 8 (8), 2283–2290.
- Keinan, A., Clark, A.G., 2012. Recent explosive human population growth has resulted in an excess of rare genetic variants. Science 336 (6082), 740–743.
- Kent, W.J., *et al.*, 2002. The human genome browser at UCSC. Genome Res. 12 (6), 996–1006.
- Korf, I., *et al.*, 2001. Integrating genomic homology into gene structure prediction. Bioinformatics 17 (Suppl. 1), S140–S148.
- Kowalczyk, M.S., Higgs, D.R., Gingeras, T.R., 2012. Molecular biology: RNA discrimination. Nature 482 (7385), 310–311.
- Kryukov, G.V., *et al.*, 2003. Characterization of mammalian selenoproteomes. Science 300 (5624), 1439–1443.
- Kryukov, G.V., Kryukov, V.M., Gladyshev, V.N., 1999. New mammalian selenocysteine-containing proteins identified with an algorithm that searches for selenocysteine insertion sequence elements. J. Biol. Chem. 274 (48), 33888–33897.
- Kumar, P., Henikoff, S., Ng, P.C., 2009. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. Nat. Protoc. 4 (7), 1073–1082.
- Kyriakopoulou, C., *et al.*, 2006. U1-like snRNAs lacking complementarity to canonical 5' splice sites. RNA 12 (9), 1603–1611.
- Krzywinski, M., *et al.*, 2009. Circos: an Information Aesthetic for Comparative Genomics. Genome Res. 19, 1639–1645.
- Landrum, M.J., *et al.*, 2016. ClinVar: Public archive of interpretations of clinically relevant variants. Nucleic Acids Res. 44 (D1), D862–D868.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. Nat. Methods 9 (4), 357–359.
- Lescur, A., *et al.*, 1999. Novel selenoproteins identified *in silico* and *in vivo* by using a conserved RNA structural motif. J. Biol. Chem. 274 (53), 38147–38154.
- Lewis, S.E., *et al.*, 2002. Apollo: A sequence annotation editor. Genome Biol. 3 (12), (RESEARCH0082).
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25 (14), 1754–1760.
- Loveland, J.E., *et al.*, 2012. Community gene annotation in practice. Database – J. Biol. Databases Curation.
- Lukashin, A.V., Borodovsky, M., 1998. GeneMark.hmm: New solutions for gene finding. Nucleic Acids Res. 26 (4), 1107–1115.
- MacArthur, J., *et al.*, 2017. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). Nucleic Acids Res. 45 (D1), D896–D901.
- Madupu, R., *et al.*, 2010. Meeting report: A workshop on best practices in genome annotation. Database (Oxford) 2010, baq001.
- Margulies, E.H., *et al.*, 2003. Identification and characterization of multi-species conserved sequences. Genome Res. 13 (12), 2507–2518.
- Mariotti, M., Guigo, R., 2010. Selenoprofiles: Profile-based scanning of eukaryotic genome sequences for selenoprotein genes. Bioinformatics 26 (21), 2656–2663.

- Meyer, I.M., Durbin, R., 2002. Comparative *ab initio* prediction of gene structures using pair HMMs. *Bioinformatics* 18 (10), 1309–1318.
- Meyer, I.M., Durbin, R., 2004. Gene structure conservation aids similarity based gene prediction. *Nucleic Acids Res.* 32 (2), 776–783.
- Miller, W., 2001. Comparison of genomic DNA sequences: Solved and unsolved problems. *Bioinformatics* 17 (5), 391–397.
- Monaco, A.P., *et al.*, 1986. Isolation of candidate cDNAs for portions of the Duchenne muscular dystrophy gene. *Nature* 323 (6089), 646–650.
- Mortazavi, A., *et al.*, 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods* 5 (7), 621–628.
- Mott, R., 1997. EST_GENOME: A program to align spliced DNA sequences to unspliced genomic DNA. *Comput. Appl. Biosci.* 13 (4), 477–478.
- Mouse Genome Sequencing Consortium, 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* 420 (6915), 520–562.
- Mudge, J.M., Harrow, J., 2016. The state of play in higher eukaryote gene annotation. *Nat. Rev. Genet.* 17 (12), 758–772.
- Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48 (3), 443–453.
- Nesterova, T.B., *et al.*, 2001. Characterization of the genomic Xist locus in rodents reveals conservation of overall gene structure and tandem repeats but rapid evolution of unique sequence. *Genome Res.* 11 (5), 833–849.
- Parra, G., *et al.*, 2003. Comparative gene prediction in human and mouse. *Genome Res.* 13 (1), 108–117.
- Parra, G., *et al.*, 2006. Tandem chimerism as a means to increase protein complexity in the human genome. *Genome Res.* 16 (1), 37–44.
- Parra, G., Bradnam, K., Korf, I., 2007. CEGMA: A pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics* 23 (9), 1061–1067.
- Patel, A.A., Steitz, J.A., 2003. Splicing double: Insights from the second spliceosome. *Nat. Rev. Mol. Cell Biol.* 4 (12), 960–970.
- Pearson, W.R., 2000. Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol.* 132, 185–219.
- Pearson, W.R., Lipman, D.J., 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA* 85 (8), 2444–2448.
- Pedersen, C.N.S., Scharling, T., 2002. Comparative methods for gene structure prediction in homologous sequences. *Proc. Algorithms Bioinform.* 2452, 220–234.
- Pennisi, E., 2000. Ideas fly at gene-finding jamboree. *Science* 287 (5461), 2182. +.
- Plotkin, J.B., Kudla, G., 2011. Synonymous but not the same: The causes and consequences of codon bias. *Nat. Rev. Genet.* 12 (1), 32–42.
- Potter, S.C., *et al.*, 2004. The Ensembl analysis pipeline. *Genome Res.* 14 (5), 934–941.
- Powers, D., 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Int. J. Mach. Learn. Technol.* 2, 37–63.
- Pruitt, K.D., *et al.*, 2012. NCBI Reference Sequences (RefSeq): Current status, new features and genome annotation policy. *Nucleic Acids Res.* 40 (Database issue), D130–D135.
- Rayman, M.P., 2000. The importance of selenium to human health. *Lancet* 356 (9225), 233–241.
- Reese, M.G., *et al.*, 2000. Genome annotation assessment in *Drosophila melanogaster*. *Genome Res.* 10 (4), 483–501.
- Rivas, E., Clements, J., Eddy, S.R., 2017. A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. *Nat. Methods* 14 (1), 45–48.
- Rogers, M.F., *et al.*, 2012. SpliceGrapher: Detecting patterns of alternative splicing from RNA-Seq data in the context of gene models and EST data. *Genome Biol.* 13 (1), R4.
- Salamov, A.A., Solovyev, V.V., 2000. *Ab initio* gene finding in *Drosophila* genomic DNA. *Genome Res.* 10 (4), 516–522.
- Schweikert, G., *et al.*, 2009. mGene: Accurate SVM-based gene finding with an application to nematode genomes. *Genome Res.* 19 (11), 2133–2143.
- Sharon, D., *et al.*, 2013. A single-molecule long-read survey of the human transcriptome. *Nat. Biotechnol.* 31 (11), 1009–1014.
- Sharp, P.A., 2005. The discovery of split genes and RNA splicing. *Trends Biochem. Sci.* 30 (6), 279–281.
- Siepel, A., *et al.*, 2005. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res.* 15 (8), 1034–1050.
- Simao, F.A., *et al.*, 2015. BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics* 31 (19), 3210–3212.
- Slater, G.S., Birney, E., 2005. Automated generation of heuristics for biological sequence comparison. *BMC Bioinform.* 6, 31.
- Slueteels, F., Zwart, R., Barlow, D.P., 2002. The non-coding Air RNA is required for silencing autosomal imprinted genes. *Nature* 415 (6873), 810–813.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147 (1), 195–197.
- Smit, A.F.A., Hubley, R., Green, P., *RepeatMasker Open-4.0*, 2013–2015, <http://www.repeatmasker.org>.
- Solovyev, V., *et al.*, 2006. Automatic annotation of eukaryotic genes, pseudogenes and promoters. *Genome Biol.* 7 (Suppl. 1), S10 1–12.
- Stanke, M., *et al.*, 2006a. Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources. *BMC Bioinform.* 7.
- Stanke, M., Tzvetkova, A., Morgenstern, B., 2006b. AUGUSTUS at EGASP: Using EST, protein and genomic alignments for improved gene prediction in the human genome. *Genome Biol.* 7.
- Steijger, T., *et al.*, 2013. Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods* 10 (12), 1177–1184.
- Stein, L.D., *et al.*, 2002. The generic genome browser: A building block for a model organism system database. *Genome Res.* 12 (10), 1599–1610.
- Struhl, K., 2007. Transcriptional noise and the fidelity of initiation by RNA polymerase II. *Nat. Struct. Mol. Biol.* 14 (2), 103–105.
- Su, M., *et al.*, 2013. Small proteins: Untapped area of potential biological importance. *Front. Genet.* 4, 286.
- Tatusova, T., *et al.*, 2016. NCBI prokaryotic genome annotation pipeline. *Nucleic Acids Res.* 44 (14), 6614–6624.
- The Gene Ontology Consortium., 2017. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res.* 45 (D1), D331–D338.
- Thibaud-Nissen, F., *et al.*, 2016. The NCBI eukaryotic genome annotation pipeline. *J. Anim. Sci.* 94, 184.
- Thorvaldsdottir, H., Robinson, J.T., Mesirov, J.P., 2013. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Brief. Bioinform.* 14 (2), 178–192.
- Wheeler, S.J., Church, D.M., Ostell, J.M., 2001. Spidey: A tool for mRNA-to-genomic alignments. *Genome Res.* 11 (11), 1952–1957.
- Wiberg, R.A.W., *et al.*, 2015. Assessing recent selection and functionality at long noncoding RNA loci in the mouse genome. *Genome Biol. Evol.* 7 (8), 2432–2444.
- Wiehe, T., *et al.*, 2001. SGP-1: Prediction and validation of homologous genes based on sequence alignments. *Genome Res.* 11 (9), 1574–1583.
- Wu, J.Q., *et al.*, 2004. Identification of rat genes by TWINSKAN gene prediction, RT-PCR, and direct sequencing. *Genome Res.* 14 (4), 665–671.
- Xu, Y., Mural, R.J., Uberbacher, E.C., 1997. Inferring gene structures in genomic sequences using pattern recognition and expressed sequence tags. In: *Proceedings Ismb-97 – Fifth International Conference on Intelligent Systems for Molecular Biology*, pp. 344–353.
- Yeh, R.F., Lim, L.P., Burge, C.B., 2001. Computational inference of homologous gene structures in the human genome. *Genome Res.* 11 (5), 803–816.
- Zhang, Y., *et al.*, 2005. Pyrrolysine and selenocysteine use dissimilar decoding strategies. *J. Biol. Chem.* 280 (21), 20740–20751.
- Zheng, D., *et al.*, 2007. Pseudogenes in the ENCODE regions: Consensus annotation, analysis of transcription, and evolution. *Genome Res.* 17 (6), 839–851.
- Zinoni, F., *et al.*, 1986. Nucleotide sequence and expression of the selenocysteine-containing polypeptide of formate dehydrogenase (formate-hydrogen-lyase-linked) from *Escherichia coli*. *Proc. Natl. Acad. Sci. USA* 83 (13), 4650–4654.

Biographical Sketch



Josep F. Abril, PhD Associate Professor at the Department of Genetics, Microbiology and Statistics of the Universitat de Barcelona. He earned his Bachelor's degree in Biology from the Universitat de Barcelona, and his PhD in Bioinformatics from the Universitat Pompeu Fabra in Barcelona. His research focuses on the computational analysis of sequences and their annotated features. He has worked on a variety of genomic and transcriptomic projects, but also on the functional characterization of proteins, and on different areas in computational biology. These include the development of protocols for the assembly and annotation of genomes, the use of comparative genomics methods, the analysis of gene expression, the modeling of splice sites and exonic structure of eukaryotic genes, the visualization of whole-genome annotations – this includes the human genome map published in *Science* in 2001 – the integration of expression and variation data into interaction networks, as well as the characterization of viral samples from metagenomic experiments. His organisms of interest are planarians, flies, and humans, not necessarily in that order, although he contributed to the annotation of other species too. Finally, he has been involved in the organization and analysis of three of the most relevant gene-prediction accuracy assessments in computational gene prediction, namely GASP in the *Drosophila* genome and EGASP and RGASP in the human one.



Sergi Castellano, PhD Associate Professor at the Genetics and Genomics Medicine Programme at University College London. He received his Bachelor's degree in Biology from the Universitat de Barcelona, and his PhD in Bioinformatics from the Universitat Pompeu Fabra, also in Barcelona. His group works on understanding the role of essential micronutrients, with particular emphasis on selenium, in the adaptation of human metabolism to the different environments encountered by archaic and modern humans. Much of his early work focused on the identification and characterization of functional elements in eukaryotic genomes, primarily selenium-containing genes, which are missannotated in standard databases. At the University of Hawaii, he developed the first database, SELENODB, with correct annotations for selenium-containing genes across animal genomes. Later, while at Cornell University and Howard Hughes Medical Institute, he used population genetics and molecular evolution approaches to study the evolution of the use of selenocysteine, the selenium-containing amino acid, compared to cysteine in proteins. More recently, at Max Planck in Germany, his group contributed to the population genetics analysis of Neandertals as it relates to their coding variation and their first encounters – as early as 100,000 years ago – with modern humans. His group is currently working on the role of micronutrient deficiencies in common and rare disease.

Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?

Emmanuelle Lerat, Université Lyon, CNRS, Villeurbanne, France

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

Bp	Base pairs	NGS	Next Generation Sequencing
ChIP	Chromatin Immuno Precipitation	SINE	Short Interspersed Nuclear Elements
Kb	Kilo base pairs	SSA	Sequence self-alignment
LINE	Long Interspersed Nuclear Elements	SSR	Simple Sequence Repeat
MITes	Miniature Inverted-repeat Transposable Elements	STR	Short Tandem Repeats
		TEs	Transposable Elements
		TIR	Terminal inverted repeat

Introduction

Repeats: A Large Bestiary Grouping Very Different Sequences

Although the genome size has been shown in prokaryotes to largely correlate to its complexity, i.e. the number of genes, it is not the case in eukaryotes. This observation, often referred to as the C-value paradox, can be partly explained by the fact that coding genes in eukaryotic genomes may represent only a tiny fraction of the total genome (Elliott and Gregory, 2015). In fact, it has been shown that the proportion of non-coding sequences in genomes may be particularly huge. For example, the coding genes in the human genome only represent 2% (Lander et al., 2001). A large proportion of the non-genic sequences are represented by repeats. In a genome, repeats correspond to non-coding sequences present in several occurrences. Two main categories of repeats are described, the tandem repeats and the interspersed repeats (Fig. 1). Another type of repeats exists in a genome which correspond to segmental duplications, which are very large nearly identical sequences (from 1 to 400 kb) often referred to as “low-copy repeats” (Sharp et al., 2005).

The tandem repeats correspond to sequences having a size from few to hundred of base pairs (bp) occurring in tandem and over several hundred of kilo base pairs (kb), which characterized them as “highly repeated sequences”. We can distinguish among this category the Simple Sequence Repeats (SSR) also called Short Tandem Repeats (STR) or microsatellites, which are a short tract of adjacent DNA motifs (around 1 to 13 bp) and the minisatellites which are longer (around 14 to 500 bp), both types being repeated several times (from 5 to 50 times for example). This type of repeats usually occurs at particular regions of the chromosomes corresponding to the telomeres and centromeres, but they also can be found throughout the genome, especially in regulatory and coding regions of genes (Richard et al., 2008).

The interspersed repeats stand for sequences that are more often known under the name transposable elements (TEs). These particular sequences were discovered during the 1950s by Barbara McClintock (1950). TEs have the particularity to be able to move from one position to another along the chromosomes since the majority of them encode all the proteins necessary for their

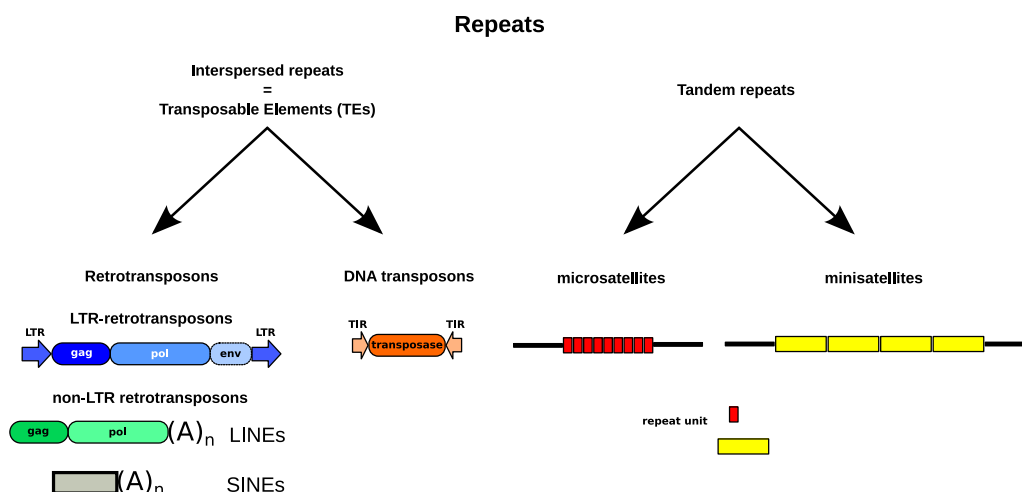


Fig. 1 Schematic representation of the different types of repeats.

transposition. Various types exist, according to structural features, the transposition intermediate (RNA or DNA), and their evolutionary origin (Wicker *et al.*, 2007; Kapitonov and Jurka, 2008). Retrotransposons use an RNA intermediate and form the class I, in which are found the Long Terminal Repeat (LTR)-retrotransposons (endogenous retrovirus-like mobile elements) which possess from two to three open reading frames (gag, pol, and env) and the non-LTR retrotransposons grouping the LINE and the SINE elements (standing for Long and Short Interspersed Nuclear Elements respectively). DNA transposons use a DNA intermediate and form the class II. They possess short terminal inverted repeats (TIRs) at their extremities surrounding one open reading frame coding for a transposase. Different types of DNA transposons have been classified mainly on the basis of the presence or absence of a catalytic site in the protein responsible for their transposition. A specific category of non-autonomous elements among the DNA transposons exist, the Miniature Inverted-repeat Transposable Elements (MITEs) that may be particularly numerous in some genomes. Depending on the organism, the proportion of TEs can be highly variable and at times very large. For example, their proportion in genomes represent 3% in yeast (Kim *et al.*, 1998), 15% in *Drosophila* (Dowsett and Young, 1982), 45% in human and in the mouse (Lander *et al.*, 2001; Waterston *et al.*, 2002), and more than 80% in maize (Schnable *et al.*, 2009).

“From Junk to Funk”: The Importance of Repeats in the Genome Functioning and Its Evolution

Repeats, and more particularly TEs, have long been considered as selfish and unnecessary components of the genomes. However, since the work of Britten and Davidson (1969) where TEs were for the first time considered as potentially playing a role in the gene regulation, numerous examples have flourished allowing to show the genuine importance of such sequences in genomes (Biemont, 2010). By their ability to move and because they are repeated, TEs can promote various types of mutations, which are expected to be mostly deleterious when affecting functional regions. When TE insertions occur in or near protein-coding genes, they can result in coding sequence modification or alteration of their splicing or polyadenylation patterns, therefore disrupting the protein coding capacity of the gene. Moreover, because TEs possess their own regulatory sequences, they can alter the normal expression pattern of neighboring genes while inserted in an intergenic region (Kidwell and Lisch, 2000; Biémont and Vieira, 2006). The possibility of homologous recombination between copies can also promote illegitimate recombinations, chromosome breakages, deletions and genome rearrangements (Kidwell and Lisch, 2000; Biémont and Vieira, 2006). In human, 0.3% of TE insertions have been suggested for causing disease (Belancio *et al.*, 2008) and approximately 96 new transposition events were directly linked to single-gene diseases (Hancks and Kazazian, 2012). For example, the *Alport* syndrome has been shown to be due to a TE mediated rearrangement resulting in the partial deletion and fusion of two genes (Segal *et al.*, 1999). More specific to cancer, the disruption of the APC gene caused by the TE insertion is involved in a colon cancer (Miki *et al.*, 1992). Despite the deleterious effects they may have, TEs have also been associated with useful adaptation for their host genome. For example, the antigen receptor gene assembly by V(D)J recombination in vertebrates is performed by genes that originated from a DNA transposon (Agrawal *et al.*, 1998). TE insertions near specific genes confer resistance to insecticide for some insects (Rostant *et al.*, 2012). These examples make TEs to be now considered as major players in genome evolution due to the genetic and epigenetic diversity they can promote (Biémont and Vieira, 2006). Similarly, tandem repeats have been found to play a fundamental role in the organization of the genome (Dumbovic *et al.*, 2017). For example, changes in mini- and micro-satellites correlate with various diseases but are also associated with gene regulation (Dumbovic *et al.*, 2017).

In addition to the fundamental biological role repeats have in genomes, in the current big sequencing era, they also represent a technical challenge that may complicate the task of genome assembly and sequence alignment. For example, the presence of repeats in a genome is the major source of genome mis-assemblies via rearrangement assembly errors and collapsed repeats but also in the assignment of splicing events and gene expression estimate in transcriptome analyses (Treangen and Salzberg, 2011). Since simply ignoring these sequences is not an issue in genomics, it is important to be able to identify them.

Bioinformatic Approaches to Identify, Annotate and Analyze Repeats in Genomes

Since several decades, numerous bioinformatics tools have been developed to allow a better identifications of repeats in genome assemblies (Lerat, 2010; Modolo and Lerat, 2013; Saha *et al.*, 2008; Bergman and Quesneville, 2007). New tools continue to arise regularly to follow the progress in sequencing technology in particular, but also to response to specific biological questions. According to the type of data on which they can work and the biological question, the different tools can be separated into various categories (Table 1).

Detection of Repeats in Assembled Genomes

Diverse methods exist to allow the detection and annotation of repeats in assembled genomes. These methods depend on the type of repeats and on the knowledge we have concerning their content inside the organism under investigation. Repeat annotation is a particularly complex computing problem due to the nature of the sequences, especially in the case of the TEs. Indeed, the TEs are not always exact repeats since a large divergence among copies can occur and TEs also can be found in the genome inserted inside each others (nested insertions). The detection of TEs has led to the development of a large number of different tools that fall in different categories according to the approach they use. Two main types of approaches exist: the library- or signature-based methods and the *ab initio* methods.

Table 1 Non-exhaustive list of tools to identify and analyze repeats

Program name	Type of repeats	Input data	Approach	References	Websites
Tandem Repeat Finder	Tandem repeat	Assembled genome	Identification	Benson (1999)	https://tandem.bu.edu/trf/trf.html
XSTREAM	Tandem repeat	Assembled genome	Identification	Newman and Cooper (2007)	http://jimcooperlab.mcdm.ucsb.edu/xstream/
MREPS	Tandem repeat	Assembled genome	Identification	Kolpakov <i>et al.</i> (2003)	http://mreps.univ-mlv.fr/
STAR	Tandem repeat	Assembled genome	Identification	Delgrange and Rivals (2004)	http://www.atgc-montpellier.fr/star/
TRED	Tandem repeat	Assembled genome	Identification	Sokol <i>et al.</i> (2007)	http://tandem.sci.brooklyn.cuny.edu/tandem/
ReD Tandem	Tandem repeat	Assembled genome	Identification	Audemard <i>et al.</i> (2012)	NA
RepeatMasker	TEs, tandem repeat	Assembled genome	Identification	Smit <i>et al.</i> (1996–2010)	http://www.repeatmasker.org/
LTR_STRUC	LTR-retrotransposons	Assembled genome	Identification	McCarthy and McDonald (2003)	http://www.mcdonaldlab.biology.gatech.edu/ltr_struct.htm
LTR_FINDER	LTR-retrotransposons	Assembled genome	Identification	Xu and Wang (2007)	http://liffe.fudan.edu.cn/ltr_finder/
find_LTR	LTR-retrotransposons	Assembled genome	Identification	Rho <i>et al.</i> (2007)	http://darwin.informatics.indiana.edu/cgi-bin/evolution/ltr.pl
LTR_harvest	LTR-retrotransposons	Assembled genome	Identification	Ellinghaus <i>et al.</i> (2008)	http://www.zbh.uni-hamburg.de/forschung/arbeitsgruppe-genom-informatik/software/ltrharvest.html
TSDfinder	Non-LTR retrotransposons	Assembled genome	Identification	Szak <i>et al.</i> (2002)	https://www.ncbi.nlm.nih.gov/CBBResearch/Landsman/TSDfinder/
SINEDR	Non-LTR retrotransposons	Assembled genome	Identification	Tu <i>et al.</i> (2004)	NA
TRANSPPO	MITes	Assembled genome	Identification	Santiago <i>et al.</i> (2002)	NA
MUST / MUSTv2	MITes	Assembled genome	Identification	Chen <i>et al.</i> (2009); Ge <i>et al.</i> (2017)	http://www.healthinformatics.org/supp/resources.php
MITE-hunter	MITes	Assembled genome	Identification	Han and Wessler (2010)	http://target.plantcollaborative.org/
One code to find them all	TEs, tandem repeat	Output files from RepeatMasker	Analysis	Bailly-Bechet <i>et al.</i> (2014)	http://doua.prabi.fr/software/one-code-to-find-them-all
LTRdigest	LTR-retrotransposons	GFF3 format	Analysis	Steinbiss <i>et al.</i> (2009)	http://www.zbh.uni-hamburg.de/forschung/arbeitsgruppe-genom-informatik/software/ltrdigest.html
RECON	TEs, tandem repeat	Assembled genome	Identification	Bao and Eddy (2003)	http://eddylib.org/software/recon/
PILER	TEs, tandem repeat	Assembled genome	Identification	Edgar and Myers (2005)	https://www.drive5.com/piler/
ReAS	TEs, tandem repeat	Assembled genome	Identification	Li <i>et al.</i> (2005)	Freely available via ReAS@genomics.org.cn
RepeatScout	TEs, tandem repeat	Assembled genome	Identification	Price <i>et al.</i> (2005)	https://bix.ucsd.edu/repeatscout/
TEclass	TEs	Individual sequences	Classification	Abrusan <i>et al.</i> (2009)	http://www.mybiosoftware.com/teclass-2-1-classification-te-consensus-sequences.html
REPCLASS	TEs	Individual sequences	Classification	Feschotte <i>et al.</i> (2010)	https://github.com/feschotte/REPCLASS
PASTECC	TEs	Individual sequences	Classification	Hoede <i>et al.</i> (2014)	https://urgi.versailles.inra.fr/Tools/PASTEClassifier
REPET	TEs, tandem repeat	Assembled genome	Pipeline (Identification and classification)	Flutre <i>et al.</i> (2011)	https://urgi.versailles.inra.fr/Tools/REPET
RepeatModeler	TEs, tandem repeat	Assembled genome	Pipeline (Identification and classification)	Smit and Hubley, unpublished	http://www.repeatmasker.org/RepeatModeler/
AAARF	TEs, tandem repeat	Short reads	Identification	DeBarry <i>et al.</i> (2008)	https://sourceforge.net/projects/aaarf
Tedna	TEs, tandem repeat	Short reads	Identification	Zytnicki <i>et al.</i> (2014)	https://urgi.versailles.inra.fr/Tools/Tedna

RepeatExplorer (SeqGraphR)	TEs, tandem repeat	Short reads	Identification	Novák <i>et al.</i> (2010); Novák <i>et al.</i> (2013)	http://www.repeatexplorer.org/
DNApipeTE	TEs, tandem repeat	Short reads	Identification	Goubert <i>et al.</i> (2015)	http://w3lamc.umbr.cas.cz/lamc/?page_id=301
RepARK	TEs, tandem repeat	Short reads	Identification	Koch <i>et al.</i> (2014)	https://lbbbe.univ-lyon1.fr/-dnaPipeTE-.html
REPdenovo	TEs, tandem repeat	Short reads	Identification	Chu <i>et al.</i> (2016)	https://github.com/PhKoch/RepARK
Transposome	TEs, tandem repeat	Short reads	Identification	Staton and Burke (2015)	https://github.com/Reedwarbler/REPdenovo
VNTRseek	Tandem repeat	Short reads	Identification	Gelfand <i>et al.</i> (2014)	https://github.com/yzhernand/VNTRseek
TAREAN	Tandem repeat	Short reads	Identification	Novák <i>et al.</i> (2017)	http://w3lamc.umbr.cas.cz/lamc/?page_id=312
TE-locate	TEs	Short reads	TE insertion variants	Platzer <i>et al.</i> (2012)	https://sourceforge.net/projects/te-locate/
TraFIC	TEs	Short reads	TE insertion variants	Tubio <i>et al.</i> (2014)	https://gitlab.com/mobilegenomes/TraFIC
TE-Tracker	TEs	Short reads	TE insertion variants	Gilly <i>et al.</i> (2014)	http://www.genoscope.cns.fr/externe/tetracker/
RelocaTE	TEs	Short reads	TE insertion variants	Robb <i>et al.</i> (2013)	https://github.com/srobb1/RelocaTE
Ngs-TE-mapper	TEs	Short reads	TE insertion variants	Linheiro and Bergman (2012)	https://github.com/bergmanlab/ngs_te_mapper
TIDAL	TEs	Short reads	TE insertion variants	Rahman <i>et al.</i> (2015)	https://github.com/lalabrandeis/TIDAL
ITIS	TEs	Short reads	TE insertion variants	Jiang <i>et al.</i> (2015)	http://bioinformatics.psc.ac.cn/software/ITIS/
T-Lex2	TEs	Short reads	TE insertion variants	Fiston-Lavier <i>et al.</i> (2015)	http://petrov.stanford.edu/cgi-bin/Tlex.html
Tea	TEs	Short reads	TE insertion variants	Lee <i>et al.</i> (2012)	http://compbio.med.harvard.edu/Tea/
RetroSeq	TEs	Short reads	TE insertion variants	Keane <i>et al.</i> (2013)	https://github.com/wtsi-sv/RetroSeq
TEMP	TEs	Short reads	TE insertion variants	Zhuang <i>et al.</i> (2014)	https://github.com/JialiUMassWengLab/TEMP
Mobster	TEs	Short reads	TE insertion variants	Thung and <i>et al.</i> (2014)	https://sourceforge.net/projects/mobster/
Tangram	TEs	Short reads	TE insertion variants	Wu <i>et al.</i> (2014)	https://github.com/jiantao/Tangram/issues
TranspoSeq	TEs	Short reads	TE insertion variants	Helman <i>et al.</i> (2015)	http://archive.broadinstitute.org/cancer/cga/transposseq
Jitterbug	TEs	Short reads	TE insertion variants	Hénaff <i>et al.</i> (2015)	https://github.com/elzbt/jitterbug
DD-Detection	TEs	Short reads	TE insertion variants	Kroon <i>et al.</i> (2016)	https://bitbucket.org/mkroon/dd_detection
popoolation_TE2	TEs	Short reads	TE insertion variants	Kofler <i>et al.</i> (2016)	https://sourceforge.net/p/popoolation-te2/wiki/Home/
MELT	TEs	Short reads	TE insertion variants	Gardner <i>et al.</i> (2017)	http://melt.igs.umaryland.edu/manual.php
LorTE	TEs	Long reads	TE insertion variants	Disdero and Filée (2017)	http://www.egec.cnrs-gif.fr/?p=6422
Repeat Histone enrichment	TEs	ChipSeq data	Histone enrichment	Day <i>et al.</i> (2010)	http://compbio.med.harvard.edu/repeats/
piPipes	TEs	RNAseq data	Differential expression analyses	Han <i>et al.</i> (2015)	https://github.com/bowhan/piPipes
TEtools	TEs	RNAseq data	Differential expression analyses	Lerat <i>et al.</i> (2017)	https://github.com/l-modolo/TEtools
EpiTEome	TEs	Short reads	TE insertion variants and DNA methylation analysis	Daron and Slotkin (2017)	https://github.com/jdaron/epiTEome

The library- or signature-based methods all require a certain amount of knowledge concerning the searched TEs. Library-based methods compare the genome sequence to a set of reference sequences corresponding to TE consensus (i.e. library) to search for their occurrences in the genome. It is thus an approach by sequence homology. The most known and used program from this category is *RepeatMasker* (Smit *et al.*, 1996–2010) whose search engines include *nhmmer*, *cross_match*, *ABBLAST/WUBLAST*, *RMBLAST* and *Decypher*. It relies on a library of consensus sequences of TEs called Repbase (Bao *et al.*, 2015). It is also able to identify low complexity DNA sequences, which can correspond to tandem repeats, by using sequence homology and the *Tandem Repeat Finder* program (Benson, 1999). The main outputs of the program are a global annotation of the repeats that are present in the query sequence as well as a modified version of the query sequence in which all the annotated repeats have been masked. Recently, a program has been developed to allow a more detailed analyses of one output as well as an easy way to retrieve the identified sequences (Bailly-Bechet *et al.*, 2014). Although *RepeatMasker* is fast and quite efficient, the main drawback consists in the need to already know the sequences of the TEs that are present in the genome under investigation or at least in not too far closely related ones. It is thus not possible by this approach to detect new TE families. The signature-based methods allow to have less knowledge concerning the searched TEs since they seek the genome sequence for nucleotidic or proteic motifs, or particular structural features from a specific class of TEs. LTR-retrotransposons can be identified based on their structure (presence of two direct repeats (LTR) at their extremities, size, presence of protein motifs inside the coding parts etc.) by different programs like *LTR_STRUC* (McCarthy and McDonald, 2003), *find_LTR* (Rho *et al.*, 2007), *LTR-finder* (Xu and Wang, 2007) and *LTRharvest* (Ellinghaus *et al.*, 2008). Outputs from *LTRharvest* can further be analyzed by *LTRdigest* (Steinbiss *et al.*, 2009) to annotate internal features of LTR-retrotransposons. These programs have various amount of success since they can find a lot of false positives, meaning that their output files need to be manually curated. They also are designed to find only full-length elements with two LTRs at both extremities of the element and sufficiently conserved. The identification of non-LTR retrotransposons has been the goal of some programs like *TSDfinder* (Szak *et al.*, 2002) and *SINEDR* (Tu *et al.*, 2004). DNA transposons can also be searched for using structural and sequence characteristics of such elements using programs like *TRANSPO* (Santiago *et al.*, 2002), *MUST* (Chen *et al.*, 2009), recently updated into a new version *MUSTv2* (Ge *et al.*, 2017), and *MITE-hunter* (Han and Wessler, 2010). All signature-based programs are thus mainly designed to help the researcher finding very specific types of TEs and thus concerning very specific biological questions since with these approaches, a large proportion of repeats remains ignored.

Alternatively to the precedent approaches, *ab initio* methods have been developed to detect virtually all kind of repeats in a genome without any *a priori* knowledge. These approaches can be separated into two distinct categories. In the first category, the methods first use self-comparison approaches of sequences to identify repeats and then use clustering methods to group them into families, before generating a consensus sequence for each detected family (method implemented in *RECON* (Bao and Eddy, 2003) or *PILER* (Edgar and Myers, 2005)). In the second category are grouped programs using k-mer and spaced seed approaches, which count “words” (method implemented in *ReAS* (Li *et al.*, 2005) or *RepeatScout* (Price *et al.*, 2005)). In that case, a repeat is defined as a subsequence that appears more than once in a longer sequence. All *ab-initio* programs have very varying rates of success in the identification of repeats depending on the data (quality of the genome assembly, proportion and age of the repeats in the genome). Moreover, the results produced by these methods are generally raw implying further analyses to identify the different type of repeats. To help with this step, some classification programs have been proposed like *TEclass* (Abrusán *et al.*, 2009), *REPCLASS* (Feschotte *et al.*, 2010), and *PASTEC* (Hoede *et al.*, 2014), which use structural features and sequence similarities to determine to which type of TE a sequence can be associated. The level of precision is variable according to the tool, *PASTEC* having currently the finer level in the assignation of TE type according to the classification proposed by Wicker and colleagues (Wicker *et al.*, 2007). Globally, no stand alone *ab initio* program can discover all repeated sequences present in a genome (Platt *et al.*, 2016). This is why approaches using several different programs to optimize repeat finding have been developed like, for example, the pipelines *REPET* (Flutre *et al.*, 2011) and *RepeatModeler* (Smit and Hubley; see Relevant Website section). They tend to give better results and have been largely used in the scientific community. For example, *RepeatModeler* has recently been used to identify TEs in the genome of a fungi (Castanera *et al.*, 2016) while this program was used in addition to *REPET* in the genome of the Atlantic salmon (Lien *et al.*, 2016).

A large number of tools also exist to detect specifically tandem repeats (see for a review Lim *et al.*, 2013). One of the most used tools to identify these sequences is the *Tandem Repeat Finder* program, which has been developed almost 20 years ago and that is still maintained by his developer (Benson, 1999). The algorithm of this program uses the approach of matching k-tuples, i.e. two windows of k consecutive characters from a nucleotide sequence that have identical content. This requires no *a priori* knowledge concerning the pattern of the repeat, its size or the number of copies. A similar approach is used in programs like *XSTREAM* (Newman and Cooper, 2007) and *MREPS* (Kolpakov *et al.*, 2003). Other programs use improved dynamic programming algorithms like *STAR* (Delgrange and Rivals, 2004) and *TRED* (Sokol *et al.*, 2007). More recently, the program *ReD tandem* (Audemard *et al.*, 2012) was developed using a flow based chaining algorithm. Some programs, like any type of dot-plot programs, are based solely on sequence self-alignment (SSA) algorithms, which are particularly efficient for the detection of long repeats. Their drawbacks are that these programs are relatively slow due to their time complexity and usually fail to identify short repeats.

Detection of Repeats in Unassembled Genomes

The development of the next generation sequencing (NGS) technologies has overturned our approach to genomics with a huge amount of data being produced everyday (Margulies *et al.*, 2005; Mardis, 2017). We thus now have access to more data on very various organisms at low cost and with fewer bias than the previous sequencing technologies (Wicker *et al.*, 2006). Currently however, the data produced by regular NGS technology like Illumina, correspond to rather short sequences due to the small size of the reads (maximum of 300 bp length) (Mardis, 2017). This short size implies that the assembly of the original DNA sequences is the most challenging and

time-consuming step especially when the considered organism is rich in repeats. To assemble a genome in this condition often leads to unfinished drafts of very numerous scaffolds, a large number of them corresponding to unplaced repeats on chromosomes.

By their repetitive nature, repeats represent portions of the genome with the best coverage, especially in the case of genome survey sequencing, where a sample of a complete genome is actually sequenced. Indeed, for a genomic coverage of 0.01X, each repeat having 1000 occurrences will theoretically have a coverage of 10X (Macas *et al.*, 2007). This situation has allowed the development of new approaches to detect repeats directly from the raw data, without the need for any homology search nor assembly.

The first method to have been developed based on this assumption and that is able to work with short reads is the AAARF algorithm (DeBarry *et al.*, 2008). This approach uses BLAST (Altschul *et al.*, 1997) to compare a read against all the others and obtain its nucleotide coverage. This value is used to determine overlapping reads that will be aligned to reconstruct a new sequence. Iteratively, the program will elongate each new sequence to assemble a set of TE contigs. Another type of approach that is also working on genomic sample is based on the construction of sequence clusters like the SeqGraphR program implemented in the RepeatExplorer pipeline (Novák *et al.*, 2010, 2013). In this method, reads are clustered using a hierarchical agglomeration algorithm. Various graph metrics are computed to discriminate between different types of repeats and the assembly of TE sequences gives consensus sequences. Based on a similar approach, the Transposome program has been proposed more recently that also uses a graph-based analysis of similarity between reads (Staton and Burke, 2015). As an alternative to read clustering, another approach, DNApipeTE (Goubert *et al.*, 2015), proposes to use the RNAseq assembler Trinity (Grabherr *et al.*, 2011) to build repeat contigs from genomic samples of less than 1X of coverage. The use of Trinity allows to recover alternative repeat consensus of a given family by producing distinct contigs for each structural variant. Alternatively, other tools like Tedna (Zytnicki *et al.*, 2014), RepARK (Koch *et al.*, 2014) and REPdenovo (Chu *et al.*, 2016) use directly a de Bruijn graph on the most represented k-mers to perform TE assembly.

Although the previous methods are supposed to be able to find any kind of repeats, they usually are best suited to discover TEs. They may be less powerful concerning tandem repeats. This is why some specific tools have been designed to specifically uncover tandem repeats from raw reads. The first tool, VNTRseek (Gelfand *et al.*, 2014), is a variant detection tool that compares reads in which tandem repeats have been detected using Tandem Repeat Finder (Benson, 1999) to a set of tandem repeats present in a reference genome. By this comparison, the variable number of tandem repeats is determined. A recent pipeline called TAREAN (Novák *et al.*, 2017) uses the principles of graph-based repeat clustering, as implemented in the RepeatExplorer pipeline, as well as tools facilitating the unsupervised identification and characterization of satellite repeats from raw reads. The reconstruction of the satellite sequences is based on k-mer decomposition and counting.

Globally, all these programs allow to determine the global proportion of repeats inside a genome, with sometime the possibility to estimate the copy number, as well as a catalog of the different types of repeats with the production of consensus sequences. However, since these programs work on raw reads, the information concerning the exact positions of these repeats is missing. Other approaches have thus been developed to specifically answer this question by comparing the copy number variation of repeats between two genomes.

Identification of the Repeat Copy Number Variation

When the first genomes were sequenced, it was considered that only one would be enough to understand the functioning and evolution of a given species. However, having only one genome is not enough to uncover the polymorphism existing among individuals. Particularly for repeats, some of which being able to move and replicate themselves in the genome, it is known that variations exist in term of copy number and insertion sites among natural populations (Petrov *et al.*, 2011; Boulesteix *et al.*, 2006). With the decrease in cost of sequencing, it is now possible to obtain data from several individuals of a given population and of several populations of a given species. This has opened the door to perform high throughput population genomics when using pooled sequencing data. Since it is not always possible to obtain good assembly with these data, new tools have been developed to specifically search for differences when compared to a reference genome. Thus, the goal of the bioinformatic tools developed for this purpose is to determine either one or the three types of TE insertions: insertions shared between the reference and the analyzed data (fixed insertions), and polymorphic insertions corresponding to insertions either absent from the analyzed data or absent from the reference genome (new insertions) (Fig. 2).

Several tools have been developed to determine the structural variation due to TEs in genomes and some attempts have been made to evaluate and review them (Ewing, 2015; Rishishwar *et al.*, 2016). The various methods have all in common in their process to first map the reads on a reference genome and/or on a set of annotated TE sequences before applying various filters and metrics to retain the informative ones. Then, two approaches exist that may be combined to analyze the results and to detect the presence/absence of a TE insertion. In the first approach, the program considers discordant read pairs, which are read pairs with one member matching uniquely on the reference genome sequence and the other matching on different copies from a TE family (Fig. 3(A)). This type of approach is used in the programs TE-locate (Platzer *et al.*, 2012), TraFiC (Tubio *et al.*, 2014) and TE-Tracker (Gilly *et al.*, 2014).

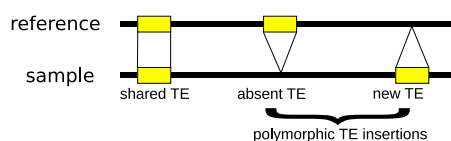


Fig. 2 The different types of TE insertions when comparing a reference genome and a newly sequenced one.

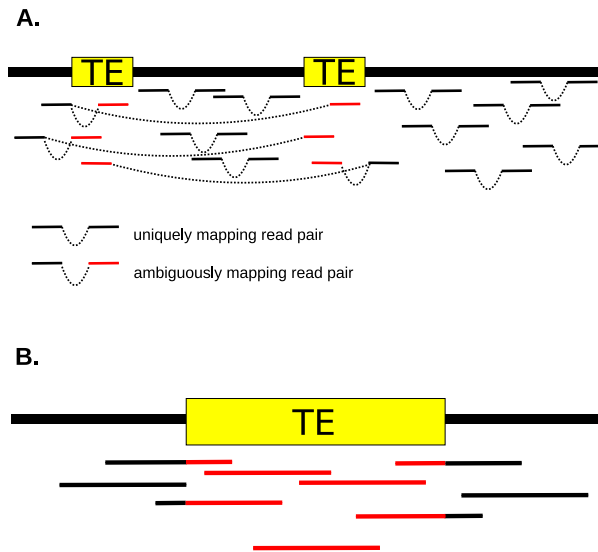


Fig. 3 Schematic representation of discordant read pairs (A) and split reads (B) that are used to identify TE insertion in raw data by comparison to a reference genome. Lines in black correspond to reads mapping on unique genomic regions and lines in red correspond to reads mapping on TE sequences.

The other approach consists in considering split reads, i.e., reads which overlap a junction between the genome and an inserted TE copies, with one part of the read mapping uniquely on the genome while the other part maps on TE sequences (**Fig. 3(B)**). The programs *RelocaTE* (Robb *et al.*, 2013), *ngs-TE-mapper* (Linheiro and Bergman, 2012), *TIDAL* (Rahman *et al.*, 2015), *ITIS* (Jiang *et al.*, 2015), and *T-Lex2* (Fiston-Lavier *et al.*, 2015) use this approach. Other programs use both approaches like *Tea* (Lee *et al.*, 2012), *RetroSeq* (Keane *et al.*, 2013), *TEMP* (Zhuang *et al.*, 2014), *Mobster* (Thung *et al.*, 2014), *Tangram* (Wu *et al.*, 2014), *TranspoSeq* (Helman *et al.*, 2014), *Jitterbug* (Hénaff *et al.*, 2015), *DD_Detection* (Kroon *et al.*, 2016), *popoolation_TE2* (Kofler *et al.*, 2016) and *MELT* (Gardner *et al.*, 2017). These different programs have been developed to answer specific biological questions, which make them either consider individual or population data to estimate the insertion frequency, to consider in majority polymorphic insertions (especially new insertions in the case of cancer research) but sometimes also shared insertions, and in some cases to take into account the genotype status of the detected insertions, i.e. if the insertion is present on only one (heterozygous) or two (homozygous) homologous chromosomes. For example, the program *T-Lex2* has been developed to compute the frequency insertions of TE copies that are present in the reference genome of *Drosophila melanogaster* in natural populations, whereas the program *Tea* has been developed to identify only new TE insertions in human cancers corresponding to somatic insertions.

Discussion

Repeats, and more particularly TEs, are important component of the eukaryote genomes that cannot be simply ignored when performing sequencing and assembly tasks. A lot of various bioinformatics tools have been developed during the last 20 years to allow a better handling of these particular sequences. Such tools need to evolve jointly with the evolution of sequencing technologies. Currently, one of the major problem with the current whole genome sequencing data produced to identify repeat insertions is the size of the available sequence reads (< 300 bp). Full-length TE insertions ranged between 500 bp to 10 kb, which implies that several reads are needed to cover an entire TE copy. This step may be particularly difficult since several insertions may present a very high degree of nucleotide identity between themselves. In that case, it is often impossible to recover some insertions and the only way to distinguish them is to take into account the genomic environment, i.e. the flanking regions around the insertion. However, sequence reads being shorter than the majority of TE insertions and reads overlapping the junction of the genomic region and the TE insertions being not numerous and difficult to map, this task is particularly challenging and lead to a loss of information. A way to tackle this difficulty is to produce longer reads. The third generation of DNA sequencing is currently under development and some already used techniques may be helpful, although still expensive or not optimized. For example, PacBio sequencing allows to produce sequences up to 20 kb but the rate of errors is still quite high (Mardis, 2017). However, this technique may be used in addition to short-read sequencing techniques to enhance the quality of a genome assembly (Pendleton *et al.*, 2015). The Illumina synthetic long-read technique has been successfully tested to perform the *de novo* assembly and resolve TE sequences in the genome of *Drosophila melanogaster* (McCoy *et al.*, 2014). However this technique is still expensive since the coverage that is needed is very high (Mardis, 2017). The MinION technology, which performs a single molecule sequencing, has been recently successfully used to detect new TE insertions in the plant genome *Arabidopsis thaliana* allowing to show that the high error rate of the technique could be compensated by the read length in this

particular question (Debladis *et al.*, 2017). Of course, as soon as these technologies will become the new standard, the current tools for TE analyses will become obsolete and new methodological procedures will need to be developed. In this way, a bioinformatic tool has recently been proposed to determine the presence/absence of TE insertions using reads produced by the PacBio technology (LorTE, Disdero and Filée, 2017).

Future Directions

The identification of TE insertions is also becoming very important in the field of epigenetics research. Indeed, TEs are known to be associated to particular epigenetic modifications that may impact the neighboring genes (Eichten *et al.*, 2012; Estécio *et al.*, 2012). Very few tools have been developed to allow the direct association between TEs and epigenetic modifications. For example, a web interface has been developed to specifically study histone modification enrichment of repeats taking advantage of their increased sequence coverage in ChIP-seq data (Day *et al.*, 2010). An advantage of this method is that it incorporates both ambiguously and uniquely mapped reads to avoid bias due to read mapping on consensus TE sequences. However, the results are not given at the insertion level and thus the information concerning the genomic environment of each insertion is lost. Several efforts have been made to develop methods allowing the association of small RNA data to TE sequences either at a global (*piPipes*, Han *et al.*, 2015) or at a individual copy level (*TEtools*, Lerat *et al.*, 2017). More recently, a new program has been developed allowing both the identification of new TE insertions and their associated DNA methylation in MethylC-seq data (*EpiTEome*, Daron and Slotkin, 2017). In summary, methodological efforts are still needed to study the epigenetic modifications directly associated to TE sequences. It is particularly important to consider the global sequence diversity of a TE family that may be very variable but also the genomic environment of a given insertion that may have consequences on the associated epigenetic modifications. These issues are currently difficult to handle due to the short size of sequence reads. However, as long read technologies will continue to be developed, it should open the door to new developments in a close future.

Closing Remarks

The domain of TE annotation in genome sequences is constantly producing new tools, which are supposed to outperform the previous ones and to offer new ways to handle the ever growing sequence data. However, the question of the impartial evaluation of these tools still remains. Indeed, the different tools have usually different competencies and may be at some point complementary. A clear problem underlying this situation is the lack of common standard data that would allow an unbiased estimation of any new tools (Hoen *et al.*, 2015). There is still room in that domain to propose various benchmarks according to the biological questions asked behind each tool.

See also: Functional Enrichment Analysis. Gene Mapping. Genome Annotation. Genome Annotation: Perspective From Bacterial Genomes. Genome Databases and Browsers. Genome Informatics. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Quantitative Immunology by Data Analysis Using Mathematical Models. Sequence Analysis. Sequence Composition. Whole Genome Sequencing Analysis

References

- Abrusán, G., Grundmann, N., DeMester, L., Makalowski, W., 2009. TEclass – A tool for automated classification of unknown eukaryotic transposable elements. *Bioinformatics* 25, 1329–1330.
- Agrawal, A., Eastman, Q.M., Schatz, D.G., 1998. Transposition mediated by RAG1 and RAG2 and its implications for the evolution of the immune system. *Nature* 394, 744–751.
- Altschul, S.F., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25, 3389–3402.
- Audemard, E., Schiex, T., Faraut, T., 2012. Detecting long tandem duplications in genomic sequences. *BMC Bioinformatics* 13, 83.
- Bailly-Bechet, M., Haudry, A., Lerat, E., 2014. “One code to find them all”: A perl tool to conveniently parse RepeatMasker output files. *Mobile DNA* 5, 13.
- Bao, W., Kojima, K.K., Kohany, O., 2015. Repbase update, a database of repetitive elements in eukaryotic genomes. *Mobile DNA* 6, 11.
- Bao, Z., Eddy, S.R., 2003. Automated *de novo* identification of repeat sequence families in sequenced genomes. *Genome Research* 13, 1269–1276.
- Belancio, V., Hedges, D.J., Deininger, P., 2008. Mammalian non-LTR retrotransposons: For better or worse, in sickness and in health. *Genome Research* 18, 343–358.
- Benson, G., 1999. Tandem repeats finder: A program to analyze DNA sequences. *Nucleic Acids Research* 27, 573–580.
- Bergman, C.M., Quesneville, H., 2007. Discovering and detecting transposable elements in genome sequences. *Briefings in Bioinformatics* 8, 382–392.
- Biemont, C., 2010. A brief history of the status of transposable elements: From junk DNA to major players in evolution. *Genetics* 186, 1085–1093.
- Biémont, C., Vieira, C., 2006. Genetics: Junk DNA as an evolutionary force. *Nature* 443, 521–524.
- Boulesteix, M., Simard, F., Antonio-Nkondjio, C., *et al.*, 2006. Insertion polymorphism of transposable elements and population structure of *Anopheles gambiae* M and S molecular forms in Cameroon. *Molecular Ecology* 16, 441–452.
- Britten, R.J., Davidson, E.H., 1969. Gene regulation for higher cells: A theory. *Science* 165, 349–357.
- Castanera, R., López-Varas, L., Borgognone, A., *et al.*, 2016. Transposable elements versus the fungal Genome: Impact on whole-genome architecture and transcriptional profiles. *PLOS Genetics* 12, e1006108.

- Chen, Y., Zhou, F., Li, G., Xu, Y., 2009. MUST: A system for identification of miniature inverted-repeat transposable elements and applications to *Anabaena variabilis* and *Haloquadratum walsbyi*. *Gene* 436, 1–7.
- Chu, C., Nielsen, R., Wu, Y., 2016. REPdenovo: Inferring *de novo* repeat motifs from short sequence reads. *PLOS ONE* 11, e0150719.
- Daron, J., Slotkin, R.K., 2017. EpiTEome: Simultaneous detection of transposable element insertion sites and their DNA methylation levels. *Genome Biology* 18, 91.
- Day, D.S., Luquette, L.J., Park, P.J., Kharchenko, P.V., 2010. Estimating enrichment of repetitive elements from high-throughput sequence data. *Genome Biology* 11, R69.
- DeBarry, J.D., Liu, R., Bennetzen, J.L., 2008. Discovery and assembly of repeat family pseudomolecules from sparse genomic sequence data using the Assisted Automated Assembler of Repeat Families (AAARF) algorithm. *BMC Bioinformatics* 9, 235.
- Debladis, E., Llauro, C., Carpentier, M.C., Mirouze, M., Panaud, O., 2017. Detection of active transposable elements in *Arabidopsis thaliana* using Oxford Nanopore Sequencing technology. *BMC Genomics* 18, 537.
- Delgrange, O., Rivals, E., 2004. STAR: An algorithm to search for tandem approximate repeats. *Bioinformatics* 20, 2812–2820.
- Disdero, E., Filée, J., 2017. LoRTE: Detecting transposon-induced genomic variants using low coverage PacBio long read sequences. *Mobile DNA* 8, 5.
- Dowsett, A., Young, M.W., 1982. Differing levels of dispersed repetitive DNA among closely related species of *Drosophila*. *Proceedings of the National Academy of Sciences of the United States of America* 79, 4570–4574.
- Dumbovic, G., Forcales, S.-V., Perucho, M., 2017. Emerging roles of macrosatellite repeats in genome organization and disease development. *Epigenetics* 12, 515–526.
- Edgar, R.C., Myers, E.W., 2005. PILER: Identification and classification of genomic repeats. *Bioinformatics* 21 (Suppl 1), 152–158.
- Eichten, S.R., Ellis, N.A., Makarevitch, I., et al., 2012. Spreading of heterochromatin is limited to specific families of maize retrotransposons. *PLOS Genetics* 8, e1003127.
- Ellinghaus, D., Kurtz, S., Willhoeft, U., 2008. LTRharvest, an efficient and flexible software for *de novo* detection of LTR retrotransposons. *BMC Bioinformatics* 9, 18.
- Elliott, T.A., Gregory, T.R., 2015. What's in a genome? The C-value enigma and the evolution of eukaryotic genome content. *Philosophical transactions of the Royal Society of London. Series B: Biological Sciences* 370, 20140331.
- Estécio, M.R., Gallegos, J., Dekmezian, M., et al., 2012. SINE retrotransposons cause epigenetic reprogramming of adjacent gene promoters. *Molecular Cancer Research* 10, 1332–1342.
- Ewing, A.D., 2015. Transposable element detection from whole genome sequence data. *Mobile DNA* 6, 24.
- Feschotte, C., Keswani, U., Ranganathan, N., Guibotsy, M.L., Levine, D., 2010. Exploring repetitive DNA landscapes using REPCLASS, a tool that automates the classification of transposable elements in eukaryotic genomes. *Genome Biology and Evolution* 1, 205–220.
- Fiston-Lavier, A.S., Barrón, M.G., Petrov, D.A., González, J., 2015. T-lex2: Genotyping, frequency estimation and re-annotation of transposable elements using single or pooled next-generation sequencing data. *Nucleic Acids Research* 43, e22.
- Flutre, T., Duprat, E., Feuillet, C., Quesneville, H., 2011. Considering transposable element diversification in *de novo* annotation approaches. *PLOS ONE* 6, e16526.
- Gardner, E.J., Lam, V.K., Harris, D.N., et al., 2017. The Mobile Element Locator Tool (MELT): Population-scale mobile element discovery and biology. *Genome Research* 218032, 116.
- Gelfand, Y., Hernandez, Y., Lovings, J., Benson, G., 2014. VNTRseek – A computational tool to detect tandem repeat variants in high-throughput sequencing data. *Nucleic Acids Research* 42, 8884–8894.
- Ge, R., Mai, G., Zhang, R., et al., 2017. MUSTv2: An improved *de novo* detection program for recently active Miniature Inverted repeat Transposable Elements (MITEs). *Journal of Integrative Bioinformatics* 14.
- Gilly, A., Etcheverry, M., Madoui, M.A., et al., 2014. TE-Tracker: Systematic identification of transposition events through whole-genome resequencing. *BMC Bioinformatics* 15, 377.
- Goubert, C., Modolo, L., Vieira, C., et al., 2015. *De novo* assembly and annotation of the Asian tiger mosquito (*Aedes albopictus*) repeatome with dnaPipeTE from raw genomic reads and comparative analysis with the yellow fever mosquito (*Aedes aegypti*). *Genome Biology and Evolution* 7, 1192–1205.
- Grabherr, M.G., Haas, B.J., Yassour, M., et al., 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* 29, 644–652.
- Han, B.W., Wang, W., Zamore, P.D., Weng, Z., 2015. piPipes: A set of pipelines for piRNA and transposon analysis via small RNA-seq, RNA-seq, degradome- and CAGE-seq, ChIP-seq and genomic DNA sequencing. *Bioinformatics* 31, 593–595.
- Han, Y., Wessler, S.R., 2010. MITE-Hunter: A program for discovering miniature inverted-repeat transposable elements from genomic sequences. *Nucleic Acids Research* 38, 1–8.
- Hancks, D.C., Kazazian, H.H., 2012. Active human retrotransposons: Variation and disease. *Current Opinion in Genetics and Development* 22, 191–203.
- Helman, E., Lawrence, M.S., Stewart, C., et al., 2014. Somatic retrotransposition in human cancer revealed by whole-genome and exome sequencing. *Genome Research* 24, 1053–1063.
- Hénaff, E., Zapata, L., Casacuberta, J.M., Ossowski, S., 2015. Jitterbug: Somatic and germline transposon insertion detection at single-nucleotide resolution. *BMC Genomics* 16, 768.
- Hoede, C., Arnoux, S., Moisset, M., et al., 2014. PASTEC: An automatic transposable element classification tool. *PLOS ONE* 9, e91929.
- Hoehn, D.R., Hickey, G., Bourque, G., et al., 2015. A call for benchmarking transposable element annotation methods. *Mobile DNA* 6, 13.
- Jiang, C., Chen, C., Huang, Z., Liu, R., Verdier, J., 2015. ITIS, a bioinformatics tool for accurate identification of transposon insertion sites using next-generation sequencing data. *BMC Bioinformatics* 16, 72.
- Kapitonov, V.V., Jurka, J., 2008. A universal classification of eukaryotic transposable elements implemented in Repbase. *Nature reviews Genetics* 9, 411–412.
- Keane, T.M., Wong, K., Adams, D.J., 2013. RetroSeq: Transposable element discovery from next-generation sequencing data. *Bioinformatics* 29, 389–390.
- Kidwell, M.G., Lisch, D.R., 2000. Transposable elements and host genome evolution. *Trends in Ecology and Evolution* 15, 95–99.
- Kim, J.M., Vanguri, S., Boeke, J.D., Gabriel, A., Voytas, D.F., 1998. Transposable elements and genome organization: A comprehensive survey of retrotransposons revealed by the complete *Saccharomyces cerevisiae* genome sequence. *Genome Research* 8, 464–478.
- Koch, P., Platzer, M., Downie, B.R., 2014. RepARK – *De novo* creation of repeat libraries from whole-genome NGS reads. *Nucleic Acids Research* 42, 1–12.
- Kofler, R., Gómez-Sánchez, D., Schlötterer, C., 2016. PoPoolationTE2: Comparative population genomics of transposable elements using Pool-Seq. *Molecular Biology and Evolution* 33, 2759–2764.
- Kolpakov, R., Bana, G., Kucherov, G., 2003. mreps: Efficient and flexible detection of tandem repeats in DNA. *Nucleic Acids Research* 31, 3672–3678.
- Kroon, M., Lameijer, E.W., Lakenberg, N., et al., 2016. Detecting dispersed duplications in high-throughput sequencing data using a database-free approach. *Bioinformatics* 32, 505–510.
- Lander, E.S., et al., 2001. Initial sequencing and analysis of the human genome. *Nature* 409, 860–921.
- Lee, S., Iskow, R., Yang, L., et al., 2012. Landscape of somatic retrotransposition in human cancers. *Science* 337, 967–971.
- Lerat, E., 2010. Identifying repeats and transposable elements in sequenced genomes: How to find your way through the dense forest of programs. *Heredity* 104, 520–533.
- Lerat, E., Fablet, M., Modolo, L., Lopez-Maestre, H., Vieira, C., 2017. TEtools facilitates big data expression analysis of transposable elements and reveals an antagonism between their activity and that of piRNA genes. *Nucleic Acids Research* 45, e17.
- Lien, S., Koop, B.F., Sandve, S.R., et al., 2016. The Atlantic salmon genome provides insights into rediploidization. *Nature* 533, 200–205.
- Lim, K.G., Kwok, C.K., Hsu, L.Y., Wirawan, A., 2013. Review of tandem repeat search tools: A systematic approach to evaluating algorithmic performance. *Briefings in Bioinformatics* 14, 67–81.
- Linheiro, R.S., Bergman, C.M., 2012. Whole genome resequencing reveals natural target site preferences of transposable elements in *Drosophila melanogaster*. *PLOS ONE* 7, e30008.

- Li, R., Ye, J., Li, S., *et al.*, 2005. ReAS: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. *PLOS Computational Biology* 1, 313–321.
- Macas, J., Neumann, P., Navrátilová, A., 2007. Repetitive DNA in the pea (*Pisum sativum* L.) genome: Comprehensive characterization using 454 sequencing and comparison to soybean and *Medicago truncatula*. *BMC Genomics* 8, 427.
- Mardis, E.R., 2017. DNA sequencing technologies: 2006–2016. *Nature Protocols* 12, 213–218.
- Margulies, M., Egholm, M., Altman, W.E., *et al.*, 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437, 376–380.
- McCarthy, E.M., McDonald, J.F., 2003. LTR_STRUC: A novel search and identification program for LTR retrotransposons. *Bioinformatics* 19, 362–367.
- McClintock, B., 1950. The origin and behavior of mutable loci in maize. *Proceedings of the National Academy of Sciences of the United States of America* 36, 344–355.
- McCoy, R.C., Taylor, R.W., Blauwkamp, T.A., *et al.*, 2014. Illumina TruSeq Synthetic Long-Reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. *PLOS ONE* 9, e106689.
- Miki, Y., Nishisho, I., Horii, A., *et al.*, 1992. Disruption of the APC gene by a retrotransposal insertion of L1 sequence in a colon cancer. *Cancer Research* 52, 643–645.
- Modolo, L., Lerat, E., 2013. Identification and analysis of transposable elements in genomic sequences. In: Poptsova, M. (Ed.), *Genome Analysis: Current Procedures and Applications*. Poole: Caister Academic Press, pp. 165–181.
- Newman, A.M., Cooper, J.B., 2007. XSTREAM: A practical algorithm for identification and architecture modeling of tandem repeats in protein sequences. *BMC Bioinformatics* 8, 382.
- Novák, P., Ávila Robledillo, L., Koblížková, A., *et al.*, 2017. TAREAN: A computational tool for identification and characterization of satellite DNA from unassembled short reads. *Nucleic Acids Research* 45, e111.
- Novák, P., Neumann, P., Macas, J., 2010. Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. *BMC Bioinformatics* 11, 378.
- Novák, P., Neumann, P., Pech, J., Steinhais, J., Macas, J., 2013. RepeatExplorer: A galaxy-based web server for genome-wide characterization of eukaryotic repetitive elements from next-generation sequence reads. *Bioinformatics* 29, 792–793.
- Pendleton, M., Sebra, R., Pang, A.W., *et al.*, 2015. Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nature Methods* 12, 780–786.
- Petrov, D.A., Fiston-Lavier, A.S., Lipatov, M., Lenkov, K., González, J., 2011. Population genomics of transposable elements in *Drosophila melanogaster*. *Molecular Biology and Evolution* 28, 1633–1644.
- Platt, R.N., Blanco-Berdugo, L., Ray, D.A., 2016. Accurate transposable element annotation is vital when analyzing new genome assemblies. *Genome Biology and Evolution* 8, 403–410.
- Platzer, A., Nizhynska, V., Long, Q., 2012. TE-Locate: A tool to locate and group transposable element occurrences using paired-end next-generation sequencing data. *Biology* 1, 395–410.
- Price, A.L., Jones, N.C., Pevzner, A., 2005. *De novo* identification of repeat families in large genomes. *Bioinformatics* 21 (Suppl. 1), 351–358.
- Rahman, R., Chirn, G.W., Kanodia, A., *et al.*, 2015. Unique transposon landscapes are pervasive across *Drosophila melanogaster* genomes. *Nucleic Acids Research* 43, 10655–10672.
- Rho, M., Choi, J.H., Kim, S., Lynch, M., Tang, H., 2007. *De novo* identification of LTR retrotransposons in eukaryotic genomes. *BMC Genomics* 8, 90.
- Richard, G.-F., Kerrest, A., Dujon, B., 2008. Comparative genomics and molecular dynamics of DNA repeats in eukaryotes. *Microbiology and Molecular Biology Reviews* 72, 686–727.
- Rishishwar, L., Mariño-Ramírez, L., Jordan, I.K., 2016. Benchmarking computational tools for polymorphic transposable element detection. *Briefings in Bioinformatics*. bbw072.
- Robb, S.M.C., Lu, L., Valencia, E., *et al.*, 2013. The use of RelocaTE and unassembled short reads to produce high-resolution snapshots of transposable element generated diversity in rice. *G3* 3, 949–957.
- Rostant, W.G., Wedell, N., Hosken, D.J., 2012. Transposable elements and insecticide resistance. *Advances in Genetics* 78, 169–201.
- Saha, S., Bridges, S., Magbanua, Z.V., Peterson, D.G., 2008. Computational approaches and tools used in identification of dispersed repetitive DNA sequences. *Tropical Plant Biology* 1, 85–96.
- Santiago, N., Herráiz, C., Goñi, J.R., Messeguer, X., Casacuberta, J.M., 2002. Genome-wide analysis of the Emigrant family of MITEs of *Arabidopsis thaliana*. *Molecular Biology and Evolution* 19, 2285–2293.
- Schnable, S., Ware, D., Fulton, R.S., *et al.*, 2009. The B73 maize genome: Complexity, diversity, and dynamics. *Science* 326, 1112–1115.
- Segal, Y., Peissel, B., Renieri, A., *et al.*, 1999. LINE-1 elements at the sites of molecular rearrangements in Alport syndrome-diffuse leiomyomatosis. *American Journal of Human Genetics* 64, 62–69.
- Sharp, A.J., Locke, D.P., McGrath, S.D., *et al.*, 2005. Segmental duplications and copy-number variation in the human genome. *American Journal of Human Genetics* 77, 78–88.
- Smit, A.F.A., Hubley, R., Green, P., 1996–2010. RepeatMasker Open-3.0. Available at: <http://www.repeatmasker.org>.
- Sokol, D., Benson, G., Tojeira, J., 2007. Tandem repeats over the edit distance. *Bioinformatics* 23, e30–e35.
- Staton, S.E., Burke, J.M., 2015. Transposome: A toolkit for annotation of transposable element families from unassembled sequence reads. *Bioinformatics* 31, 1827–1829.
- Steinbiss, S., Willhoeft, U., Gremme, G., Kurtz, S., 2009. Fine-grained annotation and classification of *de novo* predicted LTR retrotransposons. *Nucleic Acids Research* 37, 7002–7013.
- Szak, S.T., Pickeral, O.K., Makalowski, W., *et al.*, 2002. Molecular archeology of L1 insertions in the human genome. *Genome Biology* 3, (research0052).
- Thung, D.T., de Ligt, J., Vissers, L.E., *et al.*, 2014. Mobster: Accurate detection of mobile element insertions in next generation sequencing data. *Genome Biology* 15, 488.
- Treangen, T.J., Salzberg, S.L., 2011. Repetitive DNA and next-generation sequencing: Computational challenges and solutions. *Nature Reviews Genetics* 13, 36–46.
- Tu, Z., Li, S., Mao, C., 2004. The changing tails of a novel short interspersed element in *Aedes aegypti*: Genomic evidence for slippage retrotransposition and the relationship between 3' tandem repeats and the poly(dA) tail. *Genetics* 168, 2037–2047.
- Tubio, J.M.C., Li, Y., Ju, Y.S., *et al.*, 2014. Mobile DNA in cancer. Extensive transduction of nonrepetitive DNA mediated by L1 retrotransposition in cancer genomes. *Science* 345, 1251343.
- Waterston, R.H., Lindblad-Toh, K., Birney, E., *et al.*, 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* 420, 520–562.
- Wicker, T., Schlagenhauf, E., Graner, A., *et al.*, 2006. 454 Sequencing put to the test using the complex genome of barley. *BMC Genomics* 7, 275.
- Wicker, T., Sabot, F., Hua-Van, A., *et al.*, 2007. A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics* 8, 973–982.
- Wu, J., Lee, W.P., Ward, A., *et al.*, 2014. Tangram: A comprehensive toolbox for mobile element insertion detection. *BMC Genomics* 15, 795.
- Xu, Z., Wang, H., 2007. LTR_FINDER: An efficient tool for the prediction of full-length LTR retrotransposons. *Nucleic Acids Research* 35 (Web Server), W265–W268.
- Zhuang, J., Wang, J., Theurkauf, W., Weng, Z., 2014. TEMP: A computational method for analyzing transposable element polymorphism in populations. *Nucleic Acids Research* 42, 6826–6838.
- Zytnicki, M., Akhunov, E., Quesneville, H., 2014. Tedna: A transposable element *de novo* assembler. *Bioinformatics* 30, 2656–2658.

Relevant Website

<http://www.repeatmasker.org/RepeatModeler.html>
Institute for Systems Biology – RepeatModeler Download.

Biographical Sketch



Emmanuelle Lerat is a CNRS researcher since 2005 working in the Laboratory “Biométrie et Biologie Evolutive” at the University Lyon 1, France. Her major research interest concerns the evolution of genomes in the light of their repeat content and more particularly the transposable elements (TEs). She has a strong background in molecular biology, in evolution, and in bioinformatics, with a long history in the annotation and analysis of TEs in various eukaryotic genomes, especially in *Drosophila*. She currently coordinates different subjects allowing to link informatic to biological questions concerning the functional impact of TEs on genome and gene evolution. She is a member of the Editorial Board of the international SBE journal “Genome Biology and Evolution” since 2008.

Bioinformatics Approaches for Studying Alternative Splicing

Prasoon K Thakur, Institute of Molecular Genetics of the ASCR, Prague Czech Republic

Hukam C Rawal, ICAR – National Research Centre on Plant Biotechnology, New Delhi, India

Mina Oluca, Institute of Molecular Genetics of the ASCR, Prague, Czech Republic

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal and University of Minho, Braga, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

Genes

Genes are genomic regions that contain necessary information for the expression of a protein or a molecule that ultimately helps in the survival, reproduction and function of the organism. The information stored in the DNA sequence of a gene is first transcribed into an RNA molecule, by RNA polymerase. For the protein coding genes, this transcript is called a messenger RNA (mRNA) which is then translated to synthesize proteins. The nascent RNA (pre-mRNA) transcripts contain intervening sequences, known as the *introns*, that do not become part of the final mRNA (Gilbert, 1978). The regions of pre-mRNA that are retained and ligated for translation are known as *exons*. The *introns* are removed from nascent RNA by a mechanism known as *splicing*. The spliced mRNA molecule forms a continuous protein-coding region ready to be translated into a protein molecule. Nearly all eukaryotes share the presence of *introns* and the mechanism of RNA *splicing*. The presence of introns plays a major role in facilitating recombination of RNA sequences and helps in developing protein diversity (Rogozin *et al.*, 2012). For a better understanding of the bioinformatics tools and approaches used to study alternative splicing, it is necessary to have a brief overview of the elements and mechanisms involved in splicing.

Eukaryotic vs Prokaryotic Splicing

Eukaryotes have devised a complex mechanism of modifying their primary RNA transcripts. Using this mechanism, called “*splicing*”, the nascent mRNA is processed so that the *introns* are selectively excised out and *exons* are ligated together. Splicing is catalyzed by a mega-dalton multi-ribonucleoprotein (RNP) complex known as the “*spliceosome*” (Will and Lührmann, 2011). The spliceosome is composed of five small nuclear RNPs (snRNPs) and numerous proteins. *Splicing* is more prevalent in higher eukaryotes and leads to the expression of a higher number of unique proteins from a single gene (Nilsen and Graveley, 2010). On the other hand, protein coding genes in prokaryotes lack introns which is complemented by the lack of a spliceosomal machinery (Rogozin *et al.*, 2012). This is why bacterial genes have a one-to-one correspondence of bases between a gene and its mRNA. However, some bacteria and viruses do contain some rare introns in non-coding RNAs that have been observed to self-splice *in vitro* (Woodson, 1998; Belfort, 1990; Schmidt, 1985; Berget *et al.*, 1977; Chow *et al.*, 1977). Presence of introns in early life forms has been discovered recently and this still remains a confounding fact (Roy and Gilbert, 2006).

Spliceosome and the Mechanism of Splicing

The post-transcriptional modification of the nascent mRNA (pre-mRNA) for the removal of introns is catalyzed by the spliceosome. The spliceosome is a large RNP complex composed of five snRNPs (U1, U2, U4, U5 and U6) and other accessory proteins (Staley and Guthrie, 1998; Jurica and Moore, 2003). Fig. 1 depicts various elements of the spliceosome. Each snRNP contains the corresponding uridine rich non-coding small RNA molecule (snRNA) and a variable number of complex-specific proteins (Hoskins *et al.*, 2011). Another variant of the spliceosome is known to exist; it is composed of other functionally analogous snRNPs and splices a rare class of introns (Patel and Steitz, 2003). Below, we describe the mechanism that relates to the well-explored and major spliceosome.

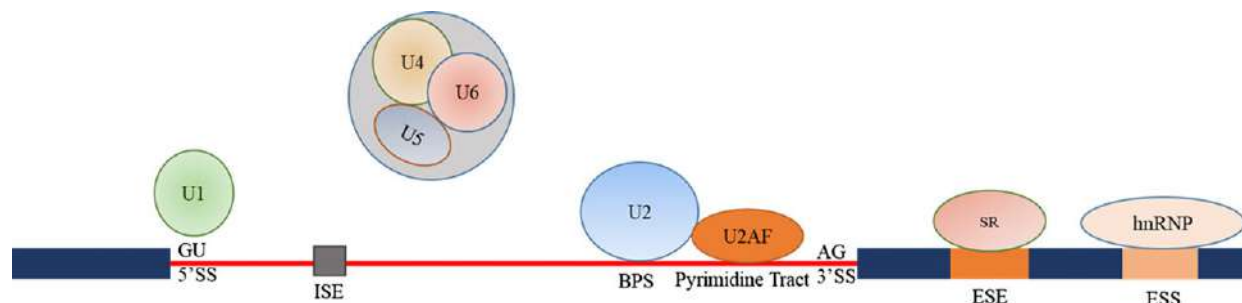


Fig. 1 Alternative splicing mechanism and events. Splicing factors recognize different sequences along the pre-mRNA. Conserved recognition sequences include intron boundaries named 5 ss and 3 ss. These elements interact with the spliceosome, recruiting snRNPs (U1, U2, U4, U5 and U6). Other regulatory elements are embedded in the exon (ESE, ESS) or in the intron (ISE) and could be recognized by proteins of the SR and/or hnRNP families.

The splicing reaction is a coordinated series of RNA–RNA, RNA–protein and protein–protein interactions (Trowitzsch *et al.*, 2009; Hoskins and Moore, 2012). First, U1 binds to the 5' splice site of an exon and U2 binds near branch point sequence (BPS) just upstream of the 3' splice site of the adjacent exon (Peled-Zehavi *et al.*, 2001). The U1 and U2 binding is important for the intron definition, thus for the accuracy of the splicing reaction. Later, a tri-snRNP complex, composed of U4/U6 and U5, joins in and leads to the formation of an active complex that catalyzes splicing. Once the splicing is over, the spliceosome disassembles and all components are recycled for future splicing reactions (Hnilicova and Stanek, 2011). Each splicing alternative is regulated through an interplay between constitutive splicing motifs, such as 5' splice sites, branch points, poly-pyrimidine tracts and 3' splice sites and components of the core splicing machinery. Exons and introns often contain *cis*-regulatory sequences, intronic enhancers or silencers that affect the splicing in a positive and negative manner. Splicing regulatory sequences (enhancer or silencer) are usually 6–10 base pair (bp) in length and induce binding of specific regulatory proteins such as serine/arginine rich (SR) proteins or heterogeneous nuclear ribonucleoproteins (hnRNPs) (Black, 2003; Wang and Burge, 2008; Chasin, 2007; Han *et al.*, 2010).

Types of Splicing

Splicing of a multi-exon pre-mRNA transcripts leads to two or more distinct mRNA products (Black, 2003; Wang and Burge, 2008; Nilsen and Graveley, 2010; Calarco *et al.*, 2011). Based on the complexity of the events, two different types of splicing have been described: *constitutive* splicing and *alternative* splicing (van den Hoogenhof *et al.*, 2016). *Constitutive splicing* (CS) is the process where the introns are removed from the pre-mRNA and the exons are joined together to form a mature mRNA. On the other hand, *alternative splicing* (AS), is the process where exons can be included or excluded in different combinations to create a diverse range of mRNA transcripts from a single pre-mRNA (Nilsen and Graveley, 2010). Computational and experimental studies have concurred that splicing signals at AS sites are weaker, the length of AS exons are shorter than that of CS exons, skipped exons tend to preserve the reading frame at a greater frequency than CS, and orthologous AS are more conserved than orthologous CS (Zheng *et al.*, 2005; Clark and Thanaraj, 2002; Garg and Green, 2007). Because of a rising interest towards AS in past years, this article is focused on it.

Alternative Splicing and its Mechanism

Alternative splicing (AS) is a mechanism by which one gene gives rise to multiple protein products with different functions (Roy *et al.*, 2013; Brett *et al.*, 2002). It was discovered when two different mRNA species were observed to be encoded by the same gene (Early *et al.*, 1980; Rosenfeld *et al.*, 1982). Recent high-throughput transcriptome analysis has suggested that about 95% of the mammalian multi-exonic genes are subjected to AS (Pan *et al.*, 2008). Four main AS events have been reported in literature namely, exon skipping, intron retention, alternative donor and alternative acceptor splice sites. AS events can be classified into different splicing patterns that include cassette exons, mutually exclusive exons, retained intron, alternative 5' splice sites, alternative 3' splice sites, alternative promoters, and alternative poly-A sites (Fig. 2; Keren *et al.*, 2010; Chen, 2011). Among them, the most common type of alternative splicing is including or skipping a cassette exon in the mature mRNA. In cassette exons, internal

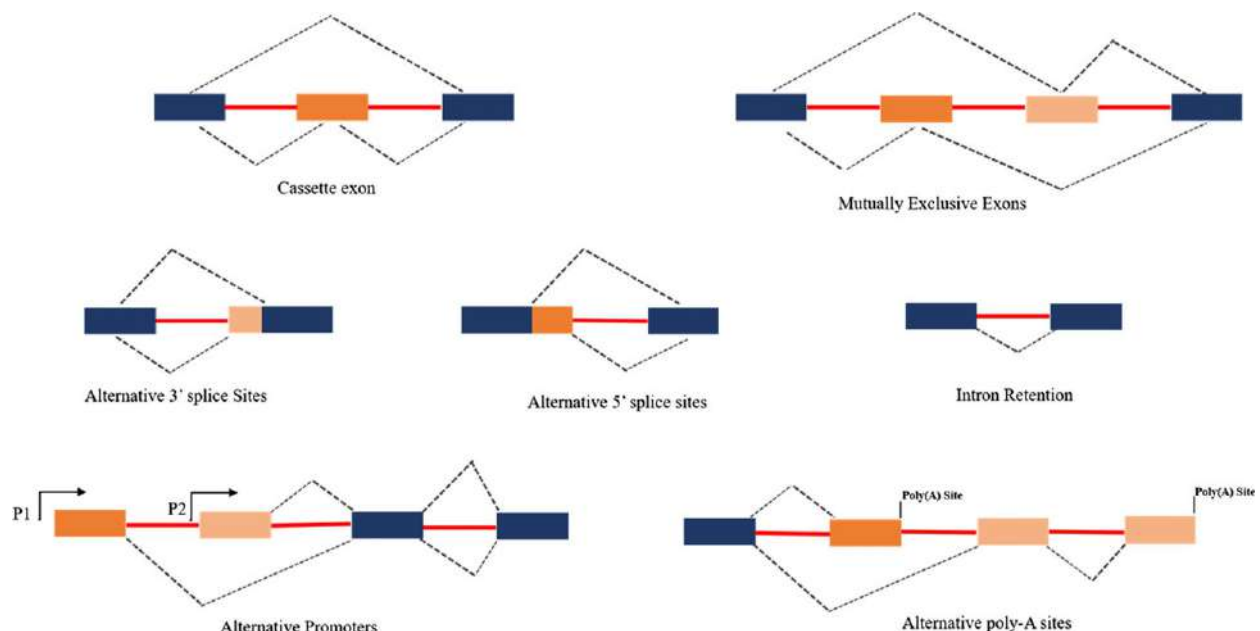


Fig. 2 Types of alternative splicing.

exons can either be included or excluded from the mature mRNA (Zhang *et al.*, 2016). In case of intron retention, the excision of an intron can be suppressed, which results in the retention of the entire intron and exons can be extended or shortened through the use of alternative 5' or 3' splice sites. In contrast, alternative promoters and alternative poly-A sites are alternative selection of transcription start sites or poly-A sites and are not due to alternative splicing. Incompletely spliced transcripts may contain intron fragments that can be inaccurately identified as intron retention, so it is hard to distinguish them from experimental artifacts. In general, intron retention is the most difficult AS event to detect. Many genes have multiple alternative splicing events with complex combinations of exons and produce diverse transcript isoforms. Most interesting example is, *Drosophila melanogaster's* Down syndrome cell adhesion molecule (Dscam) gene contains 20 constitutive and 95 cassette exons. These 95 cassette exons can express 38,016 different mature mRNAs (Graveley *et al.*, 2004; Missler and Sudhof, 1998).

Alternative Splicing Regulation

As already mentioned, AS increases the proteomic diversity in cells to adapt to various conditions. Therefore, it is important that AS be precisely regulated. Except the minimal core elements 5' splice site, 3' splice site and BP that are crucial for defining splicing sites, there are others instructions that help in discriminating between correct and incorrect splice sites and play important roles in AS regulation. These instructions are called as (exonic or intronic) splicing enhancers and/or silencers which are represented as 4–10 nucleotides. Two major classes of widely expressed *trans*-acting factors that are involved in binding these sequences and thus in controlling of splice site recognition are the SR proteins and heterogeneous ribonucleoproteins (hnRNPs). SR proteins got their name according to the domain enriched in dipeptides arginine/serine (RS). They bind to the pre-mRNA by their N terminal domains which is an RNA recognition motif (RRM) and mediates protein-protein interaction by SR domain facilitating assembly of spliceosome proteins (Long and Javier, 2009). Bradley and colleagues found that all eight SR proteins in *Drosophila* affect approximately 560 endogenous simple ASS events proving that they are indeed the key regulators of AS (Bradley *et al.*, 2015). The splicing repressor family of proteins are hnRNPs that were first identified as proteins that interact with heterogeneous nuclear RNA. They have their roles in telomerase maintenance, mRNA stability, regulation of transcription and translation (Geuens *et al.*, 2016). Several of the hnRNPs are identified as splicing repressors that suppress splicing by different mechanisms such as directly displacing snRNP binding, blocking interaction between U1 and U2 snRNP or looping out exons (Martinez-Contreras *et al.*, 2007). It is worth to mention that nuclear concentration of regulators help proper splicing patterns and that SR as well as hnRNP proteins control their own splicing patterns. At the end, these proteins can compete for binding to an overlapping ESE/ESS element (Zahler *et al.*, 2004) and this binding competition between them provides a combinatorial control in final decision of splice site choice.

Tissue-Specific AS in Humans

AS is considered an important mechanism for tissue-specific expression of transcript isoforms in humans. Recent EST based analyses have shown that the human brain contains highest fraction of alternatively spliced genes, followed by the liver and testis whereas human muscle, uterus, breast, stomach and pancreas were observed to have the lowest levels of genes undergoing AS (Lee *et al.*, 2003; Xu *et al.*, 2002; Stamm *et al.*, 2000). Moreover, human AS events have different usage pattern of alternative 5' splice site exons (A5Es) or alternative 3' splice site exons (A3Es). These studies suggest that the fraction of genes containing A3Es and A5Es have significantly high in the liver as in any other human tissue (Yeo *et al.*, 2004). Brain was observed to be the tissue with the second highest level of alternative usage for both 5' splice sites and 3' splice sites whereas muscle, uterus, breast, pancreas and stomach have shown the lowest level of A5Es and A3Es (Yeo *et al.*, 2004). AS events variation in human tissues are primarily regulated by *trans*-acting factors that bind on exonic and intronic *cis*-acting RNA elements (CAEs). A computational study using publicly available Affymetrix Genechip Human Exon Array dataset identified 652 *cis*-acting RNA elements (CAEs) across 11 human tissues (Wang *et al.*, 2009). Approximately, one third of all predicted CAEs matched with exonic splicing regulator databases and the vast majority of predicted CAEs were observed in the intronic regulatory regions. Most of these CAEs contribute to the AS between two tissues, while some are important in multiple tissues. Overall, the analysis suggests that genome-wide AS patterns are regulated by a combination of tissue-specific *cis*-acting elements and "general elements" whose functional activities are important but differ across multiple tissues (Wang *et al.*, 2009).

Bioinformatics Approaches for AS Detection

Until recently, systematic analysis of AS was done using expressed sequence tags (EST) (Gupta *et al.*, 2004; Sorek *et al.*, 2004; Xie *et al.*, 2002) or splicing-specific microarrays (Castle *et al.*, 2008; Clark *et al.*, 2002; Pan *et al.*, 2008). However, the detection of alternative splicing events has been significantly improved by next-generation sequencing (NGS). From this data, AS is identified through bioinformatics approaches by aligning the EST sequences with mRNA and genomic sequences using sequence alignment tools (Kim and Lee, 2008). Genomic sequence acts as the main reference sequence for detection and validation of AS events. With the availability of high-throughput techniques, multiple genomes have been sequenced and it has become feasible to study AS on a genomic scale (Pan *et al.*, 2008; Wang *et al.*, 2008). Therefore, a large number of alternative transcripts have been discovered along with extraction of distinctive features of alternatively spliced exons using bioinformatics (Roy *et al.*, 2013). With the

advances in tools used to study AS, it has also become easier to uncover the splicing dysregulations that lead to diseases. It is observed that dysregulation of alternative splicing affects various human conditions, including cancers (Ghigna *et al.*, 2010; Yap and Makeyev, 2013; Mills and Janitz, 2012; Poulos *et al.*, 2011; Mittendorf *et al.*, 2012; Manetti *et al.*, 2011; Endo-Umeda *et al.*, 2012; Medina and Krauss, 2013; Bogdanov, 2006; Lara-Pezzi *et al.*, 2012; Cooper, 2005; Miura *et al.*, 2011; Sampath and Pelus, 2007; Hagen and Ladomery, 2012; Yi and Tang, 2011; Omenn *et al.*, 2010). Databases like dbSNP (see “Relevant Websites section”) and ssSNPTarget (see “Relevant Websites section”) help to search for the splice site SNPs located in the genes of interest for the identification of any association to diseases (Tang *et al.*, 2013). These tools prove helpful in developing new splicing-targeted drugs or therapeutics.

AS detection usually involves three distinct stages. During the first step, transcript and genomic sequence data are processed to eliminate repetitive or ambiguous sequences. Poly-A tails are also removed during this pre-processing even though there may be some genuine poly-A genomics sequences. Second, the transcript sequences are aligned to the genomic sequence and “gene models” are deduced. The alignment to genomic sequence can be performed using tools like BLAT, GMAP or SPA (Kent, 2002; Wu and Watanabe, 2005; van Nimwegen *et al.*, 2006) with or independent of SIM4 (Florea *et al.*, 1998). Next, several possible alignments for a given sequence are screened out by choosing only the best hit as measured by percent identity and alignment coverage. Rest of the unspliced alignments can be removed if strictly focused on AS detection. Third, each of the alternative splicing events are identified and various alternative “gene models” are constructed. In order to exclude artifacts, individual AS events only consist of a pair of genomically non-overlapping splicing events. From here, liberal methods generate all possible combinations of the splicing events (splicing graphs) whereas more conservative methods seek only a minimal set of isoforms. Below, we present an overview of the role of bioinformatics approaches and tools in detecting alternative splicing along with its relevance in present era and challenges involved.

AS detection using sequence alignment: We can identify alternative splicing events by aligning ESTs with genomic and mRNA sequences using alignment tools like BLAST, BLAT (BLAST-Like Alignment Tool) or ClustalW. Using the alignment between ESTs and genomic sequences, one can detect the locations of exons and introns, and then by comparing their structures we can identify the alternative splicing events. For identifying alternative splicing by sequence alignment, we can further use alignment tools like GMAP or SPA to correct the genome alignments and generate valid alignments. Once we have identified the alternative splicing events, we can construct full-length alternatively spliced transcripts using graphical display tools including “Splice graph”. Although these alignments based methods are rapid, possible limitations that could affect the analysis include high sequencing errors, contamination, misalignments, low sequence coverage of ESTs.

AS detection using sequence conservation: AS events are conserved among different organisms *e.g.*, human and mouse, and therefore are of biological importance. Comparative genomics can provide us different means to predict the conserved exons that were spliced alternatively featuring the evolutionarily conserved alternative splicing events. As alternative exons have comparatively higher conservation level than constitutive exons in flanking intronic regions, hence can be used as good identifier for the alternative splicing (Chen, 2011).

AS detection using microarray data analysis: The above mentioned two approaches of sequence alignment and comparative genomics for identification of alternative splicing events, provides us only with the existence of an alternative splicing event but neither the degree nor the regulation of these alternative splicing events. The microarray study can provide us quantity of an alternative splicing event for a particular stage, condition or tissue of a cell. Previously, Affymetrix exon arrays were used by researchers to identify tissue-specific exons (Clark *et al.*, 2007) and differentially expressed AS between different human-cells (Yeo *et al.*, 2007). Some of the tools or methods that have been used widely for the alternative splicing microarray data analysis includes the splicing index calculation, Analysis of splice variation (ANOSVA), Finding isoforms using robust multichip analysis (FIRMA), a gene structure-based splice variant deconvolution method (DECONV), splicing prediction and concentration estimation (SPACE), and Generative model for the alternative splicing array platform (GenASAP). The details along with pros and cons of these methods is well described (Chen, 2011).

AS detection using RNA-seq data analysis: Alternative splicing is considered as main mechanism governing protein diversity and gene regulation. With the advancement of RNA-seq technology. It is possible to analyze the global impact and regulation of this biological process. There are increasing number of studies illustrates that the selection of wrong splice sites causes human disease. Therefore, the identification and quantification of differentially spliced transcripts are crucial for RNA-seq analysis.

Splicing Efficiency

Splicing efficiency (SE) (or splicing score; or splicing index) is used for the quantification of alternative splicing. SE is calculated as the ratio between the amounts of (spliced) mRNA and the (nascent) pre-mRNA. The conventional approach of using real-time quantitative PCR (RT-qPCR) with primers spanning exon-intron and exon-exon junctions (Hao and Baltimore, 2013) to determine SE is feasible for only a limited number of genes. However, with the availability of strong computational hardware and bioinformatics tools that can analyze the large amounts of RNA-seq data, it is now possible to determine genome-wide SE. Using RNA-seq data, several approaches have been used for calculating SE based on RNA-seq read counts from intronic, exonic or exon-exon junctions (Herzel and Neugebauer, 2015; Volanakis *et al.*, 2013). Below, we provide the main approaches for calculating SE (or scores):

- 1) *Exon-centered splicing score (ECSS):* ECSS is calculated using the splicing frequency around a given exon, by subtracting the read coverage over 2 kb of the upstream intron from the read coverage over 2 kb of the 5' end of the downstream intron (Pandya-Jones, 2011; Tilgner *et al.*, 2012).

- 2) *Intron-centered splicing score (ICSS)*: ICSS for each intron is calculated as the ratio of reads around the 3' splice site. The read coverage over the last 25 bp of a given intron is divided by the read coverage over the first 25 bp of the downstream exon (Carrillo Oesterreich *et al.*, 2010).
- 3) *Gene-based splicing score (GBSS)*: It is the splicing frequency for each gene, by dividing the read coverage over exons by the read coverage over the whole locus (Bhatt *et al.*, 2012).

ECSS is optimal for alternative cassette exon usage. However, it is not suitable for the first and terminal exons which have to be analyzed differently. Using ICSS, first and terminal exons can be included in the analysis but short exons are a disadvantage. In GBSS, the noise is reduced by the involvement of many reads in the analysis. However, this approach neither provides SE for individual introns nor any information about AS events per gene. The user may refer to Brugiolo *et al.* for further discussion regarding calculation of splicing efficiency using bioinformatics analysis (Brugiolo *et al.*, 2013).

Application of RNA-seq for Analyzing AS

RNA-sequencing (RNA-seq) uses next generation sequencing technology to sequence DNA molecules reversely transcribed from mRNAs. It has allowed the identification of RNA populations like miRNAs, lncRNAs, and several other species of RNA that in fact have opened new areas of biology (Lynch, 2015). Using RNA-seq data, it is indeed easier to quantify gene expression, previously unknown coding or non-coding transcripts, the composition of various isoforms, and study events like RNA editing, gene fusion and nucleotide variations. RNA-seq also provides an unprecedented opportunity to study AS quantitatively in a systematic way (Feng *et al.*, 2013). In recent years, many researchers used RNA-seq analysis to study AS. The bioinformatics tools and steps involved in the quantitative study of AS with RNA-seq are now well established, including visualization approaches (Feng *et al.*, 2013; Liu *et al.*, 2012; Song *et al.*, 2016; Mezlini *et al.*, 2013; Li *et al.*, 2011; Rogers *et al.*, 2012; Barann *et al.*, 2017). Therefore, several groups have studied AS using RNA-seq data. Pan *et al.* (2008) used RNA-seq for the first time to analyze the complexity of AS in 5 different human tissues. They estimated that about 95% of multi-exon genes undergo AS and discovered thousands of new splice junctions, as demonstrated by earlier studies as well (Sultan *et al.*, 2008). This observation was later corroborated by Wang *et al.* (2008) who analyzed the RNA-seq data from 15 different human tissue samples and cell line transcriptomes and observed that 92%–94% of human genes undergo AS. The same study also analyzed tissue-specific AS regulation, individual specific AS isoform variation, and alternative cleavage and polyadenylation (APA) events using RNA-seq data. Several important diseases that precipitate due to erroneous splicing have been studied using RNA-seq data (Garcia-Blanco *et al.*, 2004). Many studies have studied AS in the context of highly complicated diseases like cancer (Kalari *et al.*, 2012; Ren *et al.*, 2012; Shapiro *et al.*, 2011). In order to study genome-wide splicing regulation and mechanisms involved in transcriptome-splicing coupling, it is also possible to integrate many human RNA-seq datasets to identify splicing modules – a unit in the splicing regulatory network (Dai *et al.*, 2012; Li *et al.*, 2012). Schreiber *et al.* (2015) used the RNA-seq data from *Saccharomyces cerevisiae* to assess the impact of AS on its transcriptome. The detection of AS events has rapidly advanced with NGS in spite of the limitation due to the short sequencing read length. The upcoming 3rd generation sequencing technology like PacBio and Nanopore meant for long reads of RNA-seq will hopefully resolve these limitations in future.

Bioinformatics Tools for AS Detection

Several software tools have been developed for the identification of alternatively spliced exons and isoforms during the last decade. Herein, we are highlighting basic characteristics of some of the commonly used tools:

AStalavista (2007): AStalavista (Alternative Splicing transcriptional landscape visualization tool) is the first tool (a web server) for the dynamically and exhaustive extraction of complex AS events from annotated genes in order to compare different types of AS and distributions (Foissac and Sammeth, 2007). It is a JAVA-based tool available as a web server provided in “Relevant Websites section” and the latest version can be downloaded from the website provided in “Relevant Websites section”. For any given set of transcripts annotated with known exon–intron structure, AStalavista first performs an exhaustive pairwise comparison between all transcripts in a locus. To ensure an exhaustive detection of AS events, it pools all transcripts from a single transcriptional locus that overlap on the same strand of the genome sequence. It then dynamically assigns an AS code to the splicing events observed based on the relative position of splice sites. This code helps in automatic identification and exhaustive extraction of variations in the exon–intron structure (Sammeth *et al.*, 2008). AS events of the same type are given an identical code and classified in the same structural group whereas a new, concise and univocal AS code is assigned to each of the variant splicing structures. This generic protocol is applicable to any genome with or without annotation. Hence, AStalavista has an advantage over other methods that otherwise require a predefined splice form (as a reference transcript) for comparison. It detected more than 24,000 AS mechanistically un-elucidated events involving more than two alternatives in humans (Sammeth, 2009). Using this analysis, AStalavista generates a ranked list for each detected event type with its unique code (relative-position notation) and builds an AS landscape (in the form of a pie chart) for the given set of transcripts. This landscape of AS events can be used to investigate the transcriptome diversity across genes, chromosomes, and species.

AltAnalyze (2010): AltAnalyze (see “Relevant Websites section”) is a comprehensive tool for performing the analysis of alternative splicing data from Affymetrix Exon and Gene Arrays and their functional prediction at proteins and domains level (Emig *et al.*, 2010). It requires neither any programming knowledge nor any exon-array analysis expertise. It was designed for

computing alternative exon statistics based on the widely used rigorous statistical methods like FIRMA and MiDAS using ‘detection above background’ (DABG) *P*-value thresholds with other alternative exon analysis parameters for any number of raw Affymetrix CEL files. AltAnalyze provides outputs in tab-delimited text file format which can be either opened with Microsoft Excel like spreadsheet programs or can be visualized with DomainGraph (see “Relevant Websites section”) for further analysis.

GPSeq (2010): GPSeq is a tool to analyze RNA-seq data to estimate gene and exon expression, identify differentially expressed genes (DEGs), and differentially spliced exons (DSEs) through log-likelihood ratio approaches (Srivastava and Chen, 2010). It is based on a two-parameter (i.e., θ and λ) generalized Poisson (GP) model to fit to the position-level read counts across all of the positions of a gene/exon. Here the estimated parameter θ represents the transcript amount for the gene and λ represents the average bias during the sample preparation and sequencing process. After normalization of mapped reads a likelihood ratio test is used to identify DEGs or DSEs by treating the λ estimates as true values. Moreover, a simulation strategy can be adopted to better estimate the *P*-values. It deals with the fundamental problem of the distribution of the position-level read counts by proposing a two-parameter generalized Poisson (GP) model. This GP model fits the data much better than the traditional Poisson model by separating true signals from sequencing bias and aids in the better estimation of gene or exon expression, in performing a more reasonable normalization across different samples, and hence improve the identification of DEGs as well as DSEs. More importantly, it can deal with multiple RNA-seq data sets. The codes for the GP model were written in C and ‘GPseq’ is an R-package to implement all these methods available for download at the website provided in “see Relevant Websites section”. This tool is not available anymore.

MISO (2010): Mixture of Isoforms (MISO) is a statistical model that quantitates the expression level of alternatively spliced genes from RNA-seq data, and identifies differentially regulated isoforms or exons across samples (Katz et al., 2010). MISO model uses Bayesian inference by modeling the generative process to reproduce the reads from isoforms in RNA-seq for calculating the probability that a read originated from a particular isoform. It treats the expression level of a set of isoforms as a random variable and estimates a distribution over the values of this variable using a sampling based algorithm known as Markov Chain Monte Carlo (“MCMC”). These estimates can be quantified by the confidence intervals. It not just estimates the expression level of a single alternatively spliced exon (“exon-centric”), or of each transcript belonging to a gene (“isoform-centric”), but can quantify the levels of multiple isoforms produced by several nearby alternative splicing events. ISO is available as a Python package at the website provided in “Relevant Websites section”. It outputs the results as the exon/isoform expression levels in each sample along with confidence intervals which can be visualize alongside the RNA-seq data with sashimi-plot.

JuncBASE (2011): JuncBASE is used to identify and classify alternative splicing events from RNA-seq data (Brooks et al., 2010). JuncBASE is available from the GitHub repository located at the website provided in “Relevant Websites section”. Alternative splicing events are identified from splice junction reads from RNA-seq read alignments and annotated exon coordinates. JuncBASE also uses read counts to quantify the relative expression of each isoform and identifies splice events that are significantly differentially expressed across two or more samples.

SpliceTrap (2011): SpliceTrap is a method to quantify local exon inclusion levels by estimating the expression-levels of each exon using paired-end RNA-seq data (Wu et al., 2011). SpliceTrap generates alternative splicing profiles for different splicing patterns, such as exon skipping, alternative 5’ or 3’ splice sites, and intron retention. It can also identify major classes of alternative splicing events under a single cellular condition, without requiring a background set of reads to estimate relative splicing changes. It utilizes a comprehensive human exon database called TXdb (see section “Bioinformatics Approaches for AS Detection”) to estimate the expression level of every exon as an independent Bayesian inference problem. Unlike microarray-based methods, SpliceTrap relies on RNA-seq, and therefore it can determine the inclusion level of every exon within a single cellular condition, without requiring a background set of reads. Compared to Cufflinks and Scripture, it was shown to improve the accuracy, robustness and reliability in quantifying a large fraction of AS activity. SpliceTrap is useful in studying changes at the single-exon levels and can be helpful in the discovery of nearby *cis*-regulatory elements in diverse applications. It can also be implemented online through the CSH Galaxy server the website provided in “Relevant Websites section” and is also available for download and installation at the website provided in “Relevant Websites section”.

MATS (2012): MATS is a computational tool to detect differential AS events from RNA-seq data (Shen et al., 2012). The statistical model of MATS calculates the *P*-value and false discovery rate that the difference in the isoform ratio of a gene between two conditions exceeds a given user-defined threshold. From the RNA-seq data, MATS can automatically detect and analyze AS events corresponding to all major types of AS patterns. MATS handles replicate RNA-seq data from both paired and unpaired study design and has been designed to handle two RNA-seq samples.

SpliceSeq (2012): SpliceSeq is a Java based application, freely available at the website provided in “Relevant Websites Section”, for visualization and quantitation of RNA-seq reads for alternative splicing and to identify potential functional changes resulted from splice variation (Ryan et al., 2012). It aligns RNA-seq reads to gene splice graphs for accurate analysis of large and complex transcript variants. Firstly, an alignment database is generated from the imported RNA-seq data by using Bowtie and then SpliceSeq aligns reads to the splice graphs or gene models. Further, it evaluates the patterns of predicted transcript splicing for AS events and classifies these events in different types including exon skip, cassette exons, alternate promoter etc. The resulted alternative protein sequences are predicted with the weighted traversal of each gene’s splice graph.

SplicingViewer (2012): SplicingViewer is an integrated tool, freely accessed at the website provided in “Relevant Websites section” for detecting the splice junctions from known gene models or RNA-seq data, annotating the alternative splicing patterns using the splice junctions and visualizing these patterns (Liu et al., 2012). Firstly, the RNA-seq short reads are mapped to the provided or selected reference genome using aligners like BWA, Bowtie and SOAP. The mapped data are then converted to SAM/

BAM format by SAMtools to be used in SplicingViewer as input for further analysis and display alternative splicing patterns and the RNA-seq mapping result with a user-friendly interface, in a memory efficient and quick manner.

ASprofile (2013): ASprofile is computational framework for the identification of AS events in different RNA-seq samples (Florea et al., 2013). It comprises of three main programs for extracting (*extract-as*), quantifying (*extract-as-fpkm*) and comparing (*collect-fpkm*) AS events from transcripts assembled from RNA-seq data in multiple conditions. First, *extract-as* program takes as input a GTF transcript file and compares all pairs of transcripts within a gene to determine exon-intron structure differences that indicate an AS event. Second, *extract-as-fpkm* calculates the FPKM of each AS event from those of transcripts. Finally, *collect-fpkm* collects the FPKM event values for all RNA-seq samples, calculates and compares splicing ratios across samples.

ASprofile are providing following types of AS event such as exon skipping (SKIP), cassette exons (MSKIP), alternative transcript start and termination (TSS, TTS), retention of single or multiple introns (IR, MIR), and alternative exon (AE). ASprofile has been implemented using RNA-seq data from Illumina's Human Body Map project and authors have provided global view of alternative splicing events in 16 different human tissues. The list of all AS events catalog and the ASprofile software are freely available and can be access through their web site the website provided in "Relevant Websites section".

DiffSplice (2013): DiffSplice is tools for the identification and quantification of alternative splicing events (Hu et al., 2013). This software is available at the website provided in "Relevant Websites section". This approach starts with the identification of alternative splicing module (ASMs) from the splice graph that created directly from the exons and introns predicted from RNA-seq read alignments. The abundance of alternative splicing isoforms residing in each ASM is estimated for each sample and is compared across sample groups. DiffSplice takes as input the SAM files and results are summarized as a decomposition of the genome. It can be visualized using the UCSC genome browser. The approach does not depend on transcript or gene annotations. It evades the need for full transcript inference and quantification, which is a usually difficult because of short read lengths, as well as various sampling biases.

DSGseq (2013): DSGseq is method to identify differentially spliced genes between two groups of samples by comparing read counts on all exons (Wang et al., 2013). DSGseq software is available at the website provided in "Relevant Websites section". DSGseq tools is use negative binomial (NB) distribution to model sequencing reads on exons, and propose a NB-statistic to detect differentially spliced genes between two groups of RNA-seq samples by comparing read counts on all exons. It produces the tabular output the differences in the relative abundance of the isoforms of each gene in two groups of samples and relative abundance of the isoforms of each exon in each gene. It also ranks the exon that the most significant difference. This is a novel exon-based approach and it does not need isoform composition information and isoform expression estimation. Experiment on simulated and real RNA-seq data shows that this method has good performance and applicability. DSGseq method can identify the exons that contribute the best to the differential splicing. It can also identify previously unknown alternative splicing events.

GLiMMPs (2013): GLiMMPs (Generalized Linear Mixed Model Prediction of sQTL) is a robust statistical method for detecting splicing quantitative trait loci (sQTLs) from RNA-seq data (Zhao et al., 2013). It works at a low false positive rate and characterizes the genetic variation of alternative splicing. It takes into account the individual variation in sequencing coverage and the noise prevalent in RNA-seq data. To begin with, it needs a set of identified AS events and RNA-seq reads mapped to splice junctions detected for the estimation of the exon inclusion levels. GLiMMPs uses the reads information from both exon inclusion and skipping isoforms to model the estimation uncertainty of exon inclusion level. The source code used to be available from the website provided in "Relevant Websites section". However, currently, it appears to be discontinued.

SpliceR (2014): SpliceR is an R package for classification of alternative splicing and prediction of coding potential (Vitting-Seerup et al., 2014). SpliceR is implemented as an R package based on standard Bioconductor classes and is freely available from the Bioconductor repository (see "Relevant Websites section"). spliceR uses the full-length transcript output from RNA-seq assemblers (like *Cufflinks*) to detect single or multiple exon skipping, alternative donor and acceptor sites, intron retention, alternative first or last exon usage, and mutually exclusive exon events. For each gene, spliceR constructs the hypothetical pre-RNA based on the exon information from all transcripts originating from that gene. Subsequently, all transcripts are compared to this hypothetical pre-RNA in a pairwise manner. AS events are classified and annotated. It is flexible and can be easily integrated in existing pipelines or workflows. It predicts coding potential of transcripts, calculates untranslated region (UTR) and open reading frame (ORF) lengths. Moreover, it also predicts whether transcripts are nonsense mediated decay (NMD)-sensitive based on compatible annotated start codon positions and their downstream ORF.

GESS (2014): Graph-based exon-skipping scanner (GESS) is computational method to detect *de novo* exon-skipping events directly from raw RNA-seq data without the prior knowledge of gene annotation information (Wang et al., 2017). It can also detect the dominant isoform among the skipping- or inclusion- isoforms, generated for these skipping event sites and presents a more accurate and comprehensive data of skipping events associated with a particular physiological condition within a cell. GESS tools is available at the website provided in "Relevant Websites section". First, a splice-site-link graph is created from the splicing-aware aligned reads using a greedy algorithm. The sub-graphs are navigated iteratively by implementing a walking strategy to reveal a pattern corresponding to an exon-skipping event. Finally, the MISO model is implemented to obtain the ratio of skipping isoform *vs.* inclusion isoform and the dominated isoform is determined. In this method, the input must be sorted bam file and exon-skipping sites output from GESS can be use by MISO for Ψ -value (Percent Spliced Isoform) calculation. GESS method is capable of capturing *de novo* exon-skipping events directly from using raw RNA-seq reads.

rMATS (2014): As described previously, MATS works for detecting differential AS between two RNA-seq samples. Similarly, rMATS can be used to detect differential AS from replicate RNA-seq data using a hierarchical model to highlight the sampling

uncertainty in individual replicates and the possible variability among them (Shen *et al.*, 2014). rMATS is flexible in testing the splicing difference above any user-defined threshold. Being a general method for analyzing mRNA isoform ratios from read-counts, rMATS method can also be used for sequencing-based analyses of other types of mRNA isoform variations including alternative polyadenylation and RNA editing. The rMATS source code is freely available at the website provided in “Relevant Websites section”. It takes the raw RNA-seq reads, a genome sequence file, and a transcript annotation file as the input to identify the alternative splicing events corresponding to all major types of alternative splicing patterns and calculates the *P* value and FDR for differential splicing.

FineSplice (2014): FineSplice is a Python wrapped splice junction detection algorithm combined with TopHat2 for a reliable identification of expressed exon junctions from RNA-seq data (Gatto *et al.*, 2014). FineSplice software is freely available at the website provided in “Relevant Websites section”. During the first step, it performs the transcriptome alignment with *de novo* splice junction discovery using TopHat2 with available annotations for known transcript isoforms. The resulting binary alignment map (BAM) file is used as input by FineSplice. At next step, it computes the set of split-read overhangs across each junction for each of the uniquely mapping read. It labels the splice junctions with no matching overhang and defines a potential false positives subset. Then for each junction it constructs a feature vector based on the log₂ deviation. With a defined class label and feature vector, it fits a L1-regularized logistic regression model over the whole set of junctions. FineSplice outputs a confident set of expressed splice junctions with the corresponding read counts. Potential false positives arising from spurious alignments are filtered out via a semi-supervised anomaly detection strategy based on logistic regression. Multiple mapping reads with a unique location after filtering are rescued and reallocated to the most reliable candidate location. This is conjugate approach for an efficient mapping solution with a semi-supervised anomaly detection scheme to filter out false positives. It allows reliable estimation of expressed junctions from the alignment output.

Splicing Code Prediction

Mutations in the splicing signals (5' splice site, 3' splice site and branch point), splicing silencer and enhancer motifs can affect the proper binding of spliceosome and other RNA binding proteins that changes AS patterns. These changes would lead to altered gene products and several diseases as a consequence. Therefore, it is important to be able to predict splicing codes that potentially affect AS. There are several online bioinformatics tools available for the prediction of splicing code that could be useful to understand human variation on splicing and its consequences in human diseases. These predictions are taking advantage of experimentally binding data of RNA binding proteins such as SR proteins as well as experimentally validated splicing signals sequences. These tools take primary sequence as input and predict the splicing regulatory binding sites, properties of exon/intron such as splice site strength, branch point and other regulatory binding motifs.

Tools for Splice Code Prediction: Table 1 summarizes some of such online bioinformatics tools, like Human Splicing Finder (HSF), RegRNA, ESEfinder, Alternative Splice Site Predictor (ASSP), SplicePort, EX-SKIP, RESCUE-ESE, Maxentscan and SROOGLE (Desmet *et al.*, 2009; Huang *et al.*, 2006; Cartegni *et al.*, 2003; Wang and Marin, 2006; Dogan *et al.*, 2007; Raponi *et al.*, 2011; Fairbrother *et al.*, 2004; Eng *et al.*, 2004; Schwartz *et al.*, 2009). Maxentscan (Yeo and Burge, 2004), SplicePort (Dogan *et al.*, 2007), HSF (Desmet *et al.*, 2009) and SFmap (Paz *et al.*, 2010) can be used for the prediction of the basic *cis*-acting elements (5' splice site, 3' splice site and branch point). All these programs provide a useful indication of whether donor, acceptor and branch-point are well defined compared with ideal consensus sequences. ESEfinder and RESCUE-ESE are dedicated to disruption or creation of splicing regulatory elements (SREs) (Cartegni *et al.*, 2003; Fairbrother *et al.*, 2004). In addition, SFmap server is useful to search for the splicing factor binding motifs in the sequence of interest (Paz *et al.*, 2010). RegRNA and SROOGLE are servers can be useful in searching for both basic splicing signals and SREs (Huang *et al.*, 2006; Schwartz *et al.*, 2009). SROOGLE, is one of the most used tools, which provides a graphic output and displays the four core splicing signals with scores based on nine different algorithms. It also highlights the sequences belonging to 13 different groups of SREs. EX-SKIP application compares the ESE/ESS profile of a wild-type and a mutated allele to quickly determine which exonic variant has the highest chance to skip this exon (Raponi *et al.*, 2011).

Performance of Splice Code Predictors: One of the key questions that naturally arises is regarding the performance of these programs for the identification of possible splicing mutations. In humans, SREs tend to be reasonably conserved (Burset *et al.*, 2001) and the programs that evaluate their relative strengths seem to be more successful than those that aim to target the much more loosely conserved SRE elements. MaxEntScan takes the nucleotide dependencies within donor site sequences into account and has been shown to be the best predictors of cryptic splice site activation in disease-causing mutations (Houdayer *et al.*, 2008). Currently, SROOGLE is the most used tool that provides information on splice-sites or enhancer/silencer disruption in a single interface (Schwartz *et al.*, 2009). Some programs such as ESEfinder have worked well in some contexts (Zatkova *et al.*, 2004; Kralovicova and Vorechovsky, 2007; Wang *et al.*, 2005) and have led to a scientific debate in others (Cartegni and Krainer, 2002; Cartegni *et al.*, 2006; Fairbrother *et al.*, 2004; Pfarr *et al.*, 2005; Deburgrave *et al.*, 2007). Combination of all these resources still represents the best chance of “predicting” putative splicing mutations whether in conserved or less conserved regions (Houdayer *et al.*, 2008; Soukariéh *et al.*, 2016). Therefore, it appears that the future trend in developing such prediction algorithms or software will be to integrate all analyses on a single platform. This way, it will be easy to obtain most of the in depth details on global splicing signals on the same platter/platform. Due the progress in this field and improvement in bioinformatics predictions, methodologies related to splicing diagnostics are enticing. In such methodologies, the information obtained from *in silico* bioinformatics approaches are translated to wet lab (*in vitro* and *in vivo*) systems to evaluate splicing efficiencies (Baralle and Buratti, 2017).

Table 1 Online tools for splicing code prediction

<i>Tools</i>	<i>Input</i>	<i>Output</i>	<i>Weblink</i>
Human splicing finder	mRNA sequence	Consensus values of potential splice sites and search for branch points	http://www.umd.be/HSF3/
RegRNA	mRNA sequence	Motifs in mRNA 5'-UTR and 3'-UTR, motifs involved in mRNA splicing, motifs involved in transcriptional regulation, other motifs in mRNA, such as riboswitches, prediction of the splice sites, such as splicing donor/acceptor sites, RNA structural features, such as inverted repeat, and miRNA target sites	http://regrna.mbc.nctu.edu.tw/html/prediction.html
ESEfinder	Exonic sequence	ESE finder use to find the presence of exonic splicing enhancer elements	http://exon.cshl.edu/ESE/
Alternative Splice Site Predictor (ASSP)	Raw sequence, DNA/RNA	Putative alternative exon isoform, cryptic, and constitutive splice sites of internal (coding) exons	http://wangcomputing.com/assp/index.html
SplicePort	Multiple or single sequence	Splice-site predictions and user can also browse feature associated with the prediction	http://spliceport.cbcb.umd.edu
EX-SKIP	Two exonic sequence (up to 4000bp)	It calculates the total number of ESSs, ESEs and their ratio	http://ex-skip.img.cas.cz/
RESCUE-ESE	RNA or DNA Sequence (4 k)	Exonic splicing enhancers (ESEs) prediction	http://genes.mit.edu/burgelab/rescue-ese/
Maxentscan	Each sequence must be the same length	Building distributions over short sequence motifs	http://genes.mit.edu/burgelab/maxent/Xmaxent.html
	5' splice site sequence [3 bases in exon + 6 bases in intron]	5' splice site score	http://genes.mit.edu/burgelab/maxent/Xmaxentscan_scoreseq.html
	3' splice site sequence. [20 bases in the intron + 3 base in the exon]	3' splice site score	http://genes.mit.edu/burgelab/maxent/Xmaxentscan_scoreseq_acc.html
SROOGLE	Exon along with the introns	Graphic display of splicing related data on DNA segments	http://sroogle.tau.ac.il/
SFmap	Human genomic sequence or a list of sequences in FASTA format	Splicing factor binding motifs	http://sfmap.technion.ac.il/

Concluding Remarks

Field of alternative splicing has become very active and attractive in recent years, especially due to the development of high throughput technologies. Consequently, a large number of bioinformatics tools are available for various analyses regarding AS. In fact, it is hard to list every tool developed as of yet, nonetheless, we have reviewed bioinformatics software and algorithms that are used for the detection of splicing events using RNA-seq or other data. Drawbacks or advantages of each approach has been put forth. Some popular online tools and software that can be used to study intronic and exonic mutations leading to splicing defects, have also been discussed.

Acknowledgment

PKT was supported by the Czech Science Foundation (P305/12/G034) and the institutional support (RVO68378050).

See also: Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome Annotation. Genome Annotation: Perspective From Bacterial Genomes. Genome Databases and Browsers. Genome Informatics. Integrative Analysis of Multi-Omics Data. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Prediction of Coding and Non-Coding RNA. Quantitative Immunology by Data Analysis Using Mathematical Models. Sequence Analysis. Whole Genome Sequencing Analysis

References

- Baralle, D., Buratti, E., 2017. RNA splicing in human disease and in the clinic. *Clin. Sci. (Lond.)* 131 (5), 355–368.
- Barann, M., Zimmer, R., Birzele, F., 2017. Manananggal – A novel viewer for alternative splicing events. *BMC Bioinform.* 18 (1), 120.
- Belfort, M., 1990. Phage T4 introns: Self-splicing and mobility. *Annu. Rev. Genet.* 24 (1), 363–385.
- Berget, S.M., Moore, C., Sharp, P.A., 1977. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proc. Natl. Acad. Sci.* 74 (8), 3171–3175.
- Bhatt, D.M., *et al.*, 2012. Transcript dynamics of proinflammatory genes revealed by sequence analysis of subcellular RNA fractions. *Cell* 150 (2), 279–290.
- Black, D.L., 2003. Mechanisms of alternative pre-messenger RNA splicing. *Annu. Rev. Biochem.* 72, 291–336.
- Bogdanov, V.Y., 2006. Blood coagulation and alternative pre-mRNA splicing: An overview. *Curr. Mol. Med.* 6 (8), 859–869.
- Bradley, T., Cook, M.E., Blanchette, M., 2015. SR proteins control a complex network of RNA-processing events. *RNA* 21 (1), 75–92.
- Brett, D., *et al.*, 2002. Alternative splicing and genome complexity. *Nat. Genet.* 30 (1), 29–30.
- Brooks, A.N., *et al.*, 2010. Conservation of an RNA regulatory map between *Drosophila* and mammals. *Genome Res.*
- Brugiolio, M., Herzl, L., Neugebauer, K.M., 2013. Counting on co-transcriptional splicing. *F1000Prime Rep.* 5, 9.
- Burset, M., Seledtsov, I.A., Soloviyev, V.V., 2001. SpliceDB: Database of canonical and non-canonical mammalian splice sites. *Nucleic Acids Res.* 29 (1), 255–259.
- Calarco, J.A., Zhen, M., Blencowe, B.J., 2011. Networking in a global world: Establishing functional connections between neural splicing regulators and their target transcripts. *RNA* 17 (5), 775–791.
- Carrillo Oesterreich, F., Preibisch, S., Neugebauer, K.M., 2010. Global analysis of nascent RNA reveals transcriptional pausing in terminal exons. *Mol. Cell* 40 (4), 571–581.
- Cartegni, L., *et al.*, 2003. ESEfinder: A web resource to identify exonic splicing enhancers. *Nucleic Acids Res.* 31 (13), 3568–3571.
- Cartegni, L., *et al.*, 2006. Determinants of exon 7 splicing in the spinal muscular atrophy genes, SMN1 and SMN2. *Am. J. Hum. Genet.* 78 (1), 63–77.
- Cartegni, L., Krainer, A.R., 2002. Disruption of an SF2/ASF-dependent exonic splicing enhancer in SMN2 causes spinal muscular atrophy in the absence of SMN1. *Nat. Genet.* 30 (4), 377–384.
- Castle, J.C., *et al.*, 2008. Expression of 24,426 human alternative splicing events and predicted cis regulation in 48 tissues and cell lines. *Nat. Genet.* 40 (12), 1416–1425.
- Chasin, L.A., 2007. Searching for splicing motifs. *Adv. Exp. Med. Biol.* 623, 85–106.
- Chen, L., 2011. Statistical and computational studies on alternative splicing. In: Lu, H.H.-S., Schölkopf, B., Zhao, H. (Eds.), *Handbook of Statistical Bioinformatics*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 31–53.
- Chow, L.T., *et al.*, 1977. An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. *Cell* 12 (1), 1–8.
- Clark, F., Thanaraj, T.A., 2002. Categorization and characterization of transcript-confirmed constitutively and alternatively spliced introns and exons from human. *Hum. Mol. Genet.* 11 (4), 451–464.
- Clark, T.A., *et al.*, 2007. Discovery of tissue-specific exons using comprehensive human exon microarrays. *Genome Biol.* 8 (4), R64.
- Clark, T.A., Sugnet, C.W., Ares Jr., M., 2002. Genomewide analysis of mRNA processing in yeast using splicing-specific microarrays. *Science* 296 (5569), 907–910.
- Cooper, T.A., 2005. Alternative splicing regulation impacts heart development. *Cell* 120 (1), 1–2.
- Dai, C., *et al.*, 2012. Integrating many co-splicing networks to reconstruct splicing regulatory modules. *BMC Syst. Biol.* 6 (1), S17.
- Deburgrave, N., *et al.*, 2007. Protein- and mRNA-based phenotype-genotype correlations in DMD/BMD with point mutations and molecular basis for BMD with nonsense and frameshift mutations in the DMD gene. *Hum. Mutat.* 28 (2), 183–195.
- Desmet, F.O., *et al.*, 2009. Human Splicing Finder: An online bioinformatics tool to predict splicing signals. *Nucleic Acids Res.* 37 (9), e67.
- Dogan, R.I., *et al.*, 2007. SplicePort – An interactive splice-site analysis tool. *Nucleic Acids Res.* 35 (Web Server issue), W285–W291.
- Early, P., *et al.*, 1980. Two mRNAs can be produced from a single immunoglobulin mu gene by alternative RNA processing pathways. *Cell* 20 (2), 313–319.
- Emig, D., *et al.*, 2010. AltAnalyze and DomainGraph: Analyzing and visualizing exon expression data. *Nucleic Acids Res.* 38 (Web Server issue), W755–W762.
- Endo-Umeda, K., *et al.*, 2012. Differential expression and function of alternative splicing variants of human liver X receptor alpha. *Mol. Pharmacol.* 81 (6), 800–810.
- Eng, L., *et al.*, 2004. Nonclassical splicing mutations in the coding and noncoding regions of the ATM Gene: Maximum entropy estimates of splice junction strengths. *Hum. Mutat.* 23 (1), 67–76.
- Fairbrother, W.G., *et al.*, 2004. RESCUE-ESE identifies candidate exonic splicing enhancers in vertebrate exons. *Nucleic Acids Res.* 32 (Web Server issue), W187–W190.
- Fairbrother, W.G., *et al.*, 2004. Single nucleotide polymorphism-based validation of exonic splicing enhancers. *PLOS Biol.* 2 (9), E268.
- Feng, H., Qin, Z., Zhang, X., 2013. Opportunities and methods for studying alternative splicing in cancer with RNA-Seq. *Cancer Lett.* 340 (2), 179–191.
- Florea, L., *et al.*, 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.* 8 (9), 967–974.
- Florea, L., Song, L., Salzberg, S.L., 2013. Thousands of exon skipping events differentiate among splicing patterns in sixteen human tissues. *F1000Res* 2, 188.
- Foissac, S., Sammeth, M., 2007. ASTALAVISTA: Dynamic and flexible analysis of alternative splicing events in custom gene datasets. *Nucleic Acids Res.* 35 (Web Server issue), W297–W299.
- Garcia-Blanco, M.A., Baraniak, A.P., Lasda, E.L., 2004. Alternative splicing in disease and therapy. *Nat. Biotechnol.* 22, 535.
- Garg, K., Green, P., 2007. Differing patterns of selection in alternative and constitutive splice sites. *Genome Res.* 17 (7), 1015–1022.
- Gatto, A., *et al.*, 2014. FineSplice, enhanced splice junction detection and quantification: A novel pipeline based on the assessment of diverse RNA-Seq alignment solutions. *Nucleic Acids Res.* 42 (8), e71.
- Geuens, T., Bouhy, D., Timmerman, V., 2016. The hnRNP family: Insights into their role in health and disease. *Hum. Genet.* 135, 851–867.
- Ghigna, C., *et al.*, 2010. Pro-metastatic splicing of Ron proto-oncogene mRNA can be reversed: Therapeutic potential of bifunctional oligonucleotides and indole derivatives. *RNA Biol.* 7 (4), 495–503.
- Gilbert, W., 1978. Why genes in pieces? *Nature* 271 (5645), 501.
- Graveley, B.R., *et al.*, 2004. The organization and evolution of the dipteran and hymenopteran Down syndrome cell adhesion molecule (Dscam) genes. *RNA* 10 (10), 1499–1506.
- Gupta, S., *et al.*, 2004. Strengths and weaknesses of EST-based prediction of tissue-specific alternative splicing. *BMC Genom.* 5, 72.
- Hagen, R.M., Ladomery, M.R., 2012. Role of splice variants in the metastatic progression of prostate cancer. *Biochem. Soc. Trans.* 40 (4), 870–874.
- Han, S.P., *et al.*, 2010. Functional implications of the emergence of alternative splicing in hnRNP A/B transcripts. *RNA* 16 (9), 1760–1768.
- Hao, S., Baltimore, D., 2013. RNA splicing regulates the temporal order of TNF-induced gene expression. *Proc. Natl. Acad. Sci. USA* 110 (29), 11934–11939.
- Herztl, L., Neugebauer, K.M., 2015. Quantification of co-transcriptional splicing from RNA-Seq data. *Methods* 85, 36–43.
- Hnilicova, J., Stanek, D., 2011. Where splicing joins chromatin. *Nucleus* 2 (3), 182–188.
- Hoskins, A.A., *et al.*, 2011. Ordered and dynamic assembly of single spliceosomes. *Science* 331 (6022), 1289–1295.
- Hoskins, A.A., Moore, M.J., 2012. The spliceosome: A flexible, reversible macromolecular machine. *Trends Biochem. Sci.* 37 (5), 179–188.
- Houdayer, C., *et al.*, 2008. Evaluation of in silico splice tools for decision-making in molecular diagnosis. *Hum. Mutat.* 29 (7), 975–982.
- Hu, Y., *et al.*, 2013. DiffSplice: The genome-wide detection of differential splicing events with RNA-seq. *Nucleic Acids Res.* 41 (2), e39.
- Huang, H.Y., *et al.*, 2006. RegRNA: An integrated web server for identifying regulatory RNA motifs and elements. *Nucleic Acids Res.* 34 (Web Server issue), W429–W434.
- Jurica, M.S., Moore, M.J., 2003. Pre-mRNA splicing: Awash in a sea of proteins. *Mol. Cell* 12 (1), 5–14.
- Kalari, K.R., *et al.*, 2012. Deep sequence analysis of non-small cell lung cancer: integrated analysis of gene expression, alternative splicing, and single nucleotide variations in lung adenocarcinomas with and without oncogenic KRAS mutations. *Front. Oncol.* 2, 12.

- Katz, Y., *et al.*, 2010. Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nat. Methods* 7 (12), 1009–1015.
- Kent, W.J., 2002. BLAT – The BLAST-like alignment tool. *Genome Res.* 12 (4), 656–664.
- Keren, H., Lev-Maor, G., Ast, G., 2010. Alternative splicing and evolution: Diversification, exon definition and function. *Nat. Rev. Genet.* 11 (5), 345–355.
- Kim, N., Lee, C., 2008. Bioinformatics detection of alternative splicing. In: Keith, J.M. (Ed.), *Bioinformatics: Data, Sequence Analysis and Evolution*. Totowa, NJ: Humana Press, pp. 179–197.
- Kralovicova, J., Vorechovsky, I., 2007. Global control of aberrant splice-site activation by auxiliary splicing sequences: Evidence for a gradient in exon and intron definition. *Nucleic Acids Res.* 35 (19), 6399–6413.
- Lara-Pezzi, E., Dopazo, A., Manzanares, M., 2012. Understanding cardiovascular disease: A journey through the genome (and what we found there). *Dis. Model Mech.* 5 (4), 434–443.
- Lee, C., *et al.*, 2003. ASAP: The alternative splicing annotation project. *Nucleic Acids Res.* 31 (1), 101–105.
- Li, J.J., *et al.*, 2011. Sparse linear modeling of next-generation mRNA sequencing (RNA-Seq) data for isoform discovery and abundance estimation. *Proc. Natl. Acad. Sci. USA* 108 (50), 19867–19872.
- Liu, Q., *et al.*, 2012. Detection, annotation and visualization of alternative splicing from RNA-Seq data with SplicingViewer. *Genomics* 99 (3), 178–182.
- Li, W., *et al.*, 2012. Algorithm to identify frequent coupled modules from two-layered network series: Application to study transcription and splicing coupling. *J. Comput. Biol.* 19 (6), 710–730.
- Long, J.C., Javier, F.C., 2009. The SR protein family of splicing factors: Master regulators of gene expression. *Biochem. J.* 417 (1), 15–27.
- Lynch, K.W., 2015. Thoughts on NGS, alternative splicing and what we still need to know. *RNA* 21 (4), 683–684.
- Manetti, M., *et al.*, 2011. Impaired angiogenesis in systemic sclerosis: The emerging role of the antiangiogenic VEGF(165)b splice variant. *Trends Cardiovasc. Med.* 21 (7), 204–210.
- Martinez-Contreras, R., *et al.*, 2007. hnRNP proteins and splicing control. *Adv. Exp. Med. Biol.* 623, 123–147.
- Medina, M.W., Krauss, R.M., 2013. Alternative splicing in the regulation of cholesterol homeostasis. *Curr. Opin. Lipidol.* 24 (2), 147–152.
- Mezlini, A.M., *et al.*, 2013. iReckon: Simultaneous isoform discovery and abundance estimation from RNA-seq data. *Genome Res.* 23 (3), 519–529.
- Mills, J.D., Janitz, M., 2012. Alternative splicing of mRNA in the molecular pathology of neurodegenerative diseases. *Neurobiol. Aging* 33 (5), 1012. e11–24.
- Missler, M., Sudhof, T.C., 1998. Neurexins: Three genes and 1001 products. *Trends Genet.* 14 (1), 20–26.
- Mittendorf, K.F., *et al.*, 2012. Tailoring of membrane proteins by alternative splicing of pre-mRNA. *Biochemistry* 51 (28), 5541–5556.
- Miura, K., Fujibuchi, W., Sasaki, I., 2011. Alternative pre-mRNA splicing in digestive tract malignancy. *Cancer Sci.* 102 (2), 309–316.
- Nilsen, T.W., Graveley, B.R., 2010. Expansion of the eukaryotic proteome by alternative splicing. *Nature* 463, 457–463.
- Omenn, G.S., Yocum, A.K., Menon, R., 2010. Alternative splice variants, a new class of protein cancer biomarker candidates: Findings in pancreatic cancer and breast cancer with systems biology implications. *Dis. Markers* 28 (4), 241–251.
- Pan, Q., *et al.*, 2008. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nat. Genet.* 40, 1413–1415.
- Pandya-Jones, A., 2011. Pre-mRNA splicing during transcription in the mammalian system. *Wiley Interdiscip. Rev. RNA* 2 (5), 700–717.
- Patel, A.A., Steitz, J.A., 2003. Splicing double: Insights from the second spliceosome. *Nat. Rev. Mol. Cell Biol.* 4, 960.
- Paz, I., *et al.*, 2010. SFmap: A web server for motif analysis and prediction of splicing factor binding sites. *Nucleic Acids Res.* 38 (Web Server issue), W281–W285.
- Peled-Zehavi, H., *et al.*, 2001. Recognition of RNA branch point sequences by the KH domain of splicing factor 1 (mammalian branch point binding protein) in a splicing factor complex. *Mol. Cell. Biol.* 21 (15), 5232–5241.
- Pfarr, N., *et al.*, 2005. Linking C5 deficiency to an exonic splicing enhancer mutation. *J. Immunol.* 174 (7), 4172–4177.
- Poulos, M.G., *et al.*, 2011. Developments in RNA splicing and disease. *Cold Spring Harb. Perspect. Biol.* 3 (1), a000778.
- Raponi, M., *et al.*, 2011. Prediction of single-nucleotide substitutions that result in exon skipping: Identification of a splicing silencer in BRCA1 exon 6. *Hum. Mutat.* 32 (4), 436–444.
- Ren, S., *et al.*, 2012. RNA-seq analysis of prostate cancer in the Chinese population identifies recurrent gene fusions, cancer-associated long noncoding RNAs and aberrant alternative splicings. *Cell Res.* 22 (5), 806–821.
- Rogers, M.F., *et al.*, 2012. SpliceGrapher: Detecting patterns of alternative splicing from RNA-Seq data in the context of gene models and EST data. *Genome Biol.* 13 (1), R4.
- Rogozin, I.B., *et al.*, 2012. Origin and evolution of spliceosomal introns. *Biol. Direct* 7, 11–11.
- Rosenfeld, M.G., *et al.*, 1982. Calcitonin mRNA polymorphism: Peptide switching associated with alternative RNA splicing events. *Proc. Natl. Acad. Sci.* 79 (6), 1717–1721.
- Roy, S.W., Gilbert, W., 2006. The evolution of spliceosomal introns: Patterns, puzzles and progress. *Nat. Rev. Genet.* 7 (3), 211–221.
- Roy, B., Haupt, L.M., Griffiths, L.R., 2013. Review: Alternative splicing (AS) of genes as an approach for generating protein complexity. *Curr. Genom.* 14 (3), 182–194.
- Ryan, M.C., *et al.*, 2012. SpliceSeq: A resource for analysis and visualization of RNA-Seq data on alternative splicing and its functional impacts. *Bioinformatics* 28 (18), 2385–2387.
- Sammeth, M., 2009. Complete alternative splicing events are bubbles in splicing graphs. *J. Comput. Biol.* 16 (8), 1117–1140.
- Sammeth, M., Foissac, S., Guigó, R., 2008. A general definition and nomenclature for alternative splicing events. *PLOS Comput. Biol.* 4 (8), e1000147.
- Sampath, J., Pelus, L.M., 2007. Alternative splice variants of survivin as potential targets in cancer. *Curr. Drug Discov. Technol.* 4 (3), 174–191.
- Schmidt, F.J., 1985. RNA splicing in prokaryotes: Bacteriophage T4 leads the way. *Cell* 41 (2), 339–340.
- Schreiber, K., *et al.*, 2015. Alternative splicing in next generation sequencing data of *saccharomyces cerevisiae*. *PLOS ONE* 10 (10), e0140487.
- Schwartz, S., Hall, E., Ast, G., 2009. SROOGLE: Webserver for integrative, user-friendly visualization of splicing signals. *Nucleic Acids Res.* 37 (Web Server issue), W189–W192.
- Shapiro, I.M., *et al.*, 2011. An EMT – Driven alternative splicing program occurs in human breast cancer and modulates cellular phenotype. *PLOS Genet.* 7 (8), e1002218.
- Shen, S., *et al.*, 2012. MATS: A Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data. *Nucleic Acids Res.* 40 (8), e61.
- Shen, S., *et al.*, 2014. rMATS: Robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proc. Natl. Acad. Sci. USA* 111 (51), E5593–E5601.
- Song, L., Sabuncuyan, S., Florea, L., 2016. CLASS2: Accurate and efficient splice variant annotation from RNA-seq reads. *Nucleic Acids Res.* 44 (10), e98.
- Sorek, R., Shamir, R., Ast, G., 2004. How prevalent is functional alternative splicing in the human genome? *Trends Genet.* 20 (2), 68–71.
- Soukari, O., *et al.*, 2016. Exonic splicing mutations are more prevalent than currently estimated and can be predicted by using in silico tools. *PLOS Genet.* 12 (1), e1005756.
- Srivastava, S., Chen, L., 2010. A two-parameter generalized Poisson model to improve the analysis of RNA-seq data. *Nucleic Acids Res.* 38 (17), e170.
- Staley, J.P., Guthrie, C., 1998. Mechanical devices of the spliceosome: Motors, clocks, springs, and things. *Cell* 92 (3), 315–326.
- Stamm, S., *et al.*, 2000. An alternative-exon database and its statistical analysis. *DNA Cell Biol.* 19 (12), 739–756.
- Sultan, M., *et al.*, 2008. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* 321 (5891), 956–960.
- Tang, J.Y., *et al.*, 2013. Alternative splicing for diseases, cancers, drugs, and databases. *Sci. World J.* 2013, 703568.
- Tilgner, H., *et al.*, 2012. Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs. *Genome Res.* 22 (9), 1616–1625.
- Trowitzsch, S., *et al.*, 2009. Crystal structure of the Pml1p subunit of the yeast precursor mRNA retention and splicing complex. *J. Mol. Biol.* 385 (2), 531–541.
- van den Hoogenhof, M.M., Pinto, Y.M., Creemers, E.E., 2016. RNA splicing: Regulation and dysregulation in the heart. *Circ. Res.* 118 (3), 454–468.
- van Nimwegen, E., *et al.*, 2006. SPA: A probabilistic algorithm for spliced alignment. *PLOS Genet.* 2 (4), e24.
- Vitting-Seerup, K., *et al.*, 2014. spliceR: An R package for classification of alternative splicing and prediction of coding potential from RNA-seq data. *BMC Bioinform.* 15, 81.
- Volanakis, A., *et al.*, 2013. Spliceosome-mediated decay (SMD) regulates expression of nonintron genes in budding yeast. *Genes Dev.* 27 (18), 2025–2038.

- Wang, E.T., *et al.*, 2008. Alternative isoform regulation in human tissue transcriptomes. *Nature* 456, 470–476.
- Wang, J., *et al.*, 2017. Computational methods and correlation of exon-skipping events with splicing, transcription, and epigenetic factors. *Methods Mol. Biol.* (Clifton, N.J.) 1513, 163–170.
- Wang, J., *et al.*, 2005. Distribution of SR protein exonic splicing enhancer motifs in human protein-coding genes. *Nucleic Acids Res.* 33 (16), 5053–5062.
- Wang, M., Marin, A., 2006. Characterization and prediction of alternative splice sites. *Gene* 366 (2), 219–227.
- Wang, W., *et al.*, 2013. Identifying differentially spliced genes from two groups of RNA-seq samples. *Gene* 518 (1), 164–170.
- Wang, X., *et al.*, 2009. Genome-wide prediction of cis-acting RNA elements regulating tissue-specific pre-mRNA alternative splicing. *BMC Genom.* 10 (1), S4.
- Wang, Z., Burge, C.B., 2008. Splicing regulation: From a parts list of regulatory elements to an integrated splicing code. *RNA* 14 (5), 802–813.
- Will, C.L., Lührmann, R., 2011. Spliceosome structure and function. *Cold Spring Harb. Perspect. Biol.* 3 (7).
- Woodson, S.A., 1998. Ironing out the kinks: Splicing and translation in bacteria. *Genes Dev.* 12 (9), 1243–1247.
- Wu, J., *et al.*, 2011. SpliceTrap: A method to quantify alternative splicing under single cellular conditions. *Bioinformatics* 27 (21), 3010–3016.
- Wu, T.D., Watanabe, C.K., 2005. GMAP: A genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* 21 (9), 1859–1875.
- Xie, H., *et al.*, 2002. Computational analysis of alternative splicing using EST tissue information. *Genomics* 80 (3), 326–330.
- Xu, Q., Modrek, B., Lee, C., 2002. Genome-wide detection of tissue-specific alternative splicing in the human transcriptome. *Nucleic Acids Res.* 30 (17), 3754–3766.
- Yap, K., Makeyev, E.V., 2013. Regulation of gene expression in mammalian nervous system through alternative pre-mRNA splicing coupled with RNA quality control mechanisms. *Mol. Cell. Neurosci.* 56, 420–428.
- Yeo, G., *et al.*, 2004. Variation in alternative splicing across human tissues. *Genome Biol.* 5 (10), R74.
- Yeo, G.W., *et al.*, 2007. Alternative splicing events identified in human embryonic stem cells and neural progenitors. *PLoS Comput. Biol.* 3 (10), 1951–1967.
- Yeo, G., Burge, C.B., 2004. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. *J. Comput. Biol.* 11 (2–3), 377–394.
- Yi, Q., Tang, L., 2011. Alternative spliced variants as biomarkers of colorectal cancer. *Curr. Drug Metab.* 12 (10), 966–974.
- Zahler, A.M., Tuttle, J.D., Chisholm, A.D., 2004. Genetic suppression of intronic +1G mutations by compensatory U1 snRNA changes in *Caenorhabditis elegans*. *Genetics* 167 (4), 1689–1696.
- Zatkova, A., *et al.*, 2004. Disruption of exonic splicing enhancer elements is the principal cause of exon skipping associated with seven nonsense or missense alleles of NF1. *Hum. Mutat.* 24 (6), 491–501.
- Zhang, X., *et al.*, 2016. Recognition of alternatively spliced cassette exons based on a hybrid model. *Biochem. Biophys. Res. Commun.* 471 (3), 368–372.
- Zhao, K., *et al.*, 2013. GLIMMPS: Robust statistical model for regulatory variation of alternative splicing using RNA-seq data. *Genome Biol.* 14 (7), R74.
- Zheng, C.L., Fu, X.-D., Gribskov, M., 2005. Characteristics and regulatory elements defining constitutive splicing and different modes of alternative splicing in human and mouse. *RNA* 11 (12), 1777–1787.

Relevant Websites

- <http://www.altanalyze.org>
AltAnalyze.
- <http://ccb.jhu.edu/software/ASprofile>
ASprofile
CCB.
- <http://genome.imim.es/astalavista>
AStalavista.
- <http://www.domaingraph.de>
DomainGraph.
- <http://sammeth.net/confluence/display/ASTA/2+-+Download>
2
Download
AStalavista
Confluence.
- <http://www.netlab.uky.edu/p/bioinfo/DiffSplice>
DiffSplice.
- <http://bioinfo.au.tsinghua.edu.cn/software/DSGseq>
DSGseq.
- <https://sourceforge.net/p/finesplice/>
FineSplice.
- <https://github.com/anbrooks/juncBASE>
GitHub
anbrooks/juncBASE.
- http://compbio.uthscsa.edu/GESS_Web/
GESS-RNA.
- rnaseq-mats.sourceforge.net/
Multivariate Analysis of Transcript Splicing.
- <http://www.ncbi.nlm.nih.gov/snp>
NCBI
NIH.
- <http://www-rcf.usc.edu/~liangche/software.html>
GPSeq.
- <https://pypi.python.org/pypi/misopy/>
Python Package Index.
- <http://variome.kobic.re.kr/ssSNPTarget/>
ssSNPTarget.
- <http://cancan.cshl.edu/splicetrap>
SpliceTrap.
- <http://rulai.cshl.edu/splicetrap/>
SpliceTrap.

<http://bioinformatics.mdanderson.org/main/SpliceSeq:Overview>
 SpliceSeq:Overview.
<http://bioinformatics.zj.cn/splicingviewer>
 SpliceSeq:Overview.
<https://codeload.github.com/Xinglab/GLiMMPS/zip/master>
 GLiMMPS.
<http://www.bioconductor.org/packages/2.13/bioc/html/spliceR.html>
 SpliceR
 Bioconductor.

Biographical Sketch



Praseon Kumar Thakur is presently working as a bioinformatician at the laboratory of RNA Biology at the Institute of Molecular genetics, Prague. He is pursuing a PhD degree in Charles University, Prague, Czech Republic. He has a bachelor's degree in Bioinformatics from the Institute of Advance Studies in Education, Sardarshahr (IASES), India and M.Sc (Bioinformatics) from the Jamia Millia Islamia, New Delhi, India. The focus of his current research is RNA splicing and high-throughput data analysis.



Hukam C. Rawal is presently working as a Research Associate at NRCPB, New Delhi, India. He has over 10 years' experience in different aspects of computational biology. He has good exposure to a wide range of computational approaches for the analysis related to human disease, pathogen infection and crop sciences. He has expertise on advance and popular genomics studies and has worked on different aspects of comparative genomics. He is quite familiar with high throughput sequencing data. He has several publications in the field of comparative genomics, transcriptome assembly, genome level analysis, chloroplast and mitochondrial genome in reputed journals like Genome Biology & Evolution, Genes, Scientific Reports, etc.



Mina Obuca finished her bachelor and master studies in the Faculty of Sciences at the University of Novi Sad, Serbia. Currently, she is a PhD student at the Charles University in Prague, Czech Republic where she is doing her PhD at the Institute of Molecular Genetics in the Laboratory of RNA Biology. She is interested in answering why mutations in splicing proteins are causing retinitis pigmentosa.



Sandeep Kaushik is a passionate computational biologist with a wide exposure of computational approaches. Presently, he is presently carrying out his research as an Assistant Researcher (equivalent to Assistant Professor) at 3B's Research Group, University of Minho, Portugal. He has a PhD in Bioinformatics from National Institute of Immunology, New Delhi, India. He has gained a wide exposure on analysis of scientific data ranging from transcriptomic data on mycobacteria and human samples to genomic data from wheat. His research experience and expertise entails molecular dynamics simulations, RNA-sequencing data analysis, *de novo* genome assembly, protein prediction and annotation, database mining, agent-based modeling and simulations. He has a cumulative experience of more than 10 years of programming (using PERL, R and NetLogo languages). He has published his research in reputed international journals like Molecular Cell, Biomaterials, Biophysical Journal and others.

Genome-Wide Association Studies

William S Bush, Case Western Reserve University, Cleveland, OH, United States

© 2019 Elsevier Inc. All rights reserved.

Glossary

Common disease/common variant hypothesis The concept that genetic susceptibility to diseases common in a population is likely due to genetic variation that is also common in that population.

Direct association An association between a trait and SNP that is directly genotyped in a study and itself induces an influential biological change.

False positive A result that is statistically significant when the null hypothesis is true.

Genome-wide significance A significance threshold used to account for multiple hypothesis tests conducted in a GWAS which corresponds to the number of effective independent regions in the human genome.

Genotype imputation The use of population-based genome reference panels to infer the state of ungenotyped variants from GWAS data.

Indirect association An association between a trait and a SNP that is in linkage disequilibrium with another nearby SNP that induces an influential biological change.

Linkage disequilibrium The genetic correlation among alleles of co-located SNPs within the genome that is due to

limited recombination events between those SNPs within a population.

Manhattan plot A plot illustrating the magnitude and genomic location of GWAS associations.

Meta-analysis An approach for combining the results of multiple independent studies by analyzing summary statistics.

Population stratification An effect where many SNPs within a study are systematically associated with a trait due to differences in both allele frequency and trait distribution across genetic ancestries.

P-value The probability of observing a value as extreme or more extreme than the test statistic when the null hypothesis is true.

QQ-plot A diagnostic plot used in GWAS to identify systematic inflation of test statistics across all SNPs.

Single nucleotide polymorphism A single base-pair change to the DNA sequence that is common within a population.

TagSNPs A SNP that is used to predict the state of other nearby SNPs via linkage disequilibrium.

Introduction: The Common Disease-Common Variant Hypothesis

Early successes in human genetics led to the discovery of disease genes for a variety of inherited disorders such as cystic fibrosis (Rommens *et al.*, 1989) and Huntington disease (MacDonald *et al.*, 1992). These Mendelian diseases were studied by genotyping families affected by the disease using a collection of genetic markers across the genome, and examining how those genetic markers segregate with the disease across multiple families (Bush and Haines, 2010). This approach, called linkage analysis, was largely unsuccessful when applied to more common disorders, like heart disease, type 2 diabetes, or stroke, implying that the genetic architecture of commonly occurring diseases is likely different from that of rare disorders (Hirschhorn and Daly, 2005). This observation, along with the discovery of some relatively frequent risk alleles for common diseases such as *APOE* alleles for Alzheimer's Disease (Corder *et al.*, 1993) and *PPARG* alleles for type II diabetes (Lander *et al.*, 2000), led to the formalization of a *common disease/common variant* (CD/CV) hypothesis (Reich and Lander, 2001), which simply states that disorders which are common in a population are likely influenced by genetic variation which is also common in that population. To test the CD/CV hypothesis, a new type of population-based study was developed, Genome-Wide Association Studies (GWAS). GWAS was designed to broadly capture common genetic variation, generally defined as DNA base-pair changes with a minor allele frequency (MAF) greater than 5% for the population being studied. Before GWAS could be effectively implemented, however, a catalog of common variation assessed across multiple populations was needed as a reference.

Genetic Variation and Linkage Disequilibrium

Through a large-scale coordinated effort, the International HapMap Project (The International HapMap Project, 2003) captured much of the common variation among 11 human subpopulations, and established the groundwork for GWAS (Manolio *et al.*, 2008). The HapMap Project focused primarily on characterizing *single nucleotide polymorphisms* (SNPs), a single base-pair change to the DNA sequence. The project also produced extensive maps of *linkage disequilibrium* (LD), a property of SNPs that lie on a single chromosome and are co-inherited through the transmission of the chromosome within a population. Humans are diploid, having two copies of each chromosome (with exceptions for X and Y), and recombination events break apart contiguous stretches of SNPs on a chromosome during meiosis when germ cells are created. Through successive generations within a population, repeated recombination events distributed across each chromosome produce stretches of alleles that are correlated to one another. The

degree to which LD exists for a given chromosomal region depends on many factors, including the population size, the number of generations for which the population has existed, the recombination rate, and the number of potential recombination points along the chromosome. The influence of these factors is clear from examining different global populations; African-descent populations on average have smaller regions of LD compared to European-descent or East Asian-descent populations, which were the result of a series of founder events and as a result have larger regions of LD.

Maps of LD were instrumental in the initial design of many fluorescence-based genotyping arrays, which relied on the use of *tagSNPs*, specific variants that were highly correlated (and predictive of) other nearby sites. The use of tagSNPs prevented genotyping SNPs that essentially provide redundant information, and allowed the capture of greater than 80% of all common SNPs in Europeans using 500,000 to 1 Million SNPs from across the genome (Li *et al.*, 2008). Notably, as LD patterns are population-dependent, early genotyping arrays were potentially biased toward European-descent populations as different tagSNPs were needed for optimal capture of variation in other populations.

While tagSNPs greatly improved the capture of genetic information, they also limit any causal inference that can be made from GWAS. When a tagSNP is associated to a disease, this association can either be a *direct association*, where the genotyped tagSNP is the true variant inducing a functional effect, or it may be an *indirect association*, where the true disease-associated variant is in high LD with the genotyped tagSNP. As a result, significant SNP associations from GWAS are not always the true functional/causal variant, and additional fine-mapping studies may be necessary to elucidate the true disease variant.

GWAS Experimental Designs

A critical first step for any genetic study is to precisely define the trait or outcome thought to be influenced by genetic variation. Though there are some exceptions, GWAS generally examine either case/control or continuous outcomes. Continuous, quantitative traits are often preferred as they increase statistical power to detect a genetic effect, and provide a unit change per risk allele. Quantitative traits such as body mass index (BMI) and low-density lipoprotein (LDL) levels are routinely collected through standardized procedures and represent intermediate outcomes for more severe conditions such as heart disease. However, quantitative traits are not always easy to define or collect for certain disorders. For these conditions, dichotomous or case/control categorical variables are used as the primary outcome. Categorical variables however are more susceptible to problematic definitions of “cases” and “controls”, and often require adjudication by clinical experts. For many conditions, consensus criteria have been established for defining case/control status, such as the McDonald criteria for multiple sclerosis (Polman *et al.*, 2011), or the NINCDS-ADRDA criteria for Alzheimer’s disease (McKhann *et al.*, 1984). These standardized, evidence-based approaches can be uniformly applied by multiple diagnosing clinicians to ensure that consistent phenotype definitions are used for a genetic study.

Standardized criteria for study phenotypes are especially critical for multi-center studies. If different case/control definitions are used at different study sites, a site-based effect may be introduced into the study. Even when standardized criteria are used across all study sites, there may still be variability among how clinicians interpret them to assign case/control status. While generally thought to be more accurate, quantitative traits are also susceptible to bias in measurement. For example, in studies of cataract severity, lens (eye) photography is used to assign cataract cases to one of three grades of lens opacity. These photographs are adjudicated by clinicians whose characterizations of cataract stage may not be consistent. Interrater agreement is often evaluated across multiple grading clinicians to measure phenotype consistency (Chew *et al.*, 2010). Significant disagreement between raters typically indicates that disease criteria are not being uniformly applied or interpreted, and a narrower set of phenotype rules are needed.

Once a phenotype has been selected and standardized for a particular study population, genetic information must be collected on study participants. While DNA can be extracted from any tissue, the most common source of DNA for GWAS is whole blood. An individual blood sample is separated into fractions via centrifugation to isolate the buffy coat, a layer between erythrocytes (which do not contain nuclei or DNA) and plasma. The buffy coat contains white blood cells which are subsequently lysed to capture participant DNA. DNA is then processed via vendor specifications for large-scale genotyping using microarray-based technologies, generally from either Illumina (San Diego, CA) or Affymetrix (Santa Clara, CA). These two technologies have relative strengths and weaknesses (DiStefano and Taverna, 2011), and other approaches exist to measure genetic variation, but are not generally used for primary GWAS data collection.

Statistical Analysis

The primary results for a typical GWAS are generated through systematic statistical evaluation of each genetic variant captured by the genotyping process. As part of conducting the statistical analysis for GWAS, genotype data must be encoded, and different encoding schemes reflect different underlying models of allelic effect, and have implications on the degrees of freedom and statistical power of a test. An allelic encoding assumes that each person in the dataset represents two separate chromosomes, and examines the association of the allele to the phenotype. A genotypic encoding assumes that each person is a distinct observation of a combination of two alleles, and genotypes can further be grouped into specific modes of action such as dominant, recessive, multiplicative, or additive models (Lewis, 2002). While these various encoding schemes exist and are used to different degrees in GWAS, it is common practice to examine additive genotypic models only, as this has the best statistical power to detect a variety of modes of action (Lettre *et al.*, 2007).

After choosing an encoding, each SNP is examined independently for association the study phenotype, and the statistical test employed depends multiple factors, most critically whether the phenotype is a quantitative trait or a case/control outcome. In either case, generalized linear models (GLM) are generally used. In the case of quantitative traits, Analysis of Variance (ANOVA) is used to test the deviation from the null hypothesis that there is no difference between the mean trait value across individuals by genotype. Assumptions of the ANOVA apply, most critically that trait follows a normal distribution and that the variance of the trait is the same across groups (the groups are homoscedastic). Case/control traits are typically analyzed using a variant of GLM, logistic regression, which transforms a linear outcome to a probability using a logit curve. While some early GWAS used contingency table analysis methods, such as a chi-square test, logistic regression is now the preferred approach as it allows for adjustment for other relevant factors and can provide adjusted effect estimates for each allele of a SNP. Logistic regression is a mature approach with well-established interpretations and diagnostic procedures to assess the goodness of model fit and the influence of potential outliers (Hosmer *et al.*, n.d.).

Each SNP-based statistical model should also be adjusted for other factors with established effects on the study trait. This often includes demographic factors (i.e., age, sex), clinical factors (i.e., body mass index, comorbidity status), and design factors (i.e., study site, experimental batch). Adjustment for covariates like these reduces artifacts in the statistical analysis that may differentially inflate the frequency of alleles/genotypes between cases and controls. By far the most common confounding factor in GWAS is population substructure. For nearly any phenotype, there are often known differences in the occurrence rate or prevalence across different racial, ethnic, and geographic backgrounds. Similarly, due to human migration patterns, the founding of new populations, and dramatic changes to population size, allele frequencies are highly variable across human subpopulations. Therefore, if a study collects individuals from cases and controls from different ancestral populations, any systematic differences in allele frequency between the ancestral populations will appear as differences between cases and controls – a phenomenon known as *population stratification*. In GWAS, population stratification is generally assessed by estimating the ancestry of each sample in the dataset using either sample clustering within the STRUCTURE software (Falush *et al.*, 2003), or using principal components implemented in the EIGENSTRAT (Price *et al.*, 2006), GCTA (Yang *et al.*, 2011), and PLINK 1.9 (Chang *et al.*, 2015) software packages. Principal component values can then be used as a covariate in statistical models to adjust for minute degrees of ancestry effects present in the study. Linear mixed models (also known as mixed linear models) are also now commonly used to adjust for stratification due to either population differences or familial relationships, including GCTA (Yang *et al.*, 2011), REACTA (Cebamano *et al.*, 2014), EMMAX (Kang *et al.*, 2010), FaST-LMM (Lippert *et al.*, 2011), GEMMA (Zhou and Stephens, 2012), and GRAMMAR-Gamma (Svishcheva *et al.*, 2012).

For each SNP, a statistical model including the encoded effect of the SNP alleles along with any covariate adjustments is evaluated. The effect (also referred to as the β or coefficient) for the encoded alleles is evaluated (by *T*-test in the case of linear regression or Wald test in the case of logistic regression) to produce a *p*-value corresponding to the significance of the SNP. Assuming a null hypothesis that there is no effect of the SNP on the phenotype, a *p*-value is the probability of observing a test statistic equal to or greater than the statistic reported from the analysis of the SNP in the GWAS data. In effect therefore, very small *p*-values indicate that if there is no real association between the SNP and the phenotype, observing the reported test statistic is extremely unlikely. Statistical tests like the *T*-test or Wald test generally conducted in GWAS are often labeled “significant”, meaning the null hypothesis of no association is rejected, if the *p*-value falls below a pre-defined significance threshold (often referred to as α). For most statistical tests, the α is generally pre-set to 0.05, meaning that for 5% of all statistical tests, the null hypothesis is rejected when it is in fact true – a phenomenon called a *false positive*. This 5% probability of a false positive is relative to a single statistical test, and in the case of GWAS, many thousands or even millions of individual hypothesis tests are conducted with each one having a probability of false positive. As a result, assuming α of 0.05, the cumulative probability of finding one or more false positives over a GWAS is much higher. Put a different way, assume you paint one side of a twenty-sided die red; a single roll of this die is unlikely to show the red side, but if you roll this die 500,000 times, it is extremely likely that you roll the red side at least once. The easiest way to adjust for the multiple statistical comparisons conducted in GWAS is to use a Bonferroni correction. This correction adjusts the alpha value for each statistical test from $\alpha=0.05$ to $\alpha=(0.05/k)$ where *k* is the number of statistical tests conducted. Assuming that there is some degree of linkage disequilibrium and thus correlation among statistical tests of a GWAS, this correction is rather conservative as it assumes completely independent tests. Perhaps the most commonly used approach is to assume a level of *genome-wide significance*, which is an estimate of the effective number of independent genomic regions and thus statistical tests for which the analysis should be corrected. This assessment is population specific, owing to differences in LD patterns among different ancestral populations, but in the case of European-descent studies, the genome-wide significance threshold has been estimated at $7.2e-8$ (Dudbridge and Gusnanto, 2008).

There are two commonly used data visualizations shown in GWAS publications, a quantile-quantile (QQ) plot, and a “Manhattan” plot, both implemented in the *qqman* R package (Turner, 2014). A QQ plot (Fig. 1) is a comparison of the *p*-value distribution across all evaluated SNPs to an expected theoretical uniform distribution. Values are typically plotted as the negative log (base 10) of the *p*-values, which provides a unit increase per order of magnitude decrease in the *p*-value (a value of 7 is equivalent to 1×10^{-7}). A QQ plot of log *p*-values is an easy way to visually identify systematic deviations from the null hypothesis – an indicator that other factors like population stratification produce systematic biases across all test statistics. In some cases, the deviation of observed *p*-values from the expected uniform distribution is quantified as a *genomic inflation factor*. A Manhattan plot (Fig. 2) is a similar visualization, showing the negative log *p*-value plotted by chromosome position. These plots often color-code chromosomes on the X-axis to help identify where significant SNPs lie in the genome. A Manhattan plot illustrates two properties of GWAS results, first the physical location of SNPs with extreme *p*-values, and secondly the degree to which a SNP association is corroborated by other nearby SNPs in linkage disequilibrium. Often these plots are labeled with additional information, such as a line demarking the statistical significance criteria used for the analysis, and previously established true positive SNP associations.

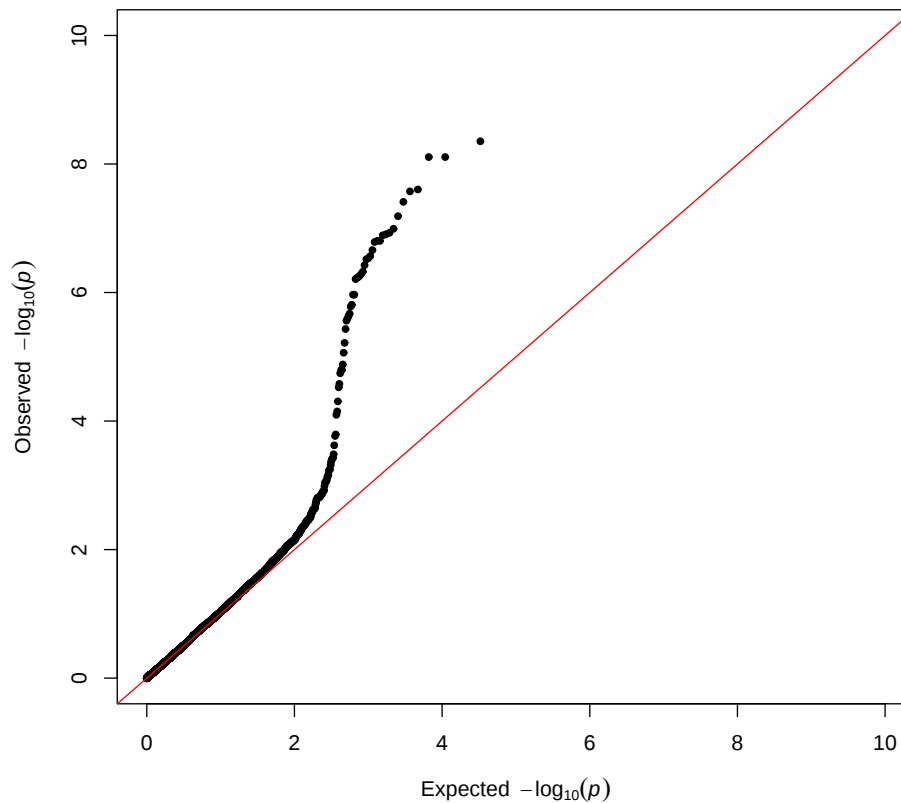


Fig. 1 A quantile-quantile plot of GWAS p-values. The negative log10 p-values for each analyzed SNP are plotted on the y-axis versus the expected p-values under the null hypothesis of no associated SNPs on the x-axis. Points deviating from the diagonal line are more significant than expected under the null hypothesis.

The vast majority of published GWAS were conducted using PLINK ([Chang et al., 2015](#); [Purcell et al., 2007](#)) software for statistical analysis. PLINK has numerous functions for conducting genotype and sample quality control, and for performing basic analyses of continuous and binary traits. As the number of genetic variants analyzed in GWAS studies have grown, genetic datasets are often split (by chromosome or other metrics) to facilitate more rapid processing. Analyzing data subsets, however, requires more detailed file and process management, and can introduce errors. To address these issues, software such as HAIL ([Hail, n.d.](#)) has been developed to enable distributed genomic data processing. HAIL provides much of the functionality of PLINK, but performs its operations over a HADOOP/SPARK distributed computing environment. By leveraging this memory and process rich environment, an entire GWAS analysis can be completed in minutes.

Meta-Analysis, Replication, and GWAS Follow-Up

Shortly after initial publications of GWAS for many traits, there was a growing realization within the genomics community that ever larger sample sizes are needed to improve statistical power for the detection of new risk variants. To reduce the need to share individual-level genomic data, research groups began to combine the statistical results from multiple independent GWAS through *meta-analysis*. Techniques for meta-analysis were originally developed to summarize statistical results from multiple independent published studies which examined the same hypothesis. When applied to GWAS, these approaches allow for multiple sets of genome-wide statistical results to be combined using sample size based or inverse variance based approaches as implemented in the software METAL ([Sanna et al., 2008](#); [Willer et al., 2008](#)), or the *metafor* R package, among others. GWAS meta-analysis can be challenging in practice. The multiple studies participating in a meta-analysis must take care to conduct their quality control procedures in a standardized way, include consistent covariate adjustments, and ensure that all statistical results are reported relative to a pre-defined reference allele for each SNP ([Zeggini and Ioannidis, 2009](#)). Even with standardization across studies, there is often some degree to which studies differ, including some planned differences in design and analysis. Heterogeneity across multiple studies participating in a GWAS meta-analysis is often assessed using either the Q statistic or the I^2 index ([Huedo-Medina et al., 2006](#)), with I^2 typically favored as it represents the approximate proportion of variability in a SNP coefficient attributed to heterogeneity between studies ([Higgins, 2008](#)). High degrees of heterogeneity may point to specific outlier studies that may be excluded for legitimate reasons, but may also simply point to unstable associations for a given SNP.

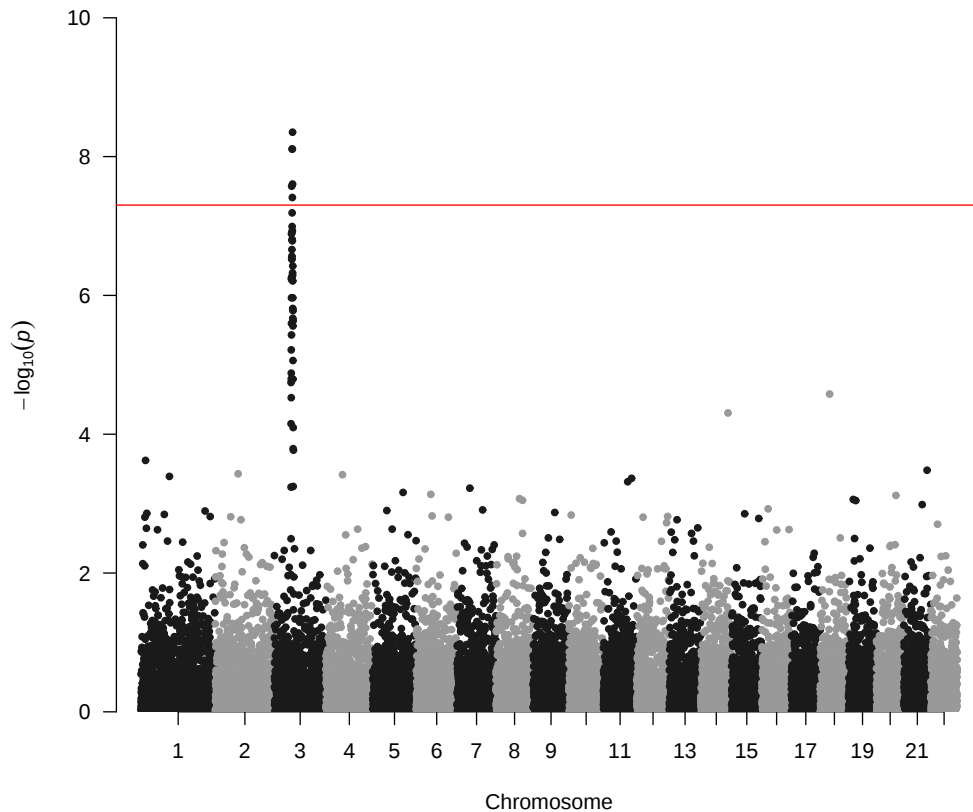


Fig. 2 GWAS Manhattan Plot. The y-axis represents the negative log10 of the p-value for each analyzed SNP. The x-axis illustrates the chromosome (note color coding) and position of each SNP. Horizontal lines mark different levels of statistical significance.

Meta-analyses of GWAS require estimates of consistent alleles from a consistent set of SNPs. In reality, different GWAS often use different genotyping platforms and technologies, leading to different collections of genotyped SNPs for each study. To resolve this issue, studies perform *genotype imputation*, which exploits known patterns of linkage disequilibrium from a selected reference panel to estimate the alleles of ungenotyped SNPs (Li *et al.*, 2009). Popular algorithms for genotype imputation include BimBam (Guan and Stephens, 2008), IMPUTE (Howie *et al.*, 2009), MaCH (Biernacka *et al.*, 2009), and Beagle (Browning and Browning, 2009). Genotype imputation produces an estimate for the allele state at each ungenotyped SNP for each individual, often reported as the probability of each possible allele. Allele probabilities for imputed SNPs should be used in the GWAS analysis to take into account the uncertainty of the genotype imputation process (Marchini *et al.*, 2007). Also, as with other GWAS design factors, population ancestry is of critical importance; the reference panel used in the imputation process must contain individuals drawn from the same population as the GWAS dataset.

Once a key set of SNP-trait associations are discovered from a GWAS, the next critical step is to replicate these associations in an additional independent sample (Chanock *et al.*, 2007). A well-designed replication study should have a sample size to detect the predicted effect of any discovered SNP – ideally large enough to detect an effect slightly smaller than the original (Zollner and Pritchard, 2007). Replication studies should be drawn from the same population as the original GWAS to confirm the effect within the target population. Studies in additional populations are sometimes conducted to determine if the effect of a SNP “generalizes” to other populations (Carlson *et al.*, 2013). Replication studies should use identical phenotype criteria to ensure a clear interpretation of the genetic results. Finally, analyses of replication datasets should show a similar magnitude and direction of effect across both studies.

Conclusions

Genome-wide association studies have played a critical role in identifying thousands of new SNP-trait associations. GWAS as a study design is sometimes considered as both a success and a failure. They are a failure in that even though tens or hundreds of SNPs associate to any given common disease, the collective effect of these SNPs explain very little of the disease risk that is known to be genetic (Maher, 2008). They are a success in that for many diseases, entirely new biological mechanisms of disease have been elucidated for common diseases which have long had little genetic understanding (Hirschhorn, 2009). While many large-scale GWAS have been conducted on many thousands of samples, ever larger studies, coupled with decreasing costs of genotyping and sequencing will likely yield an even better understanding of the genomics of human disease.

See also: Comparative Genomics Analysis. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Detecting and Annotating Rare Variants. Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Functional Genomics. Genetics and Population Analysis. Genome Databases and Browsers. Genome Informatics. Genome-Wide Haplotype Association Study. Integrative Analysis of Multi-Omics Data. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Population Analysis of Pharmacogenetic Polymorphisms. Quantitative Immunology by Data Analysis Using Mathematical Models. Single Nucleotide Polymorphism Typing. Whole Genome Sequencing Analysis

References

- Biernacka, J.M., Tang, R., Li, J., *et al.*, 2009. Assessment of genotype imputation methods. BMC Proceedings 3 (Suppl. 7), S5. Retrieved from: <http://www.ncbi.nlm.nih.gov/pubmed/20018042>.
- Browning, B.L., Browning, S.R., 2009. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. American Journal of Human Genetics 84 (2), 210–223. Available at: <https://doi.org/10.1016/j.ajhg.2009.01.005>.
- Bush, W.S., Haines, J., 2010. Overview of linkage analysis in complex traits. Current Protocols in Human Genetics 64 (1), Chapter 1, Unit 1.9.1–18. Available at: <https://doi.org/10.1002/0471142905.hg0109s64>.
- Carlson, C.S., Matise, T.C., North, K.E., *et al.* PAGE Consortium, 2013. Generalization and dilution of association results from European GWAS in populations of non-European ancestry: The PAGE study. PLOS Biology 11 (9), e1001661. Available at: <https://doi.org/10.1371/journal.pbio.1001661>.
- Cebamanos, L., Gray, A., Stewart, I., Tenesa, A., 2014. Regional heritability advanced complex trait analysis for GPU and traditional parallel architectures. Bioinformatics (Oxford, England) 30 (8), 1177–1179. Available at: <https://doi.org/10.1093/bioinformatics/btt754>.
- Chang, C.C., Chow, C.C., Tellier, L.C., *et al.*, 2015. Second-generation PLINK: Rising to the challenge of larger and richer datasets. GigaScience 4 (1), 7. Available at: <https://doi.org/10.1186/s13742-015-0047-8>.
- Chanock, S.J., Manolio, T., Boehnke, M., *et al.*, 2007. Replicating genotype–phenotype associations. Nature 447 (7145), 655–660. Available at: <https://doi.org/10.1038/447655a>.
- Chew, E.Y., Kim, J., Sperduto, R.D., *et al.*, 2010. Evaluation of the age-related eye disease study clinical lens grading system AREDS report No. 31. Ophthalmology 117 (11), 2112–2119. e3. Available at: <https://doi.org/10.1016/j.ophtha.2010.02.033>.
- Corder, E.H., Saunders, A.M., Strittmatter, W.J., *et al.*, 1993. Gene dose of apolipoprotein E type 4 allele and the risk of Alzheimer's disease in late onset families. Science (New York, N.Y.) 261 (5123), 921–923. Retrieved from: <http://www.ncbi.nlm.nih.gov/pubmed/8346443>.
- DiStefano, J.K., Taverna, D.M., 2011. Technological issues and experimental design of gene association studies. Methods in Molecular Biology (Clifton, N.J.) 700, 3–16. Available at: https://doi.org/10.1007/978-1-61737-954-3_1.
- Dudbridge, F., Gusnanto, A., 2008. Estimation of significance thresholds for genomewide association scans. Genetic Epidemiology 32 (3), 227–234. Available at: <https://doi.org/10.1002/gepi.20297>.
- Falush, D., Stephens, M., Pritchard, J.K., 2003. Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies. Genetics 164 (4), 1567–1587. Retrieved from: <http://www.ncbi.nlm.nih.gov/pubmed/12930761>.
- Guan, Y., Stephens, M., 2008. Practical issues in imputation-based association mapping. PLOS Genetics 4 (12), e1000279. Available at: <https://doi.org/10.1371/journal.pgen.1000279>.
- Hail (n.d.). Retrieved from: <https://github.com/hail-is/hail>.
- Higgins, J.P.T., 2008. Commentary: Heterogeneity in meta-analysis should be expected and appropriately quantified. International Journal of Epidemiology 37 (5), 1158–1160. Available at: <https://doi.org/10.1093/ije/dyn204>.
- Hirschhorn, J.N., 2009. Genomewide association studies – Illuminating biologic pathways. New England Journal of Medicine 360 (17), 1699–1701. Available at: <https://doi.org/10.1056/NEJMp0808934>.
- Hirschhorn, J.N., Daly, M.J., 2005. Genome-wide association studies for common diseases and complex traits. Nature Reviews Genetics 6 (2), 95–108. Available at: <https://doi.org/10.1038/nrg1521>.
- Hosmer, D.W., Lemeshow, S., Sturdivant, R.X. (n.d.). Applied logistic regression. Retrieved from: https://books.google.co.in/books?hl=en&lr=&id=64JYAwAAQBAJ&oi=fnd&pg=PA313&dq=HOSMER+LEMESHOW+logistic+regression&ots=Dscl3YathK&sig=52_KJ7diR7s39CH4MD3idTdLzd8&redir_esc=y#v=onepage&q=HOSMER+LEMESHOW+logistic+regression&f=false.
- Howie, B.N., Donnelly, P., Marchini, J., 2009. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLOS Genetics 5 (6), e1000529. Available at: <https://doi.org/10.1371/journal.pgen.1000529>.
- Huedo-Medina, T.B., Sánchez-Meca, J., Marín-Martínez, F., Botella, J., 2006. Assessing heterogeneity in meta-analysis: Q statistic or I² index? Psychological Methods 11 (2), 193–206. Available at: <https://doi.org/10.1037/1082-989X.11.2.193>.
- Kang, H.M., Sul, J.H., Service, S.K., *et al.*, 2010. Variance component model to account for sample structure in genome-wide association studies. Nature Genetics 42 (4), 348–354. Available at: <https://doi.org/10.1038/ng.548>.
- Lander, E.S., Altshuler, D., Hirschhorn, J.N., *et al.*, 2000. The common PPARgamma Pro12Ala polymorphism is associated with decreased risk of type 2 diabetes. Nature Genetics 26 (1), 76–80. Available at: <https://doi.org/10.1038/79216>.
- Lettre, G., Lange, C., Hirschhorn, J.N., 2007. Genetic model testing and statistical power in population-based association studies of quantitative traits. Genetic Epidemiology 31 (4), 358–362. Available at: <https://doi.org/10.1002/gepi.20217>.
- Lewis, C.M., 2002. Genetic association studies: Design, analysis and interpretation. Briefings in Bioinformatics 3 (2), 146–153. Retrieved from: <http://www.ncbi.nlm.nih.gov/pubmed/12139434>.
- Li, M., Li, C., Guan, W., 2008. Evaluation of coverage variation of SNP chips for genome-wide association studies. European Journal of Human Genetics 16 (5), 635–643. Available at: <https://doi.org/10.1038/sj.ejhg.5202007>.
- Li, Y., Willer, C., Sanna, S., Abecasis, G., 2009. Genotype imputation. Annual Review of Genomics and Human Genetics 10 (1), 387–406. Available at: <https://doi.org/10.1146/annurev.genom.9.081307.164242>.
- Lippert, C., Listgarten, J., Liu, Y., *et al.*, 2011. FaST linear mixed models for genome-wide association studies. Nature Methods 8 (10), 833–835. Available at: <https://doi.org/10.1038/nmeth.1681>.
- MacDonald, M.E., Novelletto, A., Lin, C., *et al.*, 1992. The Huntington's disease candidate region exhibits many different haplotypes. Nature Genetics 1 (2), 99–103. Available at: <https://doi.org/10.1038/ng0592-99>.
- Maier, B., 2008. Personal genomes: The case of the missing heritability. Nature 456 (7218), 18–21. Available at: <https://doi.org/10.1038/456018a>.
- Manolio, T.A., Brooks, L.D., Collins, F.S., 2008. A HapMap harvest of insights into the genetics of common disease. The Journal of Clinical Investigation 118 (5), 1590–1605. Available at: <https://doi.org/10.1172/JCI34772>.

- Marchini, J., Howie, B., Myers, S., McVean, G., Donnelly, P., 2007. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics* 39 (7), 906–913. Available at: <https://doi.org/10.1038/ng2088>.
- McKhann, G., Drachman, D., Folstein, M., *et al.*, 1984. Clinical diagnosis of Alzheimer's disease: Report of the NINCDS-ADRDA Work Group under the auspices of Department of Health and Human Services Task Force on Alzheimer's Disease. *Neurology* 34 (7), 939–944. Retrieved from: <http://www.ncbi.nlm.nih.gov/pubmed/6610841>.
- Polman, C.H., Reingold, S.C., Banwell, B., *et al.*, 2011. Diagnostic criteria for multiple sclerosis: 2010 Revisions to the McDonald criteria. *Annals of Neurology* 69 (2), 292–302. Available at: <https://doi.org/10.1002/ana.22366>.
- Price, A.L., Patterson, N.J., Plenge, R.M., *et al.*, 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* 38 (8), 904–909. Available at: <https://doi.org/10.1038/ng1847>.
- Purcell, S., Neale, B., Todd-Brown, K., *et al.*, 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *American Journal of Human Genetics* 81 (3), 559–575. Available at: <https://doi.org/10.1086/519795>.
- Reich, D.E., Lander, E.S., 2001. On the allelic spectrum of human disease. *Trends in Genetics* 17 (9), 502–510. Retrieved from: <http://www.ncbi.nlm.nih.gov/pubmed/11525833>.
- Rommens, J., Iannuzzi, M., Kerem, B., *et al.*, 1989. Identification of the cystic fibrosis gene: Chromosome walking and jumping. *Science* 245 (4922), 1059–1065. Available at: <https://doi.org/10.1126/science.2772657>.
- Sanna, S., Jackson, A.U., Nagaraja, R., *et al.*, 2008. Common variants in the GDF5-UQCC region are associated with variation in human height. *Nature Genetics* 40 (2), 198–203. Available at: <https://doi.org/10.1038/ng.74>.
- Svishcheva, G.R., Axenovich, T.I., Belonogova, N.M., van Duijn, C.M., Aulchenko, Y.S., 2012. Rapid variance components-based method for whole-genome association analysis. *Nature Genetics* 44 (10), 1166–1170. Available at: <https://doi.org/10.1038/ng.2410>.
- The International HapMap Project, 2003. *Nature* 426 (6968), 789–796. Available at: <https://doi.org/10.1038/nature02168>.
- Turner, S.D., 2014. qqman: An R package for visualizing GWAS results using Q-Q and manhattan plots. *bioRxiv*. 5165. Available at: <https://doi.org/10.1101/005165>.
- Willer, C.J., Sanna, S., Jackson, A.U., *et al.*, 2008. Newly identified loci that influence lipid concentrations and risk of coronary artery disease. *Nature Genetics* 40 (2), 161–169. Available at: <https://doi.org/10.1038/ng.76>.
- Yang, J., Lee, S.H., Goddard, M.E., Visscher, P.M., 2011. GCTA: A tool for genome-wide complex trait analysis. *American Journal of Human Genetics* 88 (1), 76–82. Available at: <https://doi.org/10.1016/j.ajhg.2010.11.011>.
- Zeggini, E., Ioannidis, J.P.A., 2009. Meta-analysis in genome-wide association studies. *Pharmacogenomics* 10 (2), 191–201. Available at: <https://doi.org/10.2217/14622416.10.2.191>.
- Zhou, X., Stephens, M., 2012. Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics* 44 (7), 821–824. Available at: <https://doi.org/10.1038/ng.2310>.
- Zöllner, S., Pritchard, J.K., 2007. Overcoming the winner's curse: Estimating penetrance parameters from case-control data. *American Journal of Human Genetics* 80 (4), 605–615. Available at: <https://doi.org/10.1086/512821>.

Gene Mapping

Kelly L Williams, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Gene mapping is the sequential allocation of loci to a relative position on a chromosome. Genetic maps are species-specific and comprised of genomic markers and/or genes and the genetic distance between each marker. These distances are calculated based on the frequency of chromosome crossovers occurring during meiosis, and not on their physical location on the chromosome. There are existing dense genetic marker maps available for humans, and the introduction of next-generation sequencing technologies is facilitating increased construction of genetic maps for other species. Genetic maps are a necessary tool for mapping of disease genes or trait loci, a method also commonly known as linkage mapping. Integrating genetic mapping and disease gene mapping with next-generation sequencing has proven to be a powerful strategy in genetic research.

From Meiosis to centiMorgans

Observation of the frequent co-inheritance of traits in *Drosophila* led Morgan, one of the pioneers of genetic research, to first describe the concept of linkage in 1911 as “associations of factors” that are located near together in the chromosomes” (Morgan, 1911). Morgan’s student Sturtevant, another pioneer in genetic research, extended these observations and determined that the proportion of recombinant gametes resulting from crossovers during meiosis allows for the calculation of genetic distance between two loci in *Drosophila* (Sturtevant, 1913). Chromosomal crossing-over is the biological phenomenon occurring during meiosis that underpins genetic variation in populations. No two gametes produced during meiosis are the same. In the process of meiosis, the paternal and maternal inherited chromosomes replicate to form sister chromatids. During prophase I, homologous chromosomes are paired in the cell. Then, two non-sister chromatids undergo recombination (or ‘crossing-over’) where homologous portions of the chromatids exchange places. This results in recombinant chromatids (which are a combination of the maternal and paternal chromosomes) and non-recombinant chromatids (identical to the parental chromosomes) per chromosome (Fig. 1). The recombination fraction (θ) is the probability of a crossing over event occurring between two loci. Crossovers occur semi-randomly throughout the genome and thus allow genetic maps to be constructed.

Genetic distance between two loci is measured in Morgans (M) and is the expected number of crossovers to occur between those two loci on a chromosome in a gamete. A centiMorgan (cM) is a more commonly used measure of distance and corresponds to $\frac{1}{100}$ of a Morgan. Based on Sturtevant’s seminal work, if two loci are unlinked (for example on difference chromosomes), we would expect to see 50% recombinant gametes, and 50% non-recombinant gametes. However if we see lower percentage of recombinant gametes that differs to the expected, such as 13% recombinants, the two loci are considered linked at a genetic distance of 13 cM. One cM roughly corresponds to 1 Mb (1 megabase, or 1,000,000 base pairs), but differs substantially throughout the genome. Recombination rate can vary from 0.1 cM Mb⁻¹ to 3 cM Mb⁻¹ across a chromosome (Kong et al., 2002). Females display a higher crossover frequency than males, therefore the human female genetic map is approximately 44 Morgans, compared to 27 Morgans for males (Broman et al., 1998; Table 1). Dense genetic maps are usually released in three versions: male, female and sex-averaged. Table 1 outlines the map distances and physical distance of each chromosome.

Evolution of Genetic Markers and Human Genetic Maps

The Human Genome Project represented a time of enormous progress in human genetics, and resulted in the construction of dense genetic maps that are still in use by geneticists today. Prior to the publication of the first draft human genome sequence completed under the Human Genome Project, all genetic maps were constructed without having a reference genome sequence. Simply determining the order of genetic markers was a considerable and time-consuming task. The first genetic map of the human genome comprised 403 polymorphic markers, the majority of which were low-informative bi-allelic RFLPs (restriction fragment

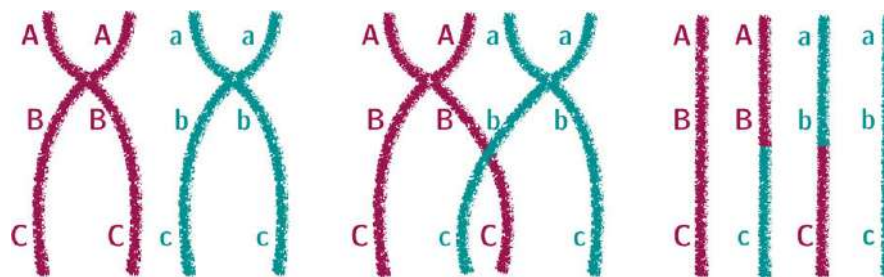


Fig. 1 Crossing over between sister chromatids of the same chromosome. Alleles at three loci (A, B and C) are shown. There is a single recombination event between B and C. The parental BC and bc haplotypes will become Bc or bC in the recombinant gametes.

Table 1 Chromosome lengths in terms of Sex-Averaged, Female and Male genetic distances and physical distances

Chromosome	Marshfield map length (cM)			Physical length (Mb)
	Sex-averaged	Female	Male	GRCh38
1	290	365	216	249
2	269	331	209	242
3	228	270	190	198
4	212	264	160	190
5	198	245	151	182
6	193	254	131	171
7	182	231	134	159
8	168	224	113	145
9	169	193	143	138
10	173	211	138	134
11	148	180	115	135
12	171	214	128	133
13	115	130	95	114
14	138	155	117	107
15	122	136	110	102
16	134	169	101	90
17	126	149	105	83
18	126	156	97	80
19	105	114	96	59
20	101	121	81	64
21	58	65	51	47
22	62	74	49	51
X		184		156
Total	3488	4435	2730	3031

Source: Genetic distances obtained from Broman, K.W., Murray, J.C., Sheffield, V.C., White, R.L., Weber, J.L., 1998. Comprehensive human genetic maps: Individual and sex-specific variation in recombination. *American Journal of Human Genetics* 63, 861–869. Physical distances obtained from The Genome Reference Consortium. Available from: <https://www.ncbi.nlm.nih.gov/grc/human/data>.

length polymorphisms), with an average resolution of 10 cM (Donis-Keller *et al.*, 1987). When microsatellites were identified in the genome, these became the genotyping tool of choice by geneticists, as they were multi-allelic and therefore highly heterozygous in the population (Litt and Luty, 1989; Weber and May, 1989). These polymorphic short tandem repeat sequences, usually di-, tri- or tetra-nucleotide repeats, are able to be amplified by PCR and therefore less cumbersome to genotype when compared to RFLPs. Their prevalence throughout the genome, and informativeness for recombination, led to the construction of the first dense human genetic maps, some of which are still in use today. The Marshfield genetic map (see “Relevant Websites section”) provided the first complete genome-wide comprehensive genetic map comprising more than 8000 microsatellite markers, including heterozygosity values for each marker (Broman *et al.*, 1998). Once the first draft of the human genome sequence was released, geneticists began to construct denser microsatellite marker genetic maps in a comparatively shorter period of time (Kong *et al.*, 2002). Comprehensive sequencing of the human genome identified an increasing number of bi-allelic single nucleotide polymorphisms (SNPs), which would ultimately become the next genetic marker of choice. SNPs are less informative than microsatellites (they only have a maximum heterozygosity of 0.5), but their abundance throughout the human genome has allowed for construction of very dense genetic maps. Due to their bi-allelic nature, hundreds of thousands of SNPs can be genotyped on a microarray in parallel with a small input volume of DNA. The next major human genetic map to be published was the deCODE genetic map in 2010 (Kong *et al.*, 2010). In contrast to previous microsatellite marker genetic maps, the deCODE genetic map comprises 300,000 less informative bi-allelic SNPs, but at a higher density to get resolution to 10 kb.

With the publication of the first reference human genome (Lander *et al.*, 2001; Venter *et al.* 2001) and the public repository of genetic mapping data, combined genetic-physical maps were able to be constructed (Kong *et al.*, 2004; Matise *et al.*, 2007). These maps are able to be used for interpolation of genetic map positions of unmapped markers or genomic variants. Web-based interpolation tools exist (such as that at Rutgers: see “Relevant Websites section”), or scripts from Github Repository for interpolation on a larger scale (an example python script: see “Relevant Websites section”). Interpolation of genetic maps (see Section Using Interpolated Genetic Maps for Shared IBD Segment Analysis below) is now particularly relevant with whole-genome sequencing datasets available with genotypes at each nucleotide position in the genome.

Generating a Genetic Map

Construction of genetic marker maps begins with a small set of loci, with the gradual addition of more loci. Somatic cell genetics and cytogenetics was used in the 1970s to first assign genetic markers to specific regions on single chromosomes. New genetic

markers, whether RFLPs, microsatellites or SNPs, are then genotyped in large families and recombination fractions are determined. As part of the Human Genome Project, DNA samples from the CEPH Reference Family Panel was used collaboratively worldwide for genotyping for the construction of genetic maps (Dausset *et al.*, 1990). CEPH Reference Family Panel DNA samples that were made available to the genetics community, allowed standardised genotyping by using a common set of families. Using observed recombination fractions from genotyping data, there are established software tools to construct a genetic map including LINKMAP (Lathrop *et al.*, 1984), MAPMAKER (Lander and Green, 1987; Lander *et al.*, 1987) and CRI-MAP (P. Green, K. Falls, and S. Crooks, documentation for CRI-MAP, version 2.4). These tools utilise map functions, which are mathematical relationships that transform non-additive recombination fractions (θ) into additive map distances (x). The most commonly used transformations are Haldane's and Kosambi's mapping functions (Haldane, 1919; Kosambi, 1944), both of which are still used by geneticists for mapping genomes. To calculate genetic distances (x) from recombination fractions (θ):

$$x = -1/2 \ln(1 - 2\theta) \quad \text{Haldane's map function}$$

$$x = 1/4 \ln \frac{1 + 2\theta}{1 - 2\theta} \quad \text{Kosambi's map function}$$

The mapping function that Kosambi derived assumes genetic interference, whereas the Haldane-derived mapping function does not allow for genetic interference. Genetic interference is the reduction in randomness of crossovers on a chromosome due to a decreased likelihood of a new crossover in the immediate vicinity of an existing crossover. Conversions between θ and x using several map functions can be performed interactively using the MAPFUN linkage utility program (see "Relevant Websites section"). Nonetheless, with today's dense human genetic maps, there is little need for mapping functions.

However in plant and animal genetics, construction of genetic maps of rare species is still an active field of research, and will be outlined below in Section Gene Mapping in Plants and Animals.

Linkage Analysis for Mapping Mendelian Disease Genes

When elucidating the cause of a Mendelian disease, identifying the causal gene would be considered the most important aspect of the research. Pinpointing the approximate location of the disease locus on a genetic map is the first step in identifying the causal gene. Subsequent fine mapping using additional genetic markers, and then sequencing of gene exons contained within the linkage region to find a mutation, is performed. Once a causal gene is identified, the mechanisms by which the mutation, and therefore mutated gene, causes disease is able to be further investigated. Identifying disease genes through linkage mapping offers the clinical potential for diagnostic tests and treatment.

Linkage Simulation and Genotyping Prior to Analysis

Disease gene mapping differs to constructing a genetic map of markers, as disease loci are mapped based on genotypes from families specifically collected for the study. Historically, to successfully identify a locus significantly linked to disease, a large multigenerational family with many DNA samples from both affected and unaffected individuals was required. Prior to generating genotype data, geneticists commonly perform a computer simulation on the family of interest. SLINK (Ott, 1989; Weeks *et al.*, 1990) and SIMLINK (Boehnke, 1986; Ploughman and Boehnke, 1989) are examples of such simulation software and require input data including pedigree, number of genetic markers used in the study, allele frequencies, recombination fractions between loci, mode of inheritance and disease penetrance. This prospective analysis can determine the statistical power of the proposed linkage study, and therefore the suitability of the family to conduct a linkage study.

DNA samples are then collected from available family members, including both individuals affected with the disease of interest and those unaffected. Extensive genotyping is then performed on the family members using either microsatellite markers or SNP microarrays, or a combination of both. Current SNP microarrays can genotype 500,000 – 2 million SNPs on a single chip. Although the density of markers is significantly greater when using SNP microarrays, they are less informative, frequently occur in LD (linkage disequilibrium) and have a maximum heterozygosity of 0.5. Since multiallelic microsatellite markers are highly informative due to their heterozygosity (which can be as high as 0.95), less markers are required for a comparable amount of linkage information (Dunn *et al.*, 2005). Genotype data and pedigree information is then converted into input files for linkage programs.

Linkage Programs and Lod Scores

The statistical measure of linkage for a disease locus is known as a lod score (or logarithm of the odds) and was first defined by Morton (1955). A lod score $z(x)$ is the logarithm (base 10) of the likelihood ratio: The likelihood of the data if the disease locus is linked to the genetic marker and the likelihood of the data if the disease locus is unlinked to the marker, where x represents a particular value of recombination fraction $0 \leq \theta \leq 1/2$:

$$z(x) = \log_{10} \left[\frac{L(\text{pedigree if linked, } \theta = x)}{L(\text{pedigree if unlinked, } \theta = 0.5)} \right]$$

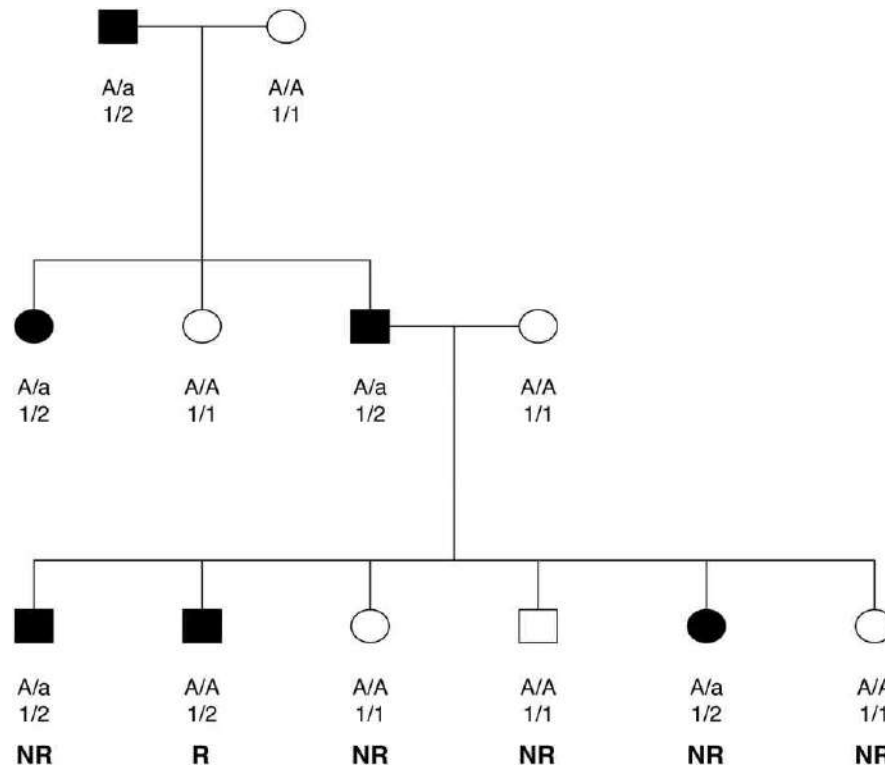


Fig. 2 Pedigree showing segregation of an autosomal dominant trait and a phase-known genetic marker. Recombinants (R) and non-recombinants (NR) between the trait locus and the marker locus are shown.

A lod score is a function of θ in the range of $0 \leq \theta \leq 1/2$ that always has a value of 0 at $\theta = 1/2$. The maximum of the lod score function occurs at the maximum likelihood estimate to θ . In the example pedigree in Fig. 2, the likelihood of observing linkage data is:

$$L = \theta^R (1 - \theta^{NR})$$

where θ is the recombination fraction, R is the number of recombinants and NR is the number of non-recombinants at a specific genetic marker. The likelihood of not observing linkage is

$$L = (0.5)^{(R+NR)}$$

Since grandparent genotypes are available this pedigree is considered phase-known, and therefore there is 1 recombinant and 5 non-recombinants in this pedigree. To calculate a two-point lod score at a range of recombination fractions is as follows:

$$z(\theta) = \log_{10} \left[\frac{(\theta)^1 (1 - \theta)^5}{(0.5)^6} \right]$$

$$z(x) = \log_{10} \left[\frac{0.5(\theta^1 (1 - \theta)^5) + 0.5(\theta^5 (1 - \theta)^1)}{(0.5)^6} \right]$$

At $\theta = 0.01, 0.05, 0.1, 0.2, 0.3$, and 0.4 , the calculated lod scores are $-0.22, 0.40, 0.58, 0.62, 0.51$, and 0.30 . However, if the pedigree only has genotype information in one or two generations, the pedigree is considered phase-unknown, and this must be taken into consideration in lod score calculations. Consider the example pedigree in Fig. 3 where there are two possible phases. Phase 1 has 1 recombinant and 5 non-recombinants, whereas Phase 2 has 5 recombinants and 1 non-recombinant. To calculate lod scores for this example, both phases need to be considered:

At $\theta = 0.01, 0.05, 0.1, 0.2, 0.3$, and 0.4 , the calculated lod scores are $-0.51, 0.09, 0.28, 0.32, 0.22$, and 0.08 . Calculating lod scores by hand is a very laborious task. A ground-breaking development in human genetics studies was the development of LIPED, the first generally available computer program to perform linkage analysis, thus computational calculation of lod scores (Ott, 1974, 1976). LIPED is based on the Elston-Stewart algorithm (Elston and Stewart, 1971) and is restricted to two-point linkage analysis, which is pairwise comparisons between a disease (or trait) locus and each marker locus. Linkage programs output lod scores for each marker at different recombination fractions (such as the example for Arts Syndrome in Table 2), and can also include the exact recombination fraction where the lod score is at a maximum ($Z_{\max}$).

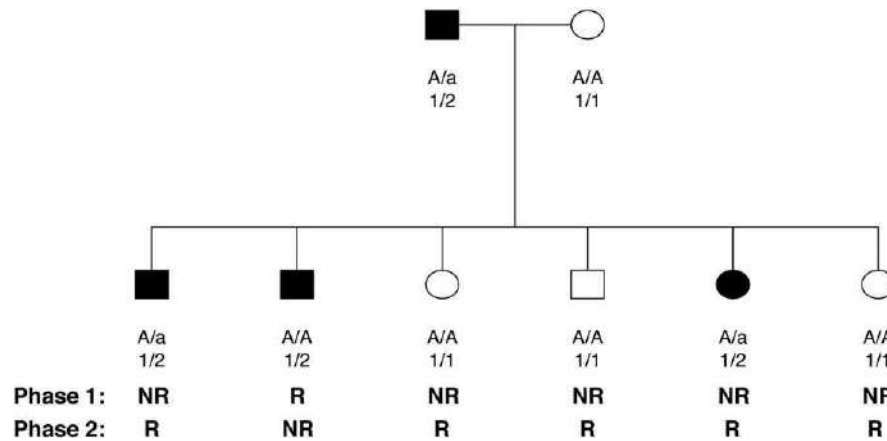


Fig. 3 Pedigree showing segregation of an autosomal dominant trait and a phase-unknown genetic marker. Recombinants (R) and non-recombinants (NR) between the trait locus and the marker locus are shown for the two different haplotype phases.

Table 2 Pairwise lod scores of Arts Disease and markers at Xq21.33 – Xq24. The descending order of markers is proximal to distal. The underlying disease gene was found to be *PRPS1* and maps to proximal of marker DXS1105.

Marker	Locus	$\theta = 0.00$	$\theta = 0.10$	$\theta = 0.20$	$\theta = 0.30$	$\theta = 0.40$
DXS1231	Xq21.33–q22.1	$-\infty$	-1.00	-0.33	-0.06	0.03
DXS178	Xq22.1	3.69	2.98	2.26	1.53	0.78
DXS1106	Xq22.2	6.97	5.57	4.08	2.53	1.06
DXS1105	Xq22.3	6.84	5.45	3.94	2.35	0.85
COL4A5	Xq22.3	5.38	4.18	2.95	1.78	0.76
DXS456	Xq22.3–q23	5.77	4.36	2.96	1.73	0.70
DXS424	Xq23	4.23	3.57	4.24	1.99	1.05
DXS1001	Xq24	$-\infty$	2.01	1.97	1.47	0.76

Note: Modified from Kremer, H., Hamel, B.C., van den Helm, B., *et al.*, 1996. Localization of the gene (or genes) for a syndrome with X-linked mental retardation, ataxia, weakness, hearing impairment, loss of vision and a fatal course in early childhood. *Human genetics* 98, 513–517 and de Brouwer, A.P.M., Williams, K.L., Duley, J.A., *et al.*, 2007. 'Arts syndrome is caused by loss-of-function mutations in *PRPS1*'. *American Journal of Human Genetics* 81, 507–518.

Since lod scores can be additive across families with an identical disease, two-point lod scores were historically reported at the following fixed set of θ values: 0, 0.01, 0.05, 0.1, 0.2, 0.3 and 0.4 (Conneally *et al.*, 1985). Lod scores (or sums of lod scores) of 3.0 or more are indicative of genome-wide significant evidence for linkage corrected for multiple testing (or greater than 2.0 on the X chromosome), whereas lod scores of less than -2.0 indicate non-linkage and can be used for exclusion purposes (Morton, 1955). Lod scores between -2.0 and 3.0 are inconclusive and require additional families, and thus more linkage analysis to be performed. Lander and Kruglyak (Lander and Kruglyak, 1995) built on Morton's work to determine that a more statistically accurate genome-wide significant lod score would be 3.3, and that a lod score of 1.86 is evidence of suggestive linkage.

The LIPED program was subsequently extended to provide multipoint analysis, resulting in the suite of LINKAGE programs for Linux and Windows ("see Relevant Websites section"), which are still in use today (Lathrop *et al.*, 1984). Dense genetic marker maps have made multipoint analysis possible, where simultaneous analysis of several linked loci is performed. Other multipoint linkage analysis programs are outlined in Table 3. In multipoint linkage analysis studies, geneticists will evaluate whether to apply parametric or non-parametric multipoint linkage analysis. Parametric multipoint analysis is usually applied to a few large pedigrees that have a known (or assumed) mode of inheritance of a disease or trait. A 'parameter' is set during linkage analysis: An explicit model of inheritance specifying the relationship between genotype and disease/trait. Parametric multipoint analysis is to identify linkage between the disease/trait gene and markers on the genetic map. In comparison, non-parametric multipoint analysis does not consider inheritance of the disease/trait. This analysis is usually considered an 'allele sharing' method that can look at many small families with the same disease.

Limitations of Current Linkage Analysis Programs

Although the introduction of computerised calculation of lod scores provided a platform for geneticists to regularly perform disease gene mapping studies, linkage programs have computational limitations. Improvement and enhancement in the algorithms over recent decades has improved efficiency of the programs, but computational burden is the major limitation, and is an ongoing problem with the current linkage programs in Table 3. Computational complexity increases with an increasing number of genetic markers or pedigree individuals. The Elston–Stewart algorithm (Elston and Stewart, 1971), which is used by LINKAGE and

Table 3 Multipoint linkage analysis software

Program	Multipoint analysis	Algorithm	Reference
LINKAGE	Parametric	Elston-Stewart	Ott (1974)
FASTLINK	Parametric	Elston-Stewart	Cottingham <i>et al.</i> (1993)
GENHUNTER	Parametric and Non-parametric	Lander-Green	Kruglyak <i>et al.</i> (1996)
ALLEGRO ^a	Parametric and Non-parametric	Lander-Green	Gudbjartsson <i>et al.</i> (2000)
MERLIN	Parametric and Non-parametric	Lander-Green	Abecasis <i>et al.</i> (2002), Abecasis and Wigginton (2005)

^aDeprecated.

FASTLINK programs, scales exponentially with the number of marker loci considered, and generally cannot handle more than 5–10 markers at a time. This is a serious limitation in the current environment where hundreds of thousands of genetic markers can be used per person in linkage analysis. Conversely, GENEHUNTER, ALLEGRO AND MERLIN use the Lander-Green algorithm which uses a hidden Markov model (HMM) to calculate probability distributions of inheritance vectors (Lander and Green, 1987). Computational time and memory requirements only increase linearly with additional markers, but increase exponentially as the number of meioses in a pedigree increases. In addition, there is an upper limit of 32 meioses in a pedigree because of the representation of inheritance vectors as 32-bit integers (Markianos *et al.*, 2001).

To date, there is no widely used linkage analysis program that can handle both large pedigrees and a large number of markers. There is a need for more powerful methods and enhancements to gene-mapping algorithms.

Human Linkage Analysis in the Age of Next-Generation Sequencing

Although linkage mapping for disease gene discovery was largely surpassed by large-scale GWAS studies due to the introduction of SNP microarrays, the era of next-generation sequencing has enabled linkage mapping to be revisited by geneticists. Combining linkage analysis results, with either whole-exome or whole-genome sequencing results has become an extremely powerful tool for identifying disease genes. Fig. 4 outlines standard procedures geneticists use when combining data from these technologies.

In the current genetics/genomics environment, usually smaller families undergo linkage analysis using either microsatellites or SNP microarrays. The main reason for this being that many large families with clear Mendelian inheritance have already undergone linkage analysis in the previous decade, and thus have data already available. Prospective simulation analysis will frequently determine that these smaller families cannot meet genome-wide significance (i.e., a lod score > 3.3), however lod scores of < -2.0 can exclude large regions of the genome that will not harbour the causal mutation. DNA from all available family members will undergo linkage analysis, however due to higher costs, only a few informative individuals from the pedigree will undergo next-generation sequencing.

Standard bioinformatic workflows applied to raw next-generation sequencing data results in a VCF file, comprised of all identified variants in the entire genome. These bioinformatic workflows are covered in detail in the Next generation sequence analysis Reference Module. With genomic variant data, instead of geneticists spending years of research performing laborious Sanger sequencing for every exon in every gene in each linked region, next-generation sequencing provides a list of all the variants present in a comparably short period of time (often weeks). For example, a recent gene discovery publication for a large autosomal dominant amyotrophic lateral sclerosis/frontotemporal dementia family used genome-wide microsatellite linkage analysis from all available family members combined with exome sequencing data from four affected individuals within the family. Linkage analysis pinpointed the disease gene to a 7.5 Mb linked region on chromosome 16, containing more than 100 genes comprised of over 2000 exons. A considerable effort, and cost, would be required to Sanger sequence each of these exons. In a demonstration of the power of this combined technique, whole-exome sequencing and standardised filtering of the VCF (unshared variants, MAF > 0.001 in public databases) identified only a single non-synonymous mutation in the linkage region, in the *CCNF* gene. Subsequent Sanger sequencing of this gene in all family members demonstrated strong segregation of the mutation with disease (Williams *et al.*, 2016).

Using Interpolated Genetic Maps for Shared IBD Segment Analysis

Linkage maps are not only used for disease gene mapping, but can be integrated with other software packages to identify cryptic relatedness within a cohort of interest. Currently programs exist to work with large GWAS cohorts. These programs include PLINK (Purcell *et al.*, 2007) and KING (Manichaikul *et al.*, 2010) which can determine first and second degree relatives. Often, geneticists want to identify these relations to remove them from a study to prevent confounding, biasing or inflating GWAS results. However, these tools are not accurate past 2nd degree relatives. More complex programs exist to identify more distantly related individuals (up to 7th degree) using IBD segment analysis, and can utilise either SNP microarray data or whole-genome sequencing variant files (VCF). First the genotype data needs to be phased using Beagle (Browning and Browning, 2007), usually in comparison to a phased references population. The phased data is compared alongside interpolated genetic maps to discover long shared segments of Identity by Descent (IBD) between pairs of individuals (GERLINE program; Gusev *et al.*, 2009). To then identify cryptic relatedness through shared ancestry, a maximum likelihood method is applied to the IBD segment data, using the number and genetic lengths of IBD segments, for the estimation of recent shared ancestry (ERSA) (Huff *et al.*, 2011).

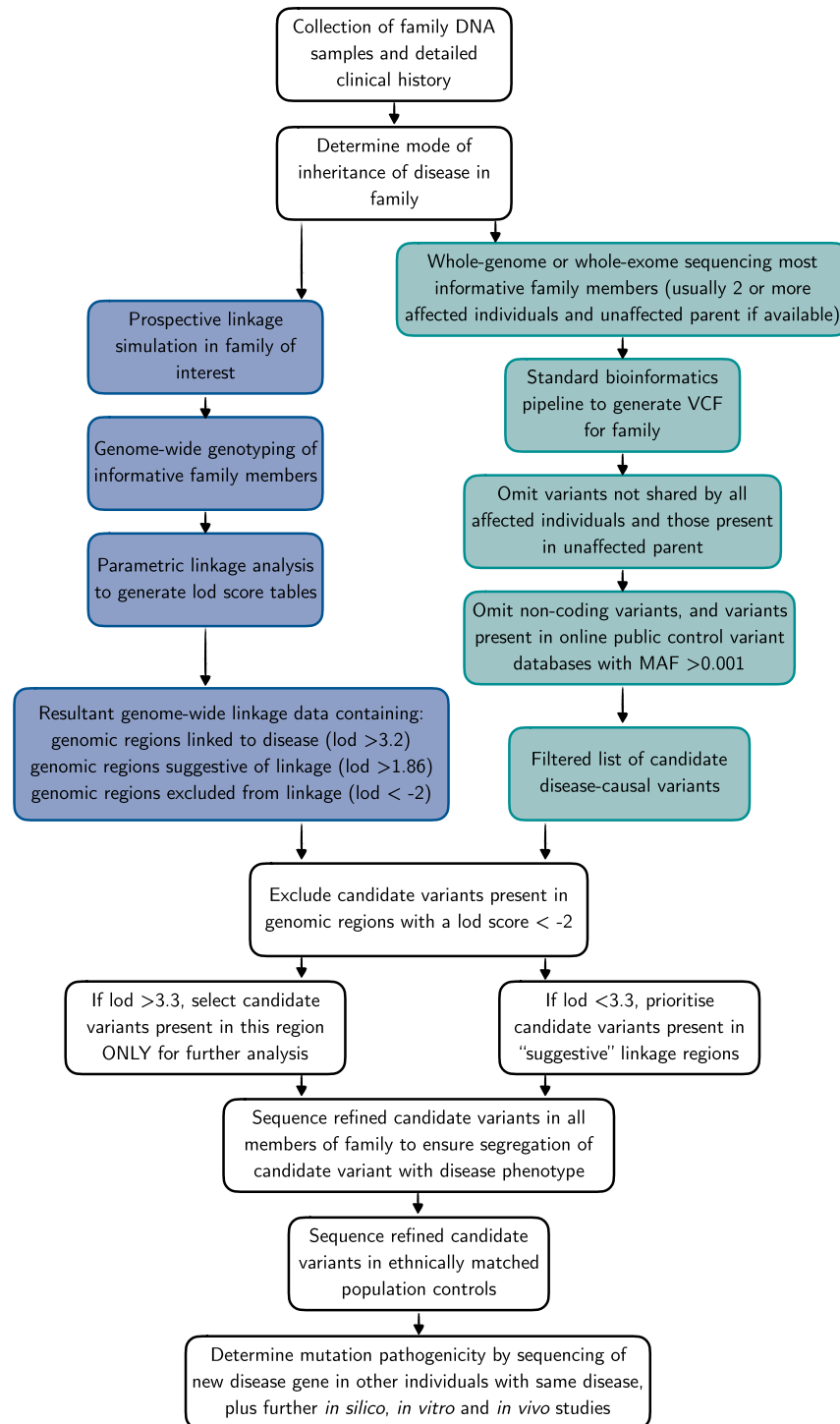


Fig. 4 Workflow for disease gene mapping using a combination of linkage analysis and next-generation sequencing methods. Linkage analysis workflow is outlined in blue, next-generation sequencing workflow is outlined in green, combined analysis is in black.

Gene Mapping in Plants and Animals

Existing human genetic maps are dense, well constructed, highly informative and amenable to interpolation, thus there is a limited need to develop new human genetic maps. Yet in plants and other animals, there is still a strong desire by geneticists to develop genetic maps for their species of interest. In contrast to human gene mapping studies, gene mapping in plants and animals, particularly those with agricultural interests, can be for identification of beneficial traits (quantitative trait loci, QTL). Genetic

markers linked to QTLs are then used for marker-assisted plant selection to expedite genetic improvement in the species. Next-generation sequencing technologies have opened a new article in gene mapping for rare species. Plant and animal geneticists can use whole-genome sequencing to construct draft reference genomes, identify species-specific genetic markers (such as SNPs), and then to use these markers in species-specific genetic maps. Existing genetic maps of similar species can also be used to refine and complete genetic maps of other species. For example, a recently completed genetic map of the Atlantic Salmon was used to improve the genome sequence of the Rainbow Trout (Lien *et al.*, 2016).

One of the major challenges for rare species is that draft reference genomes constructed by *de novo* assemblies are often incomplete and with alignment mistakes in the assembly. *De novo* assembly has good fine-scale accuracy, but poor large-scale accuracy for constructing chromosomes (Fierst, 2015). Geneticists are able to use genetic maps and recombination to resolve these errors by ordering, correctly orientating and concatenating the contigs into chromosomes. New tools coupling *de novo* assembly with linkage mapping provide a powerful method to generate a high-quality reference sequence (Rastas *et al.*, 2013; Liu *et al.*, 2014).

Conclusions

Gene mapping has gone through peaks and troughs over the past several decades. In the 1980s and 1990s, increasing number of publications were reported linking human Mendelian disease genes to loci. The Human Genome Project represented the peak of gene mapping efforts to develop dense gene maps of the human genome containing thousands of informative markers. In the 2000s, these genetic maps were highly used to pinpoint disease loci for more Mendelian disorders. However, for complex diseases, genome-wide association studies (GWAS) using SNP microarrays became increasingly popular. In the last decade, the introduction of next-generation sequencing dramatically decreased the use of linkage mapping in human disease. Instead, geneticists sequence the entire exome or genome of multiple affected individuals and search for shared variants as putative causal mutations. This does not require large families, usually only a few affected individuals in a family. This has proven more fruitful in recessive diseases compared to dominant diseases, and geneticists are beginning to return to use linkage mapping coupled with next-generation sequencing as a powerful approach to identify disease genes. Conversely, animal and plant genetics are thriving by using next-generation sequencing technologies to facilitate generating genetic maps of their chosen species. Prior to next generation sequencing, the cost, time and difficulty in generating a genetic map of a rare species was prohibitive. Now using variant identification following *de novo* assembly, coupled with linkage maps of similar species, an increased number of publications are coming from these fields of research. This has the potential for major impact on agricultural genetics, as high-density linkage maps facilitate mapping quantitative trait loci (QTL) for important traits. The main challenge facing geneticists now is overcoming, or reducing, sequencing errors encountered in next-generation sequencing.

See also: Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome Annotation. Genome Databases and Browsers. Genome Informatics. Genome-Wide Association Studies. Integrative Analysis of Multi-Omics Data. Linkage Disequilibrium. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Quantitative Immunology by Data Analysis Using Mathematical Models. Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?. Sequence Analysis. Sequence Composition

References

- Abecasis, G.R., Cherny, S.S., Cookson, W.O., Cardon, L.R., 2002. Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. *Nature Genetics* 30, 97–101.
- Abecasis, G.R., Wigginton, J.E., 2005. Handling marker-marker linkage disequilibrium: Pedigree analysis with clustered markers. *American Journal of Human Genetics* 77, 754–767.
- Boehnke, M., 1986. Estimating the power of a proposed linkage study: A practical computer simulation approach. *American Journal of Human Genetics* 39, 513–527.
- Broman, K.W., Murray, J.C., Sheffield, V.C., White, R.L., Weber, J.L., 1998. Comprehensive human genetic maps: Individual and sex-specific variation in recombination. *American Journal of Human Genetics* 63 (3), 861–869.
- Browning, S.R., Browning, B.L., 2007. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *American Journal of Human Genetics* 81, 1084–1097.
- Conneally, P.M., Edwards, J.H., Kidd, K.K., *et al.*, 1985. Report of the committee on methods of linkage analysis and reporting. *Cytogenetics and Cell Genetics* 40, 356–359.
- Cottingham, R.W., Idury, R.M., Schaffer, A.A., 1993. Faster sequential genetic linkage computations. *American Journal of Human Genetics* 53, 252–263.
- Dausset, J., Cann, H., Cohen, D., *et al.*, 1990. Centre d'étude du polymorphisme humain (CEPH): Collaborative genetic mapping of the human genome. *Genomics* 6 (3), 575–577.
- de Brouwer, A.P.M., Williams, K.L., Duley, J.A., *et al.*, 2007. Arts syndrome is caused by loss-of-function mutations in PRPS1. *American Journal of Human Genetics* 81, 507–518.
- Donis-Keller, H., Green, P., Helms, C., *et al.*, 1987. A genetic linkage map of the human genome. *Cell* 51, 319–337.
- Dunn, G., Hinrichs, A.L., Bertelsen, S., *et al.*, 2005. Microsatellites versus single-nucleotide polymorphisms in linkage analysis for quantitative and qualitative measures. *BMC Genetics* 6 (Suppl 1), S122.
- Elston, R.C., Stewart, J., 1971. A general model for the genetic analysis of pedigree data. *Human Heredity* 21, 523–542.
- Fierst, J.L., 2015. Using linkage maps to correct and scaffold *de novo* genome assemblies: Methods, challenges, and computational tools. *Frontiers in Genetics* 6, 220.
- Gudbjartsson, D.F., Jonasson, K., Frigge, M.L., Kong, A., 2000. Allegro, a new computer program for multipoint linkage analysis. *Nature Genetics* 25, 12–13.
- Gusev, A., Lowe, J.K., Stoffel, M., *et al.*, 2009. Whole population, genome-wide mapping of hidden relatedness. *Genome Research* 19, 318–326.
- Haldane, J.B.S., 1919. The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics* 8, 299–309.
- Huff, C.D., Witherspoon, D.J., Simonson, T.S., *et al.*, 2011. Maximum-likelihood estimation of recent shared ancestry (ERSA). *Genome Research* 21, 768–774.

- Kong, A., Gudbjartsson, D.F., Sainz, J., *et al.*, 2002. A high-resolution recombination map of the human genome. *Nature Genetics* 31, 241–247.
- Kong, A., Thorleifsson, G., Gudbjartsson, D.F., *et al.*, 2010. Fine-scale recombination rate differences between sexes, populations and individuals. *Nature* 467 (7319), 1099–1103.
- Kong, X., Murphy, K., Raj, T., *et al.*, 2004. A combined linkage-physical map of the human genome. *American Journal of Human Genetics* 75, 1143–1148.
- Kosambi, D.D., 1944. The estimation of map distances from recombination values. *Annals of Eugenics* 12, 172–175.
- Kremer, H., Hamel, B.C., van den Helm, B., *et al.*, 1996. Localization of the gene (or genes) for a syndrome with X-linked mental retardation, ataxia, weakness, hearing impairment, loss of vision and a fatal course in early childhood. *Human Genetics* 98, 513–517.
- Kruglyak, L., Daly, M.J., Reeve-Daly, M.P., Lander, E.S., 1996. Parametric and nonparametric linkage analysis: A unified multipoint approach. *American Journal of Human Genetics* 58, 1347–1363.
- Lander, E., Kruglyak, L., 1995. Genetic dissection of complex traits: Guidelines for interpreting and reporting linkage results. *Nature Genetics* 11, 241–247.
- Lander, E.S., Green, P., 1987. Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Sciences of the United States of America* 84, 2363–2367.
- Lander, E.S., Green, P., Abrahamson, J., *et al.*, 1987. Mapmaker: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1, 174–181.
- Lander, E.S., Linton, L.M., Birren, B., *et al.*, 2001. Initial sequencing and analysis of the human genome. *Nature* 409, 860–921.
- Lathrop, G.M., Lalouel, J.M., Julier, C., Ott, J., 1984. Strategies for multilocus linkage analysis in humans. *Proceedings of the National Academy of Sciences of the United States of America* 81 (11), 3443–3446.
- Lien, S., Koop, B.F., Sandve, S.R., *et al.*, 2016. The Atlantic salmon genome provides insights into rediploidization. *Nature* 533, 200–205.
- Litt, M., Luty, J.A., 1989. A hypervariable microsatellite revealed by in vitro amplification of a dinucleotide repeat within the cardiac muscle actin gene. *American Journal of Human Genetics* 44, 397–401.
- Liu, D., Ma, C., Hong, W., *et al.*, 2014. Construction and analysis of high-density linkage map using high-throughput sequencing data. *PLOS ONE* 9 (6), e98855.
- Manichaikul, A., Mychaleckyj, J.C., Rich, S.S., *et al.*, 2010. Robust relationship inference in genome-wide association studies. *Bioinformatics* 26, 2867–2873.
- Markianos, K., Daly, M.J., Kruglyak, L., 2001. Efficient multipoint linkage analysis through reduction of inheritance space. *American Journal of Human Genetics* 68, 963–977.
- Matisse, T.C., Chen, F., Chen, W., *et al.*, 2007. A second-generation combined linkage physical map of the human genome. *Genome Research* 17, 17831786.
- Morgan, T.H., 1911. Random segregation versus coupling in Mendelian inheritance. *Science* 34, 384.
- Morton, N.E., 1955. Sequential tests for the detection of linkage. *American Journal of Human Genetics* 7 (3), 277–318.
- Ott, J., 1974. Estimation of the recombination fraction in human pedigrees: Efficient computation of the likelihood for human linkage studies. *American Journal of Human Genetics* 26 (5), 588–597.
- Ott, J., 1976. A computer program for linkage analysis of general human pedigrees. *American Journal of Human Genetics* 28, 528–529.
- Ott, J., 1989. Computer-simulation methods in human linkage analysis. *Proceedings of the National Academy of Sciences of the United States of America* 86, 4175–4178.
- Ploughman, L.M., Boehnke, M., 1989. Estimating the power of a proposed linkage study for a complex genetic trait. *American Journal of Human Genetics* 44, 543–551.
- Purcell, S., Neale, B., Todd-Brown, K., *et al.*, 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *American Journal of Human Genetics* 81, 559–575.
- Rastas, P., Paulin, L., Hanski, I., Lehtonen, R., Auvinen, P., 2013. Lep-MAP: Fast and accurate linkage map construction for large SNP datasets. *Bioinformatics* 29 (24), 3128–3134.
- Sturtevant, A.H., 1913. The linear arrangement of six sex-linked factors in *Drosophila*, as shown by their mode of association. *Journal of Experimental Zoology* 14, 43–59.
- Venter, J.C., Adams, M.D., Myers, E.W., *et al.*, 2001. The sequence of the human genome. *Science (New York, NY)* 291, 1304–1351.
- Weber, J.L., May, P.E., . Abundant class of human DNA polymorphisms which can be typed using the polymerase chain reaction. *American Journal of Human Genetics* 44, 388–396.
- Weeks, D., Ott, J., Lathrop, G.M., 1990. SLINK: A general simulation program for linkage analysis. *American Journal of Human Genetics* 47, A204.
- Williams, K.L., Topp, S., Yang, S., *et al.*, 2016. CCNF mutations in amyotrophic lateral sclerosis and frontotemporal dementia. *Nature Communications* 7, 11253.

Further Reading

- Ott, J., 1999. *Analysis of Human Genetic Linkage*. Johns Hopkins University Press. Available at: <https://books.google.com.au/books?id=BKJqAAAAMAAJ>.
- Ott, J., Wang, J., Leal, S.M., 2015. Genetic linkage analysis in the age of whole-genome sequencing. *Nature Reviews Genetics* 16 (5), 275–284.
- Dudbridge, F., 2003. A survey of current software for linkage analysis. *Human Genomics* 1, 63–65.
- Fierst, J.L., 2015. Using linkage maps to correct and scaffold de novo genome assemblies: Methods, challenges, and computational tools. *Frontiers in Genetics* 6, 220.

Relevant Websites

- http://compugen.rutgers.edu/map_interpolator.shtml
Computational Genetics.
- <http://www.jurgott.org/linkage/LinkagePC.html>
Genetic LINKAGE programs for Windows and Linux.
- https://github.com/joepickrell/1000-genomes-genetic-maps/blob/master/scripts/interpolate_maps.py
GitHub.
- <http://www.bli.uzh.ch/BLI/Projects/genetics/maps/marsh.html>
Marshfield Genetic Maps.
- <http://www.jurgott.org/linkage/util.htm>
Statistical Genetics Utility programs.

Genome Databases and Browsers

Dean Southwood and Shoba Ranganathan, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

"In God we trust. All others must bring data." – W. Edwards Deming

"Data matures like wine, applications like fish." – James Governor

The effective use of genomic data is vital for many different fields, from disease research, to drug development, to biosecurity, to agriculture. Analysing this data lets us determine how organisms function, what causes certain traits and features, and what variation there is between different organisms, both within the same species as well as across species. As ideas and desires such as personalised medicine gain traction, the importance of careful and considered genomic analysis will only increase. However, the field of genomics is growing at an unprecedented rate, spurred on in recent times by the development and improvement of various genomic sequencing techniques. In the 15 years since the completion of the Human Genome Project in 2003, the time, manpower and financial costs of sequencing the genome of an organism have all dramatically decreased. It is therefore no surprise that there has been an explosion of sequencing data on both familiar and novel organisms in the years since. However, while it is of great benefit to science to have such a wealth of data available, it is difficult use effectively without the right tools. Appropriate methods of storing, organising, curating, and accessing the data (in the form of databases), as well as viewing, editing, and analysing the data (in the form of browsers) are vital in progressing the field.

As whole-genome sequencing (WGS) becomes more routine, smaller sequencing projects of specific genes or groups of genes have become even more common. The amount of data that these projects produce needs to be managed effectively, so that it can be stored, accessed and shared among researchers. However, the sheer volume of the data, as well as the number of organisms, data sources, and study types mean that there are thousands of different databases available to access genomic data, depending on what is being sought. Individual research communities will have specific databases that are more useful than others, but there are still a number of common features across them, and a small number of repositories that span multiple organisms.

Being able to access data is an important first step in doing high-quality research (or research at all), but if the data cannot be visualised and analysed effectively, there is little point in accessing it in the first place. This is especially true for genomic data, where its sheer volume coupled with its repetitive nature creates a situation in which annotation and visualisation are vital to making the data have any meaning. Genome browsers have been developed alongside databases to deliver on this requirement. However, like databases, there are many different browsers available with which to analyse data. These largely vary on how robust they are, whether they are server-based or local, and what specialised tools they have embedded in them. In general however, there are a small number of key browsers that tend to be suitable for broad, generic purposes, which have then served as the basis for the development of more specialised browsers and tools.

The current article is divided into four sections. The first section discusses currently available databases for accessing genomic data, including important multi-organism databases, as well as more specialised ones. The second section discusses the state-of-the-art genome browsers available to visualise and analyse genomic data, including their range of different capabilities and specialised tools. The third section uses WormBase as an example database, and visualises data from WormBase using GBrowse as an example of a broad-purpose genome browser. The fourth section discusses the future directions and issues involved in developing genome databases and browsers, and provides further reading to the interested reader.

Fundamentals

Genome Databases

Genomic databases allow for the storing, sharing and comparison of data across research studies, across data types, across individuals and across organisms. These are not a new invention – even before the popularisation of the modern internet, 'online' databases have been available in order to share data on key organisms, such as *Escherichia coli* (Blattner *et al.*, 1997) and *Saccharomyces cerevisiae* (Cherry *et al.*, 2012). Recent advances in both data sharing technology and genome sequencing technology have created an explosion of databases, based around particular organisms, as has been historically the case, as well as around particular data types, such as transcriptional data or short-read sequencing data. This section discusses a number of key classes of genome databases, namely broad cross-organism databases such as NCBI Genome and Ensembl Genome, specific organism

databases and the GMOD project family, and the broad variety of databases available for human genome research. The IDs provided in brackets correspond to those in the ID column of [Table 1](#).

Broad Databases

From the early development of genome databases several decades ago, there have been ongoing efforts to combine and index multiple databases in central repositories for ease of access. With the avalanche of genomic data in the last decade, these efforts have become more difficult. However, there are still two key databases which serve as first-point-of-call references for genomes, especially if the desired genomic information is likely to be accessed by many researchers. These are the NCBI Genome database (1), co-ordinated by the National Centre for Biotechnology Information in the United States, and the Ensembl Genomes database (2), jointly co-ordinated by EMBL-EBI and the Wellcome Trust Sanger Institute in the United Kingdom.

At time of writing, NCBI Genome contains genomic information on 35,211 distinct organisms, which are further indexed as eukaryotes (5413 entries), prokaryotes (134,749 entries), viruses (14,054 entries), plasmids (11,946 entries), and organelles (11,504 entries). These can contain multiple assemblies, and can be downloaded or searched on the database itself through a number of tools.

Ensembl Genomes also acts as a repository of genomic data across multiple kingdoms of organism, and is complemented by five sub-sites: Bacteria-Ensembl, Protist-Ensembl, Fungi-Ensembl, Plants-Ensembl, and Metazoa-Ensembl, containing the genomes of bacteria (50,364 entries), protists (200 entries), fungi (882 entries), plants (55 entries), and invertebrate metazoa (71 entries) respectively. This is in addition to the 12 central Ensembl organisms, including zebrafish (*Danio rerio*), chicken (*Gallus gallus*), human (*Homo sapiens*), and mouse (*Mus musculus*). Ensembl Genomes also allows for data to be downloaded or analysed using online tools. There is also the option to analyse the genomic data in Ensembl, the genome browser associated with the database itself (discussed further below).

Specific Organism Databases and the GMOD Project

It is possibly unsurprising that with the evolution of sequencing technology and the power to sequence the genome of most any organism, given a reasonable amount of time and a reasonable amount of research effort, individual databases have developed around the genomes of specific organisms. In the past, this was mostly focussed around so-called ‘model’ organisms, or ones with large research bases, such as the mouse (*Mus musculus*) ([Smith et al., 2018](#)) and nematode (*Caenorhabditis elegans*) ([Lee et al., 2018](#)). In many cases, these were created by their own research communities, to suit their own needs, both in terms of how the data could be accessed, as well as what tools were provided to dissect the data. As efforts continued, there have been moves to create some consistency between databases and the tools they offer, meaning new organism databases are not required to ‘re-invent the wheel’ so to speak. In this regard, the Generic Model Organism Database (GMOD) project has served to provide a framework of tools and database methods to allow new databases to be created. The ‘users’ of the GMOD project are no longer limited to ‘model’ organisms, and now consist of a variety of different species and databases. The GMOD project also has its own genome browser associated with it, GBrowse (as discussed further below), which can be integrated into the participating databases as a web-based genome browser.

Human Genome Databases

The breadth and depth of human genome databases is vast, as is to be expected when an organism attempts to study itself, and analyse its own biological problems. These databases are often structured around various data sources, such as transcriptional data, as is the case for the H-Invitational database (H-InvDB) (4). Particular study types have also given rise to specific databases: genome-wide association study databases such as GWASCentral (5), and structural variant study databases such as dbVar (6) and DGV (7). As is the case with other organisms, there are also some databases which seek to be more comprehensive in scope: DNA

Table 1 Prominent databases of genomic data; links current as of April 2018

Name	ID	URL	References
NCBI Genome	1	https://www.ncbi.nlm.nih.gov/genome	NCBI Resource Coordinators (2018)
Ensembl Genome	2	http://ensemblgenomes.org/	Kersey et al. (2018)
GMOD Project	3	http://gmod.org/wiki/Main_Page	N/A
H-InvDB	4	http://www.h-invitational.jp/	Yamasaki et al. (2009)
GWASCentral	5	https://www.gwascentral.org/	Beck et al. (2014)
dbVar	6	https://www.ncbi.nlm.nih.gov/dbvar	Lappalainen et al. (2013)
DGV	7	http://dgv.tcag.ca/dgv/app/home	MacDonald et al. (2014)
ENCODE	8	https://www.encodeproject.org/	Sloan et al. (2016)
IGSR (1000 Genomes)	9	http://www.internationalgenome.org/	The Genomes Project Consortium (2015)

element databases such as ENCODE (8), and the 1000 Genomes project database, now hosted as the International Genome Sample Resource (IGSR) (9). Databases for even more specific purposes exist, such as a wealth of databases on cancer genomic data, and will need to be searched for on a case-by-case basis depending on need.

Genome Browsers

Data access and quality means very little if no meaning can be gained from it. In a field with as complex and abstract data as genomics, methods for data visualisation and analysis are of even greater importance. These must be able to cope with vast amounts of data, in the order of gigabytes or terabytes, as well as be able to connect these to tangible, biological meaning in the form of genes and products. Genome browsers seek to fill this need by providing a pre-existing software basis to visualise and analyse genomic data. Due to the sheer variety of researchers, purposes, expectations, and goals involved in the field, a number of genome browsers are available. For the new user, there are three broad-class, easy-to-pick-up databases for generic uses that stand out at present: the UCSC Genome Browser, managed by the University of California, Santa Cruz (Casper *et al.*, 2018); GBrowse, managed by the GMOD project (see “Relevant Website section”); and Ensembl, managed by EMBL-EBI and the Wellcome Trust Sanger Institute (Zerbino *et al.*, 2018). This section will consider each of these browsers in turn, and then give an overview of the more specific browsers which have been created based on these three forerunners.

UCSC Genome Browser

UCSC Genome Browser is the one of the most widely regarded broad-class browser, and has been integrated into a number of major databases. Its initial conception in 2000 was to visualise the first working draft of the Human Genome Project, but has been adapted in the following years to include a broad variety of organisms, and a vast suite of tools for visualising and analysing data.

Gbrowse

Due to the ‘generic’ nature of the GMOD project (discussed above at 2.2), there was a need for a generic browser to accompany the suite of tools provided for new databases. GBrowse developed from this idea, and is therefore one of the more flexible genome browsers available. It has had a number of spin-off browsers created since its conception, tailored for particular purposes. As it is a part of the GMOD project, it is also available across many different databases.

Ensembl

The Ensembl genome browser created by EMBL-EBI and the Wellcome Trust Sanger Institute is the native genome browser for the Ensembl Genomes databases. Due to the broad nature of the databases it is used for, it contains a broad variety of tools for visualisation and analysis across a variety of kingdoms of organisms.

Specialised Browsers

Browsers for more specialised purposes have been developed by particular groups, largely based on one of the primary three browsers. Due to its generic nature, the majority of ‘subsidiary’ browsers are based on GBrowse in particular. Lighter implementations have been created, such as JBrowse, as well as browsers more suited to collaboration and annotation, such as Apollo.

Applications

Wormbase and GBrowse

An illustrative example of a user-friendly genome database, as well as a common all-purpose genome browser, is the WormBase-GBrowse pairing. WormBase (Lee *et al.*, 2018) is a member of the GMOD family, meaning it has the GBrowse genome browser embedded in the site. This makes quick searches extremely easy, and allows for visualisation of data before download.

The basic functionality of WormBase is illustrated in Fig. 1, through the homepage, search, and GBrowse features. The homepage is very user-friendly with logical menus and news items. The search function allows the user to access gene information across a variety of species, including *C. elegans*, *C. briggsae*, and *C. brenneri*. It also allows for search by class of information, including gene, strain, protein, or sequence feature. Searching for a particular gene gives the option to open the gene in GBrowse, as shown in Fig. 1(c) for the *set-23* gene in *C. elegans*. The *set-23* gene has been shown to be an embryonic lethal gene, and as such could be a possible CRISPR target in the future – knowing the location (chromosome IV), length (2625 bp), and protein coding regions (as shown in pink) of the gene are important, and easily accessible from the browser view. GBrowse is also equipped with an extensive functionality and toolbox – further information can be found through the GMOD wiki (see “Relevant Website section”), including tutorials on more specialised features of GBrowse.

(a) WormBase Homepage

WormBase Version: WS363

My WormBase (2) Login For Developers Contact Us

Search directory... for a gene

About Directory Tools Downloads Community Support

Submit Data Micropublication ParaSite

Explore Worm Biology
facilitating insights into
nematode biology

control what you see on the page skip tutorial

see a star? click on it to save to My Wormbase

Page Content

- News
 - Please specify allele and strain names in publications Tue, 10 Apr 2016
 - WormBase curates data from published papers and attaches different types of data such as phenotype, overview, expression, human disease
 - Sir John Sulston Tue, 15 Mar 2016
 - We mourn the passing of one of the founders of our field. John Sulston was personally responsible for establishing the use of *C. elegans* to study
 - WormBase Release WS363 Thu, 08 Mar 2016
 - We would like to announce the availability of the WormBase WS363 release on the WormBase website and FTP. Some of the highlights of this
 - View More
- My WormBase
 - My Favorites
 - My Library
- Recent Activity
 - turn on history
 - history logging is off

Gene name changes

Below are changes in gene names since the previous release WS262. Gene name changes for each release since WS252 are archived [here](#).

Genes with new primary names

Search: Save table

Show 10 entries	New primary name	Gene ID	Sequence
	<i>ascc-1</i>	WBGene00016019	C23H3.3
	<i>ddx-52</i>	WBGene00011032	R05D11.4
	<i>efr-3</i>	WBGene00016311	C32D5.3
	<i>ftuap-3.1</i>	WBGene00007534	C12D
	<i>ftuap-3.2</i>		
	<i>ftuap-3.3</i>		

Need help or have

(b) WormBase Search Tool

search... for a gene in all species

Classes

Analysis	Expression profile	Microarray results	Position matrix	Strain
Anatomy term	GO term	Molecule	Process&Pathway	Structure Data
Antibody	Gene	Motif	Protein	Transcript
CDS	Gene class	Operon	Pseudogene	Transgene
Clone	Gene cluster	PCR Product/Oligo	RNAi	Transposon
Construct	Homology group	Paper	Reagent	Transposon Family
Disease Ontology	Interaction	Person	Rearrangement	Variation
Expression cluster	Laboratory	Phenotype	Sequence	
Expression pattern	Life stage	Picture	Sequence feature	

Species

All Species	<i>C. brenneri</i>	<i>C. elegans</i> CB4856	<i>C. remanei</i> px356	<i>O. volvulus</i>
<i>B. malayi</i>	<i>C. briggsae</i>	<i>C. japonica</i>	<i>C. sinica</i>	<i>P. pacificus</i>
<i>C. angaria</i>	<i>C. elegans</i>	<i>C. remanei</i>	<i>C. tropicalis</i>	<i>P. redivivus</i>

(c) GBrowse

File Help

C. elegans (BioProject PRJNA13758): 2.625 kbp from IV:1,602,985..1,605,609

Browser Select Tracks Snapshots Custom Tracks Preferences

Search Overview Region Details

165% 1 kbp

Curated Genes

set-23 (457kb, 12)

protein coding

Polymorphisms

Select Tracks Clear highlighting

Fig. 1 Example of a genome database and browser pairing using WormBase and GBrowse, showing (a) the homepage of WormBase, (b) the search tool by search class and by organism, and (c) GBrowse, with the Set-23 gene of *C. elegans* selected as the track.

Discussion and Future Directions

With the wealth of databases and browsers available today for genomic research, researchers are in some ways spoilt for choice. However, as new pursuits such as personalised medicine progress, databases and browsers will need to be developed or adapted to meet these new demands. In a field with such fast-paced progress as genomics, not all future issues will be able to be predicted, and many will have to be dealt with as they arise. However, a few key directions are evident. The continued development of

high-throughput sequencing technology will likely continue to bring sequencing time and money costs down, increasing the availability of genomic data. Therefore, current databases will need to be optimised and adapted, and new databases created, to allow for such an influx of data. The curation and annotation of genomic data is already an issue, significant efforts put into generating the data, but less plentiful efforts put into assigning it with accurate annotations before adding it to databases; as data creation increases, this problem will only continue to get more and more relevant. More effective strategies around who is charged with curating and annotating the data in order to give it meaning, while keeping it reliable and reputable, need to be devised.

Closing Remarks

There are a multitude of databases and browsers available for genomic purposes, and with future increases in genomic data, there will only be more and more new databases formed, and new browsers created. No static article can keep up with the furious developments in the field; the hope is that this article can provide an introduction to the researcher looking into genomic databases and browsers for the first time, or looking for a refresher. This article is also not comprehensive, nor does it purport to be; with the already vast sea of resources available, no article could hope to offer a view of the whole current offering. If the reader requires more specialised databases and tools, hopefully the notions discussed herein provide a guide for what to look for.

See also: Bioinformatics Approaches for Studying Alternative Splicing. Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Comparative and Evolutionary Genomics. Comparative Genomics Analysis. Data Storage and Representation. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Functional Genomics. Gene Mapping. Genome Alignment. Genome Annotation. Genome Annotation: Perspective From Bacterial Genomes. Genome Informatics. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Linkage Disequilibrium. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Ontology-Based Annotation Methods. Phylogenetic Footprinting. Quantitative Immunology by Data Analysis Using Mathematical Models. Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?. Transcriptomic Databases. Whole Genome Sequencing Analysis

References

- Beck, T., Hastings, R.K., Gollapudi, S., *et al.*, 2014. GWAS Central: A comprehensive resource for the comparison and interrogation of genome-wide association studies. *Eur. J. Human Genet.* 22 (7), 949–952.
- Blattner, F.R., Plunkett 3rd, G., Bloch, C.A., *et al.*, 1997. The complete genome sequence of *Escherichia coli* K-12. *Science* 277 (5331), 1453–1462.
- Casper, J., Zweig, A.S., Villarreal, C., *et al.*, 2018. The UCSC genome browser database: 2018 update. *Nucleic Acids Res.* 46, D762–D769.
- Cherry, J.M., Hong, E.L., Amundsen, C., *et al.*, 2012. *Saccharomyces* genome database: The genomics resource of budding yeast. *Nucleic Acids Res.* 40, D700–D705.
- Kersey, P.J., Allen, J.E., Allot, A., Barba, M., *et al.*, 2018. Ensembl genomes 2018: An integrated omics infrastructure for non-vertebrate species. *Nucleic Acids Res.* 46, D802–D808.
- Lappalainen, I., Lopez, J., Skipper, L., *et al.*, 2013. DbVar and DGVA: Public archives for genomic structural variation. *Nucleic Acids Res.* 41, D936–D941.
- Lee, R.Y.N., Howe, K.L., Harris, T.W., *et al.*, 2018. WormBase 2017: Molting into a new stage. *Nucleic Acids Res.* 46, D869–D874.
- MacDonald, J.R., Ziman, R., Yuen, R.K., *et al.*, 2014. The Database of genomic variants: A curated collection of structural variation in the human genome. *Nucleic Acids Res.* 42, D986–D992.
- NCBI Resource Coordinators, 2018. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 46, D8–D13.
- Sloan, C.A., Chan, E.T., Davidson, J.M., *et al.*, 2016. ENCODE data at the ENCODE portal. *Nucleic Acids Res.* 44, D726–D732.
- Smith, C.L., Blake, J.A., Kadin, J.A., *et al.*, 2018. Mouse Genome Database (MGD)-2018: Knowledgebase for the laboratory mouse. *Nucleic Acids Res.* 46, D836–D842.
- The Genomes Project Consortium, 2015. A global reference for human genetic variation. *Nature* 526, 68–74.
- Yamasaki, C., Murakami, K., Takeda, J.I., *et al.*, 2009. H-InvDB in 2009: Extended database and data mining resources for human genes and transcripts. *Nucleic Acids Res.* 38, D626–D632.
- Zerbino, D.R., Achuthan, P., Akanni, W., *et al.*, 2018. Ensembl 2018. *Nucleic Acids Res.* 46, D754–D761.

Relevant Website

http://gmod.org/wiki/Main_Page
GMOD.

Biographical Sketch



Dean Southwood is a PhD student in the Department of Molecular Sciences at Macquarie University. His background is in theoretical quantum physics, quantum computation, and bioinformatics. His current research interests involve genome analysis and improving basecalling algorithms using machine learning.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books in as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.

Comparative and Evolutionary Genomics

Takeshi Kawashima, National Institute of Genetics, Mishima, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Comparative genomics is a fundamental tool of genome analysis. Typically, DNA sequences from whole genomes and whole gene sets are compared to elucidate the common and different genomic features among two or more target organisms. Features common to two species are considered properties originating from the genome of their common ancestry, whereas different features reflect the distinct histories of each genome traced since the common ancestor. This rationale can be applied to any two species in a comparative analysis because all organisms on the Earth can be connected to a single phylogenetic tree (Fig. 1). Seen in this sense, comparative genomics is equivalent to evolutionary genomics.

However, caution is required in comparative analyses because exceptional events, such as horizontal gene transfer and convergent evolution, can often be detected at levels more than expected. In such cases, it is necessary to carefully confirm the accuracy of the data and the correctness of the analytical process. Whether the result can be verified by other independent methods should also be considered.

Previous reports on comparative genomics are rich examples of the types of analyses and modifications performed for each case. In this report, the achievements of comparative genome analysis are discussed in order to unravel the evolution of metazoans, and the analytical methods used in these studies and the knowledge obtained from application of these methods are outlined.

Aspects of Comparative Genomics: Structure, Type of Change, and Evolutionary Distances

When performing comparative genomic analyses, the following three biological aspects should be considered: genome structure, types of structural change, and evolutionary distance.

Genome Structure and Types of Change

For convenience in genome analyses, a genomic DNA sequence can be divided into genomic “parts”, such as protein-coding genes, noncoding genes, and intergenic regions that include repeat and interspersed sequences, and other functional regions represented by cis-/trans-regulatory elements.

Each of these parts undergoes changes throughout a generation through the combination of several evolutionary mechanisms. In the evolution of protein-coding genes, Koonin classified the major mechanisms into the following five types of change: (i) direct descendants with some nucleotide changes; (ii) gene duplication; (iii) gene deletion; (iv) gene gain by horizontal gene transfer;

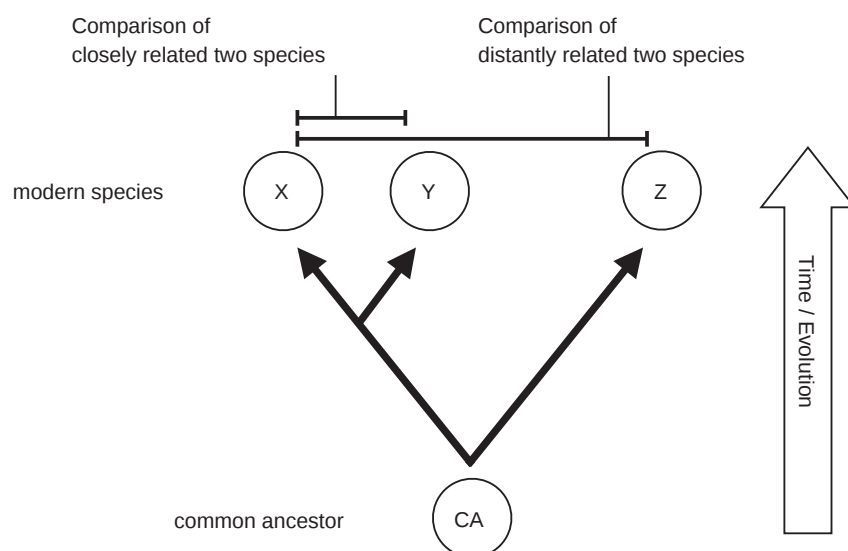


Fig. 1 Conceptual Diagram of Relationship of Taxonomic Distances in Comparative Genomics. Genome data can be obtained for X, Y and Z, each of which is an extant organism. X and Y are closer to each other than to Z. X and Z have evolved since they diverged from the common ancestral organism CA. When comparing the three types of X, Y, Z, the commonly observed features may be the characteristics of CA.

and (v) fusion, fission, and other genetic restructuring, including loss or gain of new domains on the gene (Koonin, 2005). In addition to these, (vi) emergence of a de novo gene is also mentioned often.

“Emergence of a de novo gene (have been frequently occurred)” is a relatively new concept. In this mechanism, a protein-coding gene emerges from noncoding DNA with some mutations. There are many unresolved problems in gene evolution, including the contribution rate. The mechanism was mainly discovered from the comparative genome analysis of closely related fly species (Begun *et al.*, 2006). The fixation rate of a de novo gene is still unknown. However, it seems certain that de novo genes are constantly created from noncoding regions at a certain evolutionary rate (Zhao *et al.*, 2014).

The above six mechanisms (i) to (vi) were originally described for the evolution of protein-coding genes. However, the same classification may also be applied to the other genomic parts by replacing them more general concepts, such as: (i) direct descendants, (ii) duplication, (iii) deletion, (iv) gain by horizontal transfer, (v) changes in partial structure, and (vi) de novo emergence.

The proportion of each genomic part differs greatly among organisms (Lindblad-Toh *et al.*, 2005). This suggests that each evolutionary mechanism works independently on each genomic part, resulting in the creation of differences in the characteristics of various genomes as a basis for forming biodiversity from common ancestry (Simakov and Kawashima, 2017).

For example, the number of gene losses since the common mammalian ancestor has been estimated as about 1000 for humans and about 2000 for dogs, which is almost twice the rate (Demuth *et al.*, 2006). As other examples from an comparative analysis of interspersed sequences, the proportion of repetitive sequences in the genome of limpet (*Lottia gigantea*) is reported to be only around 21%, whereas those in the genomes of Hydra (*Hydra magnipapillata*) and humans are 57% and 43%, respectively (Simakov *et al.*, 2013; Chapman *et al.*, 2010). In addition, the evolutionary rate of the relative position of each part also differs in genome evolution. Preservation of the gene position on a chromosome is represented by the concepts of synteny and linkage, each of which forms a genomic feature. Many microsynteny derived from the common ancestry are detected between the genomes of amphiox and chickens, but are not detected between the genomes of amphiox and tunicates. This suggests that the evolutionary rate of the chromosomal structure has considerable differences among animals.

Although there are major differences in the genomic features of each organism, we can infer those features of the common ancestor. Table 1 shows an example of the estimation value of genomic features of the metazoan ancestor (Simakov and Kawashima, 2017).

In any case, the genome has evolved by changing all genomic parts based on the above six mechanisms, making up the genomic features of extant organisms.

Evolutionary Distances and Taxonomic Data Bias

Evolutionary distances for comparative analysis

Evolutionary distances between the target species of comparative analysis are largely related to choosing an appropriate data type and the method to be used.

By way of one of the most basic methods of comparative genomics, finding a common genetic set is relatively simple in most cases, for any pair of organisms (e.g., even for human and bacteria, in The *C. elegans* Sequencing Consortium, 1998). In such analyses, it is necessary to correctly understand the concepts of “orthologs,” “paralogs,” and other related concepts. These concepts are detailed in the next Section “Homologous Genes: Orthologs, Paralogs, and Others”.

In contrast, biological discovery becomes difficult if the distance between the compared species is too far or too close in some cases. This can be easily understood by considering the sequence comparison of intergenic regions as an example. Generally, in comparing intergenic regions between two genetically distant species, few conserved sequences are found. On another front, comparison between very closely related or among the same species reveals nearly identical DNA sequences, such that almost no differences may be detected between two species unless a considerable number of sequences from multiple individuals are compared.

Conservation and differences among relative positions of genes on the DNA between two species provides a history of chromosomal evolution, also called as “conserved-synteny” or “conserved linkage.” Although more than hundreds of orthologous genes are conserved between fungi and animals (Li *et al.*, 2003), almost no conservation is found in most of the gene-order between these

Table 1 An example of the estimation value of genomic features of the metazoan ancestor

Genomic feature	Inferred ancestral values
Presumptive genome size	~300 Mb
Gene family number	7,000–8,000
Total gene number	>20,000
Average exon count	3–4
Repeat content	~30%
Macro-syntenic linkage groups	~10–17
Micro-syntenic linkage groups	~400

Note: Reproduced from Simakov O., Kawashima, T., 2017. Independent evolution of genomic characters during major metazoan transitions. *Dev. Biol.* 427, 179–192.

two possibly because they are highly distant evolutionarily. Only exceptional example is reported in which the order of three genes, *psmb 1*, *Tbp*, and *Pdcd 2*, is conserved among yeast, humans, and some other animals (Trachtulec and Forejt, 2001).

As described above, it is necessary to carefully ensure appropriate evolutionary distance among subject data before initiating comparative studies.

Biased genomic data

The result of comparative genome analyses is susceptible to taxon sampling. Unfortunately, considerable taxonomic-bias is commonly found in any usable sequence data in practice. To perform a comparative analysis correctly, researchers need to be recognize the extent of bias in the genomic data which they used.

In metazoans for example, the genome sequences of chordate animals are overwhelmingly numerous by total published data, followed by those of Arthropoda (the column of “Number of decoded species” in Fig. 2). In contrast, for other animal taxonomic groups at the phylum level, it is often observed that no genomic data have been decoded or only one or two species have decoded genomes (e.g., gray vertical bar A and B in Fig. 2). Of the genome data for the phylum Chordata, most of the data are overly concentrated in sub-phylum Vertebrata compared to the other two subphyla, Urochordata and Cephalochordata.

The main reason for the sequence data bias towards vertebrates is presumably because human-related research accounts for a major portion of biological studies. Especially from a medical perspective, the amount of sequence data for humans and their closely related species tends to be increased. Genome analysis of domestic animals is also actively pursued as it is probably considered valuable from an agricultural point of view, regardless mammals have a larger genome size being more than 1 Gb (Anderson and Georges, 2004). 16 metazoan species are known as major domestic animals including 15 of mammals (dog, goats, sheep, cattle, pig, horse, cat, camel, ferret, chicken, turkey, rabbit, guinea pig, llama, and alpaca) and an insect, silkworm (Morey, 1994; Banno et al., 2010). Of the 16 major domestic animals, genomes of 12 mammals and one insect have been decoded

Phylum Level Taxonomy *	Number of Decoded Spec.**	Common Name	Species Name	Size (Mb)	Year	Authors	Remarkable Genomic Findings
Deuterostomia							
Chordata	291	● Round Worm	<i>C. elegans</i>	97	1998	CSC.,	The first sequenced metazoan genome. Olfactory receptor expansion.
Hemichordata	2	● Fly	<i>D. melanogaster</i>	180	2000	Adams et al.,	The first metazoan WGS genome. Hetero-euchromatin transition zone.
Echinodermata	7	● Human	<i>H. sapiens</i>	3,289	2001	IHGSC., Venter et al.,	Total human genes fewer than thought.
Ecdysozoa							
Nematoda	81						
Priapulida	1						
Nematomorpha	0	● Purple sea urchin	<i>S. purpuratus</i>	814	2006	SUGSC et al.,	Genes associated with vision, balance, and chemosensation in urchin.
Loricifera	0	● Starlet sea anemone	<i>N. vectensis</i>	357	2007	Putnam et al.,	Ancient metazoan genome complement and organization.
Kinorhyncha	0	● Placozoa	<i>T. adhaerens</i>	98	2008	Srivastava et al.,	Diversity of developmental genes more than cell types of this organism.
Onychophora	0	● Schistosoma	<i>S. mansoni</i>	363	2009	Berriman et al.,	Micro exon and unusual intron size distribution.
Tardigrada	1	● Demosponge	<i>A. queenslandica</i>	167	2010	Srivastava et al.,	Six hallmarks of animal multicellularity and the evolution of relevant genes.
Arthropoda	239						
Lophotrochozoa							
Mollusca	10	● Owl limpet	<i>L. gigantea</i>	348	2012	Simakov et al.,	Multiple duplications of Hox complements in the leech.
Annelida	2	● Polychaeta	<i>C. teleta</i>	324			
Brachiopoda	1	● Freshwater leech	<i>H. robusta</i>	228			
Cycliophora	0	● Bdelloid rotifer	<i>A. vaga</i>	224	2013	Flot et al.,	Intragenomic synteny and ameiotic evolution.
Chaetognatha	0	● Comb jellies	<i>M. leidyi</i>	150	2013	Ryan et al.,	Mesodermal genes absent from Ctenophore genome.
Phoronida	1		<i>P. bachei</i>	156	2014	Moroz et al.,	Independent evolution of the nervous system.
Bryozoa (Ectoprocta)	0	● Lingula	<i>L. anatina</i>	425	2015	Luo et al.,	Expansion of chitin synthase corresponding to the shell formation.
Entoprocta	0	● Water bear	<i>H. dujardini</i>	212	2015	Boothby et al.,	
Acanthocephala	0	● Acornworm (Atlantic)	<i>S. kowalevskii</i>	758	2015	Simakov et al.,	Deuterostome novelties and the pharyngeal cluster.
Gastrotricha	0	● Acornworm (Pacific)	<i>P. flava</i>	1229			
Gnathostomulida	0	● Orthonectid	<i>I. linei</i>	43	2016	Mikhailov et al.,	Elemental genes of the metazoan sensory systems.
Nemertea	1	● Ribbon worm	<i>N. geniculatus</i>	859	2017	Luo et al.,	Lophotrochozoans share an ancestral bilaterian gene repertoire with deuterostomes.
Rhombzoa	0	● Horseshoe worm	<i>P. australis</i>	498			
Orthonectida	1						
Rotifera	2						
Platyhelminthes	29						
Basal-Bilateria							
Xenacoelomorpha	0						
Non-Bilateria							
Cnidaria	10						
Placozoa	1						
Ctenophora	2						
Porifera	1						

Fig. 2 History of animal genomics. On the left side of the figure, 32 phyla of animals (metazoa) are described. There are various opinions as to how many are appropriate number for phylum level classification. Recently it classified often as fewer than 32 phylum. In the recent phylogenetic interpretation, all animal phyla are divided into two large groups, non-bilateria and bilateria. And the latter is classified further into three groups, Ecdysozoa, Lophotrochozoa and Deuterostomia. In the column of “the number of decoded species”, how many species of its genome was decoded from each animal phylum was described based on the data of NCBI Taxonomy Database at the end of 2017. On the left side of the figure, among the respective animal phyla, the animal species whole genome was first decoded are shown in chronological form. Looking at the gray bar (A and B), we can see that there are considerable bias in the number of genomes decoded for each animal phylum.

(Kirkness *et al.*, 2003; Lindblad-Toh *et al.*, 2005; Dong *et al.*, 2013; Jiang *et al.*, 2014; The Bovine Genome Sequencing and Analysis Consortium *et al.*, 2009; Rubin *et al.*, 2012; Wade *et al.*, 2009; Pontius *et al.*, 2007; The Bactrian Camels Genome Sequencing and Analysis Consortium, 2012; Peng *et al.*, 2014; Rubin *et al.*, 2010; Dalloul *et al.*, 2010; Carneiro *et al.*, 2014; Mita *et al.*, 2004; Xia *et al.*, 2004). For the remaining three other animals (guinea pig, llama, and alpaca), genome sequencing projects are still ongoing.

In several cases for the genome projects of the domestic animals, multiple strains and individuals have been re-sequenced for applying various methods of population genetics to identify the genes responsible for genetic features of domestic animals that have been acquired through the history of domestication. Some examples of such genome projects of domestic animals will be described later.

Early studies decoded the insect genome presumably because it includes the fruit-fly, which is a major model organism of basic experimental biology, and also because many of them have relatively small genome sizes (Adams *et al.*, 2000; Gregory, 2011). Another reason is that several insects are deeply involved in the human society; some of them are useful, such as bees and silkworms whereas others are pests including aphids and flour beetles (Grimmelikhuijzen *et al.*, 2007).

Thus, analysis of genome sequence data cannot avoid bias for various reasons and it may simply be correlated with the researcher's population in most cases. Thus, it is important to determine the existence of bias in published data when performing comparative genome analysis.

Comparison of Homologous Genes

Homologous Genes: Orthologs, Paralogs, and Others

During comparative analysis of genes, it is necessary to understand the types of homologous genes (Fig. 3). "Homologous genes" are defined as genes with a common evolutionary ancestry (Brown, 2007). "Homologous genes" are sometimes called homologs (or homologues), but homolog(ue) is originally a histological term introduced by Owen, and defined as "the same organ in different animals under every variety of form and function (Owen, 1843)". Thus, this term is often inappropriate for use in genomics. Instead "orthologs" and "paralogs" are used frequently, both of which are simply basic concepts for genes that have a common ancestor with respect to molecular evolution (Fig. 3(A)). Orthologs are determined as homologous genes that diverged as a result of a speciation event. Paralogs are defined as homologous genes that diverged as a result of a gene duplication event. Another related term that is not used frequently, is "xenologs," defined as "homologous genes that diverged as a result of lateral gene transfer" (Kristensen *et al.*, 2011).

"Outparalogs (Alloparalogs)," "Inparalogs (Symparalogs)," and "Co-orthologs" are also defined as terms for further fine classification (Sonnhammer and Koonin, 2002). Outparalogs are defined as paralogs that were duplicated before the speciation event. Inparalogs are defined as paralogs that were duplicated after the speciation event. If lineage-specific gene-duplications (or expansion) has occurred in one or both of a pair of orthologs after speciation, sister genes of such duplicated orthologs constitute one-to-many or many-to-many relationships, respectively. Such gene relationships are called co-orthologs. Co-orthologs are thus defined as paralogs produced by duplication of orthologs after a given speciation event.

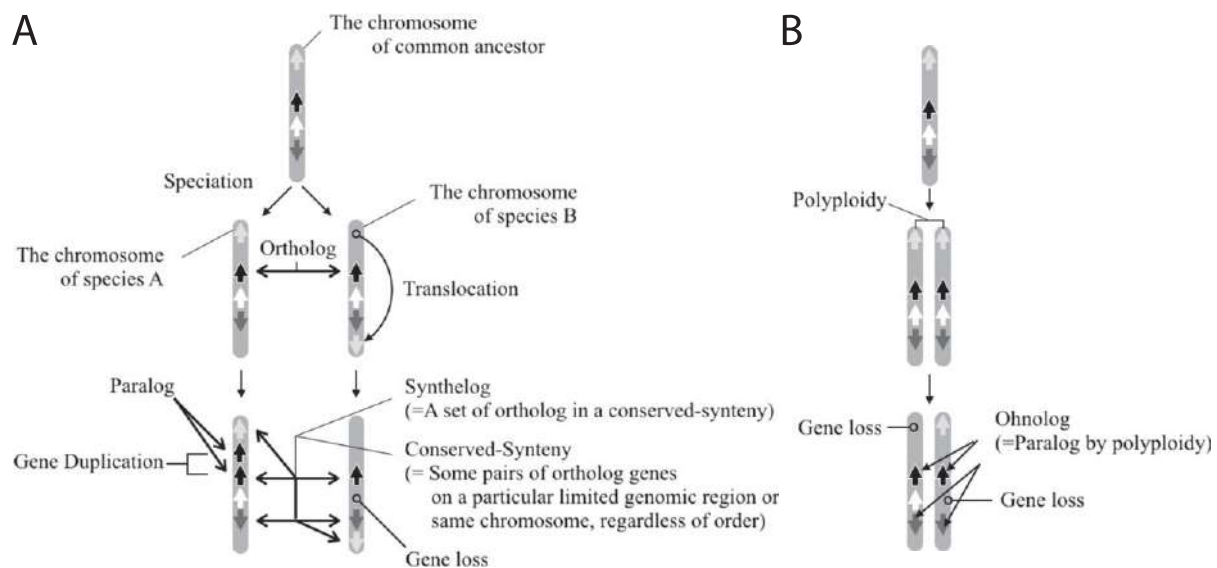


Fig. 3 Homolog Related Concepts. Illustrated various homolog related concepts shown in the main text.

Homology Search and Functional Prediction

One of the main purposes of searching homologous genes is prediction of gene function. This is based on the knowledge that homologous genes often have similar functions (Toh *et al.*, 1983). Since orthologs and paralogs are descendants of a common ancestral gene, it is estimated that both their functions and sequences are likely to be preserved. For this reason, determining orthologs and paralogs is the most basic step in prediction of gene function in bioinformatics (Cabaldón and Koonin, 2013).

Since the number of genes for which function has been elucidated experimentally is overwhelmingly smaller than the number of sequenced genes, it is frequently noted that gene prediction based on homology search may have problems with accuracy (Benner, 1999). Meaningless annotations like “similar to a gene of unknown function,” are also accumulated in genome databases, and these are likely to be reused in other databases. Despite this, the search for homologous genes has not lost its significance as the first step in annotation of genes (Joshi and Xu, 2007).

Various methods are still being developed for ortholog determination in comparative genomics because it is difficult to accurately decide true homologous genes pairs for the entire gene set between two species. Therefore, in practice, “bidirectional best-hits” (BBH, also called “mutual best-hits” or “reciprocal best-hits”) from all gene-sets are treated as conventional candidates of homologous genes (Tatusov *et al.*, 1997; Kristensen *et al.*, 2011). Although BBH are commonly used, a little study has been performed to ascertain their reliability. BBH are considered to be appropriate for use in prokaryotes (Wolf and Koonin, 2012). In contrast, a marked increase in mistakes is reported when BBH are used for comparisons among metazoans or plants, probably because these have many duplicated genes (Dalquen and Dessimoz, 2013). Ortholog determination in the comparison of whole gene sets is thus a product of some compromise. The fact that various ortholog-databases continue to be generated based on various criteria also implies the difficulty of ortholog determination (Dessimoz *et al.*, 2012).

Besides predicting gene function, as will be described later, determination of homologous genes is required to find conserved-synteny and linkage in the entire genome to reconstruct chromosomal evolution.

Whole Genome Duplication

Whole genome duplication (WGD) is an evolutionary event in which the whole gene set of an organism has been duplicated, probably by chromosomal polyploidy and its fixation. WGD is postulated as one of the major sources of evolutionary innovations.

Although polyploidy is a common phenomenon in eukaryotes, evolutionary fixed WGDs are exceedingly rare. In metazoa, known WGD events are almost limited to vertebrate lineages includes famous two round WGD (2R) which is occurred on the common ancestor of vertebrates (Ohno, 1970; Van de Peer *et al.*, 2009, 2010). In invertebrates, it is also suggested that some lineages have experienced WGD although those traces are not so clearly like vertebrates (Yoshida *et al.*, 2011; Kenny *et al.*, 2016). In plants and fungi, WGD occurs relatively more frequently than in animals (Van de Peer *et al.*, 2009) but there is still some bias.

Each type of duplicated genes created by WGD has named based on their forming mechanisms, respectively (Fig. 3(B)). Gene pairs produced by WGD are a special subset of paralogs (Wolfe, 2000). Because of their evolutionary significance as described above, they are sometimes distinguished by terms such as “ohnologs,” or its synonym “syntenic paralogs” (Wolfe, 2004; Schnable *et al.*, 2012). A similar word is “homoeologs” (or “homeologs” as another spelling) defined as homologous genes resulting from allopolyploidy (Glover *et al.*, 2016). The term “ohnolog” is derived from the name of Susumu Ohno who first proposed the hypothesis that gene duplication plays a key role in evolution (Ohno, 1970; Wolfe, 2000). Allopolyploid is a polyploid individual which composed of a chromosome set composed of two (or more) chromosome set. It formed from the hybridization of two (or more) separate but closely related species. The difference between the definitions of ohnolog and homoeolog is that the former refers to paralog made by autopolyploidy and the latter refers to paralog made by allopolyploidy. However, if polyploidization arose very long ago, it can not distinguish which mechanism truly occurred for creating the paralogous gene-sets. In such cases where autopolyploidy or allopolyploidy can not be distinguished, the term “paleolog” is sometimes used conveniently. Another similar term is “syntelogs” representing orthologous genes shared across species (or isolates for viruses) with a common syntenic genome location (Hatcher *et al.*, 2014).

The evolutionary process of WGD is one of the key themes in evolutionary genomics. In general, duplicated genes tend to be fixed only when those are to take on different functions but otherwise to be lost immediately after the duplication event. However, WGD is an event that a large number of duplication has occurred at once, it was quite difficult to investigate the process of gene losses until the era of genomics.

Because the function of an ohnolog pairs is presumably redundant each other just after WGD, it is theoretically expected that one of the duplicated genes to be lost gradually. Interestingly, the rate of gene losses may differ depending on the types of genes. For example in metazoa, it is known that ohnolog genes which relevant to embryonic development tend to be conserved as paralog gene-set (Holland *et al.*, 1994).

To distinguish a true pair of ohnolog or homoeolog from other common paralogs is technically difficult. Various methods have been devised to extract the set of ohnolog that were thought to be made by WGD. Several models have been proposed why the conservation rate of developmental genes after WGD is high (for example, Holland *et al.*, 1994). A future explanation will be awaited as to why the same trend is shown for cell cycle regulators.

To extract the ohnolog created by 2R in vertebrate evolution, Dehal and Boore utilized a genome of an invertebrate which is branched before the 2R. Using a graph-based method for this purpose, they have developed a way to extract homologous genes of

which number encoded in the genome of invertebrates and vertebrates at a ratio of approximately 1:2 to 1:4, respectively (Dehal and Boore, 2005).

Nishida and his colleagues developed a method to determine ohnolog gene-pair and examined the rate of gene loss after the WGD which occurred the common ancestor of teleost fish (teleost genome duplication: TGD) (Inoue *et al.*, 2015). As a result of the analysis for the obtained ohnolog gene set, they found that the rate of loss of ohnolog is roughly approximated by a double-exponential curve (2-phase model). According to their “2 phase model” of gene loss after TGD, duplicated genes are rapidly lost immediately after the WGD event, and then slowly lose at a roughly constant rate.

It is estimated that the rapid rate of gene loss in the first phase probably depends on the deletion of contiguous clusters of genes or large chromosomal segments (“block loss”), but the details have not been clarified yet.

It may be even more difficult to accurately identify the set of paralogs in an allopolyploid (homoeolog). The african clawed frog *Xenopus laevis* is proposed to be an allotetraploid animal that arose via the interspecific hybridization of diploid progenitors followed by subsequent genome doubling. In the sequence-decoding and reconstruction of the whole-genome of this frog, researchers noticed that the possibility that the subgenome derived from the two progenitor frogs each retained unique transposable elements in their genome and that those elements had been fossilized in each genome (Session *et al.*, 2016). They have actually found three of such transposable element and succeeded in separating reconstructed DNA sequences into two homoeologous subgenomes using those fossilized transposable elements as markers.

Homoeologs in the *X. laevis* were extracted from those separated subgenomes thus obtained and the tendency of the gene-loss after WGD was investigated. Interestingly, it was found that not only the developmental genes but also cell cycle regulators are tend to be conserved as paralogs after WGD.

Comparative Domain Structure of Genes

Many proteins retain several functional domains in their amino acid sequences. Particularly in the evolution of eukaryotic protein coding regions, the complexity of the domain structure may have played a significant role in gene evolution (Patthy, 1999; Chothia *et al.*, 2003). As mentioned in Section “Genome Structure and Types of Change”, the evolution of the domain architecture of a gene is a known mechanism for the changes in the partial structures of genes. The presence of genes with a lineage-specific or clade-specific domain structure makes it difficult to find true orthologous relationships (Uchiyama, 2006).

In eukaryotes, because genes are divided into several segmented structures of exons, duplication, shuffling, or deletion of functional domains can occur relatively easily by DNA recombination, potentially producing genes with a novel domain structure.

Choanoflagellates are unicellular eukaryotes and the closest known relatives of animals. King *et al.* (2008) decoded the genome of a choanoflagellate, *Monosiga brevicollis*, and compared the protein structure against those of metazoans. They found that choanoflagellates and metazoans shared many domains in their genomes, but the multidomain structure in their genes differed. These findings suggested that abundant domain shuffling followed the separation of the choanoflagellate and metazoan lineages. Such shuffling may have created animal-specific proteins during evolution.

Acquisition of a new domain structure may also be involved in the evolution of metazoan traits. In vertebrates, the domain structure of aggrecan protein, a major component of cartilage, was created by a domain-shuffling event that occurred in the common ancestry of vertebrates (Kawashima *et al.*, 2009).

Comparative Genomics of Intergenic Regions

Studies of intergenic regions have exploited comparative genomics methodologies. Such strategies are based on the known tendency that not only the coding-region but also other functional elements are likely to be well conserved in DNA among closely related species; thus, functional genomic regions can be predicted to some degree from sequence comparisons. Unlike the coding region, which can be predicted using transcriptome data, prediction of functional motifs in intergenic regions has been difficult. One of the established strategies is collating with known elements in databases such as TRANSFAC, JSPAR, and PAZAR. However, it is difficult to obtain strong candidates of the true element using this method alone because usually many false positives are detected. Additionally, this method is often not suitable for taxonomically separated species. Thus, comparative genome sequences provide a simple and widely applicable method for narrowing the candidates.

Nowadays, more direct and powerful methods for experimentally targeting such regions have been developed. Typical examples of these experimental methods include chromatin immunoprecipitation-seq (ChIP-seq), DNase-seq, formaldehyde-assisted isolation of regulatory elements-seq (FAIRE-seq), and assay for transposase-accessible chromatin-seq (ATAC-seq); of these, the former is a method for determining target positions on DNA of particular DNA-binding proteins, and the latter three are methods for detecting the open-chromatin regions on DNA. However, comparative sequences are still useful for surveying functional elements through intergenic regions because they can be combined with a variety of other methods.

Boffelli reviewed some examples of early studies of surveying functional elements in intergenic regions by comparative genomics among vertebrates (Boffelli *et al.*, 2004). Two representative researches introduced in the review are referred here. Nobrega and Rubin compared the 2.6-Mb region of the human genome with the homologous region of the mouse, frog, and puffer fish genomes and found 20 short conserved sequences in the vicinity of the *DACH* gene (Nobrega *et al.*, 2003). At least six of

the sequences have been confirmed to function as enhancers in mouse embryos. As another example, Lettice and colleagues compared the sequences in the neighboring area of *SHH* genes among humans, chickens, and puffer fish and found a sequence preserved in the introns of another gene separated by 1 Mb or more. The sequence was experimentally confirmed as a strong candidate enhancer of the *SHH* gene (Lattice *et al.*, 2003).

Some tools (e.g., VISTA and PipMaker) can be used to detect the conserved element in noncoding regions by comparative sequences (Mayor *et al.*, 2000; Schwartz *et al.*, 2000). Using these tools, comparative analyses of intergenic regions has become easy for two or more organisms if the taxonomic distances are selected appropriately. However, if the distance between two species is far, detecting functional motifs is still difficult. Future studies in comparative genomics are needed to overcome this unsolved problem.

Comparison of Gene Order

Linkage and Synteny

During evolution, chromosomal rearrangements change the gene order little by little as generations recapitulate due to deletions, insertions, duplications, inversions, and translocations of some chromosomal parts. Nevertheless, the signatures of the gene order in the genome of the common ancestor have been preserved for a long period in their offspring. Such traces of gene order are called “synteny”, which is defined as the presence at least two pairs of homologous genes on a particular limited genomic region or the same chromosome, regardless of order (Nadeau and Sankoff, 1998; Trachtulec and Forejt, 2001). Another similar concept is conserved linkage, which is defined as the synteny of two or more pairs of orthologous genes in the same order and orientation. However, the nomenclature for the evolutionarily conserved gene order is not unified. Although the original definition is as described above, it is often confusing (Passarge *et al.*, 1999). Therefore, researchers should consider the intentions of authors when consulting the literature. Some reports have referred to synteny as the conservation of blocks of gene orders within two sets of chromosomes or DNA fragments (scaffolds) that are being compared with each other. Since draft genome analysis has become popular, comparative genomic analysis of fragmental DNAs with only several kb or Mb can be easily performed. In such analyses, analysts cannot find the conservation of genes at the chromosomal level, but can find groups of orthologs in the relatively short fragmented DNAs of two organisms. Such synteny is sometimes called conserved synteny-block.

Long-Term Conservation of Synteny

Particularly in eukaryotes, synteny and linkage are often conserved for a surprisingly extended period. For example, many synteny blocks have been found in humans and mice, although the two species diverged from a common ancestor about 75 million years ago. The synteny blocks between these two species are approximately 65 Mb in length. The total length of these synteny blocks has been reported to cover over 90% of the entire genome of humans and mice.

Synteny is found even among the more evolutionarily distant species of eukaryotes, in contrast to bacteria, in which traces of gene orders disappeared almost immediately in some cases (Koonin *et al.*, 2000). However, as the evolutionary distance increased, the conservation degree of gene order, i.e., both the number and size of detectable blocks, tends to decrease gradually. When analyzing such slight traces of synteny, we can distinguish between two types of conserved gene orders that span different genomic scales, i.e., microsynteny and macrosynteny (Putnam *et al.*, 2007). Microsynteny is a local genomic region in which a very limited number of gene orders are conserved. Compared with microsynteny, macrosynteny describes much larger ranges, those on the scale of (partial) chromosomes and including several (or sometimes hundreds of) genes.

In a comparison of two markedly distant species, it is necessary to determine whether synteny and linkage found in modern organisms are indeed derived from the gene order in the common ancestor. If traces of synteny are very faint, we should focus on short stretches of conserved gene orders (microsynteny), composed of only two or three genes with some inserted genes, in order to consider the possibility of false positives due to coincidence. By validating the conservation in more than three species, false positives for surveying synteny blocks under such soft filtering can be eliminated.

In the following, three examples of micro- and macrosynteny are shown. In all cases, the human genome was compared with the genome of another organism. The first was a comparison of humans with lancelets, the invertebrate that is closest to vertebrates; the second was a comparison of humans with sea anemones, the non-bilateral animal closest to bilateral animals; and the third was a comparison of humans and fungi.

Lancelets, or amphioxii, belong to the subphylum cephalochordata and are classified as the phylum chordata together with the subphyla Tunicata and Vertebrata. Molecular phylogenetic analysis has revealed that cephalochordata were the first branched organisms among these three about 550 million years ago. Putnam and colleagues reported that more than 1000 microsynteny blocks were found between lancelets and humans under conditions allowing up to 20 genes to intervene between consecutive pairs of orthologs on the respective genomes. Although this may include some false positives, the fact that the number was over six times the discovery rate compared with the randomly reordered control strongly suggested that most of the gene orders were conserved since the common ancestor (Putnam *et al.*, 2008).

It is unclear why microsynteny blocks are preserved for so long-period. In an attempt to explain this problem, Irimia *et al.* (2011) found that genes encoding transcription factors were more likely to preserve the order with neighboring genes than other types of genes.

Based on this trend, they proposed a model in which regulatory elements of the intergenic region moved into the intron of the adjacent gene and in which the order of two genes was often fixed under the evolutionary timescale.

Domestication and Comparative Animals

Domestic Organisms are Invaluable Resources In Comparative Genomics

Domestic animals and agricultural plants are good research subjects for comparative genomic analysis between closely related species (Andersson and Georges, 2004). These strains have well-known traits that have been chosen over the years, and multiple DNA samples of closely related strains are available. To utilize the advantages of domestic animals, we typically attempt two types of analyses for comparing genomes to elucidate the domestication process of target organisms. In the first approach, the main subject is the elucidation of the origin of domestication and the history of subsequent propagation. In contrast, in the second approach, the aim is to find genes associated with traits relevant to domestication. In both of the approaches, data for sequence variations (primarily single nucleotide polymorphisms [SNPs]) are collected from several strains.

Origin and History of Domestication

Molecular phylogenetic analyses are used for the first approach to elucidate the domestication process. For this purpose, in addition to the genomes of multiple domestic strains, the genomes of the wild species closest to those domesticated strains are decoded, as are genomes of as many extant domestic strains as possible. When creating a species tree from amino acid or RNA sequences, using a long sequence created by concatenating a set of orthologs, rather than using only one gene, is the most common method (Delsuc *et al.*, 2005). For comparing closely related species, such as domestic organisms of different lines, SNP data gathered from comparative whole genomes are also useful for constructing a species tree (Guo *et al.*, 2013). In order to correct information on branching times, it is useful to calibrate the molecular phylogenetic tree based on paleontological data, such as fossil records and geological information. Some projects have succeeded in decoding even the DNA sequence from the fossils of ancestral animals to trace the process of domestication. In cases in which the fossil DNA of ancestral-related organisms can be obtained, it is possible to deduce the evolutionary rate up to the branching time with the extant taxa.

As an example of research on horse domestication, one research group succeeded in sequencing a draft genome from a horse bone recovered from permafrost dated to approximately 560–780 thousand years ago (Orland *et al.*, 2013). The group compared the obtained genome with the genomes of several modern horses and concluded that the common ancestor of all contemporary horses, zebras, and donkeys originated 4.0–4.5 million years ago.

Dogs are the first domesticated animal for which the genome was decoded. In the dog genome, many genes are missing compared with other mammals. Indeed, approximately 2000 genes may have been substantially mutated or lost from the dog genome after branching of dogs from the common ancestor of other mammals (Kirkness *et al.*, 2003; The Bovine Genome Sequencing and Analysis Consortium *et al.*, 2009). However, in the decoding of other domesticated animals, e.g., cows and pigs, such substantial gene losses have not been observed. Although it is difficult to determine the specific number of genes, and no reports have clearly described the number of genes lost during dog evolution, the results of comparative gene numbers among mammals in some papers are all similar; all results show that approximately 2000 genes have been lost in dogs. These findings related to gene losses may involve the evolution of dogs with various unique morphologies; further studies are needed to clarify these findings.

Notably, dogs were the first domesticated animals, separated from a group of wolves living with humans. The first branching time was estimated to be about 14,000 years ago, after which, two major changes occurred. A wide range of multiplications have also been made, particularly in the last 200 years.

Genome-Wide Association Study (GWAS) and Selective Sweep

The second approach is called GWAS, which uses sequence variations in the whole genome together with the phenotype and pedigree information to perform association analysis and to identify the loci of genes or regulatory elements that are important for the traits of interest (Zhang *et al.*, 2012). Each domestic animal has evolved specific traits, such as meat, egg, or fur productivity. Those traits are expected to be acquired by strong selection pressures. Therefore, by comparing the genomes among strains, the loci dominantly selected in each strain can be identified.

When analyzing the distribution of SNPs from a GWAS, “selective sweep” is frequently noted (Boffelli *et al.*, 2004). Selective sweep refers to a process by which a new advantageous mutation eliminates or reduces variation in linked neutral sites as it increases in frequency in the population (Nielsen *et al.*, 2005). This phenomenon is also called “genetic hitchhiking”. Since specific traits of domestic animals work predominantly by artificial selection, selective sweeps are expected to be observed in the neighboring region of the causative gene(s) of the trait. For example, according to the report of the first whole genome SNP analysis in cattle, the *MSTN* gene, which is related to muscle formation, was present in one of the regions where a strong signal of a selective sweep was detected in the genome of two beef cattle strains (The Bovine HapMap Consortium, 2009).

Another similar example is the genomic analyses of chickens. In chicken genomics, SNPs have been collected from eight different populations of domestic chickens, including broiler (meat-producing) and layer (egg-producing) lines as well as a red jungle fowl, the major wild ancestor. A selective sweep at the locus for thyroid-stimulating hormone receptor (TSHR) was performed for all domestic chickens. The *TSHR* gene is known to be involved in the regulation of photoperiod control of reproduction in birds and mammals. The authors demonstrated that mutations in this gene and its neighboring region were related to early domestication of chickens, as shown by the absence of strict regulation of seasonal reproduction in natural populations.

Future Directions

In the 2000s, whole genome analysis became possible in both prokaryotes and eukaryotes, and a comparative genome analysis was undertaken. Initially, there was a focus on the presence or absence of orthologs as well as the detection of conserved intergenic regions in major comparative genomics. Subsequently, the performance of the high-throughput DNA sequencing has improved, and it is now feasible to compare the genome sequences of multiple individuals for the same species or closely related strains. For this reason, the theories advocated in classical population genetics are now being actively tested with various actual data. This trend is expected to continue.

With the development of several “finishing” technologies, many projects aiming to reconstruct genomic sequences that had been stopped at the “draft” level may be able to be completed to the number of chromosomal arms in the near future (Eisenstein, 2015). If more genomes are completely sequenced, researchers in the field of evolutionary genomics will be able to elucidate the history of structural changes in the chromosome, such as chromosomal rearrangement or segmental duplication.

Closing Remarks

Both the terms “comparative genomics” and “evolutionary genomics” tend to be used when referring to a wide range of fields. In this report, the following definition was applied: “comparison of all gene sets in two or more species of organisms”. However, this definition has expanded to include comparisons of intergenic regions, SNPs, and chromosomal structures. Examples in this report focuses on comparative and evolutionary genomics of eukaryotes. Because comparative genomic analysis of bacteria also differs greatly from that of eukaryotes, it has not been described in this paper. Additionally, the examples provided herein are biased toward the study of animals; however, it is expected that the information presented in this report will still be helpful because the basic trends of the described studies were not different from those in fungi and plants. Notably, the genomic structure of unicellular eukaryotes has not yet been elucidated. It is possible that the basic structures of genomes in these organisms may deviate considerably from those of previously decoded genomes. Accordingly, further genome decoding and comparative analyses are needed.

Acknowledgements

I thank Yi-jun Luo, Oleg Simakov, Jun Inoue, and Hirokazu Chiba who kindly commented on the very early version of this document. I'm grateful to anonymous reviewer K.K. who reads the final manuscript intensively and gave useful comments. I greatly appreciate that Masanori Arita, Naoko Murakata and Takako Ohnuki supported the office procedure related to writing work. Part of this work was supported by the Kaken-hi Grant (no.: 17K19248).

See also: Algorithms for Strings and Sequences: Multiple Alignment. Bioinformatics Approaches for Studying Alternative Splicing. Comparative Genomics Analysis. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Gene Duplication and Speciation. Genome Alignment. Genome Annotation. Genome Databases and Browsers. Genome Informatics. Genome-Wide Association Studies. Homologous Protein Detection. Identification of Homologs. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Integrative Analysis of Multi-Omics Data. Metagenomic Analysis and its Applications. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Phylogenetic analysis: Early evolution of life. Phylogenetic Footprinting. Quantitative Immunology by Data Analysis Using Mathematical Models. Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?. Sequence Analysis. Whole Genome Sequencing Analysis

References

- Adams, M.K., *et al.*, 2000. The genome sequence of *Drosophila melanogaster*. *Science* 287, 2185–2195.
 Anderson, L., Georges, M., 2004. Domestic-animal genomics: Deciphering the genetics of complex traits. *Nat. Rev. Genet.* 5, 202–212.
 Banno, Y., Shimada, T., Kajiura, Z., Sezutsu, H., 2010. The silkworm – An attractive bioresource supplied by Japan. *Exp. Anim.* 59, 139–146.
 Begun, D.J., Lindfors, H.A., Thompson, M.E., Holloway, A.K., 2006. Recently evolved genes identified from *Drosophila yakuba* and *D. erecta* accessory gland expressed sequence tags. *Genetics* 172, 1675–1681.

- Benner, S.E., 1999. Errors in genome annotation. *Trends Genet.* 15, 132–133.
- Boffelli, D., Nobrega, M.A., Rubin, E.M., 2004. Comparative genomics at the vertebrate extremes. *Nat. Rev. Genet.* 5, 456–465.
- Brown, T.A., 2007. *Genomes 3*. Garland Science Publishing.
- Carneiro, M., Rubin, C., Di Palma, F., Albert, F.W., *et al.*, 2014. Rabbit genome analysis reveals a polygenic basis for phenotypic change during domestication. *Science* 345, 1074–1079.
- Chapman, J.A., Kirkness, E.F., Simakov, O., Hampson, S.E., *et al.*, 2010. The dynamic genome of *Hydra*. *Nature* 464, 592–596.
- Chothia, C., *et al.*, 2003. Evolution of the protein repertoire. *Science* 300, 1701–1703.
- The C. elegans Sequencing Consortium, 1998. Genome sequence of the nematode *C. elegans*: A platform for investigating biology. *Science* 282, 2012–2018.
- Dalloul, R.A., Long, J.A., Zimin, A.V., Aslam, L., *et al.*, 2010. Multi-platform next-generation sequencing of the domestic turkey (*Meleagris gallopavo*): Genome assembly and analysis. *PLoS Biol.* 8, e1000475.
- Dalquen, D.A., Dessimoz, C., 2013. Bidirectional best hits miss many orthologs in duplication-rich clades such as plants and animals. *Genome Biol. Evol.* 5, 1800–1806.
- The Bactrian Camels Genome Sequencing and Analysis Consortium, 2012. Genome sequences of wild and domestic bactrian camels. *Nat. Commun.* 3, 1202.
- Dehal, P., Boore, J.L., 2005. Two rounds of whole genome duplication in the ancestral vertebrate. *PLOS Biol.* 3, e314.
- Delsuc, F., Brinkmann, H., Philippe, H., 2005. Phylogenomics and the reconstruction of the tree of life. *Nat. Rev. Genet.* 6, 361–375.
- Demuth, J.P., *et al.*, 2006. The evolution of mammalian gene families. *PLOS ONE* 1, e85.
- Dessimoz, C., Gabaldón, T., Roos, D.S., Sonnhammer, E.L.L., *et al.*, 2012. Toward community standards in the quest for orthologs. *Bioinformatics* 28, 900–904.
- Dong, Y., Xie, M., Jiang, Y., *et al.*, 2013. Sequencing and automated whole-genome optical mapping of the genome of a domestic goat (*Capra hircus*). *Nat. Biotechnol.* 31, 135–141.
- Eisenstein, M., 2015. Startups use short-read data to expand long-read sequencing market. *Nat. Biotechnol.* 33, 433–435.
- Gabaldón, T., Koonin, E.V., 2013. Functional and evolutionary implications of gene orthology. *Nat. Rev. Genet.* 14, 360–366.
- Glover, N.M., Redestig, H., Dessimoz, C., 2016. Homoeologs: What are they and how do we infer them? *Trends Plant Sci.* 21, 609–621.
- Gregory, R.R., 2011. *The Evolution of the Genome*. Academic Press.
- Grimmelikhuijzen, C.J.P., *et al.*, 2007. The promise of insect genomics. *Pest Manag. Sci.* 63, 413–416.
- Guo, S., *et al.*, 2013. The draft genome of watermelon (*Citrullus lanatus*) and resequencing of 20 diverse accessions. *Nat. Genet.* 45, 51–58.
- Hatcher, E.L., Hendrickson, R.C., Lefkowitz, E.J., 2014. Identification of nucleotide-level changes impacting gene content and genome evolution in orthopoxviruses. *J. Virol.* 88, 13651–13668.
- Holland, P.W.D., Garcia-Fernandez, J., Williams, N.A., Sidow, A., 1994. Gene duplications and the origins of vertebrate development. *Dev. Suppl.* 125–133.
- Inoue, J., Sato, Y., Sinclair, R., Tsukamoto, K., Nishida, M., 2015. Rapid genome reshaping by multiple-gene loss after whole-genome duplication in teleost fish suggested by mathematical modeling. *Proc. Natl. Acad. Sci. USA* 112, 14918–14923.
- Jiang, Y., Xie, M., Chen, W., Talbot, R., *et al.*, 2014. The sheep genome illuminates biology of the rumen and lipid metabolism. *Science* 344, 1168–1173.
- Joshi, T., Xu, D., 2007. Quantitative assessment of relationship between sequence similarity and function similarity. *BMC Genom.* 8, 222.
- Kawashima, T., *et al.*, 2009. Domain shuffling and the evolution of vertebrates. *Genome Res.* 19, 1393–1403.
- Kenny, N.J., Chan, K.W., Nong, W., Qu, Z., *et al.*, 2016. Ancestral whole-genome duplication in the marine chelicerate horseshoe crabs. *Heredity* 116, 190–199.
- King, N., *et al.*, 2008. The genome of the choanoflagellate *Monosiga brevicollis* and the origin of metazoans. *Nature* 451, 783–788.
- Kirkness, E.F., Bafna, V., Halpern, A.L., Levy, S., *et al.*, 2003. The dog genome: Survey sequencing and comparative analysis. *Science* 301, 1898–1903.
- Koonin, E.V., 2005. Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.* 39, 308–309.
- Kristensen, D.M., Wolf, Y.I., Mushegian, A.R., Koonin, E.V., 2011. Computational methods for gene orthology inference. *Brief. Bioinform.* 12, 379–391.
- Lattice, L.A., *et al.*, 2003. A long-range *Shh* enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. *Hum. Mol. Genet.* 12, 1725–1735.
- Li, L., Stoeckert, C.J., Roos, D., 2003. OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Res.* 13, 2178–2189.
- Lindblad-Toh, K., Wade, C.M., Mikkelsen, T.S., Karlsson, E.K., *et al.*, 2005. Genome sequence, comparative analysis and haplotype structure of the domestic dog. *Nature* 438, 803–819.
- Mayor, C., *et al.*, 2000. VISTA: Visualizing global DNA sequence alignments of arbitrary length. *Bioinformatics* 16, 1046–1047.
- Mita, K., Kasahara, M., Sasaki, S., Nagayasu, Y., *et al.*, 2004. The genome sequence of silkworm, *Bombyx mori*. *DNA Res.* 11, 27–34.
- Morey, D.F., 1994. The early evolution of the domestic dog. *Am. Sci.* 82, 336–347.
- Nielsen, R., Williamson, S., Kim, Y., Hubisz, M.J., *et al.*, 2005. Genomic scans for selective sweeps using SNP data. *Genome Res.* 15, 1566–1575.
- Ohno, S., 1970. *Evolution by Gene Duplication*. Springer-Verlag New York Inc..
- Orland, L., *et al.*, 2013. Recalibrating *Equus* evolution using the genome sequence of an early Middle Pleistocene horse. *Nature* 499, 74–78.
- Owen, R., 1843. *Lectures on comparative anatomy and physiology of the invertebrate animals, delivered at the royal college of surgeons in 1843*. Longman, Brown, Green and Longman, London.
- Passarge, E., Horsthemke, B., Farber, R.A., 1999. Incorrect use of the term synteny. *Nat. Genet.* 23, 387.
- Patthy, L., 1999. Genome evolution and the evolution of exon-shuffling – A review. *Gene* 238, 103–114.
- Peng, X., Alfoldi, J., Gori, K., Eisfeld, A.J., *et al.*, 2014. The draft genome sequence of the ferret (*Mustela putorius furo*) facilitates study of human respiratory disease. *Nat. Biotechnol.* 32, 1250–1255.
- Pontius, J.U., Mullikin, J.C., Smith, D.R., *et al.*, 2007. Agencourt Sequencing Team, Initial sequence and comparative analysis of the cat genome. *Genome Res.* 17, 1675–1689.
- Putnam, N.H., *et al.*, 2008. The amphioxus genome and the evolution of the chordate karyotype. *Nature* 453, 1064–1071.
- Rubin, C., Zody, M.C., Eriksson, J., Meadows, J.R.S., *et al.*, 2010. Whole-genome resequencing reveals loci under selection during chicken domestication. *Nature* 464, 587–591.
- Rubin, C., Megens, H., Barrio, A.M., Maqbool, K., *et al.*, 2012. Strong signatures of selection in the domestic pig genome. *Proc. Nat. Acad. Sci. USA* 109, 19529–19536.
- Schnable, J.C., Freeling, M., Lyons, E., 2012. Genome-wide analysis of syntenic gene deletion in the grasses. *Genome Biol. Evol.* 4, 265–277.
- Schwartz, S., *et al.*, 2000. PipMaker – A web server for aligning two genomic DNA sequences. *Genome Res.* 10, 577–586.
- Session, A.M., Uno, Y., Kwon, T., Chapman, J.A., *et al.*, 2016. Genome evolution in the allotetraploid Frog *Xenopus laevis*. *Nature* 538, 336–343.
- Simakov, O., Kawashima, T., 2017. Independent evolution of genomic characters during major metazoan transitions. *Dev. Biol.* 427, 179–192.
- Simakov, O., Marletaz, F., Cho, S., *et al.*, 2013. Insights into bilaterian evolution from three spiralian genomes. *Nature* 493, 526–531.
- Sonnhammer, E.L., Koonin, E.V., 2002. Orthology, paralogy and proposed classification for paralog subtypes. *Trends Genet.* 18, 619–620.
- Tatusov, R.L., Koonin, E.V., Lipman, D.J., 1997. A genomic perspective on protein families. *Science* 278, 631–637.
- The Bovine Genome Sequencing and Analysis Consortium, Elsik, C.G., Tellam, R.L., *et al.*, 2009. The genome sequence of taurine cattle: A window to ruminant band evolution. *Science* 324, 522–528.
- The Bovine HapMap Consortium, 2009. Genome-wide survey of SNP variation uncovers the genetic structure of cattle breeds. *Science* 324, 528–532.
- Toh, H., Hayashida, H., Miyata, T., 1983. Sequence homology between retroviral reverse transcriptase and putative polymerases of hepatitis B virus and cauliflower mosaic virus. *Nature* 305, 827–829.
- Trachtulec, Z., Forejt, J., 2001. Synteny of orthologous genes conserved in mammals, snake, fly, nematode, and fission yeast. *Mammalian Genome.* 12, 227–231.
- Uchiyama, I., 2006. Hierarchical clustering algorithm for comprehensive orthologous-domain classification in multiple genomes. *Nucleic Acids Res.* 34, 647–658.

- Van de Peer, Y., Maere, S., Meyer, A., 2009. The evolutionary significance of ancient genome duplications. *Nat. Rev. Genet.* 10, 725–732.
- Van de Peer, Y., Maere, S., Meyer, A., 2010. 2R or not 2R is not the question anymore. *Nat. Rev. Genet.* 11, 166.
- Wade, C.M., Giulotto, E., Sigurdsson, S., Zoli, M., *et al.*, 2009. Genome sequence, comparative analysis, and population genetics of the domestic horse. *Science* 326, 865–867.
- Wolf, Y.I., Koonin, E.V., 2012. A tight link between orthologs and bidirectional best hits in bacterial and archaeal genomes. *Genome Biol. Evol.* 4, 1286–1294.
- Wolfe, K., 2000. Robustness – It's not where you think it is. *Nat. Genet.* 25, 3–4.
- Wolfe, K., 2004. Evolutionary genomics: Yeasts accelerate beyond BLAST. *Curr. Biol.* 14, R392–R394.
- Xia, Q., Zhou, Z., Lu, C., Cheng, D., *et al.*, 2004. A draft sequence for the genome of the domesticated silkworm (*Bombyx mori*). *Science* 306, 1937–1940.
- Yoshida, M., Ishikura, Y., Moritaki, T., Shoguchi, E., *et al.*, 2011. Genome structure analysis of molluscs revealed whole genome duplication and lineage specific repeat variation. *Bioinformatics*, 483, 63–71.
- Zhang, H., Wang, Z., Wang, S., Li, H., 2012. Progress of genome wide association study in domestic animals. *J. Anim. Sci. Biotechnol.* 3, 26.
- Zhao, L., Saelao, P., Jones, C.D., Begun, D.J., 2014. Origin and spread of de novo genes in *Drosophila melanogaster* populations. *Science* 343, 769–772.

Genome Alignment

Tetsushi Yada, Kyushu Institute of Technology, Fukuoka, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Genome alignment is the alignment of genome sequences; it can be considered as a way to reformat a genome sequence dataset in order to facilitate the comparison of evolutionarily related genomes. We can also consider a genome alignment as a prediction of homology (i.e., shared origin, see the next section) at the nucleotide level between two or more genomes (Dewey and Pachter, 2006; Dewey, 2012).

Regarding the question of why we (re)construct genome alignments, the following sentence would neatly summarize its answers that have been given thus far (e.g., Dubchak and Pachter, 2002; Frazer *et al.*, 2003; Brudno and Dubchak, 2005; Dewey and Pachter, 2006; Margulies and Birney, 2008; Dewey, 2012; Thompson, 2016a; Bleidorn, 2017): “we (re)construct genome alignments *in order to exploit the evolutionary information inscribed in the genome sequences.*” (Note: This sentence itself is our homage to Dr. Hitoshi Kihara, a renowned plant geneticist who once wrote: “The history of the earth is recorded in the layers of its crust; the history of all organisms is *inscribed* in the chromosomes” (originally in Japanese) (Kihara, 1947)).

Here, let us elaborate on this sentence. The genome sequences of all extant living organisms are, after all, the products of billions of years of evolution, starting from the genome(s) of the most recent common ancestor(s) of life.

During the course of evolution, the genome sequences have gradually (or suddenly) changed, and have accumulated mutations such as base substitutions, insertions, deletions (e.g., Graur and Li, 2000), and large-scale rearrangements such as inversions, translocations, duplications, etc. (e.g., Lupski, 2007; Lynch, 2007; Gu *et al.*, 2008; Lee *et al.*, 2017). In other words, information on the evolution is ‘inscribed’ in the genome sequences in the form of accumulated mutations. Broadly speaking, the amount of difference between two genome sequences should be bigger as they diverge earlier. Therefore, by comparing the genome sequences of two species (or individuals) and quantifying the differences, we can estimate the *evolutionary distance* between them, that is, how closely (or distantly) the two species (or individuals) are related to each other (e.g., Waterston *et al.*, 2002; Gibbs *et al.*, 2004). If we extend the comparison to that of three or more genomes, we can infer the phylogenetic relationships among them, that is, the order in which each genome (or lineage) diverged from others (e.g., Clark *et al.*, 2007; Sheng *et al.*, 2011; Sahl *et al.*, 2012; Prabha *et al.*, 2014). Moreover, by meticulously examining the patterns of differences between the genomes, we can identify what types of mutations (like substitutions, insertions, deletions, inversions, translocations and duplications) occurred, and how and when (or where in the phylogenetic tree of the genomes) they occurred (e.g., Kent *et al.*, 2003; Watanabe *et al.*, 2004; Ma *et al.*, 2006; Clark *et al.*, 2007; Wang *et al.*, 2015). Thus, genome alignments provide integral materials for the study of molecular evolution and phylogenetics (e.g., Gascuel, 2005; Saitou, 2013).

But this is not all; genome alignments also enable us to identify functionally important regions of the genomes. According to the neutral theory of molecular evolution (Kimura, 1968, 1983), non-functional parts of a genome by default evolve neutrally, i.e., without any selective pressure, and mutations on them are occasionally fixed in the population due to genetic drift. On the other hand, if the parts have some biologically important functions, mutations on them are likely to disrupt the functions and thus less likely to be fixed than neutrally (Kimura and Ohta, 1974). Therefore, the alignment of genomes that are appropriately evolutionarily far apart will highlight potentially functional parts of the genomic sequences as those showing significantly less frequent changes than the neutral background (e.g., Kellis *et al.*, 2003; Birney *et al.*, 2007; Stark *et al.*, 2007; Tseng and Tompa, 2009; Hupalo and Kern, 2013). In contrast, if the parts show significantly more frequent changes than the neutral background, the parts may have been under (continuous) positive selection pressure, maybe because they are involved in the responses to incessantly changing environments, including interactions between hosts and pathogens or parasites and between males and females (e.g., Hughes and Nei, 1988; Swanson and Vacquier, 2002; Sackton *et al.*, 2007; Wong, 2011).

For that matter, the conservation of synteny (i.e., colocalization within a single chromosome, see Section “Conserved Synteny and Collinearity”) of genomic elements (including genes) during a long evolutionary period may indicate either that their synteny itself is functionally significant and/or that the region in between the elements is also functionally essential (e.g., Ferrier and Holland, 2001; Hurst *et al.*, 2004; Kikuta *et al.*, 2007; Engström *et al.*, 2007; Dewey, 2012; Schnable, 2015; Bleidorn, 2017), because otherwise the synteny would eventually be disrupted by some genomic rearrangements. Also in this sense, genome alignments are useful because they enable us to reveal the regions of conserved synteny and possibly to infer the large-scale structures of ancestral genomes (e.g., Tesler, 2002; Kent *et al.*, 2003; Ma *et al.*, 2006, 2008).

Because of such multi-faceted utility, genome alignment has been a corner stone of comparative genomics, whose main purposes are to identify conserved (and thus functionally important) elements and regions in the genomes and to study underlying evolutionary mechanisms and forces operating on the genomes (e.g., Frazer *et al.*, 2003; Brudno and Dubchak, 2005; Margulies and Birney, 2008; Dewey, 2012).

During the past two decades, genome alignment has been a subject of intensive study and vigorous efforts, and numerous algorithms, methods, and programs have been published thus far. Therefore, it is virtually impossible to exhaust the past studies. Hence we will overview the past developments in this area of research while discussing a number of methods that particularly caught our attention. If the readers want more extensive, or nearly comprehensive, information on this subject, we recommend them to read not only this article but also some other reviews, such as (Dubchak and Pachter, 2002; Frazer *et al.*, 2003; Brudno and Dubchak, 2005; Jayaraj, 2006; Dewey and Pachter, 2006; Margulies and Birney, 2008) for early developments, and (Dewey, 2012;

Earl *et al.*, 2014; Kehr *et al.*, 2014; Thompson, 2016a) for relatively recent developments. Moreover, because of the space limitation, we will focus on the methods to align two or more long sequences (at least on the order of 100 kb), and will not discuss methods to align shorter sequences, such as the standard aligners for, e.g., protein-coding or RNA genes (reviewed, e.g., in Kumar and Filipinski, 2007; Notredame, 2007; Aniba *et al.*, 2010; Löytynoja, 2012; Iantorno *et al.*, 2014) or methods to map the reads from Next-Generation Sequencers to a reference genome (reviewed, e.g., in Cordero *et al.*, 2012; Lee and Tang, 2012; Bahassi and Stambrook, 2014; Mielczarek and Szyda, 2016; Smith and Yun, 2017). Those who are interested in these topics should refer to the reviews cited just above and/or the relevant articles in this Encyclopedia.

Background/Fundamentals

This section provides some basic information that is necessary for discussing genome alignment.

Homology

In molecular evolution (including evolutionary genomics), the term “homology” means the sharing of evolutionary origin between two genomic elements (or among three or more elements). That is, two genomic elements are “homologous” to each other if they are descended from a common ancestral element (e.g., Fitch, 1970, 2000). The concept of homology can be extended (or reduced) to the nucleotide level (Fitch, 2000; Dewey and Pachter, 2006; Dewey, 2012).

In the traditional matrix (row-column) format of a sequence alignment, each column consists of homologous nucleotides from the aligned sequences, and each row represents an aligned sequence interspersed with gaps that indicate the lack of the corresponding nucleotides. Usually, each nucleotide is represented by a one-letter abbreviation code of its base (‘A’ for adenine, ‘C’ for cytosine, ‘G’ for guanine, and ‘T’ for thymine), and each gap is represented by a hyphen (‘-’). Actually, this traditional matrix representation is not enough to represent a genome alignment, because it is not flexible enough to accommodate genome rearrangements (such as inversions) and duplications. Nevertheless, the matrix representation still remains a crucial building block of almost all formats of genome alignments.

In molecular evolution, the homology relationship can be broadly classified into three categories: “orthology”, “paralogy”, and “xenology” (Fitch, 1970, 2000; Gray and Fitch, 1983). The two homologous elements are “orthologous” if they diverged from the common ancestor as a result of speciation; they are “paralogous” if their divergence is due to duplication; they are “xenologous” if at least one horizontal transfer contributed to the evolutionary path between them.

When discussing different types of genome alignments, it is sometimes convenient to further sub-classify the categories of orthology and paralogy, because such sub-classes could help clarify particular goals of different methods. Especially, it is important to note that, in general, orthology is not necessarily a one-to-one correspondence, because a genome element may have duplicated after speciation. “Topoorthology” (or “positional orthology”) is a special subclass of orthology (Dewey and Pachter, 2006; Dewey, 2012); a pair of orthologous elements are topoorthologous if the evolutionary path between them does not go through any duplication event that changed the genomic position of the element. (Such duplication events include retrotranspositions and interspersed segmental duplications, but do not include tandem duplications.) In other words, a pair of topoorthologous elements have never changed their genomic positions due to duplications (though they may have changed positions via rearrangements,) since their divergence due to speciation. This notion of “topoorthology” may be closer to what some readers might have considered as “orthology”; because topoorthologous genes usually share the same ancestral genomic context, they are much more likely to conserve functions than non-topoorthologous orthologous genes are (Dewey, 2012).

Conserved Synteny and Collinearity

During the course of evolution, a genome has undergone rearrangements, thus some long-range organizations of the genomic sequence have been disrupted. When aligning whole genomes, we are supposed to align segments that have escaped (especially large-scale) rearrangements since the divergence of the genomes. Thus, some terms have been defined to represent such situations (e.g., Gregory *et al.*, 2002; Frazer *et al.*, 2003; Margulies and Birney, 2008). When two or more elements in a genome reside on the same chromosome, they are referred to as “syntenic”. When two or more elements reside on a single chromosome in one species and their orthologous counterparts also reside on a single chromosome in the other species, the situation is referred to as “conserved synteny”. It should be noted that conserved syntenic blocks (or regions) may have undergone small-scale rearrangements confined within them, thus the order among the elements may not be conserved. When the order is also conserved within the pair of conserved syntenic regions, Gregory *et al.* (2002) (and also Frazer *et al.*, 2003) referred to them as “conserved segments”. However, they did not mention the relative orientations of the elements, which may change via (small-scale) inversions. When both the order and the orientations are also conserved within a pair of conserved syntenic regions, we usually refer to the regions as having “conserved collinearity” or, for short, being “collinear” (to each other). (Conserved) collinearity implies that the regions have not undergone any rearrangements since their divergence. Considering these original definitions, the term “synteny mapping” should rigorously mean pairing up conserved syntenic blocks in two genomes (or its extension to more genomes), though the term has come to imply finding collinear regions (as pointed out by Margulies and Birney, 2008). Moreover, the notions of

conserved syntenic and collinearity can be extended to any homologous segments (including paralogous ones), not limited to orthologous ones. Thus, [Margulies and Birney \(2008\)](#) proposed to use the term “reconstructing homologous collinearity” instead of “syntenic mapping”. It should be noted, however, that some programs are *literally* syntenic mappers, because they do not care much about the order or orientations of the elements within each block. Therefore, we will refer to this kind of programs collectively as “syntenic mappers”, while keeping in mind that some of them are actually programs to reconstruct homologous collinearity. (When necessary, we will clearly distinguish whether a program reconstructs conserved syntenic or collinearity).

Local, Global, and “Glocal” Alignment

Traditionally, there have been two types of sequence alignment methods, local and global. Local alignment methods align only those regions of the sequences that are suspected to be homologous (based on their similarities). In contrast, global alignment methods “force” to align the sequences from the beginning to the end. Merits of local alignment are: (i) it could avoid false alignment of non-homologous regions, especially those neighbouring homologous regions; and (ii) it can even align regions that are no longer collinear as a result of rearrangements. Demerits of local alignment are: (i) it could miss weakly homologous regions located near highly homologous regions; and (ii) it could mistake by-chance similarities as homologies, even if they are isolated from other homologous regions. Pros and cons of global alignment are opposite from those of local alignment.

In 2003, an intermediate concept, “glocal alignment” was proposed ([Brudno et al., 2003c](#)). Glocal alignment is intended to be like local alignment in that it can align those regions that have undergone rearrangements and duplications (and, in addition, suggests the presence of these events), and like global alignment in that it can potentially detect weak homologies if they are neighbouring strongly homologous regions.

Originally, [Brudno et al. \(2003c\)](#) defined a (pairwise) glocal alignment as “a series of operations that transform one sequence into the other”. They considered that the operations include insertions, deletions, point mutations, inversions, translocations and duplications, and that the total edit distance between the two sequences is the sum of penalties incurred by individual operations.

Topic

Major Challenges on Genome Alignment

When aligning genome sequences, some major challenges need to be overcome (e.g., [Dubchak and Pachter, 2002](#); [Margulies and Birney, 2008](#); [Thompson, 2016a](#)); especially pressing among them are: (1) time (and space) requirement, and (2) rearrangements and duplications.

- (1) When optimising simple objective functions for pair-wise sequence alignment (PWA), dynamic programming (DP) algorithms have long been used, such as Needleman-Wunsch (NW) ([Needleman and Wunsch, 1970](#)) for global alignment, Smith-Waterman (SW) ([Smith and Waterman, 1981](#)) for local alignment, and their various improvements (e.g., [Hirschberg, 1975](#); [Gotoh, 1982](#); [Myers and Miller, 1988](#)). These algorithms have the time complexity quadratic in the sequence length and space complexity quadratic or linear in the sequence length. When aligning sequences such as protein-coding genes and RNA genes, which are not so long, they do their jobs in a reasonable amount of time. However, when aligning genome sequences, each of which consists of millions to billions of bases, these DP algorithms could consume an enormous amount of time, making such computation impractical.

One heuristic to avoid this problem is to first find short regions showing high similarity (commonly called “seeds”) via a fast heuristic method and to restrict the (2-dimensional) search space of DP to those around, and/or in between, the seeds. This could drastically reduce the computational time. There are two major approaches based on this “seeding” heuristics (e.g., [Dubchak and Pachter, 2002](#); [Jayaraj, 2006](#); [Dewey and Pachter, 2006](#); [Margulies and Birney, 2008](#); [Dewey, 2012](#); [Thompson, 2016a](#)): one is to extend each seed on both sides via some modified version of the SW algorithm (“seed-and-extend”), somewhat similarly to gapped BLAST ([Altschul et al., 1997](#)), and the other is to chain mutually consistent seeds first and then to use the chained seeds as “anchors” and to align the regions in between the neighbouring anchors (“seed-and-chain”, or “anchoring”). (For the chaining algorithms in early days, see, e.g., [Ohlebusch and Abouelhoda, 2005](#)). Usually, the regions in between the neighbouring anchors are globally aligned, but some aligners locally align the regions. (The “seed-and-extend” strategy usually produces a set of local alignments, which may sometimes be chained afterwards (e.g., [Kent et al., 2003](#))). When aligning three or more sequences, even if the sequences are not so long, it could take an enormous amount of time even for a DP to exhaustively search the alignment space (e.g., [Gusfield, 1997](#)); in fact, it has been shown that such a problem with some-of-pairs scoring is NP hard ([Wang and Jiang, 1994](#)). Thus, to reconstruct a multiple sequence alignment (MSA), we must resort to some heuristics (e.g., [Gusfield, 1997](#); [Notredame, 2007](#); [Kemena and Notredame, 2009](#); [Löytynoja, 2012](#)). Among the most popular heuristics is the progressive alignment ([Hogeweg and Hesper, 1984](#); [Feng and Doolittle, 1987](#)), which iteratively performs pairwise alignment of sequences or sub-alignments at each internal node of a guide tree, from leaves upwards, and finally constructs an MSA of all input sequences at the root. Multiple genome aligners that have been developed thus far mostly combine some heuristics for MSA with either “seed-and-extend” or “anchoring” heuristics.

- (2) Traditionally, sequence alignment algorithms have been built under simple models that only consider base substitutions, insertions, and deletions (e.g., [Gusfield, 1997](#); [Kumar and Filipinski, 2007](#); [Notredame, 2007](#); [Löytynoja, 2012](#)), thus they have only been able to produce collinear alignments, where the order and orientations of the homologous elements are conserved. However, it is undeniable that genome sequences have undergone other types of mutations such as rearrangements (inversions, translocations, etc.) and duplications (e.g., [Kent et al., 2003](#); [Pevzner and Tesler, 2003](#); [Ma et al., 2006](#); [Lupski, 2007](#); [Lynch, 2007](#); [Clark et al., 2007](#); [Lee et al., 2017](#)). Therefore, when aligning genome sequences, it becomes inevitable to handle these rearrangements and duplications, in addition to substitutions, insertions, and deletions that have been traditionally dealt with. Currently, a common strategy to address this challenge is splitting the genome alignment problem into two sub-problems (e.g., [Dewey and Pachter, 2006](#); [Margulies and Birney, 2008](#); [Dewey, 2012](#)): (1) finding the regions of conserved collinearity (or conserved synteny), and (2) aligning each collinear (or syntenic) region at the nucleotide level. In early days, these two subproblems were addressed by separate programs (or separate components of a pipeline) (e.g., [Couronne et al., 2003](#); [Brudno et al., 2004](#)), namely, a homologous collinearity reconstruction method (or a “synteny mapper”) and a collinear aligner. Around since two important concepts, “glocal alignment” ([Brudno et al., 2003c](#)) (see also Section “Local, Global, and “glocal” Alignment” above) and a “threaded blockset” ([Blanchette et al., 2004](#)), were proposed, however, many research groups have developed “non-collinear aligners”, each of which addresses the two subproblems in a single run (see also the “Approaches” Section below). Such programs may be advantageous, because they may potentially become able to refine the result of the first component taking account of the result of the second component (e.g., [Paten et al., 2011](#)). Nevertheless, the development of collinear genome aligners is still an area of active research, because they are important components of the non-collinear aligners.

[Blanchette et al. \(2004\)](#) defined the “threaded blockset” as a set of “blocks” that are “threaded” by each of the original input sequences. (Here, a “block” is a set of mutually homologous collinear segments from the original input sequences (or their reverse complements), and a sequence is said to “thread” the blockset if the blockset contains completely and non-redundantly the sequence as a whole. It should be noted that a block is permitted to consist only of a segment of a single sequence.) In our view, this threaded blockset could be regarded as a representation of a glocal alignment, if the threading information is explicitly presented. In fact, a threaded blockset is highly similar to a graph-based alignment (reviewed e.g. in [Kehr et al., 2014](#)), which could be considered as a sort of glocal alignment (as argued by [Darling et al., 2010](#)). And, indeed, the developers of a graph-based aligner (the A-Brujin alignment method) even stated that their aligner “is able to automatically generate threaded blockset for genomic sequence alignment” ([Raphael et al., 2004](#)).

A yet another similar, but distinct, class of non-collinear aligners is the positional homology genome alignment method ([Darling et al., 2004, 2010](#)), which may be traced back to MUMmer (version 2) ([Delcher et al., 2002](#)). Like glocal aligners, they can deal with rearrangements, such as inversions and translocations. Unlike glocal aligners, however, they will not attempt to align paralogous regions within the same genome, because their primary goal is to align positional orthologs (topoorthologs) of different genomes. This type of aligner will be useful if the focus of the downstream study is on identifying and analyzing topoorthologs, although it may not be suitable for detailed analyses of duplication-related phenomena ([Darling et al., 2010](#)).

Major Steps in Genome Alignment

As somewhat discussed above, alignment of genome sequences poses several challenges. Therefore, traditionally, genome alignments have been constructed in several steps. The major steps involved in pairwise alignment are: (1) reconstructing homologous collinearity (e.g., [Margulies and Birney, 2008](#)) (also known as “synteny mapping” (e.g., [Thompson, 2016a](#)) or “homology mapping” (e.g., [Dewey and Pachter, 2006](#); [Dewey, 2012](#))); then, for each collinear region, (2) finding “seeds” (i.e., short, highly similar pairs (or sets) of subsequences), (3) selecting promising seeds, (4) completing the alignments around the selected seeds (for seed-and-extend) or in between them (for anchoring); and (5) displaying and viewing the resulting genome alignment.

Many programs perform steps (2)–(3) or (2)–(4) recursively, and gradually narrows down the search space, aiming to enhance the sensitivity and/or accuracy of the resulting alignment, and/or computational speed. When a pipeline consists mainly of a synteny mapper and a collinear aligner, all the steps from (1) to (4) are usually performed separately. Many graph-based aligners also seem to follow this practice; in this case, step (1) is usually performed by a component of the program separate from the component dealing with steps (2)–(4). Some non-collinear aligners, however, functionally merges step (1) with steps (2)–(3). Furthermore, some seed-and-extend type local aligners perform steps (2)–(4) first and, when necessary, performs step (1) later, using the output of step (4).

As briefly mentioned above, multiple genome alignment involves a heuristic MSA step (such as progressive alignment) in addition to the aforementioned steps (1)–(4). (Note: Here, we consider step (5) as a separate, absolutely final step.) Therefore, another dimension of variety arises regarding when to perform the MSA-dedicated heuristics; it could be either before step (1), before step (2) or (3), before step (4), or after all steps, (1)–(4).

In early days, step (1) mostly focused on finding orthologous collinear (or syntenic) regions. However, some later methods can also find paralogous collinear (or syntenic) regions.

Step (2) can vary widely depending on the types of seeds and the types of data structure to store the seeds. Seeds can be un-spaced (i.e., consecutive) or spaced (i.e., containing some positions that will be ignored), and also can be exact matches or inexact matches, or even with some gaps. Particular types of spaced seed patterns were shown to enhance the sensitivity of the alignment as well as computational speed, because they substantially reduce the correlation between overlapping seeds

(Ma *et al.*, 2002); these seed patterns have been used by some aligners. Regarding the seed-storing data structure, suffix-trees (or suffix-arrays) and hash-tables seem to be most widely used. Suffix-trees are good at finding exact or nearly exact matches (e.g., Delcher *et al.*, 1999), whereas hash-tables can be used for finding considerably in-exact matches (e.g., Schwartz *et al.*, 2003). A notable seed-finding method is CHAOS, which can find even gapped matches and use them as seeds (Brudno *et al.*, 2003a); it is thus expected to work also on genomes that are somewhat distantly related to each other. More recently, a new type of seeds, “adaptive seeds”, have been proposed (Kielbasa *et al.*, 2011). An adaptive seed is a short match (of any length) that occurs in a target genome less frequently than a specified threshold. They are robust against regional heterogeneity of base composition and repetitive elements.

Step (3) is performed either by imposing some thresholds on each seed (mainly for seed-and-extend) and/or by chaining mutually consistent seeds (for “anchoring”).

Methods for step (4) could be classified in terms of the broad category (local, global, or probabilistic), the scoring scheme (a traditional score with match rewards, mismatch penalties and affine gap penalties, a consistency score (e.g., Paten *et al.* 2008a), an expected accuracy (e.g., Bradley *et al.* 2009), etc.), and so forth. Some methods even apply external collinear aligners (e.g., Thompson *et al.*, 1994; Edgar, 2004; Katoh and Toh, 2008) to this step.

Regarding step (5), there are some programs (viewers and browsers) dedicated particularly to displaying the alignment (e.g., Mayor *et al.*, 2000; Schwartz *et al.*, 2000; Frazer *et al.*, 2004; Elnitski *et al.*, 2010; Nguyen *et al.*, 2014; Poliakov *et al.*, 2014). But many aligners come with their own internal (or accompanying) viewers (e.g., Darling *et al.*, 2010). In addition, some web-sites, such as Ensembl (Herrero *et al.*, 2016), Galaxy (Afgan *et al.*, 2016), UCSC Genome Browser (Miller *et al.*, 2007; Tyner *et al.*, 2017), and VISTA browser (e.g., Poliakov *et al.*, 2014), store a number of genome alignments that are too large to be reconstructed by a single researcher (or laboratory), and their own dedicated browsers enable us to view the alignments in interactive and convenient manners, possibly with some annotations.

Approaches

To the best of our knowledge, the development of genome aligners started with MUMmer (Delcher *et al.*, 1999), a pairwise collinear local genome aligner employing “seed-and-chain” strategy based on a suffix-tree. (Note: MUMmer evolved into a non-collinear local aligner afterwards (Delcher *et al.*, 2002; Kurtz *et al.*, 2004)). After that, some (either local or collinear global) pairwise genome aligners were developed (e.g., Batzoglou *et al.*, 2000; Kent and Zahler, 2000; Ma *et al.*, 2002; Bray *et al.*, 2003; Schwartz *et al.*, 2003). Then, in 2002, “multiple genome aligner” (MGA), the first collinear multiple global genome aligner, was published (Höhl *et al.*, 2002), and some (either global or local) collinear multiple aligners followed (e.g., Brudno *et al.*, 2003a, b; Blanchette *et al.*, 2004; Bray and Pachter, 2004). (Note: Later, MLAGAN and TBA evolved into non-collinear multiple aligners; the former is glocal (Dubchak *et al.*, 2009), and the latter is local (discussed, e.g., in: Raphael *et al.*, 2004; Ovcharenko *et al.*, 2005; Margulies *et al.*, 2007)). It was in 2003 that the first (pairwise) non-collinear “glocal” genome aligner, Shuffle-LAGAN, was published (Brudno *et al.*, 2003c). And, in 2004, some non-collinear multiple genome aligners of different types were published (e.g., Darling *et al.*, 2004; Raphael *et al.*, 2004). Since then, non-collinear multiple aligners have been in the mainstream of genome aligner developments (e.g., Darling *et al.*, 2010; Angiuoli and Salzberg, 2011; Paten *et al.*, 2011), while pairwise aligners and collinear multiple aligners have also been vigorously developed with a variety of advanced features (e.g., Harris, 2007; Rausch *et al.*, 2008; Bradley *et al.*, 2009; Paten *et al.*, 2009; Kryukov and Saitou, 2010; Nakato and Gotoh, 2010; Kielbasa *et al.*, 2011).

Before non-collinear genome aligners became popularized, “synteny mappers” played a crucial role in whole-genome alignment pipelines (e.g., Couronne *et al.*, 2003; Kent *et al.*, 2003; Brudno *et al.*, 2004; Dewey and Pachter, 2006; Margulies and Birney, 2008; Paten *et al.*, 2008a). Even after the advent of non-collinear aligners, though, synteny mappers have been advanced steadily (Hachiya *et al.*, 2009; Pham and Pevzner, 2010; Drillon *et al.*, 2014; Tang *et al.*, 2015), maybe partly because gene-based analyses of synteny (and collinearity) remain handy and important (both by themselves and as preliminaries of functional and evolutionary analyses). Now, some state-of-the-art synteny mappers can handle quite a large number of genomes (e.g., Rödelberger and Dieterich, 2010; Proost *et al.*, 2012; Wang *et al.*, 2012; Minkin *et al.*, 2013).

In Supplementary Table S1, we have listed the approaches (such as methods, programs, and algorithms) discussed above and in the previous “Topic” section, as well as some additional ones that also caught our attention. In the table, the approaches are classified from several points of view, and each of them is also accompanied by some comments (on algorithms, scope, output formats, etc.) and notes, if at all. (The relevant references are listed in a separate sheet).

Supplementary data associated with this article can be found in the online version at <https://doi.org/10.1016/B978-0-12-809633-8.20237-9>.

Here are some caveats. It should be kept in mind that Supplementary Table S1 is far from exhaustive. Information that somewhat complements the table can be obtained, e.g., from the following sources: regarding aligners, the references cited in Introduction of this article; regarding synteny mappers, table 1 of Rödelberger and Dieterich (2010), supplementary table S1 of Proost *et al.* (2012), table 3 of Wang *et al.* (2012), table 1 of Tang *et al.* (2015), and a review (Ghiurcuta and Moret, 2014); regarding browsers and viewers, Introduction of Nguyen *et al.* (2014), and a review (Wang *et al.*, 2013).

As far as we know, among the numerous (stand-alone) synteny mappers that have been developed thus far, only Mercator, Enredo, Shuffle-LAGAN and its successor, SuperMap (Dubchak *et al.*, 2009), and some methods based on BLAT (Kent, 2002) (e.g., Brudno *et al.*, 2004) were combined with collinear aligners to produce multiple genome alignments. Other synteny mappers have

been used mainly for the evolutionary analyses of genome architectures (including whole-genome duplications, segmental duplications, and tandem arrays of repeats) or genomic contexts of genes (including orthology, regulatory elements and gene families). It may be worth exploring the combinations of such unused synteny mappers and (especially collinear) multiple aligners.

Regarding visualization, many aligners (e.g., [Darling et al., 2010](#)) are internally equipped with, or packaged with, their own viewers, which may be sufficient for usual purposes.

Illustrative Examples

In genome alignment programs, multi-FASTA (MFA) format and multiple alignment format (MAF) files are frequently adopted as input and output, respectively (e.g., [Schwartz et al., 2003](#); [Blanchette et al., 2004](#); [Angiuoli and Salzberg, 2011](#); [Paten et al., 2011](#)). A MFA file contains multiple FASTA-formatted sequence data, each of which consists of a sequence identifier line followed by one or more sequence lines. The FASTA format is used in a wide range of sequence analysis programs, such as sequence alignment, motif discovery and gene finding. See the NCBI web site (see “Relevant Websites section”) for more details on the FASTA format. The MAF, which was also adopted as the submission format in Alignathon (see Section “Alignathon” below) ([Earl et al., 2014](#)), stores a series of MSAs which covers some (or most) parts of entire genomes. Each MSA is expected to consist of genome sequences which are descended from a common ancestral sequence. Depending on what types of genome rearrangements occurred during the course of evolution, MSAs in MAF files show characteristic composition. For instance, duplications lead to MSAs which include more than one sequence from a single species, while deletions lead to MSAs which do not include sequences from a specific species. Moreover, inversions lead to MSAs which include sequences from the reverse strand. See the UCSC Genome Browser web site (see “Relevant Websites section”) for more details on the MAF.

A traditional way to view (the large-scale structure of) a (typically non-collinear) genome alignment is to draw dot-plots ([Gibbs and McIntyre, 1970](#)). A dot-plot graphically visualizes similarities between two nucleotide or amino acid sequences. Pairs of similar regions, which are identified by sequence alignments, are represented as diagonal segments in a dot-plot. [Fig. 1](#) shows the dot-plot of a pairwise alignment between simulated mouse and rat genomes. To create the figure, we downloaded a MAF file, which stores the “correct” alignment of artificial mammalian genomes, from the Alignathon web site (see “Relevant Websites section”) and extracted the pairwise alignment between simulated mouse and rat genomes. Then, we drew its dot-plot by using *maffTools* ([Earl et al., 2014](#)), the HAL package ([Hickey et al., 2013](#)) and a few of our perl and R scripts (available on request to TY). The artificial mammalian genomes were generated by a forward-time simulation of genome evolution. *MafTools* is a toolkit for MAF file (pre)processing, and the HAL package is a toolkit for comparative genome analyses.

A dot-plot is useful to visualize the overall picture of a pairwise genome alignment and infer what types of genome rearrangements occurred during the course of evolution. [Fig. 1](#) also shows typical patterns of diagonal segments in dot-plots and [Fig. 2](#) shows the corresponding parsimonious genome rearrangements. (And [Fig. 3](#) shows some typical patterns that are obscure in [Fig. 1](#).) The more complicated patterns can be understood as combinations of the typical patterns.

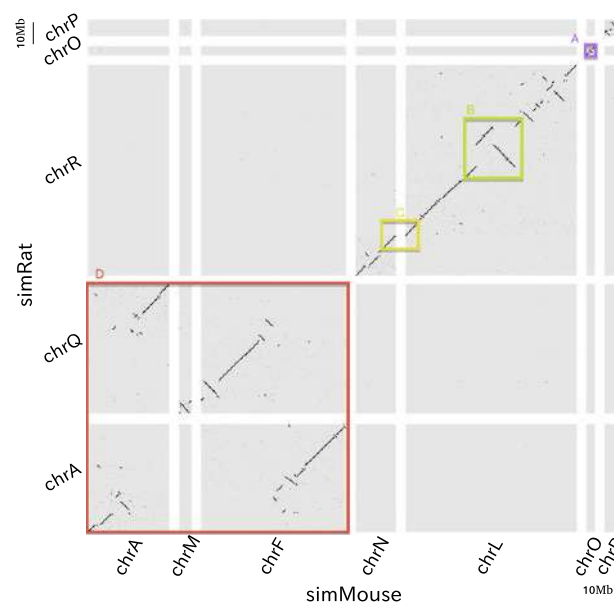


Fig. 1 Dot-plot of pairwise alignment between genomes of simulated mouse (simMouse) and rat (simRat). The colored rectangles enclose typical patterns indicative of genome rearrangements. See [Fig. 2](#) for their interpretations. This dot-plot is based on a simulated alignment used as input for Alignathon. Adapted from Earl, D., Nguyen, N., Hickey, G., et al., 2014. Alignathon: A competitive assessment of whole-genome alignment methods. *Genome Res.* 24, 2077–2089.

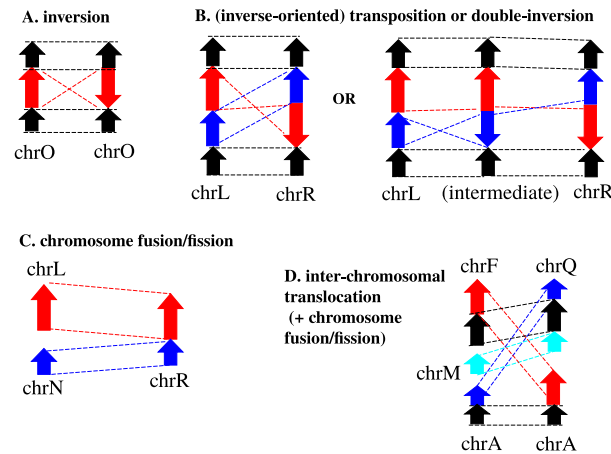


Fig. 2 Interpretation of diagnostic patterns in dot-plot (Fig. 1). Panels A-D in this figure correspond to the colored rectangles labeled A-D, respectively, in Fig. 1. In panel D, we ignored small-scale rearrangements such as inversions, in order to focus only on the inter-chromosomal translocation and the chromosome fusion/fission. In each panel, an arrow represents a chromosomal region. The arrows are oriented according to the genome on the left, which corresponds to the horizontal axis of the dot-plot (Fig. 1). A black arrow indicates that the region remain unarranged. A colored arrow indicates that the region was rearranged. A dashed line indicates the homology of the region ends that it ties.

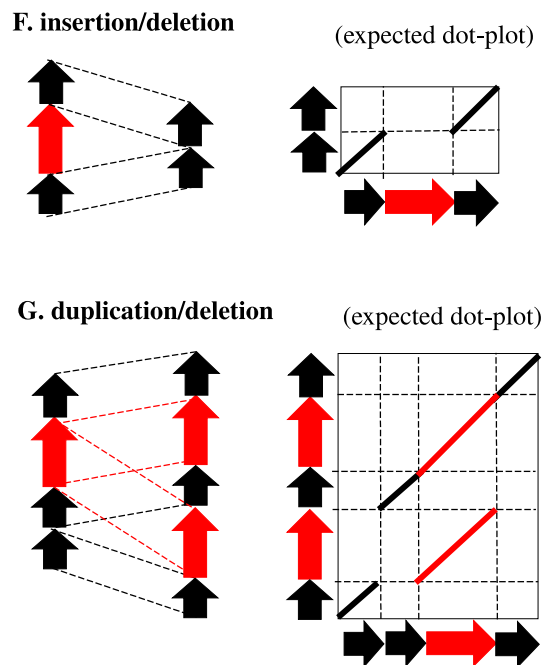


Fig. 3 Other diagnostic patterns common in dot-plots. These patterns are generally common in dot-plots, but are missing or obscure in Fig. 1.

Results and Discussion: Evaluation of Alignment Methods

It is utterly crucial to evaluate the accuracy of alignments output by each aligner, because an aligner would be useless unless its resulting alignments are reasonably accurate at least, no matter how fast it can align genome sequences. It is desirable that a third party should evaluate a number of aligners in a fair setting. However, an overwhelming majority of assessments thus far have been conducted by the groups that developed the aligners, thus the results have tended to favor the developers' own aligners. Nevertheless there have been a few studies where independent groups evaluated some aligners (e.g., Margulies *et al.*, 2007; Chen and Tompa, 2010), and a competitive assessment called the "Alignathon", in which a number of aligner-developing groups participated under a considerably fair setting (Earl *et al.*, 2014). In this section, we discuss these evaluations, after briefly discussing major evaluation methods.

Major Evaluation Methods

It is notoriously difficult to evaluate the accuracy of a genome alignment, because we do not know the “correct” alignment that resulted from the true evolution of the genome sequences, and that can be used as the “gold standard” to be compared against alignments reconstructed by aligners. Therefore, it is inevitable to settle for some surrogate methods for evaluating the alignment accuracy. According to [Earl et al. \(2014\)](#), such evaluation methods conducted thus far can be classified into four broad categories: (i) those using simulation, (ii) those using expert information, (iii) those using direct statistical assessment, and (iv) those that assess how well an alignment functions for a downstream analysis. Each of these methods has its own pros and cons. Here, we briefly summarize the discussions on them by [Earl et al. \(2014\)](#). (See [Iantorno et al., 2014](#) for a more comprehensive review on the topic).

In category (i), researchers create some “correct” alignments via simulations of genome evolution, and compare them with alignments reconstructed by aligners. A merit of this category is that we know the “correct” alignment that can be compared to the reconstructed ones. Its major demerit is that it is not certain whether the simulation adequately reflects the evolutionary forces operating on the true genomes.

In category (ii), researchers evaluate reconstructed alignments using some expert biological information, such as annotations of functional elements, structural alignments, and expertly curated alignments (e.g., [Blackshields et al., 2006](#); [Wilm et al., 2006](#); [Kemena et al., 2013](#)). A merit of this category is that the methods provide objective evolutionary contexts derived from the real genomes, giving us some relief. Its demerit is that each method may address only a small fraction of the alignment and that each method may itself rely on other forms of inference, thus making it uncertain whether the assessment is indeed correct or not.

In category (iii), evaluations are made using statistical measures that indicate the reliability (or uncertainty) of each portion of reconstructed alignments (e.g., [Prakash and Tompa, 2007](#); [Landan and Graur, 2008](#); [Penn et al., 2010](#); [Kim and Ma, 2011](#); [Chang et al., 2014](#)). A merit of this category is that we can assess the complete alignments of real sequences. Its demerit is that, unless carefully calibrated using gold standard alignments, they are only a proxy to a true assessment of accuracy.

In category (iv), aligners are assessed, typically by biologists, according to intuition or downstream analyses. Because of their ad-hoc nature, these assessments are difficult if not impossible to generalize from. For whole-genome alignments (WGAs), their effects were assessed, e.g., on de novo ncRNA predictions ([Gorodkin et al., 2010](#)) and on the prediction of conserved elements ([Margulies et al., 2007](#)).

Before Alignathon

Before the Alignathon ([Earl et al., 2014](#)) took place, there have been relatively few independent or community organized assessments of WGA pipelines. Notable among such evaluations are the assessment as a part of the ENCODE pilot project ([Margulies et al., 2007](#)) and the evaluation by [Chen and Tompa \(2010\)](#).

As a part of the ENCODE pilot project ([Birney et al., 2007](#)), [Margulies et al. \(2007\)](#) aligned sequences orthologous to the ENCODE-regions obtained from 23 mammalian species (including human and two non-eutherian mammals). They used four pipelines (each represented by its major components): TBA/BLASTZ (abbreviated as TBA) ([Blanchette et al., 2004](#); [Schwartz et al., 2003](#)), MLAGAN/Shuffle-LAGAN (abbr. as MLA-GAN) ([Brudno et al., 2003b,c](#); [Dubchak et al., 2009](#)), MAVID/Mercator (abbr. as MAVID) ([Bray and Pachter, 2004](#); [Dewey, 2007](#)), and Pecan ([Paten et al., 2008a, 2009](#)). (NOTE: The Pecan pipeline actually used Shuffle-LAGAN as a pre-processor, because Pecan is a collinear aligner.) Their comparisons of the four pipelines revealed large-scale consistency but substantial differences in terms of small genomic rearrangements, sensitivity (sequence coverage) and specificity (alignment accuracy).

In terms of the above classification, their alignment assessments belong to category (ii). Because their alignments are of real genomic sequences, they cannot be compared with the true alignments, which are inherently unknown. Therefore [Margulies et al. \(2007\)](#) used some surrogates for sensitivity and specificity. For sensitivity, they used two surrogates. One is the coverage of annotated protein-coding sequences in each alignment. And the other is the coverage of ancestral repeats (ARs) in each alignment. Overall, these coverage measures varied considerably among different alignments: MAVID showed the smallest coverage, MLAGAN showed the largest coverage for protein-coding sequences, and Pecan showed the largest coverage for ARs. For specificity as well, they used two surrogates. One was the “*Alu* exclusion”, which is how well each pipeline can avoid the false-positive alignments of human *Alu* elements to any non-primate mammalian sequence. (It is well known that *Alu* elements are specific to primates.) And the other was the substitution periodicity in coding regions, which takes advantage of the fact that the third codon positions are enriched in synonymous sites and thus prone to substitutions. TBA was the most specific in terms of both measures, and Pecan almost tied with TBA in terms of the substitution periodicity. MLAGAN was the least specific in terms of the *Alu* exclusion, and MAVID was the least specific in terms of the substitution periodicity.

[Chen and Tompa \(2010\)](#) also evaluated the alignments of the ENCODE-regions generated by the aforementioned four pipelines. One difference from the study of [Margulies et al. \(2007\)](#) is that [Chen and Tompa \(2010\)](#) included non-mammalian vertebrates as well. (Like [Margulies et al., 2007](#), they analyzed the pairwise alignments of human and each non-human species extracted from the reconstructed multiple genome alignments.) Another difference is that the evaluation of [Chen and Tompa \(2010\)](#) was more comprehensive, involving all sites of the aligned sequences. The evaluation results were also broken down into four location categories, namely, coding, untranslated (UTR), intronic, and intergenic. In the comparison of the alignments generated by TBA, MLAGAN, and MAVID, the degree of agreement substantially decreased from coding to non-coding regions, and as the evolutionary distance between species increases. Their alignment evaluation belongs to category (iii). As a surrogate for the

sensitivity, they examined the coverage of each alignment, and to evaluate the alignment accuracy, they used a statistical measure computed via StatSigMa-w (Prakash and Tompa, 2007). Regarding the coverage, the results were consistent with those of Margulies *et al.* (2007), that is, MLAGAN showed the highest coverage and MAVID showed the lowest coverage. For every pipeline, the coverage decreased as the divergence between species increases, especially for non-coding regions. Regarding the alignment accuracy, Pecan was the most accurate, and MLAGAN was the least accurate. TBA was nearly as accurate as Pecan for eutherian mammals, but was substantially less accurate than Pecan for more distant species. For every pipeline, the accuracy suddenly dropped from eutherian mammals to more divergent species, and from coding to non-coding regions. These results indicated that reconstructing whole-genome multiple sequence alignments remains a challenge, especially for non-coding regions and distant species (like non-eutherian mammals and non-mammalian vertebrates). One caveat given by Chen and Tompa (2010) is that their accuracy measure based on StatSigMa-w may somewhat overestimate the true accuracy; thus, the reconstructed alignments should have more misalignments than their accuracy measure indicates. Chen and Tompa (2010) attributed this to StatSigMa-w's conservative calls of suspicious regions in the alignment.

Alignathon

The Alignathon (Earl *et al.*, 2014) is a competitive evaluation of whole-genome alignment tools in which teams submitted their alignments and then assessments were performed collectively after all the submissions were received. As the input, they used three data sets. Two were simulated DNA data: one based on the phylogeny of four primates (human, chimpanzee, gorilla, and orangutan), and the other based on the phylogeny of five eutherian mammals (human, mouse, rat, dog, and cow). And the remaining one was the real DNA sequence data, comprised of 20 fly genomes. In total, the Alignathon received 35 submissions, which were generated by 12 different alignment pipelines. The 12 pipelines they used were: three pipelines used by genome browsers (VISTA-LAGAN for the VISTA Browser (Frazer *et al.*, 2004; Dubchak *et al.*, 2009), MULTIZ for the UCSC Genome Browser (Miller *et al.*, 2007; Meyer *et al.*, 2013), and Pecan and EPO for the Ensembl Browser (Paten *et al.*, 2008a, 2009; Flicek *et al.*, 2013); seven standalone WGA tools (progressiveMauve (Darling *et al.*, 2010), TBA (Blanchette *et al.*, 2004), Cactus (Paten *et al.*, 2011), Mugsy (Angiuoli and Salzberg, 2011), Robusta (Notredame, 2012), which combines results from multiple stand-alone tools, PSAR-Align (Kim and Ma, 2014), which was used to re-align MULTIZ-based alignments and the pipelines of AutoMZ and GenomeMatch teams. (Specific settings of the pipelines are described in the Supplemental material of Earl *et al.*, 2014).

For each reconstructed alignment of the simulated data, they computed the “precision” (the proportion of aligned pairs of nucleotides in the reconstructed alignment that are also true) and the “recall” (the proportion of aligned pairs of nucleotides in the true alignment that are also reconstructed). And they also defined the F-score as the harmonic mean of precision and recall. This evaluation belongs to category (i) in Section “Major Evaluation Methods”. They found that many of the submissions were able to align primate data set with both relatively high recall and precision, consistently across annotation types. For the mammalian simulations, they found a much wider variation of results, both between alignments and between different annotation classes. Especially, recall varied much more widely than precision, which was relatively high for most aligners. Among submissions, Cactus showed the highest F-score. Generally, submissions performed best in genic regions and most poorly in repetitive regions. And precision and recall values got lower as the distance between species increased.

For each reconstructed alignment of the real genomic data, the aforementioned recall or precision cannot be computed. Thus, they resorted to a statistical evaluation (belonging to category (iii) in Section “Major Evaluation Methods”). As a proxy to the precision, they computed the *PSAR-precision* based on the output of PSAR statistical alignment tool (Kim and Ma, 2011). As a proxy to the recall, they used the *coverage*. Then, as a proxy to the F-Score, they also defined the *pseudo F-Score* as the harmonic mean of PSAR-precision and coverage. Before applying these statistical measures to alignments of the real fly genomes, they examined their consistency with the measures for simulations, after calculating both for the simulated alignments. In the simulated mammals, they found a very good, linear correlation between recall and coverage ($r^2=0.984$), but no linear correlation between PSAR-precision and precision. Regarding the F-Score and the pseudo F-Score, they linearly correlated strongly ($r^2=0.975$), likely because recall varied much more widely among the alignments than precision. Then, using the statistical measures, they evaluated the reconstructed alignments of the real 20 fly genomes. In general, they saw good concordance between the fly and simulated results.

These results are more or less consistent with those of the aforementioned two evaluations (Margulies *et al.*, 2007; Chen and Tompa, 2010). In addition, the Alignathon produced some interesting results. For example, direct comparisons between the submitted alignments suggested that sharing the same synteny mapper influenced the results more strongly than sharing the same (collinear) synteny block aligner. Moreover, they found that no pipelines except Cactus can capture significantly nonzero duplications into the reconstructed alignment. In their conclusions, they cautioned against overinterpretation of the results because of the limitations of their assessment methods, and they encouraged more independent lines of assessment: more simulations, more statistical assessments, and assessments at different scales, etc.

Future Directions

In the past two decades, vigorous efforts and dedication of many researchers have greatly improved genome alignment methods in terms of their speed, accuracy, and scope (such as the number of input sequences, evolutionary distance, the range of acceptable mutations, etc.). However, as the field of comparative genomics gets more advanced and more and more genome sequences become available, genome aligners will need to become faster, more accurate, more robust and more flexible. Indeed,

researchers eagerly attempt to improve the aligners' performance even today (e.g., Frith and Kawaguchi, 2015; Uricaru *et al.*, 2015; Mai *et al.*, 2017; Suarez *et al.*, 2017). Here, we discuss a few representatives among possible future directions that the study of genome alignment should head for.

Pursuit of Further Space- and Time-Efficiency

Genome alignment methods have become substantially efficient in computational time and space (memory) requirement (e.g., Paten *et al.*, 2009; Nakato and Gotoh, 2010; Angiuoli and Salzberg, 2011). However, as the number of available genome sequences has still been growing exponentially (e.g., Margulies and Birney, 2008; Dewey, 2012; Earl *et al.*, 2014; Thompson, 2016a), the demand also keeps growing for a method that can align a large number of genomes in a reasonable amount of time (e.g., Dewey, 2012).

Regarding the multiple alignment of small to medium size sequences, some new heuristic methods have recently been proposed that enable the alignment of thousands to even millions of sequences (e.g., Katoh and Toh, 2007; Sievers *et al.*, 2011; Mirarab *et al.*, 2015; Nguyen *et al.*, 2015). (For a review of such aligners, see, e.g., Thompson, 2016b and Warnow (2017)). By combining in some way these heuristics with the heuristics that have been developed for genome alignment thus far (such as "anchoring" (e.g., Delcher *et al.*, 1999; Höhl *et al.*, 2002; Brudno *et al.*, 2003a), also known as the "divide-and-conquer" strategy for a long alignment (e.g., Kryukov and Saitou, 2010)), it may be possible to develop aligners that can align, e.g., hundreds to thousands of genomes in a reasonable amount of time. A framework using the meta-alignment methods, Crumble and Prune (Roskin *et al.*, 2011), would probably provide a good starting point for such efforts. And it would also be important to extend the framework to non-collinear aligners.

Pursuit of Further Alignment Accuracy

Although the accuracy of genome aligners have improved considerably (e.g., Chen and Tompa, 2010; Earl *et al.*, 2014), there is still big room for improvement. The problem of alignment accuracy may be broadly divided into two subproblems: (a) it seems that the specificity (or precision) have reached a plateau that is still considerably below (near) perfection; (b) the sensitivity (or recall) is far below perfection even with the current best aligner (e.g., Cactus). The subproblem (a) may be overcome, e.g., by (1) incorporating information other than the primary structures (i.e., the strings of bases) of DNA sequences (reviewed, e.g., in Kemena and Notredame, 2009), and by (2) taking account of the intrinsic stochasticity of genome evolution more seriously. The subproblem (b) may be addressed by (3) more accurate detection of genome rearrangements and duplications. And both subproblems may be partly solved by (4) properly dealing with errors in sequencing and assembly as well as false-positive matches. Let us discuss them in more details.

- (1) One possible reason for the imperfect specificity of current aligners is that the scoring schemes for them (including the quite sophisticated probabilistic scoring (e.g., Paten *et al.*, 2008b; Bradley *et al.*, 2009; Paten *et al.*, 2009)), are mostly based on simple "space-homogeneous" models, which provide homogeneous mutation rates along the sequence. In reality, it is well-known that functionally important regions evolve more slowly than neutral regions, and that evolution tend to conserve features related to functions (such as the secondary and tertiary structures of proteins and RNA genes) than the primary structures. Therefore, if these functional features are known prior to the alignment (via some experiments), such information could help reconstruct more accurate genome alignments (e.g., Kemena and Notredame, 2009). One implementation of this idea is via template-based aligners, which align sequences guided by templates that represent such functional information (e.g., Pei *et al.*, 2008; Wilm *et al.*, 2008; Katoh and Toh, 2008).

A possible challenge would be how to make template-based aligners scale to genome alignments, because they tend to be slower than simple aligners. Besides, this method would work only for functional regions with experimental information, which generally account for only a small fraction of genome sequences.

- (2) Another possible reason for the imperfect specificity is that the evolutionary processes are inherently stochastic; in consequence, a true alignment resulting from an evolutionary process is very frequently *not* optimal even under the ideal objective score function, i.e., (the logarithm of) the probability that natural evolutionary processes give rise to the alignment. To support this idea, it was shown that taking account of the stochasticity of evolution substantially improves the accuracy and consistency of the pairwise sequence comparisons (e.g., Lunter *et al.*, 2008; Cartwright, 2009). Moreover, a simulation study demonstrated that, in a (near) majority of errors in multiple sequence alignments, the true alignments indeed do *not* optimize the alignment probabilities calculated with the particular evolutionary model used for the simulations. These results strongly suggest that, in order to further improve the alignment accuracy, it is essential to present the stochastic nature of evolutionary processes as well, rather than merely attempting to reconstruct a single optimal alignment.

Attempts to cast the sequence alignment problems into a statistical framework with some probabilistic modelling of gaps (often referred to as "statistical alignment" (Hein *et al.*, 2000)) started with two groundbreaking studies (Bishop and Thompson, 1986; Thorne *et al.*, 1991). Then, the efforts extended to various problems (including multiple sequence alignments) using Hidden Markov Models (HMMs) (including transducers) (e.g., Holmes and Bruno, 2001; Holmes, 2003; Bradley and Holmes, 2007; Miklós *et al.*, 2009; Herman *et al.*, 2015; Rivas and Eddy, 2015). Currently, HMMs are the dominating models in the study of statistical alignment, partly because they are convenient enough to enable various dynamic programming algorithms (for, e.g.,

finding an optimum alignment and calculating the posterior probabilities of matches) (Durbin *et al.*, 1998), and partly because they are flexible enough to accommodate site-specific mutation rates (e.g., Durbin *et al.*, 1998; Rivas and Eddy, 2015) and mixed geometric length distributions of insertions/deletions (indels) (e.g., Miklós *et al.*, 2004; Do *et al.*, 2005; Lunter *et al.*, 2008; Bradley *et al.*, 2009; Paten *et al.*, 2009), which can approximate reasonably (up to dozens of residues) the power-law indel length distributions that have been repeatedly observed in molecular evolutionary analyses (e.g., Gonnet *et al.*, 1992; Benner *et al.*, 1993; Gu and Li, 1995; Zhang and Gerstein, 2003; Chang and Benner, 2004; Yamane *et al.*, 2006; Fan *et al.*, 2007). (For a recent review on these developments, see, e.g., Holmes, 2017).

Unfortunately, however, some recent studies revealed the limitations of alignment methods based on HMMs (e.g., Darling *et al.*, 2010; Rivas and Eddy, 2015).

These results imply that, in order to pursue near perfect accuracy, we will ultimately need a probabilistic aligner that is based on a genuine sequence evolutionary model with realistic instantaneous insertion/deletion rates, such as the substitution/insertion/deletion models (Miklós *et al.*, 2004). (NOTE: Roughly speaking, in a genuine sequence evolutionary model, the probability of an evolutionary process of a sequence is calculated as a multiplicative accumulation of infinitesimal-time transition probabilities of the states of an entire sequence over a given time interval). However, there still remain some technical challenges, such as the development of adequately fast algorithms, to achieve this crucial goal of developing a probabilistic aligner based on a genuine evolutionary model of a sequence.

- (3) In most of the current genome alignment methods, detecting genome rearrangements and duplications is the job of a separate synteny mapping program or the synteny mapping component of a non-collinear aligner. These methods have been based on heuristics, including the graph construction and modification procedures, and a number of involved parameters had to be adjusted via an extensive trial and error. Therefore, more principled methods based on probabilistic or, ideally, evolutionary modelling of synteny mapping may improve the accuracy (both sensitivity and specificity) of the detection of rearrangements and duplications (as suggested, e.g., by Dewey, 2012; Ghiurcuta and Moret, 2014). (Although some maximum likelihood methods have already been proposed (e.g., Hachiya *et al.*, 2009), the underlying probabilistic models seem somewhat heuristic). Besides, because synteny blocks (or regions of homologous collinearity) are products of inherently stochastic evolutionary processes, the true synteny maps may not optimize even the true occurrence probability, as in the case of collinear alignments. Therefore, also in synteny mapping, it will become essential to take adequate account of inherent stochastic nature of evolution, for example by presenting multiple near-optimal solutions and their probabilities (as suggested, e.g., by Dewey, 2012).

Probably, the best way would be to construct a genuine stochastic evolutionary model of a genome that incorporate rearrangements and duplications in addition to substitutions, insertions, and deletions, and to establish a probabilistic framework for synteny mapping based on the model. Thus far, some evolutionary models of genome rearrangements (and duplications) seem to have been proposed (e.g., Tesler, 2002; Ma *et al.*, 2008; Lin and Moret, 2011; Da Silva *et al.*, 2012; Shao *et al.*, 2013; Ghiurcuta and Moret, 2014; Braga and Stoye, 2015). To the best of our knowledge, however, there have been no genuine stochastic evolutionary models describing genome rearrangements and duplications applicable to the genome alignment problem. If such genuine stochastic evolutionary models are developed in the future, they may be applied to the problem of “statistical whole-genome alignment”, including “statistical synteny mapping”. We do admit that lots of challenges lie ahead in order to achieve this formidable goal. Especially, in addition to some methodological breakthroughs to enable the efficient computation of the likelihoods of possible synteny mappings, it should become indispensable to determine feature-dependent rates of rearrangements and duplications via data analyses that are more extensive and more meticulous than the past ones (e.g., Kent *et al.*, 2003; Church *et al.*, 2009; Mills *et al.*, 2011).

- (4) Because errors in the sequencing and assembly processes are likely to cause serious errors in the final alignment, dealing with these errors prior to or during the alignment process could significantly improve the alignment accuracy (as discussed, e.g., in Pevzner and Tesler, 2003; Margulies and Birney, 2008). An almost equally important issue is the handling of false-positive matches (or seeds) that are erroneously chosen as anchors. Traditionally, chaining (or clustering) algorithms (e.g., Bourque *et al.*, 2004; Ohlebusch and Abouelhoda, 2005) have served to filter potentially false-positive matches, and more recently, some graph-based non-collinear aligners and synteny mappers seem to deal with these problems by “cleaning” or “smoothing” the graphs via some intuition-based heuristics (e.g., Paten *et al.*, 2008a; Pham and Pevzner, 2010). Theoretically, these problems could also be addressed in a more principled way, if we model these errors as some stochastic processes (e.g., Verzotto *et al.*, 2016). And the error-handling and non-collinear alignment (or synteny mapping) might be performed in a unified manner if we merge the stochastic model of the errors with the genuine stochastic evolutionary model of genomes, as in some population genetic studies.

Closing Remarks

Genome sequence alignment is a cornerstone of comparative genomic study. Unlike traditional alignment of e.g., protein coding sequences or RNA sequences, genome alignment has been extremely challenging because of the sheer sizes of genome sequences and of the fact that genome sequences have undergone rearrangements such as inversions, translocations, and duplications during their evolution, in addition to substitutions, insertions, and deletions that have been dealt with by traditional alignment methods. Over the past two decades, a number of methodological innovations have greatly advanced the genome alignment methods, to

cope with these challenges and to improve the accuracy of the resulting alignments. However, as the number of sequenced genomes has been growing exponentially, and as further accurate comparative genomic analyses have been getting required, the genome alignment methods must still keep improving, e.g., via new strategies discussed in the “Future Directions” section.

Acknowledgments

We appreciate the sincere efforts of reviewers, editors, and other dedicated members of Elsevier and its contractors. We are also grateful to Dr. Glenn Hickey at UC Santa Cruz for his technical support for the use of the HAL package.

Appendix A Supplementary Information

Supplementary data associated with this article can be found in the online version at <https://doi.org/10.1016/B978-0-12-809633-8.20237-9>.

See also: Algorithms for Strings and Sequences: Pairwise Alignment. Comparative and Evolutionary Genomics. Comparative Genomics Analysis. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome Informatics. Identification of Homologs. Integrative Analysis of Multi-Omics Data. Metagenomic Analysis and its Applications. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Phylogenetic Footprinting. Quantitative Immunology by Data Analysis Using Mathematical Models. Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?. Sequence Analysis. Sequence Composition. Whole Genome Sequencing Analysis

References

- Afgan, E., Baker, D., Van Den Beek, M., *et al.*, 2016. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.* 44, W3–W10.
- Altschul, S., Madden, T., Schäffer, A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.
- Angiuoli, S., Salzberg, S., 2011. Mugsy: Fast multiple alignment of closely related whole genomes. *Bioinformatics* 27, 334–342.
- Aniba, M., Poch, O., Thompson, J., 2010. Issues in bioinformatics benchmarking: The case study of multiple sequence alignment. *Nucleic Acids Res.* 38, 7353–7363.
- Bahassi, E.M., Stambrook, P., 2014. Next-generation sequencing technologies: Breaking the sound barrier of human genetics. *Mutagenesis* 29, 303–310.
- Batzoglou, S., Pachter, L., Mesirov, J., *et al.*, 2000. Human and mouse gene structure: Comparative analysis and application to exon prediction. *Genome Res.* 10, 950–958.
- Benner, S., Cohen, M., Gonnet, G., 1993. Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol.* 229, 1065–1082.
- Birney, E., Stamatoyannopoulos, J., Dutta, A., *et al.*, 2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 447, 799–816.
- Bishop, M., Thompson, E., 1986. Maximum likelihood alignment of DNA sequences. *J. Mol. Biol.* 190, 159–165.
- Blackshields, G., Wallace, I., Larkin, M., *et al.*, 2006. Analysis and comparison of benchmarks for multiple sequence alignment. *In Silico Biol.* 6, 321–339.
- Blanchette, M., Kent, W., Riemer, C., *et al.*, 2004. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.* 14, 708–715.
- Bleidorn, C., 2017. *Phylogenomics: An introduction*. Cham: Springer, (Chapter 6).
- Bourque, G., Pevzner, P., Tesler, G., 2004. Reconstructing the genomic architecture of ancestral mammals: Lessons from human, mouse, and rat genomes. *Genome Res.* 14, 507–516.
- Bradley, R., Roberts, A., Smoot, M., *et al.*, 2009. Fast statistical alignment. *PLOS Comput Biol.* 5, e1000392.
- Bradley, R., Holmes, I., 2007. Transducers: An emerging probabilistic framework for modeling indels on trees. *Bioinformatics* 23, 3258–3262.
- Braga, M., Stoye, J., 2015. Sorting linear genomes with rearrangements and indels. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 12, 500–506.
- Bray, N., Dubchak, I., Pachter, L., 2003. AVID: A global alignment program. *Genome Res.* 13, 97–102.
- Bray, N., Pachter, L., 2004. MAVID: Constrained ancestral alignment of multiple sequences. *Genome Res.* 14, 693–699.
- Brudno, M., Chapman, M., Göttgens, B., *et al.*, 2003a. Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinform.* 4, 66.
- Brudno, M., Do, C., Cooper, G., *et al.*, 2003b. LAGAN and Multi-LAGAN: Efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res.* 13, 721–731.
- Brudno, M., Dubchak, I., 2005. Comparisons of long genomic sequences: Algorithms and applications. In: ALURU, S. (Ed.), *Handbook of Computational Molecular Biology*. Chapman and Hall/CRC.
- Brudno, M., Poliakov, A., Salamov, A., *et al.*, 2004. Automated whole-genome multiple alignment of rat, mouse, and human. *Genome Res.* 14, 685–692.
- Brudno, M., Malde, S., Poliakov, A., *et al.*, 2003c. Global alignment: Finding rearrangements during alignment. *Bioinformatics* 19, i54–i62.
- Cartwright, R., 2009. Problems and solutions for estimating indel rates and length distributions. *Mol. Biol. Evol.* 26, 473–480.
- Chang, J., Di Tommaso, P., Notredame, C., 2014. TCS: A new multiple sequence alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction. *Mol. Biol. Evol.* 31, 1625–1637.
- Chang, M., Benner, S., 2004. Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J. Mol. Biol.* 341, 617–631.
- Chen, X., Tompa, M., 2010. Comparative assessment of methods for aligning multiple genome sequences. *Nat. Biotechnol.* 28, 567–572.
- Church, D., Goodstadt, L., Hillier, L., *et al.*, 2009. Lineage-specific biology revealed by a finished genome assembly of the mouse. *PLOS Biol.* 7, e1000112.
- Clark, A., Eisen, M., Smith, D., *et al.*, 2007. Evolution of genes and genomes on the *Drosophila* phylogeny. *Nature* 450, 203–218.
- Cordero, F., Beccuti, M., Donatelli, S., *et al.*, 2012. Large disclosing the nature of computational tools for the analysis of next generation sequencing data. *Curr. Top. Med. Chem.* 12, 1320–1330.
- Couronne, O., Poliakov, A., Bray, N., *et al.*, 2003. Strategies and tools for whole-genome alignments. *Genome Res.* 13, 73–80.
- Darling, A., Mau, B., Blattner, F., *et al.*, 2004. Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *Genome Res.* 14, 1394–1403.
- Darling, A., Mau, B., Perna, N., 2010. progressiveMauve: Multiple genome alignment with gene gain, loss and rearrangement. *PLOS ONE* 5, e11147.

- Da Silva, P., Machado, R., Dantas, S., *et al.*, 2012. Restricted DCJ-indel model: Sorting linear genomes with DCJ and indels. *BMC Bioinform.* 13 (Suppl. 19), S14.
- Delcher, A., Kasif, S., Fleischmann, R., *et al.*, 1999. Alignment of whole genomes. *Nucleic Acids Res.* 27, 2369–2376.
- Delcher, A., Phillippy, A., Carlton, J., *et al.*, 2002. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Res.* 30, 2478–2483.
- Dewey, C., 2012. Whole-genome alignment. In: Anisimova, M. (Ed.), *Evolutionary Genomics. Methods in Molecular Biology (Methods and Protocols)*, vol. 855. Totowa, NJ: Humana Press, pp. 237–257.
- Dewey, C., 2007. Aligning multiple whole genomes with Mercator and MAVID. *Methods Mol Biol.* 395, 221–236.
- Dewey, C., Pachter, L., 2006. Evolution at the nucleotide level: The problem of multiple whole-genome alignment. *Hum. Mol. Genet.* 15, R51–R56.
- Do, C., Mahabhashyam, M., Brudno, M., *et al.*, 2005. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.* 15, 330–340.
- Drillon, G., Carbone, A., Fischer, G., 2014. SynChro: A fast and easy tool to reconstruct and visualize synteny blocks along eukaryotic chromosomes. *PLOS ONE* 9, e92621.
- Dubchak, I., Poliakov, A., Kislyuk, A., *et al.*, 2009. Multiple whole-genome alignments without a reference organism. *Genome Res.* 19, 682–689.
- Dubchak, I., Pachter, L., 2002. The computational challenges of applying comparative-based computational methods to whole genomes. *Brief. Bioinform.* 3, 18–22.
- Durbin, R., Eddy, S., Krogh, A., *et al.*, 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, UK: Cambridge University Press.
- Earl, D., Nguyen, N., Hickey, G., *et al.*, 2014. Alignathon: A competitive assessment of whole-genome alignment methods. *Genome Res.* 24, 2077–2089.
- Edgar, R., 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* 32, 1792–1797.
- Elinski, L., Burhans, R., Riemer, C., *et al.*, 2010. MultiPipMaker: A comparative alignment server for multiple DNA sequences. *Curr. Protoc. Bioinform.* 30, 10.4.1–10.4.14. <https://doi.org/10.1002/0471250953.bi1004s30>.
- Engström, P., Ho Sui, S., Drivenes, O., *et al.*, 2007. Genomic regulatory blocks underlie extensive microsynteny conservation in insects. *Genome Res.* 17, 1898–1908.
- Fan, Y., Wang, W., Ma, G., *et al.*, 2007. Patterns of insertion and deletion in mammalian genomes. *Curr. Genom.* 8, 370–378.
- Feng, D., Doolittle, R., 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* 25, 351–360.
- Ferrier, D., Holland, P., 2001. Ancient origin of the Hox cluster. *Nat. Rev. Genet.* 2, 33–38.
- Fitch, W., 1970. Distinguishing homologous from analogous proteins. *Syst. Zool.* 19, 99–113.
- Fitch, W., 2000. Homology a personal view on some of the problems. *Trends Genet.* 16, 227–231.
- Flicek, P., Ahmed, I., Amode, M.R., *et al.*, 2013. Ensembl 2013. *Nucleic Acids Res.* 41, D48–D55.
- Frazer, K., Elinski, L., Chrch, D., *et al.*, 2003. Cross-species sequence comparisons: A review of methods and available resources. *Genome Res.* 13, 1–12.
- Frazer, K., Pachter, L., Poliakov, A., *et al.*, 2004. VISTA: Computational tools for comparative genomics. *Nucleic Acids Res.* 32, W273–W279.
- Frith, M., Kawaguchi, R., 2015. Split-alignment of genomes finds orthologies more accurately. *Genome Biol.* 16, 106. <https://doi.org/10.1186/s13059-015-0670-9>.
- Gascuel, O. (Ed.), 2005. *Mathematics of Evolution and Phylogeny*. New York, NY: Oxford University Press.
- Ghiurcuta, C., Moret, B., 2014. Evaluating synteny for improved comparative studies. *Bioinformatics* 30, i9–i18.
- Gibbs, R., McIntyre, G., 1970. The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur. J. Biochem.* 16, 1–11. <https://doi.org/10.1111/j.1432-1033.1970.tb01046.x>.
- Gibbs, R., Weinstock, G., Metzker, M., *et al.*, 2004. Genome sequence of the brown Norway rat yields insights into mammalian evolution. *Nature* 428, 493–521. <https://doi.org/10.1038/nature02426>.
- Gonnet, G., Cohen, M., Benner, S., 1992. Exhaustive matching of the entire protein sequence database. *Science* 256, 1443–1445.
- Gorodkin, J., Hofacker, I., Torarinsson, E., *et al.*, 2010. De novo prediction of structured RNAs from genomic sequences. *Trends Biotechnol.* 28, 9–19.
- Gotoh, O., 1982. An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 705–708.
- Graur, D., Li, W., 2000. *Fundamentals of Molecular Evolution*, second ed. Sunderland, MA: Sinauer Associates.
- Gray, G., Fitch, W., 1983. Evolution of antibiotic resistance genes: The DNA sequence of a kanamycin resistance gene from *Staphylococcus aureus*. *Mol. Biol. Evol.* 1, 57–66.
- Gregory, S., Sekhon, M., Schein, J., *et al.*, 2002. A physical map of the mouse genome. *Nature* 418, 743–750.
- Gusfield, D., 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge, UK: Cambridge University Press.
- Gu, X., Li, W., 1995. The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment. *J. Mol. Evol.* 40, 464–473.
- Gu, W., Zhang, F., Lupski, J., 2008. Mechanisms for human genomic rearrangements. *Pathogenetics* 1, 4.
- Hachiya, T., Osana, Y., Pependorf, K., *et al.*, 2009. Accurate identification of orthologous segments among multiple genomes. *Bioinformatics* 25, 853–860.
- Harris, R., 2007. *Improved Pairwise Alignment of Genomic DNA*. The Pennsylvania State University. (PhD thesis).
- Hein, J., Wiuf, C., Knudsen, B., *et al.*, 2000. Statistical alignment: Computational properties, homology testing and goodness-of-fit. *J. Mol. Biol.* 302, 265–279.
- Herman, J., Novák, A., Lyngsø, R., *et al.*, 2015. Efficient representation of uncertainty in multiple sequence alignments using direct acyclic graphs. *BMC Bioinform.* 16, 108. <https://doi.org/10.1186/s12859-015-0516-1>.
- Herrero, J., Muffato, M., Beal, K., *et al.*, 2016. Ensembl comparative genomics resources. *Database (Oxford)* 2016, bav096.
- Hickey, G., Paten, B., Earl, D., *et al.*, 2013. HAL: A hierarchical format for storing and analyzing multiple genome alignments. *Bioinformatics* 29, 1341–1342.
- Hirschberg, D., 1975. A linear space algorithm for computing maximal common subsequences. *Commun. ACM* 18, 341–343.
- Hogeweg, P., Hesper, B., 1984. The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *J. Mol. Evol.* 20, 175–186.
- Höhl, M., Kurtz, S., Ohlebusch, E., 2002. Efficient multiple genome alignment. *Bioinformatics* 18, S312–S320.
- Holmes, I., 2003. Using guide trees to construct multiple-sequence evolutionary HMMs. *Bioinformatics* 19, i147–i157.
- Holmes, I., 2017. Solving the master equation for indels. *BMC Bioinform.* 18, 255. <https://doi.org/10.1186/s12859-017-1665-1>.
- Holmes, I., Bruno, W., 2001. Evolutionary HMMs: A Bayesian approach to multiple alignment. *Bioinformatics* 17, 803–820.
- Hughes, A., Nei, M., 1988. Pattern of nucleotide substitution at major histocompatibility complex class I loci reveals overdominant selection. *Nature* 335, 167–170.
- Hupalo, D., Kern, A., 2013. Conservation and functional element discovery in 20 angiosperm plant genomes. *Mol. Biol. Evol.* 30, 1729–1744.
- Hurst, L., Pál, C., Lercher, M., 2004. The evolutionary dynamics of eukaryotic gene order. *Nat. Rev. Genet.* 5, 299–310.
- Iantorno, S., Gori, K., Goldman, N., *et al.*, 2014. Who watches the watchmen? an appraisal of benchmarks for multiple sequence alignment. In: Russell, D. (Ed.), *Multiple Sequence Alignment Methods. Methods in Molecular Biology (Methods and Protocols)*, vol. 1079. Totowa, NJ: Humana Press, pp. 59–73.
- Jayaraj, J., 2006. Computational methods for multiple genome alignment and synteny detection. Available at: <https://www.semanticscholar.org>.
- Katoh, K., Toh, H., 2007. PartTree: An algorithm to build an approximate tree from a large number of unaligned sequences. *Bioinformatics* 23, 372–374.
- Katoh, K., Toh, H., 2008. Recent developments in the MAFFT multiple sequence alignment program. *Brief Bioinform.* 9, 286–298.
- Kehr, B., Trappe, K., Holtgrewe, M., *et al.*, 2014. Genome alignment with graph data structures: A comparison. *BMC Bioinform.* 15, 99.
- Kellis, M., Patterson, N., Endrizzi, M., *et al.*, 2003. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* 423, 241–254.
- Kemena, C., Bussotti, G., Capriotti, E., *et al.*, 2013. Using tertiary structure for the computation of highly accurate multiple RNA alignments with the SARA-Coffee package. *Bioinformatics* 29, 1112–1119.
- Kemena, C., Notredame, C., 2009. Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics* 25, 2455–2465.
- Kent, W., 2002. BLAT-like alignment tool. *Genome Res.* 12, 656–664.
- Kent, W., Zahler, A., 2000. Conservation, regulation, synteny, and introns in a large-scale *C. briggsae*-*C. elegans* genomic alignment. *Genome Res.* 10, 1115–1125.
- Kent, W., Baertsch, R., Hinrichs, A., *et al.*, 2003. Evolution's cauldron: Duplication, deletion, and rearrangement in the mouse and human genomes. *Proc. Natl. Acad. Sci. USA* 100, 11484–11489.
- Kielbasa, S., Wan, R., Sato, K., *et al.*, 2011. Adaptive seeds tame genomic sequence comparison. *Genome Res.* 21, 487–493.

- Kikuta, H., Laplante, M., Navratilova, P., *et al.*, 2007. Genomic regulatory blocks encompass multiple neighbouring genes and maintain conserved synteny in vertebrates. *Genome Res.* 17, 545–555.
- Kimura, M., 1968. Evolutionary rate at the molecular level. *Nature* 217, 624–626.
- Kimura, M., 1983. *The Neutral Theory of Molecular Evolution*. Cambridge, UK: Cambridge University Press.
- Kimura, M., Ohta, T., 1974. On some principles governing molecular evolution. *Proc. Natl. Acad. Sci. USA* 71, 2848–2852.
- Kim, J., Ma, J., 2014. PSAR-align: Improving multiple sequence alignment using probabilistic sampling. *Bioinformatics* 30, 1010–1012.
- Kim, J., Ma, J., 2011. PSAR: Measuring multiple sequence alignment reliability by probabilistic sampling. *Nucleic Acids Res.* 39, 6359–6368.
- Kihara, H., 1947. *Ancestors of Common Wheat*. Tokyo: Sogensha, (in Japanese).
- Kryukov, K., Saitou, N., 2010. MISHIMA – A new method for high speed multiple alignment of nucleotide sequences of bacterial genome scale data. *BMC Bioinform.* 11, 142.
- Kumar, S., Filipiński, A., 2007. Multiple sequence alignment: In pursuit of homologous DNA positions. *Genome Res.* 17, 127–135.
- Kurtz, S., Phillippy, A., Delcher, A., *et al.*, 2004. Versatile and open software for comparing large genomes. *Genome Biol.* 5, R12.
- Landan, G., Graur, D., 2008. Local reliability measures from sets of co-optimal multiple sequence alignments. *Pac. Symp. Biocomput.* 13, 15–24.
- Lee, T., Kim, J., Robertson, J., *et al.*, 2017. Plant genome duplication database. In: Van Dijk, A. (Ed.), *Plant Genomic Databases. Methods in Molecular Biology*, vol. 1533. New York, NY: Humana Press, pp. 267–277.
- Lee, H., Tang, H., 2012. Next-generation sequencing technologies and fragment assembly algorithms. *Methods Mol. Biol.* 855, 155–174.
- Lin, Y., Moret, B., 2011. A new genomic evolutionary model for rearrangements, duplications, and losses that applies both eukaryotes and prokaryotes. *J. Comput. Biol.* 18, 1055–1064.
- Löytynoja, A., 2012. Alignment methods: Strategies, challenges, benchmarking, and comparative overview. In: Anisimova, M. (Ed.), *Evolutionary Genomics. Methods in Molecular Biology (Methods and Protocols)*, vol. 855. Totowa, NJ: Humana Press, pp. 203–235.
- Lunter, G., Rocco, A., Mimouni, N., *et al.*, 2008. Uncertainty in homology inferences: Assessing and improving genomic sequence alignment. *Genome Res.* 18, 298–309.
- Lupski, J., 2007. Genomic rearrangements and sporadic disease. *Nat. Genet.* 39, S43–S47.
- Lynch, M., 2007. *The Origins of Genome Architecture*. Sunderland, MA: Sinauer Associates.
- Mai, H., Lam, T., Ting, H., 2017. A simple and economical method for improving whole genome alignment. *BMC Genom.* 18 (Suppl 4), 362.
- Margulies, E., Birney, E., 2008. Approaches to comparative sequence analysis: Towards a functional view of vertebrate genomes. *Nat. Rev. Genet.* 9, 303–313.
- Margulies, E., Cooper, G., Asimenos, G., *et al.*, 2007. Analyses of deep mammalian sequence alignments and constraint predictions for 1% of the human genome. *Genome Res.* 17, 760–774.
- Mayor, C., Brudno, M., Schwartz, J., *et al.*, 2000. VISTA: Visualizing global DNA sequence alignments of arbitrary length. *Bioinformatics* 16, 1046–1047.
- Ma, J., Ratan, A., Raney, B., *et al.*, 2008. The infinite sites model of genome evolution. *Proc. Natl. Acad. Sci. USA* 105, 14254–14261.
- Ma, B., Tromp, J., Li, M., 2002. PatternHunter: Faster and more sensitive homology search. *Bioinformatics* 18, 440–445.
- Ma, J., Zhang, L., Suh, B., *et al.*, 2006. Reconstructing contiguous regions of an ancestral genome. *Genome Res.* 16, 1557–1565.
- Meyer, L.R., Zweig, A.S., Hinrichs, A.S., *et al.*, 2013. The UCSC Genome Browser database: Extensions and updates 2013. *Nucleic Acids Res.* 41, D64–D69.
- Mielczarek, M., Syzda, J., 2016. Review of alignment and SNP calling algorithms for next-generation sequencing data. *J. Appl. Genet.* 57, 71–79.
- Miklós, I., Lunter, G., Holmes, I., 2004. A “long indel” model for evolutionary sequence alignment. *Mol. Biol. Evol.* 21, 529–540.
- Miklós, I., Novák, A., Satiya, R., *et al.*, 2009. Stochastic models of sequence evolution including insertion-deletion events. *Stat. Methods Med. Res.* 18, 453–485.
- Miller, W., Rosenbloom, K., Hardison, R., *et al.*, 2007. 28-way vertebrate alignment and conservation track in the UCSC Genome Browser. *Genome Res.* 17, 1797–1808.
- Mills, R., Walter, K., Steward, C., *et al.*, 2011. Mapping copy number variation by population-scale genome sequencing. *Nature* 470, 59–65.
- Minkin, I., Patel, A., Kolmogorov, M., *et al.*, 2013. Sibelia: A scalable and comprehensive synteny block generation tool for closely related microbial genomes. In: DARLING, A., STOEY, J. (Eds.), *Algorithms in Bioinformatics. WABI 2013. Lecture Notes in Computer Science*, vol. 8126. Berlin, Heidelberg: Springer.
- Mirarab, S., Nguyen, N., Guo, S., *et al.*, 2015. PASTA: Ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *J. Comput. Biol.* 22, 377–386.
- Myers, E., Miller, W., 1988. Optimal alignments in linear space. *Comput. Appl. Biosci.* 4, 11–17.
- Nakato, R., Gotoh, O., 2010. Cgaln: Fast and space-efficient whole-genome alignment. *BMC Bioinform.* 11, 224.
- Needleman, S., Wunsch, C., 1970. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.* 48, 443–453.
- Nguyen, N., Hickey, G., Raney, B., *et al.*, 2014. Comparative assembly hubs: Web-accessible browsers for comparative genomics. *Bioinformatics* 30, 3293–3301.
- Nguyen, N., Mirarab, S., Kumar, K., *et al.*, 2015. Ultra-large alignments using phylogeny-aware profiles. *Genome Biol.* 16, 124. <https://doi.org/10.1186/s13059-015-0688-z>.
- Notredame C., 2012. *Notredame C. A meta-multiple genome alignment tool*. <http://www.tcoffee.org/Projects/robusta/>.
- Notredame, C., 2007. Recent evolutions of multiple sequence alignment algorithms. *PLOS Comput. Biol.* 3, e123.
- Ohlebusch, E., Abouelhoda, M., 2005. Chaining algorithms and applications in comparative genomics. In: ALURU, S. (Ed.), *Handbook of Computational Molecular Biology*. Chapman and Hall/CRC.
- Ovcharenko, I., Loots, G., Giardine, B., *et al.*, 2005. Mulan: Multiple-sequence local alignment and visualisation for studying function and evolution. *Genome Res.* 15, 184–194.
- Paten, B., Earl, D., Nguyen, N., *et al.*, 2011. Cactus: Algorithms for genome multiple sequence alignment. *Genome Res.* 21, 1512–1528.
- Paten, B., Herrero, J., Beal, K., *et al.*, 2009. Sequence progressive alignment, a framework for practical large-scale probabilistic consistency alignment. *Bioinformatics* 25, 295–301.
- Paten, B., Herrero, J., Beal, K., *et al.*, 2008a. Enredo and Pecan: Genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome Res.* 18, 1814–1828.
- Paten, B., Herrero, J., Fitzgerald, S., *et al.*, 2008b. Genome-wide nucleotide-level mammalian ancestor reconstruction. *Genome Res.* 18, 1829–1843.
- Pei, J., Kim, B., Grishin, N., 2008. PROMALS3D: A tool for multiple protein sequence and structure alignments. *Nucleic Acids Res.* 36, 2295–2300.
- Penn, O., Privman, E., Ashkenazy, H., *et al.*, 2010. GUIDANCE: A web server for assessing alignment confidence scores. *Nucleic Acids Res.* 38, W23–W28.
- Pevzner, P., Tesler, G., 2003. Genome rearrangements in mammalian evolution: Lessons from human and mouse genomes. *Genome Res.* 13, 37–45.
- Pham, S., Pevzner, P., 2010. DRIMM-Synteny: Decomposing genomes into evolutionary conserved segments. *Bioinformatics* 26, 2509–2516.
- Poliakov, A., Foong, J., Brudno, M., *et al.*, 2014. Genome VISTA – An integrated software package for whole-genome alignment and visualization. *Bioinformatics* 30, 2654–2655.
- Prabha, R., Singh, D., Gupta, S., *et al.*, 2014. Whole genome phylogeny of *Prochlorococcus marinus* group of cyanobacteria: Genome alignment and overlapping gene approach. *Interdiscip. Sci. Comput. Life Sci.* 6, 149–157.
- Prakash, A., Tompa, M., 2007. Measuring the accuracy of genome-size multiple alignments. *Genome Biol.* 8, R124.
- Proost, S., Fostier, J., De Witte, D., *et al.*, 2012. i-ADHoRe 3.0-fast and sensitive detection of genomic homology in extremely large data sets. *Nucleic Acids Res.* 40, e11.
- Raphael, B., Zhi, D., Tang, H., *et al.*, 2004. A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res.* 14, 2336–2346.
- Rausch, T., Emde, A., Weese, D., *et al.*, 2008. Segment-based multiple sequence alignment. *Bioinformatics* 24, i187–i192.
- Rivas, E., Eddy, S., 2015. Parametrizing sequence alignment with an explicit evolutionary model. *BMC Bioinformatics* 16, 406. <https://doi.org/10.1186/s12859-015-0832-5>.
- Rödelsperger, C., Dieterich, C., 2010. CYNTENATOR: Progressive gene order alignment of 17 vertebrate genomes. *PLOS ONE* 5, e8861.
- Roskin, K., Paten, B., Haussler, D., 2011. Meta-alignment with Crumble and Prune: Partitioning very large alignment problems for performance and parallelization. *BMC Bioinform.* 12, 144.
- Sahl, J., Matalka, M., Rasko, D., 2012. Phylomark, a tool to identify conserved phylogenetic markers from whole-genome alignments. *Appl. Environ. Microbiol.* 78, 4884–4892.
- Saitou, N., 2013. *Introduction to Evolutionary Genomics. Computational Biology*. vol. 17. London, UK: Springer.
- Sackton, T., Lazzaro, B., Schlenke, T., *et al.*, 2007. Dynamic evolution of the innate immune system in *Drosophila*. *Nat. Genet.* 39, 1461–1468.
- Schwartz, S., Kent, W., Smit, A., *et al.*, 2003. Human-mouse alignments with BLASTZ. *Genome Res.* 13, 103–107.
- Schwartz, S., Zhang, Z., Frazer, K., *et al.*, 2000. PipMaker – A web server for aligning two genomic DNA sequences. *Genome Res.* 10, 577–586.

- Shao, M., Lin, Y., Moret, B., 2013. Sorting genomes with rearrangements and segmental duplications through trajectory graphs. *BMC Bioinform.* 14 (Suppl. 15), S9.
- Sheng, X., Liang, D., Wen, J., *et al.*, 2011. Multiple genome alignments facilitate development of NPCL markers: A case study of tetrapod phylogeny focusing on the position of turtles. *Mol. Biol. Evol.* 28, 3237–3252. <https://doi.org/10.1093/molbev/msr148>.
- Sievers, F., Wilm, A., Dineen, D., *et al.*, 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* 7, 539. <https://doi.org/10.1038/msb.2011.75>.
- Smith, T., Waterman, M., 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197.
- Smith, H., Yun, S., 2017. Evaluating alignment and variant-calling software for mutation identification in *C. elegans* by whole-genome sequencing. *PLOS ONE* 12, e0174446.
- Schnable, J., 2015. Genome evolution in maize: From genomes back to genes. *Annu. Rev. Plant Biol.* 66, 329–343.
- Stark, A., Lin, M., Kheradpour, P., *et al.*, 2007. Discovery of functional elements in 12 *Drosophila* genomes using evolutionary signatures. *Nature* 450, 219–232.
- Suarez, H., Langer, B., Ladde, P., *et al.*, 2017. chainCleaner improves genome alignment specificity and sensitivity. *Bioinformatics* 33, 1596–1603.
- Swanson, W., Vacquier, V., 2002. The rapid evolution of reproductive proteins. *Nat. Rev. Genet.* 3, 137–144.
- Tang, H., Bomhoff, M., Briones, E., *et al.*, 2015. SynFind: Compiling syntenic regions across any set of genomes on demand. *Genome Biol. Evol.* 7, 3286–3298.
- Tesler, G., 2002. GRIMM: Genome rearrangements web server. *Bioinformatics* 18, 492–493.
- Thompson, J., 2016a. *Statistics for Bioinformatics: Methods for Multiple Sequence Alignment*, first ed. ISTE Press - Elsevier (Chapter 6).
- Thompson, J., 2016b. *Statistics for Bioinformatics: Methods for Multiple Sequence Alignment*, first ed. ISTE Press - Elsevier (Chapter 7).
- Thompson, J., Higgins, D., Gibson, T., 1994. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* 22, 4673–4680.
- Thorne, J., Kishino, H., Felsenstein, J., 1991. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Biol.* 33, 114–124.
- Tseng, H., Tompa, M., 2009. Algorithms for locating extremely conserved elements in multiple sequence alignments. *BMC Bioinform.* 10, 432.
- Tyner, C., Barber, G., Casper, J., *et al.*, 2017. The UCSC Genome Browser database: 2017 update. *Nucleic Acids Res.* 45, D626–D634.
- Uricaru, R., Michotey, C., Chiapello, H., *et al.*, 2015. YOC, a new strategy for pairwise alignment of collinear genomes. *BMC Bioinform.* 16, 111. <https://doi.org/10.1186/s12859-015-0530-3>.
- Verzotto, D., Teo, A., Hillmer, A., *et al.*, 2016. OPTIMA: Sensitive and accurate whole-genome alignment of error-prone genomic maps by combinatorial indexing and technology-agnostic statistical analysis. *Gigascience* 5, 2. <https://doi.org/10.1186/s13742-016-0110-0>.
- Wang, J., Kong, L., Gao, G., *et al.*, 2013. A brief introduction to web-based genome browsers. *Brief. Bioinform.* 14, 131–143.
- Wang, L., Jiang, T., 1994. On the complexity of multiple sequence alignment. *J. Comput. Biol.* 1, 337–348.
- Wang, X., Wang, J., Jin, D., *et al.*, 2015. Genome alignment spanning major Poaceae lineages reveals heterogeneous evolutionary rates and alters inferred dates for key evolutionary events. *Mol. Plant* 8, 885–898.
- Wang, Y., Tang, H., Debary, J., *et al.*, 2012. MCSanX: A toolkit for detection and evolutionary analysis of gene synteny and collinearity. *Nucleic Acids Res.* 40, e49.
- Warnow, T., 2017. *Computational Phylogenetics: An Introduction to Designing Methods for Phylogeny Estimation*. Cambridge University Press (Chapter 9).
- Watanabe, H., Fujiyama, A., Hattori, M., *et al.*, 2004. DNA sequence and comparative analysis of chimpanzee chromosome 22. *Nature* 429, 382–388.
- Waterston, R., Lindblad-Toh, K., Birney, E., *et al.*, 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* 420, 520–562.
- Wilm, A., Higgins, D., Notredame, C., 2008. R-Coffee: A method for multiple alignment of non-coding RNA. *Nucleic Acids Res.* 36, e52.
- Wilm, A., Mainz, I., Steger, G., 2006. An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms Mol Biol.* 1, 19.
- Wong, A., 2011. The molecular evolution of animal reproductive tract proteins: What have we learned from mating-system comparisons? *Int. J. Evol. Biol.* 2011, 908735. <https://doi.org/10.4061/2011/908735>.
- Yamane, K., Yano, K., Kawahara, T., 2006. Pattern and rate of indel evolution inferred from whole chloroplast intergenic regions in sugarcane, maize and rice. *DNA Res.* 13, 197–204.
- Zhang, Z., Gerstein, M., 2003. Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res.* 31, 5338–5348.

Further Reading

- Brudno, Dubchak, 2005. A fairly comprehensive and systematic review of genome alignment methods in the early days (until 2004), as well as of visualization and applications of the alignments. In: Aluru, S. (Ed.), *Handbook of Computational Molecular Biology*. Chapman and Hall/CRC. (ISBN: 1420036270, 9781420036275).
- Dewey, 2012. A quite comprehensive and well-organized review of whole-genome alignment methods (including quite recent ones) and related issues, based on the concept of ‘topoorthology’ and on the broad classification of alignment strategies into the ‘hierarchical’ and ‘local’ approaches. https://doi.org/10.1007/978-1-61779-582-4_8.
- Dubchak, Pachter, 2002. This review discusses challenges that computational biologists would face when they address genome alignment, gene finding and regulatory element discovery. <https://doi.org/10.1093/bib/3.1.18>.
- Earl *et al.*, 2014. This paper describes ‘Alignathon’, the biggest-to-date competitive evaluation of genome alignment methods, in which 10 different teams (with 12 different alignment pipelines) participated. <https://doi.org/10.1101/gr.174920.114>.
- Frazer *et al.*, 2003. It discusses resources and tools that were available around 2003 for comparative genomic study, including genome alignment. <https://doi.org/10.1101/gr.222003>.
- Jayaraj, 2006. Though being unpublished, it reviews quite a few genome alignment methods developed in the early days (until 2005). (Available at: <https://www.semanticscholar.org>.)
- Kehr *et al.*, 2014. This article compares different types of graphs, which are essential for some whole-genome aligners (and synteny mappers), and discusses relationships between these graphs. <https://doi.org/10.1186/1471-2105-15-99>.
- Kim, J., Sinha, S., 2007. Indelign: A probabilistic framework for annotation of insertions and deletions in a multiple alignment. *Bioinformatics* 23, 289–297.
- Margulies, Birney, 2008. It critically overviews the sequence data and computational methods available (around 2008) for comparative genomic analyses, especially genome sequence alignment and functional element detection. <https://doi.org/10.1038/nrg2185>.
- Thompson, 2016a. A fairly comprehensive and well-organized review of whole-genome alignment methods (including recent ones) and related issues. In: Thompson, J., *Statistics for Bioinformatics: Methods for Multiple Sequence Alignment*, first ed. ISTE Press-Elsevier. (ISBN: 0081019610, 9780081019610).

Relevant Websites

- <https://compbio.soe.ucsc.edu/alignathon/>
Alignathon.
- <http://www.ncbi.nlm.nih.gov/BLAST/fasta.shtml>
BLAST TOPICS (FASTA format) – NCBI.
- <http://www.ensembl.org>
Ensembl Browser.

<https://usegalaxy.org/>

Galaxy.

<http://genome.ucsc.edu/FAQ/FAQformat.html>

Genome Browser FAQ – UCSC Genome Browser.

<https://genome.ucsc.edu>

UCSC Genome Browser.

<http://genome.lbl.gov/vista/index.shtml>

VISTA Browser.

Phylogenetic Footprinting

Hiroyuki Toh, Kwansei Gakuin University, Sanda, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Identification of regulatory motifs such as transcription factor binding sites in genomic non-coding regions is one of the approaches to understand the mechanisms of gene expression regulation. Recent progress of the sequencing technologies such as 'ChIP-Seq' has made it possible to identify the genome-wide identification of regulatory regions, although such 'wet' studies are not enough to reveal overall regulatory motifs in a genome. The development of the sequencing technologies has also contributed to the accumulation of genomic sequences from various organisms. Thereby, a wide variety of computational approaches for the prediction of regulatory motifs with the sequence data have been developed to complement or accelerate the experimental studies on the gene expression regulation. The computational approaches are roughly classified into two types, which are referred to as 'single species, multiple genes' approach and 'single gene, multiple species' approach, respectively (Wang and Stormo, 2003). The former examines the regulatory motifs of multiple genes in a single genome, which are indicated to be regulated under the same regulatory mechanism. The latter analyzes the regulatory motifs of an orthologous set of a single gene from multiple organisms, in which the conservation of the regulatory mechanism for the orthologous genes from different organisms is implicitly assumed. 'Phylogenetic footprinting' belongs to the latter, the 'single gene, multiple species' approach.

Phylogenetic Footprinting and Related Approaches

Phylogenetic footprinting was first developed by Tagle *et al.* (1988) to predict *cis*-regulatory motifs of embryonic ϵ and γ globin genes of primates. The idea behind the method is quite simple. The motifs involved in gene expression regulation are assumed to be subject to strong functional constraints. Suppose that a residue in such a region is replaced with other residue. If the mutation has ill effects on the regulatory mechanism, the progeny with the mutation would be rapidly excluded from a population by negative selection or purifying selection. By that means, genomic regions responsible for regulatory functions are more conserved than non-coding regions without function. So, it is the first step of phylogenetic footprinting to collect nucleotide sequences of the orthologous non-coding regions from multiple species. For example, the 5' upstream region of a gene is often used for the analysis, although *cis*-regulatory elements are not always restricted in the 5' regions. Next, a multiple alignment of the nucleotide sequences thus collected is made. Standard tools for global multiple alignment can be used for this step. Finally, the degree of conservation was evaluated at each site or with a sliding window. Highly conserved non-coding regions detected from the alignment are predicted as putative regulatory motifs (see Fig. 1). Several methods to improve the performance of phylogenetic footprinting have been reported. For example, an approach to use statistical alignment instead of a fixed alignment was proposed for phylogenetic footprinting (Satija *et al.*, 2008). A Bayesian method named BigFoot (Satija *et al.*, 2009), which was developed as an extension of the method with the statistical alignment, used Markov chain Monte Carlo sampling for both alignment and conserved regions.

There are several limitations in phylogenetic footprinting. One of them is related to sequence divergence. Moderate sequence divergence can highlight the conserved regions in a multiple sequence alignment. When sequences under consideration are highly diverged, however, an inaccurate alignment may be generated. Regulatory motifs are often short, comparing to the lengths of the

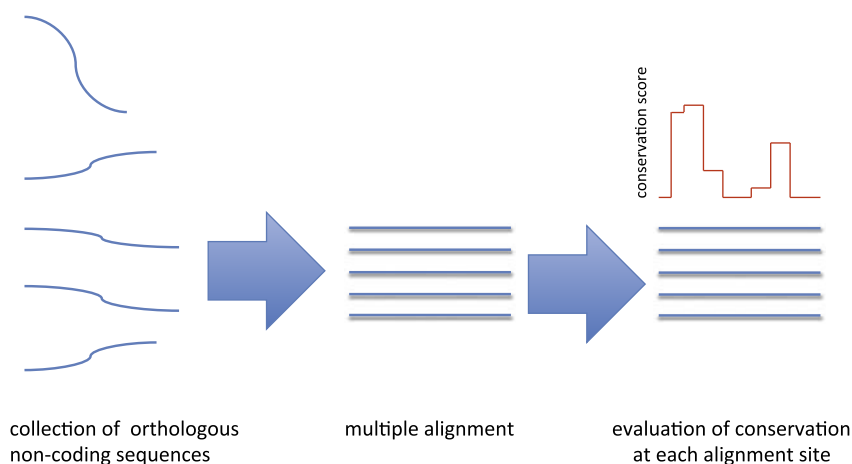


Fig. 1 Standard flowchart of phylogenetic footprinting.

entire sequences under examination. In that case, even if each sequence has the regulatory motif, they may not be correctly matched each other in the alignment, which leads to failure in detection of the regulatory motifs. Inversely, when sequences under consideration are closely related, the footprinting procedure described above would not work well to detect regulatory motifs, since the number of substituted sites in the closely related sequences is too small to reveal conserved sites. The other limitation is that the implicit assumption behind the analysis does not always hold true. The regulatory motifs have been sometimes deleted or added by the lineage-specific manner. Therefore, complete conservation of regulatory motifs among all the collected orthologous sequences is not always expected. Various methods have been developed to overcome the limitations of the original phylogenetic footprinting.

The problem related to high sequence divergence has been addressed by searching for the short segments shared by the collected orthologous non-coding sequences, instead of making a global multiple alignment. The idea is the same as those of the motif discovery tools for the 'single species, multiple genes' approach. For example, McCue *et al.* (2001) used Gibbs sampler (Lawrence *et al.*, 1993) as a tool for phylogenetic footprinting to detect motifs from bacterial genomes. However, the subject of the motif discovery tools for the 'single species, multiple genes' approach is different from that of the phylogenetic footprinting. The set of non-coding regions of the genes under the same regulatory mechanism, which are indicated by co-expression or immunoprecipitation, is the subject of the approach (Wang and Stormo, 2003). That is, the genes used for the analysis are not always homologous and basically independent. Blanchette and Tompa (2002) pointed out a problem to apply the motif discovery tools for the 'single species, multiple genes' approach to phylogenetic footprinting. Most of the motif discovery tools are designed to find motifs from a set of independent sequences as described above. However, the subject of phylogenetic footprinting consists of orthologous sequences. Consider a set of orthologous sequences, in which the majority of the sequences are closely related to each other and only a few sequences are distantly related to the majority. Then, the patterns present in the closely related sequences would be highly weighted, even if the patterns were not conserved in the distantly related sequences. Therefore, such taxonomic bias should be removed for the motif discovery from the orthologous sequences. FootPrinter (Blanchette *et al.*, 2002; Blanchette and Tompa, 2003), PhyME (Sinha *et al.*, 2004), PhyloGibbs (Siddharthan *et al.*, 2005), and a phylogenetic Gibbs sampler (Newberg *et al.*, 2007) are the tools for phylogenetic footprinting developed by introducing the idea.

A different approach is required when only closely related sequences are available. Boffelli *et al.* (2003) developed a method referred to as 'phylogenetic shadowing' to identify primate specific functional regions in human genome by comparing closely related primate sequences. In the analysis, the log likelihood ratio between the fast and slow mutation models is calculated for each column of the alignment with the phylogenetic tree and mutation rates for 'non-conserved' and 'conserved' regions estimated from a training set. A low log likelihood ratio indicates that the corresponding site is subject to strong constraints. Later, the group extended their method to develop an online shadowing tool, eShadow (Ovcharenko *et al.*, 2004). Ray *et al.* (2008) devised a new shadowing method named CSMET by combining a probabilistic graphical model with a hidden Markov model.

The problem of addition or deletion of regulatory elements is an important problem. Actually, it is considered that such events may have occurred by lineage specific manners. The problem is explicitly treated in an algorithmic method developed by Blanchette *et al.* (2002).

There are studies to integrate phylogenetic footprinting with the motif discovery methods for the 'single species, multiple genes' approach. Wang and Stormo (2003) first identified conserved regions from the alignments of non-coding regions for the co-regulated genes. The conserved regions are expressed as profiles and are compared the profiles from different co-regulated genes to identify the shared motifs. ConSite is also a tool to predict transcription factor binding sites by integrating phylogenetic footprinting with a motif discovery method (Lenhard *et al.*, 2003). PFR-Sampler can integrate the phylogenetic footprints with the information of co-regulation of genes with the Markov chain discrimination algorithm to identify the similar phylogenetic footprinted regions (PFRs) near the co-regulated genes (Grad *et al.*, 2004). Bigelow *et al.* (2004) developed a pipeline named CisOrtho, in which non-coding sequences from two species under consideration are first scanned by a weight matrix respectively. Then, the hit regions thus obtained are compared to identify the conserved motifs between the two species. The latter step was regarded as phylogenetic footprinting. TargetOrtho is an expanded version of CisOrtho (Glenwinkel *et al.*, 2014). One of the interesting expansions is the introduction of relaxed definition of motif conservation based on the alignment-free approach (Elemento and Tavazoie, 2005; Gordán *et al.*, 2010). Probabilistic frameworks to integrate phylogenetic footprinting and motif discovery are described in a review by van Nimwegen (2007).

Some of the online tools currently available are listed in Table 1.

Table 1 List of the online tools for phylogenetic footprinting and shadowing

Tool	Ref.	URL
MicroFootprinter	Neph and Tompa (2006)	http://bio.cs.washington.edu/microfootprinter
PhyloGibbs	Siddharthan <i>et al.</i> (2005)	http://www.phylogibbs.unibas.ch/cgi-bin/phylogibbs.pl
eShadow	Ovcharenko <i>et al.</i> (2004)	https://eshadow.dcode.org/
CSMET	Ray <i>et al.</i> (2008)	http://www.sailing.cs.cmu.edu/main/?page_id=360
TargetOrtho	Glenwinkel <i>et al.</i> (2014)	http://hobertlab.org/targetortho/

Examples of Application of Phylogenetic Footprinting

As described above, phylogenetic footprinting was first applied to the prediction of the *cis*-regulatory motifs of embryonic ϵ and γ globin genes of primates in 1988 (Tagle *et al.*, 1988). The same group continued the studies of the *cis*-regulatory regions of globin genes with phylogenetic footprinting. In 1992, they identified several conserved sequences in the 5'-flanking regions of the γ and ϵ globin genes by phylogenetic footprinting, some of which were suggested to function as repressors by experiments (Gumucio *et al.*, 1992). They also applied the method to the 5' region of the ϵ globin gene to identify conserved sequences in 1993. The combination of the computational analysis with the experimental study revealed the complex binding profile of trans factors in the 5' region of the ϵ globin gene (Gumucio *et al.*, 1993).

After the pioneering works, the method has been applied to various organisms. Sauer *et al.* (2006) evaluated the validity and usability of phylogenetic footprinting through the comparison of human and rodent based on the known 2678 transcription factor binding sites. They discussed the dependency of phylogenetic footprinting on the criterion of conservation, alignment algorithm and the types of transcription factors. von Bubnoff *et al.* (2005) compared the upstream regions of *Xenopus* and human *Vent2* genes to identify a 21 bp conserved sequence. The conserved sequence overlaps the known BMP response element. Their experimental study suggested the necessity and sufficiency of the conserved sequence for the BMP responsiveness. Matsunami *et al.* (2010) applied the phylogenetic footprinting to the identification of the conserved non-coding regions among the paralogues of Hox clusters generated by the 2-round whole genome duplications. Cliften *et al.* (2003) compared six *Saccharomyces* genomes. They found that some of the conserved sequences identified by the analysis are present in the upstream of the genes with similar annotations, similar expression patterns, or are in the regions bound by the same transcription factor. Buchanan *et al.* (2004) compared the 5'-non-coding regions of *rab16/17* genes from sorghum, maize, and rice to identify the *cis*-regulatory elements. Lutz *et al.* (2017) aligned the genomic sequence of *Arabidopsis* *FLM* (*FLOWERING LOCUS M*) with those of *FLM* homologues from five other *Brassicaceae* species, together with that of its closest homologue *MAF3*, for the phylogenetic footprinting analysis. They found conserved non-coding sequences in the promotor region and the first intron. They revealed that the conserved regions thus detected are involved in the basal expression of the gene by experimental studies. Aceto *et al.* (2010) combined the motif search with the phylogenetic footprinting to identify putative transcription factor binding sites corresponding to the conserved 5'-non-coding sequences of the *OrcPI* gene from the orchids *Orchis italica*. The homologous non-coding sequences from two other monocots, *Lilium regale* and *Oryza sativa*, and one dicots, *Arabidopsis thaliana* were used for the footprinting analysis. Phylogenetic footprinting is applied not only to eukaryotes, but also to prokaryotes (Neph and Tompa, 2006; Katara *et al.*, 2012; Liu *et al.*, 2016).

Phylogenetic shadowing is also widely utilized for the detection of conserved non-coding regions. The method was first introduced by Boffelli *et al.* (2003). They compared the sequences of primates consisting of Old World monkeys, New World monkeys and hominoids, to identify primate specific regulatory elements and exon boundaries. Clark *et al.* (2005) applied the phylogenetic shadowing to the identification of the conserved non-coding regions of *CAPN10* (calpain 10), a genetic factor of type 2 diabetes, through the comparison of 10 primates sequences. They discussed the possibility that some of the detected regions may be associated with the expression of *CAPN10*.

What phylogenetic footprinting does is to identify the conserved or slowly evolving regions. So, the application of the method is not restricted to the detection of regulatory motif. Actually, both phylogenetic footprinting and shadowing are applied not only to regulatory motif discovery but also to the detection of microRNA genes (Berezikov *et al.*, 2005; Gao *et al.*, 2013).

The examples described above are quite limited, and there are more literatures on the applications of phylogenetic footprinting and shadowing. However, the term, "phylogenetic footprinting" or "phylogenetic shadowing", is often missing from abstracts of the literatures. The search with combination of keywords such as "conservation" and "regulatory elements" could lead to the literatures of the studies with phylogenetic footprinting or shadowing.

Concluding Remarks

The idea behind the phylogenetic footprinting and the development of the approach were described in this section. All the methods described above utilize only sequence data to predict regulatory elements as conserved non-coding regions. However, an insightful review by Maeso *et al.* (2013) suggests novel directions to predict regulatory elements beyond the sequence-based techniques of current phylogenetic footprinting. Metazoans share basic developmental process such as genetic toolkits. So, it was expected that the *cis*-regulatory elements related to the process are conserved across animal phyla. However, it turned out that the prediction of *cis*-regulatory elements based on sequence conservation such as phylogenetic footprinting is difficult when genomes of different phyla are compared. They stressed that the approach with the next generation sequencer (NGS) such as ChIP-seq and DNase I footprint is efficient for the genome-wide identification of *cis*-regulatory elements. They also suggested the possibility that the functionally equivalent *cis*-regulatory elements occupy similar syntenic positions, and that the inclusion of such positional information is useful to avoid the false positives of sequence-based predictions. The introduction of NGS data and the syntenic information could shed new light on the conservation and evolution of *cis*-regulatory elements across phyla.

See also: Algorithms for Strings and Sequences: Pairwise Alignment. Comparative and Evolutionary Genomics. Comparative Genomics Analysis. Genome Alignment. Genome Annotation. Genome Databases and Browsers. Genome Informatics. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Binding Sites in DNA Sequences. Quantitative Immunology by Data Analysis Using Mathematical Models. Sequence Analysis. Sequence Composition

References

- Aceto, S., Cantone, C., Chiaiese, P., *et al.*, 2010. Isolation and phylogenetic footprinting analysis of the 5'-regulatory region of the floral homeotic gene *OrcPI* from *Orchis italica* (Orchidaceae). *J. Heredity* 101, 124–131.
- Berezikov, E., Guryev, V., van de Belt, J., *et al.*, 2005. Phylogenetic shadowing and computational identification of human microRNA genes. *Cell* 120, 21–24.
- Bigelow, H.R., Wenick, A.S., Wong, A., Hobert, O., 2004. CisOrtho: A program pipeline for genome-wide identification of transcription factor target genes using phylogenetic footprinting. *BMC Bioinform.* 5, 27.
- Blanchette, M., Schwikowski, B., Tompa, M., 2002. Algorithms for phylogenetic footprinting. *J. Comput. Biol.* 9, 211–223.
- Blanchette, M., Tompa, M., 2002. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res.* 12, 739–748.
- Blanchette, M., Tompa, M., 2003. FootPrinter: A program designed for phylogenetic footprinting. *Nucleic Acids Res.* 31, 3840–3842.
- Boffelli, D., McAuliffe, J., Ovcharenko, D., *et al.*, 2003. Phylogenetic shadowing of primate sequences to find functional regions of the human genome. *Science* 299, 1391–1394.
- Buchanan, C.D., Klein, P.E., Mullet, J.E., 2004. Phylogenetic analysis of 5'-noncoding regions from the ABA-responsive *rab16/17* gene family of sorghum, maize and rice provides insight into the composition, organization and function of *cis*-regulatory modules. *Genetics* 168, 1639–1654.
- Clark, V.J., Cox, N.J., Hammond, M., Hanis, C.L., Rienzo, A.D., 2005. Haplotype structure and phylogenetic shadowing of a hypervariable region in the *CAPN10* gene. *Hum. Genet.* 117, 258–266.
- Cliften, P., Sudarsanam, P., Desikan, A., *et al.*, 2003. Finding functional features in *Saccharomyces* genomes by phylogenetic footprinting. *Science* 301, 71–76.
- Elemento, O., Tavazoie, S., 2005. Fast and systematic genome-wide discovery of conserved regulatory elements using a non-alignment based approach. *Genome Biol.* 6, R18.
- Gao, D., Middleton, R., Rasko, J.E.J., Ritchie, W., 2013. miREval 2.0: A web tool for simple microRNA prediction in genome sequences. *Bioinformatics* 29, 3225–3226.
- Glenwinkel, L., Wu, D., Minevich, G., Hobert, O., 2014. TargetOrtho: A phylogenetic footprinting tool to identify transcription factor targets. *Genetics* 197, 61–76.
- Gordân, R., Narlikar, L., Hartemink, A.J., 2010. Finding regulatory DNA motifs using alignment-free evolutionary conservation information. *Nucleic Acids Res.* 38, e90.
- Grad, Y.H., Roth, F.P., Halfon, M.S., Church, G.M., 2004. Prediction of similarly acting *cis*-regulatory modules by subsequence profiling and comparative genomics in *Drosophila melanogaster* and *D. pseudoobscura*. *Bioinformatics* 20, 2738–2750.
- Gumucio, D.L., Heilstedt-Williamson, H., Gray, T.A., *et al.*, 1992. Phylogenetic footprinting reveals a nuclear protein which binds to silencer sequences in the human γ and ϵ globin genes. *Mol. Cell. Biol.* 12, 4919–4929.
- Gumucio, D.L., Shelton, D.A., Bailey, W.J., Slightom, J.L., Goodman, M., 1993. Phylogenetic footprinting reveals unexpected complexity in trans factor binding upstream from the ϵ -globin gene. *Proc. Natl. Acad. Sci. USA* 90, 6018–6022.
- Katara, P., Grover, A., Sharma, V., 2012. Phylogenetic footprinting: A boost for microbial regulatory genomics. *Protoplasma* 249, 901–907.
- Lawrence, C.E., Altschul, S.F., Boguski, M.S., *et al.*, 1993. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* 262, 208–214.
- Lenhard, B., Sandelin, A., Mendoza, L., *et al.*, 2003. Identification of conserved regulatory elements by comparative genome analysis. *J. Biol.* 2, 13.
- Liu, B., Zhang, H., Zhou, C., *et al.*, 2016. An integrative and applicable phylogenetic footprinting framework for *cis*-regulatory motifs identification in prokaryotic genomes. *BMC Genom.* 17, 578.
- Lutz, U., Nussbaumer, T., Spannagl, M., *et al.*, 2017. Natural haplotypes of FLM non-coding sequences fine-tune flowering time in ambient spring temperatures in *Arabidopsis*. *eLife* 6, e22114.
- Maeso, I., Irimia, M., Tena, J.J., Casares, F., Gómez-Skarmeta, J.L., 2013. Deep conservation of *cis*-regulatory elements in metazoans. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 368, 20130020.
- Matsunami, M., Sumiyama, K., Saitou, N., 2010. Evolution of conserved non-coding sequences within the vertebrate Hox clusters through the two-round whole genome duplications revealed by phylogenetic footprinting analysis. *J. Mol. Evol.* 71, 427–436.
- McCue, L., Thompson, W., Carmack, C.S., *et al.*, 2001. Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Res.* 29, 774–782.
- Neph, S., Tompa, M., 2006. MicroFootPrinter: A tool for phylogenetic footprinting in prokaryotic genomes. *Nucleic Acids Res.* 34, W366–W368.
- Newberg, L.A., Thompson, W.A., Conlan, S., *et al.*, 2007. A phylogenetic Gibbs sampler that yields centroid solutions for *cis*-regulatory site prediction. *Bioinformatics* 23, 1718–1727.
- Ovcharenko, I., Boffelli, D., Loots, G.G., 2004. eShadow: A tool for comparing closely related sequences. *Genome Res.* 14, 1191–1198.
- Ray, P., Shringarpure, S., Kolar, M., Xing, E.P., 2008. CSMET: Comparative genomic motif detection via multi-resolution phylogenetic shadowing. *PLOS Comput. Biol.* 4, e1000090.
- Satija, R., Novák, Á., Miklós, I., Lyngsø, R., Hein, J., 2009. BigFoot: Bayesian alignment and phylogenetic footprinting with MCMC. *BMC Evol. Biol.* 9, 217.
- Satija, R., Pachter, L., Hein, J., 2008. Combining statistical alignment and phylogenetic footprinting to detect regulatory elements. *Bioinformatics* 24, 1236–1242.
- Sauer, T., Shelest, E., Wingender, E., 2006. Evaluating phylogenetic footprinting for human–rodent comparisons. *Bioinformatics* 22, 430–437.
- Siddharthan, R., Siggia, E.D., van Nimwegen, E., 2005. PhyloGibbs: A gibbs sampling motif finder that incorporates phylogeny. *PLOS Comput. Biol.* 1, e67.
- Sinha, S., Blanchette, M., Tompa, M., 2004. PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinform.* 5, 170.
- Tagle, D.A., Koop, B.F., Goodman, M., *et al.*, 1988. Embryonic ϵ and γ globin genes of a prosimian primate (*Galago crassicaudatus*): Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J. Mol. Biol.* 203, 439–455.
- van Nimwegen, E., 2007. Finding regulatory elements and regulatory motifs: a general probabilistic framework. *BMC Bioinform.* 8, S4.
- von Bubnoff, A., Peiffer, D.A., Blitz, I.L., *et al.*, 2005. Phylogenetic footprinting and genome scanning identify vertebrate BMP response elements and new target genes. *Dev. Biol.* 281, 210–226.
- Wang, T., Stormo, G.D., 2003. Combining phylogenetic data with co-regulated genes to identify regulatory motifs. *Bioinformatics* 19, 2369–2380.

Chromatin: A Semi-Structured Polymer

Wayne K Dawson, CeNT, University of Warsaw, Warsaw, Poland and BiLab, The University of Tokyo, Tokyo, Japan

Michał Łazniewski, CeNT, University of Warsaw, Warsaw, Poland and Medical University of Warsaw, Warsaw, Poland

Dariusz Plewczynski, CeNT, University of Warsaw, Warsaw, Poland and MINI, Warsaw University of Technology, Warsaw, Poland

© 2019 Elsevier Inc. All rights reserved.

Introduction

It has long been a question whether the genome has spatial organization and to what extent it influences its function. A critical part of modern medicine might depend on our understanding how the genome organization works in the context of replication and gene expression and its regalia of promoters, enhancers, the DNA associated proteins, including RNA polymerase II (Rpol2), and the spliceosome apparatus (SA). The most crucial question is does gene expression depend on the spatial proximity of the promoters, inhibitors, and enhancers? What role, if any, does proximity play? How do Rpol2 and SA operate in conjunction with unraveling the nucleosome and how does this factor into exon/intron splicing? How does a developing organism determine what genes to express and when and how is that controlled? What happens when proper functioning of the cell fails? There are many unanswered questions in the field of genomics; nevertheless, recent research has opened the door to at least a partial understanding of some of these important issues.

Progress in our understanding of nuclear architecture has advanced due to the ongoing development of various imaging techniques like FISH (fluorescent in situ hybridization), confocal microscopy (O'Connor, 2008; Wurtele and Chartrand, 2006; Wang *et al.*, 2015; Lomvardas *et al.*, 2006; Rosa *et al.*, 2010; Fraser *et al.*, 2015; Giorgetti *et al.*, 2014; Muller *et al.*, 2010) and electron microscopy (Ou *et al.*, 2017) as well as in contact mapping techniques especially chromatin conformation capture (3C) method (Wurtele and Chartrand, 2006; Dekker *et al.*, 2002; Lomvardas *et al.*, 2006) and various techniques derived from 3C; e.g., Hi-C (Rao *et al.*, 2014; Lieberman-Aiden *et al.*, 2009; Jin *et al.*, 2013; Fraser *et al.*, 2015; Dixon *et al.*, 2012), ChiA-PET (Ji *et al.*, 2016; Tang *et al.*, 2015; Li *et al.*, 2017; Fullwood *et al.*, 2009a) or HiChIP (Mumbach *et al.*, 2016). The details of these methods are available elsewhere (Fraser *et al.*, 2015) and in this review we provide only a brief summary regarding some of these techniques. There is a growing consensus that there is considerable (quasi) organization in the arrangement of eukaryotic chromatin in the nucleus and that the chromatin fiber exhibits a considerable degree of compartmentalization.

It should seem puzzling that a 6 billion base pair (bp, 6 Gbp) somatic chromatin fiber arranges itself in such a way that it does not become tied up in knots, and organizes itself efficiently into 46 chromosomes in a repeatable way during its mitosis phase. As a thought experiment, consider what happens to a pair of earphone when dumped into a carry bag with other stuff and rattled around for a while. The earphones quickly become tangled into at least one knot. Moreover, the cell itself is a highly dynamic yet in many ways disordered system too. How then does a 6 Gbp genome keep from getting thoroughly tangled up like the earphones yet at the same time remain only partially ordered?

In this review, we will look briefly through the levels of the DNA organization starting from the double stranded DNA (dsDNA) up to the interactions between different parts of the chromatin. We will briefly discuss some of the current experimental and computational methods aiming to predict chromatin conformations.

Chromatin 3D Interactions: Short Range to Long Range

Here we take a progressive picture of chromatin from the components at the coarse-grained scale of histones (~nm) to the long range binding between distant parts of the chromatin, which extends to about 1 Mbp and perhaps even as large as 3 Mbp.

The Local Structure of the Nucleosomes

Chromatin is a complex heteropolymer comprised of many different components. Double-stranded DNA (dsDNA) in eukaryotic organisms is bundled into packages called nucleosomes. The dsDNA wraps about an octamer of core histone proteins (roughly 7/4 turns; about 147 bp) like on a solenoid (Fig. 1). The octamer contains a tetrameric core of H3 and H4 proteins bound together and forming a dimer structure (H3-H4)₂ and this dimer is capped on both sides of the solenoid like structure by a heterodimer of H2A-H2B (Nacheva *et al.*, 1989). In contrast to core histones, the linker histones, like H1, and its isoforms are not components of the nucleosome core, but instead bind to the linker DNA at the nucleosome entry and exit sites, stabilizing the entire complex (Bustin *et al.*, 2005; Kalashnikova *et al.*, 2016). Some examples of structural fragments of dsDNA wrapped around a core of histone proteins – obtained via X-ray diffraction – can be found in the Protein Data Bank (PDB); these include 5DNM, 5VA6, 5B31 (Fig. 1) or 5KGF.

Unless specified otherwise, the word “genome” represents a sequence corresponding to one complete set of autosomes (chromosomes not associated with the sex of the organism) and one of each type of sex chromosome; the result representing both possible sexes. In addition, the “genome sequence” is often a composite read from various individuals. Therefore, the length of a “genome” roughly represents to the length of the haploid cell or gamete (sperm/ovum). The more typical somatic cells of an

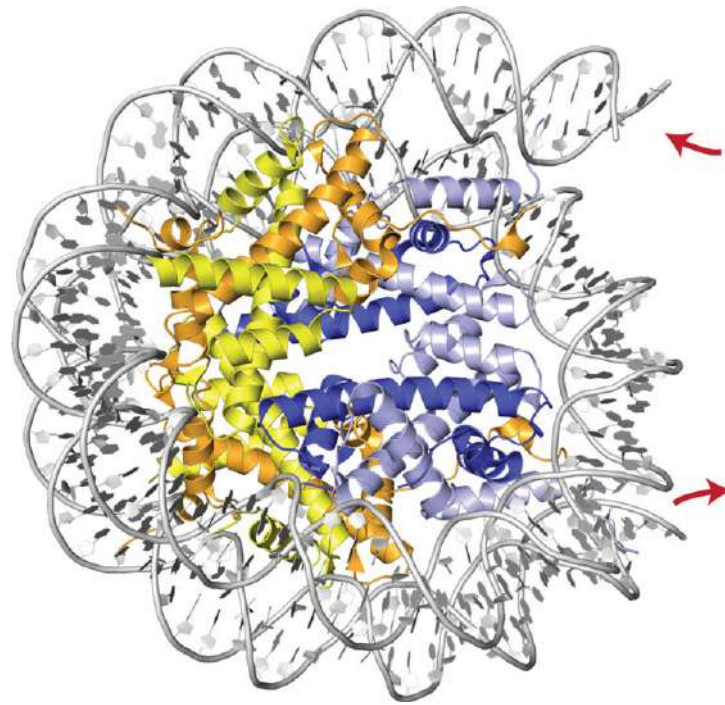


Fig. 1 Example of the histone and double stranded DNA (dsDNA) complex (PDB id: 5B31) showing the 1 and 3/4th turn of the dsDNA around the histone core. The coloring scheme for the histone subunits and DNA are as follows: silver blue – H3 (chains A + E), blue – H4 (chains B + F), orange – H2A (chains C + G), yellow – H2B (chains D + H), silver – dsDNA (chain I, J). The Figure is created using Pymol (Delano, 2004; pymol available from <http://www.pymol.org>).

organism contain a fused copy of both the male and female haploid cell DNA. Haploid cells are formed through meiosis (splitting in half the diploid). Somatic cells are replicated through the process of mitosis.

The human genome consists of three billion base pairs (3 Gbp) of dsDNA. The haploid cells have 23 chromosomes and the dsDNA sequence is a monoploid roughly 3 Gbp long. The somatic cells are diploids with 46 chromosomes and a sequence length of approximately 6 Gbp. Naked dsDNA (the dsDNA without the histones) from somatic cells would reach two meters ($2 \text{ [haploid]})(3 \times 10^9 \text{ [bp/haploid]})(0.34 \text{ [nm/bp]})$ if it were stretched out over its full contour length; overstretched (Olson and Zhurkin, 2000), this length would even double. When packaged in the form of chromatin fiber, the overall length is reduced by about one third (Rosa and Zimmer, 2014) where each solenoid shaped nucleosome core is about 6 nm by 11 nm. The distance between each nucleosome is called the *nucleosome repeat length*, (NRL (Routh et al., 2008)). The histone core is extended by H1 linkers. There is nearly always at least one H1 between adjoining nucleosome cores leading to a $n + 1$ rule (Routh et al., 2008). Each H1 linker tends to increase the NRL in increments of 10 bp; coincidentally almost a full 360° turn of the dsDNA helix (Voong et al., 2016; Brogaard et al., 2012). As a result, the NRL ranges from 167 bp (the histone core plus H1 plus one H1 linker, 10 bp; e. g., yeast) to 240 bp (core plus H1 plus an 80 bp linker, echinoderm sperm) (Bascom and Schlick, 2017). The nucleosomes appear to be connected as in an irregular zig-zag (Woodcock et al., 1993; Bednar et al., 1998), which resembles a loose coil formed when “flaking” a rope (Bascom and Schlick, 2017); i.e., rolling out a mountain climbing rope. The globules of dsDNA and histones (the nucleosomes), and the linker(s) form the basic chromatin fiber of somatic cells and gametes.

The H1 linker and other related factors can affect the compactness of the chromatin. For example, the length of the H1 linker appears to influence the organization of the chromatin into an ordered structure. In a study by Routh et al. (2008), chromatin compaction was analyzed when the NRL was either 167 or 197 bp and the linker histone H5 was present or not. The H5 histone is found in chicken erythrocytes and is a variant of the linker histone H1. H5 has a greater binding affinity to DNA than H1, but the influence of H5 on compaction is rather similar. In the absence of H5 (Fig. 2, top two panels), the dsDNA appear to be disordered, especially in the case when NRL was 197 bp. When H5 is present, the structure appears to be ordered. In Fig. 2 (bottom left panel, “+ H5”), when H5 is present and the NRL is equal to 167 bp, the nucleosome appear to form a structure similar to a 20 nm fiber. On the other hand, the 197 bp NRL yields something closer to the 30 nm fiber (Fig. 2, lower right panel).

Depending on the concentration of salt in the buffer, at the length-scale of several nucleosomes and corresponding linkers (NRL approximately 200 bp), there is a tendency for the nucleosomes to become partially ordered. It was long thought that the chromatin beaded up into 30 nm globules (Finch and Klug, 1976); however, such structural regularity has also been challenged (Davie, 1997; Woodcock et al., 1993). Based on observations of small angle x-ray scattering (SAXS) studies on chromatin from HeLa cells, the structure of the chromatin fiber appears to take on a more complex local structure or even possibly random

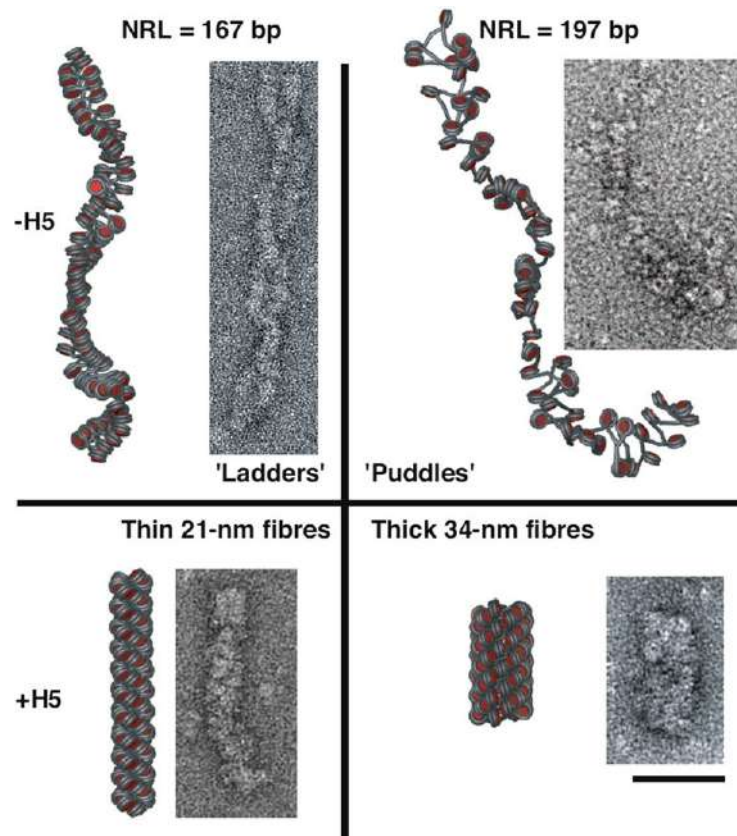


Fig. 2 The role of the linker histone in determining higher-order chromatin structure for different nucleosome repeat lengths (NRL); 167 bp and 197 bp. (*Upper*) Without the linker histone. (*Lower*) With the linker-histone. To the left of each example, a sample of a simulated structure is shown to provide an impression of the overall arrangement of the nucleosomes. (*Upper*) Here, it can be seen that both the 167 and 197 bp NRL are largely disordered or certainly rather loose in the absence of H5; the 197 case is more so than the 167 case, which is at least wrapped. (*Lower*) Here, it can be seen that both the 167 and 197 bp fibers are quite compact in the presence of H5. (Scale bar: 50 nm.). Reproduced from Routh, A., Sandin, S., Rhodes, D., 2008. Nucleosome repeat length and linker histone stoichiometry determine chromatin fiber structure. *Proc. Natl. Acad. Sci. USA* 105, 8872–8877.

structure (Maeshima *et al.*, 2010; Joti *et al.*, 2012; Nishino *et al.*, 2012; Maeshima *et al.*, 2016). The studies using SAXS to analyze the rough structure of the nuclear material of the HeLa cells revealed a clear 30 nm signal (Joti *et al.*, 2012; Nishino *et al.*, 2012) in the presence of Rpol2. However, after washing away the Rpol2, the peak disappeared. The SAXS data also shows clear 6 nm and 11 nm peaks and a broadband peak for longer length scales (Maeshima *et al.*, 2014) indicating a local lattice-like clustering of 3 or 4 nucleosomes; consistent with a local stacking and tandem arrangement nucleosome units.

The H1 histone protein is particularly sensitive to environmental factors such as ionic strength (Davie, 1997). For example, the existence of 30 nm fibers appears to be partially dependent on the concentration of Mg^{2+} . Later work from Maeshima and coworkers focused more on the salt buffer in the absence of Rpol2 (Maeshima *et al.*, 2016). In Fig. 3(a), the diameter of the HeLa cell nucleus is found to be significantly inflated in the absence of Mg^{2+} (16.4 μm) and becomes increasingly compact with addition of Mg^{2+} (around 8.6 μm , where normal HeLa cell diameters appear to be 7.36 μm (Milo *et al.*, 2010) BNID 104990). In Fig. 3(b), SAXS measurements of nucleosome oligomers (blue) and nuclear material (red) show similar profiles; both show no obvious structure other than a broad peak in the absence of Mg^{2+} and sharper images of 6 and 11 nm peaks at increasing concentrations of Mg^{2+} . In the same study by Maeshima *et al.*, the authors also examined isolated native chicken chromatin fragments that showed similar profiles as the human HeLa cells in Fig. 3(b). Hence, using the chicken linker histone (H5) in Fig. 2 (Routh *et al.*, 2008) instead of human H1 does not change the behavior of chromatin. Finally, Fig. 3(c) shows a proposed progression from a disorder local structure of the nucleosomes in chromatin in the absence of Mg^{2+} , to very compact and possibly disordered structure at higher concentrations of Mg^{2+} . The ion concentrations within the cell of most animals is roughly 140 mM K^+ , 12 mM Na^+ and about 1 mM Mg^{2+} whereas outside the cell it is about 150 mM Na^+ (Auffinger *et al.*, 2011). Therefore, the middle panel in Fig. 3(c) represents the chromatin state closest to the physiological conditions typically found in human cells.

It might be tempting to assume that a regular double helix of dsDNA has almost no influence on the positioning of the nucleosomes on the dsDNA; however, this is not the case. In Todolli *et al.* (2017), it was observed that the bases in the dsDNA sequence can influence the positioning of the histones and consequently the arrangement of the nucleosomes. Presently, there are no first principles prediction methods at the bp level that can explain the arrangement of the nucleosomes (Todolli *et al.*, 2017). If

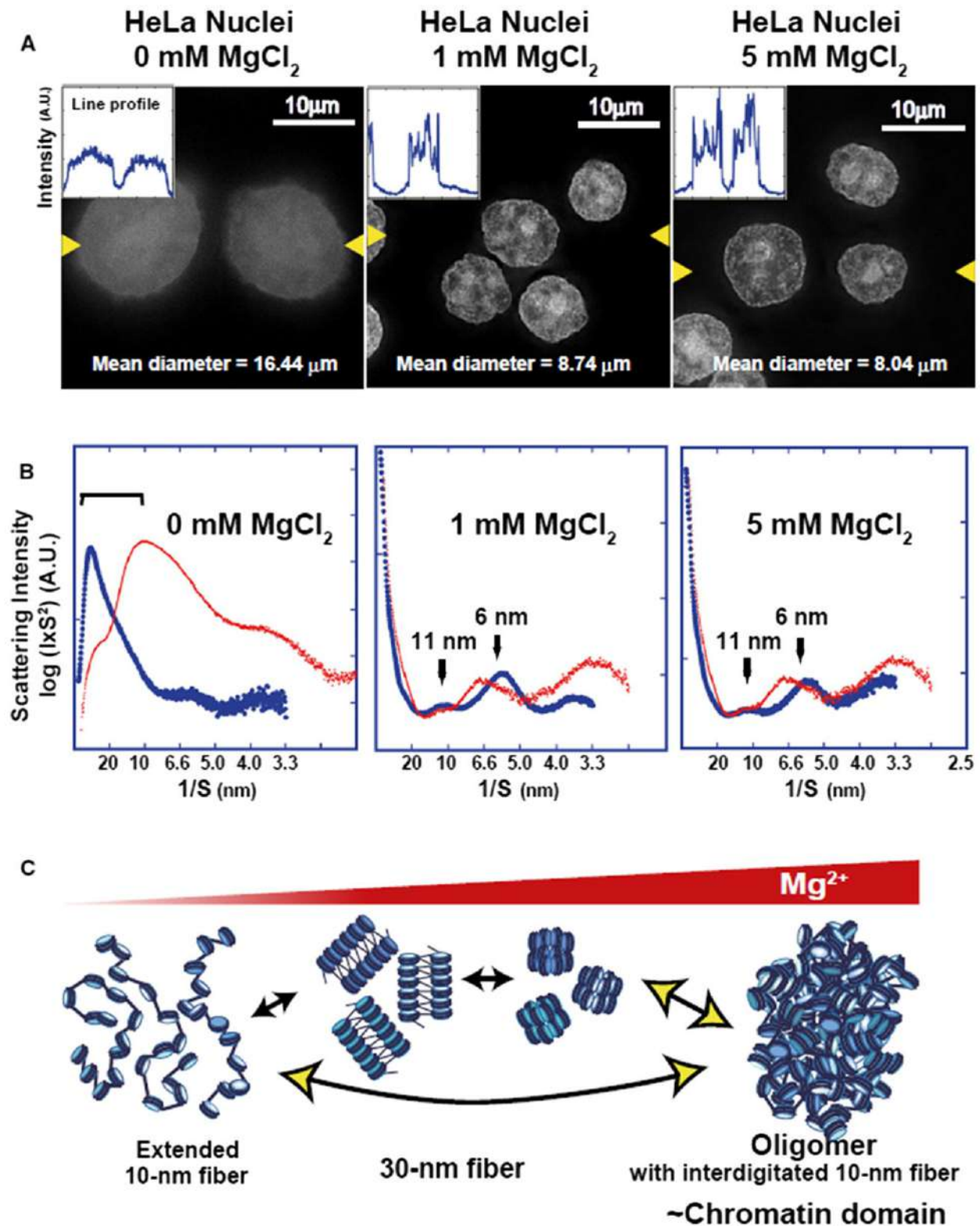


Fig. 3 Isolated HeLa nuclei showing the effects of Mg^{2+} concentration on the overall chromatin structure (nuclear material). (A) FM images of chromatin structure in the nuclei with 0 (left), 1 (center), and 5 mM Mg^{2+} (right). The insets show the intensity line profiles as measured by scanning (horizontally) between the two yellow arrow heads in each of the images. (B) Sax profiles of nucleosome arrays formed by an oligomer of H1-linked nucleosomes (red) and by HeLa nuclei (red); 0 mM (left), 1 mM (center) and 5 mM Mg^{2+} (right). (C) A schematic of the effects of Mg^{2+} concentration on the characteristic form of chromatin fibers for a 12-mer nucleosomal array. In 0 mM Mg^{2+} , the nucleosomes are highly disordered and separated, in 1–2 mM Mg^{2+} , there is some evidence of regular arrangements of nucleosomes that to some extent generated a 30 nm chromatin fiber structure, though the 30 nm peak is not so visible in the spectrum. With additional increases in Mg^{2+} , the structure collapses into a structure which is largely disordered but compact. Reproduced from Maeshima, K., Rogge, R., Tamura, S., *et al.*, 2016. Nucleosomal arrays self-assemble into supramolecular globular structures lacking 30-nm fibers. *EMBO J.* 35, 1115–1132.

we were able to better understand this local structure, it would be possible to understand the short-range nucleosome-nucleosome (histone coated dsDNA and linker) structure.

It is a well-established fact that histone modifications impact gene expression by altering the overall nucleosome-nucleosome interactions in the chromatin structure through acylation (Davie, 1997), methylation (Kouzarides, 2002; Cheung *et al.*, 2000), phosphorylation (Cheung *et al.*, 2000) and sometimes sumoylation (Shiio and Eisenman, 2003) and ubiquitylation (Nathan *et al.*, 2003). Acylation and deacetylation of the exposed parts of the histone assists in turning transcription on and off (Davie, 1997). It also promotes the recruitment of proteins that can bind diverse parts of the chromatin fiber into loops, which ultimately form transcription factories that become the loci of highly expressed regions through Rpol2 activity (Dunn *et al.*, 2003; He *et al.*, 2008; Jahan *et al.*, 2016; Warns *et al.*, 2016).

A recent molecular dynamics simulation (Zhang *et al.*, 2017) shows that lysine acetylation of the H4 tail with K16Ac yields an increased diversity of conformations and, consequently, the interactions between nearest neighboring nucleosome units. The positively charged side chains of lysine and arginine tend to repel each other in an isolated tail of H4. K16Ac relieves some of this tension and contributes to the folding and stabilization for a variety of conformations of the chromatin. At the same time, since the dsDNA has negatively charged phosphates, there are extensive electrostatic interactions, particularly salt bridges. When the lysine is acetylated, these electrostatic interactions are disrupted yielding a more unstable interaction at the H4 tail and resulting in a more diverse conformation space.

The role of these various histone modifications is quite complex, but the effects appear to strongly influence gene expression and suppression by exposing some parts in unbound regions and burying other parts rendering them largely inaccessible to transcription. In Rychkov *et al.* (2017), Rychkov and coworkers used atomic force microscopy to study various combination of partially assembled nucleosome states: (dsDNA bound to partial histone subunits); hexasome (H2A · H2B) · (H3 · H4)₂, tetrasome (H3 · H4)₂ and disome H3 · H4. The authors argue that the free energy cost of unwrapping the histones is on the order of 40 kcal/mol, which is extremely strong for proteins. Hence, the collective interaction would result in regions of the dsDNA that are protected by the histone octamer subunits and likely to strongly impede the progress of Rpol2 and SA from unwrapping and transcribing and splicing, respectively. Such a large free energy barrier would result in kinetics where unwrapping the nucleosome could consume as much as 10 min of time, yet the unimpeded rate appears to be around 1.4 kbp/s (Shermoe and O'farrell, 1991) and certainly at least a single nucleosome (~200 bp/s) per second (Tims *et al.*, 2011).

To overcome this issue, it has been argued that the histone binding may very well be rather dynamic (Bustin *et al.*, 2005; Bascom and Schlick, 2017) because the histone octamer appear to exist in dynamic equilibrium, regularly binding and unbinding to the dsDNA. This may also result from the partial separation of a much smaller structural unit like a disome (H3 · H4) separating with only a 40 bp sequence being in contact with the positively charged core. In such a case, the structure would only involve some 10 kcal per mole (because the 40 kcal/mol binding is spread over 8 histone proteins). Such a free energy would only require an unwrapping time of a few milliseconds (Rychkov *et al.*, 2017), which is more consistent with the observed speed of transcription of Rpol2 (Shermoe and O'farrell, 1991), though perhaps there are other possibilities.

The details of nucleosome, the interplay of the histones and dsDNA, dynamics of histone modifications such as above mentioned acetylation, other histone modifications, and how these factors influence transcription remain largely unanswered questions. The interplay between these modified histones and the overall structure of chromatin is currently largely unknown (Bascom and Schlick, 2017). How all these variables combine together to determine the distance between the nucleosomes and the local arrangement of nucleosome-nucleosome interactions remains an area extensive research.

Flexibility of Chromatin: Persistence Length/Kuhn Length

In polymer chemistry (Flory, 1953, 1969; Grosberg and Khokhlov, 1994), the stiffness of a polymer is typically expressed in terms of the persistence length (l_p) or the Kuhn length (ξ). The Kuhn length can be thought of as the average length of segments of a chain and persistence length as describing the average curvature of a wire.

In the case of Kuhn length (Kuhn, 1936), one imagines a polymer that is generated as though one were taking equal-distant steps in random directions. For example, Fig. 4(A) shows the mer-to-mer distance (distance between monomers, the red beads) laid on top of a green cord. On top of the green cord, purple beads are shown at a fixed link distance (yellow), which represents the Kuhn length. The overlay of purple beads is sometimes called a freely-jointed polymer chain, and the overall path can be modeled as a Markov chain and approximated using a Gaussian function. The Gaussian function has the property of self-similarity; a feature where at every scale of resolution, the random walk looks the same. Hence, Kuhn length reflects a selected uniform length scale of the random walk; it is often described as beads on a chain, where the "beads" represent the purple dots and the Kuhn length is the distance between the yellow bars in Fig. 4(A). In general, real polymers only reflect this self-similarity at discrete length scales.

The persistence length (l_p) is based on a similar concept known as the worm-like-chain (Flory, 1969) or Kratky-Porod model ((Kratky and Porod, 1949), *op cit.* (Flory, 1969)). In this model, one imagines a long flexible cord like a rope or a rubber hose of length l . Consider two consecutive points along this chain (k and k' , where $k \neq k'$), with each point along the cord represented by its unit vectors $\mathbf{u}_k$ and the angle between $\mathbf{u}_k$ and $\mathbf{u}_{k'}$: $\mathbf{u}_k \cdot \mathbf{u}_{k'} = \cos(\theta_{k,k'})$. It can be shown that the integration with respect to all the angles $\theta_{k,k'}$ would result in $\cos(\theta) = \exp(-l/l_p)$. The l_p (persistence length) expresses the minimum length of a cord that will flex to a curvature of one radian (about 57.3°) without kinking and remembering its original straight position (bending back). In other words, imagine there are two points on the straight polymer with distance d between them. If we can bend the polymer so that the

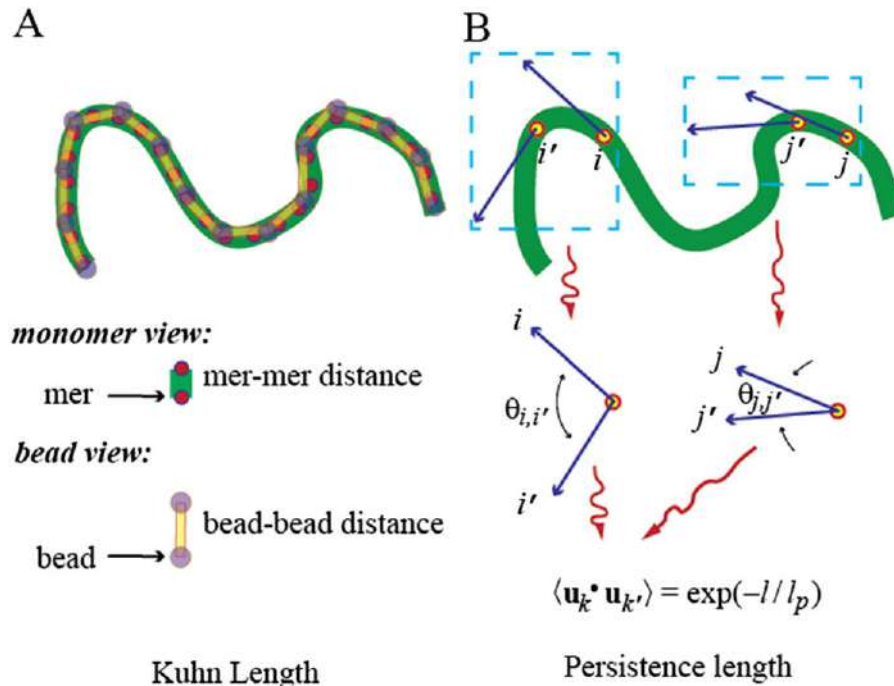


Fig. 4 Comparison of flexibility in terms of Kuhn length and persistence length. (A) Kuhn length where blue beads represent monomers and purple beads represent the beads. (B) Persistence length where vectors i, i' involve a large bend of angle $\theta_{i,i'}$ and j, j' involve only a small bend of angle $\theta_{j,j'}$. All the angles $\theta_{k,k'}$ are integrated yielding $\langle \mathbf{u}_k \cdot \mathbf{u}_{k'} \rangle = \exp(-n/l_p)$, where $n = l/l_p$ is the persistence length of the polymer chain.

angle between the unit vectors from these points (e.g., $\theta_{i,i'}$ or $\theta_{j,j'}$ in Fig 4B) is 1 radian but after we remove the “bending force” the polymer can return to the straight position, then d is effectively l_p . For example, compare l_p for a rubber hose and a thick steel cable; the thick steel cable cannot bend to 57.3° over the same distance as the rubber hose. How about an iron bar? Given a long enough bar, one will see some visible bending but that flexing may only be obvious on length scales on the order of a football field.

For most real polymers, it is found that the distance between the monomers is shorter than the Kuhn length or persistence length. Hence, both models reflect the tendency of the polymer to behave as though the fundamental structural unit is larger than the size of a single monomer (in most cases).

Moreover, real systems are not completely self-similar at all scales; therefore, there is typically a defined Kuhn length for every polymer. The minimum Kuhn length is typically greater than the distance between the monomers. In the case of *naked* dsDNA (without the histones), this can exceed more than 100 bp, depending on salt concentration (Hagerman, 1988). It can be thought of as a kind of coarse-grained resolution length. In typical polymers, it is treated as a statistical parameter and averaged over the entire structure; however, the value should depend on a variety of factors such as chemical binding interactions, local structural interactions, and other considerations. In RNA, the variability depends strongly on the length of the duplex stem and to some extent single-stranded interactions (Dawson *et al.*, 2013).

In the case of chromatin, the polymer comprises a composite of histone proteins and dsDNA. Because the dsDNA is wrapped around the histone core (Fig. 1), there is a local bending of the dsDNA on the scale of 5–10 bp. Because the wrapping of the dsDNA to the histone core has a binding energy of about 40 kcal/mol (Rychkov *et al.*, 2017), a second Kuhn length exists at the scale of the nucleosome (around 200 bp). Figs. 2 and 3 suggest yet a further hierarchical level; The nucleosomes, (about 10 nm in diameter) appear to aggregate into larger collective structures as a result of their mutual attraction. Therefore, the bead size of the cluster of nucleosomes may not necessarily be precisely 30 nm units (Hansen, 2002; Maeshima *et al.*, 2016); however, at physiological conditions, there is little dispute that there is structure encompassing at least several nucleosomes (Maeshima *et al.*, 2016). The collective interaction of the nucleosomes appears to range from 90 nm (Davie, 1997) (roughly the size of 9 nucleosomes) to possibly 40 nm (about 4 nucleosomes).

Although the length scale is probably better understood in terms of clusters of nucleosomes, the NRL is a variable in the cell; therefore, a single base pair of dsDNA is the most regular and invariant unit to define as the “monomer” and five nucleosomes comprise roughly 1 kbp.

Since the differences in the NRL generate very different patterns in Fig. 1, one should probably surmise that the Kuhn length of chromatin is a variable. Bystricky *et al.* (2004) found an overall persistence length of about 170–220 nm, which corresponds to a Kuhn length of 340–440 nm (comprising 34 to 44 nucleosomes). This latter number seems too excessive but heterochromatin is a highly compact material that may have an effective Kuhn length in this range. The variability might also be explained by various histone modifications including (I) the H1 linker (Bocharova *et al.*, 2012; Bascom and Schlick, 2017), (II) H3 phosphorylation,

(III) H4 K16 acylation (H4K16Ac, (Zhang *et al.*, 2017)), (IV) various expression related CpG islands with their corresponding histone modifications (H4K20me1, H2BK5me1 and H3K79me1/2/3) near the transcription start site, and (V) dynamic histone mobility (Bustin *et al.*, 2005). A reasonable estimate to derive from these considerations is a Kuhn length ranging between 3 and 9 nucleosomes (and possibly up to as many as 22). Further, the difference is partly reflected in whether the region is euchromatin (typically loose) or heterochromatin (typically packed). Thus it must be considered that a truer representation of the chromatin chain is that it has a variable bead-to-bead length (Todolli *et al.*, 2017).

Presently, due to both computational issues and the lack of sufficient resolution of experimental methods, computational models of chromatin typically assume that the chromatin chain is composed of beads of a uniform size ranging between 1 kbp to several Mbp. The distance between beads is, therefore, usually much longer than any Kuhn length associated with either heterochromatin or euchromatin. For the shortest lengths of 1 kbp, there may be consequences to assuming complete plasticity of the chromatin beads if we assume the maximum range of 44 nucleosomes (roughly 8 kbp). However, the SAXS data in Fig. 3 Maeshima *et al.* (2016) suggests that 44 is probably rather extreme. The evidence suggests something akin to 30 nm fibers at physiological salt, possibly ranging between 3 and 6 nucleosomes. Hence, certainly at 5 kbp resolution, chromatin can be treated as a set of uncoupled bead on a chain, and, it would be better to be thinking of corrections for the fact that the true bead size is actually *shorter* than the 5 kbp segments that a computational program would bin the data into. Hence, there is likely to be some loss of information in coarse-graining to such length scales.

It seems arguable that using 5 kbp “beads” represents a range where it is possible to describe chromatin as a Gaussian polymer chain (with a self-avoiding walk correction) and possibly some non-ideal polymer behavior associated with the solvent and buffering characteristics (Flory, 1953, 1969; Grosberg and Khokhlov, 1994; Dawson and Kawai, 2015) of the environment (the nucleus) where the chromatin resides. To this polymer model, one would compute the long-range chromatin interactions mediated by various protein factors (such as CTCF, or Rpol2). Nevertheless, we should be mindful that this uniform length scale homo-polymer Kuhn length is a very rough approximation.

The Role of CTCF Protein and Cohesin in Genome Organization

In addition to the obvious structure of chromosomes during cell division, the chromatin in the interphase state appears to organize into compartments (Munkel *et al.*, 1999; Bystricky *et al.*, 2004). Moreover, on the length scale of a 20 kbp to about 3 Mbp, there appears to be considerable long-range pairwise organization of the chromatin according to both FISH (Guo *et al.*, 2015; Xu and Corces, 2016; Munkel *et al.*, 1999; Bystricky *et al.*, 2004) and any of the chromosome conformation capture (3C) methods particularly ChIA-PET (Guo *et al.*, 2015; Xu and Corces, 2016; Tang *et al.*, 2015) and Hi-C (Dixon *et al.*, 2012; Lieberman-Aiden *et al.*, 2009; Rao *et al.*, 2014). These methods will be discussed in the Subsection titled “Experimental Methods”.

At the level of genetics, there is a plethora of interactions within the chromatin (Ji *et al.*, 2016; Beagan *et al.*, 2016). Several kinds of genetic features that are of special interest here are insulators, enhancers and promoters. The genetic boundary elements, known as insulators, have the ability to effectively block the interaction between enhancers and promoters. The promoters mark the region of the dsDNA sequence that attracts proteins that initiate transcription and are typically located near the transcription start sites (upstream on the sense strand). Enhancers are relatively short fragments of dsDNA where transcription factors can bind to increase the likelihood of transcription of a particular gene. Enhancers amplify the transcription rate of a specified gene (Sapojnikova *et al.*, 2009). Unlike promoters, enhancers can be located far away from the transcription start site on the dsDNA chain (Fraser *et al.*, 2015).

The CTCF protein factor has long been associated with insulators (Lewis and Murrell, 2004; Dunn *et al.*, 2003), tending to mark the boundaries between salt soluble regions (often expressed) and insoluble regions (rarely expressed) (Jahan *et al.*, 2016). The CTCF protein is an 11 zinc-fingers protein that binds CCCTC motifs (Ji *et al.*, 2016; Tang *et al.*, 2015; West *et al.*, 2002; Rao *et al.*, 2014). The binding of CTCF involves at least 4 of the 11 zinc-fingers and is to specific sequences (Renda *et al.*, 2007). It is generally thought that two CTCFs form a homodimer that can cause two distal regions of DNA to form into a loop (Ji *et al.*, 2016); though a recently published article describing the structure of CTCF bound to dsDNA noted that no dimerization was observed (Yin *et al.*, 2017). By creating a loop mediated by CTCF proteins, the promoter(s) and enhancers of a given gene, and distant associated parts of the gene on the DNA sequence, are brought in close spatial proximity (Doyle *et al.*, 2014). The CTCF binding sites also correlate with alternative splicing (Warns *et al.*, 2016) and differentially methylated domains (Lewis and Murrell, 2004).

Cohesin is also a key factor in the formation of many CTCF contacts. Cohesin is a large complex of proteins and is closely associated with the formation of the chromosomes during mitosis and meiosis and plays a general role in chromatin stability (Hirano, 2015; Nasmyth and Haering, 2009; Kudo *et al.*, 2009; Ohta *et al.*, 2011). However, cohesin also appears to play an important role in the binding of some CTCF homodimers (Ji *et al.*, 2016; Nasmyth and Haering, 2009; De Wit *et al.*, 2015; Tang *et al.*, 2015; Tark-Dame *et al.*, 2014; Downen *et al.*, 2014). The chromatin loops are often closed by a pair of CTCF proteins that preferentially adopt a parallel orientation where the CTCF binding partners point into the loop (De Wit *et al.*, 2015; Tang *et al.*, 2015; Szalaj *et al.*, 2016; Guo *et al.*, 2015) as shown schematically in Fig. 5(A). There are some indications that the CTCF tends to bend the chromatin at its binding position (Macpherson and Sadowski, 2010). Since the dimer was not observed (Yin *et al.*, 2017), dimer formation may require the presence of cohesin and possibly other factors. When the CTCF partners are pointing away from the loop in divergent loops (Fig. 5(B)), the binding is far weaker, and likewise, tandem loops (Fig. 5(C) and (D)) appear to have an intermediate frequency of occurrence (Tang *et al.*, 2015). It is currently not clear whether divergent and tandem loops also involve

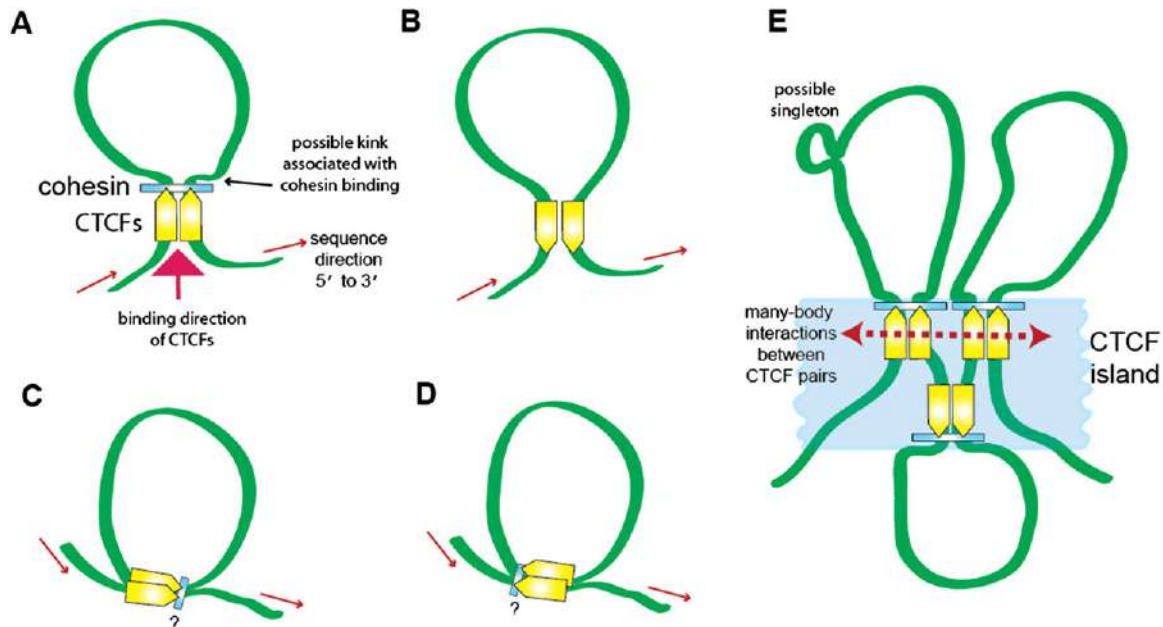


Fig. 5 Examples of various types of loops formed by CTCF dimerization (combined in some cases with cohesin) and its interactions with chromatin. (A) A convergent loop, (B) a divergent loop, (C) a tandem right loop, (D) a tandem left loop, and (E) CTCF islands composed of loops from A and, in the top left hand region of the loop, a singleton is also indicated. The 5' to 3' direction of the sense strand of the DNA sequence is indicated by the red arrows, cohesin is indicated by the blue bar that closes one end of the CTCF, and the implied sequence direction for the CTCF motifs are indicated by the pointed-end of the CTCF objects.

cohesin. Models for the interplay of CTCF and cohesin will be discussed in the last section of this review (particularly in the subsection on loop extrusion).

Multiple loops can co-localize within the same three dimensional genomic loci; the process is mediated by many CTCF dimers that form together into CTCF islands, or rafts (see Fig. 5(E)). Such proximate collections of loops are defined here as chromatin contact domains (CCDs), similar to the term topologically associating domains (TADs), which is used widely in the field of 3D genomics. Such CCDs lengths often range between several kbp to several Mbp and show considerable similarity between cells and/or stages of cell development (Dekker and Mirny, 2016; Beagan *et al.*, 2016; Ji *et al.*, 2016; Tang *et al.*, 2015; Rao *et al.*, 2014). To a lesser extent, Rpol2 also influences large regions of chromatin structure (Dunn *et al.*, 2003; He *et al.*, 2008; Jahan *et al.*, 2016; Warns *et al.*, 2016).

On the global scale of chromatin associated with this histone-dsDNA complex, the overall topology of the chromatin appears to regulate gene expression by delineating regions of transcription where active and repressed chromatin can be found because distant parts of the chromatin chain are brought into proximity of each other in loops (Ji *et al.*, 2016; Ulianov *et al.*, 2016; Tang *et al.*, 2015; Beagan *et al.*, 2016; Dekker and Mirny, 2016; Buenrostro *et al.*, 2015; Rao *et al.*, 2014). Chromosomal rearrangements and changes in the shape of the cell are also strongly correlated with malignant cells (He *et al.*, 2008; Davie, 1997; Schuster-Bockler and Lehner, 2012) suggesting that some cancers occur when normal patterns of CTCF binding begin to break down. The interplay of cohesin with enhancers, repressors, insulators and promoters and various histone modifications is very intricate (Merkenschlager and Odom, 2013) and beyond the scope of this work. Beside CTCF and cohesin, several other proteins are postulated to be involved in forming chromatin loops. They include Rpol2 mediator, Yin Yang 1 (YY1) and BORIS (Pugacheva *et al.*, 2015; Weintraub *et al.*, 2017; Kagey *et al.*, 2010).

It remains unclear how the nucleosomes are unwound and transcribed into pre-mRNA and processed by the spliceosome to produce mRNA. Moreover, although the nucleosome loci affect the type of exon/intron splicing that occurs (Warns *et al.*, 2016; Voong *et al.*, 2016; Shukla *et al.*, 2011), the details of the mechanism are still unclear. There appear to be promoter-proximal nucleosomes, where some of the positively charged nucleosome contributes to the pausing of Rpol2 (Voong *et al.*, 2016; Shukla *et al.*, 2011). How these processes are carried out remains an area of intense research.

Methods for Evaluating Chromatin Interactions

In this section, we consider both the essential experimental and computational techniques that are used to study chromatin interactions and structure. We start by reminding the reader that the field of biology is very complex and demands considerable openness to multidisciplinary approaches to interpret and understand what we observe. For example, many research papers in the field have been written by people unaware of many decades of research in polymer physics. Likewise, as we forge ahead in new

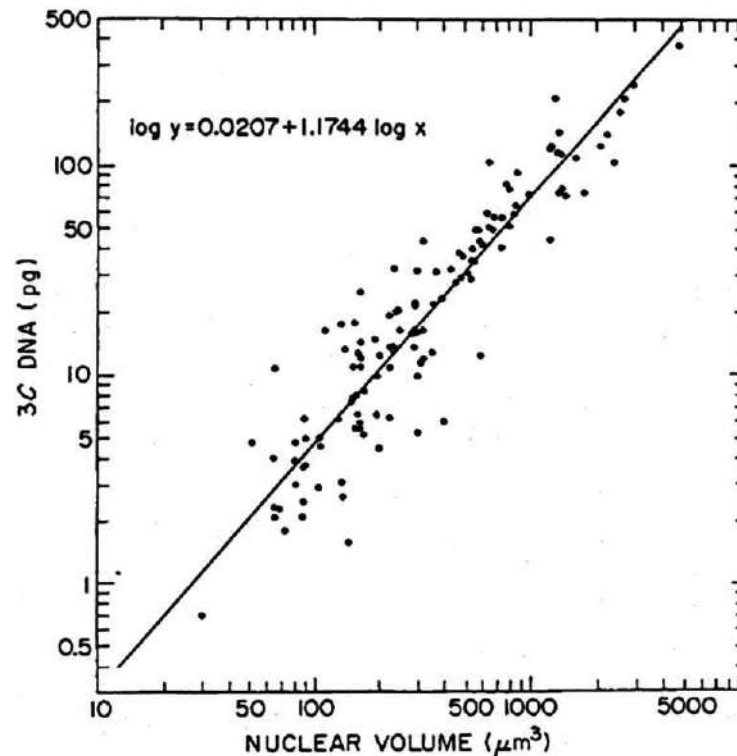


Fig. 6 Comparison of the mass of the DNA (directly proportional to the length of the sequence) with respect to the volume of the cell nucleus for plant species. (Taken from government document Nauman, A.F. 1979. Plant Nuclear Data Handbook: Radiosensitivity in Plants. Available at: <https://www.osti.gov/scitech/servlets/purl/5369196>, public domain).

techniques and are faced with the explosive sea of literature, we often think that work 50 years ago is irrelevant today. For example, in the early '60s in a somewhat obscure study (Baetcke *et al.*, 1967), it was discovered that the volume of the cell depends linearly on the length of the genome or the number of base pairs (bp) of the chromatin. Moreover, a correlation between cell volume and nuclear volume has been known since the late 19th century (explained in Gregory (2005)). Hence, to some extent, once again, we have reinvented the wheel – we should have expected this result, not been surprised by it. The important thing, however, is not just “what” but “why” we observe certain features of chromatin and recent developments in the field have helped us understand more of the “why” question and are an exciting area of new discoveries.

Experimental Methods

This section summarizes some of the most commonly used experimental techniques in studying chromatin and partly reviews some of the history. The optical techniques of fluorescence in situ hybridization (FISH) have forged ahead on discovering 3D compartment in the nucleus (Fraser *et al.*, 2015). Recently developed techniques like chromatin conformation capture (3C) and derivative technologies such as Hi-C (Rao *et al.*, 2014) and ChIA-PET (Li *et al.*, 2017) allow us to propose 2D maps of the interacting parts of chromatin. These method allow us not only to see which parts of the genome are in spatial proximity but also, in case of ChIA-PET, which proteins are responsible for maintaining the genome stability. This invaluable knowledge relies heavily on being able to propose a 3D structure of the chromatin using computational techniques. Beside the short introduction to the abovementioned methods we briefly discuss of their strengths and weaknesses.

Size of the Cell Nucleus Correlates with Chromatin Chain Length

Since the early '50s, numerous studies have shown that there is a strong correlation between the quantity of DNA in a given cell and the size of its nucleus in both the animal and plant kingdoms (Baetcke *et al.*, 1967; Price *et al.*, 1973; Gregory, 2017, 2005; Jovtchev *et al.*, 2006; Beaulieu *et al.*, 2008; Nauman, 1979). To obtain the size of the genome, most of these studies measured the mass (C-value) of the DNA (in pg). Very similar conclusions were even reached in the 19th century when the volume of a given cell was compared with the volume of its nucleus (*op. cit.*, (Gregory, 2005)). Likewise, DNA content and chromosome volume (Baetcke *et al.*, 1967) correlate. This phenomenon is also observed throughout the plant and animal kingdoms (Hodgson *et al.*, 2010; Mirsky and Ris, 1951; Beaulieu *et al.*, 2008). Furthermore, when the size of the nucleus deviates from this tendency, it is usually an indication of diseases like cancer (Webster *et al.*, 2009). In Fig. 6 we show the correlation between nuclear volume and C-value (the mass of the DNA) from Nauman (1979) for various plant species. Fig. 7(A) shows a similar fit for animal species reproduced from

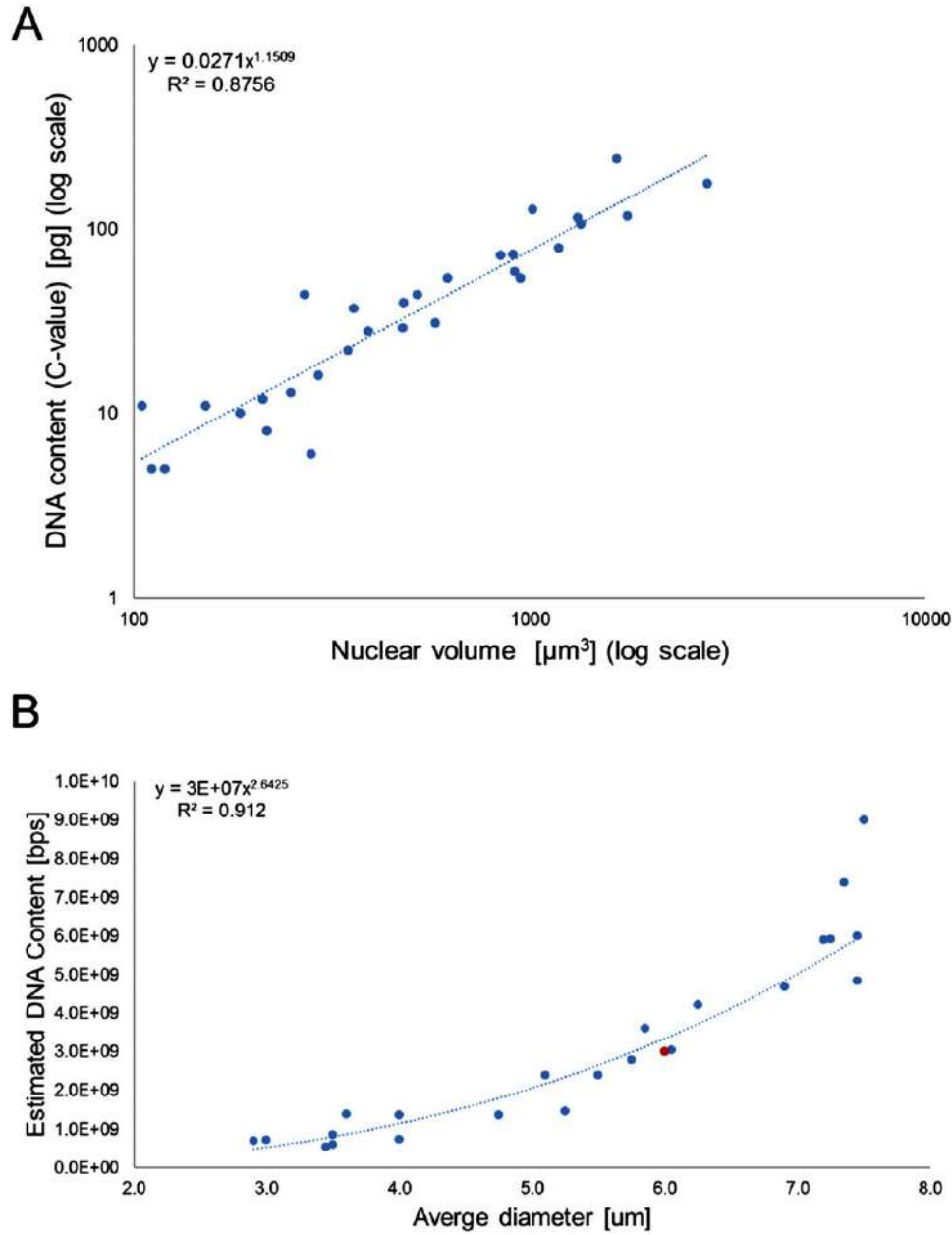


Fig. 7 Similar examples to Fig. 6 (A) A reproduction of the data from Baetcke *et al.* (1967). (B) A comparison of the quantity of nuclear DNA (measured in base pairs) with respect to average diameter of the nucleus for measured species of fish (the red dot in the figure indicates the position of the human genome and is included for reference and comparison). From data obtained from the Animal Genome Size Database. Reproduced from Gregory, T.R., 2017. Animal genome size database: Cell size dataBase. In: Gregory, T.R. (Ed.). Available at: <http://www.genomesize.com/cellsizes/>. Fig. 7(B) is plotted with linear axes for easier readability, plotted log-log, it looks similar to 5A.

Baetcke *et al.* (1967) and Fig. 7(B) shows a fit of C-values with respect to the average nuclear diameter for available information on ichthyic genomes (from Gregory (2017)) and the human genome for reference and comparison (the red dot). Whereas there is some scatter, the conclusions from the abovementioned articles are quite reproducible.

To understand and appreciate the significance of this behavior, we need to delve into some of the essential mathematical formulas. In the traditional model of an ideal polymer (Flory, 1953; Grosberg and Khokhlov, 1994), the distance between the ends of the polymer chain (r , also known as the end-to-end distance) can be shown to be a function of the square root of the number of monomers (N); $r \propto N^{1/2}$. This is the general property of the random walk (Flory, 1953; Grosberg and Khokhlov, 1994) and is known as the Gaussian polymer chain (gpc) model. The formal definition is given with Eq. (1):

$$r = b\sqrt{\xi N} \quad (1)$$

where b is the distance between adjacent monomers on the polymer chain, ξ is the Kuhn length (Kuhn, 1936) – a measure of the stiffness of the polymer – and r is measured in units monomer-monomer separation distance b . It must be noted that, in the literature, the parameter b often incorporates the Kuhn length. Thus in some cases, b would be equivalent to ξb in Eq. (1); $b \Rightarrow \xi b$.

This *ideal polymer* relationship often requires additional modification to Eq. (1) because real polymers can end up in good solvent (e.g., polystyrene in hexane) where the polymer tends to soak up the surrounding solvent and consequently swell in size. Likewise, a polymer can end up in poor solvent (e.g., polyvinylchloride in water) and become very compact because the monomers in the polymer tend to exclude solvent in favor of its own residues. This behavior is known as the excluded volume and it affects the value of the exponent in Eq. (1), which is often labeled ν ;

$$r = \xi^{1-\nu} N^\nu b \quad (2)$$

Thus for the ideal polymer where $\nu = 0.5$, we obtain a formula identical to Eq. (1); $r = (\xi N)^{0.5} b$. For non-ideal polymer ν can range from $1/3 < \nu < 2/3$.

Examining Figs. 6 and 7 in terms of volume relationships it can be seen that the typical volume of the nucleus (V_{nuc}) grows essentially linearly with genomic content: $V_{nuc} \approx \rho N$, where ρ is a proportionality constant. In Fig. 7(A), the exponent is close to 1 (when fitted to volume) and in Fig. 7(B) (when fitted to diameter) the exponent is close to 3. Although there is some scatter, the weight of the measured data suggests a strong tendency toward $V_{nuc} \propto r_{nuc}^3 \propto N$. If we use the formula for radius from Eq. (1) (i.e., $r_{nuc} = b\sqrt{\xi N}/2$) to calculate the volume of an ideal sphere we get $V_p = (4\pi/3)(r/2)^3 = (\pi b^3/6)(\xi N)^{3/2}$. We can see that the volume of an ideal polymer (V_p) should thus grow as $V_p \sim N^{3/2}$. The ratio of V_p and V_{nuc} is given in Eq. (3).

$$\frac{V_p}{V_{nuc}} \approx \frac{(\pi/6)(\xi N b^2)^{3/2}}{\rho N} = \frac{\pi \xi^{3/2} b^3 N^{1/2}}{6\rho} \propto N^{1/2} \quad (3)$$

Thus it can be clearly seen that chromatin in the nucleus must be far more compact than what can be expected for an ideal polymer. Since the size of a genome can range from 10^4 (for viruses) all the way to more than 10^{10} (for some species of plants and some animals, particularly amphibians), this compacted structure is probably essential for optimal functioning of any cell and some specific mechanism must exist that allows chromatin to remain so condensed.

In eukaryotic cells, the nucleus provides confinement for the chromatin chain. When a polymer is inserted into a capillary where the inner radius of the capillary (r_{cap}) is smaller than the size of the ideal polymer (r_p from Eq. (1); i.e., $r_{cap} < r_p$), the maximum radius of the polymer vertical to the axis of the capillary approaches the inner radius of the capillary and the polymer balls up into as many globules as required to pack the polymer into the capillary; a process known as reptation (Grosberg and Khokhlov, 1994; Thirumalai and Shi, 2017).

One might therefore imagine the chromatin fiber to behave as an ideal polymer chain embedded in a capillary, to some extent. To preserve $V_{nuc} \propto N$, one might assume that $\nu \sim 1/3$. However, the environment of the cell is not particularly ideal for DNA with its somewhat hydrophobic bases; however, its sugars are quite hydrophilic and the cellular environment is a vast heterogeneous mixture of many proteins, small molecules, water, ions and DNA/RNA. Hence, it would be better to imagine that the heterogeneous properties of the nucleus (without other considerations) should correspond to V_p , an ideal polymer, in the "polymer melt" consisting of all the different components of the cell's nucleus. Nevertheless, $\nu \sim 1/3$ would explain how a nuclear membrane could compress a range of genome sizes from 10 Mbp to 10 Gbp; i.e., it doesn't have to.

More recently, independent of the above findings, a linear relationship between nucleus volume and the amount of DNA ($V \propto N$) was observed with the first high throughput sequencing experiment using the Hi-C strategy (Lieberman-Aiden *et al.*, 2009). In that study, it was noticed that the likelihood of observing contact positions was inversely proportional to the genomic distance s between the contact points (wherein the number of contacts vanished as $1/s$ rather than $1/s^{3/2}$ (what would be expected if the chromatin behaved like an ideal polymer). The measuring of contact frequency is obviously different from the volume but whether the contacts had been random or specific; their frequency should correlate with the end-to-end distance, for a given genomic distance s . The frequency of observed contacts in any polymer should drop off as $1/s^\alpha$, where α is a constant. In the work of Lieberman-Aiden and co-workers, they proposed a fractal globule to explain this discrepancy from the behavior expected for an ideal polymer.

This again prompts the proposal that the chromatin is essentially a polymer in poor solvent and, therefore, $\nu \approx 1/3$; even though it is a heterogeneous polymer melt of diverse materials. However, studies of various models of chromatin tend to disagree with the simplistic collapsed polymer model (Rosa and Zimmer, 2014; Jerabek and Heermann, 2014). What has been discovered over the years, particularly from fluorescence in situ hybridization (FISH) experiments (Wang *et al.*, 2015; Brackley *et al.*, 2016; Giorgetti *et al.*, 2014; Bickmore and Van Steensel, 2013) and more recently experiments using Hi-C (Thurman *et al.*, 2012; Lieberman-Aiden *et al.*, 2009; Rao *et al.*, 2014; Ulianov *et al.*, 2016; Jin *et al.*, 2013; Dixon *et al.*, 2012) and ChIA-PET (Fullwood and Ruan, 2009; Tang *et al.*, 2015; Szalaj *et al.*, 2016; Li *et al.*, 2017; Ji *et al.*, 2016; Downen *et al.*, 2014) strongly suggest that the genome in the cell is highly organized and different segments of the genome are compartmentalized for protein expression and maintenance. The main counter argument about organization of the genome has been that the diffusion rates for various factors could also be roughly 10^{-6} s; hence, the promotor is just generated in large enough quantities that it can diffuse to the desired gene in reasonable time. The rates are arguably fast enough. However, it suggests that a lot of factors are generated (indeed, wasted) and the cell machinery must remove these factors. With the introduction of CTCF binding and cohesin, this argument has become increasingly moot in recent years.

In terms of structure, there is nothing in the theory of a collapsed polymer (Grosberg and Khokhlov, 1994; Mckenzie, 1976) stating that the agents associated with this collapse simply cannot be the result of protein factors located at specific locations on the dsDNA chain (such as the CTCF and cohesin, discussed in the previous section). What the theory assumes is that there are mutual weak binding interactions between the residues of the polymer. In a simple collapsed polymer model, only random contacts are observed and the dynamics of the polymer chain would quickly lead to knots. (As pointed out in the Introduction, an earphone dropped in a bag quickly knots.) This is because there are just so many degrees of freedom in a flexible cord (Mckenzie, 1976). Because the cause of the globular structure is uniform attractive self-interaction forces between the polymer itself (Mckenzie, 1976), the enormous number of degrees of freedom mean that even if the contacts have a definite organization, natural selection pressures will have forced that organization to adopt a structure around the minimum free energy that resembles one of the many random configurations that could be generated with the same free energy for a set of heterogeneous contacts. Hence, whereas in principle v represents a globular structure with uniform attractive contacts, the overall size of the organized structure with heterogeneous contacts can also follow an $r^3 \propto N$ rule. The nuclear envelope is reflecting just exactly the size of the nuclear material, the chromatin fiber. Hence, although more organized than a traditional globular structure (Grosberg and Khokhlov, 1994; Flory, 1953), the overall distribution of the contacts still resembles a collapsed polymer.

Optical Approach: FISH

Before the advent of high-throughput sequencing techniques that can directly quantify proximity ligation products in contact libraries, the commonly used method for determining nuclear organization and chromatin conformations was fluorescence in situ hybridization (FISH).

This optical technique was first developed in the '80s (Langer-Safer *et al.*, 1982) and has been used from the beginning for various cytogenetic studies. In the earliest stages, it was used to identify the presence or absence of certain DNA sequences and therefore effectively to map the location of various genes on the chromosomes. This was done by attaching a probe that carried a fluorophore to a sequence of DNA using in situ hybridization and next observing the specific part of DNA to which probed attached using fluorescence (O'Connor, 2008). Fluorescence microscopy can be used to track gene expression with probes binding to mRNA and be used to identify the relative distance between two positions on the chromosome, producing a 2D map. It can also be used to identify the overall structure of a region of the nucleus and it can be used to model dynamics.

FISH is a cytogenetic approach; i.e., a study of genetics in terms of the structure and function of the chromosomes, the structure, architecture and function of the genome. Since the observed information comes in the form of visible light (about 900 to 400 nm), this limits the direct resolution to the scale of tenths of micrometers in the best case (less than 200 nm in the xy -plane and less than 500 nm along z -axis (optical axis)). However, with knowledge of the DNA sequence and targeted placement of the fluorescence probes in the genome sequence the method permits to infer that distant regions on the linear sequence are in proximity of each other. We will not discuss super-resolution microscopy or EM microscopy, which are also rapidly advancing.

Specific examples of the interaction between the cohesin complex and the CTCFs at the level of x-ray crystallography have yet to be obtained. Therefore, we can only infer the interactions between these components by their regular association. What is more typically done is to use FISH to tag different loci of the chromatin and in this way the dynamic association of diverse parts of the chromatin chain in live cells can be visualized (Chuang *et al.*, 2006; Tumber and Belmont, 2001; Chubb *et al.*, 2002; Muller *et al.*, 2010). In such studies, it has been shown that the arrangement of the chromosomes can change from a largely peripheral location to a more internal region of the nucleus and that there is considerable compartmentalization in the architectural arrangement of the chromosomes within the nucleus; at least that there is a clear tendency for various sectors of the chromatin to appear in a similar distribution from cell to cell, though dependent upon the particular phase of the cell. It would seem reasonable that an efficient way to use the transcription production machinery is to place it in various central locations and bring the diverse parts of the chromatin to those loci rather than move the very complex spliceosomal machinery with a plethora of moving parts to specific locations in the cell. Efficient, shared usage of common but complex factory resources is used in designing assembly lines.

FISH Methods: Experimental Considerations

Whereas the broad 3D features of chromatin can be deduced by FISH experiments, to set up an experiments that identify the interactions within the chromatin on the length scales that might be considered to influence expression and transcription requires an ingenious choice of the right markers on the right genes. Only then FISH can be used to directly infer the loci between two locations in the genome and other information. One can imagine that this is very much a trial and error approach. It should be clear that this process of identifying these interactions requires considerable labor, and one is still largely limited to a resolution of maybe 0.5 μm . In the end, one will need to employ a wealth of computational tools to interpret the observed data. A different way to approach the interactions of chromatin is to map what areas of the chromatin are in contact with each other; i.e., showing the interactions between diverse parts of the chromatin chain. The number of instances of a given contact in a very large collection of the same type of cell in a similar phase of the cells is a measure of how likely a given interaction occurs. Once a 2D map is constructed, it is possible to infer the local 3D structure of the chromatin – or more correctly, the ensemble of structures that comprise the chromatin in the given region of the cell. At some point, the larger picture of the whole cell has too many degrees of freedom to be modeled solely with a mere contact map, but in terms of the many compartments within the cell nucleus, these length scales border closer to the minimum resolution of the 3D FISH experiments.

3C Methods

There are several experimental contact mapping techniques (Hakim and Misteli, 2012) based largely on the “chromosome conformation capture” (3C), which was developed by Dekker *et al.* (2002). The essence of the concept is that a population of cells is chemically fixed with formaldehyde. The formaldehyde forms cross-links between neighboring parts of the chromatin in the form of covalent bonds (Dekker *et al.*, 2002; Jackson, 1999; Orlando *et al.*, 1997). Restriction enzymes are used in the next step to digest the chromatin leaving small fragments that represent the parts of the chromatin that were in proximity of each other. After diluting the DNA, ligase is employed to link pairwise fragments (also sometimes called anchors) that are then amplified with PCR (Dekker *et al.*, 2002; Abou El Hassan and Bremner, 2009; Hagege *et al.*, 2007) and finally sequenced. During analysis, the sequences at similar locations of the genome are clustered according to the resolution; roughly on the order of the square of the number of cells analyzed. This basic protocol was often modified to improve the signal-to-noise ratio or to limit the contacts to ones resulting from mutual interactions of specific protein factors (like CTCF or Rpol2).

The term “anchor” is used to refer both to a uniquely identifiable sequence of DNA (Mckernan *et al.*, 2009) and to a pair of uniquely identifiable sequences that interact with each other in the form of loops (Fullwood and Ruan, 2009). The term “intra-ligated” indicates contacts (cross-links) that result from the coincidental interactions of the chromatin or local bending and compaction of the chromatin. Such interactions are not likely to serve any significant physiological purpose in terms of gene expression or regulation. Inter-ligated contacts are ones that result from long-distance interactions mediated by, for example, CTCFs and cohesin.

The chromosome conformation capture-on-chip (4C) technique was designed to help speed up the throughput and as a consequence to increase the resolution of the 3C (Lomvardas *et al.*, 2006; Simonis *et al.*, 2006; Wurtele and Chartrand, 2006; Zhao *et al.*, 2006). As with 3C, the interactions are cross-linked and selected restriction enzymes are used to digest the chromatin away from the contacts of interest. The method relies on generating circular ligation products between a selected fragment and the rest of the genome. The circular products are quantified using microarrays or sequencing. The advantage is that one can deduce all the interactions between a given, specified location and the rest of the genome. The chromosome conformation capture carbon copy (5C) is an extension of 3C in that, rather than the ligation products from specific restriction fragments (like 4C), all the restriction fragments are used and amplified in a microarray. The “carbon copy” is because it essentially amplifies the 3C data. The advantage is that rather than detecting the interactions of a specific fragment, it measures the relative proportions of the amplified product.

Hi-C Methods

The Hi-C method is a high resolution genomic approach that can capture chromatin loops, domains (in this case called topologically associated domains TADs), and many local features related to unspecific long-range protein-mediated chromatin interactions. The TADs have borders strongly enriched with the CTCF binding sites (Nichols and Corces, 2015; Alipour and Marko, 2012) and their sizes are typically found over length scales of 1 to 3 Mbps.

The Hi-C method essentially adds high-throughput sequencing to help build libraries that quantify the proximity ligation products. The method is able to cover the entire genome at a certain size resolution. Like 3C, the population of cells are fixed with formaldehyde and digested with different types of restriction enzymes. At this stage, the overhangs in the restriction fragments

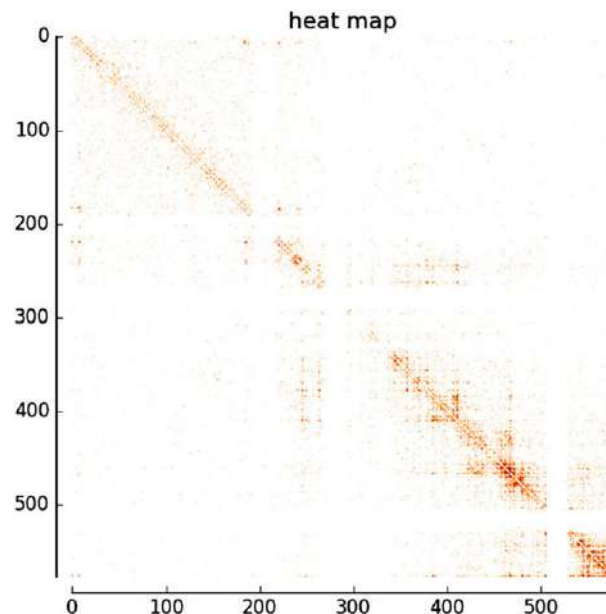


Fig. 8 Example of ChIA-PET data of the X chromosome from 150882227 to 153764804 in human lymphoblastoid cells (GM12878).

are filled with Klenow enzyme and various deoxynucleoside triphosphates (dNTPs) including biotin-14-dCTP and the products ligated and purified. The libraries are sonicated and subjected to pulldown with streptavidin-coated beads.

ChIA-PET Methods

The “chromatin interaction analysis by paired-end tag sequencing” (ChIA-PET) is an extension of Hi-C in the sense that ChIA-PET selects for ligation products of interactions of specific proteins; typically Rpo12 or CTCF (Fullwood *et al.*, 2009a; Handoko *et al.*, 2011; Tang *et al.*, 2015; Sandhu *et al.*, 2012). The paired-end tag sequence (PET) refers to a short DNA sequence that is unique enough to identify a particular segment of the genome, a sequence at least 13 bp long (Fullwood *et al.*, 2009b). Rather than applying restriction enzymes, chromatin fragments are created by sonicating the cells and selection for specific proteins is achieved by applying chromatin immunoprecipitation (ChIP); specifically targeting proteins like CTCF that are bound to the chromatin and retained in the cross-linked fragments. To the ends of the co-immunoprecipitated DNA segments, biotinylated DNA linkers are added and the DNA fragments are ligated together. This is the paired end tag (PET). Purification is carried out using streptavidin beads and paired-end sequencing. The ChIA-PET experiments yield pair-end-tag (PET) data; ligation products of different parts of the chromatin chain. Each half of the ChIA-PET products is next mapped to the reference genome. An example of this method for a part of the X chromosome from 150,882,227 to 153,764,804 from human lymphoblastoid cells (GM12878) is shown in Fig. 8.

Results of ChIA-PET/Hi-C Methods

There are several important observations to consider that result from these genome mapping experiments. First, we compare the general outcomes of 3DFISH and the more recent ligation products. Second, we examine some of the larger differences found in the ligation techniques.

3DFISH vs Ligation Products of ChIA-PET and Hi-C Methods

The ligation products of both Hi-C and ChIA-PET experiments yield both intra-ligated and inter-ligated products. With Hi-C data, any contact can be either intra-ligation or inter-ligated, irrespective of whether the interactions involve CTCF or other target proteins. On the other hand, for ChIA-PET data, the intra-ligation is limited to a range of about 8 kbp and the remaining data can all be treated as inter-ligation pairs of sequence fragments (Szalaj *et al.*, 2016). This distinction can be made because ChIA-PET data involves the targeting of ChIP products: the CTCF proteins located at chromatin binding sites or other well defined markers such as proteins associated with Rpo12. Since there is a significant hyper-exponential shift in the number of counts for data less than 8 kbp and the highly flexible chromatin chain is likely to cross within this range, the majority of contributions within 8 kbp clearly must be coming from self-ligated products. On the other hand, beyond this 8 kbp range, the behavior is quite different. The paired end tag (PET) counts or anchors can show a drastic increase in counts at specific locations on the chromatin chain. This is certainly characteristic of the expected behavior of CTCF protein-protein binding sites and similar observations are obtained from Rpo12 sites (a collection of protein subunits) using ChIA-PET. Therefore, whereas we cannot completely rule out the possibility of some small fraction of self-ligated products within the data beyond 8 kbp, such contributions are certainly small and cannot contribute significantly to the statistical weight even if given they are present.

After filtering, both Hi-C data and the singleton ChIA-PET data appear to be rather similar overall, characterized by high Pearson correlation coefficient between both heatmaps (Tang *et al.*, 2015; Szalaj *et al.*, 2016). With Hi-C, there is a mixture of ligation between diverse parts of the chromatin chain and self-ligation interactions. Therefore it is more difficult to untangle self-ligation for Hi-C data compared to ChIA-PET data.

Although the positions of the CTCF dimers is accurate to roughly the length scale of the nucleosome (roughly between 150 and 200 bps), practical considerations currently set limits on the amount of statistics that can be acquired; e.g., limits on data acquisition, storage, and processing, and limits due to time and budget constraints. Currently, these limits are perhaps on the order of 1 kbp (Tang *et al.*, 2015; Rao *et al.*, 2014) – at least five times larger than the approximate size of a nucleosome.

There remain some questions about the differences of FISH and the various 3C methods. One important issue is coverage. It remains unclear if the cross-links created during fixing the cells are uniform throughout the chromatin sample or if they favor the boundaries of the nucleus; for example (Gavrilov *et al.*, 2013; Ulianov *et al.*, 2016). So specificity and stability may be issues.

Important Considerations of Ligation 2D Mapping

A major limiting factor in ChIA-PET and Hi-C measurements is statistics. As a practical example, there are some 30 million nucleosomes (nuc) in the human genome ($2 \times 3 \times 10^9$ [bp]/200 [bp/nuc]). In a recent ChIA-PET measurement (Tang *et al.*, 2015), roughly 30.8 million reads were obtained from roughly 100 million human lymphoblastoid cells (GM12878). Hence, there would be roughly 1 count per nucleosome (on average). However, if the data is binned into 5 kbp segments (about 25 nucleosomes per bead: 5000 [bp]/200 [bp/nuc]), then this leads to an average coverage of about 25 counts per bin. For ChIA-PET data exceeding the 8 kbp boundary, the CTCF or Rpo12 only yield inter-ligated products. In other words, the short range intra-ligated contacts that generally result from local chromatin looping, would be discarded, and the random long-distance spurious intra-ligated contacts are eliminated because ChIA-PET, as already mentioned, selects only specific interactions such as CTCF and Rpo12. Clearly, single counts in this type of measurement can be spurious. This is the advantage of a thermodynamic model involving statistical weights; the most significant interaction show up constantly and the weak interactions contribute minimally to the overall structure.

With these considerations, for typical sampling of some 100 M reads, it is probably more instructive to focus on 5 kbp resolution data (5 kbp beads) because it represented the best compromise between obtaining sufficient details of the CTCF

binding sites without rarefying the intensity of individual pair contacts in the heat maps to such an extent as to render the noise from potential spurious contacts excessive. In terms of the polymer behavior of the chromatin, a 5 kbp resolution means that each 5 kbp genomic segment is represented as single bead. At this length scale, the detailed conformations of the individual nucleosomes (~ 200 bp) and mutual interactions within each segment are blurred out rendering a segment that can be treated as though it had the flexibility of a single bead on a string in the polymer chain model. Finally, a 5 kbp resolution permits detection of interligation products just at the boundary of the 8 kbp window. Unlike the data in a range less than 8 kbp, the number of PET counts (from 31 M reads) appears to be less than four for nearest interactions at 5 kbp resolution but anchors describing very distant contact points can possess hundreds of PET counts.

Measured in terms of the genomic distance (s) between bead i and bead j (for $j > i$), where $s = (j - i + 1)$, the likelihood of finding contact points on the chromatin chain between bead i and bead j (the contact probability) for Hi-C and ChIA-PET data tends to decrease as a function of the inverse of the genomic distance

$$p_c(s) \propto s^{-\alpha} \quad (4)$$

where α is a dimensionless scaling parameter. This is also known as the pair interaction frequency (PIF).

Presently, it appears that estimates of α have varied drastically (Barbieri *et al.*, 2013a,b; Chiariello *et al.*, 2014) depending on the organism (e.g., drosophila, 0.7 for active regions and 0.85 for inactive; yeast, 1.5), cell type (e.g., human embryonic stem cells, 1.6, interphase lymphoblast cell, 1.1, and metaphase HeLa cells, 0.5), and theoretical models (e.g., (Meluzzi and Arya, 2013) indicates that unrestrained fits yielded 2.23, and restrained fits 3–8). These values for the α -parameter are estimated using Hi-C data. In the latter examples, the large values for α may reflect the harmonic potential that was used. Nevertheless, predictions of $p(s)$ are not simple in general. If we presume the polymer characteristics of the chromatin as discussed in the previous section, then α should be assumed to be around 1/3 if measured in one dimension, and 1 if the probability is measured in volume. Since s (Lieberman-Aiden *et al.*) is specifically genomic distance (the number of base pairs or the linear distance covered by said base pairs), this is probably a source of confusion in the literature.

Computational Models of the Mechanism

There are a variety of computational models that have emerged in recent years based on polymer science methods that try to explain the behavior of chromatin and satisfy the volume probability measure of $\alpha = 1$. A comprehensive review of all the approaches is beyond the scope of this work. Here, we will focus on two such models that have become most popular in recent years: the loop extrusion model and the strings and binders model. We will not discuss molecular dynamics and Monte Carlo simulation techniques – such as the random walk giant-loop model, Multi-loop subcompartment (MLS) model (Munkel *et al.*, 1999), Entropy-driven chromosome organization (Rosa *et al.*, 2010) and many, many others – because these topics are covered in detail in a recent review (Rosa and Zimmer, 2014) and follow the general concepts of polymers outlined above.

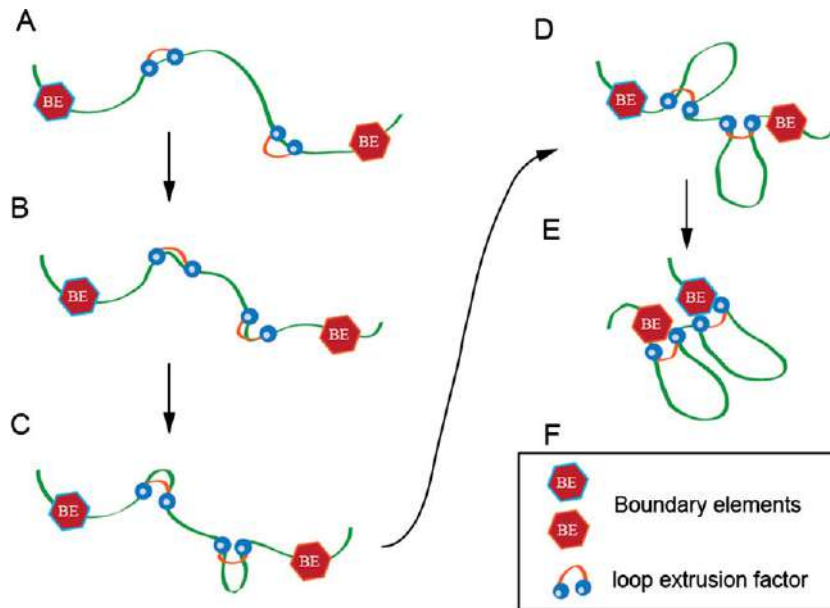


Fig. 9 The loop extrusion (LE) model. In this model, the loop extrusion factor (F) binds to the polymer chain in A, and progressively ratchets along the polymer chain one unit at a time (perhaps nucleosome by nucleosome) gradually creating a larger loop (B, C, D and E). Eventually, the loop cannot grow any larger because the loop extrusion factor collides with a boundary element (labeled BE) or it runs into an additional loop extrusion factor as shown in E.

Loop Extrusion Model

The loop extrusion (LE) model is an example how to understand the process of loop formation in chromatin features. The model was introduced with the notion that it behaves conceptually like the long-distance intracellular delivery systems like kinesin and dynein motor proteins that transport their cargoes along microtubule tracks (Hirokawa and Takemura, 2005; Vallee *et al.*, 2004).

The LE model was originally proposed in Marko and coworkers (Alipour and Marko, 2012) and further developed by Mirny and coworkers (Dekker and Mirny, 2016; Fudenberg *et al.*, 2016) and is depicted in Fig. 9. The chromatin is represented as a polymer consisting of beads separated by some specific distance; e.g., 1 kbp, 5 kbp, etc. Initially, loop extruding factors (LEF) – probably cohesin – binds to a region on the chromatin comprising neighboring beads or, possibly, a small loop formed by a few beads. Although the role of LEF was not assigned to any particular protein, it is now widely accepted that this role is realized by cohesin, a complex of several proteins (Ji *et al.*, 2016; Hirano, 2015; Nasmyth and Haering, 2009; De Wit *et al.*, 2015). Recent work also suggests that the cohesin loading complex localizes near the transcription initiation sites, thus it is plausible that the extrusion process begins in the vicinity of actively transcribed genes (Busslinger *et al.*, 2017). The LEF then begins to traverse along the chromatin with a biased rate of hopping between adjacent beads, where (for illustration) the forward rate r_+ on the right side hops faster than the reverse rate (r_-) on the left hand side. (Here we presume the beginning of the chain to be on the left). This motor behavior is known to occur with transport of cargo on microtubules (Vallee *et al.*, 2004; Hirokawa and Takemura, 2005). The extrusion process stops when cohesin reaches the insulator (also called boundary element) or another cohesin ring traversing in opposite direction (Fudenberg *et al.*, 2016). How cohesin slides along DNA is still an open question; however, it was recently postulated that this role can be fulfilled by RPol2 (Davidson *et al.*, 2016).

Strings and Binders Switch Model

The strings and binders switch (SBS) model (Barbieri *et al.*, 2012, 2013a, 2013b; Chiariello *et al.*, 2016; Bianco *et al.*, 2016) is worked out from the concepts of chemical equilibrium and the polymer physics of many body interactions containing various components. In its simplest form shown in Fig. 10, one imagines a polymer chain composed of beads (green and maroon circles)

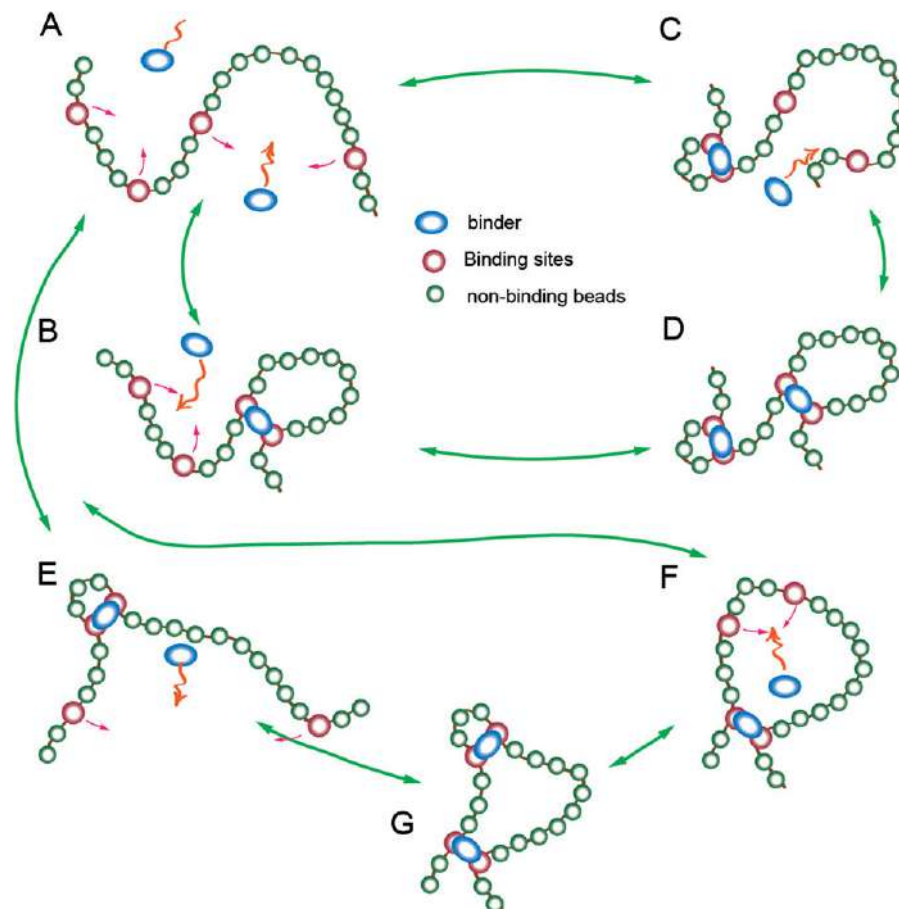


Fig. 10 The strings and binder switch (SBS) model. The green beads are effectively non-interacting, the maroon beads mark the binding sites (a CTCF-like binding sites), and the blue beads indicate the binder (cohesin-like unit). The unfolded chain is shown in A. There are two general pathways; ABCD and AEFG. These pathways are relatively mutually exclusive; however, B, C, E and F are all potential initial directions for the SBS to go and they exist in competition with one another in thermodynamic equilibrium.

where the green beads are effectively non-interacting elements and the maroon beads tend to bind with the mobile factor (represented as blue beads). The analogy is that the maroon beads would resemble something like the CTCF binding sites and cohesin, and the blue beads would represent the CTCF/cohesin binder. The system is in equilibrium, meaning that there is dynamic exchange between all the possible configurations shown in the Figure. In Fig. 10, the pathways of ABCD form one way that the strings and binders can combine; however, additional states might be also present like in pathway AEFG. Note that the arrows indicate that everything is in equilibrium. Each binding configuration resembles a switch that can be turned on and off for a given pattern.

One clear difference between the strings and binder model and the loop extrusion model is that the loop extrusion model presumes that the cohesin-like system forms an ATP-driven molecular machine that begins with binding between adjacent beads and mechanically extends in both directions until it encounters a CTCF-like obstruction. Hence, in its most rigid form, the structure in Fig. 10(G) would not form. However, in the earliest version of this model (Alipour and Marko, 2012), it was assumed that somehow the machine could skip positions. Since the precise hopping mechanism is not established, there is no reason presently to assume any restrictions per se. On the other hand, because there are many more possibilities for the dynamics of the chromatin contacts in the SBS model, the model often requires that the binding sites and binders are distinguishable and relatively mutually exclusive.

Summary

In this review, we have briefly indicated the critical features of chromatin structure, the interactions and relationships between enhancers and inhibitors, the role of CTCF proteins in forming the overall architecture of the chromatin, the volume of the nucleus using various experimental techniques, the essential methods of measuring chromatin structure and finally some examples of computational models. In the study of genomics, our understanding of how expression is regulated and how the regalia of spliceosome proteins, transcription factors, promoters and RNA polymerase II interactions will be greatly enhanced as our knowledge of chromatin binding structures and their control and regulation grows. Ultimately, it is increasingly clear that the genome is not an amorphous blob of DNA and histones randomly squeezed into a nucleus; rather, it is a well-organized system that, while possibly lacking poster-child-like publishable shapes, still exhibits a deep and highly organized meta-structural landscape.

Acknowledgments

This work was supported by grants from the Polish National Science Center (2014/15/B/ST6/05082, 2013/09/B/NZ2/00121); Foundation for Polish Science (TEAM to DP); and by grant 1U54DK107967-01 “Nucleome Positioning System for Spatiotemporal Genome Organization and Regulation” within 4DNucleome NIH program. We appreciate the assistance of Michal Kadlof in reviewing the text.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Genome Informatics. Learning Chromatin Interaction Using Hi-C Datasets. Natural Language Processing Approaches in Bioinformatics. Nucleosome Positioning. Protein Post-Translational Modification Prediction. Protein Structural Bioinformatics: An Overview. Protein-DNA Interactions. Quantitative Immunology by Data Analysis Using Mathematical Models

References

- Abou El Hassan, M., Bremner, R., 2009. A rapid simple approach to quantify chromosome conformation capture. *Nucleic Acids Res.* 37, e35.
- Alipour, E., Marko, J.F., 2012. Self-organization of domain structures by DNA-loop-extruding enzymes. *Nucleic Acids Res.* 40, 11202–11212.
- Auffinger, P., Grover, N., Westhof, E., 2011. Metal ion binding to RNA. *Met. Ions Life Sci.* 9, 1–35.
- Baetcke, K.P., Sparrow, A.H., Nauman, C.H., Schwemmer, S.S., 1967. The relationship of DNA content to nuclear and chromosome volumes and to radiosensitivity (LD50). *Proc. Natl. Acad. Sci. USA* 58, 533–540.
- Barbieri, M., Chotalia, M., Fraser, J., *et al.*, 2012. Complexity of chromatin folding is captured by the strings and binders switch model. *Proc. Natl. Acad. Sci. USA* 109, 16173–16178.
- Barbieri, M., Fraser, J., Lavitas, L.M., *et al.*, 2013a. A polymer model explains the complexity of large-scale chromatin folding. *Nucleus* 4, 267–273.
- Barbieri, M., Scialdone, A., Piccolo, A., *et al.*, 2013b. Polymer models of chromatin organization. *Front. Genet.* 4, 113.
- Bascom, G., Schlick, T., 2017. Linking chromatin fibers to gene folding by hierarchical looping. *Biophys. J.* 112, 434–445.
- Beagan, J.A., Gilgenast, T.G., Kim, J., *et al.*, 2016. Local genome topology can exhibit an incompletely rewired 3D-folding state during somatic cell reprogramming. *Cell Stem Cell* 18, 611–624.
- Beaulieu, J.M., Leitch, I.J., Patel, S., Pendharkar, A., Knight, C.A., 2008. Genome size is a strong predictor of cell size and stomatal density in angiosperms. *New Phytol.* 179, 975–986.
- Bednar, J., Horowitz, R.A., Grigoryev, S.A., *et al.*, 1998. Nucleosomes, linker DNA, and linker histone form a unique structural motif that directs the higher-order folding and compaction of chromatin. *Proc. Natl. Acad. Sci. USA* 95, 14173–14178.
- Bianco, S., Chiariello, A.M., Annunziata, C., Esposito, A., Nicodemi, M., 2016. Polymer physics of the large-scale structure of chromatin. *Methods Mol. Biol.* 1480, 201–206.

- Bickmore, W.A., Van Steensel, B., 2013. Genome architecture: Domain organization of interphase chromosomes. *Cell* 152, 1270–1284.
- Bocharova, T.N., Smirnova, E.A., Volodin, A.A., 2012. Linker histone H1 stimulates DNA strand exchange between short oligonucleotides retaining high sensitivity to heterology. *Biopolymers* 97, 229–239.
- Brackley, C.A., Brown, J.M., Waithe, D., *et al.*, 2016. Predicting the three-dimensional folding of cis-regulatory regions in mammalian genomes using bioinformatic data and polymer models. *Genome Biol.* 17, 59.
- Broggaard, K., Xi, L., Wang, J.P., Widom, J., 2012. A map of nucleosome positions in yeast at base-pair resolution. *Nature* 486, 496–501.
- Buenrostro, J.D., Wu, B., Litzenburger, U.M., *et al.*, 2015. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* 523, 486–490.
- Busslinger, G.A., Stocsits, R.R., Van Der Lelij, P., *et al.*, 2017. Cohesin is positioned in mammalian genomes by transcription, CTCF and Wapl. *Nature* 544, 503–507.
- Bustin, M., Catez, F., Lim, J.H., 2005. The dynamics of histone H1 function in chromatin. *Mol. Cell* 17, 617–620.
- Bystrycky, K., Heun, P., Gehlen, L., Langowski, J., Gasser, S.M., 2004. Long-range compaction and flexibility of interphase chromatin in budding yeast analyzed by high-resolution imaging techniques. *Proc. Natl. Acad. Sci. USA* 101, 16495–16500.
- Cheung, P., Tanner, K.G., Cheung, W.L., *et al.*, 2000. Synergistic coupling of histone H3 phosphorylation and acetylation in response to epidermal growth factor stimulation. *Mol. Cell* 5, 905–915.
- Chiariello, A., Bianco, S., Piccolo, A., *et al.*, 2014. Polymer Models of the Organization of Chromosomes in the Nucleus of Cells. Napoli, Italy: Complesso Universitario di Monte S. Angelo.
- Chiariello, A.M., Annunziatella, C., Bianco, S., Esposito, A., Nicodemi, M., 2016. Polymer physics of chromosome large-scale 3D organisation. *Sci. Rep.* 6, 29775.
- Chuang, C.H., Carpenter, A.E., Fuchsova, B., *et al.*, 2006. Long-range directional movement of an interphase chromosome site. *Curr. Biol.* 16, 825–831.
- Chubb, J.R., Boyle, S., Perry, P., Bickmore, W.A., 2002. Chromatin motion is constrained by association with nuclear compartments in human cells. *Curr. Biol.* 12, 439–445.
- Davidson, I.F., Goetz, D., Zaczek, M.P., *et al.*, 2016. Rapid movement and transcriptional re-localization of human cohesin on DNA. *EMBO J.* 35, 2671–2685. doi:10.1525/embj.201695402.
- Davie, J.R., 1997. Nuclear matrix, dynamic histone acetylation and transcriptionally active chromatin. *Mol. Biol. Rep.* 24, 197–207.
- Dawson, W., Kawai, G., 2015. A new entropy model for RNA: Part V, Incorporating the Flory-Huggins model in structure prediction and folding. *J. Nucl. Acids Investig.* 6, 1. DOI: 10.4081/jnai.2015.2657.
- Dawson, W., Yamamoto, K., Shimizu, K., Kawai, G., 2013. A new entropy model for RNA: Part II. Persistence-related entropic contributions to RNA secondary structure free energy calculations. *J. Nucl. Acids Investig.* 4, e2.
- Dekker, J., Mirny, L., 2016. The 3D genome as moderator of chromosomal communication. *Cell* 164, 1110–1121.
- Dekker, J., Rippe, K., Dekker, M., Kleckner, N., 2002. Capturing chromosome conformation. *Science* 295, 1306–1311.
- Delano, W.L., 2004. Pymol. San Carlos, California: Now Distributed by Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204. Available at: <http://www.pymol.org>.
- De Wit, E., Vos, E.S., Holwerda, S.J., *et al.*, 2015. CTCF binding polarity determines chromatin looping. *Mol. Cell* 60, 676–684.
- Dixon, J.R., Selvaraj, S., Yue, F., *et al.*, 2012. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485, 376–380.
- Down, J.M., Fan, Z.P., Hnisz, D., *et al.*, 2014. Control of cell identity genes occurs in insulated neighborhoods in mammalian chromosomes. *Cell* 159, 374–387.
- Doyle, B., Fudenberg, G., Imakaev, M., Mirny, L.A., 2014. Chromatin loops as allosteric modulators of enhancer-promoter interactions. *PLOS Comput. Biol.* 10, e1003867.
- Dunn, K.L., Zhao, H., Davie, J.R., 2003. The insulator binding protein CTCF associates with the nuclear matrix. *Exp. Cell Res.* 288, 218–223.
- Finch, J.T., Klug, A., 1976. Solenoidal model for superstructure in chromatin. *Proc. Natl. Acad. Sci. USA* 73, 1897–1901.
- Flory, P.J., 1953. Principles of Polymer Chemistry. Ithaca: Cornell University Press.
- Flory, P.J., 1969. Statistical Mechanics of Chain Molecules. New York: Wiley.
- Fraser, J., Williamson, I., Bickmore, W.A., Dostie, J., 2015. An overview of genome organization and how we got there: From FISH to Hi-C. *Microbiol. Mol. Biol. Rev.* 79, 347–372.
- Fudenberg, G., Imakaev, M., Lu, C., *et al.*, 2016. Formation of chromosomal domains by loop extrusion. *Cell Rep.* 15, 2038–2049.
- Fullwood, M.J., Liu, M.H., Pan, Y.F., *et al.*, 2009a. An oestrogen-receptor- α -bound human chromatin interactome. *Nature* 462, 58–64.
- Fullwood, M.J., Wei, C.L., Liu, E.T., Ruan, Y., 2009b. Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses. *Genome Res.* 19, 521–532.
- Fullwood, M.J., Ruan, Y.J., 2009. ChIP-based methods for the identification of long-range chromatin interactions. *J. Cell Biochem.* 107, 30–39.
- Gavrilov, A.A., Gushchanskaya, E.S., Strelkova, O., *et al.*, 2013. Disclosure of a structural milieu for the proximity ligation reveals the elusive nature of an active chromatin hub. *Nucleic Acids Res.* 41, 3563–3575.
- Giorgetti, L., Galupa, R., Nora, E.P., *et al.*, 2014. Predictive polymer modeling reveals coupled fluctuations in chromosome conformation and transcription. *Cell* 157, 950–963.
- Gregory, T.R., 2005. The C-value enigma in plants and animals: A review of parallels and an appeal for partnership. *Ann. Bot.* 95, 133–146.
- Gregory, T.R., 2017. Animal genome size database: Cell size database. In: Gregory, T.R. (Ed.), Available at: <http://www.genomesize.com/cellsizes/>.
- Grosberg, A.Y., Khokhlov, A.R., 1994. Statistical Physics of Macromolecules. New York: AIP Press.
- Guo, Y., Xu, Q., Canzio, D., *et al.*, 2015. CRISPR inversion of CTCF sites alters genome topology and enhancer/promoter function. *Cell* 162, 900–910.
- Hagege, H., Klous, P., Braem, C., *et al.*, 2007. Quantitative analysis of chromosome conformation capture assays (3C-qPCR). *Nat. Protoc.* 2, 1722–1733.
- Hagerman, P.J., 1988. Flexibility of DNA. *Annu. Rev. Biophys. Biophys. Chem.* 17, 265–286.
- Hakim, O., Misteli, T., 2012. SnapShot: Chromosome confirmation capture. *Cell* 148 (1068), e1–e2.
- Handoko, L., Xu, H., Li, G., *et al.*, 2011. CTCF-mediated functional chromatin interactome in pluripotent cells. *Nat. Genet.* 43, 630–638.
- Hansen, J.C., 2002. Conformational dynamics of the chromatin fiber in solution: Determinants, mechanisms, and functions. *Annu. Rev. Biophys. Biomol. Struct.* 31, 361–392.
- He, S., Dunn, K.L., Espino, P.S., *et al.*, 2008. Chromatin organization and nuclear microenvironments in cancer cells. *J. Cell Biochem.* 104, 2004–2015.
- Hirano, T., 2015. Chromosome dynamics during mitosis. *Cold Spring Harb. Perspect. Biol.* 7.
- Hirokawa, N., Takemura, R., 2005. Molecular motors and mechanisms of directional transport in neurons. *Nat. Rev. Neurosci.* 6, 201–214.
- Hodgson, J.G., Sharafi, M., Jalili, A., *et al.*, 2010. Stomatal vs. genome size in angiosperms: The somatic tail wagging the genomic dog? *Ann. Bot.* 105, 573–584.
- Jackson, V., 1999. Formaldehyde cross-linking for studying nucleosomal dynamics. *Methods* 17, 125–139.
- Jahan, S., Xu, W., He, S., *et al.*, 2016. The chicken erythrocyte epigenome. *Epigenet. Chromatin* 9, 19.
- Jerabek, H., Heermann, D.W., 2014. How chromatin looping and nuclear envelope attachment affect genome organization in eukaryotic cell nuclei. *Int. Rev. Cell Mol. Biol.* 307, 351–381.
- Jin, F., Li, Y., Dixon, J.R., *et al.*, 2013. A high-resolution map of the three-dimensional chromatin interactome in human cells. *Nature* 503, 290–294.
- Ji, X., Dadon, D.B., Powell, B.E., *et al.*, 2016. 3D chromosome regulatory landscape of human pluripotent cells. *Cell Stem Cell* 18, 262–275.
- Joti, Y., Hikima, T., Nishino, Y., *et al.*, 2012. Chromosomes without a 30-nm chromatin fiber. *Nucleus* 3, 404–410.
- Jovtchev, G., Schubert, V., Meister, A., Barow, M., Schubert, I., 2006. Nuclear DNA content and nuclear and cell volume are positively correlated in angiosperms. *Cytogenet. Genome Res.* 114, 77–82.
- Kagey, M.H., Newman, J.J., Bilodeau, S., *et al.*, 2010. Mediator and cohesin connect gene expression and chromatin architecture. *Nature*, 467, pp. 430–435. doi:10.1038/nature09380.
- Kalashnikova, A.A., Rogge, R.A., Hansen, J.C., 2016. Linker histone H1 and protein-protein interactions. *Biochim. Biophys. Acta* 1859, 455–461.
- Kouzarides, T., 2002. Histone methylation in transcriptional control. *Curr. Opin. Genet. Dev.* 12, 198–209.
- Kratky, O., Porod, G., 1949. Röntgenuntersuchung gelöster Fadenmoleküle. *Rec. Trav. Chim. Pays-Bas.* 68, 1106–1123.

- Kudo, N.R., Anger, M., Peters, A.H., *et al.*, 2009. Role of cleavage by separase of the Rec8 kleisin subunit of cohesin during mammalian meiosis I. *J. Cell Sci.* 122, 2686–2698.
- Kuhn, W., 1936. Beziehungen zwischen Molekulgröße, statistischer Molekulgestalt und elastischen Eigenschaften hochpolymerer Stoffe (Relations between molecular size, statistical molecular shape and elastic properties of high polymers). *Kolloidzeitschrift* 76, 258.
- Langer-Safer, P.R., Levine, M., Ward, D.C., 1982. Immunological method for mapping genes on *Drosophila* polytene chromosomes. *Proc. Natl. Acad. Sci. USA* 79, 4381–4385.
- Lewis, A., Murrell, A., 2004. Genomic imprinting: CTCF protects the boundaries. *Curr. Biol.* 14, R284–R286.
- Lieberman-Aiden, E., Van Berkum, N.L., Williams, L., *et al.*, 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 326, 289–293.
- Li, X., Luo, O.J., Wang, P., *et al.*, 2017. Long-read ChIA-PET for base-pair-resolution mapping of haplotype-specific chromatin interactions. *Nat. Protoc.* 12, 899–915.
- Lomvardas, S., Barnea, G., Pisapia, D.J., *et al.*, 2006. Interchromosomal interactions and olfactory receptor choice. *Cell* 126, 403–413.
- Macpherson, M.J., Sadowski, P.D., 2010. The CTCF insulator protein forms an unusual DNA structure. *BMC Mol. Biol.* 11, 101.
- Maeshima, K., Hihara, S., Eltsov, M., 2010. Chromatin structure: Does the 30-nm fibre exist in vivo? *Curr. Opin. Cell Biol.* 22, 291–297.
- Maeshima, K., Imai, R., Hikima, T., Joti, Y., 2014. Chromatin structure revealed by X-ray scattering analysis and computational modeling. *Methods* 70, 154–161.
- Maeshima, K., Rogge, R., Tamura, S., *et al.*, 2016. Nucleosomal arrays self-assemble into supramolecular globular structures lacking 30-nm fibers. *EMBO J.* 35, 1115–1132.
- Mckenzie, D.S., 1976. Polymers and scaling. *Phys. Rep.* 27C, 35–88.
- Mckernan, K.J., Peckham, H.E., Costa, G.L., *et al.*, 2009. Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Res.* 19, 1527–1541.
- Meluzzi, D., Arya, G., 2013. Recovering ensembles of chromatin conformations from contact probabilities. *Nucleic Acids Res.* 41, 63–75.
- Merkenschlager, M., Odom, D.T., 2013. CTCF and cohesin: Linking gene regulatory elements with their targets. *Cell* 152, 1285–1297.
- Milo, R., Jorgensen, P., Moran, U., Weber, G., Springer, M., 2010. BioNumbers – The database of key numbers in molecular and cell biology. *Nucleic Acids Res.* 38, D750–D753.
- Mirsky, A.E., Ris, H., 1951. The desoxyribonucleic acid content of animal cells and its evolutionary significance. *J. Gen. Physiol.* 34, 451–462.
- Muller, I., Boyle, S., Singer, R.H., Bickmore, W.A., Chubb, J.R., 2010. Stable morphology, but dynamic internal reorganisation, of interphase human chromosomes in living cells. *PLOS ONE* 5, e11560.
- Mumbach, M.R., Rubin, A.J., Flynn, R.A., *et al.*, 2016. HiChIP: Efficient and sensitive analysis of protein-directed genome architecture. *Nat. Methods* 13, 919–922.
- Munkel, C., Eils, R., Dietzel, S., *et al.*, 1999. Compartmentalization of interphase chromosomes observed in simulation and experiment. *J. Mol. Biol.* 285, 1053–1065.
- Nacheva, G.A., Guschin, D.Y., Preobrazhenskaya, O.V., *et al.*, 1989. Change in the pattern of histone binding to DNA upon transcriptional activation. *Cell* 58, 27–36.
- Nasmyth, K., Haering, C.H., 2009. Cohesin: Its roles and mechanisms. *Annu. Rev. Genet.* 43, 525–558.
- Nathan, D., Sterner, D.E., Berger, S.L., 2003. Histone modifications: Now summoning sumoylation. *Proc. Natl. Acad. Sci. USA* 100, 13118–13120.
- Nauman, A.F., 1979. *Plant Nuclear Data Handbook: Radiosensitivity in Plants*. Available at: <https://www.osti.gov/scitech/servlets/purl/5369196>.
- Nichols, M.H., Corces, V.G., 2015. A CTCF code for 3D genome architecture. *Cell* 162, 703–705.
- Nishino, Y., Eltsov, M., Joti, Y., *et al.*, 2012. Human mitotic chromosomes consist predominantly of irregularly folded nucleosome fibres without a 30-nm chromatin structure. *EMBO J.* 31, 1644–1653.
- O'Connor, C., 2008. Fluorescence in situ hybridization (FISH). *Nat. Educ.* 1, 171.
- Ohta, S., Wood, L., Bukowski-Wills, J.C., Rappsilber, J., Earnshaw, W.C., 2011. Building mitotic chromosomes. *Curr. Opin. Cell Biol.* 23, 114–121.
- Olson, W.K., Zhurkin, V.B., 2000. Modeling DNA deformations. *Curr. Opin. Struct. Biol.* 10, 286–297.
- Orlando, V., Strutt, H., Paro, R., 1997. Analysis of chromatin structure by in vivo formaldehyde cross-linking. *Methods* 11, 205–214.
- Ou, H.D., Phan, S., Deerinck, T.J., *et al.*, 2017. ChromEMT: Visualizing 3D chromatin structure and compaction in interphase and mitotic cells. *Science*. 357.
- Price, H.J., Sparrow, A.H., Nauman, A.F., 1973. Correlations between nuclear volume, cell volume and DNA content in meristematic cells of herbaceous angiosperms. *Experientia* 29, 1028–1029.
- Pugacheva, E.M., Rivero-Hinojosa, S., Espinoza, C.A., *et al.*, 2015. Comparative analyses of CTCF and BORIS occupancies uncover two distinct classes of CTCF binding genomic regions. *Genome Biology*, 16, p. 161. doi:10.1186/s13059-015-0736-8.
- Rao, S.S., Huntley, M.H., Durand, N.C., *et al.*, 2014. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 159, 1665–1680.
- Renda, M., Baglivo, I., Burgess-Beusse, B., *et al.*, 2007. Critical DNA binding interactions of the insulator protein CTCF: A small number of zinc fingers mediate strong binding, and a single finger-DNA interaction controls binding at imprinted loci. *J. Biol. Chem.* 282, 33336–33345.
- Rosa, A., Becker, N.B., Everaers, R., 2010. Looping probabilities in model interphase chromosomes. *Biophys. J.* 98, 2410–2419.
- Rosa, A., Zimmer, C., 2014. Computational models of large-scale genome architecture. *Int. Rev. Cell Mol. Biol.* 307, 275–349.
- Routh, A., Sandin, S., Rhodes, D., 2008. Nucleosome repeat length and linker histone stoichiometry determine chromatin fiber structure. *Proc. Natl. Acad. Sci. USA* 105, 8872–8877.
- Rychkov, G.N., Ilatovskiy, A.V., Nazarov, I.B., *et al.*, 2017. Partially assembled nucleosome structures at atomic detail. *Biophys. J.* 112, 460–472.
- Sandhu, K.S., Li, G., Poh, H.M., *et al.*, 2012. Large-scale functional organization of long-range chromatin interaction networks. *Cell Rep.* 2, 1207–1219.
- Sapojnikova, N., Thorne, A., Myers, F., Staynov, D., Crane-Robinson, C., 2009. The chromatin of active genes is not in a permanently open conformation. *J. Mol. Biol.* 386, 290–299.
- Schuster-Bockler, B., Lehner, B., 2012. Chromatin organization is a major influence on regional mutation rates in human cancer cells. *Nature* 488, 504–507.
- Shermoen, A.W., O'Farrell, P.H., 1991. Progression of the cell cycle through mitosis leads to abortion of nascent transcripts. *Cell* 67, 303–310.
- Shiio, Y., Eisenman, R.N., 2003. Histone sumoylation is associated with transcriptional repression. *Proc. Natl. Acad. Sci. USA* 100, 13225–13230.
- Shukla, S., Kavak, E., Gregory, M., *et al.*, 2011. CTCF-promoted RNA polymerase II pausing links DNA methylation to splicing. *Nature* 479, 74–79.
- Simonis, M., Klous, P., Splinter, E., *et al.*, 2006. Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture-on-chip (4C). *Nat. Genet.* 38, 1348–1354.
- Szalaj, P., Tang, Z., Michalski, P., *et al.*, 2016. An integrated 3-dimensional genome modeling engine for data-driven simulation of spatial genome organization. *Genome Res.* 26, 1697–1709.
- Tang, Z., Luo, O.J., Li, X., *et al.*, 2015. CTCF-mediated human 3D genome architecture reveals chromatin topology for transcription. *Cell* 163, 1611–1627.
- Tark-Dame, M., Jerabek, H., Manders, E.M., *et al.*, 2014. Depletion of the chromatin looping proteins CTCF and cohesin causes chromatin compaction: Insight into chromatin folding by polymer modelling. *PLOS Comput. Biol.* 10, e1003877.
- Thirumalai, D., Shi, G., 2017. Chromatin is stretched but intact when the nucleus is squeezed through constrictions. *Biophys. J.* 112, 411–412.
- Thurman, R.E., Rynes, E., Humbert, R., *et al.*, 2012. The accessible chromatin landscape of the human genome. *Nature* 489, 75–82.
- Tims, H.S., Gurunathan, K., Levitus, M., Widom, J., 2011. Dynamics of nucleosome invasion by DNA binding proteins. *J. Mol. Biol.* 411, 430–448.
- Todolli, S., Perez, P.J., Clauvelin, N., Olson, W.K., 2017. Contributions of sequence to the higher-order structures of DNA. *Biophys. J.* 112, 416–426.
- Tumbar, T., Belmont, A.S., 2001. Interphase movements of a DNA chromosome region modulated by VP16 transcriptional activator. *Nat. Cell Biol.* 3, 134–139.
- Ulianov, S.V., Khrameeva, E.E., Gavrillov, A.A., *et al.*, 2016. Active chromatin and transcription play a key role in chromosome partitioning into topologically associating domains. *Genome Res.* 26, 70–84.
- Vallee, R.B., Williams, J.C., Varma, D., Barnhart, L.E., 2004. Dynein: An ancient motor protein involved in multiple modes of transport. *J. Neurobiol.* 58, 189–200.
- Voong, L.N., Xi, L., Sebeson, A.C., *et al.*, 2016. Insights into nucleosome organization in mouse embryonic stem cells through chemical mapping. *Cell* 167, 1555–1570. (e15).

- Wang, S., Xu, J., Zeng, J., 2015. Inferential modeling of 3D chromatin structure. *Nucleic Acids Res.* 43, e54.
- Warns, J.A., Davie, J.R., Dhasarathy, A., 2016. Connecting the dots: Chromatin and alternative splicing in EMT. *Biochem. Cell Biol.* 94, 12–25.
- Webster, M., Witkin, K.L., Cohen-Fix, O., 2009. Sizing up the nucleus: Nuclear shape, size and nuclear-envelope assembly. *J. Cell Sci.* 122, 1477–1486.
- Weintraub, A.S., Li, C.H., Zamudio, A.V., *et al.*, 2017. YY1 Is a Structural Regulator of Enhancer-Promoter Loops. *Cell*, 172, pp. 1573–1588. doi:10.1016/j.stem.2015.11.007.
- West, A.G., Gaszner, M., Felsenfeld, G., 2002. Insulators: Many functions, many mechanisms. *Genes Dev.* 16, 271–288.
- Woodcock, C.L., Grigoryev, S.A., Horowitz, R.A., Whitaker, N., 1993. A chromatin folding model that incorporates linker variability generates fibers resembling the native structures. *Proc. Natl. Acad. Sci. USA* 90, 9021–9025.
- Wurtele, H., Chartrand, P., 2006. Genome-wide scanning of HoxB1-associated loci in mouse ES cells using an open-ended chromosome conformation capture methodology. *Chromosome Res.* 14, 477–495.
- Xu, C., Corces, V.G., 2016. Towards a predictive model of chromatin 3D organization. *Semin. Cell Dev. Biol.* 57, 24–30 Yin, M., Wang, J., Wang, M. *et al.*, 2017. Molecular mechanism of directional CTCF recognition of a diverse range of genomic sites. *Cell Research* 27, 1365–1377 doi:10.1038/cr.2017.131.
- Yin, M., Wang, J., Wang, M., *et al.*, 2017. Molecular mechanism of directional CTCF recognition of a diverse range of genomic sites. *Cell Research* 27, 1365–1377. doi:10.1038/cr.2017.131.
- Zhang, R., Erler, J., Langowski, J., 2017. Histone acetylation regulates chromatin accessibility: Role of H4K16 in inter-nucleosome interaction. *Biophys. J.* 112, 450–459.
- Zhao, Z., Tavoosidana, G., Sjolinder, M., *et al.*, 2006. Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions. *Nat. Genet.* 38, 1341–1347.

Nucleosome Positioning

Vladimir B Teif and Christopher T Clarkson, University of Essex, Colchester, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

The genome of a eukaryotic cell is stored inside the nucleus in the form of the nucleoprotein complex called chromatin. If one would gently remove chromatin from the human cell nucleus, the ~ 2 m long DNA would appear as a string with beads (nucleosomes). Each nucleosome consists of two copies of each of histones H2A, H2B, H3 and H4 (so called histone octamer) and about 147 DNA base pairs (bp) wrapped around these histones (Luger *et al.*, 1997). The distances between neighbouring nucleosomes can vary from zero to tens of base pairs (van Holde, 1989). Nucleosomes can be formed at any location along the DNA, but their positioning in the genome is not random – it is affected by the DNA sequence and other factors described below, which allow the same genome to be packed differently depending on the cell state. Nucleosome positioning determines the accessibility of DNA to transcription factors and other proteins, and is thus an important regulator of gene expression. Therefore, nucleosome positioning has been an active area of research since the discovery of the nucleosome (Kornberg, 1974; Olins and Olins, 1974). These investigations have further intensified in the 2000s with the developments of new methods allowing direct measurements of genome-wide nucleosome locations using high-throughput sequencing (Ioshikhes *et al.*, 2006b; Segal *et al.*, 2006b; Yuan *et al.*, 2005). Nowadays nucleosome positioning studies have advanced up to the level of human patients, down to single cells and cell-free DNA in blood (Snyder *et al.*, 2016).

Parameters Characterizing Nucleosome Positioning

The regulation of DNA accessibility by nucleosome positioning is performed through different mechanisms that can be characterised by the following four major parameters:

Nucleosome Dyad Positions

This term refers to the position of the center of the DNA segment symmetrically wrapped around the histone octamer (so called dyad), that can be directly measured *in vitro* or inferred from averaging in high-throughput sequencing experiments. Thus, the term “nucleosome positioning” can be understood either in its general sense (including all the parameters listed below), or in a narrow sense, referring to the location of the nucleosome dyads (see Fig. 1(A)).

Nucleosome Occupancy

This term refers to the probability of a given DNA site to be occupied by a histone octamer. It can be inferred from averaging over an ensemble of cells studied via Next Generation Sequencing (NGS) experiments. In such experiments the nucleosome occupancy at a given locus is reflected by the density of sequencing reads corresponding to the nucleosomal DNA, which determines the shape of the continuous nucleosome occupancy landscape (Fig. 1(B)).

Nucleosome Stability, Accessibility and Fuzziness

In the context of genomic nucleosome positioning, nucleosome stability is usually determined by comparing the nucleosome occupancy landscape of the same genomic region obtained at different levels of chromatin digestion. The better the correlations between nucleosome landscapes in different experiments the higher the nucleosome stability. Physically, the nucleosome stability is characterised by the energetic cost to partially unwrap the nucleosome (which depends on the DNA sequence, covalent modifications of DNA and histones, and the length of the unwrapped DNA part). This thermodynamic value can be measured directly in single-molecule experiments (Poirier *et al.*, 2008) or inferred indirectly from genome-wide experiments in the living cells. In the latter case the nucleosome stability is usually defined as a measure of its sensitivity to different levels of chromatin digestion (Chereji *et al.*, 2017; Mueller *et al.*, 2017; Teif *et al.*, 2014). One can similarly define the opposite parameter called accessibility, which refers to the probability to make the nucleosomal DNA accessible for protein binding (Mueller *et al.*, 2017). Another related parameter is called nucleosome fuzziness, which is proportional to the standard error of determining the nucleosome occupancy at a given genomic location based on the averaging of several replicate experiments (Vainshtein *et al.*, 2017) (Fig. 1(C)).

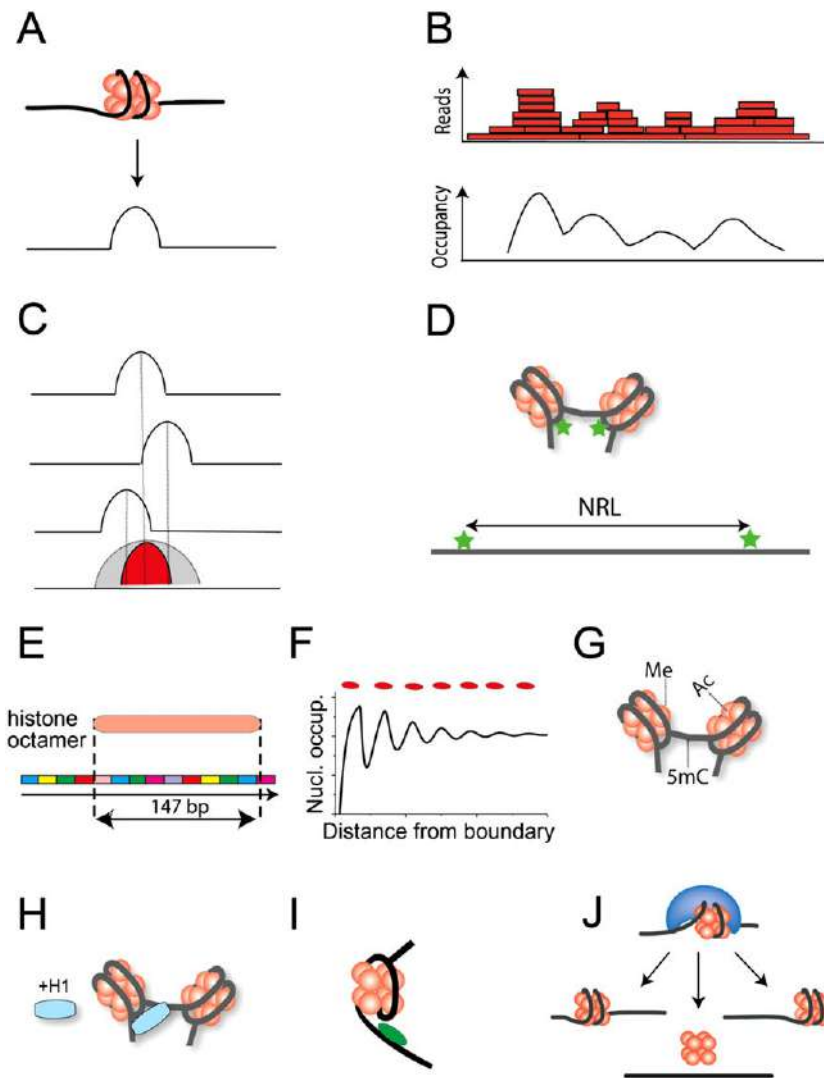


Fig. 1 Schematic representations of the main characteristics of nucleosome positioning and main factors affecting it. (A-D): Main characteristics of nucleosome positioning include the dyad position (A), nucleosome occupancy landscape (B), nucleosome stability/fuzziness (C), nucleosome repeat length (NRL) (D). (E-J): Main factors affecting nucleosome positioning include the DNA sequence affinity of the histone octamer (E), statistical positioning by the boundaries (F), covalent modifications of DNA and histones (G), interaction with linker histones (H), competition with transcription factors (I), and action of chromatin remodellers (J).

Nucleosome Repeat Length (NRL)

NRL is an integrative parameter equal to the average distance between the centres of neighbouring nucleosomes (**Fig. 1(D)**). NRL can be calculated either for large enough genomic regions or the whole genome. Genome wide average NRL depends on the cell type and state. For example, it can vary from as small as 160 bp in Yeast to up to 210 bp in human or mouse ([van Holde, 1989](#)). NRL also can change during cells differentiation ([Teif et al., 2012](#); [van Holde, 1989](#)) and can be different within the same cells in specific genomic regions, e.g., a 10-bp NRL decrease with respect to genome-average NRL has been reported near bound transcription factors CTCF or SP1 ([Teif et al., 2017](#)), and NRLs associated with transcriptionally active and inactive genomic regions are also known to be different ([Valouev et al., 2011](#)).

Genetic and Epigenetic Factors Affecting Nucleosome Positioning

Genomic nucleosome positioning is non-random, and represents a unique characteristic of a given cell state and type. Several counteracting processes affect nucleosome positioning both at the level of the genome (DNA sequence) and epigenome (beyond the DNA sequence). One can distinguish six major determinants of genomic nucleosome positioning:

- Intrinsic DNA sequence affinity of the histone octamer
- Statistical positioning of nucleosomes by genomic boundaries
- Chemical modifications of DNA or histones
- Interaction of nucleosomes with linker histones
- Binding of transcription factors and other chromatin proteins
- ATP-dependent nucleosome repositioning by chromatin remodellers

These factors will be considered in detail in the next sections.

Intrinsic DNA Sequence Affinity of the Histone Octamer

DNA has a natural affinity for histones as its backbone is constituted by negatively charged phosphate residues and can be neutralised by histones which bear positive charges. Thus, nucleosomes can form at any location, but some locations are more likely to be selected for nucleosome formation for a number of reasons. The geometry of the DNA double helix is characterised by six parameters: twist, shift, slide, roll, rise and tilt ([van Holde, 1989](#)). In the first approximation most of these parameters can be determined from the dinucleotide content neglecting the effect of longer than nearest-neighbour nucleotides either based on available crystal structures ([Olson et al., 1998](#)) or molecular dynamics simulations ([Lankas et al., 2003](#); [Lavery et al., 2010](#)). Therefore, studies of dinucleotide compositions of genomic DNA have contributed significantly to the establishment of the nucleosome positioning field. Almost 40 years ago it was noticed that genomic DNA is characterised by ~ 10 bp periodicity of dinucleotide distribution, which theoretically could play a role in histone recruitment to specific sites in the DNA ([Trifonov and Sussman, 1980](#)). Later such periodicities were indeed observed in the nucleosome core DNA sequences ([Satchwell et al., 1986](#)).

In order to find the strongest nucleosome positioning DNA sequence, *in vitro* nucleosome reconstitution via salt-dialysis was performed for a large pool of randomly generated DNA sequences ([Lowary and Widom, 1998](#)). As a result of this study the sequence named “601” (later referred to as the “Widom 601” was found to have the highest affinity for the histone octamer). Furthermore, it has been demonstrated by genome-wide sequencing that G/C rich regions have higher nucleosome density, while A/T rich regions are more nucleosome depleted ([Segal and Widom, 2009](#)). At the intermediate scale, one can distinguish longer, nucleosome-size motifs, which are either attractive for the histone octamer, such as e.g., the Widom 601 sequence ([Lowary and Widom, 1998](#)) and so called Trifonov’s “strong nucleosomes” ([Trifonov and Nibhani, 2015a](#)), or present nucleosome excluding barriers ([Drillon et al., 2016](#)). The cumulative evidence from many experiments conducted in the pre-NGS era confirmed the hypothesis that the DNA sequence at least partially affects the positions of nucleosomes. This hypothesis was further supported when first genome-wide locations of nucleosomes have been mapped ([Ioshikhes et al., 2006b](#); [Lee et al., 2007](#); [Segal et al., 2006b](#); [Yuan et al., 2005](#)). Later genome-wide studies have revealed that nucleosome positions change with cell differentiation and during the cell cycle, which has led to the refinement of the concept of DNA-sequence dependent nucleosome organization ([Schones et al., 2008](#); [Whitehouse et al., 2007](#)). It is currently believed that $\sim 9\%$ of nucleosomes are arranged consistently by the DNA sequence across different types of human cells ([Gaffney et al., 2012](#)).

Since the core histone proteins are extremely conserved among different species, the physics that determines DNA sequence preferences for the histone octamer is universal from yeast to human. However, the ensemble of nucleosomal DNA sequences can differ between species, and so are the uses of nucleosomes in gene regulation. For example, nucleosome positioning at functional genomic elements can be either favoured by default (the genomic element is always closed by the nucleosome unless active nucleosome displacement happens leading to the activation of this element), or it may be unfavourable (nucleosome depletion by default allowing TF binding to a given site unless a nucleosome is actively repositioned there). A recent study showed that the former mechanism is probably preferred in higher eukaryotes while the latter is implemented in unicellular organisms ([Tompitak et al., 2017](#)). This study reported that promoters of multicellular organisms are in general characterised by nucleosome-favouring sequences, while unicellular organisms have nucleosome-disfavouring promoters.

Statistical Positioning of Nucleosomes by Genomic Boundaries

In 1988 Kornberg and Stryer proposed that nucleosome positioning, at least to some extent, is governed by the boundary effect: nucleosomes will form ordered arrays next to any physical boundary in the genome ([Kornberg and Stryer, 1988](#)). In the post-NGS era genome-wide maps of nucleosome binding have proved this hypothesis. The average nucleosome occupancy shows regular oscillations near genomic locations that act as a boundary, such as the nucleosome depleted transcription start sites ([Mavrich et al., 2008](#)), replication origins ([Eaton et al., 2010](#)) or large DNA-bound transcription factors such as CTCF ([Cuddapah et al., 2009](#); [Fu et al., 2008](#)). Even large genomic intervals such as genes experience boundary effects by their nucleosome-depleted starts and ends ([Chevereau et al., 2009](#)). Nucleosomes are different from the analogy of “beads on the string” in that nucleosomes are not freely moving along the DNA. Indeed, nucleosome reconstitution experiments show that the nucleosome occupancy oscillations around TSS can be recapitulated *in vitro* only in the presence of ATP and energy-dependent chromatin remodellers which facilitate nucleosome movements ([Zhang et al., 2011](#)). Nevertheless, even with the non-equilibrium component introduced by chromatin remodellers, nucleosome density oscillation near genomic barriers can be quantitatively predicted from equilibrium

thermodynamics consistently with the experiments (Beshnova *et al.*, 2014; Mobius and Gerland, 2010; Riposo and Mozziconacci, 2012; Rube and Song, 2014). Interestingly, recent studies show that up to 37.5% of human nucleosome positions can be accounted for by considering boundary effects created by nucleosome-excluding sequences (Drillon *et al.*, 2016).

Chemical Modifications of DNA or Histones

Covalent modifications of DNA and histones represent a large layer of epigenetic regulation (Portela and Esteller, 2010). Most common DNA modifications are cytosine methylation and hydroxymethylation in the context of CpG dinucleotides ("CpG" means cytosine followed by phosphate followed by guanine). Histone modifications are numerous and may have combinatorial nature (several different amino acids of the same histone can be covalently modified, and different histones in the same nucleosome can carry different modifications). Epigenetic modifications are established by the "writer" enzymes, and recognized by specific "reader" proteins, which modulate the genetic program based on these modifications (Strahl and Allis, 2000). The molecular action of covalent modifications of DNA and histones is often through nucleosome repositioning. A number of statistically defined relations between nucleosome positioning and covalent chromatin modification have been reported, but there seems to be no simple code connecting DNA/histone modifications with the presence/absence of a nucleosome at a given location. For example, the action of DNA methylation is context-dependent – it is rarely seen in CpG islands and when this is observed, it is associated with recruited nucleosomes. Outside of CpG islands DNA methylation is most likely between nucleosomes (Teif *et al.*, 2014). On the other hand, DNA hydroxymethylation can decrease the nucleosome stability. A methylation/hydroxymethylation switch model has been proposed to explain some of the changes in nucleosome occupancy and stability during embryonic stem cell differentiation when an almost 10% fraction of genomic CpGs change their state from hydroxymethylation to methylation, increasing the nucleosome density of the corresponding chromatin regions (Teif *et al.*, 2014). Histone modifications also have characteristic relations with nucleosome positioning. For example, regions with "active" histone modification marks such as acetylation of histone H3 lysine 4 or 9 are characterised by lower nucleosome density, while "inactive" chromatin marks such as methylation of histone H3 lysine 9 or 27 are characterised by higher nucleosome density (Teif *et al.*, 2014).

Interaction of Nucleosomes With Linker Histones

Proteins involved in nucleosome positioning are not limited to the core histones composing the nucleosome. Nucleosome packing also critically depends on the linker histones, which belong to a class of basic nuclear proteins binding the DNA entry/exit from the nucleosome and providing a separation and interaction medium between nucleosomes in chromatin (Bednar *et al.*, 2017). It is well established that the abundance of linker histones affects the NRL (Woodcock *et al.*, 2006), which may in turn determine different types of chromatin packing (Bascom *et al.*, 2017; Routh *et al.*, 2008). Usually NRL increases with the increase of linker histone concentration in the nucleus, but it can be also affected by the competition of linker histones with other abundant chromatin proteins (Beshnova *et al.*, 2014).

Binding of Transcription Factors and Other Chromatin Proteins

Transcription factors (TFs), which bind DNA sequence-specifically, can affect nucleosome positioning in a number of ways considered below (see Fig. 1(E)–(J)).

TF Competition With Histone Octamer

The energy of complete nucleosome unwrapping or partial dissociation of the histone core from the DNA is much larger than typical energies of TF-DNA binding. Therefore, for most TFs nucleosomes can be viewed as almost immobile obstacles that do not really compete with TFs on a timescale of usual TF binding to the DNA. If TF/nucleosome arrangement would have been determined by equilibrium thermodynamics, nucleosome positioning would be determining TF binding and not the other way around (Teif and Rippe, 2010). Only few chromatin proteins such as CTCF have DNA binding energy comparable to that of the histone octamer, allowing them to act as a boundary affecting up to several nucleosomes in its vicinity (Cuddapah *et al.*, 2009; Fu *et al.*, 2008). It is important to keep in mind that the chromatin is never in a true thermodynamic equilibrium, and thus active energy-dependent processes, as well as the kinetics of binding determine the order of events in terms of the nucleosome/TF competition (Table 1).

TF Binding to DNA Partially Unwrapped From the Histone Octamer

Since the nucleosome is not a single entity, it can "breathe" by partially uncoiling the nucleosomal DNA. Transcription factors then can bind the nucleosomal DNA, depending on how far their binding site is hidden inside the nucleosome (North *et al.*, 2012). An

Table 1 Computational tools to predict nucleosome positioning (sorted alphabetically)

Description	Online/local installation	Dl(tri)nucleotide/periodicity/ specific motifs/empirical feature
FineStr : Single-base-resolution nucleosome mapping server (Trifonov, 2010; Gabdank <i>et al.</i> , 2009, 2010). The analysis is performed using the probe based on the 117-bp DNA bendability matrix derived from <i>C. elegans</i> . The authors suggested the universality of this pattern for other species. http://www.cs.bgu.ac.il/~nucleom/	+/-	+ / + / + / -
ICM Web : ICM Web allows users to assess nucleosome stability and fold any sequence of DNA into a 3D model of chromatin (Stolz and Bishop, 2010b; Sereda and Bishop, 2010). The model is displayed in the visual browser JSmol or can be downloaded. ICM takes a DNA sequence and generates (i) a nucleosome energy level diagram, (ii) coarse-grained representations of free DNA and chromatin and (iii) plots of the helical parameters (Tilt, Roll, Twist, Shift, Slide and Rise) as a function of position. http://dna.engr.atech.edu/icm-du	+/-	- / - / - / +
iNuc-PhysChem : Identifying nucleosomal or linker sequences from physicochemical properties (Chen <i>et al.</i> , 2012). The algorithm identifies nucleosomal sequences by incorporating twelve physicochemical properties defined elsewhere, such as A-phlicity, base stacking, B-DNA twist, bendability, bending stiffness, DNA denaturation energy, Z-DNA potential. The model was trained on data from <i>H. sapiens</i> , <i>C. elegans</i> and <i>D. melanogaster</i> . http://lin.uestc.edu.cn/server/iNuc-PhysChem	+ / +	+ / - / + / +
iNuc-PseKNC : A sequence-based predictor for nucleosome positioning in genomes with pseudo k-tuple nucleotide composition (Guo <i>et al.</i> , 2014). This is another software package from the developers of iNuc-PhysChem . Here, the samples of DNA sequences were formulated using six basic DNA local structural properties trained on datasets from <i>H. sapiens</i> , <i>C. elegans</i> and <i>D. melanogaster</i> . http://lin.uestc.edu.cn/server/iNuc-PseKNC	+/-	+ / - / + / +
Mapping_CC : Displays the nucleosome predictions based on the DNA dinucleotide correlation pattern. This algorithm was initially associated with one of the first high-throughput genome-wide nucleosome maps in Yeast (Ioshikhes <i>et al.</i> , 2006a). An updated version is available at http://nucbrowser.ucoz.com/load/	- / +	+ / + / - / -
MOSAICS : Methodologies for Optimization and Sampling in Computational Studies (Minary and Levitt, 2014). Perl scripts and a precompiled package to perform training-free atomistic prediction of nucleosome occupancy based on all-atom force field calculations. The effect of DNA methylation can be taken into account. http://www.cs.ox.ac.uk/mosaics/nucleosome/nucleosome.html	- / +	+ / - / - / +
NucEnerGen : Nucleosome energetics predictions based on high throughput sequencing (Locke <i>et al.</i> , 2010). It utilizes dynamic programming to calculate allowed nucleosome configurations and the Percus equation to infer sequence-dependent energies from the experimental occupancy profiles. http://nucleosome.rutgers.edu/nucenergen/	- / +	+ / + / + / -
nuMap : A web application implementing the YR and W/S schemes to predict nucleosome positioning (Alharbi <i>et al.</i> , 2014; Cui and Zhurkin, 2010, 2014). The methodology is based on the sequence-dependent anisotropic bending, which dictates how DNA is wrapped around a histone octamer. This application allows users to specify a number of options such as schemes and parameters for threading calculation and provides multiple layout formats. http://numap.rit.edu/app/dna/index.xhtml	+/-	+ / + / - / +
NuPoP : Nucleosome Positioning Prediction Engine (Xi <i>et al.</i> , 2010; Wang <i>et al.</i> , 2008). NuPoP is built upon a duration hidden Markov model, in which the linker DNA length is explicitly modeled. NuPoP outputs the Viterbi prediction, nucleosome occupancy score (from backward and forward algorithms) and nucleosome affinity score. NuPoP has three formats including a web server prediction engine, a stand-alone Fortran program, and an R package. The latter two can predict nucleosome positioning for a DNA sequence of any length. http://nucleosome.stats.northwestern.edu	+ / +	+ / + / - / -
Nu-OSCAR : Nucleosome-Occupancy Study for Cis-elements Accurate Recognition. It is devoted to identifying binding sites of known transcription factors, which further incorporates nucleosome occupancy around sites on promoter regions. The derivation of the algorithm is based on a biophysical view of interactions between protein factors and nucleosome DNA. http://bioinfo.au.tsinghua.edu.cn/nu_oscar/oscar.html	+/-	+ / + / - / -
nuScore : A nucleosome-positioning score calculator based on the DNA curvature properties (Tolstorukov <i>et al.</i> , 2008). This software allows an important type of analysis, where a user enters many sequences to calculate the average nucleosome energy profile. http://compbio.med.harvard.edu/nuScore	+/-	+ / + / - / +
N-score : MATLAB and Python codes using a wavelet analysis based model for predicting nucleosome positions from DNA sequence (Yuan and Liu, 2008). http://bcb.dfci.harvard.edu/~gcyuan/software.html	- / +	+ / + / - / -

effect called “collaborative competition” allows two TFs help each other to bind the nucleosomal DNA, because it is easier to the second TF to bind if the first TF is already bound and the DNA is already partially unwrapped from the nucleosome (Polach and Widom, 1996). This can lead to nonlinear effects of nucleosome positioning at regulatory sites such as enhancers (Mirny, 2010; Teif and Rippe, 2011).

TF-Mediated Recruitment of ATP-Dependent Enzymes

Sequence-specific binding of transcription factors can be used as a strategy to recruit to a given site an enzyme that chemically modifies DNA or histones, or actively translocates the nucleosome (Agalioti *et al.*, 2000). More details on this effect can be found in the next section.

ATP-Dependent Nucleosome Repositioning by Chromatin Remodellers

Random sliding of nucleosomes along the DNA is very slow at physiological conditions, and this effect is mostly limited to in vitro studies (Meersseman *et al.*, 1992), while nucleosome repositioning in live systems is usually an active process requiring ATP-dependent molecular motors, so called chromatin remodellers (Clapier *et al.*, 2017). For example, it has been shown that the oscillatory nucleosome density pattern near Yeast promoters can be recovered by in vitro nucleosome reconstitution on the same DNA sequences only in the presence of the cell extract and added ATP (Zhang *et al.*, 2011). In the absence of ATP the remodeller activity is abolished and nucleosome positioning tends to reflect the thermodynamically favoured pattern. Remodellers can act both in favour and against the thermodynamically preferred pattern depending on the context. A simplistic mathematical representation of remodeller rules is that a remodeller randomly binds a nucleosome, and then depending on the remodeller type, DNA sequence and chromatin context it either moves the nucleosome left/right, or evicts it completely (Teif and Rippe, 2009). Remodellers usually move nucleosomes in discrete steps such as 5 or 10 base pairs along the DNA, which is explained by the mechanics of the double helix repositioning along the nucleosome through formation of small loops relocated along the histone octamer (Clapier *et al.*, 2017). In many instances remodellers bind their target nucleosome non-randomly, and their recruitment is achieved by multiple remodeller subunits that recognize certain TFs or histone modifications (Ho and Crabtree, 2010).

Theoretical Approaches to Predict Nucleosome Positioning

Computational algorithms for predicting nucleosome positions can be roughly categorized into biophysical (taking into account physical properties of DNA and histones), bioinformatical (learning the rules of preferred nucleosome distributions without knowing details of molecular interactions), and hybrid models representing the mixture of these two approaches. A list of more than 20 web servers that offer nucleosome positioning prediction has been compiled elsewhere (Teif, 2016).

In typical bioinformatics-inspired models no assumptions are made about the underlying forces that determine nucleosome positions. Instead, features of the DNA content are analysed for available experimental nucleosome positioning datasets and it is assumed that nucleosome positioning in other systems follows the same rules. For example, the nucleosomal and linker DNA sequences can be collected from experiments in order to be able to find some common themes in these genomic features, and then analysed using e.g., wavelet analysis to detect changes in di-nucleotide frequency allowing to distinguish between linker and nucleosomal sequences (Yuan and Liu, 2008). The problem with training a model based on nucleotide content is that there are 4^{147} possible arrangements within a 147 bp DNA window, which is larger than the length of any known eukaryotic genome. Even if the model is trained on many datasets from different genomes, these still do not cover all possible nucleotide combinations that can be theoretically encountered. For this reason, models trained on one species are in general expected to underperform on other species (Kaplan *et al.*, 2009b), although few universal nucleosome sequence combinations are also expected (Trifonov and Nibhani, 2015a).

In typical biophysical models the prediction of nucleosome positioning is based on available nucleosome crystal structures to infer bending energies corresponding to all dinucleotides and then calculate the nucleosome score for each sliding window of 147 bp with a given dinucleotide distribution. In contrast to bioinformatics models which are trained on data that may or may not be representative of all genomic sequences, biophysical models are based on the assumptions about the underlying molecular dynamics of the process of nucleosome formation (Chereji and Morozov, 2015; Chevereau *et al.*, 2009; Stolz and Bishop, 2010a; Tompitak *et al.*, 2017). Most biophysical models are quite demanding in terms of the computational power if they perform realistic simulations. The power of these models is that they allow connecting intrinsic DNA sequence affinities with active repositioning scenarios such as TF/nucleosome competition and remodeller action (Teif and Rippe, 2009). Furthermore, biophysical models allow treating such basic phenomena as statistical nucleosome positioning by the boundary (Chevereau *et al.*, 2009), the exclusion of nucleosomes by their neighbours (Segal *et al.*, 2006b) and partial nucleosome unwrapping (Teif and Rippe, 2011). In principle, biophysical models allow treating any level of complexity in nucleosome positioning, provided there is a way to parameterise the models based on some experimental data. The latter is frequently becoming the bottleneck.

Future Directions

The number of computational algorithms and experimental datasets devoted to nucleosome positioning continues to increase, but the fundamental questions of how to predict nucleosome cell type/state-specific nucleosome re-positioning and the corresponding changes in gene expression are still not solved. One major complication is that the rules of chromatin remodelling are difficult to derive from experiments. Another complication is that even when the remodeller rules will be determined experimentally, it will be still a challenge to integrate them in the computational models where both equilibrium and non-equilibrium processes coexist. A third complication is the 3D genome structure – nucleosome positioning happening in the 1D is influenced by and influences the 3D chromatin packing. In order to address these challenges one can expect three major directions of the nucleosome positioning field. Firstly, new high-throughput sequencing methods will continue to emerge, with a special emphasis on single-molecule and single-patient studies. Secondly, the gap in theoretical descriptions will require developing new approaches integrating the dynamics of nucleosome positioning in the description of in vivo processes. Thirdly, new biophysical models will have to be developed to account for the interplay of the 1D and 3D nucleosome arrangements based on the experimental data on 3D chromatin packing. The latter is already a very active area of research supported by large funding initiatives (Dekker *et al.*, 2017).

Acknowledgement

This work was supported by the Wellcome Trust grant 200733/Z/16/Z.

See also: Chromatin: A Semi-Structured Polymer. Data Mining: Classification and Prediction. Genome Informatics. Learning Chromatin Interaction Using Hi-C Datasets. Natural Language Processing Approaches in Bioinformatics. Protein-DNA Interactions. Quantitative Immunology by Data Analysis Using Mathematical Models

References

- Agalioti, T., Lomvardas, S., Parekh, B., *et al.*, 2000. Ordered recruitment of chromatin modifying and general transcription factors to the IFN- β promoter. *Cell* 103 (4), 667–678.
- Alharbi, B.A., Alshammari, T.H., Felton, N.L., *et al.*, 2014. nuMap: A web platform for accurate prediction of nucleosome positioning. *Genom. Proteom. Bioinform.* 12, 249–253.
- Bascom, G.D., Kim, T., Schlick, T., 2017. Kilobase pair chromatin fiber contacts promoted by living-system-Like DNA linker length distributions and nucleosome depletion. *J. Phys. Chem. B* 121 (15), 3882–3894.
- Bednar, J., Garcia-Saez, I., Boopathi, R., *et al.*, 2017. Structure and dynamics of a 197 bp nucleosome in complex with linker histone H1. *Mol. Cell* 66 (3), 384–397. e8.
- Beshnova, D.A., Cherstvy, A.G., Vainshtein, Y., Teif, V.B., 2014. Regulation of the nucleosome repeat length in vivo by the DNA sequence, protein concentrations and long-range interactions. *PLoS Comput. Biol.* 10 (7), e1003698.
- Chen, W., Lin, H., Feng, P.M., *et al.*, 2012. iNuc-PhysChem: A sequence-based predictor for identifying nucleosomes via physicochemical properties. *PLOS ONE* 7, e47843.
- Chereji, R.V., Morozov, A.V., 2015. Functional roles of nucleosome stability and dynamics. *Brief Funct. Genom.* 14 (1), 50–60.
- Chereji, R.V., Ocampo, J., Clark, D.J., 2017. MNase-sensitive complexes in yeast: Nucleosomes and non-histone barriers. *Mol. Cell* 65 (3), 565–577. e3.
- Chevreau, G., Palmeira, L., Thermes, C., Arneodo, A., Vaillant, C., 2009. Thermodynamics of intragenic nucleosome ordering. *Phys. Rev. Lett.* 103 (18), 188103.
- Clapier, C.R., Iwasa, J., Cairns, B.R., Peterson, C.L., 2017. Mechanisms of action and regulation of ATP-dependent chromatin-remodelling complexes. *Nat. Rev. Mol. Cell Biol.* 18, 407–422.
- Cuddapah, S., Jothi, R., Schones, D.E., *et al.*, 2009. Global analysis of the insulator binding protein CTCF in chromatin barrier regions reveals demarcation of active and repressive domains. *Genome Res.* 19 (1), 24–32.
- Cui, F., Zhurkin, V.B., 2010. Structure-based analysis of DNA sequence patterns guiding nucleosome positioning in vitro. *J. Biomol. Struct. Dyn.* 27, 821–841.
- Cui, F., Zhurkin, V.B., 2014. Rotational positioning of nucleosomes facilitates selective binding of p53 to response elements associated with cell cycle arrest. *Nucleic Acids Res.* 42, 836–847.
- Dekker, J., Belmont, A.S., Guttman, M., *et al.*, 2017. The 4D nucleome project. *Nature* 549 (7671), 219–226.
- Drillon, G., Audit, B., Argoul, F., Arneodo, A., 2016. Evidence of selection for an accessible nucleosomal array in human. *BMC Genom.* 17, 526.
- Eaton, M.L., Galani, K., Kang, S., Bell, S.P., MacAlpine, D.M., 2010. Conserved nucleosome positioning defines replication origins. *Genes Dev.* 24 (8), 748–753.
- Field, Y., Kaplan, N., Fondue-Mittendorf, Y., *et al.*, 2008. Distinct modes of regulation by chromatin encoded through nucleosome positioning signals. *PLoS Comput. Biol.* 4, e1000216.
- Fu, Y., Sinha, M., Peterson, C.L., Weng, Z., 2008. The insulator binding protein CTCF positions 20 nucleosomes around its binding sites across the human genome. *PLOS Genet.* 4 (7), e1000138.
- Gabdanck, I., Barash, D., Trifonov, E.N., 2009. Nucleosome DNA bendability matrix (C. elegans). *J. Biomol. Struct. Dyn.* 26, 403–411.
- Gabdanck, I., Barash, D., Trifonov, E.N., 2010. FineStr: A web server for single-base-resolution nucleosome positioning. *Bioinformatics* 26, 845–846.
- Gaffney, D.J., McVicker, G., Pai, A.A., *et al.*, 2012. Controls of nucleosome positioning in the human genome. *PLOS Genet.* 8 (11), e1003036.
- Guo, S.H., Deng, E.Z., Xu, L.Q., *et al.*, 2014. iNuc-PseKNC: A sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics* 30, 1522–1529.
- Ho, L., Crabtree, G.R., 2010. Chromatin remodelling during development. *Nature* 463 (7280), 474–484.
- Ioshikhes, I.P., Albert, I., Zanton, S.J., *et al.*, 2006a. Nucleosome positions predicted through comparative genomics. *Nat. Genet.* 38, 1210–1215.
- Ioshikhes, I.P., Albert, I., Zanton, S.J., Pugh, B.F., 2006b. Nucleosome positions predicted through comparative genomics. *Nat. Genet.* 38 (10), 1210–1215.
- Kaplan, N., Moore, I.K., Fondue-Mittendorf, Y., *et al.*, 2009a. The DNA-encoded nucleosome organization of a eukaryotic genome. *Nature* 458, 362–366.
- Kaplan, N., Moore, I.K., Fondue-Mittendorf, Y., *et al.*, 2009b. The DNA-encoded nucleosome organization of a eukaryotic genome. *Nature* 458 (7236), 362–366.
- Kornberg, R.D., 1974. Chromatin structure: A repeating unit of histones and DNA. *Science* 184 (4139), 868–871.
- Kornberg, R.D., Stryer, L., 1988. Statistical distributions of nucleosomes: Nonrandom locations by a stochastic mechanism. *Nucleic Acids Res.* 16, 6677–6690.

- Lankas, F., Spomer, J., Langowski, J., Cheatham 3rd, T.E., 2003. DNA basepair step deformability inferred from molecular dynamics simulations. *Biophys. J.* 85 (5), 2872–2883.
- Lavery, R., Zakrzewska, K., Beveridge, D., *et al.*, 2010. A systematic molecular dynamics study of nearest-neighbor effects on base pair and base pair step conformations and fluctuations in B-DNA. *Nucleic Acids Res.* 38 (1), 299–313.
- Lee, W., Tillo, D., Bray, N., *et al.*, 2007. A high-resolution atlas of nucleosome occupancy in yeast. *Nat. Genet.* 39 (10), 1235–1244.
- Levitsky, V.G., 2004. RECON: A program for prediction of nucleosome formation potential. *Nucleic Acids Res.* 32, W346–W349.
- Levitsky, V.G., Babenko, V.N., Vershinin, A.V., 2014. The roles of the monomer length and nucleotide context of plant tandem repeats in nucleosome positioning. *J. Biomol. Struct. Dyn.* 32, 115–126.
- Levitsky, V.G., Ponomarenko, M.P., Ponomarenko, J.V., *et al.*, 1999. Nucleosomal DNA property database. *Bioinformatics* 15, 582–592.
- Locke, G., Tolkunov, D., Moqladeri, Z., *et al.*, 2010. High-throughput sequencing reveals a simple model of nucleosome energetics. *Proc. Natl. Acad. Sci. USA* 107, 20998–21003.
- Lowary, P.T., Widom, J., 1998. New DNA sequence rules for high affinity binding to histone octamer and sequence-directed nucleosome positioning. *J. Mol. Biol.* 276 (1), 19–42.
- Luger, K., Mader, A.W., Richmond, R.K., Sargent, D.F., Richmond, T.J., 1997. Crystal structure of the nucleosome core particle at 2.8 Å resolution. *Nature* 389 (6648), 251–260.
- Luyckx, P., Bajic, I.V., Khuri, S., 2006. NXSensor web tool for evaluating DNA for nucleosome exclusion sequences and accessibility to binding factors. *Nucleic Acids Res.* 34, W560–W565.
- Mavrich, T.N., Ioshikhes, I.P., Venters, B.J., *et al.*, 2008. A barrier nucleosome model for statistical positioning of nucleosomes throughout the yeast genome. *Genome Res.* 18 (7), 1073–1083.
- Meersseman, G., Pennings, S., Bradbury, E.M., 1992. Mobile nucleosomes – A general behavior. *EMBO J.* 11 (8), 2951–2959.
- Minary, P., Levitt, M., 2014. Training-free atomistic prediction of nucleosome occupancy. *Proc. Natl. Acad. Sci. USA* 111, 6293–6298.
- Mirny, L.A., 2010. Nucleosome-mediated cooperativity between transcription factors. *Proc. Natl. Acad. Sci. USA* 107 (52), 22534–22539.
- Mobius, W., Gerland, U., 2010. Quantitative test of the barrier nucleosome model for statistical positioning of nucleosomes up- and downstream of transcription start sites. *PLOS Comput. Biol.* 6 (8), e1000891.
- Mueller, B., Mieczkowski, J., Kundu, S., *et al.*, 2017. Widespread changes in nucleosome accessibility without changes in nucleosome occupancy during a rapid transcriptional induction. *Genes Dev.* 31 (5), 451–462.
- Nibhani, R., Trifonov, E.N., 2015. TA-periodic (“601”-like) centromeric nucleosomes of *A. thaliana*. *J. Biomol. Struct. Dyn.* 1–4.
- Nikolaou, C., Althammer, S., Beato, M., *et al.*, 2010. Structural constraints revealed in consistent nucleosome positions in the genome of *S. cerevisiae*. *Epigenetics Chromatin* 3, 20.
- North, J.A., Shimko, J.C., Javadi, S., *et al.*, 2012. Regulation of the nucleosome unwrapping rate controls DNA accessibility. *Nucleic Acids Res.* 40 (20), 10215–10227.
- Olins, A.L., Olins, D.E., 1974. Spheroid chromatin units (v bodies). *Science* 183 (4122), 330–332.
- Olson, W.K., Gorin, A.A., Lu, X.J., Hock, L.M., Zhurkin, V.B., 1998. DNA sequence-dependent deformability deduced from protein-DNA crystal complexes. *Proc. Natl. Acad. Sci. USA* 95 (19), 11163–11168.
- Poirier, M.G., Bussiek, M., Langowski, J., Widom, J., 2008. Spontaneous access to DNA target sites in folded chromatin fibers. *J. Mol. Biol.* 379 (4), 772–786.
- Polach, K.J., Widom, J., 1996. A model for the cooperative binding of eukaryotic regulatory proteins to nucleosomal target sites. *J. Mol. Biol.* 258 (5), 800–812.
- Portela, A., Esteller, M., 2010. Epigenetic modifications and human disease. *Nat. Biotechnol.* 28 (10), 1057–1068.
- Riposo, J., Mozziconacci, J., 2012. Nucleosome positioning and nucleosome stacking: Two faces of the same coin. *Mol. Biosyst.* 8, 1172–1178.
- Routh, A., Sandin, S., Rhodes, D., 2008. Nucleosome repeat length and linker histone stoichiometry determine chromatin fiber structure. *Proc. Natl. Acad. Sci. USA* 105 (26), 8872–8877.
- Rube, T.H., Song, J.S., 2014. Quantifying the role of steric constraints in nucleosome positioning. *Nucleic Acids Res.* 42 (4), 2147–2158.
- Salihi, B., Tripathi, V., Trifonov, E.N., 2015. Visible periodicity of strong nucleosome DNA sequences. *J. Biomol. Struct. Dyn.* 33, 1–9.
- Satchwell, S.C., Drew, H.R., Travers, A.A., 1986. Sequence periodicities in chicken nucleosome core DNA. *J. Mol. Biol.* 191, 659–675.
- Schones, D.E., Cui, K., Cuddapah, S., *et al.*, 2008. Dynamic regulation of nucleosome positioning in the human genome. *Cell* 132 (5), 887–898.
- Segal, E., Fondutse-Mittendorf, Y., Chen, L., *et al.*, 2006a. A genomic code for nucleosome positioning. *Nature* 442, 772–778.
- Segal, E., Fondutse-Mittendorf, Y., Chen, L., *et al.*, 2006b. A genomic code for nucleosome positioning. *Nature* 442 (7104), 772–778.
- Segal, E., Widom, J., 2009. Poly(dA:dT) tracts: Major determinants of nucleosome organization. *Curr. Opin. Struct. Biol.* 19 (1), 65–71.
- Sereda, Y.V., Bishop, T.C., 2010. Evaluation of elastic rod models with long range interactions for predicting nucleosome stability. *J. Biomol. Struct. Dyn.* 27, 867–887.
- Snyder, M.W., Kircher, M., Hill, A.J., Daza, R.M., Shendure, J., 2016. Cell-free DNA comprises an in vivo nucleosome footprint that informs its tissues-of-origin. *Cell* 164 (1–2), 57–68.
- Stolz, R.C., Bishop, T.C., 2010a. ICM Web: The interactive chromatin modeling web server. *Nucleic Acids Res.* 38 (Web Server issue), W254–W261.
- Stolz, R.C., Bishop, T.C., 2010b. ICM Web: The interactive chromatin modeling web server. *Nucleic Acids Res.* 38, W254–W261.
- Strahl, B.D., Allis, C.D., 2000. The language of covalent histone modifications. *Nature* 403 (6765), 41–45.
- Teif, V.B., 2016. Nucleosome positioning: Resources and tools online. *Brief Bioinform.* 17 (5), 745–757.
- Teif, V.B., Beshnova, D.A., Vainshtein, Y., *et al.*, 2014. Nucleosome repositioning links DNA (de)methylation and differential CTCF binding during stem cell development. *Genome Res.* 24 (8), 1285–1295.
- Teif, V.B., Mallm, J.-P., Sharma, T., *et al.*, 2017. Nucleosome repositioning during differentiation of a human myeloid leukemia cell line. *Nucleus* 8 (2), 188–204.
- Teif, V.B., Rippe, K., 2009. Predicting nucleosome positions on the DNA: Combining intrinsic sequence preferences and remodeler activities. *Nucleic Acids Res.* 37 (17), 5641–5655.
- Teif, V.B., Rippe, K., 2010. Statistical-mechanical lattice models for protein-DNA binding in chromatin. *J. Phys. Condens. Matter* 22 (41), 414105.
- Teif, V.B., Rippe, K., 2011. Nucleosome mediated crosstalk between transcription factors at eukaryotic enhancers. *Phys. Biol.* 8, 044001.
- Teif, V.B., Vainshtein, Y., Caudron-Herger, M., *et al.*, 2012. Genome-wide nucleosome positioning during embryonic stem cell development. *Nat. Struct. Mol. Biol.* 19 (11), 1185–1192.
- Tolstorukov, M.Y., Choudhary, V., Olson, W.K., *et al.*, 2008. nuScore: A web-interface for nucleosome positioning predictions. *Bioinformatics* 24, 1456–1458.
- Tompitak, M., Vaillant, C., Schiessel, H., 2017. Genomes of multicellular organisms have evolved to attract nucleosomes to promoter regions. *Biophys. J.* 112 (3), 505–511.
- Trifonov, E.N., 2010. Base pair stacking in nucleosome DNA and bendability sequence pattern. *J. Theor. Biol.* 263, 337–339.
- Trifonov, E.N., Nibhani, R., 2015a. Review fifteen years of search for strong nucleosomes. *Biopolymers* 103 (8), 432–437.
- Trifonov, E.N., Nibhani, R., 2015b. Review fifteen years of search for strong nucleosomes. *Biopolymers* 103, 432–437.
- Trifonov, E.N., Sussman, J.L., 1980. The pitch of chromatin DNA is reflected in its nucleotide sequence. *Proc. Natl. Acad. Sci. USA* 77 (7), 3816–3820.
- Vainshtein, Y., Rippe, K., Teif, V.B., 2017. NucTools: Analysis of chromatin feature occupancy profiles from high-throughput sequencing data. *BMC Genom.* 18 (1), 158.
- Valouev, A., Johnson, S.M., Boyd, S.D., *et al.*, 2011. Determinants of nucleosome organization in primary human cells. *Nature* 474 (7352), 516–520.
- van der Heijden, T., van Vugt, J.J., Logie, C., *et al.*, 2012. Sequence-based prediction of single nucleosome positioning and genome-wide nucleosome occupancy. *Proc. Natl. Acad. Sci. USA* 109, E2514–E2522.
- van Holde, K.E., 1989. *Chromatin*. New York: Springer-Verlag.
- Wang, J.P., Fondutse-Mittendorf, Y., Xi, L., *et al.*, 2008. Preferentially quantized linker DNA lengths in *Saccharomyces cerevisiae*. *PLOS Comput. Biol.* 4, e1000175.
- Whitehouse, I., Rando, O.J., Delrow, J., Tsukiyama, T., 2007. Chromatin remodelling at promoters suppresses antisense transcription. *Nature* 450 (7172), 1031–1035.
- Woodcock, C.L., Skoultschi, A.I., Fan, Y., 2006. Role of linker histone in chromatin structure and function: H1 stoichiometry and nucleosome repeat length. *Chromosome Res.* 14 (1), 17–25.
- Xi, L., Fondutse-Mittendorf, Y., Xia, L., *et al.*, 2010. Predicting nucleosome positioning using a duration Hidden Markov Model. *BMC Bioinform.* 11, 346.
- Yuan, G.-C., Liu, J.S., 2008. Genomic sequence is highly predictive of local nucleosome depletion. *PLOS Comp. Biol.* 4, e13.

Yuan, G.C., Liu, Y.J., Dion, M.F., *et al.*, 2005. Genome-scale identification of nucleosome positions in *S. cerevisiae*. *Science* 309 (5734), 626–630.

Zhang, Z., Wippo, C.J., Wal, M., *et al.*, 2011. A packing mechanism for nucleosome organization reconstituted across a eukaryotic genome. *Science* 332 (6032), 977–980.

Further Reading

Arneodo, A., Drillon, G., Argoul, F., Audit, B., 2017. The role of nucleosome positioning in genome function and evolution. In: Lavelle, C., Victor, J.M. (Eds.), *Nuclear Architecture and Dynamics*. Academic Press, Elsevier.

Blossey, R., 2017. Chromatin: Structure, Dynamics, Regulation. Chapman & Hall/CRC.

Clapier, C.R., Iwasa, J., Cairns, B.R., Peterson, C.L., 2017. Mechanisms of action and regulation of ATP-dependent chromatin-remodelling complexes. *Nat. Rev. Mol. Cell Biol.* 18, 407–422.

Hughes, A.L., Rando, O.J., 2014. Mechanisms underlying nucleosome positioning in vivo. *Annu. Rev. Biophys.* 43, 41–63.

Jiang, C., Pugh, B.F., 2009. Nucleosome positioning and gene regulation: Advances through genomics. *Nat. Rev. Genet.* 10 (3), 161–172.

Lai, W.K.M., Pugh, B.F., 2017. Understanding nucleosome dynamics and their links to gene expression and DNA replication. *Nat. Rev. Mol. Cell Biol.* (advance online publication).

Längst, G., Teif, V.B., Rippe, K., 2011. Chromatin remodeling and nucleosome positioning. In: Rippe, K. (Ed.), *Genome Organization and Function in the Cell Nucleus*. Weinheim: Wiley-VCH Verlag GmbH & Co. KGaA, pp. 111–138.

Struhl, K., Segal, E., 2013. Determinants of nucleosome positioning. *Nat. Struct. Mol. Biol.* 20 (3), 267–273.

Teif, V.B., 2016. Nucleosome positioning: Resources and tools online. *Brief Bioinform.* 17 (5), 745–757.

Trifonov, E.N., Nibhani, R., 2015. Review fifteen years of search for strong nucleosomes. *Biopolymers* 103 (8), 432–437.

Relevant Website

<http://generegulation.info/index.php/nucleosome-positioning>
Catalog of nucleosome positioning resources.

Biographical Sketch



Vladimir Teif is a Lecturer in the University of Essex. Previously he has been working in the German Cancer Research Center, University of California in San Diego, Hebrew University in Jerusalem, CEA/Saclay and Belarus National Academy of Sciences. His current interests include modelling of epigenetic regulation in chromatin, broadly defined.



Christopher Clarkson is a PhD student in the University of Essex. Previously he did MSc in Bioinformatics in the Imperial College London. Current interests include chromatin topology and prediction of genetic pathways.

Learning Chromatin Interaction Using Hi-C Datasets

Wing-Kin Sung, National University of Singapore, Singapore and Genome Institute of Singapore, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

Our genome consists of a set of chromosomes where each chromosome is a linear sequence of four nucleotides A, C, G and T. This linear model organizes the genes in functional units that help to understand many biological processes and diseases. For example, the histone genes are mostly clustered in [Albig and Doenecke \(1997\)](#). We also know that genes in the same pathways are more likely to be close proximity in our genome ([Lee and Sonnhammer, 2003](#)). For disease, Down syndrome is caused by the amplification of chromosome 21. Charcot-Marie-Tooth disease type I and Hereditary neuropathy with pressure palsies are caused by duplication and deletion of [Roa et al. \(1993\)](#). Although the linear model is useful for studying our genome, the linear model is insufficient to explain every biological processes. For instance, we cannot explain how enhancers control the expressions of their target genes which can be millions base pairs away or even in different chromosomes.

One possible explanation is that our genome folds into a 3 dimensional structure. The folding brings the enhancers close to the target genes to control their expressions. It also can explain why cells of different tissue types have different expression profiles. Although the genome in different tissue cells are the same, the 3 dimensional structures of our genome are different in different tissues ([Parada et al., 2004](#)). The tissue-specific spatial organization of our genome may change the spatial distance between the enhancers and their target genes, that finally create the tissue-specific expression profile ([Krijger and de Laat, 2016](#)). Hence, decoding the 3D structure of our genome is important to learn the mystery of our life and understand diseases.

To understand the 3D organization of our genome, a few biotechnologies are developed. They can be classified into two approaches: Fluorescence in situ hybridization (FISH) and Chromosome Conformation Capture (3C).

FISH experiments use fluorescence probes to examine the co-localization of different DNA fragments with microscopies. It allows us to measure the 3D distance between two genomic loci. By the distance, we determine if these two genomic loci are interacting. However, since FISH is based on fluorescence probes, we can only measure the 3D distance of the genomic loci for a few loci. It is difficult to expand this approach to the genome-wide scale.

Chromosome Conformation Capture (3C) was proposed by [Dekker et al. \(2002\)](#). Instead of observing the 3D distance of the interacting loci, the 3C method digests the genomic DNA with enzyme, like HindIII, followed by ligation. If two different loci are spatially close, the corresponding DNA fragments have higher chance to be ligated. Using specific PCR primer pair, the interaction between the two loci can be captured. Unlike FISH, 3C cannot tell the interaction of two loci in the single cell level and the 3D distance between them. It only tells the average contact frequency.

Moreover, it is possible to enhance the 3C method to detect the genomic interactions of multiple loci. Many 3C-based methods are developed, they include: 4C (Circular Chromosome Conformation Capture ([Hakim and Misteli, 2012](#)) or Chromosome Conformation Capture-on-Chip ([Simonis et al., 2006](#))), 5C (Chromosome Conformation Capture Carbon Copy method ([Dostie et al., 2006](#))), Hi-C ([Lieberman-Aiden et al., 2009](#)) and ChIA-PET ([Fullwood et al., 2009](#)). 4C methods are used to identify the long chromatin interactions from one locus to all other loci as “one-to-all” approach. 5C method is used to identify the chromatin interactions between many loci specified by primers as “many-to-many” approach. By taking advantage of high-throughput sequencing, Hi-C and ChIA-PET were introduced in 2009 to call genome-wide chromatin interaction.

However, the raw sequencing datasets from Hi-C and ChIA-PET do not provide direct evidences of chromatin interactions. Computational methods are required to extract the interactions among loci and determine the three dimensional structure of our genome. This review aims to review the computational methods for analyzing Hi-C datasets.

The manuscript is organized as follows. We first review the experimental method for Hi-C. Then, we give an overview of the analysis pipeline for Hi-C, which consists of 7 parts. Finally, we detail the currently available tools for these 7 steps.

Review of Hi-C Experiment

Hi-C is proposed by [Lieberman-Aiden et al. \(2009\)](#). It aims to recover the genome-wide chromatin interactions. Precisely, for every two loci in the genome that are spatially close, HiC experiment will generate a chimeric DNA fragment formed by the ligation of the two DNA fragments in these two loci. The steps of Hi-C experiment are as follows. First, cells are cross-linked with formaldehyde. This ensures the interacting DNA fragment pairs are covalently linked. Then, the chromatin is digested with restriction enzyme (say, HindIII). The sticky ends are filled in with biotinylated nucleotides. Ligation is performed under extremely dilute conditions to create chimeric molecules that link the two interacting DNA fragments. These chimeric molecules represent chromatin interactions. Then, these resulting DNA fragments are purified and sheared. By pulling down the biotinylated junctions, we obtain a set of DNA fragments that are formed by connecting two interacting DNA fragments. These fragments are called Hi-C ligation products. Through high-throughput sequencing, Hi-C paired-end reads are obtained from the two ends of the Hi-C ligation products.

Overview of Hi-C Analysis Pipeline

The Hi-C experiment generates paired-end reads which provide information on chromatin interaction. The aim of Hi-C analysis is to decrypt such interactions. Hi-C analysis involves the following steps:

- (1) *Mapping*: Given a Hi-C library, this step aligns the Hi-C reads on the reference genome.
- (2) *Read-level filtering*: Some aligned read pairs may not be coming from Hi-C ligation products. This step aims to filter the noise and retain only valid Hi-C read pairs.
- (3) *Normalization*: Due to experimental noises and mapping biases, systematic errors may occur. This step aims to correct the Hi-C signal.
- (4) *Finding local structures*: Our genome is folded into multiple local structures like compartments and TADs. This step aims to call these local structures.
- (5) *Calling mid-range or long-range chromatin interactions*: Locally, the genome folds into local structure. Globally, the loci are interacted through mid-range or long-range chromatin interactions. This step aims to call these global interactions.
- (6) *Visualization*: A large number of local structures, mid-range and long-range chromatin interactions are predicted in the previous steps. To help users to study these results, a number of tools are developed to visualize those results.
- (7) *3D structure prediction*: The Hi-C experiment provides the pairwise interaction information. However, it does not explicitly report the actual chromatin structure. This step aims to predict the 3D structure based on the information provided by Hi-C.

Below, we will detail tools for these 7 steps.

Hi-C Paired Read Mapping

Hi-C paired reads are extracted from the two ends of Hi-C ligation products, where each Hi-C ligation product is formed by ligating two genomic fragments crossing a ligation junction. Depending on the location of the ligation junction, there are 5 cases as shown in Fig. 1. Mapping of such Hi-C read pair is complicated by the location of the ligation junction.

Early methods assume that, for most cases, the ligation junctions do not occur near to the ends of the Hi-C ligation products (like the form of Fig. 1(a)). Then, simple approaches (like (Yaffe and Tanay, 2011)) use the first 50bp of each read for mapping. They assume the first 50bp does not cross the ligation junction.

HiClib (also called ICE) (Imakaev et al., 2012) is the first ligation junction-aware method. It first aligns the 25bp of each read in the HiC read pair. If it is aligned uniquely, we accept the alignment. Otherwise, extends by 5bp and realign again until the read is uniquely aligned.

HiClib is slow since it requires to extend the alignment 5bp by 5bp. Later, chimeric alignment algorithms are proposed. They include: HIPPIE (Hwang et al., 2015), HiCCUPS (Rao et al., 2014), diffHic and HiC-Pro (Servant et al., 2015). These methods run STAR, BWA or Bowtie2 to align the read pair first. If both reads align uniquely, the alignment is done. Otherwise, one of the reads is soft-clipped. These methods trim the reads and realign the clipped portion with the potential to identify the ligation junction.

Read-Level Filtering

After read mapping, the uniquely mapped paired-end reads are subjected to read-level filtering. The read-level filtering removes paired-end reads that are not coming from Hi-C ligation products. These noisy paired-end reads include:

- (1) Self-ligation read pairs: Self-ligation product is formed by a DNA fragment that is circularized and self-ligated. The corresponding paired-end read forms an outward alignment and has short insert size.
- (2) Dangling read pairs: Dangling product is a normal DNA fragment with no ligation. The corresponding paired-end read forms an inward alignment and has short insert size.
- (3) Redundant read pairs: When multiple paired-end reads map to exactly the same genome location, these are redundant read pairs generated by PCR amplification.
- (4) Random breaking read pairs: A random breaking read pair Paired alignments where both ends are far from restriction enzyme cut sites:

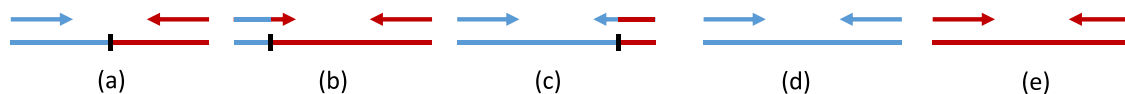


Fig. 1 Possible forms of Hi-C ligation product formed by ligating the blue and red DNA fragments. Depending on the location of the ligation junction, there are 5 forms. (a) is the common form where the ligation junction is in the middle, and is not covered by the sequenced reads, (b) is the form where the ligation junction is near to the left end and covered by one of the reads in a pair, (c) is the form where the ligation junction is near to the right end and covered by one of the reads in a pair, (d) and (e) are the form where there is no ligation.

In this step, the alignment of every read pair is examined. If the read pair is classified as self-ligation read pair, dangling read pair, redundant read pair, or random breaking read pair, it is filtered.

The read-level filtering is implemented in a number of software. They include: HICUP (Wingett *et al.*, 2015), HiC-inspector, HiC-Box, HiCdat, HiCLib (Imakaev *et al.*, 2012), HiC-Pro (Servant *et al.*, 2015), HIPPIE (Hwang *et al.*, 2015).

Hi-C Data Normalization

Due to biases in the wet-lab steps and mapping, the Hi-C paired reads are also biasedly mapped on the genome. One (arguably the most) important step is to normalize the Hi-C signal. There are two approaches: (1) explicitly model the biases and (2) matrix balancing approach.

Yaffe and Tanay (2011) proposed Hicpipe to explicitly model the biases. For any two fragment ends a and b , they observed three types of biases:

- (1) Mappability: For any fragment end x , the mappability $M(x)$ is defined to be the proportion of loci in $[x-500, x+500]$ that can be uniquely aligned. Yaffe and Tanay observed that the number of interaction reads between two fragment ends a and b is proportional to $M(a)M(b)$.
- (2) GC content of the fragment ends: Yaffe and Tanay observed that GC content around the fragment ends affects the ligation efficiency.
- (3) Fragment length: Restriction fragment length is the length of the fragment after cutting the genome with the restriction enzyme. Yaffe and Tanay observed that restriction fragment length affects the ligation efficiency.

By analyzing these three biases, Yaffe and Tanay (2011) proposed a model to compute the by-chance probability of the interaction of two fragment ends. Normalized contact map is computed by dividing the observed number of contacts between 1-Mb chromosomal bins by the expected number of contacts predicted by the model.

HiCNorm (Hu *et al.*, 2012) is another method for normalizing the contact map. It also aims to remove the above three biases. HiCNorm uses a generalized linear regression-based method. HiCNorm directly models the contact frequency between two bins using Poisson or negative binomial distribution, which simplify the set of parameters. Hence, HiCNorm is computationally more efficient. A few pipelines use HiCNorm for normalization. They include: HiCdat (Schmid *et al.*, 2015), HIPPIE (Hwang *et al.*, 2015).

Apart from explicitly models the biases, another approach is called matrix-balancing approach. The matrix-balancing approach does not assume the knowledge of the sources of biases. It assumes that, in the “true” model, every locus has the same number of interactions with other genomic regions. This is known as the equal visibility assumption.

Under the equal visibility assumption, the simplest solution is to use vanilla coverage (Lieberman-Aiden *et al.*, 2009). Consider any two bins i and j , suppose the observed number of paired end reads between two bins i and j is $O(i,j)$. Then, the number of paired end reads with one end in bin i is $O(i) = \sum_j O(i,j)$. The normalized count between two bins i and j is $N(i,j) = \frac{O(i,j)}{O(i)O(j)}$.

Vanilla coverage is not good enough, since the noise in the original interaction frequency will affect the normalized count significantly. Later, ICE (Imakaev *et al.*, 2012) is proposed. ICE is an iterated version of Vanilla coverage. It repeatedly runs Vanilla coverage until the signal is converged. A few methods use ICE for normalization. They include: HiCdat (Schmid *et al.*, 2015), Fit-Hi-C (Ay *et al.*, 2014), HiC-Pro (Servant *et al.*, 2015), HOMER (Heinz *et al.*, 2010).

Local Structure Calling

The 3 dimensional structure of our genome has two levels of structures: local and global structures. Locally, our genome is folded into multiple hierarchical of local structures, from compartments (Lieberman-Aiden *et al.*, 2009) to TADs (Dixon *et al.*, 2012) and nested sub-TAD (Rao *et al.*, 2014). Then, those local structures interact and form the global structure. Here, we detail the local structure calling methods.

Compartments partition the genome into regions with high frequency of interactions and regions with low frequency of interactions. The initial approach (Lieberman-Aiden *et al.*, 2009; Imakaev *et al.*, 2012) uses principal component analysis (PCA) to identify them. Given the normalized contact probability T_{ij} for every pair of bins i and j , $T_{ij} = \sum_{k=1}^N \lambda_k \cdot E_i^k \cdot E_j^k + \bar{T}$, where $\bar{T}$ is the mean value of T , λ_k is the Eigen value and E^k is the Eigen vector. It is observed that the first Eigen vector divides the genome into compartments with high and low frequency of interactions (Sometimes, the first Eigen vector represents the short and long chromosome arms. In this case, the compartments are represented by the second Eigen vector.) Compartments typically are of size several megabases. Compartments with high frequencies of interactions correspond to Euchromatin region with high density of genes while compartments with low frequency of interactions correspond to heterochromatin and is largely made up of gene deserts.

Using Hi-C data, Dixon *et al.* (2012) visualized 2D-interaction matrices using a variety of bin sizes. They noticed that at bin sizes less than 100kb, highly self-interacting regions begin to emerge (triangles in the heatmap, see Fig. 2). These regions are termed as “Topologically-Associated Domain” (TAD). Many methods are proposed to call TADs. They include: DomainCaller(DI/HMM) (Dixon *et al.*, 2012), Insulation score (Crane *et al.*, 2015), Armatus (Filippova *et al.*, 2014), HiCseg (Levy-Leduc *et al.*, 2014), DomainCaller (Dixon *et al.*, 2012), TopDom (Shin *et al.*, 2016), IC-Finder (Haddad *et al.*, 2017), TADbit (Serra *et al.*, 2017) and HiFive (Sauria *et al.*, 2015).

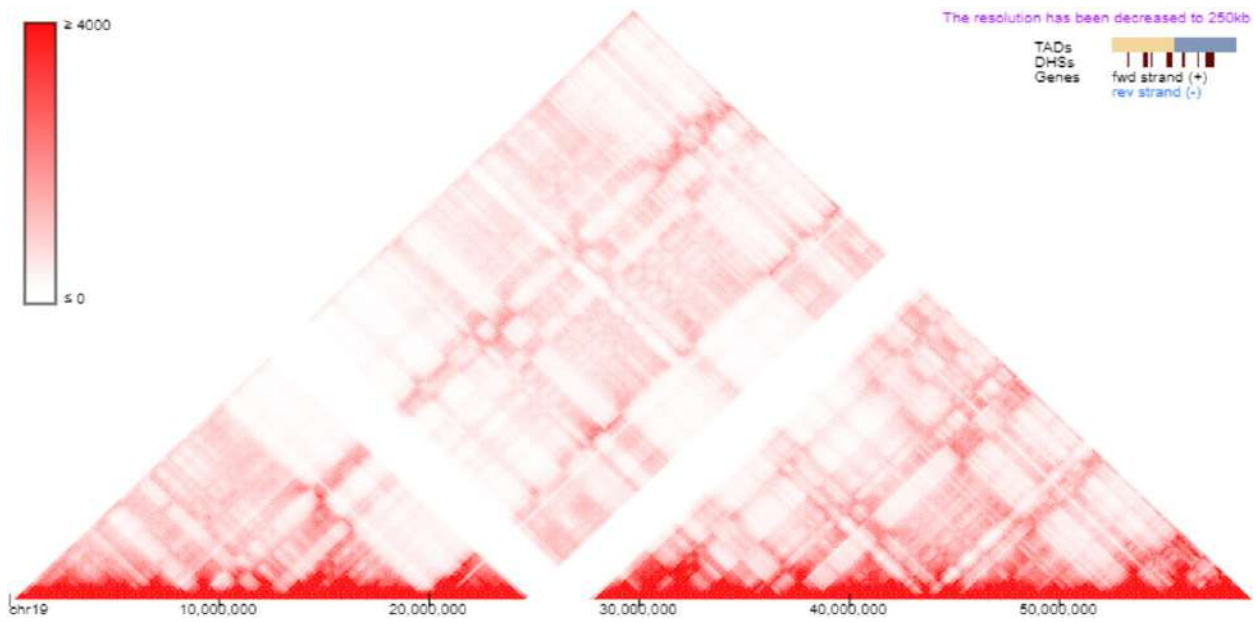


Fig. 2 This is the visualization of the 2D-interaction matrices of chr19 using bins of size 25 k. A number of self-interacting regions (as visualized as triangles) are observed. (The figure is generated using 3D Genome Browser. Reproduced from Wang, Y., Zhang, B., Zhang, L., *et al.*, 2017b. The 3d genome browser: A web-based browser for visualizing 3d genome organization and long-range chromatin interactions. bioRxiv).

Rao *et al.* (2014) observed that within a TAD, it contains some smaller structures called subTADs. Juicer (Arrowhead) (Rao *et al.*, 2014), TADtree (Weinreb and Raphael, 2016), CaTCH (Zhan *et al.*, 2017), HBM (Shavit *et al.*, 2016), and HiTAD (Wang *et al.*, 2017a) are developed to predict these subTADs.

Recently, Forcato *et al.* (2017) and Dali and Blanchette (2017) have assessed the performance of these methods. Since different TAD prediction tools are based on different assumptions, they report different sets of TADs. In the future, more works are needed to improve the predictions of TADs and subTADs.

The above methods predict TADs using Hi-C data only. There are methods that predict TADs using other omics datasets. One example is HubPredictor (Huang *et al.*, 2015), which predicts TADs using both Hi-C and histone mark ChIP-seq data.

Calling Mid-Range or Long-Range Chromatin Interactions

Our genome locally forms structures like compartments, TADs and subTADs. Globally, the loci are interacting through mid-range or long-range chromatin interactions. A number of methods have been proposed to call these global interactions.

Duan *et al.* (2010). gives the early method to call global interactions. A background expected contact frequency is learnt for each genomic distance. When the actual contact frequency between two loci is higher than expected, these two loci are deemed to be interacting.

After that, a number of papers are proposed for calling global interactions. The major improvement is on the modeling of the background contact frequency. AFC (Jin *et al.*, 2013) models the background contact frequencies as a negative binomial distribution and a global background model was devised that consists of both systematic bias factors and the linear genomic distance factor. Fit-Hi-C (Ay *et al.*, 2014) uses a non-parametric spline approach to model the background-chromatin contact frequency. The above two methods have many false positives. GOTHIC (Mifsud *et al.*, 2017) stated that the background contact frequency also depends on the read coverage. It improves the global background interaction frequency of two loci to depend also on the relative genome-wide coverage.

All above methods assume the chromatin interaction frequencies for different loci pairs are independent. Instead of assuming a global background, HiCCUPS (Rao *et al.*, 2014) uses a combination of local and global background to model the contact frequency. HMRF (Xu *et al.*, 2016a) proposed to use a hidden Markov random field based Bayesian method to explicitly model the spatial dependency among adjacent loci. HMRF is slow. The same authors proposed FastHiC (Xu *et al.*, 2016b), which uses a simulated field approximation to approximate the joint distribution of the hidden peak status by a set of independent random variables, leading to more tractable computation.

Visualization

After the Hi-C data is processed, local structures like TADs and subTADs, mid-range and long-range chromatin interactions are predicted. To help the users, it is good to visualize them. A number of software is available for Hi-C data visualization. First, we can

use traditional genome browser to visualize the Hi-C data. These browsers include: USCS genome browser, Ensembl, Integrated genome browser (IGB), Integrated genome viewer (IGV), etc.

Some specialized visualization tools for HiC data are also available. They include: WashU Epigenome Browser (Zhou *et al.*, 2013), CytoHiC (Shavit and Lio, 2013), HiCPlotter (Akdemir and Chin, 2015), QuIN (Thibodeau *et al.*, 2016), 3D Genome Browser (Wang *et al.*, 2017b). WashU Epigenome Browser is modified that allows us to view long-range contact data. CytoHiC is a Cytoscape plugin that allow users to view and compare spatial maps of genomic landmarks, based on normalized Hi-C datasets. HiCPlotter and 3D Genome Browser are simple tools that visualize Hi-C data as a Hi-C matrix. QuIN (Thibodeau *et al.*, 2016) represents the Hi-C interactions as a network. Precisely, the genomic loci are represented as vertices while the interactions between two genomic loci are represented as edges.

3D Modeling

The above steps extract interactions among pairs of loci. They did not recover the actual three dimensional structure of the genome. This step aims to reconstruct the three dimensional chromatin structure from the pairwise interactions predicted from Hi-C data. Given the contact frequencies among the genomic loci, most existing methods run in two steps. The first step is to convert the contact frequency between any two loci into their spatial distance. Then, the second step learns the 3D structure from the spatial distance matrix. There are two approaches for the second step: Consensus approach and ensemble approach.

Consensus approach uses the spatial distance (or the contact frequency directly) as the constraint, then the 3D structure is reconstructed. This approach is first used by Duan *et al.* (2010). Then, a number of methods are developed using the same approach. They include: (Tanizawa *et al.*, 2010), ChromSDE (Zhang *et al.*, 2013), ShRec3D (Lesne *et al.*, 2014), 3D-GNOME (Szalaj *et al.*, 2016), PASTIS (Varoquaux *et al.*, 2014), and HSA (Zou *et al.*, 2016).

Another approach is ensemble approach. Observe that the chromatin interactions extracted from the Hi-C experiment are actually the interactions from a pool of millions of cells. The conformations in different cells are different. The ensemble approach aims to produce a set of 3D models which can fit the observed contact frequencies. There are two strategies. The first strategy identifies multiple independent models that can fit the Hi-C data. Hence, this strategy can identify multiple models instead of just one local optimal model. Trieu and Cheng (Trieu and Cheng, 2014), MOGEN (Trieu and Cheng, 2016), LorDG (Trieu and Cheng, 2017), AutoChrom3D (Peng *et al.*, 2013), (Bau and Marti-Renom, 2012) (and TADbit (Serra *et al.*, 2017)), MCMC5C (Rousseau *et al.*, 2011) use this strategy.

Another strategy finds multiple models that can fit the Hi-C data in a coordinated manner. BACH-MIX (Hu *et al.*, 2013) uses MCMC to find a mixture of models that can fit the Hi-C data. InfMod3DGen (Wang *et al.*, 2015) uses EM algorithm to infer an ensemble of models that best fit the data. Other methods include Kalhor *et al.* (2011)

Summary

Understanding the 3 dimensional organization of the genome is an important topic. We started to understand that our genome is not simply organized linearly. Instead, our genome is folded into local structures called TADs; then, the local structures are interacting through mid-range and long-range interactions. Researchers found that the 3 dimensional structure of our genome can help to explain different biological processes like the enhancer and promoter interaction and the tissue-specific expression profiles.

In the last decades, many tools have been developed to recover the three dimensional structure of our genome. Although those tools provide useful information, the predictions are still far from accurate. In the future, more works need to be done.

See also: Chromatin: A Semi-Structured Polymer. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Genome Informatics. Natural Language Processing Approaches in Bioinformatics. Nucleosome Positioning. Protein-DNA Interactions. Quantitative Immunology by Data Analysis Using Mathematical Models

References

- Akdemir, K.C., Chin, L., 2015. Hicplotter integrates genomic data with interaction matrices. *Genome Biology* 16, 198.
- Albig, W., Doenecke, D., 1997. The human histone gene cluster at the D6S105 locus. *Hum. Genet.* 101, 284–294.
- Ay, F., Bailey, T.L., Noble, W.S., 2014. Statistical confidence estimation for hi-c data reveals regulatory chromatin contacts. *Genome Research* 24, 999–1011.
- Bau, D., Marti-Renom, M.A., 2012. Genome structure determination via 3c-based data integration by the integrative modeling platform. *Methods (San Diego, CA)* 58, 300–306.
- Crane, E., Bian, Q., McCord, R.P., *et al.*, 2015. Condensin-driven remodelling of X chromosome topology during dosage compensation. *Nature* 523, 240–244.
- Dali, R., Blanchette, M., 2017. A critical assessment of topologically associating domain prediction tools. *Nucleic Acids Research* 45, 2994–3005.
- Dekker, J., Rippe, K., Dekker, M., Kleckner, N., 2002. Capturing chromosome conformation. *Science (New York, N.Y.)* 295, 1306–1311.
- Dixon, J.R., Selvaraj, S., Yue, F., *et al.*, 2012. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485, 376–380.
- Dostie, J., Richmond, T.A., Arnaout, R.A., *et al.*, 2006. Chromosome conformation capture carbon copy (5c): A massively parallel solution for mapping interactions between genomic elements. *Genome Research* 16, 1299–1309.
- Duan, Z., Andronescu, M., Schutz, K., *et al.*, 2010. A three-dimensional model of the yeast genome. *Nature* 465, 363–367.
- Filippova, D., Patro, R., Duggal, G., Kingsford, C., 2014. Identification of alternative topological domains in chromatin. *Algorithms for Molecular Biology* 9, 14.

- Forcato, M., Nicoletti, C., Pal, K., 2017. Carmen Maria Livi, Francesco Ferrari, and Silvio Bicciato. Comparison of computational methods for hi-c data analysis. *Nature Methods* 14, 679–685.
- Fullwood, M.J., Liu, M.H., Pan, Y.F., *et al.*, 2009. An oestrogen-receptor-alpha-bound human chromatin interactome. *Nature* 462, 58–64.
- Haddad, N., Vaillant, C., Jost, D., 2017. Ic-finder: Inferring robustly the hierarchical organization of chromatin folding. *Nucleic Acids Research* 45, e81.
- Hakim, O., Misteli, T., 2012. Snapshot: Chromosome conformation capture. *Cell* 148, 1068.e1–1068.e2.
- Heinz, S., Benner, C., Spann, N., *et al.*, 2010. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular Cell* 38, 576–589.
- Huang, J., Marco, E., Pinello, L., Yuan, G.-C., 2015. Predicting chromatin organization using histone marks. *Genome Biology* 16, 162.
- Hu, M., Deng, K., Qin, Z., *et al.*, 2013. Bayesian inference of spatial organizations of chromosomes. *PLOS Computational Biology* 9, e1002893.
- Hu, M., Deng, K., Selvaraj, S., *et al.*, 2012. Hicnorm: Removing biases in hi-c data via poisson regression. *Bioinformatics (Oxford)* 28, 3131–3133.
- Hwang, Y.-C., Lin, C.-F., Valladares, O., *et al.*, 2015. Hippie: A high-throughput identification pipeline for promoter interacting enhancer elements. *Bioinformatics (Oxford)* 31, 1290–1292.
- Imakaev, M., Fudenberg, G., McCord, R.P., *et al.*, 2012. Iterative correction of hi-c data reveals hallmarks of chromosome organization. *Nature Methods* 9, 999–1003.
- Jin, F., Li, Y., Dixon, J.R., *et al.*, 2013. A high-resolution map of the three-dimensional chromatin interactome in human cells. *Nature* 503, 290–294.
- Kalhor, R., Tjong, H., Jayatilaka, N., Alber, F., Chen, L., 2011. Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. *Nature Biotechnology* 30, 90–98.
- Krijger, P.H.L., de Laat, W., 2016. Regulation of disease-associated gene expression in the 3d genome. *Nature Reviews. Molecular Cell Biology* 17, 771–782.
- Lee, J.M., Sonhammer, E.L.L., 2003. Genomic gene clustering analysis of pathways in eukaryotes. *Genome Research* 13, 875–882.
- Lesne, A., Riposo, J., Roger, P., Courmac, A., Mozziconacci, J., 2014. 3d genome reconstruction from chromosomal contacts. *Nature Methods* 11, 1141–1143.
- Levy-Leduc, C., Delattre, M., Mary-Huard, T., Robin, S., 2014. Two-dimensional segmentation for analyzing hi-c data. *Bioinformatics (Oxford)* 30, i386–i392.
- Lieberman-Aiden, E., van Berkum, N.L., Williams, L., *et al.*, 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science (New York, N.Y.)* 326, 289–293.
- Mifsud, B., Martincorena, I., Darbo, E., *et al.*, 2017. Gothic, a probabilistic model to resolve complex biases and to identify real interactions in hi-c data. *PLOS ONE* 12, e0174744.
- Parada, L.A., McQueen, P.G., Misteli, T., 2004. Tissue-specific spatial organization of genomes. *Genome Biology* 5, R44.
- Peng, C., Fu, L.-Y., Dong, P.-F., *et al.*, 2013. The sequencing bias relaxed characteristics of hi-c derived data and implications for chromatin 3d modeling. *Nucleic Acids Research* 41, e183.
- Rao, B.B., Garcia, C.A., Suter, U., *et al.*, 1993. Charcot-Marie-Tooth Disease Type 1A – Association with a Spontaneous Point Mutation in the PMP22 Gene. *N Engl J Med.* 329, 96–101.
- Rao, S.S.P., Huntley, M.H., Durand, N.C., *et al.*, 2014. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 159, 1665–1680.
- Rousseau, M., Fraser, J., Ferraiuolo, M.A., Dostie, J., Blanchette, M., 2011. Three-dimensional modeling of chromatin structure from interaction frequency data using markov chain monte carlo sampling. *BMC Bioinformatics* 12, 414.
- Sauria, M.E., Phillips-Cremins, J.E., Corces, V.G., Taylor, J., 2015. Hifive: A tool suite for easy and efficient hic and 5c data analysis. *Genome Biology* 16, 237.
- Schmid, M.W., Grob, S., Grossniklaus, U., 2015. Hicdat: A fast and easy-to-use hi-c data analysis tool. *BMC Bioinformatics* 16, 277.
- Serra, F., Bau, D., Goodstadt, M., *et al.*, 2017. Automatic analysis and 3d-modelling of hi-c data using tadbit reveals structural features of the fly chromatin colors. *PLOS Computational Biology* 13, e1005665.
- Servant, N., Varoquaux, N., Lajoie, B.R., *et al.*, 2015. Hic-pro: An optimized and flexible pipeline for hi-c data processing. *Genome Biology* 16, 259.
- Shavit, Y., Lio, P., 2013. Cytoscape: A cytoscape plugin for visual comparison of hi-c networks. *Bioinformatics (Oxford)* 29, 1206–1207.
- Shavit, Y., Walker, B.J., Lio, P., 2016. Hierarchical block matrices as efficient representations of chromosome topologies and their application for 3c data integration. *Bioinformatics (Oxford)* 32, 1121–1129.
- Shin, H., Shi, Y., Dai, C., *et al.*, 2016. Topdom: An efficient and deterministic method for identifying topological domains in genomes. *Nucleic Acids Research* 44, e70.
- Simonis, M., Klous, P., Splinter, E., *et al.*, 2006. Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture-on-chip (4c). *Nature Genetics* 38, 1348–1354.
- Szalaj, P., Michalski, P.J., Wroblewski, P., *et al.*, 2016. 3d-gnome: An integrated web service for structural modeling of the 3d genome. *Nucleic Acids Research* 44, W288–W293.
- Tanizawa, H., Iwasaki, O., Tanaka, A., *et al.*, 2010. Mapping of long-range associations throughout the fission yeast genome reveals global genome organization linked to transcriptional regulation. *Nucleic Acids Research* 38, 8164–8177.
- Thibodeau, A., Marquez, E.J., Luo, O., *et al.*, 2016. Quin: A web server for querying and visualizing chromatin interaction networks. *PLOS Computational Biology* 12, e1004809.
- Trieu, T., Cheng, J., 2014. Large-scale reconstruction of 3d structures of human chromosomes from chromosomal contact data. *Nucleic Acids Research* 42, e52.
- Trieu, T., Cheng, J., 2016. Mogen: A tool for reconstructing 3d models of genomes from chromosomal conformation capturing data. *Bioinformatics (Oxford)* 32, 1286–1292.
- Trieu, T., Cheng, J., 2017. 3d genome structure modeling by lorentzian objective function. *Nucleic Acids Research* 45, 1049–1058.
- Varoquaux, N., Ay, F., Noble, W.S., Vert, J.-P., 2014. A statistical approach for inferring the 3d structure of the genome. *Bioinformatics (Oxford)* 30, i26–i33.
- Wang, X.-T., Cui, W., Peng, C., 2017a. Hitad: Detecting the structural and functional hierarchies of topologically associating domains from chromatin interactions. *Nucleic Acids Research* 45, e163.
- Wang, S., Xu, J., Zeng, J., 2015. Inferential modeling of 3d chromatin structure. *Nucleic Acids Research* 43, e54.
- Wang, Y., Zhang, B., Zhang, L., *et al.*, 2017b. The 3d genome browser: A web-based browser for visualizing 3d genome organization and long-range chromatin interactions. *bioRxiv*.
- Weinreb, C., Raphael, B.J., 2016. Identification of hierarchical chromatin domains. *Bioinformatics (Oxford)* 32, 1601–1609.
- Wingett, S., Ewels, P., Furlan-Magaril, M., *et al.*, 2015. Hicup: Pipeline for mapping and processing hi-c data. *F1000Research* 4, 1310.
- Xu, Z., Zhang, G., Jin, F., *et al.*, 2016a. A hidden markov random field-based bayesian method for the detection of long-range chromosomal interactions in hi-c data. *Bioinformatics (Oxford)* 32, 650–656.
- Xu, Z., Zhang, G., Wu, C., Li, Y., Hu, M., 2016b. Fasthic: A fast and accurate algorithm to detect long-range chromosomal interactions from hi-c data. *Bioinformatics (Oxford)* 32, 2692–2695.
- Yaffe, E., Tanay, A., 2011. Probabilistic modeling of hi-c contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nature Genetics* 43, 1059–1065.
- Zhang, Z.Z., Li, G., Toh, K.-C., Sung, W.-K., 2013. Inference of spatial organizations of chromosomes using semi-definite embedding approach and hi-c data. In: *RECOMB*, pp. 317–332.
- Zhan, Y., Mariani, L., Barozzi, I., *et al.*, 2017. Reciprocal insulation analysis of hi-c data shows that tads represent a functionally but not structurally privileged scale in the hierarchical folding of chromosomes. *Genome Research* 27, 479–490.
- Zhou, X., Lowdon, R.F., Li, D., *et al.*, 2013. Exploring long-range genome interactions using the washu epigenome browser. *Nature Methods* 10, 375–376.
- Zou, C., Zhang, Y., Ouyang, Z., 2016. Hsa: Integrating multi-track hi-c data for genome-scale reconstruction of 3d chromatin structure. *Genome Biology* 17, 40.

Transcriptome Informatics

Liang Chen and Garry Wong, University of Macau, Macau, China

© 2019 Elsevier Inc. All rights reserved.

Introduction

Since the elucidation of the double-helix as the structure of DNA in biological systems, and the advent of genetic engineering techniques in laboratory science, DNA has been at the forefront of the mind of many scientists. However, RNA has for long served as a central character in the process of life. This can be easily seen in the central dogma which states that DNA is transcribed into RNA and RNA is translated into protein (Crick, 1970). Whereas DNA maintains and replicates the information, it requires an intermediary, RNA, in order to produce protein. Indeed, RNA can perform the functions of DNA as a repository of information and template for replication, as well as catalyze chemical reactions in the form of a ribozyme (Serganov and Patel, 2007). Moreover, RNA is capable of regulating itself from various feedback and feed forward loops. Furthermore, RNA's ability to form secondary structure, which allows it to fold into tertiary structure, makes it a good structural molecule in large protein complexes ranging from those needed in transcription to capping the ends of telomeres (Azzalin *et al.*, 2007; Lodish *et al.*, 1995). Because RNA can perform so flexibly and has been so well conserved, it has been speculated that evolution began with RNA in a RNA world and only later evolved to include DNA and protein (Neveu *et al.*, 2013). It should be no surprise that even the founding group of molecular biologists in the 1950s (James D. Watson, George Gamow, Sydney Brenner and others) formed their club with the name "RNA tie club" rather than DNA (Kay, 2000). Basic fundamental discoveries concerning the structure, function, and roles of RNA in cellular systems continue to amaze and surprise.

Theoretical Background

Transcription

The process of transcription describes the steps required to copy the genetic information encoded in DNA to RNA. Each step involves specific chemical reactions and actions at the subcellular level to bring molecular components together to form these reactions. RNA (ribonucleic acid) differs from DNA (deoxyribonucleic acid) by having a hydrogen atom in the 2' position of the ribose sugar. However, both form polymers based on the ribose sugar backbone and connecting phosphodiester bonds with orientations at 5' and 3' ends. In transcription, the thymidine residue of DNA is replaced by uracil in RNA. Transcription occurs in 4 discrete steps: initiation, promoter escape, elongation, and termination (Kaiser *et al.*, 2007). Initiation describes the step where double stranded DNA, which is wrapped around histones becomes unwound and accessible to the transcription initiation complex (Murakami *et al.*, 2002a,b; Sainsbury *et al.*, 2015). This complex consists of a large number of proteins and ribosomal RNA factors that separate the double stranded DNA and begin synthesizing a RNA polymer complementary to the DNA sequence. The initiation site on DNA is defined by a promoter sequence, CpG islands, and a transcription factor binding site on the double stranded DNA that is proximal, and enhancer DNA sequences that are distal to the initiation site. Activators, co-activators, repressors, and co-repressors are proteins that form part of the transcription initiation complex and provide an additional level of control (Juven-Gershon and Kadonaga, 2010; Ptashne and Gann, 1997). Promoter escape describes the step where the RNA polymer has begun to be synthesized (Dvir, 2002). A RNA polymerase recruited by the transcription initiation complex synthesizes a RNA strand in reverse complement (A→U, C→G, G→C, T→A) using one DNA strand as the template. RNA is synthesized as antiparallel: 5' to 3' orientation. The DNA template used is termed the antisense DNA strand and the opposite DNA strand the sense strand. Different classes of RNA molecules are synthesized depending upon the RNA polymerase used: RNA polymerase I synthesizes ribosomal RNA (rRNA), RNA polymerase II synthesizes messenger RNA (mRNA), and RNA polymerase III synthesizes tRNA as well as other small RNAs. After a ~15-mer RNA nucleotide is synthesized, the RNA polymerase may continue. At this step, promoter escape is completed and elongation begins (Fouqueau and Werner, 2017). Transcription termination following elongation of the RNA strand is signalled by DNA sequences downstream of the promoter and results in RNA polymerase stopping its synthesis (Richard and Manley, 2009). The DNA sequences recognized by the transcription initiation complex to the termination signal define a transcription unit and are also referred to as a genes (Alberts, 2007). Post-transcriptional processing of newly synthesized RNAs occurs following termination. These include chemical modifications to RNA such as 5' capping, polyadenylation, and splicing of mRNAs, RNA editing, and degradation. Regulatory processes that influence the synthesis and abundance of RNAs that occur before and during transcription are termed pre-transcriptional and transcriptional regulation, respectively, while those that occur after the RNA has been synthesized are termed post-transcriptional (Jacob and Monod, 1961). Prokaryotes differ from eukaryotes in transcription by less complexity, but basic principles are maintained (Eick *et al.*, 1994). Some RNA based viruses use RNA as the template in transcription (Baltimore, 1971). Reverse transcription describes the process where RNA serves as the template for the synthesis of DNA (Baltimore, 1970; Temin and Mizutani, 1970). It has been estimated that greater than 85% of an eukaryotic genome is transcribed into RNA at some time during the life of the organism (Hangauer *et al.*, 2013).

Major RNA Classes

While most RNA classes have been historically defined by their chemical and functional properties, the power of next-generation sequencing has blurred some of these distinctions. A recent example shows how a single molecule can be classified as either an eRNA or lncRNA (Espinosa, 2016; Paralkar *et al.*, 2016). RNA classes may also be defined via the proteins family members they bind such as Argonautes to distinguish between miRNAs and piRNAs (Meister, 2013). Below, we list major RNA classes as defined in the current literature.

mRNA – messenger RNAs represent protein coding and noncoding RNAs transcribed by RNA polymerase II. They are capped by addition of a methyl-7 guanine moiety and modification of 2'-O-methyl ribose (Fabrega *et al.*, 2004; Kiss, 2001). After synthesis, adenine ribonucleic acid residues (poly A +) are added by polyadenylate polymerase via a complex that recognizes a poly A signal sequence (AAUAAA) in the last 10–30 nucleotides of the mRNA (Colgan and Manley, 1997). The poly A + mRNA is exported from the nucleus to the cytoplasm where it is edited to remove introns and splices together the exons. These edited mRNA molecules are translated into proteins by ribosomes located either in rough endoplasmic reticulum or in the cytosol (Stephens *et al.*, 2008). Noncoding RNAs may undergo further processing. It is estimated that humans have around 20,000 protein coding genes which may be sliced and edited to form approximately 10 different forms for each gene on average.

tRNA – also known as transfer RNA, it is the adaptor molecule that translates triplet codon information from mRNA to amino acid. It is part of the translation machinery called the ribosome that also contains ribosomal RNAs and proteins. The size of a tRNA is ~75 to 90 nucleotides. tRNAs are transcribed from clusters in the genome by RNA polymerase III and is highly conserved across species (Sharp *et al.*, 1985).

rRNA – also known as ribosomal RNA, it is a structural molecule that forms the ribosomal complex responsible for translating mRNA into protein (Palade, 1955). The ribosomal complex consists of multiple rRNAs and over a dozen proteins depending upon the species, along with tRNAs (Wilson and Doudna Cate, 2012). rRNAs are also highly conserved and present in all taxa and therefore are often used in evolutionary studies (Woese, 1987). rRNA is the most abundant class of RNAs and constitutes ~50% of all RNA species in a cell. rRNAs range in size from 1.5 to 5 kb (Noller, 1991).

Ribozymes – these are RNA molecules that have catalytic ability, typically to generate RNA polymers by ligating or cleaving RNA fragments together or in the ribosome to catalyze protein peptide production (Guerrier-Takada *et al.*, 1983; Kruger *et al.*, 1982). Ribozymes can be very small: around 50 nucleotides in total. Ribozymes were originally naturally occurring RNA molecules, however, they have been actively pursued and engineered as biotechnology products especially to cleave unwanted RNA molecules such as from viruses or those generated in diseases (Pyle, 1993).

eRNA – enhancer RNAs are recently discovered non coding RNAs transcribed from enhancer regions (Kim *et al.*, 2010). They are short sequences and form multiple overlapping transcripts that are not typically spliced or polyadenylated. They commonly occur at distal enhancers several kb away from the promoter and are believed to act by aiding in the recruitment of co-activators to the transcription initiation complex (Kowalczyk *et al.*, 2012). The size of eRNA varies but can be as short as 50 nucleotides and as large as several kb (Natoli and Andrau, 2012).

lncRNA – long non-coding RNAs define a class of RNAs that are >200 nucleotides long but do not code for proteins. It is now estimated that the human genome may contain >50,000 lncRNAs (Iyer *et al.*, 2015). They may be transcribed antisense to a coding RNA transcript or in intergenic spaces. They are polyadenylated and spliced and therefore share many of the same biosynthesis properties as mRNAs. While they are abundant, the precise functions of only a few are known, such as Xist which acts in dosage compensation of the X chromosome via recruitment of a polycomb repressor complex that lays down repressive epigenetic marks (Ballabio and Willard, 1992; Calabrese *et al.*, 2012). Another example is the telomerase RNA component, TERC that acts as scaffold and template to add DNA ends to telomeres (Greider and Blackburn, 1989; Kapranov *et al.*, 2007; Kung *et al.*, 2013).

miRNA – microRNAs are small noncoding RNAs transcribed by RNA polymerase II mostly but sometimes by RNA polymerase III as well (Borchert *et al.*, 2006; Lee *et al.*, 1993; Lee *et al.*, 2004). They may also be spliced from introns. These are termed mirtrons. They may also be transcribed in intergenic spaces. The initial transcript is a primary miRNA (pri-miRNA) and after its cleavage to a 75–90 nucleotide hairpin secondary structure, it is termed a pre-miRNA (precursor miRNA) (Ha and Kim, 2014). The finally processed 21 or 22 nucleotide form is the mature miRNA. miRNAs function as down-regulators of mRNA either pre-transcriptionally or post-transcriptionally. Human genomes may express >6000 different miRNAs with a very wide range of expression levels (Londin *et al.*, 2015). Isomeric forms of miRNAs termed isomiRs are also known to exist and are derived from differential processing of pre-miRNAs by Dicer enzyme (He and Hannon, 2004).

moRNA – microRNA off-set RNAs are small RNA fragments 15–22 nucleotides in length that are derived from the pre-miRNA but are “offset” on the hairpin either located upstream of the 5p mature miRNA sequence or downstream of the 3p mature miRNA sequence (Langenberger *et al.*, 2009; Shi *et al.*, 2009). The abundance of moRNAs compared to miRNAs is very low: they typically represent <1% of miRNAs derived from the same pre-miRNA. moRNAs have been found most abundantly in stem cells, embryonic bodies, and germ cells (Asikainen *et al.*, 2015). moRNAs may act to regulate mRNA expression in molecular pathways similar or shared with miRNAs (Langenberger *et al.*, 2009; Shi *et al.*, 2009).

snRNA – small nuclear RNAs are 100–300 nucleotides in length and highly abundant in the nucleus. Their sequences are Uridine rich and they function as a scaffold in splicing introns within the spliceosome complex. snRNAs exist in riboprotein complexes termed snrps (small nuclear riboproteins). They are a family of molecules and their high abundance in cells often leads to them being used as a loading control in experiments involving quantitation of small non coding RNAs (Matera *et al.*, 2007).

snoRNA – small nucleolar RNAs are 70 nucleotides in length and processed from intron fragments. Vertebrates express several hundred snoRNAs. snoRNAs act as a guide for rRNA processing, rRNA maturation, or chemical modification of complementary sequences in rRNA (Matera *et al.*, 2007).

piRNA – piwi interacting RNAs are 26–31 nucleotide RNAs transcribed from specific loci (Saito *et al.*, 2006; Vagin *et al.*, 2006). Their sequences are highly diverse and thousands of different piRNAs may be transcribed from different regions of a chromosome. Simple model organisms like fruit flies and nematodes may express tens of thousands of piRNAs while more complex organisms like mice and humans may express even more (Das *et al.*, 2008). piRNAs typically begin with Uridine, however remaining sequences may be highly variable and no consistent secondary structure has so far been found. Their initial function has been described as inhibiting transcription of transposon sequences in the genome, but more recently they have been shown as essential for controlling transcription in sperm cells (Iwasaki *et al.*, 2015).

circRNA – circular RNAs are recently discovered covalently closed RNA molecules derived from splicing of exonic sequences of mRNA or lncRNAs (Hansen *et al.*, 2013; Memczak *et al.*, 2013). Some circRNAs also contain intronic sequences. The covalent linkage to form a circular molecule occurs via a 3–5′ or 2–5′ phosphate. Although initially found in higher organisms, they now appear to be found in a wide variety of organisms including most eukaryotes studied (Ebbesen *et al.*, 2016). While their function remains under active investigation, their high abundance in brain tissues and during development suggests involvement in specific molecular functions in particular biological processes. It is hypothesized that they may act as molecular sponges by sequestering active mature miRNAs, although other potential mechanisms of action have been suggested. Sizes of circRNAs range from less than 200 nucleotides to greater than 4 kb (Lasda and Parker, 2014; Salzman, 2016).

endo-siRNA – endogenous silencing RNAs are short ~20 to 27 nucleotide sequences that are similar to miRNAs as short RNA species that attenuate gene expression (Asikainen *et al.*, 2008; Okamura and Lai, 2008). They can be derived from dsRNA structures, but unlike miRNAs do not form hairpin structures. They may however share components of the RNA silencing machinery with miRNAs including amplification pathways and binding to Argonautes. They are highly abundant in many different species and can be found in germline cells such as sperm and thus may be important in developmental processes (Nilsen, 2008; Yuan *et al.*, 2016).

rasiRNA - Repeat associated silencing RNAs are short 24–29 nucleotide RNAs that form a related but distinct family of small noncoding RNAs. Similar to piRNAs, they act in transposon and retrotransposon silencing in the genome, but also control transcription of other repeat sequences in the genome (Aravin *et al.*, 2001). Similar to piRNAs, they bind to piwi family Argonaute proteins, but are expressed in organisms such as yeast and plants where piRNAs have not been found, and are therefore distinct species from piRNAs (Aravin *et al.*, 2001).

ceRNA – Competing endogenous RNAs comprise a class including all RNA molecules that may compete for miRNAs via complementary pairing and thus attenuate miRNA actions (Salmena *et al.*, 2011). The concept has been considered as a type of molecular sponge that absorbs miRNAs and prevents their function. The growing list of ceRNAs includes molecules from other RNA classes including mRNAs, lncRNAs, circRNAs, and pseudogenes (Hansen *et al.*, 2013). ceRNAs may also act in coordination to regulate a specific gene (Tay *et al.*, 2011) (Fig. 1).

Biological Sources of RNA

Early sources of RNA for biological study came from dissected tissues that were easily accessible, abundant, soft, non-fibrous, and were easy to homogenize in order to perform biochemical fractionations (Chomczynski and Sacchi, 1987). These tissues included liver, kidney, brain, spleen, and heart from mammals. Skin, although the most abundant organ by weight from humans was typically not used in isolations because of the presence of RNAase that degraded RNA upon contact (Probst *et al.*, 2006). As different biological questions arose and more sensitive methods for biochemical analysis developed, specific tissue regions were dissected and RNA isolated. These included different brain regions such as hippocampus from brain, lymphocytes from blood, and veins from liver. As even more tissue specific resolution was sought, laser capture microdissection methods became available and isolation of ~50 cell samples from tissues were possible (Mikulowska-Mennis *et al.*, 2002). More recently, RNA isolation from single cells has come to the forefront of experimental research (Saliba *et al.*, 2014). However, even higher resolution is still being pursued and while not routine, subcellular RNA fractions from nucleus, cytoplasm, or mitochondria to name just a few structures, have been available for many years (Derrien *et al.*, 2012).

In contrast to intracellular organelles with RNA components, an extracellular structure termed the exosome, has become an exciting and unusual place to isolate RNA (Johnstone *et al.*, 1987). Exosomes are small <150 nm lipid vesicles that contain RNA, usually miRNAs but other species as well (Valadi *et al.*, 2007). Exosomes are excreted into the extracellular space and can eventually end up in body fluids or the circulation. The body fluids from where exosomes have been isolated include blood, urine, and amniotic fluid (Keller *et al.*, 2007). The amount of exosomal RNA secreted into these tissues is small and estimates of their abundance include 1–10 ng/mL fluid such as plasma (Enderle *et al.*, 2015). Exosomal sources of RNA are different than cRNA which are cellular free RNAs which have nonspecific cellular origins. cRNAs can be isolated like DNA from a large list of body fluids including amniotic fluid, spinal fluid, saliva, tears, and breast milk (Li *et al.*, 2014a). Like DNA used for diagnostics, it is possible to isolate RNA from a fetal amniotic fluid and this species is termed cffRNA, cell free fetal RNA (Vora *et al.*, 2017). The principle advantage of isolating RNAs from body fluids is due to its noninvasive nature, thus sparing the organism from losing tissues. Moreover many of these fluids such as urine and blood are highly abundant (Gahan, 2015; Huang *et al.*, 2013).

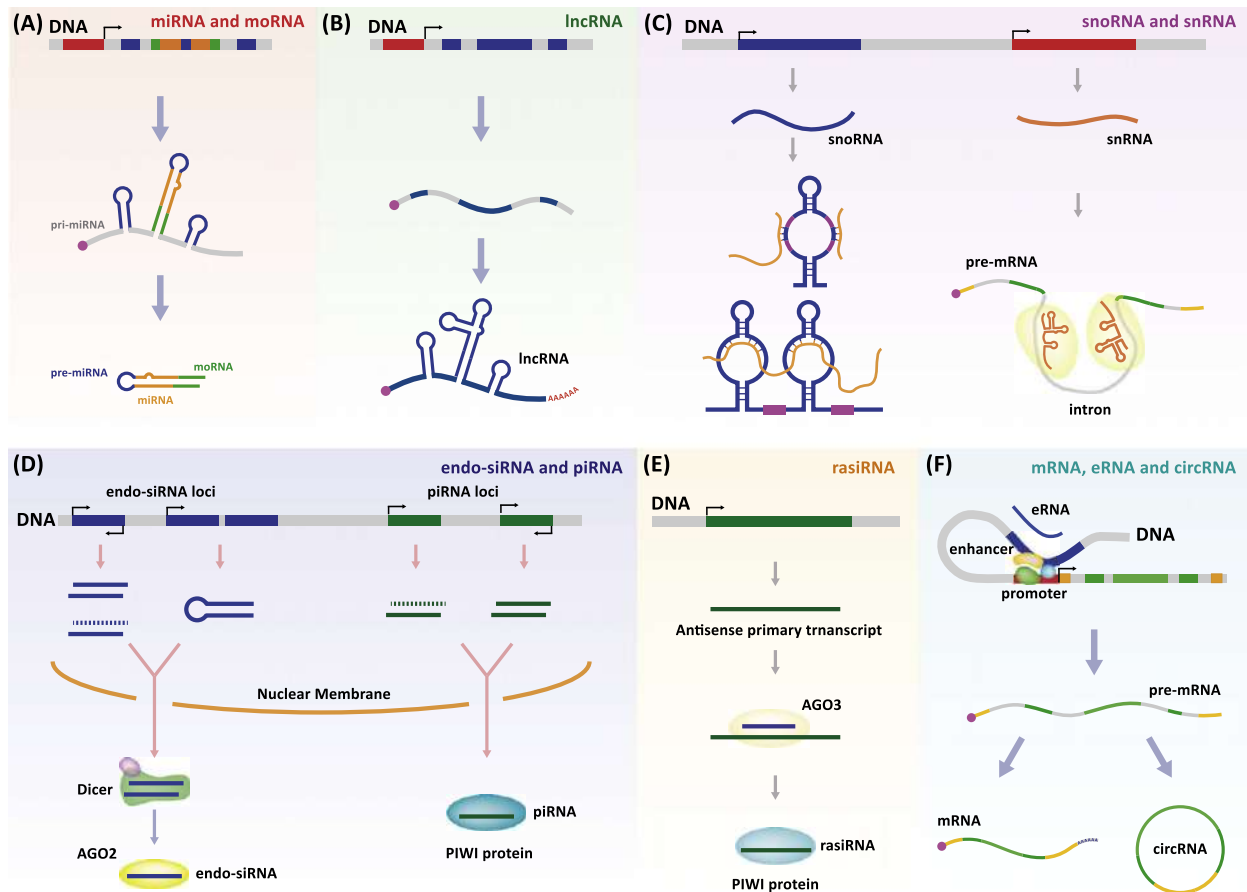


Fig. 1 Basic biogenesis of different RNA classes. (A) Biogenesis of miRNA and moRNA. A brief cartoon shows the origin of miRNA and moRNA. miRNA is marked in orange and moRNA is marked in green. (B) Biogenesis of lncRNA. lncRNA undergoes a splicing processing event. (C) Biogenesis of snoRNA and snRNA. The left side shows snoRNAs base pairing with ribosomal RNA to guide folding and cleavage of precursor rRNA. The right side shows snRNAs removing introns from pre-mRNA. (D) Biogenesis of endo-siRNA and piRNA. endo-siRNA and piRNA are exported from the nucleus to cytoplasm and different RNA binding proteins (RBPs) (Argonaute 2, AGO2 and PIWI protein are shown as examples) are required to process them. endo-siRNA and piRNA are marked by blue and green, respectively. (E) Biogenesis of rasiRNA. The process requires the Argonaute protein AGO3, and PIWI protein. (F) Biogenesis of mRNA, eRNA and circRNA. eRNAs are transcribed from enhancer regions. Canonical splicing of mRNAs produces linear transcripts while back-splicing produces circRNAs.

Meta-samples, derived from the metagenomics field, describe samples with mixtures of organisms (Hugenholtz *et al.*, 1998). These are typically obtained from the environment such as soil or sea water, but can also be complex cellular mixtures from discrete sources such as fecal matter (David *et al.*, 2014). Both RNA and DNA can be sequenced from these samples. A classical application of sequencing RNA from these samples is to identify ribosomal RNA as these are unique identifications for species in the sample and thus make it possible to identify individual species in a population living in a complex biological mixture.

Laboratory cell culture remains one of the most common sources of RNA. These can include primary cell cultures, transformed cell cultures, and stem cell derived cultures including induced pluripotent stem cells (iPSC), embryonic stem cells (eSC), or embryoid bodies (EB) to name a few. The laboratory success in growing and maintaining these different types of cells including adherent and suspended in liquid type cultures has made these cells a very flexible and dynamic source of RNA for studies.

Fractionation and enrichment or purification of specific classes of RNAs occurs independent of the sources. A total RNA purification from a biological sample contains >80% rRNAs, while mRNA and lncRNAs together account for 1%–2% of the total RNA (Kaiser *et al.*, 2007). The amount of small RNAs by weight are less than this. mRNAs can be separated from total RNAs by isolation methods using oligodT binding to take advantage of the polyadenylation of this RNA class (Morrison *et al.*, 1979). Small noncoding RNAs can be enriched by size fractionation on columns or more cleanly by size fractionation followed by elution on polyacrylamide gels followed by elution. Any carryover DNA from these fractionation procedures are digested with an RNase free DNase. As detection methods become more sensitive, a greater amount of attention has been paid to increasing the purity of each fraction.

RNA Modifications

Chemical modifications to RNA molecules has been known for a long time, however, interest in their study has been hampered by lack of experimental tools to detect their presence and low abundance of some modifications (Song and Yi, 2017). This is now changing rapidly and with highly sensitive mass spectrometry techniques as well as other analytical chemistry methods, more than 140 chemical modifications of RNA are known of which >100 are known to occur naturally (Helm and Motorin, 2017). More than half of these modifications have been characterized on rRNA and tRNA with at least 13 described for mRNA. The most common modification for mRNA is addition of a m⁷G (methylation of guanine nucleoside moiety) cap followed on the same molecule by 2'-O-methyl (methylation of ribose 2') secondary cap (Shatkin, 1976). The purpose of this modification is to increase stability of the RNA. Similarly, tRNA has a m⁵C modification that functions to prevent degradation. The most prevalent internal mRNA modification is N(6)-methyladenosine, m⁶A (Dominissini *et al.*, 2012). The most common mRNA modification in vertebrates are adenosine to inosine (6-deaminated adenosine) which is catalyzed enzymatically via a family of adenosine deaminase acting on RNA (ADAR) proteins. This RNA modification is not randomly distributed along the transcriptome but appears to occur at double stranded RNA regions such as in 3' UTRs (untranslated regions) and miRNAs. These modifications have been shown to have effects on levels of transcripts and thus can be considered a means for post-transcriptional modification. Another abundant modification in mRNAs and lncRNAs is m⁶A (methyl-6-Adenosine) which has been shown to affect transcript stability and translation as regions in the 5' UTR and stop codons are enriched for this modification. RNA modifications at the transcriptomic level have led to the advent of a new field termed "epitranscriptomics". This field shares many similar concepts with the epigenetics field including the identification of RNA epitranscriptomic readers (Li *et al.*, 2016), writers, and erasers (Frye *et al.*, 2016). The identity of specific RNA metabolism pathways combined with transcriptome scale experiments has made this young field move very quickly.

Driving the field of epitranscriptomics is next generation sequencing. RNA modifications at a transcriptomic scale are possible by immune precipitating RNA fragments with the modification via an antibody and subsequent sequencing, commonly referred to as RIP-seq (Zhao *et al.*, 2010). Using this and modifications of this method enable single nucleotide resolution of modified RNAs. A challenge for this approach are contaminants since RNA modification events, with a few exceptions, are not very abundant in the cell. Another challenge is in mapping RNA modification events to DNA sequences, so called RNA-DNA difference (RDD) comparisons (Chi, 2017). For example, adenosine→inosine modifications can be identified by comparing RNA to DNA since adenosine base pairs with thymine and inosine pairs with adenine, cytosine, or uracil. After synthesis of cDNA from RNA for sequencing, these differences between the original DNA sequence transcribed and the RNA sequence can identify specific locations of differences. However, mapping algorithms may not correctly assign mismatched sequences.

RNA Transcriptomics Nomenclature

Gene nomenclature committees representing various scientific communities have the responsibility to produce, maintain, and update gene names. These committees are necessary to maintain the stability of gene names and to ensure their accuracy. An example of such a group is the nomenclature committee of the Human Genome Organization (HUGO) that maintains *Homo sapiens* genes (Povey *et al.*, 2001). Similar committees exist to maintain gene names for mouse, rat, and *Drosophila* scientific research communities.

In general, genes names are italicized while protein names are not. *INS* refers to the human insulin protein while *INS* refers to the human insulin gene. Gene names are one type of identifier for a gene which may have many depending upon the database. The National Center of Biotechnology Information GENE ID for human insulin is 3630, while the HUGO nomenclature committee GENE ID is 6081. Gene names provided by a nomenclature committee are useful as they provide a common identifier that may be used as a standard among many users.

Gene names are typically 3–4 letters. For example, the human insulin gene is *INS* and the human insulin receptor gene is *INSR*. Model organisms such as worms (*Caenorhabditis elegans*), and fruit flies (*Drosophila melanogaster*) have gene names provided by the phenotypes of mutants recovered from forward genetic screens and maintained by their scientific communities (Gelbart *et al.*, 1997; Stein *et al.*, 2001). Examples include *Arg⁻* for a strain requiring arginine as a phenotype in yeast, *dpy-3* for dumpy appearance in worms, and *w* or *white* for white eye in fruit flies. Names are also assigned based on orthology to other organisms such as *cyp35a* for cytochrome P450 family 35a gene in *C. elegans* (Aarnio *et al.*, 2011). In bacteria, a 3 letter designation of the molecular pathway or biological process the gene product is involved in plus a capital letter denotes a gene, for example, *hisA* for gene A of the histidine biosynthesis pathway.

RNA transcripts can always be mapped to a chromosome and nucleotide position on a sequenced genome (Hubbard *et al.*, 2002; Kent *et al.*, 2002). The units are essentially coordinates on the genome. For example, the human insulin gene is located at Chromosome 11, nucleotides 2,159,779–2,161,341. Likewise, all transcript elements such as exons, enhancers, miRNAs etc. can also be identified by their coordinates. Another coordinate system identifies the continuous sequence (contig) and the order of genes. A contig may originate artificially from a yeast artificial chromosome library, phage library, or naturally from a chromosome. For example, the *daf-2* gene of *C. elegans* can also be named Y55D5A.5 as it is 5th gene on contig Y55D5A which is derived from a yeast artificial chromosome library.

miRNA names begin with the species (*hsa-* for *Homo sapiens*), followed by miR-, and then a number (e.g., *hsa-miR-128*). The mature miRNA has the miR (uppercase R) designation while the precursor is denoted with mir (lower case r, e.g., *hsa-mir-128*)

(Griffiths-Jones *et al.*, 2006). The numbers are provided sequentially for new miRNAs, but if an ortholog exists, they will be given the same number as the other orthologs (e.g., mus-mir-128). New mir ortholog names simply require the species prefix to be changed. Numbered suffixes indicate distinct precursors from unique genomic loci but identical mature miRNA sequences (e.g., mus-mir-128-1). Lettered suffixes indicate mature miRNAs closely related but from distinct precursors (e.g., mir-128b). The final suffix is given for the side of the hairpin from which the miRNA is located: either the 5' side designated 5p or 3' side designated 3p (e.g., miR-128-1-3p) (Budak *et al.*, 2016).

Other RNA classes also have their own nomenclature and naming conventions such as those for rRNAs, lncRNAs, and piRNAs provided by the HUGO Gene Nomenclature Committee (Gray *et al.*, 2015).

Transcriptome Measurements

Three essential measurements can be made on the transcriptome level of an organism: sequence, abundance, and structure (Conesa *et al.*, 2016). From a gene's sequence, specific information regarding the transcription start site, intron-exon usage, transcript termination, sequence variation, can be determined, and gene identity and orthology can be inferred. Gene abundance measurements can be determined by the number of transcripts representing a gene sequenced while secondary structure can be determined by various programs that fold the RNA sequence based on minimal free energy constraints as well as other structural factors (Miao and Westhof, 2017). Orthology is inferred by best alignment for similarity with other nucleotide sequences from other organisms while paralogy is inferred by alignment within the same organisms to detect genes belonging to the same family (Doyle and Gaut, 2000).

Whole transcriptome sequencing is performed on a sequencing platform that integrates sample preparation protocols, sequencing chemistry and detection, and data analysis as a single activity (Korpelainen *et al.*, 2014). Each transcriptome level sequencing has its own properties, history, biases, advocates, and advantages and disadvantages. The productivity and accuracy of sequencing platforms are constantly and rapidly improving (van Dijk *et al.*, 2014). Since the chemistry and technology for sequencing DNA is more mature and advanced than RNA, transcriptomic platforms typically reverse transcribe RNA into DNA and then sequence the complementary DNA. Among the commonly used platforms are microarrays, Illumina, Ion Torrent, Pacific Biosystems, and Oxford Nanopore.

Transcriptome Sequencing Platforms

Microarrays use the chemical principle of nucleic acid hybridization (DeRisi *et al.*, 1997). For a given sequence A, C, G, and T, hybridize with their complementary nucleotides via hydrogen bonding T, G, C, and A, respectively. Short DNA fragments of known sequence or identity are bound or directly synthesized to a solid support and a fluorescently labelled cDNA reverse transcribed from an RNA sample is hybridized to the fragments on the support. Since the abundance of a specific cDNA is directly correlated with the original amount of RNA in the sample, the amount hybridized is an indicator for the abundance of an RNA sequence. Since DNA sequences bound to the solid support are known and are fixed to a specific location, the identification of the gene can be determined by the location of the hybridization signal on the support. This platform has several advantages including relatively simple workflow and chemistry for detection and analysis. It is reasonably cost effective as well and entire transcriptomes can be interrogated in single replicates for a couple of hundred dollars. A disadvantage is that sequences to be interrogated need to be determined beforehand for synthesis of the microarray, cross hybridization of cDNAs can occur, and sensitivity of detection is limited to fluorescent labelling. In the early use of microarrays, many academic labs produced and printed their own microarrays, however, the field has lately been dominated by Affymetrix due to their ability to synthesize oligonucleotides directly onto a substrate.

Next-generation sequencing technologies (NGS) exploit massively parallel sequencing by synthesis (MPSS) as a chemical principle for transcriptome level sequencing (Marioni *et al.*, 2008). In the Illumina platform, RNA isolated from a sample is fragmented, and then indexes/bar codes, sequencing primer, and amplification primers are added as an adaptor molecule to the RNA. The RNA is reverse transcribed to cDNA and a second DNA strand is synthesized to yield a double stranded DNA. This DNA molecule is amplified and then sequenced by synthesis one nucleotide at a time. After each addition a specific fluorescent signal is produced for each nucleotide.

On the Illumina platform, the number of nucleotides on a DNA fragment can be up to 200 nucleotides, however 100 is more common. The number of templates sequenced, equivalent to the number of reactions on a single chip, can reach 2×10^9 . Each sequence obtained from a template is called a "read". The number of reads can be doubled to 4×10^9 using a dual flow cell, and further can be doubled to 8×10^9 by sequencing from both ends of the DNA called paired end sequencing. This can then generate up to 1.6×10^{12} of sequence information. Since the chemical reactions occur on the chip and reactants need to be removed for each chemical reaction, the solid substrate is termed a flow cell, and each flow cell is divided into 8 lanes. Indexing allows a user to add a tag, also known as a bar code, to each RNA molecule that will be sequenced. All RNAs from a single library have a specific bar code, and this allows mixing of libraries to be sequenced on a flow cell. The result is that the 2×10^9 reaction capacity can be divided into an unlimited number of libraries to be sequenced. This allows multiplexing of reactions, but more importantly drives down the cost of sequencing on a per reaction basis. The cost per reaction which equates to a read sequence can be as low as 0.001 US cents. While this might seem very cheap per reaction, there are other viable and competitive MPSS platforms.

Among these are Ion Torrent's personal genome sequencers (Liu *et al.*, 2012; Merriman and Rothberg, 2012; Quail *et al.*, 2012). While also based on a sequencing by synthesis principle, the detection of each added nucleotide is based on the release of a hydrogen ion which occurs after each addition. A pH change is measured in real time using semiconductors that are multiplexed in the platform. The advantage of this chemistry is that no modified nucleotides are needed. If homopolymer regions are encountered, for example, TTTT regions, then multiple reactions will occur leading to bigger signals. This platform, while not generating as many reads as Illumina, has lower instrumentation cost and the total run time for generating sequence is substantially lower, for example, 2 h compared to approximately 24 h. Also, for infrequently run instruments or low through-put, it provides an advantage since it is easier to fill and run at full capacity. Length of reads is also important and Ion Torrent routinely uses 200 nucleotide reads with 400 read chemistry available. The number of reads produced on the Ion Torrent platform depends upon the chip used but can be as low as 0.5×10^6 to as high as 5×10^6 per run. At 200 nucleotide chemistry, a run generating 5×10^6 reads would produce 1×10^9 of sequence information.

Another sequencing platform is Pacific Biosystems which also utilizes a MPSS principle but differs from Illumina in utilizing the detection of newly added nucleotides at single molecule resolution enabled by detection of fluorescent phospholinked nucleotides (Rhoads and Au, 2015). In this platform, a single well contains a synthesis reaction of an enzyme, the DNA template, and labelled nucleotides, of which the addition of a specific nucleotide results in a specific signal. The signal is detected at the bottom of the well using a zero mode wave guide that directs one molecule of template and one molecule of DNA polymerase per well. Signals are detected as the synthesis of each nucleotide occurs, thus is giving this technology the name single molecule real-time (SMRT) sequencing. The addition of nucleotides produces a natural DNA molecule and thus can produce very long reads on average of above 10,000 nucleotides (Rhoads and Au, 2015) with up to 75,000 reads giving 0.75×10^9 of sequencing data. While the super long reads are advantageous especially in genome sequencing applications, the Pac Biosystems platform has also been of use in *de novo* transcriptome assembly of novel transcriptomes in RNA sequence informatics. The instrumentation cost is quite high compared to other platforms so this has been another limitation as well as a higher error rate. However, the ability to produce large reads has made it indispensable for applications such as genome assembly, transcriptome assembly, and sequencing complex genetic samples such as those from metagenomics.

Oxford Nanopore is another single molecule sequencing platform (Jain *et al.*, 2016). The technology depends upon small changes in electrical current when characteristic nucleotides pass through the pores. The small size of the platform, similar to a USB memory stick makes this an attractive technology. A recent version contains 500 nanopores from which single DNA molecules pass at 450 nucleotides per second with a generation of more than 5×10^9 of sequence according to a technical update from the company. While accuracy has been a problem for this platform in the past, technology improvements have increased this parameter dramatically in recent months.

Table 1 shows the comparison of sequencing platforms (Fox *et al.*, 2014; Li *et al.*, 2014b; Mardis, 2017).

Transcriptome Mapping

Microarrays contain spatially addressed genes already identified. NGS platforms require the reads obtained from sequencing to be mapped onto a transcriptome if one exists, and to produce a *de novo* transcriptome if it does not. In the former case mapping is typically performed by alignment of the read to a sequence from the transcriptome and then assignment of the read to a particular gene (Mortazavi *et al.*, 2008). In the latter, reads are aligned with each other to obtain continuous sequences (contigs) to generate an assembly, and then assigned to that particular contig (Martin and Wang, 2011). While mapping is intuitively straightforward, the large number of reads and large size of the transcriptome require efficient algorithms to improve upon the computational task of mapping. One algorithm, the Burrows-Wheeler transform (BWT), converts the transcriptome sequences into blocks, sorts these, and because of the many repeats of sequences in the transcriptome, makes it more efficient to search through rather than going nucleotide to nucleotide through the transcriptome. Another algorithm to align reads is to use a hash table, which is simply a table of possible short nucleotide sequences that uses a hash function to compute their locations. Finally, de Bruijn graphs are used since a transcript might have many alternative exons and introns and these need to be considered in the alignment.

In Table 2, different tools which align reads as un-spliced or spliced are shown.

Table 1 Comparison of sequencing platforms

	<i>Microarrays</i>	<i>Illumina</i>	<i>Ion Torrent</i>	<i>Pacific biosystems</i>	<i>Oxford nanopore</i>
Chemistry	Hybridization	MPSS	Hydrogen ion	SMRT	SMRT
Read lengths	N/A	50–300 bp	200–600 bp	250 bp to 40 kb	– 150 kb
Read modalities	N/A	Single, paired end	Single, paired end	Single	1D or 2D reads
Throughput	10–100 MB	7.5–300 Gb	1.5–4.5 Gb	500 Mb to 1 Gb	10–20 Gb
Accuracy		100.00%	99.99%	99.99%	99.96%

Note: Columns indicate different sequencing platforms.

Rows indicate common features of platforms. Chemistry provides technical differences of each platform. Read Lengths measures the input read size. Read Modalities provide the type of read each platform can generate. Throughput estimates the amount of data that is processed by each platform. Accuracy indicates the accuracy of reads generated by the platform. Abbreviations: bp, basepair; Gb, gigabyte; kb, kilobase; MPSS, massively parallel sequencing by synthesis; MB, megabyte; SMRT, single molecule real-time.

Table 2 Aligner and *de novo* assembler

Tools	Alignment type	Read length	Algorithm	System	Link
SOAP	Un-spliced	Short	BWT-based	Linux/Unix/MAC OS	http://soap.genomics.org.cn/
Bowtie	Un-spliced	Short	BWT-based	Linux/Unix/MAC OS	http://bowtie-bio.sourceforge.net/
Bowtie2	Un-spliced	Long	BWT-based	Linux/Unix/MAC OS	http://bowtie-bio.sourceforge.net/bowtie2
BWA	Un-spliced	Short/Long	BWT-based	Linux/Unix/MAC OS	http://bio-bwa.sourceforge.net/
TopHat	Spliced	Short/Long	BWT-based	Linux/Unix/MAC OS	https://ccb.jhu.edu/software/tophat
HISAT	Spliced	Short/Long	BWT-based	Linux/Unix/MAC OS	http://ccb.jhu.edu/software/hisat2
MAQ	Un-spliced	Short	Hash tables	Linux/Unix/MAC OS/Windows	http://maq.sourceforge.net/
PerM	Un-spliced	Short/Long	Hash tables	Linux/Unix/MAC OS	https://code.google.com/archive/p/perm/
SHRIMP	Un-spliced	Short/Long	Hash tables	Linux/Unix/MAC OS	http://compbio.cs.toronto.edu/shrimp/
Trinity	<i>de novo</i>	Long	<i>de Bruijn</i> graph	Linux/Unix/MAC OS	http://trinityrnaseq.github.io
Oases	<i>de novo</i>	Short	<i>de Bruijn</i> graph	Linux/Unix/MAC OS	https://www.ebi.ac.uk/~zerbino/oases/

Note: Rows indicate aligner examples. Columns list the common features of the aligners. The algorithm gives the background algorithm used in the aligner. The System lists the operating system where the aligner works. Link shows the home web page for each aligner.

Mapping values for a data set are provided as a percentage of reads mapped to the transcriptome or genome. More precise values can be given as percentage of read nucleotides mapped to a transcriptome calculated by taking all nucleotides aligned to the transcriptome divided by the number of nucleotides sequenced. These are usually termed % on-target.

Mapping data from NGS platforms generates an alignment file that contains read information and location on transcriptome where it maps. These are sometimes provided as coordinates in the genome where the transcriptome is located that also provides exon and intron data. These files are known as BAM, SAM, WIGGLE files, and have .bam, .sam, .wig extensions (Zhang, 2016). This data is very dense, and an index file can provide a means of searching through the alignment file. Index files have names such as BAI and have .bai extensions.

Transcriptome Output and Normalization

Microarray or NGS sequence data can typically be output as a tab-delimited text file with gene names in rows and samples names in columns (Korpelainen *et al.*, 2014; Laine *et al.*, 2014). Each field contains an abundance value that corresponds to a specific gene and specific sample. For microarrays the value is a fluorescence measurement and for NGS platforms the value is for number of reads. These values need to be “normalized” since fluorescence signals may vary from experiment to experiment in microarrays. Per-chip normalization of microarrays divides all values on the microarray by a number while per-gene normalization divides only a specific gene across different microarrays by a number. Other microarray normalization procedures may convert values to Z-scores, indicating a fluorescence measurement at Z-value, equivalent to one standard deviation from the mean. Thus, a value might be converted to -3.5 meaning its intensity is -3.5 standard deviations from the mean intensity of values on the microarray. Another microarray normalization method converts values to ordinals, so a gene might be the 503rd most highly expressed from 20,000 genes.

For data from NGS platforms, normalization also needs to be performed because while reads are an absolute measurement of transcript abundance, the values for each gene are also a function of the number of overall reads obtained from a sample (Korpelainen *et al.*, 2014). One normalization method is to divide the number of reads for each gene per million reads that map to the transcriptome. This value is called RPM. Values can also take into account the size of a transcript because a transcript 4 kb in size will have 8 times more reads mapping to it than a gene 0.5 kb in size if all other factors are equal. To normalize for transcript size, values are divided by the size of the gene over 1 kb. These values are termed RPKM for reads per kilo bp of the transcript per million mapped reads. Because Illumina can produce paired-end reads by sequencing from both sides of a transcript, the number of reads can be converted into fragments, so that reads from 2 ends of a transcript produce a fragment and these values are normalized as FPKM for fragments per 1 kbp gene size per million mapped reads.

Analysis of Transcriptome Data

Differential gene expression, commonly abbreviated as DG or DGE analysis refers to the analysis and interpretation of differences in abundance of gene transcripts within a transcriptome (Conesa *et al.*, 2016). Lists of genes that differ between 2 sample sets are often provided by RNA-seq data analysis tools, or can be generated manually by statistical testing of data sets. Due to the large number of genes to be tested, (e.g., $> 20,000$ in the human genome), multiple testing correction such as Bonferroni correction is usually applied. Because the number of gene that are differentially expressed between samples may still be high (e.g., > 1000), a method to understand and interpret the meaning of so many gene expression changes is needed. One method to solve this problem is termed “gene set enrichment” or GSE (Hung *et al.*, 2012; Subramanian *et al.*, 2005). In this method, a gene or group of genes that belong to a particular category that are enriched in one sample is compared to another sample. For example, if a breast cancer sample has more genes regulated that are annotated to the “cell cycle genes group” than a control sample. Grouping genes can be performed based on their annotation to a number of sources including the Kyoto Encyclopedia Gene Group (KEGG)

pathway. Genes can also be annotated to a group based on a particular encoded protein motif such as F-box binding domain containing genes. Gene ontology (GO) annotations provide another method to annotate and group genes (Ashburner *et al.*, 2000; Schulze-Kremer, 1997). Ontologies are a structured vocabulary with a hierarchy so that genes that belong in a lower level of the vocabulary also belong to higher levels. For example, genes that are annotated with lower level mitochondrion cristae (GO:0030061), are also annotated with the higher level GO term mitochondrion (GO:0005739). Genes can also belong to multiple groups at the same level. Thus, a differentially expressed gene list may be statistically over-represented in a GO category such as rough ER. There are actually 3 GO categories including molecular function, biological process, and cell component.

A gene co-expression network is a group of genes whose level of expression across different samples and conditions for each sample are similar (Gardner *et al.*, 2003). This network identifies similarly behaving genes from the perspective of abundance and infers a common function that can then be hypothesized to work on the same biological process. A good example of a co-expression network are genes that are coregulated during diauxic shift when changing the energy consumption of yeast from glucose to galactose. During this shift, a new set of enzymes are needed and so genes in the galactose metabolism pathway are activated together (DeRisi *et al.*, 1997). Inverse co-expression can also be useful as a network of co-expressed genes that are inversely related include miRNAs and their targets, where in conditions miRNA levels increase, the mRNA targets should be decreased (Ambros, 2004). DG expression has two measures, magnitude and direction. Co-expressed genes can also be identified by using correlation measures (e.g., Pearson Correlation coefficient) between 2 or more genes.

Transcriptomes can be assembled *de novo*, without other information, or based on a reference sequence. The reference sequence can be a genome sequence from the same organism as the source RNA or a closely related species. Reference sequences can also be closely related transcriptome sequences. The number of reads needed to sequence a transcriptome can be determined by the concept of depth. Given that the human transcriptome accounts for 3% of the nucleotides of the human genome consisting of 3 billion nucleotides, sequencing $0.03 \times 3,000,000,000 = 90$ M nucleotides. Thus, sequencing 90 M would be $1 \times$ depth and on average cover each nucleotide once. However, some genes are more highly expressed than others and some genes are rarely expressed, so even $1000 \times$ depth at 90×10^9 would only provide an even chance of sequencing a transcript that is 1 in a thousand in a cell. Therefore, it is likely that many transcriptome assemblies are not complete. Once assembled, several statistical measures reveal the quality and completeness of the assembly. The number and distribution of contigs indicate whether sufficient sequencing has been performed. Since the human transcriptome contains approximately 20,000 genes, a primate transcriptome might be expected to yield the same number. A smaller number might indicate insufficient coverage while a larger number might indicate contigs that need to be joined. Another measure is N50 size, which indicates the median size of a contig, another is the average size of a contig. Since the average size of a gene in humans is approximately 1.5 kb, values in this range suggest a good transcriptome assembly. Following assembly, novel transcriptomes are annotated usually by alignment with known genes from a closely related species. This is performed either by comparison of gene sequences, or translated protein sequences. In both cases, the annotation provided by the alignment can then be grouped to known functional categories of proteins known to be present in all metazoan species. Examples of these complete protein sets include CEGMA or core genes expected to be present in most metazoan species and BUSCO for benchmarking universal single copy orthologs (Parra *et al.*, 2007; Simao *et al.*, 2015). The inclusion and percentage measurement of genes present in these core gene data sets can give a rough idea of the transcriptome completeness.

RNA-seq data can also identify novel transcripts such as fusion genes (Levin *et al.*, 2009). Fusion genes can be produced based on chromosome rearrangements including translocations, inversions, deletions, and duplications. The junction site for each of the rearrangements can potentially give rise to a Fusion gene. These genes are important as an indicator for cytogenetic events, but also some fusion genes can produce protein products that are functional and their activity may have consequences on the organism where it is expressed (Mitelman *et al.*, 2007). Detecting fusion genes is a complex problem since it may be difficult to tell the difference between an authentic fusion gene and a sequencing artifact. More than 20 software are available for fusion gene detection (Kumar *et al.*, 2016). Generally, these software tools look for reads that begin alignment at one gene and continue at another gene in the genome. Filters are used to decrease the number of false positives, however, it is typical to validate identified fusion genes using experimental laboratory tools such as PCR.

Single nucleotide polymorphisms (SNPs) can be detected from RNA-seq data as well by comparison of the obtained read data to a reference genome. This requires a reference genome with identified SNPs such as those from the 1000 genomes project (Quinn *et al.*, 2013). To differentiate true SNPs from false positives, high coverage of the transcript ($> 10 \times$) is usually required. Another strategy is termed "exome sequencing" where only RNA representing exons are sequenced (Ng *et al.*, 2009). The purpose of this approach is to identify any nucleotide changes, both SNPs and other polymorphisms, in protein coding genes that might account for a specific phenotype (Biesecker, 2010). Sequencing the exome is more efficient by about 100 fold than sequencing the entire genome. In addition, some target enrichment strategies are used to sequence only specific exomes but perhaps in a larger number of samples. This type of sample design fits well sequencing platforms built for speed and simplicity rather than throughput.

RNA-Seq/Immunoprecipitation Methods

RNA sequencing (RNA-seq) has become arguably the most powerful approach for characterizing the abundance and structure of the transcriptome. However, variations in how the RNA is isolated, enriched, and purified have made this approach even more useful. Coupled with the modest cost of sequencing RNA via conversion to a cDNA library, RNA-seq will likely yield huge amounts of important biological data for many years to come (Wang *et al.*, 2009). Below are some common variations of RNA-seq.

Biochemically, the approaches can be broken down into those based on size selection (miRNA-seq), selection based upon immunoprecipitation of RNA-proteins or RNA-protein complexes (e.g., RIP-seq, CLIP-seq, GRO-seq), or selection based on modification of RNA.

Cel-seq: Cell sequencing collects RNAs that are sourced from a single cell and is also known as single cell RNA sequencing or scRNA-seq (Hashimshony *et al.*, 2012; Picelli, 2017; Ziegenhain *et al.*, 2017). Single cells may be isolated in situ or manually from cultures. In some experiments, single cells may also be pooled to gather sufficient RNA for sequencing. Single cells can also be separated from whole animal embryos or complex tissues. Once separated, an emerging technique termed drop-seq joins each individual cell with a unique DNA barcoded bead into a nanoliter droplet using microfluidics. Once in a droplet, a library can be synthesized that contains a unique barcode identifier for each individual cell (Macosko *et al.*, 2015).

CLIP-seq, PAR-CLIP-seq, HITS-CLIP-seq: Cross-linking immunoprecipitation sequencing refers to techniques that physically cross-links RNA to protein, usually via UV light, and subsequently immunoprecipitates the product (Licatalosi *et al.*, 2008). In the PAR-CLIP-seq variation, photo-activatable RNA incorporation into RNA provides a more efficient cross-linking solution (Hafner *et al.*, 2010). HITS-CLIP-seq simply refers to High-throughput sequencing of cross-linked immunoprecipitated RNA products.

GRO-seq: Global Run-On sequencing captures nascently transcribed RNA molecules by selecting for BrUTP nucleotides that are incorporated into the RNA strand during transcription (Danko *et al.*, 2015). This method is generally used to identify active transcriptional regulatory elements.

i-CLIP-seq: Individual nucleotide resolution cross-linking immunoprecipitation sequencing refers to an approach to sequence specific nucleotides (Rosbach *et al.*, 2014; Yao *et al.*, 2014). Access to single nucleotides are provided by first selecting via immunoprecipitation and subsequently by selecting RNAs using sequence specific DNA primers to construct the library for sequencing.

miRNA-seq: microRNA sequencing uses size selection of small RNA molecules to enrich for small noncoding RNAs typically mature miRNAs (Luo, 2012).

Ribo-seq: Ribosome sequencing captures RNAs as they are being translated in the ribosome complex (Calviello and Ohler, 2017). The large size of the ribosome allows this approach to be performed using classical biochemical techniques such as ultracentrifuge gradients. This approach has the same goal as TRAP-seq which is to sequence RNAs actively translated.

RIP-seq: RNA Immunoprecipitation sequencing refers to any technique that uses antibody reactivity to a protein-RNA complex in order select RNAs (Zhao *et al.*, 2010).

SHAPE-seq: This is one of several approaches to identify RNA structure motifs via chemical modifications to the RNA at these motifs. The general principle is to chemically probe RNA structural motifs, perform reverse transcription of the RNA which will terminate at the modified site, and then generate cDNA libraries for RNA-sequencing to uncover the specific RNA sites where the motif was located. SHAPE-seq stands for selective-2'-hydroxyl acylation primer extension and follows these principles (Lucks *et al.*, 2011; Mortimer *et al.*, 2012; Watters and Lucks, 2016).

TN-seq: Transposon insertion sequencing refers to an approach to sequence regions where transposons have inserted into the genome (van Opijnen *et al.*, 2009). Transposon insertion is a molecular biology tool to mutate the genome by insertion of DNA sequences into random genomic loci and then uncover phenotypes resulting from insertional mutagenesis. The method relies on massive parallel sequencing of transposon adjacent areas in order to map insertion sites. It is the most commonly used approach of IN-seq or insertion sequencing. This approach has been used with much success in microorganisms.

TRAP-seq: Translating ribosome affinity purification sequencing is an approach to purify RNAs as they are being translated (Reynoso *et al.*, 2015). The method relies on affinity tagging of a ribosome component (e.g., 80S rRNA) and then utilizing immunoprecipitation to pull down associated RNAs. If successful, this provides transcripts that are part of the translatome.

Fig. 2 shows different RNA transcriptome sequencing methods mapped onto the process of transcription.

Data Integration

Data integration in the context of transcriptional informatics refers to the process of combining RNA structure, abundance, or sequence information with other types of biological information. Other types of information can include DNA sequence, epigenetic marks, protein abundance, protein-protein interaction, or metabolomics data. For example, RNA-seq and ChIP-seq data can be integrated (Angelini and Costa, 2014; Klein and Schafer, 2016). A very common data integration combines transcriptomic and proteomic data (Haider and Pal, 2013). Data integration can also occur between different RNA classes. The scale at which integration is performed is typically at the genome, transcriptome, or proteome level, although sub-genome scale integrations are common especially in pathway or biological process studies.

The purpose of data integration is to gain both a broader and a more precise view of RNA dynamics as it relates to other biologically active molecules and the molecular level (Hawkins *et al.*, 2010). Early data integration efforts were aimed at correlating RNA levels with protein levels to confirm aspects of the central dogma. These basic correlations are still performed and have been extended to relate RNA abundance within RNA classes. For example, negative correlations would be expected of miRNAs with their target mRNAs (Camps *et al.*, 2014). Positive correlations would be expected between mRNAs of transcription factors and their targets. More complex correlations can be found by integrating RNA abundance of genes within a biological pathway with products of that pathway. Data integration has been performed for a large group of physiologic processes such during the cell cycle, cell stress, cell death, and energy metabolism.

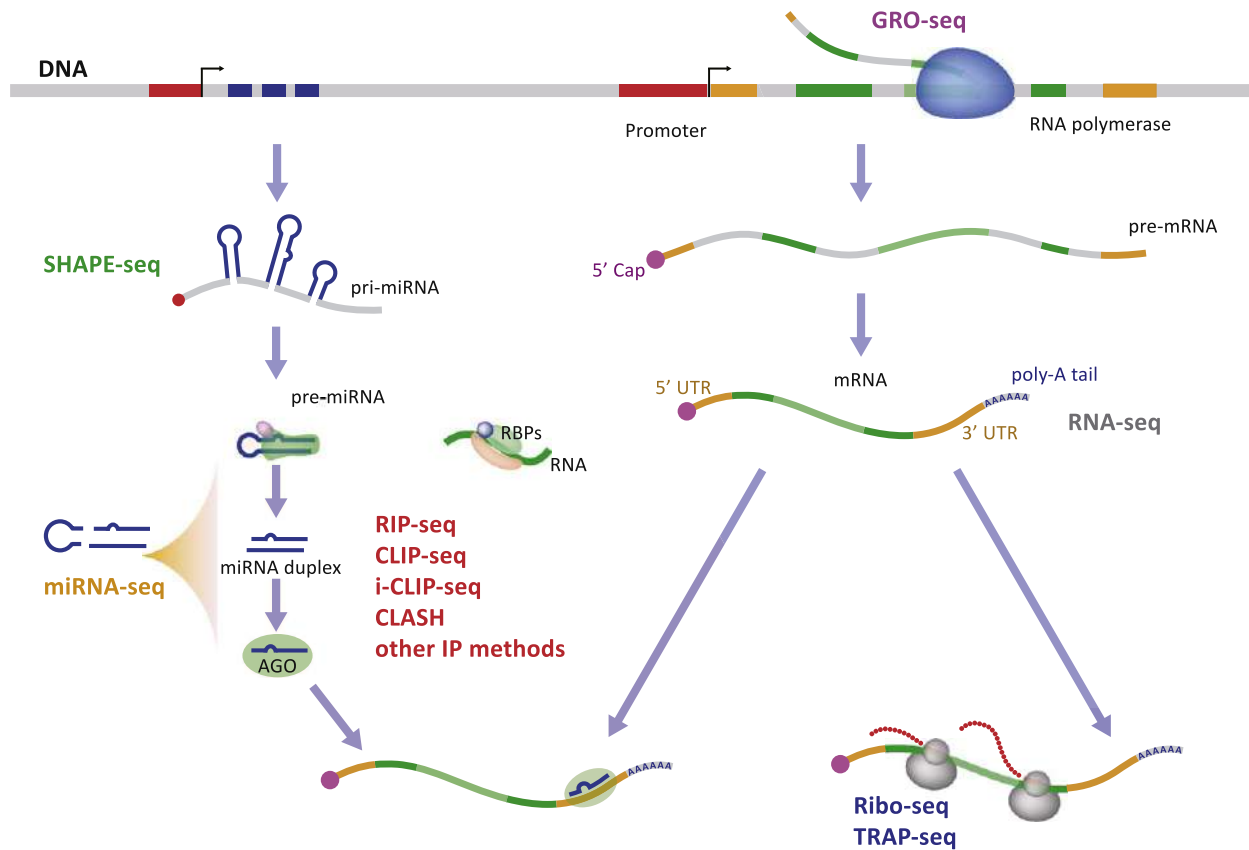


Fig. 2 Targets of RNA transcriptome sequencing methods. The image shows biogenesis of miRNAs and mRNAs. Each sequencing method is marked near the target sequences they focus on. GRO-seq (purple) captures nascent transcripts from actively engaged RNA polymerase. RNA-seq (gray) captures mRNAs (in any status, with or without poly-A tail) as input. Ribo-seq and TRAP-seq (blue) captures the mRNAs which are being translated. SHAPE-seq (green) target RNA with a specific structure motif. miRNA-seq aims to capture the small non coding RNA. RIP-seq, CLIP-seq, i-CLIP-seq, CLASH, and other IP methods (red) aim to sequence RNA bound to RNA-binding proteins (RBPs).

At the whole transcriptome level the integration of abundance data, so called expression RNA or eRNA data with other data has proven to be a powerful tool to identify both important genes in disease and physiological processes. For example, integrating eRNA (RNA abundance data) with SNP data from a single tissue of an individual, using a large population, has made it possible to identify specific DNA sequence variations that influence the expression levels of specific genes or sets of genes including both mRNAs and miRNAs (Lappalainen *et al.*, 2013). These SNPs that influence RNA expression levels have been termed eQTLs for expression quantitative trait loci. Such a strategy has also been used to integrate transcriptomic data with metabolomics data in order to identify genes whose expression levels correlate with the abundance of biologically active small molecules in human blood (Bartel *et al.*, 2015).

Although there have been many exceptionally performed integration studies, data integration in the transcriptomic field remains challenging. One bottleneck is that the biological sources of material should be the same and this is not always possible. DNA from cells is relatively stable and can be isolated thousands of years after the organism has died, however, RNA and protein are extremely labile. Another problem is that for genome scale statistics to be performed, the number of samples needed for sufficient power is large. Finally, for reasons still yet to be understood, data do not always correlate as expected on the genome, transcriptome, and protein level. While this complexity is frustrating, it also presents an opportunity to better understand the relationship between all of these molecules in living systems.

Visualization

The RNA research community is currently experiencing the challenge of rapid growth in volume, variety, and velocity of big data. A key step in understanding and learning from RNA structure, transcriptome expression, RNA-RNA interactions, and other RNA properties is visualization (Shabash and Wiese, 2017). An advanced visualization figure is worth a thousand words.

Visualization of RNA is a critical element in interpreting the structure and function of RNA molecules. At its best, visualization provides a means to analyze and interpret RNA data. As the function of RNA depends on its primary sequence, the structure in 2D

and 3D provides insight into its putative function as well as possible mechanism. Most visualization programs provide 2D views of RNA based on thermodynamic considerations although other factors such as base pairing probabilities are also considered. More recently, chemical probing has been used to determine structures (Siegfried *et al.*, 2014). RNA visualization tools are also becoming more sophisticated as required by the discovery of long noncoding RNAs which also function via their secondary structures but may be several kb in length compared to classical tRNAs which are at least 10 times smaller.

Visualization of transcriptome assemblies, alignments, expression, variations and annotations is an efficient way to reveal the unknown. Visualization of transcriptome data can be obtained in various viewers and portals which plot the abundances of transcripts versus different conditions or samples. The most common and simple way to visualize transcriptome expression data is as a heat map with rows representing genes and columns representing conditions. Integrated viewers can provide sequence, structure, function, and abundance visualization within a single tool (Thorvaldsdottir *et al.*, 2013). These views can also integrate genomic information with transcriptomic data.

Visualization of interactions as a network is one way of coping with such data complexity. Since the RNA information for a single gene is related to the information about others, for example, by being co-expressed in cells, or co-regulated by inducers, RNA networks provide an efficient way to visualize the relationships between different RNA molecules. One example is a ceRNA network where miRNAs are connected to mRNAs, lncRNAs, and circRNAs (Salmena *et al.*, 2011). These networks permit discovery of new knowledge and allows inference to form conclusions. Hubs in these networks can help to identify central RNA molecules in a biological process or specific disease.

Artificial Intelligence Approaches

Artificial intelligence (A.I.) is generally defined as the property of machines that mimic human intelligence as characterized by behaviours such as cognitive ability, memory, learning, and decision making. It is an exploding field that encompasses many applied fields including language translation, self-driving cars, and game playing. The success of A.I. techniques in industry fields, has aroused a great deal of interest among biologists. Modern A.I., such as deep learning, promises to take advantage of large data sets for finding hidden structure within them, and for making accurate predictions (LeCun *et al.*, 2015). In the context of transcriptional informatics, a branch of A.I. that has found abundant applications is machine learning. Given a training set of data, A.I. algorithms can determine characteristics or properties of the data and based on this knowledge can act on new input data in various intelligent ways such as classification, prediction, and performing calculations (Swan *et al.*, 2015). Such an approach has been used, for example, to identify biomarkers during disease processes. Machine learning applied to transcription informatics includes neural network algorithms to cluster gene expression data from microarrays, predict mRNA targets from miRNA sequences, fold RNA molecules to secondary structures, and identify and correct biases in RNA-seq data (Heikkinen *et al.*, 2011; Toronen *et al.*, 1999). The common characteristic of these problems in RNA biology is that the precise answer is not clear, although there may be some basic principles to be followed. A large training set exists that can be used to identify and weigh specific parameters that will provide the best possible way to solve the problem given that set of information. Cross validation via external sample data sets or independent experimental validation have been used to provide confidence and to verify answers provided by A.I.

Tools and Resources: Databases, Browsers, and Portals

Recent advances in NGS technologies have created unprecedented amounts of big RNA-seq data sets for study, and many software and database resources are being developed for assisting biology researchers to deal with RNA-seq datasets. Many analysis tools have been developed for different study purposes, by diverse implementation techniques and in distinct interfaces. Many bioinformatics tools have been implemented to analyze RNA-seq data, including tools for preprocessing, read alignment, and other downstream analysis tools such as transcriptome quantification, differential expression analysis, peak finding etc. **Fig. 3** shows the basic steps of RNA-seq analysis and example tools involved in each step.

Tools for RNA-seq analysis include standalone software, databases, and web services. The Sequence Read Archive (SRA) is a database portal which collects many RNA-seq raw data sets and makes them publically available for new discoveries, meta-analysis, and reanalysis (O'Sullivan *et al.*, 2017). The most basic and essential part of RNA-seq is mapping reads to the reference genome. MAQ, BWA, Bowtie, SOAP, STAR, etc. are the commonly used aligners and many of them are command line tools running on the Linux system (Bao *et al.*, 2011). Visualization tools help view the alignment, such as the stand-alone software Integrative Genomics Viewer (IGV) or the web-based tools Genome Browser (Pavlopoulos *et al.*, 2015). While the basic tools are limited to a single function like a piece from a LEGO building block, workflow tools (including integrated analysis tools, workflow management systems) aim to combine and efficiently use separate tools and resources. Galaxy and Chipster are two outstanding representatives of a workflow management system where the user can choose certain tools from a large tools collection to make their own custom workflow through a web-based user friendly interface (Poplawski *et al.*, 2016). Workflow tools are typically hard to configure, but easy to use. Take Chipster as an example. It starts from a java web start desktop client. The user then chooses the tools from a large catalogue and selects the input, sets parameters, uploads raw data files and clicks a button to submit jobs. The whole process is program-free and combined with advanced visualization can more easily become available to the broader community. Also workflow tools can be deployed in a private cloud or local high performance cluster (HPC) for advanced users.

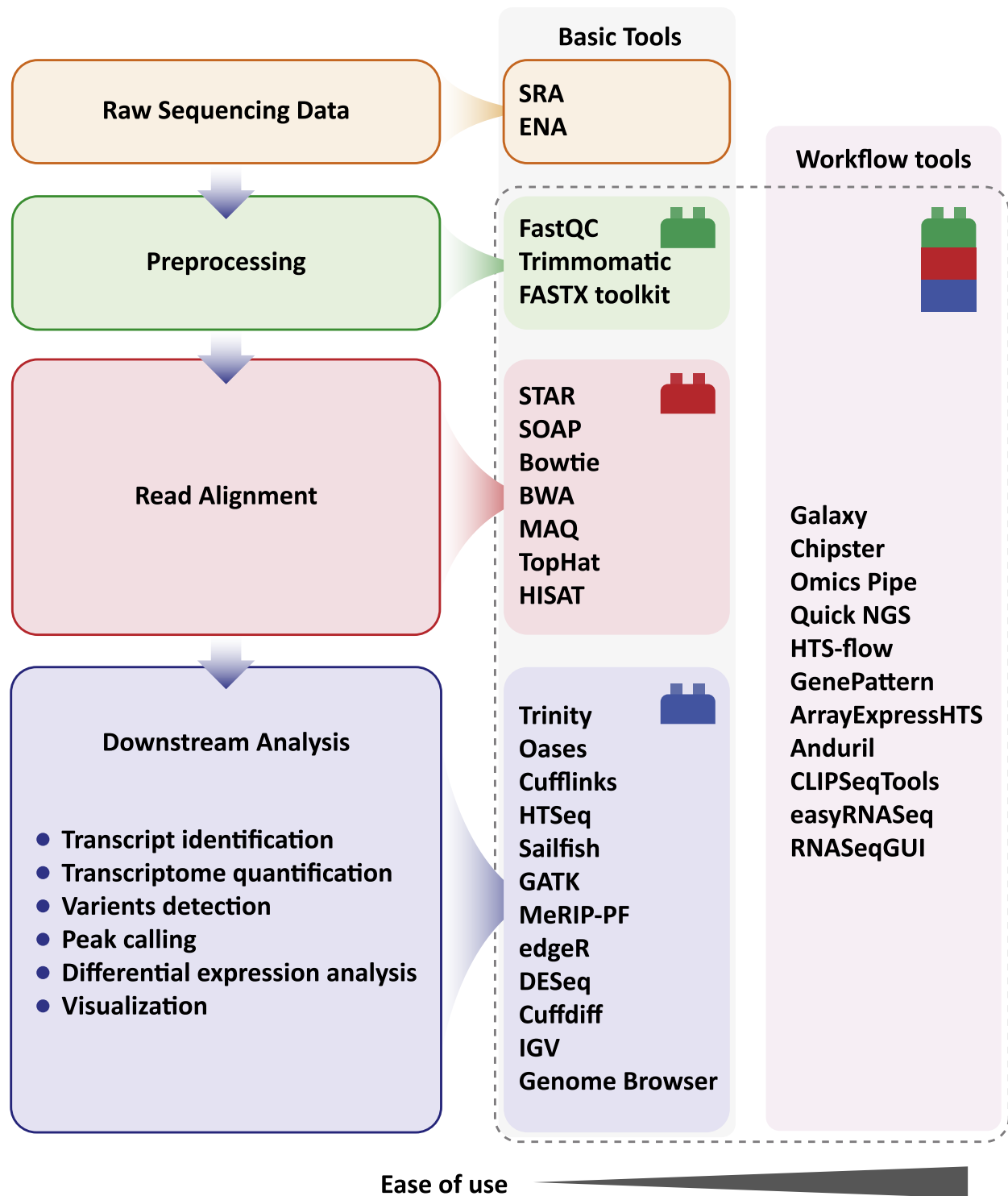


Fig. 3 Basic workflow of RNA-seq analysis and available tools. The left is the basic workflow for NGS data analysis, including raw sequence data (orange), preprocessing (green), read alignment (red), and other downstream analyses (blue). Names of tools are provided that correspond to specific sections in the workflow. In the figure, we only list a small portion of existing tools for each analysis step. All-in-one integrated analysis workflow tools are also being developed to utilize a variety of tools and these are indicated with multiple colored blocks.

Many tools have a unique feature for a particular application. Workflows may report different results although they have the same analysis goals. These differences may be caused by a different combination of tools in the workflow, or different parameter settings. Large scale tools are available, but no gold standard for how to integrate them exists. Due to bandwidth, network speed,

storage space and other hardware limitations, it is still difficult to analyze an RNA-seq dataset through internet based workflow tools only. Even Galaxy, a popular workflow tool, can only handle a small sized data set. To perform batch analysis, a large and private data set still needs to run on the workflow locally and rely on large computational resources. Although tools are diverse and extremely fragmented, biologists or bioinformaticians can nearly always find handy tools for their own study.

Summary

The RNA field is broad. From structure to function to sequence, the vastness from which biological systems have modified, edited, extended, and synthesized RNA is truly astounding. Scientists are only now beginning to grasp the depths and variety of RNAs in living systems. Major advancements in engineering, computer science, and biological understanding have greatly aided RNA transcriptomics as a field of study and research. Indeed, the tailwind created by the decreased cost of sequencing RNA, storing, and processing the information has expanded the field enormously. Yet there is still a sense that hardware and software is just keeping up with the demands of the field. Finally, while many important discoveries in RNA biology have only been reported recently (e.g., RNA interference, long noncoding RNAs, circular RNAs) we wonder what new discoveries await us. As one would think of an encyclopedia as a reference work of factual information, we look forward to when we can add more details at greater accuracy and in abundance. Indeed, it is likely that next iterations of this encyclopedia will change over the years in large and unexpected ways. Such is life in an RNA world.

Acknowledgments

We thank members of the Wong Laboratory Dr. Merja Lakso, Yang Yang, Menglei Zhang, and Changliang Wang for critical reading and discussion. This manuscript would not be possible without the guidance and enthusiasm of Dr. Eija Korpelainen. This work was supported by a grant from the University of Macau MYRG-2016-00101-FHS.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Analyzing Transcriptome-Phenotype Correlations. Bioinformatics Approaches for Studying Alternative Splicing. Characterization of Transcriptional Activities. Characterizing and Functional Assignment of Noncoding RNAs. Codon Usage. Comparative Transcriptomics Analysis. Computational Systems Biology. Computing Languages for Bioinformatics: R. Data Cleaning. Data Mining: Clustering. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Data Reduction. Differential Expression from Microarray and RNA-seq Experiments. Exome Sequencing Data Analysis. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Expression Clustering. Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine. Genome Informatics. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Introduction to Biostatistics. MicroRNA and lncRNA Databases and Analysis. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequence Analysis. Next Generation Sequencing Data Analysis. Pre-Processing: A Data Preparation Step. Regulation of Gene Expression. Sequence Analysis. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation. Transcriptomic Data Normalization. Transcriptomic Databases. Utilising IPG-IEF to Identify Differentially-Expressed Proteins. Whole Genome Sequencing Analysis

References

- Aarnio, V., Lehtonen, M., Storvik, M., *et al.*, 2011. *Caenorhabditis elegans* mutants predict regulation of fatty acids and endocannabinoids by the CYP-35A gene family. *Frontiers in Pharmacology* 2, 12.
- Alberts, B., 2007. Molecular biology of the cell. In: *Artificial Life*, fourth ed., vol. 10. New York, NY: Garland Publishing, pp. 82–95.
- Ambros, V., 2004. The functions of animal micRNAs. *Nature* 431, 350–355.
- Angelini, C., Costa, V., 2014. Understanding gene regulatory mechanisms by integrating ChIP-seq and RNA-seq data: Statistical solutions to biological problems. *Frontiers in Cell and Developmental Biology* 2, 51.
- Aravin, A.A., Naumova, N.M., Tulin, A.V., *et al.*, 2001. Double-stranded RNA-mediated silencing of genomic tandem repeats and transposable elements in the *D. melanogaster* germline. *Current Biology* 11, 1017–1027.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. gene ontology: Tool for the unification of biology. The gene Ontology Consortium. *Nature Genetics* 25, 25–29.
- Asikainen, S., Heikkinen, L., Juhila, J., *et al.*, 2015. Selective microRNA-offset RNA expression in human embryonic stem cells. *PLOS ONE* 10, e0116668.
- Asikainen, S., Heikkinen, L., Wong, G., Storvik, M., 2008. Functional characterization of endogenous siRNA target genes in *Caenorhabditis elegans*. *BMC Genomics* 9, 270.
- Azzalin, C.M., Reichenbach, P., Khoriatuli, L., *et al.*, 2007. Telomeric repeat containing RNA and RNA surveillance factors at mammalian chromosome ends. *Science* 318, 798–801.
- Ballabio, A., Willard, H.F., 1992. Mammalian X-chromosome inactivation and the XIST gene. *Current Opinion in Genetics & Development* 2, 439–447.
- Baltimore, D., 1970. RNA-dependent DNA polymerase in virions of RNA tumour viruses. *Nature* 226, 1209–1211.
- Baltimore, D., 1971. Expression of animal virus genomes. *Bacteriological Reviews* 35, 235–241.
- Bao, S., Jiang, R., Kwan, W., *et al.*, 2011. Evaluation of next-generation sequencing software in mapping and assembly. *Journal of Human Genetics* 56, 406–414.
- Bartel, J., Krumsiek, J., Schramm, K., *et al.*, 2015. The human blood metabolome-transcriptome interface. *PLOS Genetics* 11, e1005274.
- Biesecker, L.G., 2010. Exome sequencing makes medical genomics a reality. *Nature Genetics* 42, 13–14.

- Borchert, G.M., Lanier, W., Davidson, B.L., 2006. RNA polymerase III transcribes human microRNAs. *Nature Structural and Molecular Biology* 13, 1097–1101.
- Budak, H., Bulut, R., Kantar, M., Alptekin, B., 2016. MicroRNA nomenclature and the need for a revised naming prescription. *Briefings in Functional Genomics* 15, 65–71.
- Calabrese, J.M., Sun, W., Song, L., *et al.*, 2012. Site-specific silencing of regulatory elements as a mechanism of X inactivation. *Cell* 151, 951–963.
- Calviello, L., Ohler, U., 2017. Beyond read-counts: Ribo-seq data analysis to understand the functions of the transcriptome. *Trends in Genetics* 33, 728–744.
- Camps, C., Saini, H.K., Mole, D.R., *et al.*, 2014. Integrated analysis of microRNA and mRNA expression and association with HIF binding reveals the complexity of microRNA expression regulation under hypoxia. *Molecular Cancer* 13, 28.
- Chi, K.R., 2017. The RNA code comes into focus. *Nature* 542, 503–506.
- Chomczynski, P., Sacchi, N., 1987. Single-step method of RNA isolation by acid guanidinium thiocyanate–phenol–chloroform extraction. *Analytical Biochemistry* 162, 156–159.
- Colgan, D.F., Manley, J.L., 1997. Mechanism and regulation of mRNA polyadenylation. *Genes & Development* 11, 2755–2766.
- Conesa, A., Madrigal, P., Tarazona, S., *et al.*, 2016. A survey of best practices for RNA-seq data analysis. *Genome Biology* 17, 13.
- Crick, F., 1970. Central dogma of molecular biology. *Nature* 227, 561–563.
- Danko, C.G., Hyland, S.L., Core, L.J., *et al.*, 2015. Identification of active transcriptional regulatory elements from GRO-seq data. *Nature Methods* 12, 433–438.
- Das, P.P., Bagijn, M.P., Goldstein, L.D., *et al.*, 2008. Piwi and piRNAs act upstream of an endogenous siRNA pathway to suppress Tc3 transposon mobility in the *Caenorhabditis elegans* germline. *Molecular Cell* 31, 79–90.
- David, L.A., Maurice, C.F., Carmody, R.N., *et al.*, 2014. Diet rapidly and reproducibly alters the human gut microbiome. *Nature* 505, 559–563.
- DeRisi, J.L., Iyer, V.R., Brown, P.O., 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–686.
- Derrien, T., Johnson, R., Bussotti, G., *et al.*, 2012. The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. *Genome Research* 22, 1775–1789.
- Dominissini, D., Moshitch-Moshkovitz, S., Schwartz, S., *et al.*, 2012. Topology of the human and mouse m6A RNA methylomes revealed by m6A-seq. *Nature* 485, 201–206.
- Doyle, J.J., Gaut, B.S., 2000. Evolution of genes and taxa: A primer. *Plant Molecular Biology* 42, 1–23.
- Dvir, A., 2002. Promoter escape by RNA polymerase II. *Biochimica et Biophysica Acta* 1577, 208–223.
- Ebbesen, K.K., Kjems, J., Hansen, T.B., 2016. Circular RNAs: Identification, biogenesis and function. *Biochimica et Biophysica Acta* 1859, 163–168.
- Eick, D., Wedel, A., Heumann, H., 1994. From initiation to elongation: Comparison of transcription by prokaryotic and eukaryotic RNA polymerases. *Trends in Genetics* 10, 292–296.
- Enderle, D., Spiel, A., Coticchia, C.M., *et al.*, 2015. Characterization of RNA from exosomes and other extracellular vesicles isolated by a novel spin column-based method. *PLOS ONE* 10, e0136133.
- Espinosa, J.M., 2016. Revisiting lncRNAs: How do you know yours is not an eRNA? *Molecular Cell* 62, 1–2.
- Fabrega, C., Hausmann, S., Shen, V., *et al.*, 2004. Structure and mechanism of mRNA cap (guanine-N7) methyltransferase. *Molecular Cell* 13, 77–89.
- Fouqueau, T., Werner, F., 2017. The architecture of transcription elongation. *Science* 357, 871–872.
- Fox, E.J., Reid-Bayliss, K.S., Emond, M.J., Loeb, L.A., 2014. Accuracy of next generation sequencing platforms. *Next Generation, Sequencing & Applications* 1.
- Frye, M., Jaffrey, S.R., Pan, T., *et al.*, 2016. RNA modifications: What have we learned and where are we headed? *Nature Reviews. Genetics* 17, 365–372.
- Gahan, P.B., 2015. *Circulating Nucleic Acids in Early Diagnosis, Prognosis and Treatment Monitoring*. Springer.
- Gardner, T.S., di Bernardo, D., Lorenz, D., Collins, J.J., 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301, 102–105.
- Gelbart, W.M., Crosby, M., Matthews, B., *et al.*, 1997. FlyBase: A Drosophila database. The FlyBase consortium. *Nucleic Acids Research* 25, 63–66.
- Gray, K.A., Yates, B., Seal, R.L., *et al.*, 2015. Genenames.org: The HGNC resources in 2015. *Nucleic Acids Research* 43, D1079–D1085.
- Greider, C.W., Blackburn, E.H., 1989. A telomeric sequence in the RNA of Tetrahymena telomerase required for telomere repeat synthesis. *Nature* 337, 331–337.
- Griffiths-Jones, S., Grocock, R.J., van Dongen, S., *et al.*, 2006. MiRBase: Microna sequences, targets and gene nomenclature. *Nucleic Acids Research* 34, D140–D144.
- Guerrier-Takada, C., Gardiner, K., Marsh, T., *et al.*, 1983. The RNA moiety of ribonuclease P is the catalytic subunit of the enzyme. *Cell* 35, 849–857.
- Ha, M., Kim, V.N., 2014. Regulation of microRNA biogenesis. *Nature Reviews Molecular Cell Biology* 15, 509–524.
- Hafner, M., Landthaler, M., Burger, L., *et al.*, 2010. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell* 141, 129–141.
- Haider, S., Pal, R., 2013. Integrated analysis of transcriptomic and proteomic data. *Current Genomics* 14, 91–110.
- Hangauer, M.J., Vaughn, I.W., McManus, M.T., 2013. Pervasive transcription of the human genome produces thousands of previously unidentified long intergenic noncoding RNAs. *PLOS Genetics* 9, e1003569.
- Hansen, T.B., Jensen, T.I., Clausen, B.H., *et al.*, 2013. Natural RNA circles function as efficient microRNA sponges. *Nature* 495, 384–388.
- Hashimshony, T., Wagner, F., Sher, N., Yanai, I., 2012. CEL-Seq: Single-cell RNA-Seq by multiplexed linear amplification. *Cell Reports* 2, 666–673.
- Hawkins, R.D., Hon, G.C., Ren, B., 2010. Next-generation genomics: An integrative approach. *Nature Reviews Genetics* 11, 476–486.
- He, L., Hannon, G.J., 2004. MicroRNAs: Small RNAs with a big role in gene regulation. *Nature Reviews Genetics* 5, 522–531.
- Heikkinen, L., Kolehmainen, M., Wong, G., 2011. Prediction of microRNA targets in *Caenorhabditis elegans* using a self-organizing map. *Bioinformatics* 27, 1247–1254.
- Helm, M., Motorin, Y., 2017. Detecting RNA modifications in the transcriptome: Predict and validate. *Nature Reviews. Genetics* 18, 275–291.
- Huang, X., Yuan, T., Tschannen, M., *et al.*, 2013. Characterization of human plasma-derived exosomal RNAs by deep sequencing. *BMC Genomics* 14, 319.
- Hubbard, T., Barker, D., Birney, E., *et al.*, 2002. The Ensembl genome database project. *Nucleic Acids Research* 30, 38–41.
- Hugenholtz, P., Goebel, B.M., Pace, N.R., 1998. Impact of culture-independent studies on the emerging phylogenetic view of bacterial diversity. *Journal of Bacteriology* 180, 4765–4774.
- Hung, J.H., Yang, T.H., Hu, Z., *et al.*, 2012. Gene set enrichment analysis: Performance evaluation and usage guidelines. *Briefings in Bioinformatics* 13, 281–291.
- Iwasaki, Y.W., Siomi, M.C., Siomi, H., 2015. PIWI-interacting RNA: Its biogenesis and functions. *Annual Review of Biochemistry* 84, 405–433.
- Iyer, M.K., Niknafs, Y.S., Malik, R., *et al.*, 2015. The landscape of long noncoding RNAs in the human transcriptome. *Nature Genetics* 47, 199–208.
- Jacob, F., Monod, J., 1961. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology* 3, 318–356.
- Jain, M., Olsen, H.E., Paten, B., Akeson, M., 2016. The Oxford nanopore MinION: Delivery of nanopore sequencing to the genomics community. *Genome Biology* 17, 239.
- Johnstone, R.M., Adam, M., Hammond, J.R., *et al.*, 1987. Vesicle formation during reticulocyte maturation. Association of plasma membrane activities with released vesicles (exosomes). *Journal of Biological Chemistry* 262, 9412–9420.
- Juven-Gershon, T., Kadonaga, J.T., 2010. Regulation of gene expression via the core promoter and the basal transcriptional machinery. *Developmental Biology* 339, 225–229.
- Kaiser, C.A., Krieger, M., Lodish, H., Berk, A., 2007. *Molecular Cell Biology*. WH Freeman.
- Kapranov, P., Cheng, J., Dike, S., *et al.*, 2007. RNA maps reveal new RNA classes and a possible function for pervasive transcription. *Science* 316, 1484–1488.
- Kay, L.E., 2000. *Who Wrote the Book of Life? A History of the Genetic Code*. Stanford University Press.
- Keller, S., Rupp, C., Stoeck, A., *et al.*, 2007. CD24 is a marker of exosomes secreted into urine and amniotic fluid. *Kidney International* 72, 1095–1102.
- Kent, W.J., Sugnet, C.W., Furey, T.S., *et al.*, 2002. The human genome browser at UCSC. *Genome Research* 12, 996–1006.
- Kim, T.K., Hemberg, M., Gray, J.M., *et al.*, 2010. Widespread transcription at neuronal activity-regulated enhancers. *Nature* 465, 182–187.
- Kiss, T., 2001. Small nucleolar RNA-guided post-transcriptional modification of cellular RNAs. *EMBO Journal* 20, 3617–3622.
- Klein, H.U., Schafer, M., 2016. Integrative analysis of histone ChIP-seq and RNA-seq data. *Current Protocols in Human Genetics*, 90. pp. 20 23 21–20 23 16.
- Korpelainen, E., Tuimala, J., Somervuo, P., *et al.*, 2014. RNA-seq Data Analysis: A Practical Approach. CRC Press.
- Kowalczyk, M.S., Hughes, J.R., Garrick, D., *et al.*, 2012. Intragenic enhancers act as alternative promoters. *Molecular Cell* 45, 447–458.

- Kruger, K., Grabowski, P.J., Zaug, A.J., *et al.*, 1982. Self-splicing RNA: Autoexcision and autocyclization of the ribosomal RNA intervening sequence of tetrahymena. *Cell* 31, 147–157.
- Kumar, S., Vo, A.D., Qin, F., Li, H., 2016. Comparative assessment of methods for the fusion transcripts detection from RNA-Seq data. *Scientific Reports* 6, 21597.
- Kung, J.T., Colognori, D., Lee, J.T., 2013. Long noncoding RNAs: Past, present, and future. *Genetics*, 193. pp. 651–669.
- Laine, M.M., Pasanen, T., Saarela, J. *et al.*, 2014. DNA Microarray Data Analysis, second edition.
- Langenberger, D., Bermudez-Santana, C., Hertel, J., *et al.*, 2009. Evidence for human microRNA-offset RNAs in small RNA sequencing data. *Bioinformatics* 25, 2298–2301.
- Lappalainen, T., Sammeth, M., Friedlander, M.R., *et al.*, 2013. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* 501, 506–511.
- Lasda, E., Parker, R., 2014. Circular RNAs: Diversity of form and function. *RNA (New York, NY)* 20, 1829–1842.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444.
- Lee, R.C., Feinbaum, R.L., Ambros, V., 1993. The *C. elegans* heterochronic gene *lin-4* encodes small RNAs with antisense complementarity to *lin-14*. *Cell* 75, 843–854.
- Lee, Y., Kim, M., Han, J., *et al.*, 2004. MicroRNA genes are transcribed by RNA polymerase II. *EMBO Journal* 23, 4051–4060.
- Levin, J.Z., Berger, M.F., Adiconis, X., *et al.*, 2009. Targeted next-generation sequencing of a cancer transcriptome enhances detection of sequence variants and novel fusion transcripts. *Genome Biology* 10, R115.
- Li, M., Zeringer, E., Barta, T., *et al.*, 2014a. Analysis of the RNA content of the exosomes derived from blood serum and urine and its potential as biomarkers. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 369.
- Li, S., Tighe, S.W., Nicolet, C.M., *et al.*, 2014b. Multi-platform assessment of transcriptome profiling using RNA-seq in the ABRF next-generation sequencing study. *Nature Biotechnology* 32, 915–925.
- Li, X., Xiong, X., Yi, C., 2016. Epitranscriptome sequencing technologies: Decoding RNA modifications. *Nature Methods* 14, 23–31.
- Licalosi, D.D., Mele, A., Fak, J.J., *et al.*, 2008. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature* 456, 464–469.
- Liu, L., Li, Y., Li, S., *et al.*, 2012. Comparison of next-generation sequencing systems. *Journal of Biomedicine & Biotechnology* 2012, 251364.
- Lodish, H., Berk, A., Zipursky, S.L., *et al.*, 1995. *Molecular Cell Biology*. New York, NY: Scientific American Books.
- Londin, E., Lohar, P., Telonis, A.G., *et al.*, 2015. Analysis of 13 cell types reveals evidence for the expression of numerous novel primate- and tissue-specific microRNAs. *Proceedings of the National Academy of Sciences of the United States of America* 112, E1106–E1115.
- Lucks, J.B., Mortimer, S.A., Trapnell, C., *et al.*, 2011. Multiplexed RNA structure characterization with selective 2'-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). *Proceedings of the National Academy of Sciences of the United States of America* 108, 11063–11068.
- Luo, S., 2012. MicroRNA expression analysis using the Illumina microRNA-seq platform. *Methods in Molecular Biology* 822, 183–188.
- Macosko, E.Z., Basu, A., Satija, R., *et al.*, 2015. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* 161, 1202–1214.
- Mardis, E.R., 2017. DNA sequencing technologies: 2006–2016. *Nature Protocols* 12, 213–218.
- Marioni, J.C., Mason, C.E., Mane, S.M., *et al.*, 2008. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research* 18, 1509–1517.
- Martin, J.A., Wang, Z., 2011. Next-generation transcriptome assembly. *Nature Reviews. Genetics* 12, 671–682.
- Matera, A.G., Terns, R.M., Terns, M.P., 2007. Non-coding RNAs: Lessons from the small nuclear and small nucleolar RNAs. *Nature Reviews Molecular Cell Biology* 8, 209–220.
- Meister, G., 2013. Argonaute proteins: Functional insights and emerging roles. *Nature Reviews Genetics* 14, 447–459.
- Memczak, S., Jens, M., Elefsinioti, A., *et al.*, 2013. Circular RNAs are a large class of animal RNAs with regulatory potency. *Nature* 495, 333–338.
- Merriman, B., Rothberg, J.M., 2012. Progress in Ion Torrent semiconductor chip based sequencing. *Electrophoresis* 33, 3397–3417.
- Miao, Z., Westhof, E., 2017. RNA structure: Advances and assessment of 3D structure prediction. *Annual Review of Biophysics* 46, 483–503.
- Mikulowska-Mennis, A., Taylor, T.B., Vishnu, P., *et al.*, 2002. High-quality RNA from cells isolated by laser capture microdissection. *BioTechniques* 33, 176–179.
- Mitelman, F., Johansson, B., Mertens, F., 2007. The impact of translocations and gene fusions on cancer causation. *Nature Reviews Cancer* 7, 233–245.
- Morrison, M.R., Brodeur, R., Pardue, S., *et al.*, 1979. Differences in the distribution of poly(A) size classes in individual messenger RNAs from neuroblastoma cells. *Journal of Biological Chemistry* 254, 7675–7683.
- Mortazavi, A., Williams, B.A., McCue, K., *et al.*, 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods* 5, 621–628.
- Mortimer, S.A., Trapnell, C., Aviran, S., *et al.*, 2012. SHAPE-Seq: High-throughput RNA structure analysis. *Current Protocols in Chemical Biology* 4, 275–297.
- Murakami, K.S., Masuda, S., Campbell, E.A., *et al.*, 2002a. Structural basis of transcription initiation: An RNA polymerase holoenzyme–DNA complex. *Science* 296, 1285–1290.
- Murakami, K.S., Masuda, S., Darst, S.A., 2002b. Structural basis of transcription initiation: RNA polymerase holoenzyme at 4 Å resolution. *Science* 296, 1280–1284.
- Natoli, G., Andrau, J.C., 2012. Noncoding transcription at enhancers: General principles and functional models. *Annual Review of Genetics* 46, 1–19.
- Neveu, M., Kim, H.J., Benner, S.A., 2013. The “strong” RNA world hypothesis: Fifty years old. *Astrobiology* 13, 391–403.
- Ng, S.B., Turner, E.H., Robertson, P.D., *et al.*, 2009. Targeted capture and massively parallel sequencing of 12 human exomes. *Nature* 461, 272–276.
- Nilsen, T.W., 2008. Endo-siRNAs: Yet another layer of complexity in RNA silencing. *Nature Structural and Molecular Biology* 15, 546–548.
- Noller, H.F., 1991. Ribosomal RNA and translation. *Annual Review of Biochemistry* 60, 191–227.
- Okamura, K., Lai, E.C., 2008. Endogenous small interfering RNAs in animals. *Nature Reviews Molecular Cell Biology* 9, 673–678.
- O'Sullivan, C., Busby, B., Mizrahi, I.K., 2017. Managing sequence data. *Methods in Molecular Biology (Clifton, NJ)* 1525, 79–106.
- Palade, G.E., 1955. A small particulate component of the cytoplasm. *Journal of Biophysical and Biochemical Cytology* 1, 59–68.
- Paralkar, V.R., Taborda, C.C., Huang, P., *et al.*, 2016. Unlinking an lncRNA from its associated cis element. *Molecular Cell* 62, 104–110.
- Parra, G., Bradnam, K., Korf, I., 2007. CEGMA: A pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics* 23, 1061–1067.
- Pavlopoulos, G.A., Malliarakis, D., Papanikolaou, N., *et al.*, 2015. Visualizing genome and systems biology: Technologies, tools, implementation techniques and trends, past, present and future. *GigaScience* 4, 38.
- Picelli, S., 2017. Single-cell RNA-sequencing: The future of genome biology is now. *RNA Biology* 14, 637–650.
- Poplawski, A., Marini, F., Hess, M., *et al.*, 2016. Systematically evaluating interfaces for RNA-seq analysis from a life scientist perspective. *Briefings in Bioinformatics* 17, 213–223.
- Povey, S., Lovering, R., Bruford, E., *et al.*, 2001. The HUGO gene nomenclature Committee (HGNC). *Human Genetics* 109, 678–680.
- Probst, J., Brechtel, S., Scheel, B., *et al.*, 2006. Characterization of the ribonuclease activity on the skin surface. *Genetic Vaccines and Therapy* 4, 4.
- Ptashne, M., Gann, A., 1997. Transcriptional activation by recruitment. *Nature* 386, 569–577.
- Pyle, A.M., 1993. Ribozymes: A distinct class of metalloenzymes. *Science* 261, 709–714.
- Quail, M.A., Smith, M., Coupland, P., *et al.*, 2012. A tale of three next generation sequencing platforms: Comparison of ion torrent, Pacific biosciences and Illumina MiSeq sequencers. *BMC Genomics* 13, 341.
- Quinn, E.M., Cormican, P., Kenny, E.M., *et al.*, 2013. Development of strategies for SNP detection in RNA-seq data: Application to lymphoblastoid cell lines and evaluation using 1000 Genomes data. *PLOS One* 8, e58815.
- Reynoso, M.A., Juntawong, P., Lancia, M., *et al.*, 2015. Translating Ribosome Affinity Purification (TRAP) followed by RNA sequencing technology (TRAP-SEQ) for quantitative assessment of plant translomes. *Methods in Molecular Biology* 1284, 185–207.
- Rhoads, A., Au, K.F., 2015. PacBio sequencing and its applications. *Genomics, Proteomics & Bioinformatics* 13, 278–289.
- Richard, P., Manley, J.L., 2009. Transcription termination by nuclear RNA polymerases. *Genes Dev* 23, 1247–1269.
- Rosbach, O., Hung, L.H., Khrameeva, E., *et al.*, 2014. Crosslinking-immunoprecipitation (iCLIP) analysis reveals global regulatory roles of hnRNP L. *RNA Biology* 11, 146–155.

- Sainsbury, S., Bernecky, C., Cramer, P., 2015. Structural basis of transcription initiation by RNA polymerase II. *Nature Reviews. Molecular Cell Biology* 16, 129–143.
- Saito, K., Nishida, K.M., Mori, T., *et al.*, 2006. Specific association of Piwi with rasiRNAs derived from retrotransposon and heterochromatic regions in the *Drosophila* genome. *Genes and Development* 20, 2214–2222.
- Saliba, A.E., Westermann, A.J., Gorski, S.A., Vogel, J., 2014. Single-cell RNA-seq: Advances and future challenges. *Nucleic Acids Research* 42, 8845–8860.
- Salmena, L., Poliseno, L., Tay, Y., *et al.*, 2011. A ceRNA hypothesis: The Rosetta stone of a hidden RNA language? *Cell* 146, 353–358.
- Salzman, J., 2016. Circular RNA expression: Its potential regulation and function. *Trends in Genetics* 32, 309–316.
- Schulze-Kremer, S., 1997. Adding semantics to genome databases: Towards an ontology for molecular biology. *Proceedings. International Conference on Intelligent Systems for Molecular Biology* 5, 272–275.
- Serganov, A., Patel, D.J., 2007. Ribozymes, riboswitches and beyond: Regulation of gene expression without proteins. *Nature Reviews Genetics* 8, 776–790.
- Shabash, B., Wiese, K.C., 2017. RNA Visualization: Relevance and the current state-of-the-art focusing on pseudoknots. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14. pp. 696–712.
- Sharp, S.J., Schaack, J., Cooley, L., *et al.*, 1985. Structure and transcription of eukaryotic tRNA genes. *CRC Critical Reviews in Biochemistry* 19, 107–144.
- Shatkin, A.J., 1976. Capping of eucaryotic mRNAs. *Cell* 9, 645–653.
- Shi, W., Hendrix, D., Levine, M., Haley, B., 2009. A distinct class of small RNAs arises from pre-miRNA-proximal regions in a simple chordate. *Nature Structural and Molecular Biology* 16, 183–189.
- Siegfried, N.A., Busan, S., Rice, G.M., *et al.*, 2014. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nature Methods* 11, 959–965.
- Simao, F.A., Waterhouse, R.M., Ioannidis, P., *et al.*, 2015. BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics* 31, 3210–3212.
- Song, J., Yi, C., 2017. Chemical modifications to RNA: A new layer of gene expression regulation. *ACS Chemical Biology* 12, 316–325.
- Stein, L., Sternberg, P., Durbin, R., *et al.*, 2001. WormBase: Network access to the genome and biology of *Caenorhabditis elegans*. *Nucleic Acids Research* 29, 82–86.
- Stephens, S.B., Dodd, R.D., Lerner, R.S., *et al.*, 2008. Analysis of mRNA partitioning between the cytosol and endoplasmic reticulum compartments of mammalian cells. *Methods in Molecular Biology* 419, 197–214.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* 102, 15545–15550.
- Swan, A.L., Stekel, D.J., Hodgman, C., *et al.*, 2015. A machine learning heuristic to identify biologically relevant and minimal biomarker panels from omics data. *BMC Genomics* 16 (Suppl. 1), S2.
- Tay, Y., Kats, L., Salmena, L., *et al.*, 2011. Coding-independent regulation of the tumor suppressor PTEN by competing endogenous mRNAs. *Cell* 147, 344–357.
- Temin, H.M., Mizutani, S., 1970. RNA-dependent DNA polymerase in virions of Rous sarcoma virus. *Nature* 226, 1211–1213.
- Thorvaldsdottir, H., Robinson, J.T., Mesirov, J.P., 2013. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Briefings in Bioinformatics* 14, 178–192.
- Toronen, P., Kolehmainen, M., Wong, G., Castren, E., 1999. Analysis of gene expression data using self-organizing maps. *FEBS Letters* 451, 142–146.
- Vagin, V.V., Sigova, A., Li, C., *et al.*, 2006. A distinct small RNA pathway silences selfish genetic elements in the germline. *Science* 313, 320–324.
- Valadi, H., Ekstrom, K., Bossios, A., *et al.*, 2007. Exosome-mediated transfer of mRNAs and microRNAs is a novel mechanism of genetic exchange between cells. *Nature Cell Biology* 9, 654–659.
- van Dijk, E.L., Auger, H., Jaszczyszyn, Y., Thermes, C., 2014. Ten years of next-generation sequencing technology. *Trends in Genetics* 30, 418–426.
- van Opijnen, T., Bodi, K.L., Camilli, A., 2009. Tn-seq: High-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms. *Nature Methods* 6, 767–772.
- Vora, N.L., Smeester, L., Boggess, K., Fry, R.C., 2017. Investigating the role of fetal gene expression in preterm birth. *Reproductive Sciences (Thousand Oaks, CA)* 24, 824–828.
- Wang, Z., Gerstein, M., Snyder, M., 2009. RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews Genetics* 10, 57–63.
- Watters, K.E., Lucks, J.B., 2016. Mapping RNA structure in vitro with SHAPE chemistry and next-generation sequencing (SHAPE-Seq). *Methods in Molecular Biology (Clifton, NJ)* 1490, 135–162.
- Wilson, D.N., Doudna Cate, J.H., 2012. The structure and function of the eukaryotic ribosome. *Cold Spring Harbor Perspectives in Biology* 4.
- Woese, C.R., 1987. Bacterial evolution. *Microbiological Reviews* 51, 221–271.
- Yao, C., Weng, L., Shi, Y., 2014. Global protein-RNA interaction mapping at single nucleotide resolution by iCLIP-seq. *Methods in Molecular Biology (Clifton, NJ)* 1126, 399–410.
- Yuan, S., Schuster, A., Tang, C., *et al.*, 2016. Sperm-borne miRNAs and endo-siRNAs are important for fertilization and preimplantation embryonic development. *Development (Cambridge, England)* 143, 635–647.
- Zhang, H., 2016. Overview of sequence data formats. *Methods in Molecular Biology (Clifton, NJ)* 1418, 3–17.
- Zhao, J., Ohsumi, T.K., Kung, J.T., *et al.*, 2010. Genome-wide identification of polycomb-associated RNAs by RIP-seq. *Molecular Cell* 40, 939–953.
- Ziegenhain, C., Vieth, B., Parekh, S., *et al.*, 2017. Comparative analysis of single-cell RNA sequencing methods. *Molecular Cell* 65 (631–643), e634.

Further Reading

- Asikainen, S., Heikkinen, L., Juhila, J., *et al.*, 2015. Selective microRNA-Offset RNA expression in human embryonic stem cells. *PLOS One* 10, e0116668.
- Korpelainen, E., Tuimala, J., Somervuo, P., Huss, M., Wong, G., 2014. *RNA-seq Data Analysis: A Practical Approach*. New York: CRC Press.
- Quail, M.A., Smith, M., Coupland, P., *et al.*, 2012. A tale of three next generation sequencing platforms: Comparison of ion torrent, Pacific biosciences and Illumina MiSeq sequencers. *BMC Genomics* 13, 341.

Relevant Websites

- <https://github.com/samtools/hts-specs>
GitHub.
- <https://nanoporetech.com/about-us/news/highlights-clive-g-browns-technical-update>
Oxford Nanopore Technologies.
- http://www.pacb.com/wp-content/uploads/2015/09/PacBio_RS_II_Brochure.pdf
Pacific Biosciences.
- <http://simpsonlab.github.io/2016/08/23/R9/>
Simpson Lab.

Transcriptomic Databases

Askhat Molkenov, Altyn Zhelambayeva, Akbar Yermekov, Saule Mussurova, Aliya Sarkytbayeva, Yerbol Kalykhbergenov, Zhaxybay Zhumadilov, and Ulykbek Kairov, Nazarbayev University, Astana, Kazakhstan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Since the late 90s, the rapid development of the high-throughput omics technologies has led to a generation of the available transcriptomic data that demands effective solutions for organization, storage and further progress of new analysis tools. Database platforms have therefore been extensively developed to address these needs. Data repositories containing both curated and user-submitted entries are vital sources of knowledge for both researchers across various fields such as bioinformaticians, biologists, and clinicians for data analysis and for elucidating numerous processes within a disease or its development stages. As the development of omics technologies and bioinformatics continues, possibilities for data integration would allow inferring many more conclusions that could be crucial for a variety of problems arising from fundamental researches or practical applications in biomedicine and agriculture. The development and appearance of transcriptomic databases, in particular, can be useful to prevent pharmacological disease and even increase yield by determination of biomarkers or potential effector genes in cancer and plant pathogens, as well as identification of favorable alleles from the crop landraces.

We have conducted a Pubmed publications analysis for the occurrences of the terms “database” and “transcriptome” in both the title and the abstract of a publication to infer the “chronology” of this development. Thus, the first occurrence of the term “database” in a publication’s title is dated as early as 1974; “transcriptome” made the first appearance much later in 1998 and both terms were used together in both the title and the abstract of a publication for the first time in 1999.

As Fig. 1 illustrates, a log10 scale increase trend in the terms’ use showed a corresponding growth of interest in the omics research, however, even nowadays, the occurrences of both “transcriptome” and “database” terms in the titles and abstract reflect, to the best of our knowledge, an absence of publications that could provide comprehensive overviews of the currently available transcriptomic resources. Thus, the aim of this review is to provide an overview of the current state of transcriptomic databases as broadly divided into the categories of human, animal and plant sources. A short summary of the database types will be provided for each category.

Human Transcriptome Databases

The human databases are mostly disease-specific sources with a particular focus on cancer data in addition to tissue-, interaction- and development-specific data platforms. Other human data sources include circRNA/ncRNA repositories, platforms containing raw full transcriptome data with the SNP information, the sources containing transcriptional signatures from selected experiments in the GEO databases, as well as functional databases.

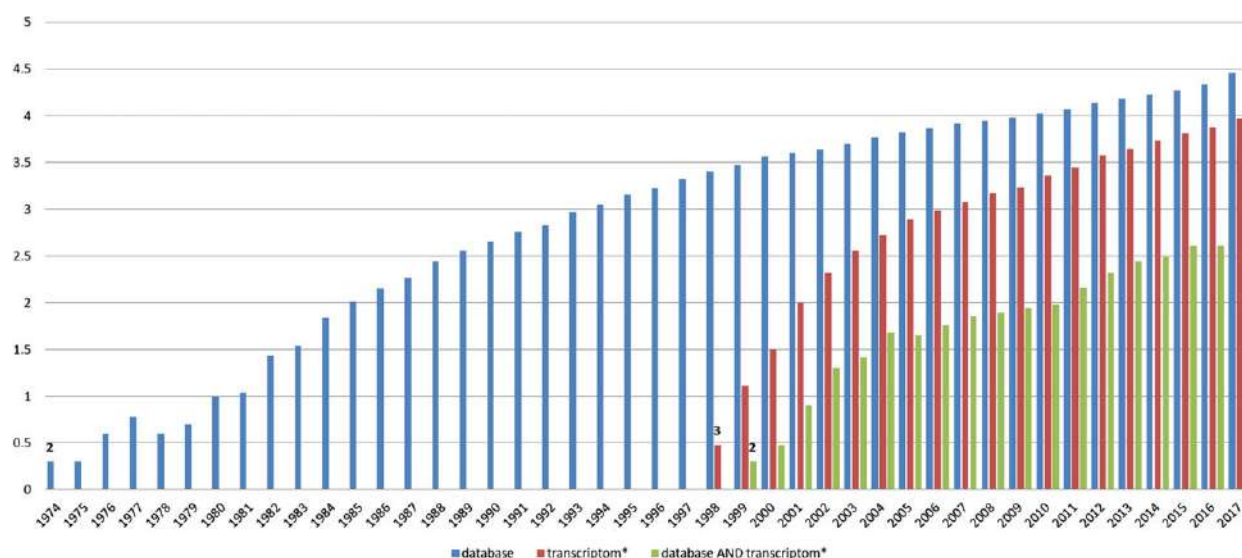


Fig. 1 Appearances of the terms “database” and “transcriptome” in the titles and abstracts of the available PubMed publications (as of March 2018); the number of publications is presented on a log10 scale.

Multiple Cancer Sources, Single Cancer-Specific Sources and an RNA-Specific Cancer Database

The largest source of cancer data estimated at 2.5 petabytes is represented by TCGA (Chang *et al.*, 2013) (The Cancer Genome Atlas). TCGA is a result of a concentrated effort between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) with 20 collaborating institutions to present the range of genetic changes and comparison between the samples and with the normal tissue profiles for 23 most common types of cancer in addition to 10 rare cancers. TCGA provides information on molecular, DNA, protein and epigenetic aberrations resulting in a better understanding of cancer subtypes and higher precision of treatment options as the types of cancer chosen for study have a poor prognosis and result in a major detrimental public health impact (e.g., pancreatic ductal adenocarcinoma, acute myeloid leukemia, cutaneous melanoma). As of now, the data is compiled from 11,000 patients and includes whole genome and exome profiles, mRNA/microRNA sequences, copy number and methylation profiles as well as protein activation status information. Clinical data and data regarding the sample (e.g., weight) are available for some of the sample entries.

Another major data source which contains cancer data in addition GEO-NCBI (the Gene Expression Omnibus) (Barrett *et al.*, 2012) is a public submission data repository containing high-throughput sequencing and microarray data that was created and is under maintenance of the National Center for Biotechnology Information (NCBI). The submitted data comes from the scientific communities worldwide resulting in MIAME (Minimum Information About a Microarray Experiment)-compliant data entries representing more than 1 min samples and 1600 organisms (humans, animals, plants, bacteria and fungi) with links to 20,000 related publications, 90% of the data coming from gene expression studies characterizing such processes as development, metabolism and disease. The database is a user-friendly archival resource that has a number of useful options such as “similar studies” and “find pathways” (i.e., gene mapping to a list of frequency weighted pathways in NCBI BioSystems database) search options, repository browser and GEO2R web app which allows to perform differential gene expression analysis on more than 90% of GEO entries. The data is divided into 3 categories of Platform (array/sequencer description), Sample (biological materials/experimental protocol descriptions) and Series (a link between a group of related Sample and the point of the whole study). GEO DataSet compiles the comparable Samples into a consistently processed format that is divided into categories based on experimental variables and has a description of the study. Thus, a search for term “cancer” in GEO Datasets (all fields) results in 646,300 entries with 1272 DataSets and 616,885 Samples (as of March 2018). The cancer information is available for both humans, animals (e.g., mouse and zebrafish) and constructs (e.g., studies on protein expression data from using viral constructs and cancer xenografts); a number of DataSets contain expression information on metastatic cancers.

Another database which includes transcriptome information on multiple types of cancer is BioXpress (Wan *et al.*, 2015). BioXpress is a database of gene expression and cancer association. It includes gene expression data from 64 cancer types from 6361 patients. It also provides 17,469 genes with 9513 of them showing differential gene expression between normal and tumor samples. To facilitate pan-cancer analysis, all cancer types are mapped to Disease Ontology terms. BioXpress database includes mappings of expression levels to genes using RNA-seq data extracted from The Cancer Genome Atlas, International Cancer Genome Consortium, Expression Atlas. Expression levels are also obtained from publications of large-scale studies.

CoReCG, HNdb and curatedOvarianData databases are cancer-specific and discussed below. CoReCG database contains a comprehensive review of previous studies on colon-rectal cancer (Agarwal *et al.*, 2016). They collected the information about the genes associated with colon-rectal cancer from high-quality published literature and organized this data into the CoReCG database. CoReCG contains 2056 cancer genes annotation involved in distinct stages of colon-rectal cancer. CoReCG database contains comprehensive information about genes such as locations in chromosomes, homology, ontology, SNP information, genomic coordinates, mRNA, information about proteins, etc.

The Head and Neck (HNdb) database (Henrique *et al.*, 2016) was created to provide a comprehensive resource of the Head and Neck squamous cell carcinoma genes (HNSCC) and associated proteins. The retrieved previously performed studies from PubMed database and developed a methodology for literature mining to identify genes associated with HNSCC. The information about the gene can be searched by the name of the gene or its associated protein, by gene symbol, aliases. Also, the information about chromosome regions and gene associated with HNSCC can also be obtained. The database also includes the information about gene ontologies, metabolic pathways and links for comparison of the expression features in normal tissues. The protein page includes the proteins' 3D structures, posttranslational modifications, metabolite and protein-protein interaction as well as drug targets.

Transcriptomic database of ovarian cancer (Ganzfried *et al.*, 2013) was implemented as the *curatedOvarianData* Bioconductor package in R. It contains microarray data of 2970 patients from 23 studies with curated and documented clinical metadata. They pre-processed available Affymetrix microarray data by the same normalization algorithm. For the Affymetrix U133A and U133 Plus 2.0, they used fRMA normalizing algorithm. Then, the analysis of expression levels of *CXCL12* was performed in all primary tumor samples included in the database. The samples were centered by their means and divided by their standard deviation to ensure that expression levels are compared on the same scale. They confirmed the hypothesis that poor outcome in 2108 patients from 13 independent studies with different stages and histologies was associated with upregulation of *CXCL12*. They emphasized the importance of biomarker validation in sufficiently large collections of data.

Cancer RNA-seq Nexus (Li *et al.*, 2016) is the first publicly available RNA-seq database that contains comprehensive data analyses and archives for transcript profiles of coding sequences, long non-coding RNA (lncRNA), coexpression networks, visualization of differential gene expression profiles and phenotype-based data searching and organization. It contains in total 54 human cancer RNA-seq datasets for 11,030 samples and 326 phenotype-specific subsets where each subset represents the group of samples under specific cancer type and stage conditions. mRNA-lncRNA coexpression networks can be visualized, compared and

analyzed for different pairs of phenotype-specific subsets. Coding transcript/lnc expression profiles can also be visualized. Its functionality and user-friendly interface makes Cancer RNA-seq Nexus serve as a handful tool for personalized medicine research.

Other Disease-Specific Sources: Kidney Diseases, Parkinson's, Inner Ear Diseases, Prion Disease, Complex Diseases

Renal Gene Expression Database (RGED) was developed in [Zhang et al. \(2014\)](#) to provide researchers with an access to gene expression profiles of various kidney diseases. RGED includes profiles obtained from renal cell lines, human kidney tissues and murine model kidneys. They searched for gene expression profiling experiments from NCBI and selected a total of 88 studies that published between 2004 and 2014. The platforms are varying across different studies but most of them used Affymetrix, Inc., Illumina, Inc. and Agilent Technologies, Inc. RGED contains profiles of kidney carcinoma, acute kidney injury, autosomal dominant polycystic kidney disease, IgA nephropathy, chronic kidney disease. RGED release 1.0 has a collection of 5354 samples which include cell lines, human tissues and model animals. Previously created databases for kidney research contains only genes or genomic loci in a human which may not be sufficient for a comprehensive study of kidney diseases. RGED allows exploration of gene profiles, relationships between genes of interests, identification of biomarkers and drug targets in different kidney diseases.

The first queryable database ParkDB for gene expression in Parkinson disease was developed in [Taccioli et al. \(2011\)](#). ParkDB has three main functions. The first is to collect all microarray data related to Parkinson disease from publicly available sources such as Array express, Gene Expression Omnibus (GEO) and Stanford microarray database. The second is to provide information about differentially expressed genes across different tissues, treatments and species. The third is to provide information about statistical analysis of microarray data to provide users with the comparison between different experiments.

Shared Harvard Inner Ear Laboratory Database (SHIELD) was developed by [Shen et al. \(2015\)](#). SHIELD is an integrated resource of genomic, transcriptomic and proteomic information of the inner ear. It contains five datasets that are combined into a relational database and provides experimental data and annotations. They derived annotations from public databases and publications. Annotations include gene symbols, gene name, synonyms, human and mouse chromosome cytogenetic banding, genomic coordinations, RefSeq RNA, Ensembl, ontology, etc. Five datasets represent different developmental stages of the variety of model organism species and different cell types. Most commonly used platforms are RNAseq (3*DGE and full-length mRNA), Microarray (GeneChip Mouse Gene 2.0 ST and MOE430) and mass spectrometry.

Prion Disease Database (PDDDB) ([Gehlenborg et al., 2009](#)) developed by is a comprehensive transcriptomic database of mouse prion disease, which provides primary experimentally obtained data. The PDDDB contains mRNA measurements across the time spanning the interval from prion inoculation through the onset of clinical signs in eight mouse strain-prion strain combinations.

dbPTB database was created by [Uzun et al. \(2012\)](#) to facilitate the identification of genetic variants associated with complex diseases. The database combines information about genes, their SNPs, genetic variations and various pathways involved in the preterm birth. They searched publicly available databases and publications that describe genome- and transcriptome-wide studies. Genes, which expression levels were increased or decreased in the association with preterm delivery were included in the database.

Tissue-, Process- and Interaction-Specific Sources: Human Oral Microbiome, Human Brain Transcriptome, Human Ageing, Neurological Ageing, Genotype-Tissue and Genotype-Phenotype Interactions

Human Oral Microbiome Database (HOMD) ([Chen et al., 2010](#)) provides datasets for more than 600 prokaryote species of a human oral cavity. HOMD provides both taxonomic and genomic types of information. Each species was given a unique Human Oral Taxon (HOT) number based on the 16S rRNA gene sequence analysis. Each HOT contains genomic, clinical, phenotypic, phylogenetic and bibliographic information for each taxon. Gene sequences can be matched to full-length 16s rRNA reference dataset by using BLAST tool. HOMD can serve as a model for microbiome database of gut, skin and human body cavities.

The HBT (Human Brain Transcriptome) database ([Johnson et al., 2009](#)) is a public database for transcriptome data and the associated metadata for the developing and adult brain, from data on males and females of different ethnicities, from different brain regions and neocortical areas. The database is built upon the results of whole-genome, exon-level expression analysis of 16 brain regions, generated using Affymetrix GeneChip Human Exon 1.0 ST Arrays. Over 1340 tissue samples from both hemispheres of postmortem human brains were analyzed. The authors applied ANOVA for differential expression analysis and unsupervised hierarchical co-clustering for analyzing brain regions with respect to their transcriptional profiles. The online database enables searching for individual genes and assessing their temporal activity in an age-intensity graph, or searching for the most active genes with regard to their spatial (by brain structure) and temporal (by start and end period) activity.

GenAge, a database of genes potentially associated with human ageing ([Tacutu et al., 2013](#)), was developed as a part of The Human Ageing Genomic Resources (HAGR) framework. GenAge is a publicly available, continuously curated database containing a comprehensive list of genes potentially associated with ageing in humans and model organisms, as well as a list of differentially expressed genes in mammals, including humans, from a meta-analysis of microarray studies. The manually-curated list of human genes in GenAge is 288, annotated with respect to their role in the human ageing process as a whole, as a result of analysis of more than 2200 references. In addition, it includes over 1700 genes associated with longevity in model organisms, which are based on experimentally validated results from the multitude of studies, such as overexpression of certain transcripts or RNA interference with age. Each entry in the database a description of its main functions and its relation to the ageing process, as well as the reason for inclusion into the database.

Brain transcriptome studies on gene dysregulation in Alzheimer's disease (AD) and closely related processes such as ageing and neurological disorders were collected and curated in AlzBase database (Bai *et al.*, 2016). Aside from gene dysregulation in AD, it includes information on correlation of gene dysregulation with severity of AD, functional and regulatory annotations, and a network of the corresponding gene-gene interactions, built via WGCNA. It includes a total of 14,145 dysregulated genes. For gene annotations, eQTL experiments (11,055), in situ hybridization experiments (1167), and GWAS experiments (12,287) were used. The authors constructed a gene coexpression network using several large datasets on brain transcriptome taken from GEO, which covered AD-associated data on several brain regions and peripheral blood, and four stages of progression of the disease. For the selection of differentially expressed genes, the RankProd algorithm ($p < 0.05$) was used. To further group the most significant genes (277), the authors used hierarchical clustering and grouped them into 11 clusters based on the temporal pattern of gene expression.

The Genotype-Tissue Expression (GTEx) database was created as part of GTEx project which aimed at establishing a data resource for studying the relationship between genetic variation and gene expression in human tissues (GTEx Consortium, 2017). The GTEx represents the results of high-throughput sequencing and microarray analysis used for eQTL analysis from samples (7051) of 44 different tissue types from 449 donors. The authors compared Spearman's similarity between various tissues for cis- and trans- eQTLs and used agglomerative hierarchical clustering to group the tissues that share any given eQTL. The GTEx's web portal enables searching for genes by Gene or SNP Id and outputs a visualization of its median expression level across all samples for each tissue.

dbGaP, the database of Genotypes and Phenotypes, was developed for storing and distributing the data from studies of genotype-phenotype interaction in humans (Mailman *et al.*, 2007). It is a large NCBI-sponsored public repository that assigns unique identifiers to studies and datasets. The main objective of the project is to store, collect and distribute phenotype data that can be associated with other data types, such as sequence-derived genotypes and haplotypes, gene expression data, proteomic, metabolomic, and epigenetic data. Currently, the main focus of the GWAS studies covered in dbGaP is microarray data. dbGaP has integration with Sequence Read Archive (SRA), and thus the sequences for the queried genes can be downloaded from the NCBI web portal. Aside from that, users can view the association results in Genome Browser's chromosomal viewer, where loci with the smallest p-value are color-coded, and in the sequencing view, users can see the corresponding gene/ transcript/protein annotations.

Other Tools and Sources: Circular RNA/Small Non-Coding RNA Databases, Raw Read Archives, a Transcriptome Browser and a Human Transcriptome Functional Annotation Source

CircNet is a database for circular RNAs which is the type of regulatory noncoding RNA (Liu *et al.*, 2016). The expression of circRNAs was identified by using transcriptome sequencing datasets in 464 RNA-seq samples. CircNet provides an access novel circRNAs, integrated miRNA-target networks, expression profiles, genomic annotations and sequences of circRNA isoforms. It also displays the regulation between circRNAs, miRNAs, and genes by generating the integrated regulatory network. By using the search box, users can find the gene-miRNA-circRNA regulatory network of particular genes. CircNet allows studying circRNA tissue-specific functions and correlation to diseases due to its expression profile and accessibility.

Human genome data generated from next-generation sequencing and high-resolution comparative genomic hybridization array were collected into the Total Integrated Archive of short-Read and Array (TIARA) database (Hong *et al.*, 2013). TIARA provides an access for variant information and raw data for 16 sequenced transcriptomes, 13 sequenced whole genomes and 33 high-resolution array assays (for September 2012). Genomic variants data includes a total of approximately 9.56 million SNPs, where 23,025 constitute non-synonymous SNPs and approximately 1.19 million indels. The abundance of genomic variants data makes it a useful resource for personalized genomics studies. Genome browser allows the matching of whole-genome sequence with transcriptome sequencing data. Also, the levels of gene expression can be observed with allele-specific quantification.

For storing next-generation sequence data (Leinonen *et al.*, 2011), the Sequence Read Archive (SRA), which serves as NIH's primary archive of high-throughput sequencing data, was created. The SRA stores raw sequencing data and alignment information through such high-throughput sequencing platforms as Roche 454, Illumina Genome Analyzer, SOLiD™, Helicos HeliScope™, Complete Genomics, and SMRT™. The SRA archives raw oversampling NGS data with EMBL and DDBJ, as well as the data from human clinical samples. The database enables users searching for genes of interest via metadata in the query request, as well as performing a sequence-based search using BLAST.

TranscriptomeBrowser (TBrowser) (Lopez *et al.*, 2008) was created for hosting a large integrated database of transcriptional signatures extracted from more than 1400 (currently ~4000) experiments stored in the GEO database. The authors used a modified version of the Markov Clustering algorithm – DBF-MCL, to systematically extract clusters of co-regulated genes, which allowed to define over 18,000 transcriptional signatures (currently around ~40,000). The authors tested established transcriptional signatures for functional enrichment using DAVID knowledgebase, and over-representation of functional terms was found in a large proportion (84%) of these transcriptional signatures. The TBrowser enables searching by GeneID and annotation, and supports Boolean operators.

The GENCODE is an online database developed as a part of the ENCODE (Encyclopedia of DNA Elements) project, which contains functional annotations for 20,687 protein-coding genes, 9640 long non-coding RNA loci for 15,512 transcripts, and 33,977 coding transcripts not represented in RefSeq and UCSC (Harrow *et al.*, 2012). The gene annotations are derived from a combination of manual curation and automatic gene annotation from Ensembl, different computational analysis techniques (PhyloCSF, APRIS annotation pipeline), and targeted experiments. GENCODE contains a comprehensive and manually annotated set of pseudogenes, in addition to the sets of protein-coding and non-coding genes. The authors also used transcriptome

sequencing data facilitated by high-depth RNA sequencing for a very detailed overview of alternative splicing. The web portal of GENCODE enables searching by ENS id, ENS gene id, and gene name (for a corresponding BioDalliance genome visualization), as well as downloading all the relevant GTF/GFF3 datasets, FASTA datasets, and the metadata files.

Database of small human noncoding RNA (DASHR) is the most comprehensive database on human sncRNA genes and mature sncRNA products (Leung *et al.*, 2016). DASHR integrates over 180 smRNA smRNA-seq datasets over 2.5 billion reads and annotation data from various publicly available sources. DASHR contains 7641 sncRNA gene records, and 9703 mature sncRNA product records, corresponding to ~48000 genomic loci. Among the sncRNA genes, 82% are expressed in one or more tissues. As such, 6301 annotated sncRNA genes can be queried in DASHR. DASHR has an intuitive web interface that allows users to search by full name, partial name or ID of sncRNA, compare expression levels in various human tissues, locate RNA loci overlapping with the given genomic coordinates using h19 transcript annotation as a reference, or query by raw RNA sequence. It also provides a summary of annotation and expression information, with an expression heatmap, an expression profile across tissues, and a genome browser for mapped sequences.

Animal Transcriptome Databases

Animal transcriptome databases are mainly represented by the data sources containing the genome and transcriptome sequencing/RNASeq/expression/functional annotation information on the model and mammalian organisms such as *Xenopus*, *Mus musculus*, *Rattus norvegicus* and *Sus scrofa*, combining sequencing and expression data. *M. musculus*-focused databases are represented by data repositories on embryo and brain development as well ageing, reflecting the fact that mouse is a model organism for most of the crucial mammalian processes as important to human health and development research. Ageing-related databases also include entries from *Caenorhabditis elegans*, *Drosophila melanogaster*, *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe* as the organisms are widely used for understanding ageing processes *in vivo*. Arthropods and parasites are also represented by the relevant database source reflecting the importance of the omics data for studies in comparative genomics, evolutionary and developmental biology as well as ecology and plant/human disease management (e.g., rice pest *Chilo suppressalis* and *Leishmania* databases). Another important animal transcriptome database source is a *Ctenopharyngodon idellus* (grass carp) GCGD database; the grass carp is an important farmed fish for the global freshwater aquaculture business. In addition, as mentioned in the previous section, GEO-NCBI also contains a wide range of data from animal sources.

Model and Mammalian Organism Sources

Xenbase is an online database that provides an access to the sequencing data for *Xenopus laevis* and *Xenopus tropicalis* (Karimi *et al.*, 2018). The data were obtained from different resources such as publications and bioinformatics analyses. The database includes genomes for both *X. laevis* and *X. tropicalis*, chromosomal assemblies, annotations, genome segmentation, visualization of RNA-Seq data, protein interaction, gene ontologies, updated ChIP-Seq mapping, etc. Users can use BLAST pages from the main navigator bar and select the genes of interest. They also provide the graphical representation of the genomic and chromosomal structure by using the gene coordinates data.

cSynechococcus sp. PCC 7002 omics studies were combined into the CyanOmics database (Yang *et al.*, 2015). It comprises one genomic dataset, one proteomic dataset and 29 transcriptomic datasets. CyanOmics provides functional annotation for complete genome sequence, transcriptomes under different experimental conditions and proteomic analyses. They also integrated tools for browsing, navigation, sequence alignment and data visualization. The database provides links.

The Mammalian Transcriptomic Database (MTD) was developed by Sheng *et al.* (2017) for characterization and comparative analysis of transcriptome data, and it comprises RNA-seq data from most available tissues of certain mammalian species. Currently, MTD contains RNA-seq data from *Homo sapiens* (83), *Sus scrofa* (27), *Rattus norvegicus* (36) and *Mus musculus* (108). Among the core features of the database is that it allows finding genes based on their neighboring genomic coordinates or joint KEGG pathway, and displays detailed expression information on exons, transcripts, and genes in a genome browser (GBrowse). The MTD enables comparative transcriptomic analysis in both intraspecies and interspecies manner. The database is implemented as a web interface, which provides the following functionalities: Browse (by chromosome, region, or pathway), Search (by gene feature or isoform feature), Analysis (intraspecies and interspecies), and Download (for raw SRA data or gene expression data by tissue/cell line).

Database of Transcriptome in Mouse Embryos (DBTMEE) (Park *et al.*, 2015) was developed for systematic analysis of *mus musculus* fertilization dynamics. DBTMEE is based on ultra-large-scale RNA-seq analysis, with more than 1.5×10^5 oocytes sequenced. The authors also included data from other publicly available resources, such as microarray gene expression data, RNA-seq data prepared by different sequencing protocols and at various cell types and embryonic stages, as well as mass-spectrometry proteomic data, DNA-methylation data and ChIP-seq of histone variants in spermatozoa. The database has a comprehensive web interface, through which the users can identify fertilization-specific genes, assess the impact of histone modifications on early embryo development, and use genetic and epigenetic characteristics to investigate molecular characteristics among totipotent, pluripotent, and differentiated cells. The authors applied hierarchical clustering to categorize the gene expression patterns with respect to mouse embryonic development, and the results of the clustering (with further details and links for each gene within each cluster) are accessible from DBTMEE's front page.

The Brain Transcriptome Database (BrainTx) was developed by Sato *et al.* (2008) as a successor to Cerebellar Development Transcriptome Database (CD-DBT, alternative/former name), an online database centered around transcriptomes related to a cerebellar development of a mouse. The current version of BrainTx represents an integrated online platform for visualization and

analysis of transcriptomes related to various stages and states of the mammalian brain (primarily brain of *mus musculus*, but also includes vast amounts of data on other species). Aside from search by NCBI Gene ID and CD ID, it enables searching for genes and transcription factors via associated keywords, e.g., name of the related gene, name of the associated protein structure or function, or cell function. The online database enables searching for differentially expressed genes during the different stages of brain development, analyze expression patterns with respect to the region (Brain distribution) and tissue distribution pattern (Brain specificity). BrainTx's Gene Ontology Tree Browser includes 6603 genes annotated for their associated biological processes, 5782 genes annotated for their associated cellular components, and 6968 genes annotated for their associated molecular functions.

AGEMAP (Atlas of Gene Expression in Mouse Aging Project) (Zahn *et al.*, 2007) is a gene expression database that catalogs changes in gene expression profiles as a function of age in mice. The authors of the database generated mRNA transcript profiles in mice with respect to changes in age and included 8932 genes in 16 tissues at 4 times during the aging. The authors used hierarchical clustering to categorize the data on age-related transcriptional changes into 3 groups (3 modes of aging for different mouse tissues): neural tissues, vascular tissues, and steroid-responsive group (thymus, gonads), which suggests that different types of tissues have different rates of aging. Certain mice tissues exhibit the strongest transcriptional differences as a function of age, while other mice tissues show little to no difference with age. The database can be downloaded as Z-normalized expression data for each tissue.

GenDR is a database also developed under The Human Ageing Genomic Resources (HAGR) framework (Wuttke *et al.*, 2012). GenDR is the first database that contains consistent gene expression changes induced by dietary restriction, in addition to gene mutations interfering with dietary restriction-mediated lifespan extension. The authors established transcriptional signatures induced by dietary restriction (DR), analyzing differentially expressed genes from microarray data in model organisms (*Caenorhabditis elegans*, *Drosophila melanogaster*, *Mus musculus*, *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe*). For all the model organisms, they assessed the influence of gene expression changes upon DR on the multiple interactomes within an interaction network (using ExprEssence algorithm, analysis of suppressed and stimulated interactions, and DAVID functional enrichment analysis). They identified more than 100 genes that are essential for an effect of dietary restriction to take place (DR-essential genes) in model organisms. The web interface of GenDR enables matching genes with the list of DR-essential genes by Entrez ID, gene symbol or name, or browsing the list of DR-essential genes by species. Finally, it allows comparing the DR-essential genes with their homologs across other species, including *Homo sapiens*.

Arthropod and Parasite Sources

Genomic and transcriptomic database ChiloDB for rice insect *Chilo suppressalis* was developed by Yin *et al.* (2014). The aim of ChiloDB was to provide researchers with access to recent genomic and transcriptomic data to allow comparative genomic studies. They integrated the data with protein-coding genes, microRNAs, piwi-interacting RNAs (piRNAs) and RNA-Seq data. They also provide gene annotations for *Chilo suppressalis*. ChiloDB contains the first version of the *Chilo suppressalis* gene set that includes 10,221 annotated protein-coding genes and 80,479 scaffolds. They identified 82,639 piRNAs with piRNAPredictor software and predicted 262 microRNA genes from small RNA library. They also integrated 37,040 transcripts from midgut transcriptome and 69,977 transcripts from a mixed sample.

Assembled Searchable Giant Arthropod Read Database (ASGARD) database (Zeng and Extavour, 2012) is a resource for transcriptomic data of arthropod species. Transcriptomes were assembled *de novo* from milkweed bug *Oncopeltus fasciatus*, the cricket *Gryllus bimaculatus* and amphipod crustacean *Parhyale hawaiiensis*. Putative ontology, coding region determination, protein domain identification was annotated. Assemblies can be searched by orthology annotation, gene ontology term annotation or Basic Local Alignment Search Tool. Users can download FASTA format assembly product sequences, find links to NCBI predicted orthologs data. Also, the locations of protein domains can be graphically visualized and matched to similar sequences from the NCBI. The database is useful for studies in different fields such as comparative genomics, evolutionary biology, physiology, developmental biology, phylogenomics, and ecology.

The database LeishDB of *Leishmania braziliensis* was developed by Torres *et al.* (2017) to update and improve publicly available data for non-coding RNAs (ncRNAs) and coding gene annotations of *L. braziliensis*. They used updated databases and five prediction algorithms: GENAN, GLIMMER, SNAP, RATT, and AUGUSTUS. A total of 11491 Open reading frames (ORF), including 5263 (45.80%) of them that are associated with proteins were identified. Also, they found 11,243 ncRNAs that belong to different classes distributed along the whole genome. The accuracy of their predictions was verified by the comparison of predicted sequences with those obtained from RNA-seq. The automatic predictions and annotations of non-coding sequences were performed by covariance models comparisons and sequence similarity searches.

GCGD: A Farmed Fish Database

Grass carp genome database (GCGD) was developed by Chen *et al.* (2017) to provide researchers with the centralized database of genomic data for *Ctenopharyngodon idellus*. Genome, amino-acid sequences of predicted protein-coding genes, annotations and their visualizations are included into the database. GCGD also provides an access to transcriptomic datasets, genetic linkage maps, Short Sequence Repeats. They integrated different useful tools which enable efficient data retrieval, analysis, and data visualization. For example, JBrowse provides genomic annotation navigation, BLAST enables to perform sequence alignment, EC2KEGG facilitates the comparison of metabolic pathways, IDConvert allows the conversion of terms across databases and ReadContigs facilitates the extraction of sequences from the genome of grass carp.

Plant Transcriptome Databases

Most transcriptome plant databases are dedicated to agronomically important species such as crop, tree and vegetable plants. For the crop plants, the main data repositories were created for bread wheat, rice, the burclover (*Medicago truncatula*) legume and sorghum. Vegetable and tree species such as carrot and mulberry tree (important for the mulberry-silkworm interaction), eucalyptus (commercially important for the production of paper and cellulosic ethanol) have corresponding transcriptome database resources. Other important sources include transcriptome databases of medicinal plants and metabolic pathway information.

Crop Plants

As wheat provides an estimated 20% of all the consumed calories worldwide, considerable interest in the crop drives efforts into developing more commercially advantageous varieties. Omics resources are the first port of knowledge to identify potential traits of interest which could further help to narrow down to alleles for the use in breeding programs. With this goal, the functional annotation database called dbWFA for bread wheat *Triticum aestivum* was developed in [Vincent et al. \(2013\)](#). dbWFA is based on the reference 56,954 transcripts NCBI wheat UniGene set which is an expressed gene catalogue created based on full-length 17,541 TriFLBD database coding sequences by using expressed sequence tag clustering. Annotation information from the model plant *Arabidopsis thaliana* and *Oryza sativa* L. was linked back to the *T. aestivum* sequences using BLAST-based homology searches. Putative functions were assigned to 45% of the UniGene transcripts and 81% of the TriFLDB full-length coding sequences. The database contains information on functional annotations, Gene Ontologies, gene families and metabolic pathways.

Burclover is also a forage crop of commercial interest. MtDB is a relational database ([Lamblin et al., 2003](#)) contains transcriptomic data for *Medicago truncatula* legume *M. truncatula* samples that represent various developmental stages and pathogen-challenged tissues. Unigene sets generated by different assemblers such as Phrap, Cap3 and Cap4 can be selected and compared. MtDB provides users with quick access and identification of sequences for particular research interests. To facilitate comparison between different sites, the authors performed cross-reference of sequence identifiers from all public *M. Truncatula* sites such as GenBank ID, CCGB, TIGR, NCGR, INRA. Also, the database provides hypertext links for databases for all queries' results.

Sorghum genomic data and functional annotations were collected by [Tian et al. \(2016\)](#) to create SorghumFDB database. The database provides detailed gene annotations, orthologous pairs in Arabidopsis, rice and maize, miRNA and target gene information, genome browser and gene loci conversations. It also includes the information about sorghum gene family classification such as transcription factors and regulators, carbohydrate-active enzymes, ubiquitins, protein kinases, etc. Co-expression data, protein-protein interactions, and miRNA-target pairs were used to construct a dynamic network of biological relationships.

TENOR (Transcriptome Encyclopedia Of Rice) is a database for comprehensive mRNA-seq experiments in rice that was developed by [Kawahara et al. \(2016\)](#). The database includes large-scale mRNA-seq data from rice in a wide variety of conditions. TENOR was developed to enable the scientists better understand the regulatory networks of genes responsible for adaptation to different environmental conditions. As such, the authors used mRNA-seq and applied time-course transcriptome analysis of rice (*Oryza sativa* L.), under 2 plant hormone treatment and 10 abiotic stress conditions. Differential expression analysis detected a large number of genes responsive to abiotic stress and plant hormones treatment. The online database enables users to download rice transcriptome data, co-expression data of the responsive genes, search relevant genes by keyword, genomic coordinates, and by responsive expression patterns. TENOR also includes functionality for visualization of the expression levels under various conditions for each transcript.

Tree and Vegetable Species

The CarrotDB database of *Daucus carota* L. was assembled and analyzed in [Xu et al. \(2014\)](#). They downloaded 14 carrot RNA-seq raw data from Sequence Read Archive (SRA) and National Center for Biotechnology Information (NCBI) and mapped them to the whole genome sequence. They classified a total of 2826 transcription factors (TF) into 57 families and identified them in the entire genomic sequence. Carrot-DB integrates carrot genome sequence, coding sequences, putative genes sequences, transcript sequences, etc. The part of CarrotDB called 'GERMPLASM' provides 45 carrot genotypes taproot photos and information about names, accession numbers, colors of the cortex, countries of origin, phloem and xylem.

Morus Notabilis is an important plant for sericulture and plant-insect interaction studies ([Li et al., 2014](#)). The genome and transcriptome of *M. Notabilis* were sequenced and analyzed. They developed open-access Morus Genome Database (MorusDB). The database provides access to large-scale genomic sequencing and assembly, genes and functional annotation transposable elements (TEs), gene ontology (GO) and expressed sequence tags (ESTs). To provide an access for identification of orthologous and paralogous groups of *M. Notabilis*, the most recent versions of protein sequences of related plants such as *Arabidopsis thaliana*, *Populus trichocarpa*, *Malus domestica* and *Fragaria vesca* were downloaded. Gene ontologies of *M. Notabilis* can be analyzed using Mulberry GO tool package. Gene ontologies can be searched using Search GO or obtained using gene ID and Fetch GO.

EUCANEXT ([Nascimento, 2017](#)) database is an integrated gene expression (RNAseq) and ESTs data platform with a set of the web-based tool allowing sequence similarity/keyword/transcript ID searches as well as allowing to find differentially expressed sets of genes under different custom-chosen conditions. The database is based on 14 Illumina RNA-Seq libraries and 165,268 ESTs; genome annotation information is also available with GO, orthoMCL, NR, Uniref90, Swiss-Prot, Tair and CDD terms. A genome

Table 1 Transcriptomic databases

#	Database name (short name)	Organism	Type of data	URL	Reference (PMID/DOI)
1	TCGA	human	Whole genome and exome sequencing profiles, mRNA/microRNA sequences, copy number and methylation profiles RNA-seq, microarrays	https://cancergenome.nih.gov	10.1038/ng.2764
2	GEO	human/animal/plant/ fungi	RNA-seq	https://www.ncbi.nlm.nih.gov/geo/	10.1093/nar/gks1193
3	BioXpress	human	SNP, gene expression	http://hive.biochemistry.gwu.edu/tools/bioexpress	https://doi.org/10.1093/database/bav019
4	CoReCG	human	Gene expression, microarray	https://www.gencapofamerp.br/hncdb	10.1093/database/baw059
5	HNdb	human	Gene expression, microarray, RT-PCR	http://www.gencapofamerp.br/hncdb	10.1093/database/baw026
6	curatedOvarianData	human	Gene expression, microarray, RT-PCR	http://www.gencapofamerp.br/hncdb	10.1093/database/baw013
7	Cancer RNA-Seq Nexus	human	Gene expression, RNA-seq	http://syslab4.ndhu.edu.tw/CRN	10.1093/nar/gkv1282
8	RGED	human	Gene expression, microarray, RNA-seq	http://ged.wall-eva.net	10.1093/database/bau092
9	ParkDB	human	Microarrays, gene expression	http://www2.cancer.ucl.ac.uk/Parkinson_Db2/	10.1093/database/bar007
10	SHIELD	human	Gene expression, microarray, ChIP-seq	https://shield.hms.harvard.edu	10.1093/database/bav071
11	PDDb	human	Gene expression, microarray	http://prion.systemsbiology.net	10.1093/database/bap011
12	dbPTB	human	SNPs, gene expression	http://ptbdb.cs.brown.edu/dbPTBv1.php	10.1093/database/bar069
13	HOMD	human	Microarray	http://www.homd.org	10.1093/database/baq013
14	HBT	human	Microarrays	http://www.homd.org	10.1016/j.neuron.2009.03.027
15	GenAge	human	Microarray data	http://hbatlas.org/	10.1093/nar/gks1155
16	AlzBase	human	SNP and expression microarray data	http://genomics.senescence.info/genes/	10.1007/s12035-014-9011-3
17	GTEX	human	RNA-seq, microarray data	http://alz.big.ac.cn/alzBase	10.1038/nature24277
18	dbGaP	human	SNP and expression microarray data, RNA-seq	https://www.gtexportal.org/	10.1038/ng1007-1181
19	CircNet	human	Gene expression, RNA-seq	https://www.ncbi.nlm.nih.gov/gap/	10.1093/nar/gkv940
20	TIARA	human	Gene expression, RNA-seq	http://circnet.mbc.nctu.edu.tw/	10.1093/database/bat003
21	SRA	human	RNA-seq, ChIP-seq	http://tiara.gmi.ac.kr	10.1093/nar/gkq1019
22	TranscriptomeBrowser	human	Microarray data	http://www.ncbi.nlm.nih.gov/Traces/sra	10.1371/journal.pone.0004001
23	GENCODE	human	RNA-seq, RT-PCR-seq, RACE-seq	http://tagc.univ-mrs.fr/browser/	10.1101/gr.135350.111
24	DASHR	human	RNA-seq	https://www.gencodegenes.org/	10.1093/nar/gkv1188
25	Xenbase	Xenopus laevis and Xenopus tropicalis	Gene expression, RNA-seq, ChIP-seq	http://www.xenbase.org/	https://doi.org/10.1093/nar/gkx936
26	CyanOmics	Cyanobacterium <i>Synechococcus</i>	Gene expression, RNA-seq	http://lag.ihb.ac.cn/cyanomics	10.1093/database/bau127
27	MTD	Sus scrofa, Rattus norvegicus, Mus musculus, Homo sapiens	RNA-seq	http://mtd.cbi.ac.cn/	10.1093/bib/bbv117

28	DBTMEE	Mus musculus	RNA-seq, ChIP-seq, DNA methylation data, MS proteomic data	http://dbtmee.hgc.jp/	10.1093/nar/gku1001
29	BrainTx	Mus musculus	Microarray data, RT-PCR-seq	http://www.cdttdb.neuroinf.jp/CDT/Top.jsp	10.1016/j.neuro.2008.05.004
30	AGEMAP	Mus musculus	Microarray data	http://cnmg.stanford.edu/~kimlab/aging_mouse/	10.1371/journal.pgen.0030201
31	GenDR	Caenorhabditis elegans, Drosophila melanogaster, Mus musculus, Saccharomyces cerevisiae, Schizosaccharomyces pombe	Microarray data, RNA-seq	http://genomics.senescence.info/diet/	10.1371/journal.pgen.1002834
32	ChiloDB	Chilo suppressalis	RNA-seq, RT-PCR	http://ento.njau.edu.cn/ChiloDB	10.1093/database/bau065
33	ASGARD	arthropods	Gene expression, RNA-seq	http://asgard.rc.fas.harvard.edu/	doi:10.1093/database/bas048
34	LeishDB	Leishmania braziliensis	RNA-seq	www.leishdb.com	10.1093/database/bax047
35	GCGD	Ctenopharyngodon idellus	Genomic data	http://bioinfo.iitb.ac.cn/gcgd	10.1093/database/bax051
36	dbWFA	Triticum aestivum	Microarray	http://versailles.inra.fr/dbWFA/	10.1093/database/bat014
37	MtDB	Medicago truncatula	Transcriptomic data	http://www.medicago.org/MtDB	PMID: 12519981
38	SorghumFDB	Sorghum bicolor	Gene expression, RNA-seq, microarray	http://structuralbiology.cau.edu.cn/sorghum/index.html	10.1093/database/baw099
39	TENOR	Oryza sativa L.	mRNA-seq data	http://tenor.dna.affrc.go.jp/	10.1093/pcp/pcv179
40	CarrotDB	Daucus carota L	RNA-seq data	http://apiaceae.njau.edu.cn:8080/carrotdb/	10.1093/database/bau096
41	MorusDB	Morus notabilis	Gene expression	http://morus.swu.edu.cn/morusdb	10.1093/database/bau054
42	EUCANEXT	Eucalyptus genus	RNAseq and ESTs	http://bioinfo03.ibi.unicamp.br/eucalyptusdb/	10.1093/database/bax079
43	EGENES	93 plant species (monocots, eudicots, ferns, mosses, green/red algae, basal magnoliophyta)	ESTs	http://www.genome.jp/kegg-bin/create_kegg_menu?category=plants_egenes	10.1104/pp.106.095059

browse tool is also available as well as the statistical testing for GO terms' enrichment. Overall, the resources integrated the omics data with a user-friendly platform with a potential to incorporate more data such as microRNAs or SNPs.

Other Sources

EGENES (Masoudi-Nejad *et al.*, 2007) is an extended plant transcriptome-based database of metabolic pathway genes integrated into the KEGG (Kyoto Encyclopedia of Genes and Genomes) database. EGENES is different from the set of other transcriptome data platforms because it does not concentrate mostly on the molecular properties but covers the information on metabolism, signal transduction and the cellular cycle steps. The data comes from assembled EST contigs (base on the public ESTs) with the information on the sources as well as functional annotation and a gene/EST index available for each genome. The main distinguishing feature of EGENES is the integration of genomic and pathway information. Functional assignments are made by connecting gene sets with a list of molecules that interact with it in the cell so that a higher order biological function pathway is created. EGENES pathways are created according to the same protocols which KEGG uses for functional assignment in other species.

At the start of the database in 2006, the pathway information was compiled for 25 species with 178 manually curated unique diagrams and divided into 4 main categories: (1) metabolism (2) genetic information processing (3) environmental information processing (4) cellular processes. As of now, the current EGENES version consists of 79 plant species.

Conclusion

According to the recent of 2018 NAR Molecular Biology Database Collection report the current number of databases is 1737 (Rigden and Fernández, 2018). However, this comprehensive list covers the broad spectrum of different primary and secondary biological databases containing multiple types of "omics" data. Considering the increasing number of biological databases year by year it becomes time inefficient to find a useful resource that would fit exactly to a researcher's specific task. This review is no means comprehensive but aimed to present a current overview of the major available transcriptome sources for human, animal and plant data. This review covers 43 main transcriptomic data sources. The following Table 1 is the list of the sources discussed and provides a link to the databases as well as the publications. At a glance, most of the databases are built on human data with a particular focus on cancer and other major diseases such as Parkinson's, Alzheimer's and kidney diseases. Animal databases also reflect this focus on understanding the transcriptomic profile of major diseases with most of the transcriptome databases dedicated to model organism data crucial for understanding the processes of ageing, embryonic development (in mammals), brain transcriptome and oxygenic photosynthesis (in blue-green algae). Other animal database sources are also focused on arthropod, parasite transcriptome information with a focus on a rice pest and a leishmaniasis-causing parasite. Another database is dedicated to an important farmed fish. Plant transcriptome databases are also created and maintained with an emphasis on agriculturally-relevant species such as wheat, burdock legume, rice, sorghum and carrot. Other databases are dedicated to the economically important mulberry tree and eucalyptus as well as 79 other plant species with extended pathway information, which integrates genomic and biological function pathways. Overall, it is easy to see that technological advances and an overabundance of available data allow rapid integration with other findings and possibilities of performing advanced bioinformatics analysis and in-silico queries (e.g., gene enrichment, pathway construction, ID searches) in a user-friendly interface.

Competing Interests

The authors declared that there are no competing interests.

Acknowledgements

This work was supported by the grant research project "Pan-cancer deconvolution of omics data using Independent Component Analysis" (IRN: AP05135430) and research project "Analysis of cancer transcriptome data using Independent Component Analysis" (budget program "Creation and development of genomic medicine in Kazakhstan" – 0115RKO1931) of the Ministry of Education and Science of the Republic of Kazakhstan.

See also: Analyzing Transcriptome-Phenotype Correlations. Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Characterization of Transcriptional Activities. Characterizing and Functional Assignment of Noncoding RNAs. Comparative Transcriptomics Analysis. Data Storage and Representation. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Information Retrieval in Life Sciences. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Quantitative Immunology by Data Analysis Using Mathematical Models. Regulation of Gene Expression. Standards and Models for Biological Data: FGED and HUPO. Statistical

and Probabilistic Approaches to Predict Protein Abundance. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation. Transcriptome Informatics. Utilising IPG-IEF to Identify Differentially-Expressed Proteins. Whole Genome Sequencing Analysis

References

- Agarwal, R., *et al.*, 2016. CoReCG: A comprehensive database of genes associated with colon-rectal cancer. Database: The Journal of Biological Databases and Curation 2016.
- Bai, Z., *et al.*, 2016. AlzBase: An integrative database for gene dysregulation in alzheimer's disease. Molecular Neurobiology 53 (1), 310–319.
- Barrett, T., *et al.*, 2012. NCBI GEO: Archive for functional genomics data sets – Update. Nucleic Acids Research 41 (D1), D991–D995.
- Chang, K., *et al.*, 2013. The Cancer Genome Atlas Pan-Cancer analysis project. Nature Genetics 45 (10), 1113–1120.
- Chen, T., *et al.*, 2010. The Human Oral Microbiome Database: A web accessible resource for investigating oral microbe taxonomic and genomic information. Database: The Journal of Biological Databases and Curation 2010, baq013.
- Chen, Y.X., *et al.*, 2017. The Grass Carp Genome Database (GCGD): An online platform for genome features and annotations. Database-The Journal of Biological Databases and Curation 2017, bax051.
- Ganzfried, B.F., *et al.*, 2013. curatedOvarianData: Clinically annotated data for the ovarian cancer transcriptome. Database: The Journal of Biological Databases and Curation 2013, bat013.
- Gehlenborg, N., *et al.*, 2009. The Prion Disease Database: A comprehensive transcriptome resource for systems biology research in prion diseases. Database: The Journal of Biological Databases and Curation 2009, bap011.
- GTEX Consortium, *et al.*, 2017. Genetic effects on gene expression across human tissues. Nature 550 (7675), 204–213.
- Harrow, J., *et al.*, 2012. GENCODE: The reference human genome annotation for The ENCODE Project. Genome Research 22 (9), 1760–1774.
- Henrique, T., *et al.*, 2016. HNdb: An integrated database of gene and protein information on head and neck squamous cell carcinoma. Database: The Journal of Biological Databases and Curation 2016, baw026.
- Hong, D., *et al.*, 2013. TIARA genome database: Update 2013. Database: The Journal of Biological Databases and Curation 2013, bat003.
- Johnson, M.B., *et al.*, 2009. Functional and evolutionary insights into human brain development through global transcriptome analysis. Neuron 62 (4), 494–509.
- Karimi, K., *et al.*, 2018. Xenbase: A genomic, epigenomic and transcriptomic model organism database. Nucleic Acids Research 46 (D1), D861–D868.
- Kawahara, Y., *et al.*, 2016. TENOR: Database for comprehensive mRNA-Seq experiments in rice. Plant & Cell Physiology 57 (1), doi:10.1093/pcp/pcv179.
- Lamblin, A.F., *et al.*, 2003. MiDB: A database for personalized data mining of the model legume *Medicago truncatula* transcriptome. Nucleic Acids Research 31 (1), 196–201.
- Leinonen, R., *et al.*, 2011. The sequence read archive. Nucleic Acids Research 39, D19–D21. doi:10.1093/nar/gkq1019.
- Leung, Y.Y., *et al.*, 2016. DASHR: Database of small human noncoding RNAs. Nucleic Acids Research 44 (D1), D216–D222. doi:10.1093/nar/gkv1188.
- Li, J.R., *et al.*, 2016. Cancer RNA-Seq Nexus: A database of phenotype-specific transcriptome profiling in cancer cells. Nucleic Acids Research 44 (D1), D944–D951.
- Liu, Y.C., *et al.*, 2016. CircNet: A database of circular RNAs derived from transcriptome sequencing data. Nucleic Acids Research 44 (D1), D209–D215.
- Li, T., *et al.*, 2014. MorusDB: A resource for mulberry genomics and genome biology. Database: The Journal of Biological Databases and Curation 2014.
- Lopez, F., *et al.*, 2008. TranscriptomeBrowser: A powerful and flexible toolbox to explore productively the transcriptional landscape of the Gene Expression Omnibus database. PLOS ONE 3 (12).
- Mailman, M.D., *et al.*, 2007. The NCBI dbGaP database of genotypes and phenotypes. Nature Genetics 39 (10), 1181–1186.
- Masoudi-Nejad, A., *et al.*, 2007. EGENES: Transcriptome-based plant database of genes with metabolic pathway information and expressed sequence tag indices in KEGG. Plant Physiology 144 (2), 857–866.
- Nascimento, L.C., 2017. EUCANEXT: An integrated database for the exploration of genomic and transcriptomic data from Eucalyptus species. Database: The Journal of Biological Databases and Curation 2017, bax079.
- Park, S.J., *et al.*, 2015. DBTME: A database of transcriptome in mouse early embryos. Nucleic Acids Research. 43), D771–D776. doi:10.1093/nar/gku1001.
- Rigden, D.J., Fernández, X.M., 2018. The 2018 Nucleic Acids Research database issue and the online molecular biology database collection. Nucleic Acids Research 46 (D1), D1–D7.
- Sato, A., *et al.*, 2008. Cerebellar development transcriptome database (CDT-DB): Profiling of spatio-temporal gene expression during the postnatal development of mouse cerebellum. Neural Networks 21 (8), 1056–1069.
- Shen, J., *et al.*, 2015. SHIELD: An integrative gene expression database for inner ear research. Database: The Journal of Biological Databases and Curation 2015, bav071.
- Sheng, X., *et al.*, 2017. MTD: A mammalian transcriptomic database to explore gene expression and regulation. Briefings in Bioinformatics 18 (1), 28–36. doi:10.1093/bib/bbv117.
- Taccioli, C., *et al.*, 2011. ParkDB: A Parkinson's disease gene expression database. Database: The Journal of Biological Databases and Curation 2011, bar007.
- Tacutu, R., *et al.*, 2013. Human Ageing Genomic Resources: Integrated databases and tools for the biology and genetics of ageing. Nucleic Acids Research 41, D1027–D1033. doi:10.1093/nar/gks1155.
- Tian, T., *et al.*, 2016. SorghumFDB: Sorghum functional genomics database with multidimensional network analysis. Database: The Journal of Biological Databases and Curation 2016.
- Torres, F., *et al.*, 2017. LeishDB: A database of coding gene annotation and non-coding RNAs in *Leishmania braziliensis*. Database-The Journal of Biological Databases and Curation 2017, bax047.
- Uzun, A., *et al.*, 2012. dbPTB: A database for preterm birth. Database: The Journal of Biological Databases and Curation 2012, bar069.
- Vincent, J., *et al.*, 2013. dbWFA: A web-based database for functional annotation of *Triticum aestivum* transcripts. Database: The Journal of Biological Databases and Curation 2013, bat014.
- Wan, Q., *et al.*, 2015. BioXpress: An integrated RNA-seq-derived gene expression database for pan-cancer analysis. Database: The Journal of Biological Databases and Curation 2015.
- Wuttke, D., *et al.*, 2012. Dissecting the gene network of dietary restriction to identify evolutionarily conserved pathways and new functional genes. PLOS Genetics 2012.
- Xu, Z.S., *et al.*, 2014. CarrotDB: A genomic and transcriptomic database for carrot. Database: The Journal of Biological Databases and Curation 2014.
- Yang, Y., *et al.*, 2015. CyanOmics: An integrated database of omics for the model cyanobacterium *Synechococcus* sp. PCC 7002. Database: The Journal of Biological Databases and Curation 2015.
- Yin, C., *et al.*, 2014. ChiloDB: A genomic and transcriptome database for an important rice insect pest *Chilo suppressalis*. Database: The Journal of Biological Databases and Curation 2014.
- Zahn, J.M., *et al.*, 2007. AGEMAP: A gene expression database for aging in mice. PLOS Genetics 3 (11), e201.
- Zeng, V., Extavour, C.G., 2012. ASGAR: An open-access database of annotated transcriptomes for emerging model arthropod species. Database: The Journal of Biological Databases and Curation 2012, bas048.
- Zhang, Q., *et al.*, 2014. Renal Gene Expression Database (RGED): A relational database of gene expression profiles in kidney disease. Database: The Journal of Biological Databases and Curation 2014.

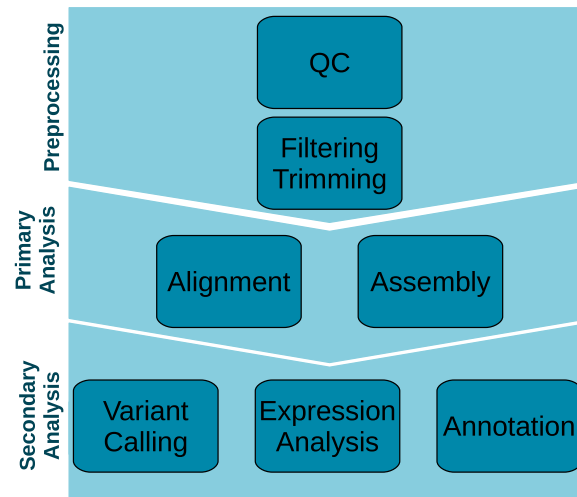


Fig. 1 Example workflow of basic NGS analysis on the levels of preprocessing, primary analysis and subsequent steps.

Herein, p is the probability for an incorrect base call. A second definition was introduced in 2004 by [Bennett \(2004\)](#):

$$Q_{\text{Solexa}} = -10 \log_{10} \frac{p}{1-p} \quad (2)$$

Different variants of FASTQ evolved with those two variations of quality values and are described in [Table 1](#). The Q scores in Sanger FASTQ format are encoded in printable ASCII characters 33–126, which allows assigning quality values between 0 and 93. A Q₃₀ Phred score means a base call accuracy of 99.9% and a error probability of 1 in 1.000 ($10^{-3.0}$).

The underlying FASTQ variant should be aware. There are also several tools to interconvert FASTQ formats such as Biopython (Cock *et al.*, 2009a), EMBOS (Rice *et al.*, 2000) or MAQ (Li *et al.*, 2008a).

Quality Control of Raw Data

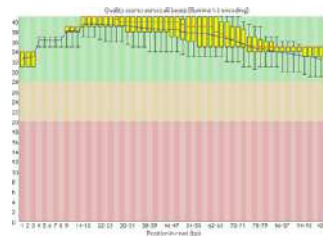
Quality control is a crucial aspect in nearly every step of the NGS analysis pipeline. The results of sequencing data analysis are strongly dependent on the quality of raw data. Checking their quality should be the initial step of every data analysis. Errors can origin from several factors, including the library protocol, the sequencing and the wet-lab preparation. Commercial sequencing platforms provide their own quality control but erroneous reads often remain in raw data. Therefore, various metrics should be checked in the first step of a data processing pipeline. Many tools for this purpose are publicly available. One of the most common programs is **FastQC**. It allows a direct import of raw reads in different file formats and offers an initial quality control validation with multiple metrics to get a quick overview of the data. **FastQC** reports provide graphical summaries, tables and highlighted problems. But it is important to evaluate those results as they are impacted by their sequencing platforms or library types. The most important parameters to check with **FastQC** are listed below:

List Items		Basic requirements table	
Category	Item Name	Requirement	Value
✓	Basic hardware	Hardware	Value
✓	Basic software	Software	Value
✓	Basic network	Network	Value
✓	Basic security	Security	Value
✓	Basic performance	Performance	Value
✓	Basic reliability	Reliability	Value
✓	Basic maintainability	Maintainability	Value
✓	Basic scalability	Scalability	Value
✓	Basic flexibility	Flexibility	Value
✓	Basic interoperability	Interoperability	Value
✓	Basic compatibility	Compatibility	Value
✓	Basic portability	Portability	Value
✓	Basic reusability	Reusability	Value
✓	Basic extensibility	Extensibility	Value
✓	Basic modifiability	Modifiability	Value
✓	Basic testability	Testability	Value
✓	Basic observability	Observability	Value
✓	Basic auditability	Auditability	Value
✓	Basic accountability	Accountability	Value
✓	Basic transparency	Transparency	Value
✓	Basic inclusivity	Inclusivity	Value
✓	Basic diversity	Diversity	Value
✓	Basic equity	Equity	Value
✓	Basic justice	Justice	Value
✓	Basic freedom	Freedom	Value
✓	Basic peace	Peace	Value
✓	Basic happiness	Happiness	Value
✓	Basic well-being	Well-being	Value
✓	Basic health	Health	Value
✓	Basic education	Education	Value
✓	Basic employment	Employment	Value
✓	Basic social security	Social security	Value
✓	Basic housing	Housing	Value
✓	Basic energy	Energy	Value
✓	Basic environment	Environment	Value
✓	Basic climate	Climate	Value
✓	Basic biodiversity	Biodiversity	Value
✓	Basic ecosystems	Ecosystems	Value
✓	Basic natural resources	Natural resources	Value
✓	Basic human resources	Human resources	Value
✓	Basic capital resources	Capital resources	Value
✓	Basic technology resources	Technology resources	Value
✓	Basic knowledge resources	Knowledge resources	Value
✓	Basic innovation resources	Innovation resources	Value
✓	Basic entrepreneurship resources	Entrepreneurship resources	Value
✓	Basic leadership resources	Leadership resources	Value
✓	Basic management resources	Management resources	Value
✓	Basic organizational resources	Organizational resources	Value
✓	Basic cultural resources	Cultural resources	Value
✓	Basic social resources	Social resources	Value
✓	Basic economic resources	Economic resources	Value
✓	Basic political resources	Political resources	Value
✓	Basic legal resources	Legal resources	Value
✓	Basic moral resources	Moral resources	Value
✓	Basic spiritual resources	Spiritual resources	Value
✓	Basic religious resources	Religious resources	Value
✓	Basic philosophical resources	Philosophical resources	Value
✓	Basic scientific resources	Scientific resources	Value
✓	Basic artistic resources	Artistic resources	Value
✓	Basic literary resources	Literary resources	Value
✓	Basic musical resources	Musical resources	Value
✓	Basic theatrical resources	Theatrical resources	Value
✓	Basic cinematic resources	Cinematic resources	Value
✓	Basic television resources	Television resources	Value
✓	Basic radio resources	Radio resources	Value
✓	Basic newspaper resources	Newspaper resources	Value
✓	Basic magazine resources	Magazine resources	Value
✓	Basic book resources	Book resources	Value
✓	Basic record resources	Record resources	Value
✓	Basic video resources	Video resources	Value
✓	Basic audio resources	Audio resources	Value
✓	Basic image resources	Image resources	Value
✓	Basic graphic resources	Graphic resources	Value
✓	Basic design resources	Design resources	Value
✓	Basic architecture resources	Architecture resources	Value
✓	Basic engineering resources	Engineering resources	Value
✓	Basic computer resources	Computer resources	Value
✓	Basic internet resources	Internet resources	Value
✓	Basic mobile resources	Mobile resources	Value
✓	Basic cloud resources	Cloud resources	Value
✓	Basic big data resources	Big data resources	Value
✓	Basic artificial intelligence resources	Artificial intelligence resources	Value
✓	Basic blockchain resources	Blockchain resources	Value
✓	Basic cryptocurrency resources	Cryptocurrency resources	Value
✓	Basic virtual reality resources	Virtual reality resources	Value
✓	Basic augmented reality resources	Augmented reality resources	Value
✓	Basic mixed reality resources	Mixed reality resources	Value
✓	Basic extended reality resources	Extended reality resources	Value
✓	Basic immersive reality resources	Immersive reality resources	Value
✓	Basic virtual world resources	Virtual world resources	Value
✓	Basic virtual economy resources	Virtual economy resources	Value
✓	Basic virtual society resources	Virtual society resources	Value
✓	Basic virtual culture resources	Virtual culture resources	Value
✓	Basic virtual identity resources	Virtual identity resources	Value
✓	Basic virtual citizenship resources	Virtual citizenship resources	Value
✓	Basic virtual governance resources	Virtual governance resources	Value
✓	Basic virtual justice resources	Virtual justice resources	Value
✓	Basic virtual freedom resources	Virtual freedom resources	Value
✓	Basic virtual peace resources	Virtual peace resources	Value
✓	Basic virtual happiness resources	Virtual happiness resources	Value
✓	Basic virtual well-being resources	Virtual well-being resources	Value
✓	Basic virtual health resources	Virtual health resources	Value
✓	Basic virtual education resources	Virtual education resources	Value
✓	Basic virtual employment resources	Virtual employment resources	Value
✓	Basic virtual social security resources	Virtual social security resources	Value

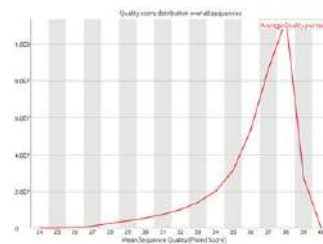
Basic Statistics (Total sequences, sequence length, %GC): The basic statistics section provide the total number of sequence, their length (that depend on the sequencer and the library protocol) and the GC content. The latter varies across species but there should not be a significant deviation ($>10\%$) from the expected value. All values should conform with the experiment design.

Table 1 FASTQ Variants. Quality types are explained in formula (1) and (2). The ASCII offset is needed to produce printable characters

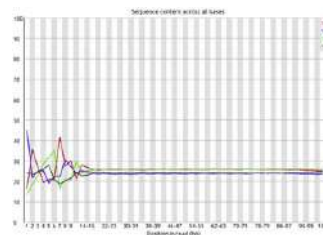
Description	Quality type	ASCII offset	Quality range
Sanger	Phred	33	0–93
Solexa	Solexa	64	–5–62
Illumina 1.3 +	Phred	64	0–62
Illumina 1.5 +	Phred	64	3–62
Illumina 1.8 +	Phred	33	0–93



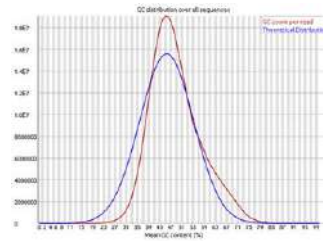
Per Base Sequence Quality: The distribution of quality scores across all bases at each read position is shown. The Phred-scaled scores are assigned by the base calling algorithms of the sequencer. Due to differences in chemistry, hardware and image processing, the Q-scores are not comparable among platforms. There is a general degradation of quality with increasing read length and it is advised to truncate bases with poor quality scores. Also mid-read drops in quality can effect the mapping efficiency.



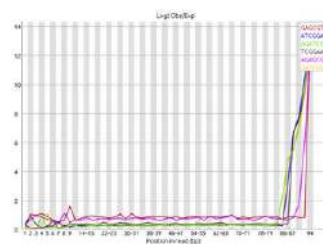
Per Sequence Quality Scores: This reports the number of reads (y-axis) versus their mean sequence quality (x-axis). The most frequently observed mean Phred-score should be higher than 27 with an error rate below 0.2%. A significant amount of reads with an overall low quality could indicate problems during sequencing. This can commonly occur to a subset of sequences. Long-read sequencing runs tend to lower read scores. This could be rectified by quality trimming.



Per Base Sequence Content: The plot shows the proportion of each base across all read positions. Four parallel lines would be expected in a random library. Some types of libraries will induce biases in sequence composition (e.g., hexamer priming during cDNA generation produces biases at the beginning of nearly all RNA-Seq reads) (Hansen *et al.*, 2010). Furthermore bisulfite treatment converts most cytosines to thymines and results in a high difference of the relative amount of C and T.



Per Sequence GC Content: The distribution of the mean GC content of each read should follow the shape of a normal distribution. Sharper shapes indicate a contamination with a specific organism or adaptor dimers. Biologically overrepresented sequences result in small spikes. A broader shape hints at different contaminants. The peak is expected at the overall %GC of the genome. Peak shifts are caused by systematic bias.



k-mer Content: The coverage of 7-mers (subsequences of 7 nt) over each read position in the whole library should be evenly distributed. The comparison between expected and observed frequency with binomial test reports possible positionally biased enrichments of k-mers. Overrepresented sequences and random hexamer primers in RNA-Seq libraries lead to highly enriched k-mers.

Sequence	Count	Binomial p-value	Possible Source
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG
AGGAGAGAG	1000	0.000000	AGGAGAGAG

Overrepresented Sequences: A high diversity of sequences is expected in a random fragmented library. If a sequence is represented by a significant proportion of the total library, it is considered as overrepresented. This could be caused by contamination, a low diverse library (e.g., small RNA libraries) or artifacts of adaptor sequences. These highly duplicated reads can be searched against databases of common contaminants or primer sequences and should be filtered out.

Trimming and Filtering

There are various causes for low quality bases/reads in sequencing data, as already mentioned in Section “Quality Control of Raw Data”. Some NGS applications (e.g., *de novo* assembly or detection of low frequency mutations) require high accuracy reads, because poor quality input data would result in erroneous outputs. Therefore it is desired to discard ambiguous sequence parts and keep as much as possible. A quality control of trimmed/corrected reads is recommended to exclude remaining low quality reads.

Trimming

There are two approaches to handle this problem. The first removes low quality regions from reads, which is called “trimming”. Different implementations, for instance with sliding window approaches, try to achieve the aim of keeping the longest possible high quality subsequence while cutting off poor quality bases. Trimming is advised before most NGS applications.

Available trimming tools are CUTADAPT (Martin, 2011), TRIMMOMATIC (Bolger *et al.*, 2014), PRINSEQ (Schmieder and Edwards, 2011), FASTX (Gordon, 2008) or SOLEXAQA (Cox *et al.*, 2010).

Filtering

The second method of correcting reads is superimpose them to each other. The essential assumption behind this approach is the observation, that erroneous bases occur infrequently and randomly and can be modified by the most frequent reads with this pattern. These approaches base on using multiple sequence alignments, hidden Markov models, suffix trees/arrays or k-mers, whereby the last one seems to be the most suitable (Akogwu *et al.*, 2016). Corrected reads can improve alignments or assemblies significantly. However, it is ineligible for applications with a non-uniform read distribution (e.g., RNA-Seq, see chapter 4.3) or if a low coverage is expected (e.g., ctDNA analysis or discovering of rare variants). Common filtering tools are QUAKE (Kelley *et al.*, 2010), REPTILE (Yang *et al.*, 2010), MUSKET (Liu *et al.*, 2012), SEECER (Le *et al.*, 2013) or Rcorrector (Song and Florea, 2015).

Primary Analysis

File Formats: SAM/BAM

The *de facto* standard for storing alignments is the Sequence Alignment/Map (SAM) format (Li *et al.*, 2009a). Alignment information is organized in a tabular structure in plain text. The human readable SAM format can be easily converted into Binary Alignment/Map (BAM) format, which contains the same information in a compressed, indexable, binary form. It is suitable for many alignment programs because of its flexibility. This also allows the conversion from other alignment formats. The ability of streaming records directly reduces the RAM requirements. Indexing allows a fast retrieving of alignments at a specific genomic position.

SAM & BAM (see "Relevant Websites section") files consist of an optional header section followed by the alignments. Each line of the header begins with the @ character followed by a two letter code, specified tags and their value. Supplemental information about SAM version, reference sequence, library protocol and various comments can be placed here. Each alignment line contains 11 mandatory fields, shown in Table 2.

The FLAG value of the second column can be converted in a binary number, where each bit holds the value (0 or 1) of a specific property (e.g., pairing, mapping and stranding informations). The CIGAR string (6th column) describes how the read is aligned against the reference with the possible operations (M: match; I: insert; D: deletion; N: skipped region; S: soft clipped; H: hard clipped; P: padding; =: read match; X: read mismatch) and how often they occur (As an example, the CIGAR string 34M3D specifies a sequence with 34 nucleotide bases matching to the reference and 3 deleted bases). The associated software package and library is called samtools (Li *et al.*, 2009a). It provides numerous options to:

- Convert between SAM and BAM and other formats,
- Sort alignments in BAM file according position on reference sequence,
- Index sorted BAM file for faster access,
- Extract header information or specific alignments (based on FLAG or region),
- Visualize interactively,
- Mark/remove (PCR) duplicates.

It should be mentioned here that the BAM format is also used to store raw sequence reads (called unaligned BAM), since it requires less disk space than FASTQ and the conversion between FASTQ and SAM/BAM can be performed easily. But compressing Deorowicz and Grabowski (2011) the FASTQ format also reduces its size significantly and gzipped data can be directly processed by a lot of tools.

Table 2 Description of the 11 mandatory fields of the SAM/BAM file format

Field	Name	Description
1	QNAME	Query name of read (pair)
2	FLAG	Bitwise flag (combination for various properties)
3	RNAME	Reference sequence name
4	POS	Leftmost mapping position of alignment
5	MAPQ	Mapping quality (Phred-scale)
6	CIGAR	CIGAR string (notation for variants)
7	RNEXT	Reference name of mate
8	PNEXT	Position of mate
9	TLEN	Template length (insert size)
10	SEQ	Query sequence
11	QUAL	Query quality (Phred score)

Note: Modified from Li H., *et al.*, 2009. The sequence alignment/map format and SAMtools. Bioinformatics 25 (16), 2078–2079.

Sequence Alignment

Sequence alignment describes the arrangement of two or more DNA/RNA/protein sequences based on their similarity. In the field of NGS, this method is used to align short reads to a reference sequence, also called read mapping. Alignments are crucial for whole genome sequencing, exome sequencing, transcriptome sequencing, ChIP-Seq and much more applications. Reference sequences can be obtained from public databases and are usually stored in FASTA format. Current next-generation sequencers produce hundreds of million reads with a typically length > 200 bp per run and represent them in FASTQ files.

Sequence alignment is not as CPU-intensive as assembly but yet not trivial. Depending on the size of the input files (sequencers with one terabases throughput are available), powerful hardware is still required. Due the short lengths of reads, there may be multiple matching positions on the reference sequence. Genomes of many organisms contain highly repetitive elements (half of the human genome belongs to known families of transposable elements Schmid and Deininger, 1975, Treangen and Salzberg, 2013), therefore it is commonly encountered that reads can not span the repeats and they map at multiple positions. The resequenced DNA always differs from the reference sequence. Small structural variations such as InDels (insertions or deletions of a single base) and substitutions must be considered. Ambiguous reads can not be aligned with confidence if there occur several multiple alignments with the same similarity scores. Furthermore, these problems are exacerbated by sequencing errors of reads and reference sequence.

To process the ever increasing number of reads in an acceptable amount of time, traditional dynamic programming algorithms, Needleman Wunsch alike, are not applicable anymore. Efficiency can be gained by preprocessing the reads, the reference sequence, or sometimes both. Two methods stand out here: hashing and Burrows-Wheeler transformation. All algorithms relying on hash tables follow the seed-and-extend strategy, which was already used by BLAST (Altschul *et al.*, 1990). Notable aligners basing on this methods are MAQ (Li *et al.*, 2008a; Homer *et al.*, 2009) and SSAHA2 (Ning *et al.*, 2001).

Another clever approach is using data structures such as prefix/suffix trees/arrays. Several tools employ those with Burrows-Wheeler transform (BWT), an algorithm origins from text data compression. The authors of BWA (Li and Durbin, 2009) explain this method in detail in their paper. Further aligners using this strategy are BOWTIE2 (Langmead *et al.*, 2009), and SOAP (Li *et al.*, 2008b). HISAT2 (Kim *et al.*, 2015), TOPHAT2 (Kim *et al.*, 2013) and STAR (Dobin *et al.*, 2013) provide extended functionality, such as spliced alignments and should be considered for mapping RNA-Seq data.

Both methods have their strengths and weaknesses to be aware of. Hashing methods show good results with erroneous reads. A drawback is the increased runtime with seeds on highly repeated sequences, because the extend step is performed for every seed. This is the advantage of BWT aligners, as all matching positions are joined in one path. On the downside, they have difficulties to resolve sequencing errors.

Last but not least, the correct implementation is important regarding feasibility and runtime. Hard disk I/O operation are extremely slow. Keeping required data in CPU cache or RAM speeds up analysis. Furthermore, mapping is highly parallelizable and many aligners provide the option to use multiple cores/threads. To increase sensitivity and specificity of the alignment, paired end (PE) and mate-pair (MP) mapping should be regarded. Although the size of FASTA is significantly smaller than that of FASTQ, the additional quality values improve alignment accuracy remarkably.

Alignment quality control is essential, even if used reads passed their raw QC. Distinct NGS applications have different control parameters to be regarded. Overall mapping rate is a general metric to be observed. In case of exome sequencing the proportion of reads mapped to targeted exon regions is of importance to estimate the capture efficiency. The mean/median read coverage and percentage of the covered genome are relevant, especially for whole genome sequencing. Moreover, SAM/BAM files provide the mapping quality and insert size for each read.

Programs to be considered for QC are BAMSTATS (Ashelford, 2011), PICARD TOOLS (Picard-Team, 2017), QPLOT (Li *et al.*, 2013) and the Genome Analysis Tool Kit (GATK) (McKenna *et al.*, 2010).

Genome/Transcriptome Assembly

Assembling in bioinformatics describes the reconstruction of a longer sequence from smaller fragments. This is a computational challenging task, especially for large and complex genomes (e.g., from humans and plants), because each read has to be compared with every other read. Repetitive DNA sequences, heterozygosity, high ploidy, and transposons makes it arduous to assemble the genome from short (paired end) reads. Using very long reads (e.g., PacBio Systems) and mate pair libraries with various insert sizes will facilitate and improve the assembly. The general procedure is assembling NGS reads into larger contiguous sequences, called contigs. This is the first step of assembling followed by the arranging of contigs in scaffolds in their right order, orientation and distance. Gaps (caused by repeats or no coverage) between contigs are filled by N's instead of nucleotides. Next, all scaffolds that belong to the same chromosome are arranged in one sequence. Closing all gaps completes the assembly.

It is hard, labor-intensive and expensive to achieve a complete assembly at the chromosome-level. Many genome assemblies are already available in public databases. It is recommended to always align to accessible reference genomes, because it improves and accelerates the assembly. Reconstructing genome or transcriptome sequences without using a reference is called *de novo* assembly. Numerous software tools are available, but they vary strongly with respect to their performance, hardware requirements and output. Therefore no general recommendation is possible. Various teams compete periodically in the Assemblathon (see "Relevant Websites section") to compare different assemblers and methods. Assemblathon 1 (Earl *et al.*, 2011) and 2 (Bradnam *et al.*, 2013) revealed that one assembler may perform well in one situation (e.g., one species) but will not work as well in other situations. Two methods are utilized by most assemblers and will be described in the next sections.

Overlap-layout-consensus method

As the name implies, Overlap-Layout-Consensus (OLC) works in three stages:

- O pairwise alignments of all reads find overlaps and construct a graph of reads (nodes) and overlaps (edges),
- L overlapping reads are ordered to identify redundant reads which are removed during the layout step,
- C finally, the most likely nucleotides are chosen to build a consensus sequence.

CELERA (Myers, 2000), ARACHNE (Batoglou *et al.*, 2002), SGA (Simpson *et al.*, 2012) and NEWBLER (Margulies *et al.*, 2006) are exemplary mentioned assemblers using the OLC method.

de Bruijn graph method

The de Bruijn Graph (DBG) approach chops all reads into short strings of a fixed size k . A directed graph is built by connecting k -mers, where adjacent nodes overlap by $k-1$ nucleotides. The number of matches from short reads to each node is tracked. The genome sequence can be constructed following the paths in the graph and those node coverages. The layout step is a Hamiltonian path problem (NP hard) and building the consensus sequence requires multiple sequence alignments, which is very CPU consuming. Constructing the contig sequence in DBG approach infers the Euler path problem and does not require read alignments, which is algorithmically easier to solve and more CPU/RAM efficiently. DBG approaches seem to perform better with large NGS datasets.

Common DBG-tools are SOAPDENOV2 (Luo *et al.*, 2012) ALLPATHS-LG (Gnerre *et al.*, 2011), ABYSS (Simpson *et al.*, 2009) and VELVET (Zerbino and Birney, 2008).

Transcriptome reconstruction

Assemblers, as e.g., TRINITY (Grabherr *et al.*, 2013), SOAPDENOV-TRANS (Xie *et al.*, 2014), OASES (Schulz *et al.*, 2012) and TRANS-ABYSS (Robertson *et al.*, 2010) are mentioned here for the transcriptome reconstruction using RNA-Seq data instead of DNA. The challenge of RNA-based assembly is the high coverage caused by gene expression. These tools are also capable to detect splicing events and can reconstruct different isoforms. It is suggested to combine several assemblers and methods for better results.

The assembly quality can be evaluated by different metrics, such as N50, multiplicity or coverage. Especially the contig or scaffold lengths are used to describe the completeness of draft assemblies. The N50 statistic characterizes the length distribution of contigs and can be seen as a weighted median. In an set of contigs, ordered by their lengths, N50 is the length of the contig at 50% of total assembly length. That means at least 50% of the nucleotides of the entire assembly are contained in contigs with the N50 length or longer. Linked to this value is the L50 metric, which represents the number of contigs whose lengths form the N50.

The authors of the Assemblathon (Bradnam *et al.*, 2013) proposed ten different metrics, which should be weighted to the own demands.

Secondary Analysis

Formats: VCF

The Variant Call Format (VCF) (Danecek *et al.*, 2011) is a text file format for storing DNA variations, such as single nucleotide variant (SNV), InDels and structural variations (SVs) together with annotations. It was developed for the 1000 Genomes Project with the aim of representing a wide variety of genomic variations regarding one reference. Therefore it stores only variations and no redundant genetic data are stored e.g., in the General Feature Format (GFF). Compressing saves disk space and indexing allows fast data access. A collection of tools to filter, compare, summarize and convert VCF files is combined in the VCFtools package.

A VCF file is segmented in a header and a data section. Lines starting with `##` define the header and provide space for meta information. Special keywords allow an easy parsing. The line starting with a single `#` character marks the field definition and beginning of the data section. Each data line corresponds to a variant at one genomic position or region. They are tab separated in 8 mandatory fields, which are described in Table 3.

Table 3 Description of the fields of the VCF format

Field	Name	Description
1	Chrom	name of reference sequence (typically chromosome)
2	POS	position of variation on reference
3	ID	unique identifier, e.g., dbSNP ID
4	REF	Reference allele
5	ALT	comma-separated list of alternate alleles
6	QUAL	associated quality score (Phred-scale)
7	FILTER	FLAG for site filtering information
8	INFO	user extensible annotation
9	FORMAT	(optional) genotype fields
10	SAMPLE	(optional) sample identifiers

Variant Discovery

Single nucleotide variants

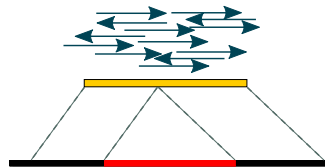
Variant discovery is an important procedure for a lot of whole genome sequencing (WGS) and whole exome sequencing (WES) experiments due the fact of differences between the reference and the sequenced individual. The most abundant genomic variation is the substitution of a single nucleotide base, called single nucleotide variant (SNV). Synonymous SNVs are called single nucleotide polymorphism (SNP). SNVs are often used as genetic markers in genome-wide association studies (GWAS), especially if they are related to diseases.

Another form of genomic variations are insertions or deletions of a single base, also called InDels (Note that in this context an InDel is only *a* single base variation and that an insertion/deletion of multiple bases is considered multiple consecutive InDels). There is a variety of algorithms for SNV detection based on probabilistic or heuristic methods. One has to distinguish SNVs from sequencing errors or other systematic biases. Another common source of error are misalignments. GATK offers a local realignment function around SNVs and InDels to minimize errors due misalignments.

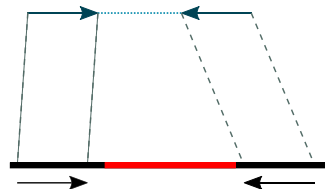
Common tools for SNP calling are FREEBAYES (Garrison and Marth, 2012), SAMTOOLS MPILEUP, SOAPsnp (Li *et al.*, 2009b), VARSCAN (Koboldt *et al.*, 2009) and GATKs HAPLOTYPECALLER.

Structural variations

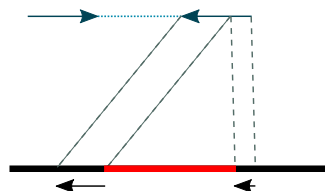
Structural variations (SV) affect larger sequence fragments (> 50 nucleotides (Mills *et al.*, 2011)). They include insertions, deletions, duplications, translocations and inversions. SVs can be assessed by four approaches, which are shortly described below:



Assembly In general the assembly method is capable to detect all possible SV. However, assembling is error-prone, as described in chapter 3.3. It is recommended to use a reference-assisted local assembly instead a *de novo* whole genome assembly. It is also advised to reduce the set of reads to ones, that are missing a mate or had other mapping difficulties in this region.



Read pair (RP) SVs are detected by odd mapping patterns. Differences between expected library insert size and distances of mapped read pairs point as insertions, if the mapping distance shorter than the insert size or as deletions (vice versa). Translocations and inversions can cause different relative mapping orientation of a read pair.



Split read (SR) Split read analysis show similarities to the RP method but the breakpoint of the variant is located directly at the reads. It is necessary to split the read in two or more fragments and realign them independently. Breakpoints can be detected on single base level, however short fragments may map at many genomic positions. After aligning the fragments successful, SVs can be determined by distance and orientation.



Read depth (RD) The read coverage of unique genomic regions should be equally distributed. Deletions and duplications result in significantly lower or higher read depth and copy number variants of large regions can be discovered easily. Insertions of (novel) sequences or inversions are not detectable through this method.



Numerous tools utilize a combination of these methods to reach a higher specificity and sensitivity for SV detection. BREAKDANCER (Chen *et al.*, 2009), CNVNATOR (Abyzov *et al.*, 2011), DELLY (Rausch *et al.*, 2012), HYDRA (Quinlan *et al.*, 2010) and PINDEL (Ye *et al.*, 2009) are mentioned here as examples.

Expression Analysis

Sequencing mRNA, an intermediate between genome and proteome has a broad variety of applications in life sciences. RNA-seq enables a combined identification and quantification of gene activity and associated mechanisms. The diversity of scientific questions of interests makes it impossible to provide an universal analysis approach. One primary objective is gene expression profiling between samples.

A typical workflow begins with the mapping of quality controlled RNA-Seq reads to a reference genome/transcriptome or *de novo* assembled contigs. This step is followed up by an identification of transcripts, exons or genes. Tools, such as CUFFLINKS (Trapnell *et al.*, 2010) and STRINGTIE (Pertea *et al.*, 2015), assemble transcripts of each data set and estimate their expression levels. Using a spliced alignment as input (e.g., from TOPHAT2 or HISAT2) allows to summarize novel transcripts and spliced isoforms along with annotated genes.

Normalization

First, all transcript assemblies from different RNA-seq libraries must be merged to one master transcriptome. The next essential step is the normalization of expression values to enable comparisons within and between samples. Intralibrary normalization allows the quantification of genes with different lengths. A common method is calculating RPKM/FPKM (reads/fragments per kilobase exon per million mapped reads) values. Comparing expression levels of individual genes between samples requires interlibrary normalization. A simple approach is the adjustment with respect to the library size but there are further methods, such as different scaling factors or quantile normalization. Multiple methods are described and compared by Risso *et al.* (2014), Li *et al.* (2015), Bullard *et al.* (2010).

Differential expression

Differential expression analysis aims to detect genes, which significantly differ in their expression level across samples/conditions. RNA-Seq offers a wide dynamic range and a low background noise but inherits other difficulties, such as non-uniform read coverage distribution and sequencing biases through nucleotide compositions. Different statistical models and methods to counter these are implemented in several tools and packages. Many of them are part of Bioconductor (Huber *et al.*, 2015) and based on the statistical programming language R (R Core Team, 2017).

DESeq2 (Love *et al.*, 2014), edgeR (Robinson *et al.*, 2010), LIMMA (Ritchie *et al.*, 2015), EBSeg (Leng and Kendziorski, 2015), BALLGOWN (Frazee *et al.*, 2015) and BAYSeq (Hardcastle, 2012) are notable representatives. The Basic Tuxedo Suite (Trapnell *et al.*, 2012) (including BOWTIE TOPHAT CUFFLINKS CUMMERBUND (Goff *et al.*, 2013)) and the “new Tuxedo” protocol (Pertea *et al.*, 2016) (HISAT, STRINGTIE, BALLGOWN) offer a complete RNA-Seq pipeline. There are also fast methods to estimate abundances without aligning reads. SALMON (Patro *et al.*, 2015), SAILFISH (Patro *et al.*, 2014), KALLISTO (Bray *et al.*, 2015) and RSEM (Li and Dewey, 2011) should be noted. However, it is important to choose tools and workflow fitting to the experiment settings. All tools offer detailed documentations and examples, which should be regarded.

Genome Annotation

Even perfect assembled genome sequences are mostly useless if they are not complemented by biological relevant information. This information includes location, structure and function of genes and gene models. The process of assigning biological

information to pure sequence data is called annotation. A high quality annotation is expensive and labor-intensive, so most genome projects rely on automated gene annotation, but still requires editing by human experts. The basic steps are described below.

First annotation step

First of all, it should be noted that annotation strongly depends on the quality of the genome assembly. The automated annotation process can roughly be divided in two phases: A computational prediction phase and a synthesizing annotation phase.

Repeat identification is the first step to mask repetitive sequences, low complexity regions and transposable elements. Excluding those repeats with tools like REPEATMASKER (Smit *et al.*, 2013) prevents numerous false annotations.

The next step is the *AB INITIO* gene prediction to identify genes using mathematical models. This is done by scanning for open reading frames (ORF) and training for species-specific traits like codon frequencies. Programs like AUGUSTUS (Stanke *et al.*, 2006) or GENEMARK (Lomsadze *et al.*, 2014) implemented this approach but are usually limited to find coding sequences (CDS). Improvements can be gained by aligning external evidence, such as transcript or protein sequences from related organisms.

Mapping ESTs and RNA-Seq data extends evidence for UTRs and provides additional splice sites. GNOMON (Souvorov *et al.*, 2010) and the PASA (Haas *et al.*, 2011) pipeline support this feature. Protein sequences from related organisms can be obtained from UniProt/SwissProt (Wasmuth and Lima, 2016) and are aligned by BLAST (Altschul *et al.*, 1990).

Second annotation step

The second annotation phase synthesizes the results of the *ab initio* prediction into a consensus set of gene annotations. Evidence from external sources are weighted and the most representative gene model is chosen. This is incorporated in automated pipelines, such as MAKER (Cantarel *et al.*, 2008), PASA and GLEAN (Elsik *et al.*, 2007). However, manual curation is advised to detect systematic problems.

Conclusion

The field of NGS technologies is constantly evolving and novel methods become a routine part in biological and medical applications. e.g., ultra deep sequencing or read lengths of several hundred kb were considered as impossible a few years ago. There are also approaches which seem to be very promising today. Beside their advantages, challenges which cannot be solved satisfactorily, may cause a future disappearance of this technology. Nevertheless, every advancement or shift in technology requires adjustments of analyzing software. Hardware- and time requirements must be considered, too. Best results can be achieved by combining different methods and tools which fulfill the desired demands.

See also: Bioinformatics Approaches for Studying Alternative Splicing. Exome Sequencing Data Analysis. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. MicroRNA and lncRNA Databases and Analysis. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Pre-Processing: A Data Preparation Step. Sequence Analysis. Transcriptome Informatics. Whole Genome Sequencing Analysis

References

- Abyzov, A., Urban, A.E., Snyder, M., Gerstein, M., 2011. CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Research* 21 (6), 974–984.
- Akogwu, I., Wang, N., Zhang, C., *et al.*, 2016. A comparative study of k-spectrum-based error correction methods for next-generation sequencing data analysis. *Human Genomics* 10 (S2), 20.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3), 403–410.
- Ashelford, K., 2011. BAMStats: An interactive desktop GUI tool for summarising next generation sequencing alignments. Available at: <http://bamstats.sourceforge.net/>.
- Batzoglou, S., Jaffe, D.B., Stanley, K., *et al.*, 2002. ARACHNE: A whole-genome shotgun assembler. *Genome Research* 12, 177–189.
- Bennett, S., 2004. Solexa Ltd. *Pharmacogenomics* 5, 433–438.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30 (15), 2114–2120.
- Bradnam, K.R., Fass, J.N., Alexandrov, A., *et al.*, 2013. Assemblathon 2: Evaluating *de novo* methods of genome assembly in three vertebrate species. *GigaScience* 2 (1), 10.
- Bray, N., Pimentel, H., Melsted, P., Pachter, L., 2015. Near-optimal RNA-Seq quantification. *arXiv*. 1505.02710.
- Bullard, J.H., Purdom, E., Hansen, K.D., Dudoit, S., 2010. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* 11 (1), 94.
- Cantarel, B.L., Korf, I., Robb, S.M., *et al.*, 2008. MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Research* 18 (1), 188–196.
- Chen, K., Wallis, J.W., McLellan, M.D., *et al.*, 2009. BreakDancer: An algorithm for high-resolution mapping of genomic structural variation. *Nature Methods* 6 (9), 677–681.
- Cock, P.J.A., Antao, T., Chang, J.T., *et al.*, 2009a. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25 (11), 1422–1423.

- Cock, P.J.A., Fields, C.J., Goto, N., Heuer, M.L., Rice, P.M., 2009b. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research* 38 (6), 1767–1771.
- Cox, M.P., Peterson, D.A., Biggs, P.J., 2010. SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. *BMC Bioinformatics* 11 (1), 485.
- Danecek, P., Auton, A., Abecasis, G., *et al.*, 2011. The variant call format and VCFtools. *Bioinformatics* 27 (15), 2156–2158.
- Deorowicz, S., Grabowski, S., 2011. Compression of DNA sequence reads in FASTQ format. *Bioinformatics* 27 (6), 860–862.
- Dobin, A., Davis, C.A., Schlesinger, F., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29 (1), 15–21.
- Earl, D., Bradnam, K., St. John, J., *et al.*, 2011. Assemblathon 1: A competitive assessment of *de novo* short read assembly methods. *Genome Research* 21 (12), 2224–2241.
- Elsik, C.G., Mackey, A.J., Reese, J.T., *et al.*, 2007. Creating a honey bee consensus gene set. *Genome Biology* 8 (1), R13.
- Frazee, A.C., Collado-Torres, L., Jaffe, A.E., Leek, J.T., 2015. Ballgown: Flexible, isoform-level differential expression analysis. R package version 2.8.4.
- Garrison, E., Marth, G., 2012. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*. 9. (q-bio.GN).
- Gnerre, S., Maccallum, I., Przybylski, D., *et al.*, 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences of the United States of America* 108 (4), 1513–1518.
- Goff, L., Trapnell, C., Kelley, D., 2013. cummeRbund: Analysis, exploration, manipulation, and visualization of Cufflinks high-throughput sequencing data. R package version 2.12.1.
- Gordon, A., 2008. FASTX-Toolkit: FASTQ/A short-reads pre-processing tools. Available at: http://hannonlab.cshl.edu/fastx_toolkit/.
- Grabherr, M.G., Haas, B.J., Yassour, M., *et al.*, 2013. Trinity: Reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nature Biotechnology* 29 (7), 644–652.
- Haas, B.J., Zeng, Q., Pearson, M.D., Cuomo, C.A., Wortman, J.R., 2011. Approaches to fungal genome annotation. *Mycology* 2 (3), 118141.
- Hansen, K.D., Brenner, S.E., Dudoit, S., 2010. Biases in Illumina tran-scriptome sequencing caused by random hexamer priming. *Nucleic Acids Research* 38 (12), 1–7.
- Hardcastle, T.J., 2012. baySeq: Empirical Bayesian analysis of patterns of differential expression in count data. R package version 2.4.1.
- Homer, N., Merriman, B., Nelson, S.F., 2009. BFAST: An alignment tool for large scale genome resequencing. *PLOS ONE* 4, 11.
- Huber, W., Carey, V.J., Gentleman, R., *et al.*, 2015. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* 12 (2), 115–121.
- Kelley, D.R., Schatz, M.C., Salzberg, S.L., 2010. Quake: Quality-aware detection and correction of sequencing errors. *Genome Biology* 11 (11), R116.
- Kim, D., Langmead, B., Salzberg, S.L., 2015. HISAT: A fast spliced aligner with low memory requirements. *Nature Methods* 12 (4), 357–360.
- Kim, D., Pertea, G., Trapnell, C., *et al.*, 2013. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology* 14 (4), R36.
- Koboldt, D.C., Chen, K., Wylie, T., *et al.*, 2009. VarScan: Variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* 25 (17), 2283–2285.
- Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L., 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10 (3), R25.
- Le, H.S., Schulz, M.H., Mccauley, B.M., Hinman, V.F., Bar-Joseph, Z., 2013. Probabilistic error correction for RNA sequencing. *Nucleic Acids Research* 41 (10), 1–11.
- Leng, N., Kendzioriski, C., 2015. EBSeq: An R package for gene and isoform differential expression analysis of RNA-seq data. R package version 1.10.0.
- Li, B., Dewey, C.N., 2011. RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* 12 (1), 323. *arXiv: NIHMS150003*.
- Li, B., Zhan, X., Wing, M.-K., *et al.*, 2013. QPLOT: A quality assessment tool for next generation sequencing data. *BioMed Research International* 2013, 1–4.
- Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009a. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25 (16), 2078–2079.
- Li, H., Ruan, J., Durbin, R., 2008a. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18, 1851–1858.
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25 (14), 1754–1760.
- Li, P., Piao, Y., Shon, H.S., Ryu, K.H., 2015. Comparing the normalization methods for the differential analysis of Illumina high-throughput RNA-Seq data. *BMC Bioinformatics* 16 (1), 347.
- Li, R., Li, Y., Kristiansen, K., Wang, J., 2008b. SOAP: Short oligonucleotide alignment program. *Bioinformatics* 24 (5), 713–714.
- Li, R., Li, Y., Fang, X., *et al.*, 2009b. SNP detection for massively parallel whole-genome resequencing SNP detection for massively parallel whole-genome resequencing. *Genome Research* 19 (6), 1124–1132.
- Liu, Y., Schröder, J., Schmidt, B., 2012. Muskiet: A multistage k-mer spectrum based error corrector for Illumina sequence data. *Bioinformatics* 29 (3), 308–315.
- Lomsadze, A., Burns, P.D., Borodovsky, M., 2014. Integration of mapped RNA-Seq reads into automatic training of eukaryotic gene finding algorithm. *Nucleic Acids Research* 42 (15), 1–8.
- Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15 (12), 550.
- Luo, R., Liu, B., Xie, Y., *et al.*, 2012. SOAPdenovo2: An empirically improved memory-efficient short-read *de novo* assembler. *GigaScience* 1 (1), 18.
- Margulies, M., Egholm, M., Altman, W.E., *et al.*, 2006. Genome sequencing in open microfabricated high density picoliter reactors. *Nature Biotechnology* 437 (7057), 376–380.
- Martin, M., 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.Journal* 17 (1), 10.
- McKenna, A., Hanna, M., Banks, E., *et al.*, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20 (9), 1297–1303.
- Mills, R., Walter, K., Stewart, C., *et al.*, 2011. Mapping copy number variation by population-scale genome sequencing. *Nature* 470 (2010), 59–65.
- Myers, E.W., 2000. A whole-genome assembly of drosophila. *Science* 287 (5461), 2196–2204.
- Ning, Z., Ning, Z., Cox, A.J., *et al.*, 2001. SSAHA: A fast search method for large DNA databases. *Genome Research* 2, 1725–1729.
- Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C., 2015. Salmon provides accurate, fast, and bias-aware transcript expression estimates using dual-phase inference. *bioRxiv* 14 (4), 417–419. *arXiv: 1308.3700*.
- Patro, R., Mount, S.M., Kingsford, C., 2014. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology* 32 (5), 462–464. *arXiv: 1505.02710*.
- Pearson, W.R., Lipman, D.J., 1988. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America* 85 (8), 2444–2448.
- Pertea, M., Kim, D., Pertea, G.M., Leek, J.T., Salzberg, S.L., 2016. Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown. *Nature Protocols* 11 (9), 1650–1667.
- Pertea, M., Pertea, G.M., Antonescu, C.M., *et al.*, 2015. StringTie enables improved reconstruction of a tran-scriptome from RNA-seq reads. *Nature Biotechnology* 33 (3), 290–295.
- Picard-Team, 2017. A set of command line tools (in Java) for manipulating high-throughput sequencing (HTS) data and formats such as SAM/BAM/CRAM and VCF. Available at: <http://broadinstitute.github.io/picard>.
- Quinlan, A.R., Clark, R.A., Sokolova, S., *et al.*, 2010. Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Research* 20, 623–635.
- Rausch, T., Zichner, T., Schlattl, A., *et al.*, 2012. DELLY: Structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* 28 (18), 333–339.
- R Core Team, 2017. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.
- Rice, P., Longden, I., Bleasby, A., 2000. EMBOS: The European molecular biology open software suite. *Trends in Genetics* 16 (1), 276–277.
- Risso, D., Ngai, J., Speed, T.P., Dudoit, S., 2014. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature Biotechnology* 32 (9), 896–902.

- Ritchie, M.E., Phipson, B., Wu, D., *et al.*, 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* 43 (7), e47.
- Robertson, G., Schein, J., Chiu, R., *et al.*, 2010. *De novo* assembly and analysis of RNA-seq data. *Nature Methods* 7 (11), 909–912.
- Robinson, M.D., McCarthy, D.J., Smyth, G.K., 2010. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26, 1.
- Sanger, F., Nicklen, S., Coulson, A.R., 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences* 74 (12), 5463–5467.
- Schmid, C.W., Deininger, P.L., 1975. Sequence organization of the human genome. *Cell* 6 (3), 345–358.
- Schmieder, R., Edwards, R., 2011. Quality control and preprocessing of metagenomic datasets. *Bioinformatics* 27 (6), 863–864.
- Schulz, M.H., Zerbino, D.R., Vingron, M., Birney, E., 2012. Oases: Robust *de novo* RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics* 28 (8), 1086–1092.
- Simpson, J.T., Durbin, R., Zerbino, D.R., *et al.*, 2012. Efficient *de novo* assembly of large genomes using compressed data structures sequence data. *Genome Research* 19, 549–556.
- Simpson, J.T., Wong, K., Jackman, S.D., *et al.*, 2009. ABySS: A parallel assembler for short read sequence data. *Genome Research* 19, 1117–1123.
- Smit, A., Hubley R., Green P., 2013. RepeatMasker Open-4.0. Available at: <http://www.repeatmasker.org>.
- Song, L., Florea, L., 2015. Rcorrector: Efficient and accurate error correction for Illumina RNA-seq reads. *GigaScience* 4 (1), 48.
- Souvorov, A., Kapustin, Y., Kiryutin, B., *et al.*, 2010. GnomonNCBI eukaryotic gene prediction tool. National Center for Biotechnology Information. 1–24.
- Stanke, M., Schoffmann, O., Morgenstern, B., Waack, S., 2006. Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources. *BMC Bioinformatics* 7 (1), 62.
- Trapnell, C., Roberts, A., Goff, L., *et al.*, 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols* 7 (3), 562–578.
- Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* 28 (5), 511–515.
- Treangen, T.J., Salzberg, S.L., 2013. Repetitive DNA and next-generation sequencing: Computational challenges and solutions. *Nature reviews. Genetics* 13 (1), 36–46.
- Wasmuth, E.V., Lima, C.D., 2016. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (2016), 1–12.
- Xie, Y., Wu, G., Tang, J., *et al.*, 2014. SOAPdenovo-Trans: *De novo* transcriptome assembly with short RNA-Seq reads. *Bioinformatics* 30 (12), 1660–1666.
- Yang, X., Dorman, K.S., Aluru, S., 2010. Reptile: Representative tiling for short read error correction. *Bioinformatics* 26 (20), 2526–2533.
- Ye, K., Schulz, M.H., Long, Q., Apweiler, R., Ning, Z., 2009. Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* 25 (21), 2865–2871.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome Research* 18 (5), 821–829.

Relevant Websites

<http://maq.sourceforge.net/fastq.shtml>
 FASTQ Format
 Maq
 SourceForge.

<https://github.com/samtools/hts-specs>
 SamTools · GitHub.

<http://assemblathon.org/>
 The Assemblathon.

Transcriptomic Data Normalization

Fatemeh Vafaee, University of New South Wales, Sydney, NSW, Australia

Hamed Dashti, Sharif University of Technology, Tehran, Iran

Hamid Alinejad-Rokny, University of New South Wales, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Cells are built, developed, and reproduced by genetic instructions stored in the form of DNA, a long double-stranded molecule of nucleic acids. DNA is *transcribed* into RNA, which is then *translated* into proteins to carry out the duties specified by the instructions encoded in the genes. The complete set of RNA transcripts produced by the genome, under a specific condition or in a specific cell, is called transcriptome (Adams, 2008). Transcriptomics is the study of the transcriptome through which researchers gain insight into gene activities and better understand how cells normally function or how changes in RNA activities can contribute to disease (Kalia and Gupta, 2005). Transcriptomes can basically enable researchers to depict a comprehensive, genome-wide picture of gene activities across different cells (Adams, 2008). In addition, comparative transcriptomics facilitates the study of the evolution of gene regulation and provides deeper understanding of the species conservation at the molecular level (Breschi *et al.*, 2017).

Transcriptomics has been characterized by technological innovations that continuously transform the field. Key contemporary transcriptomic technologies include DNA microarrays and NGS technologies called RNA-seq. Microarrays measure the expressions of a predetermined set of sequences while RNA-seq uses high-throughput sequencing to capture all sequences (Lowe *et al.*, 2017a). Transcription can also be studied at the level of individual cells using single-cell transcriptomic techniques, for example, single-cell RNA-seq (Kanter and Kalisky, 2015).

Analysis of microarray or RNA-seq data relies on the hypothesis that the measured values (e.g., florescent intensities or read counts) for each RNA represent its relative expression level in a specific sample under study. To compare RNAs across multiple samples and to reveal biologically relevant patterns of gene expression, transcriptomic data shall undergo a preprocessing transformation step referred to as normalization. Normalization adjusts for technical biases and other systematic variations to balance intensities or read counts appropriately so that meaningful biological comparisons can be made (Quackenbush, 2002; Mortazavi *et al.*, 2008). Although microarray and RNA-seq experiments carry different sources of systematic variations, it has been shown that normalization is an essential step in the analysis of expression data derived from both technologies (e.g., Park *et al.*, 2003a; Bullard *et al.*, 2010).

Several normalization methods developed during the past decades. They mainly focus on the mean and the variance, which are quantifying the center and the spread of the data, respectively. These methods try to adjust mean and variance between different samples. Fig. 1, for instance, shows how normalization would adjust the distribution of RNA expressions across different samples.

Microarray Data Normalization

A microarray is a laboratory tool used to quantify the expression of tens of thousands of genes simultaneously. DNA microarrays (a.k.a. biochips or DNA chips) are glass, plastic, or silicon-based slides printed with thousands of microscopic spots in defined positions, with each spot containing picomoles (10^{-12} mol) of a specific DNA sequence. The DNA molecules attached to a microarray slide act as probes to capture messenger RNA (mRNA) transcripts expressed by genes.

A microarray experiment typically starts with purification of mRNAs from the biological sample (i.e., RNA isolation). RNAs are converted into complementary DNA (cDNA) using reverse transcriptase enzyme and labeled with fluorescent dyes (e.g., cyanine dyes, a synthetic dye family belonging to polymethine group). After labeling, cDNA molecules are allowed to bind or *hybridize* to the DNA probes on the microarray slide. The microarray is then scanned to visualize fluorescence intensities that measure the expression of each gene printed on the slide – that is, brighter spots indicates higher level of expressions.

DNA microarrays come in *two-channel* versus *one-channel* designs. Two-channel or two-color microarrays are hybridized with cDNA prepared from two samples to be compared (e.g., diseased versus healthy tissues) and labeled with two different fluorophores (i.e., Cy3 and Cy5 for green and red fluorescence emission, respectively). Relative intensities of each fluorophore through a ratio-based analysis can then identify downregulated and upregulated genes. A single-channel or one-color microarray is hybridized with one sample and the array provides intensity data for each probe as the relative level of hybridization with the labeled cDNA or the expression level of the corresponding gene. Fig. 2 illustrates key steps of a typical DNA microarray experiment and analysis of resulting expression values.

Several technical errors may occur during the microarray experimental procedure such as irregular spot printing, nonuniform intensity of the fluorescent compound, dusty arrays, purification errors, difference in efficiency of labeling via fluorescent dyes, hybridization efficiencies, and systematic biases in quantified expression levels (Quackenbush, 2002; Park *et al.*, 2003b; Yang *et al.*, 2002). These artifacts have bearings on capturing data leading to different measurements of same expression values. Hence, this is important to eliminate technical noises prior to any downstream analysis and normalization plays an important role in reducing potential systematic noises.

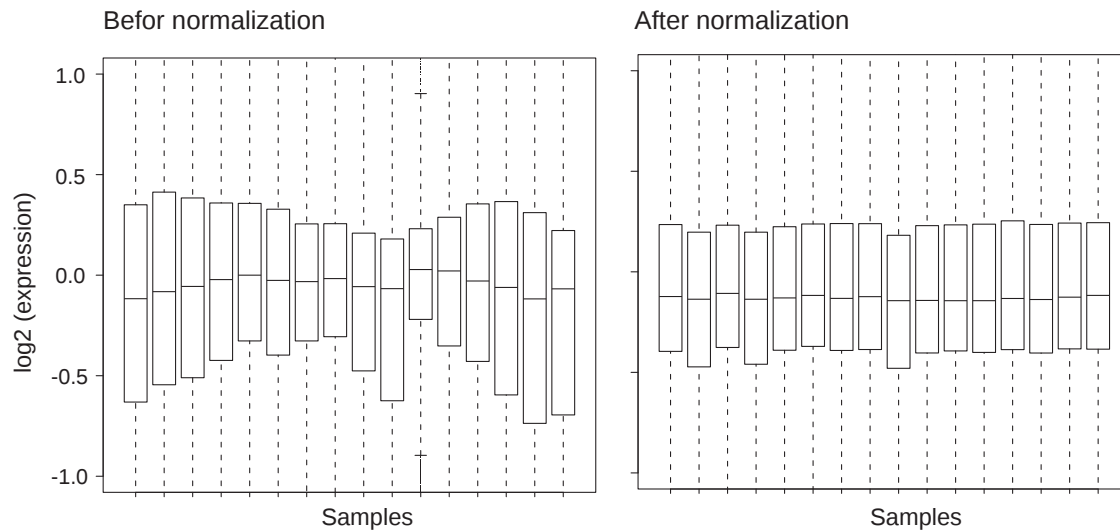


Fig. 1 Sample boxplots. Box plots of $\log_2(\text{expression})$ array data before and after normalization. Dataset: GSE4158, normalization method: Quantile normalization implemented using “Rnits” R package (Sangurdekar, 2014).

Key steps of microarray experiment & analysis pipeline

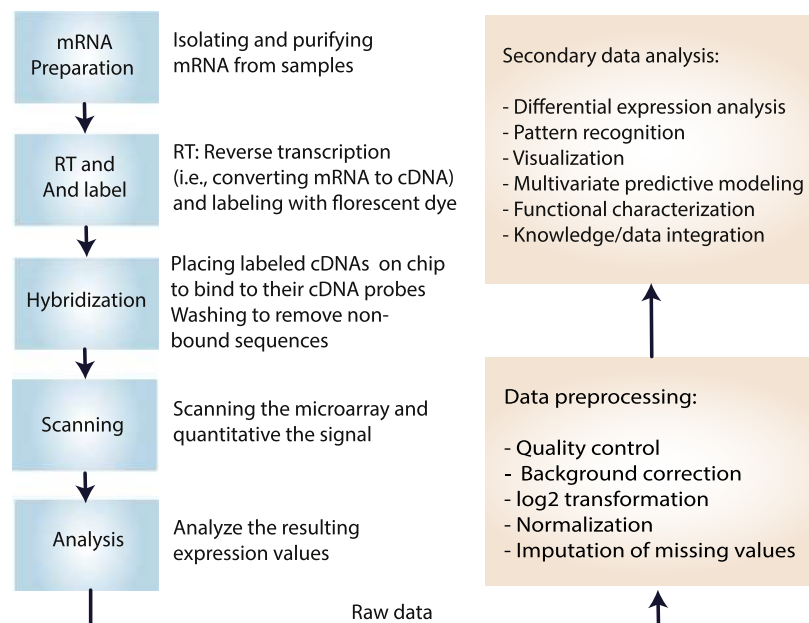


Fig. 2 Microarray workflow. Key steps of performing a microarray experiment for gene expression studies after sample preparation. Raw expression data will be preprocessed to assure the data quality and then undergo various downstream analyses.

There are several normalization methods. Some have been developed for particular platforms and the analysis may be proprietary for commercial platforms. Yet, several freely accessible resources are available to normalize microarray data including various Bioconductor R packages (see “Relevant Websites section”). Selecting the best normalization method is not an obvious decision and suitable choices may vary depending on the data under study. Complex methods do not guarantee better performance and background assumptions may add bias and eliminate important data (Park *et al.*, 2003a). Nonetheless, there are several scientific works comparing the performance of different normalization methods with respect to different performance criteria (Smyth and Speed, 2003; Bolstad *et al.*, 2003; Park *et al.*, 2003b) which can help to opt for an appropriate normalization method based on the experimental platform and the quality of the data under study.

MA plot (Dudoit *et al.*, 2002) is a visualization tool traditionally used to assess the performance of normalization methods by revealing systematic intensity-dependent effects in the measurements taken from two samples. We denote two, that is, query and

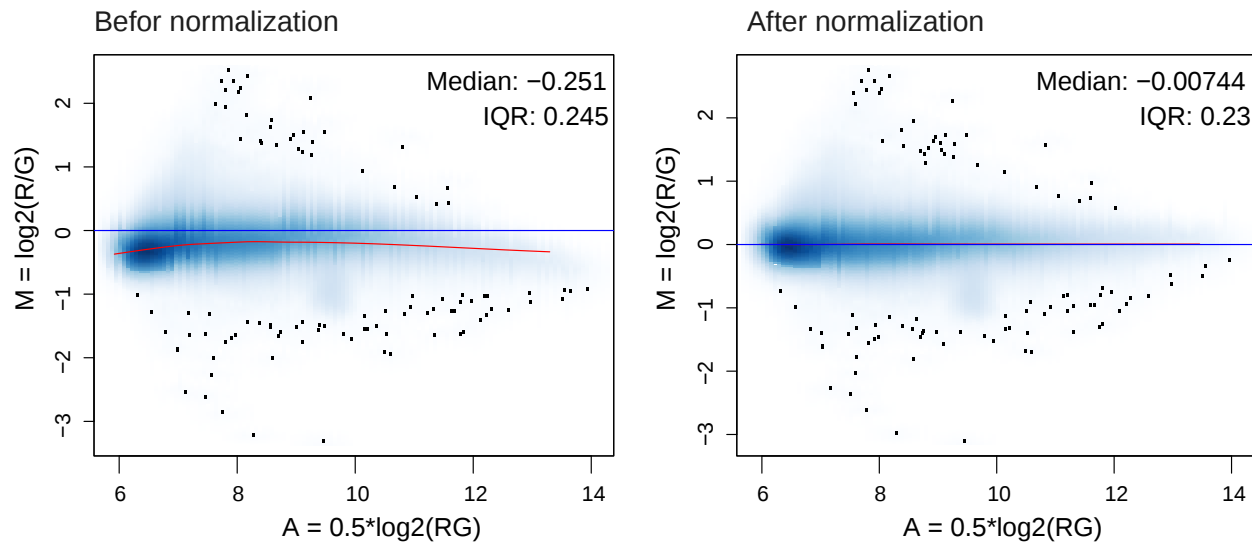


Fig. 3 Sample MA plots. MA plots before and after normalization using smoothScatter method which produces a smoothed color density representation of a scatterplot. Red line is the lowest smoothed line to visually show the bias. Dataset: Dilution (array 20B v 10A) from “affydata” R package. Reproduced from Gautier, L. 2011. Affydata: Affymetrix Data for Demonstration Purpose. R package version, 1, 18. Normalization method: Quantile normalization implemented using “affy” R package. Reproduced from Gautier, L., Cope, L., Bolstad, B.M., Irizarry, R.A., 2004. Affy – Analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20, 307–315.

reference, samples as R and G (for the red and green colors commonly used to represent Cy5 and Cy3 intensities in two-channel microarrays). MA plot is a scatter plot whose y -axis and x -axis respectively display $M = \log_2(R_i/G_i)$ and $A = \log_2(R_i * G_i)$ where R_i and G_i represent the intensity of the i th gene in R and G samples. An underlying assumption in many microarray gene expression experiments is that most of the genes do not change in their expressions, and thus the majority of the points would have 0 y -coordinate and lie on the x -axis, that is, $M = \log_2(1) = 0$. If this is not the case, a proper normalization method should be applied to the data to adjust imbalanced intensities. **Fig. 3** shows MA plots for a sample before and after normalization.

Lowess Normalization

Lowess/loess normalization is one of the earliest developments in microarray normalization proposed by Terry Speed’s group (Dudoit *et al.*, 2002; Yang *et al.*, 2002). They observed nonlinear bias in two-color log ratios changing as a function of average intensities in two-channel microarrays (as seen in MA plots). Similar bias has been also observed in single-channel arrays when plotting intensities in one chip against intensities averaged over all chips (Reimers, 2010).

The purpose of lowess normalization is to estimate this bias using a nonparametric curve known as *local weighted regression* (lowess). At each point on MA plot, M value is adjusted by subtracting the estimated bias (the height of the loess curve) at the same A value, that is, $\log_2(R/G) = \log_2(R/G) - c(A)$, where $c(A)$ is the lowess fit to MA plot. Lowess normalization operates on individual chips (within-chip normalization), but was intended to make measures comparable across chips as well (between-chip normalization). It can perform robust locally linear fits not affected by outliers in the MA plot (Yang *et al.*, 2002).

Lowess normalization was further improved by Speed’s group to account for spatial dependence in dye biases (Yang *et al.*, 2002) which was based on the observation that there were pronounced differences in the lowess curves fit to log ratios in different regions of the same chip. Accordingly, they proposed *within print-tip group normalization*, which aims at fitting separate lowess curves to each “print tip,” that is, $\log_2(R/G) = \log_2(R/G) - c_i(A)$, where $c_i(A)$ is the lowess fit to MA plot for the i th print tip only. A print tip is a set of probes on a common grid of a robotically printed cDNA array.

Researchers continued to further complicate lowess normalization adjusting for other possible biases. Examples include rescaling variation within each print-tip group or estimating a plate order effect (see, e.g., Smyth, 2002; Smyth and Speed, 2003).

Quantile Normalization

Quantile normalization is a simple nonparametric normalization method initially proposed for single-channel arrays (Bolstad *et al.*, 2003). It is a between-array normalization method that makes the distribution of all arrays identical in statistical properties. The algorithm maps every expression value on each chip to the corresponding quantile of a reference distribution that is determined by pooling across distributions of all individual chips. It is motivated by the idea that a quantile–quantile plot shows that the distribution of two data vectors is the same only if the plot is a straight diagonal line.

While being simple and fast, quantile normalization could outperform most of the more complex contemporary methods including loess. It has been widely used for normalization of single-channel high intensity oligonucleotide arrays, such as

Affymetrix arrays (one of the most frequently used platforms for microarray experiments), and is also applicable to dual-channel chips (Reimers, 2010). Quantile normalization was made available as the default normalization method in the widely used “affy” package of R Bioconductor, which contains functions for exploratory oligonucleotide array analysis (Gautier *et al.*, 2004).

Despite its clear advantages, quantile normalization explicitly assumes that the distribution of gene expression measures is identical across the samples. This can be violated when there are biologically meaningful differences between samples’ transcriptomic distributions (e.g., when studying treatments with severe effects on the transcriptome, or cancer samples with severe genomic aberrations). Also, in its canonical form, quantile normalization forces genes with high intensities into the reference distribution shape which may reduce biological differences (Reimers, 2010). The latter has been addressed in a later version of the “affy” package.

Overall, due to its simplicity, efficiency, generality, and availability, quantile normalization has become a comparatively popular choice of normalization for microarray data.

Variance Stabilization

Various widely used standard statistical methods (e.g., regression, ANOVA) assume that data is normally or symmetrically distributed with a constant variance not depending on the mean of the data. However, microarray data fail to comply with this canonical assumption underlying standard methodologies (Rocke and Durbin, 2001). It is therefore a common practice to stabilize the variance of microarray expression values by transforming the expression values to a logarithmic base 2 scale. While logarithmic transformation makes variations among measurements that span several orders of magnitudes comparable, it can inflate variances of low-level, near-background observations (Durbin *et al.*, 2002), which may necessitate removal of low-intensity probes a priori. To address this issue, (Durbin *et al.*, 2002) introduced a transformation that stabilizes the variance of microarray data across the full range of expression values without needing to remove low-level observations. For more details and other options, interested readers may wish to consult (Durbin *et al.*, 2002; Huber *et al.*, 2002; Durbin and Rocke, 2004; Lin *et al.*, 2008).

Nonetheless, while more sophisticated methods may provide more coherent variance stabilization, the ultimate gain is possibly small compared to the simplicity of log transformation (Speed, 2001) and thus, researchers usually stick by log2 transformation.

RNA-seq Data Normalization

RNA-seq uses next-generation sequencing (NGS) to quantify the presence and abundance of RNAs in a biological sample at a specific moment of time. It is a revolutionary tool that provides a comprehensive view of the transcriptome, and enables the detection of novel transcripts and characterization of alternative splicing (Wang *et al.*, 2009). It is rapidly replacing gene expression microarrays for whole-genome transcriptome profiling (Mortazavi *et al.*, 2008; Nagalakshmi *et al.*, 2010). That is basically due to limitations involved in microarray-based experiments that have been resolved using RNA-seq technologies. For example, RNA-seq intrinsically avoids errors introduced during hybridization of microarrays as it does not depend on genome annotation for prior probe selection. RNA-seq also captures a broader dynamic range that ensures the detection of more differentially expressed genes with higher fold-change compared to microarrays (Zhao *et al.*, 2014).

A typical RNA-seq experiment consists of RNA purification and isolation, library preparation (i.e., converting RNA to cDNA and adding sequencing adapters), sequencing using a NGS platform, and analysis of resulting sequencing reads (Fig. 4(a)). RNA-seq analysis pipeline usually involves demultiplexing and trimming sequencing reads, mapping sequencing reads to a reference transcriptome, annotating transcripts to which reads are mapped, counting mapped reads to estimate the abundance of transcripts, normalizing reads, and performing downstream statistical analyses – for example, identifying differentially expressed genes, visualization, etc. (Fig. 4(b)).

Although RNA-seq experiments do not suffer from technical noises inherent to microarray technologies, they are still subject to some systematic variations such as library size (i.e., sequencing depth) differences between samples, as well as transcript length bias and GC content within a specific sample (Oshlack and Wakefield, 2009b). It is therefore essential to normalize data in order to adjust for such biases. Several normalization methods have been developed for RNA-seq data. Some of the most frequently used ones have been reviewed in here.

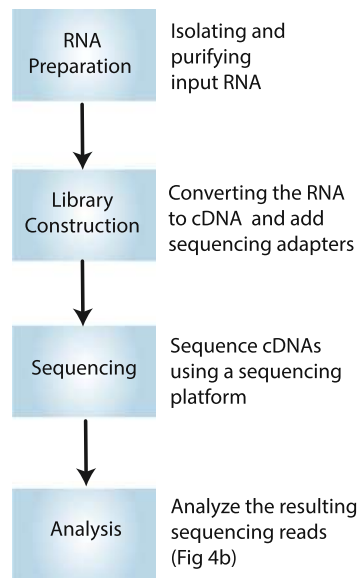
Total Count

Differences in the library size (i.e., sequencing depth) account for the most apparent source of intersample variations in RNA-seq experimentations. Therefore, the simplest form of between-sample normalization has been realized by scaling raw read counts in each sample (or sequencing “lane”) by a sample-specific factor (Dillies *et al.*, 2013).

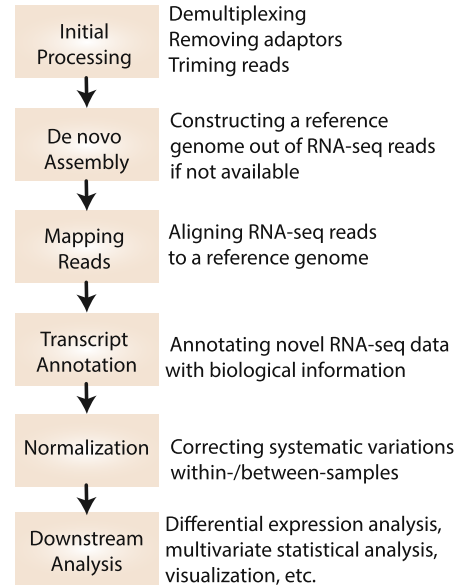
In total count (TC) normalization, for each sample, gene counts are divided by the sample’s library size (i.e., total number of mapped reads associated with that sample) and multiplied by the mean TC across all the samples of the dataset.

Upper Quartile

Upper quartile (UQ) normalization is very similar to TC and only replaces the TCs (i.e., library size) by the UQ (75th percentile) of read counts, after excluding genes with zero reads across all lanes (Bullard *et al.*, 2010).

Key steps of RNA-seq experiment

(a)

Key steps of RNA-seq analysis pipeline

(b)

Fig. 4 RNA-seq workflow. (a) Key steps of a typical RNA-seq workflow after designing an RNA-seq experiment. (b) Main steps of RNA-seq analysis pipeline. For more details refer to RNA-seqlopedia (<https://rnaseq.uoregon.edu>).

Reads Per Kilo Base Per Million Mapped Reads

This approach was aimed to facilitate comparison of transcript levels both within and between samples as it rescales gene counts to correct for differences in both library sizes and gene length (Mortazavi *et al.*, 2008). Two replicate samples with different total library size would have proportionally different sequenced reads mapped to the same gene. Also, longer transcripts can potentially produce more fragments than shorter ones. To adjust for these variations, reads per kilo base per million mapped reads (RPKM) divides gene counts by the library size as well as the length of the transcript in kilobase. It has been shown that adjusting differences in gene length can potentially introduce bias in per-gene variances, in particular for lowly expressed genes (Oshlack and Wakefield, 2009a). Nonetheless, RPKM is a widely held choice in many practical applications. ERANGE tool (website available at “Relevant Websites section”) implements RPKM normalization method (Mortazavi *et al.*, 2008).

Trimmed Mean of M-Values

This normalization method assumes that most genes are not differentially expressed. Trimmed Mean of M-values (TMM) considers one sample as a reference sample and the others as test samples. For each test sample, it then computes the TMM normalization factor as the weighted mean of log ratios (i.e., *M* values) between this test and the reference, after trimming genes based on a predefined percentage of *M* values and absolute intensities. This TMM factor should be close to 1 according to the underlying assumption (i.e., low proportion of differentially expressed genes). Otherwise, the library sizes shall be corrected by TMM normalization factor to satisfy the assumption. These normalization factors are rescaled by the mean of the normalized library sizes. Raw read counts are then divided by rescaled normalization factors to obtain normalized read counts. TMM normalization has been implemented in ‘edgeR’ Bioconductor R package, version 2.5 (Robinson *et al.*, 2010).

DESeq

Similar to TMM, DESeq normalization method assumes that most genes are not differentially expressed – that is, have the same read counts across all samples (Anders and Huber, 2010). For each gene in a given sample, a DESeq scaling factor is computed as the median of the ratio of the gene’s read count over its geometric mean across all samples. The underlying hypothesis (i.e., low number of differentially expressed genes) assumes that this ratio is close to 1. Accordingly, the median of this ratio for the sample provides an estimate of the normalization factor to be applied to all read counts of this sample to fulfill the assumption. DESeq normalization is included in Bioconductor R package DESeq, version 1.30.0 (Anders and Huber, 2010). DESeq normalization factor adjusts for differences in sequencing depth between samples. An improved version of this method, implemented in DESeq2

Bioconductor package version 1.18.1 (Love *et al.*, 2014), calculates *gene-specific* normalization factors to account for further sources of technical biases such as differing dependence on GC content, gene length, etc. Interested readers can consult (Love *et al.*, 2014) for more details.

Single-Cell RNA-seq

Single-cell RNA sequencing (scRNA-seq) is becoming a powerful tool for high-throughput, high-resolution transcriptomic analysis of individual cells yielding new insights into the cellular diversity underlying homogeneous populations (Tang *et al.*, 2009). It has enabled scientists to investigate fundamental research questions not previously addressable by bulk-level experiments – for example, identifying novel cell types, unraveling tumor heterogeneity, and finding novel microbiome identities (Shapiro *et al.*, 2013).

Currently, scRNA-seq experimental pipeline typically includes isolation of single cell and RNA, reverse transcription (i.e., creation of cDNA from RNA templates), amplification, library generation, sequencing, data preprocessing and downstream analyses. Normalization is a critical data preprocessing step to adjust for unwanted biological effects and technical biases specific to the technology of interest. A distinctive feature of RNA-seq data is “zero inflation,” that is, the high proportion of zero read counts (Pierson and Yau, 2015). Single-cell RNA-seq data are also highly heterogeneous as individual cells are captured from potentially very different cell types (Finak *et al.*, 2015).

Nonetheless, the majority of currently available scRNA-seq analysis tools rely on normalization methods developed for bulk RNA-seq (Vallejos *et al.*, 2017). Given that the assumptions underlying most bulk RNA-seq normalization techniques do not necessarily apply to the single-cell setting, using existing normalization methods for scRNA-seq data may introduce artifacts that bias downstream analyses (Bacher *et al.*, 2017).

Recent methods have been proposed for scRNA-seq normalization specifically. Some of these methods rely on RNA *spike-in* controls (i.e., RNA transcripts of known sequence and quantity used to calibrate expression measurements) such as GRM (Ding *et al.*, 2015) and SAMstrt (Katayama *et al.*, 2013). However, it has been recently observed that the performance of spike-ins in scRNA-seq is often compromised, and have not been used for normalization in many laboratories (Bacher *et al.*, 2017). Here, we therefore only elaborate on two recently developed, state-of-the-art scRNA-seq normalization methods that do not require spike-ins – that is, Scnorm (Bacher *et al.*, 2017) and Scrn (Lun *et al.*, 2016). Nevertheless, bulk-based approaches as well as methods particularly tailored to scRNA-seq data have been compared in a recent review (Vallejos *et al.*, 2017) to which interested readers are referred for further details.

Scran

This method estimates cell-specific size factors from pools of cells to enhance normalization accuracy in the presence of zero inflation and unbalanced differential expression of genes across groups of cells. Scran normalization consists of the following key steps (Lun *et al.*, 2016):

- (1) Defining pools of cells: to alleviate the problem of data heterogeneity, scran preclusters cells into smaller, more homogeneous sets and perform normalization for each cluster before between-cluster normalization.
- (2) Summing expression values across all cells in each pool: summation across cells results in fewer zeroes in each pool, which implies that the subsequent normalization is less susceptible to zero inflation.
- (3) Normalizing pool-based size factors (i.e., summed expression values) against an average reference.
- (4) Deconvolving the pool-based size factors into the size factors for its constituent cells to adjust for cell-specific biases.

The scran normalization method is publicly available using the `computeSumFactors` function in the “scran” package, version 1.6.6 on Bioconductor (see “Relevant Websites section”).

SCnorm

This method is based on the observation that scRNA-seq data shows systematic biases in the relationship between transcript-specific expression and sequencing depth – referred to as count-depth relationship – that cannot be adjusted by a single global scale factor common to all genes in a cell. Accordingly, normalization methods based upon global scale factors can contribute to overcorrection of weakly/moderately expressed genes and undernormalization of highly expressed genes (Bacher *et al.*, 2017).

To address this, SCnorm performs the following key steps:

- (1) Estimating the dependence of transcript expression on sequencing depth for every gene using quantile regression.
- (2) Grouping genes with similar dependence.
- (3) Estimating scale factors within each group through a second quantile regression.
- (4) Providing normalized estimates of expressions by performing within-group adjustment for sequencing depth using the estimated scale factors.

The SCnorm normalization is publicly available as an R package “SCnorm” version 1.0.0 on Bioconductor (see “Relevant Websites section”).

Conclusions

Transcriptomics has paved the way for a comprehensive understanding of how genes are expressed and interconnected. Over the last three decades, methodological breakthroughs have repeatedly revolutionized transcriptome profiling and redefined what is possible to investigate. Integration of transcriptomic data with other omics is giving an increasingly integrated view of cellular complexities facilitating holistic approaches to biomedical research (Lowe *et al.*, 2017b).

Normalization of transcriptomic data is an essential preprocessing step aimed at correcting unwanted biological effects and technical noises prior to any downstream analysis. Normalization methods shall be chosen according to the undertaken technology and can be platform-specific. While there is no consensus on the best normalization methods across different transcriptomic technologies, several efforts have been taken to develop additional robust and effective normalization techniques and to systematically assess their performance on individual data sets.

Author Contributions

FV designed and supervised the study. FV, HD, and HAR wrote and revised the article. FV prepared the final version of the article. HD and HAR generated Fig. 1 and FV generated Figs. 2–4. All authors have read and approved the final version of the article.

See also: Analyzing Transcriptome-Phenotype Correlations. Characterization of Transcriptional Activities. Comparative Transcriptomics Analysis. Data Mining: Outlier Detection. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequence Analysis. Pre-Processing: A Data Preparation Step. Regulation of Gene Expression. Statistical and Probabilistic Approaches to Predict Protein Abundance. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation. Transcriptome Informatics. Utilising IPG-IEF to Identify Differentially-Expressed Proteins

References

- Adams, J., 2008. Transcriptome: Connecting the genome to gene function. *Nature Education* 1 (1), 195.
- Anders, S., Huber, W., 2010. Differential expression analysis for sequence count data. *Genome Biology* 11, R106.
- Bacher, R., Chu, L.-F., Leng, N., *et al.*, 2017. SCnorm: Robust normalization of single-cell RNA-seq data. *Nature Methods* 14, 584–586.
- Bolstad, B.M., Irizarry, R.A., Åstrand, M., Speed, T.P., 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19, 185–193.
- Breschi, A., Gingeras, T.R., Guigó, R., 2017. Comparative transcriptomics in human and mouse. *Nature Reviews Genetics* 18, 425–440.
- Bullard, J.H., Purdom, E., Hansen, K.D., Dudoit, S., 2010. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* 11.
- Dillies, M., Rau, A., Aubert, J., *et al.*, 2013. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics* 14, 671–683.
- Ding, B., Zheng, L., Zhu, Y., *et al.*, 2015. Normalization and noise reduction for single cell RNA-seq experiments. *Bioinformatics* 31, 2225–2227.
- Dudoit, S., Yang, Y.H., Callow, M.J., Speed, T.P., 2002. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*. 111–139.
- Durbin, B.P., Hardin, J.S., Hawkins, D.M., Rocke, D.M., 2002. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics* 18, S105–S110.
- Durbin, B.P., Rocke, D.M., 2004. Variance-stabilizing transformations for two-color microarrays. *Bioinformatics* 20, 660–667.
- Finak, G., McDavid, A., Yajima, M., *et al.*, 2015. MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology* 16, 278.
- Gautier, L., Cope, L., Bolstad, B.M., Irizarry, R.A., 2004. Affy – Analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20, 307–315.
- Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A., Vingron, M., 2002. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* 18, S96–S104.
- Kalia, A., Gupta, R.P., 2005. Proteomics: A paradigm shift. *Critical Reviews in Biotechnology*. 173–198.
- Kanter, I., Kalisky, T., 2015. Single cell transcriptomics: Methods and applications. *Frontiers in Oncology* 5.
- Katayama, S., Töhönen, V., Linnarsson, S., Kere, J., 2013. SAMstr: Statistical test for differential expression in single-cell transcriptome with spike-in normalization. *Bioinformatics* 29, 2943–2945.
- Lin, S.M., Du, P., Huber, W., Kibbe, W.A., 2008. Model-based variance-stabilizing transformation for Illumina microarray data. *Nucleic Acids Research* 36, e11.
- Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15, 550.
- Lowe, R., Shirley, N., Bleackley, M., Dolan, S., Shafe, T., 2017a. Transcriptomics technologies. *PLOS Computational Biology* 13.
- Lowe, R., Shirley, N., Bleackley, M., Dolan, S., Shafe, T., 2017b. Transcriptomics technologies. *PLOS Computational Biology* 13, e1005457.
- Lun, A.T., Bach, K., Marioni, J.C., 2016. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology* 17, 75.
- Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B., 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods* 5, 621–628.
- Nagalakshmi, U., Waern, K., Snyder, M., 2010. RNA-Seq: A method for comprehensive transcriptome analysis. *Current Protocols in Molecular Biology*. 4.11. 1–4.11. 13.
- Oshlack, A., Wakefield, M., 2009a. Transcript length bias in RNA-seq data confounds systems biology. *Biology Direct* 4.
- Oshlack, A., Wakefield, M.J., 2009b. Transcript length bias in RNA-seq data confounds systems biology. *Biology Direct* 4, 14.
- Park, T., Yi, S.-G., Kang, S.-H., *et al.*, 2003a. Evaluation of normalization methods for microarray data. *BMC Bioinformatics* 4.
- Park, T., Yi, S.-G., Kang, S.-H., *et al.*, 2003b. Evaluation of normalization methods for microarray data. *BMC Bioinformatics* 4, 33.
- Pierson, E., Yau, C., 2015. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology* 16, 241.
- Quackenbush, J., 2002. Microarray data normalization and transformation. *Nature Genetics* 32, 496–501.
- Reimers, M., 2010. Making informed choices about microarray data analysis. *PLOS Computational Biology* 6, e1000786.

- Robinson, M., McCarthy, D., Smyth, G., 2010. EdgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26, 139–140.
- Rocke, D.M., Durbin, B., 2001. A model for measurement error for gene expression arrays. *Journal of Computational Biology* 8, 557–569.
- Sangurdekar, D.P., (2014). *Rnits: R Normalization and Inference of Time Series data*. R package version 1.12.0.
- Shapiro, E., Biezuner, T., Linnarsson, S., 2013. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics* 14, 618–630.
- Smyth, G.K., 2002. Print-order normalization of cDNA microarray. Available at: <http://www.statsci.org/smyth/pubs/porder/porder.html>.
- Smyth, G.K., Speed, T., 2003. Normalization of cDNA microarray data. *Methods* 31, 265–273.
- Speed, T. 2001. Always log spot intensities and ratios. Speed Group Microarray Page: Available at: <https://www.stat.berkeley.edu/~terry/zarray/Html/log.html>.
- Tang, F., Barbacioru, C., Wang, Y., *et al.*, 2009. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* 6, 377–382.
- Vallejos, C.A., Risso, D., Scialdone, A., Dudoit, S., Marioni, J.C., 2017. Normalizing single-cell RNA sequencing data: Challenges and opportunities. *Nature Methods* 14 (6), 565–571.
- Wang, Z., Gerstein, M., Snyder, M., 2009. RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews Genetics* 10, 57–63.
- Yang, Y.H., Dudoit, S., Luu, P., *et al.*, 2002. Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research* 30, e15.
- Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., Liu, X., 2014. Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLOS ONE* 9.

Relevant Websites

<https://bioconductor.org/packages>
 Bioconductor all packages.

<https://bioconductor.org/packages/3.6/bioc/html/SCnorm.html>
 Bioconductor
 SCnorm.

<http://bioconductor.org/packages/scrna>
 Bioconductor scrna.

<http://woldlab.caltech.edu/maseq>
 RNASeq
 Wold Lab
 Caltech.

Differential Expression From Microarray and RNA-seq Experiments

Marc Delord, Université Paris Diderot-Paris 7, Sorbone Paris Cité, Paris, France

© 2019 Elsevier Inc. All rights reserved.

Introduction

Gene transcription is the process by which genes are copied into different types of RNAs, such as messenger RNA (mRNA) leading to the synthesis of proteins through translation, or noncoding RNA such as micro RNA (mRNA), transfer RNA (tRNA), or ribosomal RNA (rRNA) (Fig. 3) (Bolsover *et al.*, 1997). In the past decades, various methods have been developed for quantifying mRNA from biological samples at different scale, including quantitative reverse transcription-polymerase chain reaction (qRT-PCR) (Heid *et al.*, 1996), serial analyses of gene expression (SAGE) (Velculescu *et al.*, 1995), hybridization of cDNA (Schena *et al.*, 1995) or oligonucleotide chips (Lockhart *et al.*, 1996), and high-throughput RNA sequencing (RNA-seq) (Nagalakshmi *et al.*, 2008; Holt and Jones, 2008; Wang *et al.*, 2009) (Fig. 1).

Differential expression refers to the supervised analysis of gene expression data. This definition supposes therefore that expression of one or more genes has been quantified in various experimental conditions, or with respect to a quantitative trait or a censored time to event outcome. In large scale gene profiling experiments, the expression of thousands of genes or transcribed regions is quantified from relatively few biological samples. This situation where the number of measured items far exceeds the sample size is known as the *large p small n paradigm* (West, 2003; Kosorok and Ma, 2007) as it reverses the standard frequentist paradigm in which experiments are designed and powered in order to highlight a meaningful difference of one or few response variables observed in various experimental conditions.

Gene profiling experiments are therefore relatively underpowered and can be considered numerically as ill-posed problems, the uniqueness and stability of results being not guaranteed (Kabanikhin, 2008; Philippe, 2008). Besides, allowances that have to be made for the amount of multiple testing by the control of the type I error rate (familywise error rate or false discovery rate (FDR) (Benjamini and Hochberg, 1995; Dudoit *et al.*, 2003)) also reduce the statistical power available for detecting differential expression at the gene level. Differential analysis of gene expression is therefore a challenging task from a statistical point of view.

The challenge of the analysis of gene expression data and more generally high-throughput genomic data has favored the development and the spreading of statistical methods and procedures that are able to take advantage of the massively parallel structure of genomic data. By *borrowing strength* across features (Morris, 1983; Kerr and Churchill, 2001; Newton *et al.*, 2001; Smyth, 2004), the so-called empirical Bayes approaches (Efron and Morris, 1973; Efron and Morris, 1977; Carlin and Louis, 1996) outperform the naive feature-by-feature estimations. The idea behind these methods stems from the pioneering work of Stein Charles, (1955) who shows that the simultaneous estimate of multiple parameters is more accurate on average than sequential optimal estimations.

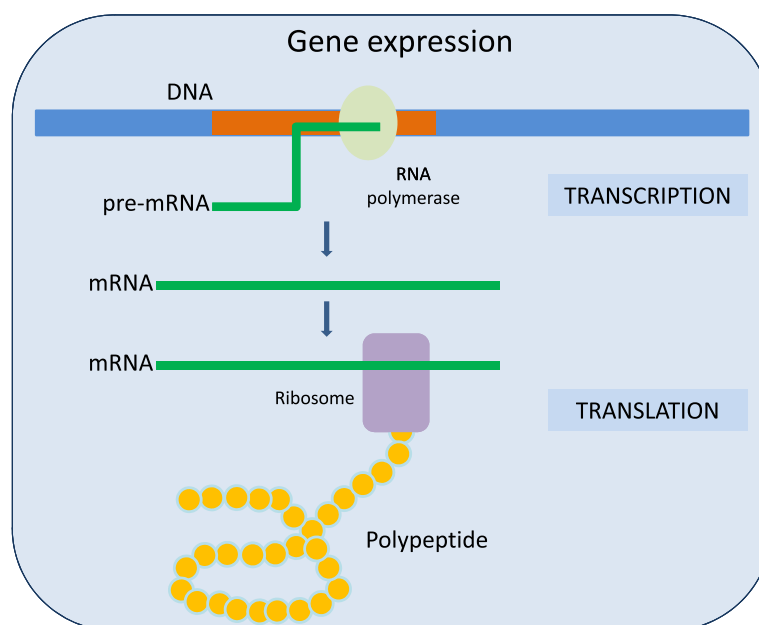


Fig. 1 Gene expression: Transcription and translation are the two steps of gene expression. During transcription, DNA is used as a template by RNA polymerase (green) to produce a pre-mRNA transcript. The pre-mRNA is then processed into a mature messenger RNA molecule (mRNA) finally translated into a protein (polypeptide).

However, having a better estimator on average does not guarantee that a particular single estimate is more accurate. Therefore, analysts and practitioners have to keep in mind that the aim of gene profiling experiments is less to test for a prespecified hypothesis on the expression of single genes than to set expression signatures associated to particular experimental conditions, cell types, tumors, or clinical outcomes. It represents therefore a shift from an hypothesis testing paradigm to an hypothesis generating paradigm (Biesecker, 2013) where a relevant task would be to prioritize genes with respect to some statistical summaries that are not directly used to assess for departure from the null hypothesis (that is no association between expression of a given gene and a covariate) in favor of the alternative hypothesis of differential expression (Noma *et al.*, 2010).

Since the emergence of high-throughput gene profiling technologies such as the serial analysis of gene expression (SAGE) (Velculescu *et al.*, 1995), cDNA microarrays (Schena *et al.*, 1995; Lennon and Lehrach, 1991) and oligonucleotide microarrays (Lockhart *et al.*, 1996), or RNA sequencing (RNA-seq) (Holt and Jones, 2008; Wang *et al.*, 2009; Wilhelm *et al.*, 2008; Mortazavi *et al.*, 2008; Cloonan *et al.*, 2008; Sultan *et al.*, 2008), the consensus that gene expression data should be analyzed at once in the framework of empirical Bayes procedures has emerged (Cui and Churchill, 2003; Sonesson and Delorenzi, 2013). This idea has been translated into numerous statistical libraries allowing to perform differential analysis of gene expression data in computational frameworks such as the *R Language and Environment for Statistical Computing* (R. Core Team, 2017) and the *Bioconductor* project (Gentleman *et al.*, 2004; Huber *et al.*, 2015).

Though differential expression analysis refers preferentially to methods and tools developed in the context of gene expression data arising from microarray or RNA-seq, the paradigm of differential expression analysis extends to the analysis of data arising from any massively parallelized technologies producing intensity measurements such as array-based proteomics methods (MacBeath and Schreiber, 2000) or overdispersed count data such as ChIP-Seq data (Robertson *et al.*, 2007) or 4C (Simonis *et al.*, 2006).

This article is organized as follows: Section “Background” gives an overview of gene profiling experiments and describes principles of hybridization- (microarrays) and sequencing-based (RNA-seq) approaches. Section “Differential Expression” introduces differential expression of gene profiling experiments in the framework of microarrays (first subsection) and in the framework of RNA-seq experiments (second subsection). The Section “Conclusion” proposes a summary and some concluding remarks.

Background

Transcriptomics refers to the mapping and quantification of the sum of all transcripts present in particular cell type or tissue (de la Grange *et al.*, 2010; Aguet *et al.*, 2017), in determined conditions (Hughes *et al.*, 2000), development stages (Graveley *et al.*, 2011), cell cycle (Cho *et al.*, 1998), treatments or states of health (Lenz *et al.*, 2008). Gene expression microarrays (Schena *et al.*, 1995; Lockhart *et al.*, 1996) and deep sequencing of RNA (RNA-seq) (Nagalakshmi *et al.*, 2008; Holt and Jones, 2008; Wang *et al.*, 2009) are the two dominant high-throughput technologies used to quantify transcripts at the genome-wide scale. These techniques allow to capture a snapshot of the transcriptional activity of a sample of cells or a tissue at reasonable costs and labor (Holt and Jones, 2008; Wang *et al.*, 2009; Aguet *et al.*, 2017; Heller, 2002; Lowe *et al.*, 2017) (Figs. 2 and 4).

Differential expression is arguably one of the most common uses of gene expression profiling experiments (Sonesson and Delorenzi, 2013; Pan, 2002). Its application includes the molecular characterization of tissues (Mortazavi *et al.*, 2008; Aguet *et al.*, 2017), the classification and prediction of diseases subtypes (Golub, 1999; Perou *et al.*, 2000), the development of prognostic

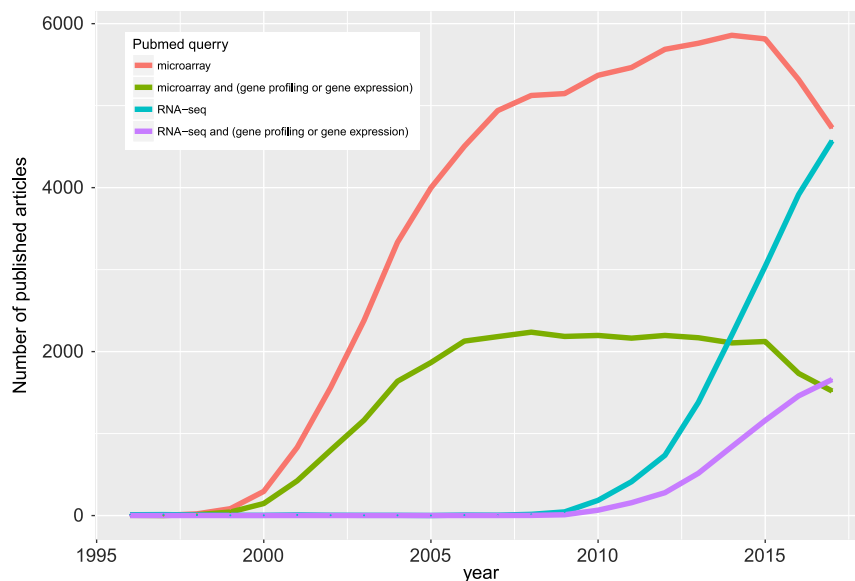


Fig. 2 Reference of gene expression profiling methods over time: Number of published papers referenced in pubmed referring to “microarray,” “microarray + gene expression or gene profiling,” “RNA-seq,” “RNA-seq + gene expression or gene profiling” over time in their title or abstract.

tools (van't Veer *et al.*, 2002; Paik *et al.*, 2004) and diagnostic tools (Spira *et al.*, 2007; Griffith *et al.*, 2006), or the functional annotation of gene through the analysis of clusters of genes associated to developmental stages (Graveley *et al.*, 2011; Bassett *et al.*, 1999) or diseases subtypes (Hughes *et al.*, 2000; Figueroa *et al.*, 2013). In drug discovery, differential expression analysis allows to validate candidate targets and get further insight on the functional connections among diseases and drug action at the transcriptome level (Hughes *et al.*, 2000; Lamb *et al.*, 2006). Gene expression profiling and differential expression are also powerful tools to characterize at the molecular level biological systems perturbed by genes knock-out or mutations (Hughes *et al.*, 2000).

Of note, although the molecular classification and characterization of oncogenic subtypes have represented an early success of gene profiling experiments (Golub, 1999; van't Veer *et al.*, 2002), their clinical utility remain unclear (Edén *et al.*, 2004; Bonastre *et al.*, 2014; Blok *et al.*, 2018). Indeed, assessment of molecular signatures for personalizing patient care is pursued preferentially using dedicated prospective clinical trials (Simon and Wang, 2006; Delord, 2015; Tajik *et al.*, 2013) in which the marginal contribution of molecular signatures is assessed over gold standards of clinical care. In practice, however, it is frequent that the information provided by molecular profiling of tumors overlaps with standard phenotypic or anatomopathological biomarkers (Edén *et al.*, 2004; Simon *et al.*, 2009), making it therefore unlikely to provide an improvement of the clinical practice. Also, when gene expression signatures are found to be clinically relevant, cost-effectiveness analyses are needed to evaluate their utility in the daily clinical practice (Bonastre *et al.*, 2014; Blok *et al.*, 2018).

Gene expression profiling technologies are based on the following principles: in the microarrays approach (Schena *et al.*, 1995; Lockhart *et al.*, 1996; Lennon and Lehrach, 1991; Heller, 2002), gene expression is estimated via hybridization of transcript cDNA to an array of predefined complementary probes spotted or synthesized at determined locations. On the other hand, in the sequencing approach, transcripts cDNA are sequenced using high-throughput sequencing technologies (Holt and Jones, 2008; Wang *et al.*, 2009) and quantification is derived from counts of sequences associated (in silico) to a given transcript (Fig. 3) (Wang *et al.*, 2009).

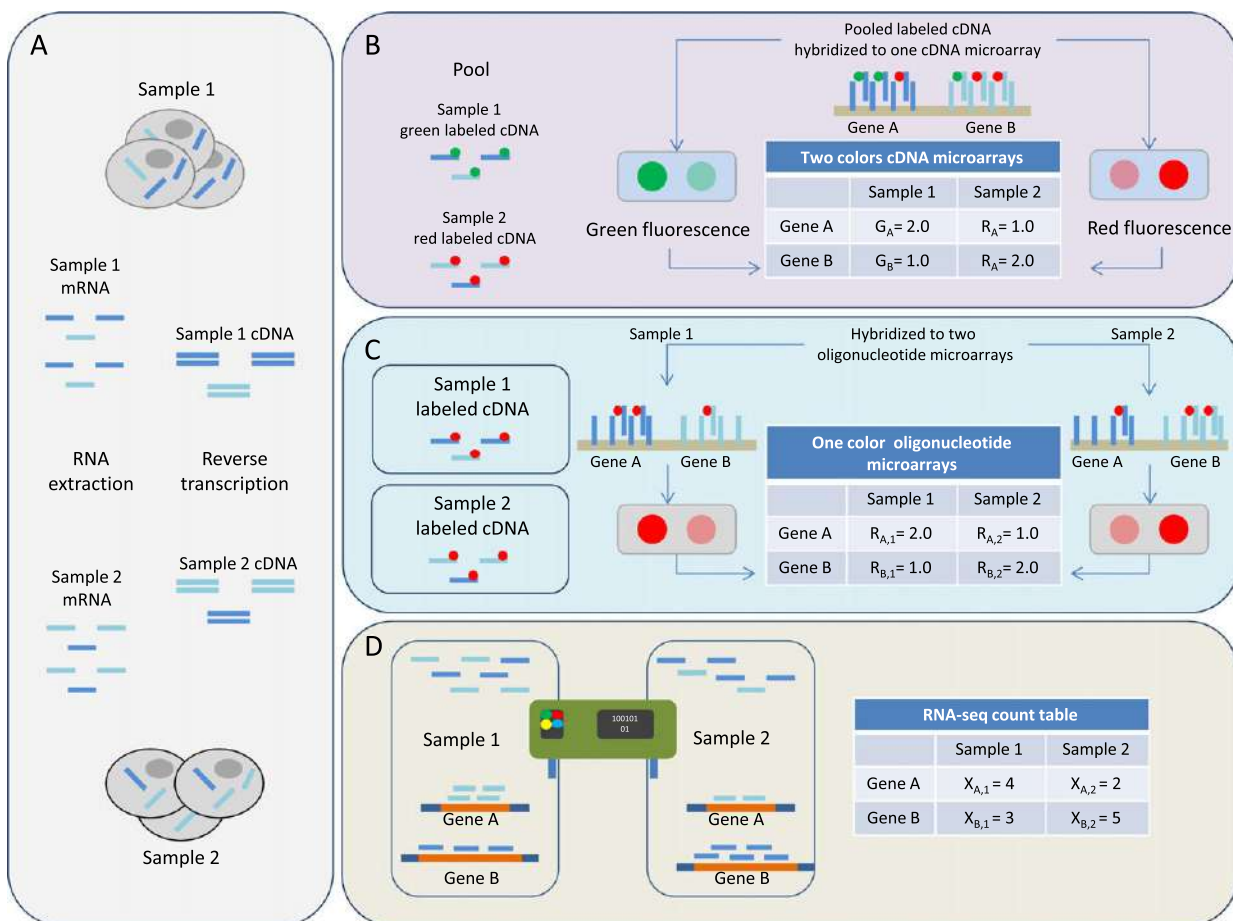


Fig. 3 Differential gene expression profiling using two-color microarrays (B), one-color microarray (C), and RNA-seq (D): In the proposed experiment, differential expression is assessed between biological samples 1 and 2. mRNA is extracted from biological samples and reverse transcribed to double strand cDNA (A). In the two-color microarray approach (B), single strand cDNA from both samples are labeled using different dyes, pooled, and incubated for hybridization into a single microarray. In the one-color microarray approach (C), single strand cDNA is labeled using the same dye and incubated for hybridization into separate oligonucleotide microarrays. Once hybridization is completed, microarrays are scanned and row expression values are derived from intensity of fluorescence for each color and each probe. In the RNA sequencing procedure (D), mRNA is reverse transcribed and amplified to cDNA followed by deep sequencing. Resulting short reads are aligned in silico and quantified.

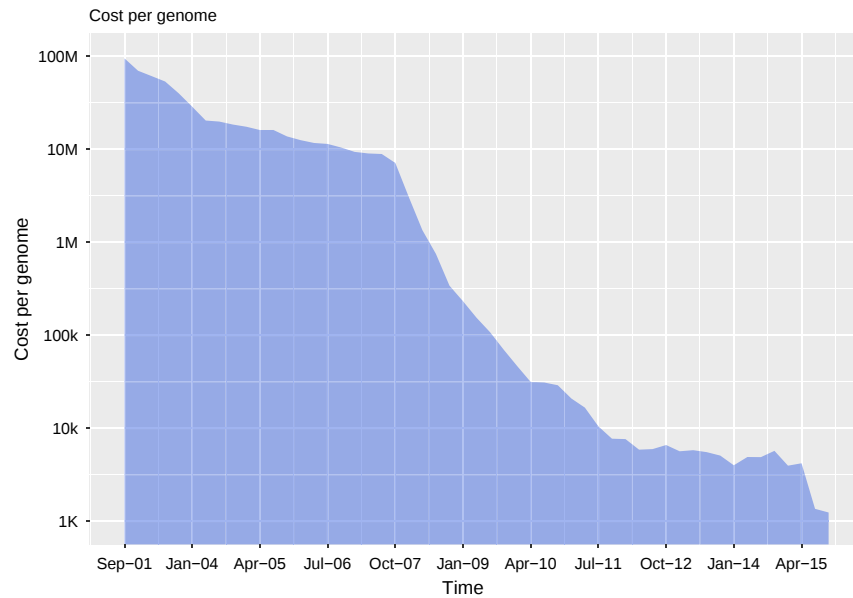


Fig. 4 Estimated cost per genome: Estimation includes global production costs such as labor, utilities, reagents and consumables, sequencing instruments, informatics activities, and indirect costs such as general laboratory supplies, etc. Data from the National Human Genome Research Institute (NHGRI) Genome Sequencing Program (GSP) (<https://www.genome.gov/27541954/dna-sequencing-costs-data/NationalHumanGenomeResearchInstitute>).

Since the publications of the first RNA sequencing studies using high-throughput sequencing platforms in 2008 (Nagalakshmi *et al.*, 2008; Wilhelm *et al.*, 2008; Mortazavi *et al.*, 2008; Cloonan *et al.*, 2008; Sultan *et al.*, 2008; Denoeud *et al.*, 2008), this approach has become widespread (Fig. 2). This evolution, which is due to the effectiveness of this technique (higher sensitivity and dynamic range and lower technical variations (Griffith *et al.*, 2010)), has been also favored by the dramatic decrease in sequencing costs (Fig. 4) (Holt and Jones, 2008; Wang *et al.*, 2009), and by the development of bioinformatic tools and dedicated statistical software (Soneson and Delorenzi, 2013; Huber *et al.*, 2015; Oshlack *et al.*, 2010). Nevertheless, the use of microarrays remains a powerful tool to assess the expression level of large sets of genes in the biological and clinical setting (Oshlack *et al.*, 2010). The use of microarrays is also favored by the availability of numerous preprocessing and normalizing procedures as well as an established theoretical and computational framework (Smyth, 2004; Cui and Churchill, 2003; R. Core Team, 2017; Huber *et al.*, 2015; Li and Wong, 2001; Lönnstedt and Speed, 2002; Efron and Tibshirani, 2002; Irizarry *et al.*, 2003; Storey *et al.*, 2007; Ritchie *et al.*, 2015).

The Microarray Approach to Gene Expression Profiling

In the late 1990s and into the 2000s, advances in genome sequencing of multiple species have provided the necessary information to produce microarrays allowing therefore to quantify the expression of large sets of genes (Brown and Botstein, 1999). Two microarray technologies have been widely used: the two-color approach (Schena *et al.*, 1995), and the one-color (or single channel) approach (Lockhart *et al.*, 1996). In two-color microarrays, two samples of transcript cDNA are labeled with different fluorophores and mixed. The resulting solution is incubated for competitive hybridization into a single microarray (Fig. 2(A) and (B)). In one-color microarrays, transcript cDNA from a single sample is labeled and incubated for hybridization into a single microarray (Fig. 2(A) and (C)).

While the two-color approach produces ratios of fluorescence intensities that reflect the relative abundance of labeled transcripts from the samples and the reference, the one-color approach yields absolute fluorescence intensities. In both cases, intensity of fluorescence is assumed to be related to the abundance of transcripts extracted from a biological sample hybridized to a microarray.

Dedicated studies have investigated the validity and equivalence of both methods (Patterson *et al.*, 2006; Larkin *et al.*, 2005; Oberthuer *et al.*, 2010) and have concluded in an overall good concordance between them. Of note, due to advantages in terms of logistics, simpler experimental designs, and data interpretation, the one-color oligonucleotide microarray has progressively supplanted the two-color spotted microarray in the 2000s (Bumgarner, 2013).

However, despite the successful use of microarrays in gene expression profiling experiments, this technology presents a number of inherent limitations such as dye-based detection issues, cross-hybridization artifacts, and a range of detection limited by a high background level and saturation of signals (Okoniewski and Miller, 2006). The hybridization-based approach relies also on prior knowledge on organisms under study including the coverage of all possible genes and associate isoforms, noncoding RNAs, and splicing events. Microarray may also produce poor results in profiling genes from highly variables genomes such as those from bacteria (Bumgarner, 2013). Advances of high-throughput DNA sequencing technologies in the mid-2000s have allowed to overcome previous constraints associated to the sequencing-based approach and provided an effective alternative to the hybridization-based approach to gene profiling experiments.

The Sequencing Approach to Gene Expression Profiling

The sequencing approach has emerged as an indirect application of DNA sequencing technologies (Velculescu *et al.*, 1995; Holt and Jones, 2008; Wang *et al.*, 2009). The SAGE (Velculescu *et al.*, 1995) proposed in the 1990s is one of the first successful attempts for quantifying expression of multiple genes in a single experiment. This approach uses Sanger sequencing and provides *digital* gene expression through the following steps:

- Mature mRNA is reverse transcribed into cDNA,
- cDNA is fragmented into 11 base-pair (bp) fragments (*tags*) using restriction enzymes,
- Tags are concatenated into long strands of cDNA and sequenced using Sanger technology,
- The resulting sequence tags are aligned along the reference genome and identify corresponding genes,
- Gene expression can be derived from counts of associated tags.

Differential gene expression can then be performed by using the SAGE approach on biological samples from various experimental conditions.

However, SAGE and other tag-based approaches that are based on traditional Sanger sequencing technologies are expensive and lack specificity as a significant proportion of tags cannot be unambiguously mapped to the reference genome. Following the advent of high-throughput sequencing technologies, such as 454 technology allowing to sequence transcripts of random length (from 25 to 300 base pair (bp)) the principle of digital gene expression has evolved and allowed the quantification of transcripts by sequencing (RNA-seq) through the following steps:

- mRNA is extracted from a biological sample, fragmented, and reverse transcribed into a library of double strand cDNA,
- Double strand cDNA is sequenced using high-throughput sequencing technologies,
- Resulting reads are aligned to the reference genome and identify corresponding genes,
- Gene expression is derived from counts of associated reads.

Different options are available at each step, for instance the cDNA library can be amplified by PCR to allow sequencing of low amounts of input RNA (Shanker *et al.*, 2015), high-throughput sequencing of cDNA can be performed using one of the different technologies mentioned above, from one end (single-end) or both ends (pair-end). However, the single-end sequencing is faster, cheaper, and sufficient for simple gene profiling experiments and differential gene expression analysis and is therefore recommended for these applications. Each biological sample produces up to 10^9 short sequences, which are associated to quality scores and listed in the fastq format in so-called fastq files. A human transcriptome can be adequately captured with approximately 30 million 100 bp reads (Hart *et al.*, 2013; Conesa *et al.*, 2016), corresponding to 1.8 gigabytes compressed fastq files.

Preprocessing of RNA-seq data

Prior to differential expression analysis using RNA-seq experiments the following three steps are needed: (1) quality control of raw reads, (2) reads alignment, and (3) read counts or transcript quantification (Fig. 5).

Quality control: Fastq files contain the list of raw reads sequence data and associated quality scores. Tools such as FastQC (Babraham Bioinformatics - FastQC A, 2017) (Illumina platform) or NGSQC (Dai *et al.*, 2010) (any platform) allow dealing with sequencing errors such as duplicated reads, GC and k-mers content, or presence of adapters. A common trait to all sequencing platforms is to produce reads whose quality decreases toward their 3' end. Software such as FASTX-Toolkit (Hannon lab, 2010) or Trimmomatic (Bolger *et al.*, 2014) can be used to discard low-quality reads and trim low-quality bases. Prior to the mapping step, the filtering out of poor quality reads allows to prevent for mismatches and downstream errors and bias.

Alignment: Different options are available for the alignment step: if no reference genome is available or is incomplete, the *de novo* transcript reconstitution is required using tools such as SOAPdenovo-Trans (Xie *et al.*, 2014), Oases (Schulz *et al.*, 2012), trans-ABYSS (Grabherr *et al.*, 2011), or Trinity (Haas *et al.*, 2013) (for a comprehensive evaluation of *de novo* transcriptome programs in differential gene expression analysis see Wang and Grubbskov (2016)). When a reference genome is available, the possible options are mapping to the genome or mapping to the transcriptome. Mapping to the genome requires the use of mapper allowing for gaps to account for (alternative) splice junctions. This approach allows the discovery of novel transcripts, gene

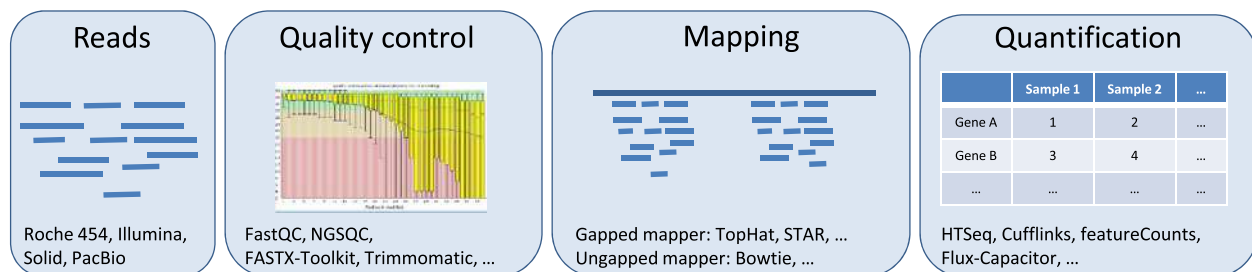


Fig. 5 RNA-seq workflow prior to differential expression analysis.

isoforms, or fusion transcripts using tools such as Tophat (Trapnell *et al.*, 2009; Kim *et al.*, 2013), MapSplice (Wang *et al.*, 2010), STAR (Dobin *et al.*, 2013), or OSA (Hu *et al.*, 2012) (for a review of genome-based sequence reads mappers, see Xie *et al.* (2014)). Finally, if a comprehensively annotated transcriptome is available, sequence reads alignment can be performed using an unspliced mapper such as Bowtie (Langmead *et al.*, 2009) allowing for gene identification and quantification simultaneously.

Transcript quantification: Once a genomic position is allocated to as many sequence reads as possible the next task is the quantification of transcripts. Depending on a chosen biologically meaningful unit, i.e., exon, transcript, or gene, this task is performed following specific sets of rules and results in a table of read counts. Quantification tools include HTSeq-count (Anders *et al.*, 2015), featureCounts (Liao *et al.*, 2014), or Cufflinks (Trapnell *et al.*, 2010). For instance, the basic strategy proposed by HTSeq is to account for a given transcript all read mapping unambiguously to the corresponding location while discarding multimapping reads. On the other hand, the Cufflinks strategy is to allocate ambiguous reads to the most probable transcribed or gene.

From fastq files containing reads for each sample to gene quantification, many options are therefore available at each step. The choice of an analysis pipeline will depend on subjective preferences of practitioners and can result in quite different expression level estimated in subsets of genes (Fonseca *et al.*, 2014). It can be noted also that other factors, such as overall sequencing coverage or gene characteristics (length or GC-content), may also affect gene profiling results (Fonseca *et al.*, 2014). When prior transcriptome knowledge is required, a paired strand-specific sequencing strategy is recommended as it provides more information to compute a *de novo* assembly of the transcriptome (Haas *et al.*, 2013). However, when a reference genome or transcriptome is available, this strategy does not improve significantly gene expression estimates (Fonseca *et al.*, 2014). Different studies have reported overall good performance of gapped mappers including Tophat, STAR, and OSA (Hu *et al.*, 2012; Fonseca *et al.*, 2014) and suggest that any of these tools could be used in the context of gene profiling experiments. On the other hand, Fonseca *et al.* (2014) have reported the critical role played by quantification methods in numerous pipelines and the good performance of the simple counting strategy of HTSeq over more sophisticated tools such as Cufflinks or Flux-Capacitor.

Differential Expression

Microarrays

Microarray experiments provide indirect estimates of gene expression through the measurement of fluorescent *spots* supposed to be related to the abundance of labeled transcripts hybridized at given positions on an array.

On a two-color microarray, where mRNA samples are hybridized against a common reference using red and green dyes respectively, relative expression values are computed as a log-ratio of fluorescence intensities:

$$M_g = \log_2(R_g) - \log_2(G_g), \quad g = 1, \dots, N \quad (1)$$

where R_g and G_g are respectively the red and green intensities for gene g , and N is the total number of genes. Most of the time, the scatter plot of $M_g = \log_2(R_g) - \log_2(G_g)$ versus $A = (\log_2(R_g) + \log_2(G_g))/2$, $g = 1, \dots, N$ (MA plots) reveals a systematic relation between log-ratios versus mean intensities. Making the assumption that most of the genes share the same expression between samples, this effect can therefore be considered as a dyes bias and corrected for using a method such as the locally weighted scatterplot smoothing (LOWESS) regression of Cleveland (1979). Of note, the assumption of equal expression of the majority of genes between biological samples can be verified by conducting a dye-swap experiment. In this experiment, the two-color procedure is duplicated from the same biological samples except for dyes, which are inverted. Dyes swap experiments allow therefore to verify that the relation between log-ratios and the mean intensity is a dye artifact that has to be corrected for.

The assumption of equal global expression level between biological samples allows also to consider that variations in global intensity between microarrays are technical artifacts that should be corrected for by scaling expression level between microarrays using a method such as quantile normalization. Once log ratios have been properly corrected for within-sample (dyes bias) and between-sample artifacts, differential expression related to treatment, conditions, or phenotype of interest can be conducted.

For one-color microarrays, a single vector of log-intensities is derived from each microarray. Summarization and normalization of intensities between arrays are usually computed following methods such as model based expression index (MBEI) implemented in dChip (Li and Wong, 2001), robust multichip average (RMA) (Irizarry *et al.*, 2003) or RMA using sequence information (GCRMA) (Wu *et al.*, 2004). These routines produce expression values suitable for further analysis. These procedures are also based on the assumption of equal global expression level between biological samples.

We introduce now differential expression analysis using data arising from a microarray experiment in a simple two-condition experimental design. In this experiment the expression of N genes was estimated from n_1 wild type samples and n_2 muted samples, using single channel microarrays. We assume that expression values were summarized and normalized between arrays and represented in a $N \times (n = n_1 + n_2)$ matrix (Y). The response vector for gene g (i.e., the estimated expression of gene g measured in the n biological samples) is $y_g = (y_{g,1}^{wt}, \dots, y_{g,n_1}^{wt}, y_{g,1}^m, \dots, y_{g,n_2}^m)$. We compute also for gene g , $\bar{y}_{wt,g}$ and $\bar{y}_{m,g}$ the sample mean of expression, and $s_{wt,g}$ and $s_{m,g}$ the sample standard deviation in wild type and mutates samples respectively. From the above we obtain the difference in mean log-expression or log fold change for gene g :

$$D_g = \bar{y}_{m,g} - \bar{y}_{wt,g}$$

and the sample variance:

$$s_g^2 = \frac{(n_1 - 1)s_{wt,g}^2 + (n_2 - 1)s_{m,g}^2}{n_1 + n_2 - 2}$$

Using these simple quantities, different procedures have been used for identifying differentially expressed genes in microarray experiments. Their basic principle is to rank genes according to some relevant statistic and to consider genes whose statistics stand above an arbitrary chosen cut-off as differentially expressed (Noma *et al.*, 2010; Cui and Churchill, 2003; Draghici, 2002). The log fold change has been used as a simple criterion to rank genes in early gene expression profiling studies (Schena *et al.*, 1995), however as its computation does not integrate a measure of the variability observed in each group, this statistic lacks confidence and therefore is not recommended in practice (Cui and Churchill, 2003).

The *signal to noise ratio* integrates a measure of the observed variance:

$$\frac{D_g}{s_{wt,g} + s_{m,g}}$$

Though this statistic allows to rank genes according to their differential expression, it is not related to a standard probability distribution and therefore does not allow to compute a type I error rate when testing the hypothesis of differential expression: $D_g \neq 0$.

Despite small sample size, the need for controlling the type I error rate has led to use of statistics following tabulated distributions such as the *t*-statistic. Assuming normality of log-expression values and equal variance between groups, under H_0 (i.e., $D_g = 0$), the t_g statistic:

$$\frac{D_g}{s_g \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

follows a Student distribution with $(n_1 + n_2 - 2)$ degrees of freedom and can be approximated by a standard normal distribution if samples size are large enough. This probabilist framework allows to compute a *p*-value:

$$2 \times P(t_{(n_1+n_2-2)} > |t_g|)$$

which can be interpreted as a measure of the evidence against H_0 given the data. If the assumption of equal variance between groups does not stand, the Welch *t*-test can be used by computing the statistic:

$$\frac{D_g}{\sqrt{\frac{s_{wt,g}^2}{n_1} + \frac{s_{m,g}^2}{n_2}}}$$

which follows a Student distribution with modified degrees of freedom if $D_g = 0$ (Welch, 1947).

The *t*-statistic presents appealing properties as it allows to rank genes in order of evidence against H_0 (Smyth, 2004; Noma *et al.*, 2010) and to choose a *p*-value cutoff associated to a desired overall type I error rate. However, the ranking obtain by the *t*-statistic would not be optimal if the sample variance is not reliable due to the few degrees of freedom available. Underestimated sample variances would lead to large values of the *t*-statistics and high rate of false positives, while overestimated sample variances would lead to low values of the *t*-statistics and therefore high rate of false negative, i.e., a loss of power.

To overcome this difficulty, the assumption of equal variance across genes has been proposed (Arfin *et al.*, 2000). This assumption leads to use the empirical mean of the sample variance estimated from all genes:

$$s^2 = \frac{1}{N} \sum_{i=1}^N s_i^2$$

This average pooled variance has therefore $N \times (n_1 + n_2 - 2)$ degrees of freedom, and the denominator of all *t* statistics becomes $s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$. The resulting test statistics can be compared to a standard normal variable to assess the probability of observed expression values under H_0 . This approach suffers however from the following two major restrictions: (1) it relies on highly unlikely assumptions and (2) it is equivalent to a log fold change ranking procedure and shares the same limitations, it is therefore not recommended to assess differential expression.

To prevent erratic values of the *t*-statistic that may be due to unreliable estimates of the sample variance Tusher *et al.* (2001) have proposed to add a small positive constant to the denominator of the *t*-statistic. In practice the constant can be derived as the 90th percentile of the standard deviation of all genes (Efron *et al.*, 2000). The statistical significance is then assessed using a permutation procedure.

Alternative approaches have been proposed in which advantage is taken from the parallel structure of the data while gene-specific variance is also considered (Smyth, 2004; Lönnstedt and Speed, 2002; Baldi and Long, 2001; Wright and Simon, 2003).

The regularized *t*-statistic proposed by Baldi and Long (2001) can be presented as follows: In a Bayesian framework, prior information on the gene-specific variance is obtained from d_0 genes sharing similar intensity (σ_0^2). The *Bayesian adjusted denominator* of the regularized *t*-statistic is computed as a weighted average of the gene-specific sample standard error (s_g) and σ_0 . The regularize *t*-statistic is:

$$\tilde{t}_g = \frac{D_g}{\tilde{s} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

with

$$\tilde{s}^2 = \frac{(n-1)s_g^2 + d_0\sigma_0^2}{(n-2) + d_0}$$

The *moderated* t -statistic of Smyth (2004) is similar to the Baldi and Long statistic but with a prior information on the gene-specific variance, which is derived from the proportion (p_0) of genes assumed to be differentially expressed between conditions where p_0 is estimated from the data.

In both cases, a background pooled variance σ_0^2 on d_0 degrees of freedom is estimated from the data and used as prior information on the gene-specific sample variance to moderate the gene-specific sample variance. The resulting modified t -statistic follows a Student distribution with $(n-2) + d_0$ degrees of freedom under H_0 . That is, an extra d_0 degrees of freedom are considered for $\tilde{t}_g$ as compared to the $n-2$ degrees of freedom used for the classical gene-wise t -statistic t_g .

These methods are part of empirical Bayes approaches as prior information is derived from the whole dataset. The added degrees of freedom used to infer from individual genes is borrowed from all genes involved in the experiment. Individual sample variances are therefore shrunk toward a common variance and result in a far more stable inference, especially when the sample size is small, and with overall gain in statistical power (Smyth, 2004). Finally, even in cases where the normality assumption does not hold, empirical Bayes test statistics lead to more accurate ranking of genes as compared to gene-wise ranking procedures.

It is worth mentioning also the B statistic of Lönnstedt and Speed (2002), which is an empirical Bayes log posterior odds-ratio of differential expression. The numerator of the B statistic is the probability that a given gene is differentially expressed, while the denominator is the probability that the gene is not differentially expressed. Both quantities are posterior probabilities combining gene-specific information, and information derived from all other genes, producing therefore more stable inference and overall more powerful procedure than the naive gene-wise approach.

Review of tests and procedures used for differential expression using microarrays can be found in Pan (2002) and Cui and Churchill (2003). In practice, the approach proposed by Smyth (2004) is the most widely used. It is implemented in the R library *limma* (Ritchie et al., 2015) which provides also facilities for reading and normalizing data from many microarray technologies including two-color microarrays and single-channel oligonucleotide platforms such as Affymetrix microarrays. It allows also to handle complex experimental designs and multiple comparison experiments in the framework of linear models.

RNA-seq

Counts resulting from RNA-seq experiments are supposed to be linked proportionally to the number of mRNA copies extracted from a biological sample and yield therefore a direct estimation of the transcriptional activity of cells from biological samples under study. However, as read counts not only depend on gene expression levels but also on length of transcripts (Oshlack and Wakefield, 2009) and overall coverage level, (Robinson and Oshlack, 2010) an appropriate normalization procedure is needed before differential expression analysis. This situation is illustrated by the RNA-seq experiment of Fig. 6(A) where gene A is assumed to be expressed twice as much as gene B in both samples. Gene A is also assumed to have the same expression between samples. Genes A and B have length $l_A=2$ and $l_B=3$ respectively, and total read counts in sample 1 and 2 are $N_1=7$ and $N_2=14$ respectively. Y_{ij} denotes read count of gene i in sample j . Computing fold-change of gene expression using raw read counts gives the following results:

- Fold change of gene expression between genes A and B in sample 1:

$$\frac{Y_{A,1}}{Y_{B,1}} = \frac{3}{4} = 0.75$$

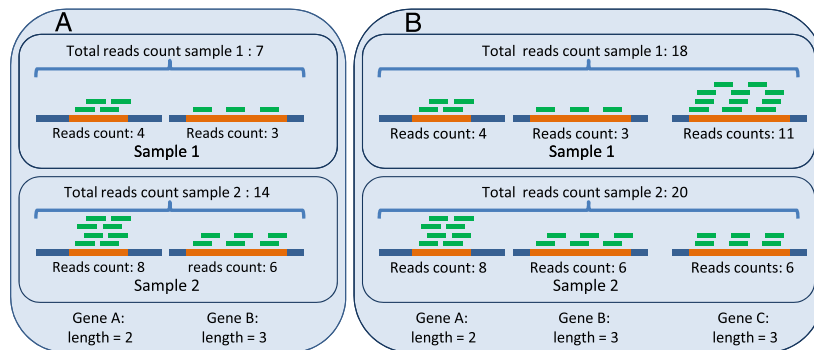


Fig. 6 The need for normalization procedure in differential expression analysis using RNA-seq data.

- Fold change of gene expression between conditions 2 and 1 for gene B:

$$\frac{Y_{B,2}}{Y_{B,1}} = \frac{6}{3} = 2$$

which does not reflect the *true* relative expression level of gene A versus gene B. This discrepancy is due to differences in gene lengths within samples and overall coverage between samples. Therefore, alike for the hybridization approach, the need arises for normalization procedures to adjust for within-sample and/or between-sample characteristics prior to differential analysis in RNA-seq experiments (Bullard *et al.*, 2010).

Within-Sample Normalization

Taking into account the gene length as a normalization factor for the within-sample comparison of gene expression gives the expected results:

$$\frac{\frac{Y_{A,1}}{l_A}}{\frac{Y_{B,1}}{l_B}} = \frac{4/2}{3/3} = 2$$

In practice, for within-sample normalization of RNA-seq experiments, different metrics of gene expression have been proposed among which are RPKM (reads per kilobase of exon per million mapped reads) (Mortazavi *et al.*, 2008), FPKM (replace reads by fragments) (Grabherr *et al.*, 2011; Trapnell *et al.*, 2010), and TPM (transcript per million) (Li *et al.*, 2010), all of which are essentially identical.

To compute these quantities, we can obtain first the count per million (CPM) for gene i (Law *et al.*, 2014):

$$\text{CPM}_i = \frac{Y_i}{N} \cdot 10^6 \quad (2)$$

where Y_i is the read count for gene i and N the total number of reads. This quantity represents simply the number of reads from gene i per million reads.

RPKM and FPKM are analogous quantities. FPKM is a more generic quantity as it explicitly accommodates data from single- or paired-end RNA-seq experiments. It is equivalent to RPKM in single-end RNA-seq experiments.

FPKM _{i} is proportional to CPM _{i} divided by the length of gene i (l_i):

$$\text{FPKM}_i = \frac{\text{CPM}_i}{l_i} \cdot 10^3 = \frac{Y_i}{l_i N} \cdot 10^9 \quad (3)$$

Finally, to obtain TPM _{i} , transcript i reads count needs to be normalized by length of gene i to get the rate of reads of type i , then to adjust for the total number of reads, divide rate i by the sum of all rates and scale to get a *per million reads* quantity:

$$\text{TPM}_i = \frac{Y_i}{l_i} \cdot \left(\frac{1}{\sum_j \frac{Y_j}{l_j}} \right) \cdot 10^6 \quad (4)$$

Remarks

- As shown in (3) the FPKM metric can be obtained by dividing the rate of transcript i by the total number of reads N and multiplying the result by 10^9 . Using this formulation in (4) the TPM metric can be expressed as (Pachter, 2011):

$$\begin{aligned} \text{TPM}_i &= \frac{Y_i}{l_i N} \cdot 10^9 \cdot \left(\frac{1}{\sum_j \frac{Y_j}{l_j N} \cdot 10^9} \right) \cdot 10^6 \\ &= \frac{\text{FPKM}_i}{\sum_j \text{FPKM}_j} \cdot 10^6 \end{aligned}$$

Basically, both FPKM and TPM give the number of reads per base of feature i , per $N/10^6$ million reads in the FPKM metric and per million reads in the TPM metric. TPM is therefore more generic than FPKM (or RPKM) and its use is therefore recommended (Wagner *et al.*, 2012).

- The number of reads for a given gene depends also on the overall coverage and therefore the number of reads in a given sample. Though TPM or FPKM take into account the total number of reads, the use of these metrics would allow for comparison between samples, only if samples show similar expression profiles such as in the experiment shown in Fig. 6(A):

$$\frac{\frac{Y_{B,2}}{l_B N_2}}{\frac{Y_{B,1}}{l_B N_1}} = \frac{\frac{6}{3 \times 14}}{\frac{3}{3 \times 7}} = 1$$

However, this assumption is highly unlikely in any RNA-seq experiments. Fig. 6(B) shows a more realistic RNA-seq experiment where expression profiles differ between samples. Total read counts in samples 1 and 2 are $N_1^* = 18$ and $N_2^* = 20$ respectively. Differential expression of gene B between samples using FPKM gives a misleading result:

$$\frac{\frac{Y_{B,2}}{L_B N_2^*}}{\frac{Y_{B,1}}{L_B N_1^*}} = \frac{\frac{6}{3 \times 20}}{\frac{3}{3 \times 18}} = 1.8$$

This simple example illustrates the need for normalizing procedures allowing to estimate the *true* differences in overall coverage between samples. It suggests also that a good normalizing procedure between samples should reduce as much as possible the influence of differentially expressed genes as reads from these genes may represent a significant proportion of the total number of reads.

Between Sample Normalization

Recall that the goal of differential expression is to estimate ratios of gene expression between two or more experimental conditions. Using the sampling framework of Robinson and Oshlack (2010), in RNA-seq experiments, the expected count of reads associated to transcript i in condition k is assumed to be related to the expected number of transcript i per cell ($\mu_{i,k}$), the length of gene i ($l_{i,k}$), the total number of reads (N_k), and the length of transcriptome under study (S_k) in condition k :

$$E[C_{i,k}] = \frac{\mu_{i,k} l_i}{S_k} N_k \quad (5)$$

with

$$S_k = \sum_i \mu_{i,k} l_i$$

Using the delta method, it can be shown from expression (5) (Robinson and Oshlack, 2010) that the expected value of the ratio of read counts between conditions 1 and 2 for gene i is:

$$E\left[\frac{Y_{i,2}}{Y_{i,1}}\right] = \frac{\mu_{i,2} S_1 N_2}{\mu_{i,1} S_2 N_1} \quad (6)$$

Expression (6) shows clearly that for gene i , the expected fold change of read counts between conditions 1 and 2 depends on the ratio of the expected number of transcripts ($\mu_{i,2}/\mu_{i,1}$), the relative overall coverage (N_2/N_1), and the relative size of transcriptomes under study (S_1/S_2), which can vary drastically between conditions (Robinson and Oshlack, 2010). An appropriate normalization procedure would therefore aim at estimating Y_1^* and Y_2^* such that $E[Y_{i,1}^*/Y_{i,2}^*] = \mu_{i,1}/\mu_{i,2}$.

Among the many approaches available for between-sample normalization (Dillies *et al.*, 2013), three can be used to deal with both bias due to transcriptome size and sequencing depth: the trimmed mean of M -values procedure (TMM) (Robinson and Oshlack, 2010) implemented in the edgeR R library (Robinson *et al.*, 2010), the relative log-expression procedure (RLE) implemented in the DESeq2 R library (Anders and Huber, 2010; Anders *et al.*, 2013; Love *et al.*, 2014), and the mean ratio normalization procedure (MRN) (Maza *et al.*, 2013).

These schemes rely on the hypothesis that technical bias affects uniformly all genes in a given sample. Therefore, non-differentially expressed genes should have the same level of normalized counts between conditions, on average. A normalization factor can therefore be derived for each sample by equalizing read counts of nondifferentially expressed genes to a reference sample or a pseudoreference. In the TMM procedure, the reference sample is the first replicate of the first condition, whereas in the RLE and MRN procedures, a pseudoreference sample is defined as the geometric mean of all samples and the simple mean of the replicate of a given condition respectively.

For each sample, computation of the relative size of transcriptome with respect to the (pseudo) reference sample is based on gene-wise count ratio. As the bias is assumed to affect identically all genes, the same normalization factor can therefore be used for all genes within a given sample.

These normalization schemes are widely used and give roughly similar results (for a detailed description and comparison of these procedures see Maza (2016)).

It remains possible nevertheless that assumptions on which these normalization schemes rely are not verified in practice. This is especially true if all or most of the genes are overexpressed in one condition only (Athanasiadou *et al.*, 2016; Zheng *et al.*, 2014; Lin *et al.*, 2012; Zuqin *et al.*, 2012). In this special case, negative controls are needed. The negative control can be a gene for which relative expression between experimental conditions is known or a spike-in control. In the resulting normalizing procedure, scaling factors will be determined such that relative expression of the control gene (or spike-in control) matches its expected value between experimental conditions.

Of note, from (6), it appears that the expected fold change of read counts for a given feature between two conditions does not depend on length of that feature. Therefore, normalization within sample is not needed when differential expression is computed between samples (Soneson and Delorenzi, 2013; Oshlack *et al.*, 2010). Moreover, in this situation, any approximation resulting from a within-sample normalization procedure is likely to introduce bias in further differential expression

analysis, therefore within-sample normalization is not recommended when between-sample differential expression analysis is pursued (Dillies *et al.*, 2013).

Differential Expression From RNA-seq

At this stage, we assume an RNA-seq experiment where read counts of N genes were derived from biological samples from two conditions, say n_1 wild type samples and n_2 muted samples, which is arguably the most encountered experimental design for differential expression experiments (Soneson and Delorenzi, 2013). We assume also that a procedure such as TMM, RLE, or MRN has been used for read count normalization. Recall that the use of these normalization procedures relies on the assumption that most genes are not differentially expressed, the remaining genes being overexpressed and underexpressed in a balanced proportion (Dillies *et al.*, 2013; Bolstad *et al.*, 2003; Calza and Pawitan, 2010). Finally and despite an appropriate experimental design (Auer and Doerge, 2010), principal component analysis may reveal presence of batch effects, which can be removed using methods such as COMBAT (Johnson *et al.*, 2007) or ARSyN (Nueda *et al.*, 2012).

Discrete distribution of read counts

An RNA-seq experiment can be considered as a sampling procedure in which the probability of drawing a read mapping gene i is equal to the proportion of cDNA fragments arising from that gene. Under this sampling design, Y_i , the number of reads mapping gene i follows a binomial distribution with parameters N , the total number of reads for that sample and p_i , the unknown proportion of cDNA fragment arising from gene i :

$$Y_i \sim B(N, p_i) \quad (7)$$

As N far exceeds Y_i , the binomial distribution can be approximated by a Poisson distribution with parameter Np_i . This approximation is supported empirically in RNA-seq experiments even with technical replicates (Marioni *et al.*, 2008). In such experiments, as technical replicates can be seen as drawn from the same large pool of cDNA fragments, the resulting sum of counts for gene i will also follow a Poisson distribution with expected mean $p_i(N_1 + \dots + N_k)$ for k technical replicates.

On the other hand, in presence of biological replicates, counts mapping to a given gene are *overdispersed* as compared to counts arising from technical replicates for the same gene. In this situation, the Poisson approximation still holds for individual samples with expected counts for gene i in sample j being:

$$E[Y_{ij}] = N_j p_{ij} \quad (8)$$

However, in a given experimental condition the sampling design becomes a two-step scheme:

1. draw the parameter of the Poisson distribution for gene i and sample j from a distribution with support on the nonnegative real numbers, and
2. draw the count of reads for gene i in sample j (Y_{ij}) from a Poisson distribution with parameter obtained in 1.

That is, the parameter of the Poisson distribution is also a random parameter and the resulting distribution is a compound or a mixture of Poisson distributions. In practice the gamma distribution satisfies properties required for the parameter of the Poisson distribution (Anders and Huber, 2010; Robinson and Smyth, 2007a). The Poisson–gamma mixture distribution is commonly referred to as the negative binomial (NB) distribution with parameters μ and ν ($NB(\mu, \nu)$), with mean and variance:

$$E(Y) = \mu, \text{ and } \text{Var}(Y) = (\mu + \nu)\mu^2 \quad (9)$$

Other possible discrete distributions for modeling counts in RNA-seq experiments are the Poisson log-normal distribution (Bulmer, 1974; Lu *et al.*, 2005) or the quasi-Poisson distribution (Lund *et al.*, 2012).

Counts from a Poisson log-normal distribution are supposed to be drawn from a Poisson distribution with parameter e^z , Z being drawn from the normal distribution with parameters μ and σ^2 . The NB and Poisson log-normal distributions look similar for low values of μ and σ but diverge for larger values of σ , the NB distribution giving more weight to $P(y_{ij}=0)$, whereas the Poisson log-normal distribution puts more weight on its right tail (Love, 2013). Like the NB distribution, the quasi-Poisson distribution has two parameters. However, the variance of the quasi-Poisson distribution is a linear function of its mean, whereas in the NB distribution the variance is a quadratic function of the mean. The choice of an appropriate distribution can therefore be based on the variance-to-mean relationship (Ver Hoef and Boveng, 2007). As RNA-seq experiments involves few replicates, the diagnostic plot may include all read counts of an experiment.

Once a distribution is chosen, the hypothesis of equal mean in read counts between the two conditions can be tested using the generalized linear model (GLM) framework. Using the NB distribution, we have:

$$Y_{ij} \sim NB(\mu_{ij}, (\nu)_i) \quad (10)$$

and

$$\log_2(\mu_{ij}) = x_j^t \beta_i \quad (11)$$

where μ_{ij} is the mean parameter of the NB distribution, that is the expected value of read count for gene i in sample j , ν_i is the overdispersion parameter for gene i , x_j is a column vector of covariates for sample j , and β_i a column vector of coefficients associated to gene i . The link function here is the $\log_2$ function for interpretation convenience. In a simple RNA-seq experiment

with two conditions, β_i can be written as:

$$\beta_i = \begin{pmatrix} \beta_{i,0} \\ \beta_{i,1} \end{pmatrix} \quad (12)$$

In this formalism, $\beta_{i,0}$ represents the $\log_2$ of the mean of normalized counts in the first condition and $\beta_{i,1}$ represents $\log_2$ fold change between conditions 1 and 2. For instance, with $\beta_a^t = (10, 1)$, and sample 1 and 2 being in the first and the second condition respectively, we have:

$$x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad x_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

then for sample 1:

$$\begin{aligned} \log_2(\mu_{a,1}) &= x_1^t \beta_a = (1 \ 0) \begin{pmatrix} 10 \\ 1 \end{pmatrix} = 10 \\ \Rightarrow \mu_{a,1} &= 2^{10} = 1024 \end{aligned}$$

and for sample 2:

$$\begin{aligned} \log_2(\mu_{a,2}) &= x_2^t \beta_a = (1 \ 1) \begin{pmatrix} 10 \\ 1 \end{pmatrix} = 11 \\ \Rightarrow \mu_{a,2} &= 2^{11} = 2048 \end{aligned}$$

Then, the hypothesis of nondifferential expression between both conditions (H_0) can be tested with the Wald test. Using the standard iteratively reweighted least squares (IRLS), the maximum likelihood estimate is derived for $\beta_{a,1}$ as well as its standard deviation ($\text{se}(\hat{\beta}_{a,1})$). Under H_0 , $\hat{\beta}_{a,1}/\text{se}(\hat{\beta}_{a,1})$ follows approximately a standard normal distribution. If $|\hat{\beta}_{a,1}|/\text{se}(\hat{\beta}_{a,1}) > U_{1-\alpha/2}$, where α is a prespecified type I error rate, gene a is called differentially expressed between conditions 1 and 2.

Borrowing strength across genes

From a practical point of view however, as RNA-seq experiments involve often small number of biological replicates, estimation of both parameters of the NB model remain challenging. Especially, a poor estimation of the overdispersion parameter can lead to unreliable estimates of the standard error of the mean parameter and therefore leads to false positive or false negative calls. To overcome this difficulty and take advantage of the massively parallel structure of the data, various methods were developed. [Robinson and Smyth \(2007\)](#) proposed to reduce ([Robinson and Smyth, 2007b](#)) or shrink ([Robinson and Smyth, 2007a](#); [Wu et al., 2013](#)) the gene-wise estimate of ν to a common value estimated from the whole dataset. [Love et al. \(2014\)](#) moderate the gene-wise dispersions by assuming a log normal prior on this parameter conditional on mean counts for a given gene. These approaches are implemented in the edgeR ([Robinson et al., 2010](#)) and the DESeq2 ([Love et al., 2014](#)) R libraries respectively.

Parametric approach on transformed counts

It has been argued that after extensive normalization and/or batch effect removal the read counts data have eventually lost their discrete nature, allowing practitioners to use parametric model and statistical workflows developed for microarray data. This approach has been used since the emergence of RNA-seq experiments after appropriate counts transformation ([Law et al., 2014](#)). However, the main challenge here is to transform RNA-seq data in order to meet hypotheses allowing the use of parametric linear regression models also used for microarray data, assumed to follow log-linear distributions.

In the linear regression model framework, the response variable k is assumed to be a linear combination of the predictors X plus an error term and the hypothesis of homoscedasticity implies that the error term is independent of X and k . However, one of the characteristics that have led to the use of Poisson distributions or compound Poisson distributions for modeling RNA-seq data is heterogeneous dispersion, i.e., heteroscedasticity. When a linear mean-variance relationship is suspected (relation (9)), the square root transformation of counts can stabilize the variance of mean counts ([t Hoen et al., 2008](#)), but more sophisticated procedures such as variance stabilizing transformations developed for microarray data ([Huber et al., 2002](#); [Durbin et al., 2002](#)), or the $\log_2$ transformation can be used if counts are overdispersed ([Zwiener et al., 2014](#)).

It has been shown nevertheless that variance stabilizing transformations such as the log-transformation would result in a reduced variance for large counts as compared to small counts ([Law et al., 2014](#)). To overcome this difficulty two approaches developed to stabilize variance of RNA-seq count data have been proposed. In the first approach denoted as *vst* (variance stabilizing transformations), the moderate gene-wise dispersions of a NB model ([Love et al., 2014](#)) are used as a variance stabilizing parameter ([Anders and Huber, 2010](#)). In the second approach, denoted as the *voom* method (variance modeling at the observation-level), a mean-variance trend is estimated nonparametrically ([Cleveland, 1979](#)) from the dataset of normalized log-counts. Fitted values associated to each individual normalized log-count are then incorporated into *precision weights*, which are used in the downstream statistical inference step of dedicated statistical pipelines such as the *limma* suite ([Smyth, 2004](#); [Ritchie](#)

et al., 2015). Of note it has been reported that this approach performs well as compared to methods based on discrete distribution of read counts (Soneson and Delorenzi, 2013; Law *et al.*, 2014).

Conclusion

Since the 1990s, progresses in genetic engineering and information technologies have allowed the sequencing of the human genome (Human Genome Sequencing Consortium International, 2004) as well as many other model organisms. At the same time, these advances have provided prior knowledge needed to perform gene expression profiling experiments using cDNA and oligonucleotide microarrays (Schena *et al.*, 1995; Lockhart *et al.*, 1996; Brown and Botstein, 1999). Since the pioneering work of Schena *et al.* (1995) tens of thousands of scientific publications have referred to gene profiling experiments using microarrays (Fig. 2). More than a decade later, and owing for the advent of high-throughput sequencing platforms, the sequencing-base approach or RNA-seq has emerged as a promising tool for gene expression profiling, as it allows both the mapping and the quantification of the sum of transcripts extracted from a biological sample. RNA-seq presents many advantages over array-based approaches, such as an higher sensitivity and dynamic range as well as a lower background noise (Wang *et al.*, 2009; Griffith *et al.*, 2010; 't Hoen *et al.*, 2008). This approach benefits also from the decrease of sequencing costs and is now the most widespread method used in gene profiling experiments (Fig. 2).

Results of gene profiling experiments using high-throughput technologies such as microarrays or RNA-seq consist of large numerical matrices in which expression level of many genes are recorded from few biological samples. Intensity records from microarrays are assumed to follow a log-normal distribution, whereas RNA-seq experiments produce counts assumed to follow Poisson related distributions. Though nonparametric approaches have also been proposed such as in Li and Tibshirani (2013) or Tarazona *et al.* (2011) most of the methods used in practice assume probabilistic distributions to expression values in order to fit parametric models in the framework of linear models or GLM theory (McCullagh and Nelder, 1989). Despite allowances that have to be made for a specific distributional hypothesis, and the choice of a model, the analysis of large scale expression data remains a high-dimensional inference problem.

In this context, empirical Bayes methods (Efron and Morris, 1973; Efron and Morris, 1977; Carlin and Louis, 1996) have provided a general theoretical framework to analyze large scale gene expression data (Newton *et al.*, 2001; Smyth, 2004; Lönnstedt and Speed, 2002; Baldi and Long, 2001; Robinson *et al.*, 2010; Anders and Huber, 2010; Love *et al.*, 2014; Robinson and Smyth, 2007a; Hardcastle and Kelly, 2010; Leng *et al.*, 2013; Van De Wiel *et al.*, 2013; Yanming *et al.*, 2011). The key point of empirical Bayes methods, and related methods, states that optimal statistical procedures to infer from a single feature became suboptimal for high-dimensional inference problems (Stein Charles, 1955). This result implies that more efficient procedures involving sharing information across features exist. In practice, estimation of a given statistic, say the standard deviation of the expression of a gene through biological samples, is shrunk toward a common value, or distribution, estimated from other genes. This principle has been applied to microarray data to moderate the denominator of test statistics and therefore prevent erratic values of the test statistic observed by chance. By preventing false positive and false negative calls, these procedures result in more stable inferences and provide therefore overall higher statistical power.

This principle has been also acknowledged for RNA-seq statistical procedures in which estimation of the dispersion parameter for each gene remains crucial while small sample size is generally involved. This situation has therefore motivated the development of empirical Bayes approaches involving information sharing across genes to obtain more reliable estimates of single dispersion parameters and improved overall inference procedure. Empirical Bayes approaches are also *de facto* convened when RNA-seq read counts are transformed using variance stabilizing procedures to enter subsequent microarray analysis pipelines.

Many tools exist to compute differential expression, none of which perform optimally in all circumstances (Soneson and Delorenzi, 2013; Conesa *et al.*, 2016). The practitioner challenge remains therefore to design a strategy that suits its particular needs, possibly in view of performances of the different tools as reported in similar experimental conditions, for instance in terms of their sensitivity/specificity and control of the FDR (Conesa *et al.*, 2016). Of note, analysis requirements such as the need to adjust results for confounding covariates and to perform multiple comparisons may also limit the list of available options to software allowing for complex statistical designs, generally in the theoretical background of GLMs and empirical Bayes procedures such as limma (Smyth, 2004) svt/voom-limma (Smyth, 2004; Law *et al.*, 2014; Love *et al.*, 2014), edgeR (Robinson *et al.*, 2010), or DESeq (Love *et al.*, 2014). The choice of an analysis strategy may also depend on individual preferences and habits, the quality of the documentation of software, and the existence of an active community of users.

See also: Analyzing Transcriptome-Phenotype Correlations. Characterization of Transcriptional Activities. Comparative Transcriptomics Analysis. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Expression Clustering. Functional Enrichment Analysis. Functional Enrichment Analysis Methods. Genome-Wide Scanning of Gene Expression. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics. Regulation of Gene Expression. Statistical and Probabilistic Approaches to Predict Protein Abundance. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation. Transcriptome Informatics. Utilising IPG-IEF to Identify Differentially-Expressed Proteins

References

- Aguet, F., Ardlie, K.G., Cummings, B.B., *et al.*, 2017. Genetic effects on gene expression across human tissues. *Nature* 550, 204–213.
- Anders, S., Huber, W., 2010. Differential expression analysis for sequence count data. *Genome Biology* 11, R106.
- Anders, S., McCarthy, D.J., Chen, Y., *et al.*, 2013. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nature Protocols* 8, 1765–1786.
- Anders, S., Pyl, P.T., Huber, W., 2015. HTSeq a Python framework to work with high-throughput sequencing data. *Bioinformatics* 31, 166–169.
- Arfin, S.M., Long, A.D., Ito, E.T., Toller, L., *et al.*, 2000. Global gene expression profiling in *Escherichia coli* K12. The effects of integration host factor. *The Journal of Biological Chemistry* 275, 29672–29684.
- Athanasiadou, N., Neymotin, B., Brandt, N., *et al.*, 2016. Growth rate-dependent global amplification of gene expression. *bioRxiv*. 044735.
- Auer, P.L., Doerge, R.W., 2010. Statistical design and analysis of RNA sequencing data. *Genetics* 185, 405–416.
- Babraham Bioinformatics – FastQC A, 2017. Quality Control tool for High Throughput Sequence Data. Available at: <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> (accessed 24.11.17).
- Baldi, P., Long, A.D., 2001. A Bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics* 17, 509–519.
- Bassett, D.E., Eisen, M.B., Boguski, M.S., 1999. Gene expression informatics – It's all in your mine. *Nature Genetics* 21, 51–55.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*. 289–300.
- Biesecker, L.G., 2013. Hypothesis-generating research and predictive medicine. *Genome Research* 23, 1051–1053.
- Blok, E.J., Bastiaannet, E., Hout, W.B., *et al.*, 2018. Systematic review of the clinical and economic value of gene expression profiles for invasive early breast cancer available in Europe. *Cancer Treatment Reviews* 62, 74–90.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics (Oxford, England)* 30, 2114–2120.
- Bolsover, S., Hyams, J., Jones, S., *et al.*, 1997. From Genes to Cells. New York: John Wiley & Sons, Inc.
- Bolstad, B.M., Irizarry, R.A., Astrand, M., Speed, T.P., 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19, 185–193.
- Bonastre, J., Marguet, S., Lueza, B., *et al.*, 2014. Cost effectiveness of molecular profiling for adjuvant decision making in patients with node-negative breast cancer. *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 32, 3513–3519.
- Brown, P.O., Botstein, D., 1999. Exploring the new world of the genome with DNA microarrays. *Nature Genetics* 21 (33), 37.
- Bullard, J.H., Purdom, E., Hansen, K.D., Dudoit, S., 2010. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* 11, 94.
- Bulmer, M.G., 1974. On fitting the Poisson lognormal distribution to species-abundance data. *Biometrics* 30, 101–110.
- Bumgarner, R., 2013. Overview of DNA microarrays: Types, applications, and their future. *Current Protocols in Molecular Biology*. (Chapter 22:Unit 22.1).
- Calza S., Pawitan Y., 2010. Normalization of gene-expression microarray data. In: *Computational Biology*. Totowa, NJ: Humana Press, pp. 37–52.
- Carlin, B.P., Louis, T.A., 1996. Bayes and Empirical Bayes Methods For Data Analysis. Chapman & Hall.
- Cho, R.J., Campbell, M.J., Winzler, E.A., *et al.*, 1998. A Genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell* 2, 65–73.
- Cleveland, W.S., 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* 74, 829–836.
- Cloonan, N., Forrest Alistair, R.R., Kolle, G., *et al.*, 2008. Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods* 5, 613–619.
- Conesa, A., Madrigal, P., Tarazona, S., *et al.*, 2016. A survey of best practices for RNA-seq data analysis. *Genome Biology* 17, 13.
- Cui, X., Churchill, G.A., 2003. Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology* 4, 210.
- Dai, M., Thompson, R.C., Maher, C., *et al.*, 2010. NGSQC: Cross-platform quality analysis pipeline for deep sequencing data. *BMC Genomics* 11, S7.
- de la Grange, P., Grataadou, L., Delord, M., *et al.*, 2010. Splicing factor and exon profiling across human tissues. *Nucleic Acids Research* 38, 2825–2838.
- Delord M., 2015. Pharmacogénétique de l'Imatinib dans la Leucémie Myéloïde Chronique et Données Censurées par Intervalles en présence de Compétition. PhD Thesis, Université Paris-Saclay.
- Denoeud, F., Aury, J.-M., Da Silva, C., *et al.*, 2008. Annotating genomes with massive-scale RNA sequencing. *Genome Biology* 9, R175.
- Dillies, M.-A., Rau, A., Aubert, J., *et al.*, 2013. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics* 14, 671–683.
- Di, Y., Schafer, D.W., Cumbie, J.S., Chang, J.H., 2011. The NBP negative binomial model for assessing differential gene expression from RNA-Seq. *Statistical Applications in Genetics and Molecular Biology* 10.
- Dobin, A., Davis, C.A., Schlesinger, F., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29, 15–21.
- Draghici, S., 2002. Statistical intelligence: Effective analysis of high-density microarray data. *Drug Discovery Today* 7, S55–S63.
- Dudoit, S., Shaffer, J.P., Boldrick, J.C., 2003. Multiple hypothesis testing in microarray experiments. *Statistical Science* 18, 71–103.
- Durbin, B.P., Hardin, J.S., Hawkins, D.M., Rocke, D.M., 2002. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics (Oxford, England)* 18 (Suppl 1), S105–S110.
- Edén, P., Ritz, C., Rose, C., *et al.*, 2004. "Good Old" clinical markers have similar power in breast cancer prognosis as microarray gene expression profilers. *European Journal of Cancer* 40, 1837–1841.
- Efron, B., Morris, C., 1973. Stein's estimation rule and its competitors – An empirical Bayes approach. *Journal of the American Statistical Association* 68, 117–130.
- Efron, B., Morris, C., 1977. Stein's paradox in statistics. *Scientific American* 236, 119–127.
- Efron, B., Tibshirani, R., 2002. Empirical bayes methods and false discovery rates for microarrays. *Genetic Epidemiology* 23, 70–86.
- Efron, B., Tibshirani, R., Goss, V., Chu, G., 2000. Microarrays and their use in a comparative experiment. Technical Report, Division of Biostatistics, Stanford University.
- Figuerola, M.E., Chen, S.-C., Andersson, A.K., *et al.*, 2013. Integrated genetic and epigenetic analysis of childhood acute lymphoblastic leukemia. *The Journal of Clinical Investigation* 123, 3099–3111.
- Fonseca, N.A., Marioni, J., Brazma, A., 2014. RNA-Seq gene profiling – A systematic empirical comparison. *PLOS ONE* 9, e107026.
- Gentleman, R.C., Carey, V.J., Bates, D.M., *et al.*, 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology* 5, R80.
- Golub, T.R., 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.
- Grabherr, M.G., Haas, B.J., Yassour, M., *et al.*, 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* 29, 644–652.
- Graveley, B.R., Brooks, A.N., Carlson, J.W., *et al.*, 2011. The developmental transcriptome of *Drosophila melanogaster*. *Nature* 471, 473–479.
- Griffith, M., Griffith, O.L., Mwenifumbo, J., *et al.*, 2010. Alternative expression analysis by RNA sequencing. *Nature Methods* 7, 843–847.
- Griffith, O.L., Melck, A., Jones, S.J., Wiseman, S.M., 2006. Meta-analysis and meta-review of thyroid cancer gene expression profiling studies identifies important diagnostic biomarkers. *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 24, 5043–5051.
- Haas, B.J., Papanicolaou, A., Yassour, M., *et al.*, 2013. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature Protocols* 8, 1494–1512.

- Hannon Lab, 2010. FASTX-Toolkit: FASTQ/A short-reads pre-processing tools (accessed 24.11.17).
- Hardcastle, T.J., Kelly, K.A., 2010. baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics* 11, 422.
- Hart, S.N., Therneau, T.M., Zhang, Y., *et al.*, 2013. Calculating sample size estimates for rna sequencing data. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology* 20, 970–978.
- Heid, C.A., Stevens, J., Livak, K.J., Williams, P.M., 1996. Real time quantitative PCR. *Genome Research* 6, 986–994.
- Heller, M.J., 2002. DNA microarray technology: Devices, systems, and applications. *Annual Review of Biomedical Engineering* 4, 129–153.
- Holt, R.A., Jones, S.J., 2008. The new paradigm of flow cell sequencing. *Genome Research* 18, 839–846.
- Huber, W., Carey, J.V., Gentleman, R., *et al.*, 2015. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* 12, 115–121.
- Huber, W., Heydebreck, A., Sultmann, H., *et al.*, 2002. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* 18, S96–S104.
- Hughes, T.R., Marton, M.J., Jones, A.R., *et al.*, 2000. Functional discovery via a compendium of expression profiles. *Cell* 102, 109–126.
- Human Genome Sequencing Consortium International, 2004. Finishing the euchromatic sequence of the human genome. *Nature* 431, 931–945.
- Hu, Z., Chen, K., Xia, Z., *et al.*, 2014. Nucleosome loss leads to global transcriptional up-regulation and genomic instability during yeast aging. *Genes & Development* 28, 396–408.
- Hu, J., Ge, H., Newman, M., Liu, K., 2012. OSA: A fast and accurate alignment tool for RNA-Seq. *Bioinformatics* 28, 1933–1934.
- Irizarry, R.A., Hobbs, B., Collin, F., *et al.*, 2003. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* 4, 249–264.
- Johnson, W.E., Li, C., Rabinovic, A., 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 8, 118–127.
- Kabanikhin, S.I., 2008. Definitions and examples of inverse and ill-posed problems. *Journal of Inverse and Ill-Posed Problems* 16, 317–357.
- Kerr, M.K., Churchill, G.A., 2001. Experimental design for gene expression microarrays. *Biostatistics* 2, 183–201.
- Kim, D., Pertea, G., Trapnell, C., *et al.*, 2013. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology* 14, R36.
- Kosorok, M.R., Ma, S., 2007. Marginal asymptotics for the “large p , small n” paradigm: With applications to microarray data. *The Annals of Statistics* 35, 1456–1486.
- Lamb, J., Crawford, E.D., Peck, D., *et al.*, 2006. The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313, 1929–1935.
- Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L., 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10, R25.
- Larkin, J.E., Frank, B.C., Gavras, H., *et al.*, 2005. Independence and reproducibility across microarray platforms. *Nature Methods* 2, 337–344.
- Law, C.W., Chen, Y., Shi, W., Smyth, G.K., 2014. Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology* 15, R29.
- Leng, N., Dawson, J.A., Thomson, J.A., *et al.*, 2013. EBSeq: An empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics* 29, 1035–1043.
- Lennon, G.G., Lehrach, H., 1991. Hybridization analyses of arrayed cDNA libraries. *Trends in Genetics* 7, 314–317.
- Lenz, G., Wright, G., Dave, S.S., *et al.*, 2008. Stromal gene signatures in large-B-cell lymphomas. *The New England Journal of Medicine* 35922359, 2313–2323.
- Liao, Y., Smyth, G.K., Shi, W., 2014. FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* 30, 923–930.
- Lin, C.Y., Lovén, J., Rahl, P.B., *et al.*, 2012. Transcriptional amplification in tumor cells with elevated c-Myc. *Cell* 151, 56–67.
- Li, B., Ruotti, V., Stewart, R.M., *et al.*, 2010. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics* 26, 493–500.
- Li, J., Tibshirani, R., 2013. Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-Seq data. *Statistical Methods in Medical Research* 22, 519–536.
- Li, C., Wong, W.H., 2001. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences* 98, 31–36.
- Lockhart, D.J., Dong, H., Byrne, M.C., *et al.*, 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology* 14, 1675–1680.
- Lönnstedt, I., Speed, T., 2002. Replicated microarray data. *Statistica Sinica* 12, 31–46.
- Love, M.I., 2013. Statistical analysis of high-throughput sequencing count data. PhD Thesis, Freien Universität Berlin.
- Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15, 550.
- Low, R., Shirley, N., Bleackley, M., *et al.*, 2017. Transcriptomics technologies. *PLOS Computational Biology* 13, e1005457.
- Lund, S.P., Nettleton, D., McCarthy, D.J., Smyth, G.K., 2012. Detecting differential expression in RNA-sequence data using quasi-likelihood with shrunken dispersion estimates. *Statistical Applications in Genetics and Molecular Biology* 11.
- Lu, J., Tomfroh, J.K., Kepler, T.B., 2005. Identifying differential expression in multiple SAGE libraries: An overdispersed log-linear model approach. *BMC Bioinformatics* 6, 165.
- MacBeath, G., Schreiber, S.L., 2000. Printing proteins as microarrays for high-throughput function determination. *Science (New York, N.Y.)* 289, 1760–1763.
- Marioni, J.C., Mason, C.E., Mane, S.M., *et al.*, 2008. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research* 18, 1509–1517.
- Maza, E., 2016. In Papyro Comparison of TMM (edgeR), RLE (DESeq2), and MRN normalization methods for a simple two-conditions-without-replicates RNA-Seq experimental design. *Frontiers in Genetics* 7, 164.
- Maza, E., Frasse, P., Senin, P., *et al.*, 2013. Comparison of normalization methods for differential gene expression analysis in RNA-Seq experiments: A matter of relative size of studied transcriptomes. *Communicative & Integrative Biology* 6, e25849.
- McCullagh, P., Nelder, J.A., 1989. Generalized linear models, Monograph on Statistics and Applied Probability, no. 37.
- Morris, C.N., 1983. Parametric empirical Bayes inference: Theory and applications. *Journal of the American Statistical Association* 78, 47–55.
- Mortazavi, A., Williams, B.A., McCue, K., *et al.*, 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*.
- Nagalakshmi, U., Wang, Z., Waern, K., *et al.*, 2008. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320, 1344–1349.
- Newton, M.A., Kendzierski, C.M., Richmond, C.S., *et al.*, 2001. On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology* 8, 37–52.
- Nie, Z., Hu, G., Wei, G., *et al.*, 2012. c-Myc is a universal amplifier of expressed genes in lymphocytes and embryonic stem cells. *Cell* 151, 68–79.
- Noma, H., Matsui, S., Omori, T., Sato, T., 2010. Bayesian ranking and selection methods using hierarchical mixture models in microarray studies. *Biostatistics* 11, 281–289.
- Nueda, M.J., Ferrer, A., Conesa, A., 2012. ARSYN: A method for the identification and removal of systematic noise in multifactorial time course microarray experiments. *Xi Biostatistics* 13, 553–566.
- Oberthuer, A., Juraeva, D., Li, L., *et al.*, 2010. Comparison of performance of one-color and two-color gene-expression analyses in predicting clinical endpoints of neuroblastoma patients. *The Pharmacogenomics Journal* 10, 258–266.
- Okoniewski, M.J., Miller, C.J., 2006. Hybridization interactions between probesets in short oligo microarrays lead to spurious correlations. *BMC Bioinformatics* 7, 276.
- Oshlack, A., Robinson, M.D., Young, M.D., 2010. From RNA-seq reads to differential expression results. *Genome Biology* 11, 220.
- Oshlack, A., Wakefield, M.J., 2009. Transcript length bias in RNA-seq data confounds systems biology. *Biology Direct* 4, 14.
- Pachter, L., 2011. Models for transcript quantification from RNA-Seq. *arXiv:1104.3889*.
- Paik, S., Shak, S., Kim, C., *et al.*, 2004. A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer. *The New England Journal of Medicine* 35127351, 2817–2826.
- Pan, W., 2002. A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics* 18, 546–554.
- Patterson, T.A., Lobenhofer, E.K., Fulmer-Smentek, S.B., *et al.*, 2006. Performance comparison of one-color and two-color platforms within the Microarray Quality Control (MAQC) project. *Nature Biotechnology* 24, 1140–1150.
- Perou, C.M., Sørlie, T., Eisen, M.B., *et al.*, 2000. Molecular portraits of human breast tumours. *Nature* 406, 747–752.

- Philippe, B., 2008. Regression pls et donnees censurees. PhD Thesis, Conservatoire National des Arts et Métiers.
- R. Core Team, 2017. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.
- Ritchie, M.E., Phipson, B., Wu, D., *et al.*, 2015. Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* 43, e47.
- Robertson, G., Hirst, M., Bainbridge, M., *et al.*, 2007. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods* 4, 651–657.
- Robinson, M.D., McCarthy, D.J., Smyth, G.K., 2010. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)* 26, 139–140.
- Robinson, M.D., Oshlack, A., 2010. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11, R25.
- Robinson, M.D., Smyth, G.K., 2007a. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics* 23, 2881–2887.
- Robinson, M.D., Smyth, G.K., 2007b. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics* 9, 321–332.
- Schena, M., Shalon, D., Davis, R.W., Brown, P.O., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science (New York, N.Y.)* 270, 467–470.
- Schulz, M.H., Zerbino, D.R., Vingron, M., Birney, E., 2012. Oases: Robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics* 28, 1086–1092.
- Shanker, S., Paulson, A., Edenberg, H.J., *et al.*, 2015. Evaluation of commercially available RNA amplification kits for RNA sequencing using very low input amounts of total RNA. *Journal of Biomolecular Techniques: JBT* 26, 4–18.
- Simon, R.M., Paik, S., Hayes, D.F., 2009. Use of archived specimens in evaluation of prognostic and predictive biomarkers. *Journal of the National Cancer Institute* 101, 1446–1452.
- Simonis, M., Klous, P., Splinter, E., *et al.*, 2006. Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture-on-chip (4C). *Nature Genetics* 38, 1348–1354.
- Simon, R., Wang, S.-J., 2006. Use of genomic signatures in therapeutics development in oncology and other diseases. *The Pharmacogenomics Journal* 6, 166–173.
- Smyth, G.K., 2004. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* 3, 1–25.
- Soneson, C., Delorenzi, M., 2013. A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics* 14, 91.
- Spira, A., Beane, J.E., Shah, V., *et al.*, 2007. Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. *Nature Medicine* 13, 361–366.
- Stein Charles, 1955. Inadmissibility of usual estimator for the mean of multivariate normal distributions. In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, 4, 197–206.
- Storey, J.D., Dai, J.Y., Leek, J.T., 2007. The optimal discovery procedure for large-scale significance testing, with applications to comparative microarray experiments. *Biostatistics* 8, 414–432.
- Sultan, M., Schulz, M.H., Richard, H., *et al.*, 2008. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science (New York, N.Y.)* 321, 956–960.
- Tajik, P., Zwinderman, A.H., Mol, B.W., Bossuyt, P.M., 2013. Trial designs for personalizing cancer care: A systematic review and classification. *Clinical Cancer Research* 19, 4578–4588.
- Tarazona, S., Garca-Alcalde, F., Dopazo, J., *et al.*, 2011. Differential expression in RNA-seq: A matter of depth. *Genome Research* 21, 2213–2223.
- 't Hoen, P.A.C., Ariyurek, Y., Thygesen, H.H., *et al.*, 2008. Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms. *Nucleic Acids Research* 36, e141.
- Trapnell, C., Pachter, L., Salzberg, S.L., 2009. TopHat: Discovering splice junctions with RNA-Seq. *Bioinformatics* 25, 1105–1111.
- Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* 28, 511–515.
- Tusher, V.G., Tibshirani, R., Chu, G., 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America* 98, 5116–5121.
- van't Veer, L.J., Dai, H., Viver, M.J., *et al.*, 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536.
- Velculescu, V.E., Zhang, L., Vogelstein, B., Kinzler, K.W., 1995. Serial analysis of gene expression. *Science (New York, N.Y.)* 270, 484–487.
- Ver Hoef, J.M., Boveng, P.L., 2007. Quasi-poisson vs. negative binomial regression: How should we model overdispersed count data? *Ecology* 88, 2766–2772.
- Wagner, G.P., Kin, K., Lynch, V.J., 2012. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory in Biosciences* 131, 281–285.
- Wang, Z., Gerstein, M., Snyder, M., 2009. RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews Genetics* 10, 57–63.
- Wang, S., Gribkov, M., 2016. Comprehensive evaluation of de novo transcriptome assembly programs and their effects on differential gene expression analysis. *Bioinformatics* 33, btw625.
- Wang, K., Singh, D., Zeng, Z., *et al.*, 2010. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research* 38, e178.
- Welch, B.L., 1947. The generalization of student's problem when several different population variances are involved. *Biometrika* 34, 28–35.
- West, M., 2003. Bayesian factor regression models in the "Large p, Small n" paradigm. *Bayesian Statistics* 7, 723–732.
- Van De Wiel, M.A., Leday, G.G.R., Pardo, L., *et al.*, 2013. Bayesian analysis of RNA sequencing data by estimating multiple shrinkage priors. *Biostatistics* 14, 113–128.
- Wilhelm, B.T., Marguerat, S., Watt, S., *et al.*, 2008. Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature* 453, 1239–1243.
- Wright, G.W., Simon, R.M., 2003. A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics* 19, 2448–2455.
- Wu, Z., Irizarry, R.A., Gentleman, R., *et al.*, 2004. A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association* 99, 909–917.
- Wu, H., Wang, C., Wu, Z., 2013. A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data. *Biostatistics* 14, 232–243.
- Xie, Y., Wu, G., Tang, J., *et al.*, 2014. SOAPdenovo-Trans: De novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics* 30, 1660–1666.
- Zwiener, I., Frisch, B., Binder, H., 2014. Transforming RNA-Seq data to improve the performance of prognostic gene signatures. *PLOS ONE* 9, e85150.

Expression Clustering

Xiaoxin Ye and Joshua WK Ho, Victor Chang Cardiac Research Institute, Darlinghurst, NSW, Australia and University of New South Wales, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Quantification of gene expression is very important in biological and biomedical research. A major breakthrough in gene expression analysis is the development of DNA microarrays in the 1990s (DeRisi *et al.*, 1997; Schena *et al.*, 1995). Prior to the development of DNA microarrays, expression of genes is mainly measured by relatively low-throughput experimental techniques such as quantitative polymerase chain reaction (qPCR), Southern and Northern blotting. All these techniques measure the absolute or relative amount of a small number of target complementary DNA (cDNA) or RNA molecules in a sample. One key innovation of microarray technology is its ability to simultaneously measure the expression of thousands of genes in a single assay. It enables researchers to ask new questions about the entire gene expression system as a whole, instead of focusing on individual molecules one at a time.

In late 2000s, the advent of next-generation sequencing (NGS) brought about the next major breakthrough in gene expression analysis, namely RNA sequencing (RNA-seq) (Cloonan *et al.*, 2008; Mortazavi *et al.*, 2008; Nagalakshmi *et al.*, 2008; Wilhelm *et al.*, 2008). The idea is very simple. Expressed RNA transcripts are reversed transcribed into cDNA, which can be amplified, fragmented, and sequenced using NGS technologies. Expression of a gene can be quantified by the number of sequence reads aligned to the reference transcriptome. The raw readout of RNA-seq is therefore read counts of each gene, which can then be further normalised by the total number of reads in a library and sometimes by gene length. RNA-seq is found to have better detection sensitivity and specificity than microarrays (Bottomly *et al.*, 2011; Fu *et al.*, 2009; Marioni *et al.*, 2008). It is also much more flexible since we are no longer restricted by the collection of probes that are available on a microarray. RNA-seq is currently the preferred method for gene expression profiling (Wang *et al.*, 2009).

Regardless of the technology used to obtain the gene expression profiles, the aim of gene expression profiling is to identify similarities and differences in gene expression patterns among a group of samples from different biological conditions (i.e., phenotypic, genotypic, temporal, or environmental differences). Each sample can be derived from a cell line or tissue from an individual or an experimental animal model. A microarray data set is commonly represented by an $n \times m$ matrix,

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \cdots & \cdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}$$

where n is the number of genes and m is the number of samples. Each element $x_{i,j}$ in the matrix represents the expression value of gene i and sample j . For example, in a study that profiles the genome-wide mRNA expression levels of 50 heart disease patients and 50 matched healthy individuals, n is approximately 20,000 (the number of protein coding genes in human), and m is 100. Gene expression analysis refers to the manipulation of this matrix.

The use of genome-wide gene expression profiling in biological or biomedical research has grown rapidly in the last decade. One of the most widely used public repositories of gene expression data is the National Centre for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) (Barrett *et al.*, 2005, 2013). To date (October 2017), GEO contains gene expression profiles from close to 2 million samples. These gene expression profiles cover many species, cell types, and biological conditions. Proper analysis of these data is crucial in bioinformatics.

One of the most basic analyses that one can perform on a gene expression matrix is clustering (Eisen *et al.*, 1998). This includes the clustering of genes and clustering of samples. Roughly speaking, the goal of clustering is to group objects into clusters such that similar objects are grouped in a cluster, whereas dissimilar objects are grouped in different clusters. In the context of gene expression analysis, clustering can be performed to cluster genes (where each gene is represented by its expression across a number of samples) or samples (where each sample is represented by the expression of the genes).

A small data set is shown in Fig. 1 to illustrate the power of gene expression clustering. This toy data set consists of 10 genes and 8 samples. A popular way to visualise a gene expression matrix is to use a heat map. In the heat map in Fig. 1, red indicates high gene expression and blue indicates low expression. Without clustering, it is quite difficult to discern any patterns in this gene expression matrix. After clustering is performed on the genes and samples (hierarchical clustering is used in this case), we could reorder the genes and samples such that genes/samples that are within the same cluster are closer together. Clear patterns emerge from the clustering analysis. We can readily identify three groups of samples: the brown cluster (sample 2, 6, and 7), blue cluster (sample 1 and 3), and the red cluster (sample 4, 5, and 8). We could also identify three distinct groups of genes: the green cluster (gene 6, 7, and 10), the yellow cluster (gene 4 and 9), and the purple cluster (gene 2, 3, and 8). There are also 2 genes that have expression patterns that are quite distinct from other gene clusters. They can be viewed as outliers.

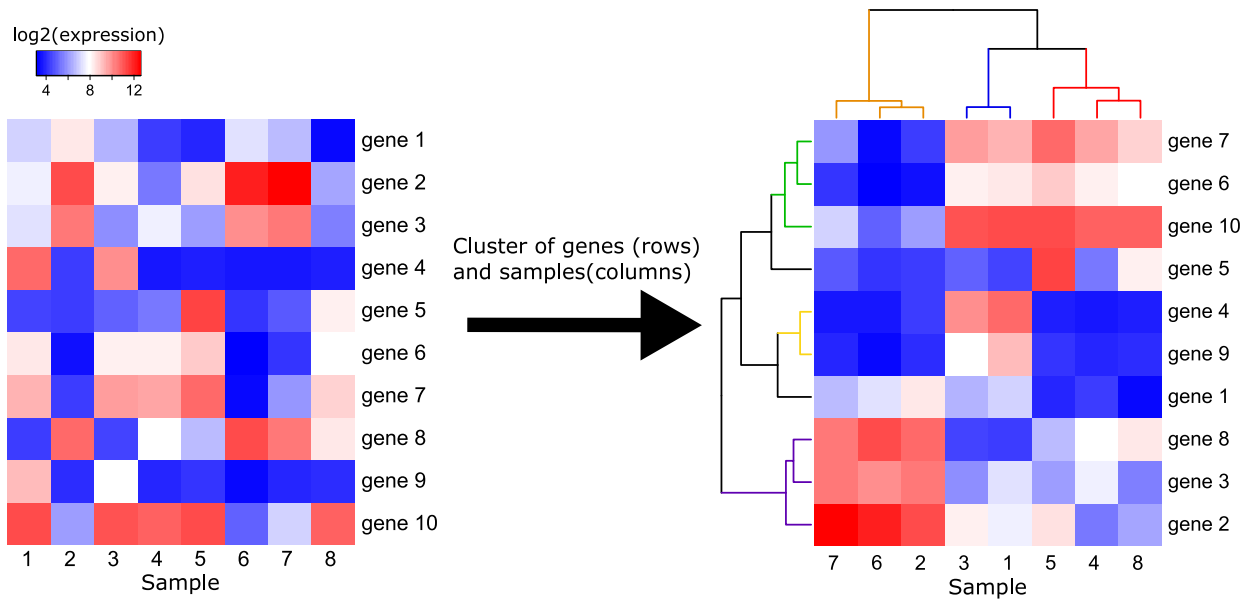


Fig. 1 A toy example illustrating the power of gene expression clustering in discovering groupings of samples and genes. A gene expression matrix consists of 10 genes (rows) and 8 samples (columns) are subjected to clustering analysis. After hierarchical clustering, it is much easier to identify three distinct groups of samples, and three sets of co-expressed genes, along with two 'outlier' genes that have very distinct expression patterns compared to other gene clusters.

Michael Eisen and colleagues were one of the first groups that applied clustering to analyse and visualise gene expression data (Eisen *et al.*, 1998). Through clustering of genes, they were able to show that genes of similar functions tend to group together in a cluster. This is a major observation because it suggests that genes that have similar expression patterns across multiple samples may be co-regulated at the cellular level. This motivated the use of clustering to identify co-expression modules as a way to discover cellular gene regulation programs (Basso *et al.*, 2005; Friedman *et al.*, 2000; Stuart *et al.*, 2003). Furthermore, clustering of samples can lead to discovery of patient subtypes and rare cell types (Golub *et al.*, 1999; Ramaswamy *et al.*, 2001; Su *et al.*, 2001). Nowadays, clustering is one of the most routinely performed analyses on gene expression data.

Clustering Methods

Clustering is an unsupervised method to discover patterns in the data. It is unsupervised because it depends on the data alone (i.e., gene expression values only), without the need to include additional information about how the data should be grouped. Clustering can be framed as an optimisation problem in which the objective is to maximize the similarity between every item within each subset (i.e., cluster), and maximize the dissimilarity between every subset (i.e., cluster). Solving this optimisation problem exactly is computationally difficult because in general one needs to enumerate all k^n possible clustering configurations to identify the optimal clustering for n points and k clusters. Instead, various algorithms have been developed to find a good local optimal solution (Jain *et al.*, 1999). In this section, we will review three of the most popular clustering methods used in gene expression analysis: k -means clustering, hierarchical clustering, and DBSCAN (Fig. 2). All these methods are easily accessible via the scikit-learn package in python (Pedregosa *et al.*, 2011) and the cluster (Maechler *et al.*, 2017) and dbscan (Hahsler *et al.*, 2017) packages in R.

K-Means Clustering

K -means clustering is a partitional clustering method, in which a set of items is partitioned into k disjoint subsets such that each item belongs to exactly one subset. The basic idea of k -means is that each cluster is assigned a cluster centre (i.e., cluster mean), and the optimisation objective is to minimise the distance between each data point and its closest cluster centre. The classical k -means clustering algorithm starts with selecting k random data points as the initial cluster centres, and iteratively perform the following two steps:

- 1) Assigns each point to its closest cluster centre
- 2) Based on the points in each cluster after step 1, calculate the cluster mean and make it the new cluster centre

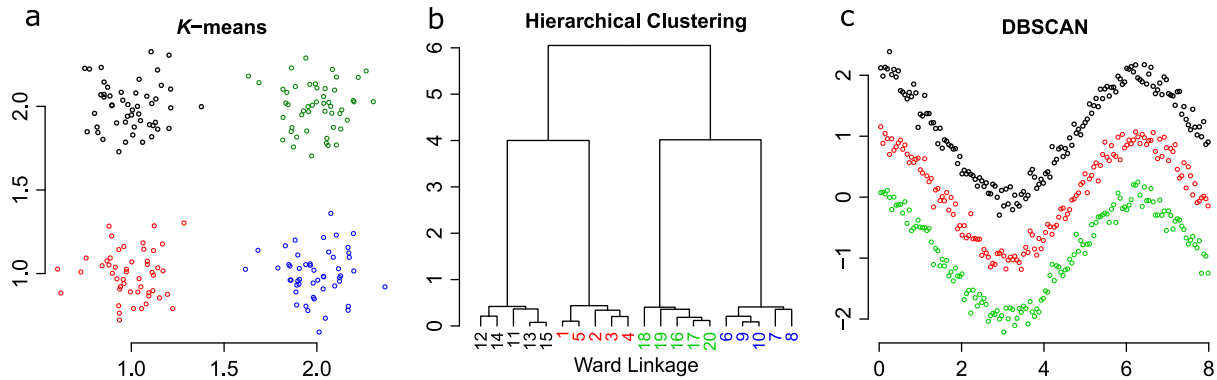


Fig. 2 Illustration of three popular clustering methods: (a) *k*-means clustering, (b) hierarchical clustering, and (c) DBSCAN.

These two steps are repeated until the maximum number of iteration is reached or the cluster assignment no longer change (i.e., no points are re-assigned in step 1). The *k*-means algorithm is fast and simple. The algorithm works very well for ball-shape and cubic-shape clusters, especially when *k* (the number of cluster) is somehow known. *K*-means is memory efficient, and it is easy to parallelise (Zhang *et al.*, 2011). Nonetheless, *k*-means is sensitive to the initialisation of cluster centres. To improve the quality of clustering, *k*-means++ has been developed (Arthur and Vassilvitskii, 2007). The idea is to perform a simple randomised seeding step to obtain good initial cluster centres. This simple step has been shown to significantly improve both clustering accuracy and speed (by reducing the number of iterations required).

Hierarchical Clustering

The key difference between hierarchical clustering and *k*-means clustering is the structure of clusters. While *k*-means clustering groups all points into distinct non-overlapping clusters, hierarchical clustering groups all the points in a tree-like hierarchy, also known as a dendrogram. The root of the tree represent the whole data set and every branch node represents a subset of the data. Hierarchical clustering methods are categorised into agglomerative (bottom-up) and divisive (top-down) algorithms. In bioinformatics, the agglomerative algorithm is more popular, and therefore will be the focus of this section.

The basic idea of agglomerative hierarchical clustering is recursive agglomeration of two nearest clusters or data points. Agglomeration criteria include single-linkage, complete-linkage, averaged-linkage clustering, and the Ward's method. Different linkage criteria can lead to different results, with the Ward's method generally considered to produce consistently good results for expression data (Meunier *et al.*, 2007). The Ward method (Ward, 1963) merges two clusters with minimum weighted squared distance between the new cluster centres.

In the initial step of the agglomerative hierarchical clustering algorithm, each of the *n* points is labelled as a distinct cluster. Then an *n*×*n* dissimilarity matrix is calculated to reflect the dissimilarity (or distance) between every pair of clusters. Clusters are then recursively merged to form larger clusters based on the following steps:

- 1) Merge two clusters with minimum cluster distance (based on the an agglomeration criterion)
- 2) Update the dissimilarity matrix

The two steps are repeated until only one cluster is left.

The main benefit of hierarchical clustering is that there is no need to determine the number of clusters prior to the analysis. This allows information about the relationships between different clusters (or data points) to be preserved. If it is desirable to obtain discrete clusters, one can always 'cut' the dendrogram at a certain level to obtain them. In practice, one drawback of hierarchical clustering is that the pairwise dissimilarity matrix can become prohibitively large when the number of data point is large, which makes clustering impractical for data sets with more than several thousand points using standard computers.

DBSCAN

Density based clustering (Kriegel *et al.*, 2011) is a category of algorithm that could detect clusters of any shape by dividing a data set into a set of connected 'high density' components. It includes DBSCAN (Ester *et al.*, 1996), OPTICS (Ankerst *et al.*, 1999) and DENCLUE (Hinneburg and Keim, 1998). The key idea is to define a cluster as a region in the data space that has high density, separated from other clusters by low density regions. The definition is intuitive, does not require the number of clusters to be predetermined, and has the potential to recover complex non-spherical clusters. DBSCAN is a frequently used density-based clustering method in bioinformatics. The basic idea of DBSCAN is outlined as follows (Schubert *et al.*, 2017):

- 1) Go through every point in the data set to check if each point *p* has at least *m* neighbours within a distance of ϵ . If *p* satisfies this criterion, it is said to be a core point. Otherwise, it is a non-core point.

- 2) Join neighbouring core points into clusters
- 3) Go through each non-core point, and try to joint it to a reachable cluster. If this point can be joint, it becomes part of that cluster. Otherwise this point is said to be an outlier (or noise).

With suitable user-specified parameters (m and ε), DBSCAN can work very well in detecting clusters with complex shape, and in the presence of outliers. DBSCAN is fast and has low memory requirement. However, its performance is sensitive to the choice of parameters and data dimensionality. A large ε may merge too many clusters into one, while a small ε may incorrectly create too many outliers. In high dimension, finding suitable parameters becomes very difficult because of the increased sparsity. Various improvements have been made to the original DBSCAN algorithm (Rehman *et al.*, 2014). For example, FDBSCAN dramatically reduces the query time of neighbours and the effect of user-specified parameters (Liu, 2006).

Evaluation and Interpretation of Clustering Results

Of the three clustering algorithms discussed in the previous sections, k -means algorithm generally scales very well with increasing number of data points because its algorithmic complexity is linear with respect to the number of data points. K -means is especially helpful if the user has prior knowledge about the expected number of clusters. Nonetheless, k -means clustering is ineffective in discovery clusters that have a non-spherical shape. Also, the results are highly sensitive to the presence of outliers (Gan and Ng, 2017). Therefore, it is important to visualise and carefully inspect the quality of the resulting clusters to ensure they are not subjected to significant biases.

Hierarchical clustering provides more information than flat partitional clustering. It is particularly useful in the case where we have no idea what the correct number of cluster should be. Nonetheless, the algorithm requires quadratic runtime and space. This means in practice it is often not possible to cluster more than several thousand points on a standard desktop machine. If the number of points is small, hierarchical clustering can often provide a highly informative view of the clustering structure of the data.

Density-based clustering algorithms such as DBSCAN is much more flexible in terms of detecting complex cluster structures, and they can deal with larger data sets as these algorithms generally scale quasi-linearly with the number of points. One major drawback with DBSCAN is that the determination of local density is very sensitive to increased dimensionality. Also, choosing the right parameters can often be quite difficult.

It should be noted that most clustering methods find clusters even when there is no natural clustering structure in the data. Therefore it is important to evaluate the quality of the clustering results and interpret its significance cautiously (Altman and Krzywinski, 2017; Ronan *et al.*, 2016). Data preprocessing and dimensionality reduction can dramatically affect the quality of clustering results (see discussion in the following section).

In terms of evaluation of the quality of the clustering results, if there is external information (such as sample labels or even partial labels), such information can be used to determine whether a clustering is reasonable or not. For example, if all the labels are given, one can calculate an adjusted rand index to quantify the accuracy of the clustering. Often only partial information is given, such as which samples are replicates. In this case, one might consider the quality of the overall clustering based on whether replicates are correctly grouped together.

Practical Considerations

In gene expression analysis, one of the most important decisions is to identify the appropriate feature set. A genome-wide gene expression data set typically contains the expression levels of tens of thousands of genes. Prior to clustering, it is often useful to perform dimensionality reduction to reduce the number of features used in the clustering analysis. This may be necessary because of both biological reasons and technical reasons.

In terms of biological reasons, sometimes scientists want to focus on a specific set of key genes, such as transcription factors, receptors, or signalling molecules. In this case, it is useful to only include these key genes in clustering analysis to discover clusters based on these key genes.

If there is no prior knowledge about which genes should be included in the analysis, there are still a number of technical reasons why dimensionality reduction is necessary in practice. In microarray data analysis, often a gene is represented by several probes. One might wish to first combine the expression levels of multiple probes to generate a single expression value for each gene. Several methods are possible, such as only including the probe with the highest median expression level, or simply taking the average of the probe expression values.

Furthermore, it is often observed that many genes have no detectable expression across all the samples profiled in the entire data set. In this case, including those non-expressed genes in the clustering might artificially inflate the dimensionality of the data without providing any real signal for clustering. It is often advisable to remove these 'undetectable' genes, using criteria such as removing all genes which have detectable expression in less than 2 samples.

After all these filtering steps, a genome-wide gene expression data set may still have a large number of genes (i.e., in the order of thousands). Clustering high dimensional data can often be problematic, because as the number of dimensions increases, the space become more sparse, and the difference between within- and between-cluster distances become smaller – this phenomena is known as the curse of dimensionality. Many methods have been designed to deal with clustering of high dimensional data (Assent, 2012), but as a general rule, doing sensible feature selection and dimensionality reduction is often critical in gene expression analysis.

A simple method for feature selection involves only selecting features that have the high variance (or some measure of variability) across all the samples in the data set. For example, one might consider selecting only those genes in which the maximum and minimum expression levels are at least 2 fold different. Besides feature selection, another popular dimensionality reduction method is Principal Component Analysis (PCA) (Ma and Dai, 2011). PCA identifies a series of orthogonal principal components (PCs), in which each PC corresponds to a linear combination of the original features. The PCs are constructed such that the first PC (PC1) contains the largest proportion of variance of the data, followed by PC2, then by PC3, and so on. Dimensionality reduction can be achieved by selecting only the first k PCs that cumulatively explain sufficient proportion of variance. In other words, PCA is used to project a data set of n dimensions into k projected PCs.

Another important consideration in gene expression clustering analysis is the definition of distance between two points (two genes or two samples). In practice, the most common distance measures are Euclidean distance and Pearson correlation coefficient (when used as a distance, it is typically represented as $1 - |\text{correlation}|$ or $(1 + \text{correlation})/2$). A more comprehensive comparison of distance measures for gene expression clustering can be found elsewhere (Jaskowiak *et al.*, 2014).

Correlation coefficient is less sensitive to the scale of data than Euclidean distance. Consider the case of a pair of co-expressed genes in which one gene is highly expressed while other one has significantly lower expression. These two genes would have a large Euclidean distance but strong correlation. In this case the choice of Euclidean distance or correlation as a measure of dissimilarity is largely driven by the underlying biological question. Sometimes it might be important to distinguish genes that have large difference in expression levels, whereas sometimes it is more important to determine correlation without worrying too much about the absolute expression levels. Another possible analysis strategy is to first normalise the expression of both genes to have a mean of zero and a standard deviation of one (i.e., conversion to standardised scores). In this case, both Euclidean distance and correlation would yield similar results.

Euclidean distance and correlation coefficient are both sensitive to outliers or missing data. Missing data is a major issue in the analysis of single-cell RNA-seq (scRNA-seq) data because many transcripts are not detectable due to dropout events in the experimental process. In the case of scRNA-seq data, it is often not possible to distinguish whether a lack of gene expression (zero read count per gene) is an artefact caused by dropout or it reflects true non-expression of that gene. One simple approach to alleviate the impact of dropouts in scRNA-seq data set is to use ‘implicit imputation’ in the calculation of Euclidean distance (Lin *et al.*, 2017). This approach is implemented in an R package CIDR for dimensionality reduction and clustering of scRNA-seq data.

Case Study

Let’s put everything together. In this case study, we analysed a time series gene expression data set of mouse embryonic dental epithelium (Epi) and dental mesenchym (Mes) between embryonic day 11.5 and 13.5 (E11.5-E13.5) (O’Connell *et al.*, 2012), with three replicates per time point per tissue. This data set consists of 30 gene expression profiles in total (5 time points x 2 tissues x 3 replicates). The gene expression profiles were obtained by an Illumina microarray platform. The data can be downloaded from GEO using accession number GSE32321. The goal of the analysis is to identify the gene expression dynamics in these two adjacent developing dental tissues, and to identify genes that are driving those dynamic patterns.

After downloading the data, we first checked that the quality of the data. Fig. 3(a) shows the histogram of expression levels (in logarithmic scale) of one representative sample in the data set. This plot illustrates the characteristics of a microarray-based gene expression profile: a sharp peak on the left which consists of mostly unexpressed or lowly expressed genes, and a long tail on the right which consists of expressed genes. Based on this plot, approximately half of the genes are unexpressed or have low expression (the left peak), and the other half has a wide range of expression levels (the long right tail). The plot also shows that there is an absence of outliers (values that are exceptionally small or large). These are all characteristics of a good quality gene expression profile. In RNA-seq data, the distinction between the left peak and the right tail is often much clearer.

We next checked whether the gene expression distributions were consistent across all samples in a data set. Fig. 3(b) is a boxplot showing the distribution of all 30 samples in this data set. The interquartile range and median of all the samples are highly consistent, suggesting that further normalisation is not necessary.

We then wanted to visualise the gene expression data using a heat map. Nonetheless, rendering a heat map of the entire data set (consisting of 45,281 gene probes) is computationally difficult on standard computer screens and difficult to visually interpret by humans. We therefore decided to analyse and visualise the 100 most variably expressed genes, as defined as the 100 genes with the highest variance across the 30 samples. Analysis of these high-variance genes often allows us to gain insight into the major significant patterns in the full data set. We performed hierarchical clustering on the genes and samples. The hierarchical clustering was performed using Euclidean distance and the Ward method for aggregation. The results are displayed using a heat map in which a gene represent as a row and a sample is represented as a column (Fig. 3(c)). One striking feature is that samples from epithelial and mesenchymal tissues are grouped into two distinct clusters. The replicates generally group together, with only one minor exception (E11.5 epithelium), suggesting that the replicates are generally highly consistent. Within each tissue, the samples appear to be grouped based on temporal ordering of samples. In particular, samples from E11.5 and E12 tend to group together, and samples from E12.5-E13.5 tend to group together.

We further visualised the relationships among the samples by performing PCA, and displaying the sample based on the first two principal components (PC1 and PC2; Fig. 3(d)). The first two components collectively explains about 90% of the variance in the data set, with most of the variance explained by the PC1 (76%). We can see that PC1 clearly separates epithelial and mesenchymal tissues. PC2

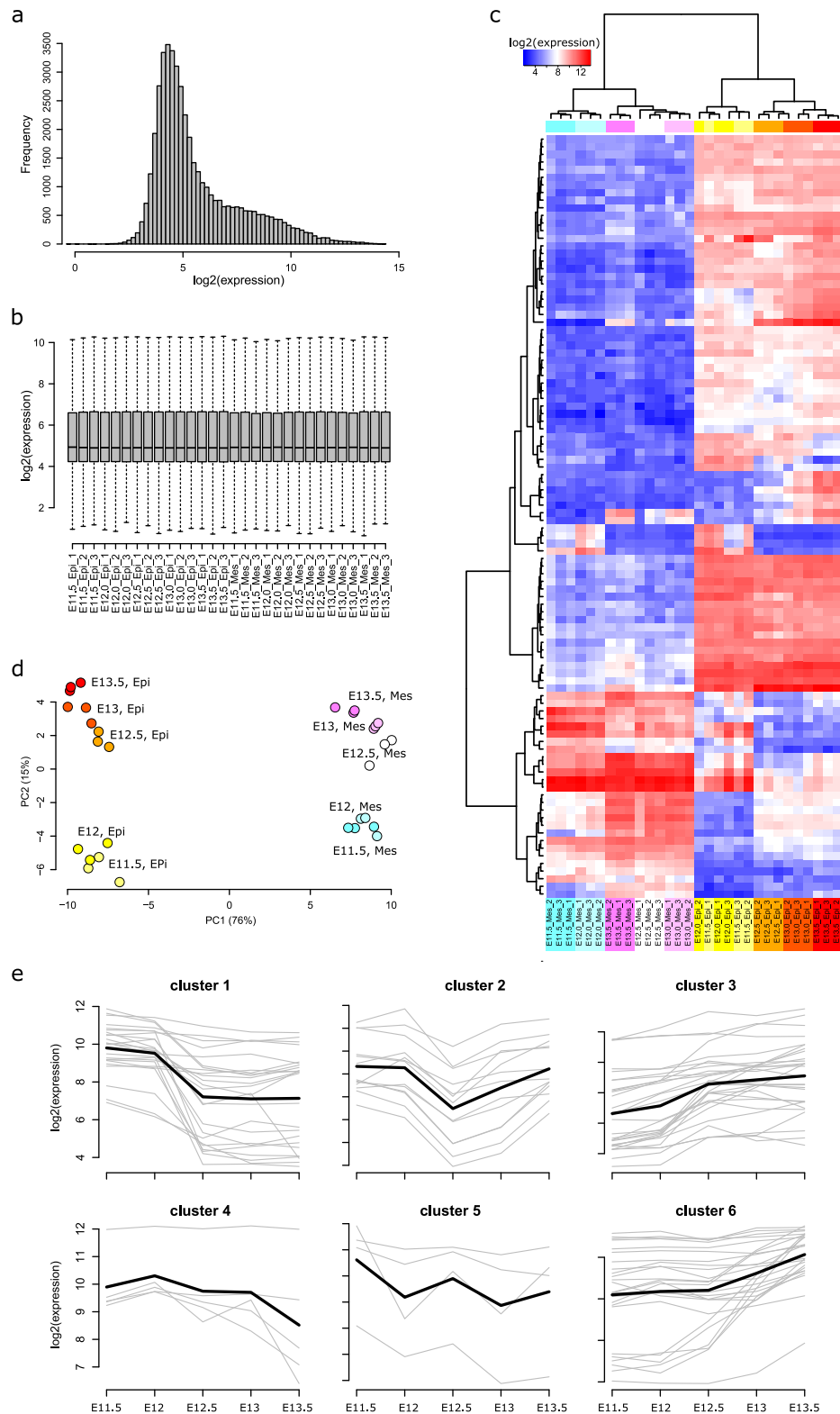


Fig. 3 A case study on expression clustering of an embryonic dental tissue development data set. (a) The gene expression distribution of a typical sample in this data set. (b) A box plot showing the distribution of gene expression values is largely consistent across all the samples in the data set. (c) A heat map showing the gene expression levels of the 100 most variably expressed genes in this data set. Hierarchical clustering has been performed on the samples and genes. The clustering results are displayed as dendrograms around the heat map. (d) The data set is subjected to the principal component analysis (PCA), and the samples are displayed based on first two principal components (PC1 and PC2). (e) The top 100 variably expressed genes are subjected to k -means clustering to identify 6 clusters. The dark line in each plot represents the mean expression pattern of all the genes in a cluster.

generally separates early time points (E11.5 and E12) and later time points (E12.5-E13.5) in both tissues. This implies that there is a major transcriptomic change between E12 and E12.5 in both tissues. This plot clearly shows the spatial temporal clustering of samples.

Finally we would like to identify clusters of genes with similar gene expression patterns. Here we only analysed the epithelial samples for simplicity (the mesenchymal samples can be analysed in a similar manner). We first normalised the data such that each gene has a mean value of zero and a standard deviation of one. This process ensures that all the genes are comparable in scale. After normalisation, we performed k -means clustering to identify clusters of genes. In this example, we set $k=6$, but in general one should attempt a reasonable range of k . The resulting 6 clusters are displayed in Fig. 3(e). One salient pattern is that all clusters have a major transition point at E12.5, corroborating the pattern observed in the PCA plot (Fig. 3(d)). Through looking at the plots, some clusters can clearly be further divided into smaller clusters. For example, cluster 6 clearly consists of two major clusters – one characterised by a smooth and gradual increase in expression of mostly highly expressed genes, and the other characterised by a massive up-regulation of lowly expressed genes starting at E12. Further downstream bioinformatics analyses, such as gene set analysis (Djordjevic *et al.*, 2016; de Leeuw *et al.*, 2016), can be performed to identify what are genes that are enriched in each of these clusters.

This case study illustrates how clustering of genes and samples of a gene expression data set can lead to discovery of biologically relevant patterns, and highlight the overall workflow of clustering in gene expression bioinformatics analysis.

Summary

Clustering is one of the most fundamental bioinformatics tools, especially in terms of gene expression analysis. As discussed in this article, care must be taken to ensure that proper feature filtering, dimensionality reduction and quality controls are undertaken prior to clustering. Also, clustering results must be evaluated and interpreted carefully, especially mindful of the characteristics of different clustering algorithms. When performed properly, clustering is a powerful tool to discover co-regulated genes, patient groupings, and sub-population of cells.

Acknowledgement

The authors are supported by the Victor Chang Cardiac Research Institute. JWKH is also supported by a Career Development Fellowship by the National Health and Medical Research Council of Australia and a Future Leader Fellowship from the National Heart Foundation of Australia.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Analyzing Transcriptome-Phenotype Correlations. Characterization of Transcriptional Activities. Comparative Transcriptomics Analysis. Computation Cluster Validation in the Big Data Era. Data Mining: Clustering. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Differential Expression from Microarray and RNA-seq Experiments. Functional Enrichment Analysis Methods. Genome-Wide Scanning of Gene Expression. Natural Language Processing Approaches in Bioinformatics. Regulation of Gene Expression. Statistical and Probabilistic Approaches to Predict Protein Abundance. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation. Transcriptome Informatics. Unsupervised Learning: Clustering. Utilising IPG-IEF to Identify Differentially-Expressed Proteins

References

- Altman, N., Krzywinski, M., 2017. Points of significance: Clustering. *Nat. Methods* 14, 545–546.
- Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J., 1999. OPTICS: Ordering points to identify the clustering structure. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, (New York, NY, USA: ACM), pp. 49–60.
- Arthur, D., Vassilvitskii, S., 2007. K-means ++: The advantages of careful seeding. In: *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*.
- Assent, I., 2012. Clustering high dimensional data. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2, 340–350.
- Barrett, T., Suzek, T.O., Troup, D.B., *et al.*, 2005. NCBI GEO: Mining millions of expression profiles – Database and tools. *Nucleic Acids Res.* 33, D562–D566.
- Barrett, T., Wilhite, S.E., Ledoux, P., *et al.*, 2013. NCBI GEO: Archive for functional genomics data sets – Update. *Nucleic Acids Res.* 41, D991–D995.
- Basso, K., Margolin, A.A., Stolovitzky, G., *et al.*, 2005. Reverse engineering of regulatory networks in human B cells. *Nat. Genet.* 37, 382–390.
- Bottomly, D., Walter, N.A.R., Hunter, J.E., *et al.*, 2011. Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-Seq and microarrays. *PLOS ONE* 6, e17820.
- Cloonan, N., Forrest, A.R.R., Kolle, G., *et al.*, 2008. Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nat. Methods* 5, 613–619.
- de Leeuw, C.A., Neale, B.M., Heskies, T., Posthuma, D., 2016. The statistical properties of gene-set analysis. *Nat. Rev. Genet.* 17, 353–364.
- DeRisi, J.L., Iyer, V.R., Brown, P.O., 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–686.
- Djordjevic, D., Kusumi, K., Ho, J.W.K., 2016. XGSA: A statistical method for cross-species gene set analysis. *Bioinformatics* 32, i620–i628.
- Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95, 14863–14868.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, (Portland, Oregon: AAAI Press), pp. 226–231.
- Friedman, N., Linial, M., Nachman, I., Pe'er, D., 2000. Using Bayesian networks to analyze expression data. *J. Comput. Biol.* 7, 601–620.
- Fu, X., Fu, N., Guo, S., *et al.*, 2009. Estimating accuracy of RNA-Seq and microarrays with proteomics. *BMC Genomics* 10, 161.
- Gan, G., Ng, M.K.-P., 2017. k -means clustering with outlier removal. *Pattern Recognit. Lett.* 90, 8–14.
- Golub, T.R., Slonim, D.K., Tamayo, P., *et al.*, 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.

- Hahsler, M., Piekenbrock, M., Doran, D., 2017. dbscan: Fast density-based Clustering with R.
- Hinneburg, A., Keim, D.A., 1998. An efficient approach to clustering in large multimedia databases with noise. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. (New York, NY: AAAI Press), pp. 58–65.
- Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. *ACM Comput. Surv.* 31, 264–323.
- Jaskowiak, P.A., Campello, R.J., Costa, I.G., 2014. On the selection of appropriate distances for gene expression data clustering. *BMC Bioinform.* 15, S2.
- Kriegel, H.-P., Kröger, P., Sander, J., Zimek, A., 2011. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 1, 231–240.
- Lin, P., Troup, M., Ho, J.W.K., 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.* 18, 59.
- Liu, B., 2006. A fast density-based clustering algorithm for large databases. In: *2006 International Conference on Machine Learning and Cybernetics*, pp. 996–1000.
- Ma, S., Dai, Y., 2011. Principal component analysis based methods in bioinformatics studies. *Brief. Bioinform.* 12, 714–722.
- Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K., 2017. *Cluster: Cluster analysis basics and extensions*.
- Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., Gilad, Y., 2008. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.* 18, 1509–1517.
- Meunier, B., Dumas, E., Pic, I., *et al.*, 2007. Assessment of hierarchical clustering methodologies for proteomic data mining. *J. Proteome Res.* 6, 358–366.
- Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B., 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods* 5, 621–628.
- Nagalakshmi, U., Wang, Z., Waern, K., *et al.*, 2008. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320, 1344–1349.
- O'Connell, D.J., Ho, J.W.K., Mammoto, T., *et al.*, 2012. A Wnt-Bmp feedback circuit controls Intertissue signaling dynamics in tooth organogenesis. *Sci. Signal* 5, ra4.
- Pedregosa, F., Varoquaux, G., Gramfort, A., *et al.*, 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Ramaswamy, S., Tamayo, P., Rifkin, R., *et al.*, 2001. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci.* 98, 15149–15154.
- Rehman, S.U., Asghar, S., Fong, S., Sarasvady, S., 2014. DBSCAN: Past, present and future. In: *Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pp. 232–238.
- Ronan, T., Qi, Z., Naegle, K.M., 2016. Avoiding common pitfalls when clustering biological data. *Sci. Signal* 9, re6.
- Schena, M., Shalon, D., Davis, R.W., Brown, P.O., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467–470.
- Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X., 2017. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst.* 42, 19:1–19:21.
- Stuart, J.M., Segal, E., Koller, D., Kim, S.K., 2003. A gene-coexpression network for global discovery of conserved genetic modules. *Science* 302, 249–255.
- Su, A.I., Welsh, J.B., Sapinoso, L.M., *et al.*, 2001. Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Res.* 61, 7388–7393.
- Wang, Z., Gerstein, M., Snyder, M., 2009. RNA-Seq: A revolutionary tool for transcriptomics. *Nat. Rev. Genet.* 10, 57–63.
- Ward, J.H., 1963. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* 58, 236–244.
- Wilhelm, B.T., Marguerat, S., Watt, S., *et al.*, 2008. Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature* 453, 1239–1243.
- Zhang, J., Wu, G., Hu, X., Li, S., Hao, S., 2011. A parallel K-means clustering algorithm with MPI. In: *2011 Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms and Programming*, pp. 60–64.

Biographical Sketch



Xiaoxin Ye is a Masters student at the University of New South Wales (UNSW), and an intern at the Bioinformatics and Systems Medicine Laboratory at the Victor Chang Cardiac Research Institute. He received his BSc in Statistics from Guangzhou Medical University and a Graduate Certificate in Mathematics and Statistics from UNSW. His research focuses on developing fast and scalable algorithms for clustering single cell RNA-seq (scRNA-seq) data. His interests include scalable clustering algorithms, shared neighbours clustering algorithms, as well as statistical and parallelised machine learning.



Joshua Ho is the Head of Bioinformatics and Systems Medicine Laboratory at the Victor Chang Cardiac Research Institute in Sydney Australia. He is also a conjoint Senior Lecturer at UNSW Sydney. Dr Ho completed his BSc (Hon 1, Medal) and PhD in Bioinformatics at the University of Sydney, and undertook postdoctoral research at the Harvard Medical School in the United States. His current research focuses on developing fast and reliable bioinformatics methods to identify the genetic cause and mechanism of heart diseases, using a range of approaches such as whole genome sequencing, single-cell RNA-seq, ChIP-seq, machine learning, systems biology, cloud computing, and software testing and quality assurance. His research excellence has been recognised by the 2015 NSW Ministerial Award for Rising Star in Cardiovascular Research, the 2015 Australian Epigenetics Alliance' Illumina Early Career Research Award, and the 2016 Young Tall Poppy Science Award.

Metabolome Analysis

Saravanan Dayalan, The University of Melbourne, Parkville, VIC, Australia

Jianguo Xia, McGill University, Sainte Anne de Bellevue, QC, Canada and McGill University, Montreal, QC, Canada

Rachel A Spicer and Reza Salek, European Bioinformatics Institute (EMBL-EBI), Hinxton, United Kingdom

Ute Roessner, The University of Melbourne, Parkville, VIC, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Metabolites, including lipids, are biological active compounds important as building blocks for macromolecules and carriers of energy, both required in all living cells to participate in all biological processes. They are members of thousands of biosynthetic pathways providing the chemical diversity to fulfil specific functions. The chemical diversity of metabolites is enormous and has challenged biochemists and physiologists to provide knowledge on many biological processes.

This challenge has led to the development of metabolomics utilizing technological advances in analytical chemistry; it is today a scientific field used in many sectors, including biomedicine, plant and agricultural sciences, food and beverage sciences, microbial and environmental sciences, forensic, and many more. The analytical challenge is to tackle mostly the chemical diversity of metabolites which are characterized by thousands of unique chemical structures with diverse physical and chemical properties, including molecular weight, molecular size and shape, polarity, stability, volatility, solubility, and so on. These chemical and physical features determine the requirements for extraction, separation, detection and quantification. Therefore, a range of complementary methodologies for extraction and analysis must be used to cover the range of metabolites present in any living cell, making metabolomics an exciting playground for analytical chemists and biochemists.

A range of analytical instrumentation has been successfully utilized to analyze metabolites in many different organisms, tissue types or biofluids. These include Mass Spectrometry (MS) and Nuclear Magnetic Resonance (NMR) for the detection and quantification of metabolites, the former often coupled to chromatographic separation techniques to separate the highly complex metabolite extracts obtained from any biological sample. One of the well-established “work horses” in metabolomics is gas chromatography (GC) coupled to mass spectrometry. It is characterized by its great separation power and reproducibility, which allowed for the development of comprehensive mass spectral libraries for compound identification which are easily adaptable for routine applications on any GC–MS system available ([Hill and Roessner, 2013](#)). GC–MS in general enables the analysis of around 400 compounds in a single run depending on tissue type, including amino, organic and fatty acids, sugars, amines and sterols. Important to note here is that GC–MS also plays an important role in metabolic flux analysis using stable isotope labelling techniques. This is because of its great ability to detect isotopic patterns in each metabolite detected therefore allowing quantification of stable isotope enrichments which can then be refereed back to actual carbon or nitrogen flux within a pathway of interest. One major drawback of GC–MS is, however, that only compounds which are either volatile or can be made volatile using chemical derivatization are amenable for analysis, which are mainly low molecular weight compounds, therefore complementary techniques such as liquid chromatography (LC) coupled to MS need to be used to cover the diversity of the metabolome.

LC-MS in metabolomics has attracted enormous attention in the past few years, mainly driven by great technological step-changes making LC-MS much for reproducible, sensitive and easy to use ([Hill and Roessner, 2014](#)). Today, common LC-MS methodologies allow analysis of thousands of metabolites simultaneously, therefore covering a large portion of the metabolome, making LC-MS an ideal tool for biomarker discovery and for generating novel hypotheses. An important aspect for LC-MS-based metabolomics is the choice of separation chemistry and suitable elution procedures addressing the chemical diversity of the metabolome. Unfortunately, no single methodology is able to separate all compound classes found in any biological system. The main used separation chemistries are based on reverse phase and hydrophilic interactions, when considering different ionization techniques available, e.g. electrospray ionization or atmospheric pressure ionization, which will again ionize different compounds, and positive and negative ionization to cover differentially charged ions produced in the ionization process, the number of potential methods increase enormously. However, if one aims to measure “all” metabolites within a sample, all these different options need to be explored and utilized. Each individual approach may detect between 500 and 2000 compounds (depending on tissue type and extraction procedure), represented as mass features with an accurate mass, isotopic pattern and chromatographic retention time. Accurate mass and isotopic pattern in addition to secondary and tertiary MS-based fragmentation (MS^2 or MS^3) can aid structural elucidation of compounds. A difficulty in this approach are identifications of metabolites which are similar in structure, mass and their fragmentation patterns, such as sugars. This makes unambiguous identification almost impossible and therefore requires good chromatographic separation and confirmation of identification with authentic standards. Commercial and open source mass spectral libraries such as MassBank (see Relevant Websites section), Human Metabolome Database (HMD; see Relevant Websites section), Metlin (see Relevant Websites section) or ChempSpider (see Relevant Websites section) can aid identification, however often these libraries only record accurate mass of compounds but not their structure-specific fragmentation patterns or chromatographic retention times. Therefore unambiguous identification of compounds detected using LC-MS approaches remain limited leading to limited ability of biochemical and biological interpretation ([Sumner et al., 2014](#)). There have been many initiatives globally attempting to develop better solutions for metabolite identification such as [Creek et al. \(2014\)](#), [Bingol et al. \(2016\)](#), [Lynn et al. \(2015\)](#), [Pahler and Brink \(2013\)](#), [Shen et al. \(2014\)](#).

As described above, methodologies in metabolomics are as diverse as metabolites themselves, but taken together allow detection and quantification of thousands of compounds in any given sample. This leads to the almost greater challenge of extracting information from the analytical instrumentation, handling complex and multi-dimensional data sets and ultimately interpreting the vast amount of information in a biological context. This offers an exciting opportunity for collaboration between biologists, analytical chemists, bioinformaticians and statisticians to develop and apply sophisticated computational and statistical tools for metabolomics data analysis and interpretation. Below, we discuss in detail the various software and statistical tools and methods that are used for the analysis, visualization and interpretation of metabolomics data including metabolomics workflows, metadata, standards in metabolomics, data processing and workflow softwares, metabolomics experiments repositories, metabolomics databases and statistical analysis methods for analyzing metabolomics data. A collation of these software tools, repositories and databases are shown in [Table 1](#).

Metabolomics Workflow and Lims Solutions

The typical workflow of a metabolomics experiment starts with establishing the experimental design followed by sample collection, sample preparation, running of samples in analytical instruments, data processing, statistical analysis and pathway analysis ([Fig. 1](#)).

Of these steps, the first step of establishing the experimental design could easily be termed as one of the most important steps in the experimental workflow. Experiments that are setup with careful consideration of how different variables may affect the experimental outcome would produce the best results. Some of the basic aspects that needs careful consideration are establishing the correct groups, sampling the population appropriately such that the samples best describe the underlying population, establishing the number of sample (biological replicates) and reducing variability of samples within groups. The adage used so frequently in Computer Science ‘Garbage In, Garbage Out’ is well-suited for experimental designs and the eventual results they would produce. In order to establish the most suited experimental design that would best tease out the answer to the scientific query, consulting with a biostatistician before the beginning of the experiment is imperative. RA Fisher, the father of modern statistics and experimental design rightly said, “To call in the statistician after the experiment is done may be no more than asking him to perform a post-mortem examination: he may be able to say what the experiment died of.” (“RA Fisher” 1938). Therefore, statistical considerations of metabolomics experiments should not come into play after the data has been processed, but even before the experiment is designed.

Bioinformatics analyses of metabolomics data starts once the samples are run in analytical instruments and raw data generated. There are various algorithms and softwares that process the instrument generated raw data thereby converting them into a two-dimensional data matrix consisting of a measure of the different metabolites found in the samples. A detailed discussion of the data processing softwares of different platforms is presented below. After the raw data is processed and the data matrix obtained, various types of statistical analyses are performed in order to find metabolites that are significantly different between the compared groups. A detailed discussion on the different types of statistical tests is presented below. More often than not, statistical significance is determined using the p-value. In recent years, there have been substantial discussion in the scientific community about the dangers of relying only on measures such as the p-value to draw conclusions on the importance of a measured variable ([Baker, 2016](#); [Huber, 2016](#)). While p-values give a measure of statistical significance of the measured metabolites, this may not necessarily translate to biological significance. Other measures such as the 95% confidence intervals needs to be taken into consideration before deciding on the biological significance of metabolites. In metabolomics experiments, hundreds if not thousands of metabolites are measured and while investigating the significance of these metabolites, care must be taken to correct for multiple tests. When multiple tests of significance are performed as part of a single experiment, the resulting statistical significance needs to be controlled for false positives ([Curran-Everett, 2000](#)).

There are several softwares and software systems that focus on different parts of the metabolomics experimental workflow. Software systems known as Laboratory Information Management Systems (LIMS) focus on capturing all meta data and any resulting data of an experiment. Generally, LIMS solutions capture information about the project, the several experiments under the project, information on the type of analytical platforms the experiments are run, details of the runs themselves such as the analytical methods and any SOPs used and detailed information about the samples that are run. This include details such as the biological nature of the samples and any metadata associated with the samples such as age, sex, treatment, quantity, weight etc. There are several LIMS solutions currently present for metabolomics.

MASTR-MS ([Hunter et al., 2017](#)) is a web-based LIMS solution that can be deployed either within a single laboratory or in a collaborative multi-site network. It comprises a Node Management System that can be used to link and manage projects across one or multiple collaborating laboratories; a User Management System which defines different user groups and privileges of users; a Quote Management System where client quotes are managed; a Project Management System in which metadata is stored and all aspects of project management, including experimental setup, sample tracking and instrument analysis, are defined, and a Data Management System that allows the automatic capture and storage of raw and processed data from the analytical instruments to the LIMS. SetupX ([Scholz and Fiehn, 2007](#)) is a web based metabolomics LIMS solution that is XML compatible and built around a relational database management core. It is particularly oriented towards the capture and display of GC–MS metabolomics data through its metabolic annotation database, BinBase ([Skogerson et al., 2011](#)). SetupX is able to handle a wide variety of BioSources (spatial, historical, environmental and genotypic descriptions of biological objects undergoing metabolomic investigations) and

Table 1 Metabolomics software tools, repositories and databases

LIMS solutions			
Software tool	Platform		Reference
MASTR-MS	Web-based		Hunter <i>et al.</i> (2017)
SetupX	Stand-alone		Scholz and Fiehn (2007)
Sesame	Web-based		“Sesame LIMS” (2017)
Data preprocessing tools			
Software tool	Instrument data type		Reference
XCMS	LC-MS, GC-MS		Smith <i>et al.</i> (2006)
MS-DIAL	LC-MS		Tsugawa <i>et al.</i> (2015)
mzMatch	LC-MS		Scheltema <i>et al.</i> (2011)
MetAlign	LC-MS		Lommen and Kools (2012)
AMDIS	GC-MS		Meyer <i>et al.</i> (2010)
MetaboliteDetector	GC-MS		Hiller <i>et al.</i> (2009)
MET-IDEA	GC-MS		Broeckling <i>et al.</i> (2006)
MeltDB	LC-MS, GC-MS		Kessler <i>et al.</i> (2013)
MSeasy	GC-MS		Nicolè <i>et al.</i> (2012)
Metabolite annotation tools			
Software tool	Identification		Reference
CAMERA	Level 4		Kuhl <i>et al.</i> (2012)
MZedDB	Level 4		Draper <i>et al.</i> (2009)
SIRIUS	Level 4		Bocker <i>et al.</i> (2009)
MI-PACK	Level 3		Weber and Viant (2010)
PUTMEDID-LCMS	Level 3		Brown <i>et al.</i> (2011)
CFM-ID	Level 2a		Allen <i>et al.</i> (2014)
MAGMa	Level 2a		Ridder <i>et al.</i> (2013)
MetFrag	Level 2a		Ruttkies <i>et al.</i> (2016)
BATMAN	NMR		Hao <i>et al.</i> (2012)
Bayesil	NMR		Ravanbakhsh <i>et al.</i> (2015)
Data analysis softwares			
	Software system	Instrument data type	Reference
Workflow Management Systems (WMS)	Workflow4metabolomics	LC-MS, GC-MS NMR	Giacomoni <i>et al.</i> (2015)
	Galaxy-M	LC-MS, DIMS	Davidson <i>et al.</i> (2016)
	PhenoMeNal	LC-MS NMR, Isotope labelled data	http://phenomenal-h2020.eu/
Non-WMS tools	XCMS Online	LC-MS, GC-MS	Tautenhahn <i>et al.</i> (2012)
	MZmine2	LC-MS, GC-MS	Pluskal <i>et al.</i> (2010)
	FOCUS	NMR	Alonso <i>et al.</i> (2014)
	MatNMR	NMR	Van Beek (2007)
Metabolomics repositories			
Repository			Reference
MetaboLights			Haug <i>et al.</i> (2013)
Metabolomics Workbench			Sud <i>et al.</i> (2016)
GNPS			Wang <i>et al.</i> (2016)
MeRY-B			Ferry-Dumazet <i>et al.</i> (2011)
MetaPhen			Carroll <i>et al.</i> (2010)
MetabolomeXchange			http://www.metabolomexchange.org/
OmicsDI			Perez-Riverol <i>et al.</i> (2016)
Metabolomics databases			
Database type	Database		Reference
Mass Spectral Databases	NIST 17		“NIST Standard Reference Database 1A v17” (2017)
	Wiley Registry		“Wiley Registry of Mass Spectral Data, 11th Edition” (2017)
	Massbank		Horai <i>et al.</i> (2010)
	METLIN		Smith <i>et al.</i> (2005)

(Continued)

	Golm	Kopka <i>et al.</i> (2005)
	MMCD	Cui <i>et al.</i> (2008)
	SDBS	"SDBSWeb" (2017)
Species-Specific Databases	HMDB	Wishart <i>et al.</i> (2013)
	KNAPSAcK	Nakamura <i>et al.</i> (2014)
	ReSpect	Sawada <i>et al.</i> (2012)
Pathway Databases	KEGG	Arakawa <i>et al.</i> (2005)
	MetaCyc	Caspi <i>et al.</i> (2016)
	BioCyc	Caspi <i>et al.</i> (2016)
	Reactome	Fabregat <i>et al.</i> (2016)
	PMN	Kruger and Ratcliffe (2012)

Treatments (experimental alterations that influence the metabolic states of BioSources). Sesame ("[Sesame LIMS](#)", 2017) is also a web-based, platform-independent LIMS. It was originally developed to facilitate NMR-based structural genomics studies. The Sesame module for metabolomics is called 'Lamp'. The Lamp module was originally designed to process NMR metabolomic analyses of Arabidopsis, although it is flexible enough to be easily adapted to other biological systems and other analytical methods.

Data Processing and Workflow Management Solutions

In metabolomics, a number of analytical techniques are typically used for analysis, with the most popular being liquid chromatography-mass spectrometry (LC-MS), gas chromatography-mass spectrometry (GC-MS) and nuclear magnetic resonance (NMR) spectroscopy. The data produced by these analytical techniques is distinct and requires different data handling methods. There is therefore the need for analysis tools and workflows designed for handling each data format.

The analysis of metabolomics data typically includes: pre-processing, annotation, post-processing (pre-treatment) and statistical analysis as stages of data processing. However, the order of processing varies based on the analytical technique used. Untargeted LC-MS data analysis is generally performed in the above order, whilst with targeted LC-MS data only metabolites of interest are measured and there is no further annotation stage. Typically, with GC-MS pre-processing and annotation are performed concurrently due to the wide availability of reference libraries. NMR is an inherently more quantitative technique than MS and when metabolites are identified they are also typically relatively quantified.

Typically for mass spectrometry data, prior to preprocessing, data must be converted into an open format e.g. mzML, mzXML and netCDF, as the majority of tools do not accept proprietary formats. Most proprietary formats can be converted to mzML or mzXML using msconvert, a proteowizard ([Chambers *et al.*, 2012](#)) project tool. For NMR, there is also an open data exchange format known as JCAMP-DX version 6.0 11, with many different vendor-dependent variants. Recently, as part of the COSMOS (COordination of Standards in MetabOlomicS) EU FP7 consortium initiative ([Salek *et al.*, 2015](#)), the XML open standard nmrML, has been introduced ([nrmML, 2017](#)).

The most popular MS metabolomics software for preprocessing is XCMS ([Weber *et al.*, 2016](#)). XCMS ([Smith *et al.*, 2006](#)) is a well established software, having first been released over a decade ago. It is a bioconductor R package that can be used for the preprocessing of LC-MS or GC-MS data. Other popular software for LC-MS data preprocessing includes mzMatch ([Scheltema *et al.*, 2011](#)), Mass Spectrometry-Data Independent AnaLysis (MS-DIAL) ([Tsugawa *et al.*, 2015](#)) and MetAlign ([Lommen and Kools, 2012](#)). Like XCMS, mzMatch is also an R package. As well as preprocessing, it additionally provides isotopic labelling analysis ([Chokkathukalam *et al.*, 2013](#)) and probabilistic metabolite annotation ([Daly *et al.*, 2014](#)). Both MS-DIAL and MetAlign are exclusively available for windows operating systems. MS-DIAL can preprocess LC- and GC- MS and tandem MS (MS/MS) data, whereas MetAlign is only able to process MS1 LC-MS and GC-MS data. For GC-MS data, AMDIS ([Meyer *et al.*, 2010](#)) (Automated Mass Spectral Deconvolution and Identification System) is the most widely used processing tool. Pure compound peaks, free of overlapping signals, are extracted by deconvoluting spectra. A user defined target library can then be used for metabolite identification. However, it is worth noting that AMDIS does not include spectral alignment.

There are also many alternative software specifically for GC-MS preprocessing, including: MetaboliteDetector ([Hiller *et al.*, 2009](#)), MET-IDEA ([Broeckling *et al.*, 2006](#)), MeltDB ([Kessler *et al.*, 2013](#)), metaMS ([Wehrens *et al.*, 2014](#)) and MSeasy ([Nicolè *et al.*, 2012](#)). Baseline correction, smoothing, peak detection and deconvolution are provided by MetaboliteDetector. MET-IDEA quantifies AMDIS outputs, also generating a list of mass spectral tags. MeltDB is a suite of molecular tools, including peak picking, retention indices calculation and sum formula annotation. metaMS is based on XCMS, but performs pseudospectra analysis to avoid the alignment stage that can be hard to execute with GC-MS.

Compared to for MS data preprocessing, there are far less tools for NMR data. Examples include Dolphin ([Gómez *et al.*, 2014](#)) and rNMR ([Lewis *et al.*, 2009](#)). Both 1D and 2D NMR spectra can be analysed by Dolphin and rNMR. Dolphin automatically quantifies a set of target metabolites. rNMR uses a regions of interest (ROIs) based approach for analysis.

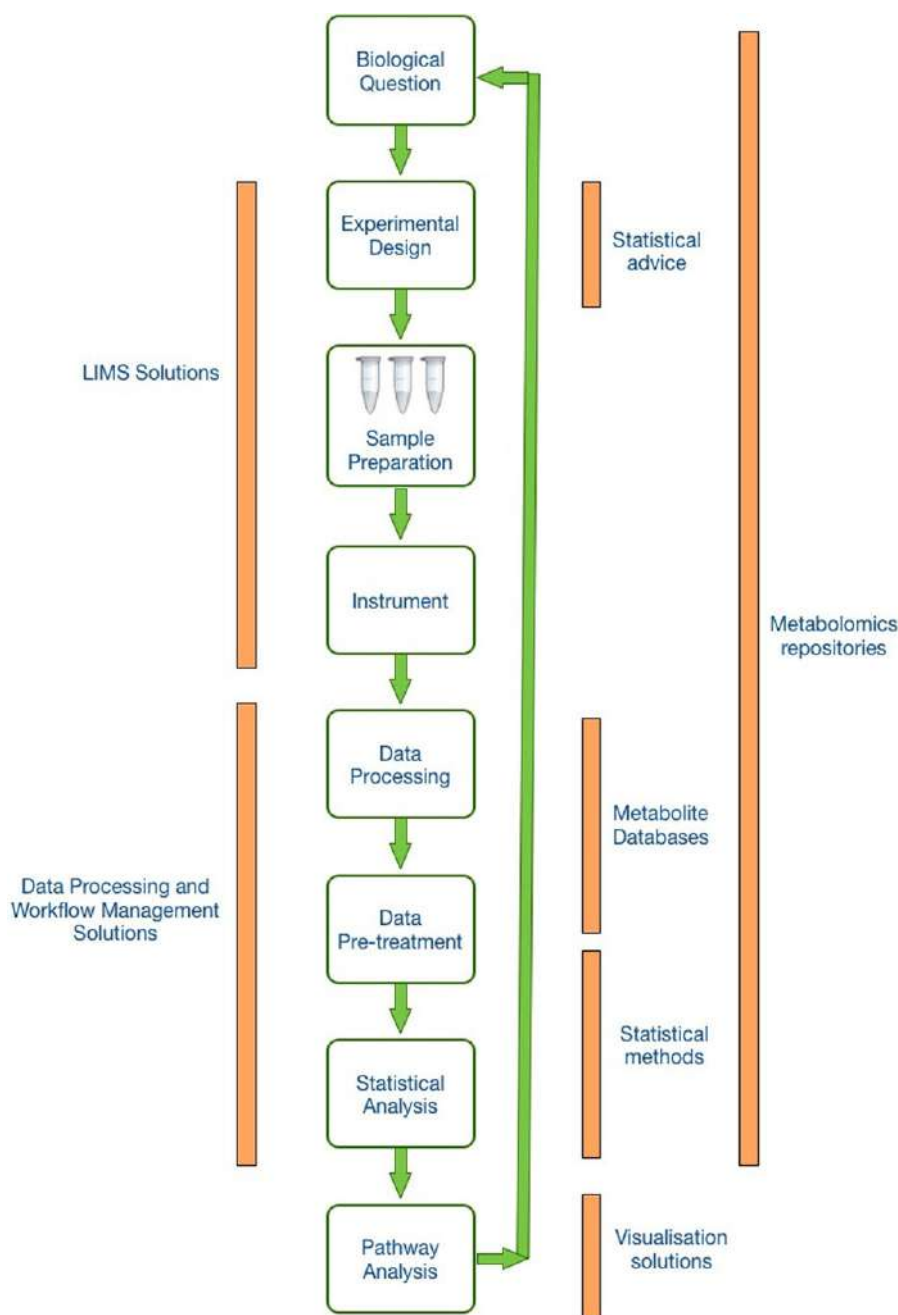


Fig. 1 Metabolomics experimental workflow.

Various software tools for the annotation of preprocessed MS data provide different levels of identification. Some annotate adducts, isotopes and fragments (e.g. CAMERA [Kuhl et al., 2012](#), MZedDB [Draper et al., 2009](#)) or empirical formula (e.g. SIRIUS [Bocker et al., 2009](#)). Others automatically search metabolite databases to provide tentative metabolite candidates (e.g. MI-PACK [Weber and Viant, 2010](#), PUTMEDID-LCMS [Brown et al., 2011](#)). MS/MS or MSⁿ data can be assigned a candidate metabolite at a higher level of confidence by searching against spectral libraries acquired using authenticated chemical standards. Examples of software that perform automatic database matching includes: CFM-ID ([Allen et al., 2014](#)), CSI: FingerID ([Dührkop et al., 2015](#)) and MCID ([Li et al., 2013a](#)). MAGMa ([Ridder et al., 2013](#)) and MetFrag ([Ruttkies et al., 2016](#)) are both *in silico* fragmentation tools. Metabolite identification and quantification are often performed simultaneously for NMR data. BATMAN ([Hao et al., 2012](#); [Ravanbakhsh et al., 2015](#)) deconvolutes and quantifies ¹H NMR spectra and Bayesil ([Ravanbakhsh et al., 2015](#)) performs automatic spectral processing and identification of serum, plasma and cerebrospinal fluid (CSF) 1D ¹H NMR spectra. These tools provide different levels of identification (levels 1 to 4) as described by the Metabolomics Standards Initiative ([Sumner et al., 2007](#)).

For a more in depth look of metabolomics data handling, data processing and data analysis see reviews (Spicer *et al.*, 2017; O'sullivan *et al.*, 2011; Hoffmann and Stoye, 2012; Beiske *et al.*, 2015).

Workflows in Metabolomics

Data analysis workflows or pipelines function to provide all stages of data analysis in a single space, including sequential data processing steps provided by multiple interconnected tools. This benefits the researcher by increasing computational reproducibility and removing problems caused by lack of interoperability between tools. In metabolomics, the majority of workflows have been designed to handle LC-MS data, as it is the most widely used instrumental technique.

Workflow management systems (WMS) are platforms that provide an infrastructure where workflows can be set-up, executed and monitored. Examples of WMS that have been utilized for metabolomics data analysis include Galaxy (Afgan *et al.*, 2016), Taverna (Wolstencroft *et al.*, 2013) and Konstanz Information Miner (KNIME) (Berthold *et al.*, 2009).

Galaxy was initially designed for the analysis of genomics data, but its use has since widely expanded across many bioinformatics disciplines. The Galaxy project has three stated project goals: accessibility, reproducibility and transparency. It has been designed for ease-of-use, so that researchers without programming experience are able to run workflows without difficulty. Galaxy enables the exact replication of analysis by capturing information on all modules and parameters used. Users are able to share Galaxy workflows, histories, datasets and pages (interactive, web-based documents describing analysis), either publicly or with specific individuals, facilitating transparent research. Examples of tools that utilise Galaxy for metabolomics data analysis include Workflow4metabolomics (Giacomoni *et al.*, 2015), Galaxy-M (Davidson *et al.*, 2016) and PhenoMeNal (see Relevant Websites section).

Workflow4metabolomics 3.0 is a web-based tool that includes workflows for the analysis of LC-MS, GC-MS and NMR data. Both MS workflows consist of three analysis stages: preprocessing, statistics and annotation. In the LC-MS workflow, data are preprocessing using XCMS (Smith *et al.*, 2006) (XCMS can additionally be used for preprocessing GC-MS data). Adducts and fragments are then annotated using CAMERA (Kuhl *et al.*, 2012). Alternatively users can upload a peak list of data that has already undergone preprocessing for further analysis. Modules for filtering and batch correction can then optionally be performed. The available statistical analysis modules include both univariate (Student's *t*-test, Wilcoxon *t*-test, ANOVA, Kruskal-Wallis test, etc.) and multivariate analysis (PCA, PLS, OPLS and PLS-DA). These modules are applicable to all data types. Metabolite databases can be searched for the annotation of LC-MS data, including the Human Metabolome Database (HMDB) (Wishart *et al.*, 2013), LipidMaps Structure database (LMSD) (Sud *et al.*, 2007) and MassBank (Horai *et al.*, 2010). For GC-MS data the Golm database (Kopka *et al.*, 2005) is available for annotation, or users can upload a local database. Bruker bucketing and integration, and normalization are modules supplied specifically for NMR data.

Galaxy-M is designed for the analysis of LC-MS and DIMS metabolomics data. It provides Galaxy installations of MI-PACK (Weber and Viant, 2010), XCMS and CAMERA. Preprocessing is performed by XCMS for raw LC-MS data and SIM-stitch for raw DIMS data. Following peak picking and alignment, data can be filtered by appearance in blank samples or by removing peaks that are not present in *x*-out-of-*n* samples. In preparation for statistical analysis missing values can be imputed, and data can be normalized and transformed. For statistical analysis PCA and univariate analysis are provided. If users have the commercial MATLAB toolbox, PLS-Toolbox available, this can be installed for chemometric multivariate analysis. Tools from MI-PACK are implemented for metabolite annotation.

The PhenoMeNal App library consists of ~50 tools or apps, not including individual algorithms as some tools such as OpenMS includes multiple submodules. These tools cover analysis of a wide variety of experimental data types including LC-MS, NMR and isotopically labelled data. As well as users being able to design their own workflows with these tools, there are pre-existing workflows also available. All the tools are available via Galaxy workflows and Jupyter libraries through the Cloud Research Environment (see Relevant Websites section).

Like Galaxy, KNIME is available as a graphical user interface (GUI) with a modular environment, constructed for usability. This allows for new tools or algorithms to be easily integrated into to workflow. For metabolomics, the workflows OpenMS (Aiche *et al.*, 2015) and MassCascade (Beiske *et al.*, 2014) are available as KNIME implementations. MassCascade is specifically designed for the analysis of LC-MSⁿ data. OpenMS was originally designed for proteomics data, but has been expanded to include metabolomics data analysis (Kenar *et al.*, 2014). Similarly to MassCascade, OpenMS focuses on high resolution LC-MS data. It provides an infrastructure for the rapid development of tools. There are over 185 different tools and ready-made workflows for common MS data processing tasks (Röst *et al.*, 2016; Sturm *et al.*, 2008).

The Taverna Workflow Management System provides a suite of tools enabling users to design, edit, execute and share workflows. Publicly available workflows can be found at <http://www.myexperiment.org/>. As of 5th October 2017, searching for 'metabolomics' returns 19 workflows.

There are also some widely used tools for metabolomics data analysis that do not use WMS. These include XCMS Online (Tautenhahn *et al.*, 2012), MetaboAnalyst (Xia *et al.*, 2015) and MZmine 2 (Pluskal *et al.*, 2010). XCMS Online is an online implementation of XCMS, that includes the entire data processing pipeline for MS data, from preprocessing to metabolite identification, statistical and pathway analysis. Users can also share their data and workflows either with collaborators or publicly. MetaboAnalyst 3.0 is a web server, that contains eight independent modules for analysis, including statistical, pathway, biomarker and enrichment analysis. The majority of modules require data to have been already preprocessed using other software. MZmine 2 has a GUI and provides a toolkit of modules for MS data analysis. It is able to profile, centroid and MSⁿ data. Modules include raw data processing, peak detection and visualization.

The majority of workflows for NMR processing are written in MATLAB, including FOCUS (Alonso *et al.*, 2014) and MatNMR (Van Beek, 2007). FOCUS is specifically designed for ^1H NMR data, whilst MatNMR is more general and can process both 1D and 2D NMR data.

Metabolomics Experiments Repositories and Metabolite Databases

Metabolomics data repositories can store both raw and annotated data. Raw data can be deposited as either property data formats (e.g. Thermo .RAW, Agilent .d) or open formats, such as .mzML, .mzXML, .mzData or .netCDF. Studies in repositories can also contain lists of identified metabolites, along with their relative abundance, concentration, what spices and type of samples they are found (Salek *et al.* (2017), Haug *et al.* (2017)).

Currently there are two global repositories for metabolomics data: the European Bioinformatics Institute's MetaboLights (Haug *et al.*, 2013) and United States Government National Institute of Health (NIH) Metabolomics Workbench (Sud *et al.*, 2016). These repositories accept submissions of all types of metabolomics data. Both repositories include multiple layers, storing data and metadata from metabolomics studies along with reference libraries of metabolites.

The two layers of the MetaboLights repository are: (i) the repository, where raw data, along with annotated metadata, are stored, and (ii) the reference layer that contains a library of reference compounds (Kale *et al.*, 2016). The repository and compound library can be queried by species, instrumental analysis technology, organism part (e.g. serum, leaf) or study validation status. As of 31st August 2017 there are 279 publicly available studies on MetaboLights and 24957 compounds, with 3171 including MS spectra and 407 NMR spectra.

There is a focus of the curation of data in MetaboLights (Salek *et al.*, 2013), and it therefore has more stringent submission guidelines that Metabolomics Workbench. Experimental metadata must be submitted to MetaboLights in the ISA-TAB format (Rocca-Serra *et al.*, 2010). In order to submit a study, users must first register. There are four stages of curation a submitted study must pass through until it becomes publicly available: submitted, in curation, in review and public. During the submitted stage the submission is not yet complete and mandatory metadata is missing. When all mandatory fields are complete curators will appraise the study, looking for any missing optional fields or errors not detected by automatic validation (Kale *et al.*, 2016). When curation is finished, the submitter is given a read-only reviewer link that can be shared with a journal or collaborators. The study will not become public until a submitter specified publication date has been reached. The analysis tools section of MetaboLights is currently under development. A new tool that plots study factor distribution using parallel coordinates is in the beta development stage.

Compared to MetaboLights, Metabolomics Workbench provides greater analysis capability. Data in publicly available studies can be normalized, clustered, and univariate and multivariate statistical analysis (PCA, linear discriminate analysis and PLS-DA) can be performed. For MS data additional analyses are provided: classification and feature analysis, pathway mapping, MS search for metabolite identification and comparison of metabolites between studies (meta-analysis). Users also have the option of performing statistical analysis on their own uploaded dataset.

There are 440 publicly available studies in Metabolomics Workbench as of October 2017. Bubble plots classifying studies by disease, sample source, species, pathway or metabolite class allow users to select a specific subset of studies. Metabolomics Workbench also provides a REST service where HTTP requests can be used to access data and metadata, including metabolite structures and experimental results. The Metabolomics Workbench Metabolite Database stores metabolite structures and annotations. It contains >61,000 entries collected from public databases including LIPID MAPS (Sud *et al.*, 2007), ChEBI (Hastings *et al.*, 2013), HMDB (Wishart *et al.*, 2013), BMRB (Ulrich *et al.*, 2008), PubChem (Kim *et al.*, 2016) and KEGG (Kanehisa *et al.*, 2012). Users can employ either text based search or m/z value from untargeted MS search to search for metabolites.

To submit a study to Metabolomics Workbench, users must register and obtain authorization. Once this has been completed (i) the study is registered, (ii) users must submit metadata in the specified format, either via an online form or a supplied excel template and (iii) raw data and/or supplementary material must be uploaded. It is encouraged that common names from the RefMet (A Reference list of Metabolite names) database are used for annotation.

The Global Natural Products Social Molecular Networking (GNPS) (Wang *et al.*, 2016) platform provides analysis and storage of natural products mass spectrometry data. Data are stored in the Mass Spectrometry Interactive Virtual Environment (MassIVE) repository, originally developed for proteomics studies (Perez-Riverol *et al.*, 2015). Compared to other repositories, GNPS is unique in providing continuous metabolite identification for MS/MS spectra, with data sets being reanalysed for new identifications once a month. Unlike other repositories GNPS has no requirements for reporting experimental metadata.

As well as the global metabolomics repositories, there are also a number of smaller, more specific repositories. These include Metabolomics Repository Bordeaux (MeRy-B) (Ferry-Dumazet *et al.*, 2011) and Metabolic Phenotype Database (MetaPhen) (Carroll *et al.*, 2010, 2015). MeRy-B is a repository specifically designed for plant ^1H NMR data, but also contains GC-MS data.

MetaPhen is a data repository that is available as part of MetabolomeExpress, a GC-MS metabolomics data analysis platform. It contains a variety of tools for the analysis of the publicly available data, including ResponseFinder, MetaAnalyser, PhenoMeter, and MSRI Library manager. Users can directly search for metabolites of interest using ResponseFinder, which includes search by species and organ/tissue/fluid type. MetaAnalyser performs meta-analysis by comparing the results of multiple experiments, plotting heatmaps of aligned and clustered data. PhenoMeter allows users to search the reference library via metabolite response patterns.

The MetabolomeXchange (see Relevant Websites section) consortium is based on the successful ProteomeXchange (Vizcaino *et al.*, 2014). It serves as a data aggregator, collecting studies from the MetaboLights, Metabolomics Workbench and MeRy-B repositories. Users have the option of subscribing in order to be notified about new datasets or updates to existing datasets.

OmicsDI is an open-source platform based at EMBL-EBI, that captures and integrates proteomics, genomics, metabolomics and transcriptomics datasets (Perez-Riverol *et al.*, 2016). For metabolomics the data repositories: MetaboLights, GNPS, MetaPhen and Metabolomics Workbench, are included. Since different repositories use different data models, ontologies, metadata representation and identifiers, OmicsDI uses metadata normalization and annotation expansion to harmonise searching across the various resources. Users are therefore able to discover multi-omics datasets that are linked to publications, as well as similar experimental datasets, suggested by metadata matching.

Most omics, including metabolomics, aim to follow FAIR principles by making data Findable, Accessible, Interoperable and Reusable (Van Rijswijk *et al.*, 2017; Wilkinson *et al.*, 2016). The data handling, analysis tools, data standards and resources discussed here should help users to adhere to FAIR principles.

Metabolomics Databases

The substantial growth of the field of Metabolomics during the past decade has instilled a need for sharable resources. Amongst other resources such as publicly available data processing softwares and workflow pipelines, data repositories, libraries and databases that store experiment and metabolite related information have been on the rise.

As detailed above, public repositories such as MetaboLights (Salek *et al.*, 2013) and the Metabolomics Workbench (Sud *et al.*, 2016) attempt to capture the data output of the entire workflow of a metabolomics experiment. These repositories store information on the experimental design, details on projects, experiments and samples as well as all the instrument generated raw data and the processed data of experiments. Commercial and public libraries and databases store exhaustive details about metabolites such as their physicochemical properties, related structures and links to other databases. Metabolite databases can be divided into three categories, a) metabolite mass spectral databases, b) Species-specific metabolite databases and c) metabolic pathway databases.

Metabolite mass spectral databases

Once the raw data is acquired from the instruments, data processing softwares are used to convert the raw data into a data matrix of molecular features with measures of their abundance in the samples. One of the last step in processing of raw data is to identify these molecular features as specific metabolites and this identification is performed using metabolite spectral libraries or databases. Metabolite spectral libraries store information on the physicochemical properties of metabolites such as molecular name, molecular weight, chemical formula, structure information, retention time, high-quality spectral data and links to external databases. Currently, there are several mass spectral libraries available for metabolomics.

NIST 17 ("NIST Standard Reference Database 1A v17", 2017) is the latest mass spectral database release of the National Institute of Standards and Technology, USA. NIST 17 contains an Electron Ionization (EI) mass spectral library of over 306,000 spectral belonging to over 267,000 unique compounds, a Gas Chromatography (GC) data library with over 404,000 retention index values for over 99,000 compounds covering both polar and nonpolar columns and an MS/MS library of over 650,000 spectra of over 14,000 compounds. It also includes the NIST MS search software for searching and identifying compounds and the automated mass spectral deconvolution and identification system (AMDIS) software used for deconvoluting chromatograms. The Wiley Registry of Mass Spectral Data is one of the largest collection of EI mass spectra. The 11th edition of Wiley ("Wiley Registry of Mass Spectral Data, 11th Edition", 2017) stores about 775,500 mass spectra for over 599,000 unique compounds and over 740,000 searchable structures. Massbank (Horai *et al.*, 2010) is a public repository that stores mass spectral data of high-resolution MSⁿ spectra. It is a distributed database in the sense where participating research groups are able to submit mass spectral information from local MassBank data servers. Users are then able to access spectral information from multiple groups through Massbank's web interface. Currently Massbank stores over 41,000 spectra. The METLIN Metabolite Database (Smith *et al.*, 2005) is a repository of MS and MS/MS data that stores details of more than 961,000 molecules. In addition, it also stores over 200,000 high-resolution in silico MS/MS spectra. The Golm Metabolome Database (Kopka *et al.*, 2005) stores information of EI mass spectra and associated retention indices of peaks quantified by GC-MS. The Madison-Qingdao Metabolomics Consortium Database (MMCD) (Cui *et al.*, 2008) is a library that stores details of metabolites analysed through NMR and MS. MMCD currently contains details of over 20,000 compounds. The Spectral Database for Organic Compounds (SDBS) ("SDBSWeb", 2017) stores mass spectrum information for around 34,000 organic compounds. The different types of spectra it stores are electron impact Mass spectrum (EI-MS), Fourier transform infrared spectrum (FT-IR), ¹H nuclear magnetic resonance (NMR) spectrum, ¹³C NMR spectrum, laser Raman spectrum, and electron spin resonance (ESR) spectrum.

Species-specific metabolite databases

The Human Metabolome Database (HMDB) (Wishart *et al.*, 2013) is a public repository containing detailed information about metabolites found in the human body. HMDB currently stores more than 330,000 spectra, stores details of more than 114,000 metabolites and has information derived from 17 different biofluids. HMDB is designed to contain or link externally to chemical data, clinical data and biochemistry data. The KNApSACk family databases (Nakamura *et al.*, 2014) are an integrated metabolite-plant species databases for multifaceted plant research. It currently has over 111,000 species-metabolite relationships

encompassing 22,350 species and over 50,000 metabolites. The RIKEN MSn spectral database for phytochemicals (ReSpect) (Sawada *et al.*, 2012) stores plant-specific MS/MS spectral data. ReSpect currently stores around 9000 spectral records.

Metabolic pathway databases

Metabolic pathway databases store information on biochemical pathways and the various enzyme reactions along the pathways. In addition, they store data on genes, gene products and compounds that are part of these reactions along with links to external databases on further information on the participants.

The KEGG Pathway Database (Arakawa *et al.*, 2005) is a collection of manually drawn pathway maps detailing metabolic reactions and the resulting molecular interactions. KEGG is a collection of fifteen databases that are divided into four categories: Systems Information, Genomic Information, Chemical Information and Health Information. The KEGG PATHWAY database that stores metabolic maps currently contains 519 pathway maps totalling to more than 530,000 different pathways and are detailed under the Systems Information category.

The MetaCyc database (Caspi *et al.*, 2016) stores information on metabolic pathways and enzymes that are experimentally determined and are curated from scientific literature. The database stores pathways for a wide range of organisms and currently contains 2572 pathways from 2844 different organisms. MetaCyc is widely used as the online encyclopaedia of metabolism, used to predict metabolic pathways in sequenced genomes and also used in metabolic engineering.

BioCyc (Caspi *et al.*, 2016) is a collection of pathway and genome databases for organisms whose genomes have been sequenced. Each of these databases contain the full genome, predicted metabolic network and additional information on enzymes, reactions and predicted operons. The BioCyc database also includes the Pathway Tools Software, a symbolic systems biology software system. The tools allow the creation of new pathway and genome database, has options to query visualize and analyze the databases, supports the development of metabolic flux models and allows the interactive editing of the underlying genome databases.

The Reactome pathway knowledgebase (Fabregat *et al.*, 2016) is a manually curated database of pathways. Reactome's pathways are peer reviewed where experts and the Reactome editorial staff work together in finalising the contents and are cross-referenced to other bioinformatics resources such as NCBI Gene, Ensembl, UniProt, KEGG Compound, PubMed and Gene Ontology. Reactome also provides pathway analysis tools that performs ID mapping, pathway assignment and enrichment analysis.

The Plant Metabolic Network (Kruger and Ratcliffe, 2012) is a collection of one multi-species reference database called PlantCyc and 76 species/taxon-specific databases. PlantCyc currently contains more than 1000 pathways and compounds from over 350 plant species. The pathways have been manually curated from literature and also includes hypothetical pathways and computationally predicted pathways that are published in peer-reviewed journals.

Statistical Analysis and Functional Interpretation

The widespread adoption of metabolomics in basic research and large-scale clinical investigations have resulted in the generation of increasingly complex data sets. Making sense of such high-dimensional data requires a good understanding of modern statistical and functional analysis approaches. In the previous sections, we have described experimental design, raw data processing, databases and data repositories. In this section, we will introduce the key concepts and main approaches commonly used in statistical analysis and functional interpretation of metabolomics data.

Data Input

The standard input for statistical data analysis is a high-dimensional data matrix with variables in rows and samples in columns (or its transposed format). Such data can be produced from most metabolomics data processing tools (Smith *et al.*, 2006; Pluskal *et al.*, 2010; Hao *et al.*, 2012; Ravanbakhsh *et al.*, 2015; Tautenhahn *et al.*, 2012). For data from targeted metabolomics, variables are compound IDs; and for untargeted metabolomics data, variables are typically peak locations or spectral bins. Data dimension refers to the number of variables in the matrix, which can vary from 100s to 1000s, with untargeted metabolomics data typically containing more variables. Samples are biological replicates associated with class labels or other metadata reflecting experimental designs. Quality control (QC) samples are often included for quality assurance during analysis.

Data Cleaning

The first, but often neglected step in metabolomics data analysis, is data cleaning or data filtering. Data from comprehensive metabolomics experiments usually contains a large amount of noise that can negatively impact downstream data analysis. Proper data cleaning will not only improve data quality, but also enhance statistical power and reduce false positives (Bourgon *et al.*, 2010). To identify low-quality data for cleaning, several empirical rules are frequently used, including a) hard-to-measure variables such as those close to baseline noises or showing large variations in QC samples or technical replicates; b) uninformative variables that are relatively constant throughout the experimental conditions based on their inter-quantile ranges, standard deviations, etc.; and c) outlier samples with extreme deviations from the main clusters. As outliers could be generated by unexpected but biologically important processes, their exclusion should be better justified with technical or experimental reasons, in addition to visual inspections.

Data Normalization

This step aims to reduce overall undesirable variations in the data to facilitate biologically and statistically meaningful comparisons. As different analytical platforms are currently used in metabolomics, a wide variety of normalization methods have been developed over the past decade (Van Den Berg *et al.*, 2006). Some of them are generic statistical methods such as log transformation and auto-scaling; while others are biologically motivated approaches such as the probabilistic quotient normalization that was initially developed to address the dilution effects in NMR-based metabolomics on urine samples (Dieterle *et al.*, 2006). The MetaboAnalyst web server currently supports a comprehensive selection of 13 different normalization methods (Xia and Wishart, 2016; Xia *et al.*, 2015). A new web-based tool, NOREVA allows users to evaluate the effects of 24 normalization methods for MS-based metabolomics data (Li *et al.*, 2017). No consensus has been reached on the best normalization methods for metabolomics data. In general, we should start with simple methods with minimal transformations of the original data. More complicated approaches may be necessary for complex data from large-scale studies.

Unsupervised Methods

After cleaning and normalization, the data is now suitable for statistical analysis. The first step is to examine if there are any interesting patterns or trends in the data, and whether these patterns are likely to be technical artifacts or biological effects. Unsupervised methods are designed for such tasks. As the name suggests, these type of methods use only the data (X) itself without considering the class labels (Y) during the analysis. Here we will introduce two main approaches - principal component analysis (PCA) and cluster analysis. PCA is the most widely used method in metabolomics for data overview and pattern discovery. PCA is a dimensionality reduction method which aims to capture the most variation of the data in the top few principal components (PCs) created by linear combinations of the original variables. These PCs can be visualized as a 2D or 3D scatter plot (also known as scores plot) to see if there are major grouping patterns among samples. The corresponding loading plots can be used to identify the variables that contribute to such separation patterns. In addition to PCA, cluster analysis (or clustering) is also widely used to help reveal inherent patterns in data. These type of methods work by first computing similarities or distances among samples or variables, and then assigning them to different clusters so that items within the same cluster will be more similar to each other than they are to those outside the cluster. The result of the cluster analysis can often be visualized in heatmaps, a technique that uses color gradients to represent data values. In clustered heatmaps, main patterns or clusters will stand out as distinct “patches” displaying more homogenous colors as compared to the surrounding background. Several clustering methods are often used in metabolomics data analysis, including hierarchical clustering, k-means clustering, self-organizing maps (SOM), etc. More detailed introductions on these clustering algorithms can be found in this excellent review paper (Ren *et al.*, 2015).

Supervised Approaches

These methods are used to formally evaluate the associations between the variations in data (X) and class labels (Y). Metabolomics has a long history of using various multivariate statistical methods and machine learning algorithms for classification and regression analysis (Broadhurst and Kell, 2006). Here we will introduce a widely used multivariate statistical method - partial least squares discriminant analysis (PLS-DA), and then briefly comment on a powerful machine learning method - support vector machine (SVM). Similar to PCA, PLS-DA is a dimensionality reduction method by creating latent variables (LV) through linear combinations of original variables in the data (X). These LVs are computed by referring to the class labels (Y) to maximize the explained variance in Y in the top few LVs. The PLS-DA results can also be visualized in scores and loadings plots for interpretation. Due to the supervised nature of the method, the performance (as seen from the separation patterns in the scores plot) is usually much better compared to PCA. However, this may be misleading. It has been shown that PLS-DA can produce good separation patterns even with randomly generated group labels (Westerhuis *et al.*, 2008). Therefore, PLS-DA is very susceptible to “over-fitting” – a well-known issue associated with using supervised methods. To address this issue, results from PLS-DA need to be evaluated using both cross-validations (or double cross-validation if sample size is sufficient) and permutation tests. In cross-validation, samples are divided into trainings and testing groups, and the training group is then used to build PLS-DA models whose performances are evaluated using the test group. The permutation tests evaluate whether similar performances can be obtained using models created based on randomly labelled data, compared to the model created using the original group labels. SVM is a powerful machine learning method commonly employed in metabolomics data analysis (Mahadevan *et al.*, 2008; Heinemann *et al.*, 2014). The algorithm uses kernel functions to map the original data to a higher dimensional feature space and separating it by means of a maximum margin hyperplane (Noble, 2006). Compared to PLS-DA, SVM generally gives better performance and is less susceptible to over-fitting issues. However, the interpretation of the underlying models is very complicated and much less intuitive (i.e. black box).

Functional Analysis

Direct functional analysis is generally limited to data from targeted metabolomics. For untargeted metabolomics, researchers need to first identify significant peaks or spectral bins, and then annotate the underlying metabolites before performing functional analysis. A key concept in functional analysis is that the unit of analysis has been shifted from individual metabolites to a set of

metabolites (termed a metabolite set). Metabolite sets are groups of metabolites that share a particular property, for instance, a group of metabolites involved in the same pathways, altered in the same diseases, or associated with the same genetic or epigenetic variations. If particular metabolite sets are significantly enriched in the data, the corresponding shared properties (i.e. pathways) are considered to be relevant to the experimental conditions of interest. Many different algorithms have been developed to perform enrichment tests. For instance, the over-representation analysis (ORA) methods accept a list of significant metabolites and evaluate whether certain metabolite sets appear more often than expected by chance, using Fisher's Exact tests or hyper-geometrics tests. For instance, the Metabolites Biological Role (MBROLE) web application uses hypergeometric tests to perform comprehensive functional enrichment analysis (Lopez-Ibanez *et al.*, 2016). The quantitative enrichment analysis (QEA, also known as functional class scoring) is a relatively new approach for enrichment tests. Compared to ORA, QEA utilizes the complete data to calculate enrichment scores for individual metabolite sets, and is more sensitive for detecting subtle but consistent changes. A wide range of QEA tools are available, showing many statistical and practice differences (Tarca *et al.*, 2013). For instance, the QEA module in MSEA (now part of MetaboAnalyst) is based on the *globaltest* R package (Goeman *et al.*, 2004; Xia and Wishart, 2010). Detailed discussions on the performance characteristics and statistical properties for these different approaches can be found in a recent excellent review paper (De Leeuw *et al.*, 2016). Due to the inherent analytical bias in metabolomics platforms and the limited metabolome coverage, results from current enrichment analysis should be interpreted with caution.

From Peaks to Biological Activities

The increased application of sensitive and high-resolution MS-based metabolomics makes it possible to perform functional analysis directly from raw spectral peaks. The key idea is to redefine the metabolite sets using the spectral features of the corresponding metabolites, and then evaluate the enrichment of these chemical signals directly from spectral peaks without formal compound identification. The *mummichog* method is the first bioinformatics approach in this direction (Li *et al.*, 2013b). The algorithm accepts two lists of spectral peaks - a significant peak list obtained through statistical analysis (i.e. *t*-tests) and a reference peak list (i.e. all peaks detected in the experiment). It computes matching scores for different pathways from the significant peak lists, and enrichment *p*-values are calculated by comparing the scores to those obtained using peak lists of the same size, but randomly drawn from the reference peaks. Further development in this direction holds the potential to greatly facilitate metabolomics functional analysis.

Due to space limitations, this section covers only those most common methods used in statistical and functional analysis of metabolomics data, while ignoring very basic methods such as univariate statistics (i.e. *t*-tests and ANOVA) for the detection of significant variables. For the same reason, we have also intentionally omitted some advanced topics including biomarker analysis and time-series data analysis. Readers are referred to several comprehensive reviews and tutorials for more detailed discussions on these topics (Broadhurst and Kell, 2006; Smilde *et al.*, 2010; Xia *et al.*, 2013; Worley and Powers, 2013; Ren *et al.*, 2015). Most of these concepts and procedures have been implemented and are available thorough easy to use and freely accessible XCMS online and MetaboAnalyst web applications.

Acknowledgements

SD and UR thank for support from Metabolomics Australia Pty Ltd which is funded through the National Collaborative Research Infrastructure Strategy (NCRIS), 5.1 Biomolecular Platforms and Informatics with co-investments from the University of Melbourne. U.R. is funded through an Australian Research Council Future Fellowship program. JX would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs (CRC) program.

See also: Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Investigating Metabolic Pathways and Networks. Mass Spectrometry-Based Metabolomic Analysis. Metabolic Models. Metabolic Profiling. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Standards and Models for Biological Data: FGED and HUP0. Two Decades of Biological Pathway Databases: Results and Challenges

References

- Afgan, E., Baker, D., Van Den Beek, M., *et al.*, 2016. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.* 44, W3–W10.
- Aiche, S., Sachsenberg, T., Kenar, E., *et al.*, 2015. Workflows for automated downstream data analysis and visualization in large-scale computational mass spectrometry. *Proteomics* 15, 1443–1447.
- Allen, F., Pon, A., Wilson, M., Greiner, R., Wishart, D., 2014. CFM-ID: A web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra. *Nucleic Acids Res.* 42, 94–99.
- Alonso, A., Rodríguez, M.A., Vinaixa, M., *et al.*, 2014. Focus: A robust workflow for one-dimensional NMR spectral analysis. *Anal. Chem.* 86, 1160–1169.
- Arakawa, K., Kono, N., Yamada, Y., Mori, H., Tomita, M., 2005. KEGG-based pathway visualization tool for complex omics data. *In Silico Biol.* 5, 419–423.

- Baker, M., 2016. Statisticians issue warning over misuse of P values. *Nature* 531, 151.
- Beisken, S., Earll, M., Portwood, D., Seymour, M., Steinbeck, C., 2014. MassCascade: Visual programming for LC-MS data processing in metabolomics. *Mol. Inform.* 33, 307–310.
- Beisken, S., Eiden, M., Salek, R.M., 2015. Getting the right answers: Understanding metabolomics challenges. *Expert Rev. Mol. Diagn.* 15, 97–109.
- Berthold, M.R., Cebon, N., Dill, F., *et al.*, 2009. KNIME – The Konstanz Information Miner: Version 2.0 and Beyond. *SIGKDD Explor. Newsl.* 11, 26–31.
- Bingol, K., Bruschweiler-Li, L., Li, D., *et al.*, 2016. Emerging new strategies for successful metabolite identification in metabolomics. *Bioanalysis* 8, 557–573.
- Bocker, S., Letzel, M.C., Liptak, Z., Pervukhin, A., 2009. SIRIUS: Decomposing isotope patterns for metabolite identification. *Bioinformatics* 25, 218–224.
- Bourgon, R., Gentleman, R., Huber, W., 2010. Independent filtering increases detection power for high-throughput experiments. *Proc. Natl. Acad. Sci. USA* 107, 9546–9551.
- Broadhurst, D.I., Kell, D.B., 2006. Statistical strategies for avoiding false discoveries in metabolomics and related experiments. *Metabolomics* 2, 171–196.
- Broeckling, C.D., Reddy, I.R., Duran, A.L., *et al.*, 2006. MET-IDEA: Data Extraction Tool for Mass Spectrometry-Based Metabolomics. *Anal. Chem.* 78, 4334–4341.
- Brown, M., Wedge, D.C., Goodacre, R., *et al.*, 2011. Automated workflows for accurate mass-based putative metabolite identification in LC/MS-derived metabolomic datasets. *Bioinformatics* 27, 1108–1112.
- Carroll, A.J., Badger, M.R., Harvey Millar, A., 2010. The MetabolomeExpress Project: Enabling web-based processing, analysis and transparent dissemination of GC/MS metabolomics datasets. *BMC Bioinformatics* 11, 376.
- Carroll, A.J., Zhang, P., Whitehead, L., *et al.*, 2015. PhenoMeter: A metabolome database search tool using statistical similarity matching of metabolic phenotypes for high-confidence detection of functional links. *Frontiers Bioeng. Biotechnol.* 3, 106.
- Caspi, R., Billington, R., Ferrer, L., *et al.*, 2016. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res* 44, D471–D480.
- Chambers, M.C., Maclean, B., Burke, R., *et al.*, 2012. A cross-platform toolkit for mass spectrometry and proteomics. *Nat. Biotechnol.* 30, 918–920.
- Chokkathukalam, A., Jankevics, A., Creek, D.J., *et al.*, 2013. mzMatch-ISO: An R tool for the annotation and relative quantification of isotope-labelled mass spectrometry data. *Bioinformatics* 29, 281–283.
- Creek, D.J., Dunn, W.B., Fiehn, O., *et al.*, 2014. Metabolite identification: Are you sure? And how do your peers gauge your confidence? *Metabolomics* 10, 350–353.
- Cui, Q., Lewis, I.A., Hegeman, A.D., *et al.*, 2008. Metabolite identification via the Madison Metabolomics Consortium Database. *Nat. Biotechnol.* 26, 162–164.
- Curran-Everett, D., 2000. Multiple comparisons: Philosophies and illustrations. *Am. J. Physiol. Regul. Integr. Comp. Physiol.* 279, R1–R8.
- Daly, R., Rogers, S., Wandy, J., *et al.*, 2014. MetAssign: Probabilistic annotation of metabolites from LC–MS data using a Bayesian clustering approach. *Bioinformatics* 30, 2764–2771.
- Davidson, R.L., Weber, R.J.M., Liu, H., Sharma-Oates, A., Viant, M.R., 2016. Galaxy-M: A Galaxy workflow for processing and analyzing direct infusion and liquid chromatography mass spectrometry-based metabolomics data. *Gigascience* 5, 10.
- De Leeuw, C.A., Neale, B.M., Heskes, T., Posthuma, D., 2016. The statistical properties of gene-set analysis. *Nat. Rev. Genet.* 17, 353–364.
- Dieterle, F., Ross, A., Schlotterbeck, G., Senn, H., 2006. Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. Application in ¹H NMR metabolomics. *Anal. Chem.* 78, 4281–4290.
- Draper, J., Enot, D.P., Parker, D., *et al.*, 2009. Metabolite signal identification in accurate mass metabolomics data with MZedDB, an interactive *m/z* annotation tool utilising predicted ionisation behaviour 'rules'. *BMC Bioinformatics* 10, 227.
- Dührkop, K., Shen, H., Meusel, M., Rousu, J., Böcker, S., 2015. Searching molecular structure databases with tandem mass spectra using CSI: Fingerid. *Proc. Natl. Acad. Sci. USA* 112, 12580–12585.
- Fabregat, A., Sidiropoulos, K., Garapati, P., *et al.*, 2016. The Reactome pathway Knowledgebase. *Nucleic Acids Res.* 44, D481–D487.
- Ferry-Dumazet, H., Gil, L., Deborde, C., *et al.*, 2011. MeRy-B: A web knowledgebase for the storage, visualization, analysis and annotation of plant NMR metabolomic profiles. *BMC Plant Biol.* 11.
- Giacomoni, F., Le Corguillé, G., Monsoor, M., *et al.*, 2015. Workflow4Metabolomics: A collaborative research infrastructure for computational metabolomics. *Bioinformatics* 31, 1493–1495.
- Goeman, J.J., Van De Geer, S.A., De Kort, F., Van Houwelingen, H.C., 2004. A global test for groups of genes: Testing association with a clinical outcome. *Bioinformatics* 20, 93–99.
- Gómez, J., Brezmes, J., Mallol, R., *et al.*, 2014. Dolphin: A tool for automatic targeted metabolite profiling using 1D and 2D ¹H NMR data. *Anal. Bioanal. Chem.* 406, 7967–7976.
- Hao, J., Astle, W., De Iorio, M., Ebbels, T.M.D., 2012. Batman-an R package for the automated quantification of metabolites from nuclear magnetic resonance spectra using a bayesian model. *Bioinformatics* 28, 2088–2090.
- Hastings, J., De Matos, P., Dekker, A., *et al.*, 2013. The ChEBI reference database and ontology for biologically relevant chemistry: Enhancements for 2013. *Nucleic Acids Res.* 41, D456–D463.
- Haug, K., Salek, R.M., Conesa, P., *et al.*, 2013. MetaboLights – an open-access general-purpose repository for metabolomics studies and associated meta-data. *Nucleic Acids Res.* 41, D781–D786.
- Haug, K., Salek, R.M., Steinbeck, C., 2017. Global open data management in metabolomics. *Curr. Opin. Chem. Biol.* 36, 58–63.
- Heinemann, J., Mazurie, A., Tokmina-Lukaszewska, M., Beilman, G.J., Bothner, B., 2014. Application of support vector machines to metabolomics experiments with limited replicates. *Metabolomics* 10, 1121–1128.
- Hill, C., Roessner, U., 2013. Plant metabolomics using GC–MS. In: Weckwerth, W. (Ed.), *GC–MS Metabolomics Handbook*. Springer.
- Hill, C., Roessner, U., 2014. Advances in high-throughput untargeted LC-MS analysis for plant metabolomics. *Advanced LC-MS Applications for Metabolomics*. UK: Future Science Group.
- Hiller, K., Hangebrauk, J., Jäger, C., *et al.*, 2009. Metabolite detector: Comprehensive analysis tool for targeted and nontargeted GC/MS based metabolome analysis. *Anal. Chem.* 81, 3429–3439.
- Hoffmann, N., Stoye, J., 2012. Generic software frameworks for GC–MS based metabolomics. *Metabolomics*. InTech.
- Horai, H., Arita, M., Kanaya, S., *et al.*, 2010. MassBank: A public repository for sharing mass spectral data for life sciences. *J. Mass Spectrom.* 45, 703–714.
- Huber, W., 2016. A clash of cultures in discussions of the P value. *Nat. Methods* 13, 607.
- Hunter, A., Dayalan, S., De Souza, D., *et al.*, 2017. MASTR-MS: A web-based collaborative laboratory information management system (LIMS) for metabolomics. *Metabolomics* 13, 14.
- Kale, N.S., Haug, K., Conesa, P., *et al.*, 2016. MetaboLights: An Open-Access Database Repository for Metabolomics Data. *Curr. Protoc. Bioinformatics*, 53. pp. 14.13.1–14.13.18.
- Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., Tanabe, M., 2012. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res.* 40, D109–D114.
- Kenar, E., Franken, H., Forcisi, S., *et al.*, 2014. Automated label-free quantification of metabolites from liquid chromatography–mass spectrometry data. *Mol. Cell. Proteomics* 13, 348–359.
- Kessler, N., Neuweiger, H., Bonte, A., *et al.*, 2013. MeltDB 2.0-advances of the metabolomics software system. *Bioinformatics* 29, 2452–2459.
- Kim, S., Thiessen, P.A., Bolton, E.E., *et al.*, 2016. PubChem Substance and Compound databases. *Nucleic Acids Res.* 44, D1202–D1213.
- Kopka, J., Schauer, N., Krueger, S., *et al.*, 2005. GMD@CSB.DB: The Golm metabolome database. *Bioinformatics* 21, 1635–1638.
- Kruger, N.J., Ratcliffe, R.G., 2012. Pathways and fluxes: Exploring the plant metabolic network. *J. Exp. Bot.* 63, 2243–2246.

- Kuhl, C., Tautenhahn, R., Böttcher, C., Larson, T.R., Neumann, S., 2012. CAMERA: An integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. *Anal. Chem.* 84, 283–289.
- Lewis, I.A., Schommer, S.C., Markley, J.L., 2009. rNMR: Open source software for identifying and quantifying metabolites in NMR spectra. *Magn. Reson. Chem.* 47, Li, B., Tang, J., Yang, Q., *et al.*, 2017. NOREVA: Normalization and evaluation of MS-based metabolomics data. *Nucleic Acids Res.*
- Li, L., Li, R., Zhou, J., *et al.*, 2013a. MyCompoundID: using an evidence-based metabolome library for metabolite identification. *Anal. Chem.* 85, 3401–3408.
- Li, S., Park, Y., Duraisingham, S., *et al.*, 2013b. Predicting network activity from high throughput metabolomics. *PLOS Comput Biol.* 9, e1003123.
- Lommen, A., Kools, H.J., 2012. MetAlign 3.0: Performance enhancement by efficient use of advances in computer hardware. *Metabolomics* 8, 719–726.
- Lopez-Ibanez, J., Pazos, F., Chagoyen, M., 2016. MBROLE 2.0-functional enrichment of chemical compounds. *Nucleic Acids Res.* 44, W201–W204.
- Lynn, K.S., Cheng, M.L., Chen, Y.R., *et al.*, 2015. Metabolite identification for mass spectrometry-based metabolomics using multiple types of correlated ion information. *Anal. Chem.* 87, 2143–2151.
- Mahadevan, S., Shah, S.L., Marrie, T.J., Slupsky, C.M., 2008. Analysis of metabolomic data using support vector machines. *Anal. Chem.* 80, 7562–7570.
- Meyer, M.R., Peters, F.T., Maurer, H.H., 2010. Automated mass spectral deconvolution and identification system for GC–MS screening for drugs, poisons, and metabolites in urine. *Clin. Chem.* 56, 575–584.
- Nakamura, Y., Afendi, F.M., Parvin, A.K., *et al.*, 2014. KnapSack Metabolite Activity Database for retrieving the relationships between metabolites and biological activities. *Plant Cell Physiol.* 55, e7.
- Nicolè, F., Guittion, Y., Courtois, E.A., *et al.*, 2012. MSeasy: Unsupervised and Untargeted GC–MS data processing. *Bioinformatics* 28, 2278–2280.
- NIST Standard Reference Database 1A v17, 2017. Available at: <https://www.nist.gov/srd/nist-standard-reference-database-1a-v17>. [accessed 20.10.17].
- nmrML, 2017. Available at: <http://nmrml.org/>. [accessed 20.10.2017].
- Noble, W.S., 2006. What is a support vector machine? *Nat. Biotechnology* 24, 1565–1567.
- O'sullivan, A., Avizonis, D., Bruce German, J., Slupsky, C.M., 2011. Software Tools for NMR Metabolomics. *Encyclopedia of Magnetic Resonance*.
- Pahler, A., Brink, A., 2013. Software aided approaches to structure-based metabolite identification in drug discovery and development. *Drug Discov. Today Technol.* 10, e207–e217.
- Perez-Riverol, Y., Alpi, E., Wang, R., Hermjakob, H., Vizcaino, J.A., 2015. Making proteomics data accessible and reusable: Current state of proteomics databases and repositories. *Proteomics* 15, 930–949.
- Perez-Riverol, Y., Bai, M., Leprevost, F., *et al.*, 2016. Omics Discovery Index – Discovering and Linking Public Omics Datasets. *bioRxiv*.
- Pluskal, T., Castillo, S., Villar-Briones, A., Oresic, M., 2010. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics* 11, 395.
- Ravanbakhsh, S., Liu, P., Bjordahl, T.C., *et al.*, 2015. Accurate, fully-automated NMR spectral profiling for metabolomics. *PLOS One* 10, e0124219.
- Ren, S., Hinzman, A.A., Kang, E.L., Szczesniak, R.D., Lu, L.J., 2015. Computational and statistical analysis of metabolomics data. *Metabolomics* 11, 1492–1513.
- Ridder, L., Van Der Hooft, J.J.J., Verhoeven, S., *et al.*, 2013. Automatic chemical structure annotation of an LC–MSn based metabolic profile from green tea. *Anal. Chem.* 85, 6033–6040.
- Rocca-Serra, P., Brandizi, M., Maguire, E., *et al.*, 2010. ISA software suite: Supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics* 26, 2354–2356.
- Röst, H.L., Sachsenberg, T., Aiche, S., *et al.*, 2016. OpenMS: A flexible open-source software platform for mass spectrometry data analysis. *Nat. Methods* 13, 741–748.
- Ruttikies, C., Schymanski, E.L., Wolf, S., Hollender, J., Neumann, S., 2016. MetFrag relaunched: Incorporating strategies beyond in silico fragmentation. *J. Cheminform.* 8, 3.
- Salek, R.M., Conesa, P., Cochrane, K., *et al.*, 2017. Automated assembly of species metabolomes through data submission into a public repository. *Gigascience* 6, 1–4.
- Salek, R.M., Haug, K., Conesa, P., *et al.*, 2013. The MetaboLights repository: Curation challenges in metabolomics. *Database* 2013, bat029.
- Salek, R.M., Neumann, S., Schober, D., *et al.*, 2015. COordination of Standards in Metabolomics (COSMOS): Facilitating integrated metabolomics data access. *Metabolomics*, 11. . pp. 1587–1597.
- Sawada, Y., Nakabayashi, R., Yamada, Y., *et al.*, 2012. RIKEN tandem mass spectral database (ReSpect) for phytochemicals: A plant-specific MS/MS-based data resource and database. *Phytochemistry* 82, 38–45.
- Schelltema, R.A., Jankevics, A., Jansen, R.C., Swertz, M.A., Breitling, R., 2011. PeakML/mzMatch: A file format, Java library, R library, and tool-chain for mass spectrometry data analysis. *Anal. Chem.* 83, 2786–2793.
- Scholz, M., Fiehn, O., 2007. SetupX – A public study design database for metabolomic projects. *Pac. Symp. Biocomput.* 169–180.
- SDBSWeb, 2017. Spectral Database for Organic Compounds (SDBS). Available at: <http://sdb.sdb.aist.go.jp/> [accessed 20.10.17].
- Sesame LIMS, 2017. Available at: http://www.sesame.wisc.edu/sesame_home.html. [accessed 20.10.17].
- Shen, H., Duhrkop, K., Bocker, S., Rousu, J., 2014. Metabolite identification through multiple kernel learning on fragmentation trees. *Bioinformatics* 30, i157–i164.
- Skogerson, K., Wohlgemuth, G., Barupal, D.K., Fiehn, O., 2011. The volatile compound BinBase mass spectral database. *BMC Bioinformatics* 12, 321.
- Smilde, A.K., Westerhuis, J.A., Hoefsloot, H.C., *et al.*, 2010. Dynamic metabolomic data analysis: A tutorial review. *Metabolomics* 6, 3–17.
- Smith, C.A., O'maille, G., Want, E.J., *et al.*, 2005. METLIN: A metabolite mass spectral database. *Ther. Drug Monit.* 27, 747–751.
- Smith, C.A., Want, E.J., O'maille, G., Abagyan, R., Siuzdak, G., 2006. XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal. Chem.* 78, 779–787.
- Spicer, R., Salek, R.M., Moreno, P., Cañueto, D., Steinbeck, C., 2017. Navigating freely-available software tools for metabolomics analysis. *Metabolomics* 13, 106.
- Sturm, M., Bertsch, A., Gröpl, C., *et al.*, 2008. OpenMS – an open-source software framework for mass spectrometry. *BMC Bioinformatics* 9, 163.
- Sud, M., Fahy, E., Cotter, D., *et al.*, 2016. Metabolomics Workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools. *Nucleic Acids Res.* 44, D463–D470.
- Sud, M., Fahy, E., Cotter, D., *et al.*, 2007. LMSD: Lipid MAPS structure database. *Nucleic Acids Res.*, 35. pp. D527–D532.
- Sumner, L.W., Amberg, A., Barrett, D., *et al.*, 2007. Proposed minimum reporting standards for chemical analysis Chemical Analysis Working Group (CAWG) Metabolomics Standards Initiative (MSI). *Metabolomics* 3, 211–221.
- Sumner, L.W., Lei, Z., Nikolau, B.J., *et al.*, 2014. Proposed quantitative and alphanumeric metabolite identification metrics. *Metabolomics* 10, 1047–1049.
- Tarca, A.L., Bhatti, G., Romero, R., 2013. A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity. *PLOS One* 8, e79217.
- Tautenhahn, R., Patti, G.J., Rinehart, D., Siuzdak, G., 2012. XCMS online: A web-based platform to process untargeted metabolomic data. *Anal. Chem.* 84, 5035–5039.
- Tsugawa, H., Cajka, T., Kind, T., *et al.*, 2015. MS-DIAL: Data-independent MS/MS deconvolution for comprehensive metabolome analysis. *Nat. Methods*. 12.
- Ulrich, E.L., Akutsu, H., Doreleijers, J.F., *et al.*, 2008. BioMagResBank. *Nucleic Acids Res.* 36, 402–408.
- Van Beek, J.D., 2007. matNMR: A flexible toolbox for processing, analyzing and visualizing magnetic resonance data in Matlab. *J. Magn. Reson.* 187, 19–26.
- Van Den Berg, R.A., Hoefsloot, H.C., Westerhuis, J.A., Smilde, A.K., Van Der Werf, M.J., 2006. Centering, scaling, and transformations: Improving the biological information content of metabolomics data. *BMC Genomics* 7, 142.
- Van Rijswijk, M., Beirnaert, C., Caron, C., *et al.*, 2017. The future of metabolomics in ELIXIR. *F1000Res.* 6.
- Vizcaino, J.A., Deutsch, E.W., Wang, R., *et al.*, 2014. ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotechnol.* 32, 223–226.
- Wang, M., Carver, J.J., Phelan, V.V., *et al.*, 2016. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. *Nat. Biotechnol.* 34, 828–837.

- Weber, R.J.M., Lawson, T.N., Salek, R.M., *et al.*, 2016. Computational tools and workflows in metabolomics: An international survey highlights the opportunity for harmonisation through Galaxy. *Metabolomics* 13, 12.
- Weber, R.J.M., Viant, M.R., 2010. MI-Pack: Increased confidence of metabolite identification in mass spectra by integrating accurate masses and metabolic pathways. *Chemometrics Intellig. Lab. Syst.* 104, 75–82.
- Wehrens, R., Weingart, G., Mattivi, F., 2014. metaMS: An open-source pipeline for GC–MS-based untargeted metabolomics. *J. Chromatogr. B* 966, 109–116.
- Westerhuis, J.A., Hoefsloot, H.C.J., Smit, S., *et al.*, 2008. Assessment of PLS-DA cross validation. *Metabolomics* 4, 81–89.
- Wiley Registry of Mass Spectral Data, 2017. eleventh ed. Available at: <http://au.wiley.com/WileyCDA/WileyTitle/productCd-1119171016.html>. [accessed 20.10.2017].
- Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J.J., *et al.*, 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data*, 3. . p. 160018.
- Wishart, D.S., Jewison, T., Guo, A.C., *et al.*, 2013. HMDB 3.0 – The Human Metabolome Database in 2013. *Nucleic Acids Res.* 41, 801–807.
- Wolstencroft, K., Haines, R., Fellows, D., *et al.*, 2013. The Taverna workflow suite: Designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res.* 41, W557–W561.
- Worley, B., Powers, R., 2013. Multivariate analysis in metabolomics. *Curr. Metabolomics* 1, 92–107.
- Xia, J., Broadhurst, D.I., Wilson, M., Wishart, D.S., 2013. Translational biomarker discovery in clinical metabolomics: An introductory tutorial. *Metabolomics* 9, 280–299.
- Xia, J., Sinelnikov, I.V., Han, B., Wishart, D.S., 2015. MetaboAnalyst 3.0 – making metabolomics more meaningful. *Nucleic Acids Res.* 43, W251–W257.
- Xia, J., Wishart, D.S., 2010. MSEA: A web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic Acids Res.* 38, W71–W77.
- Xia, J., Wishart, D.S., 2016. Using MetaboAnalyst 3.0 for Comprehensive Metabolomics Data Analysis. *Curr. Protoc. Bioinformatics* 55, 14 10 1–14 10 91.

Relevant Websites

- <http://www.chemspider.com/>
ChemSpider.
- <http://www.hmdb.ca/>
HMDB The Human Metabolome Database.
- <http://www.massban.jp>
MassBank.
- <http://www.metabolomexchange.org/>
MetabolomeXchange.
- <http://metlin.scripps.edu/>
METLIN.
- <http://phenomenal-h2020.eu/>
PhenoMeNal.
- <http://www.myexperiment.org/>
WorkFlows.

Mass Spectrometry-Based Metabolomic Analysis

Russell Pickford, University of New South Wales, Kensington, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Metabolites are the intermediates and end products of biological metabolism, the definition typically being limited to small molecules such as amino acids, sugars and lipids.

The field of metabolomics aims to measure metabolites within biological samples to obtain profiles or ‘metabolomes’. These can be compared across sample types – different disease states or gene knockouts for example – to learn more about the underlying biology or to find metabolites which can be used to recognise or define the disease state – biomarkers.

The term ‘metabolome’ was defined as “the complete set of metabolites/low-molecular-weight intermediates, which are context dependent, varying according to the physiology, developmental or pathological state of the cell, tissue, organ or organism” (Oliver, 2002).

Metabolomics is somewhat of an ancient science – ants were used by the ancient Chinese (2000–1500 BC) to detect high levels of glucose in urine – and hence diabetes. Currently, Doctors often test a panel of metabolites within patient’s blood – for example glucose or cholesterol – to ensure that they are within a normal range. These are examples of targeted metabolomics.

Science is now aiming to study organisms as integrated systems to obtain as complete a picture as possible. This systems biology approach combines experimental data from a range of ‘omics experiments’:

- Genomics measure the DNA, a ‘blueprint’ for the organism.
- Transcriptomics measure the RNA transcripts produced by the genome under specific circumstances. Proteomics measures the proteins produced under these circumstances, including metabolic enzymes.
- Metabolomics measures the small molecules or metabolites of the system.

As the final downstream product, the metabolome can be considered the closest reflection of the phenotype (Kell et al., 2005).

The analytical chemistry measurements and subsequent data analysis required to examine a metabolome are complex. Trying to simultaneously measure many different metabolites of widely-varying properties – such as concentration, molecular weight and polarity – is very difficult. No current analytical technique can measure an entire metabolome within a single experiment, or perhaps even with a combination of experiments.

The two techniques most frequently used for metabolomic analysis are mass spectrometry (MS) and Nuclear Magnetic Resonance (NMR). Both have advantages and disadvantages.

NMR has a much poorer detection limit and so can measure fewer metabolites, but is fast, simple and non-destructive to the sample. Also, metabolites observed can almost always be identified from the rich data obtained.

Mass spectrometry is more complex experimentally, destructive to the sample and identification of detected species can be a major challenge.

MS does however offer some key strengths:

- wide dynamic range – the ability to simultaneously measure analytes of very different concentrations, important as the range in biological samples can be many orders of magnitude.
- high sensitivity – the ability to measure analytes at very low concentration.
- reproducible, quantitative analysis – the signal from a metabolite is proportional to concentration.

Data analysis of metabolomics data acquired using mass spectrometry relies on a good understanding of how the data is generated and what information it can and cannot provide. Data analysis ranges from simple univariate statistics to complex manipulations and clustering algorithms.

Instrumentation Employed for Mass Spectrometry-Based Metabolomics

Mass Spectrometry

A mass spectrometer is an instrument that measures the molecular weight and abundance of molecular species within a sample.

The instrument consists of several functional stages (Fig. 1):

- Ionisation
- Mass Analysis
- Detection

Manipulating and detecting molecules with a mass spectrometer requires them to be charged. Neutral species cannot be observed. Thus, the first part of mass spectrometry analysis is ionisation. This stage converts neutral species into gaseous charged ions which can then be manipulated, measured and detected.

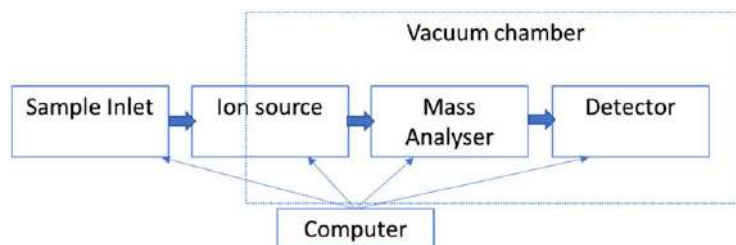


Fig. 1 Schematic detailing the major components of a mass spectrometer.

There is no universal ionisation procedure – different choices here will allow observation of different panels of analytes, but hide others.

Choices of ionisation are typically electrospray (ES) (Fenn *et al.*, 1989; Gaskell, 1997) and atmospheric pressure chemical ionisation (APCI) (Carroll *et al.*, 1974; Byrdwell, 2001) for solution phase analysis and electron ionisation (EI) (Bleakney, 1929) or chemical ionisation (CI) (Munson and Field, 1966) for gaseous analysis. ES, APCI and CI are ‘gentle’ ionisation methods which generally send an intact cationised molecule into the mass analyser. This is important as fragmentation during ionisation can greatly complicate the data produced. EI however produces a radical cation which then usually fragments, sometimes so completely that the intact precursor molecule is not observed.

Except for EI, which only produces positively-charged species, each can be employed in positive or negative polarity. The choice of polarity will again allow observation of different analytes. For example, basic species generally are easier to observe in positive mode and acidic species in the negative mode as they will easily become charged by gaining or losing a proton respectively.

The generated ions are directed into a mass analyser where they are sorted by molecular weight (really mass to charge ratio, m/z). All mass analysers require a vacuum to be functional and operate using a combination of electric fields and (less frequently) magnetic fields to manipulate the ions.

Mass analysers can be divided into two main categories – low resolution and high resolution. Resolution refers to the ability to discriminate analytes which are close in m/z .

Low resolution mass analysers include linear quadrupoles (Douglas, 2009) and quadrupole ion traps (March, 2009), typically operating at unit resolution – they will discriminate mass 100 from mass 99 and mass 101.

High resolution mass analysers include Time-of-Flight (Boesl, 2017), Fourier Transform Ion Cyclotron Resonance (Marshall *et al.*, 1998) and Orbitrap (Zubarev and Makarov, 2013). These are a finer toothed comb – they will separate mass 100.0 from 99.9 and 100.1, and potentially much finer than this – allowing measurement of molecular weight much more accurately and discrimination of species of very similar molecular weight.

Mass analysers should however not be judged by resolution alone. Each has specific advantages and disadvantages and are employed for different experiments (Brune, 1987). For example, advantages of quadrupoles include high transmission, robustness, tolerance of higher pressures and low cost.

When looking for specific targets and requiring high sensitivity then a quadrupole is an excellent choice. When screening samples for as many metabolites as possible then resolution and mass accuracy become important, therefore Time-of-Flight or FT-ICR or Orbitrap are ideal for these profiling experiments.

Different mass analysers and combinations thereof allow different experiments to be performed.

The final component is detection – the abundance of each ion is measured, and a mass spectrum produced. The mass spectrum is a 2D plot with m/z on the x-axis and abundance on the y-axis. The units of the abundance axis vary with instrument, with detector and with manufacturer. It is frequently scaled to ‘relative abundance’ where the most intense signal in the spectrum is assigned 100% and the other signals shown relative to this (Fig. 2).

Separation Techniques

Whilst direct analysis of complex samples is possible, it is generally preferable to ‘hyphenate’ the mass spectrometer with a separation technique. This separates the components in time and space according to some chosen physical property, therefore simplifying the mass spectra obtained at any individual time point. Addition of a separation technique also facilitates automated analysis as well as providing potential enhancements such as potential discrimination of isomeric species (which have the same molecular weight but different structure) and enhancement of signal due to decreased competition during ionisation.

Gas chromatography (GC) is typically employed where the analytes are heat stable and volatile (for example gas, flavour, fragrance analysis) or can be made volatile using simple chemical derivatisations.

Samples are injected into a hot liner where they are vaporised and then directed through a fused-silica column using a carrier gas (mobile phase). Interactions between the analytes and column surface (stationary phase) will differ based on their chemistry, so that some are retained longer and some shorter as they pass through to the detector – providing separation in time. GC separations can be tuned and optimised using column chemistry and column temperature.

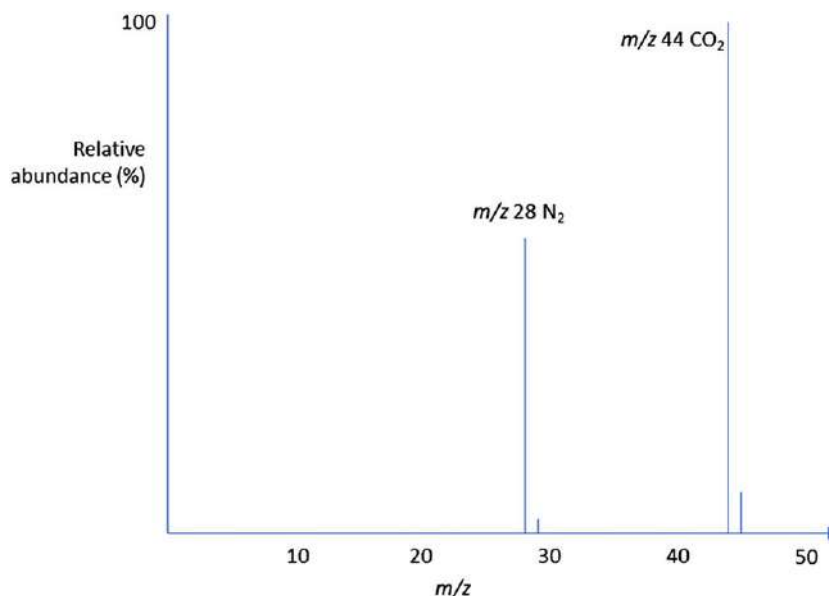


Fig. 2 Mass Spectrum obtained from electron ionisation of a mixture of nitrogen and carbon dioxide gases.

Liquid chromatography (LC) is typically employed for involatile analytes (for example amino acids, sugars, lipids) and does not generally require any prior chemical derivatisation. This makes it a popular choice for untargeted metabolomics as the sample preparation is minimal, reducing experimental bias.

Samples are injected into a stream of liquid – mobile phase – and pass through a stainless-steel or plastic column packed with silica beads derivatised to some specific functionality.

For example, C18 columns are widely used and have an 18-carbon hydrocarbon chain attached to the beads. This provides a hydrophobic surface (stationary phase) for the analytes to interact with as they pass through, therefore non-polar analytes will have a strong interaction with this surface and be retained longer than polar analytes. LC separations can be optimised using a wide range of parameters such as column chemistry, choice of mobile phase solvents and temperature.

Less frequently used, but no less useful, is capillary electrophoresis where analytes are separated by charge and mobility within a buffer solution as they move down a fused silica capillary under the influence of high voltage.

GC–MS and LC–MS

During these analysis methods the output of the separation device is connected to the ionisation source of the mass spectrometer. The output is continuously ionised, and the ions produced directed into the mass spectrometer. Mass spectra are recorded, typically every second or so, throughout the whole run time. The data therefore consists of several hundred to thousand consecutive mass spectra – it is now a 3D plot with the axes of time, m/z and abundance. This can be visualised as an ‘ion map’ with time and m/z as the axes and intensity of colour as the abundance (**Fig. 3**).

The use of mass spectrometry as a detector places some limits on what can be used for the separation part of the experiment – for example LC–MS mobile phases and sample components need to be volatile – salts cannot be used.

Tandem Mass Spectrometry – MS/MS

Mass spectra generated by ES, APCI and CI generally only contain intact cationised analytes, yielding the molecular weights of the compounds within the sample. An extra dimension of information can be obtained using tandem mass spectrometry. This process is performed within the mass spectrometer and involves a specific precursor ion (m/z) being isolated and fragmented, normally via collisions with a neutral gas such as nitrogen or argon. These product ions are then detected and can yield a ‘fingerprint’ helping to identify the analyte. Two analytes can have identical molecular weight – isobaric species – but very different MS/MS fingerprints and can therefore be discriminated by MS/MS but not by MS (Yost and Fetterolf, 1983; de Hoffmann, 1996).

The tandem mass spectrum produced from MS/MS analysis of the precursor ion at m/z 784 is shown (**Fig. 4**). Some remaining (unfragmented) precursor can be observed together with a large product ion at m/z 184. This ion is structurally diagnostic – it is formed from lipid molecules containing a choline group and therefore yields information that the precursor at m/z 784 is a phosphatidylcholine.

The use of MS/MS can add an additional dimension to the data if employed together with MS scans, such as in the case of profiling experiments. The data now contain a time component and both MS and MS/MS scans.

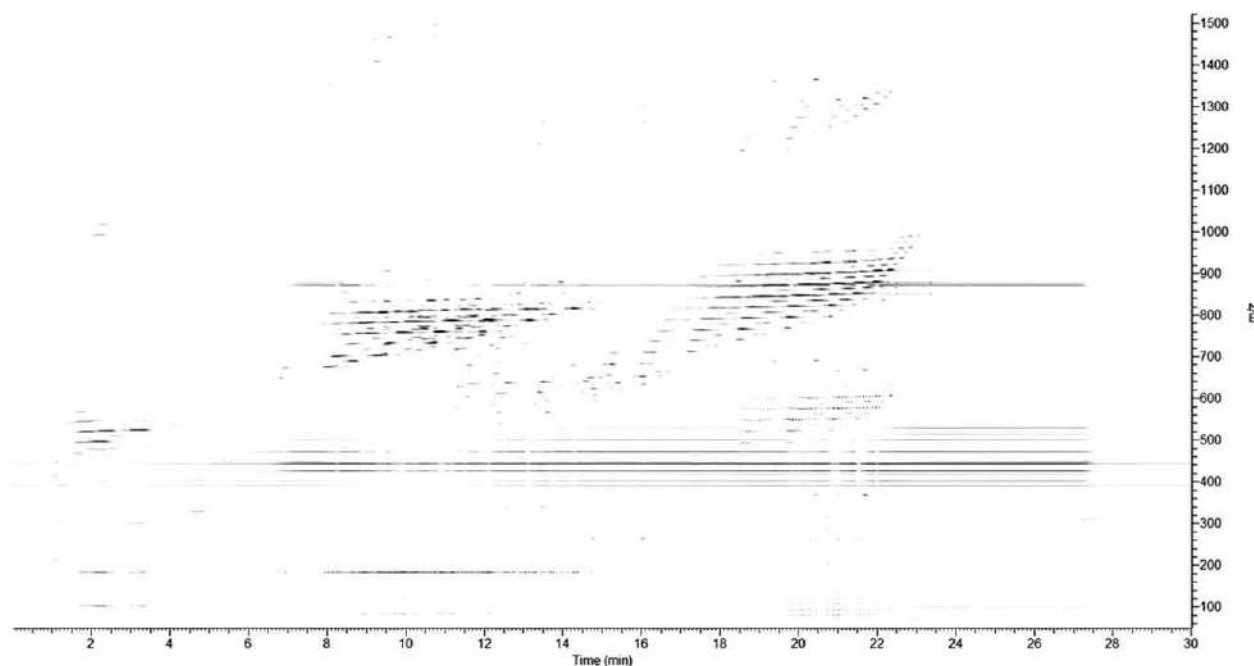


Fig. 3 Ion map obtained from LC-MS analysis of a plasma lipid extract.

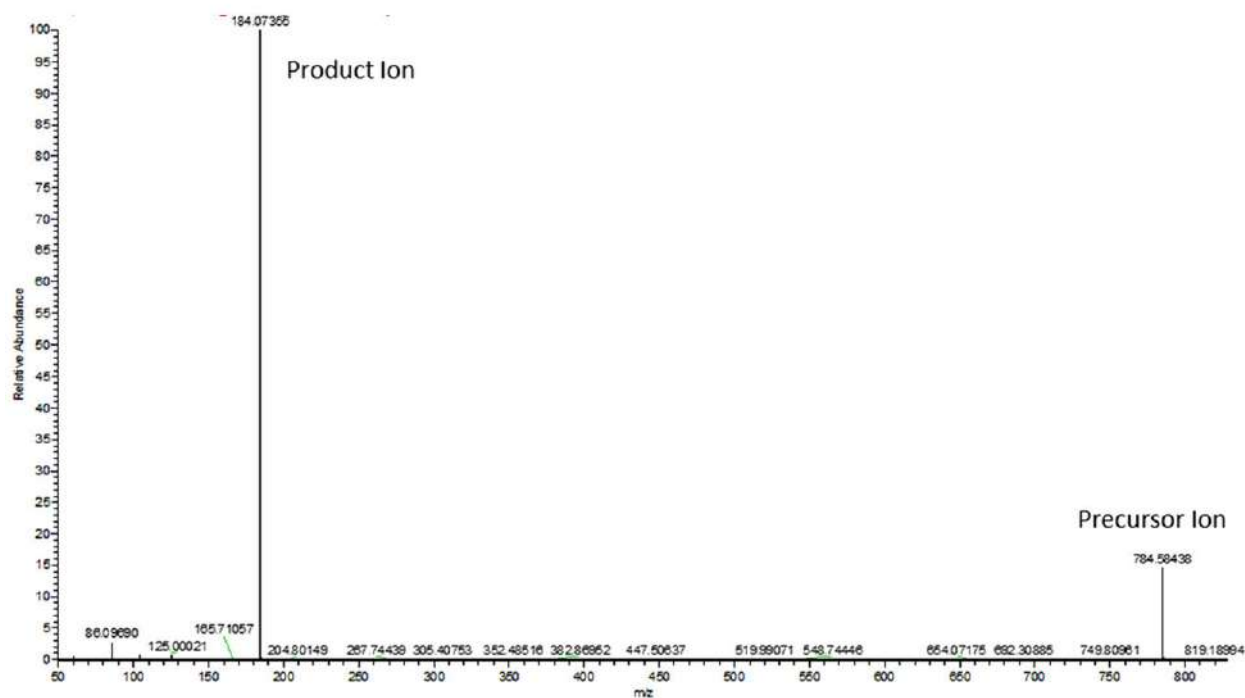
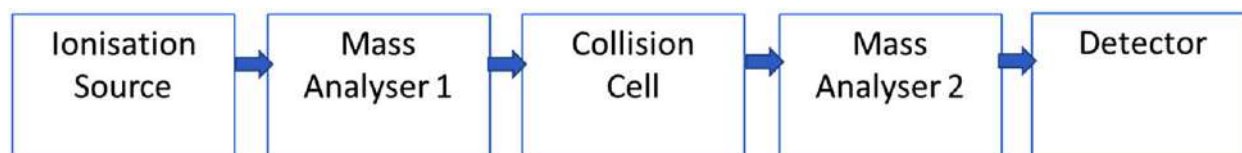


Fig. 4 Tandem mass spectrometry instrument components and MS/MS spectrum.

When performing untargeted or profiling experiments the analyte masses are not known. Automated, rules-based approaches are therefore employed to select precursor ions for MS/MS analysis. For example, a full MS 'survey' scan performed and then consecutive MS/MS scans on the 6 most intense species in this survey scan, and the cycle repeats throughout the run. Rules can be added – to include or exclude selected masses or charge states for example. This is generally termed data-dependent analysis – DDA ([Mann et al., 2001](#)).

Performing MS/MS with a very wide mass selection window is becoming popular, for example simultaneously fragmenting everything over the mass range 150–250 Th. These have less specificity and the data are harder to interpret since the connectivity between precursor and product ions is no longer certain. They do however provide much greater MS/MS coverage. This approach is termed data independent analysis – DIA ([Doerr, 2014](#)), MS^E ([Plumb et al., 2006](#)) and SWATH ([Gillet et al., 2012](#))

Mass Spectrometer Data Files

The data structure remains similar, but each instrument manufacturer produces their own data format – .d from Agilent, .wiff from Applied Biosystems and .raw from ThermoFisher Scientific for example.

These formats are proprietary and need to be analysed within the vendor's own software or be converted to a community format such as NetCDF ([Unidata, 2016](#)), mzDATA ([Orchard et al., 2007](#)) or MzXML ([Pedrioli et al., 2004](#); [Lin et al., 2005](#)). Since 2008 the community has largely deprecated prior formats and adopted mzML ([Deutsch, 2008](#)) as the unified standard. These formats are readable and accessible to freeware or web-based data processing tools. Open-source tools such as ProteoWizard ([Kessner et al., 2008](#)) can be employed to batch convert files from proprietary to open source.

General Considerations for Metabolomics Experiments using Mass Spectrometry

Experimental Design

The experimental design is important to consider prior to any practical work being performed. The design will depend upon the statistical rigour required (power analysis) as well as the experimental goals and practical limitations. If we are comparing changes across two sample types, then large changes in metabolite concentration are easier to see and have confidence in. Small changes require much more rigour to be able to observe confidently.

Replication is important – technical replicates are repeat analyses of the same sample which allow the variability of the measurement to be calculated. Biological replicates allow the natural variability of the target metabolites within a sample set to be examined. These are both very important – if the target analyte varies 10% between test and control but varies 20% within control then it is very hard to draw any confident conclusions about the result. Acceptable minimum standards for replication vary between scientists and experiments. The data is generally assessed using at least mean and standard deviation (coefficient of variation). These require a normal distribution of data and therefore at least 6 replicates to describe this. Generally, more is better!

It is also important to randomise the samples during instrumental acquisition. The instrument's performance can change over time and the effect of this on the data needs to be minimised, so the comparison is really between test vs control and not data from clean instrument vs data from dirty instrument. For the same reason all samples should be run in a single batch and not on different days.

There are many confounding variables when dealing with biological samples, which must be controlled as much as possible. The experiment may be testing the blood of cancer patients vs healthy patients, but not all differences observed will be due to the cancer. Metabolite profiles can vary for many reasons, for example, age, sex, race, obesity, exercise.... Even diet is an influence as what is eaten makes its way into the blood, this is one reason that pathology labs prefer to take patient blood in a fasted state.

It is therefore important to know about the variability in the sample population, and control as best possible. It is no good looking for a cure for cancer and instead finding a biomarker that differentiates males from females.....

Sample Preparation for Mass Spectrometry Analysis

Preparing the biological sample so that it may be analysed by mass spectrometry is a key part of metabolomics experiments ([Raterink et al., 2014](#)). The sample preparation will have a large effect on the result – changes to sample preparation will change the metabolites observed in the experiment.

The general principle is to extract the metabolites from the sample with as light a touch as possible. For example, urine metabolomics may just require a simple dilution, plasma will require protein precipitation and tissue will require homogenisation and extraction.

If the experiment is targeting a metabolite or panel of metabolites then the sample preparation can be optimised experimentally – choices of solvents, pH, volumes etc can all be optimised for maximum signal. A preconcentration step such as solid phase extraction may also be employed. If the experiment is untargeted then the extraction and preparation normally seek to extract as wide a range of metabolites as possible with as little perturbation to the sample as possible.

Targeted Metabolomics

Targeted metabolomic experiments seek to accurately quantify specific metabolites within a sample (Roberts *et al.*, 2012; Griffiths *et al.*, 2010; Koal and Deigner, 2010). They target a limited number of analytes. For example, to observe what happens to the concentration of a drug in a patient's body over time only detection of this drug is required. To study heart disease a panel of key analytes involved in cardiovascular metabolism may be chosen (Griffin *et al.*, 2011).

Only the targeted analytes are observed in the data, the rest of what is happening with the sample is hidden. This is therefore a trade off, achieving excellent data concerning one panel of metabolites but no data about the rest of what may be happening with the sample.

Absolute quantification obtains the amount of analyte within a sample, normally measured in moles or by weight, for example nanograms analyte per gram of tissue.

Quantification with mass spectrometry is performed by regression analysis of a calibration curve plotting signal against concentration, therefore an authentic standard is required for every individual compound quantified. Internal standards are employed to correct for experimental error, these are generally isotopically-labelled (C13, N15) analogues of the analyte, or close structural analogues which are known not to be contained in the sample, again with each analyte ideally having its own internal standard (Green, 1996).

It is possible to perform relative quantification across samples and sample sets when authentic standards are not available, however the results are reported only as a fold-change difference in peak areas obtained, which is difficult to draw biologically-relevant conclusions from. Typically, untargeted metabolomics will produce relative quantification results and then the more interesting changes can be confirmed and more accurately characterised using targeted experiments.

Data Characteristics of Targeted Metabolomics Experiments

The data output of the mass spectrometer during targeted quantification experiments is a plot of signal (relative abundance) against time for each analyte (Fig. 5). The signal can be from the intact analyte – normally termed Selected Ion Monitoring – or from a selected fragment ion produced by MS/MS of the analyte – normally termed Selected Reaction Monitoring. The fragment ion is selected to optimise for sensitivity and specificity.

The plot is then integrated, and the area of the peak produced by the sample signal recorded. Integration is performed in an automated manner with each mass spectrometry vendor providing their own set of algorithms. The shape of the peaks produced are somewhat variable with analyte and with separation method, therefore the integration process requires some manual checking and tuning to do the best job possible.

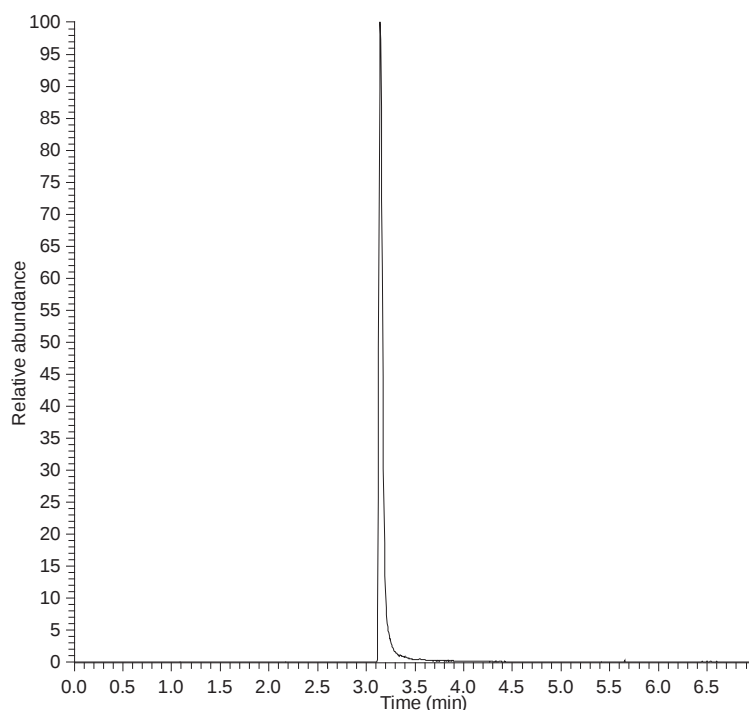


Fig. 5 LC-SRM trace obtained from analysis of Verapamil.

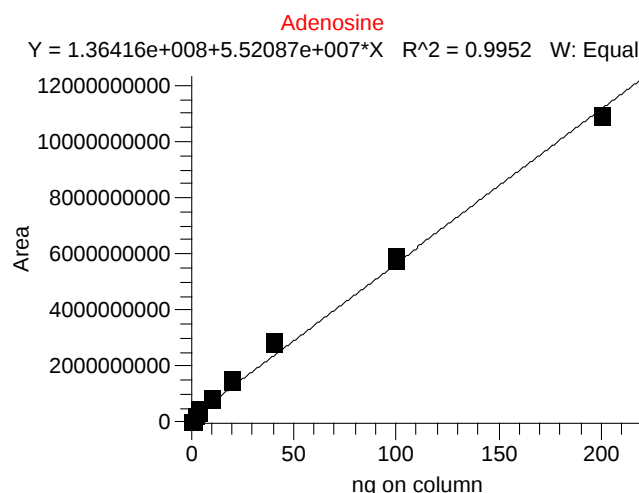


Fig. 6 Calibration Curve obtained from LC-SRM analysis of Adenosine Standard.

Filename	Sample Type	Inj Vol	Retention	Calculated Amount	Units
4ngc	Standard	30.00	1.2	3.579	ng on column
4ngd	Standard	30.00	1.2	3.816	ng on column
27	Unknown Sample	20.00	1.2	0.000	ng on column
28	Unknown Sample	20.00	1.2	3.155	ng on column
29	Unknown Sample	20.00	1.2	3.979	ng on column
3	Unknown Sample	20.00	1.2	0.021	ng on column
30	Unknown Sample	20.00	1.2	3.744	ng on column
31	Unknown Sample	20.00	1.2	4.586	ng on column
4	Unknown Sample	20.00	1.2	0.055	ng on column
5	Unknown Sample	20.00	1.2	0.029	ng on column
6	Unknown Sample	20.00	1.2	0.029	ng on column
7	Unknown Sample	20.00	1.2	0.284	ng on column
8	Unknown Sample	20.00	1.2	0.122	ng on column
9	Unknown Sample	20.00	1.2	0.107	ng on column

Fig. 7 Part of an output table after LC-SRM quantitative analysis.

Metabolite Quantification

A calibration curve is obtained by regression analysis after plotting the peak area against concentration for a known set of standard concentrations ([Fig. 6](#)). If an internal standard is used for normalisation, then the plot becomes (peak area standard/peak area internal standard) vs concentration.

The concentration of the analyte is calculated using regression analysis of the standard curve – the peak area obtained is transformed into concentration.

Data Analysis

Targeted metabolomic data ([Fig. 7](#)) is typically analysed in a spreadsheet using univariate analysis – the comparison of a single variable (analyte amount) between samples and sample sets ([Vinaixa et al., 2012](#)). For example, the amount of glucose in the blood of diabetics vs healthy individuals.

Analysis is normally performed with replicates – both biological and technical. Therefore, a mean and a standard deviation or coefficient of variation are the required outputs. Outliers can be removed to improve the data quality after appropriate analysis, although this requires much care – the sample measurement may be very different to the rest of the set, but it may be due to genuine biological variation not experimental error.

Typical statistical tests employed are T-tests (to compare the mean value across two groups) and Analysis of Variance (ANOVA) to compare multiple groups.

Fluxomics

A subset of targeted metabolomics is fluxomics (Winter and Krömer, 2013; Munger *et al.*, 2008), which seeks to determine the rates of metabolic reactions within the samples. This differs from targeted metabolomics which provide a snapshot of metabolite concentrations.

Fluxomics can be performed using ^{13}C or ^{15}N labelled precursors introduced to the biological system – these have the same chemical properties as the unlabelled counterpart but an increased mass. They can therefore be discriminated using a mass spectrometer. The progress of the label down metabolic pathways can be quantified – the appearance of ‘heavy’ downstream metabolites – and the metabolic flux estimated.

Untargeted Metabolomics

Untargeted metabolomics experiments seek to measure as many compounds/metabolites as possible within a sample – it is an untargeted approach which ‘casts the fishing net’ as widely as possible. Samples are most frequently analysed using LC–MS (although GC–MS can be used) with a high-resolution mass analyser scanning a wide mass range, e.g., 50–2000 Th.

Detected components are compared across sample sets to find differences and yield biologically-relevant information.

Experimental Considerations

The components detected will depend upon the LC(GC)-MS experiment performed. Electrospray ionisation will allow observation of a different set of ions than APCI. Positive polarity will show different compounds than negative polarity. Even the choice of separation method, and derivatisation method for GC, will change the panel of metabolites observed. It is important to remember there is no global ‘see everything’ metabolomics assay.

Biological replicates are very important here – the mean peak area obtained from compound X is compared between ‘test’ and ‘control’ sample sets to search for (significant) differences which may provide information about the biological state of the samples – an upregulation in a certain metabolic pathway for example. Sufficient replication to accurately describe the within-sample-set variation is therefore required.

It is also a good idea to include process blanks. Mass spectrometry is a very sensitive detection technique and will detect many background and contaminant ions from the solvents, solutions, plasticware etc used in the experiment – interfering signals not due to the sample. By including blanks in experiments and subsequent data analysis these ‘background components’ can be recognised and dealt with appropriately.

Workflow

Untargeted metabolomics experiments generally follow the same workflow, shown schematically in Fig. 8.

There are many software packages available to facilitate the analysis of untargeted metabolomics data.

Instrument manufacturers generally produce a commercially-available software to work with data from their own instruments. SIEVE and Compound Discoverer (ThermoFisher), ProfileAnalysis (Bruker), Progenesis QI (Waters) and MassHunter (Agilent) for example. Progenesis QI also will analyse data from other instrument manufacturers.

There are many freely-available tools available on the web, which are more universally-applicable. The most popular being XCMS(2) (Smith *et al.*, 2006; Gowda *et al.*, 2014), MetaboAnalyst (Xia *et al.*, 2009, 2015) and MzMine2 (Pluskal (2010)).

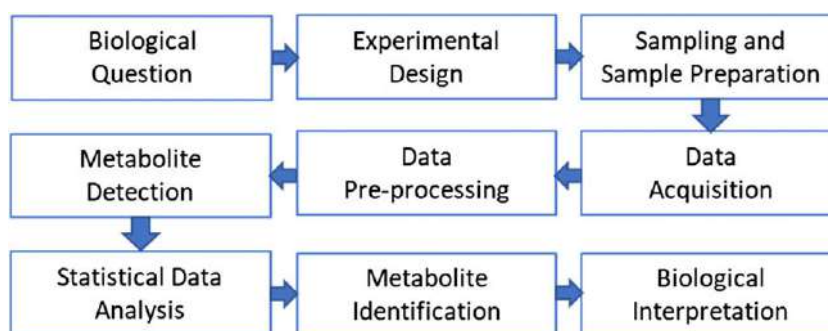


Fig. 8 Workflow of an untargeted metabolomics experiment.

MetabolomeExpress (Carroll *et al.*, 2010) is also used for GC-MS data analysis. Data generated from Data Independent Analysis (DIA) experiments can be analysed using MS-DIAL (Tsugawa *et al.*, 2015).

At the time of writing there is no universally recognised 'best' software or approach, usage depending on the experiment, user preference and software availability.

Data Pre-Processing

Data conversion

The output from the mass spectrometer is encoded in a format specific to the instrument manufacturer. These generally need to be converted into open source formats prior to software interrogation, particularly when using vendor-independent or freely available software both on- and off-line.

Widely usable data formats are NetCDF, mzDATA, mzXML and mzML. Instrument manufacturers frequently include an option to export data to this format from within their own data analysis software.

A useful data conversion tool is the open source ProteoWizard (Kessner *et al.*, 2008), this tool accepts data from a wide variety of instrument manufacturers and will batch convert into open source formats.

Data import

The first step in data analysis is to import the data into the metabolomics analysis software, this is normally performed in an automated manner. The 3-dimensional data sets (time vs m/z vs intensity) are then prepared for analysis. The data may contain an extra dimension – MS/MS – which is useful to aid in compound identification but is ignored during the early processing stages.

The three-dimensional data is generally converted into stacks of two-dimensional chromatograms by plotting the signal intensity of narrow m/z windows against time to produce extracted ion chromatograms. The size of these bins will depend on the mass analyser used – the data obtained from high resolution instruments allows many small bins to be used and therefore have high discriminating power between compounds of very similar molecular weight. The data from low resolution instruments can only be put into large windows or bins.

Data alignment

The data is then aligned – the instrument time required to analyse large numbers of samples can be significant, and variations during the data acquisition produce small shifts or drifts in chromatographic retention time. Signal from m/z X at time Y cannot therefore simply be compared across samples. There needs to be some alignment (shifting/warping) in the time dimension to line up the data and ensure correct comparison across samples and sample groups (Fig. 9). The shift is not a linear one and so warping algorithms are preferred. These are numerous (Smith *et al.*, 2013) and include Time Correlation Optimised Warping, Dynamic Time Warping and Parametric Time Warping (Tomasi *et al.*, 2004). There is no clear 'best' algorithm, with results being context and data dependent.

Missing values can be a significant problem for metabolomics statistical analysis – what is done when a signal is present in one sample set or one sample within a sample set but not the other(s)? Various approaches have been employed to solve this including missing value imputation (Armitage *et al.*, 2015; Hrydziusko and Viant, 2012; Di Guida *et al.*, 2016) or producing an aggregate run from all samples in the dataset and then using this for peak picking. This ensures the same ion is detected in every individual run and produces a complete data set which is required for valid multivariate analysis. Backfilling with zeros for missing values does not produce a valid dataset for multivariate analysis and should be avoided. However, there is no clear 'best' solution to the missing values problem, with results being context and data dependent.

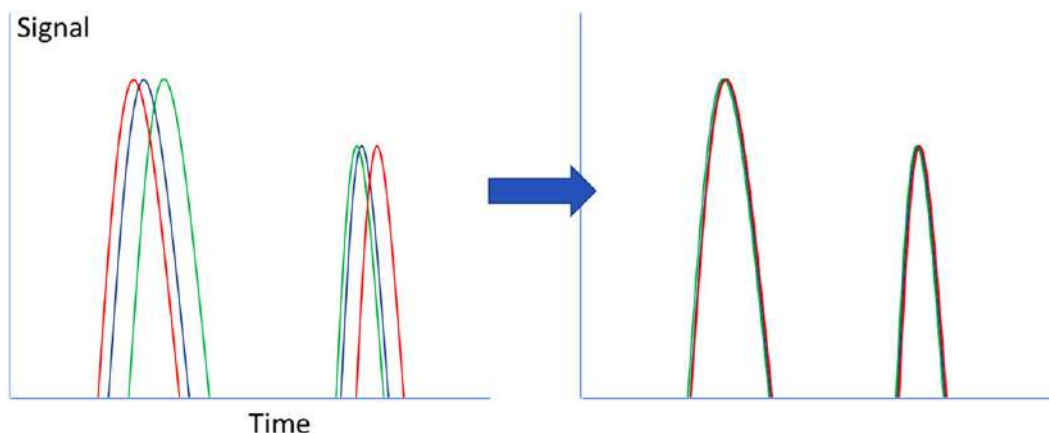


Fig. 9 Cartoon example of unaligned and aligned LC-MS data.

Data normalisation

After alignment the data is then normalised (scaled) to adjust for experimental noise and variation. The goal is to remove experimental bias whilst keeping the signals relating to genuine biological differences (De Livera *et al.*, 2015). Examples of sample systematic bias include differences in sample concentration, particularly with biofluids – for example urine can be more dilute or more concentrated depending on the individual's hydration status. Experimental variations in signal due to sample preparation and instrumental analysis can also arise. Targeted metabolomic analysis can correct to a 'housekeeping' endogenous metabolite such as creatinine in urine, although this can be problematic due to the assumptions made. The ideal solution is the use of internal standards to correct for experimental bias, however a single internal standard will not correct properly for all the compounds detected in an untargeted experiment. The use of multiple internal standards improves the situation although this is still not ideal. Normalisation methods based on statistical methods (unit norm, median and quantile), scaling methods (auto scaling, range scaling, Pareto scaling, vast scaling and level scaling) and data transformation (log and power) are all employed (Sugimoto *et al.*, 2012). There is no consensus best approach, and several should be tested (Di Guida *et al.*, 2016).

Metabolite Detection

Peak picking and deconvolution

Peak picking algorithms are then applied to the aligned and normalised data to identify peaks over a user specified threshold, for example peak height or peak area. Wavelet transformation and Gaussian-curve fitting is commonly used to distinguish signal to noise (Smith *et al.*, 2006; Tautenhahn *et al.*, 2008), although manual tuning and checking is required due to the compound-specific variation in chromatographic peak shape, particularly with LC-MS and CE-MS. The output is generally a list of mass-retention time pairs e.g., *m/z* 100-4.5 min refers to a compound of *m/z* 100 which elutes with a peak intensity maximum at 4.5 min. Each of these will have an associated integrated peak area.

This data can then be deconvoluted or reduced – each compound may ionise in different ways or fall victim to fragmentation in the source region of the mass spectrometer. For example, a compound may be protonated, sodiated and be observed as protonated with loss of water $[M + H - H_2O]^+$. These can be recognised as they will have matching chromatographic profiles (signal vs time) and can then be deconvoluted to a single data entry of mass and retention time. Reducing the size of the dataset at this point simplifies the downstream data analysis. Caution must be applied however – these could be separate species which happen to have the same chromatographic profile.

Feature list

The final output of this process is a feature list or data matrix. The matrix contains the features detected and their abundance (integrated peak area) within the different samples.

Data can then be filtered – for example only those features showing a fold change greater than $2 \times$ between test and control or only those features observed with high significance or with a low coefficient of variation.

Exploratory data analysis methods are then used to find patterns in the filtered or unfiltered data

Statistical Data Analysis

Data analysis normally employs multivariate analysis (Worley and Powers, 2013). This is used to identify metabolite patterns that correlate in a statistically significant manner with a sample set/phenotype so that they can be used for diagnostics.

The main reason for using multivariate as compared to univariate analysis for metabolomics is simply since there are many variables in each sample, and these usually correlate to the phenotype in a concerted fashion rather than correlating independently. For example, your fingerprint contains many data points which combine to a unique whole. Consequently, it is important to identify metabolic profiles or 'fingerprints' that relate to the phenotype rather than individual, independent biomarkers.

It is important to apply appropriate statistical tools for the experiment. For example, comparing 'test' and 'control' samples can be done using one-factor ANOVA to search for statistically significant differences between groups. However, if the samples are a time course study with several samples being taken from the same source (for example a cell culture being sampled at 0, 4, 8, 24 h) then one-factor ANOVA is not appropriate as the groups being compared are not truly independent. In this case different tools such as repeated measures ANOVA must be employed.

Multivariate analysis tools can be supervised or unsupervised. Unsupervised methods are exploratory in nature, showing previously unknown patterns in the data (Fig. 10) or helping to confirm suspected patterns. Supervised methods use experimental knowledge (for example sample groups) to discover associated patterns.

Principal Component Analysis (PCA) (Jolliffe, 2002) is frequently used for untargeted metabolomics data analysis. This is an unsupervised data analysis tool – the algorithm is unaware of which are test, and which are control samples. PCA allows clustering of sample sets and to determine which variables (compounds/features) have the highest ability to discriminate between the sample sets.

PCA is an unbiased process and may reveal previously unknown clusters in the data – for example when comparing healthy vs diseased patients, we may see further splitting into groups male and female or fed and fasted. This can help to recognise limitations or errors in the experimental design.

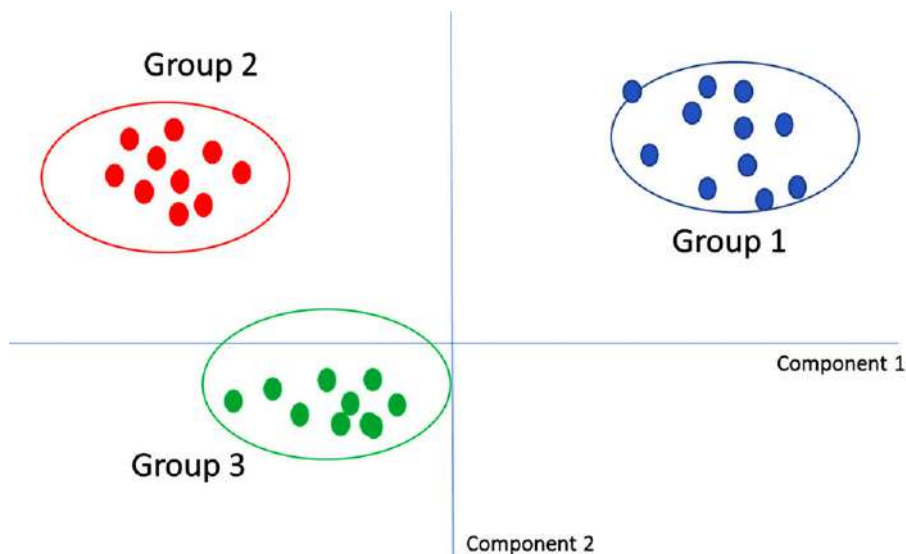


Fig. 10 Cartoon Principal Component Analysis (PCA) Plot demonstrating clustering of 3 sample groups.

The clusters/groupings also allow recognition of samples which may be outliers – they do not cluster with the rest of their biological group.

However, PCA only reveals group structure when within-group variation is less than between-group variation and so may not provide separation between very similar samples.

Alternative tools employ Partial Least Squares algorithms to discriminate groups (Barker and Rayens, 2003). Partial Least Squares Discriminant Analysis (PLS-DA) and Orthogonal Partial Least Squares Discriminant Analysis (OPLS-DA) are the most well-known tools for multivariate analysis in metabolomics (Brereton and Lloyd, 2014). These are supervised analysis tools – information about which samples belong to which group is given to the algorithm. The algorithm then seeks to maximise the differences between the sample groups and show the variables responsible for this separation.

A danger of employing these algorithms is that they can aggressively over-fit models to the data, almost always managing to separate the defined groups or classes. Consequently, PLS-DA and OPLS-DA can generate excellent class separation from random data! This can be controlled for somewhat by creating training sets and testing them with other data or by performing permutation tests.

An alternative statistical tool is to use clustering methods on the data. These are unsupervised machine learning techniques which seek to group samples together using similarities in their measurements. Frequently used clustering techniques in metabolomics are K-Means (Hartigan and Wong, 1979), Hierarchical Clustering (Johnson, 1967), and Self-Organising Maps (Kohonen, 1990).

Further data analysis

Most metabolomics software packages will allow export of the data matrix (feature list) in .csv format. This file can then be further analysed in statistical packages such as the commercial SIMCA-P or the freely available R software using scripts.

Metabolite Identification

The final stage of data analysis is to identify those compounds (Kind *et al.*, 2017; Vaniya and Fiehn, 2015) found to be of likely biological interest – statistically different between groups and/or discriminating between groups for example.

Identification is not a trivial task, hence efforts are generally directed towards ‘interesting’ metabolites rather than trying to identify every feature identified in the experiment.

The information obtained about a compound will be molecular weight, retention time on the LC or GC and perhaps an MS/MS fingerprint. These properties can be targeted in database searches.

Several online and accessible chemical and metabolomic databases are available

- ChemSpider (Pence and, Williams, 2010) is a large chemical database, containing over 60 million structures from hundreds of data sources. It is the most complete and therefore the biggest chance of a ‘hit’. It is also the biggest chance of a false positive.
- PubChem (Wang *et al.*, 2009) is another large general chemical database provided by the NIH.
- Metlin (Smith *et al.*, 2005) is curated specifically for metabolomics experiments, containing MS and MS/MS spectra.
- Human Metabolome Database – HMDB (Wishart *et al.*, 2017), contains detailed information about metabolites found in the human body.
- Small Molecule Pathway Database – SMPDB (Jewison *et al.*, 2014) contains more than 700 small molecule pathways found in humans and can assist with pathway elucidation and discovery.

- Kyoto Encyclopaedia of Genes and Genomes – KEGG ([Kanehisa et al., 2017](#)) contains pathway information including genomes, proteomes and metabolomes from a variety of species.
- MassBank ([Horai et al., 2010](#)) is a large shared mass spectral data repository.

The largest database is not always the best – it will contain many species not related to the samples under analysis, therefore increasing the search space and complicating the analysis. A reasonable strategy would be to attempt identification using the smaller and better curated databases first, moving on to larger databases as required.

Identifying a compound by molecular weight alone is very difficult – imagine trying to identify a single person in a stadium full of people using only their weight. It is obvious that there will be many people/compounds of the same or similar (molecular) weight, and that discriminating these will be difficult. High resolution and accurate mass data is helpful here as it significantly narrows the search space. However, even measuring with sufficient accuracy to yield an elemental composition does not usually provide sufficient discrimination to narrow down to a single compound – isomeric or isobaric alternatives are possible.

Chromatographic retention time information can be helpful to narrow the search space, but is not frequently employed due to the wide variation in separation methods employed in different laboratories and consequent wide variation in experimental retention times obtained. An exception to this is the Golm Metabolome Database which contains GC–MS data and uses retention times scaled to a set of standards.

Further information within the mass spectrum can be useful – the relative intensities of the lower abundance isotopes of the compound to confirm isotopic composition for example – but are not often utilised.

Tandem mass spectra or MS/MS fingerprints are useful discriminatory tools but currently only a limited number exist in the available databases for LC–MS based metabolomics ([Vinaixa et al., 2016](#)). This is due to experimental variation in spectra obtained from different instrument configurations and different instrument manufacturers, which makes production of a library difficult. However, in the case of Electron Ionisation GC–MS the spectra obtained are very similar across different instruments and different manufacturers and hence a large generic GC–MS database is commercially-viable. The most commonly used of these are from Wiley and NIST. Employing these large spectral libraries makes confident identification of species observed in GC–MS currently much easier than LC–MS.

An increasingly popular alternative is to compare the experimental spectra with those obtained from rules based in silico fragmentation of a large library of molecules using software such as the commercial Mass Frontier, Progenesis QI, or the freely available MetFrag ([Ruttkies et al., 2016](#)).

Once a tentative identification has been made using database searching then experimental confirmation should be performed using a purchased or synthesised authentic standard. The standard should have the same behaviour – observed m/z , retention time and MS/MS fingerprint – as the unknown. It is preferable to perform large scale purification and NMR as a ‘gold standard’ confirmation. Standards are not always available, making this the only option in these cases.

Pathway Analysis

Metabolomics data can be employed as part of a systems biology approach looking to understand the interactions between genes, proteins and metabolites ([Krumsiek et al., 2016](#); [Johnson et al., 2016](#)).

This can be employed using various ‘pathway analysis’ tools. Some are incorporated into online metabolomics software such as XCMSOnline and MetPa within MetaboAnalyst. Others are commercially-available packages such as Ingenuity Pathway Analysis from Qiagen Bioinformatics.

The tools will input metabolite information – identity and concentration/peak area – and combine it with protein and gene data from different experiments. The combination allows a bigger overview and provides mechanistic insight – upregulated enzymes resulting in an increase of their metabolic output for example.

This valuable information comes at significant time and money cost – the input must be heavily curated, only including truly significant differences between sample groups. Often further experiments must be performed to confirm the identities of the metabolites observed.

Biological Interpretation

The final stage in metabolomics analysis is of course biological interpretation – making sense of the how the observed changes in metabolites map to changes in the phenotype. Having well-curated data at this stage is paramount. Interpretation can be somewhat of an iterative process – if unexpected changes are observed are we confident about the statistics and identity of this metabolite? If expected changes are not observed have we missed something by over filtering the data or misidentifying the metabolites?

Major Challenges in Mass Spectrometry-Based Metabolomics

Mass spectrometry powered metabolomics is a powerful tool, providing rich, reproducible and high-quality data from the samples analysed. It is not without shortcomings however, which can be broadly categorised into experimental and data analysis.

Experimental

- Control of samples – requirements for large numbers of well-controlled biological replicates.
- High cost of equipment.
- High level of expertise required to perform sample preparation reliably and reproducibly.
- High level of expertise required to operate mass spectrometers and obtain reliable, reproducible data.
- Importance of experimental design – the data obtained must be able to answer the questions asked.

Data analysis

- Variety of data formats produced from different instruments – often require conversion before analysis.
- Data quantity and large file sizes (up to several GB per sample) make data handling difficult and time consuming.
- Appropriate handling of outliers can be difficult to decide.
- Requires appropriate selection of data analysis tools, PCA vs PLS-DA for example.
- Difficult to identify metabolites from mass spectrometric data alone.
- Observation of a metabolite change between phenotypes does not mean that this metabolite is responsible for the phenotype – causality or coincidence cannot be established directly from the data.

Lack of standardisation of rigour, experimental approach, outputs and reporting between different labs and instrument manufacturers is currently a large problem in metabolomics (Rocca-Serra *et al.*, 2016). This makes comparisons and reproducing results between different laboratories very difficult. Also, for example, the data output format of the metabolomics experiments may not match the requirements for input for pathway analysis, depending on the software used. Consortia such as the Metabolomics Standards Initiative (Fiehn *et al.*, 2007) and COSMOS (Salek *et al.*, 2015) have been set up to tackle some of these issues. There is movement in the right direction, for example the adoption of International Chemical Identifiers (InChIKeys) to describe metabolite structure. The Metabolomics Workbench (Sud *et al.*, 2016) also offer a public repository for metabolomics data, helping standardisation and research efforts.

Applications

Targeted and untargeted metabolomics have an incredibly wide variety of applications. These include toxicology, study of disease, nutrition and functional genomics.

Future Directions

Metabolomics has a bright future and is an area of intensive research around the globe. Future areas for improvement include

- Advances in instrumentation such as more accurate and more sensitive mass spectrometers and more powerful separation technologies.
- Advances in community analysis such as widely-available cloud-based databases for compound identification.
- Increasing numbers of MS/MS libraries suitable for searching for identities within LC-MS/MS data.
- Community databases of ‘known unknowns’ where the metabolite’s identity could not be ascertained, but is observed by other laboratories in their samples.

See also: Computational Lipidomics. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Investigating Metabolic Pathways and Networks. Metabolic Profiling. Metabolome Analysis. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics

References

- Armitage, E.G., Godzien, J., Alonso-Herranz, V., López-González, Á., Barbas, C., 2015. Missing value imputation strategies for metabolomics data. *Electrophoresis* 36 (24), 3050–3060. Available at: <https://doi.org/10.1002/elps.201500352>.
- Barker, M., Rayens, W., 2003. Partial least squares for discrimination. *Journal of Chemometrics* 17, 166–173. Available at: <https://doi.org/10.1002/cem.785>.
- Bleakney, W., 1929. A new method of positive ray analysis and its application to the measurement of ionization potentials in mercury vapor. *Physical Review* 34 (1), 157–160. Available at: <https://doi.org/10.1103/PhysRev.34.157>.
- Boesl, U., 2017. Time-of-flight mass spectrometry: Introduction to the basics. *Mass Spectrometry Reviews*. Available at: <https://doi.org/10.1002/mas.21520>.

- Brereton, R.G., Lloyd, G.R., 2014. Partial least squares discriminant analysis: Taking the magic away. *Journal of Chemometrics* 28 (4), Available at: <https://doi.org/10.1002/cem.2609>.
- Brune, C., 1987. The ideal mass analyser: Fact or fiction? *International Journal of Mass Spectrometry and Ion Processes* 76, 125–237.
- Byrdwell, W.C., 2001. Atmospheric pressure chemical ionization mass spectrometry for analysis of lipids. *Lipids* 36 (4), 327–346. Available at: <https://doi.org/10.1007/s11745-001-0725-5>.
- Carroll, A.J., Badger, M.R., Millar, A.H., 2010. The metabolomeexpress project: Enabling web-based processing, analysis and transparent dissemination of GC/MS metabolomics datasets. *BMC Bioinformatics* 11, 376–388. Available at: <https://doi.org/10.1016/j.febslet.2005.01.029>.
- Carroll, D.I., Dzidic, I., Stillwell, R.N., Horning, M.G., Horning, E.C., 1974. Subpicogram detection system for gas phase analysis based upon atmospheric pressure ionization (API) mass spectrometry. *Analytical Chemistry* 46 (6), 706–710. Available at: <https://doi.org/10.1021/ac60342a009>.
- de Hoffmann, E., 1996. Tandem mass spectrometry: A primer. *Journal of Mass Spectrometry* 31 (2), 129–137. Available at: [https://doi.org/10.1002/\(SICI\)1096-9888\(199602\)31:2<129::AID-JMS305.3.0.CO;2-T](https://doi.org/10.1002/(SICI)1096-9888(199602)31:2<129::AID-JMS305.3.0.CO;2-T).
- De Livera, A.M., Sysi-Aho, M., Jacob, L., et al., 2015. Statistical methods for handling unwanted variation in metabolomics data. *Analytical Chemistry* 87 (7), Available at: <https://doi.org/10.1021/ac502439y>.
- Deutsch, E., 2008. mzML: A single, unifying data format for mass spectrometer output. *Proteomics* 8 (14), 2776–2777. Available at: <https://doi.org/10.1002/pmic.200890049>.
- Di Guida, R., Engel, J., Allwood, J.W., et al., 2016. Non-targeted UHPLC-MS metabolomic data processing methods: A comparative investigation of normalisation, missing value imputation, transformation and scaling. *Metabolomics* 12 (5), 1–14. Available at: <https://doi.org/10.1007/s11306-016-1030-9>.
- Doerr, A., 2014. DIA mass spectrometry. *Nature Methods* 12 (1), 35. Available at: <https://doi.org/10.1038/nmeth.3234>.
- Douglas, D.J., 2009. Linear quadrupoles in mass spectrometry. *Mass Spectrometry Reviews*, 937–960. Available at: <https://doi.org/10.1002/mas>.
- Fenn, J.B., Mann, M., Meng, C.K.A.I., Wong, S.F., Whitehouse, C.M., 1989. Electrospray ionization for mass spectrometry of large biomolecules. *Science* 246 (4926), 64–71.
- Fiehn, O., Robertson, A.D., Griffin, A.J., et al., 2007. The metabolomics standards initiative (MSI). *Metabolomics* 3, 175–178. Available at: <https://doi.org/10.1007/s11306-007-0070-6>.
- Gaskell, S.J., 1997. Special feature: Electrospray: Principles and practice. *Journal of Mass Spectrometry* 32, 677–688.
- Gillet, L.C., Navarro, P., Tate, S., et al., 2012. Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis. *Molecular & Cellular Proteomics* 11 (6), (O111.016717) Available at: <https://doi.org/10.1074/mcp.O111.016717>.
- Gowda, H., Ivanisevic, J., Johnson, C.H., et al., 2014. Interactive XCMS online: Simplifying advanced metabolomic data processing and subsequent statistical analyses. *Analytical Chemistry* 86 (14), Available at: <https://doi.org/10.1021/ac500734c>.
- Green, J.M., 1996. Analytical method validation. *Analytical Chemistry* 305A.
- Griffin, J.L., Atherton, H., Shockey, J., Atzori, L., 2011. Metabolomics as a tool for cardiac research. *Nature Reviews Cardiology* 8 (11), 630–643. Available at: <https://doi.org/10.1038/nrcardio.2011.138>.
- Griffiths, W.J., Koal, T., Wang, Y., et al., 2010. Targeted metabolomics for biomarker discovery. *Angewandte Chemie International Edition*. Available at: <https://doi.org/10.1002/anie.200905579>.
- Hartigan, J., Wong, M., 1979. A K-means clustering algorithm. *Journal of the Royal Statistical Society Series C (Applied Statistics)* 28 (1), 100–108. Available at: <https://doi.org/10.2307/2346830>.
- Horai, H., Arita, M., Kanaya, S., et al., 2010. MassBank: A public repository for sharing mass spectral data for life sciences. *Journal of Mass Spectrometry* 45 (7), 703–714. Available at: <https://doi.org/10.1002/jms.1777>.
- Hrydziusko, O., Viant, M.R., 2012. Missing values in mass spectrometry based metabolomics: An undervalued step in the data processing pipeline. *Metabolomics* 8, 161–174. Available at: <https://doi.org/10.1007/s11306-011-0366-4>.
- Jewison, T., Su, Y., Disfany, F.M., et al., 2014. SMPDB 2.0: Big improvements to the small molecule pathway database. *Nucleic Acids Research* 42 (D1), 478–484. Available at: <https://doi.org/10.1093/nar/gkt1067>.
- Johnson, C.H., Ivanisevic, J., Siuzdak, G., 2016. Metabolomics: Beyond biomarkers and towards mechanisms. *Nature Reviews Molecular Cell Biology* 17 (7), Available at: <https://doi.org/10.1038/nrm.2016.25>.
- Johnson, S.C., 1967. Hierarchical clustering schemes. *Psychometrika* 32, 241–254.
- Jolliffe, I.T., 2002. *Principal Component Analysis*, second ed. Springer.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K., 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research* 45 (D1), D353–D361. Available at: <https://doi.org/10.1093/nar/gkw1092>.
- Kell, D.B., Brown, M., Davey, H.M., et al., 2005. Metabolic footprinting and systems biology: The medium is the message. *Nature Reviews* 3, 557. Available at: <https://doi.org/10.1038/nrmicro1177>.
- Kessner, D., Chambers, M., Burke, R., Agus, D., Mallick, P., 2008. ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* 24 (21), 2534–2536. Available at: <https://doi.org/10.1093/bioinformatics/btn323>.
- Kind, T., Tsugawa, H., Cajka, T., et al., 2017. Identification of small molecules using accurate mass MS/MS search. *Mass Spectrometry Reviews*. Available at: <https://doi.org/10.1002/mas.21535>.
- Koal, T., Deigner, H.-P., 2010. Challenges in mass spectrometry based targeted metabolomics. *Current Molecular Medicine* 10, 216–226. Available at: <https://doi.org/10.2174/156652410790963312>.
- Kohonen, T., 1990. The self-organizing map. *Proceedings of the IEEE* 78 (9), 1464–1480. Available at: <https://doi.org/10.1109/5.58325>.
- Krumsiek, J., Bartel, J., Theis, F.J., 2016. Computational approaches for systems metabolomics. *Current Opinion in Biotechnology*. Available at: <https://doi.org/10.1016/j.copbio.2016.04.009>.
- Lin, S.M., Zhu, L., Winter, A.Q., Sasinowski, M., Kibbe, W.A., 2005. What is mzXML good for? *Expert Review of Proteomics* 2 (6), 839–845. Available at: <https://doi.org/10.1586/14789450.2.6.839>.
- Mann, M., Hendrickson, R.C., Pandey, A., 2001. Analysis of proteins and proteomes by mass spectrometry. *Annual Reviews in Biochemistry* 70, 437–473. Available at: <https://doi.org/10.1146/annurev.biochem.70.1.437>.
- March, R.E., 2008. Quadrupole ion traps. *Mass Spectrometry Reviews*, 961–989. Available at: <https://doi.org/10.1002/mas>.
- Marshall, A.G., Hendrickson, C.L., Jackson, G.S., 1998. Fourier transform ion cyclotron resonance mass spectrometry: A primer. *Mass Spectrometry Reviews* 17, 1–35. Available at: [https://doi.org/10.1002/\(SICI\)1098-2787\(1998\)17:1<1::AID-MAS1.3.0.CO;2-K](https://doi.org/10.1002/(SICI)1098-2787(1998)17:1<1::AID-MAS1.3.0.CO;2-K).
- Munger, J., Bennett, B.D., Parikh, A., et al., 2008. Systems-level metabolic flux profiling identifies fatty acid synthesis as a target for antiviral therapy. *Nature Biotechnology* 26 (10), 1179–1186. Available at: <https://doi.org/10.1038/nbt.1500>.
- Munson, M.S.B., Field, F.H., 1966. Chemical ionization mass spectrometry. I. General introduction. *Journal of the American Chemical Society* 88 (12), 2621–2630. Available at: <https://doi.org/10.1021/ja00964a001>.
- Oliver, S.G., 2002. Functional genomics: Lessons from yeast. *Philosophical Transactions of the Royal Society B: Biological Sciences* 357 (1417), 17–23. Available at: <https://doi.org/10.1098/rstb.2001.1049>.
- Orchard, S., Montechi-Palazzi, L., Deutsch, E.W., et al., 2007. Five years of progress in the standardization of proteomics data 4th annual spring workshop of the HUPO-proteomics standards initiative – April 23–25, 2007. Ecole Nationale Supérieure (ENS), Lyon, France. *Proteomics* 7 (19), 3436–3440. Available at: <https://doi.org/10.1002/pmic.200700658>.
- Pedrioli, P.G.A., Eng, J.K., Hubley, R., et al., 2004. A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology* 22 (11), 1459–1466. Available at: <https://doi.org/10.1038/nbt1031>.
- Pence, H.E., Williams, A., 2010. Chempider: An online chemical information resource. *Journal of Chemical Education* 87 (11), 1123–1124. Available at: <https://doi.org/10.1021/ed100697w>.

- Plumb, R.S., Johnson, K.A., Rainville, P., *et al.*, 2006. UPLC/MSE: A new approach for generating molecular fragment information for biomarker structure elucidation. *Rapid Communications in Mass Spectrometry* 20 (13), 1989–1994. Available at: <https://doi.org/10.1002/rcm.2550>.
- Raterink, R.J., Lindenburg, P.W., Vreeken, R.J., Ramautar, R., Hankemeier, T., 2014. Recent developments in sample-pretreatment techniques for mass spectrometry-based metabolomics. *Trends in Analytical Chemistry*. Available at: <https://doi.org/10.1016/j.trac.2014.06.003>.
- Roberts, L.D., Souza, A.L., Gerszten, R.E., Clish, C.B., 2012. Targeted metabolomics. *Current Protocols in Molecular Biology* 1 (Suppl. 98), Available at: <https://doi.org/10.1002/0471142727.mb3002s98>.
- Rocca-Serra, P., Salek, R.M., Arita, M., *et al.*, 2016. Data standards can boost metabolomics research, and if there is a will, there is a way. *Metabolomics* 12 (1), 1–13. Available at: <https://doi.org/10.1007/s11306-015-0879-3>.
- Ruttkies, C., Schymanski, E.L., Wolf, S., Hollender, J., Neumann, S., 2016. MetFrag relaunched: Incorporating strategies beyond in silico fragmentation. *Journal of Cheminformatics* 8 (1), Available at: <https://doi.org/10.1186/s13321-016-0115-9>.
- Salek, R.M., Neumann, S., Schober, D., *et al.*, 2015. Coordination of Standards in Metabolomics (COSMOS): Facilitating integrated metabolomics data access. *Metabolomics* 11 (6), Available at: <https://doi.org/10.1007/s11306-015-0810-y>.
- Smith, C.A., Want, E.J., O'Maille, G., Abagyan, R., Siuzdak, G., 2006. XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical Chemistry* 78 (3), 779–787. Available at: <https://doi.org/10.1021/ac051437y>.
- Smith, Colin A., O'Maille, Grace, Want, Elizabeth J., *et al.*, 2005. A metabolite mass spectral database. *Therapeutic Drug Monitoring* 27 (6), 747–751.
- Smith, R., Ventura, D., Prince, J.T., 2013. LC-MS alignment in theory and practice: A comprehensive algorithmic review. *Briefings in Bioinformatics* 16 (1), 104–117. Available at: <https://doi.org/10.1093/bib/bbt080>.
- Sud, M., Fahy, E., Cotter, D., *et al.*, 2016. Metabolomics Workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools. *Nucleic Acids Research* 44 (D1), Available at: <https://doi.org/10.1093/nar/gkv1042>.
- Sugimoto, M., Kawakami, M., Robert, M., Soga, T., 2012. Bioinformatics tools for mass spectroscopy-based metabolomic data processing and analysis. *Current Bioinformatics* 7, 96–108.
- Tautenhahn, R., Bottcher, C., Neumann, S., 2008. Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics* 9, 1–16. Available at: <https://doi.org/10.1186/1471-2105-9-504>.
- Tornasi, G., Van Den Berg, F., Andersson, C., 2004. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *Journal of Chemometrics* 18 (5), 231–241. Available at: <https://doi.org/10.1002/cem.859>.
- Tsugawa, H., Caijka, T., Kind, T., *et al.*, 2015. MS-DIAL: Data-independent MS/MS deconvolution for comprehensive metabolome analysis. *Nature Methods* 12 (6), Available at: <https://doi.org/10.1038/nmeth.3393>.
- Unidata, Available at: <https://www.unidata.ucar.edu>, <https://doi.org/10.5065/D6H70CW6>.
- Vaniya, A., Fiehn, O., 2015. Using fragmentation trees and mass spectral trees for identifying unknown compounds in metabolomics. *Trends in Analytical Chemistry*. Available at: <https://doi.org/10.1016/j.trac.2015.04.002>.
- Vinaixa, M., Samino, S., Saez, I., *et al.*, 2012. A guideline to univariate statistical analysis for LC/MS-based untargeted metabolomics-derived data. *Metabolites* 2 (4), 775–795. Available at: <https://doi.org/10.3390/metabo2040775>.
- Vinaixa, M., Schymanski, E.L., Neumann, S., *et al.*, 2016. Mass spectral databases for LC/MS- and GC/MS-based metabolomics: State of the field and prospects. *Trends in Analytical Chemistry*. Available at: <https://doi.org/10.1016/j.trac.2015.09.005>.
- Wang, Y., Xiao, J., Suzek, T.O., *et al.*, 2009. PubChem: A public information system for analyzing bioactivities of small molecules. *Nucleic Acids Research* 37 (Suppl. 2), S623–S633. Available at: <https://doi.org/10.1093/nar/gkp456>.
- Winter, G., Krömer, J.O., 2013. Fluxomics – Connecting 'omics analysis and phenotypes. *Environmental Microbiology* 15 (7), 1901–1916. Available at: <https://doi.org/10.1111/1462-2920.12064>.
- Wishart, D.S., Feunang, Y.D., Marcu, A., *et al.*, 2017. HMDB 4.0: The human metabolome database for 2018. *Nucleic Acids Research*. 1–10. Available at: <https://doi.org/10.1093/nar/gkx1089>.
- Worley, B., Powers, R., 2013. Multivariate analysis in metabolomics. *Current Metabolomics* 1, 92–107.
- Xia, J., Psychogios, N., Young, N., Wishart, D.S., 2009. MetaboAnalyst: A web server for metabolomic data analysis and interpretation. *Nucleic Acids Research*. Available at: <https://doi.org/10.1093/nar/gkp356>.
- Xia, J., Sinelnikov, I.V., Han, B., Wishart, D.S., 2015. MetaboAnalyst 3.0-making metabolomics more meaningful. *Nucleic Acids Research* 43 (W1), Available at: <https://doi.org/10.1093/nar/gkv380>.
- Yost, R.A., Fetterolf, D.D., 1983. Tandem mass spectrometry (MS/MS) instrumentation. *Mass Spectrometry Reviews* 2 (1), 1–45. Available at: <https://doi.org/10.1002/mas.1280020102>.
- Zubarev, R.A., Makarov, A., 2013. Orbitrap mass spectrometry. *Analytical Chemistry* 85 (11), 5288–5296. Available at: <https://doi.org/10.1021/ac4001223>.

Further Reading

- Barnes, S., Benton, H.P., Casazza, K., *et al.*, 2016. Training in metabolomics research. II. Processing and statistical analysis of metabolomics data, metabolite identification, pathway analysis, applications of metabolomics and its future. *Journal of Mass Spectrometry*. Available at: <https://doi.org/10.1002/jms.3780>.
- Beger, R.D., Dunn, W., Schmidt, M.A., *et al.*, 2016. Metabolomics enables precision medicine: “A white paper, community perspective.” *Metabolomics* 12 (10), Available at: <https://doi.org/10.1007/s11306-016-1094-6>.
- Bouatra, S., Aziat, F., Mandal, R., *et al.*, 2013. The human urine metabolome. *PLOS ONE* 8 (9), Available at: <https://doi.org/10.1371/journal.pone.0073076>.
- Coble, J.B., Fraga, C.G., 2014. Comparative evaluation of preprocessing freeware on chromatography/mass spectrometry data for signature discovery. *Journal of Chromatography A* 1358. Available at: <https://doi.org/10.1016/j.chroma.2014.06.100>.
- Dunn, W.B., Lin, W., Broadhurst, D., *et al.*, 2014. Molecular phenotyping of a UK population: Defining the human serum metabolome. *Metabolomics* 11 (1), 9–26. Available at: <https://doi.org/10.1007/s11306-014-0707-1>.
- Engskog, M.K.R., Haglöf, J., Arvidsson, T., Pettersson, C., 2016. LC–MS based global metabolite profiling: The necessity of high data quality. *Metabolomics*. Available at: <https://doi.org/10.1007/s11306-016-1058-x>.
- Gates, S.C., Sweeley, C.C., 1978. Quantitative metabolic profiling based on gas chromatography. *Clinical Chemistry* 24 (10), 1663–1673.
- Gorrochategui, E., Jaumot, J., Lacorte, S., Tauler, R., 2016. Data analysis strategies for targeted and untargeted LC-MS metabolomic studies: Overview and workflow. *Trends in Analytical Chemistry* 82, 425–442. Available at: <https://doi.org/10.1016/j.trac.2016.07.004>.
- Griffin, J.L., Shockcor, J.P., 2004. Metabolic profiles of cancer cells. *Nature Reviews Cancer* 4 (7), 551–561. Available at: <https://doi.org/10.1038/nrc1390>.
- Huan, T., Forsberg, E.M., Rinehart, D., *et al.*, 2017. Systems biology guided by XCMS Online metabolomics. *Nature Methods* 14 (5), 461–462. Available at: <https://doi.org/10.1038/nmeth.4260>.
- Johnson, C.H., Ivanisevic, J., Benton, H.P., Siuzdak, G., 2015. Bioinformatics: The next frontier of metabolomics. *Analytical Chemistry*. Available at: <https://doi.org/10.1021/ac5040693>.
- Johnson, C.H., Ivanisevic, J., Siuzdak, G., 2016. Metabolomics: Beyond biomarkers and towards mechanisms. *Nature Reviews Molecular Cell Biology* 17 (7), Available at: <https://doi.org/10.1038/nrm.2016.25>.
- Krumsiek, J., Theis, F.J., 2013. Statistical methods for the analysis of high-throughput metabolomics data. *Computational and Structural Biotechnology Journal* 4 (5).

- Mahieu, N.G., Genenbacher, J.L., Patti, G.J., 2016. A roadmap for the XCMS family of software solutions in metabolomics. *Current Opinion in Chemical Biology*. Available at: <https://doi.org/10.1016/j.cbpa.2015.11.009>.
- Martin, J.C., Mailliot, M., Mazerolles, G., Verdu, A., *et al.*, 2015. Can we trust untargeted metabolomics? Results of the metabo-ring initiative, a large-scale, multi-instrument inter-laboratory study. *Metabolomics* 11 (4), 807–821. Available at: <https://doi.org/10.1007/s11306-014-0740-0>.
- Misra, B.B., van der Hooft, J.J.J., 2016. Updates in metabolomics tools and resources: 2014–2015. *Electrophoresis*. Available at: <https://doi.org/10.1002/elps.201500417>.
- Pluskal, T., Castillo, S., Villar-briones, A., Ore, M., 2010. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics* 11 (395).
- Ren, S., Hinzman, A.A., Kang, E.L., Szczesniak, R.D., Lu, L.J., 2015. Computational and statistical analysis of metabolomics data. *Metabolomics* 11. Available at: <https://doi.org/10.1007/s11306-015-0823-6>.
- Rocca-Serra, P., Salek, R.M., Arita, M., *et al.*, 2016. Data standards can boost metabolomics research, and if there is a will, there is a way. *Metabolomics* 12 (1), 1–13. Available at: <https://doi.org/10.1007/s11306-015-0879-3>.
- Spicer, R., Salek, R.M., Moreno, P., Cañueto, D., Steinbeck, C., 2017. Navigating freely-available software tools for metabolomics analysis. *Metabolomics*. Available at: <https://doi.org/10.1007/s11306-017-1242-7>.
- Steuer, R., Morgenthal, K., Weckwerth, W., Selbig, J., 2007. A gentle guide to the analysis of metabolomic data. *Methods in Molecular Biology* 358, 105–128.
- Sugimoto, M., Kawakami, M., Robert, M., Soga, T., 2012. Bioinformatics tools for mass spectroscopy-based metabolomic data processing and analysis. *Current Bioinformatics*, 96–108.
- Viant, M.R., Kurland, I.J., Jones, M.R., Dunn, W.B., 2017. How close are we to complete annotation of metabolomes? *Current Opinion in Chemical Biology* 36, 64–69. Available at: <https://doi.org/10.1016/j.cbpa.2017.01.001>.
- Vinaixa, M., Schymanski, E.L., Neumann, S., *et al.*, 2016. Mass spectral databases for LC/MS- and GC/MS-based metabolomics: State of the field and prospects. *Trends in Analytical Chemistry*. Available at: <https://doi.org/10.1016/j.trac.2015.09.005>.
- Wu, Y., Li, L., 2016. Sample normalization methods in quantitative metabolomics. *Journal of Chromatography A*. Available at: <https://doi.org/10.1016/j.chroma.2015.12.007>.
- Xia, J., Broadhurst, D.I., Wilson, M., Wishart, D.S., 2013. Translational biomarker discovery in clinical metabolomics: An introductory tutorial. *Metabolomics* 9 (2), Available at: <https://doi.org/10.1007/s11306-012-0482-9>.
- Xia, J., Wishart, D.S., 2016. Using MetaboAnalyst 3.0 for comprehensive metabolomics data analysis. *Current Protocols in Bioinformatics*. Available at: <https://doi.org/10.1002/cpbi.11>.
- Yi, L., Dong, N., Yun, Y., *et al.*, 2016. Chemometric methods in data processing of mass spectrometry-based metabolomics: A review. *Analytica Chimica Acta* 914, 17–34. Available at: <https://doi.org/10.1016/j.aca.2016.02.001>.

Relevant Websites

<http://metabolomicssociety.org/resources/metabolomics-software>
Metabolomics Software
Metabolomics Society.

<http://metabolomicssociety.org/resources/metabolomics-standards>
Metabolomics Standards
Metabolomics Society.

<http://metabolomicssociety.org/resources/metabolomics-databases>
Metabolomics Society: Databases.

<http://www.metaboanalyst.ca/>
MetaboAnalyst.

<https://mzmine.github.io/>
MZmine 2.

<https://xcmsonline.scripps.edu/>
XCMS Online
The Scripps Research Institute.

Biographical Sketch



Doctor Russell Pickford was first introduced to metabolite mass spectrometry during his undergraduate degree in 1999 (Sheffield University). He then went on to study an MSc in Analytical Chemistry (Warwick University) and a PhD in biological mass spectrometry (University of Manchester and University of York). Following this he moved to the Bioanalytical Mass Spectrometry Facility at the University of New South Wales in 2003. He is responsible for the 'small molecule' mass spectrometric analysis team. He enjoys work on a wide range of projects including small molecule quantification and characterisation, metabolomics and lipidomics.

Metabolic Profiling

Joram M Posma, Imperial College London, London, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

^{13}C	Carbon-13	MHz	Megahertz
^1H	Hydrogen, proton	MS	Mass spectrometry
ANOVA	Analysis of variance	MS/MS	Tandem mass spectrometry
BMRB	Bio-magnetic resonance bank	MWAS	Metabolome-wide association study
COSY	Correlation spectroscopy	n	Number of samples
δ	Chemical shift	NMR	Nuclear magnetic resonance
DIMS	Direct-infusion mass spectrometry	OLS	Ordinary least squares
FDR	False discovery rate	OPLS	Orthogonal projections to latent structures
FID	Free induction decay	OSC	Orthogonal signal correction
FT-ICR-MS	Fourier-transform ion cyclotron resonance mass spectrometry	p	Number of variables (predictors)
GC-MS	Gas chromatography coupled mass spectrometry	PC	Principal component
GWAS	Genome-wide association study	PCA	Principal component analysis
HMBC	Hetero-nuclear multiple bond correlation	PLS	Partial least squares
HMDB	Human metabolome database	QqQ	Triple quadrupole mass analyzer
HSQC	Hetero-nuclear single quantum coherence	QTOF	Quadrupole time-of-flight
Hz	Hertz	RF	Radio frequency
ICA	Independent component analysis	STOCSY	Statistical total correlation spectroscopy
KEGG	Kyoto encyclopaedia of genes and genomes	TOCSY	Total correlation spectroscopy
LC-MS	Liquid chromatography coupled mass spectrometry	TOF	Time-of-flight
m/z	Mass-to-charge	TSP	Trimethylsilyl- $[\text{}^2\text{H}_4]$ -propionate
		X	Data matrix
		Y	Outcome variable

Introduction: Metabolic Profiling, Phenotyping and Fingerprinting

Metabolic profiling offers a powerful manner for providing a top-down systems level overview that captures the information from both genetic (host and microbiome – metagenome) and environmental (diet, environmental factors, exposures and lifestyle) origins. It involves measuring metabolites, small molecules that are typically under 1 kDa in molecular weight, in biofluids and other body samples (e.g., tissues, cells, breath) using spectroscopic or spectrometric methods to provide knowledge of the metabolic phenotype of an individual. The metabolic phenotype is the combined contribution of genetic and environmental impacts on the metabolic state under a particular set of conditions (Nicholson, 2006). The metabolic profile associated with a particular phenotype captures information that cannot be directly obtained from a genotype, gene expressions, epigenetic changes or the individual proteome, and in addition all these operate on different time scales adding to the complexity of drawing causal inferences from observed patterns (Nicholson and Lindon, 2008) – metabolic profiling aims to combat these problems by monitoring the global outcome and phenotypic traits of all these factors (environmental exposure, physiological status, system biochemistry) as a whole for utilization in personalized medicine (Holmes *et al.*, 2008b).

There are two main starting points of modern metabolic profiling using high-throughput spectroscopic (Nuclear Magnetic Resonance (NMR)) and spectrometric (mass spectrometry (MS)) techniques. The first is ‘metabonomics’ (Nicholson *et al.*, 1999) which is the analysis of the time-related, global, dynamic metabolic response in living systems to biological stimuli – it was exemplified using NMR spectroscopy and pattern recognition methods and focussed on finding the metabolites that are different between for instance disease states using multivariate statistical analysis for the first time in 1999. The second is ‘metabolomics’ (Fiehn, 2002) which was described in 2002 as the quantification, characterization and identification of the global metabolic profile in living systems – this initially focussed on using MS to identify as many metabolites in a sample. Over the years, these initial terms have been used interchangeably for studies using MS and/or NMR, with/without multivariate statistical methods and targeted or untargeted approaches. In the same way, as time passed, many different terms were introduced for these types of analyses such as metabolic fingerprinting, metabolic foot-printing, metabolic profiling and metabolite target analysis, for example, see review in Ellis *et al.* (2007), each aimed at a specific type of analysis of the metabolites with overlap in use of these terms by researchers. The first mentions of metabolic profiling in literature date back to the 1970s, way before the current time of high-throughput analysis that characterizes the current use of the term.

A metabolic profile is the result from a chain of conditional probabilistic interactions between metabolites themselves and with other components of the biochemical network, including gene-environment interactions amongst which bacteria, fungi, yeasts, archaea, viruses and parasites (Hooper *et al.*, 2012; Nicholson *et al.*, 2012a). Specific changes in metabolic flow that are out of the systemic homeostatic control can be mapped to pathway perturbations to connect with disparate biological events that operate on different time scales and in different compartments to provide a deeper understanding of the multi-layered perturbations and disruptions in disease development and therapeutic interventions. Metabolites that are shown to be associated with a type of physiological status, both reproducibly and quantitatively, can be used as potential biomarkers for disease diagnosis, therapeutic targets or for monitoring of treatment efficacy.

Backgrounds and Fundamentals of Metabolic Profiling

Analytical Workhorses

Metabolic profiling relies on two main platforms that combine the gathering of systemic information from non- or minimally-invasive sampling of biological samples, for instance biological fluids such as blood/plasma/serum and urine, with high-throughput technologies capable to analyse thousands of samples. Both techniques have their advantages and disadvantages for the detection of metabolites, but because the biologically interesting molecules have different physicochemical properties the methods are complementary as no single platform is able to measure all metabolites in a biological sample (Bouatra *et al.*, 2013).

Nuclear magnetic resonance (NMR) spectroscopy

Metabonomics was exemplified using ^1H NMR spectroscopy in the first instance (Nicholson *et al.*, 1999). NMR spectroscopy measures specific quantum mechanical properties of the nuclei from atoms with an odd number of (protons + neutrons). The simplest example of an atom that has an intrinsic angular momentum, and thus non-zero spin, is hydrogen (^1H) which only has a single proton in the nucleus. The quantum mechanical spin can either be up or down and it is this property that is used to analyse molecules with NMR spectroscopy.

A very strong magnetic field (in MHz) forces all quantum mechanical spins of nuclei to align either parallel to the field ("spin up," lower energy state) or anti-parallel ("spin down," higher energy state). After which a broad range of radio frequency (RF) pulses are used to excite all protons and bring them in a higher energy state perpendicular to the magnetic field (90° pulse), the difference in energy between states is dependent on the magnetic field strength and on the molecule the proton is attached to, as electrons shield protons slightly from applied magnetic fields. Immediately after the pulse stops, all spins are perpendicular to the magnetic field in a higher energy state and they go back to the lower energy, or relaxation, states (aligned with the field) and they emit the absorbed radiation which induces a current in a coil that is wrapped around the sample tube. The signal, known as the free induction decay (FID), contains all the different observed frequencies in the time domain with the difference in frequencies (in Hz) known as the 'chemical shift.' In order to obtain a frequency domain NMR spectrum the FID is Fourier transformed and this results in a spectrum of peaks, the average over multiple FIDs is used to improve the signal-to-noise ratio.

Comparison of spectra in metabolic profiling is made possible by optimized pulse sequences and by the addition of an internal standard which usually is the sodium salt of trimethylsilyl- $[\text{}^2\text{H}_4]$ -propionate (TSP) in aqueous (biofluid) media (Beckonert *et al.*, 2007). The nine protons of TSP are magnetically equivalent and are maximally shielded because the carbon atoms they are attached to are attached to the most electronegative atom, silicon. This means the emitted energy of TSP is highest and therefore it is used as reference at a chemical shift (δ) of 0. To standardize over different magnetic field strengths, the δ of each signal is expressed as parts per million (ppm) – Hz per MHz.

A number of other properties can be used to differentiate different molecules from each other. First, the integral of different peaks from the same molecule are directly proportional to the number of protons. Second, nuclei have a small magnetic moment which influences the signals observed from other nuclei in the vicinity. This is called spin-spin coupling and causes the splitting of NMR peaks in specific patterns, also known as the multiplicity of a peak, and the relative intensities of the peaks in the multiplet follow Pascal's triangle. The most important of the coupling types is scalar coupling, which is the interaction of two nuclei through at most 3 chemical bonds. In a sample there are many molecules of the same compound of which the protons emit different energies when excited due to the different arrangements (up/down) of the spins. The fine structure of peaks from each metabolite is different and this is used to structurally identify metabolites. Most biological molecules have a carbon backbone, thus it is the CH bonds that are typically observed. NH and OH signals can still be observed; however they give weaker signals and are less shielded.

The advantage of NMR spectroscopy is that it requires relatively little sample preparation (Beckonert *et al.*, 2007; Dona *et al.*, 2014), the sample is not destroyed in the process of analysing it, analysis takes only a few minutes and a spectrum contains information about the structure of molecules. This makes NMR spectroscopy a good method for the high-throughput analysis of biofluid samples (Dona *et al.*, 2014) and a very reproducible method. The downside is that because many CH signals will have a similar 'molecular environment' the observed peaks/peak patterns can overlap which makes it difficult to distinguish between peaks from the same metabolite, in addition NMR spectroscopy is typically less sensitive compared to mass spectrometry.

Two-dimensional NMR spectroscopy techniques can be used for metabolite identification to uncover molecular interactions between nuclei. 2D J-resolved NMR spectroscopy separates homonuclear J-couplings along an orthogonal dimension from the

standard axis where both the chemical shifts and J-couplings appear. This has the benefit of reducing the overlap between peaks in an NMR spectrum. Other 2D experiments include the COReLation SpectroscopY (COSY) experiment, which measures correlations between nuclei separated by at most 3 chemical bonds, to study neighbouring atoms and TOverlaid Correlation SpectroscopY (TOCSY) which does the same but over longer distances to create 2D $^1\text{H} - ^1\text{H}$ spectra. Hetero-nuclear 2D experiments do the same but for $^1\text{H} - ^{13}\text{C}$ in the Hetero-nuclear Single Quantum Coherence (HSQC) spectroscopy (one bond) and Hetero-nuclear Multiple Bond Correlation (HMBC) spectroscopy (more than one bond). With the exception of J-resolved spectroscopy, ^2D NMR experiments currently take too much time to be used for profiling studies and are therefore solely used for metabolite identification purposes.

Mass Spectrometry (MS)

The description of metabolomics in 2002 (Fiehn, 2002) focussed heavily on MS approaches, including gas chromatography prefaced MS (GC-MS), liquid chromatography prefaced MS (LC-MS) and Fourier-transform ion cyclotron resonance MS (FT-ICR-MS), but also considered NMR spectroscopy. The concept of mass spectrometry is to measure the mass of a compound that is ionized, expressed as a mass-to-charge (m/z) ratio, and can determine the elemental composition and isotopic signature of a molecule. For detecting the m/z ratios there are two main types of detectors used in metabolic profiling, most discovery based research is done using time-of-flight (TOF) detectors and for analyses requiring higher mass accuracies different types of 'ion traps' are used.

For TOF detectors ions are accelerated by an electric field of a specified strength so that each identically charged ion has the same kinetic energy. Ions with large m/z ratios have lower velocities than lower m/z do, therefore the time it takes for an ion to reach the detector (typically an electron multiplier tube) is directly proportional to the m/z ratio. Quadrupole mass analyzers use four parallel metal rods of which the opposing rods are connected electrically. An RF voltage is applied between the pairs resulting in ions with small m/z ranges being able to pass through the quadrupole, whereas others will collide with the rods. This allows specific m/z to pass through and reach the detector based on the variable voltage. Quadrupoles can be followed by an electron multiplier tube detector alone or precede a TOF detector (QTOF).

The second type of detectors are ion traps and they come in many different forms; the main ones used in metabolic profiling are triple quadrupole mass analyzers (QqQ) where the second quadrupole serves as a collision cell for fragmentation of ions, FT-ICR ion traps and Orbitraps. QqQ's work the same as a normal quadrupole, however now the third quadrupole now selects the fragmented ions (from the second quadrupole) to reach the detector – this often used as method for identification of specific molecules based on their fragmentation patterns and is called tandem-MS or MS/MS. FT-ICR and Orbitraps work based on the concept of having ions go in orbit around a central point and measuring the image current generated by the orbiting ions. In FT-ICR this is achieved by using a magnetic field to force the ions to spiral and in an orbitrap ions are orbiting around a central electrode in a spiral due to electrostatic forces. The different frequencies of the image currents depend on the oscillations of ions of different m/z ratios. Fourier transforming the image current results in a mass spectrum. Different methods exist to perform tandem-MS experiments in FT-ICR and Orbitraps.

It is important to realize is that all these detectors require the molecules to be in the gas phase. In metabolic profiling aqueous samples are usually ionized using electrospray ionization, which disperses the sample to create an aerosol and applies high voltage to create either positively charged ions ("positive mode"), usually by the addition of a cation (H or Na), or negative ions ("negative mode") by loss of a hydrogen.

MS has the advantage over ^1H NMR in that it can provide information of the mass of a compound that can be used to identify a compound. Measuring all masses simultaneously in mass spectrometry can be done using direct-infusion MS (DIMS) and a spectrum can be obtained within minutes (Draper *et al.*, 2013) which makes it a good method for high-throughput analysis of biofluid samples.

However, it has some disadvantages such as reduced ionization efficiency due to the simultaneous ionization of all compounds in the sample, which may contain less volatile compounds, and the corresponding co-elution of metabolites into the mass detector (Draper *et al.*, 2013), the inability to distinguish between molecules with molecular masses that fall into the same mass-to-charge (m/z) bin and batch differences due to changed experimental conditions (Kirwan *et al.*, 2013). The former two of the disadvantages can be solved using a separation technique before ionization, such as LC, GC or capillary electrophoresis to separate compounds based on physicochemical properties. The separation step improves the identification of compounds not only because less metabolites co-elute, but also because additional information is available from the separation set (retention time) that can be used to identify metabolites based on physicochemical properties in addition to the m/z -ratio (see Fig. 1).

However, these 'hyphenated' techniques also have disadvantages, for example, the increased run-time makes high-throughput experiments less feasible, the conditions of the chromatographic separation step can change when analysing many samples which then makes the identification of important metabolites more difficult and also metabolites may ionize favourably in either positive or negative mode, therefore experiments would have to be done using both modes of ionization. Dedicated data processing packages, such as XCMS (Smith *et al.*, 2006), focus on retention time alignment, and peak detection, filtering and matching, between spectra. Recent advancements in sample preparation, automated measurements and methodologies for MS-based metabolic profiling are improving its reproducibility for the analysis of large data sets (Want *et al.*, 2010; Dunn *et al.*, 2011). Specific assays and libraries exist for different applications, such as those specifically tailored to the analysis of lipids (Fahy *et al.*, 2009; Kind *et al.*, 2013), and online platforms for data processing (Tautenhahn *et al.*, 2012b).

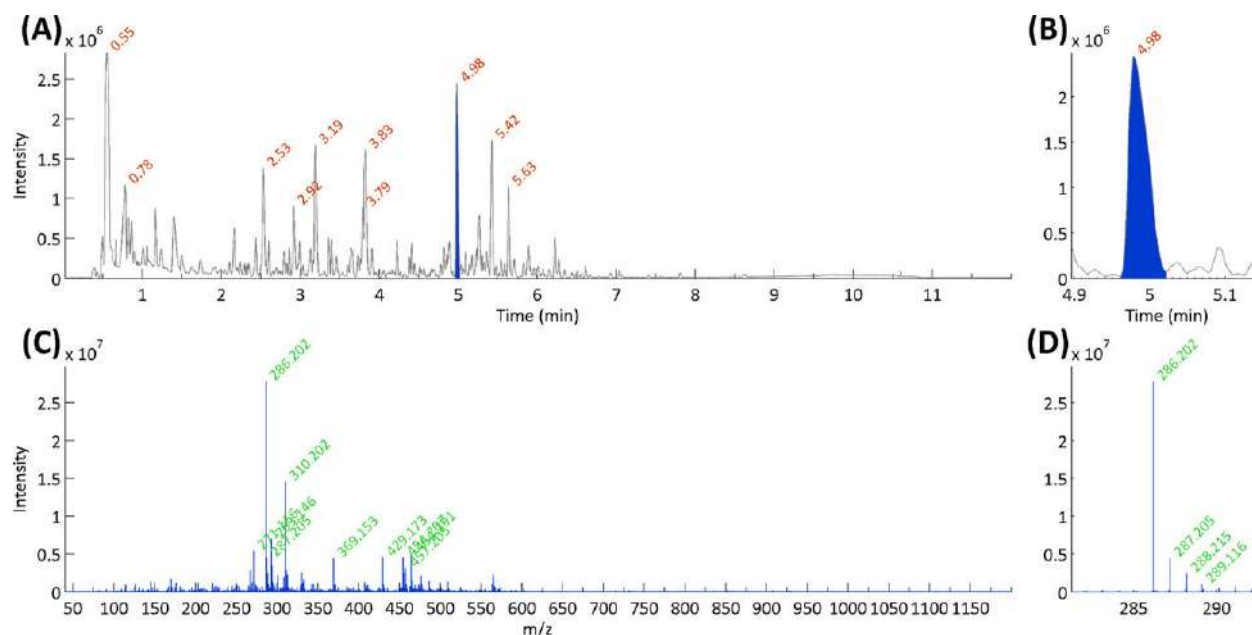


Fig. 1 Example of LC-MS global metabolic profiling analysis of a human urine sample in positive ionization mode. (A) Baseline corrected chromatogram with the ten most intense peaks labelled by their retention time and one selected peak highlighted in blue. (B) Zoom of the selected peak. (C) Mass spectrum extracted from the selected peak, the ten most intense masses are labelled by their m/z ratio. (D) Zoom of the most intense peak in the mass spectrum shows the isotope pattern for a compound with likely elemental composition $C_{15}H_{28}NO_4$.

Statistical Methods

The high-throughput analysis of samples using NMR spectroscopy and MS yields data (X) with many thousands of variables, thus for data analysis this results in data sets where the number of variables far exceeds the number of samples ($n < p$). This is problematic in standard regression methods, such as ordinary least squares (OLS), because variables will no longer be linearly independent and thus $X^T X$ might become (close to) singular and non-invertible. In metabolic profiling different methods are used to deal with the multivariate nature of the data and deal with collinearity of variables to find models that are able to predict samples well and simultaneously provide a metric of assessing the relative contribution of each variable in the predictive model.

Dimension reduction techniques

One approach used in metabolic profiling is to reduce the dimension of the data by decomposing the data into multiple orthogonal linear combinations of the original variables. Principal Component Analysis (PCA) (Hotelling, 1933), also known as Singular Value Decomposition (SVD), is used to decompose X into three parts: left singular vectors (U), a diagonal matrix with singular values (Σ) and right singular vectors (V), where the original data can be reconstructed using these three matrices using $X = U(\Sigma)V^T$ —PCA is a form of a linear-mixture model. The PC scores ($T = U(\Sigma)$) are used to look for differences between groups of samples in specific PCs by using the scores as a new coordinate system. The sum of the squared elements on the diagonal of (Σ) is equal to the total variance of X and indicates the relative importance of each component. V is used to pinpoint the variables responsible for separation by looking at the magnitude of a variable in a PC that gives information about the weight of a variable ('loading') on the transformation of a spectrum into a score. PCA is often used to 'get a feel' for the data, to see if unsupervised decomposition into latent variables based on the variance can show some signs of clustering according to the experimental design. The benefit of PCA is that components are ranked based on how much variance of X they describe, which can be used as a proxy for their importance.

However, in modern metabolic profiling PCA is rarely sufficient on its own to show differences between groups. Many different processes might impact the concentration of metabolites that may not be easily captured by linear methods. Decomposition of data in specific components according to experimental design can be performed in a supervised manner using ANOVA-based decomposition of the data into effect (data that co-varies with experimental design) and residual (data whose variation is unique and unrelated to experimental design) matrices before analysis of both of these matrices, or only the effect matrices, by PCA (Zwanenburg *et al.*, 2011).

A related approach to PCA is Independent Component Analysis (ICA) which can be used to capture non-linear, and non-Gaussian, metabolic processes from the data. It does so by decomposing X into a mixing (A) and a separating matrix (W), where A gives information about the contribution of samples in each independent component and W indicates the relative importance of variables in each component (Hyvarinen and Oja, 2000). The benefit of ICA is that it aims to find non-Gaussian, thus potentially informative, latent variables and that these are orthogonal to each other as with PCA, a related technique to ICA called Projection

Pursuit does not have the orthogonality constraint. One potential disadvantage of ICA is that it is not straightforward to assess *a priori* how many independent components are required, however the variance explained by k ICs is the same as for the first k PCs. Therefore similar methods can be used as with PCA to determine this for instance to select a cut-off for the eigenvalues or the percentage of variance of X explained by the components.

Classification algorithms

High variance is not necessarily equivalent to high information content. Therefore most metabolic profiling studies make use of supervised methods that aim to identify linear combinations of variables that co-vary with an outcome vector or response variable (Y). The most commonly used methods in metabolic profiling for multivariate regression and discriminant analysis are Partial Least Squares (PLS) (Wold, 1973) and its extension Orthogonal Projections to Latent Structures (OPLS) (Trygg and Wold, 2002).

PLS can be seen as an as a supervised extension of PCA, in that it seeks direction in the data with highest (co-)variance with the response variable, like PCA it calculates multiple orthogonal components. However, unlike PCA – or SVD specifically – these components cannot be obtained in a single step, but are found using a sequential procedure. The weights of the first PLS component are found based on the decomposition of $X^T Y$, and the first score vector is found by multiplying X with the weights, $t = Xw = X(X^T Y)$ – for a univariate Y , decomposition of $X^T Y$ is $X^T Y$ itself. The corresponding loading vector is found by multiplying X^T with normalized scores, $p = \frac{X^T t}{t^T t}$. The residual, or ‘deflated,’ X is found by subtracting tp^T from X . The same process, finding the corresponding loading and finding the residual, is performed for Y . Sequential components are obtained from the residual X and Y using the same process.

The popularity of PLS in metabolic profiling can be explained by the simple linear algebraic nature of PLS, and the low computational costs associated with it. Optimizing the number of components is performed by minimizing the cross-validation error. Including more components will improve the prediction of the model (training) set, but comes at a cost of over-fitting and possibly sub-optimal prediction of the validation (test) set. In order to assess a model’s predictive ability a second validation set must be set aside prior to data analysis to avoid biased results, this can be incorporated into a double cross-validation framework (Filzmoser *et al.*, 2009). In addition, to avoid reliance on a single model, some recent metabolic profiling studies have used Monte Carlo double cross-validation which has an important benefit in terms of providing a means of assessing variable importance (Garcia-Perez *et al.*, 2017).

Traditionally, the assessment of important variables in PLS models in metabolic profiling are based on either (a visual inspection of) the magnitude of variables in the loadings, the magnitude of the regression coefficients (by combining multiple loadings into a single regression coefficient vector), calculating the correlation between the predicted scores and the original data, or to assess the importance of a variable in terms of magnitude of each component weighted by the contribution to the model of that component (variable importance on projection) (Mehmood *et al.*, 2012). The difficulty lies in the fact that all variables contribute in a PLS model, i.e., the coefficients are all non-zero – unlike coefficients in penalized regression techniques such as the lasso and elastic net. Some approaches can borrow from univariate statistics and multiple testing corrections. For example, calculating the correlation between the predicted scores and the original data will give information about which variables are likely to be associated with the predicted scores. Despite the fact that the scores come from a multivariate model, calculating correlations of it with all original variables has the potential to give false positive associations with the predicted outcome and as well with the true outcome. A simple form of multiple testing correction on the P -values associated with each of the correlations can limit the number of false positives for correlations with predicted scores. However, outliers in the predicted scores, or even a bad predictive model, can still give rise to ‘significant’ correlations. Therefore these correlations may not explain the true outcome, as the predicted scores are used as proxy for the true outcome. In addition, reliance on a single model may not generalize well. From all individual models from a Monte Carlo model the mean regression coefficient can be used as a more robust estimate of the true coefficients. Dividing the regression coefficient by its variance can give a t -score which can be converted into a P -value and adjusted for multiple testing. Estimating the mean and variance from the same models can give biased results. Garcia-Perez *et al.* (2017) therefore opted to include a number of bootstrap resamplings of each individual model to obtain an unbiased estimate of the variance of regression coefficients in a Monte Carlo model.

The methods to determine variable importance above are all filtering methods, in that they are calculated *a posteriori*. If a subset of variables is deemed important enough, a new sparser model using only those variables can be calculated, however, it will need to account for the fact that all data that has been used are now biased. Therefore, this requires a third independent validation set that has been completely set aside from the get go to inform on the predictive ability of the new model. Most metabolic profiling studies are not set up for this to be possible.

OPLS is an extension of PLS where data is separated into variation orthogonal to Y and non-orthogonal (informative for outcome) to Y . Two main methods are used that use Orthogonal Signal Correction (OSC) (Eriksson *et al.*, 2000) to extract orthogonal variation. The first is to adjust the data matrix by removing variation from X that is orthogonal to Y . The first score from PCA is used and orthogonalized to Y , once stable the corresponding loading of the orthogonal score is calculated as the loading in PLS is calculated, and this variation is removed from X . This process can be repeated as much as deemed necessary (based on cross-validation). This process of removing non-orthogonal variation from X allows more variation non-orthogonal to Y to be captured in the first few PLS components. Analysis is carried out using regular PLS on the OSC corrected X matrix, X_{OSC} . The second method is to find the first component using PLS and adjust this *a posteriori* for non-orthogonal variation. Often the first predictive and first orthogonal components are used to visualize the class separation in a two dimensional score plot for this method.

Molecular epidemiology

Molecular epidemiology data is characterized by a wealth of metadata in addition to metabolic profiling data. Data analysis focusses on correcting for possible confounding effects that can be identified or gathered from the metadata. Typically this is done using univariate regression including possible confounding factors into a model with one variable at a time. The regression coefficients from all variables are converted to *P*-values and these are adjusted for multiple testing. Recently, these corrections have moved away from controlling the Family-Wise Error Rate, by for instance a Bonferroni correction, toward False Discovery Rates (FDRs) as some FDR methods are able to account for correlation between variables, whereas the Bonferroni correction assumes independent variables which for $n < p$ does not hold and in addition metabolites are intrinsically correlated. The Benjamini-Hochberg (1995) FDR and the *q*-value method (Storey and Tibshirani, 2003) are most commonly used, but owing to dedicated software packages (Strimmer, 2008) other methods able to deal with correlated data are now commonly used.

Bottlenecks in Metabolic Profiling

Following data analysis, important variables need to be identified in order to study them in relation to perturbations in the biological system under study. Metabolite identification is simultaneously one of the key aspects as well as a major bottleneck of metabolic profiling for both NMR spectroscopy and MS, as without it the interpretation of results is not-feasible (Wishart, 2011). Both statistical and bioinformatic, and analytical approaches are used complementarily for NMR spectroscopy and MS (Posma *et al.*, 2017).

Due to the accurate determination of the molecular mass of a metabolite, a chemical formula can be deduced from an MS experiment and both the mass and formula can be used to search in databases for chemical compounds with matching masses or formulae. A number of databases are dedicated to MS data specifically, such as METLIN (Tautenhahn *et al.*, 2012a) and MassBank (Horai *et al.*, 2010), to specific classes of compounds such as lipids, such as LIPID MAPS (Fahy *et al.*, 2009) and LipidBlast (Kind *et al.*, 2013), or tailored to specific platforms such as GC-MS and the FiehnLib (Kind *et al.*, 2009). Some of these databases include species that have not been formally identified or measured, for instance by including all possible combinations of di- and tripeptides in METLIN or inclusion of putative lipids in LIPID MAPS. MS/MS fragmentation patterns of compounds are often also included and searchable in the databases. Other databases focus on data from multiple sources other than MS, with the most widely used resource being the Human Metabolome DataBase (HMDB) (Wishart *et al.*, 2013). HMDB includes background information gathered from different sources and MS and NMR spectra of authentic chemical standards. PubChem (Kim *et al.*, 2016) and ChEBI (Hastings *et al.*, 2016) are the most widely used chemically annotated and curated chemical databases that can be used to find chemical structures and cross-reference with other databases, the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000) also contains information on the mass and structure of metabolites and links metabolites via chemical reactions and maps them in biochemical pathways. HMDB, as mentioned, contains a huge collection of NMR spectra, including 2D spectra, of authentic chemical standards. The Bio-Magnetic Resonance Bank (BMRB) (Markley *et al.*, 2008) is another online resource that contains many 1D and 2D NMR spectra of biologically relevant compounds, as well as MS spectra.

The benefit of NMR spectroscopy is that multiple peaks in a spectrum can belong to one and the same metabolite, their chemical shifts and multiplicities can give more information about the chemical structure. The fact that peaks from the same metabolite always appear in the same ratio to one another can be exploited in NMR metabolic profiling by calculating the correlation between one variable and all other variables in the data set, this has been dubbed statistical TOCSY (STOCSY) (Cloarec *et al.*, 2005) and has been one of major workhorses in metabolite identification for NMR metabolic profiling to identify the peaks with high correlations to a 'driver' peak as these are likely from the same metabolite. Different extensions have been proposed that improve on certain aspects, such as to use subset selection and multiple testing to improve the information recovery (Posma *et al.*, 2012), as well as methods to extract metabolite-metabolite interactions from the data and calculate cross-platform correlations (Robinette *et al.*, 2013). Once multiple peaks have been found to be associated with each other, these combinations of chemical shifts and multiplicities can be searched in databases such as HMDB and BMRB to find a match with known compounds and spectra (see Fig. 2).

However, not always do databases give an exact match, they might give no (exact) matches or a whole ranges of possible compounds. For the former, the only option is to perform additional experiments using 2D NMR and/or MS/MS to uncover more and complementary information about an unknown metabolite (Posma *et al.*, 2017). For the latter, experiments can be done by spiking in chemical standards to find a match, or more recently a new approach has emerged that uses metabolic network information, for example, from KEGG, to predict which match has the highest probability of being true based on a joint statistical model with other compounds that have already been identified in the sample (Cai *et al.*, 2017). This uses both the data and prior biological knowledge to narrow down the number of compounds that can be matched to a specific peak. Once a match has been found, it is confirmed using spike-ins with increased concentrations of an authentic chemical standard in a sample which are re-run using the same experiment and experimental conditions to confirm the identity.

Databases rely on the quality of the data provided to them, and in recent years a number of consortia have focussed on harmonising research in metabolic profiling, including analysis protocols, data processing, data analysis, annotation and data storage. This allows for research and data to be more effectively shared within the research community when specific reporting

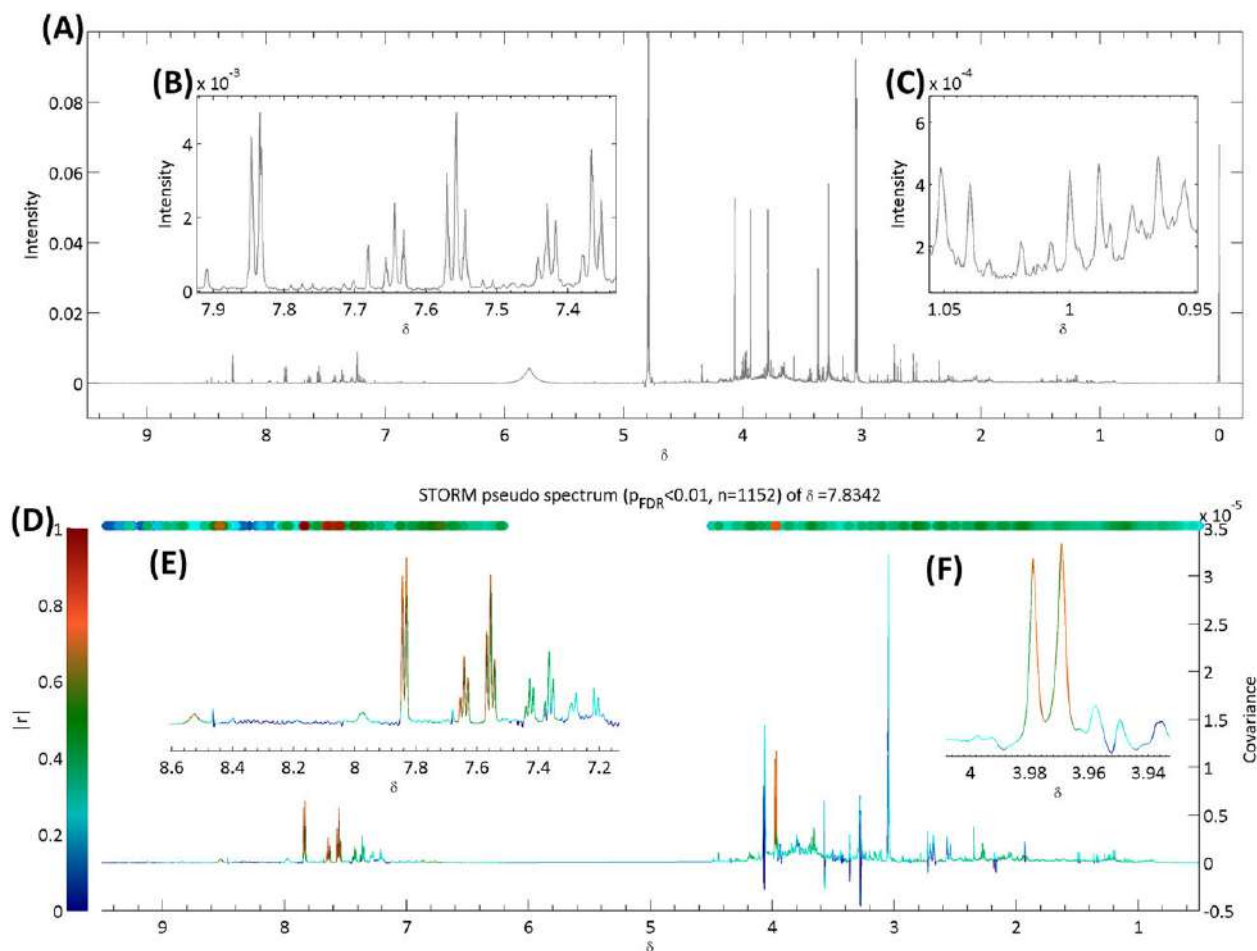


Fig. 2 Example of ^1H NMR spectroscopy metabolic profiling of a human urine sample. (A) Unprocessed 600 MHz ^1H NMR spectrum, including the internal standard TSP at δ 0 and water resonance at δ 4.74. (B) and (C) Zooms of two regions showing the splitting patterns (multiplicity) of peaks along the chemical shift axis. (D) Statistical spectroscopy analysis using STORM for metabolite identification (of δ 7.83) on a data set with 1208 urine samples shows correlation and covariance between variables, adjusted for multiple testing (p_{FDR} at 1%) and after subset selection ($n=1152$). Top bar is a 1D projection of the correlations and highly correlated variables belong to the same metabolite. (E) and (F) Zooms of two regions containing peaks from hippuric acid.

guidelines are followed. One example is the 'Coordination Of Standards in MetabOmicS' (COSMOS) (Salek *et al.*, 2015) which proposes to use specific data formats that allow simple linkage between data and metadata to facilitate improved data curation and uses the already existing and dedicated repository for metabolic profiling data MetaboLights (Haug *et al.*, 2013). More research focus has to go into automated annotation and (statistical) identification of metabolites in large data sets as these are cost-effective methods when compared to wet-lab chemistry methods that can narrow down the actual experiments that need to be performed for metabolite identification and biomarker profile validation.

Another challenge in metabolic profiling pertains to the analysis of datasets with $n < p$ and pinpointing important variables, as well as the often overlooked aspect of data processing. For instance in MS it is important to filter out noise peaks from the data prior to data analysis. One approach is to filter variables based on the number of non-zero values, i.e., measurements higher than the limit of detection. This procedure is prone to introducing bias into a model when class information is used for filtering, for instance to only select variables that are present in all samples from at least one class. Ideally the filtering step is included in the data analysis, where any noise variables do not contribute to the model. However, when methods are used for which all variables have non-zero coefficients this does not hold. Random Forests and other recursive partitioning methods are able to deal with huge numbers of variables, at relatively low computational cost, and simultaneously perform variable selection and are ideally suited for the analysis of high-dimensional MS data (Enot *et al.*, 2008). However, for NMR these methods are not ideal because of the correlated nature of NMR data, each metabolite can have multiple peaks which each have multiple data points (variables). Likewise, penalized regression methods such as the Lasso will not pinpoint all important variables, whereas the Elastic Net (Zou and Hastie, 2005) could do this, however optimizing the mixing and penalty parameters over a whole range of possible values comes at a heavy computational cost. This opens up the gap for methods, and specifically dedicated software packages, that simultaneously are able to deal with multi-collinearity, perform variable selection including

filtering out noise and are able to deal with large numbers of variables that can be used for MS- and NMR-based metabolic profiling alike.

Different Approaches in Experimental and Statistical Design

Some approaches focus on ‘global metabolic profiling’ in that they assay as many metabolites as possible, whereas others perform targeting of specific metabolites. While targeting can be more accurate experimentally, as the assay can be tailored to the measurement of a number of compounds, the resulting statistical model may be limited in the information it provides of the biological system for exactly this reason. Performing a global profiling analysis first and uncovering important metabolites which are then as a second step targeted for inclusion in a more sparse model, is potentially less biased than starting out using targeted methodology from the get go, and will provide a full(er) systems view.

Another major difference between studies is the statistical analysis of the data, specifically whether a univariate or multivariate approach is chosen. From a statistical point of view, there is no single right (or wrong) way to go about the analysis. From a more conceptual point of view, multivariate analysis make more sense considering the metabolites interact in a metabolic network themselves and can thus influence changes in disparate metabolic compartments. The most important aspect is that the results are properly validated, tested and that models are free of any bias.

Illustrative Examples of Metabolic Profiling

Metabolome-Wide Association Studies (MWAS) (Holmes *et al.*, 2008a; Elliott *et al.*, 2015) are used to find links between epidemiological and metabolic data, analogous to Genome-Wide Associations Studies (GWAS), that are collected at the same time to make inferences about the connections between disease risk factors and phenotype variation. The first study of its kind (Holmes *et al.*, 2008a) showed specific changes in the metabolic profile related to dietary, genetic, gut microbial co-metabolic and xenometabolic factors in relation to blood pressure in four different populations which provided testable hypotheses and important starting points for further research, such as investigating some blood pressure biomarker metabolites in relation to renal ion balance and possible kidney stone formation (Garcia-Perez *et al.*, 2012). The second study (Elliott *et al.*, 2015) focussed on the global profiling of urine and the relation to adiposity as a first stage using univariate statistics and multiple testing correction, and as a second stage specific additional metabolites were targeted, based on other targeted studies performed in other biofluids (Newgard *et al.*, 2009), and modelled in a penalized regression model (Elastic Net) to predict Body Mass Index. This was done on a reduced list of variables to alleviate the computational cost. The model was validated using an additional set of data of the same individuals obtained at a later time and tested in another population. The benefit of using a global profiling approach opposed to a targeted one was demonstrated by the fact that multiple areas known to be associated with obesity were all found simultaneously, such as muscle turnover, diet, amino acids and gut microbial metabolism, whereas targeted studies only identified bits and pieces.

Metabolic profiling can be used to provide information about disease aetiology, diagnosis or progression, for instance for prostate cancer progression (Sreekumar *et al.*, 2009), development of diabetes (Wang *et al.*, 2011) and prediction of survival from sepsis (Langley *et al.*, 2013). For prostate cancer it is important to have non-invasive measures of tracking progression, Sreekumar *et al.* found that one metabolite detectable in urine could be used to indicate progression and further experiments showed that knock-out of enzymes involved with sarcosine synthesis and degradation showed attenuation and increases of cancer invasion, respectively. Likewise, for diabetes and sepsis it was shown that a panel of metabolites in blood could be used to predict future diabetes and survival in sepsis patients.

Then there are the examples that highlight the capabilities of metabolic profiling to be used for personalized medicine, nutrition and care, for instance to predict drug effects (Clayton *et al.*, 2009), detect tumour margin status during surgery *in vestigio* (Balog *et al.*, 2013) and to assess adherence to dietary advice without relying on biased patient reports/records (Garcia-Perez *et al.*, 2017). Pharmacogenomics aims to stratify patients in specific groups that benefit from specific (drug) treatments, however the downside is that it does not capture the environmental variation, pharmacometabonomics aims to predict post-dose response to a drug or treatment based on pre-dose metabolic profile (Everett, 2015). For instance, paracetamol (acetaminophen) was shown to be metabolized differently, in the glucuronide rather than the sulfate form, by individuals based on the excretion of 4-cresylsulfate in urine (Clayton *et al.*, 2009) and, in addition, this observation highlighted the contribution of the gut microbiota that cannot be captured by other approaches such as pharmacogenomics. The second example details the potential of metabolic profiling to be used for the instantaneous detection of cancer tissue based on the metabolic profile of lipids in tissue. In cancer surgery it is imperative that no cancer tissue is left behind after a surgery and at the same time that as little healthy tissue is removed as possible. It is well known that cancer and healthy tissue, as well as different cancer types are characterized by different lipid profiles. The demonstration of the intelligent knife, mass spectrometry coupled to electrosurgery tools to analyse the biochemical composition of smoke after cutting tissue, has shown that on-line metabolic profiling using statistical modelling can match histopathology diagnosis of tissues perfectly and at the same time provide information on tumour biochemistry (Balog *et al.*, 2013). Finally there is public health nutrition which suffers from misreporting that confounds associations between diet and health. Metabolic profiling of urine was used to show that adherence to different diets could be assessed from the urine profile and that the metabolic profile from free-living individuals could be matched to different diets from a controlled clinical trial (Garcia-Perez *et al.*, 2017).

Discussion

Global metabolic profiling allows for an unbiased assessment of metabolites in a sample and can reveal novel and possibly unexpected perturbations in a biological system. The bottleneck point is the identification of metabolites that are deemed most important in a particular model. Specifically those that are considered ‘unknown unknowns,’ for instance compounds for which no match is found in any database or for which multiple matches remain, even after further structure elucidation using MS/MS and/or 2D NMR experiments. Identifying a compound that does not appear in any database and for which no authentic chemical standard is available is a tedious and cumbersome process requiring many different experiments (Posma *et al.*, 2017). Therefore most metabolic profiling studies focus on identifying unknowns that can be found using information in databases and validated using authentic chemical standards. Similarly, targeted profiling focuses on measurement of predefined groups of metabolites and thus may only provide a limited view. Combining a global profiling as a first stage and targeted analysis as second stage is a more powerful approach in which biomarker metabolites can be validated.

Most NMR-based metabolic profiling studies focus on global profiling, as all metabolites are measured in the same sample simultaneously. Hyphenated MS-based approaches have the benefit that they can be used both to target specific molecules based on retention-time and m/z , as well as to do global profiling. The benefit of NMR is that the sample is not destroyed in the process, which is beneficial when additional experiments are performed on the same sample for identification. Whereas MS is more sensitive than NMR which allows for more metabolites, specifically those present in low concentrations, to be profiled in a single analysis. In addition NMR can be used to analyse tissue samples in a similar way as biofluid samples are, whereas for MS this requires different techniques such as evaporative ionization MS techniques (Balog *et al.*, 2013).

In order to uncover multidimensional metabolic trajectories underlying a particular phenotype, different statistical methods can be used to establish a ‘biomarker profile’. Unlike other omics approaches, metabolites interact and are more likely to influence the concentrations of others, even those in disparate metabolic compartments. Intuitively this makes multivariate statistical methods more suited to analysing metabolic profiling data, however whether the choice is made for multi- or univariate statistical methods, the key aspect to take under consideration is to avoid introducing bias into a model, for instance by careful selection of training, test and validation sets, and utilization of robust methods for the identification of important metabolites in a model.

Future Directions for Metabolic Profiling

The future prospects of metabolic profiling do not solely relate to the information it provides, but more in its integration into global systems biology. Specifically, what it can provide for patient stratification in personalized medicine (Nicholson *et al.*, 2012b) and how it can be used in conjunction with other omic data for diagnosis, prognosis and treatment monitoring, as well as for prevention in precision medicine (Collins and Varmus, 2015). Specific attention will be focussed on understanding the environmental variability and trans-genomic interactions that together with the host genome determine the molecular phenotype (Li *et al.*, 2008; Nicholson *et al.*, 2011). This includes understanding how gut microbes are part of host metabolism and integrate these data for modelling metabolic reactions (Posma *et al.*, 2014) and metabolic signalling information (Rodriguez-Martinez *et al.*, 2017), and to better understand metabolism by studying the metabolic fluxes and regulation in a biological system (Zamboni *et al.*, 2015).

Owing to the data tsunami that the omic sciences create, the effective storage of biological samples in biobanks (Elliott *et al.*, 2008) as well as the storage of omic, clinical and other metadata (Haug *et al.*, 2013) has a major impact on future science. In the same light, reproducibility efforts (Fiehn *et al.*, 2016) of previous studies (Ward *et al.*, 2010) aim to address concerns about possible reproducibility issues in scientific research (Ioannidis, 2005). These approaches, either the effective storage of biological specimens for analysis at a later date by analytical techniques, publically available data sets that can be re-analysed using more sophisticated statistical methods and reproduction efforts of high profile studies, contribute to reducing effects of reporting bias.

Development and improvement of bioinformatic and statistical methods to make suitable for analysis of metabolic profiling data, for instance using linear-mixed effects models to correct for confounders as in GWAS (Listgarten *et al.*, 2012), or those tackling specific tasks such as expanding automated and robust data processing pipelines and tools to facilitate metabolite identification, are essential to understand the systems-level effects of metabolites and to deal with the high-dimensionality and collinearity of metabolic profiling data. In addition, the integration of metabolic profiling with other omic data such as metabolic Quantitative Trait Loci mapping and metabolic GWAS (Robinette *et al.*, 2012) or investigating the interplay between the microbiome and metabolism will play an important role in advancing health and understanding of the aetiology truly at a systems level (Johnson *et al.*, 2016).

Closing Remarks

Metabolic profiling has established itself as a field that can provide information about a biological system that are not possible using other systems biology methods. More focus on automated and robust processing pipelines, data analysis algorithms to reduce the level of bias and confounding in the model, development of new approaches to aid in the computational aspect of metabolite identification and efforts that aim to improve the reproducibility of research will increase the impact that metabolic profiling has in human health and disease.

Acknowledgement

The author has no funding information relevant to this work to report.

See also: Epidemiology: A Review. Investigating Metabolic Pathways and Networks. Mass Spectrometry-Based Metabolomic Analysis. Metabolome Analysis. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics

References

- Balog, J., Sasi-Szabo, L., Kinross, J., *et al.*, 2013. Intraoperative tissue identification using rapid evaporative ionization mass spectrometry. *Science Translational Medicine* 5, 194ra93.
- Beckonert, O., Keun, H.C., Ebbels, T.M.D., *et al.*, 2007. Metabolic profiling, metabolomic and metabonomic procedures for NMR spectroscopy of urine, plasma, serum and tissue extracts. *Nature Protocols* 2, 2692–2703.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate – A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B-Methodological* 57, 289–300.
- Bouatra, S., Aziat, F., Mandal, R., *et al.*, 2013. The human urine metabolome. *Plos One* 8, e73076.
- Cai, Q.P., Alvarez, J.A., Kang, J., Yu, T.W., 2017. Network marker selection for untargeted LC-MS metabolomics data. *Journal of Proteome Research* 16, 1261–1269.
- Clayton, T.A., Baker, D., Lindon, J.C., Everett, J.R., Nicholson, J.K., 2009. Pharmacometabonomic identification of a significant host-microbiome metabolic interaction affecting human drug metabolism. *Proceedings of the National Academy of Sciences of the United States of America* 106, 14728–14733.
- Cloarec, O., Dumas, M.E., Craig, A., *et al.*, 2005. Statistical total correlation spectroscopy: An exploratory approach for latent biomarker identification from metabolic H-1 NMR data sets. *Analytical Chemistry* 77, 1282–1289.
- Collins, F.S., Varmus, H., 2015. A new initiative on precision medicine. *New England Journal of Medicine* 372, 793–795.
- Dona, A.C., Jimenez, B., Schafer, H., *et al.*, 2014. Precision high-throughput proton NMR spectroscopy of human urine, serum, and plasma for large-scale metabolic phenotyping. *Anal Chem* 86, 9887–9894.
- Draper, J., Lloyd, A.J., Goodacre, R., Beckmann, M., 2013. Flow infusion electrospray ionisation mass spectrometry for high throughput, non-targeted metabolite fingerprinting: A review. *Metabolomics* 9, S4–S29.
- Dunn, W.B., Broadhurst, D., Begley, P., *et al.*, 2011. Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nature Protocols* 6, 1060–1083.
- Elliott, P., Peakman, T.C., UK Biobank Consortium, 2008. The UK Biobank sample handling and storage protocol for the collection, processing and archiving of human blood and urine. *International Journal of Epidemiology* 37, 234–244.
- Elliott, P., Pasma, J.M., Chan, Q., *et al.*, 2015. Urinary metabolic signatures of human adiposity. *Science Translational Medicine* 7, 285ra62.
- Ellis, D.I., Dunn, W.B., Griffin, J.L., Allwood, J.W., Goodacre, R., 2007. Metabolic fingerprinting as a diagnostic tool. *Pharmacogenomics* 8, 1243–1266.
- Enot, D.P., Lin, W., Beckmann, M., *et al.*, 2008. Preprocessing, classification modeling and feature selection using flow injection electrospray mass spectrometry metabolite fingerprint data. *Nature Protocols* 3, 446–470.
- Eriksson, L., Trygg, J., Johansson, E., Bro, R., Wold, S., 2000. Orthogonal signal correction, wavelet analysis, and multivariate calibration of complicated process fluorescence data. *Analytica Chimica Acta* 420, 181–195.
- Everett, J.R., 2015. Pharmacometabonomics in humans: A new tool for personalized medicine. *Pharmacogenomics* 16, 737–754.
- Fahy, E., Subramaniam, S., Murphy, R.C., *et al.*, 2009. Update of the LIPID MAPS comprehensive classification system for lipids. *J Lipid Res* 50, S9–S14.
- Fiehn, O., 2002. Metabolomics – The link between genotypes and phenotypes. *Plant Mol Biol* 48, 155–171.
- Fiehn, O., Showalter, M.R., Schaner-Tooley, C.E., 2016. Registered Report: The common Feature of Leukemia-Associated IDH1 and IDH2 Mutations is a Neomorphic Enzyme Activity Converting Alpha-Ketoglutarate to 2-Hydroxyglutarate, 5. Cambridge: Elife.
- Fitzmoser, P., Liebmann, B., Varmuza, K., 2009. Repeated double cross validation. *Journal of Chemometrics* 23, 160–171.
- Garcia-Perez, I., Pasma, J.M., Gibson, R., *et al.*, 2017. Objective assessment of dietary patterns by use of metabolic phenotyping: A randomised, controlled, crossover trial. *The Lancet Diabetes and Endocrinology* 5, 184–195.
- Garcia-Perez, I., Villaseñor, A., Wijeyesekera, A., *et al.*, 2012. Urinary metabolic phenotyping the slc26a6 (chloride-oxalate exchanger) null mouse model. *Journal of Proteome Research* 11, 4425–4435.
- Hastings, J., Owen, G., Dekker, A., *et al.*, 2016. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Res* 44, D1214–D1219.
- Haug, K., Salek, R.M., Conesa, P., *et al.*, 2013. MetaboLights – An open-access general-purpose repository for metabolomics studies and associated meta-data. *Nucleic Acids Res* 41, D781–D786.
- Holmes, E., Loo, R.L., Stampler, J., *et al.*, 2008a. Human metabolic phenotype diversity and its association with diet and blood pressure. *Nature* 453, 396–400.
- Holmes, E., Wilson, I.D., Nicholson, J.K., 2008b. Metabolic phenotyping in health and disease. *Cell* 134, 714–717.
- Hooper, L.V., Littman, D.R., Macpherson, A.J., 2012. Interactions between the microbiota and the immune system. *Science* 336, 1268–1273.
- Horai, H., Arita, M., Kanaya, S., *et al.*, 2010. MassBank: A public repository for sharing mass spectral data for life sciences. *Journal of Mass Spectrometry* 45, 703–714.
- Hottelling, H., 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24, 417–441.
- Hyvarinen, A., Oja, E., 2000. Independent component analysis: Algorithms and applications. *Neural Networks* 13, 411–430.
- Ioannidis, J.P.A., 2005. Why most published research findings are false. *PLOS Medicine* 2, 696–701.
- Johnson, C.H., Ivanisevic, J., Siuzdak, G., 2016. Metabolomics: Beyond biomarkers and towards mechanisms. *Nature Reviews Molecular Cell Biology* 17, 451–459.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* 28, 27–30.
- Kim, S., Thiessen, P.A., Bolton, E.E., *et al.*, 2016. PubChem substance and compound databases. *Nucleic Acids Res* 44, D1202–D1213.
- Kind, T., Liu, K.H., Lee, D.Y., *et al.*, 2013. LipidBlast in silico tandem mass spectrometry database for lipid identification. *Nat Methods* 10, 755–758.
- Kind, T., Wohlgemuth, G., Lee, D.Y., *et al.*, 2009. FiehnLib: Mass spectral and retention index libraries for metabolomics based on quadrupole and time-of-flight gas chromatography/mass spectrometry. *Anal Chem* 81, 10038–10048.
- Kirwan, J.A., Broadhurst, D.I., Davidson, R.L., Viant, M.R., 2013. Characterising and correcting batch variation in an automated direct infusion mass spectrometry (DIMS) metabolomics workflow. *Analytical and Bioanalytical Chemistry* 405, 5147–5157.
- Langley, R.J., Tsalik, E.L., Van Velkinburgh, J.C., *et al.*, 2013. An integrated clinico-metabolomic model improves prediction of death in sepsis. *Science Translational Medicine* 5, 195ra95.
- Li, M., Wang, B.H., Zhang, M.H., *et al.*, 2008. Symbiotic gut microbes modulate human metabolic phenotypes. *Proceedings of the National Academy of Sciences of the United States of America* 105, 2117–2122.

- Listgarten, J., Lippert, C., Kadie, C.M., *et al.*, 2012. Improved linear mixed models for genome-wide association studies. *Nat Methods* 9, 525–526.
- Markley, J.L., Ulrich, E.L., Berman, H.M., *et al.*, 2008. BioMagResBank (BMRB) as a partner in the Worldwide Protein Data Bank (wwPDB): New policies affecting biomolecular NMR depositions. *Journal of Biomolecular Nmr* 40, 153–155.
- Mehmood, T., Liland, K.H., Snipen, L., Sæbo, S., 2012. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems* 118, 62–69.
- Newgard, C.B., An, J., Bain, J.R., *et al.*, 2009. A branched-chain amino acid-related metabolic signature that differentiates obese and lean humans and contributes to insulin resistance. *Cell Metabolism* 9, 311–326.
- Nicholson, G., Rantalainen, M., Maher, A.D., *et al.*, 2011. Human metabolic profiles are stably controlled by genetic and environmental variation. *Molecular Systems Biology* 7.
- Nicholson, J.K., 2006. Global systems biology, personalized medicine and molecular epidemiology. *Molecular Systems Biology* 2, 52.
- Nicholson, J.K., Holmes, E., Kinross, J., *et al.*, 2012a. Host-gut microbiota metabolic interactions. *Science* 336, 1262–1267.
- Nicholson, J.K., Holmes, E., Kinross, J.M., *et al.*, 2012b. Metabolic phenotyping in clinical and surgical environments. *Nature* 491, 384–392.
- Nicholson, J.K., Lindon, J.C., 2008. Systems biology: Metabonomics. *Nature* 455, 1054–1056.
- Nicholson, J.K., Lindon, J.C., Holmes, E., 1999. 'Metabonomics': Understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological NMR spectroscopic data. *Xenobiotica* 29, 1181–1189.
- Posma, J.M., Garcia-Perez, I., De Iorio, M., *et al.*, 2012. Subset optimization by reference matching (STORM): An optimized statistical approach for recovery of metabolic biomarker structural information from (1)H NMR spectra of biofluids. *Analytical Chemistry* 84, 10694–10701.
- Posma, J.M., Garcia-Perez, I., Heaton, J.C., *et al.*, 2017. Integrated analytical and statistical two-dimensional spectroscopy strategy for metabolite identification: Application to dietary biomarkers. *Anal Chem* 89, 3300–3309.
- Posma, J.M., Robinette, S.L., Holmes, E., Nicholson, J.K., 2014. MetaboNetworks, an interactive Matlab-based toolbox for creating, customizing and exploring sub-networks from KEGG. *Bioinformatics* 30, 893–895.
- Robinette, S.L., Holmes, E., Nicholson, J.K., Dumas, M.E., 2012. Genetic determinants of metabolism in health and disease: From biochemical genetics to genome-wide associations. *Genome Med* 4, 30.
- Robinette, S.L., Lindon, J.C., Nicholson, J.K., 2013. Statistical spectroscopic tools for biomarker discovery and systems medicine. *Anal Chem* 85, 5297–5303.
- Rodriguez-Martinez, A., Ayala, R., Posma, J.M., *et al.*, 2017. MetaboSignal: A network-based approach for topological analysis of metabolite regulation via metabolic and signaling pathways. *Bioinformatics* 33, 773–775.
- Salek, R.M., Neumann, S., Schober, D., *et al.*, 2015. COordination of Standards in MetabOmicS (COSMOS): Facilitating integrated metabolomics data access. *Metabolomics* 11, 1587–1597.
- Smith, C.A., Want, E.J., O'Maille, G., Abagyan, R., Siuzdak, G., 2006. XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem* 78, 779–787.
- Sreekumar, A., Poisson, L.M., Rajendiran, T.M., *et al.*, 2009. Metabolomic profiles delineate potential role for sarcosine in prostate cancer progression. *Nature* 457, 910–914.
- Storey, J.D., Tibshirani, R., 2003. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America* 100, 9440–9445.
- Strimmer, K., 2008. fdrtool: A versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics* 24, 1461–1462.
- Tautenhahn, R., Cho, K., Uritboonthai, W., *et al.*, 2012a. An accelerated workflow for untargeted metabolomics using the METLIN database. *Nature Biotechnology* 30, 826–828.
- Tautenhahn, R., Patti, G.J., Rinehart, D., Siuzdak, G., 2012b. XCMS Online: A web-based platform to process untargeted metabolomic data. *Anal Chem* 84, 5035–5039.
- Trygg, J., Wold, S., 2002. Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics* 16, 119–128.
- Wang, T.J., Larson, M.G., Vasani, R.S., *et al.*, 2011. Metabolite profiles and the risk of developing diabetes. *Nature Medicine* 17, 448–453.
- Want, E.J., Wilson, I.D., Gika, H., *et al.*, 2010. Global metabolic profiling procedures for urine using UPLC-MS. *Nature Protocols* 5, 1005–1018.
- Ward, P.S., Patel, J., Wise, D.R., *et al.*, 2010. The common feature of leukemia-associated IDH1 and IDH2 mutations is a neomorphic enzyme activity converting alpha-ketoglutarate to 2-hydroxyglutarate. *Cancer Cell* 17, 225–234.
- Wishart, D.S., 2011. Advances in metabolite identification. *Bioanalysis* 3, 1769–1782.
- Wishart, D.S., Jewison, T., Guo, A.C., *et al.*, 2013. HMDB 3.0 – The Human Metabolome Database in 2013. *Nucleic Acids Res* 41, D801–7.
- Wold, H.O., 1973. Operative aspects of econometric and sociological models current developments of Fp (Fix-Point) estimation and nipals (nonlinear iterative partial least squares) modelling. *Economie Appliquee* 26, 385–421.
- Zamboni, N., Saghatelian, A., Patti, G.J., 2015. Defining the metabolome: Size, flux, and regulation. *Mol Cell* 58, 699–706.
- Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B-Statistical Methodology* 67, 301–320.
- Zwanenburg, G., Hoefsloot, H.C.J., Westerhuis, J.A., Jansen, J.J., Smilde, A.K., 2011. ANOVA-Principal component analysis and ANOVA-simultaneous component analysis: A comparison. *Journal of Chemometrics* 25, 561–567.

Further Reading

- Bouatra, S., Aziat, F., Mandal, R., *et al.*, 2013. The human urine metabolome. *PLOS ONE* 8, e73076.
- De Iorio, M., Ebbels, T.M.D., Stephens, D.A., 2008. Statistical techniques in metabolic profiling. *Handbook of Statistical Genetics*. John Wiley & Sons, Ltd.. [Chapter 11].
- Hastie, T., Tibshirani, R., Friedman, J.H., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.
- Holmes, E., Loo, R.L., Stamlor, J., *et al.*, 2008a. Human metabolic phenotype diversity and its association with diet and blood pressure. *Nature* 453, 396–400.
- Holmes, E., Wilson, I.D., Nicholson, J.K., 2008b. Metabolic phenotyping in health and disease. *Cell* 134, 714–717.
- Nicholson, J.K., Lindon, J.C., 2008. Systems biology: Metabonomics. *Nature* 455, 1054–1056.
- Nicholson, J.K., Lindon, J.C., Holmes, E., 2007. *The Handbook of Metabonomics and Metabolomics*. Amsterdam; Boston: Elsevier.
- Nicholson, J.K., Lindon, J.C., Holmes, E., 2018. *The Handbook of Metabolic Phenotyping*. Amsterdam; Boston: Elsevier.
- Salek, R.M., Neumann, S., Schober, D., *et al.*, 2015. COordination of Standards in MetabOmicS (COSMOS): Facilitating integrated metabolomics data access. *Metabolomics* 11, 1587–1597.
- Sreekumar, A., Poisson, L.M., Rajendiran, T.M., *et al.*, 2009. Metabolomic profiles delineate potential role for sarcosine in prostate cancer progression. *Nature* 457, 910–914.
- Trygg, J., Holmes, E., Lundstedt, T., 2007. Chemometrics in metabonomics. *Journal of Proteome Research* 6, 469–479.

Relevant Websites

Databases.

<http://www.bmrwisc.edu>

Bio-Magnetic Resonance Bank.

<http://www.hmdb.ca>
Human Metabolome DataBase.
<http://www.lipidmaps.org>
LIPID MAPS.
<http://metlin.scripps.edu>
METLIN.
Online software.
<http://www.metaboanalyst.ca>
MetaboAnalyst.
<http://xcmsonline.scripps.edu>
XCMS-online.

Biographical Sketch



Joram Matthias Posma's work involves the development of new algorithms and methods to maximize information recovery from omic data including multivariate regression and classification algorithms, software for the interactive and immersive visualization of metabolic reaction network data, statistical spectroscopy methods for metabolite identification, and bioinformatic and statistical workflows for the analysis of metabolic profiling data. He is first author of publications that appeared in *Science Translational Medicine*, *The Lancet Diabetes & Endocrinology*, *Bioinformatics*, *Analytical Chemistry* and *Analytical and Bioanalytical Chemistry*, and is co-author of publications in *Nature Communications*, *Hypertension*, *Bioinformatics*, *Analytical Chemistry* and *Journal of Proteome Research*. He obtained his MSc in Chemistry *cum laude* from Radboud University Nijmegen in 2011 and his PhD in Bioinformatics and Clinical Medicine Research from Imperial College London in 2014. He is a Member of the Royal Society of Chemistry.

Metabolic Models

Jean-Marc Schwartz, University of Manchester, Manchester, United Kingdom

Zita Soons, Maastricht University, Maastricht, The Netherlands

© 2019 Elsevier Inc. All rights reserved.

Introduction

Metabolism is an essential part of living organisms, constituted by the set of biochemical reactions that process nutrients and produce the molecular components needed by cells to maintain themselves and grow. Metabolic reactions are biochemical transformations but most of them are not spontaneous, they are catalysed by enzyme proteins which are specifically adapted to process a precise transformation. Metabolic reactions have been studied since the 19th century using both *in vitro* and *in vivo* analyses, and are well characterised for many species. Therefore metabolism is well suited for the development of mathematical models, which are aimed at calculating the levels of flux going through metabolic reactions and the amounts of substrates and products used by these reactions.

Different types of metabolic models can be produced, which differ in their level of complexity and in the precision of their outcomes. In the following we explain the principle of the most important approaches for metabolic modelling, with a focus on those that have shown the greatest range of uses and applications.

Kinetic Models

Principle

Kinetic models are the most flexible type of metabolic models. They generally use differential equations to represent detailed mechanisms of enzymatic reactions and enable a detailed computation of reaction fluxes and metabolite concentrations over time.

The formulation of a kinetic model of a metabolic system uses two main types of relations. The first type are mass conservation equations, which reflect the fact that matter is conserved in any type of chemical transformation (Fig. 1). For each metabolite, the mass conservation equation is written by expressing that the change in the metabolite concentration is equal to the sum of all incoming reaction fluxes minus the sum of all outgoing reaction fluxes.

The second type of relations are kinetic equations, which are generally expressed in the form of ordinary differential equations. The kinetic equation expresses the dependence of the reaction rate (flux) on the concentration of metabolites participating in the reaction and any regulators. It is worth noting that if the reaction is reversible, the flux can take positive or negative values and the positive direction can be chosen arbitrarily; if the reaction is irreversible only positive values are allowed.

Types of Kinetic Equations

There are many types of possible kinetic equations, depending on the complexity of the reaction mechanism and the desired level of precision (Cornish-Bowden, 2004). The simplest type is mass action kinetics, in which the reaction rate is directly proportional to the concentration of substrate(s). While this relation is sometimes used due to its simplicity, it is only strictly valid for elementary chemical reactions and is generally not an accurate depiction of enzymatic reaction kinetics. In practice, when all enzyme active sites are occupied the reaction rate reaches a plateau determined by the amount of available enzyme, but this property is not captured by the mass action relation.

The Michaelis-Menten relation offers a more accurate depiction of enzymatic reaction kinetics, which captures the saturation property. It is based on the assumption that the enzyme forms a complex with the substrate in a reversible process, followed by an irreversible decomposition of the complex into enzyme and product; moreover the concentration of enzyme is assumed to be much lower than the concentration of substrate. More complex forms of the Michael-Menten relation can be derived depending on

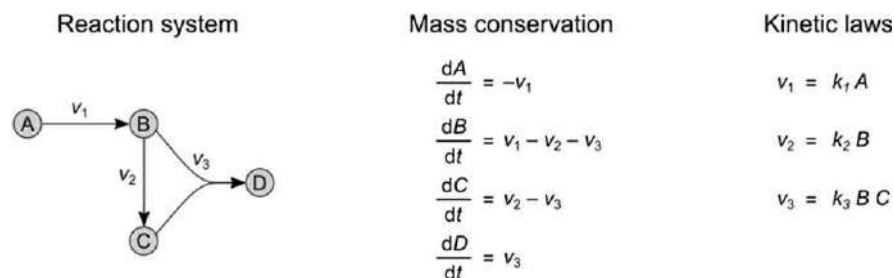


Fig. 1 From metabolic network to kinetic model.

the number of substrates and products, the order of binding of different substrates to the enzyme-substrate complex, the presence of activators and inhibitors, and whether the reaction is reversible or not.

In the case of reactions with multiple substrates and products, it can be difficult to know the exact reaction mechanism and to derive the most accurate form of Michaelis-Menten equation. For this reason, generic forms of enzyme reaction kinetics have been proposed which are applicable to any type of reaction at the expense of simplifying assumptions. One of the most versatile forms is the convenience kinetics equation, which assumes random-order and reversible binding and dissociation of substrates with the enzyme (Liebermeister and Klipp, 2006).

Simulation and Applications

When the kinetic model is built, the time course of flux and concentration values is obtained by resolving the system of differential equations. This is generally not manually feasible, except for very simple systems, because of the mathematical complexity of equations. A number of computational tools are available that are more specifically targeted at simulating biochemical systems, in particular Copasi, CellDesigner, VCell (see “Relevant Website sections”).

The construction and use of kinetic metabolic models is challenging, not only because complex equations and numerous parameters need to be determined for their construction, but also because large amounts of quantitative experimental data need to be measured for their validation. For this reason, relatively few practical applications have been reported compared to other methods, despite the fact that such models have been made for a long time. For example, several kinetic models of yeast glycolysis were proposed which were validated against measured concentrations and fluxes, and were further used to explain glycolytic oscillations in yeast (Hynne *et al.*, 2001; Williamson *et al.*, 2012). Kinetic models were developed of the red blood cell and used to predict effects of enzymopathies (Jamshidi *et al.*, 2002), and of iron metabolism in order to simulate haemochromatosis (Mitchell and Mendes, 2013). The Grape software has been developed more specifically to facilitate the construction of large metabolic models using generic rate equations and to enable parameter estimation using ‘omics’ datasets (Adiamah *et al.*, 2010).

Petri Nets

Although they are generally not classified as a kinetic modelling method, Petri nets are another technique that can be used to simulate dynamic metabolic networks. Petri nets are a concept derived from informatics which is well suited for the modelling of biochemical reaction networks (Chaouiya, 2007). They are based on a bipartite graph representation consisting of *places* and *transitions*, connected by directed arcs. In a model of metabolic pathway, places correspond to metabolites and transitions correspond to enzymatic reactions. Each place holds a certain number of *tokens*, which represents the amount or concentration of the metabolite. When the required number of tokens is present, a transition is *fired* meaning that tokens of substrates are consumed and those of products are produced, in quantities defined by the weights of the arcs connecting them to the transition. This process represents a good analogy of what occurs in a metabolic reaction, therefore many concepts derived from Petri net theory have a real-world interpretation in metabolic networks (Baldan *et al.*, 2010). For example, steady-state flux distributions are the equivalent of *T-invariants* in Petri net theory, and elementary flux modes correspond to *minimal T-invariants*. Several models of metabolic pathways using Petri nets were developed, including both qualitative and quantitative models, for example of sucrose breakdown in potato, human iron metabolism, glycolysis in liver, etc.

Stoichiometric Models

Stoichiometric Matrix

Stoichiometric models are a class of metabolic models that do not use kinetic information but only rely on the mass conservation principle. Mass conservation is embedded in the stoichiometric coefficients of metabolic reactions, which indicate the number of molecules of each substrate and product needed to complete the transformation. For example in the adenylate kinase reaction:



two molecules of adenosine diphosphate (ADP) are transformed into one molecule of adenosine triphosphate (ATP) and one molecule of adenosine monophosphate (AMP).

For computational modelling, it is generally convenient to represent stoichiometric coefficients in the form of a matrix, where each column represents a reaction and each row a metabolic compound. For example the stoichiometric matrix of the adenylate kinase reaction becomes:

$$\begin{array}{c} \text{ADP} \\ \text{ATP} \\ \text{AMP} \end{array} \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$$

It is worth noting that the stoichiometric coefficient of the substrate (ADP) is negative, indicating that it is consumed by the reaction, and that of products (ATP and AMP) is positive, indicating that they are produced by the reaction. In the case of reversible reactions, an arbitrary positive orientation needs to be chosen.

Quasi Steady State Assumption

In a metabolic network, the stoichiometric matrix can be used to express mass conservation relations in a condensed form:

$$S \cdot v = \frac{dC}{dt}$$

where S is the stoichiometric matrix, v is the vector of reaction fluxes and C is the vector of metabolite concentrations.

The quasi steady state assumption consists in considering that metabolite concentrations do not change over time, then the previous equation becomes:

$$S \cdot v = 0$$

The advantage of this approach is to reduce the system to a set of linear equations, which can be solved according to standard linear algebraic techniques. In practice, this assumption is generally valid in biological systems because metabolism adjusts fast to changes in gene expression, therefore at any time metabolic fluxes can be considered balanced (Fell, 1997). Changes in the flux distribution over time can still be modelled as a succession of steady states resulting from changing boundary conditions. Moreover, it is generally admitted that internal metabolites cannot accumulate or deplete over a long period of time in living systems (Reimers and Reimers, 2016).

Flux Balance Analysis

The linear system of equations obtained from the quasi steady state assumption is generally underdetermined, which means that there is an infinite number of possible solutions. It is possible to reduce the range of possible solutions by adding an assumption of optimality: living organisms do not behave randomly but are expected to have evolved to maximise the production of certain compounds that are essential for their survival and growth. The production of these compounds can be represented by an artificial biomass reaction whose composition reflects the overall molecular composition of the cell. In some cases a different objective is more relevant, for example ATP production may be maximised in order to optimise energetic availability, or in a metabolic engineering context one may seek to maximise the production of certain compounds of biotechnological interest. In all cases, the fluxes leading to the production of objective compounds can be represented by an *objective function* Z . The optimisation of Z is added as a constraint to the linear system of equations defined by $S \cdot v = 0$, resulting in a considerable reduction of the solution space. Additional constraints apply to metabolic fluxes, such as the maximum capacity of enzymes which sets an upper limit on the allowable flux, and the irreversibility of some reactions which forbids the flux from running in the reverse direction.

Taken together with the conservation of mass in steady state $S \cdot v = 0$, the optimisation of the objective function Z , and information about the cellular environment such as uptake rates of certain nutrients, this set of constraints define the method of Flux Balance Analysis (FBA) (Orth *et al.*, 2010). These equations can be solved by *linear programming* algorithms using computational tools such as the COBRA toolbox (Schellenberger *et al.*, 2011) (see "Relevant Websites" section).

A major advantage of FBA is that this method is fast enough to calculate flux distribution in genome-scale metabolic models containing several thousands of reactions. For this reason, FBA has been one of the most widely used methods for metabolic modelling and has led to numerous applications. Among others it has been used for determining optimal growth conditions of organisms, effects of environmental changes and mutations, understanding how certain pathogens rely on their host's metabolism, determining drug targets, designing optimised strains to produce compounds of interest for biotechnology, etc. (Bordbar *et al.*, 2014; Chapman *et al.*, 2015).

Range of Optimal Solutions

Although the FBA algorithm results in a unique value of the objective function Z , there may still be a range of flux distributions satisfying the optimal constraints of FBA. For example, flux may be distributed along several internal branches in the metabolic network that are neutral in terms of the objective function. The method of Flux Variability Analysis was developed to calculate the upper and lower flux values of all reactions that can lead to the optimal value of Z (Gudmundsson and Thiele, 2010). Besides making it possible to identify degeneracies leading to multiple solutions, this method can also be used to identify blocked reactions, whose flux is constrained to zero due to inaccuracies in the model or unavailable nutrients.

However, any combination of fluxes comprised between the bounds of FVA does not necessarily make a feasible flux distribution. Hence, other methods were developed to sample feasible flux distributions that optimise the objective function, making it possible to get a comprehensive view of the range of possible solutions (Schellenberger and Palsson, 2009).

A different possibility to reduce the multiplicity of solutions is offered by the method of parsimonious FBA (pFBA). pFBA adds to standard FBA an assumption of minimising the overall flux in the metabolic network, which amounts to minimise the amount of enzyme required. This assumption may not hold true in all cases but was shown to produce good agreement with experimental data, in particular for evolved strains where there is competition for fastest growth (Lewis *et al.*, 2010).

Genome-Scale Metabolic Models

Advances in genome sequencing have made it possible to get a good knowledge of the enzyme content of many species. This in turn has made it possible to reconstruct the full set of metabolic reactions occurring in several of these species, which are then called genome-scale metabolic models (GEMs). A GEM contains not only all known metabolic reactions with their substrates and stoichiometries, but also all the genes and proteins associated to these reactions. Some methods were developed to automate in part the reconstruction process of GEMs (Swainston *et al.*, 2011), but a high level of manual input and curation is generally needed to achieve the best quality (Feist *et al.*, 2009). A functional model must not only integrate as many as possible of the known metabolic reactions of the species, but also be able to produce all the essential cellular compounds under known experimental growth conditions.

GEMs are currently available for a range of species, including bacteria, archaea, fungi, plants and human cell types. Among the most complete and curated are the models of *E. coli* (Orth *et al.*, 2011), yeast (Heavner *et al.*, 2013) and human (Swainston *et al.*, 2016). These models were developed through several iterations over years, and the latter two involved a collaborative approach associating several research groups around the world. More recently, methods were developed that use transcriptomics or proteomics data to construct specific models for different types of human cells or tissues (Uhlén *et al.*, 2015).

GEMs have been used for numerous types of applications. They greatly facilitate the interpretation of high-throughput data such as transcriptomics, proteomics and metabolomics data, since overlaying them on a GEM can show what type of metabolic functions are active (Oberhardt *et al.*, 2009). They can be used to understand metabolic functions and their regulation in an organism, effects of environmental or nutritional changes on an organism's growth or adaptation, interactions between multiple species in an ecosystem, and define genetic manipulations in order to improve some functionalities of an organism in metabolic engineering.

Elementary Mode Analysis

Principle

Constrained based methods such as FBA are biased towards a certain biological objective. Pathway analysis has the advantage of identifying all pathways inherent to a metabolic network. A popular method for pathway analysis, Elementary Modes (EMs), identifies all admissible minimal pathways connecting substrates with biomass and products inherent to a metabolic network (Schuster *et al.*, 1999). Here, admissible means that the pathway satisfies internal mass balances and the direction constraints on reactions arising from thermodynamic conditions. For example, in the toy network shown in Fig. 2, four EMs exist, which cannot be further decomposed. EM₁ presents a cycle between metabolites B and C with no net production or consumption. EM₂ presents the formation of extracellular metabolite D from substrate Aext. EM₃ and EM₄ produce biomass from A through either reaction F6 or through reactions F7 and F9.

In addition to EMs, several other concepts such as Extreme Pathways (EPs) (Schilling *et al.*, 2000) and Generating Vectors (GVs) (Wagner and Urbanczik, 2005) are promising. EP analysis identifies the minimal set of independent pathways through the network, which is a subset of the EMs. GV are in turn a subset of the EPs. These methods for pathway analysis have been evaluated by several authors, amongst others (Klamt and Stelling, 2003). Several tools have been developed to enumerate the pathways, amongst others the widely used METATOOL (Pfeiffer *et al.*, 1999) and efntool (Terzer and Stelling, 2008).

Large-Scale Enumeration

Despite improvements in the algorithms for computation of EMs, EPs and GV, their application to larger metabolic networks has been hampered by the combinatorial explosion in the number of modes as the network size increases. The enumeration of the complete set of EMs for genome-scale networks has been infeasible, and perhaps even undesirable due to the hardly manageable number of modes that would be generated. An alternative approach is the enumeration of a subset of pathways representing the

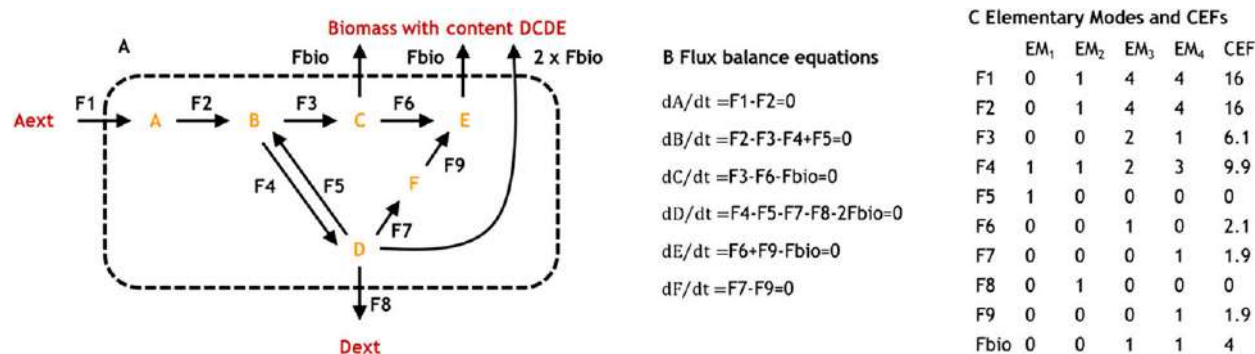


Fig. 2 Control effective flux analysis on an example network. **(A)** Toy network modified from (Rabinowitz and Vastag, 2012). **(B)** Flux balance equations **(C)** Elementary modes and CEFs.

complete system. Examples are enumeration of the shortest pathways (De Figueiredo *et al.*, 2009), enumeration of pathways including a specific target reaction (Kaleta *et al.*, 2009a,b), enumeration based on available measurements (Jungers *et al.*, 2011), elementary flux patterns (Kaleta *et al.*, 2009a,b), decomposition of the network in modules (Schuster *et al.*, 2002; Schwartz *et al.*, 2007) and random sampling (Machado *et al.*, 2012).

Applications

EMs have been used to understand the cellular metabolism through analysis of the network structure, regulations and characterization of all possible phenotypes (Çakir *et al.*, 2007; Schuster *et al.*, 1999, 2002; Stelling *et al.*, 2002; Soons *et al.*, 2013; Schwartz *et al.*, 2015). Examples of other applications of pathway analysis are the determination of minimum medium requirements (Schilling and Palsson, 2000) and the derivation of reduced kinetic models (Provost and Bastin, 2004). They also play an essential role in the development of model-based metabolic engineering strategies for strain optimisation by identification of suitable intervention targets (Hädicke and Klamt, 2010; Trinh *et al.*, 2008; Boghigian *et al.*, 2010). Of particular interest, Hädicke and Klamt (2011) proposed a gene deletion strategy based on minimal cut sets to identify a minimal set of knockouts disabling the operation of a specified set of target elementary modes, while keeping a set of desired modes.

Control Effective Fluxes and Structural Fluxes

Structural fluxes combine the advantages of both objective function-centred and pathway enumeration-centred approaches. They account for a biological objective function and are derived from the enumeration of all pathways in a given metabolic network. Structural fluxes are inspired from the concept of control effective flux (CEF) to understand changes in transcriptional regulation (Stelling *et al.*, 2002). In the original formulation of CEF, the efficiency of each elementary mode i is defined as the ratio of EM's output (the cellular objective) to the investment required to establish the EM (the sum of the absolute flux values in the EM) for a specific carbon source:

$$\varepsilon_i = \frac{Y_i^{\text{objective}}}{\sum_k |e_i^k|} \quad (1)$$

The control effective flux of each reaction k is then obtained by the weighted average of the product of mode-specific efficiencies and reaction-specific fluxes over the sum of all mode efficiencies:

$$CEF_k = \frac{1}{Y_{X/S}^{\text{max}}} \cdot \frac{\sum_i \varepsilon_i \cdot |e_i^k|}{\sum_i \varepsilon_i} \quad (2)$$

where $Y_{X/S}^{\text{max}}$ denotes the maximum yield obtained for the specified cellular objective.

In order to find a measure that can predict fluxes across mutants and for growth on multiple substrates, Soons *et al.* (2013) and Schwartz *et al.* (2015) proposed three modifications in Eqs. (1)–(2) and thereby introduced Structural Fluxes. In summary, the yield is computed towards total substrate uptake (in terms of C-moles or molecular weight) and an additional normalization was added to account for networks of different sizes.

As an example, we determined the yields, efficiencies and CEFs for the toy network in Fig. 2(C) with biomass as the cellular objective (reaction Fbio) and uptake of external metabolite A as the input (reaction F1). Both EM₃ and EM₄ produce biomass with the same yield towards the substrate uptake reaction F1 (1/4). EM₄ involves one additional enzyme (through F7 and F9) compared to EM₃ (through F6) and is hence less efficient in terms of investment of enzymes (Eq. (1)). This is reflected by a slightly higher CEF for reaction F6 compared with F7 and F9. In real life networks, these trade-offs may be for instance presented in cancer cells by a high ATP yield per substrate for the TCA cycle, but also high investment costs in the number of enzymes in comparison with aerobic glycolysis.

Future Directions

As procedures for building genome-scale metabolic models are getting well established and the number of available models is growing, new research directions aim at enriching metabolic models with additional information. Thermodynamics plays an important role for metabolic reactions, since the energy levels of substrates and products determine if a reaction is feasible or not. Several studies have attempted to integrate thermodynamic data into metabolic models in order to estimate metabolite concentrations and determine reaction directionality (Ataman and Hatzimanikatis, 2015), determine which elementary modes are feasible or not (Gerstl *et al.*, 2016) and discover reactions that favour growth at higher or lower temperature (Paget *et al.*, 2014).

Another direction to enrich metabolic models is to incorporate transcriptional regulation mechanisms. The metabolic flux is controlled by enzymes which are themselves regulated by transcription factors and other regulatory mechanisms, therefore these interactions must be taken into account to achieve better quantitative predictions. The construction of integrated transcriptional and metabolic models has started, using both kinetic and constraint-based approaches, and constitutes an important objective for future research (Imam *et al.*, 2015).

Eventually, metabolic models from different species need to be connected. In biological environments ranging from human microbiota to global ecosystems, different organisms exchange compounds with each other or compete for resources. The behaviour of each species cannot be fully determined without considering its interactions with other species, which requires moving from species-level to community-level metabolic models (Biggs *et al.*, 2015).

Closing Remarks

Metabolic models have shown broad utility for both biomedical and biotechnological applications. Different approaches can be used to model metabolic systems and each of them has specific advantages and limitations, which is why it is important to choose the best suited technique depending on the expected outcomes of the model. Metabolic reactions are not isolated but are interconnected with other cellular processes, therefore a challenge in any model is often to define suitable boundaries; this choice should always be directed by biological knowledge to include enough but no superfluous components to explain the expected biological functions and reach the desired level of precision.

See also: Biological Pathways. Cell Modeling and Simulation. Investigating Metabolic Pathways and Networks. Metabolome Analysis. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Network Models. Pathway Informatics. Two Decades of Biological Pathway Databases: Results and Challenges

References

- Adiamah, D.A., Handl, J., Schwartz, J.M., 2010. Streamlining the construction of large-scale dynamic models using generic kinetic equations. *Bioinformatics* 26, 1324–1331.
- Ataman, M., Hatzimanikatis, V., 2015. Heading in the right direction: Thermodynamics-based network analysis and pathway engineering. *Current Opinion in Biotechnology* 36, 176–182.
- Baldan, P., Cocco, N., Marin, A., Simeoni, M., 2010. Petri nets for modelling metabolic pathways: A survey. *Natural Computing* 9, 955–989.
- Biggs, M.B., Medlock, G.L., Kolling, G.L., Papin, J.A., 2015. Metabolic network modeling of microbial communities. *WIREs Systems Biology and Medicine* 7, 317–334.
- Boghigian, B.A., Shi, H., Lee, K., Pfeifer, B.A., 2010. Utilizing elementary mode analysis, pathway thermodynamics, and a genetic algorithm for metabolic flux determination and optimal metabolic network design. *BMC Systems Biology* 4.
- Bordbar, A., Monk, J.M., King, Z.A., Palsson, B.Ø., 2014. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics* 15, 107–120.
- Çakir, T., Kirdar, B., Onsan, Z.I., Ulgen, K.O., Nielsen, J., 2007. Effect of carbon source perturbations on transcriptional regulation of metabolic fluxes in *Saccharomyces cerevisiae*. *BMC Systems Biology* 1.
- Chaouiya, C., 2007. Petri net modelling of biological networks. *Briefings in Bioinformatics* 8, 210–219.
- Chapman, S.P., Paget, C.M., Johnson, G.N., Schwartz, J.M., 2015. Flux balance analysis reveals acetate metabolism modulates cyclic electron flow and alternative glycolytic pathways in *Chlamydomonas reinhardtii*. *Frontiers in Plant Science* 6, 474.
- Cornish-Bowden, A., 2004. *Fundamentals of Enzyme Kinetics*, third ed. London: Portland Press.
- Feist, A.M., Herrgård, M.J., Thiele, I., Reed, J.L., Palsson, B.Ø., 2009. Reconstruction of biochemical networks in microorganisms. *Nature Reviews Microbiology* 7, 129–143.
- Fell, D., 1997. *Understanding the Control of Metabolism*. London: Portland Press.
- De Figueiredo, L.F., Podhorski, A., Ribio, A., *et al.*, 2009. Computing the shortest elementary flux modes in genome-scale metabolic networks. *Bioinformatics* 25, 3158–3165.
- Gerstl, M.P., Jungreuthmayer, C., Müller, S., Zanghellini, J., 2016. Which sets of elementary flux modes form thermodynamically feasible flux distributions? *FEBS Journal* 283, 1782–1794.
- Gudmundsson, S., Thiele, I., 2010. Computationally efficient flux variability analysis. *BMC Bioinformatics* 11, 489.
- Hädicke, O., Klamt, 2010. CASOP: A computational approach for strain optimization aiming at high productivity. *Journal of Biotechnology* 147, 88–101.
- Hädicke, O., Klamt, S., 2011. Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic Engineering* 13, 204–213.
- Heavner, B.D., Smallbone, K., Price, N.D., Walker, L.P., 2013. Version 6 of the consensus yeast metabolic network refines biochemical coverage and improves model performance. *Database* 2013, bat059.
- Hynne, F., Danø, S., Sørensen, P.G., 2001. Full-scale model of glycolysis in *Saccharomyces cerevisiae*. *Biophysical Chemistry* 94, 121–163.
- Imam, S., Schäuble, S., Brooks, A.N., Baliga, N.S., Price, N.D., 2015. Data-driven integration of genome-scale regulatory and metabolic network models. *Frontiers in Microbiology* 6, 409.
- Jamshidi, N., Wiback, S.J., Palsson, B.Ø., 2002. In silico model-driven assessment of the effects of single nucleotide polymorphisms (SNPs) on human red blood cell metabolism. *Genome Research* 12, 1687–1692.
- Jungers, R.M., Zamorano, F., Blondel, V.D., Wouwer, A.V., 2011. Fast computation of minimal elementary decompositions of metabolic flux vectors. *Automatica* 47, 1255–1259.
- Kaleta, C., De Figueiredo, L.F., Behre, J., Schuster, S., 2009a. In: Gross, I., Neumann, S., Posch, S. (Eds.), *EFMEvovler: Computing Elementary Flux Modes in Genome-Scale Metabolic Networks*. Bonn, pp. 179–189.
- Kaleta, C., De Figueiredo, L.F., Schuster, S., 2009b. Can the whole be less than the sum of its parts? Pathway analysis in genome-scale metabolic networks using elementary flux patterns. *Genome Research* 19, 1872–1883.
- Klamt, S., Stelling, J., 2003. Two approaches for metabolic pathway analysis? *Trends in Biotechnology* 21, 64–69.
- Lewis, N.E., Hixson, K.K., Conrad, T.M., *et al.*, 2010. Omic data from evolved *E. coli* are consistent with computed optimal growth from genome-scale models. *Molecular Systems Biology* 6, 390.
- Liebermeister, W., Klipp, E., 2006. Bringing metabolic networks to life: Convenience rate law and thermodynamic constraints. *Theoretical Biology and Medical Modelling* 3, 41.
- Machado, D., Soons, Z.I.T.A., Patil, K.R., Ferreira, E.C., Rocha, I., 2012. Random sampling of elementary flux modes in large-scale metabolic networks. *Bioinformatics* 28, i515–i521.
- Mitchell, S., Mendes, P., 2013. A computational model of liver iron metabolism. *PLOS Computational Biology* 9, e1003299.
- Oberhardt, M.A., Palsson, B.Ø., Papin, J.A., 2009. Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology* 5, 320.
- Orth, J.D., Conrad, T.M., Na, J., *et al.*, 2011. A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism–2011. *Molecular Systems Biology* 7, 535.
- Orth, J.D., Thiele, I., Palsson, B.Ø., 2010. What is flux balance analysis? *Nature Biotechnology* 28, 245–248.

- Paget, C.M., Schwartz, J.M., Delneri, D., 2014. Environmental systems biology of cold-tolerant phenotype in *Saccharomyces* species adapted to grow at different temperatures. *Molecular Ecology* 23, 5241–5257.
- Pfeiffer, T., Sanchez-Valdenebro, I., Nuno, J.C., Montero, F., Schuster, S., 1999. METATOOL: For studying metabolic networks. *Bioinformatics* 15, 251–257.
- Provost, A., Bastin, G., 2004. Dynamic metabolic modelling under the balanced growth condition. *Journal of Process Control* 14, 717–728.
- Rabinowitz, J.D., Vastag, L., 2012. Teaching the design principles of metabolism. *Nature Chemical Biology* 8, 497–501.
- Reimers, A.M., Reimers, A.C., 2016. The steady-state assumption in oscillating and growing systems. *Journal of Theoretical Biology* 406, 176–186.
- Schellenberger, J., Palsson, B.O., 2009. Use of randomized sampling for analysis of metabolic networks. *Journal of Biological Chemistry* 284, 5457–5461.
- Schellenberger, J., Que, R., Fleming, R.M., *et al.*, 2011. Quantitative prediction of cellular metabolism with constraint-based models: The COBRA Toolbox v2.0. *Nature Protocols* 6, 1290–1307.
- Schilling, C.H., Letscher, D., Palsson, B.O., 2000. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology* 203, 229–248.
- Schilling, C.H., Palsson, B.O., 2000. Assessment of the metabolic capabilities of *Haemophilus influenzae* Rd through a genome-scale pathway analysis. *Journal of Theoretical Biology* 203, 249–283.
- Schuster, S., Dankdekar, T., Fell, D.A., 1999. Detection of elementary flux modes in biochemical networks: A promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology* 17, 53–60.
- Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., Dankdekar, T., 2002. Exploring the pathway structure of metabolism: Decomposition into subnetworks and application to *Mycoplasma pneumoniae*. *Bioinformatics* 18, 351–361.
- Schwartz, J.M., Barber, M., Soons, Z., 2015. Metabolic flux prediction in cancer cells with altered substrate uptake. *Biochemical Society Transactions* 43, 1177–1181.
- Schwartz, J.M., Gauguin, C., Nacher, J.C., De, D.A., Kanehisa, M., 2007. Observing metabolic functions at the genome scale. *Genome Biology* 8, R123.
- Soons, Z.I., Ferreira, E.C., Patil, K.R., Rocha, I., 2013. Identification of metabolic engineering targets through analysis of optimal and sub-optimal routes. *PLOS ONE* 8, e61648.
- Stelling, J., Klamt, S., Bettenbrock, K., Schuster, S., Gilles, E.D., 2002. Metabolic network structure determines key aspects of functionality and regulation. *Nature* 420, 190–193.
- Swainston, N., Smallbone, K., Hefzi, H., *et al.*, 2016. Recon 2.2: From reconstruction to model of human metabolism. *Metabolomics* 12, 109.
- Swainston, N., Smallbone, K., Mendes, P., Kell, D., Paton, N., 2011. The SuBliMinal toolbox: Automating steps in the reconstruction of metabolic networks. *Journal of Integrative Bioinformatics* 8, 186.
- Terzer, M., Stelling, J., 2008. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics* 24, 2229–2235.
- Trinh, C.T., Unrean, P., Sreenc, F., 2008. Minimal *Escherichia coli* cell for the most efficient production of ethanol from hexoses and pentoses. *Applied and Environmental Microbiology* 74, 3634–3643.
- Uhlén, M., Fagerberg, L., Hallström, B.M., *et al.*, 2015. Tissue-based map of the human proteome. *Science* 347, 1260419.
- Wagner, C., Urbanczik, R., 2005. The geometry of the flux cone of a metabolic network. *Biophysical Journal* 89, 3837–3845.
- Williamson, T., Adiamah, D., Schwartz, J.M., Stateva, L., 2012. Exploring the genetic control of glycolytic oscillations in *Saccharomyces cerevisiae*. *BMC Systems Biology* 6, 108.

Relevant Websites

<http://bigg.ucsd.edu/>
BiGG database.

<http://www.ebi.ac.uk/biomodels-main/>
BioModels database.

<http://celldesigner.org/>
CellDesigner modelling software.

<http://opencobra.github.io/>
COBRA toolbox for constraint-based modelling.

<http://copasi.org/>
COPASI biochemical system simulator.

<http://metabolicatlas.com/>
Human Metabolic Atlas.

<https://omictools.com/metatool-tool>
Metatool for elementary modes analysis.

<http://vcell.org/>
Virtual Cell.

Biographical Sketch

Jean-Marc Schwartz is a senior lecturer in the Faculty of Biology, Medicine and Health at the University of Manchester. He has been working for 20 years at the interface between physical and biological sciences, focusing on computational techniques for modelling biological systems. He holds a PhD in computer engineering from Laval University and has worked as a postdoctoral researcher at the Kyoto University Bioinformatics Center, which develops the KEGG genomic and metabolic database, before joining the university of Manchester. His research applies a range of computational techniques, including kinetic, logical and constraint-based modelling, to understand the functioning of biological pathways and cellular systems. Of particular interest are the development of metabolic models of human cells to understand diseases such as cancer, and of microorganisms to discover how they are affected by environmental change and develop biotechnological applications.

Zita Soons works as an assistant professor at the Department of Surgery of Maastricht University, The Netherlands. Her main topic is the integration of stable isotope labelling and MS-Imaging to understand cancer metabolism. She is a specialist in computational systems biology of human and microbial metabolism. In previous jobs, she integrated empirical data on metabolism with metabolic models ranging from biotechnological production to cancer. She received a personal grant to work as a postdoctoral researcher at the University of Minho (Portugal) in 2009–2012, which also involved a stay at the European Molecular Biology Laboratory in Heidelberg (Germany). She received a PhD degree on systems and control theory at Wageningen University/Netherlands Vaccine Institute (The Netherlands). In her current job, she also enjoys incorporating the latest advances in her field of research into teaching.

Protein Structural Bioinformatics: An Overview

M Michael Gromiha, Tokyo Institute of Technology, Tokyo, Japan

Raju Nagarajan, Indian Institute of Technology Madras, Chennai, India and Vanderbilt University Medical Center, Nashville, TN, United States

Samuel Selvaraj, Bharathidasan University, Tiruchirappalli, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteins

Proteins are one of the important biological macromolecules built from 20 amino acids. They perform several functions in living organisms such as enzymatic catalysis, transportation of ions and molecules, antibodies and regulation of cellular and physiological activities. The functions of proteins mainly depend on their three-dimensional (3D) structures. Currently, more than 130,000 structures of proteins and their complexes are deposited in Protein Data Bank, PDB ([Burley et al., 2017](#)) whereas the amino acid sequences are known for more than 80 million proteins ([The UniProt Consortium, 2017](#)). [Anfinsen \(1973\)](#) stated that the amino acid sequence of a protein contains all the information necessary to specify its 3D structure. Deciphering the 3D structure of a protein from its amino acid sequence is a longstanding goal in molecular and computational biology. Further, the structures of proteins and their complexes provide ample insights to understand inter-residue interactions, protein folding mechanisms, folding and unfolding rates, stability of protein structures, stability upon mutations, recognition mechanism of protein-protein, protein-nucleic acid and protein-ligand complexes, and structure-based drug design ([Gromiha, 2010](#)). [Fig. 1](#) illustrates the major themes covered in this review.

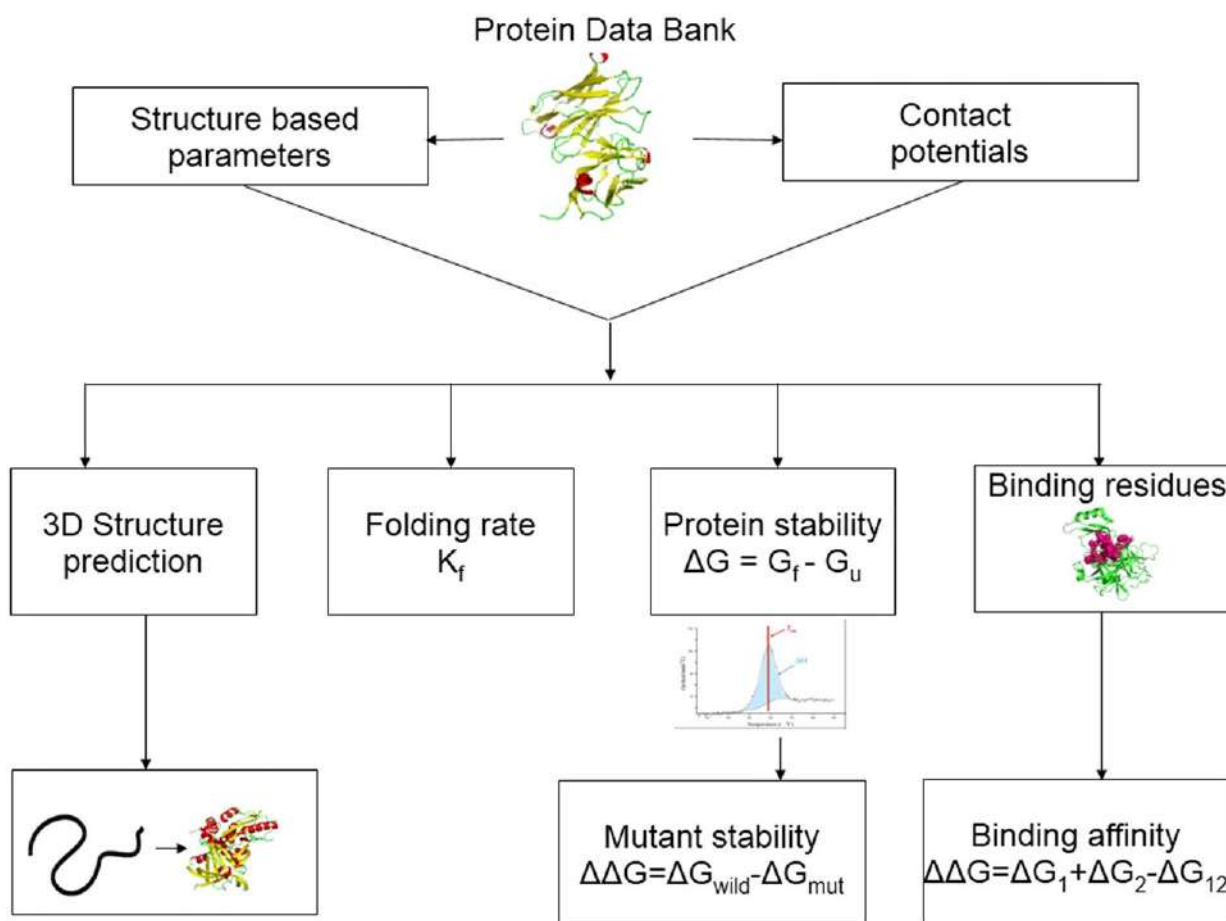


Fig. 1 Flowchart showing various aspects of computational studies on protein structures.

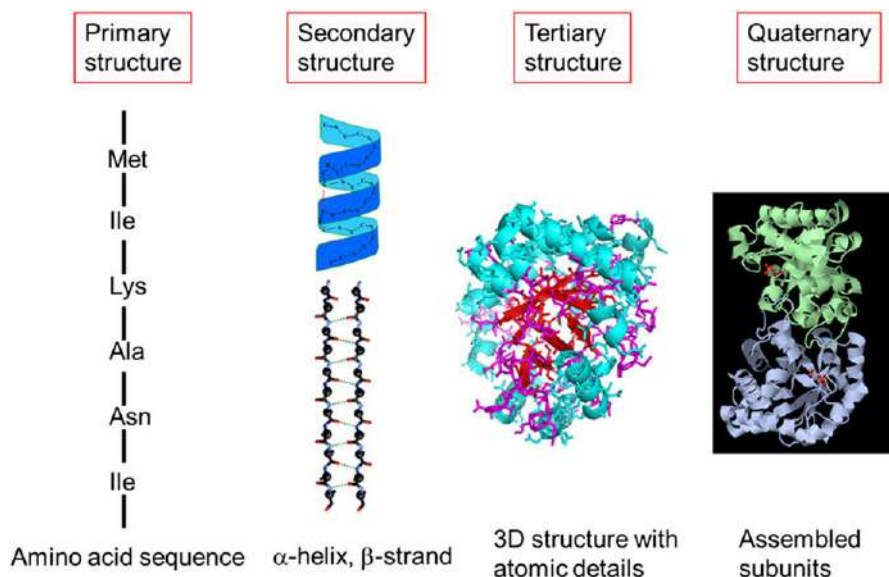


Fig. 2 For major levels of protein structures. Adopted from Gromiha, M.M., 2010. Protein Bioinformatics. Academic Press, New Delhi.

Structural Organization of Proteins

Protein structures are usually described in terms of four levels: primary, secondary, tertiary and quaternary (**Fig. 2**). Primary structure describes the linear sequence of amino acid residues in a protein. Secondary structure is the regular, recurring arrangements of adjacent amino acid residues in a polypeptide chain and major secondary structures are α -helices and β -strands.

They are formed by hydrogen bonds between amide hydrogens and carbonyl oxygens of the peptide backbone. Tertiary structure provides atomic and spatial level information of all residues in a protein. Quaternary structure refers to the association of two or more polypeptide chains into a multisubunit or an oligomeric protein.

Classification of Proteins Based on Structure and Function

Proteins fall into three major groups: fibrous, globular and membrane proteins. Fibrous proteins are insoluble in solvents and the polypeptide chains are arranged in long strands or sheets. They play important structural roles, providing external protection, support and shape. Examples of fibrous proteins include α -keratin, the major component of hair and nails, and collagen, the major protein component of tendons, skin, bones and teeth.

Globular proteins are important for their functional roles and the amino acid residues in a globular protein fold to a proper, unique three-dimensional structure. Globular proteins are classified into four structural classes, namely, all- α , all- β , $\alpha + \beta$ and α/β (**Fig. 3**). The all- α and all- β classes are dominated by α -helices and β -strands, respectively. The $\alpha + \beta$ and α/β classes contain both α -helices and β -strands. In $\alpha + \beta$ class, helices and strands usually segregate while they are mixed together in the α/β class.

Proteins that are embedded in biological membranes are called membrane proteins and they serve as receptors, transporters and signaling molecules. Membrane proteins are of two kinds: transmembrane helical proteins that span the cellular membrane with α -helices and transmembrane β -sheet proteins that traverse the outer membrane of gram negative bacteria with β -strands. In most of genomes (bacteria, archaea and eukaryota), 20–30% of the proteins are estimated to be membrane proteins (Almeida *et al.*, 2017) and are identified as targets for drug design.

Structure Databases for Proteins and Their Complexes

The Protein Data Bank (PDB) is a unique resource for experimentally determined structures of proteins and their complexes (Burley *et al.*, 2017). Utilizing the information available in PDB, several secondary databases have been developed for structural classes and architectures of proteins such as SCOP (Structural classification of proteins) (Andreeva *et al.*, 2008), CATH (Class, Architecture, Topology and Homologous superfamily) (Sillitoe *et al.*, 2015), PDBTM (Protein Data Bank of Trans Membrane proteins) (Kozma *et al.*, 2013), and protein-protein, protein-nucleic acid, protein-carbohydrate and protein-ligand complexes. **Table 1** lists the available structural databases for proteins and their complexes.

Analysis of Protein 3D Structures

The availability of protein 3D structures paves the way for researchers to derive the principles governing the folding and stability of proteins, binding sites in protein complexes and development of several structural parameters. Further, various structure-based

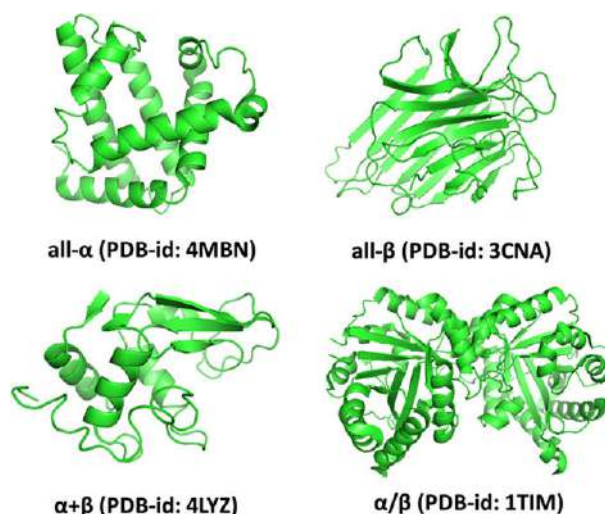


Fig. 3 Major structural classes of globular proteins.

parameters for proteins and physicochemical, energetic and conformational parameters of amino acids have been developed, using 3D structures of proteins deposited in PDB.

Assignment of Secondary Structure and Solvent Accessibility

Kabsch and Sander (1983) utilized the hydrogen bonding pattern in protein structures and assigned secondary structures to each residue in a protein. Lee and Richards (1971) proposed the concept of accessible surface area (ASA) to locate the interior seeking and exposed residues in a protein. ASA is defined as the locus of the centre of the solvent molecule as it rolls over the van der Waals surface of the protein. Generally, water is assumed to be the solvent, represented by a sphere of radius 1.4 Å. The solute molecules are represented by a set of interlocking spheres of appropriate van der Waals radii assigned to each atom and the solvent molecule is rolled along the envelope of the van der Waals surface at planes conveniently sectioned. Several methods have been proposed to compute the solvent accessibility of amino acid residues from protein structures (Gromiha and Ahmad, 2005). These programs provide numerical data for accessible surface area and hence Ahmad *et al.* (2004) developed a tool, ASAview to represent them pictorially. The ASA values for the Zif268 zinc finger protein are shown in Fig. 4 as spiral circles with different colors to visualize the exposed and interior seeking residues, based on their chemical behavior and ASA values (see Relevant Websites section).

Residue-Residue Contacts

The contacts between amino acid residues in a protein as well as with the surrounding medium mainly dictate the folding and stability of protein structures. These contacts influence all major non-covalent interactions such as hydrophobic, electrostatic, hydrogen bonding and van der Waals interactions as well as disulfide bonds (Fig. 5). Inter-residue contacts are classified as short, medium and long-range, based on the location of the contacting residues in the 3D structure as well as in the sequence (Gromiha and Selvaraj, 2004). The information about preferred contacts in protein structures has been used to understand the residue-residue co-operativity in protein folding as well as to develop contact potentials.

The most widely used approach for deriving contact potentials is computing frequencies of sequence and structure features, and converting them into free energies using the following relation

$$\Delta G = -kT \log(f) \quad (1)$$

where k is the Boltzmann's constant, f denotes frequencies and T is the temperature (Sippl, 1990). Further, contacts between amino acid residues in proteins structures reveal the importance of specific contact pairs in different structural classes and folding types of globular proteins, α -helical and β -barrel membrane proteins, mesophilic and thermophilic proteins (Gromiha, 2010).

Non-Covalent Interactions in Protein Structures

Hydrophobic interactions play a dominant role in the folding and stability of protein structures. Hydrophobicity is inversely proportional to ASA and hence, hydrophobic free energy is estimated indirectly using the computable parameter, ASA. It is computed by (Gromiha, 2010):

$$G_{hy} = \sum \Delta \sigma_i [A_i(\text{folded}) - A_i(\text{unfolded})] \quad (2)$$

where, i stands for different atom types, and $A_i(\text{folded})$ and $A_i(\text{unfolded})$ are the ASA of each atom type in the folded and

Table 1 List of databases and servers for protein structures, structure prediction, stability and interactions

<i>Name of the database/web server</i>	<i>URL</i>
<i>Structure and classification</i>	
PDB	http://www.rcsb.org/pdb/home/home.do
SCOP	http://scop.mrc-lmb.cam.ac.uk/scop/
CATH	http://www.cathdb.info/
PDBTM	http://pdbtm.enzim.hu/
<i>Structure-based computations and parameters</i>	
ASAvlew	http://www.abren.net/asaview/
PDB param	http://www.iitm.ac.in/bioinfo/pdbparam/
PIC	http://crick.mbu.iisc.ernet.in/~PIC
CMWeb	http://cmweb.enzim.hu/
INTAA	http://bioinfo.uochb.cas.cz/INTAA/
<i>Structure prediction</i>	
<i>Homology modelling</i>	
MODELLER	https://salilab.org/modeller/
SWISS-MODEL	https://swissmodel.expasy.org/
PRIMO	https://primo.rubi.ru.ac.za/
PyMod	http://schubert.bio.uniroma1.it/pymod/index.html
MaxMod	http://www.immt.res.in/maxmod/
<i>Fold recognition</i>	
GenTHREADER	http://bioinf.cs.ucl.ac.uk/psipred/
Phyre2	http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index
MUSTER	http://zhanglab.ccmb.med.umich.edu/MUSTER/
ORION	http://www.dsmb.inserm.fr/orion/
DN-Fold	http://iris.rnet.missouri.edu/dnfold/
<i>Ab initio structure prediction</i>	
Robetta	http://www.robetta.org/submit.jsp
I-TASSER	http://zhanglab.ccmb.med.umich.edu/I-TASSER/
QUARK	http://zhanglab.ccmb.med.umich.edu/QUARK/
BHAGEERATH	http://www.scfbio-iitd.res.in/bhageerath/index.jsp
EVfold	http://evfold.org/evfold-web/evfold.do
CABS-fold	http://biocomp.chem.uw.edu.pl/CABSfold/
PEP-FOLD	http://bioserv.rpbs.univ-paris-diderot.fr/services/PEP-FOLD/
<i>Thermodynamic data and effect of mutations on stability</i>	
ProTherm	http://www.abren.net/protherm/
PoPMuSiC	https://soft.dezyme.com/
CUPSAT	http://cupsat.tu-bs.de/
SDM	http://mordred.bioc.cam.ac.uk/~sdm/sdm.php
STRUM	http://zhanglab.ccmb.med.umich.edu/STRUM/
<i>Interaction databases</i>	
DIP	http://dip.doe-mbi.ucla.edu/dip/Main.cgi
IntAct	http://www.ebi.ac.uk/intact/
MINT	http://mint.bio.uniroma2.it/mint/
BioGRID	https://thebiogrid.org
SKEMPI	https://life.bsc.es/pid/mutation_database/
PROXIMATE	http://www.iitm.ac.in/bioinfo/PROXIMATE/
NDB	http://ndbserver.rutgers.edu/
ProNIT	http://www.abren.net/pronit/
PDIdb	http://melolab.org/pdldb
PROCARB	http://www.procarb.org/
<i>Druggable binding sites</i>	
Sc-PDB	http://bioinfo-pharma.u-strasbg.fr/scPDB/

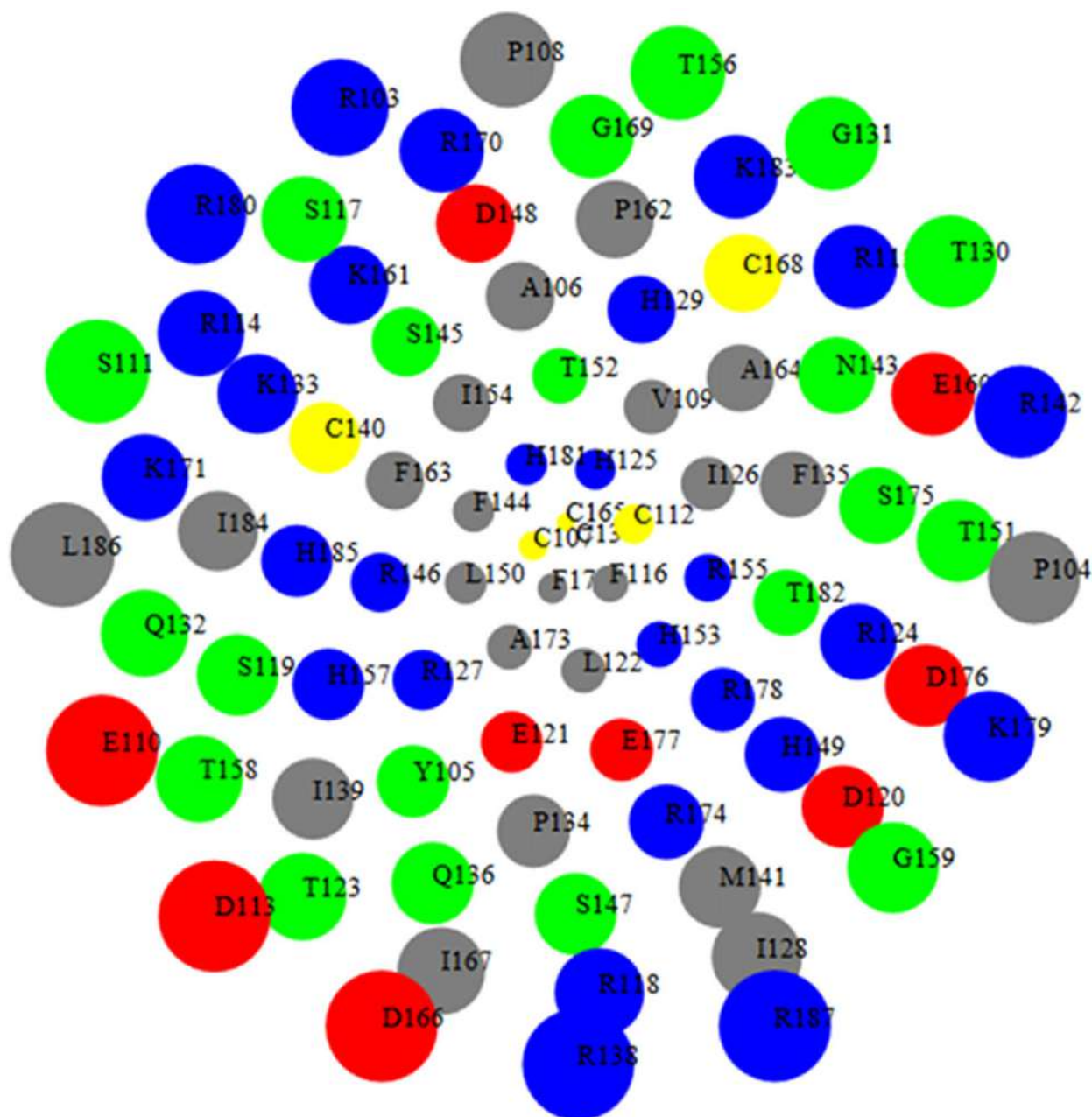


Fig. 4 Graphical representation of ASA values of amino acid residues in Zif268 zinc finger protein (PDB-id: 1AAY).

unfolded (extended) states of the protein, respectively. $\Delta\sigma_i$ refers to atomic solvation parameters obtained by fitting ΔG and ASA in extended conformation of each atom type in the 20 amino acid residues. It is calculated as:

$$\Delta G = \sum \Delta\sigma_i A_i \quad (3)$$

Electrostatic free energy is generally computed by Coulomb's law using the charge of interacting atoms and the distance between them.

$$E_{es} \propto q_i q_j / r_{ij} \quad (4)$$

where q_i and q_j are the effective partial charges for atoms i and j , respectively and r_{ij} is the distance between them.

Hydrogen bonds are responsible for the formation and stability of protein secondary structures, α -helices and β -strands. Hydrogen bonds are formed between two electronegative atoms mediating with a hydrogen atom. The contribution of hydrogen bonds is computed using the distance between the heavy atoms and the angle formed by the three atoms involved.

van der Waals interaction energy is generally computed using Lennard-Jones 6–12 potential:

$$G_{vw} = \left(A_{ij}/r_{ij}^{12} - B_{ij}/r_{ij}^6 \right) \quad (5)$$

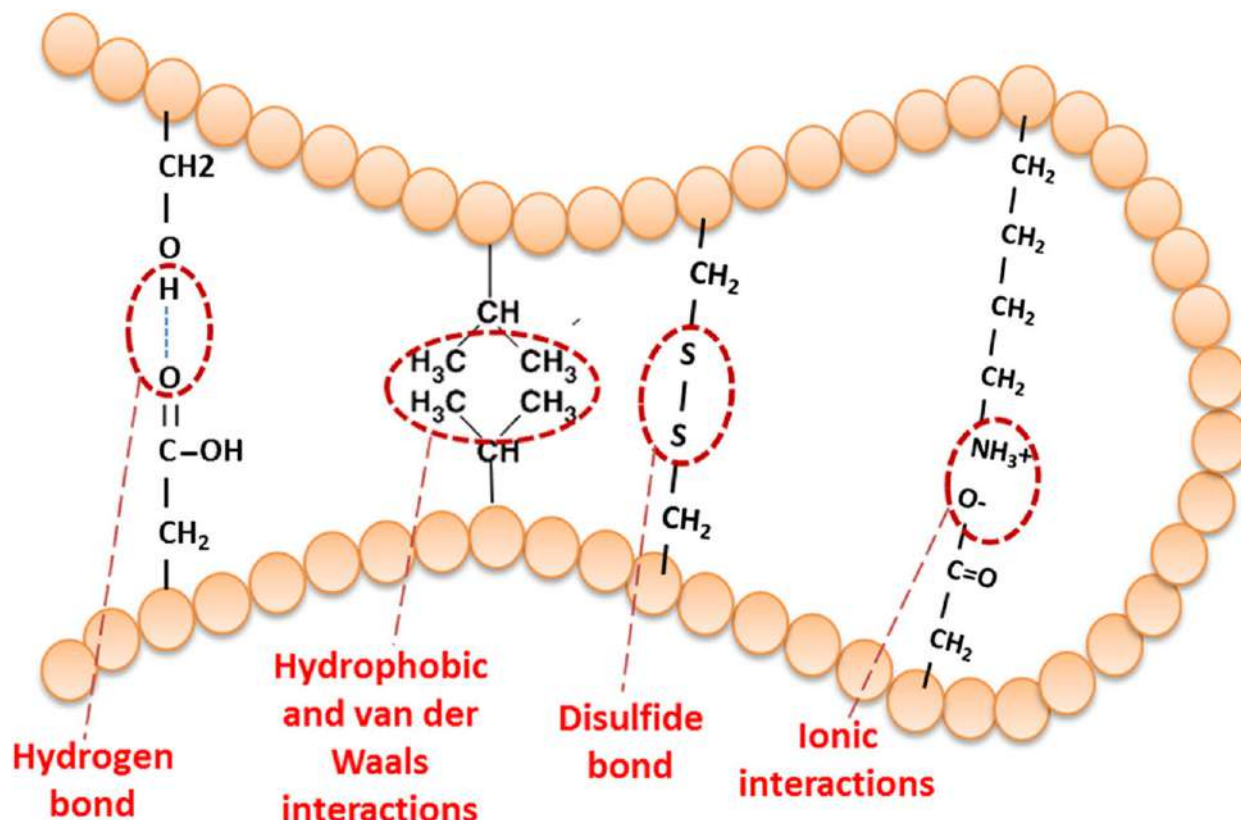


Fig. 5 Various non-covalent interactions and disulfide bonds in protein structures.

where $A_{ij} = \epsilon_{ij}^* (R_{ij}^*)^{12}$ and $B_{ij} = 2 \epsilon_{ij}^* (R_{ij}^*)^6$; $R_{ij}^* = (R_i^* + R_j^*)$ and $\epsilon_{ij}^* = (\epsilon_i^* \epsilon_j^*)^{1/2}$, R^* and ϵ^* are, respectively, the van der Waals radius and well depth.

In addition, cation- π interactions (Dougherty, 1996) are recognized to play an important role in globular, membrane, DNA binding, mesophilic and thermophilic proteins as well as in the recognition of protein-protein complexes. These are formed by the interaction between the positive charged residues Lys and Arg and aromatic residues, Phe, Trp and Tyr. Cation- π interactions in protein structures have been identified using distance and energy-based criteria: (i) distance between the cationic group and the centers of all aromatic rings should be less than 10 Å and (ii) the interaction energy between the positive charged and aromatic residues (sum of electrostatic and van der Waals) should be less than -2 kcal/mol.

Amino Acid Properties Derived from Protein Structural Data

Protein structures have been used to derive several properties for amino acid residues as well as the whole protein.

- (i) Medium and long-range contacts: Medium-range contacts refer to the contacts between amino acid residues within the distance of 8 Å and a sequence separation of three or four residues while the sequence separation is more than four residues in long-range contacts.
- (ii) Number of neighbouring residues: For each central residue, number of contacting residues within a specific cut-off distance (8 Å).
- (iii) Surrounding hydrophobicity: For a specific residue, this is the sum of the experimental hydrophobicity indices of residues, which occur within a sphere of radius 8 Å (Ponnuswamy, 1993). It is estimated using

$$H_p(i) = \sum_{j=1}^{20} n_{ij} h_j \quad (6)$$

where, n_{ij} is the total number of surrounding residues of type j around i th residue of the protein and h_j is the experimental hydrophobic index of residue type j in kcal/mol (Jones, 1975).

- (iv) Transfer free energy: The ASA of each residue provides information about the occurrence of that residue either in the interior of the protein or the surface and the ratio between them is used to compute the partition coefficient (f), which has

been used to derive the transfer free energy:

$$\Delta G_t = -RT \ln f \quad (7)$$

- (v) Buriedness and reduction in accessibility: Buriedness is defined as the ratio between the number of residues (of type *i*) in the interior of the protein to the total number of residues (of type *i*) in a protein. Similarly, reduction in accessibility is given by:

$$\langle R_A \rangle = A^0 - \langle A \rangle / A^0, \quad (8)$$

where, A^0 and $\langle A \rangle$ represent the accessible area in the unfolded (extended) and folded states of the protein, respectively.

- (vi) Flexibility: This parameter deals with the root mean square displacement of amino acid residues and is also related to the temperature factor (B factor) of amino acid residues in protein structures.
- (vii) Numerical values for amino acid properties: Kawashima *et al.* (2008) developed a database, AAindex, which contains the numerical indices for 516 physicochemical and biochemical properties of amino acids and pairs of amino acids. Gromiha *et al.* (1999a) compiled a set of 49 physicochemical, energetic and conformational properties of amino acid residues for understanding the folding and stability of proteins.

On the other hand, several parameters have been derived for a whole protein or protein chains.

- (viii) Contact Order: This measure reflects the relative importance of local and non-local contacts to the native structure of a protein (Plaxco *et al.*, 1998). It is defined as

$$CO = \sum \Delta S_{ij} / L \cdot N, \quad (9)$$

where ΔS_{ij} is the sequence separation between contacting residues *i* and *j* within a distance of 6 Å between any heavy atoms, *L* is the total number of residues in the protein and *N* is the total number of contacts.

- (ix) Long-range order: A knowledge of long-range contacts (contacts between two residues that are close in space and far in the sequence) in protein structure (Gromiha and Selvaraj, 2001) defines long-range order:

$$LRO = \sum n_{ij} / N; \quad n_{ij} = 1 \text{ if } |i - j| > 12; = 0 \text{ otherwise,} \quad (10)$$

where *i* and *j* are two contacting residues in space within a distance of 8 Å and *N* is the total number of residues in a protein.

- (x) Multiple contact index: When a residue has multiple contacts, the multiple contact index is used for computing long-range order (Gromiha, 2009) and is defined as

$$n_{ci} = \sum n_{ij}; \quad n_{ij} = 1 \text{ if } r_{ij} < 7.5 \text{ Å; } |i - j| > 12 \text{ residues; } 0 \text{ otherwise}$$

$$MCI = \sum n_{mi} / N; \quad n_{mi} = 1 \text{ if } n_{ci} \geq 4; 0 \text{ otherwise} \quad (11)$$

where n_c is the number of contacts for each residue, and r_{ij} is the distance between the residues *i* and *j*.

Nagarajan *et al.* (2016) developed a web-based tool, PDBparam, to derive a set of more than 50 features on four different themes, viz., (i) identification of binding sites, (ii) inter-residue interactions, (iii) secondary structure propensities and (iv) physicochemical properties. Similar web servers have also been developed for various weak and strong interactions, PIC (Tina *et al.*, 2007), contact between amino acid residues, CMWeb (Kozma *et al.*, 2012) and binding site residues in protein-DNA complexes, INTAA (Galgonek *et al.*, 2017).

Prediction of Protein Tertiary Structures

Deciphering the native conformation of a protein from its amino acid sequence, termed as the “protein folding problem” is a longstanding challenge in molecular and computational biology. Several methods have been proposed to achieve the goal and the performances of these methods have been critically evaluated by the forum, Critical Assessment of Techniques for Protein Structure Prediction (CASP), using blind prediction experiments. The developments of reliable methods are outlined in this section. Three major approaches namely, homology modeling, fold recognition and *ab initio* have been developed for the prediction of protein tertiary structures. In recent years, the above three approaches have been grouped into template-based modeling and template-free modeling (Li *et al.*, 2015; Källberg *et al.*, 2012; Xu *et al.*, 2009).

Homology Modeling

Homology modeling, also known as comparative modeling, is based on the biological fact that when two sequences share high similarity/identity, their respective structures are also similar. In this method, the 3D structure of a protein is obtained with the following steps: (i) for a given target sequence, identifying the proper template using BLAST search, (ii) sequence alignment (iii) alignment corrections to ensure the conserved or functionally important residues are aligned, (iv) backbone generation (v) loop modeling, (vi) side chain modeling using rotamer libraries, (vii) optimizing the model using energy minimization and (viii) validating the model by stereochemical evaluation using the residues in the allowed regions of Ramachandran plot as well as favourable energies. MODELLER (Webb and Sali, 2014) is one of the widely used computational tools for predicting protein 3D

structures using homology modeling. MaxMod, PyMod, and PRIMO are other recent methods/servers for homology modeling and their URLs are included in [Table 1](#).

Fold Recognition

If the sequence identity between the target sequence and template structures fall below 30%, homology modeling methods may not be able to produce a reliable model and under such situations fold recognition methods are used for prediction ([Buchan and Jones, 2017](#); [Lhota and Xie, 2016](#); [Jo et al., 2015](#)). These methods follow different steps to predict the 3D structure of a protein, which include (i) advanced sequence comparison methods, (ii) comparison of predicted secondary structure strings with those of known folds, (iii) tests of the compatibility of sequences with 3D folds and (iv) the use of human expert knowledge. Consequently, several methods and web servers have been developed for fold recognition such as, 3D-1D profiles ([David et al., 2004](#)), GenTHREADER ([Jones, 1999](#)), MUSTER ([Wu and Zhang, 2008](#)), ORION ([Ghouzam et al., 2016](#)), Phyre2 ([Kelley et al., 2015](#)), TMFoldRec ([Kozma and Tusnady, 2015](#)), DescFold ([Yan et al., 2009](#)) and DN-Fold ([Jo et al., 2015](#)). [Table 1](#) includes the web addresses of these servers.

Ab Initio Protein Structure Prediction

Ab initio prediction methods consider all bonded and non-bonded interactions involved in the process of folding, and choose the structure with lowest free energy. This approach is based on the ‘thermodynamic hypothesis’, which states that the native structure of a protein is the one for which the free energy achieves the global minimum. Two major components to *ab initio* prediction are (i) devising a scoring function that can distinguish between correct (native or native-like) structures from incorrect ones and (ii) a search method to explore the conformational space. One of the best performing methods for *ab initio* structure prediction is the Rosetta algorithm ([Simons et al., 1997](#)). Several other recent methods include I-TASSER ([Roy et al., 2010](#); [Yang et al., 2015](#)), Robetta ([Kim et al., 2004](#)), EVfold ([Marks et al., 2011](#)), CABS-fold ([Blaszczyk et al., 2013](#)), QUARK ([Xu and Zhang, 2012](#)), PEP-FOLD ([Thevenet et al., 2012](#); [Shen et al., 2014](#)) and Bhageerath ([Jayaram et al., 2014](#)) ([Table 1](#)).

Modeling on the Web

Few 3D structure prediction methods are available online with user friendly interface. SWISS-MODEL is one of the widely used comparative protein modeling servers to predict the structure for a given amino acid sequence directly ([Bienert et al., 2017](#)). It has different options such as (i) ‘first approach mode’, which requires only an amino acid sequence of a protein; the server automates subsequent steps including template selection, alignment and model building, (ii) ‘alignment mode’, which is the modeling process based on a user-defined target-template alignment and (iii) ‘project mode’ in which complex modeling tasks can be handled.

Protein Folding Kinetics

Many small proteins are known to fold rapidly by simple two-state kinetics and the mechanism of protein folding has been elaborately addressed with folding rates and Φ -value analysis. Φ -value is given by the equation:

$$\Phi_{\#} = \Delta\Delta G_{\#-U} / \Delta\Delta G_{F-U} \quad (12)$$

where $\Delta\Delta G_{F-U}$ is the change in stability between the folded (F) and unfolded states (U), and $\Delta\Delta G_{\#-U}$ is the energy difference between mutant and wild type in the transition state, #. $\Phi=0$ corresponds to no structure formation in the transition state as in the unfolded state and a Φ value of 1 is interpreted that the residue has a native like structure in transition state ([Itzhaki et al., 1995](#)). The analysis of experimental Φ values of 296 mutants in seven single domain proteins showed that more than 82% of them exhibit a Φ value below 0.6 ([Raleigh and Plaxco, 2005](#)). In addition, negative Φ values are interpreted as mutations which stabilize the folding transition state while it destabilizes the native state ([De los Rios et al., 2005](#)).

Folding Nuclei

The residues with high Φ values form folding nuclei in protein structures, which mediate important interactions for folding and assembly of their native states ([Dobson, 2003](#)). Folding nuclei could be computationally identified using different techniques: (i) searching free-energy saddle points on a network of protein unfolding pathways, (ii) elucidating the correspondence between conserved hydrophobic positions in amino acid sequences and folding nuclei, (iii) computing the conservation scores of amino acid residues based on physicochemical properties and (iv) coupling ensembles of pathways with the lowest effective contact order and identifying contacts that are crucial for folding.

Protein Folding Rates

Protein folding rate is a measure of the slow/fast folding of a protein from its unfolded state to the native 3D structure. The data for folding rates for proteins and their mutants are available in literature ([Gromiha, 2010](#); [Chaudhary et al., 2015](#)).

Relating Protein Folding Rates Using Structural Parameters

Several structure-based parameters have been proposed to understand and predict protein folding rates. [Plaxco *et al.* \(1998\)](#) proposed the concept of contact order (CO) by estimating the average sequence separation of all contacting residues in the native state within a distance of 6Å normalized with total number of contacts and total number of residues. [Gromiha and Selvaraj \(2001\)](#) considered only the long range contacts with a sequence separation of 12 residues, which are within 8Å limit and proposed the concept of LRO. [Istomin *et al.* \(2007\)](#) showed that LRO is the only parameter that shows good correlation in all structural classes of proteins. Further, protein folding rates are well studied with total contact distance, first principles of protein folding, combination of CO and stability, topomer search model, topological properties of protein conformations, cliquishness, multiple contact index, helix parameter, n-order contact distance and chain connectivity due to crosslinks ([Gromiha, 2010](#)).

Prediction of Protein Folding Rates From Amino Acid Sequence

The analysis of the relationship between amino acid properties and protein folding rates showed a poor correlation between them. On the other hand, the secondary structure content of a protein and chain length showed a linear relationship with protein folding rates. Although chain length showed a good inverse correlation with the folding rates of three-state proteins no correlation between them was observed in two-state proteins ([Galzitskaya *et al.*, 2003](#)).

[Punta and Rost \(2005\)](#) predicted amino acid contacts using amino acid sequence and utilize the information to predict protein folding rates using the concept of long-range order. We have developed multiple regression models for predicting the folding rates of protein using amino acid properties ([Gromiha *et al.*, 2006](#)). Furthermore, protein folding rates are predicted with amino acid composition, rigidity and quadratic response surface models, n-order contact distance, secondary structure, solvent accessibility, compression capability of protein sequence, reactivity of proteins in solvents, contact surface of the unfolded chain and solvent, Gibbs free energy of hydration in denatured proteins, the average range of amino acid contact, the polarity value of amino acids, number of torsion angles of the side chain and residue-level coevolutionary information. The different algorithms for predicting the folding rates of proteins have been detailed in recent review articles ([Gromiha, 2010](#); [Gromiha and Huang, 2011](#); [Chang *et al.*, 2015](#)).

Prediction of Protein Folding Rates Upon Mutation

Amino acid substitutions may accelerate or decelerate the folding rate of a protein and it mainly depends on the location and type of mutations. [Huang and Gromiha \(2010\)](#) proposed the first approach to distinguish between accelerating and decelerating mutants using amino acid properties. The method has been extended further to predict the real value change in folding rates ([Huang and Gromiha, 2012](#)). The integrated approach using secondary structure information, location of mutants in a protein as well as the distribution along the sequence is reported to substantially improve the prediction accuracy of protein folding rate change upon mutation ([Chaudhary *et al.*, 2015](#); see Relevant Websites section). The analysis also showed that folding rate is primarily influenced with hydrophobicity, emphasizing the dominant influence of hydrophobic effect during the folding process. [Mallik *et al.* \(2016\)](#) utilized residue-level coevolutionary information for predicting protein folding rates upon mutations.

Protein Stability

Protein stability (ΔG) is the net balance between free energies in the folded and unfolded states. While the folded state is stabilized through disulphide bonds and non-covalent interactions, hydrophobic, electrostatic, hydrogen bonding and van der Waals interactions, the unfolded state is influenced by entropy. However, the difference between the folded and unfolded states is marginal and ΔG lies in the range of 5–25 kcal/mol.

Thermodynamic Database for Proteins and Mutants

[Gromiha *et al.* \(2016\)](#) compiled the data on stability of proteins and mutants from the literature and developed a database, ProTherm, which contains the thermodynamic parameters for more than 25,000 normal (or wild-type) and mutant proteins along with sequence, structure, function and literature information. The analysis of ΔG values of protein structures along with the contribution of free energies showed that the stability is mainly achieved by hydrophobic interactions by overcoming the entropic factors whereas other interactions help to maintain stability, provide the shape and keep the folded state from falling apart.

Stability of Thermophilic Proteins

Thermophilic proteins maintain their stability at high temperatures (80–100°C) and there is a direct relationship between environmental growth temperature and melting temperature ([Gromiha *et al.*, 1999b](#); [Gaucher *et al.*, 2008](#)). Analysis of amino acid sequences in thermophilic proteins as well as their mesophilic homologues showed a preference of Ala, Thr, Arg and Glu over Gly, Ser, Lys and Asp, respectively ([Gromiha and Suresh, 2008](#)). Additionally, the role of specific amino acid properties such as shape,

Gibbs free energy change of hydration in native proteins, dipeptide composition, contacts between amino acid residues, number of ion pairs, hydrogen bonds, packing and aromatic clusters are reported to be important to enhance the stability of thermophilic proteins (Gromiha *et al.*, 1999b; Pica and Graziano, 2016). On the other hand, available pairs of mesophilic and thermophilic protein structures have been used effectively to detect interactions important for the stability of thermophilic proteins. Gromiha *et al.* (2013) carried out a comprehensive analysis on the influence of all interactions and showed that hydrophobicity is the major factor in the stability of thermophilic proteins followed by ion pairs and hydrogen bonds. The systematic elimination of mesophilic proteins based on surrounding hydrophobicity, interaction energy and ion pairs/hydrogen bonds can be used to identify 95% of the thermophilic proteins correctly.

Stability of Proteins Upon Mutations

The availability of thermodynamic data for proteins and mutants in the ProTherm database prompted researchers to understand the factors influencing the stability of mutants as well as predicting the free energy change upon mutation. It has been shown that buried mutations are influenced mainly by hydrophobicity whereas partially buried and exposed mutants are affected by neighboring and surrounding residues (Gromiha *et al.*, 1999b).

The effect of mutation on free energy change of a protein has been systematically analyzed with the development of torsion and distance potentials and environment dependent substitution tables, amino acid properties and energy based approach. Further, the mutational information as well as amino acid properties have been trained with machine learning techniques for predicting the change in free energy upon mutation. Saraboji *et al.* (2006) proposed an “average assignment method” based on the average free energy change of all possible 380 mutants, which provides first-hand information on the capability of a method for predicting the free energy change upon mutation. Further, several methods were developed to estimate the protein stability changes upon point mutations using sequence/structure information such as PoPMuSiC, CUPSAT, SDM, STRUM etc (Table 1). Recent reviews comprehensively describe the methods and applications for predicting the stability of mutants and their applications (Gromiha and Huang, 2011; Gromiha *et al.*, 2016).

Protein Interactions

The interactions of proteins with other molecules govern various cellular processes. The functions of protein complexes have been studied with a view to understand the recognition mechanism, predicting the binding partners and binding site residues as well as the binding affinity of protein-protein, protein-nucleic acid and protein-ligand complexes.

Protein-Protein Interactions

Protein-protein interactions have been studied with two major aspects, (i) within protein-protein complexes and (ii) large scale analysis on protein-protein interaction networks. Protein-protein complex level approach includes the analysis and identification of binding sites, development of structural, functional, thermodynamic and kinetic databases for protein-protein complexes, understanding the binding affinity and recognition mechanism of protein-protein complexes. Interaction network analysis deals with the interaction of each protein with other proteins within the same organism or between different organisms (E.g. host-pathogen interactions) and the complexity of interactions among a set of proteins.

Several databases have been developed to relate various aspects of protein-protein interactions such as interacting pairs of proteins (DIP; Xenarios *et al.*, 2002, IntAct; Orchard *et al.*, 2014), functional interactions between proteins (MINT; Zanzoni *et al.*, 2002), interactions between proteins in human proteome and other organisms (BioGRID; Chatr-Aryamontri *et al.*, 2017), kinetic data on binding affinity (SKEMPI; Moal and Fernández-Recio, 2012) and thermodynamic data for protein-protein interactions and mutants (PROXIMATE; Jemimah *et al.*, 2017) (Table 1). The 3D structures of protein-protein complexes are deposited in Protein Data Bank (Burley *et al.*, 2017).

The availability of experimental data on protein-protein complexes in several databases as well literature prompted researchers to understand the recognition mechanism using statistical analysis of binding sites, identification of binding site residues, prediction of protein-protein complex structures as demonstrated with Critical Assessment of Protein-protein Interactions (CAPRI; Lensink *et al.*, 2017), distinguishing high and low affinity complexes, predicting the binding affinity of protein-protein complexes using sequence and structure based parameters and energy based approach for understanding the recognition mechanism. Recently, Kulandaisamy *et al.* (2017) dissected the important residues, which are involved in both folding and binding, and revealed the characteristic features of these residues. The developments on binding site prediction, binding affinity, complex structure prediction and scoring schemes have been reviewed in detail (Gromiha and Yugandhar, 2017; Gromiha *et al.*, 2017).

Protein-Nucleic Acid Interactions

Owing to the importance of protein-DNA interactions in gene regulation, DNA replication and repair, several computational studies have been performed to understand the recognition mechanism (Gromiha and Nagarajan, 2013). Databases have been developed to accumulate the experimental structures of protein-nucleic acid complexes (NDB) (Narayanan *et al.*, 2013), pairs of contacting amino acid residues and DNA bases (PDIdb) (Norambuena and Melo, 2010), and thermodynamic data of protein-

nucleic acid interactions (ProNIT) (Kumar *et al.*, 2006) (Table 1). Utilizing these information, computational analyses have been carried out to extract important interactions, such as hydrogen bonds, physicochemical properties at the interface, interface surface area, structure based potentials, cation- π interactions, moments of electric charge distribution, DNA stiffness, clustering of protein-DNA complexes and thermodynamics for binding. Further, the role of direct and indirect readout mechanisms (Gromiha *et al.*, 2004) and the importance of scoring functions have been explored for understanding the recognition mechanism of protein-DNA complexes (Gromiha and Fukui, 2011). On the other hand, several methods have been proposed to discriminate DNA binding proteins and predicting the binding sites (Gromiha and Nagarajan, 2013). Nagarajan *et al.* (2013) showed that the performance of prediction methods depends on the structure and function of the specific protein and DNA.

Similar to protein-DNA interactions, the structures of protein-RNA complexes have been effectively used to understand the importance of specific interactions in the recognition mechanism and to predict binding sites (Nagarajan and Gromiha, 2014). In addition, Nagarajan *et al.* (2015) utilized statistical analysis and molecular dynamics simulation of the same protein-RNA complex from different organisms and showed that the recognition mechanism is specific to the target organism.

Protein-Carbohydrate and Protein-Ligand Interactions

The availability of data for protein-carbohydrates is relatively limited, compared to protein-protein and protein-nucleic acid complexes. Malik *et al.* (2010) developed the only database for protein-carbohydrate complex structures (PROCARB) (Table 1) and annotated the binding sites. The available 3D structures have been utilized to reveal the important factors and to understand the recognition mechanism of protein-carbohydrate complexes (Gromiha *et al.*, 2014).

Several databases are available for protein-ligand complexes, specifically for binding free energies, structural information, and druggable binding sites in protein structures (sc-PDB) (Kellenberger *et al.*, 2006). Accordingly, several methods have been proposed to predict the ligand binding sites and binding affinity of complexes upon binding (Tsujikawa *et al.*, 2016; Singh *et al.*, 2016; Pai *et al.*, 2017).

Proteins and Drug Design

Proteins, the key functional molecules in all organisms, are one of the major drug targets to combat diseases. Small organic molecules obtained from natural sources or synthesized in the laboratory are used to inhibit the biological activity of enzymes, receptors etc., which are involved in disease processes. When the 3D structure of a target receptor is known, computational methods (*in silico*) such as virtual screening or docking are used to screen a large number of compounds available in chemical databases to identify potential inhibitors. Protein-ligand docking involves different steps such as identifying the active sites, ligand flexibility and interaction energy between ligand and protein. In the absence of 3D structures of target receptors, homology modeling is used to construct a 3D model of the receptor to be used for virtual screening/docking. In recent years, several attempts have been made to identify hit compounds as potential inhibitors to target proteins for addressing various diseases such as dengue, chikungunya, cancer etc. (Anusuya *et al.*, 2016; Akhtar and Jabeen, 2017; Leal *et al.*, 2017; Zou *et al.*, 2017; Ramakrishnan *et al.*, 2017). These compounds are taken further for experimental verifications and drug development. Chiba *et al.* (2015) proposed a contest based approach for identifying potential inhibitors for cYes kinase using *in silico* screening and experimental validations. In protein-ligand docking, two important bottlenecks are conformational sampling and scoring functions. AutoDock and Glide are widely used freely available and commercial docking programs to identify probable drug compounds. Recently, Anusuya *et al.* (2017) reviewed the methods and applications of drug-target interactions.

On the other hand, quantitative structure-activity relationship (QSAR) model(s) are developed for relating the biological activity of compounds such as IC₅₀ (Half maximal inhibitory concentration) and/or EC₅₀ (Half maximal effective concentration) with structure based features of ligands such as hydrophobic, electronic, steric and topological parameters. The structural features of the ligands are combined using multiple regression to relate with experimental activity values. This procedure is widely used to identify hit compounds for several diseases (Kanakaveti *et al.*, 2017). Alternatively, a set of molecules could be analyzed to identify key pharmacophore features, such as hydrogen bond donors, hydrogen bond acceptors, aromatic rings/hydrophobic groups and related with biological activity.

Conclusions and Future Perspectives

The 3D structures of proteins and their complexes have been effectively utilized to derive several structure based parameters and contact potentials. These features have been successfully used to predict protein folding rates, 3D structures of proteins, protein stability, stability upon mutations, binding sites and binding affinity of protein complexes. The refinement of these parameters would improve the prediction performance as well as successful prediction of diverse systems. We have reviewed the methods and applications as well as latest developments on protein folding, stability and interactions. Most of the prediction methods could be refined with larger datasets and effective learning methodologies such as machine learning algorithms and deep learning. In addition, high throughput *in silico* screening of small molecules and next generation sequence analysis would aid the identification of lead compounds for drug discovery.

Acknowledgements

MMG thanks Prof. Shoba Ranganathan for her invitation to contribute the review article. The work is partially supported by the Department of Science and Technology, Government of India to MMG (EMR/2016/001476).

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Assessment of Structure Quality (RNA and Protein). Biomolecular Structures: Prediction, Identification and Analyses. Cloud-Based Molecular Modeling Systems. Computational Protein Engineering Approaches for Effective Design of New Molecules. Computational Tools for Structural Analysis of Proteins. Drug Repurposing and Multi-Target Therapies. Epitope Predictions. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Molecular Dynamics and Simulation. Mutation Effects on 3D-Structural Reorganization Using HIV-1 Protease as A Case Study. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Population-Based Sampling and Fragment-Based De Novo Protein Structure Prediction. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Design. Protein Properties. Protein Structure Analysis and Validation. Protein Structure Classification. Protein Structure Databases. Protein Structure Visualization. Protein Three-Dimensional Structure Prediction. Protocol for Protein Structure Modelling. Rational Structure-Based Drug Design. Secondary Structure Prediction. Small Molecule Drug Design. Structural Genomics. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow. Study of The Variability of The Native Protein Structure. Transmembrane Domain Prediction

References

- Ahmad, S., Gromiha, M.M., Fawareh, H., Sarai, A., 2004. ASAView: Database and tool for solvent accessibility representation in proteins. *BMC Bioinform* 5, 51.
- Akhtar, N., Jabeen, I., 2017. Pharmacoinformatic approaches to design novel inhibitors of protein kinase B pathways in cancer. *Curr Cancer Drug Targets*. in press.
- Almeida, J.G., Preto, A.J., Koukos, P.I., Bonvin, A.M.J.J., Moreira, I.S., 2017. Membrane proteins structures: A review on computational modeling tools. *Biochim Biophys Acta* 1859, 2021–2039.
- Andreeva, A., Howorth, D., Chandonia, J.-M., *et al.*, 2008. Data growth and its impact on the SCOP database: New developments. *Nucleic Acids Res* 36 (Database Issue), D419–D425.
- Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. *Science* 181 (96), 223–230.
- Anusuya, S., Keshewani, M., Priya, K.V., *et al.*, 2017. Drug–target interactions: Prediction methods and applications. *Curr Protein Pept Sci*. in press.
- Anusuya, S., Velmurugan, D., Gromiha, M.M., 2016. Identification of dengue viral RNA-dependent RNA polymerase inhibitor using computational fragment-based approaches and molecular dynamics study. *J Biomol Struct Dyn* 34 (7), 1512–1532.
- Bienert, S., Waterhouse, A., de Beer, T.A., *et al.*, 2017. The SWISS-MODEL repository – New features and functionality. *Nucleic Acids Res* 45 (D1), D313–D319.
- Błaszczak, M., Jamroz, S., Kmiecik, S., Kolinski, A., 2013. CABS-fold: Server for the de novo and consensus-based prediction of protein structure. *Nucleic Acids Res* 41 (Web Server Issue), W406–W411.
- Buchan, D.W., Jones, D.T., 2017. EigenTHREADER: Analogous protein fold recognition by efficient contact map threading. *Bioinformatics* 33 (17), 2684–2690.
- Burley, S.K., Berman, H.M., Kleywegt, G.J., *et al.*, 2017. Protein Data Bank (PDB): The single global macromolecular structure archive. *Methods Mol Biol* 1607, 627–641.
- Chang, C.C., Tey, B.T., Song, J., Ramanan, R.N., 2015. Towards more accurate prediction of protein folding rates: A review of the existing Web-based bioinformatics approaches. *Brief Bioinform* 16 (2), 314–324.
- Chatr-Aryamontri, A., Oughtred, R., Boucher, L., *et al.*, 2017. The BioGRID interaction database: 2017 Update. *Nucleic Acids Res* 45 (Database Issue), D369–D379.
- Chaudhary, P., Naganathan, A., Gromiha, M.M., 2015. Folding RaCe: A robust method for predicting changes in protein folding rates upon point mutations. *Bioinformatics* 31 (13), 2091–2097.
- Chiba, S., Ikeda, K., Ishida, T., *et al.*, 2015. Identification of potential inhibitors based on compound proposal contest: Tyrosine-protein kinase Yes as a target. *Sci Rep* 5, 17209.
- David, R., Korenberg, M.J., Hunter, I.W., 2000. 3D–1D threading methods for protein fold recognition. *Pharmacogenomics* 1 (4), 445–455.
- De los Rios, M.A., Daneshi, M., Plaxco, K.W., 2005. Experimental investigation of the frequency and substitution dependence of negative phi-values in two-state proteins. *Biochemistry* 44 (36), 12160–12167.
- Dobson, C.M., 2003. Protein folding and misfolding. *Nature* 426, 884–890.
- Dougherty, D.A., 1996. Cation-pi interactions in chemistry and biology: A new view of benzene, Phe, Tyr, and Trp. *Science* 271 (5246), 163–168.
- Galgonek, J., Vymetal, J., Jakubec, D., Vondrášek, J., 2017. Amino Acid Interaction (INTAA) web server. *Nucleic Acids Res* 45, W388–W392.
- Galzitskaya, O.V., Garbuzynskiy, S.O., Ivankov, D.N., Finkelstein, A.V., 2003. Chain length is the main determinant of the folding rate for proteins with three-state folding kinetics. *Proteins* 51, 162–166.
- Gaucher, E.A., Govindarajan, S., Ganesh, O.K., 2008. Palaeotemperature trend for Precambrian life inferred from resurrected proteins. *Nature* 451 (7179), 704–707.
- Ghouzani, Y., Postic, G., Guerin, P.E., de Brevern, A.G., Gelly, J.C., 2016. ORION: A web server for protein fold recognition and structure prediction using evolutionary hybrid profiles. *Sci Rep*, 6, p. 28268.
- Gromiha, M.M., 2009. Multiple contact network is a key determinant to protein folding rates. *J Chem Inf Model* 49 (4), 1130–1135.
- Gromiha, M.M., 2010. *Protein Bioinformatics*. New Delhi: Academic Press.
- Gromiha, M.M., Ahmad, S., 2005. Role of solvent accessibility in structure based drug design. *Curr Comp Aided Drug Des* 1, 65–72.
- Gromiha, M.M., Anooosha, P., Huang, L.T., 2016. Applications of protein thermodynamic database for understanding protein mutant stability and designing stable mutants. *Methods Mol Biol* 1415, 71–89.
- Gromiha, M.M., Fukui, K., 2011. Scoring function based approach for locating binding sites and understanding recognition mechanism of protein–DNA complexes. *J Chem Inf Model* 51 (3), 721–729.
- Gromiha, M.M., Huang, L.T., 2011. Machine learning algorithms for predicting protein folding rates and stability of mutant proteins: Comparison with statistical methods. *Curr Protein Pept Sci* 12 (6), 490–502.
- Gromiha, M.M., Nagarajan, R., 2013. Computational approaches for predicting the binding sites and understanding the recognition mechanism of protein–DNA complexes. *Adv Protein Chem Struct Biol* 91, 65–99.

- Gromiha, M.M., Oobatake, M., Kono, H., Uedaira, H., Sarai, A., 1999a. Relationship between amino acid properties and protein stability: Buried mutations. *J Protein Chem* 18, 565–578.
- Gromiha, M.M., Oobatake, M., Sarai, A., 1999b. Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins. *Biophys Chem* 82 (1), 51–67.
- Gromiha, M.M., Pathak, M.C., Saraboji, K., Ortlund, E.A., Gaucher, E.A., 2013. Hydrophobic environment is a key factor for the stability of thermophilic proteins. *Proteins* 81 (4), 715–721.
- Gromiha, M.M., Selvaraj, S., 2001. Comparison between long-range interactions and contact order in determining the folding rates of two-state proteins: Application of long-range order to folding rate prediction. *J Mol Biol* 310, 27–32.
- Gromiha, M.M., Selvaraj, S., 2004. Inter-residue interactions in protein folding and stability. *Prog Biophys Mol Biol* 86 (2), 235–277.
- Gromiha, M.M., Siebers, J.G., Selvaraj, S., Kono, H., Sarai, A., 2004. Intermolecular and intramolecular readout mechanisms in protein–DNA recognition. *J Mol Biol* 337, 285–294.
- Gromiha, M.M., Suresh, M.X., 2008. Discrimination of mesophilic and thermophilic proteins using machine learning algorithms. *Proteins* 70 (4), 1274–1279.
- Gromiha, M.M., Thangakani, A.M., Selvaraj, S., 2006. FOLD-RATE: Prediction of protein folding rates from amino acid sequence. *Nucleic Acids Res* 34, W70–W74.
- Gromiha, M.M., Veluraja, K., Fukui, K., 2014. Identification and analysis of binding site residues in protein–carbohydrate complexes using energy based approach. *Protein Pept Lett* 21 (8), 799–807.
- Gromiha, M.M., Yugandhar, K., 2017. Integrating computational methods and experimental data for understanding the recognition mechanism and binding affinity of protein–protein complexes. *Prog Biophys Mol Biol* 128, 18–33.
- Gromiha, M.M., Yugandhar, K., Jemimah, S., 2017. Protein–protein interactions: Scoring schemes and binding affinity. *Curr Opin Struct Biol* 44, 31–38.
- Huang, L.T., Gromiha, M.M., 2010. First insight into the prediction of protein folding rate change upon point mutation. *Bioinformatics* 26 (17), 2121–2127.
- Huang, L.T., Gromiha, M.M., 2012. Real value prediction of protein folding rate change upon point mutation. *J Comput Aided Mol Des* 26 (3), 339–347.
- Istomin, A.Y., Jacobs, D.J., Livesay, D.R., 2007. On the role of structural class of a protein with two-state folding kinetics in determining correlations between its size, topology, and folding rate. *Protein Sci* 16 (11), 2564–2569.
- Itzhaki, L.S., Otzen, D.E., Fersht, A.R., 1995. The structure of the transition state for folding of chymotrypsin inhibitor 2 analysed by protein engineering methods: Evidence for a nucleation-condensation mechanism for protein folding. *J Mol Biol* 254, 260–288.
- Jayaram, B., Dhingra, P., Mishra, A., *et al.*, 2014. Bhageerath-H: A homology/ab initio hybrid server for predicting tertiary structures of monomeric soluble proteins. *BMC Bioinform* 15, Suppl 16:S7.
- Jemimah, S., Yugandhar, K., Gromiha, M.M., 2017. PROXiMATE: A database of mutant protein–protein complex thermodynamics and kinetics. *Bioinformatics*. in press.
- Jones, D.D., 1975. Amino acid properties and side-chain orientation in proteins: A cross correlation approach. *J Theor Biol* 50, 167–183.
- Jones, D.T., 1999. GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol* 287 (4), 797–815.
- Jo, T., Hou, J., Eickholt, J., Cheng, J., 2015. Improving protein fold recognition by deep learning networks. *Sci Rep* 5, 17573.
- Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.
- Källberg, M., Wang, M., Wang, S., *et al.*, 2012. Template-based protein structure modeling using the RaptorX web server. *Nat Protoc* 7, 1511–1522.
- Kanakaveti, V., Sakthivel, R., Rayala, S.K., Gromiha, M.M., 2017. Importance of functional groups in predicting the activity of small molecule inhibitors for Bcl-2 and Bcl-xL. *Chem Biol Drug Des* 90 (2), 308–316.
- Kawashima, S., Pokarowski, P., Pokarowska, M., *et al.*, 2008. AAindex: Amino acid index database, progress report 2008. *Nucleic Acids Res* 36 (Database Issue), D202–D205.
- Kellenberger, E., Muller, P., Schalon, C., *et al.*, 2006. sc-PDB: An annotated database of druggable binding sites from the Protein Data Bank. *J Chem Inf Model* 46 (2), 717–727.
- Kelley, L.A., Mezulis, S., Yates, C.M., Wass, M.N., Sternberg, M.J., 2015. The Phyre2 web portal for protein modeling, prediction and analysis. *Nat Protoc* 10 (6), 845–858.
- Kim, D.E., Chivian, D., Baker, D., 2004. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res* 32 (Web Server Issue), W526–W531.
- Kozma, D., Simon, I., Tusnády, G.E., 2012. CMWeb: An interactive on-line tool for analysing residue–residue contacts and contact prediction methods. *Nucleic Acids Res* 40 (Web Server Issue), W329–W333.
- Kozma, D., Simon, I., Tusnády, G.E., 2013. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. *Nucleic Acids Res* 41 (Database issue), D524–D529.
- Kozma, D., Tusnády, G.E., 2015. TMFoldRec: A statistical potential-based transmembrane protein fold recognition tool. *BMC Bioinformatics* 16, 201.
- Kulandaisamy, A., Lathi, V., ViswaPoorani, K., Yugandhar, K., Gromiha, M.M., 2017. Important amino acid residues involved in folding and binding of protein–protein complexes. *Int J Biol Macromol* 94 (Pt A), 438–444.
- Kumar, M.D., Bava, K.A., Gromiha, M.M., *et al.*, 2006. ProTherm and ProNIT: Thermodynamic databases for proteins and protein–nucleic acid interactions. *Nucleic Acids Res* 34 (Database Issue), D204–D206.
- Leal, E.S., Aucar, M.G., Gebhard, L.G., *et al.*, 2017. Discovery of novel dengue virus entry inhibitors via a structure-based approach. *Bioorg Med Chem Lett* 27 (16), 3851–3855.
- Lee, B., Richards, F.M., 1971. The interpretation of protein structures: Estimation of static accessibility. *J Mol Biol* 55, 379–400.
- Lensink, M.F., Velankar, S., Wodak, S.J., 2017. Modeling protein–protein and protein–peptide complexes: CAPRI 6th edition. *Proteins* 85 (3), 359–377.
- Lhota, J., Xie, L., 2016. Protein-fold recognition using an improved single-source K diverse shortest paths algorithm. *Proteins* 84 (4), 467–472.
- Li, J., Adhikari, B., Cheng, J., 2015. An improved integration of template-based and template-free protein structure modeling methods and its assessment in CASP11. *Protein Pept Lett* 22 (7), 586–593.
- Malik, A., Firoz, A., Jha, V., Ahmad, S., 2010. PROCARB: A database of known and modelled carbohydrate-binding protein structures with sequence-based prediction tools. *Adv Bioinform*. 436036.
- Mallik, S., Das, S., Kundu, S., 2016. Predicting protein folding rate change upon point mutation using residue-level coevolutionary information. *Proteins* 84 (1), 3–8.
- Marks, D.S., Colwell, L.J., Sheridan, R., *et al.*, 2011. Protein 3D structure computed from evolutionary sequence variation. *PLOS One* 6 (12), e28766.
- Moal, I.H., Fernández-Recio, J., 2012. SKEMPI: A structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics* 28, 2600–2607.
- Nagarajan, R., Gromiha, M.M., 2014. Prediction of RNA binding residues: An extensive analysis based on structure and function to select the best predictor. *PLOS One* 9 (3), e91140.
- Nagarajan, R., Ahmad, S., Gromiha, M.M., 2013. Novel approach for selecting the best predictor for identifying the binding sites in DNA binding proteins. *Nucleic Acids Res* 41 (16), 7606–7614.
- Nagarajan, R., Archana, A., Thangakani, A.M., *et al.*, 2016. PDBparam: Online Resource for Computing Structural Parameters of Proteins. *Bioinform Biol Insights* 10, 73–80.
- Nagarajan, R., Chothani, S.P., Ramakrishnan, C., Sekijima, M., Gromiha, M.M., 2015. Structure based approach for understanding organism specific recognition of protein–RNA complexes. *Biol Direct* 10, 8.
- Narayanan, B.C., Westbrook, J., Ghosh, S., *et al.*, 2013. The Nucleic Acid Database: New features and capabilities. *Nucleic Acids Res.* 42, D114–D122.
- Norambuena, T., Melo, F., 2010. The Protein–DNA Interface database. *BMC Bioinformatics* 11, 262.
- Orchard, S., Ammar, M., Aranda, B., *et al.*, 2014. The MIntAct project – IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res* 42 (Database Issue), D358–D363.
- Pai, P.P., Dattatreya, R.K., Mondal, S., 2017. Ensemble architecture for prediction of enzyme–ligand binding residues using evolutionary information. *Mol Inform.* in press.
- Pica, A., Graziano, G., 2016. Shedding light on the extra thermal stability of thermophilic proteins. *Biopolymers* 105 (12), 856–863.

- Plaxco, K.W., Simons, K.T., Baker, D., 1998. Contact order, transition state placement and the refolding rates of single domain proteins. *J Mol Biol* 277, 985–994.
- Ponnuswamy, P.K., 1993. Hydrophobic characteristics of folded proteins. *Prog Biophys Mol Biol* 59 (1), 57–103.
- Punta, M., Rost, B., 2005. Protein folding rates estimated from contact predictions. *J Mol Biol* 348, 507–512.
- Raleigh, D.P., Plaxco, K.W., 2005. The protein folding transition state: What are Phi-values really telling us? *Protein Pept Lett* 12 (2), 117–122.
- Ramakrishnan, C., Thangakani, A.M., Velmurugan, D., *et al.*, 2017. Identification of type I and type II inhibitors of c-Yes kinase using in silico and experimental techniques. *J Biomol Struct Dyn.* in press.
- Roy, A., Kucukural, A., Zhang, Y., 2010. I-TASSER: A unified platform for automated protein structure and function prediction. *Nat Protoc* 5, 725–738.
- Saraboji, K., Gromiha, M.M., Ponnuswamy, M.N., 2006. Average assignment method for predicting the stability of protein mutants. *Biopolymers* 82 (1), 80–92.
- Shen, Y., Maupetit, J., Derreumaux, P., Tufféry, P., 2014. Improved PEP-FOLD approach for peptide and miniprotein structure prediction. *J Chem Theor Comput* 10, 4745–4758.
- Sillitoe, I., Lewis, T.E., Cuff, A.L., *et al.*, 2015. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res* 43 (Database Issue), D376–D381.
- Simons, K.T., Kooperberg, C., Huang, E., Baker, D., 1997. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J Mol Biol* 268, 209–225.
- Singh, H., Srivastava, H.K., Raghava, G.P., 2016. A web server for analysis, comparison and prediction of protein ligand binding sites. *Biol Direct* 11 (1), 14.
- Sippl, M.J., 1990. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *J Mol Biol* 213, 859–883.
- The UniProt Consortium, 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Res* 45, D158–D169.
- Thévenet, P., Shen, Y., Maupetit, J., *et al.*, 2012. PEP-FOLD: An updated de novo structure prediction server for both linear and disulfide bonded cyclic peptides. *Nucleic Acids Res* 40, W288–W293.
- Tina, K.G., Bhadra, R., Srinivasan, N., 2007. PIC: Protein interactions calculator. *Nucleic Acids Res* 35 (suppl_2), W473–W476.
- Tsujikawa, H., Sato, K., Wei, C., *et al.*, 2016. Development of a protein–ligand-binding site prediction method based on interaction energy and sequence conservation. *J Struct Funct Genomics* 17 (2–3), 39–49.
- Webb, B., Sali, A., 2014. Comparative protein structure modeling using modeller. *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc. pp. 5.6.1–5.6.32.
- Wu, S., Zhang, Y., 2008. MUSTER: Improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins* 72 (2), 547–556.
- Xenarios, I., Salwinski, L., Duan, X.J., *et al.*, 2002. DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Res* 30 (1), 303–305.
- Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* 80, 1715–1735.
- Xu, J., Peng, J., Zhao, F., 2009. Template-based and free modeling by RAPTOR ++ in CASP8. *Proteins* 77 (Suppl. 9), 133–137.
- Yang, J., Roy, A., Zhang, Y., 2015. The I-TASSER Suite: Protein structure and function prediction. *Nature Methods* 12, 7–8.
- Yan, R., Si, J., Wang, C., Zhang, Z., 2009. DescFold: A web server for protein fold recognition. *BMC Bioinform* 10, 416.
- Zanzoni, A., Montecchi-Palazzi, L., Quondam, M., *et al.*, 2002. MINT: A Molecular INTERaction database. *FEBS Lett* 513 (1), 135–140.
- Zou, Y., Wang, Y., Wang, F., *et al.*, 2017. Discovery of potent IDO1 inhibitors derived from tryptophan using scaffold-hopping and structure-based design approaches. *Eur J Med Chem* 138, 199–211.

Relevant Websites

<http://www.abren.net/asaview/>

ASAView: Solvent Accessibility Graphics for proteins.

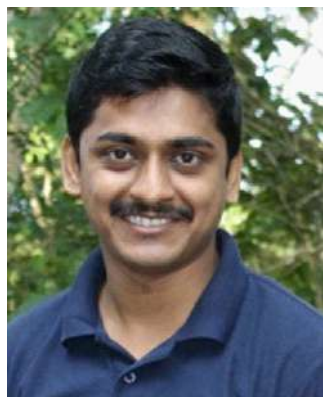
<http://www.iitm.ac.in/bioinfo/protein/folding/foldingrace.html>

Indian Institute of Technology Madras.

Biographical Sketch



M. Michael Gromiha received his PhD in Physics from Bharathidasan University, India and served as STA fellow, RIKEN Researcher, Research Scientist and Senior Scientist at Computational Biology Research Center, AIST, Japan till 2010. Currently, he is working as an Associate Professor at Indian Institute of Technology (IIT) Madras, India. His main research interests are structural analysis, prediction, folding and stability of globular and membrane proteins, protein interactions and development of bioinformatics databases and tools. He has published over 200 research articles, 40 reviews, 5 editorials and a book on Protein Bioinformatics: From Sequence to Function by Elsevier/Academic Press. His papers received more than 9000 citations and h-index is 51. He is an Associate Editor of BMC Bioinformatics as well as Editorial Board Member of Scientific Reports, Biology Direct, Journal of Bioinformatics and Computational Biology and Current Computer Aided Drug design. He has received several awards including Oxford University Press Bioinformatics prize, Okawa Science Foundation Research Grant, Young Scientist Travel awards from ISMB, JSPS, AMBO, ICTP etc., Best paper award at ICIC2011, ICTP Associateship award, ICMR International fellowship for Senior Biomedical Scientists, INSA senior scientist award, Best paper award in Bioinformatics by Department of Biotechnology, India, Institute Research and Development Award at IIT Madras and Outstanding Performance award from Initiative for Parallel Bioinformatics (IPAB), Tokyo Institute of Technology, Japan. He is a member of the National Academy of Sciences, India.



R. Nagarajan completed his PhD degree in Computational Biology from Department of Biotechnology, Indian Institute of Technology Madras, India. He is currently working as a Postdoctoral Research Fellow in Vanderbilt Vaccine Center, Vanderbilt University Medical Center, USA. His main research interests are computational analysis of protein–nucleic acid interactions, recognition mechanism, protein aggregation, computational immunology and development of prediction algorithms, web servers and databases. He has 14 publications (including two book chapters) in reputed journals and received more than 100 citations with h-index 6. He has received Research award from IIT Madras, AU-CBT Excellence award from Biotech Research Society of India, Incentive award from DBT, Govt of India and travel awards from International Society of Computational Biology and Asia Pacific Bioinformatics network.



S. Selvaraj received his PhD degree in Biophysics from the University of Madras, Chennai, India. He has worked as a Collaborative Scientist in RIKEN Life Science Centre, Tsukuba, Japan, and later as Assistant Professor in the Department of Bioinformatics, School of Life Sciences, Bharathidasan University, Tiruchirappalli, India. Post-retirement he has been awarded the Emeritus Fellowship by the University Grants Commission, New Delhi, India. He has published about 60 research articles and 6 reviews. His publications have received more than 2000 citations with h-index of 20. He has been awarded the National Associateship by the Department of Biotechnology, Government of India.

Protein Structure Databases

David R Armstrong, John M Berrisford, Matthew J Conroy, Alice R Clark, Deepti Gupta, and Abhik Mukhopadhyay, Protein Data Bank in Europe, EMBL-EBI, Hinxton, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

The Value of Understanding Protein Structure

Our scientific understanding of the molecular basis of life has expanded dramatically since the very first macromolecular structures were determined in late 1950s. These huge achievements in building atomic models of myoglobin (Kendrew *et al.*, 1958) and hemoglobin (Perutz *et al.*, 1960) paved the way for resolving structures of more complex proteins. More recently the structures of enormous complexes such as the ribosome and drug targets such as G protein-coupled receptors have been determined. These highly informative structures have provided answers to the basic understanding of the molecular machinery in many biological processes and have contributed enormously to human health. An early example of drug design informed by knowledge of the target protein structure is that of the carbonic anhydrase inhibitor dorzolamide which was approved in 1995 (Kubinyi, 1999). Several anti-virals directed against HIV protease were designed using mainly structure-based methods (Greer *et al.*, 1994). The 22 Nobel prizes awarded in the field of structural biology between 1946 and 2016 highlight its importance.

While technological advances and talented scientists have been key to these discoveries, no small part has been played by the structural biology community's willingness to make their data, in the form of atomic models, freely available. To enable this, the community has instigated and supported resources which store and distribute these data. Some of these resources archive experimental data and make it available, others take this archive data and classify it according to their own analyses and annotations. This chapter reviews some of these resources, all of which are available online. URLs of those mentioned are given in Table 1.

The Protein Data Bank

The oldest and largest database of protein structure is the Protein Data Bank (PDB) which archives, curates and distributes experimentally determined structural information. Analysis of citations of the PDB archive, and the data it contains indicate how vital this resource is to the biomedical community (Bousfield *et al.*, 2016). In maintaining a centralised, consistent archive of protein structures, the PDB has played a crucial role in the field of structural bioinformatics (Lesk and Chothia, 1980; Chothia and Lesk, 1986).

The PDB is a freely accessible repository for experimentally determined 3D structures of proteins, nucleic acids, and complex assemblies. Many journals now require scientists to have deposited their data with the PDB before a manuscript is accepted for publication. When the PDB was formed as a collaboration between the Brookhaven National Laboratory (BNL) in the US, and the Cambridge Crystallographic Data Center in the UK (Protein Data Bank, 1971), it contained just seven structures. All of these were of proteins and solved by X-ray crystallography. 46 years later, the archive contains more than 130,000 structures of over 40,000 unique macromolecules (see Fig. 1). Methodology too has expanded and the PDB archives structures determined by several experimental techniques. Around 90% of structures in the archive are solved by diffraction techniques with structures solved by nuclear magnetic resonance (NMR) and electron microscopy (EM) making up most of the remaining 10%. Though historically this was not the case, experimental data from which the models are derived must also be deposited to support the atomic coordinates. In the case of diffraction studies, structure factors are deposited to the PDB and for NMR derived models, the PDB archives chemical shifts and geometric restraints. Further NMR data can be archived with the Biological Magnetic Resonance Data Bank (Ulrich *et al.*, 2008). In the case of electron microscopy studies, data are deposited to the Electron Microscopy Data Bank (Tagari *et al.*, 2002).

By 1977 the PDB had added a further distribution center in Japan (Bernstein *et al.*, 1977), and over time multiple sites mirrored PDB data- a common phenomenon in the days of low speed web connections. PDB format data could be deposited either to BNL or the Macromolecular Structure Database (MSD) at the European Bioinformatics Institute (Keller *et al.*, 1998; Sussman *et al.*, 1999). In 1998 the Research Collaboratory for Structural Bioinformatics (RCSB) took over from BNL as the US PDB center. By the turn of the millennium, the Protein Data Bank Japan (PDBj) based at the Institute for Protein Research, Osaka University was also handling deposition of PDB data. In 2002, to recognise the growing international nature of structural biology and its data archiving activities, the Worldwide Protein Data Bank (wwPDB) was formed (Berman *et al.*, 2003). Founder members were RCSB, MSD (now Protein Data Bank in Europe (PDBe)) and PDBj. In 2008, Biological Magnetic resonance Bank (BMRB (Markley *et al.*, 2008)) which deals specifically with NMR data, became a wwPDB partner. The history of the PDB and wwPDB is well documented elsewhere (Berman *et al.*, 2016; Berman *et al.*, 2014; Berman *et al.*, 2012; Meyer, 1997). The wwPDB manages the PDB archive and ensures that the PDB is freely and publicly available to the global community, conforming to the FAIR (Findable, Accessible, Interoperable, and Re-usable) guiding principles (Wilkinson *et al.*, 2016).

Depositing data

Since 2014, data has been deposited to the wwPDB via a unified deposition interface across all wwPDB sites. This system, OneDep (Young *et al.*, 2017), directs data depositions to the relevant wwPDB partner site for processing based on geography. Thus RCSB

Table 1 URLs of resources described in the text.

<i>Resource</i>	<i>url</i>
wwPDB resources	
Worldwide PDB	wwpdb.org
Protein Data Bank in Europe	PDBe.org
Protein Data Bank Japan	PDBj.org
RCSB PDB	rcsb.org
BioMagResBank	BMRB.org
EMDB Resources	
EBI	emdb-empair.org
RCSB	emsearch.rutgers.edu
PDBj	pdbj.org/emnavi
EmDataBank	emdatabank.org
Other Experimental Archives	
SASBDB	sasbdb.org
BIOISIS	bioisis.net
PRIDE	www.ebi.ac.uk/pride/
Model repositories	
SWISS-MODEL Repository	swissmodel.expasy.org/repository
Protein Model Portal	proteinmodelportal.org
Secondary protein structure atlases	
PDBsum	www.ebi.ac.uk/pdbsum/
OCA	oca.weizmann.ac.il/oca-bin/ocamain
Proteopedia	proteopedia.org/
JenaLib	jenalib.leibniz-fli.de/
MMDB	ncbi.nlm.nih.gov/structure/
Fold Classification Databases	
SCOP	scop.mrc-lmb.cam.ac.uk/scop/
SCOPe	scop.berkeley.edu
CATH	cathdb.info/
ECOD	prodata.swmed.edu/ecod/
ArchDB	sbi.imim.es/archdb/
Subject-specific resources	
VIPERdb	viperdbscripps.edu/
GPCRdb	gpcrdb.org/
PyIgClassify	dunbrack2.fccc.edu/PyIgClassify/
SABDb	opig.stats.ox.ac.uk/webapps/sabdab-sabpred/
SpliProt3D	iimcb.genesilico.pl/SpliProt3D/
PDBTM	pdbtm.enzim.hu/
Membrane Protein Data Bank	www.lipidat.tcd.ie/
MemProt	memprot.bicpu.edu.in/
Membrane Proteins of Known Structure	blanco.biomol.uci.edu/mpstruc
Indian PDB	iris.physics.iisc.ernet.in/ipdb/
Non-structural resources which integrate structural data	
COSMIC 3D	cancer.sanger.ac.uk/cosmic3d
EBI search	www.ebi.ac.uk/
NCBI	ncbi.nlm.nih.gov/
Open Targets	opentargets.org/
Genome3D	genome3d.eu/

PDB handle data from the Americas and Oceania, PDBj process data from Asia and PDBe annotate all data from Europe and Africa. NMR data not linked to a PDB structure deposition or additional experimental data (e.g., free induction decay) can be deposited directly to BMRB.

Once deposited, data can be held privately for up to one year and are usually released on publication of the accompanying paper. the PDB archive is updated weekly at 00.00 UTC each Wednesday and the complete updated archive is available from the master FTP site (see Relevant Websites section) or from any of the wwPDB partners.

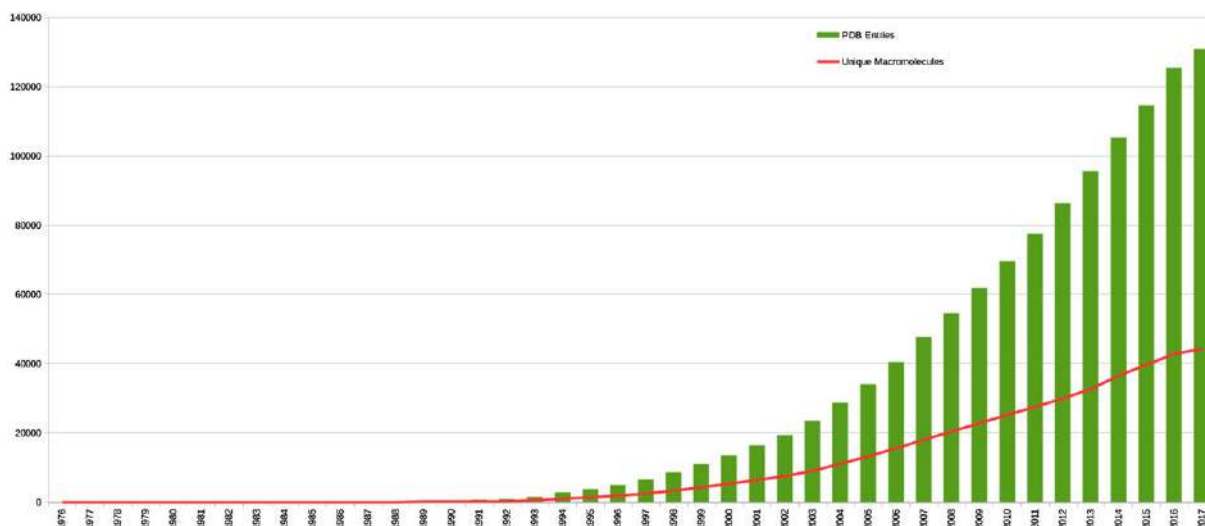


Fig. 1 Growth of the PDB archive over time. The green bars show total number of PDB entries and the red line unique macromolecules (as determined by standardised molecule names).

Curation and validation

Atomic coordinates merely define the shape of a molecule. To make the structure data truly useful, they must be related to other information (Gerstein, 2000). Thus structures are almost always accompanied by a host of metadata. In the PDB archive, these are either entered by the scientists solving the structure, or added by the curation procedures of wwPDB (Dutta *et al.*, 2008; Sen *et al.*, 2014). These annotations include details of the experiment performed to produce the model, cross references to polymer sequence databases, and any discrepancies between the polymer sequences in the model and in a reference database. Where a structure is described in a scientific paper, the citation is included.

Curation also includes adding information about the quaternary structure of the entry. The convention in crystallography is to deposit only the smallest repeating unit of the crystal to the PDB (the asymmetric unit, ASU) thus the true quaternary structure may contain only a part of the ASU or consist of multiple ASUs.

There is frequently a misconception that the atomic coordinates define the absolute positions of all the atoms in a given macromolecule. It should be remembered however that the coordinates are a model which is an interpretation of experimental data. Some data may enable a more precise model to be built than other data. Users of structural databases need to be aware of the limitations of the model when drawing conclusions from it. To address this issue, wwPDB convened expert task forces representing the structural biology community which published their recommendations on how the wwPDB should assess the quality of structures in the PDB (Read *et al.*, 2011; Henderson *et al.*, 2012; Montelione *et al.*, 2013; Adams *et al.*, 2016). Following these recommendations, extensive validation reports now accompany the structures in the PDB archive.

File formats

Early in the history of structural biology, as the possibility of viewing atomic coordinates on a computer rather than as a physical model became reality, the community realised that standards were needed if structure data were to be read computationally between different research groups. Thus the PDB file format was introduced using the 80-column Hollerith card format.

The PDB format defined the standard for representation of atomic coordinates in structural biology for decades and as a human readable text file, is still used by many today. The historical fixed width of the PDB format fields imposes severe limitations for modern structural biology and in 2014 the wwPDB officially moved to use PDBx/mmCIF (Bourne *et al.*, 1997; Westbrook and Bourne, 2000) as the master format for the PDB archive. This format is free of the limitations inherent in the PDB format and is much more extensible to record the diverse metadata which accompanies modern coordinate data.

PDB data is also distributed by the wwPDB in PDBML format (Westbrook *et al.*, 2005), which is an XML-like format, and as RDF format. Both of these are direct translations of the PDBx/mmCIF formatted files.

Access to PDB Data

Websites of wwPDB Members

“The PDB” refers to the FTP archive of files distributed by the wwPDB partners. While the wwPDB website provides details of archive policies and procedures, it is the wwPDB partners who provide tools to search, download, visualize, and analyse protein structures and annotations from the archive. Data from each entry are presented on a series of pages which have been termed

'atlases' detailing the structure and its annotations. Whilst hosting identical data, each site develops its own website, providing unique views of the primary data, and a variety of tools and resources to analyse that data. All sites also have RESTful API calls which grant programmatic access to the archive. A brief summary of the wwPDB partner sites is given below.

Protein Data Bank in Europe

The Protein Data Bank in Europe (PDBe) (Velankar *et al.*, 2016) is an interactive and image-rich site providing access to all data in the PDB archive via a faceted search system. Search results can be ranked based on a series of metrics including the quality of structures. Each entry is detailed on its own series of pages on which macromolecules can be viewed in three dimensions using the in-browser viewer LiteMol. Visualisation of electron density maps for X-ray structures and electric potential maps for those derived by electron microscopy is possible via this viewer. Annotations such as sequence and structure domains, and validation data can be displayed on the model in this viewer and also on sequence and topology views, which are all interlinked (see Fig. 2).

PDBe works with Europe PMC to list publications which cite the paper describing a particular structure, as well as those which mention the PDB code but do not cite the paper.

Advanced tools to analyse the data include PDBePISA (Krissinel and Henrick, 2007), which assesses stability of interactions within a crystal structure, and PDBeFold (Krissinel and Henrick, 2004), which is used to search for structures of a similar three dimensional topology.

Structure integration with function, taxonomy and sequence (SIFTS)

PDBe, along with UniProt (The UniProt Consortium, 2017) produces the SIFTS resource (Velankar *et al.*, 2013) which maintains up-to-date cross-references between the UniProt sequence database and the sample sequence in the PDB entry. Through mapping

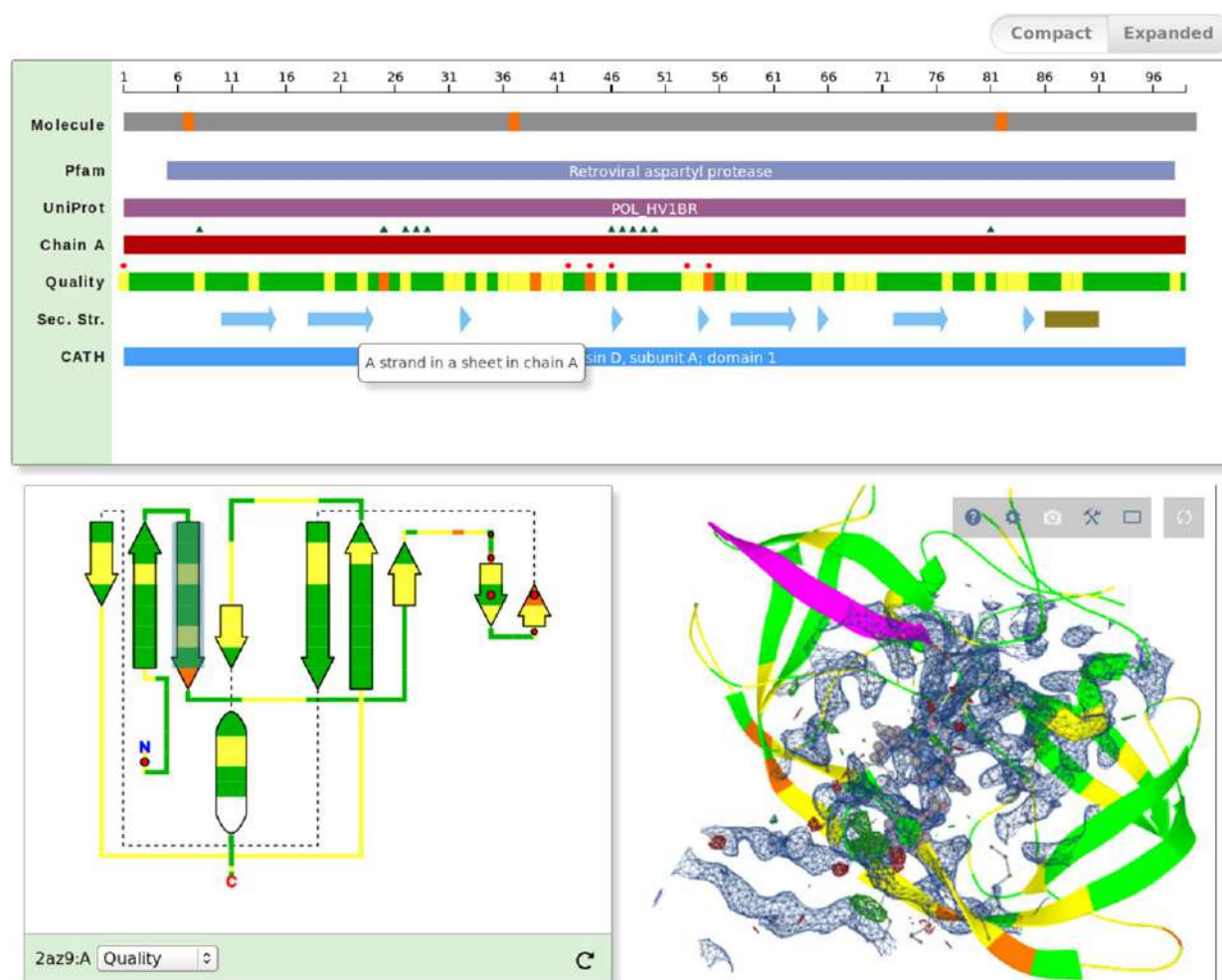


Fig. 2 1D, 2D and 3D interactive views of a HIV protease (PDB entry 2az9) on the PDBe website. The topology diagram and 3D structure are shown coloured by model geometry quality according to the wwPDB validation report and electron density is displayed.

to UniProt the SIFTS service adds cross references to Gene Ontology (GO) (Jones *et al.*, 2014; Ashburner *et al.*, 2000) terms, to sequence motifs from InterPro (Jones *et al.*, 2014), and Pfam sequence domains (Finn *et al.*, 2016). SIFTS also collates cross references to structural domains (SCOP, CATH). This service is used by the other wwPDB partners to maintain up-to date cross-references.

The Protein Data Bank Japan

The Protein Data Bank Japan (PDBj) (Kinjo *et al.*, 2017) provides search, advanced search and visualisation facilities for all entries in the PDB, not only in English but also in Japanese, Chinese and Korean. On each entry page, structures can be viewed in 3D in a variety of molecular viewers. The layout of the PDBj site is user-configurable and all PDB entries are provided in Resource Description Framework (RDF) format.

A variety of advanced services are supplied by PDBj including calculations of electrostatic surfaces and eF-seek, which searches protein structures for similar binding sites. GIRAF (Kinjo and Nakamura, 2007) finds similarities in small molecule binding sites between proteins which might lack detectable sequence similarity. Omokage is a PDBj service which finds structures of similar shapes across PDB and EMDB archives (Suzuki *et al.*, 2016).

Research Collaboratory for Structural Bioinformatics PDB

The Research Collaboratory for Structural Bioinformatics (RCSB) PDB website (Berman *et al.*, 2002) also has a search, advanced search, and pages for each structure in the PDB archive. Search results can be saved in customisable reports. Each PDB entry has a summary page (see Fig. 3) with data described across a series of subpages which tabulate data and annotations, including sequence and structure similarity with other molecules in the PDB archive. Structures can be viewed in 3D using the online NGL viewer (Rose and Hildebrand, 2015) and various other visualisation programs. Recent updates have included linking to other databases

Structure Summary | 3D View | Annotations | Sequence | Sequence Similarity | Structure Similarity | Experiment

Biological Assembly 1

2AZ9

HIV-1 Protease NL4-3 1X mutant
DOI: 10.2210/pdb2az9/pdb

Classification: [HYDROLASE / HYDROLASE INHIBITOR](#)
Deposited: 2005-09-09 **Released:** 2006-02-28
Deposition author(s): [Heaslet, H.](#), [Kutilek, V.](#), [Morris, G.M.](#), [Lin, Y.-C.](#), [Elder, J.H.](#), [Torbett, B.E.](#), [Stout, C.D.](#)
Organism: [Human immunodeficiency virus 1](#)
Expression System: Escherichia coli
Mutation(s): 3

Experimental Data Snapshot
Method: X-RAY DIFFRACTION
Resolution: 2.5 Å
R-Value Free: 0.302
R-Value Work: 0.248

wwPDB Validation | 3D Report | Full Report

Metric	Percentile Ranks	Value
Rfree		0.255
Clashscore		17
Ramachandran outliers		3.1%
Sidechain outliers		3.7%
RSRZ outliers		6.1%

Legend:
 ■ Percentile relative to all X-ray structures
 ■ Percentile relative to X-ray structures of similar resolution

Literature | Download Primary Citation

Structural Insights into the Mechanisms of Drug Resistance in HIV-1 Protease NL4-3

[Heaslet, H.](#), [Kutilek, V.](#), [Morris, G.M.](#), [Lin, Y.-C.](#), [Elder, J.H.](#), [Torbett, B.E.](#), [Stout, C.D.](#)
 (2006) J.Mol.Biol. 356: 967-981
PubMed: 16403521 [Search on PubMed](#)
DOI: 10.1016/j.jmb.2005.11.094
 Primary Citation of Related Structures: 2AZ8 2AZ9 2AZB 2AZC

PubMed Abstract:
 The development of resistance to anti-retroviral drugs targeted against HIV is an increasing clinical problem in the treatment of HIV-1-infected individuals. Many patients develop drug-resistant strains of the virus after treatment with inhibitor cocktails (HAART therapy), which include multiple protease

View in 3D: [NGL](#) or [JSmol](#) or [PV](#) (in Browser)

Standalone Viewers
[Simple Viewer](#) | [Protein Workshop](#) | [Ligand Explorer](#) | [Kiosk Viewer](#)

Protein Symmetry: Cyclic - C2 (View in 3D)
Protein Stoichiometry: Homo 2-mer - A2
 Biological assembly 1 assigned by authors

Macromolecule Content
 • Unique protein chains: 1

Fig. 3 Summary page for HIV-1 protease PDB entry 2az9 at the RCSB PDB website.

and mapping human proteins to their locations on the genome (Rose *et al.*, 2017). Mentions of the PDB code in pubmed are plotted per year.

RCSB enables pairwise structure and sequence alignments using a variety of different algorithms.

Statistics of the PDB archive are readily available at RCSB displayed as interactive graphs across a wide range of data criteria.

In common with its partner sites, RCSB has the facility to search the PDB archive for small molecules which are bound to macromolecules.

Biological Magnetic Resonance Data Bank (BMRB)

The Biological Magnetic Resonance Data Bank (Ulrich *et al.*, 2008) collects and distributes data from nuclear magnetic resonance experiments on proteins and other biological molecules. In addition to the atomic model and data directly used to derive it, BMRB serves as a repository for other kinds of NMR experimental data, such as kinetic and NMR relaxation parameters, and free induction decay (FID) data. BMRB webpages for NMR derived protein structures display validation data and links to the experimental data in addition to the model. BMRB has mirror sites in Osaka and Florence in addition to the master site in Wisconsin.

Protein Structure Databases outside of the wwPDB

In addition to members of the wwPDB consortium, several other databases archive protein structural models and/or the experimental data used to derive them. Some of these are listed below.

Electron Microscopy Data Bank

The Electron Microscopy Data Bank (EMDB) is a repository for three dimensional volumes of macromolecular complexes and subcellular structures derived by electron microscopy. It was founded by the MSD team (now PDBe) at EBI in 2002 (Tagari *et al.*, 2002). EMDB archives data from a range of electron microscopic techniques including single-particle reconstruction, tomography, and electron crystallography. The scale of data in the EMDB archive ranges from tomography of whole cells, to electron crystallographic studies of peptides at atomic resolution. In the last few years, improvements in instrumentation and software have been enormously beneficial to the field. These advances, termed the ‘resolution revolution’ (Kühlbrandt, 2014) have resulted in a huge increase in both the quantity of data in EMDB and its quality, certainly in terms of resolution (Patwardhan, 2017).

EMDataBank, a collaboration between PDBe at EMBL-EBI, the RCSB PDB and the National Center for Macromolecular Imaging (NCMI) (Lawson *et al.*, 2011), has pushed forward data standards in the EM community. Outcomes have included EMDB files being archived jointly in the wwPDB FTP archive, electron microscopy validation reports and the integration of EM deposition with the wwPDB deposition system allowing simultaneous deposition of EM maps to EMDB and the fitted model to PDB. EMDB has a weekly release cycle, synchronized with that of the wwPDB.

Accessing EMDB data

Whilst not formally a wwPDB partner, the data in EMDB archive is available in the wwPDB FTP tree and it can be further accessed through four websites: the EMDB pages at EMBL-EBI, at the RCSB PDB website, PDBj and via the emdatabank.org website. Each website presents the data in different ways. A summary is provided below.

EMDB at EBI provides an advanced search, with EM maps available for download or viewable online using a number of tools and viewers. Many of these visualisation tools also serve as validation tools. In the visual analysis page (Lagerstedt *et al.*, 2013), the map data is presented to the user as a surface, projection, slices and overlaid with the fitted model. The volume slicer tool (Salavert-Torres *et al.*, 2016) is of particular use when visualising tomogram volumes and allows navigation through the slices of a tomogram. Statistics for the EMDB archive are provided in live graphs.

PDBj also provides a search functionality (including shape recognition) and access to EMDB data through the EM Navigator which also provides visual analysis and validation tools. Maps can be viewed as images and movies from UCSF Chimera (Pettersen *et al.*, 2004), the session files for which are available for download. The Yorodumi viewer allows interactive viewing of the map and the PDB model. To analyse overall trends of the data in the EMDB archive an interactive statistics table is provided. The RCSB PDB provide a basic search tool, links to download and view the data in the EMDB.

The EMDataBank site provides an access point for both deposition to and retrieval of data from the EMDB archive in addition to the derived models in the PDB. It also provides data for various ‘challenges’ which are helping drive developments in the electron microscopy field.

Solution Scattering Databases

X-ray and neutron scattering techniques can provide data on the size and molecular envelope (i.e., shape) of a macromolecule. Small Angle Scattering Biological Databank (SASBDB) (Valentini *et al.*, 2015) and BIOISIS (Hura *et al.*, 2009) are repositories of structural information from these experimental techniques. They allow researchers to deposit and access experimental scattering

data and derived models from the scattering experiment. Both provide experimental and derived information and other relevant manually curated data such as experimental conditions, sample details and structural models that have been provided by the depositor during the time of data deposition.

Hybrid and Integrative Techniques

With the advancement of scientific methods, researchers working on challenging biological systems are now able to use a whole suite of different experimental techniques, sometimes supplemented by theoretical data, to determine three dimensional structures of proteins and large complexes. Such approaches are referred to as integrative or hybrid methods (Sali *et al.*, 2015). For instance, mass spectrometry and cross linking data can give insights into a protein's fold or the composition of a protein-protein complex. PRIDE at EBI (Vizcaíno *et al.*, 2016) archives such mass spectrometry data. More generally, where such data should be archived, and vitally, how they should be validated, remains a challenge for the scientific community.

Modelling Archives

On the advice of the community, from 2006 the wwPDB ceased accepting structures determined *in silico* (i.e., by homology or *ab initio* methods) (Berman *et al.*, 2007) and such models which were in the PDB archive were removed. Nevertheless, homology modelling and protein structure prediction is a critical area of research in structural biology with competitions such as CASP (Kryshtafovych *et al.*, 2014) and CAPRI (Lensink *et al.*, 2016) assessing structure prediction and docking methodologies. There is therefore a community demand to store the information collected from theoretical protein modelling.

SWISS-MODEL Repository

The SWISS-MODEL repository (Bienert *et al.*, 2017) is a database of annotated 3D protein structure models generated by the SWISS-MODEL homology-modelling pipeline. Currently the repository contains models for protein sequences in the UniProt database as well as experimentally determined structures from PDB with mapping to UniProt. If a particular protein is absent from the repository, users have the option to attempt building the model. Model quality is indicated over a variety of validation criteria.

Protein Model Portal

Protein Model Portal (Haas *et al.*, 2013) provides access to experimentally determined protein structures and also theoretical models based on sequences available in UniProt. Currently, over 5 million sequences are covered by a model. Crucially, the service also gives an indication of the quality of the model.

Other services exist which will generate a protein model based on homology, for example I-TASSER (Yang *et al.*, 2015) and Phyre2 (Yang *et al.*, 2015; Kelley *et al.*, 2015), but unlike the SWISS-MODEL repository these lack archives of previously calculated structures.

'Derived' Protein Structure Databases

A plethora of specialist databases exist which do not archive structures, but take data from the PDB archive and provide further manual or computational analysis of the data. Some use the whole PDB archive, whilst others might focus on specific protein families. Approximately 25% of resources described in the Nucleic Acids Research database issue each year make use of PDB data in some way. An extensive list of protein structure databases is available online at the Nucleic Acids Research website (see Relevant Websites section).

Here we will detail just a few of these as space allows. For the sake of clarity, we have divided these databases into different types.

- (1) Secondary protein structure atlases
- (2) Fold classification databases
- (3) Subject-specific resources

Secondary Protein Structure Atlases

Several resources integrate information from other resources with three dimensional protein structures and provide a wealth of information to the research community. Some could be considered 'atlases' providing individual webpages for each structure in the PDB archive. Often these were created to provide functionality which wwPDB partner sites lacked, but in many cases, this functionality is now present at one or more of the wwPDB partners. While the members of the wwPDB consortium release new data each Wednesday, other sites necessarily do not have access to these data prior to release and will clearly lag behind the wwPDB partners' sites.

PDBsum

PDBsum (Laskowski *et al.*, 1997) is a pictorial database that provides an overview of the contents of each entry in the PDB archive. It provides several tools to analyse three dimensional macromolecular structures. PDBsum's Ligplot tool gives a pictorial description of a ligand binding site while Nucplot displays interactions between proteins and nucleic acids, and the Prot-prot tool generates a schematic diagram of protein-protein interfaces.

OCA

OCA displays information from the PDB archive and integrates data from several other sources. Its front page is an advanced search form for users to generate specific search queries. Unusually for protein structure databases, the pages for PDB entries in OCA are image free.

Proteopedia

Proteopedia (Hodis *et al.*, 2010) is a wiki-based encyclopedia of macromolecular structures in which annotations are community sourced. Thus the content of each page is extremely variable. Each page has a 3D structure visualiser adjacent to user-editable text describing the structure. The 3D views are scriptable such that users can illustrate a text description with specific views in 3D. Proteopedia has pages that describe generic molecules and complexes in addition to pages for specific PDB entries.

Jenalib

Jenalib (Reichert and Sühnel, 2002) has an emphasis on visualisation and analysis of all entries from the PDB. It provides tools such as SiteDB, GO2PDB and HeteroDB, which are useful to find information on ligand binding sites, Gene ontology and bound small molecules, respectively.

Molecular modelling database

Molecular modelling database, MMDB (Madej *et al.*, 2014) contains experimental structures derived from PDB. MMDB links proteins of similar sequence and structure and displays information about bound small molecules. It allows users to analyse the macromolecular structure in an interactive 3D manner.

Fold Classification Databases

It was recognised relatively early in the history of structural biology that structure is better conserved than sequence (Rao and Rossmann, 1973; Chothia and Lesk, 1986). Several resources classify proteins by their three dimensional fold and, through fold similarity, help infer evolutionary and functional relationships.

SCOP

Structural Classification of Proteins (SCOP) (Murzin *et al.*, 1995) is a database which classifies protein domains according to regions of structural similarity, utilizing both manual and automatic curation. This database uses known structures from the PDB to classify protein folds hierarchically, based on structural and evolutionary relationships. Users can identify known protein folds within their structure of interest and use this information to suggest potential functions of the protein.

More recently, SCOP2 (Andreeva *et al.*, 2014) was launched which presents structural relationships as a network, rather than a tree-like hierarchy. While it is currently a prototype version, SCOP2 enables more complex representations of the relationships between different folds than was possible in the original SCOP. The SCOPextended (SCOPE) database (Chandonia *et al.*, 2017) adds many more structures from the PDB archive to the original SCOP classification.

CATH

The CATH database (Sillitoe *et al.*, 2015) also classifies protein folds. It is named according to the hierarchy used in classification: Class, Architecture, Topography and Homology. Philosophically, CATH and SCOP are very similar, differing in precisely what folds and domains of proteins are identified. CATH also links with Gene3D (Lam *et al.*, 2016) to extend this classification to protein sequences for which a 3D structure is not yet available but can be predicted. In addition to browsing the CATH database, CATH's web pages allow searches based on an uploaded sequence or coordinate file.

ECOD

Both SCOP and CATH databases require manual curation and neither of the two databases has kept up to date with releases in the PDB (though recent developments are addressing this). A further fold classification methodology, Evolutionary Classification of Protein Domains (ECOD) (Cheng *et al.*, 2014; Schaeffer *et al.*, 2017) aims to overcome this problem by offering only automatic annotation of structural domains. ECOD also allows users to upload their own coordinate file for fold classification.

ArchDB

CATH and SCOP classify folds based on the secondary structure elements within a protein, but ArchDB (Bonet *et al.*, 2014) curates instead those non-regular portions of a protein, termed loops, which still have defined structures. Loops are classified on their

geometry into 10 separate types. The interface allows users to find structures of loops based on geometry and classification and to search the database using a variety of other criteria.

Subject-Specific Resources

There are many resources that provide functional annotation for specific types of proteins, dependant on the interests of the creators. The following descriptions provide just a taste of what is available, many are the work of individual labs and their longevity is not guaranteed.

VIPERdb (Carrillo-Tripp *et al.*, 2009) focusses on capsid structures from icosahedral viruses. VIPERdb also provides data on computational analyses on icosahedral virus systems. GPCRdb (Isberg *et al.*, 2017) provides annotations and tools focussing on G-protein coupled receptors.

PyIgClassify (Adolf-Bryfogle *et al.*, 2015) and SAbDab (Dunbar *et al.*, 2014) provide extra annotation on antibodies, SpliProt3D (Korneta *et al.*, 2012) brings together structures and models of the human spliceosome components.

The structures of membrane proteins are also notoriously difficult to solve experimentally but of great biological relevance. Several resources focus on membrane proteins. Among these resources are PDBTM (Kozma *et al.*, 2013) which takes proteins from the PDB with likely membrane spanning regions, classifies them and indicates the probable membrane embedded region. The Membrane Protein Data Bank, MPDB (Raman *et al.*, 2006), MemPROT and “Membrane Proteins of known structure” also list transmembrane structures from the PDB.

Rather than subject specificity, Indian PDB provides a database of structures solved by Indian structural biologists, sorted by institute.

Integrating Structure into the Bigger Picture

Protein structures do not exist without a context, and many databases are integrating structures into their resources or ensuring that they link to relevant structural resources. Thus biologists searching for data relevant to their field of research may encounter protein structure outside of the specialist protein structure databases. A few are mentioned below.

COSMIC 3D

COSMIC 3D maps cancer mutation data from COSMIC (Forbes *et al.*, 2017), a database of somatic mutations in cancer, to three dimensional protein structures, allowing users to visualize the location of these mutations in the three dimensional structure.

EBI search

Whilst all of the data resources at the European Bioinformatics Institute have their own search systems, the EBI-wide search engine provides a facility to search all of these resources, highlighting the interconnectivity of biological data resources. In a similar way, searches at NCBI (NCBI Resource Coordinators, 2016) integrate MMDB, which is a secondary atlas for the PDB (Park *et al.*, 2017).

Open Targets

Open Targets (Koscielny *et al.*, 2017) is a platform which brings together data from a variety of different biomedical data resources, including structure data, with the aim of aiding its user community in identification of potential drug targets.

Genome3D

Genome3D (Lewis *et al.*, 2015) integrates a variety of resources of fold classification and structure prediction to extend structural annotation across the genomes of several model organisms.

Challenges for the Future

Technological advances in crystallography, not least robotics enabling high throughput, microfocus beamlines, free electron lasers and the amazing progress in electron microscopy are resulting in a huge leap in the volume of protein structure data. As mentioned above, integrative and hybrid techniques combine very diverse types of data.

Advances in speed and reduction in cost of genome sequence have resulted in sequences of many related proteins being easily available. New modelling techniques based on the sequence comparisons can predict even novel protein folds (Ovchinnikov *et al.*, 2017).

With such an avalanche of data, protein structure databases have an exciting challenge going forward in archiving these data. They need to be made available in a meaningful way such that specialists and non-specialists alike can locate the data they need and draw relevant conclusions from it.

Acknowledgements

We thank EMBL and the Wellcome Trust for funding and Drs Aleksandras Gutmanas and Sameer Velankar for critical feedback on the manuscript.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Biomolecular Structures: Prediction, Identification and Analyses. Computational Tools for Structural Analysis of Proteins. Data Storage and Representation. Drug Repurposing and Multi-Target Therapies. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Information Retrieval in Life Sciences. Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Protein Structural Bioinformatics: An Overview. Protein Structure Classification. Protein Three-Dimensional Structure Prediction. Rational Structure-Based Drug Design. Secondary Structure Prediction. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Study of The Variability of The Native Protein Structure. The Evolution of Protein Family Databases

References

- Adams, P.D., Aertgeerts, K., Bauer, C., *et al.*, 2016. Outcome of the first wwPDB/CCDC/D3R ligand validation workshop. *Structure* 24 (4), 502–508.
- Adolf-Bryfogle, J., Xu, Q., North, B., Lehmann, A., Dunbrack, R.L. Jr., 2015. PylgClassify: a database of antibody CDR structural classifications. *Nucleic Acids Research* 43 (Database issue), D432–D438.
- Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., Murzin, A.G., 2014. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Research* 42 (Database issue), D310–D314.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics* 25 (1), 25–29.
- Berman, H., Henrick, K., Nakamura, H., 2003. Announcing the worldwide Protein Data Bank. *Nature Structural Biology* 10 (12), 980.
- Berman, H., Henrick, K., Nakamura, H., Markley, J.L., 2007. The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic acids research* 35 (Database issue), D301–D303.
- Berman, H.M., Battistuz, T., Bhat, T.N., *et al.*, 2002. The Protein Data Bank. *Acta Crystallographica D: Biological Crystallography* 58 (Pt 61), 899–907.
- Berman, H.M., Kleywegt, G.J., Nakamura, H., Markley, J.L., 2012. The Protein Data Bank at 40: reflecting on the past to prepare for the future. *Structure* 20 (3), 391–396.
- Berman, H.M., Kleywegt, G.J., Nakamura, H., Markley, J.L., 2014. The Protein Data Bank archive as an open data resource. *Journal of Computer-Aided Molecular Design* 28 (10), 1009–1014.
- Berman, H.M., Burley, S.K., Kleywegt, G.J., *et al.*, 2016. The archiving and dissemination of biological structure data. *Current Opinion in Structural Biology* 40, 17–22.
- Bernstein, F.C., Koetzle, T.F., Williams, G.J., *et al.*, 1977. The Protein Data Bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology* 112 (3), 535–542.
- Bienert, S., Waterhouse, A., de Beer, T.A.P., *et al.*, 2017. The SWISS-MODEL Repository-new features and functionality. *Nucleic Acids Research* 45 (D1), D313–D319.
- Bonet, J., Planas-Iglesias, J., Garcia-Garcia, J., *et al.*, 2014. ArchDB 2014: structural classification of loops in proteins. *Nucleic Acids Research* 42 (Database issue), D315–D319.
- Bourne, P.E., Berman, H.M., McMahon, B., *et al.*, 1997. Macromolecular crystallographic information file. *Methods in Enzymology* 277, 571–590.
- Bousfield, D., McEntyre, J., Velankar, S., *et al.*, 2016. Patterns of database citation in articles and patents indicate long-term scientific and industry value of biological data resources. *F1000Research* 5.
- Carrillo-Tripp, M., Shepherd, C.M., Borelli, I.A., *et al.*, 2009. VIPERdb2: an enhanced and web API enabled relational database for structural virology. *Nucleic Acids Research* 37 (Database issue), D436–D442.
- Chandonia, J.-M., Fox, N.K., Brenner, S.E., 2017. SCOPe: manual curation and artifact removal in the structural classification of proteins - extended Database. *Journal of Molecular Biology* 429 (3), 348–355.
- Cheng, H., Schaeffer, R.D., Liao, Y., *et al.*, 2014. ECOD: an evolutionary classification of protein domains. *PLoS Computational Biology* 10 (12), e1003926.
- Chothia, C., Lesk, A.M., 1986. The relation between the divergence of sequence and structure in proteins. *The EMBO Journal* 5 (4), 823–826.
- Dunbar, J., Krawczyk, K., Leem, J., *et al.*, 2014. SABDab: the structural antibody database. *Nucleic Acids Research* 42 (Database issue), D1140–D1146.
- Dutta, S., Burkhardt, K., Swaminathan, G.J., *et al.*, 2008. Data deposition and annotation at the worldwide protein data bank. *Methods in Molecular Biology* 426, 81–101.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016. The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research* 44 (D1), D279–D285.
- Forbes, S.A., Beare, D., Boutselakis, H., *et al.*, 2017. COSMIC: somatic cancer genetics at high-resolution. *Nucleic Acids Research* 45 (D1), D777–D783.
- Gerstein, M., 2000. Integrative database analysis in structural genomics. *Nature Structural Biology* 7, 960–963.
- Greer, J., Erickson, J.W., Baldwin, J.J., Varney, M.D., 1994. Application of the three-dimensional structures of protein target molecules in structure-based drug design. *Journal of Medicinal Chemistry* 37 (8), 1035–1054.
- Haas, J., Roth, S., Arnold, K., *et al.*, 2013. The Protein Model Portal—a comprehensive resource for protein structure and model information. *Database: The Journal of Biological Databases and Curation* 2013, bat031.
- Henderson, R., Sali, A., Baker, M.L., *et al.*, 2012. Outcome of the first electron microscopy validation task force meeting. *Structure* 20 (2), 205–214.
- Hodis, E., Prilusky, J., Sussman, J.L., 2010. Proteopedia: a collaborative, virtual 3D web-resource for protein and biomolecule structure and function. *Biochemistry and Molecular Biology Education: A Bimonthly Publication of the International Union of Biochemistry and Molecular Biology* 38 (5), 341–342.
- Hura, G.L., Menon, A.L., Hammel, M., *et al.*, 2009. Robust, high-throughput solution structural analyses by small angle X-ray scattering (SAXS). *Nature Methods* 6 (8), 606–612.
- Isberg, V., Mordalski, S., Munk, C., *et al.*, 2017. GPCRdb: an information system for G protein-coupled receptors. *Nucleic Acids Research* 45 (5), 2936.
- Jones, P., Binns, D., Chang, H.-Y., *et al.*, 2014. InterProScan 5: genome-scale protein function classification. *Bioinformatics* 30 (9), 1236–1240.
- Keller, P.A., Henrick, K., McNeil, P., Moodie, S., Barton, G.J., 1998. Deposition of macromolecular structures. *Acta Crystallographica. Section D, Biological Crystallography* 54 (Pt 6 Pt 1), 1105–1108.
- Kelley, L.A., Mezulis, S., Yates, C.M., Wass, M.N., Sternberg, M.J.E., 2015. The PyMol web portal for protein modeling, prediction and analysis. *Nature Protocols* 10 (6), 845–858.

- Kendrew, J.C., Bodo, G., Dintzis, H.M., *et al.*, A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature* 181 (4610), 662–666.
- Kinjo, A.R., Bekker, G.-J., Suzuki, H., *et al.*, 2017. Protein Data Bank Japan (PDBj): updated user interfaces, resource description framework, analysis tools for large structures. *Nucleic Acids Research* 45 (D1), D282–D288.
- Kinjo, A.R., Nakamura, H., 2007. Similarity search for local protein structures at atomic resolution by exploiting a database management system. *Biophysics* 3, 75–84.
- Korneta, I., Magnus, M., Bujnicki, J.M., 2012. Structural bioinformatics of the human spliceosomal proteome. *Nucleic Acids Research* 40 (15), 7046–7065.
- Koscielny, G., An, P., Carvalho-Silva, D., 2017. Open Targets: a platform for therapeutic target identification and validation. *Nucleic Acids Research* 45 (D1), D985–D994.
- Kozma, D., Simon, I., Tusnády, G.E., 2013. PDBTM: protein Data Bank of transmembrane proteins after 8 years. *Nucleic Acids Research* 41 (Database issue), D524–D529.
- Krissinel, E., Henrick, K., 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica D: Biological Crystallography* 60 (Pt 12 Pt 1), 2256–2268.
- Krissinel, E., Henrick, K., 2007. Inference of macromolecular assemblies from crystalline state. *Journal of Molecular Biology* 372 (3), 774–797.
- Kryshtafovych, A., Monastyrskyy, B., Fidelis, K., 2014. CASP prediction center infrastructure and evaluation measures in CASP10 and CASP ROLL. *Proteins* 82 (Suppl 2), 7–13.
- Kubinyi, H., 1999. Chance favors the prepared mind – from serendipity to rational drug design. *Journal of Receptor and Signal Transduction Research* 19 (1–4), 15–39.
- Kühlbrandt, W., 2014. Biochemistry. The resolution revolution. *Science* 343 (6178), 1443–1444.
- Lagerstedt, I., Moore, W.J., Patwardhan, A., *et al.*, 2013. Web-based visualisation and analysis of 3D electron-microscopy data from EMDB and PDB. *Journal of Structural Biology* 184 (2), 173–181.
- Lam, S.D., Dawson, N.L., Das, S., *et al.*, 2016. Gene3D: expanding the utility of domain assignments. *Nucleic Acids Research* 44 (D1), D404–D409.
- Laskowski, R.A., Hutchinson, E.G., Michie, A.D., *et al.*, 1997. PDBsum: a Web-based database of summaries and analyses of all PDB structures. *Trends in Biochemical Sciences* 22 (12), 488–490.
- Lawson, C.L., Baker, M.L., Best, C., *et al.*, 2011. EMDatabank.org: unified data resource for CryoEM. *Nucleic Acids Research* 39 (Database issue), D456–D464.
- Lensink, M.F., Velankar, S., Kryshtafovych, A., *et al.*, 2016. Prediction of homoprotein and heteroprotein complexes by protein docking and template-based modeling: a CAPRI experiment. *Proteins* 84 (Suppl 1), 323–348.
- Lesk, A.M., Chothia, C., 1980. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *Journal of Molecular Biology* 136 (3), 225–270.
- Lewis, T.E., Sillitoe, I., Andreeva, A., *et al.*, 2015. Genome3D: exploiting structure to help users understand their sequences. *Nucleic Acids Research* 43 (Database issue), D382–D386.
- Madej, T., Lanczycki, C.J., Zhang, D., *et al.*, 2014. MMDB and VAST + : tracking structural similarities between macromolecular complexes. *Nucleic Acids Research* 42 (Database issue), D297–D303.
- Markley, J.L., Ulrich, E.L., Berman, H.M., *et al.*, 2008. BioMagResBank (BMRB) as a partner in the Worldwide Protein Data Bank (wwPDB): new policies affecting biomolecular NMR depositions. *Journal of Biomolecular NMR* 40 (3), 153–155.
- Meyer, E.F., 1997. The first years of the Protein Data Bank. *Protein Science: A Publication of the Protein Society* 6 (7), 1591–1597.
- Montelione, G.T., Nilges, M., Bax, A., *et al.*, 2013. Recommendations of the wwPDB NMR validation task force. *Structure* 21 (9), 1563–1570.
- Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology* 247 (4), 536–540.
- NCBI Resource Coordinators, 2016. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 44 (D1), D7–D19.
- Ovchinnikov, S., Park, H., Varghese, N., *et al.*, 2017. Protein structure determination using metagenome sequence data. *Science* 355 (6322), 294–298.
- Park, Y.M., Squizzato, S., Buso, N., Gur, T., Lopez, R., *et al.*, 2017. The EBI search engine: EBI search as a service-making biological data accessible for all. *Nucleic Acids Research*. Available at: <http://dx.doi.org/10.1093/nar/gkx359>.
- Patwardhan, A., 2017. Trends in the Electron Microscopy Data Bank (EMDB). *Acta Crystallographica D: Structural Biology* 73 (Pt 6), 503–508.
- Perutz, M.F., Rossmann, M.G., Cullis, A.F., *et al.*, 1960. Structure of haemoglobin: a three-dimensional Fourier synthesis at 5.5-Å. resolution, obtained by X-ray analysis. *Nature* 185 (4711), 416–422.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF Chimera – a visualization system for exploratory research and analysis. *Journal of Computational Chemistry* 25 (13), 1605–1612.
- Protein Data Bank, 1971. *Protein Data Bank. Nature New Biology* 233, p223.
- Raman, P., Cherezov, V., Caffrey, M., 2006. The Membrane Protein Data Bank. *Cellular and Molecular Life Sciences: CMLS* 63 (1), 36–51.
- Rao, S.T., Rossmann, M.G., 1973. Comparison of super-secondary structures in proteins. *Journal of Molecular Biology* 76 (2), 241–256.
- Read, R.J., Adams, P.D., Arendall, W.B., 3rd, *et al.*, 2011. A new generation of crystallographic validation tools for the protein data bank. *Structure* 19 (10), 1395–1412.
- Reichert, J., Sühnel, J., 2002. The IMB Jena Image Library of Biological Macromolecules: 2002 update. *Nucleic Acids Research* 30 (1), 253–254.
- Rose, A.S., Hildebrand, P.W., 2015. NGL Viewer: a web application for molecular visualization. *Nucleic Acids Research* 43 (W1), W576–W579.
- Rose, P.W., Prlić, A., Altunkaya, A., *et al.*, 2017. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Research* 45 (D1), D271–D281.
- Salavert-Torres, J., Iudin, A., Lagerstedt, I., *et al.*, 2016. Web-based volume slicer for 3D electron-microscopy data from EMDB. *Journal of Structural Biology* 194 (2), 164–170.
- Sali, A., Berman, H.M., Schwede, T., *et al.*, 2015. Outcome of the First wwPDB Hybrid/Integrative Methods Task Force Workshop. *Structure* 23 (7), 1156–1167.
- Schaeffer, R.D., Liao, Y., Cheng, H., Grishin, N.V., *et al.*, 2017. ECOD: new developments in the evolutionary classification of domains. *Nucleic Acids Research* 45 (D1), D296–D302.
- Sen, S., Young, J., Berrisford, J.M., *et al.*, 2014. Small molecule annotation for the Protein Data Bank. *Database: the Journal of Biological Databases and Curation* 2014, bau116.
- Sillitoe, I., Lewis, T.E., Cuff, A., *et al.*, 2015. CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research* 43 (Database issue), D376–D381.
- Sussman, J.L., Abola, E.E., Lin, D., *et al.*, 1999. The protein data bank. Bridging the gap between the sequence and 3D structure world. *Genetica* 106 (1–2), 149–158.
- Suzuki, H., Kawabata, T., Nakamura, H., 2016. Omokage search: shape similarity search service for biomolecular structures in both the PDB and EMDB. *Bioinformatics* 32 (4), 619–620.
- Tagari, M., Newman, R., Chagoyen, M., Carazo, J.M., Henrick, K., *et al.*, 2002. New electron microscopy database and deposition system. *Trends in Biochemical Sciences* 27 (11), 589.
- The UniProt Consortium, 2017. UniProt: the universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Ulrich, E.L., Akutsu, H., Doreleijers, J.F., *et al.*, 2008. BioMagResBank. *Nucleic Acids Research* 36 (Database issue), D402–D408.
- Valentini, E., Kikhney, A.G., Previtali, G., Jeffries, C.M., Svergun, D.I., *et al.*, 2015. SASBDB, a repository for biological small-angle scattering data. *Nucleic acids research* 43 (Database issue), D357–D363.
- Velankar, S., Dana, J.M., Jacobsen, J., *et al.*, 2013. SIFTS: structure integration with function, taxonomy and sequences resource. *Nucleic Acids Research* 41 (Database issue), D483–D489.
- Velankar, S., van Ginkel, G., Alhroub, Y., *et al.*, 2016. PDBe: improved accessibility of macromolecular structure data from PDB and EMDB. *Nucleic Acids Research* 44 (D1), D385–D395.
- Vizcaino, J.A., Csordas, A., del-Toro, N., *et al.*, 2016. update of the PRIDE database and its related tools. *Nucleic Acids Research* 44 (D1), D447–D456.

- Westbrook, J., Ito, N., Nakamura, H., Henrick, K., Berman, H.M., 2005. PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics* 21 (7), 988–992.
- Westbrook, J.D., Bourne, P.E., 2000. STAR/mmCIF: an ontology for macromolecular structure. *Bioinformatics* 16 (2), 159–168.
- Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J.J., *et al.*, 2016. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data* 3, 160018.
- Yang, J., Yan, R., Roy, A., *et al.*, 2015. The I-TASSER Suite: protein structure and function prediction. *Nature Methods* 12 (1), 7–8.
- Young, J.Y., Westbrook, J.D., Feng, Z., *et al.*, 2017. OneDep: unified wwPDB system for deposition, biocuration, and validation of macromolecular structures in the PDB archive. *Structure* 25 (3), 536–545.

Further Reading

- Branden, C., Tooze, J., 1991. *Introduction to Protein Structure*. New York: Garland publishing.
- Burley, S.K., Berman, H.M., Kleywegt, G.J., *et al.*, 2017. Protein Data Bank (PDB): the single global macromolecular structure archive. *Methods in Molecular Biology* 1607, 627–641.
- Lamb, A.L., Kappock, T.J., Silvaggi, N.R., 2015. You are lost without a map: navigating the sea of protein structures. *Biochimica et Biophysica Acta* 1854 (4), 258–268.
- Mackay, J.P., Landsberg, M.J., Whitten, A.E., Bond, C.S. Whaddaya know: a guide to uncertainty and subjectivity in structural biology. *Trends in Biochemical Sciences* 42 (2), 155–167.
- Mackenzie, C.O., Grigoryan, G., 2017. Protein structural motifs in prediction and design. *Current Opinion in Structural Biology* 44, 161–167.
- Patwardhan, A., Lawson, C.L., 2016. Databases and archiving for CryoEM. *Methods in Enzymology* 579, 393–412.
- Paxman, J.J., Heras, B., 2017. Bioinformatics tools and resources for analyzing protein structures. *Methods in molecular biology* 1549, 209–220.
- Sillitoe, I., Dawson, N., Thornton, J., Orengo, C., 2015. The history of the CATH structural classification of protein domains. *Biochimie* 119, 209–217.
- Westbrook, J.D., Fitzgerald, P.M.D., 2003. The PDB format, mmCIF, and other data formats.

Relevant Websites

- <http://www.oxfordjournals.org/nar/database/subcat/4/14>
Oxford University Press.
- <ftp.wwpdb.org>
wwPDB.

Protein Structure Classification

Natalie L Dawson, Sayoni Das, Jonathan G Lees, and Christine Orengo, Institute of Structural and Molecular Biology, University College London, London, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction to Protein Structure Classifications

The first three-dimensional protein structure, myoglobin, was solved by X-ray crystallography in the late 1950s. The number of solved protein structures grew at a steady rate and a computational archive was built in 1971 to collect, standardise and distribute this data. This archive, known as the Protein Data Bank (PDB) has been growing at an exponential rate since the mid 1990s, increasing from 3000 structures to 135,000 entries in November 2017. The dramatic increase in the data made many manual analyses impractical. Organising the data into structural and evolutionary classifications, gave biologists insights into common structural units and evolutionary relationships.

Several publicly available, structural classifications have been developed providing information on fold groups, evolutionary relationships and sequence families (see [Table 1](#)). These include: SCOP ([Hubbard *et al.*, 1997](#)), SCOP2 ([Andreeva *et al.*, 2014](#)), CATH ([Orengo *et al.*, 1997](#); [Dawson *et al.*, 2017](#)), ECOD ([Cheng *et al.*, 2014](#)), and HOMSTRAD ([Mizuguchi *et al.*, 1998](#)). These classifications are built on a combination of sequence and structure comparison methods (e.g., SSAP ([Taylor and Orengo, 1989](#)), Dali ([Holm and Sander, 1995](#)), SSM ([Krissinel and Henrick, 2004](#)), and CE ([Shindyalov and Bourne, 2001](#))).

There are also a number of structural comparison web servers that provide lists of structural neighbours, including: the protein structure comparison service, PDBFold at the European Bioinformatics Institute ([Krissinel and Henrick, 2004](#)); VAST at the NCBI ([Madej *et al.*, 1995](#); [Gibrat *et al.*, 1996](#)); Combinatorial Extension (CE) ([Shindyalov and Bourne, 2001](#)) at the RCSB Protein Data Bank; Dali ([Holm and Sander, 1995](#)) at the University of Helsinki.

Table 1 Protein structure classifications and nearest-neighbour resources

Acronym/Name	Description	URL
RCSB PDB Protein Comparison Tool	The Combinatorial Extension method is integrated into the RCSB PDB to compare pairs of protein structures. It compares a structure with the PDB and identifies its nearest neighbours. The RCSB PDB protein comparison tool provides access to CE to perform pairwise structural alignments.	http://www.rcsb.org/pdb/workbench/workbench.do
CATH	CATH assigns protein domain structures into a structural classification hierarchy: Class, Architecture, Topology (fold), Homologous superfamily. A mixture of automated and manual curated steps are used in its continual development.	http://cathdb.info/
Dali server	The Dali server allows users to perform structural comparisons for: one structure against the whole PDB; one structure against user-defined list of PDBs; a list of structures to get an all-against-all comparison.	http://ekhidna2.biocenter.helsinki.fi/dali/
ECOD	Evolutionary Classification of Domains. The classification consists of proteins with experimentally-determined 3D structures and focuses on establishing remote evolutionary relationships.	http://prodata.swmed.edu/ecod/
HOMSTRAD	HOMologous STRucture Alignment Database. Comprises protein structural alignments for homologous families.	http://mizuguchilab.org/homstrad/
PDBeFOLD	Uses the Secondary Structure Matching (SSM) algorithm to compare a given structure with the whole PDB archive, and identifies the nearest structural neighbours.	http://www.ebi.ac.uk/msd-srv/ssm/
SCOP	Structural Classification of Proteins. Classifies protein domains into a hierarchy consisting of: class, fold, superfamily and family.	http://scop.mrc-lmb.cam.ac.uk/scop/
SCOP2	Structural Classification of Proteins 2. A prototype successor of SCOP, which uses a directed acyclic graph to describe complex structural and evolutionary relationships. Each graph node represents a particular type of relationship.	http://scop2.mrc-lmb.cam.ac.uk/
VAST	Vector Alignment Search Tool. Identifies 3D domains for a given query and then finds a list of nearest structural neighbours within the Molecular Modelling Database (MMDB).	https://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml
VAST +	Vector Alignment Search Tool Plus. Uses all the domains identified by VAST for a query structure, then finds macromolecular structures that have a structurally similar biological unit. This differs from VAST, which focuses on individual protein molecules (e.g., domains).	https://www.ncbi.nlm.nih.gov/Structure/vastplus/vastplus.cgi

Modified from Pearl, F.M.G., Sillitoe, I., Orengo, C.A., 2015. Protein Structure Classification. eLS, pp. 1–10.

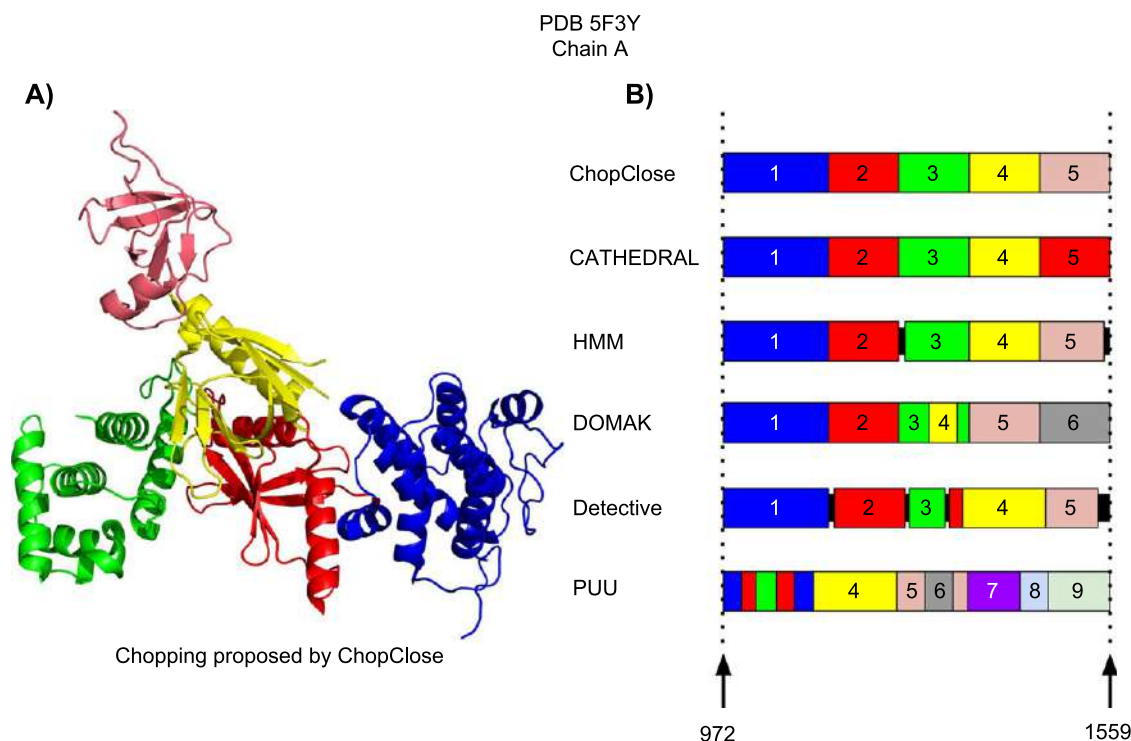


Fig. 1 Figure showing how automatic detection of domains can be non-trivial for multi-domain proteins. (A) The 3D structure of the PDB 5F3Y, chain A. The domain chopping proposed by the method ChopClose is shown, where each colour represents a different domain. (B) The proposed domain choppings by the different algorithms used to identify domains for classification in CATH. The different colours represent different domains proposed by each method, as in (A). See descriptions of the methods below. The chain start (972) and stop (1559) PDB residue labels are shown at the bottom of the figure. The domains have been numbered where space permits.

Protein Domains and Methods Used to Identify Them

Early analysis of protein structures revealed that proteins can have one or more distinct structural units, or domains. A structural domain is created by the protein polypeptide chain folding into a compact unit with its own hydrophobic core. Domains consist of secondary structures (e.g., alpha helices and beta strands) and supersecondary structures (i.e., any structural motif, e.g., helix-turn-helix, beta hairpin, alpha-beta motif), and have an average size of 150 ± 50 residues (see [Pearl et al., 2015](#) for a review). They are typically formed of contiguous parts of the polypeptide chain but may be formed of discontinuous sections. Domains can be defined as semi-independently evolving units, and sometimes have their own independent functions ([Hadley and Jones, 1999](#)).

The multi-domain architecture (MDA) of a protein describes the sequential order of domains within the protein polypeptide chain. Studies have found that over two-thirds of known prokaryotic proteins, and almost 80% of eukaryotic proteins, contain at least two domains ([Teichmann et al., 1998](#); [Gerstein, 1998](#)). Domains within multi-domain proteins can be difficult to identify if they have extensive contacts between them, making it hard to recognise boundaries between the domains. [Fig. 1](#) illustrates the difficulties of identifying domain boundaries in a multi-domain protein when the methods aiding manual curation disagree with each other (see subsequent sections for details on these algorithms). The curator uses these results, and the literature, to decide on the correct domain boundaries.

Although the protein domain structure data is still increasing at an exponential rate, the number of new folds discovered has plateaued in recent years, indicating that existing fold libraries cover the majority of structural space. The comprehensive fold libraries now available can be searched with newly solved structures, using structure based algorithms, to find related domain folds and inherit the domain boundaries (see Section Sequence-based methods for recognising domains in proteins). Protein sequence data, i.e., for domains of known and predicted domain structure, is also very comprehensive and can be searched with powerful HMM-based algorithms to find domain matches and inherit boundaries (see Section Structure-based methods for recognising domains in proteins). CATH and ECOD use their own automated protocols for domain assignment, based on structure- and sequence-based algorithms. In addition, CATH uses a selection of *ab initio* algorithms, which are used to detect domains with novel, uncharacterised folds (see Section *Ab-initio* methods for recognising domains in proteins). These different types of algorithms are described below.

Algorithms to identify protein structural domains

Structure-based methods for recognising domains in proteins

To identify protein structural domains for the CATH resource, the ChopClose automated protocol ([Greene et al., 2007](#)) is first used to scan newly determined protein structures from the PDB against the structures of multidomain proteins in CATH, that have

PDB search results [Job: #12845]

Query Structure



CATH Version 4.1
Algorithm CATHEDRAL
Library CATH_S35

CATHEDRAL found 82 matching structures in CATH (v4.1)

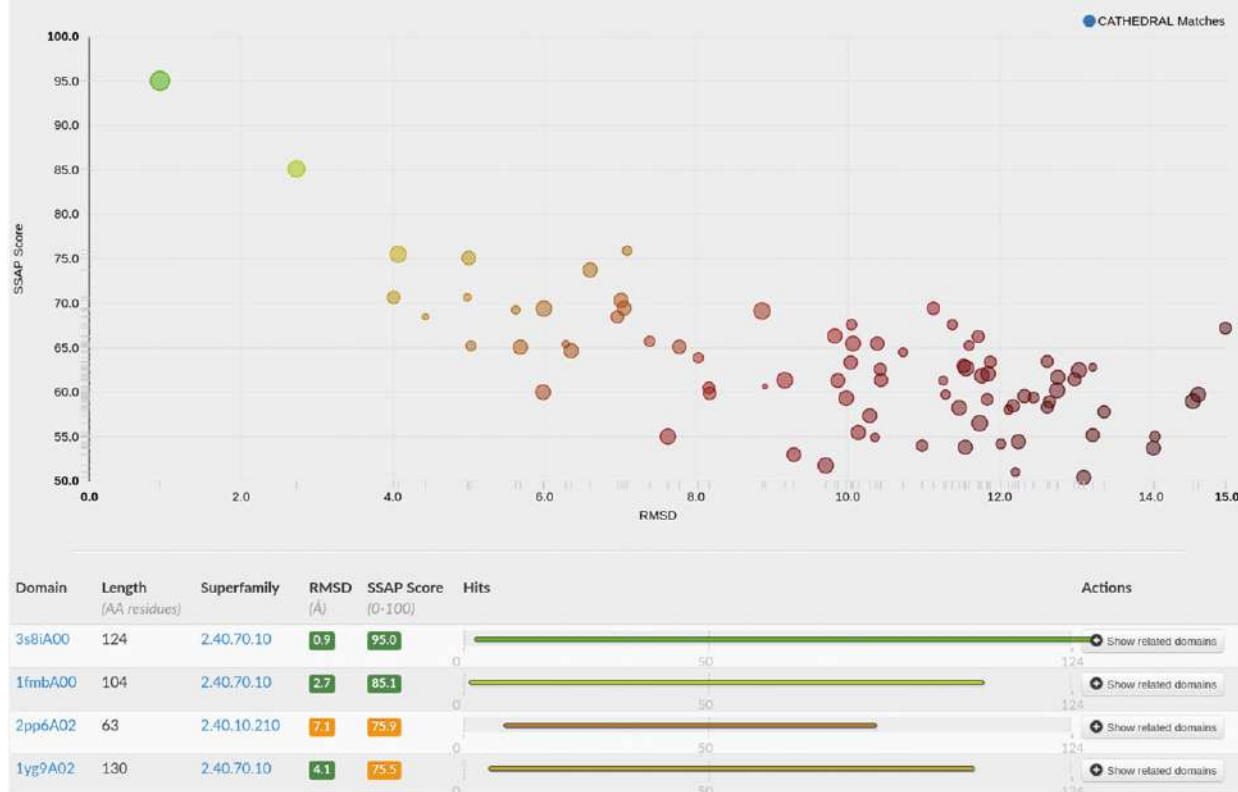


Fig. 2 Results from running a structure-based search with the CATHEDRAL web server using an example case PDB 4RGH, downloaded from the PDB. A total of 82 matching structures were found in CATH with the most confident domain matches shown in green, through to the least confident matches in red.

already had their domain boundaries identified. For query structures that have high structure similarity to multi-domain structures already classified in CATH (a score ≥ 80 by the SSAP algorithm (range 0–100) (Taylor and Orengo, 1989), $\text{RMSD} \leq 6\text{Å}$) the domain boundaries from the match are inherited to the query.

If there is no significant match, the chain is submitted to a number of additional algorithms (structure-based: CATHEDRAL, sequence-based: CATH-HMMscan) to propose domain boundaries (see Sections “Sequence-based methods for recognising domains in proteins”, “Ab-initio methods for recognising domains in proteins”, “Hierarchical Structural Classification”, “Class”, “Architecture”, “Topology or fold similarity”, “Homologous superfamily”, “Homologous family”, “Nearest-Neighbour Clustering”, “Integration of the SCOP And CATH Resources”, “Current Status of the Protein Structural Universe”, “Structural Families in the Genomic Era”). The CATHEDRAL algorithm (Redfern *et al.*, 2007) is used to perform a structure-based search of a query chain against a library of previously classified protein domains in CATH. First, an initial fast secondary structure comparison is run between structures using graph theory to find putative fold matches. These putative-fold-matching domains are realigned to the query with the slower, more accurate SSAP to produce a structural alignment, from which the domain boundaries can be inferred. High scoring matches (i.e., SSAP score ≥ 80 and $\text{RMSD} \leq 6\text{Å}$) provide confidence to infer the domain boundaries of the match (see Fig. 2).

In ECOD, query protein structures from the PDB are first scanned against protein chains and domains in ECOD fast sequence based approaches (see below). If the sequence-based methods do not return any significant match, the query chain is searched against a library of representative domain structures in ECOD using a DaliLite search (Holm and Park, 2000). The domain boundaries are inherited if there is significant structural similarity and an acceptable BLOSUM-based alignment score (Cheng *et al.*, 2014). If this does not return any results, the chain is marked for manual curation.

Table 2 Algorithms used in the CATH Update protocol to automatically identify protein domains. The best result from each algorithm is also provided to the CATH curators for manual curation purposes

Acronym/Name	Description	Reference
ChopClose	Compares query chains against previously chopped chains in CATH to find the best match.	Greene <i>et al.</i> (2007)
CATHEDRAL	CATH's Existing Domain Recognition ALgorithm. Uses GRATH to find the best fold match and then SSAP to produce an accurate structural alignment of these fold matches. Publically available to use through: http://cathdb.info/search/by_structure	Redfern <i>et al.</i> (2007)
HMMs	Hidden Markov Models. Probabilistic models are created from multiple sequence alignments to capture information on the amino acid probabilities at each alignment position. Protein sequences are searched against these models to identify homology and to aid the inference of domain boundaries.	Karplus <i>et al.</i> (1998)
DETECTIVE	<i>Ab initio</i> algorithm that identifies protein domains by searching for the hydrophobic core and residues in contact with the core.	Swindells (1995)
DOMAK	DOmain MAKer. <i>Ab initio</i> algorithm that splits a given protein into arbitrary pieces (i.e. domains) until it has found the pieces that maximise intra-domain contacts and minimise inter-domain contacts.	Siddiqui and Barton (1995); Holm and Sander (1994)
PUU	<i>Ab initio</i> algorithm that studies the motion of residue clusters to analyse how different parts of the protein move in relation to each other. Each domain in a multi-domain protein has its own continuous, relative motion, which is used to identify domain boundaries.	Holm and Sander (1994)

Modified from Pearl, F.M.G., Sillitoe, I., Orengo, C.A., 2015. Protein Structure Classification. eLS, pp. 1–10.

Sequence-based methods for recognising domains in proteins

In CATH, all new query structures from the PDB are also scanned against sequence profiles or hidden Markov models (HMMs) built with SAM (Sequence Alignment and Modelling) (Karplus *et al.*, 1998) from representative domain structures in CATH. The best-matching HMM to the query is that with the lowest E-value, which is significant if it is less than 0.001.

BLAST (Altschul *et al.*, 1990) and HHsearch (Söding, 2005) are used in the ECOD pipeline to search PDB chain queries against existing domain data in the ECOD database. The protein chain is first searched with BLAST against a library of full-length chains from ECOD that have had domains identified. The domains in the most significantly-matching chain (i.e., a match with a sequence similarity E-value $< 2 \times 10^{-03}$) are used to infer domain boundary positions onto the query chain. Next, the protein chain is search with BLAST against a library of ECOD domain sequences. Domains are assigned individually to the best match (i.e., sequence similarity E-value $< 2 \times 10^{-03}$ and hit coverage $> 80\%$). Finally, to detect more remotely related domains, HHblits (Söding, 2005) is used to generate a sequence profile of the query, which is then searched against the ECOD representative domain profiles using HHsearch (Cheng *et al.*, 2014).

Ab-initio methods for recognising domains in proteins

When there are no existing fold and domain matches to assist in defining domain boundaries, it is often useful to apply *ab initio* methods. Three *ab-initio* methods are used in CATH to identify protein domains: DETECTIVE (Swindells, 1995), DOMAK (Siddiqui and Barton, 1995), and PUU (Holm and Sander, 1994). Each algorithm uses a different method to identify domain boundaries, but they all use information on contacts between residues in the structure. They are all able to predict boundaries for both contiguous and discontinuous domains. DETECTIVE searches for the hydrophobic core of each domain. DOMAK randomly divides up a protein to maximise intra-domain contacts and minimise inter-domain contacts. PUU analyses the motion of predicted domains in relation to each other.

The majority of the chains that come into CATH are automatically chopped into domains using the ChopClose, CATHEDRAL and HMM protocols. However, due to the strict criteria imposed for quality control purposes, around 5% of chains need manual curation which is guided by the *ab-initio* methods.

Table 2 summarises the structure, sequence and *ab-initio* methods used to assign domain boundaries for domains classified in CATH.

Hierarchical Structural Classification

The three major domain structure classifications: SCOP, CATH and ECOD, use similar hierarchical classifications, with a few differences between them (Fig. 3). In the CATH classification the main levels from top to bottom are: class, architecture, topology or fold, homologous superfamily, and family. In the SCOP classification the different levels are: class, topology or fold, superfamily, and family. In the ECOD classification, which specialises in detecting remote homologies, there are five main levels: architecture, possible homology, homology, topology or fold, and family. These different levels will be discussed below.

Class

The protein class (C-level) considers the proportion of residues within a structure that fall within α -helix and/or β -strand secondary structure elements. There are three major classes: mainly α , mainly β , and a mixture of α and β . SCOP separates the latter into $\alpha + \beta$

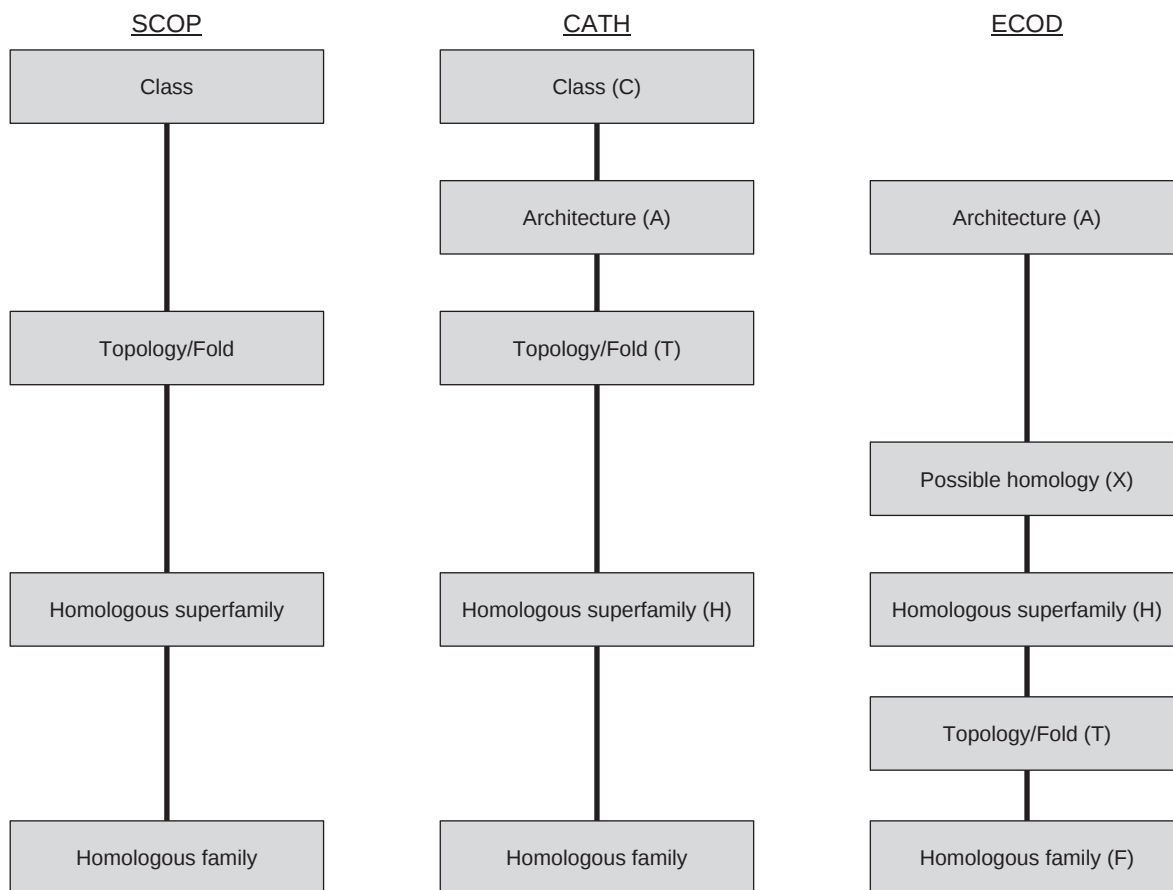


Fig. 3 Comparison of the structural classification hierarchies used by SCOP, CATH and ECOD. The single letter used to represent a particular level has been included. While there are many similarities between the resources, there are clear differences such as the addition of the possible homology (X) level in ECOD, the lack of an architecture level in SCOP, and no class level in ECOD.

and α/β , giving this resource four main classes. CATH considers these as one class after finding considerable overlap between them since their original description in [Levitt and Chothia \(1976\)](#). CATH has a fourth class that represents domains with few secondary structures. Class is not featured in ECOD.

Architecture

Protein architecture (the A-level) describes the 3D spatial arrangement of secondary structure elements, independent of any connectivity information. Currently in CATH there are 41 different architectures: 5 mainly α , 21 mainly β , 14 α and β , and 1 in few secondary structures. Each architecture can represent 3D arrangements of secondary structures with diverse connectivities, which is represented by the topology level. There are more β -based architectures as constraints on the packing of β -sheets lead to very distinct and easily recognisable architectures such as β -barrels, β -prisms, and β -propellers.

Topology or fold similarity

The topology of a protein domain is given by its 3D spatial arrangement together with connectivity information, i.e., two domains having the same fold should have the same sequence of secondary structure elements packed together in similar orientations in 3D. In CATH, protein domains that have significant structural similarity to each other (i.e., a SSAP score of ≥ 70), but no sequence or functional similarity, are grouped into the same T-level. ECOD sub-classifies its A-levels into possible homology (X-levels), to represent groups of domains that have incomplete evidence for being homologous ([Cheng et al., 2014](#)). The T-level in ECOD comes after the homologous superfamily, H-level, rather than the A-level, so homologous domains are subdivided into topology-based groups. This is because in some superfamilies, relatives can diverge structurally to such an extent that they could be considered to have different folds. In CATH this structural variation is captured by grouping homologues into different structurally-similar clusters (SC5s), where relatives superpose within 5 Å RMSD or less.

Homologous superfamily

Domains are grouped at the homologous superfamily level (H-level) where there is clear evidence of homology, i.e., the domains have evolved from a common ancestor. There can be considerable divergence across some superfamilies in terms of structure,

sequence and function. Fig. 4 shows that for representative relatives (i.e., representatives of relatives clustered at 95% sequence identity) within the same superfamily there can be a wide range of sequence identity scores.

For these very remote homologues, structure and function information can be used to detect an evolutionary relationship. Protein structure is generally much more conserved than sequence, making it easier to use in homology detection. As with sequence, there can also be considerable variation of structure across some superfamilies (Fig. 5).

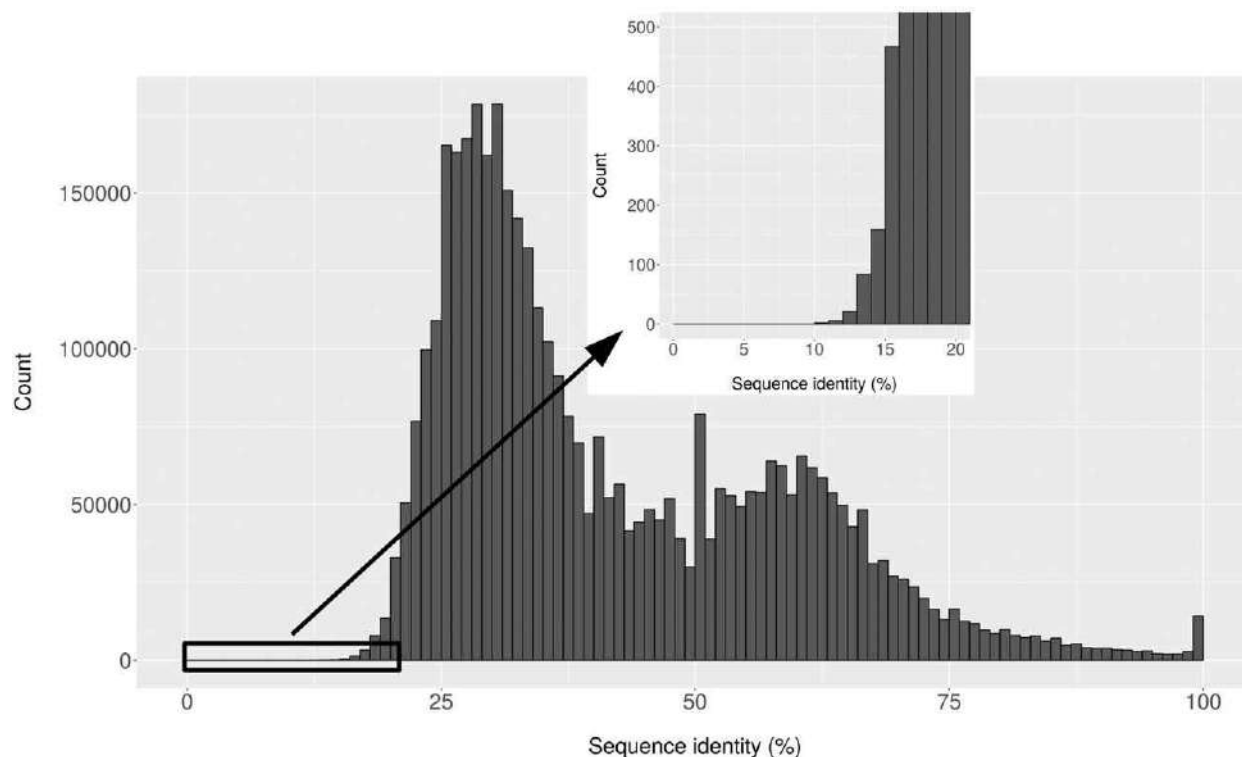


Fig. 4 The distribution of sequence identity scores between pairs of representative homologous domains (non-redundant at 95% sequence identity) in CATH. All-against-all comparisons were performed using BLASTp with the default maximum E-value (i.e., 10) and default maximum number of target sequences (i.e., 500). If we relax both of these thresholds then we can see results from comparing more relatives which have even lower sequence identities, however there are too many false positives (data not shown).

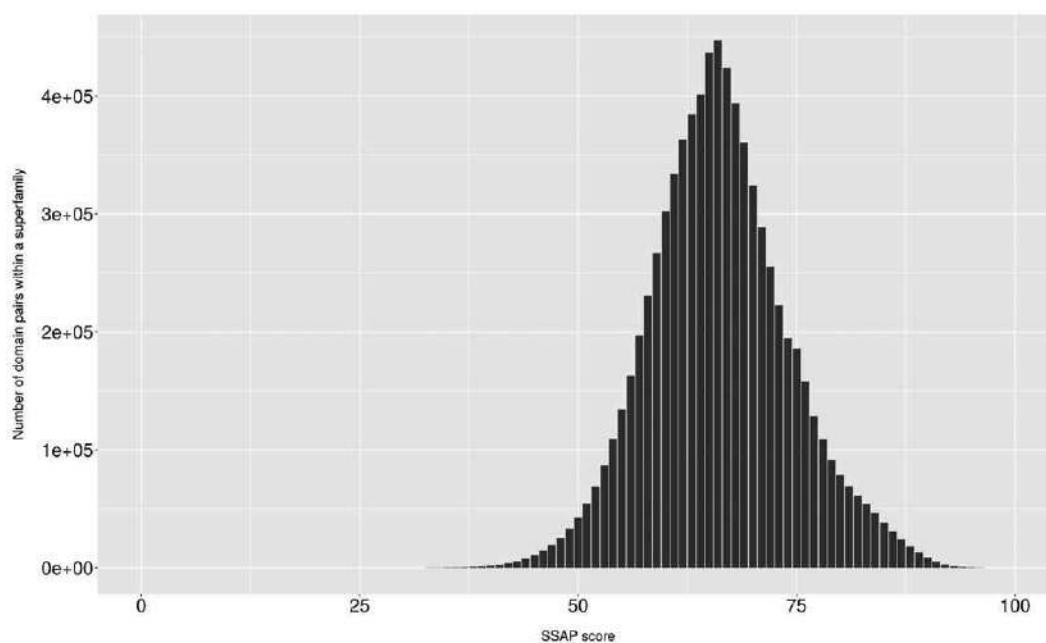


Fig. 5 The distribution of structural similarity scores (using SSAP) between pairs of homologous domains in CATH.

Homologous family

Protein families consist of close relatives which generally have at least 30% pairwise sequence identity to each other, or evidence of close functional similarity, or taxonomic similarity. SCOP places domains in the same family if they have at least 30% sequence identity, or if they have some level of sequence identity and a similar function (Hubbard *et al.*, 1997). These families generally consist of taxonomically close relatives. CATH recognises functionally-related families of homologues through similarities in specificity-determining residues (see FunFHMmer in Section “Capturing Functional Diversity by Sub-Classification of Super-families”). ECOD domains are scanned against Pfam family HMMs to define family memberships and those domains that cannot be assigned to a Pfam family are grouped together by HHsearch (Cheng *et al.*, 2014).

Nearest-Neighbour Clustering

Nearest-neighbour clustering is an alternative to hierarchical classification. A structural comparison program such as Dali or VAST+ is used to compare one or more structures against each other or the whole PDB archive, for example. These resources provide lists of the closest structural neighbours, where the significance of the matches is determined by taking the distribution of scores and calculating a Z-score for each match. In Dali (Holm and Sander, 1995; Holm and Laakso, 2016), the structures from the Z-score-ranked list can then be clustered to visualise the structural relationships as a dendrogram. These resources simply provide information on structural similarity and do not take into account any other information (e.g., similarity in functional features or rare sequence motifs). Hence they do not distinguish homologous and analogous relationships.

The resources providing nearest-neighbour clustering and hierarchical classification are listed in Table 1.

Integration of the SCOP and CATH Resources

While SCOP and CATH are both hierarchical classifications, there are differences in their methodologies. For example, while both resources use both manual and automated methods to identify domains and assign them to superfamilies, SCOP relies more on manual curation. There are also differences in the philosophies behind what defines a structural domain: SCOP recognises domains when they are found to occur across different multi-domain contexts, whereas CATH also uses physical information such as the domain size, globularity, and compactness (Hadley and Jones, 1999).

The Genome3D collaborative project (Lewis *et al.*, 2013, 2015) was developed to integrate the important information in these two resources and use it to annotate genomic sequence data. Well-known structure prediction resources from six structural bioinformatics groups across the UK were incorporated: DomSerf (Buchan *et al.*, 2010), FUGUE (Shi *et al.*, 2001), Gene3D (Yeats *et al.*, 2011), pDomTHREADER (Lobley *et al.*, 2009), Phyre2 (Kelley and Sternberg, 2009), SUPERFAMILY (Gough and Chothia, 2002), and VIVACE. UniProt sequence data from ten genomes were annotated by these resources, which used SCOP- and CATH-based structural data to predict structural domains within the sequences and build structural models.

Consensus superfamilies were identified between SCOP and CATH to also facilitate mapping between the structure predictions. This was done by comparing pairs of SCOP and CATH domains, and calculating the residue overlap between them. SCOP and CATH superfamilies sharing a high proportion of significantly overlapping domains were considered equivalent. The level of equivalence was graded into Bronze, Silver and Gold Standard categories, with gold being the best. A pair of superfamilies (SCOP-CATH) meeting the Bronze Standard are “more similar to each other than to any other superfamily” (Lewis *et al.*, 2013). A Silver Standard pair meets the Bronze Standard and, in addition, at least 80% of domains from each superfamily map between them with an average residue overlap of at least 80%. A Gold Standard pair, in addition to the other two Standards, has domains that map between both superfamilies with a minimum of 80% residue overlap. Mapping SCOP v1.75 to CATH v4.0 domains: 938 Gold Standard, 213 Silver Standard and 388 Bronze Standard superfamily consensus pairs were found. The Gold Standard superfamily pairs are currently being integrated into the InterPro resource (Finn *et al.*, 2017), where they have been added in as homologous superfamily entries. This work has greatly increased the coverage of both Gene3D and SUPERFAMILY domain and superfamily entries in InterPro.

Current Status of the Protein Structural Universe

As previously mentioned, as of November 2017, there are almost 135,000 entries in the PDB. Table 3 describes the statistics for SCOP, CATH and ECOD. The SCOP resource was last updated in 2009, which accounts for the large difference in the number of PDBs processed and domains identified between the three resources.

Of the 110,800 domains in SCOP v1.75, 15% of domains are mainly α -helical in structure, 24% are mainly β -structure, and 49% have mixed α/β or $\alpha + \beta$ structural content. The CATH classification has processed 90% of PDBs, and contains almost 445,000 structural domains. Currently, 22% of the domains in CATH superfamilies are mainly α -helical in structure, 25% are mainly β structure, and over half (52%) have mixed $\alpha\beta$ secondary structure content. The ECOD classification has processed slightly more PDBs than CATH (~3500) and has detected more remote homologies, leading to a smaller number of superfamilies. It is expected that the domains in ECOD follow a similar distribution, but as the ECOD hierarchy starts at the architecture level rather than class, it was not possible to separate the domains by class information.

All three of these resources provide free, publically available data. CATH and ECOD provide frequent updates, with ECOD publishing an update each week and CATH publishing an update every day. In addition to offering the daily-updated CATH

Table 3 Statistics for the SCOP, CATH and ECOD resources

	<i>SCOP v1.75</i>	<i>CATH v4.2</i>	<i>ECOD v199</i>
PDBs processed	38,221	131,091	134,556
Domains identified	110,800	434,857	577,454
Number of folds	1195	1391	3738 ^a
Number of superfamilies	1962	6119	3537
Number of families	3902	68,069	13,426

^aAll SCOP and ECOD data were obtained from their websites, apart from the number of folds, which was calculated by finding the number of unique X.H.T strings in their download file.

structural domain and superfamily assignment data, there are also CATH-Plus releases with a wealth of extra data. The latest CATH-Plus (version 4.2) release was recently published in October 2017. This release comprises 434,857 protein structural domains, grouped into 6119 homologous superfamilies. Other data included in CATH-Plus includes: functional family data, function annotations from Gene Ontology (GO) and Enzyme Commission (EC) data; enzyme function evolution information with FunTree; species diversity information from taxonomic information, structural diversity information through structurally similar groups (SSGs) and superposition images; domain organisation information with ArchSchema (see Fig. 6). Additional sequence domain data is also provided by Gene3D (see Section “Structural Families in the Genomic Era”), which contributes ~95 million sequence domains to CATH. Briefly, Gene3D uses the curated structural information in CATH to predict domain boundaries for UniProt sequences, which do not have a known 3D structure.

There is an uneven distribution of domains across the protein fold universe. For example, of the 1391 folds in CATH, the top five most populated folds account for 30% for all structural domain sequences in CATH. The top five most populated folds, including the: Rossmann fold, TIM barrel, and Immunoglobulin fold, are shown in Fig. 7. This distribution bias is also clearly seen in the homologous superfamilies as the most populated 100 CATH superfamilies represent almost half (49%) of all domain sequences.

Structural Families in the Genomic Era

Structural databases have increased massively over recent years. However, the sequence databases continue to grow even faster, with ~100 million sequences now in UniProt-KB. These numbers will continue to grow rapidly over the coming years. New sequence read technologies that provide longer reads will make sequencing animal genomes more easy and affordable, and will contribute to an explosion in genome sequences over the coming years. Despite this, the various resources for predicting domain families from sequences (see Table 4) have continued to cope with the current increase in datasets (Lewis *et al.*, 2017; Finn *et al.*, 2017).

Many of the family annotation resources such as Pfam and Gene3D use sequence profiles in the form of HMMs, built from structural representatives of a family, to recognise relatives in genome sequences. The sensitivity of HMMs comes from their ability to capture information of conserved residue positions across a given family, and positions where residue insertions and deletions are more tolerated.

One reason for this ability to scale arises from improvements in HMM-based software with large speedups produced recently by the HMMER package (Eddy, 2011). The Gene3D v16 resource assigns domains for over 52 million protein sequences, producing over 95 million domain assignments (Lewis *et al.*, 2017). For each CATH S95 sequence, a HMM is built using 1–2 rounds of Jackhmmer (for details see Lewis *et al.*, 2017). These HMMs are then scanned against the protein sequences using the hmmsearch program as part of the HMMER package. Sometimes different HMM models overlap the same region and it is necessary to resolve a final set of non-overlapping domain annotations. Gene3D makes use of the cath-resolve-hits (CRH) software (available from the cath-tools resource pages (see Relevant Websites Section)), which uses dynamic programming to efficiently select the optimal set of non-overlapping domains, using the bit score of each hit as a measure of the hits overall quality. Importantly CRH is both extremely fast, memory efficient and can deal with discontinuous domain assignments whilst still finding an optimal solution.

Evolution of Protein Structure, Sequence and Function

Evolution of Protein Folds

As more and more structural data and functional annotations accumulated in protein structure databases, such as CATH and SCOP, it appeared likely that only a limited number of folding arrangements exist in nature. For example, there now are 131,091 protein structures in the CATH database (version 4.2) that have been classified into 434,857 structural domains, which is a nearly 50% increase in the number of domain structures over the last five years. However, the number of folds identified has increased very slowly with still only 1391 different folds identified (Dawson *et al.*, 2017).

Protein fold similarity can often provide important clues regarding the functional role of a protein. However, studies on different folds have shown that protein domains that share the same fold can show functional divergence (Martin *et al.*, 1998). As mentioned already above, some folds, also known as ‘superfolds’, are highly populated and occur more frequently than others.

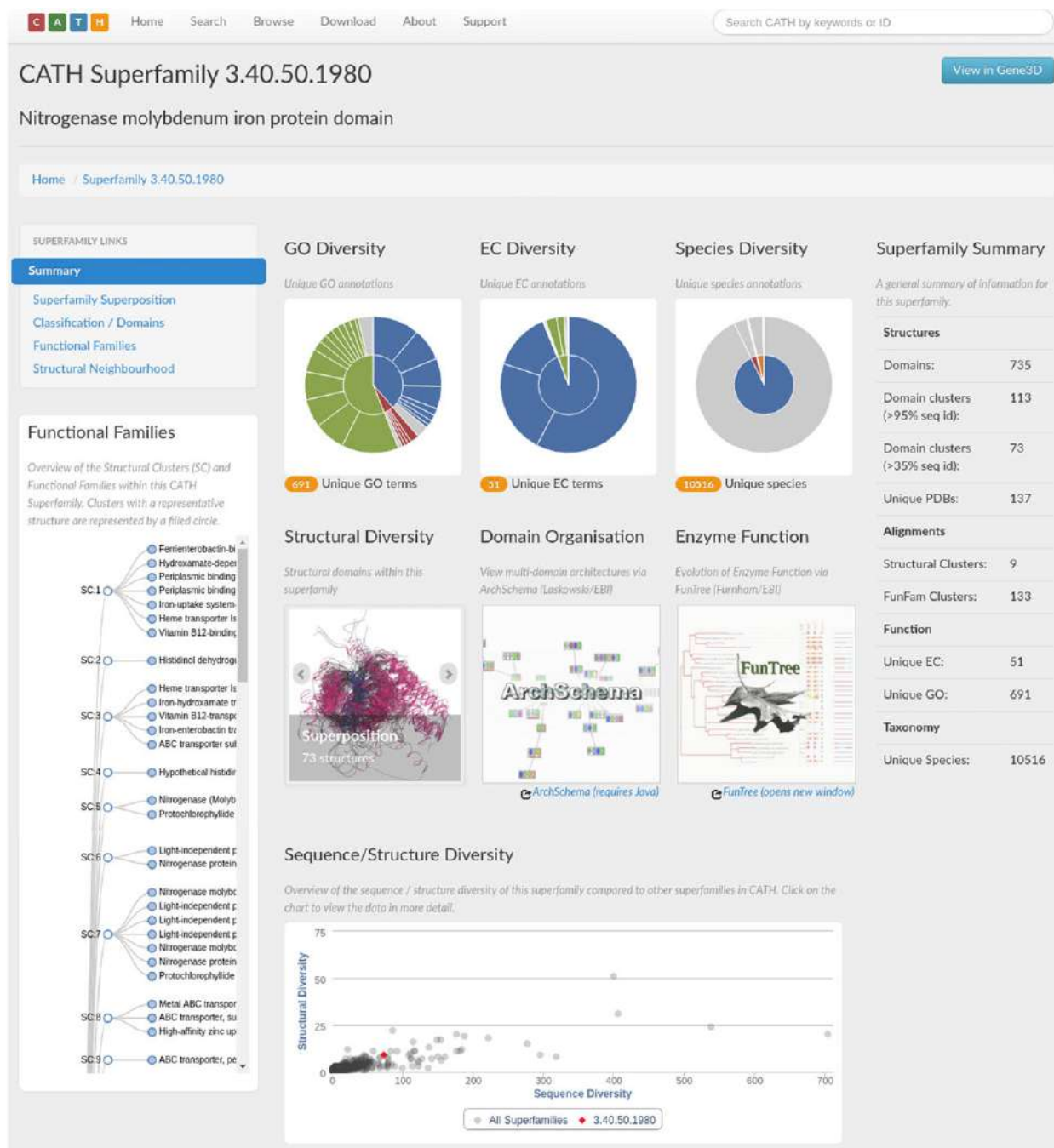


Fig. 6 Example of a superfamily summary web page, which is available for each superfamily in CATH. These pages contain information on: function annotation (GO and EC terms), species diversity, structural diversity, sequence diversity, and functional families.

These generally adopt Rossmann and TIM barrel folds or other folds which possess very regular architectures such as layers of beta-sheets and/or alpha-helices (Orengo *et al.*, 1994). It has been suggested that the regularity of these folding arrangements may provide a stable core to support diverse sequences. Hence, it is not surprising that they comprise multiple superfamilies. For example, the Rossmann fold comprises 130 superfamilies and the TIM barrel constitutes 30 superfamilies in the CATH database.

Exploiting CATH Superfamilies to Examine the Evolution of Protein Functions

The CATH-Gene3D resource (Dawson *et al.*, 2017; Lewis *et al.*, 2017), which brings together evolutionarily-related structural and sequence domains into superfamilies along with their functional annotations, provides an ideal platform to systematically analyse and explore how function is modulated by sequence and structural changes.

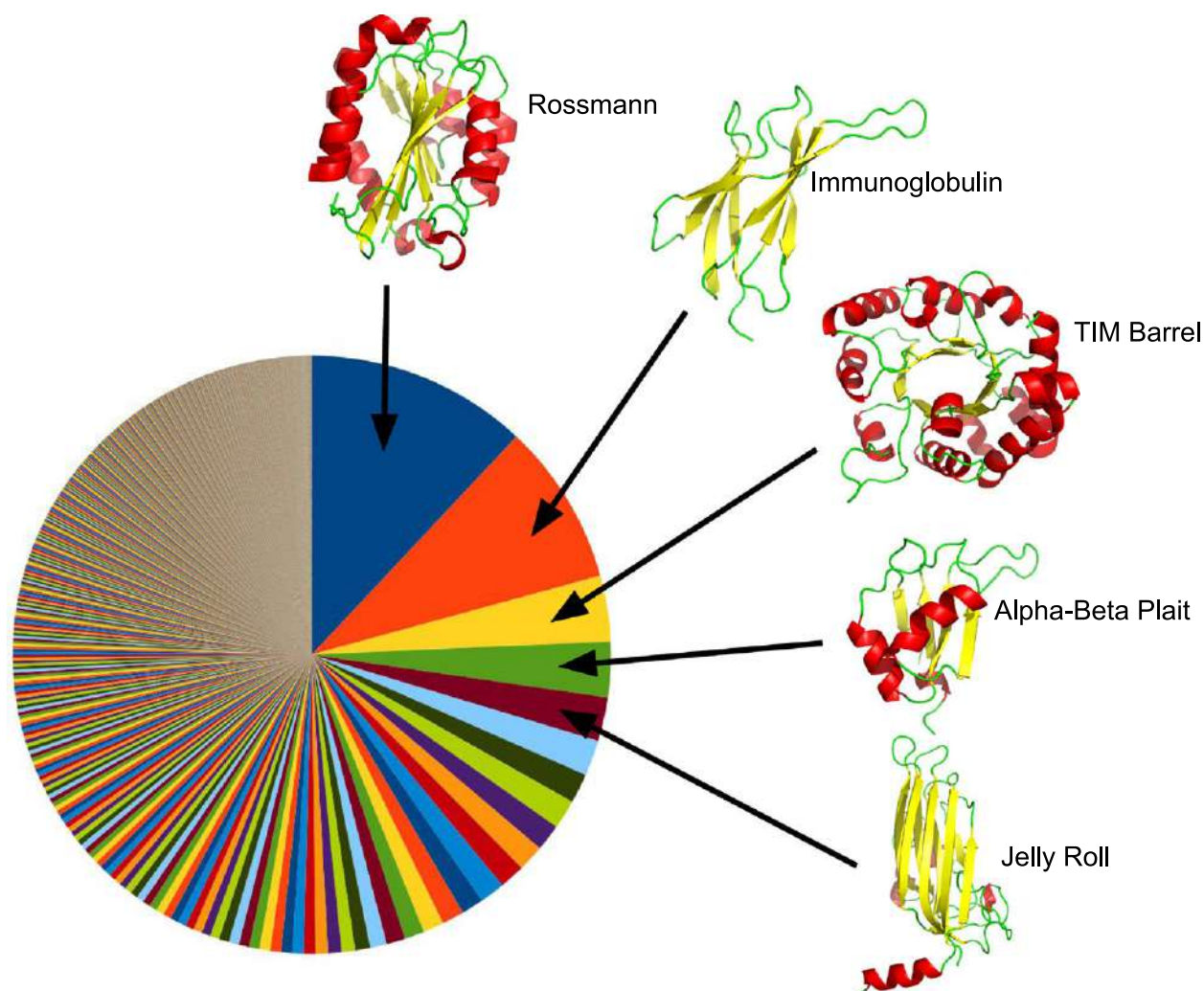


Fig. 7 Fold counts based on the number of CATH structural domain sequences. The top five folds account for 30% of all structural domain sequences in CATH. A structural representative from each of the top five folds is illustrated, coloured according to their secondary structure content.

Table 4 Example resources with genome annotations of structural assignments along with details of some of their unique features

Resource	Description	Useful Features
PFAM http://pfam.xfam.org/	Comprehensive prediction of protein families using HMMs.	User friendly tools to scan datasets. Extensive annotations of the Pfam families that helps to functionally interpret results. Combines families into larger Clans based on shared evolutionary origin.
InterPro https://www.ebi.ac.uk/interpro/	Integrates multiple resources assigning protein families domains and functional sites.	Scope and annotations has proven it to be a cornerstone of function prediction. Annotates entries with extensive text, GO terms etc.
Gene3D http://gene3d.biochem.ucl.ac.uk/	Predicts CATH structural domains.	Powered by CATH and so is largely up to date relative to the PDB. Functional Family level provides state of the art function and structure prediction.
SUPERFAMILY http://supfam.org/SUPERFAMILY	Predicts SCOP structural domains.	Established resource that is widely used in research projects e.g., provides tools for comparative genomics and genome annotations.
Genome3D http://genome3d.eu/	Provides consensus structural annotations and 3D models for model organisms.	Integrates SCOP, CATH, SUPERFAMILY, Gene3D, FUGUE, THREADER, PHYRE.

Modified from Pearl, F.M.G., Sillitoe, I., Orengo, C.A., 2015. Protein Structure Classification. eLS, pp. 1–10.

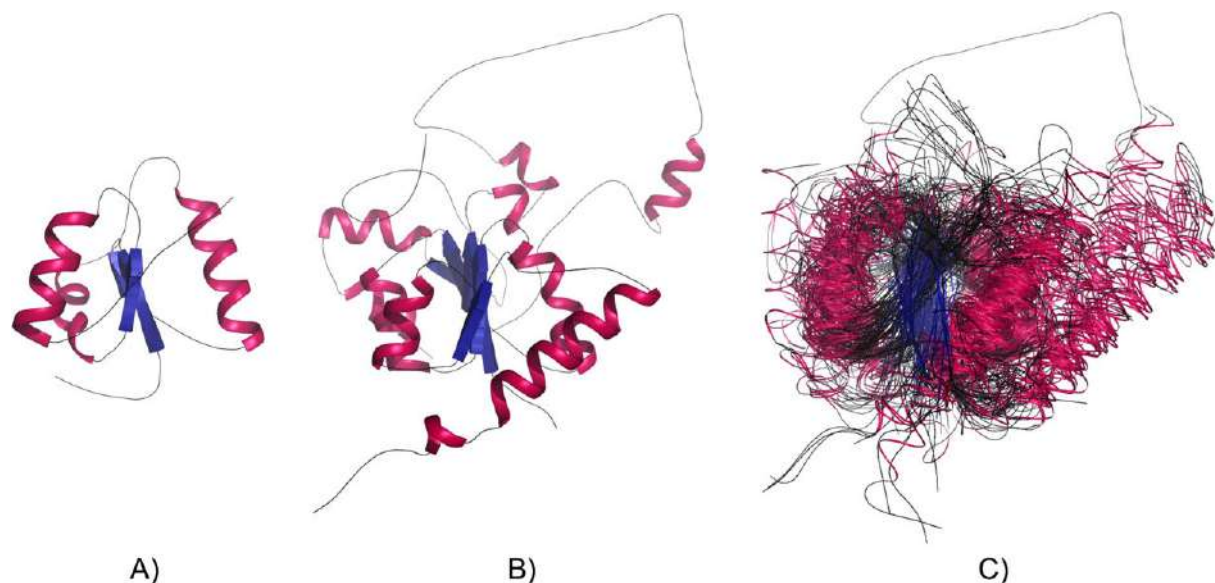


Fig. 8 Structural diversity in the “Nitrogenase molybdenum iron protein domain” superfamily (CATH ID: 3.40.50.1980). This superfamily is highly structurally diverse with nine structurally similar clusters. The figure shows structures of: (A) the smallest, (B) largest domain and (C) superposition of representative non-redundant superfamily members to highlight the conserved structural core.

Some CATH superfamilies are very highly populated and universal to all kingdoms of life (Reid *et al.*, 2010). As previously mentioned, the 100 most highly populated superfamilies in CATH account for half of all known protein domains. The domain relatives in these superfamilies can incorporate large amounts of structural and functional diversities even though they share a conserved structural core. In some superfamilies relatives can vary in size by as much as 5-fold (Fig. 8(A) and (B)). This has been described as the ‘Russian doll’ effect (Swindells *et al.*, 1998). However, most relatives across a superfamily share the same common structural core comprising at least 50% of their structure (Fig. 8(C)).

Mechanisms of functional divergence

Various studies (Das *et al.*, 2015; Todd *et al.*, 2001; Reeves *et al.*, 2006; Brown and Babbitt, 2014; Galperin and Koonin, 2012) on the evolution of function in diverse superfamilies have illustrated that proteins can evolve new functions by one or a combination of different molecular mechanisms (Fig. 9). Some of these mechanisms include use of different sets of residues in binding or active site, addition of secondary structure embellishments to the core protein structure which alters the geometry of the active site or an interface on the protein or due to domain-shuffling in multi-domain proteins which can alter the context of the domain. Examples of some these mechanisms of functional divergence have been briefly discussed below.

Molecular tinkering

Relatively small changes in a binding or active site can result in changes in shape, physicochemical or other characteristics of the site which can alter the function of a protein significantly. Although, the active site may be highly conserved in some diverse enzyme superfamilies, for example, functionally diverse enzymes of the α/β hydrolase superfamily utilize the same catalytic triad (Todd *et al.*, 2001), a recent study indicated that relatives of a large number of diverse enzyme superfamilies show changes in catalytic machinery (Furnham *et al.*, 2016). One such superfamily is the Aldolase Class I superfamily, the members of which can carry out reactions with 31 different enzyme chemistry by using different sets of catalytic residues.

Structural mechanisms

Superfamily relatives can also vary in size to a great extent. For more than 150 CATH superfamilies, at least a 2-fold variation in the size has been observed between the most diverse domains and up to 5-fold in some superfamilies (Reeves *et al.*, 2006). The structural diversity among the domains can range from small changes within the active site to extensive embellishments which can alter their function, as seen in the HUP (High-signature proteins, UspA, and PP-ATPase) superfamily (Dessailly *et al.*, 2013).

Multi-domain context

Changes in the multi-domain architecture (MDA) of a protein can significantly alter its context, thereby modifying its function. For example, enzymes of the Thiamine pyrophosphate (TPP)-dependent superfamily, catalyse a large number of different reactions using TPP as the co-factor, by changing the domain partnership which alters the size and physicochemical properties of the active site pocket (Vogel and Pleiss, 2014).

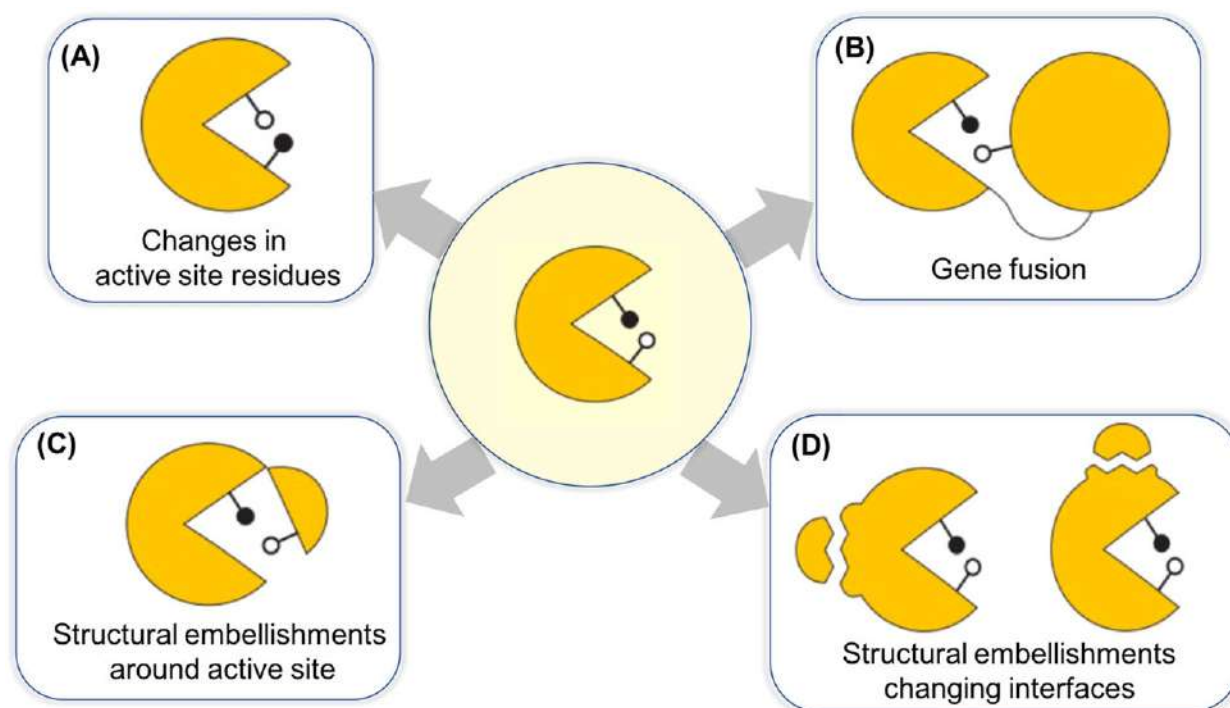


Fig. 9 Functional divergence in protein domains can arise due to one or combination of some of the following molecular mechanisms: (A) changes in active site residue, (B) gene fusion, (C) structural embellishments around active site and (D) structural embellishments changing interfaces. Modified from Das, S., Dawson, N.L., Orengo, C.A., 2015. Diversity in protein domain superfamilies. *Current Opinion in Genetics and Development* 35, 40–49.

In the majority of the large functionally diverse superfamilies, functional diversity generally results from a combination of different molecular mechanisms.

Capturing Functional Diversity by Sub-Classification of Superfamilies

One way to capture and understand this functional diversity among protein homologues is to further classify the superfamilies into functional groups. This functional classification also helps in inheriting annotations between homologous proteins that are most likely to perform the same function. Additionally, it allows identification of residues that are highly conserved among family members and which are most likely to have a functional role.

The CATH superfamilies have been functionally classified into Functional Families (FunFams) that group together relatives likely to have highly similar functions. The functional classification is derived from sequence similarities of superfamily relatives by optimal partitioning of a hierarchical sequence clustering tree for the superfamily which is generated by an agglomerative clustering algorithm (Lee *et al.*, 2010). This optimal partition is done by the FunFHMmer algorithm (Das *et al.*, 2016) which is based on maximising coherence of conserved residues within clusters and separation of differentially conserved residue positions (also known as functional determinants or FDs) between different clusters. The resulting clusters from the partition make up the FunFams for the superfamily. The FunFams have been shown to be functionally coherent based on different benchmarks using known functional annotations from the EC, GO, and annotations from the gold-standard SFLD superfamilies (Das *et al.*, 2016).

Hidden Markov Models (HMMs) are generated for all FunFams in CATH superfamilies and are used for predicting functions for uncharacterized sequences. When an uncharacterised query sequence is scanned against the library of CATH FunFam HMMs, it provides FunFam assignments. This provides both structural and functional insights for identified sequence domains. Functions for the query sequence domains are predicted by inheriting the known functional annotations of the sequence relatives of the assigned FunFam. This function prediction pipeline has been consistently ranked among the top 10 performing methods by the Critical Assessment of Function Annotation (CAFA) challenges (Radivojac *et al.*, 2013; Jiang *et al.*, 2016), an independent large-scale assessment of function prediction methods using different benchmarks and evaluation metrics. A web server (See Relevant Websites Section), Fig. 10(A) has been set up to allow users to submit query sequences in FASTA format for assignment to CATH FunFams where possible for the constituent domains of the query sequence (Das *et al.*, 2015). For each of the constituent FunFams of the query sequence, the server (Fig. 10(B)) reports functional annotations of the FunFam domain relatives and highly conserved residues for alignments with sufficient information content.

(A) Search CATH

>tr|A0A0Q0Y989|A0A0Q0Y989_9BACI
 MDPFHDTMAEVDLDAIYONVANLSRLPDOTHIMAVKANAYGHDQVQVARTALEAGASRLAVAFLEALALEKGI
 EAPILVLGASRFADAAALAAQRIALTVPFRSDWLEASALYSGFFPIHFHLKMDTGMRGLGVKDEETKRIVALIERHP
 HFVLEGVYTHFATADEVNTDYSQYTRFLMLKLEKLPSEFPLVHCANSAASLRFPORTFNKVRFGIANYGLAPSPGIX
 PLLPYLKEAFSLHRLVHVKKLQPGKVSYGATTAQTEEMIGTIFIGYADGHLRLQHFLVLDGQKAPIVGRICH

Search Clear

Search by Text or ID Search by Sequence Search by Structure

Progress Matching Domains Matching FunFams

API Help

Paste your protein sequence into the text box above (or use an example) then click 'Search'.

- The progress bar below will let you know when your results are available.
- Click [Help](#) to find out more information on this search.
- Click [API](#) to find out how to use this service in your programs.

(B) Search CATH

Search by Text or ID Search by Sequence Search by Structure

Progress 157 Matching Domains 25 Matching FunFams

API Help

Match	3D	EC	Matching Regions	Evalue
Putative alanine racemase 2 (3.20.20.10/FF/9715)	3D	EC		1.0e-76
Putative alanine racemase 2 (2.40.37.10/FF/6607)	3D	EC		1.8e-58
Alanine racemase 1, PLP-binding, biosynthetic (3.20.20.10/FF/9728)		EC		9.5e-40
Alanine racemase (2.40.37.10/FF/3164)		EC		4.3e-32
Alanine racemase (2.40.37.10/FF/849)		EC		1.1e-24
Amino-acid racemase (2.40.37.10/FF/2130)		EC		1.1e-23
Alanine racemase (2.40.37.10/FF/5098)		EC		6.1e-20
Alanine racemase (2.40.37.10/FF/6334)		EC		6.6e-19
Alanine racemase 1, PLP-binding, biosynthetic (2.40.37.10/FF/2915)		EC		6.8e-18

Fig. 10 CATH sequence search: (A) The server (http://www.cathdb.info/search/by_sequence) takes a query protein sequence (in FASTA format) as input and scans it against the library of CATH-FunFam HMMs. (B) The CATH FunFam matches for the constituent domains of the query sequence are reported along with their corresponding E-values.

The functional classification also provides a platform to detect subtle changes in conservation patterns that can suggest shifts in binding specificities or catalytic machineries of domain relatives in a superfamily. This kind of information would be helpful to guide experiments for target selection which would help to characterise unusual domain relatives of all protein superfamilies.

Conclusions

Protein structural classifications categorise protein structures deposited in the PDB according to their 3D structural characteristics and evolutionary relationships. There are three major classifications publically available to use: SCOP, CATH, and ECOD. A range of algorithms have been developed to recognise domain boundaries within protein structures using sequence-, structure-, and ab-initio-based methods. Algorithms have also been developed to detect homologous relationships, which are used to assign these domains into homologous superfamily groups and thereby classifying them within a classification hierarchy.

Nearest-neighbour clustering methods are an alternative way of searching for structural homologues. Structural comparison methods (e.g., Dali and VAST +) are used to compare one or more query structures against each other or the whole PDB archive,

for example. Only information on structural similarity is provided, offering a fast way to generate lists of potential structural relatives and analogues.

SCOP, CATH and ECOD have many similarities in their protein structural classification hierarchies, which are discussed in Section “Hierarchical Structural Classification”. Consensus SCOP and CATH superfamily data have been identified through the Genome3D project and are currently being integrated into the InterPro resource to increase the coverage of data for the SUPERFAMILY and Gene3D resources (see Section “Integration of the SCOP and CATH Resources”).

A look into the current status of the protein structural universe highlights the large amounts of structural data that has been analysed and made publically available. As seen over recent years, there is an uneven distribution of domains assigned to protein folds and superfamilies. For example, the 100 most-populated CATH superfamilies have consistently accounted for around half of all structural domain sequences.

Taking a look at the vast amounts of genomic sequence data available, Section “Structural Families in the Genomic Era” describes how structural family data is used to annotate these data, which do not have any known structures. HMM libraries are built from structural domain data in CATH, which are then used by the sister resource Gene3D to scan genome sequences against to find significant structural matches for domain and homology detection. These scanning methods and libraries provide a very powerful way to annotate genomic data.

CATH superfamilies and functional families can be used to examine the evolution of protein sequence, structure and function. The mechanisms behind these divergences are discussed in Section “Evolution of Protein Structure, Sequence and Function”. Superfamily superpositions illustrate any structural diversity within a superfamily, yet clearly show that even diverse superfamily have a highly conserved structural core. Functional family data are an important part in exploring functional diversity within a superfamily, determining functional relatives and in annotating protein function for unknown sequences.

Acknowledgements

N.L.D. acknowledges funding from the Wellcome Trust (Award number: 104960/Z/14/Z). J.G.L. acknowledges funding from the BBSRC [BB/L002817/1]. S.D. acknowledges funding from the BBSRC.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Biomolecular Structures: Prediction, Identification and Analyses. Computational Tools for Structural Analysis of Proteins. Drug Repurposing and Multi-Target Therapies. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Protein Structural Bioinformatics: An Overview. Protein Structure Databases. Protein Three-Dimensional Structure Prediction. Secondary Structure Prediction. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Study of The Variability of The Native Protein Structure. Supervised Learning: Classification. Supervised Learning: Classification. The Evolution of Protein Family Databases. Transmembrane Domain Prediction

References

- Altschul, S.F., *et al.*, 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3), 403–410.
- Andreeva, A., *et al.*, 2014. SCOP2 prototype: A new approach to protein structure mining. *Nucleic Acids Research* 42 (Database issue), D310–D314.
- Brown, S.D., Babbitt, P.C., 2014. New insights about enzyme evolution from large scale studies of sequence and structure relationships. *The Journal of Biological Chemistry* 289 (44), 30221–30228.
- Buchan, D.W.A., *et al.*, 2010. Protein annotation and modelling servers at University College London. *Nucleic Acids Research* 38 (Web Server issue), W563–W568.
- Cheng, H., *et al.*, 2014. ECOD: An evolutionary classification of protein domains. *PLOS Computational Biology* 10 (12), e1003926.
- Das, S., *et al.*, 2016. Functional classification of CATH superfamilies: A domain-based approach for protein function annotation. *Bioinformatics* 32 (18), 2889.
- Das, S., Dawson, N.L., Orengo, C.A., 2015. Diversity in protein domain superfamilies. *Current Opinion in Genetics & Development* 35, 40–49.
- Das, S., Sillitoe, I., *et al.*, 2015. CATH FunFHMmer web server: Protein functional annotations using functional family assignments. *Nucleic Acids Research* 43 (W1), W148–W153.
- Dawson, N.L., *et al.*, 2017. CATH: An expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research* 45 (D1), D289–D295.
- Dessailly, B.H., *et al.*, 2013. Functional site plasticity in domain superfamilies. *Biochimica Et Biophysica Acta* 1834 (5), 874–889.
- Eddy, S.R., 2011. Accelerated profile HMM searches. *PLOS Computational Biology* 7 (10), e1002195.
- Finn, R.D., *et al.*, 2017. InterPro in 2017-beyond protein family and domain annotations. *Nucleic Acids Research* 45 (D1), D190–D199.
- Furnham, N., *et al.*, 2016. Large-scale analysis exploring evolution of catalytic machineries and mechanisms in enzyme superfamilies. *Journal of Molecular Biology* 428 (2 Pt A), 253–267.
- Galperin, M.Y., Koonin, E.V., 2012. Divergence and convergence in enzyme evolution. *The Journal of Biological Chemistry* 287 (1), 21–28.
- Gerstein, M., 1998. How representative are the known structures of the proteins in a complete genome? A comprehensive structural census. *Folding and Design* 3 (6), 497–512.
- Gibrat, J.-F., Madej, T., Bryant, S.H., 1996. Surprising similarities in structure comparison. *Current Opinion in Structural Biology* 6 (3), 377–385.
- Gough, J., Chothia, C., 2002. SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. *Nucleic Acids Research* 30 (1), 268–272.
- Greene, L.H., *et al.*, 2007. The CATH domain structure database: New protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Research* 35 (Database issue), D291–D297.

- Hadley, C., Jones, D.T., 1999. A systematic comparison of protein structure classifications: Scop, CATH and FSSP. *Structure* 7 (9), 1099–1112.
- Holm, L., Laakso, L.M., 2016. Dali server update. *Nucleic Acids Research* 44 (W1), W351–W355.
- Holm, L., Park, J., 2000. DaliLite workbench for protein structure comparison. *Bioinformatics* 16 (6), 566–567.
- Holm, L., Sander, C., 1995. Dali: A network tool for protein structure comparison. *Trends in Biochemical Sciences* 20 (11), 478–480.
- Holm, L., Sander, C., 1994. Parser for protein folding units. *Proteins* 19 (3), 256–268.
- Hubbard, T.J.P., *et al.*, 1997. SCOP: A Structural Classification of Proteins database. *Nucleic Acids Research* 25 (1), 236–239.
- Jiang, Y., *et al.*, 2016. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology* 17 (1), 184.
- Karplus, K., Barrett, C., Hughey, R., 1998. Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 14 (10), 846–856.
- Kelley, L.A., Sternberg, M.J.E., 2009. Protein structure prediction on the Web: A case study using the Phyre server. *Nature Protocols* 4 (3), 363–371.
- Krissinel, E., Henrick, K., 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D, Biological Crystallography* 60 (Pt 12 Pt 1), 2256–2268.
- Lee, D.A., Rentzsch, R., Orengo, C., 2010. GeMMA: Functional subfamily classification within superfamilies of predicted protein structural domains. *Nucleic Acids Research* 38 (3), 720–737.
- Levitt, M., Chothia, C., 1976. Structural patterns in globular proteins. *Nature* 261 (5561), 552–558.
- Lewis, T., *et al.*, 2017. Gene3D: Extensive prediction of globular domains in proteins. *Nucleic Acids Research*. Available at: <https://doi.org/10.1093/nar/gkx1069>.
- Lewis, T.E., *et al.*, 2013. Genome3D: A UK collaborative project to annotate genomic sequences with predicted 3D structures based on SCOP and CATH domains. *Nucleic Acids Research* 41 (Database issue), D499–D507.
- Lewis, T.E., *et al.*, 2015. Genome3D: Exploiting structure to help users understand their sequences. *Nucleic Acids Research* 43 (Database issue), D382–D386.
- Lobley, A., Sadowski, M.I., Jones, D.T., 2009. pGenTHREADER and pDomTHREADER: New methods for improved protein fold recognition and superfamily discrimination. *Bioinformatics* 25 (14), 1761–1767.
- Madej, T., Gibrat, J.F., Bryant, S.H., 1995. Threading a database of protein cores. *Proteins* 23 (3), 356–369.
- Martin, A.C., *et al.*, 1998. Protein folds and functions. *Structure* 6 (7), 875–884.
- Mizuguchi, K., *et al.*, 1998. HOMSTRAD: A database of protein structure alignments for homologous families. *Protein Science: A Publication of the Protein Society* 7 (11), 2469–2471.
- Orengo, C.A., *et al.*, 1997. CATH – A hierarchic classification of protein domain structures. *Structure* 5 (8), 1093–1109.
- Orengo, C.A., Jones, D.T., Thornton, J.M., 1994. Protein superfamilies and domain superfolds. *Nature* 372 (6507), 631–634.
- Pearl, F.M.G., Sillitoe, I., Orengo, C.A., 2015. Protein Structure Classification. *eLS*. pp. 1–10.
- Radivojac, P., *et al.*, 2013. A large-scale evaluation of computational protein function prediction. *Nature Methods* 10, 221.
- Redfern, O.C., *et al.*, 2007. CATHEDRAL: A fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. *PLOS Computational Biology* 3 (11), e232.
- Reeves, G.A., *et al.*, 2006. Structural diversity of domain superfamilies in the CATH database. *Journal of Molecular Biology* 360 (3), 725–741.
- Reid, A.J., Ranea, J.A., Orengo, C.A., 2010. Comparative evolutionary analysis of protein complexes in *E. coli* and yeast. *BMC Genomics* 11, 79.
- Shi, J., Blundell, T.L., Mizuguchi, K., 2001. FUGUE: Sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of Molecular Biology* 310 (1), 243–257.
- Shindyalov, I.N., Bourne, P.E., 2001. A database and tools for 3-D protein structure comparison and alignment using the Combinatorial Extension (CE) algorithm. *Nucleic Acids Research* 29 (1), 228–229.
- Siddiqui, A.S., Barton, G.J., 1995. Continuous and discontinuous domains: An algorithm for the automatic generation of reliable protein domain definitions. *Protein Science: A Publication of the Protein Society* 4 (5), 872–884.
- Söding, J., 2005. Protein homology detection by HMM-HMM comparison. *Bioinformatics* 21 (7), 951–960.
- Swindells, M.B., 1995. A procedure for detecting structural domains in proteins. *Protein Science: A Publication of the Protein Society* 4 (1), 103–112.
- Swindells, M.B., *et al.*, 1998. Contemporary approaches to protein structure classification. *BioEssays: News and Reviews in Molecular, Cellular and Developmental Biology* 20 (11), 884–891.
- Taylor, W.R., Orengo, C.A., 1989. Protein structure alignment. *Journal of Molecular Biology* 208 (1), 1–22.
- Teichmann, S.A., Park, J., Chothia, C., 1998. Structural assignments to the *Mycoplasma genitalium* proteins show extensive gene duplications and domain rearrangements. *Proceedings of the National Academy of Sciences of the United States of America* 95 (25), 14658–14663.
- Todd, A.E., Orengo, C.A., Thornton, J.M., 2001. Evolution of function in protein superfamilies, from a structural perspective. *Journal of Molecular Biology* 307 (4), 1113–1143.
- Vogel, C., Pleiss, J., 2014. The modular structure of ThDP-dependent enzymes. *Proteins: Structure, Function, and Bioinformatics*. Available at: <http://onlinelibrary.wiley.com/doi/10.1002/prot.24615/full>.
- Yeats, C., *et al.*, 2011. The Gene3D Web Services: A platform for identifying, annotating and comparing structural domains in protein sequences. *Nucleic Acids Research* 39 (suppl), W546–W550.

Further Reading

- Branden, C., Tooze, J., 1999. *Introduction to Protein Structure*, second ed. New York: Garland Science.
- Dawson, N.L., Sillitoe, I., Lees, J.G., Lam, S.D., Orengo, C.A., 2017. CATH-Gene3D: Generation of the Resource and its use in Obtaining Structural and Functional Annotations for Protein Sequences. *Protein Bioinformatics: From Protein Modifications and Networks to Proteomics*. New York, NY: Humana Press, pp. 79–110.
- Gu, J., Bourne, P.E. (Eds.), 2009. *Structural Bioinformatics*. Hoboken, NJ: Wiley-Blackwell.
- Lesk, A.M., 2010. *Introduction to Protein Science: Architecture, Function and Genomics*, second ed. Oxford: Oxford University Press.
- Mount, S., 2004. *Bioinformatics: Sequence and Genome Analysis*, second ed. Cold Spring Harbor Press.
- Mukherjee, S., Seshadri, R., Varghese, N.J., *et al.*, 2017. 1003 reference genomes of bacterial and archaeal isolates expand coverage of the tree of life. *Nature Biotechnology* 35, 676–683.
- Orengo, C.A., Bateman, A., Uversky, V. (Eds.), 2013. *Protein Families: Relating Protein Sequence, Structure, and Function*. Wiley-Blackwell.
- Orengo, C.A., Thornton, J.M., Jones, D.T. (Eds.), 2003. *Bioinformatics: Genes, Proteins and Computers*. Oxford: BIOS Scientific.
- Shendure, J., Balasubramanian, S., Church, G.M., *et al.*, 2017. DNA sequencing at 40: Past, present and future. *Nature* 550, 345–353.
- Williamson, M., 2012. *How Proteins Work*. New York: Garland Science, Taylor & Francis Group, LLC.

Relevant Websites

<http://cath-tools.readthedocs.io/>

cath-tools.

http://www.cathdb.info/search/by_sequence

CATH.

http://www.cathdb.info/search/by_structure

CATH.

Secondary Structure Prediction

Jonas Reeb and Burkhard Rost, Technical University of Munich (TUM), Garching/Munich, Germany

© 2019 Elsevier Inc. All rights reserved.

Introduction

The study of proteins is fuelled by the desire to understand their function. The molecular understanding of function requires the understanding of structure. However, determining the three-dimensional (3D) structure of proteins is a highly non-trivial process. Despite substantial advances in technologies, the experimental determination of a single protein is still costly and might take years. Conversely, the advent of next generation sequencing techniques has tremendously increased the amount of raw sequence data about which there is very little structural knowledge (Goodwin *et al.*, 2016; Martinez and Nelson, 2010). Due to ever decreasing costs, the speed at which this sequencing data is being generated far outpaces the annotation of the data with structural (or functional) knowledge. This creates a divide that has been referred to as the sequence-structure gap. For instance, the UniProtKB database currently contains almost 60 million proteins which are at most 90% identical (The Uniprot Consortium, 2016; Suzek *et al.*, 2015). In contrast, the Protein Data Bank (PDB), the major database of protein 3D structures, contains just 48,000 protein structures at that similarity cutoff (Berman *et al.*, 2000).

The prediction of the protein secondary structure is one step toward bridging this gap. Due to its simplicity and elementary importance, secondary structure is also one of the earliest features that has been predicted from sequence and has grown over many decades into a mature field where predictions can be performed at relatively little computational cost, yet at a high performance.

If the 3D structure could be predicted directly, then one would also have an implicit prediction of secondary structure. However, despite crucial breakthroughs in the last 5–10 years, the number of proteins for which 3D structure can be predicted from sequence remains limited (Kinch *et al.*, 2016; Hopf *et al.*, 2012; Moult *et al.*, 2016; Marks *et al.*, 2012). Even for most proteins for which comparative modelling can infer 3D structures (Biasini *et al.*, 2014; Yang *et al.*, 2011), dedicated secondary structure prediction methods still tend to perform better (Eyrich *et al.*, 2001). Additionally, the computational costs of more advanced methods are substantial which is in contrast to the decrease in resources needed for generating novel sequence (Muir *et al.*, 2016). Hence, both effective and efficient annotation of this sequencing data is necessary.

On top of the knowledge about the structural elements, secondary structure predictions also serve as input to predict more complex protein features such as disordered regions, interaction sites or aspects of protein function. Solvent accessibility is another feature of protein structure and often predicted by many of the tools for secondary structure prediction.

This contribution succinctly summarizes the underlying biological concepts of protein secondary structure and continues discussing some of the main concepts in the field of secondary structure prediction before presenting a selection of such methods in more detail. Finally, the level at which state-of-the-art methods perform is examined.

Background

Secondary Structure Elements in Protein Folding

Protein 3D structure is encoded by its amino acid sequence which is sometimes misleadingly referred to as the “primary structure”. The order in which the 20 biogenic amino acids are arranged uniquely defines the structure and determines its folding in 3D (Anfinsen, 1973). Amino acids are linked by a covalent, so-called peptide bond which forms the backbone of the protein. Given the chemical properties of the amino acids and the physical constraints of this bond, the number of possible protein conformations is immense. The forces that drive the folding process that strives for a stable energetic minimum, are non-covalent interactions such as electrostatic interactions, much weaker but more common van der Waals interactions, or the hydrophobic effect, i.e., non-polar components aggregating in water (Kessel and Ben-Tal, 2011).

For the formation of secondary structure elements, a specific type of electrostatic interactions, the hydrogen bonds, are often claimed as the main stabilizing force. While this is most likely not the case, they still give structure to the elements in question and are important to overall protein stability (Kessel and Ben-Tal, 2011). Hydrogen bonds form between two dipoles which act as donor and acceptor, such as two atoms of amino acid side chains. However, in the case of secondary structure elements, the relevant atoms are part of the peptide bonds. Here, a hydrogen atom that is covalently bound to a nitrogen, gets in proximity of, and is therefore attracted by, an electronegative oxygen (Fig. 1). The stable secondary structure elements then further develop interactions between themselves and form a tertiary structure.

Various types of secondary structure are observed repeatedly out of which helices and strands are the most prominent ones. Both are characterized by specific patterns of hydrogen bonds. Due to these patterns, those two types are also often referred to as “regular secondary structure”. Most prediction methods target three “states” using those two regular types and merging everything else into the state “other”.

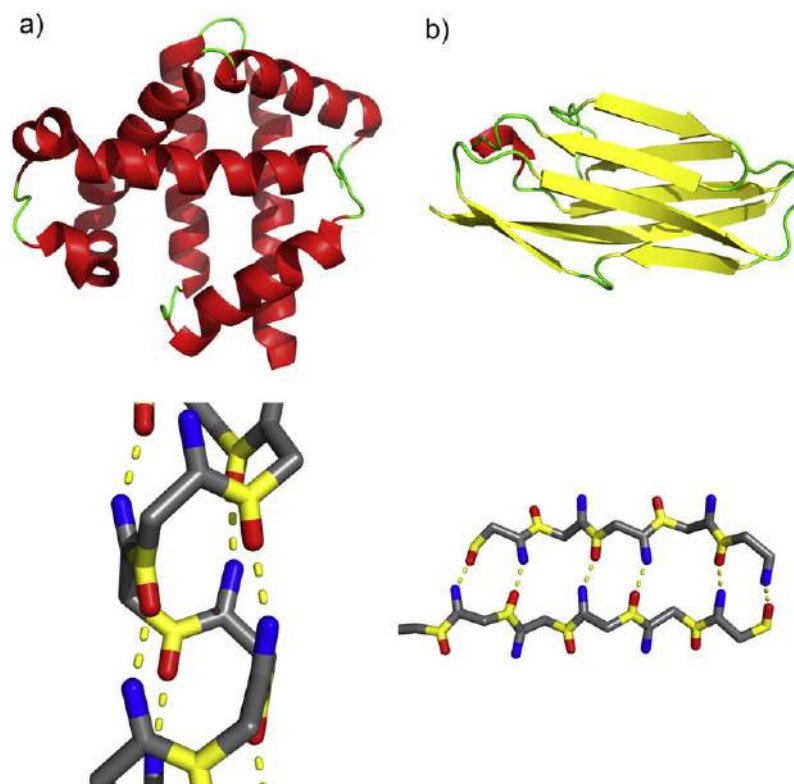


Fig. 1 α -helical and β -sheet secondary structure elements. The two most common secondary structure elements are visualized in PyMol (Schrodinger, 2015). Above, the protein is visualized in a cartoon view that highlights the secondary structure elements and hides details such as the amino acid side chains. Below, the protein backbone, without amino acid sidechains, of parts of the same secondary structures are shown in a sticks view. Here, oxygen atoms are colored blue, nitrogen yellow, hydrogen red and all others grey. Hydrogen bonds between atoms of the protein backbone are visualized as yellow dashed lines. (a) The human protein myoglobin is shown with α -helices highlighted in red and the rest of the protein in green (PDB identifier 3rgk). This is an “all-alpha” protein fold which consists of only α -helices and loops. (b) A part of the human vascular cell adhesion molecule is shown with β -sheets highlighted in yellow and loops in green. A small, single-turn α -helix can be seen in red at the left side.

Helices (α , 3_{10} , π)

As mentioned above, helices are defined by a repeating pattern of hydrogen bonds between residues. In an ideal α -helix, these bonds form between residues i and $i + 4$. This type of helix allows a relatively compact formation where four residues form a single “turn” of a helix. 3_{10} helices are more compact with just three residues (or ten atoms) forming a single “turn”. π -helices on the other hand are wider and complete one “turn” every five residues. Among these three types, α -helices are energetically most favorable and thus most often observed. In literature and the output of prediction methods, helices are typically abbreviated by the letter H.

Strands and sheets (β -strands, parallel and antiparallel β -sheets)

β -sheets are a more spacious type of secondary structure formed from β -strands. Strands consist of the protein backbone “zig-zagging”, typically for four to ten residues. Single β -strands are not energetically favorable. However, they can form β -sheets which are characterized by a pattern of hydrogen bonds between the residues on two different β -strands. Residue i in strand 1 forms a hydrogen bond with residue j in strand 2. The next bond can be formed in two different ways: either residue $i + 2$ in strand 1 binds to $j + 2$ in strand 2 (referred to as a parallel β -sheet), or residue $i + 2$ in strand 1 binds to $j - 2$ in strand 2 (referred to as antiparallel β -sheet). In contrast to the situation for helices, the hydrogen bonding residues in β -sheets might be far away in sequence. Most β -sheets are formed from more than two strands. Typically, β -strands and -sheets are abbreviated by the letter E.

Other/loops

All residues that do not participate in helices or strands are often joined in one category as “other”. Historically, this is also referred to as “loop” or, very misleadingly, as “(random) coil”. Although the residues in question do not form hydrogen-bonded regular secondary structure, they are often constrained by bonds to other residues and are certainly not all random. Some methods distinguish different types within the class “other” such as “(reverse) turns” or “bends” where the residues in question change the direction of the amino acid chain. This non-regular class is typically abbreviated by the letter L.

Inference of Secondary Structure Elements From 3D Protein Structures

DSSP

Define Secondary Structure of Proteins (DSSP) is the standard tool for the annotation of secondary structure elements from protein structures (Kabsch and Sander, 1983; Touw *et al.*, 2015). Based primarily on hydrogen bonding patterns and some geometric constraints, it assigns every residue to one of eight possible states. States corresponding to helical structures are α -, 3_{10} - and π -helices. β -bridges are short fragments that show β -sheet like binding patterns. Multiple consecutive β -bridges form the state referred to as β -ladder. Multiple ladders can then form the above-mentioned β -sheets. However, DSSP does not have a separate state for β -sheet residues and the one for β -ladders is used. The remaining three states, called turn, bend and other, describe loop structures. Typically, secondary structure prediction methods simplify these eight states into just three: α -helix, β -sheet and loop.

STRIDE

Given a high-resolution 3D structure, annotation of secondary structure elements remains a matter of definition to some degree. STRIDE represents an alternative approach to DSSP. It aims to provide secondary structure assignments that are more consistent with the assignments performed by experimentalists who determined the protein structure (Frishman and Argos, 1995; Heinig and Frishman, 2004). Next to hydrogen bonds it includes other aspects such as the backbone geometry in the form of dihedral angle propensities. Another difference to DSSP is that some of its decision thresholds have been optimized on data from the PDB which makes it a knowledge-based approach. Nonetheless, the agreement between DSSP and STRIDE annotations is very high at 95% (Martin *et al.*, 2005).

Assessing Secondary Structure Predictions

Assessing secondary structure predictions requires all the care generally exercised when evaluating the performance of machine learning models. One aspect is the careful choice of a meaningful scoring model.

The simplest and most commonly used performance measure is the three-state per-residue accuracy referred to as Q_3 . It represents the fraction of all residues that have been predicted accurately in the three states helix, strand, other. Despite its popularity, this score has several shortcomings. Q_3 ignores the uneven distribution in the three states: most residues are in the state other, i.e., methods that poorly predict the more important regular states (in particular the least common state of strand) might still reach high performance. Additional scores to measure these aspects include Q_H and Q_E , the percentages of correctly predicted helix and strand residues. Many other scores have been proposed and are useful in different contexts (Rost and Sander, 1993b). All these per-residue ignore the fact that secondary structure segments span over several residues. This shortcoming is corrected by per-segment scores (Rost *et al.*, 1994). The simplest such score compares the average length of predicted and observed segments. More advanced is a composite score referred to as segment overlap (SOV) (Fidelis *et al.*, 1999). Instead of measuring single-residue accuracy, SOV scores the correct placement of secondary structure segments. Such scores tend to give leeway with respect to predicting the precise begin and end positions of segments. Predictions that roughly predict the location of most segments score high.

Given performance measures, one also needs a test dataset to evaluate predictions. Authors exercise varying degrees of care in trying to report unbiased performance estimates. Best would be to assess all relevant methods based on proteins that differ substantially in sequence from those used for development of any of the methods.

One effort for such an assessment was EVA, a server that collected newly released protein structures as an unknown test set and then automatically queried secondary structure prediction methods for their predictions of those proteins' secondary structure elements (Eyrich *et al.*, 2001). This process was performed on a weekly basis and the results published online. A similar concept was followed by LiveBench (Rychlewski and Fischer, 2005). Unfortunately, both servers have now been offline for more than 10 years. CAMEO is a current effort in automated protein structure prediction evaluation and its operators have stated that it may include an assessment of secondary structure prediction in the future if there is a community interest (Haas *et al.*, 2013).

Approaches

Amino Acid Propensities

Since the very beginning, the prediction of secondary structure elements was based on the fact that amino acid occurrences are not evenly distributed between those elements but show preferences (Kessel and Ben-Tal, 2011). For example, due to their compactness helices often contain amino acids with small, or more precisely linear, side chains such as alanine, glutamic acid or methionine. On the other hand, proline is rarely found in helices. Its pyrrolidine ring and bound backbone amide group disturb the geometry and hydrogen bonding pattern of the helix. This leads to a kink in the helix that is energetically unfavorable and gives proline the nickname "helix breaker". In the same way β -sheets have preferences for certain amino acids, in particular at their termini. Although weak, these propensities suffice to predict secondary structure better than random. In one way or another even the most recent methods developed today still exploit such features. However, finding the right relationships is far from trivial as the formation of secondary structure depends on more than just the local amino acid content. The most extreme evidence for this are so-called chameleon sequences. These sequence fragments can fold into both helix and strand conformations depending on context (Jacoboni *et al.*, 2000; Guo *et al.*, 2007; Li *et al.*, 2015).

Basic Concepts of Secondary Structure Prediction

Even before high-resolution protein structures became available, the earliest methods used exactly these amino acid propensities mentioned in the previous section together with empirically defined rules to gain insights into the secondary structure content of proteins of interest. For example, Szent-Györgyi and Cohen described the relationship between helices and proline as outlined above (Szent-Györgyi and Cohen, 1957). Chou and Fasman determined a simple rule as to which a clustering of certain helix or strand “former” residues leads to formation of the respective secondary structure element (Chou and Fasman, 1974). One could refer to those early approaches as “first generation” methods because they ultimately scanned for features of single residues.

Stepping up, the “second generation” methods improved performance by including more information. The basic idea was to consider the sequence neighborhood of a residue, i.e., for a residue at position i those before it ($i - 1$, $i - 2$, ...) and those directly after it ($i + 1$, $i + 2$, ...). At the end of the 80s such statistical approaches were complemented by the first machine learning solutions. While any of the usual machine learning models is potentially applicable, (artificial) neural networks (NNs) have been particularly popular in the field. Early machine learning-based methods used a limited set of input features which may be as simple as the amino acid sequence itself. Typically, the sequence would be parsed in a sliding window of uneven size X , i.e., to predict the secondary structure of the central amino acid residue, all $\lfloor X/2 \rfloor$ neighboring residues are considered. With increasingly complex models and more computational power becoming available, a multitude of other features was added to increase the chances of the model finding the intricate correlations between amino acids which lead to a respective secondary structure. This allowed to predict strands and helices at equal levels of accuracy (Rost and Sander, 1993b). However, all those tools seemingly hit a performance limit since the information available within the window was too small and increasing the window size introduced more noise than signal.

This led to the leap into the “third generation” methods that combined machine learning with evolutionary information (Rost and Sander, 1993a). To use evolutionary information, one first needs to create a multiple sequence alignment (MSA). Toward this end, one needs to find all proteins with sufficiently high sequence similarity to a query which guarantees that those proteins have similar secondary structure. Most important here is not the number of related proteins, but the amount of diversity, i.e., the sequences more distantly related are more important to improve the prediction than those that are very similar. The MSA is then converted into a position-specific scoring matrix (PSSM), often also referred to as “profile”. This profile can be used by the prediction method to replace a much simpler 20-dimensional binary vector with 1 for the amino acid at that position and 0 for all 19 amino acids not at that position. An example of a simple neural network with homology information as input is shown in Fig. 2.

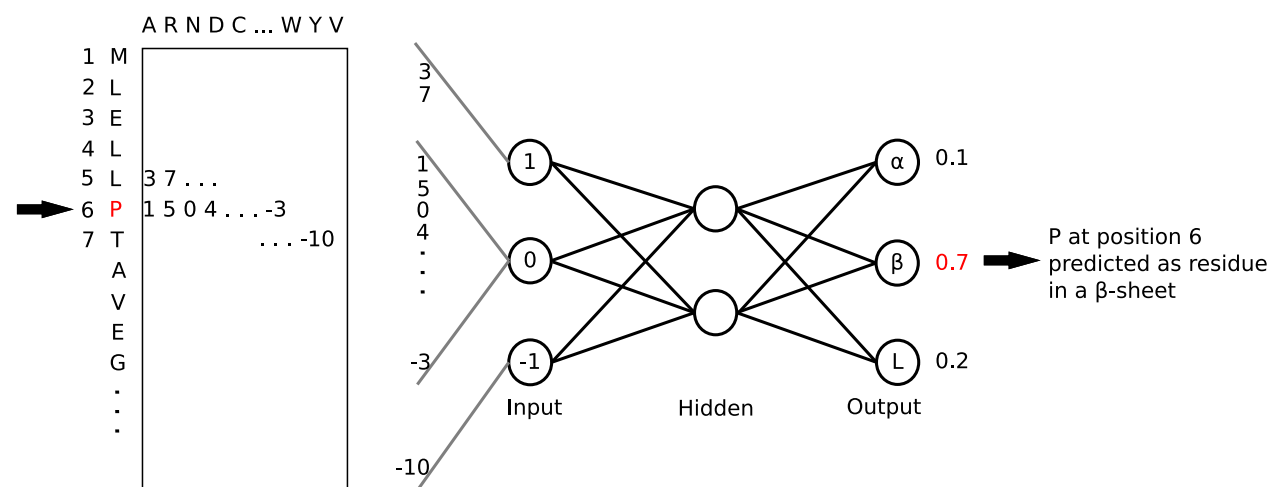


Fig. 2 A simple feed-forward neural network using homology information for the prediction of secondary structure elements. On the left, the input sequence for which the secondary structure is to be predicted runs from top to bottom. Right of it, the position specific scoring matrix (PSSM) for this input sequence is shown. Such a PSSM is created for example by performing a PSI-BLAST query with the input sequence against a large sequence database. The matrix contains information about which amino acids are commonly found in homologous sequences and which substitutions are rare. Typically, these values are used as log-odds such that negative values determine an unlikely substitutions and positives ones a likely one. These values can then be used as input to the neural network which leads to 20 inputs for every residue. To allow for a better visualization, the network shown here uses a very small sliding window of just three residues. Therefore, when predicting the secondary structure state at position 6, the PSSM values of positions 5, 6 and 7 are used as input to the network. The three input nodes labelled -1 , 0 and 1 are in fact 20 nodes each, leading to a total of 60 inputs. These values are then fed through a hidden layer to the output layer which predicts one of the three secondary structure states. Weights on the connections between these layers are optimized during the training of the network. A typical neural network for the prediction of secondary elements may use a sliding window of size 13 and have hundreds of nodes in the hidden layer. In deep learning, more than one hidden layer is used and in recurrent neural networks nodes from one layer may be connected to those of a previous layer.

Over the last few years, increasingly cheap, yet powerful hardware capable of highly parallelized calculations has given a new rise to deep learning approaches that has also reached the field of secondary structure prediction. These approaches are also based on NNs. However, they have more than one hidden layer and their architecture often differs significantly from that of traditional feed-forward NNs.

Results and Discussions

Secondary Structure Prediction Methods

In the following, a few secondary structure prediction methods are described in more detail starting with some of the oldest approaches to the current state-of-the-art (Table 1). The methods were chosen to represent some of the historically most valuable publications, to give an overview of how the field developed and highlight some of the currently most promising methods.

Prediction based on amino acid composition statistics

GOR, named by the initials of its authors is an information theoretical approach first published in 1978 (Garnier *et al.*, 1978, 1996). Using a sliding window of 17 residues the method calculates the most likely secondary structure element based on amino acid propensities which were extracted from a database of proteins with known secondary structure. The method was later extended by updating the underlying database with more recent data in GOR II. In addition to the previous single-residue statistics, GOR III included correlations between pairs of residues, i.e., between the central amino acid and amino acids at all other positions in the window. GOR IV further increased the number of residue pairs considered. Instead of only pairs with the central residue, all possible residue pairs in the window were now included in the calculation. In 2002 the final version of the method, GOR V, further extended the approach by adding statistics of amino acid triplets, as well as by including homology information, choosing the windows size depending on sequence length and again updating the underlying database (Kloczkowski *et al.*, 2002).

Prediction based on two levels of neural networks

PHDsec is one of the earliest methods to apply machine learning to the task of secondary structure prediction and the first to harness homology information to further improve it (Rost and Sander, 1993a). The method is based on NNs and uses only sequence information as input. For a given input sequence an MSA with homologous sequences is constructed from the HSSP database (Sander and Schneider, 1991). Then the amino acid occurrences at every position in the sequence are calculated. The method uses a sliding window of 13 residues and every position translates to 21 input units which are fed with the amino acid occurrences from the MSA at that position. 21 units are required (as opposed to just 20) to model cases near the N- and C-terminus of the input sequence where parts of the sliding window are “empty”. The total 273 input units are connected to a hidden layer and an output layer which contains three nodes corresponding to an α -helix, β -sheet or loop prediction for the central residue in the sliding window. The output of this first NN is used as input to another NN with a 17-residue sliding window. This second layer aims to smooth the output such that segments are predicted more continuously. Later extensions to the method added more input features derived from the MSA as well as global amino acid composition information (Rost and Sander, 1994). The method has also been renamed several times, with the most recent version, ReProf, being available as part of the PredictProtein webserver (Yachdav *et al.*, 2014). PSIPRED is another common NN based prediction method with an approach similar to that of the PHDsec family (Jones, 1999). It uses sequence profiles

Table 1 A selection of secondary structure prediction methods discussed in this article

Method	Learning model	Other predicted features	Primary reference	Webserver address
GOR I-V	Bayes rules on amino acid propensities	None	Kloczkowski <i>et al.</i> (2002)	
PhDsec/ReProf	Neural Network	Solvent accessibility	Rost and Sander (1994)	https://predictprotein.org/
PSIPRED	Neural network	None	Jones (1999)	http://bioinf.cs.ucl.ac.uk/psipred/
JPred4	Neural network	None	Drozdetskiy <i>et al.</i> (2015)	http://www.compbio.dundee.ac.uk/jpred/
Frag1D	Database Lookup (Fragment matching)	None	Zhou <i>et al.</i> (2010)	http://frag1d.bioshu.se/
SSpro5	Database lookup + Deep Neural Network	None	Magnan and Baldi (2014)	http://scratch.proteomics.ics.uci.edu/
S2D	Neural Network	Disorder	Sormanni <i>et al.</i> (2015)	http://www-mvsoftware.ch.cam.ac.uk/
SPIDER2	Deep Neural Network	Solvent accessibility, torsion angles	Heffernan <i>et al.</i> (2015)	http://sparks-lab.org/server/SPIDER2/
Raptor X Property	Deep Neural Network	Solvent accessibility, disorder	Wang <i>et al.</i> (2016b)	http://raptorx.uchicago.edu/StructurePropertyPred/predict/

generated by PSI-BLAST (Altschul, 1997) as input to a NN with a sliding window of size 15, followed by a second smoothing NN with the same window size. The method has been continually updated since its first publication and the network architecture was recently improved (Buchan *et al.*, 2013). Finally, JPred4, a webserver built on the underlying Jnet also employs the two-level neural network methodology and has been updated many times since its first publication in 2000 (Drozdetskiy *et al.*, 2015; Cole *et al.*, 2008; Cuff and Barton, 2000). The method builds sequence profiles using PSI-BLAST as well as HMMER3, a hidden Markov model-based sequence search tool (Eddy, 2011), and trains two separate NN-pairs on the respective data. If these two models do not agree on a residue's prediction a third "jury" NN is used on the output of the previous two.

Prediction based on deep learning

SSpro5 is the most recent iteration of a secondary structure predictor which first performs a lookup of already known structures (Magnan and Baldi, 2014). The input sequence is compared to structures in the PDB using BLAST (Camacho *et al.*, 2009). If the resulting (local) alignments satisfy a set of similarity criteria the most common secondary structure class annotated in DSSP is used as the prediction for the respective residue. If no similar positions exist or the DSSP annotations do not form a majority, a neural network predictor is invoked to fill in predictions for the missing residues. This model consists of 100 bi-directional recurrent NNs. In contrast to the window-based feed-forward NNs covered so far, these networks can parse information of an arbitrarily large input sequence at once. In addition to the middle segment with the residue of interest and its neighbors, two other components of the network parse the rest of the sequence (Pollastri *et al.*, 2002). This could allow the recognition of global contacts which go beyond the correlations captured in the local window. The number of 100 networks result from the double cross-validation procedure which first splits the training set into 10 folds and then performs 10-fold cross-validation on each of them.

SPIDER2 is a deep-learning approach that combines the prediction of secondary structure elements with the prediction of solvent accessibility, backbone torsion angles and two more angles around the backbone C_α atoms (Heffernan *et al.*, 2015). The underlying idea is that these properties all correlate with each other to some degree. This makes the machine learning model aware of all classes at the same time which might improve prediction performance. SPIDER2 achieves this with an iterative approach: Given a PSI-BLAST PSSM of the input sequence and seven physicochemical indices, three NNs are separately trained for the prediction of secondary structure, solvent accessibility and the backbone angles. In the two following iterations, the input to each respective network is extended by the input from the other two networks. For example, predicted angles and solvent accessibility are used as input to the secondary structure predictor. In sum, this leads to three deep NN predictors with three hidden layers each. An improvement of the method using long short-term memory neural networks is currently in development.

Similar to SPIDER2, RaptorX Property is a webserver predicting several sequence features, including secondary structure and solvent accessibility (Wang *et al.*, 2016a,b). However, the prediction models are treated separately and do not interfere with each other. The authors use a deep convolutional neural field, which is the combination of a deep convolutional NN (DCNN) with seven hidden layers and a final output layer which together with the last hidden layer forms a conditional neural field (CNF). The rationale behind this design is that the DCNN can capture global relationships that go beyond the local window size of 11 residues while the final CNF learns the correlations between the secondary structure of adjacent residues. The input to the network is either just the sequence or a PSI-BLAST profile of that sequence.

Prediction based on other approaches

Frag1D performs secondary structure prediction based on a fragment matching approach (Zhou *et al.*, 2010). Given a query sequence, a sliding window of nine residues is used to find the 100 best-scoring fragments in a database of proteins with known structure. These 100 fragments are selected by comparing PSI-BLAST profiles and structural profiles between the input and database sequences. The structural profiles were created from so-called ShapeStrings which are defined based on the torsion angle pairs of the peptide bond. Same as for the direct sequence comparison, the database was searched for similar nine-residue stretches of ShapeStrings to build the structural profile which represents the conservation pattern in similar local structures. Further weighting the 100 fragments results in a preliminary secondary structure prediction. Given this prediction, the structural profile of the query sequence can now be computed and the whole process is repeated once to yield the final prediction.

s2D combines secondary structure prediction with the identification of disordered regions in proteins (Sormanni *et al.*, 2015). Such regions do not adopt a well-defined structure but instead fluctuate between different conformations (Habchi *et al.*, 2014). Typically, other methods extract secondary structure elements mostly from X-Ray crystallography structures through tools such as DSSP. In contrast, the s2D training data consists of annotations extracted from the chemical shifts of structures determined by nuclear magnetic resonance. In this way, all residues that show neither α -helical nor β -sheet properties can be considered as some form of disordered segments. Given these annotations, two NNs are trained with window sizes 11 and 15 but otherwise equal architecture. PSI-BLAST profiles are used as input to these networks. Next, a global network is trained with the same profiles and the output of the previous two networks resulting in a prediction of the mean secondary structure content in the whole sequence. Finally, the output of that network is combined with the initial inputs in a NN that smooths the prediction with a sliding window of five residues.

Estimates of Prediction Performance

As mentioned above in Section Assessing Secondary Structure Predictions two previous efforts for an automated independent assessment of secondary structure prediction are not maintained any longer. A related effort is CASP, a biannual challenge in which organizers collect

target proteins with known but unpublished structures (Moult *et al.*, 1995). The target sequences are given to the community who can submit their predictions for the proteins' structures. After the structures are publicly disclosed, the assessors evaluate the performance of the prediction methods on these previously unseen proteins. This enables an independent evaluation where neither the competition organizers nor the participating groups know the answer at the time of submission. CASP has a focus on protein 3D structure prediction but in the past also included an evaluation of secondary structure prediction methods. In CASP5 assessors found that helices are predicted better than sheets and that sequences with no homology to a solved structure were harder to predict (Aloy *et al.*, 2003).

Overall, methods seem to have approached a saturation point with only small improvements over the previous years. Therefore, public assessments were stopped. This means that current estimates of prediction performance are only regularly given by the authors of newly developed methods. Although these may have the best intentions in providing a fair comparison to previous approaches they typically focus only on the flawed Q_3 score and their reported numbers may still be biased to favor their own method – for example, by the selection of a specific dataset. Traditionally, reported performance values have often proven to be overestimated. Apart from over-training, over-estimates can also be caused by discovering completely new types of structures. Ultimately, one therefore has two ways to evaluate performance by either using old results from the “last public” assessments, thereby excluding all new methods, or using method-skewed values from publications, likely over-estimating performance for the methods reported. In this chapter, the second solution was favored. However, readers must be aware that the published evaluation numbers need to be viewed with substantial skepticism.

Keeping that in mind, the most recent methods such as s2D, JPred4, RaptorX Property and SSpro5, all report Q_3 values around 80–85%. These values seem realistic, given the last independently determined performance estimates along with the increase in database size. However, choosing the best method among them is impossible given their heterogeneous evaluation schemes.

Future Directions

Without any independent evaluations, it is hard to say whether the top has already been reached in secondary structure prediction. Reinstating secondary structure prediction evaluation through a CAMEO revival of approaches such as EVA and LiveBench may help greatly. Certainly, creating such an independent evaluation entity is far from trivial and great care will be necessary when choosing the underlying datasets and scoring procedure. Clearly the limit is not given by a Q_3 of 100%. Due to errors in experimental structure determination and since proteins are in motion, secondary structure assignments will always contain some amount of error (Kihara, 2005). For the time being, the main problem with the assessment of secondary structure prediction is that although the tools are essential as input to many subsequent “higher level” prediction methods, there are very few incentives to invest substantial efforts to doing evaluations right. At the same time, it has become so difficult to publish new prediction methods in this field that almost nothing less than the claim that the ultimate has been reached will even appear on the screen of experts. There are many good reasons for these two opposing realities but they clearly make it difficult to assess today's state-of-the-art. It also remains debatable whether there is a large value in another method that shows increasingly smaller improvements over the current state-of-the-art. With the recent advances in the prediction of 3D protein structures from sequence, the interest in dedicated secondary structure prediction methods may continue to fade. However, at this point they cannot be substituted yet, since the difference in runtime between the two fields reaches several orders of magnitude and secondary structure prediction methods currently still perform better than inferring helices and sheets from predicted 3D structures (Faraggi *et al.*, 2012).

Closing Remarks

Predicting protein secondary structure elements from sequence is among the oldest prediction approaches in the field. Benefitting from decades of work, predictions are now both fast and highly accurate although determining just how accurate exactly is proving hard.

Recently, interest in these predictions has decreased, partly owing to a focus shift towards higher goals such as the prediction of protein 3D structures. Nonetheless, the respective methods are still relevant and a crucial help in making sense of the current sequence data deluge.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Artificial Intelligence and Machine Learning in Bioinformatics. Biomolecular Structures: Prediction, Identification and Analyses. Data Mining: Classification and Prediction. Data Mining: Prediction Methods. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Machine Learning in Bioinformatics. Multilayer Perceptrons. Natural Language Processing Approaches in Bioinformatics. Protein Structural Bioinformatics: An Overview. Protein Three-Dimensional Structure Prediction. Protocol for Protein Structure Modelling. Sequence Analysis. Sequence Composition. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Supervised Learning: Classification. Transmembrane Domain Prediction

References

- Aloy, P., Stark, A., Hadley, C., Russell, R.B., 2003. Predictions without templates: New folds, secondary structure, and contacts in CASP5. *Proteins: Structure, Function and Genetics* 53, 436–456.
- Altschul, S., 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25, 3389–3402.
- Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. *Science* 181, 223–230.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28, 235–242.
- Biasini, M., Bienert, S., Waterhouse, A., *et al.*, 2014. SWISS-MODEL: Modelling protein tertiary and quaternary structure using evolutionary information. *Nucleic Acids Research* 42, W252–W258.
- Buchan, D.W.A., Minnici, F., Nugent, T.C.O., Bryson, K., Jones, D.T., 2013. Scalable web services for the PSIPRED protein analysis workbench. *Nucleic Acids Research* 41, 349–357.
- Camacho, C., Coulouris, G., Avagyan, V., *et al.*, 2009. BLAST+: Architecture and applications. *BMC Bioinformatics* 10, 421.
- Chou, P.Y., Fasman, G.D., 1974. Prediction of protein conformation. *Biochemistry* 13 (2), 222–245.
- Cole, C., Barber, J.D., Barton, G.J., 2008. The Jpred 3 secondary structure prediction server. *Nucleic Acids Research* 36, 197–201.
- Cuff, J.A., Barton, G.J., 2000. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* 40, 502–511.
- Drozdetskiy, A., Cole, C., Procter, J., Barton, G.J., 2015. JPred4: A protein secondary structure prediction server. *Nucleic Acids Research* 43, W389–W394.
- Eddy, S.R., 2011. Accelerated profile HMM searches. *PLOS Computational Biology* 7. doi:10.1371/journal.pcbi.1002195.
- Eyrich, V., Marti-Renom, M.A., Przybylski, D., *et al.*, 2001. EVA: Continuous automatic evaluation of protein structure prediction servers. *Bioinformatics* 17, 1242–1243.
- Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., Zhou, Y., 2012. SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of Computational Chemistry* 33, 259–267.
- Fidelis, K., Rost, B., Zemla, A., 1999. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins* 223, 220–223.
- Frishman, D., Argos, P., 1995. Knowledge-based protein secondary structure assignment. *Proteins-Structure Function and Genetics* 23, 566–579.
- Garnier, J., Gibrat, J.-F., Robson, B., 1996. GOR method for predicting protein secondary structure from amino acid sequence. *Methods in Enzymology* 266, 540–553.
- Garnier, J., Osguthorpe, D.J., Robson, B., 1978. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology* 120, 97–120.
- Goodwin, S., McPherson, J.D., McCombie, W.R., 2016. Coming of age: Ten years of next-generation sequencing technologies. *Nature Reviews Genetics* 17, 333–351.
- Guo, J.-T., Jaromczyk, J.W., Xu, Y., 2007. Analysis of chameleon sequences and their implications in biological processes. *Proteins* 67, 548–558.
- Haas, J., Roth, S., Arnold, K., *et al.*, 2013. The protein model portal – A comprehensive resource for protein structure and model information. *Database* 2013, 1–8.
- Habchi, J., Tompa, P., Longhi, S., Uversky, V.N., 2014. Introducing protein intrinsic disorder. *Chemical Reviews* 114, 6561–6588.
- Heffernan, R., Paliwal, K., Lyons, J., *et al.*, 2015. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific Reports* 5, 11476. doi:10.1038/srep11476.
- Heinig, M., Frishman, D., 2004. STRIDE: A web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Research* 32, 500–502.
- Hopf, T.A., Colwell, L.J., Sheridan, R., *et al.*, 2012. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell* 149, 1607–1621.
- Jacoboni, I., Martelli, P.L., Fariselli, P., Compiani, M., Casadio, R., 2000. Predictions of protein segments with the same aminoacid sequence and different secondary structure: A benchmark for predictive methods. *Proteins* 41, 535–544.
- Jones, D.T., 1999. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology* 292, 195–202.
- Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.
- Kessel, A., Ben-Tal, N., 2011. Protein structure. In: *Introduction to Proteins*. Boca Raton, FL: CRC Press.
- Kihara, D., 2005. The effect of long-range interactions on the secondary structure formation of proteins. *Protein Science: A Publication of the Protein Society* 14, 1955–1963.
- Kinch, L.N., Li, W., Monastyrsky, B., Kryshchovych, A., Grishin, N.V., 2016. Evaluation of free modeling targets in CASP11 and ROLL. *Proteins: Structure, Function and Bioinformatics*. doi:10.1002/prot.24973.
- Kloczkowski, A., Ting, K.L., Jernigan, R.L., Garnier, J., 2002. Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins: Structure, Function and Genetics* 49, 154–166.
- Li, W., Kinch, L.N., Karplus, P.A., Grishin, N.V., 2015. ChSeq: A database of chameleon sequences. *Protein Science* 24, 1075–1086.
- Magnan, C.N., Baldi, P., 2014. SSpro/ACCpro 5: Almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics (Oxford, England)* 30, 2592–2597.
- Marks, D.S., Hopf, T.A., Chris, S., Sander, C., 2012. Protein structure prediction from sequence variation. *Nature Biotechnology* 30, 1072–1080.
- Martinez, D.A., Nelson, M.A., 2010. The next generation becomes the now generation. *PLOS Genetics* 6, e1000906.
- Martin, J., Letellier, G., Marin, A., *et al.*, 2005. Protein secondary structure assignment revisited: A detailed analysis of different assignment methods. *BMC Structural Biology* 5, 17.
- Moult, J., Fidelis, K., Kryshchovych, A., Schwede, T., Tramontano, A., 2016. Critical assessment of methods of protein structure prediction: Progress and new directions in round XI. *Proteins* 84 (Suppl. 1), 4–14.
- Moult, J., Pedersen, J.T., Judson, R., Fidelis, K., 1995. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Genetics* 23, ii–iv.
- Muir, P., Li, S., Lou, S., *et al.*, 2016. The real cost of sequencing: Scaling computation to keep pace with data generation. *Genome Biology* 17, 53.
- Pollastri, G., Przybylski, D., Rost, B., Baldi, P., 2002. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics* 47, 228–235.
- Rost, B., Sander, C., 1993a. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proceedings of the National Academy of Sciences* 90, 7558–7562.
- Rost, B., Sander, C., 1993b. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology* 232 (2), 584–599.
- Rost, B., Sander, C., 1994. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function, and Bioinformatics* 19, 55–72.
- Rost, B., Sander, C., Schneider, R., 1994. Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology* 235, 13–26.
- Rychlewski, L., Fischer, D., 2005. LiveBench-8: The large-scale, continuous assessment of automated protein structure prediction. *Protein Science* 14, 240–245.
- Sander, C., Schneider, R., 1991. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics* 9, 56–68.
- Schrodinger, L.L.C., 2015. The PyMOL molecular graphics system, Version 1.8.
- Sormanni, P., Camilloni, C., Fariselli, P., Vendruscolo, M., 2015. The s2D method: Simultaneous sequence-based prediction of the statistical populations of ordered and disordered regions in proteins. *Journal of Molecular Biology* 427, 982–996.
- Suzek, B.E., Wang, Y., Huang, H., Mcgarvey, P.B., Wu, C.H., 2015. UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* 31, 926–932.
- Szent-Györgyi, A.G., Cohen, C., 1957. Role of proline in polypeptide chain configuration of proteins. *Science* 126, 697.

- The Uniprot Consortium, 2016. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45, 1–12.
- Touw, W.G., Baakman, C., Black, J., *et al.*, 2015. A series of PDB-related databanks for everyday needs. *Nucleic Acids Research* 43, D364–D368.
- Wang, S., Li, W., Liu, S., Xu, J., 2016a. RaptorX-Property: A web server for protein structure property prediction. *Nucleic Acids Res* 44, W430–W435.
- Wang, S., Peng, J., Ma, J., Xu, J., 2016b. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports* 6, 18962.
- Yachdav, G., Kloppmann, E., Kajan, L., *et al.*, 2014. PredictProtein – An open resource for online prediction of protein structural and functional features. *Nucleic Acids Research* 42, W337–W343.
- Yang, Z., Lasker, K., Schneidman-Duhovny, D., *et al.*, 2011. UCSF Chimera, MODELLER, and IMP: An integrated modeling system. *Journal of Structural Biology* 179 (3), 269–278.
- Zhou, T., Shu, N., Hovmöller, S., 2010. A novel method for accurate one-dimensional protein structure prediction based on fragment matching. *Bioinformatics (Oxford, England)* 26, 470–477.

Relevant Websites

<https://www.cameo3d.org>

CAMEO.

<https://www.wwpdb.org/>

The Protein Data Bank.

Biographical Sketch



Jonas Reeb is a PhD student in the Rostlab at the Technical University of Munich, Germany (TUM). During his studies at TUM he has worked on evaluating methods for the prediction of transmembrane helices from sequence and on developing a crystallization propensity predictor for the NYCOMPS structural genomics pipeline. His doctoral thesis focusses on predicting the effects of sequence variants.



Burkhard Rost is a professor and Alexander von Humboldt award recipient at the Technical University of Munich, Germany (TUM). Rost was the first to combine machine learning with evolutionary information, and realized this combination to predict secondary structure. Since, his group has repeated this success in many other tools predicting aspects of protein structure and protein function. All tools are available through the first internet server in the field of protein structure prediction (PredictProtein), which has been online for over 25 years. Over the last years, the Rostlab has been shifting focus on the development of methods that predict and annotate the effect of sequence variation and their implications for precision medicine and personalized health.

Protein Three-Dimensional Structure Prediction

Sanne Abeln and Klaas Anton Feenstra, Center for Integrative Bioinformatics VU (IBIVU), The Netherlands
Jaap Heringa, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

© 2019 Elsevier Inc. All rights reserved.

What is the Protein Structure Prediction Problem?

Predicting the Structure for a Protein Sequence

This article revolves around a simple question: “given an amino acid sequence, what is the folded structure of the protein?” (Fig. 1) Even though this seems like a simple question, the answer is far from straightforward. In fact, whether we can give an answer at all depends heavily on the sequence in question and available protein structures that can be used as modelling templates. While the number of structures deposited in the Protein Data Bank (PDB) (Berman *et al.*, 2000) continues to rise rapidly (see “Relevant Websites section”), the number of sequenced genes rises much faster. The large and widening gap between protein structures and sequences makes structure prediction an important problem to solve. Fortunately, recently developed methods can use these large resources of sequence data to increase the quality of some predictions. Here, we will give an overview of current structure prediction methods, and describe some tools that provide insight into how reliable the structure predicted will be.

The typical problem is that we want to generate a structural model for a protein with a sequence, but without an experimentally determined structure. In this article, we will build up a workflow for tackling this problem, starting from the easy options that, if applicable, are likely to generate a good structural model, and gradually working up to the more hypothetical options whose results are much more uncertain.

Another very important remark is in place here: the modelling strategy should depend heavily on what we want to do with the structure. Do we want to predict where the functional site of the protein is, whether a specific substrate binds, or if a certain residue may be exposed to the surface? These different questions imply a different degree of accuracy in the answer, and may lead to choices regarding technology and methods to carry out these predictions. It is important to keep in mind that one of the most important aspects of any scientific model is whether a research question may be answered with the model produced or not. Even if we do have an experimental structure available, some of these questions may not be straightforward to answer; we will come back to this issue later in the article.

Structure is More Conserved Than Sequence

Almost all structure prediction relies on the fact that, for two homologous proteins, structure is more conserved than (Bajaj and Blundell, 1984; see Fig. 2). The real power of this observation manifests itself when we turn this statement around: if two protein sequences are similar, these two proteins are likely to have a very similar structure. The latter statement has very important consequences. It means that if our sequence of interest is similar to a protein sequence with a known structure, we have a good starting point for a structural model. In such a scenario we use sequence similarity, suggesting an homologous relation between the proteins, to predict the structure. The vast majority of accurate structure prediction methods use structure conservation as an underlying principle; while methods that have been developed to deal with the more difficult modelling questions, exploit the sequence-structure-conservation relation in an advanced manner, as discussed towards the end of this article.

Terminology in Structure Prediction

Firstly, we should take care to lay down a good problem definition. Here we will generously borrow the nomenclature from the Critical Assessment of Protein Structures (CASP). CASP is a scientific competition, in which structure prediction groups and

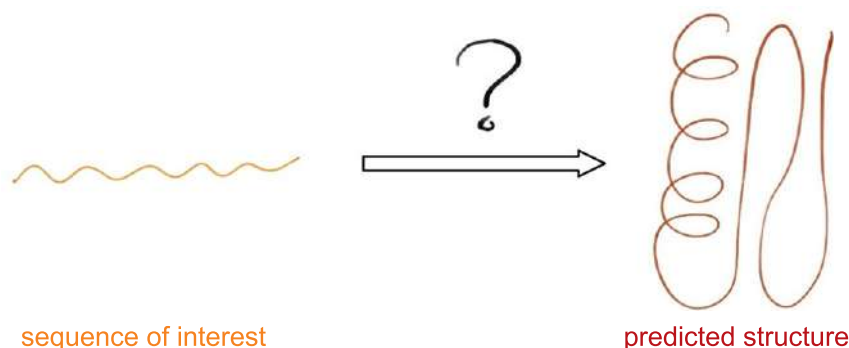


Fig. 1 Structure prediction methods try to answer the question: Given an amino acid sequence, what is the folded protein structure?

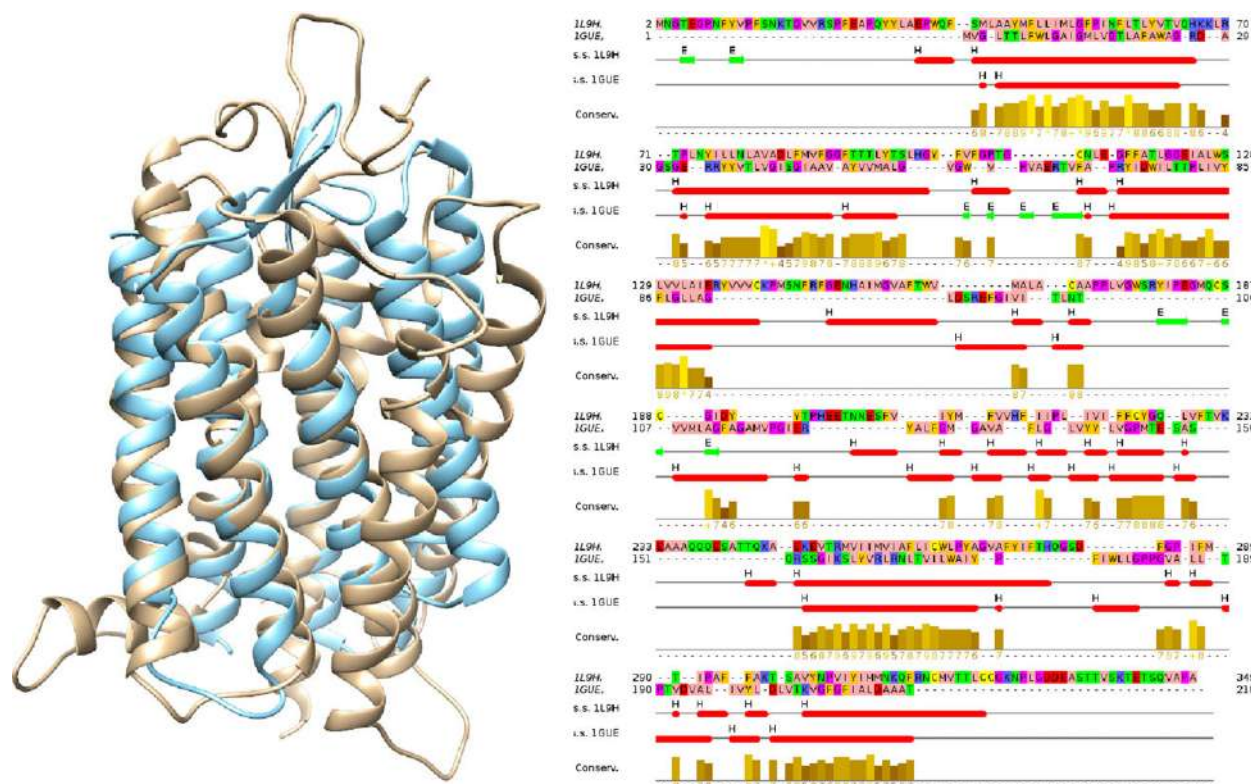


Fig. 2 Protein structure is more conserved than sequence. Here the output of a structural alignment is shown on the left, created using *Chimera* (Molecular graphics and analyses were performed with the UCSF Chimera package. Chimera is developed by the Resource for Biocomputing, Visualisation, and Informatics at the University of California, San Francisco (supported by NIGMS P41-GM103311)). Reproduced from Pettersen, E. F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF Chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25 (13), 1605–1612. The structural alignment shows both proteins are highly similar; the RMSD is 2.3 over 144 aligned residues. Furthermore, the function of the two proteins, one from cattle (PDB:1L9H, light brown) and one from a archaeon (PDB:1GUE, light blue), is similar: both are light sensitive rhodopsins, used for vision and phototaxis, respectively. However, as can be seen in the sequence alignment on the right, the sequence identity is only 7%. This is lower than would be expected for any two random sequences. The alignment shown is based on the structural alignment on the left, and visualised using *JalView*. Reproduced from Waterhouse, A.M., Procter, J.B., Martin, D.M.A., Clamp, M., Barton, G.J., 2009. Jalview Version 2 – A multiple sequence alignment editor and analysis workbench. *Bioinformatics* 25 (9), 1189–1191.

structure prediction servers compete to predict the structure for an unknown sequence, that has been running since 1995 (Moult *et al.*, 1995). The sequence for which we will predict a structure is called the *target* sequence. If there is a suitable structure to build a model for our query sequence we call this structure a *template*, see also Fig. 3. Using the structure of the template and using the *sequence alignment* between the template and the target sequence, we can create one or more structural *models*: the predicted structure, for a target sequence. In CASP structural models from different prediction methods, are compared to the experimentally determined solution or target structure.

Different Classes of Structure Prediction Methods

We can classify structure prediction strategies into two categories of difficulty: template-based modelling, and template-free modelling (see Fig. 4). In the first case, it is possible to find a suitable template for the target sequence in the PDB, as a basis for the model, whereas for template-free modelling no such experimental structure is available. Note that it may not be trivial at all to find out in which of these two categories a structure prediction problem falls. Only if we can find a close homolog – based on sequence similarity – in the PDB we can be sure that a template based modelling strategy will suffice; this is also referred to as *homology modelling*. With a template, the constraints from the alignment between the model and the template sequence, in addition to the template structure, will give sufficient constraints to build a structural model for the target sequence. Even in this case, small missing substructures in the alignment, e.g., loops, may require a template-free modelling strategy.

If no close homologs are available in the PDB, we may need to use more advanced template finding strategies, such as remote homology detection or fold recognition methods.

If no suitable template is available, we will need to resort to a template-free modelling strategy. In the *ab initio* approach knowledge-based energy terms are used to generate structural models based on the sequence of the template alone. Small, suitable fragments, from various PDB structures are assembled to generate possible structural models. In some cases, we can find

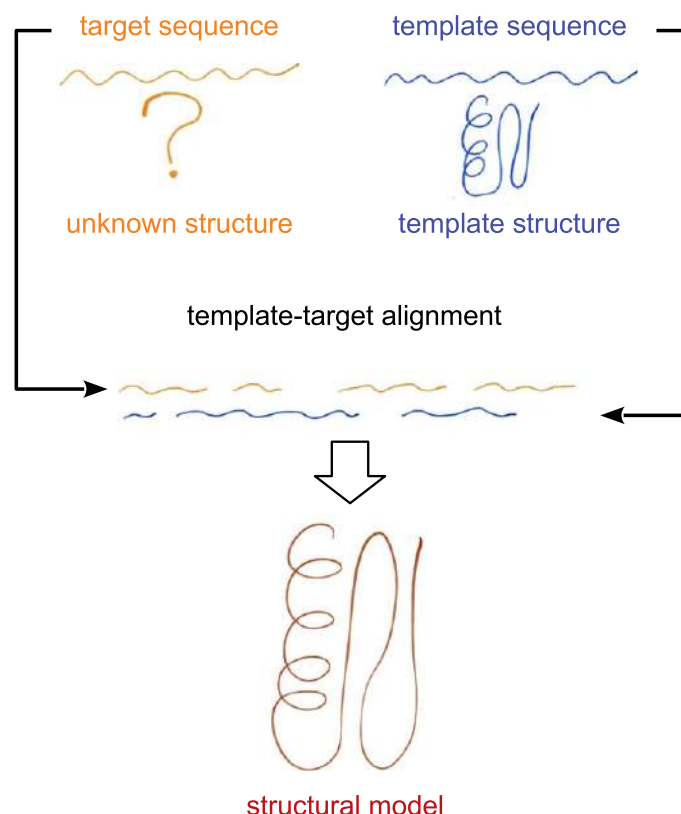


Fig. 3 Terminology used in protein structure prediction. We start from our protein of interest (with no known structure): the target sequence. First step is find a matching protein: a template sequence with known structure; the template structure. We then create a template-target sequence alignment, and from this alignment create the structural model which is the solution structure for our target protein.

additional constraints, for example from experiments, such as NMR or cryoEM, or from contact prediction methods; in that case we have a much better chance of building a suitable model (Moult *et al.*, 2016). In fact, we could consider such constraints an alternative for the constraints provided by homology.

Lastly, several steps may be taken to refine the model, and to select the most likely model, from several model building attempts. Note that some structure prediction methods, may also include variations of model refinement and model selection steps higher up in the modelling workflow.

Domains

So far we have implied that we may follow the above strategy for an entire protein, however, this generally is not the case. In fact many proteins consist of multiple domains. If this is the case, it is wise to also run one or multiple disorder prediction methods on the target sequence; Any large regions (> 25 residues) predicted to be disordered should be left out for further structure prediction and template finding.

Most structure prediction methods only work well at the domain level. This means that a sequence first needs to be split in multiple domains, before we can start to make models. However, domain splitting is often ambiguous both given the sequence and the structure, while combining models built from various domains is far from trivial. In practice, this means that multiple templates might be necessary for a single target sequence and that it is difficult to resolve the orientation of the modelled domains with respect to each other.

Predicting the orientation for several domains is currently an unsolved problem, unless there is a suitable, homologous, template available – with the domains in the same orientation. In some cases, coarse constraints on the domain orientations such as data from small-angle scattering experiments, or distance restraints from NMR, chemical cross-links or co-evolution may help to put different homology models in the correct orientation.

Typically, it only makes sense to generate a model, be it template-based or template-free, for a single domain. In fact, in CASP model predictions are assessed per structural domain, separately. Therefore, it is essential to split the target sequence into its constituent domains – which is a non-trivial task, particularly if no homologous templates are available for each of the domains.

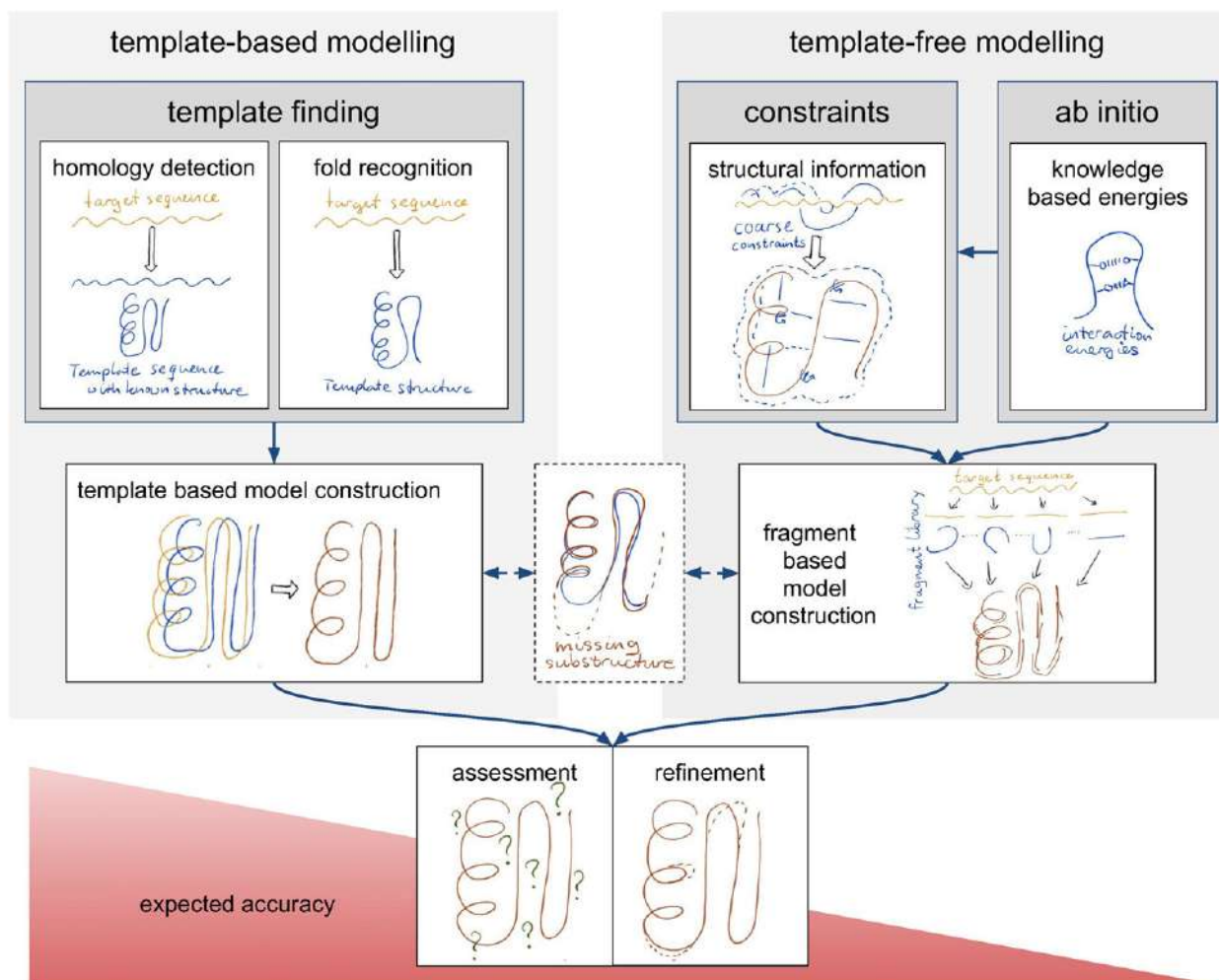


Fig. 4 Overview of Structure Prediction. Template-based modelling: a template is found on the basis of homology between the template and the target. Fold recognition: no obvious homologous structure can be found in the PDB, we need fold recognition methods to find a suitable template. Template-free modelling: no suitable template for protein domains can be found. Without template, we need to use a combination of coarse constraints from experiment or co-evolution analysis, and *ab initio* prediction. *Ab initio* methods typically work with taking fragment templates from various proteins, and assemble these into a model or decoy structure. Expected model accuracy declines from left to right: good accuracy is expected if based on homology; in contrast, *ab initio* modelling should only be considered if no other options remain.

Assessing the Quality of Structure Prediction Methods

Generally, as with any prediction problem, we can assess the quality of a prediction if we have a true answer to the question. Here, truth will be represented by an experimentally determined protein structure (of high quality). Fortunately, there are now (November 2017) over 120,000 deposited protein structures in the PDB (structures in the PDB: see “Relevant Websites section”). However, simply assessing how well a method performs over this set is problematic. The methods have been trained on structures contained in this data set, which implies there may be a strong bias in these methods to predict good models for sequences in their training dataset, and homologs of these sequences. In order to truly assess a method, a completely independent data set is required.

Critical Assessment of Protein Structure Prediction

Every other year CASP, a Critical Assessment of Protein Structure Prediction, provides such an independent validation benchmark. CASP is a blind test or competition: experimentalists provide sequences for which they know the structures will be solved imminently; modelling groups and servers try to predict the structure (Moult *et al.*, 1995). Once the structure is solved, the models can be evaluated using the solution structure of the target (see also Fig. 5).

CASP was started because the protein structure prediction problem was claimed to have been solved several times. The problem was, that algorithms were trained on databases that contained the structures that were evaluated in benchmarking tests. CASP overcomes this problem.

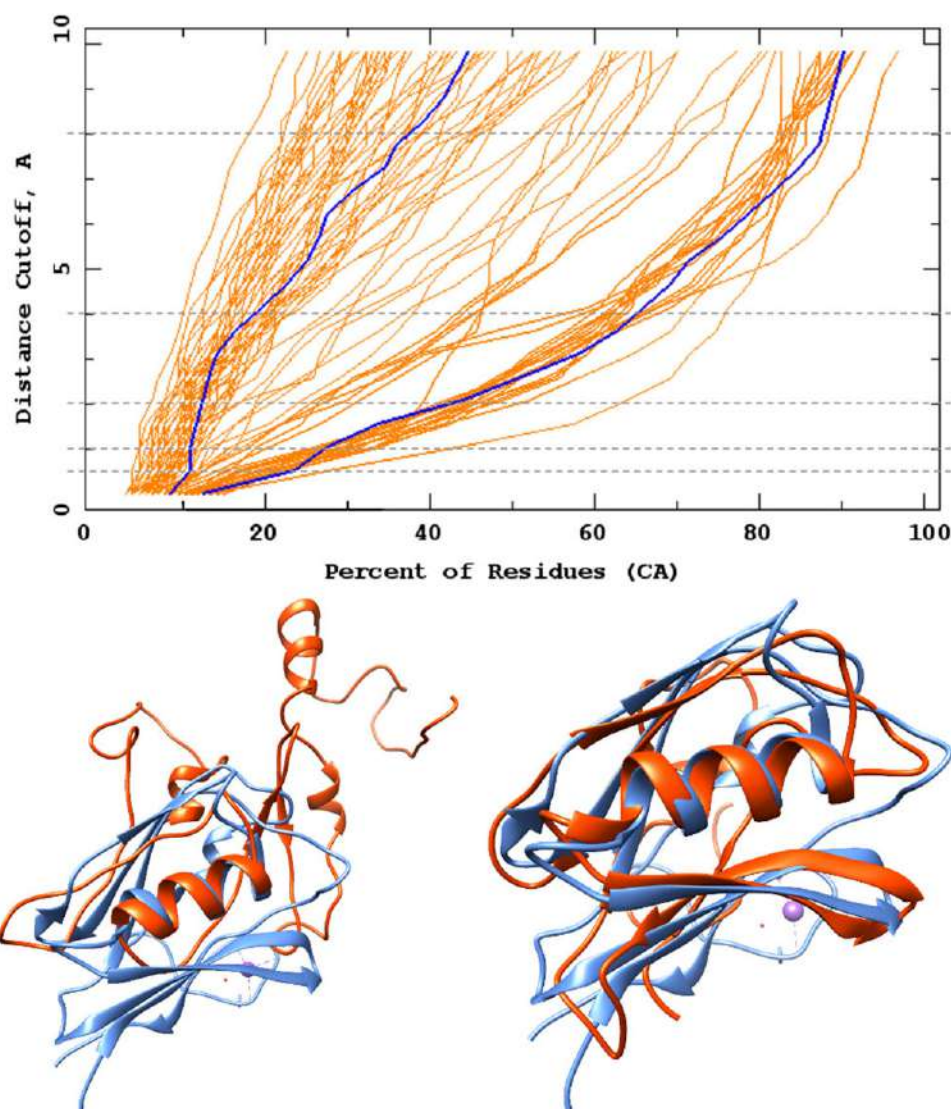


Fig. 5 Example of structural comparison for the target *T0886-D2* and two models submitted to CASP12. The top panel shows individual traces for all models generated for this target; the distance cutoff (vertical axis, in Å) is plotted against the fraction of residues (horizontal axis, in %) that can be aligned within this cutoff. The traces were obtained from predictioncenter.org/casp12. The dotted lines indicate the thresholds used in the GDT_TS (1, 2, 4, 8 Å) and GDT_HA (0.5, 1, 2, 4 Å) scores. Two models are highlighted in blue: a bad model (TS236; GDT_TS=18.90) on the left, and a good model (TS173; GDT_TS=51.97) on the right. Both model structures are also shown in the panels below in red, superposed onto the solution crystal structure in blue (PDB:5FHY). Structural superposition created using LGA at proteinmodel.org/AS2TS/LGA/. Reproduced from Zemla, A., 2003. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Research*, 31 (13), 3370–4. 3D visualisation using *Chimera 1.11.2*. Reproduced from Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF Chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25 (13), 1605–1612.

Note that the very first step in any practical structure prediction approach, should be to inspect the results from the latest CASP round (Moult *et al.*, 2016) via the CASP website (CASP website: see “Relevant Websites section”) to see what the state of the art methods are, and what their expected performance is.

Root-Mean-Square Deviation (RMSD)

If we want to assess the quality of a method, we need to measure the quality of the predictions made by the method. Hence, one would like to structurally compare atomic coordinates of the model and of the solution structure, and quantify the (dis)similarity.

The problem of comparing a model to a solution structure, is less difficult than the comparison between two homologous protein structures. This is because the alignment is trivial: the model has the same sequence as the solution structure; we know which residues, and atoms should correspond in the two structures.

The easiest way to compare structures, is to calculate the Root-Mean-Square Deviation (RMSD) after a structural superpositioning (Marti-Renom *et al.*, 2009). The superpositioning is required, because two arbitrary structures will typically not be positioned at coordinates suitable for comparison; first a translation and rotation needs to be applied to one of the two structures, to minimise the RMSD; the resulting RMSD after superpositioning can be used as a dissimilarity measure.

The Root-Mean-Square Deviation (RMSD) calculates the squared difference between two sets of atoms, and can be defined as follows:

$$\begin{aligned} \text{RMSD}(v, w) &= \sqrt{\frac{1}{n} \sum_{i=1}^n \|v_i - w_i\|^2} \\ &= \sqrt{\frac{1}{n} \sum_{i=1}^n (v_{ix} - w_{ix})^2 + (v_{iy} - w_{iy})^2 + (v_{iz} - w_{iz})^2} \end{aligned}$$

Here, v_i is the position vector of i^{th} atom of structure v ; w_i is the position vector of i^{th} atom of structure w ; and n is the total number of aligned atoms.

The RMSD takes the average over all aligned pairs. In protein structures typically one representative atom per residue is chosen, such as C α or C β .

GDT – Global Distance Test

If a model gets a loop very wrong, it tends to stick out and can be positioned very distant from the true structure, even though the remaining structure may be reasonably accurate. This partial outlier weighs heavily on the average distance calculated. Hence the RMSD is oversensitive to such outliers.

The global distance test total score (GDT_TS) is a more robust structural similarity measure that is well defined given an alignment between two structures. The key idea is to count the number of residues that can maximally be fitted within a certain distance cutoff, see also Fig. 5. The GDT score will therefore produce a percentage. In the formula below, the final score is the average over four different distance cutoffs (1, 2, 4, 8 Å).

$$\text{GDT_TS} = \frac{1}{4} \sum_{v=1,2,4,8\text{\AA}} \frac{G(v)}{t} \quad (1)$$

Here, $G(v)$ is the number of aligned residues within given RMSD cutoff v (in Ångstrom – 10^{-10}m) and t is the total number of aligned residues. A related score called GDT_HA was introduced in CASP some time ago (Read and Chavali, 2007) using stricter distance cutoffs (0.5, 1, 2, 4 Å), to cater for targets in the template-based modelling category where very high accuracies can be realized:

For a typical “difficult” CASP target no model even comes close to the experimentally solved structure; typical results would be similar to the left-most model in Fig. 5. If we have a look at the latest CASP results one will see that a performance of $\text{GDT_TS} < 20\%$ is not an exception. In other words, the protein structure prediction problem has NOT yet been solved, especially not if one considers targets without a good template structure.

How Difficult is it to Predict?

Overall, if one can find a good template, the quality of the predicted model will be relatively good. CASP results show that for homology modelling based on close homologs, it is possible to obtain models similar to the experimentally determined structure (Moult *et al.*, 2016). The modelled structure will typically have a good accuracy for the regions that can be well aligned between the target and template (using the sequences). The top two bars in Fig. 6 shows that one may expect the majority of such models to be accurate for $> 50\%$ of their residues. Gaps in an alignment will typically lie in loop regions of a structure and are more difficult to model. So, if we are interested in a large loop region that is not present in our template, we still may not be able to answer our scientific question with the resulting model structure (Moult *et al.*, 2016).

If no acceptable template can be found, the chances of successfully answering our scientific question will become very low. As a last resort, *ab initio* modelling can provide us with structural models. Typically, *ab initio* methods use very small templates from various proteins (see Fig. 4). The state of the art is that on average one may expect to find one structure that looks somewhat like the solution structure for the target among the top five or ten models (Moult *et al.*, 2016). However, be aware that the best model is typically not recognised as being the best through the scores of the prediction program. In Fig. 6 one sees that very clearly in the bottom few bars: without template, even with predicted contacts, one may have less than 20% of the structure correct in the majority of models; even in the best cases at most 40% of the residues are modelled accurately.

For Which Gene Sequences can we Predict a Three-Dimensional Structure?

If and only if there is a structure of a homologous protein present in the PDB, it is possible to generate a structural model of reasonable accuracy. Based on this notion, we can estimate for which (fraction of) gene sequences it is possible to predict a structure. This way it

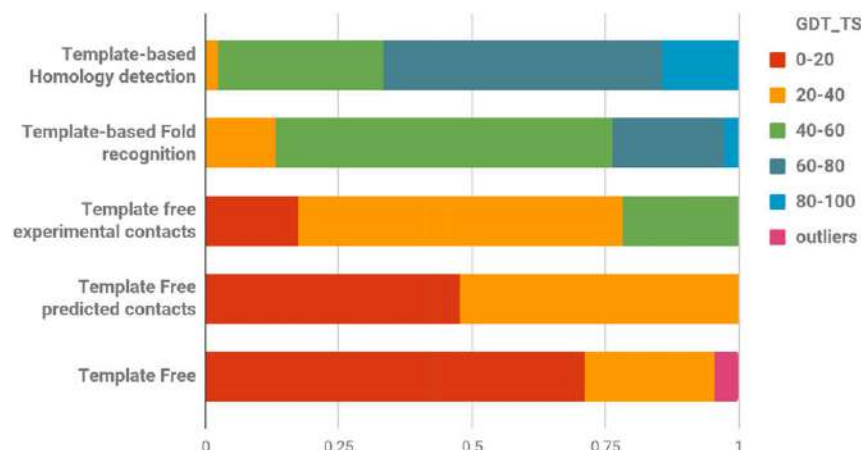


Fig. 6 Distribution of GDT_TS scores for the different model categories in CASP11 for template-based, template-free with contact information and template-free. The legend coloring corresponds to the GDT_TS scores, the bars indicate the fraction of models in each GDT_TS range for the six categories (GDT_TS scores for Modi *et al.* were estimated from the reported GDT_HA scores using their Fig. 4(A)). “Outliers” targets have unusually high GDT_TS due to being very short (~50 residue) with extended structures. Targets selected for server prediction (top bar) were considered easier than those for human prediction (second from top), average sequence identity was 26% vs. 20%, respectively. It is clear that overall prediction accuracy sharply declines going down this list of categories. For template-free modelling, the quality of contact information used is crucial. Experimental information (from chemical cross linking or simulated NMR) can give reasonable models. Predicted contacts do not guarantee that an acceptable model can be obtained, but without even predicted contacts, more than two-thirds of models are at most 20% correct. Reproduced from Modi, V., Xu, Q., Adhikari, S., Dunbrack, R.L., 2016. Assessment of template-based modeling of protein structure in CASP11. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S200–S220. Kinch, L.N., Li, W., Monastyrskyy, B., Kryshtafovych, A., Grishin, N.V., 2016a. Assessment of CASP11 contact-assisted predictions. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S164–S180. Kinch, L. N., Li, W., Monastyrskyy, B., Kryshtafovych, A., Grishin, N.V., 2016b. Evaluation of free modeling targets in CASP11 and ROLL. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S51–S66.

has been estimated that for a about 44% of residues in Eukaryotic gene sequences, we cannot yet make a homology model, and 15% of these residues lie within a gene for which we can not make a homology model for a single domain (Perdigão *et al.*, 2015). Especially membrane proteins are underrepresented in the PDB, due to the experimental difficulty of determining these structures. Note that these residues, may also lie in natively disordered regions (see also Section “Is there such a Concept as a Single Native Fold?”).

Similarly, it is possible to predict the range of protein structures present in an organism, based on the gene sequences identified in their completed genome. Gene prediction over a large number of organisms has revealed that there is a subset of protein structures that is present in nearly all organisms, for example TIM-barrels or Rossmann-folds (Abeln and Deane, 2005; Edwards *et al.*, 2013). Nevertheless, there is also a group of structures that is extremely lineage specific. It is to be expected that for this group of proteins many new structures remain to be discovered. This also implies that it will remain difficult to find suitable templates for homology modelling for these lineage-specific protein families.

How Accurate do we Need to be?

We already mentioned that we may approach the modelling of a protein structure of interest differently, depending on the biological question we want to ask, e.g., which residues are likely to be crucial for the functioning of the protein. Sometimes an answer to the research question may be possible in a simpler way, without full-scale prediction of the protein structure, e.g., by direct prediction of the impact of certain mutations or of protein-protein interaction sites. Examples of fully-automated webserver that do just that, are HOPE- (Venselaar *et al.*, 2010) and SeRenDIP (Hou *et al.*, 2017). In some cases, a rough homology model inspires the understanding of experimental results, spurring forward the project and eventually ending with crystal structures highlighting the protein function (in this case, protein-protein interactions) of interest (e.g., De Vries-van Leeuwen *et al.*, 2013). Also, specifically for enzymes, such as for example cytochromes P450, modelling of the protein structure should be done in combination with that of the ligand (de Graaf *et al.*, 2005).

In CASP11, three functional aspects, selected on being amenable to qualitative evaluation, were explicitly scored: multimeric state, (small) ligand binding, and mutation impact. Targets were selected that in solved crystal structure were dimeric, or had a ligand bound, or where from the crystallographers or in literature interest was expressed for evaluating mutants (Huwe *et al.*, 2016).

For prediction of dimer structures, only in two out of ten cases a dimer model with reasonable accuracy could be generated for the majority of monomer model structures (Huwe *et al.*, 2016). In the critical assessment of prediction of protein interaction (CAPRI), between 30%–80% of models were of ‘acceptable’ or ‘medium’ quality for easy dimer targets, while for harder targets (difficult dimers, multimers and heteromers), this fraction dropped to below 10% (Lensink *et al.*, 2016). Encouragingly, it was seen that also structure models of lower quality could sometimes lead to acceptable or even medium quality models of the bound proteins (Lensink *et al.*, 2016).

For ligand binding, it was found that the accuracy of even the best models (~ 2 Å) was not good enough for accurate ligand docking; the best ligands were around 5 Å RMSD (Huwe *et al.*, 2016). Something similar was found for mutation impact prediction; for most targets, model accuracy did not correlate with accuracy of impact prediction (Huwe *et al.*, 2016). Apparently, either homology models are not yet accurate enough for this purpose, or methods are tuned to particular characteristics of crystal structures.

Template Based Protein Structure Modelling

In the next two sections, we will in detail discuss first template-based and then template-free modelling. An overview of protein structure modelling, including both template-based and template-free modelling is given in Fig. 7; see also Fig. 3 for the terminology used.

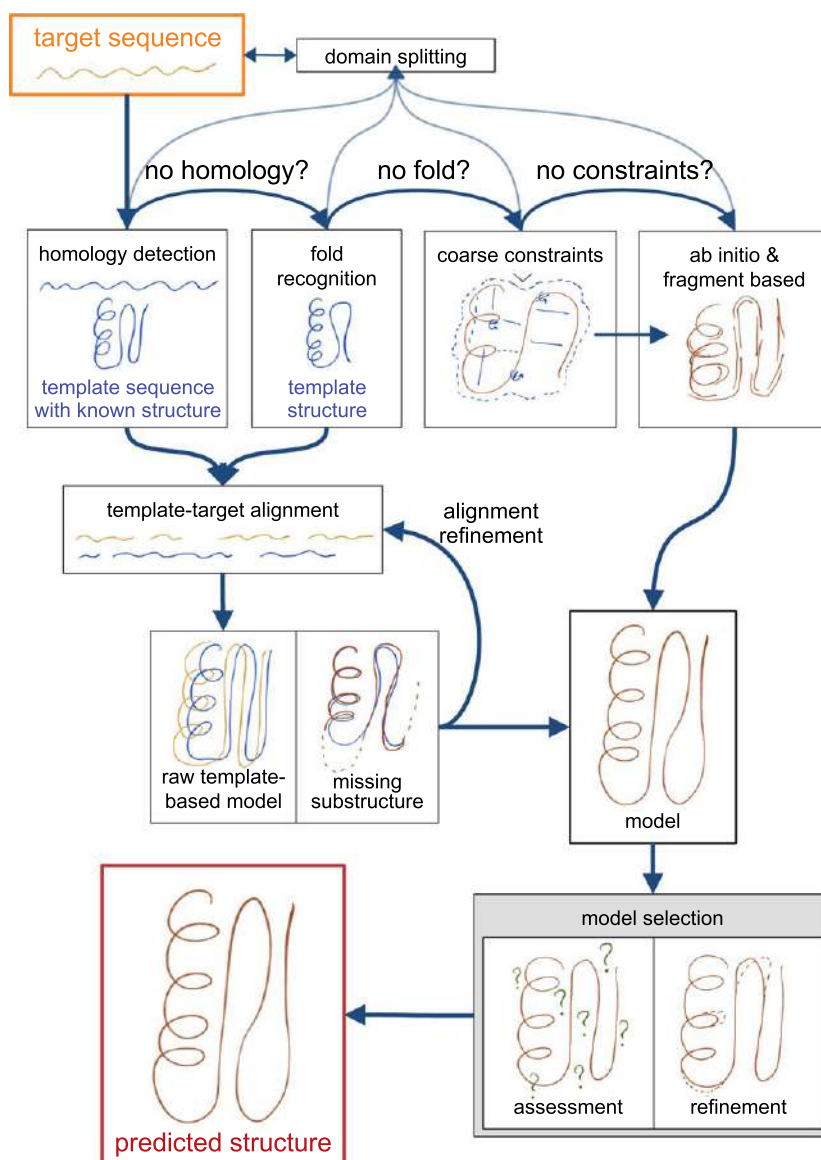


Fig. 7 Flowchart of protein three-dimensional structure prediction. It starts at the top left with the target protein sequence of interest, and ends with a predicted 3D structure at the bottom. Depending on the availability of a homologous template, a suitable fold, or coarse/experimental constraints, different options are available, with sharply decreasing expected model accuracy for each step. Homology-based models (far left) are most accurate, while ab-initio modelling (far right) is notoriously unreliable. Template-based (by homology or fold recognition) models require an alignment with the target sequence, from which the initial model will be built. Course constraints may sometimes be incorporated in this stage. The raw model may need to be completed by separate modelling of the missing substructures. Template-free modelling can benefit greatly if such constraints are available, if not the only sources left are fragment libraries and knowledge-based energy functions. The model, whether template-based or -free, is usually refined until a desired level of (estimated) quality is reached to produce the final predicted structure for our target protein sequence of interest.

Homology Based Template Finding

Homology modelling is a type of template based modelling with a template that is homologous to the target protein. As mentioned before, homology modelling works well, because structure is more conserved than sequence (Bajaj and Blundell, 1984; see Fig. 2). Typically, we will use a sequence-based homology detection method, such as BLAST, to search for homologous protein sequences in the full PDB dataset. If we find a sequence that has significant sequence similarity to the full length of our target sequence we have found a template. Of course it is possible that a template only covers part of the target sequence, see also Section “Domains” on domains above.

If a simple BLAST search against the PDB gives no good results we may need to start using alignment methods that can detect more distant homologs based on sequence comparison, such as PSI-BLAST or HMMs. PSI-BLAST uses hits from a previous iteration of BLAST to create a profile for our query sequence (Altschul *et al.*, 1997). This allows successive iterations to give more weight to very conserved residues, allowing to find more distant homologs. Even more sensitive homology detection tools typically consider sequence profiles of both the query and the template sequence, and try to align or score these profiles against each other (Petrokovski, 1996), with more recent implementations including Compass (Sadreyev and Grishin, 2003), HMMer (Finn *et al.*, 2011) and HHpred (Söding *et al.*, 2005). Profile-profile matching is known to increase detection of evolutionary sequence signals, leading to optimised alignment or sensitive sequence searching (Wang and Dunbrack, 2004). Note that to make full use of such methods it is important that the searching profiles are generated using extended sequence databases (so not just on the PDB). The evolutionary sequence profiles used in these methods can ensure that the most conserved, and therefore structurally most important, residues are more likely to be aligned accurately.

Once we have a good template, and an alignment of the target sequence with the template structure, we can build a structural model; this is called homology modelling in cases of clear homology between the target and the template.

Fold Recognition

If no obvious homologs with a known structure can be found in the PDB, it becomes substantially more difficult to predict the structure for a sequence. We may then use another trick to find a template: remember that the template does not only have a sequence, but also a structure. Therefore, we may be able to use structural information in this search. Fold recognition methods look explicitly for a plausible match between our query sequence and a *structure* from a database. Typically, such methods use structural information of the putative template to determine a match between the target sequence and the putative template sequence and structure.

Note that, in contrast to homology-based template search, it is not strictly necessary for a target sequence and a template sequence to be homologous – they may have obtained similar structures through convergent evolution. Therefore some fold recognition methods are trained and optimised for detection of structural similarity, rather than homology. In fact, for very remote homologs we may not be able to differentiate between convergent and divergent evolution. For the purpose of structure prediction this may not be an important distinction, nevertheless divergent evolution, i.e., homology through a shared common ancestor, invokes the principle of structure being more conserved, giving more confidence in the final model.

Threading is an example of a fold recognition method (Jones *et al.*, 1992). The query sequence is threaded through the proposed template structure. This means structural information of the template can be taken into account when score if the (threaded) alignment between the target and the potential template is a good fit. Typically threading works by scoring the pairs in the structure that make a contact, given the sequence composition. A sequence should also be aligned (threaded) onto the structure giving the best score. Scores are typically based on knowledge based potentials. For example, if two hydrophobic residues are in contact this would give a better score than a contact between a hydrophobic and polar amino acid. Note that threading does not necessarily search for homology. Threading remains a popular fold recognition methods, with several implementations available (Jones *et al.*, 1992; Zhang, 2008; Song *et al.*, 2013).

Another conceptually different fold recognition approach is to consider that amino-acid conservation rates may strongly differ between different structural environments. For example, one would expect residues on the surface to be less conserved, compared to those buried in the core. Similarly, residues in a α -helix or β -strand or typically more conserved than those in loop regions. In fact, the chance to form an insertion or deletion in a loop regions is seven times more likely than within (other) secondary structure elements. Since we know the structural environment of the residues for the potential template, we can use this to score an alignment between the target sequence and potential template sequence. FUGUE is a method that scores alignments using structural environment-specific substitution matrices and structure-dependent gap penalties (Shi *et al.*, 2001).

Generating the Target-Template Alignment

Once a suitable template has been found, one can start building a structural model. Typical model building methods will need the following inputs: (1) the target sequence, (2) the template structure, (3) sequence alignment between target and template and (4) any additional known constraints. The output will be a structural model, based on the constraints defined by the template structure and the sequence alignment.

Here, it is important to note that the methods that recognize good potential templates, are not necessarily the methods that will produce the most accurate alignments between the target and template sequence. As the final model will heavily depend upon the

alignment used, it is important to consider different methods, including for example structure and profile based methods, multiple sequence alignment programs or methods that can include structural information of the template in constructing the alignment. Examples of good sequence-based alignment methods, which can also exploit evolutionary signals and profiles, are *Praline* (Simossis *et al.*, 2005) and *T-coffee* (Notredame *et al.*, 2000). The T-coffee suite also includes the structure-aware alignment method *3d-coffee* (O'Sullivan *et al.*, 2004). Some aligners are context-aware, taking into account for example secondary structure (Simossis and Heringa, 2005) or trans-membrane (TM) regions explicitly (Pirovano *et al.*, 2008, 2010; Floden *et al.*, 2016), which are useful extension as TM proteins are severely underrepresented in the PDB. Of particular interest to the general user are automated template detection tools such as HHpred (Söding *et al.*, 2005). Bawono *et al.* (2017) give a good and up-to-date overview of multiple sequence alignment methodology, including profile-based and hidden-Markov-based methods.

Moreover, once a first model is created, it may be wise to interactively adapt the alignment, based on the resulting model, which might lead to an updated model. This procedure may also be carried out in an interactive fashion.

Generating a Model

Here we consider the *MODELLER* software to generate template based alignments (Sali and Blundell, 1993), which is one of several alternative approaches to construct homology models (Schwede *et al.*, 2003; Zhang, 2008; Song *et al.*, 2013). Firstly, the known template 3D structures should be aligned with the target sequence. Secondly, spatial features, such as C α -C α distances, hydrogen bonds, and mainchain and sidechain dihedral angles, are transferred from the template(s) to the target. *MODELLER* uses "knowledge based" constraints. The constraints are based on the template distances, the alignment, but also on knowledge- Based energy functions (probability distribution). The constraints are optimised using molecular dynamics with simulated annealing. Finally, a 3D model can be generated by satisfying all the restraints as well as possible.

Loop or Missing Substructure Modelling

We now have a model for all the residues that were aligned well between the template and the target. The remaining substructure(s), that are not covered by the template, will show as gaps in the alignment between target sequence and template. 'Loop modelling' is used to determine the structure for these missing parts. Loop models are typically based on fragment libraries, knowledge based potentials and constraints from the aligned structure. This problem is in fact closely related to the template-free modelling procedure, as we need to generate a structure without a readily available template (template-free approaches are discussed in more detail in the next section). In CASP11, consistent refinement overall as well as for loop regions was achieved; the limiting factor for effective refinement was concluded to be the energie functions used, in particular missing physicochemical effects and balance of energy terms (Lee *et al.*, 2016).

Template-Free Protein Structure Modelling

What if no Suitable Template Exists?

If on the other hand, no suitable template is available for our target protein of interest, we will need to follow a 'template-free' modelling strategy. Without a direct suitable template, we need an *ab initio* strategy that can suggest possible structural models based on the sequence of the template alone. In this case, we need to resort to fragment based approaches. Here small, suitable fragments, from various PDB structure, are assembled to generate possible structural models. As the fragments are typically matched using sequence similarity, one may even consider this as template based modelling at a smaller scale. However, since the sequence match is based on a limited number of residues, this would not generally imply a homologous relation between the fragment template and the target sequence. It is also important to note that this type of *ab initio* modelling is still "knowledge based": the structural models are generated from small substructures present in the PDB, assessed by energy scoring functions generated by mining the PDB. In other words, these model are not based on physical principles and physico-chemical properties alone. This also means, that such models are likely to share any of the biases that are present in the PDB, such as lack of trans-membrane proteins and absence of disordered regions.

Generating Models From Structural Fragments

Here we follow the keys ideas in the *Rosetta* based *ab initio* modelling suite (Simons *et al.*, 1999). Quark, another fragment-based approach provides a similar performing alternative (Xu and Zhang, 2012). The overall approach is to split the target sequence into 3 and 9-residue overlapping sequence fragments, i.e., sliding windows, and find matching structural fragments from PDB.

A fragment library is generated by taking 3 and 9-residue fragments from the PDB and clustering these together into groups of similar structure. For each fragment in the database sequence profiles are created. These profiles subsequently are used to search for suitable fragments for our query/target sequence, as we only have sequence information (see Fig. 8 on the left-hand side).

The target sequence also will be split into 3 and 9-residue overlapping sequence fragments. These target sequence fragments are then matched with the structural PDB based fragment library using profile-profile matching. Note that this procedure will generate multiple fragments for each fragment window in the sequence. The fragment windows on the sequence typically also overlap (see Fig. 8 middle panel).

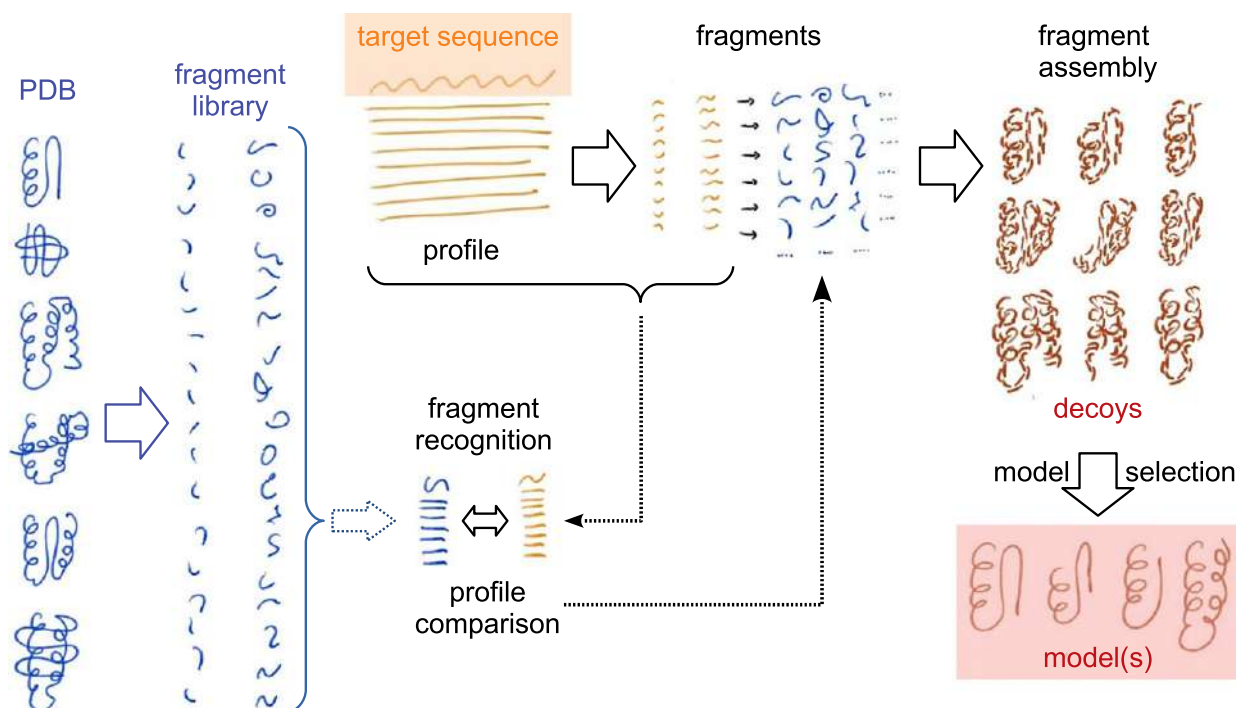


Fig. 8 Overview of the fragment-based modelling strategy. A library of structure fragments was created once from the PDB; all small 3-residues and larger 9-residue fragments are collected and clustered. A target sequence of interest is also separated into 3- and 9-residue sequence fragments. For each of these, a profile-profile search is performed to find matching fragments from the fragment library; typically for each target fragment, multiple hits with different structure are retrieved. This collection of fragments of alternate structure are then assembled through a Monte Carlo algorithm into a large set of possible structures, called 'decoys'. Using knowledge-based potentials and overall statistics, from the decoy set, a final selection of model structures is made.

Fragment Assembly Into Decoys

The above scenario leads to many possible structural fragments to cover a sequence position. To find the best structural model is a combinatorial problem in terms of fragment combinations, see Fig. 8 top right. A Monte Carlo algorithm is used to search through different fragment combinations. Good combinations are those that give a low energy. Each MC run will produce a different model, since it is a stochastic algorithm. Note that to be able to optimise a model, we need a scoring function. A knowledge based energy function is used, including the number of neighbours, given amino acid type, residue pair interactions, backbone hydrogen bonding, strand arrangement, helix packing, radius of gyration, Van der Waals repulsion. These are all terms that are relatively cheap to compute when a new combination of fragments is tried. The structural model is optimised by slowly lowering the MC temperature – this is also called simulated annealing. The generated models are called 'decoys'.

Thereafter, decoys are refined using additional Monte Carlo cycles, and a more fine-grained energy function including: backbone torsion angles, Lennard-Jones interactions, main chain and side-chain hydrogen bonding, solvation energy, rotamers and a comparison to unfolded state.

Finally, the most difficult task is to select, from all the refined decoys, a structure that is a suitable model for the target sequence, see Fig. 8 bottom right. Again, using a more detailed, knowledge-based energy function, decoys can be scored to assess how 'protein-like' they are. Such a selection procedure may get rid of very wrong models. However, selecting the best model, without any additional information (from for example experiments or co-evolution-based contact-prediction), is likely to lead to poor results, as is shown in Fig. 6.

Constraints From Co-Evolution Based Contact Prediction or Experiments

As already mentioned, valuable additions to the modelling process are coarse constraints from experimental data or contact prediction. Experimental data from NMR and chemical cross-linking can yield distance restraints that are particularly useful in the template-free modelling to narrow down the conformational space to be searched; still average accuracy of models produced remains extremely limited, (see Fig. 6; Kinch *et al.*, 2016a). Other sources of information are contours or surfaces that can be obtained from cryo-EM, or small angle scattering experiments, either with electrons, neutrons, or x-ray radiation. However, since these techniques are employed mostly for elucidating larger macromolecular complexes, they are considered out of scope for the current article.

Of more general applicability may be methods for predicting intra-protein residue contacts; the main approach currently is based on some form of co-evolution information obtained by direct-coupling methods from 'deep' alignments (Marks *et al.*, 2011; Jones *et al.*, 2012; Morcos *et al.*, 2011). Depth here signifies the amount of sequence variation present in the alignment in relation

to the length of the protein (the longer the protein, the more variation is needed). [Ovchinnikov et al. \(2017\)](#) expresses this as the *effective protein length*: $Nf = N80\%ID/\sqrt{l}$ where l is the protein length, and $N80\%ID$ the number of cluster at 80% sequence identity. They showed that Nf can be greatly enhanced by the use of metagenomic sequencing data, and that this leads to a marked improvement in model quality, and estimate that this would triple the number of protein families for which the correct fold might be predicted ([Ovchinnikov et al., 2017](#)). [Wuyun et al. \(2016\)](#) investigated ‘consensus’-based methods, which combine both direct-coupling and machine-learning approaches, and find that the machine-learning methods are less sensitive to alignment depth and target difficulty, which are crucial factors for success for the direct-coupling methods.

Selecting and Refining Models From Structure Prediction

Once we have created (several) models, we need to assess which model is the best one. Typically this can be done by scoring models on several properties using model quality assessment programs and visual inspection with respect to “protein like” features. Moreover, if any additional knowledge about the structure or function of our target protein is available, this may also help to assess the quality of the model(s). In addition, one may in some cases want to improve a model, or parts of it; this is called model refinement.

Model Refinement

For many years in CASP, model refinement was a no-go area; the rule of thumb was: build our homology model and do not touch it! An impressive example of the failure of refinement methods was shown by the David Shaw group, who concluded that “simulations initiated from homology models drift away from the native structure” ([Raval et al., 2012](#)). Since CASP10 in 2014 ([Nugent et al., 2014](#)) and continuing in the latest CASP11, there is reason for moderate optimism. General refinement strategies report small but significant improvements of 3%–5% over 70% of models ([Modi and Dunbrack, 2016](#)). Interestingly, and in stark contrast to the earlier results by [Raval et al. \(2012\)](#), the average improvement of GDT_HA using simulation-based refinement now also is about 3.8, with an improvement (more than 0.5) for 26 models ([Feig and Mirjalili, 2016](#)). For five models, the scores became worse (by more than 0.5), and another five showed no significant change. Particularly, for very good initial models (GDT_HA > 65), models were made worse. Moreover, they also convincingly showed that both more and longer simulations consistently improved these results; note however that protocol details such as using C α restraints, are thought to be the limiting factor ([Feig and Mirjalili, 2016](#)), as already used previously (e.g., [Keizers et al., 2005](#); [Feenstra et al., 2006](#)), and replicated by others (e.g., [Cheng et al., 2017](#)). Most successful refinement appears to come from correctly placing β -sheet or coil regions at the termini ([Modi and Dunbrack, 2016](#)).

Model Quality Assessment Strategies

It may be generally helpful to compare models generated by different prediction methods; if models from different methods look alike (more precisely if the pair has a low RMSD and/or high GDT_TS) they are more likely to be correct. Similarly, templates found by different template finding strategies, found for example both by homology sequence searches and fold recognition methods are more likely to yield good modelling results ([Moult et al., 2016](#); [Kryshtafovych et al., 2016](#)). Such a consensus template is generally more reliable than the predictions from individual methods – especially if the individual scores are barely significant. Lastly, one can consider biological context to select good models.

Whether a model is built using homology modelling, fold recognition and modelling or *ab initio* prediction, all models can be given to a Model Quality Assessment Program (MQAP) for model validation. A validation program provides a score predicting how reliable the model is. These scores typically take into account to what extent a model resembles a “true” protein structure. The best performing validation programs take a large set of predicted models, and indicate which out of these is expected to be the most reliable.

Validation scoring may be based on similar ideas as validation for experimental structures or may be specific to structure prediction. For example, it can be checked if the amount of secondary structure, e.g., helix and strand vs. loop, has a similar ratio as in known protein structures; if a model for a sequence of 200 amino acids does not contain a single helix or β -strand, the model does not resemble true protein structures, and is therefore very unlikely to be the true structural solution for the sequence. A similar type of check may be done for the amount of buried hydrophobic groups and globularity of the protein.

Different models may also be compared to each other. One trick that is commonly used, is that if multiple prediction methods create structurally similar models, these models are more likely to be correct. Hence, a good prediction strategy is to use several prediction methods, and pick out the most consistent solution. A pitfall here is that if all models are based on the same, or very similar, templates, they will look similar but this may not indicate the likelihood of them being correct.

Secondary Structure Prediction

Secondary structure prediction is relatively accurate, and is in fact much easier to solve than three-dimensional structure prediction, see, e.g., the review by [Pirovano and Heringa \(2010\)](#). The accuracy of assigning strand, helix or loops to a certain residue can go up

to 80% with the most reliable methods. Typically such methods use (hydrophobic) periodicity in the sequence combined with phi and psi angle preferences of certain amino acid types to come to accurate predictions. The real challenge lies in assembling the secondary structure element in a correct topology. Nevertheless, secondary structure prediction may be used to assess the quality of a model built with a (tertiary) structure prediction method. Many (automated) methods also incorporate secondary structure information during (Simossis and Heringa, 2005), homology detection (Söding *et al.*, 2005; Shi *et al.*, 2001) and contact prediction (Terashi and Kihara, 2017; Wang *et al.*, 2017).

Is There Such a Concept as a Single Native Fold?

Before we conclude, we should consider a more physical description of protein structure. In fact, protein folding from a physical point of view is a very interesting process: given a sequence, a protein tends to fold always, and exactly into the same functional structure. In material design, it is extremely difficult to mimic such high specificity. The apparent observation of folding specificity also leads to the question, is there such a concept as a single native fold? Or, more pragmatically, is sequence-to-structure truly a one-to-one relation?

In fact, if one wants to start making quantitative predictions, such as the stability of a protein fold, or the binding strength between two proteins in terms of free energy, it is much more helpful to think in ensembles of structural configurations for a protein sequence (e.g., May *et al.*, 2014; Pucci *et al.*, 2017). The probability to find a protein in a specific ensemble of structural configurations will depend on conditions such as the presence or absence of binding partners, the pressure, the pH or the temperature (e.g., van Dijk *et al.*, 2015, 2016). There are a few specific cases, common cases, for which even the functional or biologically relevant structural ensembles do not resemble a single globular folded structure.

Disordered Proteins

Not all proteins fold into single configurations, some proteins stay natively unfolded, i.e., they can take up a large variety of more extended, and very different configurations (Uversky *et al.*, 2000; Mészáros *et al.*, 2007). Some disordered regions contain elements that do form stable structures upon binding. The regions that remain disordered are thought to be important to prevent aggregation within the cell (Abeln and Frenkel, 2008). Missing residues in X-ray structures are typically removed for crystallization; for this reason disorder prediction methods have been developed. Disordered regions are relatively easy to predict in protein sequences just like secondary structures; broadly speaking, prediction can be based on the large amount of charged/polar (hydrophilic) amino acids in combination with the presence of amino acids that disrupt the secondary structure (proline and glycine) in these regions (Oldfield *et al.*, 2005; Wang *et al.*, 2016). We know sequences of many proteins contain large disordered segments (33% of eukaryotic, 2% archaeal, and 4% bacterial proteins; Ward *et al.* (2004)).

Allostery and Functional Structural Ensembles

It is important to realize that one protein, typically, does not correspond to one defined three-dimensional structure. Disordered regions or proteins are one particularly salient case, but also proteins which fold into specific three-dimensional configurations, may exist in multiple functional states each with a specific structure. The biological question of interest dictates which state is relevant. Most proteins have only been crystallized in one particular state, and often it is not known to which biological condition this crystal structure may correspond. One may have cases where a homology model of the relevant state may be preferred over a crystal structure of a different or unknown state (e.g., de Graaf *et al.*, 2005).

Amyloid Fibrils

Lastly, we should consider a competing state of folded proteins: the aggregated state, where multiple peptide chains clog together in fibrillar structures or amorphous aggregates. Amyloid fibres are formed by β -strands formed between different protein or peptide (small protein) chains. Fibril formation is associated with various neurodegenerative diseases, such as Alzheimer's, Creutzfeldt-Jakob and Parkinson's (Chiti and Dobson, 2006). In fact, the fibrillar state is more favorable than the state of separately folded structures for several protein types. The general cellular toxicity of such aggregates, puts evolutionary pressure on avoiding structural characteristics on the surface of proteins; hence it is extremely rare to observe solvent accessible β -strand edges or large hydrophobic surface patches (Richardson and Richardson, 2002; Abeln and Frenkel, 2011). The propensity proteins have to form Amyloid fibrils is relatively easy to predict (Graña-Montes *et al.*, 2017). However, reference databases are still small so it is difficult to verify such methods.

Acknowledgements

We thank Nicola Bonzanni, Kamil K. Belau, Ashley Gallagher and Jochem Bijlard for insightful discussions and critical proof-reading of early versions.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Assessment of Structure Quality (RNA and Protein). Biomolecular Structures: Prediction, Identification and Analyses. Cloud-Based Molecular Modeling Systems. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Population-Based Sampling and Fragment-Based De Novo Protein Structure Prediction. Protein Structural Bioinformatics: An Overview. Protein Structure Analysis and Validation. Protein Structure Classification. Protein Structure Databases. Protocol for Protein Structure Modelling. Rational Structure-Based Drug Design. Secondary Structure Prediction. Small Molecule Drug Design. Structural Genomics. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow. Transmembrane Domain Prediction

References

- Abeln, S., Deane, C.M., 2005. Fold usage on genomes and protein fold evolution. *Proteins* 60 (4), 690–700.
- Abeln, S., Frenkel, D., 2008. Disordered flanks prevent peptide aggregation. *PLOS Computational Biology* 4 (12), e1000241.
- Abeln, S., Frenkel, D., 2011. Accounting for protein-solvent contacts facilitates design of nonaggregating lattice proteins. *Biophysical Journal* 100 (3), 693–700.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25 (17), 3389–3402.
- Bajaj, M., Blundell, T., 1984. Evolution and the tertiary structure of proteins. *Annual Review of Biophysics and Bioengineering* 13 (1), 453–492.
- Bawono, P., Dijkstra, M., Pirovano, W., *et al.*, 2017. Multiple Sequence Alignment. In: *Methods in Molecular Biology – Bioinformatics – Volume I: Data, Sequence Analysis, and Evolution*. New York: Humana Press, pp. 167–189.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28 (1), 235–242.
- Cheng, Q., Joung, I., Lee, J., 2017. A simple and efficient protein structure refinement method. *Journal of Chemical Theory and Computation* 13 (10), 5146–5162.
- Chiti, F., Dobson, C.M., 2006. Protein misfolding, functional amyloid, and human disease. *Annual Review of Biochemistry* 75, 333–366.
- de Graaf, C., Vermeulen, N.P.E., Feenstra, K.A., 2005. Cytochrome P450 in silico: An integrative modeling approach. *Journal of Medicinal Chemistry* 48 (8), 2725–2755.
- De Vries-van Leeuwen, I.J., da Costa Pereira, D., Flach, K.D., *et al.*, 2013. Interaction of 14-3-3 proteins with the estrogen receptor alpha F domain provides a drug target interface. *Proceedings of the National Academy of Sciences of the United States of America* 110 (22), 8894–8899.
- Edwards, H., Abeln, S., Deane, C.M., 2013. Exploring fold space preferences of new-born and ancient protein superfamilies. *PLOS Computational Biology* 9 (11), e1003325.
- Feenstra, K.A., Hofstetter, K., Bosch, R., *et al.*, 2006. Enantioselective substrate binding in a monooxygenase protein model by molecular dynamics and docking. *Biophysical Journal* 91 (9), 3206–3216.
- Feig, M., Mirjalili, V., 2016. Protein structure refinement via molecular-dynamics simulations: What works and what does not? *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S282–S292.
- Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Research* 39 (Suppl. 2), W29–W37.
- Floden, E.W., Tommaso, P.D., Chatzou, M., *et al.*, 2016. PSI/TM-Coffee: A web server for fast and accurate multiple sequence alignments of regular and transmembrane proteins using homology extension on reduced databases. *Nucleic Acids Research* 44 (W1), W339–W343.
- Grafia-Montes, R., Pujols-Pujol, J., Gómez-Picanyol, C., Ventura, S., 2017. Prediction of protein aggregation and amyloid formation. In: *From Protein Structure to Function with Bioinformatics*. Dordrecht: Springer, pp. 205–263.
- Hou, Q., De Geest, P., Vranken, W., Heringa, J., Feenstra, K., 2017. Seeing the trees through the forest: Sequence-based homo- and heteromeric protein-protein interaction sites prediction using random forest. *Bioinformatics* 33 (10), 1–10.
- Huwe, P.J., Xu, Q., Shapovalov, M.V., *et al.*, 2016. Biological function derived from predicted structures in CASP11. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), 370–391.
- Jones, D.T., Buchan, D.W.A., Cozzetto, D., Pontil, M., 2012. PSICOV: Precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* 28 (2), 184–190.
- Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. A new approach to protein fold recognition. *Nature* 358 (6381), 86–89.
- Keizers, P.H.J., de Graaf, C., de Kanter, F.J.J., *et al.*, 2005. Metabolic Regio- and Stereoselectivity of Cytochrome P450 2D6 towards 3,4-Methylenedioxy-N-alkylamphetamines: In silico predictions and experimental validation. *Journal of Medicinal Chemistry* 48 (19), 6117–6127.
- Kinch, L.N., Li, W., Monastyrskyy, B., Kryshchuk, A., Grishin, N.V., 2016a. Assessment of CASP11 contact-assisted predictions. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S164–S180.
- Kryshchuk, A., Barbato, A., Monastyrskyy, B., *et al.*, 2016. Methods of model accuracy estimation can help selecting the best models from decoy sets: Assessment of model accuracy estimations in CASP11. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S349–S369.
- Lee, G.R., Heo, L., Seok, C., 2016. Effective protein model structure refinement by loop modeling and overall relaxation. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S293–S301.
- Lensink, M.F., Velankar, S., Kryshchuk, A., *et al.*, 2016. Prediction of homoprotein and heteroprotein complexes by protein docking and template-based modeling: A CASP-CAPRI experiment. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S323–S348.
- Marks, D.S., Colwell, L.J., Sheridan, R., *et al.*, 2011. Protein 3D structure computed from evolutionary sequence variation. *PLOS ONE* 6 (12), e28766.
- Marti-Renom, M.A., Capriotti, E., Shindyalov, I.N., Bourne, P.E., 2009. Structure comparison and alignment. In: Gu, J., Bourne, P.E. (Eds.), *Structural Bioinformatics*, second ed. John Wiley & Sons, Inc., pp. 397–418.
- May, A., Pool, R., van Dijk, E., *et al.*, 2014. Coarse-grained versus atomistic simulations: Realistic interaction free energies for real proteins. *Bioinformatics* 30 (3), 326–334.
- Mészáros, B., Tompa, P., Simon, I., Dosztányi, Z., 2007. Molecular principles of the interactions of disordered proteins. *Journal of Molecular Biology* 372 (2), 549–561.
- Modi, V., Dunbrack, R.L., 2016. Assessment of refinement of template-based models in CASP11. *Proteins: Structure, Function, and Bioinformatics* 84 (S1), S260–S281.
- Morcos, F., Pagnani, A., Lunt, B., *et al.*, 2011. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America* 108 (49), E1293–E1301.
- Moult, J., Fidelis, K., Kryshchuk, A., Schwede, T., Tramontano, A., 2016. Critical assessment of methods of protein structure prediction: Progress and new directions in round XI. *Proteins: Structure, Function and Bioinformatics* 84 (S1), S4–S14.
- Moult, J., Pedersen, J.T., Judson, R., Fidelis, K., 1995. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Genetics* 23 (3), ii–iv.
- Notredame, C., Higgins, D.G., Heringa, J., 2000. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302 (1), 205–217.
- Nugent, T., Cozzetto, D., Jones, D.T., 2014. Evaluation of predictions in the CASP10 model refinement category. *Proteins: Structure, Function, and Bioinformatics* 82 (S2), S98–S111.

- O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D.G., Notredame, C., 2004. 3DCoffee: Combining protein sequences and structures within multiple sequence alignments. *Journal of Molecular Biology* 340 (2), 385–395.
- Oldfield, C.J., Cheng, Y., Cortese, M.S., *et al.*, 2005. Comparing and combining predictors of mostly disordered proteins. *Biochemistry* 44 (6), 1989–2000.
- Ovchinnikov, S., Park, H., Varghese, N., *et al.*, 2017. Protein structure determination using metagenome sequence data. *Science* 355 (6322), 294–298.
- Perdigão, N., Heinrich, J., Stolte, C., *et al.*, 2015. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences* 112 (52), 15898–15903.
- Petrokovski, S., 1996. Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Research* 24 (19), 3836–3845.
- Pirovano, W., Abeln, S., Feenstra, K. A., Heringa, J., 2010. Multiple alignment of transmembrane protein sequences. In: *Structural Bioinformatics of Membrane Proteins*, Vienna: Springer, pp. 103–122.
- Pirovano, W., Feenstra, K.A., Heringa, J., 2008. PRALINETM: A strategy for improved multiple alignment of transmembrane proteins. *Bioinformatics* 24 (4), 492–497.
- Pirovano, W., Heringa, J., 2010. Protein secondary structure prediction. In: Carugo, O., Eisenhaber, F. (Eds.), *Data Mining Techniques for the Life Sciences*. Humana Press, pp. 327–348.
- Pucci, F., Kwasigroch, J.M., Rooman, M., 2017. SCooP: An accurate and fast predictor of protein stability curves as a function of temperature. *Bioinformatics*.
- Raval, A., Piana, S., Eastwood, M.P., Dror, R.O., Shaw, D.E., 2012. Refinement of protein structure homology models via long, all-atom molecular dynamics simulations. *Proteins: Structure, Function, and Bioinformatics* 80 (8), 2071–2079.
- Read, R.J., Chavali, G., 2007. Assessment of CASP7 predictions in the high accuracy template-based modeling category. *Proteins: Structure, Function, and Bioinformatics* 69 (S8), S27–S37.
- Richardson, J.S., Richardson, D.C., 2002. Natural beta-sheet proteins use negative design to avoid edge-to-edge aggregation. *Proceedings of the National Academy of Sciences of the United States of America* 99 (5), 2754–2759.
- Sadreyev, R., Grishin, N., 2003. COMPASS: A tool for comparison of multiple protein alignments with assessment of statistical significance. *Journal of Molecular Biology* 326 (1), 317–336.
- Sali, A., Blundell, T.L., 1993. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology* 234 (3), 779–815.
- Schwede, T., Kopp, J., Guex, N., Peitsch, M.C., 2003. SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Research* 31 (13), 3381–3385.
- Shi, J., Blundell, T.L., Mizuguchi, K., 2001. FUGUE: Sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal Molecular Biology* 310 (1), 243–257.
- Simons, K.T., Bonneau, R., Ruczinski, I., Baker, D., 1999. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins. Suppl.* 3, S171–S176.
- Simossis, V.A., Heringa, J., 2005. PRALINE: A multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic Acids Research* 33 (Web Server), W289–W294.
- Simossis, V.A., Kleinjung, J., Heringa, J., 2005. Homology-extended sequence alignment. *Nucleic Acids Research* 33 (3), 816–824.
- Söding, J., Biegert, A., Lupas, A.N., 2005. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Research* 33 (Web Server issue), W244–W248.
- Song, Y., Dimaio, F., Wang, R.Y.R., *et al.*, 2013. High-resolution comparative modeling with RosettaCM. *Structure* 21 (10), 1735–1742.
- Terashi, G., Kihara, D., 2017. Protein structure model refinement in CASP12 using short and long molecular dynamics simulations in implicit solvent. *Proteins: Structure, Function and Bioinformatics* 86 (Suppl. 1), S189–S201.
- Uversky, V.N., Gillespie, J.R., Fink, A.L., 2000. Why are “natively unfolded” proteins unstructured under physiologic conditions? *Proteins* 41 (3), 415–427.
- van Dijk, E., Hoogveen, A., Abeln, S., 2015. The hydrophobic temperature dependence of amino acids directly calculated from protein structures. *PLOS Computational Biology* 11 (5), e1004277.
- van Dijk, E., Varilly, P., Knowles, T.P.J., Frenkel, D., Abeln, S., 2016. Consistent treatment of hydrophobicity in protein lattice models accounts for cold denaturation. *Physical Review Letters* 116 (7), 078101.
- Venselaar, H., te Beek, T.A., Kuipers, R.K., Hekkelman, M.L., Vriend, G., 2010. Protein structure analysis of mutations causing inheritable diseases. An e-Science approach with life scientist friendly interfaces. *BMC Bioinformatics* 11 (1), 548.
- Wang, G., Dunbrack, R.L., 2004. Scoring profile-to-profile sequence alignments. *Protein Sciences* 13 (6), 1612–1626.
- Wang, S., Ma, J., Xu, J., 2016. AUCpred: Proteome-level protein disorder prediction by AUC-maximized deep convolutional neural fields. *Bioinformatics* 32 (17), i672–i679.
- Wang, S., Sun, S., Xu, J., 2017. Analysis of deep learning methods for blind protein contact prediction in CASP12. *Proteins* 86 (Suppl.1), S67–S77.
- Ward, J.J., Sodhi, J.S., McGuffin, L.J., *et al.*, 2004. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *Journal of Molecular Biology* 337, 635645.
- Wuyun, Q., Zheng, W., Peng, Z., Yang, J., 2016. A large-scale comparative assessment of methods for residue-residue contact prediction. *Briefings in Bioinformatics*. bbw106.
- Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function and Bioinformatics* 80 (7), 1715–1735.
- Zhang, Y., 2008. I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics* 9 (1), 40.

Relevant Websites

<http://predictioncenter.org/>
 Prediction Center.
<https://www.rcsb.org/pdb/statistics/holdings.do>
 RCSB PDB - Holdings Report.
<https://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total&seqid=100>
 Yearly Growth of Total Structures.

Protein Structure Analysis and Validation

Tsuyoshi Shirai, Nagahama Institute of Bio-Science and Technology, Nagahama, Japan

© 2019 Elsevier Inc. All rights reserved.

Glossary

Clashscore Representative score for validating the quality of a molecular structural model, that is, the number of too-close atom–atom contacts per 1000 atoms in the model.

Density map Electron density maps and EM maps are maps of a material's density; they are direct outputs of X-ray crystallography and electron microscopy, respectively, and models of molecular structures are fitted into them.

Distance constraint Nuclear magnetic resonance spectroscopy generates a list of atom–atom distances in a molecule, which are used to determine the molecule's conformation.

Electron microscopy Method to determine a structure by observing molecular images with an electron microscope and projecting a density map of the molecules.

Nuclear magnetic resonance spectroscopy Method to determine a structure by irradiating radioisotope-labeled samples with microwaves and evaluating interatom distances and bond angles constraints.

Ramachandran plot Two-dimensional plot for validating a protein main-chain conformation in which main-chain torsion ϕ angles are plotted against ψ angles, and the plot area is divided into allowed and disallowed regions.

RMSZ score Representative score for validating the quality of a molecular structural model by the Z-score deviations of atom bond lengths and angles of models from canonical values of reference structures.

X-ray crystallography Method to determine a structure by measuring X-ray diffractions from crystals and synthesizing an electron density map of molecules.

Introduction

Proteins are the major working biological molecules in the cell. Proteins fold into three-dimensional (3D or tertiary) structures under natural conditions. Knowing the 3D structures is required to understand the chemical and physical bases of their functions. It is also important that most biological molecules work in collaboration by forming complexes (supramolecules) of subunits including proteins, nucleic acids, carbohydrates, lipids, and small molecules. Recently, analysis of supramolecules is increasingly important in structural biology, and many innovative improvements have been made in the methods of structural analyses and validations for application to larger supramolecules.

Experimental Methods to Determine Structure

There are three major experimental methods to determine the structure of biological molecules: X-ray crystallography (XRC or often referred to as XP, which stands for protein crystallography) (Drenth and Mesters, 2007), nuclear magnetic resonance spectroscopy (NMR) (Wuthrich, 1986), and electron microscopy (EM) (Frank, 2006). Each method has both advantages and disadvantages (Table 1). XP and NMR have been used so far mainly to determine structures at atomic resolution. XP has a much higher resolution and limit of molecular weight than NMR, and has been the most utilized method. However, XP first requires preparation of crystals of molecules, which is often difficult. In contrast, NMR does not need crystals, and structures in solution, which are probably closer to their native structures, can be determined. EM is, to date, lower in resolution in comparison to the other methods, but it has a very high molecular weight limit and can be used to determine supramolecular structures. Thus, these methods are often applied collaboratively, and such approaches are called integrative/correlative structure analyses.

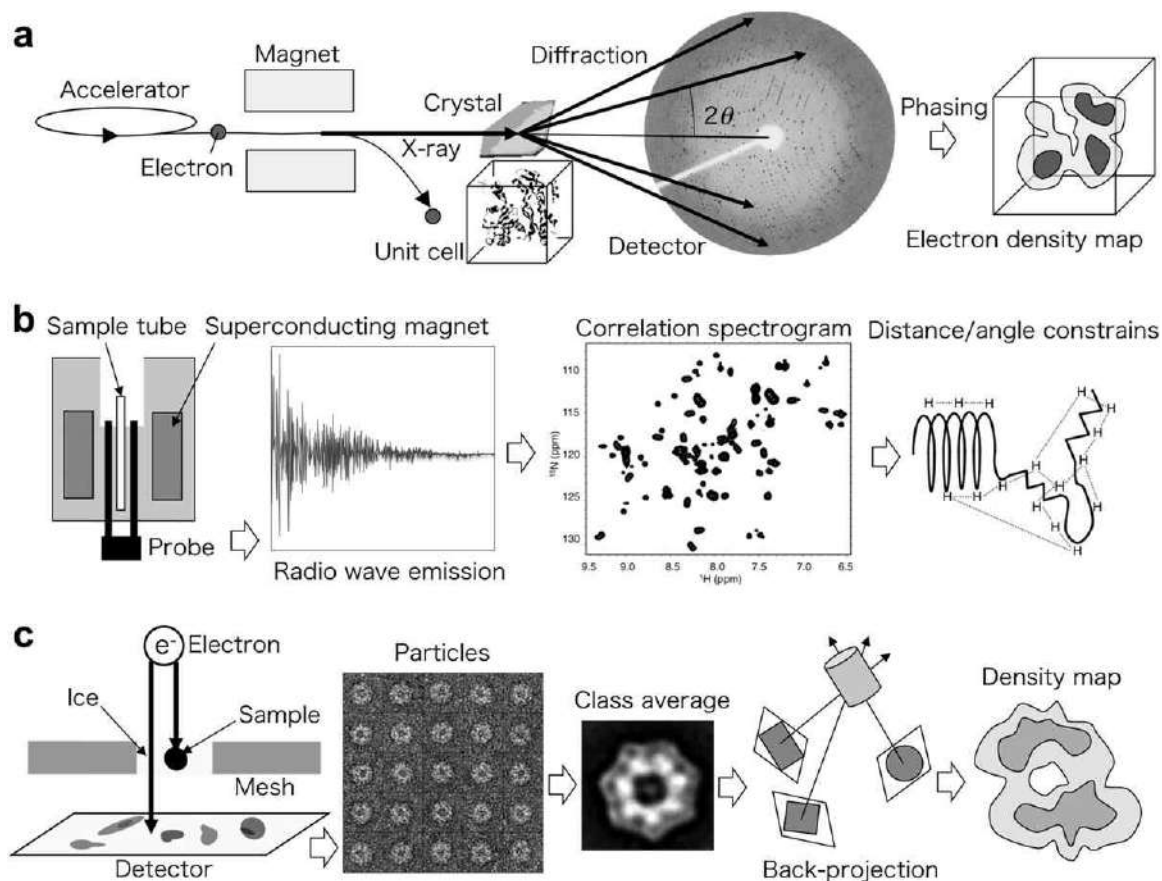
X-ray Crystallography

XP has been the most used method to date, and more than 90% of structures in the Protein Data Bank (PDB) were determined by this method (Berman *et al.*, 2000). Biological molecules like proteins in 1–30 mg/mL concentrations produce an aggregate (precipitate) of regularly assembled molecules, that is, crystals, under appropriate solution conditions. A unit cell, which contains defined number of molecules in defined structures, is a virtual building block of crystals (Drenth and Mesters, 2007). The form of a crystal is defined by the axis dimensions (a , b , c), angles between axes (α , β , γ), and symmetry (space groups) of the unit cell. The unit cell is further divided into asymmetric units (ASU), which is the minimum set of atoms that must be determined independently. The unit cell can be reproduced by applying symmetry operations defined by the space group to the ASU, and a whole crystal is constructed by simply compiling unit cells.

Crystals are exposed to an X-ray beam of uniform wavelength and high coherency from a synchrotron or an X-ray generator for diffraction experiments (Fig. 1(a)). An X-ray is light with a wavelength 0.1–100 Å (0.01–10 nm). The electrons in a crystal scatter X-rays, and the scattered X-rays, which have delayed phases depending on the locations of the scatter, interfere with each other.

Table 1 Brief summary of protein structure analysis and validation methods

Method	X-ray crystallography (XP/XRC)	Nuclear magnetic resonance spectroscopy (NMR)	Electron microscopy (EM)
Sample form	Crystal	Solution/powder	Stained/frozen solution on mesh
Probe	X-ray	Microwave	Electron
Direct data	Diffraction intensities	Spectrograms	Particle images
Resolution	<1.0–4.0 Å	~2.5 Å equivalent (to XP)	10–20 Å (2–4 Å in cryo-EM)
Model constraints	Electron density map	Distance/angle constraints	EM density map
Model characteristic (see also Fig. 2)	Atomic model without hydrogens	Multiple atomic models with hydrogens	Subunits fit in map
Specific validation metrics of data	Resolution, completeness, $\langle I/\sigma I \rangle$, R_{sym}	Completeness of signal assignment, number of constraints	Resolution, number of particle images
Specific validation metrics of model	R -factor (R), free R -factor (R_{free})	Average numbers of distance/angle constraint violations, largest distance/angle violations	Map-model cross correlation
Advantage	Atomic resolution	Dynamic structure	Large supramolecule
Disadvantage	Crystallization required (partly fixed in XFEL)	Low MW limit	Low resolution (partly fixed in cryo-EM)

**Fig. 1** Schematic processes of (a) X-ray crystallography, (b) nuclear magnetic resonance spectroscopy, and (c) electron microscopy (cryo-EM).

Therefore, scattered X-rays yield information about the locations of electrons/atoms. Scattered X-rays are very weak and usually cancel each other out. Due to the regular structure of crystals, however, scattered X-rays are not muted in the directions where the phase delay between crystal unit cells is equal to an integer multiple of the wavelength, and such outputs, observed as X-ray spots from the crystal, are called diffractions. Thus, crystals are necessary for XP as “amplifiers” of scattered X-rays.

Resolution is the smallest distance between imaginary mirror planes defined by the unit cell lattice, which diffract (reflect) the incident X-ray, and therefore is the important parameter in judging how finely the structures are measured. The smaller the value, the higher the resolution, and the resolution is typically from 1.0 (very high) to 3.0 Å (low) for protein crystals. Each diffraction spot is indexed with a Miller index (h, k, l), which defines the mirror plane reflecting the corresponding diffraction (mirror plane is defined by three points a/h , b/k , and c/l , where a , b , and c are unit cell axes).

Diffractions are the Fourier transform of electron density in the unit cell. Their intensities are recorded by diffractometers, such as an imaging plate or digital camera. The intensities of diffraction are integrated from the diffraction images. The programs HKL2000 and iMosflm are often used for this process (Battye *et al.*, 2011; Otwinowski and Minor, 1997). The electron density map in a unit cell is retrievable as the inverse Fourier transform of the diffractions. The phase information, however, cannot be recorded by diffractometers, and must be recovered by other means. This is called the phasing problem of XP. Multiple isomorphous replacement (MIR) and anomalous scattering (AS) are experimental methods for recovering phases, in which scatterings from a few heavy atoms introduced into the crystals are overlaid in diffractions in order to deduce phases from the amplitude differences. Molecular replacement (MR) is an alternative computational method, in which calculated phases from homologous protein structures (placed appropriately in the unit cell) are used instead of experimental phases.

Once the electron density is successfully synthesized, an atomic model of the molecule is built so that it fits the density as far as possible, usually half-manually using a computer graphics program (Fig. 2(a)). This process of adjusting the model (set of atomic coordinates) in order to explain diffractions is called refinement, and CCP4, X-PLOR-NIH, SHELEX, and PHENIX program suites are frequently used for this purpose (Adams *et al.*, 2010; Schwieters *et al.*, 2003; Sheldrick, 2008; Winn *et al.*, 2011). The modeling process can be automated if a very fine electron density is obtained.

The small angle X-ray scattering (SAXS) method also utilizes X-rays to determine the overall shape of molecules. When a protein solution (not crystallized) is irradiated with X-rays, the protein molecules scatter X-rays in a spherically averaged manner. The observed X-ray intensity over the scattering angle is converted into the frequency distribution of pairs of points within the molecule, and this pair distance distribution can be used to define the shape of a molecule. Although SAXS does not provide structures in atomic detail, this method is often employed to determine the size, molecular association, and interaction of proteins (Tuukkanen *et al.*, 2017).

Recently, XFEL (X-ray free electron laser) facilities are provided in major synchrotron X-ray sources (Spence, 2017). XFEL is a laser oscillated from accelerated free electrons. Because of its very high intensity and coherency, XFEL can be used for diffraction of microcrystals or even a single particle. Each single microcrystal or particle is exposed to a pulse X-ray of fs (10^{-12} s) duration only once. Therefore, the XFEL diffraction data are theoretically unaffected by X-ray radiation damage, and it enables time-resolved tracing of chemical reactions on 3D structures.

Nuclear Magnetic Resonance Spectroscopy

Atoms behave like magnets (due to the spin magnetic moment of nuclei), and these magnets can be manipulated by radio waves externally. The properties of these atomic magnets depend on the local molecular environment, and their measurement provides information about how the atoms are linked chemically, and how close they are in space. This information can be used to determine the distance between atoms, and then the overall structure of biological molecules (Wuthrich, 2001).

NMR samples are either in solution or solid (typically powder) phases. Usually the protein molecules are labeled with isotopes, such as ^{15}N , ^{13}C , and/or ^2H , to facilitate the measurements. A highly purified protein solution at a concentration of 0.1–3 mM is placed in the superconductance magnet, the sample is irradiated with a series of radio waves (pulse sequence) to activate particular nuclei (spin magnetization), and the radio wave emissions from the sample are measured (Fig. 1(b)).

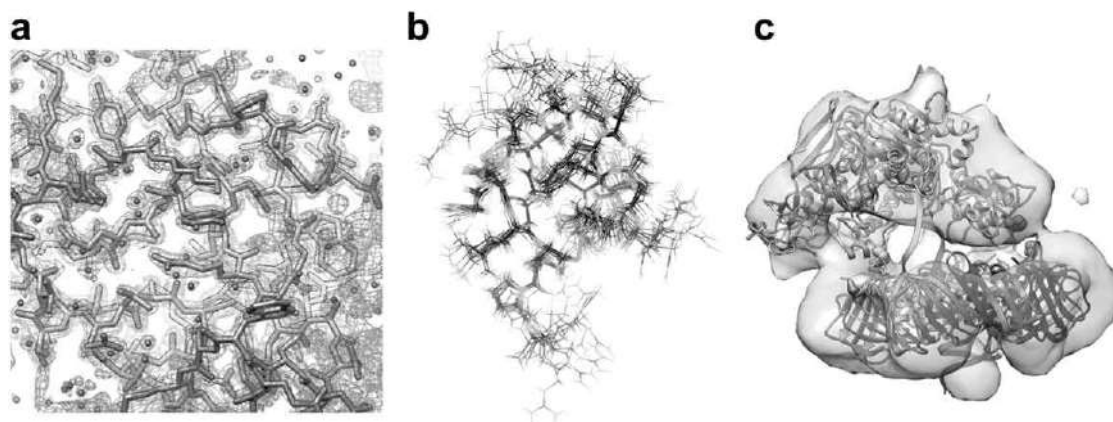


Fig. 2 Typical model images of (a) X-ray crystallography (atomic models fit into an electron density map), (b) nuclear magnetic resonance spectroscopy (multiple atomic models satisfying distance/angle constraints), and (c) electron microscopy (models fit into a density map).

A series of observed radio wavelengths, expressed as per-million difference (units in ppm) from reference wavelengths are the “fingerprints” of nuclei, and are called chemical shifts. Spin magnetization is transferred through chemical bonds among covalently connected atoms and also through space among atoms in close proximity. The chemical shifts of nuclei are perturbed by the status of adjacent nuclei. Correlation spectrograms, which depict the spin coupling (correlations among nuclei), are measured by magnetizing specific nuclei and observing whether the other nuclei are affected by them. The introduction of multiple isotopes is required to resolve the spectra, which tend to overlap each other (multidimensional NMR spectroscopy).

Various types of multidimensional NMR spectroscopic techniques have been devised to efficiently correct spin coupling, and they are usually combined step by step to construct molecular structural models. Typically, the first step is assigning the chemical shifts to specific residues and atoms using the heteronuclear single-quantum correlation (HSQC) method, in which one amino acid residue (except for proline) gives one signal. The $^1\text{H}^{15}\text{N}$ -HSQC spectra are often referred to as the fingerprint of a protein because each protein has a unique pattern of signal positions. These experiments allow each $^1\text{H}^{15}\text{N}$ peak to be linked to the preceding carbonyl carbon by tracing the coupling between nuclei through chemical bonds with the HNCA (measuring coupling through $\text{C}\alpha\text{--NH--C}\alpha$ atoms of a peptide) and the HN(CO)CA (measuring coupling through $\text{NH--CO--C}\alpha$ atoms of a peptide) methods (Bax and Ikura, 1991). The peaks of sidechain atoms are assigned through COSY (correlation spectroscopy) and TOCSY (total correlation spectroscopy). In this way, chemical shifts are assigned to an amino acid sequence.

Nuclear Overhauser effect (NOE) spectroscopy (NOESY) is additionally required in modeling protein conformation. Because magnetizations are transferred through space in this experiment, NOESY will show cross peaks for all ^1H atoms that are close in space regardless of whether they belong to same residue or not. A cross-peak in a NOESY experiment signifies spatial proximity (~ 6.0 Å) between the two nuclei in question, and thus it is used for distance constraints. NOESY is also used to obtain torsion angle constraints. The experimentally determined constraints of distance and angle are used as input for the structure calculation process. Usually, multiple models, which equally satisfy the constraints, are generated in this process. Computer programs such as XPLOR-NIH, CYANA, or ARIA/CNS attempt to construct atomic models that satisfy as many of the constraints as possible (Fig. 2(b)) (Linge *et al.*, 2001; Schwieters *et al.*, 2017; Wurtz *et al.*, 2017).

Important parameters, such as the interactions and dynamics of molecules, can be additionally obtained with NMR experiments. NMR spectroscopy is nucleus specific; thus, ^1H (hydrogen) and ^2H (deuterium) are distinguished. The amide protons (labeled with ^1H for example) in a peptide exchange readily with the hydrogen of a solvent containing a different isotope ($^2\text{H}_2\text{O}$). This H–D (hydrogen–deuterium) exchange can be used to investigate the folding process of proteins, because the parts of a protein folded early in the folding process are protected from hydrogen–deuterium exchanges (Englander and Kallenbach, 1983).

NOEs are also observed between atoms of a protein and interacting molecules. If the NOE spectrum of a residue is different in the presence and absence of ligands (chemical shift perturbation), it suggests that the residue takes part in the molecular interactions (Meyer and Peters, 2003). NMR can also yield information on the dynamics of a protein by measuring relaxation times, that is, times before magnetizations of nuclei degenerate. NMR relaxation is a consequence of overall or local molecular motions and can be measured by various types of HSQC methods. The motions occur on a time-scale of about 10^{-12} – 10^{-9} s, and usually the relaxation time becomes longer when the molecular motion is faster (Sapienza and Lee, 2010).

Electron Microscopy

EM is a rather intuitive method for determining molecular structures because it directly observes images of the molecule (Adrian *et al.*, 1984; Crowther and Klug, 1975). An electron microscope uses electrons, instead of the light in traditional microscopes, for imaging. High-energy electrons are generated and condensed, and samples are irradiated and focused on detectors by using electromagnets as lenses (Fig. 1(c)). The protein solution samples are adsorbed on a carbon grid and stained by heavy atoms, such as uranium acetate, for contrasting images. In a frequently employed method, the interprotein regions rather than the proteins are stained with heavy atoms, and the method is called negative staining. Recently, a method of rapidly freezing the sample solution in mesh pores, which is called cryo-electron microscopy (cryo-EM), is becoming popular (Cheng *et al.*, 2015). Damage of samples by irradiation and the heavy atom stain is reduced in cryo-EM, although image contrast is generally lower than with the negative staining method.

Transmission electron microscopy (TEM), which observes the electrons transmitted through a sample, is generally used for determining molecular structures. First, images of the biological molecules (particles) spread on a mesh are taken. The images of each particle are generally obscure, and further image processing is required. In the canonical procedure, the particle images are picked up separately, and categorized into clusters of those viewing the same molecule in the same direction. The images in each cluster are aligned and averaged into class averages. The class averages are then assigned Euler angles (of viewing angles), and a 3D molecular density map is constructed with the back-projection method (Orlov *et al.*, 2006).

Usually, back-projection is done in an iterative manner. First, a spherical density map is prepared, and the density map and the assigned Euler angles of the class average are refined step-by-step to be more consistent with each other. Back-projection might be done in either real or reciprocal spaces. The latter method is similar to the synthesis of an electron density map in XP: the 2D class averages are Fourier transformed and assigned angles, and the density map is synthesized as a reverse Fourier transformation. Although these processes might be executed automatically, human intervention is usually required because appropriately selecting particles and/or class averages is often critical for the quality of the final density map. The most used program suites for this process are EMAN2, SPIDER, and Relion (Fernandez-Leiro and Scheres, 2017; Shaikh *et al.*, 2008; Tang *et al.*, 2007).

Molecular structural models are constructed to fit the density map. For standard EM density maps of typical resolution, 10–20 Å, detailed atomic models cannot be constructed based on the maps (Fig. 2(c)). Usually, therefore, the atomic models that have been

determined by XP or NMR are used to build pieces of the model. The pieces (partial models) are placed in the density map so that the experimental and model-derived densities correlate with each other. This process is often executed manually by inspecting the density map on computer graphic programs.

Recently, mainly owing to the development of the direct electron detector (DED), advanced cryogenic technique, and sophisticated image processing software such as Relion, atomic resolution (2–4 Å resolution) analyses have become possible with cryo-EM (Callaway, 2015; Subramaniam *et al.*, 2016). The density map in this resolution range can be interpreted like the electron density maps from XP, and refinement procedures/programs for XP might be used for constructing and refining models (Murshudov, 2016).

Structure Validation Methods

Since protein structures are “theoretical” models that fit experimental data as far as possible, it is very important to detect modeling errors and validate how accurate the constructed models are determined. A structure validation process consists of three major parts: evaluations of (1) experimental data quality/quantity; (2) consistency between experimental data and structural models, that is, to what extent the experimental data is realized in the models; and (3) quality of model atomic coordinates, that is, how much the model parameters deviate from the canonical (dictionary) values derived from certified examples. The former two criteria are mainly specific to each structure-determination method, and only the last one is largely common among all methods (Table 1). The last one is based on a set of rules that any molecules must observe as a real entity, and thus also applicable to evaluation of the models from various computational predictions.

X-ray Crystallography

The quality of XP experimental data can be defined by a large variety of parameters. Among them, resolution, completeness, $\langle I/\sigma I \rangle$, and R_{sym} (also called R_{merg}) are the most fundamental set of parameters (Drenth and Mesters, 2007). Resolution (d) is inversely proportional to the sine of the scattering angle of diffraction (θ) as $2d \sin \theta = n\lambda$ (Bragg's law; λ is X-ray wavelength). Thus, the higher (smaller in figure) the resolution, the larger the number of diffraction data points, and the more finely a structure might be defined. Completeness is the fraction of diffraction data actually observed of that theoretically observable for the resolution range (usually more than 90% is required). The average of diffraction intensity (I) over the standard deviation of the intensity (σI) in the proximal background is $\langle I/\sigma I \rangle$, and the higher the values, the greater the significance of the signals. There are sets of independent diffractions, the intensities of which are theoretically equal due to the nature of diffraction and crystal symmetry. The equivalent diffraction intensities are scaled and averaged into a single datum during the X-ray data reduction process. R_{sym} is the residual of the observed intensities (I) from the averaged one ($\langle I \rangle$) over total intensities, which is defined as $R_{sym} = \sum_{hkl} (I - \langle I \rangle) / \sum_{hkl} \langle I \rangle$. R_{sym} below 0.1 (10%) is preferable.

The consistency of models and experimental data in XP is also evaluated by various criteria, and R -factor and free R -factor are the basic components of such values. R (-factor) represents the residual of observed structure factors (F_o ; square root of I) from those evaluated from the model (F_c), as $R = \sum_{hkl} (|F_o| - |F_c|) / \sum_{hkl} |F_o|$. F_c values are calculated by simulating X-ray diffraction from the model atomic coordinates. R is used as a target function to be minimized during model refinement. Usually, a fraction (5%–10%) of observed diffractions are selected randomly and uniformly over resolution ranges, and are not referenced during refinement for cross-validation purposes. The R -factor calculated on those set-aside diffractions alone is called the free R (R_{free} or fR). Usually, an R below 0.2 (20%) and a difference between R and R_{free} less than 0.1 are acceptable for XP models.

Nuclear Magnetic Resonance Spectroscopy

Due to the difference in methodology, resolution is not definable for NMR data. The quality of data in NMR is defined by the completeness of resonance assignments, that is, to what extent the observed chemical shifts are assigned to specific atoms, and the number of conformational constraints derived from correlation spectroscopy, where a higher number of constraints, if consistent, is preferable (Rosato *et al.*, 2013).

The consistency of models and experimental data in NMR is assessed by the numbers of experimental constraints that were not satisfied by the models, that is, average numbers of distance constraint violations (typically separately evaluated as short 0.1–0.2 Å, middle 0.2–0.5 Å, and long > 0.5 Å ranges, for example) and dihedral angle constraints (also separately evaluated in small 1–10° and large > 10° angle ranges, for example) per structure. The largest distance and dihedral angle violations should also be indicated. Because NMR generates multiple models, these metrics are usually presented as an average and a standard deviation over models. An NOE completeness score, which is defined as the ratio of the number of experimentally observed NOEs to the number of expected NOEs from the model, is also used (Doreleijers *et al.*, 1999).

Electron Microscopy

The major parameters characterizing EM data are the number of particles and map resolution. The number of particles is the number of individual molecular images cut out from the original microscopic images, which often contain hundreds of molecules

in the field of vision. More than 10,000 to 100,000 particle images are usually collected for a structure determination. Because a 3D density map is constructed by back-projection of 2D class averages, a nonbiased distribution of views (Euler angles assigned to class averages) is also required to justify the reconstruction.

Resolution in EM analysis is defined differently from that of XP. To estimate EM map resolution, particle images are randomly divided into two sets of equal numbers, and a map is constructed for each set independently. Then, the density maps are Fourier-transformed, and correlation coefficients (FCS) between sets are evaluated in resolution (wavelength) shells. Generally, the FCS decreases as resolution increases, and the resolution is defined as the point where the FCS falls at 0.5 (low resolution analyses) or 0.143 (high resolution analyses) (Baker *et al.*, 2010).

Consistency of models and experimental data in EM is mainly evaluated as cross correlation between the density of the experimental map and that calculated from the model (Monroe *et al.*, 2017; Xu and Volkman, 2015).

Consistency With Dictionary Values

The stereochemical parameters of model atomic coordinates, such as bond length, bond angle, dihedral angle, and chiral volume, should be statistically consistent with the authentic reference values (Fig. 3(a)) (Evans, 2007). Methods/programs such as PROCHECK or Molprobity are often used for evaluating stereochemical parameters and detecting errors in models (Chen *et al.*, 2010; Hintze *et al.*, 2016; Laskowski *et al.*, 1993). Deposition of structural models in the PDB before publication is recommended/required, and the model atomic coordinates are checked through a program pipeline of the PDB validation server (Gore *et al.*, 2017). The set of scores in the PDB validation report are representative metrics of the stereochemical quality of the models.

The deviations of chemical bond lengths and angles of models from high-quality reference structures (including crystal structures of small molecules) are some of the most fundamental metrics (Bruno *et al.*, 2004). These values are presented as RMSZ, which are the root-mean-square value of the Z-scores (normalized values with average and standard deviation of reference structures) (Tickle, 2007). RMSZ scores of bond length and bond angles are around 0.5 for well-defined models. The root-mean-square deviations in an unnormalized scale are typically ~ 0.01 Å and $\sim 1.2^\circ$ for bond lengths and angles, respectively.

A Ramachandran plot is a traditional yet still-in-use method to characterize the main-chain conformation of proteins (Ramachandran *et al.*, 1963). When main-chain torsion angles ϕ (horizontal axis) are plotted against ψ (vertical axis) (Fig. 3(b)), the plot field is divided into allowed (originally further divided into favored, allowed, and generously allowed regions) and disallowed regions (Fig. 3(c)). Empirically, nearly 100% of residues fall into the allowed region, and the residues outside of this region are thought to be outliers (probable modeling defects). High-quality reference structures contain less than 0.5% of residues as outliers.

The clashscore depicts atomic steric hindrance as the number of too-close contacts per 1000 atoms. An all-atom model is constructed by providing hydrogen atoms on the deposited atomic coordinates, and when van der Waals surfaces overlap more than ~ 0.4 Å between nonbonded atoms, the pair of atoms is identified as a clash (Fig. 3(d)) (Word *et al.*, 1999). An average clashscore in reference structures is 5–20.

Sidechain conformation of amino acid residues can be defined as a set of χ torsion angles. These angles adopt a certain preferred set of values depending on the amino acid. A frequently observed sidechain conformation is called a rotamer or a rotameric conformer. The sidechain conformations observed in less than 0.3% of the residues in reference structures are defined as outliers. Models are scored by the percent of outlier sidechains in total (Williams *et al.*, 2017).

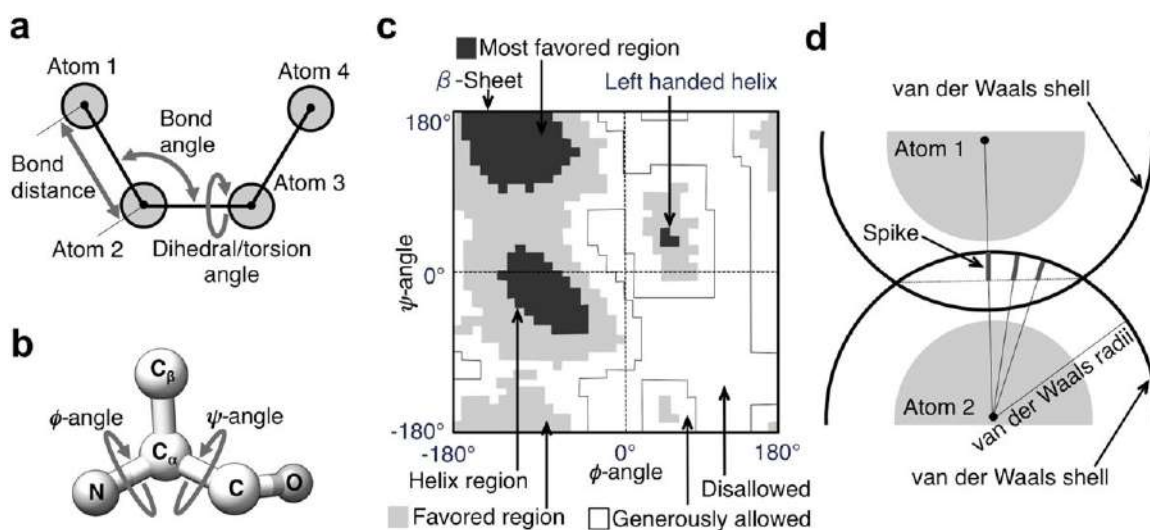


Fig. 3 (a) Schema of basic stereochemical parameters. (b) Definition of main-chain torsion angles ϕ and ψ . (c) Schema of the Ramachandran plot and area participations. (d) An overlap between van der Waals shells of two nonbonded atoms is defined by the lengths of spikes.

Other knowledge-based validation methods are also proposed. For example, a protein structural model with perfect stereochemical parameters may be still “unnatural” because amino acids show preferred distributions in 3D structures depending on their chemical properties, such as hydrophilicity or hydrophobicity. Programs such as PROSA2 or Verify3D are used for evaluating the molecular environments of each residue in a model, and numerically assess the fitness of the residues to the model structure by using an empirically derived function, which is used in threading methods (method to detect fitting of amino acid sequences to protein folds) (Luthy *et al.*, 1992; Wiederstein and Sippl, 2007). These knowledge-based validation methods are often useful to find local mistakes in modeling.

Conclusion and Prospects

Since 1950, when the structure of myoglobin was first determined using XP (Kendrew *et al.*, 1958), many improvements have been made in structure-determination methods, and several revolutionary changes, represented by cryo-EM and XFEL, are currently ongoing. These efforts are largely dedicated to determining large supramolecular structures, and elucidating their functions in atomic detail. Although current validation methods mainly concern the structure of each molecule (subunit), methods suitable to evaluate the correctness of complex structures as a whole will be required in the near future.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Assessment of Structure Quality (RNA and Protein). Biomolecular Structures: Prediction, Identification and Analyses. Drug Repurposing and Multi-Target Therapies. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Protein Structural Bioinformatics: An Overview. Protein Structure Databases. Protein Three-Dimensional Structure Prediction. Protocol for Protein Structure Modelling. Rational Structure-Based Drug Design. Secondary Structure Prediction. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Study of The Variability of The Native Protein Structure

References

- Adams, P.D., Afonine, P.V., Bunkoczi, G., *et al.*, 2010. PHENIX: A comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.* 66, 213–221.
- Adrian, M., Dubochet, J., Lepault, J., McDowell, A.W., 1984. Cryo-electron microscopy of viruses. *Nature* 308, 32–36.
- Baker, M.L., Zhang, J., Ludtke, S.J., Chiu, W., 2010. Cryo-EM of macromolecular assemblies at near-atomic resolution. *Nat. Protoc.* 5, 1697–1708.
- Battye, T.G., Kontogiannis, L., Johnson, O., Powell, H.R., Leslie, A.G., 2011. iMOSFLM: A new graphical interface for diffraction-image processing with MOSFLM. *Acta Crystallogr. D Biol. Crystallogr.* 67, 271–281.
- Bax, A., Ikura, M., 1991. An efficient 3D NMR technique for correlating the proton and 15N backbone amide resonances with the alpha-carbon of the preceding residue in uniformly 15N/13C enriched proteins. *J. Biomol. NMR* 1, 99–104.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The Protein Data Bank. *Nucleic Acids Res.* 28, 235–242.
- Bruno, I.J., Cole, J.C., Kessler, M., *et al.*, 2004. Retrieval of crystallographically-derived molecular geometry information. *J. Chem. Inf. Comput. Sci.* 44, 2133–2144.
- Callaway, E., 2015. The revolution will not be crystallized: A new method sweeps through structural biology. *Nature* 525, 172–174.
- Cheng, Y., Grigorieff, N., Penczek, P.A., Walz, T., 2015. A primer to single-particle cryo-electron microscopy. *Cell* 161, 438–449.
- Chen, V.B., Arendall 3rd, W.B., Headd, J.J., *et al.*, 2010. MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D Biol. Crystallogr.* 66, 12–21.
- Crowther, R.A., Klug, A., 1975. Structural analysis of macromolecular assemblies by image reconstruction from electron micrographs. *Annu. Rev. Biochem.* 44, 161–182.
- Doreleijers, J.F., Ravest, M.L., Rullmann, T., Kaptein, R., 1999. Completeness of NOEs in protein structure: A statistical analysis of NMR. *J. Biomol. NMR* 14, 123–132.
- Drenth, J., Mesters, J., 2007. *Principles of Protein X-Ray Crystallography*, third ed. New York: Springer.
- Englander, S.W., Kallenbach, N.R., 1983. Hydrogen exchange and structural dynamics of proteins and nucleic acids. *Q. Rev. Biophys.* 16, 521–655.
- Evans, P.R., 2007. An introduction to stereochemical restraints. *Acta Crystallogr. D Biol. Crystallogr.* 63, 58–61.
- Fernandez-Leiro, R., Scheres, S.H.W., 2017. A pipeline approach to single-particle processing in RELION. *Acta Crystallogr. D Struct. Biol.* 73, 496–502.
- Frank, J., 2006. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. New York: Oxford University Press.
- Gore, S., Sanz Garcia, E., Hendrickx, P.M.S., *et al.*, 2017. Validation of structures in the Protein Data Bank. *Structure* 25 (12), 1916–1927.
- Hintze, B.J., Lewis, S.M., Richardson, J.S., Richardson, D.C., 2016. Molprobity's ultimate rotamer-library distributions for model validation. *Proteins* 84, 1177–1189.
- Kendrew, J.C., Bodo, G., Dintzis, H.M., *et al.*, 1958. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature* 181, 662–666.
- Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M., 1993. PROCHECK: A program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.* 26, 283–291.
- Linge, J.P., O'Donoghue, S.I., Nilges, M., 2001. Automated assignment of ambiguous nuclear overhauser effects with ARIA. *Methods Enzymol.* 339, 71–90.
- Luthy, R., Bowie, J.U., Eisenberg, D., 1992. Assessment of protein models with three-dimensional profiles. *Nature* 356, 83–85.
- Meyer, B., Peters, T., 2003. NMR spectroscopy techniques for screening and identifying ligand binding to protein receptors. *Angew. Chem. Int. Ed. Engl.* 42, 864–890.
- Monroe, L., Terashi, G., Kihara, D., 2017. Variability of protein structure models from electron microscopy. *Structure* 25, 592–602. e592.
- Murshudov, G.N., 2016. Refinement of atomic structures against cryo-EM maps. *Methods Enzymol.* 579, 277–305.
- Orlov, I.M., Morgan, D.G., Cheng, R.H., 2006. Efficient implementation of a filtered back-projection algorithm using a voxel-by-voxel approach. *J. Struct. Biol.* 154, 287–296.
- Otwinowski, Z., Minor, W., 1997. Processing of X-ray diffraction data collected in oscillation mode. *Methods Enzymol.* 276, 307–326.
- Ramachandran, G.N., Ramakrishnan, C., Sasisekharan, V., 1963. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* 7, 95–99.
- Rosato, A., Tejero, R., Montelione, G.T., 2013. Quality assessment of protein NMR structures. *Curr. Opin. Struct. Biol.* 23, 715–724.

- Sapienza, P.J., Lee, A.L., 2010. Using NMR to study fast dynamics in proteins: Methods and applications. *Curr. Opin. Pharmacol.* 10, 723–730.
- Schwieters, C.D., Bermejo, G.A., Clore, G.M., 2017. Xplor-NIH for molecular structure determination from NMR and other data sources. *Protein Sci* 27 (1), 26–40.
- Schwieters, C.D., Kuszewski, J.J., Tjandra, N., Clore, G.M., 2003. The Xplor-NIH NMR molecular structure determination package. *J. Magn. Reson.* 160, 65–73.
- Shaikh, T.R., Gao, H., Baxter, W.T., *et al.*, 2008. SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. *Nat. Protoc.* 3, 1941–1974.
- Sheldrick, G.M., 2008. A short history of SHELX. *Acta Crystallogr. A* 64, 112–122.
- Spence, J.C.H., 2017. XFELs for structure and dynamics in biology. *IUCrJ* 4, 322–339.
- Subramaniam, S., Kuhlbrandt, W., Henderson, R., 2016. CryoEM at IUCrJ: A new era. *IUCrJ* 3, 3–7.
- Tang, G., Peng, L., Baldwin, P.R., *et al.*, 2007. EMAN2: An extensible image processing suite for electron microscopy. *J. Struct. Biol.* 157, 38–46.
- Tickle, I.J., 2007. Experimental determination of optimal root-mean-square deviations of macromolecular bond lengths and angles from their restrained ideal values. *Acta Crystallogr. D Biol. Crystallogr.* 63, 1273–1282.
- Tuukkanen, A.T., Spilotros, A., Svergun, D.I., 2017. Progress in small-angle scattering from biological solutions at high-brilliance synchrotrons. *IUCrJ* 4, 518–528.
- Wiederstein, M., Sippl, M.J., 2007. ProSA-web: Interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Res.* 35, W407–W410.
- Williams, C.J., Headd, J.J., Moriarty, N.W., *et al.*, 2017. MolProbity: More and better reference data for improved all-atom structure validation. *Protein Sci.* 27 (1), 293–315.
- Winn, M.D., Ballard, C.C., Cowtan, K.D., *et al.*, 2011. Overview of the CCP4 suite and current developments. *Acta Crystallogr. D Biol. Crystallogr.* 67, 235–242.
- Word, J.M., Lovell, S.C., LaBean, T.H., *et al.*, 1999. Visualizing and quantifying molecular goodness-of-fit: Small-probe contact dots with explicit hydrogen atoms. *J. Mol. Biol.* 285, 1711–1733.
- Wurz, J.M., Kazemi, S., Schmidt, E., Bagaria, A., Guntert, P., 2017. NMR-based automated protein structure determination. *Arch. Biochem. Biophys.* 628, 24–32.
- Wuthrich, K., 1986. *NMR of Proteins and Nucleic Acids*. John Wiley & Sons Inc.
- Wuthrich, K., 2001. The way to NMR structures of proteins. *Nat. Struct. Biol.* 8, 923–925.
- Xu, X.P., Volkman, N., 2015. Validation methods for low-resolution fitting of atomic structures to electron microscopy data. *Arch. Biochem. Biophys.* 581, 49–53.

Protein Structure Visualization

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal and University of Minho, Braga, Portugal

Soumya Lipsa Rath, Nagoya University, Nagoya, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

The usage of graphical presentations greatly helps to successfully describe, communicate or understand structural concepts related to various biological phenomena (Sánchez-Ferrer *et al.*, 1995). A new era of structural biology dawned when the first 3D structure of a protein (myoglobin) was solved by Kendrew *et al.* (1958). They built a physical “ball and spoke” model made of brass at a scale of 5 cm/Å and supported by 2500 vertical rods and colored clips that signified the electron density. The size of the model made it cumbersome and problematic to move whereas the forest of rods obscured the view of the model and made it hard to adjust. Nonetheless, this step was so important that for their studies on the structures of globular proteins, Max Perutz and John C Kendrew won the 1962 Nobel Prize in chemistry. In the meanwhile, Byron Rubin invented a machine for bending wire to follow the backbone trace of a protein and build small backbone wire models. These “wireframe models” were the most manipulable and portable models available at the time. Later during the 1960s and 1970s, the earliest of computer representations for molecules surfaced. Cyrus Levinthal and his colleagues at MIT developed a system that displayed rotating “wireframe” representations of macromolecular structures on an monochrome oscilloscope (Levinthal, 1966). Later, a program known as ORTEP was released by Carroll K. Johnson that could generate stereoscopic drawings of molecular and crystal structures with a pen-plotter. David and Jane Richardson and colleagues (Beem *et al.*, 1977) solved the structure of superoxide dismutase, modeled the metal sites of the enzyme using an interactive density-fitting computer system called “GRIP” and visualized entirely with computers for the first time. Building models for newly solved protein crystals with computers rather than with physical Kendrew-style models became more and more popular.

Since then, scientists all over the world have been studying proteins and other macromolecules at atomic level using structure determination techniques like X-ray, NMR, etc. The availability of protein structures enabled the study of their surfaces and surface characteristics, based on atomic contribution. Consequently, a huge amount of protein structures (134436 on 23 Oct 2017) have been determined and deposited in the Protein Data Bank (PDB; see “Relevant Websites section”) (Berman *et al.*, 2000). PDB stores each structure in a specified format (see “Relevant Websites section”) that describes each molecule as a list of its constituent atoms and their three dimensional (3D) coordinates. This format allows researchers to exchange protein coordinates through a database system. The availability of structural data and advances in computer graphics technologies together have pushed development of computational tools for molecular visualization and analysis (Lesk and Hardman, 1982). Biologists, computer scientists, and application developers usually get together for the development of methods or software for molecular visualization. Some examples of molecular visualization software include VMD (Humphrey *et al.*, 1996), Chimera (Pettersen *et al.*, 2004), and PyMOL (Schrodinger, 2015). In this article, first we describe molecular visualization within the required level of details.

Protein Topology (2D) Visualization

Structural Topology

After determining and analyzing many protein structures, it became clear that protein structures for a particular biological function remained conserved even if the sequences diverged. Wherever 3D structure information is absent, working with secondary structure information can provide a lot of relevant information. In fact, secondary structure information is as important as the final 3D structure of a protein. Therefore, another way of describing the structure of a protein is by describing its 3D fold as a sequence of secondary structure elements (SSEs). These SSEs can be easily assigned using DSSP algorithm (Kabsch and Sander, 1983) and later represented graphically in a diagram. Topology, in terms of protein secondary structure, can be defined as the structural elements which remain unchanged as a protein undergoes continuous deformation, or simply the sequence of SSEs (Martin, 2000). Using these SSEs, several authors have tried to develop a system for the topological classification of proteins (Flower, 1994; Koch *et al.*, 1992; Richardson, 1977). Below we provide the most accepted tools regarding the algorithms and methods used for protein topology depiction and searching/matching.

HERA

HERA is an application for generating hydrogen bonding diagrams for proteins (Hutchinson and Thornton, 1990). The program uses SSEs determined by DSSP algorithm and plots a diagram in which helical residues are represented as helical wheels or helical nets. It also provides an option to automatically search for simple β -sheet motifs in a database. PDBsum (Laskowski *et al.*, 2018), a database which provides summary information about each experimentally determined structure in PDB, uses HERA for generating

hydrogen bonding plots (Laskowski, 2009). An example hydrogen bonding diagram for rabbit L-gulonate 3-dehydrogenase [3ADO] is given in Fig. 1(a).

TOPS

Another such method for plotting protein topology is known as topology cartoons or diagrams (Westhead *et al.*, 1998; Levitt and Chothia, 1976). It has been popular among several others approaches (Richardson, 1977; Tsukamoto *et al.*, 1997; Bansal *et al.*, 2000; Nagano, 1977). The TOPS algorithm was developed to generate topology cartoons for proteins depicting α -helices as circles and β strands with triangles, both of which are used as nodes in a mathematical graph connected by five edge sets (Flores *et al.*, 1994; Westhead *et al.*, 1999). The five edge sets include the directed N- to C-terminal sequential edges, and the four other undirected pairwise relationships between SSEs. The undirected relationships include hydrogen bonding, helix packing, chirality and neighbor relationships. Direction information for strands is duplicated, upward pointing triangles indicating 'up' strands and *vice versa* (Gilbert *et al.*, 1999). It ignores the details like the lengths of SSEs and loops however describes their spatial adjacency within the fold and approximate orientation. One example database that stores this topological descriptions of protein structures is TOPS database (Michalopoulos *et al.*, 2004). TOPS representations and database has been used for visualizing, searching and comparing (especially within a given structural family) fold topologies (Gilbert *et al.*, 2001; Viksna and Gilbert, 2001). Computer scientists and bioinformaticians used it for automated extraction of patterns relating protein sequence to topology and function. Unfortunately, the TOPS servers located at the website provided in "Relevant Websites section" are no more functional. The TOPS graph model was enhanced by replacing the graph model with a new string model (TOPS+) which included previously omitted loops and other structural and biochemical features, like ligand interaction and length of the SSEs (Veeramalai and Gilbert, 2008).

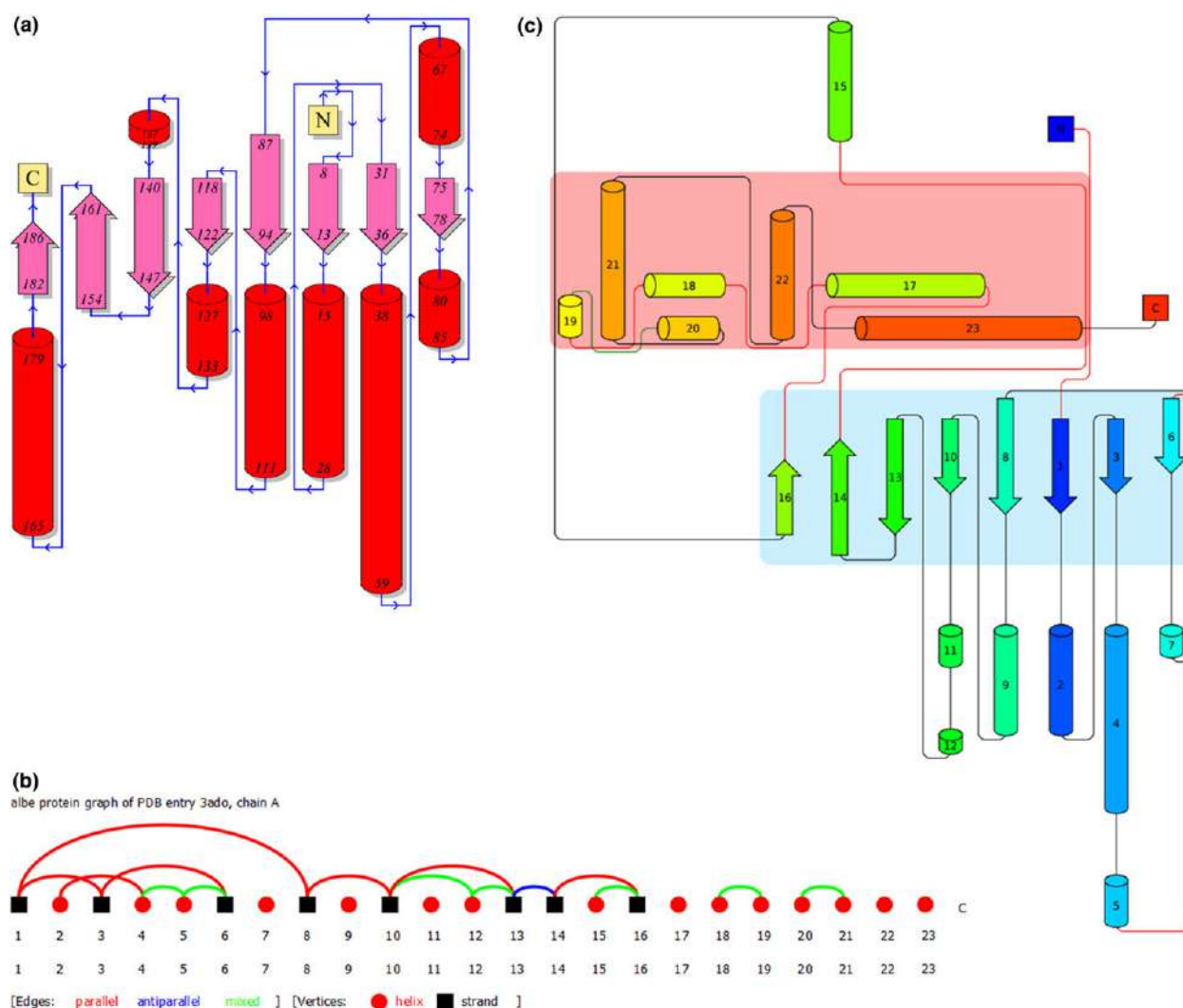


Fig. 1 (a) A hydrogen bonding diagram for rabbit gulonate dehydrogenase [3ADO] using the program HERA. This image has been generated using PDBsum website. (b) Protein topology graph generated for same 3ADO using PTGL. (c) Pro-Origami diagram for 3ADO topology.

Other Approaches for Protein Topology Analysis

Bostick *et al.* developed a topological representation of proteins by identifying natural nearest neighboring residues and the distance between residues in sequence space (Bostick *et al.*, 2004; Bostick and Vaisman, 2003). This representation allowed a quantitative and computationally inexpensive comparison of protein structural topologies. TOPSCAN was developed for a rapid comparison of protein structures (Martin, 2000). It uses a string notation which reduces the protein structure to a topology string where the letters encode the primary or secondary topology. The topology strings describe about 52 vectors of characters and incorporates information (like direction) from the three-dimensional structure. It can be accessed at the website provided in “Relevant Websites section”. On the other hand, Protein Topology Graph Library (PTGL; see “Relevant Websites section”) relies simply on the graph definitions without explicitly considering the order of SSEs (May *et al.*, 2010). Helices and strands are drawn as red circles (or cylinders) and black squares (or arrows), respectively (Fig. 1(b)). It also serves as a database on protein topologies with facilities for search, visualization, and getting additional information. Circuit topology (CT) is the latest method proposed for describing topology in proteins and nucleic acids (Mashaghi *et al.*, 2014). The idea behind CT is that the arrangement of a linear polymer’s intra-molecular contacts (like disulfide bonds) determines its topology. The fundamental rules that intra-chain contacts must obey have been identified and CT can now be used to determine structural equivalence, study the topological structure of genomes and even protein folding. The latest approach for automatically generating protein structure cartoons is Pro-origami (Stivala *et al.*, 2011). Arrows and cylinders are used to depict β -strands and α -helices, respectively, and are drawn proportional in length to the number of residues (Fig. 1(c)). The Pro-origami web server (see “Relevant Websites section”) allows diagrams to be generated from any PDB file.

3D Structure Visualization

Different molecular visualization (MV) software have provided different levels of flexibility to visualize or perform analytical computations. However, one common feature of these programs is that the structural information contained in a PDB file is translated into the position of atoms in 3D space. This positional information when combined with other chemical information like bond connectivity, atom types, atomic radius, electric charges, electron density maps, molecular surfaces, hydrophobicity scales and so on, helps a user observe molecular structure and various other molecular features (Andrei *et al.*, 2012). When not available, the bond connectivity for each atom is usually determined through a nearest-neighbor distance search (Humphrey *et al.*, 1996). Other dimensions of molecular data such as scalar (e.g., charge, hydrophobicity), vector (e.g., force fields, electric field) or tensor (e.g., anisotropic temperature factors) may be spatially located at isolated points in two dimensional (2D) or 3D spaces and may change with time (Duncan *et al.*, 1995). The translation of all this structural data into a geometric representation is a key element of the visualization process. These geometric representations usually are depicted using spheres, lines, polygons, tubes, cones and ellipsoids. Atoms in a molecule are drawn using (solid or dotted) spheres under space-filling (CPK) and variable ball-and-stick representations. These atomic spheres have a radius equal to the van der Waals radius of the depicted atom. Bonds can be drawn either as lines or as cylinders with variable, user selectable radii. Other display attributes such as color, shininess, and opacity can be set independently for each item. The secondary structure elements of a molecule are usually represented using ribbons, licorice, tube, or cartoon representations. Molecular surfaces can also be drawn for the molecules.

Advanced molecular visualization: Besides the basic MV application, most of the MV software can also calculate and represent advanced surface features such as electrostatic potential and hydrophathy using field lines and/or as ranges of colors. The electrostatic potential is usually calculated with algorithms like APBS or DelPhi (Baker *et al.*, 2001; Rocchia *et al.*, 2001) whereas the hydrophathy is calculated using Kyte-Doolittle scale (Kyte and Doolittle, 1982). These basic and advanced components are drawn and displayed as a scene by the MV software. This generated scene can then be output as an image file for storage or publication. The visual quality of a scene can be further enhanced by a technique known as ray-tracing. In ray-tracing, a path is traced from an imaginary eye through each pixel in a virtual screen and the effects of its encounters with virtual objects are simulated. Some examples of algorithms used to ray-trace are POV-Ray and Rayshade. A very high degree of visual realism is achieved using these techniques but at a greater computational cost. The visual experience is further enhanced with additional features like transparency. Transparency is introduced using a variable known as the alpha (α) channel. A concept closely tied to the transparency of an object is the notion that the rendered appearance of that object contributes only partially to the net color assigned to pixels it occludes. Another technique used to enhance the visual appearance of a scene is known as “antialiasing”. The term antialiasing is applied to techniques that reduce the jaggedness of lines and edges in a raster image. Combined together, these algorithms and approaches generate a highly appealing and detailed image of a molecular structure.

Representations

Various ways of representing the constituent atoms of a molecule have been used. Each of them have different advantages and disadvantages. Here we describe some details about the most common representations used by various molecular visualizations software. And, to create a visual distinction between various representations, we depict the structure of insulin, 2HIU (Hua *et al.*, 1995), using VMD (Humphrey *et al.*, 1996) in Fig. 2.

Points: In this representation, each atom is drawn as a point. The bonds are not drawn at all.

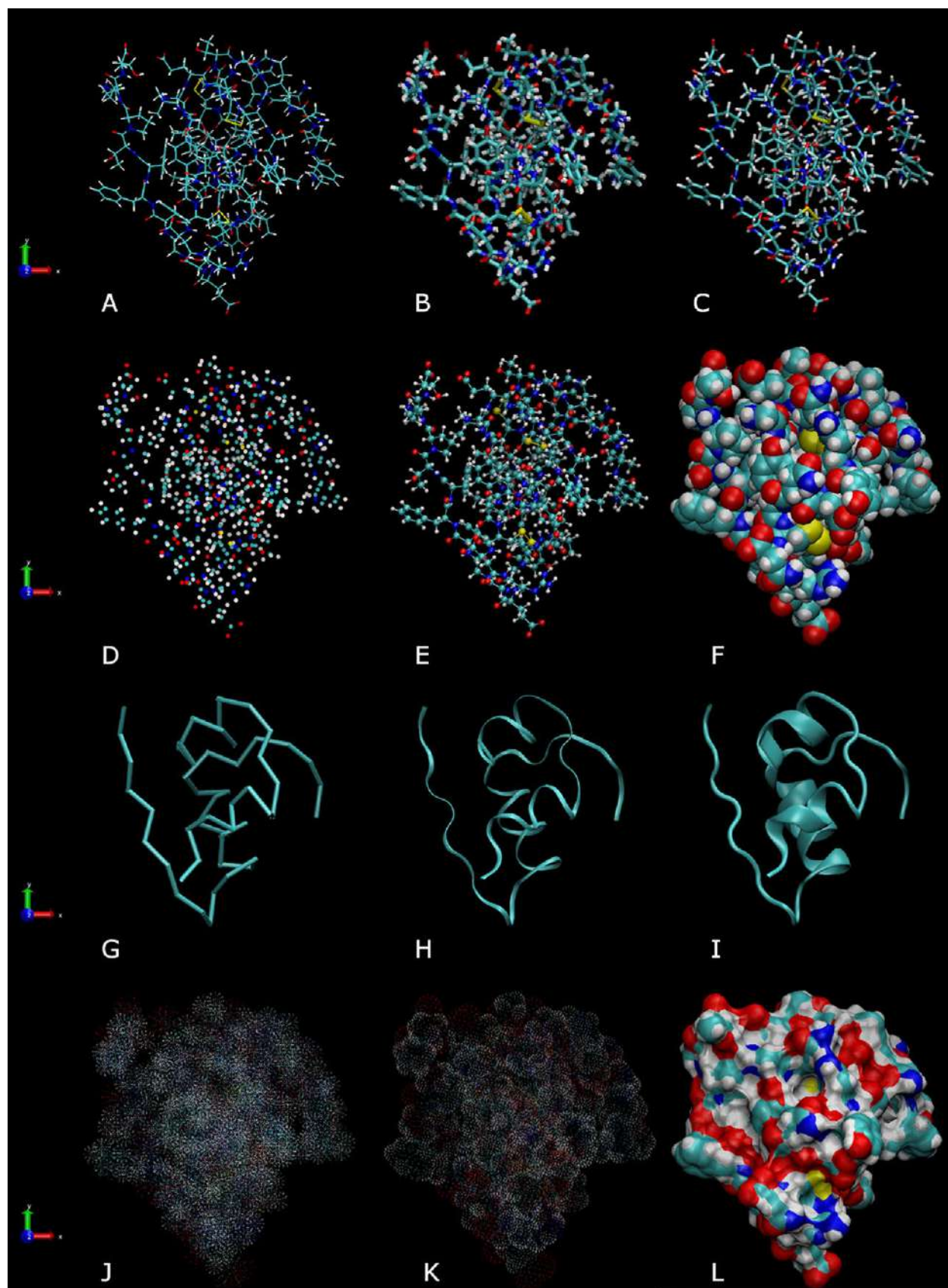


Fig. 2 Various molecular representations used to display molecules especially proteins and nucleic acids. Structure of insulin (2HIU) has been used to depict various representations using VMD. Twelve representations have been shown namely (a) lines (b) bonds (c) licorice or Drieden's structure (d) points (e) CPK (f) van der Waals (g) backbone or trace (h) ribbons (i) cartoon (j) dotted (k) solvent and (l) surface.

Lines: In the 'Lines' (or 'wireframe') representation, a line is drawn between each of the bonded atoms. The first half of each bond is colored according to the color of the first atom, while the second half is colored according to the second atom. The user can adjust the line thickness.

Bonds: In this representation, a cylinder is drawn between two atoms instead of a line. This cylinder is drawn as an n -sided prism, where the number of sides and the radius can be specified.

Backbone (or Trace): In this representation, successive residues of unbroken chains are connected by cylinders with adjustable width, in a fashion similar to "Tube". Only the main backbone atoms, $C\alpha$ atoms in case of proteins and $C4'/C3'$ atoms for RNA/DNA, respectively, are considered.

Ball and stick: In this representation, the atoms are displayed as spheres (or balls) whereas the bonds are shown as cylinders (sticks). It is the most fundamental and common representation used where the atoms and bonds are colored according to atom type. The original function of physical ball and stick models were the support of measurements of structure angles and bonds lengths, leaving the real structure representation to space filling models.

VDW: In this representation, the atoms are drawn as spheres with a radius equal to the *van der Waals* radius. The spheres are built using many polygons.

Dotted: This representation is same as VDW except that the spheres are drawn as dotted instead of solid. Like mentioned above, spheres are built using many polygons. Here, a dot is placed at each of the vertices of the triangle making up each sphere. It can be used to speculate about the surface of the molecule.

CPK: This scheme is named after chemists Robert Corey, Linus Pauling and Walter Koltun. This representation draws the atoms as spheres and the bonds as cylinders. The radius of the sphere drawn in CPK mode is usually smaller than VDW mode. Therefore, it can be called as a combination of both *Bonds* and VDW. It is also called as the space filling model.

Licorice or Dreiding model: In this representation, the atoms and bonds are drawn as spheres and cylinders, respectively. The sphere radius is not controllable and is same size as the bond. Therefore, it contains less distractors compared to the normal ball-and-stick.

Tube: The tube representation is a smooth spline curve that passes through backbone atom of proteins ($C\alpha$) or nucleic acids (P atoms of phosphates). The spline radius and resolution can be set by the user. The curve connecting the two $C\alpha$ atoms is broken into six line segments by determining five evenly spaced interpolation points along the spline curve. The first three segments are colored by the color assigned to first $C\alpha$ and the last three segments are colored by the color of the second $C\alpha$.

Ribbon: This representation also follows the same spline curve for both the protein and nucleic acids as in Tube. Additionally, it uses the O atom of the protein backbone or some of the phosphate oxygens for nucleic acids to find a normal for drawing the oriented ribbon. Given the coordinates of each atom and the offset vector for the ribbon vector, the drawing code finds the spline curves for the top and bottom of the ribbon. The two splines are connected by triangles and both splines are drawn as small tubes.

Cartoon: This representation is based on the secondary structure of the protein. Helices are drawn as cylinders, beta sheets as solid ribbons, and all other structures (coils and turns) as a tube. A least squares linear fit along at least three $C\alpha$ atoms of the helix is used to construct a helix cylinder. The solid beta ribbon is constructed by building a spline along the center points between each beta sheet residue.

Solvent: This representation gives a quick estimate of the molecular surface with a collection of dots. It is similar to the *Dotted* representation and provides a more uniform coverage of the surface. The "probe radius" and "density of the dots" can be set or adjusted by the user.

Surface: This representation uses a molecular surface renderer for example MSMS which allows to compute very efficiently triangulations of solvent excluded surfaces. Representation of molecular surfaces needs a special detailed description. Therefore, below we have provided some details on the rendering of molecular surfaces.

Molecular Surface

Molecular surface (MS) is one of the most important geometric feature of a protein as the size and shape of indentations in the surface play an important role in various functions like protein folding, docking and interactions between proteins (Connolly, 1983). The computation of MS area is important for the understanding of interactions of protein surfaces and internal cavities with ligands and hence crucial in structure-based drug design. It also allows one to incorporate the effects of solvent in the potential energy calculations (Juba and Varshney, 2008). Molecular surfaces are most usually displayed as dot-surfaces or solid renderings. Various algorithms have been developed for molecular surface area computation for example Connolly's (Connolly, 1983), MSMS (Sanner *et al.*, 1996), GETAREA (Lesk and Hardman, 1982), LSMS (Humphrey *et al.*, 1996), 3V (Pettersen *et al.*, 2004), and an adaptive grid-based algorithm (Schrodinger, 2015) included in TexMol (Kabsch and Sander, 1983). Some software that have been used for surface depiction include Grasp, AVS, VOIDOO, etc.

Broadly, three types of molecular surface models have been described which are van der Waals (vdW) surface, solvent-accessible-surface (SAS), and solvent excluded surface (SES) (Deanda and Pearlman, 2002). Ligand excluded surface is an extension of the SES concept. Below these surfaces have been described.

Van der Waals surface: This surface is a reasonable approximation of the molecular surface. Here the atomic radius of each atom is its van der Waals radius and each atom is represented as a hard sphere. Therefore, the van der Waals surface of the molecule can be defined as the union of all portions of all atomic sphere surfaces not occluded by neighboring atomic spheres.

Solvent accessible surface (SAS): Atoms located in narrow crevices may not be accessible to the solvent if the solvent molecule is bigger than the crevice. Therefore, SAS is calculated by rolling a probe sphere of a defined radius (e.g., 1.50 Å for a water molecule)

over the entire van der Waals surface of a molecule of interest. The molecular surface created by the center of the probe is called as SAS (Lee and Richards, 1971).

Solvent excluded surface (SES): SES is also calculated by rolling a probe sphere of a defined radius over the entire van der Waals surface of a molecule. However unlike SAS, SES is defined as the surface traced out by the inward-facing (towards the macro-molecule) surface of a probe sphere (Richards, 1977; Greer and Bush, 1978). In other words, SES resembles the van der Waals surface of the molecule except that crevices too small for the probe sphere to enter are eliminated whereas the clefts between atoms are smoothed over. SES area provides a better model for describing hydration effects and a better choice to visualize and study molecular properties. Surface computation is easier for the vdW and SAS but not for the SES.

Ligand excluded surface (LES): LES represents a more accurate approximation of the regions accessible to the ligand under consideration than SES because here the probe sphere is replaced with the full vdWs geometry of the ligand.

Other advanced approaches for surface representation: Voronoi diagram of atoms, is an approach that is independent of the probe size therefore invariable for a given molecule, have also been used to calculate MS of a molecule (Ryu *et al.*, 2007). With advancements in GPU hardware, several alternative approaches have been proposed for faster rendering of molecular surfaces like ambient occlusion, halos, GPU ray-casting, improved ball-and-stick representation called “HyperBalls”, theory of implicit surfaces, depth peeling, tessellation shaders, and dynamic view-dependent level-of-detail (LOD) representation have been proposed (Guo *et al.*, 2015).

Ray Tracing

Shaded images for displaying molecules have been a popular approach for conveying 3D information especially in the absence of animation. Therefore, algorithms have been developed that can handle reflections, transparency with refraction, shadows, intensity depth cuing, texturing and multiple local light sources. Ray-tracing algorithms provide the best solution to these aspects and are capable of generating realistic visual effects that are difficult with other rendering techniques (Palmer *et al.*, 1989). During ray tracing, a ray is fired from the eye point through each pixel in the image where it can either get reflected or transmitted through the surface (refraction). This reflected or transmitted ray can further interact with other surfaces, until it gets terminated after either leaving the scene or being absorbed. Ray tracing is capable of simulating motion blur, soft shadows, depth of field, diffuse inter-reflections and caustics by using various extensions.

Ray tracing in a scene requires a substantial additional computational cost compared to a non-ray-traced scene which is a significant problem. Nonetheless, ray tracing amplifies the clarity of a structure enormously. Fig. 3 compares a non-ray-traced image of insulin (2HIU (Hua *et al.*, 1995)) with a ray-traced version using the *spheres* (van der Waals) representation using PyMOL (Brunger and Wells, 2009). Ray tracing is an effective and flexible technique for producing high-quality visualizations of molecular structures especially for scientific publications. Persistence-of-vision (POV) ray (see “Relevant Websites section”), RADIANCE (see “Relevant Websites section”) and OSPRay (see “Relevant Websites section”) are some of the many ray tracing software available online.

Depth Perception

Depth perception in 2D is conveyed by motion cues during rotation and (directional) lighting. Surfaces where light bounces off the surface and is reflected toward the viewer appear bright and give a strong sense of depth for smooth, slowly varying surfaces. Lighting

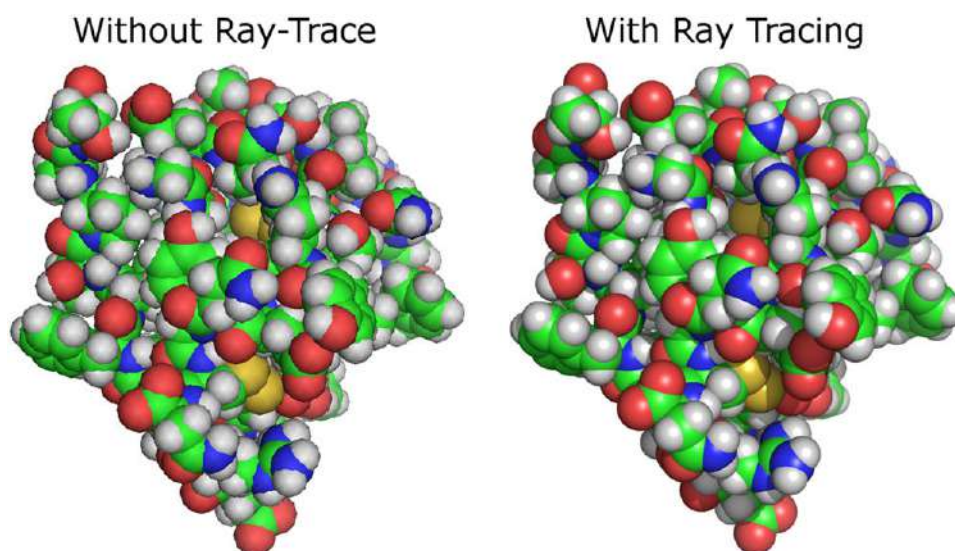


Fig. 3 The advantage of ray tracing in amplifying the clarity of a structure has been depicted using (a) non ray-traced and (b) ray-traced images of insulin (2HIU) in *spheres* (van der Waals) representation using PyMOL.

techniques like ambient occlusion are computationally demanding but appear highly effective for such space-filling sphere rendering as surface regions in recesses or concavities appear darker. Fig. 4 depicts how PyMol overlays fog on objects to assist in emphasizing what is in the foreground and background of the image with respect to the camera. Another technique to enhance 3D appearance to highlight protrusions of a surface overlaying an identically colored surface is to add black outlines at edges where a jump in depth occurs in an image. An improvement to this algorithm was introduced to make the line thickness depend on the magnitude of the depth change so that larger depth changes are highlighted with thicker lines while small depth jumps may not be highlighted to simplify appearance (Goddard and Ferrin, 2007). Similarly, eliminating depth is another alternative approach that is getting attention. For larger assemblies, the individual residues appear like countries on a map divided by boundary lines with text labels. Recent graphics processing units (GPUs) frequently have an order of magnitude more floating-point computation speed than the main “computer processing unit”. Some GPU-based techniques called textural and procedural impostors enable rendering of atomic models faster by a factor of 10 or more. Therefore, new effects such as the depth-dependent edge highlighting are possible.

Visualizing Molecular Dynamics Trajectories

Molecular dynamics (MD) studies of the function of the structurally known biopolymers are being widely applied. The results of such studies are typically large molecular trajectory files consisting of hundreds or thousands of frames of atomic coordinates. The trajectories contain substantial amounts of dynamical data that require suitable visualization tools to display sequences of structures (Humphrey *et al.*, 1996). Therefore, programs were developed that could display the molecular structure in motion. The investigator may relate various thermodynamic quantities better to molecular motions by the complementarity provided by molecular visualization. Indeed, by visualizing a computational model it becomes easier to have a subjective assessment or make a non-quantitative judgment.

Visualization of Molecular Assemblies

The hierarchy of organization of atoms in a biomolecule may start from atoms and go to secondary, tertiary and quaternary structures. For certain systems this hierarchy may even be followed by further organization into molecular complexes such as virus

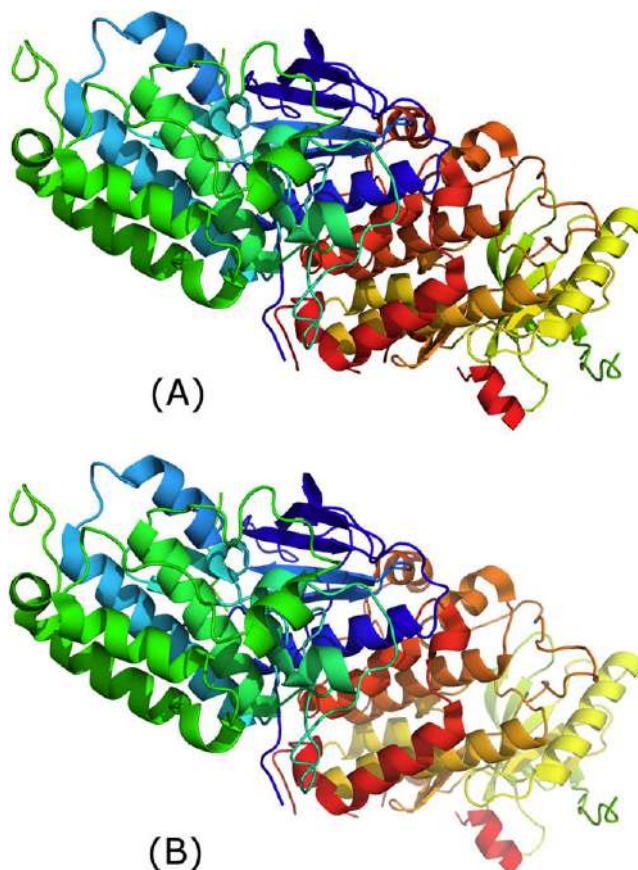


Fig. 4 The perception of depth in a scene is depicted using the structure of EFG receptor (2JIT) in the *cartoon* representation using PyMOL. (a) Without depth cueing and (b) with depth cueing. The obscurity and haziness of the distant domain of EGFR clearly depicts the impact of depth cueing.

capsids. The visualization software would be required to render millions to billions of atoms in real time. Therefore, visualization of larger biological structures needs a humongous computational power and is still a challenging task. To circumvent this problem, lesser details for each component of the larger assemblies are used to decrease the computational burden (Goddard and Ferrin, 2007). The approximate shape of domains, molecules, or complexes are drawn as smooth surfaces at any desired resolution using methods like spherical harmonics, density contours, and surface decimation. Each method has its own advantage and disadvantage but discussing that is beyond the scope of this article.

Evolution of Molecular Visualization Software

A number of molecular visualization programs have been developed since the days of first molecular visualization using computers. These programs use component modules each specifically written to create the structural components like spheres, tubes, ribbons, etc. Not all the programs can be described in this article but we try to describe the best known molecular visualization programs. The programs are arranged according to their year of release to generate a sense of evolution in terms of molecular visualization. Each program is listed and the capabilities and mode of action of the underlying algorithm(s) have been described.

RIBBONS (1987)

Mike Carson constructed an algorithm that generated aesthetically pleasing and smooth 3D representations of a protein using a set of nearly parallel B-spline curves fitted to the peptide plane (Carson, 1987). The peptide planes were used to generate a series of 'guide coordinates' equally spaced along the desired ribbon width. Smooth and regular cubic B-spline curves fitted to these guide coordinates formed the roughly parallel threads of the ribbon. Ribbon models clearly displayed the secondary structure of the backbone. It was also possible to code by residue to depict information such as residue type or temperature factors.

MAGE (1992)

A "kinemage" or kinetic image was a term designed for a structure presented on a computer display (Richardson and Richardson, 1992). A utility called PREKIN was used to prepare kinemage files from input PDB files which were then viewed using a graphics program called MAGE. The kinemages were plain text files with commented display lists and accompanying explanations that could be edited by user. The interactive image could be rotated in real time, parts of the display could be turned on or off, cursor selection of atoms or points was possible, and different forms could be changed. The depth-cueing, rotation, rocking, distance measurement and animation of two or more conformations was possible.

RasMol (1993)

RasMol (stands for *raster molecules*) is an easy to use molecular visualization tool for proteins, nucleic acids and small molecules (Sayle and Milner-White, 1995). It was developed by Roger A. Sayle (*also forms R.A.S. of RasMol*) for Unix, Windows and Macintoshes and is available freely from the websites provided in "Relevant Websites section". Rasmol can read a PDB file both locally or directly from PDB database, which can then be displayed interactively with a variety of color schemes and molecule representations including depth-cued wireframes, 'Dreiding' sticks, spacefilling (CPK) spheres, ball and stick, solid and strand biomolecular ribbons, hydrogen bonding and dot surfaces (Sayle and Milner-White, 1995). The users could interact with a molecule by rotating, translating, zooming and z-clipping (slabbed) using either the mouse, scroll bars, command line or an attached dial box. Ligands, active sites, multiple subunits, hydrogen bonds and various parts of the molecule could be displayed selectively or in a combination of display modes. It could measure interactive distances, angles and torsion angles. RasMol allowed depiction of stereo images, multiple NMR models and labeling of atoms. The structural features like β -sheets, protein backbones, double and triple bonds were represented as arrows, smoothed-tube, multiple lines and, respectively. RasMol can also take a list of commands using a 'script' file to (re)generate a given image or viewpoint that may be written out in a variety of formats, e.g., PostScript, GIF, BMP, etc. No doubt, RasMol quickly became the most popular molecular graphics software with over 15,000 sites using it.

AVS (1995)

AVS or Advanced Visual Systems, was designed to be a general-purpose visualization environment with no elaborate data structures for specific applications areas (Duncan *et al.*, 1995). Several programs, e.g., AutoDock, Harmony and SURFDock use AVS as an important visualization tool as the rendering capability of AVS is of very high quality. AVS uses several modules to translate molecular coordinate data to a geometric representation, e.g., SPHERES (for spheres), DISJLINES (for bonds), POLYLINE, POLYGON (to define triangles, and other convex polygons), POLYTUBE (a cylindrical object from a polyline), ELBOW (for smooth joints), COLORTUBE (produces a tube from disjoint line), MSMS (for molecular surfaces), MV102 (to control the display style of selected subset of atoms). Intermolecular interfaces can be calculated using the INTERFACE module. The PLAY BINPOS

module displays MD simulation trajectories with facilities for selecting individual frames and controlling the speed of the display. Both, a graphical user interface (GUI) and a command line interpreter (CLI) can be used to control most interactions with AVS. AVStk, a Tcl/Tk interface to AVS implements the facilities for expressions, variable assignment, loop control, or functions. The NAB (nucleic acid builder) language is used for building chemistry modules as it has data structures and operations designed specifically for molecular data. AVS is suitable for biomolecular visualization as it provided general-purpose and chemistry-specific modules, and AVS-tool and NAB to facilitate network control and module development.

Raster3D (1996)

Raster3D is a hardware independent suite of programs used to generate photorealistic molecular graphics (Merritt and Bacon, 1997). It is available for various Windows, various linux distributions and macintosh operating systems. It offered several advantages compared to other available programs. For example, it provided platform-independent tools for composition and rendering. It was much faster than general-purpose ray-tracing programs. Raster3D suite has four different types of programs namely, composition tools (*balls*, *ribbon*, and *rods*), input conversion utilities, the central rendering program, and output conversion filters. The composition tools read the atomic coordinates found in a PDB formatted file and generate van der Waals surface (*balls*), a peptide backbone trace (*ribbons*) and a ball-and-stick model (*rods*) of the bonded atoms. This information is stored or converted into a series of header lines specifying global rendering options, followed by a series of individual object descriptors. The input stream can have further input from one or more files. Also, the size of the rendered image in pixels could be controlled using header records input to the render program. The main program of the Raster3D suite, *render*, takes this input and produces a raster image in one of three standard formats (AVS, TIFF, or SGI libimage). Rendered images could be processed further to add labels, edit colors, form composite images, apply “gamma correction” or be converted to other formats. Raster3D could also import objects from other molecular visualization tools using two conversion utilities *ungrasp* and *normal3d*.

Raster3D generated *shadows* to convey an impression of depth, obviate the need for stereo pairs and to convey information that otherwise would not be apparent in the rendered image (Merritt and Bacon, 1997). Raster3D implemented “Z buffer” algorithms rather than true ray tracing to produce a remarkably photorealistic appearance for the rendered objects. The rendering algorithm can be optimized to output images that include effects like shading, transparency and effective transparency (or α channel). It required much less computation than full ray tracing of the same scene. One can generate various kinds of pictures in Raster3d such as black and white, colored, and animated images. We can also add labels, shadows, transparency and stereo images. Ribbons, ring, surface and rods are the common kinds of representations available. It has been implemented in other protein visualization packages such as VMD, Xtalview, Molscript, GRASP, MSMS, ORTEX, Conscript, etc.

MD Display (1996)

MD Display was developed initially as a means of visualizing molecular dynamic trajectories generated by Amber (Callahan et al., 1996). The program runs on Silicon Graphics workstations, and features a simple user interface, and convenient display and analysis options. The program can accept input from several other molecular dynamics programs. A preprocessor handles output files from different computational programs (like Amber, CHARMM, Discover, and GROMOS) and to generate a consistent input for the display program. It can load PDB files and allows construction of a pseudo-trajectory using multiple PDB files. It provides animation control in terms of speed (and *hyperspeed*) and direction of animation so that freezing one frame and single step in either direction is possible in the frame sequence. Periodic boundary conditions are handled and multiple frames can also be superimposed to create a “smear” image. The user can monitor the interatomic distances, angles, dihedrals, hydrogen bonds (based on donor-acceptor distances), and Ramachandran plot (ϕ - ψ values) in the trajectory.

MOLMOL (1996)

MOLMOL is developed specifically for the display, analysis, and manipulation of sets of conformers in nuclear magnetic resonance (NMR) structures of proteins and nucleic acids (Koradi et al., 1996). It is downloadable from the website provided in “Relevant Websites section”. It also implemented special functions to represent the structural uncertainty by the spatial spread among groups of NMR conformers. It could be run on Silicon Graphics workstations, unix machines using either OpenGL or X11 library. The selection of sets of atoms, bonds, distances, or primitives was possible by expression syntax or mouse. This selected set could be represented as ribbons, ellipsoids, or other simple geometric shapes. The secondary structures were either identified using the DSSP algorithm or read directly from the PDB file. MOLMOL can draw dots or shaded molecular surfaces for three different kinds of surfaces namely van der Waals surfaces, solvent accessible surfaces and contact (SES) surfaces.

This program was unique as it allowed modification of chemical structures by adding and deleting atoms and bonds, and generation of new 3D structures by variation of dihedral angles about individual covalent bonds. It could also analyze the trajectories from MD simulations. For quantitative comparisons, the program calculated root mean square distances (RMSDs), short interatomic contacts, dihedral angles, hydrogen bonds, radius of gyration and solvent-accessible surfaces. The program could generate high-quality plots as PostScript, or FrameMaker (MIF), TIFF format or as an input file for ray-tracing programs.

Cn3D (1996)

Cn3D (pronounced ‘see in threedee’) is a visualization tool developed by NCBI to simultaneously display biomolecular structures, sequences, and sequence alignments for all platforms (Hogue, 1997; Wang *et al.*, 2000). It provided all the functionality of RASMOL along with other enhanced rendering and labeling options. It could also play animations for multiple conformations of NMR structures or multiple superposed structures. To provide a better understanding of the 3D structure, by default the protein’s α -carbon trace is colored by secondary structure and α -helices are rendered as hollow cylinders whereas a DNA molecule is colored by CPK color scheme and metal ions are shown as spheres. It is also possible to ‘hide’ chains, domains or residues at one’s discretion. It is also easy to set custom rendering or labeling for a set of atoms or residues. It offers a programming layer as MMDB-API along with a GUI called Viewer3D. The co-ordinates data from a PDB file is translated into MMDB-API representation and converted to graphics primitives (like spheres, cylinders, lines, etc.) to be displayed in Viewer3D.

Cn3D includes algorithms to import and align multiple sequences or structures. Homologous sequences to a protein with known 3D structure can be imported and aligned into the sequence viewer. After importing multiple homologous sequences, the structure window displays the 3D superposition while the sequence window shows a sequence view of the structural alignment. A seamless combination of structure visualization and comparative analysis makes Cn3D a powerful tool to map functional sites between structure and sequence. This proves helpful in finding conserved regions among aligned proteins, locating the residues corresponding to known disease mutations in a crystal structure or mapping the locations of conserved binding site or active site residues from a family of homologous sequences. Because of these capabilities, Cn3D is the default structure viewer for NCBI’s Entrez Structure database and the primary alignment curation tool for the CDD project (Wang *et al.*, 2000). Cn3D also has powerful annotation and alignment editing features.

VMD (1996)

VMD, stands for Visual Molecular Dynamics, is a tool written in C++ and designed to visualize structural features of various biopolymers like proteins and nucleic acids using a wide variety of rendering styles and coloring methods (Humphrey *et al.*, 1996). The source code and documentation for VMD is available at the website provided in “Relevant Websites section”. VMD can display standard PDB files and animate trajectories obtained from molecular dynamics (MD) simulations. The trajectories can be loaded both from stored files and from a direct connection to a MD simulation running on a remote supercomputer or high-performance workstation. For remote trajectory sources, VMD uses a set of daemons and library routines known as the MDCOMM that buffer data transfer from the remote connection. A portable MD simulation program called NAMD, available with VMD, has been designed for performance, scalability and modularity to take advantage of the parallel architecture of processors. VMD together with NAMD constituted a larger set of computational tools for structural biology known as MDScope (Nelson *et al.*, 1995).

VMD requires Silicon Graphics GL library or the OpenGL library for 3D graphics rendering. The GUI can be used to perform tasks such as changing the current molecular display characteristics or animating selected molecules using molecular dynamics trajectories. All actions in VMD are available *via* text commands so that users can write and execute their own *tcl* scripts. To read data from file formats other than PDB, VMD has an interface to Babel (Walters and Stahl, 1996) to read data from other formats. Molecules are drawn as one or more “representations” for all or a subset of constituent atoms. A particular set of atoms can be selected using Boolean operators and regular expressions and assigned one of the various rendering styles and coloring schemes. The solid objects are illuminated using up to four hardware-accelerated, independent, infinitely distant light sources. VMD provides an extensive program control, using both graphical user interface (GUI) and a text interface using the *Tcl* embeddable parser, to generate high-resolution raster images for photorealistic image-rendering applications.

Trajectory display and analysis: As the name suggests, VMD was primarily developed with the ability to animate and study molecular dynamics (MD) simulations. A molecular trajectory may be read (from PDB files or from direct connection) and edited (using a trajectory editor) that provides options to delete or write (*to a new file*) specific frames or set of frames. VMD can also perform complex analysis on a molecular dynamics trajectory such as computing the RMS deviation or correlation functions. With the advent of petascale computers (e.g., Blue Waters supercomputer) and GPU-accelerated ray tracing engine, atomic-detail simulation and visualization of really large cellular apparatus and processes like the 100 M-atom photosynthetic membrane complex in purple bacteria, is feasible with VMD (Stone *et al.*, 2016).

WebMol (1997)

With the advent of Java and web-browsers with in-built support for Java applications, programs could be run virtually and remotely using regular web-browsers. The client computer system doesn’t need to install the programs code (‘program once - run anywhere’) because Java programs are directly transferred and interpreted to the client upon request. One such application for molecular visualization, called WebMol, was developed at EMBL.

WebMol is a Java based interactive graphical program to display and analyze molecular structures stored locally or remotely in the PDB database using Java-supporting web-browsers only (Walther, 1997). It provided a similar functionality online as RasMol provided offline but suffered slow rendering and java-dependent security problems. The main advantage was that novice users didn’t require to install programs or their source codes. They could directly go to WebMol and start visualizing or analyzing molecular structures. A new version of WebMol (see “Relevant Websites section”) is under development in HTML5 and Javascript

with the support of PhiloGL. The application is designed to be accessible from Internet and compatible with the JSON format of protein provided by the project WebPdb (see “Relevant Websites section”).

PyMOL (1999)

PyMOL is an elegant open source and cross-platform program for visualization of complex macromolecular systems based on OpenGL and Python (Schrodinger, 2015). PyMOL can be downloaded freely for common platforms like Windows, Linux, and Mac OSX from the website provided in “Relevant Websites section” (v1.8) or from the website provided in “Relevant Websites section” (v2.0). Along with a graphical user interface (GUI), PyMOL offers a command line mode that resembles the Python interpreter. In fact, ‘Py’ part in PyMOL refers to the programming language Python. PyMOL commands are a series of Python function calls that function as a superset of the Python language. It also supports reusable scripts written in the PyMOL command language or in Python. PyMOL supports most of the common representations like wire, cylinders, backbone and cartoon ribbons, spheres, ball-and-stick, dot surfaces, solid surfaces, wire mesh surfaces. It supports labels, dashed bonds (for hydrogen bonding interactions and distances), transparent surfaces, reads and displays CCP4 and XPLOR electron density map files, generate symmetry-related molecules and load multiple common file formats. PyMOL was the first intuitive molecular graphics program that provided the click-and-drag functionality that we now see in almost all the molecular visualization software, no doubt it became really popular (Brunger and Wells, 2009). PyMOL was originally designed to dynamically visualize single or multiple conformations (trajectories or conformational ensembles) of a single structure with professional strength graphics. It has an integrated ray-tracing engine (in addition to PovRay engine) for generating publication quality figures that are complete with lighting, specular reflections, and shadows. Fig. 5 depicts some of the representations available in PyMOL for the same insulin structure (2HIU (Hua *et al.*, 1995)). Indeed, figures generated with PyMol appear superior than many other software. This obviated the need for command-line operations which were otherwise required by other programs such as Molscript, Raster3D, and ImageMagick. Different states of molecule can be assembled into QuickTime or AVI movie by rendering simple or Ray-traced frames. Multiple atom selection was made possible using arbitrary extended algebraic expressions. Molecular editing allowed users to create new objects out of atom selections (across any number of other objects) and edit (delete, replace, or grow) bonds, angles, torsions, and positions on an atomic basis using just click-and-drag operation. Another important aspect of PyMOL is that other applications can utilize it as a molecular display window while providing their own external menus, windows, dialog boxes, and controls or even add additional geometries. PyMOL can also display the results of the macromolecular electrostatics calculations by APBS (Baker *et al.*, 2001) as an electrostatic potential molecular surface using the APBS Plugin. PyMOL also provides an automated qualitative electrostatic representation for a protein’s contact potential by generating a charge-smoothed surface. It performs a “charge smoothing” using a quasi-Coulombic-shaped convolution function to average the charges over a small region of space. Fig. 6 displays such an electrostatic contact potential for insulin generated using PyMOL.

UCSF Chimera (2004)

The developments in electron microscopy (EM) led to the determination of the atomic resolution structure of large-scale sub-cellular systems. Compared to other crystallographic and NMR structures, which are usually not very large, the structural complexity for large-scale molecular assemblies like viral particles, chaperonins and chromosomes, increases rapidly. Such large assemblies or molecules may contain several million atoms and therefore normal visualization software may not be able display it properly. The reason is that normal representations are so detailed that it requires impractically large computer memory and leads to insufficient graphics rendering speed (Goddard *et al.*, 2005). Therefore, to display such large molecules, better algorithms are required that can perform higher order calculations efficiently on desktop computing resources. UCSF Chimera was launched as a solution to this problem (Goddard *et al.*, 2007). Chimera is Python based freely available to academic and nonprofit users from the website provided in “Relevant Websites section”. Its GUI and command-line interfaces provide rich and overlapping sets of functionality under Windows, Mac OSX and Linux or other unix based systems. It generates a mixture of low-resolution depictions of assembly components with high resolution depiction of regions of interest to avoid the above-mentioned drawbacks. It offers interactive visualization and analysis of atomic-resolution molecular models as well as specialized capabilities for low resolution depictions, contact calculations and quaternary structure navigation to study large multimeric assemblies such as virus capsids, ribosomes, microtubules, etc. To interactively explore such large complexes, it is important to have the abilities to build multimeric forms, display low-resolution representations and define levels of structure. The widely used molecular visualization programs can also display such density maps but do not offer a broad selection of tools for studying macromolecules.

Chimera was designed from its predecessor MIDAS (Ferrin *et al.*, 1988) with extensibility as a primary goal. Therefore, the architecture of Chimera is composed of a core, for basic services and state-of-the-art visualization, and extensions that provide all extra higher level functionality (Pettersen *et al.*, 2004). The C++ based functions in “core” can perform molecular file input/output, graphical display as representations (e.g., wire-frame, ball-and-stick, ribbon, and sphere), generation of molecular surface using MSMS algorithm, select parts of structures, control transparency, near and far clipping planes, and lenses. On the other hand, the extensions include *Multiscale*, to visualize large-scale molecular assemblies; *Collaboratory*, to share Chimera sessions remotely in real time; *Multalign Viewer*, for reading, writing and displaying multiple sequence alignments together with associated structures; *ViewDock*, facilitates interactive screening of docked ligand orientations from DOCK; *Movie*, for displaying and saving MD

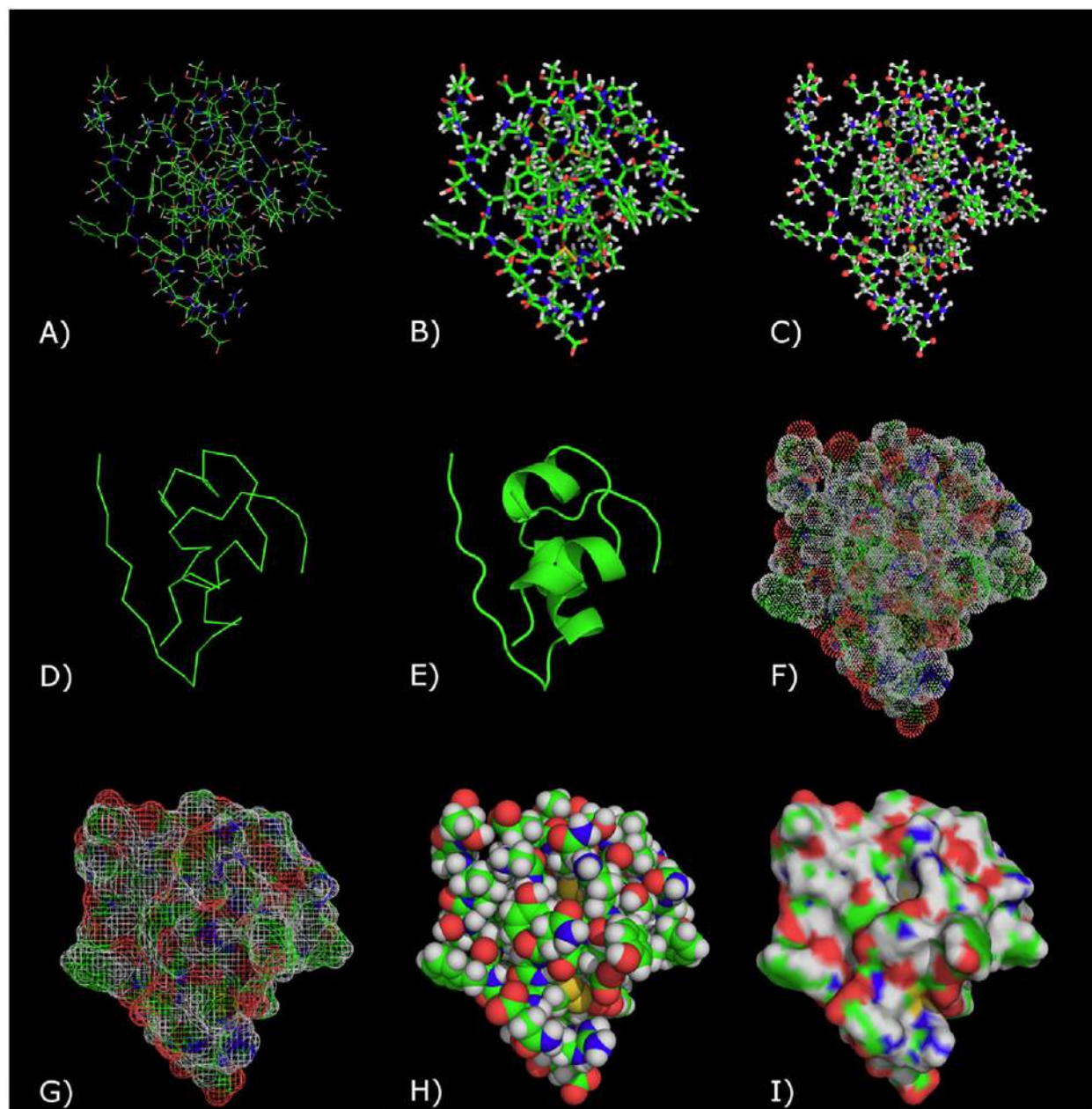


Fig. 5 Molecular representations used in PyMOL to display proteins and other types molecules. Structure of insulin (2HIU) for all these representations namely (a) lines (b) sticks (c) ball and stick (d) ribbons (e) cartoon (f) dots (g) mesh (h) spheres (i) and (j) surface.

trajectories as a video; and *Volume Viewer*, for displaying volumetric data as isosurfaces, meshes, and translucent solids. It now also integrates MODELLER and other modeling tools for structural modeling of multiprotein complexes from sequence to 3D structure (Yang *et al.*, 2012). High-quality images and animations can be generated for publication using Chimera.

Jmol (2004)

Jmol has been developed as an open source Java-based molecular visualization software by an active user community of professional crystallographers, educators and students over many years (Herraez, 2006). Initially, it was made available at the website provided in “Relevant Websites section” however currently, it is available from the website provided in “Relevant Websites section”. Jmol creates every pixel of the model on the fly, using exceptionally efficient rastering techniques and without any external 3D graphics packages. That is why Jmol runs independent of the operating system (like Linux, Windows or Macintosh OS) and the browsers. It was designed as a web-based replacement for RasMol or Chime (Rasmol as a web browser plug-in) for crystallographic visualization and analysis. Being web-based, Jmol offered possibilities of revolutionizing the way we learned

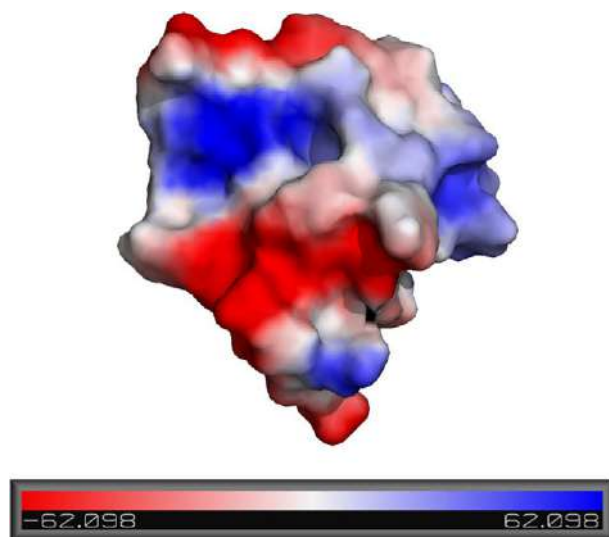


Fig. 6 A qualitative representation for insulin's (2HIU) electrostatic contact potential generated by PyMOL using a charge-smoothed surface. The "charge smoothing" is performed by averaging the charges over a small region of space.

about molecular structure (Hanson, 2010; Cass *et al.*, 2005). Jmol can read, display, manipulate, analyze, and output data from molecular structures stored in various file formats. It allowed diverse molecular representations like wireframe, sticks, balls and sticks, spheres, ribbons and cartoons. Its extensive mathematical scripting capability allows extensive analysis of molecular structure by writing functions/scripts using the popular command flow syntaxes. It can produce high-quality images in popular formats like JPG and PNG with or without point-of-view ray (POV-Ray) tracing. Jmol can also depict electron density maps and isotropic B-value ('temperature') data. Finally, it can also be used to read, load and display MD simulations trajectories.

Bioblender (2012)

Traditionally, the electrostatic and hydrophobic characteristics of proteins are visualized as range of colors that vary according to the tool used. The simultaneous visualization of both these properties has been almost impossible, except when visualized in different images. A novel and intuitive software, BioBlender, attempts to describe protein motion with simultaneous visualization of their chemical and physical features (Andrei *et al.*, 2012). It can be downloaded freely from the website provided in "Relevant Websites section" for Windows, Linux and Macintosh. It has been developed as an add-on to an open-source, free, and cross-platform application called as Blender that can access several scientific programs. BioBlender uses Python based scripts for building the interface, various calculations and managing other component programs like PyMOL (for loading PDB files, molecular surface calculations), PDB2PQR (for continuum electrostatics calculations), APBS (for calculating the electrostatic potential), and Blender (for 3D graphics). Blender itself has many benefits, one of them is that physics-based animations can be achieved by simulating forces such as gravity, magnetic, vortex, wind, etc.

BioBlender has introduced simultaneous representation of surface physico-chemical properties of proteins, even in motion. Using features different from color permits their simultaneous delivery in photo-realistic images leaving the utilization of color space for the description of other biochemical information. The new visual code introduced in BioBlender entails the depiction of molecular lipophilic potential (MLP) as a range of optical features going from smooth-shiny for hydrophobic regions to rough-dull for hydrophilic ones. Similarly, animated line particles are used for electrostatic potential (EP) that flow along field lines, proportional to the total charge of the protein (Andrei *et al.*, 2012). A python script named "*pyMLP.py*" is used to calculate the MLP for the protein space. Finally, a custom software "*scivis.exe*" is used to calculate field lines. Therefore, the outcome is the ability to visualize the molecular surface, the EP grid, the gradient grid and the field lines. A continuous perception and visualization of molecular features can be generated by the program using various conformations of macromolecules during their motion. For a protein in motion or with multiple conformations, BioBlender uses Blender Game Engine (BGE). BGE is equipped with special rules to simulate atomic behavior, to interpolate between known conformations and obtain a physically plausible sequence of intermediate conformations. The program gives as output both the intermediate PDB files and the rendered images as a movie. This approach makes nanoscale "un-seen" phenomena such as hydrophathy or charges of proteins become more understandable and familiar to our everyday life by drawing viewer's attention to the most active regions of the protein.

JSmol (2013)

JSmol is a JavaScript-only version of Jmol (Hanson, 2010) with the complete set of Jmol functionalities (Hanson *et al.*, 2013). Being Java-based has its own demerits in terms of security. Therefore, Java does not run on some handheld devices but JavaScript

(JS) does. Therefore, Jmol has been completely rewritten in JavaScript to produce identical graphical results. Thus, JSmol is the first full-featured molecular viewer based on JS, and the first ever JS-based viewer for proteins. For small molecules it performs at par with Jmol, but for larger molecules it doesn't scale as Jmol. Nonetheless, JSmol has been successfully implemented in Proteopedia (Hanson *et al.*, 2013; Prilusky *et al.*, 2011).

YASARA View (2015)

The spherical representation of an atom usually requires about 320 triangles, therefore visualizing larger proteins with atomic details requires tens of millions of triangles, far too many for smooth interactive frame rates. A new algorithm called YASARA View was designed to solve this problem (Krieger and Friend, 2014). It does not depend on high-end shader tricks and therefore works on smartphones too. Too complex objects are replaced with 'impostors', i.e., entities that have pre-calculated textures attached, which make them look like the original object. It allowed sharing the GPU and multiple CPU cores for producing high-quality scenes with perfectly round spheres, shadows and ambient lighting. For changes in the position of the light source, the texture is updated from a collection of 200 different pre-rendered views. Using this approach, YASARA View can display larger complexes on smartphones as well.

NGL Viewer (2015)

With HTML5 feature set and tremendous performance gains of JavaScript, web-browsers have now become more powerful in integrating GPU accelerated image processing, physics and other graphical effects as part of the web page canvas (Yuan *et al.*, 2017). This is due to the development and integration of a JavaScript based Web Graphics Library (WebGL) into modern compatible browsers. WebGL renders interactive 2D and 3D graphics without the use of plug-ins. NGL Viewer makes use of this capability of web browsers to interactively display molecular structures (Rose and Hildebrand, 2015). It can create rich visualization for common structural file formats like PDB, GRO or mmCIF using various molecular representations like *line*, *point*, *cartoon*, *space fill*, *tube*, *ribbon*, *licorice*, etc. In order to greatly reduce the geometric complexity in displaying large macromolecules, the image rendering includes ray-casted impostors. NGL viewer can be embedded into structural database websites like RCSB PDB (see "Relevant Websites section") to allow loading, quick depiction and subsequent manipulation of molecular structures online (Rose *et al.*, 2017). In fact, only NGL Viewer is used by PDB database for structure larger than 10,000 residues (Rose *et al.*, 2016).

3Dmol.js (2015)

3Dmol.js is a pure JavaScript, hardware-accelerated, object-oriented molecular visualization library (Rego and Koes, 2015). It has been developed as an alternative to the software-based rendering of Jmol and JSmol. The major focus is online interactive visualization for molecular structures with near native performance. It uses WebGL (*as described above for NGL viewer*) which comes preinstalled in modern browsers. Therefore, 3Dmol.js does not require any additional plugin. The users can either embed the viewer or use it through a hosted viewer using a URL. It can read various file formats like pdb, sdf, mol2, xyz and cube. Its performance is at par with Jmol, JSmol.

Web3DMol (2017)

Like NGL Viewer, besides allowing interactive displays and manipulation of 3D structures, Web3DMol provides some extra functions such as sequence plot, fragment segmentation, measure tool and meta-information display for a better understanding (Shi *et al.*, 2017). Additionally, NGL Viewer does not allow rendering different parts of a protein under different modes. Web3DMol, on the other hand, can display both the primary, secondary structures and related meta-information (e.g., molecular classification, structural resolution and experimental details) for a structure. It can render different parts of a molecule as different modes using the fragment segmentation and interactively measure distances, angles and area between a group of atoms. Another interesting feature is Web3DMol's graphical display sharing using an automatically generated *sharing-URL*. Anyone can share the complete interactive behaviors such as rotating, zooming and translating, can be easily shared with others using this URL.

Additional Insights From Structure

The growth of tools that help in online rendering of structures have boosted development of several web resources that provide more insights related to protein structures. Such resources are more helpful for undergraduate students in understanding the relationship between 3D protein structure and function (Terrell and Listenberger, 2017; Hayes, 2014). With the same goal in mind, several journals now provide interactive 3D structure visualization while others have started using interactive 3D figures. However, there is nothing better than a dedicated online server or tool or database for this purpose. Below are details on some excellent tools that provide additional insights along with structural visualization.

Aquaria

Aquaria (see “Relevant Websites section”) generates a concise visual summary of all related PDB structures for any sequence or PDB structure of interest (O’Donoghue *et al.*, 2015). It is capable of loading information regarding domains, post-translational modifications or single nucleotide polymorphisms (Fig. 7). The server generates an all-against-all sequence comparison monthly from PDB and SWISS-Prot. Using this pre-calculated data or comparison, Aquaria ranks and serves the list of structures related to the sequence loaded. Using such information, Aquaria contains a median of 35 structures per protein and no other resources can match this depth of sequence-to-structure information.

PDBsum

PDBsum is a web server capable of performing various analyses that include protein secondary structure, protein-ligand and protein-DNA interactions, structural quality, among others. It is freely available at the website provided in “Relevant Websites section”. Previously, PDBsum used static images generated using PyMOL however now it uses 3Dmol.js, RasMol, Jmol, PyMOL, and Strap to display the 3D structures interactively (Schrodinger, 2015; Laskowski *et al.*, 2018; Sayle and Milner-White, 1995; Rego and Koes, 2015; Gille and Frommel, 2001). It uses the SURFNET program (Laskowski, 1995) to calculate cleft regions which can then be visualized. PDBsum provides links to homologous proteins using links to the Sequences Annotated by Structure (SAS) server (Milburn *et al.*, 1998). Secondary structure assignments and topology diagram are computed using PROMOTIF and HERA (Hutchinson and Thornton, 1990; Hutchinson and Thornton, 1996), respectively. Interactions of a protein with other proteins, ligands, or metal ions are generated using HBPLUS and LIGPLOT (Wallace *et al.*, 1995; McDonald and Thornton, 1994). PDBsum shows information related to domain architecture from Pfam and CATH (Finn *et al.*, 2016; Sillitoe *et al.*, 2015). Pores and tunnels in a protein are calculated using Mole (Sehnal *et al.*, 2013). For enzymes, PDBsum shows a reaction diagram obtained from Kyoto Encyclopedia of Genes and Genomes (KEGG) database (Kanehisa *et al.*, 2017). A subset server of PDBsum, DrugPort, identifies all “drug targets” in the PDB using the DrugBank database (Law *et al.*, 2014) and any drug molecules that occur as ligands in PDB structures.

Proteopedia

Proteopedia (available at the website provided in “Relevant Websites section”) is a collaborative effort to generate an interactive wiki-based encyclopedia for visualizing 3D structures of protein, nucleic acid and other biomolecules (Hodis *et al.*, 2008). The structural annotation here is intuitive, interactive, and the knowledgeable users can even contribute to the annotations pages in the website (like Wikipedia; see “Relevant Websites section”). It can also be used as a pedagogic tool for teaching the relationships between biomolecular structure and function in the classroom. Initially, Jmol (Hanson, 2010; Cass *et al.*, 2005) was used to display the structures but it required users to have Java installed on their computers. Therefore, Proteopedia has now shifted to JSmol, which does not require Java to be installed. There are now more than 130,000 pages in Proteopedia which have been contributed by more than 3300 users worldwide. Users can also generate content for their scientific publications using the interactive 3D visualizations while keeping these articles absolutely private so that only they may view and edit (Prilusky *et al.*, 2011).

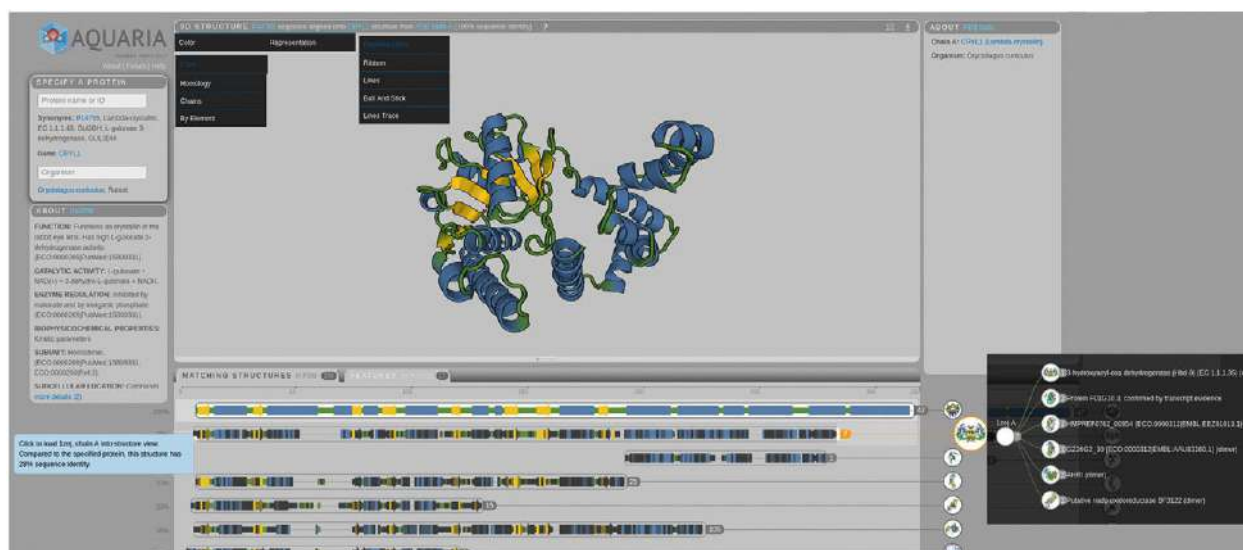


Fig. 7 A composite image generated using multiple screenshots for Aquaria tool. Different options available in the tool have been brought together to provide a general idea of the interface. We loaded the 3ADO, rabbit gulonate dehydrogenase, crystal structure from PDB as an example.

MolviZ.Org

MolviZ.org is another similar web resource that brings together molecular visualization resources. It is available freely at the website provided in “Relevant Websites section”. It has a lot of content related to molecular visualization, for example, interactive tutorials, software related to molecular exploration, an atlas that contains images and descriptions of selected macromolecules, etc.

POLYVIEW (-2D/3D/MM)

The POLYVIEW is a 2D and 3D molecular visualization server that offers a flexible annotation tool for generating protein sequence annotations, including secondary structures, relative solvent accessibilities (RSA), functional motifs and polymorphic sites (Porollo and Meller, 2007). 2D graphical representations in a customizable format may be generated for both known protein structures and predictions obtained using protein structure prediction servers like CASP. For example, one can identify putative globular soluble as well as membrane domains to make preliminary conclusions about the domain structure of a protein (Porollo *et al.*, 2004). POLYVIEW may be used for automated generation of pictures with structural and functional annotations like secondary structure (SS) states (H, E, C), RSA, hydrophobicity, polarity and charge profiles for publications. POLYVIEW-3D, is a major update of POLYVIEW. It uses PyMol (Schrodinger, 2015) for high quality rendering of models and structures (Porollo and Meller, 2007). It can also perform advanced structure and function analysis, like mapping interaction interfaces, binding pockets, and comparison and scoring of protein docking models. POLYVIEW also provides extensive cross-linking with several rigorously validated annotation and prediction servers, such as ConSurf, CASTp, ClusPro, and SPPIDER. POLYVIEW-MM (for molecular motion) enables integration of high-quality animation, e.g., trajectories generated by molecular dynamics and related simulation techniques, with structural annotation (Porollo and Meller, 2010).

Future Applications in Molecular Visualization

The molecular visualization software for viewing 3D models is improving in terms of many aspects for example, level of details, perception of depth using lighting, etc. People from various fields like experimental biologists, database developers, computer scientists, and package developers join hands to bring about these advances. The latest and most exotic computer hardware like 3D printers that create three-dimensional objects or displays that produce stereoscopic depth without special glasses and force feedback (haptic) devices that provide a sense of touch are finding applications to molecular assemblies (Nagata *et al.*, 2002; Stocks *et al.*, 2009). With force feedback technology, the user can ‘touch’ and sense the electrostatic potential field of a protein molecule and aid in various molecular modeling approaches (Haptic-driven, 2012). A globular probe is used to scan the surface of a protein, the electrostatic forces between the protein and the probe are calculated in real-time and fed into the force feedback device (Wollacott and Merz, 2007; Bolopion *et al.*, 2010; Hou *et al.*, 2014; Subasi and Basdogan, 2006). Multi-modality enhancements of such tangible models are being created by superimposing graphical information on physical models, by adding voice commands and by providing haptic feedback (Sankaranarayanan *et al.*, 2003). However, it would require another huge space to discuss these advances in molecular visualization.

Concluding Remarks

By describing the state-of-the-art features used for structural visualization of protein and other macromolecules, this article has provided a brief overview of topic. We hope that its content successfully explains most of the concepts related to molecular visualization, especially related to biomolecules. In fact, there is a continuous need for improvement in the field as there is always new details made available by the advances in other technologies related to structure determination. This keeps creating a void which needs to be filled by more advances and improvements in the existing visualization approaches. The development of better and newer software for molecular visualization continues and keeps advancing the field.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Biomolecular Structures: Prediction, Identification and Analyses. Computational Tools for Structural Analysis of Proteins. Drug Repurposing and Multi-Target Therapies. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Protein Structural Bioinformatics: An Overview. Protein Structure Classification. Protein Structure Databases. Protein Three-Dimensional Structure Prediction. Protocol for Protein Structure Modelling. Secondary Structure Prediction. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Study of The Variability of The Native Protein Structure

References

- Andrei, R.M., Callieri, M., Zini, M.F., Loni, T., Maraziti, G., *et al.*, 2012. Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics* 13, S16.
- Baker, N.A., Sept, D., Joseph, S., Holst, M.J., McCammon, J.A., 2001. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences of the United States of America* 98, 10037–10041.
- Bansal, M., Kumart, S., Velavan, R., 2000. HELANAL: A program to characterize helix geometry in proteins. *Journal of Biomolecular Structure and Dynamics* 17, 811–819.
- Beem, K.M., Richardson, D.C., Rajagopalan, K.V., 1977. Metal sites of copper-zinc superoxide dismutase. *Biochemistry* 16, 1930–1936.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28, 235–242.
- Bolopion, A., Cagneau, B., Redon, S., Régner, S., 2010. Comparing position and force control for interactive molecular simulators with haptic feedback. *Journal of Molecular Graphics and Modelling* 29, 280–289.
- Bostick, D., Vaisman, I.I., 2003. A new topological method to measure protein structure similarity. *Biochemical and Biophysical Research Communications* 304, 320–325.
- Bostick, D.L., Shen, M., Vaisman, I.I., 2004. A simple topological representation of protein structure: Implications for new, fast, and robust structural classification. *Proteins: Structure Function and Genetics* 56, 487–501.
- Brunger, A.T., Wells, J.A., 2009. Warren L. DeLano 21 June 1972–3 November 2009. *Nature Structural & Molecular Biology* 16, 1202–1203.
- Callahan, T.J., Swanson, E., Lybrand, T.P., 1996. MD Display: An interactive graphics program for visualization of molecular dynamics trajectories. *Journal of Molecular Graphics* 14 (39–41), 32.
- Carson, M., 1987. Ribbon models of macromolecules. *Journal of Molecular Graphics* 5, 103–106.
- Cass, M.E., Rzepa, H.S., Rzepa, D.R., Williams, C.K., 2005. The use of the free, open-source program Jmol to generate an interactive web site to teach molecular symmetry. *Journal of Chemical Education* 82, 1736.
- Connolly, M., 1983. Analytical molecular surface calculation. *Journal of Applied Crystallography* 16, 548–558.
- Deanda, F., Pearlman, R.S., 2002. A novel approach for identifying the surface atoms of macromolecules. *Journal of Molecular Graphics and Modelling* 20, 415–425.
- Duncan, B.S., Macke, T.J., Olson, A.J., 1995. Biomolecular visualization using AVS. *Journal of Molecular Graphics* 13, 271–282.
- Ferrin, T.E., Huang, C.C., Jarvis, L.E., Langridge, R., 1988. The MIDAS display system. *Journal of Molecular Graphics* 6, 13–27.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., Eddy, S.R., Mistry, J., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research* 44, D279–D285.
- Flores, T.P., Moss, D.S., Thornton, J.M., 1994. An algorithm for automatically generating protein topology cartoons. *Protein Engineering* 7, 31–37.
- Flower, D.R., 1994. β -Sheet topology. A new system of nomenclature. *FEBS Letters* 344, 247–250.
- Gilbert, D., Westhead, D., Nagano, N., Thornton, J., 1999. Motif-based searching in TOPS protein topology databases. *Bioinformatics* 15, 317–326.
- Gilbert, D., Westhead, D., Viskna, J., Thornton, J., 2001. A computer system to perform structure comparison using TOPS representations of protein structure. *Computers & Chemistry* 26, 23–30.
- Gille, C., Frommel, C., 2001. STRAP: Editor for STRUCTURAL Alignments of Proteins. *Bioinformatics* 17, 377–378.
- Goddard, T.D., Ferrin, T.E., 2007. Visualization software for molecular assemblies. *Current Opinion in Structural Biology* 17, 587–595.
- Goddard, T.D., Huang, C.C., Ferrin, T.E., 2005. Software extensions to UCSF chimera for interactive visualization of large molecular assemblies. *Structure* 13, 473–482.
- Goddard, T.D., Huang, C.C., Ferrin, T.E., 2007. Visualizing density maps with UCSF Chimera. *Journal of Structural Biology* 157, 281–287.
- Greer, J., Bush, B.L., 1978. Macromolecular shape and surface maps by solvent exclusion. *Proceedings of the National Academy of Sciences of the United States of America* 75, 303–307.
- Guo, D., Nie, J., Liang, M., *et al.*, 2015. View-dependent level-of-detail abstraction for interactive atomistic visualization of biological structures. *Computers & Graphics* 52, 62–71.
- Hanson, R., 2010. Jmol – A paradigm shift in crystallographic visualization. *Journal of Applied Crystallography* 43, 1250–1260.
- Hanson, R.M., Prilusky, J., Renjian, Z., Nakane, T., Sussman, J.L., 2013. JSmol and the next-generation web-based representation of 3D molecular structure as applied to proteopedia. *Israel Journal of Chemistry* 53, 207–216.
- Hayes, J.M., 2014. An integrated visualization and basic molecular modeling laboratory for first-year undergraduate medicinal chemistry. *Journal of Chemical Education* 91, 919–923.
- Herraez, A., 2006. Biomolecules in the computer: Jmol to the rescue. *Biochemistry and Molecular Biology Education* 34, 255–261.
- Hodis, E., Prilusky, J., Martz, E., *et al.*, 2008. Proteopedia – A scientific 'wiki' bridging the rift between three-dimensional structure and function of biomacromolecules. *Genome Biology* 9, R121.
- Hogue, C.W.V., 1997. Cn3D: A new generation of three-dimensional molecular structure viewer. *Trends in Biochemical Sciences* 22, 314–316.
- Hou, X., Sourina, O., Klimenko, S., 2014. Visual haptic-based collaborative molecular docking. In: *Proceedings of the 15th International Conference on Biomedical Engineering*.
- Hua, Q.X., Gozani, S.N., Chance, R.E., *et al.*, 1995. Structure of a protein in a kinetic trap. *Nature Structural Biology* 2, 129–138.
- Humphrey, W., Dalke, A., Schulten, K., 1996. VMD: Visual molecular dynamics. *Journal of Molecular Graphics* 14 (33–38), 27–38.
- Hutchinson, E.G., Thornton, J.M., 1990. HERA – A program to draw schematic diagrams of protein secondary structures. *Proteins* 8, 203–212.
- Hutchinson, E.G., Thornton, J.M., 1996. PROMOTIF – A program to identify and analyze structural motifs in proteins. *Protein Science* 5, 212–220.
- Juba, D., Varshney, A., 2008. Parallel, stochastic measurement of molecular surface area. *Journal of Molecular Graphics and Modelling* 27, 82–87.
- Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K., 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research* 45, D353–D361.
- Kendrew, J.C., Bodo, G., Dintzis, H.M., *et al.*, 1958. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature* 181, 662–666.
- Koch, I., Kaden, F., Selbig, J., 1992. Analysis of protein sheet topologies by graph theoretical methods. *Proteins: Structure Function and Bioinformatics* 12, 314–323.
- Koradi, R., Billeter, M., Wuthrich, K., 1996. MOLMOL: A program for display and analysis of macromolecular structures. *Journal of Molecular Graphics* 14 (51–55), 29–32.
- Krieger, E., Vriend, G., 2014. YASARA view – Molecular graphics for all devices – From smartphones to workstations. *Bioinformatics* 30, 2981–2982.
- Kyte, J., Doolittle, R.F., 1982. A simple method for displaying the hydrophobic character of a protein. *Journal of Molecular Biology* 157, 105–132.
- Laskowski, R.A., 1995. SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. *Journal of Molecular Graphics* 13 (323–330), 307–328.
- Laskowski, R.A., 2009. PDBsum new things. *Nucleic Acids Research* 37, D355–D359.
- Laskowski, R.A., Jabłońska, J., Pravda, L., Vařeková, R.S., Thornton, J.M., 2018. PDBsum: Structural summaries of PDB entries. *Protein Science* 27, 129–134.
- Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A.C., *et al.*, 2014. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Research* 42, D1091–D1097.
- Lee, B., Richards, F.M., 1971. The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology* 55, 379. (IN374).
- Lesk, A.M., Hardman, K.D., 1982. Computer-generated schematic diagrams of protein structures. *Science* 216, 539–540.
- Levinthal, C., 1966. Molecular model-building by computer. *Scientific American* 214, 42–52.
- Levitt, M., Chothia, C., 1976. Structural patterns in globular proteins. *Nature* 261, 552.
- Martin, A.C.R., 2000. The ups and downs of protein topology: Rapid comparison of protein structure. *Protein Engineering Design and Selection* 13, 829–837.
- Mashaghi, A., van Wijk Roeland, J., Tans Sander, J., 2014. Circuit topology of proteins and nucleic acids. *Structure* 22, 1227–1237.

- May, P., Kreuchwig, A., Steinke, T., Koch, I., 2010. PTGL: A database for secondary structure-based protein topologies. *Nucleic Acids Research* 38, D326–D330.
- McDonald, I.K., Thornton, J.M., 1994. Satisfying hydrogen bonding potential in proteins. *Journal of Molecular Biology* 238, 777–793.
- Merritt, E.A., Bacon, D.J., 1997. Raster3D: Photorealistic Molecular Graphics. *Methods in Enzymology*. Academic Press, pp. 505–524.
- Michalopoulos, I., Torrance, G.M., Gilbert, D.R., Westhead, D.R., 2004. TOPS: An enhanced database of protein structural topology. *Nucleic Acids Research* 32, D251–D254.
- Milburn, D., Laskowski, R.A., Thornton, J.M., 1998. Sequences annotated by structure: A tool to facilitate the use of structural information in sequence analysis. *Protein Engineering* 11, 855–859.
- Nagano, K., 1977. Triplet information in helix prediction applied to the analysis of super-secondary structures. *Journal of Molecular Biology* 109, 251–274.
- Nagata, H., Mizushima, H., Tanaka, H., 2002. Concept and prototype of protein–ligand docking simulator with force feedback technology. *Bioinformatics* 18, 140–146.
- Nelson, M., Humphrey, W., Kufner, R., *et al.*, 1995. MDScope – A visual computing environment for structural biology. In: Atluri, S.N., Yagawa, G., Cruse, T. (Eds.), *Computational Mechanics '95: Theory and Applications*. Berlin, Heidelberg: Springer, pp. 476–481.
- O'Donoghue, S.I., Sabir, K.S., Kalemans, M., Stolte, C., 2015. Aquaria: Simplifying discovery and insight from protein structures. *Nature Methods* 12, 98–99.
- Palmer, T.C., Hausheer, F.H., Saxe, J.D., 1989. Applications of ray tracing in molecular graphics. *Journal of Molecular Graphics* 7, 160–164.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF Chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry* 25, 1605–1612.
- Porollo, A., Meller, J., 2007. Versatile annotation and publication quality visualization of protein complexes using POLYVIEW-3D. *BMC Bioinformatics* 8, 316.
- Porollo, A., Meller, J., 2010. POLYVIEW-MM: Web-based platform for animation and analysis of molecular simulations. *Nucleic Acids Research* 38, W662–W666.
- Porollo, A.A., Adamczak, R., Meller, J., 2004. POLYVIEW: A flexible visualization tool for structural and functional annotations of proteins. *Bioinformatics* 20, 2460–2462.
- Prilusky, J., Hodis, E., Canner, D., *et al.*, 2011. Proteopedia: A status report on the collaborative, 3D web-encyclopedia of proteins and other biomolecules. *Journal of Structural Biology* 175, 244–252.
- Rego, N., Koes, D., 2015. 3Dmol.js: Molecular visualization with WebGL. *Bioinformatics* 31, 1322–1324.
- Ricci, A., Anthopoulos, A., Massarotti, A., Grimstead, I., Brancale, A., 2012. Haptic-driven applications to molecular modeling: State-of-the-art and perspectives. *Future Medicinal Chemistry* 4, 1219–1228.
- Richards, F.M., 1977. Areas, volumes, packing and protein structure. *Annual Review of Biophysics and Bioengineering* 6, 151–176.
- Richardson, D.C., Richardson, J.S., 1992. The kinemage: A tool for scientific communication. *Protein Science* 1, 3–9.
- Richardson, J.S., 1977. β -Sheet topology and the relatedness of proteins. *Nature* 268, 495.
- Rocchia, W., Alexov, E., Honig, B., 2001. Extending the applicability of the nonlinear Poisson–Boltzmann equation: Multiple dielectric constants and multivalent ions. *The Journal of Physical Chemistry B* 105, 6507–6514.
- Rose, A.S., Hildebrand, P.W., 2015. NGL Viewer: A web application for molecular visualization. *Nucleic Acids Research* 43, W576–W579.
- Rose, P.W., Prlić, A., Altunkaya, A., *et al.*, 2017. The RCSB protein data bank: Integrative view of protein, gene and 3D structural information. *Nucleic Acids Research* 45, D271–D281.
- Rose, A.S., Bradley, A.R., Valasatava, Y., *et al.*, 2016. Web-based molecular graphics for large complexes. In: *Proceedings of the 21st International Conference on Web3D Technology*, pp. 185–186. Anaheim, California: ACM.
- Ryu, J., Park, R., Kim, D.-S., 2007. Molecular surfaces on proteins via beta shapes. *Computer-Aided Design* 39, 1042–1057.
- Sánchez-Ferrer, A., Núñez-Delgado, E., Bru, R., 1995. Software for viewing biomolecules in three dimensions on the internet. *Trends in Biochemical Sciences* 20, 286–288.
- Sankaranarayanan, G., Weghorst, S., Sanner, M., Gillet, A., Olson, A., 2003. Role of haptics in teaching structural molecular biology. In: *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*, pp. 363–366, March 22–23, 2003.
- Sanner, M.F., Olson, A.J., Spehner, J.C., 1996. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers* 38, 305–320.
- Sayle, R.A., Milner-White, E.J., 1995. RASMOL: Biomolecular graphics for all. *Trends in Biochemical Sciences* 20, 374.
- Schrodinger, L.L.C., 2015. The PyMOL Molecular Graphics System, Version 1.8.
- Sehnal, D., Svobodová Váňková, R., Berka, K., *et al.*, 2013. MOLE 2.0: Advanced approach for analysis of biomacromolecular channels. *Journal of Cheminformatics* 5, 39.
- Shi, M., Gao, J., Zhang, M.Q., 2017. Web3DMol: Interactive protein structure visualization based on WebGL. *Nucleic Acids Research*.
- Sillitoe, I., Lewis, T.E., Cuff, A., *et al.*, 2015. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research* 43, D376–D381.
- Stivala, A., Wybrow, M., Wirth, A., Whisstock, J.C., Stuckey, P.J., 2011. Automatic generation of protein structure cartoons with Pro-origami. *Bioinformatics* 27, 3315–3316.
- Stocks, M.B., Hayward, S., Laycock, S.D., 2009. Interacting with the biomolecular solvent accessible surface via a haptic feedback device. *BMC Structural Biology* 9, 69.
- Stone, J.E., Sener, M., Vandivort, K.L., *et al.*, 2016. Atomic detail visualization of photosynthetic membranes with GPU-accelerated ray tracing. *Parallel Computing* 55, 17–27.
- Subasi, E., Basdogan, C., 2006. A New Approach to Molecular Docking in Virtual Environments with Haptic Feedback, pp. 141–145.
- Terrell, C.R., Listenberg, L.L., 2017. Using molecular visualization to explore protein structure and function and enhance student facility with computational tools. *Biochemistry and Molecular Biology Education* 45, 318–328.
- Tsukamoto, Y., Takiguchi, K., Satou, K., *et al.*, 1997. Application of a deductive database system to search for topological and similar three-dimensional structures in protein. *Computer Applications in the Biosciences*. 183–190.
- Veeramalai, M., Gilbert, D., 2008. A novel method for comparing topological models of protein structures enhanced with ligand information. *Bioinformatics* 24, 2698–2705.
- Viksna, J., Gilbert, D., 2001. Pattern matching and pattern discovery algorithms for protein topologies. In: Gascuel, O., Moret, B.M.E. (Eds.), *Proceedings of the First International Workshop on Algorithms in Bioinformatics, WABI 2001*, pp. 98–111. Århus Denmark, Berlin, Heidelberg: Springer, August 28–31, 2001.
- Wallace, A.C., Laskowski, R.A., Thornton, J.M., 1995. LIGPLOT: A program to generate schematic diagrams of protein–ligand interactions. *Protein Engineering* 8, 127–134.
- Walters, P., Stahl, M., 1996. Babel.
- Walther, D., 1997. WebMol – A Java-based PDB viewer. *Trends in Biochemical Sciences* 22, 274–275.
- Wang, Y., Geer, L.Y., Chappey, C., Kans, J.A., Bryant, S.H., 2000. Cn3D: Sequence and structure views for Entrez. *Trends in Biochemical Sciences* 25, 300–302.
- Westhead, D.R., Hatton, D.C., Thornton, J.M., 1998. An atlas of protein topology cartoons available on the world-wide web. *Trends in Biochemical Sciences* 23, 35–36.
- Westhead, D.R., Slidel, T.W.F., Flores, T.P.J., Thornton, J.M., 1999. Protein structural topology: Automated analysis and diagrammatic representation. *Protein Science* 8, 897–904.
- Wollacott, A.M., Merz, K.M., 2007. Haptic applications for molecular structure manipulation. *Journal of Molecular Graphics and Modelling* 25, 801–805.
- Yang, Z., Lasker, K., Schneidman-Duhovny, D., *et al.*, 2012. UCSF Chimera, MODELLER, and IMP: An integrated modeling system. *Journal of Structural Biology* 179, 269–278.
- Yuan, S., Chan, H.C.S., Hu, Z., 2017. Implementing WebGL and HTML5 in macromolecular visualization and modern computer-aided drug design. *Trends in Biotechnology* 35, 559–571.

Relevant Websites

<http://aquaria.ws>

Aquaria.

<http://www.bioblender.eu>

Bioblender.

www.jmol.org

Jmol.

<https://sourceforge.net/projects/jmol/>

Jmol download.

<https://sourceforge.net/projects/molmol/>

Molmol.

<https://www.umass.edu/microbio/chime/index.html>

MolviZ.Org.

<https://sourceforge.net/projects/~openrasmol/>

OpenRasMol.

<http://www.ospray.org/>

OSPRay.

<https://www.rcsb.org/pdb/home/home.do>

PDB.

<http://www.wwpdb.org/documentation/file-format>

PDB File Format Documentation.

<http://www.ebi.ac.uk/pdbsum>

PDBsum - EMBL-EBI.

<http://munk.csse.unimelb.edu.au/pro-origami/>

Protein Structure Cartoons.

<http://www.proteopedia.org>

Proteopedia, life in 3D.

<http://ptgl.uni-frankfurt.de/>

PTGL - The Protein Topology Graph Library.

<https://pymol.org/>

PyMOL.

<https://sourceforge.net/projects/pymol/>

PyMOL Molecular Graphics System download.

<http://radsite.lbl.gov/radiance/framew.html>

RADIANCE.

<http://rasmol.org/>

Rasmol.

<http://www.rcsb.org>

RCSB PDB: Homepage.

<http://www.povray.org/>

The Persistence of Vision Raytracer.

<http://www.bioinf.org.uk/topscan/>

TOPSCAN - Rapid protein structure comparison.

<http://tops.ebi.ac.uk/tops>

TOPS - Protein topology atlas - EMBL-EBI.

<http://www.cgl.ucsf.edu/chimera/>

UCSF Chimera Home Page - RBVI.

<http://www.ks.uiuc.edu/Research/vmd/>

VMD - Visual Molecular Dynamics.

<https://github.com/cvdlab-projects/webmol>

webmol: Web Protein Viewer.

<https://github.com/cvdlab-projects/webpdb>

webpdb: Web Protein Data Bank.

<https://www.wikipedia.org/>

Wikipedia.

Computational Tools for Structural Analysis of Proteins

Luciano A Abriata, Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland and Swiss Institute of Bioinformatics, Lausanne, Switzerland

© 2019 Elsevier Inc. All rights reserved.

Glossary

Coevolution coupling Extent of correlated amino acid substitutions between pairs of positions in a protein sequence.

Force field (in molecular dynamics simulations) Set of equations and parameters that describe the potential energy of a system, required to compute and propagate forces during molecular dynamics simulations.

HTML The code standard for writing web pages.

NOE restraint Restraint about upper distances between pairs of – typically – hydrogen atoms, as retrieved from the nuclear Overhauser effect in special nuclear magnetic resonance (NMR) spectra.

Site (in the context of proteins) Position in a protein's sequence, residue.

SMILES (small molecules) A computer-readable text code to encode the chemical constitution of molecules.

Introduction

Work on any protein begins by defining the exact amino acid sequence of the relevant protein form and by retrieving annotations from resources like UniProt in [Table 1](#) ([Pundir et al., 2016](#)). One can then make extensive analysis of the protein's sequence in structural and functional terms; but at the deepest residue and atomic levels, protein-specific details are best available from analyses of experimental structures or models of reasonable confidence. Such structural analyses help to readily explain biochemical observations and suggest experiments, toward the final aim of dissecting structure–function relationships for the system and a full physicochemical description of the protein's role in organismal physiology. This article deals with a variety of tools to carry out such structural analyses of proteins.

Table 1 Sources of protein sequences with structural annotations, databases of three dimensional structures, and methods to model biomacromolecules

Retrieval of protein sequences	
UniProt	http://www.uniprot.org/
P-FAM	http://pfam.xfam.org/
Basic local alignment search tool (BLAST) (and BLAST Protein Data Bank (PDB))	http://blast.ncbi.nlm.nih.gov/Blast.cgi
PDBFINDER2	ftp://ftp.cmbi.ru.nl/pub/molbio/data/pdbfinder2/
Experimental structures	
WorldWide PDB	http://wwpdb.org/
RCSB PDB	http://www.rcsb.org/
PDB Europe	http://www.ebi.ac.uk/pdbe/node/1
PDB Japan	http://pdj.org/
PDB-derived databases	(see Abriata <i>Briefings in Bioinformatics</i> 2016)
Primary data for experimental structures	
Nuclear magnetic resonance	http://bmrb.wisc.edu/
X-ray diffraction	http://eds.bmc.uu.se/eds/
Electron microscopy	http://www.rcsb.org/
Resources for structural modeling excluding homology modeling	
Rosetta Suite	https://www.rosettacommons.org/software
QUARK	http://zhanglab.ccmb.med.umich.edu/QUARK/
EVFold	http://evfold.org/evfold-web/evfold.do
RBO Aleph	http://compbio.robotics.tu-berlin.de/rbo_aleph/
RaptorX contact predictor	http://raptorex.uchicago.edu/ContactMap/
Tools for modeling complexes and assemblies	
PoWER	http://lbm.epfl.ch/resources
HADDOCK	http://haddock.science.uu.nl/
Packmol	http://www.ime.unicamp.br/~martinez/packmol/home.shtml
LipidBuilder	http://lipidbuilder.epfl.ch/home
CHARMM-GUI	http://charmm-gui.org/

Retrieving Experimental Protein Structures and Modeling Unknown Structures for Analysis

All experimentally determined structures are deposited in a validated form at the servers of the worldwide Protein Data Bank (PDB) partnership (Table 1; Berman *et al.*, 2003), containing at the moment over 130,000 structures (Berman *et al.*, 2013). One can reach a given structure either from a PDB ID indicated in a publication, by performing a BLAST or HHpred search against the PDB, or through detailed searches in the PDB data centers themselves, which typically allow extensive filtering based on sequence similarity, the presence of specific molecules, the experimental technique used to solve the structure, structure quality parameters, etc. Note that PDB searches can yield the structure of the protein of interest itself or can yield template structures for homology modeling of the subject protein. Also, PDB searches often return structures of mutants, complexes with antibodies or small molecules, and other forms that do not correspond exactly to the isolated, native protein.

Importantly, many PDB-derived webservers and databases specialize on specific kinds of molecules and even on specific protein families, or in PDB entries containing specific interactions, or in sequence-structure or structure-dynamics relationships, etc. They are extremely useful for the structural biologist and bioinformatician, as they facilitate many kinds of analyses, as recently reviewed in Abriata (2016a).

When no experimental structures are available for the protein or assembly of interest, there are still several avenues that can be exploited to obtain at least a structural model of it, or parts of it, from various sources of information. Such modeling strategies might be exclusively computer-based or might integrate sparse experimental data with computational modeling tools (Tamo *et al.*, 2015). An excellent, up-to-date briefing on protein modeling strategies is available (Kc, 2016), while a very important resource is the collection of articles produced after each biannual critical assessment of structure prediction (CASP) competition (Molt *et al.*, 2016). Here only a brief overview of modeling methods based on protein homology, residue coevolution and *ab initio* folding is given.

Homology modeling consists in building a model of the coordinates of a protein's atoms using the experimental structure of another protein as a template, both assumed structurally similar given their high sequence similarity (Khan *et al.*, 2016). Since high sequence similarity with a good coverage of the subject sequence is needed, modeling efforts focus mainly on well-defined domains. A new emerging strategy, far less established than homology modeling but applicable when no structures of homologs are available, exploits the strong coevolution couplings undergone by pairs of residues that make contacts in folded proteins. After several works reporting the proof of concept and extended testing (Hopf *et al.*, 2012, 2014; Kamisetty *et al.*, 2013; Marks *et al.*, 2011; Morcos *et al.*, 2011; Ovchinnikov *et al.*, 2014), actual use to model structures began in the last ~2 years (Abriata, 2016b; Kassem *et al.*, 2016; Ovchinnikov *et al.*, 2015, 2017; Tian *et al.*, 2015). Some online tools like EVFold and RaptorX take a sequence as input, build an alignment from it, compute coevolution couplings and finally build 3D models. Others like Gremlin input a sequence, build an alignment and provide a list of couplings and structural restraints that can be used in external modeling programs.

Last, a series of methods attempt to fold proteins "*ab initio*" or "*de novo*" (Błaszczuk *et al.*, 2013; Bradley *et al.*, 2005; Mabrouk *et al.*, 2015; Xu and Zhang, 2012). The success rate of these methods is relatively low, so extensive external validation is important. (Plus, note that even in successful cases, the folding protocols and pathways do not necessarily mimic the true events that lead to the folded protein.) Among *ab initio* methods, there are examples based on classical molecular dynamics simulations started from extended conformations; these approaches are computationally very expensive, even if enhanced sampling methods are used, and are currently limited to peptides, although some small proteins have been folded using specialized supercomputers (Lindorff-Larsen *et al.*, 2011; Piana *et al.*, 2013).

Analyzing Protein Structures in Their Biological Contexts: Protein–Protein, Protein–Nucleic Acid and Protein–Membrane Assemblies

Very often, protein structures are determined in nonnative conditions, for example, a membrane protein is solved in micelle lipids instead of a lipid bilayer, or a protein is solved isolated but physiologically exists only within the framework of a larger assembly. One can use a number of tools to build up models of their complexes, and even high-order assemblies, using contact information from experiments or computation, and experimental data about shape and volume of the complex, even accounting for subunit dynamics (Tamo *et al.*, 2015). It is also possible to explore protein oligomerization through – for example, coarse-grain molecular dynamics simulations or docking, typically aided by or tested through mutagenesis experiments. Atomistic molecular dynamics simulations are also potentially useful but are still far from becoming routine due to the extensive sampling required, the typically large systems involved, and many shortcomings of current force fields (Abriata and Dal Peraro, 2015; Bandaranayake *et al.*, 2012; Petrov and Zagrovic, 2014); however, they have potential for scoring predicted complexes (Sarti *et al.*, 2016).

When insertion of proteins into membranes is important, there are tools that help assemble proteins into lipid bilayers, disks, micelles and other lipid-based structures (Bovigny *et al.*, 2015; Jo *et al.*, 2009; Lomize *et al.*, 2006; Stansfeld *et al.*, 2015). Some of these tools further provide starting points for simulations with popular programs. For combining even more elements, in principle with no limitations, programs like Packmol and Cellpack (Johnson *et al.*, 2015; Martinez *et al.*, 2009) become very handy. These programs facilitate the setup of atomistic or coarse-grained systems for visualization, comparison of geometries, serving as starting points for molecular dynamics simulations of complex multimolecular systems as in Spiga *et al.* (2014).

A Note on Computational Models of Proteins and Assemblies

As an important note, each computational model is useful for tasks and analysis of varying complexity depending on the approach and the quality of the underlying experimental and computational information utilized to derive it. For example, a model of a

transmembrane protein based on packing of predicted transmembrane helices will be enough to approximate how the protein embeds in biological membranes and to estimate the relative location of specific residues, especially along the membrane normal, possibly even leading to propose mutations that can be tested experimentally. However, it will very likely be unstable in atomistic molecular dynamics simulations and certainly will not serve for docking or virtual screening campaigns.

Alignments in a Structural Context

It is standard practice in biology to think about a subject protein in the context of its family, as this helps to identify residues that are ultimately important for the protein to achieve its function. Moreover, protein 3D structures are conserved much more than their sequences, such that highly similar structures often imply a similar function, which is very important for function prediction of new protein sequences.

These observations have practical parallels in structural bioinformatics, with the advantage that (1) typically much larger numbers of related sequences are analyzed, and – as a consequence of this – (2) that specific reasons for why different amino acids are more or less tolerated at each position can be explored deeply. The second advantage stems from an often overlooked aspect of amino acid variation in proteins, namely that the probability of observing a given substitution in a given condition arises as the combination of a variety of factors that can be affected on top of effects on function itself. The list includes trivial deleterious effects of mutations at active site residues and well-known effects on the stability of the protein, but also effects on protein internal dynamics (González *et al.*, 2016), on the response to unexpected interactions forced at high concentrations (Abriata *et al.*, 2016a), and even effects not directly related to the protein itself, for example, at the transcription level (Weatheritt and Babu, 2013). Moreover, substitutions at multiple sites are often coupled through structural constraints or might display nonadditive effects on some protein traits, facts that often contain structural and functional information. Keeping all these points in mind while analyzing alignments is hard, but as shown in this section many computational tools help to understand some of these effects and thus gain more insights into the subject protein.

On one hand, the new methods for calculating *de novo* protein structures and protein complexes from coevolution patterns as described above, have also proved powerful to find out relevant alternative protein conformations (Morcos *et al.*, 2013). Therefore, given a protein structure or model, mapping of contacts predicted through coevolution methods can reveal alternative conformations of functional significance (Table 2). On the other hand, alignments that are consistent with structure according to residue coevolution analysis have also proved informative about the physical and biological chemistry that underlie sequence variation in the family, in such a way that helps to derive structural and functional information (Abriata *et al.*, 2016b). The PsychoProt webserver, originally developed to dissect amino acid variability from deep-sequence-based experiments on saturated mutational libraries (Abriata *et al.*, 2015), helps to easily quantify conservation and variability throughout a protein's sequence and structure and to extract structural and functional information about the protein. For example, PsychoProt can easily unveil polarity/hydrophobicity requirements in soluble and transmembrane proteins, or residues that evolve under functionally relevant constraints for steric hindrance or conformational dynamics. Such kind of analysis is useful for augmenting the amount of information of both experimental structures and models, as well as to justify the position of hydrophobic and polar amino acids in models and to hypothesize about a protein's dynamic features and functional elements.

Tools for Analysis of Protein Structures and Models

This section focuses on two main kinds of analysis one can perform on structures and models. On one hand, treated first, one can (and should, for many applications) check the quality of the structural information; on the other hand, there are those analyses that deal with extracting information from the models.

Checking the Qualities of Experimental and Modeled Structures

Quality checks are important to estimate how well the protein structure or model satisfies the known geometries and the underlying data. These checks are critical when atomic-level details affect the interpretation or further studies that one wants to achieve from the structure or model. This is the case when, for example, setting up systems for molecular docking, virtual screening and molecular dynamics simulations; but on the other hand is less of a problem, for example, for mapping nuclear magnetic resonance (NMR) data on a model or structure, or if a model/structure is used for solving the X-ray structure of a related protein through molecular replacement.

A large array of tools (RPF, PROCHECK, MolProbity, Verify3D, Prosa, WHAT_CHECK, ERRAT, PROVE (Chen *et al.*, 2010; Colovos and Yeates, 1993; Eisenberg *et al.*, 1997; Huang *et al.*, 2012; Wiederstein and Sippl, 2007; Willard *et al.*, 2003) exist that check for geometries and chemistry, such as reasonable backbone and side chain dihedrals, correct stereochemistry, correct packing without clashes, fulfilment of hydrogen-bonding networks, swapped side chains, etc. Other tools are more specialized for checking the quality of the structures in relation to the underlying X-ray or NMR data. Many of these geometry-, chemistry- and data-related tests are actually compulsory upon submission of new structures to the PDB, which is required to ensure the good quality of PDB

Table 2 Online resources for structural analysis of protein sequences, and for structural interpretation of sequence alignments

Disordered regions, residue exposure and (super)secondary structures from sequence	
Disordered regions	http://dis.embl.de/ http://prdos.hgc.jp/cgi-bin/top.cgi http://iupred.enzim.hu/ http://bioinf.cs.ucl.ac.uk/psipred/?disopred=1 http://morf.chibi.ubc.ca/ http://webapp.yama.info.waseda.ac.jp/fang/MoRFs.php http://bioinf.cs.ucl.ac.uk/psipred/?disopred=1 http://anchor.enzim.hu/ http://sable.cchmc.org/ http://www.cbs.dtu.dk/services/NetSurfP/
Disordered regions prone to interaction	http://bioinf.cs.ucl.ac.uk/psipred/ http://www.compbio.dundee.ac.uk/jpred/ https://npsa-prabi.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_phd.html http://www.ibi.vu.nl/programs/yaspinwww/ http://split.pmfst.hr/split/4/ http://www.cbs.dtu.dk/services/TMHMM/ http://www.ch.embnet.org/software/TMPRED_form.html http://bioinf.cs.ucl.ac.uk/software_downloads/memsat/ http://harrier.nagahama-i-bio.ac.jp/sosui/ http://octopus.cbr.su.se/ http://phobius.binf.ku.dk/
Solvent accessibility	http://www.ch.embnet.org/software/COILS_form.html http://paircoil2.csail.mit.edu/ http://multicoil2.csail.mit.edu/cgi-bin/multicoil2.cgi https://npsa-prabi.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_lupas.html
Secondary structure	http://www.ch.embnet.org/software/COILS_form.html http://paircoil2.csail.mit.edu/ http://multicoil2.csail.mit.edu/cgi-bin/multicoil2.cgi https://npsa-prabi.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_lupas.html
Transmembrane helices, Transmembrane topology and signal peptide	Fit to a structure: http://arteni.cs.dartmouth.edu/cccp/index.fit.html Generate: http://www.grigoryanlab.org/cccp/index.gen.html Generate: http://coiledcoils.chm.bris.ac.uk/app/cc_builder/ Draw wheel diagram: http://www.grigoryanlab.org/drawcoil/ http://www.ebi.ac.uk/Tools/pfa/radar/
Coiled-coil prediction	
Coiled-coil analysis	
Repeat detection	
Coevolution from alignments	
Intragenic residue–residue coevolution (for single proteins)	http://evfold.org/evfold-web/ http://gremlin.bakerlab.org/submit.php http://dca.rice.edu/portal/dca/ http://mistic.leloir.org.ar/index.php http://gremlin.bakerlab.org/cplx_submit.php https://evcomplex.hms.harvard.edu/ http://csbg.cnb.csic.es/mtserver/ http://i-coms.leloir.org.ar/index.php
Intergenic residue–residue coevolution (for protein complexes and interactions)	http://lucianoabriata.altervista.org/evocoupldisplay/gremlin.html http://lucianoabriata.altervista.org/evocoupldisplay/evcouplings.html
Contact map against coevolution patterns	
Structural and physicochemical features from alignments	
PsychoProt	http://psychoprot.epfl.ch/
Variability from alignment	http://psychoprot.epfl.ch/aln2data.html
ProtParam, ProtScale	http://web.expasy.org/protparam/ , http://web.expasy.org/protscale/
MultiProtScale	http://lucianoabriata.altervista.org/multirotscale/multirotscale.html
Alignment filter	http://lucianoabriata.altervista.org/multirotscale/alignmentfilterer.html

entries, derived experimentally. It must be said on the other hand, though, that as shown in recent CASP experiments, computational models of better geometry/chemistry scores do not warrant agreement to “true” structures.

Most checking tools are available as servers, and moreover there are servers that integrate multiple tools: PSVS, SAVES, PROSESS, VADAR, CING, ResProx, Vivaldi (Berjanskii *et al.*, 2010, 2012; Bhattacharya *et al.*, 2007; Doreleijers *et al.*, 2012; Hendrickx *et al.*, 2013). Some servers, notably MolProbity, return an overall score that puts together different quality statistics, in this case such that lower numbers overall indicate “better” structures.

For initial checks on specific PDB entries, there are databases of precomputed quality checks (Hooft *et al.*, 1996) and of optimized PDB entries (Joosten *et al.*, 2014). Yet other servers focus on specific problems structures might suffer from, for example, on the refinement of metal sites (Zheng *et al.*, 2014), on trans-cis, amide side chain and peptide plane flips (Touw *et al.*, 2015; Weichenberger and Sippl, 2007), etc.

Visualization of Biomolecular Structures and Models

Moving on to the second kind of structural analyses, more directed to extract information out of the experimental or modeled structure, there are several questions that computational approaches can solve or at least shed light into. These include analysis of structural features such as surface properties, electrostatics, dynamics; identification of active sites, allosteric sites and interaction sites, mapping of conserved and variable regions, testing for the effects of mutations, measuring the degree of exposure of given residues, etc.

Visualization of molecular structures is key to work in structural biology, and a key step of structural analysis. Moreover, careful inspection of structures and models is critical before running automated analysis or molecular dynamics simulations on them, especially to highlight problems that are invisible to the analyses and could actually introduce errors on their results.

Among the most popular visualization tools that are free for academics, there are COOT (especially suited for working with X-ray data), MOLMOL (specialized for NMR structures), VMD (mainly tailored to setup and analyze MD simulations), PyMOL and Chimera (both with very simple ways to model secondary structures and nonnatural groups and to get stunning images). Many of these visualization programs actually do much more than displaying structures, containing tens of commands for molecular building and analysis. With some advanced knowledge of these tools one can also create detailed movies; on the other hand, simple movies about rotations, morphing and vibrations among other few possibilities, can be rendered online with the MovieMaker webserver (Maiti *et al.*, 2005).

Nowadays, integration of biomolecular structures directly inside HTML for online visualization without plug-ins is possible thanks to tools like JSmol (Hanson *et al.*, 2013), PV, GLmol (Virag *et al.*, 2016), 3dmol.js (Rego and Koes, 2015), Molmil (Bekker *et al.*, 2016), Litemol, and the NGLviewer (Rose and Hildebrand, 2015) soon to be updated to handle large macromolecular complexes online (Rose *et al.*, 2016). Such tools make it extremely easy to openly share customized, interactive 3D views of molecules (see Relevant Websites).

Properties of Protein Surfaces and Cores

The physicochemical properties of a protein's core and surface are important to its stability, particularly regarding solubility, and to its interactions with specific substrates and proteins or with other solutes present at high concentrations in crowded media. An easy first qualitative look at a protein's surface properties is to simply inspect its surface colored by residue type in any visualization program, or more quantitatively, by mapping the amino acid hydrophobic moments on the protein surface (Eisenberg *et al.*, 1984). You expect mostly polar and charged amino acids at the surface of a globular soluble protein, and mostly hydrophobic amino acids in membrane-embedded regions. Patches of inconsistent polarity point at sites of potential functional importance (e.g., for protein-protein interactions) or that can be mutated to improve solubility (e.g., in an enzyme) or simply to errors in the case of models. Note that a given protein might contain an isolated amino acid that does not match polarity requirements while the whole family does contain information about the correct polarity preferred at the position; here is where the joint analysis of alignments in the context of protein structures with tools like PsychoProt as mentioned above (Abriata *et al.*, 2016b) becomes useful. Last, there are also tools to simplify visualization and comparison of the shape and physicochemical features of protein surfaces through mapping in 2D (Yang *et al.*, 2012).

Closely related to surface polarity is the global electrostatics of a protein or complex (Dong *et al.*, 2008). The APBS program and online server (Baker *et al.*, 2001) computes electrostatic fields and electrostatic surface maps from the coordinates of a molecular system, which can be visualized easily in programs like PyMOL and VMD. This requires assignment of charges to all atoms, which can be carried out with PDB2PQR (Dolinsky *et al.*, 2007) through a handful of charge models and which might in turn require assessment of the protonation states of several groups as automatically carried out by PROPKA (Olsson *et al.*, 2011). The main PDB2PQR server (Unni *et al.*, 2011) performs all these calculations, in its simplest mode taking a PDB file and returning an APBS file for visualization (as electrostatic surfaces or fields) in external programs like PyMOL and VMD. VMD can also itself calculate the electric moment of a selection of atoms, which can then be used to track its evolution over time.

Regarding protein cores, structure-based analyses of alignments with PsychoProt revealed that not only hydrophobicity is required in a protein's interior but also precise combinations of volume and steric hindrance, adorned with regions of certain flexibility or conformational preferences closer to the protein surface (Abriata *et al.*, 2015, 2016b). Stability of the core of globular proteins upon mutation can be tested with tools like FoldX, I-Mutant, Dmutant, PopMusic and Rosetta, among others. These programs tend to be better predictors of destabilizing than stabilizing mutations (for an assessment of different tools see (Khan and Vihinen, 2010)). Last, another informative way to test packing in a protein structure or model is to compare its contact map to the coevolutionary couplings computed from a related alignment (tools for this are available in Table 2).

As a final point, for precise quantification of solvent accessibility, programs like VMD have built-in functions that can easily be executed on multiple structures or on simulation trajectories, while servers like POPS (Cavallo *et al.*, 2003) provide easier ways to run the calculation but on single structures.

Small-Molecule Pockets and Protein-Small Molecule Interactions

Testing the interactions between small molecules and proteins requires special treatments, and the procedure is usually split into finding pockets where small molecules can fit, and finding how small molecules can bind to it.

Many tools allow users to search for potential small molecule-binding sites in protein structures. Among them, fpocket (Schmidtke *et al.*, 2010) implements a high-performance algorithm which allows detection and tracking of pockets in large ensembles of structures or molecular dynamics trajectories. Another very recent and complete tool, PockDrug (Hussein *et al.*, 2015), is a server specialized in finding druggable pockets, which inputs a PDB file and returns the detected pockets with complete physicochemical descriptions plus scores that estimate how druggable they are.

Once a druggable pocket has been identified in a protein, or if it is already known, one can choose among a large list of academic and paid programs for docking small molecules to the target pockets (full searches are also possible but much more time consuming and leading to less significant results). These programs can be used to test how a given molecule would dock in a pocket, often called “virtual docking,” and can be run iteratively on a library to predict which molecules are expected to bind stronger, often called “virtual screening.” It is important that such campaigns usually seek to enrich libraries in drug leads and potentially active molecules, but are currently not capable of delivering definitive answers on the perfect drug for a given pocket. For a recent review of potentials and limitations for realistic uses in biology, see Forli (2015), Irwin and Shoichet (2016). For the special case of docking peptides as small molecules onto proteins, there are specialized tools from the Rosetta and HADDOCK families (London *et al.*, 2011; Spiliotopoulos *et al.*, 2016).

Note that there are many challenges in the field of docking and virtual screening, related to poor sampling of flexibility especially in the target protein and to overlooked or only poorly described effects of water (Spyrakis and Cavasotto, 2015). Such problems could in principle be solved by using atomistic molecular dynamics simulations; however, they are orders of magnitude more expensive and hence currently outperformed by docking-based methods, although they are increasingly becoming useful to score poses obtained through docking-based methods (for a thorough discussion on the role of molecular dynamics simulations in drug discovery, see De Vivo *et al.* (2016).

Regarding the structural universe of small molecules, among the most important servers for structural research there are some (with Chemicalize.org (Southan and Stracz, 2013) and CORINA standing out) that facilitate conversion between compound names, SMILES codes, 2D schemes and 3D structures of small molecules, some of them linking directly to existing annotations and retrieving or computing on-the-fly physicochemical descriptors for the small molecules. Other resources for small molecules act as mere databases of small molecules, useful along the drug discovery process, such as ZINC and PubChem (Irwin and Shoichet, 2005; Kim *et al.*, 2016).

Dynamics From Structures

Experimentally, protein dynamics are accessible at the residue level from crystallographic B-factors if properly analyzed (see the B-factor DataBase (Touw and Vriend, 2014)) and from NMR ^{15}N relaxation (note that simply variability in NMR structures is not a good indication of dynamics). Computationally, the ultimate tool is provided by atomistic molecular dynamics simulations, but such methods are costly and often unnecessary if only identification of flexible regions is required.

It is possible to predict flexibility as “disorder” from sequences with variable confidence, or to inspect a sequence manually by mapping coil propensity and other amino acid descriptors with programs like ProtScale (Table 2). Meanwhile, analysis of alignments with tools like PsychoProt can point at residues where amino acids of small volumes, low steric hindrance, large flexibility or low conformational preference are better tolerated, again indicating dynamics.

Adding a structural level to improve confidence, there are programs that predict flexible regions from structures. Some of them exploit the fact that regions of low atomic packing or high solvent accessibility tend to be more flexible (Marsh, 2013), whereas others like PDBFlex (Hrabe *et al.*, 2016) are based on the comparison of PDB entries for related proteins or even same proteins crystallized in different conditions. Another tool especially useful to identify domain motions mediated by loops acting as hinges is Normal Mode Analysis, implemented together with other related tools in the program ProDy (Bakan *et al.*, 2011). Last and probably highest in detail but only reporting on loops, a tool developed by the Bruschweiler group from a dataset of sub-microsecond MD simulations validated with NMR data, looks directly at protein loops in a structure returning their predicted flexibility and motion timescales (Gu *et al.*, 2015).

Predicting Active Sites and Functions

Prediction of function for a novel protein, or of even subtle functional changes upon mutation or upon interactions, are particularly challenging even if experimental structures are available. While annotation databases or deep searching on Pubmed, for example, with text-mining tools like ChiliBot (Chen and Sharp, 2004) can lead to functional hints directly from written information, there are also tools that can propose approximate functions from sequences and structures. The critical assessment of functional annotation (CAFA) regularly evaluates the state of function evaluation methods from information at the sequence level (Wass *et al.*, 2014), which includes methods based on sequence comparison, on co-occurrence in operons, regulons and genomes, on analysis of gene fusions, on protein colocation, and on structure. Other efforts and groups focus specifically on predicting function from structures, as briefly covered here.

Methods for predicting protein function from sequence and structural homology have been around for some time, as reviewed by Petrey *et al.* (2015). They perform reasonably well at least at the level of broadly defining function, but cannot give precisions on, for example, the exact substrates of enzymes, which is often very sensitive to point mutations (Abriata *et al.*, 2012). As a proxy for function assignment, many programs actually predict ligand-binding sites through comparison with known data, as recently

evaluated in a subcategory of CASP10 (Gallo Cassarino *et al.*, 2014). Some modeling servers like I-TASSER actually provide such analysis on their top models.

When a large alignment is available, there is evidence that gradients of amino acid substitution rates mapped on an enzyme's structure can approximate the location of its active site (Jack *et al.*, 2016), which one might better define through experimental evaluation of mutants of the most conserved residues. One can also analyze what physicochemical properties shape the preference for each of the 20 amino acids at each position of the alignment, with tools like the PsychoProt server, under the premise that preferences for flexibility, steric hindrance, volumes and backbone conformational descriptors could hint at functionally relevant regions of a protein as exemplified in Abriata *et al.* (2016b).

Data Mining

The far end within the field of computational analysis of protein structures consists in (semi)automated mining of PDB entries, either on the full PDB or on PDB-derived databases. Mining the PDB with the help of clever connections to other sources of information, powerful analytical methods, and often using specialized sub-databases, has brought to light several fine details about protein structure, dynamics and hydration, features of transmembrane proteins, metal centers, cofactor and ligand binding, and noncovalent interactions between biomolecules (a few examples among a few hundred such works in the last decade: (Gallivan and Dougherty, 1999; Jackson *et al.*, 2007; Kumar and Balaji, 2014; Lanzarotti *et al.*, 2011; Ringer *et al.*, 2007; Shi *et al.*, 2002; Valley *et al.*, 2012)). But also, mining studies can disclose effects of different experimental methods, diffraction resolution, processing software, etc. (Abriata, 2012, 2013; Djinojic-Carugo and Carugo, 2015; Harding, 2002).

Mining protocols usually involve writing specific programs and scripts for the analysis of interest, but tools aimed at simplifying mining exist (Sehna *et al.*, 2015; Valasatava *et al.*, 2014).

Comparing Structures in 3D

Comparison of biomolecular structures is at the heart of structural biology, as this science attempts to explain function in relation to structure, for example, when classifying domains, inferring function from structure, modeling proteins from homologs, and at higher level of detail when explaining functional differences as consequences of structural differences.

The problem of comparing two structures has two main steps, namely the detection of similar elements in both structures, which are to be superimposed upon alignment, and the subsequent computation that aligns the coordinates of the proteins based on such similarities. Multiple structure alignments usually start by computing all possible pairwise alignments, and then running a global optimization.

At the computer, most visualization tools allow the user to align and compare structures in 3D with simple methods that work fairly well when there is obvious similarity. However, more complex protocols are available from *ad hoc* programs and servers. Among the most widely used, currently up-to-date tools, there are CE, DALI, FATCAT and FATCAT POSA, MAMMOTH, SALIGN, SSP, TopMatch, TM-align and VAST/VAST+ (Holm and Laakso, 2016; Madej *et al.*, 2014; Ortiz *et al.*, 2002; Shindyalov and Bourne, 2001; Sippl and Wiederstein, 2008; Ye and Godzik, 2004a, 2004b; Zhang and Skolnick, 2005). It is important to highlight that different programs usually incur in inconsistencies among each other, even for relatively similar proteins (Sadowski and Taylor, 2012). Some of these tools allow comparisons between multiple provided structures, others (most notably DALI) allow querying a structure against all PDB entries; some facilitate instant online visualization; and in particular, FATCAT includes internal flexibility in its alignments, with a version (POSA) for multiple protein comparisons and another for scanning structural databases. The RCSB PDB currently facilitates online pairwise comparisons of its entries through some of these programs.

Other More Specific Analyses

A multitude of computational tools are continuously developed tailored to more specific questions, tasks and molecules. Examples are the Protein Knot Server to detect knots in structures (Kolesov *et al.*, 2007), the CheckMyMetal server to validate metal ions in protein structures (Zheng *et al.*, 2014), servers for advanced superposition and 3d-searching of proteins structures (Holm and Rosenström, 2010; Maiti *et al.*, 2004), structural databases containing precomputed descriptions for specific molecules (Abriata, 2016a), servers to predict spectroscopic observables like circular dichroism and NMR spectra from protein structures (Bulheller and Hirst, 2009; Han *et al.*, 2011), servers to validate protein structures and quantify their qualities (Willard *et al.*, 2003), and certainly much more.

A Glance at the Power of Molecular Simulation Methods

This article has focused on modeling and analyzing static structures; however, macromolecules undergo dynamics over orders-of-magnitude timescales ranging from the fastest pico-to-nanosecond vibrations to the slowest loop motions relevant for catalysis and allostery, conformational transitions, breathing motions and folding events in the microsecond to second timescales. Understanding protein folding, function, regulation, interactions, assembly processes, etc. requires consideration of these dynamic features of molecules, for which computational molecular simulations are as important as experiments. Moreover, simulations have the potential to embrace most other kinds of analysis, for example, predicting secondary and even tertiary structures through

folding simulations, predicting small-molecule binding even after free diffusion, exploring reactivity, etc., although far much less efficiently and not necessarily more accurately due to problems in the force fields.

The most important tools to study protein dynamics can be split into those based on quantum methods (which allow probing bond breaking and formation, i.e., reactions, as well as energy-exchange processes related to spectroscopic observables), and those based on classical molecular mechanics either at all-atom level (where bonds cannot be broken or formed, but all interatomic dynamics are accessible allowing the sampling of conformational dynamics) or at coarse-grain level (where atoms are grouped into beads of variable size according to the coarse-graining scheme, loosing atomic detail but allowing for extended diffusional sampling). The three levels of detail can in turn be mixed, being particularly popular in structural biology the combination of atomistic molecular mechanics with quantum-level calculations (QM/MM) to study, for example, enzyme catalysis.

Note that one way or another, all simulations methods require parametrizations (most importantly in the form of force fields for classical molecular mechanics simulations, or orbital descriptions in quantum calculations) and assumptions on how to calculate and propagate forces. Therefore, just like results from most other computational tools, results from simulations need to be tested against experimental data, or eventually taken as mere predictions to guide experiments or explain existing results. Classical dynamics can, for example, be compared against NMR dynamic data, quantum calculations can be tested by predicting and comparing spectroscopic observables.

For classical dynamics, several force fields and functional forms are available to treat proteins, nucleic acids, and some ions and lipids. If small molecules are to be used in molecular dynamics simulations, there are a few online databases of ready-to-use parameter files and also online tools that assist in parametrization (Kirschner *et al.*, 2008; Mallocci *et al.*, 2015; Zoete *et al.*, 2011) and building complex systems (Bovigny *et al.*, 2015; Jo *et al.*, 2009; Martinez *et al.*, 2009).

See also: Biomolecular Structures: Prediction, Identification and Analyses. Computational Protein Engineering Approaches for Effective Design of New Molecules. Drug Repurposing and Multi-Target Therapies. Identification of Homologs. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Protein Design. Protein Structural Bioinformatics: An Overview. Protein Structure Analysis and Validation. Protein Structure Classification. Protein Structure Visualization. Protein Three-Dimensional Structure Prediction. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). The Evolution of Protein Family Databases

References

- Abriata, L.A., 2012. Analysis of copper-ligand bond lengths in X-ray structures of different types of copper sites in proteins. *Acta Crystallographica Section D, Biological Crystallography* 68, 1223–1231.
- Abriata, L.A., 2013. Investigation of non-corrin cobalt(II)-containing sites in protein structures of the Protein Data Bank. *Acta Crystallographica Section D, Biological Crystallography* 69, 176–183.
- Abriata, L.A., 2016a. Structural database resources for biological macromolecules. Briefings in Bioinformatics. pii: bbw049 (Epub ahead of print).
- Abriata, L.A., 2016b. Homology- and coevolution-consistent structural models of bacterial copper-tolerance protein CopM support a “metal sponge” function and suggest regions for metal-dependent protein-protein interactions. *bioRxiv*.
- Abriata, L.A., Bovigny, C., Dal Peraro, M., 2016b. Detection and sequence/structure mapping of biophysical constraints to protein variation in saturated mutational libraries and protein sequence alignments with a dedicated server. *BMC Bioinformatics* 17, 242.
- Abriata, L.A., Dal Peraro, M., 2015. Assessing the potential of atomistic molecular dynamics simulations to probe reversible protein–protein recognition and binding. *Scientific Reports* 5, 10549.
- Abriata, L.A., Palzkill, T., Dal Peraro, M., 2015. How structural and physicochemical determinants shape sequence constraints in a functional enzyme. *PLOS ONE* 10, e0118684.
- Abriata, L.A., Salverda, M.L.M., Tomatis, P.E., 2012. Sequence-function-stability relationships in proteins from datasets of functionally annotated variants: The case of TEM β -lactamases. *FEBS Letters* 586, 3330–3335.
- Abriata, L.A., Spiga, E., Peraro, M.D., 2016a. Molecular effects of concentrated solutes on protein hydration, dynamics, and electrostatics. *Biophysical Journal* 111, 743–755.
- Bakan, A., Meireles, L.M., Bahar, I., 2011. ProDy: Protein dynamics inferred from theory and experiments. *Bioinformatics (Oxford, England)* 27, 1575–1577.
- Baker, N.A., Sept, D., Joseph, S., Holst, M.J., McCammon, J.A., 2001. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences of the United States of America* 98, 10037–10041.
- Bandaranayake, R.M., Ungureanu, D., Shan, Y., *et al.*, 2012. Crystal structures of the JAK2 pseudokinase domain and the pathogenic mutant V617F. *Nature Structural & Molecular Biology* 19, 754–759.
- Bekker, G.-J., Nakamura, H., Kinjo, A.R., 2016. Molmil: A molecular viewer for the PDB and beyond. *Journal of Cheminformatics* 8, 42.
- Berjanskii, M., Liang, Y., Zhou, J., *et al.*, 2010. PROSESS: A protein structure evaluation suite and server. *Nucleic Acids Research* 38, W633–W640.
- Berjanskii, M., Zhou, J., Liang, Y., Lin, G., Wishart, D.S., 2012. Resolution-by-proxy: A simple measure for assessing and comparing the overall quality of NMR protein structures. *Journal of Biomolecular NMR* 53, 167–180.
- Berman, H., Henrick, K., Nakamura, H., 2003. Announcing the worldwide Protein Data Bank. *Nature Structural & Molecular Biology* 10, 980.
- Berman, H.M., Coimbatore Narayanan, B., Di Costanzo, L., *et al.*, 2013. Trendspotting in the Protein Data Bank. *FEBS Letters* 587, 1036–1045.
- Bhattacharya, A., Tejero, R., Montellione, G.T., 2007. Evaluating protein structures determined by structural genomics consortia. *Proteins* 66, 778–795.
- Blaszczyk, M., Jamroz, M., Kmiecik, S., Kolinski, A., 2013. CABS-fold: Server for the de novo and consensus-based prediction of protein structure. *Nucleic Acids Research* 41, W406–W411.
- Bovigny, C., Tamò, G., Lemmin, T., Mäino, N., Dal Peraro, M., 2015. LipidBuilder: A framework to build realistic models for biological membranes. *Journal of Chemical Information and Modeling* 55, 2491–2499.
- Bradley, P., Malmström, L., Qian, B., *et al.*, 2005. Free modeling with Rosetta in CASP6. *Proteins* 61 (Suppl 7), 128–134.

- Bulheller, B.M., Hirst, J.D., 2009. DichroCalc – Circular and linear dichroism online. *Bioinformatics* (Oxford, England) 25, 539–540.
- Cavallo, L., Kleinjung, J., Fraternali, F., 2003. POPS: A fast algorithm for solvent accessible surface areas at atomic and residue level. *Nucleic Acids Research* 31, 3364–3366.
- Chen, H., Sharp, B.M., 2004. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics* 5, 147.
- Chen, V.B., Arendall, W.B., Headd, J.J., *et al.*, 2010. MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallographica Section D, Biological Crystallography* 66, 12–21.
- Colovos, C., Yeates, T.O., 1993. Verification of protein structures: Patterns of nonbonded atomic interactions. *Protein Science* 2, 1511–1519.
- De Vivo, M., Masetti, M., Bottegoni, G., Cavalli, A., 2016. Role of molecular dynamics and related methods in drug discovery. *Journal of Medicinal Chemistry* 59, 4035–4061.
- Djinovic-Carugo, K., Carugo, O., 2015. Structural biology of the lanthanides-mining rare earths in the Protein Data Bank. *Journal of Inorganic Biochemistry* 143, 69–76.
- Dolinsky, T.J., Czodrowski, P., Li, H., *et al.*, 2007. PDB2PQR: Expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Research* 35, W522–W525.
- Dong, F., Olsen, B., Baker, N.A., 2008. Computational methods for biomolecular electrostatics. *Methods in Cell Biology* 84, 843–870.
- Doreleijers, J.F., Sousa da Silva, A.W., Krieger, E., *et al.*, 2012. CING: An integrated residue-based structure validation program suite. *Journal of Biomolecular NMR* 54, 267–283.
- Eisenberg, D., Lüthy, R., Bowie, J.U., 1997. VERIFY3D: Assessment of protein models with three-dimensional profiles. *Methods in Enzymology* 277, 396–404.
- Eisenberg, D., Schwarz, E., Komaromy, M., Wall, R., 1984. Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *Journal of Molecular Biology* 179, 125–142.
- Forli, S., 2015. Charting a path to success in virtual screening. *Molecules* 20, 18732–18758.
- Gallivan, J.P., Dougherty, D.A., 1999. Cation- π interactions in structural biology. *Proceedings of the National Academy of Sciences of the United States of America* 96, 9459–9464.
- Gallo Cassarino, T., Bordoli, L., Schwede, T., 2014. Assessment of ligand binding site predictions in CASP10. *Proteins* 82 (Suppl 2), 154–163.
- González, M.M., Abriata, L.A., Tomatis, P.E., Vila, A.J., 2016. Optimization of conformational dynamics in an epistatic evolutionary trajectory. *Molecular Biology and Evolution* 33 (7), 1768–1776.
- Gu, Y., Li, D.-W., Brüsweiler, R., 2015. Decoding the mobility and time scales of protein loops. *Journal of Chemical Theory and Computation* 11, 1308–1314.
- Han, B., Liu, Y., Ginzinger, S.W., Wishart, D.S., 2011. SHIFTX2: Significantly improved protein chemical shift prediction. *Journal of Biomolecular NMR* 50, 43–57.
- Hanson, R.M., Prilusky, J., Renjian, Z., Nakane, T., Sussman, J.L., 2013. JSmol and the next-generation web-based representation of 3D molecular structure as applied to proteopedia. *Israel Journal of Chemistry* 53, 207–216.
- Harding, M.M., 2002. Metal-ligand geometry relevant to proteins and in proteins: Sodium and potassium. *Acta Crystallographica Section D, Biological Crystallography* 58, 872–874.
- Hendrickx, P.M.S., Gutmanas, A., Kleywegt, G.J., 2013. Vivaldi: Visualization and validation of biomacromolecular NMR structures from the PDB. *Proteins* 81, 583–591.
- Holm, L., Laakso, L.M., 2016. Dali server update. *Nucleic Acids Research* 44, W351–W355.
- Holm, L., Rosenström, P., 2010. Dali server: Conservation mapping in 3D. *Nucleic Acids Research* 38, W545–W549.
- Hooft, R.W., Vriend, G., Sander, C., Abola, E.E., 1996. Errors in protein structures. *Nature* 381, 272.
- Hopf, T.A., Colwell, L.J., Sheridan, R., *et al.*, 2012. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell* 149, 1607–1621.
- Hopf, T.A., Schärfe, C.P.I., Rodrigues, J.P.G.L.M., *et al.*, 2014. Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife* 3, e03430.
- Hrabe, T., Li, Z., Sedova, M., *et al.*, 2016. PDBFlex: Exploring flexibility in protein structures. *Nucleic Acids Research* 44, D423–D428.
- Huang, Y.J., Rosato, A., Singh, G., Montelione, G.T., 2012. RPF: A quality assessment tool for protein NMR structures. *Nucleic Acids Research* 40, W542–W546.
- Hussein, H.A., Borrel, A., Geneix, C., *et al.*, 2015. PockDrug-server: A new web server for predicting pocket druggability on holo and apo proteins. *Nucleic Acids Research* 43, W436–W442.
- Irwin, J.J., Shoichet, B.K., 2005. ZINC – A free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling* 45, 177–182.
- Irwin, J.J., Shoichet, B.K., 2016. Docking screens for novel ligands conferring new biology. *Journal of Medicinal Chemistry* 59, 4103–4120.
- Jack, B.R., Meyer, A.G., Echave, J., Wilke, C.O., 2016. Functional sites induce long-range evolutionary constraints in enzymes. *PLOS Biology* 14, e1002452.
- Jackson, M.R., Beahm, R., Duvvuru, S., *et al.*, 2007. A preference for edgewise interactions between aromatic rings and carboxylate anions: The biological relevance of anion-quadrupole interactions. *The Journal of Physical Chemistry B* 111, 8242–8249.
- Jo, S., Lim, J.B., Klauda, J.B., Im, W., 2009. CHARMM-GUI membrane builder for mixed bilayers and its application to yeast membranes. *Biophysical Journal* 97, 50–58.
- Johnson, G.T., Autin, L., Al-Alusi, M., *et al.*, 2015. cellPACK: A virtual mesoscope to model and visualize structural systems biology. *Nature Methods* 12, 85–91.
- Joosten, R.P., Long, F., Murshudov, G.N., Perrakis, A., 2014. The PDB_REDO server for macromolecular structure model optimization. *IUCr* 1, 213–220.
- Kamisetty, H., Ovchinnikov, S., Baker, D., 2013. Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proceedings of the National Academy of Sciences of the United States of America* 110, 15674–15679.
- Kassem, M.M., Wang, Y., Boomsma, W., Lindorff-Larsen, K., 2016. Structure of the bacterial cytoskeleton protein bactofilin by NMR chemical shifts and sequence variation. *Biophysical Journal* 110, 2342–2348.
- Kc, D.B., 2016. Recent advances in sequence-based protein structure prediction. *Briefings in Bioinformatics*. doi:10.1093/bib/bbw070.
- Khan, F.I., Wei, D.-Q., Gu, K.-R., Hassan, M.I., Tabrez, S., 2016. Current updates on computer aided protein modeling and designing. *International Journal of Biological Macromolecules* 85, 48–62.
- Khan, S., Vihinen, M., 2010. Performance of protein stability predictors. *Human Mutation* 31, 675–684.
- Kim, S., Thiessen, P.A., Bolton, E.E., *et al.*, 2016. PubChem substance and compound databases. *Nucleic Acids Research* 44, D1202–D1213.
- Kirschner, K.N., Yongye, A.B., Tschampel, S.M., *et al.*, 2008. GLYCAM06: A generalizable biomolecular force field. *Carbohydrates. Journal of Computational Chemistry* 29, 622–655.
- Kolesov, G., Virnau, P., Kardar, M., Mirny, L.A., 2007. Protein knot server: Detection of knots in protein structures. *Nucleic Acids Research* 35, W425–W428.
- Kumar, M., Balaji, P.V., 2014. C-H... π interactions in proteins: Prevalence, pattern of occurrence, residue propensities, location, and contribution to protein stability. *Journal of Molecular Modeling* 20, 2136.
- Lanzarotti, E., Biekořky, R.R., Estrin, D.A., Marti, M.A., Turjanski, A.G., 2011. Aromatic-aromatic interactions in proteins: Beyond the dimer. *Journal of Chemical Information and Modeling* 51, 1623–1633.
- Lindorff-Larsen, K., Piana, S., Dror, R.O., Shaw, D.E., 2011. How fast-folding proteins fold. *Science* 334, 517–520.
- Lomize, M.A., Lomize, A.L., Pogozheva, I.D., Mosberg, H.I., 2006. OPM: Orientations of proteins in membranes database. *Bioinformatics* (Oxford, England) 22, 623–625.
- London, N., Raveh, B., Cohen, E., Fathi, G., Schueler-Furman, O., 2011. Rosetta FlexPepDock web server – High resolution modeling of peptide-protein interactions. *Nucleic Acids Research* 39, W249–W253.
- Mabrouk, M., Putz, I., Werner, T., *et al.*, 2015. RBO Aleph: Leveraging novel information sources for protein structure prediction. *Nucleic Acids Research* 43, W343–W348.
- Madej, T., Lanczycki, C.J., Zhang, D., *et al.*, 2014. MMDB and VAST + : Tracking structural similarities between macromolecular complexes. *Nucleic Acids Research* 42, D297–D303.
- Maiti, R., Van Domselaar, G.H., Wishart, D.S., 2005. MovieMaker: A web server for rapid rendering of protein motions and interactions. *Nucleic Acids Research* 33, W358–W362.

- Maiti, R., Van Domselaar, G.H., Zhang, H., Wishart, D.S., 2004. SuperPose: A simple server for sophisticated structural superposition. *Nucleic Acids Research* 32, W590–W594.
- Mallocci, G., Vargiu, A.V., Serra, G., *et al.*, 2015. A database of force-field parameters, dynamics, and properties of antimicrobial compounds. *Molecules Basel Switzerland* 20, 13997–14021.
- Marks, D.S., Colwell, L.J., Sheridan, R., *et al.*, 2011. Protein 3D structure computed from evolutionary sequence variation. *PLOS ONE* 6, e28766.
- Marsh, J.A., 2013. Buried and accessible surface area control intrinsic protein flexibility. *Journal of Molecular Biology* 425, 3250–3263.
- Martinez, L., Andrade, R., Birgin, E.G., Martinez, J.M., 2009. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry* 30, 2157–2164.
- Morcos, F., Jana, B., Hwa, T., Onuchic, J.N., 2013. Coevolutionary signals across protein lineages help capture multiple protein conformations. *Proceedings of the National Academy of Sciences of the United States of America* 110, 20533–20538.
- Morcos, F., Pagnani, A., Lunt, B., *et al.*, 2011. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America* 108, E1293–E1301.
- Moult, J., Fidelis, K., Krysztofowicz, A., Schwede, T., Tramontano, A., 2016. Critical assessment of methods of protein structure prediction (CASP) – Progress and new directions in Round XI. *Proteins* 84 (S1), 4–14.
- Olsson, M.H.M., Søndergaard, C.R., Rostkowski, M., Jensen, J.H., 2011. PROPKA3: Consistent treatment of internal and surface residues in empirical pKa predictions. *Journal of Chemical Theory and Computation* 7, 525–537.
- Ortiz, A.R., Strauss, C.E.M., Olmea, O., 2002. MAMMOTH (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science* 11, 2606–2621.
- Ovchinnikov, S., Kamisetty, H., Baker, D., 2014. Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. *eLife* 3, e02030.
- Ovchinnikov, S., Kinch, L., Park, H., *et al.*, 2015. Large-scale determination of previously unsolved protein structures using evolutionary information. *eLife* 4. doi:10.7554/eLife.09248.
- Ovchinnikov, S., Park, H., Varghese, N., *et al.*, 2017. Protein structure determination using metagenome sequence data. *Science* 355, 294–298.
- Petrey, D., Chen, T.S., Deng, L., *et al.*, 2015. Template-based prediction of protein function. *Current Opinion in Structural Biology* 32, 33–38.
- Petrov, D., Zagrovic, B., 2014. Are current atomistic force fields accurate enough to study proteins in crowded environments? *PLOS Computational Biology* 10, e1003638.
- Piana, S., Lindorff-Larsen, K., Shaw, D.E., 2013. Atomic-level description of ubiquitin folding. *Proceedings of the National Academy of Sciences of the United States of America* 110, 5915–5920.
- Pundir, S., Martin, M.J., O'Donovan, C., UniProt Consortium, 2016. UniProt Tools. *Current Protocols in Bioinformatics* 53, 1.29.1–1.29.15.
- Rego, N., Koes, D., 2015. 3Dmol.js: Molecular visualization with WebGL. *Bioinformatics (Oxford, England)* 31, 1322–1324.
- Ringer, A.L., Senenko, A., Sherrill, C.D., 2007. Models of S/pi interactions in protein structures: Comparison of the H2S benzene complex with PDB data. *Protein Science* 16, 2216–2223.
- Rose, A.S., Bradley, A.R., Valasatava, Y., *et al.*, 2016. Web-based molecular graphics for large complexes. In: *Proceedings of the 21st International Conference on Web3D Technology*, New York, NY: ACM, pp. 185–186.
- Rose, A.S., Hildebrand, P.W., 2015. NGL Viewer: A web application for molecular visualization. *Nucleic Acids Research* 43, W576–W579.
- Sadowski, M.I., Taylor, W.R., 2012. Evolutionary inaccuracy of pairwise structural alignments. *Bioinformatics (Oxford, England)* 28, 1209–1215.
- Sarti, E., Gladich, I., Zamuner, S., Correia, B.E., Laio, A., 2016. Protein–protein structure prediction by scoring molecular dynamics trajectories of putative poses. *Proteins* 84, 1312–1320.
- Schmidtke, P., Le Guilloux, V., Maupetit, J., Tufféry, P., 2010. fpocket: Online tools for protein ensemble pocket detection and tracking. *Nucleic Acids Research* 38, W582–W589.
- Sehnal, D., Pravda, L., Svobodová Vařeková, R., Ionescu, C.-M., Koča, J., 2015. PatternQuery: Web application for fast detection of biomacromolecular structural patterns in the entire Protein Data Bank. *Nucleic Acids Research* 43, W383–W388.
- Shi, Z., Olson, C.A., Bell, A.J., Kallenbach, N.R., 2002. Non-classical helix-stabilizing interactions: C–H...O H-bonding between Phe and Glu side chains in alpha-helical peptides. *Biophysical Chemistry* 101–102, 267–279.
- Shindyalov, I.N., Bourne, P.E., 2001. A database and tools for 3-D protein structure comparison and alignment using the combinatorial extension (CE) algorithm. *Nucleic Acids Research* 29, 228–229.
- Sippl, M.J., Wiederstein, M., 2008. A note on difficult structure alignment problems. *Bioinformatics (Oxford, England)* 24, 426–427.
- Southan, C., Stracz, A., 2013. Extracting and connecting chemical structures from text sources using chemicalize.org. *Journal of Cheminformatics* 5, 20.
- Spiga, E., Abriata, L.A., Piazza, F., Dal Peraro, M., 2014. Dissecting the effects of concentrated carbohydrate solutions on protein diffusion, hydration, and internal dynamics. *The Journal of Physical Chemistry B* 118, 5310–5321.
- Spiliotopoulos, D., Kastiritis, P.L., Melquiond, A.S.J., *et al.*, 2016. dMM-PBSA: A new HADDOCK scoring function for protein-peptide docking. *Frontiers in Molecular Biosciences* 3, 46.
- Spyrakakis, F., Cavasotto, C.N., 2015. Open challenges in structure-based virtual screening: Receptor modeling, target flexibility consideration and active site water molecules description. *Archives of Biochemistry and Biophysics* 583, 105–119.
- Stansfeld, P.J., Goose, J.E., Caffrey, M., *et al.*, 2015. MemProtMD: Automated insertion of membrane protein structures into explicit lipid membranes. *Structure* 23, 1350–1361.
- Tamo, G., Abriata, L., Dal Peraro, M., 2015. The importance of dynamics in integrative modeling of supramolecular assemblies. *Current Opinion in Structural Biology* 31, 28–34.
- Tian, P., Boomsma, W., Wang, Y., *et al.*, 2015. Structure of a functional amyloid protein subunit computed using sequence variation. *Journal of the American Chemical Society* 137, 22–25.
- Touw, W.G., Vriend, G., 2014. BDB: Databank of PDB files with consistent B-factors. *Protein Engineering, Design and Selection* 27, 457–462.
- Touw, W.G., Joosten, R.P., Vriend, G., 2015. Detection of trans-cis flips and peptide-plane flips in protein structures. *Acta Crystallographica Section D, Biological Crystallography* 71, 1604–1614.
- Unni, S., Huang, Y., Hanson, R.M., *et al.*, 2011. Web servers and services for electrostatics calculations with APBS and PDB2PQR. *Journal of Computational Chemistry* 32, 1488–1491.
- Valasatava, Y., Rosato, A., Cavallaro, G., Andreini, C., 2014. MetalS(3), a database-mining tool for the identification of structurally similar metal sites. *Journal of Biological Inorganic Chemistry* 19, 937–945.
- Valley, C.C., Cembran, A., Perlmutter, J.D., *et al.*, 2012. The methionine-aromatic motif plays a unique role in stabilizing protein structure. *Journal of Biological Chemistry* 287, 34979–34991.
- Virag, I., Stoicu-Tivadar, L., Crișan-Vida, M., 2016. Gesture interaction browser-based 3D molecular viewer. *Studies in Health Technology and Informatics* 226, 17–20.
- Wass, M.N., Mooney, S.D., Linial, M., Radivojac, P., Friedberg, I., 2014. The automated function prediction SIG looks back at 2013 and prepares for 2014. *Bioinformatics (Oxford, England)* 30, 2091–2092.
- Weatheritt, R.J., Babu, M.M., 2013. Evolution. The hidden codes that shape protein evolution. *Science* 342, 1325–1326.
- Weichenberger, C.X., Sippl, M.J., 2007. NQ-Flipper: Recognition and correction of erroneous asparagine and glutamine side-chain rotamers in protein structures. *Nucleic Acids Research* 35, W403–W406.

- Wiederstein, M., Sippl, M.J., 2007. ProSA-web: Interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Research* 35, W407–W410.
- Willard, L., Ranjan, A., Zhang, H., *et al.*, 2003. VADAR: A web server for quantitative evaluation of protein structure quality. *Nucleic Acids Research* 31, 3316–3319.
- Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* 80, 1715–1735.
- Yang, H., Qureshi, R., Sacan, A., 2012. Protein surface representation and analysis by dimension reduction. *Proteome Science* 10 (Suppl 1), S1.
- Ye, Y., Godzik, A., 2004a. FATCAT: A web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Research* 32, W582–W585.
- Ye, Y., Godzik, A., 2004b. Database searching by flexible protein structure alignment. *Protein Science* 13, 1841–1850.
- Zhang, Y., Skolnick, J., 2005. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research* 33, 2302–2309.
- Zheng, H., Chordia, M.D., Cooper, D.R., *et al.*, 2014. Validation of metal-binding sites in macromolecular structures with the CheckMyMetal web server. *Nature Protocols* 9, 156–170.
- Zoete, V., Cuendet, M.A., Grosdidier, A., Michielin, O., 2011. SwissParam: A fast force field generation tool for small organic molecules. *Journal of Computational Chemistry* 32, 2359–2368.

Relevant Websites

<http://webchemdev.ncbr.muni.cz/LiteMol/>
 LiteMol.

<https://lucianoabriata.altervista.org/modelshome.html>
 Luciano A. Abriata.

https://www.mn-am.com/online_demos/corina_demo
 MNAM.

<http://psychoprot.epfl.ch/>
 PsychoProt.

<https://biasmv.github.io/pv/>
 PV - JavaScript Protein Viewer.

<http://research.bmh.manchester.ac.uk/bryce/amber>
 University of Manchester.

Biographical Sketch



Luciano Abriata is a postdoctoral researcher at the Laboratory for Biomolecular Modeling at École Polytechnique Fédérale de Lausanne and the Swiss Institute of Bioinformatics, Switzerland. His research focuses on the use of experiments and computation to achieve integrative models of biomolecular structure and function. Website: <http://lucianoabriata.altervista.org/>

Molecular Dynamics and Simulation

Liangzhen Zheng, Amr A Alhossary, Chee-Keong Kwoh, and Yuguang Mu, Nanyang Technological University, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

History

Molecular dynamics (MD) simulations have been evolved as powerful tools to explore atomic level of mechanisms, which are usually out the scopes of current experimental tools. Molecular dynamics are numeric tools based on the Newton's equations of motions for N -body system simulations by iteratively updating particle forces and system potential energies composed by inter-atomic/intra-atomic potentials or mechanic force fields. During MD simulations, coordinates, velocities, forces and potential energies are calculated every iterative step, whose temporal gap between each other is called "time step". The smaller the time step, the more accurate but more computational intensive the simulation is.

The original idea of MD simulations came up in 1950s in theoretical physics, but later it was also introduced to chemical, material science, and biological molecular simulations. Later in 1960s, Lennard-Jones potentials were introduced for lipid simulations (Rahman, 1964). However, at that time, computational power limited the simulation system size and time scale.

McCammon *et al.* (1977) reported the first MD simulation study of the dynamics of a protein bovine pancreatic trypsin inhibitor. Ever since then, more and more researches were carried out, and MD simulation has developed as a powerful tool in studying macromolecules and their interactions. Ogata *et al.* (2013) reported a work using MD simulation to study the photosynthesis system II, which is a signal that we are embracing a period where simulations of complicated systems become possible.

Anton, as well as Anton II, created by D.E. Shaw's group, as a specialized simulation machine, outperforms any other same scale computational devices in speed and efficiency. It was successfully used to perform long time scale all atomic simulations (Lindorff-Larsen *et al.*, 2011; Piana *et al.*, 2011; Shaw *et al.*, 2010).

As the knowledge of chemistry and physics increase, we would know much more about the micro-world behaviors. Along with the development of hardware of computers and the algorithms, we are now towards a theoretic biology era, when every experiment could be performed *in silico* rather than by bench works. This goal, using computers to simulate every aspect and every level of life sciences, is difficult to imagine but would be finally achieved.

Background/Fundamentals

Molecular dynamics (MD) simulation is the computer based N -Body simulation of systems containing atoms and molecules. In MD simulation, the system components are left to interact together for a predetermined period, while the system physical properties are monitored. Those properties are either macroscopic system properties, $\{V$ (volume), P (pressure), T (temperature), N (number of particles) $\}$, or microscopic system properties, $\{v_i$ (velocities), r_i (positions) $\}$.

Ensemble and trajectory

The output of the simulation comes in the form of ensemble of frames. All frames share the same macroscopic/thermodynamic state but may differ in the microscopic states. Each frame represents the system at a specific point of time (a specific microscopic state). If the ensemble is sequence (time) dependent, it is called a trajectory. In this case, the trajectory represents the time-dependent evolution of the system.

Canonical ensemble (NVT)

The canonical ensemble contains all possible states in thermal equilibrium with a heat bath. The system remains in the absolute temperature T but may exchange energy with the heat bath. Three parameters of the system are fixed throughout the simulation: the absolute temperature (T), the number of atoms (N), and the volume (V). T is the most influential parameter on the system states among them.

Micro canonical ensemble (NVE)

The micro canonical ensemble represents an isolated system. No change in mass/number of atoms (N), Volume (V), nor exchange of Energy (E) is allowed.

Isothermal–isobaric ensemble (NPT)

The system has a fixed temperature (T), hence it is isothermal, and fixed pressure (P), hence it is isobaric.

Force field

Nearly all the simulation methods rely on force field for calculation. Force field, a potential energy (Eq. (1)) field in atomic and molecular level, describes the topology and motion behavior of atoms in molecules. When we use force field to describe the properties of molecules, spectrum constant force field and empirical potential function force field are often introduced. There are also Dreiding force field and universal force field (Mayo *et al.*, 1990; Setubal and Meidanis, 1997) which are not discussed here in detail.

$$E_{total} = E_{kinetic} + U_{potential} \quad (1)$$

In spectrum constant field, the energy is simply linked with distance between two atoms. (See Eq. (2), K_i is a constant measured by spectrum data, ΔR_i is the distance between atom i and atom j .) This only applies for simple molecules such as water molecules.

$$E_{vibrant} = \frac{1}{2} \sum_i K_i (\Delta R_i)^2 \quad (2)$$

For multi-atom molecules, internal coordinates are introduced. Four internal coordinates are commonly used as depicted in Fig. 1. These four coordinates (bond stretching, bond angle, torsion angle and out-of-plane angle) do not affect the overall movements in the space, but they are enough to describe the internal motion of the molecules.

In other simulation methods, potential energy is composed by several terms: non-binding potential, bonding stretching term potential, angle bending term potential, torsion (dihedral) angle term potential, out-of-plane bending term potential, columbic interaction term potential (Frenkel and Smit, 2002). For many force fields, the calculations are based on the terms mentioned above. The most frequently applied non-binding potential is described by Lennard-Jones (LJ) potential force. (See Eq. (3), r is distance between two atoms, ε and σ are atomic specific constants).

$$U_r = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (3)$$

In Eq. (3), for the U_r of two atoms A and B, the parameter σ could be approximated by the following equation. (See Eq. (4)).

$$\sigma_{AB} = \frac{1}{2}(\sigma_A + \sigma_B), \varepsilon_{AB} = \sqrt{\varepsilon_A \varepsilon_B}; \quad (4)$$

$$U_b = \frac{1}{2} \sum_i k_b (r_i - r_i^0)^2 \quad (5)$$

Eq. (5) is a common expression of bonding stretching term potential function (a simple harmonic vibration model). r_i is the bond length of the no. i th bond, while r_i^0 is the average bond length of the bond i :

$$U_\theta = \frac{1}{2} \sum_i k_\theta (\theta_i - \theta_i^0)^2 \quad (6)$$

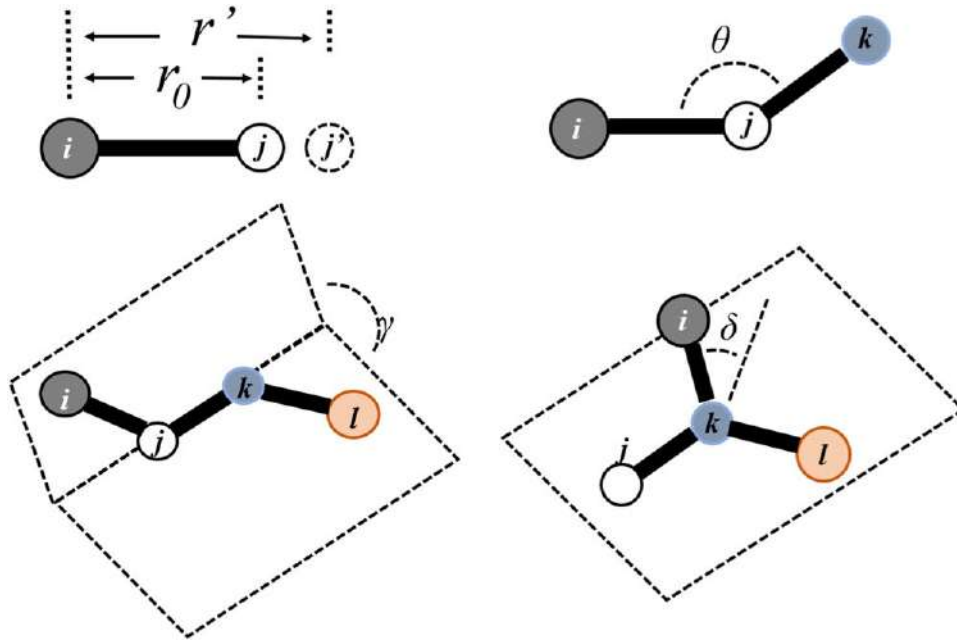


Fig. 1 Four types of internal coordinates used to potential energies in MD force fields. The 4 types of internal coordinates are bond, angle, dihedral angle, and out-of-plane angle.

$$U_{\tau} = \frac{1}{2} \sum_i [V_1(1 + \cos\tau) + V_2(1 - \cos 2\tau) + V_3(1 + \cos 3\tau)] \quad (7)$$

Common bending angle term potential is defined by Eq. (6). θ_i is the angle between the two bonds on the atom i . The Eq. (7) concerns the torsion angle term potential energy. While τ is torsion angle using the no. i atom as the center. V_1 , V_2 and V_3 are force constants.

$$U_{\alpha} = \frac{1}{2} \sum_i k_{\alpha} \alpha^2 \quad (8)$$

Eq. (8) is out-of-plane term potential energy function. k_{α} is a constant and α is the out-of-plane angle.

$$U_{\text{coulombic}} = \sum_{ij} \frac{q_i \cdot q_j}{D \cdot r_{ij}} \quad (9)$$

This coulombic term potential energy is the electric energy between the no. i and no. j atom. D is dielectric constant, and r_{ij} is the distance between two atoms. q is the electric charge of atoms.

Most famous classical force fields used for modeling of *Macromolecules* are GROMOS (Groningen Molecular Simulation) (Christen *et al.*, 2005; van Gunsteren *et al.*, 1996), AMBER (Assisted Model Building with Energy Refinement) (Cornell *et al.*, 1995), and CHARMM (Chemistry at HARvard Macromolecular Mechanics) (Brooks *et al.*, 1983). CHARMM can also be used for *small molecules*. OPLS (Optimized Potentials for Liquid Simulations) (Jorgensen *et al.*, 1996; Jorgensen and Tirado-Rives, 1988) is used to model *liquids* and it can as well be used in *Energy minimization*.

Another group of force fields called second generation force fields, integrates more complicated quantum mechanics and experimental results to pursuit more precise simulation. Some examples are CFF91, PCFF, CFF95, MMFF94 (Halgren, 1996), etc. There are also some special force fields (such as COMPASS and Gay-Berne) “costumed” for specific molecules, and thus are not explained in detail here.

Methodology

The total energy of a system is the sum of both kinetic and potential energies. The kinetic energy can be calculated from the atoms/particles velocities, while the potential energy is related to the positions of all atoms/particles in the 3 directions (that is $3N$, where N is number of atoms). This is too big number of parameters to have any analytical solution. Therefore, numerical methods only can be used to calculate.

Integrators

The MD simulation is based on Newton's second law of motion, ($F=ma$), where F is the force affecting a particle, m is the particle mass and a is its acceleration. Most integrators assume that the velocity and acceleration can be approximated using Taylor's expansion. From particles' relative position (r), velocity (v), and acceleration (a), the time series time-dependent evolution of the system can be predicted step wise.

$$r(t + \delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2 + \dots$$

$$v(t + \delta t) = v(t) + a(t)\delta t + \frac{1}{2}b(t)\delta t^2 + \dots$$

$$a(t + \delta t) = a(t) + b(t)\delta t + \dots$$

Most famous integrators are *Verlet*, *Leapfrog*, and *Velocity Verlet*.

Methods to Overcome Limitations and Decrease Load

The calculation complexity of the MD simulation problem is a $O(n^2)$ problem. This limits the ability to simulate large systems. Methods exist to minimize the computational load. Some of them are mentioned next sections.

Neighborhood list

Having a neighborhood list for each atom, reduces the calculation complexity from $O(n^2)$ to $O(n)$. The neighborhood list needs to be updated periodically, in a suitable frequency: Neither too infrequent so that important updates may be missed, nor so frequent that the calculation efficiency gained in reduced complexity calculations may be lost.

Periodic boundary check (PBC)

To neglect the solvent surface tension effect, there are two options: having an infinite amount of solvent, which is impractical, or to have the periodic boundary check (PBC). In the PBC, every atom is considered replicated to image atoms in all directions.

PME, PPPM

Another advantage of the PBC is the ability to model long interactions as periodic interactions to be solved in the frequency domain. Particle Mesh Ewald summation (PME) is an efficient way to achieve this goal which depends on sorting and calculating image atoms in a certain order that simplifies the calculation. Another advancement of PME method is “Particle-Particle, Particle-Mesh” (PPPM) method. In this method, the short range interactions are calculated as particle-particle interactions, while for the long range interactions, it is calculated as follows: first, all particles effect is approximated to a set of charges spread on a mesh. The long range interactions with the mesh are calculated efficiently in the frequency domain.

Advanced Techniques

The large-scale conformation differences between meta-stable states of macromolecules are often separated by high energy barriers which are unlikely accessible through conventional MD simulations. Based on the transition state theory, there is an exponential relationship between the time scale of state transition and the height of an energy barrier (Huang, 1987). Therefore, the rare dynamics events of interest, e.g., the folding process of a polypeptide, are often happening in a rather large time scale (Kubelka *et al.*, 2004). For example, according to Kubelka *et al.* (2004) the folding time limit of a generic single-domain protein from unfolded to folded state, could be expressed as following: $N/100 \mu s$, where N is the number of residues in the protein. In order to obtain a statistical meaningful transition free energies, multiple times trans-passing of the energy local minima are required based on the assumption of the ergodicity (Tolman, 1938) for conventional MD simulations, which probably requires extremely long simulation time. Therefore, for large proteins, using conventional MD simulations to sample the large energy barrier crossing events could be an impossible mission. However, a number of enhanced sampling methods, such as the replica exchange MD simulation (REMD) method (Sugita and Okamoto, 1999a), solute tempering (Liu *et al.*, 2005), Hamiltonian replica exchange MD (HREMD) (Liu *et al.*, 2005), Wang-Landau method and simulated tempering (ST) (Marinari and Parisi, 1992; Zhang *et al.*, 2015), have been developed to address the difficulties. Besides, other enhanced sampling methods, including umbrella sampling (Torrie and Valleau, 1977) and metadynamics (Laio and Parrinello, 2002), based on introduced bias potentials, could reconstruct the probability distribution along one or a few CVs (Sutto *et al.*, 2012). What's more, the combinations of these methods are also proved to be efficient tools to explore the free energy surface (FES). For example, the hybrid Hamiltonian method by introducing the Generalized Born energy for polar solvation energy calculations, is a useful tool to quickly reconstruct the FES which is similar to the standard REMD simulation (Mu *et al.*, 2007). Also the combination of parallel tempering with metadynamics by Bussi *et al.* is also proved to be powerful (Bussi *et al.*, 2006b).

Umbrella sampling

Umbrella sampling (Torrie and Valleau, 1974, 1977), as an enhanced sampling method, could sample the conformational dynamics along reaction coordinates thus we could estimate the relative free energy of different states along the reaction coordinates. A reaction coordinate (ξ) could be a kind of continuous parameter which could describe the system from a higher dimensional space. If the reaction coordinate is good enough to differentiate distinct states, by biasing the system along the reaction coordinate, thus we would calculate the free energy differences between the states.

In a general umbrella sampling scheme, multiple windows were set with initial structures with different reaction coordinate values. A bias potential is applied to each window (Eq. (10)) according to a harmonic bias function (Eq. (11)) or an adaptive bias function (Kästner, 2011). Therefore, the system in each window would be constrained to sample a narrow phase space along the reaction coordinate to ensure potential energy distribution overlap between adjacent windows. After the simulations, a post-processing method, weighted histogram analysis method (WHAM) (Rosenberg, 1992), could recover the unbiased free energy by the umbrella integration. Multiple integration methods and biasing potential styles are existed, among them, WHAM as the integration method, harmonic bias potential as the biasing potential style are relatively widely used. The following part is a brief introduction to the general procedures of umbrella sampling using harmonic bias potential and WHAM post-processing method.

When we add a reaction coordinate dependent bias potential $\omega_i(\xi)$ to the system in a window, the total biased energy could be expressed as following:

$$E_{bias} = E_{unbias} + \omega_i(\xi) \quad (10)$$

where i represents the i th window of the umbrella sampling. The harmonic bias potential adding to the system is a simple bias potential expressed like this:

$$\omega_i(\xi) = \frac{1}{2} K (\xi - \xi_i)^2 \quad (11)$$

Here ξ_i works as a reference coordinate point. During the simulations, if the system is escaping the reaction coordinate, then a bias potential is added to push the system back.

To calculate the unbiased free energy, we need to obtain the unbiased distribution of the reaction coordinate, according the following equation:

$$P_i^u(\xi) = \frac{\int \exp[-\beta E(r)] \delta[\xi'(r) - \xi] d^{N_r}}{\int \exp[-\beta E(r)] d^{N_r}} \quad (12)$$

Furthermore, the unbiased probability $P_i^u(\xi)$ could be determined by:

$$P_i^u(\xi) = P_i^b(\xi) \exp[\beta \omega_i(\xi)] \exp[-\beta \omega_i(\xi)] \quad (13)$$

From the simulation in each window, the biased probability $P_i^b(\xi)$ is known, and the free energy of the window thus could be induced by:

$$E_{unbias} = - \left(\frac{1}{\beta} \right) \ln P_i^b(\xi) - \omega_i(\xi) + F_i \quad (14)$$

where F_i is a constant which could be solved by self-iteration until a convergence reached (Banavali and Roux, 2005). Besides, the unbiased free energy could be combined by all the windows to produce a whole free energy along the reaction coordinate.

Metadynamics simulation

Different from the conventional MD simulations, metadynamics (Laio and Parrinello, 2002) adopts gaussian-like bias to bypass high energy barriers in FES. By adding bias potentials, the predefined collective variables (CVs), metadynamics sampling could reconstruct the FES based on the biased CVs (Fig. 2) and be able to accelerate the sampling process to capture rare dynamics events, such as the large-scale conformation changes.

Ever since the first algorithm of the metadynamics (Laio and Parrinello, 2002), many modified versions of metadynamics have been proposed, such as well-tempered (WT) metadynamics (Barducci *et al.*, 2008), the bias-exchange approach (Piana and Laio, 2007) and the parallel tempering (PT) metadynamics (Bussi *et al.*, 2006a). Besides, new CVs as reviewed in the reference (Sutto *et al.*, 2012), such as widely used path variables, principle component analysis (PCA) variables, alpha RMSD, beta RMSD, and contact map variables, have accelerating the sampling efficiency vastly.

For a general understand of the metadynamics, we take the well-tempered metadynamics (Barducci *et al.*, 2008) as an example, and briefly introduce the rationales behind. Supposing that the microscopic coordinate $\mathbf{R}$ at temperature T has a potential $U(\mathbf{R}, t)$, the ultimate goal of metadynamics is to obtain the FES from the unbiased probability distribution $P(s(\mathbf{R}))$ along the CV $s(\mathbf{R})$ by solving the following equation:

$$F(s) = - \frac{1}{\beta} \log P(s) \quad (15)$$

where the $\beta = \frac{1}{k_B T}$, k_B is the Boltzmann constant. For the unbiased, $P(s)$ we could have:

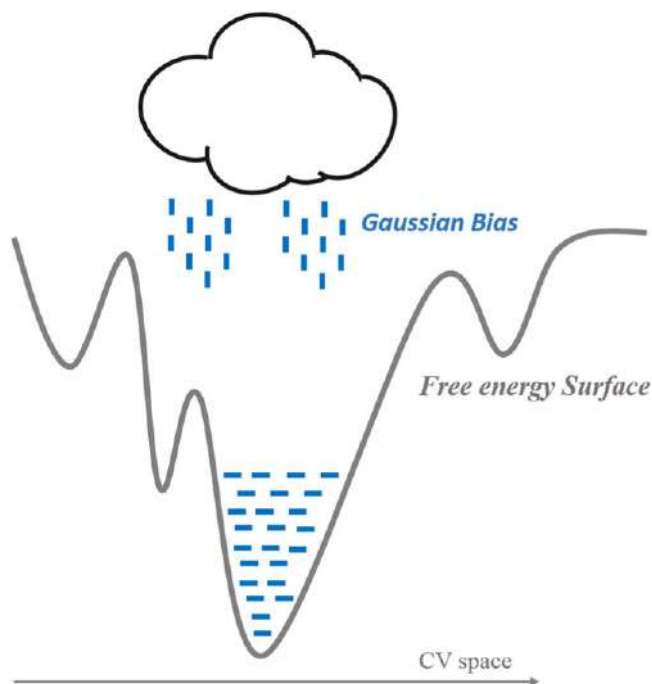


Fig. 2 A cartoon representation of metadynamics. During metadynamics simulations, gaussian shape biases are deposited along specific coordinates (or CVs).

$$P(R, t) = \frac{e^{-\beta U(R)}}{\int dR e^{-\beta U(R)}} \quad (16)$$

By adding a history-dependent bias potential as the following

$$V(s, t) = \Delta T \ln \left(1 + \frac{\omega N(s, t)}{\Delta T} \right) \quad (17)$$

where $N(s, t) = \int_0^t \delta_{s,s(t')} dt'$ is the histogram of the CV $s(\mathbf{R})$, ω is energy related term, ΔT is a temperature which we would explain in detail later. For this equation, $V(s, t)$ disfavors the frequently visited space in CV $s(\mathbf{R})$. The first order derivate of the $V(s, t)$ could be expressed as following:

$$\dot{V}(s, t) = \frac{\omega \Delta T \delta_{s,s(t)}}{\Delta T + \omega N(s, t)} = \omega e^{-\frac{V(s, t)}{\Delta T}} \delta_{s,s(t)} \quad (18)$$

Then we use τ_G to replace $\delta_{s,s(t)}$, and let $w = \omega e^{-\frac{V(s, t)}{\Delta T}} \tau_G$, where τ_G is the Gaussian height deposit time step. By incorporating the Eq. (18) into the standard metadynamics, where the height of Gaussian deposited is a constant, we could modulate the bias potential as a history dependent style.

As the time t flows, $\dot{V}(s, t)$ could be determined by $\frac{V(s, t)}{\Delta T}$, which indicates at position where $V(s, t)$ is large, the $\dot{V}(s, t)$ is close to zero to resemble a thermodynamic equilibrium. At this situation, one might get $P(s, t) ds \propto e^{-\frac{F(s) - V(s, t)}{T}} ds$, based on that, then one could have the following relationship between $\dot{V}(s, t)$ and $F(s)$:

$$\dot{V}(s, t) = \omega e^{-\frac{V(s, t)}{\Delta T}} P(s, t) = \omega e^{-\frac{V(s, t)}{\Delta T}} \frac{e^{-\frac{F(s) - V(s, t)}{T}}}{\int ds e^{-\frac{F(s) - V(s, t)}{T}}} \quad (19)$$

While the time $t \rightarrow \infty$,

$$V(s) = -\frac{\Delta T}{T + \Delta T} F(s) \quad (20)$$

If we go a step further, we could have $F(s) + V(s) = \frac{T}{T + \Delta T} F(s)$. Until now, we have established the relationship between $V(s, t)$ with $F(s)$ in a simpler way. Combining Eq. (19) with Eq. (20), we then can deduce the estimation of $F(s)$:

$$\tilde{F}(s, t) = -\frac{T + \Delta T}{\Delta T} V(s, t) = -(T + \Delta T) \ln \left(1 + \frac{\omega N(s, t)}{\Delta T} \right) \quad (21)$$

From Eq. (21), we could estimate the final FES by calculating the histogram $N(s, t)$ of CV $s(\mathbf{R})$. The modulation of the constant ΔT could define different behavior of the Gaussian deposit. For $\Delta T = 0$, the bias potential added in the system is always 0 along the simulation time t . Another limiting case, if $T \rightarrow \infty$, then $-\frac{T + \Delta T}{\Delta T} \rightarrow -1$, thus we could give the conclusion that $\tilde{F}(s, t) \approx -V(s, t)$, which is the case in the standard metadynamics where the bias potential energy, added in the system along the CVs space, has the same absolute value as the FES along the same CVs space. What's more, a finite value of ΔT , therefore restricts the system from exploring all the physical space and ensures a thermodynamic equilibrium at local minima in FES. Fine tuning of this ΔT could facilitate us to better explore the CVs space. By calculating the $V(s, t)$ in the following equation:

$$V(s, t) = w \sum_{t' = \tau_G, 2\tau_G, 3\tau_G, \dots} e^{\left(\frac{-s(\mathbf{R}) + s(R_G(t'))}{2\delta^2} \right)^2} \quad (22)$$

where w and δ have been defined already. Finally, we could recover the $\tilde{F}(s, t)$ from the biased probability distribution of $s(\mathbf{R})$.

In practice, we often use the following bias factor $\gamma = \frac{T + \Delta T}{T}$ to set up the ΔT . τ_G is the Gaussian deposition stride, and another input value ω in $w = \omega e^{-\frac{V(s, t)}{\Delta T}} \tau_G$, is the initial Gaussian height. And the δ is the width of the Gaussian for the CV $s(\mathbf{R})$.

Plumed (Bonomi et al., 2009) and METAGUI (Biamés et al., 2012) (a VMD plugin), are example softwares which make the simulation and the post-process of the data much more convenient. The usage and tutorials of the tool could be found here: see "Relevant Websites section".

Replica exchange MD (REMD)

Conventional MD simulation takes long time simulation to sample large conformational changes, or observe biological relevant behaviors. In most cases, simulation system would be trapped in local potential energy minima, which limit the sampling efficiency. To overcome the potential energy barriers and search for the global energy basins, REMD (sometimes also called generalized ensemble method), emerged as an efficient sampling technique (Sugita and Okamoto, 1999b; La Penna et al., 2004; Mitsutake et al., 2001; Tai, 2004). A lot of researches have demonstrated that the REMD method is much more efficient than the conventional MD (García and Sanbonmatsu, 2002; Sugita and Okamoto, 1999b).

In practice, M replicas in canonical ensemble are simulated with fixed distinct temperatures independently and spontaneously for some MD (or MC) steps, though the optimal number of steps between exchanges are controversial (Sindhikara et al., 2008; Cecchini et al., 2004; Zhang et al., 2005b). Then, the adjacent replicas (say m and $m + 1$) could exchange their configurations based

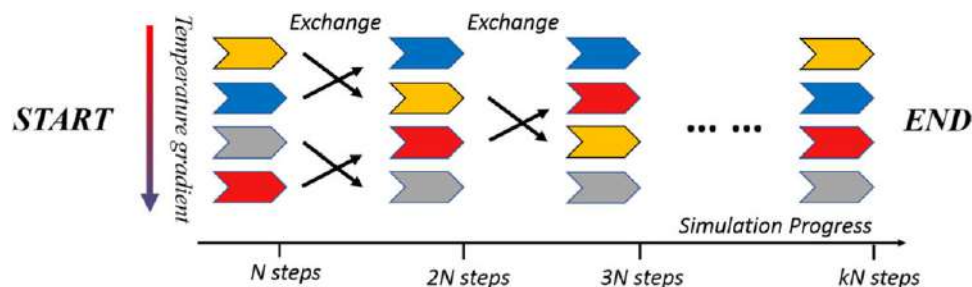


Fig. 3 A cartoon representation of REMD methods.

on the Metropolis criterion (Swendsen and Wang, 1987). Once the exchange occurs, replica m swamps the configuration with replica $m + 1$ and continues at the fixed temperature. Fig. 3 shows a sketch representation of the idea.

The idea was based on the hypothesis that at higher temperature, the potential energy of the simulation system is much higher and the system is much easier to escape the local energy minima to explore global energy minima. The potential energies of the parallel systems (replicas) in different simulating temperatures may overlap, thus the low energy conformations from high temperature replicas could be exchanged to lower temperature replicas and stabilized and sampled in lower temperature replicas. This way, in normal temperature range, the replicas could sample the structures with low potential energy, and the sampling efficiency therefore suppresses the conventional single trajectory MD simulation.

Here, we briefly illustrate the exchange method, the Metropolis scheme, used in REMD. The Metropolis criterion specifies that the exchange between neighboring replicas m and $m + 1$ at temperature T_m and T_{m+1} could take place with an acceptance probability P to ensure the detailed balance, $P = e^{(\beta_m - \beta_{m+1})(F_m - F_{m+1})}$, where F_m and F_{m+1} are the potential energies of replica m and replica $m + 1$. And $\beta_m = 1/(k_B T_m)$, where k_B is the Boltzmann's constant. To achieve acceptable exchange ratio, certain temperature intervals between replicas should be selected to ensure enough overlapping of the potential energies of neighboring replicas (Kone and Kofke, 2005; Sugita and Okamoto, 1999b).

The convergence problem is one of the major issues to achieve a statistical meaningful REMD simulation (Lin and Shell, 2009; Denschlag *et al.*, 2008; Okur *et al.*, 2007; Zhang *et al.*, 2005a). It has been widely discussed in many researches and reviews, therefore we are not going to elucidate it in detail here.

Hamiltonian REMD

The conventional REMD tends to sample low energy structure and then deposits it in the low temperature replicas, thus it is blind sampling and requires no prior knowledge of the simulation system. During REMD simulation, the number of required replicas to ensure enough potential energy overlapping between adjacent replicas is exponentially increased along with the system size, therefore relies on tremendous amount of computational power, and increases the difficulty of convergence, thus those limit the usage of REMD for huge simulation systems.

Instead of exchanging between different temperatures, Hamiltonian could also be exchanged between replicas if there are possible potential energies overlapping (Affentranger *et al.*, 2006; Fukunishi *et al.*, 2002; Curuksu and Zacharias, 2009). Hamiltonian REMD is quite similar to REMD, where each replica samples independently in different environments (temperature for REMD, Hamiltonian for Hamiltonian REMD respectively). The exchanges every N step in Hamiltonian REMD also obey the metropolitan rule.

For Hamiltonian REMD, the Hamiltonian of each replica is scaled by a factor λ . It has to be ensured that the first replica has a normal Hamiltonian, while other λ values increase or decrease sequentially. In the same time, the λ values should be set appropriately thus enough overlapping of potential energies between adjacent replicas must be retained. There are several ways to modulate the Hamiltonian of simulation systems, for example, reducing the atomic charges of a molecule, increasing intra-molecular repulsions (Mu, 2009), decreasing interatomic attractions, increasing molecular hydrophobic interactions, or modifying the atomic charges of atoms.

The combination of Hamiltonian simulation with REMD is proved to be a good practice. Using different Hamiltonian, we could actually bias the systems towards different directions. Hamiltonian REMD thus is not blindly sampling, nor the CV based sampling, but something in between.

Monte Carlo Simulation

The essence of Monte Carlo (MC) simulation lies on the "randomness". Multiple repeated random sampling using MC simulation could obtain statistical data for systems with multiple degrees of freedom. However, in general, in order to sample the configurational space of the system sufficiently, brute force MD method requires tons of millions of repeats to complete, therefore, analytic techniques must be used (Frenkel and Smit, 2001; Heermann, 1990).

Metropolis method was introduced to facilitate random sampling in MC simulation. In each step, a small random displacement Δ of a particle is tried. The displacement Δ is generated from a relative probability distribution proportional to the Boltzmann

factor $e^{-\beta U(o)}$, where β is $\frac{1}{k_B T}$. $U(o)$ is the potential of the system in previous step. This trial displacement is then assessed to accept or reject based on many rules, one of which is the Metropolis scheme, simple but applicable (Kalos and Whitlock, 2008).

To make it simpler, the system in previous state is called o (old), the system state after the displacement is denoted as n (new). The transition probability from o to n , therefore is determined here:

$$P(o \rightarrow n) = e^{-\beta[U(n) - U(o)]} \quad (23)$$

$U(n)$ and $U(o)$ are the potential energies of the old state and the new state. After the trial displacement, a random number r from the uniform distribution between 0 and 1 is generated and compared to $P(o \rightarrow n)$. The displacement would be accepted if $r < P(o \rightarrow n)$, otherwise, it would be rejected. This way, it is ensured that the acceptance probability from state o to n is equal to $P(o \rightarrow n)$.

The Metropolis scheme is efficient, but only thought to be ergodic for simple systems, therefore, the mixed schemes with efficient Metropolis scheme with other in principle ergodic scheme would achieve an statistical reliable simulation (Frenkel and Smit, 2001). In contrast to MD simulation, in general, MC simulation focuses on sampling, rather than time-dependent events, therefore it is not advisable to extract time dependent phenomena from the MC simulations.

Systems/Applications

Molecular dynamics have been widely adopted in molecular modeling of organic chemicals, short peptides (Woutersen *et al.*, 2002; Snow *et al.*, 2002; Marinelli *et al.*, 2009; Gnanakaran *et al.*, 2003; Blanco *et al.*, 1994; Ma and Nussinov, 2002), protein-protein interactions (Arkin *et al.*, 2014; Baaden and Marrink, 2013; Rajamani *et al.*, 2004), lipid-protein complexes (Shrivastava and Sansom, 2000; Im and Roux, 2002; Khalili-Araghi *et al.*, 2009; Gumbart *et al.*, 2005; Woutersen *et al.*, 2002; Snow *et al.*, 2002; Marinelli *et al.*, 2009; Gnanakaran *et al.*, 2003; Blanco *et al.*, 1994; Ma and Nussinov, 2002), even virus capsid (Freddolino *et al.*, 2006; Miao *et al.*, 2010). By harnessing the computation power from the supercomputing facilities, the physical properties of sub-micrometer particles could be revealed. Using MD simulations, as well as other simulation methods, stable conformations, the folding process, energy landscape, transitions between macro-states, enzyme catalytic mechanisms, ligand binding/association and aggregation states could be examined. Besides, MD simulations are also commonly used in computational drug discovery and design (Borhani and Shaw, 2012; De Vivo *et al.*, 2016; Durrant and McCammon, 2011).

For example, the enhanced sampling methods, umbrella sampling and metadynamics simulation could be applied to recover the free energy landscape of a macromolecule system along some specific collective variables, such as RMSD, PCA components, helical contents, angles, and other high dimensional coordinates. Bias potentials are deposited in simulations and the post-processing reweight techniques would then recover the true energy landscape of the simulation system along selected dimensions. The combination of metadynamics with REMD, or parallel exchanges between replicas biasing difference collective variables are proved to be more efficient and converges faster than the single-trajectory metadynamics simulations (Abrams and Bussi, 2013; Piana and Laio, 2007; Barducci *et al.*, 2011).

By incorporating multiple short MD simulations, Markov state model (HMM) could be applied to analyze the trajectories and compute the transition dynamics of proteins or protein-ligand systems between macro-states of the systems (Beauchamp *et al.*, 2012; McGibbon and Pande, 2013; Suárez *et al.*, 2016). Starting from different initial structures, it is expected that multiple short simulations starting in different directions are more powerful in conformational sampling than a single long trajectory. Combining the short simulations, several macro-states of the system could be identified using different methods, such as clustering and independent component analysis. The transition probabilities between the microstate then would be derived from the trajectories, and More importantly, the transition time scales could be estimated.

What's more, combining MD simulation with quantum calculation, more accurate enzyme activation mechanisms (van der Kamp and Mulholland, 2013; Hu *et al.*, 2011; Riccardi *et al.*, 2010; Senn and Thiel, 2007; Zhang *et al.*, 2000). The statistic properties could also be calculated and compared to experiments.

MD Simulation With Molecular Docking

Molecular docking is a simplified form of MD simulation. It can be used on intervals to replace lengthy segments of MD simulation trajectories, especially in cases where certain domains undergo large translations, rotations, and conformation changes. A typical example is biological interactions that include large protein folding, like capsid or vesicle formation. To be able to replace lengthy MD simulation intervals with docking, it is necessary to be able to perform "blind docking" of a protein (or a domain) to the whole surface of another protein. Some tools claim the ability to perform blind docking, for example QuickVina-W (111), PatchDock (222), BSP-SLIM (333) (Nafisa *et al.*, 2017; Schneidman-Duhovny *et al.*, 2005; Lee and Zhang, 2012).

Markov State Model (MSM)

The dynamics and kinetics properties are extremely valuable for a deep understanding of the structure-function relationships. MD simulation method has its innate advantage over current experimental techniques in exploring macromolecules kinetics and dynamics. By using MSM, or Hidden Markov State Model (HMM), the dynamics properties of proteins, as well as other

macromolecules, could be estimated with near to experiment accuracy (Beauchamp *et al.*, 2012; Suárez *et al.*, 2016; McGibbon and Pande, 2013; Noé *et al.*, 2009; McGibbon *et al.*, 2014; Husic and Pande, 2018).

Here we briefly explain the four general steps applied in most MSM studies for macromolecules dynamics and kinetics.

1. Extensive short MD simulations. The idea of constructing transitions between macro-states is based on such a hypothesis that we could adequately sample all the microstates, as well as their transitions. Therefore, in practice, other than a single long MD trajectory, multiple short MD simulations could be performed to sample as many intermediate states as possible (Noé *et al.*, 2009). Certain intermediate structures are selected as "seeding" initial structures for further high throughput short MD simulations, or adaptive sampling (Singhal and Pande, 2005; Pande *et al.*, 2010; Doerr and De Fabritiis, 2014; Doerr *et al.*, 2016).
2. Features dimension reduction. The features harnessed in previous step are then transformed into low dimensional vectors and clustering of the vectors could be applied to capture the microstates of the conformations. The coordinates of the trajectories are transformed into vector features. The features could be RMSD of a structure, dihedral angles of backbone atoms, distance between alpha-carbon atoms, or contacts between alpha-carbon atoms (Naritomi and Fuchigami, 2011; Pérez-Hernández *et al.*, 2013). Radius of gyrate, coordination number, PCA and time lagged independent component analysis (tICA) (Schwantes *et al.*, 2015; Noé and Clementi, 2015) are commonly applied in this dimension reduction step.
3. Microstates clustering. The distance matrices with clustering strategy is proved to be reliable for partitioning the phase space (Pande *et al.*, 2010; Chodera and Noé, 2014; McGibbon and Pande, 2013). The conformations from trajectories would be clustered, based on the low dimensional vectors calculated in step 2, to form microstates which could be rapidly interconverted between each other. Several commonly used clustering methods include K-mean-like clustering and hierarchical clustering. For an easy visualization of the MSM network, fewer macrostate are chosen, using the so-called "lumping" scheme using principal canonical correlation analysis (PCCA) or PCCA + methods.
4. MSM construction. Then using maximum likelihood method to estimate the transition rates between the states, and then construct the MSM, and estimate the kinetics of the MSM network (Weber, 2011; Röblitz and Weber, 2013).

Currently, there are majorly two tools, Pyemma (Scherer *et al.*, 2015) and MSMBuilder (Harrigan *et al.*, 2017), available for MSM or HMM construction. Their tutorials and manuals could be found here: see "Relevant Websites section" and see "Relevant Websites section".

MD Simulation With Machine Learning

The combination of machine learning and chemical informatics in drug discovery has been proved to be very successful for predicting ligand binding affinity (Ballester and Mitchell, 2010; Cheng *et al.*, 2012), drug toxicity (Walters and Murcko, 2002; Judson *et al.*, 2008), membrane permeability (Doniger *et al.*, 2002; Hou *et al.*, 2006; Walters and Murcko, 2002) and so on. In structural biology, machine learning models could advance the protein-protein binding interface prediction (Lise *et al.*, 2009), protein folding prediction, and protein structure prediction (Hua and Sun, 2001; Cai *et al.*, 2002; Heffernan *et al.*, 2015; Wang *et al.*, 2016).

Recently, machine learning algorithms have also been introduced into MD simulations for faster, more accurate sampling and predictions (Behler, 2016), for example, in metadynamics simulations. Proper CVs should be carefully chosen to achieve successful sampling. A recent work illustrates the probability of using neural network to reduce dimensions in tICA calculation and aiding the enhanced sampling (Sultan *et al.*, 2018). Or using machine learning algorithms to perform or instruct the way of bias quantum deposit to achieve high dimensional enhanced sampling (Galvelis and Sugita, 2017).

Quantum calculation based first-principles molecular dynamics (FPMD) are useful in understanding chemical processes, however, FPMD could only simulate up to hundreds of atoms within a very short simulation time to the limitations of computation power. There have been a lot of studies trying to adopt machine learning methods, such as neural network and Gaussian Processes, to predict the quantum level potential energy surface, thus further to obtain the atomic forces imposed in atoms (Behler and Parrinello, 2007; Bartók *et al.*, 2010; Botu and Ramprasad, 2015; Li *et al.*, 2015).

Illustrative Example(s) or Case Studies

Here, we use two examples to illustrate the applications of molecular dynamics and simulations in biophysics field.

Conformations and Energy Landscapes of *apo*-form PR LBD

Progesterone receptor (PR) is a member of the nuclear receptor (NR) family. PR is the natural receptor of the important female hormone progesterone, which affects every aspect in female development, maintenance and reproductive. The ligand binding domain (LBD) of PR, shares a similar sandwich like folding as other LBDs in NR. Though, several crystal structures of ligand bound (and co-peptides bound) PR LBD have been deposited in PDB databank, the *apo*-form (non-ligand bound state) PR LBD conformation, which is valuable for structural based virtual screening to identify druggable ligands, is still void. Combining conventional MD, umbrella sampling (Wang *et al.*, 2014), and metadynamics (Laio and Parrinello, 2002; Laio and Gervasio, 2008), the *apo*-form PR LBD has been thoroughly examined using the popular amber99SB-ildn force in explicit water environment (Zheng *et al.*, 2016). Multiple analysis methods, such as dihedral PCA, correlation analysis, reweighting, were applied. The free

energy landscape of the *apo*-form PR LBD was constructed, indicating the agonistic conformation of *apo*-form PR LBD is a meta-stable state, though not the most stable one. Several stable conformations were identified and are useful for further virtual screening study. E723, N719 and R899 are identified as key residues for conformation dynamics for helix 3 and helix 12., and metadynamics (Laio and Parrinello, 2002; Laio and Gervasio, 2008), the *apo*-form PR LBD has been thoroughly examined using the popular amber99SB-ildn force in explicit water environment (Zheng *et al.*, 2016). Multiple analysis methods, such as dihedral PCA (Mu *et al.*, 2005), correlation analysis (Göbel *et al.*, 1994), reweighting (Tiwarý and Parrinello, 2014), were applied. The free energy landscape of the *apo*-form PR LBD was constructed, indicating the agonistic conformation of *apo*-form PR LBD is a meta-stable state, though not the most stable one. Several stable conformations were identified and are useful for further virtual screening study. E723, N719 and R899 are identified as key residues for conformation dynamics for helix 3 and helix 12.

Understanding SAC6 Inter-Domain Peptides Flexibility

Another example is the dynamics study of a short peptide from yeast SAC6 inter-domain loop region (Miao *et al.*, 2016). SAC6 composed by two domains, is actin binding protein upon phosphorylation. T103 is a high potent phosphorylation site. However, why phosphorylated SAC6 has higher actin binding activity is poorly understood. We firstly hypothesized that the inter-domain peptides where T103 resides, behave differently with or without phosphorylation.

By comparing the dynamics property of two simulation systems (T103 phosphorylated and non-phosphorylated respectively), it turns out that phosphorylated T103 could restrict the peptide flexibility through electrostatic interactions with lysine and arginine residues, meanwhile phosphorylated T103 could form a hydrogen bond with G109. This hydrogen bond maintains the turn structure around G109. Free energy landscape and clustering analysis both suggest that phosphorylation at T103 stabilizes the peptides, thus facilitates the actin binding with the actin binding domains in SAC6.

Future Directions

The future of molecular simulations moves towards a more accurate, large system scale, longer simulation time scale. The polarized force field, though computational intensive, is supposed to be more precise than conventional single charge force field. The combination of quantum simulation, MMQM simulations are powerful for enzyme catalytic center accurate enough comparing to experimental results.

The fast advancement of supercomputer facilities, as well as the adaption from CPU simulations towards GPU simulations, vastly increases our confidences of future of molecular simulation. In another hand, less accurate models, such as coarse graining methods, have drawn great attention due to their ability of decreasing computational power and increasing simulation time scale.

GPU Accelerated MD Simulations

GPUs have been gradually adopted in many analytic fields, such as machine learning and simulations. Recent years, many popular simulation packages, such as Amber, Gromacs, and NAMD, have all embraced the powerful GPUs based using CUDA library. Other

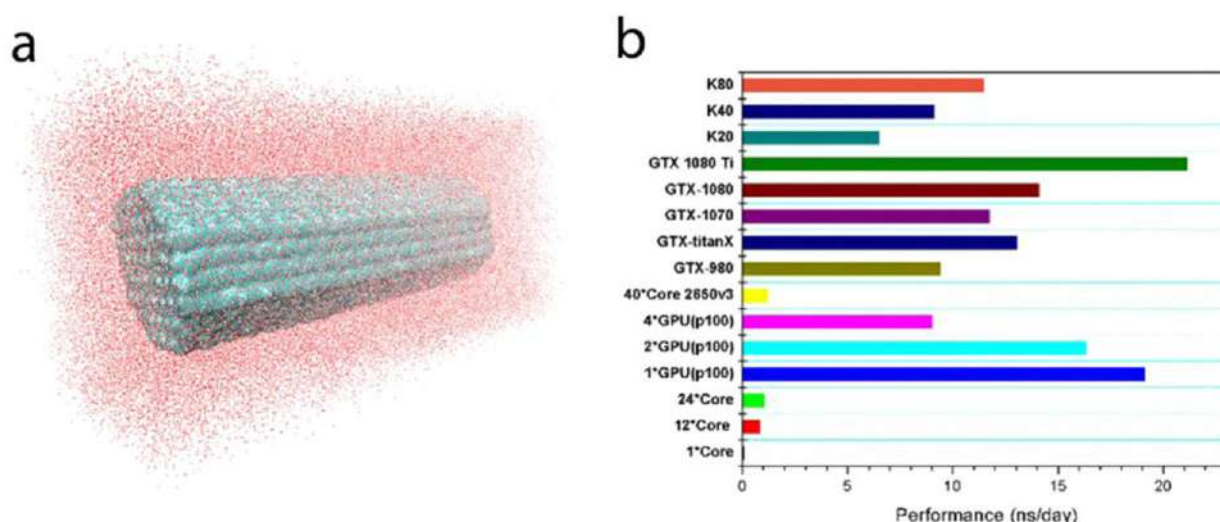


Fig. 4 GPU benchmark for cellulose system MD simulations with Amber 16. a) the simulation box of cellulose in explicit waters, with 408,609 atoms in total. b), performance of different GPUs and different CPU cores. The CPUs used in this benchmark are Intel Xeon E5-2683 V4 and E2650 V3.

new generation of simulation packages, such as openmm, htmd, and acemd all support GPU calculations. Gromacs, starting from version 4.5, implemented GPU calculations and combined them with CPUs to accelerate MD simulations (Abraham *et al.*, 2015; van der Spoel and Hess, 2011). Later in version 4.6 and higher, native GPU support has been developed to handle the most computational intensive non-bonded interactions based on CUDA library, while the PME electrostatics, bonded interactions and input-output are subjected to CPUs. Whereas, in Amber, GPUs handle nearly all calculations, such as energy evaluation, bonded and non-bonded forces.

Taking cellulose-water system (~41,000 atoms) as a benchmark using Amber 16 for *NPT* MD simulations, mid-end GPU cards (GTX 1080 and 1080Ti) show rather surprising performance comparing to high-end GPUs, such as Tesla P100, Tesla K40 and K80, as shown in Fig. 4. Given that the prices of the mid-end Nvidia GPU cards are relatively cheap (see “Relevant Websites section”), these data thus indicate that GPUs outperform CPUs, and could be adopted for efficient MD simulations with an affordable budget.

Coarse Graining

Current all-atom molecular modeling techniques enable biological systems description with details limited to simulation times and system sizes less than 1000 ns and 100 nm respectively. Coarse Graining is developed for bridging the “time-scale” and “length-scale” gaps between computational and experimental methods. The general contract of all CG methods is representing a system by a reduced number of degrees of freedom (DOFs) and fine interaction details, by mapping from atomically detailed configuration (fine grains) into a CG configuration called super atom (coarse grain) (Zhang *et al.*, 2008). That reduces the system size about *ten* folds and allows time steps up to 25–50 fs (Shih *et al.*, 2006). This way, the simulation would go faster and require less computational resources than the all-atom representation while achieving an increase of orders of magnitude in the simulated time and length scales.

Defining a coarse grained system needs three components: 1) a method to map a set of fine grains to a coarse grain, 2) a method to define the location of the center of the newly introduced coarse grain, and 3) a method to derive a suitable force field for the system. The mapping process depends on the size and complexity of the biomolecules. Small and low details molecules mapping is straightforward while the mapping process of large and complex molecules is a challenging one.

Although coarse graining is not a new concept, it is expected to take bigger rule in the near future, as the direction would be towards simulating systems on larger time scale and size scale.

The *Implicit hydrogen* model (AKA *united atom*) is an old and simple form of coarse graining, where nonpolar hydrogens are omitted and implicitly included in their connected carbon atoms with reduced number of interaction sites to around half. The resulting interaction sites for CH₂, CH₃, and CH₄ are usually parametrized Lennard-Jones potentials.

There are two general CG approaches: the residue-based CG (Shih *et al.*, 2007), and the shape-based CG (Arhipov *et al.*, 2008). Zhang *et al.* (2008) designed a method called “essential dynamics coarse-graining (ED-CG)” to define CG sites from protein primary sequence that reflect the essential dynamics characterized by PCA of an atomistic trajectory.

Multiscale coarse-graining (MC-CG)

The multiscale coarse-graining method (MS-CG, aka “force matching”) is a method of decreasing the computational load of simulating a system by “optimizing a CG potential to reproduce many-body potentials of mean force (PMF) calculated from atomistic configurations” initially for non-bonded potentials and then for bonded interactions (Lu *et al.*, 2013).

Instead of the commonly used two-body approximation, Larini *et al.* (2010) in 2010 proved that using explicit three-body potentials is superior to the two-body approximation.

LY Lu and G. Voth in 2009 implemented B-spline basis function in the MS-CG method. Which shows dramatic reduction of memory requirements and increase in computational efficiency of the MS-CG calculation. Their work showed that the MS-CG force field can approximate the many-body PMF in the coarse-grained coordinates (Lu and Voth, 2009). Then in 2011, they showed that MS-CG method can also be used to analyze the CG interactions from atomistic MD trajectories via PMF calculations and its performance is comparable to various free energy computation methods (Lu and Voth, 2011). The distribution function defined in the coarse-grained potential could be reproduced by iteratively applying the MS-CG algorithm (Lu *et al.*, 2013). It has been proven that both of the two common methods “postprocessing method” and “fixed part method” (Das and Andersen, 2009; Noid *et al.*, 2008) can similarly make MS-CG potentials which are limited in accuracy only by the incompleteness of the basis set usable (Das *et al.*, 2012).

Finally, it is worth noting that it is necessary to intervene in the variational calculation in some way, in case we need to produce accurate MS-CG potential in both high and average potential energy regions (Das *et al.*, 2012).

Multilevel coarse graining

Multilevel coarse graining is the future of coarse graining. This direction is not yet well defined. Generally, it is expected that future simulation packages can simulate systems in a heterogeneous way. Fine details can be simulated using a fine detailed model on minor steps, while coarser details can be simulated using less detailed models and over bigger time steps. One way to achieve this goal can be including the defined coarse grains into *further coarser* grains in a hierarchal way. Examples of coarse details may include diffusion and movement in the solvent, which can be simulated using models like Brownian dynamics, ignoring the detailed atom-atom interactions, but rather modeling their overall effect as a continuum.

Force fields used in CG MD simulation

Pair-potentials vs. many-body potentials

One essential requirement for any simulation package is the definition of a potential function, which describes how the simulated particles in the system would interact. Such function is known as the Force Field. Force Fields are usually empirical, and they usually employ preset bonding arrangements, which makes them capable of modeling structural and conformational changes but not chemical reactions (bond breaking and reformation) explicitly.

Force field potential function can be either *pair-potential* function, where the force is calculated as a sum of pairwise interactions between particles (e.g., Lennard-Jones potential of Van der Waals force and Born (ionic) model of the ionic lattice), or *many-body potentials*, where three or more particles interact with each other. Examples of the later are Tersoff potential (Tersoff, 1989), embedded-atom method (EAM) (Daw *et al.*, 1993), and Tight-Binding Second Moment Approximation (TBSMA) (Cleri and Rosato, 1993) potentials.

Static partial charges vs. polarizable potentials

Most force fields (especially experimental Force fields) model the effect of polarizability as static partial charges. While new techniques model polarizability as fluctuating charges, electronic structural theory, induced dipole, distributed multipoles, point charges, Bond Polarization Theory (BPT). Increased accuracy was achieved using fluctuating charges in water molecules (Lamoureux *et al.*, 2006) and with some promising results for proteins (Patel *et al.*, 2004).

Coarse-grained force fields

Coarse-Grained Force Fields can be based on generic atom types (e.g., MARTINI Force Field), or on secondary structure (e.g., VAMM).

MARTINI (Marrink *et al.*, 2004, 2007) is a very common Coarse Grained Force Field, based on having four categories of particles/beads (Q (charged), P (polar), N (nonpolar) and C (apolar)). They are then split in 4 or 5 levels, summing up to a total of 20 bead types. It has parameters for lipids, proteins, carbohydrates, DNA, RNA, and molecule types available for download on their website.

On the other hand, VAMM (Virtual Atom Molecular Mechanics) (Korkut and Hendrickson, 2009) is a “knowledge based” force field developed to model large scale conformational transitions based on the virtual interactions of C-alpha atoms and on secondary structure and residue specific contact features.

Finite Element Method

Finite element method is a common method of numerical solution of complex problems including force and heat acting on irregular structures. It is mostly used for structural mechanics applications. In this method, the system is modelled as a set of finite elements of appropriate properties (e.g., stiffness, toughness, etc.), interconnected at points called nodes.

Although the method is famous for use in civil engineering applications to solve displacement versus boundary and load conditions, it can be applied on smaller scale objects with high efficiency in terms of speed and accuracy. For instance, in 1999 O'Brien *et al.* simulated different fractured objects of different toughness property after collisions, using continuously remeshing technique and it was comparative to experimental results (O'Brien and Hodgins, 1999).

Even on the molecular levels there have been promising trails to combine FEM and MD simulation to model silicon (Izumi *et al.*, 1999) and LASER induced pressure wave propagation (Smirnova *et al.*, 1999). In 2013, Lee *et al.* published a multiscale modeling technique for bridging molecular dynamics with finite element method, based on weighted average momentum principle. Their work was applied on 2-D problems, but to the best of our knowledge, no 3D applications are available so far (Lee and Basaran, 2013).

Closing Remarks

Along with the vast increasing in supercomputing power, MD simulations, as well as other simulation variants, are gradually adopted to explore more complex biological systems, from electron level to molecule level, and evolve to achieve free energy surface construction and states-transition with affordable time and resources. In this chapter, we went through the basics of MD simulation, and then explained the advanced sampling methods. Besides, we also illustrated some applications in alignment with MD simulation in biophysical and drug design field, and examples of applying MD simulation to address biological systems have been presented. Lastly, we discussed possible future developments and directions of MD and simulations. This review thus serves as a general introduction to MD simulation and its applications.

Acknowledgement

This work was partially supported by MOE Tier 1 Grant [RG 138/15], [2015-T1-001-169]; as well as Nanyang Technological University School of Computer Science and Engineering RSS [M060020005-7070680]. We'd like to thank Dr Ning Lulu for her contribution of the GPU benchmark data we used in Fig. 4.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Docking. Biomolecular Structures: Prediction, Identification and Analyses. Cloud-Based Molecular Modeling Systems. Drug Repurposing and Multi-Target Therapies. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. In Silico Identification of Novel Inhibitors. Molecular Dynamics Simulations in Drug Discovery. Multi-Scale Modelling in Biology. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Structural Bioinformatics: An Overview. Protein Structure Visualization. Protein Three-Dimensional Structure Prediction. Rational Structure-Based Drug Design. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow. Study of The Variability of The Native Protein Structure

References

- Abrams, C., Bussi, G., 2013. Enhanced sampling in molecular dynamics using metadynamics, replica-exchange, and temperature-acceleration. *Entropy* 16 (1), 163–199.
- Abraham, M.J., Murtola, T., Schulz, R., *et al.*, 2015. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1, 19–25.
- Affentranger, R., Tavernelli, I., Di Iorio, E.E., 2006. A novel Hamiltonian replica exchange MD protocol to enhance protein conformational space sampling. *J. Chem. Theory Computation* 2 (2), 217–228.
- Arkhipov, A., Yin, Y., Schulten, K., 2008. Four-scale description of membrane sculpting by BAR domains. *Biophysical Journal* 95, 2806–2821.
- Arkin, M.R., Tang, Y., Wells, J.A., 2014. Small-molecule inhibitors of protein-protein interactions: Progressing toward the reality. *Chemistry & Biology* 21, 1102–1114.
- Baaden, M., Marrink, S.J., 2013. Coarse-grain modelling of protein–protein interactions. *Current Opinion in Structural Biology* 23, 878–886.
- Ballester, P.J., Mitchell, J.B., 2010. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* 26, 1169–1175.
- Banavali, N.K., Roux, B., 2005. Free energy landscape of A-DNA to B-DNA conversion in aqueous solution. *Journal of the American Chemical Society* 127, 6866–6876.
- Barducci, A., Bussi, G., Parrinello, M., 2008. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. *Physical Review Letters* 100, 020603.
- Barducci, A., Bonomi, M., Parrinello, M., 2011. Metadynamics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 1 (5), 826–843.
- Bartók, A.P., Payne, M.C., Kondor, R., Csányi, G., 2010. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical Review Letters* 104, 136403.
- Beauchamp, K.A., McGibbon, R., Lin, Y.S., Pande, V.S., 2012. Simple few-state models reveal hidden complexity in protein folding. *Proceedings of the National Academy of Sciences of the United States of America* 109, 17807–17813.
- Behler, J., 2016. Perspective: Machine learning potentials for atomistic simulations. *The Journal of Chemical Physics* 145, 170901.
- Behler, J., Parrinello, M., 2007. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters* 98, 146401.
- Biamés, X., Pietrucci, F., Marinelli, F., Laio, A., 2012. METAGUI. A VMD interface for analyzing metadynamics and molecular dynamics simulations. *Computer Physics Communications* 183, 203–211.
- Blanco, F.J., Rivas, G., Serrano, L., 1994. A short linear peptide that folds into a native stable β -hairpin in aqueous solution. *Nature Structural & Molecular Biology* 1, 584–590.
- Bonomi, M., Branduardi, D., Bussi, G., *et al.*, 2009. PLUMED: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications* 180, 1961–1972.
- Borhani, D.W., Shaw, D.E., 2012. The future of molecular dynamics simulations in drug discovery. *J. Comput.-Aided Mol. Des.* 26 (1), 15–26.
- Botu, V., Ramprasad, R., 2015. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry* 115, 1074–1083.
- Brooks, B.R., Brucoleri, R.E., Olafson, B.D., *et al.*, 1983. CHARMM A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry* 4, 187–217.
- Bussi, G., Gervasio, F.L., Laio, A., Parrinello, M., 2006a. Free-energy landscape for beta hairpin folding from combined parallel tempering and metadynamics. *Journal of the American Chemical Society* A 128, 13435–13441.
- Bussi, G., Gervasio, F.L., Laio, A., Parrinello, M., 2006b. Free-energy landscape for β hairpin folding from combined parallel tempering and metadynamics. *Journal of the American Chemical Society* 128, 13435–13441.
- Cai, Y.-D., Liu, X.-J., Xu, X.-B., Chou, K.-C., 2002. Prediction of protein structural classes by support vector machines. *Computers & Chemistry* 26, 293–296.
- Cecchini, M., Rao, F., Seiber, M., Cafilisch, A., 2004. Replica exchange molecular dynamics simulations of amyloid peptide aggregation. *Journal of Chemical Physics* 121, 10748–10756.
- Cheng, T., Li, Q., Zhou, Z., Wang, Y., Bryant, S.H., 2012. Structure-based virtual screening for drug discovery: A problem-centric review. *The AAPS journal* 14, 133–141.
- Chodera, J.D., Noé, F., 2014. Markov state models of biomolecular conformational dynamics. *Curr. Opin. Struct. Biol.* 25, 135–144.
- Christen, M., Hünenberger, P.H., Bakowies, D., *et al.*, 2005. The GROMOS software for biomolecular simulation GROMOS05. *Journal of Computational Chemistry* 26, 1719–1751.
- Cleri, F., Rosato, V., 1993. Tight-binding potentials for transition-metals and alloys. *Physical Review B* 48, 22–33.
- Cornell, W.D., Cieplak, P., Bayly, C.I., *et al.*, 1995. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society* 117, 5179–5197.
- Curuksu, J., Zacharias, M., 2009. Enhanced conformational sampling of nucleic acids by a new Hamiltonian replica exchange molecular dynamics approach. *J. Chem. Phys.* 130 (10), 03B610.
- Das, A., Andersen, H.C., 2009. The multiscale coarse-graining method. III. A test of pairwise additivity of the coarse-grained potential and of new basis functions for the variational calculation. *Journal of Chemical Physics* 131, 034102.
- Das, A., Lu, L., Andersen, H.C., Voth, G.A., 2012. The multiscale coarse-graining method. X. Improved algorithms for constructing coarse-grained potentials for molecular systems. *Journal of Chemical Physics* 136, 194115.
- Daw, M.S., Foiles, S.M., Baskes, M.I., 1993. The embedded-atom method – A review of theory and applications. *Materials Science Reports* 9, 251–310.
- De Vivo, M., Masetti, M., Bottegoni, G., Cavalli, A., 2016. Role of molecular dynamics and related methods in drug discovery. *J. Med. Chem.* 59 (9), 4035–4061.
- Denschlag, R., Lingenheil, M., Tavan, P., 2008. Efficiency reduction and pseudo-convergence in replica exchange sampling of peptide folding-unfolding equilibria. *Chemical Physics Letters* 458, 244–248.
- Doniger, S., Hofmann, T., Yeh, J., 2002. Predicting CNS permeability of drug molecules: Comparison of neural network and support vector machine algorithms. *Journal of Computational Biology* 9, 849–864.
- Doerr, S., De Fabritiis, G., 2014. On-the-fly learning and sampling of ligand binding by high-throughput molecular simulations. *J. Chem. Theory Comput.* 10 (5), 2064–2069.
- Doerr, S., Harvey, M.J., Noe, F., De Fabritiis, G., 2016. HTMD: high-throughput molecular dynamics for molecular discovery. *J. Chem. Theory Comput.* 12 (4), 1845–1852.
- Durrant, J.D., McCammon, J.A., 2011. Molecular dynamics simulations and drug discovery. *BMC biology* 9 (1), 71.
- Freddolino, P.L., Arkhipov, A.S., Larson, S.B., McPherson, A., Schulten, K., 2006. Molecular dynamics simulations of the complete satellite tobacco mosaic virus. *Structure* 14, 437–449.
- Frenkel, D., Smit, B., 2002. Understanding molecular simulation: From algorithms to applications. *Comput. Sci. Ser.* 1, 1–638.

- Frenkel, D., Smit, B., 2001. *Understanding Molecular Simulation: From Algorithms to Applications*, vol. 1. Elsevier.
- Fukunishi, H., Watanabe, O., Takada, S., 2002. On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *The Journal of chemical physics* 116 (20), 9058–9067.
- Galvelis, R., Sugita, Y., 2017. Neural network and nearest neighbor algorithms for enhancing sampling of molecular dynamics. *Journal of Chemical Theory and Computation* 13, 2489–2500.
- García, A.E., Sanbonmatsu, K.Y., 2002. α -Helical stabilization by side chain shielding of backbone hydrogen bonds. *Proceedings of the National Academy of Sciences* 99, 2782–2787.
- Gnanakaran, S., Nymeyer, H., Portman, J., Sanbonmatsu, K.Y., García, A.E., 2003. Peptide folding simulations. *Current Opinion in Structural Biology* 13, 168–174.
- Göbel, U., Sander, C., Schneider, R., Valencia, A., 1994. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics* 18, 309–317.
- Gumbart, J., Wang, Y., Aksimentiev, A., Tajkhorshid, E., Schulten, K., 2005. Molecular dynamics simulations of proteins in lipid bilayers. *Current Opinion in Structural Biology* 15, 423–431.
- Halgren, T.A., 1996. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of Computational Chemistry* 17, 490–519.
- Harrigan, M.P., Sultan, M.M., Hernández, C.X., *et al.*, 2017. MSMBuilder: Statistical models for biomolecular dynamics. *Biophysical Journal* 112, 10–15.
- Heermann, D.W., 1990. Computer-simulation methods. In: *Computer Simulation Methods in Theoretical Physics*. Springer.
- Heffernan, R., Paliwal, K., Lyons, J., *et al.*, 2015. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific Reports* 5, 11476.
- Hou, T., Wang, J., Zhang, W., Wang, W., Xu, X., 2006. Recent advances in computational prediction of drug absorption and permeability in drug discovery. *Current Medicinal Chemistry* 13, 2653–2667.
- Huang, K., 1987. *Statistical Mechanics*. 18. Wiley. 3.
- Hua, S., Sun, Z., 2001. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *Journal of Molecular Biology* 308, 397–407.
- Hu, L., Soderhjelm, P., Ryde, U., 2011. On the convergence of QM/MM energies. *Journal of Chemical Theory and Computation* 7, 761–777.
- Husic, B.E., Pande, V.S., 2018. Markov state models: from an art to a science. *J. Am. Chem. Soc.* 140 (7), 2386–2396.
- Im, W., Roux, B.T., 2002. Ions and counterions in a biological channel: A molecular dynamics simulation of OmpF porin from *Escherichia coli* in an explicit membrane with 1M KCl aqueous salt solution. *Journal of Molecular Biology* 319, 1177–1197.
- Izumi, S., Kawakami, T., Sakai, S., 1999. A FEM-MD combination method for silicon. In: *Proceeding of the 1999 International Conference on Simulation of Semiconductor Processes and Devices, SISPAD '99*, pp. 143–146.
- Jorgensen, W.L., Maxwell, D.S., Tirado-rives, J., 1996. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society* 118, 11225–11236.
- Jorgensen, W.L., Tirado-Rives, J., 1988. The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society* 110, 1657–1666.
- Judson, R., Elloumi, F., Setzer, R.W., Li, Z., Shah, I., 2008. A comparison of machine learning algorithms for chemical toxicity classification using a simulated multi-scale data model. *BMC Bioinformatics* 9, 241.
- Kalos, M.H., Whitlock, P.A., 2008. *Monte Carlo Methods*. John Wiley & Sons.
- Kästner, J., 2011. Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1, 932–942.
- Khalili-Araghi, F., Gumbart, J., Wen, P.-C., *et al.*, 2009. Molecular dynamics simulations of membrane channels and transporters. *Current Opinion in Structural Biology* 19, 128–137.
- Kone, A., Kofke, D.A., 2005. Selection of temperature intervals for parallel-tempering simulations. *The Journal of Chemical Physics* 122, 206101.
- Korkut, A., Hendrickson, W.A., 2009. A force field for virtual atom molecular mechanics of proteins. *Proceedings of the National Academy of Sciences of the United States of America* 106, 15667–15672.
- Kubelka, J., Hofrichter, J., Eaton, W.A., 2004. The protein folding ‘speed limit’. *Current Opinion in Structural Biology* 14, 76–88.
- Laio, A., Gervasio, F.L., 2008. Metadynamics: A method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. *Reports on Progress in Physics* 71, 126601.
- Laio, A., Parrinello, M., 2002. Escaping free-energy minima. *Proceedings of the National Academy of Sciences* 99, 12562–12566.
- Lamoureux, G., Harder, E., Vorobyov, I.V., Roux, B., Mackerell, A.D., 2006. A polarizable model of water for molecular dynamics simulations of biomolecules. *Chemical Physics Letters* 418, 245–249.
- Larini, L., Lu, L., Voth, G.A., 2010. The multiscale coarse-graining method. VI. Implementation of three-body coarse-grained potentials. *The Journal of Chemical Physics* 132, 164107.
- Lee, H.S., Zhang, Y., 2012. (333) BSP-SLIM: A blind low-resolution ligand-protein docking approach using theoretically predicted protein structures. *Proteins* 80, 93–110.
- Lee, Y.C., Basaran, C., 2013. A multiscale modeling technique for bridging molecular dynamics with finite element method. *Journal of Computational Physics* 253, 64–85.
- Lindorff-Larsen, K., Piana, S., Dror, R.O., Shaw, D.E., 2011. How fast-folding proteins fold. *Science* 334, 517–520.
- Lin, E., Shell, M.S., 2009. Convergence and heterogeneity in peptide folding with replica exchange molecular dynamics. *Journal of Chemical Theory and Computation* 5, 2062–2073.
- Lise, S., Archambeau, C., Pontil, M., Jones, D.T., 2009. Prediction of hot spot residues at protein-protein interfaces by combining machine learning and energy-based methods. *BMC Bioinformatics* 10, 365.
- Liu, P., Kim, B., Friesner, R.A., Berne, B., 2005. Replica exchange with solute tempering: A method for sampling biological systems in explicit water. *Proceedings of the National Academy of Sciences of the United States of America* 102, 13749–13754.
- Li, Z., Kermode, J.R., De Vita, A., 2015. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Physical Review Letters* 114, 096405.
- Lu, L., Dama, J.F., Voth, G.A., 2013. Fitting coarse-grained distribution functions through an iterative force-matching method. *Journal of Chemical Physics* 139, 121906.
- Lu, L., Voth, G.A., 2011. The multiscale coarse-graining method. VII. Free energy decomposition of coarse-grained effective potentials. *The Journal of Chemical Physics* 134, 224107.
- Lu, L.Y., Voth, G.A., 2009. Systematic coarse-graining of a multicomponent lipid bilayer. *Journal of Physical Chemistry B* 113, 1501–1510.
- Ma, B., Nussinov, R., 2002. Stabilities and conformations of Alzheimer's β -amyloid peptide oligomers (A β 16–22, A β 16–35, and A β 10–35): Sequence effects. *Proceedings of the National Academy of Sciences* 99, 14126–14131.
- Mayo, S.L., Olafson, B.D., Goddard, W.A., 1990. DREIDING: A generic force field for molecular simulations. *J. Phys. Chem.* 94 (26), 8897–8909.
- Marinari, E., Parisi, G., 1992. Simulated tempering: A new Monte Carlo scheme. *EPL (Europhysics Letters)* 19, 451.
- Marinelli, F., Pietrucci, F., Laio, A., Piana, S., 2009. A kinetic model of trp-cage folding from multiple biased molecular dynamics simulations. *PLOS Computational Biology* 5, e1000452.
- Marrink, S.J., De Vries, A.H., Mark, A.E., 2004. Coarse grained model for semiquantitative lipid simulations. *Journal of Physical Chemistry B* 108, 750–760.
- Marrink, S.J., Risselada, H.J., Yefimov, S., Tieleman, D.P., De Vries, A.H., 2007. The MARTINI force field: Coarse grained model for biomolecular simulations. *Journal of Physical Chemistry B* 111, 7812–7824.
- McCammon, J.A., Gelin, B.R., Karplus, M., 1977. Dynamics of folded proteins. *Nature* 267, 585–590.
- McGibbon, R., Ramsundar, B., Sultan, M., Kiss, G., Pande, V., 2014. Understanding protein dynamics with L1-regularized reversible hidden Markov models. In: *Proceeding of the International Conference on Machine Learning*, pp. 1197–1205.
- McGibbon, R.T., Pande, V.S., 2013. Learning kinetic distance metrics for Markov state models of protein conformational dynamics. *J. Chem. Theory Comput.* 9 (7), 2900–2906.
- Miao, Y., Johnson, J.E., Ortoleva, P.J., 2010. All-atom multiscale simulation of cowpea chlorotic mottle virus capsid swelling. *The Journal of Physical Chemistry B* 114, 11181–11195.
- Miao, Y.S., Han, X.M., Zheng, L.Z., *et al.*, 2016. Fimbrin phosphorylation by metaphase Cdk1 regulates actin cable dynamics in budding yeast. *Nature Communications* 7.

- Mitsutake, A., Sugita, Y., Okamoto, Y., 2001. Generalized-ensemble algorithms for molecular simulations of biopolymers. *Peptide Science* 60, 96–123.
- Mu, Y., 2009. Dissociation aided and side chain sampling enhanced Hamiltonian replica exchange. *Journal of Chemical Physics* 130, 164107.
- Mu, Y., Nguyen, P.H., Stock, G., 2005. Energy landscape of a small peptide revealed by dihedral angle principal component analysis. *Proteins: Structure, Function, and Bioinformatics* 58, 45–52.
- Mu, Y., Yang, Y., Xu, W., 2007. Hybrid Hamiltonian replica exchange molecular dynamics simulation method employing the Poisson-Boltzmann model. *The Journal of Chemical Physics* 127, 084119.
- Naritomi, Y., Fuchigami, S., 2011. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *The Journal of Chemical Physics* 134, 02B617.
- Nafisa, M., Hassan, 2017. (111) Protein-Ligand Blind Docking Using QuickVina-W With Inter-Process Spatio-Temporal Integration. *Scientific Reports* 7 (1), doi:10.1038/s41598-017-15571-7.
- Noé, F., Schütte, C., Vanden-Eijnden, E., Reich, L., Weikl, T.R., 2009. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences* 106, 19011–19016.
- Noé, F., Clementi, C., 2015. Kinetic distance and kinetic maps from molecular dynamics simulation. *J. Chem. Theory Comput.* 11 (10), 5002–5011.
- Noid, W.G., Liu, P., Wang, Y., *et al.*, 2008. The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models. *Journal of Chemical Physics* 128, 244115.
- O'Brien, J.F., Hodgins, J.K., 1999. Graphical modeling and animation of brittle fracture. In: *Siggraph 99 Conference Proceedings*, pp. 137–146.
- Ogata, K., Yuki, T., Hatakeyama, M., Uchida, W., Nakamura, S., 2013. All-atom molecular dynamics simulation of photosystem II embedded in thylakoid membrane. *Journal of the American Chemical Society* 135, 15670–15673.
- Okur, A., Roe, D.R., Cui, G., Hornak, V., Simmerling, C., 2007. Improving convergence of replica-exchange simulations through coupling to a high-temperature structure reservoir. *Journal of Chemical Theory and Computation* 3, 557–568.
- Patel, S., Mackerell Jr., A.D., Brooks 3rd, C.L., 2004. CHARMM fluctuating charge force field for proteins: II protein/solvent properties from molecular dynamics simulations using a nonadditive electrostatic model. *Journal of Computational Chemistry* 25, 1504–1514.
- La Penna, G., Morante, S., Perico, A., Rossi, G.C., 2004. Designing generalized statistical ensembles for numerical simulations of biopolymers. *Journal of Chemical Physics* 121, 10725–10741.
- Pande, V.S., Beauchamp, K., Bowman, G.R., 2010. Everything you wanted to know about Markov State Models but were afraid to ask. *Methods* 52 (1), 99–105.
- Pérez-Hernández, G., Paul, F., Giorgino, T., De Fabritiis, G., Noé, F., 2013. Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics* 139, 07B604_1.
- Piana, S., Laio, A., 2007. A bias-exchange approach to protein folding. *Journal of Physical Chemistry B* 111 (17), 4553–4559.
- Piana, S., Lindorff-Larsen, K., Shaw, D.E., 2011. How robust are protein folding simulations with respect to force field parameterization? *Biophysical Journal* 100, L47–L49.
- Rahman, A., 1964. Correlations in the motion of atoms in liquid argon. *Physical Review* 136, A405.
- Rajamani, D., Thiel, S., Vajda, S., Camacho, C.J., 2004. Anchor residues in protein-protein interactions. *Proceedings of the National Academy of Sciences of the United States of America* 101, 11287–11292.
- Riccardi, D., Yang, S., Cui, Q., 2010. Proton transfer function of carbonic anhydrase: Insights from QM/MM simulations. *Biochimica et Biophysica Acta* 1804, 342–351.
- Röblitz, S., Weber, M., 2013. Fuzzy spectral clustering by PCCA+: Application to Markov state models and data classification. *Adv. Data Anal. Classif.* 7 (2), 147–179.
- Rosenberg, J.M., 1992. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *Journal of Computational Chemistry* 13, 1011–1021.
- Scherer, M.K., Trendelkamp-Schroer, B., Paul, F., *et al.*, 2015. PyEMMA 2: A software package for estimation, validation, and analysis of Markov models. *Journal of Chemical Theory and Computation* 11, 5525–5542.
- Schwantes, C.R., Pande, V.S., 2015. Modeling molecular kinetics with tICA and the kernel trick. *J. Chem. Theory Comput.* 11 (2), 600–608.
- Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., Wolfson, H.J., 2005. (222) PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucl. Acids. Res.* 33, W363–367.
- Senn, H.M., Thiel, W., 2007. QM/MM studies of enzymes. *Current Opinion in Chemical Biology* 11, 182–187.
- Setubal, Joao Carlos, Meidanis, Joao, Setubal-Meidanis, 1997. Introduction to computational molecular biology 1997 (No. 04), QH506, S4. PWS Pub.
- Shaw, D.E., Maragakis, P., Lindorff-Larsen, K., *et al.*, 2010. Atomic-level characterization of the structural dynamics of proteins. *Science* 330, 341–346.
- Shih, A.Y., Arkhipov, A., Freddolino, P.L., Schulten, K., 2006. Coarse grained protein – lipid model with application to lipoprotein particles. *The Journal of Physical Chemistry B* 110, 3674–3684.
- Shih, A.Y., Freddolino, P.L., Arkhipov, A., Schulten, K., 2007. Assembly of lipoprotein particles revealed by coarse-grained molecular dynamics simulations. *Journal of Structural Biology* 157, 579–592.
- Shrivastava, I.H., Sansom, M.S., 2000. Simulations of ion permeation through a potassium channel: Molecular dynamics of KcsA in a phospholipid bilayer. *Biophysical Journal* 78, 557–570.
- Sindhikara, D., Meng, Y.L., Roitberg, A.E., 2008. Exchange frequency in replica exchange molecular dynamics. *Journal of Chemical Physics* 128.
- Singhal, N., Pande, V.S., 2005. Error analysis and efficient sampling in Markovian state models for molecular dynamics. *J. Chem. Phys.* 123 (20), 204909.
- Smirnova, J.A., Zhigilei, L.V., Garrison, B.J., 1999. A combined molecular dynamics and finite element method technique applied to laser induced pressure wave propagation. *Computer Physics Communications* 118, 11–16.
- Snow, C.D., Zagrovic, B., Pande, V.S., 2002. The Trp cage: Folding kinetics and unfolded state topology via molecular dynamics simulations. *Journal of the American Chemical Society* 124, 14548–14549.
- Suárez, E., Adelman, J.L., Zuckerman, D.M., 2016. Accurate estimation of protein folding and unfolding times: Beyond Markov state models. *Journal of Chemical Theory and Computation* 12, 3473–3481.
- Sugita, Y., Okamoto, Y., 1999a. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters* 314, 141–151.
- Sugita, Y., Okamoto, Y., 1999b. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters* 314, 141–151.
- Sultan, M.M., Wayment-Steele, H.K., Pande, V.S., 2018. Transferable neural networks for enhanced sampling of protein dynamics. *arXiv preprint arXiv:1801.00636*.
- Sutto, L., Marsili, S., Gervasio, F.L., 2012. New advances in metadynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 2, 771–779.
- Swendsen, R.H., Wang, J.-S., 1987. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters* 58, 86.
- Tai, K., 2004. Conformational sampling for the impatient. *Biophysical Chemistry* 107, 213–220.
- Tersoff, J., 1989. Modeling solid-state chemistry – Interatomic potentials for multicomponent systems. *Physical Review B* 39, 5566–5568.
- Tiwary, P., Parrinello, M., 2014. A time-independent free energy estimator for metadynamics. *The Journal of Physical Chemistry B* 119, 736–742.
- Tolman, R.C., 1938. The principles of statistical mechanics. Courier Corporation.
- Torrie, G.M., Valleau, J.P., 1974. Monte Carlo free energy estimates using non-Boltzmann sampling: Application to the sub-critical Lennard-Jones fluid. *Chemical Physics Letters* 28, 578–581.
- Torrie, G.M., Valleau, J.P., 1977. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics* 23, 187–199.
- van der Kamp, M.W., Mulholland, A.J., 2013. Combined quantum mechanics/molecular mechanics (QM/MM) methods in computational enzymology. *Biochemistry* 52, 2708–2728.
- van Gunsteren, W.F., Billeter, S.R., Eising, A.A., *et al.*, 1996. Biomolecular simulation: The (GROMOS96) manual and userguide. Hochschuleverlag AG an der ETH Zürich.
- van der Spoel, D., Hess, B., 2011. GROMACS—the road ahead. *Wiley Interdiscip. Rev.: Comput. Molecular Science* 1 (5), 710–715.
- Walters, W.P., Murcko, M.A., 2002. Prediction of 'drug-likeness'. *Advanced Drug Delivery Reviews* 54, 255–271.

- Wang, J., Shao, Q., Xu, Z., *et al.*, 2014. Exploring transition pathway and free-energy profile of large-scale protein conformational change by combining normal mode analysis and umbrella sampling molecular dynamics. *Journal of Physical Chemistry B* 118, 134–143.
- Wang, S., Peng, J., Ma, J., Xu, J., 2016. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports* 6.
- Weber, M., 2011. A subspace approach to molecular Markov state models via a new infinitesimal generator.
- Woutersen, S., Pfister, R., Hamm, P., *et al.*, 2002. Peptide conformational heterogeneity revealed from nonlinear vibrational spectroscopy and molecular-dynamics simulations. *The Journal of Chemical Physics* 117, 6833–6840.
- Zhang, T., Nguyen, P.H., Nasica-Labouze, J., Mu, Y., Derreumaux, P., 2015. Folding atomistic proteins in explicit solvent using simulated tempering. *The Journal of Physical Chemistry B*.
- Zhang, W., Wu, C., Duan, Y., 2005a. Convergence of replica exchange molecular dynamics. *Journal of Chemical Physics* 123, 154105.
- Zhang, W., Wu, C., Duan, Y., 2005b. Convergence of replica exchange molecular dynamics. *Journal of Chemical Physics* 123.
- Zhang, Y., Liu, H., Yang, W., 2000. Free energy calculation on enzyme reactions with an efficient iterative procedure to determine minimum energy paths on a combined ab initio QM/MM potential energy surface. *The Journal of Chemical Physics* 112, 3483–3492.
- Zhang, Z.Y., Lu, L.Y., Noid, W.G., *et al.*, 2008. A systematic methodology for defining coarse-grained sites in large biomolecules. *Biophysical Journal* 95, 5073–5083.
- Zheng, L., Lin, V.C., Mu, Y., 2016. Exploring flexibility of progesterone receptor ligand binding domain using molecular dynamics. *PLOS ONE* 11, e0165824.

Further Reading

- Anderson, J.A., Lorenz, C.D., Travesset, A., 2008. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics* 227 (10), 5342–5359.
- Bonomi, M., Parrinello, M., 2010. Enhanced sampling in the well-tempered ensemble. *Physical Review Letters* 104 (19), 190601.
- Borodin, O., 2009. Polarizable force field development and molecular dynamics simulations of ionic liquids. *The Journal of Physical Chemistry B* 113 (33), 11463–11478.
- Braun, G.H., *et al.*, 2008. Molecular dynamics, flexible docking, virtual screening, ADMET predictions, and molecular interaction field studies to design novel potential MAO-B inhibitors. *Journal of Biomolecular Structure and Dynamics* 25 (4), 347–355.
- Chmiela, S., *et al.*, 2018. Towards exact molecular dynamics simulations with machine-learned force fields. *Bulletin of the American Physical Society*.

Relevant Websites

- <https://www.geforce.com/hardware/compare-buy-gpus>
GeForce
GeForce.com.
- <http://www.emma-project.org>
MD with MSM (or HMM) Pyemma.
- <http://msmbuilder.org>
MSMBuilder.
- <https://plumed.github.io/doc.html>
Plumed tutorials and manual.
- <http://docs.markovmodel.org/>
Principle of MSM.

Biographical Sketch



Zheng Liangzhen is a PhD candidate in School of Biological Sciences, Nanyang Technological University, Singapore since 2014. He obtained his bachelor's degree in biological sciences from College of Life Science, Wuhan University, in 2010. His current interests are protein dynamics, protein-nucleic acid complex dynamics, and machine learning aided drug discovery.



Amr Alhossary received his M.B.B.S. from School of Medicine, Cairo University in 2003, then his Graduate diploma and M.Sc. in Bioinformatics from Faculty of Computers and Information, Helwan University in 2008 and 2012 respectively. He is currently a PhD candidate in School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include Structural Computational Biology, Molecular Docking, and Drug Design.



Kwoh Chee Keong is currently in the School of Computer Science and Engineering since 1993. He is currently the Assistant Chair (Graduate Studies). He received his Bachelor degree in Electrical Engineering (1st Class) and Master in Industrial System Engineering from the National University of Singapore in 1987 and 1991 respectively. He received his Ph.D. degrees from the Imperial College, University of London in 1995. He is often invited for conferences and journals, including GIW, IEEE BIBM, RECOMB, PRIB etc. He is a member of The Institution of Engineers Singapore, Association for Medical and Bio-Informatics, Imperial College Alumni Association of Singapore (ICAAS). He was conferred the Public Service Medal, the President of Singapore in 2008. His research interests include Data Mining and Soft Computing and Graph-Based inference; applications areas include Bioinformatics and Biomedical Engineering.



Mu Yuguang received his B.Sc. in Physics, M.S. in Quantum Chemistry, and Ph.D. Physics from Shandong University, China in 1991, 1994, and 1997 respectively. He worked as a Lecturer in Department of photoelectronics, Shandong University before moving to Germany in 2009 to work as a research fellow in Fraunhofer Institute of Integrated Systems and Device Technology (IISB) then Institute of Theoretical Chemistry Frankfurt. He has been in School of Biological Sciences, NTU, since 2003, as a Lee Kuan Yew Fellowship Scholar then Assistant and associate professor until now. His research interests include MD simulation method and data-analysis method development; protein folding and unfolding; RNA dynamics; DNA dynamics; DNA-protein; and DNA-counterions interaction study.

Nucleic-Acid Structure Database

Airy Sanjeev and Venkata SK Mattaparthi, Tezpur University, Tezpur, India

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Barco, Guimaraes, Portugal and University of Minho, Braga, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

Nucleic acids have a significant role in the development of living systems. They contain wholesome interesting information related to the molecular biologists. They are extensively studied both for biological applications and for various databases. The nucleic acids that consist of RNAs are considered as the main target for several specialized databases. The sequence of nucleic acids consists of numerous control regions such as promoters, splice junctions, origins of replication, etc. The 3D structure of nucleic acid can be well understood by investigating the different types of nucleic acid functions such as nucleic acid–protein interactions, functional RNAs, etc. Several databases have been developed till date based on the nucleic acid sequence apart from some that have been developed from the structures storing information related to the secondary and tertiary structures.

Background

The molecules of nucleic acids allow to transfer the genetic information from one generation to the next generation. The nucleic acids are abundantly found in the nucleus of the cell. These nucleic acids are long polymers that are built up of several units of nucleotide monomers such as C (cytosine), A (adenine), G (guanine), T (thymine), and U (uracil). There are two types of nucleic acids, that is, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). The structure of DNA is coiled together by both the strands of nucleotide polymer chains (C, A, G, T) to form a double-helical pattern. The double-helical pattern of DNA is further stabilized by the formation of hydrogen bonds between the bases of nucleotides characterized on the chemical affinity. These chemical affinities defined by the nucleotides signify adenine with thymine and cytosine with guanine. On the other hand, RNA consists of a single strand of polymer with nucleotides C, A, G, and U. The structure of RNA has the ability to fold itself and obtain a secondary structure that is important for carrying out the various biological processes. Among both the nucleic acids, the DNA molecule stores the genetic information of the cell and carries that information from parent to offspring. In case of some viruses, the RNA stores the genetic information that is needed for the cell. However the information that is related to the life cycle of the cell and also its activities are stored in the DNA. The molecule of RNA is being built up when the complex biochemical decodification machinery present in the cell reacts on the DNA molecule to acquire information required for a specific function. The RNA molecule is the primary constituent for the process of protein synthesis. It is also accountable for transmitting the genetic information that is encoded within the DNA to construct a specific protein that is required for a specific function in a specific procedure. There exist three main types of RNA, which are *mRNA*, *tRNA*, and *rRNA*. *mRNA* is messenger RNA, which initiates the information required for building the protein molecule from the nucleus to the cytoplasm. *tRNA*, also known as transfer RNA, on the other hand transports the amino acids to the ribosomes for making the proteins. The *rRNA*, or ribosomal RNA, is connected with a variety of proteins that form the ribosomes. The complex structures thus formed catalyze the association of amino acids to form protein chains by moving along with the *mRNA* molecule. Also they are involved in the process of protein synthesis by binding to the *tRNAs* and other accessory molecules. Apart from these, there also exists regulatory RNA, which has the ability to regulate the gene expression from the various mechanisms such as blocking or interference (Alberts *et al.*, 2002; Bailey, 2017; Chen *et al.*, 1995; Darnell, 1985; Dickerson *et al.*, 1982; Felsenfeld, 1985; Katsuyuki *et al.*, 2016; Krieger *et al.*, 2004; Mirkin, 2001; Travers and Muskhelishvili, 2015; Watson and Crick, 1953; Weinberg, 1985).

The proteins interact with the nucleic acids, mainly DNA and RNA, by means of similar physical forces that consist of dipolar interactions (hydrogen bonding, H-bonds), electrostatic interactions (salt-bridges), effect of entropy (hydrophobic interactions), and also forces of dispersion (base stacking). These physical forces, which are acted upon by the interactions, help in varying degrees of binding of the proteins in a sequence-specific (*tight*) or non-sequence-specific (*loose*) orientation. Apart from that, the affinity and specificity of a particular protein–nucleic acid interaction is increased via the protein oligomerization or multiprotein complex formation. The secondary and tertiary structures that are formed from the nucleic acid sequences give an additional mechanism from which the proteins recognize themselves and then bind to the specific nucleic acid sequences. The DNA is a linear macromolecule that is found in most of the cells of living beings. Unlike proteins, it is made up of four diverse building blocks known as *nucleotides*. These nucleotides are made up of a base (purine or pyrimidine group) and a phosphate group (2'-deoxyribosyl-tri-phosphate). The bases present in the nucleotides are of four types. They are adenine and guanine present in *purine* and thymine and cytosine present in *pyrimidine*. The sugar present in the DNA is the 2'-deoxy ribose that is phosphorylated at position 5' of the hydroxyl group. The free nucleotide radical consists of any of one, two, or three phosphates that represent mono-, di-, or triphosphate, which are better known as *dATP*, *dGTP*, *dTTP*, and *dCTP*. These nucleotides are however linked covalently to the DNA via 5'-phosphate to 3'-hydroxyl of the subsequent nucleotide. The single strands of DNA are not stable, although they remain connected with another strand in order to form a double-helical structure such that both strands interlink around each other. The four bases situated at the center of the helix are H-bonded to one another in a very precise manner. The specific way that

the bases point towards the center of the helix and H-bond is as follows: G with C (3 H-bonds) and A with T (2 H-bonds). The linear geometry and rigidity associated with the H-bonds restricts the formation of *base pairs* (bp). The helical axis and the plane of the base pair are perpendicular to one another.

The B-DNA is the right-handed physiological form of double-helical DNA structure which was described by Watson and Crick in the year 1953. It is an antiparallel double-helical structure with a diameter of 2 nm. The B-DNA consists of 10 bp per turn, 36° per bp twist, 3.4 Å per base, 34 Å pitch, 2 nm diameter. These characteristics describe B-DNA as an idealized helical structure. The actual DNA conformation deviates from the B-form DNA conformation in both a sequence dependent manner and DNA-binding interaction protein.

The A-form of the DNA is a wider and flatter helical conformer with the base pair plane tilted to the helical axis. The helix of A-DNA possess the major and minor groove that consists of the respective parameters such as 11 bp per turn, 33° per turn twist, 0.26 nm per base, 2.8 nm pitch, 2.6 nm diameter, the base plane tilting to the helical axis with 20°.

Another form of DNA is the Z-DNA, which is a left-handed conformation consisting of the subsequent structural parameters, which are 12 bp per turn, 0.37 nm per base, 1.8 nm diameter, a tilt base of 7°, 4.5 nm pitch.

RNA is another nucleic acid, which is similar to DNA and is built up from four different building blocks that are the *ribonucleotides*. In case of RNA, thymine is modified such that it lacks a methyl group and results in the formation of uracil in the base-pairing mechanism. The ribose sugar present in the RNA comes along with a hydroxylated form. The presence of uracil instead of thymine and 2'-OH in the ribose sugar are the main chemical differences between the two nucleic acids, DNA and RNA. RNA is basically a single stranded helical structure. But both RNA and DNA form bp that result in the formation of a heteromeric double-helical secondary structure consisting of a single DNA and RNA strand. This process of annealing of RNA strand to the complementary DNA strand is known as the *hybridization process*. And this process however plays a significant role in the transcription and translation mechanism of the genetic sequences into protein sequences. As the RNA forms a short double strand structure on itself, hence it is known as the stem and loop structure. The double-helical structures DNA/RNA and RNA/RNA form an A-DNA-like conformation, better known as A-RNA or RNA-II. This helix consists of the following parameters: 11 bp per turn, 3.0 nm pitch (30 Å). In the RNA conformation, three major types exist, which are *messenger RNA (mRNA)*, *transfer RNA (tRNA)*, and *ribosomal RNA (rRNA)* (Alberts *et al.*, 2002; Bailey, 2017; Bansal, 2003; Chen *et al.*, 1995; Churchill, 1996; Darnell, 1985; Dickerson *et al.*, 1982; Felsenfeld, 1985; Katsuyuki *et al.*, 2016; Krieger *et al.*, 2004; Mirkin, 2001; Travers and Muskhelishvili, 2015; Watson and Crick, 1953; Weinberg, 1985).

The binding function of DNA or RNA is restricted in the distinct conserved domains that are within the tertiary structure. The affinity and specificity can be increased by binding the single protein with multiple nucleic acid binding domains for a specific target nucleic acid sequence. This can however lead to the development of a hierarchy of interactions as the proteins bind directly or indirectly to the nucleic acids via the other bound proteins. However we can gain a significant binding affinity for the protein and nucleic acids complex by understanding the strength of interactions influencing the approaches for the complex assembly. Among these interactions, some are recognized as transient, which need stabilization through chemical cross-linking prior to isolation of the complexes. Hence by studying the protein–nucleic acid interactions, the identification of protein in the complexes can be known and also the structure that is required to assemble these complexes is vital to understanding the role these complexes play in regulating cellular processes (Govil *et al.*, 1985; Luger and Phillips, 2010; Park *et al.*, 2014; Sathyapriya *et al.*, 2008; Siggers and Gordan, 2014; Valuchova *et al.*, 2016).

It has been observed that small, cationic, and planar molecules interpolate between the various bp whereas the large nonpolar molecules interact by the mechanism of groove binding. Various molecules for in vitro and in vivo activities have been tested and also some molecules from them have entered clinical trials. Hence various parameters that contain the experimental dataset required for drug-nucleic acid interactions have been established. These datasets contain the various types of ligands and binding parameters for their interaction with the associated nucleic acid. Although it becomes a challenging task to gather the whole information specifically for the ligand and its targets. However, by understanding the various interactions improvement in the projects of drug discovery can be achieved. Hence it is essential to combine those experimental data and information at a particular platform.

Nucleic Acid Database

NDB History, Sources and Number of NDB Data

The Nucleic Acid Database (NDB) was the first database for nucleic acid structures (Coimbatore Narayanan *et al.*, 2014), which was designed in the early 1990s and is considered as the main reference in the field of nucleic acids. The newly designed database (see “Relevant Websites section”), is a web portal that can provide access to various information related to 3D nucleic acid structures and their complexes. Apart from the contents of primary data, NDB also contains some derived geometric data, classification of structure and motifs, standards for describing nucleic acid features, and also tools and software for analysis of nucleic acids. In this review, we describe the recently redesigned NDB website with some special features related to new RNA-derived data and annotations and their implementation and integration into the search capabilities. This NDB database was founded in 1991 to assemble and distribute structural information about nucleic acids (Berman *et al.*, 1992). It contains annotations that are specific to the structure of nucleic acid and function and tools that allow users to search, download, analyze, and to learn more about nucleic acids apart from the contents available in PDB (Berman *et al.*, 2003). Thus, this database is a value-added one that provides services that are specific to the nucleic acid community. The main importance of NDB on its discovery was to focus on the DNA structural biology. However various tools and annotations have been developed that can address the various characteristics of these molecules as more RNA structures have been determined. It serves as a central medium for the nucleic acid

structural information and annotations. NDB at present consists of more than 8000 experimental structures for DNA and RNA molecules that are acquired and curated from the Protein Data Bank (PDB). The structures available in the NDB are annotated with various information that are unique to the nucleic acids but are almost not available from the original entry of PDB. The structures are further analyzed by means of computation of structural geometries and thereby classification was done by searching the sequence, structure levels, and secondary structure by incorporating the PDB entries to the NDB. By using the various tools that are available at the NDB server linked to the other resources, the specific analyses and online visualization in 2D and 3D can be carried out by the users. Apart from that, various new tools were developed in 2014 that are required to analyze the RNA sequences, DNA molecules, aligning RNA and DNA molecules, calculating and visualizing RNA structural geometries and base-pairing patterns that are comparatively more complex and diverse than DNA. This server also includes various statistics related to ideal geometries that are well suited for sugars, bases, and also for various hydrogen bonding pairing patterns that consist of a new catalog of base-pairing in RNA, an educational section, and software for further offline analyses.

NDB Entry Format

The main content of NDB is the primary structure information that is related to the nucleic acids that are acquired from the PDB and also from classification and derived data. The primary data present in the NDB that is obtained from PDB consists of experimental files, identifiers, structural descriptions, citations, crystal data, coordinate information, and various details related to the crystallization, data collection, and structure refinement. NDB also contains ERFs (external reference files), which mainly consist of nucleic acid classifications, calculated derived data, and additional derived data on RNA structural features. The web interface provides search, reporting, and also download functionalities. The results of these searches are returned as either group of structures or individual structures based on the query. These search results are related to a number of reporting features that include predefined feature reports, navigation to individual structure summary reports that further permit primary data files, derived data, and molecular images download. There are about three search options that are present from the secondary persistent header, that is, *ID search*, *search*, and *advanced search*. These search options can however be accessed in the *Search Structures*, a section available on the homepage.

The asymmetric unit's atomic coordinates and the biological assemblies are accessible from the *Downloads* section obtained from the corresponding NDB structure summary report, which is available both in *PDB* as well as *mmCIF* formats. This *mmCIF* format contains the structure factor data whereas the NMR restraint information is available in deposited program format. During the database update, the files are updated weekly and are downloadable from the NDB ftp server at website provided in "Relevant Websites section".

From the advanced search query, the featured reports for any result set are available. The predefined set of reports is such that it contains *NDB status*, *cell dimensions*, *citation*, *refinement data*, *backbone torsion*, *base pair and base step parameters*, *descriptor*, *sequence*, and *RNA motifs*. The report containing NDB status consists of parameters such as *NDB and PDB IDs*, *structure title*, *authors*, *deposition*, and *release dates* is the default report for any advanced search query.

The structures available in the NDB database possess their own personal summary page that is mostly appropriate to that particular structure. There are about four sections of data in the structure summary page, including (a) *primary structure information*, (b) *downloads*, (c) *derived structural data*, and (d) *images*. The primary structure information, which is the main division, encompasses the entry title, sequence, citation, experimental details, refinement information, and various structural descriptions. In the *Download Data* section, the details of the atomic coordinates, structure factors, and NMR restraint information are present. An RNA View Image is provided for the entries of RNA that represents 2D base pairing. Under the *More Images* link, various images are available including options such as *biological assemblies*, *crystal packing*, and *ensemble images*. Also in the *Structural Features* option, more advanced RNA-containing structures are now available. The motif for the base pair signature is provided that is available as the common name of the motif.

The various annotations used in nucleic acids are nucleic acid type and conformation, secondary structure, and description. Also information pertaining to the drug binding mode and bound proteins are also available. A schematic representation is shown that involves the various steps involved in the extraction and monitoring of data from the NDB (**Fig. 1**). A schematic representation of the NDB is depicted in **Fig. 1**.

The resources available for NDB can be retrieved via two constant headers that are available on top page of the website. There are six tabs that are contained in the first constant header. They are: *About NDB*, *Standards*, *Education*, *Tools*, *Software*, and *Download*. Introduction to nucleic acids; definitions of terms used in the website; nucleic acid-related features from PDB-101, an educational component of RCSB PDB (Rose *et al.*, 2013), and links to other educational activities and sites. The software is downloadable includes geometric and symbolic 3D motif search (*FR3D*) (Sarver *et al.*, 2008), visualization of secondary structure (*RNA View*) (Yang *et al.*, 2003), and visualization of 3D structures (*3DNA*) (Lu and Olson, 2003). There are various links to the software packages that include various software for statistical folding of nucleic acids and studies of regulatory RNAs (*Sfold*; see "Relevant Websites section") (Ding and Lawrence, 2003; Ding *et al.*, 2005), a visualization applet for RNA (*VARNA*) (Darty *et al.*, 2009) and the *UNAFold* web server (Markham and Zuker, 2005, 2008).

A representation of the structural summary report for the NDB is clearly shown that represents the various parameters and information obtained from the protein (**Fig. 2**). From the schematic representation, a snapshot for the reported data summary of the information related to ribozymes from the NDB can be obtained.

The information illustrated in the NDB has a good connection with the information related to the PDB. The summary report obtained from the PDB that is retained in the NDB is summarized in **Table 1**. From the table, we can observe the association between the NDB and PDB in terms of their contents and other specifiers.

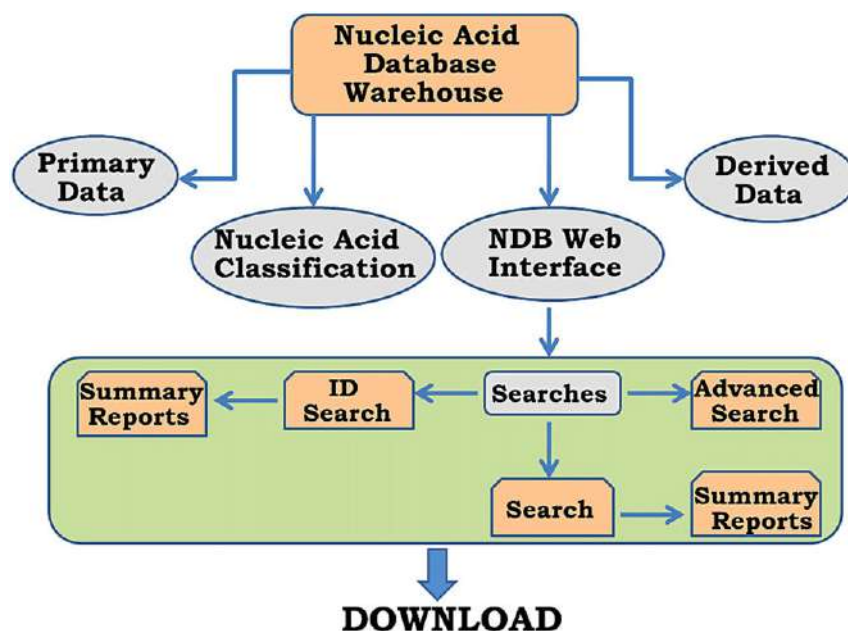


Fig. 1 A database curator workflow for the various steps involved in the extraction and monitoring of data obtained from the NAD. Understanding bioinformatics, Marketa Zvelebil & Jeremy O. Baum.

NDB User Interface

Both the primary and derived data classification is obtained in the NDB. The derived and calculated data based on the structural features of RNA from the manually annotated nucleic acid classification are separately managed from the entries of primary structure. The data thus recorded and stored are kept as external reference files (ERFs). The respective PDB structures contain the primary data entries including the experimental files, structural descriptions, citations, crystal data, identifiers, coordinate information, details regarding crystallization, data collection, and structure refinement. The coordinate data that are stored in the searchable database store the various derived structural features such as bond distances, angles, torsions, and base morphology (Lu and Olson, 2003, 2008). The information related to the RNA structural features including pair-wise nucleotide interactions for each RNA structure, equivalence classes, and NR sets of RNA structure files and RNA 3D motifs extracted from structures were recently reported.

NDB Applications

For RNA base-pairing and base stacking interactions the *FR3D* server (Sarver *et al.*, 2008) is used to annotate the pair-wise interactions between RNA nucleotides, and is described in (Zirbel *et al.*, 2009) for base phosphate interactions. The basis for RNA Base Triple Atlas and RNA 3D Motif Atlas are formed from these annotations. Also statistics about pair-wise interaction frequency are provided. For modelers and computational scientists who are keen to determine the characteristics of structured RNAs, these data can be a useful target. Reports suggest that the entries of RNA are clumped into 'equivalence classes' of structures that share the same, or nearly the same, sequence and geometry (Leontis and Zirbel, 2012). For every week the equivalence classes are computed such that the 3D structure databases are reflected with new additions. The same equivalence class contains different structures of the same RNA from the same organism whereas difference class contains the homologous RNAs structures from different organisms.

Other Databases

Apart from NDB, there also exist other databases that are unique for RNA and that combine data from the other sources including NDB and also from directed PDB sites. These however supply an elaborated annotation that is combined with the sequence based predictions in most of the cases (Appasamy *et al.*, 2016; Chojnowski *et al.*, 2014; Firdaus-Raih *et al.*, 2014; Popenda *et al.*, 2010; Zirbel *et al.*, 2015). The majority of the servers however permit secondary structures and RNA sequences searches. They also help in analyzing and searching from the uploaded PDB file the likely coordinates of RNA. These findings depend on the geometries but they are not sustained presently by NDB. RNA 3D hub is another useful database related to the structure of RNA that gives details about the various information associated with the precomputed structural analyses part of RNA molecules available at NDB (Zirbel *et al.*, 2015).

Databases such as NDB, G4LDB, SMMRNA, etc., have been provided by various groups earlier. These databases are dedicated particular type of nucleic acid or its special structure. Databases such as NDB consist of the various information related to three-dimensional structures of nucleic acids and their complexes.

[Search DNA](#) [Search RNA](#) [Advanced Search](#)

 Enter an NDB ID or PDB ID 
 Search for released structures

NDB ID: 4P95 **PDB ID: 4P95** 
Title:

SPECIATION OF A GROUP I INTRON INTO A LARIAT CAPPING RIBOZYME (CIRCULARLY PERMUTATED RIBOZYME)

Molecular Description:

RNA (189-MER)

Structural Keywords:
Nucleic Acid Sequence:
[Click to show/hide 1 nucleic acid sequences](#)
Protein Sequence:

No Protein Sequence Found

Primary Citation:

 Meyer, M., Nielsen, H., Olieric, V., Roblin, P., Johansen, S.D., Westhof, E., Masquida, B.
 Speciation of a group I intron into a lariat capping ribozyme. 
Proc.Natl.Acad.Sci.USA, **111**, pp. 7659 - 7664, 2014.

Experimental Information:

X-RAY DIFFRACTION

Space Group:

P 21 21 21

Cell Constants:

a = 57.63 b = 85.71 c = 108.53 (Ångstroms)

α = 90.0 β = 90.0 γ = 90.0 (degrees)

Refinement:

The structure was refined using the PHENIX program. The R value is 0.1906 for 34663 reflections in the resolution range 47.824 to 2.5 Ångstroms with Fobs > 1.59 sigma(Fobs) and with I > 0.0 sigma(I)

Download Data:
[Asymmetric Unit coordinates \(pdb format, Unix compressed\(.gz\)\)](#)
[Asymmetric Unit coordinates \(cif format, Unix compressed\(.gz\)\)](#)
[Biological Assembly coordinates \(pdb format\) 1](#)
[Structure Factors \(cif format\)](#)
[XML | Complete with coordinates \(xml format, GNU compressed\(.gz\)\)](#)
[XML | Coordinates only \(xml format, GNU compressed\(.gz\)\)](#)
[XML | Header only \(xml format, GNU compressed\(.gz\)\)](#)
Structural Features
[RNA Base-Pairs, Stacking, etc.](#)
[Similar Structures](#)
[Interactive basepair map with 3D fragment visualization for: 4P95](#) 
[Symbolic WebFR3D search for:4P95](#) 
[Geometric WebFR3D search for:4P95](#) 
[RNAML](#)
[Base Pair Hydrogen Bonding Classification](#)
[Nucleic Acid Backbone Torsions](#)
[Base Pair Morphology Parameters](#)
[Base Pair Morphology Step Parameters](#)
Biological Assembly 1

RNA View


Fig. 2 A reported summary of the particular ribozyme that depicts the headers of gray and red color with the individual sections: (a) Primary structural information, (b) atomic coordinate and experimental file download, (c) derived structural data, (d) images, and (e) for RNA structures an additional RNA view image. Understanding bioinformatics, Marketa Zvelebil & Jeremy O. Baum.

The G4LDB database (G4LDB, see “Relevant Websites section”) is related to the G-quadruplex ligands with the docking tool. This database gives information about the unique collection of the reported G-quadruplex ligands associated with the streamline ligand/drug discovery targeting G-quadruplexes. The G-quadruplexes are those nucleic acid sequences that are guanine-rich found in the human telomeres and the promoter region of the gene. This database at present consists of more than 800 G-quadruplex ligands with 4000 activity records, which is the most extensive collection to our knowledge. It also provides a user-friendly interface that necessitates various data inquired from researchers (Li *et al.*, 2012).

The SMMRNA database relates to the RNA structure and its complexes with small molecules (Li, *et al.*, 2013; Mehta *et al.*, 2014). This is an interactive database that is available at website provided in “Relevant Websites section” and focuses on the various small ligand

Table 1 NDB primary content acquired from PDB and its description

Main content of Nucleic Acids Database (NDB)	Explanation about the contents
Coordinate information	Description about the atomic coordinates for the asymmetrical and biological unit
Experimental files	The details about the <i>NMR</i> restraints and the structure factor files
Identifiers	The <i>IDs</i> for <i>NDB</i> and <i>PDB</i>
Structural description	The description about the asymmetric and biological unit, sequence, mismatches, base pairing, and modification
Citation	The details about the authors, title, journals, year, pages, volume, <i>DOI</i> , and <i>PUBMED ID</i>
Crystal data	A description about the space group information and cell parameters
Crystallization details	The conditions for temperature, <i>pH</i> , method, and crystallization
Data collection information	The information related to wavelength, temperature, radiation source, detector, resolution, <i>R</i> -merge, and number of reflections
Refinement details	Information regarding programs and method used, reflection number, resolution, <i>R</i> -factor, refinement of occupancies, and temperature factors

Table 2 Summary of the various nucleic acid databases with vivid description

Examples of Nucleic Acid Database (NDB)	Description of database
ProNIT	Provides experimentally determined thermodynamic interaction data between proteins and nucleic acids. It contains the properties of the interacting protein and nucleic acid, bibliographic information, and several thermodynamic parameters such as the binding constants, changes in free energy, enthalpy, and heat capacity
RNA FRABASE	An engine with database to search the 3D fragments within 3D RNA structures using as an input the sequences and/or secondary structures given in the dot-bracket notation
RNA Bricks	Database of RNA 3D structure motifs and their contacts, both with themselves and with proteins. The database provides structure-quality score annotations and tools for the RNA 3D structure search and comparison
RNA 3D Motif Atlas	RNA 3D motifs are recurrent structural modules that are essential for many biological functions and RNA folding. Usually drawn as unstructured hairpin and internal loops, these motifs are organized by noncanonical base pairs, supplemented by characteristic stacking and base-backbone interactions
RNAJunction	Provides a user-friendly way of searching structural elements by PDB code, structural classification, sequence, and keyword or interhelix angles. Contains structure and sequence information for RNA structural elements such as helical junctions, internal loops, bulges, and loop-loop interactions

molecules that target the RNA. At present this database contains approximately 770 unique ligands with the structural images pertaining to the RNA molecules. Also it gives information related to comprehensive resources that are required for further design, development, and refinement of small molecule modulators necessary for selective target of the RNA molecules (Mehta *et al.*, 2014).

Apart from G4LDB, there is also an important database consisting of detailed information related to the proteins' interaction with the nucleic acids forming a G-quadruplex structure, *Nucleic acid G-quadruplex structure (G4) Interacting Proteins DataBase (G4IPDB)*. This first database gives accurate knowledge related to the interaction at a single platform. It also consists of more than 200 entries with various interaction details such as interacting protein names and their synonyms, their UniProt-ID, source organism, target name and its sequences, ΔT_m , binding/dissociation constants, protein gene name, protein FASTA sequence, interacting residue in protein, related PDB entries, interaction ID, graphical view, PMID, author's name, and techniques that were used to detect their interactions. A vivid summary of the various examples of NDB with their detailed explanation is illustrated in Table 2. From the table, examples of various NDB have been described that give a clear picture about the interconnection with the nucleic acids.

Future Directions and Closing Remarks

After the redesign of NDB, new improvements can be noticed highlighting the data content that consists of derived data, annotations, and presentation. However, NDB will further continue to develop and expand its scope. The original NDB contained fewer than 100 crystal structures of nucleic acids but now has more than 700. As the technology is growing faster day by day, it will help us to observe a wide variety of structures in the NDB as it has permitted to determine number of structures. Also this database will facilitate the addition of more search options, annotations, visualizations, and various reports related to the nucleic acid-containing structures in the near future. Hence to our knowledge and information, no such database exists containing detailed experimental information related to drug-nucleic acid binding (K_d , T_m , IC_{50} , etc.,) for all the majorly available types and forms of nucleic acid. A clear view of the NDB can provide us much information related to the structure and functionality and also about the various sequences. Meanwhile the website has refurbished the information related to the query and its reporting capabilities. And this can help us to observe a wide variety of structures in the NDB as it has permitted to determine number of structures.

Acknowledgment

We thank the Central Library of Tezpur University for providing us resources to complete the article.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Assessment of Structure Quality (RNA and Protein). Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Biomolecular Structures: Prediction, Identification and Analyses. Computational Design and Experimental Implementation of Synthetic Riboswitches and Riboregulators. Data Storage and Representation. Engineering of Supramolecular RNA Structures. Genome-Wide Probing of RNA Structure. Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Predicting RNA-RNA Interactions in Three-Dimensional Structures. Protein Structure Databases

References

- Alberts, B., Johnson, A., Lewis, J., *et al.*, 2002. *Molecular Biology of the Cell*, fourth ed. New York: Garland Science the Structure and Function of DNA.
- Appasamy, S.D., Hamdani, H.Y., Ramlan, E.I., *et al.*, 2016. InterRNA: A database of base interactions in RNA structures. *Nucleic Acids Research* 44, D266–D271.
- Bailey, Regina. (2017). Learn about nucleic acids. ThoughtCo, thoughtco.com/nucleic-acids-373552.
- Bansal, M., 2003. DNA structure: Revisiting the Watson – Crick double helix. *Current Science* 85 (11), 1556–1563.
- Berman, H.M., Henrick, K., Nakamura, H., 2003. Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology* 10, 980.
- Berman, H.M., Olson, W.K., Beveridge, D.L., *et al.*, 1992. The Nucleic Acid Database – A comprehensive relational database of three-dimensional structures of nucleic acids. *Biophysical Journal* 3, 751–759.
- Chen, X., Ramakrishnan, B., Sundaralingam, M., 1995. Crystal structures of B-form DNA-RNA chimeras complexed with distamycin. *Nature Structural Biology* 2 (9), 733–735. doi:10.1038/nsb0995-733.
- Chojnowski, G., Walen, T., Bujnicki, J.M., 2014. RNA Bricks – A database of RNA 3D motifs and their interactions. *Nucleic Acids Research* 42, D123–D131.
- Churchill, M.E.A., 1996. The latest on DNA form and function. In: Eckstein, Fritz, Lilley, David MJ (Eds.), *Nucleic Acids and Molecular Biology*, vol. 9. Berlin, Heidelberg: SpringerVerlag, p. 376.
- Coimbatore Narayanan, B., Westbrook, J., Ghosh, S., *et al.*, 2014. The nucleic acid database: New features and capabilities. *Nucleic Acids Research* 42, D114–D122.
- Darnell Jr., J.E., 1985. RNA. *Scientific American* 253, 54–72.
- Darty, K., Denise, A., Ponty, Y., 2009. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25. . pp. 1974–1975.
- Dickerson, R.E., Drew, H.R., Conner, B.N., *et al.*, 1982. The anatomy of A-, B-, and Z-DNA. *Science* 216 (4545), 475–485. doi:10.1126/science.7071593. PMID 7071593.
- Ding, Y., Chan, C.Y., Lawrence, C.E., 2005. RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA* 11, 1157–1166.
- Ding, Y., Lawrence, C.E., 2003. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Research* 31, 7280–7301.
- Felsenfeld, G., 1985. DNA. *Scientific American* 253, 54–72.
- Firdaus-Raihi, M., Hamdani, H.Y., Nadzirin, N., *et al.*, 2014. COGNAC: A web server for searching and annotating hydrogen-bonded base interactions in RNA three-dimensional structures. *Nucleic Acids Research*, 42. pp. W382–W388.
- Govil, G., Kumar, N.Y., Ravi Kumar, M., *et al.*, 1985. Recognition schemes for protein–nucleic acid interactions. *Journal of Bioscience* 8, 645. doi:10.1007/BF02702763.
- Katsuyuki, A., Kazutaka, M., Hu, N., 2016. Chapter 3, section 3. Nucleic Acid Constituent complexes. In: Sigel, A., Sigel, H., Sigel, R.K.O. (Eds.), *The Alkali Metal Ions: Their Role in Life. Metal Ions in Life Sciences* 16. Springer, pp. 43–66. doi:10.1007/978-3-319-21756-7_3.
- Krieger, M., Scott, M.P., Matsudaira, P.T., *et al.*, 2004. Section 4.1: Structure of nucleic acids. In: *Molecular Cell Biology*. New York: W.H. Freeman and CO, ISBN 0-7167-4366-3.
- Leontis, N.B., Zirbel, C.L., 2012. Nonredundant 3D structure datasets for RNA knowledge extraction and benchmarking. In: Leontis, N.B., Westhof, E. (Eds.), *RNA 3D Structure Analysis and Prediction* 27. Berlin, Heidelberg: Springer, pp. 281–298.
- Li, Q., Xiang, J.F., Zhang, H., *et al.*, 2012. Searching drug-like anti-cancer compound(s) based on G-quadruplex ligands. *Curr. Pharm. Des.* 18, 1973–1983.
- Li, Q., Xiang, J.F., Yang, Q.F., *et al.*, 2013. G4LDB: A database for discovering and studying G-quadruplex ligands. *Nucleic Acids Research* 41, D1115–D1123.
- Luger, K., Phillips, S.E.V., 2010. Protein-nucleic acid interactions. *Current Opinion in Structural Biology* 20 (1), 70–72. doi:10.1016/j.sbi.2010.01.006.
- Lu, X.J., Olson, W.K., 2003. 3DNA: A software package for the analysis, rebuilding and and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Research* 31 (17), 5108–5121.
- Lu, X.J., Olson, W.K., 2008. 3DNA: A versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nature Protocols* 3, 1213–1227.
- Markham, N.R., Zuker, M., 2005. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Research* 33, W577–W581.
- Markham, N.R., Zuker, M., 2008. UNAFold: Software for nucleic acid folding and hybridization. *Methods in Molecular Biology* 453, 3–31.
- Mehta, A., Sonam, S., Gouri, I., *et al.*, 2014. SMMRNA: A database of small molecule modulators of RNA. *Nucleic Acids Research* 42, D132–D141.
- Mirkin, S.M., 2001. DNA Topology: Fundamentals. *Encyclopedia of Life Sciences*. doi:10.1038/hpg.els.0001038.
- Park, B., Kim, H., Han, K., 2014. DBBP: Database of binding pairs in protein-nucleic acid interactions. *BMC Bioinformatics* 15 (Suppl. 15), S5. doi:10.1186/1471-2105-15-S15-S5.
- Popenda, M., Szachniuk, M., Blazewicz, M., *et al.*, 2010. RNA FRABASE 2.0: An advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures. *BMC Bioinformatics* 11, 231.
- Rose, P.W., Bi, C., Bluhm, W.F., *et al.*, 2013. The RCSB Protein Data Bank: New resources for research and education. *Nucleic Acids Research* 41, D475–D482.
- Sarver, M., Zirbel, C.L., Stombaugh, J., Mokdad, A., Leontis, N.B., 2008. FR3D: Finding local and composite recurrent structural motifs in RNA 3D structures. *Journal of Mathematical Biology* 56, 215–252.
- Sathyapriya, R., Vijayabaskar, M.S., Vishveshwara, S., 2008. Insights into Protein – DNA interactions through structure network analysis. *PLOS Computational Biology* 4 (9), e1000170. doi:10.1371/journal.pcbi.1000170.
- Siggers, T., Gordan, R., 2014. Protein – DNA binding: Complexities and multi-protein codes. *Nucleic Acids Research* 42 (4), 2099–2111. doi:10.1093/nar/gkt1112.
- Travers, A., Muskhelishvili, G., 2015. DNA Structure and Function. *FEBS Journal* 282, 2279–2295.
- Valuchova, S., *et al.*, 2016. A rapid method for detecting protein-nucleic acid interactions by protein induced fluorescence enhancement. *Scientific Reports* 6, 39653. doi:10.1038/srep39653.
- Watson, J.D., Crick, F.H.C., 1953. Molecular structure of nucleic acids: A Structure for deoxyribose nucleic acid. *Nature* 4356, 737.
- Weinberg, R.A., 1985. The Molecules of Life. *Scientific American* 253, 34–43.
- Yang, H., Jossinet, F., Leontis, N., *et al.*, 2003. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Research* 31, 3450–3460.

- Zirbel, C.L., Sporer, J.E., Sporer, J., Stombaugh, J., Leontis, N.B., 2009. Classification and energetics of the base-phosphate interactions in RNA. *Nucleic Acids Research* 37, 4898–4918.
- Zirbel, C.L., Roll, J., Sweeney, B.A., *et al.*, 2015. Identifying novel sequence variants of RNA 3D motifs. *Nucleic Acids Research* 43, 7504–7520. doi:10.1093/nar/gkv651.

Relevant Websites

<http://www.g4ldb.org>
G-Quadruplex Ligands Database.

<http://ndbserver.rutgers.edu>
Nucleic Acid Database.

<ftp://ndbserver.rutgers.edu/NDB/>
Nucleic Acid Database.

<http://sfold.wadsworth.org>
Sfold.

<http://www.smmrna.org>
SMMRNA.

Biographical Sketch



Airy Sanjeev is presently working as a research scholar in the Department of Molecular Biology and Biotechnology in Tezpur University under the supervision of Dr. Venkata Satish Kumar Mattaparthi. Her research work is based on understanding the aggregation pathway of α -synuclein which is responsible for Parkinson's Disease. She has expertise in various softwares such as AutoDock, PyRex, Schrodinger, AMBER, GROMACS, and also programming languages such as Perl, Java, C, C++ . She has also skills in operating systems, protein modeling, graphing and data analysis suites, etc. She has published almost 10 papers in the peer-reviewed journals. She has completed her Masters from Pondicherry University in 2015 in Centre for Bioinformatics. She did her undergraduate work at Cotton College, Guwahati, Assam with Botany as her honors subject.



Dr. Venkata Satish Kumar Mattaparthi received his PhD in Biotechnology from Indian Institute of Technology Guwahati, Assam, India in 2010. From 2010 to 2012, he was with the Centre for Condensed Matter Theory (CCMT), Department of Physics, Indian Institute of Science Bangalore, India. He is currently working as Assistant Professor in the Department of Molecular Biology and Biotechnology, School of Sciences, Tezpur University, Assam, India. His research interests include understanding the functioning of intrinsically disordered proteins (IDPs), pathways of protein aggregation mechanism in neurodegenerative diseases like Alzheimer, Parkinson, etc., and also the features of protein–RNA interactions using computational techniques. He has published his research in reputed international journals like *Journal of Biomolecular Structure and Dynamics*, *PLoS ONE*, *Journal of Physical Chemistry B*, *Biophysical Journal*, *Journal of Bioinformatics and Computational Biology*, and others.



Dr. Sandeep Kaushik is a passionate computational biologist with a wide exposure of computational approaches. Presently, he is carrying out his research as an Assistant Researcher (equivalent to Assistant Professor) at 3B's Research Group, University of Minho, Portugal. He has a PhD in Bioinformatics from National Institute of Immunology, New Delhi, India. He has a wide exposure on analysis of scientific data ranging from transcriptomic data on mycobacterial and human samples to genomic data from wheat. His research experience and expertise entails molecular dynamics simulations, RNA-sequencing data analysis using Bioconductor (R language based package), de novo genome assembly using various software like *AbySS*, *automated protein prediction and annotation*, *database mining*, *agent-based modeling and simulations*. He has a cumulative experience of more than 10 years of programming using PERL, R language and NetLogo. He has published his research in reputed international journals like *Molecular Cell*, *Biomaterials*, *Biophysical Journal*, and others. Currently, his research interest involves in silico modeling of disease conditions like breast cancer, using agent-based modeling and simulations approach.

RNA Structure Prediction

Junichi Iwakiri and Kiyoshi Asai, University of Tokyo, Kashiwa, Japan and Artificial Intelligence, Research Center, Koto-ku, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Three major types of RNAs, namely messenger RNA (mRNA) as coding RNA, as well as transfer RNA (tRNA) and ribosomal RNA (rRNA) as non-coding RNAs, are involved in the translation of genetic code. Among them, tRNAs fold into the common clover leaf-like secondary structure, having anti-codons that interact with the codons of the mRNAs, to perform their function as adaptors during the translation of mRNAs into proteins. Such common secondary structures are often observed in other non-coding RNAs such as snoRNAs and various snRNAs involved in splicing. It is also known that the secondary structural stability of mRNAs, especially around the start codon, affects the efficiency of translation. Micro RNAs (miRNAs) repress the translation of target mRNAs, are first transcribed as pri-miRNAs to form the stem-loop secondary structures for processing to double-stranded pre-miRNAs and then to mature miRNAs. Many non-coding RNAs other than miRNAs can be transcribed, but few of their functions have been identified. The functions of RNA molecules often depend on their high sequence specificity for interacting with other RNAs. These interactions are based on the base pairing of complementary bases, which are also observed in the secondary structures. Thus, it is important to identify the secondary structures of RNAs to elucidate and understand their functions. In this article, we introduce currently available knowledge on the structure prediction of RNAs with a particular focus on secondary structures.

A number of software tools related to RNA secondary structures are available. Researchers should select the appropriate tools for their specific purposes. It should be noted that Vienna RNA package (Lorenz *et al.*, 2011) is the most popular and convenient suite of the tools and the software libraries.

Models of RNA Secondary Structures

Secondary Structure

Single-stranded RNA molecules form hydrogen bonds between the Watson-Crick (WC) edges of the complementary pairs of bases, namely the canonical WC pairs A-U and G-C, and the wobble pair G-U. The set of base-pairs of an RNA molecule is called the secondary structure of the RNA. RNA molecules fold into tertiary structures via complicated interactions inside of the molecules, but the secondary structures are often used as abstract forms of the RNA structure. It is known that the secondary structures are often well conserved between RNAs of the same functional family.

The secondary structure of an RNA sequence $x = x_1 \dots x_L$ of the length L is represented by an upper triangle binary matrix, specifically the secondary structure matrix (2DSM), $\{\sigma_{ij}\}$ ($1 \leq i < j \leq L$) where in the secondary structure $\sigma_{ij} = 1$ shows that x_i and x_j form a base pair, $\sigma_{ij} = 0$ shows that there is no base pair between them, and σ_{ij} satisfies $\sum_i \sigma_{ij} \leq 1$ and $\sum_j \sigma_{ij} \leq 1$. The interactions of the base pairs are illustrated in Fig. 1.

Pseudoknots

Fig. 2 presents examples of non-crossing and crossing interactions in secondary structures. Secondary structures with such crossing interactions are called *pseudo-knots*. The 2DSM of a pseudoknot-free secondary structure σ_{ij} satisfies: $\sigma_{ij}\sigma_{k\ell} = 0$ if $i < k < j < \ell$. The majority of software tools for RNA secondary structure prediction only produce pseudoknot-free structures.

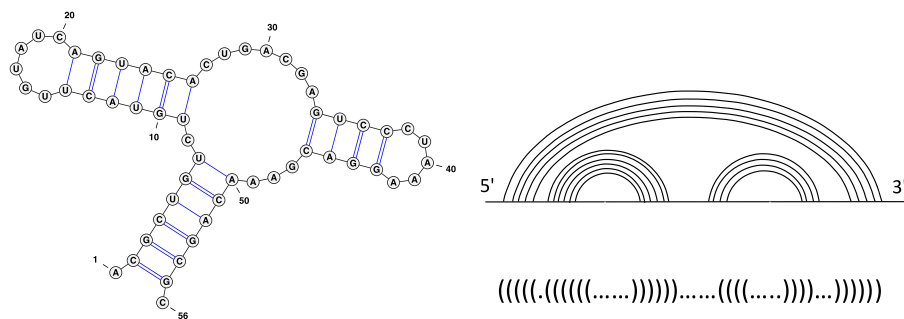


Fig. 1 Representations of a secondary structure (Hammerhead ribozyme). Two-dimensional (2D) representation (left), linear representation (right top) and sequence representation using parentheses (right bottom).

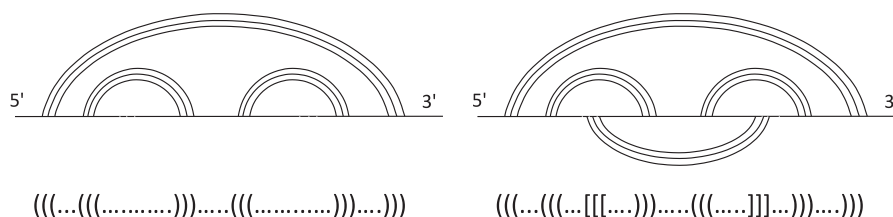


Fig. 2 Interactions in RNA secondary structures. The interactions of base pairs without pseudoknots (left) and with pseudoknots (right) are shown.

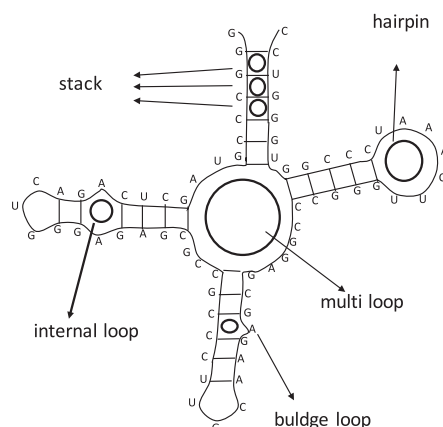


Fig. 3 Loops in secondary structures.

Extended secondary structures

Three-dimensional (3D) RNA structures consist of WC base pairs and non-WC base pairs and various hydrogen bonds. Such base-pairing interactions are categorized into 12 groups based on the combination of interacting edges (WC, Hoogsteen and sugar) in nucleotide bases and the relative orientations of glycosidic bonds (cis or trans) (Leontis *et al.*, 2002), which totally form the *extended secondary structures*. Among these groups, cis WC-WC base pairs are the standard WC base pairs, and the remaining 11 groups are non-WC base pairs.

Energy Models

An energy model of RNA secondary structures and their parameters define the free energy of a given RNA sequence. The most widely used energy model for pseudoknot-free 2D structures is Turner nearest neighbor model (Mathews *et al.*, 1999; Mathews, 2004). The Turner model assumes that the free energy of a secondary structure is the sum of all the *loops* surrounded by the backbone and the base-pairing hydrogen bonds, such as stacks, internal loops, bulge loops, hairpin loops, and multi-loops (Fig. 3). The free energy of the whole RNA structure is given by

$$E(x, \sigma) = \sum_{\sigma_i \in \sigma} E(\sigma_i) \quad (1)$$

The parameters of the Turner model represent the *nearest neighbors* of those loops and the length of the included nucleotides, as determined experimentally (Mathews *et al.*, 1999; Mathews, 2004).

The constraint generation (CG) method (Andronescu *et al.*, 2007) estimates Turner parameters from structural and experimental data. It applies the constraint that the reference structure has lower energy than that of alternative structures and achieved better accuracy in secondary structure prediction. Boltzmann Likelihood (BL) method for the Turner model further improved the predictive accuracy of secondary structures. The energy parameters have also been calculated by combination of experiments and molecular dynamics calculations (Sakuraba *et al.*, 2015).

The number of model parameters is the key issue in secondary structure prediction. Approximately 70,000 free parameters were used in ContextFold (Zakov *et al.*, 2011), and 50% reduction in error rate was reported, but there is serious danger for overfitting.

Boltzmann distribution

Given an RNA sequence x , the probability distribution of the secondary structure σ with free energy $E(x, \sigma)$ is written by the Boltzmann distribution,

$$P(\sigma|x) = \frac{1}{Z(x)} e^{-\frac{E(x,\sigma)}{RT}} \quad (2)$$

where $Z(x)$, the partition function of this distribution, is the sum of the Boltzmann factors for the set of all possible secondary structures S , as follows:

$$Z(x) = \sum_{\sigma \in S} e^{-\frac{E(\sigma,x)}{RT}} \quad (3)$$

This partition function $Z(x)$ can be calculated efficiently, costing $O(L^3)$ in time and $O(L^2)$ in memory for an RNA sequence of the length L , via the McCaskill algorithm (McCaskill, 1990), which is a dynamic programming (DP) method that will be explained in a subsequent section.

Stochastic Models

The Turner energy model and its parameters represent the thermodynamic character of the secondary structure and consequently define the probability via the Boltzmann distribution. Meanwhile, stochastic models can directly define the probabilities of the secondary structures. Stochastic context free grammar (SCFG) was first used to model the secondary structure of tRNA (Sakakibara *et al.*, 1994), and it has been widely used for pseudoknot-free secondary structure prediction.

For example, the following set of stochastic production rules gives a type of stochastic model of RNA secondary structure:

$$S \rightarrow xS\bar{x} \text{ with probability } P(S \rightarrow xS\bar{x}) \quad (4)$$

$$S \rightarrow x \text{ with probability } P(S \rightarrow x) \quad (5)$$

where $\bar{x}$ represents the complementary base of x . Using more complicated production rules, SCFG can represent the nearest neighbor parameters (Knudsen and Hein, 2003; Rivas *et al.*, 2012). In the stochastic models of RNA secondary structures, the set of parameters are learned from the set of RNA sequences for which the secondary structures are determined.

Do *et al.* maximized the conditional likelihood of the stochastic log-linear model for a structural dataset by using a gradient descent for optimization in their secondary structure prediction software CONTRAfold (Do *et al.*, 2006). The CONTRAfold model does not reliably determine free-energy changes, but it is important for estimating bindings.

To treat pseudoknotted secondary structures using stochastic models, formal grammars beyond context-free grammar are necessary (Rivas and Eddy, 2000). The complexity of the formal grammar depends on the complexity of pseudoknots.

Secondary Structure Prediction

We mainly treat the simplest problem in RNA secondary structure prediction to predict the secondary structure of a single RNA sequence for pseudoknot-free structures without any additional information. To solve this problems, several software tools are available, including Mfold (Zuker, 2003), RNAfold (Lorenz *et al.*, 2011), Pfold (Knudsen and Hein, 2003), RNAstructure (Mathews, 2014), CONTRAfold (Do *et al.*, 2006), CentroidFold (Sato *et al.*, 2009; Hamada *et al.*, 2009a), TurboFold (Harmanci *et al.*, 2011), and ContextFold (Zakov *et al.*, 2011).

Meanwhile, tools for pseudoknotted secondary structures including Pknots (Rivas and Eddy, 1999), IPknot (Sato *et al.*, 2011), tend to incorrectly predict pseudoknot-free structures. They are useful for finding pseudoknotted local structures, but often show lower total accuracy even for RNAs including pseudoknotted structures.

RNA secondary structure prediction is reasonably accurate for RNAs shorter than 1000, but the accuracy is often unsatisfactory for longer RNAs. The accuracy is improved using homologous sequences (Hamada *et al.*, 2009b) because the local secondary structures are often evolutionarily conserved.

There are several variations of secondary structure prediction problems, such as the prediction of suboptimal structures, the common secondary structure and alignment of a group of sequences, the common secondary structure from aligned sequences, the secondary structure of an interacting pair of sequences, and the secondary structure combining experimental data.

Criteria of Secondary Structure Prediction

Prediction of minimum free energy structures

The Boltzmann distribution Eq. (2) of the secondary structure indicates that the maximum likelihood estimator (MLE) $\hat{\sigma}^{MLE}$ of the secondary structure is the MFE structure $\hat{\sigma}^{MFE}$ as follows.

$$\hat{\sigma}^{MLE} = \operatorname{argmax}_{\sigma} P(\sigma|x) = \operatorname{argmin}_{\sigma} E(x, \sigma) = \hat{\sigma}^{MFE} \quad (6)$$

The dynamic programming algorithm proposed by Zuker and Stiegler (1981) computes the MFE/MLE structure efficiently costing $O(L^3)$ in time and (L^2) in memory. Mfold and RNAfold of the Vienna RNA package are popular tools for finding MFE structures via dynamic programming.

It is natural to believe that the MFE/MLE structures are the *best* structures we can predict, but the probability of a specific secondary structure is extremely small. For example, the probability of MFE structure of a tRNA is often less than 1%, and for

longer RNAs it is easily less than 10^{-8} . Whereas the MFE/MLE is an extremely important criterion, the MFE/MLE structures are not as reliable as they appear considering their thermodynamic probability.

Predictions Based on the Maximum Expected Accuracy

The MLE structure $\hat{\sigma}^{MLE}$ in Eq. (6) can be re-written as follows:

$$\hat{\sigma}^{MLE} = \operatorname{argmax}_{\sigma} P(\sigma|x) = \operatorname{argmax}_{\sigma} \sum_{\hat{\sigma}} \delta(\sigma, \hat{\sigma}) P(\sigma|x) \quad (7)$$

where $\delta(a,b)$ is the δ function, which is 1 only if the two parameters are equal and 0 otherwise. According to the aforementioned equation, the MLE structure, or equivalently the MFE structure, maximizes the probability that the predicted structure is exactly the same as the *true* structure. This means that no neighboring structure different from the *true* structure, is considered in the evaluation.

In CONTRAfold, which applies the criterion of MEA, the gain function, which reflects an accuracy measure of the secondary structure prediction, was used instead of using the δ function in Eq. (7) as follows:

$$G^{MEA} = TN_{\text{position}} + \gamma \times TP_{\text{position}} \quad (8)$$

where TN_{position} and TP_{position} are, the numbers of positions correctly predicted as unpaired (*true negative*), and paired (*true positive*), respectively. The γ is the sensitivity/specificity tradeoff parameter. Whereas this gain function uses the true negatives and positives with respect to the positions, the accuracy of secondary structure prediction has been usually measured using the number of correctly paired/unpaired *base pairs* which CentroidFold adopted in its gain function

$$G^{\gamma} = TN_{\text{base-pair}} + \gamma \times TP_{\text{base-pair}} \quad (9)$$

Marginal Probabilities and Reliability of Predictions

Marginal Probabilities of Secondary Structures

Although the probability that an RNA sequence will form a specific secondary structure is extremely small, the marginal probabilities of the structure are often reasonably large. The most popular marginal probability is the base-pairing probability (BPP), or the probability that the specific pair of bases form a base pair in the secondary structure. BPPs include rich information regarding the probability distribution of the RNA secondary structure. They are often illustrated in an upper triangular matrix (Lorenz *et al.*, 2011) suggesting the existence of hidden alternative structures, or presented on the predicted secondary structure with colors that illustrate the reliability of the stem structures.

The marginal probabilities of various structural contexts, such as stems, loops, and bulges, also carry rich information about the secondary structure. Although it is difficult to detect distinct structural motifs from a set of functional RNAs, local pattern of marginal probabilities has been observed (Fukunaga *et al.*, 2014).

Reliability of Predicted Structures

The accuracies of predictions for secondary structures are often evaluated regarding the accuracy of base pair prediction using sensitivity (SEN, or recall), the positive predicted value (PPV, or precision) and balanced measures such as Matthew's correlation coefficient (MCC) and F-measures, defined as follows.

$$\text{SEN} = \frac{TP}{TP + FN} = \frac{\# \text{ of correctly predicted base - pairs}}{\# \text{ of true base - pairs}} \quad (10)$$

$$\text{PPV} = \frac{TP}{TP + FP} = \frac{\# \text{ of correctly predicted base - pairs}}{\# \text{ of predicted base - pairs}} \quad (11)$$

$$\text{MCC} = \frac{TP \cdot TN + FT \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (12)$$

$$\text{F-measure} = 2 \frac{\text{PPV} \cdot \text{SEN}}{\text{PPV} + \text{SEN}} \quad (13)$$

Whereas base-pairing probabilities are convenient for partially representing the probability distribution, they do not clearly reveal the entire picture of the distribution. RNAbor (Freyhult *et al.*, 2007) computes the marginal probability of secondary structures on the Hamming distance from the reference structure. Mori *et al.* (2014) has formalized the calculation of Distribution of the Hamming distance based on the general scheme proposed by Newberg and Lawrence (2009). When the sum of the probabilities within a range of Hamming distance exceeds $c\%$, the smallest such Hamming distance is called the $c\%$ *credibility limit*. A lower credibility limit with a higher percent indicates that the distribution is more concentrated, and that the prediction is more reliable.

Algorithms on RNA Secondary Structures

Dynamic Programming Algorithms

As described previously, pseudoknot-free RNA secondary structure can be modeled using SCFGs corresponding to pseudoknot-free energy models. The DP algorithms for pseudoknot-free RNA secondary structures therefore corresponds to the standard SCFG algorithms.

Dynamic programming for partition function

The partition function Eq. (3) of the Boltzmann distribution Eq. (2) can be calculated by the McCaskill algorithm (McCaskill, 1990), a DP algorithm corresponding to the inside algorithm of SCFG.

The McCaskill algorithm

Initialization: for $1 \leq i \leq N$

$$Z_{i,i} = 1.0, \quad Z_{i,i}^1 = Z_{i,i}^b = Z_{i,i}^m = Z_{i,i-1}^m = Z_{i,i}^{m1} = 0.0$$

Recursion: for $1 \leq i < j \leq N$ (from short intervals to long intervals)

$$Z_{i,j} = 1.0 + \sum_{h=i-1}^{j-1} Z_{i,h} Z_{h+1,j}^1 \quad (14)$$

$$Z_{i,j}^1 = \sum_{h=i+1}^j Z_{i,h}^b$$

$$\begin{aligned} Z_{i,j}^b = & e^{-f_1(i,j)/kT} + \sum_{h=i+1}^{j-2} \sum_{\ell=h+1}^{j-1} Z_{h,\ell}^b e^{-f_2(i,j,h,\ell)/kT} \\ & + \sum_{h=i+2}^{j-1} Z_{i+1,h-1}^m Z_{h,j-1}^{m1} e^{-f_3(i,j)/kT} \end{aligned} \quad (15)$$

$$Z_{i,j}^m = \sum_{h=i+1}^{j-1} (e^{-f_4(i,h-1)/kT} + Z_{i,h-1}^m) Z_{h,j}^{m1} e^{-f_5/hT} \quad (16)$$

$$Z_{i,j}^{m1} = \sum_{h=i+1}^j Z_{i,h}^b e^{-f_4(h+1,j)/kT}$$

$Z_{i,j}$ is the partition function of subsequence $[i, j]$, $Z_{i,j}^b$ is that of i - j base pairing, and $Z_{i,j}^m$ is that of multi-loop. $Z_{i,j}^1$ and $Z_{i,j}^{m1}$ are the partitions functions that include exactly one outmost base pair. The functions f_1 to f_5 reflect the free energy of unpaired bases, depending on the structural contexts.

Dynamic programming for MFE

The first reported DP algorithm for secondary structure prediction was the Nussinov algorithm (Nussinov *et al.*, 1978). It maximizes the number of base pairs in the secondary structure, which we can regard as a rough approximation of MFE, by iterating the following equations from short intervals to long intervals:

$$M(i,j) = \max \begin{cases} M(i+1, j) \\ M(i, j-1) \\ M(i+1, j-1) + 1 \text{ if } (i,j) \text{ can form a base pair} \\ \max_h M(i, h) M(h+1, j) \end{cases}$$

The Nussinov algorithm correspond to the CYK algorithm (Kasami *et al.*, 1965; Younger, 1967) for the parsing of SCFGs. with complexities $O(L^3)$ in time and $O(L^2)$ in memory for the length L of the sequence. Assuming a Turner type energy model for pseudoknot-free structures, MFE structures can be calculated via Zuker algorithm (Zuker and Stiegler, 1981), a CYK-like DP algorithm, using more complicated iterations, but with complexities $O(L^3)$ in time and $O(L^2)$ in memory.

DP for Base-Pairing Probability

The BPP of RNA is the marginal probability that a specific pair of bases will form a base pair in the secondary structure. BPPs are calculated using inside/outside partition functions as follows:

$$P_{i,j}^b = \frac{Z_{ij}^b W_{ij}^b}{Z_{1L}}$$

where Z_{ij}^b is the *inside* partition function of subsequence $[i, j]$ when the i -th and j -th bases form a base pair, which appeared in Eq. (15) of the McCaskill algorithm, and W_{ij}^b is the corresponding *outside* partition function, which is the sum of all the Boltzmann factors outside of $[i, j]$ when the i -th and j -th bases form a base pair. The $W^b(i, j)$ is calculated by a DP algorithm, which is the outside algorithm of the McCaskill (inside) algorithm for partition function. A DP algorithm that directly calculate BPPs similar to outside algorithm for $W^b(i, j)$ was reported in the same study reporting the McCaskill algorithm (McCaskill, 1990).

The computational complexity of those DP algorithms is also $O(L^3)$ in time and $O(L^2)$ in memory. For long RNA sequences, $O(L^3)$ in time and $O(L^2)$ are too expensive for practical analysis. By limiting the maximal span of base pairs to W , Rfold (Kiryu *et al.*, 2008) reduces them to $O(W^2L)$ in time and $O(L + W^2)$ in memory. The Rfold algorithm has been extended to parallel computations in ParasoR (Kawaguchi and Kiryu, 2016), including calculations of marginal probabilities implemented in CapR (Fukunaga *et al.*, 2014). Because $O(W^2L)$ is linear with respect to sequence length L , these tools are applicable for analyzing long non-coding RNAs.

Posterior Decoding for Maximum Expected Gain Prediction

As described in Predictions Based on the Maximum Expected Accuracy (MEA), predictions based on MEA are maximum expected gain estimators with the associated gain functions. The MEA estimator of CONTRAfold and the γ -centroid estimator of CentroidFold are calculated by the using dynamic programming as follows (Do *et al.*, 2006):

$$M(i, j) = \max \begin{cases} M(i+1, j) \\ M(i, j-1) \\ M(i+1, j-1) + g(\gamma, P_{ij}^b) \\ \max_h M(i, h)M(h+1, j) \end{cases}$$

where $g(\gamma, P_{ij}^b) = 2\gamma P_{ij}^b - (1 - \sum_i P_{ij}^b) - (1 - \sum_j P_{ij}^b)$ for the MEA estimator, and $g(\gamma, P_{ij}^b) = (\gamma + 1)P_{ij}^b - 1$ for the γ -centroid estimator. P_{ij}^b is the BPP.

Such DP is called *posterior decoding* of the marginal probability. Note that the DP in this case is extremely simple and similar to the Nussinov algorithm.

Additional Information on Secondary Structure Prediction

RNA Secondary Structure Prediction Incorporating Experimental Data

In recent decades, experimental investigations have been common approaches for understanding RNA structures. In the experimental approaches, recently emerging high-throughput sequencing technology changed the scale of target RNAs from a single transcript to the whole transcriptome. Currently, experimental investigations of RNA structures are categorized into the following two approaches: (1) *in vitro/in vivo* probing of RNA base pairs, (2) *in vivo* probing of RNA duplexes.

In the first approach, the pairing state (paired or unpaired) of individual bases in RNA molecules is analyzed using chemical and enzymatic probing. Selective 2'-hydroxyl acylation analyzed via primer extension and sequencing (SHAPE-seq) (Loughrey *et al.*, SHAPE; Lucks *et al.*, 2011), SHAPE and mutational profiling (SHAPE-MaP) (Siegfried *et al.*, 2014; Smola *et al.*, 2015), *in vivo* click selective 2'-hydroxyl acylation and profiling experiment (icSHAPE) (Flynn *et al.*, 2016; Spitale *et al.*, 2015), Structure-seq (Ding *et al.*, 2015), dimethyl sulfate (DMS) treatment and sequencing (DMS-seq) (Rouskin *et al.*, 2014), DMS mutational profiling with sequencing (DMS-MaPseq) (Zubradt *et al.*, 2017), Mod-seq (Talkish *et al.*, 2014), and chemical inference of RNA structures followed by massive parallel sequencing (CIRS-seq) (Incarnato *et al.*, 2014) are categorized as chemical probing methods employing various chemical compounds reacting with unpaired nucleotides (single-stranded regions) in RNA molecules. Fragmentation sequencing (Fragseq) (Underwood *et al.*, 2010), parallel analysis of RNA structure (PARS) (Kertesz *et al.*, 2010; Wan *et al.*, 2014) and parallel analysis of RNA structures with temperature elevation (PARTE) (Wan *et al.*, 2012) are enzymatic probing methods using structure-specific RNase, including RNase V1 and S1, for identifying both paired and unpaired nucleotides. For all of the aforementioned methods, raw sequencing reads must be processed to obtain the base-pairing or single-stranded signals of RNA bases at single-nucleotide resolution. The signal data derived from the limited number of structure probing experiments, such as icSHAPE and DMS-seq, is pre-computed and provided by Structure Surfer database (Berkowitz *et al.*, 2016) and FoldAtlas (Norris *et al.*, 2017).

As the second approach, RNA proximity ligation (RPL) (Ramani *et al.*, 2015), psoralen analysis of RNA interactions and structures (PARIS) (Lu *et al.*, 2016), ligation of interacting RNA and high-throughput sequencing (LIGR-seq) (Sharma *et al.*, 2016), sequencing of psoralen crosslinked, ligated and selected hybrids (SPLASH) (Aw *et al.*, 2016), and mapping RNA interactome *in vivo* (MARIO) (Nguyen *et al.*, 2016) employ proximity ligation of RNA duplexes to directly identify two RNA regions involved in base-pairing interactions.

The base-pairing or single-stranded signals derived from the probing experiments can be combined with computational prediction of RNA secondary structures to improve their prediction accuracy. In MC-fold prediction (Parisien and Major, 2008), three grades of energetic penalties are applied according to the SHAPE signal of each nucleotide. RNAstructure (Deigan *et al.*, 2009), RNAsc (Zarringhalam *et al.*, 2012), and RNAbifold (Washietl *et al.*, 2012) incorporate the signal data into their energy calculation as pseudo-energy contribution. Currently, these three approaches are also implemented in the RNAfold Vienna RNA package (Lorenz *et al.*, 2016). As a sampling-based approach, Seqfold (Ouyang *et al.*, 2013) was developed for incorporating various type of probing data, such SHAPE-seq, PARS, and FragSeq data.

Non-DP Algorithms

Stochastic sampling

In order to estimate marginal probabilities of the secondary structures of RNA, stochastic sampling can be used (Ding and Lawrence, 2003). In stochastic sampling, the structures are sampled from Boltzmann distribution of the secondary structures.

Integer programming

It is computationally expensive to predict pseudoknotted secondary structures using DP algorithms. Integer programming has been used to predict pseudoknotted secondary structures in IPknot (Sato *et al.*, 2011). Predicting the joint secondary structure of two interacting RNA sequences has the same computational complexity as pseudoknotted secondary structure prediction, and integer programming has been applied in RactIP (Kato *et al.*, 2010).

Common Secondary Structure Prediction

RNA secondary structures often conserved during evolution. Therefore, it is important to detect the common secondary structure of the related sequences and find the functions correlated to the secondary structures. Several tools exist for locating the consensus secondary structures from multiple alignments of RNA sequences, such as CentroidAlifold (Hamada *et al.*, 2011) and RNAalifold (Bernhart *et al.*, 2008). PPFold (Sukosd *et al.*, 2012) incorporates SHAPE data in predicting the consensus secondary structures. The aforementioned tools require multiple alignments of RNA sequences to detect common secondary structures. As inputs of those tools, we can use RNA structural multiple alignment tools, which find the multiple alignments considering secondary structures, such as MAFFT (Kato and Toh, 2008), LocARNA (Will *et al.*, 2012), LARA (Bauer *et al.*, 2007), MXSCARNA (Tabei *et al.*, 2008), and CentroidAlign (Hamada *et al.*, 2009c).

Joint Secondary Structures Prediction

Functional RNAs interact with other molecules to perform their functions. Predictions of the joint secondary structures of two RNA molecules require predictions of inter-molecular and intra-molecular base pairs. RactIP (Kato *et al.*, 2010) and IntaRNA (Busch *et al.*, 2008) efficiently predicts joint secondary structures. PETcofold (Seemann *et al.*, 2011) predicts the conserved joint secondary structures.

Prediction of RNA Three Dimensional Structure

Accurate prediction of the 3D structures of RNA molecules is one of the most challenging problems in the RNA bioinformatics field. In the RNA-Puzzles (Cruz *et al.*, 2012; Miao *et al.*, 2015, 2017), a community-wide blind test for RNA 3D structure prediction, most participant groups could not predict near-native structures for several problems (target RNA molecules). RNA 3D structures consist of not only WC base-pairs but also non-WC base-pairs and various hydrogen bonds, in contrast to RNA 2D structures, which are simply constructed from WC base-pairs. Currently, base-pairing interactions are categorized into 12 groups based on the combination of interacting edges (Watson-Crick, Hoogsteen and sugar) in nucleotide bases and the relative orientations of glycosidic bonds (cis or trans) (Leontis *et al.*, 2002). Among these groups, cis WC-WC base-pair are known as standard WC base pairs, and the remaining 11 groups are non-WC base-pairs. Currently, there are two types of 3D structure prediction methods: template-based and physical chemistry-based methods.

In template-based methods, small structural elements derived from several known structures, referred to modules/motifs or templates, are assembled to construct a predicted structure as building blocks. These structural modules or motifs contain many non-WC base pairs. An appropriate selection of these structural modules leads to accurate prediction of the RNA 3D structures. Template sizes and template selection methods depend on the prediction software. As the largest template-based methods, ModeRNA (Rother *et al.*, 2011) and RNABuilder (Flores *et al.*, 2010) use the whole structures as the templates for their predictions. MC-fold/MC-sym (Parisien and Major, 2008), RNAComposer (Popenda *et al.*, 2012) and 3dRNA (Zhao *et al.*, 2012) employ medium-sized templates derived from small elements in RNA secondary structures that are computationally predicted or provided by users. FARNA (Das and Baker, 2007) is a fragment assembly method, which is also known as ROSETTA framework in protein

structure prediction fields, which uses a few nucleotides as fragments. The FARNAs-predicted structures could be refined at atomic resolution using FARFAR (Das *et al.*, 2010).

In physical chemistry-based methods, each nucleotide of an input RNA sequence is represented with coarse-grained beads instead of the full atoms in 3D space. Monte Carlo simulation or molecular dynamics simulation is used for conformational sampling of the coarse-grained RNA structure. There are two types of energy function for scoring the simulated (predicted) structures. The first type is knowledge-based statistical potential derived from known 3D structures as employed in SimRNA (Boniecki *et al.*, 2016). The second type of energy function is based on physicochemical interactions, such as hydrogen bonds of base pairing and electrostatic repulsion of the phosphates, implemented in HiRE-RNA (Cragnolini *et al.*, 2015). These methods could provide accurate predictions for small (10–20 nucleotides) to medium (50–80 nucleotides) of RNA molecules, whereas an application of these methods to moderately large (> 100 nucleotides) RNA molecules remains challenging. The physical chemistry-based methods require high computational cost for efficient conformational sampling of RNA structures.

Conclusion

The structures of RNAs are important for understanding their functions. Among them, RNA secondary structures are important and well studied. Most functional RNAs interact with other RNAs via complementary base pairing, which complete the intra-molecule interactions that form the secondary structures. The minimum free energy secondary structures are usually selected as the predicted structures, but the MEA estimator or gamma-centroid estimator have higher expected accuracy with respect to the number of correctly predicted base-pairs. Recent progress in experiments has allowed us to incorporate the experimentally predicted base pairs into computational predictions of the secondary structures. Even if using experimental data, the probabilities of the predicted secondary structures remain low because the number of possible secondary structures are large. The marginal probabilities, such as base-pairing probabilities, represent structurally important information, and they are not necessarily low. Predicting pseudoknotted structures, the interactions of two RNA molecules, the common secondary structures of two RNA sequences, are all computationally expensive if the strict DP algorithms are applied, but stochastic sampling or integer programming is often useful.

Computational methods have been developed for RNA secondary structures, and they are becoming useful for the structural analysis of RNA. Although the point estimations of RNA secondary structures have limited reliability, we can understand the total picture of the secondary structures via combinations of bioinformatics tools. The probability distribution of Hamming distance and the credibility limits are relevant examples. The prediction of the 3D structures remains challenging, but its accuracy is expected to improve due to the increased availability of experimentally validated 3D structures.

Acknowledgements

The authors also thank to the members of Artificial Intelligence Research Center, Hisanory Kiryu's laboratory, and Kiyoshi Asai's laboratory for useful discussions. This work was supported in part by JSPS KAKENHI Grant Numbers JP16H02484 and JP16H06279.

See also: Algorithms for Strings and Sequences: Pairwise Alignment. Algorithms for Strings and Sequences: Pairwise Alignment. Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Assessment of Structure Quality (RNA and Protein). Biomolecular Structures: Prediction, Identification and Analyses. Characterizing and Functional Assignment of Noncoding RNAs. Computational Design and Experimental Implementation of Synthetic Riboswitches and Riboregulators. Engineering of Supramolecular RNA Structures. Genome-Wide Probing of RNA Structure. MicroRNA and lncRNA Databases and Analysis. Natural Language Processing Approaches in Bioinformatics. Nucleic-Acid Structure Database. Predicting RNA-RNA Interactions in Three-Dimensional Structures. Prediction of Coding and Non-Coding RNA

References

- Andronescu, M., Condon, A., Hoos, H.H., Mathews, D.H., Murphy, K.P., 2007. Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics* 23 (13), 19–28.
- Aw, J.G., Shen, Y., Wilm, A., *et al.*, 2016. In vivo mapping of eukaryotic RNA interactomes reveals principles of higher-order organization and regulation. *Mol. Cell* 62 (4), 603–617.
- Bauer, M., Klau, G.W., Reinert, K., 2007. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinform.* 8, 271.
- Berkowitz, N.D., Silverman, I.M., Childress, D.M., *et al.*, 2016. A comprehensive database of high-throughput sequencing-based RNA secondary structure probing data (Structure Surfer). *BMC Bioinform.* 17 (1), 215.
- Bernhart, S.H., Hofacker, I.L., Will, S., Gruber, A.R., Stadler, P.F., 2008. RNAalifold: Improved consensus structure prediction for RNA alignments. *BMC Bioinform.* 9, 474.
- Boniecki, M.J., Lach, G., Dawson, W.K., *et al.*, 2016. SimRNA: A coarse-grained method for RNA folding simulations and 3D structure prediction. *Nucleic Acids Res.* 44 (7), e63.
- Busch, A., Richter, A.S., Backofen, R., 2008. IntaRNA: Efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics* 24 (24), 2849–2856.

- Cragolin, T., Laurin, Y., Derreux, P., Pasquali, S., 2015. Coarse-grained HiRE-RNA model for ab Initio RNA folding beyond simple molecules, including noncanonical and multiple base pairings. *J. Chem. Theory Comput.* 11 (7), 3510–3522.
- Cruz, J.A., Blanchet, M.F., Boniecki, M., *et al.*, 2012. RNA-Puzzles: A CASP-like evaluation of RNA three-dimensional structure prediction. *RNA* 18 (4), 610–625.
- Das, R., Baker, D., 2007. Automated de novo prediction of native-like RNA tertiary structures. *Proc. Natl. Acad. Sci. USA* 104 (37), 14664–14669.
- Das, R., Karanicolas, J., Baker, D., 2010. Atomic accuracy in predicting and designing noncanonical RNA structure. *Nat. Methods* 7 (4), 291–294.
- Deigan, K.E., Li, T.W., Mathews, D.H., Weeks, K.M., 2009. Accurate SHAPE-directed RNA structure determination. *Proc. Natl. Acad. Sci. USA* 106 (1), 97–102.
- Ding, Y., Kwok, C.K., Tang, Y., Bevilacqua, P.C., Assmann, S.M., 2015. Genome-wide profiling of in vivo RNA structure at single-nucleotide resolution using structure-seq. *Nat. Protoc.* 10 (7), 1050–1066.
- Ding, Y., Lawrence, C.E., 2003. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.* 31 (24), 7280–7301.
- Do, C.B., Woods, D.A., Batzoglou, S., 2006. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics* 22 (14), e90–98.
- Flores, S.C., Wan, Y., Russell, R., Altman, R.B., 2010. Predicting RNA structure by multiple template homology modeling. *Pac. Symp. Biocomput.* 216–227.
- Flynn, R.A., Zhang, Q.C., Spitale, R.C., *et al.*, 2016. Transcriptome-wide interrogation of RNA secondary structure in living cells with icSHAPE. *Nat. Protoc.* 11 (2), 273–290.
- Freyhult, E., Moulton, V., Clote, P., 2007. Boltzmann probability of RNA structural neighbors and riboswitch detection. *Bioinformatics* 23 (16), 2054–2062.
- Fukunaga, T., Ozaki, H., Terai, G., *et al.*, 2014. CapR: Revealing structural specificities of RNA-binding protein target recognition using CLIP-seq data. *Genome Biol.* 15 (1), R16.
- Hamada, M., Kiryu, H., Sato, K., Mituyama, T., Asai, K., 2009a. Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics* 25 (4), 465–473.
- Hamada, M., Sato, K., Asai, K., 2011. Improving the accuracy of predicting secondary structure for aligned RNA sequences. *Nucleic Acids Res.* 39 (2), 393–402.
- Hamada, M., Sato, K., Kiryu, H., Mituyama, T., Asai, K., 2009b. Predictions of RNA secondary structure by combining homologous sequence information. *Bioinformatics* 25 (12), i330–i338.
- Hamada, M., Sato, K., Kiryu, H., Mituyama, T., Asai, K., 2009c. CentroidAlign: Fast and accurate aligner for structured RNAs by maximizing expected sum-of-pairs score. *Bioinformatics* 25 (24), 3236–3243.
- Harmanci, A.O., Sharma, G., Mathews, D.H., 2011. TurboFold: Iterative probabilistic estimation of secondary structures for multiple RNA sequences. *BMC Bioinform.* 12, 108.
- Incarlato, D., Neri, F., Anselmi, F., Oliviero, S., 2014. Genome-wide profiling of mouse RNA secondary structures reveals key features of the mammalian transcriptome. *Genome Biol.* 15 (10), 491.
- Kasami, T., 1965. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report R257*.
- Katoh, K., Toh, H., 2008. Improved accuracy of multiple ncRNA alignment by incorporating structural information into a MAFFT-based framework. *BMC Bioinform.* 9, 212.
- Kato, Y., Sato, K., Hamada, M., *et al.*, 2010. RactIP: Fast and accurate prediction of RNA-RNA interaction using integer programming. *Bioinformatics* 26 (18), i460–466.
- Kawaguchi, R., Kiryu, H., 2016. Parallel computation of genome-scale RNA secondary structure to detect structural constraints on human genome. *BMC Bioinform.* 17 (1), 203.
- Kertesz, M., Wan, Y., Mazon, E., *et al.*, 2010. Genome-wide measurement of RNA secondary structure in yeast. *Nature* 467 (7311), 103–107.
- Kiryu, H., Kin, T., Asai, K., 2008. Rfold: An exact algorithm for computing local base pairing probabilities. *Bioinformatics* 24 (3), 367–373.
- Knudsen, B., Hein, J., 2003. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.* 31 (13), 3423–3428.
- Leontis, N.B., Stombaugh, J., Westhof, E., 2002. The non-Watson-Crick base pairs and their associated isostericity matrices. *Nucleic Acids Res.* 30 (16), 3497–3531.
- Lorenz, R., Bernhart, S.H., Honer Zu Siederdissen, C., *et al.*, 2011. ViennaRNA Package 2.0. *Algorithms Mol. Biol.* 6, 26.
- Lorenz, R., Luntzer, D., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T., 2016. SHAPE directed RNA folding. *Bioinformatics* 32 (1), 145–147.
- Loughrey, D., Watters, K.E., Settle, A.H., Lucks, J.B., 2014. SHAPE-Seq 2.0: Systematic optimization and extension of high-throughput chemical probing of RNA secondary structure with next generation sequencing. *Nucleic Acids Res.* 42 (21), doi: 10.1093/nar/gku909.
- Lucks, J.B., Mortimer, S.A., Trapnell, C., *et al.*, 2011. Multiplexed RNA structure characterization with selective 2'-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). *Proc. Natl. Acad. Sci. USA* 108 (27), 11063–11068.
- Lu, Z., Zhang, Q.C., Lee, B., *et al.*, 2016. RNA duplex map in living cells reveals higher-order transcriptome structure. *Cell* 165 (5), 1267–1279.
- Mathews, D.H., Sabina, J., Zuker, M., Turner, D.H., 1999. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* 288 (5), 911–940.
- Mathews, D.H., 2004. Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA* 10 (8), 1178–1190.
- Mathews, D.H., 2014. Using the RNAstructure software package to predict conserved RNA structures. *Curr. Protoc. Bioinform.* 46, 1–22.
- McCaskill, J.S., 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* 29 (6–7), 1105–1119.
- Miao, Z., Adamiak, R.W., Antczak, M., *et al.*, 2017. RNA-Puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA* 23 (5), 655–672.
- Miao, Z., Adamiak, R.W., Blanchet, M.F., *et al.*, 2015. RNA-Puzzles Round II: Assessment of RNA structure prediction programs applied to three large RNA structures. *RNA* 21 (6), 1066–1084.
- Mori, R., Hamada, M., Asai, K., 2014. Efficient calculation of exact probability distributions of integer features on RNA secondary structures. *BMC Genomics* 15 (Suppl 10), S6.
- Newberg, L.A., Lawrence, C.E., 2009. Exact calculation of distributions on integers, with application to sequence alignment. *J. Comput. Biol.* 16 (1), 1–18.
- Nguyen, T.C., Cao, X., Yu, P., *et al.*, 2016. Mapping RNA-RNA interactome and RNA structure in vivo by MARIO. *Nat. Commun.* 7, 12023.
- Norris, M., Kwok, C.K., Cheema, J., *et al.*, 2017. FoldAtlas: A repository for genome-wide RNA structure probing data. *Bioinformatics* 33 (2), 306–308.
- Nussinov, R., Pieczek, G., Griggs, J., Kleitman, D., 1978. Algorithms for Loop Matchings. *SIAM J. Appl. Math.* 35 (1), 68–82.
- Ouyang, Z., Snyder, M.P., Chang, H.Y., 2013. SeqFold: Genome-scale reconstruction of RNA secondary structure integrating high-throughput sequencing data. *Genome Res.* 23 (2), 377–387.
- Parisien, M., Major, F., 2008. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature* 452 (7183), 51–55.
- Popenda, M., Szachniuk, M., Antczak, M., *et al.*, 2012. Automated 3D structure composition for large RNAs. *Nucleic Acids Res.* 40 (14), e112.
- Ramani, V., Qiu, R., Shendure, J., 2015. High-throughput determination of RNA structure by proximity ligation. *Nat. Biotechnol.* 33 (9), 980–984.
- Rivas, E., Eddy, S.R., 2000. The language of RNA: A formal grammar that includes pseudoknots. *Bioinformatics* 16 (4), 334–340.
- Rivas, E., Eddy, S.R., 1999. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* 285 (5), 2053–2068.
- Rivas, E., Lang, R., Eddy, S.R., 2012. A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA* 18 (2), 193–212.
- Rother, M., Rother, K., Puton, T., Bujnicki, J.M., 2011. ModeRNA: A tool for comparative modeling of RNA 3D structure. *Nucleic Acids Res.* 39 (10), 4007–4022.
- Rouskin, S., Zubradt, M., Washietl, S., Kellis, M., Weissman, J.S., 2014. Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. *Nature* 505 (7485), 701–705.
- Sakakibara, Y., Brown, M., Hughey, R., *et al.*, 1994. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res.* 22 (23), 5112–5120.
- Sakuraba, S., Asai, K., Kameda, T., 2015. Predicting RNA duplex dimerization free-energy changes upon mutations using molecular dynamics simulations. *J. Phys. Chem. Lett.* 6 (21), 4348–4351.
- Sato, K., Hamada, M., Asai, K., Mituyama, T., 2009. CENTROIDFOLD: A web server for RNA secondary structure prediction. *Nucleic Acids Res.* 37 (Web Server issue), W277–W280.
- Sato, K., Kato, Y., Hamada, M., Akutsu, T., Asai, K., 2011. IPknot: Fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics* 27 (13), 85–93.

- Seemann, S.E., Menzel, P., Backofen, R., Gorodkin, J., 2011. The PETfold and PETcofold web servers for intra- and intermolecular structures of multiple RNA sequences. *Nucleic Acids Res.* 39 (Web Server issue), W107–111.
- Sharma, E., Sterne-Weiler, T., O'Hanlon, D., Blencowe, B.J., 2016. Global mapping of human RNA-RNA interactions. *Mol. Cell* 62 (4), 618–626.
- Siegfried, N.A., Busan, S., Rice, G.M., Nelson, J.A., Weeks, K.M., 2014. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nat. Methods* 11 (9), 959–965.
- Smola, M.J., Rice, G.M., Busan, S., Siegfried, N.A., Weeks, K.M., 2015. Selective 2'-hydroxyl acylation analyzed by primer extension and mutational profiling (SHAPE-MaP) for direct, versatile and accurate RNA structure analysis. *Nat. Protoc.* 10 (11), 1643–1669.
- Spitale, R.C., Flynn, R.A., Zhang, Q.C., *et al.*, 2015. Structural imprints in vivo decode RNA regulatory mechanisms. *Nature* 519 (7544), 486–490.
- Sukosd, Z., Knudsen, B., Kjems, J., Pedersen, C.N., 2012. PPfold 3.0: Fast RNA secondary structure prediction using phylogeny and auxiliary data. *Bioinformatics* 28 (20), 2691–2692.
- Tabei, Y., Kiryu, H., Kin, T., Asai, K., 2008. A fast structural multiple alignment method for long RNA sequences. *BMC Bioinform.* 9, 33.
- Talkish, J., May, G., Lin, Y., Woolford, J.L., McManus, C.J., 2014. Mod-seq: High-throughput sequencing for chemical probing of RNA structure. *RNA* 20 (5), 713–720.
- Underwood, J.G., Uzilov, A.V., Katzman, S., *et al.*, 2010. FragSeq: Transcriptome-wide RNA structure probing using high-throughput sequencing. *Nat. Methods* 7 (12), 995–1001.
- Wan, Y., Qu, K., Ouyang, Z., *et al.*, 2012. Genome-wide measurement of RNA folding energies. *Mol. Cell* 48 (2), 169–181.
- Wan, Y., Qu, K., Zhang, Q.C., *et al.*, 2014. Landscape and variation of RNA secondary structure across the human transcriptome. *Nature* 505 (7485), 706–709.
- Washietl, S., Hofacker, I.L., Stadler, P.F., Kellis, M., 2012. RNA folding with soft constraints: Reconciliation of probing data and thermodynamic secondary structure prediction. *Nucleic Acids Res.* 40 (10), 4261–4272.
- Will, S., Joshi, T., Hofacker, I.L., Stadler, P.F., Backofen, R., 2012. LocARNA-P: Accurate boundary prediction and improved detection of structural RNAs. *RNA* 18 (5), 900–914.
- Younger, D., 1967. Recognition and parsing of context-free languages in time n^3 . *Inf. Control* 10 (2), 189–208.
- Zakov, S., Goldberg, Y., Elhadad, M., Ziv-Ukelson, M., 2011. Rich parameterization improves RNA structure prediction. *J. Comput. Biol.* 18 (11), 1525–1542.
- Zarringhalam, K., Meyer, M.M., Dotu, I., Chuang, J.H., Clote, P., 2012. Integrating chemical footprinting data into RNA secondary structure prediction. *PLOS ONE* 7 (10), e45160.
- Zhao, Y., Huang, Y., Gong, Z., *et al.*, 2012. Automated and fast building of three-dimensional RNA structures. *Sci. Rep.* 2, 734.
- Zubradt, M., Gupta, P., Persad, S., *et al.*, 2017. DMS-MaPseq for genome-wide or targeted RNA structure probing in vivo. *Nat. Methods* 14 (1), 75–82.
- Zuker, M., 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31 (13), 3406–3415.
- Zuker, M., Stiegler, P., 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* 9 (1), 133–148.

Rational Structure-Based Drug Design

Varun Khanna, Flinders University, Adelaide, SA, Australia and Vaxine Pty Ltd., Adelaide, SA, Australia

Shoba Ranganathan, Macquarie University, Sydney, NSW, Australia

Nikolai Petrovsky, Flinders University, Adelaide, SA, Australia and Vaxine Pty Ltd., Adelaide, SA, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The discovery, optimization and evaluation of small molecules which interact with a biological target of pharmaceutical relevance is the core of the drug discovery process. So far, this has been mostly dominated by the conventional brute-force approach of lead discovery via high throughput screening (HTS) of the natural product, synthetic or combinatorial compound libraries to identify potential leads (Macarron *et al.*, 2011). While effective in many cases, HTS is expensive, requires a readily available diverse set of compounds to screen, often yields hits with poor efficacy and provides little information to guide further optimization. This has driven the development of more rational and cost-efficient screening methods. The advancement in molecular biology, crystallographic techniques and computational power over the last 30 years have provided researchers access to high-quality structural information on a wide variety of biological targets. Structural information is the ultimate rational drug design tool which can streamline all aspects of drug discovery from target selection to lead optimization and can significantly reduce the development cost. Indeed, structure-guided drug design efforts have led to the discovery of many high profile drugs including amprenavir (Agenerase) and nelfinavir (Viracept) developed against the HIV protease (Kaldor *et al.*, 1997), zanamivir (Relenza) against neuraminidase (Varghese, 1999), imatinib mesylate (Gleevec) which inhibits Abl tyrosine kinase (Schindler *et al.*, 2000), erlotinib (Tarceva) for the treatment of metastatic lung cancer (Pollack *et al.*, 1999) and the thrombin inhibitor ximelagatran (Exanta) as an oral anticoagulant (Gustafsson *et al.*, 2001). More recently, lapatinib, an ErbB2 inhibitor for cancer, sitagliptin (Januvia) for type-2 diabetes, vorinostat (Zolinza) to inhibit histone deacetylase, and rivaroxaban, as an oral Factor Xa inhibitor are examples of successful application of rational structure-based methods in drug development.

Structure-based drug design (SBDD) is the process in which novel drug-like molecules are designed based on structural knowledge of the relevant macromolecular target (Wang *et al.*, 2016). In SBDD, virtual chemical libraries containing millions of compounds and structural data of macromolecules are used in tandem to assess the ability of compounds to interact with the target of interest. Many virtual libraries are publicly available, a few notable examples being PubChem (Kim *et al.*, 2016), ChEMBL (Bento *et al.*, 2014), BindingDB (Gilson *et al.*, 2016), DrugBank (Law *et al.*, 2014) and ZINC (Irwin *et al.*, 2012). In addition to public sources, commercial vendors (Maybridge, Enamine) maintain virtual libraries. Structural information on biological targets can be obtained from Protein Data Bank (PDB) (Parasuraman, 2012). If the structure of the desired target is not available, it may be possible to create a homology model using related structures available in PDB. In either case, the compounds in the virtual library are subsequently docked into the binding cavity of the target to determine the relative rank for the entire dataset. Automatic and manual data analysis of the docking result is then carried out to predict the compounds with highest binding affinity to the target. This results in a smaller subset of compounds as a starting point for downstream analysis.

This article summarizes the process of structure-based drug design and discusses the choice of target, lead identification and the various docking-based methods. Key concepts in SBDD will be illustrated through a case study that explores the identification of potential novel anti-malarial agents targeting the hemoglobin degrading enzyme, Plasmeprin II, of *Plasmodium falciparum*. The aim of this article is to guide novice computational users by elucidating the general steps involved in such a drug discovery exercise.

Background and Overview of the Process

The process of drug discovery and development generally requires extensive examination of target proteins and potential drug-like compounds before a lead is ready for phase 1 clinical trials. The first stage is identification, purification and structure determination of the target (usually a protein). The selection of the target is the most crucial step as it sets the course for all future aspects of research in the drug discovery pipeline. Once the biological target has been selected, its structure is determined by one of three methods: *X-ray diffraction* generally known as *X-ray crystallography*, *nuclear magnetic resonance spectroscopy* generally known as *NMR* and *comparative modeling* generally known as *in silico homology modeling*. In the second stage, computational techniques are used to screen and rank compound or fragment libraries based on their electrostatic and steric interaction with the target. Promising hits are then tested in biological assays to identify the compounds with desired activity (lead discovery). Once a candidate series has been identified, the lead optimization stage begins. In this stage, key lead compound properties like adsorption, metabolism, distribution and excretion are optimized while avoiding risk of toxicity. Further steps include synthesis of the optimized leads, testing, determination of the target lead complex and additional optimization. After several rounds of the process, optimized compounds usually show marked improvement in these pharmacokinetic properties and specificity for the target. The lead optimization stage usually concludes with the successful demonstration of *in vivo* efficacy of the compound in an appropriate animal model.

Rational Structure-Based Drug Design

Choice of the Drug Target and Structure Determination

The choice of the target protein is primarily made based on therapeutic and biological relevance. The 'druggable' target is a protein, peptide or nucleic acid whose activity can be modulated by a small molecule (Owens, 2007). Proteins are often used as drug targets due to their significant role in metabolic or signaling pathways specific to a disease condition and can either be activated or inhibited by small molecule drugs. Once the target has been identified it is essential to determine its three-dimensional (3D) structure ideally with a well-defined drug binding pocket to enable high throughput virtual screening in the SBDD approach. The structure of the target can be determined by one of the following methods:

X-ray crystallography and NMR

Both X-ray crystallography and NMR produce data on the relative position of atoms of a molecule. The basis of X-ray crystallography is the scattering of X-rays from electron clouds of atoms whereas NMR measures the interaction of atomic nuclei. The end-product of crystallographic structure determination is an electron density map which is essentially a contour plot indicating positions in the crystal structure where electrons are most likely to be found. This data must be interpreted in terms of a 3D model using semi-automatic computational methodologies. On the other hand, the end-product of an NMR experiment is usually a set of distances between atomic nuclei that define both bonded and non-bonded close contacts in a molecule. These must be interpreted manually to produce a 3D molecular structure using computational tools. The structure determination in each case requires assumptions and approximations, hence the resulting molecular structures obtained may have errors. The choice of technique depends on many factors including molecular weight, ease of solubility and crystallization of the macromolecule under study. X-ray crystallography remains the main workhorse of structure determination for SBDD. Currently, the RCSB Protein Data bank (Parasuraman, 2012) database contains over 130,000 structures of which 90% were solved through X-ray crystallography (Fig. 1).

Homology modeling

In the absence of an experimentally determined 3D structure of a protein, *in silico* homology modeling can provide structural models that are comparable to the best results achieved experimentally. In general, 30% target-template sequence identity is required to generate a useful structural model (Forrest *et al.*, 2006). This allows researchers to use the generated *in silico* models for

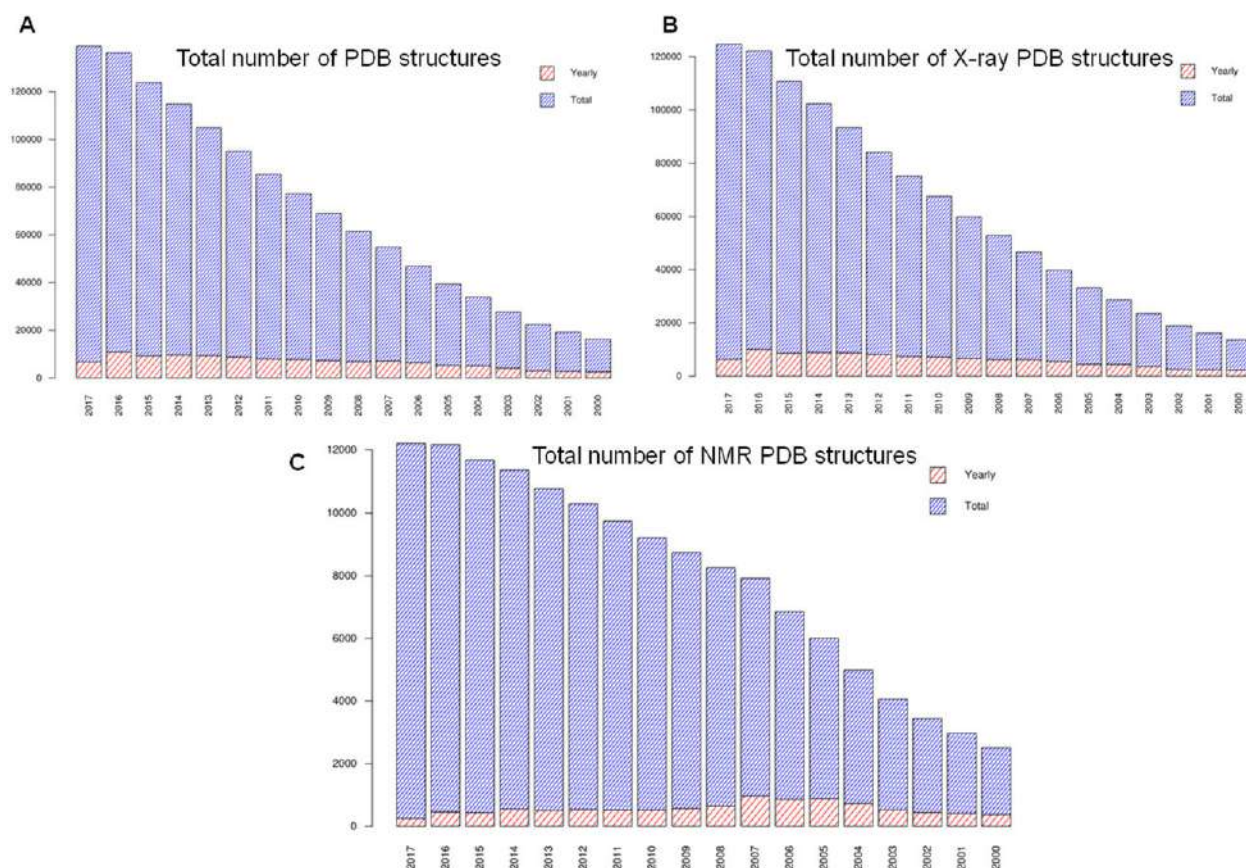


Fig. 1 Incremental increase in the structures of macromolecules in PDB database since year 2000. (Data obtained from PDB).

functional analysis and to predict interactions with other molecules. Homology modeling predicts the structure of the target protein primarily by aligning the target sequence called a query with the sequence of one or more known structures called templates and is based on two major assumptions:

1. The structure of a protein is encoded in its sequence thus knowing the sequence should at least, in theory, suffice to obtain the structure. This observation was elegantly demonstrated by Anfinsen when he showed that bovine pancreatic ribonuclease, following exposure to a denaturant, could spontaneously regain its native folded structure (Anfinsen, 1973).
2. The structure of a protein is more stable and conserved than its sequence. Therefore, closely related similar sequences will essentially adopt similar structures and more distantly related sequences will at least have similar folding, a relationship first identified by Chothia and Lesk (1986).

In practice homology modeling is a multistep process which can be summarized in the following seven steps: a). Template identification, b). Alignment correction, c). Model building d). Loop modeling, e). Side-chain modeling f). Model optimization and g). Model validation. A comprehensive review of homology modeling is beyond the scope of this article. However, interested readers are referred to a series of publications and reviews on homology modeling (Eswar *et al.*, 2006; Fiser and Sali, 2003; Martí-Renom *et al.*, 2000). A list of tools for homology modeling are given in Table 1. Repositories such as Protein model portal (Arnold *et al.*, 2009), Modbase (Pieper *et al.*, 2011) and SWISS-MODEL (Kiefer *et al.*, 2009) contain proteins models generated using various methods.

Identification of Binding Site

Once the 3D structure of the target protein is generated, its potential ligand binding sites are determined to facilitate docking computation and virtual screening (VS). Typically, the ligand binding site is a pocket with a variety of potential hydrogen bond donors, acceptors, hydrophobic characteristics and a well-defined stereochemical arrangement. The ligand binding site can be the active site where the substrate binds or can be an assembly site where another macromolecule binds or a communication site necessary to relay the information. Although empirical data suggest that actual ligand binding sites tend to coincide with the largest and deepest pockets on the target surface (Weisel *et al.*, 2007), there exist cases where ligands are known to bind exposed and shallow clefts (Nisius *et al.*, 2012). Due to the importance of binding sites in molecular recognition and interaction, a large number of computational methods for binding site recognition and analysis have been developed that scan the surface of the target for cavities or pockets which might be potential binding sites. All these methods take a target structure as input and then output an ordered list of putative binding sites. Generally, not all reported sites correspond to true binding sites, however, it is expected that entries at the top will correspond to regions with a higher probability of being true binding sites. The methods can be broadly divided into sequence-, structure- and energy-based methods.

Table 1 Tools for homology modeling

Name	URL	Summary	Reference
RaptorX	http://raptorx.uchicago.edu/	Predict secondary, tertiary, solvent accessibility, disordered regions and binding sites. Remote homology detection	Källberg <i>et al.</i> (2012)
Biskit	http://biskit.pasteur.fr/	Python library for typical tasks of structural bioinformatics including homology modeling, docking and dynamics	Grünberg <i>et al.</i> (2007)
Phyre2	http://www.sbg.bio.ic.ac.uk/~phyre2	Uses HMM to predict, analyze and build protein 3D structure and binding site. Analyzes the effect of SNPs.	Kelley <i>et al.</i> (2015)
EsyPred3D	http://www.unamur.be/sciences/biologie/urbm/bioinfo/esypred/	Automated webserver based on MODELLER. Uses neural networks to improve the alignment	Lambert <i>et al.</i> (2002)
Modeller	https://salilab.org/modeller/	Predicts three-dimensional structure by satisfying spatial restraints also performs multiple sequence alignments, clustering, de novo modeling of loops and optimization of the models	Eswar <i>et al.</i> (2006)
Robetta	http://robeta.bakerlab.org/	Robetta parses proteins chains into putative domains and model these domains either by homology modeling or by ab initio modeling	Kim <i>et al.</i> (2004)
I-TASSER	https://zhanglab.ccmb.med.umich.edu/I-TASSER/	I-TASSER generated 3D models from multiple threading alignments. The function of the protein is inferred by structural matching the model to protein function database	Roy <i>et al.</i> (2010)
Bhageerath-H	http://www.scfbio-iitd.res.in/bhageerath/bhageerath_h.jsp	Hybrid of ab initio folding and homology methods	Jayaram <i>et al.</i> (2014)

Sequence-based methods

Sequence-based methods are based on evolutionary conservation and exploit the propensity of conserved residues in the binding site. In the LIGSITE_{Esc} algorithm (Huang and Schroeder, 2006) a sequence conservation measure of neighboring residues is used to re-rank the top three sites predicted by LIGSITE (explained under structure-based methods), which leads to an improved success rate in binding site prediction. Unlike LIGSITE_{Esc}, in ConCavity (Capra *et al.*, 2009) the conservation information is not only used to re-rank the predictions but is also incorporated into the binding site detection procedure.

Structure-based methods

In contrast to sequence-based methods, structure-based method predict ligand binding site by analyzing geometrical features like clefts or cavities. Some methods are solely based on geometrical features (LIGSITE (Hendlich *et al.*, 1997), PocketPicker (Weisel *et al.*, 2007)) while others take into account additional features like physicochemical information, polarity or charge (Fpocket (Le Guilloux *et al.*, 2009), SiteFinder by MOE (Locating Binding Sites in Protein Structures, 2017)).

Energy-based methods

Energy-based methods depend on the calculation of interaction energy between various probes placed on the grid points around the target surface and subsequent clustering of the probes with low interaction energy to identify the most energetically favorable binding pockets. Notable examples of energy-based methods are Q-SiteFinder (Laurie and Jackson, 2005) and SiteHound (Ghera and Sanchez, 2009).

Consensus method

A consensus method is essentially a meta approach that combines results from several algorithms mentioned above. For example, Metapocket 2.0 (Huang, 2009) collects results of eight different methods by taking the top three sites from each method with the authors demonstrating that Metapocket 2.0 performs better than any one individual method alone.

For a detailed review of binding site prediction methods, readers can refer to Henrich *et al.* (2010), Nisius *et al.* (2012), Xie and Hwang (2015).

Molecular Docking Based Virtual Screening

Once the structure and the target site are determined, molecular docking based virtual screening of compound and fragment libraries can be carried out to identify potential leads. VS is regarded as a computational counterpart of the experimental high-throughput screening method and is one of the most widely used strategies for SBDD (Stahura and Bajorath, 2004). The major advantage of docking approaches is their speed and the guidance they provide to follow-on wet lab experiments (López-Vallejo *et al.*, 2011). Molecular docking is also useful in the study of ligand-target interactions as the docking programs can analyze the microscopic atomic interactions between a ligand and a target.

In VS, typically a database of small molecules is docked onto the region of interest and is scored based on the predicted interactions with the site. However, in the fragment based approach, small drug-like fragments such as benzene rings, amino, hydroxy, carbonyl or other functional groups can be positioned, scored, optimized in the binding site and finally linked *in silico* to generate lead candidates. The compounds obtained *in silico* by linking fragments can then be synthesized and tested. In general, molecular docking based VS consists of several steps; (i) target preparation, (ii) compound or fragment database selection, (iii) molecular docking, (iv) post-docking evaluation analysis, and (v) molecular dynamics and binding free energy calculation. Careful and thorough literature survey regarding the target, binding site and known ligands (if available) is often necessary to select the docking algorithms best suited for the given target.

Target preparation

The preparation of the target protein requires great care because experimentally determined structures frequently have many problems like incomplete side chains, missing loops and result in inaccurate models due to incorrect interpretation of the experimental data. Structural characterization of the targeted protein includes a choice of tautomeric forms of histidine residues, correct assignment of protonation states of amino acids and conformation of residues, especially in the binding site. It is also important to add missing hydrogen atoms, build missing residues or loops, identify overlapping atoms to reduce clash and optimize hydrogen bonding network. Water molecules and cofactors (metal) in the protein active site may need to be removed or retained (if they are known to be critical for ligand binding). Metals and cofactors which form an integral part of the binding interaction with the ligand are considered part of the docking site and hence are retained. If the software allows for flexible docking in order to account for conformational changes during ligand binding, the number, the identity of flexible residues and degrees of flexibility needs to be defined.

Compound library preparation

Prior to generating the 3D structure of the library of ligands, it is important to clean up the 2D structures by removing the ions, salts, incomplete structures, inorganic molecules, duplicates and water molecules. All reactive, promiscuous and otherwise undesirable compounds should also be removed. Additionally, compounds may be filtered based on several physicochemical

Table 2 List of publically available databases of chemical compounds

Database	No. of compounds	Description	URL	Reference
PubChem	~ 120 million	Contains compounds and associated bioassays	https://pubchem.ncbi.nlm.nih.gov/	Kim <i>et al.</i> (2016)
ZINC	~ 35 million	Ready to dock, purchasable compounds	http://zinc.docking.org/	Irwin <i>et al.</i> (2012)
ChemSpider	~ 25 million	Data linked to 300 diverse sources	http://www.chemspider.com/	Williams (2010)
ChEMBL	~ 2 million	Bioactive compounds with drug target information	https://www.ebi.ac.uk/chembl/	Bento <i>et al.</i> (2014)
CCDC	~ 900,000	Crystal structures of small molecules	https://www.ccdc.cam.ac.uk/	Groom <i>et al.</i> (2016)
BindingDB	621,060	Binding affinities of drug-targets	www.bindingdb.org/bind/index.jsp	Gilson <i>et al.</i> (2016)
NCI	265,242	Anticancer and anti-HIV screening data	https://cactus.nci.nih.gov/download/nci/index.html	Ihlenfeldt <i>et al.</i> (2002)
HMDB	74,507	Metabolites in human body	http://www.hmdb.ca/	Wishart <i>et al.</i> (2013)
DrugBank	9,591	Drugs and drug like compounds	https://www.drugbank.ca/	Law <i>et al.</i> (2014)

parameters such as molecular weight, number of hydrogen bond acceptors, number of hydrogen bond donors, number of rotatable bonds and log P to increase the bioavailability of the remaining compounds (Doak *et al.*, 2014; Lipinski, 2000). Ligands should also be checked for proper valence states and geometry, including reasonable bond distances and angles. Energy minimization and protonation of ligands should also be performed to achieve correct ligand-protein binding (ten Brink and Exner, 2009). Table 2 list the major organizations which maintain the public repositories of chemical compounds from where 2D and 3D structures can be obtained for molecular docking.

Molecular docking

Molecular docking is a method which predicts the preferred relative orientation of one molecule (key) when bound in an active site of another molecule (lock) to form a stable complex such that free energy of the overall system is minimized. It exploits the concept of molecular shape and physicochemical complementarity. The structures interact like a hand in a glove, where both shape and physicochemical properties contribute to the fit. Molecular docking process can be separated into two major steps i) searching and ii) scoring.

Search algorithm

The search algorithm implemented in any docking tool should enumerate an optimum number of ways two molecules can be put together. The size of the search space grows exponentially with the increase in the size of the molecules. For example, the number of possible conformations for a small molecule with 10 rotatable bonds with 30 degrees of increments are 10^{12} . If the target protein is also allowed to be flexible, this quickly becomes an intractable problem. Docking applications frequently employ one or more of the following search algorithms; Simulated Annealing, Fast shape matching, Incremental construction, Particle Swarm optimization and Evolutionary algorithms (Heberlé and de Azevedo, 2011).

Scoring functions

The scoring function measures and ranks the binding of a ligand-receptor complex. It is therefore not only important to have an efficient, accurate scoring function that gives the best rank to true bound structures, but it must give the correct relative rank of each ligand in the database. Ideally, the score should directly correspond to the binding affinity of the ligand for the protein so that the top scoring compounds are also the best binders. There are three main types of scoring functions that are used by various docking programs; 1) Force field based – GLODscore, DOCK, AutoDOCK derived from AMBER and CHARMM; 2) Empirical scoring – PLP, CHemscore, Glide SP/XP and PLANTSchemplp and PLANTSplp and 3) Knowledge-based – PMF, DrugScore and its derivatives DrugScoreCSD, Astex statistical potential.

For a detailed review of molecular docking, the reader can refer to the following publications (Halperin *et al.*, 2002; Meng *et al.*, 2011; Yuriev and Ramsland, 2013). A substantial number of docking tools are now available which have been applied for discovery of novel bioactive molecules. Table 3 summarizes basic features, license terms, algorithms and scoring functions of currently available docking software. The list is not exhaustive as many docking tools have become available in recent years. Selecting a particular docking tool is always a challenge. In one study it was reported that AutoDock offered a combination of accuracy and speed as opposed to other methods for prediction of protein kinase inhibitors (Buzko *et al.*, 2002). Kellenberger *et al.* (2004) when compared the performance of eight docking programs to recover the X-ray poses of 100 ligands, they reported that GLIDE, GOLD and SURFLEX had the best docking accuracy. Therefore, choosing a docking program largely depend upon the protein family under consideration.

Table 3 Currently available docking tools

Docking tool	License terms	Short description	URL	Reference
Autodock	Freeware	Genetic algorithm, Lamarckian genetic algorithm and Simulated annealing	http://autodock.scripps.edu/	Forli <i>et al.</i> (2016)
PatchDock	Freeware	Rigid docking	https://bioinfo3d.cs.tau.ac.il/PatchDock/	Schneidman-Duhovny <i>et al.</i> (2005)
GEMDOCK	Freeware	Generic evolutionary method for docking	http://gemdock.life.nctu.edu.tw/dock/	Yang and Chen (2004)
Autodock VINA	Open source	New generation of Autodock	http://vina.scripps.edu/manual.html	Trott and Olson (2010)
rDOCK	Open source	For HTVS of small molecules against proteins and nucleic acids	http://rdock.sourceforge.net/	Ruiz-Carmona <i>et al.</i> (2014)
PLANTS	Free for academic use	Stochastic optimization algorithms	http://www.uni-tuebingen.de/	Korb <i>et al.</i> (2009)
DOCK	Free for academic use	Geometric matching algorithm, docks either small molecules or fragments and include solvent effect	http://dock.compbio.ucsf.edu/	Allen <i>et al.</i> (2015)
FRED	Free for academic	Exhaustive examination of all possible poses of small molecules	https://www.eyesopen.com/oedocking	McGann (2011)
HADDOCK	Free for academic	Protein-protein docking	http://www.bonvinlab.org/software/haddock2.2/	Dominguez <i>et al.</i> (2003)
ICM	Commercial	Pseudo brownian sampling and Monte Carlo minimization for protein ligand docking	https://www.molsoft.com/docking.html	Neves <i>et al.</i> (2012)
GLIDE	Commercial	Exhaustive search and two different scoring function to rank-order compounds	https://www.schrodinger.com/glide	Repasky <i>et al.</i> (2007)
GOLD	Commercial	Genetic algorithm, flexible ligand, partial flexibility of proteins	https://www.ccdc.cam.ac.uk/solutions/csd-discovery/components/gold/	Verdonk <i>et al.</i> (2003)
FlexX	Commercial	Incremental construction algorithm for protein ligand docking	https://www.biosolveit.de/FlexX/	Kramer <i>et al.</i> (1999)

The individual docking procedure associated with different docking tools may differ, for example, formats of the ligands and target files, scoring functions and algorithms for ligand placement or the ligand may be docked in entirety or in fragments. However, the general principles of docking remain similar; compounds are first placed inside the binding pocket using the algorithm implemented by the docking software and then evaluated for non-covalent interactions using a scoring algorithm available in the same package. If the position of the ligand is known *a priori*, this information can be exploited during and after the docking process. Some programs such as DOCK allow an anchor fragment, ligand placement can thus be guided during the docking process.

Post-docking evaluation and analysis

The output of docking and scoring is a ranked list of predicted bound ligands. Using computer graphics software, these can be visually evaluated for goodness of fit, the formation of key hydrogen bonds, electrostatic interactions, surface complementarities and stability of the bound conformation compared to free conformation. Compounds that get selected in the initial run are then subjected to further *in silico* optimization using focused libraries of potential ligands. Furthermore, the 3D structure of the complex comprising the target protein and the compound is often determined experimentally in order to validate the binding mode predicted by the docking software. Eventually, several high scoring ligands are purchased or synthesized and evaluated for binding and biological activity in the wet lab.

Molecular dynamics and binding free energy calculation

Molecular dynamics (MD) is a technique where the time evolution of a set of interacting atoms is followed by numerical integration of Newton's equation of motion (Adcock and McCammon, 2006). The field of MD simulations is rapidly progressing with improvement in simulation methodology and increasing accuracy of bio-molecular force fields (Hansson *et al.*, 2002). Experiments that are difficult or even impossible to perform in the wet lab can be simulated using MD. From a drug discovery perspective, MD simulation can be used to study the stability of docked complexes and to ensure that molecular docking has

yielded accurate results. The stability of the docked complex is often estimated using the root mean square deviation (RMSD) and root mean square fluctuation (RMSF) during the period of the simulations (Kuzmanic and Zagrovic, 2010). RMSF values are calculated to study the thermal stability and structural flexibility (Kuzmanic and Zagrovic, 2010) while the RMSD value measures the changes in the structure during simulation. Changes in the order of 1–3 Å are considered acceptable.

Limitations of molecular docking

Docking protocols are the combination of search algorithms and scoring functions. The scoring function is a mathematical construct that is used to calculate the strength of non-covalent interactions or binding affinity (Halperin *et al.*, 2002). Some docking experiments fail due to the inability of docking methods to account for conformational changes that occur during the binding process of protein and ligand while searching the potential binding pose of the ligand in the target cavity. Predicting target receptor structural rearrangements during ligand binding is a complex problem. Unfortunately, docking tools have limited ability to follow the exact modeling of flexibility available to the protein during the binding process (Teodoro and Kavraki, 2003). This problem can be solved by MD simulations although it needs to be remembered that MD simulations are computationally expensive.

Studies comparing different docking tools on a large test set, reported 30% to 80% success rate (Cross *et al.*, 2009; Lape *et al.*, 2010). Modifying basic parameters in the docking software can drastically affect the docking and virtual screening results, indicating that expert knowledge is critical for optimizing the accuracy of docking predictions. Particular docking tool works best for specific targets, it is possible to use multiple docking tools and scoring functions in order to enhance accuracy (Charifson *et al.*, 1999). A universal docking tool (algorithm and scoring function) is not available at this time.

Illustrative Example or Case Study

Plasmodium Falciparum Plasmepsin-II Case Study

Malaria is a widespread problem estimated to kill about two million people each year (Fidock, 2010). Malaria is caused by a protozoan parasite which belongs to the genus *Plasmodium*. Despite considerable progress, researchers are yet to develop a vaccine to prevent malaria. Meanwhile, the emergence of anti-malarial drug resistance has become a major problem, particularly in Africa and south-east Asia (Murray *et al.*, 2012). Development of resistance to artemisinin derivatives and the majority of the potential partner drugs has contributed to the failures of Artemisinin-based combination treatment (ACT), recommended by World Health Organization as the current cure for malaria (Ariey *et al.*, 2014). Apart from drug resistance, other current drug issues such as poor efficacy, toxicity and high production cost also hamper malaria control.

Several targets play a vital role in the survival of *P. falciparum* in human blood stages. Hemoglobin (Hb) metabolism is a critical process during the parasite's life cycle as it provides the main source of amino acids for the growth and maturation of the parasite (Francis *et al.*, 1997). Plasmepsins (PM) are a class of enzymes that play a key role in the degradation of host hemoglobin and have been validated as potential malaria drug targets (Ersmark *et al.*, 2006). The active site cleft of PM II is different from its human ortholog, therefore, selective inhibition is possible. Moreover, PM II appears to be indispensable for parasite survival. It has been shown that PM inhibitors induce malaria parasite death both in culture and in animal models (Ersmark *et al.*, 2006). To date, several peptidic and non-peptidic inhibitors of PM II have been identified including molecules with the statin-based core (Fig. 2). However, most of these exhibits low bioavailability and are difficult to synthesize. The availability of the X-ray crystal structure of PM II makes these enzymes a useful choice for illustrating the process of structure-based drug design to discover novel anti-malarial compounds.

Three different ligand binding modes of *P. falciparum* PM II viz. 'close', 'partially open' and 'open' were described by Luksch *et al.* in 2008 based on the extent of coverage of the flap region on the binding pocket (Luksch *et al.*, 2008) (Fig. 3).

The flap region in the first group folds on the inhibitors and puts the binding pocket in the closed state (e.g., 1XE5, 1M43, 1SME). In the second group, the binding pocket is in partially open conformation (e.g., 1LEE, 1LF2, 1LF3) whereas in the third group the binding pocket is wide open due to the movement of flap lid away from the pocket (e.g., 2BJU, 2IGX). However, since then several new structures have become available in PDB which prompted us to re-look at the data. To classify the new structures, RMSD values from the structural alignment of all the available PM II inhibitor were calculated, which was later used to build a dendrogram for clustering new structures (Fig. 4).

The RMSD dendrogram confirms the three ligand-binding modes of PM II and classifies all the available PM II structures into the close, open or partially open groups. Out of all the structures available in open conformation, a high-resolution crystal structure of 2BJU was selected to demonstrate the principle of SBDD.

Materials and Methods

UCSF DOCK v6.7 (also referred to as DOCK) (Allen *et al.*, 2015) and AutoDock Vina (also referred to as VINA) (Trott and Olson, 2010) docking programs were used in this study. UCSF Chimera (Pettersen *et al.*, 2004) was used to visualize and prepare the receptor structure. A molecular format conversion program called OpenBabel (O'Boyle *et al.*, 2011) and ChemAxon tools

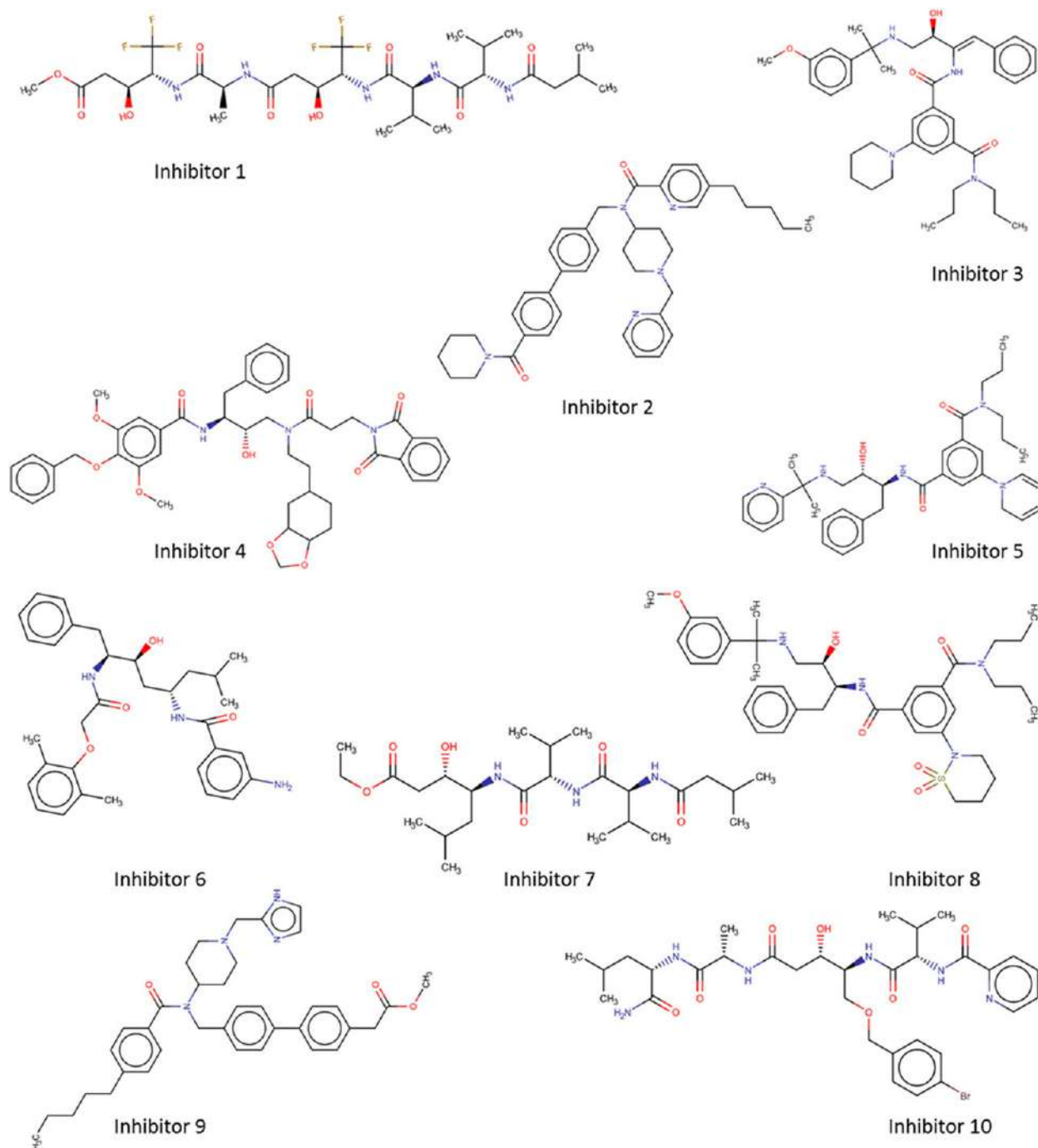


Fig. 2 Known inhibitors of *Plasmodium falciparum* Plasmepsin II.

(ChemAxon – Software for Chemistry and Biology, 2017) were used to prepare ligands for virtual screening and docking. The preparation of receptor files and the determination of the grid box were carried out using AutoDock Tools version 1.5.4.

Receptor preparation

The crystal structure of *P. falciparum* Plasmepsin, 2BJU, in complex with a potent inhibitor IH4 (IC₅₀: 34 nM) was retrieved from PDB. The bound water molecules were removed. The incomplete side chains were replaced using Dunbrack rotamer library (Dunbrack and Karplus, 1993) and the charges on the ionizable residues were made consistent with the acidic conditions found in food vacuoles (pH 5). Subsequently, Dock Prep tool in UCSF Chimera was used to complete the initial receptor preparation.

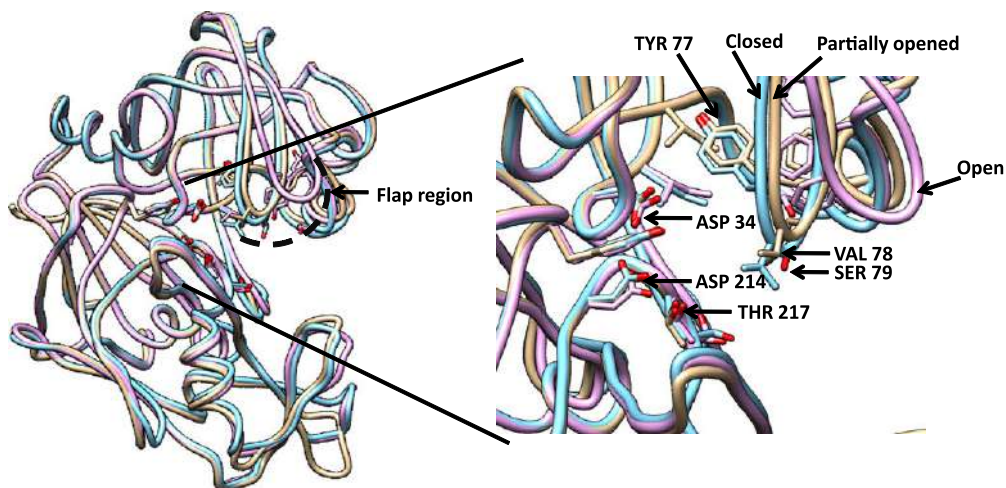


Fig. 3 Three different ligand binding conformations (closed, open and partially open) of *Plasmodium falciparum* Plasmeprin II described in the literature.

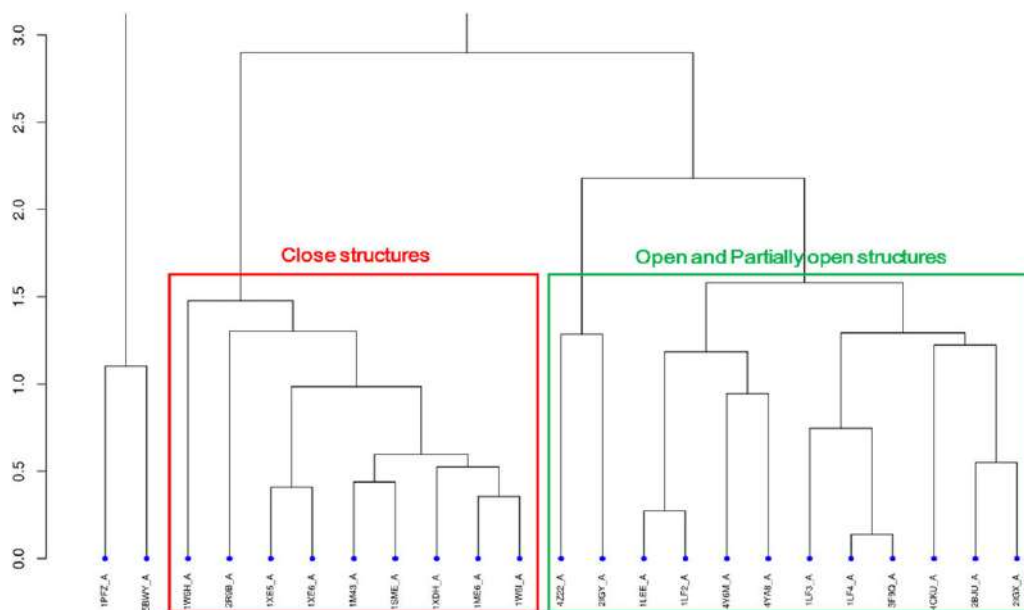


Fig. 4 Dendrogram based on RMSD calculated by structurally aligning all available *Plasmodium falciparum* Plasmeprin II structures in PDB. Structures belonging to closed and open or partially open conformations of Plasmeprin II clearly separate in distinct groups.

UCSF DOCK

After the initial preparation of the receptor, DOCK specific steps were implemented prior to virtual screening as described in DOCK manual. The first step was to calculate the solvent accessible surface area of the receptor without hydrogen atoms using a probe radius of 1.4 Å. The negative image of the surface was created by overlapping spheres using SPHGEN program. The final cluster used for docking had 44 spheres. The binding site was defined comprising all the atoms within the area of 8 Å of the co-crystallized ligand. The definition of the active site covered the catalytic dyad ASP 34 and ASP 214, along with other residues of known relevance. Subsequently, the box was constructed around the active site using SHOWBOX program and the grids were calculated using an accessory program called GRID using default parameters to complete receptor preparation.

AutoDock VINA

For VINA, the receptor PDB files were converted to PDBQT format using `prepare_receptor4.py` in MGL Tools (version 1.5.4). The default box size was calculated following the protocol mentioned by the authors of VINA (Trott and Olson, 2010). Briefly, an

initial docking box is generated based on the coordinates of the native ligand and the box dimensions are increased by 10 Å. If the box size in any dimension is smaller than 22.5 Å, it is extended to this value on all sides.

Library preparation

Around 4.5 million compounds were downloaded from the “lead-like” subset of the ZINC database (Irwin *et al.*, 2012). The dataset was cleaned by applying a number of filters to remove duplicate compounds, incorrect valence compounds and compounds violating Rule of five (Ro5). This step removed over a million compounds. The overall methodology followed for library preparation and molecular docking is shown in Fig. 5.

Next, compounds with sub-structural features commonly found in Pan Assay Interference Compounds (PAINS) were also filtered out using SMARTS strings inspired from the original Sybyl Line Notation patterns provided in the reference (Baell and Holloway, 2010). This resulted in the removal of 111,305 compounds. Additionally, the dataset was screened for potentially reactive and promiscuous compounds described by a set of 275 medicinal chemistry rules, developed at Lilly over a period of 18 years (Bruns and Watson, 2012). Reasons for rejection include reactivity, interference with assay measurements, instability, lack of

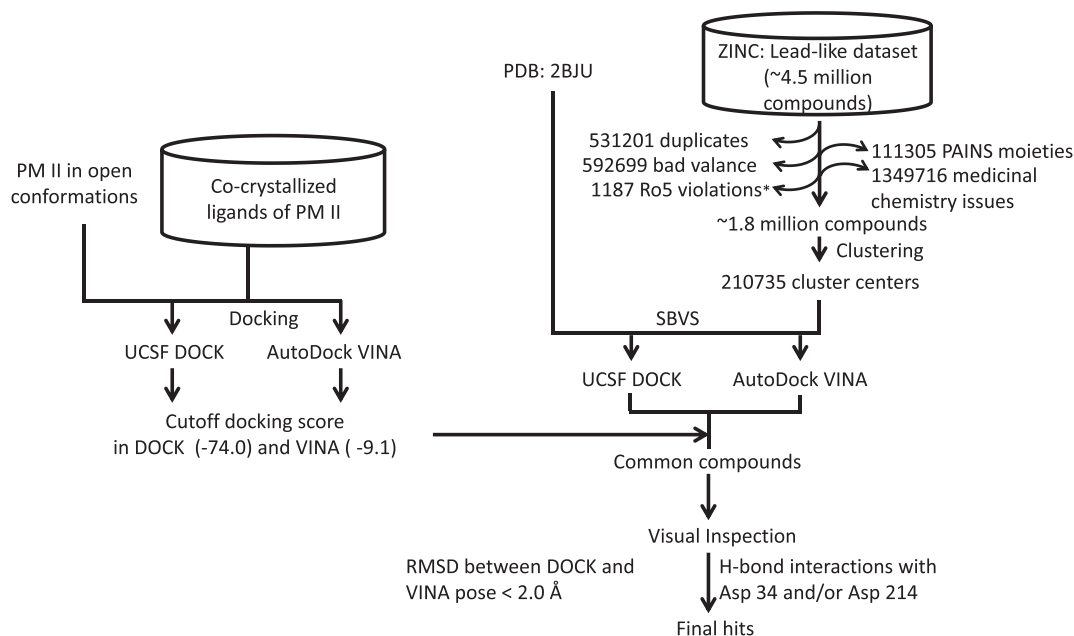


Fig. 5 The overall virtual screening methodology adopted to find novel drug leads against *Plasmodium falciparum* Plasmeprin II.

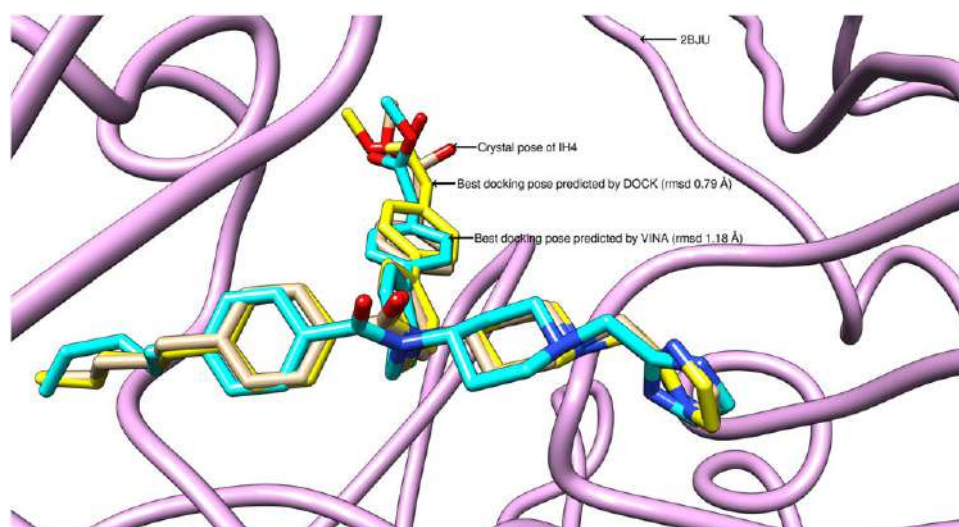


Fig. 6 Superimposition of the best docked poses of the ligand IH4 on the co-crystallized pose in the binding pocket of 2BJU. The DOCK pose (in yellow color) and the VINA pose (in cyan color) has the positional RMSD of 0.79 Å and 1.18 Å, respectively.

drug-ability and activities that damage proteins, among others. A total of 1,349,716 compounds were removed and leaving 1,864,623 filtered compounds. To ensure chemical diversity, the approximately 1.8 million remaining molecules were clustered using PubChem fingerprints as implemented in the ChemmineR package (Cao *et al.*, 2008). The maximum Tanimoto distance between a compound and the cluster center was set at 0.3. This resulted in the multi-molecule file containing 210,735 molecules which were first protonated for the acidic pH 5 and then minimized using the Ghemical force field. Subsequently, the minimized structures were used as the input to VINA and DOCK for molecular docking analysis.

Results and Discussions

Validation of the Docking Methodology

To access the suitability of the prepared target structure for structure-based drug discovery, a preliminary validation of the docking protocol was carried out by re-docking the IH4 ligand to the prepared 2BJU structure with VINA and DOCK programs. The 3D structure of the crystal ligand was obtained by removing it from protein crystallographic complex. The crystal ligand was docked to the receptor protein and from the list of conformations the pose in which the ligand bound with the maximum number of vander Waal and electrostatic interactions and lowest binding energy was chosen as the top ranked binding pose. The experimental pose of IH4 was reproduced. Fig. 6 shows the superimposition of the docked ligand (IH4) with the crystal structure position. The structure is seen to overlap well with the positional RMSD of 1.18 Å in VINA and 0.79 Å in DOCK.

As both VINA and DOCK softwares were able to reproduce the bound conformation of the co-crystallized ligand, therefore, we considered them suitable for further analysis. Overall, VINA produced better results than DOCK in ranking co-crystallized ligands in open and partially open conformations of PM II (Table 4). However, DOCK was superior in ranking 2BJU and 2IGX ligands (both in open conformation) with positional RMSD of 0.7 Å and 0.8 Å respectively, as compared to VINA with positional RMSD of 1.3 Å and 1.1 Å.

Table 4 Comparison of docking scores and rmsd values calculated by redocking co-crystallized ligands of Plasmeprin II in open and partially open conformations

PDB ID	Ligand ID	IC50	DOCK score	DOCK rmsd	VINA score	VINA rmsd	No. of rotatable bonds
Open conformation							
2BJU	1H4	34 nM	− 87.23	0.7 Å	− 10.8	1.3 Å	14
2IGX	A1T	54 nM	− 90.28	0.8 Å	− 11.5	1.1 Å	14
2IGY	A2T	113 nM	− 74.63	2.4 Å	− 9.1	1.3 Å	14
4Z22	4KG	340 nM	− 42.13	10.4 Å	− 8.2	5.9 Å	5
Partially open conformation							
1LEE	R36	18 nM	− 60.35	3.8 Å	− 9.9	1.4 Å	15
1LF2	R37	30 nM	− 58.52	2.4 Å	− 9.3	1.4 Å	17
4Y6M	48Q	70 nM	− 71.89	1.7 Å	− 8.2	1.3 Å	17
4YA8	49W	80 nM	− 76.67	1.0 Å	− 8.5	1.7 Å	16
1LF3	EH5	100 nM	− 74.54	2.6 Å	− 8.5	9.9 Å	19
4CKU	P2F	150 nM	− 82.39	1.6 Å	− 8.5	0.6 Å	17

Table 5 Novel predicted PM II lead compounds with DOCK score, VINA score, hydrogen bond interaction residues and vendor information

S. no.	ID	DOCK score	VINA score	Hbond residues	Vendors
1	ZINC09467542	− 74.57	− 10.2	SER 37, ASP 214, THR 217	eMolecules: 357481 Enamine: Z734242026
2	ZINC89512612	− 75.11	− 9.5	ASP 34, ASP 214, THR 217, GLY 36	NA
3	ZINC97236464	− 94.72	− 9.3	ASP 34, ASP 214, GLY 36	Molport: 030-015-561 Enamine: Z1631599836
4	ZINC72478028	− 75.84	− 9.3	ASP 214	eMolecules: 43171874 ChemBridge: 89912227
5	ZINC7189459	− 79.91	− 9.2	ASP 34, ASP 214	NA
6	ZINC67868828	− 84.67	− 9.1	TYR 77, ASP 214, THR 217	eMolecule: 36456659 ChemBridge: 78484059
7	ZINC25344579	− 79.04	− 9.1	ASP 34, ASP 214	Ambinter: Amb14002467 Enamine: Z339889742
8	ZINC54837115	− 81.18	− 9.1	ASP 34, ASP 214	eMolecules: 32048500 Enamine: Z734242026

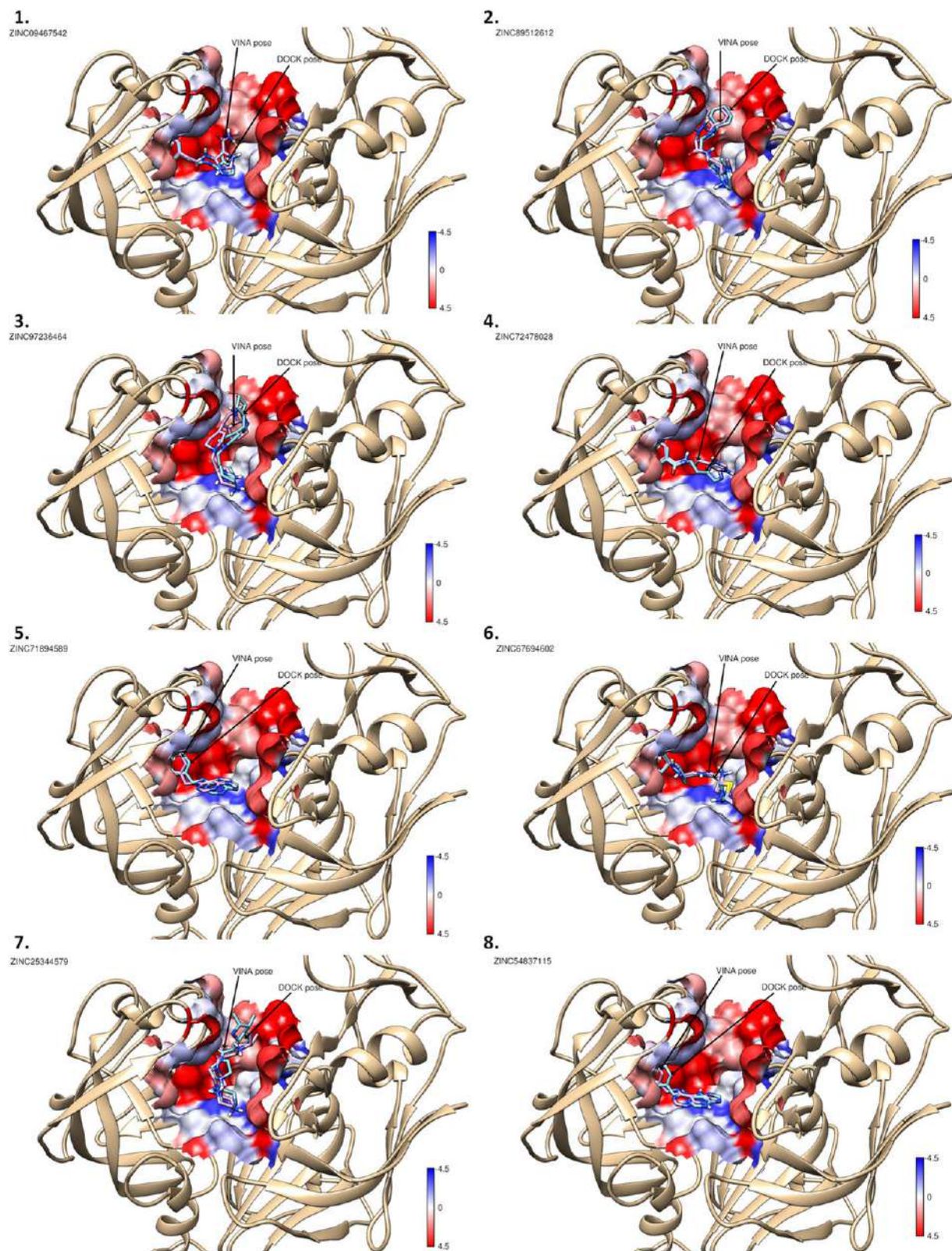


Fig. 7 Structure of eight novel potential Plasmepsin II inhibitors lead molecules. The predicted binding modes of the molecules by DOCK (cyan) and VINA (pink) in the cavity of Plasmepsin II are depicted.

Structure-Based Virtual Screening

In this study, two different docking programs were used to predict novel ligands with high binding affinity to Plasmepsin II. The final library of 210,735 compounds was screened and compounds were ranked based on the docking score to select molecules that had a score better than the cut-off score of -74.0 for DOCK and -9.1 for VINA. This resulted in a set of 823 and 1525 docked compounds from DOCK and VINA, respectively. Compounds that did not form hydrogen bonds with catalytic residues Asp34 and Asp214 were filtered out for further analysis. Subsequently, 25 common compounds were retained and visualized for binding conformations. Finally, compounds with greater than $2.0 \text{ \AA}$ RMSD between binding conformations predicted by DOCK and VINA were removed resulting in the final set of 8 compounds which are listed in Table 5 and shown in Fig. 7.

Since binding of a ligand produces a conformational change in the protein, we proposed that compounds binding in a similar fashion to the co-crystallized complex (ligand IH4 bound to 2BJU) will have a high probability to inhibit PM II *in vivo*.

Future Directions

Structure-based drug design is a powerful method, especially when combined with combinatorial chemistry and high throughput screening. Where access to large compounds libraries is limited or where HTS fails to produce a viable ligand, structure-based virtual screening continues to gain credibility as an alternative source of initial lead identification. The major developments that have facilitated the rise of SBDD to a place of prominence in drug development are dramatic increase in computational speed and efficiency of docking algorithms, ability to incorporate receptor flexibility information during docking, easily available structural data of proteins and compounds, better understanding of interaction of small molecules in the binding site and more accurate methods for estimating protein-ligand binding free energy. The application of computer-aided drug discovery approaches has been extended in both directions of the drug development pipeline, upstream for target identification and validation and downstream for ADME/T predictions.

In the modern genomic era, there has been a dramatic increase in biological data from gene sequences to structural data of proteins and small molecules. The number of potential new drug targets identified has increased more rapidly than the discovery of new drugs. One of the major challenges is to bring the wealth of structural and bioassay data together in an integrated way to design novel, improved compounds that can be readily synthesized.

Currently, there are over 60 docking tools available, evaluation and comparison of different docking programs and scoring functions is an active area of research. Further, protein flexibility during docking has only been addressed recently and the development of computational methods for protein flexibility is still in its infancy and is thereby one of the major future directions (Lill, 2011).

Closing Remarks

This article should provide the reader with an overall perspective of the field of structure-based drug discovery. We have listed and summarized several cheminformatics tools and databases with a special emphasis on software packages that are freely available. We have also outlined the major steps involved in the process of rational structure-based drug design, namely target identification, structure determination, binding site prediction, receptor and compound library preparation, molecular docking, virtual screening and binding free energy calculation. Subsequently, we have applied different SDBB approaches to predict novel leads for hemoglobin degrading enzyme of *Plasmodium falciparum* viz. Plasmepsin II. In addition, we 1) compared the performance of two different docking tools on the Plasmepsin II inhibitor dataset and 2) demonstrated the use of consensus docking methodology for virtual screening of novel PM II inhibitors.

In conclusion, computer-aided drug discovery approaches like ligand- or structure-based pharmacophore modeling and VS can successfully complement their wet-lab counterparts to expedite the drug discovery process. Although computer-aided approaches hold a great promise, it is an evolving technology and still has some shortcomings that should be remembered. The major challenges in the field are 1) to determine the structure of the target, 2) identify suitable hits, 3) progress rapidly to a molecule with the desired activity that does not fail during development and 4) ability to design novel patentable chemical compounds that are easy to synthesize. Finally, *in vitro* testing of the discovered lead molecules is necessary for further optimization of the leads.

See also: Algorithms for Structure Comparison and Analysis; Docking; Biomolecular Structures: Prediction, Identification and Analyses. Chemical Similarity and Substructure Searches. Chemoinformatics: From Chemical Art to Chemistry in Silico. Comparative Epigenomics. Computational Protein Engineering Approaches for Effective Design of New Molecules. Drug Repurposing and Multi-Target Therapies. Fingerprints and Pharmacophores. In Silico Identification of Novel Inhibitors. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Protein Structural Bioinformatics: An Overview. Protein Three-Dimensional Structure Prediction. Public Chemical Databases. Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow

References

- Adcock, S.A., McCammon, J.A., 2006. Molecular dynamics: Survey of methods for simulating the activity of proteins. *Chem. Rev.* 106, 1589–1615. Available at: <https://doi.org/10.1021/cr040426m>.
- Allen, W.J., Balius, T.E., Mukherjee, S., *et al.*, 2015. DOCK 6: Impact of new features and current docking performance. *J. Comput. Chem.* 36, 1132–1156. Available at: <https://doi.org/10.1002/jcc.23905>.
- Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. *Science* 181, 223–230.
- Ariey, F., Witkowski, B., Amarutunga, C., *et al.*, 2014. A molecular marker of artemisinin-resistant *Plasmodium falciparum* malaria. *Nature* 505, 50–55. Available at: <https://doi.org/10.1038/nature12876>.
- Arnold, K., Kiefer, F., Kopp, J., *et al.*, 2009. The protein model portal. *J. Struct. Funct. Genomics* 10, 1–8. Available at: <https://doi.org/10.1007/s10969-008-9048-5>.
- Baell, J.B., Holloway, G.A., 2010. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* 53, 2719–2740. Available at: <https://doi.org/10.1021/jm901137j>.
- Bento, A.P., Gaulton, A., Hersey, A., *et al.*, 2014. The ChEMBL bioactivity database: An update. *Nucleic Acids Res.* 42, D1083–D1090. Available at: <https://doi.org/10.1093/nar/gkt1031>.
- Bruns, R.F., Watson, I.A., 2012. Rules for identifying potentially reactive or promiscuous compounds. *J. Med. Chem.* 55, 9763–9772. Available at: <https://doi.org/10.1021/jm301008n>.
- Buzko, O.V., Bishop, A.C., Shokat, K.M., 2002. Modified AutoDock for accurate docking of protein kinase inhibitors. *J. Comput. Aided Mol. Des.* 16, 113–127.
- Cao, Y., Charisi, A., Cheng, L.-C., Jiang, T., Girke, T., 2008. ChemmineR: A compound mining framework for R. *Bioinformatics* 24, 1733–1734. Available at: <https://doi.org/10.1093/bioinformatics/btn307>.
- Capra, J.A., Laskowski, R.A., Thornton, J.M., Singh, M., Funkhouser, T.A., 2009. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLOS Comput. Biol.* 5, e1000585. Available at: <https://doi.org/10.1371/journal.pcbi.1000585>.
- Charifson, P.S., Corkery, J.J., Murcko, M.A., Walters, W.P., 1999. Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. *J. Med. Chem.* 42, 5100–5109. Available at: <https://doi.org/10.1021/jm990352k>.
- ChemAxon – Software for Chemistry and Biology, 2017. (WWW Document). ChemAxon – Softw. Chem. Biol. RD. Available at: <https://www.chemaxon.com/> (accessed 09.11.17).
- Chothia, C., Lesk, A.M., 1986. The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5, 823–826.
- Cross, J.B., Thompson, D.C., Rai, B.K., *et al.*, 2009. Comparison of several molecular docking programs: Pose prediction and virtual screening accuracy. *J. Chem. Inf. Model.* 49, 1455–1474. Available at: <https://doi.org/10.1021/ci900056c>.
- Doak, B.C., Over, B., Giordano, F., Kihlberg, J., 2014. Oral druggable space beyond the rule of 5: Insights from drugs and clinical candidates. *Chem. Biol.* 21, 1115–1142. Available at: <https://doi.org/10.1016/j.chembiol.2014.08.013>.
- Dominguez, C., Boelens, R., Bonvin, A.M.J.J., 2003. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* 125, 1731–1737. doi:10.1021/ja026939x.
- Dunbrack, R.L., Karplus, M., 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. *J. Mol. Biol.* 230, 543–574. Available at: <https://doi.org/10.1006/jmbi.1993.1170>.
- Ersmark, K., Samuelsson, B., Hallberg, A., 2006. Plasmeprins as potential targets for new antimalarial therapy. *Med. Res. Rev.* 26, 626–666. Available at: <https://doi.org/10.1002/med.20082>.
- Eswar, N., Webb, B., Marti-Renom, M.A., *et al.*, 2006. Comparative Protein Structure Modeling Using Modeller. *Curr. Protoc. Bioinforma.* Ed. Board Andreas Baxevanis AI O 5, Unit-5.6. Available at: <https://doi.org/10.1002/0471250953.bi050615>.
- Fidock, D.A., 2010. Drug discovery: Priming the antimalarial pipeline. *Nature* 465, 297–298. Available at: <https://doi.org/10.1038/465297a>.
- Fiser, A., Sali, A., 2003. Modeller: Generation and refinement of homology-based protein structure models. *Methods Enzymol.* 374, 461–491. Available at: [https://doi.org/10.1016/S0076-6879\(03\)74020-8](https://doi.org/10.1016/S0076-6879(03)74020-8).
- Forli, S., Huey, R., Pique, M.E., Sanner, M.F., Goodsell, D.S., Olson, A.J., 2016. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nat. Protoc.* 11, 905–919. doi:10.1038/nprot.2016.051.
- Forrest, L.R., Tang, C.L., Honig, B., 2006. On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins. *Biophys. J.* 91, 508–517. Available at: <https://doi.org/10.1529/biophysj.106.082313>.
- Francis, S.E., Sullivan, D.J., Goldberg, D.E., 1997. Hemoglobin metabolism in the malaria parasite *Plasmodium falciparum*. *Annu. Rev. Microbiol.* 51, 97–123. Available at: <https://doi.org/10.1146/annurev.micro.51.1.97>.
- Gherzi, D., Sanchez, R., 2009. EasyMIFs and SiteHound: A toolkit for the identification of ligand-binding sites in protein structures. *Bioinformatics* 25, 3185–3186. Available at: <https://doi.org/10.1093/bioinformatics/btp562>.
- Gilson, M.K., Liu, T., Baitaluk, M., *et al.*, 2016. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.* 44, D1045–D1053. Available at: <https://doi.org/10.1093/nar/gkv1072>.
- Grünberg, R., Nilges, M., Leckner, J., 2007. Biskit—a software platform for structural bioinformatics. *Bioinforma. Oxf. Engl.* 23, 769–770. doi:10.1093/bioinformatics/btl655.
- Groom, C.R., Bruno, I.J., Lightfoot, M.P., Ward, S.C., 2016. The Cambridge Structural Database. *Acta Crystallogr. Sect. B Struct. Sci. Cryst. Eng. Mater.* 72, 171–179. doi:10.1107/S2052520616003954.
- Gustafsson, D., Nyström, J., Carlsson, S., *et al.*, 2001. The direct thrombin inhibitor melagatran and its oral prodrug H 376/95: Intestinal absorption properties, biochemical and pharmacodynamic effects. *Thromb. Res.* 101, 171–181.
- Halperin, I., Ma, B., Wolfson, H., Nussinov, R., 2002. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins* 47, 409–443. Available at: <https://doi.org/10.1002/prot.10115>.
- Hansson, T., Oostenbrink, C., van Gunsteren, W., 2002. Molecular dynamics simulations. *Curr. Opin. Struct. Biol.* 12, 190–196.
- Heberlé, G., de Azevedo, W.F., 2011. Bio-inspired algorithms applied to molecular docking simulations. *Curr. Med. Chem.* 18, 1339–1352.
- Hendlich, M., Rippmann, F., Barnickel, G., 1997. LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graph. Model.* 15, 359–363. (389).
- Henrich, S., Salo-Ahen, O.M.H., Huang, B., *et al.*, 2010. Computational approaches to identifying and characterizing protein binding sites for ligand design. *J. Mol. Recognit. JMR* 23, 209–219. Available at: <https://doi.org/10.1002/jmr.984>.
- Huang, B., 2009. MetaPocket: A meta approach to improve protein ligand binding site prediction. *OMICS J. Integr. Biol.* 13, 325–330. Available at: <https://doi.org/10.1089/omi.2009.0045>.
- Ihlenfeldt, W.-D., Voigt, J.H., Bienfait, B., Oellien, F., Nicklaus, M.C., 2002. Enhanced CACTVS Browser of the Open NCI Database. *J. Chem. Inf. Comput. Sci.* 42, 46–57. doi:10.1021/ci010056s.
- Huang, B., Schroeder, M., 2006. LIGSITEcsc: Predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct. Biol.* 6, 19. Available at: <https://doi.org/10.1186/1472-6807-6-19>.
- Irwin, J.J., Sterling, T., Mysinger, M.M., Bolstad, E.S., Coleman, R.G., 2012. ZINC: A free tool to discover chemistry for biology. *J. Chem. Inf. Model.* 52, 1757–1768.
- Jayaram, B., Dhingra, P., Mishra, A., Kaushik, R., Mukherjee, G., Singh, A., Shekhar, S., 2014. Bhageerath-H: A homology/ab initio hybrid server for predicting tertiary structures of monomeric soluble proteins. *BMC Bioinformatics* 15, S7. doi:10.1186/1471-2105-15-S16-S7.
- Källberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H., Xu, J., 2012. Template-based protein structure modeling using the RaptorX web server. *Nat. Protoc.* 7, 1511–1522. doi:10.1038/nprot.2012.085.

- Kaldor, S.W., Kalish, V.J., Davies, J.F., *et al.*, 1997. Viracept (nelfinavir mesylate, AG1343): A potent, orally bioavailable inhibitor of HIV-1 protease. *J. Med. Chem.* 40, 3979–3985. Available at: <https://doi.org/10.1021/jm9704098>.
- Kellenberger, E., Rodrigo, J., Muller, P., Rognan, D., 2004. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. *Proteins* 57, 225–242. Available at: <https://doi.org/10.1002/prot.20149>.
- Kelley, L.A., Mezulis, S., Yates, C.M., Wass, M.N., Sternberg, M.J.E., 2015. The Phyre2 web portal for protein modeling, prediction and analysis. *Nat. Protoc.* 10, 845–858. doi:10.1038/nprot.2015.053.
- Kiefer, F., Arnold, K., Kunzli, M., Bordoli, L., Schwede, T., 2009. The SWISS-MODEL repository and associated resources. *Nucleic Acids Res.* 37, D387–D392. Available at: <https://doi.org/10.1093/nar/gkn750>.
- Kim, D.E., Chivian, D., Baker, D., 2004. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res.* 32, W526–W531. doi:10.1093/nar/gkh468.
- Kim, S., Thiessen, P.A., Bolton, E.E., *et al.*, 2016. PubChem substance and compound databases. *Nucleic Acids Res.* 44, D1202–D1213. Available at: <https://doi.org/10.1093/nar/gkv951>.
- Korb, O., Stützel, T., Exner, T.E., 2009. Empirical scoring functions for advanced protein-ligand docking with PLANTS. *J. Chem. Inf. Model.* 49, 84–96. doi:10.1021/ci800298z.
- Kramer, B., Rarey, M., Lengauer, T., 1999. Evaluation of the FLEXX incremental construction algorithm for protein-ligand docking. *Proteins* 37, 228–241.
- Kuzmanic, A., Zagrovic, B., 2010. Determination of ensemble-average pairwise root mean-square deviation from experimental B-factors. *Biophys. J.* 98, 861–871. Available at: <https://doi.org/10.1016/j.bpj.2009.11.011>.
- Lambert, C., Léonard, N., De Bolle, X., Depiereux, E., 2002. ESyPred3D: Prediction of proteins 3D structures. *Bioinforma. Oxf. Engl.* 18, 1250–1256.
- Lape, M., Elam, C., Paula, S., 2010. Comparison of current docking tools for the simulation of inhibitor binding by the transmembrane domain of the sarco/endoplasmic reticulum calcium ATPase. *Biophys. Chem.* 150, 88–97. Available at: <https://doi.org/10.1016/j.bpc.2010.01.011>.
- Laurie, A.T.R., Jackson, R.M., 2005. Q-SiteFinder: An energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics Oxford* 21, 1908–1916. Available at: <https://doi.org/10.1093/bioinformatics/bti315>.
- Law, V., Knox, C., Djombou, Y., *et al.*, 2014. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Res.* 42, D1091–D1097. Available at: <https://doi.org/10.1093/nar/gkt1068>.
- Le Guilloux, V., Schmidtke, P., Tuffery, P., 2009. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinform.* 10, 168. Available at: <https://doi.org/10.1186/1471-2105-10-168>.
- Lill, M.A., 2011. Efficient incorporation of protein flexibility and dynamics into molecular docking simulations. *Biochemistry (Moscow)* 50, 6157–6169. Available at: <https://doi.org/10.1021/bi2004558>.
- Lipinski, C.A., 2000. Drug-like properties and the causes of poor solubility and poor permeability. *J. Pharmacol. Toxicol. Methods* 44, 235–249. Available at: [https://doi.org/10.1016/S1056-8719\(00\)00107-6](https://doi.org/10.1016/S1056-8719(00)00107-6).
- Locating Binding Sites in Protein Structures, 2017. (WWW Document). Available at: <https://www.chemcomp.com/journal/sitefind.htm> (accessed 09.09.17).
- López-Vallejo, F., Caulfield, T., Martínez-Mayorga, K., *et al.*, 2011. Integrating virtual screening and combinatorial chemistry for accelerated drug discovery. *Comb. Chem. High Throughput Screen.* 14, 475–487.
- Luksch, T., Chan, N.-S., Brass, S., *et al.*, 2008. Computer-aided design and synthesis of nonpeptidic plasmepsin II and IV inhibitors. *Chem. Med. Chem.* 3, 1323–1336. Available at: <https://doi.org/10.1002/cmdc.200700270>.
- Macarron, R., Banks, M.N., Bojanic, D., *et al.*, 2011. Impact of high-throughput screening in biomedical research. *Nat. Rev. Drug Discov.* 10, 188–195. Available at: <https://doi.org/10.1038/nrd3368>.
- Martí-Renom, M.A., Stuart, A.C., Fiser, A., *et al.*, 2000. Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.* 29, 291–325. Available at: <https://doi.org/10.1146/annurev.biophys.29.1.291>.
- McGann, M., 2011. FRED Pose Prediction and Virtual Screening Accuracy. *J. Chem. Inf. Model.* 51, 578–596. doi:10.1021/ci100436p.
- Meng, X.-Y., Zhang, H.-X., Mezei, M., Cui, M., 2011. Molecular docking: A powerful approach for structure-based drug discovery. *Curr. Comput. Aided Drug Des.* 7, 146–157.
- Murray, C.J.L., Rosenfield, L.C., Lim, S.S., *et al.*, 2012. Global malaria mortality between 1980 and 2010: A systematic analysis. *Lancet London* 379, 413–431. Available at: [https://doi.org/10.1016/S0140-6736\(12\)60034-8](https://doi.org/10.1016/S0140-6736(12)60034-8).
- Neves, M.A.C., Totrov, M., Abagyan, R., 2012. Docking and scoring with ICM: The benchmarking results and strategies for improvement. *J. Comput. Aided Mol. Des.* 26, 675–686. doi:10.1007/s10822-012-9547-0.
- Nisius, B., Sha, F., Gohlke, H., 2012. Structure-based computational analysis of protein binding sites for function and druggability prediction. *J. Biotechnol.* 159, 123–134. Available at: <https://doi.org/10.1016/j.jbiotec.2011.12.005>.
- O’Boyle, N.M., Banck, M., James, C.A., *et al.*, 2011. Open Babel: An open chemical toolbox. *J. Cheminform.* 3, 33. Available at: <https://doi.org/10.1186/1758-2946-3-33>.
- Owens, J., 2007. Target validation: Determining druggability. *Nat. Rev. Drug Discov.* 6, nrd2275. Available at: <https://doi.org/10.1038/nrd2275>.
- Parasuraman, S., 2012. Protein data bank. *J. Pharmacol. Pharmacother.* 3, 351–352. Available at: <https://doi.org/10.4103/0976-500X.103704>.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF Chimera – A visualization system for exploratory research and analysis. *J. Comput. Chem.* 25, 1605–1612. Available at: <https://doi.org/10.1002/jcc.20084>.
- Pieper, U., Webb, B.M., Barkan, D.T., *et al.*, 2011. ModBase, a database of annotated comparative protein structure models, and associated resources. *Nucleic Acids Res.* 39, D465–D474. Available at: <https://doi.org/10.1093/nar/gkq1091>.
- Pollack, V.A., Savage, D.M., Baker, D.A., *et al.*, 1999. Inhibition of epidermal growth factor receptor-associated tyrosine phosphorylation in human carcinomas with CP-358,774: Dynamics of receptor inhibition in situ and antitumor effects in athymic mice. *J. Pharmacol. Exp. Ther.* 291, 739–748.
- Repasky, M.P., Shelley, M., Friesner, R.A., 2007. Flexible ligand docking with Glide. Chapter 8, Unit 8.12. *Curr. Protoc. Bioinforma.* doi:10.1002/0471250953.bi0812s18.
- Roy, A., Kucukural, A., Zhang, Y., 2010. I-TASSER: A unified platform for automated protein structure and function prediction. *Nat. Protoc.* 5, 725–738. doi:10.1038/nprot.2010.5.
- Ruiz-Carmona, S., Alvarez-Garcia, D., Foloppe, N., *et al.*, 2014. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Comput. Biol.* 10. doi:10.1371/journal.pcbi.1003571.
- Schindler, T., Bornmann, W., Pellicena, P., *et al.*, 2000. Structural mechanism for STI-571 inhibition of abelson tyrosine kinase. *Science* 289, 1938–1942.
- Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., Wolfson, H.J., 2005. PatchDock and SymmDock: Servers for rigid and symmetric docking. *Nucleic Acids Res.* 33, W363–W367. doi:10.1093/nar/gki481.
- Stahura, F.L., Bajorath, J., 2004. Virtual screening methods that complement HTS. *Comb. Chem. High Throughput Screen.* 7, 259–269.
- ten Brink, T., Exner, T.E., 2009. Influence of protonation, tautomeric, and stereoisomeric states on protein-ligand docking results. *J. Chem. Inf. Model.* 49, 1535–1546. Available at: <https://doi.org/10.1021/ci800420z>.
- Teodoro, M.L., Kavasaki, L.E., 2003. Conformational flexibility models for the receptor in structure based drug design. *Curr. Pharm. Des.* 9, 1635–1648.
- Trott, O., Olson, A.J., 2010. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. *J. Comput. Chem.* 31, 455–461. Available at: <https://doi.org/10.1002/jcc.21334>.
- Varghese, J.N., 1999. Development of neuraminidase inhibitors as anti-influenza virus drugs. *Drug Dev. Res.* 46, 176–196. Available at: [https://doi.org/10.1002/\(SICI\)1098-2299\(199903/04\)46:3/4<176::AID-DDR4>3.0.CO;2-6](https://doi.org/10.1002/(SICI)1098-2299(199903/04)46:3/4<176::AID-DDR4>3.0.CO;2-6).
- Verdonk, M.L., Cole, J.C., Hartshorn, M.J., Murray, C.W., Taylor, R.D., 2003. Improved protein-ligand docking using GOLD. *Proteins* 52, 609–623. doi:10.1002/prot.10465.
- Wang, T., Wu, M.-B., Zhang, R.-H., *et al.*, 2016. Advances in computational structure-based drug design and application in drug discovery. *Curr. Top. Med. Chem.* 16, 901–916.
- Weisel, M., Proschak, E., Schneider, G., 2007. PocketPicker: Analysis of ligand binding-sites with shape descriptors. *Chem. Cent. J.* 1, 7. Available at: <https://doi.org/10.1186/1752-153X-1-7>.

- Williams, A.J., 2010. ChemSpider: Integrating Structure-Based Resources Distributed across the Internet, in: Enhancing Learning with Online Resources, Social Networking, and Digital Libraries, ACS Symposium Series. American Chemical Society, pp. 23–39. <https://doi.org/10.1021/bk-2010-1060.ch002>
- Wishart, D.S., Jewison, T., Guo, A.C., *et al.*, 2013. HMDB 3.0—The Human Metabolome Database in 2013. *Nucleic Acids Res.* 41, D801–807. doi:10.1093/nar/gks1065.
- Xie, Z.-R., Hwang, M.-J., 2015. Methods for predicting protein-ligand binding sites. *Methods Mol. Biol. Clifton* 1215, 383–398. Available at: https://doi.org/10.1007/978-1-4939-1465-4_17.
- Yang, J.-M., Chen, C.-C., 2004. GEMDOCK: A generic evolutionary method for molecular docking. *Proteins* 55, 288–304. doi:10.1002/prot.20035.
- Yuriev, E., Ramsland, P.A., 2013. Latest developments in molecular docking: 2010–2011 in review. *J. Mol. Recognit. JMR* 26, 215–239. Available at: <https://doi.org/10.1002/jmr.2266>.

Biographical Sketch



Varun Khanna is a research scientist at Vaxine Pty Ltd, a biotechnology company involved in vaccine design. He completed his PhD in Cheminformatics from Macquarie University, Sydney, Australia in 2011. During his PhD he analyzed physicochemical properties, occurrence and co-occurrence patterns of molecular fragments in pharmaceutically important chemical compounds for advancing the knowledge of computational library design in drug discovery. He then moved to University of California, Riverside, USA as a post-doc where he was involved in the chemogenomics, data mining project aimed at carrying out a comprehensive selectivity-centric analysis of ligand-target interactions in public activity profile databases such as PubChem Bioassay. His current research interest includes developing novel techniques for computational drug design, immuno-informatics, fragment-based virtual screening, machine learning and reverse vaccinology. He has authored several research papers in international peer-reviewed journals and two book chapters.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books in as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.



Nikolai Petrovsky is a Professor in the School of Medicine at Flinders University, Director of Endocrinology at Flinders Medical Center and the founder of Vaxine Pty Ltd, a biotechnology company involved in vaccine design. His research interests including vaccine design, adjuvants, immuno-informatics, and autoimmunity. He has been a principal investigator on multiple grants from National Institutes of Health and has won prestigious awards including the Ernst & Young Entrepreneur of the Year Award and a Biopharma Asian Executive of the Year Award. He has taken vaccines for seasonal and pandemic influenza, hepatitis B and insect sting allergy from the bench to human clinical trials and authored over 140 research papers.

Chemoinformatics: From Chemical Art to Chemistry in Silico

Jaroslav Polanski, University of Silesia Institute of Chemistry, Katowice, Poland

© 2019 Elsevier Inc. All rights reserved.

Introduction

Chemoinformatics (cheminformatics) a term associating chemistry and informatics (computer science) on the lexical level describes a discipline organizing and coordinating the application of computers in chemistry. However, some branches of computational chemistry usually are not included in the field, although they depend on computers. The example here is quantum chemistry which is more frequently insisted to be a part of theoretical physics than cheminformatics because, in principle, it is just the physics of atoms and mathematics that allow the correct modeling of molecular objects, and pure mathematics, hypothetically, can be done without computers (Polanski, 2009b). However, the larger the molecules, the more remote and inaccessible the precise mathematical model. The history of fundamental concepts or theories in chemistry illustrates the complexity of chemical researches. Consider chemical bonding. The molecular orbitals or valence bonding theories describe atomic scale aggregation into molecules. Both models have been competing with each other and chemists still discuss which is correct and which is better. In physics, it is possible to develop a relatively simple model to explain certain facts of nature. In contrast, in chemistry, a theory often partially interprets some data, also offering partial solutions. Thus, we need several high-quality models for the correct theory (Brock, 1993). Theoretical chemistry is an empirical science based on the Schrödinger equation; however, we are aware that its general solution will not be available. In science we use mathematics for modeling and developing theories. In the language of informatics this means mathematics is a compression tool unifying the facts of nature. Now we do not need individual facts any longer. Details disappear and information is coded into a form of a model (occasionally a single equation) that explains the reality. Compressing the complexity of natural information to such models is used typically for an approach called reductionism (Cohen and Stewart, 1994). What if, the Nature appears too complex for an accurate mathematical description or the developed model does not have a precise solution. The answers could be still important but unavailable by precise mathematics. In such situations, we need further substantial simplifications, even in cost of reliability. This brings into the game speculation or educated guess which are still better than blind guess or no answer. Chemists discovered that “educated guess is being supported by the computer” (Kowalski and Bender, 1975). This describes the first application of computers in chemistry, which is to assist a chemist in a calculation or computation that requires the calculation or simulation by computers. Why can computations still be possible, flexible, and efficient in data processing when human calculations fail? The efficiency of *in silico* mathematics is first of all, not by computer intuition or flexibility but by a brute force under relatively simple mathematical formalism (Audibert, 2013; Bock and Goode, 2002). The methods played in computers are evidently different than those involving human brain. Coupling human intelligence with the speed and competence in low-level manipulations observed in computer appeared efficient enough to solve “formerly intractable problems, and explore areas beyond the reach of human calculation” (Audibert, 2013; Jager and Krebs, 2003). Accordingly, the domain of cheminformatics can be preferentially outlined by calculations that cannot do without mathematics in *in silico*, or in other words, those chemistry branches that process massive or ill-defined and fuzzy data for which standard mathematical modeling is unavailable. It was suggested that, chemistry branches that “generate more numbers than information, e.g., physical and chemical property calculation” are not in cheminformatics (Oprea, 2003). It seems however that even though the calculations in the latter case can be performed efficiently in *in silico*, hypothetically, they can do completely without computer assistance basing on the relatively simple mathematics.

The Scope of Cheminformatics

Imitation or simulation of natural systems by building physical or virtual models is a core of scientific method. Basically, cheminformatics is a tool for chemistry simulation in computer. Recent definitions: *using informatics method to solve chemical problems* (Gasteiger and Engel, 2004) or just *in silico chemistry* (Polanski, 2009b) well define the objectives and the scope of the discipline. A term cheminformatics has been coined in the late 90', when computer applications in chemistry had been already controlled by chemometrics, another discipline which associates chemistry and informatics. At the same time chemists were already aware of the potential of computer applications in advanced molecular problems at the same time however also realized that such problems as drug (molecular) design will not proceed as smooth as previously expected. The experience that chemists gained here quenched the early enthusiasm. We understood far more advanced methods would be needed for the efficient *in silico* chemistry. At the same time chemometrics which developed from analytical chemistry only partially cover molecular problems. Therefore, cheminformatics originated as a result of early disappointment of computer assisted drug design as a discipline conceived for “the combination of all the information resources that a scientist needs to optimize the properties of a ligand to become a drug” (Brown, 1998). In Fig. 1 we briefly illustrated the scope of chemical investigations performed in computer in the association with other disciplines. Eventually, a new drug is not only biophysics, but also economic market with its unpredictable behavior. Therefore, the scope of cheminformatics is broad, extending from physics to economy. The large scope decides the complexity of problems is enormous high. In turn, high complexity implies low precision. This fact explains the

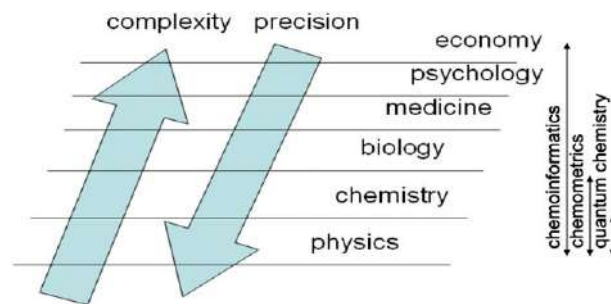


Fig. 1 A precision and complexity in chemoinformatics which scope extends from physics to economy.

autonomous character of chemoinformatics which had to develop their own methodology to trigger the computer to zoom on chemical problems recognizing specific chemical patterns and replying to them accordingly. The history of the computer chemistry can be found in Polanski (2009b), Noordik (2004), Polanski and Gasteiger (2016).

Teaching Computers Chemistry: From Molecular Data to Chemical Information and Chemical Ontology

The development of technology, in particular, scientific equipment capable of describing human environment by any type of records provided us with data, where data can be interpreted in the broadest sense as anything that is recorded, including metadata, i.e., data referring to other data. This can be both ordered and unordered collections of both nominal and numerical values, whereas the latter can be discrete numbers, intervals or ratios. Another type of data (binary large objects, BLOBs) is used to describe audio, video and graphic files (Maheshwari, 2014). With the enormous increase of data volume the so-called big data appeared recently. It is generally believed that big data brings new value and innovation. For example, Szlezak *et al.* cited the recent McKinsey research that suggests that the potential use of big data in US health care could reduce costs by \$300 billion a year (Szlezak *et al.*, 2014). Current impact of big data for drug design is however far less noticeable that could be expected. The reason is that, first, this kind of information is much less clearly defined and messy. Accordingly, its analysis causes serious problems. Second, 120 million compounds were synthesized and registered in databases; however, this should be compared to 7 billion individuals in human society. Data gets bigger while going from chemistry to biology or economy (where humans are interacting). A single genome for a single individual can be interpreted as special type “big data”, e.g., already a structure of the genome is a large record. A phenotype yielded by the genome is a specific property. In comparison, a population of drugs or bioactive molecules is represented by much smaller data, in particular measured properties. Therefore, the generation of big data here must go by the increase of the number of properties registered for a single molecule, e.g., polypharmacology. Third, the availability of big data for drugs is limited because of the need to keep the secret during drug development. Therefore, in order to replace such data a library of building blocks is analyzed to probe big data behavior of the large molecular populations (Polanski *et al.*, 2016). Formal classifications of big data and their analyses were discussed in Polanski (2017).

Science, in particular, chemistry is knowledge organized from data, that is, facts and numbers making up information. In turn, information develops chemical knowledge. Atoms, molecules, substances and their properties and transformations are the main objectives of chemistry. An unbelievably large number of molecules can be arranged from a matter available in the universe. The efficient data storage and management is needed for the control of this molecular representation, which indicates the next field of computer application in chemistry. Generally, a structure of chemical information is often chaotic which prevents its obvious transformation to in silico form. This can be illustrated by a fact that chemists often underline chemistry is an art. This means that a human expert is needed for the efficient playing with uncertainty in the field. In turn, computer is an unsophisticated *device* for which uncertainty is a problem. Therefore, computer-understandable chemistry needed to be developed to store and process chemical data and information. Translating molecular data into a machine-readable and -processable structure is far from triviality. A problem is also the interaction between chemist and computer.

Chemical compounds (CC) are the main objects of chemical investigations. Surprisingly, although commonly used in chemistry a term chemical compound has never been defined by IUPAC. Historically, the ambiguity of the term CC can be related to the early term of *mixts*, a result of *mixing different bodies* (not necessarily in the contemporary chemical sense). Mixts was replaced in 18th century by composition or compounds (Bensaude-Vincent and Simon, 2012). What originally related to substances can be however also associated with molecules which are the combinations of chemical elements. Therefore, CC refers both to molecules and substances (chemical species). Actually, the meaning of the term CC is intuitively interpreted by a chemist and can indicate (Polanski and Gasteiger, 2016):

- A *synonym of a molecule* (composed of at least two different atoms) representing a single chemical body for virtual needs (coding, in silico processing, visualization, etc.)
- A *molecule* representing a single chemical body under measurement (e.g., a single DNA strand that can be observed, e.g., under microscopy).

- A *substance* composed of a replicated but single molecular entity under measurement.
- A *chemical species* under measurement when replicated is more than one molecule type.

However, a simplified meaning of a molecule is also a model-like representation that can refer both to a chemical compound that has been yet synthesized and described, or to a virtual structure (hypothetical compound) that is under design or speculation. Historical reasons decide that molecules belong to the domain of organic or inorganic chemistry. Although the organic vs. inorganic typology is more and more fuzzy the proportions of chemical compounds are approximately 1:200 in favor of organic chemistry, if we follow standard rules. The management of chemical compounds needs the efficient machine-searchable databases registering all virtual structures and real compounds that have been synthesized and described in chemical literature from the very early days to today. This problem of substantial importance for the organization of chemistry in silico is known under the term *structure representation and searching* (Willet, 2003) which basic concepts has been formulated as early as 1960s. Atomic composition given by molecular formulae is not sufficient to unambiguously identify a molecule. Therefore, in the broadest sense the structure of the chemical entity is defined by constitution and stereochemistry. What we mean by constitution is “the description of the identity and connectivity (and corresponding bond multiplicities) of the atoms in a molecular entity (omitting any distinction arising from their spatial arrangement, i.e., molecular stereochemistry)”. Isomers are chemical entities of the same atomic composition but different constitution and/or stereochemistry or precisely isomers are “one of several species (or molecular entities) that have the same atomic composition (molecular formulae) but different line formulae or different stereochemical formulae and hence different physical and/or chemical properties.” In particular, a line formula is an example of molecular representation showing atoms that are “joined by lines representing single or multiple bonds, without any indication or implication concerning the spatial direction of bonds.”

We more and more understand in chemistry the importance of the precise shape of chemical formalism, typical for mathematics and physics. However, at the same time we would like to preserve the completeness of chemical information with its uncertainty. Therefore, we need an intelligent in silico system fitted to receive the *soft* chemical data without information reduction or deterioration. In this context *chemical ontology* is a recent solution of this problem (Hastings *et al.*, 2011). It is a flexible dictionary like chemistry representation, a system capable of organizing the complete chemical information in silico. In other words, the strategy here is that the chaotic nature of chemical information is tamed not by the significant information reduction but by building a structure capable of accepting all variety of data and facts. The inspiration for the chemical ontology seems to be object-oriented programming where a processing of information attempts to imitate reality. We are defining here not only the data type of a data structure, but also the types of operations or functions that can be applied to the data structure. Therefore, data are structured by objects that includes both data and functions. Furthermore, relationships between the objects are defined. In particular a category of class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods). Accordingly, let focus on chemical compound which can be defined as a class having a fixed ratio of defined atoms and a chemical structure that can be expressed by one connection table of non/hydrogen atoms and one or more connection tables that include hydrogen atoms and bond orders as well, connected by OR logic. This allows, for example, to clearly classified all tautomeric forms of vitamin C as a single chemical compound (Bobach *et al.*, 2012).

Between Descriptors and Properties

Basically, chemical compounds can be represented by descriptors or properties, whereas a property is to be measured in experiment and a descriptor is calculated or more precisely is a final result of a logic and mathematical procedure transforming chemical information encoded within a symbolic representation of a molecule into a useful number (Todeschini and Consonni, 2000). As in real world we usually have a contact with substances the properties will mainly refer to the substances, while descriptors to the molecules. However, this is not always true because we can also measure the property of a single molecule, e.g., MW in the MS spectrometer like conditions. A fully consistent differentiation between properties and descriptors could not always be easy. Interestingly, while IUPAC identifies only properties (descriptors are also classified as properties); in chemoinformatics all parameters are often referred to as descriptors. Properties are recorded here as physicochemical descriptors, e.g., molecular descriptors are divided into two main classes: experimental measurements, such as log P, molar refractivity, dipole moment, polarizability, and, in general, physicochemical properties, and theoretical molecular descriptors” (Consonni and Todeschini, 2010). In turn Polanski and Gasteiger argued for a clear and consistent descriptors versus properties differentiation (Polanski and Gasteiger, 2016). It has been shown recently how important and informative this taxonomy can appear (Polanski *et al.*, 2016; Polanski and Tkocz, 2017). Accordingly it has been demonstrated that *ligand efficiency* (LE), a molecular descriptor designed for an early evaluation of drug candidates (Hopkins *et al.*, 2014); if interpreted not as a descriptor but as statistical property allows to explain an evident preference of small ligands. Actually, LE does not characterize a real binding potency but is related to the statistics involved in the binding of the 1 g sample of the ligands (Polanski and Tkocz, 2017). This indicated the next difference between descriptor which can be clearly related to a single molecule and a property of the substance which can also be a statistical parameter not so obviously connected to a single molecule.

From Problems to Concepts

In the previous articles we illustrate the difficulties of the adaptation of chemistry to an in silico method. A variety of terms that are interpreted by human chemists intuitively are not well suited for computers. This should have been first understood well enough

to be transferred to a form understandable for computers. Structure representation is an example of the early problem that should have been solved to teach computers chemistry. We briefly presented the examples of the molecular problems and in silico concepts to deal with them after (Polanski and Gasteiger, 2016).

Problems

- *Chemical data documentation and searches*: database searches and management
- *Calculating molecular descriptors*: 2D (chemical graphs) and 3D (molecular modeling) representations are mapped into single numbers (0D); vectors, fingerprints (1D); matrixes, surface maps (2D); surface, atomic representations, force fields, virtual or real receptor data (3D); etc.
- *Molecular modeling*: molecular structure 3D data generation in silico from their 2D or 1D representations. This includes both predictions of molecular topography and molecular descriptors i.e., atomic annotations data by calculated (simulated) atomic properties.
- *Structure elucidation*: (synonym: structure-spectra correlations) property to molecular structure is mapped in factual chemical space (FCS) when we are attempting to find a structure having a certain spectra or in structure to property version we are simulating a spectra for a given molecule (substance).
- *Mapping structure to activity* (SAR): a series of structures (FCS substances) is needed to study SAR, which are usually synthesized. The real goal of SAR is usually to predict the training series for designing new structures and substances by property predictions in a qualitative or quantitative procedures.
- *Property prediction*: compounds' series (FCS) are mapped from FCS into virtual chemical space (VCS); two basic versions are available, i.e., property vs. property or structure vs. property. However, in the design step for novel compounds in VCS (we can design both compounds VCS or FCS where some properties can be registered in databases and/or literature) we are always to use a structure version (no property is available in VCS).

More Important Concepts

- chemical space (CS), factual CS (FCS) or virtual CS (VCS) SMILES coding
- connectivity (graph theory) approaches, additivity concept, molecular modeling, force fields
- molecular mechanics, molecular dynamics, force field, molecular interaction field, partial atomic charges, lipophilic potential
- RDF structure coding (2D), 2D structure–2D spectra correlation
- SAR, QSAR, QSAR domain, similarity measures, privileged structures, fragonomics
- QSAR, QPAR, log*P* versus partition coefficient, fragonomics, virtual screening
- Synthons, retrosynthetic analysis, synthesis tree

Some of them are adopted directly from chemistry where they originated not necessarily with a thought of computer applications. For some others in silico methods were an obvious inspiration.

Chemical Space: From Storing Information to Advanced Ontology Formalism

The population of chemical data has significantly increased in recent years. In particular, Chemical Abstracts Service (CAS) has registered almost 130 million unique organic and inorganic chemical substances, 67 million sequences. The daily update records 15,000 substances. Substance information, includes both molecular descriptors, property and predicted property data includes more than 7.6 billion property values, data tags and spectra. CASREACT contains 83.6 million single and multistep reaction data entries. The Pubchem BioAssay Database records count to millions (Wang *et al.*, 2017). The number of potential compounds, i.e., the population of chemical space is however much higher. This number is estimated between 10^{18} and 10^{200} , whereas 10^{60} is probably the most representative value found in the literature. We can compare this to the factual CS of the order of $10^7/10^8$. Alternatively, a number of stars in the universe is estimated to 10^{22} (Polanski and Gasteiger, 2016; Baldi, 2005; Bohacek *et al.*, 1996). The enormous population of CS can be realized better if we evaluate the potential expansion of *n*-hexane into a family of analogues decorated with 150 different substituents. Accordingly, if we bring together a full collection of mono- to 14-substituted analogues count to a population of 10^{29} (Lipinski and Hopkins, 2004). Therefore, chemical synthesis or the construction of new chemical molecules cannot do without a well-organized data mining system that allows us for a verification of physical and/or chemical data by screening a large population of the variety of compounds described. This problem is fundamental for chemistry, where the access to information is a common need. Moreover we would like to have proper data delivered at the proper time to the chemist's desk. An efficient chemical information system on chemical compounds needed to be developed from the very beginning of chemical investigations. Actually, *Chemisches Zentralblatt* appeared as early as 1830; while the first edition of Beilstein's *Handbuch der Organischen Chemie* (BH) was published in 1881. Even in that time the latter contained two volumes, registering 1500 compounds, with more than 2000 pages. In chemistry we realized quite early that the improvement in data storage could be of the crucial importance for the development of chemical sciences. However, searching a compound within thousands of pages of the printed book was tedious and impractical. Instead, the computer desktop could provide an

efficient environment for the management of chemical data and chemistry could evidently profit from computers. Accordingly, chemical information is a core component of chemoinformatics.

This means that we need new tools for storing and manipulating chemical data. *Chemical space* (CS) is an example of the idea that contributes not only into the computations but also to the chemical documentation. The original CS concept was inspired by the cosmological universe populated by stars, where chemical compounds replace cosmological objects to navigate chemical space (Lipinski and Hopkins, 2004). The vague cosmological analogy can be replaced by more precise parallels. Accordingly, mathematical model was designed. Practically, clear illustration of the chemical operations by in vitro and in silico operators appeared to be the most important functionality in this method (Polanski, 2009b; Polanski and Gasteiger, 2016). Summing up CS is a concept which appeared in the hope of organizing a whole population of chemical compounds. Essentially, CS is a structure for the mapping of chemical compounds by descriptors and properties. Furthermore, potential operations schemes accompany crude data. If we realize these functionalities we can easily understand that in the current chemoinformatics *chemical ontology* is an advanced formal structure that represents a CS system for in silico chemistry that allows for a flexible multipurpose property and descriptor registering and manipulation in silico. Ontologies encode human knowledge in computationally accessible forms. They are designed to narrow the gap between the knowledge of human experts and the functionality available in computer systems, by expressing expert knowledge in a manner computers can manipulate and reason over (Hastings et al., 2011).

Basic In Silico Functionalities

Coding Chemical Information Into Molecular Descriptors

Molecular graphs encode chemical information that can be transformed to a variety of useful numbers, or in other words molecular descriptors (Todeschini and Consonni, 2000). A variety of molecular descriptors, their taxonomy and detailed algorithms or software for their calculations from molecular representation are available nowadays. This decides that usually a large number of calculated molecular descriptors are accompanied by few measured property data. This effect described as a property deficit (Polanski and Gasteiger, 2016; Polanski, 2017) results from a fact that the measurements are much more expensive than in silico calculations. On the other side this illustrates the success of in silico chemistry.

Computer-Processable Molecular Codes

Two-dimensional atom arrangement of chemical entities can be coded into molecular graphs which are easy-to-read for chemists; however, are not computer-friendly. Accordingly, we need a method for a proper translation of molecular graphs to a form that would be understandable for a computer system. Coding molecular graphs by linear notation or connection table forms a typical computer representation. In the linear notation a molecule is represented by a string of the line formulae type. An example of SMILES (Simplified Molecular Input Line Entry) notation can be an up-to-date illustration of such a system (Weininger, 2003). The detailed tutorial can be found online in the Daylight Chemical Information Systems. Matrix representation is an alternative for coding chemical graphs. The examples here are adjacency matrix, atom connectivity matrix, incidence matrix, bond electron matrix. Molecular structures can also be coded by connection tables which record atoms and bonds within a molecule. The need of the individual application decides a form of the computer representation that we use. An in-depth description of the matrix and connection table codes can be found in the Handbook of Chemoinformatics edited by Gasteiger (2003). A precise definition of chemical compound could need a combination of connection tables, which can be achieved by the chemical ontology method.

A connection table or linear notation can be formed arbitrarily. This means that numbers can be assigned to the atoms differently and there is none standard molecular representation. Canonical labeling is a solution for this problem. This provides a unique representation for a certain molecular graph. Unique SMILES are example of such canonical labeling system (Daylight Chemical Information System). Chirality is an important chemical structure property and ISOMERIC SMILES is a system allowing for various chiral and isotopic specifications.

Molecular Editors

Molecular graphs are unambiguous, chemist friendly and illustrative way for the presentation of molecules constitution and stereochemistry. Molecular editor is an interface that allows a user not only to draw professionally presented molecular structures but it is also a tool for the translation of such a structure into the computer processable molecular codes. A number of systems have been developed that are capable of the translation of molecular formulas introduced into computer by its user in a form of direct drawing, e.g., using a mouse, to the machine readable code. ISIS, ChemSketch ACDLAB, JME, RasMol are molecular editors available free of charge at the web sites, respectively (Polanski, 2009b).

Coding Chemical Reactions

Chemical reaction refers to a dynamic behavior involving both a breaking and formation of chemical bonds, in which chemical compounds are transformed from reagents to products. Accordingly, chemical reaction changes the atom bonding system in molecules. The problems encountered while naming chemical reactions resemble those that are typical in the nomenclature of chemical

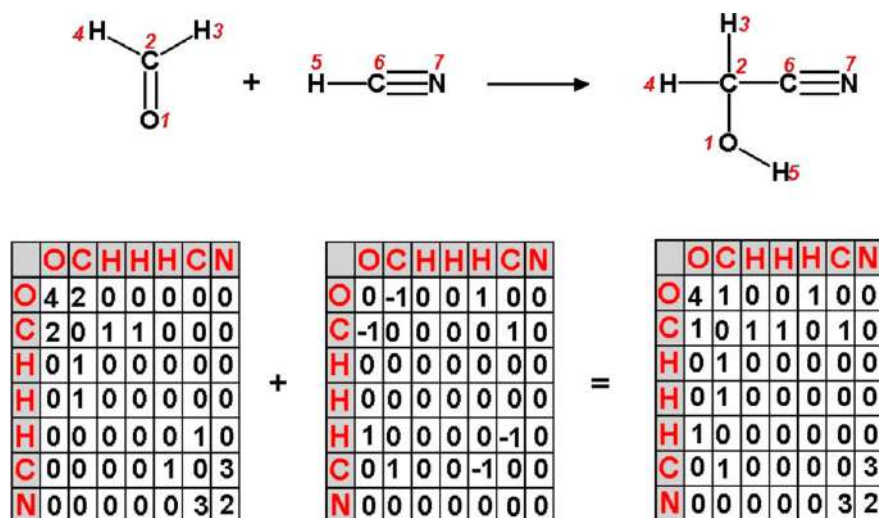


Fig. 2 Reaction coded by the $B + R = E$ matrix. Reproduced from Gasteiger, J., Engel, T., 2004. Chemoinformatics: A Textbook. Weinheim: Wiley-VCH, p. 186.

compounds. For example common reaction nomenclature often honors distinguished chemists, the discoverers of the certain reactions. This remind us the trivial chemical compounds nomenclature, because trivial name encodes no information on the reaction itself. Merck Index can serve as a dictionary guiding us through the name reaction chemistry (Smith *et al.*, 2001). An online alternative is the name reaction database of the Organic Chemistry Portal (Organic Chemistry Portal). Similarly to the nomenclature of chemical compounds IUPAC developed a systematic reaction classification capable of coding a variety of useful information on the molecular transformation; however, this is used only very seldom (Polanski, 2009b). Molecular transformations in silico can be coded by an algebraic model based on logical connectivity and matrix equation $B + R = E$, where B (beginning) represents an initial reaction stage, E (end) codes a final state and R is a reaction matrix. Fig. 2 illustrates an example of the reaction noted in such an approach (Gasteiger and Engel, 2004).

Other representations and classifications of chemical reaction, including SMARTS, has been developed but will not be discussed here further and a reader is referred to the references available in Gasteiger and Engel (2004), Chen (2003) or at the internet site of the Daylight Chemical Information System.

Organizing Chemical Facts Into Databases

Finally, what we need to enable chemists using computers to perform efficient chemistry is the access to chemical information, i.e., chemical data represented by chemical facts and numbers. Thus, for example, coding chemical transformation as described in section "Coding Chemical Reactions" does not bring information on this specific reaction gathered by experimental chemistry which are often fundamental for the chemists. Therefore, the additional data on the reaction conditions, solvents, temperatures, catalysts or byproducts have to be shaped to a form compatible with the computer platform. Accordingly, organizing chemical data in searchable databases is a focal point. A variety of chemical databases are available on-line having user-friendly interfaces. This includes chemical reaction or substances databases as Reaxys (formerly Beilstein) or Chemical Abstracts, patent databases as esp@cenet, chemical substance catalogues, e.g., Aldrich, or on-line chemical journals.

Managing interactions between a chemist and computer needs chemical information and software which puts together to a chemoinformatic platform capable of understanding chemical data efficiently assisting computer-aided chemistry or advising chemists in more traditional chemistry branches. Below we discussed some of typical problems of data processing and data output.

Computer Generated Chemical Names

Chemical structures can be nowadays easily transformed to chemical names. Chemical name generators typically are supported by a molecular editor. Historically, Beilstein pioneered in this field with the AutoNom program, which succeeded in 86.3%, if tested for more than 63,000 structures. Currently, AutoNom can generate both Beilstein or ACS nomenclature forms. Alternative software has been developed by Advanced Chemistry Development Inc. The ChemSketch freeware is a part of the extensive platform available at the ACD/Labs internet site (ACD Lab). Coding name generator was discussed in the Wisniewski (2003).

Molecular Modeling

Basically, what we mean by molecular modeling is simulating molecular structures in silico. This involves molecular manipulations such as visualizing, merging, superimposing or rotating individual molecules in space but also rotating bonds within individual molecules, etc.

More generally, molecular modeling involves the predictions of various molecular behaviors and properties. In particular, this includes predicting molecular shape, e.g., by 3D structure generation, simulation of chemical or biological effects. Modeling virtual molecular structures is not a trivial problem and includes a variety of computational schemes on the different level of approximation (Holtje *et al.*, 2003). The importance of molecular modeling can be understood better if we realized that chemistry often needs to investigate virtual or hypothetical structures, for example, active complexes that would be extremely unstable or even could have never been synthesized. In such cases a single possibility offering 3D molecular representation is computer modeling.

2D and 3D Structure Generators

Current methods often demand to analyze thousands or even millions of structures. Obviously, this cannot be realized by any system operated manually. Instead, we need the automated algorithms. Actually, nowadays this can be easily programmed in a variety of environments. Alternatively, a ready to use databases of virtual compounds are available (Reymond *et al.*, 2010). It is important to realize that chemists usually simplify a real structure of a chemical molecule to its molecular configuration. For example, the shape and stability of E and Z isomers which are two different configuration series are commonly estimated solely on their configuration type, irrespective of the actual 3D structure of the individual molecules. Such simplification is often sufficient, in particular, in organic chemistry where more detailed molecular representation will be even too complex and impractical. However, molecules are three dimensional and we can indicate the exact space location of individual atoms with high resolution. In some areas as medicinal chemistry such high resolution molecular representation can be of substantial importance. Traditionally, we use X-ray diffraction on crystals. However, as this effect is limited to the condensed matter (crystals), many further approaches appeared that allow to disclose various structural data defining 3D structure; however, general technology is still unavailable here. Despite an impressive progress X-ray crystallography is still a tedious process demanding producing crystals. Moreover, we can doubt a full correspondence between atom configuration in condensed matter vs. this in other environments. Accordingly, even nowadays only a small fraction of factual chemical space is described by the respective 3D structures despite a fact that such a data are sought after in particular for peptides or drug–ligand complexes (Motherwell, 2004). In other words 3D structures as measured properties are rare.

The importance of molecular modeling can be illustrated by a variety of descriptors and properties influenced by 3D molecular structure. Shape is an example of molecular descriptor that directly depend upon 3D structure. This is, however, a fuzzy category because molecules are capable of adopting different configurations and/or conformations (representing different shapes), depending upon the requirements of the environment. Accordingly, a population of the available shapes depends on the energy of the molecule. An uncertainty of this category explains a fuzzy nature of the properties and effects of various chemical or biological effects that are controlled by shape. This includes drug-receptor recognition and interactions. All shape effects can be currently modeled in silico (Polanski, 2003).

The generators of 3D structures are designed for a high-speed conversion of 2D into 3D molecular representations. Formally, this will be molecular descriptors, obtained by property prediction. How such descriptors can be calculated? Usually, X-ray structures are a kind of standard for experimental atomic 3D coordinates. 3D structure generators are programmed to use a model atom types for certain hybridizations to calculate standard bond length and bond angles. Typically, the strategy includes additional rules as preferred conformations of ring structures. As the speed and program performance is preferred, non-typical atom input should not crash the software. Instead, reasonable value would be expected.

LHASA is an early example of 3D generator designed to evaluate influence of steric hindrance on the reactivity of cyclohexane conformations. Model builders are another example of the program block capable of generating 3D atomic coordinates in an interactive mode. This allows to convert a molecular graph drew by a user at the program interface into a 3D molecular structure. Typically, the programs use built-in rules for the generation of the approximate structures having standard bond lengths, bond angles, torsion angles and stereochemistry. Usually generated structures require further refinement by geometry optimization. High-speed conversion of 2D to 3D representations of the large molecular populations are the main task of such programs. The examples of the software for automated structure data conversion includes CORINA, Cobra, Alcogen, Chem-X, Molgeo. Obvious priority here is a speed performance which can achieve more than 10^5 small to medium-sized molecules in ca. 2000 s on a 1.0 GHz workstation with a performance of 23 ms/cpd yielding a 99.99% conversion rate for the Corina program (Sadowski and Gasteiger, 1993). Optionally we can generate multiple ring conformers, rotamers or tautomers (ROTATE and STERGEN).

Modeling 3D-structures

Virtual molecular models in silico provides a useful platform widely used in nowadays chemistry. At the same time, theoretical chemistry developed a variety of methods aimed at modeling structure and bonding systems using a various molecular or quantum mechanics approaches which owe their origins to the Born-Oppenheimer (BO) approximation. In particular Schrödinger equation:

$$H\Psi = E\Psi$$

relating the wave function Ψ , Hamiltonian operator H , and energy E terms can be given in a simplified form. Formally, H includes components responsible for nuclear kinetic energy, electron kinetic energy, nuclear repulsion energy, electron repulsion energy, electron-nuclear attraction. In the BO approximation we assume that electron distribution depends on the fixed nuclear position only, which means that nuclear kinetic energy term can be neglected in the H operator. Introductory tutorials to molecular

modeling with representative references are widely available (Holtje *et al.*, 2003). A term *molecular mechanics* (MM) has been coined in 1970'. This refers to a so-called force field method (Hinchliffe, 2003) based on the assumption that a population of atoms in the molecule can be described by the potential energy depending upon the space locations of the atoms. This allows us for the optimization of the molecular geometry. The mechanics needed for the calculations of potential energy have been developed using classical physics. A suggestive model where atoms are balls and bonding are springs connecting the balls are commonly associated with this method in the chemical audience. Analytical functions are used to describe atomic interactions defining the so called force field. A variety of force fields have been developed to adopt the method for the different chemical functionalities and compounds' classes. Illustrative examples are MM+, AMBER, BIO+, OPLS. Via parameterization a force field concept can incorporate electronic energy in MM calculations. The MM method give us an insight into the molecular geometries and energies which minimization provides us the simulated low energy molecular model within the accessible molecular geometry space. Currently, a variety of MM software packages are available, e.g., HYPERCHEM, Sybyl including freeware options, e.g., the Tinker package. Informative tutorial and review of the methods and applications can be found in Goodman (1999), Keseru and Kolossvary (1999). Alternative online materials are offered by Rzepa.

Semiempirical quantum mechanical methods (SM) can be used for further approximation of the molecular models. This provides us with the deeper insight into the electron distribution within a molecule. Various levels of approximation are available in SM for the simulation of molecular orbitals. The calculations involve only valence electrons and are based on a set of experimental parameters forming a database referring to a certain set of chemical compounds. Such a strategy makes the calculations much easier. The examples are the AM1, MNDO, PM3, CNDO or INDO methods (Holtje *et al.*, 2003).

Molecular dynamics (MD) is an approach in which a single point atom location is modified to cover the dynamic atom behavior. In this model a nuclear system goes into a motion under a certain temperature forces. The motion is simulated by the Newtonian dynamic equations which solution indicates the set of possible atom locations. In particular, this allows to determine a so-called conformational ensemble profile for a molecule. Thermodynamic and dynamic properties of the molecules can be calculated using the MD method. The MD appeared especially useful for the simulations or refinement of protein shapes including X-ray structures. For further discussion of MD methods and applications compare Rapaport (2004). Many recent advances involve molecular dynamics simulations of proteins in explicit mixed solvents applied to various problems in protein biophysics and drug discovery including protein folding, protein surface characterization, fragment screening, allostery or druggability assessment (Kimura *et al.*, 2017). AQUA-DUCT: is an example of the up-to-date tool for the identification and tracking of molecules which enter active site cavity among thousands of single molecules along several thousand molecular dynamic steps (Magdziarz *et al.*, 2017).

Structure and Substructure Searches

Comparison is an important method in chemistry. In particular, this includes molecular structures. Therefore, structure or substructure searches is a substantial problem in a variety of the issues discussed in this article. An illustrative example is a need for the registration of novel chemical compounds and reaction within databases. A structure query is the essential operation for which molecular graph has to be translated to a canonical molecular representation. An example of a function that enable data structure organization is the so-called hash key. In turn, hash codes can be obtained from canonical SMILES strings or Augmented Connectivity Molecular Formula, e.g., in ACS Registry System. Additional information and references are available in Leach and Gillet (2003).

Structure query is the most simple database operation. Nowadays substructure search is also a routine. This provides us a method for the identification of the defined molecular fragment included in a series of molecules (Noordik, 2004). Graph theory, in particular the so-called subgraph isomorphism, can provide theoretical framework while practically a binary string molecular representation enables rapid structure screening (Leach and Gillet, 2003). The procedures developed include structural key (Boolean array) representation which can be a bitmap where each element coding a true or false refers to the presence or the absence of a certain structural feature or pattern. Alternatively, high-speed structure queries can involve fingerprint representations. In turn, a concept of molecular similarity measures enables the similarity based searches. For the essentials compare (Kochev *et al.*, 2004). Occasionally, for example, in drug discovery finding a common 3D pattern by mining 3D structure databases is an important problem. Database mining and identifying such a pattern known as a pharmacophore can assist a search for new ligands in the ligand based drug discovery paradigm (Nicklaus, 2003). Software available for the procedures discussed in this article can be found in the Kochev *et al.* (2004). Recently a concept of the so-called chemotype appeared. This represents molecules, chemical substructures and patterns, reaction rules, and reactions. This method is capable of integrating types of information beyond what is possible using current representation methods (e.g., SMARTS patterns) or reaction transformations (e.g., SMIRKS, reaction SMILES). For example new publicly available chemical query language, CSRML, to support chemotype representations for application to data mining and modeling has been developed recently by Yang *et al.* (2015).

Molecular Graphics

Formally, molecular graphics refers to a visualization of molecular objects, but this term is also used as a synonym of molecular modeling. Because molecular objects are complicated solving a chemical problem needs high speed and quality in virtual reality. Therefore, molecular graphics has significantly contributed to the currently available computer visualization technology.

Computer screen has been used for scientific visualization of physical models as early as in 1960s in the MIT during the Mathematics and Computation project. Developing technology for molecular visualization engages both chemistry and computer sciences. Currently, virtual chemistry on screen is a routine and highly interactive user-friendly systems are expected to be a part of each molecular platform. Molecular representations here can range from the atoms to the surfaces including a variety of others symbolic forms (Keil *et al.*, 2003). For a brief discussion on the differences between physical and virtual model compare Morris (2002).

Advanced Functionalities: In Silico Chemistry

Chemical Syntheses and Retro-Syntheses (Disconnections)

Chemical objects can be constructed via chemical synthesis. We can understand the importance of this problem in chemistry if we realize that even today chemical synthesis can be a bottleneck in research and a variety of applications. Chemists still believe chemical synthesis is an art. For the better illustration of this issue let us take a textbook example. Atropine, a natural product available from nightshade or belladonna was isolated and used already in ancient Rome and India. Of course the structure of this compound cannot be identified at that time. A core atropine moiety was proved to be tropinone in 1901 when Willstätter synthesized this compound. This needed however a complex more than 20 step synthesis which resulted in a total yield of no more than 1%. Low yield suggests inefficiency and makes easy to overlook the stunning success of this masterpiece of a synthetic work performed by a nobel prize winner. Even today this project would pose a complex organizational challenge, despite an availability of precise and rapid NMR or MS spectrometers easily and rapidly identifying the structure of compounds synthesized. True excitement comes however with the Robinson approach, Fig. 3. One step condensation and decarboxylation can result in more than 90% yield of tropinone (42% yield was originally reported).

It is still true, that the efficiency of synthesis critically depends on the skills and creativity of chemists. Can computers play a role in synthesis design providing us useful hints and supporting our skills?

The Development of Product to Reagents Strategy in Synthesis Design

Practical applications needs novel substances which are to be designed and produced for the industrial, pharmaceutical, agricultural use. This indicates the importance of a chemical product. Products are formed in chemical reactions; however, products appears at the very end of the process which starts from reagents. Therefore, in order to know how to design the products in a rational way we should have first explored a conversion of a variety of chemical compounds and functionalities. This explains how historical development of the chemical knowledge has imprinted into the organic chemistry textbooks where chemical reactivity of a molecule is a core issue. This decides also chemists are trained in a reagents to product chemistry where products can be designed basically only by a nondirect screening of possible reactions. However, an efficient general solution in such a strategy would need for example a database for the direct searches of products. This however, cannot be realized because a database registering all possible virtual products would be impracticable due to a fact that chemical space is too highly populated. Therefore, for years

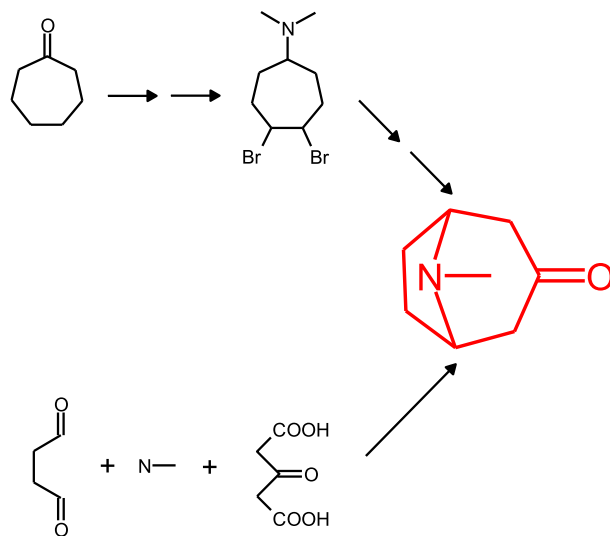


Fig. 3 Willstätter (upper) and Robinson (lower) approaches to the tropinone synthesis. Reproduced from Polanski, J., 2009. Chemoinformatics. In: Walczak, B., Tauler, R., Brown, S. (Eds.), *Comprehensive Chemometrics* vol. 4. Amsterdam: Elsevier, pp. 459–505.

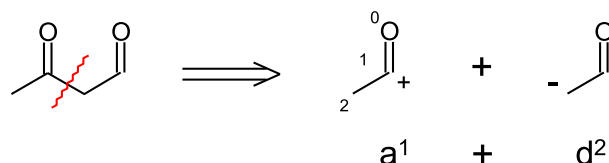


Fig. 4 A simple example of the disconnection to an acceptor a^1 and donor d^2 synthon. Reproduced from Polanski, J., 2009. Chemoinformatics. In: Walczak, B., Tauler, R., Brown, S. (Eds.), *Comprehensive Chemometrics* vol. 4. Amsterdam: Elsevier, pp. 459–505.

synthesis design were focused on a reagent to product method using a variety of nondirect approaches which can involve (Barone and Chanon, 2003):

- exploring synthetic availability of a substructure of the target molecule (TM), which can provide the major synthetic step opening the way to target molecule,
- an identification of chemical compound (reagent) similar to the target molecule. This allows to transform synthesis design to a problem of the conversion of the structure of this reagent to the structure of the target molecule,
- an identification of a substructure of the target molecule available as a natural product, if present that can be included as a ready-to-use fragment or so-called building block, e.g., an example can be chiral pool synthesis where building blocks are enantiopure substances.

Eventually, we know a bunch of serendipitous discoveries that provided us with the efficient reagents to product conversions.

A direct product to reagent strategy for designing synthesis has been developed by Corey and Cheng (1989). This allows to start synthesis design from a product, commonly referred as a target molecule (TM) or synthetic target. Operations called retro-synthesis, transform or disconnection identify retron or synthon representation of reagents. The meaning of retron and synthon is very close. Retron means a subunit structural pattern identified in the molecule of product designed as TM, provided that we can discover also a transformation representing (retro-reaction). Synthon represents any group of atom(s) indicated by disconnecting chemical bonds in TM in such a way that we can find real reagents representing these synthons and chemical reactions yielding the product called TM. Synthons are virtual chemical entities. Heterolytic bond disconnection results in the moiety having a potential for the electrophilic or nucleophilic behavior, coded by plus (+) or minus (−) signs indicating the acceptor or donor synthon types, respectively. In Fig. 4 we illustrated a simple example of the disconnection. Instead of clearly indicated virtual synthons in many advanced analyses we can find directly reagents, for example Corey did not use this term in his recent monograph (Corey and Cheng, 1989). Some essentials of synthon chemistry is discussed below. For further details a reader should compare Smit *et al.* (1998).

Synthon Nomenclature

Carbon chains formed of C-C and C-H bonds are the most common motifs in organic compounds. Although some reactions may be possible in such systems, in order to observe useful reactions we usually need to differentiate individual atoms by bonding to chemical elements other than C or H. The resulted reactivity differentiation is called chemoselectivity. For example oxygen or nitrogen if bonded to the carbon atom significantly modifies the reactivity of their chemical neighborhood. Therefore, molecular modules other than C-C and C-H are called functional groups or functionalities (FG). FG are of key importance in the contemporary synthetic chemistry approaches. In particular, the mode in which FG influences the reactivity of the neighboring hydrocarbon plays a role in designing potential chemical reactions of the systems. Accordingly, synthon atoms are numbered in relation to the location of the FG and a disconnected (usually carbon) atom (Fuhrhop and Penzlin, 1983) Fig. 4 illustrates a simple disconnection yielding a common carbonyl acceptor synthon a^1 and carbonyl donor synthon d^2 .

Operations on Synthons

Synthons formed in disconnections are virtual entities that need additional operations in order to indicate a corresponding reagents. Synthon to reagent transformation needs at least an addition of an ending at the terminal atom, but can also be much more complex. The basic assumption of organic chemistry is that a certain FG determines a certain reactivity for a series of chemical compounds, e.g., aldehydes react with alcohols to form acetals. A concept of synthons allows us further generalization. Synthon chemistry indicates the reactivity type, comprising various FGs. Some common operations beyond synthetic procedures includes modifications to control synthon reactivity (in blocking, activating or umpolung, i.e., activity reversal mode). A brief introduction to these topics can be found in Polanski (2009b), Smit *et al.* (1998).

Computer-Assisted Synthesis Design

Total synthesis is a term that describes a synthesis yielding natural products. Mimicking Nature and designing synthesis of natural products demands experience, talent and art. Longifolene is an illustrative example of a first disconnection noted from product to

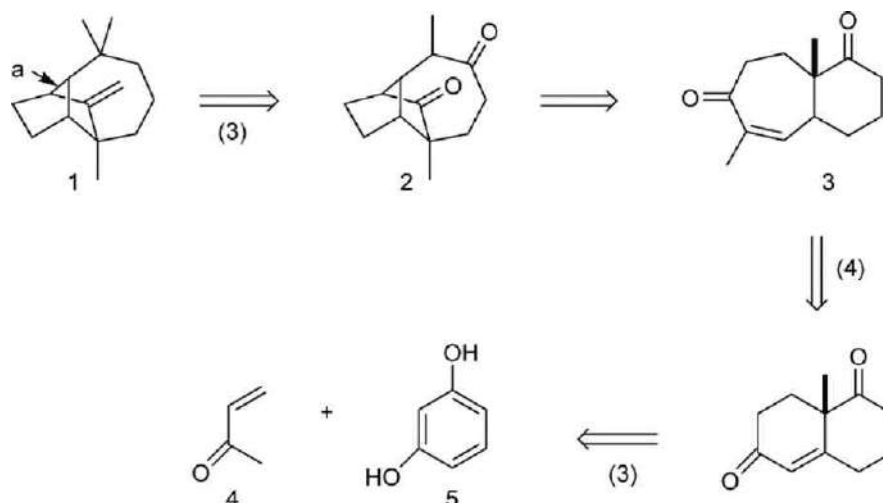


Fig. 5 First disconnections by Corey. Arrows indicate the direction from product to reagent. Modified from Polanski, J., 2009. Chemoinformatics. In: Walczak, B., Tauler, R., Brown, S. (Eds.), *Comprehensive Chemometrics* vol. 4. Amsterdam: Elsevier, pp. 459–505.

reagent scheme (Fig. 5) (Corey and Cheng, 1989; Corey, 1993). Precisely the scheme designed by Corey in 1957 refers not to synthons but directly to reagents. The application of computers in the CASD method was also pioneered by Corey. As usually a single disconnection of TM does not solve the problem but the first level reactants must be disconnected further and further to the next level reagents, forming the so-called synthesis tree. Eventually, the availability of the reactants indicates a moment for the last level disconnection. Practically a full disconnection architecture appeared to be an extremely complex computational problem due to an expansion of a number of possible routes analyzed. For example, medium complexity synthesis can involve 10 level synthesis tree in which a number of precursors can exceed a thousand. The expansion of synthetic tree needs to be pruned (Barone and Chanon, 2003).

The strategy of CASD can be compared to the chess strategy where relatively simple laws result in a complex architecture of decision-making where human intuition and experience sometime can find optimal solutions at a first glance (Todd, 2005). Various successful paths are possible; however, the strategy and further paths are strongly limited by the early choices. The expansion of the decision-making tree usually needs pruning, usually by rules of thumb. Both games often rewards a gambit strategy. In chess we can open a game sacrificing a chess piece for the strategic advantages gained. In turn, an unexpected disconnection that breaks common chemical rules may appear a winning strategy. Basically, the CASD software is programmed to find synthons. An important part is also the generation of precursors and sketching synthetic route that corresponds to the decided disconnection tree. A first CASD software, Logic and Heuristics Applied to Synthetic Analysis (LHASA), was programmed in the late 1960' by Corey group (LHASA). Searching for disconnections was coded here by CHemistry TRaNslator (CHMTRN), chemical language constructed exceptionally for this purpose (Ott, 2004). In turn, *Chematica* is a recent retrosynthetic system based on the complete chemical information available (Szymkuc *et al.*, 2016). Accordingly, seven million chemicals connected by the number of reactions and catalogues 86,000 chemical rules. Cost-cutting measure is an important indication controlling synthesis paths, shorter and more economical paths are preferred, which nicely illustrates the scope of chemoinformatics (Fig. 1). A variety of other retrosynthetic software is currently available.

Computer Assisted Structure Elucidation

Chemical syntheses are designed to provide chemical compounds which structures usually are to be proved by analytical procedures and/or spectroscopic methods (MS, IR NMR) which use nowadays is a routine in modern chemistry. Chemistry as a soft science relatively early took advantage from the artificial intelligence methods which were adopted by chemometrics. This allowed us to develop an efficient Computer Assisted Structure Elucidation which engages both the domains of organic and analytical chemistry (Steinbeck, 2003).

Computer Assisted Knowledge Discovery

Currently, a database is an efficient replacement for a printed version of documentation that enables registration and identifications of chemical compounds. The database system can also be used for the extensive investigations and analyses of chemical data working in a teacher-like mode, modifying, falsifying, or improving existing models, or in other words extending our chemical knowledge. Such data extraction and analysis by database mining is nontrivial and can be an attractive tool giving us a novel and fresh insight into the architecture of chemistry in the approach that is referred to as the discovery of knowledge (Frawley *et al.*, 1991).

Typically, chemical databases equipped with molecular editor and the efficient user friendly searching machine allow us the advanced data queries of the complex syntax. The management of traditional printed information system cannot be any longer a competition for such a technology. Therefore, the last update of the printed BH version appeared in 1998. The printed handbook that has been developed as a breaking innovation in the 19th century science, has been finally overpowered by in silico technologies. Data mining enabled Computer Assisted Knowledge Discovery bringing a substantial improvement into chemical investigations. chemical data can be efficiently accessed and processed all together in a real time. A question appears if this has significantly influenced chemistry. The increased performance is the first issue. But can we really discover chemical laws on the basis of database mining? The answer is positive. To show several examples. The analysis of molecular weight MW of chemical compounds registered in the Beilstein database indicated for example the mean MW of the most commonly used substrates are those near 150 g/mol and the most common products near 250 g/mol. Interestingly this values did not changed between 1850 and 2004. In turn, the distribution of MW for druggable substances and this for all registered compounds are virtually identical (Fialkowski *et al.*, 2005). An *architecture of organic chemistry* was further outlined by Rucker and Meringer who described the difference between theoretical graphs and the graphs mapped by real chemical compounds registered in Beilstein (Rucker and Meringer, 2002). Technically, self-organizing neural network was used for knowledge discovery in the reaction databases (Chen and Gasteiger, 1997). For review in this area a reader can compare Ester and Sander (2000).

Biological data at unprecedented tera- and petabyte scales were generated recently and the integrated computational platforms (D'Souza *et al.*, 2017) or new visualization tools (Tsiolaki *et al.*, 2017) have been developed for the efficient data mining, accordingly.

Chemometrics – Adopting Mathematics to Chemistry and Chemistry to Mathematics

The origins of chemometrics are in analytical chemistry where the first applications of computers basically engaged the problems which were more of the mathematical than strictly chemical background. Clearly, this application is available on the relatively early stage of the development of computer science. The experience we got by chemometrics can be then applied to fully molecular problems which originated a domain of chemoinformatics. Chemometrics helped to increase the potential of informatics for in silico chemistry expanding at the same time the scope of interest of chemical computer simulations. Historically, chemometrics focuses on mathematics, including statistics, programming and so on (Wold, 1995). This explains why we use a term chemometrical analysis as a synonym of the application of the advanced statistical methods for the extraction of chemical knowledge from chemical data (Myshkin and Wang, 2003; Pierce *et al.*, 2005; Pytela *et al.*, 1994) or even in the more narrow sense as PCA or PLS analyses (Rodriguez-Barrios and Gago, 2004). Mathematics forms a common language generally understood by science. However, each science branch has its specific language. Accordingly, an efficient translator support can significantly enhance the quality and understanding of mathematics in chemistry. At the same time chemical problems can be formulated in a form suitable for mathematical applications.

Clearly, chemoinformatics is a broader term than chemometrics and sometime chemometrics is included here as a sub-discipline. Usually, however, chemometricians prefer to form an autonomous science.

Computer Assisted Molecular Design

With the increasing number of chemical compounds the kind of inflation of new compounds have appeared. Therefore, we need to indicate and design chemical compounds that would be interesting objects of investigations. The architecture of organic chemistry indicates chemical space interesting for chemists (Fialkowski *et al.*, 2005). Actually, a term molecular design focuses on two different rationale categories. First is a design of any compound or in other words synthesis design discussed in article Chemical Syntheses and Retro-Syntheses. Second design rationale relates more to property than structural design. Chemistry contributes to industry constructing a variety of chemicals of the crucial importance for a nowadays civilization such as drugs, preservatives, flavors, etc. However, individual chemical molecules having a certain property can be integrated with different chemical structures. Accordingly, we can arranged atoms in the molecules into a variety of combinations of the different compounds hopefully possessing individual properties as desired. In practice, the available molecular configurations even today depend upon synthetic capability of individual laboratories and current chemical technology. This mean molecular design is to some extent limited by synthetic capability. Are there any simple rules indicating a formation of useful chemical compounds? Surprisingly, some chemical frameworks are suggested to be preferred patterns in the drug molecules (Fattori, 2004; Kubinyi, 2006; Schneider and Schneider, 2017).

From Data to Drug Candidates and Drugs

Basically, a molecule of drug is a man-made moiety designed or discovered in other way to fit the biological counterpart and produce the required action. Accordingly, drugs are to manipulate biological effects generated by macromolecular proteins designated as targets or receptors (a precise definition of the receptor can be found in Cohen, 1996). Thus, the drug-receptor interactions define a potential activity. Currently we attempt to simulate these interactions in silico. Drug design or molecular design are two terms used as synonyms to refer to in silico technology in this area. This duality nicely illustrates the difference

between a molecule and a substance (drug). A term drug design scores high on optimism because currently we do not have an efficient technology to virtually construct a real drug in a single step engineering like process that would resemble for example the design of a bridge or a building. Therefore, precisely we can define molecular design as a method attempting to find *drug candidates* on the basis of computational techniques. Drug discovery or drug development are broader categories referring to the investigations (both in silico and in vitro) of drug candidates as potential drugs. Basically drugs can be developed by target based or phenotypic strategies (Swinney and Anthony, 2011). The analysis of the economic efficiency of current drug design R&D efficiency is pessimistic. The decreasing financial success is revealed and described in the so-called backward Moors (Erooms) law (Scannell *et al.*, 2012). Big pharma is driven by innovations. This means new drugs are needed for new patents capable of bringing new incomes. Therefore, companies are forced to fierce R&D search. At the same time regulations decided that development strategy was turned to target oriented approaches which, apparently, appeared too optimistic. Stiff quality and safety requirements and organizational factors could also be a success barrier here (Leeson and St-Gallay, 2011). On the other hand there are good examples of the successful drug design (Borman, 2005) and drugs get better and better imitating natural biological effectors.

Structure Based Design

What we mean by structure based design is a method that is based on the known receptor structure. Basically, two basic approaches of docking and de-novo design are available to fill of a receptor with a ligand structure (Schneider and Fechner, 2005). By docking we understand fitting the ligand structure into the receptor cavity (Kitchen *et al.*, 2004). Available programs are specified in Warren *et al.* (2006). In turn, in de-novo design a potential receptor ligand is constructed directly in the receptor cavity from the molecular fragments or even individual atoms. Available programs and illustrative examples are given in Warren *et al.* (2006), Kirkpatrick (2005). Docking protocols were used for High Throughput Screening (HTS) mode of in silico searching for the receptor ligands. The Intel-United Devices Cancer Research Project is here an example of internet distributed computing in which 3.5 billion molecules were investigated as potential anticancer drugs in the simulation performed by downloadable screensaver (Richards, 2002).

Recent advances in technology and novel approaches to drug discovery are now available in the context of the description of the target structure. This includes *structure, dynamics, mutational activation and inactivation, and signaling mechanisms*. An illustrative example can be small-molecule inhibitors of KRAS (Ostrem and Shokat, 2016). *Mechanism based design* is a term used as a formal term referring to such methods.

Ligand Based Design

Often not enough data for target receptor is available. In such a case the ligands that are known forms the basis for the comparison of the new molecular representations. This approach is known as a *ligand based design*. Accordingly, this method is based on the nondirect investigation of the ligand-target interactions or in other words *receptor or pharmacophore mapping* where a pharmacophore refers to the receptor or receptor sector model *reconstructed* from a series of its ligands. Depending upon individual method, pharmacophores or pseudoreceptors are usually assembled from virtual elements which can be represented by various functionalities, interaction types, or even fragments of molecules, e.g., aminoacids, atom or atom types. Actually, generally only very rarely any relation between a real receptor and a pharmacophore exists. Extensive review of the methods is available for example in Cohen (1996), Horvath *et al.* (2005).

Mapping Structure to Property in QSAR Approach

Quantitative structure activity relationship (QSAR) is a strategy of the essential importance for chemistry and pharmacy, based on the idea that when we change a structure of a molecule then also the activity or property of the substance will be modified. Structure modification can result from the virtual in silico operations, intentional in vitro projects (usually synthetic research) or we can just investigate available substances. The importance of the QSAR concept can be recognized by a volume of the literature and variants of the methods described. A brief introduction to this area can be found in Polanski (2009a,b), while current state of QSAR is thoroughly reviewed by Cherkasov *et al.* (2014). Basically, in QSAR we map chemical space to property space modeling a function relating property to chemical structure or more accurately to the molecular descriptors that are related to this chemical structure. This function should work like a dictionary between two spaces hopefully predicting the activity of novel compounds of the desired properties. Theoretically, these molecules can be then synthesized and tested as drug candidates while iterative QSAR modeling will further optimize second level synthetic targets and the property itself. Because a QSAR function relates property to structure, this is an indirect design of the molecular representations having this property (Polanski *et al.*, 2006).

Multidimensionality of QSAR (0-7D) relates to a complexity of the ligand or ligand-receptor data coded by molecular descriptors used during modeling. In the majority of examples QSARs are modeled without receptor data (receptor independent mode) but receptor dependent QSARs are also possible (Polanski, 2009a). In the majority of described applications QSAR realizes a strategy from molecules to property, however, an inverse strategy from property to molecules can also be hypothesized

(De Julian-Ortiz, 2000). According to the scope of chemoinformatics (Fig. 1) analyzing and/or modeling economic properties are hot issues that could significantly contribute to molecular design. Accordingly, the analysis of big data from molecular market was performed to obtain the first quantitative structure economy relationship (Polanski *et al.*, 2016).

Drug-Likeness and Druggability Concept

Data dependency and uncertainty is an immanent part of QSAR modeling (Polanski, 2009a; Polanski *et al.*, 2006) and we still do not have a technology that would be capable of designing in a single act a molecule having properties desired. Instead, molecular design is a method that improve the success ratio in a search for new drug candidates. Therefore, this kind of design prefers rather general than rigorous rules. The example can be the so-called drug-likeness concept insisting that some common features or even privileged substructures can be indicated in drugs. The Lipinski rule of five lists here molecular weight <500 , $\log P < 5$, a number of hydrogen bond acceptors <10 , and a number of hydrogen bond donors <5 that rule advantageous drug-likeness for the orally available drugs (Lipinski *et al.*, 1997). Precisely, the Lipinski rule focuses on molecular descriptors. In turn, the ADMET concept shifted our interest to the properties, in particular, these deciding compounds bioavailability and/or biocompatibility, i.e., Adsorption, Distribution, Metabolism, Elimination and Toxicology (Davis and Riley, 2004; Hodgson, 2001; Van de Waterbeemd, Gifford, 2003). Finally, a concept of privileged structures appeared. For example, it has been discovered that a “group of 32 common shapes or frameworks accounted for 50% of the 5120 molecules considered. Whether these fragments had intrinsic characteristics that gave them drug-like properties or their presence was a result of chemists’ habits, familiarities or synthetic versatility was an issue that was recognized but not addressed” (Fattori, 2004). NCI DTP AIDS Antiviral Screen database (40,000 compounds) was searched for the common frameworks and molecular fragment distribution typical for anti-HIV activity (Helma *et al.*, 2002). The related approach allowed Oprea to cluster more than 12,000 compounds into different activity levels (Oprea, 2004). Some privileged drug-like moieties can be found in Kubinyi (2006). Eventually, the comparative or combined QSAR databases appeared in a hope for identifying more molecular drug-likeness rules (Hansch *et al.*, 2002; Oprea, 2002). Shen *et al.* developed a concept of “the application of predictive QSAR as a virtual screening tool for database mining” (Shen *et al.*, 2004). Protein and gene expression data can be processed similarly to molecular representations (Helma *et al.*, 2002). Accordingly, druggability relates to the selected receptors that provide the privileged fit to drug-like molecules while we can look for druggable genomes expressing druggable proteoms (Lipinski and Hopkins, 2004).

Bioinformatics in Drug Design

In bioinformatics we focuses of the biological space rather than on the chemical one. Formally, we are mapping here the biological to chemical space. Pharmacogenomics is a concept that by the analysis of genomic data we can obtain novel drugs. Novel data measured form DNA microarray gene expression experiments pose new computational and processing challenges that formed bioinformatics to an important interdisciplinary research.

The relation of bioinformatics and chemoinformatics resembles this of chemoinformatics and chemometrics. Accordingly, chemoinformatics can be interpreted as a part of bioinformatics.

Internet Resources for Chemistry and Chemoinformatics

A first cable computer connection and networking took place in 1960’ in the United States. Nobody could have expected the expansion of social interaction by Internet. Nowadays, the impact of Internet on the economy and science is clear. Web technologies are more and more common in computer science and chemoinformatics. A number of chemical resources available in the web steadily increases. First of all, we can indicate a number of sites with chemical data. A variety of other web resources include educational materials, software (free or commercial), on-line chemistry journals and many others. E-commerce is also a common way of the distribution of chemicals. The enumeration of all available web sites that are of the interest for chemoinformatics is not possible any-longer in such a brief review. The addition to Wikipedia and Google and the enormous wide SMILES presence in web technologies are examples of the success of chemoinformatics.

Current Practices and Capabilities

Basically, the methods of chemoinformatics have been formed in 1990’ and early 2000’. Accordingly, a four volume Handbook of Chemoinformatics edited by Gasteiger (2003) is still the most comprehensive up-to-date introduction to the topic. Generally, what we observe recently is a move from method development to the efficient applications and vigorous software development. On one hand chemical informatics was moved into public domain. On the other hand still a lot of payable services are under development. In this context current practices follow general directions of computer related

technology development. A lot of the services are available. A mobile CAS service can be an illustrative example here. Chemoinformatics has answered the expectations and needs of pharma for new methods in computer-aided molecular design. Drug discovery and development is still the most important research field. Organization and utilization of drug discovery data involving small molecules, analysis of structure-activity relationships, and prediction of active compounds have remained hallmarks (Bajorath, 2015). Much higher computation capability, significantly higher and less expensive storage and memory provided significantly better opportunities for the information storage and processing. Docking is routinely used to identify potential hits from large compound libraries and Zanamivir and oseltamivir (Tamiflu) are examples of anti-influenza drugs developed in structure-based strategies which combined computer-based approaches, such as docking and pharmacophore-based virtual screening with X-ray crystallographic structural analyses (Mallipeddi *et al.*, 2014). Big data is a good example illustrating a level of the development of chemo and info counterparts of chemoinformatics. We need a proper structure of the data, in particular, the measured properties should be reliable, precise and extend enough to describe the complexity of the problem. For example, we usually analyze drug candidate action by the activity value versus a single target. In fact, however, a xenobiotic in an organism can induce a significant changes in signaling of various targets. This concept is known as polypharmacology. In this context, the polypharmacology and lipidomics inspired scheme of HTS data screening has been developed recently by Gabriele Cruciani group (Goracci *et al.*, 2017). The method is limited by an inexpensive and rapid procedure for the extraction and MS based determination of a medium size population of lipid species that can characterize the complex answer of any organism for a drug (drug candidate) administration. In 20 min this provides us a data on the serum concentration of 1000 different lipids in serum. Such a data can be arranged in a fingerprint-like lipid profile which if tested as a function of time, can give us a real information on the activity and toxicity of the xenobiotics used. This illustrated that more efficient big data methods are limited not necessarily by the statistical issues but by new technology in inexpensive property measurements (Polanski and Gasteiger, 2016). New HTS methods for property measurements should undoubtedly bring new quality in big data analyses in drug design. Another recent big data study is an attempt to model the Alzheimer disease. The extend experimental property measurement has been referred here as *smart data* approach (Geerts *et al.*, 2016). Femto or attosecond serial X-ray protein nanocrystallography which generates as much as 3 million. X-ray patterns (measured 3D structure data) in seconds (Chapman *et al.*, 2011) is another illustrative example.

Closing Remarks and Future Directions

We can interpret a formation of chemoinformatics as an answer to a fascination of the potential of in silico chemistry but also to the disappointment of computer assisted drug design in early 90'. A need to teach computers the essentials of chemistry not only provided us with new tools but also allowed us for better understanding of chemistry. Currently chemoinformatics is intelligent and flexible enough to adopt the uncertainty of fuzzy data produced by in vitro chemistry. In this context chemical ontology is an example of the trend which will probably be followed in silico. Molecular design and drug design are two terms used for the description of the main objective of chemoinformatics. This nicely illustrates the dichotomy between molecules and substances forming a class of chemical compounds. While molecules are represented mainly by molecular descriptors, properties represent substances. A flood of molecular descriptors is accompanied by property deficit. Properties are rare because their measurements are expensive. This means property predictions are often required. Property deficit explains low success ratio of drug design. In fact, the basic objective of chemoinformatics is a design of drug candidates by the search of advantageous molecular descriptors on the basis of few properties available. Then, we hope some of candidates will appear successful. Hopefully, with the physicalization of chemistry, properties will appear more available making drug candidates better. The example are *omics* methods which provided us with new technologies providing more and more property data. Big pharma is driven by innovation. To develop innovation capable of winning on the market these data should be cheap and capable for rapid measurements. Basically, we know how to efficiently transform the data into chemical knowledge and further into drug candidates. With the increasing availability of properties better quality of drug candidates will increase the success ratio in developing new drugs. New in silico methods are appearing and still are to appear with wider availability of the property data to efficiently treat this kind of data.

Nowdays Internet is a philosophy of the informatics, in particular, chemoinformatics. Chemistry depended on data and the web has conserved this dependency making the data broadly available. Our addiction to the Internet will definitely increase, stimulating also further cooperative development of the discipline.

Eventually, this text is a significantly shortened but at the same time revised and upgraded version of the article in the Elsevier Encyclopedia of Comprehensive Chemometrics.

See also: Binding Site Comparison – Software and Applications. Chemical Similarity and Substructure Searches. Drug Repurposing and Multi-Target Therapies. Fingerprints and Pharmacophores. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Public Chemical Databases. Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications. Rational Structure-Based Drug Design. Small Molecule Drug Design

References

- Audibert, P., 2013. Mathematics for Informatics and Computer Science. Hoboken, London: Wiley-ISTE.
- Bajorath, J., 2015. Entering new publication territory in chemoinformatics and chemical information science. *F1000Research* 4, 35.
- Baldi, P., 2005. Chemoinformatics, drug design, and systems biology. *Genome Inform.* 16, 281–285.
- Barone, R., Chanon, M., 2003. Computer-assisted synthesis design. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 1428–1456.
- Bensaude-Vincent, B., Simon, J., 2012. *Chemistry – The Impure Science*. Rosewood Drive: Imperial College Press.
- Bobach, C., Boehme, T., Laube, U., Puschel, A., Weber, L., 2012. Automated compound classification using a chemical ontology. *J. Cheminf.* 4 (40), 1–12.
- Bock, G., Goode, J.A., 2002. *Silico' Simulation of Biological Processes*. Chichester: John Wiley and Sons.
- Bohacek, R.S., McMartin, C., Guida, W.C., 1996. The art and practice of structure-based drug design: A molecular modelling perspective. *Med. Res. Rev.* 16, 3–50.
- Borman, S., 2005. Drugs by design. *Chem. Eng. News* 83, 28–30.
- Brock, W.H., 1993. *The Fontana History of Chemistry*. London: Fontana Press.
- Brown, F., 1998. Chemoinformatics: What is it and how does it impact drug discovery. *Annu. Rep. Med. Chem.* 33, 375–384.
- Chapman, H.N., Fromme, P., Barty, A., *et al.*, 2011. Femtosecond X-ray protein nanocrystallography. *Nature* 470, 73–77.
- Chen, L., 2003. Reaction classification and knowledge acquisition. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 348–388.
- Chen, L.R., Gasteiger, J., 1997. Knowledge discovery in reaction databases: Landscaping organic reactions by a self-organizing neural network. *J. Am. Chem. Soc.* 119, 4033–4042.
- Cherkasov, A., Muratov, E.N., Fourches, D., *et al.*, 2014. QSAR modeling: Where have you been? Where are you going to? *J. Med. Chem.* 57, 4977–5010.
- Cohen, J., Stewart, J., 1994. *The Collapse of Chaos: Discovering Simplicity in a Complex World*. New York: Viking.
- Cohen, N.C., 1996. *Guidebook on Molecular Modelling in Drug Design*. San Diego: Academic Press.
- Consonni, V., Todeschini, R., 2010. Molecular descriptors. In: Puzyn, T., *et al.* (Eds.), *Recent Advances in QSAR Studies*. Amsterdam: Springer, pp. 29–102.
- Corey, E.J., 1993. The logic of chemical synthesis: Multistep synthesis of complex Carbogenic molecules. In: Malmstrom, B.G. (Ed.), *Nobel Lectures in Chemistry 1981–1990*. Singapore: World Scientific, pp. 686–708.
- Corey, E.J., Cheng, X.-M., 1989. *The Logic of Chemical Synthesis*. New York: Wiley.
- De Julian-Ortiz, J., 2000. Virtual Darwinian drug design: Qsar inverse problem. *Comb. Chem. High Throughput Screening* 4, 295–310.
- D'Souza, M., Sulakhe, D., Wang, S., *et al.*, 2017. Strategic integration of multiple bioinformatics resources for system level analysis of biological networks. *Methods Mol. Biol.* 1613, 85–99.
- Davis, A.M., Riley, R.J., 2004. Predictive ADMET studies, the challenges and the opportunities. *Curr. Opin. Chem. Biol.* 8, 378–386.
- Ester, M., Sander, J., 2000. *Knowledge discovery in databases Techniken und Anwendungen*. Berlin: Springer.
- Fattori, D.D., 2004. Molecular recognition: The fragment approach in lead generation. *Drug Discovery Today* 9, 229–238.
- Fialkowski, M., Bishop, K.J., Chubukov, V.A., Campbell, C.J., Grzybowski, B.A., 2005. Architecture and evolution of organic chemistry. *Angew. Chem. Int. Ed. Engl.* 44, 7263–7269.
- Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C., 1991. Knowledge discovery in databases: An overview. In: Piatetsky-Shapiro, G., Frawley, W.J. (Eds.), *Knowledge Discovery in Databases*. Cambridge, MA: AAAI Press/MIT Press, pp. 1–30.
- Fuhrhop, J., Penzlin, G., 1983. *Organic Synthesis Concepts, Methods, Starting Materials*. Weinheim: VCH.
- Gasteiger, J., Engel, T., 2004. *Chemoinformatics a Textbook*. Weinheim: Wiley-VCH.
- Geerts, H., Dacks, P.A., Devanarayan, V., Haas, M., Khachaturian, Z., *et al.*, 2016. Big data to smart data in Alzheimer's disease: The brain health modeling initiative to foster actionable knowledge. *Alzheimers Dement.* 12, 1014–1021.
- Goodman, J., 1999. *Chemical Applications of Molecular Modeling*. London: Royal Society of Chemistry.
- Goracci, L., Tortorella, S., Tiberi, P., *et al.*, 2017. Lipostar, a comprehensive platform-neutral cheminformatics tool for lipidomics. *Anal. Chem.* 89, 6257–6264.
- Hansch, C., Hoekman, D., Leo, A., Weininger, D., Selassie, C., 2002. Chembioinformatics: Comparative QSAR at the interface between chemistry and biology. *Chem. Rev.* 102, 783–812.
- Hastings, J., Adams, N., Ennis, M., Hull, D., Steinbeck, C., 2011. Chemical ontologies: What are they, what are they for and what are the challenges. *J. Cheminf.* 3 (Suppl 1), O4.
- Helma, C.H., Kramer, S., De Raedt, L., 2002. The molecular feature miner MOLFEA. In: M.G. Hicks, M.C., Kettner, C. (Eds.), *Molecular Informatics: Confronting Complexity, Proceedings of the Beilstein-Institut Workshop*, Bozen, pp. 1–15.
- Hinchliffe, A., 2003. *Molecular Modelling for Beginners*. Chichester: Wiley.
- Hodgson, J., 2001. ADMET—Turning Chemicals Into Drugs. *Nat. Biotechnol.* 19, 722–726.
- Holtje, H.-D., Sippl, W., Rognan, D., Folkers, G., 2003. *Molecular Modeling*. Weinheim: Wiley-VCH.
- Hopkins, A.L., Keseru, G.M., Leeson, P.D., Rees, D.C., Reynolds, C.H., 2014. The role of ligand efficiency metrics in drug discovery. *Nat Rev Drug Discov.* 13, 105–121.
- Horvath, D., Mao, B., Gozalbes, R., Barbosa, F., Rogalski, S.L., 2005. Limitations Strengths and of pharmacophore-based virtual screening. In: T.I. Oprea (Ed.), *Chemoinformatics in Drug Discovery*. In: Mannhold, R., Kubinyi, H., Timmerman, H. (Eds.), vol. 23, *Methods and Principles in Medicinal Chemistry*. Weinheim: Wiley-VCH, pp. 117–140.
- Jager, W., Krebs, H.-J. (Eds.), 2003. *Mathematics-Key Technology for the Future*. Berlin: Springer.
- Keil, M., Borosch, T., Exner, T.E., Brinkman, J., 2003. Computer visualization of molecular models tools for man-machine communication in molecular science. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 320–344.
- Keseru, G., Kolossvary, I., 1999. *Molecular Mechanics and Conformational Analysis in Drug Design*. Oxford: Blackwell Publishing.
- Kimura, S.R., Hu, H.P., Ruvinsky, A.M., Sherman, W., Favia, A.D., 2017. Deciphering cryptic binding sites on proteins by mixed-solvent molecular dynamics. *J. Chem. Inf. Model.* 57, 388–1401.
- Kirkpatrick, P., 2005. Computational chemistry: Docking on trial. *Nat. Rev. Drug Discov.* 4, 813.
- Kitchen, D.B., Decornez, H., Furr, J.R., Bajorath, J., 2004. Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nat. Rev. Drug Discov.* 3, 935–949.
- Kochev, N., Monev, V., Bangov, I., 2004. Searching chemical structures. In: Gasteiger, J., Engel, T. (Eds.), *Chemoinformatics a Textbook*. Weinheim: Wiley-VCH, pp. 291–318.
- Kowalski, B.R., Bender, C.F., 1975. Solving chemical problems with pattern recognition. *Naturwissenschaften* 62, 10–14.
- Kubinyi, H., 2006. Privileged structures and analogue-based drug discovery. In: Fischer, J., Ganellin, C.R. (Eds.), *Analogue-Based Drug Discovery*. Weinheim: Wiley-VCH, pp. 53–65.
- Leach, A.R., Gillet, V.J., 2003. *An Introduction to Chemoinformatics*. Dordrecht: Kluwer.
- Leeson, P.D., St-Gallay, S.A., 2011. The influence of the 'organizational factor' on compound quality in drug discovery. *Nat. Rev. Drug Discovery* 10, 749–765.
- Lipinski, C., Hopkins, A., 2004. Navigating chemical space for biology and medicine. *Nature* 432, 855–861.
- Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney, P.J., 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* 23, 3–25.

- Magdziar, T., Mitusinska, K., Goldowska, S., *et al.*, 2017. AQUA-DUCT: A ligands tracking tool. *Bioinformatics* 33, 2045–2046.
- Maheshwari, A., 2014. *Data Analytics Made Accessible*, Kindle edition Amazon.
- Mallipeddi, P.L., Kumar, G., White, S.W., Webb, T.R., 2014. Recent advances in computer-aided drug design as applied to anti-influenza drug discovery. *Curr. Top. Med. Chem.* 14, 875–889.
- Morris, P.J.T., 2002. *From Classical to Modern Chemistry*. London: Royal Society of Chemistry.
- Motherwell, S., 2004. Chemoinformatics and crystallography. The Cambridge structural database. In: Noordik, J.H. (Ed.), *Chemoinformatics Developments, History, Reviews and Current Research*. Amsterdam: IOS Press, pp. 37–68.
- Myshkin, E., Wang, B.J., 2003. Chemometrical classification of Ephrin ligands and Eph Kinases using GRID/CPCA approach. *J. Chem. Inf. Comput. Sci.* 43, 1004–1010.
- Nicklaus, M.C., 2003. Pharmacophore and drug discovery. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 1687–1711.
- Noordik, J.H., 2004. *Chemoinformatics Developments*. Amsterdam: IOS Press.
- Oprea, T.I., 2003. Chemoinformatics and the quest for leads in drug discovery. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*, pp. 1509–1531.
- Oprea, T., 2002. Current trends in lead discovery. Are we looking for the appropriate properties? *J. Comput. -Aided Mol. Des.* 16, 325–334.
- Oprea, T., 2004. 3D-QSAR modeling in drug design. In: Tolleneare, J., De Winter, H., Langenaeker, W., Bultinck, P. (Eds.), *Computational Medicinal Chemistry for Drug Discovery*. New York: Marcel Dekker, pp. 571–616.
- Ostrem, J.M., Shokat, K.M., 2016. Direct small-molecule inhibitors of KRAS: From structural insights to mechanism-based design. *Nat. Rev. Drug Discov.* 15, 771–785.
- Ott, M.A., 2004. Chemoinformatics and organic chemistry. Computer assisted synthetic analysis. In: Noordik, J.H. (Ed.), *Chemoinformatics Developments, History, Reviews and Current Research*. Amsterdam: IOS Press, pp. 83–110.
- Pierce, K.M., Wood, L.F., Wright, B.W., Synovec, R.E.A., 2005. Comprehensive two-dimensional retention time alignment algorithm to enhance chemometric analysis of comprehensive two-dimensional separation data. *Anal. Chem.* 77, 7735–7743.
- Polanski, J., 2003. Molecular shape analysis. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 302–319.
- Polanski, J., 2009a. Receptor dependent multidimensional QSAR for modeling drug-receptor interactions. *Curr. Med. Chem.* 16, 3243–3257.
- Polanski, J., 2009b. Chemoinformatics. In: Walczak, B., Tauler, R., Brown, S. (Eds.), *Comprehensive Chemometrics vol. 4*. Amsterdam: Elsevier, pp. 459–505.
- Polanski, J., 2017. Big data in structure-property studies – From definitions to models. In: Leszczynski, J., Roy, K. (Eds.), *Advances in QSAR Modeling With Applications in Pharmaceutical, Chemical, Food, Agricultural, and Environmental Sciences*. Berlin/Heidelberg: Springer.
- Polanski, J., Gasteiger, J., 2016. Computer representation of chemical compounds. In: Leszczynski, J., Puzyn, T. (Eds.), *Handbook of Computational Chemistry*. Dordrecht: Springer.
- Polanski, J., Gieleciak, R., Bak, A., Magdziar, T., 2006. Robust QSAR modeling. *J. Chem. Inf. Model.* 46, 2310–2318.
- Polanski, J., Kucia, U., Duszkiwicz, R., *et al.*, 2016. Molecular descriptor data explain market prices of a large commercial chemical compound library. *Sci. Rep.* 6, 28521.
- Polanski, J., Tkocz, A., 2017. Between descriptors and properties: Understanding the ligand efficiency trends for G protein-coupled receptor and kinase structure-activity data sets. *J. Chem. Inf. Model.* 57, 1321–1329.
- Pytela, O., Kulhanek, J., Ludwig, M., 1994. Chemometrical analysis of substituent effects. IV. Additivity of substituent effects in dissociation of 3,5-Disubstituted benzoic acids in organic solvents. *Collect. Czech. Chem. Commun.* 59, 1637–1644.
- Rapaport, C., 2004. *The art of molecular dynamics simulation*. Cambridge: Cambridge University Press.
- Reymond, J.-L., Van Deursen, R., Blum, L.C., Ruddigkeit, L., 2010. Chemical space as a source for new drugs. *Med. Chem. Commun.* 1, 30–38.
- Richards, W.G., 2002. Virtual screening using grid computing: The screensaver project. *Nat. Rev. Drug Discov.* 1, 551–555.
- Rodriguez-Barrios, F., Gago, F., 2004. Chemometrical identification of mutations in Hiv-1 reverse transcriptase conferring resistance or enhanced sensitivity to Arylsulfonylbenzonitriles. *J. Am. Chem. Soc.* 126, 2718–2719.
- Rucker, C., Meringer, M., 2002. How many organic compounds are graph-theoretically nonplanar? *MATCH Commun. Math. Comput. Chem.* 45, 153–172.
- Sadowski, J., Gasteiger, J., 1993. From atoms and bonds to three-dimensional atomic coordinates: Automatic model Builders. *Chem. Rev.* 93, 2567–2581.
- Scannell, J.W., Blankley, A., Boldon, H., Warrington, B., 2012. Diagnosing the decline in pharmaceutical R&D efficiency. *Nat. Rev. Drug Discovery* 11, 191–200.
- Schneider, G., Fechner, U., 2005. Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug Discov.* 4, 649–663.
- Schneider, P., Schneider, G., 2017. Privileged structures revisited. *Angew. Chem. Int. Ed. Engl.* Available at: <https://doi.org/10.1002/anie.201702816>.
- Shen, M., Beguin, C., Golbraikh, A., *et al.*, 2004. Application of predictive QSAR models to database mining: Identification and experimental validation of novel anticonvulsant compounds. *J. Med. Chem.* 47, 2356–2364.
- Smith, A., Heckelman, P.E., O'Neil, M.J., Budavari, S. (Eds.), 2001. *The Merck Index an Encyclopedia of Chemicals, Drugs, & Biologicals*, thirteenth ed. Whitehouse Station: Merck & Co. Inc.
- Smit, W.A., Caple, R., Bochkov, A.F., 1998. *Organic synthesis the science behind the Art*. London: Royal Society of Chemistry.
- Steinbeck, C.H., 2003. Computer-assisted structure elucidation. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 1378–1406.
- Swinney, D.C., Anthony, J., 2011. How were new medicines discovered? *Nat. Rev. Drug Discovery* 10, 507–519.
- Szlezak, N., Evers, M., Wang, J., Perez, L., 2014. The role of big data and advanced analytics in drug discovery, development, and commercialization. *Clin. Pharmacol. Ther.* 95, 492–4955.
- Szymkuc, S., Gajewska, E.P., Klucznik, T., Molga, K., Dittwald, P., 2016. Computer-assisted synthetic planning: The end of the beginning. *Angew. Chem. Int. Ed. Engl.* 55, 5904–5937.
- Todd, M.H., 2005. Computer-aided organic synthesis. *Chem. Soc. Rev.* 34, 247–266.
- Todeschini, R., Consonni, V., 2000. *Handbook of Molecular Descriptors*. Weinheim: Wiley-VCH.
- Tsiolaki, P.L., Nastou, K.C., Hamodrakas, S.J., Iconomidou, V.A., 2017. Mining databases for protein aggregation: A review. *Amyloid.* 24, 143–152.
- Van de Waterbeemd, H., Gifford, E., 2003. ADMET in silico modelling: Towards prediction paradise? *Nat. Rev. Drug Discov.* 2, 192–204.
- Wang, Y., Bryant, S.H., Cheng, T., *et al.*, 2017. PubChem bioassay statistics. *Nucleic Acids Res.* 45, D955–D963.
- Warren, G.L., Andrews, C.W., Capelli, A.-M., *et al.*, 2006. Critical assessment of Docking programs and scoring functions. *J. Med. Chem.* 49, 5912–5931.
- Weininger, D., 2003. SMILES – A language for molecules and reactions. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 80–102.
- Willet, P.A., 2003. History of Chemoinformatics. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 6–20.
- Wisniewski, J., 2003. Chemical nomenclature and structure representation: Algorithmic generation and conversion. In: Gasteiger, J. (Ed.), *Handbook of Chemoinformatics From Data to Knowledge*. Weinheim: Wiley-VCH, pp. 51–79.
- Wold, S., 1995. Chemometrics; What do we mean with it, and what do we want from it? *Chemometr. Intell. Lab. Syst.* 30, 109–115.
- Yang, C., Tarkhov, A., Maruszczyk, J., *et al.*, 2015. New publicly available chemical query language, CSRML, to support chemotype representations for application to data mining and modeling. *J. Chem. Inf. Model.* 15 (55), 510–528.

Further Reading

- Gasteiger, J. (Ed.), 2003. Handbook of Chemoinformatics From Data to Knowledge. Weinheim: Wiley-VCH.
- Gasteiger, J., Engel T. (Eds.), 2003. Chemoinformatics a Textbook, Weinheim: Wiley-VCH.
- Heller, S.R., 1997. (Ed.), The Beilstein Online Database – Implementation, Content, and Retrieval ACS Symposium Series 436, American Chemical Society, Washington, DC, 1990. 13. A.D. McNaught, A. Wilkinson, Compendium of Chemical Terminology (The Gold Book), second ed., Blackwell Scientific Publications, Oxford, UK.
- Kubinyi, H. (Ed.), 2008. QSAR: Hansch Analysis and Related Approaches. Weinheim, New York: VCH.
- Leszczynski, J., Puzyn, T. (Eds.), 2016. Handbook of Computational Chemistry. Dordrecht: Springer.
- Polanski, J., 2009. Chemoinformatics. In: Walczak, B., Tauler, R., Brown, S. (Eds.), Comprehensive Chemometrics. Amsterdam: Elsevier, pp. 459–505.
- Polanski, J., 2017. Big data in structure-property studies-from definitions to models. In: Leszczynski, J., Roy, K. (Eds.), Advances in QSAR Modeling With Applications in Pharmaceutical, Chemical, Food, Agricultural, and Environmental Sciences. Berlin/Heidelberg: Springer.
- Polanski, J., Gasteiger, J., 2016. Computer representation of chemical compounds. In: Leszczynski, J., Puzyn, T. (Eds.), Handbook of Computational Chemistry. Dordrecht: Springer.
- Roy, K. (Ed.), 2015. Quantitative Structure-Activity Relationships in Drug Design, Predictive Toxicology and Risk Assessment. Hershey PA: Medical Information Science Reference.
- Schweitzer, S.O., 2007. Pharmaceutical Economics and Policy. Oxford: University Press.
- Smith, M.B., March, J., 2001. March's Advanced Organic Chemistry Reactions Mechanisms, and Structure. New York: Wiley.
- Varnek, A. (Ed.), 2017. Tutorials in Chemoinformatics. Hoboken, Chichester: Wiley.
- Walczak, B., Tauler, R., Brown, S. (Eds.), 2009. Comprehensive Chemometrics, Vols. 1–4. Amsterdam: Elsevier.

Relevant Websites

- www.acdlabs.com
ACD Labs.
- www.beilstein-institut.de
Beilstein-Institut.
- www.chematica.net
CHEMATICA.
- www.cas.org/expertise/cascontent
CAS.
- www.daylight.com
DAYLIGHT.
- lhasa.harvard.edu
LHASA.
- <http://www.ch.ic.ac.uk/local/organic/mod/>
Molecular Modeling for Organic Chemistry.
- <http://www.organic-chemistry.org>
Organic Chemistry Portal.

Fingerprints and Pharmacophores

Daniela Schuster, Paracelsus Medical University of Salzburg, Salzburg, Austria

© 2019 Elsevier Inc. All rights reserved.

Introduction

In drug discovery and development as well as related fields such as toxicology or environmental chemical research, molecular similarity assessment is a central theme for various reasons. Comparing compounds and their properties may lead to information on the compound's physicochemical properties and even biological activities. Indeed, about 30% of compounds similar to an active compound have been reported to be active themselves (Martin *et al.*, 2002). Very often, researchers look out for similar compounds to identify related molecules, for example for studying structure-activity relationships. However, in many cases also dissimilar compounds are sought, for example when novel chemical scaffolds are acquired for a compound collection or when intellectual property opportunities are evaluated (Maggiora *et al.*, 2014).

Historically, similarity analysis was long based on fragment comparison or restricted to structurally very similar compound series. However, approaches describing the structures more completely and globally, thereby also allowing for more structural diversity within the studied data set, were needed (Cramer *et al.*, 1974). Nowadays, many similarity calculations and metrics are available (Cereto-Massagué *et al.*, 2015; Maggiora *et al.*, 2014); however, similarity assessment is still subjective and there is so far no consensus on the 'best' methodology.

This encyclopaedia entry will focus on two widely used similarity assessment methods: fingerprints and pharmacophore models. They represent global and local representations of the chemical structures involved in the comparison and therefore serve as examples to discuss this aspect as well.

Background and Fundamentals

Fingerprints are bit-string representations of molecular structure and properties. The bit-string encodes the absence (0) or presence (1) of a specific property, which can be a molecular structure or fragment. Depending on the method transforming the molecule into a bit-string, several types of fingerprints are available. While most of them are merely based on the 2D structure of the compounds, a few can also store 3D information. Fingerprints usually consider the whole molecule for generating the bit-string and are therefore classified as so-called global descriptors. They are straightforward to compute, and so fingerprint-based comparisons are among the fastest similarity calculations available.

Historically, the concept of a pharmacophore as the essential part of a drug determining its binding to a receptor was first presented by Paul Ehrlich (Ehrlich, 1898). The concept, its understanding and application evolved over the next decades (Güner and Bowen, 2014), and is nowadays an established and fundamental approach to understanding and predicting activities of small organic molecules. A pharmacophore model consists of the chemical and spatial arrangement of chemical functionalities, which are essential for the biological activity of a compound (Wermuth *et al.*, 1998). It contains so-called features like hydrogen bond acceptors/donors, hydrophobic areas, aromatic rings, positively or negatively ionizable groups or metal chelators. According to this definition, most features directly translate into target-ligand interactions essential for biological activity. Because only those crucial interaction points are considered when comparing compounds, pharmacophore models are classified as so-called local descriptors.

Approaches

The main fingerprint approaches include substructure keys-based fingerprints, topological fingerprints and circular fingerprints (Cereto-Massagué *et al.*, 2015). They differ in the way they are translating structural information into the final bit string. First, in the substructure keys-based fingerprints each bit stands for a chemical substructure, for example a carboxyl group, primary amine or phenyl. The individual fingerprints belonging into this group have a pre-set bit length and definitions at which position which substructure is encoded (Fig. 1).

Topological fingerprints analyze all the fragments of a molecule. Starting from each atom, a usually linear path up to a pre-defined number of bonds is followed. Each fragment produced in this procedure corresponds to a position in the bit string (Fig. 2).

The third class, the circular fingerprints, records the environment of each atom up to a determined radius (Fig. 3). These fingerprints are widely used for full structure similarity virtual screening.

Apart from directly describing a chemical structure, fingerprints can also represent other properties of molecules such as functional classes (functional class fingerprints, FCFP), pharmacophores (pharmacophore fingerprints, PF), reactions (reaction fingerprints, RF) and others.

Once the bit-strings resulting from the individual approaches are calculated, the similarities of the compared molecules need to be quantified. In principle, several similarity coefficients are available to choose from, for example the Tanimoto, Cosine,

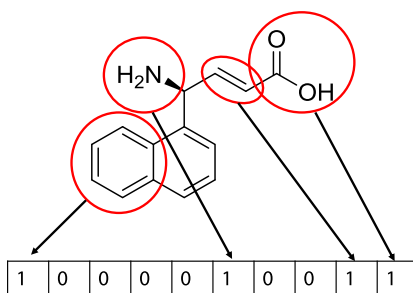


Fig. 1 Substructure keys-based fingerprints translate the presence or absence of chemical substructures into a bit string.

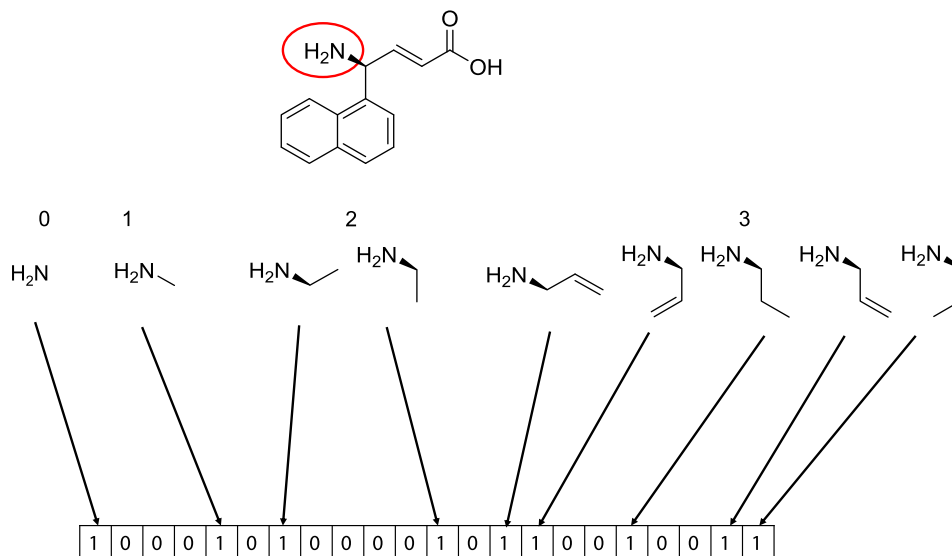


Fig. 2 The concept of topological fingerprints exemplified on the primary amino group of the molecule. Depending on the pre-set number of bonds, several linear pathway fragments are produced, which are then translated into a bit string.

Hamming, Russell-Rao or Forbes coefficients (Willett, 2006). In the last decade, the Tanimoto coefficient (TC) has evolved as the dominating similarity function. It is calculated according to Eq. (1)

$$TC = \frac{c}{a + b - c} \quad (1)$$

where the two compared molecules have a and b bit-sets in their fragment bit-strings and c of these bits are found in both of the bit-strings. The TC has values between 0 and 1, sometimes also expressed in percent. A value of 0 means no similarity, while 1 is calculated for identical molecules. A TC of ≥ 0.85 is considered to describe two quite similar compounds (Martin *et al.*, 2002).

Also pharmacophore models can be seen as a kind of fingerprint for a molecule. Each substructure of a molecule can be translated into a pharmacophore feature, such as for example a hydrogen bond acceptor, and a fingerprint can be created featuring the presence or absence of these features. However and more often, 3D pharmacophore models are created and used to identify chemical functionalities essential for biological activity and to search large 3D molecular databases for fitting, putatively active compounds.

In the ligand-based approach, active molecules are first transformed into 3D conformational models. These consist of representative sets of energetically favorable 3D structures for known active molecules. Those 3D structures are then superimposed in a way that equal chemical functionalities overlay. Most algorithms start with the most rigid active molecules, for which this 3D alignment is rather straightforward, and then proceed with more and more flexible molecules, which are aligned onto the more rigid ones. Finally, pharmacophore features are placed in the 3D space where chemical functionalities are common between the active molecules, which is called the common features or active analogue approach (Fig. 4).

There is usually a need for spatial restriction of the ligands to prevent large molecules from being reported as hits, which are likely too spacious to fit into the binding site. Technically, so-called exclusion volumes – forbidden areas for the ligands – are placed in the space surrounding the locations where the active molecules fit. If available, inactive molecules fitting the features of the pharmacophore can be used to strategically place exclusion volumes outside the active molecules' volumes.

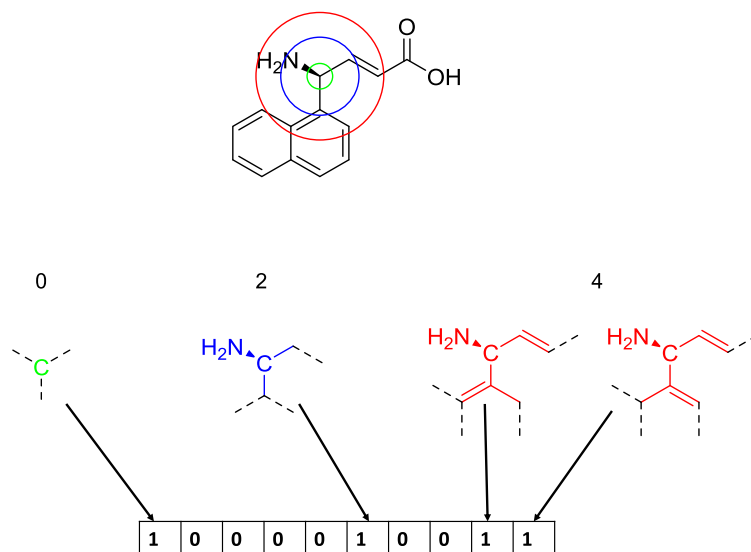


Fig. 3 Circular fingerprints start at a central atom and then consider the molecular environment in a pre-set diameter of e.g., two or four bonds, respectively. The thereby generated fragments are then translated into a bit string.

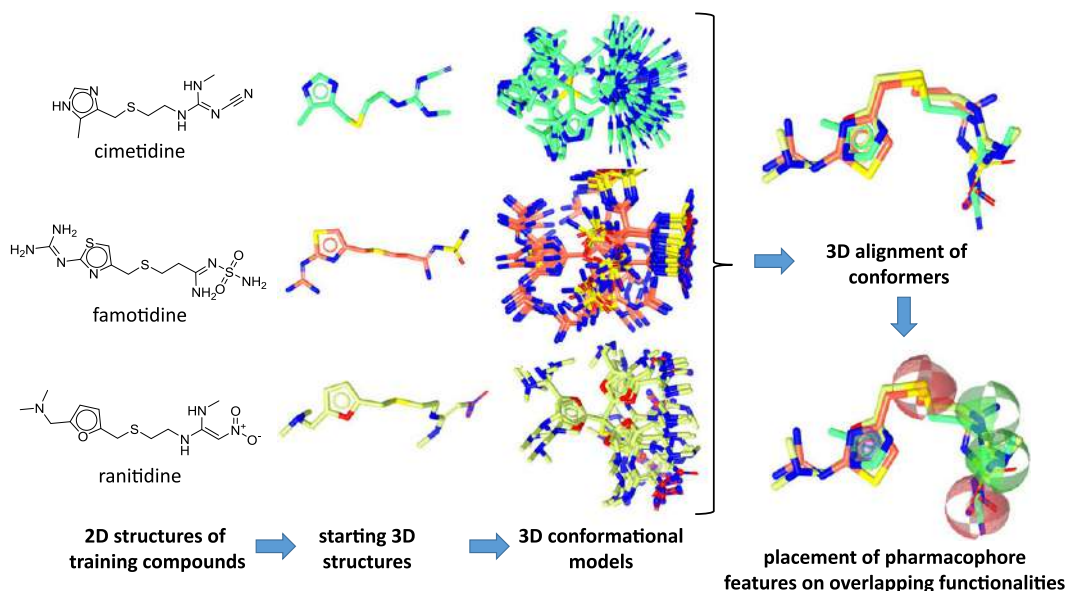


Fig. 4 Ligand-based pharmacophore model generation exemplified on three histamine A_2 receptor antagonists. Starting from a 2D structure, corresponding 3D structures and conformational models are calculated for the training ligands. The conformers are then aligned in a way that equal chemical functionalities overlay. Finally, pharmacophore features are placed on these common functionality locations. In this example, the aromatic ring feature is shown in blue, hydrogen bond donor points as green, and hydrogen bond acceptor points in red.

If there is a high quality data set available, the development of quantitative pharmacophore models aimed to predict a compound's potency is also feasible. The main limitation in this respect is the data quality. For model generation, at least 16 molecules spanning four orders of magnitude of activity, all tested in the same assay, and preferably in the same lab are required. Ideally, many more ligands from the same test are available for quantitative model validation afterwards. This stringent requirements often hamper the generation of quantitative pharmacophore models that are especially useful for ligand optimization.

In the structure-based pharmacophore modeling approach, the 3D structure of the target is known – ideally complexed to a bioactive molecule. This 3D structure can be derived experimentally by X-ray crystallography or NMR studies. Alternatively,

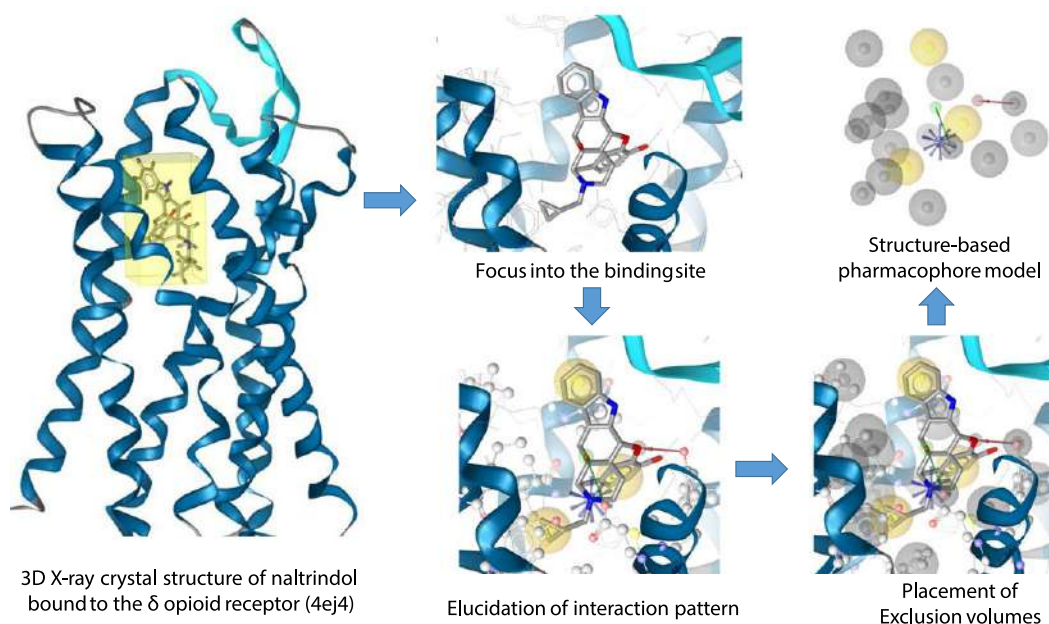


Fig. 5 The workflow of structure-based pharmacophore model generation exemplified on the δ opioid receptor antagonist naltrindole co-crystallized with its target (Protein Data Bank entry 4ej4). Starting from the 3D complex, the focus is put on just the direct environment of the ligand binding site. The ligand's surrounding amino acids are analyzed for possible direct interactions with the ligand based on their chemical functionalities, distances and contact angles. At interacting positions, pharmacophore features (yellow – hydrophobic, blue star – positively ionizable, red arrow – hydrogen bond acceptor) are placed. Exclusion volumes (grey) are placed on amino acid side chains forming the binding site. Finally, all information on the receptor and the ligand are deleted and the interaction pattern with the exclusion volumes remains as an abstract representation of binding requirements. This is the starting point for model validation, optimization and use in virtual screening.

theoretical target models, so-called homology models, with a docked active molecule can serve as basis for the model. In a first step, all observed target-ligand interactions are mapped onto the respective parts of the ligand (Fig. 5).

Such interaction models usually consist of far more chemical features than are necessary for biological activity. So how can the modeler find out which are the more important features? If the data situation is good, information can come from mutational analysis and other 3D target-ligand complexes. Amino acid mutants leading to a change of ligand affinity can help to identify important anchoring points for the molecule. Additionally, interactions observed in several target-ligand complexes most likely play an important role for activity modulation (Sun *et al.*, 2011). Furthermore, interaction maps from molecular dynamics simulations can help to prioritize specific interactions (Moorthy *et al.*, 2016). If such data is not available, a so-called test data set containing active and inactive compounds can help. The initial interaction model is systematically altered and applied to screen the test set. Any alteration that leads to a better retrieval of active compounds and no increase of found inactive compounds is a step towards a more general and representative model for modulators of the respective target. Furthermore, exclusion volumes can be strategically placed on amino acid residues lining the binding site (Vuorinen *et al.*, 2014). This strategy can also be followed to optimize ligand-based models.

Shape-based searches and pharmacophore features are frequently combined. Many pharmacophore modeling tools include a shape 'feature', which is derived from the shape and volume of active compounds fitting the model. The shape is included as a spatial restriction of the size of fitting hits. Also some shape-based screening programs like Openeyes ROCS can include pharmacophore-like features, which are termed color features. Although they may not be directly comparable with true physico-chemical property-based pharmacophore features, they share several functionalities such as 'hydrogen bond acceptor', 'hydrogen bond donor', 'negative charge', and some more. Another shape-pharmacophore-combining screening algorithm is the ultrafast shape recognition with CREDO atom types (USRCAT, Schreyer and Blundell, 2012).

Protein binding site-based pharmacophore models can be used for model generation without prior knowledge of active ligands. A so-called grid is calculated on the binding site surface and analyzed for potential interaction points for ligand binding (Fig. 6). According to available interacting amino acid residues, locations for pharmacophore features are suggested, where the modeler can place the respective features.

Similarly, pharmacophore-based docking has been reported. One of these approaches is PharmDock (Hu and Lill, 2014). It can be used for pose prediction and compound ranking like docking programs, thereby being much faster than common docking algorithms. The pharmacophore-guided docking has been shown to be superior over unbiased docking, which is not surprising. Unbiased docking lacks information on the really important interactions in the binding site, while pharmacophore-guided docking includes this information, which is added in the workflow development. However, docking with constraints forcing a docking ligand to establish certain interactions may eventually lead to similar results as the pharmacophore-guided approach. Like

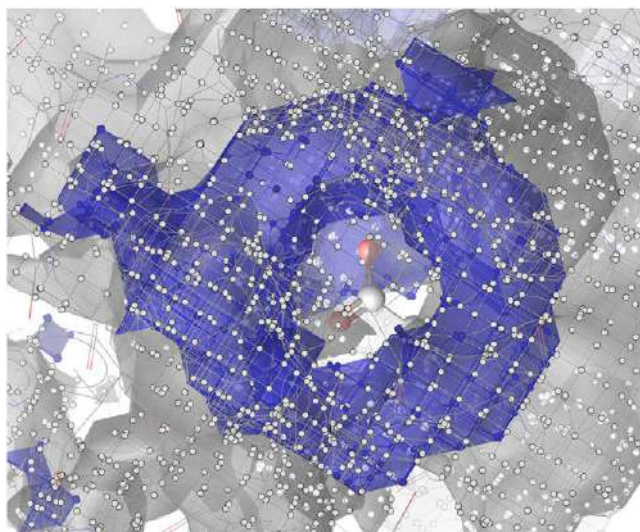


Fig. 6 Visualized grid for positively ionizable ligand functionalities (blue) surrounding Asp128 in the δ opioid receptor binding site (4ej4). The empty binding site is filled with small dots representing possible locations for respective pharmacophore features. The individual dots can be selected for placing the features and manually constructing a model filling the binding site with complementary functionalities.

all other docking approaches, the inaccuracies of the available scoring functions in predicting binding affinities limit the use of pharmacophore-guided docking in compound optimization.

Illustrative Examples and Case Studies

Fingerprint-based similarity searches are widely used in cheminformatics. Large bioactivity databases like ChEMBL (Bento *et al.*, 2014) and PubChem (Wang *et al.*, 2017a) as well as chemical databases such as the SciFinder database operated by Chemical Abstracts Service have implemented fingerprints to search for similar molecules with defined Tanimoto coefficient range. A broad similarity search in various, huge databases using different fingerprints is implemented in the SwissSimilarity platform (Zoete *et al.*, 2016). For database analysis and library design, fingerprint methods are used to group similar compounds and evaluate databases for their structural diversity. They are also used to represent whole compound libraries for comparison amongst each other (Fernández de Gortari *et al.*, 2017). Furthermore they are successfully used for comparing chemical structures with compounds of known bioactivities or ensembles thereof, for example the ECFP4-based similarity ensemble approach (SEA) prediction platform (Keiser *et al.*, 2007). This tool has been successfully used for identifying off-target effects for a variety of approved drugs (Keiser *et al.*, 2009). For example amitriptyline has been identified as a nanomolar serotonin transporter inhibitor in this study, which has been previously unknown. The SEA platform has also been used to find a target for the natural product miconidin acetate (Zatelli *et al.*, 2016). In studies comparing several virtual screening approaches for their predictive power, SEA proved to be very successful in finding new active compounds (Kaserer *et al.*, 2015, 2016). However, the correct classification of inactive compounds turned out to be a challenge, possibly because the structure-activity database in the background is a limited source of information on inactive compounds. Another example for a fingerprint-based profiler is the SwissTargetPrediction platform, which uses 2D fingerprints besides 3D similarity measures for assessing ligand similarities and associations to putative targets (Gfeller *et al.*, 2014). Other fingerprint-based activity prediction tools include SuperPred (Nickel *et al.*, 2014), MOST (Huang *et al.*, 2017) and the polypharmacology browser, which even combines 10 fingerprints for similarity assessment (Awale and Reymond, 2017).

Pharmacophore models are established as useful tools to discover novel active compounds for specific targets, predict mechanism-based side effects or calculate a bioactivity profile for a molecule. For example, Markt *et al.* (2009) used two ligand-based, shape-restricted pharmacophore models for the discovery of novel cannabinoid receptor 2 (CB₂) ligands. From a training set of five potent CB₂ ligands, common feature pharmacophore models were generated and merged with one of the training compound's shape, respectively. A test set of 15 active CB₂ ligands and over 67,000 decoys (putative inactive compounds) were used to determine the enrichment of active compounds in a virtual screening setting. The two best performing models were selected and employed to virtually screen commercial compound databases. From initially over 920,000 compounds in the databases, nearly 30,000 fitted into either or both models. Further filtering using physicochemical properties, structural similarity analysis and visual inspection led to the selection of 14 hits for biological evaluation. Out of these 14 compounds, three showed K_i values ≤ 2 μ M, which corresponds to a success rate of over 20% true positive hits.

A classical side effect modeling study was performed by Kratz *et al.* (2014). They assembled a data set of human ether-a-go-go-related (hERG) potassium channel blockers. In the course of this study, not just one but a whole set of models was created for hERG blockers. This is because one pharmacophore model is unable to predict all active molecules for a target. Usually, this

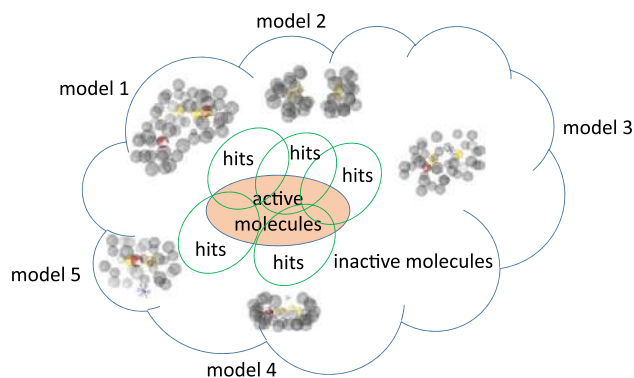


Fig. 7 The parallel use of several complementary pharmacophore models allows for a more complete retrieval of active compounds. The cloud incorporates a database of active (rose) and inactive (white) compounds. Each model used for screening derives a hit list of fitting compounds (green circles). While a single model only finds a fraction of true active compounds, the parallel use of models finding mostly different active compounds improves the overall retrieval of active molecules.

doesn't pose a problem because in a "cherry picking" scenario, the scientists are interested to find some hits for further development. In side effect studies however, the project aims to identify all potentially harmful compounds. Therefore, several models have to be generated for an adverse effect target not to miss the majority of active hits (Fig. 7).

The model set by Kratz *et al.* was applied to databases containing synthetic and natural products and successfully identified 13 previously unknown hERG blockers from 50 experimentally tested virtual hits.

Sometimes also the use of several pharmacophore modeling tools is beneficial because they may find complementary active hits. Temml *et al.* (2014) generated structure-based pharmacophore models from one cyclooxygenase-2 – inhibitor X-ray crystal structure using the two pharmacophore modeling programs LigandScout and Catalyst (implemented in Discovery Studio). The LigandScout model found 53 virtual hits and the Discovery Studio model 74 hits, respectively. Surprisingly, not a single hit was present in both hit lists. However, when 10 virtual hits from every hit list were biologically tested, each found 4 active compounds, respectively. So although the resulting hits of the virtual screening were completely different, each of the programs retrieved active hits.

The concept of using several models can be further expanded by applying several models for a broad set of targets. This concept is termed pharmacophore-based parallel screening or pharmacophore-based target fishing. The openly available online platform PharmMapper offers over 53,000 structure-based pharmacophore models for activity predictions (Wang *et al.*, 2017b). In comparative studies on virtual screening tools, PharmMapper performed well for cherry picking with 11 active compounds out of 19 hits predicted for cyclooxygenase (Kaserer *et al.*, 2015) and one active ligand out of two predicted ones for the peroxisome proliferator-activated receptor γ (Kaserer *et al.*, 2016), respectively. However, many active molecules from these studies were not found by this platform. Additionally, several companies offer pharmacophore databases along with their software such as Biovia Discovery Studio (PharmaDB; see Relevant Websites section) and inte:ligand (pharmacophoreDB; see Relevant Websites section). Examples from the academia include target fishing for compounds of the herbal remedy *Ruta graveolens* (Rollinger *et al.*, 2010) and the natural product leoligin, the main lignan from *Leontopodium alpinum* (Duwensee *et al.*, 2011).

Results and Discussion

Fingerprint-based methods and pharmacophore models are established and important tools in cheminformatics and virtual screening. They are likewise used in industry and in the academic field. A wide variety of fingerprints are available and used for comparing molecules for diverse purposes. In library analysis, they are important to identify the chemical diversity of a compound collection, which is regarded as beneficial for drug discovery libraries. Also when planning to purchase new compounds, fingerprint methods can report if the compounds to purchase have sufficient dissimilarity to the substances that are already on stock.

Because of their speed, fingerprint-based methods are very popular for activity predictions, especially in the field of target fishing, where query compounds are compared against millions of compounds with known activities. More and more platforms based on fingerprints are published each year and most of them are publicly available. However, success stories reporting a prospective use of these tools are rare and therefore their predictive power still needs to be critically evaluated and compared to other tools. At some point one gets the impression that developing new tools may not be meaningful unless the existing platforms have been properly validated and their strengths and weaknesses have been revealed. First validation studies suggest that the predictive power in the identification of active compounds is satisfying but still highly depending on the activity database in the background. For the same reason, the correct prediction of inactive compounds seems to remain a challenge, but will improve with the growing amount of structure-activity data. A combination with other tools that are more powerful in this respect may be a good strategy to overcome this current drawback.

In comparison, success studies reporting prospective pharmacophore-based applications are easy to spot in the scientific literature as exemplified above. Pharmacophore models are used as standalone virtual screening tools or used in combination with physicochemical filtering, structural clustering, shape-based screening or docking or a combination thereof. Nowadays, pharmacophore modeling software

like many other virtual screening tools is convenient to use, with an easy to navigate graphical user interface and sometimes even ‘wizards’ that guide through the model generation process. This easy-to-use mentality sometimes distracts from the actually very sophisticated process of model development. At the beginning, a data set for model generation and theoretical validation is to be assembled. This data need to be of high quality and be derived from suitable assays. Essentially, if a model should reflect binding to a specific protein target, also data reporting the direct binding to this target needs to be used for the modeling. Data from cell-based assays, where also other events than direct ligand binding influence activity, are not suitable. Model generation is technically not a challenge any more, which makes a rigorous validation protocol crucial. Theoretical model optimization and validation using independent data sets must be followed by a prospective, experimental evaluation of virtual hits. The validation experiments must also comply with the rules for the data set. For instance, watching cells die is not a proof for kinase inhibition or any other specific mechanism of action. Cell-free, target-based binding assays are required to prove the pharmacophore hypothesis. Models lacking high quality experimental validation are of very limited value and journals should carefully consider if they are worth publishing at all. Of course, favorable hits should also be active in cell-based systems, but this is only the second step after prospective model validation.

Excitingly, many *in silico* tools are nowadays openly available to the public, either as online platforms operated on a remote server or as stand-alone programs that can be downloaded and locally installed on the workstation (Pirhadi *et al.*, 2016). This way, those tools are freely accessible to interested scientists all over the world for a more targeted planning of experimental work.

Future Directions

Both fingerprint-based methods and pharmacophore modeling have been successfully employed in a variety of research projects and are important, established tools in the analysis and discovery of active compounds. However, there is also room for improvement for both approaches.

Regarding fingerprint-based target fishing, the community currently seems to be struggling with quantitative target ranking. How shall hundreds of predicted targets be judged by the user? Which ones are the most probable ones and should be prioritized for experimental evaluation? There is currently a need to introduce a measure of significance for these tools, which enable an estimation of the chance that a virtually predicted activity is indeed probable. While there are first attempts in the field (Vogt and Bajorath, 2017), it will be a long way to go until these more sophisticated measures are implemented into the similarity-based target fishing tools. Furthermore, new fingerprint algorithms are reported regularly and their optimal application in similarity screening need to be determined.

The pharmacophore modeling branch currently suffers from severe drawbacks in feature definition and screening algorithms that rely on different concepts (Temml *et al.*, 2014; Wolber *et al.*, 2008). This leads to incompatibilities between different software or – even worse – sometimes between different versions of the same software. It will be a challenge to bring feature definitions to an up-to-date state including halogen bonds and sulfur bonds as well as a better consideration of internal hydrogen bonds in model fitting. Furthermore, hydrophobic areas are currently represented as globular spheres in the programs, which doesn’t reflect reality. Adjusting the hydrophobic contacts feature to more accurate geometries may also improve quantitative scoring of hit molecules. Finally, incorporating protein-ligand-contact information from molecular dynamics simulations in the development of pharmacophore models will lead to a better understanding of ligand binding and more predictive models in the future.

Closing Remarks

Fingerprints and pharmacophore models are indispensable methods in drug research. As in every cheminformatics field, research in the fingerprint and pharmacophore area is very active and there is constant progress in the improvement of the available tools and development of new approaches. Nevertheless, critical validation studies and especially benchmarking and prospective case studies are essential to identify the power and also the pitfalls of new developments. It is also important to see and apply these tools in context with other property calculation and virtual screening tools to make the most out of them.

Acknowledgement

I thank Philipp Schuster for help in the preparation of this manuscript and Inte:Ligand GmbH for providing an academic license for the generation of pharmacophore-related figures. This work was supported by the Austrian Science Fund project P26782.

See also: Binding Site Comparison – Software and Applications. Chemical Similarity and Substructure Searches. Chemoinformatics: From Chemical Art to Chemistry in Silico. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Population Analysis of Pharmacogenetic Polymorphisms. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Structural Bioinformatics: An Overview. Rational Structure-Based Drug Design. Small Molecule Drug Design. Structure-Based Drug Design Workflow

References

- Awale, M., Reymond, J.L., 2017. The polypharmacology browser: A web-based multi-fingerprint target prediction tool using ChEMBL bioactivity data. *Journal of Cheminformatics* 9, 11.
- Bento, A.P., Gaulton, A., Hersey, A., *et al.*, 2014. The ChEMBL bioactivity database: An update. *Nucleic Acids Research* 42, D1083–D1090.
- Cereto-Massagué, A., Ojeda, M.J., Valls, C., *et al.*, 2015. Molecular fingerprint similarity search in virtual screening. *Methods* 71, 58–63.
- Cramer III, R.D., Redl, G., Berkoff, C.E., 1974. Substructural analysis. A novel approach to the problem of drug design. *Journal of Medicinal Chemistry* 17, 533–535.
- Duwesen, K., Schwaiger, S., Tancevski, I., *et al.*, 2011. Leoligin, the major lignan from Edelweiss, activates cholesteryl ester transfer protein. *Atherosclerosis* 219, 109–115.
- Ehrlich, P., 1898. Über die Constitution des Diphtheriegiftes. *Deutsche Medizinische Wochenschrift* 24, 597–600.
- Fernández de Gortari, E., García-Jacas, C.R., Martínez-Mayorga, K., Medina-Franco, J.L., 2017. Database fingerprint (DFP): An approach to represent molecular databases. *Journal of Cheminformatics* 9, 9.
- Gieller, D., Grosdidier, A., Wirth, M., *et al.*, 2014. SwissTargetPrediction: A web server for target prediction of bioactive small molecules. *Nucleic Acids Research* 42, W32–W38.
- Güner, O.F., Bowen, J.P., 2014. Setting the record straight: The origin of the pharmacophore concept. *Journal of Chemical Information and Modeling* 54, 1269–1283.
- Hu, B., Lill, M.A., 2014. PharmDock: A pharmacophore-based docking program. *Journal of Cheminformatics* 6, 14.
- Huang, T., Mi, H., Lin, C.Y., *et al.*, 2017. MOST: Most similar ligand based approach to target prediction. *BMC Bioinformatics* 18, 165.
- Kaserer, T., Obermoser, V., Weninger, A., *et al.*, 2016. Evaluation of selected 3D virtual screening tools for the prospective identification of peroxisome proliferator-activated receptor (PPAR) γ partial agonists. *European Journal of Medicinal Chemistry* 124, 49–62.
- Kaserer, T., Temml, V., Kutil, Z., *et al.*, 2015. Prospective performance evaluation of selected common virtual screening tools. Cast study: Cyclooxygenase (COX) 1 and 2. *European Journal of Medicinal Chemistry* 96, 445–457.
- Keiser, M.J., Roth, B.L., Armbruster, B.N., *et al.*, 2007. Relating protein pharmacology by ligand chemistry. *Nature Biotechnology* 25, 197–206.
- Keiser, M.J., Setola, V., Irwin, J.J., *et al.*, 2009. Predicting new molecular targets for known drugs. *Nature* 462, 175–182.
- Kratz, J.M., Schuster, D., Edtbauer, M., *et al.*, 2014. Experimentally validated hERG pharmacophore models as cardiotoxicity prediction tools. *Journal of Chemical Information and Modeling* 54, 2887–2901.
- Maggiora, G., Vogt, M., Stumpfe, D., Bajorath, J., 2014. Molecular similarity in medicinal chemistry. *Journal of Medicinal Chemistry* 57, 3186–3204.
- Markt, P., Feldmann, C., Röllinger, J.M., *et al.*, 2009. Discovery of novel CB₂ receptor ligands by a pharmacophore-based virtual screening workflow. *Journal of Medicinal Chemistry* 52, 369–378.
- Martin, Y.C., Kofron, J.L., Traphagen, L.M., 2002. Do structurally similar molecules have similar biological activity? *Journal of Medicinal Chemistry* 45, 4350–4358.
- Moorthy, N.S.H.N., Sousa, S.F., Ramos, M.J., Fernandes, P.A., 2016. Molecular dynamic simulations and structure-based pharmacophore model development for farnesyltransferase inhibitors discovery. *Journal of Enzyme Inhibition and Medicinal Chemistry* 31, 1428–1442.
- Nickel, J., Gohlke, B.O., Erehman, J., *et al.*, 2014. SuperPred: Update on drug classification and target prediction. *Nucleic Acids Research* 42, W26–W31.
- Pirhadi, S., Sunseri, J., Koes, D.R., 2016. Open source molecular modeling. *Journal of Molecular Graphics and Modeling* 69, 127–143.
- Röllinger, J.M., Schuster, D., Danz, B., *et al.*, 2010. In silico target fishing for rationalized ligand discovery exemplified on constituents of *Ruta graveolens*. *Planta Medica* 75, 195–204.
- Schreyer, A.M., Blundell, T., 2012. USRCAT: Real-time ultrafast shape recognition with pharmacophoric constraints. *Journal of Cheminformatics* 4, 27.
- Sun, H.P., Zhu, J., Chen, F.H., You, Q.D., 2011. Structure-based pharmacophore modeling from multicomplex: A comprehensive pharmacophore generation of protein kinase CK2 and virtual screening based on it for novel inhibitors. *Molecular Informatics* 30, 579–592.
- Temml, V., Kaserer, T., Kutil, Z., *et al.*, 2014. Pharmacophore modeling for COX-1 and -2 inhibitors with LigandScout in comparison with Discovery Studio. *Future Medicinal Chemistry* 6, 1869–1881.
- Vogt, M., Bajorath, J., 2017. Modeling Tanimoto similarity value distributions and predicting search results. *Molecular Informatics* 36, 1600131.
- Vuorinen, A., Nashev, L.G., Odermatt, A., Röllinger, J.M., Schuster, D., 2014. Pharmacophore model refinement for 11 β -hydroxysteroid dehydrogenase inhibitors: Search for modulators of intracellular glucocorticoid concentrations. *Molecular Informatics* 33, 15–25.
- Wang, Y., Bryant, S.H., Cheng, T., *et al.*, 2017a. PubChem BioAssay: 2017 update. *Nucleic Acids Research* 45, D955–D963.
- Wang, X., Shen, Y., Wang, S., *et al.*, 2017b. PharmMapper 2017 update: A web server for potential drug target identification with a comprehensive target pharmacophore database. *Nucleic Acids Research* 45, W356–W360.
- Wermuth, C.G., Ganellin, C.R., Lindberg, P., Mitscher, L.A., 1998. Glossary of terms used in medicinal chemistry. *Pure & Applied Chemistry* 70, 1129–1143.
- Willet, P., 2006. Similarity-based virtual screening using 2D fingerprints. *Drug Discovery Today* 11, 1046–1053.
- Wolber, G., Seidel, T., Bendix, F., Langer, T., 2008. Molecule-pharmacophore superpositioning and pattern matching in computational drug design. *Drug Discovery Today* 13, 23–29.
- Zatelli, G.A., Temml, V., Kutil, Z., *et al.*, 2016. Miconidin acetate and primin as potent 5-lipoxygenase inhibitors from Brazilian *Eugenia hiemalis* (myrtaceae). *Planta Medica Letters* 3, e17–e19.
- Zoete, V., Daina, A., Bovigny, C., Michielin, O., 2016. SwissSimilarity: A web tool for low to ultra high throughput ligand-based virtual screening. *Journal of Chemical Information and Modeling* 56, 1399–1404.

Further Reading

- Akram, M., Kaserer, T., Schuster, D., 2017. Pharmacophore modeling and pharmacophore-based virtual screening. In: Cavasotto, C.N. (Ed.), *In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications*. Boca Raton, FL, USA: CRC Press, pp. 123–153.
- Maggiora, G.M., Shanmugasundaram, V., 2011. Molecular similarity measures. In: Bajorath, J. (Ed.), *Springer Protocols: Chemoinformatics and Computational Chemical Biology*. New York: Humana Press.
- Muegge, I., Mukherjee, P., 2016. An overview of molecular fingerprint similarity search in virtual screening. *Expert Opinion on Drug Discovery* 11, 137–148.
- Pharmacophores and Pharmacophore Searches. Wiley VCH, Thierry Langer. Available at: [https://www.wiley.com/en-at/Pharmacophores + and + Pharmacophore + Searches-p-9783527608720](https://www.wiley.com/en-at/Pharmacophores+%26amp%3B+Pharmacophore+Searches-p-9783527608720).
- Rogers, D., Hahn, M., 2010. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* 50, 742–754.
- Willet, P., Barnard, J., Downs, G., 1998. Chemical similarity searching. *Journal of Chemical Information and Computer Sciences* 38, 983–996.

Relevant Websites

<http://accelrys.com/products/collaborative-science/biovia-discovery-studio/pharmacophore-and-ligand-based-design.html>

BIOVIA Discovery Studio I Pharmacophore and Ligand-Based Design.

<http://www.inteligand.com/pharmdb/>

Inteligand.

www.click2drug.org

List of programs, tools and software for molecular modeling, analysis and drug discovery.

<http://lilab.ecust.edu.cn/pharmmapper/>

PharmMapper.

<http://gdbtools.unibe.ch:8080/PPB/>

Polypharmacology browser.

<http://sea16.ucsf.bkslab.org/>

SEA activity profiler.

<http://prediction.charite.de/>

SuperPred.

<http://www.swisstargetprediction.ch/>

SwissTargetPrediction.

Biographical Sketch



Daniela Schuster studied pharmacy at the University of Innsbruck, Austria, and graduated in 2003. In her PhD time, she worked in an e-learning project at the University of Graz, Austria. She also worked as senior application scientist for Inte:Ligand in Vienna. As the first Erika Cremer habilitation fellow and one of the first Ingeborg Hochmair professors at the University of Innsbruck, she led the Computer-Aided Molecular Design Group at the Institute of Pharmacy/Pharmaceutical Chemistry. In 2018, she accepted a position as professor for Pharmaceutical and Medicinal Chemistry at the Paracelsus Medical University in Salzburg, Austria.

Public Chemical Databases

Sunghwan Kim, National Institutes of Health, Bethesda, MD, United States

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

ADMET	Absorption, distribution, metabolism, excretion, and toxicity	HTS	High-throughput screening
ADR	Adverse drug reaction	IC₅₀	Half-maximal inhibitory concentration
BPS	British Pharmacological Society	InChI	International Chemical Identifier
CPIC	Clinical Pharmacogenomics Implementation Consortium	INN	International non-proprietary name
CSSP	ChemSpider SyntheticPages	IUPHAR	International Union of Basic and Clinical Pharmacology
CTD	Comparative Toxicogenomics Database	K_d	Dissociation constant
CYP450	Cytochrome P450	K_i	Inhibitory constant
DOI	Digital object identifier	MESH	Medical subject headings
EBI	European Bioinformatics Institute	MS	Mass spectrometry
EMBL	European Molecular Biology Laboratory	NC-IUPHAR	IUPHAR Committee on Receptor Nomenclature and Drug Classification
FDA	Food and Drug Administration	NER	Named entity recognition
GC-MS	Gas chromatography-mass spectrometry	NHR	Nuclear hormone receptor
GO	Gene ontology	NIH	U.S. National Institutes of Health
GPCR	G protein-coupled receptor	NMR	Nuclear magnetic resonance
GRAC	Guide to receptors and channels	NLP	Natural language processing
GtoPdb	Guide to PHARMACOLOGY	PD	Pharmacodynamics
hERG	Human ether-a-go-go related gene	PDB	Protein Data Bank
HMDB	Human Metabolome Database	PGRN	Pharmacogenomics Research Network
HSDB	Hazardous Substances Data Bank	PK	Pharmacokinetics
		SIDER	Side effect resource
		SRP	Scientific review panel

Introduction

For the past decade, the amount of chemical data available in the public domain has been rapidly increasing (Cheng *et al.*, 2014; Hu and Bajorath, 2012; Nicola *et al.*, 2012). This led to the establishment of many databases that collect, curate, and disseminate publicly available chemical information with the aim of providing easier access to and better utility of these data (Akhondi *et al.*, 2012; Hersey *et al.*, 2015; Lipinski *et al.*, 2015; Ma *et al.*, 2012; Muresan *et al.*, 2011; Nicola *et al.*, 2012; Tiikkainen and Franke, 2012; Tiikkainen *et al.*, 2013; Williams, 2008a,b; Williams *et al.*, 2012). Importantly, these databases annotate chemical data with information from other scientific domains (such as genetics, genomics, pharmacology, toxicology, medicine, bioinformatics, and systems biology), assisting biomedical researchers in taking advantage of these public data for the discovery of new medications. To maximize the usefulness of the public chemical data, scientists should be aware of which databases have the information they need, and know how to exploit their full potential.

This paper provides a brief overview of some important chemical databases available in the public domain (see Fig. 1). Their uniform resource locators (URLs) are listed in Fig. 1 as well as in the Relevant Websites section at the end of this article. These databases vary in data contents, size, and quality (see Figs. 1 and 2) (Akhondi *et al.*, 2012; Hersey *et al.*, 2015; Lipinski *et al.*, 2015; Muresan *et al.*, 2011; Tiikkainen and Franke, 2012; Tiikkainen *et al.*, 2013; Williams *et al.*, 2012). Some of them provide a wide variety of information on millions of molecules, aggregated from hundreds of data sources. Others contain specialized information for a few thousand molecules, manually curated from peer-reviewed scientific articles. Because data exchange between these public databases is a very common practice, there are some overlaps in the data contents, but each database does have unique contents valuable to the scientific community. While they provide various tools and services for searching, browsing, visualization, download, programmatic access, etc., the primary focus of this article is on their data contents, to provide some guidance on what database to use to get a particular kind of information.

• PubChem (https://pubchem.ncbi.nlm.nih.gov) Provides comprehensive information on >93 million unique compounds, collected from >500 data sources, along with experimentally determined bioactivity data.
• ChemSpider (http://www.chemspider.com) Contains information for >59 million compounds, collected from ~500 data sources.
• ChEMBL (https://www.ebi.ac.uk/chembl/) Contains information on bioactive drug-like small molecules and their bioactivities, abstracted from literature.
• BindingDB (https://www.bindingdb.org) Contains experimental binding affinities, focusing on the interactions of proteins considered to be drug-targets with drug-like small molecules.
• PDBbind (http://www.pdbbind.org.cn) Provides experimental binding affinity data for the biomolecular complexes archived in PDB.
• Binding MOAD (http://www.bindingmoad.org) Provides experimental binding affinity data for the high-quality structures of protein-ligand complexes derived from PDB.
• DrugBank (https://www.drugbank.ca) Provides comprehensive information for FDA-approved and investigational drugs.
• IUPHAR/BPS Guide To PHARMACOLOGY (http://www.guidetopharmacology.org) Contains structures of small molecule ligands, peptides, and antibodies, with their affinities at protein targets.
• PharmGKB (Pharmacogenomics Knowledge Base) (https://www.pharmgkb.org) Provides curated information about the impact of genetic variation on drug response.
• SIDER (Side Effect Resource) (http://sideeffects.embl.de) Provides information on adverse drug reactions (ADRs) for drugs, extracted from drug labels and package inserts.
• Comparative Toxicogenomics Database (CTD) (https://ctdbase.org) Provides information on interactions between chemicals and genes/proteins and their relationship to diseases.
• Hazardous Substances Data Bank (HSDB) (https://www.toxnet.nlm.nih.gov/newtoxnet/hsdb.htm) Provides toxicological information for more than 5,800 potentially hazardous chemicals.
• Chemical Entities of Biological Interest (ChEBI) (https://www.ebi.ac.uk/chebi/) A freely available dictionary of molecular entities, focused on small chemical compounds.
• Human Metabolome Database (HMDB) (http://www.hmdb.ca) Contains detailed information about small molecule metabolites found in the human body.
• UniChem (https://www.ebi.ac.uk/unichem/) A free compound identifier mapping service, containing more than 145 million chemical structures stored in >30 databases.

Fig. 1 List of public chemical databases discussed in this paper.

Large-Scale Data Aggregators

PubChem

PubChem (Kim, 2016; Kim *et al.*, 2016a; Wang *et al.*, 2017) is a public chemical information resource, developed and maintained by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM), an institute within the U.S. National Institutes of Health (NIH). It collects chemical substance descriptions and their biological activities from more than 500 data sources and disseminates these data to the public free of charge. Since the launch in 2004 as a component of the NIH Molecular Libraries Roadmap Initiatives, PubChem has been a key information resource for biomedical research communities in many areas such as cheminformatics, chemical biology, medicinal chemistry, and drug discovery.

PubChem contains various types of chemical information, including 2-D and 3-D structures, chemical and physical properties, bioactivity data, pharmacology, toxicology, drug target, metabolism, safety and handling, relevant patents and scientific papers, etc. While the majority of PubChem's records are about small molecules, it also contains information on a broad range of chemical

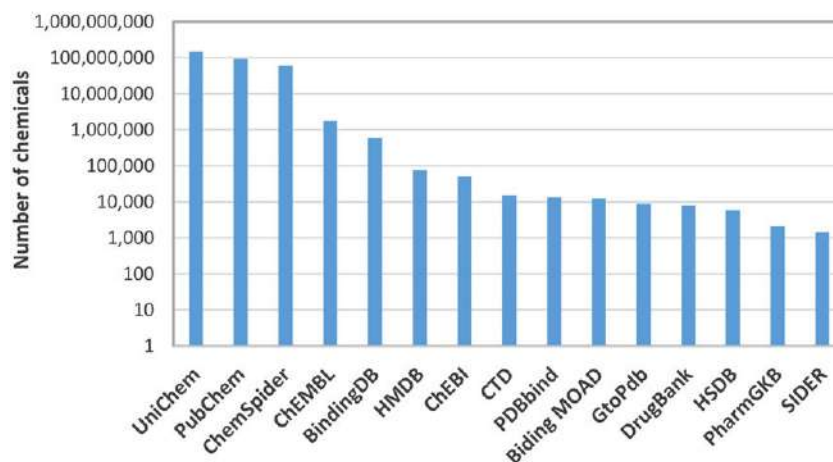


Fig. 2 Number of chemical structures contained in the databases discussed in the present paper (as of June 2017).

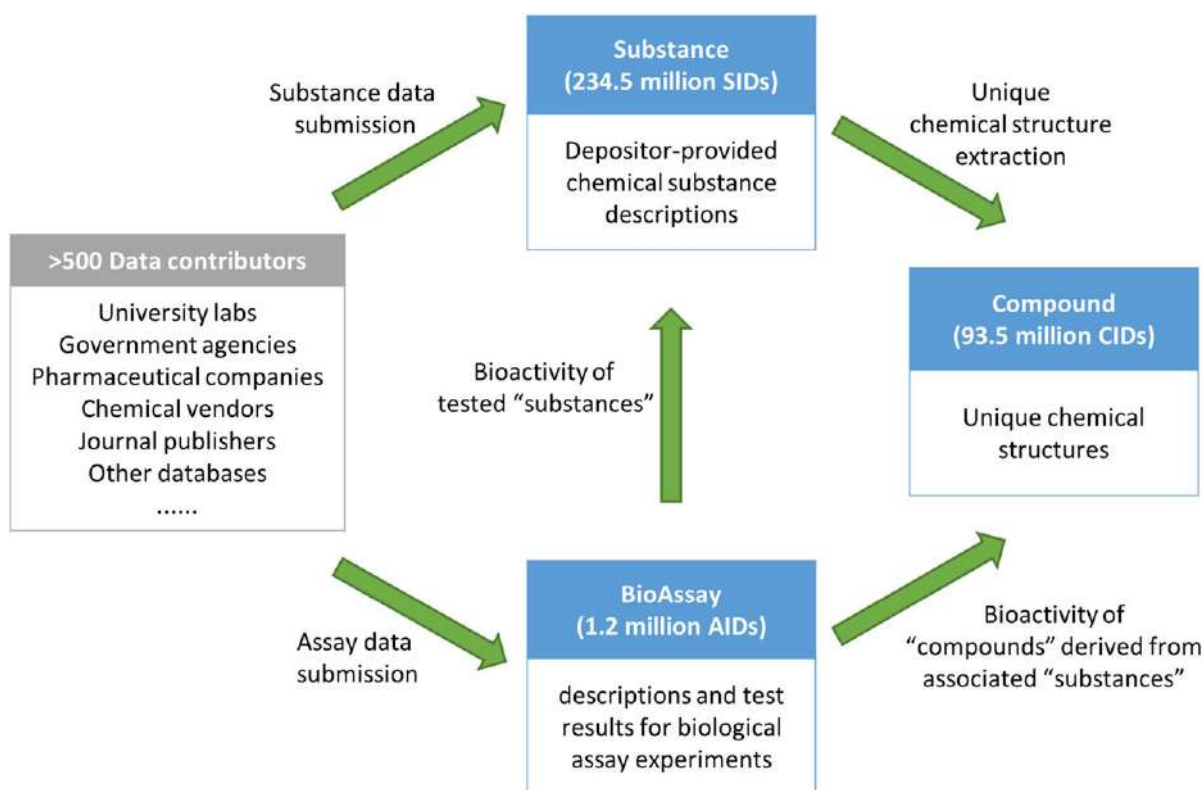


Fig. 3 Data organization in PubChem. PubChem organizes its data into three inter-linked databases called Substance, Compound, and BioAssay. SID, CID, and AID are record identifiers used in the Substance, Compound, and BioAssay databases, respectively.

entities, including siRNAs, miRNAs, carbohydrates, lipids, peptides, chemically modified macromolecules, and many others. These data are provided by various contributors, including government agencies, university labs, pharmaceutical companies, chemical vendors, publishers and a number of chemical biology resources. Most of the chemical databases discussed in this paper also contribute their data to PubChem.

As shown in **Fig. 3**, PubChem organizes its data into three inter-linked databases: Substance, Compound, and BioAssay (Kim *et al.*, 2016a) (see the Relevant Websites section). The Substance database contains chemical substance descriptions submitted by individual data depositors. Unique chemical structures are extracted from the Substance database and stored in the Compound database. The BioAssay database contains descriptions and results of biological assay experiments performed on chemical

PubChem	- Served as a central data repository for the NIH's Molecular Libraries Program (MLP) - Mostly high-throughput screening (HTS) data from NIH's MLP and other HTS projects. - Also contains literature-extracted screening data submitted by data contributors.
ChEMBL	- Bioactivity data for binding, functional, and ADMET assays. - Extracted from scientific articles in medicinal chemistry and natural product domains. - Also contains bioactivity data from deposited data sets.
BindingDB	- Quantitative binding affinity data extracted from scientific articles and patent documents. - Also captures key experimental conditions such as temperature, pH, and buffer composition. - Focuses on complexes between potential drug targets and drug-like small molecules.
PDBbind	- Quantitative binding affinity data extracted from scientific articles. - Covers protein-ligand complexes stored in PDB (with any resolution). - Also compiles the refined and core sets to provide high-quality data.
Binding MOAD	- Quantitative binding affinity data extracted from scientific articles. - Focuses on protein-ligand complexes stored in PDB with a resolution of 2.5 Å or better. - Provides a non-redundant data set that removes biases due to overrepresented proteins in PDB.

Fig. 4 Comparison of PubChem and other databases that provide bioactivity data of small molecules.

substances. The records in the Substance, Compound, and BioAssay databases are called substances, compounds, and bioassays, respectively. Similarly, the record accessions used for the respective PubChem databases are the Substance ID (SID), Compound ID (CID), and Assay ID (AID). Currently, PubChem contains more than 234 million depositor-provided substances, 93 million unique compounds, and 233 million bioactivity test results from 1.25 million bioassays, covering more than 10,000 protein target sequences.

Although this article describes PubChem under Section Large-Scale Data Aggregators, it should be emphasized that PubChem contains the largest amount of bioactivity data available in the public domain. These data are primarily generated from high-throughput screening (HTS) experiments, because PubChem served as a central repository for the now-concluded NIH's Molecular Libraries Program (MLP). However, it also contains a substantial amount of high-quality bioactivity data extracted from research articles and patent documents, thanks to data contributions by many bioactivity databases, including ChEMBL, BindingDB, and PDBbind (to be discussed later in the present paper) (Kim *et al.*, 2016b). In Fig. 4, the PubChem BioAssay database is compared with other bioactivity databases discussed in the present article.

ChemSpider

Another example of large-scale chemical data aggregators in the public domain is ChemSpider, owned by the Royal Society of Chemistry (RSC) (Williams and Tkachenko, 2014; Pence and Williams, 2010). ChemSpider provides information for ~59 million unique chemical structures integrated from ~500 data sources. To improve the accuracy of the data, it uses a crowdsourcing approach, in which registered users can enter information as well as annotate and curate the records. Especially, users can upload various spectral data for chemicals, including infrared, Raman, nuclear magnetic resonance (NMR), and mass spectra, and ChemSpider displays these spectra in an interactive visualization widget that allows zooming and expansion.

ChemSpider extends the crowdsourcing capabilities to chemical reactions and syntheses, by providing a database service called ChemSpider SyntheticPages (CSSP; see the Relevant Websites section). CSSP is essentially a publishing platform through which a short article about a chemical synthesis procedure is submitted by the user and published after rapid review by one or two editorial board members (typically within 48 h). Once published, it can be commented on by the community. All published articles are assigned digital object identifiers (DOI), making them searchable and citable.

Bioactivity Databases

ChEMBL

ChEMBL is a large-scale bioactivity database developed by the European Bioinformatics Institute (EBI), a part of the European Molecular Biology Laboratory (EMBL) (Gaulton *et al.*, 2012, 2017; Bento *et al.*, 2014). The majority of the bioactivity data in

ChEMBL are manually extracted from the full text of peer-reviewed scientific articles published in a variety of journals in medicinal chemistry and natural product domain. From each publication, the details of the compounds tested, the assays performed and any target information for these assays are extracted. These data are further curated and standardized to maximize their quality and utility across a wide range of chemical biology and drug-discovery research problems.

ChEMBL regularly updates its data, with new releases every 3–4 months. The current release of ChEMBL (release 23) provides information extracted from >67,000 publications and ~50 deposited data sets. In total, it contains >14 million bioactivity values for >1.7 million unique compounds from >1.3 million assays. These assays cover more than 11,000 targets (including ~4300 human protein targets).

While ChEMBL's primary focus is on the bioactivity data extracted from the medicinal chemistry literature, it also collects bioactivity data and annotations from various sources. For example, ChEMBL contains a subset of PubChem's bioactivity data (for confirmatory assays with dose-response endpoints). In addition, bioactivity data extracted by BindingDB from patent documents are integrated into ChEMBL (see Section BindingDB below). Recently, ChEMBL's data scope has expanded to several new areas, including data sets from neglected disease screening, crop protection data for agrochemicals, and drug metabolism and disposition data (Gaulton *et al.*, 2017).

It is worth mentioning more about data exchange between ChEMBL and PubChem (Kim, 2016; Gaulton *et al.*, 2012; Kim *et al.*, 2016a). ChEMBL focuses on compiling bioactivity data for drug-like small molecules that can perturb a biological system in a dose-response way. With that said, the majority of PubChem's bioassay data are not suitable for ChEMBL's data collection, because they were generated from HTS experiments that measured bioactivity of a large number of molecules typically at a "single" concentration. However, a small portion of PubChem assays (confirmatory and panel assays with dose-response endpoints) are within the scope of ChEMBL's data coverage, and have been loaded into ChEMBL. In turn, all ChEMBL assays derived from literature are integrated into PubChem.

BindingDB

BindingDB is a database of experimental protein-ligand affinity data, focusing primarily on proteins considered to be drug targets and their interactions with drug-like small molecules (Gilson *et al.*, 2016; Liu *et al.*, 2007). It currently contains more than 1.3 million binding data between ~7000 protein targets and ~600,000 small molecules. Many of these data are manually curated from scientific journals in the chemical biology and biochemistry domains. BindingDB captures not only the quantitative affinity data (e.g., IC_{50} , K_i , and K_d) but also important experimental conditions such as the temperature, pH, and buffer composition. In addition, it extracts quantitative binding data from patent documents. This patent-derived bioactivity data set is a valuable resource, considering that most of them are not reported in journal articles and that curation efforts in other databases (e.g., SureChEMBL; Papadatos *et al.*, 2016) usually focuses on chemical "mentions" in patent documents without capturing quantitative binding affinity values.

BindingDB also contains affinity data collected from other public databases, such as PubChem (Wang *et al.*, 2017) and ChEMBL (Gaulton *et al.*, 2017), while BindingDB's data are integrated into them in turn. Because BindingDB aims to collect binding affinity data for protein-ligand complexes, it imports ChEMBL's binding assay data but excludes those from functional and ADMET assays. Similarly, PubChem's confirmatory assays with binding affinity data are integrated into BindingDB, whereas PubChem's primary screening data are not imported in BindingDB because they are mostly HTS data generated at a single concentration.

PDBbind

The PDBbind database (Liu *et al.*, 2015) provides experimentally measured binding data for the biomolecular complexes archived in the Protein Data Bank (PDB) (Berman *et al.*, 2003). While PDBbind was originally created at the University of Michigan in 2004 (Wang *et al.*, 2004), it has been maintained and further developed since 2007 at the Shanghai Institute of Organic Chemistry. The biomolecular complexes covered by PDBbind include: (1) complexes between proteins and small molecule ligands; (2) complexes between nucleic acids and small molecule ligands; (3) complexes between two proteins; and (4) complexes between protein and nucleic acid.

High-quality data sets are required for developing and validating molecular docking and scoring functions for structure-based drug design. However, the data quality in PDBbind varies in terms of the resolution of the protein-ligand structures as well as the accuracy of binding affinities. To address this issue, PDBbind provides the "refined set," which consists of high-quality protein-ligand complexes selected by applying a set of rules that ensure the quality of the complex structures and binding data as well as the biological/chemical nature of the complex (Li *et al.*, 2014).

Whereas the refined set contains high-quality data, it is not appropriate to use as a standard benchmark set for evaluating docking/scoring methods, because it suffers from a substantial sample redundancy. For example, ~10% of the complexes in the refined set are those with HIV-1 protease (Liu *et al.*, 2015). Therefore, the refined set was further reduced to the "core set" through a systematic, non-redundant sampling after grouping proteins into clusters with a 90% sequence identity cut off (Li *et al.*, 2014). Relative to these refined and core sets, the entire PDBbind data set is called the "general" set.

PDBbind updates its data annually to keep up with the growth of PDB, along with updated refined and core sets. Currently, PDBbind (v.2016) contains the binding data of 16,179 biomolecular complexes, including 13,308 protein-ligand, 118 nucleic

acid-ligand, 777 protein-nucleic acid, and 1976 protein-protein complexes. The refined set of this release contains 4057 protein-ligand complexes, which are further reduced to the core set of 290 complexes in 58 protein clusters.

Binding MOAD

Binding MOAD provides high-quality structures of protein-ligand complexes derived from PDB (with a resolution of 2.5 Å or better) and, where available, their experimental binding affinity data (K_d , K_i , and IC_{50}) (Ahmed *et al.*, 2015; Benson *et al.*, 2008; Hu *et al.*, 2005). The data in Binding MOAD are validated through manual curations of the references cited within the PDB structure file. The proteins are clustered into different families based on 90% sequence identity, and one representative protein is selected from each family (typically, the one complexed with the tightest-binding ligand). This allows BindingDB to provide a non-redundant data set that removes biases from proteins heavily represented in the PDB. The current version of Binding MOAD (released in 2014) contains 25,769 complexes between 12,440 distinct ligands and 7599 distinct protein families, along with 9142 binding affinities.

Binding MOAD and PDBbind have similar goals to each other in the sense that both aim to provide binding data for biomolecular complexes archived in PDB, which is a key difference of the two databases from BindingDB. All three databases provide binding affinity information for protein-ligand complexes, but only a small fraction of protein-ligand affinity data in BindingDB have corresponding crystal structures in PDB. On the other hand, Binding MOAD and PDBbind do have some notable differences from each other. For example, whereas Binding MOAD considers only high-quality structures with a resolution of 2.5 Å or better, PDBbind covers all complexes regardless of the resolution. Instead, PDBbind compiles the refined and core sets to provide high-quality data (see Section PDBbind above). In addition, BindingDB focuses on protein-ligand complexes, but PDBbind has a broader coverage, including protein-ligand complexes as well as other types of complexes (e.g., protein-protein complexes).

Databases of drug information

DrugBank

DrugBank offers comprehensive information on more than 8000 small-molecule drugs and biotech drugs (Law *et al.*, 2014; Knox *et al.*, 2011; Wishart *et al.*, 2006, 2008). While the initial release of DrugBank provided information on selected U.S. Food and Drug Administration (FDA)-approved drugs and their targets, it has been expanding in the depth and breadth of its data and the current release (DrugBank 5.0) contains ~2000 FDA-approved (human and veterinary) drugs, ~1000 investigational drugs (in phase I, II, and III clinical trials), and ~5000 experimental drugs (experimentally shown to bind to specific proteins in humans as well as known bacterial, viral, and fungal pathogens). It also covers hundreds of withdrawn drugs, illicit drugs, and nutraceuticals. For each drug, DrugBank compiles a wide range of information, including their targets, metabolic enzymes, transporters, carriers, drug-drug and drug-food interactions, pharmacogenomic and pharmacoeconomic data, and many others. Because most of these data are manually curated by domain experts and biocurators, DrugBank serves as the source of reference drug data for many chemical databases such as PubChem, PharmGKB, and ChEBI.

Notably, DrugBank has a rich collection of drug metabolism information, including 1200 drug metabolites and 1300 drug metabolism reactions. In addition, a large amount of ADMET (absorption, distribution, metabolism, excretion and toxicity) data are recently added to DrugBank, including Caco2 cell permeability, blood-brain barrier permeability, human intestinal absorption levels, P-glycoprotein activity, CYP450 substrate preferences, renal transport activity, carcinogenicity, Ames test activity, human Ether-a-go-go Related Gene (hERG) activity, etc. DrugBank also provides reference mass spectrometry (MS) and NMR spectra of pure drugs and their metabolites, along with specialized tools for visualizing and searching these spectral data. These spectral data and tools are very useful for identification and/or quantification of compounds in biological matrices.

IUPHAR/BPS Guide to PHARMACOLOGY

The Guide to PHARMACOLOGY (GtoPdb) is a pharmacological information portal that provides expert-curated information on molecular interaction between drug targets and their selective ligands (Southan *et al.*, 2016; Pawson *et al.*, 2014). It was created from collaborative efforts between the International Union of Basic and Clinical Pharmacology (IUPHAR) and the British Pharmacological Society (BPS), by curating and integrating data from the IUPHAR-DB and the published BPS “Guide to Receptors and Channels” (GRAC) compendium.

The early versions of GtoPdb covered the G protein-coupled receptors (GPCRs), ion channels, and nuclear hormone receptors (NHRs) because many drug targets belong to these protein classes. However, the scope of GtoPdb has been expanded to other protein classes, including catalytic receptors, transporters, proteases, kinases, enzymes, etc., with the aim of capturing the likely targets of future medicines. While focused on human proteins, GtoPdb also contains information on mouse and rat orthologs because rodent binding data are most commonly encountered in scientific articles. For each target, GtoPdb provides selective ligands and their quantitative binding data (e.g., K_i , IC_{50} , or K_d), manually curated from scientific research articles. These ligands may be endogenous (e.g., metabolites, hormones, neurotransmitters, and cytokines) or exogenous (e.g., drugs, research leads, toxins, and probes). Currently, GtoPdb covers 2813 targets and 8900 ligands.

GtoPdb uses a unique curation strategy based on the collaboration between in-house curators and the IUPHAR Committee on Receptor Nomenclature and Drug Classification (NC-IUPHAR), which has a network of hundreds of domain experts organized into 60 subcommittees specialized in individual target families. These subcommittees guide target and ligand annotation in GtoPdb, by identifying key pharmacological properties of each target and its quantitative bioactivity data with important ligands. They also monitor de-orphanization of receptors (i.e., identifying new endogenous ligands). In addition, GtoPdb incorporates nomenclature recommendations from NC-IUPHAR. It also adopts and disseminates NC-IUPHAR-derived standards and terminology in quantitative pharmacology.

Pharmacogenomics Knowledge Base (PharmGKB)

The Pharmacogenomics Knowledge Base (PharmGKB) offers information on the impact of genetic variations on drug responses (Whirl-Carrillo *et al.*, 2012; Hernandez-Boussard *et al.*, 2008; Hewett *et al.*, 2002). It contains the relationships between drugs, diseases/phenotypes and genes involved in pharmacodynamics (PD) and pharmacokinetics (PK), extracted from literature using manual curation and natural language processing (NLP) techniques. Importantly, PharmGKB extracts from scientific publications the reported associations between gene variants (single-polymorphism or haplotype) and drug phenotypes, along with key study parameters such as the study size, population ethnicity, and statistics (e.g., *p*-values and odd ratios). In addition, it collects information on drug-centered pathways that contain the genes involved in the PD/PK of a particular drug. These pieces of information are used to create Very Important Pharmacogene (VIP) summaries that provide a concise overview of critical genes involved in drug response, with links to the literature, gene variant and haplotype details, and relevant drugs.

In PharmGKB, multiple annotations derived from different publications may exist for a given association between a single genetic variant and a drug phenotype. PharmGKB combines these annotations into a “clinical annotation” for that variant-drug-phenotype association, which essentially provides a genotype-based summary of the clinical impact of a genomic variant based on the information in PharmGKB. Each clinical annotation is scored for the strength of supporting evidence on a scale of 1A to 4, with 1A being the strongest evidence.

PharmGKB plays an important role in clinical implementation of pharmacogenomics knowledge. With the Pharmacogenomics Research Network (PGRN), PharmGKB participates in the Clinical Pharmacogenomics Implementation Consortium (CPIC), which publishes genotype-based drug guidelines to help clinicians optimize drug therapy using available genetic test results (Caudle *et al.*, 2014; Relling and Klein, 2011) (see the Relevant Websites section). Upon publication in a scientific journal after peer-review, these guidelines are simultaneously posted to PharmGKB with supplemental data and updates.

SIDER

In the drug discovery and development process, it is critical to identify potential undesired side effects of drug candidates as early as possible in order to reduce the late-stage attrition rate. Therefore, many attempts have been made to predict potential adverse drug reactions (ADRs) and elucidate the underlying biological mechanisms of the ADRs. In addition, several databases have been developed to provide information on ADRs and related drug targets (Juan-Blanco *et al.*, 2015; Cai *et al.*, 2015). One of them is the “Side Effect Resource” (SIDER) database, which currently contains 140,064 drug-ADR pairs between 1430 drugs and 5880 ADRs (Kuhn *et al.*, 2010, 2016). These data are extracted from drug labels and package inserts (including the Structured Product Labels from U.S. FDA) through a dictionary-based named entity recognition (NER) approach. SIDER also uses NLP to extract drug indications from the package inserts, which are used to identify false positives in the ADRs detected through NER.

Databases of chemical toxicity

Comparative Toxicogenomics Database (CTD)

The Comparative Toxicogenomics database (CTD) is a toxicogenomics resource that provides information on interactions between chemicals and genes/proteins and their relationships to diseases (Davis *et al.*, 2013, 2015, 2017). The core contents of CTD are chemical-gene, chemical-disease, and gene-disease interactions, extracted from scientific articles and manually curated by professional biocurators using controlled vocabularies and structured notation. This information is integrated with data from selected external resources, including Gene Ontology (GO) (Carbon *et al.*, 2017), Reactome (Fabregat *et al.*, 2016), NCBI Taxonomy and Gene, BioGRID (Chatr-Aryamontri *et al.*, 2017), and Medical Subject Headings (MeSH).

While the primary focus of CTD has been on environmental chemicals, it also contains a substantial amount of curated data from 88,000 articles that describe the toxic actions of pharmaceuticals on cardiovascular, neurological, hepatic and renal systems, thanks to the collaboration with Pfizer (Davis *et al.*, 2013).

In addition to curated interactions among the chemical-gene-disease triads, CTD also provides inferred relationships between them. For example, if a chemical has a curated interaction with a gene and if that gene has a curated association with a disease, the chemical is inferred to have a relationship with the disease and a putative link between them is created. This helps users to generate testable hypotheses. To assist users in navigating and prioritizing these inferences, CTD computes a ranking score for each inferred relationship, based on the topology of a local network, in which chemicals, genes, and diseases are represented as nodes connected

by edges corresponding to the curated interactions (King *et al.*, 2012). Integration with external annotations yields additional inferred relationships.

CTD currently contains more than 1.7 million manually curated interactions (> 1.5 million chemical-gene interactions, > 34 thousand gene-disease associations, and > 200 thousand chemical-disease associations) for 14,975 chemicals, 43,537 genes, and 6416 diseases extracted from 119,720 peer-reviewed scientific papers. It also has 20 million inferred gene-disease relationships and 1.9 million inferred chemical-disease relationships. In total, CTD offers more than 31 million toxicogenomic connections useful for analysis and hypothesis development.

Hazardous Substances Data Bank (HSDB)

The Hazardous Substances Data Bank (HSDB) at the NLM provides toxicological information for more than 5800 potentially hazardous chemicals, including heavy metal compounds, pollutants, herbicides, insecticides, radionuclides, solvents, pharmaceuticals, dietary supplements, venoms, and nanomaterials (Fonger *et al.*, 2014). Chemicals for HSDB record creation and update are nominated by NLM staff, scientific and regulatory agencies, advisory groups, and the public. These candidate chemicals are evaluated and selected by the HSDB chemical selection team, based on a number of considerations, including the level of toxicity, human and environmental exposure, the amount of production and use, and related factors such as regulatory status in the United States and other countries.

For the selected compounds, HSDB compiles various kinds of information, including (but not limited to) human exposure, industrial hygiene, emergency handling procedures, environmental fate, and regulatory requirements. These data are collected from a core set of books, government documents, technical reports, and journal articles. The collected information is peer-reviewed by the Scientific Review Panel (SRP), which consists of experts in major subject areas within the HSDB's scope.

The data contents in HSDB are labeled with one of three data quality tags: "PEER REVIEWED", "QC REVIEWED", and "UNREVIEWED". The data peer-reviewed by the SRP are tagged as "PEER REVIEWED". The "QC REVIEWED" tag means that a quality control review has been conducted by a HSDB senior reviewer or SRP subcommittee while a full peer review by SRP is pending. The "UNREVIEWED" tag indicates that the data has not been evaluated for scientific accuracy. Because this tag is used in the internal record building process, public HSDB data do not have the "UNREVIEWED" tag unless there is a special circumstance.

Databases in other categories

Human Metabolome Database (HMDB)

The Human Metabolome Database (HMDB) is a major metabolomics database, which offers comprehensive information on human metabolites, including their abundance, associated proteins, and disease-related properties (Wishart *et al.*, 2007, 2009, 2013). Currently, HMDB (version 3.6) contains detailed information on 52,658 metabolites, which can be classified into two groups: detected metabolites and expected metabolites. The detected metabolites are further divided into two groups: (1) detected and quantified and (2) detected but not quantified. The expected metabolites are those which could be or will be detected in the future, based on the current knowledge of human biochemistry along with human food and drug consumption patterns. Metabolites are further classified as being endogenous, microbial, drug derived, a toxin/pollutant, food derived, or different combinations of these.

The HMDB provides information on the experimentally determined concentrations of metabolites in various biofluids and/or tissues (e.g., cerebrospinal fluid, serum, urine, saliva) and tissues (e.g., prostate). Disease association and abnormal concentrations were manually curated from literature. In addition, it has reference NMR, tandem mass spectrometry (MS/MS), and gas chromatography-mass spectrometry (GC-MS) spectra. MS and NMR spectra were collected, assigned, and/or annotated by the HMDB curation team. Hundreds of additional annotated/assigned MS and NMR reference spectra were obtained from the BioMagResBank (Ulrich *et al.*, 2008), METLIN (Smith *et al.*, 2005; Tautenhahn *et al.*, 2012), and MassBank (Horai *et al.*, 2010). These spectral data are available for download as image files or in widely used spectral data exchange formats, such as JCAMP-DX (Davies and Lampen, 1993; Lampen *et al.*, 1994), nmrML, and mzML (Deutsch, 2008; Martens *et al.*, 2011), etc, which capture relevant spectral features, spectral collection conditions, assignments, and chemical structure information.

Chemical Entities of Biological Interest (ChEBI)

Chemical Entities of Biological Interest (ChEBI) at the EMBL-EBI is a free online dictionary of "small" molecular entities (Hastings *et al.*, 2013, 2016; De Matos *et al.*, 2010). The term "molecular entity" refers to "any constitutionally or isotopically distinct atom, molecule, ion, ion pair, radical, radical ion, complex, conformer, etc., identifiable as a separately distinguishable entity" (Degtyarenko *et al.*, 2008). While ChEBI focuses on small chemical compounds (either naturally-occurring or synthetic) that affect biological processes of living organisms, it excludes molecules directly encoded by the genome (e.g., nucleic acids, proteins and peptides derived from proteins by cleavage) as a rule.

The current release of ChEBI (Release 152) contains more than 50,000 fully curated entries with another ~50,000 entries that have not been processed yet. For each annotated entry, a wide range of manually curated data items are provided, including chemical structures and chemical structure representations (International Chemical Identifier (InChI) (Heller *et al.*, 2013, 2015), InChIKey, and simplified molecular-input line-entry system (SMILES) strings (Weininger, 1988, 1990; Weininger *et al.*, 1989)),

chemical names (IUPAC names, International Non-proprietary Names (INNs), brand names, and other synonyms), and literature citations. In addition, each entry is extensively cross-referenced to related records in external databases.

An important feature of ChEBI is that each entry in ChEBI is classified within the “ChEBI ontology”. It consists of two main subontologies: (1) a chemical entity ontology in which chemical entities are classified based on shared structural features and (2) a role ontology in which classifies entities based on their activities in biological or chemical systems or their use in applications. Through this ontological classification, ChEBI specifies the relationship between molecular entities (or classes of entities).

UniChem

Chemical information available in the public domain is scattered across many different databases. Identifying equivalent chemical entities from these resources is not a trivial task because individual resources adopt different data models, release schedules, structure standardization rules, chemical nomenclatures, etc. UniChem is a free compound identifier mapping service developed by EMBL-EBI to address this data integration issue (Chambers *et al.*, 2013, 2014; Hersey *et al.*, 2015). It contains standard InChIs (Heller *et al.*, 2013, 2015) for more than 145 million chemical structures stored in > 30 databases, along with pointers between these structures and chemical identifiers from all the individual databases. Therefore, UniChem is very useful to cross-reference equivalent chemical records between different databases.

Originally, UniChem produced mappings between chemicals in different sources on the basis of complete identity between their standard InChIs (i.e., only if their standard InChIs are completely identical) (Chambers *et al.*, 2013). However, this approach appeared too narrow and stringent for some purposes because stereoisomers, isotopes, and salts of otherwise identical molecules could not be related. For this reason, UniChem introduced a service called “Connectivity Search”, in which the equivalence between molecules can be determined at different levels of structural specification (Chambers *et al.*, 2014). This service exploits the layered structural representation of standard InChIs, by matching molecules on the basis of the complete identity of their connectivity layers and then comparing the remaining layers to check the stereochemical and isotopic differences. It also supports mixtures and salts, by using standard InChI sublayers.

UniChem is a highly automated system. When new data are loaded into one source, all links between this source and all the others are automatically calculated and immediately available. UniChem also provides users with a RESTful web service interface for programmatic access to its data.

Closing Remarks

In this article, several public chemical databases useful for cheminformatics and bioinformatics research have been described. Some of these databases like PubChem and ChemSpider contain a very large number of molecules (91 millions and 59 millions, respectively) and provide a wide variety of information, including chemical, physical, and spectral properties, pharmacology, toxicology, metabolism, safety and handling, environmental health and many others. Essentially, these databases are data aggregators, which collect and integrate information from various sources.

Other databases usually offer specialized information for a small number of compounds relevant to narrow focus areas. For example, ChEMBL, BindingDB, PDBbind, and Binding MOAD extract bioactivity data from scientific articles and/or patent documents. DrugBank contains manually curated information for FDA-approved and experimental drugs. GtoPdb provides information on the interaction between drug targets and their ligands, and PharmGKB focuses on pharmacogenomics aspects of drug response. SIDER collects information on the side effect of drugs. CTD and HSDB are specialized in toxicological information, and HMDB compiles metabolomics information. ChEBI provides ontology-based classifications of chemicals and UniChem deals with cross-reference information among different databases.

Although these databases often show substantial overlaps with each other in terms of the compound coverage as well as the type of information they provide, each database does have unique data contents, which can complement the information in other databases. While data exchange between different databases has become a very common practice, this also has raised serious concerns over quality control of public-domain chemistry data. For example, the lack of a standard in chemical structure representation has made individual database developers adopt different chemical structure standardization rules that suit their needs. As a result, it is not trivial to figure out which compound in one database is the same as a compound in another database, often leading to incorrect mapping between compounds from different databases.

Users of public chemical databases should be aware of potential data quality issues (Southan *et al.*, 2013; Kramer *et al.*, 2012; Hu and Bajorath, 2012; Muresan *et al.*, 2011; Tiikkainen *et al.*, 2013; Williams *et al.*, 2012; Williams and Ekins, 2011; Hersey *et al.*, 2015; Akhondi *et al.*, 2012; Fourches *et al.*, 2010; Tiikkainen and Franke, 2012; Williams, 2008b). Therefore, some important papers about the data quality issues are listed in the Further Reading section. In addition, this section contains papers about (1) programmatic access to selected databases (Kim *et al.*, 2015; Davies *et al.*, 2015; Swainston *et al.*, 2016); (2) Resource Description Framework (RDF)-formatted chemistry data for data exchange, sharing, and integration (Willighagen *et al.*, 2013; Jupp *et al.*, 2014; Fu *et al.*, 2015); and (3) commonly used chemical structure representations (Warr, 2011; Heller *et al.*, 2015; Weininger, 1988).

Acknowledgements

This work was supported by the Intramural Research Program of the National Institutes of Health, National Library of Medicine. The author thanks Evan Bolton at PubChem and Bradley Otterson at the NIH Library Editing Service for reviewing this manuscript.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Chemoinformatics: From Chemical Art to Chemistry in Silico. Data Storage and Representation. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Information Retrieval in Life Sciences. Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer. Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer. Natural Language Processing Approaches in Bioinformatics. Rational Structure-Based Drug Design. Small Molecule Drug Design

References

- Ahmed, A., Smith, R.D., Clark, J.J., Dunbar, J.B., Carlson, H.A., 2015. Recent improvements to Binding MOAD: A resource for protein-ligand binding affinities and structures. *Nucleic Acids Research* 43, D465–D469.
- Akhondi, S.A., Kors, J.A., Muresan, S., 2012. Consistency of systematic chemical identifiers within and between small-molecule databases. *Journal of Cheminformatics* 4, 35.
- Benson, M.L., Smith, R.D., Khazanov, N.A., *et al.*, 2008. Binding MOAD, a high-quality protein-ligand database. *Nucleic Acids Research* 36, D674–D678.
- Bento, A.P., Gaulton, A., Hersey, A., *et al.*, 2014. The ChEMBL bioactivity database: An update. *Nucleic Acids Research* 42, D1083–D1090.
- Berman, H., Henrick, K., Nakamura, H., 2003. Announcing the worldwide Protein Data Bank. *Nature Structural Biology* 10, 980.
- Cai, M.C., Xu, Q., Pan, Y.J., *et al.*, 2015. ADReCS: An ontology database for aiding standardization and hierarchical classification of adverse drug reaction terms. *Nucleic Acids Research* 43, D907–D913.
- Carbon, S., Dietze, H., Lewis, S.E., *et al.*, 2017. Expansion of the gene ontology knowledge base and resources. *Nucleic Acids Research* 45, D331–D338.
- Caudle, K.E., Klein, T.E., Hoffman, J.M., *et al.*, 2014. Incorporation of pharmacogenomics into routine clinical practice: The Clinical Pharmacogenetics Implementation Consortium (CPIC) guideline development process. *Current Drug Metabolism* 15, 209–217.
- Chambers, J., Davies, M., Gaulton, A., *et al.*, 2013. UniChem: A unified chemical structure cross-referencing and identifier tracking system. *Journal of Cheminformatics* 5, 3.
- Chambers, J., Davies, M., Gaulton, A., *et al.*, 2014. UniChem: Extension of InChI-based compound mapping to salt, connectivity and stereochemistry layers. *Journal of Cheminformatics* 6, 43.
- Chatr-Aryamontri, A., Oughtred, R., Boucher, L., *et al.*, 2017. The BioGRID interaction database: 2017 update. *Nucleic Acids Research* 45, D369–D379.
- Cheng, T., Pan, Y., Hao, M., Wang, Y., Bryant, S.H., 2014. PubChem applications in drug discovery: A bibliometric analysis. *Drug Discovery Today* 19, 1751–1756.
- Davies, A.N., Lampen, P., 1993. Jcamp-DX for NMR. *Applied Spectroscopy* 47, 1093–1099.
- Davies, M., Nowotka, M., Papadatos, G., *et al.*, 2015. ChEMBL web services: Streamlining access to drug discovery data and utilities. *Nucleic Acids Research* 43, W612–W620.
- Davis, A.P., Grondin, C.J., Johnson, R.J., *et al.*, 2017. The comparative toxicogenomics database: Update 2017. *Nucleic Acids Research* 45, D972–D978.
- Davis, A.P., Grondin, C.J., Lennon-Hopkins, K., *et al.*, 2015. The comparative toxicogenomics database's 10th year anniversary: Update 2015. *Nucleic Acids Research* 43, D914–D920.
- Davis, A.P., Murphy, C.G., Johnson, R., *et al.*, 2013. The comparative toxicogenomics database: Update 2013. *Nucleic Acids Research* 41, D1104–D1114.
- Davis, A.P., Wieggers, T.C., Roberts, P.M., *et al.*, 2013. A CTD–Pfizer collaboration: Manual curation of 88,000 scientific articles text mined for drug-disease and drug-phenotype interactions. *Database (Oxford)* 2013, bat080.
- Degtyarenko, K., De Matos, P., Ennis, M., *et al.*, 2008. ChEBI: A database and ontology for chemical entities of biological interest. *Nucleic Acids Research* 36, D344–D350.
- De Matos, P., Alcantara, R., Dekker, A., *et al.*, 2010. Chemical entities of biological interest: An update. *Nucleic Acids Research* 38, D249–D254.
- Deutsch, E., 2008. MzML: A single, unifying data format for mass spectrometer output. *Proteomics* 8, 2776–2777.
- Fabregat, A., Sidiropoulos, K., Garapati, P., *et al.*, 2016. The reactome pathway knowledgebase. *Nucleic Acids Research* 44, D481–D487.
- Fonger, G.C., Hakkinen, P., Jordan, S., Publicker, S., 2014. The national library of medicine's (NLM) hazardous substances data bank (HSDB): Background, recent enhancements and future plans. *Toxicology* 325, 209–216.
- Fourches, D., Muratov, E., Tropsha, A., 2010. Trust, but verify: On the importance of chemical structure curation in cheminformatics and QSAR modeling research. *Journal of Chemical Information and Modeling* 50, 1189–1204.
- Fu, G., Batchelor, C., Dumontier, M., *et al.*, 2015. PubChemRDF: Towards the semantic annotation of PubChem compound and substance databases. *Journal of Cheminformatics* 7, 34.
- Gaulton, A., Bellis, L.J., Bento, A.P., *et al.*, 2012. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research* 40, D1100–D1107.
- Gaulton, A., Hersey, A., Nowotka, M., *et al.*, 2017. The ChEMBL database in 2017. *Nucleic Acids Research* 45, D945–D954.
- Gilson, M.K., Liu, T.Q., Baitaluk, M., *et al.*, 2016. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research* 44, D1045–D1053.
- Hastings, J., De Matos, P., Dekker, A., *et al.*, 2013. The ChEBI reference database and ontology for biologically relevant chemistry: Enhancements for 2013. *Nucleic Acids Research* 41, D456–D463.
- Hastings, J., Owen, G., Dekker, A., *et al.*, 2016. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research* 44, D1214–D1219.
- Heller, S., McNaught, A., Pletnev, I., Stein, S., Tchekhovskoi, D., 2015. InChI, the IUPAC international chemical identifier. *Journal of Cheminformatics* 7, 23.
- Heller, S., McNaught, A., Stein, S., Tchekhovskoi, D., Pletnev, I., 2013. InChI – The worldwide chemical structure identifier standard. *Journal of Cheminformatics* 5, 7.
- Hernandez-Boussard, T., Whirl-Carrillo, M., Hebert, J.M., *et al.*, 2008. The pharmacogenetics and pharmacogenomics knowledge base: Accentuating the knowledge. *Nucleic Acids Research* 36, D913–D918.
- Hersey, A., Chambers, J., Bellis, L., *et al.*, 2015. Chemical databases: Curation or integration by user-defined equivalence? *Drug Discovery Today: Technologies* 14, 17–24.
- Hewett, M., Oliver, D.E., Rubin, D.L., *et al.*, 2002. PharmGKB: The Pharmacogenetics Knowledge Base. *Nucleic Acids Research* 30, 163–165.
- Horai, H., Arita, M., Kanaya, S., *et al.*, 2010. MassBank: A public repository for sharing mass spectral data for life sciences. *Journal of Mass Spectrometry* 45, 703–714.
- Hu, L.G., Benson, M.L., Smith, R.D., Lerner, M.G., Carlson, H.A., 2005. Binding MOAD (Mother of all databases). *Proteins-Structure Function and Bioinformatics* 60, 333–340.
- Hu, Y., Bajjorath, J., 2012. Growth of ligand-target interaction data in ChEMBL is associated with increasing and activity measurement-dependent compound promiscuity. *Journal of Chemical Information and Modeling* 52, 2550–2558.
- Juan-Blanco, T., Duran-Frigola, M., Aloy, P., 2015. IntSide: A web server for the chemical and biological examination of drug side effects. *Bioinformatics* 31, 612–613.
- Jupp, S., Malone, J., Bolleman, J., *et al.*, 2014. The EBI RDF platform: Linked open data for the life sciences. *Bioinformatics* 30, 1338–1339.
- Kim, S., 2016. Getting the most out of PubChem for virtual screening. *Expert Opinion on Drug Discovery* 11, 843–855.

- Kim, S., Thiessen, P.A., Bolton, E.E., Bryant, S.H., 2015. PUG-SOAP and PUG-rest: Web services for programmatic access to chemical information in PubChem. *Nucleic Acids Research* 43, W605–W611.
- Kim, S., Thiessen, P.A., Bolton, E.E., *et al.*, 2016a. PubChem Substance and Compound databases. *Nucleic Acids Research* 44, D1202–D1213.
- Kim, S., Thiessen, P.A., Cheng, T., *et al.*, 2016b. Literature information in PubChem: Associations between PubChem records and scientific articles. *Journal of Cheminformatics* 8, 32.
- King, B.L., Davis, A.P., Rosenstein, M.C., Wiegiers, T.C., Mattingly, C.J., 2012. Ranking transitive chemical-disease inferences using local network topology in the comparative toxicogenomics database. *Plos One* 7, e46524.
- Knox, C., Law, V., Jewison, T., *et al.*, 2011. DrugBank 3.0: A comprehensive resource for 'Omics' research on drugs. *Nucleic Acids Research* 39, D1035–D1041.
- Kramer, C., Kallioikoski, T., Gedeck, P., Vulpetti, A., 2012. The experimental uncertainty of heterogeneous public K-i data. *Journal of Medicinal Chemistry* 55, 5165–5173.
- Kuhn, M., Campillos, M., Letunic, I., Jensen, L.J., Bork, P., 2010. A side effect resource to capture phenotypic effects of drugs. *Molecular Systems Biology* 6, 343.
- Kuhn, M., Letunic, I., Jensen, L.J., Bork, P., 2016. The SIDER database of drugs and side effects. *Nucleic Acids Research* 44, D1075–D1079.
- Lampen, P., Hillig, H., Davies, A.N., Linscheid, M., 1994. JCAMP-DX for mass-spectrometry. *Applied Spectroscopy* 48, 1545–1552.
- Law, V., Knox, C., Djoumbou, Y., *et al.*, 2014. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Research* 42, D1091–D1097.
- Lipinski, C.A., Litterman, N.K., Southan, C., *et al.*, 2015. Parallel worlds of public and commercial bioactive chemistry data. *Journal of Medicinal Chemistry* 58, 2068–2076.
- Liu, T.Q., Lin, Y.M., Wen, X., Jorissen, R.N., Gilson, M.K., 2007. BindingDB: A web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research* 35, D198–D201.
- Liu, Z.H., Li, Y., Han, L., *et al.*, 2015. PDB-wide collection of binding data: Current status of the PDBbind database. *Bioinformatics* 31, 405–412.
- Li, Y., Liu, Z.H., Li, J., *et al.*, 2014. Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the test set. *Journal of Chemical Information and Modeling* 54, 1700–1716.
- Martens, L., Chambers, M., Sturm, M., *et al.*, 2011. MzML-a community standard for mass spectrometry data. *Molecular & Cellular Proteomics* 10.R110.000133.
- Ma, X.H., Zhu, F., Liu, X., *et al.*, 2012. Virtual screening methods as tools for drug lead discovery from large chemical libraries. *Current Medicinal Chemistry* 19, 5562–5571.
- Muresan, S., Petrov, P., Southan, C., *et al.*, 2011. Making every SAR point count: The development of chemistry connect for the large-scale integration of structure and bioactivity data. *Drug Discovery Today* 16, 1019–1030.
- Nicola, G., Liu, T.Q., Gilson, M.K., 2012. Public domain databases for medicinal chemistry. *Journal of Medicinal Chemistry* 55, 6987–7002.
- Papadatos, G., Davies, M., Dedman, N., *et al.*, 2016. SureChEMBL: A large-scale, chemically annotated patent document database. *Nucleic Acids Research* 44, D1220–D1228.
- Pawson, A.J., Sharman, J.L., Benson, H.E., *et al.*, 2014. The IUPHAR/BPS guide to PHARMACOLOGY: An expert-driven knowledgebase of drug targets and their ligands. *Nucleic Acids Research* 42, D1098–D1106.
- Pence, H.E., Williams, A., 2010. ChemSpider: An online chemical information resource. *Journal of Chemical Education* 87, 1123–1124.
- Relling, M.V., Klein, T.E., 2011. CPIC: Clinical pharmacogenetics implementation consortium of the pharmacogenomics research network. *Clinical Pharmacology & Therapeutics* 89, 464–467.
- Smith, C.A., O'maille, G., Want, E.J., *et al.*, 2005. METLIN – A metabolite mass spectral database. *Therapeutic Drug Monitoring* 27, 747–751.
- Southan, C., Sharman, J.L., Benson, H.E., *et al.*, 2016. The IUPHAR/BPS guide to PHARMACOLOGY in 2016: towards curated quantitative interactions between 1300 protein targets and 6000 ligands. *Nucleic Acids Research* 44, D1054–D1068.
- Southan, C., Williams, A.J., Ekins, S., 2013. Challenges and recommendations for obtaining chemical structures of industry-provided repurposing candidates. *Drug Discovery Today* 18, 58–70.
- Swainston, N., Hastings, J., Dekker, A., *et al.*, 2016. LibChEBI: An API for accessing the ChEBI database. *Journal of Cheminformatics* 8, 11.
- Tautenhahn, R., Cho, K., Uritboonthai, W., *et al.*, 2012. An accelerated workflow for untargeted metabolomics using the METLIN database. *Nature Biotechnology* 30, 826–828.
- Tiikkainen, P., Bellis, L., Light, Y., Franke, L., 2013. Estimating error rates in bioactivity databases. *Journal of Chemical Information and Modeling* 53, 2499–2505.
- Tiikkainen, P., Franke, L., 2012. Analysis of commercial and public bioactivity databases. *Journal of Chemical Information and Modeling* 52, 319–326.
- Ulrich, E.L., Akutsu, H., Doreleijers, J.F., *et al.*, 2008. BioMagResBank. *Nucleic Acids Research* 36, D402–D408.
- Wang, R.X., Fang, X.L., Lu, Y.P., Wang, S.M., 2004. The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry* 47, 2977–2980.
- Wang, Y., Bryant, S.H., Cheng, T., *et al.*, 2017. PubChem BioAssay: 2017 update. *Nucleic Acids Research* 45, D955–D963.
- Warr, W.A., 2011. Representation of chemical structures. *Wiley Interdisciplinary Reviews–Computational Molecular Science* 1, 557–579.
- Weininger, D., 1988. Smiles, a chemical language and information-system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* 28, 31–36.
- Weininger, D., 1990. Smiles. 3. Depict - graphical depiction of chemical structures. *Journal of Chemical Information and Computer Sciences* 30, 237–243.
- Weininger, D., Weininger, A., Weininger, J.L., 1989. Smiles. 2. Algorithm for generation of unique smiles notation. *Journal of Chemical Information and Computer Sciences* 29, 97–101.
- Whirl-Carrillo, M., McDonagh, E.M., Hebert, J.M., *et al.*, 2012. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics* 92, 414–417.
- Williams, A.J., Ekins, S., 2011. A quality alert and call for improved curation of public chemistry databases. *Drug Discovery Today* 16, 747–750.
- Williams, A.J., Ekins, S., Tkachenko, V., 2012. Towards a gold standard: Regarding quality in public domain chemistry databases and approaches to improving the situation. *Drug Discovery Today* 17, 685–701.
- Williams, A., Tkachenko, V., 2014. The royal society of chemistry and the delivery of chemistry data repositories for the community. *Journal of Computer-Aided Molecular Design* 28, 1023–1030.
- Williams, A.J., 2008a. A perspective of publicly accessible/open-access chemistry databases. *Drug Discovery Today* 13, 495–501.
- Williams, A.J., 2008b. Public chemical compound databases. *Current Opinion in Drug Discovery & Development* 11, 393–404.
- Willighagen, E.L., Waagmeester, A., Spjuth, O., *et al.*, 2013. The ChEMBL database as linked open data. *Journal of Cheminformatics* 5, 23.
- Wishart, D.S., Jewison, T., Guo, A.C., *et al.*, 2013. HMDB 3.0 – The Human Metabolome Database in 2013. *Nucleic Acids Research* 41, D801–D807.
- Wishart, D.S., Knox, C., Guo, A.C., *et al.*, 2006. DrugBank: A comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research* 34, D668–D672.
- Wishart, D.S., Knox, C., Guo, A.C., *et al.*, 2008. DrugBank: A knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research* 36, D901–D906.
- Wishart, D.S., Knox, C., Guo, A.C., *et al.*, 2009. HMDB: A knowledgebase for the human metabolome. *Nucleic Acids Research* 37, D603–D610.
- Wishart, D.S., Tzur, D., Knox, C., *et al.*, 2007. HMDB: The human metabolome database. *Nucleic Acids Research* 35, D521–D526.

Further Reading

- Akhondi, S.A., Kors, J.A., Muresan, S., 2012. Consistency of systematic chemical identifiers within and between small-molecule databases. *Journal of Cheminformatics* 4, 35.
- Davies, M., Nowotka, M., Papadatos, G., *et al.*, 2015. ChEMBL web services: Streamlining access to drug discovery data and utilities. *Nucleic Acids Research* 43, W612–W620.
- Fu, G., Batchelor, C., Dumontier, M., *et al.*, 2015. PubChemRDF: Towards the semantic annotation of PubChem compound and substance databases. *Journal of Cheminformatics* 7, 34.
- Heller, S., McNaught, A., Pletnev, I., Stein, S., Tchekhovskoi, D., 2015. InChI, the IUPAC international chemical identifier. *Journal of Cheminformatics* 7, 23.
- Hersey, A., Chambers, J., Bellis, L., *et al.*, 2015. Chemical databases: Curation or integration by user-defined equivalence? *Drug Discovery Today: Technologies* 14, 17–24.

- Jupp, S., Malone, J., Bolleman, J., *et al.*, 2014. The EBI RDF platform: Linked open data for the life sciences. *Bioinformatics* 30, 1338–1339.
- Kim, S., Thiessen, P.A., Bolton, E.E., Bryant, S.H., 2015. PUG-SOAP and PUG-rest: Web services for programmatic access to chemical information in PubChem. *Nucleic Acids Research* 43, W605–W611.
- Swainston, N., Hastings, J., Dekker, A., *et al.*, 2016. libChEBI: An API for accessing the ChEBI database. *Journal of Cheminformatics* 8, 11.
- Tiikkainen, P., Bellis, L., Light, Y., Franke, L., 2013. Estimating error rates in bioactivity databases. *Journal of Chemical Information and Modeling* 53, 2499–2505.
- Warr, W.A., 2011. Representation of chemical structures. *Wiley Interdisciplinary Reviews-Computational Molecular Science* 1, 557–579.
- Weininger, D., 1988. Smiles, a chemical language and information-system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* 28, 31–36.
- Williams, A.J., Ekins, S., 2011. A quality alert and call for improved curation of public chemistry databases. *Drug Discovery Today* 16, 747–750.
- Williams, A.J., Ekins, S., Tkachenko, V., 2012. Towards a gold standard: Regarding quality in public domain chemistry databases and approaches to improving the situation. *Drug Discovery Today* 17, 685–701.
- Willighagen, E.L., Waagmeester, A., Spjuth, O., *et al.*, 2013. The ChEMBL database as linked open data. *Journal of Cheminformatics* 5, 23.

Relevant Websites

<http://www.BindingMOAD.org>
Binding MOAD.

<https://www.bindingdb.org>
BindingDB.

<https://www.ebi.ac.uk/chembl>
ChEMBL.

<https://www.ebi.ac.uk/chebi/>
Chemical Entities of Biological Interest (ChEBI).

<http://www.chemspider.com>
ChemSpider.

<http://cssp.chemspider.com>
ChemSpider SyntheticPages.

<https://ctdbase.org>
Comparative Toxicogenomics Database (CTD).

<https://cpicpgx.org>
Clinical Pharmacogenomics Implementation Consortium (CPIC).

<http://www.drugbank.ca>
DrugBank.

<http://www.guidetopharmacology.org>
Guide To PHARMACOLOGY.

<https://www.toxnet.nlm.nih.gov/newtoxnet/hsdb.htm>
Hazardous Substances Data Bank (HSDB).

<http://www.hmdb.ca>
Human Metabolome Database (HMDB).

<http://www.pdbbind.org.cn>
PDBbind.

<https://www.pharmgkb.org>
Pharmacogenomics Knowledge Base (PharmGKB).

<https://www.pgrn.org>
Pharmacogenomics Research Network (PGRN).

<https://pubchem.ncbi.nlm.nih.gov>
PubChem.

<https://www.ncbi.nlm.nih.gov/pcassay/>
PubChem BioAssay.

<https://www.ncbi.nlm.nih.gov/pccompound/>
PubChem Compound.

<https://www.ncbi.nlm.nih.gov/pccsubstance/>
PubChem Substance.

<http://sideeffects.embl.de>
SIDER.

<https://www.ebi.ac.uk/unicchem>
UniChem.

Biographical Sketch

Sunghwan Kim is a Staff Scientist at the National Center for Biotechnology Information (NCBI), National Library of Medicine (NLM), U.S. National Institutes of Health (NIH). As a computational chemist and cheminformatician, he is actively involved in the PubChem project, which develops and maintains a small-molecule database called PubChem. Specifically, his research has been focused on building and improving “PubChem3D,” which is PubChem’s chemical information resource derived from 3-dimensional (3-D) molecular structures. He holds a MSc in Inorganic Chemistry (from Hanyang University, South Korea) and a PhD in Physical Chemistry (from the University of Georgia at Athens).

Chemical Similarity and Substructure Searches

Nils M Kriege and Lina Humbeck, TU, Dortmund, Germany

Oliver Koch, Technical University of Dortmund, Dortmund, Germany

© 2019 Elsevier Inc. All rights reserved.

Introduction

A fundamental concept in cheminformatics is the assumption that chemically similar molecules share similar bioactivities. Therefore, comparison of molecules and the determination of similarities between molecules is an important and often applied task for the identification and development of bioactive molecules (Humbeck and Koch, 2017). Many different methods and concepts have been developed to solve this essential task and it is crucial to understand and know what they are based on as similarity is not a rigid, globally defined or fixed property but rather reflects the perspective of the investigator. As with so many things in life, similarity is in the eye of the beholder.

Firstly, the molecules have to be represented. This can be achieved by a descriptor, a fingerprint or a structural formula. A descriptor is a number like the molecular masses of two molecules, a fingerprint is a vector representation and a structural formula is a molecular graph. The latter one is an intuitive as well as exact representation where vertices represent atoms and edges represent bonds. Secondly, one can discriminate between different types of searches. For example there are cases where one wants to find the exact same structure, only molecules which contain the query molecule as a substructure or molecules sharing the same fragment. In other cases only molecules that are somehow similar are of interest.

These similarity based concepts can be further divided into similarity based on substructures, pharmacophores, fingerprints, shapes, biological activities and so on. Two molecules that share a similar molecular surface (shape) are expected to be similar. A fingerprint could be a bit representation of the presence or absence of fragments and two molecules with similar fingerprints are also expected to be similar. A pharmacophore is a three dimensional arrangement of molecular features that is important for the interaction with the protein target (Guner, 2002). The three dimensional molecular graph of a query molecule is compared to this spatial arrangement and the molecule fulfills this pharmacophore, if its feature arrangement based on its graph is also similar. The search for a largest possible substructure which is present in two molecules is called MCS (Maximum Common Substructure) search. Nevertheless, often different kinds of representations and searches are used together.

In the following some specific concepts are presented to get an idea of the importance of chemical similarity. Quantitative Structure-Property Relationships (QSPR) and Quantitative Structure-Activity Relationships (QSAR) are two such concepts. One of the first QSPR analyses was done in 1947 and included the Wiener Index as description of molecular properties to predict the boiling points of alkane molecules (Wiener, 1947). A more specialized property – the bioactivity – is analyzed using corresponding QSAR (Leach and Gillet, 2007). As described, it might be valuable to compare the shapes of molecules. Therefore, tools like ROCS (Hawkins *et al.*, 2007) or ShaEP (Vainio *et al.*, 2009) can be used. One can also group molecules regarding their target proteins (Keiser *et al.*, 2007) or correlate descriptors with certain properties to guide synthesis to optimized molecules.

Another often discussed approach are privileged scaffolds, which are scaffolds – meaning same core substructures of different molecules – that show activities on different protein families. Hence, they are interesting starting points for drug design projects and additionally close to the medicinal chemists view of molecules. A tool which enables a scaffold focused view of molecular databases is Scaffold Hunter (Schäfer *et al.*, 2017). One also has to be aware of substructures that interfere with assay systems termed PAINS (Pan Assay Interference Substances) (Baell *et al.*, 2013) or that are unspecific destabilizers of proteins, e.g. aggregators (Irwin *et al.*, 2015).

As mentioned above the topic of similarity and comparability is closely related to the challenge to properly represent the molecule. Often molecules are represented as vectors which make them applicable to a bunch of methods, e.g., QSAR approaches. However, this representation loses information, e.g., regarding the connectivity and structure of molecules. In contrast a graph based representation is closer to the intuitive chemical understanding of molecules. Unfortunately, efficient algorithms are needed to keep the increased calculation times small. In this article we introduce some graph based algorithms to compare molecules and discuss their applications.

Background and Fundamentals

By the middle of the nineteenth century the growing knowledge about chemical atoms and how they combine to form larger structures led to meaningful diagrams, which can be seen as early applications of *graph theory* (Biggs *et al.*, 1986). This section introduces into graph theory as well as molecular graph representation and summarizes graph theoretical fundamentals using modern terms and notation.

Graph Theory

A graph $G = (V, E)$ consists of a finite set $V(G) = V$ of *vertices* and a finite set $E(G) = E$ of *edges*, where each edge connects two distinct vertices. We denote an edge connecting a vertex u and a vertex v by uv or vu , where both refers to the same edge. A *path* of length n is

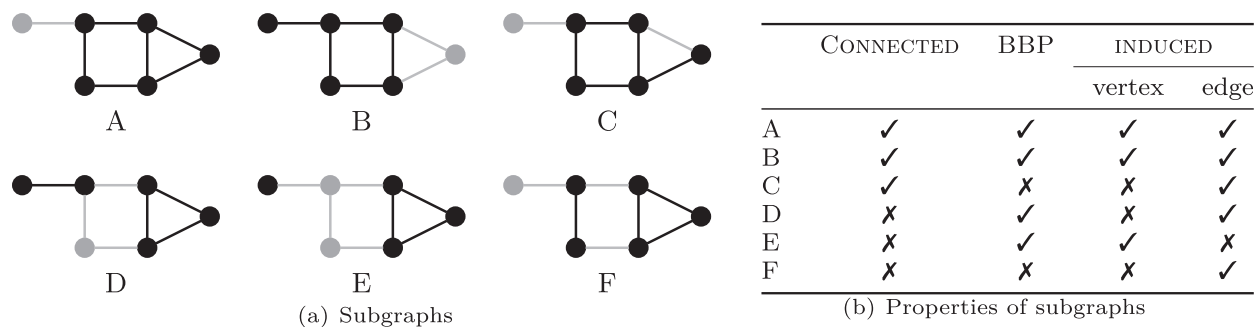


Fig. 1 Examples of subgraphs and their properties, where gray vertices and edges are not contained in the subgraph.

a sequence of vertices $(v_0, \dots, v_n)$ such that $v_i v_{i+1} \in E$ for $0 \leq i < n$. A graph is *connected* if at least one path between any two vertices exists and is *disconnected* otherwise. A *cycle* is a path of length at least 3 with no repeated vertices except $v_0 = v_n$. A graph $G' = (V', E')$ is a *subgraph* of a graph $G = (V, E)$, written $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. A subgraph $G' \subseteq G$ is said to *span* G if $V(G') = V(G)$. Let $V' \subseteq V$ and $E' = \{uv \in E \mid u, v \in V'\}$, then $G' = (V', E')$ is said to be a (*vertex-*)*induced* subgraph of G and is denoted by $G[V']$. Let $E' \subseteq E$ and V' be the set of vertices that appear as endpoint of at least one edge in E' , then $G' = (V', E')$ is said to be an *edge-induced* subgraph of G and denoted by $G[E']$. An edge is a *bridge* if it is not contained in any cycle. A *block* is a maximal connected subgraph that does not contain a bridge. A subgraph $G' \subseteq G$ is *block and bridge preserving* (BBP) if (i) each bridge in G' is a bridge in G , (ii) any two edges in different blocks in G' are in different blocks in G . Fig. 1 shows examples of different subgraphs.

An *isomorphism* between two graphs G and H is a bijection $\psi: V(G) \rightarrow V(H)$ such that $uv \in E(G) \Leftrightarrow \psi(u)\psi(v) \in E(H)$ for all $u, v \in V(G)$. Two graphs G and H are said to be *isomorphic*, written $G \cong H$, if an isomorphism between G and H exists. A *subgraph isomorphism* from a graph G to a graph H is an isomorphism between G and a subgraph $H' \subseteq H$. A *common subgraph isomorphism* between G and H is an isomorphism between subgraphs $G' \subseteq G$ and $H' \subseteq H$.

A graph is called *complete* if every pair of distinct vertices is adjacent. Given a graph $G = (V, E)$, a *clique* is a subset of vertices $C \subseteq V$ such that $G[C]$ is a complete graph. A clique C is said to be *maximum* if there is no clique C' with $|C| < |C'|$. A *tree* is a connected graph containing no cycles. The following concepts intuitively allow to measure how “tree-like” a graph is. A graph G is called a *k-almost tree* if $|E(B)| < |V(B)| + k$ is satisfied for every block B in G . A graph is said to be a *partial k-trees* if it has a tree decomposition of width at most k (see, e.g., Brandstadt et al., 1999, for details). The partial 2-trees are also called *series-parallel graphs*. A graph is *planar* if it admits a drawing in the plane such that no two edges cross; it is called *outerplanar* if it admits such a drawing with every vertex lying on the boundary of the unbounded region.

Graph Representations of Molecules

The *molecular graph* of a chemical compound is the graph, where the vertices represent the atoms and the edges the bonds of the molecule. The vertices of a molecular graph are typically annotated by the element symbols of the atoms they represent and the edges by the bond types. Additional information like a more differentiated atom type classification or 3D coordinates may be used also. An isomorphism between molecular graphs typically has to preserve such labels. Hydrogen atoms may or may not be represented explicitly. Molecular graphs exhibit several characteristic properties.

- The distribution of vertex and edge labels is non-uniform, e.g., the majority of vertices represent carbon atoms and is labeled by ‘C’.
- The maximum degree of molecular graphs is bounded by a small constant (≤ 4 with very few exceptions) as a result of the atom valency.
- The vast majority of molecular graphs are planar, and most are even outerplanar (Horváth et al., 2010). Here, we mean planar in a graph theoretical sense, while otherwise a molecule typically is considered planar if all atoms are on the same plane w.r.t. their 3D arrangement.
- Molecular graphs typically are tree-like. Yamaguchi et al. (2003) computed the tree width of 9712 molecular graphs derived from the LIGAND database (Goto et al., 2002) and found that there is only one graph with tree width 4, while all other have tree width at most 3. Actually, 19.4% of the graphs are trees and 95% are series-parallel graphs.

These properties can be exploited in order to obtain algorithms, which solve subgraph isomorphism problems efficiently for molecular graphs.

Apart from this natural representation so-called *reduced graph* models have been developed, which represent groups of connected atoms by a single vertex (see Birchall and Gillet, 2011, and references therein). Reduced graphs typically contain only a fraction of the vertices and edges of molecular graphs, but may still contain cycles. Simplifying the representation further, Rarey and Dixon (1998) proposed to represent molecules by trees, which encode their hydrophobic fragments and functional groups.

Substructure Search

Given a data set of molecules and a query structure, the *substructure search problem* is to determine the subset of molecules that contain the query structure. Hence, this task requires to decide for each molecular graph G in the data set whether a subgraph isomorphism from the query graph Q to G exists. The subgraph isomorphism problem is well-studied and algorithms have been developed in cheminformatics since the 1950s (Barnard, 1993). Experimental evaluations in the 1980s suggested that the backtracking algorithm by Ullmann (1976) achieves a superior performance to these early algorithms (see Willett, 1999). Since then backtracking algorithms were heavily applied in practice and have been further enhanced. These approaches extend a partial mapping from the vertices of Q to the vertices of G step-by-step and perform backtracking if the current mapping cannot be completed. Most approaches maintain for every vertex of Q a set of eligible candidate vertices of G , which is filtered whenever the mapping is extended. Moreover, the ordering, in which the mapping is extended is a crucial component of this type of algorithm and has an important effect on the running time. The early algorithm by Ullmann (1976) uses elaborate candidate filtering referred to as *refinement*. Later computational less demanding filtering techniques in combination with certain vertex orderings were shown to be more efficient for practical instances. Cordella *et al.* (2004) proposed the VF2 algorithm and experimentally showed it to outperform Ullmann's algorithm on instances from pattern recognition. More recently, it was shown that straight-forward backtracking approaches similar to VF2, which do not manage candidate sets explicitly, but exploit the low degree and diverse vertex labels of molecular graphs, are highly efficient for molecular graphs in practice (Kriege, 2009; Klein *et al.*, 2011; Ehrlich and Rarey, 2012). Moreover, Ullmann (2011) proposed an improved version of his algorithm relying more on search and less on a demanding refinement procedure.

Although the subgraph isomorphism problem is NP-complete for general graphs, the above mentioned backtracking algorithms with exponential worst-case running time are convenient for molecular graphs. Therefore, polynomial-time algorithms, which exist for many graph classes (Marx and Pilipczuk, 2014) (including almost all molecular graphs) are of little practical relevance. However, since chemical databases contain millions of molecules and must be able to answer a large number of substructure queries within a short response time, indexing techniques have been developed. In a preprocessing step an index data structure is build, which then typically allows to process a query as follows. First, in the filtering step the existence of a subgraph isomorphism is ruled out for most of the molecules. Then, only the remaining candidates are tested by a subgraph isomorphism algorithm in the subsequent verification step. We refer the reader to Barnard (1993), and references therein for more details on implementations in chemical information systems and to Klein *et al.* (2011), and references therein for more recent approaches studied in the broader field of graph indexing and mining.

Maximum Common Subgraph Problems

The identification of substructures which two molecules share is related to the maximum common subgraph problem. This problem is independent of the used graph model and is defined as follows. Given two graphs, determine the maximum size of a common subgraph isomorphism between them. This leads to several problem variants, which differ w.r.t. the definition of a size function and the types of allowed subgraphs. The considered subgraphs may be (i) required to be connected, or (ii) allowed to be disconnected. Moreover, they may be required to preserve ring systems, i.e., to be block and bridge preserving, and may be required to be vertex- or edge- induced. This already leads to twelve possible variants, which are summarized in Table 1 together with an adequate size function and a name commonly used (although the terminology in the literature is far from being consistent).

All the above mentioned variants of the maximum common subgraph problem are NP-hard in general graphs. It is suspected that no polynomial-time algorithms for such problems exist. However, several special cases for restricted graph classes are known that allow to obtain polynomial-time algorithms. There is a complex interplay between graph classes, the properties of the desired common subgraph and the complexity of the problem, which is yet not fully understood. Table 2 summarizes known results that are particularly relevant for molecular graphs.

We review classical techniques in Sections "Reduction to the Clique problem" and "Direct branch-and-bound approaches" and summarize polynomial-time algorithms of practical relevance in Section "Polynomial-time algorithms for restricted graph classes".

Algorithms

As a consequence of the practical relevance of the problem maximum common subgraph algorithms have been studied intensively in cheminformatics and other disciplines like pattern recognition (Conte *et al.*, 2004). Since maximum common subgraph

Table 1 Variants of the maximum common subgraph problem with commonly used size functions. In addition, each variant can be restricted to common subgraphs that are connected or block and bridge preserving

Name: Maximum...	Abbrev.	Size	Induced	Connected	Preserve rings
Common Subgraph	MCS	$ V(G) + V(E) $	–	MCCS	BBP-MC(C)S
Common Induced Subgraph	MCIS	$ V(G) $	vertex	MCCIS	BBP-MC(C)IS
Common Edge Subgraph	MCES	$ E(G) $	edge	MCCES	BBP-MC(C)ES

Table 2 Summary on complexity results for the maximum common induced subgraph problem and its variants. Note that the graph classes are related as follows: trees $\subset$ outerplanar graphs $\subset$ series-parallel graphs $\subset$ partial k -trees; and trees $\subset$ k -almost trees $\subset$ partial k -trees (Bodlaender, 1986); (Δ -bounded – the maximum degree is bounded by a constant; * – the result was originally shown for the edge-induced variant)

Input graph class	Constraint	Complexity
Trees	–	NP-hard (Brandenburg, 2000)
	connected	P (Matula, 1978; Droschinsky et al., 2016)
k-almost trees	connected	NP-hard for any $k \geq 1$ (Akutsu, 1993)
Δ -bounded	connected	P (Akutsu, 1993)
Outerplanar graphs	connected	NP-hard (Syslo, 1982)
	biconnected	P (Droschinsky et al., 2017)
	connected, BBP	P (Droschinsky et al., 2017)
Δ -bounded	connected	P (Akutsu and Tamura, 2013)*
Series-parallel graphs	biconnected	P (Kriege and Mutzel, 2014)
	connected, BBP	P (Kriege et al., 2014, 2018)
Partial k-trees	j -connected, $j < k$	NP-hard for any $k \geq 1$ (Brandenburg, 2000)
	k -connected	Open for $k > 2$
Δ -bounded, $k = 11$	connected	NP-hard (Akutsu and Tamura, 2012b)
Δ -bounded	connected	Open for $k \in \{2, \dots, 10\}$

problems are NP-hard unless the input graphs are heavily restricted, general purpose algorithms commonly used in practice have an exponential worst-case running time. These algorithms typically rely on branch-and-bound techniques to eventually solve the problem to optimality, but still differ in their approach. We group these techniques accordingly.

Reduction to the Clique problem

The most common approach in practice is to exploit the one-to-one correspondence between common induced subgraph isomorphisms of the input graphs and cliques in their association graph (also *compatibility graph* (Koch, 2001; Durand et al., 1999), *product graph* (Koch, 2001; Cazals and Karande, 2005), *derived graph* (Barrow and Burstall, 1976), or *weak modular product graph* (Hammack et al., 2011)), which was described by Levi (1973) and later by Barrow and Burstall (1976). For two graphs $G=(V, E)$, $H=(V', E')$, their *association graph* is denoted by $G \nabla H = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = V \times V'$ and $\mathcal{E}$ contains an edge connecting (u, u') , $(v, v') \in \mathcal{V}$ if $u \neq v$, $u' \neq v'$ and $uv \in E \Leftrightarrow u'v' \in E'$. Each vertex of the association graph represents a mapping of a vertex of G to a vertex of H . Two vertices in the association graph are adjacent if their corresponding vertex mappings are compatible, i.e., the involved vertices exhibit the same relation (adjacent or non-adjacent) in both graphs.

Levi (1973) has shown that two graphs G and H have a common induced subgraph with k vertices if and only if there is a clique with k vertices in $G \nabla H$. Further, there is a one-to-one correspondence between the common induced subgraph isomorphisms and cliques in the association graph. Let $C = \{(v_1, v'_1), \dots, (v_k, v'_k)\} \subseteq \mathcal{V}$ be a clique in $G \nabla H$, $U = \{v_1, \dots, v_k\} \subseteq V(G)$ and $U' = \{v'_1, \dots, v'_k\} \subseteq V(H)$, then $\psi(v_i) = v'_i$, for $i \in \{1, \dots, k\}$, is an isomorphism between $G[U]$ and $H[U']$. Consequently, the maximum cliques in the association graph correspond to the isomorphisms between maximum common induced subgraphs. Note that there may be different isomorphisms acting on the same sets of vertices of the two input graphs.

This relation allows to reduce the MCIS problem to the maximum clique problem. There is a multitude of algorithms devised for maximum clique detection, see the surveys by Pardalos and Xue (1994) and Bomze et al. (1999) for an overview. The algorithm by Bron and Kerbosch (1973) enumerates all maximal cliques and is often used in practice (see, e.g., Koch, 2001). If one is only interested in an arbitrary maximum clique or just the size of a maximum clique, branch-and-bound techniques are used to speed up the search. These essentially allow to ignore unfruitful branches in the search tree, whenever an upper bound of the possible solution size within the branch drops below the best known current solution. Consequently, these techniques benefit from heuristics to find a good solution early. One example of these algorithms is due to Wood (1997), which is employed by Stahl et al. (2005). Conte et al. (2007) performed an extensive experimental comparison including several clique detection algorithms. However, the study does not reveal clear evidence for one algorithm being preferable in practice.

In order to preserve labels of molecular graphs with this approach, it suffices to create a vertex in the association graph only if the vertex labels of the input graphs agree (and analogously for edges and edge labels). More generally, a similarity measure on the vertex and edge labels can be defined, which leads to a weighted association graph in which the clique of maximum weight corresponds a maximum common subgraph isomorphism w.r.t. the weight function (Barrow and Burstall, 1976).

Finding edge-induced common subgraphs. Nicholson et al. (1987) proposed a general reduction from MCES to MCIS based on the above approach, which was later applied on various occasions (Tonnelier et al., 1990; Durand et al., 1999; Koch, 2001). Instead of directly creating the association graph of the two input graphs, the association graph of their line graphs is created. The *line graph* $L(G)$ of a graph G is the graph with vertex set $V(L(G)) = E(G)$, in which two vertices are adjacent if and only if the two corresponding edges are adjacent in G . Obviously, two isomorphic graphs have isomorphic line graphs. A classical result states that, except for the graphs depicted in Fig. 2, the reverse holds as well.

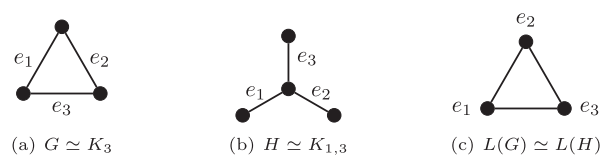


Fig. 2 The non-isomorphic graphs (a) and (b) have the same line graph (c).

Theorem: (Whitney, 1932). Given two connected graphs G and H . If $L(G) \simeq L(H)$ then $G \simeq H$, unless $G \simeq K_{1,3}$ and $H \simeq K_3$ or vice versa (The result was originally formulated without the notion of line graphs and instead directly relates graph isomorphism to edge isomorphism. An *edge isomorphism* is a bijection between edges such that any two edges are adjacent in one graph if and only if the associated edges in the other graph are adjacent).

The single exception is referred to as ΔY -exchange (Raymond and Willett, 2002a; Raymond et al., 2002b) or triode-triangle interchange (Durand et al., 1999).

The family of line graphs is closed under vertex removal, i.e., each induced subgraph of a line graph again is a line graph. Furthermore each vertex-induced subgraph of $L(G)$ directly corresponds to an edge-induced subgraph of G and vice versa. Based on this idea common edge subgraphs can be detected by finding cliques in $L(G) \nabla L(H)$, if ΔY -exchanges are handled adequately: For a common induced subgraph K_3 between two line graphs, it must be verified that the corresponding edge-induced subgraphs in the original graphs are either both K_3 or both $K_{1,3}$. The check can be incorporated in the clique detection algorithm, but the same idea also can be used for other exact MCIS algorithms. The transformation can be extended to cope with labeled graphs by creating labeled line graphs (Raymond et al., 2002b). Tonnelier et al. (1990) as well as Durand et al. (1999) and later Koch (2001) directly define an equivalent edge association graph without using the notion of line graphs.

Finding connected common subgraphs. The above approach can be adapted to find connected common subgraphs. Tonnelier et al. (1990) considered the problem of enumerating all maximal connected common edge subgraph isomorphisms between two graphs G and H . A naïve approach is to enumerate all maximal cliques in the association graph $L(G) \nabla L(H)$ and to keep only the connected solutions. Koch (2001) avoids the exhaustive enumeration by directly restricting the search space. The key idea is to distinguish between two types of edges in the association graph: Edges that are due to common adjacencies are called *c-edges*, all other edges represent the absence of adjacency in both graphs and are referred to as *d-edges*. A clique C in an association graph A is called *c-clique* if $A[C]$ has a spanning subgraph induced by *c-edges*. Koch (2001) showed that a clique in the association graph corresponds to an isomorphism between common connected edge subgraphs if and only if it is a *c-clique*. Based on this observation the algorithm by Bron and Kerbosch (1973) is extended to find only *c-cliques*, which reduces the solution space considerably. Some flaws in the algorithm were later corrected by Cazals and Karande (2005).

Direct branch-and-bound approaches

McGregor (1982) proposed an early branch-and-bound algorithm for MCES, which has been used for classifying chemical reactions (McGregor and Willett, 1981). The approach incrementally extends an initially empty partial mapping between $V(G)$ and $V(H)$, where at each step the i -th vertex of G is mapped to a vertex of H or is omitted from the correspondence. The generated partial mappings form a tree with the empty mapping represented by the root, and children extending the mapping of their parent by an additional vertex pair. In order to reduce the search space no further extension is performed when the size of a total solution can be foreseen not to surpass the best solution found so far. By this means unfruitful branches are cut-off. Therefore, heuristics for finding good solutions early strengthen the effect. Krissinel and Henrick (2004) improved McGregor's algorithm by adding extensions in a prescribed ordering and refining the set of possible extensions, which strengthens pruning. Such branch-and-bound optimizations are an integral part of constraint programming approaches, see Section "General purpose solvers". The algorithm has been used for searching the EBI-MSD ligand database (Boutselakis et al., 2003). The refinement procedure is inspired by a classical subgraph isomorphism algorithm due to Ullmann (1976). Cao et al. (2008) proposed a branch-and-bound approach to MCIS inspired by the VF2 subgraph isomorphism algorithm (Cordella et al., 2004), which involves a new ordering and bound based on maximum bipartite matching. Moreover, the approach allows to constraint the number of connected components of the common subgraph.

General purpose solvers

Constraint programming (CP) is a general approach to solve combinatorial problems, which are stated by a set of variables allowed to take values from a predefined domain and a set of constraints which must be satisfied by a solution. CP solvers typically follow an elaborated branch-and-bound approach. Vismara and Valery (2008) formulated two explicit CP models for MCCES, one inspired by the methods detailed in Section "Reduction to the Clique problem", and an original one based on a subdivision graph. The resulting problem again corresponds to a variant of the clique problem in a product graph with additional constraints. Ndiaye and Solnon (2011) proposed improved CP models for MCIS and MCES. In a recent experimental comparison McCreesh et al. (2016) showed that clique-based approaches for MCIS and MCCIS remain competitive with state-of-the-art CP solvers when using modern clique detection algorithms, in particular for labeled graphs. Moreover, maximum common subgraph problems have been formulated as integer linear programs (see Manić et al., 2009; Bahiense et al., 2012; Piva and de Souza, 2012). These approaches support to incorporate additional side constraints easily, but are not yet competitive regarding running time with clique-based algorithms (Bahiense et al., 2012).

Polynomial-time algorithms for restricted graph classes

Polynomial-time algorithms for MCCIS/MCCES in trees have been pioneered by J. Edmonds and Matula in the 1960s. They rely on solving a series of maximum bipartite matching instances (Matula, 1978). Only recently it has been shown by Droschinsky *et al.* (2016) that the approach can be implemented in cubic time. Akutsu (1993) showed that MCCES in k -almost trees is NP-hard for any $k \geq 1$, but can be solved in polynomial time in almost trees of bounded degree. Moreover, MCCIS can also be solved in polynomial time when one of the input graphs is a bounded-degree partial k -tree and the other is a connected graph with a polynomial number of possible spanning trees (Yamaguchi *et al.*, 2004). Solving BBP-MCCES in outerplanar graphs can be achieved in polynomial time (Schietgat *et al.*, 2013); the same holds for BBP-MCCIS in outerplanar (Droschinsky *et al.*, 2017) and series-parallel graphs (Kriege *et al.*, 2014, 2018; Kriege and Mutzel, 2014). By non-trivial modification of an algorithm for BBP-MCCES Akutsu and Tamura (2012a, 2013) proved that MCCES in outerplanar graphs of bounded degree is polynomial-time solvable. Moreover, Akutsu and Tamura (2012b) have shown that MCCIS and MCCES both are NP-hard in vertex-labeled partial 1-trees of bounded degree. Kann (1992) studied the approximability of maximum common subgraph problems and Abu-Khzam (2014); Abu-Khzam *et al.* (2015, 2017) their parametrized complexity.

Most of the above mentioned results focus on the theoretical complexity and are not promising for practical applications. However, the BBP-MCCIS and BBP-MCCES algorithms have been implemented and applied to molecular graphs in practice. The BBP-MCCES approach by Schietgat *et al.* (2013) runs in $\mathcal{O}(n^4)$ time (Different worst-case bounds have been provided over time in consecutive publications: Starting with $\mathcal{O}(n^7)$ (Schietgat *et al.*, 2007), then $\mathcal{O}(n^5)$ (Schietgat *et al.*, 2008; Schietgat, 2010) and finally $\mathcal{O}(n^2 \sqrt{n})$ (Schietgat *et al.*, 2013). However, it has been shown that the analysis is flawed and the algorithm allows no better bound than $\mathcal{O}(n^4)$ (Droschinsky *et al.*, 2016).) and was shown to outperform the direct backtracking algorithm by Cao *et al.* (2008) in terms of running time in practice (Schietgat, 2010; Schietgat *et al.*, 2013). The BBP-MCCIS of Droschinsky *et al.* (2017) runs in $\mathcal{O}(\Delta n^2)$ time in outerplanar graphs on n vertices with maximum degree Δ and, thus, has a quadratic time complexity in molecular graphs. Experimentally, the technique is shown to outperform the BBP-MCCES approach by Schietgat *et al.* (2013) by orders of magnitude and allows to structurally compare drug-like molecular graphs in the range of a few milliseconds.

Heuristic algorithms

Some applications have strict constraints on running time, but do not require an accurate solution. In this case heuristic algorithms are applicable, which quickly compute a possibly non-optimal solution. We refer the reader to Englert and Kovács (2015), and references therein for heuristic algorithms. Note that the branch-and-bound approaches mentioned above may benefit from heuristics which find good solution quickly.

Variations of the Maximum Common Subgraph Problem

Fuzziness in the detection of maximum common subgraphs is often mentioned as a possible improvement over the strict matching rules (see, e.g., Sheridan and Miller, 1998; Raymond *et al.*, 2003; Hattori *et al.*, 2003). The above mentioned algorithms typically already allow, or can be adapted, to take this into account. The *multiple maximum common subgraph problem* asks for the maximum common subgraph of more than two input graphs. The problem already has been studied by Varkony *et al.* (1979), later by Bayada *et al.* (1992) and more recently by Hariharan *et al.* (2011) and Dalke and Hastings (2013). When large diverse data sets are considered, their common subgraph typically consists of small disconnected fragments, which are not meaningful. Therefore, the *frequent subgraph mining problem* asks for subgraphs that are contained in at least a certain percentage of the data set. This problem and several variants have been studied extensively, see, e.g., Cheng *et al.* (2010).

Applications

Every scientist working with molecules has for a certainty applied common substructure searches based on a maximum common subgraph. In general there are two major application domains for small molecule similarity searches and structure comparisons: prediction/identification and clustering. The former relies on the assumption that the properties of similar structures are similar, which enables the prediction of a property unknown for the molecule of interest but known for a similar one. In the latter application domain a segmentation of a dataset is of interest, e.g., to investigate the diversity. A good overview about possible applications can be found in Humbeck and Koch (2017). In the following some softwares will be mentioned for identification or prediction purposes and clustering tasks. A special open source software called *small molecule subgraph detector* was developed by Rahman *et al.* (2009). It includes different algorithms and chooses the most appropriate one by analyzing the input structures. The software allows to take chemical knowledge into account and finds best solutions w.r.t. atom type match and bond sensitivity.

Identification of Similar Molecules and Prediction of Properties

Raymond *et al.* (2002b) proposed RASCAL for the calculation of graph similarities based on MCES. The approach consists of an initial screening to decide quickly if a certain minimum similarity cannot be reached. Otherwise an elaborated MCES algorithm based on the approach described in Section "Reduction to the Clique problem" is employed, which uses a tailored clique detection

algorithm and exploits symmetries. The approach incorporates results of a long line of research in one MCES algorithm and employs domain-specific heuristics (Raymond *et al.*, 2002a). Similarly, Marialke *et al.* (2007) proposed *graph-based molecular alignment*, where the common subgraph not necessarily has to preserve adjacencies, but the shortest-path distances between atoms. This approach assures consistent arrangements of connected components.

Hartenfeller *et al.* (2012) developed a method which compares molecules using a graph representation to predict similar bioactivity. They collate *in silico* synthesized molecules with known bioactive molecules for the *de novo* design of molecules with similar bioactivity. Furthermore, they use three interesting approaches: A graph representation of the pharmacophoric features of the molecule, reduced graphs and a difference in feature count corrected Tanimoto similarity measure. Although they use a graph representation they do not use MCS algorithms for comparison. Algorithms based on MCS for virtual screening purposes are developed by Lešnik *et al.* (2015), Vainio *et al.* (2009) and Droschinsky *et al.* (2017). The first is called LiSiCA and uses a maximum clique algorithm to detect two and three dimensional similarities of molecular graphs. In contrast to the representation used by Hartenfeller *et al.* (2012) atom types instead of features are modeled by vertices. A product graph is generated and the maximum clique gets detected. The second approach, ShaEP, as well as LiSiCA works with a three dimensional representation, but uses electrostatical potentials and a local shape descriptor as vertex labels, which are not coincident with the heavy atoms of the molecule. Furthermore, they generate a completely connected graph and use the backtracking algorithm of Krissinel and Henrick (2004). ShaEP can also be used for the alignment of molecules.

Substructure searches are often applied to query large databases. For example PubChem supports this kind of approaches for the identification of molecules in their database that contain a molecule drawn on their webpage as substructure. This allows determining other molecules with, e.g., known bioactivity to get a first impression about the possible bioactivity profile of the query molecule.

Classification of Datasets

Due to the high complexity and long run time, clustering of datasets is often based on fingerprint based similarity. However, with increasing computational power and the development of sophisticated algorithms, molecular clustering based on graph representation arouse great interest. Schäfer and Mutzel (2017), for example, present a frequent subgraph based clustering approach termed StruClus including the identification of cluster representatives. Thus, the clustering is chemically intuitive and comprehensible. As the algorithm scales linearly with the dataset size it is capable of clustering datasets of tens of millions of molecules and therefore meets the challenges of ever growing (*in silico*) databases.

A clustering method based on MCES was proposed by Stahl *et al.* (2005) and introduced a similarity measure with penalty terms for inconsistent arrangements of the connected components of the common subgraph in the input graphs. The approach is shown to be able to produce clusterings of molecules which are in a better alignment with their target classes than previous common subgraph or fingerprint based approaches.

Another solution is given by Gardiner *et al.* (2007). They only focus on finding representatives of clusters by finding MCES using the RASCAL algorithm in a non-hierarchically preclustered dataset. The performance is improved by using reduced graphs, where functional groups are represented by a single vertex.

Results and discussion

The general suitability of graph based similarity and structure comparisons was demonstrated in many applications. Nevertheless, one has to choose an appropriate algorithm for a given task and attention should be paid to the graph representation of molecules. Vertices and edges may have different meanings and weighting may be possible. Moreover, reduced graph models allow for more efficient maximum common subgraph computations. Another critical point are the applied similarity or distance measures. Raymond and Willett (2002b) give an overview about common measures and present some intuitive similarity measures between molecules that can be defined on the basis of the size of their maximum common subgraph.

Typically maximum common subgraph algorithms are utilized as a subroutine in more complex applications like clustering. It depends on the application which type of algorithm is most suitable. If it is acceptable to trade accuracy for running time, heuristics may provide an adequate solution. Well-engineered exact algorithms have been developed, which solve the maximum common subgraph problem for most pairs of graphs within milliseconds, but are reported to take a very long time for few tough instances (Stahl *et al.*, 2005). The recent development of exact polynomial-time algorithms for restricted graph classes constitutes a step to overcome this problem. Yet further progress in this direction has to be made to obtain generally applicable algorithms.

Acknowledgement

This work was supported by the German Research Foundation (DFG), priority programme “Algorithms for Big Data” (SPP 1736). O. Koch is funded by the German Federal Ministry for Education and Research (BMBF, Medizinische Chemie in Dortmund, Grant BMBF 1316053).

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Binding Site Comparison – Software and Applications. Biological Database Searching. Biomolecular Structures: Prediction, Identification and Analyses. Chemoinformatics: From Chemical Art to Chemistry in Silico. Drug Repurposing and Multi-Target Therapies. Fingerprints and Pharmacophores. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Protocol for Protein Structure Modelling. Public Chemical Databases. Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications. Rational Structure-Based Drug Design. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow

References

- Abu-Khzam, F.N., 2014. Maximum common induced subgraph parameterized by vertex cover. *Information Processing Letters* 114 (3), 99–103. ISSN 0020-0190. Available at: <http://www.sciencedirect.com/science/article/pii/S0020019013002755>.
- Abu-Khzam, F.N., Bonnet, E., Sikora, F., 2015. On the complexity of various parameterizations of common induced subgraph isomorphism. In: Jan, K., Miller, M., Froncek, D. (Eds.), *Combinatorial Algorithms: 25th International Workshop, IWoca 2014, Duluth, MN, October 15–17, 2014, Revised Selected Papers*, pp. 1–12. Springer International Publishing, Cham. ISBN 978-3-319-19315-1. Available at: <https://doi.org/10.1007/978-3-319-19315-1>.
- Abu-Khzam, F.N., Bonnet, E., Sikora, F., 2017. On the complexity of various parameterizations of common induced subgraph isomorphism. *Theoretical Computer Science*. ISSN 0304-3975. Available at: <http://www.sciencedirect.com/science/article/pii/S0304397517305480>.
- Akutsu, T., 1993. A polynomial time algorithm for finding a largest common subgraph of almost trees of bounded degree. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E76-A (9).
- Akutsu, T., Tamura, T., 2012a. A polynomial-time algorithm for computing the maximum common subgraph of outerplanar graphs of bounded degree. In: Branislav Rován, V., Sassone Widmayer, P. (Eds.), *Mathematical Foundations of Computer Science*, vol. 7464 of *Lecture Notes in Computer Science*. Berlin/Heidelberg: Springer, pp. 76–87. ISBN 978-3-642-32588-5. Available at: https://doi.org/10.1007/978-3-642-32589-2_10.
- Akutsu, T., Tamura, T., 2012b. On the complexity of the maximum common subgraph problem for partial fc-trees of bounded degree. In: Chao, K.-M., Hsu, T.-s., Lee, D.-T. (Eds.), *Algorithms and Computation*, vol. 7676 of *Lecture Notes in Computer Science*. Berlin Heidelberg: Springer, pp. 146–155. ISBN 978-3-642-35260-7. Available at: https://doi.org/10.1007/978-3-642-35261-4_18.
- Akutsu, T., Tamura, T., 2013. A polynomial-time algorithm for computing the maximum common connected edge subgraph of outerplanar graphs of bounded degree. *Algorithms* 6 (1), 119–135. Available at: <http://www.mdpi.com/1999-4893/6/1/119>.
- Baell, J.B., Ferrins, L., Falk, H., Nikolakopoulos, G., 2013. Pains: Relevance to tool compound discovery and fragment-based screening. *Australian Journal of Chemistry* 66 (12), 1483. doi:10.1071/CH13551. ISSN 0004-9425.
- Bahiense, L., Manic, G., Piva, B., de Souza Cid C., 2012. The maximum common edge subgraph problem: A polyhedral investigation. *Discrete Applied Mathematics* 160(18), 2523–2541. ISSN 0166-218X. Available at: <http://www.sciencedirect.com/science/article/pii/S0166218X12000340>. V Latin American Algorithms, Graphs, and Optimization Symposium, Gramado, Brazil, 2009.
- Barrow, H.G., Burstall, R.M., 1976. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters* 4 (4), 83–84.
- Bayada, D.M., Simpson, R.W., Johnson, A.P., Laurencio, C., 1992. An algorithm for the multiple common subgraph problem. *Journal of Chemical Information and Computer Sciences* 32 (6), 680–685. Available at: <http://pubs.acs.org/doi/abs/10.1021/ci00010a015>.
- Biggs, N.L., Lloyd, E.K., Wilson, R.J., 1986. *Graph Theory* 1736–1936. New York, NY: Clarendon Press, (ISBN 0-198-53916-9).
- Birchall, K., Gillet, V.J., 2011. Reduced graphs and their applications in chemoinformatics. *Methods in Molecular Biology* 672, 197–212. Available at: https://doi.org/10.1007/978-1-60761-839-3_8.
- Bodlaender, H.L., 1986. *Classes of graphs with bounded treewidth*. Technical Report RUU-CS-86-22, Department of Computer Science, Utrecht University.
- Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M., 1999. The maximum clique problem. In: Du, D.-Z., Pardalos, P.M. (Eds.), *Handbook of Combinatorial Optimization*, vol. A. Kluwer Academic Publishers.
- Boutselakis, H., Dimitropoulos, D., Fillon, J., *et al.*, 2003. E-msd: The european bioinformatics institute macromolecular structure database. *Nucleic Acids Research* 31, 458–462. (ISSN 1362-4962).
- Brandenburg, F.J., 2000. Subgraph isomorphism problems for k-connected partial k-trees. Unpublished Manuscript.
- Brandstadt, A., Le, Van Bang, Spinrad, J.P., 1999. *Graph Classes: A Survey*. Philadelphia, PA: Society for Industrial and Applied Mathematics, (ISBN 0-89871-432-X).
- Bron, C., Kerbosch, J., 1973. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM* 16, 575–577. ISSN 0001-0782. Available at: <http://doi.acm.org/10.1145/362342.362367>.
- Cao, Y., Jiang, T., Girke, Thomas, 2008. A maximum common substructure-based algorithm for searching and predicting drug-like compounds. *ISMB*.
- Cazals, F., Karande, C., 2005. An algorithm for reporting maximal c-cliques. *Theoretical Computer Science* 349 (3), 484–490.
- Cheng, H., Yan, X., Han, J., 2010. Mining graph patterns. In: Aggarwal, C., Wang, H. (Eds.), *Managing and Mining Graph Data*. US, Boston, MA: Springer, pp. 365–392. ISBN 978-1-4419-6045-0. Available at: https://doi.org/10.1007/978-1-4419-6045-0_12.
- Conte, D., Foggia, P., Sansone, C., Vento, M., 2004. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*. Available at: <https://doi.org/10.1142/S0218001404003228>.
- Conte, D., Foggia, P., Vento, M., 2007. Challenging complexity of maximum common subgraph detection algorithms: A performance analysis of three algorithms on a wide database of graphs. *Journal of Graph Algorithms and Applications* 11 (1), 99–143.
- Cordella, L.P., Foggia, P., Sansone, C., Vento, M., 2004. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (10), 1367–1372. ISSN 0162-8828. Available at: <https://doi.org/10.1109/TPAMI.2004.75>.
- Dalke, A., Hastings, J., 2013. FMCS: A novel algorithm for the multiple mcs problem. *Journal of Cheminformatics* 5 (1), O6. Available at: <https://doi.org/10.1186/1758-2946-5-S1-O6>.
- Droschinsky, A., Kriege, N.M., Mutzel, P., 2016. Faster algorithms for the maximum common subtree isomorphism problem. In: Faliszewski, P., Muscholl, A., Niedermeier, R. (Eds.), *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, vol. 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Dagstuhl, Germany, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, pp. 33:1–33:14. ISBN 978-3-95977-016-3. Available at: <http://drops.dagstuhl.de/opus/volltexte/2016/6447>.
- Droschinsky, A., Kriege, N., Mutzel, P., 2017. Finding Largest Common Substructures of Molecules in Quadratic Time. In: Steffen, B., Baier, C., van den Brand, M., *et al.* (Eds.), *SOFSEM 2017: Theory and Practice of Computer Science. SOFSEM 2017. Lecture Notes in Computer Science*. Cham, vol 10139: Springer International Publishing. ISBN 978-3-319-51963-0. Available at: <https://doi.org/10.1007/978-3-319-51963-024>.
- Durand, P.J., Pasari, R., Baker, J.W., Tsai, C.-c., 1999. An efficient algorithm for similarity analysis of molecules. *Internet Journal of Chemistry* 2, 1–12.
- Ehrlich, H.-C., Rarey, M., 2012. Systematic benchmark of substructure search in molecular graphs – from ullmann to vf2. *Journal of Cheminformatics* 4 (1), 13. Available at: <https://doi.org/10.1186/1758-2946-4-13>.

- Englert, P., Kovács, P., 2015. Efficient heuristics for maximum common substructure search. *Journal of Chemical Information and Modeling* 55 (5), 941–955. Available at: <https://doi.org/10.1021/acs.jcim.5b00036>. PMID: 25865959.
- Gardiner, E.J., Gillet, V.J., Willett, P., Cosgrove, D.A., 2007. Representing clusters using a maximum common edge substructure algorithm applied to reduced graphs and molecular graphs. *Journal of Chemical Information and Modeling*. Available at: <https://doi.org/10.1021/ci600444g>.
- Goto, S., Okuno, Y., Hattori, M., *et al.*, 2002. Ligand: Database of chemical compounds and reactions in biological pathways. *Nucleic Acids Research* 30 (1), 402–404. Available at: <http://nar.oxfordjournals.org/content/30/1/402.abstract>.
- Guner, O., 2002. History and evolution of the pharmacophore concept in computer-aided drug design. *Current Topics in Medicinal Chemistry* 2 (12), 1321–1332. doi:10.2174/1568026023392940. (ISSN 15680266).
- Hammack, R., Imrich, W., Klavzar, S., 2011. *Handbook of Product Graphs. Discrete Mathematics and Its Applications*. Taylor and Francis. (ISBN 9781439813041).
- Hariharan, R., Janakiraman, A., Nilakantan, R., *et al.*, 2011. Multimcs: A fast algorithm for the maximum common substructure problem on multiple molecules. *Journal of Chemical Information and Modeling* 51 (4), 788–806. Available at: <https://doi.org/10.1021/ci100297y>.
- Hartenfeller, M., Zettl, H., Walter, M., *et al.*, 2012. Dogs: reaction-driven de novo design of bioactive compounds. *PLOS Computational Biology* 8 (2), e1002380. doi:10.1371/journal.pcbi.1002380. (ISSN 1553-7358).
- Hattori, M., Okuno, Y., Goto, S., Kanehisa, M., 2003. Heuristics for chemical compound matching. *Genome Information* 14, 144–153.
- Hawkins, P.C.D., Skillman, A.G., Nicholls, A., 2007. Comparison of shape-matching and docking as virtual screening tools. *Journal of Medicinal Chemistry* 50 (1), 74–82. doi:10.1021/jm0603365. (ISSN 0022-2623).
- Horváth, T., Ramon, J., Wrobel, S., 2010. Frequent subgraph mining in outerplanar graphs. *Data Mining and Knowledge Discovery* 21, 472–508. Available at: <https://doi.org/10.1007/s10618-009-0162-1>.
- Humbeck, L., Koch, O., 2017. What can we learn from bioactivity data? Chemoinformatics tools and applications in chemical biology research. *ACS Chemical Biology* 12 (1), 23–35. doi:10.1021/acscchembio.6b00706. (ISSN 1554-8937).
- Irwin, J.J., Duan, D., Torosyan, H., *et al.*, 2015. An aggregation advisor for ligand discovery. *Journal of Medicinal Chemistry* 58 (17), 7076–7087. doi:10.1021/acs.jmedchem. (ISSN 0022-2623).
- John, M., 1993. Barnard. Substructure searching methods: Old and new. *Journal of Chemical Information and Computer Sciences* 33 (4), 532–538.
- Kann, V., 1992. On the approximability of the maximum common subgraph problem. In: *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science, STACS '92*, pages 377–388. London, UK, Springer-Verlag. ISBN 3-540-55210-3. Available at: <http://dl.acm.org/citation.cfm?id=646508.694493>.
- Keiser, M.J., Roth, B.L., Armbruster, B.N., *et al.*, 2007. Relating protein pharmacology by ligand chemistry. *Nature Biotechnology* 25 (2), 197–206. doi:10.1038/nbt1284. (ISSN 1087-0156).
- Klein, K., Kriege, N., Mutzel, P., 2011. CT-index: Fingerprint-based graph indexing combining cycles and trees. In: *IEEE Proceedings of the 27th International Conference on Data Engineering (ICDE)*, pp. 1115–1126, April. doi:10.1109/ICDE.2011.5767909.
- Koch, I., 2001. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science* 250 (1-2), 1–30.
- Kriege, N., 2009. *Erweiterte Substruktursuche in Molekuldatenbanken und ihre Integration in Scaffold Hunter*. Master's thesis, TU Dortmund.
- Kriege, N., Kurpicz, F., Mutzel, P., 2014. On maximum common subgraph problems in series-parallel graphs. In: Jan, K., Miller, M., Froncek, D. (Eds.), *International Workshop on Combinatorial Algorithms, IWCOA 2014*, vol. 8986 of *Lecture Notes in Computer Science*. Springer International Publishing, pp. 200–212. ISBN 978-3-319-19314-4. Available at: https://doi.org/10.1007/978-3-319-19315-1_18.
- Kriege, N., Kurpicz, F., Mutzel, P., 2018. On maximum common subgraph problems in series-parallel graphs. *European Journal on Combinatorics (EJC)* 68, 79–95. <https://doi.org/10.1016/j.ejc.2017.07.012>.
- Kriege, N., Mutzel, P., 2014. Finding maximum common biconnected subgraphs in series-parallel graphs. In: Csuhaj-Varjú, E., Dietzfelbinger, M., Ésik, Zoltán (Eds.), *Mathematical Foundations of Computer Science 2014*, vol. 8635 of *Lecture Notes in Computer Science*. Berlin Heidelberg: Springer, pp. 505–516. ISBN 978-3-662-44464-1. Available at: https://doi.org/10.1007/978-3-662-44465-8_43.
- Krissinel, E.B., Henrick, K., 2004. Common subgraph isomorphism detection by backtracking search. *Software: Practice and Experience* 34 (6), 591–607. Available at: <https://doi.org/10.1002/spe.588>.
- Leach, A.R., Gillet, V.J., 2007. *An introduction to chemoinformatics*, rev ed. edition Dordrecht and London: Springer, (ISBN 1402062915).
- Lešnik, S., Štular, T., Brus, B., *et al.*, 2015. Lisica: A software for ligand-based virtual screening and its application for the discovery of butyrylcholinesterase inhibitors. *Journal of Chemical Information and Modeling* 55 (8), 1521–1528. doi:10.1021/acs.jcim.5b00136. (ISSN 1549-9596).
- Levi, G., 1973. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, Jan. Available at: <http://www.springerlink.com/index/B37657486G578502.pdf>.
- Manić, G., Bahiense, L., Souza, C.D., 2009. A branch&cut algorithm for the maximum common edge subgraph problem. *Electronic Notes in Discrete Mathematics*, 35(0):47–52. ISSN 1571-0653. Available at: <http://www.sciencedirect.com/science/article/pii/S1571065309001620>. *Proceedings of the Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS '09)*.
- Marialke, J., Korner, R., Tietze, S., Apostolakis, J., 2007. Gma: Graph-based molecular alignment (gma). *Journal of Chemical Information and Modeling* 47 (2), 591–601. Available at: <http://dx.doi.org/10.1021/ci600387r>.
- Marx, D., Pilipczuk, M., 2014. Everything you always wanted to know about the parameterized complexity of Subgraph Isomorphism (but were afraid to ask). In: Mayr, E.W., Portier, N. (Eds.), *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 542–553, Dagstuhl, Germany. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. ISBN 978-3-93989765-1. Available at: <http://drops.dagstuhl.de/opus/volltexte/2014/4486>. arXiv:1307.2187.
- Matula, D.W., 1978. Subtree isomorphism in $O(n^{5/2})$. In: Hell, P., Alspach, B., Miller, D.J. (Eds.), *Algorithmic Aspects of Combinatorics*, Vol 2 of *Annals of Discrete Mathematics*. Elsevier, pp. 91–106. Available at: <http://www.sciencedirect.com/science/article/pii/S0167506008703248>.
- McCreesh, C., Ndiaye, S.N., Prosser, P., Solnon, C., 2016. Clique and Constraint Models for Maximum Common (Connected) Subgraph Problems. Cham: Springer International Publishing, ISBN 978-3-319-44953-1. Available at: https://doi.org/10.1007/978-3-319-44953-1_23.
- McGregor, J.J., 1982. Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience* 12 (1), 23–34. ISSN 1097-024X. Available at: <https://doi.org/10.1002/spe.4380120103>.
- McGregor, J.J., Willett, P., 1981. Use of a maximum common subgraph algorithm in the automatic identification of ostensible bond changes occurring in chemical reactions. *Journal of Chemical Information and Computer Sciences* 21 (3), 137–140.
- Ndiaye, S., Ndjih, Solnon, C., 2011. CP Models for Maximum Common Subgraph Problems. Berlin, Heidelberg: Springer, ISBN 978-3-642-23786-7. Available at: https://doi.org/10.1007/978-3-642-23786-7_48.
- Nicholson, V., Tsai, C.-C., Johnson, M., Naim, M., 1987. A subgraph isomorphism theorem for molecular graphs. In *Graph Theory and Topology in Chemistry*, number 51 in *Stud. Physical Theoretical Chemistry*. Elsevier.
- Pardalos, P.M., Xue, J., 1994. The maximum clique problem. *Journal of Global Optimization* 4 (3), 301–328. Available at: <https://doi.org/10.1007/BF01098364>.
- Piva, B., de Souza, C.C., 2012. Polyhedral study of the maximum common induced subgraph problem. *Annals of Operations Research* 199 (1), 77–102. Available at: <https://doi.org/10.1007/s10479-011-1019-8>.
- Rahman, S.A., Bashton, M., Holliday, G.L., Schrader, R., Thornton, J.M., 2009. Small molecule subgraph detector (smsd) toolkit. *J Cheminform* 1 (1), 12. Available at: <https://doi.org/10.1186/1758-2946-1-12>.
- Rarey, M., Dixon, J.S., 1998. Feature trees: A new molecular similarity measure based on tree matching. *Journal of Computer-Aided Molecular Design* 12, 471–490. Available at <https://doi.org/10.1023/A:1008068904628>.

- Raymond, J.W., Blankley, J.C., Willett, P., 2003. Comparison of chemical clustering methods using graph- and fingerprint-based similarity measures. *Journal of Molecular Graphics and Modelling* 21 (5), 421–433. Available at: [https://doi.org/10.1016/S1093-3263\(02\)00188-2](https://doi.org/10.1016/S1093-3263(02)00188-2).
- Raymond, J.W., Gardiner, E.J., Willett, P., 2002a. Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm. *Journal of Chemical Information and Computer Sciences* 42 (2), 305–316.
- Raymond, J.W., Gardiner, E.J., Willett, P., 2002b. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal* 45 (6), 631–644.
- Raymond, J.W., Willett, P., 2002a. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design* 16 (7), 521–533.
- Raymond, J.W., Willett, P., 2002b. Effectiveness of graph-based and fingerprint-based similarity measures for virtual screening of 2d chemical structure databases. *Journal of Computer-Aided Molecular Design* 16, 59–71. Available at: <https://doi.org/10.1023/A:1016387816342>.
- Schäfer, T., Kriege, N., Humbeck, L., *et al.*, 2017. Scaffold hunter: A comprehensive visual analytics framework for drug discovery. *Journal of Cheminformatics* 9 (1), 1075. doi:10.1186/s13321-017-0213-3.
- Schäfer, T., Mutzel, P., 2017. Struclus: Scalable structural graph set clustering with representative sampling. In: *Proceedings of the 13th International Conference on Advanced Data Mining and Applications (ADMA 2017)*, Singapore, accepted for publication.
- Schietgat, L., 2010. Graph-Based Data Mining for Biological Applications. Schietgat, Leander, 2010. Graph-Based Data Mining for Biological Applications. PhD Thesis, Informatics Section, Department of Computer Science, Faculty of Engineering, Hendrik Blockeel and Maurice Bruynooghe (supervisors). Available at: <https://lirias.kuleuven.be/handle/123456789/267094>.
- Schietgat, L., Ramon, J., Bruynooghe, M., 2007. A polynomial-time metric for outerplanar graphs. In: *Frasconi, P., Kersting, K., Koji Tsuda, (Eds.), Mining and Learning with Graphs, MLG 2007 Proceedings Florence, Italy, August 1-3, 2007*, pp. 67–70.
- Schietgat, L., Ramon, J., Bruynooghe, M., 2013. A polynomial-time maximum common subgraph algorithm for outerplanar graphs and its application to cheminformatics. *Annals of Mathematics and Artificial Intelligence* 69 (4), 343–376. Available at: <https://doi.org/10.1007/s10472-013-9335-0>.
- Schietgat, L., Ramon, J., Bruynooghe, M., Blockeel, H., 2008. An efficiently computable graph-based metric for the classification of small molecules. In: *Jean-Francois Boullicaut, M., Berthold Horvath, T. (Eds.), Discovery Science, Vol. 5255 of Lecture Notes in Computer Science*. Berlin / Heidelberg: Springer, pp. 197–209. Available at: http://dx.doi.org/10.1007/978-3-540-88411-8_20.
- Sheridan, R.P., Miller, M.D., 1998. A method for visualizing recurrent topological substructures in sets of active molecules. *Journal of Chemical Information and Computer Sciences* 38 (5), 915–924.
- Stahl, M., Mauser, H., Tsui, M., Taylor, N.R., 2005. A robust clustering method for chemical structures. *Journal of Medicinal Chemistry* 48 (13), 4358–4366. Available at: <https://doi.org/10.1021/jm040213p>.
- Syslo, Maciej M., 1982. The subgraph isomorphism problem for outerplanar graphs. *Theoretical Computer Science* 17 (1), 91–97. Available at: <http://www.sciencedirect.com/science/article/pii/0304397582901335>.
- Tonneller, C., Jauffret, P., Hanser, T., Kaufmann, G., 1990. Machine learning of generic reactions: 3. An efficient algorithm for maximal common substructure determination. *Tetrahedron Computer Methodology* 3 (6), 351–358. Available at: [http://dx.doi.org/10.1016/0898-5529\(90\)90061-C](http://dx.doi.org/10.1016/0898-5529(90)90061-C).
- Ullmann, J.R., 1976. An algorithm for subgraph isomorphism. *Journal of the ACM* 23 (1), 31–42. ISSN 0004-5411. doi: <http://doi.acm.org/10.1145/321921.321925>.
- Ullmann, J.R., 2011. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *Journal of Experimental Algorithmics* 15, 1.6:1.1–1.6:1.64. Available at: <http://doi.acm.org/10.1145/1671970.1921702>.
- Vainio, M.J., Puranen, J.S., Johnson, M.S., 2009. Shaep: Molecular overlay based on shape and electrostatic potential. *Journal of Chemical Information and Modeling* 49 (2), 492–502. doi:10.1021/ci800315d.
- Varkony, T.H., Shiloach, Y., Smith, D.H., 1979. Computer-assisted examination of chemical compounds for structural similarities. *Journal of Chemical Information and Computer Sciences* 19 (2), 104–111. Available at: <http://pubs.acs.org/doi/abs/10.1021/ci60018a014>.
- Vismara, P., Valery, B., 2008. Finding maximum common connected subgraphs using clique detection or constraint satisfaction algorithms. In: *Thi, H.L., Bouvry, P., Dinh, T.P. (Eds.), Modelling, Computation and Optimization in Information Systems and Management Sciences*, vol. 14 of *Communications in Computer and Information Science*. Berlin Heidelberg: Springer, pp. 358–368. ISBN 978-3-540-87476-8. Available at: https://doi.org/10.1007/978-3-540-87477-5_39.
- Whitney, H., 1932. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics* 54 (1), 150–168. ISSN 00029327. Available at: <http://www.jstor.org/stable/2371086>.
- Wiener, H., 1947. Structural determination of paraffin boiling points. *Journal of the American Chemical Society* 69 (1), 17–20. doi:10.1021/ja01193a005. (ISSN 0002-7863).
- Willett, P., 1999. Matching of chemical and biological structures using subgraph and maximal common subgraph isomorphism algorithms. *The IMA Volumes in Mathematics and its Applications* 108, 11–38.
- Wood, D.R., 1997. An algorithm for finding a maximum clique in a graph. *Operations Research Letters* 21 (5), 211–217. Available at: <http://www.sciencedirect.com/science/article/B6V8M-3V6083C-1/2/fc0edc68f2eca1ec3d6ab81ac824778c>.
- Yamaguchi, A., Aoki, K.F., Mamitsuka, H., 2003. Graph complexity of chemical compounds in biological pathways. *Genome Informatics* 14, 376–377.
- Yamaguchi, A., Aoki, K.F., Mamitsuka, H., 2004. Finding the maximum common subgraph of a partial k-tree and a graph with a polynomially bounded number of spanning trees. *Information Processing Letters* 92 (2), 57–63.

Further Reading

- Barnard, J.M., 1993. Substructure searching methods: Old and new. *Journal of Chemical Information and Computer Sciences* 33 (4), 532–538.
- Chen, L., 2003. Substructure and maximal common substructure searching. In: *Bultinck, P., Winter, H. De, Langenaeker, W., Tollenare, J.P. (Eds.), Computational Medicinal Chemistry for Drug Discovery*. CRC Press, pp. 483–514.
- Ehrlich, H.-C., Rarey, M., 2011. Maximum common subgraph isomorphism algorithms and their applications in molecular science: A review. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1 (1), 68–79. Available at: <https://doi.org/10.1002/wcms.5>.
- Raymond, J.W., Gardiner, E.J., Willett, P., 2002b. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *Computer Journal* 45 (6), 631–644.
- Raymond, John W., Willett, Peter, 2002a. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design* 16 (7), 521–533.

Relevant Websites

<http://insilab.org/lisica/>
In Silico Laboratory.

<http://users.abo.fi/mivainio/shaep/index.php>
ShaEP.

Binding Site Comparison – Software and Applications

Christiane Ehart and Tobias Brinkjost, TU Dortmund University, Dortmund, Germany

Oliver Koch, Technical University of Dortmund, Dortmund, Germany

© 2019 Elsevier Inc. All rights reserved.

Introduction

The automated comparison of binding sites can provide useful insights into enzyme function, evolutionary relationships, poly-pharmacology, off-target effects and can assist in drug repurposing. Therefore, it has become a valuable tool especially in medicinal chemistry during the process of drug discovery. Another possible application domain is the functional characterization of novel binding sites by a comparison against already annotated and characterized ligand cavities. Recent scientific publications demonstrate that binding site similarity is by no means as obvious as one might expect, i.e., cavities of functionally and structurally distinct proteins might show a high similarity (Ehart *et al.*, 2016). Usually, the focus is on evolutionary related proteins by comparing protein structures or sequences. This strategy completely dismisses the possibility that unrelated proteins could bind similar ligands. Some very recent publications focus on the impact of local protein similarities on protein function (Garma *et al.*, 2016; Mudgal *et al.*, 2017). They underpin the general observation that a similar fold does not necessarily result in a similar function whereas a common function can be fulfilled by proteins with different folds. Therefore, the comparison of the binding site of interest against a data set of diverse binding sites might unravel some disregarded similarities. Nevertheless, it has to be stated that proteins binding to similar ligands or sharing a common function do not necessarily share a similar small molecule binding site (Barelir *et al.*, 2015).

Encouraged by a variety of useful applications, a multitude of binding site comparison tools were developed. They can be classified by the way the binding site is modeled, the comparison algorithm, or the similarity metric used for binding site similarity ranking. Binding site modeling is of special interest for the choice of the appropriate method according to the particular problem. Therefore, the first part of this article presents software with a special focus on binding site modeling. The second part is dedicated to the model principles and corresponding algorithms for binding site comparison explaining the different approaches in more detail. Finally, four examples for the impact of binding site comparison are discussed. The chosen problems illustrate that tools which are based on rather different binding site modeling, comparison, and similarity scoring concepts led interesting findings in the past. Therefore, the combination of various tools – also those not presented herein – might prove beneficial for a certain application.

A summary of all methods and their characteristics presented herein can be found in Table 1. The number of published tools is considerably large. Hence, only a small subset of software tools, which were successfully applied in various scientific projects and are available as standalone software, are discussed in this article. Additionally, we decided to exclusively introduce tools that allow for a comparison of user-defined datasets.

One aspect that is beyond the scope of this article is the automated identification of binding sites. It represents an important pre-processing step, for example, for the binding site annotation of novel targets. Cavity prediction is possible via elaborate binding site detection algorithms such as energy-based (SiteHound (Gherzi and Sanchez, 2009)), knowledge-based (ConCavity (Capra *et al.*, 2009)), and geometry-based (LIGSITE^{CS} (Huang and Schroeder, 2006)) methods. An overview over binding site identification software and its impact can be found elsewhere (Krone *et al.*, 2016).

In the following, we will first discuss different approaches to model small molecule binding sites with a special focus on one method per modeling approach (SMAP, FLAP, APF, SiteEngine). Second, the different model principles of those tools are explained in more detail. Afterward, we provide one application example per tool to illustrate the broad applicability of the software introduced herein. Finally, some hints are provided for successfully using binding site comparison tools.

Binding Site Modeling

The characterization of the binding site is a crucial part of the comparison process as it determines the final outcome of the study (Henrich *et al.*, 2010). The different binding site representation schemes are illustrated in Fig. 1. While some tools solely rely on residues and a crude abstraction of residue properties, other tools use more or less elaborate surface calculations for a projection of physicochemical properties on the binding site surface patches or solvent accessible residue atoms. Moreover, a continuous representation of pharmacophoric properties in 3D space can be applied, i.e., different pharmacophoric potentials are assigned to all 3D points. The probably highest level of abstraction is a comparison based on a description of the interactions between protein and ligand. The outcome of such methods is often highly dependent on the chemical nature of the small molecule interacting with cavity residues. In the following, one method per modeling scheme is explained in more detail to characterize the different concepts. For the remaining methods, a short summary is given.

Binding Site Modeling Based on Residues

A straightforward approach is the description of a binding site by 3D and physicochemical characteristics of the underlying residues. Nevertheless, a broad variety of modeling approaches is possible.

Table 1 Binding site comparison tools presented and discussed in this article

Software tool	Binding site representation	Model principle	Field of successful application (Ehrt <i>et al.</i> , 2016)	Availability
Cavbase (Schmitt <i>et al.</i> , 2001, 2002)	Residue-based	Graph	Protein–ligand interactions Virtual screening Evolutionary relationships Drug repurposing	CCDC
SMAP (Xie <i>et al.</i> , 2009)		Graph	Polypharmacology	Download
FuzCav (Weill and Rognan, 2010)		Fingerprint	Drug repurposing	Upon request
SiteAlign (Schalon <i>et al.</i> , 2008)		Fingerprint	Protein–ligand interactions	Upon request
PocketMatch (Yeturu and Chandra, 2008)		Distance lists	Protein–ligand interactions	Download
TM-align (Zhang and Skolnick, 2005)		Matrix	Function prediction Polypharmacology Homolog analysis Drug repurposing	Download
IsoMIF (Chartier and Najmanovich, 2015)	Interaction-based	Graph(grid)	Drug repurposing	Download
FLAP (Baroni <i>et al.</i> , 2007)		Fingerprint	Prediction of compound selectivity profiles and affinities	Molecular Discovery
KRIPO (Ritschel <i>et al.</i> , 2014)		Fingerprint	Off-target prediction	Download
TIFF (Desaphy <i>et al.</i> , 2013)		Fingerprint	Virtual screening	Upon request
APF (Totrov, 2011)	Observer point-based	3D-points (grid)	Off-target prediction Homolog analysis	Molsoft
ProBIS (Konc and Janezic, 2010)	Surface-based	Graph	Function prediction	Download
PSIM (Spitzer <i>et al.</i> , 2011)		3D-points (grid)	Off-target prediction	BioPharmics LLC
Shaper (Desaphy <i>et al.</i> , 2012)		3D points (grid)	Protein–ligand interactions	Upon request
SiteEngine (Shulman-Peleg <i>et al.</i> , 2004)		3D points	Protein–protein interactions	Download

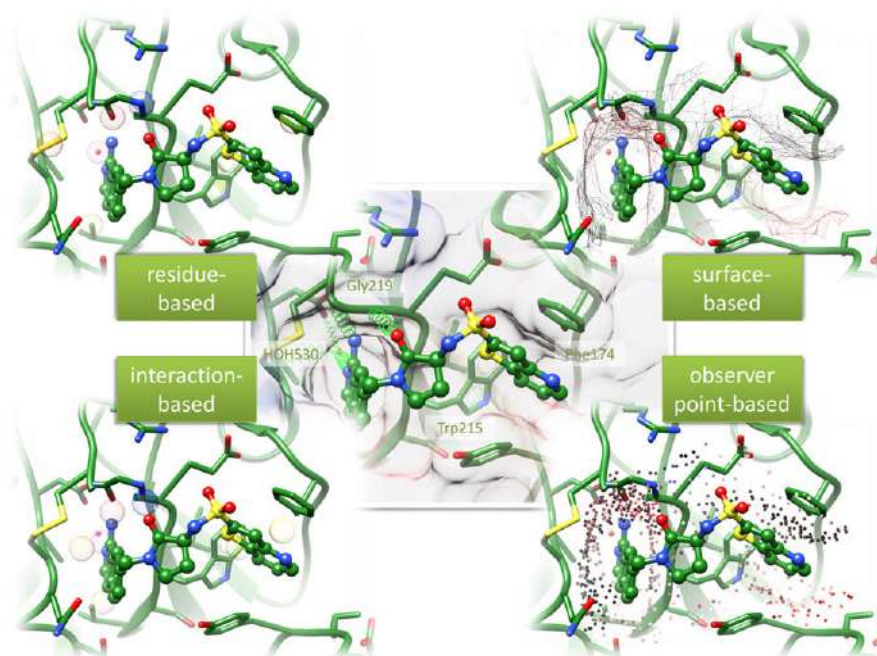


Fig. 1 Four different ways of binding site modeling that are applied by the comparison algorithms presented in this article. The chosen example shows the binding site of coagulation factor Xa (1f0r.A@pdb) in complex with the nanomolar inhibitor RPR208815 in four different representation schemes.

The representation in SMAP (Xie *et al.*, 2009) relies on the residues' C α atoms as well as their characteristics. It is an extension of the previously developed algorithm named Sequence Order Independent Profile-Profile Alignments (SOIPPA) (Xie and Bourne, 2008). Binding sites are represented by C α atoms and a mesh surface based on their Delaunay tessellation. Each C α atom is assigned a normal vector that is perpendicular to the surrounding surface and a weight that accounts for chemical similarity and evolutionary relationship. The McLachlan (1971) chemical similarity matrix is utilized to represent physicochemical similarity. It is based on polar or non-polar character, size, shape, and charge of residues. A position-specific scoring matrix of the 20 amino acids which depends on the residue's sequence position expresses the evolutionary relationship. Taken together, the binding site C α atoms, their spatial orientation, and the weights of their respective residues are modeled by a graph.

The basic idea of Cavbase (Schmitt *et al.*, 2001, 2002) is to characterize the binding site properties by means of pseudocenters representing the physicochemical properties of binding site residue moieties: hydrogen bond acceptor and donor, mixed hydrogen bond acceptor/donor, center of an aromatic ring, centers comprising π electrons, aliphatic groups, and metal ions. In FuzCav (Weill and Rognan, 2010), the C α atom coordinates of binding sites are labeled with six pharmacophoric properties of the respective residue: hydrogen bond acceptor and donor, positive and negative ionizable, aromatic, and aliphatic. A final fingerprint representation encodes all unique pharmacophore triplets and the respective binned C α atom distances. PocketMatch (Yeturu and Chandra, 2008) classifies binding site residues according to their physicochemical properties: aliphatic, positively and negatively polarized, aromatic, and polar. They are represented by three types of points (C α , C β , the geometric center of the side chain atoms). SiteAlign (Schalon *et al.*, 2008) includes topological properties (e.g., side chain orientation, size) as well as physicochemical properties (e.g., hydrogen bond acceptor/donor, charge, aromaticity). These characteristics are projected on a discretized 80 triangle sphere by deriving a geometrical vector from the C α atom of each binding site residue to the sphere center. The TM-align (Zhang and Skolnick, 2005) algorithm was originally developed for protein structure alignment. Nonetheless, it was also successfully applied to compare protein binding sites. The comparison is based on the C α coordinates and the underlying secondary structures elements based on the DSSP assignment (Kabsch and Sander, 1983). However, TM-align is the only approach presented herein which is sequence order dependent, i.e., a discontinuous alignment of binding site residues is impossible.

Binding Site Modeling Based on Interactions

These methods represent the binding site by modeling known ligand interaction patterns to find distinct relationships between pockets. However, some of them rely on the presence of structurally characterized protein-ligand complexes and are not suited to compare novel predicted binding sites against already known ones.

Fingerprints for Ligands and Proteins (FLAP) (Baroni *et al.*, 2007) utilizes GRID (Goodford, 1985) molecular interaction fields (MIFs) to analyze the protein cavity. Six probes (probes for hydrogen bond interactions, salt bridges, hydrophobic interactions, shape etc.) can be chosen to define energetically favorable and unfavorable interactions for the binding site of interest. This

information is used to determine target-based pharmacophoric points using a weighted energy-based function. All possible energetically favorable arrangements of four pharmacophoric points are generated. A subsequent selection of groups of most important points can be achieved manually or automatically. Finally, a fingerprint is derived based on the distances, probe types, and chirality as four probes assume relative chiral positions in 3D space.

MIFs are also the basis for the method IsoMIF (Chartier and Najmanovich, 2015), but the GetCleft (Gaudreault *et al.*, 2015) algorithm is used to label grid points with the interaction energy for different probes. The final MIF is the set of interaction vectors at all vertex positions in the grid. Key Representation of Interaction in Pockets (KRIPO) (Ritschel *et al.*, 2014) uses intermolecular interaction features and generates 3-point pharmacophore fingerprints for ligands as well as for ligand fragments. Each triplet is encoded by the interaction properties and binned distances. Pharmacophoric properties including cation-aromatic interactions and metal coordination are also used for Interaction Fingerprint Triplets (TIFF) (Desaphy *et al.*, 2013). The interaction patterns can be represented by the protein atom coordinates, the ligand atom coordinates, or those of the geometric center of two interacting atoms.

Binding Site Modeling Based on Continuous Binding Site Properties

The method Atomic Property Fields (APF) (Totrov, 2011) treads another and unique path for modeling. It relies on a continuous binding site representation using seven atom properties: hydrogen bond acceptor and donor, lipophilicity, size, electronegativity, charge, aromaticity/hybridization. They are projected on a grid by the means of Gaussian functions. The binding site comprises all receptor atoms within a 6 Å radius of the bound ligand. The method is derived from an approach to describe the similarity of 3D property distributions of two ligands, which led to an optimized measure of chemical similarity (Totrov, 2008).

Binding Site Modeling Based on Surface Properties

The assignment of physicochemical properties in SiteEngine (Shulman-Peleg *et al.*, 2004) is similar to that of the Cavbase (Schmitt *et al.*, 2001, 2002) approach. The atoms of each binding site residue in a 4 Å radius of the binding partner are grouped according to their potential ligand interaction. A Connolly surface of the binding site is constructed and only pseudocenters with at least one surface exposed atom are retained. Pseudocenter triplets are generated and encoded with the respective side lengths and a physicochemical index.

Protein Binding Site (ProBiS) (Konc and Janezic, 2010) defines pseudocenters as realized in the Cavbase approach (Schmitt *et al.*, 2001, 2002). Solvent accessible surface atoms of a ligand binding site or a protein structure of interest are identified and labeled. Structurally related cavities can be identified based on binding sites or protein structures. In Protein SIMilarity (PSIM) (Spitzer *et al.*, 2011) the cavity is represented by surface moieties that can recognize a ligand. Distances to the molecular surface from so-called *observer points* are compared by placing them on a uniform grid. An additional weight is assigned in dependence of the minimum distance to the surface resulting in a defined number of *observer points*. Gaussian functions of the distance deviations from each *observer point* are calculated (minimum distance to any surface point to represent the shape, to a positive polarized atomic surface, and to a negative polarized atomic surface point). The binding site representation for Shaper comparisons is generated by VolSite (Desaphy *et al.*, 2012). A cube is placed at the center of mass of the bound ligand and filled with a grid. *IN* and *OUT* properties are assigned to the grid cells depending on the presence or absence of protein atoms and their degree of buriedness. The resulting site points are assigned pharmacophoric properties and a druggability prediction is performed. Drug-gable annotated binding sites can subsequently be compared.

Model Principles, Their Definitions, and Criteria for Similarity

The choice of an appropriate model principle is crucial for every application, so it is for the comparison of protein binding sites. The model principle defines and limits the level of abstraction, which is sometimes referred to as accuracy. It also defines the complexity of answering certain questions such as: what is the highest similarity of two given models? The complexity has a strong influence on the runtime required to answer such questions. There are three frequently used model principles: graphs, fingerprints, and points in 3D space. Fingerprints and graphs are classical data structures whereas points in 3D space are usually regarded as input data and the way they are stored and managed is referred to as their data structure (e.g., k-d trees (Bentley, 1975)). In contrast to these model principles, hashing is a function to map the input onto the output, without being a data structure such as graphs or fingerprints, and without an implicit similarity such as the root mean square deviation (RMSD) of points in 3D space.

Graphs

A graph $G=(V, E)$ consists of a set of vertices V and a set of edges E connecting these vertices. Metaphorically speaking, graphs contain objects and their relations to each other. In many of the available methods, both (vertices and edges) are labeled with additional information such as pharmacophore features at vertices and distances at edges for instance. The definition of the similarity of two given graphs is frequently based on the maximum common (induced) subgraph (MCS). A graph $G'=(V', E')$ is a

subgraph of $G=(V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. It is said to be (vertex-)induced by V' if $E'=(V' \times V') \cap E$ holds. Let $G=(V, E)$, $G_1=(V_1, E_1)$, and $G_2=(V_2, E_2)$ be graphs. G is subgraph-isomorphic to a graph G_1 if there exists an injection $(\varphi): V \rightarrow V_1$ in such a way that $\forall u, v \in V: (u, v) \in E \Rightarrow ((\varphi)(u), (\varphi)(v)) \in E_1$. G is said to be a common subgraph of G_1 and G_2 if G is subgraph-isomorphic to G_1 and G_2 . The MCS G_{max} of G_1 and G_2 is a common subgraph of G_1 and G_2 for which $|G'| \leq |G_{max}|$ holds for all common subgraphs G' of G_1 and G_2 . Note that the MCS is not necessarily unique.

The determination of the MCS of two graphs can be reduced to the problem of finding the largest clique in an appropriately defined modular product graph, also known as compatibility, correspondence, or association graph and was first described by [Levi \(1973\)](#). A clique C in an undirected graph $G=(V, E)$ is a subset of vertices $C \subseteq V$ such that every two distinct vertices are adjacent. In a modular product graph $G_P=(V_P, E_P)$ of two labeled graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ the set of vertices $V_P \subseteq V_1 \times V_2$ contains a vertex (v_1, v_2) with $v_1 \in V_1$ and $v_2 \in V_2$ if their labels are compatible. The set of edges E_P contains an edge connecting the vertices $(u_1, u_2), (v_1, v_2) \in V_P$ if either $e_1=(u_1, v_1) \in E_1$ and $e_2=(u_2, v_2) \in E_2$ and if their labels are compatible, or if $e_1 \notin E_1$ and $e_2 \notin E_2$. There is no general definition of the compatibility of labels.

The maximal clique detection algorithm by [Bron and Kerbosch \(1973\)](#) is widely used (e.g., by Cavbase, IsoMIF, ProBiS). It is a recursive backtracking algorithm that uses sets of vertices R (temporary result), P (possible extensions), X (excluded set) and finds the maximal cliques that include all of the vertices in R , some of the vertices in P , and none of the vertices in X . However, a graph $G=(V, E)$ can contain up to $3^{V/3}$ cliques ([Moon and Moser, 1965](#)) which makes their determination a demanding objective. Roughly speaking, a linear increase in the size of the input graphs may result in an exponential increase of cliques and therefore of the time required for their determination (runtime). There are optimizations available ([Tomita et al., 2006](#)) to reduce the branches of the recursion tree, the recursive calls, respectively.

SMAP represents ligand binding sites as graphs modeling C α atoms as vertices and the connections of the tessellation (mesh) as edges. The similarity of two graphs is based on determining the maximum weighted common subgraph (MWCS) which also uses an algorithm for the maximum-weight clique problem ([Kumlander, 2004](#)). This algorithm is based on the assumption that two vertices of an independent set cannot be in the same clique. An independent set is a set of vertices that are pairwise not connected by an edge (non-adjacent). The difference between a MCS and a MWCS is that the maximality is not based on the size of the subgraph but on the sum of weights reflecting the compatibility of each pair of matched vertices. Each such pair does not match in a binary manner (yes/no) but to a certain weight of compatibility. This weight is based on Gaussian density functions taking residue substitution matrices and geometrical properties into account.

Fingerprints

Fingerprints, such as molecular fingerprints ([Todeschini et al., 2009](#)) or extended-connectivity fingerprints ([Rogers and Hahn, 2010](#)), are widely used in bioinformatics, chemoinformatics, and computational biology. In general, fingerprints are vectorial descriptors consisting of binary or integer values. In the binary case, each value is usually correlated to the absence (0) or occurrence (1) of a certain property in the input. In the integer case, each value represents a counter for the number of occurrences of a certain property such as pharmacophore features or distances. The latter are usually binned to discrete value ranges. The major advantage of fingerprints is the ability to determine the similarity of two given fingerprints fast. Given two fingerprints $F_1=(v_1, v_2, \dots, v_n)$ and $F_2=(w_1, w_2, \dots, w_m)$ with $|F_1|=n$ and $|F_2|=m$ the similarity is defined by the number of compatible values at mapping indices. If both fingerprints are of equal length $n=m$, there is a direct mapping of v_1 on w_1 , v_2 on w_2 and so on. Otherwise, the fingerprints have to be aligned to obtain an indices mapping.

The definition of compatibility is trivial for binary values, but in the case of integer values, a slight deviation of values at mapping indices may still be defined as a match or compatible, respectively. A final score can be obtained using a similarity or distance metric such as the Tanimoto coefficient, which, among others, has proven to be a meaningful metric ([Bajusz et al., 2015](#)). It is based on the number of compatible values C , numbers of non-zero values Z_0 in F_1 and Z_1 in F_2 , and defined by: $Tanimoto(F_1, F_2)=C/(Z_0+Z_1-C)$. Thus, determining the similarity of two fingerprints is based on counting zero and non-zero values, which can be done by scanning both fingerprints simultaneously and once only. Thus, a linear increase in input size only results in a linear increase in runtime.

FLAP uses fingerprints in a different way. Here, a fingerprint representing a protein's binding site is an array of varying length containing several 11 integer value fingerprints. These 11 fields contain the information of one favorable arrangement of four pharmacophoric points, all pairwise distances of these points, and a value proportional to the sum of their energy. During the comparison of two protein binding sites P_1 and P_2 a fingerprint in P_1 similar to a fingerprint in P_2 is searched for. This is done by sorting the fingerprints with respect to their distances which supports the use of sophisticated search algorithms. Once a similar pair of fingerprints is found, the matching of the four pharmacophoric points of each fingerprint is used to determine the orientation of the corresponding binding sites. If these also superimpose within a certain threshold, the two protein binding sites are finally reported as similar.

Points in 3D

The comparison of clouds of points in 3D space $C_1=(p_1, p_2, \dots, p_n)$ and $C_2=(q_1, q_2, \dots, q_n)$ of the same size is in general an optimization problem of finding an alignment which has the lowest RMSD value: $RMSD(C_1, C_2)=\sqrt{\left(\frac{1}{n} \sum_{i=1}^n d(p_i, q_i)^2\right)}$ with

distance function d . During the optimization routine one cloud is usually fixed in space while the other one undergoes several translations and rotations to obtain an alignment with a minimum RMSD value. If the clouds are of different size, one has to define a mapping of the points of the smaller cloud onto the ones of the larger cloud.

APF utilizes a Monte Carlo algorithm to obtain a mapping of the two binding sites to be compared. A Monte Carlo algorithm is a randomized algorithm whose output may be incorrect with a certain probability. In contrast, a deterministic algorithm is always expected to give correct answers. APF uses a pseudo-Brownian Monte Carlo algorithm to minimize local gradients. In each iteration, the algorithm calculates a random rotation of the molecule with respect to chemical meaningfulness, for example, avoid atom clashes, to find a favorable and energetically minimized orientation. Although the selection of a randomized rotation is fast, several rotations are obtained and evaluated to find a satisfactory solution which is not necessarily the optimal one.

Hashing

Hashing has a broad field of applications among which are data storage, searching, and cryptography for instance. Hashing or more precisely a hash function f maps elements of a universe U to a set of keys K : $f: U \rightarrow K$. The basic idea is that $|U|$ is usually much larger than $|K|$ or even unlimited whereas K is of fixed size. Because of this, collisions can appear: $f(u) = f(v)$ with $u, v \in U$ and $u \neq v$. Therefore, the choice of a specific hash function for the universe of the individual application is crucial and should minimize the chance of collisions. A very simple hash function is the modulo function: $f(u) := u \bmod |K|$. However, there are many different hash functions such as double hashing using two hash functions, middle-square hashing based on [von Neumann \(1951\)](#), and Cuckoo hashing ([Pagh and Rodler, 2004](#)). A combination of a hash function and a fingerprint-like bit vector is the Bloom filter ([Bloom, 1970](#)). It is used to test whether an element $u \in U$ is a member of a set or not. If u is added to the set the bit at the index of the fingerprint obtained from the hash function applied on u is set to 1. In a test the algorithm also uses the function on u and checks whether the corresponding bit is set to 1 and if true, the element may be in. Otherwise, it is definitely not part of the set. Usually, a Bloom filter uses multiple hash functions.

Hashing is a comparably fast technique, but the runtime strongly relies on the complexity of the function itself, in particular the underlying input (e.g., graphs, fingerprints, integers, etc.).

The hash function of SiteEngine ([Shulman-Peleg et al., 2004](#)) is based on 4-tuples consisting of the side length of triangles defined by a triplet of pseudocenters and a physicochemical index encoding the properties of the pseudocenters. It is a geometric hash function and finds almost congruent matching triangles of which each pair defines a candidate transformation. The resulting matchings are finally re-evaluated by an elaborate filtering and scoring workflow to obtain the final superposition.

Examples for Successfully Applied Binding Site Comparison Tools

The following examples will illustrate that binding site similarities are by no means easily detectable via sequence or protein structure comparison. An abstraction of the various available binding site properties is often required. This abstraction and spatial comparison led to some astonishing results in the past. Examples for unrelated proteins with respect to sequence and overall fold that share similar ligands and highly similar binding sites underline the necessity of binding site comparison. Additionally, examples for similar proteins with rather dissimilar binding sites are given that underline the importance of binding site comparison to elucidate probable off-targets and ways to enhance selectivity. The examples were chosen to illustrate that various approaches that make use of different types of binding site modeling, comparison, and similarity scoring can shed light on highly interesting relationships between otherwise unrelated proteins.

Polypharmacology

The first example illustrates the impact of binding site comparison on polypharmacology ([Xie et al., 2011](#)). The binding site comparison tool SMAP ([Xie et al., 2009](#)) was used to compare the nelfinavir binding site of the HIV protease dimer (1ohr@pdb) against 5985 human protein structures or their homologues from other organisms. The 126 most similar superimposed protein binding sites were used for molecular docking of nelfinavir. This led to 92 putative off-targets. Most of the hits were protein kinases belonging to tyrosine, cAMP-dependent, cGMP-dependent kinase families, or protein kinase C family. Exhaustive MD (molecular dynamics) simulation studies and Molecular Mechanics/Generalized Born Surface Area (MM/GBSA) free energy calculations led to the conclusion that nelfinavir interacts with EGFR, might also interact with IGF-1R, FAK, Akt2, CDK2, ARK, PKD1, and will probably not bind to FGFR, EphB4 and Abl. In a subsequent biochemical screening, the authors provided evidence for a nelfinavir-induced inhibition of EGFR and ErbB2. **Fig. 2** shows a superposition of the protease (1ohr.A@pdb) and the EGFR pocket (2j6m.A@pdb) to illustrate the similarities.

Selectivity Profiling

The second example provides a glimpse into the wide variety of possible application domains for binding site comparison. The method FLAP was applied to correlate inhibition profiles for kinase inhibitors with identified binding site features for various kinase binding sites ([Sciabola et al., 2010](#)). Based on a binding site feature similarity matrix of fourteen protein kinases partial least-squares models were created to predict pIC₅₀ values. The authors of this study were able to choose a combination of probes

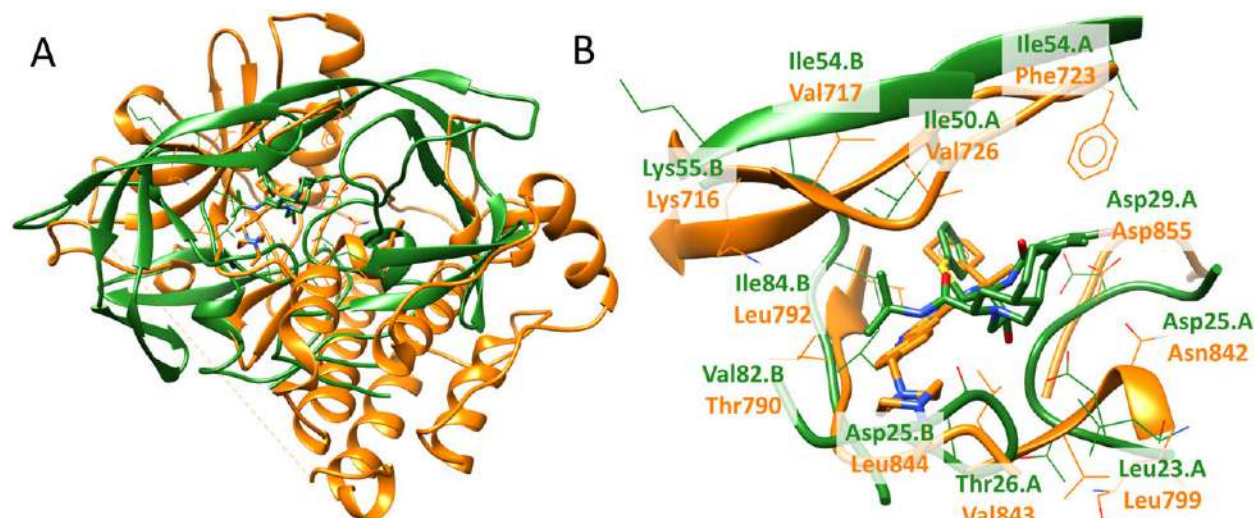


Fig. 2 Binding site similarities between the HIV protease (1ohr@pdb, green) and EGFR (2j6m.A@pdb, orange) based on a SMAP (Xie *et al.*, 2009) comparison. (A) Complete structures superimposed based on the binding site similarities. (B) A detailed view of the binding site similarities. Nelfinavir bound to HIV protease is shown in green sticks while the EGFR ligand with the PDB-ID AEE is represented by orange sticks.

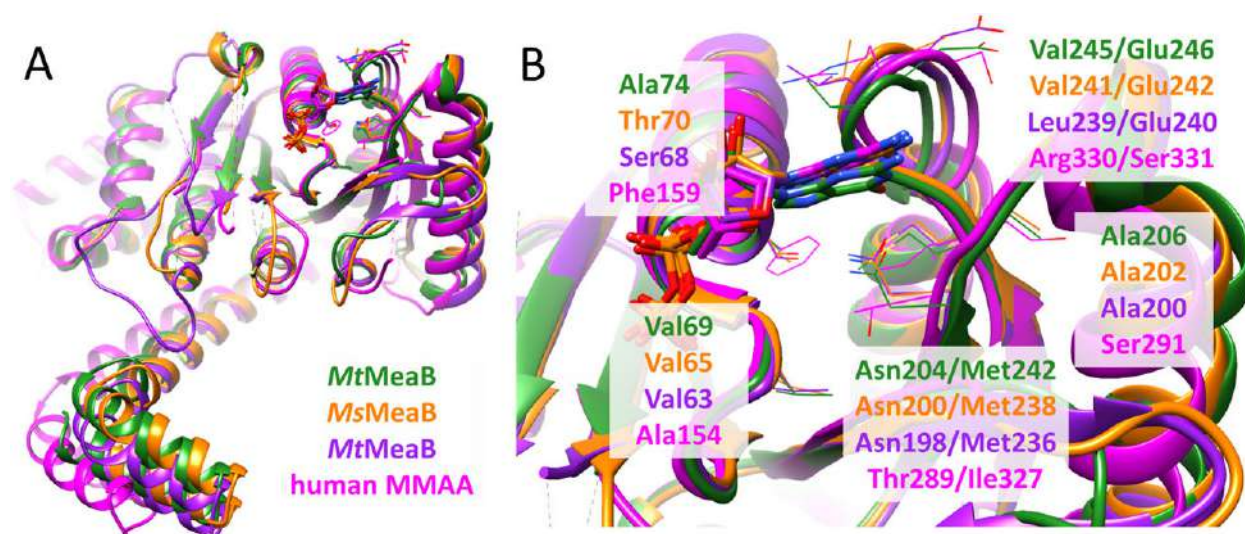


Fig. 3 Comparison between homologous proteins from different Mycobacteria species and a human related enzyme. The MeaB structures from *M. tuberculosis* (3md0@pdb), *M. smegmatis* (3nxs@pdb), and *M. thermoresistibile* (3tk1@pdb) are represented in green, orange, and purple, respectively. The structure of the human homolog MMAA is shown in pink. Reprinted and modified with permission from (Ehrt *et al.*, 2016). Copyright 2016 American Chemical Society. (A) The superposition of all structures illustrates their close relationship. (B) A detailed comparison of the underlying binding sites provided insights into possible starting points to increase the selectivity of inhibitors towards the mycobacterial enzymes.

to generate MIFs that enabled them to predict the level of inhibition of a compound on different protein kinases. Consequently, this study underlines the possible impact of binding site comparison for selectivity prediction for protein kinase inhibitors.

Off-Target Analysis

The method APF was applied to analyze differences between the binding sites of homologous structures (Edwards *et al.*, 2015). The methylmalonyl CoA mutase-associated GTPase MeaB which is an essential enzyme for the growth of pathogenic bacteria (e.g., *Mycobacterium tuberculosis*) represents an interesting drug target. The nucleotide binding sites of crystal structures from different bacteria and a structure of the human homologous protein methylmalonic aciduria associated protein A (MMAA, 2www@pdb) were compared to analyze differences between the structures. They can be exploited to circumvent a possible inhibition of the human enzyme which might result in fatal methylmalonic aciduria. Fig. 3 shows the result of a binding site comparison between

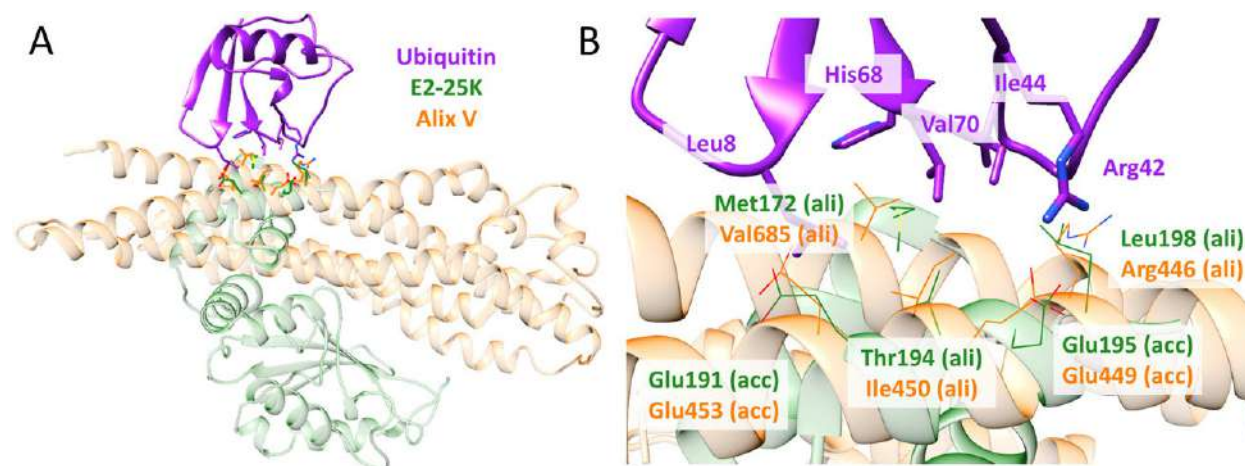


Fig. 4 Binding site similarities between the Ubiquitin-conjugating enzyme E2 K (3k9p.A@pdb) and a surface patch of the ALIX-V domain (2ojq@pdb). The template structures of the Ubiquitin-conjugation enzyme E2 K is shown in green with the bound Ubiquitin molecule in purple. The ALIX-V domain is depicted in orange. (A) Superposition of both protein structures with respect to the identified local similarity. (B) Zoomed view to illustrate identified binding site similarities with respect to the ubiquitin binding site of the Ubiquitin-conjugating enzyme. Some of the crucial similarities between both binding sites are depicted and the involved residues are labeled accordingly.

three mycobacterial MeaB structures and the human MMAA. A proper visualization of the major dissimilarities can assist the development of safe MeaB inhibitors.

Protein–Protein Interactions

The last example illustrates the applicability of binding site comparison for the analysis of protein–protein interactions. SiteEngine (Shulman-Peleg *et al.*, 2004) which is based on similarities between surface patches of binding sites was utilized to analyze potential protein–protein interactions (Keren-Kaplan *et al.*, 2013). A template ubiquitin-binding domain represented the starting point to search for new potential ubiquitin-binding domains within the PDB. Ubiquitylation plays a crucial role in trafficking, for example, of retroviral transmembrane proteins. The ALIX-V domain (2ojq@pdb) was identified as new potential ubiquitin binding domain due to its high local similarity to the ubiquitin binding site. Subsequent protein–protein docking and experimental binding affinity measurements confirmed this result. The finding was also verified in an independent study (Dowlatshahi *et al.*, 2012). The match between the template structure (3k9p@pdb) and ALIX-V is depicted in Fig. 4.

Remarks Concerning Binding Site Comparison

Finally, we want to provide some useful hints which might assist in applying binding site comparison tools and software for various projects.

First, proteins are no rigid constructs as commonly observed in protein crystal structures. NMR structures and MD simulations can provide useful information with respect to flexible protein regions and binding site flexibility. The knowledge of binding site dynamics can have a huge impact on computational strategies (Stank *et al.*, 2016). As shown in previous studies, it can be beneficial to use an ensemble of binding site structures to identify or compare ligand binding sites (Lanig *et al.*, 2015; Miguel *et al.*, 2015; Möller-Acuña *et al.*, 2015; Zhao *et al.*, 2012). This is especially important as the results of some comparison algorithms are very sensitive toward binding site definition. Given a large amount of crystal structures of the protein bound to different ligands, all available binding sites should be considered.

Moreover, although various methods provide some measure of significance to analyze the impact of the identified similarities, the obtained results should be visualized and rationalized. Sometimes the use of two or three tools is advisable. Different binding site comparison methods were evaluated by the means of highly distinct datasets and designed to address problems of varying application domains. Therefore, the used benchmark datasets can provide insights to appropriate tools for the aim of the study. Tools validated with respect to classification performance are promising tools to infer evolutionary relationships and protein function. Other tools were analyzed using datasets of proteins binding similar ligands. Those might be an appropriate choice for applications such as off-target prediction, polypharmacology, and drug repurposing. Nevertheless, there are examples for various tools that underline their impact on fields they were not developed for.

A further challenge is the ranking of binding site similarities. Many tools offer different similarity measures and it is often beneficial to use a scoring scheme that is most appropriate for the aim of the study. A method which models and compares a binding site based on residues might nonetheless offer scores that consider the shape overlap of two cavities or common interaction patterns. This issue was not addressed in this article and the reader is referred to the publications cited herein.

Finally, a database search cannot only be performed based on known binding sites. In cases of novel targets, it is highly recommended to predict binding sites based on geometrical, evolutionary information, or probe interaction potentials and compare them against large databases of annotated binding sites (e.g., sc-PDB (Kellenberger *et al.*, 2006)) to gain insights into potential druggable binding sites.

Acknowledgement

C.E. is funded by the Kekulé Mobility Fellowship of the Chemical Industry Fund (FCI). O.K. is funded by the German Federal Ministry for Education and Research (BMBF, Medizinische Chemie in Dortmund, TU Dortmund University, Grant BMBF 1316053).

Appendix

Connolly surface: solvent-accessible molecular surface as calculated by applying a virtual solvent molecule as probe sphere; docking: computational prediction of small molecule binding modes within predefined protein cavities; drug repurposing: investigation of new applications for existing drugs with known properties outside their original medicinal scope; Gaussian function: a continuous bell-shaped function which represents probability distributions; MD simulation: molecular dynamics simulation, a method to calculate the time-dependent behavior of molecular systems; Monte Carlo algorithm: a randomized algorithm that is, with a certain and usually small probability, allowed to provide none or wrong output; substitution matrix: a scoring matrix which contains a value for all possible substitutions (e.g., of amino acid residues) to calculate the degree of similarity between two objects; tessellation: a gapless and non-overlapping covering of shapes by repeating smaller shapes (often triangles) to represent, for example, molecular surfaces; virtual screening: *in silico* search for molecules that might modulate the activity of a protein in a desired manner.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Chemical Similarity and Substructure Searches. Chemoinformatics: From Chemical Art to Chemistry in Silico. Computational Tools for Structural Analysis of Proteins. Fingerprints and Pharmacophores. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Prediction of Protein-Binding Sites in DNA Sequences. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Structural Bioinformatics: An Overview. Small Molecule Drug Design

References

- Bajusz, D., Rácz, A., Héberger, K., 2015. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics* 7, 20.
- Barelrier, S., Sterling, T., O'Meara, M.J., Shoichet, B.K., 2015. The recognition of identical ligands by unrelated proteins. *ACS Chemical Biology* 10 (12), 2772–2784.
- Baroni, M., Cruciani, G., Sciabola, S., Perruccio, F., Mason, J.S., 2007. A common reference framework for analyzing/comparing proteins and ligands. Fingerprints for Ligands and Proteins (FLAP): Theory and application. *Journal of Chemical Information and Modeling* 47 (2), 279–294.
- Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18 (9), 509–517.
- Bloom, B.H., 1970. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM* 13 (7), 422–426.
- Bron, C., Kerbosch, J., 1973. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM* 16 (9), 575–577.
- Capra, J.A., Laskowski, R.A., Thornton, J.M., Singh, M., Funkhouser, T.A., 2009. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLOS Computational Biology* 5 (12), e1000585.
- Chartier, M., Najmanovich, R., 2015. Detection of binding site molecular interaction field similarities. *Journal of Chemical Information and Modeling* 55 (8), 1600–1615.
- Desaphy, J., Azdimousa, K., Kellenberger, E., Rognan, D., 2012. Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. *Journal of Chemical Information and Modeling* 52 (8), 2287–2299.
- Desaphy, J., Raimbaud, E., Ducrot, P., Rognan, D., 2013. Encoding protein-ligand interaction patterns in fingerprints and graphs. *Journal of Chemical Information and Modeling* 53 (3), 623–637.
- Dowlatshahi, D.P., Sandrin, V., Vivona, S., *et al.*, 2012. ALIX is a Lys63-specific polyubiquitin binding protein that functions in retrovirus budding. *Developmental Cell* 23 (6), 1247–1254. doi: 10.1016/j.devcel.2012.10.023.
- Edwards, T.E., Baugh, L., Bullen, J., *et al.*, 2015. Crystal structures of Mycobacterial MeaB and MAA-like GTPases. *Journal of Structural and Functional Genomics* 16 (2), 91–99.
- Ehrt, C., Brinkjost, T., Koch, O., 2016. Impact of binding site comparisons on medicinal chemistry and rational molecular design. *Journal of Medicinal Chemistry* 59 (9), 4121–4151.
- Garma, L.D., Medina, M., Juffer, A.H., 2016. Structure-based classification of FAD binding sites: A comparative study of structural alignment tools. *Proteins* 84 (11), 1728–1747.
- Gaudreault, F., Morency, L.-P., Najmanovich, R.J., 2015. NRGsuite: A PyMOL plugin to perform docking simulations in real time using FlexAID. *Bioinformatics* 31 (23), 3856–3858.
- Gherzi, D., Sanchez, R., 2009. EasyMIFS and SiteHound: A toolkit for the identification of ligand-binding sites in protein structures. *Bioinformatics* 25 (23), 3185–3186.
- Goodford, P.J., 1985. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry* 28 (7), 849–857.
- Henrich, S., Salo-Ahen, O.M.H., Huang, B., *et al.*, 2010. Computational approaches to identifying and characterizing protein binding sites for ligand design. *Journal of Molecular Recognition* 23 (2), 209–219.
- Huang, B., Schroeder, M., 2006. LIGSITE_{cs}: Predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Structural Biology* 6, 19.

- Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22 (12), 2577–2637.
- Kellenberger, E., Muller, P., Schalon, C., *et al.*, 2006. sc-PDB: An annotated database of druggable binding sites from the Protein Data Bank. *Journal of Chemical Information and Modeling* 46 (2), 717–727.
- Keren-Kaplan, T., Attali, I., Estrin, M., *et al.*, 2013. Structure-based in silico identification of ubiquitin-binding domains provides insights into the ALIX-V: Ubiquitin complex and retrovirus budding. *The EMBO Journal* 32 (4), 538–551.
- Konc, J., Janežič, D., 2010. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics* 26 (9), 1160–1168.
- Krone, M., Kozlíková, B., Lindow, N., *et al.*, 2016. Visual Analysis of Biomolecular Cavities: State of the Art. *Computer Graphics Forum* 35, 527–551.
- Kumlander, D., 2004. A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-colouring and a backtrack search. In: *Proceedings of the Fourth International Conference on Engineering Computational Technology*, Lisbon, Portugal. Stirlingshire: Civil-Comp Press.
- Lanig, H., Reisen, F., Whitley, D., *et al.*, 2015. In silico adoption of an orphan nuclear receptor NR4A1. *PLOS ONE* 10 (8), e0135246.
- Levi, G., 1973. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo* 9 (4), 341–352.
- McLachlan, A.D., 1971. Tests for comparing related amino-acid sequences. Cytochrome c and cytochrome c 551. *Journal of Molecular Biology* 61 (2), 409–424.
- Miguel, A., Hsin, J., Liu, T., *et al.*, 2015. Variations in the binding pocket of an inhibitor of the bacterial division protein FtsZ across genotypes and species. *PLOS Computational Biology* 11 (3), e1004117.
- Möller-Acuña, P., Contreras-Riquelme, J.S., Rojas-Fuentes, C., *et al.*, 2015. Similarities between the binding sites of SB-206553 at serotonin type 2 and alpha7 acetylcholine nicotinic receptors: Rationale for its polypharmacological profile. *PLOS ONE* 10 (8), e0134444.
- Moon, J.W., Moser, L., 1965. On cliques in graphs. *Israel Journal of Mathematics* 3 (1), 23–28.
- Mudgal, R., Srinivasan, N., Chandra, N., 2017. Resolving protein structure-function-binding site relationships from a binding site similarity network perspective. *Proteins*.
- Pagh, R., Rodler, F.F., 2004. Cuckoo hashing. *Journal of Algorithms* 51 (2), 122–144.
- Ritschel, T., Schirris, T.J., Russel, F.G., 2014. KRIPO – A structure-based pharmacophores approach explains polypharmacological effects. *Journal of Cheminformatics* 6 (Suppl. 1), 026.
- Rogers, D., Hahn, M., 2010. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* 50 (5), 742–754.
- Schalon, C., Surgand, J.-S., Kellenberger, E., Rognan, D., 2008. A simple and fuzzy method to align and compare druggable ligand-binding sites. *Proteins* 71 (4), 1755–1778.
- Schmitt, S., Hendlich, M., Klebe, G., 2001. From structure to function: A new approach to detect functional similarity among proteins independent from sequence and fold homology. *Angewandte Chemie International Edition* 40 (17), 3141–3144.
- Schmitt, S., Kuhn, D., Klebe, G., 2002. A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology* 323 (2), 387–406.
- Sciabola, S., Stanton, R.V., Mills, J.E., *et al.*, 2010. High-throughput virtual screening of proteins using GRID molecular interaction fields. *Journal of Chemical Information and Modeling* 50 (1), 155–169.
- Shulman-Peleg, A., Nussinov, R., Wolfson, H.J., 2004. Recognition of functional sites in protein structures. *Journal of Molecular Biology* 339 (3), 607–633.
- Spitzer, R., Cleves, A.E., Jain, A.N., 2011. Surface-based protein binding pocket similarity. *Proteins* 79 (9), 2746–2763.
- Stank, A., Kokh, D.B., Fuller, J.C., Wade, R.C., 2016. Protein binding pocket dynamics. *Accounts of Chemical Research* 49 (5), 809–815.
- Todeschini, R., Consonni, V., Mannhold, R., Kubinyi, H., Folkers, G., 2009. *Molecular Descriptors for Chemoinformatics*. Volume I: Alphabetical Listing, Volume II: Appendices. Hoboken: Wiley-VCH.
- Tomita, E., Tanaka, A., Takahashi, H., 2006. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science* 363 (1), 28–42.
- Totrov, M., 2008. Atomic property fields: Generalized 3D pharmacophoric potential for automated ligand superposition, pharmacophore elucidation and 3D QSAR. *Chemical Biology & Drug Design* 71 (1), 15–27.
- Totrov, M., 2011. Ligand binding site superposition and comparison based on atomic property fields: Identification of distant homologues, convergent evolution and PDB-wide clustering of binding sites. *BMC Bioinformatics* 12 (Suppl. 1), S35.
- von Neumann, John, 1951. Various techniques used in connection with random digits. In: Householder, A.S., Forsythe, G.E., Germond, H.H. (Eds.), *Monte Carlo Method* 12. Washington, DC: US Government Printing Office, National Bureau of Standards Applied Mathematics Series, pp. 36–38.
- Weill, N., Rognan, D., 2010. Alignment-free ultra-high-throughput comparison of druggable protein-ligand binding sites. *Journal of Chemical Information and Modeling* 50 (1), 123–135.
- Xie, L., Bourne, P.E., 2008. Detecting evolutionary relationships across existing fold space, using sequence order-independent profile-profile alignments. *Proceedings of the National Academy of Sciences of the United States of America* 105 (14), 5441–5446.
- Xie, L., Evangelidis, T., Xie, L., Bourne, P.E., 2011. Drug discovery using chemical systems biology: Weak inhibition of multiple kinases may contribute to the anti-cancer effect of nelfinavir. *PLOS Computational Biology* 7 (4), e1002037.
- Xie, L., Xie, L., Bourne, P.E., 2009. A unified statistical model to support local sequence order independent similarity searching for ligand-binding sites and its application to genome-based drug discovery. *Bioinformatics* 25 (12), i305–i312.
- Yeturu, K., Chandra, N., 2008. PocketMatch: A new algorithm to compare binding sites in protein structures. *BMC Bioinformatics* 9, 543.
- Zhang, Y., Skolnick, J., 2005. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research* 33 (7), 2302–2309.
- Zhao, Y., Wang, J., Wang, Y., Huang, J., 2012. A comparative analysis of protein targets of withdrawn cardiovascular drugs in human and mouse. *Journal of Clinical Bioinformatics* 2 (1), 10.

Further Reading

- Broomhead, N.K., Soliman, M.E., 2017. Can we rely on computational predictions to correctly identify ligand binding sites on novel protein drug targets? Assessment of binding site prediction methods and a protocol for validation of predicted binding sites. *Cell Biochemistry and Biophysics* 75 (1), 15–23.
- Brown, N., 2011. Algorithms for chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1 (5), 716–726.
- Desaphy, J., Bret, G., Rognan, D., Kellenberger, E., 2015. sc-PDB: A 3D-database of ligandable binding sites – 10 years on. *Nucleic Acids Research* 43 (Database issue), D399–D404.
- Ehrlich, H.-C., Rarey, M., 2011. Maximum common subgraph isomorphism algorithms and their applications in molecular science: A review. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1 (1), 68–79.
- Inhester, T., Rarey, M., 2014. Protein-ligand interaction databases: Advanced tools to mine activity data and interactions on a structural level. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4 (6), 562–575.
- Jalencas, X., Mestres, J., 2013. Identification of similar binding sites to detect distant polypharmacology. *Molecular Informatics* 32 (11–12), 976–990.
- Kellenberger, E., Schalon, C., Rognan, D., 2008. How to measure the similarity between protein ligand-binding sites? *Current Computer Aided-Drug Design* 4 (3), 209–220.
- Konc, J., Janežič, D., 2014. Binding site comparison for function prediction and pharmaceutical discovery. *Current Opinion in Structural Biology* 25, 34–39.
- Pérot, S., Sperandio, O., Miteva, M.A., Camproux, A.-C., Villoutreix, B.O., 2010. Druggable pockets and binding site centric chemical space: A paradigm shift in drug discovery. *Drug Discovery Today* 15 (15–16), 656–667.

- Raymond, J.W., Willett, P., 2002. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design* 16 (7), 521–533.
- Shoemaker, B.A., Zhang, D., Tyagi, M., *et al.*, 2012. IBIS (Inferred Biomolecular Interaction Server) reports, predicts and integrates multiple types of conserved interactions for proteins. *Nucleic Acids Research* 40 (Database issue), D834–D840.
- Volkamer, A., Rarey, M., 2014. Exploiting structural information for drug-target assessment. *Future Medicinal Chemistry* 6 (3), 319–331.
- Xie, Z.-R., Hwang, M.-J., 2015. Methods for predicting protein-ligand binding sites. *Methods in Molecular Biology* 1215, 383–398.

Relevant Websites

<https://www.ccdc.cam.ac.uk/>
Cavbase.

<http://www.moldiscovery.com/software/flap/>
FLAP.

<http://bioinfo-pharma.u-strasbg.fr/labwebsite/download.html>
FuzCav, SiteAlign, TIFP, Shaper.

<http://biophys.umontreal.ca/nrg/NRG/IsoMIF.html>
IsoMIF.

<http://3d-e-chem.github.io/kripodb/>
KRIPO.

<http://proline.physics.iisc.ernet.in/pocketmatch/>
PocketMatch.

<http://www.biopharmics.com/>
PSIM.

<http://bioinfo3d.cs.tau.ac.il/SiteEngine/>
SiteEngine.

<http://compsci.hunter.cuny.edu/~leixie/smap/smap.html>
SMAP.

<https://zhanglab.ccmb.med.umich.edu/TM-align/>
TM-align.

Biographical Sketch

Christiane Ehrt received her MSc in Biochemistry at the Martin-Luther-University Halle-Wittenberg. Currently, she works as a PhD student at the Faculty of Chemistry and Chemical Biology at the TU Dortmund University in the group of Dr. Oliver Koch. Her interests and current focus include the identification and comparison of protein ligand binding sites, virtual screening studies for novel targets, and in this context, comparative modeling, MD simulations, and biochemical assay development. Tobias Brinkjost received his Diploma in Computer Science from the TU Dortmund University in 2012. Afterward he joined the Medicinal Chemistry group of Dr. Oliver Koch in collaboration with the Chair for Algorithm Engineering of Prof. Dr. Petra Mutzel at the TU Dortmund University for his PhD studies. His current research mainly focusses on the development of models and algorithms in the field of rational drug design. Oliver Koch studied pharmacy and computer science at the Philipps-University Marburg, Germany, where he also obtained his PhD in Pharmaceutical Chemistry with Prof. G. Klebe. After postdoctoral research at the Cambridge Crystallographic Data Center in 2008 and working in drug discovery at MSD Animal Health Innovation, he started his independent academic career in 2012 as an independent junior group leader for medicinal chemistry at the TU Dortmund University, Germany. His research interests involve the development and application of computational methods in computational molecular design and medicinal chemistry.

Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications

Swathik Clarancia Peter¹, Tamil Nadu Agricultural University, Coimbatore, India
Jaspreet Kaur Dhanja¹, Vidhi Malik, and Navaneethan Radhakrishnan, Indian Institute of Technology Delhi, New Delhi, India
Mannu Jayakanthan, Tamil Nadu Agricultural University, Coimbatore, India
Durai Sundar, Indian Institute of Technology Delhi, New Delhi, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Quantitative structure-activity relationship (QSAR) approach relies on the basic principle of chemistry that states that the biological activity of any ligand or compound is associated with the arrangement of atoms forming the molecular structure. In other words, structurally related molecules possess similar biological activities. This structural information can be defined in terms of a series of parameters called molecular descriptors. In QSAR, the biological activity is represented as a function of these molecular descriptors as depicted in Eq. (1).

$$\text{Biological response or activity} = f(\text{molecular descriptors}) \dots \quad (1)$$

The model thus developed based on the biological activities of known ligands is used to predict the response of new compounds.

QSAR finds applicability in a wide range of fields including toxicology (Wang *et al.*, 2014; Rochani *et al.*, 2010), ecotoxicology (Hermens *et al.*, 1984; Van Gestel and Ma, 1990; Escher *et al.*, 2006), drug design and discovery (Zernov *et al.*, 2003; Speck-Planche *et al.*, 2012; Buolamwini and Assefa, 2002), chemical data mining (Shen *et al.*, 2004), combinatorial library design (Ghose *et al.*, 1999; Zhang *et al.*, 2008) and so on.

QSAR studies therefore involve selection of active and inactive compounds with the measure of their biological activity, description and calculation of molecular descriptors, selection of appropriate features followed by construction of the mathematical model and its evaluation.

QSAR and QSPR

Quantitative structure-activity relationship (QSAR) prediction depends on the structure of molecules and atoms present in the compound. Biological activity is understood in terms of numerical values (example bioavailability, inhibitory concentration) and presence/absence of a condition (example infected/not infected, mutagenic/non mutagenic). Various QSAR studies have been carried out to understand biological properties such as pharmacokinetics (Vieira *et al.*, 2014; Gombar and Hall, 2013), blood brain barrier penetration (BBB) (Zhang *et al.*, 2008), carcinogenicity (Fjodorova *et al.*, 2010; Kar and Roy, 2011), drug metabolism (Braga and Andrade, 2012; Lewis, 2000), bio-concentration (Grisoni *et al.*, 2016; Papa *et al.*, 2007), permeability (Gozalbes *et al.*, 2011; Fujikawa *et al.*, 2007), drug clearance (Manga *et al.*, 2003; Boik and Newman, 2008), mutagenicity (Valencia *et al.*, 2013; Barber *et al.*, 2016), and so on. Another term associated with this approach is Quantitative structure-property relationship (QSPR). In QSPR, physiochemical properties of the chemical compounds are determined based on the molecular structure information. Physiochemical properties such as melting point (Katritzky *et al.*, 2002; Modarresi *et al.*, 2006), boiling point (Sola *et al.*, 2008; Dai *et al.*, 2013), solubility (Duchowicz and Castro, 2009; Gao *et al.*, 2002), stability (Dioury *et al.*, 2014; Ghasemi *et al.*, 2010), dielectric constant (Achary, 2014; Soltanpour *et al.*, 2016), reactivity (Toropov *et al.*, 2004), diffusion coefficient (Mirkhani *et al.*, 2012), thermodynamic properties (Puri *et al.*, 2002; Duchowicz *et al.*, 2006), hydrophobicity (Zou *et al.*, 2016; Berinde, 2013) have been exploited to determine quantitative structure-property relationships.

History

The concept of QSAR had begun a century ago. Crum-Brown and Fraser in 1868 proposed the physiological activity of molecules based on their composition (Crum-Brown and Fraser, 1868). Then the narcotic effect of the primary alcohols respective to the molecular weight was studied by Richardson in 1869 (Richardson, 1869). Following this, studies of simple organic compounds in response to water solubility (Richet, 1893), potency variation of narcotic compounds (Meyer, 1899), study of chemical reactivity of substituted benzenes (Hammett, 1937), narcotic study based on logP and thermodynamic (Ferguson, 1939), physical organic chemistry to linear steric energy relationships (Taft, 1952, 1953a, 1953b), linear free energy relationship model by Hansch and Fujita (1964), QSAR study based on molecular fragments by Free and Wilson (1964) and substituent-based structure-activity

¹Equal contribution.

relationship (Fujrra and Ban, 1971) are some of the important hallmarks in the history of QSAR. The development of 2D-QSAR began in early 1970s and the development of 3D-QSAR started in early 1980s. With the arrival of new technologies and perspectives, QSAR has now become multidimensional. More robust and accurate QSAR approaches came into practice with increased dimensionality like 4D, 5D and 6D, which has led to increased predictability, reliability and precision of the models.

Types of QSAR Methodologies

QSAR can be broadly classified in two ways (Qiao *et al.*, 2014). The first classification is based on the dimensions of the descriptors involved in the model such as 1D, 2D, 3D, 4D, 5D and so forth. Other is based on the type of biological activity predicted as a dependent variable. This includes Quantitative structure-toxicity relationship (QSTR) (Can, 2014), Quantitative structure-metabolism relationship (QSMR), Quantitative structure-reactivity relationship or Quantitative structure-retention relationship (QSRR) (Hemmateenejad *et al.*, 2009; Goryński *et al.*, 2013), Quantitative structure-permeability relationship or Quantitative structure-pharmacokinetics relationship (QSPR) (Moss *et al.*, 2002; Mayer and Van De Waterbeemd, 1985), Quantitative structure-bioavailability relationship or Quantitative structure-binding affinity relationship (QSBR) (Andrews *et al.*, 2000; Zhang *et al.*, 2006), and so forth.

QSAR models can also be grouped based on analysis of correlation-linear and non-linear (Roy and Mandal, 2008), or depending on binding nature of molecule and receptor-receptor dependent and receptor independent (Magdziarz *et al.*, 2009).

QSAR Model Construction

The quality of the QSAR model to a large extent depends on the data used for its construction. Hence, prior to the model development, it is necessary to gain insights about the data. Thorough understanding of the problem and influencing factors assists in discerning meaningful relationships. Relevant background information about the system under study either biological or chemical is to be collected via literature search. Data sets for model construction need to be chosen carefully as poor and inconsistent data would lead to corrupt model. There are several other factors like division of data into training and test data sets, molecular descriptors, statistical methods for model development that influence the quality of the QSAR model.

The schematic representation of QSAR model construction is given in Fig. 1. A brief description of the steps involved in QSAR model generation is discussed in the following section.

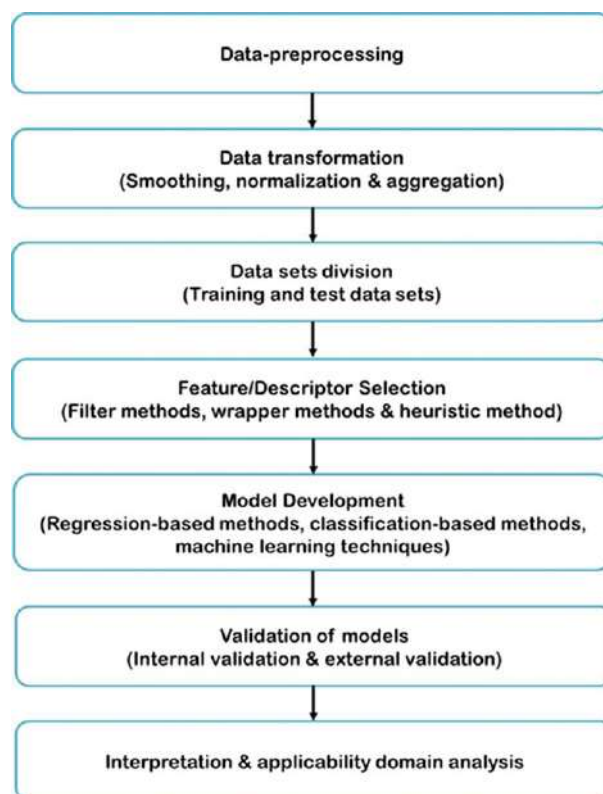


Fig. 1 Schematic representation of QSAR (quantitative structure-activity relationship) model development.

Data Pre-Processing

Pre-processing eliminates noise and redundancy in the data sets. It involves data transformation that includes smoothing, normalization and aggregation (Tomar and Agarwal, 2014), data reduction, sampling when data sets are large, noise elimination, feature selection, data cleaning, data integration when data is collected from heterogeneous sources, and discretization (Cocu *et al.*, 2008).

Training and Test Data Sets

The data is divided into training and test data sets. The training set is used to formulate the QSAR model, while the test set is used to evaluate its predictability and accuracy. Dataset is generally divided in such a way that both the sets occupy entire descriptor space. Employing appropriate data splitting techniques improves the model prediction. The various approaches available for the division of dataset include k-means clustering, based on X response, based on Y response, random selection, statistical molecular design, sphere exclusion, Kennard-Stone selection, Kohonen's self-organizing map selection, and extrapolation-oriented test set selection (Roy *et al.*, 2008).

Calculation of Molecular Descriptors

The information about the structure of molecules, defined by molecular descriptors obtained from different representations such as 2D, 3D, etc., is embedded into the QSAR model. Molecular conformations used should be correct for a better predictive model. There are different kinds of descriptors like count descriptors (0D), fingerprints (1D), topological descriptors (2D), geometrical (3D), grid based (4D) and so forth. The complexity of information and power to discriminate between similar structures as provided by different descriptors increases with dimensionality. 0D and 1D descriptors provide basic information like molecular weight and number of constituent elements that are directly derived from molecular formula. Net charge of the molecule is a 1D descriptor. Topological indices that are computed from the structural formula are 2D descriptors. These are based on the graph theory and reflect the connections in the structure. The most widely used topological descriptor is connectivity index proposed by Randić (2001); Li and Shi (2008); Kier (1985). Other topological indices are Wiener's index W (Ivanciuc, 2000; Wiener, 1947), Connectivity indices (Kier *et al.*, 1975), Kier Shape (Kier, 1985), Balaban J Index (Balaban, 1982) and Zagreb indices (Gutman and Trinajstić, 1972). The 3D descriptors are based on three-dimensional coordinates of atoms comprising the compounds. Some commonly used methods for calculating 3D descriptors are CoMFA, CoMSIA, CoMBINE, GERM, CoMMA, GRIND, WHIM, HoloQSAR and CoSA. 4D, 5D and 6D descriptors are multidimensional descriptors, which include the parameters involved in the structure and flexibility of the receptor-binding site in conjunction with ligand topology. 4D descriptors are based on reference grids and molecular dynamic simulations. Descriptors calculated using multiple conformations, orientations, protonation states and isosteromers of the ligand constitute 5D descriptors. The solvation terms constitute 6D descriptors. Molecular descriptors can be calculated using various software, some of which are listed in Table 1 (Damale *et al.*, 2014).

Feature Selection

Feature selection reduces the dataset horizontally. Among the large number of calculated descriptors, only few are chosen to define the model. Feature selection after descriptor calculation removes collinearity between the descriptor pairs. Selection of the most appropriate features is done using filter and wrapper methods (Goodarzi *et al.*, 2012). Filter methods involve filtering out descriptors, thereby reducing the pool size of descriptors based on inter-variable correlations. So, molecular descriptors that show inter-correlation are removed retaining only one descriptor from a pair (Roy *et al.*, 2015a). Descriptors with lowest variance are also removed. Filter methods use techniques like chi-square analysis, Shannon entropy, odds ratio, GSS coefficient (Liu, 2004), correlation based feature selection (Demel *et al.*, 2009), Fisher Score, Kolmogorov-Smirnov statistics (Guyon *et al.*, 2002), and principle component analysis. Distance based methods like Euclidean distance measures are also grouped under filter methods. Wrapper methods use regression-based approaches to select descriptors. In general, wrapper methods involve more computational power and perform better than filter methods. Recursive feature elimination (Xue *et al.*, 2004), variable selection and modeling based on the prediction (Liu *et al.*, 2003), k nearest neighbor, backward elimination, forward selection, genetic algorithm, Bayesian regularized neural network, factor analysis and combinatorial protocol are some of the commonly used wrapper methods. Hybrid methods are also being used that combine both filter and wrapper methods for selecting features (Goodarzi *et al.*, 2012).

Table 1 Software for calculation of molecular descriptors

Software	Description	Availability
ACD/LogP Freeware	logP prediction by fragment-based algorithm	Freely available
Dragon	Calculation of topological, constitutional & geometrical descriptors	Commercial
MOLGEN	Calculation of topological, constitutional & geometrical descriptors	Freely available
PaDEL Descriptors	Calculation of 2D & 3D descriptors	Freely available

Heuristic methods based on multiple linear regression are also used in selecting descriptors for QSAR models. It is fast when compared to other methods. It discards descriptors with constant values and removes descriptors whose values are not available for all the structures in the dataset thereby removing trivial descriptors. It also removes highly correlated descriptors (Liu and Long, 2009). Heuristic method has used for searching descriptor space and selection of vital descriptors in QSAR study of 1,4-dihydropyridine calcium channel antagonists (Si *et al.*, 2006), to select descriptors for multivariable linear model to predict the Percent of Applied Dose Dermally Absorbed (PADA) of the polycyclic aromatic hydrocarbons (Wang *et al.*, 2008), and descriptors selection in QSAR analysis of photosystem II electron transfer inhibitors (Karacan *et al.*, 2012).

QSAR Methods

Statistical methods are used in model construction and feature selection when there are large numbers of descriptors. They are helpful in obtaining functional endpoints. The statistical methods can be classified into regression-based approaches, classification based approaches, and machine learning techniques. QSAR modeling can be done for both linear and non-linear properties. Some of the methods for modeling linear properties include linear regression and partial regression, while artificial neural networks are being employed for modeling non-linear properties.

Models can be constructed using both supervised and unsupervised techniques. The chance effects in the unsupervised learning are less when compared to supervised learning, as it does not change to fit the model. Semi-supervised learning is advantageous over the supervised and unsupervised techniques. It considers both the labeled and unlabeled data giving better performance (Settles, 2012). Comparison of supervised learning algorithms and semi-supervised algorithms in different data sets suggested that semi-supervised learning can assist in understanding the addition of unlabeled data and hence is helpful for certain type of dataset and methods (Levatic *et al.*, 2013).

Regression based methods

Multiple linear regression

Multiple Linear Regression (MLR) method helps in establishing correlation between the independent and dependent variables. Here, the dependent variables are the biological activity or physiochemical property of the system that is being studied and the independent variables are molecular descriptors obtained from different representations. In linear regression models, the dependent variable is predicted using only one descriptor or feature. Multiple linear regression models consider more than one descriptor for the prediction of property/activity in question. The model based on the linear regression can be represented as a mathematical equation given below-

$$y = a + bx \dots \quad (2)$$

where, y is the dependent/response variable representing the physiochemical property or biological activity, x is the independent or predictor variable accounting for the molecular descriptor, and b is the regression coefficient.

Examples of QSAR studies involving the use of MLR method include the prediction of binding affinities of H3 antagonists (Dastmalchi *et al.*, 2012) and inhibitory activity of human non-pancreatic secretory phospholipase A₂ (Singh and Verma, 2014).

Partial least squares method

Partial Least Squares (PLS), developed from the principal component regression, helps in building models predicting more than one dependent variable (Lorber *et al.*, 1987). This method is used when the number of variables are more than the number of compounds in the datasets and where the variables considered for the study are correlated (Cramer, 1993). It is applied in 3D-QSAR technique, Comparative Molecular Field Analysis (CoMFA) to reduce the number of descriptors. PLS is also used in the validation metrics of the models (Mota *et al.*, 2009). It is advantageous over other regression models (Cramer, 1993). PLS has been used in the construction of many successful QSAR models. Prediction of binding affinity of polycyclic aromatic compounds with the rat liver 2,3,7,8-tetrachlorodibenzo-p-dioxin (TCDD) receptor (Johnels *et al.*, 1989) is one such example. PLS in combination with other methods – Genetic Partial Least Squares (G/PLS), Factor analysis Partial Least Squares (FA-PLS) and Orthogonal Signal Correction Partial Least Squares (OSC-PLS) (Liu and Long, 2009) is also being used for QSAR studies.

Classification based methods

Cluster analysis

Clustering involves placing similar data into a group in a way that maximizes similarity within groups and dissimilarity between groups. It involves methods like hierarchical clustering and k-means clustering. In hierarchical clustering, clusters are grouped on the basis of the dissimilarities calculated through the distances between the objects (Euclidean distances). The k-means clustering is a non-hierarchical method. It is based on k-centroids. Some of the other classification based methods are linear discriminant analysis and logistic regression (Roy *et al.*, 2015c; Agresti, 2007; Harrell, 2001).

Machine learning techniques

Artificial neural network

Artificial Neural Network (ANN) mimics the behavior of biological neurons. The ANN has input layer, hidden layer(s) and output layer. The molecular information is fed through the input layer, which is processed by a number of processing units in

parallel and the biological activity or property is obtained as an output. The commonly used ANNs in QSAR are back propagation neural networks, probabilistic neural networks, Kohonen self-organizing maps and Bayesian regularized neural networks. Neural networks can be supervised or unsupervised in nature. The learning is supervised when the trained model is validated by a separate test set. The training set helps in fitting weight parameters and decides the number of hidden layers in the network architecture. ANN methods have been proven to be highly adaptable and are deployed in modeling non-linear systems with high variability in data sets. Superior models can be obtained from ANNs compared to traditional approaches like MLR and PLR (Shi *et al.*, 2010). Implementing techniques like dropout in training data set reduces over-fitting of data and produces improved results when compared to conventional ANN models. However, Bayesian networks still outstand in their performance.

ANNs have been recently used in many QSAR studies. Some examples include the study of neurotrophic effects of N-p-Tolyl/phenylsulfonyl L-amino acid thioester derivatives (Luo *et al.*, 2011) and the study of antibacterial activity of oxazolidinone derivatives (Zou and Zhou, 2007).

Support vector machine

Support Vector Machine (SVM) is a machine learning approach that uses a linear classifier to classify data into two categories. The classifier is non-probabilistic. It performs better than other 3D QSAR models. In a comparative study of 3D QSAR modeling and SVMs for predicting the activity of BRAF-V600E and HIV integrase inhibitors, SVMs outperformed 3D-QSAR models (Wesley *et al.*, 2016). SVMs are also being used in combination with other methods like MLR, PLS and so forth for building more powerful and accurate QSAR models. In one of the QSAR reports where the anti-Alzheimer activity of triazolythiopenes as cyclin dependent kinase 5 inhibitors was studied (Garkani-Nejad and Ghanbari, 2016), MLR was used for selecting molecular descriptors, and support vector regression and PLS were used for constructing non-linear and linear models. When these methods were compared, support vector regression outperformed other methods. SVMs are employed in dimensionality reduction through variable ranking and selection (Bi *et al.*, 2003). SVMs also overcome the issue of over-fitting observed in artificial networks.

In a study conducted for predicting the reduction of dihydrofolate reductase by pyrimidines, SVMs outperformed three neural networks, namely radial basis function network, nearest neighbor classifier and decision tree (Burbidge *et al.*, 2001). QSPR models are also being developed by employing a combination of SVM with other methods like Principal Component Least Square methods and so forth (Veyseh *et al.*, 2015; Khorshidi *et al.*, 2014).

Gene expression programming

Gene Expression Programming (GEP) is based on the genetic algorithm and genetic programming. GEP has been used in QSAR modeling for the prediction of dermal penetration (PADA, Percent of Applied Dose Dermally Absorbed) of polycyclic aromatic hydrocarbons (Wang *et al.*, 2008), prediction of EC₅₀ of anti-HIV drugs (Si *et al.*, 2008), prediction of binding affinity of substituted 1-(3,3-diphenylpropyl)-piperidiny l amides and ureas with the chemokine receptor 5, and also for the prediction of toxicity of aromatic compounds (Shi *et al.*, 2010). GEP proved to be better in prediction when compared to the previously discussed methods like ANNs (Shi *et al.*, 2010), and SVMs (Si *et al.*, 2008). Further, Improved Gene Expression Programming (IGEP) proves to be more efficient than existing methods (Fu *et al.*, 2010).

Some of other methods deployed in QSAR studies include Monte Carlo Simulations (Kumar and Chauhan, 2017), principal component analysis (Suzuki *et al.*, 2001), and decision trees and random forest algorithm (Polishchuk *et al.*, 2009; Simeon *et al.*, 2016). Still newer methods like Projection Pursuit Regression (Du *et al.*, 2011) and Local Lazy Regression (Guha *et al.*, 2006; Lei *et al.*, 2010) are also implemented in QSAR models.

Software for QSAR Studies and Modeling

QSAR studies are carried out using various platforms that help in building models for predicting chemical, biological and toxicological activities. Some of these are listed in Table 2.

List of Databases Used in QSAR Studies

Attempt has also been made to archive the constructed QSAR models for further reference and usage. Some of these are listed in Table 3.

Validation of Models

Once the model is constructed, it must be validated. Validating the models avoid chance correlation of numerous descriptors used in the model and also over-fitting of data. It helps in assessing the accuracy and prediction of the model. The Organization for Economic Cooperation and Development (OECD) has put forth five principles to test the model. They are (1) a defined endpoint,

Table 2 Platforms used for QSAR modeling

Software	Description	Availability
3D-QSAR	To build 3-D QSAR models	Freely available
ACD/Tox Suite	Used for prediction of toxicity endpoints	Commercial, free web service
ADMET Predictor	Calculation of ADMET properties	Commercial
AZOrange	Machine learning platform for QSAR modeling	Freely available
BioPPsy	Prediction of pharmacokinetic properties of drug candidates by QSPR modeling	Freely available
BioTriangle	Web-based platform for calculating molecular descriptors	Freely available
BlueDesc	Molecular Descriptor Calculator	Freely available
CACTVS	Molecular Descriptor Calculator	Freely available
CAESAR	Models for developmental toxicity	Freely available
ChemDes	Descriptor and fingerprint calculation (web-based)	Freely available
CODESSA	Generate predictive QSAR models from Quantum chemical, topological & electrostatic descriptors	Commercial
CoFFer	Web-based QSAR service for the prediction of chemical compounds	Freely available
CORALSEA	Building of quantitative structure - property / activity relationships	Freely available
Derek	Rules based system with structural alerts for developmental toxicity, teratogenicity, testicular toxicity, and oestrogenicity	Commercial
DMax	Data mining tool for QSAR, virtual screening and compound screening data analysis	Freely available
DWFS	Parallel GA wrapper Feature selection (web-based)	Freely available
ECOSAR	Calculates aquatic toxicities	Freely available
EPISuite	Suite of programs for estimation of physiochemical property calculation and environmental fate	Freely available
eTOXlab	Development and validation of QSAR models	Freely available
GUSAR	Development of QSAR/QSPR models (web-based)	Freely available
HASL	Software package for 3D-QSAR	Commercial
HYBOT-PLUS	Descriptor calculation	Commercial
Leadscope	QSAR models to predict reproductive and developmental toxicity for rodent foetus	Commercial
Mathematica	Software package for ANN development	Commercial
Matlab	Software package for ANN development	Commercial
MC – 3DQSAR	Generates QSAR equations	Freely available
Molcode Toolbox	Prediction of toxicological endpoints	Commercial
MultiCASE	Models for developmental toxicity	Commercial
Neuralware	Software package for ANN development	Commercial
OECD QSAR Application Toolbox	QSAR models to fill data gaps & missing data	Freely available
PASS	Predicts biological activity using Bayesian algorithm, predicts embryotoxicity and teratogenicity	Commercial
QSARpro	Predicts activity and optimizes lead compounds using QSAR models	Commercial
SPSS	Software package for ANN development	Commercial
Statistica	Software package for ANN development	Commercial
TerraQSAR	Database compounds with structure-specific Biological activity	Freely available
T.E.S.T	Predicts toxicities of compounds by applying QSAR methodologies	Freeware
TIMES	Prediction of oestrogen, androgen and aryl hydrocarbon binding compound	Commercial
TOPKAT	Prediction of toxicological endpoints	Commercial
Toxmatch	Provides chemical similarity indices to assist in read-cross assessments & developing categories	Freely available
WEBCDK	Calculation of molecular descriptors (web-based)	Freely available
VCCL	Suite of programs for descriptor calculation, dimensionality reduction & data analysis	Freely available
VEGA-QSAR	QSAR models for regulatory purposes can be accessed and new QSAR models can be built	Freely available
NVirtualToxLab	Based on the combination of Auto flexible docking and mQSAR	Commercial

(2) an unambiguous algorithm, (3) a defined domain of applicability, (4) appropriate measures of goodness-of-fit, robustness and prediction accuracy, and (5) a mechanistic interpretation, if possible (Roy *et al.*, 2015b). Models can be validated through techniques such as internal validation, external validation, and cross validation (Veerasamy *et al.*, 2011). In internal validation, activity is predicted and parameters are estimated to analyze the precision of the prediction based on the compounds used for model construction. This is not suitable when new test set of compounds is used. But the external validation technique works well

Table 3 List of databases related to QSAR studies

Database	Description	URL
Danish QSAR database	Database of QSAR predictions	http://qsar.food.dtu.dk
ECOTOX	Online database that provides toxicity data on aquatic life, terrestrial plants & wild life	https://cfpub.epa.gov/ecotox/
EDKB (Endocrine Disruptor Knowledge Base)	Online database with predictive models to predict binding affinity of the compounds with oestrogen & androgen nuclear receptor proteins	http://edkb.fda.gov/webstart/edkb/index.html
JRC QSAR Model Database	Database of QSAR models	https://eurl-ecvam.jrc.ec.europa.eu/databases/jrc-qsar-model-database
MOE	Database of molecular data and QSAR modeling	https://www.chemcomp.com/MOE-Molecular_Operating_Environment.htm
MOLE db	Free database for molecular descriptors	http://michem.disat.unimib.it/mole_db/
QsarDB	Database of QSAR/QSPR models	https://qsar.db.org/

even with the new datasets. In this case, the dataset is divided into test and training data sets. Model validation is done by test set compounds that are independent of training set (Roy *et al.*, 2015b; Veerasamy *et al.*, 2011; Roy and Kar, 2015). However, external validation is not worthy as it leaves a large portion of data set for testing (Hawkins *et al.*, 2003). Golbraikh and Tropsha (2002) proposed high value of cross-validated R^2 to be one of the criteria to have high predictive power for a QSAR model. It was emphasized that depending on q^2 for predictivity is incorrect. Instead external data set should be used for validation to have high predictive power for the model (Golbraikh and Tropsha, 2002).

Validation metrics are of two types based on type of QSAR model (Kar and Roy, 2011; Roy *et al.*, 2015b).

1. Regression-based QSAR models
2. Classification-based QSAR models

Both regression and classification based methods have their unique metrics for validation.

Validation Metrics for Regression-Based Methods

Validation metrics for regression-based models are calculated for both internal and external validation strategies.

Validation metrics for internal validation

Internal validation of QSAR models employs the use of molecules from training set to test the predictability of the model. Some of the most common methods used for internal validation of QSAR models are described here.

Least square fitting

Least square fitting is similar to linear regression and is the most commonly used validation method. It is the measure of square correlation coefficient (R^2) between the predicted and experimental value of activity. Outliers can be removed from the training data set, in order to optimize QSAR model, if difference between R^2 and R^2_{adj} is less than 0.3 (Veerasamy *et al.*, 2011).

Chi-squared (χ^2) and root-mean squared error (RMSE)

The χ^2 and RMSE values are used to assess the predictive quality of a model. χ^2 value shows the difference between experimentally determined bioactivity values and the values predicted by the model, whereas the RMSE value is the depiction of error between the mean of experimental and predicted activity values. Even for models with large R^2 value (that is ≥ 0.7), values of χ^2 and RMSE should be lower than 0.5 and 0.3 respectively, for good predictive ability of the model (Veerasamy *et al.*, 2011).

Cross validation

Cross validation approaches for internal validation include Leave-Group-Out (LGO), which involves leaving of a molecule or a group of molecules while creating model and evaluating the predictability of the model using the molecules left. Some of the important measures used in the internal cross validation of QSAR models are listed below.

Leave-One-Out (LOO) cross validation

In LOO cross validation, one compound is left out and the QSAR model is constructed using remaining compounds. The eliminated compound is used as a test for the predicted model. This process is repeated eliminating each of the compounds in the dataset one by one. The results so obtained from this are used for estimation of parameters involved in validation metrics. The predictability of the model is assessed by Predicted Residual Sum of Squares (PRESS) and cross-validated R^2 (Q^2) when Standard Deviation of Error of Prediction (SDEP) is obtained from PRESS (Roy and Kar, 2015; Roy *et al.*, 2015b).

Leave-Some-Out cross validation

In case of Leave-Some-Out (LSO) or Leave-Many-Out (LMO) a set of data compounds are eliminated and models are created with rest of the compounds. The left out compounds are then used to check the predictability of the model. Similar to LOO approach, LMO approach also involves repetitive cycles of elimination and model creation until each and every compound has been treated as a test set (Veerasamy *et al.*, 2011; Kar and Roy, 2011; Roy *et al.*, 2015b). On completion of all cycles of model training and testing, overall LMO- Q^2 value is calculated based on the compounds' predicted activity values. LMO methods is more consistent as compared to LOO (Veerasamy *et al.*, 2011).

Value of Q^2 is usually smaller than value of R^2 . In order to avoid over-fitting of the model, the difference between R^2 and Q^2 should not exceed 0.3. Over-fitted model may very well predict the activity of compounds for training set but for new compounds predictivity is compromised (Veerasamy *et al.*, 2011).

True Q^2 and r_m^2 metrics

True Q^2 proposed by Hawkins *et al.* is used for small data sets and r_m^2 metric is calculated based on the scaled values of observed and predicted activity. Q^2 should not be treated as an ultimate proof for good predictability of models. Value of Q^2 higher than 0.5 should not be interpreted as high predictive power of QSAR models; until the ability of model to predict the activity of large number of compounds that are not used for training of model is tested (Golbraikh and Tropsha, 2002). Some of the other metrics used for internal validation are true r_m^2 (LOO) and Y-Randomization, a metric for chance correlation. The true r_m^2 (LOO) metric reveals external validation characteristics as its value is derived from the model developed after repetitive cycles of LOO. Y randomization test involves process randomization and model randomization to validate the model by permuting the response values with respect to unaltered matrix (Roy *et al.*, 2015b).

Validation metrics for external validation

The validation metrics employed in external validation are as following:

- Predictive R^2 that can also be given as $Q^2_{(F1)}$ says about the correlation of observed and predicted data. Model is said to have good predictive power if the value of $Q^2_{(F1)}$ is greater than 0.5 (Roy *et al.*, 2015b).
- $Q^2_{(F2)}$ and $Q^2_{(F3)}$ using the mean of test data set and training data set respectively (Schüürmann *et al.*, 2008). For validation of QSAR model, threshold value of 0.5 is defined for both metrics (Roy *et al.*, 2015b).
- Golbraikh and Tropsha's criteria puts forth condition for selection of training and test data sets. For having a good predictive power, QSAR model should satisfy following conditions (Golbraikh and Tropsha, 2002; Veerasamy *et al.*, 2011):
 - $Q^2_{\text{training}} > 0.5$
 - $R^2_{\text{test}} > 0.6$
 - $(r^2 - r_0^2)/r^2 < 0.1$ or $(r^2 - r_0'^2)/r^2 < 0.1$, where r_0^2 is R^2 of predicted *vs.* observed activities and $r_0'^2$ is R^2 of observed *vs.* predicted activities.
 - $0.85 \leq k \leq 1.15$ or $0.85 \leq k' \leq 1.15$, where k and k' are the slopes of regression lines through the origin.
- Other metrics includes Root Mean Square Error of Prediction (RMSEP) to calculate prediction error of QSAR model (Roy *et al.*, 2015b); Concordance Correlation Coefficient (CCC), the most restrictive and precautionary measure, with ideal value of 1 (Chirico and Gramatica, 2011); r_m^2 (rank) which makes rank order predictions (Roy *et al.*, 2015b); and r_m^2 (test) to understand the relationship between observed and predicted values (Roy and Mitra, 2011; Roy *et al.*, 2015b).

Validation Metrics for Classification-Based Methods

The validation matrix employed in classification-based methods is the Wilks lambda (λ) statistics. It is used to test the significance of discriminant model function and is calculated as the ratio of within-category sum of squares to total dispersion. The value ranges between $0 < \lambda < 1$, with lower value corresponding to higher level of discrimination. Further, canonical index (Rc) is used to estimate the strength of relationship between various dependent and independent variables; Chi-square (χ^2) to check the quality of the classification based model; and Squared Mahalanobis distance is a measure calculated using random data points (Roy and Mitra, 2011; Roy *et al.*, 2015b).

Interpretation & Applicability Domain Analysis

The parameters or descriptors used in the model should be interpretable. Mechanistic interpretation of the built QSAR model helps in understanding the influence of descriptors in the predicted activity. Applicability domain analysis helps us to understand whether the built QSAR model can be used for any set of compounds. The applicability domain model is built on the theoretical region present in the chemical space of descriptors and activity modeled. It enables to understand the feasibility of activity or response prediction by the constructed QSAR model for a given set of compounds. So, the QSAR model prediction for a set of compounds is only reliable if the chemical space or applicability domain of those compound falls within the applicability domain of the compounds used for training the model (Roy *et al.*, 2015b). The theoretical region in the chemical space is identified or estimated by different methods. Application domain assessment is done through probability density distribution, geometrical

methods, distance-based methods, ranges in descriptor space when descriptor space are used, and the range of response variable when modeled response space is used (Jaworska *et al.*, 2005).

Multidimensional QSAR

QSAR started with 0D and has evolved to 6D. Each dimension arose as a need to overcome the limitations of previous dimensions and to be more advantageous than the former. 1D QSAR calculated the molecular properties such as electronic, hydrophobic, steric and so forth. 2D QSAR considers geometric parameters, topological indices, molecular fingerprints, polar surface area but it excludes steric properties. 3D QSAR technique focuses on the spatial properties of the compound. So, the drawbacks of 2D QSAR get addressed in 3D QSAR method. The descriptor methods used in 3D QSAR are alignment-dependent and alignment-independent. The alignment dependent methods are Comparative Molecular Field Analysis (CoMFA), Comparative Molecular Similarity Indices Analysis (CoMSIA), Comparative Binding Energy Analysis (CoMBINE), Comparative Residue Interaction Analysis (CoRIA), Hint Interaction Field Analysis (HIFA) and so forth. The alignment independent methods include Comparative Molecular Moment Analysis (CoMMA), Comparative Spectral Analysis (CoSA), and Holo-QSAR (HQSAR). 3D QSAR employs methods like Artificial Neural networks (ANN), Partial Least Squares Method (PLS), cluster analysis, and principal component analysis, and others for descriptor selection makes it more powerful than 2D QSAR. Even then, the 3D QSAR technique faces difficulties when there are large numbers of compounds in the data set, and has to compromise with the prediction accuracy. To overcome these drawbacks, 4D QSAR evolved. 4D-QSAR with fourth dimension of ensemble sampling addresses the issues of 3D QSAR. It includes descriptors for gird occupancy measures. 4D-QSAR can be applied for both, receptor independent and receptor dependent analysis (Hopfinger *et al.*, 1997). Further, 5D-QSAR evolved with the addition of new dimension to the 4D-QSAR, the new dimension being the multiple ligand topology representations. Ensembles of multiple representations deployed in 5D-QSAR makes this approach less biased when compared to 4D QSAR (Vedani and Dobler, 2002). 6D-QSAR improves the former 5D-QSAR strategy by including another dimension for solvation function that helps in analyzing different solvation models (Polanski, 2009).

QSAR in Drug Designing and Discovery

Drug design and discovery is a laborious and time-consuming process. It takes roughly 10–12 years for a molecule to be identified and approved as a drug. Most of the drugs fail during pre-clinical and clinical trials. The cost of drug discovery is very high. Determining drug candidates through QSAR studies would reduce cost of production and failure at an early stage. QSAR approaches help in identifying hits from a large library of compounds. The identified hit molecules can be purchased and studied for activity through experiments (Tang *et al.*, 2009; Montero-Torres *et al.*, 2006). The molecules with proven activity can be further optimized to design promising drug candidates. Thereby, QSAR studies avoid synthesis and testing of large number of compounds saving enormous time and cost.

Case Studies

A number of success stories reflecting the potential of QSAR in building reliable models have been reported in the literature. We have discussed here a few studies that dealt with a wide range of problems to get insights into the applications of QSAR approach.

The first example presented here is a QSAR model for designing better drugs for combating cancer. Apoptosis is an important process that can decide the fate of a cell. Malfunctioning of the process with atypical expression of B-cell lymphoma-2 (Bcl-2) anti-apoptotic proteins is a promising hallmark of cancer and in most of the cases results in resistance to chemo and radiotherapy. Multiple approaches including antisense oligonucleotides (ASOs), peptides and small molecule inhibitory compounds have been designed against these anti-apoptotic proteins. However, low cost and easy delivery makes small molecules a method of choice. To develop better compounds on the basis of available information, various QSAR models have been generated. But these models are limited to a single scaffold. In one such study, attempt was made to develop QSAR models for seven different classes of inhibitors reported in literature targeting Bcl-2 and Bcl-xL (Kanakaveti *et al.*, 2017). 453 such small molecule inhibitors with known IC_{50} (ranging from 1 nM to 100 μ M) were grouped into seven categories comprising of Apogossypol (89 compounds), Quinazoline thione (51 compounds), Pyrazole pyrimidine phenyl acyl (110 compounds), Quinolone (56 compounds), Thiomorpholine (42 compounds), Benzothiazole hydrazine (78 compounds) and Polyquinoline (27 compounds) depending upon the core structure. A total of 787 features (including constitutional, topological, electrostatic, geometrical and physicochemical descriptors) were tested. The most relevant descriptors were chosen and fitted using multiple linear regression analysis. Two to three parametric models were generated for all the different classes of compounds. ($n - 1$) and ($n - 10$) leave-out cross validations were used to evaluate the performance of the generated models. The QSAR analysis resulted in models showing Pearson correlation coefficient ranging from 0.95 to 0.985. Three already known inhibitors- ABT-199, Navitoclax and Sabutoclax were tested against their generated models. The predicted IC_{50} was comparable to the reported activity. A correlation between pIC_{50} and pKi ($-\log Ki$) was delineated and also found commonalities in the activity shown by the seven families with respect to structural disparities using an

approach called similarity-descriptor coupling. The study has been translated into a user-friendly webtool to predict a pan or specific inhibitor for Bcl-2 and Bcl-xL targets (Kanakaveti *et al.*, 2017).

This second example illustrates how pharmacokinetic properties of drug molecules can be predicted using QSAR modeling. Penetration through blood brain barrier (BBB) is one of the most important features that reflect the drug-likeness of small molecular compounds. Drugs designed for central nervous system should be able to cross this barrier however penetration of BBB by peripherally acting drugs should be minimize to avoid side effects. So estimation of pharmacokinetic properties of candidate compounds *in silico* before testing them experimentally can save enormous time and money. In this study, a set of 529 organic compounds was used to correlate their chemical structures and distribution coefficients between the brain and blood using artificial neural networks. The molecular structures were defined in terms of occurrence of number of various types of fragments containing up to 10 atoms. Out of all the fragment descriptors, the most important ones were selected using the stepwise multiple linear regression. It was seen that the best model was based on the fragments comprising up to nine atoms. The model was further validated using a dataset of 2053 compounds, which was categorized as BBB+ (penetrating) and BBB- (non-penetrating). The constructed model could correctly classify 90% of BBB+ compounds from a test set, however the prediction specificity for BBB- category was low. Some of the features that have the strongest positive effects on the LogBB included hydrophobic fragments like alkyl or aromatic and presence of hydroxyl group in α - position to the carbonyl group. On the other hand, it was found that permeability decreases if the structure contains strongly polar groups like hydroxyl, carboxyl, and guanidine. This model has been integrated into a web service for predicting ADMET parameters of drugs developed in the Laboratory of Medicinal Chemistry, Department of Chemistry, Lomonosov Moscow State University (Dyabina *et al.*, 2016).

With increasing industrialization, degradation of pollutants has become a crucial need to keep the environment clean. Benzene and its derivatives are the most common chemical structure found in the nature. It is also a structural part of degradation intermediates of complex pollutants like pesticides, pharmaceuticals, surfactants or synthetic dyes. However, the chemical properties deciding their environmental fate and behavior depends on the type, number and position of functional groups present at substitution sites. Here QSAR proved to be a fast and reliable approach for testing the degradation of aromatic compounds. Thirty six congeneric single-benzene ring compounds with known biodegradability were divided into training and test sets (24 and 12 compounds, respectively). The semi-empirical quantum-chemical descriptors like dipole moment, energy of the highest occupied molecular orbital (EHOMO), energy of the lowest unoccupied molecular orbital (ELUMO), energetic difference between EHOMO and ELUMO, final heat of formation and ionization potential, and various other molecular descriptors were calculated using various software. The correlation between descriptors and biodegradability prior and at half-life was obtained using variable selection Genetic Algorithm and Multiple Linear Regression Analysis methods. The validation of best-selected models based on statistical parameters was performed using Leave Many Out and "Y-scrambling" tests. The generated models were thus used to study the key structural features that influence the biodegradability of the compound of interest and correlate to its degradation mechanisms by UV-C/H₂O₂. Molecular mass, number of C-C bonds determining the rate of saturation of benzene ring making it susceptible to cleavage into more readily degradable aliphatic compounds, electron donating/withdrawing groups, symmetry of the molecule, presence of sulfo-group, ionization potential and electrotopological states were some of the important descriptors related to the biodegradability of aromatic compounds in water. The potential of such models can be extended to more extensive purposes like risk assessment studies (Cvetnic *et al.*, 2017).

In the following example, structural factors of ionic liquids (ILs) have been correlated with the process of micelization. Critical Micelization Concentration (CMC) of any ionic liquid depends upon the types of ions it possess. Micelization affects the synthesis, purification and regeneration routes of these ionic liquids and thus is an important feature to be considered. In this study, an attempt was made to derive a qualitative relationship between structural features of ions and their effect on micelization of ILs. It was also verified whether the micelization process is governed by the constituent ions separately or they have additive effects. Literature was explored to collect experimental data of ILs with their CMC. A dataset of 59 structurally diversified IL's with the CMC ranging between the 0.098 and 902 mM was prepared. 42 compounds were used to train the model while rests were used for testing. Various molecular descriptors were generated using DRAGON software and the most appropriate ones were chosen using genetic algorithm implemented in QSARINS software. The correlation between the descriptors and CMCs was derived using Multiple Linear Regression (MLR) technique. Various statistical parameters like determination coefficient, Concordance Correlation Coefficient, Root Mean Square Error, Mean Average Error and F-value were used to evaluate the fitting of model and its significance. The leave-one-out and Y-scrambling approach were used for internal validation and investigating the robustness of the generated model. An altogether new test data set was used for external validation. Decrease in CMC was attributed to less spherical, improperly folded cations containing larger hydrophobic domain. For anions, bigger size was associated with decrease in CMC. Also the effect of cations and anions in determining the CMC was found to be independent of each other (Barycki *et al.*, 2017).

People today have become more careful about their eating habits to adopt a healthy life style. They avoid overconsumption of high-calorific food to reduce the risk of various metabolic disorders, cardiovascular diseases, obesity and diabetes. Sugars or saccharides are major contributors in this. Industry is focusing on finding out new compounds, natural or synthetic, with low calories but high sweetness. There are various QSAR models in the literature that help in extracting various structural features responsible for imparting the sweetness to the compounds, or distinguishing between sweet and non-sweet compounds. A study was carried out for virtually screening known natural compounds using QSAR modeling for identification of new sweeteners (Chéron *et al.*, 2017). A database of 316 compounds belonging to seventeen chemical families and sweetness values ranging from 0.20 to 225,000 was created. The sweetness index for all the compounds was relative to sucrose. The protonation state of compounds was adjusted according to the pH value of saliva, i.e., 6.5. Descriptors were calculated for both 2D and 3D structures of

compounds using Dragon software. Random Forest and Support Vector Regression, machine learning algorithms were used to generate the models. Training set consisted of 225 molecules while the test set had 91 molecules. Leave-one-out cross-validation methods were used for both 2D and 3D QSAR models. Additional filters to avoid undesirable properties like bitterness and toxicity were applied. It was found that majority of the identified natural sweeteners were from terpene family, less than 200 molecules belonged to the category of saccharides, polyphenols or phenylpropanoids. The most potent natural sweeteners were based on saponin and stevioside scaffolds, with 1000–10,000 times more sweetness than sucrose (Chéron *et al.*, 2017).

See also: Biomolecular Structures: Prediction, Identification and Analyses. Chemoinformatics: From Chemical Art to Chemistry in Silico. Cross-Validation. Data Mining: Clustering. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer. Kernel Methods: Support Vector Machines. Measurements of Accuracy in Biostatistics. Molecular Dynamics and Simulation. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Parametric and Multivariate Methods. Protein Properties. Public Chemical Databases. Rational Structure-Based Drug Design. Regression Analysis. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow

References

- Achary, P., 2014. QSPR modelling of dielectric constants of π -conjugated organic compounds by means of the CORAL software. SAR and QSAR in Environmental Research 25, 507–526.
- Agresti, A., 2007. An Introduction to Categorical Data Analysis. John Wiley.
- Andrews, C.W., Bennett, L., Lawrence, X.Y., 2000. Predicting human oral bioavailability of a compound: Development of a novel quantitative structure-bioavailability relationship. Pharmaceutical Research 17, 639–644.
- Balaban, A.T., 1982. Highly discriminating distance-based topological index. Chemical Physics Letters 89, 399–404.
- Barber, C.E., Marshall, D.A., Mosher, D.P., *et al.*, 2016. Development of system-level performance measures for evaluation of models of care for inflammatory arthritis in Canada. The Journal of Rheumatology. 150839. [rheum].
- Barycki, M., Sosnowska, A., Puzyn, T., 2017. Which structural features stand behind micelization of ionic liquids? Quantitative structure-property relationship studies. Journal of Colloid and Interface Science 487, 475–483.
- Berinde, Z., 2013. A QSPR study of hydrophobicity of phenols and 2-(aryloxy- α -acetyl)-phenoxythiuron derivatives using the topological index ZEP. Creative Mathematics and Informatics 22, 33–40.
- Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M., 2003. Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research 3, 1229–1243.
- Boik, J.C., Newman, R.A., 2008. Structure-activity models of oral clearance, cytotoxicity, and LD50: A screen for promising anticancer compounds. BMC Pharmacology 8, 12.
- Braga, R.C., Andrade, C.H., 2012. QSAR and QM/MM approaches applied to drug metabolism prediction. Mini Reviews in Medicinal Chemistry 12, 573–582.
- Buolamwini, J.K., Assefa, A., 2002. CoMFA and CoMSIA 3D QSAR and docking studies on conformationally-restrained cinnamoyl HIV-1 integrase inhibitors: Exploration of a binding mode at the active site. Journal of Medicinal Chemistry 45, 841–852.
- Burbidge, R., Trotter, M., Buxton, B., Holden, S., 2001. Drug design by machine learning: Support vector machines for pharmaceutical data analysis. Computers & Chemistry 26, 5–14.
- Can, A., 2014. Quantitative structure-toxicity relationship (QSTR) studies on the organophosphate insecticides. Toxicology Letters 230, 434–443.
- Chéron, J.-B., Casciuc, I., Golebiowski, J., Antonczak, S., Fiorucci, S., 2017. Sweetness prediction of natural compounds. Food Chemistry 221, 1421–1425.
- Chirico, N., Gramatica, P., 2011. Real external predictivity of QSAR models: How to evaluate it? Comparison of different validation criteria and proposal of using the concordance correlation coefficient. Journal of Chemical Information and Modeling 51, 2320–2335.
- Cocu, A., Dumitriu, L., Craciun, M., Segal, C., 2008. A Hybrid Approach for Data Preprocessing in the QSAR Problem. Knowledge-Based Intelligent Information and Engineering Systems. Springer. pp. 565–572.
- Cramer, R.D., 1993. Partial least squares (PLS): Its strengths and limitations. Perspectives in Drug Discovery and Design 1, 269–278.
- Crum-Brown, A., Fraser, T., 1868. On the connection between chemical constitution and physiological action. Part 1. On the physiological action of the ammonium bases, derived from Strychia, Brucia, Thebaia, Codeia, Morphia and Nicotia. Transactions of the Royal Society of Edinburgh 25, 151–203.
- Cvetnic, M., Perisic, D.J., Kovacic, M., *et al.*, 2017. Prediction of biodegradability of aromatics in water using QSAR modeling. Ecotoxicology and Environmental Safety 139, 139–149.
- Dai, Y.-M., Zhu, Z.-P., Cao, Z., *et al.*, 2013. Prediction of boiling points of organic compounds by QSPR tools. Journal of Molecular Graphics and Modelling 44, 113–119.
- Damale, M.G., Harke, S.N., Kalam Khan, F.A., Shinde, D.B., Sangshetti, J.N., 2014. Recent advances in multidimensional QSAR (4D-6D): A critical review. Mini Reviews in Medicinal Chemistry 14, 35–55.
- Dastmalchi, S., Hamzeh-Mivehroud, M., Asadpour-Zeynali, K., 2012. Comparison of different 2D and 3D-QSAR methods on activity prediction of histamine H3 receptor antagonists. Iranian Journal of Pharmaceutical Research: IJPR 11, 97.
- Demel, M.A., Janacek, A.G., Gansterer, W.N., Ecker, G.F., 2009. Comparison of contemporary feature selection algorithms: Application to the classification of ABC-transporter substrates. Molecular Informatics 28, 1087–1091.
- Dioury, F., Duprat, A., Dreyfus, G.R., Ferroud, C., Cossy, J., 2014. QSPR prediction of the stability constants of gadolinium (III) complexes for magnetic resonance imaging. Journal of Chemical Information and Modeling 54, 2718–2731.
- Du, H., Hu, Z., Bazzoli, A., Zhang, Y., 2011. Prediction of inhibitory activity of epidermal growth factor receptor inhibitors using grid search-projection pursuit regression method. PLOS ONE 6, e22367.
- Duchowicz, P.R., Castro, E.A., 2009. QSPR studies on aqueous solubilities of drug-like compounds. International Journal of Molecular Sciences 10, 2558–2577.
- Duchowicz, P.R., Castro, E.A., Fernandez, F., Pankratov, A., 2006. QSPR evaluation of thermodynamic properties of acyclic and aromatic compounds. Anales de la Asociación Química Argentina. SciELO Argentina. 31–45.
- Dyabina, A., Radchenko, E., Palyulin, V., Zefirov, N., 2016. Prediction of blood-brain barrier permeability of organic compounds. Doklady Biochemistry and Biophysics. Springer. pp. 371–374.
- Escher, B.I., Bramaz, N., Richter, M., Lienert, J., 2006. Comparative ecotoxicological hazard assessment of beta-blockers and their human metabolites using a mode-of-action-based test battery and a QSAR approach. Environmental Science & Technology 40, 7402–7408.

- Ferguson, J., 1939. The use of chemical potentials as indices of toxicity. *Proceedings of the Royal Society of London Series B, Biological Sciences* 127, 387–404.
- Fjodorova, N., Vračko, M., Novič, M., Roncaglioni, A., Benfenati, E., 2010. New public QSAR model for carcinogenicity. *Chemistry Central Journal*. Springer. p. S3.
- Free, S.M., Wilson, J.W., 1964. A mathematical contribution to structure-activity studies. *Journal of Medicinal Chemistry* 7, 395–399.
- Fujikawa, M., Nakao, K., Shimizu, R., Akamatsu, M., 2007. QSAR study on permeability of hydrophobic compounds with artificial membranes. *Bioorganic & Medicinal Chemistry* 15, 3756–3767.
- Fujira, T., Ban, T., 1971. Structure-activity study of phenethylamines as substrates of biosynthetic enzymes of sympathetic enzymes of sympathetic transmitters. *ZMed. CRem* 14, 148–152.
- Fu, W., Zhang, Y., Cheng, Z., 2010. Improved gene expression programming and its application to QSAR. In: *Proceedings of the Sixth International Conference on Natural Computation (ICNC)*, IEEE, 4057–4061.
- Gao, H., Shanmugasundaram, V., Lee, P., 2002. Estimation of aqueous solubility of organic compounds with QSPR approach. *Pharmaceutical Research* 19, 497–503.
- Garkani-Nejad, Z., Ghanbari, A., 2016. Application of support vector machine in QSAR study of triazolyl thiophenes as cyclin dependent kinase-5 inhibitors for their anti-alzheimer activity.
- Ghasemi, J.B., Ahmadi, S., Ayati, M., 2010. QSPR modeling of stability constants of the Li-hemispherands complexes using MLR: A theoretical host-guest study. *Macrocyclics* 3, 234–242.
- Ghose, A.K., Viswanadhan, V.N., Wendoloski, J.J., 1999. A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. 1. A qualitative and quantitative characterization of known drug databases. *Journal of Combinatorial Chemistry* 1, 55–68.
- Golbraikh, A., Tropsha, A., 2002. Beware of q²! *Journal of Molecular Graphics and Modelling* 20, 269–276.
- Gombar, V.K., Hall, S.D., 2013. Quantitative structure-activity relationship models of clinical pharmacokinetics: Clearance and volume of distribution. *Journal of Chemical Information and Modeling* 53, 948–957.
- Goodarzi, M., Dejaegher, B., Heyden, Y.V., 2012. Feature selection methods in QSAR studies. *Journal of AOAC International* 95, 636–651.
- Goryński, K., Bojko, B., Nowaczyk, A., *et al.*, 2013. Quantitative structure-retention relationships models for prediction of high performance liquid chromatography retention time of small molecules: Endogenous metabolites and banned compounds. *Analytica Chimica Acta* 797, 13–19.
- Gozalbes, R., Jacewicz, M., Annand, R., Tsaioun, K., Pineda-Lucena, A., 2011. QSAR-based permeability model for drug-like compounds. *Bioorganic & Medicinal Chemistry* 19, 2615–2624.
- Grisoni, F., Consonni, V., Vighi, M., Villa, S., Todeschini, R., 2016. Investigating the mechanisms of bioconcentration through QSAR classification trees. *Environment International* 88, 198–205.
- Guha, R., Dutta, D., Jurs, P.C., Chen, T., 2006. Local lazy regression: Making use of the neighborhood to improve QSAR predictions. *Journal of Chemical Information and Modeling* 46, 1836–1847.
- Gutman, I., Trinajstić, N., 1972. Graph theory and molecular orbitals. Total (ϕ)-electron energy of alternant hydrocarbons. *Chemical Physics Letters* 17, 535–538.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422.
- Hammett, L.P., 1937. The effect of structure upon the reactions of organic compounds. Benzene derivatives. *Journal of the American Chemical Society* 59, 96–103.
- Hansch, C., Fujita, T., 1964. ρ - σ - π Analysis. A method for the correlation of biological activity and chemical structure. *Journal of the American Chemical Society* 86, 1616–1626.
- Harrell, F., 2001. *Regression modeling strategies*. 2001. Nashville: Springer CrossRef Google Scholar.
- Hawkins, D.M., Basak, S.C., Mills, D., 2003. Assessing model fit by cross-validation. *Journal of Chemical Information and Computer Sciences* 43, 579–586.
- Hemmateenejad, B., Sanchooli, M., Mehdipour, A., 2009. Quantitative structure-reactivity relationship studies on the catalyzed Michael addition reactions. *Journal of Physical Organic Chemistry* 22, 613–618.
- Hermens, J., Canton, H., Janssen, P., De Jong, R., 1984. Quantitative structure-activity relationships and toxicity studies of mixtures of chemicals with anaesthetic potency: Acute lethal and sublethal toxicity to *Daphnia magna*. *Aquatic Toxicology* 5, 143–154.
- Hopfinger, A., Wang, S., Tokarski, J.S., *et al.*, 1997. Construction of 3D-QSAR models using the 4D-QSAR analysis formalism. *Journal of the American Chemical Society* 119, 10509–10524.
- Ivanciuc, O., 2000. QSAR comparative study of Wiener descriptors for weighted molecular graphs. *Journal of Chemical Information and Computer Sciences* 40, 1412–1422.
- Jaworska, J., Nikolova-Jeliazkova, N., Aldenberg, T., 2005. QSAR applicability domain estimation by projection of the training set descriptor space: A review. *ATLA-NOTTINGHAM* 33, 445.
- Johnels, D., Gillner, M., Nördén, B., Toftgård, R., Gustafsson, J.Å., 1989. Quantitative structure-activity relationship (QSAR) analysis using the partial least squares (PLS) method: The binding of polycyclic aromatic hydrocarbons (PAH) to the rat liver 2, 3, 7, 8-tetrachlorodibenzo-P-dioxin (TCDD) receptor. *Molecular Informatics* 8, 83–89.
- Kanakaveti, V., Sakthivel, R., Rayala, S., Gromiha, M.M., 2017. Importance of functional groups in predicting the activity of small molecule inhibitors for Bcl-2 and Bcl-xL. *Chemical Biology & Drug Design*.
- Kar, S., Roy, K., 2011. Development and validation of a robust QSAR model for prediction of carcinogenicity of drugs. *Indian Journal of Biochemistry & Biophysics* 48 (2), Karacan, M.S., Yakan, Ç., Yakan, M., *et al.*, 2012. Quantitative structure-activity relationship analysis of perfluoroisopropylidinitrobenzene derivatives known as photosystem II electron transfer inhibitors. *Biochimica et Biophysica Acta (BBA)-Bioenergetics* 1817, 1229–1236.
- Katritzky, A.R., Lomaka, A., Petrukhin, R., *et al.*, 2002. QSPR correlation of the melting point for pyridinium bromides, potential ionic liquids. *Journal of Chemical Information and Computer Sciences* 42, 71–74.
- Khorshidi, N., Sarkhosh, M., Niazi, A., 2014. QSPR study of maximum absorption wavelength of various flavones by multivariate image analysis and principal components-least squares support vector machine. *Journal of Scientific and Innovative Research* 3, 189–202.
- Kier, L.B., 1985. A shape index from molecular graphs. *Molecular Informatics* 4, 109–116.
- Kier, L.B., Hall, L.H., Murray, W.J., Randi, M., 1975. Molecular connectivity I: Relationship to nonspecific local anesthesia. *Journal of Pharmaceutical Sciences* 64, 1971–1974.
- Kumar, A., Chauhan, S., 2017. Monte Carlo method based QSAR modelling of natural lipase inhibitors using hybrid optimal descriptors. *SAR and QSAR in Environmental Research* 28, 179–197.
- Lei, B., Ma, Y., Li, J., *et al.*, 2010. Prediction of the adsorption capability onto activated carbon of a large data set of chemicals by local lazy regression method. *Atmospheric Environment* 44, 2954–2960.
- Levatic, J., Dzeroski, S., Supek, F., Smuc, T., 2013. Semi-supervised learning for quantitative structure-activity modeling. *Informatica* 37, 173.
- Lewis, D.F., 2000. Structural characteristics of human P450s involved in drug metabolism: Qsars and lipophilicity profiles. *Toxicology* 144, 197–203.
- Li, X., Shi, Y., 2008. A survey on the Randic index. *MATCH Communications in Mathematical and in Computer Chemistry* 59, 127–156.
- Liu, P., Long, W., 2009. Current mathematical methods used in QSAR/QSPR studies. *International Journal of Molecular Sciences* 10, 1978–1998.
- Liu, S.-S., Liu, H.-L., Yin, C.-S., Wang, L.-S., 2003. VSMP: A novel variable selection and modeling method based on the prediction. *Journal of Chemical Information and Computer Sciences* 43, 964–969.
- Liu, Y., 2004. A comparative study on feature selection methods for drug discovery. *Journal of Chemical Information and Computer Sciences* 44, 1823–1828.
- Lorber, A., Wangen, L.E., Kowalski, B.R., 1987. A theoretical foundation for the PLS algorithm. *Journal of Chemometrics* 1, 19–31.
- Luo, J., Hu, J., Fu, L., Liu, C., Jin, X., 2011. Use of artificial neural network for a QSAR study on neurotrophic activities of Np-tolyl/phenylsulfonyl L-amino acid thiolester derivatives. *Procedia Engineering* 15, 5158–5163.

- Magdziarz, T., Mazur, P., Polanski, J., 2009. Receptor independent and receptor dependent CoMSA modeling with IVE-PLS: Application to CBG benchmark steroids and reductase activators. *Journal of Molecular Modeling* 15, 41–51.
- Manga, N.N., Duffy, J.C., Rowe, P.H., Cronin, M.T., 2003. A hierarchical QSAR model for urinary excretion of drugs in humans as a predictive tool for biotransformation. *Molecular Informatics* 22, 263–273.
- Mayer, J.M., Van De Waterbeemd, H., 1985. Development of quantitative structure-pharmacokinetic relationships. *Environmental Health Perspectives* 61, 295.
- Meyer, H., 1899. Zur theorie der alkoholnarkose. *Naunyn-Schmiedeberg's Archives of Pharmacology* 42, 109–118.
- Mirkhani, S.A., Gharagheizi, F., Sattari, M., 2012. A QSPR model for prediction of diffusion coefficient of non-electrolyte organic compounds in air at ambient condition. *Chemosphere* 86, 959–966.
- Modarresi, H., Dearden, J.C., Modarress, H., 2006. QSPR correlation of melting point for drug compounds based on different sources of molecular descriptors. *Journal of Chemical Information and Modeling* 46, 930–936.
- Montero-Torres, A., García-Sánchez, R.N., Marrero-Ponce, Y., *et al.*, 2006. Non-stochastic quadratic fingerprints and LDA-based QSAR models in hit and lead generation through virtual screening: Theoretical and experimental assessment of a promising method for the discovery of new antimalarial compounds. *European Journal of Medicinal Chemistry* 41, 483–493.
- Moss, G.P., Dearden, J.C., Patel, H., Cronin, M.T., 2002. Quantitative structure–permeability relationships (QSPRs) for percutaneous absorption. *Toxicology in Vitro* 16, 299–317.
- Mota, S.G., Barros, T.F., Castilho, M.S., 2009. 2D QSAR studies on a series of bifonazole derivatives with antifungal activity. *Journal of the Brazilian Chemical Society* 20, 451–459.
- Papa, E., Dearden, J., Gramatica, P., 2007. Linear QSAR regression models for the prediction of bioconcentration factors by physicochemical properties and structural theoretical molecular descriptors. *Chemosphere* 67, 351–358.
- Polanski, J., 2009. Receptor dependent multidimensional QSAR for modeling drug-receptor interactions. *Current Medicinal Chemistry* 16, 3243–3257.
- Polishchuk, P.G., Muratov, E.N., Artemenko, A.G., *et al.*, 2009. Application of random forest approach to QSAR prediction of aquatic toxicity. *Journal of Chemical Information and Modeling* 49, 2481–2488.
- Puri, S., Chicks, J.S., Welsh, W.J., 2002. Three-dimensional quantitative structure–property relationship (3d-qspr) models for prediction of thermodynamic properties of polychlorinated biphenyls (PCBs): Enthalpy of sublimation. *Journal of Chemical Information and Computer Sciences* 42, 109–116.
- Qiao, L.S., Cai, Y.-L., He, Y.-S., *et al.*, 2014. Trend of multi-scale QSAR in drug design. *Asian Journal of Chemistry* 26, 5917.
- Randic, M., 2001. The connectivity index 25 years after. *Journal of Molecular Graphics and Modelling* 20, 19–35.
- Richardson, B., 1869. Physiological research on alcohols. *The Medical Times and Gazette*. 703–706.
- Richet, M., 1893. Note sur le rapport entre la toxicité et les propriétés physiques des corps. *Compt Rend Soc Biol ((Paris))* 45, 775–776.
- Rochani, A.K., Suma, B., Kumar, S., Jays, J., Madhavan, V., 2010. QSAR, ADME AND QSTR Studies of Some Synthesized Anti-Cancer 2-Indolinone Derivatives. *International Journal of Pharma and Bio Sciences* 1, 208–218.
- Roy, K., Kar, S., 2015. How to judge predictive quality of classification and regression based QSAR models. In: Haq, Z.U., Madura, J. (Eds.), *Frontiers of Computational Chemistry*. Sharjah, UAE: Bentham Science, pp. 71–120.
- Roy, K., Kar, S., Das, R.N., 2015a. A Primer on QSAR/QSPR Modeling: Fundamental Concepts. Springer.
- Roy, K., Kar, S., Das, R.N., 2015b. Statistical methods in QSAR/QSPR. A Primer on QSAR/QSPR Modeling. 37–59.
- Roy, K., Kar, S., Das, R.N., 2015c. Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment. Academic Press.
- Roy, K., Mandal, A.S., 2008. Development of linear and nonlinear predictive QSAR models and their external validation using molecular similarity principle for anti-HIV indolyl aryl sulfones. *Journal of Enzyme Inhibition and Medicinal Chemistry* 23, 980–995.
- Roy, K., Mitra, I., 2011. On various metrics used for validation of predictive QSAR models with applications in virtual screening and focused library design. *Combinatorial Chemistry & High Throughput Screening* 14, 450–474.
- Roy, K., Roy, P., Leonard, J., 2008. On some aspects of validation of predictive QSAR models. *Chemistry Central Journal* 2, P9.
- Schüßmann, G., Ebert, R.-U., Chen, J., Wang, B., Kühne, R., 2008. External validation and prediction employing the predictive squared correlation coefficient test set activity mean vs training set activity mean. *Journal of Chemical Information and Modeling* 48, 2140–2145.
- Settles, B., 2012. Active learning: Synthesis lectures on artificial intelligence and machine learning. Long Island, NY: Morgan & Clay Pool.
- Shen, M., Béguin, C., Golbraikh, A., *et al.*, 2004. Application of predictive QSAR models to database mining: Identification and experimental validation of novel anticonvulsant compounds. *Journal of Medicinal Chemistry* 47, 2356–2364.
- Shi, W., Zhang, X., Shen, Q., 2010. Quantitative structure-activity relationships studies of CCR5 inhibitors and toxicity of aromatic compounds using gene expression programming. *European Journal of Medicinal Chemistry* 45, 49–54.
- Si, H., Yuan, S., Zhang, K., *et al.*, 2008. Quantitative structure activity relationship study on EC50 of anti-HIV drugs. *Chemometrics and Intelligent Laboratory Systems* 90, 15–24.
- Si, H.Z., Wang, T., Zhang, K.J., De Hu, Z., Fan, B.T., 2006. QSAR study of 1, 4-dihydropyridine calcium channel antagonists based on gene expression programming. *Bioorganic & Medicinal Chemistry* 14, 4834–4841.
- Simeon, S., Anuwongcharoen, N., Shoombuatong, W., *et al.*, 2016. Probing the origins of human acetylcholinesterase inhibition via QSAR modeling and molecular docking. *PeerJ* 4, e2322.
- Singh, K., Verma, N., 2014. 3-Dimensional QSAR and Molecular Docking Studies of a Series of Indole Analogues as Inhibitors of Human Non-Pancreatic Secretory Phospholipase. A2.
- Sola, D., Ferri, A., Banchero, M., Manna, L., Sicardi, S., 2008. QSPR prediction of N-boiling point and critical properties of organic compounds and comparison with a group-contribution method. *Fluid Phase Equilibria* 263, 33–42.
- Soltanpour, S., Shahbazy, M., Omidikia, N., Kompany-Zareh, M., Baharifard, M.T., 2016. A comprehensive QSPR model for dielectric constants of binary solvent mixtures. *SAR and QSAR in Environmental Research* 27, 165–181.
- Speck-Planche, A., Kleandrova, V.V., Luan, F., Cordeiro, M.N.D., 2012. Rational drug design for anti-cancer chemotherapy: Multi-target QSAR models for the in silico discovery of anti-colorectal cancer agents. *Bioorganic & Medicinal Chemistry* 20, 4848–4855.
- Suzuki, T., Ide, K., Ishida, M., Shapiro, S., 2001. Classification of environmental estrogens by physicochemical properties using principal component analysis and hierarchical cluster analysis. *Journal of Chemical Information and Computer Sciences* 41, 718–726.
- Taft Jr, R.W., 1952. Polar and steric substituent constants for aliphatic and o-Benzoate groups from rates of esterification and hydrolysis of esters1. *Journal of the American Chemical Society* 74, 3120–3128.
- Taft Jr, R.W., 1953a. The general nature of the proportionality of polar effects of substituent groups in organic chemistry. *Journal of the American Chemical Society* 75, 4231–4238.
- Taft Jr, R.W., 1953b. Linear steric energy relationships. *Journal of the American Chemical Society* 75, 4538–4539.
- Tang, H., Wang, X.S., Huang, X.-P., *et al.*, 2009. Novel inhibitors of human histone deacetylase (HDAC) identified by QSAR modeling of known inhibitors, virtual screening, and experimental validation. *Journal of Chemical Information and Modeling* 49, 461–476.
- Tomar, D., Agarwal, S., 2014. A survey on pre-processing and post-processing techniques in data mining. *International Journal of Database Theory and Application* 7, 99–128.
- Toropov, A., Kudyskin, V., Voropaeva, N., Ruban, I., Rashidova, S.S., 2004. QSPR modeling of the reactivity parameters of monomers in radical copolymerizations. *Journal of Structural Chemistry* 45, 945–950.
- Valencia, A., Prous, J., Mora, O., Sadrieh, N., Valerio, L.G., 2013. A novel QSAR model of Salmonella mutagenicity and its application in the safety assessment of drug impurities. *Toxicology and Applied Pharmacology* 273, 427–434.
- Van Gestel, C., Ma, W.-C., 1990. An approach to quantitative structure-activity relationships (QSARs) in earthworm toxicity studies. *Chemosphere* 21, 1023–1033.

- Vedani, A., Dobler, M., 2002. 5D-QSAR: The key for simulating induced fit? *Journal of Medicinal Chemistry* 45, 2139–2149.
- Veeraramy, R., Rajak, H., Jain, A., *et al.*, 2011. Validation of QSAR models-strategies and importance. *International Journal of Drug Design & Discovery* 3, 511–519.
- Veyseh, S., Hamzehali, H., Niazi, A., Ghasemi, J.B., 2015. Application of multivariate image analysis in qspr study of pKa of various acids by principal components-least squares support vector machine. *Journal of the Chilean Chemical Society* 60, 2985–2987.
- Vieira, J.B., Braga, F.S., Lobato, C.C., *et al.*, 2014. A QSAR, pharmacokinetic and toxicological study of new artemisinin compounds with anticancer activity. *Molecules* 19, 10670–10697.
- Wang, D.-D., Feng, L.-L., He, G.-Y., Chen, H.-Q., 2014. QSAR studies for the acute toxicity of nitrobenzenes to the *Tetrahymena pyriformis*. *Journal of the Serbian Chemical Society* 79, 1111–1125.
- Wang, T., Si, H., Chen, P., Zhang, K., Yao, X., 2008. QSAR models for the dermal penetration of polycyclic aromatic hydrocarbons based on Gene Expression Programming. *Molecular Informatics* 27, 913–921.
- Wesley, L., Veerapaneni, S., Desai, R., *et al.*, 2016. 3D-QSAR and SVM Prediction of BRAF-V600E and HIV integrase inhibitors: A comparative study and characterization of performance with a new expected prediction performance metric. *American Journal of Biochemistry and Biotechnology* 12, 253–262.
- Wiener, H., 1947. Structural determination of paraffin boiling points. *Journal of the American Chemical Society* 69, 17–20.
- Xue, Y., Li, Z.-R., Yap, C.W., *et al.*, 2004. Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents. *Journal of Chemical Information and Computer Sciences* 44, 1630–1638.
- Zernov, V.V., Balakin, K.V., Ivaschenko, A.A., Savchuk, N.P., Pletnev, I.V., 2003. Drug discovery using support vector machines. The case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *Journal of Chemical Information and Computer Sciences* 43, 2048–2056.
- Zhang, L., Zhu, H., Oprea, T.I., Golbraikh, A., Tropsha, A., 2008. QSAR modeling of the blood–brain barrier permeability for diverse organic compounds. *Pharmaceutical Research* 25, 1902.
- Zhang, S., Golbraikh, A., Tropsha, A., 2006. Development of quantitative structure–binding affinity relationship models based on novel geometrical chemical descriptors of the protein–ligand interfaces. *Journal of Medicinal Chemistry* 49, 2713–2724.
- Zou, C., Zhou, L., 2007. QSAR study of oxazolidinone antibacterial agents using artificial neural networks. *Molecular Simulation* 33, 517–530.
- Zou, J.-W., Huang, M., Huang, J.-X., Hu, G.-X., Jiang, Y.-J., 2016. Quantitative structure–hydrophobicity relationships of molecular fragments and beyond. *Journal of Molecular Graphics and Modelling* 64, 110–120.

Biographical Sketch



Swathik Clarancia Peter received her M.Tech. in Computational Biology from Pondicherry University, India in 2017 and B.Tech. in Bioinformatics from Tamil Nadu Agricultural University, Coimbatore, India in 2015. At present she is working as a senior research fellow in ICAR-Sugarcane Breeding Institute, Coimbatore, India. She works in the field of drug design & discovery and transcriptome data analysis.



Mannu Jayakanthan is an Assistant Professor in the Department of Plant Molecular Biology and Bioinformatics at Tamil Nadu Agricultural University, Coimbatore, India. He obtained his PhD from Pondicherry University and pursued his postdoctoral fellowship at the Centre for Cellular and Molecular Biology (CCMB) in Hyderabad, India. His research interest lies in computer-aided drug discovery.



Jaspreet Kaur received her M.Tech. in Bioinformatics from Delhi Technological University, India in 2013, and B. Tech. in Biotechnology from Amity University, India (2011). She joined Indian Institute of Technology, Delhi, India in 2014 for the Ph.D. program. She works in the field of computer aided drug designing and devising computational approaches for aiding in targeted genome editing.



Vidhi Malik received her M.Tech. in Bioinformatics from Delhi Technological University, India in 2013 and B. Tech. in Biotechnology from Sardar Vallabhbhai Patel University of Agriculture and Technology, India in 2011. She joined Indian Institute of Technology, Delhi, India in 2015 for the Ph.D. program. Her research interest lies in next generation sequence (NGS) data analysis and computer-aided drug designing.



Navaneethan Radhakrishnan received his B.Tech. in Bioinformatics from Tamil Nadu Agricultural University, India in 2016. He joined Indian Institute of Technology Delhi, India in 2016 as Junior Research Fellow in the Department of Biochemical Engineering and Biotechnology. His research interest lies in understanding the biological systems using computational approaches.



Durai Sundar is a DuPont Young Professor in the Department of Biochemical Engineering and Biotechnology at Indian Institute of Technology, Delhi. He obtained his education from Pondicherry University and Johns Hopkins University, Baltimore, United States. He is a specialist in molecular and computational biology and his current research interests are in rational design of genome editing tools and in the biological activity of natural drugs.

Pharmacophore Development

Balakumar Chandrasekaran and Nikhil Agrawal, University of KwaZulu-Natal, Durban, South Africa

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal and University of Minho, Braga, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

Discovery of new drug molecules is an ever challenging task, even until today. It is a fact that the success rate in this area is extremely low. Typically, only one out of a million compounds finally obtain approval to be used as a drug. Starting from the process of target identification, a huge investment (approximately 2 billion dollars) and a long period (ranging 8–16 years) in research and development (R&D) activities are normally required to yield a new drug. After the post-marketing surveillance (phase 4 of the clinical trials), if the drug is found to have any adverse effects, it would be withdrawn from the market with a loss which amounts to about million dollars. Hence, the entire process starting from the drug discovery to design is a risky affair and within a short period of time, the pharmaceutical companies may need to take back the revenue out of their investment. Hence, there is an enormous pressure on pharmaceutical industries to come up with new drug molecules to treat existing diseases. Moreover, the environmental changes around the globe have heavily contributed to the evolution of drug-resistant strains; therefore, there is an urgent need for alternate/new more effective drugs.

Computer-aided drug design (CADD) tools have assisted medicinal chemists to a large extent by reducing the time, cost and number of compounds to be synthesized as leads. In this computational approach, the integrated aspects of chemistry (ligands) and biology (proteins/enzymes/receptors) are studied in order to search for new drug candidates. Currently, 20% of pharmaceutical R&D expenditure is allotted to CADD segment which helps face various challenges encountered during the R&D of new drugs. Further, the computational drug design is based on two major approaches namely, structure-based drug designing (SBDD) and ligand-based drug designing (LBDD). SBDD approach uses the structural information of the target proteins like enzymes or receptors, to identify compounds that can potentially be used as a drug (Kalva *et al.*, 2016; Agrawal *et al.*, 2016; Agrawal *et al.*, 2018). On the other hand, the ligand-based approach involves the development of 3D pharmacophore models and modeling quantitative structure-activity relationship (QSAR) or quantitative structure-property relationship (QSPR) (Balakumar *et al.*, 2017; Bheemanapalli *et al.*, 2013). This involves using the physicochemical properties of the ligand molecules for drug development. The 3D pharmacophore modeling is one of the useful drug design approaches in new drug discovery process.

Definition of a Pharmacophore: Initially, the term pharmacophore was used to refer to – “chemical groups” of a molecule that are responsible for a biological effect (Güner and Bowen, 2014). However, with time its meaning has shifted to patterns of “abstract features” that are responsible for a biological effect. The basis of modern day International Union of Pure and Applied Chemistry (IUPAC) definition of a pharmacophore is a redefinition of the above statements as given by Schueler (1960). The IUPAC definition is as follows:

“A pharmacophore is an ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or to block) its biological response.”

This is the modern accepted definition of a pharmacophore, presently (Güner, 2000; Wermuth *et al.*, 1998). Pharmacophore modeling has a wide range of applications in drug design and discovery such as 3D database search, scaffold hopping, virtual screening, ligand profiling, pose filtering, fragment designing and predicting the biological activities. Most commonly observed features in pharmacophore models are hydrogen bond donors, hydrogen bond acceptors, aromatic rings, hydrophobic sites, positively ionizable groups and negatively ionizable groups.

Historical Perspective of Pharmacophore Modeling

Paul Ehrlich, the father of chemotherapy, initially conceptualized the pharmacophores and that concept is still unperturbed for the past century (Ariens, 1966; Ehrlich, 1909, 1898). In simpler words, the bioactive molecules interact with their targets via a 3D arrangement of abstract features that define interaction types rather than specific functional groups. These interaction types could be the formation of hydrogen bonds, charged interactions, metal interactions, or hydrophobic and aromatic contacts (Qing *et al.*, 2014). Using this knowledge compounds with desired biological activities have been synthesized. The earliest success of a 2D pharmacophore model was demonstrated by synthesizing *trans*-diethylstilbestrol (an estrogenic agent) based on the pharmacophore similarities with estradiol even though the latter had a non-planar conformation (Fig. 1; Dodds and Lawson, 1938). Similarly, an anti-hypertensive drug called clonidine (Fig. 2) was found to interact with the central norepinephrine receptor through 3-point attachment (an ionic bond, hydrogen bond and stacking interaction) (Andén *et al.*, 1970). Though norepinephrine and clonidine had different chemical structures, they retained common features exhibiting functional similarities. Another important discovery in this regard was the determination of critical intramolecular distances from protonated nitrogen (N^+) to the aromatic ring ($D=5.1\text{--}5.2\text{ \AA}$) and for the elevation of the positive charge to the aromatic plane ($H=1.2\text{--}1.4\text{ \AA}$) (Pullman *et al.*, 1972).

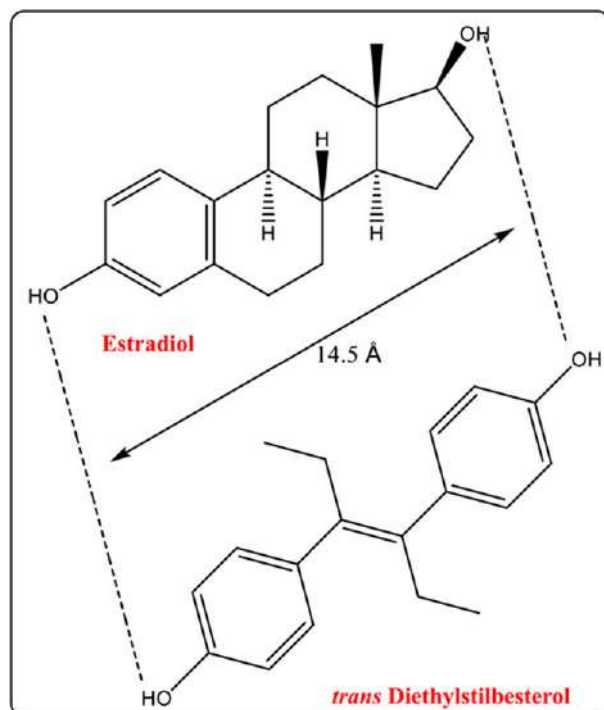


Fig. 1 Structural comparison between estradiol and *trans*-diethylstilbestrol.

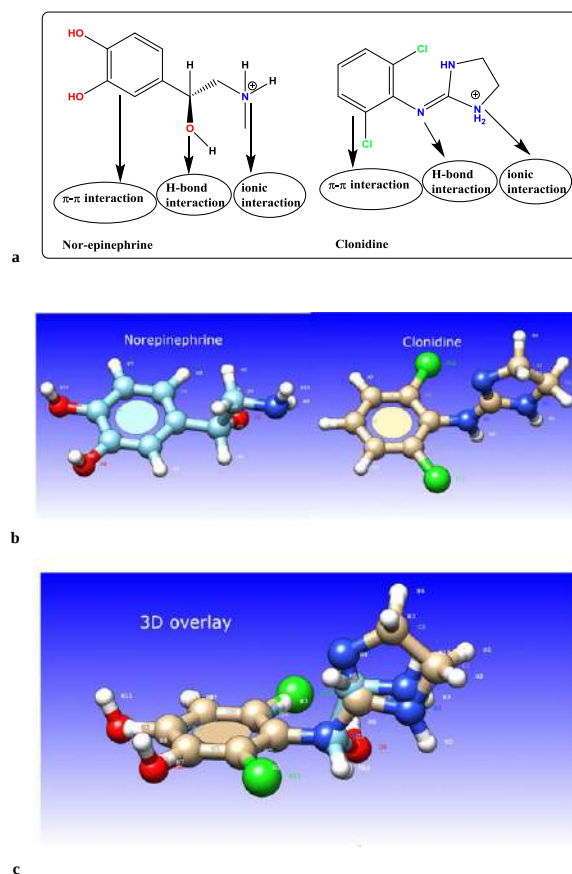


Fig. 2 (a) Clonidine presents the same type of molecular interactions as nor-epinephrine despite the difference in the orientation of the imidazolidinium ring towards the phenyl ring. (b) 3D structures of nor-epinephrine (cyan) and clonidine (grey) (c) 3D overlay of nor-epinephrine (cyan) and clonidine (grey).

Pharmacophore Modeling Approaches

Two types of pharmacophore modeling techniques are employed in drug design, namely, ligand-based pharmacophore modeling and structure-based pharmacophore modeling. This section describes the two different approaches used for pharmacophore modeling.

Ligand-Based Pharmacophore Modeling

If the target protein structure is not available, then the ligand-based pharmacophore modeling approach is employed for the design of new chemical entities (NCE). Two important steps of the ligand-based pharmacophore modeling are

1. Feature analysis in the training set molecules/ligands, and
2. Alignment of all bioactive conformations of the ligand to establish better overlay of features.

Currently, many automated software and tools (**Table 1**) for pharmacophore model generation are available for example, HipHop and HypoGen (Catalyst software packaged as a part of BIOVIA Discovery Studio Dassault Systems, 2016), DISCO (Martin *et al.*, 1993), GASP (Jones *et al.*, 1995), GALAHAD (Richmond *et al.*, 2006), PHASE (Dixon *et al.*, 2006), MOE (Molecular Operating Environment MOE, 2017) and several other free academic programs (Bandyopadhyay and Agrafiotis, 2008; Baroni *et al.*, 2007; Nettles *et al.*, 2007). Each program differs in its algorithm that handles the generation and validation of pharmacophore models. There are several challenges and hurdles that one faces during pharmacophore development. For example, bioactive conformations of the compound may be unknown, then conformational flexibility remains one of the significant hurdles in developing a pharmacophore model. Another challenge is the molecular alignment of the ligand conformations in the training set.

Pharmacophore alignment: Molecular alignment can be classified into two types namely, point-based or property-based approaches (Wolber *et al.*, 2008). Atoms or chemical feature point distances are minimized in the point-based approaches whereas the property based approaches use molecular field descriptors to generate alignments. Point-based alignment algorithms have been used by a majority of the programs. The point-based method can be further categorized based on whether points represent atoms, fragments or chemical features. Each of the points can be superimposed using a least-squares fitting method. However, these assumptions become one of the greatest limitations for the all point-based methods while aligning dissimilar ligands.

On the other hand, molecular field descriptors or Gaussian functions are used for the property-based algorithms for the alignments. The stochastic proximity embedding, atomic property fields, fuzzy pattern recognition and grid-based interaction

Table 1 Some commercial tools and web servers for pharmacophore development

Software	Brief description	Web link
GASP	• Uses a genetic algorithm for pharmacophore identification. • GASP finds similarity between functional groups in different molecules and the alignment of these groups in a common geometry. • GASP is accessible through the SYBYL molecular modeling package.	https://www.certara.com/pressreleases/certara-enhances-sybyl-x-drug-design-and-discovery-software-suite/
CATALYST	• Identifies promising new molecular entities with or without target-structured data. • CATALYST is a part of the BIOVIA Discovery Studio.	http://accelrys.com/products/collaborative-science/biovia-discovery-studio/pharmacophore-and-ligand-based-design.html
MOE pharmacophore modeling	• MOE can perform ligand, structure and fragment based pharmacophore modeling. • MOE Pharmacophore methods use a generalized molecular recognition representation and geometric constraints.	https://www.chemcomp.com/MOEPharmacophore_Discovery.htm
PHASE	• Phase uses a newly developed common pharmacophore perception algorithm. • Uses pharmacophore-based shape alignments to quickly create high-quality hypothesis from a handful to hundreds of known active ligands. • PHASE is accessible through Schrodinger molecular modeling package.	https://www.schrodinger.com/phase
PharmaGist	• PharmaGist is a freely available web server for ligand based pharmacophore identification. • PharmaGist server calculates candidate pharmacophores by multiple flexible alignments of the input ligands.	http://bioinfo3d.cs.tau.ac.il/pharma/about.html
PharmMapper	• PharmMapper freely available web-server designed to identify potential target candidates for the given probe small molecules.	http://lilab.ecust.edu.cn/pharmmapper/help.php
ZINCPharmer	• PharmMapper uses semi-rigid pharmacophore mapping protocol. • ZINCPharmer is free pharmacophore search software for screening the purchasable subset of the ZINC database. • Uses the Pharmer open source pharmacophore search technology to efficiently search a large database of fixed conformers for pharmacophore matches.	http://zincpharmer.csb.pitt.edu/

energies are the recently developed alignment methods (Bandyopadhyay and Agrafiotis, 2008; Baroni *et al.*, 2007; Nettles *et al.*, 2007; Totrov, 2008).

Appropriate selection of the training set is another crucial step in the generation of a valid pharmacophore model. When a different training set is employed for the pharmacophore generation using the same software program, it usually results in entirely different models (Hecker *et al.*, 2002; Toba *et al.*, 2006; Vadivelan *et al.*, 2007). For example, three different pharmacophore models were proposed for cyclin-dependent kinase 2 (CDK2) inhibitors bearing different features and location constraints using the same software program Catalyst (now a part of BIOVIA Discovery Studio Dassault Systems, 2016; Vadivelan *et al.*, 2007).

The ligand-based pharmacophore models can be developed in either qualitative (common feature-based hypothesis) or quantitative (3D-QSAR based models) manner. Below we describe each of the approach in brief.

Qualitative Models: The qualitative model is based on common feature pharmacophore generation from a set of bioactive ligands/training set (usually 2–16 ligands) of diversified structures. In the qualitative model, we assume similar binding modes for the training set with the same target protein and the 3D spatial arrangements of chemical features shared among the training set ligands is deduced. In this common feature-based hypothesis, explicit biological activity data is not required, partial matches are allowed with the preference of structural diversity among the ligands. Commonly used software like Hip-Hop, Phase, and MOE analyse ligands based on their chemical features and superimpose all the features to identify a specific conformation (Dassault Systems, 2016; Dixon *et al.*, 2006; Molecular Operating Environment MOE, 2017).

Quantitative Models: In case of the quantitative model, the biological activity data (*e.g.*, IC_{50} or K_i) are required while generating the predictive hypothesis to correlate chemical structure with biological activity thereby predicting the activity of novel compounds. The three important phases namely constructive, subtractive and optimization are involved in the generation of a quantitative pharmacophore model. Appropriate alignment is the key factor that decides the prediction power of a 3D quantitative model for virtual screening tasks. In this approach, a training dataset of 18–30 ligands depicting appropriate structural diversity with their pharmacological activity data are required (as represented by IC_{50} , EC_{50} , or K_i values of similar assay conditions). As the quality of a pharmacophore model and its predictability depends on the training data set, it is very crucial to have ligands with a wide range of biological activity in the training dataset. The activity could span over four orders of magnitude comprising categories of most active, moderately active, least active ligands and decoys. Commonly used software such as Hypogen, Phase-Macromodel and MOE are widely used for the generation of 3D-QSAR pharmacophore models (Dassault Systems, 2016; Dixon *et al.*, 2006; Molecular Operating Environment MOE, 2017). A 3D-conformational search is performed for each of the ligands within an energy range to yield multiple conformers using an appropriate algorithm. These algorithms consider coverage of the energy landscape and diversity among the conformational models of the ligands under study. Other algorithms implemented in commercial software programs are (Rubicon Daylight Chemical Information Systems, Inc.), (Omega, OpenEye Scientific Software) and stochastic proximity embedding (SPE) (Agrafiotis *et al.*, 2013).

Structure-Based Pharmacophore Modeling

The term structure in 'structure-based pharmacophore modeling' refers to the 3D structure of a macromolecule which is usually a receptor or an enzyme with or without a bound ligand. This structure is determined through X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy techniques or may be constructed using homology modeling (Vyas *et al.*, 2012). If the protein structure is available as a potential drug target, then structure-based pharmacophore modeling will be preferred as this approach considers interaction points within the protein binding site. This method involves key aspects such as protein preparation, identification or prediction of ligand binding site, pharmacophore feature generation and selection of crucial features contributing to the pharmacological activity. In this approach, protein structure is the crucial source of information whether obtained from protein data bank (PDB) or modeled using a suitable template by homology modeling. The second most important step is the binding site prediction which is performed using different algorithms. To distinguish the binding site from other parts of the protein surface, most of these methods have exploited one or more of the following properties namely, evolutionary, geometric, energetic, statistical, and combined or knowledge-based methods (Xie and Hwang, 2015). After a careful analysis of the binding site, the programs identify the complementary chemical features of the binding-site amino acid residues and their spatial arrangements to derive a meaningful pharmacophore model. These programs initially generate interaction maps and convert them into several pharmacophoric features. This methodology yields several chemical features which must be optimized using customized tools implemented in the respective software programs. This task can be accomplished using approaches like clustering of features, removing certain unimportant features, comparing the features which complement the docked pose of ligands, and site-directed mutagenesis data. Finally, the key features responsible for contributing the desired pharmacological response will be carefully selected. For structure-based pharmacophore modeling, software programs like BIOVIA Discovery Studio (Dassault Systems, 2016), Sybyl (SYBYL7.3 Tripos Inc, 2003), MOE (Molecular Operating Environment MOE, 2017), Chemogenomics (Bredel and Jacoby, 2004), GBPM (Ortuso *et al.*, 2006), LigandScout (Wolber and Langer, 2005), Schrodinger-Maestro (Schrodinger, 2017), Pocket v2 (Chen and Lai, 2006), and Snooker (Sanders *et al.*, 2011) are commonly employed.

If the protein exists in its ligand-bound form, the key interaction points between these two can help derive at the pharmacophore model. To construct the structure-based pharmacophore model from the complex, the ligand is initially hybridized and the bonds are characterized. A widely employed program for this purpose is LigandScout (Wolber and Langer, 2005). It uses a heuristic approach along with the template-based numeric analysis of the protein-ligand complex which may have been obtained

through PDB or the docking procedures. According to the protein-ligand binding information, a consensus model will be derived either by overlaying all the obtained models or by model refinement procedures.

Two major drawbacks associated with structure-based pharmacophore modeling are protein flexibility and conformational flexibility of the ligand. To overcome these drawbacks, two approaches can be used (1) deriving a model from the docked complex obtained through a flexible docking of ligands into the protein binding-site; (2) building and aligning the models simultaneously from the different snapshots of the protein or protein-ligand complexes of molecular dynamics (MD) simulations (Drwal and Griffith, 2013).

The energetically optimized structure-based pharmacophores (e-pharmacophores) can be used to screen millions of ligands as implemented in Schrodinger software (Therese *et al.*, 2014). These-pharmacophores remove the requirement of known active ligands while generating pharmacophore models and assist in identifying new regions of the active-site for drug design. In this protocol, the fragment docking (Glide XP Friesner *et al.*, 2004, 2006) data will be used to generate the hypotheses for each conformation using the e-pharmacophore script. This script maps the Glide XP energies for the top 2000 docked poses onto the ligands aligned to the receptor conformations followed by the clustering of the fragments. The software, Phase (Dixon *et al.*, 2006), will be used to generate the hypotheses from the docked poses with default features.

Role of Excluded Volumes in Pharmacophore Modeling

The so-derived pharmacophore model may possess the requisite features to bind to the target protein but yet fail to bind. One factor that may be responsible for this observation is the steric clashes between the ligand and the receptor. In such cases, the quality of the generated model can be improved by incorporating exclusion volume constraints. Ligand excluded surface represents an approximation of the regions accessible to the ligand under consideration. Receptor-based excluded volumes are represented as spheres centred on appropriate atoms in and around the binding site, with sizes dictated by the associated atomic van der Waals radii. If the ligand maps all the key features of the model, it is considered as active whereas if the functional groups of the ligand overlap the excluded volume regions, it will be inferred as pharmacologically inactive ligands. In some cases, this excluded volume constraint plays a crucial role in identifying potent ligands which match exclusively essential pharmacophoric features without occupying the exclusion volume (Schuster *et al.*, 2006).

A hard-sphere approximation is usually encountered while employing an exclusion volume in the model. In cases of inadequate sampling, this approximation may be deemed as too strict. Therefore, this approximation can be relaxed by customizing the excluded volume in the following ways

- (1) Decreasing the size of the spheres,
- (2) Removing spheres that are in close proximity to the bound ligand, and
- (3) Tolerating a certain amount of overlap between the ligand and the excluded volumes.

In case of ligand-based pharmacophore modeling, when the receptor structure is unavailable, assigning meaningful excluded volumes is not so easy. However, this limitation may be overcome by inferring the excluded volumes from a set of known active ligands (Shrink-Wrap Method Van Drie, 1997) or inactive ligands (Murray *et al.*, 1999). Various software packages allow either manual or automated placement of excluded volumes based on a visual inspection of aligned structures and generate spheres at the desired locations.

The quality of the generated pharmacophore models can be assessed based on a scoring and ranking to determine how well the ligands map onto the pharmacophore model. A higher ranking indicates a better reliability and quality.

Validation of Pharmacophore Models

Once a pharmacophore model has been generated, it must be validated to ascertain its reliability and quality for successive applications in different molecular modeling tasks. To test whether or not our models are good enough to predict the active compounds, we perform pharmacophore validation. For the identification of a valid model, various validation parameters are analyzed through statistical analysis, the goodness of hit list (GH), receiver operating characteristic (ROC) curve, test set prediction, cost analysis (Hypogen), and Fischer's randomization test (Hypogen). This helps in differentiating the active ligand from inactive ligands for the specified drug targets. Below we discuss these analyses in brief.

Statistical Analysis

A regression analysis of the model can be performed against the test set ligands to verify a pharmacophore model. If the pharmacophore model demonstrates the correlation of $r^2 > 0.7$ between the predicted activity data and experimentally reported data, it is considered to be a valid model. To perform this task, a test set of ligands is constructed all of which act on the same target as that of training set ligand. The test set is classified as most active, moderately active, and less active based on the available range of the activity data to understand the predictability of the generated pharmacophore model.

Güner-Henry (GH) Method

The Güner-Henry method is a commonly employed validation protocol to score the hit lists and provide a quantitative "goodness of hit list" (GH) score. While using qualitative data, our aim is to identify and retrieve the maximum possible number of

molecules that have the same activity as the target structure and to simultaneously minimize the number of inactive molecules. Güner and Henry introduced the GH score to evaluate the effectiveness of 3D database searches and can be applied to the evaluation of any sort of search for which qualitative bioactivity data are available (Willett, 2004). It helps in the retrieval of active ligands from a large pool of ligands that includes inactive and decoy ligands as well. For this analysis, a dataset is built using inactive ligands from the literature or from the database of decoys (DUD-E) tools (Mysinger *et al.*, 2012). These decoys are mostly less active/inactive ligands that resemble ligands physically but are topologically dissimilar to minimize the likelihood of actual binding at the same time (Mysinger *et al.*, 2012). The value of GH score approaching 1 indicates an ideal model (Guner *et al.*, 2004). The following Eqs. 1–4 are used for calculation of GH score.

$$\text{Percent yield of actives (\%y)} = \frac{Ha}{Ht} \times 100 \quad (1)$$

$$\text{Percent ratio of actives (\%A)} = \frac{Ha}{A} \times 100 \quad (2)$$

$$\text{Enrichment factor (EF)} = \frac{Ha/Ht}{A/D} \quad (3)$$

$$\text{Goodness of Hit – list (GH)} = \left(\frac{HA}{4HtA} \right) \times (3A + Ht) \times \left(1 - \frac{Ht - Ha}{D - A} \right) \quad (4)$$

Here, D is total number of molecules in a database, A is total number of active molecules in a database, Ht is total hits, Ha is active hits, GH is goodness of hit score.

Receiver Operating Characteristic (ROC) Curve

The ROC plot (specificity vs sensitivity) is used to determine the ability of the generated pharmacophore to distinguish between active and inactive ligands. This can be represented as an area under the curve (AUC) and other relevant parameters such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) (Triballeau *et al.*, 2005). Ideally, AUC near to 1 demonstrates the perfection of the valid model. The formula to calculate the above-mentioned parameters are mentioned in the Eqs. 5–7.

$$\text{Sensitivity (Se)} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{Specificity (Sp)} = \frac{TN}{TN + FP} \quad (6)$$

$$\text{Concordance} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Cost Analysis

The cost analysis method (as implemented in Hypogen, BIOVIA-DS program (Dassault Systems, 2016)) is used to analyse the statistical significance of the pharmacophore model in terms of fixed cost, null cost, and total cost. The fixed cost is the simplest model that fits all data while the null cost represents a high cost model with no features (Kandakatla and Ramakrishnan, 2014). The activity in the high cost model is determined as an average of the activity data from the training set compounds. For a valid pharmacophore model, ideally the total cost should be close to the fixed cost and the difference between the null and fixed cost value is large. A cost difference of 60 means, between 40 and 60 and below 40 means indicates a true correlation, 75%–90% and poor correlation (Kandakatla and Ramakrishnan, 2014). Similarly, configuration cost and error costs are also considered during the validation. The former refers to the complexity of the pharmacophore model space and the latter is represented by root-mean-square deviations (RMSDs) between the estimated and the experimental activities of the training set molecules. A valid pharmacophore model should comprise of good correlation, highest cost difference, and the lowest RMSD.

Fischer's Randomization Test

This test utilizes the Cat Scramble program of Catalyst/BIOVIA-DS (Dassault Systems, 2016) to create the random spreadsheets for the training set and reassign the activity values to each of the ligands to generate different pharmacophore models bearing same features. Models are generated at high ($\geq 95\%$) confidence level with a cost value lesser than the original pharmacophore model. This indicates that there is a 95% chance for the model to represent a true correlation which offers strong confidence in obtaining an accurate and reasonable pharmacophore model.

Applications of Pharmacophore Modeling

Pharmacophore modeling techniques are widely used in the qualitative and quantitative analysis of ligands to predict the pharmacological activity of hypothetical molecules (Böhm *et al.*, 2004; Sun *et al.* 2012). Hence, the pharmacophore modeling is a crucial and integral part of CADD approach in drug discovery projects. Once a pharmacophore model has been generated, it can be used for searching for potential ligands from a chemical database. Therefore, the pharmacophore model is used for 3D database search, scaffold hopping, virtual screening (VS), ligand profiling, pose filtering, fragment designing and predicting the pharmacological activities. Fig. 3 depicts various applications of pharmacophore modeling in drug design and discovery stages. A brief description of the applications of pharmacophore modeling is provided in the following subsections.

3D Database Search

A validated pharmacophore model can be used for searching bioactive molecules from a chemical database with a large number of chemical structures. The “hits” obtained can provide good candidates for the lead optimization processes. For the purpose of design and optimization of novel lead compounds, programs such as BUILDER and LUDI (Bohm, 1992; Roe and Kuntz, 1995) are helpful to search different scaffolds that possess the essential features of the generated pharmacophore model.

Scaffold Hopping

While determining the best candidate for drug development potential, hits are identified by high-throughput or virtual screening of corporate compound collections. The manipulation of the scaffold (or the core structure) of a hit molecule is much more complicated and associated with a considerable loss of activity (Zhao, 2007). That is why, the most vital step to the success of drug development is the selection of compounds with the optimal scaffold. For this purpose, a given pharmacophore model template is used to search for a hit among ligands that share a common pharmacological activity but have different scaffolds (Yang, 2010). This is known as scaffold hopping or lead hopping. It typically starts with a set of known active compounds and tries to discover structurally novel compounds by modifying the scaffold of the molecule (Böhm *et al.*, 2004). After accomplishment of an initial structure-activity relationship (SAR) study, a valid pharmacophore model is incorporated for scaffold hopping to deduce potential hits compounds (Sun *et al.*, 2012).



Fig. 3 Applications of pharmacophore modeling in drug design and discovery processes.

Pharmacophore-Based Virtual Screening

Pharmacophore-based virtual screening (PVS) is a subtype of ligand-based VS approach that utilizes the pharmacophoric feature perspectives in the identification of hit compounds (Balakumar *et al.*, 2017). In contrast to the structure-based VS such as docking (Bali *et al.*, 2012), the major advantage of PVS is that it is easy and less time consuming. A validated pharmacophore model serves as a template to search a large database of ligands to derive novel scaffolds bearing chemical features similar to the query pharmacophore. Handling of the conformational flexibility of database ligands and pharmacophore mapping are the two major steps in VS. The former considers multiple conformations for each ligand in the database while the latter verifies whether a query pharmacophore template is present in a given conformer or not. In case the query finds a hit, then both rigid and flexible fitting approaches will be employed. Rigid method of fitting considers the rigidity of conformations of ligands while calculating the best fit, whereas the flexible fitting permits conformational flexibility to the ligands by energy minimization, inter-feature distance and atom mapping parameters. Thus, the fit value will be assigned for each of the ligands and a higher value would represent the best mapping to the pharmacophore model and will result in the identification of highly active hit molecules.

Ligand Profiling

The structure-based pharmacophore modeling helps in the profiling of a set of ligands to predict the most likely targets for orphan bioactive ligands and anticipate adverse reactions, side effects (off-target effects) and proposing novel targets for drugs. Though, ligand-based pharmacophore modeling is also used for ligand profiling (Rognan, 2010), the application of structure-based pharmacophore modeling to profile ligands is usually preferred as it is fast, reliable and an efficient alternative to molecular docking while still representing specific ligand-protein interactions (Rella *et al.*, 2006).

Pose Filtering

The ligands that score highly as per the docking program scoring may not bind well in reality. Therefore, new methods are used where a docking program generates various ligand poses without considering the score given by the docking algorithm. Then, the poses are filtered with receptor-based pharmacophore searches. In pose filtering, the pharmacophore query can be placed into the binding pocket of the protein to filter potential ligand binding poses bearing all essential pharmacophore features to improve the probability of retrieving highly potent molecules for *in vitro* and *in vivo* experiments (Qing *et al.*, 2014).

Pharmacophore-Guided Fragment Design

It is generally easier to build up a fragment using smaller molecules than it is to reduce the size of a large one. This fragment-based approach is highly desirable because combining low molecular weight fragments offers the advantage of increased sampling of chemical space and the possibility of improved drug-like properties. Pharmacophore modeling can guide the designing of such a fragment that settles in an appropriate location in the binding site. When two or more pharmacologically active fragments are integrated, the chances of identifying the ligand pose seems to be greater. Thereby, we generate a group of fragments that may collectively yield pharmacologically active ligands. This application contributes to the lead molecule optimization stage wherein focussed hits are considered for refinement using the validated pharmacophore query (Miller and Roitberg, 2013).

Prediction of Pharmacological Activity

The 3D-QSAR based pharmacophore model has been used to predict the binding affinity of the test set ligands thereby predicting the pharmacological activity of those ligands well before an actual synthesis. Also, building pharmacophores of human ADME-related proteins (cytochrome enzymes) aids in the identification of various pharmacokinetic properties such as ADME (absorption, distribution, metabolism, excretion, and toxicity). Hence, pharmacophore modeling is very useful in the prediction of toxicity effects of drug-like molecules or even drugs (Rathee *et al.*, 2017).

Limitations of Pharmacophore Modeling Approaches

Unlike structure-based CADD approaches such as molecular docking, pharmacophore-based VS approach lacks reliable and good scoring metrics (Qing *et al.*, 2014). This significantly limits its applications in extending the scope of stand-alone predictions. Due to this missing element, even the ligands that may have all the requisite pharmacophoric features, may fail to exhibit *in vitro/in vivo* pharmacological activity. One of the reason for this observation is that pharmacophore modeling has inadequate consideration on the similarity with known potent ligands and overall compatibility with the target protein that complements the ligand. Hence, the structure of the identified hits may not complement the binding site of the target protein and may have other functional groups attached. Also, during a pharmacophore-based virtual screening from precomputed conformation databases, the database usually have only a limited number of low-energy conformations of the ligands. Therefore, successfully retrieval of active ligands may not be possible due to the missing conformations of the active ligand. Importantly, if a ligand has many rotatable bonds in its

structure, then it is very difficult to differentiate multiple rotations during the conformation generation process. As a rule of thumb, training set ligands are selected and thereby pharmacophore model is constructed. In such cases, it results in the generation of different pharmacophore models which upon VS yield different scaffolds claiming to be active hits. For example, in most of the cases, the kinase-inhibitor driven pharmacophore models result in the identification of novel scaffolds which may not be active for the specified kinase enzyme (Vancraenenbroeck *et al.*, 2014). Also, the success of pharmacophore modeling may be limited because the project is dealing with new compounds or targets that lack sufficient molecular information (Scior *et al.*, 2012).

Another challenging aspect of the pharmacophore-based virtual screening is the yield of a higher 'false positive' rate. This means that only a small percentage of the virtual hits are really bioactive (Yang, 2010). In simple words, the virtually identified hit ligands may not be pharmacologically 'active'. There are many factors associated with this limitation such as deficiency of required hypothesis, quality, the composition of the model, and the realistic behaviour of the ligand under biological environment. To overcome this major limitation of 'false positive' models, the following factors need to be considered (1) blending expert knowledge, (2) incorporating crucial target information, (3) rigorous validation, and (4) integrating other *in silico* approaches that offer a synergistic contribution to the success of the pharmacophore modeling.

Illustrative Examples

A couple of successful application of pharmacophore modeling have been discussed in this section to illustrate their potential significance in drug design and discovery of novel ligands.

Recently, a virtual screening of National Cancer Institute (NCI) database against the IdeR (a transcription factor of *Mycobacterium tuberculosis*) DNA binding domain was performed (Rohilla *et al.*, 2017). It was followed by *in vitro* inhibition studies using Electrophoretic Mobility Shift Assay (EMSA) and 9 lead compounds were identified and subjected to the structure-based similarity search to obtain four potent compounds. Further, lead optimization through the development of energy based pharmacophore and VS of ZINC database followed by molecular docking studies. These efforts yield one highly potent IdeR inhibitor. Authors developed five point energy based pharmacophore model using e-pharmacophore script under Schrodinger Package (Schrödinger, 2017). An energy based pharmacophore extracted and weighed interactions made by an inhibitor with the protein by using the descriptor information generated by the docking. A map of critical features contributing to the interaction between inhibitor and protein was generated using Glide module (Friesner *et al.*, 2004). LigPrep module of Schrodinger package was employed to generate energy minimized conformers and stereoisomers. All the resulted conformers were docked at the active site of the protein (PDB ID:1U8R (Wisedchaisri *et al.*, 2004)) by using Glide XP-precision mode. Based on the obtained XP descriptor file, the pharmacophore hypothesis was generated and was submitted to the e-pharmacophore script to yield a pharmacophore model. Thus, the created model consists of 5 features such as two hydrogen bond acceptors, two rings and one negative ionizable group. Based on the proximity with the key residues involved in DNA binding, each pharmacophoric points were carefully selected by these researchers. Further, screening of ZINC drug-like database (~13 million compounds) was carried out by using Phase (Dixon *et al.*, 2006). This resulted in ~110,000 molecules with a filter of 4 out of 5 molecular features to be matched. In this study, a five-point pharmacophore model provided valuable insight into the pharmacophoric features essential for IdeR inhibition. Also, the study led to the identification of one hit molecule exhibiting potent inhibition of IdeR.

Recently, we identified a hit molecule inhibiting the human kinesin spindle protein (KSP) through ligand- and structure-based *in silico* studies (Balakumar *et al.*, 2017). We have developed ligand-based pharmacophore models based on the available KSP inhibitors using BIOVIA-Discovery Studio software package (Dassault Systems, 2016). All the generated pharmacophore models were validated against *in house* test set ligands (54 compounds). Based on the validation parameters such as enrichment factor (E) and goodness of hit score, Hypo1 was selected as a valid model that consists of one ring aromatic (R), one positively ionizable (P), one hydrophobic (H), and two hydrogen bond acceptors (A₁ and A₂). The validated model (Hypo1) was then taken for database screening (Maybridge and ChemBridge) to yield hits, which were further filtered for their drug-likeness. To identify the ligand binding landscape, the potential hits retrieved from virtual database screening were docked to the active-site of the KSP (PDB: 4BXN Talapatra *et al.*, 2013) using CDocker (Wu *et al.*, 2003). The top-ranked hits obtained from molecular docking were used in molecular dynamics simulations to deduce the ligand binding affinity. This study identified MB-41570 as an inhibitor of KSP.

Conclusion and Future Perspectives

Unequivocally, the pharmacophore modeling approaches still serve as one of the most widely used and successful tools in CADD projects and medicinal chemistry. Though a considerable development in the pharmacophore modeling has happened, further improvements in pharmacophore modeling techniques are required. To enhance its success rate in real-time experiments, several aspects such as developing accurate, optimized/predictive models, better handling of ligand flexibility and efficient algorithms for ligand alignments must be considered. There could be a positive contribution in the form of the target binding-site information that can assist in developing the structure-based models more efficiently. Better algorithms can be developed in the area of fragment-based drug design involving pharmacophore fingerprint-based similarity search and generation of 3D pharmacophore queries. Another potential scope of pharmacophore modeling is the development of small molecule inhibitors of protein-protein interactions (PPIs), wherein the pharmacophoric queries will be generated by encoding the crucial interactions at the PPI interface. Hence, there are multiple new horizons to be explored through the application of pharmacophore modeling techniques in designing therapeutics.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Chemoinformatics: From Chemical Art to Chemistry in Silico. Drug Repurposing and Multi-Target Therapies. Fingerprints and Pharmacophores. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Population Analysis of Pharmacogenetic Polymorphisms. Protein Structural Bioinformatics: An Overview. Rational Structure-Based Drug Design. Small Molecule Drug Design. Structure-Based Drug Design Workflow

References

- Agrafiotis, D.K., Bandyopadhyay, D., Yang, E., 2013. Stochastic proximity embedding: A simple, fast and scalable algorithm for solving the distance geometry problem. In: Mucherino, A., *et al.* (Eds.), *Distance Geometry: Theory, Methods, and Applications*. New York, NY: Springer New York, pp. 291–311.
- Agrawal, N., Skelton, A.A., 2016. 12-Crown-4 Ether Disrupts the Patient Brain-Derived Amyloid- β -Fibril Trimer: Insight from All-Atom Molecular Dynamics Simulations. *ACS chemical neuroscience* 7, 1433–1441.
- Agrawal, N., Skelton, A.A., 2018. Binding of 12-crown-4 with Alzheimer's A β 40 and A β 42 monomers and its effect on their conformation: insight from molecular dynamics simulations. *Molecular Pharmaceutics* 15, 289–299.
- Andén, N.E., *et al.*, 1970. Evidence for a central noradrenaline receptor stimulation by clonidine. *Life Sciences* 9 (9), 513–523.
- Ariens, E.J., 1966. Molecular pharmacology, a basis for drug design. *Fortschr Arzneimittelforsch* 10, 429–529.
- Balakumar, C., *et al.*, 2017. Ligand- and structure-based in silico studies to identify kinesin spindle protein (KSP) inhibitors as potential anticancer agents. *Journal of Biomolecular Structure and Dynamics*. 1–18.
- Bali, A., Dhillon, S.K., Balakumar, C., 2012. Alkoxyphenyl methanesulfonamides: Synthesis, anti-inflammatory effect, and docking studies. *Medicinal Chemistry Research* 21 (10), 3053–3062.
- Bandyopadhyay, D., Agrafiotis, D.K., 2008. A self-organizing algorithm for molecular alignment and pharmacophore development. *Journal of Computational Chemistry* 29 (6), 965–982.
- Baroni, M., *et al.*, 2007. A common reference framework for analyzing/comparing proteins and ligands. Fingerprints for Ligands And Proteins (FLAP): Theory and application. *Journal of Chemical Information and Modeling* 47 (2), 279–294.
- Bheemanapalli, L.N., Balakumar, C., Kaki, V.R., Kaur, R., Akkinepally, R.R., 2013. Pharmacophore based 3D-QSAR study of biphenyl derivatives as nonsteroidal aromatase inhibitors in JEG-3 cell lines. *Medicinal Chemistry* 9, 974–984.
- BIOVIA, Discovery Studio Modeling Environment. San Diego: Dassault Systèmes.
- Bohm, H.J., 1992. The computer program LUDI: A new method for the de novo design of enzyme inhibitors. *Journal of Computer-Aided Molecular Design* 6 (1), 61–78.
- Böhm, H.-J., Flohr, A., Stahl, M., 2004. Scaffold hopping. *Drug Discovery Today: Technologies* 1 (3), 217–224.
- Bredel, M., Jacoby, E., 2004. Chemogenomics: An emerging strategy for rapid target and drug discovery. *Nature Reviews Genetics* 5 (4), 262–275.
- Chen, J., Lai, L., 2006. Pocket v.2: Further developments on receptor-based pharmacophore modeling. *Journal of Chemical Information and Modeling* 46 (6), 2684–2691.
- Dixon, S.L., *et al.*, 2006. PHASE: A new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. *Journal of Computer-Aided Molecular Design* 20 (10–11), 647–671.
- Dodds, E.C., Lawson, W., 1938. Molecular structure in relation to oestrogenic activity. Compounds without a phenanthrene nucleus. *Proceedings of the Royal Society of London. Series B, Biological Sciences* 125 (839), 222–232.
- Drwal, M.N., Griffith, R., 2013. Combination of ligand- and structure-based methods in virtual screening. *Drug Discovery Today Technology* 10 (3), e395–e401.
- Ehrlich, P., 1898. Ueber die Constitution des Diphtheriegiftes. *Dtsch med Wochenschr* 24 (38), 597–600.
- Ehrlich, P., 1909. Über den jetzigen stand der Chemotherapie. *Berichte Der Deutschen Chemischen Gesellschaft* 42 (1), 17–47.
- Friesner, R.A., *et al.*, 2004. Glide: A new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of Medicinal Chemistry* 47.
- Friesner, R.A., *et al.*, 2006. Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein — Ligand complexes. *Journal of Medicinal Chemistry* 49 (21), 6177–6196.
- Güner, O.F., 2000. Pharmacophore Perception, Development and Use in Drug Design. Dr. Igor Tsigelny.
- Güner, O.F., Bowen, J.P., 2014. Setting the record straight: The origin of the pharmacophore concept. *Journal of Chemical Information and Modeling* 54 (5), 1269–1283.
- Guner, O., Clement, O., Kurogi, Y., 2004. Pharmacophore modeling and three dimensional database searching for drug design using catalyst: Recent advances. *Current Medicinal Chemistry* 11 (22), 2991–3005.
- Hecker, E.A., *et al.*, 2002. Use of catalyst pharmacophore models for screening of large combinatorial libraries. *Journal of Chemical Information and Computer Sciences* 42 (5), 1204–1211.
- Jones, G., Willett, P., Glen, R.C., 1995. A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *Journal of Computer-Aided Molecular Design* 9 (6), 532–549.
- Kalva, S., Agrawal, N., Skelton, A.A., Saleena, L.M., 2016. Identification of novel selective MMP-9 inhibitors as potential anti-metastatic lead using structure-based hierarchical virtual screening and molecular dynamics simulation. *Molecular BioSystems* 12, 2519–2531.
- Kandakatla, N., Ramakrishnan, G., 2014. Ligand based pharmacophore modeling and virtual screening Studies to design novel HDAC2 inhibitors. *Advances in Bioinformatics* 2014, 812148.
- Martin, Y.C., *et al.*, 1993. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *Journal of Computer-Aided Molecular Design* 7 (1), 83–102.
- Miller 3rd, B.R., Roitberg, A.E., 2013. Design of e-pharmacophore models using compound fragments for the trans-sialidase of *Trypanosoma cruzi*: Screening for novel inhibitor scaffolds. *Journal of Molecular Graphics and Modelling* 45, 84–97.
- Molecular Operating Environment (MOE), 2017. 2013.08; Chemical Computing Group ULC, 1010 Sherbooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7.
- Murray, C.W., Baxter, C.A., Frenkel, A.D., 1999. The sensitivity of the results of molecular docking to induced fit effects: Application to thrombin, thermolysin and neuraminidase. *Journal of Computer-Aided Molecular Design* 13 (6), 547–562.
- Mysinger, M.M., *et al.*, 2012. Directory of useful decoys, enhanced (DUD-E): Better ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry* 55 (14), 6582–6594.
- Nettles, J.H., *et al.*, 2007. Flexible 3D pharmacophores as descriptors of dynamic biological space. *Journal of Molecular Graphics and Modelling* 26 (3), 622–633.
- Omega; OpenEye Scientific Software, 3600 Cerrillos Rd., Suite 1107, Santa Fe, NM 87507, USA.
- Ortuso, F., Langer, T., Alcaro, S., 2006. GBPM: GRID-based pharmacophore model: Concept and application studies to protein-protein recognition. *Bioinformatics* 22 (12), 1449–1455.
- Pullman, B., *et al.*, 1972. Quantum mechanical study of the conformational properties of phenethylamines of biochemical and medicinal interest. *Journal of Medicinal Chemistry* 15 (1), 17–23.
- Qing, X., L. X., De Raeymaecker, J., *et al.*, 2014. Pharmacophore modeling: Advances, limitations, and current utility in drug discovery. *Journal of Receptor, Ligand and Channel Research* 7, 81–92.

- Rathee, D., Lather, V., Dureja, H., 2017. Pharmacophore modeling and 3D QSAR studies for prediction of matrix metalloproteinases inhibitory activity of hydroxamate derivatives. *Biotechnology Research and Innovation* 1 (1), 112–122.
- Rella, M., *et al.*, 2006. Structure-Based Pharmacophore Design and Virtual Screening for Novel Angiotensin Converting Enzyme 2 Inhibitors. *Journal of Chemical Information and Modeling* 46 (2), 708–716.
- Richmond, N.J., *et al.*, 2006. GALAHAD: 1. pharmacophore identification by hypermolecular alignment of ligands in 3D. *Journal of Computer-Aided Molecular Design* 20 (9), 567–587.
- Roe, D.C., Kuntz, I.D., 1995. BUILDER v.2: Improving the chemistry of a de novo design strategy. *Journal of Computer-Aided Molecular Design* 9 (3), 269–282.
- Rognan, D., 2010. Structure-based approaches to target fishing and ligand profiling. *Molecular Informatics* 29 (3), 176–187.
- Rohilla, A., Khare, G., Tyagi, A.K., 2017. Virtual screening, pharmacophore development and structure based similarity search to identify inhibitors against IdeR, a transcription factor of *Mycobacterium tuberculosis*. *Scientific Reports* 7 (1), 4653.
- Rubicon; Daylight Chemical Information Systems, Inc., 120 Vantis, Suite 550, Aliso Viejo, CA 92656, USA.
- Sanders, M.P.A., *et al.*, 2011. Snooker: A structure-based pharmacophore generation tool applied to class A GPCRs. *Journal of Chemical Information and Modeling* 51 (9), 2277–2292.
- Schrödinger Release 2017-4: Maestro. New York, NY: Schrödinger, LLC.
- Schueler, F.W., 1960. Chemobiodynamics and drug design. *Journal of Pharmaceutical Sciences* 50, 92.
- Schuster, D., *et al.*, 2006. Pharmacophore modeling and in silico screening for new P450 19 (aromatase) inhibitors. *Journal of Chemical Information and Modeling* 46 (3), 1301–1311.
- Scior, T., *et al.*, 2012. Recognizing pitfalls in virtual screening: A critical review. *Journal of Chemical Information and Modeling* 52 (4), 867–881.
- Sun, H., Tawa, G., Wallqvist, A., 2012. Classification of scaffold-hopping approaches. *Drug Discovery Today* 17 (7), 310–324.
- SYBYL7.3 Tripos Inc., 2003. St. Louis, MO, USA; Available online: <http://www.tripos.com>.
- Talapatra, S.K., *et al.*, 2013. Mitotic kinesin Eg5 overcomes inhibition to the phase I/II clinical candidate SB743921 by an allosteric resistance mechanism. *Journal of Medicinal Chemistry* 56 (16), 6317–6329.
- Therese, P.J., *et al.*, 2014. Multiple e-Pharmacophore modeling, 3D-QSAR, and high-throughput virtual screening of hepatitis C virus NS5B polymerase inhibitors. *Journal of Chemical Information and Modeling* 54 (2), 539–552.
- Toba, S., *et al.*, 2006. Using pharmacophore models to gain insight into structural binding and virtual screening: An application study with CDK2 and human DHFR. *Journal of Chemical Information and Modeling* 46 (2), 728–735.
- Totrov, M., 2008. Atomic property fields: Generalized 3D pharmacophoric potential for automated ligand superposition, pharmacophore elucidation and 3D QSAR. *Chemical Biology & Drug Design* 71 (1), 15–27.
- Triballeau, N., *et al.*, 2005. Virtual screening workflow development guided by the "Receiver Operating Characteristic" curve approach. Application to high-throughput docking on metabotropic glutamate receptor subtype 4. *Journal of Medicinal Chemistry* 48 (7), 2534–2547.
- Vadivelan, S., *et al.*, 2007. Virtual screening studies to design potent CDK2-cyclin A inhibitors. *Journal of Chemical Information and Modeling* 47 (4), 1526–1535.
- Vancraenenbroeck, R., *et al.*, 2014. In silico, in vitro and cellular analysis with a kinome-wide inhibitor panel correlates cellular LRRK2 dephosphorylation to inhibitor activity on LRRK2. *Frontiers in Molecular Neuroscience* 7, 51.
- Van Drie, J.H., 1997. "Shrink-Wrap" Surfaces: A new method for incorporating shape into pharmacophoric 3D database searching. *Journal of Chemical Information and Computer Sciences* 37 (1), 38–42.
- Vyas, V.K., *et al.*, 2012. Homology modeling a fast tool for drug discovery: Current perspectives. *Indian Journal of Pharmaceutical Sciences* 74 (1), 1–17.
- Wermuth, C.G., *et al.*, 1998. Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998). *Pure and Applied Chemistry*, 1129.
- Willett, P., 2004. Evaluation of molecular similarity and molecular diversity methods using biological activity data. In: Bajorath, J. (Ed.), *Cheminformatics: Concepts, Methods, and Tools for Drug Discovery*. Humana Press.
- Wisedchaisri, G., Holmes, R.K., Hol, W.G.J., 2004. Crystal structure of an IdeR–DNA complex reveals a conformational change in activated IdeR for base-specific interactions. *Journal of Molecular Biology* 342 (4), 1155–1169.
- Wolber, G., Langer, T., 2005. LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters. *Journal of Chemical Information and Modeling* 45 (1), 160–169.
- Wolber, G., *et al.*, 2008. Molecule-pharmacophore superpositioning and pattern matching in computational drug design. *Drug Discovery Today* 13 (1–2), 23–29.
- Wu, G., *et al.*, 2003. Detailed analysis of grid-based molecular docking: A case study of CDOCKER-A CHARMM-based MD docking algorithm. *Journal of Computational Chemistry* 24 (13), 1549–1562.
- Xie, Z.-R., Hwang, M.-J., 2015. Methods for predicting protein–ligand binding sites. In: Kukol, A. (Ed.), *Molecular Modeling of Proteins*. New York, NY: Springer New York, pp. 383–398.
- Yang, S.-Y., 2010. Pharmacophore modeling and applications in drug discovery: Challenges and recent advances. *Drug Discovery Today* 15 (11), 444–450.
- Zhao, H., 2007. Scaffold selection and scaffold hopping in lead generation: A medicinal chemistry perspective. *Drug Discovery Today* 12 (3), 149–155.

Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer

Muhammad Yusuf, Ari Hardianto, Muchtaridi Muchtaridi, and Rina F Nuwarda, Universitas Padjadjaran, Jatinangor, Indonesia
Toto Subroto, Universitas Padjadjaran, Bandung, West Java, Indonesia

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the drug discovery process, finding new leads is an expensive and time-consuming process. The development of the combinatorial chemistry and high-throughput screening (HTS) technologies in the early 1990s has allowed enormous libraries of compounds to be synthesized and screened in a short period of time. However, many of the identified have failed in the lead optimization process due to their poor pharmacokinetic properties. Therefore, it was necessary to develop alternative strategies that might help to select appropriate series of compounds and eliminate unsuitable structures and thus could save a significant amount of time and resources. For this purpose, computational techniques such as virtual screening (VS) can be applied as an important step in the early stage of the drug discovery process (Lavecchia and Di Giovanni, 2013). While HTS aims to experimentally test a large number of compounds in the most efficient manner, VS attempts to rationalize those compounds to reduce the number for experimental testing as much as possible (Ripphausen, *et al.*, 2011). Furthermore, the compounds evaluated do not necessarily exist, or in other words, any molecule of compound theoretically can be evaluated by VS (Lavecchia and Di Giovanni, 2013). In the VS process, a large library of compounds is filtered and reduced by a computational algorithm based on their predictive binding mode with the target, which will then be tested experimentally (Tanrikulu, *et al.*, 2013). Fig. 1 shows the differences between DBVS and HTS. Unlike HTS, DBVS allows us to test only compounds with high scoring, hence lowering the cost of hits discovery (Shoichet *et al.*, 2002; Villoutreix *et al.*, 2000). VS can be classified into two categories, namely ligand-based virtual screening (LBVS) and structure-based virtual screening (SBVS). LBVS involves the utilization of structural data from a series of known active compounds to find candidates for experimental testing. It includes several methods such as similarity searching, quantitative structure relationships (QSAR), and pharmacophore fitting. Whereas SBVS, or docking-based virtual screening (DBVS), employs the three-dimensional (3D) structure of the target macromolecule that is obtained experimentally either through X-ray crystallography or NMR or computationally through homology modeling. The candidate molecules were docked to the target macromolecule to investigate the binding mode and rank them based on their predicted binding affinity (Lavecchia and Di Giovanni, 2013).

Both LBVS and SBVS approaches are powerful technologies that promise to accelerate the pace and reduce the cost of discovering new drugs, which can be applied to VS for lead identification and optimization (Villoutreix *et al.*, 2000; Lengauer *et al.*, 2004; Vyas *et al.*, 2008; Xu and Agrafiotis, 2002).

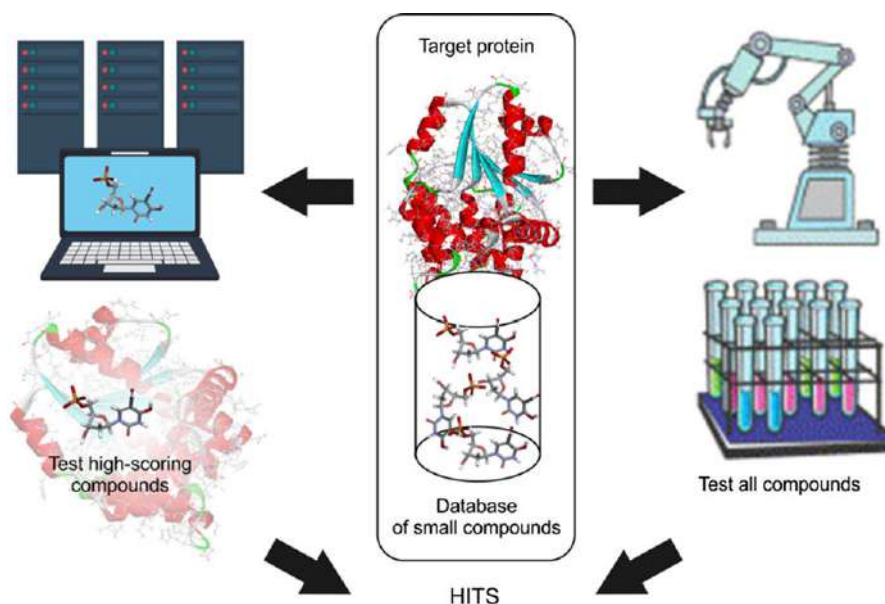


Fig. 1 DBVS selects hits compound based on the scoring function, while all compounds in HTS are being tested.

Molecular docking is a computational tool of structure-based drug design to predict protein–ligand interaction geometries and binding affinities (Cosconati *et al.*, 2010; Vyas *et al.*, 2008). In DBVS, compounds in the database are scored and ranked based on their steric and electrostatic interactions with the target site and the best compounds are tested further with biochemical assays (Singh *et al.*, 2006). The search algorithm should generate an optimum number of conformations for the ligands and of poses that should include the experimentally determined binding mode. The scoring functions commonly used in docking are force-field based, empirical-based, and knowledge-based. For this reason, there are many considerations in choosing the best compound based on the ranking in DBVS result. Several scoring functions have been developed during the past few years (Breda *et al.*, 2008; Friesner *et al.*, 2004; Jain, 2006; Kastiris and Bonvin, 2010; Kroemer, 2003; Shoichet *et al.*, 2002). There are numerous programs available for DBVS, such as Autodock (Morris *et al.*, 1996), GOLD (Jones *et al.*, 1997), DOCK (Ewing *et al.*, 2001), FlexX (Rarey *et al.*, 1996), Glide (Halgren *et al.*, 2004), LigPlot (Wallace *et al.*, 1995), and FRED (McGann, 2011). Among these programs, Autodock is the most used program for docking in scientific journals and freely available for academic work. There are some important factors to be considered in choosing docking software, including the capability for iterative refinement of docking parameter/protocol based on new results, the adaptability to additional scoring functions, pre- and/or post-docking filters, design components and results of validation studies, user learning curve, customer supports, cost, speed, user interface, input/output structural file formats, code availability, and upgrading possibility (Ghosh *et al.*, 2006).

Databases for Virtual Screening

Databases of small molecules are required for all VS projects. There are millions of compounds from structural databases of more than 32 providers gathered and stored in a single chemical database. By using the “Lipinski rule of five,” around 2.1 million of compounds are categorized as “drug-like,” and more than 1 million are lead like (Monge *et al.*, 2006).

However, the drug-likeness of chemicals from the database should be calculated, to avoid a clinical candidate that is too big, resulting in bad solubility and/or permeability (Verheij, 2006). When selecting compounds for screening, traditional “druglike” property cut-off values (e.g., Lipinski’s “rule of five”: MW < 500, LogP < 5, hydrogen acceptor < 10, and hydrogen donor < 5) (Lipinski *et al.*, 2001; Verheij, 2006) might cause problems later during the lead development stage (Verheij, 2006).

Recently, many collections of chemical library databases are available on the web. Some of the commonly used small molecule databases in VS are ZINC (free database of 3.3 million commercially available compounds), the Available Chemicals Directory (ACD; 4 million entries, not free), National Cancer Institute compound database (NCI; free database of 400,000 entries), and MDDR (MDL; Drug Data Report > 147,000 entries). Besides, a chemical library containing synthetic single compounds, combinatorial compounds, and natural product compounds (Hong, 2011; Rollinger *et al.*, 2006), and marketed drugs library (Hong, 2011; Lopez-Perez *et al.*, 2007; Quinn *et al.*, 2008) are also available. Lopez-Perez and colleagues developed NAPROC13 (Lopez-Perez *et al.*, 2007) containing ¹³C spectral information of over 6000 natural compounds. This database allows us to identify the known compounds present in the crude extracts and provides insight into the structural elucidation of unknown compounds. Moreover, Dunkel *et al.* (2006) developed a SuperNatural database that contains 3D structures and conformers of 45,917 natural compounds and about 300 natural compounds that are similar to active ingredients of drugs. It is known that 8% (3600) of this database is identical to the marketed drugs.

Another database of natural compounds is NADI (Natural Drug Discovery), which was developed by Wahab *et al.* (2009). It is a database of Malaysian medicinal plants that aimed to be a one-stop center for in silico drug discovery from natural products. It provides structural information of around 3000 different compounds, complete with the botanical sources of plants species that could be used in VS. Compounds in the NADI database can be searched based on the name, plants, structure formula, molecular weight, substructure, and SMILES code. In addition, around 40 monographs of Malaysian herbal medicine were available. They contain an integrated information regarding the description of plants, bioactivity, efficacy and usefulness, diseases target, and references. NADI is also enriched with the target protein database, composed of 645 validated drug targets useful for rapid VS. This database was successfully used to obtain lead compounds for neuraminidase inhibitors. Ikram *et al.* (2015) screened 3000 compounds from the NADI database using AutoDock against neuraminidase, a drug target of the influenza virus. The top 100 compounds were then categorized into five plants. Interestingly, some fractions and pure compounds from the five plants showed good inhibitory activity against the neuraminidase of the influenza virus (Muchtaridi *et al.*, 2014).

Case Study

In this case study, we will do a DBVS of compounds from National Cancer Institute (NCI) database to find the potential hits as *Mycobacterium tuberculosis* (Mtb) thymidylate kinase (TMPK) inhibitor. TMPK is one of the validated drug targets for Mtb. NCI provides a diversity set consisting of about 2000 compounds representing a 140,000-chemical space. The NCI diversity set is a good choice of database for virtual screening exercise because NCI also provides real compounds for further experimental testing. The compounds, when available, can be requested for free (see “Relevant Websites section”). The procedure to obtain the vialled samples is explained in this web link provided in: “Relevant Websites section”. A crystal structure of Mtb TMPK in complex with its

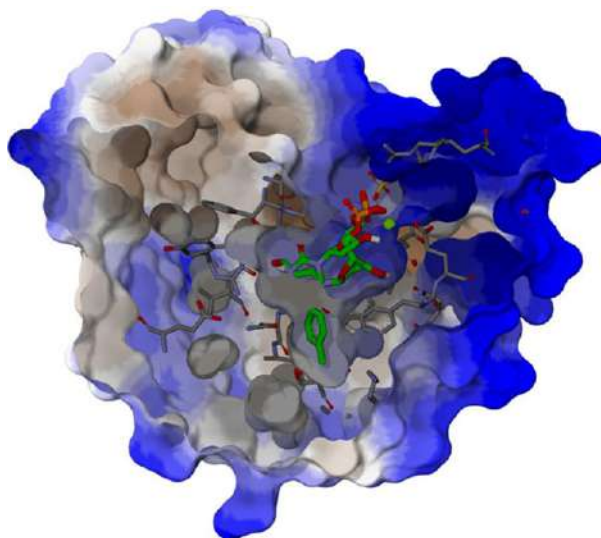


Fig. 2 Mtb TMPK in complex with TMP analog. Protein surface represents an increasing level of hydrophobicity (blue → white → brown colors). The ligand is visualized in green sticks. A green sphere, a magnesium ion, is located near the ligand. This figure is generated from PDB ID 1MRS using Biovia Discovery Studio Visualizer.

inhibitor with PDB ID 1MRS is used in this case study (**Fig. 2**). The ligand is thymidylate monophosphate (TMP) analog, possessed a nanomolar inhibition towards TMPK (Haouz *et al.*, 2003).

Programs that used in this case study are AutoDock 4.2, AutoDockTools (ADT), Raccoon, and Biovia Discovery Studio Visualizer (BDSV). ADT is a GUI of AutoDock which can be used to set up input files and docking parameters. AutoDock 4.2 and ADT is available in Windows, Linux, and Mac OS. They can be downloaded from websites provided in “Relevant Websites section” and “Relevant Websites section,” respectively. BDSV is also a freeware that can be obtained through website provided in “Relevant Websites section.” Moreover, automated DBVS of thousands of ligands will be executed by a bash script. In this case study, we choose one of the most popular and user-friendly Linux OS, that is, Ubuntu 16.04. Linux is a native environment of most of the computational chemistry program. Therefore, to have experience in operating DBVS under Linux should be beneficial to the beginner.

In general, DBVS using AutoDock consists of eight main steps: Ligand processing, receptor preparation, grid parameter setup, affinity maps calculation, protocol validation, docking parameter setup, virtual screening execution, and lastly screening result analysis.

Preparation of Ligand From Database

The database of ligands is usually compiled in a single “MOL2” file, not the individual file. For this reason, a tool called Raccoon (see “Relevant Websites section”) can be used to prepare the PDBQT format for each of ligand, a native filetype of AutoDock. The Graphical User Interface of Raccoon is presented in **Fig. 3**.

If all the ligands are wrapped in a single MOL2 file, then we must click ‘Utilities → Split a MOL2 → Open’. All the spliced MOL2 files will be generated accordingly in the same directory with its parent file. Furthermore, an automatic process of converting 2000 ligands from MOL2 format into PDBQT is accomplished by using ‘[+] Add ligands’ button. It is noted that not all ligands will be successfully converted into the PDBQT format due to some reasons, for example, a number of torsions, atom-type parameter. Therefore, in this case study, a hassle-free PDBQT-formatted NCI Diversity Set II, namely “NCI-Diversity-AutoDock-0.2.tar.gz,” can be directly downloaded from website provided in “Relevant Websites section.” This file consists of 1541 ligands, a few hundred less than the original database. All ligands with AutoDock-unsupported atom type (such as selenium) and a large number of active torsions have been discarded from the list. For the sake of accuracy, AutoDock has a default maximum rotatable bonds of 32.

Preparation of Receptor and Control Ligand

- (1) The receptor preparation will be done by ADT, which can be opened by typing ‘adt’ in the Linux terminal. The interface of ADT is presented in **Fig. 4**.
- (2) A working directory of ‘/home/USER/vs’ in ‘File → Preferences → Set → Startup Directory’ should be determined. This step will ensure all the modified files will be placed in one working directory.

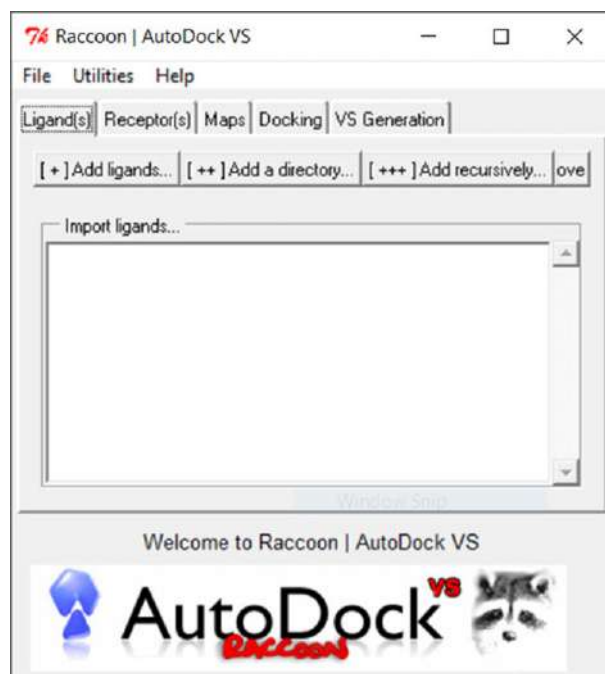


Fig. 3 The interface of Raccoon.

- (3) The PDB file of TMPK complex can be directly downloaded using ADT by clicking 'File → Import → Fetch from Web → 1MRS'.
- (4) AutoDock uses implicit water solvent model. Therefore, in most cases, all of the crystal water should be removed by using 'Edit → Delete Water'.
- (5) Most of the crystal structures in PDB lack hydrogens. All the hydrogen atoms should be added to the structure by using 'Edit → Hydrogens → Add → All Hydrogens → noBondOrder'. Furthermore, the hydrogenated PDB is saved using 'File → Save → Write PDB → 1MRS_H.pdb'.
- (6) The ligand and receptor are separated into an individual file using Linux terminal. In the terminal, go to the working directory by typing:


```
$ cd ~/vs/source
```

 In PDB 1MRS, the residue name for the ligand is "5HU". Therefore, the ligand can be extracted from the complex using grep command:


```
$ grep "5HU" 1MRS_H.pdb > 1MRS_ligand.pdb
```

 Whereas the receptor can be extracted by typing:


```
$ grep "ATOM" 1MRS_H.pdb > 1MRS_receptor.pdb
```

 As mentioned before (Fig. 2), a magnesium cofactor is located near the ligand, and hence might have an important role in the active site. This ion, with residue name "Mg," can be added to the receptor file by typing:


```
$ grep "Mg" 1MRS_H.pdb >> 1MRS_receptor.pdb
```

please note the use of ">>" instead of ">"
">>" means to add a new line(s) to the current file, while ">" means to replace the whole text with the new one. In this step, we want to add Mg as part of the receptor.
- (7) Finally, the receptor and control ligand is required to be converted into PDBQT format using ADT. Here are a few steps to be done:
 - Clean up the molecule windows by using 'Edit → Delete → Delete All Molecules'.
 - In ADT4.2 widget, select 'Ligand → Input → Open' and choose 'PDB files' in 'Files of type' menu. Select '1MRS_ligand.pdb' then click 'Open'. Save the ligand in PDBQT format by select 'Ligand → Output → Save As PDBQT', name it as '1MRS_ligand.pdbqt' then click 'Save'.
 - In ADT4.2 widget, select 'Grid → Macromolecule → Open', and choose 'all files' option in the 'Files of type'. Select 1MRS_receptor.pdb and click 'Open'. ADT will read its coordinates, add charges, merge nonpolar hydrogens, and assign appropriate atom types. Click 'OK' to accept the changes. A window will pop up to write the PDBQT file. Click 'Save' to write the file namely '1MRS_receptor.pdbqt'.
 - By default, the atomic charge of Mg will be set to 0. This value can be modified later manually using any text editor in Linux (such as gedit, nano, or geany) before docking.
 - Please note to keep the receptor and control ligand loaded in the molecule window to continue with the next step.

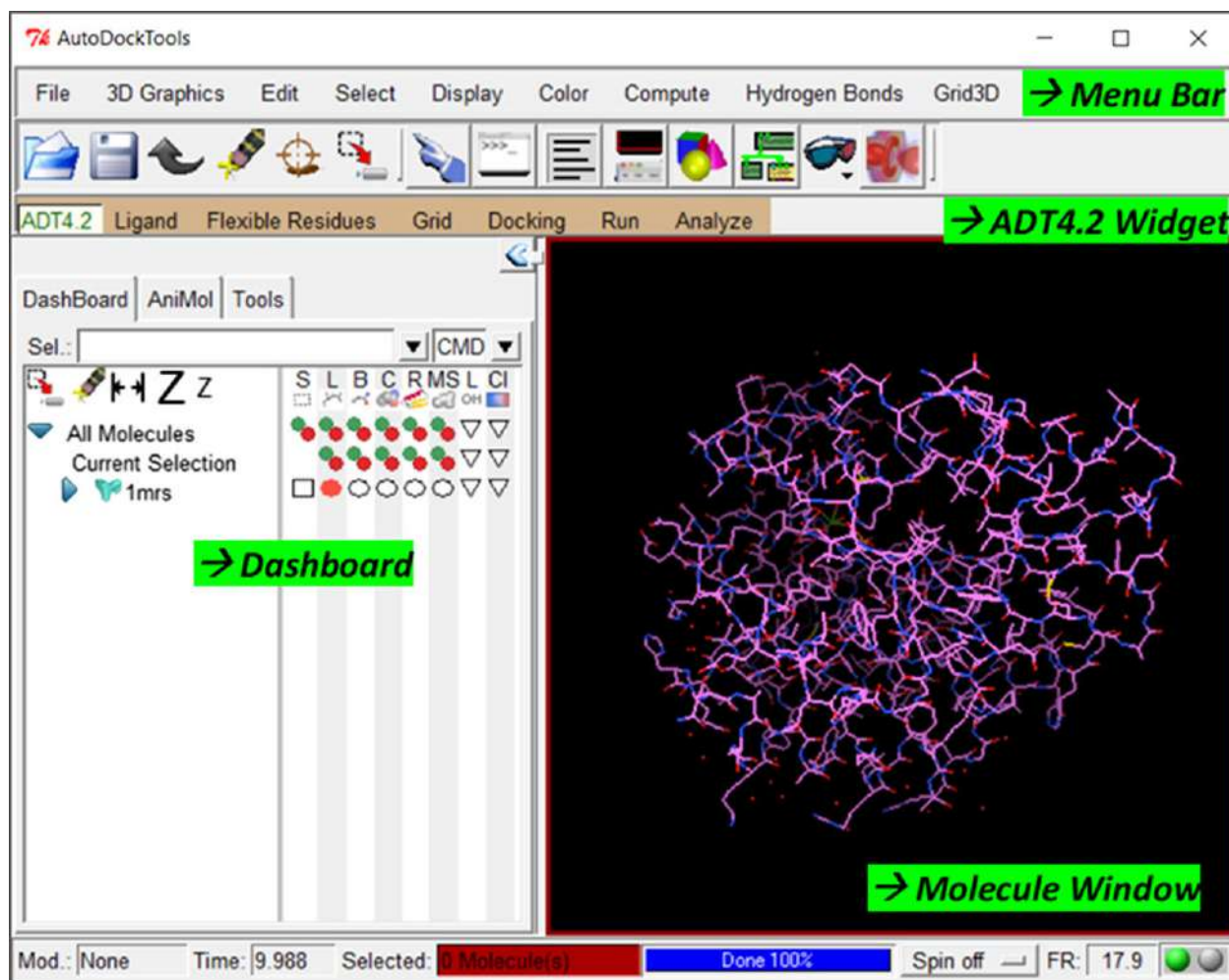


Fig. 4 Interface of AutoDockTools.

Grid Parameter Setup and Affinity Maps Calculation

In this step, we will calculate the affinity maps for each atom types of the ligand. Before that, we should determine the coordinate and the size of the ligand binding site, represented by a grid box, which is used as a target for VS.

- (1) In ADT4.2 widget, specify the atom types included in the loaded 1MRS_ligand.pdbqt by using 'Grid → Set Map Types → Choose Ligand → Select Ligand'.
- (2) Define the center of searching space based on the position of control ligand by using 'Grid → Grid Box → Center → Center on Ligand'. The size of grid box can be adjusted manually to cover the interacting residues (Fig. 5). Save the setting by clicking 'File → Close Saving Current'.
- (3) Save the control ligand's grid parameter file as "control.gpf" using 'Grid → Output → Save GPF'. However, this gpf file is not the one that will be calculated using AutoGrid.
- (4) It is noted that in a virtual screening setting, the size of grid box should consider the size of the compound in the database used for SBVS. This can be performed by using a python script included in the Utilities24 folder, that is, prepare_gpf4.py. This script can also be used to extract all the atom types consisted of the database. Here, "control.gpf" will be used as a reference file to create a global gpf file for the whole virtual screening process.

```
$ cd ~/vs/source
$ pythonsh ~/Utilities24/prepare_gpf4.py -r 1MRS_receptor.pdbqt
-d ../ligand -i control.gpf -o vs.gpf -v
```

This command will generate a new gpf file covering all atom types in the database and adjusting the size of grid box based on the size of all included compound without changing the center of the grid box. It is noted that several ligands have a larger size than the original box size. In this case, it is important to consider whether to follow the new size, which increases the searching space or to stick with the previous size (if the majority of ligands covered). *Take note that the symbol "\ " means to*

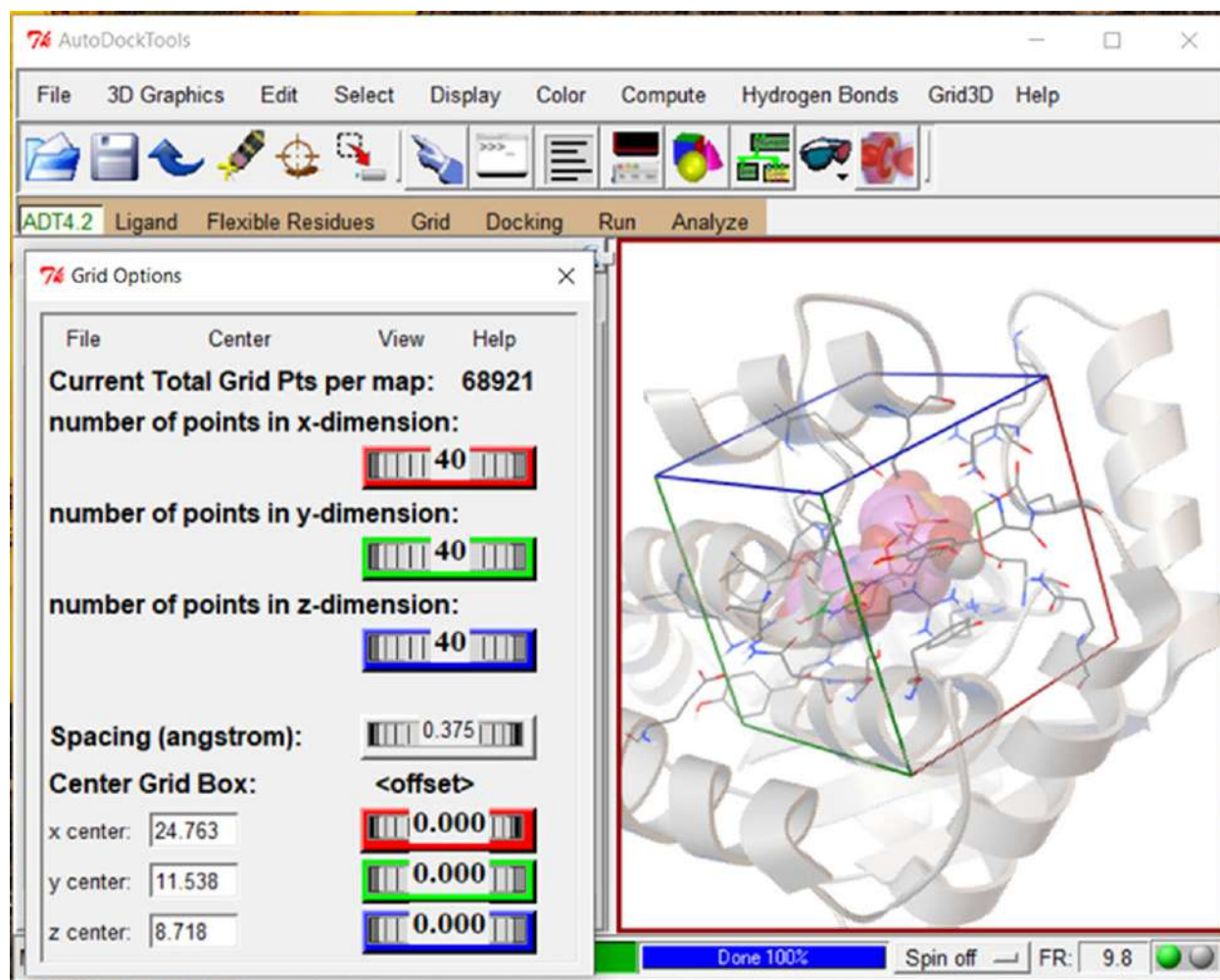


Fig. 5. Grid box setting in AutoDockTools.

continue with the next line. The symbol “\” itself does not need to be typed.

- (5) Calculate the affinity maps of all atom types in the database by using AutoGrid.

```
$ autogrid4 -p vs.gpf -l vs.glg
```

Docking Parameter Setup

In general, there is no good-for-all docking parameter setting. It depends mostly on the searching problem, which is represented by the size of searching space and ligand's degree of freedom. Before preparing the docking parameter for VS, we must do one for the control ligand, which can be used as a reference setting later. AutoDock stores the docking parameter in a DPF file, which can be prepared as follows:

- (1) Select the macromolecule by clicking 'Docking → Macromolecule → Set Rigid Filename' in ADT4.2 widget and choose the '1MRS_receptor.pdbqt'.
- (2) Select the ligand by using 'Docking → Ligand → Choose → 1MRS_ligand.pdbqt'.
- (3) Select the genetic algorithm as the docking method by using 'Docking → Search Parameters → Genetic Algorithm Parameters'. Modify the number of GA runs and a maximum number of evaluation to 100 and 5,000,000, respectively.
- (4) If needed, a more advanced parameter setting can be modified through this menu in the ADT4.2 widget: 'Docking → Docking Parameters'.
- (5) Save the docking parameter file as "control.dpf" by using 'Docking → Output → Lamarckian GA'.

Control Docking or Validation the Docking Method

We may test our docking parameter by control docking method. It is basically a redocking of the cocrystallized ligand from the random state back into its receptor. When the docking is able to reproduce the experimental conformation and position, its

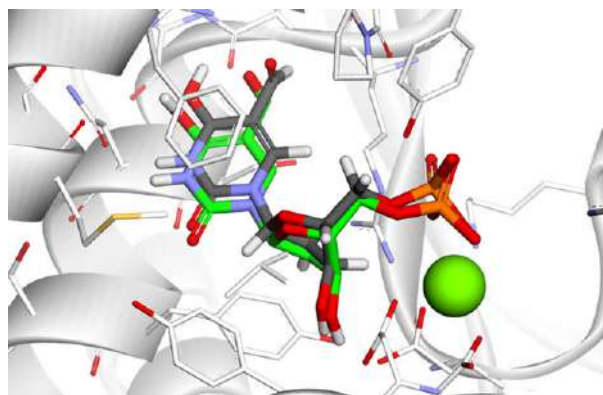


Fig. 6 The result of control docking of ligand (green colored stick) as compared to its crystal pose (gray colored stick). A green sphere is a magnesium ion located at the binding site.

parameter is considered to be used for VS. It is known that the acceptable root-mean-square-deviation (RMSD) value between the docked ligand and crystal ligand is below 2 Å.

Please note that a magnesium ion existed in the receptor file. By default, its charge was set to 0, which is not right. Before docking, we should manually edit the '1MRS_receptor.pdbqt' using any text editor by changing the atomic charge of Mg. In this session, we changed its charge from 0 to 1 (not 2) to mimic the solvent effect around the magnesium ion.

- (1) In the Linux terminal, type:

```
$ cd ~/vs/source
```

```
$ autodock4 -p control.dpf -l control.dlg
```

We can use 'tail -f control.dlg' to monitor the progress of docking process.

- (2) Analyze the DLG file by using any text editor, especially under 'Clustering Histogram' section. The lowest energy docking pose is mentioned after the RMSD Table.

RMSD TABLE						
Rank	Sub-Rank	Run	Binding Energy	Cluster RMSD	Reference RMSD	Grep Pattern
1	1	1	-9.28	0.00	0.97	RANKING
1	2	5	-8.07	1.02	1.08	RANKING
1	3	3	-8.06	0.93	1.31	RANKING
1	4	4	-7.96	1.77	1.70	RANKING

A successful control docking was indicated by the small RMSD value between docked pose and that of reference (0.97 Å) (Fig. 6). It is noted that if we did not change the atomic charge of Mg, the docking result would be worse than this. A separate docking was conducted using the Mg charge of 0, which resulted in an RMSD of 4 Å, indicating a nonreliable docking parameter.

Virtual Screening

Fig. 7 shows that our VS working directory consists of three subfolders, namely ligand, docking, and source. All ligand PDBQT are stored in ligand folder. Using a reference grid and docking parameter files, receptor PDBQT and affinity maps in the source folder, we will generate a working directory for each ligand in docking folder. Each ligand's folder will consist of PDBQT files of ligand and receptor, affinity maps, and docking parameter file. Moreover, there are Python scripts that are very useful in helping the virtual screening process, located in the folder "Utilities24" of ADT installation (by default, it is located in '/home/USER/mgltools*/MGLToolsPkgs/AutoDockTools/Utilities24'. In this practical session, the whole directory of Utilities24 has been copied to the home directory. Hence it can be accessed at '~ /Utilities24'. (note: "~" means /home/USER/ directory)

A map of working directories is visualized in Fig. 7, particularly to organize the virtual screening process in this exercise.

Bash script provided in this article might not the smartest way to do DBVS. However, it is noted that this article is provided to the beginner, especially those who want to do DBVS using their personal computer, not on the multinodes rack server. This process can be done by executing a bash script as follows: (If you are still not familiar with a bash script, please note to press "enter" at the end of each line)

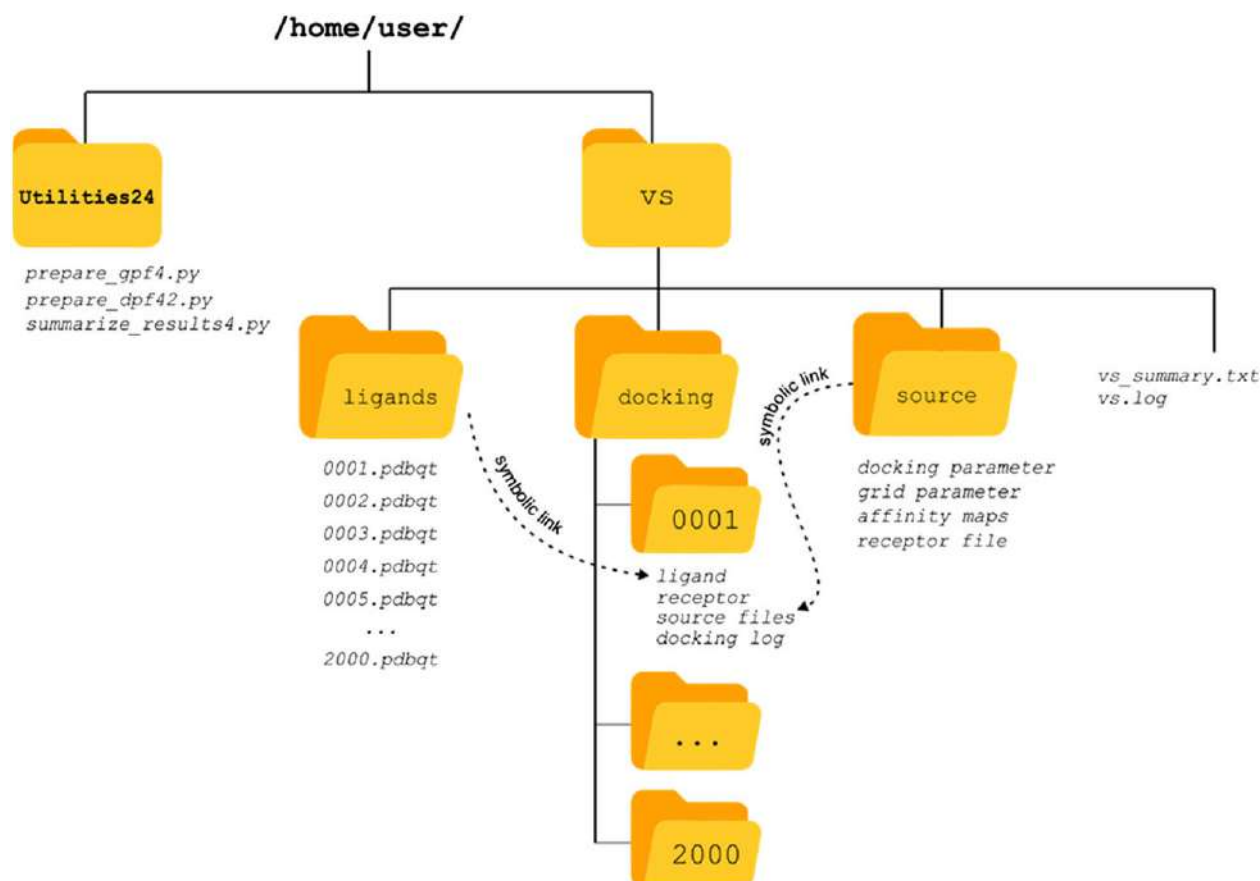


Fig. 7 Structure of directory in DBVS.

To type in the terminal	Explanation
<code>\$ cd ~/vs/docking/</code>	Enter the working directory
<code>\$ for i in \$(ls ../ligand/)</code>	For each "i" in the result of "ls" of ligand directory
<code>do echo \$i</code>	Display list of "i"
<code>j=\$(basename \$i .pdbqt)</code>	Declare "j" from the removal of ".pdbqt" from "i"
<code>echo \$j</code>	Display the list of "j"
<code>mkdir \$j</code>	Create the directory of "j"
<code>cp ../ligand/\$i \$j</code>	Copy the ligand.pdbqt to its respective folder
<code>done</code>	Done, iterate back until all "i" finished

At this point, each ligand's working directory consists of ligand.pdbqt. Since a docking process needs a receptor file, affinity maps, and docking parameter file, then the following script should be executed:

To type in the terminal	Explanation
<code>\$ cd ~/vs/docking/</code>	Enter the working directory
<code>\$ for i in \$(ls)</code>	For each "i" in the result of "ls"
<code>do echo \$i</code>	Display list of "i"
<code>cd \$i</code>	Enter the directory of "i"
<code>ln -s ../source/1MRS_receptor.pdbqt.</code>	Create a link to receptor file
<code>ln -s ../source/*map*.</code>	Create a link to map files
<code>pythonsh ~/Utilities24/prepare_dpf42.py</code>	Prepare a docking parameter file for each ligand based on the reference file (control docking)
<code>-l \$i.pdbqt -r 1MRS_receptor.pdbqt</code>	
<code>-i ../source/control.dpf -v</code>	
<code>cd..</code>	Exit the directory "i"
<code>done</code>	Done, iterate back until all "i" finished

The script of "prepare_dpf42.py" will use the parameters in control docking "dpf" as the reference to generate a new "dpf" file for each ligand docking.

Finally, a run of VS will be executed. This script will enter each ligand's directory and do docking. After finishing with one docking, this script tells to go back to the parent directory and continue with the next ligand.

<i>To type in the terminal</i>	<i>Explanation</i>
\$ cd ~/vs/docking/	Enter the working directory
\$ for i in \$(ls)	For each "i" in the result of "ls"
do echo \$i	Display list of "i"
cd \$i	Enter the directory of "i"
autodock4 -p *.dpf -l \$i.dlg	Execute docking of ligand "i"
cd ..	Exit the directory of "i"
done	Done, iterate back until all "i" finished

Analyze the Virtual Screening Results

Summarize the VS results by using a Python script from AutoDockTools, namely summarize_results4.py, which located in the "Utilities24" folder.

```
$ cd ~/vs/docking
$ for i in $(ls)
do echo $i
pythonsh ~/Utilities24/summarize_results4.py -d $i -t 2.0 -B -k -e
-r 1MRS_receptor.pdbqt -o ../vs_summary.txt -a -v > > ../vs.log
done
```

Here is the detail of the options of summarize_results4.py:

- d Set the directory name that represented the docking folder of each ligand
- t RMSD tolerance between poses to cluster the docking result
- B to result in only the best docking and the largest cluster only (not all poses will be printed)
- k to report the number of hydrogen bond and its energy
- e to break down the binding energy into electrostatic, van der Waals, hydrogen bond, and desolvation energies
- r to identify the receptor file
- o output filename
- a append to output filename. This is ideal for analyzing VS results since they will be appended to a single file
- v verbose output

The resulted "vs_summary.txt" is a comma-separated value (CSV) file that can be imported to a popular worksheet program such as Microsoft Excel (as XLS file). Here is an example of an XLS file, a popular worksheet format, derived from the VS summary file.

Ligand	Lowest energy	No. of H-bond	Electrostatic energy	H-bond energy	Van der Waals energy	Desolvation energy	No. of atoms	No. of torsions	Ligand efficiency
1	-11.42	5	-4.94	-2.35	-10.34	4.82	25	5	-0.54
2	-8.22	5	-0.66	-2.17	-10.05	3.77	23	4	-0.41
3	-7.32	4	-0.72	-1.81	-8.71	3.33	19	3	-0.46
4	-11.46	3	-5.34	-2.56	-11.28	5.45	28	7	-0.50
5	-11.17	3	-4.60	-1.54	-12.48	4.98	29	6	-0.41
Control	-10.82	6	-4.17	-2.75	-10.89	5.00	25	7	-0.49

In principle, the calculated binding energy represents the predicted affinity of ligand to the receptor. However, many DBVS efforts resulted in a false positive result if the selection of active hits was only based on the binding energy. Docking score is not the absolute binding energy, hence we should consider the other aspects in filtering the screening result. For example, the TMPK's binding site is composed of many charged residues. Therefore, electrostatic interactions between the ligand and receptor are expected, including hydrogen bond formation. The screening result then can be sorted based on its electrostatic or hydrogen bond energy. The script of "summarize_results4.py" provides the breakdown of total binding energy into its components, such as a number of the hydrogen bond, intermolecular energy, and desolvation energy. Ligand efficiency also can be used to predict the ligand potency in binding to the receptor. It is a ratio between the binding energy and number of the atom. A higher value of ligand efficiency indicates a major contribution of a ligand's atoms to the interaction with the binding site. For this reason, the result of VS can be sorted based on many parameters, not only the binding energy.

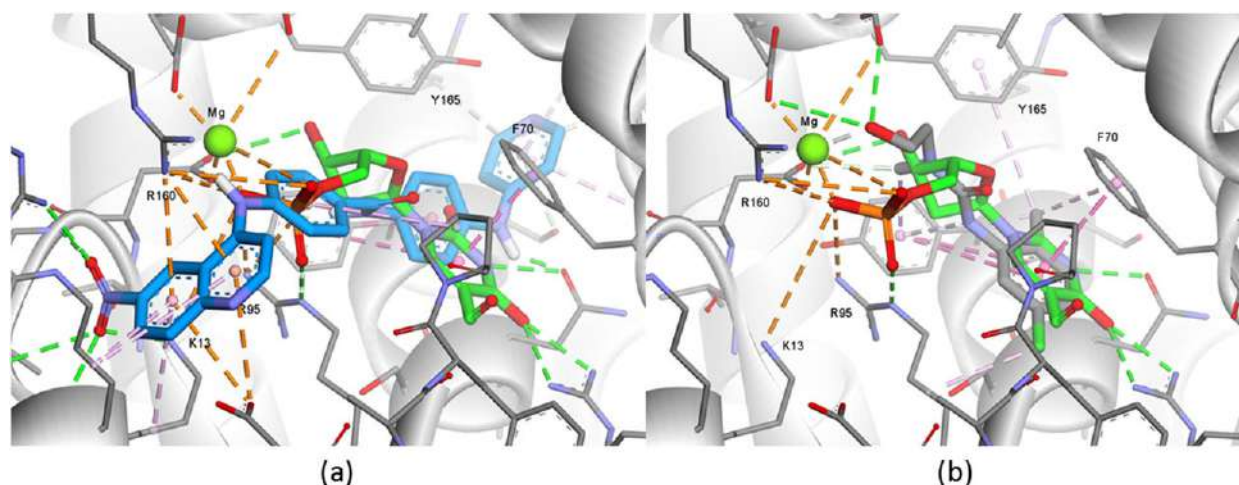


Fig. 8 Molecular interaction of (a) the best ligand (blue sticks) and (b) the ligand with similar pose with the positive control (gray sticks).

Besides the energy value, the selection of the best ligand can be assessed based on the intermolecular interaction with the receptor. For this purpose, the best conformation of each ligand can be prepared using the following script:

To type in the terminal	Explanation
\$ cd ~/vs/docking/	Enter the working directory
\$ for i in \$(ls)	For each "i" in the result of "ls"
do echo \$i	Display list of "i"
cd \$i	Enter the directory of "i"
pythonsh	Extract the ligand's conformation with the lowest energy, resulted in a
~/Utilities24/write_lowest_energy_ligand.py	"i_BE.pdbqt" file
-f \$i.dlg -v	
pythonsh ~/Utilities24/pdbqt_to_pdb.py	Convert the pdbqt format of "i_BE.pdbqt" into pdb file
-f *BE.pdbqt	
cd ..	Exit the directory of "i"
Done	Done, iterate back until all "i" finished

Then, we collect all the best conformation of ligands into a new folder called "best_pose."

To type in the terminal	Explanation
\$ cd ~/vs/docking/	Enter the working directory
\$ mkdir ../best_pose	Create a new directory of "best_pose" in the upper level
\$ cp */*BE.pdb ../best_pose	Copy all of the best poses of ligands into the new directory

Furthermore, we can combine the information derived from the table of results with the visual assessment. The PDB file resulting from the step above can be imported directly to BDSV (or any visualizer program). **Fig. 8(a)** shows that ligands with the best docking score sometimes do not form similar interaction with the control ligand. Near the magnesium ion, there are three positively charged residues, i.e., K13, R95, R160. While the control ligand formed a hydrogen bond, the best ligand formed Pi-cation interactions. Another ligand with moderate binding energy could form similar pose and interaction (**Fig. 8(b)**). The aromatic ring of this ligand perfectly fit the position of control ligand. Therefore, this ligand is also worth testing in further experiments.

Conclusion

The advances of docking software and computer hardware technology, and the availability of structure databases and experimental data, are strong reasons for integrating DBVS into each drug discovery project. DBVS is useful for narrowing hits candidates before in vitro experiment. However, in-depth analysis of DBVS result would enhance the successful rate of prediction. Deep understanding of the protein target should be useful to pick important parameters in selecting the best ligands for further testing, besides the docking binding energy. The protocol explained in this article is an introduction of how DBVS can be done using a personal

desktop computer, which is usually found in the office or laboratory. A tutorial of DBVS in a Linux environment can be also a good start to further advance VS procedures, which mostly run in Linux.

See also: Algorithms for Structure Comparison and Analysis: Docking. Biomolecular Structures: Prediction, Identification and Analyses. Chemoinformatics: From Chemical Art to Chemistry in Silico. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Integrative Bioinformatics. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Pharmacophore Development. Quantitative Structure-Activity Relationship (QSAR): Modeling Approaches to Biological Applications. Rational Structure-Based Drug Design. Small Molecule Drug Design. Structure-Based Drug Design Workflow

References

- Breda, A., Basso, L.A., Santos, D.S., de Azevedo Jr, W.F., 2008. Virtual screening of drugs: Score functions, docking, and drug design. *Current Computer-Aided Drug Designs* 4, 265–272.
- Cosconati, S., Forli, S., Perryman, A.L., *et al.*, 2010. Virtual screening with autodock: Theory and practice. *Expert Opinion on Drug Discovery* 5 (6), 597–607. doi:10.1517/17460441.2010.484460.
- Dunkel, M., Fullbeck, M., Neumann, S., Preissner, R., 2006. SuperNatural: A searchable database of available natural compounds. *Nucleic Acids Research* 34 (Database issue), D678–D683.
- Ewing, T.J., Makino, S., Skillman, A.G., Kuntz, I.D., 2001. DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases. *Journal of Computer-Aided Molecular Design* 15 (5), 411–428.
- Friesner, R.A., Banks, J.L., Murphy, R.B., *et al.*, 2004. Glide: A new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of Medicinal Chemistry* 47 (7), 1739–1749. doi:10.1021/jm0306430.
- Ghosh, S., Nie, A., An, J., Huang, Z., 2006. Structure-based virtual screening of chemical libraries for drug discovery. *Current Opinion in Chemical Biology* 10 (3), 194–202.
- Halgren, T.A., Murphy, R.B., Friesner, R.A., *et al.*, 2004. Glide: A new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *Journal of Medicinal Chemistry* 47 (7), 1750–1759. doi:10.1021/jm030644s.
- Haouz, A., Vanheusden, V., *et al.*, 2003. Enzymatic and Structural Analysis of Inhibitors Designed against Mycobacterium tuberculosis Thymidylate Kinase: New Insights into the Phosphoryl Transfer Mechanism. *Journal of Biological Chemistry* 278 (7), 4963–4971.
- Hong, J., 2011. Role of natural product diversity in chemical biology. *Current Opinion in Chemical Biology* 15 (3), 350–354.
- Ikrum, N.K., Durrant, J.D., Muchtaridi, M., *et al.*, 2015. A virtual screening approach for identifying plants with anti H5N1 neuraminidase activity. *Journal of Chemical Information and Modeling* 55 (2), 308–316. doi:10.1021/ci500405g.
- Jain, A.N., 2006. Scoring functions for protein-ligand docking. *Current Protein & Peptide Science* 7 (5), 407–420.
- Jones, G., Willett, P., Glen, R.C., Leach, A.R., Taylor, R., 1997. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology* 267 (3), 727–748.
- Kastritis, P.L., Bonvin, A.M., 2010. Are scoring functions in protein-protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark. *Journal of Proteome Research* 9 (5), 2216–2225. doi:10.1021/pr9009854.
- Kroemer, R.T., 2003. Molecular modelling probes: Docking and scoring. *Biochemical Society Transactions* 31 (Pt 5), 980–984. doi:10.1042/bst0310980.
- Lavecchia, A., Di Giovanni, C., 2013. Virtual Screening Strategies in Drug Discovery: A Critical Review. *Current Medicinal Chemistry* 20 (23), 2839–2860.
- Lengauer, T., Lemmen, C., Rarey, M., Zimmermann, M., 2004. Novel technologies for virtual screening. *Drug Discovery Today* 9 (1), 27–34.
- Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney, P.J., 2001. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews* 46 (1-3), 3–26. S0169-409X(00)00129-0 [pii].
- Lopez-Perez, J.L., Theron, R., del Olmo, E., Diaz, D., 2007. NAPROC-13: A database for the dereplication of natural product mixtures in bioassay-guided protocols. *Bioinformatics* 23 (23), 3256–3257.
- McGann, M., 2011. FRED pose prediction and virtual screening accuracy. *Journal of Chemical Information and Modeling* 51 (3), 578–596. doi:10.1021/ci100436p.
- Monge, A., Arrault, A., Marot, C., Morin-Allory, L., 2006. Managing, profiling and analyzing a library of 2.6 million compounds gathered from 32 chemical providers. *Molecular Diversity* 10 (3), 389–403. doi:10.1007/s11030-006-9033-5.
- Morris, G.M., Goodsell, D.S., Huey, R., Olson, A.J., 1996. Distributed automated docking of flexible ligands to proteins: Parallel applications of AutoDock 2.4. *Journal of Computer-Aided Molecular Design* 10 (4), 293–304.
- Muchtaridi, M., Bing, C.S., Abdurrahim, A.S., Wahab, H.A., 2014. Evidence of combining pharmacophore modeling-docking simulation for screening on neuraminidase inhibitors activity of natural product compounds. *Asian Journal of Chemistry* 14 (S.1), 59–63.
- Quinn, R.J., Carroll, A.R., Pham, N.B., *et al.*, 2008. Developing a drug-like natural product library. *Journal of Natural Products* 71 (3), 464–468. doi:10.1021/np070526y.
- Rarey, M., Kramer, B., Lengauer, T., Klebe, G., 1996. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology* 261 (3), 470–489.
- Ripphausen, P., Nisius, B., *et al.*, 2013. State-of-the-art in ligand-based virtual screening. *Drug Discovery Today* 16 (9), 372–376.
- Rollinger, J.M., Langer, T., Stuppner, H., 2006. Strategies for efficient lead structure discovery from natural products. *Current Medicinal Chemistry* 13 (13), 1491–1507.
- Shoichet, B.K., McGovern, S.L., Wei, B., Irwin, J.J., 2002. Lead discovery using molecular docking. *Current Opinion in Chemical Biology* 6 (4), 439–446. S1367593102003993 [pii].
- Singh, S., Malik, B.K., Sharma, D.K., 2006. Molecular drug targets and structure based drug design: A holistic approach. *Bioinformation* 1 (8), 314–320.
- Tanrikulu, Y., Krüger, B., *et al.*, 2013. The holistic integration of virtual screening in drug discovery. *Drug Discovery Today* 18 (7), 358–364.
- Verheij, H.J., 2006. Leadlikeness and structural diversity of synthetic screening libraries. *Molecular Diversity* 10 (3), 377–388. doi:10.1007/s11030-006-9040-6.
- Villoutreix, B.O., Eudes, R., Miteva, M.A., 2000. Structure-based virtual ligand screening: recent success stories. *Combinatorial Chemistry & High Throughput Screening* 12, 1000–1016.
- Vyas, V., Jain, A., Jain, A., Gupta, A., 2008. Virtual screening: A fast tool for drug design. *Scientia Pharmaceutica* 76, 333–360.
- Wahab, H.A., Asarudin, R.M., Ahmad, S., *et al.* (2009). Nature based drug discovery (NADI) and its application to novel neuraminidase inhibitors identification by virtual screening, pharmacophore modelling and mapping of Malaysian medicinal plants. Trieste, Italy: Paper Presented at the Drug Design and Discovery for Developing Countries.
- Wallace, A.C., Laskowski, R.A., Thornton, J.M., 1995. LIGPLOT: A program to generate schematic diagrams of protein-ligand interactions. *Protein Engineering* 8 (2), 127–134.
- Xu, H., Agrafiotis, D.K., 2002. Retrospect and prospect of virtual screening in drug discovery. *Current Topics in Medicinal Chemistry* 2, 13005. 11320.

Relevant Websites

<https://dtp.cancer.gov/RequestCompounds/index.xhtml>

Compound Request Form - Developmental Therapeutics Program.

<https://dtp.cancer.gov/organization/dscb/obtaining/vialed.htm>

DSCB | Obtain Vialled.

<http://autodock.scripps.edu/downloads>

Download Instructions - AutoDock - The Scripps Research Institute.

<http://autodock.scripps.edu/resources/databases>

Databases - AutoDock - The Scripps Research Institute.

<http://mgltools.scripps.edu/downloads>

Downloads - MGLTools - The Scripps Research Institute.

<http://accelrys.com/resource-center/downloads/freeware/index.html>

Freeware Software - Accelrys.

<http://autodock.scripps.edu/resources/raccoon>

Raccoon | AutoDock - The Scripps Research Institute.

Molecular Phylogenetics

Michael A Charleston, University of Tasmania, Hobart, TAS, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

This Section begins with a basic motivation as to why (molecular) phylogenetics is important, and then introduces some of the relevant terminology. Section “Overview” continues with an overview of the phylogenetic inference process: From sequence alignment to constructing or estimating trees. Section “Common Assumptions in Molecular Phylogenetics” then explores the assumptions that are made in the name of phylogenetic inference: To do with the alignment itself, assumptions about independence, the direction of time, the nature of speciation, and similar. The last section discusses methods of finding the best tree, followed by a brief discussion on Bayesian methods and MCMC before some final comments on what (else) can possibly go wrong with molecular phylogenetic inference.

Motivation

Phylogenetic analysis is strongly motivated by our need to understand the relationships among species that have evolved from common ancestors. As – at least in theory – *everything living* has arisen from a common ancestor around 3.5–4.4 billion years ago, it is not beyond our imagination to conceive of a single over-arching “Tree of life” that describes the relationships among all extant and extinct species, from viruses to plesiosaurs. Understanding such relationships, both in terms of patterns of speciation and the times at which they occurred, is key to our ability to compare species within a framework that makes statistical *sense*. Without frameworks like phylogenetic trees (or even networks – see later) it is not possible to properly account for the shared history of species when comparing their characteristics.

But estimating phylogenetic history is hard. There are a number of reasons for this, but perhaps the most obvious is that we cannot travel backward in time to check whether what we have attempted to uncover is in fact correct. Another difficulty is that the number of phylogenetic trees is enormous. It’s been long known (since at least 1870 ([Schröder, 1870](#)); more elegantly in ([Cavalli-Sforza and Edwards, 1967](#))) that the number of possible trees for n species grows at a frightening rate: so much so that for even modest numbers of species, around 100, there are more than 3×10^{184} trees – many, many more than the estimated number (something around 10^{82}) of particles in the universe! So not only is it hard to check whether a tree is “right” (leaving aside for a moment the potential issues of assuming that evolution is tree-like), but it’s also computationally intractable to search through the space of all possible trees for the best one.

Language of Phylogenetic Inference

Phylogenetics is rich in terminology, having gained it from mathematics, biology, and computer science, and it can be confusing. It is worth defining the terms we will be relying on at this point ([Box 1](#)).

Box 1: Terminology

Analogy: a feature of related species that is similar, but is not derived from their common ancestor (c.f., homology).

Branch: also known as an edge in a phylogenetic tree, it is equivalent to a split of the leaf nodes (taxa) into two subsets. The “same” branches can therefore occur in multiple trees, so long as they separate the taxa into the same subsets.

Character, character state: we distinguish the character, which is the particular feature such as “nucleotide” at a given site, with its state, which in this case would be A, C, G, or T.

Clade: a monophyletic group.

Graph: a collection of nodes, and pairs of nodes called edges.

Homologous: of a character, feature or site that is shared and derived from a common ancestor (c.f., analogy).

Leaf (tip) a node of a tree that is adjacent to exactly one branch (or edge); corresponding to extant taxa.

Likelihood: a method by which the probability of a given observation can be calculated, conditional on an assumed probabilistic model.

Locus; plural loci: a site or sequence within a molecular sequence.

MRCA most recent common ancestor of a set S of taxa: The node in a rooted phylogeny that is closest to the leaves, that is ancestral to all the taxa in S .

Monophyletic, monophyly: a taxon set S in a tree is monophyletic if it contains all the descendants of the most recent common ancestor of S .

Network: in phylogenetics, a graph that contains multiple paths between at least some pairs of nodes, either representing a hypothesized history of hybridization and/or horizontal gene transfer, or support for conflicting splits of the taxa.

Paraphyletic, paraphyly: a taxon set S in a tree is paraphyletic if it contains most, but not all, of the descendants of the MRCA of S .

Phylogeny: also called a phylogenetic tree, a tree whose nodes represent taxa or operational taxonomic units (OTUs) and whose branches correspond to evolutionary lineages.

Root: a node of a phylogenetic tree that is specially designated as the common ancestor of all the other nodes.
 Sequence: in the context of molecular phylogenetics, sequence means linear array of biological characters, e.g., nucleotides.
 Site: a position in a molecular sequence, e.g., the 10th nucleotide in the gene coding for *gag* protein.
 Split: a partition of the taxon set of interest into two subsets, with empty intersection and no taxa missing. Equivalent to a branch.
 Taxon; plural taxa: a phylogenetically distinct biological type, e.g., species or genus.
 Tip: a node corresponding to an extant taxon.
 Tree: a graph in which each pair of nodes has exactly one path connecting them.

Overview

Molecular phylogenetics is concerned with the estimation of the evolutionary relationships of species or other types of taxa that we can observe today. It is a computationally and statistically challenging task, but it is central to enabling comparison of species within an appropriate statistical framework.

Molecular phylogenetics is a routine pursuit in modern biology, and while it is broadly a simple process, the details are not inconsequential. The general process is as follows: (1) Obtain a set of comparable molecular sequences across the range of taxa of interest, (2) align those sequences so individual positions in each sequence correspond to each other, (3) build or search for a tree or set of trees with methods as briefly described herein, (4) test the tree(s) for their correctness, and generally (5) use the tree(s) to answer a question about the species, e.g., when was the divergence into the major clades of birds. It is possible to skip step (2) if re-analysing an existing data set.

Data

Beginning the whole inference process is the creation and curation of suitable data.

For molecular phylogenetics, the data are sets of homologous sequences, e.g., those of particular genes, represented as strings of adenine (A), guanine (G), cytosine (C) and thymine (T) for DNA, or A, C, G, uracil (U) for RNA sequences; occasionally too these sequences are transformed into simpler encoding as purines (A, G) and pyrimidines (C, T/U). Of course it is also possible to use amino acids as characters in such sequences, or even the 64 codons that encode them (usually removing the 3 “stop” codons from consideration within a protein encoding gene).

Homologous means that the sequences correspond to each other across species in the obvious desirable way: They branched and diversified from each other at the speciation events that we’re trying to recover. Sequences that are paralogous, that is, they don’t have the same evolutionary history as the species in which they reside, are more problematic for phylogenetic analysis because correctly recovering the evolutionary relationships among the sequences does not directly give you the relationships among the species.

Usually sequences that are likely to show some evolutionary “signal” are chosen for phylogenetic inference: those that have some constraints so they do not evolve too quickly as to appear random with respect to each other.

Depending on the overall age of the set of taxa of interest, slower or faster regions in the homologous sequences are chosen. For example to estimate the phylogenetic relationships among a group of viruses over 100 years, the *env* gene might be used; for investigating deep historical relationships among archaea, bacteria and eukaryota, much more conserved genetic regions must be chosen such as DNA polymerases.

Aligning the Sequences

The sequences must be aligned, that is, arranged relative to each other with the possible insertion of gaps at different places, such that for each sequence, the characters (e.g., nucleotides, amino acids, purines/pyrimidines etc) at position *i* have the same evolutionary history as each other. In this sense the positions, which are generally called *sites*, are also hoped to be homologous. [Fig. 1](#) shows how some toy DNA sequences might evolve on a tree, showing homologous sites, insertion/deletion events, and illustrating how reconstructing the branching order might be tricky. Maybe reword here.

Alignment would be trivially easy if the only mutations that could occur were changes of state, e.g., from A to T, but sequences evolve in other ways also. In the context of sequence alignment the main other mode by which sequences can change is via deletion of sections of sequence, or insertion of new sections. Faced with two sequences that have been optimally aligned, in which there is a section in one that is “extra” (see [Fig. 1](#) and the table below), it is not possible to determine whether that section was inserted in one sequence, or deleted from the other one. For this reason such regions are referred to also as “indels” and the events are indel (insertion/deletion) events.

To find the optimal alignment of just two sequences is relatively straightforward: first, choose a costing function that accounts for similarities and differences between aligned sites in the two sequences, as well as gaps in either caused by insertion or deletion events during their evolution, and then use dynamic programming to search through the complete space of possible alignments in an efficient way ([Waterman et al., 1991](#)). Costing the match or mismatch of characters like nucleotides, amino acids or codons is usually done by performing log-odds analyses of pairs of curated sequences that are agreed to be correctly aligned, at various levels of difference (e.g., BLOSUM ([Henikoff and Henikoff, 1992](#)), [Fig. 2](#); PAM ([Dayhoff et al., 1978](#)); JTT, ([Jones et al., 1992](#)) etc.). The match and mismatch scores are often rounded to integer values for computational efficiency, which is generally a “safe” assumption given that the log-odds are themselves approximations, derived from hand-picked and often hand-aligned data.

	A	4			
	G	1	5		
	C	0	-1	4	
Score =	T	1	-1	0	3
	gap	-7	-7	-7	-7
	A	G	C	T	

Fig. 3 Example cost matrices for nucleotide alignment. These scores are illustrative only.

In Alignment 1 a gap has been inserted at position 3 for taxa g and h, with the remaining sequences having a G at that position; the other sites all show multiple character states. Alignment 2 has a constant site at position 5, but at considerable expense elsewhere in the alignment as more gaps have been inserted than are biologically realistic. (Note that if Alignment 2 were optimal for some alignment program, the gap cost is set far too low in comparison to that for a mismatch between character states.) In general we hope to obtain alignments that show a degree of conservatism across species in the majority of sites, because these have a higher chance of each site being aligned to its homolog in other species. If the alignment for a set of sequences or parts thereof is difficult with much ambiguity, e.g., evidenced by particularly “gappy” alignments (as with Alignment 2 above), then such data may be dropped from the analysis. There are several software tools to automatically drop such sites from analysis, e.g., GBlocks (Castresana, 2000).

Multiple sequence alignment (MSA), for more than two sequences, is known to be computationally intractable: To guarantee an optimal scoring alignment even under the relatively simple affine scoring method above, requires an amount of space and time that grows exponentially with the number of sequences. Therefore for MSA the typical approach is to use one or more of a suite of well-established heuristic sequence alignment methods, such as MAFFT (Katoh *et al.*, 2002), muscle (Edgar, 2004), k-align (Lassmann and Sonnhammer, 2005), and T-Coffee (Notredame *et al.*, 2000) and often to then adjust their outputs “by eye”. Heuristic methods do not guarantee optimality (unlike for pair-wise alignment), and may not always yield the same output. For example two separate runs of any of these programs may in principle yield two different alignments, particularly if there are any equally high-scoring options, as ties have to be resolved either arbitrarily (giving rise to unbiased but inconsistent behaviour) or deterministically (giving rise to consistent, but biased behaviour).

MSA, or often “sequence alignment” or even simply “alignment” – context notwithstanding – just like pairwise alignment, has the aim of associating individual positions along every sequence with corresponding positions in the other sequences, such that each position corresponds to the same evolutionary history.

It is in principle possible to combine MSA with tree estimation, but these methods have yet to become standard in the phylogenetics community.

While the multiple alignment problem is computationally too demanding to solve perfectly, current heuristic methods can be very fast and reliable: Using default parameters that are moderately well tuned, it is entirely possible to perform this part of the phylogenetic inference workflow quite quickly, and without inspecting the data at all. *Caveat emptor*.

For protein coding genes, more complex alignment methods exist that take into account the three-dimensional structure of the proteins.

The oldest, but largely superseded, automatic alignment program is Clustal (Higgins *et al.*, 1992). It has default values of a gap opening cost of 10.0 and a gap extension cost of 0.1, so a gap of length 3 nucleotides has a cost of 10.3 by default.

In general the gap opening and extension costs can have different best values for different data sets (see Fig. 3).

Common Assumptions in Molecular Phylogenetics

In any inference process we must make assumptions. In phylogenetics these assumptions are mostly incorrect, but it turns out we can get away with making them in many situations. Some of the major ones are listed below:

The Sequences are Homologous

For many organisms, there is only one copy of any given gene, or of some other phylogenetic useful marker. However in eukaryotes (classically, haemoglobins in primates), it is not a safe assumption that genes that encode corresponding proteins across multiple organisms are necessarily homologous: They could be paralogous, that is, their most recent common ancestor might be before the most recent common ancestor of the organisms themselves (see Fig. 4).

The Alignment is Correct

Partly as a result of the computational complexity of inferring a tree from an alignment, it is rarely possible to move forward with phylogenetic inference without assuming that the alignment is good enough and moving on. Therefore the usual practise is to treat the alignment as fixed, and then undertake the even harder task of inferring the phylogeny based on it.

There are some methods available that attempt to perform MSA and tree estimation concurrently – but these are very lengthy approaches, and it is not yet clear that they are an ideal solution, particularly for large data sets.

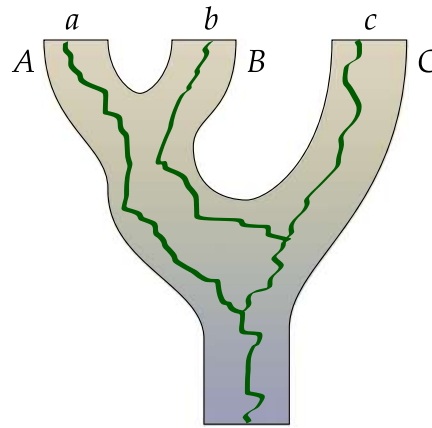


Fig. 4 The species tree here (thick pipes) has a gene tree within it that is not congruent: The two left-most genes, for *a* and *b*, are paralogous: Their last divergence is before the divergence of their host species.

Sequence Evolution is Independent and Identically Distributed

This is the most ubiquitous of assumptions made in molecular phylogenetics and is the most often realised as patently false. It states that each site in a sequence evolves in the same way, and independently of, its neighbor. A moment's thought reveals that this cannot be true: In codons first, second and third positions have different selective constraints on them. Loci that code for microRNAs (miRNAs) will have sites within them that, if they change, must change in concert else they will destroy the ability of the miRNA to fold correctly. In protein-coding genes the bulk of evolving sites correspond to amino acids that interact with others. There are many cases where concomitantly varying sequences must be a better reflection of reality than the i.i.d. assumption (Lockhart *et al.*, 1998; Wang *et al.*, 2006) but yet for practical purposes it is often still useful to make this simplification: To a great extent the huge success of molecular phylogenetics is down to the flood of phylogenetically informative data that permit the robust estimation of phylogenetic history despite this incorrect assumption.

Time is Continuous

Biologically speaking, it does not make sense that time be treated as a continuous quantity: mutations occur at discrete time points such as cell division after all. But it is an assumption that we can get away with (as we do with the i.i.d. assumption), because in most cases the evolutionary changes occur at such a gradual pace in comparison with the time periods over which we measure them, that it may as well be modelled as a stochastic process in continuous time.

The mathematical benefits are significant for moving to continuous time, as we can employ tools from stochastic modelling such as the exponentiation of rate matrices to produce probabilities of change over arbitrary time periods. Without the assumption of continuous time, we really could not do phylogenetic estimation in any statistically justifiable way, but an assumption it is and for completeness, it is included here.

Time-Reversibility

Until recently almost all models of molecular evolution have worked under the assumption that evolution does not care about the direction of time: The probability of changing from one nucleotide to another in a given time interval is always exactly the same as that probability in the reverse direction.

Clearly this is not biologically correct! As soon as one understands that there can be selection toward a particular sequence then reversibility is broken. However, this is a crucial simplification that enables the likelihood of a set of aligned sequences under a model to be calculated regardless of where the root of the tree might be: Without the assumption, we must in principle calculate the likelihood from every possible location of the root, a considerable computational burden.

Tree-Like Evolution

Possibly the most famous phylogenetic tree is that which was sketched by Charles Darwin in his book "On the origin of species" (Darwin, 1859), with the telling note, "I think." Darwin was clearly imagining the formation of new species by a process of branching, with the possibility of lineages (species) going extinct, but not in terms of them converging again.

This is an important point because, if it is true (and it mostly is), it means that *trees* are the objects that we must use to represent species' origins. If we do not adopt this idea wholly, then we must permit more complex structures, in which there could be multiple paths from a common ancestor to some descendant. In a tree, all paths are unique – this can fact be part of the definition of what a mathematical tree is: A network connecting a collection of objects, such that there is always *exactly* one path between every two such

objects (see later; other places in the book). For the most part, phylogenetic inference (whether using molecular data or anything else) makes this assumption, but there is (often heated) discussion in the scientific literature as to the extent of non-tree-like evolution, caused by such mechanisms as lateral/horizontal gene transfer (LGT, HGT respectively) and hybridization.

Therefore if we think trees describe evolutionary relationships, we need to use the tools of branching processes in order to describe them. All this discussion so far is regarding a process that is moving forward with time, but there are also a wealth of tools to be employed if we look backwards. These latter tools use the concept of lineages *coalescing* as we move into the past, and they continue to yield powerful results that enable researchers to understand better the way species, and particularly lineages within species, emerge.

Another generalisation that we must consider is non-treelike evolution, corresponding to lineages converging, e.g., by hybridization or horizontal gene transfer. In phylogenetics these representations are usually referred to as *networks* (though to a mathematician, all trees are also networks).

For the moment we restrict ourselves to phylogenies as trees (but see later).

Further, we constrain our thinking to trees in which speciation events always give rise to exactly two “new” lineages – such trees are called *binary*, and differentiate between those with a chosen root and those without one (see Time and Reversibility).

Non-Tree-Like Evolution

A given site is assumed to evolve according to a tree-like pattern, only branching as they move forwards in time, but there are other units of evolution that can converge instead. For example, genes may be horizontally (or “laterally”) transferred from one species to another (Berghorsson *et al.*, 2003), and species can hybridize (Harrison and Larson, 2014). In those cases the genes may not share the same evolutionary histories, and the species phylogeny is better represented as a hybridization network.

An alternative is that due to uncertainty in the phylogenetic signal, even though the tree history may be tree-like (branching), it is desired to show a range of alternative hypotheses in the same figure. These phylogenetic networks can illustrate support for branching patterns and splits of the taxa of interest that are conflicting: They cannot both be part of a tree, but they can be shown as, for example, a “NeighborNet” (Bryant and Moulton, 2003). NeighborNets have become a standard tool in molecular phylogenetics. Their construction is based on estimates of pair-wise distances between taxa, rather than on a scoring method for a tree.

What Makes a Tree a Good Tree

Because of the vast number of possible trees, growing super-exponentially in the number of taxa involved, it is a virtual certainty that the estimated phylogeny for very large sets of taxa will be incorrect – somewhere. However, that is not to say that everything in the tree is wrong. It is helpful to think of a phylogeny as a set of non-independent hypotheses of relatedness, many of which may be correct!

For example in Fig. 4 there are several hypotheses displayed:

1. A and B are each others’ closest relatives and form a monophyletic group (a.k.a. clade);
2. A, B and C are monophyletic/form a clade;
3. B and C are paraphyletic in this tree.

A very standard method of measuring robustness of a phylogeny is via *bootstrapping*, which is a method by which sites in an alignment are sampled at uniform random and with replacement, in order to create “pseudo-replicates” of the original data set (Efron *et al.*, 1996). Each pseudo-replicate is then subjected to the same inference methods as for the original data set, and for each tree inferred on the pseudo-replicates, a note is made of which branches are in that tree. The proportion (or percentage) of times in which each branch is recovered is a measure of the phylogenetic consistency of the data.

The philosophy behind such resampling – leading to the idea of “pulling oneself up by one’s bootstraps” is that, if the observed (assumed to be homologous, and independently evolving) sites were representative of a larger population of such sites, then resampling from them would be equivalent to resampling from that larger population. This is a strong assumption, so interpretation of the resulting bootstrap values must be done with care. It is technically not a measure of confidence, as such, as it is possible to gain arbitrarily large bootstrap values simply by having large data sets (Jermini *et al.*, 2005), but in general high values (e.g., 80% or more) are taken to be indications that the data are providing strong signal in favour of a particular branch.

Choosing the Best Tree

In order to select the tree that is the best possible explanation of the evolutionary relationships among a set of species, we must be able either to construct the tree according to some algorithm, such as Neighbor-Joining (Saitou and Nei, 1987; Studier and Keppler, 1988; Gascuel and Steel, 2006), or to evaluate each tree with some kind of score function and search through the space of all possible trees for the one(s) that maximise(s) the score. Next we discuss the two main score functions for trees, and then discuss tree searching.

In order to recover a tree from molecular sequence data we must have in mind some model that effectively describes what we know about sequence evolution. Models arise from different philosophies, and in molecular phylogenetics, the two main ones are that of maximum parsimony, and of (probabilistic) likelihood. Both of these ideas require a search through tree space: the set of

all possible trees that relate the set of taxa of interest, with trees being considered as adjacent with respect to some perturbation if one can be modified to match the other.

Maximum Parsimony

According to *maximum parsimony*, also referred to as Occam's Razor, it is the simplest explanation that is the best explanation of what happened. The origin of this idea is ancient, but it still has some value in modern science, provided one recognises its limitations.

Fitch (Fitch, 1971) devised an ingenious algorithm that counts the minimum number of changes of state that are required on a given tree in order to explain the sequences observed at its tips. The total Parsimony score – Also called the Parsimony *length*, is the sum of the minimum number of changes that must occur on the tree to account for the characters at the leaves of the tree, and it is this score that is to be minimised over all possible trees.

The algorithm is performed on each site in turn and the score (length) for site is accumulated into the total. (Obviously, one can speed things up by calculating it only on unique site patterns and then multiplying by the number of times it occurs, and there are other optimisations possible, but they are not illuminating here; the interested reader is directed to).

Fitch's algorithm was later shown to be a special case of a more general algorithm of Sankoff (Sankoff, 1975), which can also take into account weights for different substitutions. Sankoff's algorithm requires slightly more bookkeeping, in that one must calculate and retain the minimal total cost for every possible character state at each node, but it is beyond the scope of this brief article to go into that detail; the interested reader is directed to Sankoff (1975).

The Fitch algorithm is outlined in **Box 2** for rooted binary trees on nucleotide data.

Box 2: Algorithm for calculating the Parsimony length of a tree

Algorithm 1: ParsimonyLength(T, A)

```

given  $T$ , a rooted tree with tips labelled  $1, \dots, n$ 
given  $A$ , an aligned set of  $n$  observed sequences of length  $\ell$ 
let  $r$  be the root node of  $T$ 
let  $p$  be the parsimony length of  $T$ 
let  $V$  be the set  $V = \{v_1, \dots, v_{n-1}\}$  of non-leaf nodes of  $T$ 
 $p \leftarrow 0$ 
for each ( $i = 1, \dots, \ell$ ) do {
  ·  $p \leftarrow p + \text{Fitch}(r, A, i)$ 
}
return( $p$ )

```

Algorithm 2: Fitch(v, A, i)

```

given  $v$ , a node in a rooted binary tree
given  $A$ , an aligned set of  $n$  observed sequences of length  $\ell$ 
given  $i$ , the position of a site of interest in  $A$ 
let  $F(v)$  be the character state(s) assigned to  $v$ 
let  $q$  be the Parsimony score of this site
 $q \leftarrow 0$ 
if ( $v$  is a leaf) then {
  /*  $F(v)$  is our observed data */
  ·  $F(v) \leftarrow A_{i,v}$ 
  · return(0)
} else {
  · let  $S$  be a set of states allowed (e.g., A, C, G, T)
  ·  $S \leftarrow F(v.\text{leftchild}) \cap F(v.\text{rightchild})$ 
  · if ( $S = \emptyset$ ) then {
    ·  $q \leftarrow q + 1$ 
    ·  $S \leftarrow F(v.\text{leftchild}) \cup F(v.\text{rightchild})$ 
  }
  ·  $F(v) \leftarrow S$ 
}
return( $p$ )

```


For example, consider again the alignment that corresponds to the tree in [Fig. 1](#):

taxon	sequence
d	TCGC
e	TTGT
f	ACGT
g	AG-A
h	GC-C

There are five possible states: A, C, G, T, and the “gap” state -. Consider the first site pattern, TTAAG for the five taxa d, e, f, g, h in order. For this part we use the notation $(a)=X$ to mean node (a) has character state X. Proceeding from the leaves of the tree to the root, we first consider node (c), which has the two child nodes and character states (d) T and (e) T. $\{T\} \cap \{T\} = \{T\}$, so we assign T to node (c). Moving up to node (a) which now has children (c)=T and (f)=A. $\{T\} \cap \{A\}$ is empty so we set $(a) = \{A, T\}$ and add 1 to the Parsimony length of this tree. Next is node (b), which has as its child nodes (and character states) (g)=A and (h)=G. Again we find the intersection of these two sets of character states is empty so we increment the Parsimony length and assign $(b) = \{A, G\}$. The last node is the root, which now has children $(a) = \{A, T\}$ and $(b) = \{A, G\}$. Since $\{A, T\} \cap \{A, G\} = \{A\}$ is non-empty, we assign A to the root node. The Parsimony score for this site is 2.

To calculate the Parsimony score for the complete alignment we repeat the process over the other three site patterns; and if we wish to account for different costs of different substitutions, we can keep track of all possible character states at all the nodes, and retain their minimal scores as we move “up” the tree to the root.

Maximum Likelihood

By far the most common tool used currently to infer molecular phylogenies is maximum likelihood (ML). This is a simple concept but one which has grown to comprise a very rich suite of models and methods. We begin with a toy example to illustrate the key concepts and then mention some (but not all) ways in which it is generalised.

The essential idea of maximum likelihood (ML) is that under a probabilistic model, we may calculate the probability of observing a given outcome. The given outcome we choose is our set of observed data: That is, a set of aligned sequences. The simplest possible model for illustration purposes is a two-state one, in which we have sequences of 0s and 1s, evolving according to a symmetric rate matrix

$$Q_{CF} = \begin{bmatrix} -\alpha & \alpha \\ \alpha & -\alpha \end{bmatrix}$$

This matrix is labelled “CF” as it corresponds to the Cavender-Farris model ([Cavender, 1978](#)) Rate matrices correspond to limits of instantaneous probabilities of change from one state to another; but to calculate an actual probability for a given time period, we must convert from rate matrices to probabilities.

Assuming a Poisson process for mutation events we can determine that for a rate matrix Q operating over a time period t , the probability matrix that corresponds to changes of state over that time period is given by $P = \exp(Qt)$. In the case that Q is conveniently exponentiated this leads to closed forms that can be used to calculate the probability of a given (e.g., nucleotide) state into another, conditioned on that rate matrix being correct.

The above instantaneous rate matrix Q is called doubly stochastic: Both its rows and its columns individually sum to zero. Doubly stochastic models have the property that the steady state distribution of character states is uniform: In this case, our two character states will settle over a long enough period to occur in equal proportions. In the case of DNA models that are doubly stochastic, we expect their nucleotide base frequencies to tend to $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$.

The probability matrix that corresponds to this rate matrix is given by

$$\text{Exp}(Q_{CF}t) = \frac{1}{2} \begin{bmatrix} 1 + e^{-2\alpha t} & 1 - e^{-2\alpha t} \\ 1 - e^{-2\alpha t} & 1 + e^{-2\alpha t} \end{bmatrix}$$

For the next simplest model, still doubly stochastic but with four states, we have

$$Q_{JC} = \begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix}$$

yielding

$$\text{Exp}(Q_{JC}t) = \begin{bmatrix} 1 + 3e^{-4\alpha t} & 1 - e^{-4\alpha t} & 1 - e^{-4\alpha t} & 1 - e^{-4\alpha t} \\ 1 - e^{-4\alpha t} & 1 + 3e^{-4\alpha t} & 1 - e^{-4\alpha t} & 1 - e^{-4\alpha t} \\ 1 - e^{-4\alpha t} & 1 - e^{-4\alpha t} & 1 + 3e^{-4\alpha t} & 1 - e^{-4\alpha t} \\ 1 - e^{-4\alpha t} & 1 - e^{-4\alpha t} & 1 - e^{-4\alpha t} & 1 + 3e^{-4\alpha t} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 + 3\beta & 1 - \beta & 1 - \beta & 1 - \beta \\ 1 - \beta & 1 + 3\beta & 1 - \beta & 1 - \beta \\ 1 - \beta & 1 - \beta & 1 + 3\beta & 1 - \beta \\ 1 - \beta & 1 - \beta & 1 - \beta & 1 + 3\beta \end{bmatrix}$$

where $\beta = e^{-4\alpha t}$.

The most general time-reversible model is known as GTR, the General Time-Reversible model (Tavaré, 1986). GTR parameters can be expressed simply as the equilibrium base frequencies a four-dimensional vector $\pi = (\pi_A, \pi_C, \pi_G, \pi_T)$ whose non-negative entries sum to 1, and a rate matrix

$$Q_{GTR} = \begin{bmatrix} -(\alpha\pi_C + \beta\pi_C + \gamma\pi_T) & \alpha\pi_C & \beta\pi_C & \gamma\pi_T \\ \alpha\pi_A & -(\alpha\pi_A + \delta\pi_C + \varepsilon\pi_T) & \delta\pi_C & \varepsilon\pi_T \\ \beta\pi_A & \delta\pi_G & -(\beta\pi_A + \delta\pi_G + \eta\pi_T) & \eta\pi_T \\ \gamma\pi_A & \varepsilon\pi_G & \eta\pi_C & -(\gamma\pi_A + \varepsilon\pi_G + \eta\pi_C) \end{bmatrix}$$

where a moment's inspection reveals that if $\pi_A = \pi_C = \pi_G = \pi_T = 0.25$ and $\alpha = \beta = \dots = \eta$, we recover the Jukes-Cantor model as a special case.

The most general model possible for DNA is the General Markov Model, which is exactly as it sounds: Simply a general Markov model on four states.

Matrices like those above can yield well defined probabilities of change from one state (nucleotide) to another given a certain amount of time. This is the basis of constructing the likelihood on a complete tree, as follows: Consider a branch in a tree, with some length ℓ . We may sum over all the possible character states at the ends of this branch and, for each possible combination, calculate the probability of change from the state at one end to the state at the other. Felsenstein created an efficient "pruning algorithm" to do this calculation for a complete tree, over all possible internal character states (in the case of DNA, nucleotides) in the tree, by only keeping track of four possible states at each node, representing the maximum possible likelihood of character state assignments in its descendants (Felsenstein, 1973). The branch lengths and the model parameters (α , β etc. above) directly affect the likelihood. Thus an ML search for the best tree is a process that should in principle search over all possible sets of branch lengths, and all possible sets of parameter values: The computational task is immense.

There are hundreds of possible matrices for DNA, and more of course for amino acids or codons. The decision on which model to choose is often a difficult one, and various approaches have been devised to help. Early in the relatively short history of maximum likelihood applied to phylogenetic inference, likelihood ratio tests were used to compare nested models. The JC model above is a special case of the GTR, but they differ in the number of free parameters (JC can be considered as having just 1 parameter, and GTR 9: They differ by 8 free parameters). Twice the difference in log-likelihoods is known to be χ^2 -distributed, with degrees of freedom equal to the difference in the number of their free parameters, so this method has been used successfully to guide the researcher through a decision tree in the well-cited program, ModelTest (Posada, 2008). This approach has its limitations though, and a standard scheme is now to determine for a given tree the Akaike Information Criterion (AIC) score or one of its derivations (AICc, BIC) to give for each model a simple score, and the researcher may then choose that which has the best score. This avoids the computational complexity of comparing every possible pair of nested models, and enables comparison of models that are not nested.

In general constructing a tree from an algorithm is considerably faster than searching through tree space, taking in the order of $O(n^3)$ operations over all, with n taxa. To search the entirety of tree space is prohibitive: The number of trees increases with n , the number of taxa, at a rate approaching n^n (exact numbers below); hence for all but the most moderately sized examples, we must use heuristics.

Searching Tree Space

The Fitch and Sankoff algorithms solve the "Small Parsimony Problem" – Finding the minimum number of changes that are required to explain an alignment on a tree. The related "Large Parsimony Problem" is that of finding that tree that has the best score, out of all $(2n - 3)!!$ possible rooted binary trees on n taxa (though in fact, as we are regarding the substitutions as equally costly in either direction, we can treat the tree as unrooted, and there are "only" $(2n - 5)!!$ of those). Similarly if the score function is likelihood we are faced with the corresponding problem: find the tree (with branch lengths and other model parameters) that has the highest likelihood of generating the data we observed.

The Large Parsimony Problem, and the tree search problem in general, is solved using local search, typically hill-climbing or simulated annealing. The starting tree is usually constructed quickly using Neighbor-Joining (NJ) or similar. The hill-climbing search operates by perturbing the current tree in some well-defined way and calculating score of the perturbed tree. Hill-climbing is relatively fast among local searches, but it has the obvious problem that if the landscape of the search space has multiple peaks (optima) then it can become stuck in one of them that is not globally optimal. There are several ways that have been developed to get around this issue, including performing multiple hill-climbing searches from different starting points, and using more sophisticated methods such as Simulated Annealing (Van Laarhoven and Aarts, 1987) and Tabu Search (Glover, 1989). Even with these more sophisticated methods it is not possible to guarantee that the best tree has been found.

Note also that our choice of score function, whether it be Maximum Parsimony or Maximum Likelihood, has to be viewed as an approximation to “best” – the assumption is that the tree that optimises the score we choose is the correct one. Many studies find that the most parsimonious tree (that with the lowest Parsimony length) is not the same as the most likely tree (that with the highest conditional probability of having generated the observed sequences). Therefore a common practise is to perform multiple tree searches with different objective functions and hope for consensus.

Time and Reversibility

The presence of a root induces a sense of direction – away from the root and toward the tips, corresponding to increasing time. Rooted trees must be a better representation of evolutionary dynamics, because time does indeed move forward.

However, most – if not all practised – methods that are used in phylogenetic inference assume that there is no inherent direction implied by any of our molecular data: For example there is no sense in which it is quicker or easier to go from A to B than it is from B to A. These models are referred to as time-reversible: The measure of evolutionary work that must be done, or the probability of a given train of sequence evolution, is independent of whether we are moving in the positive or negative time direction.

Thus, the inferred phylogeny we initially create using methods such as maximum likelihood or maximum parsimony (see later), are perforce unrooted. Since rooted phylogenies are so desirable, we must therefore also have methods to determine where the root is.

Rooting trees can be achieved in several ways, in general based on the idea that sequences have diverged from some kind of mean, and/or, at an approximately constant rate (see later). If all the sequences have evolved at approximately the same rate as each other, then we can expect them to be approximately the same distance from their common ancestor, which would, if the branches of the tree accurately reflect evolutionary time, be in the middle of the tree in a well defined way. A simple approach to rooting the tree then is to find the point on the tree that corresponds to the average of the half-way points between every pair of leaves.

Bayesian Methods and Markov Chain Monte Carlo

Thomas Bayes’ Theory has become a central tool in molecular phylogenetics as it enables the use of prior information to be correctly incorporated into a likelihood estimation. In phylogenetics, Bayes’ Theorem takes the form of priors on model components such as the set of possible phylogenetic trees, prior probabilities of particular sequence evolution models or their parameters, or on certain dates of divergence.

For example we might have some estimate on the probability distribution of phylogenetic trees on a taxon set: It could be that “all trees are equally likely” prior to our analysis (a model that has no biological realism it must be said), or that trees arise under a standard birth/death process; the Yule model (Yule, 1924). These priors modify the posterior probability that the data arose from particular models, and enable researchers to make much more powerful conclusions about the phylogenetic relationships under study, including estimates of population size, divergence times based on fossil calibrations, and selection.

Bayesian inference has come to the forefront of phylogenetic inference through such programs as MrBayes (Huelsenbeck and Ronquist, 2001) and BEAST (Drummond *et al.*, 2012). These programs perform Markov chain Monte Carlo analyses and enable researchers to sample from the posterior distribution of trees estimated, and inferences made based on the relative likelihoods of these trees. For example given a posterior distribution of trees, each with their posterior probability, a weighted average can be formed of the overall age of the tree.

What can go Wrong

Given that there are so many components to inferring phylogenies – and the difficulty in verifying that a given phylogenetic inference is even correct – it’s no surprise that there is a veritable cornucopia of things that can go wrong. Here is a non-exhaustive list, in no particular order:

1. *Alignment error*: It is entirely possible that the alignment is wrong. Many authors attempt to avoid this possibility by removing areas of the alignment that are in particular doubt (highly “gappy” loci for example) or, less commonly, by performing analyses with different plausible alignments.
2. *Bad calibration (mutation rates)*: Good fossil records can afford useful prior probability distributions on divergence dates, but it is known that the choice of prior can have a significant effect on the accuracy of divergence time estimations in phylogenetics. Suggested ways around this problem include testing over a range of priors, more stringent screening of calibration points, and sub-sampling from the fossil calibration points to gauge sensitivity of the phylogenetic estimation.
3. *Covariation model*: Our most basic assumption is that sites evolve independently on a sequence, and we can in general get away with this. However there are cases where this assumption is violated and it has an effect on our tree estimation.
4. *Heuristics have no guarantees*: It is worth remembering that heuristic methods have no guarantee of producing an optimal solution. It’s also known that there can be many cases where even our most favoured phylogenetic scoring systems (parsimony, Likelihood) may well have multiple optima for the same data set: there can easily be cases where more than one tree has an optimal likelihood or parsimony length. This not only means that heuristic search methods can become accidentally stuck in

- the wrong places (globally optimal or not), but also that even if all the globally optimal solutions could be located, we would still have the troublesome task of choosing the best among equals.
5. *Incomplete burn-in for MCMC*: Markov chain Monte Carlo requires a burn-in to escape poor areas of the likelihood landscape and navigate to those areas that contribute significantly to the likelihood mass of phylogenies that could have given rise to our observed data. It is entirely possible – in fact it may be common – that burn-in proceeds in jumps, with long periods of proposed moves during which there is no major change in parameter values, followed later by location of new plateaus in “better” regions of tree space (Nylander *et al.*, 2007).
 6. *It's not a tree after all*: As mentioned above, it's possible that by combining phylogenetically informative sequences from different loci, such as genes, we are combining incongruent trees: In such cases the conflicting trees should be reconciled with a single species tree; often through consensus methods.
 7. *Long branch attraction*: Long branches give sequences a chance to converge towards each other simply by chance. Consider a tree in which all but two branches are short, yet the two long ones are not each other's closest relatives. Under such circumstances the two sequences at their leaves may look more like each other simply by dint of the other sequences being less diverged. This phenomenon is known as long branch attraction.
 8. *Model mis-specification*: Getting the underlying model “wrong” runs the considerable risk that the tree is also incorrect (Jermini *et al.*, 2006).
 9. *Rates across sites*: individual sites may evolve under different conditions and so modelling them as though they all evolve at the same rate is an error.
 10. *Saturation of sites*: The alphabet of available character states is very restricted: In DNA and RNA it is only 4 states, so accumulating many changes among those states renders sequences effectively random with respect to each other. If the steady state distribution of nucleotide frequencies is uniform (e.g., if the molecular evolution model is doubly stochastic) then the expected distance between sequences will tend to 0.75 as time increases.
 11. *Software error*: Testing bioinformatics software is often complex and difficult, because testing is ideally done using an “oracle” – A ground truth against which we can test, and there are few such oracles in bioinformatics. While the software currently forming the standard set of tools for phylogenetic inference is in general quite well established and tested by its large user base, it is wise to exert caution when using new software, which may not have had such rigorous testing.

Further Reading

The interested reader is directed to some excellent texts and a wealth of publications in the scientific literature. Joe Felsenstein's “Inferring phylogenies” (Felsenstein, 2004) remains a classic work that is highly readable and yet comprehensive. Nei and Kumar's “Molecular Evolution and Phylogenetics” (Nei and Kumar, 2000) is another detailed work that goes into detail of molecular evolution in amino acids and in nucleotides, discusses phylogenetic inference in far more detail than can be achieved in this short article, and includes material on population genetics. Page and Holmes' “Molecular Evolution: A Phylogenetic Approach” (Page and Holmes, 1998) is a slightly earlier work that remains informative, beginning with a focus on genome evolution as a whole, and the how function relates to evolution. Wen-Hsiung Li's simply titled “Molecular Evolution” (Li, 1997) is from around the same time, and while it also discusses molecular phylogenetics spends a great deal more time on different mechanisms of molecular evolution. Moving into the more mathematical side, Olivier Gascuel's edited book “Mathematics of Evolution and Phylogeny” (Gascuel, 2005) touches on a variety of topics that were not covered here (as well as most that were), including the elegant Hadamard Conjugation method of phylogenetic analysis, mixture models, and even reconstruction of phylogenies from distances estimated by genome rearrangements. Mike Steel and Charles Semple have produced “Phylogenetics” (Semple and Steel, 2003) – Which is largely mathematical work suitable for the mathematician, and “Phylogeny: Discrete and Random Processes in Evolution” (Steel, 2016) is a further example of enjoyable, yet dense, mathematical writing. Yang's “Molecular Evolution: A Statistical Approach” (Yang, 2014) is a more modern piece that is rigorous and accessible. Finally, Drummond and Bouckaert's “Bayesian Evolutionary Analysis with BEAST” (Drummond and Bouckaert, 2015) is an excellent handbook for this sophisticated program, which is one of the standard tools of phylogenetic inference.

Software

Molecular phylogenetics is impossible without good software. Among the standard tools are, in no particular order, BEAST and MrBayes (Bayesian analysis), PhyML and IQTree (Maximum likelihood methods), PHYLIP, PAUP and MEGA (complete suites of methods).

See also: Algorithms for Strings and Sequences: Multiple Alignment. Stochastic Processes. Comparative and Evolutionary Genomics. Evolutionary Models. Molecular Clock. Phylogenetic Tree Rooting. Tree Evaluation and Robustness Testing. Applications of the Coalescent for the Evolutionary Analysis of Genetic Data. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition. Molecular Mechanisms Responsible for Drug Resistance. Phylogenetic analysis: Early evolution of life. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Gene Duplication and Speciation. Epidemiology: A Review. Identification of Homologs

References

- Bergthorsson, U., Adams, K.L., Thomason, B., Palmer, J.D., 2003. Widespread horizontal transfer of mitochondrial genes in flowering plants. *Nature* 424, 197–201.
- Bryant, D., Moulton, V., 2003. Neighbor-net: An agglomerative method for the construction of phylogenetic networks. *Molecular Biology and Evolution* 21 (2), 255–265.
- Castresana, J., 2000. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Molecular Biology and Evolution* 17, 540–552.
- Cavalli-Sforza, L.L., Edwards, A.W.F., 1967. Phylogenetic analysis: Models and estimation procedures. *American Journal of Human Genetics* 19, 233–257.
- Cavender, J.A., 1978. Taxonomy with confidence. *Mathbio* 40, 271–280.
- Darwin, C., 1859. *On the Origin of Species by Means of Natural Selection, or Preservation of Favoured Races in the Struggle for Life*. London: John Murray.
- Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model of evolutionary change in proteins: Matrices for detecting distant relationships. In: Dayhoff, M.O. (Ed.), *Atlas of Protein Sequence and Structure*, vol. 5. Washington D.C.: National Biomedical Research Foundation, pp. 345–358.
- Drummond, A.J., Bouckaert, R.R., 2015. *Bayesian Evolutionary Analysis with BEAST*. Cambridge University Press.
- Drummond, A.J., Suchard, M.A., Xie, D., Rambaut, A., 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution* 29 (8), 1969–1973.
- Edgar, R.C., 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* 32 (5), 1792–1797.
- Efron, B., Halloran, E., Holmes, S., 1996. Bootstrap confidence levels for phylogenetic trees. *Proceedings of the National Academy of Sciences* 93, 7085–7090.
- Felsenstein, J., 1973. Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Systematic Biology* 22, 240–249.
- Felsenstein, J., 2004. *Inferring Phylogenies*, 2nd ed. Sunderland, Massachusetts: Sinauer Associates.
- Fitch, W.M., 1971. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology* 20, 406–416.
- Gascuel, O., 2005. *Mathematics of Evolution and Phylogenetics*. Oxford University Press.
- Gascuel, O., Steel, M., 2006. Neighbor-Joining revealed. *Molecular Biology and Evolution* 23, 1997–2000.
- Glover, F., 1989. Tabu search – Part 1. *ORSA Journal on Computing* 1, 190–206.
- Harrison, R.G., Larson, E.L., 2014. Hybridization, introgression, and the nature of species boundaries. *Journal of Heredity* 105 (S1), 795–809.
- Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences* 89, 10915–10919.
- Higgins, D.G., Bleasby, A.J., Fuchs, R., 1992. CLUSTAL V: Improved software for multiple sequence alignment. *Computer Applications in the Biosciences* 8, 189–191.
- Huelsenbeck, J.P., Ronquist, F., 2001. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* 17 (8), 754–755. **aug.**
- Jermiin, L.S., Ho, S.Y.W., Ababneh, F., Robinson, J., Larkum, A.W., 2006. The biasing effect of compositional heterogeneity on phylogenetic estimates may be underestimated. 61 (February 2004), pp. 1–22.
- Jermiin, L.S., Poladian, L., Charleston, M.A., 2005. Is the “Big Bang” in animal evolution real? *Science* 310 (5756), 1910–1911.
- Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences* 8 (3), 275–282.
- Katoh, K., Misawa, K., Kuma, K.-i., Miyata, T., 2002. MAFFT: A novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research* 30, 3059–3066.
- Van Laarhoven, P.J.M., Aarts, E.H.L., 1987. *Simulated Annealing: Theory and Applications*. Boston: Reidel.
- Lassmann, T., Sonnhammer, E.L.L., 2005. Kalign – An accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics* 6, 298.
- Li, W.-H., 1997. *Molecular Evolution*. Sinauer Associates Inc.
- Lockhart, P.J., Steel, M.A., Barbrook, A.C., Huson, D.H., Howe, C.J., 1998. A covariotide model describes the evolution of oxygenic photosynthesis. *Molecular Biology and Evolution* 15, 1183–1188.
- Nei, M., Kumar, S., 2000. *Molecular Evolution and Phylogenetics*. Oxford: Oxford University Press.
- Notredame, C., Higgins, D.G., Heringa, J., 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302, 205–217.
- Nylander, J.A., Wilgenbusch, J.C., Warren, D.L., Swofford, D.L., 2007. AWTY (are we there yet?): A system for graphical exploration of MCMC convergence in bayesian phylogenetics. *Bioinformatics* 24 (4), 581–583.
- Page, R.D., Holmes, E.C., 1998. *Molecular Evolution: A Phylogenetic Approach*. John Wiley & Sons.
- Posada, D., 2008. jModelTest: Phylogenetic model averaging. *Molecular Biology and Evolution* 25 (7), 1253–1256.
- Saitou, N., Nei, M., 1987. The Neighbor-Joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* 4 (4), 406–425.
- Sankoff, D., 1975. Minimal mutation trees of sequences. *SIAM Journal of Applied Mathematics* 28, 35–42.
- Schröder, E., 1870. Vier combinatorische Probleme. *Zeitschrift für Mathematik und Physik* 15, 361–376.
- Seemple, C., Steel, M.A., 2003. *Phylogenetics*. vol. 24. Oxford University Press.
- Steel, M., 2016. *Phylogeny: Discrete and Random Processes in Evolution (CBMS-NSF Regional Conference Series)*. SIAM-Society for Industrial and Applied Mathematics.
- Studier, J.A., Keppler, K.J., 1988. A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution* 5, 729–731.
- Tavare, S., 1986. Some probabilistic and statistical problems in the analysis of Dna sequences. *Lectures on Mathematics in the Life Sciences* 17, 57–86.
- Wang H.-C., Spencer M., Susko E., 2006. Testing for covarion-like evolution in protein sequences. (902).
- Waterman, M.S., Joyce, J., Eggert, M., 1991. Computer alignment of sequences. In: Miyamoto, M.M., Cracraft, J. (Eds.), *Phylogenetic Analysis of {DNA} Sequences*. Oxford: Oxford University Press, pp. 59–72.
- Yang, Z., 2014. *Molecular Evolution: A Statistical Approach*. Oxford University Press.
- Yule, G.U., 1924. A mathematical theory of evolution based upon the conclusions of Dr. J.C. Willis, FRS. *Philosophical Transactions of the Royal Society B* 213, 21–87.

Evolutionary Models

David A Liberles, Temple University, Philadelphia, PA, United States

Barbara R Holland, University of Tasmania, Hobart, TAS, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

In evolutionary biology, differences are observed between closely related species, both in their genotype and aspects of their phenotype. Alignments of homologous characters (those descended from a common ancestor), and corresponding phylogenetic trees that show the ancestral relationships between entries in the alignment, are commonly used to characterize the evolutionary process. In moving from sets of homologous characters to characterizations of evolution, explicit models of the evolutionary process are commonly used. These models describe the probabilities of different types of evolutionary changes allowing the most likely trajectory to be identified. The models are frequently structured as Markov processes. Evolutionary models can be applied to data at multiple levels of biological organization, from morphological change, to gene content in genomes, to gene duplication, to genetic sequence data. This article will focus on models for biological sequence evolution.

Multiple sequence alignment is one problem where evolutionary models are particularly useful. Probabilities of different types of sequence change are compared with the probabilities of an insertion or deletion having been introduced (commonly using affine gap penalties) to generate an alignment of sequence characters (DNA or protein) that derived from a common ancestor (Anisimova *et al.*, 2010). There are many methods to infer phylogenetic trees based on multiple sequence alignments, many of these methods are based on explicit models of sequence change. The phylogenetic tree shows the branching pattern of descendent sequences (from species or gene copies within a genome), with internal nodes reflecting ancestral character states. Branch lengths are estimated as well, with units of expected changes per site under the model as the most common units. Inference of sequences at internal nodes is a sub-field of its own, known as ancestral sequence (state) reconstruction. Lastly, models are also used in the search for lineage-specific positive selection, in characterizing the patterns of sequence change on particular branches of a phylogenetic tree (Benner *et al.*, 2007).

Because alignment and phylogenetic analysis both rely on models of the same evolutionary process, one strategy that has been developed, but is not currently widely used, is to simultaneously estimate both. Not only does this enable the consistency of using the same model for the evolutionary process in both steps, but it also enables accounting for alignment uncertainty in tree estimation. *Baliphy* is an example of an approach implemented in a Bayesian statistical framework that uses models to simultaneously estimate alignments and trees (Redelings and Suchard, 2005).

Markov Models

Models of sequence evolution range from phenomenological models that are designed to fit data as well as possible with as few parameters as possible, to mechanistic or generative models that are designed to model the process generating the data. The mechanistic models may not describe data as well as the phenomenological models on interpolated data, but may enable extrapolation to longer evolutionary distances in cases that might be prohibitive for phenomenological models (Liberles *et al.*, 2013). Because mechanistic models are meant to describe the process, there is the expectation that they will have predictive value on data that they were not fit to.

In order to model the evolution of molecular sequences, some key simplifying assumptions are made in almost all cases. A strong assumption that provides great benefits in terms of the tractability of models is that the evolution of different sites in an alignment is independent and happens according to a common process – this is known as the i.i.d. assumption (independent and identically distributed). The second assumption is that sequences evolve according to a Markov process. This means that the evolution of a particular character depends only on its current state rather than on any additional information about the past states.

With these assumptions in place, sequence evolution can be modeled with a continuous time Markov chain (CTMC). In the case of modeling sequence evolution for DNA, amino acids, or codons, the state space is discrete, e.g., for DNA the state space is {A, C, G, T}. The instantaneous rates at which any particular character state substitutes into any other state can be gathered into a rate matrix, Q , sometimes known as the generator matrix. If these rates are in effect for time t then the probability of a transition from state i to state j is given by the ij th entry of the matrix P , where $P = \exp(Qt)$.

Given an edge-weighted phylogenetic tree and a continuous time model given by Q it is possible to calculate the probability of observing any particular site pattern at the tips of the phylogenetic tree in extant sequences (Felsenstein, 2004). This means such models can be used as a basis for likelihood or Bayesian style statistical approaches to determine which choice of tree and edge weights give the highest probability of producing a particular sequence alignment.

A sequence alignment reflects the collection of substitutions that result from mutations. It should be noted that the term *mutation* formally refers to a mutation introduced to a population at frequency $1/N$ in a haploid population. When this mutation fixes (goes to frequency 1), this is referred to as a *substitution*. Inter-specific evolutionary models are designed to model substitution rates and processes. If sequences are evolving neutrally then the rate of mutation and substitution are the same (Kimura, 1983),

but if sequences are under selection then the distinction between mutation and substitution is an important modeling consideration.

Parameterized DNA Models

Standard DNA models (see [Felsenstein \(2004\)](#) for a further discussion of standard models and their original references) started with the Jukes-Cantor model ([Jukes and Cantor, 1969](#)), which has a single rate parameter that describes the equal probability of exchange between all bases. From a statistical perspective, this is a natural model that characterizes the rate of change without any biological assumptions. However, from a biological perspective, this model tends to explain sequence substitution patterns poorly. The Kimura 2-parameter model ([Kimura, 1980](#)) is a standard model in the hierarchy of models that differentiates between transitions (substitutions between purines or between pyrimidines) and transversions (substitutions that convert between purines and pyrimidines). The reason this is natural is that transitions are more likely due to oxidative deamination (at least for C to U changes) and due to polymerase error, and also because of the genetic code, where 2-fold degenerate codons preserve the encoded amino acid with a transition but not a transversion.

Another introduction of model complexity that is similarly nested over the Jukes-Cantor model involved the introduction of unequal base equilibrium frequencies with a single rate parameter to produce the ([Felsenstein, 1981](#)) model. One example of the importance of unequal equilibrium frequencies is the hypothesis of temperature dependence to GC usage because of its effect on DNA melting temperatures ([Groussin and Gouy, 2011](#)). The parameters of K2P and F81 were combined in the HKY model ([Hasegawa et al., 1985](#)). A further relaxation of this involves the GTR model ([Lanave et al., 1984](#); [Tavaré, 1986](#)), where each of the six inter-conversions has an independent rate parameter.

GTR is the most complex model that is time-reversible such that, given the equilibrium base-frequency distribution $\pi = (\pi_A, \pi_C, \pi_G, \pi_T)$, $\pi_i Q_{ij} = \pi_j Q_{ji}$. A further relaxation from six to twelve rate parameters gives the General Markov Model (GMM) ([Barry and Hartigan, 1987](#)), which is not time reversible. This set of standard models is summarized in [Fig. 1](#). Between Jukes-Cantor and GTR for symmetrical matrices and GMM to include non-symmetrical matrices, is a hierarchy of conceivable models ([Huelsenbeck et al., 2004](#)). The property of being time reversible makes models much more efficient to compute with ([Felsenstein, 2004](#)). To use non-time-reversible models, a rooted tree must be used and likelihoods calculated in a different manner ([Boussau and Gouy, 2006](#)). Despite this computational complexity, biologically one does not expect evolution to be time-reversible when the underlying processes are not in equilibrium and the changes are not neutral.

Commonly Used Additional Parameters

Additional parameters are commonly used together with phylogenetic Markov Models for DNA and for proteins. Parameters that change the equilibrium frequencies of the underlying model (particularly for empirical models described below) to fit those of the

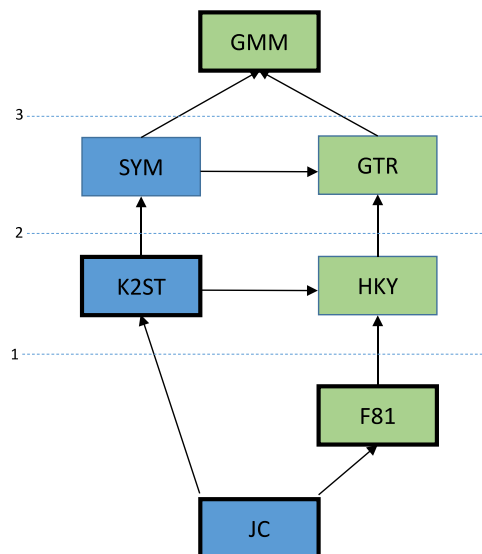


Fig. 1 An overview of common DNA models is shown. Models with equal base-frequencies are shown with a blue background, models with free base-frequencies are shown with a green background. Models with the Lie-Markov property have a bold border. Models above the first dashed line allow a transition/transversion ratio. Models above the second dashed line allow 6 different substitution rates. Models below the third dashed line are time-reversible. Arrows directed from one model to another indicate that the model is a sub-model, i.e., that it is nested within the more complicated model.

dataset (+F) are commonly included as extra parameters. Additionally, some sites are of such (context-dependent) importance in a dataset that they show a substitution rate of zero over the period of evolution over which the data were generated. Modeling a fraction of invariant sites (+I) with substitution rate zero is another common parameter that is added to phylogenetic Markov Models.

More generally, different sites evolve at different rates, both due to variation in the mutation process and due to differential selection on different sites. There have been two main approaches to tackling this: mixture models and partitioned models. The most common application of mixture models is to use the gamma distribution (+G) to model rate heterogeneity across sites (Yang, 1994). A strength of this approach is that a single shape parameter can provide a continuum of models ranging from equal rates to very right-skewed rates. This is most commonly implemented using a discretization of the gamma distribution into four rate categories, although alternative mechanisms of using a gamma distribution are possible. No index parameters are included, i.e., sites are not allocated to particular rate classes, instead all four rate categories are used as a mixture process applied to all sites. Using a gamma distribution is somewhat arbitrary, a slightly more parameter-rich alternative is the free-rate model, which allows a mixture model over an arbitrary set of rates (Yang, 1995; Pagel and Meade, 2004; Mayrose *et al.*, 2005). The other strategy for dealing with heterogeneity is to use prior knowledge of the data, e.g., gene boundaries, positions within a protein structure, and/or codon position, to partition the sequence alignment and then fit different models to each part of the partition (Bull *et al.*, 1993).

Lie-Markov Models

Even with the extensions stated above, most phylogenetic applications of evolutionary models assume that a common process operates over all sites and in all parts of the evolutionary tree. This assumption does not appear to be biologically realistic. In particular, a prediction of such models is that base frequencies should be similar in all species under consideration, but this has been shown empirically not always to be the case (Jayaswal *et al.*, 2011). Codon models have also been used to show that evolution can be a heterogeneous process with evidence for selection acting in some branches but not others.

If one allows the possibility that the evolutionary process may be different in different edges of the phylogeny or differ along edges then this introduces an interesting consideration about how one might want the Markov models in use to behave. If one considers that evolution happens according to some rates Q_1 for some period of time t_1 followed by some rates Q_2 for time t_2 then one can work out the transition probabilities over time $t_1 + t_2$ to be $P = \exp(Q_2 t_2) * \exp(Q_1 t_1)$. A Markov model has the closure property if this matrix P can still be expressed as $P = \exp(Q t)$ for some Q chosen from our model. Many commonly used Markov models, including GTR and HKY, do not have this closure property. Recent work by Sumner *et al.* (2012) has developed a hierarchy of evolutionary models, known as Lie-Markov models, that are defined by having this closure property.

Model Selection

As discussed above, a common approach to add biological realism is to differentiate between types of nucleotide substitutions. For example, Huelsenbeck *et al.* (2004) consider all 203 possible partitions of relative rates for time-reversible models, reflecting all possible models of this class. Another approach is to consider that there are different classes of bases (e.g., AG and CT) and then use the resulting symmetries to find appropriate constraints on substitution rates (Fernández-Sánchez *et al.*, 2015).

Given the wealth of possibilities for parameterized models a problem that arises is how best to choose a model to fit a particular dataset. The best inference comes when a model is appropriately complex, with all of the necessary parameters, but without extra parameters that do not contribute to data fit and/or that can cause model mis-specification (Steel, 2005). An approach, initially proposed by Posada and Crandall (2001), was to perform likelihood ratio tests of nested models (see Fig. 1). More recently the field has moved to prefer information criterion biased approaches such as the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) (Posada and Buckley, 2004). The AIC score of a model depends on both the likelihood and on the number of free parameters in the model, $AIC = -2 \ln L + 2K$ where $\ln L$ is the log likelihood and K is the number of parameters. The BIC = $-2 \ln L + K \ln n$, where n is the sample size (e.g., the length of the alignment) this penalizes additional parameters more strongly than the AIC. For both AIC and BIC, models with smaller scores are preferred. A strong advantage of both the AIC and BIC is that they do not rely on models being nested. For partitioned models the number of possible models makes it prohibitive to check the AIC or BIC of all models and it is common to take a heuristic approach to finding models that fit well instead (Lanfear *et al.*, 2012).

Empirical Models

An alternative to using parameterized models to fit substitution data is to pre-calculate empirical substitution matrices from large datasets. This approach is commonly applied when working with amino acid as opposed to DNA data. One reason to do this is that while a maximal time-reversible DNA matrix can have up to 6 rate parameters, an equivalent matrix for proteins could have 190 rate parameters. Furthermore, some transitions between rare amino acids or those that are separated by two or three nucleotide changes in the genetic code occur too rarely to be able to estimate well from typical datasets.

The first instance of an empirical amino-acid model was produced by Dayhoff *et al.* (1978), they used closely related aligned proteins on a well-established phylogeny to estimate the rate at which particular amino acids substitute for other amino acids. Similar ideas were used by later authors although they tended to use much larger databases of curated alignments and to be more computationally intensive as they incorporated tree-estimation simultaneously with estimation of the empirical substitution parameters (e.g., JTT (Jones *et al.*, 1992), WAG (Whelan and Goldman, 2001), LG (Le and Gascuel, 2008), see also Zoller and Schneider (2013)). WAG included the innovation of maximum likelihood parameter estimation, while LG included the innovation of gamma distributed substitution rates.

A number of alternative matrices for either specific datasets, for specific attributes of protein structures, or statistically defined matrices have been created. A particular example of this is the set of empirical matrices for proteins that are partitioned by secondary structure and solvent accessibility (Koshi and Goldstein, 1995).

Codon Models

There is independent information that is encompassed in DNA models and in protein models and both sources of information can be harnessed when using codon models. Codon models can be built in three different broad types. The first type is to adopt a nucleotide substitution model such as HKY85 to describe the substitution process at the DNA level, with a genetic code matrix embedded on top of it. This genetic code matrix typically has three values, 1 for synonymous changes that involve a single nucleotide change, 0 for all changes that involve more than a single nucleotide change, and free parameter ω (dN/dS) for nonsynonymous changes that involve a single nucleotide change. This is motivated by the expectation that selective constraints on nucleotide changes that change the amino acid will be different from selective constraints on nucleotide changes that do not, leading to different rates. In the Goldman-Yang model, there are independent DNA process at each codon position, such that a different rate matrix is estimated for the 1st, 2nd, and 3rd codon positions (Goldman and Yang, 1994). In the Muse-Gaut model (Muse and Gaut, 1994), there is a single DNA level process that applies equivalently to all three positions, reflected in a common substitution matrix between them.

A second type of codon model also involves free parameters estimated directly from data, but stems from an attempt to model the underlying population genetics. In Halpern-Bruno style models (also called mutation-selection models), the probability of introducing a mutation to a population is reflected by the product of the mutation rate and the population size (Halpern and Bruno, 1998). For a mutation that is introduced, the fixation probability reflects the probability of going from frequency $1/N$ ($1/2N$ in a diploid population) to fixation. This second probability depends upon the population size and the selective coefficient for the change, which in linear space is $(f'/f) - 1$, where f' is the fitness of the new amino acid and f is the fitness of the original amino acid, sampled from a vector of 20 amino acid fitnesses (19 parameters).

The third class of models is a 64×64 (61×61 when stop codons are excluded) empirical model calculated in a similar manner to amino acid substitution models. An early instance of such a model that has been constructed comes from Schneider *et al.* (2005). This class of model explicitly contains averaged data on amino acid properties that is lacking from the ω -based models.

Recent work has enabled model selection across classes of models (including nucleotide, codon, and amino acid models), but has only included the first class of codon models in the set of models that are available for testing (ModelOMatic; Whelan *et al.* (2015)).

CAT Models

One class of models that is currently widely held as the state of the art for amino acid models is the CAT models (Lartillot and Philippe, 2004), based upon its ability to fit data well for the number of parameters. This model class includes a distribution of equilibrium frequencies across sites that describes the propensities for different amino acids to occur at different positions within a protein, bearing some similarity to a mixture of F81 models in protein state space. The number of categories is either defined a priori or by a Dirichlet distribution. To account for temporal heterogeneity at a site, due to epistasis (the non-independence of the evolutionary process at each site), the CAT-BP model was introduced that allows for class switching at a site (Blanquart and Lartillot, 2008).

Covariation Models

The models described so far (besides CAT-BP and some implementations of codon models) all assume that the same evolutionary process applies to all sites in an alignment on all lineages of a phylogenetic tree. While they can be applied as heterogeneous process models with discrete breaks in process at nodes of phylogenetic trees, a class of models called covariation models describes explicit rate shifting parameters (Miyamoto and Fitch, 1995). The simplest model, Tuffley-Steel, includes a parameter for shifting between a particular rate class and an invariant site (Tuffley and Steel, 1998). Galtier described a model where sites with different rate classes can shift between each other (Galtier and Jean-Marie, 2004). Roger generalized this as the General Covariation Model, with parameters to shift between rate classes and into and out of invariant states (Wang *et al.*, 2007). Overall, this modeling

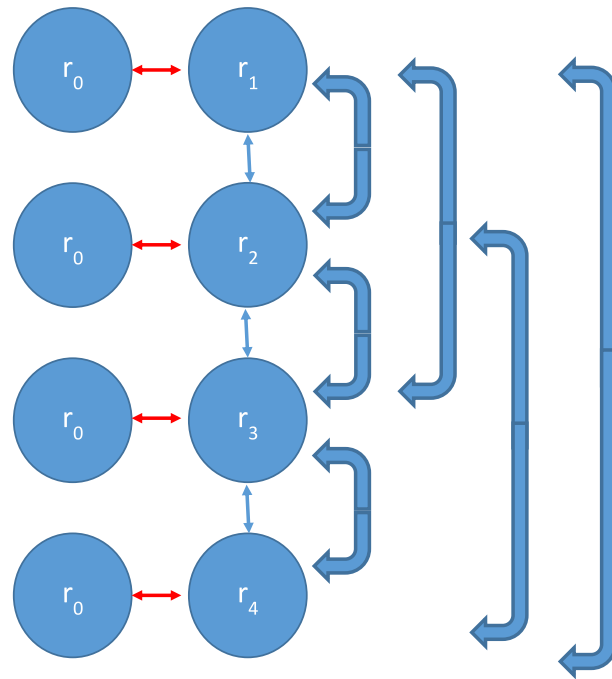


Fig. 2 An overview of covarion models is shown. r_0 states on the left reflect temporally invariant states that are accessible from states with different rates in models that incorporate this (red arrows) “Tuffley-Steel” process. Blue arrows depict the shifts between states with different rates, shown as four distinct gamma categories here. The general covarion model allows both the red and blue arrow states to occur.

framework can be applied as a general Markov process framework for temporal heterogeneity at a site. **Fig. 2** shows a summary of the modeling framework, including transitions for a site between a rate category and invariant states as well as between rate categories. Covarion processes are expected to account for the role of epistasis in enabling temporal shifts in the evolutionary processes at individual sites.

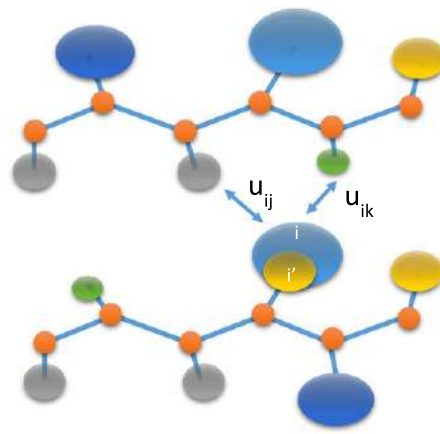
Physicochemical Models of Substitution

An alternative to using empirical matrices to characterize amino acid substitution is to use metrics of the physical chemical distance between amino acid side chains. The Grantham matrix ([Grantham, 1974](#)), the most widely used of such matrices, is calculated as a normalized RMSD between side chain atomic composition, polarity, and volume. It is most commonly used to weight amino acid differences or in metrics that examine rates of radical and conservative amino acid changes based upon Grantham matrix values. It can be extended to use as a full Markov model for phylogenetic purposes as well.

Empirical Contact Models

Another alternative to a traditional substitution matrix is to use a contact potential matrix. [Miyazawa-Jernigan \(1985\)](#) is a matrix that reflects the average pairwise potential of two amino acids in contact at a given distance based upon a statistical analysis of protein structures (see **Fig. 3**). This treatment enables explicit consideration of protein structure, that is not treated in other models that have been discussed. It does not consider either angular or distance information between residues that can be important in characterizing energies of interaction. To use this matrix for substitution, the difference in the sum of contacts between the original and substituted amino acids in the protein structure is used to generate an expected energetic change in the folding energy. Assuming that the strongest selective pressures are reflected in the fraction folded of a molecule based upon this energy, the fitness difference (selective pressure) associated with a change is then calculated by placing a Boltzmann distribution in a function to calculate the probability of fixation of the change.

Bastolla has generated updated contact potential matrices and most recently, a mean field model for use in ancestral sequence reconstruction ([Arenas *et al.*, 2017](#)). [Robinson *et al.* \(2003\)](#) introduced an early codon model with structure dependencies and novel statistics to enable computation. [Kleinman *et al.* \(2010\)](#) and [Grahnen *et al.* \(2011\)](#) have generated fuller structure-aware models for amino acid substitution that include distance and angular terms, but these methods are slow and have not performed well in characterizing individual site probabilities of substitution in actual alignments when compared with the best performing models (e.g., CAT).



$$\Delta u_{i \rightarrow i'} = (u_{i'j} - u_{ij}) + (u_{i'k} - u_{ik})$$

Fig. 3 An overview of the use of a contact potential matrix for substitution is shown. Residues j and k interact with residue i . On mutation from i to i' , $\Delta u_{i \rightarrow i'}$ reflects the corresponding change in potential resulting from the two underlying physical interactions in the protein structure.

Conclusions and Future Directions

Markov models are widely used in biological sequence analysis, as described in the approaches to date. DNA models are fairly good approximations to underlying biological processes, particularly for neutrally evolving sites. One ongoing area of research with DNA models is their implementation as time heterogeneous models that can account for changes in base frequencies over time. Codon models come in two main varieties, those built with a genetic code model over a DNA model and those built as empirical codon models. As with amino acid models, codon models approximate the complexity of selection on protein sequences, structures, and functions (see [Chi and Liberles \(2016\)](#) for a recent discussion of this). An incomplete synthesis involves incorporating structural or advanced statistical techniques to build better models for proteins. Markov models of all of these varieties are an important trajectory in bioinformatics.

Acknowledgments

We thank Peter Chi, Jeremy Sumner, Claudia Weber, and Josh Schraiber for helpful comments on both the scientific content and the general readability of this work. DAL acknowledges support from the US National Science Foundation grant DBI-1515704, to which this article will contribute to the broader impact mission.

See also: Applications of the Coalescent for the Evolutionary Analysis of Genetic Data. Gene Duplication and Speciation. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Molecular Clock. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Population Genetics. Stochastic Processes

References

- Anisimova, M., Cannarozzi, G.M., Liberles, D.A., 2010. Finding the balance between mathematical and biological optima in multiple sequence alignment. *Trends Evol. Biol.* 2, e7.
- Arenas, M., Weber, C.C., Liberles, D.A., Bastolla, U., 2017. ProtASR: An evolutionary framework for ancestral protein reconstruction with selection on folding stability. *Syst. Biol.* syw121. doi:10.1093/sysbio/syw121.
- Barry, D., Hartigan, J.A., 1987. Statistical analysis of hominoid molecular evolution. *Stat. Sci.* 2, 191–210.
- Benner, S.A., Sassi, S.O., Gaucher, E.A., 2007. Molecular paleoscience: Systems biology from the past. *Adv. Enzymol. Relat. Areas Mol. Biol.* 75, 1–132.
- Blanquart, S., Lartillot, N., 2008. A site- and time-heterogeneous model of amino acid replacement. *Mol. Biol. Evol.* 25 (5), 842–858.
- Boussau, B., Gouy, M., 2006. Efficient likelihood computations with nonreversible models of evolution. *Syst. Biol.* 55 (5), 756–768.
- Bull, J.J., Huelsenbeck, J.P., Cunningham, C.W., Swofford, D.L., Waddell, P.J., 1993. Partitioning and combining data in phylogenetic analysis. *Syst. Biol.* 42 (3), 384–397.
- Chi, P.B., Liberles, D.A., 2016. Selection on protein structure, interaction, and sequence. *Protein Sci.* 25 (7), 1168–1178.
- Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model of evolutionary change in proteins. In: Dayhoff, M.O., (Ed.), *Atlas of Protein Sequence and Structure* National Biomedical Research Foundation, Washington DC, 5(3), 345–352.
- Felsenstein, J., 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.* 17 (6), 368–376.
- Felsenstein, J., 2004. *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates, Inc.
- Fernández-Sánchez, J., Jeremy, G.S., Jarvis, P.D., Woodhams, M.D., 2015. Lie Markov models with purine/pyrimidine symmetry. *J. Math. Biol.* 4, 855–891.

- Galtier, N., Jean-Marie, A., 2004. Markov-modulated Markov chains and the covarion process of molecular evolution. *J. Comput. Biol.* 11 (4), 727–733.
- Goldman, N., Yang, Z., 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol. Biol. Evol.* 11, 725–736.
- Grahn, J.A., Nandakumar, P., Kubelka, J., Liberles, D.A., 2011. Biophysical and structural considerations for protein sequence evolution. *BMC Evol. Biol.* 11, 361.
- Grantham, R., 1974. Amino acid difference formula to help explain protein evolution. *Science* 185 (4154), 862–864.
- Groussin, M., Gouy, M., 2011. Adaptation to environmental temperature is a major determinant of molecular evolutionary rates in archaea. *Mol. Biol. Evol.* 28 (9), 2661–2674.
- Halpern, A.L., Bruno, W.J., 1998. Evolutionary distances for protein-coding sequences: Modeling site-specific residue frequencies. *Mol. Biol. Evol.* 15 (7), 910–917.
- Hasegawa, M., Kishino, H., Yano, T., 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.* 22 (2), 160–174.
- Huelsenbeck, J.P., Larget, B., Alfaro, M.E., 2004. Bayesian phylogenetic model selection using reversible jump Markov chain Monte Carlo. *Mol. Biol. Evol.* 21 (6), 1123–1133.
- Jayaswal, V., Jermini, L.S., Poladian, L., Robinson, J., 2011. Two stationary nonhomogeneous Markov models of nucleotide sequence evolution. *Syst. Biol.* 60 (1), 74–86.
- Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.* 8 (3), 275–282.
- Jukes, T.H., Cantor, C.R., 1969. *Evolution of Protein Molecules*. New York, NY: Academic Press, pp. 21–132.
- Kimura, M., 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* 16 (2), 111–120.
- Kimura, M., 1983. *The Neutral Theory of Molecular Evolution*. Cambridge: Cambridge University Press.
- Kleinman, C.L., Rodrigue, N., Lartillot, N., Philippe, H., 2010. Statistical potentials for improved structurally constrained evolutionary models. *Mol. Biol. Evol.* 27 (7), 1546–1560.
- Koshi, J.M., Goldstein, R.A., 1995. Context-dependent optimal substitution matrices. *Protein Eng.* 8 (7), 641–645.
- Lanave, C., Preparata, G., Saccone, C., Serio, G., 1984. A new method for calculating evolutionary substitution rates. *J. Mol. Evol.* 20 (1), 86–93.
- Lanfear, R., Calcott, B., Ho, S.Y., Guindon, S., 2012. Partitionfinder: Combined selection of partitioning schemes and substitution models for phylogenetic analyses. *Mol. Biol. Evol.* 29 (6), 1695–1701.
- Lartillot, N., Philippe, H., 2004. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Mol. Biol. Evol.* 21 (6), 1095–1109.
- Le, S.Q., Gascuel, O., 2008. An improved general amino acid replacement matrix. *Mol. Biol. Evol.* 25 (7), 1307–1320.
- Liberles, D.A., Teufel, A.I., Liu, L., Stadler, T., 2013. On the need for mechanistic models in computational genomics and metagenomics. *Genome Biol. Evol.* 5 (10), 2008–2018.
- Mayrose, I., Friedman, N., Pupko, T., 2005. A gamma mixture model better accounts for among site rate heterogeneity. *Bioinformatics* 21 (Suppl. 2), 151–158.
- Miyamoto, M.M., Fitch, W.M., 1995. Testing the covarion hypothesis of molecular evolution. *Mol. Biol. Evol.* 12 (3), 503–513.
- Miyazawa, S., Jernigan, R.L., 1985. Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation. *Macromolecules* 18, 534–552.
- Muse, S.V., Gaut, B.S., 1994. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Mol. Biol. Evol.* 11 (5), 715–724.
- Pagel, M., Meade, A., 2004. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. *Syst. Biol.* 53 (4), 571–581.
- Posada, D., Buckley, T.R., 2004. Model selection and model averaging in phylogenetics: Advantages of Akaike information criterion and Bayesian approaches over likelihood ratio tests. *Syst. Biol.* 53 (5), 793–808.
- Posada, D., Crandall, K.A., 2001. Selecting the best-fit model of nucleotide substitution. *Syst. Biol.* 50 (4), 580–601.
- Redelings, B.D., Suchard, M.A., 2005. Joint Bayesian estimation of alignment and phylogeny. *Syst. Biol.* 54 (3), 401–418.
- Robinson, D.M., Jones, D.T., Kishino, H., Goldman, N., Thorne, J.L., 2003. Protein evolution with dependence among codons due to tertiary structure. *Mol. Biol. Evol.* 20 (10), 1692–1704.
- Schneider, A., Cannarozzi, G.M., Gonnet, G.H., 2005. Empirical codon substitution matrix. *BMC Bioinform.* 6, 134.
- Steel, M., 2005. Should phylogenetic models be trying to ‘fit an elephant’? *Trends Genet.* 21, 307–309.
- Sumner, J.G., Fernández-Sánchez, J., Jarvis, P.D., 2012. Lie Markov models. *J. Theor. Biol.* 298, 16–31.
- Tavaré, S., 1986. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. *Lectures on Mathematics in the Life Sciences* 17, American Mathematical Society, pp. 57–86.
- Tuffley, C., Steel, M., 1998. Modeling the covarion hypothesis of nucleotide substitution. *Math. Biosci.* 147 (1), 63–91.
- Wang, H.C., Spencer, M., Susko, E., Roger, A.J., 2007. Testing for covarion-like evolution in protein sequences. *Mol. Biol. Evol.* 24 (1), 294–305.
- Whelan, S., Allen, J.E., Blackburne, B.P., Talavera, D., 2015. ModelOMatic: Fast and automated model selection between RY, nucleotide, amino acid, and codon substitution models. *Syst. Biol.* 64 (1), 42–55.
- Whelan, S., Goldman, N., 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol.* 18 (5), 691–699.
- Yang, Z., 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *J. Mol. Evol.* 39 (3), 306–314.
- Yang, Z., 1995. A space-time process model for the evolution of DNA sequences. *Genetics* 139 (2), 993–1005.
- Zoller, S., Schneider, A., 2013. Improving phylogenetic inference with a semiempirical amino acid substitution model. *Mol. Biol. Evol.* 30 (2), 469–479.

Molecular Clock

Cara Van Der Wal, University of Sydney, Sydney NSW, Australia and Australian Museum Research Institute, Australian Museum, Sydney NSW, Australia

Simon YW Ho, University of Sydney, Sydney NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Our understanding of biological patterns and processes is greatly improved by knowledge of evolutionary timescales. The inference of these timescales has traditionally been based on evidence from the fossil record, which provides an indication of when species diverged and when key biological innovations arose. Unfortunately, the fossil record is very incomplete for many branches of the Tree of Life. Groups such as bacteria and various invertebrates have left scant traces, despite the persistence of their lineages through many millions of years. Additionally, fossil evidence is usually unable to resolve the timing of very recent events, such as demographic changes occurring within species or the transmission of pathogens across their host populations (Arbogast *et al.*, 2002).

When the fossil record is uninformative about evolutionary and demographic timescales, we require temporal information from another source. The answer to this problem arrived in the 1960s when the idea of a 'molecular evolutionary clock' was proposed by Zuckerkandl and Pauling (1962, 1965). Molecular clocks describe the relationship between evolutionary rate and time, with the simplest clock model assuming that the rate of molecular evolution is constant across species. This allows us to use phylogenetic analyses of genetic data to estimate evolutionary and demographic timescales (Fig. 1). The most widely used forms of data for this purpose are nucleotide and amino acid sequences. To give estimates in terms of absolute time, however, molecular clocks need to be calibrated using independent evidence about the timing of one or more evolutionary divergences in the phylogeny.

Molecular clocks have been used to provide insights into a number of significant evolutionary events, ranging from human prehistory to the deep divergences among the kingdoms of life. They are now being applied to genome-scale data sets that comprise thousands of genes from dozens to hundreds of taxa (Tong *et al.*, 2016). For example, evolutionary timescales have been inferred using large data sets from birds (Jarvis *et al.*, 2014), flowering plants (Foster *et al.*, 2017), and insects (Misof *et al.*, 2014). The methods underpinning molecular-clock analyses continue to evolve, which is imperative if we are to make the most of the growing wealth of genomic data.

The objective of this article is to provide an overview of the molecular clock. First, we will present a brief history of the molecular-clock hypothesis and its early applications. Second, molecular-clock methods and models will be described, as will the challenges to their implementation. Following a case study on the origins of lobsters, we will discuss the role of molecular clocks in the genomic era.

Background

The Origins of the Molecular Clock

Zuckerkandl and Pauling (1962) were the first to propose the idea of a molecular clock. In their analysis of globin proteins, they assumed a constant evolutionary rate in order to estimate the timing of past gene duplications in vertebrates. They found 18 amino acid differences separating human and horse, and estimated the evolutionary rate of this protein by assuming that these two species split from each other 100–160 million years ago. This evolutionary rate was then extrapolated to the globin genes of other animals to show that humans and gorillas split from each other approximately 11 million years ago, and that different globin genes first diverged during the late Precambrian (Zuckerkandl and Pauling, 1962).

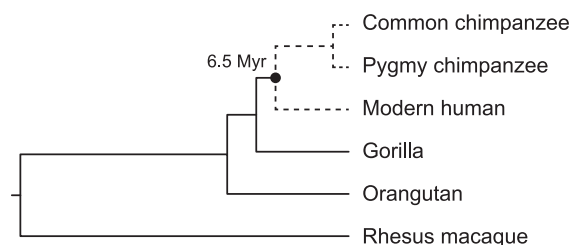


Fig. 1 Simple illustration of molecular dating. The phylogenetic tree depicts the evolutionary relationships among six primate species, inferred using genetic data from each species. Fossil evidence indicates that modern humans last shared a common ancestor with chimpanzees about 6.5 million years ago. This 'calibration' allows us to infer the rate of molecular evolution within the descendent branches (dashed lines). If this rate is assumed to remain constant throughout the tree (known as a 'strict molecular clock'), we can estimate the timing of all other divergence events.

Studies of other proteins throughout the 1960s provided further evidence of constant, clocklike evolution, leading [Zuckerlandl and Pauling \(1965\)](#) to put forward the term 'molecular evolutionary clock'. Promising to be a valuable tool in biological research, the power of the molecular clock was soon demonstrated when [Sarich and Wilson \(1967\)](#) inferred the evolutionary timescale of hominids and related primates.

The molecular-clock hypothesis relies on the idea that rates of molecular evolution, as seen in mutations in protein or DNA sequences over time, are relatively constant between lineages. The occurrence of mutations is often modelled as a Poisson process, reflecting the stochastic nature of the molecular clock. Although individual mutations do not occur at strictly regular intervals, the genetic difference between any two species will increase monotonically over time. If the evolutionary rate has been constant, then genetic divergence will be directly proportional to the time since the two species last shared a common ancestor. This relationship allows evolutionary timescales to be inferred from genetic data, provided that we know the rate of molecular evolution.

When initially proposed, the notion of a constant rate of molecular evolution created controversy. A number of researchers, including the prominent palaeontologist [Simpson \(1964\)](#), viewed the molecular clock with scepticism. It seemed implausible that a simple statistical model could describe a process as seemingly complex as that of molecular evolution. In the late 1960s, however, new ideas about molecular evolution would provide some support for the concept of a molecular clock.

Neutral and Nearly Neutral Theories of Molecular Evolution

Alongside the early criticisms of the molecular clock, there was growing evidence of high evolutionary rates in a range of proteins. This suggested that a substantial proportion of amino acid mutations have negligible impacts on fitness. In response to this evidence, [Kimura \(1968\)](#) proposed the neutral theory of molecular evolution. This theory states that many mutations do not have a measurable impact on the organism's fitness, so that their frequencies in the population vary stochastically by genetic drift rather than being driven by natural selection. These neutral mutations might represent the replacement of amino acids by other amino acids with similar biochemical properties. In protein-coding genes, some nucleotide mutations do not result in any changes to the encoded protein, so their effects on fitness are likely to be very small.

The neutral theory of molecular evolution predicts that evolutionary rates, measured on a per-generation basis, are constant across lineages ([Kimura, 1968](#)). However, generation lengths vary between species, meaning that we expect a generation-time effect: molecular evolution occurs more quickly per unit of time in species with shorter generations. For example, rodents express a higher rate of evolution than primates. Such generation-time effects have been identified in a broad range of organisms.

At the end of the 1960s, some studies found that the rate of protein evolution was independent of generation time. This contradicted the expectations of the neutral theory, leading [Ohta \(1972, 1973\)](#) to propose the nearly neutral theory of molecular evolution. In this model, most mutations are assumed to have a small effect on fitness. In contrast with the neutral theory, the nearly neutral theory gives a central role to the effect of population size. Natural selection is effective at removing harmful mutations in large populations, but the process of genetic drift dominates in small populations. However, there are more opportunities for mutations to arise in large populations. [Ohta \(1972\)](#) predicted that these two effects would offset each other, with the net effect being a constant evolutionary rate per unit of time across species.

Molecular Clocks in Practice

In the decades following the proposal of the molecular clock, genetic data have been used to estimate the evolutionary timescales of various parts of the Tree of Life. Among the most conspicuous studies were those of the deep divergences among metazoan phyla. The date estimates from these analyses pointed to a much longer timescale of evolution than indicated by the fossil record. In conflict with the idea of an explosive evolutionary radiation in the Cambrian period, the basal metazoan divergences were estimated to have occurred much deeper in the Precambrian (e.g., [Runnegar \(1982\)](#), [Bromham *et al.* \(1998\)](#)). These striking discrepancies between genetic and fossil-based estimates of evolutionary timescales led many to question the reliability of molecular clocks.

There has also been abundant evidence of rate variation across lineages, contradicting the simplistic assumption of clocklike evolution. These challenges to rate-homogeneous models have driven the development of 'relaxed' molecular clocks, the first of which appeared in the late 1990s. These clock models are able to relax the assumption of rate homogeneity among lineages ([Sanderson, 1997](#); [Thorne *et al.*, 1998](#)). There is now a wide range of relaxed-clock models, implemented in various statistical frameworks ([Ho and Duchêne, 2014](#)). Among the most commonly used in phylogenetic inference are those based on the Bayesian approach ([dos Reis *et al.*, 2016](#)).

Approaches

Molecular-Clock Methods

The molecular clock is typically used as part of a phylogenetic analysis, in a procedure known as molecular dating. This involves inferring the evolutionary relationships among a sample of taxa, along with the evolutionary rates and time durations of the branches of the phylogenetic tree. Molecular dating comprises several key steps ([Fig. 2](#)): (i) selecting a data set, (ii) identifying

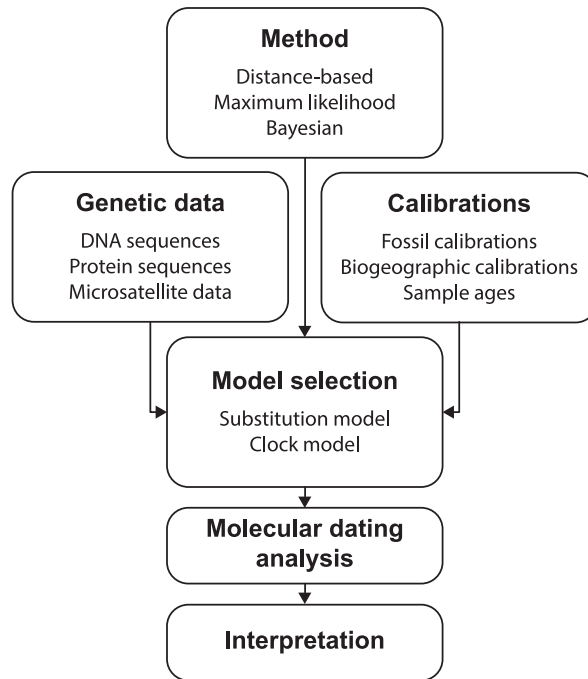


Fig. 2 Major components of a molecular dating analysis. The user begins by assembling a genetic data set, identifying suitable calibrations for the molecular clock, and selecting a method of analysis. Statistical model selection should then be performed to identify the best-fitting models of substitution and rate variation. Following the analysis of the data, the results then need to be interpreted and placed into the appropriate biological and evolutionary context.

calibrations for the molecular clock, (iii) selecting an estimation method and a model of rate variation, and (iv) running the analysis.

Choosing a data set for analysis depends on the questions that are being addressed. In most cases, nucleotide sequences are used for molecular-clock analyses because of the relative ease of obtaining these data and because of their information content. Sequences should represent the diversity of the organisms being analysed and should be sufficiently long to enable precise estimation of the amount of evolutionary change. However, genetic markers vary in their utility for molecular dating. Evolution tends to occur very slowly in functionally important genes, making them useful genetic markers for studying deep evolutionary timescales. In contrast, analyses of recent events are most effective when rapidly evolving genes are used. In animals, the mitochondrial genome typically evolves much more rapidly than the nuclear genome (Brown *et al.*, 1979). This pattern is reversed in plants: nucleotide substitutions tend to occur more rapidly in the nuclear genome than in the mitochondrial and chloroplast genomes. Mutation rates in viral genomes are higher than those in animals and plants by several orders of magnitude. In any case, analysing multiple genetic markers is now commonplace, and genome-scale data sets are seeing increasing use in molecular-clock studies (Ho, 2014).

The process of selecting calibrations, estimation methods, and models of rate variation can be informed by statistical analysis, but still involves a large degree of uncertainty. A wide range of methods are available, and these appeal to various philosophies and differ in their strengths and weaknesses (Ho and Duchêne, 2014; dos Reis *et al.*, 2016). There are also many practical factors that need to be considered, especially when analysing large data sets that comprise many sequences and/or many taxa.

Calibrating the Molecular Clock

The most critical component of molecular-clock analysis is the process of calibration. If the rate of evolution is unknown, the molecular clock must be calibrated in order for evolutionary timescales to be estimated. This involves constraining the age of at least one node in the phylogeny, allowing the average rate of its descendent branches to be inferred. The remaining nodes in the tree can then be estimated by assuming some model of rate variation across branches.

Calibrations can be chosen according to evidence from the fossil record, geological events, or past climatic events. In practice, fossil calibrations are probably the most common (Hipsley and Müller, 2014). Provided that a fossil taxon can be assigned to a lineage in the phylogeny, its age can be used to place a minimum constraint on the age of its ancestor. However, minimum age constraints need to be combined with at least one maximum age constraint in the tree. Choosing an appropriate maximum age constraint for the tree is a difficult exercise, because it relies on the assumption that a particular lineage or group did not exist before a certain time. Various criteria have been put forward for choosing appropriate fossil calibrations for the molecular clock (Parham *et al.*, 2012; Sauquet *et al.*, 2012).

In Bayesian methods, calibrations are implemented by specifying the prior distributions of the relevant node ages. For example, a node-age prior can be specified in the form of a lognormal or gamma distribution. These calibration priors can allow various sources of uncertainty to be taken into account, including errors in radiometric dating and the probability of fossil preservation (Ho and Phillips, 2009). Nevertheless, the uncertainty in the fossil evidence is rarely modelled explicitly, and the fossils themselves are only used indirectly to inform the calibration priors.

Recent developments in calibration methods have aimed to incorporate fossil evidence more comprehensively. Total-evidence dating uses combined data sets comprising genetic data from extant species and morphological data from both extant and extinct species. The morphological data are used to infer the placement of the extinct species in the tree, and the ages of these species calibrate the clock by constraining the ages of their ancestral nodes (Ronquist *et al.*, 2012). This method removes the need to choose age constraints for nodes based on fossil evidence, by instead allowing the fossils to be included directly in the analysis. Nevertheless, the method has a number of problems that still need to be addressed (O'Reilly *et al.*, 2015). In this regard, a key area of research is the development of more realistic models of morphological evolution.

Geological events can also be used for calibration, provided that they can be tied to evolutionary divergence events among the sampled taxa (Ho *et al.*, 2015). These divergences can be caused by disruptions in the range of an ancestral species (vicariance), the formation of connections between previously isolated habitats (geodispersal), or dispersal across barriers and establishment of new populations (biological dispersal). Examples of these include evolutionary diversification associated with continental fragmentation or the colonization of newly formed islands (e.g., Fleischer *et al.* (1998)). The timing of these events can be estimated using radiometric dating methods. However, biogeographic calibrations are typically associated with large amounts of uncertainty and are sometimes regarded as unreliable (Kodandaramaiah, 2011).

In some cases, the ages of the sequences can act as calibrations (Rambaut, 2000). These tip-calibrations can be used when there has been enough time separating the sequences to allow an appreciable amount of genetic change to occur (Drummond *et al.*, 2003). This is the case for some data sets comprising virus sequences sampled over time, or that include ancient sequences that have been radiocarbon-dated.

In the absence of primary calibrating information, age constraints can be applied to nodes in the tree based on previous molecular estimates. This practice is deprecated by some researchers, given the uncertainties in molecular dating and the potential sensitivity of date estimates to various confounding factors. However, such an approach might be the only feasible option in analyses of taxa with a poor fossil history. If secondary calibrations are employed, their uncertainty should be taken into account explicitly (Graur and Martin, 2004); this is readily done in Bayesian phylogenetic methods.

Models of Rate Variation

Evolutionary rates vary in a number of important ways, placing great importance on the act of selecting an appropriate clock model. Clock models describe how rates are expected to vary across branches in the tree and across genes in the data set. A simple way of accounting for different rates across genes is to assign a relative rate parameter to each gene. Accounting for rate variation across branches is more difficult, and various models have been developed for this purpose. One convenient way of classifying these models is to consider the number of distinct rates (k) that they allow in comparison with the number of branches (n) (Fig. 3; Ho and Duchêne, 2014).

The simplest model of rate variation is the strict clock, which assumes that all branches share the same rate ($k=1$). This model is often rejected in practice, with significant evidence of rate variation across branches being found for many data sets. However, the strict clock might be valid when analysing sequences from conspecific individuals or closely related species. It is also a useful null model when testing for evolutionary rate heterogeneity.

Local-clock models assume that there are several distinct evolutionary rates across the tree ($1 < k < n$). These models might be appropriate when there is reason to believe that some branches in the tree are likely to share an evolutionary rate that is different from those along other branches (Yoder and Yang, 2000). Some local-clock models require the tree topology to be fixed and for distinct rates to be assigned to pre-defined sets of branches, but the Bayesian random local clock allows all of these to be inferred jointly (Drummond and Suchard, 2010).

Variants of local-clock models allow the branches sharing the same rate to be distributed throughout the tree, rather than necessarily being closely related to each other. These are sometimes known as 'discrete clocks' (Fourment and Holmes, 2014). Discrete clocks have been implemented in a range of statistical frameworks, including likelihood and Bayesian methods (e.g., Heath *et al.* (2012)).

The assumptions of the molecular clock can be relaxed even further, to allow a distinct rate of evolution along each branch in the tree ($k=n$). In order for these rates to be identifiable, there must be some form of constraint on their variation. The earliest relaxed-clock models assumed that rates evolve gradually throughout the tree, such that rates are autocorrelated (Sanderson, 1997, 2002; Thorne *et al.*, 1998). In these models, evolutionary rates are assumed to be similar between adjacent branches in the tree. There are implementations of autocorrelated-rate models in likelihood methods, but most of the recent developments have occurred in a Bayesian framework.

In a second class of relaxed-clock models, known as uncorrelated relaxed clocks, the branch rates are drawn independently from an underlying probability distribution (Drummond *et al.*, 2006; Rannala and Yang, 2007). Thus, there is no prior assumption that the rates along adjacent branches are similar to each other. A mixed relaxed clock can incorporate both autocorrelated and uncorrelated models by partitioning the rate variation between them (Lartillot *et al.*, 2016).

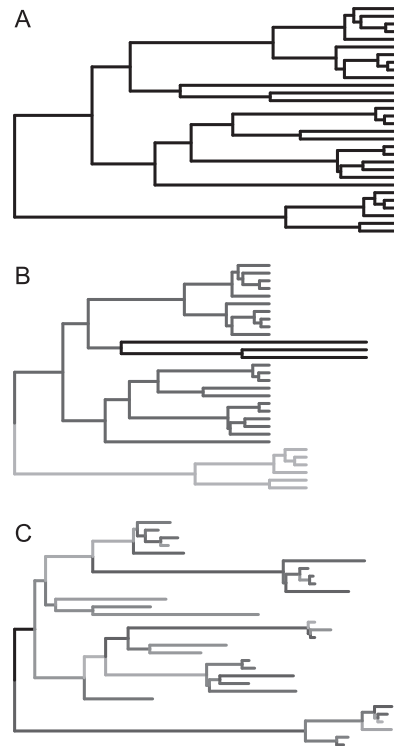


Fig. 3 The three major classes of clock model, based on the number of distinct rates that are allowed across branches in the phylogenetic tree. Branch shading indicates the relative evolutionary rate. (a) Strict clock, in which a single rate is shared by all branches. (b) Local clock, in which the number of rates is much smaller than the number of branches. Rates are typically shared between adjacent branches in the tree. In this example, there are three different rates across the tree. (c) Relaxed clock, which allows a distinct evolutionary rate for each branch in the tree.

Challenges

The molecular clock has faced a variety of challenges over the past few decades, and some of the most prominent estimates of evolutionary timescales have been controversial. The most important criticisms of molecular clocks have concerned the impacts of evolutionary rate variation across timescales, along with the use of calibrations.

The notion of constant evolutionary rates has been subject to numerous criticisms. It is now widely accepted that rates of evolution vary substantially across the Tree of Life. This rate variation is due to biological factors such as differences in population size, generation length, and natural selection, as well as abiotic factors such as exposure to ultraviolet radiation (Bromham, 2009).

Part of the controversy surrounding rate variation is the use of ‘universal’ molecular clocks, which assume a constant rate among species within a broad taxonomic group. For example, there is widespread use of universal clocks in studies of arthropods, birds, and mammals, whereby an evolutionary rate of 1% per million years is assumed for mitochondrial DNA (e.g., Weir and Schluter, 2008). However, the reliability of assuming a standard rate across an entire taxonomic group has been regularly challenged (e.g., Nabholz *et al.*, 2008; Nguyen and Ho, 2016). Given the abundant evidence that even closely related species can show substantial rate variation, the application of universal clocks can produce very misleading estimates of evolutionary timescales. Nevertheless, these clocks are still commonly used because there is often no other information about rates at hand.

The outcome of a molecular-clock analysis is substantially influenced by the choice of calibrations, and this has also been a major criticism of molecular dating. If calibrations are incorrectly modelled, the resulting date estimates can be highly inaccurate. When fossil calibrations are used, the selection of fossils can be subjective and their placement needs to be adequately justified. In Bayesian molecular dating, a further problem is that the calibration priors can interact with each other, producing joint priors on node times that no longer represent the fossil evidence used to specify the calibrations individually (Heled and Drummond, 2012; Warnock *et al.*, 2012).

The uncertainty in fossil calibrations can be modelled and quantified in various ways (e.g., Marshall (2008), Wilkinson *et al.* (2011)), and recently developed methods allow fossil data to be incorporated in a more satisfying manner. For example, total-evidence dating and the fossilized birth-death process allow fossil taxa to be treated as part of the evolutionary process (Heath *et al.*, 2014; Ronquist *et al.*, 2016). Although some aspects of calibration methodology remain debated, best practice involves using multiple calibrations for molecular dating.

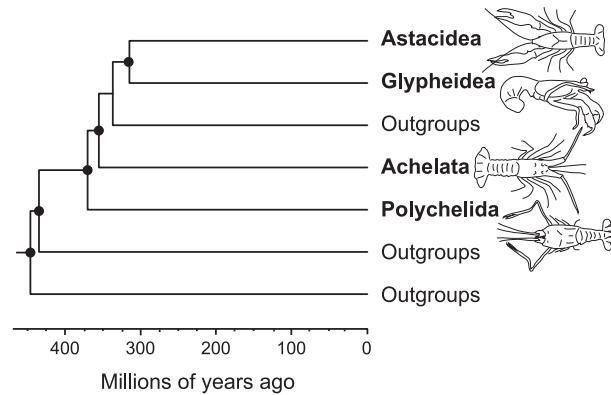


Fig. 4 Phylogeny and evolutionary timescale of lobster-like decapod crustaceans, inferred using a Bayesian phylogenetic analysis of molecular and morphological data by Bracken-Grissom *et al.* (2014). 'Outgroups' indicate groups of crustaceans that do not belong to the four infraorders of lobster-like decapods. Filled circles at internal nodes indicate the placement of fossil calibrations. Line drawings of animals show representatives of the four infraorders indicated in bold, and have been traced from images presented in the study by Bracken-Grissom *et al.*, 2014. The emergence of lobsters: phylogenetic relationships, morphological evolution and divergence time comparisons of an ancient group (Decapoda: Achelata, Astacidea, Glypheidea, Polychelida). *Systematic Biology* 63, 457–479.

Case Study: The Emergence of Lobsters

Lobsters are a morphologically diverse group of crustaceans belonging to the order Decapoda. They are economically and culturally important, representing a key resource for many fisheries throughout the world. Their exoskeletons increase their probability of preservation, meaning that they have a rich fossil record extending back approximately 350 million years. This record provides a unique opportunity to study the origins and diversification of major lineages within the group.

Bracken-Grissom *et al.* (2014) analysed data from 173 species to infer the phylogeny of lobster-like decapods. This sampling included 94% of extant genera and allowed the authors to resolve contentious relationships among and within groups (Fig. 4). Using a total-evidence approach, the authors analysed a combination of 190 morphological characters, three nuclear genes (18S, 28S, and H3), and three mitochondrial genes (16S, 12S, and CO1). The genetic markers included both fast- and slow-evolving genes, allowing various relationships to be resolved throughout the tree.

For each of the genes in the data set, Bracken-Grissom *et al.* (2014) identified the best-fitting model of nucleotide substitution. By inferring the tree from each gene individually, the authors checked for congruence in the phylogenetic signal across the six genes. Both maximum-likelihood and Bayesian phylogenetic methods were used to infer the evolutionary relationships among taxa. A relaxed molecular clock was used to account for rate variation across lineages. In one analysis, the clock was calibrated using the ages of 28 relevant fossils. These calibrations were implemented as exponential or uniform prior distributions for the ages of the corresponding nodes in the tree. In a second analysis, calibrations were based on a model of fossilization, speciation, and recovery (Wilkinson and Tavaré, 2009). The two approaches to calibration led to very similar sets of divergence-time estimates.

Bracken-Grissom *et al.* (2014) found that decapods arose in the Ordovician, with the most recent common ancestor of extant genera appearing in the Cretaceous. During the Jurassic, the infraorder Achelata rapidly diversified, coinciding with the early break-up of Pangaea. Southern and Northern Hemisphere crayfish were estimated to have diverged approximately 261 million years ago. In addition to these date estimates, Bracken-Grissom *et al.* (2014) provided a range of useful guidelines for incorporating fossil calibrations in molecular-clock studies.

Future Directions

Over the past decade, there have been substantial advances in DNA-sequencing technologies, leading to a dramatic increase in the data available for molecular dating. As a consequence, molecular-clock methods have had to evolve to deal with large data sets (Ho, 2014). Most of these data sets comprise either large numbers of taxa or large numbers of genes, but there is a growing trend towards genome-scale data sets that include many dozens of taxa (e.g., Jarvis *et al.* (2014), Misof *et al.* (2014)). These data sets can contain significant amounts of complex rate variation and can pose difficult computational challenges. The growing availability of genome-scale data sets has spurred the development of methods that can cope with large data sets, either by improved distribution of computational load or by relying on approximations of likelihood calculations (e.g., dos Reis and Yang (2011)). Nevertheless, current Bayesian phylogenetic methods are not capable of analysing data sets comprising large numbers of taxa and genes.

Molecular dating using genome-scale data sets must deal with the problem of incongruent gene trees. When the divergences between lineages have occurred in rapid succession, there is an increased probability of the phylogenetic signal being heterogeneous across the genome. In particular, the evolutionary histories of individual genes might not match the relationships among

the species, in a phenomenon known as incomplete lineage sorting. This problem has not been satisfactorily addressed in molecular dating, but it remains an active area of research.

The growing access to genome-scale data sets raises an important question: does the precision of molecular dating improve with the amount of data? Various investigations have revealed that even with infinite data, the error in Bayesian date estimates depends on the uncertainty in the calibrations (dos Reis and Yang (2013)). Further improvements in estimates of evolutionary timescales will rely on the discovery of informative fossils and the development of better methods for integrating fossil and molecular data.

Closing Remarks

The molecular clock has been used in thousands of scientific studies, with no signs of declining relevance in the genomic era. Throughout its long history, the molecular clock has undergone substantial evolution and has proven to be a valuable tool for estimating evolutionary timescales. Applications of the molecular clock have been widespread, ranging from macroevolutionary studies of speciation and extinction to monitoring the evolutionary dynamics of modern-day pathogens.

Molecular-clock approaches have evolved from the assumption of constant evolutionary rates to the current models of rate variation across lineages. There has been an increase in efforts to incorporate fossil data into molecular dating, through total-evidence dating and the fossilized birth-death process. By taking advantage of the large amounts of genomic data being generated, molecular-clock methods will continue to play an important role in understanding the timescale of the Tree of Life.

See also: Applications of the Coalescent for the Evolutionary Analysis of Genetic Data. Evolutionary Models. Gene Duplication and Speciation. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Phylogenetic Tree Rooting

References

- Arbogast, B.S., Edwards, S.V., Wakeley, J., Beerli, P., Slowinski, J.B., 2002. Estimating divergence times from molecular data on phylogenetic and population genetic timescales. *Annual Review of Ecology, Evolution, and Systematics* 33, 707–740.
- Bracken-Grisson, H.D., Ah Yong, S.T., Wilkinson, R.D., *et al.*, 2014. The emergence of lobsters: Phylogenetic relationships, morphological evolution and divergence time comparisons of an ancient group (Decapoda: Achelata, Astacidea, Glypheidea, Polychelida). *Systematic Biology* 63, 457–479.
- Bromham, L., 2009. Why do species vary in their rate of molecular evolution? *Biology Letters* 5, 401–404.
- Bromham, L., Rambaut, A., Fortey, R., Cooper, A., Penny, D., 1998. Testing the Cambrian explosion hypothesis by using a molecular dating technique. *Proceedings of the National Academy of Sciences of the USA* 95, 12386–12389.
- Brown, W.M., George, M., Wilson, A.C., 1979. Rapid evolution of animal mitochondrial DNA. *Proceedings of the National Academy of Sciences of the USA* 76, 1967–1971.
- dos Reis, M., Donoghue, P.C.J., Yang, Z., 2016. Bayesian molecular clock dating of species divergences in the genomics era. *Nature Reviews Genetics* 17, 71–80.
- dos Reis, M., Yang, Z., 2011. Approximate likelihood calculation on a phylogeny for Bayesian estimation of divergence times. *Molecular Biology and Evolution* 28, 2161–2172.
- dos Reis, M., Yang, Z., 2013. The unbearable uncertainty of Bayesian divergence time estimation. *Journal of Systematics and Evolution* 51, 30–43.
- Drummond, A.J., Ho, S.Y.W., Phillips, M.J., Rambaut, A., 2006. Relaxed phylogenetics and dating with confidence. *PLOS Biology* 4, e88.
- Drummond, A.J., Pybus, O.G., Rambaut, A., Forsberg, R., Rodrigo, A.G., 2003. Measurably evolving populations. *Trends in Ecology and Evolution* 18, 481–488.
- Drummond, A.J., Suchard, M.A., 2010. Bayesian random local clocks, or one rate to rule them all. *BMC Biology* 8, 114.
- Fleischer, R.C., McIntosh, C.E., Tarr, C.L., 1998. Evolution on a volcanic conveyor belt: Using phylogeographic reconstructions and K-Ar-based ages of the Hawaiian Islands to estimate molecular evolutionary rates. *Molecular Ecology* 7, 533–545.
- Foster, C.S.P., Sauquet, H., Van der Merwe, M., *et al.*, 2017. Evaluating the impact of genomic data and priors on Bayesian estimates of the angiosperm evolutionary timescale. *Systematic Biology* 66, 338–351.
- Fourment, M., Holmes, E.C., 2014. Novel non-parametric models to estimate evolutionary rates and divergence times from heterochronous sequence data. *BMC Evolutionary Biology* 14, 163.
- Graur, D., Martin, W., 2004. Reading the entrails of chickens: Molecular timescales of evolution and the illusion of precision. *Trends in Genetics* 20, 80–86.
- Heath, T.A., Holder, M.T., Huelsenbeck, J.P., 2012. A Dirichlet process prior for estimating lineage-specific substitution rates. *Molecular Biology and Evolution* 29, 939–955.
- Heath, T.A., Huelsenbeck, J.P., Stadler, T., 2014. The fossilized birth-death process for coherent calibration of divergence-time estimates. *Proceedings of the National Academy of Sciences of the USA* 111, 2957–2966.
- Heled, J., Drummond, A.J., 2012. Calibrated tree priors for relaxed phylogenetics and divergence time estimation. *Systematic Biology* 61, 138–149.
- Hipsley, C.A., Müller, J., 2014. Beyond fossil calibrations: Realities of molecular clock practices in evolutionary biology. *Frontiers in Genetics* 5, 138.
- Ho, S.Y.W., 2014. The changing face of the molecular evolutionary clock. *Trends in Ecology and Evolution* 29, 496–503.
- Ho, S.Y.W., Duchêne, S., 2014. Molecular-clock methods for estimating evolutionary rates and timescales. *Molecular Ecology* 23, 5947–5965.
- Ho, S.Y.W., Phillips, M.J., 2009. Accounting for calibration uncertainty in phylogenetic estimation of evolutionary divergence times. *Systematic Biology* 58, 367–380.
- Ho, S.Y.W., Tong, K.J., Foster, C.S.P., *et al.*, 2015. Biogeographic calibrations for the molecular clock. *Biology Letters* 11, 20150194.
- Jarvis, E.D., Mirarab, S., Aberer, A.J., *et al.*, 2014. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science* 346, 1320–1331.
- Kimura, M., 1968. Evolutionary rate at the molecular level. *Nature* 217, 624–626.
- Kodandaramaiah, U., 2011. Tectonic calibrations in molecular dating. *Current Zoology* 57, 116–124.
- Lartillot, N., Phillips, M.J., Ronquist, F., 2016. A mixed relaxed clock model. *Philosophical Transactions of the Royal Society B* 371, 20150132.
- Marshall, C.R., 2008. A simple method for bracketing absolute divergence times on molecular phylogenies using multiple fossil calibration points. *American Naturalist* 171, 726–742.
- Misof, B., Liu, S., Meusemann, K., *et al.*, 2014. Phylogenomics resolves the timing and pattern of insect evolution. *Science* 346, 763–767.

- Nabholz, B., Glémin, S., Galtier, N., 2008. Strong variation of mitochondrial mutation rate across mammals – the longevity hypothesis. *Molecular Biology and Evolution* 25, 120–130.
- Nguyen, J.M.T., Ho, S.Y.W., 2016. Mitochondrial rate variation among lineages of passerine birds. *Journal of Avian Biology* 47, 690–696.
- Ohta, T., 1972. Evolutionary rate of cistrons and DNA divergence. *Journal of Molecular Evolution* 1, 150–157.
- Ohta, T., 1973. Slightly deleterious mutant substitutions in evolution. *Nature* 246, 96–98.
- O'Reilly, J.E., dos Reis, M., Donoghue, P.C.J., 2015. Dating tips for divergence-time estimation. *Trends in Genetics* 31, 637–650.
- Parham, J.F., Donoghue, P.C.J., Bell, C.J., *et al.*, 2012. Best practices for justifying fossil calibrations. *Systematic Biology* 61, 346–359.
- Rambaut, A., 2000. Estimating the rate of molecular evolution: Incorporating non-contemporaneous sequences into maximum likelihood phylogenies. *Bioinformatics* 16, 395–399.
- Rannala, B., Yang, Z., 2007. Inferring speciation times under an episodic molecular clock. *Systematic Biology* 56, 453–466.
- Ronquist, F., Klopfstein, S., Vilhelmsen, L., *et al.*, 2012. A total-evidence approach to dating with fossils, applied to the early radiation of the Hymenoptera. *Systematic Biology* 61, 973–999.
- Ronquist, F., Lartillot, N., Phillips, M.J., 2016. Closing the gap between rocks and clocks using total-evidence dating. *Philosophical Transactions of the Royal Society B* 371, 20150136.
- Runnegar, B., 1982. A molecular-clock date for the origin of the animal phyla. *Lethaia* 15, 199–205.
- Sanderson, M.J., 1997. A nonparametric approach to estimating divergence times in the absence of rate constancy. *Molecular Biology and Evolution* 14, 1218–1231.
- Sanderson, M.J., 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. *Molecular Biology and Evolution* 19, 101–109.
- Sarich, V.M., Wilson, A.C., 1967. Immunological time scale for hominid evolution. *Science* 158, 1200–1203.
- Sauquet, H., Ho, S.Y.W., Gandolfo, M.A., *et al.*, 2012. Testing the impact of calibration on molecular divergence times using a fossil-rich group: The case of *Nothofagus* (Fagales). *Systematic Biology* 61, 289–313.
- Simpson, G.G., 1964. Organisms and molecules in evolution. *Science* 146, 1535–1538.
- Thorne, J.L., Kishino, H., Painter, I.S., 1998. Estimating the rate of evolution of the rate of molecular evolution. *Molecular Biology and Evolution* 15, 1647–1657.
- Tong, K.J., Lo, N., Ho, S.Y.W., 2016. Reconstructing evolutionary timescales using phylogenomics. *Zoological Systematics* 41, 343–351.
- Warnock, R.C.M., Yang, Z., Donoghue, P.C.J., 2012. Exploring uncertainty in the calibration of the molecular clock. *Biology Letters* 8, 156–159.
- Weir, J., Schluter, D., 2008. Calibrating the avian molecular clock. *Molecular Ecology* 17, 2321–2328.
- Wilkinson, R.D., Steiper, M.E., Soligo, C., *et al.*, 2011. Dating primate divergences through an integrated analysis of palaeontological and molecular data. *Systematic Biology* 60, 16–31.
- Wilkinson, R.D., Tavaré, S., 2009. Estimating primate divergence times by using conditioned birth-and-death processes. *Theoretical Population Biology* 75, 278–285.
- Yoder, A.D., Yang, Z., 2000. Estimation of primate speciation dates using local molecular clocks. *Molecular Biology and Evolution* 17, 1081–1090.
- Zuckerkandl, E., Pauling, L., 1962. Molecular disease, evolution and genic heterogeneity. In: Kasha, M., Pullman, B. (Eds.), *Horizons in Biochemistry*. New York: Academic Press, pp. 189–225.
- Zuckerkandl, E., Pauling, L., 1965. Evolutionary divergence and convergence in proteins. In: Bryson, V., Vogel, H.J. (Eds.), *Evolving Genes and Proteins*. New York: Academic Press, pp. 97–166.

Further Reading

- Bromham, L., 2011. The genome as a life-history character: Why rate of molecular evolution varies between mammal species. *Philosophical Transactions of the Royal Society B* 366, 2503–2513.
- Donoghue, P.C.J., Yang, Z., 2016. The evolution of methods for establishing evolutionary timescales. *Philosophical Transactions of the Royal Society B* 371, 20160020.
- dos Reis, M., Donoghue, P.C.J., Yang, Z., 2016. Bayesian molecular clock dating of species divergences in the genomics era. *Nature Reviews Genetics* 17, 71–80.
- Heath, T.A., Huelsenbeck, J.P., Stadler, T., 2014. The fossilized birth-death process for coherent calibration of divergence-time estimates. *Proceedings of the National Academy of Sciences of the USA* 111, 2957–2966.
- Heath, T.A., Moore, B.R., 2014. Bayesian Inference of Species Divergence Times. In: Chen, M.-H., Kuo, L., Lewis, P.O. (Eds.), *Bayesian Phylogenetics: Methods, Algorithms, and Applications*. Boca Raton: CRC Press, pp. 277–318.
- Hipsley, C.A., Müller, J., 2014. Beyond fossil calibrations: Realities of molecular clock practices in evolutionary biology. *Frontiers in Genetics* 5, 138.
- Ho, S.Y.W., 2014. The changing face of the molecular evolutionary clock. *Trends in Ecology and Evolution* 29, 496–503.
- Ho, S.Y.W., Duchêne, S., 2014. Molecular-clock methods for estimating evolutionary rates and timescales. *Molecular Ecology* 23, 5947–5965.
- Kumar, S., Hedges, S.B., 2016. Advances in time estimation methods for molecular data. *Molecular Biology and Evolution* 33, 863–869.
- O'Reilly, J.E., dos Reis, M., Donoghue, P.C.J., 2015. Dating tips for divergence-time estimation. *Trends in Genetics* 31, 637–650.
- Sauquet, H., 2013. A practical guide to molecular dating. *Comptes Rendus Palevol* 12, 355–367.
- Tong, K.J., Lo, N., Ho, S.Y.W., 2016. Reconstructing evolutionary timescales using phylogenomics. *Zoological Systematics* 41, 343–351.

Biographical Sketch

Cara Van Der Wal is a marine evolutionary biologist interested in molecular clocks, phylogenetics, computational biology, systematics, and population genetics. She is a PhD candidate at the University of Sydney and the Australian Museum. Prior to commencing her PhD, Cara studied marine biology at the University of Technology Sydney and did research on stomatopod crustaceans at the University of Sydney.

Simon Ho is a Professor of Molecular Evolution at the University of Sydney, where he leads the Molecular Ecology, Evolution, and Phylogenetics research group. He is a computational evolutionary biologist interested in molecular clocks, evolutionary rates, phylogenetic methods, genomic evolution, and ancient DNA. He previously worked at the Australian National University and the University of Oxford, after being awarded his DPhil from the University of Oxford in 2006.

Phylogenetic Tree Rooting

Richard J Edwards, University of New South Wales, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Glossary

Additive tree A tree in which the branch lengths are scaled according to time or the amount of evolutionary change.

Clade A group of terminal nodes (OTUs) that share a common ancestral node (HTU).

HTU Hypothetical Taxonomic Unit. An internal node of a phylogenetic tree.

Molecular Clock The prediction of The Neutral Theory that, if most fixed changes are the result of neutral mutations, molecular evolution will occur at a reasonably regular “clock-like” rate, determined primarily by the neutral mutation rate.

Monophyletic A trait (physical or genetic) that occurs within a single clade on a phylogenetic tree and thus can be explained by a single evolutionary event.

OTU Operational Taxonomic Unit. A sequence or organism used as the terminal nodes of a phylogenetic tree.

Parsimony The simplest explanation for an observation. The smallest number of changes needed to explain the data.

Topology The branching order of a phylogenetic tree.

Ultrametric Additive tree where all root-to-tip paths have an equal summed branch length.

Introduction: Phylogenetic Trees and the Role of the Root

A **phylogenetic tree** is a graphical representation of the evolutionary relationships between biological entities (Box 1). Phylogenetic trees can be used to model the relationships of organisms (“species trees”) or, in the case of molecular phylogenetics, biological sequences (“gene trees” or “protein trees”). The methods differ in their details but the essence is the same, in which attempts are made to generate a branching structure that recapitulates the true biological relationships of the organisms or sequences in question. Because molecular phylogenetics is (a) more common (thanks to the ease of generating genetic data), and (b) inherently quantitative (due to the digital nature of the data itself and the quantitative nature of substitutional models), the rest of this chapter will be targeted at molecular phylogenetics. Unless otherwise stated, what is true for molecular phylogenies will generally also apply to species trees inferred from other types of data (e.g. morphology).

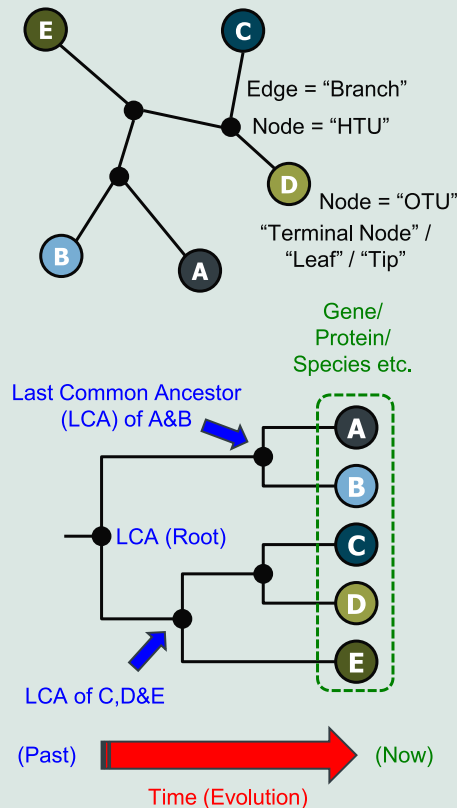
Phylogenetic trees are a special kind of dendrogram (or hierarchical clustering) in which the internal nodes are inferred to represent actual biological entities, namely common evolutionary ancestors of two or more leaves. The **role of the root** is to add direction to these relationships and clearly distinguish which internal nodes are ancestral to which tips (Fig. 1(a)). Relationships between sequences are captured by the **topology**, which is the branching order (Fig. 1(b)) and, where appropriate, **branch lengths** that indicate the amount of evolutionary change (or time) between nodes (Fig. 2). Note that **topology only** trees (such as Figs. 1(a) and 2(a)) do not have meaningful branch lengths, which limits the rooting options (discussed below).

Phylogenetic methods can broadly be divided into three classes:

1. **Parsimony methods** seeks to minimize the number of evolutionary events required to explain the data. Parsimony methods are often biologically meaningful but only use a subset of “informative” data that can be used to split sequences into two or more groups of two or more sequences.
2. **Distance methods** convert pairwise comparisons of sequences into a distance matrix and then cluster sequences based on proximity or use an algorithm to try to minimize overall branch lengths. Distance methods use all the data but add a layer of abstraction between the actual biology and the inferred phylogeny. They tend to be very computationally efficient and thus are popular choices for non-experts.
3. **Model-based methods** (generally Maximum Likelihood or Bayesian methods) use an explicit evolutionary model to try to infer the phylogenetic tree that best fits the data. In principle, these methods are best as they are both biologically meaningful *and* use all the data. The downside is that they are dependent on the selected evolutionary model and can be very computationally intensive.

Rooting is largely independent of the method used to infer the phylogeny. Here, the primary consideration is whether branches have lengths, or only the topology (branching order) is established (Fig. 2).

In **additive trees**, branch lengths represent evolutionary change. For species trees, this is often an estimation of literal times of divergence. For sequence trees, branch lengths generally represent evolutionary change more directly, in terms of substitutions per site. Converting non-sequence data into a scaling numerical value with a reasonably constant rate through time is very challenging. For this reason, trees based on morphology tend to use parsimony approaches and often do not have branch lengths. Where branch lengths (and roots) are set, it is often using independent data, such as fossils. Where fossil evidence is incorporated, it is possible to have actual historical samples in the tree. This is quite different from molecular phylogenetics where, except in very rare cases (e.g. frozen viral samples), the tree is wholly inferred from contemporary data.

Box 1 Trees as graphs

Phylogenetic trees are a special case of what is known as a graph, in which a series of **nodes** are joined by **edges**, known as **branches**. Nodes are split into terminal nodes (or “leaves” to keep with the tree analogy), also known as **Operational Taxonomic Units (OTUs)** and internal nodes, also known as **Hypothetical Taxonomic Units (HTUs)**; OTUs are generally the data on which the phylogenetic method operates, whereas HTUs are the hypothetical ancestors inferred by the method.

Unrooted trees (top figure) are “undirected graphs” in which the relationship between two connected nodes is symmetrical. Adding a **root** (bottom figure) adds directionality to the graph, such that each branch has an **ancestral node** closer to the root and a **descendant node** closer to the tips. Internal nodes now represent the **Last Common Ancestor (LCA)** of a **clade** of terminal nodes.

This chapter deals with strictly **bifurcating** (or **binary**) trees in which each HTU is connected to three branches (see Section “Midpoint Rooting”). In the case of rooted trees, this corresponds to one ancestral and two descendant branches.

Unless a **molecular clock** is assumed, substitution rates can differ throughout the tree. Many phylogenetic methods are vulnerable to large rate variations and, whilst they can compensate for an extent of variation, confidence in the phylogeny produced generally decreases as rate variation increases (see Section “Resolving Monophyly and Homoplasy”).

Rooting and Evolutionary Change

Rooting a tree adds an additional internal “node” (or HTU) to a tree, which indicates the **last common ancestor** of everything in the tree (**Box 1**). In doing so, the tree is given *direction*. In addition to inferring the relative relationships of our species/sequences based on the topology, we can now look at directions of change. One key thing to remember is that placing a root does not actually change the amount of evolutionary change in the tree. Rooting a tree on a branch simply partitions up the evolutionary changes attributed to that branch into the two post-root branches (**Fig. 3**). The critical difference is that evolutionary change now has an explicit **ancestral state** and **derived state** (**Fig. 1**) (see Section “Ancestral Sequence Prediction”).

Multifurcating Trees

For simplicity, this chapter is dealing with strictly **bifurcating** or **binary trees**, in which each internal node has one ancestral branch and two descendant branches, for a total of three branches (**Box 1**). Trees do not need to be strictly bifurcating and a node can have

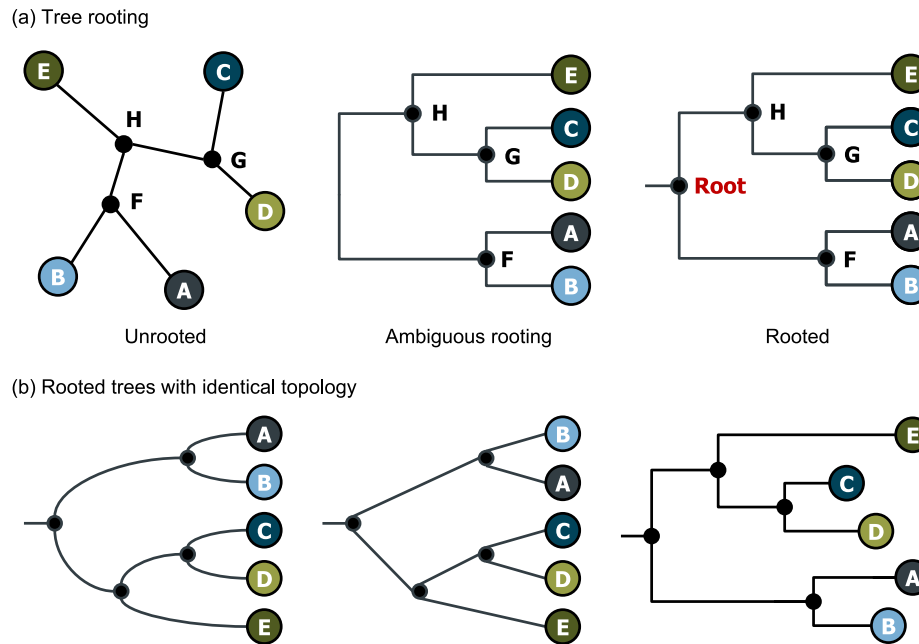


Fig. 1 Tree rooting and topology. The same phylogenetic tree of five sequences A-E with different representations. (a) Hypothetical nodes F-H represent inferred ancestors but in an unrooted tree (left) the relationship between those ancestors and other nodes is unknown. Rooting the tree (right) does not change the topology (branching order) but adds an additional ancestral node that gives directionality. In the unrooted tree, node H could be the common ancestor of (AB)E, (CD)E or (AB)(CD). In the rooted tree, node H is unambiguously the common ancestor of (CD)E. Sometimes a tree appears to be rooted but has no explicit root node marked (center). Unless otherwise stated, it is safest to assume that such a tree is unrooted. (b) Evolutionary relationships are represented by the tree topology, i.e. the branching order. This is independent of the way that branches are represented, or the vertical arrangement of terminal nodes. Despite different vertical arrangements and styles, all three trees have the same topology.

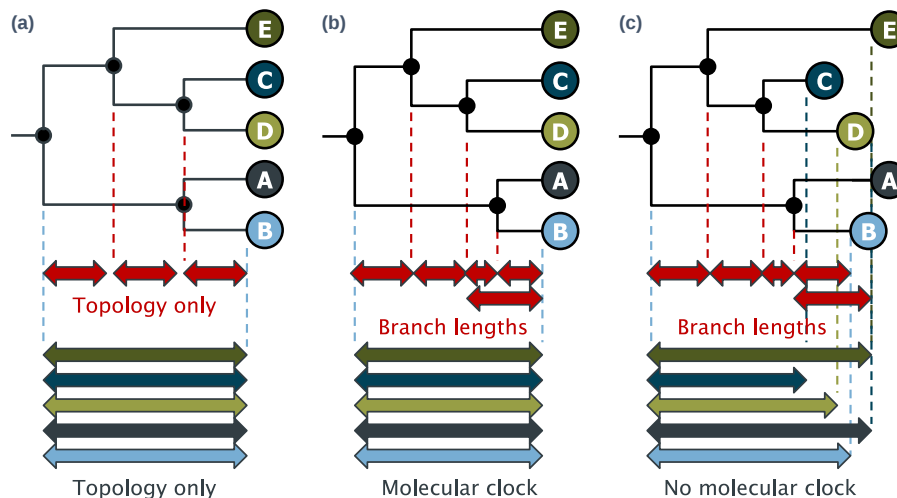


Fig. 2 Branch lengths and molecular clock. When looking at a rooted phylogenetic tree, the spacing between nodes and the root can be used to identify whether branch lengths are being shown and, if so, whether a molecular clock is in operation. (a) A topology only tree is ultrametric (all terminal nodes are the same distance from the root) and its internal nodes are typically evenly spaced at regular intervals as branches join, starting closest to the tip. (Note: vertical spacing in this representation has no meaning.) (b) If the tree is ultrametric but internal nodes are not regularly spaced then branch lengths are being used in the context of a molecular clock (a constant rate of evolution). This is usually means it is an assumption of the phylogenetic inference method but it *could* also reflect empirically clock-like evolution. Such a tree is inherently midpoint-rooted. (c) If neither terminal nor internal nodes show regular spacing, branch lengths are being used and no (universal) molecular clock is assumed. In this example, midpoint rooting is used.

more than two descendant branches. This is known as **multifurcation**. This could be a genuine biological phenomenon – a single ancestral population or sequence may radiate into more than two distinct lineages in parallel. Alternatively, it could simply represent a lack of confidence in tree topology; low confidence branches may be collapsed to zero length to explicitly acknowledge

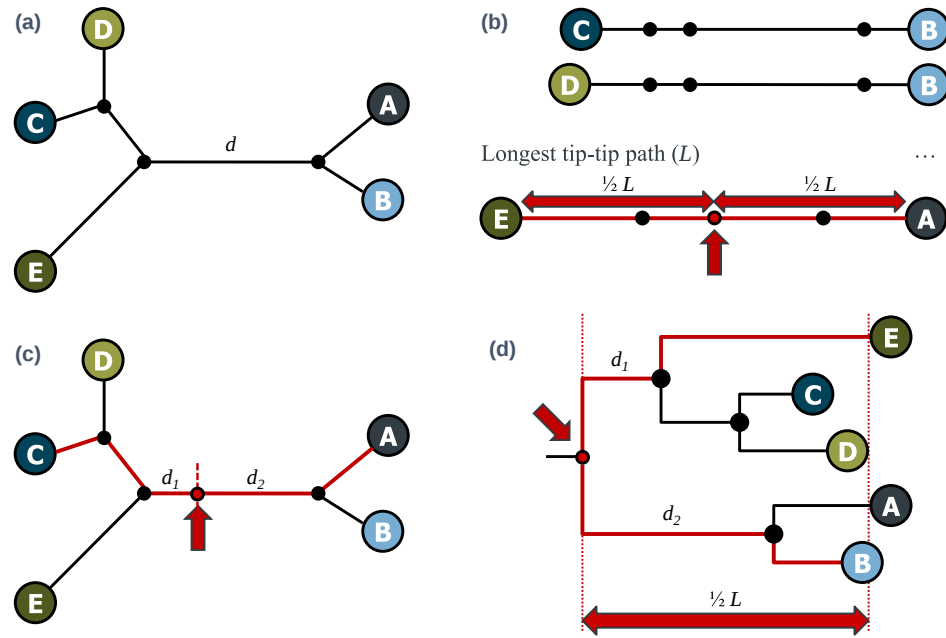


Fig. 3 Midpoint rooting. (a) Midpoint rooting starts with an unrooted tree with branch lengths. (b) Each pairwise tip-to-tip distance is calculated (only three shown) and the longest distance, L , selected to determine the root placement. The root is placed on this longest tip-to-tip path, equidistant from the two tips. (c) The evolutionary change along the rooted branches, d , is partitioned between the two new post-root branches such that $d_1 + d_2 = d$. (d) In the final tree, the two tips from the longest path, A and E, are furthest and equidistant from the root.

the lack of certainty about the branching order in that part of the tree. Unless otherwise stated, methods and considerations for rooting binary trees will also apply to multifurcating trees. (Multifurcation can be represented as a series of bifurcating branches of zero length).

Phylogenetic networks

Evolution is not always a neatly bifurcating process. Sexual reproduction, recombination and horizontal gene transfer can generate scenarios in which a given OTU has multiple ancestral HTUs. Traditional phylogenetic methods adopt a “winner takes all” approach to ancestry and it is up to the user to pre-partition the data where mixed ancestry may be an issue. **Phylogenetic network** methods will explicitly model mixed ancestry in addition to non-binary divergence. As with simpler multifurcation of descendants, rooting methods and considerations are broadly the same for a phylogenetic network and these will not be considered further in this chapter.

Rooting Methods

To correctly interpret a rooted tree, it is important to understand the assumptions and limitations of the rooting method used (see also Section “Why root a phylogenetic tree?”). There are a few different rooting methods, depending on the available data and reason for rooting (see Section “Inferring Rooting Methods”). These will be briefly summarized in this section.

Random Rooting

The simplest method for rooting a tree is to place the root on a random branch. This is clearly not a biologically meaningful rooting but repeating the process multiple times can be useful for establishing how robust the ultimate conclusions of a study are to incorrect rooting. In its simplest form, topology alone is considered. Alternatively, branches could be weighted by length, so that the root was more likely to fall in a long branch.

Midpoint Rooting

The easiest biologically meaningful rooting method is **midpoint rooting** (Fig. 3). Midpoint rooting needs an additive tree (i.e. scaled branch lengths, Fig. 2) and makes the explicit assumption that there are no major deviations in evolutionary rates (see Section “Resolving Monophyly and Homoplasy”) and that sequences are evolving in a reasonably “clock-like” fashion. Under this model, it follows that the root will be equidistant from the most divergent pair of sequences in the tree: if rates are reasonably

constant, the reason they are the most diverged is because they have had the longest period of independent evolution since splitting.

Midpoint rooting works by first establishing all the tip-to-tip distances by summing up the branch lengths of the shortest distance between two tips. The root is then placed in the midpoint of the longest tip-to-tip path (Fig. 3). If there are multiple equally-long paths, one is chosen at random, as they will all share an identical midpoint. **Ultrametric trees** (Fig. 2(b)) represent a special case of midpoint rooting where the evolutionary rate is constant throughout the tree and thus all tips end up equidistant from the root.

Midpoint rooting is attractive because it does not rely on any prior knowledge and can be calculated simply from the tree itself. However, the explicit assumption of constant evolutionary rates makes it quite unreliable in many real scenarios. It is advisable to look at how “clock-like” a tree appears, i.e. how much spread there is in root-to-tip distances following midpoint rooting (Fig. 2(c)), before deciding whether to trust midpoint rooting.

Rate-adjusted midpoint rooting

There are two main considerations for evolutionary rates:

1. The selective constraint of the sequence in that lineage.
2. The mutation rate of the sequence in that lineage, which is affected by generation time and the number of germline cell divisions per generation.

In principle, rate variations between branches can be modelled (as part of a model-based phylogenetic method) and the branch lengths normalized for different rates to convert them into evolutionary time. The tree could then be midpoint rooted based on rate-corrected branch lengths. However, this is often done by imposing a known species tree onto the gene tree to establish the branch-specific rates, which is essentially a form of outgroup rooting (see Section “Long Branch Artifacts”).

Outgroup Rooting

Outgroup rooting is the generally considered to be the Gold Standard of tree rooting. Here, prior knowledge is used to manually identify the ancestral branch in the tree. This is done by identifying a **clade** that is known to have branched off first from the rest of the tree, and then placing the root on the branch leading to that clade (Fig. 4). Often, the outgroup clade is a single OTU.

Selecting an outgroup

Selecting outgroup sequences is not always easy. As with all knowledge-based methods, outgroup rooting is only as good as the knowledge used. It will also make some explicit assumptions of the data, e.g. that all the sequences in the tree are orthologues (related by speciation). If a tree accidentally includes some paralogues (homologous sequences derived from duplication events),

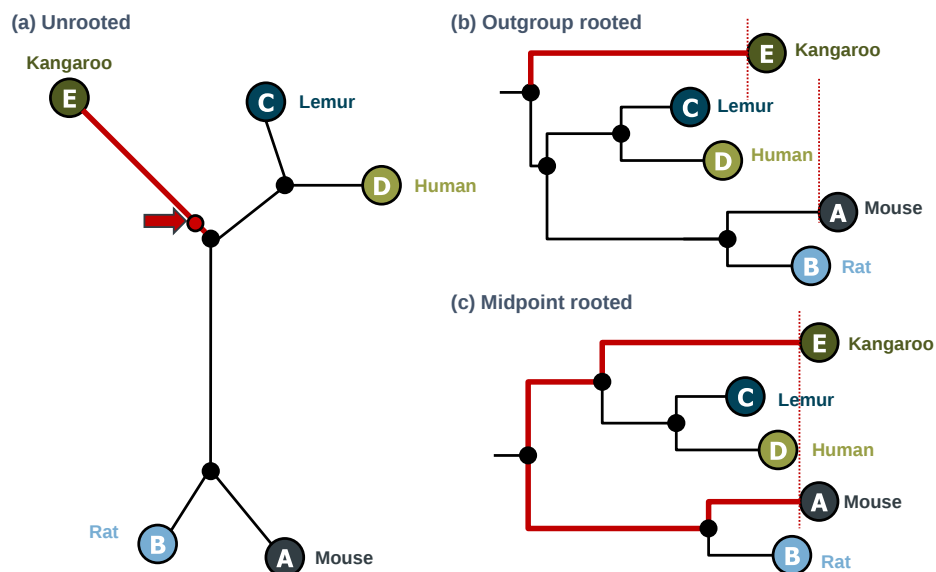


Fig. 4 Outgroup rooting. (a) Outgroup rooting identifies the OTU or clade known to have diverged first from the rest of the tree. In this case, the marsupial kangaroo diverged from the four placental mammals. (b) The root is placed at an arbitrary point on the branch leading to the outgroup. Typically, this is done to minimize the difference in the root-to-tip distances. Unlike midpoint rooting, the two most divergent tips will not necessarily be equidistant from the root. (c) In this example, the rodent lineage is evolving faster than the rest of the tree, which would result in an incorrect midpoint rooting in which primates and marsupials share a common ancestor since their divergence from rodents.

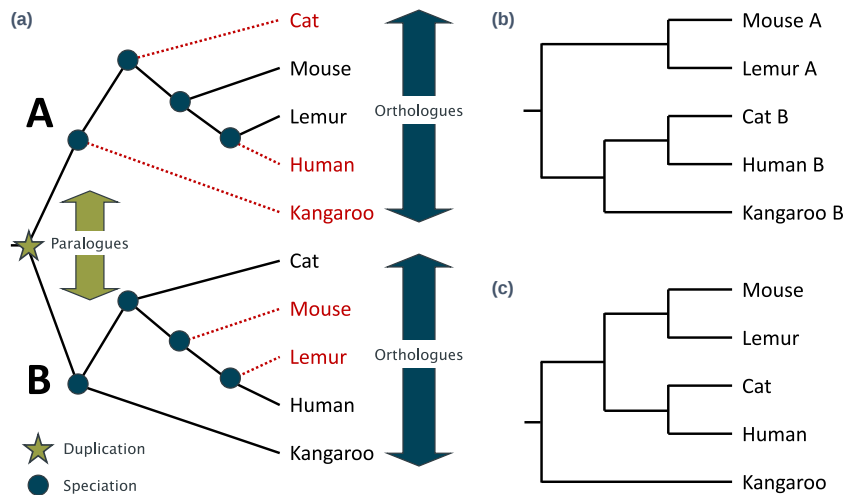


Fig. 5 Parologue rooting artefacts. (a) True topology-only tree of a gene family with two genes A and B in five mammalian species. The root of the tree is a gene duplication event (star). Other nodes (filled circles) are speciation events. Genes related by duplication are paralogues. Genes related by speciation are orthologues. Missing data for subsequent trees is marked in red. (b) The true topology-only tree without the missing data from the original tree. Orthologues still form monophyletic clades and the root is a duplication event. (c) Using kangaroo as an outgroup but ignoring the paralogous nature of some of the sequences incorrectly roots the reduced tree and produces a confusing topology in which primates are non-monophyletic.

this assumption would be violated and the outgroup rooting could possibly go wrong (Fig. 5). Single, distant OTUs are more likely to be incorrectly placed in a tree, so it is usually better practice to root on an outgroup clade rather than a single sequence.

Using gene duplications as an outgroup

Although incorrect classification of proteins as orthologues or paralogues can lead to rooting issues (Fig. 5), known relationships between different members of a gene family can be used for outgroup rooting in the absence of known taxonomic relationships. In Fig. 5(a), for example, the duplication node can be used to root the tree. In effect, gene A is acting as an outgroup for gene B and vice versa. This can be useful even when one subfamily is poorly sampled, e.g. Human B could be used to outgroup root the entire Gene A clade.

Adding outgroups

Often, there is no obvious outgroup in the set of sequences or species from which you have made your phylogeny. (Indeed, establishing the nature of relationships is often the reason for making a phylogeny from these data.) In these cases, it is useful to add one or more additional sequences from known outgroups and then use these to root the tree. For example, when generating a phylogeny of placental mammalian species or sequences, one might add one or more marsupials to root the tree (e.g. Fig. 4). Alternatively, one can add paralogous sequences as described above (see Section “Using gene duplications as an outgroup”).

Unrooting and Re-Rooting Trees

Unrooting and re-rooting a tree is very simple. The two post-root branches are simply stuck together to (re)generate an unrooted tree (e.g. converting Fig. 3(d) to (c)). The new root can then be placed using any of the standard methods.

Recognizing Rooted Trees (and other Tree Attributes)

Correct interpretation of a phylogenetic tree relies on understanding the method used for rooting. Unfortunately, this information is not always provided. Fortunately, different tree attributes and rooting methods leave typical signs that can be used to identify the rooting method with reasonable confidence. If not possible, the safest course of action is probably to re-root the tree yourself (see Section “Unrooting and Re-Rooting Trees”). (Of course, once identified, you might not agree with the method used and want to re-root anyway).

Recognizing an Unrooted Tree

The easiest rooting method to recognize is where no rooting has been applied and the tree is in a radiation representation (Fig. 1(a), left). Sometimes unrooted trees are still drawn in a traditional “square branch” configuration (Fig. 1(a), center) even though

there is no actual root. This is most common – and hardest to spot – when only the tree topology is shown (Fig. 2(a)) but is sometimes the default output for phylogenetics programs. Ideally, a rooted tree will have an actual root node and/or branch shown (Fig. 1(a), right) but this is not always the case and so trees that look rooted but lack a root are inherently ambiguous. Displaying branch lengths will often make it clear whether such a tree has been rooted; any major disparity between the root-tip distances each side of the root should be a cause for concern. If in doubt, explicitly unroot or re-root the tree (see Section “Unrooting and Re-Rooting Trees”), depending on your needs.

Inferring Rooting Methods

When interpreting a rooted phylogenetic tree, there are five basic attributes that one should look for:

1. **Topology.** This is the basic shape of the tree, e.g. its branching pattern (Fig. 1). All phylogenetic trees have a topology and it is the only attribute that one can guarantee. Topology-only trees are recognized by the uniformity of the branching pattern (Fig. 2(a)).
2. **Branch lengths.** “Additive” trees have evolutionary change information encoded in their branches, giving different branches different lengths. Additive trees can be recognized by the fact that (except for rare situations) the internal nodes are no longer spaced evenly and “line up” (Fig. 2). This is a pre-requisite for midpoint rooting (see Section “Inferring Rooting Methods”).
3. **Molecular clock.** Most phylogenetics methods perform best when evolutionary rates are constant. Some methods go as far as to enforce this, producing what is known as an ultrametric tree. In this case, all leaves are equidistant from the root even though the internal nodes are not evenly spaced (Fig. 2(b)). If no molecular clock is imposed, different leaves will be different distances from the root (Fig. 2(c)).
4. **Midpoint rooting.** Even in the absence of other knowledge, it is possible to assess whether an additive tree is consistent with midpoint rooting. If it is then the longest root-to-leaf distance will be the same for each “side” of the tree (i.e. for each of the two descendant clades that split at the root itself) (Figs. 3(d) and 4(c)). On the other hand, if one side of the tree sticks out further than the other, midpoint rooting has not been used (Fig. 4(b)). In this case, the assumption is usually that outgroup rooting has been used but this can only be confirmed using independent knowledge. Note that for reliable trees, midpoint rooting and outgroup rooting will give the same root; consistency with midpoint rooting does not therefore rule out the use of outgroup rooting.
5. **Branch confidence.** Rooting on a particular branch can only ever be as confident as the branch itself. Methods of assessing branch confidence (typically bootstrapping or probability estimates) are beyond the scope of this chapter but will be recorded on a tree in the form of numbers associated with each branch. If the rooted branch has low support, the root itself should be considered uncertain. (Note that in most cases, the confidence estimates are made on the unrooted tree and so both descendant branches from the root will have the same confidence score.) Where confidence values are absent, branch lengths can be used as a proxy. (Note: if some scores are given and not others, missing values normally indicate low confidence.) Usually, longer branches will be more robustly supported by the data; the reason the branch is long is that a lot of data supports that division of the tree. This makes rooting particularly uncertain in scenarios of rapid early radiation, where deep branches are short. Neither branch confidence nor branch length is outright confirmation that the root (or even the topology) is correct; a phylogenetic method or alignment error might be consistently biased into producing the wrong tree. However, a low confidence and/or very short rooting branch should always make one wary of root placement. In principle, one can apply the same confidence methods to the root itself, e.g. midpoint rooting every bootstrap tree to establish how many times the root ends up in the same place. In practice, this is rarely done.

Long Branch Artifacts

Phylogenetics methods work best when:

1. Evolutionary rates are reasonably constant across the whole tree.
2. Sampling of OTUs is reasonably uniform so that there is enough data to determine relationships but not so many differences that they begin to get masked by “multiple hits” (where the same nucleotide or amino acid undergoes several substitutions *on the same branch*, masking the earlier changes).

Where sequences violate these conditions, and have a lot of divergence relative to all the other OTUs, methods can get misled and interpret an increased evolutionary rate as an increased time of divergence. Distance methods are particularly prone to this issue but Maximum Parsimony (and even likelihood methods) can also struggle when substitutions are getting saturated in rapidly evolving lineages.

For distance methods, very long branches tend to migrate towards the midpoint root of the tree because the other OTUs cluster first. This can result in both an incorrect (but sometimes highly supported) basal placement of those branches but also an incorrect rooting if midpoint rooting is used.

For Maximum Parsimony and model-based methods, the problem is subtler. Large numbers of substitutions increase the chance of randomly shared substitutions. If these begin to exceed the genuine shared substitutions of these sequences with their more slowly evolving neighbors, the rapidly evolving lineages can begin to cluster together in a phenomenon known as “long branch attraction”. Again, midpoint rooting will inherently tend to put the attracted clade near the root as the long branch lengths will also make them more likely to be part of the longest tip-to-tip path (Fig. 3).

Unfortunately, outgroup rooting is often performed by selecting a single sequence known to be more distantly related to all other OTUs. Assuming reasonably constant evolutionary rates, rooting on this outgroup will match midpoint rooting. This is a good thing but can be visually indistinguishable from a scenario where a single rapidly-evolving lineage has migrated to the root. This emphasizes the need to document the rooting method used. The outgroup might also “attract” other rapidly evolving sequences, which share sequence variants by chance. Where possible, outgroup rooting should be done using multiple sequences to produce a more “balanced” tree and avoid this issue.

Why Root a Phylogenetic Tree?

Most phylogenetic methods are inherently undirected. This enables the user to understand the hierarchical relationships between sequences or species but does not reveal (or predict) the ancestral state. The first question when considering the rooting of a phylogenetic tree is whether one needs to root the tree at all. Often this is not required for an analysis. This depends on how important it is to infer the direction of change (see Section “Random Rooting”, Fig. 1).

Phylogenetic trees are often generated to provide context for certain species or sequences. In this case, we are primarily interested in which clade contains our query sequence. This is often obvious from an unrooted tree, in which case it might be considered safer not to try to establish a root. For example, in Fig. 4, mouse and rat are clearly most closely related, wherever the root is placed. Likewise, when trying to identify an unknown gene, its phylogenetic position within a clade might be enough to predict its identity, independent of strict ancestry: a complete unrooted version of Fig. 5(a) would suffice to clearly identify which mouse sequence was Gene A and which was Gene B. (Things might be less clear for the kangaroo sequences).

Ancestry becomes important when we are interested in *how* a trait evolved and want to know when it appeared. Alternatively, perhaps we know a trait appeared at a certain time and want to see what sequence changes correlate with its appearance. As with the example above, sometimes it is not actually necessary to *accurately* root the tree to get the ancestry information that one needs.

Ultrametric Phylogenetic Methods

One trivial reason to root a tree is that the chosen phylogenetic method inherently produces a rooted tree. As previously discussed, “Ultrametric” phylogenetic methods (e.g. Unweighted Pair Group Method with Arithmetic mean (UPGMA)) explicitly assume a constant rate of evolution across all branches. As such, the tree root is set as part of the phylogenetic reconstruction itself at the point where the last pair of groups are joined. In Ultrametric trees, all terminal nodes (OTUs) are equidistant from the root (Fig. 2(a) and (b)). Ultrametric tree rooting is therefore a special case of midpoint rooting where there are multiple longest tip-to-tip paths.

Ancestral Sequence Prediction

One of the main reasons for wanting to know the direction of branching is to establish the direction and timing of evolutionary change (Fig. 1). Note that it can be very challenging to correctly allocate the ancestral state of the root itself when there is disagreement between the two post-root nodes (Fig. 6). In general, the ancestral node will tend to converge on the node with the shortest branch length because each substitution is more likely to occur on the longer branch. This can lead to the slightly paradoxical situation where a non-zero-length branch has zero substitutions assigned to it. This, in turn, highlights some of the difficulties in interpreting distance-based trees, which tend to spread out discrete evolutionary events across multiple branches.

Resolving Monophyly and Homoplasy

Traits can either be **monophyletic** or **non-monophyletic**. Monophyletic traits are shared by all members of a clade, such as leucine (or valine) in Fig. 6(c). Non-mono-phyletic traits exclude one or more members of the clade, such as leucine (but not valine) in Fig. 6(b). Correct tree rooting can distinguish these scenarios (Fig. 6).

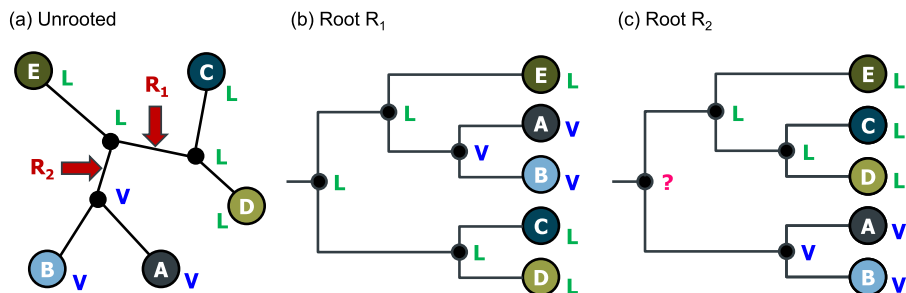


Fig. 6 Ancestral sequence assignment and rooting. (a) Five sequences have either a leucine (L) or valine (V) at a site of interest. Parsimony can be used to assign the ancestral states of internal nodes even in the absence of a root but the state of the last common ancestor and direction of change is unclear. (b) Where both post-root nodes share the same state, parsimony can be used to infer the ancestral state: in this case, leucine. (c) Where post-root nodes differ in their state, the ancestral state is unclear.

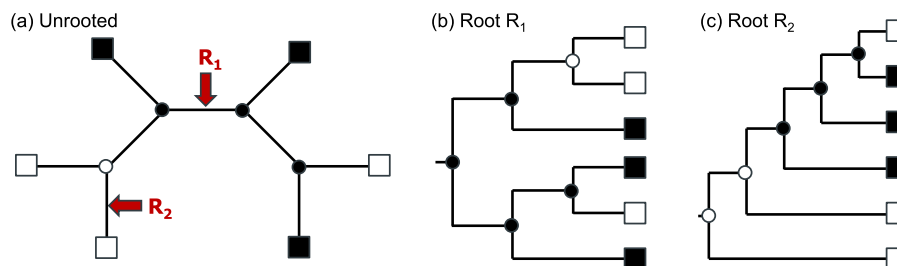


Fig. 7 Resolving homoplasy. (a) This unrooted tree shows homoplasy: the white state appears twice without shared ancestry. Rooting the tree resolves the nature of this homoplasy. (b) If the root is at R_1 the ancestral state is black and the homoplasy has arisen through (parallel) convergent evolution. (c) If the root is at R_2 , the ancestral state is white and the homoplasy is the product of secondary loss, or reversion to the ancestral state.

In **Fig. 6(b)**, the leucine is **paraphyletic**, and arises from common ancestry but does not include all members of the clade. An alternative form of non-monophyly is **polyphyly**, in which the same trait appears in unrelated clades (**Fig. 7**). This is known as **homoplasy**, which is the situation where OTUs share a characteristic (either a phenotypic trait, or a specific nucleotide/residue in the same position) that is not the product of common ancestry (**homology**).

Another common reason to root a phylogenetic tree is to resolve apparent homoplasy. Correctly rooting the phylogeny can resolve whether a dispersed trait is truly homoplastic (rather than paraphyletic as in **Fig. 6(b)**). Furthermore, it can provide insight into the evolutionary history behind the homoplasy. This could be convergent (or parallel) evolution, in which the same trait independently evolves on two lineages (**Fig. 7(b)**), or secondary loss, in which a lineage reverts to the ancestral state (**Fig. 7(c)**).

Concluding Remarks

As with all analysis, there is no one-size-fits-all rule for when to root a tree and/or how carefully it needs to be done. The key thing is to make sure that the rooting is adequate for the biological question being asked – and to be wary of roots originally placed to answer a different biological question that may have different concerns and constraints.

See also: Gene Duplication and Speciation. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Molecular Clock. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life

Further Reading

Felsenstein, J., 2004. *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates, Inc.
 Page, D.M.P., Holmes, E.C., 1998. *Molecular Evolution: a phylogenetic approach*. Oxford: Blackwell Science Ltd.

Biographical Sketch



Rich Edwards is a Senior Lecturer in Bioinformatics in the School of Biotechnology and Biomolecular Sciences (BABS) at the University of New South Wales. Rich trained a geneticist at the University of Nottingham (UK), studying the population genetics of transposable elements in bacteria for his PhD. He moved to Dublin (Ireland) to become a full time bioinformatician in 2001, developing sequence analysis methods for the rational design of biologically active short peptides based on ancestral sequence prediction. This developed into an interest in Short Linear Motifs (SLiMs), which are short regions of proteins that mediate interactions with other proteins. Rich has developed several tools for the prediction and analysis of SLiMs, distributed in the SLiMSuite package. Rich established his lab in 2007 at the University of Southampton (UK) before moving to UNSW in 2013. Research interests in the lab stem from a fascination with molecular basis of evolutionary change and how we can harness the genetic sequence patterns left behind to make useful predictions about contemporary biological systems. Core research is divided between SLiM analysis/prediction and genomics projects. Rich has also collaborated on numerous projects involving DNA and/or protein sequence analysis.

Tree Evaluation and Robustness Testing

Mahendra Mariadassou, INRA, Paris, France

Avner Bar-Hen, CNAM, Paris, France

Hirohisa Kishino, University of Tokyo, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Motivation

Applications of Phylogenies

Molecular phylogenetics is a lively field of research with a number of practical applications. Reconstructing large phylogenies, such as the bird ([Prum et al., 2015](#)) or mammal phylogeny ([Bininda-Emonds et al., 2007](#)), is of intrinsic interest to evolutionary biologists, but those phylogenies are also *the basic structures necessary to think clearly about differences between species, and to analyse those differences statistically* ([Felsenstein, 2004](#)). They arise frequently in comparative genomics, conservation issues ([Bordewich et al., 2008](#)), functional prediction of genes ([Eisen, 1998](#)) and more generally are at the heart of phylogenetic comparative methods ([Revell et al., 2008](#); [Pennell and Harmon, 2013](#)). Most, if not all, applications of phylogenetics have in common that they rely on accurate phylogenetic trees and it is crucial to validate the tree as different trees can lead to vastly different conclusions concerning the origin and evolution of a trait ([Geneva et al., 2015](#)).

With the advent of molecular data and increased formalism of the field ([Gascuel, 2005](#)), modern phylogenetic reconstruction is now essentially a statistical inference problem. Many popular reconstruction softwares such as PhyML ([Guindon and Gascuel, 2003](#)), RAxML ([Stamatakis, 2006](#)), FastTree ([Price et al., 2010](#)) or MrBayes ([Ronquist and Huelsenbeck, 2003](#)) produce a statistical estimate of the tree and we therefore frame validation in a statistical framework. We first discuss the different sources of inaccuracies in the reconstructed tree (Section Sources of Error) and distinguish between natural variability ($\approx$ variance) and modeling errors ($\approx$ bias). We then briefly describe and discuss popular support values (Section Robustness) aimed at validating a tree. In addition to support values, the variability observed in a forest of trees, can be summarized in order to produce robust tree estimates (Section Consensus Methods). Finally, we end this article by reviewing promising developments in the field of continuous distance and highlight their use for validation (Section Detection of Conicting Phylogenetic Signals).

Validating the Tree

A tree is a complex object that encodes the evolutionary relationship of a set of species. Many inference methods return a single focal tree and validation is most often concerned with it. There are two families of validation methods based on (i) the scale at which validation is performed and (ii) whether the tree is considered by itself or with respect to other trees.

- The first family is *local* and grounded on the observation that a tree is uniquely determined by its branches ([Buneman, 1971](#)). A tree can thus be validated by computing a *support value* for each of its branch. Support values tell us which parts of the tree are *reliable*, in a yet to be defined way.
- The second family is *global* and considers the focal tree as a single object rather than a collection of branches. It compares it to a set of alternative trees using statistical tests ([Shimodaira and Hasegawa, 1999](#)). The tests tell us whether the tree is strictly better (i.e. a better fit to the molecular data) than the alternatives.

The two approaches have a different focus but are complimentary. In particular, global tests can be used to compute local support values.

Robust Estimate

Most softwares return the best tree for a given criteria (e.g. likelihood for PhyML and RAxML). Support values tell us whether that tree is reliable and tests tell us whether it's much better than second best or other alternatives.

However, in the presence of outlier data, the superiority of the best tree may lie exclusively in a few data points: slight changes in the molecular data may dramatically change the best tree, with deep clades moving from one position in the tree to another ([Bar-Hen et al., 2008](#)). Since molecular data are inherently noisy, it is interesting to produce *robust* trees that nearly, but not completely, optimize the criteria while being resilient to small changes in the molecular data.

A straightforward way to build a robust estimate is to start from a forest of *good* trees and summarize them in some way to build a *consensus* tree. The forest can consist of trees that are only slightly worse than the best tree (e.g. bayesian consensus) or that are inferred from slightly perturbed data (e.g. bootstrap consensus).

However, more recently, the statistical perspective that observed variability between collections of trees is interesting of itself has gained traction. This has spawned a number of modern methods that utilize the elegant mathematics of tree space to permit new data analysis methods and associated new insights in a number of different contexts.

Sources of Error

Genome-scale analyses indicate that different genes yield conflicting phylogenies. Broadly speaking, these conflicts arise from two main sources: (i) the inability of traditional reconstruction methods to deal with the complexity of molecular evolution (methodological sources) and/or (ii) genuine biological events such as lateral gene transfer (LGT), incomplete lineage sorting and others that lead to different evolutionary histories along the genome (biological sources).

Methodological factors affecting phylogenetic reconstruction include the choice of optimality criterion, limited data availability, taxon sampling and specific assumptions in the modeling of sequence evolution. Biological processes such as the natural selection, small population size, etc may also cause the gene tree to differ from the species tree. The large number of potential explanations for observed incongruences in molecular phylogenetics makes decisions of how to handle them quite difficult (Rokas *et al.*, 2003b).

Biological Sources

Biological source of errors are very diverse and among other we may cite contamination, frameshift events, incorrect annotations, erroneous chimerical sequences, wrong orthology assessment, horizontal gene transfer, gene conversion, incomplete lineage sorting or hybridization, etc. (see Philippe *et al.*, 2017 for a survey).

With the improvement of sequencing technologies, the amount and quality of molecular data used in phylogenetics has drastically increased to the point where sequence quality is superseded by other concerns. For example, inclusion of non-orthologous sequences in a study can have drastic consequences on the final results (Laurin-Lemay *et al.*, 2012; Philippe *et al.*, 2011b). It can, for example, arise through undetected contamination of the genetic material during the sampling or sequencing steps (cross-contamination). It can also arise when paralogs are misidentified as orthologs, a common occurrence given the high frequency of gene/genome duplication, gene conversion and gene loss.

Finally, correct alignments are of crucial importance in phylogenetics as repeatedly pointed out (Morrison and Ellis, 1997; Ogden and Rosenberg, 2006; Talavera and Castresana, 2007; Wong *et al.*, 2008). Yet, due to the lack of tractable models of sequence evolution in the presence of insertion and deletion events (indels), the criteria optimized by alignment software are mostly *ad hoc* and based on the simplistic assumptions that homologous characters should be similar and that indels are rare events.

Modeling Errors

Every model is only a rough sketch of the complex reality of molecular evolution. First and foremost, the assumption of independent and identical evolutionary forces across sites is certainly not true in reality (Adrian, 1986; Yu and Jeffrey, 2006). What happens at one site depends both on its genomic context and on its interaction with other sites in the secondary and ternary structures of the molecule. Some relaxation allow sites to evolve at different sites (Goldman and Yang, 1994). Formally, each site has its own rate, modeled as a random effect drawn from a gamma distribution. Felsenstein and Churchill (1996) extended this approach to allow the rate at one site to depend on neighbouring sites rates using a hidden Markov model. This model captures dependence on the local genomic context (primary structure) and others have also been developed for dependence induced by proximity of sites in the secondary and ternary structure (Robinson *et al.*, 2003) but they are too complex for routine analyses.

For protein evolution, the rate matrices used to model evolution (e.g. JTT, PAM, LG, etc.) are derived by averaging over patterns observed in thousands of sites. However it is clear (Halpern and Bruno, 1998; Parisi and Echave, 2001; Susko *et al.*, 2002) that observed amino acid frequencies strongly deviate from the one expected under the JTT matrix. Lartillot and Philippe (2004) implemented a Bayesian mixture model which allows the inference of site specific rate matrices. These studies show that relaxing the assumption that all sites evolve according to the same rate matrices is crucial for accurate phylogenetic inference. However, such parameter-rich pose problems of their own, which can be difficult to diagnose (see Rannala (2002), for a discussion of over-parameterization in the context of Bayesian phylogenetic inference).

Additionally, many models assume stationarity, homogeneity and reversibility of sequence evolution. This is at odds with observations in archaea, where evolution exhibit a trend towards some nucleotides over very long time scales (Boussau and Gouy, 2006). In bacteria, codon usage bias (ref codon usage bias) can lead to non reversible evolution and different underlying stationarity state frequencies in different parts of the tree. If distantly related lineages begin to display similar state frequencies, these lineages can be artefactually grouped together when using evolution model that do not explicitly allow it (see for example Foster, 2004; Jermini *et al.*, 2004). Finally, phylogenetic trees are probably not the adequate tool to analyse and represent non-vertical transmission and reticulated evolution (Huson and Bryant, 2005).

Robustness

As discussed in Section Motivation, support values are a popular way to validate a focal tree. We present here the most popular ones before describing other methods to validate a tree or fortify it.

Support Values

Bootstrap

Bootstrap values (Felsenstein, 1985) are probably the most popular and easiest to understand support values. Bootstrap involves resampling with replacement from one's molecular data with to create fictional datasets, called *bootstrap replicates*, of the same size. Specifically, the molecular data is typically organized as a multiple sequence alignment (MSA) of s species $\times n$ characters. Since most models assume independent characters, we generate a replicate by sampling n characters, with replacement, from the original MSA and do this B times. Note that in each replicate, some characters are sampled more than once and some left out entirely. The B replicates are used to estimate a forest of B bootstrap trees (one per replicate). Finally the bootstrap value (BP) of a branch of the original tree is its frequency of occurrence in the forest. The process is illustrated in Fig. 1.

Intuitively, the variation obtained by resampling n sites from the original data should be the same as the variation obtained by sampling n new characters. Bootstrap values capture, among other, the *sampling* variability induced by short MSA. When n increases, so does BP in general and it is quite common to achieve very high values for all branches when working on genome-scale alignments (Rokas *et al.*, 2003a).

BP provides a guide for the amount of support a branch has: branches with high BP occur more often and are more reliable than those with low BP . Although it might be tempting to interpret BP as the probability that a branch is present in the (unknown) true tree, this is not the case in general. Zharkikh and Li (1992) showed in a simple case that BP is biased and underestimates that probability. Using simulation studies, Hillis and Bull (1993) showed that BP values as small as 70% could react highly supported branches. Many studies (Felsenstein and Kishino, 1993; Efron *et al.*, 1996; Susko, 2008, 2010) examined the theoretical properties of bootstrap values and concluded that they are indeed biased. This bias is partly induced by the peculiar geometry of tree space (see Billera *et al.*, 2001; Susko (2010); Section Detection of Conicting Phylogenetic Signals).

The final limitation of bootstrap values, shared with other support values based on resampling techniques, is their high computational cost: the budget required to compute B bootstrap trees is B -times higher than the budget for the original tree. Clever implementations can substantially reduce that cost (Stamatakis, 2014) but it remains prohibitive for very large trees.

Posterior probabilities

Posterior Probabilities (PP) are mostly used in a Bayesian framework and similar in spirit bootstrap values. The main difference lies in the forest of trees used to compute support values. Bayesian procedures estimate the posterior distribution of trees. In practice, the distribution is too complex to fully explore and software produce a Monte Carlo Markov Chain (MCMC) sample from the posterior distribution (Yang and Rannala, 1997). The PP of a branch is computed, just like BP , as the probability of occurrence

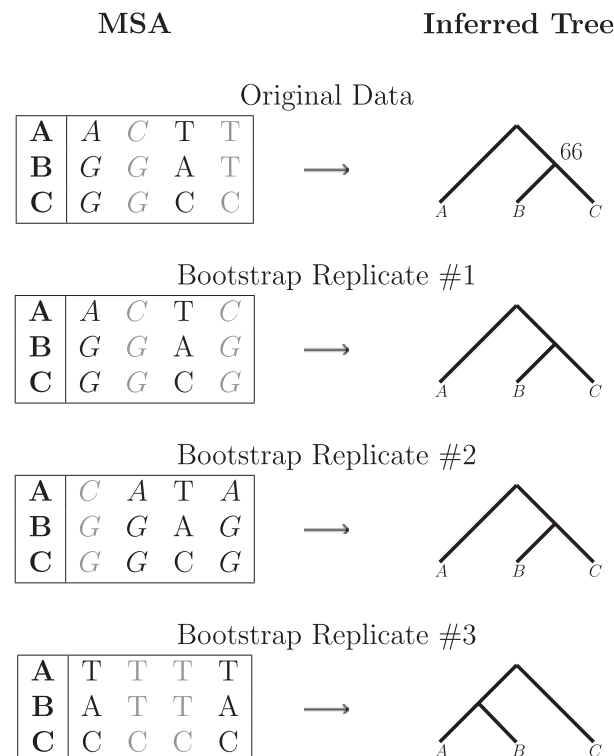


Fig. 1 Principle of the bootstrap for phylogenies. Each character is identified by its color and style. Characters are sampled with replacement to produce bootstrap replicates, which are then used to infer phylogenies. The split $A|BC$ appears in 2 out of 3 bootstrap trees and therefore has a bootstrap value of $BP=2/3$ or 66%.

of that branch in the MCMC sample. MCMC trees constitute a set of highly likely trees for the original dataset. *PP* are easier to interpret than *BP* as they approximate directly the probability that a branch is present in the true tree, given the original data. Furthermore, since MCMC trees are a natural byproduct of the bayesian estimation procedure, there is almost no overhead in computing *PP*.

Unfortunately, *PP* are not immune to bias. Empirical studies found that *PP* are generally higher than *BP* (Anisimova *et al.*, 2011) and sometimes even overconfident, with the “star-tree paradox” (Yang, 2007) being the perfect example of overconfidence. Yang (2007) showed that when the actual tree is a 3-species star tree, so that all 3 potential inner branches are wrong, the bayesian method picks at random whose *PP* goes to 100% when sequence length goes to infinity, whereas one could expect the *PP* to fluctuate around 33%.

Intuitively, *PP* are higher than *BP* because they cover fewer sources of variability. Unlike bootstrap trees, MCMC trees all originate from the same dataset. *PP* are quite good at capturing the lack of phylogenetic signal in the original MSA but not the impact of a few influential characters. For example, outlier characters with a strong effect on the tree or a tiny majority of character that favor one inner branch over another will affect all MCMC trees consistently. By contrast, they will be included in some bootstrap replicates but left out from others leading to more variation among bootstrap trees than among MCMC trees. Finally, in genome-scale context where inaccuracies are more likely to arise from modeling errors than from sampling variability, *PP* are uniformly high and as uninformative as *BP* (Philippe *et al.*, 2011a; Kumar *et al.*, 2012).

Likelihood-based support values

Both *BP* and *PP* quantify the agreement between a focal tree and forest of trees. Likelihood-based supports are fast alternatives that bypass the need for a forest and deal exclusively with the focal tree (Anisimova and Gascuel, 2006).

For any inner branch in the focal tree, there are 3 NNI configurations around that branch: the focal one T_1 and two alternatives T_2 , T_3 (see Fig. 2). If we note $\ell_i = \log \Pr(D|T_i)$ the likelihood of the data under tree i and assume that T_1 is the maximum-likelihood tree, we have $\ell_1 \geq \max(\ell_2, \ell_3)$. Likelihood-based supports values essentially test whether $\delta = \ell_1 - \max(\ell_2, \ell_3)$ is significantly larger than 0.

The most popular support values are:

- the approximate Likelihood Ratio Tests (aLRT) values which evaluates the statistics δ and compares it to $0.5\chi_0^2 + 0.5\chi_1^2$ to compute a p-value. The p-value is then converted into a support value between 1/8 and 1. A branch with high δ will have high support.
- the SH-corrected aLRT (SH-aLRT) values are based on the same idea but use the non-parametric (Shimodaira and Hasegawa, 1999) procedure to compute the p-value of δ .
- Finally approximate bayes (aBayes) is an approximation of the posterior probability of tree T_i computed as:

$$\Pr(T_i|D) = \frac{\Pr(T_i)\Pr(D|T_i)}{\sum_{j=1}^3 \Pr(T_j)\Pr(D|T_j)}$$

with a flat prior $\Pr(T_1) = \Pr(T_2) = \Pr(T_3)$.

All likelihood-based supports (aBayes, aLRT, SH-aLRT) amount to testing if T_1 is significantly better than T_2 and T_3 . By focusing on one branch at the time rather than questioning the whole tree, likelihood-based supports are less conservative than *BP* and *PP*. They can also recycle likelihood computed while estimating the focal tree and are therefore much faster to compute than standard *BP*. Finally, they proved to be accurate in simulations studies (Anisimova *et al.*, 2011). They are the default support values in PhyML (Guindon and Gascuel, 2003).

Outliers in the Data

The aforementioned support values aggregate all variations in the data set and are unable to pinpoint variation due to outliers. The nature of resampling techniques is to use the empirical distribution as a surrogate for the true distribution. However, the empirical distribution may be polluted by outliers, defined here as “entry in the data set that are anomalous with respect to the behavior seen in the majority of the other entries in the data set” (Barnett and Lewis, 1994). This is a common occurrence in multi-locus studies where some characters can evolve according to one a tree, and others according to another tree (Degnan and Rosenberg, 2009). In

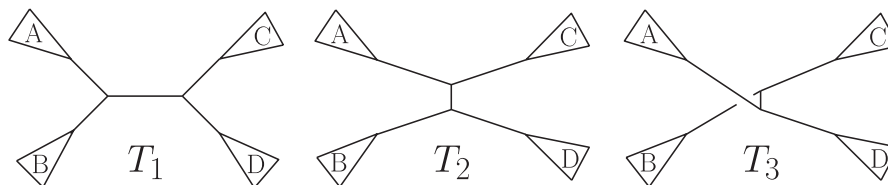


Fig. 2 The maximum likelihood tree (T_1 , left) and its two NNI-alternatives (T_2 , middle and T_3 right) corresponding to different resolutions of the inner branch. Subtrees are sketched as triangles.

that case, a single phylogeny is not a good fit to all the characters and (Swofford *et al.*, 1996) argued that it should be interesting to pinpoint where the phylogeny is not a good fit of the molecular data. Restricting the analyses to congruent characters usually leads to higher support values (Bar-Hen *et al.*, 2008).

Several approaches have been developed to identify outlier characters. Many studies (Rodríguez-Ezpeleta *et al.*, 2007; Burleigh and Mathews, 2004) advocate removing fast-evolving characters which are a well-known cause of misleading phylogenetic signal and long branch attraction (LBA) where distantly related taxa are grouped together in the tree due to parallel or convergent evolution (Felsenstein, 1978). Lopez *et al.* (1999) also suggest to investigate and remove characters with high rate variations (i.e. fast-evolving in some parts of the tree, slow-evolving in others). However both methods assume that good topologies are available to accurately estimate rates, leading to a circularity problem.

Bar-Hen *et al.* (2008) adapted instead inuence functions (Hampel, 1974) to phylogenetics in order to assess the impact of a single site on the likelihood. The main idea consists in removing one character at a time, to create *jackknife* replicates, and to infer a tree on each replicate. Jackknife trees are used to find influential characters whose removal most affect the tree likelihood. Bar-Hen *et al.* (2008) report that influential sites have a strong impact on the topology and correspond mostly to fast evolving sites. All approaches found that removing outliers leads to more stable phylogenies but none is available as a routine in popular softwares.

Taxon Sampling

In phylogenomics studies, it is common to have conflicting trees with support values higher than 95% for all inner branches (Rydin and Källersjö, 2002). This correspond to setups where the estimated tree has a very small estimation variance and differences between inferred trees result mostly from bias and modeling errors. In particular, Swofford *et al.* (1996) argues that adequate taxon sampling is one of the primary factors for accurate phylogenetic estimates, on par with enough sequence data. For example, dense taxon sampling can reduce the impact of LBA by splitting long branches. Similarly, Holland *et al.* (2003) and Shavit *et al.* (2007) showed that the inclusion of an outgroup to the analysis may disrupt the ingroup phylogeny. When there are only a few taxa, but many characters, phylogenetic analysis can produce high support values (BP, PP, etc.) for incorrect or misleading phylogenies (Rokas *et al.*, 2003a; Rokas and Carroll, 2005; Heath *et al.*, 2008).

Analysis of sensitivity to taxon inclusion should be a part of careful and thorough phylogenetic analysis (Heath *et al.*, 2008). Mariadassou *et al.* (2012) defined a the Taxon Inuence Index (TII) to assess the inuence of each taxon on the phylogeny. Using any inference method, we define T^* to be the tree inferred from the complete MSA. Let T_k be a smaller tree, inferred from the alignment deprived of taxon k and T_k^* the tree obtained by pruning taxon k from T^* . The TII is the distance between trees T_k and T_k^* , such that

$$TII(k) = d(T_k, T_k^*)$$

They found that most taxa have small TII(k) and little inuence on the topology whereas a few are highly influential *rogue taxa* and alter the phylogeny in clades even loosely related to their placement in the tree. Aberer *et al.* (2013) use a different approach to find rogue taxa, they start from a forest of trees (e.g. bootstrap trees) and search for a small set of taxa whose pruning increases the agreement between trees in the forest. The method is implemented in the webservice RogueNarok. The rationale in both cases is that reliable trees over smaller taxa sets are preferable over uncertain trees of larger taxa sets. Both methods find that pruning rogue taxa improves accuracy and results in more stable phylogenies with higher support values.

Consensus Methods

Bootstrap, jackknife and bayesian estimation naturally produce a forest of trees with the same species set. But different trees can also be estimated by using different methods or different sources of data. One way to summarize the forest is to *project* it on a focal tree to compute support values (see Section Robustness). Testing Alternatively, one can bypass the focal tree altogether and combine all trees in the forest to get a single tree. That is the purpose of consensus trees methods.

Consensus Trees

Consensus trees are trees that summarize a forest of trees with the same species set. We present here only the *strict* consensus, the *majority rule* consensus and the *extended majority rule* consensus but there are many other consensus (see Bryant (2003) for an extensive survey). The different notions are best understood on an example. Consider the forest featured in Fig. 3 with 40 copies of trees T_1 , 48 of tree T_2 and 12 of tree T_3 . Each tree is completely defined by the bipartitions it induces (or clades for rooted trees). For example, T_1 induces the partitions ABCDEF, ABCDEF and ABEFCD (or clades AB, CD, EF and CDEF if considered as rooted), in addition to all trivial partitions ABCDEF, BACDEF, etc. not shown in the figure. Our three consensus methods scan the forest to build a list of all partitions occurring in the forest with their frequency of occurrence (middle column of Fig. 3). They then select a subset of partitions according to some rules and build a consensus tree from that subset only.

Strict consensus

The *strict consensus* tree (Rohlf, 1982) only uses partitions that appear in all trees, i.e. with a 100% occurrence frequency. The strict consensus is fully compatible with all trees in the forest. However, it is less resolved than any tree and usually too strict. In our

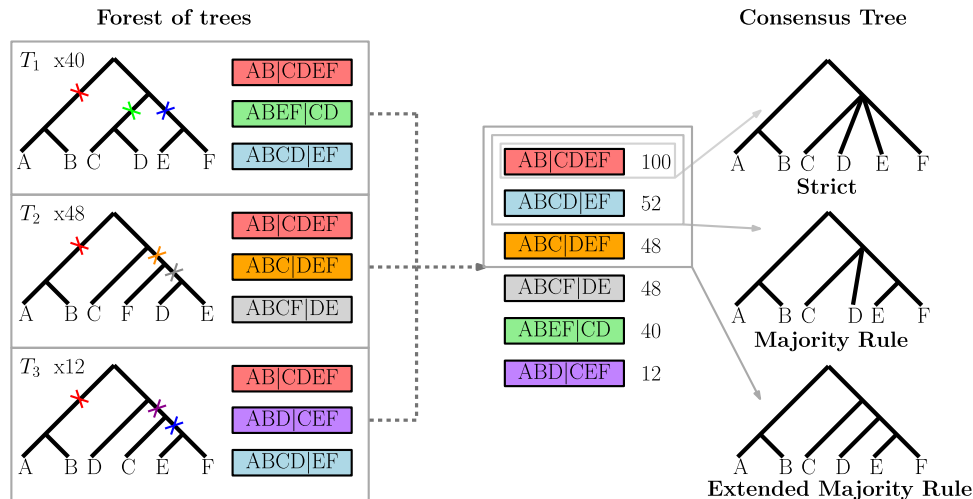


Fig. 3 Left: a forest of 100 trees, corresponding to 3 topologies. Each colored cross correspond to a non trivial bipartition. Middle: Set of all bipartitions, with their occurrence frequencies, found in the forest. Right: Different consensus made up by increasing large sets of partitions.

example, T_1 and T_2 and T_3 only differ in the position of D : if we removed D from all trees, they would be identical. We could therefore expect a branch separating EF from ABC . However, the set $CDEF$ is completely unresolved in the strict consensus.

Majority-rule consensus

The *majority-rule* consensus tree (Margush and McMorris, 1981) relaxes the condition that a bipartition must appear in all trees to be included in the consensus. Instead, it must only appear in *most* trees, i.e. have a occurrence frequency higher than 50%. Although not obvious, all such partitions are pairwise-compatible and can be used to build a proper tree (Buneman, 1971). The majority-rule tree is more resolved than the strict consensus one but is not compatible with all trees in the forest. In our example, the partition $ABCD|EF$ seen in the majority-rule consensus is in conflict with the partition $ABCF|DE$ present in T_2 .

Extended majority-rule consensus

The *extended majority-rule* consensus (Felsenstein, 2005), also called greedy consensus, relaxes the occurrence frequency condition even further. The consensus is build by sequentially adding partition one at a time, in decreasing order of occurrence and only if compatible with previously included partitions, until the tree is fully resolved or no more partitions can be added. Since all partitions with frequency higher than 50% are compatible, they are part of the selection and the greedy consensus is thus a refinement of the majority-rule consensus. In our example, after including partitions $ABCD|EF$ and $AB|CDEF$, we can add either $ABCF|DE$ or $ABEF|CD$. The latter is in conflict with $ABCD|EF$ and thus not included whereas the former is compatible with both partitions and thus included. After addition of $ABCF|DE$, the greedy consensus is fully resolved. Note that the greedy consensus is different from all trees in the forest.

Branch lengths in consensus trees

The strict, majority-rule and extended majority-rule consensus trees only use *topologies* and produce consensus topologies. One way to add branch lengths to that consensus is to take, for each branch, its average length over trees where it is present. This is the approach used in MrBayes (Ronquist and Huelsenbeck, 2003) when building a consensus tree from the sample of posterior trees.

Distance-Based Consensus

The previous consensus methods are easy to understand and implement. Some also have a theoretical grounding that leads to generalizations. Barthélemy and McMorris (1986) showed that the majority-rule consensus is the *center* of the forest in the sense that it minimizes the sum of Robinson-Foulds distances (Robinson and Foulds, 1979) to all trees in the forest. It is thus the forest *median tree*. One could minimize total squared distance to all trees in the forest to find the *mean tree*.

The Robinson-Foulds is but one of many distances between trees (see St. John (2017) for an excellent review) and one can define other consensus similarly as the forest mean or median tree for some distance between tree. Although seducing, this approach suffered in the past from two shortcomings that severely limit its use. First, it was not obvious that the mean, or median, tree is well-defined or unique for some distances (Billera et al., 2001). Second, even when the mean was well-defined, there was no routine to compute it in practice. Recent progress in the field (Miller et al., 2015a) have made it easier to compute sophisticated distances and use them to validate trees.

Detection of Conflicting Phylogenetic Signals

The very idea of a consensus tree relies on the existence of a *center* around which trees are distributed. However, trees that are far away from the consensus can be the sign of either high variability or conflicting signals. While high variability implies weak phylogenetic signal, conflicting signals implies incongruence in the underlying data. Most distances, such as RF, are discrete and lead to very coarse information about the tree distribution. By contrast, continuous distances are a natural way to characterize variance and distinguish high variability from conflicting signals. Moreover continuous distance lend themselves nicely to other standard aspects of inferential statistics such as confidence sets and hypothesis testing.

The geometric model of Billera *et al.* (2001) allows one to compare phylogenetic trees with the same taxa set (of size m) in a quantitative way. This space has a natural metric, giving a way of measuring *continuous* distances between phylogenies and providing some procedures for averaging or combining several trees with the same leaves. This geometry also shows which trees appear within a fixed distance of a given tree and enables one to build confidence convex hulls from a set of trees. It also provides a justification for calling outliers in a collection of otherwise congruent trees and limit the consensus to those congruent tree.

Tree Space Definition

The distance $d(T, T')$ between two trees T and T' accounts for differences with respect to both their tree topologies (branching structure) and branch lengths. The space is constructed by representing each of the $(2m - 3)!!$ possible tree topologies by a single non-negative Euclidean orthant of dimension $m - 3$ (the largest possible number of internal branches). The orthants are then “glued together” along appropriate axes. Specifically, nearest neighbor interchange (NNI) topologies lie in adjacent non-negative orthants along the boundary corresponding to the collapse of the relevant NNI edge.

For two trees with different topologies, the BHV distance is the length of the shortest path that link them in the treespace. The length of any path can be computed by calculating the Euclidean distance of the path restricted to each orthant that it passes through, and summing these lengths. The shortest path is called a geodesic, and will pass from one orthant to the next orthant through lower-dimensional boundaries corresponding to less resolved trees. Since the space is nonpositively curved, the geodesics are unique (Fig. 4).

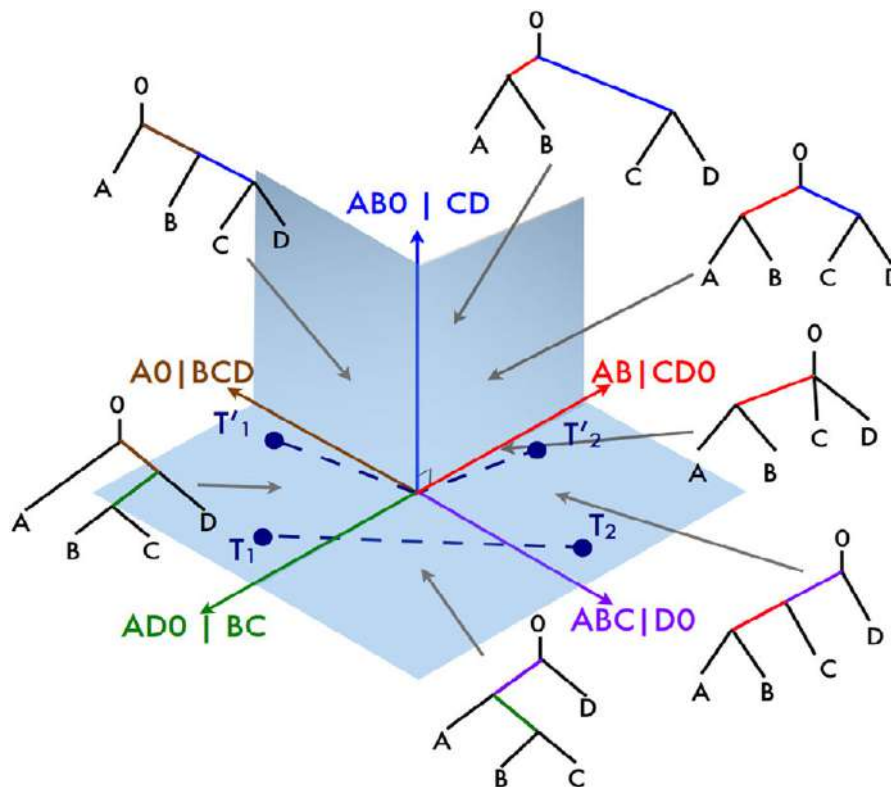


Fig. 4 Orthants in the BHV space of trees with 4 leaves and examples of two geodesics. As illustrated, the geodesics are not identical and therefore depend on both the topology and the actual branch lengths. Reproduced from Brown, D.G., Owen, M., 2017. Mean and variance of phylogenetic trees. arXiv preprint [arXiv:1708.00294](https://arxiv.org/abs/1708.00294).

In Euclidean space, the Fréchet mean is the point minimizing the sum of the squared distances to the sample points, and is equivalent to the coordinate-wise average of the sample points. Similarly, the Fréchet mean tree is the tree minimizing the sum of squared BHV distances to a set of trees and that sum is the variance which quantifies how spread out the forest is (Miller *et al.* (2015b) and Brown and Owen (2017)). Billera *et al.* (2001) showed that mean is unique and is not necessarily a refinement of the majority-rule consensus.

Use of BHV Distance

BHV distances can be used in much the same way as euclidian distance and the standard tools of multivariate statistics can be used to analyse the forest. For example, Barden *et al.* (2014) proved a Central Limit Theorem in the BHV space that can be used to either detect weakly and strongly supported edges in the mean tree or to test hypotheses on some portion of the tree. One of their main tool is the so-called log-map which projects the BHV metric space to the usual euclidian space and allows one to visualize the forest as a scatter plot to gain some understanding of the estimate uncertainty.

In the same spirit, Willis (2016) use the distance matrix to estimate the principal directions of variability *via* principal components analysis (PCA). The axes of the $\mathbb{R}^m$ ellipsoid indicate the relative directions of precision, and the ellipsoid can be shrunk to be wholly contained in the same orthant as a focal tree $\hat{T}$. This gives an unambiguous indication of the relative confidence in the edges of $\hat{T}$. Note that the procedure is also explicit about the trees contained in the confidence set for a given confidence level α . de Vienne *et al.* (2012) use a similar approach based on a different mapping of the trees to a euclidian space and use multiple co-inertia analysis (MCOA) instead of PCA to detect the principal directions of variability.

Finally Weyenberg *et al.* (2014) propose a non-parametric estimator of the distribution that generated the forest $T_1, \dots, T_n$. This estimator can be viewed as a refined version of histogram-based estimation of a density. The kernel function, is a non-negative function defined on pairs of trees, which measures how similar two trees are. Kernel density estimation use the fact that points close to sample points tend to have higher likelihood than distant outlier points. The ultimate goal is to detect outlier trees, T_j , which are not actually drawn from the true distribution.

Conclusions

Nowadays efficient algorithms permit to compute means of continuous distances in a reasonable time. Moreover a huge amount of work has been done to derive mathematical properties of these distances. Therefore it is possible to go from a theoretical work to applications. For example, Kendall and Colijn (2016) use tree mapping to show that 3 genes of Ebolavirus have markedly different phylogenies. This result is quite typical in genomic-scales analysis where different loci have different evolution history. This is a natural consequence of the gene trees/species tree problem whereby the evolutionary history of a gene can be different from the species tree (Degnan and Rosenberg, 2009). This has spanned the entire research field of reconciliation devoted to the reconstruction of species tree from sets of gene tree.

Distances between trees can help cluster trees and/or loci into congruent groups with the intent of performing one inference per group to infer more robust phylogenetic estimates. In an era of cheap and abundant sequences, tree validation is as much a matter of computing support values as a matter of validating and making sure there are not too many conflicting signals in the molecular data (taxa set, gene set) used for tree reconstruction.

See also: Comparative and Evolutionary Genomics. Evolutionary Models. Gene Duplication and Speciation. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Introduction to the Non-Parametric Bootstrap. Molecular Clock. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Phylogenetic Tree Rooting

References

- Aberer, A.J., Krompass, D., Stamatakis, A., 2013. Pruning rogue taxa improves phylogenetic accuracy: An efficient algorithm and webservice. *Syst Biol* 62 (1), 162–166. doi:10.1093/sysbio/sys078.
- Adrian, P.B., 1986. CpG-rich islands and the function of dna methylation. *Nature* 321 (6067), 209–213.
- Anisimova, M., Gascuel, O., 2006. Approximate likelihood-ratio test for branches: A fast, accurate, and powerful alternative. *Syst Biol* 55 (4), 539–552. doi:10.1080/10635150600755453.
- Anisimova, M., Gil, M., Dufayard, J.-F., Dessimoz, C., Gascuel, O., 2011. Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes. *Syst Biol* 60 (5), 685–699. doi:10.1093/sysbio/syr041.
- Barden, D., Le, H., Owen, M., 2014. Limiting behaviour of fréchet means in the space of phylogenetic trees. *Ann Inst Stat Math*. 1–31.
- Bar-Hen, A., Mariadassou, M., Poursat, M.-A., Van-den-koornhuysen, P., 2008. Influence function for robust phylogenetic reconstructions. *Mol Biol Evol* 25 (5), 869–873. doi:10.1093/molbev/msn030.
- Barnett, V., Lewis, T., 1994. *Outliers in Statistical Data*. John Wiley & Sons.
- Barthélemy, J.-P., McMorris, F.R., 1986. The median procedure for n-trees. *J Classif* 3 (2), 329–334. doi:10.1007/BF01894194.
- Billera, L.J., Holmes, S.P., Vogtmann, K., 2001. Geometry of the space of phylogenetic trees. *Adv Appl Math* 27, 733–767. Available at: <http://www.math.cornell.edu/~billera/papers/treespace.pdf>.

- Bininda-Emonds, O.R.P., Cardillo, M., Jones, K.E., *et al.*, 2007. The delayed rise of present-day mammals. *Nature* 446, 507–512. doi:10.1038/nature05634.
- Bordewich, M., Rodrigo, A.G., Sempel, C., 2008. Selecting taxa to save or sequence: Desirable criteria and a greedy solution. *Syst Biol* 57 (6), 825–834. doi:10.1080/10635150802552831.
- Boussau, B., Gouy, M., 2006. Efficient likelihood computations with nonreversible models of evolution. *Systematic Biology* 55 (5), 756–768.
- Brown, D.G., Owen, M., 2017. Mean and variance of phylogenetic trees. arXiv preprint. arXiv:1708.00294.
- Bryant, D., 2003. A classification of consensus methods for phylogenetics. *Bioconsensus* 61, 163–183.
- Buneman, P., 1971. *Mathematics the Archeological and Historical Sciences, Chapter: The Recovery of Trees from Measures of Dissimilarity*. Edinburgh University Press. (ISBN: 9780852242131).
- Burleigh, J.G., Mathews, S., 2004. Phylogenetic signal in nucleotide data from seed plants: Implications for resolving the seed plant tree of life. *Am J Bot* 91 (10), 1599–1613. doi:10.3732/ajb.91.10.1599. Available at: <http://www.amjbot.org/content/91/10/1599.abstract>.
- Degnan, J.H., Rosenberg, N.A., 2009. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol Evol* 24 (6), 332–340. doi:10.1016/j.tree.2009.01.009.
- de Vienne, D.M., Ollier, S., Aguilera, G., 2012. Phylo-mcoa: A fast and efficient method to detect outlier genes and species in phylogenomics using multiple co-inertia analysis. *Mol Biol Evol* 29 (6), 1587–1598.
- Efron, B., Halloran, E., Holmes, S., 1996. Bootstrap confidence levels for phylogenetic trees. *Proc Natl Acad Sci USA* 93 (14), 7085–7090. Available at: <http://www.pnas.org/cgi/content/full/93/23/13429>.
- Eisen, J.A., 1998. Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Res* 8 (3), 163–167.
- Felsenstein, J., 1978. Cases in which parsimony or compatibility methods will be positively misleading. *Syst Zool* 27 (4), 401–410. Available at: http://www.molecularrevolution.org/si/resources/references/files/Felsenstein_1978.pdf.
- Felsenstein, J., 1985. Confidence limits on phylogenies: An approach using the bootstrap. *Evolution* 39 (4), 783–791. doi:10.2307/2408678. Available at: [http://links.jstor.org/sici?sici=0014-3820\(198507\)39:4%3C783:CLOPAA%3E2.0.CO;2-L](http://links.jstor.org/sici?sici=0014-3820(198507)39:4%3C783:CLOPAA%3E2.0.CO;2-L).
- Felsenstein, J., 2005. Phylip (phylogeny inference package) version 3.6. Distributed by the author.
- Felsenstein, J., Kishino, H., 1993. Is there something wrong with the bootstrap on phylogenies? A reply to hillis and bull. *Syst Biol* 42 (2), 193–200. doi:10.2307/2992541. Available at: [http://links.jstor.org/sici?sici=1063-5157\(199306\)42%3A2%3C193%3AITSWwt%3E2.0.CO%3B2-Y](http://links.jstor.org/sici?sici=1063-5157(199306)42%3A2%3C193%3AITSWwt%3E2.0.CO%3B2-Y).
- Felsenstein, J., 2004. *Inferring Phylogenies*. Sinauer Associates.
- Felsenstein, J., Churchill, G.A., 1996. A hidden markov model approach to variation among sites in rate of evolution. *Mol Biol Evol* 13 (1), 93–104.
- Foster, P.G., 2004. Modeling compositional heterogeneity. *Syst Biol* 53 (3), 485–495.
- Gascuel, O., 2005. *Mathematics of Evolution and Phylogeny*. Oxford University Press.
- Geneva, A.J., Hilton, J., Noll, S., Glor, R.E., 2015. Multilocus phylogenetic analyses of hispaniolan and bahamian trunk anoles (distichus species group). *Mol Phylogenet Evol* 87, 105–117.
- Goldman, N., Yang, Z., 1994. A codon-based model of nucleotide substitution for protein-coding dna sequences. *Mol Biol Evol* 11 (5), 725–736.
- Guindon, S., Gascuel, O., 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol* 52 (5), 696–704. doi:10.1080/10635150390235520. Available at: <http://www.informaworld.com/smpp/ftinterface~content=a713850337~fulltext=713240930>.
- Halpern, A.L., Bruno, W.J., 1998. Evolutionary distances for protein-coding sequences: Modeling site-specific residue frequencies. *Mol Biol Evol* 15 (7), 910–917.
- Hampel, F.R., 1974. The influence curve and its role in robust estimation. *J Am Stat Assoc* 69, 383–393.
- Heath, T.A., Hedtke, S.M., Hillis, D.M., 2008. Taxon sampling and the accuracy of phylogenetic analyses. *J Mol Evol* 46, 239–257. Available at: http://www.plantsystematics.com/qikan/epaper/hb_zhaiyao.asp.
- Hillis, D.M., Bull, J.J., 1993. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *Syst Biol* 42 (2), 182–192. doi:10.2307/2992540.
- Holland, B.R., Penny, D., Hendy, M.D., 2003. Outgroup misplacement and phylogenetic inaccuracy under a molecular clock-a simulation study. *Syst Biol* 52 (2), 229–238. doi:10.1080/10635150390192771. Available at: <http://www.informaworld.com/smpp/content~content=a713850188~db=all~order=page>.
- Huson, D.H., Bryant, D., 2005. Application of phylogenetic networks in evolutionary studies. *Mol Biol Evol* 23 (2), 254–267.
- Jermiin, L.S., Ho, S.Y.W., Ababneh, F., Robinson, J., Larkum, A.W.D., 2004. The biasing effect of compositional heterogeneity on phylogenetic estimates may be underestimated. *Syst Biol* 53 (4), 638–643.
- Kendall, M., Colijn, C., 2016. Mapping phylogenetic trees to reveal distinct patterns of evolution. *Mol Biol Evol* 33 (10), 2735–2743. doi:10.1093/molbev/msw124.
- Kumar, S., Filipowski, A.J., Battistuzzi, F.U., Kosakovsky Pond, S.L., Tamura, K., 2012. Statistics and truth in phylogenomics. *Mol Biol Evol* 29 (2), 457–472. doi:10.1093/molbev/msr202.
- Lartillot, N., Philippe, H., 2004. A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Mol Biol Evol* 21 (6), 1095–1109.
- Laurin-Lemay, S., Brinkmann, H., Philippe, H., 2012. Origin of land plants revisited in the light of sequence contamination and missing data. *Curr Biol* 22 (15), R593–R594.
- Lopez, P., Forterre, P., Philippe, H., 1999. The root of the tree of life in the light of the co-varion model. *J Mol Evol* 49 (4), 496–508. Available at: <http://www.springerlink.com/content/hwle29fxv74ra6xy/>.
- Margush, T., McMorris, F.R., 1981. Consensus n-trees. *Bull Math Biol* 43 (2), 239–244. doi:10.1007/BF02459446.
- Mariadassou, M., Bar-Hen, A., Kishino, H., 2012. Taxon influence index: Assessing taxon-induced incongruities in phylogenetic inference. *Syst Biol* 61 (2), 337–345. doi:10.1093/sysbio/syr129.
- Miller, E., Owen, M., Provan, J.S., 2015a. Polyhedral computational geometry for averaging metric phylogenetic trees. *Adv Appl Math* 68, 51–91. Available at: <https://doi.org/10.1016/j.aam.2015.04.002>; <http://www.sciencedirect.com/science/article/pii/S0196885815000470>.
- Miller, E., Owen, M., Provan, J.S., 2015b. Polyhedral computational geometry for averaging metric phylogenetic trees. *Adv Appl Math* 68, 51–91.
- Morrison, D.A., Ellis, J.T., 1997. Effects of nucleotide sequence alignment on phylogeny estimation: A case study of 18s rDNAs of apicomplexa. *Mol Biol Evol* 14 (4), 428–441.
- Ogden, T.H., Rosenberg, M.S., 2006. Multiple sequence alignment accuracy and phylogenetic inference. *Syst Biol* 55 (2), 314–328.
- Parisi, G., Echave, J., 2001. Structural constraints and emergence of sequence patterns in protein evolution. *Mol Biol Evol* 18 (5), 750–756.
- Pennell, M.W., Harmon, L.J., 2013. An integrative view of phylogenetic comparative methods: Connections to population genetics, community ecology, and paleobiology. *Ann N.Y. Acad Sci* 1289, 90–105. doi:10.1111/nyas.12157.
- Philippe, H., Brinkmann, H., Lavrov, D.V., *et al.*, 2011a. Resolving difficult phylogenetic questions: Why more sequences are not enough. *PLOS Biol* 9 (3), e1000602. doi:10.1371/journal.pbio.1000602.
- Philippe, H., Brinkmann, H., Lavrov, D.V., *et al.*, 2011b. Resolving difficult phylogenetic questions: Why more sequences are not enough. *PLOS Biology* 9 (3), e1000602.
- Philippe, H., De Vienne, D.M., Ranwez, V., *et al.*, 2017. Pitfalls in supermatrix phylogenomics. *Eur J Taxon* 283, 1–25.
- Price, M.N., Dehal, P.S., Arkin, A.P., 2010. Fasttree 2-approximately maximum-likelihood trees for large alignments. *PLoS One* 5 (3), e9490. doi:10.1371/journal.pone.0009490.
- Prum, R.O., Berv, J.S., Dornburg, A., *et al.*, 2015. A comprehensive phylogeny of birds (Aves) using targeted next-generation DNA sequencing. *Nature* 526, 569–573. doi:10.1038/nature15697.
- Rannala, B., 2002. Identifiability of parameters in mcmc bayesian inference of phylogeny. *Syst Biol* 51 (5), 754–760.
- Revell, L.J., Harmon, L.J., Collar, D.C., 2008. Phylogenetic signal, evolutionary process, and rate. *Syst Biol* 57 (4), 591–601. doi:10.1080/10635150802302427.
- Robinson, D.F., Foulds, L.R., 1979. Comparison of weighted labelled trees. *Lectures Note in Mathematics*, vol. 748. Berlin: Springer-Verlag, pp. 119–126.
- Robinson, D.M., Jones, D.T., Kishino, H., Goldman, N., Thorne, J.L., 2003. Protein evolution with dependence among codons due to tertiary structure. *Mol Biol Evol* 20 (10), 1692–1704. doi:10.1093/molbev/msg184.

- Rodríguez-Ezpeleta, N., Brinkmann, H., Roure, B., *et al.*, 2007. Detecting and overcoming systematic errors in genome-scale phylogenies. *Systematic Biology* 56 (3), 389–399. doi:10.1080/10635150701397643.
- Rohlf, F.J., 1982. Consensus indices for comparing classifications. *Mathematical Biosciences* 59 (1), 131–144. Available at: [https://doi.org/10.1016/0025-5564\(82\)90112-2](https://doi.org/10.1016/0025-5564(82)90112-2); <http://www.sciencedirect.com/science/article/pii/0025556482901122>.
- Rokas, A., Carroll, S.B., 2005. More genes or more taxa? The relative contribution of gene number and taxon number to phylogenetic accuracy. *Mol Biol Evol* 22 (5), 1337–1344. doi:10.1093/molbev/msi121.
- Rokas, A., Williams, B.L., King, N., Carroll, S.B., 2003a. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature* 425 (6960), 798–804. doi:10.1038/nature02053.
- Rokas, A., Williams, B.L., King, N., Carroll, S.B., 2003b. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature* 425 (6960), 798.
- Ronquist, F., Huelsenbeck, J.P., 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* 19 (12), 1572–1574. Available at: <http://bioinformatics.oxfordjournals.org/cgi/reprint/19/12/1572>.
- Rydin, C., Kallersjö, M., 2002. Taxon sampling and seed plant phylogeny. *Cladis-tics* 18 (5), 485–513. doi:https://doi.org/10.1016/S0748-3007(02)00104-4. Available at: <http://www.sciencedirect.com/science/article/pii/S0748300702001044>. ISSN: 0748-3007
- Shavit, L., Penny, D., Hendy, M.D., Holland, B.R., 2007. The problem of rooting rapid radiations. *Mol Biol Evol* 24 (11), 2400–2411. doi:10.1093/molbev/msm178.
- Shimodaira, H., Hasegawa, M., 1999. Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Mol Biol Evol* 16 (8), 1114–1116.
- St. John, K., 2017. Review paper: The shape of phylogenetic treespace. *Syst Biol* 66 (1), e83–e94. doi:10.1093/sysbio/syw025.
- Stamatakis, A., 2006. Raxml-vi-hpc: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22 (21), 2688–2690. doi:10.1093/bioinformatics/btl446.
- Stamatakis, A., 2014. Raxml version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30 (9), 1312–1313. doi:10.1093/bioinformatics/btu033.
- Susko, E., 2008. On the distributions of bootstrap support and posterior distributions for a star tree. *Syst Biol* 57 (4), 602–612. doi:10.1080/10635150802302468.
- Susko, E., 2010. First-order correct bootstrap support adjustments for splits that allow hypothesis testing when using maximum likelihood estimation. *Mol Biol Evol*. doi:10.1093/molbev/msq048.
- Susko, E., Inagaki, Y., Field, C., Holder, M.E., Roger, A.J., 2002. Testing for differences in rates-across-sites distributions in phylogenetic subtrees. *Mol Bio Evol* 19 (9), 1514–1523.
- Swofford, D.L., Olsen, G.J., Waddell, P.J., Hillis, D.M., 1996. Phylogenetic inference. In: Hillis, D.M., Moritz, C., Mable, B.K. (Eds.), *Molecular Systematics*, second ed. Sunderland, MA: Sinauer Associates, Inc., pp. 407–514.
- Talavera, G., Castresana, J., 2007. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Syst Biol* 56 (4), 564–577.
- Weyenberg, G., Huggins, P.M., Schardl, C.L., Howe, D.K., Yoshida, R., 2014. Kdetrees: Non-parametric estimation of phylogenetic tree distributions. *Bioinformatics* 30 (16), 2280–2287.
- Willis, A., 2016. Confidence sets for phylogenetic trees. arXiv preprint. arXiv:1607.08288.
- Wong, K.M., Suchard, M.A., Huelsenbeck, J.P., 2008. Alignment uncertainty and genomic analysis. *Science* 319 (5862), 473–476.
- Yang, Z., 2007. Fair-balance paradox, star-tree paradox, and bayesian phylogenetics. *Mol Biol Evol* 24 (8), 1639–1655. doi:10.1093/molbev/msm081.
- Yang, Z., Rannala, B., 1997. Bayesian phylogenetic inference using DNA sequences: A markov chain monte carlo method. *Mol Biol Evol* 14 (7), 717–724. Available at: <http://mbe.oxfordjournals.org/cgi/reprint/14/7/717>.
- Yu, J., Jeffrey, L., 2006. Thorne. Dependence among sites in RNA evolution. *Mol Biol Evol* 23 (8), 1525–1537. doi:10.1093/molbev/msl015.
- Zharkikh, A., Li, W.H., 1992. Statistical properties of bootstrap estimation of phylogenetic variability from nucleotide sequences. i. four taxa with a molecular clock. *Mol Biol Evol* 9 (6), 1119–1147. Available at: <http://mbe.oxfordjournals.org/cgi/reprint/9/6/1119>.

Further Reading

- Bryant, D., 2003. A classification of consensus methods for phylogenetics. *Bioconsensus* 61, 163–183.
- Degnan, J.H., Rosenberg, N.A., 2009. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol Evol* 24 (6), 332–340. doi:10.1016/j.tree.2009.01.009.
- Felsenstein, J., 2004. *Inferring Phylogenies*. Sinauer Associates.
- Philippe, H., De Vienne, D.M., Ranwez, V., *et al.*, 2017. Pitfalls in supermatrix phylogenomics. *Eur J Taxon* 283, 1–25.
- St. John, K., 2017. Review paper: The shape of phylogenetic treespace. *Syst Biol* 66 (1), e83–e94. doi:10.1093/sysbio/syw025.
- Susko, E., 2010. First-order correct bootstrap support adjustments for splits that allow hypothesis testing when using maximum likelihood estimation. *Mol Biol Evol*. doi:10.1093/molbev/msq048.
- Yang, Z., 2007. Fair-balance paradox, star-tree paradox, and Bayesian phylogenetics. *Mol Biol Evol* 24 (8), 1639–1655. doi:10.1093/molbev/msm081.

Applications of the Coalescent for the Evolutionary Analysis of Genetic Data

Miguel Arenas, University of Vigo, Vigo, Spain

© 2019 Elsevier Inc. All rights reserved.

Introduction

The coalescent (Kingman, 1982) is a stochastic model to simulate the evolutionary history of a sample of a large population. Instead of simulating the entire population and then sampling, the coalescent simulates the history of the sample to its most recent common ancestor (MRCA). The population size is considered in the coalescent but only to determine the probability of coalescent events, not to simulate the population history (Section “The Coalescent Theory and its Extensions”). As a consequence, coalescent simulations are generally rapid, which is convenient for a variety of applications.

First coalescent approaches were rapidly extended to mimic diverse fundamental evolutionary processes such as recombination (Hudson and Kaplan, 1988; Kaplan *et al.*, 1991; Hudson, 1990), population structure with migration (Hudson, 1998; Kaplan *et al.*, 1991; Hudson, 1990) and selection (Hudson and Kaplan, 1988, 1995; Kaplan *et al.*, 1988, 1991; Hudson, 1990) (Section “The Coalescent Theory and its Extensions”), with goals but also with limitations (Section “Goals and Limitations of the Coalescent”). All these extensions were implemented into a variety of evolutionary frameworks (Section “Coalescent Simulators”). The coalescent can be applied to address diverse evolutionary questions (Section “Applications of the Coalescent”). For example, it can be used for hypothesis testing, validation of analytical methods, selection of evolutionary models and estimation of evolutionary parameters.

This article revises the current state of the coalescent with a special focus on its applications. First, it provides a brief introduction on the fundamental insights of the standard coalescent and its extensions (some complementary reviews are provided). Next it describes goals and limitations of the coalescent, currently available coalescent simulators and applications in population genetics and evolution. Finally, the article provides suggestions to design a coalescent-based experiment, including some practical examples.

The Coalescent Theory and Its Extensions

The standard coalescent only requires sample size (k) and effective population size (N) to simulate the evolutionary history of a sample. This simulation is based on a backwards in time process, where the sample is evolved from the present to the past (Fig. 1). The probability that two individuals descend from a common ancestor of the population is $1/2N$. Thus, the probability of a coalescent event between two lineages of a sample of k individuals can be obtained by calculating, $1/2N \times k(k-1)/2 = k(k-1)/4N$, because there are $k(k-1)/2$ pairs of individuals in a sample of k individuals (note that if k is 2 the result is again $1/2N$). Fig. 1 presents the probabilities of coalescent events in an illustrative example. Next, considering that the time for a coalescent event follows an exponential distribution (for $k=2$, $P(t+1) = (1 - (1/2N))^t \times (1/2N) \approx (1/2N)e^{-t/2N}$), the expectation of this time is

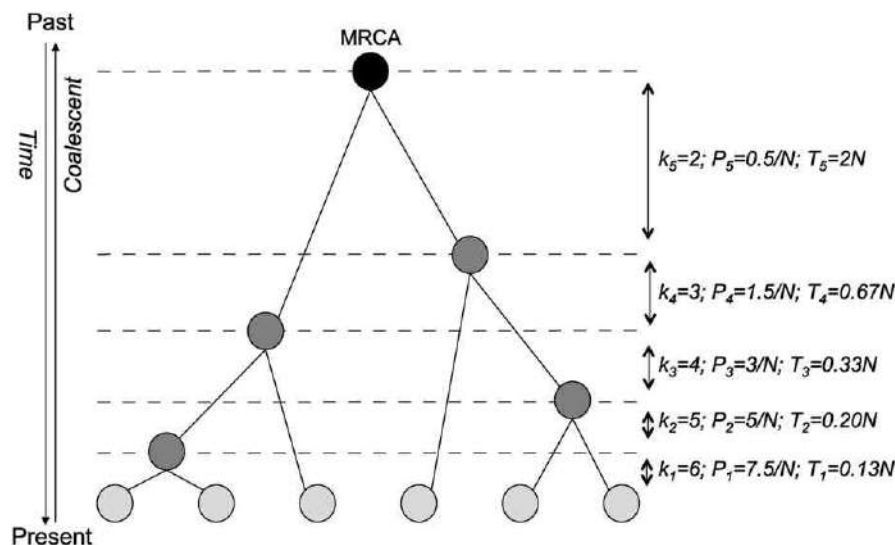


Fig. 1 Illustrative example of the standard coalescent. Simulation of a coalescent tree from a sample of six individuals (circles in clear grey) and population size N . Coalescent events generate internal nodes (circles in dark grey) that are simulated going backwards in time until reaching only one active lineage, the MRCA (circle in black). For each period of time before a coalescent event the following information is shown: number of active lineages k , probability of a coalescence P and expected time for a coalescence T .

$E(T_k) = 4N/k(k-1)$. These expressions require $4N/k(k-1) \gg 1$ that only works when the sample size is much smaller than the population size, which is one of the most important assumptions of the coalescent. The time to the MRCA (TMRCA) can be obtained by summing coalescence times from k to 2 individuals, $E(T_{MRCA}) = E(T_k) + E(T_{k-1}) + E(T_{k-2}) + \dots + E(T_{k=2}) = 4N(1 - 1/k)$. Interestingly, note that the last coalescent event (when $k=2$) requires $2N$ generations, which is almost half of the time required to reach the MRCA. Further details about the mathematical expressions of the coalescent can be found in the following reviews (Nordborg, 2007; Neuhauser and Tavaré, 2001; Hein *et al.*, 2005). It follows with an overview on the most relevant extensions of the standard coalescent.

The Coalescent With Variable Population Size

Fluctuations in population size through time are common and, consequently, one of the first extensions of the coalescent was oriented to consider this aspect. As noted above, the probability or expected time for a coalescent event depends on the population size. The larger is the population size, the smaller is the probability of a coalescent event or the longer is the expected time for a coalescent event, which leads to longer branch lengths (Fig. 2(B)). Thus, if the population size increased over time (it decreased if we go back in time) longer branches are expected to appear near present (coalescences mainly appear in the past) leading to a star-like tree (Fig. 2(A)). By contrast, if population size decreases through time, longer branches are expected to appear in the past, leading to trees with a comb-like shape (Fig. 2(C)).

The variation of the population size through time can be systematic or irregular. A systematic variation can be modeled with only a few parameters (i.e., the commonly used population growth rate g where $N(t) = N_{(0)} \times e^{-gt}$) that can be easily incorporated to the coalescent. However, the population size could also vary in a different manner at different periods of time (i.e., as it occurs many times in virus populations (Lemey *et al.*, 2006)). Since these demographic periods may not be coincident with times between coalescent events, the implementation of this scenario is slightly more complex (details in Hein *et al.* (2005)). An interesting scenario is a population bottleneck (an important decline in population size during a period of time). During a bottleneck the population size can be very small and, consequently, the rate of coalescence will increase reducing the TMRCA (Hein *et al.*, 2005).

The Coalescent With Population Structure and Migration

Sometimes individuals may only mate with others that present particular features (i.e., proximity or phenotype). Under these situations, one can devise a model where a population structure defined as a set of subpopulations (or demes) presents mating (coalescence) only between individuals of the same deme but with possible migration of individuals between demes (Fig. 3). If the sample comes from different demes and there is not a connection between them (i.e., lack of a demes tree and/or migration between demes) the coalescent cannot be applied because a MRCA common to all demes cannot be reached: migration or demes coalescence is required to coalesce all the lineages. Migration rate can affect both topology and branch lengths. The smaller is the migration rate, the longer is the expected TMRCA (Hein *et al.*, 2005). Nevertheless, migration can be modeled as a process independent of coalescence and recombination and, consequently, it can be easily considered to determine the probability or expected time of coalescent and migration events (Hein *et al.*, 2005; Notohara, 1990). For mathematical details about probabilities and expected times of migration and coalescence events the reader is referred to Hein *et al.* (2005). The choice of a receiving deme in a migration event depends on the specified subdivision (migration) model and migration rates (Fig. 4(A)). The most used migration model in the coalescent is the island model where all demes can exchange individuals with all demes (Hudson, 1998). The stepping stone migration model (Kimura and Weiss, 1964), where demes only exchange migrants with neighboring demes, is also well-established. Another useful model is the continent-island model that considers migration between a central deme and peripheral demes (Wright, 1931). The coalescent can also be structured considering a demes tree (Nielsen and Wakeley, 2001; Fig. 4(B)). Under this scenario, demes share a common ancestor and migration is not mandatory to reach the MRCA.

The Coalescent With Recombination

Recombination is a fundamental evolutionary force for most organisms, especially virus and bacteria (Perez-Losada *et al.*, 2015). Recombination events generate an ancestral recombination graph (ARG) (Arenas, 2013b; Griffiths and Marjoram, 1997, 1996)

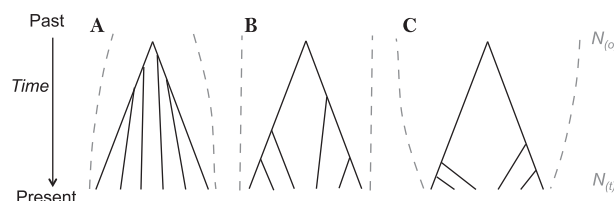


Fig. 2 Illustrative example of the influence of the population size on the coalescent process. The variation of the population size through time is shown with grey dashed lines. (B) Coalescent tree simulated with constant population size ($N(t) = N_{(0)}$). (A) Coalescent tree simulated with a positive population growth rate (population size increases with time, $N(t) > N_{(0)}$). (C) Coalescent tree simulated with a negative population growth rate (population size decreases with time, $N(t) < N_{(0)}$).

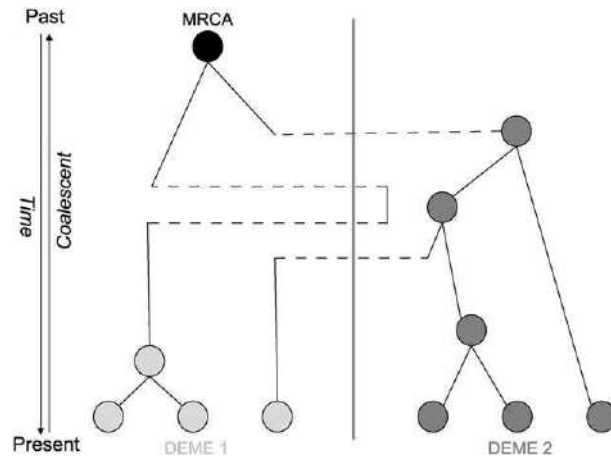


Fig. 3 Illustrative example of the coalescent with population structure and migration. Simulation of a coalescent tree from a sample of six individuals (circles at the present) that belong to two different demes (deme 1 in clear grey and deme 2 in dark grey). Coalescent events generate internal nodes (circles). Migration events are depicted with dashed lines. The process occurs going backwards in time until reaching the MRCA (circle in black).

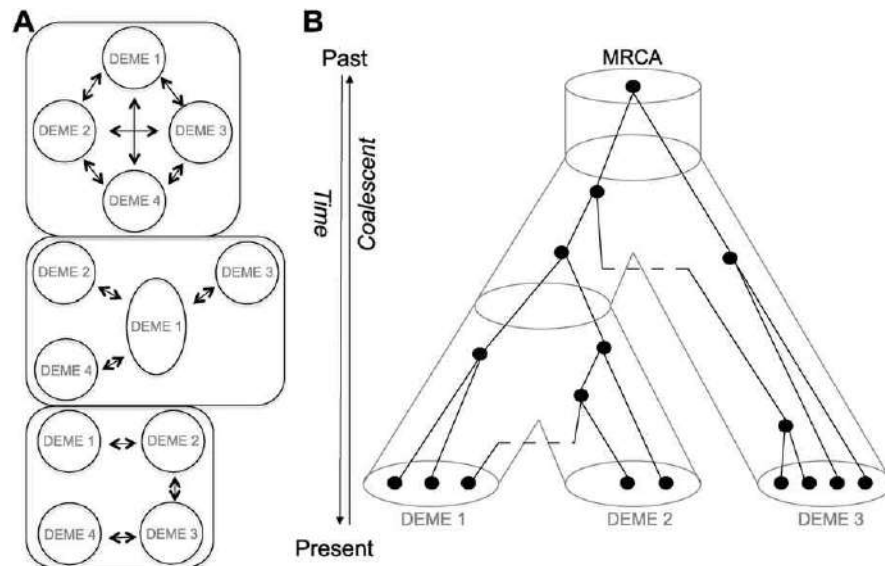


Fig. 4 Migration models and the coalescent within a demes tree. (A) Migration or subdivision models: Island model (top), continent-island model (middle), stepping stone model (bottom). (B) Simulation of a coalescent tree within a demes tree with migration. Migration events are depicted with dashed lines. The process occurs going backwards in time until reaching the MRCA.

where genomic fragments can present different evolutionary histories (**Fig. 5**) and that should be carefully considered for inferring phylogenetic trees (Schierup and Hein, 2000a; Arenas and Posada, 2010b; Mallo *et al.*, 2016; Posada and Crandall, 2002). The number of simulated recombination events is based on the population recombination rate $\rho = 4Nr$, where r is the recombination rate per generation per site and l is the sequence length. Importantly, a recombination breakpoint can only occur in sites that did not reach their MRCA yet. The expected number of recombination events in a panmictic population with constant population size and a sample of k individuals is defined as, $E(\text{number of recombination events}) = \rho \sum_{i=1}^{k-1} \frac{1}{i}$.

It is also possible to simulate the coalescent with heterogeneous recombination along the sequences. There, background (homogeneous) and hotspot (heterogeneous) recombination rates can be applied to simulate hotspot location, heterogeneity and precision (Wiuf and Posada, 2003).

An special coalescent process with recombination is the coalescent simulation of intra-codon recombination (Arenas and Posada, 2010a). Substitution models of codon evolution require complete codons to operate and a codon broken due to a recombination event can be problematic because can lead to branches assigned to incomplete codons. This problem was solved by Arenas and Posada (2010a) with a coalescent algorithm where ancestral material that does not belong to the sample but that is required to complete broken codons (called as pseudo-ancestral material) can be considered in the coalescent, leading to slightly longer ARGs but allowing codon models to operate (Arenas and Posada, 2012).

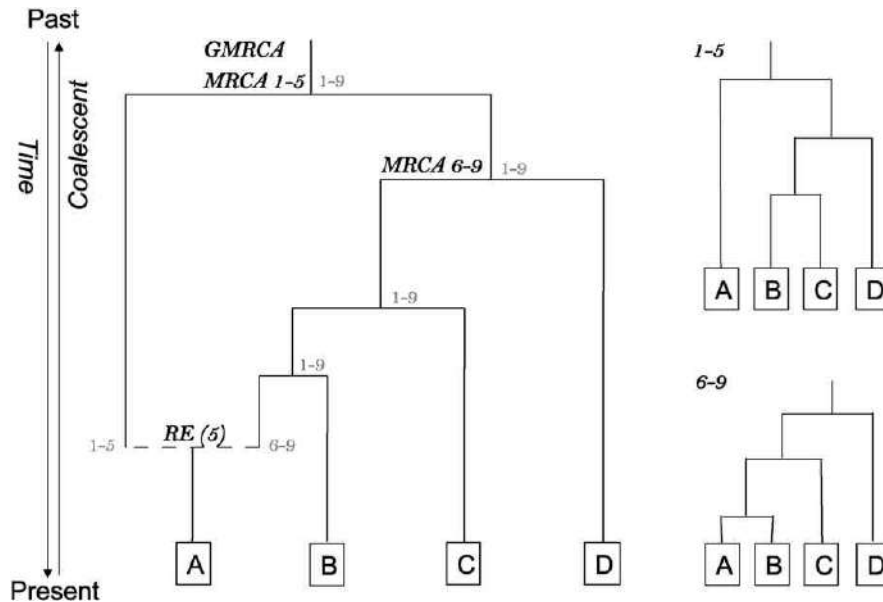


Fig. 5 Illustrative example of an ancestral recombination graph with a tree for each recombinant partition. Ancestral recombination graph for a sample of 4 sequences (A, B, C, D) with 9 sites (i.e., nucleotides). A recombination event (RE, dashed line) occurs with breakpoint at position 5, leading to the recombinant partitions 1–5 and 6–9. Each recombinant partition presents a particular coalescent tree (depicted on the right).

The Coalescent With Selection

Previous coalescent approaches assume neutral evolution. However, in the real world, some lineages may present a higher fitness (i.e., better adapted to the environment) than others. Therefore, the coalescent was also extended to mimic selection through two main approaches.

The first one is the simulation of the ancestral selection graph (ASG) (Krone and Neuhauser, 1997; Neuhauser and Krone, 1997). The ASG presents the genealogy of the lineages (or alleles) into a graph based on coalescence and branching events, similar to the ARG (Fig. 5), where removing branches (i.e., alleles with low fitness) can lead to a tree. The ASG is especially useful to simulate evolutionary histories under weak selection (moderate or strong selection in the ASG may not be computationally tractable). It was later extended to model ancestry conditioned on the frequencies of the selected alleles (Slade and Neuhauser, 2003; Slade, 2000a,b) and even to population subdivision and migration (Slade and Wakeley, 2005).

The other approach is the conditional structured coalescent (Kaplan *et al.*, 1988; Hudson and Kaplan, 1988) where historical allele frequencies are used to generate classes or subpopulations based on their levels of selection. There is not selection within each subpopulation but each subpopulation may present specific values for parameters such as population size or population growth rate. The methods for the coalescent with population structure and migration (see above) can thus be applied to this approach. A migration event can be considered as a mutation or a recombination event (i.e., a beneficial mutation or recombination event could move an allele from low to high selection – or the opposite –, thus moving it from a particular subpopulation to another one). Some examples of this approach involving balancing selection, background selection and selective sweeps are detailed described in Nordborg (2007).

Goals and Limitations of the Coalescent

In addition to the coalescent, some other approaches have been developed to simulate evolutionary histories in population genetics (Arenas, 2012; Hoban *et al.*, 2012). One of the most relevant is the forward in time approach, which simulates the evolutionary history of the entire population from the past to the present (Peng *et al.*, 2012; Carvajal-Rodriguez, 2010). Simulations generated with the forward in time approach consider the ancestral information of the entire population and are useful to study diverse complex evolutionary processes such as mating systems and complex migration (i.e., sex biased dispersal or long-distance dispersal) (e.g., Alves *et al.*, 2016; Rasteiro *et al.*, 2012; Peng and Amos, 2008) or complex selection (Peng *et al.*, 2007) that cannot be studied with the coalescent yet. However, forward in time simulations usually require extensive computational costs and memory capacity to simulate the entire population. In this concern, coalescent simulations are generally much more efficient than forward in time simulations. The coalescent is convenient for studies that require a large number of simulations such as Bayesian (Beaumont and Rannala, 2004) and approximation Bayesian computation (ABC) (Arenas, 2015; Beaumont, 2010) approaches.

A problem in the coalescent can occur with the simulation of the ARG (and, similarly, the ASG) under very high population recombination rates (i.e., $\rho > 1000$). Under this situation the number of active lineages may dramatically increase going backwards

in time leading to computer problems related with out of memory. In addition, if the population growth rate is too negative, the TMRCA can be almost infinite, again leading to computational problems.

Coalescent Simulators

A number of freely available coalescent simulators have been developed (Table 1). Most of available coalescent simulators implement demographics, population structure and recombination. However, selection is implemented in only a few coalescent programs mainly due to its complexity (see above) in comparison with forward in time approaches (Peng *et al.*, 2012; Carvajal-Rodriguez, 2010). Current trends of coalescent frameworks include the adaptation of the coalescent to simulate specific genetic markers such as codons (Arenas and Posada, 2010a, 2007), proteins (Arenas *et al.*, 2013) or whole-genome sequences (e.g., Arenas and Posada, 2014b; Kelleher *et al.*, 2016). Another trend is to develop extensions of the coalescent to mimic the evolution of specific organisms such as bacteria (De Maio and Wilson, 2017; Brown *et al.*, 2016) and cancer (Zhao *et al.*, 2014; Nicolas *et al.*, 2007). In all, the development of coalescent simulators is a very active area of research in population genetics with a continuous emergence of more efficient, and with more capabilities, frameworks.

Applications of the Coalescent

The coalescent provides clear and rapid representations of real evolutionary processes and its implementation in computer frameworks is, in general, straightforward. Consequently, the coalescent has been applied to a variety of purposes, the most relevant are presented below.

Hypothesis Testing

The coalescent has been widely used for testing hypothesis such as the effects between evolutionary parameters or the influence of evolutionary parameters on phylogenetic reconstructions. It follows with some illustrative examples.

Computer simulations of the coalescent with recombination (Arenas, 2013a), followed by the simulation of codon evolution upon the simulated coalescent histories (Arenas and Posada, 2012), were applied in several studies to analyze the influence of ignored recombination on the estimation of selection (the nonsynonymous (dN) to synonymous (dS) substitution ratio, dN/dS or ω) (Arenas and Posada, 2010a, 2014a; Anisimova *et al.*, 2003; Shriner *et al.*, 2003). These studies found that ignored recombination biases the estimation of dN/dS by generating false positively selected sites. As a consequence, those studies concluded that recombination must be considered for the estimation of dN/dS . That can be performed by the following methodology: (i) detection of recombination breakpoints, (ii) inference of a phylogenetic tree for each recombination partition and, (iii) estimation of dN/dS per site (or per recombination partition) accounting for the corresponding phylogenetic tree. This methodology was already applied in several studies (e.g., Perez-Losada *et al.*, 2009; Perez-Losada *et al.*, 2011).

Another application of coalescent simulations was the analysis of the influence of recombination on ancestral sequence reconstruction (ASR). Arenas and Posada (2010b) performed coalescent simulations under different levels of recombination and simulated DNA sequence evolution over those coalescent histories to generate sequences for every tip and internal node. Next, they performed ASR from the simulated sample and a previously inferred ML or Bayesian phylogenetic tree, to infer the ancestral sequences. Finally, they compared the simulated (true) ancestral sequences with the inferred ancestral sequences to evaluate the impact of recombination on ASR. They found that recombination seriously bias ASR, especially under high levels of genetic diversity (Arenas and Posada, 2010b). In order to reduce this bias, they suggested that ASR should be performed for each recombinant partition independently (see below).

The influence of recombination on the estimation of dN/dS and on ASR is derived from the influence of recombination on phylogenetic tree reconstruction. Recombination can bias phylogenetic tree inferences because each recombinant partition can present an evolutionary history that differs from the evolutionary history of other recombinant partitions. This was investigated by Schierup and Hein (2000a) with coalescent simulations based on the Hudson's algorithm of the coalescent with recombination (Hudson, 1983). They found that if recombination is present but ignored, derived phylogenetic trees can be biased with incorrect topologies and branch lengths (Schierup and Hein, 2000a). Moreover, ignoring recombination can result in loss of molecular clock, apparent homoplasies and incorrect substitution rate heterogeneity (Schierup and Hein, 2000a,b). In addition, phylogenetic incongruence in empirical data was observed and attributed to ignored recombination (Worobey and Holmes, 1999; Feil *et al.*, 2001). Consequently, under the presence of recombination one should either infer a phylogenetic tree for each recombinant partition (recombinant partitions can be identified with a framework for the detection of recombination breakpoints (e.g., Martin *et al.*, 2015; Kosakovsky Pond *et al.*, 2006) (Arenas and Posada, 2010b) or infer a phylogenetic recombination network (Arenas, 2013b; Griffiths and Marjoram, 1997; Huson and Bryant, 2006; Huson, 1998). Additionally, a variety of analytical evolutionary frameworks (i.e., BEAST (Drummond *et al.*, 2012)) are based on the coalescent without recombination and can therefore generate biases from data that suffered recombination events. Detecting recombination partitions and analyzing them separately is once again recommended (e.g., Perez-Losada *et al.*, 2007).

Table 1 Main coalescent simulators and their capabilities

Program	Evolutionary scenarios	Additional capabilities	Genetic data	OS	Reference
FastSimBac	Dm, Pm, Re	Bacterial recombination Recombination hotspots Species/population trees	Nt	SC	De Maio and Wilson (2017)
SimBac	Re	Bacterial recombination	Nt	SC	Brown <i>et al.</i> (2016)
msprisme	Dm, Pm, Re	Capabilities of <i>ms</i> and the following: Fast and accurate simulation of huge samples	–	SC	Kelleher <i>et al.</i> (2016)
scrm	Ls, Dm, Pm, Re	Species/population trees	–	SC	Staab <i>et al.</i> (2015)
CoalEvol	Ls, Dm, Pm, Re	Capabilities of <i>NetRecodon</i> and the following: Recombination hotspots	Nt, Cd, Aa	SC, Linux, Mac	Arenas and Posada (2014b)
SGWE	Ls, Dm, Pm, Re	Species/population trees Capabilities of <i>CoalEvol</i> and the following: Different models of evolution for different partitions	Nt, Cd, Aa	SC, Linux, Mac	Arenas and Posada (2014b)
Cosi2	Dm, Pm, Re, Se	Recombination hotspots Gene conversion	SNP	SC	Shlyakhter <i>et al.</i> (2014)
ProteinEvolver	Ls, Dm, Pm, Re	Capabilities of <i>NetRecodon</i> and the following: Recombination hotspots	Nt, Cd, Aa	SC, Linux, Mac	Arenas <i>et al.</i> (2013)
GUMS	Dm, Pm	Species/population trees	–	Linux, Mac	Heled <i>et al.</i> (2013)
FTEC	Dm, Pm, Re	–	SNP	SC	Reppell <i>et al.</i> (2012)
Fastsimcoal	Dm, Pm, Re	Capabilities of <i>SIMCOAL2</i> and the following: Demographic periods	Nt, STR, SNP	Linux, Mac, Win	Excoffier and Foll (2011)
NetRecodon	Ls, Dm, Pm, Re	Species/population trees Admixture Capabilities of <i>Recodon</i> and the following: Intracodon recombination Printed ARG ^a	Nt, Cd	All	Arenas and Posada (2010a)
msms	Dm, Pm, Re, Se	Capabilities of <i>ms</i> and the following: Selection (single diploid locus)	SNP	All	Ewing and Hermisson (2010)
mbis	Dm, Pm, Re, Se	Capabilities of <i>ms</i> and the following: Selection (biallelic site) Recombination hot-spots	SNP, Nt	SC	Teshima and Innan (2009)
MaCS	Dm, Pm, Re	Species/population trees	–	SC	Chen <i>et al.</i> (2009)
SimMLST	Dm, Re	Multi-locus sequence typing data	Nt	SC, Linux, Win	Didelot <i>et al.</i> (2009)
msHOT	Dm, Pm, Re	Capabilities of <i>ms</i> and the following: Multiple crossover hotspots and multiple gene conversion hotspots	SNP	SC	Helenthal and Stephens (2007)

(Continued)

Table 1 Continued

Program	Evolutionary scenarios	Additional capabilities	Genetic data	OS	Reference
GENOME mcoalsim	Dm, Pm, Re Dm, Pm, Re	Recombination hotspots Species/population trees Capabilities of <i>ms</i> and the following: Recombination includes hot-spots	SNP Nt	SC, Linux, Win SC, Mac, Win	Liang <i>et al.</i> (2007) Ramos-Onsins and Mitchell-Olds (2007)
Recodon	Dm, Pm, Re	Demographics according to a population growth rate and demographic periods	Nt, Cd	All	Arenas and Posada (2007)
Serial Simcoal	Ls, Dm, Pm	Capabilities of <i>SIMCOAL</i> and the following: Longitudinal sampling Species/population trees	Nt, RFLP, STR	SC, Mac, Win	Anderson <i>et al.</i> (2005)
CoaSim	Dm, Pm, Re	Gene conversion Species/population trees	SNP, STR	SC, Linux	Mailund <i>et al.</i> (2005)
Cosi	Dm, Pm, Re	Recombination hotspots	SNP	SC	Schaffner <i>et al.</i> (2005)
Simcoal2	Dm, Pm, Re	Gene conversion Species/population trees Capabilities of <i>SIMCOAL</i> and the following:	Nt, RFLP, STR, SNP	SC, Win	Laval and Excoffier (2004)
SeqSim	Re, Se	Recombination hotspots	Nt, STR, SNP	SC, Linux, Win	Spencer and Coop (2004)
SNPsim	Dm, Re	–	SNP	SC	Posada and Wiuf (2003)
CodonRecSim	Re	Recombination hotspots	Cd	SC, Win	Anisimova <i>et al.</i> (2003)
ms	Dm, Pm, Re	–	SNP	SC	Hudson (2002)
Simcoal	Dm, Pm	–	Nt, RFLP, STR	SC, Win	Excoffier <i>et al.</i> (2000)
Treewolve	Dm, Pm, Re	–	Nt	SC, Mac	Grassly <i>et al.</i> (1999)

^aThe ARG can be analyzed/visualized with *NetTest* (Arenas *et al.*, 2010).

Note. The simulators are presented by publication date, from the present to the past. The table shows: (i) implemented evolutionary scenarios (Ls, Dm, Pm, Re and Se, refers to longitudinal sampling, demographics, population structure and migration, recombination and selection, respectively); (ii) additional capabilities to the cited evolutionary scenarios; (iii) type of simulated genetic data (in addition to the simulated coalescent tree(s) like sequences of SNPs, microsatellites (STR), nucleotides (NT), codons (Cd) and amino acids (Aa); (iv) operative system (OS) considering the availability of binary files (Linux, Macintosh, Windows) and/or source code (SC); (v) references. This list is not exhaustive but is meant to highlight the diversity of the field.

Another example is the evaluation of the criteria for selecting substitution models of evolution. Luo *et al.* (2010) simulated coalescent evolutionary histories and used those trees to simulate DNA sequence evolution under different substitution models of evolution. Next, they estimated the best fitting substitution model of evolution for the simulated data under different statistical criteria such as Bayesian information criterion, decision theory, Akaike information criterion and hierarchical likelihood-ratio test. They found that Bayesian information and decision theory performed better for identifying the correct substitution model and, therefore, these criteria should be preferred in substitution model selection.

Validation of Analytical Methods

Any analytical framework must be properly validated. In population genetics, the “true” evolutionary process of a dataset is usually unknown. Therefore, the evaluation of analytical frameworks is frequently assessed with computer simulations where the “true” (simulated) evolutionary process is known. The coalescent has been widely used for the evaluation of analytical evolutionary frameworks. Some illustrative examples are described below.

It was remarkable in the previous subsection that genetic recombination should be considered in the evolutionary analysis of genetic data. Because of that, and other aspects about the important role of recombination in the evolution of many organisms (e.g., Arenas *et al.*, 2017; Castelhana *et al.*, 2017; Arenas *et al.*, 2016; Perez-Losada *et al.*, 2015), a variety of frameworks were developed to detect and quantify recombination and to infer phylogenetic recombination frameworks. Many of these frameworks were validated with the coalescent extended with recombination. The following are some examples: (i) Marttinen *et al.* (2012) with a method to detect homologous recombination events (validated comparing the number of simulated and predicted recombination events); (ii) Westesson and Holmes (2009) with a method to identify recombination breakpoints (validated comparing simulated and detected recombinant regions); (iii) Dialdestoro *et al.* (2016) with a method to estimate recombination rate and other population genetics parameters such as population size (validated comparing the distance between simulated and estimated recombination rates); (iv) White *et al.* (2013) and Sun *et al.* (2011) evaluating some of the existing recombination detection tests (validated comparing simulated and estimated recombination rate); (v) Lopes *et al.* (2014) with a method to coestimate recombination rate with other parameters of molecular evolution (also validated comparing simulated and estimated recombination rate).

Coalescent simulations are also frequently applied to evaluate reconstruction methods of phylogenetic trees (e.g., McCormack *et al.*, 2009; Chen *et al.*, 2011) and networks (e.g., Wen *et al.*, 2016; Javed *et al.*, 2011; Yu *et al.*, 2014; Yu and Nakhleh, 2015; Hassanzadeh *et al.*, 2012; Wang *et al.*, 2013) where the topology and branch lengths of simulated and reconstructed phylogenies are compared.

Coalescent simulations have also been applied to validate other analytical methods in population genetics such as the estimation of pairwise distance between sequences (Domazet-Loso and Haubold, 2009) or the estimation of population size and migration rate (Beerli, 2006).

Estimation of Evolutionary Parameters and Model Selection

A variety of computer programs based on the coalescent can estimate evolutionary parameters. Some of the most relevant are, (i) BEAST, estimation of variable population size, substitution rate and other evolutionary parameters, also phylogenetic tree reconstruction; (ii) GENETREE (Bahlo and Griffiths, 2000), estimation of population growth, mutation and migration rates; (iii) IMA (Hey and Nielsen, 2007), estimation of population size, migration rate and divergence time; (iv) LAMARC (Kuhner, 2006), estimation of population growth, migration and recombination rates; (v) MIGRATE (Beerli and Felsenstein, 2001), estimation of population size and migration rates.

The cited programs estimate evolutionary parameters through a likelihood function. However, under complex models of evolution a likelihood function could not be mathematically designed or could be computationally intractable. In these situations, the ABC approach provides a very useful alternative because it does not require the likelihood function. Moreover, ABC can outperform maximum-likelihood (ML) methods based on approximate models (Arenas *et al.*, 2015; Lopes *et al.*, 2014). Briefly, ABC provides model selection and parameters estimation by the following pipeline: (i) Design of prior distributions for parameters, these prior distributions should contain the real value. (ii) Extensive computer simulations after sampling from the prior distributions. (iii) Design and computation of summary statistics for simulated and real data. (iv) Selection of summary statistics from simulated data that are closer to the summary statistics from real data according to a given tolerance. (v) With the retained summary statistics from simulated data and summary statistics from real data, ABC applies a rejection (Pritchard *et al.*, 1999) or a multiple regression approach (Beaumont *et al.*, 2002) to obtain the posterior distributions for the studied models or parameters. For further details about the ABC approach the reader is referred to the following reviews (Beaumont, 2010; Arenas, 2015; Csillery *et al.*, 2010; Sunnaker *et al.*, 2013; Bertorelle *et al.*, 2010; Lopes and Beaumont, 2010). ABC usually requires a number of computer simulations that should be longer enough (from thousands to millions) for properly covering the prior distributions. Since the coalescent provides a rapid computation, most of ABC studies have applied the coalescent to simulate datasets (e.g., Arenas *et al.*, 2015; Lopes *et al.*, 2014; Alves *et al.*, 2016; Wilson *et al.*, 2009; Fan and Kubatko, 2011; Li and Jakobsson, 2012). Because of this important application of the coalescent, many coalescent simulators incorporate adaptations to facilitate the design of

ABC analyses (i.e., simulating from predefined prior distributions) (e.g., Arenas and Posada, 2014b; Excoffier and Foll, 2011; Pavlidis *et al.*, 2010).

Designing a Coalescent-Based Experiment

Studies involving coalescent simulations may require a previous design. The researcher has to choose a coalescent tool to simulate an evolutionary scenario and obtain a desired type of genetic data. A few aspects should be considered to design the study.

1. Evolutionary scenario. The currently available coalescent simulators implement a large variety of models to mimic diverse evolutionary scenarios (see Section “Coalescent Simulators” and Table 1). The following sentences provide suggestions from the experience of the author. The simulation of basic scenarios such as a constant population size can be easily performed with the *ms* program. This program is also useful for simulating basic (homogeneous) recombination. For scenarios with population structure and migration, the simulators *SimCoal2* and *fastsimcoal* provide a variety of parameters (i.e., a matrix of migration rates). Complex recombination (i.e., hotspots) can be simulated with *SNPsim*, which implements a wide variety of models of heterogeneous recombination. Scenarios of selection can be simulated with *mbs* and *msms*.
2. Desired output data. Any coalescent framework simulates a coalescent tree but may not simulate genetic data. Here there are two options. (i) Apply a coalescent framework that internally simulates genetic data (i.e., *CoalEvol* simulates nucleotide, codon and protein sequences under a variety of substitution models of evolution); (ii) Apply the chosen coalescent framework to simulate the coalescent history and then simulate sequence evolution (Yang, 2006) upon such a history with frameworks such as *INDELible* (Fletcher and Yang, 2009), *CoalEvol*, *SeqGen* (Rambaut and Grassly, 1997), *Pyvolve* (Spielman and Wilke, 2015) or *Phylosim* (Sipos *et al.*, 2011), see for a review (Arenas, 2012).

Illustrative Practical Examples

Here five practical examples to simulate a particular type of genetic data under a particular evolutionary scenario are presented.

1. *Codon sequences under recombination*. Probably the easiest procedure to perform this simulation is with the programs *NetRecodon* or *CoalEvol*. These programs implement the simulation of the ARG and the subsequent simulation of coding sequence evolution (under a variety of codon substitution models) over the ARG. Another possibility is to apply the program *ms* to obtain the ARG (a coalescent tree for each recombinant partition) and then apply *INDELible* (or a similar framework, see above) to simulate codon evolution upon the coalescent trees. The advantage from using the first procedure is that *NetRecodon* and *CoalEvol* implement the simulation of both intercodon and intracodon recombination, which is much more realistic than consider only intercodon recombination in the second procedure.
2. *Protein sequences under a population structure with migration*. Again, the program *CoalEvol* implements the simulation of population structure with migration and the simulation of protein sequences over the previously simulated tree. However, *CoalEvol* implements a limited number of migration parameters (i.e., same migration rate between all demes). As an alternative, one could apply *fastsimcoal*, which includes a variety of parameters to better mimic the population structure with migration, and then simulate protein sequences with *INDELible* (or similar framework).
3. *Protein sequences under selection*. First, coalescent trees under selection can be simulated with *mbs* or *msms* and then, protein sequences can be simulated over those trees with *INDELible* (or similar framework). For simulations under complex models of selection one could explore forward in time simulation frameworks (Peng *et al.*, 2012).
4. *Nucleotide sequences under heterogeneous recombination (hotspots)*. I would recommend simulate the coalescent trees (ARG) with *SNPsim* (although there are other advanced simulators such as *Cosi2*, *msHOT*, *GENOME* and *SimCoal2*) and next simulate nucleotide sequences with *INDELible* (or similar framework). Another methodology is to perform both the coalescent simulation and sequence simulation with *CoalEvol*, which implements the heterogeneous recombination models of *SNPsim*.
5. *Codon sequences with longitudinal sampling accounting for variation of the population size and population structure*. Longitudinal sampling, variable population size through time and population structure can be simulated with *Serial SimCoal*, *scrm* and *CoalEvol*. *CoalEvol* also allows the simulation of codon sequence evolution. In case of *Serial SimCoal* and *scrm*, codon sequence evolution can be simulated with *INDELible* (or similar framework).

Concluding Remarks

The coalescent is a stochastic statistical process established in population genetics for more than 30 years. Instead of reducing its popularity through time, the coalescent is progressively more famous, used and incorporated into new frameworks (usually published in important journals of the discipline because of their utility, as presented in this study) to both simulate and analyze genetic data in an evolutionary perspective.

A number of coalescent simulators exist implementing fundamental evolutionary processes of the real world such as demography, recombination, population structure and selection. Some coalescent simulators are also oriented to mimic the evolution of specific organisms such as bacteria or cancer. However, further work is required to accommodate more complex evolutionary models, especially complex selection, as done in the forward in time simulation approach.

Some famous analytical tools (i.e., *BEAST* and *LAMARC*) are based on the coalescent and consequently, users of those tools should have in mind the limitations of the underlined coalescent approach (i.e., the cited programs ignore selection). It is worth highlighting the incorporation of the coalescent into the ABC approach, which is a methodology of increasing application in population genetics and ecology to analyze genetic data under complex models of evolution. Instead of being forced to only consider the limited models implemented in ML or Bayesian tools, ABC allows the user to apply any model that can be simulated. In this concern, the coalescent provides fast simulations that are convenient to achieve the extensive number of simulations usually required in ABC.

The future of the coalescent is likely to be highly promising because of their important applications. However, we still need to develop more extensions of the coalescent, and subsequent implementation in simulation frameworks, to better fit with real evolutionary processes.

Acknowledgements

MA was supported by the Grant “Ramón y Cajal” RYC-2015-18241 from the Spanish Government.

See also: Evolutionary Models. Gene Duplication and Speciation. Genetics and Population Analysis. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Molecular Clock. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Phylogenetic Tree Rooting. Population Genetics

References

- Alves, I., Arenas, M., Currat, M., *et al.*, 2016. Long-distance dispersal shaped patterns of human genetic diversity in Eurasia. *Mol Biol Evol* 33, 946–958.
- Anderson, C.N., Ramakrishnan, U., Chan, Y.L., Hadly, E.A., 2005. Serial SimCoal: A population genetics model for data from multiple populations and points in time. *Bioinformatics* 21, 1733–1734.
- Anisimova, M., Nielsen, R., Yang, Z., 2003. Effect of recombination on the accuracy of the likelihood method for detecting positive selection at amino acid sites. *Genetics* 164, 1229–1236.
- Arenas, M., 2012. Simulation of molecular data under diverse evolutionary scenarios. *PLOS Comput Biol* 8, e1002495.
- Arenas, M., 2013a. Computer programs and methodologies for the simulation of DNA sequence data with recombination. *Front Genet* 4, 9.
- Arenas, M., 2013b. The importance and application of the ancestral recombination graph. *Front Genet* 4, 206.
- Arenas, M., 2015. Advances in computer simulation of genome evolution: Toward more realistic evolutionary genomics analysis by approximate Bayesian computation. *J Mol Evol* 80, 189–192.
- Arenas, M., Araujo, N.M., Branco, C., *et al.*, 2017. Mutation and recombination in pathogen evolution: Relevance, methods and controversies. *Infect Genet Evol*.
- Arenas, M., Dos Santos, H.G., Posada, D., Bastolla, U., 2013. Protein evolution along phylogenetic histories under structurally constrained substitution models. *Bioinformatics* 29, 3020–3028.
- Arenas, M., Lopes, J.S., Beaumont, M.A., Posada, D., 2015. CodABC: A computational framework to coestimate recombination, substitution, and molecular adaptation rates by approximate Bayesian computation. *Mol Biol Evol* 32, 1109–1112.
- Arenas, M., Lorenzo-Redondo, R., Lopez-Galindez, C., 2016. Influence of mutation and recombination on HIV-1 *in vitro* fitness recovery. *Mol Phylogenet Evol* 94, 264–270.
- Arenas, M., Patricio, M., Posada, D., Valiente, G., 2010. Characterization of phylogenetic networks with NetTest. *BMC Bioinformatics* 11, 268.
- Arenas, M., Posada, D., 2007. Recodon: Coalescent simulation of coding DNA sequences with recombination, migration and demography. *BMC Bioinformatics* 8, 458.
- Arenas, M., Posada, D., 2010a. Coalescent simulation of intracodon recombination. *Genetics* 184, 429–437.
- Arenas, M., Posada, D., 2010b. The effect of recombination on the reconstruction of ancestral sequences. *Genetics* 184, 1133–1139.
- Arenas, M., Posada, D., 2012. Simulation of coding sequence evolution. In: Cannarozzi, G.M., Schneider, A. (Eds.), *Codon Evolution*. Oxford: Oxford University Press.
- Arenas, M., Posada, D., 2014a. The influence of recombination on the estimation of selection from coding sequence alignments. In: Fares, M.A. (Ed.), *Natural Selection: Methods and Applications*. Boca Raton: CRC Press/Taylor & Francis.
- Arenas, M., Posada, D., 2014b. Simulation of genome-wide evolution under heterogeneous substitution models and complex multispecies coalescent histories. *Mol Biol Evol* 31, 1295–1301.
- Bahlo, M., Griffiths, R.C., 2000. Inference from gene trees in a subdivided population. *Theor Popul Biol* 57, 79–95.
- Beaumont, M.A., 2010. Approximate Bayesian computation in evolution and ecology. *Annu Rev Ecol Syst* 41, 379–405.
- Beaumont, M.A., Rannala, B., 2004. The Bayesian revolution in genetics. *Nat Rev Genet* 5, 251–261.
- Beaumont, M.A., Zhang, W., Balding, D.J., 2002. Approximate Bayesian computation in population genetics. *Genetics* 162, 2025–2035.
- Beerli, P., 2006. Comparison of Bayesian and maximum-likelihood inference of population genetic parameters. *Bioinformatics* 22, 341–345.
- Beerli, P., Felsenstein, J., 2001. Maximum likelihood estimation of a migration matrix and effective population sizes in *n* subpopulations by using a coalescent approach. *Proc Natl Acad Sci USA* 98, 4563–4568.
- Bertorelle, G., Benazzo, A., Mona, S., 2010. ABC as a flexible framework to estimate demography over space and time: Some cons, many pros. *Mol Ecol* 19, 2609–2625.
- Brown, T., Didelot, X., Wilson, D.J., De Maio, N., 2016. SimBac: Simulation of whole bacterial genomes with homologous recombination. *Microb Genom* 2.
- Carvajal-Rodriguez, A., 2010. Simulation of genes and genomes forward in time. *Curr Genomics* 11, 58–61.
- Castelhano, N., Araujo, N.M., Arenas, M., 2017. Heterogeneous recombination among Hepatitis B virus genotypes. *Infect Genet Evol* 54, 486–490.
- Chen, G.K., Marjoram, P., Wall, J.D., 2009. Fast and flexible simulation of DNA sequence data. *Genome Res* 19, 136–142.
- Chen, S.C., Rosenberg, M.S., Lindsay, B.G., 2011. MixtureTree: A program for constructing phylogeny. *BMC Bioinformatics* 12, 111.
- Csillery, K., Blum, M.G.B., Gaggiotti, O.E., Francois, O., 2010. Approximate Bayesian computation (ABC) in practice. *Trends Ecol Evol* 25, 410–418.

- De Maio, N., Wilson, D.J., 2017. The bacterial sequential markov coalescent. *Genetics* 206, 333–343.
- Dialdestoro, K., Sibbesen, J.A., Marett, L., *et al.*, 2016. Coalescent inference using serially sampled, high-throughput sequencing data from intrahost HIV infection. *Genetics* 202, 1449–1472.
- Didelot, X., Lawson, D., Falush, D., 2009. SimMLST: Simulation of multi-locus sequence typing data under a neutral model. *Bioinformatics* 25, 1442–1444.
- Domazet-Loso, M., Haubold, B., 2009. Efficient estimation of pairwise distances between genomes. *Bioinformatics* 25, 3221–3227.
- Drummond, A.J., Suchard, M.A., Xie, D., Rambaut, A., 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Mol Biol Evol* 29, 1969–1973.
- Ewing, G., Hermisson, J., 2010. MSMS: A coalescent simulation program including recombination, demographic structure and selection at a single locus. *Bioinformatics* 26, 2064–2065.
- Excoffier, L., Foll, M., 2011. fastsimcoal: A continuous-time coalescent simulator of genomic diversity under arbitrarily complex evolutionary scenarios. *Bioinformatics* 27, 1332–1334.
- Excoffier, L., Novembre, J., Schneider, S., 2000. SIMCOAL: A general coalescent program for the simulation of molecular data in interconnected populations with arbitrary demography. *J Hered* 91, 506–509.
- Fan, H.H., Kubatko, L.S., 2011. Estimating species trees using approximate Bayesian computation. *Mol Phylogenet Evol* 59, 354–363.
- Feil, E.J., Holmes, E.C., Bessen, D.E., *et al.*, 2001. Recombination within natural populations of pathogenic bacteria: Short-term empirical estimates and long-term phylogenetic consequences. *Proc Natl Acad Sci USA* 98, 182–187.
- Fletcher, W., Yang, Z., 2009. INDELible: A flexible simulator of biological sequence evolution. *Mol Biol Evol* 26, 1879–1888.
- Grassly, N.C., Harvey, P.H., Holmes, E.C., 1999. Population dynamics of HIV-1 inferred from gene sequences. *Genetics* 151, 427–438.
- Griffiths, R.C., Marjoram, P., 1996. Ancestral inference from samples of DNA sequences with recombination. *J Comput Biol* 3, 479–502.
- Griffiths, R.C., Marjoram, P., 1997. An ancestral recombination graph. In: Donnelly, P., Tavaré, S. (Eds.), *Progress in population genetics and human evolution*. Berlin: Springer-Verlag.
- Hassanzadeh, R., Eslahchi, C., Sung, W.K., 2012. Constructing phylogenetic supernetworks based on simulated annealing. *Mol Phylogenet Evol* 63, 738–744.
- Hein, J., Schierup, M., Wiuf, C., 2005. *Gene genealogies, variation and evolution: A primer in coalescent theory*. Oxford University Press.
- Heled, J., Bryant, D., Drummond, A.J., 2013. Simulating gene trees under the multispecies coalescent and time-dependent migration. *BMC Evol Biol* 13, 44.
- Hellenthal, G., Stephens, M., 2007. msHOT: Modifying Hudson's ms simulator to incorporate crossover and gene conversion hotspots. *Bioinformatics* 23, 520–521.
- Hey, J., Nielsen, R., 2007. Integration within the Felsenstein equation for improved Markov chain Monte Carlo methods in population genetics. *Proc Natl Acad Sci USA* 104, 2785–2790.
- Hoban, S., Bertorelle, G., Gaggiotti, O.E., 2012. Computer simulations: Tools for population and evolutionary genetics. *Nat Rev Genet* 13, 110–122.
- Huson, D.H., 1998. SplitsTree: Analyzing and visualizing evolutionary data. *Bioinformatics* 14, 68–73.
- Huson, D.H., Bryant, D., 2006. Application of phylogenetic networks in evolutionary studies. *Mol Biol Evol* 23, 254–267.
- Hudson, R.R., 1983. Properties of a neutral allele model with intragenic recombination. *Theor Popul Biol* 23, 183–201.
- Hudson, R.R., 1990. Gene genealogies and the coalescent process. *Oxford Surv Evol Biol* 7, 1–44.
- Hudson, R.R., 1998. Island models and the coalescent process. *Mol Ecol* 7, 413–418.
- Hudson, R.R., 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18, 337–338.
- Hudson, R.R., Kaplan, N.L., 1988. The coalescent process in models with selection and recombination. *Genetics* 120, 831–840.
- Hudson, R.R., Kaplan, N.L., 1995. The coalescent process and background selection. *Philos Trans R Soc Lond B Biol Sci* 349, 19–23.
- Javed, A., Pybus, M., Mele, M., *et al.*, 2011. IRIIS: Construction of ARG networks at genomic scales. *Bioinformatics* 27, 2448–2450.
- Kaplan, N.L., Darden, T., Hudson, R.R., 1988. The coalescent process in models with selection. *Genetics* 120, 819–829.
- Kaplan, N.L., Hudson, R.R., Izuka, M., 1991. The coalescent process in models with selection, recombination and geographic subdivision. *Genet Res Camb* 57, 83–91.
- Kelleher, J., Etheridge, A.M., McVean, G., 2016. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLOS Comput Biol* 12, e1004842.
- Kimura, M., Weiss, G.H., 1964. The stepping stone model of population structure and the decrease of genetic correlation with distance. *Genetics* 49, 561–576.
- Kingman, J.F.C., 1982. The coalescent. *Stoch Process Appl* 13, 235–248.
- Kosakovsky Pond, S.L., Posada, D., Gravenor, M.B., Woelk, C.H., Frost, S.D., 2006. GARD: A genetic algorithm for recombination detection. *Bioinformatics* 22, 3096–3098.
- Krone, S.M., Neuhauser, C., 1997. Ancestral processes with selection. *Theor Popul Biol* 51, 210–237.
- Kuhner, M.K., 2006. LAMARC 2.0: Maximum likelihood and Bayesian estimation of population parameters. *Bioinformatics* 22, 768–770.
- Laval, G., Excoffier, L., 2004. SIMCOAL 2.0: A program to simulate genomic diversity over large recombining regions in a subdivided population with a complex history. *Bioinformatics* 20, 2485–2487.
- Lemey, P., Rambaut, A., Pybus, O.G., 2006. HIV evolutionary dynamics within and among hosts. *AIDS Rev* 8, 125–140.
- Li, S., Jakobsson, M., 2012. Estimating demographic parameters from large-scale population genomic data using approximate Bayesian computation. *BMC Genet* 13, 22.
- Liang, L., Zollner, S., Abecasis, G.R., 2007. GENOME: A rapid coalescent-based whole genome simulator. *Bioinformatics* 23, 1565–1567.
- Lopes, J.S., Arenas, M., Posada, D., Beaumont, M.A., 2014. Coestimation of recombination, substitution and molecular adaptation rates by approximate Bayesian computation. *Heredity* 112, 255–264.
- Lopes, J.S., Beaumont, M.A., 2010. ABC: A useful Bayesian tool for the analysis of population data. *Infect Genet Evol* 10, 826–833.
- Luo, A., Qiao, H., Zhang, Y., *et al.*, 2010. Performance of criteria for selecting evolutionary models in phylogenetics: A comprehensive study based on simulated datasets. *BMC Evol Biol* 10, 242.
- Mailund, T., Schierup, M.H., Pedersen, C.N., *et al.*, 2005. CoaSim: A flexible environment for simulating genetic data under coalescent models. *BMC Bioinformatics* 6, 252.
- Mallo, D., Sánchez-Cobos, A., Arenas, M., 2016. Diverse considerations for successful phylogenetic tree reconstruction: Impacts from model misspecification, recombination, homoplasy, and pattern recognition. In: Elloumi, M., Iliopoulos, C., Wang, J., Zomaya, A. (Eds.), *Pattern Recognition in Computational Molecular Biology*. John Wiley & Sons, Inc.
- Martin, D.P., Murrell, B., Golden, M., Khoosal, A., Muhire, B., 2015. RDP4: Detection and analysis of recombination patterns in virus genomes. *Virus Evol* 1, vev003.
- Martinen, P., Hanage, W.P., Croucher, N.J., *et al.*, 2012. Detection of recombination events in bacterial genomes from large population samples. *Nucleic Acids Res* 40, e6.
- Mccormack, J.E., Huang, H., Knowles, L.L., 2009. Maximum likelihood estimates of species trees: How accuracy of phylogenetic inference depends upon the divergence history and sampling design. *Syst Biol* 58, 501–508.
- Neuhauser, C., Krone, S.M., 1997. The genealogy of samples in models with selection. *Genetics* 145, 519–534.
- Neuhauser, C., Tavaré, S., 2001. *The coalescent*. Encyclopedia of Genetics. New York: Academic Press.
- Nicolas, P., Kim, K.M., Shibata, D., Tavaré, S., 2007. The stem cell population of the human colon crypt: Analysis via methylation patterns. *PLOS Comput Biol* 3, e28.
- Nielsen, R., Wakeley, J., 2001. Distinguishing migration from isolation: A Markov chain Monte Carlo approach. *Genetics* 158, 885–896.
- Nordborg, M., 2007. Coalescent theory. In: Balding, D.J., Bishop, M., Cannings, C. (Eds.), *Handbook of Statistical Genetics*, Third ed. Chichester, UK: John Wiley & Sons, Ltd.
- Notohara, M., 1990. The coalescent and the genealogical process in geographically structured population. *J Math Biol* 29, 59–75.
- Pavlidis, P., Laurent, S., Stephan, W., 2010. msABC: A modification of Hudson's ms to facilitate multi-locus ABC analysis. *Mol Ecol Resour* 10, 723–727.
- Peng, B., Amos, C.I., 2008. Forward-time simulations of non-random mating populations using simuPOP. *Bioinformatics* 24, 1408–1409.
- Peng, B., Amos, C.I., Kimmel, M., 2007. Forward-time simulations of human populations with complex diseases. *PLOS Genet* 3, e47.
- Peng, B., Kimmel, M., Amos, C.I., 2012. *Forward-time population genetics simulations: Methods, implementation, and applications*. Hoboken, NJ: John Wiley & Sons.
- Perez-Losada, M., Arenas, M., Galán, J.C., Palero, F., Gonzalez-Candelas, F., 2015. Recombination in viruses: Mechanisms, methods of study, and evolutionary consequences. *Infect Genet Evol* 30C, 296–307.
- Perez-Losada, M., Crandall, K.A., Zenilman, J., Viscidi, R.P., 2007. Temporal trends in gonococcal population genetics in a high prevalence urban community. *Infect Genet Evol* 7, 271–278.
- Perez-Losada, M., Jobs, D.V., Sinangil, F., *et al.*, 2011. Phylodynamics of HIV-1 from a phase III AIDS vaccine trial in Bangkok, Thailand. *PLOS One* 6, e16902.
- Perez-Losada, M., Posada, D., Arenas, M., *et al.*, 2009. Ethnic differences in the adaptation rate of HIV gp120 from a vaccine trial. *Retrovirology* 6, 67.
- Posada, D., Crandall, K.A., 2002. The effect of recombination on the accuracy of phylogeny estimation. *J Mol Evol* 54, 396–402.

- Posada, D., Wiuf, C., 2003. Simulating haplotype blocks in the human genome. *Bioinformatics* 19, 289–290.
- Pritchard, J.K., Seielstad, M.T., Perez-Lezaun, A., Feldman, M.W., 1999. Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Mol Biol Evol* 16, 1791–1798.
- Rambaut, A., Grassly, N.C., 1997. Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosciences* 13, 235–238.
- Ramos-Onsins, S.E., Mitchell-Olds, T., 2007. Mcoalsim: Multilocus coalescent simulations. *Evol Bioinform Online* 3, 41–44.
- Rasteiro, R., Bouttier, P.A., Sousa, V.C., Chikhi, L., 2012. Investigating sex-biased migration during the neolithic transition in Europe, using an explicit spatial simulation framework. *Proc Biol Sci* 279, 2409–2416.
- Reppell, M., Boehnke, M., Zollner, S., 2012. FTEC: A coalescent simulator for modeling faster than exponential growth. *Bioinformatics* 28, 1282–1283.
- Schaffner, S.F., Foo, C., Gabriel, S., *et al.*, 2005. Calibrating a coalescent simulation of human genome sequence variation. *Genome Res* 15, 1576–1583.
- Schierup, M.H., Hein, J., 2000a. Consequences of recombination on traditional phylogenetic analysis. *Genetics* 156, 879–891.
- Schierup, M.H., Hein, J., 2000b. Recombination and the molecular clock. *Mol Biol Evol* 17, 1578–1579.
- Shlyakhter, I., Sabeti, P.C., Schaffner, S.F., 2014. Cosi2: An efficient simulator of exact and approximate coalescent with selection. *Bioinformatics* 30, 3427–3429.
- Shriner, D., Nickle, D.C., Jensen, M.A., Mullins, J.I., 2003. Potential impact of recombination on sitewise approaches for detecting positive natural selection. *Genet Res* 81, 115–121.
- Sipos, B., Massingham, T., Jordan, G.E., Goldman, N., 2011. PhyloSim – Monte Carlo simulation of sequence evolution in the R statistical computing environment. *BMC Bioinform* 12, 104.
- Slade, P.F., 2000a. Most recent common ancestor probability distributions in gene genealogies under selection. *Theor Popul Biol* 58, 291–305.
- Slade, P.F., 2000b. Simulation of selected genealogies. *Theor Popul Biol* 57, 35–49.
- Slade, P.F., Neuhauser, C., 2003. Nonneutral genealogical structure and algorithmic enhancements of the ancestral selection graph. *Comment Theor Biol* 8, 255–277.
- Slade, P.F., Wakeley, J., 2005. The structured ancestral selection graph and the many-demes limit. *Genetics* 169, 1117–1131.
- Spencer, C.C., Coop, G., 2004. SelSim: A program to simulate population genetic data with natural selection and recombination. *Bioinformatics* 20, 3673–3675.
- Spielman, S.J., Wilke, C.O., 2015. Pyvolve: A flexible python module for simulating sequences along phylogenies. *PLOS One* 10, e0139047.
- Staab, P.R., Zhu, S., Metzler, D., Lunter, G., 2015. SCRIM: Efficiently simulating long sequences using the approximated coalescent with recombination. *Bioinformatics* 31, 1680–1682.
- Sun, S., Evans, B.J., Golding, G.B., 2011. "Patchy-tachy" leads to false positives for recombination. *Mol Biol Evol* 28, 2549–2559.
- Sunnaker, M., Busetto, A.G., Numminen, E., *et al.*, 2013. Approximate Bayesian computation. *PLOS Comput Biol* 9, e1002803.
- Teshima, K.M., Innan, H., 2009. mbs: Modifying Hudson's ms software to generate samples of DNA sequences with a biallelic site under selection. *BMC Bioinform* 10, 166.
- Wang, J., Guo, M., Liu, X., *et al.*, 2013. LNETWORK: An efficient and effective method for constructing phylogenetic networks. *Bioinformatics* 29, 2269–2276.
- Wen, D., Yu, Y., Nakhleh, L., 2016. Bayesian inference of reticulate phylogenies under the multispecies network coalescent. *PLOS Genet* 12, e1006006.
- Westesson, O., Holmes, I., 2009. Accurate detection of recombinant breakpoints in whole-genome alignments. *PLOS Comput Biol* 5, e1000318.
- White, D.J., Bryant, D., Gemmell, N.J., 2013. How good are indirect tests at detecting recombination in human mtDNA? *G3 (Bethesda)* 3, 1095–1104.
- Wilson, D.J., Gabriel, E., Leatherbarrow, A.J., *et al.*, 2009. Rapid evolution and the importance of recombination to the gastroenteric pathogen *Campylobacter jejuni*. *Mol Biol Evol* 26, 385–397.
- Wiuf, C., Posada, D., 2003. A coalescent model of recombination hotspots. *Genetics* 164, 407–417.
- Worobey, M., Holmes, E.C., 1999. Evolutionary aspects of recombination in RNA viruses. *J Gen Virol* 80, 2535–2543.
- Wright, S., 1931. Evolution in Mendelian populations. *Genetics* 16, 97–159.
- Yang, Z., 2006. *Computational Molecular Evolution*. Oxford, England: Oxford University Press.
- Yu, Y., Dong, J., Liu, K.J., Nakhleh, L., 2014. Maximum likelihood inference of reticulate evolutionary histories. *Proc Natl Acad Sci USA* 111, 16448–16453.
- Yu, Y., Nakhleh, L., 2015. A maximum pseudo-likelihood approach for phylogenetic networks. *BMC Genomics* 16 (Suppl 10), S10.
- Zhao, H., Wei, Q., Fu, Y.-X., 2014. Coalescent analysis of modeling mutation process in colorectal cancer. *Cancer Res* 66, 370.

Further Reading

- Arenas, M., 2012. Simulation of molecular data under diverse evolutionary scenarios. *PLOS Comput Biol* 8, e1002495.
- Arenas, M., 2015. Advances in computer simulation of genome evolution: Toward more realistic evolutionary genomics analysis by approximate bayesian computation. *J Mol Evol* 80, 189–192.
- Beaumont, M.A., 2010. Approximate Bayesian computation in evolution and ecology. *Annu Rev Ecol Evol Syst* 41, 379–405.
- Hein, J., Schierup, M., Wiuf, C., 2005. *Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory*. Oxford University Press.
- Hudson, R.R., 1990. Gene genealogies and the coalescent process. *Oxford Surv Evol Biol* 7, 1–44.
- Hudson, R.R., 1998. Island models and the coalescent process. *Mol Ecol* 7, 413–418.
- Kaplan, N.L., Hudson, R.R., Iizuka, M., 1991. The coalescent process in models with selection, recombination and geographic subdivision. *Genet Res Camb* 57, 83–91.
- Kingman, J.F.C., 1982. The coalescent. *Stochas Process Appl* 13, 235–248.
- Mallo, D., Sánchez-Cobos, A., Arenas, M., 2016. Diverse considerations for successful phylogenetic tree reconstruction: Impacts from model misspecification, recombination, homoplasy, and pattern recognition. In: Elloumi, M., Iliopoulos, C., Wang, J., Zomaya, A. (Eds.), *Pattern Recognition in Computational Molecular Biology*. John Wiley & Sons, Inc.
- Neuhauser, C., Tavaré, S., 2001. The coalescent. *Encyclopedia of Genetics*. New York: Academic Press.
- Nordborg, M., 2007. Coalescent theory. In: Balding, D.J., Bishop, M., Cannings, C. (Eds.), *Handbook of Statistical Genetics*, third ed. Chichester, UK: John Wiley & Sons, Ltd.
- Notohara, M., 1990. The coalescent and the genealogical process in geographically structured population. *J Math Biol* 29, 59–75.
- Peng, B., Kimmel, M., Amos, C.I., 2012. *Forward-Time Population Genetics Simulations: Methods, Implementation, and Applications*. Hoboken, NJ: John Wiley & Sons.
- Yang, Z., 2006. *Computational Molecular Evolution*. Oxford, England: Oxford University Press.

Relevant Website

<http://www.coalescent.dk/>

Educational tools for understanding the coalescent simulation.

Biographical Sketch

Miguel Arenas is a principal investigator at the Department of Biochemistry, Genetics and Immunology of the University of Vigo. His research interests include the development of models, methods and frameworks to simulate and analyze the evolution of genetic material. He is also interested in the understanding of the evolution of organisms at both molecular and population levels.

Population Genetics

Conrad J Burden, Australian National University, Canberra, ACT, Australia

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

$A_1, A_2,$	Allele types.	V_x	The random number of offspring of individual x in the Cannings and Bienaymé-Galton-Watson models.
$B(\cdot, \cdot)$	The beta function.	$X(t)$	In the diffusion limit, relative proportion of A_1 alleles in the population at time t .
$f(x; p, t)$	Solution of the forward or backward Kolmogorov equation corresponding to the initial condition $X(0)=p$.	$Y(\tau)$	In discrete-time models, the number of individuals in the population of allele-type A_1 at timestep τ .
h	Degree of dominance.	$Y(\tau)$	In discrete-time models with multiple alleles, the number of individuals in the population of allele-type $A_1, \dots, A_K$ at timestep τ .
$I_1(\cdot)$	Modified bessel function of the first kind of order 1.	α	The relative mutation bias in a 2-allele model.
K	In multi-allelic models, the number of allele types.	γ	In the diffusion limit, the scaled selection coefficient.
m_0	Current population size for the coalescent; Initial population size in the Bienaymé-Galton-Watson model.	$\Gamma(\cdot)$	The gamma function.
M	Haploid population size, or for diploid populations equivalent haploid population size $2N$.	$\delta(\cdot)$	Dirac delta function. See 14.
n_{sample}	The size of a randomly chosen sample of individuals in the population.	θ	In the diffusion limit, the total mutation rate.
N	Diploid population size.	$\hat{\theta}_W$	The Watterson estimator.
p_{ij}	In discrete-time models, Markov transition matrix from $Y(\tau)=i$ to $Y(\tau+1)=j$.	λ	Expected number of offspring per individual in the Bienaymé-Galton-Watson model.
q_{ab}	In the diffusion limit, the rate of mutations from allele A_a to allele A_b .	π_i	Probability that allele A_1 fixes given $Y(0)=i$.
Q	In the diffusion limit with multiple alleles, the instantaneous rate matrix, whose elements are q_{ab} .	σ^2	Variance of the number of offspring per individual in the Bienaymé-Galton-Watson model.
s	Selection coefficient.	τ	Discrete time steps in Wright-Fisher and Bienaymé-Galton-Watson models.
S	The proportion of genomic sites which are segregating.	τ_C	The coalescent time in generations of two randomly chosen individuals in a population.
$\mathcal{S}$	The $(K-1)$ -dimensional simplex.	$\bar{\tau}(i)$	Expected time for allele A_1 to fix conditional on $Y(0)=i$.
t	Continuum time in the diffusion limit.	$\bar{\tau}^*(i)$	Expected time for allele A_1 to fix conditional on A_1 fixing and $Y(0)=i$.
T_i	The inter-coalescence time.	Ω_Z	The range of a random variable Z .
u_{ab}	In discrete-time models, the probability of allele A_a mutating to allele A_b in one generation.		

Introduction

If we were to zoom in on a multiple alignment of one of our chromosomes across the entire human population we might see something like [Fig. 1](#). Assuming it is not an X or Y, every individual has two copies of the chromosome. At first sight the sequence appears to be identical across the population. However we would soon notice certain isolated points called *single nucleotide polymorphisms* (SNPs) which have been highlighted by vertical bars in the figure. At most SNPs two possible letters are observed, for instance A and T in the left hand SNP, and C and T in the right hand SNP in [Fig. 1](#). The two possible letters are an example of what are referred to as *alleles*. Very occasionally three-allele SNPs occur ([Cao et al., 2015](#)), while four-allele SNPs are extremely rare indeed ([Phillips et al., 2015](#)). Sequencing of the human genome has revealed in excess of 10 million SNPs ([1000 Genomes Project Consortium, 2010](#); [Shen et al., 2013](#)), at an average separation of about 300 bases, or, to put it another way roughly 1 in 300 genomic sites on average is observed to be a SNP. Furthermore, as the number of individuals sequenced increases, so does the reported density of SNPs increase as more sites are revealed to be SNPs. SNPs occur at higher density in non-protein-coding regions than in coding regions. SNPs are the most common type of sequence variation, estimated to account for 90% of all sequence variation.

Before proceeding, a word about terminology is in order. The word allele introduced above is generically used in population genetics to mean any one of a number of alternate forms of a genetic locus. For instance, if a locus contains n SNPs with two

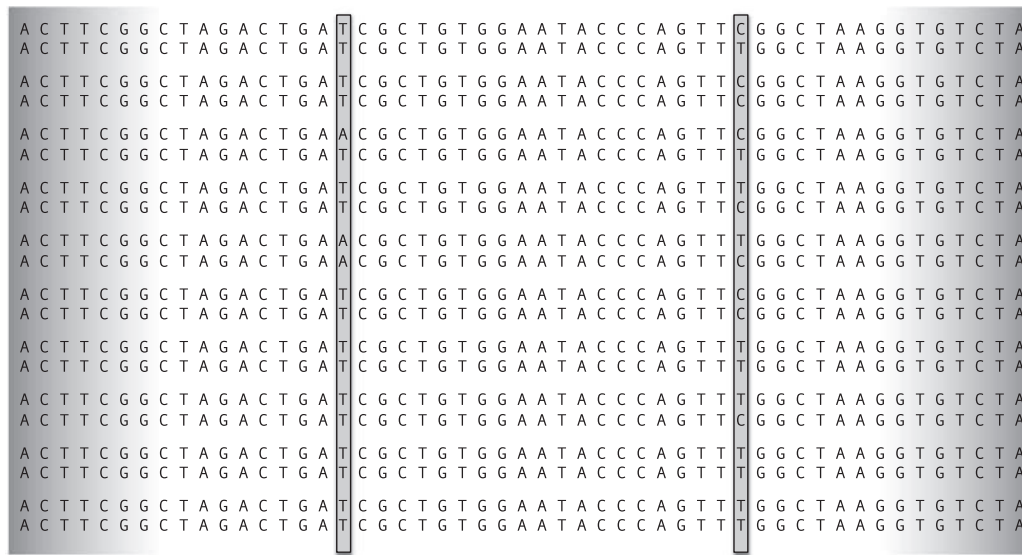


Fig. 1 A small portion of a multiple genomic alignment across the entire population showing a sample of ten individuals. Only one strand of the double helix is shown. The sequences are grouped into pairs to indicate that each individual has two copies of each chromosome. The single nucleotide polymorphisms, or SNPs, are highlighted.

alternate bases observed at each SNP, the 2^n possible patterns may be referred to as 2^n alleles. Many older texts use the word *gene* as a synonym for allele. We avoid this use here to avoid confusion with the more common use of the word in genomics, namely a protein-coding sequence of nucleotides. A more extensive discussion of various subclasses of allele can be found in the text by Gillespie (2004).

In addition to SNPs there are other genomic differences between individuals, such as the presence of copy number variations (CNVs). These occur when an extended region which may be anywhere from a kilobase to several megabases is repeated a number of times, the number of repeats varying from one individual to another. This variation accounts for roughly 5% to 10% of the human genome. Other forms of genetic variation include insertions and deletions of a single letter or of extended regions, and reversals of extended regions. Without genetic variations, people of a given sex would all look very similar to one another; in other words, we would all be clones.

As a general rule, each instance of genetic variation is caused by a copying error or mutation occurring in the germline in a single individual, which is passed on to future generations. Here the word *germline* refers to the lineage of cells which, as cells split, are passed on to the next generation. In the case of sexual reproduction, the germline comprises those sperm or eggs which contribute to the next generation and the lineage of cells from which they are descended, all the way back to the initial cell from which the individual developed, called the zygote. The vast majority of the estimated 10^{13} to 10^{14} cells in a human are of course not part of the germline. Population genetics is concerned with genetic differences across and within populations and the dynamics of how errors occurring in the germ lines of single individuals can in time propagate through an entire population to create the phenomenon we know as evolution.

In this article we will concentrate principally on SNPs and ignore other forms of genetic variation. Mathematical modelling of the propagation of SNPs through a population takes into account two processes: *genetic drift* and *selection*. Genetic drift is conceptually the more difficult of the two. It refers to the diffusion process by which an allele already present at a given SNP will become more or less prevalent in the population mainly as a result of variability in the number of offspring of each individual (Masel, 2011). In the next section we will look at the simplest a model of genetic drift, the Wright-Fisher model, which is traditionally used as a starting point from which to develop more detailed and realistic models of population genetics.

Selection refers to the force driving an allele frequency in one direction or the other as a result of the fitness conferred by the mutation from which the SNP arose. Many mutations are harmful and may for instance make it unlikely an individual will survive long enough to produce offspring. For such mutations the frequency of occurrences within the population of the mutant allele will be forced downwards over generations. Occasionally a mutation will occur which confers an advantage which makes the individual more fit to produce offspring, thus driving the allele frequency upwards over generations. Mutations which confer neither an advantage or a disadvantage, such as mutations in a redundant part of the genome, are called *neutral*. Selection is modelled mathematically by augmenting existing models of genetic drift. Mathematical models which take into account only drift and mutations and not selection are referred to as models of neutral evolution.

Naïvely one may think that for evolution to proceed from point mutations in single individuals in a large population, propagation of mutations must be assisted by positive selection: one would expect the probability of a random mutation in a single individual to drift through the entire population must be very small indeed. Surprisingly, we will see that this is not the case:

nucleotide substitutions can occur throughout an entire population through neutral evolution without the aid of positive selection, driven by nothing other than genetic drift.

The Wright-Fisher Model

The simplest and possibly the earliest serious mathematical model of population genetics, known as the Wright-Fisher (WF) model, is attributed to [Fisher \(1930\)](#) and [Wright \(1931\)](#), neither of whom published a clear definition of the model per se, though each did publish a number of consequences of the model. To begin with we will look at the simplest form, namely the two-allele WF model with genetic drift but not mutations or selection. We start with a number of assumptions:

Non-overlapping generations: Assume that evolution occurs over discrete timesteps $\tau=0, 1, 2, \dots$, so that the entire population is replaced by a new population each timestep.

Fixed population: Assume the population size N remains fixed, independent of τ .

Diploid population: Assume each individual has two copies of the genome.

Monoecious population: Assume each individual carries both male and female reproductive organs, so that mating can occur between two individuals or within one individual to produce offspring.

Random mating: Assume each individual is the offspring of two randomly and independently chosen parents from the previous generation. Since the population is monoecious, both parents may be the same individual.

This list of assumptions is highly restrictive, but does not rule out all species of life on earth. Annual plants complete their life cycle from germination to production of seed in one year and then die. Many trees, such as oaks, cedars and figs, are monoecious. In particular, sunflowers are both monoecious and annuals, and it would not be totally impossible to contrive a situation in which a sunflower crop's size is held more or less constant. In spite of these restrictions, its mathematical simplicity makes the WF model a very popular starting point for much of the published work in population genetics, even though the assumptions are clearly violated for most populations. Often one will encounter the concept of "effective population size", for which there seems to be no universally accepted unambiguous definition. Loosely speaking, it means the value to which the parameter N in an 'ideal' model with some or all of the above assumptions should be set in order to give correct results, depending on what characteristics one wants to calculate, when applied to a population for which the assumptions may not necessarily be appropriate. See Sections 1.6 and 3.7 of [Ewens \(2004\)](#) for a detailed discussion of this point.

Given the assumptions, it is more convenient mathematically to think in terms of a population of $M=2N$ copies of the genome, rather than a population of N diploid individuals. Equivalently, one may think of the model as applying to a haploid population with the following assumptions:

Non-overlapping generations: Assume that evolution occurs over discrete timesteps $\tau=0, 1, 2, \dots$, so that the entire population is replaced by a new population each timestep.

Fixed population: Assume the population size M remains fixed, independent of τ .

Haploid population: Assume each individual has one copy of the genome.

Random parents: Assume each individual is the offspring of one randomly and independently chosen parent.

Male ants, for instance, are haploid because they develop from unfertilised haploid eggs. Another example is mitochondrial DNA, which is normally only inherited from the mother in sexually reproducing species. The female portion of a population is therefore haploid with respect its mitochondrial genome.

Now consider a given locus within a genome at which there are two distinct alleles, A_1 and A_2 . Define the random variable $Y(\tau)$, whose range is $\Omega_{Y(\tau)} = \{0, 1, \dots, M\}$, to be the number of copies of allele A_1 within the (effective haploid) population of size $M=2N$. Under the assumption of fixed total population size and an assumption that every individual chromosome at timestep τ is equally likely to be the progenitor of this SNP in a given individual chromosome at timestep $\tau+1$, the number of A_1 alleles in the population at time $\tau+1$, given the number of A_1 alleles in the previous generation τ , follows a binomial distribution:

$$\text{Prob } (Y(\tau+1) = j | Y(\tau) = i) = p_{ij} \quad (1)$$

where

$$p_{ij} = \binom{M}{j} \left(\frac{i}{M}\right)^j \left(1 - \frac{i}{M}\right)^{M-j}, \quad i, j = 0, 1, \dots, M, \quad (2)$$

since each new offspring inherits either allele A_1 with probability i/M , or allele A_2 with probability $1 - i/M$. [Eqs. \(1\) and \(2\)](#) define the WF model without mutations or selection, that is, a model in which the only effect driving the dynamics is genetic drift.

This system is an example of a finite-state Markov chain, with states labelled $i=0, 1, \dots, M$ and transition matrix p_{ij} . [Eq. \(2\)](#) implies that

$$p_{0j} = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{if } j = 1, \dots, M \end{cases}, \quad p_{Mj} = \begin{cases} 0 & \text{if } j = 0, \dots, M-1 \\ 1 & \text{if } j = M. \end{cases} \quad (3)$$

Thus $Y(\tau)=0$ and $Y(\tau)=M$ are absorbing states of the Markov chain. These two states correspond respectively to the cases where the entire population has allele A_2 or A_1 at the site in question, and no further change is allowed by the model. If $Y(\tau)=M$ eventuates, we say that A_1 has become *fixed* in the population, and when $Y(\tau)=0$ we say A_2 has become fixed. If $Y(\tau) \in \{1, 2, \dots, M-1\}$, neither allele is yet fixed, and the site in question is said to be a *segregating site*.

Probability of Fixation

So far we have not built into the model any mechanism for a site to become a segregating site in the first place. We will come to that later when we deal with mutations in the section Including Mutations in the Wright-Fisher Model. For the time being, consider the initial condition $Y(0)=i$ for some $i=0, \dots, M$ and ask the question: What is the probability that allele A_1 becomes fixed? We have

$$\begin{aligned} & \text{Prob}(A_1 \text{ becomes fixed} | Y(0) = i) \\ &= \text{Prob}(Y(\infty) = M | Y(0) = i) \\ &= \sum_{j=1}^M \text{Prob}(Y(\infty) = M | Y(1) = j) \text{Prob}(Y(1) = j | Y(0) = i). \end{aligned} \quad (4)$$

We also have boundary conditions

$$\text{Prob}(Y(\infty) = M | Y(0) = 0) = 0, \quad \text{Prob}(Y(\infty) = M | Y(0) = M) = 1, \quad (5)$$

since 0 and M are absorbing states. Defining

$$\pi_i = \text{Prob}(Y(\infty) = M | Y(\tau) = i), \quad (6)$$

which we note is independent of τ since a Markov chain has no memory of prior events, Eqs. (4) and (5) become

$$\pi_i = \sum_{j=0}^M p_{ij} \pi_j, \quad \pi_0 = 0, \quad \pi_M = 1. \quad (7)$$

One easily checks that the solution is

$$\text{Prob}(A_1 \text{ becomes fixed} | Y(0) = i) = \pi_i = \frac{i}{M}. \quad (8)$$

Similarly, replacing the boundary conditions with $\pi_0 = 1, \pi_M = 0$ gives

$$\text{Prob}(A_2 \text{ becomes fixed} | Y(0) = i) = 1 - \pi_i = 1 - \frac{i}{M}, \quad (9)$$

implying that either one allele or the other must become fixed (Or, more precisely, one allele or the other becomes fixed ‘almost surely’, which is statistical jargon for ‘there exist trajectories in state space which never become fixed, but the probability of such a trajectory occurring is zero’.).

To interpret this result, consider a mutation which occurs in one member of the population at $\tau=0$. Intuitively one might expect that, for large populations, it is highly unlikely a mutation could become fixed by pure chance without the aid of natural selection. The above calculation shows that the WF model predicts a $1/M$ probability that a mutation in a single individual will become fixed in the population due to random genetic drift only. For instance, Fig. 2 shows a numerical simulation of 300 trajectories of the WF model with a population $M=100$, and, as predicted, in roughly 1 in 100 cases the mutation has become fixed (One case has not fixed on either allele yet, but A_1 looks like dying out after coming very close to fixing). On the other hand, the rate at which such mutations occur anywhere in the genome is proportional to the population size M , so we arrive at the surprising result that, even without natural selection, the model predicts that random mutations can fix in a population at a rate which is approximately independent of population size.

Expected Time to Fixation

A second question suggested by Fig. 2 is: What is the expected time for an allele to fix in the WF model? Consider first the unconditional problem of fixation in either of the two absorbing states. Define $\bar{\tau}(i)$ to be the expected number of time steps before fixation at either 0 or M given the initial conditions $Y(0)=i$. For either of the cases $i=0$ or $i=M$ the answer is trivial, while for any other value of i we can write a difference equation based on a sum of conditional expectation values labelled by the first step,

$$\bar{\tau}(i) = \sum_{j=0}^M p_{ij} \bar{\tau}(j) + 1, \quad i = 1, \dots, M-1, \quad (10)$$

$$\bar{\tau}(0) = \bar{\tau}(M) = 0. \quad (11)$$

This equation does not admit a simple analytic solution. However, assuming M is large, we can obtain a good approximation to the solution by taking a continuum limit in time

$$t = \frac{\tau}{M}, \quad \delta t = \frac{1}{M}, \quad (12)$$

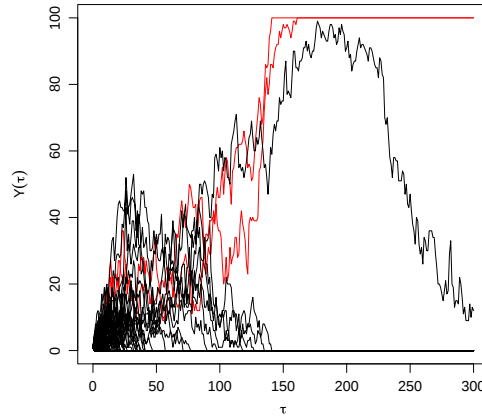


Fig. 2 Simulation of the WF model: 300 independent trajectories for a population of $M=100$ copies of the genome, with a mutant allele introduced into one member of the population at $\tau=0$. Trajectories for which the mutant allele has become fixed are shown in red.

while at the same time defining a rescaled continuous random variable

$$X(t) = \frac{1}{M} Y(\tau), \quad \Omega_{X(t)} = [0, 1], \quad (13)$$

equal to the fraction of the population with allele A_1 . In Eq. (10) we make the substitutions

$$x = \frac{i}{M}, \quad u = \frac{j}{M}, \quad \bar{t}(x) = \frac{1}{M} \bar{\tau}(i), \quad (14)$$

and set

$$\delta X = X(\delta t) - X(0) = \frac{1}{M} (Y(1) - Y(0)), \quad (15)$$

to obtain

$$\begin{aligned} \bar{t}(x) &= \int_0^1 \text{Prob}(X(\delta t) = u | X(0) = x) \bar{t}(u) du + \frac{1}{M} \\ &= \int_0^1 \text{Prob}(X(\delta t) = u | X(0) = x) \\ &\quad \times \left\{ \bar{t}(x) + (u-x) \bar{t}'(x) + \frac{1}{2} (u-x)^2 \bar{t}''(x) + \dots \right\} du + \delta t \\ &= \left\{ \bar{t}(x) + E(\delta X | X(0) = x) \bar{t}'(x) + \frac{1}{2} E(\delta X^2 | X(0) = x) \bar{t}''(x) + \dots \right\} + \delta t. \end{aligned} \quad (16)$$

Eq. (16) holds for any model scaled according to Eqs. (12) and (13) irrespective of the form of the transition matrix p_{ij} . For the WF model in particular, $Y(1) | (Y(0)=i) \sim \text{Bin}(M, i/M)$, so

$$\begin{aligned} E(\delta X | X(0) = x) &= \frac{1}{M} E(Y(1) - Y(0) | Y(0) = i) \\ &= \frac{1}{M} \{E(Y(1) | Y(0) = i) - i\} \\ &= \frac{1}{M} \left(M \times \frac{i}{M} - i \right) \\ &= 0, \end{aligned} \quad (17)$$

and

$$\begin{aligned} E(\delta X^2 | X(0) = x) &= \text{Var}(\delta X | X(0) = x) + E(\delta X | X(0) = x)^2 \\ &= \frac{1}{M^2} \text{Var}(Y(1) - Y(0) | Y(0) = i) + 0 \\ &= \frac{1}{M^2} \left\{ M \times \frac{i}{M} \times \left(1 - \frac{i}{M} \right) \right\} \\ &= x(1-x) \delta t. \end{aligned} \quad (18)$$

Similarly one can show

$$E(\delta X^k | X(0) = x) = o(\delta t), \quad \text{for } k = 3, 4, \dots \text{ as } \delta t \rightarrow 0. \quad (19)$$

Substituting back into Eq. (16) and tidying up gives the differential equation

$$\frac{1}{2}x(1-x)\bar{t}''(x) + 1 = 0, \quad (20)$$

and from Eq. (11) we have the boundary conditions

$$\bar{t}(0) = \bar{t}(1) = 0. \quad (21)$$

One easily checks that the solution is

$$\bar{t}(x) = -2 \{x \log x + (1-x) \log(1-x)\} \quad (22)$$

or

$$\bar{t}(i) \approx -2M \left\{ \frac{i}{M} \log \frac{i}{M} + \left(1 - \frac{i}{M}\right) \log \left(1 - \frac{i}{M}\right) \right\}. \quad (23)$$

For instance, a SNP with both alleles equally populated could be expected to fix in $\bar{t}(M/2) \approx 1.39M$ generations.

A more pertinent question relevant to evolution might be to ask the expected time to fixation for trajectories such as those highlighted in red in Fig. 2, where a newly arrived allele A_1 has managed to fix. To answer this question, consider a new random variable

$$Y^*(\tau) = Y(\tau) | (Y(\infty) = M), \quad \Omega_{Y^*} = 1, \dots, M, \quad (24)$$

equal to the number of individuals in the population with allele A_1 , conditioned on the restricted set of trajectories for which A_1 fixes. The corresponding Markov transition matrix is

$$\begin{aligned} p_{ij}^* &= \text{Prob}(Y(\tau+1) = j | Y(\tau) = i, Y(\infty) = M) \\ &= \frac{\text{Prob}(Y(\tau+1) = j, Y(\infty) = M | Y(\tau) = i)}{\text{Prob}(Y(\infty) = M | Y(\tau) = i)} \\ &= \frac{\text{Prob}(Y(\tau+1) = j | Y(\tau) = i) \text{Prob}(Y(\infty) = M | Y(\tau+1) = j)}{\text{Prob}(Y(\infty) = M | Y(\tau) = i)} \\ &= \frac{p_{ij}\pi_j}{\pi_i}, \quad i, j = 1, \dots, M \end{aligned} \quad (25)$$

where, in the third line, we have used the Markovian property that the trajectory $\tau \rightarrow \tau+1$ is independent of the trajectory $\tau+1 \rightarrow \infty$.

For the case of the WF model, it is straightforward to check that Eqs. (2), (8), and (25) imply

$$p_{ij}^* = \binom{M-1}{j-1} \left(\frac{i}{M}\right)^{j-1} \left(1 - \frac{i}{M}\right)^{M-j}, \quad i, j = 1, \dots, M, \quad (26)$$

and that consequently the random variable $Y^*(\tau+1) - 1$ conditioned on $Y^*(\tau) = i$ is binomial:

$$Y^*(\tau+1) | (Y^*(\tau) = i) \sim \text{Bin}(M-1, i/M) + 1. \quad (27)$$

Following the same line of reasoning as for the unrestricted case, we find that Eq. (16) also applies to an analogous set of starred variables defined by

$$X^*(t) = \frac{1}{M} Y^*(\tau), \quad \delta X^* = \frac{1}{M} (Y^*(1) - Y^*(0)), \quad \bar{t}^*(x) = \frac{1}{M} \bar{t}^*(i). \quad (28)$$

However, calculations analogous to Eqs. (17) and (18) now give

$$E(\delta X^* | X^*(0) = x) = (1-x) \delta t, \quad (29)$$

$$E(\delta X^{*2} | X^*(0) = x) = x(1-x) \delta t + O(\delta t^2), \quad (30)$$

leading to the differential equation

$$(1-x)\bar{t}^{*'}(x) + \frac{1}{2}x(1-x)\bar{t}^{*''}(x) + 1 = 0, \quad (31)$$

for the expected time to fixation conditional on A_1 fixing, $\bar{t}^*(x)$. The solution should also satisfy the boundary condition

$$\bar{t}^*(1) = 0, \quad (32)$$

and the symmetry condition

$$\begin{aligned} \bar{t}(x) &= \text{Prob}(X(\infty) = 1) \bar{t}^*(x) + \text{Prob}(X(\infty) = 0) \bar{t}^{**}(x) \\ &= x \bar{t}^*(x) + (1-x) \bar{t}^{**}(1-x), \end{aligned} \quad (33)$$

where $\bar{t}(x)$ is given by Eq. (22) and $\bar{t}^{**}(x)$ is the expected time to fixation conditional on A_2 fixing. The required solution is

$$\bar{t}^*(x) = -\frac{2}{x} (1-x) \log(1-x), \quad (34)$$

or

$$\bar{\tau}^*(i) = -\frac{2M^2}{i} \left(1 - \frac{i}{M}\right) \log \left(1 - \frac{i}{M}\right). \quad (35)$$

If a mutation is introduced into a single member of a large population at $\tau=0$ and the mutation fixes, Eq. (35) says that the expected time to fix is approximately $\bar{\tau}^*(1) \approx 2M(1 + O(1/M))$. The red trajectories in Fig. 2 are broadly in agreement with this result.

Including Mutations in the Wright-Fisher Model

So far mutations have only been included in the model as an ad hoc specification of initial conditions. In this section we introduce mutations dynamically in a way which can be generalised to incorporate instantaneous rate matrices for multiple alleles in the section Multiple Alleles.

Consider again a locus in which only two alleles are present, or equivalently, assume a site in a simplified model genome in which the genomic alphabet only consists of two letters, A_1 and A_2 . Returning to the assumptions of the WF model, we add one more assumption:

Random mutation: Following the germline over one generation, a locus occupied by an A_1 allele at the initial time of inheritance has a probability u_{12} of mutating to A_2 before reproductive maturity, and a genomic site initially occupied by an A_2 allele has a probability u_{21} of mutating to A_1 before reproductive maturity.

By ‘reproductive maturity’ we mean the point at which the genomic information is passed onto the next generation. In principle any values in the range $0 \leq u_{12}, u_{21} \leq 1$ are allowed by the mathematics, though in practice the mutation rates per generation are typically very small.

Now interpret the random variable $Y(\tau)$ as the number of copies of allele A_1 within the population at birth. Conditioning on the event $Y(\tau)=i$ for some $i=0, \dots, M$, the following four events are possible in any one member of the population:

- E1: The locus inherits allele A_1 and does not mutate;
- E2: The locus inherits allele A_1 and mutates to A_2 ;
- E3: The locus inherits allele A_2 and does not mutate;
- E4: The locus inherits allele A_2 and mutates to A_1 .

Then the probability the locus will be occupied by allele A_1 at maturity, given that $Y(\tau)=i$ at birth, is

$$\psi(i) = \text{Prob}(E_1) + \text{Prob}(E_4) = \frac{i}{M}(1 - u_{12}) + \left(1 - \frac{i}{M}\right)u_{21} \quad (36)$$

and the probability it will be occupied by allele A_2 is

$$1 - \psi(i) = \text{Prob}(E_2) + \text{Prob}(E_3) = \frac{i}{M}u_{12} + \left(1 - \frac{i}{M}\right)(1 - u_{21}) \quad (37)$$

Again we see that $Y(\tau+1)|Y(\tau)=i$ is a binomial random variable, but the Markov transition matrix defined by Eq. (1) is modified from Eq. (2) to

$$p_{ij} = \binom{M}{j} \psi(i)^j (1 - \psi(i))^{M-j}, \quad i, j = 0, 1, \dots, M, \quad (38)$$

where $\psi(i)$ is defined above.

One can check that if u_{12} and u_{21} are both non-zero, the Markov chain has no absorbing states; if $u_{12}=0$ and $u_{21}>0$, $Y(\tau)=M$ is the only absorbing state, implying that A_1 must fix (almost surely); and if $u_{21}=0$ and $u_{12}>0$, $Y(\tau)=0$ is the only absorbing state, implying that A_2 must fix (almost surely).

Fig. 3 shows a simulation of a trajectory of this model with parameter values $M=100$, $u_{12}=0.0005$ and $u_{21}=0.0003$ chosen to demonstrate the behaviour. In most real-world scenarios populations are much higher (in studies of humans effective population sizes of $\sim 10^5$ are often used) and mutation rates much lower (human genomic mutation rates at a neutral site may typically be $\sim 10^{-8}$ per individual per generation). Nevertheless, we note the important features that most of the time a genomic site is not segregating (i.e. $Y(\tau)=0$ or M), and the transition between the two almost-fixed states $Y(\tau)=0$ and M occurs over a time roughly the same order of magnitude as the fixation time $\bar{\tau}^*(1) \approx 2M$ obtained for the simpler model of Section Expected Time to Fixation. A similar effect is explained by Vogl and Bergman (2015) in the context of the 2-allele Moran model as a strong dominance of genetic drift over mutations for polymorphic sites.

The rate at which new alleles fix throughout a population is called the *substitution rate*. For the current WF model it can be estimated as follows. For a given site which is currently non-segregating of type A_1 , the number of individuals out of a population of size M who mutate to type A_2 in a given generation is Poisson with parameter $u_{12}M$. Thus the probability that at least one mutation occurs is $1 - e^{-u_{12}M} \approx u_{12}M$ for $u_{12}M \ll 1$. Thus the expected time between ‘attempted’ mutations in a population of size

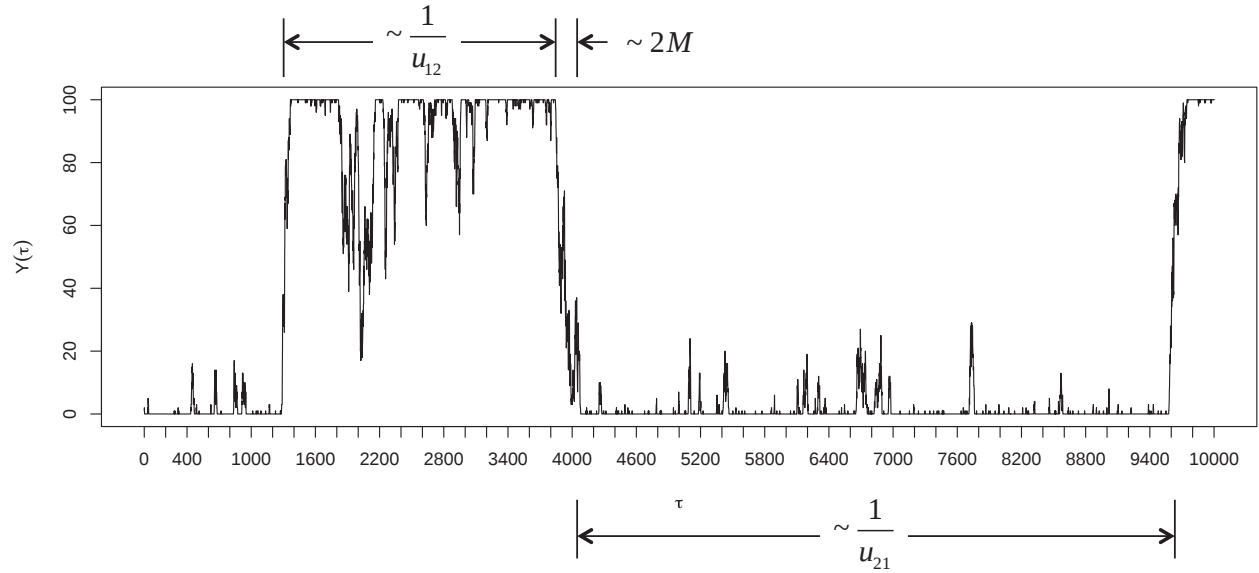


Fig. 3 Simulation of a single trajectory of the WF model with mutations. The parameters are: population of $M=100$ copies of the genome, mutation rate A_1 to A_2 of $u_{12}=0.0005$ per generation and A_2 to A_1 of $u_{21}=0.0003$ per generation. $Y(\tau)$ is the number of A_1 alleles in the population. As described in the text, the expected substitution time before an A_2 allele fixes given that a site is currently fixed at A_1 , is approximately $1/u_{12}$; the expected A_2 to A_1 substitution time is approximately $1/u_{21}$, and the expected time the substitution takes to occur is $2M$.

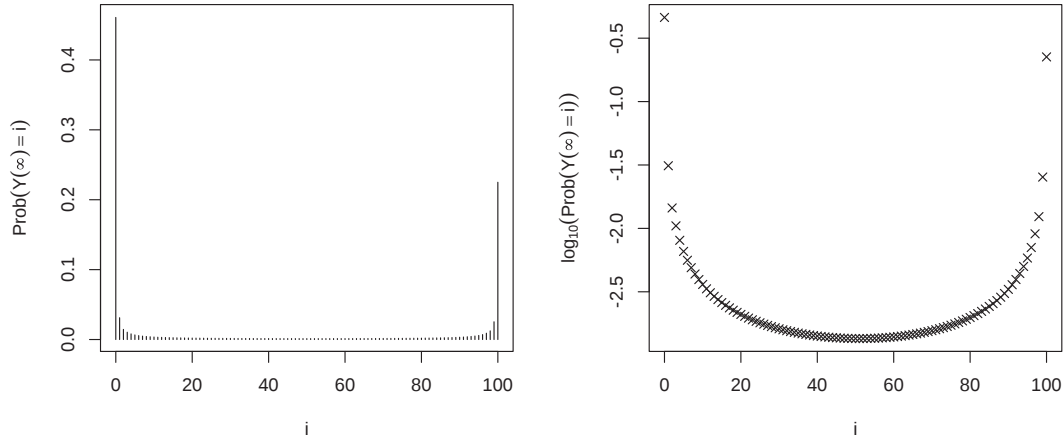


Fig. 4 Stationary distribution of WF model with mutations. Parameters are the same as for Fig. 3. The right-hand panel is the same plot on a logarithmic scale.

M is $1/(u_{12}M)$. Of those attempted mutations, Eqs. (8) and (9) tell us that a fraction $1/M$ will fix throughout the population, assuming fixation is driven mainly by drift for low mutation rates. Thus the expected time to the next time allele A_2 fixes throughout the population at the site in question is $M \times 1/(u_{12}M) = 1/u_{12}$, and the population's substitution rate for A_1 to A_2 is u_{12} . Similarly the substitution rate for A_2 to A_1 is u_{21} . In summary, for the WF model with low mutation rates, the population's substitution rate matches the genomic mutation rate. To put it another way, if we consider the genomic alphabet to consist of only 2 letters instead of 4, the instantaneous rate matrix appropriate to, say, maximum likelihood estimates of phylogenetic trees, is

$$U = \begin{pmatrix} -u_{12} & u_{12} \\ u_{21} & -u_{21} \end{pmatrix} \quad (39)$$

provided time from one generation to the next is used as the unit of time.

Another way to interpret the two-allele WF model with mutations is to consider the stationary distribution of the Markov transition matrix Eq. (38), namely $\text{Prob}(Y(\infty)=i)$. The numerically determined distribution is plotted in Fig. 4 for the same set of parameters as Fig. 3. Recall that the model can be thought of as a toy model of a genomic site corresponding to a world in which the genomic alphabet has only two possible nucleotides, A_1 and A_2 . If we consider the genome to be a set of independent genomic sites which have been evolving according to the WF model for a considerable time, this distribution can be thought of as the

distribution of allele frequencies throughout the genome. Most genomic sites are not segregating, but contribute to the spike in the left hand plot at $i=100$ if the site is occupied by nucleotide A_1 or the spike at $i=0$ if the site is occupied by nucleotide A_2 . The remaining vertical bars in the plot correspond to the segregating sites, or SNPs, the height of the bar being the proportion of sites in a large genome at which the allele A_1 occurs in a fraction i/M of the population. This distribution is called the *site frequency spectrum* (SFS). The SFS is important because it is directly observable from sequencing data, and can, in principle, enable estimation of mutation rates under the assumption of a given population genetics model. In a later section we show how to solve for the SFS for WF type models in the continuum limit $M \rightarrow \infty$.

Before moving on, we make the observation in passing that the model we have considered is of course a very crude approximation to the observed SFS of any real genome, even if the assumptions of the Wright-Fisher Model hold and we restrict ourselves to parts of the genome undergoing neutral evolution, that is, evolution without selection. One important problem is that genetic drift is an extremely slow process for large populations, calling into question any assumption of stationarity (Gillespie, 2004, pp. 25–26). Another problem is that sites are not independent to the extent that mutation rates are known to depend on neighbouring bases (Aggarwala and Voight, 2016; Zhu *et al.*, 2017). Models which include neighbouring-base context dependence are very difficult to make any progress with analytically, because the neighbouring bases themselves mutate on the same timescale. Consequently, analytic results for population genetics models which include context dependence are almost completely absent from the scientific literature.

Including Selection in the Wright-Fisher Model

Return now to the set of assumptions set out in the section. The Wright-Fisher Model leading to the WF model for a diploid population. Note that each individual in the population can be any one of three *genotypes*: A_1A_1 , A_1A_2 and A_2A_2 . Suppose we add one extra assumption:

Selection: Assume that the probability an individual survives to maturity and therefore has the potential to procreate is dependent on the genotype of the individual. This probability is known as the *fitness* of the genotype.

A common convention is to specify the ratios of relative fitnesses of the three genotypes, without explicit regard to normalisation. In order to specify three relative fitnesses, two parameters are needed. Here we will adopt the convention used in Equation (1.2b) of Ewens (2004) and Section 5.3 of Etheridge (2011) that the relative fitnesses of the three genotypes are in the proportions

$$1 + s : 1 + hs : 1 \quad (40)$$

where s is called the *selection coefficient* and h is called the *degree of dominance*: if h is close to 1, A_1 is dominant to A_2 in fitness; if h is close to zero, A_2 is dominant to A_1 in fitness; and if $h = \frac{1}{2}$ there is no dominance in fitness.

If, at the beginning of generation τ , the number of A_1 alleles in a population of size $M=2N$ is $Y(\tau)=i \in \{0, \dots, M\}$, then the assumption of random mating implies that the relative proportion of new-born genotypes A_1A_1 , A_1A_2 and A_2A_2 is

$$i^2 : 2i(M-i) : (M-i)^2. \quad (41)$$

This set of relative proportions is known as *Hardy-Weinberg equilibrium* (HWE) (Hardy, 1908; Stern, 1943). The relative proportion of surviving mature genotypes in the population is then

$$i^2(1+s) : 2i(M-i)(1+hs) : (M-i)^2 \quad (42)$$

The number of A_1 alleles present in the newborn population at time $\tau+1$, given that there were i type- A_1 alleles in the newborn population at time τ , that is $Y(\tau+1)|(Y(\tau)=i)$, will once again be a binomial random variable with Markov transition matrix of the form

$$p_{ij} = \binom{M}{j} \eta(i)^j (1-\eta(i))^{M-j}, \quad i, j = 0, 1, \dots, M. \quad (43)$$

Here $\eta(i)$ is the probability that a given newborn allele will be A_1 . Note that random mating of the previous mature generation ensures that the $\eta(i)$ are inherited from genotypes of the previous mature generation who are in the relative ratios

$$\eta(i)^2 : 2\eta(i)(1-\eta(i)) : (1-\eta(i))^2. \quad (44)$$

Comparing Eqs. (42) and (44) we have

$$\eta(i)^2 = \frac{i^2(1+s)}{w}, \quad 2\eta(i)(1-\eta(i)) = \frac{2i(M-i)(1+hs)}{w}, \quad (45)$$

where $w = i^2(1+s) + 2i(M-i)(1+hs) + (M-i)^2$. Thus

$$\begin{aligned} \eta(i) &= \eta(i)^2 + \eta(i)(1-\eta(i)) \\ &= \frac{i^2(1+s) + i(M-i)(1+hs)}{i^2(1+s) + 2i(M-i)(1+hs) + (M-i)^2}. \end{aligned} \quad (46)$$

Thus, Eqs. (43) and (46) define the Markov transition matrix of the 2-allele WF model for a randomly-mating diploid population with selection, but no mutation.

Including selection in the WF model for a haploid population is somewhat simpler. In this case there is no distinction between allele types and genotypes. Without loss of generality we can define relative fitnesses of the two allele types A_1 and A_2 in terms of a single parameter s as

$$1 + \frac{1}{2}s : 1. \quad (47)$$

If at time step τ the number of newborn A_1 alleles is $Y(\tau)=i$ and the number of newborn A_2 alleles is $M-i$, then the relative probabilities of an individual inheriting an A_1 or A_2 allele are

$$i(1 + \frac{1}{2}s) : M - i \quad (48)$$

implying that the probability of a newborn individual at time step $\tau+1$ being A_1 is

$$\eta(i) = \frac{i(1 + \frac{1}{2}s)}{i(1 + \frac{1}{2}s) + M - i}. \quad (49)$$

Eqs. (43) and (49) define the Markov transition matrix of the 2-allele WF model for a haploid population with selection, but no mutation.

Diffusion Limit and the Forward Kolmogorov Equation

Although no analytic solution exists for the SFS shown in Fig. 4 corresponding to the stationary distribution of the WF model with mutations, Eq. (38), or for the analogous stationary distribution of the WF model with selection, Eq (43), very close approximations can be found in the form of exact stationary distributions of a continuum model constructed by taking the so-called *diffusion limit* $M \rightarrow \infty$. This limit involves constructing a partial differential equation known variously as the *forward Kolmogorov equation* or *Fokker-Planck equation*, which describes the time evolution of a probability density over allele frequencies.

More generally, the Fokker-Planck equation was published by Adriaan Fokker in 1914 and Max Planck in 1917 in the context of studying the velocity distribution of particles undergoing Brownian motion. The forward Kolmogorov equation is equivalent and was published by Andrei Kolmogorov in 1931 in the context of a more general study of the continuous-time diffusion limit of Markov processes. The influence of the diffusion limit and the forward Kolmogorov equation in population genetics is generally attributed to Kimura (1964) who obtained diffusion limit solutions to a number of models including the neutral WF model (Kimura, 1955a) and the WF model with selection (Kimura, 1955b).

Appendix A contains a derivation of the forward Kolmogorov equation for a generic population genetics model based on a discrete time finite state Markov chain. The derivation given is for a genomic site at which two alleles A_1 and A_2 are possible in a population of fixed size M . We assume also that we are given a time-independent Markov transition matrix

$$p_{ij} = \text{Prob}(Y(\tau+1) = j | Y(\tau) = i), \quad i, j = 0, \dots, M, \quad (50)$$

where $Y(\tau)$ are the number of A_1 alleles present in the population at discrete times $\tau=0,1,2,\dots$,

With the intention of taking the limit $M \rightarrow \infty$, consider the substitutions which proved useful above for calculating fixation times:

$$t = \frac{\tau}{M}, \quad \delta t = \frac{1}{M}, \quad X(t) = \frac{1}{M} Y(\tau). \quad (51)$$

After taking the limit $M \rightarrow \infty$, t becomes a continuous time variable over the interval $[0, \infty)$ and the fraction $X(t)$ of the population with allele A_1 becomes a continuous random variable with range $\Omega_X = [0, 1]$. The aim of the forward Kolmogorov equation is to describe the evolution of the probability density $f(x; p, t)$ corresponding to $X(t)$, conditional on an initial condition $X(0) = p$, that is,

$$f(x; p, t) dx = \text{Prob}(x \leq X(t) < x + dx | X(0) = p), \quad 0 \leq x, p \leq 1, \quad t > 0. \quad (52)$$

In order to obtain a sensible continuum limit from Eq. (51), there is an extra requirement that the moments of the random variable $\delta X(t) = X(t + \delta t) - X(t)$ must be of the form

$$\begin{aligned} E[\delta X(t) | X(t) = u] &= a(u) \delta t + o(\delta t) \\ E[(\delta X(t))^2 | X(t) = u] &= b(u) \delta t + o(\delta t) \\ E[(\delta X(t))^k | X(t) = u] &= o(\delta t), \quad k = 3, 4, \dots, \end{aligned} \quad (53)$$

as $\delta t \rightarrow 0$ for finite, well behaved functions $a(u)$ and $b(u)$ defined for $0 \leq u \leq 1$. We have already seen this condition satisfied for the WF model by Eqs. (17) to (19) and the WF model conditioned on fixing allele A_1 by Eqs. (29) and (30), and below we will also see it satisfied for WF model with mutations and selection. For some population genetics models such as the Moran model

described later in the chapter, the continuum limit is defined differently to Eq. (51), in which case Eq. (53) must be adjusted accordingly.

With this machinery in place it is shown in Appendix A that the forward Kolmogorov equation takes the form

$$\frac{\partial f(x; p, t)}{\partial t} = -\frac{\partial}{\partial x} [a(x)f(x; p, t)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [b(x)f(x; p, t)]. \quad (54)$$

For any specific population genetics model, the functions $a(x)$ and $b(x)$ are determined from the Markov transition matrix p_{ij} .

The Continuum Wright-Fisher Model With Mutations

We now return to the WF model with 2-way mutations derived earlier. Including Mutations in the Wright-Fisher Model. To determine the functions $a(x)$ and $b(x)$ in the forward Kolmogorov equation, we need to apply the continuum limit defined by Eq. (51). Because the mutation rates u_{12} and u_{21} are defined as probabilities of mutation per generation, and the generation time $\delta t = 1/M$ becomes infinitesimal in the limit $M \rightarrow \infty$, rescaled mutation rates with respect to the continuum time t are needed. Accordingly we set

$$q_{12} = Mu_{12}, \quad q_{21} = Mu_{21}. \quad (55)$$

Note from the argument leading to Eq. (39) that when $q_{12}, q_{21} < 1$, the matrix

$$Q = \begin{pmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{pmatrix}, \quad (56)$$

is also the instantaneous rate matrix with respect to the continuum time t for allele substitution throughout the population.

For the transition matrix Eq. (38),

$$\begin{aligned} E[\delta X(t) | X(t) = x] &= \frac{1}{M} E[Y(\tau + 1) - Y(\tau) | Y(\tau) = i] \\ &= \frac{1}{M} (M\psi(i) - i) \\ &= -\frac{i}{M} u_{12} + \left(1 - \frac{i}{M}\right) u_{21} \\ &= [-q_{12}x + q_{21}(1-x)]\delta t. \end{aligned} \quad (57)$$

Comparing with Eq. (53) then gives

$$a(x) = -q_{12}x + q_{21}(1-x), \quad (58)$$

while a similar calculation using the variance gives

$$b(x) = x(1-x). \quad (59)$$

We are now in a position to determine the stationary distribution of the continuum limit of the WF model with two-way mutations. Substituting $a(x)$ and $b(x)$ into Eq. (54), setting $\partial f/\partial t = 0$ and integrating once gives the ordinary differential equation

$$[q_{12}x - q_{21}(1-x)]f(x) + \frac{1}{2} \frac{d}{dx} [x(1-x)f(x)] = \text{const.} \quad (60)$$

It is possible to show that the constant of integration on the right hand side corresponds to a flux of probability across the boundaries at $x=0$ and $x=1$, which must be zero. Then it is straightforward to check that the function

$$f(x) = \frac{1}{B(2q_{21}, 2q_{12})} x^{2q_{21}-1} (1-x)^{2q_{12}-1} \quad 0 \leq x \leq 1, \quad (61)$$

is a solution to the ordinary differential equation and is thus the required stationary distribution of the forward Kolmogorov equation. Here the prefactor, which is written in terms of the beta function, whose definition is (Abramowitz *et al.*, 1972)

$$B(z_1, z_2) = \int_0^1 x^{z_1-1} (1-x)^{z_2-1} dx, \quad z_1, z_2 > 0, \quad (62)$$

ensures that the probability density function $f(x)$ has the correct normalisation, i.e., $\int_0^1 f(x) dx = 1$. This distribution is the well known beta distribution, and was first identified as the stationary distribution in the presence of two-way mutations by Wright (1931), without explicit reference to the forward Kolmogorov equation.

Fig. 5 shows the SFS for the example in Fig. 4 reproduced with the stationary solution to the continuum theory, Eq. (61), superimposed. The horizontal and vertical axes have been rescaled by factors of $1/M$ and M respectively in order to make the comparison and the parameters $2q_{12}=0.05$ and $2q_{21}=0.03$ calculated from Eq. (55). To compare with the SFS at $i=0$ and M ,

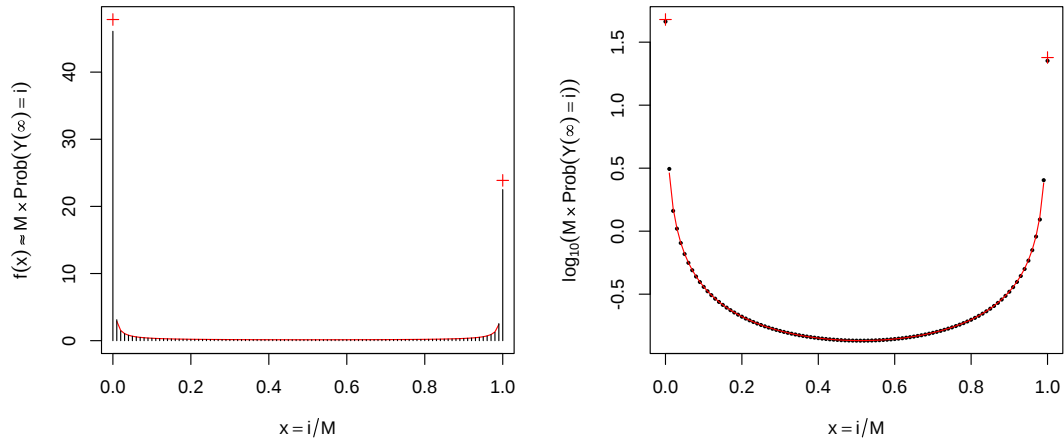


Fig. 5 The SFS computed in Fig. 4 for a WF model with two-way mutation for a finite population M , rescaled to the continuum parameter $x=i/M$ (black), and the continuum stationary solution Eq. (61) to the forward Kolmogorov equation with corresponding parameters $2q_{12}=0.05$ and $2q_{21}=0.03$ (red). The red plus signs at $x=0$ and $x=1$ are computed from Eqs. (63) and (64).

contributions from the singularities in $f(x)$ at $x=0$ and $x=1$ are estimated as

$$\begin{aligned} \text{Prob}(Y(\infty) = 0) &\approx \int_0^{1/M} f(x) dx \\ &\approx \frac{1}{B(2q_{21}, 2q_{12})} \int_0^{1/M} x^{2q_{21}-1} dx \\ &= \frac{M^{-2q_{21}}}{2q_{21}B(2q_{21}, 2q_{12})}, \end{aligned} \quad (63)$$

and

$$\begin{aligned} \text{Prob}(Y(\infty) = 1) &\approx \int_{1-1/M}^1 f(x) dx \\ &\approx \frac{1}{B(2q_{21}, 2q_{12})} \int_{1-1/M}^1 (1-x)^{2q_{12}-1} dx \\ &= \frac{M^{-2q_{12}}}{2q_{12}B(2q_{21}, 2q_{12})}. \end{aligned} \quad (64)$$

Clearly, the agreement in Fig. 5 is excellent even at this moderate value of M , indicating that the continuum limit $M \rightarrow \infty$ is reached rapidly.

In biologically relevant situations it often happens that $q_{12}, q_{21} < 1$. In this case a useful approximation to the stationary distribution is

$$f(x) \approx \frac{2q_{12}q_{21}}{q_{12} + q_{21}} \left(\frac{1}{x} + \frac{1}{1-x} \right), \quad (65)$$

valid for $q_{21} \log x < 1$ and $q_{12} \log(1-x) < 1$, with the terminal spikes approximated by

$$\begin{aligned} \text{Prob}(Y(\infty) = 0) &\approx \frac{q_{12}}{q_{12} + q_{21}} (1 - 2q_{21} \log M), \\ \text{Prob}(Y(\infty) = 1) &\approx \frac{q_{21}}{q_{12} + q_{21}} (1 - 2q_{12} \log M). \end{aligned} \quad (66)$$

These approximations are a consequence of the property of the beta function that

$$\frac{1}{B(\varepsilon, \eta)} = \frac{\varepsilon\eta}{\varepsilon + \eta} (1 + O(\varepsilon) + O(\eta)) \quad \text{as } \varepsilon, \eta \rightarrow 0. \quad (67)$$

The function Eq. (65) is easily seen to be a solution to Eq. (60) with q_{12} and q_{21} set to zero. The physical import of this approximation is therefore that the dynamics is driven by drift, and not mutation, over most of the range of x , and that the role of mutations is to set the normalisation of Eq. (60) via the influence of boundary effects at x close to 0 and 1. This interpretation is described in the context of the Moran model with mutations and selection by Vogl and Bergman (2015) who introduce the terminology *boundary-mutation model* to refer to the low mutation rate approximation.

The Continuum Wright-Fisher Model With Mutations and Selection

Analogous to the continuous time mutation rates Eq. (55), we define a continuous-time selection coefficient

$$\gamma = Ms, \quad (68)$$

where s was defined by Eqs. (40) and (47) for diploid and haploid populations respectively. A calculation analogous to Eq. (57) using the transition matrix for WF with selection, Eqs. (43), (46) and (49), yields the first order derivative coefficient

$$a(x) = \begin{cases} \gamma x(1-x)[x + h(1-2x)] & \text{(diploid),} \\ \frac{1}{2}\gamma x(1-x) & \text{(haploid),} \end{cases} \quad (69)$$

in the forward Kolmogorov equation, while the second order derivative coefficient $b(x)$ in Eq. (59) remains unchanged.

Note that forward Kolmogorov equation for the diploid case of no-dominance, $h = \frac{1}{2}$, is identical to haploid equation. If both mutations and selection are included, the no-dominance diploid or haploid forward Kolmogorov equation is

$$\begin{aligned} \frac{\partial f(x; p, t)}{\partial t} = & -\frac{\partial}{\partial x} \left[\left(\frac{1}{2}\gamma x(1-x) - q_{12}x + q_{21}(1-x) \right) f(x; p, t) \right] \\ & + \frac{1}{2} \frac{\partial^2}{\partial x^2} [x(1-x)f(x; p, t)]. \end{aligned} \quad (70)$$

The stationary solution to this equation is

$$f(x) = Cx^{2q_{21}-1}(1-x)^{2q_{12}-1}e^{\gamma x}, \quad (71)$$

where the constant C is chosen to ensure $\int_0^1 f(x) dx = 1$.

A complete time dependent solution to Eq. (70) for the case of selection but no mutation (i.e. $q_{12}=q_{21}=0$) is derived in Section 8.6 of Crow and Kimura (1970). The solution is highly complex, however it is relatively easy to derive the result that, given an initial relative frequency p , allele A_1 will fix with probability (Waxman, 2011; Ewens, 2004, Sections 4.3 and 5.3)

$$\text{Prob}(X(\infty) = 1 | X(0) = p) = \frac{1 - e^{-\gamma p}}{1 - e^{-\gamma}}. \quad (72)$$

Estimation of Mutation Rates

We return now to the stationary solution to the diffusion limit WF model with neutral mutations, namely Eq. (61), and address the problem of estimating mutation rates from sampled SNP data. In the following it will prove useful to use an alternate parameterisation similar to that used by Vogl (2014), and set

$$\theta = q_{12} + q_{21}, \quad \alpha = \frac{q_{12}}{q_{12} + q_{21}}. \quad (73)$$

The parameter θ is the scaled total mutation rate and the parameter α is a measure of the relative bias between A_1 to A_2 mutations and A_2 to A_1 mutations. In terms of the newly defined parameters, Eq. (61) becomes

$$f(x) = \frac{1}{B(2(1-\alpha)\theta, 2\alpha\theta)} x^{2(1-\alpha)\theta-1} (1-x)^{2\alpha\theta-1}, \quad 0 \leq x \leq 1. \quad (74)$$

Generally one expects an effective population to be considerably larger than that of the toy examples considered in the numerical simulations Figs. 2 to 5, and if stationarity is assumed it should be possible in principle to estimate the continuum limit parameters such as θ and α from an observed SFS. However it is important to realise that the distribution $f(x)$ of the SFS is not observed directly, simply because a census of the genomic makeup an entire population is not practical. Instead, biologists will sequence the genomes of a sample of the population, which is ideally randomly chosen.

Maximum Likelihood Estimator

Suppose we assume a sample of n_{sample} individual chromosomes across n_{site} independent genomic sites, each assumed chosen to be a site not susceptible to selective pressures. We also assume the genome to have evolved so as to represent the stationary state of the simplified model in which the genome consists only of letters A_1 and A_2 from a 2-letter alphabet, and that all the assumptions leading to the neutral WF model with two-way mutations hold.

In this case the data will consist of a set of i.i.d. random numbers $U_1, U_2, \dots, U_{n_{\text{site}}}$ of type A_1 alleles, each taking a value in the set $\{0, 1, \dots, n_{\text{sample}}\}$, observed at the n_{site} genomic sites. At any given site j , conditional on an (unobserved) underlying population fraction x of A_1 -type alleles at that site, each U_j will be a binomial random variable:

$$U_j | (X = x) \sim \text{bin}(n_{\text{sample}}, x), \quad (75)$$

where X has the beta distribution, Eq. (74). Thus the unconditional distribution of each U_j is

$$\begin{aligned}
 \text{Prob}(U_j = u) &= \int_0^1 \text{Prob}(U_j = u|X = x)f(x) dx \\
 &= \int_0^1 \binom{n_{\text{sample}}}{u} x^u (1-x)^{n_{\text{sample}}-u} \times \frac{x^{2(1-\alpha)\theta-1} (1-x)^{2\alpha\theta-1}}{B(2(1-\alpha)\theta, 2\alpha\theta)} dx \\
 &= \binom{n_{\text{sample}}}{u} \frac{1}{B(2(1-\alpha)\theta, 2\alpha\theta)} \\
 &\quad \times \int_0^1 x^{u+2(1-\alpha)\theta-1} (1-x)^{n_{\text{sample}}-u+2\alpha\theta-1} dx \\
 &= \binom{n_{\text{sample}}}{u} \frac{B(u+2(1-\alpha)\theta, n_{\text{sample}}-u+2\alpha\theta)}{B(2(1-\alpha)\theta, 2\alpha\theta)},
 \end{aligned} \tag{76}$$

for $u=0, \dots, n_{\text{sample}}$. Eq. (76) is the probability function of the beta-binomial distribution (Johnson *et al.*, 1993). This is a 3-parameter family of discrete distributions whose generic probability mass function takes the form

$$P_U(u; n, a, b) = \binom{n}{u} \frac{B(u+a, n-u+b)}{B(a, b)}, \quad u = 0, \dots, n, \tag{77}$$

where n is a non-negative integer and a and b are positive real numbers.

Given the dataset described above, the parameters θ and α can be estimated numerically via a maximum likelihood calculation. In this case one would maximize the log likelihood

$$L(\theta, \alpha | u_1, \dots, u_{n_{\text{site}}}) = \sum_{j=1}^{n_{\text{site}}} \log P_U(u_j; n_{\text{sample}}, 2(1-\alpha)\theta, 2\alpha\theta), \tag{78}$$

with respect to θ and α , where $u_1, \dots, u_{n_{\text{site}}}$ are the observed counts of type- A_1 alleles at each site. Vogl (2014) has developed an expectation-maximisation algorithm for performing this maximisation.

Relationship to the Watterson Estimator

A commonly encountered estimator of the scaled total mutation rate θ , used when $\theta \ll 1$ and $\alpha = \frac{1}{2}$, is the *Watterson estimator* (Watterson, 1975). In the language of our current analysis it states that, if n_{sample} genomes are sampled, and the proportion of sites which are observed to be segregating is S , then

$$\hat{\theta}_W = \frac{S}{\sum_{i=1}^{n_{\text{sample}}-1} (1/i)}, \tag{79}$$

is an unbiased estimate of θ . A site is said to be segregating if more than one allele is observed at that site (i.e. it is observed to be a SNP on the basis of a sample of size n_{sample}). An demonstration that the Watterson estimator is unbiased will be given from the point of view of the coalescent in the section The Kingman Coalescent and the Watterson Estimator.

By carrying out an expansion of the beta-binomial distribution in the small parameter θ , Vogl (2014) has obtained maximum likelihood estimators which generalise Watterson's result to the case $\alpha \neq \frac{1}{2}$. Vogl's estimators are

$$\hat{\theta}_V = \frac{S}{\sum_{i=1}^{n_{\text{sample}}-1} (1/i)}, \quad \hat{\alpha}_V = S_2 + \frac{1}{2}S, \tag{80}$$

where S is the proportion of sites that are segregating and S_2 is the proportion of sites that are non-segregating with the observed allele being A_2 . Here $\hat{\theta}_V$ is an unbiased, maximum likelihood estimator of the quantity

$$\vartheta = 4\alpha(1-\alpha)\theta, \tag{81}$$

which reduces to θ when $\alpha = \frac{1}{2}$.

Below we give a brief proof that $\hat{\theta}_V$ is unbiased. Note first that S is an unbiased estimator of the probability that a site chosen at random from a very large number of independent sites is observed to be segregating. Equivalently, $E[S]$ is the probability that the random variable U_j in the distribution Eq. (76) is neither 1 or n_{sample} . Thus to demonstrate that $\hat{\theta}_V$ is indeed an unbiased estimator of ϑ it suffices to show that

$$1 - \text{Prob}(U_j = 0) - \text{Prob}(U_j = n_{\text{sample}}) = 4\alpha(1-\alpha)\theta \sum_{i=1}^{n_{\text{sample}}-1} \frac{1}{i}, \quad \text{as } \theta \rightarrow 0, \tag{82}$$

where U_j has the beta-binomial distribution Eq. (76).

To obtain the small θ approximation, we start with some properties of the beta and related functions (Abramowitz *et al.*, 1972). The beta function can be written in terms of gamma functions as

$$B(z_1, z_2) = \frac{\Gamma(z_1)\Gamma(z_2)}{\Gamma(z_1 + z_2)}. \quad (83)$$

For ε small and $n=1, 2, \dots$,

$$\Gamma(\varepsilon) = \frac{1}{\varepsilon} - \gamma + O(\varepsilon), \quad \Gamma(n + \varepsilon) = (1 + \varepsilon\psi(n))\Gamma(n) + O(\varepsilon^2), \quad (84)$$

where γ is Euler's constant and $\psi(z) = d\log\Gamma(z)/dz = \Gamma'(z)/\Gamma(z)$ is called the digamma function. It satisfies

$$\psi(1) = -\gamma, \quad \psi(n) = -\gamma + \sum_{i=1}^{n-1} \frac{1}{i}, \quad n = 1, 2, \dots \quad (85)$$

Armed with these properties, we have from Eq. (76)

$$\begin{aligned} \text{Prob}(U_j = 0) &= \binom{n_{\text{sample}}}{0} \frac{B(2(1-\alpha)\theta, n_{\text{sample}} + 2\alpha\theta)}{B(2(1-\alpha)\theta, 2\alpha\theta)} \\ &= \frac{\Gamma(n_{\text{sample}} + 2\alpha\theta)}{\Gamma(n_{\text{sample}} + 2\theta)} \cdot \frac{\Gamma(2\theta)}{\Gamma(2\alpha\theta)} \\ &= \frac{1 + 2\alpha\theta\psi(n_{\text{sample}})}{1 + 2\theta\psi(n_{\text{sample}})} \cdot \frac{1/(2\theta) - \gamma}{1/(2\alpha\theta) - \gamma} + O(\theta^2) \\ &= \alpha[1 - 2(1-\alpha)\theta\psi(n_{\text{sample}}) - 2(1-\alpha)\theta\gamma] + O(\theta^2) \\ &= \alpha - 2\alpha(1-\alpha)\theta \sum_{i=1}^{n_{\text{sample}}-1} \frac{1}{i} + O(\theta^2). \end{aligned} \quad (86)$$

Similarly,

$$\text{Prob}(U_j = n_{\text{sample}}) = 1 - \alpha - 2\alpha(1-\alpha)\theta \sum_{i=1}^{n_{\text{sample}}-1} \frac{1}{i} + O(\theta^2), \quad (87)$$

and Eq. (82) follows.

Multiple Alleles

We now consider extending the number of alleles from two to any number K . For instance, at any genomic site, the possible nucleotides are $\{A, C, G, T\}$, giving $K=4$ possible alleles.

In general we will label the alleles $A_1, \dots, A_K$, and consider a model with a fixed population size of M haploid or $M=2N$ diploid individuals. Define the random variables $Y_a(\tau)$ to be the number individuals of type A_a in the population at birth at the discrete times $\tau=0, 1, 2, \dots$, with the constraint that $\sum_{a=1}^K Y_a(\tau) = M$. The transition matrix Eq. (38) of the 2-allele WF model generalises to a matrix of probabilities of transitioning from a set of allele frequencies $\mathbf{i} = (i_1, \dots, i_K)$ to a set of allele frequencies $\mathbf{j} = (j_1, \dots, j_K)$ given by the multinomial distribution

$$\begin{aligned} p_{\mathbf{j}} &= \text{Prob}(\mathbf{Y}(\tau+1) = \mathbf{j} | \mathbf{Y}(\tau) = \mathbf{i}) \\ &= \begin{cases} \frac{M!}{\prod_{a=1}^K j_a!} \prod_{a=1}^K \psi_a(\mathbf{i})^{j_a} & \text{if } \sum_{a=1}^K i_a = \sum_{a=1}^K j_a = M, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (88)$$

where $\psi_a(\mathbf{i})$ is the probability of an individual inheriting allele type A_a given the previous generation's initial allele frequencies $\mathbf{i}$. This transition matrix defines a finite state Markov chain with a state space of dimension $\binom{M+K-1}{K-1}$, corresponding to the sites of a simplicial lattice in K dimensions. An example of the lattice is shown in Fig. 6 for the case $K=4$ and $M=30$.

Consider now the neutral WF model with K alleles. Suppose that u_{ab} , where $1 \leq a, b \leq K$, $u_{ab} \geq 0$ and $\sum_{b=1}^K u_{ab} = 1$, is the probability of an individual mutating from allele type A_a to type A_b in a single timestep. Generalising Eqs. (36) and (37) we have

$$\psi_a(\mathbf{i}) = \sum_{b=1}^K \frac{i_b}{M} u_{ba}. \quad (89)$$

The diffusion limit is obtained by defining random variables $X_a(t) = Y_a(\tau)/M$ equal to the relative proportion of type- A_a alleles within the population at continuous time $t = \tau/M$. The limit $M \rightarrow \infty$ and $u_{ab} \rightarrow 0$ for $a \neq b$ is taken in such a way that the $K \times K$

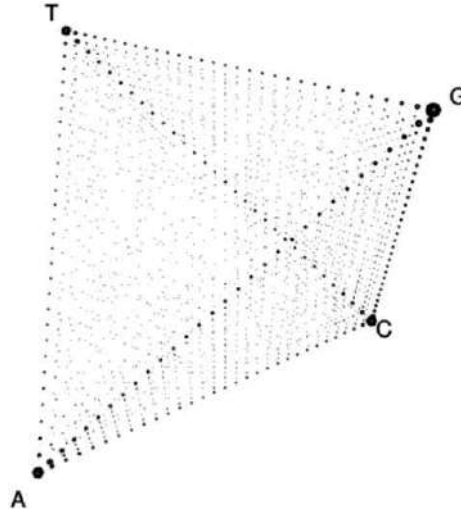


Fig. 6 Stationary distribution of allele frequencies for the multi-allele neutral WF model defined by Eqs. (88) and (89) for a haploid population of size $M=30$ and scaled rate matrix Q (see Eq. (90)) given by Eq. (95). The corners labelled A, C, G and T correspond to allele frequencies $\mathbf{i} = (N, 0, 0, 0)$, $(0, N, 0, 0)$, $(0, 0, N, 0)$ and $(0, 0, 0, N)$ respectively, and the volume of the sphere at each coordinate point is proportional to the probability mass function. Reproduced from Burden, C.J., Tang, Y., 2016. An approximate stationary solution for multi-allele neutral diffusion with low mutation rates. *Theoretical Population Biology* 112, 22–32.

instantaneous rate matrix Q , whose elements are defined by

$$q_{ab} = M(u_{ab} - \delta_{ab}), \quad (90)$$

remains finite. Here δ_{ab} is the Kronecker delta, equal to 1 if $a=b$ and 0 otherwise. Note that Eq. (90) is the generalisation of Eq. (55), and that Q is an instantaneous rate matrix (Isaev, 2004, Section 5.4) with the properties

$$q_{ab} \geq 0 \text{ for } a \neq b, \quad \sum_{b=1}^K q_{ab} = 0. \quad (91)$$

Using an argument similar to that illustrated by Fig. 3, the rate matrix Q can be identified as the whole-population substitution matrix relative to the continuum time t .

The diffusion limit gives the forward Kolmogorov equation

$$\frac{\partial f}{\partial t} = - \sum_{a=1}^{K-1} \frac{\partial}{\partial x_a} \sum_{b=1}^K x_b q_{ba} f + \frac{1}{2} \sum_{a,b=1}^{K-1} \frac{\partial^2}{\partial x_a \partial x_b} \{ (\delta_{ab} x_a - x_a x_b) f \}, \quad (92)$$

for the density function $f(x_1, \dots, x_{K-1}; t)$ of the vector of continuous random variables $X_1(t), \dots, X_{K-1}(t)$. The function f is defined over the simplex

$$\mathcal{S} = \left\{ (x_1, \dots, x_{K-1}) : x_1, \dots, x_{K-1} \geq 0, \sum_{a=1}^{K-1} x_a \leq 1 \right\}. \quad (93)$$

For notational convenience, we have defined $x_K = 1 - \sum_{a=1}^{K-1} x_a$ in Eq. (92), and also in Eq. (94) below. For details of the derivation of Eq. (92) see Lemma 4.1 of Etheridge (2011) or Eq. (5.125) of Ewens (2004).

Stationary Distribution of the Multi-Allele Neutral WF Model

The stationary distribution $f(x_1, \dots, x_{K-1})$ to the forward Kolmogorov equation is obtained by setting $\partial f / \partial t$ to zero on the left hand side of Eq. (92). The problem of finding an analytic stationary solution for an arbitrary rate matrix Q is as yet unsolved, though an exact analytic solution is known for a special case of parent-independent rate matrices in which the elements of the rate matrix take the form $q_{ab} = q_b$ (independent of a). In this case the solution is the multi-dimensional generalisation of the beta distribution, Eq. (61), namely the Dirichlet distribution (Wright, 1969; Tier and Keller, 1978; Griffiths, 1979):

$$f(x_1, \dots, x_{K-1}) = \frac{\Gamma(2 \sum_{a=1}^K q_a)}{\prod_{a=1}^K \Gamma(2q_a)} \prod_{a=1}^K x_a^{2q_a-1}. \quad (94)$$

Regrettably, there is no biological reason to consider parent-independent rate matrices, and in fact, most realistic models in phylogenetics are based on rate matrices such as the Hasegawa-Kishino-Yano (HKY) rate matrix (Hasegawa *et al.*, 1985), which are

not parent-independent. On the other hand, an accurate approximate solution is known in the biologically relevant limit of low scaled mutation rates, $q_{ab} \ll 1$ where $a \neq b$, for otherwise arbitrary rate matrices (Burden and Tang, 2016).

To understand the idea behind Burden and Tang's solution, consider for instance the numerical stationary solution to the discrete Wright-Fisher defined by Eqs. (88) to (90) shown in Fig. 6. For the purposes of illustration the solution has been simulated using an HKY matrix

$$Q = \begin{pmatrix} -0.11 & 0.03 & 0.06 & 0.02 \\ 0.02 & -0.09 & 0.03 & 0.04 \\ 0.04 & 0.03 & -0.09 & 0.02 \\ 0.02 & 0.06 & 0.03 & -0.11 \end{pmatrix} \quad (95)$$

and a small population $M=30$ in order to render the simulation numerically tractable. The mutation rates are unrealistically high by at least two orders of magnitude to enable the distribution to be visible over the entire simplex on the scale of the plot. Nevertheless, the distribution is dominated by the corners of the tetrahedron, indicating that the majority of genomic sites are not SNPs. Most of the remaining support of the distribution lies on the edges of the tetrahedron, which correspond to 2-allele SNPs. The interiors of the four faces, corresponding to 3-allele SNPs, and the interior volume of the tetrahedron, corresponding to 4-allele SNPs, account for only a small fraction of the total probability. This distribution is consistent with the observed occurrence of SNP multiplicities described at the beginning of the Introduction.

Exploiting this property of the low mutation rate limit, one can argue that the stationary distribution is concentrated close to the edges of the simplex S and can be represented accurately as a set of line densities defined along those edges. Suppose we label the corners of S by the allele prevalent within the population at that corner, so the corner A_a corresponds to the co-ordinate $(x_1, \dots, x_K) = e_a$, where e_a is the K -dimensional vector with 1 in the a th position and zeroes elsewhere, as illustrated in Fig. 6 for the 4-letter genomic alphabet. On the edge joining corner A_a to corner A_b a line density $f_{ab}(x)$ is defined for each pair of indices a and b , with the convention adopted that the argument x is the relative proportion of type- a alleles, and $1-x$ is the relative proportion of type- b alleles. The relative proportion of the remaining $K-2$ alleles along this edge is taken to be zero.

The line densities $f_{ab}(x)$ are solutions of an effective 2-allele model and satisfy a stationary forward Kolmogorov equation of the form of Eq. (60) with q_{12} and q_{21} replaced by q_{ab} and q_{ba} respectively. However, there is one major difference, namely that the constant on the right hand side can be a non-zero flux of probability Φ_{ab} flowing through the corners of the simplex S . Burden and Tang show that the presence of this flux modifies the approximate solution Eq. (65) to the form

$$f_{ab}(x) \approx C_{ab} \left(\frac{1}{x} + \frac{1}{1-x} \right) - \Phi_{ab} \left(\frac{1}{x} - \frac{1}{1-x} \right), \quad (96)$$

where the constants C_{ab} and Φ_{ab} are determined from rate matrix Q via the formulae

$$C_{ab} = \pi_a q_{ab} + \pi_b q_{ba}, \quad \Phi_{ab} = \pi_a q_{ab} - \pi_b q_{ba}. \quad (97)$$

Here $\pi^T = (\pi_1 \dots \pi_K)$ is the left eigenvector of Q satisfying

$$\pi_a \geq 0, \quad \sum_{a=1}^K \pi_a = 1, \quad \sum_{a=1}^K \pi_a q_{ab} = 0. \quad (98)$$

A sufficient condition for a unique π^T to exist is that $q_{ab} > 0$ for all $a \neq b$, which is expected to include any biologically realistic model. For computational reasons it is common in the phylogenetics literature to assume models in which rate matrices are reversible, that is, matrices satisfying the constraint $\pi_a q_{ab} = \pi_b q_{ba}$ (Lanave et al., 1984; Tavaré, 1986). We note that the introduction of non-zero values of the fluxes Φ_{ab} is equivalent to invoking non-reversible rate matrices.

The approximation Eq. (96) loses accuracy as the non-integrable singularities at $x=0$ and 1 are approached, that is, in the vicinity of the corners of S . However, one can show that for a population of size M , the value of the stationary distribution at the corners of the simplex corresponding to the discrete problem defined by Eqs. (88) and (89) is approximately

$$P(Y(\infty) = e_a) \approx \pi_a - \sum_{b \neq a} C_{ab} \log M. \quad (99)$$

The approximate stationary solution defined by Eqs. (96) and (99) reduces to the $K=2$ -allele approximate solution, Eqs. (65) and (66) on observing that, for a 2×2 rate matrix Q ,

$$(\pi_1, \pi_2) = \frac{1}{q_{12} + q_{21}} (q_{21}, q_{12}), \quad C_{12} = C_{21} = \frac{2q_{12}q_{21}}{q_{12} + q_{21}}, \quad \Phi_{12} = \Phi_{21} = 0. \quad (100)$$

Fig. 7 shows the numerically determined stationary solution for a population of size $M=30$ and the rate 4×4 matrix

$$Q^{\text{GRM}} = \begin{pmatrix} -5.375 & 3.125 & 2.125 & 0.1250 \\ 0.833 & -17.500 & 0.583 & 16.083 \\ 15.000 & 0.500 & -21.000 & 5.500 \\ 4.600 & 15.900 & 0.100 & -20.600 \end{pmatrix} \times 10^{-4}, \quad (101)$$

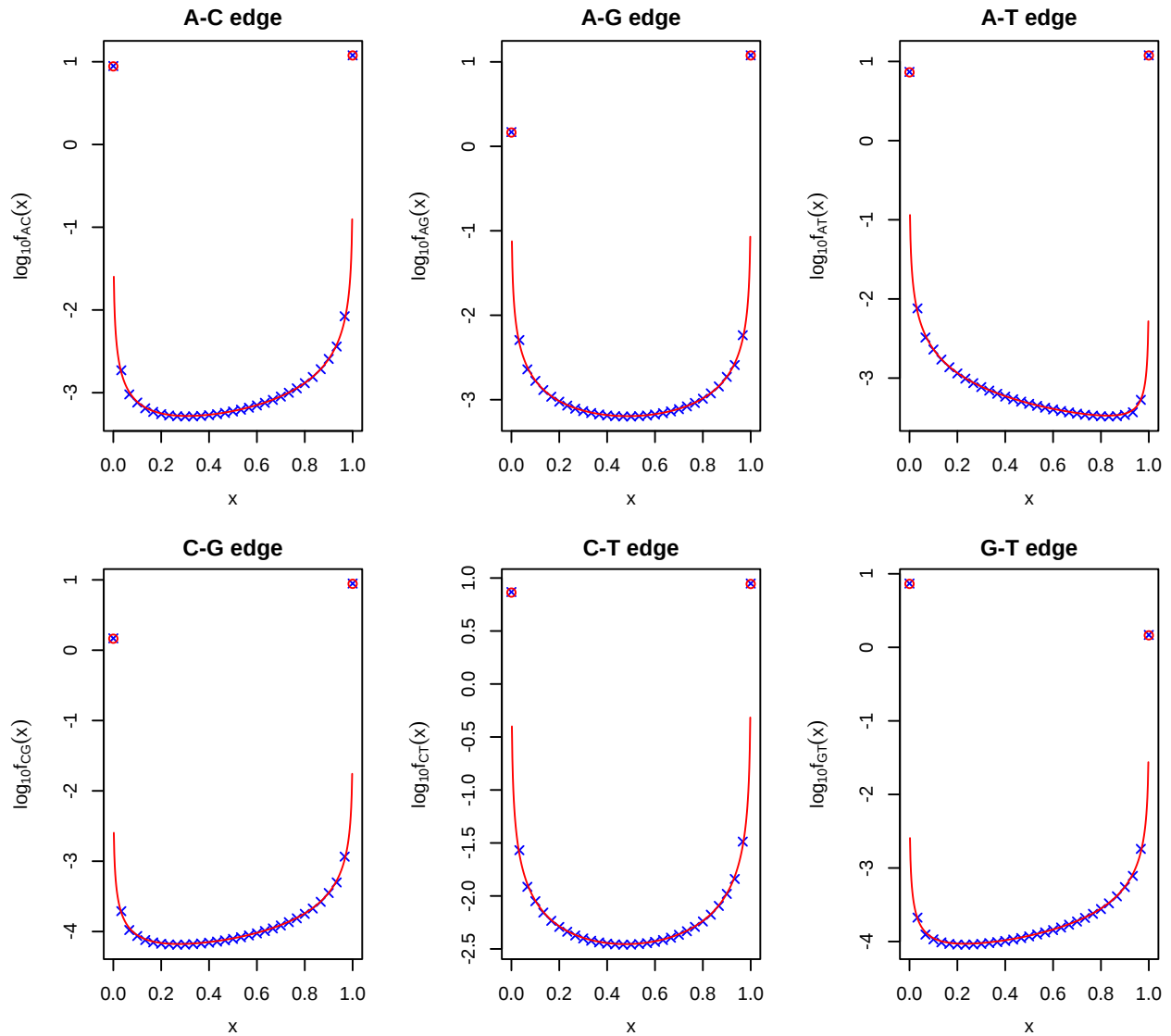


Fig. 7 Simulation of the neutral Wright-Fisher model with $K=4$ alleles and a population size $M=30$ and the rate matrix Q defined by Eq. (101). Blue crosses are the numerically determined stationary distribution along each edge of the simplex over which the site frequency distribution is defined. For any 2 alleles, A_a and A_b , the parameter x is the relative proportion of A_a alleles and $1-x$ is the relative proportion of A_b alleles. Superimposed in red are the theoretical line densities $f_{ab}(x)$, Eq. (96) plotted on a logarithmic scale. The red circles are the theoretical probabilities at the simplex corners, Eq. (99). The probabilities of the stationary distribution and the probabilities in Eq. (99) have been multiplied by a factor M for comparison with the continuum line densities.

plotted along each of the six edges of the simplex S , together with the approximate solution Eqs. (96) and (99). This matrix is taken from Eq. (53) of Burden and Tang (2017), and was chosen reproduce approximately the time-reversible matrix in Table 5 of Lanave *et al.* (1984) estimated from the rat-mouse phylogeny, with an arbitrary non-reversible part added.

Zeng (2010) has exploited the fact that stationary distributions of multi-allelic WF models are concentrated on the corners and edges of the simplex S to demonstrate that it is feasible to estimate all parameters of an evolutionary rate matrix Q from site-frequency data via numerical solution of the multi-allelic discrete WF model. Burden and Tang (2017) have extended the procedure to take advantage of the approximate analytic solution described above. Zeng's approach has the advantage the he is able to estimate selection parameters as well as mutation rates. On the other hand, Burden and Tang's approach has the advantages that the likelihood function takes a relatively simple analytic form entailing efficient numerical calculation for a given observed site-frequency data set, and that it provides a statistical test of the hypothesis, commonly assumed without biological justification, that the rate matrix is reversible.

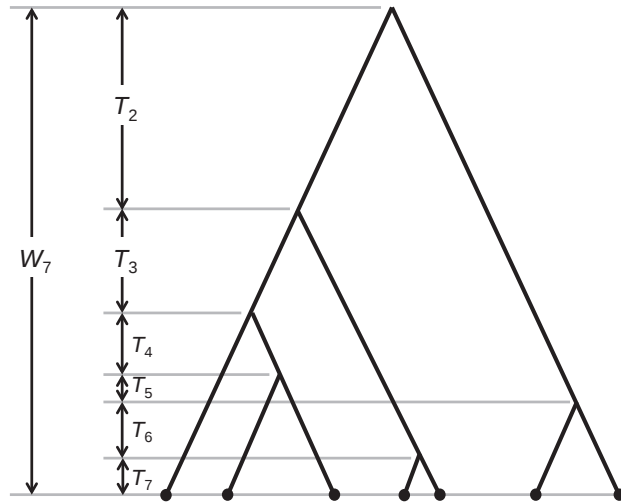


Fig. 8 A coalescent tree of $n_{\text{sample}}=7$ individuals.

The Kingman Coalescent and the Watterson Estimator

The *coalescent* of a sample of individuals in a population is the ancestral lineage of the sample traced backwards in time to their most recent common ancestor. Use of the coalescent as a tool of analysis in population genetics was developed by a number of authors in the 1970s and early 1980s, though the definitive treatment of the subject is due to [Kingman \(1982, 2000\)](#). Here we give a brief description of the coalescent in the context of the haploid Wright-Fisher model, making use primarily of the property that each individual is the offspring of one randomly and independently chosen parent from the previous generation.

Define τ_C to be the number of generations counted backwards in time to the most recent common ancestor of two randomly chosen members of a population of size M . Clearly we have that $\text{Prob}(\tau_C = 1) = 1/M$ and $\text{Prob}(\tau_C \neq 1) = 1 - 1/M$. Given that generations are independent, the event " $\tau_C = \tau$ " corresponds to $\tau - 1$ generations for which the ancestral line traced backwards does not coalesce, followed by one generation for which the ancestral line does coalesce, thus

$$\text{Prob}(\tau_C = \tau) = \left(1 - \frac{1}{M}\right)^{\tau-1} \frac{1}{M}. \quad (102)$$

Now consider the continuum limit $M \rightarrow \infty$ defined with a continuous time $t = \tau/M$, $\delta t = 1/M$, which proved useful in calculations of the fixation time and for deriving the forward Kolmogorov Equation. Defining the continuous time to coalescence of a sample of size two as $T_2 = \tau_C/M$, [Eq. \(102\)](#) implies

$$\text{Prob}(t \leq T_2 < t + \delta t) = e^{-t} \delta t + O(\delta t^2). \quad (103)$$

Thus T_2 is an exponential random variable with rate 1. Moreover, we have that the probability that the next coalescent event backwards in time between two given individuals whose ancestry has not coalesced is

$$\text{Prob}(t \leq T_2 < t + \delta t | T_2 > t) = \delta t + O(\delta t^2). \quad (104)$$

Next we consider the coalescent of a sample of n_{sample} randomly chosen individuals, as shown in [Fig. 8](#). Define T_j to be the time interval between the coalescence event which reduces the ancestry to j individuals, and the coalescence event which reduces the ancestry to $j - 1$ individuals. At a point when the ancestry consists of j individuals there are $j(j-1)/2$ possibilities for the next pairwise coalescent event backwards in time. Thus from [Eq. \(104\)](#),

$$\text{Prob}(t \leq T_j < t + \delta t | T_j > t) = \frac{j(j-1)}{2} \delta t + O(\delta t^2), \quad (105)$$

which implies that the T_j are independent exponential random variables with respective rates $j(j-1)/2$, and that

$$E(T_j) = \frac{2}{j(j-1)}, \quad \text{Var}(T_j) = \frac{4}{j^2(j-1)^2}. \quad (106)$$

Set

$$W_{n_{\text{sample}}} = \sum_{j=2}^{n_{\text{sample}}} T_j, \quad (107)$$

to be the time back to the most recent common ancestor of the entire sample. The mean and variance of $W_{n_{\text{sample}}}$ are

$$\begin{aligned}
 E(W_{n_{\text{sample}}}) &= 2 \sum_{j=2}^{n_{\text{sample}}} \frac{1}{j(j-1)} \\
 &= 2 \sum_{j=2}^{n_{\text{sample}}} \left(\frac{1}{j-1} - \frac{1}{j} \right) \\
 &= 2 \left(1 - \frac{1}{n_{\text{sample}}} \right) \\
 &= 2 + O\left(n_{\text{sample}}^{-1}\right), \quad \text{as } n_{\text{sample}} \rightarrow \infty,
 \end{aligned} \tag{108}$$

and

$$\begin{aligned}
 \text{Var}(W_{n_{\text{sample}}}) &= 4 \sum_{j=2}^{n_{\text{sample}}} \left(\frac{1}{j-1} - \frac{1}{j} \right)^2 \\
 &= 8 \sum_{j=1}^{n_{\text{sample}}-1} \frac{1}{j^2} - 12 + \frac{8}{n_{\text{sample}}} + \frac{4}{n_{\text{sample}}^2} \\
 &= \frac{4\pi^2}{3} - 12 + O\left(n_{\text{sample}}^{-1}\right), \quad \text{as } n_{\text{sample}} \rightarrow \infty.
 \end{aligned} \tag{109}$$

Note that half of the contribution to $E(W_{n_{\text{sample}}})$ comes from the time taken for the coalescence of the earliest two ancestors, namely T_2 . To put it another way, the longest arms of the tree tend to be those for which there is a small number of individuals in the ancestry. Furthermore, as $n_{\text{sample}} \rightarrow \infty$ the expected coalescence time of the entire population is $2M(1 + O(1/M))$ generations, which is consistent with the time $\bar{\tau}^*(1)$ obtained in the section Expected time to Fixation for a mutation in a single individual to fix throughout the entire population.

Recall the Watterson estimator $\hat{\theta}_W$, Eq. (79), for the total mutation rate θ in neutral evolution, calculated in terms of the proportion S of segregating sites in the genomes of a sample of individuals. The total length of the branches of the coalescent tree can be used to show that $\hat{\theta}_W$ is an unbiased estimator of θ as follows. From Fig. 8, the total tree length is

$$L = \sum_{j=2}^{n_{\text{sample}}} jT_j. \tag{110}$$

If mutations are rare over the timescale of the figure and $\frac{1}{2}\theta$ is the average mutation rate per site (see Eq. (73)), then the proportion of segregating sites, conditional on the total length L of the tree, is a Poisson random variable with mean $\frac{1}{2}\theta L$. Then using the law of total expectation,

$$\begin{aligned}
 E(S) &= E(E(S|L)) \\
 &= \frac{1}{2}\theta E(L) \\
 &= \frac{1}{2}\theta \sum_{j=2}^{n_{\text{sample}}} jE(T_j) \\
 &= \theta \sum_{j=1}^{n_{\text{sample}}-1} \frac{1}{j}.
 \end{aligned} \tag{111}$$

It is then immediately obvious from Eq. (79) that $E(\hat{\theta}_W) = \theta$ and so $\hat{\theta}_W$ is an unbiased estimate of θ .

The Coalescent for Varying Populations

Slatkin and Hudson (1991) have generalised the coalescent for $n_{\text{sample}}=2$ individuals to the case of time-varying population sizes. Suppose we have a deterministically imposed varying population of size $M(\tau)$ where τ is the number of generations counted backwards in time from the present population, $M(0)=m_0$. Eq. (102) generalises to

$$\text{Prob}(\tau_C = \tau) = \prod_{\ell=1}^{\tau-1} \left(1 - \frac{1}{M(\ell)} \right) \frac{1}{M(\tau)}. \tag{112}$$

We define a continuous-time limit by setting

$$t = \frac{\tau}{m_0}, \quad \delta t = \frac{1}{m_0}, \quad T_2 = \frac{\tau_C}{m_0}, \quad v(t) = \frac{M(\tau)}{m_0}, \tag{113}$$

and taking the limit $m_0 \rightarrow \infty$. Rearranging Eq. (112) we have

$$\begin{aligned} \log(M(\tau)\text{Prob}(\tau_c = \tau)) &= \sum_{\ell=1}^{\tau-1} \log\left(1 - \frac{1}{M(\ell)}\right) \\ &= -\sum_{\ell=1}^{\tau-1} \left[\frac{1}{m_0 v(m_0 \ell)} + O\left(\frac{1}{m_0^2}\right) \right]. \end{aligned} \quad (114)$$

Thus

$$\log\left(\frac{v(t)}{\delta t} \text{Prob}(t \leq T_2 < t + \delta t)\right) = -\int_0^t \frac{du}{v(u)} + O(\delta t), \quad (115)$$

which rearranges to

$$\text{Prob}(t \leq T_2 < t + \delta t) = \frac{1}{v(t)} \exp\left(-\int_0^t \frac{du}{v(u)}\right) \delta t + O(\delta t^2). \quad (116)$$

The density function for the coalescent time T_2 is then

$$f_{T_2}(t) = \frac{1}{v(t)} \exp\left(-\int_0^t \frac{du}{v(u)}\right). \quad (117)$$

For the case of exponential population growth,

$$v(t) = e^{-\alpha t}, \quad (118)$$

one easily obtains

$$f_{T_2}(t) = e^{\alpha t} \exp\left(\frac{1 - e^{\alpha t}}{\alpha}\right), \quad 0 \leq t < \infty. \quad (119)$$

The mean time to coalescence is

$$E(T_2) = \frac{e^{1/\alpha}}{\alpha} E_1\left(\frac{1}{\alpha}\right), \quad (120)$$

where $E_1(z) = \int_z^\infty e^{-t}/t dt$ is the exponential integral (Abramowitz *et al.*, 1972, Eq. (5.5.1)). Slatkin and Hudson (1991) note that for $\alpha \gg 1$, the distribution is narrow and concentrated around a mean $E(T_2) \approx (\log \alpha - \gamma)e^{1/\alpha}/\alpha$ where $\gamma \approx 0.577$ is the Euler-Mascheroni constant.

Note that for a time-varying population the problem of determining the coalescent for $n_{\text{sample}} \geq 3$ individuals is nontrivial as the inter-coalescence times T_j illustrated in Fig. 8 are not mutually independent: There is no simple analog of Eq. (105) as restarting the clock at the last coalescence event requires knowing the population size at that event, which in turn depends on $T_{j+1} + \dots + T_{n_{\text{sample}}}$. For further analysis of this problem, including an explicit formula for the joint distribution of $T_2, \dots, T_{n_{\text{sample}}}$ see Griffiths and Tavaré (1994) and Chen and Chen (2013).

Alternatives to Wright-Fisher

So far we have concentrated only on the Wright-Fisher model. Below is a survey of some of the more important alternative models of population genetics.

The Moran Model

In common with the WF model, the *Moran model* (Moran, 1958) assumes a fixed population size and discrete time steps $\tau = 0, 1, 2, \dots$. The main difference, however, is that in the Moran model generations are allowed to overlap.

Consider the simplest form of the model, namely a haploid population of size M , each individual carrying one of two alleles, A_1 and A_2 , which are inherited without mutations. At each time step one randomly chosen member of the population reproduces, with the offspring carrying the same allele as the parent, and one randomly chosen individual, possibly the same individual, dies. With the convention that the number of individuals carrying allele A_1 at time step τ is $Y(\tau)$ the model defines a finite state Markov chain with transition probabilities

$$\begin{aligned} p_{ij} &= \text{Prob}(Y(\tau+1) = j | Y(\tau) = i) \\ &= \begin{cases} \frac{i(M-i)}{M^2} & \text{if } j = i \pm 1, \\ \frac{i^2 + (M-i)^2}{M^2} & \text{if } j = i, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (121)$$

for $i, j=0, \dots, M$. As for the WF model, it is easy to check that Eq. (3) holds, so that $Y(\tau)=0$ and $Y(\tau)=M$ are absorbing states of the Markov chain. Also Eqs. (8) and (9) hold, and so if $Y(0)=i$, either A_1 or A_2 must fix almost surely, with probabilities i/M and $1-i/M$ respectively.

The most straightforward way to include mutations dynamically in Eq. (121) is through the *decoupled Moran model*, which was introduced by Baake and Bialowons (2008) and Etheridge and Griffiths (2009) for the more general K -allele case. For 2 alleles the model is defined as

$$p_{ij} = \begin{cases} \frac{i(M-i)}{M^2} + u_{12} \frac{i}{M} & \text{if } j = i-1, \\ \frac{i^2 + (M-i)^2}{M^2} - u_{12} \frac{i}{M} - u_{21} \frac{M-i}{M} & \text{if } j = i, \\ \frac{i(M-i)}{M^2} + u_{21} \frac{M-i}{M} & \text{if } j = i+1, \\ 0 & \text{otherwise} \end{cases} \quad (122)$$

where u_{12} and u_{21} are mutation rates per unit of the time-step τ .

To obtain a sensible diffusion limit it is necessary to use a different scaling for the continuum time axis from that used for the WF model. It turns out that, if we define

$$t = \frac{2\tau}{M^2}, \quad \delta t = \frac{2}{M^2}, \quad X(t) = \frac{1}{M} Y(\tau), \quad q_{12} = \frac{1}{2} M u_{12}, \quad q_{21} = \frac{1}{2} M u_{21}. \quad (123)$$

then we again arrive at Eqs. (58) and (59), and hence recover identical forward and backward Kolmogorov equations as those for the WF model with mutations.

The Cannings Model

The WF and Moran models are particular cases of a class of models known collectively as the *Cannings model* (Cannings, 1974). Again consider a population of fixed size M and discrete generations $\tau=0, 1, 2, \dots$. Define a random vector

$$\mathbf{V}(\tau) = (V_1(\tau), \dots, V_M(\tau)), \quad (124)$$

equal to the (non-negative integer) number of offspring produced by the M individuals at time τ . The $V(\tau)$ are assumed to be i.i.d. for different τ , and to share their distribution with a generic random variable V satisfying firstly that

$$\sum_{\alpha=1}^M V_{\alpha} = M, \quad (125)$$

and secondly that

$$\text{Prob}(V_1 = v_1, \dots, V_M = v_M) = \text{Prob}(V_1 = v_{\pi(1)}, \dots, V_M = v_{\pi(M)}), \quad (126)$$

where $\pi(\cdot)$ is any permutation of the indices $1, \dots, M$. The second requirement is known as the exchangeability property, and the model is sometimes referred to as the Cannings exchangeable model. Note that Eqs. (125) and (126) imply that $E(V_{\alpha})=1$.

Suppose now that the population partitions into two alleles types, A_1 and A_2 , which are inherited without mutation. Let $Y(\tau)$ be the number of A_1 alleles and $M-Y(\tau)$ be the number of A_2 alleles present at time τ , so by appropriate labelling of the indices we can write

$$Y(\tau+1) = \sum_{\alpha=1}^{Y(\tau)} V_{\alpha}(\tau). \quad (127)$$

The 2-allele WF model without mutations is the particular case

$$\mathbf{V}^{\text{WF}} \sim \text{multinomial}\left(M; \frac{1}{M}, \dots, \frac{1}{M}\right), \quad (128)$$

and the Moran model is the particular case

$$\mathbf{V}^{\text{Moran}} \sim \text{uniform}\{(k_1, \dots, k_M) : (k_{\pi(1)}, \dots, k_{\pi(M)}) = (2, 0, 1, \dots, 1)\}. \quad (129)$$

Other specific cases are a trivial case in which each individual has exactly 1 offspring, for which neither allele ever fixes unless $Y(0)=0$ or M , and another trivial case in which one randomly chosen individual has all M offspring, in which the allele of the chosen individual fixes in 1 generation. In general, a Cannings model can be extended to include multiple alleles and may include neutral mutations, but not selection, as that would violate exchangeability.

The coalescent generalises easily to any Cannings model (Etheridge, 2011, Section 2.2). Recall that in deriving the coalescent for a WF model we defined τ_C to be the number of generations counted backwards in time to the most recent common ancestor of two randomly chosen individuals within the population. The probability that the two chosen individuals are siblings from the same parent α is $v_{\alpha}(v_{\alpha}-1)/(M(M-1))$, where v_{α} is the number of children produced by α . Defining c_M to be the probability that τ_C

is one generation, it follows that

$$\begin{aligned}
 c_M &= \text{Prob}(\tau_C = 1) \\
 &= \sum_{v_1, \dots, v_M} \text{Prob}(\tau_C = 1 | \mathbf{V}(-1) = (v_1, \dots, v_M)) \times \text{Prob}(\mathbf{V}(-1) = (v_1, \dots, v_M)) \\
 &= \sum_{v_1, \dots, v_M} \sum_{\alpha=1}^M \frac{v_\alpha(v_\alpha - 1)}{M(M-1)} \text{Prob}(\mathbf{V}(-1) = (v_1, \dots, v_M)) \\
 &= \frac{1}{M(M-1)} \sum_{\alpha=1}^M E[V_\alpha(V_\alpha - 1)] \\
 &= \frac{1}{M-1} E[V_1(V_1 - 1)] \\
 &= \frac{1}{M-1} \text{Var}(V_1),
 \end{aligned} \tag{130}$$

where we have used the exchangeability property in the second last line and the fact that $E(V_1) = 1$ in the last line. Eq. (102) then generalises to

$$\text{Prob}(\tau_C = \tau) = (1 - c_M)^{\tau-1} c_M. \tag{131}$$

The transition to a continuum time limit depends on the scaling behaviour of $\text{Var}(V_1)$ as $M \rightarrow \infty$. For instance, Eq. (51) defines an appropriate time scale for WF, while Eq. (123) defines an appropriate time scale for the Moran model. Once an appropriate coalescence time scale is established, the results in the section The Kingman Coalescent and the Watterson Estimator will in general carry through with appropriate modifications.

Branching Models

Branching models have their origins in the unpublished work of Bienaymé dating from 1845 (see Heyde and Seneta, 1972), and the work of Watson and Galton (1875) on the fate of patrilineally inherited family surnames. The Bienaymé-Galton-Watson (BGW) branching model differs from Cannings models in that the total population size is not fixed, or even specified deterministically, but varies stochastically.

Consider a population of $M(\tau)$ haploid individuals which are assumed to reproduce in discrete, non-overlapping generations $\tau = 0, 1, 2, \dots$. The population is partitioned into K allele types such that the number of copies of type k within the population is $Y_k(\tau)$. Thus $M(\tau) = \sum_{k=1}^K Y_k(\tau)$. In the simplest version of the model described here the alleles are assumed to be neutral with respect to selection and no mutation between allele types occurs. The central tenet of the BGW model is an assumption that the (non-negative integer) number of offspring per individual in any generation is given by a set of i.i.d. random variables $V_\alpha^{(k)}$, $\alpha = 1, \dots, Y_k(\tau)$, whose common distribution is denoted by a generic non-negative integer valued random variable V with mean and variance

$$E(V) = \lambda, \quad \text{Var}(V) = \sigma^2, \tag{132}$$

and finite moments to all higher orders. Thus

$$Y_k(\tau + 1) = \sum_{\alpha=1}^{Y_k(\tau)} V_\alpha^{(k)}, \tag{133}$$

and if the $Y_k(0)$ are mutually independent, then $Y_k(\tau)$ are mutually independent at all subsequent times τ . The standard formula for the mean of the sum of a random number of i.i.d. random variables (Ewens and Grant, 2005, p90) gives $E(Y_k(\tau + 1)) = \lambda E(Y_k(\tau))$. Given initial conditions

$$Y_k(0) = i_k, \quad M(0) = m_0 = \sum_{k=0}^K i_k, \tag{134}$$

it follows that

$$E(Y_k(\tau)) = i_k \lambda^\tau, \quad E(M(\tau)) = m_0 \lambda^\tau. \tag{135}$$

Consider now the fate of any one allele, the number of copies of which we will denote generically as $Y(\tau)$. If the probability generating functions of V and $Y(\tau)$ are respectively

$$G_V(\xi) = E(\xi^V), \quad G_{Y(\tau)}(\xi) = E(\xi^{Y(\tau)}), \tag{136}$$

then the formula for the probability generating function for the sum of a random number of i.i.d. random variables (Johnson *et al.*, 1993, p344) gives

$$G_{Y(\tau+1)}(\xi) = G_{Y(\tau)}(G_V(\xi)). \tag{137}$$

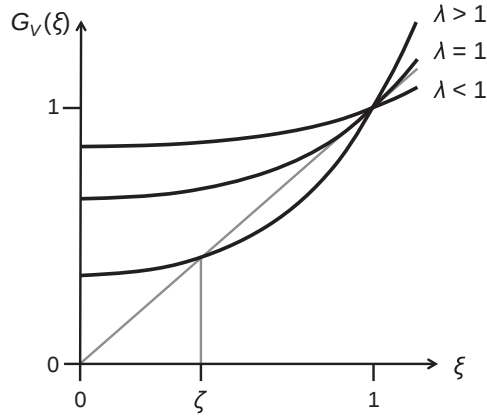


Fig. 9 The stable fixed point of $\zeta = G_V(\zeta)$.

Suppose we start with an initial population of just one individual, in which case $Y(0)=1$ and $G_{Y(0)}(\xi)=\xi$. Applying the above iterative formula gives

$$\begin{aligned} G_{Y(1)}(\xi) &= G_{Y(0)}(G_V(\xi)) = G_V(\xi), \\ G_{Y(2)}(\xi) &= G_{Y(1)}(G_V(\xi)) = G_V(G_V(\xi)), \\ &\vdots \\ G_{Y(\tau)}(\xi) &= \underbrace{G_V(G_V(\dots G_V(\xi)\dots))}_{\tau \text{ times}} = G_V(G_{Y(\tau-1)}(\xi)). \end{aligned} \quad (138)$$

Also, since

$$G_{Y(\tau)}(\xi) = \sum_{y=0}^{\infty} \xi^y \text{Prob}(Y(\tau)=y) = \text{Prob}(Y(\tau)=0) + O(\xi), \quad (139)$$

the probability of extinction at or before generation τ is

$$\text{Prob}(Y(\tau)=0) = G_{Y(\tau)}(0). \quad (140)$$

Thus the probability of eventual extinction, $\zeta = \text{Prob}(Y(\infty)=0) = G_{Y(\infty)}(0)$, is, by Eq. (138), the stable fixed point of the iterative equation

$$\zeta = G_V(\zeta). \quad (141)$$

Noting that

$$G_V(1)=1, \quad G'_V(1)=\lambda, \quad \text{and } G''_V(1)=\sigma^2 + \lambda^2 - \lambda > 0, \quad (142)$$

since $\lambda^2 - \lambda = E(V(V-1)) \geq 0$ if V only takes values $0, 1, 2, \dots$, the graph of $G_V(\xi)$ passes through $(1, 1)$ with slope λ and is convex, as shown in Fig. 9. It follows that

$$\zeta \begin{cases} < 1 & \text{if } \lambda > 1, \\ = 1 & \text{if } \lambda \leq 1. \end{cases} \quad (143)$$

Moreover, since lineages are independent, if the initial abundance of an allele is $Y(0)=i$, then the probability of eventual extinction is

$$\text{Prob}(Y(\infty)=0|Y(0)=i) = \begin{cases} \zeta^i & \text{if } \lambda > 1, \\ 1 & \text{if } \lambda \leq 1. \end{cases} \quad (144)$$

The upshot of this is that each allele will almost surely eventually become extinct in the case of sub-critical growth, $\lambda < 1$, or critical growth, $\lambda = 1$, and that there is a finite probability $1 - \zeta^i$ that an allele will never become extinct in the case of supercritical growth, $\lambda > 1$. Furthermore, in the presence of 2 alleles with initial abundances i_1 and i_2 and supercritical growth, there is a probability $(1 - \zeta^{i_1})(1 - \zeta^{i_2})$ that the population will persist indefinitely with neither allele ever fixing throughout the population.

The Continuum Bienaymé-Galton-Watson Branching Model

The diffusion approximation of a BGW branching process via a forward Kolmogorov diffusion equation has been analysed in detail by Feller (1951a) and is summarised in the texts by Bailey (1964) and Cox and Miller (1978). For an analysis in the context of genetic drift, see Burden and Simon (2016).

For each allele type A_k the continuum limit is obtained by defining

$$t = \frac{\tau}{m_0}, \quad \delta t = \frac{1}{m_0}, \quad X_k(t) = \frac{Y_k(\tau)}{m_0}, \quad x_{k0} = \frac{i_k}{m_0}, \quad (145)$$

where m_0 and i_k are the initial conditions specified in Eq. (134). The simultaneous limits $m_0 \rightarrow \infty$, $\lambda \rightarrow 1$ are then taken in such a way that

$$\alpha = m_0 \log \lambda, \quad (146)$$

and σ^2 remain fixed. Setting $\delta X_k(t) = (Y_k(\tau + 1) - Y_k(\tau))/m_0$, one obtains

$$\begin{aligned} E(\delta X_k(t) | X_k(t) = x) &= \frac{1}{m_0} (\lambda - 1) x m_0 \\ &= \alpha x \delta t + o(\delta t), \end{aligned} \quad (147)$$

and

$$\begin{aligned} E(\delta X_k(t)^2 | X_k(t) = x) &= \text{Var}(\delta X_k(t) | X_k(t) = x) + E(\delta X_k(t) | X_k(t) = x)^2 \\ &= \frac{1}{m_0^2} \sigma^2 x m_0 + O\left(\frac{1}{m_0^2}\right) \\ &= \sigma^2 x \delta t + o(\delta t). \end{aligned} \quad (148)$$

Comparing with the general form Eqs. (53) and (54) yields the forward Kolmogorov equation for the density function $f_{Xk}(x, t)$,

$$\frac{\partial f_{Xk}}{\partial t} = -\alpha \frac{\partial}{\partial x} (x f_{Xk}) + \frac{1}{2} \sigma^2 \frac{\partial^2}{\partial x^2} (x f_{Xk}). \quad (149)$$

The process defined by this forward Kolmogorov equation is often referred to as a *Feller diffusion*.

The solution obtained by Feller (1951a,b) corresponding to the initial condition $f_{Xk}(x, 0) = \delta(x - x_{k0})$ is

$$\begin{aligned} f_{Xk}(x, t) &= \delta(x) p_0(t) \\ &\quad + \tilde{\kappa}(t) \left(\frac{x_{k0} e^{-\alpha t}}{x} \right)^{\frac{1}{2}} \exp\{-\tilde{\kappa}(t)(x_{k0} + x e^{-\alpha t})\} I_1\left(2\tilde{\kappa}(t)(x_{k0} x e^{-\alpha t})^{\frac{1}{2}}\right) \end{aligned} \quad (150)$$

where $\delta(x)$ is the Dirac delta function defined in Appendix A, $I_1(\cdot)$ is the modified Bessel function of order 1,

$$\tilde{\kappa}(\tau) = \begin{cases} \frac{2\alpha}{\sigma^2(1 - e^{-\alpha t})} & \text{if } \alpha \neq 0 \\ \frac{2}{\sigma^2 t} & \text{if } \alpha = 0, \end{cases} \quad (151)$$

and

$$p_0(t) = e^{-\tilde{\kappa} x_{k0}}. \quad (152)$$

An outline of Feller's derivation of this solution using the Laplace transform of $f_{Xk}(x, t)$ is given on pages 235 and 250 of Cox and Miller (1978).

The delta-function term represents a point mass at zero indicating the finite probability $p_0(t)$ that the allele becomes extinct at or before time t . The probability of eventual extinction is

$$\lim_{t \rightarrow \infty} p_0(t) = \begin{cases} 1 & \text{if } \alpha \leq 0, \\ e^{-2\alpha x_{k0}/\sigma^2} & \text{if } \alpha > 0. \end{cases} \quad (153)$$

It can be checked that this is consistent with the corresponding probability of extinction for the discrete case as follows. From Eqs. (141) and (142),

$$\zeta = 1 + \lambda(\zeta - 1) + \frac{1}{2}(\sigma^2 + \lambda^2 - \lambda)(\zeta - 1)^2 + O((\zeta - 1)^3). \quad (154)$$

Solving for the stable root shown in Fig. 9, and noting that ζ is close to 1 for λ close to 1, we have either $\zeta = 1$ if $\lambda \leq 1$, or

$$\begin{aligned} \zeta &= 1 + \frac{2(1 - \lambda)}{\sigma^2} + O((\lambda - 1)^2) \\ &= 1 - \frac{2 \log \lambda}{\sigma^2} + O((\log \lambda)^2) \\ &= 1 - \frac{2\alpha}{m_0 \sigma^2} + O\left(\frac{1}{m_0^2}\right), \end{aligned} \quad (155)$$

if $\lambda > 1$. In the continuum limit, and noting that $i_k = m_0 x_{k0}$, Eq. (144) gives the probability of extinction for the $\lambda > 1$ case as

$$\lim_{m_0 \rightarrow \infty} \left(1 - \frac{2\alpha}{m_0 \sigma^2} \right)^{m_0 x_{k0}} = e^{-2\alpha x_{k0} / \sigma^2}, \quad (156)$$

agreeing with the $\alpha > 0$ case of Eq. (153). The $\alpha \leq 0$ case clearly corresponds to the $\lambda \leq 1$ stable solution.

For the critical value $\alpha = 0$ and $K = 2$ alleles, Burden and Simon, (2016) have shown that the diffusion limit of the BGW model has similar, but not identical, distributions of fixation times to the neutral WF and Cannings models without mutations. More specifically, they demonstrate that, given an initial population in which a fraction x_0 are of allele-type A_1 and a fraction $1 - x_0$ are of allele-type A_2 , the probability that A_1 fixes is x_0 , and the probability that A_2 fixes is $1 - x_0$, in agreement with Eqs. (8) and (9) for the WF model. Furthermore, if $T^*(x_0)$ is defined to be the time for A_1 to fix, conditional on A_1 fixing from an initial population x_0 , then

$$E(T^*(x_0)) = -\frac{2(1-x_0)}{x_0 \sigma^2} \log(1-x_0). \quad (157)$$

When $\sigma^2 = 1$ this result agrees with the fixation time for the WF model, Eq. (34). On the other hand, it can be shown that the variance of $T^*(x_0)$ differs from that of the Wright-Fisher model. Results for fixation probabilities can be extended to the case of supercritical growth, $\alpha > 0$, with complications due to the fact that there is a finite probability that neither allele ever fixes.

Closing Remarks

The intention behind this article is to provide a solid mathematical background to many of the models underpinning current research in population genetics. No survey of this length could possibly cover all aspects of mathematical population genetics.

Some of the basic topics we have not covered include infinitely-many alleles models (Kimura and Crow, 1964), tests for neutrality versus selection such as Tajima's D statistic (Tajima, 1989), and interactions between multiple loci, which is also known as epistasis. Also not covered are non-random mating, recombination, and selective sweeps, that is, loss of genetic variation in the genomic vicinity of a beneficial mutation. The basic form of the WF and more general exchangeable models can be extended to study spatially structured populations and gene flow, including the transfer of alleles from one population to another due to migration. From the mathematical point of view, the diffusion limit of any model accessible to the forward Kolmogorov equation can equally be analysed in terms of stochastic differential equations.

The coalescent has had a major influence on the population genetics literature as improvements in genotyping technologies and the development of several software packages for simulating genealogies have become available, and perhaps deserves an extensive review in its own right.

From the point of view of population sizes, loss of genetic diversity through population bottlenecks and the founder effect are also of interest. Finally, the Bienaymé-Galton-Watson branching model can be adapted in a number of ways. Two examples are the stabilization of population size by introducing logistic growth (Lambert et al., 2005), and the introduction of multiple alleles via multitype branching processes (Mode, 1971).

Appendix A Derivation of the Forward Kolmogorov Equation

In order to facilitate the derivation of the forward Komogorov equation we first make a short digression to introduce a mathematical abstraction known as the Dirac delta function. The Dirac delta function (or just 'delta function' for short) is an example of a construct known to pure mathematicians as a generalised function. It was introduced by the physicist Paul Dirac (1930) as a convenient notation for studying quantum mechanics and was subsequently formalised rigorously by the mathematician Laurent Schwartz. However, to have a practical working understanding of the delta function it is not necessary to delve into the arcane pure mathematics of generalised functions. Here we take a simple but adequate heuristic approach.

The Dirac delta function, $\delta(x)$ is equal to zero everywhere on the real number line except at $x=0$, where it is envisaged as an infinite spike, the area beneath which is 1. Thus

$$\delta(x) = \begin{cases} \infty & \text{if } x = 0, \\ 0 & \text{if } x \neq 0, \end{cases} \quad (A.1)$$

and

$$\int_{-\varepsilon}^{\varepsilon} \delta(x) dx = 1, \quad (A.2)$$

for any $\varepsilon > 0$. The delta function can be represented as a limit of the normal distribution density functions centred about zero as the variance tends to 0 (see Fig. 10). An important property of the delta function is that, for any function $f(x)$ which is continuous at $x=a$,

$$\int_{-\infty}^{\infty} f(x) \delta(x-a) dx = f(a). \quad (A.3)$$

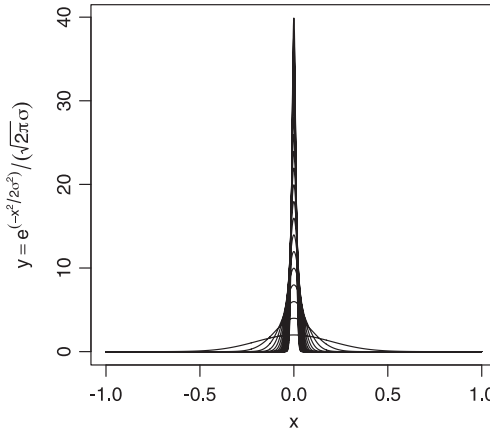


Fig. 10 Representation of the Dirac delta function as the limit of a normal distribution centred about zero as its variance σ^2 tends to zero.

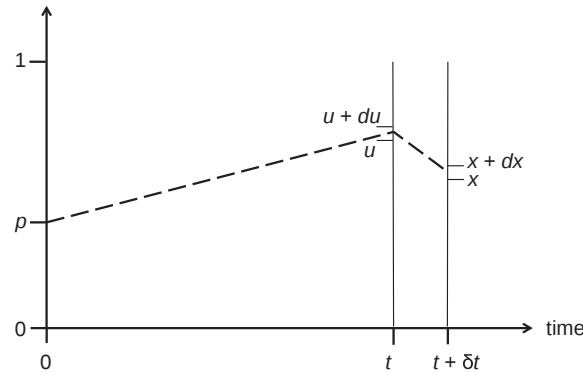


Fig. 11 Derivation of the forward Kolmogorov equation: decomposing $f(x, p, t + \delta t)dx$ into a sum over paths through intermediate events $u \leq X(t) < u + du$ at time t .

By formally integrating by parts we also obtain meaning for derivatives of the delta function, namely

$$\int_{-\infty}^{\infty} f(x) \delta'(x-a) dx = -f'(a), \quad \int_{-\infty}^{\infty} f(x) \delta''(x-a) dx = f''(a). \quad (\text{A.4})$$

We will derive the forward Kolmogorov equation for a generic population genetics model based on a discrete time finite state Markov chain. As previously, consider a genomic site at which two alleles A_1 and A_2 are possible in a population of fixed size M , and let the random variable $Y(\tau)$ be the number of A_1 alleles present at timestep $\tau=0,1,2,\dots$ labelling non-overlapping generations. The range of $Y(\tau)$ is $\Omega_{Y(\tau)} = \{0, \dots, M\}$. Assume also that we are given a time-independent Markov transition matrix

$$p_{ij} = \text{Prob}(Y(\tau+1) = j | Y(\tau) = i), \quad i, j = 0, \dots, M. \quad (\text{A.5})$$

With the intention of taking the limit $M \rightarrow \infty$, make the substitutions:

$$t = \frac{\tau}{M}, \quad \delta t = \frac{1}{M}, \quad X(t) = \frac{1}{M} Y(\tau),$$

$$\delta X(t) = X(t + \delta t) - X(t) = \frac{1}{M} [Y(\tau+1) - Y(\tau)]. \quad (\text{A.6})$$

After taking the limit $M \rightarrow \infty$, t becomes a continuous time variable over the interval $[0, \infty)$ and the fraction $X(t)$ of the population with allele A_1 becomes a continuous random variable with range $\Omega_X = [0, 1]$. Now define the probability density $f(x; p, t)$ corresponding to $X(t)$, conditional on the initial condition $X(0) = p$, that is,

$$f(x; p, t) dx = \text{Prob}(x \leq X(t) < x + dx | X(0) = p), \quad 0 \leq x, p \leq 1, \quad t > 0. \quad (\text{A.7})$$

Our aim is to find a partial differential equation governing the evolution of $f(x; p, t)$.

The derivation of the forward Kolmogorov equation begins with decomposing $\text{Prob}(x \leq X(t + \delta t) < x + dx | X(0) = p)$ into a sum over intermediate events $u \leq X(t) < u + du$, which we write formally as

$$\begin{aligned} & \text{Prob}(x \leq X(t + \delta t) < x + dx | X(0) = p) \\ &= \sum_{u \in [0,1]} \text{Prob}(x \leq X(t + \delta t) < x + dx | u \leq X(t) < u + du) \\ & \quad \times \text{Prob}(u \leq X(t) < u + du | X(0) = p). \end{aligned} \quad (\text{A.8})$$

A path through one such intermediate state is shown in Fig. 11. At this point one should think of du and dx as being infinitesimal, and δt as being finite but small, in which case we can use Eq. (A.7) to rewrite the formal sum over u as

$$f(x; p, t + \delta t) = \int_0^1 du f(x; u, \delta t) f(u; p, t). \quad (\text{A.9})$$

In order to make sense of the last equation we introduce one more assumption, namely that the moments of the random variable $\delta X(t)$ defined by Eq. (A.6) are of the form

$$\begin{aligned} E[\delta X(t) | X(t) = u] &= a(u)\delta t + o(\delta t) \\ E[(\delta X(t))^2 | X(t) = u] &= b(u)\delta t + o(\delta t) \\ E[(\delta X(t))^k | X(t) = u] &= o(\delta t), \quad k = 3, 4, \dots \end{aligned} \quad (\text{A.10})$$

as $\delta t \rightarrow 0$ for finite, well behaved functions $a(u)$ and $b(u)$ defined for $0 \leq u \leq 1$. This assumption essentially means that the density function $f(x; u, \delta t)$ is a narrow distribution centred about a mean approximately equal to $u + a(u)\delta t$ with variance approximately equal to $b(u)\delta t$ when δt is small. In other words, the point x in Fig. 11 is unlikely to stray further than a distance of $O(\sqrt{\delta t})$ from the point u . We claim we can therefore represent $f(x; u, \delta t)$ in terms of the Dirac delta function as

$$f(x; u, \delta t) = \delta(x - u) - a(u)\delta'(x - u)\delta t + \frac{1}{2}b(u)\delta''(x - u)\delta t + o(\delta t). \quad (\text{A.11})$$

One can check using the properties of the delta function that this is consistent with Eq. (A.10). For instance,

$$\begin{aligned} E[\delta X(t) | X(t) = u] &= E[X(t + \delta t) - X(t) | X(t) = u] \\ &= \int_0^1 (x - u) f(x; u, \delta t) dx \\ &= \int_0^1 (x - u) [\delta(x - u) - a(u)\delta'(x - u)\delta t + \frac{1}{2}b(u)\delta''(x - u)\delta t] dx + o(\delta t) \\ &= 0 + \int_0^1 \left[\frac{\partial}{\partial x} (x - u) \right] a(u) \delta(x - u) \delta t dx \\ & \quad + \frac{1}{2} \int_0^1 \left[\frac{\partial^2}{\partial x^2} (x - u) \right] b(u) \delta(x - u) \delta t dx + o(\delta t) \\ &= 0 + \int_0^1 a(u) \delta(x - u) \delta t dx + 0 + o(\delta t) \\ &= a(u)\delta t + o(\delta t). \end{aligned} \quad (\text{A.12})$$

Similarly one can confirm the remaining two expectation values.

Returning to Eq. (A.9) and expanding the left hand side to first order in δt , we have

$$\begin{aligned} & f(x; p, t) + \frac{\partial f(x; p, t)}{\partial t} \delta t \\ &= \int_0^1 f(u; p, t) [\delta(x - u) - a(u)\delta'(x - u)\delta t + \frac{1}{2}b(u)\delta''(x - u)\delta t] du + o(\delta t) \\ &= f(x; p, t) - \int_0^1 \frac{\partial}{\partial u} [f(u; p, t) a(u)] \delta(x - u) \delta t du \\ & \quad + \frac{1}{2} \int_0^1 \frac{\partial^2}{\partial u^2} [f(u; p, t) b(u)] \delta(x - u) \delta t du + o(\delta t) \\ &= f(x; p, t) + \left(-\frac{\partial}{\partial x} [f(x; p, t) a(x)] + \frac{1}{2} \frac{\partial^2}{\partial x^2} [f(x; p, t) b(x)] \right) \delta t + o(\delta t). \end{aligned} \quad (\text{A.13})$$

Equating the coefficient of δt on both sides finally gives the forward Kolmogorov equation

$$\frac{\partial f(x; p, t)}{\partial t} = -\frac{\partial}{\partial x}[a(x)f(x; p, t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[b(x)f(x; p, t)]. \quad (\text{A.14})$$

For any specific population genetics model, the functions $a(x)$ and $b(x)$ are determined from the Markov transition matrix p_{ij} .

One can perform a similar calculation by partitioning the path from $X(0)=p$ to $X(t+\delta t)=x$ into an interval 0 to δt and an interval from δt to $t+\delta t$ to obtain the *backward Kolmogorov equation* (Ewens, 2004, p138)

$$\frac{\partial f(x; p, t)}{\partial t} = a(p)\frac{\partial}{\partial p}f(x; p, t) + \frac{1}{2}b(p)\frac{\partial^2}{\partial p^2}f(x; p, t). \quad (\text{A.15})$$

The backward equation yields the same solution as the forward equation, the difference being that in the forward equation $f(x; p, t)$ is treated as a function with independent variables x and t , and p treated as a parameter, whereas in the backward equation $f(x; p, t)$ is treated as a function with independent variables p and t , and x treated as a parameter.

See also: Algorithms for Strings and Sequences: Multiple Alignment. Comparative Epigenomics. Epidemiology: A Review. Genetics and Population Analysis. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics

References

- Abramowitz, M., Stegun, I.A., *et al.*, 1972. Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables. New York, NY: Dover.
- Aggarwala, V., Voight, B.F., 2016. An expanded sequence context model broadly explains variability in polymorphism levels across the human genome. *Nature Genetics* 48 (4), 349.
- Baake, E., Bialowons, R., 2008. Ancestral processes with selection: Branching and Moran models. In: Miekisz, J. (Ed.), *Stochastic Models in Biological Sciences*, vol. 80. Banach Center Publications, Institute of Mathematics, Polish Academy of Sciences, pp. 33–52.
- Bailey, N.T.J., 1964. The Elements of Stochastic Processes With Applications to the Natural Sciences. New York, NY: Wiley.
- Burden, C.J., Tang, Y., 2016. An approximate stationary solution for multi-allele neutral diffusion with low mutation rates. *Theoretical Population Biology* 112, 22–32.
- Burden, C.J., Simon, H., 2016. Genetic drift in populations governed by a Galton–Watson branching process. *Theoretical Population Biology* 109, 63–74.
- Burden, C.J., Tang, Y., 2017. Rate matrix estimation from site frequency data. *Theoretical Population Biology* 113, 23–33.
- Cannings, C., 1974. The latent roots of certain markov chains arising in genetics: A new approach, i. haploid models. *Advances in Applied Probability*, 260–290.
- Cao, M., Shi, J., Wang, J., *et al.*, 2015. Analysis of human triallelic SNPs by next-generation sequencing. *Annals of Human Genetics* 79 (4), 275–281.
- Chen, H., Chen, K., 2013. Asymptotic distributions of coalescence times and ancestral lineage numbers for populations with temporally varying size. *Genetics* 194 (3), 721–736.
- Cox, D.R., Miller, H.D., 1978. The Theory of Stochastic Processes. London: Chapman and Hall.
- Crow, J.F., Kimura, M., 1970. An Introduction to Population Genetics Theory. New York, NY: Evanston, IL: London: Harper & Row, Publishers.
- Dirac, P.A.M., 1930. The Principles of Quantum Mechanics, first ed. Oxford: Oxford University Press.
- Etheridge, A., 2011. Some Mathematical Models From Population Genetics. Ecole D'Ete de Probabilités de Saint-Flour XXXIX-2009, Volume 2012 of Lecture Notes in Mathematics. Berlin; Heidelberg: Springer.
- Etheridge, A., Griffiths, R., 2009. A coalescent dual process in a Moran model with genic selection. *Theoretical Population Biology* 75 (4), 320–330.
- Ewens, W.J., 2004. Mathematical Population Genetics, second ed. New York, NY: Springer.
- Ewens, W.J., Grant, G.R., 2005. Statistical Methods in Bioinformatics: An introduction, second ed. New York, NY: Springer.
- Feller, W., 1951a. Diffusion processes in genetics. *Proceedings of the Second Berkeley Symposium in Mathematical Statistics and Probability* 227, 246.
- Feller, W., 1951b. Two singular diffusion problems. *Annals of Mathematics* 54 (1), 173–182.
- Fisher, R.A., 1930. The Genetical Theory of Natural Selection. Oxford University Press.
- 1000 Genomes Project Consortium, 2010. A map of human genome variation from population scale sequencing. *Nature* 467 (7319), 1061.
- Gillespie, J.H., 2004. Population Genetics: A Concise Guide, second ed. Baltimore, MD: Johns Hopkins University Press.
- Griffiths, R., 1979. A transition density expansion for a multi-allele diffusion model. *Advances in Applied Probability*, 310–325.
- Griffiths, R.C., Tavaré, S., 1994. Sampling theory for neutral alleles in a varying environment. *Philosophical Transactions of the Royal Society B: Biological Sciences* 344 (1310), 403–410.
- Hardy, G.H., 1908. Mendelian proportions in a mixed population. *Science* 28 (706), 49–50.
- Hasegawa, M., Kishino, H., Yano, T.-A., 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution* 22 (2), 160–174.
- Heyde, C., Seneta, E., 1972. Studies in the history of probability and statistics. xxxi. The simple branching process, a turning point test and a fundamental inequality: A historical note on I. J. Bienayme. *Biometrika* 59 (3), 680–683.
- Isaev, A., 2004. Introduction to Mathematical Methods in Bioinformatics. Berlin; Heidelberg: Springer-Verlag.
- Johnson, N., Kotz, S., Kemp, A., 1993. Univariate discrete distributions, Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics, second ed. New York: John Wiley & Sons.
- Kimura, M., 1955a. Solution of a process of random genetic drift with a continuous model. *Proceedings of the National Academy of Sciences of the United States of America* 41 (3), 144–150.
- Kimura, M., 1955b. Stochastic processes and distribution of gene frequencies under natural selection. *Cold Spring Harbor Symposium Quantitative Biology* 20, 33–53.
- Kimura, M., 1964. Diffusion models in population genetics. *Journal of Applied Probability* 1 (2), 177–232.
- Kimura, M., Crow, J.F., 1964. The number of alleles that can be maintained in a finite population. *Genetics* 49 (4), 725.
- Kingman, J.F., 1982. On the genealogy of large populations. *Journal of Applied Probability* 19, 27–43.
- Kingman, J.F., 2000. Origins of the coalescent. 1974–1982. *Genetics* 156 (4), 1461–1463.
- Lambert, A., *et al.*, 2005. The branching process with logistic growth. *The Annals of Applied Probability* 15 (2), 1506–1535.
- Lanave, C., Preparata, G., Saccone, C., Serio, G., 1984. A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution* 20 (1), 86–93.
- Masel, J., 2011. Genetic drift. *Current Biology* 21 (20), R837–R838.
- Mode, C. J., 1971. Multitype Branching Processes: Theory and Applications Volume 34 of Modern Analytic and Computational Methods in Science and Mathematics New York, NY: American Elsevier Pub. Co.
- Moran, P.A.P., 1958. Random processes in genetics. *Mathematical Proceedings of the Cambridge Philosophical Society* 54 (1), 60–71.

- Phillips, C., Amigo, J., Carracedo, A., Lareu, M., 2015. Tetra-allelic SNPs: Informative forensic markers compiled from public whole-genome sequence data. *Forensic Science International: Genetics* 19, 100–106.
- Shen, H., Li, J., Zhang, J., *et al.*, 2013. Comprehensive characterization of human genome variation by high coverage whole-genome sequencing of forty four caucasians. *PLOS ONE* 8 (4), e59494.
- Slatkin, M., Hudson, R.R., 1991. Pairwise comparisons of mitochondrial DNA sequences in stable and exponentially growing populations. *Genetics* 129 (2), 555–562.
- Stern, C., 1943. The Hardy–Weinberg law. *Science* 97 (2510), 137–138.
- Tajima, F., 1989. Statistical method for testing the neutral mutation hypothesis by dna polymorphism. *Genetics* 123 (3), 585–595.
- Tavare, S., 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences* 17, 57–86.
- Tier, C., Keller, J.B., 1978. A tri-allelic diffusion model with selection. *SIAM Journal on Applied Mathematics* 35 (3), 521–535.
- Vogl, C., 2014. Estimating the scaled mutation rate and mutation bias with site frequency data. *Theoretical Population Biology* 98, 19–27.
- Vogl, C., Bergman, J., 2015. Inference of directional selection and mutation parameters assuming equilibrium. *Theoretical Population Biology* 106, 7182.
- Watson, H.W., Galton, F., 1875. On the probability of the extinction of families. *The Journal of the Anthropological Institute of Great Britain and Ireland* 4, 138–144.
- Watterson, G., 1975. On the number of segregating sites in genetical models without recombination. *Theoretical Population Biology* 7 (2), 256–276.
- Waxman, D., 2011. A unified treatment of the probability of fixation when population size and the strength of selection change over time. *Genetics* 188 (4), 907–913.
- Wright, S., 1931. Evolution in mendelian populations. *Genetics* 16 (2), 97159.
- Wright, S., 1969. *Evolution and the Genetics of Populations: The Theory of Gene Frequencies*. vol. 2. Chicago, IL: University of Chicago Press.
- Zeng, K., 2010. A simple multiallele model and its application to identifying preferred-unpreferred codons using polymorphism data. *Molecular Biology and Evolution* 27 (6), 1327–1337.
- Zhu, Y., Neeman, T., Yap, V.B., Huttley, G.A., 2017. Statistical methods for identifying sequence motifs affecting point mutations. *Genetics* 205 (2), 843–856.

Further Reading

- Handbook of statistical genetics. In: Balding, D.J., Bishop, M., Cannings, C. (Eds.), Part 5: Population Genetics 2. Chichester: John Wiley & Sons.
- Durrett, R., 2008. *Probability Models for DNA Sequence Evolution*. New York, NY: Springer Science & Business Media.
- Haccou, P., Jagers, P., Vatutin, V.A., 2005. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge University Press. No. 5 in Cambridge Studies in Adaptive Dynamics.
- Hartl, D., Clark, A., 2007. *Principles of Population Genetics*, fourth ed. Sunderland, MA: Sinauer.
- Lambert, A., 2008. Population dynamics and random genealogies. *Stochastic Models* 24 (S1), 45–163.
- Mode, C.J., Sleeman, C.K., 2012. *Stochastic Processes in Genetics and Evolution: Computer Experiments in the Quantification of Mutation and Selection*. Singapore: World Scientific.
- Rosenberg, N.A., Nordborg, M., 2002. Genealogical trees, coalescent theory and the analysis of genetic polymorphisms. *Nature Reviews Genetics* 3 (5), 380–390.
- Tavare, S., 2004. Part I: Ancestral inference in population genetics. In: Picard, J. (Ed.), *Lectures on Probability Theory and Statistics: Ecole d'Ete de Probabilites de Saint-Flour XXXI – 2001*. Berlin; Heidelberg: Springer, pp. 1–188.
- Wakeley, J., 2009. *Coalescent Theory: An Introduction*. Greenwood Village, CO: Roberts & Company Publishers.
- Yuan, X., Miller, D.J., Zhang, J., Herrington, D., Wang, Y., 2012. An overview of population genetic data simulation. *Journal of Computational Biology* 19 (1), 42–54.

Relevant Websites

- http://www.radford.edu/~rsheehy/Gen_flash/popgen/
Population Genetics Simulation Program, Radford University.
- <http://www.mabs.at/teaching/>
Teaching Resources of the Mathematics and BioSciences Group, University of Vienna, Including a Tutorial on Population Genetics by P. Pfaffelhuber *et al.*
- <http://evolution.gs.washington.edu/pgbook/>
Theoretical Evolutionary Genetics, Draft of Text by Joe Felsenstein, University of Washington.
- <http://evolution.genetics.washington.edu/geccobw.pdf>
Tutorial on Theoretical Population Genetics by Joe Felsenstein, University of Washington.

Biographical Sketch



Conrad Burden is an associate professor in the Mathematical Sciences Institute at the Australian National University lecturing in bioinformatics and biological modelling. His research interests include mathematical population genetics, alignment-free sequence comparison, the analysis of high-throughput sequencing data, and the physico-chemical modelling of microarrays. Before making the move into bioinformatics in 2003 he spent 20 years as a theoretical physicist followed by a short period working in the IT industry. He graduated from the University of Queensland with first class honours in Applied Mathematics in 1979 and The Australian National University with a PhD in Theoretical Physics in 1983. Between 1983 and 1988 he held postdoctoral fellowships in theoretical physics at the Weizmann Institute of Science, Glasgow University, The Australian National University and Flinders University. Between 1988 and 1999 he held a research position in the Department of Theoretical Physics, Australian National University, working mainly in quantum field theory and subatomic particle physics. Between 1999 and 2002 he was a programmer/developer in the IT industry. He is a Fellow of the Australian Institute of Physics.

Computational Systems Biology

Sucheendra K Palaniappan, Ayako Yachie-Kinoshita, and Samik Ghosh, The Systems Biology Institute, Tokyo, Japan and University of Toronto, Toronto, ON, Canada

© 2019 Elsevier Inc. All rights reserved.

Introduction

Rapid advancements in novel and high throughput technologies to observe and record different biological players and their interactions in multi-dimensions have opened interesting “Big Data” challenges. Using novel computational methods to use all this data to convert them into actionable insights is crucial in the biomedical domain. This is even more relevant in systems biology based approaches where the motivation is to use computational techniques to connect and reconcile data from different spatiotemporal scales to gain a systems level understanding of biological systems.

Extensive literature is available in the community highlighting the variety of tools, algorithms and databases available for different aspects of analysis in systems biology (Ghosh *et al.*, 2011) – from Data and Knowledge management, deep curation, *in silico* simulation and model analysis, to physiological modeling, molecular interaction modeling (Kröger and Bry, 2003; Brazma *et al.*, 2006). Successful application of these methodologies in modeling cell cycle dynamics, such as a computational model that explained the effects of over 120 knockout mutations on cell cycle dynamics in yeast (Tyson *et al.*, 2001; Novak and Tyson, 1993), have been demonstrated. Significant progress has also been made in the analysis of signaling pathways – for example, in understanding the dynamics of mitogen-activated protein kinase (MAPK) signaling (Chen *et al.*, 2004; Aoki *et al.*, 2011) – and in cancer drug discovery applications, in which a reagent that was developed using model-based computational analysis is now in clinical trials (Schoeberl *et al.*, 2009, 2010).

Current tools and techniques in computational systems biology has demonstrated their usage in various application areas, as illustrated schematically in Fig. 1. At the same time, the shift in paradigm both in experimental techniques which generate vast quantities of data, and in powerful data analytics, modeling and visualization methodologies, entail the empowerment of computational systems biology models and methodologies which leverage developments in machine learning, big data management and analysis, large scale modeling and simulations, to name a few. This topic article endeavors to provide some key areas of modeling and methodologies – highlighting new directions and developments, to enable computational systems biology to address the new challenges in biology and medicine.

The rest of the paper provides a brief overview of some of the key model and methodologies – including, Pathway Curation, Network analysis, Modeling and Simulation and Data Analytics. We conclude with a perspective on the promises and perils for application of novel models and methodologies in deeper understanding of basic biology as well as drug discovery and healthcare.

Pathway Curation

Biological systems can be thought of as composition of different biological players whose concerted actions lead to the biological function. Advances in high throughput technologies for recording these interactions have led to increased data which

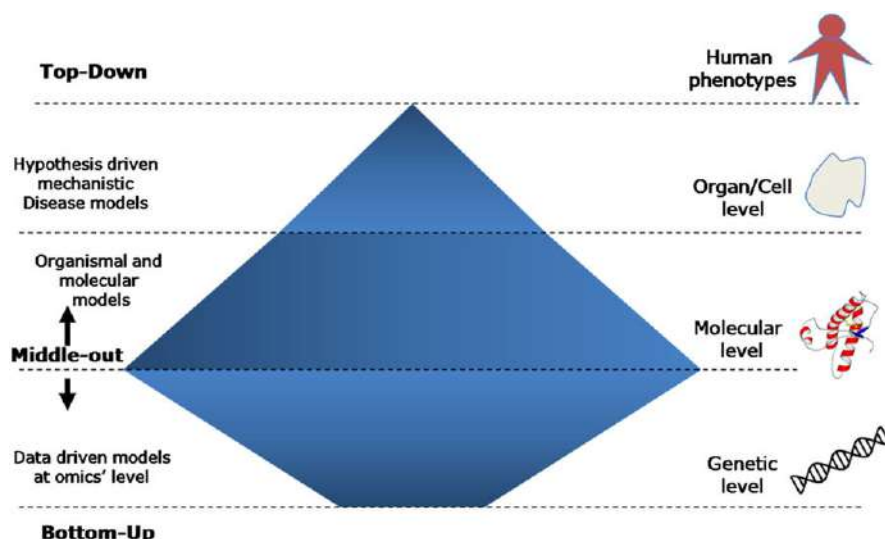


Fig. 1 Models and methodologies in computational systems biology. Taken from Hird, N., Ghosh, S., Kitano, H., 2016. Digital health revolution: Perfect storm or perfect opportunity for pharmaceutical R&D? Drug Discovery Today 21 (6), 900–911. Available at: <http://doi.org/10.1016/j.drudis.2016.01.010>.

has led to increased understanding about these molecules and their associated interactions. With this one can now more objectively reason about how these biological players work in unison at the systems level to perform various biological functions. Systematically recording and understanding these interactions is important and for this we construct and use Pathway maps. Pathway maps can be thought of as a formal representation of the collection of interactions among various biological players. They represent important insights pertaining to the flow of information within a biological system. They could either encode chemical reactions which are involved in the production or breakdown of different metabolites (metabolic pathways), or regulatory interaction between genes and proteins (gene regulatory networks) or record interactions that occur in response to external/internal stimuli (signaling pathways). Studying, analyzing and simulating these pathways can lead to crucial insights leading to designing effective cures for diseases (Kitano, 2002). One of the important undertakings in systems biology is creating formal (and machine readable) representations of these pathways and constitutes the task of Pathway curation (Kitano *et al.*, 2005). Pathways are usually encoded using standards such as SBML (Hucka *et al.*, 2003) and BioPAX (Demir *et al.*, 2010). Numerous pathway databases such as KEGG (Kanehisa and Goto, 2000) and Reactome (Joshi-Tope *et al.*, 2005) store pathway interactions. Attempts to construct detailed process level maps (such as the mTOR pathway map (Caron *et al.*, 2010) and TLR pathway map (Oda and Kitano, 2006)) and disease level pathway maps (such as Alzheimer's (Mizuno *et al.*, 2012) and Parkinson disease maps (Fujita *et al.*, 2014)) have been undertaken.

Typically, pathways are constructed and curated by expert pathway curators who manually read relevant scientific literature in detail, comprehend and select the relevant information about molecular interactions relevant in context and represent it as a pathway diagram. This process is facilitated by several graphical tools such as CellDesigner (Funahashi *et al.*, 2008).

The primary issues with manual approach to pathway curation is time consuming, importantly that the scale at which basic research is progressing, it is often hard to keep pace. Many times, the interpretation of the details is left to the judgment of the curator, hence leading to considerable inter-curator variability. Next, as new research finding are reported, these pathway maps need to be updated or augmented and doing this manually is almost impossible. Developing (semi) automated pathway curation methods using text mining and natural language processing (NLP) for has been an active field of research (Ananiadou *et al.*, 2010; Chowdhury and Sarkar, 2015; Cohen, 2015; Valenzuela-Escarcega *et al.*, 2015;) and is among important goals of large funded research efforts such as the recent DARPA's Big Mechanism Project (Cohen, 2015).

The task of automated pathway curation is essentially that of event extraction where given a text, the task of the automated system will be to first identify different biologically relevant players (using Named entity recognition), identifying the events (or reactions) and mapping the identified biological players to the corresponding reactions. There have been numerous tools such as TEES (Bjorne and Salakoski, 2011), Reach framework (Valenzuela-Escarcega *et al.*, 2015) and more recently the INDRA (Gyori *et al.*, 2017) which specifically tackle this problem. All these methods try to solve the problem of assembling individual reactions types and evaluating their performance. While this is useful, there has been limited acceptance of these techniques in the mainstream curation community. This is primarily because current extraction methods look at the atomic task of extracting at reaction level from text, while ignoring the aspect of threading these pieces together to reconstruct a holistic picture of the pathway (network). In fact, when a human curator is building a pathway map, he is thinking in terms of a network as the ultimate objective which is not well captured by the current NLP techniques.

There have been recent efforts to study and bridge this gap recently (Spranger *et al.*, 2016, 2015). These studies highlight some crucial aspects of improvement for current event extraction systems such as species normalization (improvements needed in unambiguous disambiguation of biological players), Complex identification, looking at a holistic view of the pathway outcome rather than focus only on reaction inference and finally for NLP systems to understand the level of abstraction at which a human interprets the publication and represents it on the pathway map.

Network Based Approaches

Once a pathway map (essentially a network) is obtained after curation, multiple analysis tasks can be carried out. The most straight forward is to purely consider the pathway as a network and performing static analysis on it.

Controllability/Network Motif Analysis

Controllability analysis focuses on analyzing a complex pathway network with the goal of providing insights about the key genes/proteins (called drivers) required to control the network, whose perturbation has a profound impact on the state of the network (Liu *et al.*, 2011). Network controllability has been successfully used in the context of large biological network to characterize how biological networks have special properties which confer then unique capabilities (Müller and Schuppert, 2011; Kawakami *et al.*, 2016). This can be useful for designing therapeutic interventions where identifying the drivers of a network and characterizing their relative importance in changing healthy to disease state is important for targeting them.

Since the pathway is essentially a network/graph, analyzing the structure of the graph can reveal unique substructures within the network that are expressed in majority and confer specific properties to the network. These substructures are often called network motifs and they are believed to serve as building blocks of the network. These network motifs and the impact they have on

the dynamic features and biological functions of the overall network is well studied. In fact, it is believed that these motifs play a crucial part in the regulation and robustness of the network (Shen-Orr *et al.*, 2002; Novák and Tyson, 2008).

Network Inference

While pathway curation concerns with organizing the current knowledge about pathways one the other interesting facet is to discover new previously unknown links in the pathway network. Again, thanks to the high through put technologies, we can measure thousands of biomolecules concurrently over time. The data that is produced from these experiments can be used to discover novel pathway links or study how pathways and their links get affected in diseases/healthy states. The field of network inference deals with inferring these causal relationships between biological players from experimental data. The process is challenging as the network of molecular interactions are complex have numerous control and regulatory mechanisms governing the interactions and the experimental measurements are currently sparse and noisy.

There has been a lot of effort in developing and using efficient algorithms to infer these networks (Hase *et al.*, 2013; Sachs *et al.*, 2009, 2005). Considerable effort needs to also be put into producing accurate networks even with noisy observation data and will continue to be a major challenge in this field.

Modeling and Simulation

Pathway maps such as those discussed the pathway curation section provide a static picture of the events transpiring in a cell, however the dynamics of their interaction in time and space is what confers cellular function and behavior. For this, modeling pathway systems using mathematical formulations and consequently analyzing these models using techniques such as simulations is crucial.

Usually, for modeling studies, one starts from curated pathway maps which statically encode our understand of the mechanisms. A part of the map which is relevant to the current study is isolated after which suitable modeling formalism is chosen to mathematically encode interactions or hypotheses. These mathematical formalisms have kinetic parameters which need to be tuned so that these models faithfully replicate the observed dynamics and forms a large part of the model calibration and validation efforts. Once a model is calibrated, it faithfully captures the dynamics of the underlying system and can now be used for further analysis. These models once calibrated, can be used to predict behavior of the system under perturbations which is crucial to assess robustness of the system. Additionally, they can potentially assist in designing better experiments whereby one can evaluate potential hypotheses on the models first (which have considerably less cost) before validating them using wet-lab experiments. This section attempts to provide a broad overview of field.

There are many formalisms for modeling biological systems. The choice of the framework is usually dictated by the biological process under study, the questions for which we seek to gain insights, the experimental data that is available to us for calibrating and analysis of the model and the abstract level of detail at which we seek to gain insights. Models can be built to either give qualitative (Simao *et al.*, 2005; Fisher *et al.*, 2007; Schaub *et al.*, 2007) insights in case the nature of data is limited to qualitative observations or quantitative (Hua *et al.*, 2006; Janes and Yaffe, 2006). On the other hand, model formulation can be deterministic, stochastic or a hybrid between the two paradigms. Common methods of mathematical modeling include Ordinary differential equations (Aldridge *et al.*, 2006), Partial differential equations (Leung, 1989), Boolean networks (Raeymaekers, 2002), Petri nets (Ruths *et al.*, 2008) or Rule-based approaches (Danos *et al.*, 2008) to name a few.

Deterministic vs Stochastic Approaches to Modeling

The most common approach of modeling biological systems is using deterministic kinetics. Deterministic models such as Ordinary differential equations ignore random fluctuations in molecule numbers and other sources of noise. Consequently, given an initial state of the system its dynamics is uniquely determined by the underlying kinetics.

The most widely used deterministic model is Ordinary Differential Equation (ODEs). They capture the concentration changes of different biological players through the reactions they take part in. The concentration of every biological player over time is assumed to be governed by a set of coupled differential equations. The formulation of these equations is dictated by kinetic laws that govern each reaction (Aldridge *et al.*, 2006). Formulating ODEs relevant to biological systems requires one to have a detailed knowledge about the mechanisms of the reactions, rate constants. May times, restricted classes of these models are used by making several simplifying assumptions on the model formalism. Examples include the piecewise – multi affine models (Batt *et al.*, 2008), qualitative differential equations (Trelease *et al.*, 1999).

Stochasticity is important in the context of biological systems especially when considering biological players present in very low particle numbers (such as molecules participating in transcription, translation or their regulation). At these low numbers the effect of randomness in the system is not ignorable and has major implications on function (McAdams and Arkin, 1997). Additionally, cell-to-cell variability can occur due to random events in the cell which can change the order and number of interactions (Elowitz *et al.*, 2002; Spencer *et al.*, 2009). The most popular method for modeling stochastic systems include Chemical Master Equation (Gillespie, 1977) and Rule based formalisms (Danos *et al.*, 2008) to name a few.

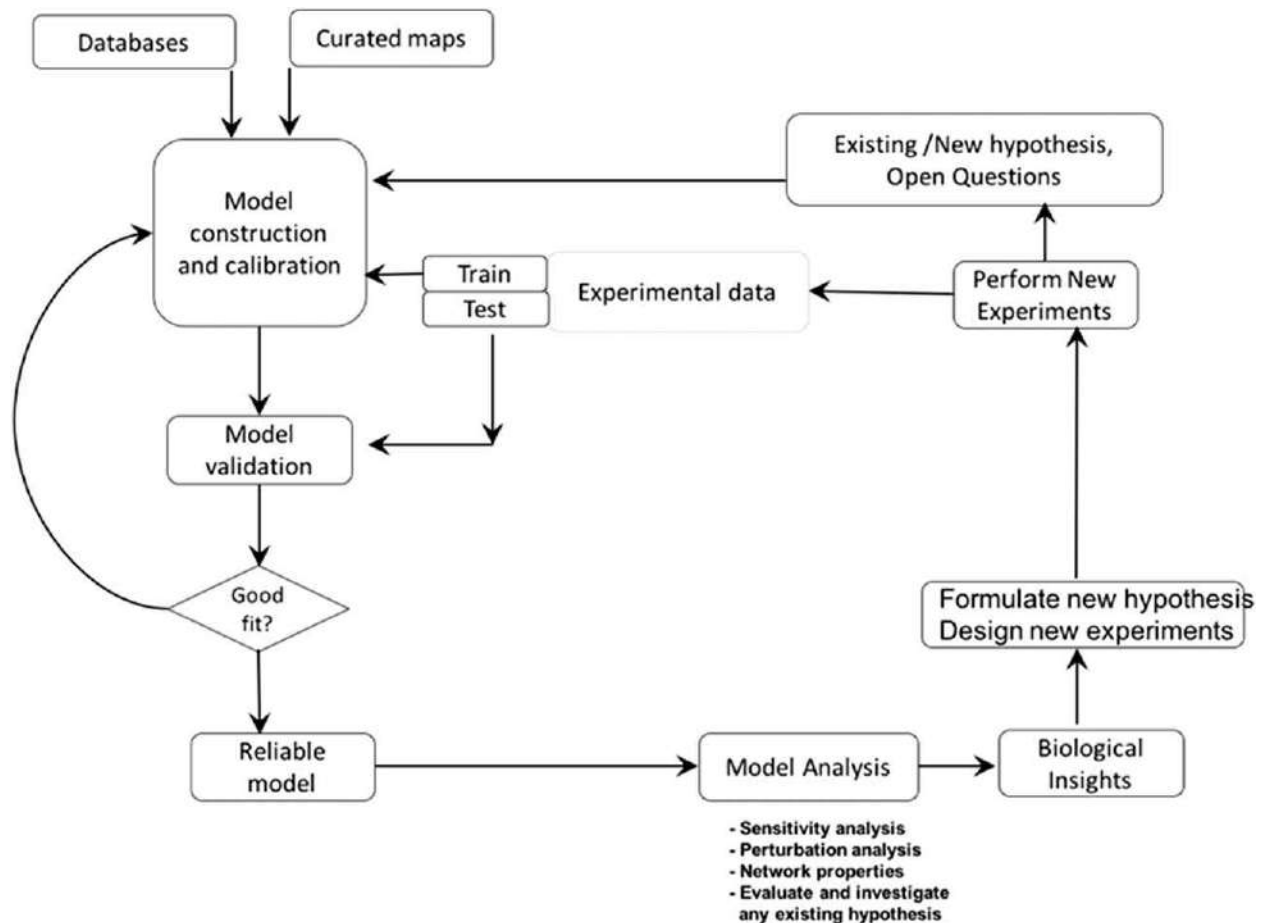


Fig. 2 Model Building steps. adapted from Palaniappan, S.K., 2013. Probabilistic verification and analysis of biopathway dynamics. (Dissertation).

Model Construction

Now we will very briefly discuss the main steps involved in building a reliable computational model as described in Fig. 2. Having decided the formalism and the scope of the modeling exercise, we first build the connections/structure of the model which incorporates all our current understanding of the pathway, this can also incorporate any hypotheses that we have. Depending on the data we have at our disposal and the questions to be asked of it, we choose a suitable modeling formalism to model the pathway.

Model Calibration and Tuning

Model calibration and tuning (often referred to as parameter estimation), deals with estimating unknown parameters (such as kinetic rate constants, initial concentrations of unmeasured players) of the model (depending on the chosen formalism). The final goal is to tune the model so that it can explain the given experimental observations. Usually, the experimental data is divided into two sets, one is used for estimating the parameters of model and the other is used as a validation or test set to check for the robustness of estimated parameters or check for overfitting. Model calibration is essentially a mathematical optimization problem where the goal is to minimize an objective function. The objective function is a goodness of fit measure which measures how far away is the model prediction from experimental data. Parameter estimation is a resource intensive task since evaluating the objective function for each parameter combination involves repeatedly simulating the model. After calibration, the model does through a round of model validation where the model output is evaluated for goodness of fit with the test data (which has not been seen by the model yet). If the fit is reasonably good, the conclusion is that the model is not over fitted and has a reasonable predictive power, it can then be used for further analysis tasks, else more calibrations rounds may be necessary or more data may be necessary to train the model.

Model Analysis

Once we have a model that is calibrated, it can be now be used for various model analysis which will help us harness the full potential of the model. Multiple analysis tasks can be performed on the model. For instance, bifurcation analysis gives insights on

the effect of parameters on qualitative behavior of the system. Bifurcation points point to areas along the parameter space where there is a switch in the system behavior and give important insights about how the system switched between states (say diseased and healthy state). It has been used in the context of biological systems for robustness analysis (Morohashi *et al.*, 2002).

Sensitivity analysis (Schoeberl *et al.*, 2008; Rodriguez-Fernandez and Banga, 2008; Rodriguez-Fernandez *et al.*, 2012) on the other hand quantitatively studies the effect of the changes in kinetic parameters or initial concentrations of model players on the dynamic behavior the model. Sensitivity analysis plays an important role in robustness analysis, optimal experimental design, model reduction and drug target selection. In industrial biology and synthetic biology applications they are used for yield improvement where perturbations to the system can lead to target dynamic profile of select biological players.

As we discussed in previous sections, modeling is essentially an iterative process where, as new data or evidence is obtained or when new hypothesis in the mechanisms needs to be incorporated, we will need to re-estimate some parameters or add new links to the model. Additionally, with every change that is done to the model there is a need to verify and validate these models. This ensures that the new model is consistent with what is known already (nothing that is known has changed during the change) and that it explains the new data or hypothesis that has been newly incorporated. Often as the scale of the modeling exercise increases, ensuring that models are in accordance with the current knowledge is a non-trivial task. Manual analysis and validation is error prone.

Automated methods for analysis and verification of these models using Formal methods such as Model checking offers an attractive alternative (Fisher and Henzinger (2007)). The basic idea is to formalize our knowledge about the system behavior either in qualitative or quantitative terms using a specification language called temporal logics. These formal statements are then automatically analyzed against the Model using efficient algorithms to check if the system conforms to them. We can now automatically reason about system behavior and establish the extent to which a given model is consistent with observations of interest. There is a lot of interest in using formal methods for analyzing the dynamics of models in systems biology (Monteiro *et al.*, 2008; Clarke *et al.*, 2008; Hlavacek, 2009). In fact, it has been suggested model checking could serve as yardstick to check the reliability of models in public domain. Formal methods have also been used successfully for model calibration and sensitivity analysis tasks also (Barnat *et al.*, 2012; Calzone *et al.*, 2006; Palaniappan *et al.*, 2013). While there has been considerable effort on Model checking for systems biology, its adoption into mainstream modeling workflows are still limited owing to limited availability of usable tools which can seamlessly connect into a modeler's workflow.

Data Analytics

While the early approaches in computational systems biology focused primarily on modeling of pathways, biochemical reactions and their dynamic simulation in space and time, combining these techniques with data-driven approaches have gained increasing traction in the community. Enrichment analysis based on biological processes, pathways and functions have formed the cornerstone of bioinformatics and systems biology analysis pipelines (Ghosh *et al.*, 2011; Glass *et al.*, 2014).

With the availability of omics-scale data, novel data analysis workflows for gene expression analysis, proteomics and metabolomics analysis have been successfully developed to study the dynamics of those systems (Glass *et al.*, 2014; Khatri *et al.*, 2012; Caron *et al.*, 2017). A recent trend in data analytics for systems biology is the development of multi-omics and trans-omics analysis – methodologies to connect multi-layered data across the omics scale to reconstruct interaction networks linking genetic factors with phenotypes and environmental factors (Yugi *et al.*, 2016).

The rapid rise in machine learning techniques, particularly the class of feature learning convolution neural network models (CNN), known popularly as deep learning architectures, have opened a range of powerful approaches for data analytics in biology (Webb, 2018). Wide-ranging applications of deep learning models have been developed in cheminformatics, translational biomarker and drug discovery, proteomics, to healthcare and clinical informatics (comprehensive reference available at [deeplearning-biology]). Particular success in genomics field including the DeepVariant model for variant calling, enhancer predictions, and population genomics (Poplin *et al.*, 2016).

New horizons in data analytics are opening rapidly and have the potential for far-reaching impact on systems level study of complex biology of health and disease states. The ability to connect the different models and methodologies in a flexible platform (Ghosh *et al.*, 2011), will play a pivotal role in the next generation of computational systems biology applications as we posit next in the conclusion section.

Conclusion

The paper outlines new directions of models and methodologies in computational systems biology ranging from text-mining guided molecular pathway curation to data-driven approaches in network analysis, inference, dynamic modeling and simulation studies.

While early work in systems biology focused on top-down, hypothesis-driven approaches (refer Fig. 1) to model relatively well-defined processes, with the rise of high-throughput omics scale data together with advances in machine learning based approaches, the recent focus has weighted more on data-driven approaches (refer Fig. 1), especially for the applications in medicine and healthcare. Data-driven approaches allow the ability to harness large volumes of data and obtain insights by extracting patterns. Contrary to hypothesis-driven approaches, these techniques do not always require deep knowledge of the underlying biological process which was studied in the experiment and generated the data used in the models. On the other hand, hypothesis-driven approaches embed the biology mechanism in the modeling approach and thus provide better *explainability* of output results compared to pure data-driven approaches, albeit at the cost of injecting a *hypothesis bias* as well as compromise on the model scope

and time to integrate the process information in the models. As we move towards systems medicine, it becomes apparent that a balanced approach which combines the different models and methodologies in a middle-out manner will be critically needed for computational systems biology to empower the future of medicine – the ability to analyze large-scale data to obtain novel insights together with the ability to explain the mechanism of those insights at molecular levels.

See also: Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: GO and GOA. Biological and Medical Ontologies: Systems Biology Ontology (SBO). Cell Modeling and Simulation. Cluster Analysis of Biological Networks. Computational Approaches for Modeling Signal Transduction Networks. Computational Lipidomics. Computational Systems Biology Applications. Coupling Cell Division to Metabolic Pathways Through Transcription. Data Formats for Systems Biology and Quantitative Modeling. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Functional Enrichment Analysis Methods. Integrative Bioinformatics. Multi-Scale Modelling in Biology. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Network Models. Networks in Biology. Pathway Informatics. Protein-Protein Interactions: An Overview. Quantitative Modelling Approaches. Standards and Models for Biological Data: FGED and HUPO. Standards and Models for Biological Data: SBML. Studies of Body Systems. Visualization of Biological Pathways

References

- Aldridge, B.B., Burke, J.M., Lauffenburger, D.A., Sorger, P.K., 2006. Physicochemical modelling of cell signalling pathways. *Nature Cell Biology* 8, 1195–1203.
- Ananiadou, S., Pyysalo, S., Tsujii, J., Kell, D., 2010. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology* 28, 381–390.
- Aoki, K., Yamada, M., Kunida, K., Yasuda, S., Matsuda, M., 2011. Processive phosphorylation of ERK MAP kinase in mammalian cells. *Proceedings of the National Academy of Sciences of the United States of America* 108, 12675–12680.
- Barnat, J., Brim, L., Krejci, A., *et al.*, 2012. On parameter synthesis by parallel model checking. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9 (3), 693–705.
- Batt, G., Belta, C., Weiss, R., 2008. Temporal logic analysis of gene networks under parameter uncertainty. *IEEE Transactions on Circuits and Systems I (Special Issue on Systems Biology)* 53, 215–229.
- Bjorne, J., Salakoski, T., 2011. Generalizing biomedical event extraction. In: *Proceedings of the BioNLP Shared Task Workshop*, pp. 183–191. (Association for Computational Linguistics).
- Brazma, A., Krestyaninova, M., Sarkans, U., 2006. Standards for systems biology. *Nature Rev. Genet.* 7, 593–605.
- Calzone, L., Chabrier-Rivier, N., Fages, F., Soliman, S., 2006. Machine learning biochemical networks from temporal logic properties. *Transactions on Computational Systems Biology VI*, 68–94.
- Caron, E., Ghosh, S., Matsuoka, Y., *et al.*, 2010. A comprehensive map of the mtor signaling network. *Molecular Systems Biology* 6.
- Caron, E., Roncagalli, R., Hase, T., *et al.*, 2017. Precise Temporal Profiling of Signaling Complexes in Primary Cells Using SWATH Mass Spectrometry. *Cell Reports* 18, 3219–3226.
- Chen, K.C., *et al.*, 2004. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell* 15, 3841–3862.
- Chowdhury, S., Sarkar, R.R., 2015. Comparison of human cell signaling pathway databases—evolution, drawbacks and challenges. *Database* 2015, bau126.
- Clarke, E., Faeder, J., Langmead, C., *et al.*, 2008. Statistical model checking in BioLab: Applications to the automated analysis of T-Cell receptor signaling pathway. *Computational Methods in Systems Biology*, 231–250.
- Cohen, P., 2015. Darpa's big mechanism program. *Physical Biology* 12 (045008), Available at: <http://stacks.iop.org/1478-3975/12/i=4/a=045008>.
- Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J., 2008. Rule-based modeling of signal cellular signalling. In: *Proceedings of the International Conference on Concurrency Theory*, pp. 17–41.
- Demir, E., Cary, M.P., Paley, S., *et al.*, 2010. The biopax community standard for pathway data sharing. *Nature Biotechnology* 28 (9), 935–942.
- Elowitz, M.B., Levine, A.J., Siggia, E.D., Swain, P.S., 2002. Stochastic gene expression in a single cell. *Science Signalling* 297 (5584), 1183.
- Fisher, J., Piterman, N., Hajnal, A., Henzinger, T.A., 2007. Predictive modeling of signaling crosstalk during c. elegans vulval development. *PLOS Computational Biology* 3 (5), e92.
- Fujita, K., Ostaszewski, M., Matsuoka, Y., *et al.*, 2014. Integrating pathways of parkinson's disease in a molecular interaction map. *Molecular Neurobiology* 49, 88–102.
- Funahashi, A., Matsuoka, Y., Jouraku, A., *et al.*, 2008. CellDesigner 3.5: A versatile modeling tool for biochemical networks. In: *Proceedings of the IEEE*, 96, pp. 1254–1265.
- Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., Kitano, H., 2011. Software for systems biology: From tools to integrated platforms. *Nature Reviews Genetics*. doi:10.1038/nrg3096.
- Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* 81 (25), 2340–2361.
- Glass, K., *et al.*, 2014. Annotation enrichment analysis: An alternative method for evaluating the functional properties of gene sets, Kimberly Glass & Michelle Girvan. *Scientific Reports* 4, 4191. doi:10.1038/srep04191.
- Gyori, B.M., Bachman, J.A., Subramanian, K., *et al.*, 2017. From word models to executable models of signaling networks using automated assembly. *Molecular Systems Biology* 13 (11), 954.
- Hase, T., Ghosh, S., Yamanaka, R., Kitano, H., 2013. Harnessing diversity towards the reconstructing of large scale gene regulatory networks. *PLOS Computational Biology* 9 (11), e1003361.
- Hlavacek, W.S., 2009. How to deal with large models. *Molecular Systems Biology* 5, 240.
- Hua, F., Hautaniemi, S., Yokoo, R., Lauffenburger, D.A., 2006. Integrated mechanistic and data-driven modelling for multivariate analysis of signalling pathways. *Journal of the Royal Society Interface* 3 (9), 515–526.
- Hucka, M., Finney, A., Sauro, H.M., *et al.*, 2003. The systems biology markup language (sbml): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19 (4), 524–531.
- Janes, K.A., Yaffe, M.B., 2006. Data-driven modelling of signal-transduction networks. *Nature Reviews Molecular Cell Biology* 7 (11), 820–828.
- Joshi-Tope, G., Gillespie, M., Vastrik, I., *et al.*, 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research* 33, D428–D432.
- Kanehisa, M., Goto, S., 2000. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28, 27–30.
- Kawakami, E., Singh, V.K., Matsubara, K., *et al.*, 2016. Network analyses based on comprehensive molecular interaction maps reveal robust control structures in yeast stress response pathways. *Npj Systems Biology and Applications* 2, 15018. Available at: <https://doi.org/10.1038/npjbsa.2015.18>.
- Khatiri, P., Sirota, M., Butte, A.J., 2012. Ten years of pathway analysis: Current approaches and outstanding challenges. *PLOS Computational Biology* 8 (2), e1002375. Available at: <https://doi.org/10.1371/journal.pcbi.1002375>.

- Kitano, H., 2002. Computational systems biology. *Nature* 420 (6912), 206–210.
- Kitano, H., Funahashi, A., Matsuoka, Y., Oda, K., 2005. Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology* 23 (8), 961–966.
- Kröger, P., Bry, F., 2003. A computational biology database digest: Data, data analysis, and data management. *Distributed and Parallel Databases* 13, 7–42.
- Leung, A.W., 1989. Systems of nonlinear partial differential equations: Applications to biology and engineering. In: *Proceedings of the Mathematics and its Applications*, Kluwer Academic Publishers.
- Liu, Y.-Y., Slotine, J.-J., Barabási, A.-L., 2011. Controllability of complex networks. *Nature* 473 (7346), 167–173. doi:10.1038/nature10011.
- McAdams, H.H., Arkin, A., 1997. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America* 94 (3), 814–819.
- Mizuno, S., Iijima, R., Ogishima, S., *et al.*, 2012. Alzpathway: A comprehensive map of signaling pathways of alzheimer's disease. *BMC Systems Biology* 6, 52.
- Monteiro, P.T., Ropers, D., Mateescu, R., Freitas, A.T., de Jong, H., 2008. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics* 24 (16), i227–i233.
- Morohashi, M., Winn, A.E., Borisuk, M.T., *et al.*, 2002. Robustness as a measure of plausibility in models of biochemical networks. *Journal of Theoretical Biology* 216 (1), 19–30.
- Müller, F.-J., Schuppert, A., 2011. Few inputs can reprogram biological networks. *Nature* 478 (7369), E4–E4. doi:10.1038/nature10543.
- Novak, B., Tyson, J.J., 1993. Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos. *Journal of Cell Science* 106, 1153–1168.
- Novák, B., Tyson, J.J., 2008. Design principles of biochemical oscillators. *Nature Reviews Molecular Cell Biology* 9, 981–991.
- Oda, K., Kitano, H., 2006. A comprehensive map of the toll-like receptor signaling network. *Molecular Systems Biology* 2.
- Palaniappan, S.K., Sucheendra K., *et al.*, 2013. Statistical model checking based calibration and analysis of bio-pathway models. In: *Proceedings of the International Conference on Computational Methods in Systems Biology*, Berlin, Heidelberg: Springer.
- Palaniappan, S.K., 2013. Probabilistic verification and analysis of biopathway dynamics. (Dissertation).
- Poplin, R., Newburger, D., Dijamco, J., *et al.*, 2016. Creating a universal SNP and small indel variant caller with deep neural networks. Available at: <https://doi.org/10.1101/092890>.
- Raeymaekers, L., 2002. Dynamics of boolean networks controlled by biologically meaningful functions. *Journal of Theoretical Biology* 218 (3), 331–341. PMID: 12381434.
- Rodriguez-Fernandez, M., Banga, J.R., Doyle III, F.J., 2012. Novel global sensitivity analysis methodology accounting for the crucial role of the distribution of input parameters: Application to systems biology models. *International Journal of Robust and Nonlinear Control* 22, 1082–1102.
- Rodriguez-Fernandez, M., Banga, J.R., 2008. Global sensitivity analysis of a biochemical pathway model. In: *Proceedings of the 2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics*, pp. 233–242.
- Ruths, D., Muller, M., Tseng, J.T., Nakhleh, L., Ram, P.T., 2008. The signaling Petri net-based simulator: A non-parametric strategy for characterizing the dynamics of cell-specific signaling networks. *PLoS Computational Biology* 4 (2), 1–15.
- Sachs, K., Itani, S., Carlisle, J., *et al.*, 2009. Learning signaling network structures with sparsely distributed data. *Journal of Computational Biology* 16 (2), 201–212.
- Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., Nolan, G.P., 2005. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 308 (5721), 523–529.
- Schaub, M.A., Henzinger, T.A., Fisher, J., 2007. Qualitative networks: A symbolic approach to analyze biological signaling networks. *BMC Systems Biology* 1 (1), 4.
- Schoeberl, B., *et al.*, 2009. Therapeutically targeting ErbB3: A key node in ligand-induced activation of the ErbB receptor-PI3K axis. *Science Signaling* 2, ra31.
- Schoeberl, B., *et al.*, 2010. An ErbB3 antibody, MM-121, is active in cancers with ligand-dependent activation. *Cancer Research* 70, 2485–2494.
- Schoeberl, B., Eichler-Jonsson, C., Gilles, E.D., Muller, G., 2008. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology* 20 (4), 370–375.
- Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U., 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics* 31, 64–68.
- Simao, E., Remy, E., Thieffry, D., Chaouiya, C., 2005. Qualitative modelling of regulated metabolic pathways: Application to the tryptophan biosynthesis in *E. coli*. *Bioinformatics* 21 (suppl. 2), ii190–ii196.
- Spencer, S.L., Gaudet, S., Albeck, J.G., Burke, J.M., Sorger, P.K., 2009. Non-genetic origins of cell-to-cell variability in trail-induced apoptosis. *Nature* 459 (7245), 428–432.
- Spranger, M., Palaniappan, S.K., Ghosh, S., 2015. Extracting biological pathway models from NLP event representations. In: *Proceedings of the ACL 2015 Workshop on Biomedical Natural Language Processing (BioNLP'15)*.
- Spranger, M., Palaniappan, S.K., Ghosh, S., 2016. Measuring the state of the art of automated pathway curation using graph algorithms – A case study of the mTOR pathway. *BioNLP*, Berlin, Germany.
- Trelease, R.B., Henderson, R.A., Park, J.B., 1999. A qualitative process system for modeling nf- κ b and ap-1 gene regulation in immune cell biology research. *Artificial Intelligence in Medicine* 17 (3), 303–321.
- Tyson, J.J., Chen, K., Novak, B., 2001. Network dynamics and cell physiology. *Nature Reviews Molecular Cell Biology* 2, 908–916.
- Valenzuela-Escarcega, M., Hahn-Powell, G., Hicks, T., Surdeanu, M., 2015. A domain-independent rule-based framework for event extraction. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL- IJCNLP)*, pp. 127–132. ACL.
- Webb, S., 2018. Deep learning for biology. In: *Proceedings of the Nature Technology Feature*, Available at: <https://www.nature.com/articles/d41586-018-02174-z#correction-0>.
- Yugi, K., *et al.*, 2016. Trans-omics: How to reconstruct biochemical networks across multiple 'Omic' layers. *Trends in Biotechnology* 34 (4), 276–290.

Relevant Website

<https://github.com/hussius/deeplearning-biology>
Deeplearning-Biology.

Pathway Informatics

Sarita Poonia and Smriti Chawla, Indraprastha Institute of Information Technology, Delhi, India

Sandeep Kaushik, University of Minho, Braga, Portugal

Debarka Sengupta, Indraprastha Institute of Information Technology, Delhi, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Traditional biology focuses on structure and function of important biomolecules such as genes and proteins. But this scientific reductionist approach hinders understanding of their interregulation and its role in execution of the various biophysical processes (Joyner and Pedersen, 2011). With the advancement of technologies and exponential growth in data availability, scientists are now probing into the higher order structures that underlie biological activities. This holistic approach has been widening our knowledge about the functionally interdependent molecular circuitry that orchestrates life (Chan and Loscalzo, 2012).

Pathway informatics is a flourishing interdisciplinary research area that is progressively growing and evolving through a blend of computational, experimental, and theoretical approaches (Karp, 2003). This field focuses on understanding how various types of biomolecules interact with each other to form context specific, complex networks, which we often refer to as biological pathways.

To understand the dynamics of biological systems, various computational approaches and pathway databases have been developed, which help in pathway representation, simulation, visualization for quantitative and qualitative analysis (Cary *et al.*, 2005), thus, helping in better understand cellular processes. In this article, we will discuss different biological pathways, their corresponding databases, pathway analysis, and computational approaches for network inference.

Broad Categories of Molecular Pathways

Metabolic Pathways

A metabolic pathway is a series of interlinked biochemical reactions catalyzed by enzymes. Three major classes of molecules in higher order organisms are proteins, lipids, and carbohydrates. Together, these are considered as the building blocks of life. Metabolic pathways that involve synthesis and storage of these biomolecules are referred to as the anabolic pathways. On the other hand, pathways responsible for breakdown of these biomolecules to generate energy are referred to as the catabolic pathways (Ward, 2016).

Regulatory Pathways

Gene regulatory pathways are responsible for turning genes on or off, which in turn determines availability and concentrations of the related protein molecules (Ma and Zhao, 2013). In eukaryotes, genes, by default, are in the off state when they are tightly bound to histones. But eukaryotic genes manage to escape this silencing due to modifications in histones. While histone acetylation makes DNA open and accessible to DNA binding protein, methylation, coupled with histone modification obscure DNA, which in turn lead to gene silencing. Once DNA is open, it is accessible to transcription factors (TFs), which can either transcribe genes or repress them (Hoopes, 2008). Apart from TFs, a number of long and short noncoding RNAs also participate in the ultra-complex gene regulatory network primarily through post transcriptional modifications (Sengupta and Bandyopadhyay, 2011, 2013; Holoch and Moazed, 2015).

Signaling Pathways

Signal transduction is a cascade of biochemical reactions that take place in a cell when a signal molecule such as hormone or biomolecule binds to a receptor on the cell membrane to perform specific biological processes (Pawson, 1995). Cells use a large number of signaling pathways to regulate biological activities. These signals pass information either through protein–protein interactions or via diffusible elements called referred as secondary messengers (Berridge, 2014). Signaling pathway can be of two types: Extracellular and intracellular. The entire signaling process is regulated by feedback pathways (Berg *et al.*, 2002). Any deviation in the signal transduction pathways could cause dysregulation of biological processes and can lead to diseases such as cancer, diabetes, etc. (Liu *et al.*, 2008). Examples of extracellular stimuli are cyclic AMP signaling pathway, voltage-operated channels (VOCs), receptor-operated channels (ROCs), etc. Endoplasmic reticulum (ER) stress signaling, metabolic messengers are the examples of intracellular stimuli (Berridge, 2014).

Drug Related Pathways

Drugs are xenobiotic, small, foreign molecules. To deal with drugs the human body attempts a number of responses. Some drugs are excreted from the body without involving metabolism. But most drugs need structural modification to facilitate excretion,

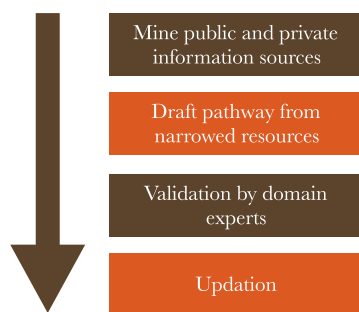


Fig. 1 Pathway construction by curation.

called drug metabolism. There are numerous pathways through which a drug is metabolized or biotransformed. Chemical processes to metabolize drugs are oxidation, hydrolysis, reduction, hydration, condensation, and conjugation. Study of these pathways as the route of metabolism of a drug can help to assess if a drug shows any pharmacological, clinical relevance, and toxicological activity. The primary site for drug metabolism is the liver (Wilkinson, 2005).

Pathway Construction: Data Curation Based Approaches

The curation process of biological networks and pathways involves manual or computer aided curation of information from various structured and unstructured sources, followed by their assembly in a centralized repository. Major steps involved in pathway building are as follows. First step in pathway construction is mining of relevant data from available public and private data sources about biological entities and their interactions. The pathway draft thus obtained is further enriched while accounting for cell, species, and disease type annotations. This specific annotated pathway is then reviewed by domain experts and iteratively updated before finalization (Viswanathan *et al.*, 2008) (see Fig. 1 for the steps).

Increasingly, natural language processing based methods are taking over manual curation processes. Such approaches are capable of automatic extraction of event–entity relationships from literature (Ananiadou *et al.*, 2010). This accelerated curation process reduces time and labor in accumulating experimentally validated relationships, which, in turn, are used in pathway construction (Yu *et al.*, 2013).

Among several methods, one of the simplest and embrasive approach is cooccurrence based literature mining (Li *et al.*, 2006). For example, if breast cancer and *BRCA1* gene cooccur in a sentence, a relationship between them is assumed. However, this system is prone to errors and may not accurately capture all true relationships that may be latently explained through a number of experimental results.

The second approach is rule-based systems, which exploits some frequently observed linguistic motifs, often used in citing biologically pertinent facts. Such systems either use hard-coded patterns or semantic analysis to find about classes of biologically nontrivial facts. Related to this, a commonly used approach is to use inferential statistics or machine learning to classify sentences and documents into classes of interests (Cohen and Hunter, 2008).

A more sophisticated approach is to infer biological association networks (BANs) through Medscan based sentence parsing. Medscan is capable of extracting relations between proteins and various cellular processes/diseases. If information regarding the direction of relation, mechanism of action and effect on a target is present in the same sentence, Medscan can easily understand and interpret the same. Also, this algorithm can identify if a same relationship is portrayed differently by different authors, thereby reduces redundancy.

Pathway Databases

To keep up with the expansion of our knowledge about biological pathways, systematic and periodic archival of the pathway related information is essential. Below is a list of the key pathway related databases, heavily referred to in biomedical research.

Databases for metabolic pathway

Metabolic pathway databases generally contain detailed information about pathways connected as series of biochemical reactions. One of the main database for metabolic pathways is MetaCyc (Caspi *et al.*, 2013). The MetaCyc is an intensively curated, nonredundant, and comprehensive database for both primary and secondary metabolic pathways. The statistical details of this database are shown in Table 1. It contains valuable information about metabolites, reactions, and enzymatic data that have been experimentally validated through extensive literature scoping from all domains of life.

Another popular database in this domain is Kyoto Encyclopedia of Genes and Genomes (KEGG), which is well-integrated reference knowledge base for molecular interaction, reactions, and network relations for metabolism (Kanehisa and Goto, 2000). Relevant statistics for KEGG are shown in Table 2.

Table 1 MetaCyC statistics

<i>Feature</i>	<i>Statistics</i>
Pathways	2,527
Reactions	14,347
Enzymes	11,547
Chemical compounds	14,003
Organisms	2,883
Citations	54,196

Table 2 KEGG statistics

<i>Feature</i>	<i>Pathway information</i>	<i>Statistics</i>
KEGG pathway	Pathway maps	519 (542,037)
KEGG genes	Genes in KEGG organisms and other categories	24,338,758
KEGG compound	Metabolites and other small molecules	18,111
KEGG reaction	Biochemical reactions	10,668
KEGG disease	Human diseases	1,910

Table 3 TRANSFAC statistics for year 2017

<i>Feature</i>	<i>Statistics</i>
Factors	23,800
miRNAs	1,279
DNA sites	49,945
mRNA sites	21,558
Genes	82,975
ChIP TFBS	27,243,261
References	36,911

Human Metabolome Database (HMDB) is an open source containing the most comprehensive curated information about human metabolites and human metabolism data (Wishart *et al.*, 2007). It features over 2180 endogenous metabolites and their details gathered from extensive literature scoping. HMDB also contains metabolite concentration data collected from mass spectra (MS) and Nuclear Magnetic resonance (NMR) data analysis. The HMDB is composed of various tools for browsing, extensive searching, and querying.

Gene regulation databases

Gene regulation databases tend to cover information on relationships between TFs and genes regulated by them. These databases have broad coverage of organisms and some of the databases tend to incorporate DNA binding data from high-throughput assays such as chromatin immunoprecipitation (ChIP). Such protein–DNA binding information only indicates that TFs bind to promoters of genes to regulate them but it doesn't provide functional consequences of such binding (Cary *et al.*, 2005). The most prominent gene regulation databases is TRANSFAC. TRANSFAC is a manually curated database for eukaryotic TFs and their genomic binding sites (Wingender *et al.*, 2000). It is an encyclopedia for transcriptional regulation and is an apt tool for extensive genomic analysis to predict potential transcription factor binding sites (TFBSs). Computationally TFBSs are predicted using basic positional weight matrices, which are involved motifs based on the detected nucleotide patterns in a set of TFBSs for the corresponding TF (Mathelier and Wasserman, 2013). More detailed statistics of TRANSFAC are shown in Table 3. The basic structure of TRANSFAC comprises of two domains, one is involved in documenting TFBSs in promoters and enhancers and another domain is involved in describing TFs (Matys *et al.*, 2006).

TRUSST (transcriptional regulatory relationships unraveled by sentence-based text-mining) is a publicly available manually curated database for human TFs target interactions. Approximately 20 million Medline abstracts have been subjected to sentence-based text-mining approach for manual curation of regulatory networks. TRUSST currently covers 8015 interactions amongst 748 TF genes and 1975 non-TF genes (Han *et al.*, 2015).

Table 4 Signaling pathway databases

Database	URL	Description
Biocarta	http://www.biocarta.com	Database of graphs and annotations about signal pathways
KEGG	http://www.genome.ad.jp/kegg	Manually drawn pathway maps of interaction and reaction network
Reactome	http://www.reactome.org	Curated resource of core reactions and pathways
PID	http://pid.nci.nih.gov	Cellular processes and biomolecular interactions assembled into reliable human signaling pathways
STKE	http://stke.sciencemag.org	Information on the components of cellular signaling pathways and their relations to one another, organized into pathways
AfCS	http://www.signaling-gateway.org	A database of cell signaling and protein interaction in pathways and complex reactions
AMAZE	http://www.amaze.ulb.ac.be	A database to manage, analyze, represent and annotate cell signaling information
BIND	http://www.bind.ca	A data platform binding public and patented sequence, interaction, and related information
DOQCS	http://doqcs.ncbs.res.in	A repository of signaling pathways models, biochemical reaction schemes, rate constants, concentrations, and annotations on the models
SigPath	http://sigpath.org	A database having pathways of dynamic signaling

Signaling pathway databases

Signaling pathway databases contain detailed information about cell signaling molecules and their interactions. Some important signaling pathway databases are discussed here.

BioCarta: BioCarta is the interactive signaling pathway database formed by a group of engineers and scientists, having detailed information of individual molecule. Information provided in BioCarta is not updated so it might be outdated now. Pathways figures are available in the website provided in “see Section Relevant Website” (Nishimura, 2001). **KEGG:** KEGG is a collection of manually drawn pathway maps. This database is easy to download but less comprehensive than BioCarta (Kanehisa and Goto, 2000). **STKE:** STKE is a freely accessible database that has general data of cells and some special signals in cells. STKE offers data access through machine-generated interactive pathway diagrams. The information in database is ordered into generalized and specific pathways (Gough, 2002). **AfCS:** AfCS database provides signaling pathway map and interactions based on molecule interaction (Gilman *et al.*, 2002). **AMAZE:** AMAZE database provides an object-oriented platform, gene regulation entries, cell signaling pathways, and interaction information (Lemer *et al.*, 2004). **DOQCS** and **sigPath** are quantitative signaling pathway databases. **DOQCS** includes rate constants, reaction schemes, concentrations, as well as annotations on the models (Sivakumaran *et al.*, 2003; Campagne *et al.*, 2004).

Some important signaling pathways databases, URLs, and a short description are given in Table 4.

Drug pathway databases

Drug pathway databases contain detailed drug data, drug target information, pharmacological information about drugs, mechanism of action, and metabolism of drugs. Some important signaling pathway databases are discussed here.

DrugBank: DrugBank database is a unique resource of cheminformatics and bioinformatics that combines detailed drug data with complete information on drug targets. Latest updated version (version 5.0.9, released 2017-10-02) of Drugbank contains 10,500 drug entries in which 1737 are approved small molecule drugs, 103 nutraceuticals, 870 approved biotech drugs, and 5023 experimental drugs (Law *et al.*, 2013). **SMPDB** (Small Molecule Pathway DataBase): This is a visual, interactive database that contains more than 30,000 human small molecule pathways. Majority of these pathways are only found in SMPDB. Each small molecule in SMPDB is hyperlinked to complete information in DrugBank and HMDB (Frolkis *et al.*, 2009). **KEGG:** KEGG is an integrated database resource that contains 16 main pathway databases. The health information category of KEGG includes DRUG, DISEASE, ENVIRON, and DGROU databases for drug and disease information (Kanehisa and Goto, 2000). **Transformer:** Transformer is a comprehensive database providing information on transport and transformation of drugs, phase-I and phase-II reactions, alimentary, and traditional Chinese medicine compounds (Hoffmann *et al.*, 2013). **Drug-Path:** Drug-Path is a database generated by KEGG pathway consisting of drug-induced pathway enrichment analysis for drug-induced downregulated and upregulated genes that are coming from datasets of drug-induced gene expression in Connectivity Map. Pipeline of this database relies on information provided by pharmaceutical partners. This provides user-friendly interfaces to download, retrieve, and visualize drug-induced pathway data. In addition, deregulated genes by a given drug are highlighted. The DrugPath database contains information on 2081 drugs, 751 companies, and 722 targets (Zeng *et al.*, 2015).

Other databases

WikiPathways

WikiPathways is a collaborative platform and freely available manually curated public resource for capturing and publicizing models of biological pathways for data curation, visualization, and analysis (Kutmon *et al.*, 2015). It provides you with the

facilities of zoomable pathway viewer, provision for pathway annotations, and convenient hyperlinks to pathways and their components, thus enabling biologists to capture their productive, intuitive mental models of biological pathways without any hassle. The most observable feature of WikiPathways is the pathway page. Each pathway has its own dedicated page providing information about the specific biological mechanism and pathway diagram, along with description and hyperlinks to genes, proteins, or metabolites for detailed information (Kelder *et al.*, 2011). It was launched in 2008 and since then it has seen a massive increase in its content, which is continuously evolving and benefiting the scientific community in several ways (Kutmon *et al.*, 2015; Kelder *et al.*, 2011). Initially, it started out with 500 pathways spanning across 6 species. But today it contains over 2300 pathways spanning over 25 different species. The human pathway coverage is the largest and most active collection by species. This database is quite flexible to use as there are no restrictions on pathway models, therefore accepting any pathway that researchers find valuable in their work. Collection of pathways in WikiPathways are in a format that can be directly and easily used for downstream data analysis using various software tools such as PathVisio and Cytoscape (Kelder *et al.*, 2011). WikiPathways will keep continue to grow, helping researchers across the globe and capturing each and every biological pathway of interest and publicizing it in as many useful ways as possible (Kutmon *et al.*, 2015).

Small molecule pathway database

SMPDB is a freely available manually curated comprehensive, interactive, and visual database intended to carry out clinical omics studies with ease, with specific focus on clinical metabolomics (Frolkis *et al.*, 2009). SMPDB covers over 600 human pathways of small molecule metabolism or processes central to small molecules and nearly 75% of these pathways are not found in other databases such as KEGG, Reactome, and WikiPathways (Jewison *et al.*, 2013). Pathways covered in this database are metabolic pathways, small molecule disease pathways, small molecule drug pathways, and small molecule signaling pathways.

It is the only pathway database that embraces substantial numbers of metabolic disease and drug pathways. SMPDB keeps on adding new information, describing each pathway in the milieu of human physiology and human biochemistry. Also, each drug or metabolite in SMPDB is hyperlinked, which provides detailed information about that particular molecule. In its pathway diagrams, it provides valuable and useful graphical content, which includes the depiction of the organelles, cofactors, and other important cellular features. SMPDB possesses a number of useful, user friendly, and unique features such as use of thumbnail images to facilitate easy pathway viewing and browsing and scrollable table for displaying pathways and pathways synopsis. Although SMPDB work is still in progress, new pathways are constantly being added and existing ones are continuously being improved. Therefore, SMPDB is a useful addition to a collection of already existing pathway databases (Frolkis *et al.*, 2009).

Reactome pathway database

Reactome is an open source of manually curated and peer-reviewed pathway database of human pathways, reactions, and processes. The Reactome data model streamlines the concept of reaction by taking into consideration transformations of different biological entities such as proteins, nucleic acids, and macromolecular complexes. These transformations could be the transport of biological entities from one compartment to another and formation of complexes. Thus, streamlining of Reactome reaction model allows capturing a range of biological processes that span across signaling, metabolic, transcriptional regulation, and apoptosis in a single integrated platform in a computationally navigated format. Reactome is a complete and comprehensive resource of human pathways for facilitating basic research, genomic analysis, pathway modeling, and analysis (Croft *et al.*, 2010).

Pathway Construction: Quantitative Approaches

For obvious reasons curation based approaches lack comprehensiveness. Use of computational and mathematical models is therefore inevitable. Quantitative approaches to model biological pathways can be broadly split into three categories:

- a. Network-based methods
- b. Mathematical modeling

Below we elaborate on the above mentioned categories.

Network-Based Methods

Network-based methods use graph theoretical representation of molecules and their interactions. Nodes in such graphs denote molecules, whereas edges denote their regulatory relationships. Probabilistic graph models are a commonly used technique for modeling molecular interdependencies as observed from genome wide molecular profiling data. Gene expression data is commonly used for such modeling.

Gene coexpression networks (GCN) are progressively being used to model the coexpression relationships among genes. Within a GCN, genes are represented by nodes and edges exists between pair of genes when their expression profiles are significantly correlated across a set of expression-measurement samples (Ficklin *et al.*, 2017). Coexpression analysis network identifies set of genes that have tendency to display a synchronized expression pattern across a group of samples. Coexpression network

construction involves use of different measures for covariability. Pearson's/Spearman's correlation coefficients, mutual information, etc., are popular among these. Coexpression network construction is often followed by identification of community of coexpressed genes using different clustering techniques (van Dam *et al.*, 2017).

Mathematical Modeling

Mathematical modeling of biological processes arises with deep knowledge of complex cellular systems (Wang *et al.*, 2012). Mathematical models learn and analyze the principal network and then convert the reactions and entities of underlying network into matrix form. To study biological pathways, complexity of networks, and different type of biochemical reactions, a number of mathematical methods have been developed (Hou *et al.*, 2015). Below we provide an excerpt of these methods for modeling signaling, regulatory, and metabolic pathways respectively.

Signaling pathways: Signaling pathways are studied in either top-down or bottom-up fashion. Top-down approaches work on omics datasets to infer downstream biological processes. On the other hand, bottom-up approaches are used to model the interaction between the biological components such as proteins, genes, and metabolites to learn about the dynamic behavior of the cell. The most widely used bottom-up approach is continuous dynamic modeling, which needs adequate kinetic parameters (synthesis and degradation rates) and mechanistic details. Discrete dynamic modeling, for example, Petri nets, Boolean network models, and multivalued logical models yield qualitative dynamic illustration of a system behavior that does not need kinetic parameters (Liu and Betterton, 2012; Wang *et al.*, 2012).

Regulatory network: Understanding the dynamism of regulatory networks is critical in understanding the mechanism of diseases that occur due to dysregulation of these regulatory networks (Karlebach and Shamir, 2008). Differential equations can be used to represent these regulatory networks, in which interactions among proteins, mRNAs are defined in terms of rate equations. As the system evolves, these rate equations specify the levels of each mRNA and protein as a function of other components. Differential equation based models use time and/or space dependent variables (mRNA and protein concentration, production rate, degradation rate).

These models can be divided into two groups:

- Ordinary differential equations (ODE), which involve a single variable such as time.
- Partial differential equations (PDE), which depend on more than one variable such as time and space.

Finding analytical solutions for ODEs is hard. Approximate solutions can be found by various approximate numerical methods (Ay and Arnosti, 2011).

Differential equations have been used in modeling regulatory networks in various organisms:

- In bacteriophage T7 developmental cycle (Endy *et al.*, 2000)
- Tryptophan synthesis in *E. coli* (Santillán and Mackey, 2001)
- In *E. coli* lac operon induction (Carrier and Keasling, 1999)
- Cell division in *Xenopus* (Novak and Tyson, 1993)
- Circadian rhythms in *Drosophila* (Leloup and Goldbeter, 1998)
- In lytic and lysogenic cycle of bacteriophage (McAdams and Shapiro, 1995)

Metabolic pathway: Mathematical interpretations of protein and gene expression, developmental biology, and enzyme kinetics are required to understand the control system that underlies metabolic flux. To design a mathematical model for metabolic pathways, dynamic mathematical models of metabolic networks and stoichiometric models and flux analysis are used. Dynamic models are designed to predict how metabolism will respond to genetics as well as environmental manipulations. Stoichiometric models and flux analysis are largely used and descriptive models, which gives the complete description of metabolism. The former is a purely descriptive tool that yields detailed quantitative snapshots of metabolism that could be employed to gain insight into physiology. The latter class of tools (e.g., dynamic models) move from the realm of description to potential prediction as to how metabolism will respond to manipulation, either environmental or genetic. Flux balance analysis (FBA) is a control based modeling method that can be used to produce computational models by integrating transcriptomic data into metabolic network regenerations. Metabolic flux analysis (MFA) is a well-known method to study metabolic flux in microorganisms but it is still a challenge to adapt these methods for a complex eukaryotic system. MFA has been used to study how various factors (genetic or environmental) and conditions can affect cellular metabolism in fungi. In mammalian cells, MFA is also used to in medical studies, toxicological research, and cell culture techniques.

A flowchart for pathway modeling is given in Fig. 2.

Some Common Applications of Pathways

Pathway analysis methods possess a wide range of applications in biomedical research. They help researchers to figure out important biomolecules, which are key in understanding the phenomena under study. Pathways also help in pathology specific determination of biological function of candidate genes, of which many have therapeutic value. Another application of pathway analysis is that it helps in the determination of functional similarities and dissimilarities between samples based on differential molecular abundance. In summary, pathway analysis is an indispensable tool for data driven sectors in biomedical and cell sciences.

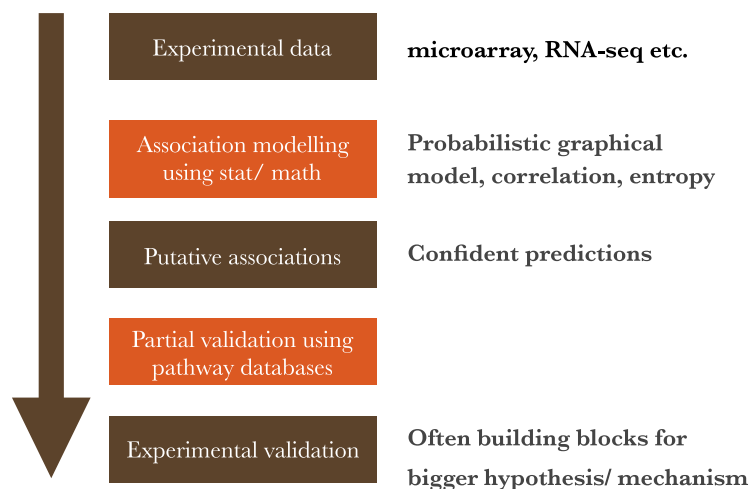


Fig. 2 Template for quantitative approaches for pathway construction.

Overrepresentation Analysis

Overrepresentation analysis (ORA) or functional enrichment analysis approach is routinely used in functional genomic studies to identify overrepresented pathways with a list of interesting or susceptible genes, genes that are likely to increase the likelihood of developing particular disease by using traditional statistical tests such as Fisher's exact test for contingency table (Jin *et al.*, 2014). ORA involves creating an input gene list based on certain criteria, which is followed by counting the number of genes in the list of susceptible genes that hit the gene set annotated by Gene Ontology, KEGG, BioCarta, and Reactome. This is followed by assessing the significance of overlaps using traditional statistical tests and finding out relevant pathways based on their *p* values. Although it is a widely used method there are certain limitations, in that it does not take into account complex gene–gene interactions and secondly it assumes that each gene is of equal importance, which is not the case in biology (Dong *et al.*, 2016). It is not suitable to use ORA when genes to be analyzed are very large as it is based on a stringent threshold. Its statistical power is low when GWAS data is used for identification of significant genes (Jin *et al.*, 2014).

Gene Set-Based Scoring

Gene set-based scoring methods are an extension of ORA but they don't assume that each individual gene is of equal importance; rather they rank these genes by some statistical criteria or *p* values. They are different from ORA in two ways. Firstly, they remove criteria of using an arbitrary threshold as used by ORA and thus consider all of the genes in an experiment. Secondly, they permute phenotype class labels to assess the significance and they allow preserving of gene–gene correlation, thus resulting in a more accurate null model. Statistical tests such as Wilcoxon rank sum or Kolmogorov–Smirnov can be used to know the overall impact of a gene set on biological phenotypes (Jin *et al.*, 2014).

One of the applications of gene set-based scoring is in gene set enrichment analysis (GSEA), a powerful method for inferring results of gene expression data at a level of gene sets. This method is quite powerful since it focuses on gene sets, i.e., genes sharing common biological functions chromosomal location or regulation. Gene sets are defined based on prior biological knowledge such as published biochemical pathway data or other coexpression experimental data. Ranking of genes is based on the correlation between their expression and class distinction can be done by using any suitable metric. The aim of GSEA is to determine whether the majority of genes from a gene set falls towards the top or bottom of the list and in which case the gene set can be correlated with the disease phenotypes. Main steps involved in GSEA are:

- Calculation of enrichment score (ES): ES represents degree to which gene set is overrepresented at the extremes (top or bottom) of the entire ranked list. This score corresponds to weighted Kolmogorov–Smirnov like statistics.
- Estimation of significance level of ES: Statistical significance is estimated by empirical phenotypic based permutation test procedures in order to generate a null distribution for the ES.
- Adjustment for multiple hypothesis testing: When a large number of gene sets are being analyzed at one time, ES for each gene set is normalized, and false discovery rate is calculated (Subramanian *et al.*, 2005).

Network Topology-Based Analysis

Topological analysis is involved in identification of global qualitative properties of the biological system. Most methods in topological analysis are developed for analyzing gene expression data, and they virtually can be extended to other data types (e.g., GWAS)

fairly easily. They combine conventionally measured molecular data and also the structural information of the pathway provided by biological databases. Classical graph theory is used by one approach to identify several motifs, which are groups of interacting biological entities capable of processing information that occurs repeatedly in a pathway and are represented as a directed graph. Boolean network analysis can be used to identify feedback loops, i.e., impact of negative or positive regulations of each interaction. Bayesian networks can also be used for inferring indirect relationships and studying complex cellular networks from quantitative experimental data (Viswanathan *et al.*, 2008). Two algorithms based on topological based analysis are:

ClipPER: ClipPER is an empirical two-step method based on Gaussian graphical models that tries to identify signal paths within significantly altered pathways. First step in ClipPER is testing of the whole pathway. Under specific conditions, strength of molecular interactions within a pathway could be transformed (Martini *et al.*, 2012). The P value of the test defining whether two graphical Gaussian models (cases and control) are homoscedastic as the weight is collected to compute the relevance of each path (Jin *et al.*, 2014). Second step involves identification within these pathways the signal paths having the greatest association with a specific phenotype (Martini *et al.*, 2012).

Signaling Pathway Impact Analysis (SPIA): SPIA is topology-based pathway analysis method that combines two types of evidence, classical enrichment analysis and actual perturbation on a given pathway (Tarca *et al.*, 2008). It is involved in the capturing of several features of data such as changes in gene expression, the pathway enrichment, and the topology of signaling pathways (Jin *et al.*, 2014). In order to assess the significance of the observed total pathway perturbation, a bootstrap procedure is used (Tarca *et al.*, 2008). This method models signaling pathways as a graph where nodes are represented by genes and edges represent interactions (Jin *et al.*, 2014). Furthermore, it defines a gene-level statistics and pathway level statistics, which is called perturbation factor (PF). The PF for a gene is defined as a sum of its measured change in expression and a linear function of the PFs of all genes in a pathway. PF in case of pathway is defined as a sum of PFs of all genes in a pathway (Tarca *et al.*, 2008).

Conclusion

Pathway analysis has emerged as an indispensable tool in functional genomics studies. It is routinely being used in generating biological hypothesis testing on the premise of quantitative molecular investigations. With the accumulation of underanalyzed genomic data deluge an unprecedented opportunity has been created to accelerate pathway research. The role of big consortia is felt in managing goal oriented crowd-sourced projects for expanding and crystallizing our knowledge about biological pathways.

Also, for a better interoperability of multiple software systems and data types, pathway databases must adopt common data standards and formats as it is highly desirable to combine data from different pathway databases. This will allow better coverage of all the reactions and biological entities involved in a pathway (Bauer-Mehren *et al.*, 2009).

In summary, advancement in the field of pathway informatics together with next-generation sequencing technologies presents an unparalleled opportunity for fully exploiting the power of pathway database and pathway analysis resources, tools, and software for understanding cell and disease biology better.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Alignment of Protein-Protein Interaction Networks. Biological and Medical Ontologies: GO and GOA. Biological Pathway Analysis. Biological Pathway Data Formats and Standards. Biological Pathways. Cell Modeling and Simulation. Cluster Analysis of Biological Networks. Community Detection in Biological Networks. Computational Approaches for Modeling Signal Transduction Networks. Computational Systems Biology Applications. Computational Systems Biology. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Functional Enrichment Analysis Methods. Gene Regulatory Network Review. Gene-Gene Interactions: An Essential Component to Modeling Complexity for Precision Medicine. Graph Algorithms. Graph Isomorphism. Graph Theory and Definitions. Graphlets and Motifs in Biological Networks. Investigating Metabolic Pathways and Networks. Metabolic Models. Metabolic Profiling. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Network Centralities and Node Ranking. Network Inference and Reconstruction in Bioinformatics. Network Models. Network Properties. Network Topology. Network-Based Analysis for Biological Discovery. Network-Based Analysis of Host-Pathogen Interactions. Networks in Biology. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein-Protein Interaction Databases. Protein-Protein Interactions: An Overview. Quantitative Modelling Approaches. Transcriptome Analysis. Two Decades of Biological Pathway Databases: Results and Challenges. Visualization of Biological Pathways. Visualization of Biomedical Networks

References

- Ananiadou, S., Pyysalo, S., Tsujii, J.I., Kell, D.B., 2010. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology* 28 (7), 381–390.
- Ay, A., Arnosti, D.N., 2011. Mathematical modeling of gene expression: A guide for the perplexed biologist. *Critical Reviews in Biochemistry and Molecular Biology* 46 (2), 137–151.
- Bauer-Mehren, A., Furlong, L.I., Sanz, F., 2009. Pathway databases and tools for their exploitation: Benefits, current limitations and challenges. *Molecular Systems Biology* 5 (1), 290.
- Berg, J.M., Tymoczko, J.L., Stryer, L., 2002. Signal-transduction pathways: An introduction to information metabolism. *Biochemistry*, 395–424.
- Berridge, M.J., 2014. Module 7: Cellular processes. *Cell Signalling Biology* 6, csb0001007.

- Campagne, F., Neves, S., Chang, C.W., *et al.*, 2004. Quantitative information management for the biochemical computation of cellular networks. *Science STKE*. 2004 (248), p11.
- Carrier, T.A., Keasling, J.D., 1999. Investigating autocatalytic gene expression systems through mechanistic modeling. *Journal of Theoretical Biology* 201 (1), 25–36.
- Cary, M.P., Bader, G.D., Sander, C., 2005. Pathway information for systems biology. *FEBS Letters* 579 (8), 1815–1820.
- Caspi, R., Altman, T., Billington, R., *et al.*, 2013. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research* 42 (D1), D459–D471.
- Chan, S.Y., Loscalzo, J., 2012. The emerging paradigm of network medicine in the study of human disease. *Circulation Research* 111 (3), 359–374.
- Cohen, K.B., Hunter, L., 2008. Getting started in text mining. *PLOS Computational Biology* 4 (1), e20.
- Croft, D., O'Kelly, G., Wu, G., *et al.*, 2010. Reactome: A database of reactions, pathways and biological processes. *Nucleic Acids Research* 39 (Suppl. 1), D691–D697.
- Dong, X., Hao, Y., Wang, X., Tian, W., 2016. LEGO: A novel method for gene set over-representation analysis by incorporating network-based gene weights. *Scientific Reports* 6.
- Endy, D., You, L., Yin, J., Molineux, I.J., 2000. Computation, prediction, and experimental tests of fitness for bacteriophage T7 mutants with permuted genomes. *Proceedings of the National Academy of Sciences* 97 (10), 5375–5380.
- Ficklin, S.P., Dunwoodie, L.J., Poehleman, W.L., *et al.*, 2017. Discovering condition-specific gene co-expression patterns using gaussian mixture models: A cancer case study. *Scientific Reports* 7 (1), 8617.
- Frolkis, A., Knox, C., Lim, E., *et al.*, 2009. SMPDB: The small molecule pathway database. *Nucleic Acids Research* 38 (Suppl. 1), D480–D487.
- Gilman, A.G., Simon, M.I., Bourne, H.R., *et al.*, 2002. Overview of the alliance for cellular signaling. *Nature* 420 (6916), 703–706.
- Gough, N.R., 2002. Science's signal transduction knowledge environment. *Annals of the New York Academy of Sciences* 971 (1), 585–587.
- Han, H., Shim, H., Shin, D., *et al.*, 2015. TRRUST: A reference database of human transcriptional regulatory interactions. *Scientific Reports* 5, 11432.
- Hoffmann, M.F., Preissner, S.C., Nickel, J., *et al.*, 2013. The transformer database: Biotransformation of xenobiotics. *Nucleic Acids Research* 42 (D1), D1113–D1117.
- Holoch, D., Moazed, D., 2015. RNA-mediated epigenetic regulation of gene expression. *Nature Reviews Genetics* 16 (2), 71–84.
- Hoopes, L., 2008. Introduction to the gene expression and regulation topic room. *National Education* 1 (1), 160.
- Hou, J., Acharya, L., Zhu, D., Cheng, J., 2015. An overview of bioinformatics methods for modeling biological pathways in yeast. *Briefings in Functional Genomics* 15 (2), 95–108.
- Jewison, T., Su, Y., Disfany, F.M., *et al.*, 2013. SMPDB 2.0: Big improvements to the small molecule pathway database. *Nucleic Acids Research* 42 (D1), D478–D484.
- Jin, L., Zuo, X.Y., Su, W.Y., *et al.*, 2014. Pathway-based analysis tools for complex diseases: A review. *Genomics, Proteomics & Bioinformatics* 12 (5), 210–220.
- Joyner, M.J., Pedersen, B.K., 2011. Ten questions about systems biology. *The Journal of Physiology* 589 (5), 1017–1030.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Karlebach, G., Shamir, R., 2008. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology* 9 (10), 770–780.
- Karp, P.D., 2003. Pathway bioinformatics. In: CSB, p. 27.
- Kelder, T., van Iersel, M.P., Hanspers, K., *et al.*, 2011. WikiPathways: Building research communities on biological pathways. *Nucleic Acids Research* 40 (D1), D1301–D1307.
- Kutmon, M., Riutta, A., Nunes, N., *et al.*, 2015. WikiPathways: Capturing the full diversity of pathway knowledge. *Nucleic Acids Research* 44 (D1), D488–D494.
- Law, V., Knox, C., Djombou, Y., *et al.*, 2013. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Research* 42 (D1), D1091–D1097.
- Leloup, J.C., Goldbeter, A., 1998. A model for circadian rhythms in *Drosophila* incorporating the formation of a complex between the PER and TIM proteins. *Journal of Biological Rhythms* 13 (1), 70–87.
- Lerner, C., Antezana, E., Couche, F., *et al.*, 2004. The aMAZE LightBench: A web interface to a relational database of cellular processes. *Nucleic Acids Research* 32 (Suppl. 1), D443–D448.
- Li, S., Wu, L., Zhang, Z., 2006. Constructing biological networks through combined literature mining and microarray analysis: A LMMMA approach. *Bioinformatics* 22 (17), 2143–2150.
- Liu, W., Li, D., Zhu, Y., He, F., 2008. Bioinformatics analyses for signal transduction networks. *Science in China Series C: Life Sciences* 51 (11), 994–1002.
- Liu, X., Betterton, M.D. (Eds.), 2012. *Computational Modeling of Signaling Networks*. Humana Press.
- Ma, H., Zhao, H., 2013. Drug target inference through pathway analysis of genomics data. *Advanced Drug Delivery Reviews* 65 (7), 966–972.
- Martini, P., Sales, G., Massa, M.S., Chiogna, M., Romualdi, C., 2012. Along signal paths: An empirical gene set approach exploiting pathway topology. *Nucleic Acids Research* 41 (1), e19.
- Mathelier, A., Wasserman, W.W., 2013. The next generation of transcription factor binding site prediction. *PLOS Computational Biology* 9 (9), e1003214.
- Matys, V., Kel-Margoulis, O.V., Fricke, E., *et al.*, 2006. TRANSFAC® and its module TRANSCOMP®: Transcriptional gene regulation in eukaryotes. *Nucleic Acids Research* 34 (Suppl. 1), D108–D110.
- McAdams, H.H., Shapiro, L., 1995. Circuit simulation of genetic networks. *Science* 269 (5224), 650–656.
- Nishimura, D., 2001. BioCarta. Biotech Software & Internet Report: The Computer Software Journal for Scientific 2 (3), 117–120.
- Novak, B., Tyson, J.J., 1993. Modeling the cell division cycle: M-phase trigger, oscillations, and size control. *Journal of Theoretical Biology* 165 (1), 101–134.
- Pawson, T., 1995. Protein modules and signalling networks. *Nature* 373 (6515), 573–580.
- Santillán, M., Mackey, M.C., 2001. Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data. *Proceedings of the National Academy of Sciences* 98 (4), 1364–1369.
- Sengupta, D., Bandyopadhyay, S., 2011. Participation of microRNAs in human interactome: Extraction of microRNA–microRNA regulations. *Molecular Biosystems* 7 (6), 1966–1973.
- Sengupta, D., Bandyopadhyay, S., 2013. Topological patterns in microRNA–gene regulatory network: Studies in colorectal and breast cancer. *Molecular Biosystems* 9 (6), 1360–1371.
- Sivakumaran, S., Hariharaputran, S., Mishra, J., Bhalla, U.S., 2003. The database of quantitative cellular signaling: Management and analysis of chemical kinetic models of signaling networks. *Bioinformatics* 19 (3), 408–415.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* 102 (43), 15545–15550.
- Tarca, A.L., Draghici, S., Khatri, P., *et al.*, 2008. A novel signaling pathway impact analysis. *Bioinformatics* 25 (1), 75–82.
- van Dam, S., Vösa, U., van der Graaf, A., Franke, L., de Magalhães, J.P., 2017. Gene co-expression analysis for functional classification and gene–disease predictions. *Briefings in Bioinformatics*. bbw139.
- Viswanathan, G.A., Seto, J., Patil, S., Nudelman, G., Sealfon, S.C., 2008. Getting started in biological pathway construction and analysis. *PLOS Computational Biology* 4 (2), e16.
- Wang, R.S., Saadatpour, A., Albert, R., 2012. Boolean modeling in systems biology: An overview of methodology and applications. *Physical Biology* 9 (5), 055001.
- Ward, C., 2016. Metabolic pathways [internet]. 2016 Jan 19; Diapedia 5105765817 rev. no. 25.
- Wilkinson, G.R., 2005. Drug metabolism and variability among patients in drug response. *New England Journal of Medicine* 352 (21), 2211–2221.
- Wingender, E., Chen, X., Hehl, R., *et al.*, 2000. TRANSFAC: An integrated system for gene expression regulation. *Nucleic Acids Research* 28 (1), 316–319.
- Wishart, D.S., Tzur, D., Knox, C., *et al.*, 2007. HMDB: The human metabolome database. *Nucleic Acids Research* 35 (Suppl. 1), D521–D526.
- Yu, D., Kim, M., Xiao, G., Hwang, T.H., 2013. Review of biological network data and its applications. *Genomics & Informatics* 11 (4), 200–210.
- Zeng, H., Qiu, C., Cui, Q., 2015. Drug-Path: A database for drug-induced pathways. *Database* 2015.

Relevant Website

http://cgap.nci.nih.gov/Pathways/BioCarta_Pathways
BioCarta_Pathways.

Network Inference and Reconstruction in Bioinformatics

Paolo Tieri, CNR National Research Council, IAC Institute for Applied Computing “Mauro Picone”, Rome, Italy

Lorenzo Farina, Department of Computer, Control and Management Engineering “A. Ruberti”, Sapienza University of Rome, Italy and CNR National Research Council, IASI Institute of Systems Analysis and Computer Science “Antonio Ruberti”, Rome, Italy

Manuela Petti, Department of Computer, Control and Management Engineering “A. Ruberti”, Sapienza University of Rome, Italy and Neuroelectrical Imaging and BCI Lab, Fondazione Santa Lucia IRCSS, Rome, Italy

Laura Astolfi, Department of Computer, Control and Management Engineering “A. Ruberti”, Sapienza University of Rome, Italy and Neuroelectrical Imaging and BCI Lab, Fondazione Santa Lucia IRCSS, Rome, Italy

Paola Paci, CNR National Research Council, IASI Institute of Systems Analysis and Computer Science “Antonio Ruberti”, Rome, Italy

Filippo Castiglione, National Research Council of Italy, Rome, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological networks inference, also referred to as ‘reverse engineering’, is the scientific process of using (low- and high-throughput) experimental data, statistical and computational techniques to reconstruct how the elements of the biological network (genes, proteins, signaling molecules, cells) interact and operate as a system.

The paradigm of ‘network’ is crucial for the sciences of complexity, as well as for systems biology. This concept represents a potent instrument for the description and analysis of complex systems, their elements and their connections, able to detect underlying topologies, structures, architectures as well as functions emerging from the arrangement of their components. Such methodology has been fruitfully employed for the representation and for the analysis of numerous different systems in various fields of study, e.g., engineering and technology, social studies and, of course, life sciences.

The power of the network approach basically resides in the capability to catch specific features of any type of systems, e.g., steady and/or physically connected systems (such as the physical internet cabling, or power grids) as well as dynamic, virtual and non-cabled (such as social networks, air traffic networks, or gene/protein interactions).

In bioinformatics, such interdisciplinary approach allowed to undertake, probably for the first time, rigorous mathematical analysis of larger portions of, or even whole biological systems.

A critical step is therefore the process of reconstruction of such networks starting from accessible data. Here, making no claim to completeness, we briefly outline inference and reconstruction approaches of some among the most relevant types of biological networks in molecular biology and neuroscience, and some of the most promising methodologies applied in the recent field of network medicine.

Molecular Networks Inference

Protein–Protein Interaction Networks

Protein-protein interactions (PPIs) are highly specific physical contacts, stable or transient in time, between two or more proteins. PPIs involve electrostatic forces between the so-called protein surfaces, i.e. the ‘exposed’ regions of the three-dimensional structures of folded proteins. PPIs are essential to almost every process in a biological system, e.g., in signal transduction, cell metabolism, membrane transport, muscle contraction, among others, and consequently their mapping and understanding is critical for the comprehension of cellular physiology in normal and as well as in pathological conditions.

The representation of interconnected webs of PPIs at the cellular level, or, in other words, the union of all considered proteins and the interactions among them, are generally referred to as protein interaction networks (PINs). Such networks are usually represented with undirected graphs and undergo mathematical and computational analysis in order to study cell processes, functions and organization at the system level.

PINs reconstruction is carried using several experimental (in vivo and in vitro) and computational methods, involving specific and different pros and cons (Rao *et al.*, 2014; Cafarelli *et al.*, 2017).

Among the most widely known experimental approaches, Yeast 2 hybrid (Y2H) is an *in vivo* method, suitable for both transient and stable interactions. It is implemented by screening the protein of interest against a library of potential interacting protein partners through the activation of a reporter gene (usually the gene *Gal4* in the yeast *Saccharomyces cerevisiae*). Y2H is affordable, fast and low-tech, but recently it has been criticized as not completely reliable in terms of yielding high numbers of false positive identifications. Y2H is the only *in vivo* method based on the detection of real physical interactions, being the other *in vivo* one, namely synthetic lethality, based on functional interactions, i.e. using deletions or mutations in genes to observe protein interactions.

Among the most used *in vitro* techniques, tandem affinity purification-mass spectroscopy (TAP-MS) is based on labelling the protein under scrutiny on its chromosomal locus with a designed protein tag, and on a two-step purification method followed by mass spectroscopic analysis. It is usually considered suitable for stable interactions only (due to the purification steps), simple to implement and quantitatively reliable. Other approaches include protein microarrays, affinity chromatography, coimmunoprecipitation, X-ray crystallography or NMR spectroscopy, among others.

Among the *in silico* methods, the ortholog-based sequence approach is a commonly used technique: it is grounded on the homologous nature of the query protein in the annotated protein databases using pairwise local sequence algorithm. Other *in silico* approaches are based on protein sequence (as the summentioned ortholog-based one) or protein structure, on gene fusion or gene expression, on chromosomal proximity, on phylogenetic tree or gene ontology.

Methods for the assessment of PPIs, among other approaches, can be implemented computationally e.g., by using specific reliability indexes such as 'expression profile reliability' (EPR) index (used to assess whole datasets by comparing the RNA expression profiles for the proteins whose interactions are found in the screen with expression profiles for known interacting and non-interacting pairs of proteins) or the paralogous verification method (PVM, used to score individual interactions), which judges an interaction likely if the putatively interacting pair has paralogs that also interact. PPI validation can be also implemented by using known macromolecular complexes that provide defined and unbiased set of protein interactions (Przulj, 2011). It is foreseen that prospective PIN reconstruction attempts will focus on the dynamic and quantitative properties, surpassing current stationary representations (Cafarelli *et al.*, 2017).

Gene Regulatory Networks

Gene regulatory networks (GRNs, also known as transcriptional regulatory networks) are networks of causal interactions among transcription factors and downstream genes, and are usually represented with directed graphs and inferred by gene expression data. Links between elements of a GRN represent biochemical process such as a reaction, transformation, interaction, activation or inhibition. GRNs are a crucial area of systems biology research, providing understanding of the inward mechanisms of a biological system.

Many methods to infer GRNs are based on mutual information (MI) (e.g., MI with background, maximal MI, conditional MI, three-way MI), a probabilistic concept and dimensionless quantity that states how much the knowledge of a random variable tells about another one: in other words, mutual information can be thought as the decrease in uncertainty of a random variable once given knowledge of another, so that mutual information equal to zero between two random variables means that the variables are completely independent (nothing can be said about a variable given the other), and increases of mutual information denote decreases in such uncertainty.

Inferred GRNs represent causal biochemical interactions between genes, RNA and proteins which links aim to correspond to real physical, directed, and quantitatively determined interaction events between such molecules that can potentially lead to the discovery of, for example, crucial functional relationships between RNA expression and chemotherapeutic susceptibility (Butte *et al.*, 2000). Recently, data from single-cell gene expression have become mature and have been approached using partial information decomposition to detect putative functional associations and to formulate systematic hypotheses (Chan *et al.*, 2017; Aibar *et al.*, 2017).

The validation of GRNs implies the definition of 'gold standards' (i.e. sets of interactions validated with a given level of certainty) and the use statistical thresholds to quantitatively evaluate the adherence of the GRN to such standards, or perturbation analysis of the biological (sub)system to observe and measure perturbation effects and subsequently validate the network under scrutiny (Chai *et al.*, 2014; Hecker *et al.*, 2009; Emmert-Streib *et al.*, 2014; Veiga *et al.*, 2010; Lee and Tzou, 2009; Bansal *et al.*, 2007).

Gene Co-Expression Networks

Gene co-expression networks (GCNs) are transcript–transcript association networks, generally reported as undirected graphs, where genes are connected when an appreciable co-expression association between them exists.

GCNs are built from gene expression data by calculating co-expression values in terms of pairwise gene similarity score and choosing a significance threshold. Debates on normalization methods, co-expression correlation (e.g., based on Pearson's or Spearman's correlation measures, among others) and significance and relevance are still alive and ongoing. Graphical Gaussian Models (also said 'concentration graph' or 'covariance selection' models) are also popular in the field of GCNs, the key idea behind them being the use of partial correlations, able to discriminate between direct and indirect interactions, as a degree of independence of any pair of genes (Schafer and Strimmer, 2005). Other strategies for GCN inference include edge removal based on gene triplets analysis (e.g., ARACNE) (Margolin *et al.*, 2006), regression methods and Bayesian networks (aimed to discover the best gene set predictor of a target gene expression).

GCNs are a potent approach to gather biologically relevant information, e.g., for the identification of genes not yet associated with explicit biological questions, and for accelerating the interpretation of molecular mechanisms at the root of significant biological processes. Trends in this field include, among other approaches, the combination of co-expression analysis with other omics techniques, such as metabolomics, for estimating the coordinated behavior between gene expression and metabolites, as well as for assessing metabolite-regulated genetic networks (Serin *et al.*, 2016; Usadel *et al.*, 2009).

Metabolic Networks

Metabolism, apart from obviously being the set of life-sustaining biochemical reactions in every organisms, is also among the major contributors to many human diseases, such as cardiovascular diseases, obesity, diabetes and cancer, to cite some. Metabolic network reconstruction is commonly referred to as the annotation process of genes and metabolites for the determination of the metabolic network's elements, relationships, structure and dynamics. It is generally possible to infer the enzymatic function of individual proteins, or to reconstruct larger (or whole) metabolic networks in their entirety.

Originally introduced at the beginning of 1900, the first successful attempts to quantitatively reconstruct small metabolic networks, involving linear sequences of few reactions, date back to the Eighties of the past century. Later on, coordinated efforts led to the reconstruction of the global human metabolic network, based on recent genome annotation and on a large amount of bibliomic data (Mo *et al.*, 2007). Today, the main aim is usually the reconstruction of genome-scale metabolic networks, and techniques such as metabolic flux analysis (MFA) and its enhancements (e.g., isotopically nonstationary metabolic flux analysis), and flux balance analysis (FBA) has become widely used for the simultaneous predictions of fluxes of multiple reactions. The methods are generally based on stable isotope-labeled substrates and measuring time courses of intracellular metabolites labels. FBA is able to generate flux predictions without prior information of enzyme-kinetic parameters, based on optimization principles that theoretically should have shaped efficiency (and fluxes) of metabolic networks. Integrated computational approaches coupling metabolic flux analysis with mass spectrometry approaches have been also recently used. From the computational side, single enzyme function prediction is carried out by using machine learning (e.g., when the enzyme does not show significant similarity to existing proteins from which predictions can be made) or the so-called ‘annotation transfer’ approaches (based on DNA sequences or reference databases or orthologs, to predict the occurrence of enzymes). Comparative pathway prediction techniques generally use existing functional annotations to check for the presence of new reactions, while explorative pathway prediction approaches, i.e. without the use of existing annotations, can be graph-theoretic (e.g., by weighting paths of metabolite connectivity) or constraint-based (e.g., elementary mode analysis), or both (Pitkanen *et al.*, 2010; Nikoloski *et al.*, 2015).

Signaling Networks

Signaling networks are cascades of molecular interactions and chemical modifications to carry stimuli (e.g., signaling molecules, hormones, pathogens, nutrients) sensed by cell membrane receptors down to the nucleus to coordinate initiate proper metabolic and genetic responses. The processes of identifying signaling networks constituents and of network reconstruction have been often carried out by using gene knockout techniques, which have been effective in describing cascades in a linear manner. Nevertheless, the complexity and intricate nature of such networks require new methods that can infer aspects and features of signaling processes from high-throughput omic data in a faster and systemic way. In this perspective, efficient algorithms are necessary to analyze and detect relations among numerous data points, and in general such inference problems can be reported to the definition of suitable optimal connected subgraphs of a network originally defined by the available data, which can result in being computationally intractable. Some of the approaches employ Steiner tree approaches (e.g., based on the determination of the shortest total lengths of paths of interacting proteins), or linear programming and maximum-likelihood (e.g., based on tagging proteins as activators or repressors to explain the maximum number of observed gene knockout); other use probabilistic network approaches, such as network flow optimization (e.g., based on Bayesian weighting schemes for underlying PPIs coupled with other omic data), or network propagation (e.g., based on a gene prioritization function that scores the strength-of-association of proteins with a given disease), or information flow analysis (a computational approach that identifies proteins dominant in the communication of biological information across the network) (Tuncbag *et al.*, 2013; Ourfali *et al.*, 2007; Missiuro *et al.*, 2009; Huang and Fraenkel, 2009; Aldridge *et al.*, 2006; Papin *et al.*, 2005; Ritz *et al.*, 2016; Molinelli *et al.*, 2013; Hyduke and Palsson, 2010)

Assessment and Validation of Inferred Molecular Networks

The methods to reconstruct biological networks as well as inferred networks need to undergo rigorous validation and assessment processes based on theoretical points, on *in silico* experiments, and on ‘wet lab’ data. Known issues about assessment approaches include the volume of information accessible related to the network entities, the general sparseness of biological networks as well as their topology, and, often neglected, the lack of experimentally verified non-interacting pairs. Some general considerations arise, applicable to the different types of networks and of methods. As a general procedure, network inference methods have been often assessed using receiver operating characteristic (ROC) curves and precision-recall (PR) curves: being them differently sensitive to the ratio between (true/false/actual/predicted) positives and negatives among the tested pairs, ROC curves seem to show and advantage in comparing different classification methods (Schrynmackers *et al.*, 2013). Another general consideration regards the relevance of the lack of experimental support for non-interacting pairs in most biological networks, especially in the field of PPIs, lack that can heavily impinge upon the assessment of false positives, issue that is recently receiving more attention (Blohm *et al.*, 2014; Smialowski *et al.*, 2010; Launay *et al.*, 2017; Srivastava *et al.*, 2016). Some specific measures have been used to assess inferential validity for GRNs, such as the Hamming distance (e.g., to measure the network topology closeness), or steady-state mass difference for network dynamic behavior similarity (Qian and Dougherty, 2013). As a general consideration, the challenging nature of network inference tasks should be highlighted, as a remarkable amount of the evaluated algorithms seems to not accomplish better results than random (Lopes and Bontempi, 2013; De Smet and Marchal, 2010), but it is also recognized that despite inferred networks might not be error-free at the level of mechanistic description, they can still be fruitfully used to comprehend emergent functions and behavior of biological systems (Stolovitzky *et al.*, 2007).

Networks in Neuroscience

In neuroscience, network science can be applied to the patterns obtained by studying the brain anatomy and activity at different spatial (molecules, neurons, circuits, whole brain, social behavior) and temporal scales (milliseconds, minutes, hour, days, years). Examples

include networks like those described above (PPI, genetic regulatory networks...) and networks consisting of anatomical or functional connections among brain areas. The concept of functional brain connectivity is crucial to understand how communication between cortical regions is organized: brain areas interacting with a functional connection are not necessarily connected by a direct physical link; on the other hand, an anatomical link does not necessarily imply that a functional connection has established at any time point.

Anatomical connectivity can be tracked by diffusion weighted imaging techniques such as Diffusion Tensor Imaging (DTI): this technique shows anatomical connections of neurons by identifying nerve fiber pathways in the brain.

Brain functional connectivity can be estimated from a wide range of biomedical signals and with different neuroimaging techniques – functional magnetic resonance imaging (fMRI), electroencephalography (EEG), magnetoencephalography (MEG), positron emission tomography (PET). In the following a description of two of the main approaches is provided.

- **Functional magnetic resonance imaging (fMRI)** is a specific magnetic resonance imaging procedure to measure brain activity by detecting associated changes in blood flow: specifically, brain activity is measured through low frequency blood oxygenation level dependent (BOLD) signal in the brain. fMRI technique is characterized by high spatial resolution, but by poor time resolution.
- **Electroencephalography** is a method to record electrical activity of the brain. Scalp EEG data have many advantages: high temporal resolution, non-invasiveness and low-cost. On the other hand, it is characterized by low spatial resolution. An EEG setup can be used in a wide range of pathological conditions and can be easily adapted to different clinical settings. To overcome the limitation of the low spatial resolution, in the past decades, the inverse EEG solutions has received considerable attention: this approach provides estimation of source distributions within the brain that correspond to the scalp-recorded signals. Different methods (e.g., minimum norm estimates (MNE), low resolution electrical tomography (LORETA), beam-forming approaches) have been proposed to solve the EEG inverse problem (Grech *et al.*, 2008). Once the inverse problem for EEG source localization is solved, connectivity estimation can be performed in the source space.

Three classes of methods are commonly used to estimate functional connectivity:

- **Model-based functional connectivity approaches.** Seed based connectivity is a hypothesis-driven method based on a priori decision regarding the regions of interest (ROI). It is usually adopted with fMRI data. *Methods based on Granger causality* (Granger, 1969) also belong to this class. They are defined both in time and in frequency domain, and being based on bivariate or multivariate autoregressive models they allow to reconstruct the direction of information flows. Examples of Granger Causality-based methods are: i) *Partial Directed Coherence* (Baccalá and Sameshima, 2001); ii) *Directed Transfer Function* (Kamiński and Blinowska, 1991); iii) *time reversed Granger Causality* (Haufe, Nikulin, and Nolte, 2012).
- **Model-free functional connectivity approaches.** These methods are especially useful in the analysis of spatially distributed functional connectivity networks. An example is the *imaginary coherence* (Nolte *et al.*, 2004). Coherency between two EEG-channels is a measure of the linear relationship of the two at a specific frequency. The idea is to isolate the imaginary part of the coherency which reflects interactions. This method does not return the information about interaction direction and it has been proposed as a possible solution to the interpretation of observed interactions between time series when they are affected by volume conduction.
- **Biologically Inspired models.** *Dynamic Causal Modelling* (DCM, (Friston *et al.*, 2003)) belongs to this class. DCM is a general framework for inferring processes and mechanisms at the neuronal level from measurements of brain activity with different techniques (fMRI, EEG/MEG, etc.): it combines a model of the hidden neuronal dynamics with a forward model that translates neuronal states into predicted measurements. This method returns directed networks.

In the last decades, several studies demonstrated that by combining neuroimaging techniques (functional MRI, EEG, etc.) with network science, it is possible to characterize the topological properties of human brain networks. Nowadays, network characterization of structural or functional connectivity data is increasing (Bassett and Sporns, 2017; Bullmore and Sporns, 2009) and rests on several important motivations. Indeed, complex network analysis promises to reliably quantify brain networks with a small number of neurobiologically meaningful and easily computable measures (Pichiorri *et al.*, 2017; Petti *et al.*, 2016; Achard and Bullmore, 2007; Toppi *et al.*, 2012).

Network Medicine

Until recently, the investigation of disease etiology, diagnosis and treatment, has been based on a conventional reductionist approach. This tenet argues that critical biological factors work in a simple linear mechanism to control disease pathobiology. Rather, they are nearly always the result of multiple pathobiological pathways that interact through an interconnected network: a disease is rarely a direct consequence of an abnormality in a single gene or molecular component (Chan and LoScalzo, 2012). For example, complex diseases like hypertension, atherosclerosis or cancers of various sorts, have extraordinary complex biological phenomena that underlie them.

Today, big data, genomics, and quantitative *in silico* methodologies integration, have the potential to push forward the frontiers of medicine in an unprecedented way. Clinicians, diagnosticians and therapists have long strived to determine single molecular traits that lead to diseases. What they had in mind was the idea that a single “golden bullet” drug might provide a cure. But, this reductionist approach, largely ignored the essential complexity of human diseases. Indeed, a large body of evidence that is now emerging from new genomic technologies, points out directly to the cause of disease as “perturbations” within the “interactome”, i.e. the comprehensive network map of molecular components and their interactions. The interactome integrates all physical

interactions within a cell, from protein-protein, regulatory protein-DNA and metabolic interactions. Currently, more than 140,000 interactions between more than 13,000 proteins are known.

Consequently, a paradigm shift is needed towards the development of temporal and spatial multi-level models, from molecular machineries to single cells, whole organism and individuals, including the environment, to reveal the underlying links among components. This new type of medical paradigm is called “Network Medicine”. Rather than trying to understand pathogenesis into a reductionist framework, network medicine entwines the many facets of disease in many different types of networks: from the physical interactions acting in a cell to the information flow through biological components. The human genome sequencing using high-throughput next-generation devices is being deeply affecting current conceptions of biomedical and clinical research. More recently, entering the era of personal whole-genome sequencing, 38 million genetic variants some of which are rare mutations and thus may be associated with large size effect.

How to use this big data and information deluge to generate better understanding of disease and find appropriate drug targets? An entirely new perspective is required as a shift from a reductionist to a holistic and data-driven research: looking at networks without a specific biomedical bias in mind and let the data speak by themselves, so to formulate new hypothesis to be further validated by experimentalist and so on, moving within a virtuous circle of shared knowledge. Most importantly, the representation of complex systems as networks is of paramount importance for visualizing the interactome underlying structure, revealing new functional roles, and proposing new and fresh interpretations of data. Networks can be obtained from any sort of information: known protein-protein interactions, gene expression profiles, functional annotation, etc. In fact, the fundamental tenet of network medicine is to look at diseases as perturbations within the ‘interactome’, i.e. the overall network map of molecular components and their interactions. Recently, has been shown the genes associated with a disease are localized in specific neighborhoods, or ‘disease modules’, within such an interactome. The overall ambition of this initiative is to both developing a global understanding of how interactome perturbations result in disease traits, and to translate computational insights into concrete clinical applications, such as new drugs and therapies or diagnostic tools.

A true inter-disciplinary environment for network medicine must be based on the accurate preparation the interactions among experts of different fields, especially from those trained in medical/biological sciences with those experts in data analysis, bioinformatics, statistical modelling, and network algorithms. The gap between the biological and the informational mindset can be daunting and might impair from the beginning the development of shared concepts. However, the network medicine setting will certainly facilitate communications across disciplines given the immediate and intuitive understanding of the “network” concept, a metaphor that can be used by molecular biologists to visualize their knowledge in a structured way ready to be translated into an algorithm on the available data, as medical or biological goals are defined. [Barabási et al. \(2011\)](#) proposed the following research pipeline:

for any specific disease, the identification and validation of disease modules consists of several steps:

- Interactome reconstruction merges the most up-to-date information on protein-protein interactions, co-complex memberships, regulatory interactions and metabolic network maps in the tissue and cell line of interest.
- Disease gene (seed) identification collects the known disease-associated genes obtained from linkage analysis, genome-wide association studies or other sources, which serve as the seed of the disease module.
- In disease module identification, the seed genes are placed on the interactome, with the aim of identifying a subnetwork that contains most of the disease-associated components, exploiting both the functional and topological modularity of the network.
- Pathway identification can be used in instances in which the number of components contained in the ascertained disease module is so large that it cannot serve as a tractable starting point for further experimental work.
- During validation disease modules are tested for their functional and dynamic homogeneity.

Disease Gene Prediction Methodologies

A classification of the disease gene prediction methodologies can be performed following different criteria ([Piro and Di Cunto, 2012](#)):

- Type of evidence: different data sources can be exploited to predict disease-genes (sequence data, protein-protein interactions, gene expression...).
- Level of knowledge: disease-gene prediction methods can be classified based on the use of prior knowledge for example about causal genes for the disease phenotype under investigation or about affected tissues/organs.
- Type of prediction: prediction methods can return a prioritization of the candidate genes (a ranking according to their likelihood of being involved in a disorder) or a selection of best candidates. In the first case the goal is to rank the disease gene as high as possible in the prioritized candidate list, while candidate selection methods provide a candidates’ set (as small as possible).
- Scope of application: based on this criterion, we can distinguish between disease-centered methods and undifferentiated approaches. In the first case, candidates are evaluated in relation to a specific disease (or disorders class), while methods belonging to the second class provide an overall evaluation of the involvement of the candidates in some disease in general.

More recently, the main classification was performed based on network analysis strategy:

- local methods, algorithms based on the search for direct neighbors of disease genes;

- global methods: methods that model the information flow in the cell to assess the proximity and connectivity between known disease genes and candidate genes.

Here we focus on disease-gene prediction methods that are based on the analysis of the topological properties of protein-protein interaction (PPI) networks and we describe with more detail some of the most used and newest approaches at the state of the art.

Local Methods

- **Direct Interaction (DI).** [Oti et al. \(2006\)](#) proposed a procedure to detect disease-genes in known disease loci by exploring the degree to which proteins linked to known disease-genes are also associated with the same phenotype. The algorithm performs the following steps:
 - identification of neighboring proteins for each known disease protein;
 - identification of chromosomal locations of the genes coding for these interacting proteins using gene location data from the Ensembl database;
 - check of the chromosomal locations: each interacting protein gene located within one or more disease loci (of the same disease) was considered a candidate gene prediction;
 - predictions counting: if a candidate gene lay within multiple (overlapping) loci of the same disease, each of them was counted as a separate prediction.
- **GenePANDA (Gene Prioritizing Approach using Network Distance Analysis).** In the last year another algorithm of this class has been proposed: GenePANDA (Gene Prioritizing Approach using Network Distance Analysis) ([Yin et al., 2017](#)). The novelty of GenePANDA is the introduction of adjusted network distance that is derived by considering not only the direct network distance between two genes, but also their respective mean network distances to all other genes in the network. This method uses the STRING network ([Franceschini et al., 2013](#)) and the Genetic Association Database (GAD) as the resource of disease data, and it consists of three steps:
 - adjusted network distance computation: adjusted network distance is computed considering not only the direct network distance between two genes, but also their respective mean network distances to all other genes in the network. Given the genes a and b , their adjusted network distance is defined as:

$$D_{ab}^{adj} = \frac{D_{ab}}{\sqrt{\mu_a \times \mu_b}} \quad (1)$$

where D_{ab} is the raw network distance between gene a and gene b , μ_a and μ_b the mean raw network distances for a and b respectively, defined as:

$$\mu_a = \frac{\sum_{j=1}^N D_{aj}}{N} \quad (2)$$

with N the total number of genes in the network, and D_{aj} the raw network distance between a and b ;

- *disease-specific gene weighting* with the hypothesis that a candidate disease gene should have stronger functional interaction with known disease genes than with random genes in the network. Based on this hypothesis, a disease-specific gene weight was defined as:

$$w_i = \frac{\sum_{j=1}^N D_{ab}^{adj}}{N} - \frac{\sum_{j=1}^K D_{ab}^{adj}}{K} \quad (3)$$

where N is the total number of genes in the network, K is the total number of disease genes. Higher weight indicating higher probability to be a candidate disease gene.

- score conversion: for a given disease, the disease-specific gene weights can be compared with each other related to the same disease, but they cannot be directly compared across diseases. To overcome this limitation, a score conversion procedure was introduced converting the weights into probabilities.

Global Methods

- **Random Walk with Restart (RWR).** [Kohler et al. \(2008\)](#) proposed the random walk analysis for definition of similarity in PPI networks demonstrating that global network-similarity measures (random walk and diffusion kernel) are greatly superior to local distance measures (direct links or shortest paths) in the disease-genes prioritization problem. They performed a random walk on the PPI network, starting at the known disease genes, and rank candidate genes by the steady state probabilities induced by the walk. In the RWR procedure, a random walk starts at one node in the set S of known disease genes (seed genes): the random walker can move to a randomly selected neighbor or it can restart at one of the proteins in S . For each restart, the probability of restarting at a specific protein is r . Formally, RWR is defined as:

$$p^{t+1} = (1 - r)Wp^t + rp^0 \quad (4)$$

where W is the column-normalized adjacency matrix of the graph and p^t is a vector of probability of being at node i at time step t . In Kohler *et al.* (2008), the authors set the initial probability vector p_0 with equal probabilities to the nodes of S , with the sum of the probabilities equal to 1 (disease genes with equal probability).

- **PRINCE (PRIoritizationN and Complex Elucidation)**. To compute the association between disease genes and candidate proteins, Vanunu *et al.* (2010) proposed a network propagation algorithm based on formulating constraints on the prioritization function that relate to its smoothness over the network and usage of prior information. PRINCE receives as input a disease-disease similarity measure and a network of protein-protein interactions. The algorithm infers a strength-of-association scoring function that models an information pump that originates at the seed sets: this function is smooth over the network (adjacent nodes are assigned with similar values) and respects the prior knowledge on causal genes for the same disease or similar ones.
- **DA DA**. In Erten *et al.* (2011), the authors proposed DA DA (Degree-Aware Algorithms for Network-Based Disease Gene Prioritization), a free available suite implemented in Matlab. This algorithm applies statistical adjustment methods to correct for degree bias in information flow based disease gene prioritization with the aim to limit the prediction errors due to the incompleteness and the noisy network of PPI networks. Despite methods based on random walks or network propagation as the two above described (Kohler *et al.*, V1.36., 2008; Vanunu *et al.* 2010), are less affected by this problem by considering multiple alternate paths and whole topology of PPI networks, DADA outperforms these methods.
- **ProDiGe**. The algorithm proposed in Mordelet and Vert (2011) implements a novel machine learning strategy based on learning from positive and unlabeled examples: it assumes that a set of gene-disease associations is already known to infer new ones. The main difference from the existing approaches, is the definition of a scoring function using both the set P of known disease genes (positive examples) and the set U of candidate genes (unlabeled examples), formulating the problem of disease genes prediction as an instance of the problem known as learning from positive and unlabeled examples (PU learning). Moreover, ProDiGe is characterized by other two properties: i) it exploits information about known disease genes across diseases; ii) it allows heterogeneous data integration (sequence features, expression levels in different conditions...).
- **DIAMOnD (DIseAse MOdule Detection)**. In Chiassian *et al.* (2015), the authors showed that disease associated proteins do not reside within locally dense communities and that connectivity significance is the most predictive quantity. Based on these results, they developed a novel algorithm (DIAMOnD) that performs a systematic analysis of the network properties exploiting the connectivity significance instead of connection density. To identify disease modules, DIAMOnD algorithm performs the following steps:
 - i. computation of the connectivity significance for all proteins with at least one connection to any of the seed proteins:

$$p\text{-value}(k, k_s) = \sum_{k_i=k_s}^k p(k, k_i) \quad (5)$$

where s_0 is the relatively small number of seed proteins associated with a particular disease and $p(k, k_s)$ is the probability that a protein with k links has exactly k_s ($k_s < k$) links to seed proteins given by the hypergeometric distribution:

$$p(k, k_s) = \frac{\binom{s_0}{k_s} \binom{N-s_0}{k-k_s}}{\binom{N}{k}} \quad (6)$$

where N is the number of proteins;

- ii. ranking of the protein according to their respective p-values;
- iii. the protein with the highest rank (i.e. lowest p-value) is added to the set of seed nodes, increasing their number;
- iv. steps (i)-(iii) are iterated with the expanded set of seed proteins, pulling in one protein at a time into the growing disease module.

The procedure (i)-(iv) can be continued until the entire network is agglomerated. The order in which the proteins are being pulled in to the module reflects their topological relevance to the disease, resulting in a ranking of all proteins.

Conclusion and Perspectives

Among the obstacles and future perspectives concerning network inference, the discovery of ‘actual’ causal regulatory networks in the gargantuan space of possible networks is among the most formidable ones, just considering that the number of possible arrangements, i.e. different networks, of a relatively ‘trivial’ gene regulatory network containing as little as 50 nodes outclasses the estimated number of atoms of the observable universe ($\sim 10^{80}$). In such a challenge, it can be easily anticipated that previous knowledge, in terms of background, exploitable information, including already extracted causal connections and subnetworks, will likely impact the quality and quantity of successive attempts (or network ‘updates’) of network reconstruction in the different fields. Optimization methods, algorithmic progresses, innovative analytics and well-founded statistical inference as well as efficient approaches for integrating omic data and information will drive advancements in network reconstruction that should eventually speed up biological discovery, by helping in the avoidance of inaccurate or far from optimal network configurations.

Lately, artificial neural networks -deep learning- approaches are receiving more attention and resources in computational biology, thanks to their ability to capitalize on the accessibility and convenience of high-throughput omic data: better performance and higher-level characteristics can be obtained over conventional modelling methodology, and they could potentially deliver crucial understanding about the organization of biological networks.

The last decade has witnessed the relentless growth of biological molecular data (the so-called “omics” revolution) driven by highly efficient biotechnological devices and methods. Also, neuroscientists recently started collecting multi-subject, multi-modal human neuroimaging datasets: speaking more in general, ‘big data’ is the new reality also in biology, medicine and neuroscience. This growth is still ongoing and even increasing in speed and quality of outcomes. However, despite initial enthusiasms, the challenge of its management and use for the understanding of diseases has become over time a daunting task. In other words, the goal of turning lead into gold, i.e. transform data into knowledge, seems to step far away every single day. As a matter of fact, experimental measurements on almost every molecular entity acting in a cell are now currently available on databases spread all over the world, and this “heap” of data is incredibly huge and ever-increasing. The inevitable consequence of such wild expansion is that now it is practically impossible to put all the pieces together for two basic, simple reasons: (1) there are many missing ones, and (2) they are virtually infinite, so that the current ‘data factory’ is unlikely to stop working or even slow down. Now, it is time for asking questions and provide frameworks for real inter-disciplinary studies.

Network representations are invaluable for this purpose, as just shown. A network is the ideal metaphor for representing relationships among data, finding “patterns” for biomarker studies and, most importantly, any kind of expertise can find the opportunity to fully express itself using a common language. In fact, the network metaphor provides plenty of words and concepts that can be exploited by biologists and clinicians to represent their knowledge, and by bioinformaticians or computational modellers to derive algorithms and find answers using computation and mathematics. In other words, we envisage a new way of conceiving a “inter-disciplinary team” where biology and computation can flourish together (Korcsmaros *et al.*, 2017). The key point is that researchers should leave behind grand visions of “universal principles” and start an “earthly” work of untangling one by one the long list of problems that ask for (at least partial) answers in a peer-to-peer inter-disciplinary setting where the network metaphor provides the language to go beyond the conceptual/computational barriers.

See also: Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Computational Systems Biology Applications. Computational Systems Biology. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Gene Regulatory Network Review. Graphlets and Motifs in Biological Networks. Natural Language Processing Approaches in Bioinformatics. Network Models. Network Properties. Network-Based Analysis for Biological Discovery. Network-Based Analysis of Host-Pathogen Interactions. Networks in Biology. Pathway Informatics

References

- Achard, S., Bullmore, E., 2007. Efficiency and cost of economical brain functional networks. *PLOS Comput. Biol.* 3, e17.
- Aibar, S., Gonzalez-Blas, C.B., Moerman, T., *et al.*, 2017. SCENIC: Single-cell regulatory network inference and clustering. *Nat. Methods* 14, 1083–1086.
- Aldridge, B.B., Burke, J.M., Lauffenburger, D.A., Sorger, P.K., 2006. Physicochemical modelling of cell signalling pathways. *Nat. Cell Biol.* 8, 1195–1203.
- Baccalá, Luiz A., Sameshima, Koichi, 2001. Partial Directed Coherence: A New Concept in Neural Structure Determination. *Biological Cybernetics* 84 (6), 463–474. doi:10.1007/PL00007990.
- Bansal, M., Belcastro, V., Ambesi-Impiombato, A., Bernardo, D.I., 2007. How to infer gene networks from expression profiles. *Mol. Syst. Biol.* 3, 78.
- Barabási, A.L., Gulbahce, N., Loscalzo, J., 2011. Network medicine: A network-based approach to human disease. *Nat. Rev. Genet.* 12, 56–68.
- Bassett, D.S., Sporns, O., 2017. Network neuroscience. *Nat. Neurosci.* 20, 353–364.
- Blohm, P., Frishman, G., Smialowski, P., *et al.*, 2014. Negatome 2.0: A database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *Nucleic Acids Res.* 42, D396–D400.
- Bullmore, E., Sporns, O., 2009. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nat. Rev. Neurosci.* 10, 186–198.
- Butte, A.J., Tamayo, P., Slonim, D., Golub, T.R., Kohane, I.S., 2000. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc. Natl. Acad. Sci. USA* 97, 12182–12186.
- Cafarelli, T.M., Desbuleux, A., Wang, Y., *et al.*, 2017. Mapping, modeling, and characterization of protein–protein interactions on a proteomic scale. *Curr. Opin. Struct. Biol.* 44, 201–210.
- Chai, L.E., Loh, S.K., Low, S.T., *et al.*, 2014. A review on the computational approaches for gene regulatory network construction. *Comput. Biol. Med.* 48, 55–65.
- Chan, S.Y., LoScalzo, J., 2012. The emerging paradigm of network medicine in the study of human disease. *Circulation Research* 111, 359–6374.
- Chan, T.E., Stumpf, M.P.H., Babbie, A.C., 2017. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Syst.* 5, 251–267. (e3).
- De Smet, R., Marchal, K., 2010. Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.* 8, 717–729.
- Emmert-Streib, F., Dehmer, M., Haibe-Kains, B., 2014. Gene regulatory networks and their applications: Understanding biological and medical problems in terms of networks. *Front. Cell Dev. Biol.* 2, 38.
- Erten, S., Bebek, G., Ewing, R.M., Koyuturk, M., 2011. DADA: Degree-aware Algorithms for network-based disease gene prioritization. *BioData Min.* 4, 19.
- Franceschini, A., Szklarczyk, D., Frankild, S., *et al.*, 2013. STRING v9.1: Protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.* 41, D808–D815.
- Friston, K.J., Harrison, L., Penny, W., 2003. Dynamic causal modelling. *Neuroimage* 19, 1273–1302.
- Ghiassian, S.D., Menche, J., Barabási, A.L., 2015. A Disease Module Detection (DIAMOND) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLOS Comput. Biol.* 11, e1004120.
- Granger, C.W.J., 1969. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica* 37 (3), 424–438. doi:10.2307/1912791.
- Grech, R., Cassar, T., Muscat, J., *et al.*, 2008. Review on solving the inverse problem in EEG source analysis. *J. Neuroeng. Rehabil.* 5, 25.

- Haufe, Stefan, Vadim, V. Nikulin, Nolte, Guido, 2012. Alleviating the Influence of Weak Data Asymmetries on Granger-Causal Analyses. In: *Latent Variable Analysis and Signal Separation*. Berlin, Heidelberg: Springer, pp. 25–33. https://doi.org/10.1007/978-3-642-28551-6_4.
- Hecker, M., Lambeck, S., Toepler, S., Van Someren, E., Guthke, R., 2009. Gene regulatory network inference: Data integration in dynamic models—a review. *Biosystems* 96, 86–103.
- Huang, S.S., Fraenkel, E., 2009. Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Sci. Signal.* 2, ra40.
- Hyduke, D.R., Palsson, B.O., 2010. Towards genome-scale signalling network reconstructions. *Nat. Rev. Genet.* 11, 297–307.
- Kaminski, M.J., Blinowska, K.J., 1991. A New Method of the Description of the Information Flow in the Brain Structures. *Biological Cybernetics* 65 (3), 203–210.
- Kohler, S., Bauer, S., Horn, D., Robinson, P.N., 2008. Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.* 82, 949–958.
- Korcsmaros, T., Schneider, M.V., Superti-Furga, G., 2017. Next generation of network medicine: Interdisciplinary signaling approaches. *Integr. Biol. (Camb.)* 9, 97–108.
- Launay, G., Ceres, N., Martin, J., 2017. Non-interacting proteins may resemble interacting proteins: Prevalence and implications. *Sci. Rep.* 7, 40419.
- Lee, W.P., Tzou, W.S., 2009. Computational methods for discovering gene networks from expression data. *Brief Bioinform.* 10, 408–423.
- Lopes, M., Bontempi, G., 2013. Experimental assessment of static and dynamic algorithms for gene regulation inference from time series expression data. *Front. Genet.* 4, 303.
- Margolin, A.A., Nemenman, I., Basso, K., *et al.*, 2006. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinform.* 7 (Suppl 1), S7.
- Missiuro, P.V., Liu, K., Zou, L., *et al.*, 2009. Information flow analysis of interactome networks. *PLOS Comput. Biol.* 5, e1000350.
- Molinelli, E.J., Korkut, A., Wang, W., *et al.*, 2013. Perturbation biology: Inferring signaling networks in cellular systems. *PLOS Comput. Biol.* 9, e1003290.
- Mo, M.L., Jamshidi, N., Palsson, B.O., 2007. A genome-scale, constraint-based approach to systems biology of human metabolism. *Mol. Biosyst.* 3, 598–603.
- Mordelet, F., Vert, J.P., 2011. ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinform.* 12, 389.
- Nikoloski, Z., Perez-Storey, R., Sweetlove, L.J., 2015. Inference and prediction of metabolic network fluxes. *Plant Physiol.* 169, 1443–1455.
- Nolte, G., Bai, O., Wheaton, L., *et al.*, 2004. Identifying true brain interaction from EEG data using the imaginary part of coherency. *Clin. Neurophysiol.* 115, 2292–2307.
- Oti, M., Snel, B., Huynen, M.A., Brunner, H.G., 2006. Predicting disease genes using protein-protein interactions. *J. Med. Genet.* 43 (8), 691–698.
- Ourfali, O., Shlomi, T., Ideker, T., Rupp, E., Sharan, R., 2007. SPINE: A framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics* 23, i359–i366.
- Papin, J.A., Hunter, T., Palsson, B.O., Subramaniam, S., 2005. Reconstruction of cellular signalling networks and analysis of their properties. *Nat. Rev. Mol. Cell Biol.* 6, 99–111.
- Petti, M., Toppi, J., Babiloni, F., *et al.*, 2016. EEG resting-state brain topological reorganization as a function of age. *Comput. Intell. Neurosci.* 2016, 6243694.
- Pichiorri, F., Petti, M., Caschera, S., *et al.*, 2017. An EEG index of sensorimotor interhemispheric coupling after unilateral stroke: Clinical and neurophysiological study. *Eur. J. Neurosci.*
- Piro, R.M., Di Cunto, F., 2012. Computational approaches to disease-gene prediction: Rationale, classification and successes. *FEBS J.* 279, 678–696.
- Pitkanen, E., Rousu, J., Ukkonen, E., 2010. Computational methods for metabolic reconstruction. *Curr. Opin. Biotechnol.* 21, 70–77.
- Przulj, N., 2011. Protein-protein interactions: Making sense of networks via graph-theoretic modeling. *Bioessays* 33, 115–123.
- Qian, X., Dougherty, E.R., 2013. Validation of gene regulatory network inference based on controllability. *Front. Genet.* 4, 272.
- Rao, V.S., Srinivas, K., Sujini, G.N., Kumar, G.N., 2014. Protein-protein interaction detection: Methods and analysis. *Int. J. Proteomics* 2014, 147648.
- Ritz, A., Poirel, C.L., Tegge, A.N., *et al.*, 2016. Pathways on demand: Automated reconstruction of human signaling networks. *NPJ Syst. Biol. Appl.* 2, 16002.
- Schaefer, J., Strimmer, K., 2005. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics* 21, 754–764.
- Schryenmackers, M., Kuffner, R., Geurts, P., 2013. On protocols and measures for the validation of supervised methods for the inference of biological networks. *Front. Genet.* 4, 262.
- Serin, E.A., Nijveen, H., Hilhorst, H.W., Ligterink, W., 2016. Learning from co-expression networks: Possibilities and challenges. *Front. Plant Sci.* 7, 444.
- Smialowski, P., Pagel, P., Wong, P., *et al.*, 2010. The negatome database: A reference set of non-interacting protein pairs. *Nucleic Acids Res.* 38, D540–D544.
- Srivastava, A., Mazzocco, G., Kel, A., Wyrwicz, L.S., Plewczynski, D., 2016. Detecting reliable non interacting proteins (NIPs) significantly enhancing the computational prediction of protein-protein interactions using machine learning methods. *Mol. Biosyst.* 12, 778–785.
- Stolovitzky, G., Monroe, D., Califano, A., 2007. Dialogue on reverse-engineering assessment and methods: The DREAM of high-throughput pathway inference. *Ann. NY Acad. Sci.* 1115, 1–22.
- Toppi, J., De Vico Fallani, F., Vecchiato, G., *et al.*, 2012. How the statistical validation of functional connectivity patterns can prevent erroneous definition of small-world properties of a brain connectivity network. *Comput. Math. Methods Med.* 2012, 130985.
- Tuncbag, N., Braunstein, A., Pagnani, A., *et al.*, 2013. Simultaneous reconstruction of multiple signaling pathways via the prize-collecting steiner forest problem. *J. Comput. Biol.* 20, 124–136.
- Usadel, B., Obayashi, T., Mutwil, M., *et al.*, 2009. Co-expression tools for plant biology: Opportunities for hypothesis generation and caveats. *Plant Cell Environ.* 32, 1633–1651.
- Vanunu, O., Magger, O., Rupp, E., Shlomi, T., Sharan, R., 2010. Associating genes and protein complexes with disease via network propagation. *PLOS Comput. Biol.* 6, e1000641.
- Veiga, D.F., Dutta, B., Balazsi, G., 2010. Network inference and network response identification: Moving genome-scale data to the next level of biological discovery. *Mol. Biosyst.* 6, 469–480.
- Yin, T., Chen, S., Wu, X., Tian, W., 2017. GenePANDA – A novel network-based gene prioritizing tool for complex diseases. *Sci. Rep.* 7, 43258.

Further Reading

- Barabási, A.L., Oltvai, Z.N., 2004. Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.* 5 (2), 101–113. PMID: 14735121.
- Bassett, D.S., Sporns, O., 2017. Network neuroscience. *Nat. Neurosci.* 20 (3), 353–364. doi:10.1038/nn.4502. PMID: 28230844.
- Csermely, P., Korcsmaros, T., Kiss, H.J., London, G., Nussinov, R., 2013. Structure and dynamics of molecular networks: A novel paradigm of drug discovery: A comprehensive review. *Pharmacol. Ther.* 138 (3), 333–408. doi:10.1016/j.pharmthera.2013.01.016. PMID: 23384594.
- Le Novère, N., 2015. Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet.* 16 (3), 146–158. doi:10.1038/nrg3885. PMID: 25645874.
- Loscalzo, Joseph, Barabási, Albert-László, Silverman, Edwin K., 2017. *Network Medicine: Complex Systems in Human Disease and Therapeutics*. Harvard University Press.

Graphlets and Motifs in Biological Networks

Laurentino Quiroga Moreno, Birkbeck College, University of London, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

Large-scale biological networks are being generated at a growing rate. Advances in experimental biotechnologies are yielding molecular interaction networks of various types, including protein-protein interaction (PPI), epistatic genetic interaction (GI), metabolic reaction, and transcriptional regulation data, among others (Snider *et al.*, 2015). As each data set in isolation carries only limited information about biology and as the data sets complement each other (Przulj and Malod-Dognin, 2016), extracting new biological information hidden in the wiring patterns of these complex interconnected “big data” has received much attention in the scientific community and industry. These data are often modelled as *networks* (also called *graphs*), in which biomolecules are modelled as *nodes* (also called *vertices*) and their interactions are modelled as *edges* (also called *links*). Hence, biological systems have been modelled and analysed by using networks (Barabasi and Oltavi, 2004).

However, analysing large networks is mostly computationally intractable. Computational complexity theory studies hardness of computational problems and many of the problems underlying analytics of large networked data fall into the category of computationally intractable problems, so called NP-hard and NP-complete problems (Bondy and Murty, 1976). In short, computational problems roughly fall into the two main categories: polynomial time solvable problems (P) and non-deterministic polynomial time hard (NP-hard). Class P contains problems for which we can compute exact solutions in the time that is polynomial of the size of the input data. These problems are called computationally *easy*. In contrast, NP-hard problems cannot be solved exactly on large scale data, so we must solve them approximately, or heuristically. Some intractable problems include network comparison and alignment, which are hard due to NP-completeness (special type of NP-hard problems) of the underlying sub-graph isomorphism problem (Cook, 1971).

Proteins are key to the functioning of the cell and hence PPI networks play important roles in the functioning of cells. Similarities in the wiring of proteins within a PPI network have been shown to also mean similarity in biological function. Similarities in wiring patterns of proteins in PPI networks can also be used to guide network alignment algorithms, which analogous to sequence alignments, find and align similarly wired and similarly functioning proteins over PPI networks of different species. This, in turn, is used to transfer annotation from model organisms to human. Network motifs and graphlets have been used to address all these issues.

Network Motifs and Graphlets

For the above reasons, many approximate methods to analyse the structure of biological networks have been proposed. They can roughly and historically be divided into *local* and *global network properties*. Simple examples include the *degree* of a node (the number of edges that the node participates in), the *degree distribution* (the distribution of degrees over all nodes of a network), the *clustering coefficient* of a node (the number of edges between the neighbours of a node as a percentage of the maximum possible number of edges between them), the *average clustering coefficient* of the network over all its nodes etc. (Newman, 2010). More detailed descriptors of network structure include methods based on *network motifs* and *graphlets*, which are the subject of this article.

Network Motifs

Network motifs have been introduced by the group of Uri Alon (Milo *et al.*, 2002). They are defined as small patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks (Milo *et al.*, 2002). They were used to study the transcriptional regulation networks of well-studied microorganisms (Alon and Mangan, 2003; Mangan *et al.*, 2003), as well as of higher order organisms (Charney *et al.*, 2017; Datta *et al.*, 2017). It was shown that these networks appear to be made up of a small set of recurring regulation patterns, captured by network motifs.

Example motifs include positive and negative autoregulation, positive and negative cascades, positive and negative feedback loops, feedforward loops (FFLs), single input modules, and combinations of these, illustrated in Fig. 1 (Shoval and Alon, 2010). They were linked with biological function (illustrated in Fig. 1). Since the same network motifs have been found in diverse organisms from bacteria to humans, it has been suggested that they serve as basic building blocks of transcription networks. Network motifs in other biological networks also seem to exist, including signalling (Ptacek *et al.*, 2005; Itzkovitz *et al.*, 2005; Ma'ayan *et al.*, 2005) and neuronal networks (Sporns and Kotter, 2004; Sakata *et al.*, 2005). Also, they were used to analyse integrated networks of transcriptional regulation and protein-protein interactions (Yeger-Lotem *et al.*, 2004). For a review, see Alon (2007).

“Temporal motifs” were introduced to study the temporal changes in local network structure over time-evolving networks. For example, static network motifs described above were counted in each network snapshot over time and then their counts compared across the snapshots (Braha and Bar-Yam, 2009). This approach ignored any motif relationships between different network

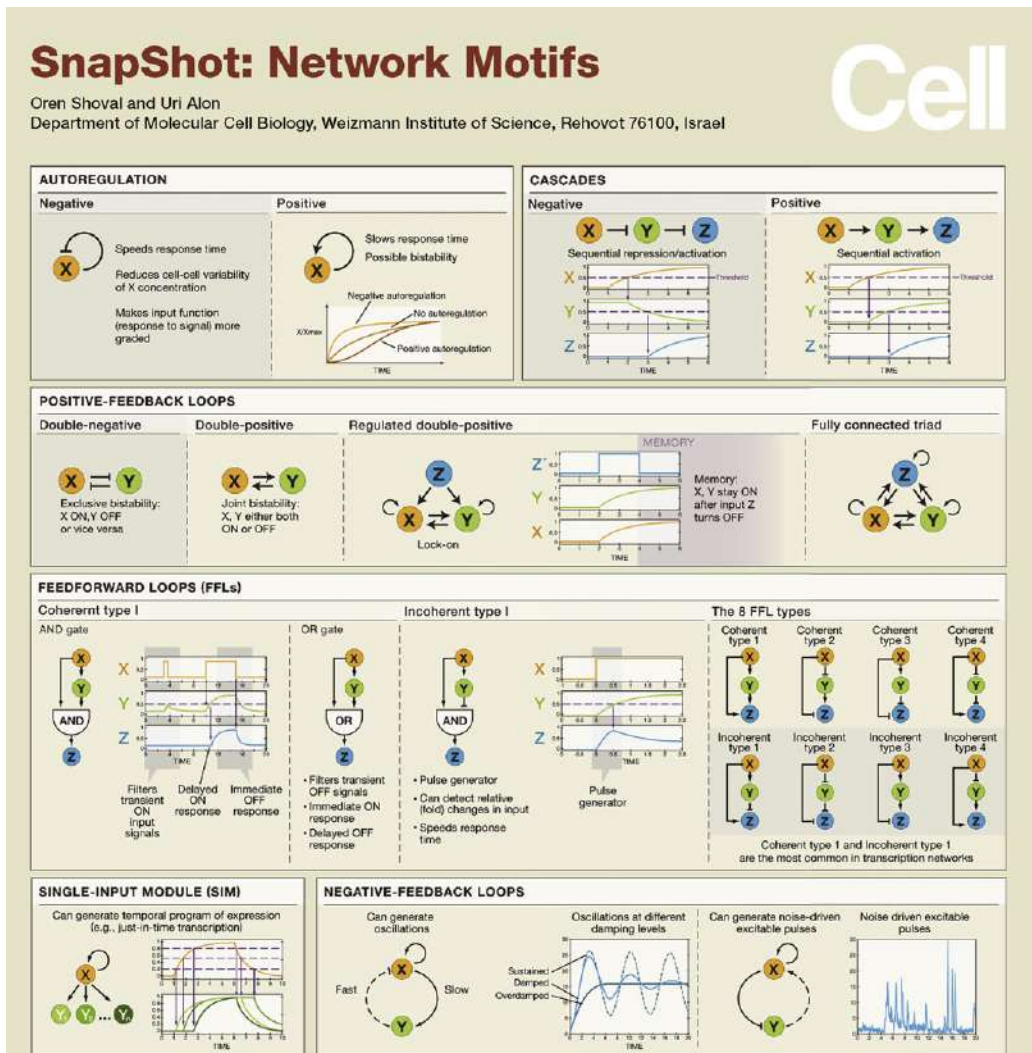


Fig. 1 An illustration of different types of network motifs and their functions. Taken from Shoval, O., Alon, U. 2010. SnapShot: Network motifs. Cell 143, 326.

snapshots. Hence, static network motifs were extended into several notions of temporal motifs (Chechik et al., 2008; Kovanen et al., 2011). Intuitively, temporal motifs are classes of sequences of similar network event sequences, where the similarity refers both to topology and to the temporal order of the events.

To understand the difference in the definition of motifs and graphlets, we need to introduce the following simple graph theoretic terms. A *partial subgraph* is a subgraph of a larger network in which once we pick the nodes that form the subgraph, we can pick any subset of edges between the chosen nodes of the larger network. An *induced subgraph* is a subgraph in which we must pick all the edges between the chosen nodes of the larger network to form the subgraph. Both of these subgraph definitions are illustrated in Fig. 2. Network motifs are partial subgraphs that are significantly overrepresented in the data compared to a chosen random graph model that is assumed to fit well the data.

However, as network comparison is computational intractable, determining a well-fitting network model is hard, as it involves comparing the data network with model networks. A random network model that is usually used as well-fitting is that of a random graph constructed to have the same degree distribution as the data network, while edges are drawn at random. Also, note that motifs are partial subgraphs, while characterizing the structure of any graph class is based on induced subgraphs (Andreas Brandstädt, 1999). An anti-motif is defined similar to a network motif, but it needs to be under-represented in the data compared to a random network model.

Hence, analyses of biological networks involving network motifs have been criticised, since the definitions of motifs and anti-motifs heavily depend on the choice of a random graph (network) model (Artzy-Randrap et al., 2004; Milo et al., 2004), since they are partial subgraphs (Przulj et al., 2004) and since they can exhibit a whole range of dynamic behaviours (Ingram et al., 2006). Furthermore, it is computationally hard to identify network motifs (and the same holds for graphlets), as the number of possible sub-graphs increases exponentially with the network and motif size (node and edge counts). Hence, various algorithms for their

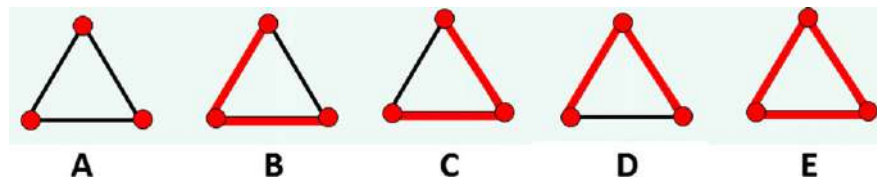


Fig. 2 Partial vs induced subgraphs. In the fully connected network with 3 nodes illustrated in panel A, if we pick all 3 of its nodes to make connected subgraphs, there exist three partial subgraphs corresponding to 3-node paths, illustrated in panels B, C and D, but only one induced subgraph, a triangle, illustrated in panel E.

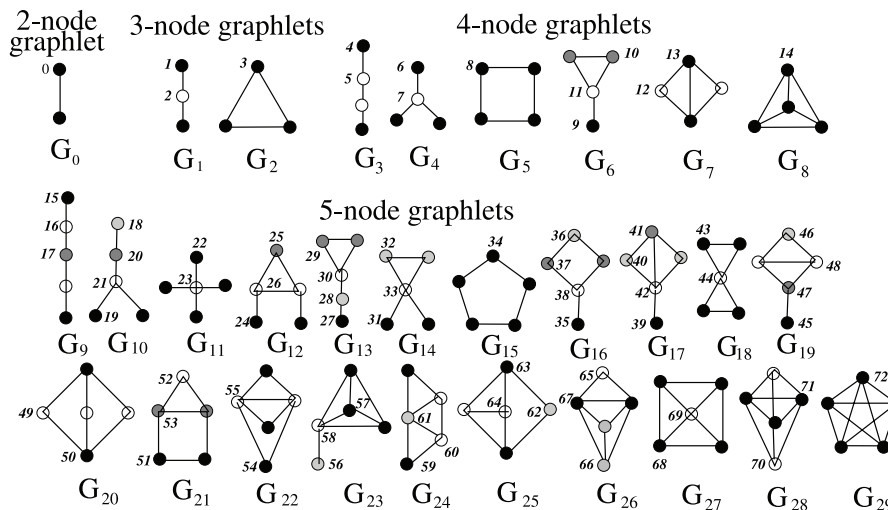


Fig. 3 All 2- to 5-node graphlets, \$G_0\$ to \$G_{29}\$, along with their symmetry groups (automorphism orbits), numbered from 0 to 72. Taken from Przulj, N., 2006. Biological Network Comparision Using Graphlet Degree. Oxford University Press, pp. e177–e183.

detection have been proposed, including those involving parallel and cloud computing (Kim *et al.*, 2013). A survey paper from 2011 discusses the biological significance of network motifs, the motivation for finding them, as well as the strategies for solving various aspects of this problem (Wong *et al.*, 2011).

Graphlets

As a response to the above mentioned drawbacks in the definition of network motifs, *graphlets* were introduced (Przulj *et al.*, 2004). Graphlets are small, connected, induced subgraphs of large networks (Przulj *et al.*, 2004). They can appear in the data network at any frequency and hence their definition does not depend on any assumed random graph model that fits well the data. Also, they are induced, so they are a suitable tool for designing various algorithms for mining domain relevant new information from the structures (topologies) of biological and other network data. All 2- to 5-node undirected graphlets are illustrated in Fig. 3.

As systems-level molecular network data present an opportunity to discover new biological information, several data analytics algorithms included graphlets as the underlying methodology aimed at uncovering the additional meaning. For instance, *graphlet frequency distribution* is introduced to be superior to the *degree distribution* (which is the first in the spectrum of 73 graphlet degree distributions), the *clustering coefficient* (the measure of “cliquishness” of the network) and *network diameter* (which measures how “far spread” the nodes of the network are) by imposing a large number of similarity constraints on the networks being compared (Przulj, 2006). It does so by generalizing the degree distribution (which measures the number of nodes touching k edges) into a spectrum of 73 distributions measuring the number of nodes touching k graphlets at a particular orbit for each value of k (the 73 orbits for 2- to 5-node graphlets are numbered in Fig. 3). Basically, it compares sequences of numbers over 73 pairs to compare two networks. It can be classified into both types of network comparison heuristics, global and local, as it does comparison globally over the entire network, but uses local network features in the comparisons.

As biological networks are globally incomplete, but locally more complete (due to biases in biotechnologies and experimental design focusing on heavily exploring the parts of networks relevant for human disease), focusing more on local network properties provides a better measure of network structure for these particular data (Przulj, 2006). This was demonstrated on 14 real PPI networks and the four network models (Przulj, 2006). Also, it was shown for the first time that *geometric random graphs* are fitting PPI networks better than other previously used models (Przulj *et al.*, 2004). In geometric random graphs, nodes correspond to points randomly distributed in a metric space and edges are drawn between the points if the corresponding points are close

enough (within a given radius) in space. This model was later refined to mimic network growth inspired by gene duplication and mutation events (Przulj *et al.*, 2010).

A more sophisticated heuristic measure of distance between networks is called *Graphlet Correlation Distance (GCD)*, which exploits correlations between graphlet orbits (symmetries in graphlets, see Fig. 3) over all nodes in a network, to compress the topology of a network of any size and density into an 11 by 11 matrix of numbers between -1 and 1 , the so called Graphlet Correlation Matrix (GCM). It compares GCMs of different networks to find the topological similarity between the networks (Yaveroglu *et al.*, 2014). The superiority of GCD over other measures was demonstrated not only in accuracy, but also in noise tolerance and computational efficiency (Yaveroglu *et al.*, 2015). GCD demonstrated relationships between unrelated networks, such as, Facebook, metabolic, and protein structure networks. Biological networks were also compared by the method involving differential graphlet communities (Wong *et al.*, 2015).

The constant production of large scale, dynamic and directed correlated data has demanded development of more sophisticated methods to analyse them and gain additional biological insights. Analysing directed networks by using graphlets was proposed, since their above described undirected counterparts were successful in analysing undirected network data. Graphlets and their degrees were extended to directed data, as well as the GCD measure (Sarajlic *et al.*, 2016). It was demonstrated that these directed versions of graphlet-based tools are more sensitive in comparing directed networks than if the undirected version was applied to directed data. Additionally, it was necessary to extend the conical correlation analysis framework to facilitate uncovering the relationships between directed network wirings and their annotations. The method was applied to metabolic networks (Sarajlic *et al.*, 2016).

Algorithms for Graphlet Identification

However, identifying graphlets is computationally demanding as one needs to search through exponentially many possible subgraphs, so heuristic approaches have been sought (Przulj *et al.*, 2006; Wang, *et al.*, 2016). For instance, by using geometric graphs as a well-fitting model of PPI networks, a heuristic exploits the property of geometric graphs being sparser at the “boundary” and denser in the “core” of the network to quickly estimate graphlet counts in geometric graphs and also in PPI networks, since they are fit well by geometric graphs. This heuristic is called Targeted Node Processing (TNP), since it targets counting of graphlets in these sparse, network boundary neighbourhoods and from them estimates the counts in the entire network. The same paper proposes another heuristic called Neighbourhood Local Search (NLS) that is based on a more standard sampling search procedure of randomly sampling nodes and counting graphlets around them to produce the estimates (Przulj *et al.*, 2006). Both heuristics work well for high-confidence PPI networks, which have power-law degree distribution, as well as for geometric random graphs which do not, indicating that the behaviour of the heuristics is dictated by the local rather than global structure of these networks. A combinatorial method was also proposed, using a system of equations that connect counts of orbits in graphlets to allow for computing of all orbit counts by enumerating a single one, hence reducing practical time complexity in sparse graphs by an order of magnitude compared to the exhaustive enumeration-based algorithms (Hocavar and Demsar, 2014).

Graphlets Link Network Structure and Biological Function

Since few protein structures have been resolved, predicting structure and function of uncharacterized proteins has been attempted by using various data sets. The traditional techniques were based on homology, but with the availability of systems-level PPI data that aim to capture the entire proteome, the questions arose of whether protein function could also be predicted from PPI network data. In this direction, a graphlet-based method demonstrated that the local structure around a protein in a PPI network and its biological function are related (Milenkovic and Przulj, 2008). The method uses a vector of graphlet degrees, called a *signature* of the protein in the PPI network, and computes the similarities of signatures of proteins across all protein pairs. It demonstrated that when the local wiring patterns around proteins are similar and proteins are grouped based on these similarities, the grouped proteins tend to perform the same biological functions, belong to the same subcellular compartments and have the same tissue expression. These observations opened up research directions for experimental design and for disease protein prediction from proteome-scale network data.

Also, new disease genes were identified from the analysis of the topology of PPI networks. It was assumed that neighbours of disease genes in the PPI network are also involved in the same or similar diseases (Ideker and Sharan, 2008). Simple network properties of disease genes were used in order to identify new gene-disease, or drug to drug-target associations. For instance, it has been demonstrated that cancer genes have larger connectivities and centralities compared with non-cancer genes (Jonsson and Bates, 2006). However, the relationship between disease genes and their network degree (number of neighbours they have in the network) needed to be analysed in more detail, because many diseases genes code for proteins that are not hubs in PPI networks (Goh *et al.*, 2007). Hence, demonstrating a relationship between cancer genes and their wirings in the PPI network required more sophisticated methodology.

Since it has been demonstrated that proteins with similar wirings in PPI networks can have the same functions (Milenkovic and Przulj, 2008), the new challenge was to establish if cancer genes also exhibit similarity of their topological graphlet degree signatures. Indeed, when signatures were used to predict new cancer genes purely from the topology of PPI networks, 80% of the predicted new cancer genes were validated in the literature (Milenkovic *et al.*, 2009). Furthermore, the biological validations with siRNA screens confirmed some of the predictions, in particular cancer-related negative regulators of melanogenesis (Milenkovic

et al., 2009). The results were more encouraging than expected, as the predictions were obtained only from PPI network topology, demonstrating their importance as a source of new biological information. Also, the study showed evidence that the structure of the PPI network around cancer genes is different from the structure around non-cancer genes.

Furthermore, graphlets were used to show that similarity of wiring of proteins and the link of their wiring with biological function is conserved over species even as distant as yeast and human (Davis *et al.*, 2014). It was noted that the variability of topology-function relationships required more investigation to make functional predictions and do annotation transfers, because the topology-function relationship may differ between different species and between different protein functions. To address this issue, a statistical framework was built by using conical correlation analysis (CCA). In this direction, the wirings around proteins in a PPI network are represented by graphlet degree vectors and correlated with Gene Ontology (GO) annotations (Ashburner *et al.*, 2000) by using CCA. As a result, statistically significant topology-function relationships were obtained between yeast and human. Functions that have conserved topology in PPI networks of these species were discovered, which were called “topologically orthologous” functions (Davis *et al.*, 2014).

A question of how much functional information could be obtained from sequence and how much from PPI network topology, and whether the two data types as sources of biological information corroborate, or complement each other has also been addressed by using graphlets (Memisevic *et al.*, 2010). Complementarity of PPI network and sequence information in homologous proteins has been examined, as PPI network topology and protein sequence may provide details into different slices of biological information (Memisevic *et al.*, 2010). The goal was to ensure that no biological information is lost by considering only sequence as a relevant source of biological information. First, it was examined if the information about homologues captured by the PPI network topology by using graphlet degree vectors (signatures) differs and to what extent from the information captured by their sequences. The similarity of topology around homologous proteins in the PPI network was statistically significantly higher than in non-homologous proteins. Second, none of the data sources, network topology and sequence, seemed to capture homology information entirely. However, it was demonstrated that both data sources provide relevant and complementary functional information, concluding that topological neighbourhoods of a protein in the PPI network could complement sequence-based information to identify homologous proteins.

Graphlets were used to analyse other biological networks as well. In a brain functional network, brain regions are modelled as nodes and if they are simultaneously active while the brain performs a cognitive task as measured by electrocorticographic signals, then the corresponding nodes are linked by an edge. It was established that these brain connectivity networks have a small-world topology (Basset and Bullmore, 2006). However, novel graph theoretic techniques analysed brain function networks to allow us to identify structural changes in the brain's wiring as a result various stimuli, or cognitive tasks. On brain functional networks of six epileptic patients, by using graphlet-based network similarity measures, it was found that during resting, these networks are differently wired than during performing cognitive tasks (Kuchaiev *et al.*, 2009).

In addition, analysing temporal real-world networks is a challenge. There are various options on how to model a temporal network, for example, as single aggregate statistic networks, or as a series of time-specific snapshots. Depending on the modelling, there can exist limitations in extracting information from them, because valuable temporal information could be lost if a static analysis is implemented. They were studied by considering evolution of their global network properties (Leskovec *et al.*, 2005; Nicosia *et al.*, 2012). Also, graphlets were used to study temporal networks by registering intermediate snapshot relationships (Hulovatyy *et al.*, 2015). This use of graphlets was proposed as an alternative to the one based on temporal motifs described above to alleviate the above noted limitations of finding the well-fitting null model that's required to define motifs, which is a complex task. This method was applied to age-specific molecular networks to better understand the processes of human aging.

Graphlets in Network Aligners

Another bioinformatics application in which graphlets are used is that of network alignment. Just as sequence alignment algorithms have revolutionized our understanding of biology, aligning systems-scale molecular networks will have similar ground-breaking impacts. However, as described above, aligning networks is computationally intractable due to NP-completeness of the underlying subgraph isomorphism problem. Hence, heuristic algorithms for network alignment have been proposed (Kelley *et al.*, 2004). One of the first algorithms to use purely network topology without sequence to align pairs of PPI networks is based on a seed-and-extend approach where nodes with the most similar graphlet-based topological signatures in the PPI network are aligned and then the alignment was extended around these seed nodes in a principled way (Kuchaiev *et al.*, 2010). It was applied to biological networks to produce the most comprehensive network alignments at the time, exposing surprisingly large regions of PPI network topological similarity even in species as distant as yeast and human. It produced new insight into phylogeny and biological function. As there are many possible alignments, the question of which one is optimal was addressed by using the Hungarian algorithm for solving the bipartite matching problem, where the bipartitions are nodes (proteins) of PPI network being aligned (e.g., yeast and human) and the edges across represent graphlet degree vector similarity scores between the nodes (Milenkovic *et al.*, 2010). The latest in the series of graphlet-based aligners is L-GRAAL, which combines the statistics of graphlets with Lagrangian relaxation to produce the state-of-the-art alignments (Malod-Dognin and Przulj, 2015).

Another application of graphlets-based network aligners is in structural biology. Alignments of protein structures aim to deliver the most accurate mappings of equivalent residues and elucidate protein function. Aligning protein structures has attracted considerable research efforts, since there is no algorithm that yields satisfactory results and since the existing algorithms are not

applicable to large scale data. Hence, proteins have been modelled as networks, in which nodes are residues and an edge exists if the corresponding residues are close enough, usually within 5 Å. As graphlets were used to design sensitive topological similarity measures between nodes and networks, the question was whether they could also be used to align protein structures. A method called GR-align was introduced for alignment of protein structures, which uses a maximum Contact Map Overlap (CMO) that is suitable for database searches (Malod-Dognin and Przulj, 2014). GR-align is more efficient in terms of speed than other similar methods. Additionally, its similarity scores agree with the structural classification of proteins. The method was applied on the Astral-40 dataset. Apart from producing better scoring than other structure alignment heuristics, it also produced results with higher level of agreement with the structural classifications proteins. GR-align also allowed for transferring more information across proteins (Malod-Dognin and Przulj, 2014).

Despite these efforts, aligning pairs of large networks still represents a challenge. Aligning multiple networks is even more challenging, as performance of such aligners decreases with increase in the number of networks being aligned. Another challenge to be met by a multiple network aligner is to deliver clusters of nodes (genes) that are evolutionarily and functionally conserved across all aligned networks, i.e., to produce both topologically and biologically meaningful alignments. The current state-of-the-art multiple network alignment algorithm, called Fuse, works in two steps (Gligorijevic *et al.*, 2015). First, it generates node (protein) similarity scores across multiple PPI networks by using penalized (graph-regularized) non-negative matrix tri-factorization (NMTF) for fusing information from wiring patterns of all aligned protein-protein interaction (PPI) networks and sequences similarities between their proteins. Second, it identifies clusters of aligned proteins over all networks, for which it proposes a new heuristic for addressing an NP-hard problem of k-partite matching for $k > 2$. Fuse outperforms others aligners and produces a larger number of biologically consistent clusters that cover all aligned protein-protein interaction (PPI) networks.

Concluding Remarks

Despite significant advances in methods based on network motifs and graphlets for analysing and extracting new biological and medial information from large-scale molecular networks, we are far from comprehensively understanding biological systems such as cells, tissues, or organs. Roles of molecular networks in health and disease are still to be uncovered and once uncovered, they need to be altered by therapeutics to improve patient treatment and care. Each individual network type gives insight only into one slice of biological information. Graphlet- and motif-based methods need to be complemented by new sophisticated graph theoretic methods capable of capturing multi-scale organization of biological systems. Furthermore, these new methods and formalisms need to be integrated within data fusion frameworks, so that we would get a data-driven, full understanding of a biological system by mining all available data collectively.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Alignment of Protein-Protein Interaction Networks. Cluster Analysis of Biological Networks. Community Detection in Biological Networks. Computational Systems Biology Applications. Computational Systems Biology. Gene Regulatory Network Review. Graph Algorithms. Graph Isomorphism. Graph Theory and Definitions. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Network Models. Network Properties. Network Topology. Network-Based Analysis for Biological Discovery. Networks in Biology. Pathway Informatics

References

- Alon, U., 2007. Network motifs: Theory and experimental approaches. *Nature Reviews Genetics*. (8), 450–461.
- Alon, U., Mangan, S., 2003. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences of the United States of America* 100 (21), 11980–11985.
- Andreas Brandstädt, V.B., 1999. GRAPH Classes: A Survey. SIAM.
- Artzy-Randrap, Y., Fleishman, S., Ben-Tal, N., Stone, L., 2004. COMMENT on "Network motifs simple blocks of complex networks" and "superfamilies of evolved and designed networks". *Science*.
- Ashburner, *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25 (1), 25–29.
- Barabasi, A., Oltavi, Z., 2004. NETWORK biology: Understanding the cell's functional organization. *Nature Reviews Genetics* 5, 101–113.
- Basset, D., Bullmore, E., 2006. Small-world brain networks. *The Neuroscientist* 12 (6), 512–523.
- Bondy, J., Murty, U., 1976. Graph Theory With Applications. Macmillan.
- Braha, D., Bar-Yam, Y., 2009. Time-dependent complex networks: Dynamic centrality, dynamic motifs, and cycles of social interactions. *Adaptive Networks*. Berlin: Springer, pp. 39–50.
- Charney, R.M., Paraiso, K.D., Blitz, I.L., Cho, K.W., 2017. A Gene Regulatory Program Controlling Early *Xenopus* Mesendoderm Formation: Network Conservation and Motifs, 66. Elsevier. pp. 12–24.
- Chechik, G., *al.*, 2008. Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network. *Nature Biotechnology* 26, 1251–1259.
- Cook, S., 1971. THE complexity of theorem-proving procedures. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. ACM, pp. 151–158.
- Datta, R., *et al.*, 2017. A feed-forward relay between bicoid and orthodenticle regulates the timing of embryonic patterning in *Drosophila*. *bioRxiv*. 1–46.
- Davis, D., Yaveroglu, O.N., Malod-Dognin, N., Stojmirovic, A., Przulj, N., 2014. Topology–Function Conservation in Protein–Protein Interactions Networks. Oxford University Press. pp. 1–7.

- Glgorijevic, V., Malod-Dognin, N., Przulj, N., 2015. FUSE: Multiple Network Alignment via DAT Fusion. Oxford University Press. pp. 1195–1203.
- Goh, K., Cusick, M., Valle, D., *et al.*, 2007. The human disease network. National Academy of Sciences. 8685–8690.
- Hocevar, T., Demisar, J., 2014. A combinatorial approach to graphlet counting. Bioinformatics. 559–565.
- Hulovatyy, Y., Chen, H., Milenkovic, T., 2015. Exploring the Structure and Function of Temporal Networks With Dynamic Graphlets. Oxford University Press. pp. i171–i180.
- Ideker, T., Sharan, R., 2008. Protein networks in disease. Genome Research 18, 644–652.
- Ingram, P., Stumpf, M., Stark, J., 2006. Network motifs: Structure does not determine function. BMC Genomics 7 (1), 108.
- Itzkovitz, S., *et al.*, 2005. Coarse-graining and self-dissimilarity of complex networks. Physical Review E 71, 016127.
- Jonsson, P., Bates, P., 2006. Global topological features of cancer proteins in the human interactome. Bioinformatics 22, 2291–2297.
- Kelley, B., Yuan, B., Lewitter, F., *et al.*, 2004. PathBLAST: A tool for alignment of protein interaction networks. Nucleic Acids Research. W83–W88.
- Kim, W., Diko, M., Rawson, K., 2013. Network motif detection: Algorithms, parallel and cloud computing, and related tools. IEEE Xplore 18 (5), 469–489.
- Kovanen, L., *et al.*, 2011. Temporal motifs in time-dependent networks. Journal of Statistical Mechanics: Theory and Experiment. P11005.
- Kuchaiev, O., Milenkovic, T., Memisevic, V., Hayes, W., Przulj, N., 2010. Topological network alignment uncovers biological function and phylogeny. Journal of the Royal Society. 1341–1354.
- Kuchaiev, O., Wang, P., Nanedic, Z., Przulj, N., 2009. STRUCTURE of brain functional networks. In: Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'09), pp. 1–5. Minneapolis, Minnesota: IEEE.
- Leskovec, J., *et al.*, 2005. Graphs over time: Densification laws, shrinking diameters and possible explanations. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press. pp. 177–187.
- Ma'ayan, A., *et al.*, 2005. Formation of regulatory patterns during signal propagation in a mammalian cellular network. Science 309, 1078–1083.
- Malod-Dognin, N., Przulj, N., 2015. L-GRAAL: Lagrangian Graphlet-Based Network Aligner. Oxford University Press. pp. 2182–2189.
- Malod-Dognin, N., Przulj, N., 2014. GR-ALIGN: Fast and Flexible Alignment of Protein 3D Structures Using Graphlet Degree Similarity. Oxford University Press. pp. 1259–1265.
- Mangan, S., Zaslaver, A., Alon, U., 2003. The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. Journal of Molecular Biology 334 (2), 197–204.
- Memisevic, V., Milenkovic, T., Przulj, N., 2010. Complementary of network and sequence information in homologous proteins. Journal of Integrative Bioinformatics. 1–15.
- Milenkovic, T., Ng, W., Leong, Hayes, W., Przulj, N., 2010. OPTIMAL network alignment with graphlet degree vectors. Cancer Informatics. 121–137.
- Milenkovic, T., Memisevic, V., Ganesan, A., Przulj, N., 2009b. Systems-levels cancer gene identification from protein interaction network topology applied to melanogenesis-related functional genomics data. Journal of the Royal Society. 1–15.
- Milenkovic, T., Przulj, N., 2008. Uncovering biological network function via graphlet degree signatures. Cancer Informatics. 257–273.
- Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Alon, U., 2004. Response to comment on "Network motifs: Simple building blocks of complex network" and "Superfamilies of evolved and designed networks". Science. 2.
- Milo, R., Shen-Orr, S., Itzkovitz, S., *et al.*, 2002. Network motifs: Simple building blocks of complex networks. Science. 824–827.
- Newman, M., 2010. Networks: An Introduction. Oxford University Press.
- Nicosia, V., *et al.*, 2012. Components in time-varying graphs. Chaos 22, 3101.
- Przulj, N., Corneil, D.G., Jurisica, I., 2004a. Modeling interactome: Scale-free or Geometric? Bioinformatics. 3508–3515.
- Przulj, N., Corneil, D., Jurisica, I., 2004b. Modeling interactome: Scale-free or geometric? Bioinformatics. 3508–3515.
- Przulj, N., Corneil, D., Jurisica, I., 2006. Efficient Estimation of Graphlet Frequency Distributions in Protein–Protein Interaction Networks. Oxford University Press. pp. 974–980.
- Przulj, N., Kuchaiev, O., Stefanovic, A., Hayes, W., 2010. Geometric evolutionary dynamics of protein interaction networks. In: Proceedings of the 2010 Pacific Symposium on Biocomputing (PSB). Big Island, Hawaii.
- Przulj, N., Malod-Dognin, N., 2016. Network analytics in the age of big data. Science. 123–124.
- Przulj, N., 2006. Biological Network Comparison Using Graphlet Degree. Oxford University Press. pp. e177–e183.
- Placek, J., *et al.*, 2005. Global analysis of protein phosphorylation in yeast. Nature 438, 679–684.
- Sakata, S., Komatsu, Y., Yamamori, T., 2005. Local design principles of mammalian cortical networks. Neuroscience Research 51, 309–315.
- Sarajlic, A., Malod-Dognin, N., Yaveroglu, O.N., Przulj, N., 2016. Graphlet-based characterization of directed networks. Scientific Reports. 1–14.
- Shoval, O., Alon, U., 2010. SnapShot: Network motifs. Cell 143, 326.
- Snider, J., Kotlyar, M., Saraon, P., *et al.*, 2015. Fundamentals of protein interaction network mapping. Molecular System Biology. 11–848.
- Sporns, O., Kotter, R., 2004. Motifs in brain networks. PLOS Biology 2, e369.
- Wang, P., Zhang, X., Li, Z., *et al.*, 2016. A fast sampling method of exploring graphlet degrees of large directed. arXiv 1604, 08691.
- Wong, E., Baur, B., Quader, S., Huang, C.-H., 2011. Biological Network Motif Detection: Principles and Practice. pp. 202–215.
- Wong, S.W., Cercone, N., Jurisica, I., 2015. Comparative network analysis via. Proteomics. 608–617.
- Yaveroglu, O.N., Malod-Dognin, N., Davis, D., *et al.*, 2014. Revealing the hidden language of complex networks. Scientific Reports. 1–9.
- Yaveroglu, O.N., Milenkovic, T., Przulj, N., 2015. Proper Evaluation of Alignment-Free Network Comparison Methods. Oxford University Press. pp. 2697–2704.
- Yeger-Lotem, E., *et al.*, 2004. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. Proceedings of the National Academy of Sciences of the United States of America 101 (16), 5934–5939.

Biographical Sketch

Laurentino Quoroga Moreno is a Colombian engineer. He moved from his home country to the UK in 2007 to study English and enrol into post-graduate studies at King's College London supported by a scholarship from Colombia. In 2012, he obtained a postgraduate certificate in Engineering with Business Management from King's College London. He worked in private and public sectors for many years. In 2016, he changed disciplines and entered an MSc program in Bioinformatics with Systems Biology at Birkbeck College, University of London. Currently, he is carrying out his research project at University College London in the department of Computer Science on disease-disease relationships from systems-level molecular network data.

Protein-Protein Interactions: An Overview

Edson L Folador, Federal University of Paraíba, João Pessoa, Brazil

Sandeep Tiwari, Federal University of Minas Gerais, Belo Horizonte, Brazil

Camila E Da Paz Barbosa, Federal University of Paraíba, João Pessoa, Brazil

Syed B Jamal, Federal University of Minas Gerais, Belo Horizonte, Brazil

Marco Da Costa Schulze, Federal University of Paraíba, João Pessoa, Brazil

Debmalya Barh, Institute of Integrative Omics and Applied Biotechnology, Purba Medinipur, India

Vasco Azevedo, Federal University of Minas Gerais, Belo Horizonte, Brazil

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological Interaction

Proteins are organic macromolecules which have the ability to act together with several other molecules, including other protein (Moraes, 2013). They are normally identified by their individual actions as catalysts for reactions in metabolic pathways, signaling molecules in intracellular regulation cascade, defense proteins such as antibodies, nutrient or storage proteins, motile or contractile proteins (Jeong *et al.*, 2001; Phizicky and Fields, 1995). But the proteins are also identified by structural actions carried out by the interaction by forming complexes such as proteins of the cytoskeleton, signal transduction, cell-to-cell communication, transcription, replication and membrane transport (Keskin *et al.*, 2016). Regardless of the activities to be structural, metabolic or regulatory, proteins are important in maintaining homeostasis of the organism through the interactions that form each other and with other molecules, are essential for any living organism.

For some proteins become active and perform its function, it is required that binds to other molecules, thus forming monomers, dimers, trimers, polymers, etc. A monomer is defined as a molecule capable of binding to other molecules. When two monomers (molecules/proteins) are bound are called dimers, and if the dimer is formed by the same protein is called homodimers. A dimer sample can be observed with a binder in the structure of thymidylate synthase (TS) (PDB: 5X69), an enzyme of the transferase class with function to catalyze the synthesis monophosphate thymidine (dTMP) from uridine monophosphate (dUMP), essential in DNA synthesis and directly linked to tumor proliferation rate (Chen *et al.*, 2017). Trimers are joints of three monomers, for example the Dynamic HIV Envelope Trimer Apex complexed with neutralizing HIV antibodies (PDB: 5V8L) (Lee *et al.*, 2017) and when it is formed by the junction of the same molecules is termed homotrimer. The polymers contain a large number of monomers linked such as retinal protein for cell-cell adhesion (retinoschisina – RS1) an octamer consisting of eight monomers (PDB: 3JD6). When they are composed of the same molecule form a homopolymer such as the Ebola virus membrane-associated matrix protein (vP40) formed by four antiparallel homodimers (PDB: 1H2D).

The biological interactions maintain proper functioning of the body and occur at all times to ensure the production or non-production of biomolecules involved in metabolism. Proteins are key elements in the interactions processes and they are responsible for modulating catabolism and anabolism processes with synthesis or degradation of essential molecules, respectively.

The biological interactions can be defined as a mutual operation of several components of an organism to ensure proper sequence of events that together ensure the development of the individual. With the premise that all living beings have evolved from a common ancestor we can assume that, in addition to genes and proteins, the interactions are conserved.

The biological interactions occur only within a single organism or cell, they also occur between different host-pathogen interactions of bacteria *Mycobacterium tuberculosis* and Homo sapiens and the knowledge of this protein-protein interaction network (PIN) is an important tool for understanding how the bacteria interact with their host and expresses its virulence (Huo *et al.*, 2015). The knowledge about the proteins and how they interacts in a PIN enables a systemic understanding of the organism, cell or metabolic pathway, and may able since the discovery of new therapies to treat diseases until the knowledge about the metabolism of the microorganisms to be used in industrial, pharmacological, agricultural or livestock bioprocesses. Estimates indicate on average 30% of the organisms' proteins are identified in the interaction networks, with PINs being a large area of study and with many possibilities still open (Hao *et al.*, 2016).

For an interaction to exist, it is necessary to have a fine space-time coordination, that is, both partner proteins must coexist in the same cellular location at the same time (Hao *et al.*, 2016). This leads to the thought that for an interaction to occur, it is not enough just to have the respective coding genes, the proteins need to be transcribed and translated at the same time and in the same place. This fact has an even greater relevance in eukaryotic, because at least the interacting proteins must be expressed or transported to the same tissue. In time, transmembrane or secreted proteins are able to interact with proteins outside the cell environment such as occurs in the host-pathogen interactions. These characteristics exert direct influence on the methods of identification and prediction of interaction and, if not considered, can lead to false positive or false negative identification.

In order to understand a PIN is necessary to know the types and characteristics of each protein interaction.

Interactions Types

Interactions between proteins can be classified as physical (structural) or functional (regulatory). An interaction is physical when there is physical contact between two proteins and it is functional when there is an activation or inhibition event between the biomolecules, often mediated by intermediary cellular processes as in a metabolic pathway.

Interactions may also be classified according to duration time as transient or persistent. The transient interactions occur for a short period of time when a protein binds to the other only at a time and, after performing its function are separated. Persistent interactions tend to be more stable and occur when proteins remain bound for a longer period or until they are degraded (Nooren and Thornton, 2003). In time, the proteins may be classified according to the amount of interaction. Proteins with many interactions are termed hubs and play a central role in PIN by connecting several other proteins, probably assuming several functions and participating in several metabolic pathways.

These types of interactions are not exclusive and may occur simultaneously. Due to the temporary nature, transient interactions are more difficult to identify and study. Physical interactions are important targets in the development of a new class of drugs with the potential to inhibit or stabilize interactions.

Interaction Representation

The representation for PINs view is based on graph theory, coming from the exact area. An interaction is symbolized by nodes (representing proteins) and by vertices linking these nodes (representing the interaction), as can be observed in Fig. 1.

In biological networks, typically the interactions are non-directional, indicating there is no source or destination for interaction and can be stated without loss of information that both the protein 'A' interacts with the protein 'B' as protein 'B' interacts with 'A'.

A directional interaction is represented by an arrow ($>$) at the end of vertex indicating the direction of interaction, thus having source and destination in the proteins of the interaction. Two proteins form an interaction, and several interactions, peer-to-peer, form a complex Protein Interaction Network (PIN), which may contain hundreds or thousands of interactions.

Protein-Protein Interaction Network (PIN)

The PINs, also known as interactome, are important tools for the systemic study of an organism, since they represent a set of protein interactions whose characteristics are susceptible of analyzes inherited from the graph theory. It is possible to observe in a PIN, since interactions between proteins in a complex or in a metabolic pathway until a set of pathways, comprising both organisms showing low complexity as those with high complexity (Szklarczyk *et al.*, 2011). By analyzing the characteristics, a PIN provides elements for a great variety of studies, such as (i) understanding the organism studied at a systemic level, (ii) visualizing and complementing metabolic pathways, (iii) presuming protein functions and assisting the annotation process of genes and hypothetical proteins, (iv) identify new targets for drugs and (v) suggest new therapies to treat diseases (Pavlopoulos *et al.*, 2011).

Biological Network Characteristics

A PIN such as the World Wide Web, as Facebook ® or personal network is a biological network. The biological networks in general, whether of people or proteins, present similar characteristics but different from non-biological networks. Using analysis of graph theory these characteristics can be identified and applied for a better understanding of the network aiming at biological purposes.

Degree: is a measure that indicates the number of edges possess a node. In PIN is a particular characteristic of each protein, indicates with how many other proteins a protein interacts. It is a feature that generally can change the topology of a PIN and have important biological applications. A protein that has a high degree of interaction, which interacts with many other proteins, is called a *hub*. The hub proteins by interacting with many proteins form large tightly connected groups and tend to participate in several metabolic pathways or biological processes, assuming diverse functions.

The hub proteins have important cellular function, indicating that if they are removed from the PIN, the organism may not perform its basic functions, justified by the disruption that the removal of a hub node causes in the network, probably affecting several metabolic pathways (Pavlopoulos *et al.*, 2011; Jeong *et al.*, 2001). *In silico* studies suggest that the hub proteins are under high evolutionary pressure, because (i) they are more conserved among organisms, (ii) they have a lower mutation rate in their sequences, they also have a conserved evolution, and (iii) they are correlated with proteins essential (Dilucca *et al.*, 2017). Just because they are conserved and under selective pressure, it indicates the hub proteins have some singular significance for the organisms.

However, how many interactions a protein must have to be considered a hub? This answer is not given in absolute numbers because the amount of interactions identified in a PIN is dependent on the methodology used, ie in the same organism, different

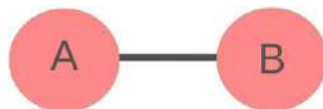


Fig. 1 Nodes represent the proteins (A and B) and the vertex is the interaction between the nodes.

methodologies identify amount of different proteins and interactions. This is a problem even to replicate or validate experiments. For this reason the hub proteins are sorted by relative values, such as 15% (Folador *et al.*, 2016) or 20% (Delprato, 2012) of the protein with a higher degree of interaction in the PIN. There is no consensus on the exact value to define hub proteins and the choice of one or another value depends on how much you want the identification is more or less sensitive.

When a hub protein interacts with its partner proteins simultaneously it is named the Party hub. Party hub generally interact physically with their partners and form complexes performing distinct function, such as proteins composing polymers or homo-polymers. Complementarily, a protein hub that interacts with different partners at different time or place is named date hub. Date hubs are proteins that participate in functional interactions, usually assuming role of regulation in signaling cascades or participating enzymatically in the stages of different metabolic pathways (Hao *et al.*, 2016).

The interactions around hub proteins influence the topology of a PIN and can form two network models, assortative or dissortative. When hub proteins are more connected to other hub proteins, the network is named assortative and, contrary, when connecting to proteins having low interaction degree is named dissortative (Pavlopoulos *et al.*, 2011). Usually biological networks are dissortatives with hub proteins interacting more with proteins having low interaction degree than protein hub (Newman, 2003). These features contribute to the specificity of biological networks and differentiation from non-biological or random networks, and may used as a first layer to in-silico validation by analysis of the interaction degree distribution for PPIs.

Scale-free distribution: It is a topological measure of interaction networks. Biological networks have a scale-free distribution to interaction degree of each node with tendency to power-law (Fig. 2 left), different from random networks showing normal distribution (Fig. 2 right).

The PIN having a scale-free distribution indicate has a large number of proteins with few interactions and a small number of proteins with a large number of interactions (Barabási and Albert, 1999). This fact is in agreement with the presence of hub proteins in PPI concentrating most interactions while other proteins have few interactions. This characteristic indicates biological interactions do not occur randomly or by chance, reinforcing the evolutionary pressure occurs in the interactions, evidencing the importance of precise interactions in the maintenance of life.

Small-world-effect: It is a topological measure of interaction networks. The effect of small world (or shortest path) indicates that any two nodes are separated by only a few edges (Barabasi and Oltvai, 2004), this explains for example that anyone in the world is connected to another on average by 5 or 6 people. In biological networks, this means that the disturbance of any protein will propagate in the network because a few proteins separate all other proteins, suggesting the interactions within a cell are closely related. Certainly, hub proteins contribute for this effect by keeping a strongly connected PIN, reducing the distance between the proteins. Without the hub nodes, the distance between all proteins would probably the same indicating both paths and random network.

Clustering coefficient: in graph theory, is a measure indicating how much the nodes tend to cluster in a network (Pavlopoulos *et al.*, 2011). Clusters are densely interconnected sets of nodes and vertices and the clustering coefficient is used as a feature of biological networks. Generally, PINs present a high clustering coefficient in relation to random networks, forming cohesive groups of interacting proteins acting on the same metabolic pathway. While in PPI the clustering coefficient reaches values greater than 0,4 on random networks is generally less than 0.01 with $p < 0,05$ (Rezende *et al.*, 2012; Folador *et al.*, 2016). Since there are several metabolic pathways in an organism, naturally there are several groups of proteins interacting more intensively in these pathways, giving the PPI a higher clustering coefficient.

Betweenness centrality: is a network centrality measure that calculates the proximity between the vertices in a graph, determining, among the shortest paths to cross the network, how many pass through the node. The higher betweenness value more shortest paths pass through node and enables cross the network side to side in a few steps. In PPI are proteins involved in different biological processes or metabolic pathways, connecting to several groups of proteins without necessarily be a protein hub.

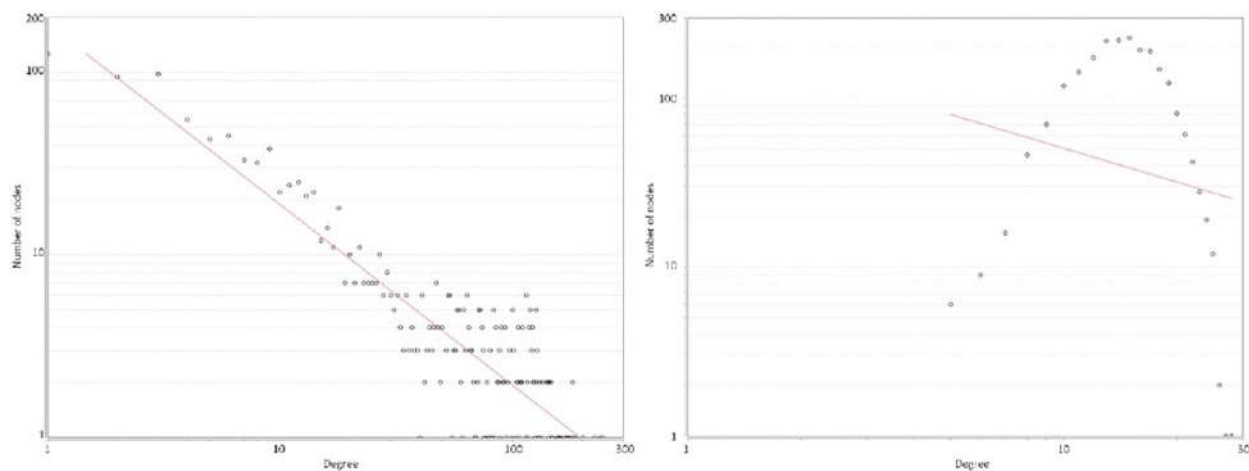


Fig. 2 Sample of interaction degree distribution in scale-free (left) and random (right) networks. The red line indicates a perfect power-law.

However, some hub proteins, because they have many interactions and connect several modules, are also characteristic of betweenness (Pavlopoulos *et al.*, 2011).

The topological measures are features that should be present in PPI generated by experimental or computational methods, and are useful as a first layer to validate the predicted PINs by computational methods. Derived from graph theory, there are several other metrics that can be applied to PPI to extract some biological significance, however without validations described. Having a PPI these and other measures can be easily calculated, e.g., using Network Analyzer plugin (Assenov *et al.*, 2008).

Methods

To investigate protein interactions, both experimental methods in the laboratory (*in vitro* or *in vivo*) and computational (*in silico*) are used. An interaction can be identified by experimental methods or predicted by computational methods, considering biological premises. In the case of PINs, the methods can be classified into (i) low throughput when used to characterize only a few interactions or (ii) high throughput when used to characterize a large number of interactions. Each method class has advantages and disadvantages. Generally, low-throughput methods are more reliable in identifying the interaction while high-throughput methods are more sensitive and identify the entire interaction network.

Experimental Methods

Experimental methods for identifying protein-protein interactions are performed by *in vitro* or *in vivo* techniques. The challenge of the test methods is to identify interactions in their natural state, just they occur in a living cell, without changing properties of proteins or the environment where they are, so as not induce or suppress interactions that naturally would not occur.

Co-immunoprecipitation (Co-IP): Co-IP is able to identify interaction between proteins or even protein complexes by using beads attached to an antibody specific for a target protein, whose interactions we wish to know. The antibodies bound to beads and the cell media are arranged together, establishing the interactions: the target protein interacts with its natural partners and interacts with the antibody. By centrifuging process the complex formed by the bead, antibody, target protein and the partner interaction are pelleted and isolated. The supernatant proteins are discarded and the pelleted proteins can be identified by mass spectrometry or other technique (Lin and Lai, 2017).

Tandem Affinity Purification (TAP): Similar to Co-IP TAP is able to investigate the interactions as they occur in the cell environment and has low contamination rate, although a high-throughput method has an accuracy similar to low-throughput methods. The method consists of attaching the TAP tag to a bait protein, arranged together with other proteins in concentrations similar to that *in vivo*, allowing interactions are established and protein complexes formed. TAP has affinity to the purification column and retains the complex formed with the bait protein in the washing process, when discarded molecules not interacting in the complex. In another purification step, the protein complex is separated from TAP and isolated, allowing the characterization of proteins by mass spectrometry (MS-TAP) (Puig *et al.*, 2001). Due consecutive purification step the method have a deficiency: proteins participating weak or transient interactions may not detected by turning off from TAP complex.

Yeast two-hybrid (Y2H): The two-hybrid method is capable of identifying physical interactions of proteins. Y2H is a method that uses transcription machinery to express a reporter gene. The transcription factor is activated only when the proteins (bait and prey) interact, and the reporter gene is transcribed. Thus, the interaction is perceived by the expression of the reporter gene or its phenotype. Bait is a target protein anchored in the DNA-binding domain which in turn binds to the promoter region Upstream Activation Sequence (UAS) of DNA. Prey are the Bait partner proteins, anchored to the transcriptional activation domain (TA), which will be interrogated. When Bait and Prey interact the transcription process is activated and the gene reporter expresses (Ito *et al.*, 2001). This process can be reproduced on a large scale and investigate the entire interactome.

Bimolecular fluorescence complementation (BiFC): is a technology used to identify physical protein interactions. It is based on the use of a fragmented fluorescent protein (e.g., enhanced yellow EYFP) in two non-fluorescent peptides, respectively the N- and C-terminal peptides, each one attached to a specific known target protein by a linker. When these two target proteins expressed on the live cell interacts, the N- and C-terminal peptides became near and interact too, reformulating to the native three-dimensional structure da fluorescent reporter protein (e.g., EYFP). The emitted fluorescent signal can detected within the cell by an inverted fluorescence microscope making possible identify the location of interacting proteins and known the interaction type by the signal intensity (Wang *et al.*, 2017).

Cross-linking coupled with mass spectrometry (XL-MS): The cross-link are markers connecting a polymer chain to another (e.g., protein-protein), usually developed by the use of reagents or cell engineering and can modify the characteristics of the investigated protein or its natural environment. Several methodologies researching *in-vivo* interactions use different cross-link to identify the interaction. The cross-linking form covalent bonds and stabilizes the interaction between proteins, making it possible to identify even the weak or transient interactions by mass spectrometry (Holding, 2015).

Laser UV Cross-linking (LUCK): The major difference of the LUCK method is to use as cross-link the aromatic residues (Phe, Trp, Tyr). These residues are naturally present in most proteins, especially at binding sites, establishing strong bonds, having the advantage of not modifying the characteristics of the investigated protein or its natural environment. When irradiated with UV light these aromatic residues produce radicals that react and generate covalent bonds with near residues, making it possible

to identify even the transient interactions. The LUCK has already been used both to identify the formation of stable dimers of GAPDH in HeLa Cell (Itri *et al.*, 2016) as to detect the protein interaction between GAPDH and alpha-enolase in living cells (Itri *et al.*, 2017).

Fluorescence resonance energy transfer (FRET): is a method that uses different fluorescent proteins (FP) to identify interactions, having potential for unveils temporal and spatial information on proteins in living cells. Both FP are named donor and acceptor respectively, which when excited by a specific wavelength range emit in response a different specific wavelength range. Thus, considering a donor with excitation and emission peaks at X and Ynm respectively and a receptor with excitation and emission peaks at Y and Znm respectively, the interaction is identified when perceive peaks at Znm when excited at Xnm. Justified by energy transfer from donor to receiver when the signal emitted by the donor excites the near receiver.

Several other methods are described in the literature taking advantage of the physicochemical property for label proteins and permits identify interactions by diverse techniques, including sequencing transcripts as bPPI-seq method (Zhang *et al.*, 2017).

Computational Methods

The computational methods, also known as *in silico*, uses computational tools to solve biological problems also considered a more viable alternative due to its low cost when compared with experimental methods. Another advantage of *in silico* approaches is the high capacity to generate and analyze data inexpensively. Due to the rapid development in computing field is increasingly feasible the use of more powerful techniques to study biological networks.

The use of previously unviable or non-existent methodologies, such as the use of high-performance computing, has been increasingly frequent, as well as the emergence of new and more effective methodologies (Keskin *et al.*, 2016). Aided by the amount of information and biological data available added to the evolution of computers and algorithms have emerged methods for PPI prediction based on diverse approaches as binding strength between molecules, sequence alignment, phylogenetic trees, text mining, artificial intelligence and even using mixed approach.

Three-Dimensional Structure-Based Interaction Prediction

Bond strength methods use three-dimensional structures of two proteins to find a contact region (binding site) and measure the affinity between these two structures. In order, to reduce the computing power required for docking, usually the protein structures are considered a rigid body, differing from biological reality but enabling the prediction be performed quickly to a pair of proteins. Conversely, molecular dynamics considers the protein as a flexible body and besides calculating the affinity between molecules also calculates the angles of each atom taking into account the possible movements that the proteins can present recursively, consequently increasing the computational requirements. To achieve a few milliseconds with molecular dynamics, it takes days of processing, depending on the computational power available (Keskin *et al.*, 2016).

These methods are considered low throughput mainly for two reasons: (i) the high computational power and time required especially when using molecular dynamics technique, or (ii) a lack of three-dimensional structures determined experimentally to be used with docking or dynamic molecular techniques compared to the amount of available sequences.

The processing time can be considerably reduced when the docking or molecular dynamics programs are able to use the resources of the graphic processing units (GPUs) that, in addition to being accessible, are currently in the order of 5000 processing units, exceeding the amount of central processing unit (CPU) available on good and expensive servers. To take best advantage of both techniques, docking is usually performed to bind two proteins and then runs the molecular dynamics to optimize the interacting partner proteins in their less energy state.

A software widely used for molecular dynamics is GROMACS (Abraham *et al.*, 2015) while docking can be performed with HEX (Ritchie, 2003), MegaDock (Ohue *et al.*, 2014) and HADDOCK (De Vries *et al.*, 2010), SwarmDock Server (Torchala *et al.*, 2013), ZDOCK Server (Pierce *et al.*, 2014), Multi-IZerD (Esquivel-Rodríguez *et al.*, 2012), LightDock, ClusPro (Jiménez-García *et al.*, 2017; Kozakov *et al.*, 2017), among others.

In order to encourage the development of new algorithms and methods as well as to improve existing ones, the Critical Assessment of PRedicted Interactions (CAPRI) was created in 2000. The CAPRI is an open event that occurs in competition format to evaluate the algorithms in blind predictions of the structure of protein-protein complexes. Prior to publication, the organizers make available to competitors proteins offered by crystallographers as targets, without prior knowledge of the complex. The competitors perform the complex structure prediction by docking the individual components. Afterwards, the predicted models are submitted to evaluators compare the geometry and interaction site against the experimental models, scoring the best predictors.

Multiple Sequence Alignment (MSA)-Based Interaction Prediction

With the popularity of DNA sequencers, many genome projects were undertaken, and various organisms had their genome sequenced and annotated. Many genomes as their amino acid or nucleotide sequences are available in several public databases, and the GenBank, EMBL and DDBJ are the largest repositories of such information. In contrast to the low number of three-dimensional structures, the number of sequences in the public databases is extremely higher, favoring the scientific community that uses these sequences in their researches, including the prediction of interaction. A class of method uses multiple sequence

alignment (MSA) to predict protein-protein interaction and each is based on a biological premise that generally aims to identify conservation among the organisms' sequences using alignment software. They are:

Phylogenetic profile-based prediction (co-occurrence): The method is based on the biological premise that if two genes remain conserved when comparing several organisms, the respective protein translated from these genes will interact. This assumption becomes even stronger when both conserved genes are not present in one or more organisms compared, demonstrating selective pressure for both remain or be removed together from genome. The uncertainty in using this method is to identify how phylogenetically distant should be the organisms compared therefore, by selecting close organisms makes it difficult to identify whether proteins are conserved because they are under evolutionary pressure or simply because there has not been enough time to diverge. The more distant organisms are, the more reliable the prediction will be, but fewer interactions will be identified (Valencia and Pazos, 2002).

Neighborhood genes-based prediction: This method is an evolution of the previous one, considers the same premise and presents the same doubt, but considers that the genes, besides being conserved, must be neighbors. This method prioritizes to identify conservation of closer genes taking into account they tend to evolve in similar periods, like operons transcribed and translated at the same time, probably generating protein structures interacting in a same function or metabolic pathway (Valencia and Pazos, 2002).

Phylogenetic tree-based prediction (Mirrortree): The method is based on the biological premise that if two proteins interact they undergo similar evolutionary pressures and co-evolve in an organism (Zhou and Jakobsson, 2013). Thus, when comparing the same protein in several organisms, the respective interacting proteins, will present a similar phylogenetic tree, one mirrored in the other, representing similar evolution in the different organisms. However, proteins evolving differently may also interact and be disregarded by this method. To identify an interaction phylogenies of various proteins are made and, the phylogenetic trees presenting similar topology reveals the respective proteins participating in the interaction (Pellegrini et al., 1999). This can be made feasible by converting pairs of phylogenetic trees into distance matrix and calculating the correlation coefficient. Matrices with high correlation coefficient indicates a phylogenetic tree with similar topology.

Gene fusion-based method: The biological premise underlying this method is the existence of two genes (ga, gb) in an organism with functional domains homologous to a single gene in another phylogenetically close organism (gc), probably the proteins encoded by the two individual genes (pa and pb) interact physically to play their role, explained by an evolutionary process that led to the fusion of genes and translation of a single protein into another organism (pc). The homology between genes can be identified via sequence alignment and this assumption becomes even stronger when one perceives in multiple alignment a group of organisms presenting the two individual genes and another group presenting the fused genes (Zahiri et al., 2013).

Interolog Mapping

This method is based on the biological premise that if it is known the interaction between two proteins in a particular organism and exist both orthologous proteins in another organism, they probably also interact. It is a high throughput method by allowing mapping known available interactions from public databases to an organism of interest using sequence alignment. The mapping of the interactions is performed in two steps: (i) identification of the orthologous genes or proteins by aligning the sequences from public databases against the sequences of your organism and (ii) composition of the orthologous pairs of interacting proteins based on interactions from public database. Although simple, due to the large volume of sequences and interactions in some databases, in the order of millions, it may require considerable computational resource for the prediction process. The confidence of the predicted interaction will depend on two factors: (i) that truly homologous proteins are mapped and, (ii) how reliable is the known interaction from public databases. Despite being an effective and easy to use method, it is able to map the interactions already determined in other organisms, without the ability to predict new interactions not yet reported.

Text Mining

The text mining is a computational technique used to obtain important information recorded in texts such as abstracts or scientific articles, having commercial, governmental and scientific applications. Briefly, if an interaction is described in a text, it is theoretically possible to be identified using different algorithms. In summary, a described interaction theoretically can be identified using different techniques and text mining algorithms (Zahiri et al., 2013). The identification of interactions is performed in four steps: (i) download of library to be used in the research, in the case summaries and articles containing information about proteins or genes; (ii) identifying entities (genes or proteins) of interest; (iii) identification and verification of relationship between the entities in sentences and paragraphs, in the case indicating interaction; and (iv) extraction of entities to form the interaction pair. As interolog mapping this is a high throughput method able to identify interactions previously described, without the ability to identify new interactions. To reduce the sampling range and increase reliability in prediction, the literature is usually retrieved from a specific species, limiting the method to identify specific interactions. Another limiting factor is how the entities are described (e.g., the gene name, the protein name, codes from different databases), if untreated, can limit the method does not to identify all possible interactions and generate a lot of false-negative. However, it can be a very useful method, especially when it is desired to know the relation of the interactions, usually identified by action verbs in sentences (e.g., bind, interact, regulate), making it more informative. Interactions identified experimentally and published, when predicted by text-mining method and curated, in many cases have experimental equivalence, since it was first identified by an experimental method.

Machine Learning

Machine learning is an Artificial Intelligence subfield of the Computing area. The algorithms form models and patterns from sets of gold standard data where the different characteristics that determine an interaction are already known. The algorithms are executed in two steps: (i) learning, where the characteristics are observed and identified in a set of gold standard data whose interactions are known aiming to establish a pattern e ; (ii) prediction, where the characteristics are checked on a real data set to determine whether an interaction exists or not, or even establish a degree of confidence in the interaction. In PPI can be used diversified characteristics such as genetic or protein sequence, physicochemical data, three-dimensional structures, cellular location, binding sites, conserved regions or even the combination of these characteristics. Prediction confidence is dependent on learning, ie if the algorithm is able to learn and determine all the relevant characteristics for an interaction to exist, it will be able to infer whether an interaction is possible or not based on these characteristics. Due to the large number of interactions available, experimentally determined or not, it is possible to create a set of interacting proteins to train the algorithms (Dick *et al.*, 2017). For an efficient training, it is important to highlight and imperative the algorithm learns which characteristics determine an interaction, as well as the characteristics determining not interaction, that is, for the learning, besides a positive control set a negative control set is necessary. This is plausible since there are cured databases of proteins that do not interact as The Negatome Database (Blohm *et al.*, 2013).

Abstractly, we can break any prediction method down into.

Sample description: how each sample can be described to the method. Not everything is accepted by all methods. For instance, in simple linear regression, each sample can only be described by two numbers (x , y). *Model*: every method assumes a certain structure between the samples. For example, in linear regression, that assumption is that the set of samples can be somewhat accurately represented by a straight line. Clustering, on the other hand, assumes “related” samples are also “close” in the sample space. *Parameters*: variables of that prediction method. *Training*: is the process of finding the best parameters that fit the samples, for some interpretation of “best”, how to find the right (or best) parameters. Training might be “supervised”, where previously classified data is used to find the appropriate parameters, or it might be “unsupervised”, when no such premade classification exists. *Prediction*: how predictions are made. Different methods accept different kinds of questions and output their predictions in different ways. Questions may include: “Given a sample with incomplete data, what best fits the blanks?” or “How well does this new sample fit?”. Predictions might be in the form of an yes/no answer, a real number, a label, or a mix of any of these.

These categories will be used to describe the prediction methods above.

Linear regression

Sample description: two (simple linear regression) or more numbers (multivariate linear regression). Linear regression assumes certain statistical properties over the training samples. Those properties can vary depending on which regression is used out of many variants, but may include assumptions such as weak exogeneity or homoscedasticity. *Model*: samples are represented as a straight line that is approximately close to all samples. *Parameters*: a and b where $y = xa + b$. In simple linear regression, a and b are real numbers. In multivariate linear regression, a and b are $m/\text{cross } n$ matrices, where m is the number of independent variables and n the number of independent variables. *Training*: always supervised and accepting only positive examples. The number of published and used estimators are too great to list here, but includes the Ordinary Least Squares method, the Maximum-Likelihood Estimation, the Bayesian Linear Regression, and the Theil-Sen Estimator. *Prediction*: given how the training phase yields a line, the distance to that line is a natural metric to evaluate how fit any new sample is. New samples can be taken from the line. *Examples*: Logistic regression (a generalization of linear regression for discrete dependent variables) was evaluated against other five methods with 16 feature categories used for sample description, including: gene expression, GO molecular function, GO biological process, GO component, and essentiality (Qi *et al.*, 2006). Logistic regression was used to create a confidence score for experimental data on protein-protein interactions. The confidence score was calculated based on the results of Luminescence-based Mammalian IntERactome (LUMIER), Mammalian Protein-Protein Interaction Trap (MAPPIT), Yeast-2-Hybrid (Y2H), Yellow Fluorescent Protein (YFP), Protein Complementation Assay (PCA), and a modified version of the Nucleic Acid Programmable Protein Array (wNAPPA) assays (Braun *et al.*, 2009).

k-means clustering

Sample description: each sample is described by a real vector with a fixed dimension. *Model*: the sample space is partitioned into k clusters and each sample belongs to a single cluster. Sample variance is minimized within a single cluster (Bishop, 2006). *Parameters*: the centroid of each cluster. *Training*: supervised, unsupervised, or a mixture of both. In the worst case (unsupervised), finding the absolute best centroids is NP-hard, requiring training to run for a time proportional to the exponential of the number of samples. Because of this, heuristics are usually used, giving “good enough” centroids most of the time. *Prediction*: k-means clustering naturally classifies samples in one of k categories. The centroid also serves as the archetype of each cluster. *Examples*: Mohamed and colleagues (Mohamed *et al.*, 2010) used k-means clustering where each sample, describing a protein-protein interaction, was represented by a vector with 27 dimensions describing features such as GO molecular function, GO biological process, GO component, and sequence similarity. K-means clustering was used as a method for dimension reduction, where each sample was the result of multiple liquid chromatography-mass spectrometry (LC-MS) alignments, each giving noisy indications of protein-protein interactions (Rinner *et al.*, 2007).

Hierarchical clustering

Sample description: anything with a meaningful distance function. *Model:* a hierarchy of clusters. Each cluster may encompass samples and other clusters, but each cluster can be encompassed by at most a single parent cluster. *Parameters:* edges and internal nodes, forming a tree. *Training:* supervised, unsupervised, or a mixture of both. Two strategies are usually used: agglomerative, where samples are grouped into clusters, or divisive, where a cluster encompassing all samples is recursively subdivided. There are many ways to decide whether to agglutinate or split, called the linkage criteria, including Complete-Linkage Clustering, Single-Linkage Clustering, and Minimum Energy Clustering. *Prediction:* the training result cannot be directly used for prediction. Instead, any new sample is added to the initial sample set and clusters are regenerated. *Examples:* Brohée and colleagues (Brohee and Van Helden, 2006) evaluates different clustering algorithms for protein-protein interaction networks. General properties of protein-protein interaction networks are noted based on their hierarchical clusters (Yook et al., 2004).

Support vector machine (SVM)

Sample description: each sample is described by a real vector with a fixed dimension. *Model:* samples separated into two areas by a hyperplane, optionally put in a transformed space. *Parameters:* a hyperplane, the minimum separation between the hyperplane and training samples (soft margin), and an optional transformation function (Haykin, 1994). *Training:* supervised only. Given a set of positive and negative samples, the parameters of usual transformation functions and the hyperplane can be found by solving a quadratic optimization problem, which is NP-hard and takes exponential time relative to the number of samples. Iterative alternatives include Sub-Gradient Descent and Coordinate Descent. *Prediction:* any sample can be classified as positive or negative by applying the transformation and checking whether the transformed point lies in the positive or negative half-space. Extensions allow for a broader classification than only positive/negative. *Examples:* Guo and colleagues (Guo et al., 2008) uses an SVM to model and predict protein interactions based on many physicochemical properties of their amino acids: hydrophobicity, hydrophilicity, volumes of side chains of amino acids, polarity, polarizability, solvent-accessible surface area, and net charge index of side chains of amino acids. Koike and colleagues uses a SVM to predict the interaction sites of proteins based on their sequence of amino acids (Koike and Takagi, 2004).

Hidden markov model (HMM)

Sample description: a sequence of symbols in a fixed alphabet. The alphabet need not be ordered (Bishop, 2006). *Model:* an HMM is composed by a set of distinct states, a set of normalized (summing to 1) probabilities for the state to change from any state to any other state (including itself), and a set of normalized probabilities that an observation occurs given a state. Alternatively, an “observation” can be seen as a letter in an appropriate alphabet that the model “emits” one at a time based on its current state and a random number. *Parameters:* transition and observation probabilities. *Training:* supervised or unsupervised. Given a set of states and possible observations (or emissions), the expectation-maximization algorithm is usually used to, iteratively, find the set of probabilities. Finding the “best” probabilities is possible but there is no known efficient algorithm for that as the problem is NP-hard. *Prediction:* a trained HMM can score sequences based on their similarity to sequences used in training. HMMs can also run in reverse, giving sequences that are likely similar to the ones used in training. *Examples:* In (Wojcik and Schächter, 2001), HMMs are used to predict protein-protein interaction (PPI) maps given a PPI map of another organism. Eddy (Eddy, 1998) gives a review of Profile Hidden Markov Models (pHMM) which can perform sequence alignment based on similarity to certain sequences.

Artificial neural network (ANN)

Sample description: any type of data with a fixed size. Depending on how the network is built, variably-sized data can be used as well (Haykin, 1994). *Model:* a set of artificial neurons forming a network, where each one combines, transforms, and filter incoming data, passing the result forward. *Parameters:* network topology, which can be premade, and parameters for each artificial neuron. *Training:* supervised and unsupervised. Training procedure is highly dependent on network type and topology, but most used are Gradient Descent, Simulated Annealing, Evolutionary Algorithms, Particle Swarm Optimization, and others. Training an ANN usually takes considerable computer resources and time. *Prediction:* any kind, depending on topology and how it was trained. *Examples:* An ANN is used to predict interaction sites between proteins given a sequence of aminoacids (Ofra and Rost, 2003). Protein are given a predicted function with an ANN. The authors also suggest that predicted function can be used to validate protein-protein interaction databases that might contain false negatives (Jensen et al., 2002).

As each predictive computational method considers different biological premise, it can of course disregard biological factors relevant to identification of interaction, thus, a good practice, is to use predictive computational methods together in order to guarantee greater sensitivity in the PIN predictions. This strategy besides useful to add information to the PIN is useful to validate interactions, because an interaction identified by more than one method reduces prediction by chance and increases the probability of being true interaction. Similarly, when possible, using both experimental and computational methods can also aggregate information to the PIN.

Different methods predicted PINs for various organisms whose interactions usually are deposited in public databases.

Public Databases

Aiming to make public and share data interactions for the scientific community, several databases were designed, containing interactions identified in different organisms by experimental and computational methods. With the wide variety of organisms and methods for PPI prediction also came the need to distinguish these public databases reliability of interactions. Thus, arose in September 2005 the International Molecular Exchange Consortium (IMEx) as an international collaboration having as participants several public databases joining efforts to validate protein-protein interactions through curation, ensuring a detailed and highly reliable set of non-redundant interactions to be made publicly available (Orchard *et al.*, 2012).

Public databases store and provides physical or functional interaction data experimentally characterized or predicted computationally or even cured, some store interaction of various organisms while others store specific organism interaction or even specialized in biological process or functionality (Table 1).

The classes of positive and negative interactions are important biologically but also computationally. Besides being useful at systems biology level for better understand an organism, they serve as training data for machine learning algorithms, still useful as the gold standard to validate computational predictive methods (Dick *et al.*, 2017). More details of methods and databases can be observed in the work of Rao and colleagues (Rao *et al.*, 2014).

The Minimum Information About a Molecular Interaction Experiment (Mimix)

Recently, to publish a PIN article, magazines began to suggest the interactions identified in the experiments were deposited in a public database, a great initiative to provide and share information with the scientific community. However, with the diversity of databases, it was also realized the need to standardize the information concerning the PIN experiments that would be deposited aiming to facilitate the comparison, exchange and verification between these databases. Thus, was developed a guide advising how to describe a molecular interaction experiment and which information it are important (Orchard *et al.*, 2007), available on the HUPO Proteomics Standards Initiative – “see Relevant Website section” (access in 2017-10-26).

Thus, interactions from the most diverse experiments, studied and applied for biological purposes are deposited in the various public databases.

PIN Applications

The PINs have various biological applications such as providing subsidies for characterization of hypothetical proteins; understand systematically an organism, process or metabolic pathway; identification of key proteins or even suggest potential targets for developing new drugs.

Protein Annotation

The no knowledge about the function of certain proteins prevents the understanding of an organism in relation to its development, proliferation, virulence and biological processes thus, annotation of hypothetical proteins is essential for a better

Table 1 Protein-protein interaction public databases

Name	Organisms	Interaction type (Física/funcional)	Protein and interaction amount	Method type (Experimental, computacional)	Curated	Link
String v 10.5	2031	Both	P: 9,6 million I: 1380 billions	Both	No	https://string-db.org/
DIP	834	Both	P: 28,826 I: 81,762	Ex	Yes (IMEX)	http://dip.doe-mbi.ucla.edu/dip/Main.cgi
Intact v 4.2.10	Organisms model	Both	P: 93,720 I: 786,143 2017/09	Text-mining	Yes (IMEX)	https://www.ebi.ac.uk/intact/
MINT	611	Both	P: 25,530 I: 125,464	Text-mining	Yes (IMEX)	http://mint.bio.uniroma2.it/
I2D	Five model organisms and human	Both	P: N/A I: 1,279,157	Co	Yes (IMEX)	http://ophid.utoronto.ca/ophidv2.204/index.jsp
MATrixDB	Extracellular matrix proteins, proteoglycans and polysaccharides interactions	N/A	P: 14,902 I: 15,018	Ex	Yes (IMEX)	http://matrixdb.univ-lyon1.fr/
InnateDB v 5.4	innate immune response of humans, mice and bovines to microbial infection	Both	P: 25,110 Ic: 367,503 Ip: 462,421	Ex	Yes (IMEX)	http://www.innatedb.com/
Negatome v 2.0	non-interacting protein	Phy	P: N/A I: 30,756	Co	Yes	http://mips.helmholtz-muenchen.de/proj/ppi/negatome/

understanding of the organism. Generally, protein annotation is performed by sequence alignment and homology identification against proteins from public databases, but even with the large number of sequences not all proteins have their function annotated, requiring other approaches.

Currently, proteins having similar or complementary functions are known to be in close proximity on genome and tend to be transcribed together such as operons. Thus, it is expected these proteins are located in a cluster on the PIN and are closely connected among themselves (Pellegrini *et al.*, 2004).

As PIN provides knowledge of the various proteins working together in biological processes, it is sometimes possible to deduce the function of a hypothetical protein based on the function of its interaction partners in the cluster (Marcotte *et al.*, 1999). This method was applied in *Eriocheir sinensis* and based on PIN modularity feature the functions of 677 proteins were annotated (Hao *et al.*, 2015).

System Biology

The identification and annotation of IPP is important to understand how proteins form complexes to perform tasks inside the cell, subsidizing to understand by the set of interactions, how the organism works systematically. A metabolic pathway can be defined as a set of actions or interactions between genes and their products that results in the formation or change of some component of the system, essential for the correct functioning of a biological system. Thus, the identification of molecules and pathways in which they are involved is essential in the studies seeking to discover mechanisms involved in phenotypes and diseases (Hu *et al.*, 2017).

The PIN use is becoming increasingly common and essential for helping to elucidate biological processes clarifying the molecular origins of many diseases. As interactions are extremely disturbed by diseases such as cancer, heart disease and neurodegenerative diseases, the data generated by PIN provides identifying reasons to study them in living cells, enabling more accurate results and near reality (Itri *et al.*, 2017).

Based on gene modules constructed from human PINs to measure the relationship between pathways, a total of 2143 KEGG pathway pairs with close connections were identified, whose results were consistent with available evidence. The method was applied to explain the potential pathway targets that may be important in the etiology and development of Parkinson's disease (Hu *et al.*, 2017). The PIN usage in *Borrelia burgdorferi* allowed to discover two proteins with important functions related to infection and the biology of the organism, having different binding sites the interaction between proteins controls the infectivity of the organism (Thakur *et al.*, 2017).

PINs are important tools enabling better understanding of an organism mainly when added to other layers of functional and experimental data such as cell location or gene expression.

Essential Proteins

For optimal functioning the organism needs proteins guaranteeing primary resources for survival, and the lack or diminution of these proteins may be deleterious (Winzeler *et al.*, 1999). High-interaction proteins (hubs) are usually related to the essential genes in organisms, having important cellular functions form densely connected clusters whose removal causes a great rupture in the PIN and the organism may not perform its basic functions (Pavlopoulos *et al.*, 2011).

Recently *in silico* experiment showed the hub proteins are correlated with essential genes, they are highly conserved among species and possess low mutation rate, probably due to strong evolutionary pressure receiving (Dilucca *et al.*, 2017). With these features, the hub proteins are potential targets for drugs, offering new possibilities to research effective drugs for pathogenic organisms (Becker and Palsson, 2005).

Another *in silico* work identified 181 hub protein in nine *Corynebacterium pseudotuberculosis* strains, 180 with proven essentiality in other organisms. Because they were highly conserved, only 41 had no homology to the host and were considered promising targets for drugs (Folador *et al.*, 2016).

Host-Pathogen Interaction

The PINs still provide the possibility to investigate host-pathogen interactions and can be applied to understand the relationship between pathogen and host at systems biology level as suggesting proteins or interactions as potential targets for drug. Aspects related to virulence such as adhesion, escape of the immune system, proliferation are often dependent on the host-pathogen interaction and the knowledge of these mechanisms, even partially, can be determinant to understand the infection and to guide new experiments.

In addition to intracellular PIN, should be considered spatio-temporal aspects of host-pathogen PIN prediction, that is, the proteins participants of interaction must coexist and must be in cellular location favoring interaction, being potential participants in host-pathogen interaction the membrane, transmembrane or secreted proteins. The identification of host-pathogen interaction mainly promotes research into new interaction-inhibitors drug preventing infection be harmful to the host.

The host-pathogen PINs were used to identify interactions among *Mycobacterium tuberculosis* and *Homo sapiens* (Huo *et al.*, 2015). Using the SVM method was predicted the PIN of viruses with *Homo sapiens* as interactions of both hepatitis C virus and papillomavirus against humans obtaining 80% accuracy (Cui *et al.*, 2012).

In the study between *Nipah* virus and *human* 101 interactions were identified, 88 never found previously, being the main targets the miRNA processing machinery and the preprocessing factor of mRNA PRP19, including proteins capable of altering the p53 gene expression (Martínez-Gil *et al.*, 2017). Many targets for existing drugs are contained in a small number of families as guanine nucleotide-binding protein (G-protein) – acute receptor, nuclear hormone receptor, ionic channel kinase, and protease.

New Drug Classes

Many targets for existing drugs are contained in a small number of families as guanine nucleotide-binding protein (G-protein) – acute receptor, nuclear hormone receptor, ionic channel kinase, and protease (Groom and Hopkins, 2002; Santos *et al.*, 2017). Discovering new drugs has become a difficult task as known targets in the human proteome are increasingly limited being PINs the future targets for new drug discovery (Shin *et al.*, 2017). To overcome the difficulty to find new targets for drugs, researchers expect PIN become drug targets for diseases not yet having effective treatments such as Parkinson's and Alzheimer's disease (Milroy *et al.*, 2014). It is estimated in the human interactome there are between 130,000 and 650,000 interactions being drug targets only a small percentage (Venkatesan *et al.*, 2009; Rognan, 2015). Only in recent years more than 40 interactions were targeted and many inhibitors are in clinical trials phase (Arkin *et al.*, 2014), however, more than 90% of the interactions in humans have not yet been deciphered, having wide scope for research and improvement of predictive methods (Shin *et al.*, 2017).

Classical drugs tend to inhibit proteins that are sometimes essential in other biological processes and can cause adversities by not acting on an exclusive component (Petrakis and Andrade-Navarro, 2016). Due to the great structural and conformational variability the interactions emerge as an important target for specific drug development that act by stabilizing or inhibiting a distinct interaction sites of an exclusive component (Bultinck *et al.*, 2012). An example is the use of docking to identify inhibitory molecules of the Aurora Kinase-TPX2 PPI and prevent the growth of malignant tumors such as colon and stomach cancer, preventing the multiplication of cancerous cells (Cole *et al.*, 2017). Research combating cancer identified the molecules NVP-CGM097 and NVP-HDM201 with affinity to MDM2 and with potential p53-MDM2 PPI inhibitor, already in phase 1 clinical development (Holzer, 2017). A successful example is the SMPPiIs inhibitors acting on bromodomain, four in clinical trials for cancer (I-BET762, CPI-0610, Ten-010 and OTX15) for the potential of binding to the bromodomain hydrophobic core and inhibiting the interaction between the two parts of bromodomo-histone (Arkin *et al.*, 2014).

The NusB-NusE interactions, fundamental to the formation of stable complexes required for RNA transcription, have antibiotic activity and are potential emerging antibacterial targets (Cossar *et al.*, 2017). Another example is 14-3-3s family proteins, abundant in the central nervous system, bind to serine and threonine regions when phosphorylated, are related to certain diseases and the use as a target for drug showed therapeutic effects (Kaplan and Fournier, 2017).

However, the identification of molecules inhibiting or stabilizing interactions presents challenges, mainly because they do not present traditional drugs characteristics as the Lipinski's five rules. The PPI interfaces are distinct from traditional targets, both geometrically when because the interactions are largely dominated by hydrophobic atoms (Shin *et al.*, 2017), being the natural compounds products potent candidates in the drug design targeting protein interface (Jin *et al.*, 2017).

Conclusions

Besides the work explained in this article, there are other public databases containing PINs from a diversity of sources, specialized in some specific organism. Similarly, there are other methods for identification or prediction of PINs and following technological developments, others are likely to be developed. Regardless of whether it is experimental or computational, low-throughput methods present greater scientific reproducibility than high-throughput methods, identifying large scale interactions. The low reproducibility of high-throughput experimental methods is probably due to the inability to reproduce a physico-chemical cellular environment similar to that occurring *in vivo*, especially with respect to variables that are not easily controlled as the space and time the interactions are carried out. For high-throughput computational methods, the low reproducibility is mainly motivated by the different and specific biological approach of each method, which use different logic, algorithm or data input, consequently generating different results. This shows that we still have a lot to learn and develop until we get PINs similar to those in nature.

PINs are applied for various purposes ranging from understanding an organism, understanding a metabolic pathway or process systematically, identifying proteins or interactions with potential use as a target for drugs or even understanding the relationship between pathogen and host. However, in order to obtain better results, it is not enough to only have PINs, it is necessary to aggregate layers of information from the different omics to the network, making possible a more complete interpretation within a biological context. Although, there are tools for visualization, formatting and analysis of PINs, the biological interpretation is usually done by a specialist researcher, with a broad knowledge of the biological context for which the network was built or knowledge about the organism. A PIN is not usually the final object of a search, but rather a tool that helps researcher in the organization, visualization, and integration of data to generate knowledge. Combining the researcher's expertise and knowledge, a PIN can serve to raise new hypotheses and to direct more assertive experiments in the laboratory.

See also: Alignment of Protein-Protein Interaction Networks. Community Detection in Biological Networks. Computational Approaches for Modeling Signal Transduction Networks. Computational Systems Biology. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Functional Genomics. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Network Models. Network Properties. Network-Based Analysis of Host-Pathogen Interactions. Networks in Biology. Pathway Informatics. Prediction of Protein Localization. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Post-Translational Modification Prediction. Protein Properties. Protein-Peptide Interactions in Regulatory Events. Protein-Protein Interaction Databases. Protein-Protein Interaction Databases. The Evolution of Protein Family Databases

References

- Abraham, M.J., Murtola, T., Schulz, R., *et al.*, 2015. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1, 19–25.
- Arkin, M.R., Tang, Y., Wells, J.A., 2014. Small-molecule inhibitors of protein-protein interactions: Progressing toward the reality. *Chemistry & Biology* 21, 1102–1114.
- Assenov, Y., Ramírez, F., Schelhorn, S.-E., Lengauer, T., Albrecht, M., 2008. Computing topological parameters of biological networks. *Bioinformatics* 24, 282–284.
- Barabási, A.-L., Albert, R., 1999. Emergence of scaling in random networks. *Science* 286, 509–512.
- Barabási, A.-L., Oltvai, Z.N., 2004. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics* 5, 101.
- Becker, S.A., Palsson, B.O., 2005. Genome-scale reconstruction of the metabolic network in *Staphylococcus aureus* N315: An initial draft to the two-dimensional annotation. *BMC Microbiology* 5, 8.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*. Springer.
- Blohm, P., Frishman, G., Smailowski, P., *et al.*, 2013. Negatome 2.0: A database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *Nucleic Acids Research* 42, D396–D400.
- Braun, P., Tasan, M., Dreze, M., *et al.*, 2009. An experimentally derived confidence score for binary protein-protein interactions. *Nature Methods* 6, 91–97.
- Brohee, S., Van Helden, J., 2006. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics* 7, 488.
- Bultinck, J., Lievens, S., Tavernier, J., 2012. Protein-protein interactions: Network analysis and applications in drug discovery. *Current Pharmaceutical Design* 18, 4619–4629.
- Chen, D., Jansson, A., Sim, D., Larsson, A., Nordlund, P., 2017. Structural analyses of human thymidylate synthase reveal a site that may control conformational switching between active and inactive states. *Journal of Biological Chemistry* 292, 13449–13458.
- Cole, D.J., Janecek, M., Stokes, J.E., *et al.*, 2017. Computationally-guided optimization of small-molecule inhibitors of the Aurora A kinase-TPX2 protein-protein interaction. *Chemical Communications* 53, 9372–9375.
- Cossar, P.J., Abdel-Hamid, M.K., Ma, C., *et al.*, 2017. Small-molecule inhibitors of the NusB–NusE protein-protein interaction with antibiotic activity. *ACS Omega* 2, 3839–3857.
- Cui, G., Fang, C., Han, K., 2012. Prediction of protein-protein interactions between viruses and human by an SVM model. *BMC Bioinformatics* 13, S5.
- Delprato, A., 2012. Topological and functional properties of the small GTPases protein interaction network. *PLOS ONE* 7, e44882.
- De Vries, S.J., Van Dijk, M., Bonvin, A.M., 2010. The HADDOCK web server for data-driven biomolecular docking. *Nature Protocols* 5, 883.
- Dick, K., Dehne, F., Golshani, A., Green, J.R., 2017. Positome: A method for improving protein-protein interaction quality and prediction accuracy. In: *Proceeding of 2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–8. IEEE.
- Dilucca, M., Cimini, G., Giansanti, A., 2017. Topological transition in bacterial protein-protein interaction networks ruled by gene conservation, essentiality and function. *arXiv preprint arXiv:1708.02299*.
- Eddy, S.R., 1998. Profile hidden Markov models. *Bioinformatics (Oxford)* 14, 755–763.
- Esquivel-Rodríguez, J., Yang, Y.D., Kihara, D., 2012. Multi-LZerD: Multiple protein docking for asymmetric complexes. *Proteins: Structure, Function, and Bioinformatics* 80, 1818–1833.
- Folador, E.L., De Carvalho, P.V.S.D., Silva, W.M., *et al.*, 2016. In silico identification of essential proteins in *Corynebacterium pseudotuberculosis* based on protein-protein interaction networks. *BMC Systems Biology* 10, 103.
- Groom, C.R., Hopkins, A.L., 2002. Protein kinase drugs—optimism doesn't wait on facts. *Drug Discovery Today* 7, 801–802.
- Guo, Y., Yu, L., Wen, Z., Li, M., 2008. Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Research* 36, 3025–3030.
- Hao, T., Peng, W., Wang, Q., Wang, B., Sun, J., 2016. Reconstruction and application of protein-protein interaction network. *International Journal of Molecular Sciences* 17, 907.
- Hao, T., Yu, A., Wang, B., Liu, A., Sun, J., 2015. Function annotation of proteins in *eriocheir sinensis* based on the protein-protein interaction network. In: *The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems*, pp. 831–837. Springer.
- Haykin, S., 1994. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR.
- Holding, A.N., 2015. XL-MS: Protein cross-linking coupled with mass spectrometry. *Methods* 89, 54–63.
- Holzer, P., 2017. Discovery of potent and selective p53-MDM2 protein-protein interaction inhibitors as anticancer drugs. *Chimia* 71, 716.
- Huo, T., Liu, W., Guo, Y., *et al.*, 2015. Prediction of host-pathogen protein interactions between *Mycobacterium tuberculosis* and *Homo sapiens* using sequence motifs. *BMC Bioinformatics* 16, 100.
- Hu, Y., Yang, Y., Fang, Z., *et al.*, 2017. Detecting pathway relationship in the context of human protein-protein interaction network and its application to Parkinson's disease. *Methods*.
- Ito, T., Chiba, T., Ozawa, R., *et al.*, 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences* 98, 4569–4574.
- Itri, F., Monti, D.M., Chino, M., *et al.*, 2017. Identification of novel direct protein-protein interactions by irradiating living cells with femtosecond UV laser pulses. *Biochemical and Biophysical Research Communications*.
- Itri, F., Monti, D.M., Della Ventura, B., *et al.*, 2016. Femtosecond UV-laser pulses to unveil protein-protein interactions in living cells. *Cellular and Molecular Life Sciences* 73, 637–648.
- Jensen, L.J., Gupta, R., Blom, N., *et al.*, 2002. Prediction of human protein function from post-translational modifications and localization features. *Journal of Molecular Biology* 319, 1257–1265.
- Jeong, H., Mason, S.P., Barabási, A.-L., Oltvai, Z.N., 2001. Lethality and centrality in protein networks. *arXiv preprint cond-mat/0105306*.
- Jiménez-García, B., Roel-Touris, J., Romero-Durana, M., *et al.*, 2017. LightDock: A new multi-scale approach to protein-protein docking. *Bioinformatics*.
- Jin, X., Lee, K., Kim, N.H., *et al.*, 2017. Natural products used as a chemical library for protein-protein interaction targeted drug discovery. *Journal of Molecular Graphics and Modelling*.
- Kaplan, A., Fournier, A.E., 2017. Targeting 14-3-3 adaptor protein-protein interactions to stimulate central nervous system repair. *Neural Regeneration Research* 12, 1040.

- Keskin, O., Tuncbag, N., Gursoy, A., 2016. Predicting protein-protein interactions from the molecular to the proteome level. *Chemical Reviews* 116, 4884–4909.
- Koike, A., Takagi, T., 2004. Prediction of protein-protein interaction sites using support vector machines. *Protein Engineering Design and Selection* 17, 165–173.
- Kozakov, D., Hall, D.R., Xia, B., *et al.*, 2017. The ClusPro web server for protein-protein docking. *Nature Protocols* 12, 255–278.
- Lee, J.H., Andrabi, R., Su, C.-Y., *et al.*, 2017. A broadly neutralizing antibody targets the dynamic HIV envelope trimer apex via a long, rigidified, and anionic β -hairpin structure. *Immunity* 46, 690–702.
- Lin, J.-S., Lai, E.-M., 2017. Protein-protein interactions: Co-immunoprecipitation. In: *Bacterial Protein Secretion Systems*. Springer.
- Marcotte, E.M., Pellegrini, M., Ng, H.-L., *et al.*, 1999. Detecting protein function and protein-protein interactions from genome sequences. *Science* 285, 751–753.
- Martínez-Gil, L., Vera-Velasco, N.M., Mingarro, I., 2017. Exploring the Human-Nipah virus protein-protein interactions. *Journal of Virology*, JVI. doi:10.1128/JVI.01461-17.
- Milroy, L.-G., Grossmann, T.N., Hennig, S., Brunsvel, L., Ottmann, C., 2014. Modulators of protein-protein interactions. *Chemical Reviews* 114, 4695–4748.
- Mohamed, T.P., Carbonell, J.G., Ganapathiraju, M.K., 2010. Active learning for human protein-protein interaction prediction. *BMC Bioinformatics* 11, S57.
- Moraes, C., 2013. *Série em biologia celular e molecular: Métodos experimentais no estudo de proteínas*. Rio de Janeiro: Fundação Oswaldo Cruz-Fiocruz.
- Newman, M.E., 2003. Mixing patterns in networks. *Physical Review E* 67, 026126.
- Nooren, I.M., Thornton, J.M., 2003. Diversity of protein-protein interactions. *The EMBO Journal* 22, 3486–3492.
- Ofran, Y., Rost, B., 2003. Predicted protein-protein interaction sites from local sequence information. *FEBS Letters* 544, 236–239.
- Ohue, M., Shimoda, T., Suzuki, S., *et al.*, 2014. MEGADOCK 4.0: An ultra-high-performance protein-protein docking software for heterogeneous supercomputers. *Bioinformatics* 30, 3281–3283.
- Orchard, S., Kerrien, S., Abbani, S., *et al.*, 2012. Protein interaction data curation: The International Molecular Exchange (IMEx) consortium. *Nature Methods* 9, 345–350.
- Orchard, S., Salwinski, L., Kerrien, S., *et al.*, 2007. The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nature Biotechnology* 25, 894–898.
- Pavlopoulos, G.A., Secrier, M., Moschopoulos, C.N., *et al.*, 2011. Using graph theory to analyze biological networks. *BioData Mining* 4, 10.
- Pellegrini, M., Haynor, D., Johnson, J.M., 2004. Protein interaction networks. *Expert Review of Proteomics* 1, 239–249.
- Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D., Yeates, T.O., 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proceedings of the National Academy of Sciences* 96, 4285–4288.
- Petrakis, S., Andrade-Navarro, M.A., 2016. Protein interaction networks in health and disease. *Frontiers in Genetics* 7.
- Phizicky, E.M., Fields, S., 1995. Protein-protein interactions: Methods for detection and analysis. *Microbiological Reviews* 59, 94–123.
- Pierce, B.G., Wiehe, K., Hwang, H., *et al.*, 2014. ZDOCK server: Interactive docking prediction of protein-protein complexes and symmetric multimers. *Bioinformatics* 30, 1771–1773.
- Puig, O., Caspary, F., Rigaut, G., *et al.*, 2001. The tandem affinity purification (TAP) method: A general procedure of protein complex purification. *Methods* 24, 218–229.
- Qi, Y., Bar-Joseph, Z., Klein-Seetharaman, J., 2006. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins: Structure, Function, and Bioinformatics* 63, 490–500.
- Rao, V.S., Srinivas, K., Sujini, G., Kumar, G., 2014. Protein-protein interaction detection: methods and analysis. *International Journal of Proteomics* 2014.
- Rezende, A.M., Folador, E.L., Resende, D.D.M., Ruiz, J.C., 2012. Computational prediction of protein-protein interactions in Leishmania predicted proteomes. *PLOS ONE* 7, e51304.
- Rinner, O., Mueller, L.N., Hubálek, M., *et al.*, 2007. An integrated mass spectrometric and computational framework for the analysis of protein interaction networks. *Nature Biotechnology* 25, 345–352.
- Ritchie, D.W., 2003. Evaluation of protein docking predictions using Hex 3.1 in CAPRI rounds 1 and 2. *Proteins: Structure, Function, and Bioinformatics* 52, 98–106.
- Rognan, D., 2015. Rational design of protein-protein interaction inhibitors. *MedChemComm* 6, 51–60.
- Santos, R., Ursu, O., Gaulton, A., *et al.*, 2017. A comprehensive map of molecular drug targets. *Nature Reviews Drug Discovery* 16, 19–34.
- Shin, W.-H., Christoffer, C.W., Kihara, D., 2017. In silico structure-based approaches to discover protein-protein interaction-targeting drugs. *Methods*.
- Szklarczyk, D., Franceschini, A., Kuhn, M., *et al.*, 2011. The STRING database in 2011: Functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research* 39, D561–D568.
- Thakur, M., Sharma, K., Chao, K., *et al.*, 2017. A protein-protein interaction dictates Borrelia infectivity. *Scientific Reports* 7.
- Torchala, M., Moal, I.H., Chaleil, R.A., Fernandez-Recio, J., Bates, P.A., 2013. SwarmDock: A server for flexible protein-protein docking. *Bioinformatics* 29, 807–809.
- Valencia, A., Pazos, F., 2002. Computational methods for the prediction of protein interactions. *Current Opinion in Structural Biology* 12, 368–373.
- Venkatesan, K., Rual, J.-F., Vazquez, A., *et al.*, 2009. An empirical framework for binary interactome mapping. *Nature Methods* 6, 83–90.
- Wang, S., Ding, M., Chen, X., Chang, L., Sun, Y., 2017. Development of bimolecular fluorescence complementation using rsEGFP2 for detection and super-resolution imaging of protein-protein interactions in live cells. *Biomedical Optics Express* 8, 3119–3131.
- Winzler, E.A., Shoemaker, D.D., Astromoff, A., *et al.*, 1999. Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. *Science* 285, 901–906.
- Wojcik, J., Schächter, V., 2001. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics* 17, S296–S305.
- Yook, S.H., Oltvai, Z.N., Barabási, A.L., 2004. Functional and topological characterization of protein interaction networks. *Proteomics* 4, 928–942.
- Zahiri, J., Hannon Bozorgmehr, J., Masoudi-Nejad, A., 2013. Computational prediction of protein-protein interaction networks: Algorithms and resources. *Current Genomics* 14, 397–414.
- Zhang, Y., Ku, W.L., Liu, S., *et al.*, 2017. Genome-wide identification of histone H2A and histone variant H2A. Z-interacting proteins by bPPI-seq. *Cell Research*.
- Zhou, H., Jakobsson, E., 2013. Predicting protein-protein interaction by the mirrortree method: Possibilities and limitations. *PLOS ONE* 8, e81100.

Relevant Website

<http://www.psidedv.info/mimix>
MIMix.

Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope

Anna Laddach, Sun Sook Chung, and Franca Fraternali, King's College, London, UK

© 2019 Elsevier Inc. All rights reserved.

Glossary

Asymmetric unit The smallest protein unit that can, by duplication and symmetry operations, generate a multimeric Protein Data Bank (PDB) structure.

Biological unit The functional protein unit of a PDB structure.

B-factor Also called temperature value, is associated with the spread of the measured electron density of an atom. It describes the uncertainty associated with the position of an atom in a crystallographic structure.

Interface core Amino acid residues which localise to the centre of a protein-protein interface and are fully buried upon interface formation.

Interface rim Amino acid residues which localise to the periphery of a protein-protein interface and are not fully buried upon complex formation.

Homologs Genes/proteins which share a common ancestor. A threshold sequence identity of 30% is normally used to detect homologs.

Obligate interaction A permanent interaction between proteins which cannot form stable structures on their own.

Orthologs Homologs which have arisen through speciation and perform similar functions in different species.

Paralogs Homologs which have arisen through a gene duplication event and have (generally) evolved to perform distinct functions.

Permanent interaction An interaction which is usually very stable and only exists in its complexed form.

Protein core The interior of a folded protein structure which is not exposed to solvent.

Protein interaction interface Regions of separate proteins which directly contact one another on complex formation.

Protein surface The surface of a protein structure which is exposed to solvent.

Solvent accessible surface area (SASA) The surface area, of a protein 3D component, which is exposed to solvent. Normally calculated at the residue level.

Transient interaction An interaction which can be formed briefly in the cell and in a reversible manner. Can be further classified as strong transient or weak transient based on measured interaction affinity.

Introduction

The genomic revolution of the past few decades has pushed technology to deliver efficient and precise tools for the fast and accurate mapping of genes in the genomes of thousands of species (Koepfli *et al.*, 2015). As a consequence, our knowledge of the human gene atlas is quite exhaustive (Telenti *et al.*, 2016). But knowledge without understanding has limited use. What one would aim for is to master the functional relationships between these genes, and their complex communication wired in a cellular context. The gene products, proteins, play an important role in this communication and in the networking capability of the cell; in this sense, they have been rightly defined as the workhorse of the cell. Therefore one of the next challenges is an exhaustive discovery of protein-protein interactions (PPIs) and the molecular aspects which play a role in their functional interplay. However, mapping the landscape of this multidimensional space of interactions is challenging, and techniques used to date vary in both complexity and efficacy. Recent renewed efforts, both in the theoretical and experimental fields, have been promoted by a number of studies which have challenged PPI detection techniques, such as Yeast Two-hybrid (Y2H) and Mass Spectrometry, to deliver much more accurate datasets for binary PPI maps (Fessenden, 2017). Data collected by the Human Proteome Project, since 2010, have recently (Feb 2016) achieved a coverage of 82% of this proteome, identifying altogether more than 90,000 unique protein interactions (Deutsch *et al.*, 2015). In spite of this progress, there is little overlap in the interactome uncovered by different experimental techniques (Fernandes *et al.*, 2010; Drew *et al.*, 2017). We are therefore obliged to look at these detailed snapshots as through a kaleidoscope and store, catalogue and assemble the unique pictures they display. From these, one can generate accurately annotated sub-sets of proteins and testable hypotheses, which can help in assigning confidence to functional modular units that play a role in the cellular machinery (Fessenden, 2017; Chung *et al.*, 2015).

Unfortunately, the large scale experimental probing of protein-protein interactions is both time-consuming and costly. However, computational methods can play a large role in filling the gap between gene knowledge and the functional communication. Moreover the results of such computational predictions can be validated by targeted experiments, thereby making more efficient use of experimental resources (Havugimana *et al.*, 2012; Carlin *et al.*, 2011).

Computational approaches to the study of PPIs include automated literature curation to generate large databases, such as STRING (Szklarczyk *et al.*, 2015) and the exploitation of sequence similarity/evolutionary relationships in order to both predict and extend functional knowledge of PPIs. Moreover further features, including the physicochemical properties of the interacting protomers, can be incorporated in methods to predict protein-protein interactions. Protein structural information, deposited in the PDB, has recently been exploited to add reliability to the collected interactions. Here, the structural coverage of the network has been extended through the large-scale homology modelling of binary interaction complexes (Mosca *et al.*, 2013). Recent work

using machine learning techniques and Bayesian approaches (Garzon *et al.*, 2016) has considerably progressed the field of protein-protein interaction prediction combining multiple scores, such as those based on protein abundance, expression profiles and partner redundancy and coarse grained structural modelling, along with the previously described features. Structural information on single protomers is also being used in docking procedures to predict and score binary complexes. These methods have considerably improved in the recent years (Xue *et al.*, 2015; Gromiha *et al.*, 2017) but are still dependent of some prior information on the binding features.

As the protein-protein interaction space covered by amalgamating data from 90 experimental investigations is sparse, it is becoming clear that effective data integration from multiple sources with different granularity, including computational analyses, can play a determinant role in expanding the known interactome and laying the foundation for improved predictions (Moal *et al.*, 2017). In this article we will give an overview of current methods for determination and prediction of PPIs, with a particular focus on predicting interfaces of interaction complexes and their structural properties.

Experimental Techniques and Data resources

Computational analyses of PPIs can extract useful information for the interpretation and prediction of properties that characterise PPIs. It is therefore important to be aware of the experimental source used for data determination, as this can impact on its underlying quality and accuracy. Moreover, an interaction can be considered more reliable if its existence is supported by multiple experimental techniques. This section briefly describes the most commonly used experimental methods for the detection of protein-protein interactions, highlighting the advantages and limitations of the different techniques, and recent advanced data resources, which have constructed protein-protein interaction networks from large-scale human proteome studies. These have stimulated the creation of integrative databases which make the results accessible to users.

Yeast Two-Hybrid (Y2H)

The yeast two-hybrid method (Two-hybrid screening) is the most widely used assay for detecting binary protein-protein interactions. It has been developed, refined and adapted to large-scale studies (Dufree *et al.*, 1993; Vidal *et al.*, 1996) since first introduced (Fields and Song, 1989). Two protein domains are tagged as a DNA binding domain (DBD) (bait protein) and an activation domain (AD) (prey protein) both of which are required for the transcription of a reporter gene and thus their interactions are inferred by expression of this reporter gene (Ito *et al.*, 2001). The Y2H method and its applications have been largely exploited to provide high-throughput protein-protein interaction (PPI) information, however some technical limitations persist. These are mainly caused by the lack of specificity between bait and prey, which may cause false-positive and false-negative interaction detection (Rao *et al.*, 2014; Westermarck *et al.*, 2013).

Affinity Purification Coupled with Mass Spectrometry (AP-MS)

Affinity Purification coupled with mass spectrometry (AP-MS) can identify protein complexes by directly purifying the protein of interest from cell lysates. This is followed by identification of the protein or proteins present by mass spectrometry analysis (Bantscheff *et al.*, 2007; Gingras *et al.*, 2007). Recent biotechnological advances (Konermann *et al.*, 2014; Yates, 2013) have overcome earlier problems such as specificity (segregation of over-expressed proteins or epitope-tagged proteins into abnormal cellular compartments) and co-purification of contaminating proteins (Rao *et al.*, 2014; Westermarck *et al.*, 2013).

Co-Fractionation Mass Spectrometry (CF-MS)

While the AP-MS method has been used to detect labelled bait proteins and the proteins with which they associate, it is limited to the identification of proteins for which antibodies are available or reliable epitope-tagged expression can be achieved. A Co-fractionation Mass Spectrometry (CF-MS) method has been developed to overcome such limitations, and establish comprehensive, less biased, protein associations of endogenous cellular proteins (Havugimana *et al.*, 2012; Kristensen *et al.*, 2012). The method involves preparing cellular lysates under mild, non-denaturing conditions. Soluble protein complexes in these lysates are then separated by several biochemical fractionation procedures, such as high-performance ion exchange chromatography, size-exclusion chromatography or isoelectro focussing. Fractions from each biochemical purification method are collected, trypsinized and the proteins present are identified by mass spectrometry analyses of each fraction. The components of each protein complex that co-fractionate are calculated by statistical and computational methods (Havugimana *et al.*, 2012).

3D Structures: X-ray, NMR, Cryo-EM

Accurate atomic detail of single and protein assemblies is regularly stored in the Protein Data Bank (PDB) (Berman *et al.*, 2000), the largest archive of structural data for biological macromolecules. The number of entries has increased rapidly due to progress in the techniques of X-ray crystallography, Nuclear Magnetic Resonance Spectrometry (NMR) and Cryo-electron

microscopy (cryo-EM) (Callaway, 2015) (60,332 homo-/hetero-mers among 121,046 protein structures, May 2017), however coverage of the human proteome and interactome is still incomplete (see Table 1).

PDB entries annotate possible assemblies of the characterised proteins. The asymmetric unit refers to the smallest protein unit that can, by duplication and symmetry operations, generate the multimeric structure. This result may actually not be related to the biological role of complex. On the other hand, the biological assembly is annotated as such if it has been shown, or predicted, to be the functional unit (Krissinel and Henrick, 2007). Thus, to understand the functional role of a structure, it is preferable to refer the biological assembly, whenever possible.

PDB entries annotate possible assemblies of the characterised proteins. The asymmetric unit refers to the smallest protein unit that can, by duplication and symmetry operations, generate the multimeric structure. This result may actually not be related to the biological role of complex. On the other hand, the biological assembly is annotated as such if it has been shown, or predicted, to be the functional unit (Krissinel and Henrick, 2007). Thus, to understand the functional role of a structure, it is preferable to refer the biological assembly, whenever possible.

Large-scale Human Proteome Studies

There have been an increasing number of high-throughput PPI studies published recently, which made use of the methods described in Sections Yeast Two-hybrid (Y2H), Affinity Purification Coupled with Mass Spectrometry (AP-MS), Co-fractionation Mass Spectrometry (CF-MS). For human protein networks, four accurate large-scale studies from different research communities, which applied different technologies such as Y2H (Rolland *et al.*, 2014), AP-MS (Hein *et al.*, 2015; Huttlin *et al.*, 2015) and CF-MS (Wan *et al.*, 2015), have been published. In spite of the fact that each dataset covers three to seven thousand proteins, with considerable overlap in protein content (30% to 68%), their overlap in protein interactions is still very limited (3% to 6%) (Drew *et al.*, 2017).

Public Databases

The need for reliable and public protein interaction data sources due to the progress in the technologies used to detect protein interaction information, has stimulated the development of a number of databases in recent years (Chatr-Aryamontri *et al.*, 2015; Orchard *et al.*, 2014; Szklarczyk *et al.*, 2015). These 180 rely on a collaborative effort throughout the International Molecular Exchange (IMEx) consortium (Orchard *et al.*, 2012) which develops solid data formats and defines curation rules to improve data integrity. Not only integrative databases, but also databases specifically tailored for structural annotations on protein-protein interaction networks, have been developed (Mosca *et al.*, 2013; Lu *et al.*, 2016) (Table 2).

Table 1 Structural coverage of the human protein-protein interaction network; statistics taken from Interactome3D

	Total	With structure	With model	Without structure
Proteins	14190	1843 (13.0%)	7600 (53.6%)	4747 (33.5%)
Interactions	95152	5757 (6.1%)	5978 (6.3%)	83417 (87.7%)

Source: Mosca, R., Ceol, A., Aloy, P., 2013. Interactome3D: Adding structural details to protein networks. Nat. Methods 10, 47–53. doi:10.1038/nmeth.2289.

Table 2 Examples of public PPI databases

Name	Key features	Public website
IntAct (Orchard <i>et al.</i> , 2014)	Protein interaction data repository curated from the literature and direct experimental results Updated monthly	http://www.ebi.ac.uk/intact/
BioGRID (Chatr-Aryamontri <i>et al.</i> , 2015)	Curated set of physical and genetic interactions including chemical associations and post-translational modifications Updated monthly	https://thebiogrid.org/
STRING (Szklarczyk <i>et al.</i> , 2015)	Known and predicted protein-protein interactions Main sources: High-throughput lab experiments, genomic context prediction, Co-expression, automated textmining, other knowledge databases	https://string-db.org
Interactome3D (Mosca <i>et al.</i> , 2013)	Structural annotation of protein-protein interaction networks from the available three-dimensional structures and modeling	http://interactome3d.irbbarcelona.org/
PinSnps (Lu <i>et al.</i> , 2016)	Genetic variant data mapped to a 3D integrated protein-protein interaction network	http://fraternalilab.kcl.ac.uk/PinSnps/
PrePPI (Garzon <i>et al.</i> , 2016)	Large scale database of interactions predicted by combining structural and non-structural methods	https://honiglab.c2b2.columbia.edu/PrePPI/

Features of Protein-Protein Interactions

The wealth of experimental data, described in Section Experimental Techniques and Data Resources, in addition to sequencing data, has been mined to uncover features which are characteristic of protein-protein interactions. These features are the key ingredients which have been exploited to predict both new interactions and interaction interfaces. Here we describe these features, followed by an account of how different approaches use such features in Section Computational Approaches to Predicting Protein-Protein Interactions.

Homology and Conservation

Since the late 80s, homology modelling has been used to infer the structure of proteins which lack experimentally resolved structures (Chothia and Lesk, 1986; Blundell *et al.*, 1988). This has played an important role in expanding the structural coverage of the proteome, as known protein sequences currently greatly outnumber experimentally solved structures (as described in Section 3D Structures: X-ray, NMR, Cryo-EM).

Importantly, this approach can be extended to expand the structural coverage of known or putative protein-protein interactions (see Fig. 1(a)); however, the coverage of known interactions by experimentally resolved binary complexes is even smaller (see Section 3D Structures: X-ray, NMR, Cryo-EM). The proviso must also be taken that, although interaction interfaces are generally conserved, a number of homologs have been observed to interact using different interfaces (Aloy *et al.*, 2003; Xue *et al.*, 2011) (see Fig. 1(c)). This is particularly the case when proteins are paralogs (proteins which have arisen through gene duplication which generally perform distinct functions (Pearson, 2013) rather than orthologs. The exact degree of interface conservation also appears to depend on a number of parameters associated with the nature of the interaction, e.g. binding sites of obligate interactions tend to be highly conserved and those of transient interactions less conserved (Xue *et al.*, 2011) (see Fig. 2 for a summary of interaction types). It has also been shown that homologs tend to interact using the same interface, even if the interaction partner is different (Esmailbeiki *et al.*, 2016) (see Fig. 1(b)). However, partner-specific interfaces have been shown to demonstrate greater conservation than generic interfaces, even with respect to those occurring in transient interactions (Xue *et al.*, 2011).

Homology-based approaches are rooted in the knowledge that, during evolution, constraints arising from protein structure and function impact on the conservation of residues (and therefore sequence identity). A classic example is that residues which reside in the core of a protein tend to be more conserved than surface residues. This is because such residues often play a crucial role in maintaining the fold of the protein and thus mutations which localise to this region are generally selected against (Echave *et al.*, 2016). Protein interactions represent another type of constraint, both structural and functional. A protein must maintain a structure which allows it to sterically and physicochemically interact with its partners; the affinity and kinetics of an interaction must also be suitably maintained so

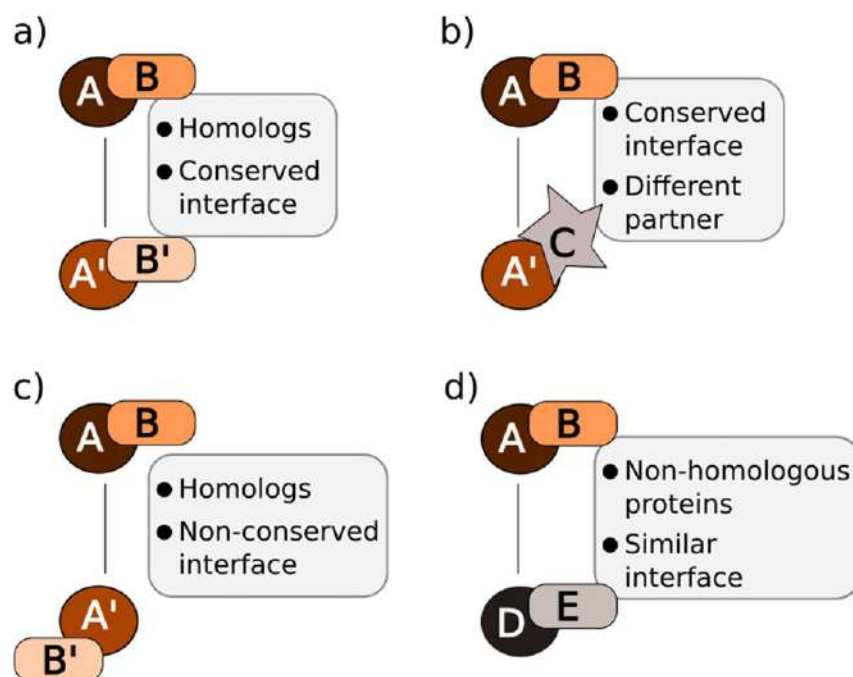


Fig. 1 Scenarios exploring the relationship between interface similarity and homology. Letters refer to protomers which form binary interaction complexes and the prime symbol is used to denote homology. (a) Homologs A' and B' use the same interface as A and B, (b) homolog A' uses the same interface to interact with a protein which is unrelated to B, (c) homologs A' and B' interact via a different interface, (d) non-homologous proteins D and E use similar interface due to similar structural properties).

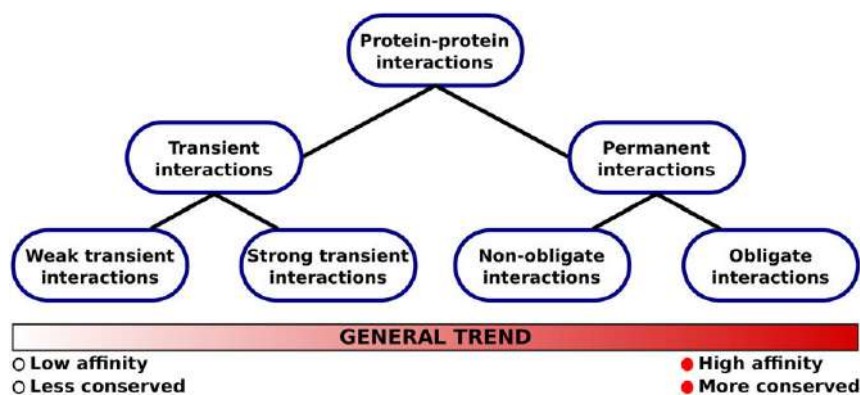


Fig. 2 Schematic showing interaction types. Obligate interactions refer to permanent complexes composed of protomers that are not stable on their own (see Glossary of terms).

that biological function is not disrupted. Indeed, it has been shown that surface residues which are involved in protein interaction interfaces are more highly conserved than non-interacting surface residues (Choi *et al.*, 2009; Ashkenazy *et al.*, 2016). An important observation, recently exploited in the prediction of protein interaction interfaces, is that when mutants do localise to an interface, this can result in the selection of compensatory mutants on opposing interface. This phenomenon, which is essentially due to co-evolution of the residues at the interface, allows functions abrogated by the initial mutation to be “rescued” by the compensatory mutation on the opposite side of the interface (de Juan *et al.*, 2013; Bitbol *et al.*, 2016).

Notably, it has also been shown that convergent evolution can take place. Here non-homologous proteins develop similar interaction interfaces (from a steric and physicochemical perspective) in order to interact with the same interaction partner (see Fig. 1(d)). An example of this is the mimicry of host protein interfaces by viral proteins in order to hijack host cell functions (Zhao *et al.*, 2011).

Protein Domains

In the vast majority of cases interactions do not involve entire proteins but rather specific regions, termed domains, which interact with one another (Raghavachari *et al.*, 2008). Therefore knowledge of a proteins underlying domain-architecture can aid in the prediction of both interactions and interaction interfaces, as discussed in Section Structure-based predictors.

Domains can be described as units within a protein which have a distinct structure or function, can fold independently, and can evolve with a degree of independence from one another. A number of resources define protein domains, including CATH (Sillitoe *et al.*, 2015), a structure-based classification system, and PFAM, which makes use of evolutionary information to infer domain boundaries (Finn *et al.*, 2014). Correct domain definitions are particularly crucial when dealing with large multi-domain proteins, such as titin (Laddach *et al.*, 2017).

Protein Recognition Mechanisms

Three different models exist to describe the nature of protein protein interactions (see Fig. 3). The lock and key model, originally proposed by Emil Fischer, describes interactions which are rigid in nature (Kastritis and Bonvin, 2013a; Fischer, 1894). Here both interaction interfaces are complementary in shape and negligible conformational changes take place on binding. In contrast the induced fit model involves conformational change on binding, allowing proteins with varying degrees of shape complementarity in the unbound state to interact (Kastritis and Bonvin, 2013a; Koshland, 1958). A subset of interactions which follow this model involve intrinsically disordered regions. Here an increase in order upon binding is associated with a decrease in entropy and results in weak, transient interactions (Perkins *et al.*, 2010). A third model, termed conformational selection, has been proposed. As proteins are dynamic in structure, each interaction partner will explore a number of conformational states, of which only particular states which will be able to interact (Kastritis and Bonvin, 2013a; Ma *et al.*, 1999; Bosshard, 2001; Boehr *et al.*, 2009).

Physicochemical Properties of Interaction Interface Regions

A number of features make important contributions to the properties of protein-protein interaction interfaces. These include the physicochemical properties of amino acids, such as charge and hydrophobicity, in addition to solvent accessibility, hydrogen bonds, secondary structural elements, steric properties (i.e., shape complementarity) and flexibility (Gromiha and Yugandhar, 2017; Keskin *et al.*, 2016). Protein interaction interfaces can be further divided into 280 core and rim regions (see Fig. 4). It has been shown that interface core regions, although surface exposed before binding, are enriched in hydrophobic residues and have amino acid compositions similar to the buried interior of a protein. In contrast, the rim regions differ less from other surface regions, and maintain a degree of solvent exposure upon binding. Hydrophilic and charged residues are usually enriched in such regions (Chakrabarti and Janin, 2002; Guharoy and Chakrabarti, 2005).

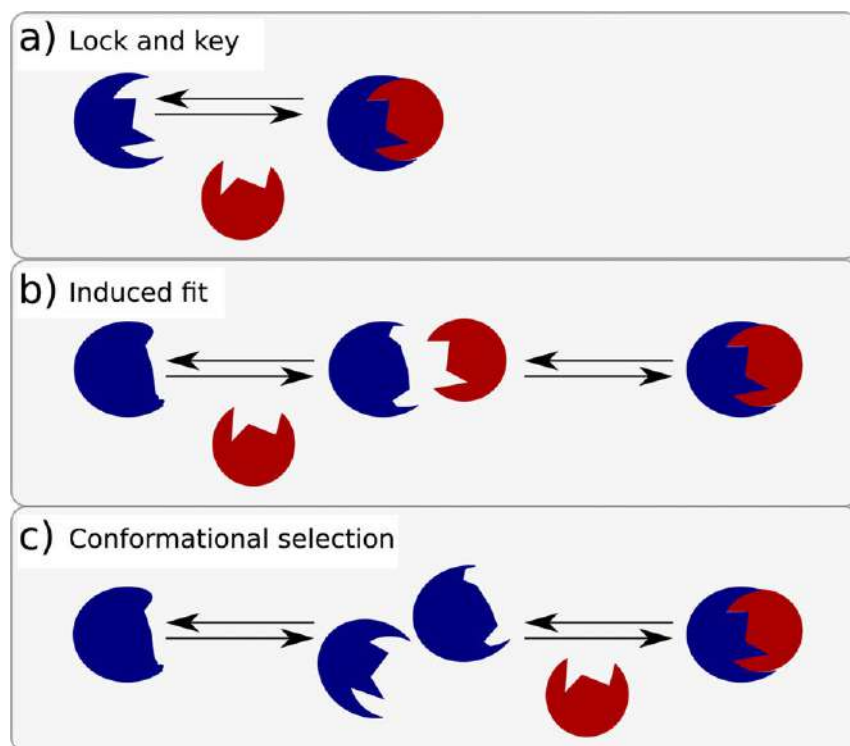


Fig. 3 Schematic showing protein mechanisms through which protein A (blue) recognises its binding partner, protein B (red): (a) lock and key, (b) induced fit, (c) conformational selection. Protein B can also undergo conformational changes, however here it is represented as a rigid body for simplicity. Adapted from Kastitis, P., Bonvin, A., 2013a. On the binding affinity of macromolecular interactions: Daring to ask why proteins interact. *J R Soc Interface* 10, 20120835. doi:10.1098/rsif.2012.0835.

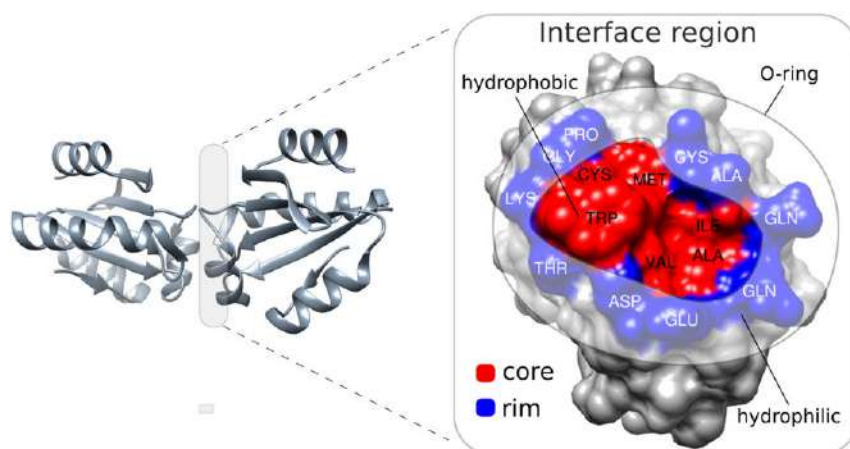


Fig. 4 Schematic showing homodimeric thioredoxin interface (PDB:3zxx) – core (red) and rim regions (blue) can clearly be distinguished. Note the core is composed of hydrophobic amino acids and the rim is composed of hydrophilic amino acids. The localisation of the interface is depicted on the left side in a ribbon representation of the homodimeric complex (Reproduced from Campos-Acevedo, A.A., Rudino-Pinera, E., 2014. Crystallographic studies evidencing the high energy tolerance to disrupting the interface disulfide bond of thioredoxin 1 from white leg shrimp *Litopenaeus vannamei*. *Molecules* 19, 21113-21126. doi:10.3390/molecules191221113).

Distinctions also exist between the properties of obligate and transient interaction interfaces. Hydrophobic interactions make a larger contribution to obligate interfaces whereas transient interfaces rely more on the formation of salt bridges (Keskin *et al.*, 2016). Phosphorylation sites also frequently play a role in transient interactions (Keskin *et al.*, 2016). Obligate interactions have been coined hairy by Kufareva *et al.* (2007), who found such interactions to rely on amino acid side chains. This contrasts with their findings on transient interactions, where protein backbones were determined to play a greater role. Although rare, disulphide bonds have been shown to play an important role in stabilising a number small complexes (Acuner Ozbabacan *et al.*, 2011).

It has been shown that only a small number of residues make large contributions to the binding affinity of protein-protein interactions (Bogan and Thorn, 1998; Moreira *et al.*, 2007). These are termed hotspot residues and tend to cluster within the core of interfaces (Keskin *et al.*, 2005); furthermore, they have been shown to be shielded from bulk solvent by the side chains of neighbouring residues (Li and Liu, 2009). Such shielding residues have been termed the O-ring due to the shape they form around core residues. Arginine, tryptophan and tyrosine have been found to be both most enriched in hotspots and to account for the greatest proportion of hotspot residues (Bogan and Thorn, 1998).

Finally, it is important to note that solvent water molecules can play an important role in protein interactions, in particular as they can fill cavities in the interaction complex, and be involved in bridging hydrogen bonds (Janin, 1999).

Computational Approaches to Predicting Protein-Protein Interactions

In this section we overview different approaches used in the computational prediction of protein-protein interactions. We then focus techniques which add a three-dimensional structural element to the interaction network, namely computational interface prediction and protein-protein docking.

Overview

The computational prediction of protein-interactions can be approached in numerous ways and with multiple levels of granularity. At the coarsest level a binary prediction of whether two proteins interact can be performed. One factor to consider in evaluating the feasibility of an interaction *in vivo*, is the cellular compartment in which the interacting protein operate and their likely co-occurrence. A task of much finer granularity is the prediction of the exact atomistic nature of protein-protein interactions; this enables protein interaction interfaces to be inferred.

A recently developed database which exemplifies the use of multiple methods to predict protein-protein interactions is PrePPI (Garzon *et al.*, 2016). Approaches used by PrePPI are summarised in Table 3. For a more general review see Keskin *et al.* (2016).

Given the wealth of information which can already be obtained from PrePPI and other databases (e.g STRING (Szklarczyk *et al.*, 2015)), the rest of this article will focus on methods which enable the prediction of interaction interfaces (i.e. exactly which residues contact one another in a complex). This information has several important applications, for example in the rational design of drugs which target protein-protein interactions (Jin *et al.*, 2014; Mosca *et al.*, 2015; Duran-Frigola *et al.*, 2017; Kategaya *et al.*, 2017), and in the interpretation of genetic variants and expression data. Here 3D structural data can be used to assess the impact of a genetic variant on the binding affinity of an interaction, and thus its edgetic impact on the network. Variants which localise to a protein core may disrupt a protein's structure, effectively removing a node from the network, whereas those which localise to interaction interfaces may affect specific interactions (edges in the network) (Sahni *et al.*, 2015; Engin *et al.*, 2016; Yi *et al.*, 2017). Therefore residue level information describing protein interaction interfaces is essential for the edgetic modelling of disease-associated variants (Mosca *et al.*, 2015).

Protein-Protein Interface Prediction

Interface predictors make use of known features of protein-protein interactions interfaces, as outlined in Section Affinity Purification Coupled with Mass Spectrometry (AP-MS), in order to predict which residues are involved in such interfaces. A distinction must be made between partner-specific and non-partner-specific predictors. Partner-specific predictors

Table 3 Methods used by PrePPI in the prediction of protein-protein interactions

Method	Description
Molecular Modelling	From the structures or models of a complex A:B, A and B are superimposed onto experimentally determined structures of A' in complex with B'. Interaction probability is then evaluated based on structural properties of the modelled interface and structural similarity of A and B with their homologs.
Phylogenetic profile	If orthologs of A and B are frequently present in the same species they are deemed more likely to interact.
Gene ontology	If proteins A and B have similar functions they are deemed more likely to interact.
Orthology	If orthologs of A and B are known to interact in other species they are deemed likely to interact.
Expression profile	If A and B demonstrate similar expression profiles they are deemed likely to interact. This can be extended to the expression profiles of orthologs.
Partner redundancy	If protein A interacts with a number n of proteins which are structurally similar to B, the likelihood of B interacting with A increases with n.
Protein peptide	If A contains a peptide (sequence motif) known to interact with a particular domain type present in B, A and B are considered likely to interact.

Source: Garzon, J.I., Deng, L., Murray, D., Shapira, S., Petrey, D., Honig, B., 2016. A computational interactome and functional annotation for the human proteome. *eLife* 5. doi:10.7554/eLife.18715.

(e.g., PS-HomPPI (Xue *et al.*, 2011), EV-complex (Hopf *et al.*, 2014)) take as input two query proteins, A and B, which are known to interact, and predict which residues constitute their interaction interface. Non-partner specific predictors (e.g. NPS-HomPPI (Xue *et al.*, 2011), SPPIDER (Porollo and Meller, 2007)) take a single query protein, A, and predict the residues that are likely to be involved in interactions with any interaction partner.

The following sections outline different categories of interface predictors and representative methods for each category are summarised in Table 4.

Sequence-based predictors

Features which can be derived from sequence alone, such as the physicochemical properties of amino acids, hydrophobicity and interface propensity can be used to predict whether residues are involved in protein-protein interactions. Properties are generally calculated using a sliding window or subsequence of a certain number of residues (Xue *et al.*, 2011; Rydevik *et al.*, 1991; Ofra and Rost, 2003). Some sequence-based predictors, such as PSIVER (Murakami and Mizuguchi, 2010) and ISIS (Ofra and Rost, 2007) also use predicted structural information, including predicted SASA and secondary structure; this has been shown to improve their performance (Esmailbeiki *et al.*, 2016). SeRenDIP, a recently developed random forest based predictor, goes one step further by incorporating predicted dynamics (predicted backbone flexibility) in its predictions (Hou *et al.*, 2017).

As interface residues are more conserved than non-interface surface residues (as detailed in Section Homology and Conservation), evolutionary information derived from multiple sequence alignments can be used in the prediction of interface residues. A more sophisticated use of evolutionary information to predict protein interaction interfaces is used by co-evolutionary methods; these make use of the theory, outlined in Section Homology and Conservation, that compensatory mutations on opposing interfaces should be selected for during evolution. A key challenge in these methods has been distinguishing between the direct coupling of residues (as they are in direct contact) and indirect coupling of residues (residue A and B covary as they are both in contact with residue C). A number of approaches, which rely on the creation of a global statistical model, have been used to address this challenge. These include direct coupling analysis, protein sparse inverse covariance and Bayesian network-based approaches (de Juan *et al.*, 2013).

Sequence-based methods are generally less accurate than structure-based methods, but are able to obtain a much greater coverage of the proteome due to the higher availability of sequence information (Gallet *et al.*, 2000). Coevolutionary methods have improved dramatically over recent years (Esmailbeiki *et al.*, 2016), but are limited to predicting interfaces where both partners are known, and a large number of homologous sequences are available. This is often a hindrance when protein partners are exclusive to a particular taxonomic branch (Hopf *et al.*, 2014).

Structure-based predictors

A number of predictors use structural features, such as solvent accessible surface area (SASA), secondary structure, geometry and B-factor, directly (Esmailbeiki *et al.*, 2016; Porollo and Meller, 2007). The performance of predictors which already use sequence-based features is greatly improved by this; particularly as, through the use of SASA, buried residues in the protein core can be immediately ruled out (Xue *et al.*, 2011). Other methods map features on to the protein structure and search for 3D patches of residues whose properties are consistent with being involved in protein-protein interaction interfaces (Porollo and Meller, 2007) (this can be seen as a 3D version of the sliding window approach outlined in Section Sequence-Based Predictors). Although not specific to protein interaction interfaces, ConSurf notably uses this method to facilitate the identification of functional sites (Ashkenazy *et al.*, 2010).

Table 4 Representative methods for the prediction of protein-protein interaction interfaces

Name	Key features	Model type	Public website
PSIVER (Murakami and Mizuguchi, 2010)	sequence-based predictor	Nai've Bayes	https://omictools.com/psiver-tool
SeRenDIP (Hou <i>et al.</i> , 2017)	sequence-based predictor	random forests	http://www.ibi.vu.nl/programs/serendipwww/
EVcomplex (Hopf <i>et al.</i> , 2014)	co-evolutionary method	maximum entropy	https://evcomplex.hms.harvard.edu/
SPPIDER (Porollo and Meller, 2007)	structural and sequence-based features used	neural network	http://sppider.cchmc.org/
ISPRED4 (Zhao and Gong, 2017)	structural and sequence-based features used	SVM with grammar based correction	https://ispred4.biocomp.unibo.it/ispred/
HomPPI (Xue <i>et al.</i> , 2011)	structural template-based method	regression	http://ailab1.ist.psu.edu/PSHOMPPV1.2/http://ailab1.ist.psu.edu/NPSHOMPPV1.2/
PriSE (Jordan <i>et al.</i> , 2012)	structural neighbourhood method	empirical scoring system	http://ailab1.ist.psu.edu/prise/index.py
PRISM (Tuncbag <i>et al.</i> , 2011)	structural alignment of interface regions	empirical scoring system	http://cosbi.ku.edu.tr/prism/

Partner specific structural predictors can take into consideration features of both structures, for example shape complementarity and electrostatics of the interface regions. Such predictions are thought to be more reliable if binding follows the lock and key model (see Section Physicochemical Properties of Interaction Interface Regions) as shape complementarity of unbound structures will be less complete if binding follows the induced fit model (Xue *et al.*, 2011). Furthermore, this strategy is most successful if available monomeric structures are similar in conformation to that in which the partners bind.

Template-based predictors of interfaces search for homologous proteins in structurally resolved complexes. Here the homologous complex may involve homologs of both proteins A and B, or it may only involve a homolog of one of the interaction partners. In the case that only one interacting partner is involved, the observation that proteins often use the same interface to interact with different partners, as outlined in Section Homology and Conservation, can be used to make a less confident prediction of the interaction interface. Where homologous complexes for partner specific interactions are available, these methods tend to give the best results. As outlined in Section Protein Domains, interactions may only occur between specific domains of two interacting proteins. Therefore, where no global templates are available to model an interaction, templates which are homologous at the domain level, if available, can be used; this is the approach which has been taken by Interac-tome3D (Mosca *et al.*, 2013), which uses the 3DID database (Mosca *et al.*, 2014) as a resource to identify interacting domain-domain templates. Some methods, such as HomPPI (Xue *et al.*, 2011), incorporate information from multiple structural templates, where available, in order to improve performance.

Structural neighbourhood methods also use template-based prediction methods. For these methods the protein of interest must have an experimental structure, despite the absence of a structure where it is present with its interaction partner. Here, rather than homologs, templates are searched for where the sub-units are structurally similar to the query protein. Methods such as PredUs (Zhang *et al.*, 2011) use global similarity of the query structure to the template whereas PrISE (Jordan *et al.*, 2012) also takes interface similarity specifically into account.

Combining features and methods

Information from different features can be combined to make predictions in multiple ways. Empirical scoring functions can be used; however, these functions can suffer if knowledge of the system (in this case the parameters which determine whether proteins interact) is incomplete. Such scoring functions have the disadvantage that their form must be predetermined and are generally linear in structure. Machine learning techniques have the advantage that they can make use of non-linear combinations of features and can, to a greater degree, infer their functional form from the data (Woźniakowski *et al.*, 2017). A number of different algorithms including Support Vector Machines (SVMs) (Sriwastava *et al.*, 2015; Zhao and Gong, 2017), neural networks (Porollo and Meller, 2007) and random forests (Hou *et al.*, 2017) can be used to classify residues as interacting or non-interacting. Researchers are beginning to explore the application of deep learning methods to the prediction of protein-protein interaction interfaces (Zhao and Gong, 2017), however these techniques require vast amounts of training data.

A number of methods, for predicting protein interaction interfaces, incorporate several distinct steps. For example, PRISM (Baspinar *et al.*, 2014), although primarily a template based method, guides structural alignments through the evolutionary prediction of hotspot residues. Succeeding structural alignment, further docking (see Section Docking for more information) and refinement steps are performed.

As more predictors become available, their results can be used as features to train meta-predictors, which generally achieve better performance than each individual predictor (Qin and Zhou, 2007; de Vries and Bonvin, 2011). Here, care must be taken that testing data used to evaluate the meta predictor does not overlap with the training data of its component predictors.

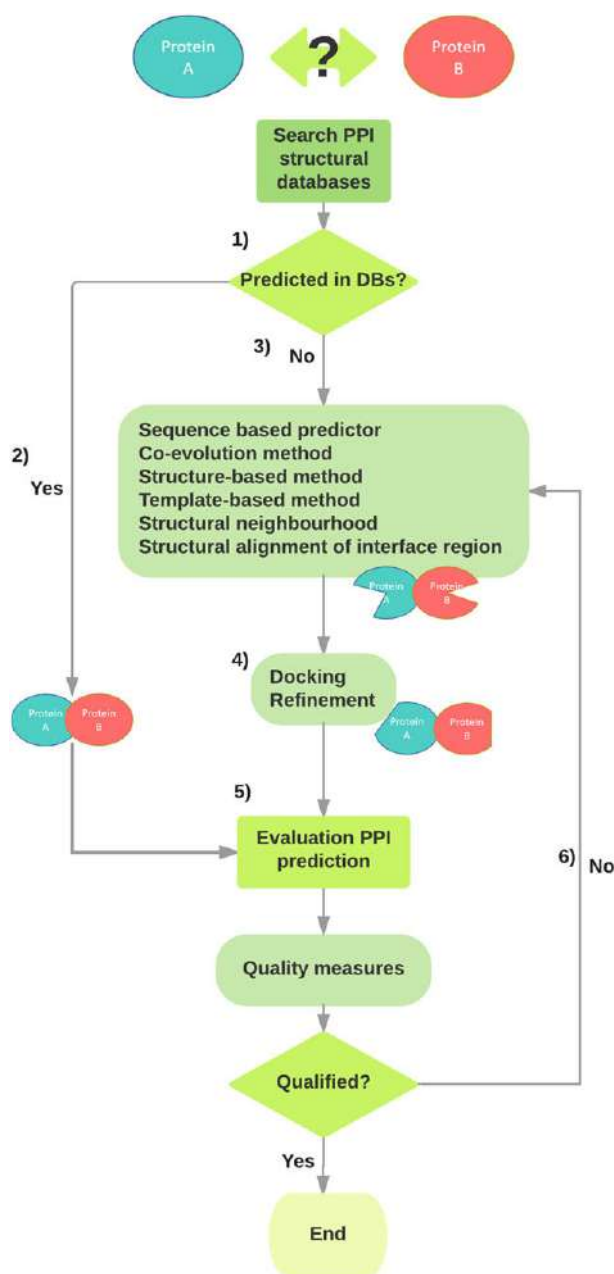
Docking

Given two protomers or proteins with structures, docking is a computational method to predict 3D structural protein-protein interaction complexes of target proteins (Rodrigues and Bonvin, 2014). It has evolved rapidly in parallel to the increase of publicly available molecular structures (Soni and Madhusudhan, 2017) and advances in computational technology (Schlick *et al.*, 2011). The Critical Assessment of Predicted Interactions (CAPRI) (Janin, 2005) is dedicated to the evaluation of protein-protein docking and organised by the research community to target unpublished crystal or NMR structural complexes. It has encouraged participations from different institutes worldwide and has become an important event in the field of protein structure prediction. Several representative computational tools and participants are listed in Table 5.

The key elements of docking are 3D structures (experimental or modelled) of target proteins, a sampling procedure to generate the conformational landscape of the predicted structural interactions, and a scoring function to compare and assess the predicted models (Rodrigues and Bonvin, 2014). Mainly, three types of scoring functions are used to evaluate binding affinity of protein interaction models: force-field scoring (calculating van der Waals (VDW) energy, electrostatic energy and bond stretching/bending/torsional forces); empirical scoring (integrating different weighted energy scores selectively to fit empirical binding data; and knowledge-based scoring (calculating statistical potential of the protein complex based on experimentally solved atomic structures) (Huang *et al.*, 2010). However, sampling and scoring are still challenging because of the incomplete quantitative information available regarding the impact of conformational change and molecular flexibility on binding affinity, as well as difficulties associated with certain protein types (e.g. lipid or membrane binding proteins) and multi-component assemblies (Kastritis and

Table 5 Examples of docking methods

Name	Key features	Public website
ATTRACT (deVries <i>et al.</i> , 2015)	Energy minimization in rotational and translational degrees of freedom	http://attract.ph.tum.de/services/ATTRACT/attract.html
ClusPro (Kozakov <i>et al.</i> , 2017)	Rigid-body docking (FFT), cluster retained conformations and refine by CHARMM minimization	https://cluspro.bu.edu
HADDOCK (Dominguez <i>et al.</i> , 2003)	Rigid body energy minimisation, semi-flexible refinement in torsion angle space, final refinement in explicit solvent refinement	http://milou.science.uu.nl/services/HADDOCK2.2/
PATCHDOCK (Schneidman-Duhovny <i>et al.</i> , 2005)	Molecular Shape representation, surface patch matching, filtering and scoring	https://bioinfo3d.cs.tau.ac.il/PatchDock/
RosettaDock (Lyskov and Gray, 2008)	Rigid-body position by random perturbation and selection based on Gaussian distribution, energy minimisation orientation, side-chain conformation optimisation	http://rosie.graylab.jhu.edu/


Fig. 5 A flowchart showing steps towards three dimensional protein-protein interaction prediction.

Bonvin, 2013b).

A Step by Step Guide to the Prediction of Three-Dimensional Protein-Protein Interactions

Here we outline how methods and resources, introduced throughout this article, can be applied to a query, consisting of two proteins (A and B), in order to obtain a 3D binary interaction complex model. Proposed steps are outlined below and illustrated in Fig. 5.

- 1) Given proteins A and B, for which we want to predict interactions, search PPI databases with structural annotations (e.g., Interactome3D (Mosca *et al.*, 2013), PrePPI (Garzon *et al.*, 2016)) and identify whether there are available models.
- 2) If there are predicted models of the complex, we can evaluate them as described in point 5) below.
- 3) If no models are available, different approaches can be applied. Models can be built using information derived from sequence based predictors and/or structure-based predictors. See Section Protein-Protein Interface Prediction and Table 4.
- 4) Apply docking or other refinement methods in order to improve predictions. See Sections Combining Features and Methods and Docking.
- 5) Evaluate models based on some quality measurement of prediction methods or scoring functions described in Docking. In addition, general protein structure assessment tools can be applied such as MolProbity (Chen *et al.*, 2010) or QMEAN (Benkert *et al.*, 2008).
- 6) Re-apply prediction methods if the qualities are not satisfied. Make use of other listed approaches, these can be applied and adapted to improve PPI model predictions.

Future Directions

This article has focussed on the prediction of binary interaction complexes, and more specifically on the prediction of protein-protein interfaces. However most biological complexes that play a crucial functional role in the cell, are made up of multiple protein components. Template based methods can be used to model such assemblies, but these are limited by the lack of available

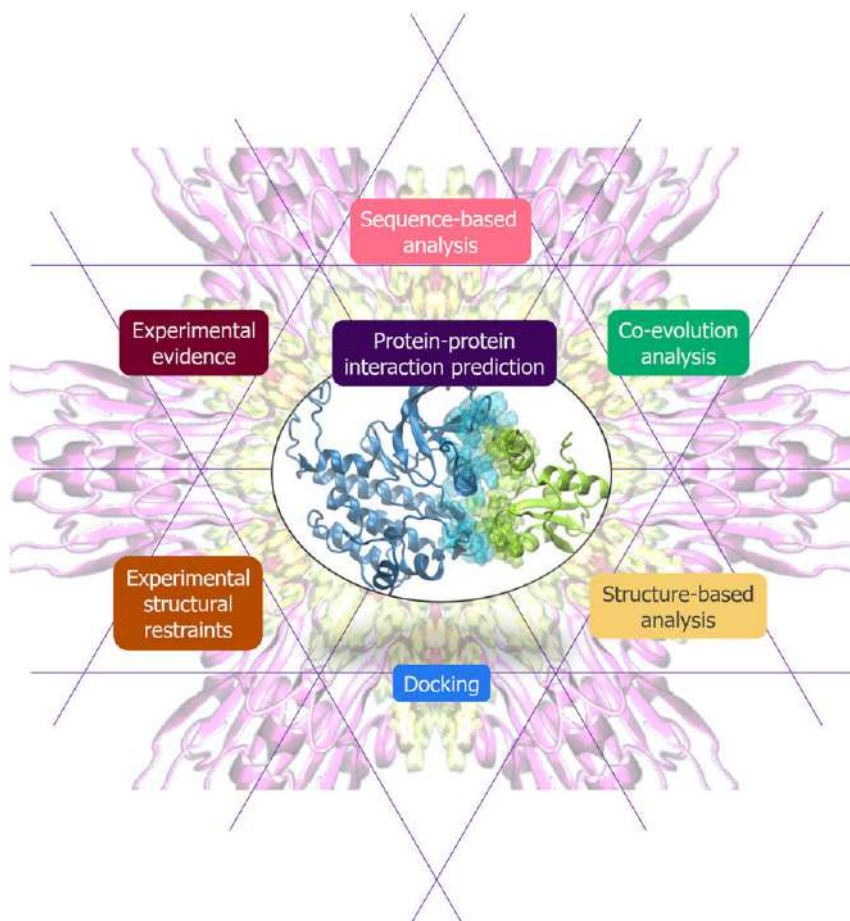


Fig. 6 A kaleidoscopic schematic for protein-protein interaction prediction.

experimentally resolved multicomponent structures. These are still particularly challenging to address by modern techniques. Notably, recent advances in the cryo-EM methodology allow for the structural determination of large complexes at near atomic resolution (Chlanda and Krijnse Locker, 2017; Orlov *et al.*, 2017; Lengyel *et al.*, 2014). Additionally, this technique can be complemented by data from crystallography and NMR experiments, as smaller complexes or single component structures can be projected onto EM density maps and help in the accurate reconstruction of a complex. In light of this, the PDB has launched a prototype framework for depositing hybrid models which result from a combination of experimental techniques (Burley *et al.*, 2017). This will allow for efficient use of integrated data in prediction methods.

A challenge will be in the extension and development of predictors to make efficient use of available multimeric information and the associated level of resolution. The Swiss-Model developers have started to address this quest with the release of a new pipeline, which enables the automated modelling of oligomeric complexes (Bertoni *et al.*, 2017). In parallel, a handful of docking programs are starting to address the automated docking of multiple components, although the procedures still rely mainly the pairwise docking of all the complex components (Soni and Madhusudhan, 2017).

As more experimental data are released, it becomes imperative to incorporate these into both computational predictors and docking methods. A number of computational procedures are already being developed to accomplish the efficient merging of restraints from data derived from different sources (this data may be varied in both nature and accuracy) (Politis and Schmidt, 2017; Schmidt *et al.*, 2017; Tamo *et al.*, 2017). Low resolution data, for example from small angle x-ray scattering (SAXS), EM, Ion Mobility Mass Spectrometry, and even data from site-directed mutagenesis experiments, can provide a conditional universe in which predictions can take place. One example which uses this approach is the docking software HADDOCK (Rodrigues and Bonvin, 2014), which already allows the integration of data from multiple sources, in the form of restraints.

Throughout this article we have seen how kaleidoscopic snapshots of the protein interaction universe (Fig. 6) can be expanded and integrated through the use of computational methods. However, only through the continued development and integration of both experimental and computational methods, it is conceivable that we may one day achieve a complete and accurate multi-dimensional map of the protein interaction landscape.

Acknowledgements

The authors thank Joseph Chi-fung Ng for his critical reading of the manuscript. This research was supported by the British Heart Foundation (to FF and AL), Bloodwise (to FF and SSC) and the Medical Research Council (MR/L01257X/1 to FF). AL is funded by the British Heart Foundation (RE/13/2/30182) and SSC is funded by a Bloodwise Gordon Piller PhD Studentship.

See also: Algorithms for Structure Comparison and Analysis: Docking. Alignment of Protein-Protein Interaction Networks. Artificial Intelligence and Machine Learning in Bioinformatics. Data Mining: Classification and Prediction. Machine Learning in Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Protein-Peptide Interactions in Regulatory Events. Protein-Protein Interaction Databases. Protein-Protein Interaction Databases. Protein-Protein Interactions: An Overview. Supervised Learning: Classification. Visualization of Biomedical Networks

References

- Acuner Ozbabacan, S., Engin, H., Gursoy, A., Keskin, O., 2011. Transient protein-protein interactions. *Protein Eng. Des. Sel.* 24, 635–648. doi:10.1093/protein/gzr025.
- Aloy, P., Ceulemans, H., Stark, A., Russell, R., 2003. The relationship between sequence and interaction divergence in proteins. *J. Mol. Biol.* 332, 989–998.
- Ashkenazy, H., Abadi, S., Martz, E., *et al.*, 2016. ConSurf 2016: An improved methodology to estimate and visualize evolutionary conservation in macromolecules. *Nucleic Acids Res.* 44, W344–W350. doi:10.1093/nar/gkw408.
- Ashkenazy, H., Erez, E., Martz, E., Pupko, T., Ben-Tal, N., 2010. ConSurf 2010: Calculating evolutionary conservation in sequence and structure of proteins and nucleic acids. *Nucleic Acids Res.* 38, W529–W533. doi:10.1093/nar/gkq399.
- Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., Kuster, B., 2007. Quantitative mass spectrometry in proteomics: A critical review. *Anal. Bioanal. Chem.* 389, 1017–1031. doi:10.1007/s00216-007-1486-6. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/17668192>.
- Baspinar, A., Cukuroglu, E., Nussinov, R., Keskin, O., Gursoy, A., 2014. PRISM: A web server and repository for prediction of protein-protein interactions and modeling their 3D complexes. *Nucleic Acids Res.* 42, W285–W289. doi:10.1093/nar/gku397.
- Benkert, P., Tosatto, S.C., Schomburg, D., 2008. Qmean: A comprehensive scoring function for model quality assessment. *Proteins: Struct., Funct., Bioinform.* 71, 261–277.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Res.* 28, 235–242. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/10592235>.
- Bertoni, M., Kiefer, F., Biasini, M., Bordoli, L., Schwede, T., 2017. Modeling protein quaternary structure of homo- and hetero-oligomers beyond binary interactions by homology. *Sci. Rep.* 7, 10480. doi:10.1038/s41598-017-09654-8.
- Bitbol, A., Dwyer, R., Colwell, L., Wingreen, N., 2016. Inferring interaction partners from protein sequences. *Proc. Natl. Acad. Sci. USA* 113, 12180–12185. doi:10.1073/pnas.1606762113.
- Blundell, T., Carney, D., Gardner, S., *et al.*, 1988. 18th Sir Hans Krebs lecture. Knowledge-based protein modelling and design. *Eur. J. Biochem.* 172, 513–520.
- Boehr, D., Nussinov, R., Wright, P., 2009. The role of dynamic conformational ensembles in biomolecular recognition. *Nat. Chem. Biol.* 5, 789–796. doi:10.1038/nchembio.232.
- Bogan, A., Thorn, K., 1998. Anatomy of hot spots in protein interfaces. *J. Mol. Biol.* 280, 1–9. doi:10.1006/jmbi.1998.1843.
- Bosshard, H., 2001. Molecular recognition by induced fit: How fit is the concept? *News Physiol. Sci.* 16, 171–173.
- Burley, S., Kurisu, G., Markley, J., *et al.*, 2017. PDB-Dev: A prototype system for depositing integrative/hybrid structural models. *Structure* 25, 1317–1318. doi:10.1016/j.str.2017.08.001.

- Callaway, E., 2015. The revolution will not be crystallized: A new method sweeps through structural biology. *Nature* 525, 172–174. doi:10.1038/525172a. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/26354465>.
- Carlini, L.M., Evans, R., Milewicz, H., *et al.*, 2011. A targeted siRNA screen identifies regulators of Cdc42 activity at the natural killer cell immunological synapse. *Sci. Signal.* 4, ra81. doi:10.1126/scisignal.2001729.
- Chakrabarti, P., Janin, J., 2002. Dissecting protein–protein recognition sites. *Proteins* 47, 334–343.
- Chatr-Aryamontri, A., Breitkreutz, B.J., Oughtred, R., *et al.*, 2015. The biogrid interaction database: 2015 Update. *Nucleic Acids Res.* 43, D470–D478. doi:10.1093/nar/gku1204. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/25428363>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4383984>, <http://nar.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=25428363>.
- Chen, V.B., Arendall, W.B., Headd, J.J., *et al.*, 2010. Mol-probity: All-atom structure validation for macromolecular crystallography. *Acta Crystallogr. Sect. D: Biol. Crystallogr.* 66, 12–21.
- Chlanda, P., Krijnse Locker, J., 2017. The sleeping beauty kissed awake: New methods in electron microscopy to study cellular membranes. *Biochem. J.* 474, 41–1053. doi:10.1042/BCJ20160990.
- Choi, Y., Yang, J., Choi, Y., Ryu, S., Kim, S., 2009. Evolutionary conservation in multiple faces of protein interaction. *Proteins* 77, 14–25. doi:10.1002/prot.22410.
- Chothia, C., Lesk, A., 1986. The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5, 823–826.
- Chung, S.S., Pandini, A., Annibale, A., *et al.*, 2015. Bridging topological and functional information in protein interaction networks by short loops profiling. *Sci. Rep.* 5, 8540. doi:10.1038/srep08540.
- Deutsch, E., Sun, Z., Campbell, D., *et al.*, 2015. State of the Human Proteome in 2014/2015 As Viewed through PeptideAtlas: Enhancing accuracy and coverage through the AtlasProphet. *J. Proteome Res.* 14, 3461–3473. doi:10.1021/acs.jproteome.5b00500.
- de Vries, S., Bonvin, A., 2011. CPORT: A consensus interface predictor and its performance in prediction-driven docking with HADDOCK. *PLOS One* 6, e17695. doi:10.1371/journal.pone.0017695.
- deVries, S., Schindler, C., ChauvotdeBeauchne, I., Zacharias, M., 2015. A web interface for easy flexible protein–protein docking with attract. *Biophys. J.* 108, 462–465. Available at: <http://www.sciencedirect.com/science/article/pii/S0006349514047602>. doi: <https://doi.org/10.1016/j.bpj.2014.12.015>.
- Dominguez, C., Boelsens, R., Bonvin, A.M., 2003. Haddock: A protein–protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* 125, 1731–1737. doi:10.1021/ja026939x. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/12580598>.
- Drew, K., Lee, C., Huizar, R.L., *et al.*, 2017. Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes. *Mol. Syst. Biol.* 13. doi:10.15252/msb.20167490. Available at: <http://msb.embopress.org/content/13/6/932.full.pdf>.
- Duran-Frigola, M., Siragusa, L., Ruppini, E., *et al.*, 2017. Detecting similar binding pockets to enable systems polypharmacology. *PLOS Comput. Biol.* 13, e1005522. doi:10.1371/journal.pcbi.1005522.
- Durfee, T., Becherer, K., Chen, P.L., *et al.*, 1993. The retinoblastoma protein associates with the protein phosphatase type 1 catalytic subunit. *Genes Dev.* 7, 555–569. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/8384581>.
- Echave, J., Spielman, S., Wilke, C., 2016. Causes of evolutionary rate variation among protein sites. *Nat. Rev. Genet.* 17, 109–121. doi:10.1038/nrg.2015.18.
- Engin, H., Kreisberg, J., Carter, H., 2016. Structure-based analysis reveals cancer missense mutations target protein interaction interfaces. *PLOS One* 11, e0152929. doi:10.1371/journal.pone.0152929.
- Esmailbeiki, R., Krawczyk, K., Knapp, B., Nebel, J., Deane, C., 2016. Progress and challenges in predicting protein interfaces. *Brief. Bioinform.* 17, 117–131. doi:10.1093/bib/bbv027.
- Fernandes, L., Annibale, A., Kleinjung, J., Coolen, A., Fraternali, F., 2010. Protein networks reveal detection bias and species consistency when analysed by information-theoretic methods. *PLOS One* 5, e12083. doi:10.1371/journal.pone.0012083.
- Fessenden, M., 2017. Protein maps chart the causes of disease. *Nature* 549, 293–295. doi:10.1038/549293a.
- Fields, S., Song, O., 1989. A novel genetic system to detect protein–protein interactions. *Nature* 340, 245–246. doi:10.1038/340245a0. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/2547163>.
- Finn, R., Bateman, A., Clements, J., *et al.*, 2014. Pfam: The protein families database. *Nucleic Acids Res.* 42, D222–D230. doi:10.1093/nar/gkt1223.
- Fischer, E., 1894. Einfluss der configuration auf die wirkung der enzyme. *Eur. J. Inorg. Chem.* 27, 2985–2993.
- Gallet, X., Charlotiaux, B., Thomas, A., Brasseur, R., 2000. A fast method to predict protein interaction sites from sequences. *J. Mol. Biol.* 302, 917–926. doi:10.1006/jmbi.2000.4092.
- Garzon, J.I., Deng, L., Murray, D., *et al.*, 2016. A computational interactome and functional annotation for the human proteome. *eLife* 5. doi:10.7554/eLife.18715.
- Gingras, A.C., Gstaiger, M., Raught, B., Aebersold, R., 2007. Analysis of protein complexes using mass spectrometry. *Nat. Rev. Mol. Cell Biol.* 8, 645–654. doi:10.1038/nrm2208. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/17593931>.
- Gromiha, M.M., Yugandhar, K., 2017. Integrating computational methods and experimental data for understanding the recognition mechanism and binding affinity of protein–protein complexes. *Prog. Biophys. Mol. Biol.* 128, 33–38. doi:10.1016/j.pbiomolbio.2017.01.001.
- Gromiha, M., Yugandhar, K., Jemimah, S., 2017. Protein–protein interactions: Scoring schemes and binding affinity. *Curr. Opin. Struct. Biol.* 44, 31–38. doi:10.1016/j.sbi.2016.10.016.
- Guharoy, M., Chakrabarti, P., 2005. Conservation and relative importance of residues across protein–protein interfaces. *Proc. Natl. Acad. Sci. USA* 102, 15447–15452. doi:10.1073/pnas.0505425102.
- Havugimana, P.C., Hart, G.T., Nepusz, T., *et al.*, 2012. A census of human soluble protein complexes. *Cell* 150, 1068–1081. doi:10.1016/j.cell.2012.08.011.
- Hein, M.Y., Hubner, N.C., Poser, I., *et al.*, 2015. A human interactome in three quantitative dimensions organized by stoichiometries and abundances. *Cell* 163, 712–723. doi:10.1016/j.cell.2015.09.053. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/26496610>, [https://linkinghub.elsevier.com/retrieve/pii/S0092-8674\(15\)01270-2](https://linkinghub.elsevier.com/retrieve/pii/S0092-8674(15)01270-2).
- Hopf, T.A., Scharfe, C.P., Rodrigues, J.P., *et al.*, 2014. Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife* 3. doi:10.7554/eLife.03430.
- Hou, Q., De Geest, P.F.G., Vranken, W.F., Heringa, J., Feenstra, K.A., 2017. Seeing the trees through the forest: Sequence-based homo- and heteromeric protein–protein interaction sites prediction using random forest. *Bioinformatics* 33, 1479–1487. doi:10.1093/bioinformatics/btx005.
- Huang, S.Y., Grinter, S.Z., Zou, X., 2010. Scoring functions and their evaluation methods for protein–ligand docking: Recent advances and future directions. *Phys. Chem. Chem. Phys.* 12, 12899–12908.
- Huttlin, E.L., Ting, L., Bruckner, R.J., *et al.*, 2015. The bioplex network: A systematic exploration of the human interactome. *Cell* 162, 425–440. doi:10.1016/j.cell.2015.06.043. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/26186194>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4617211>, [https://linkinghub.elsevier.com/retrieve/pii/S0092-8674\(15\)00768-0](https://linkinghub.elsevier.com/retrieve/pii/S0092-8674(15)00768-0).
- Ito, T., Chiba, T., Ozawa, R., *et al.*, 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA* 98, 4569–4574. doi:10.1073/pnas.061034498. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/11283351>.
- Janin, J., 1999. Wet and dry interfaces: The role of solvent in protein–protein and protein–DNA recognition. *Structure* 7, R277–R279.
- Janin, J., 2005. Assessing predictions of protein–protein interaction: The capri experiment. *Protein Sci.* 14, 278–283. doi:10.1110/ps.041081905. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/15659362>.
- Jin, L., Wang, W., Fang, G., 2014. Targeting protein–protein interaction by small molecules. *Annu. Rev. Pharmacol. Toxicol.* 54, 435–456. doi:10.1146/annurev-pharmtox-011613-140028.
- Jordan, R., El-Manzalawy, Y., Dobbs, D., Honavar, V., 2012. Predicting protein–protein interface residues using local surface structural similarity. *BMC Bioinform.* 13, 41. doi:10.1186/1471-2105-13-41.775de.
- Juan, D., Pazos, F., Valencia, A., 2013. Emerging methods in protein co-evolution. *Nat. Rev. Genet.* 14, 249–261. doi:10.1038/nrg3414.

- Kastritis, P., Bonvin, A., 2013a. On the binding affinity of macromolecular interactions: Daring to ask why proteins interact. *J. R. Soc. Interface* 10, 20120835. doi:10.1098/rsif.2012.0835.780.
- Kastritis, P.L., Bonvin, A.M., 2013b. Molecular origins of binding affinity: Seeking the archimedean point. *Curr. Opin. Struct. Biol.* 23, 868–877. doi:10.1016/j.sbi.2013.07.001. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/23876790>.
- Kategaya, L., Di Lello, P., Rouge, L., et al., 2017. USP7 small-molecule inhibitors interfere with ubiquitin binding. *Nature*. doi:10.1038/nature24006.
- Keskin, O., Ma, B., Nussinov, R., 2005. Hot regions in protein–protein interactions: The organization and contribution of structurally conserved hot spot residues. *J. Mol. Biol.* 345, 1281–1294. doi:10.1016/j.jmb.2004.10.077.
- Keskin, O., Tuncbag, N., Gursay, A., 2016. Predicting protein–protein interactions from the molecular to the proteome level. *Chem. Rev.* 116, 4884–4909. doi:10.1021/acs.chemrev.5b00683.
- Koepfli, K., Paten, B., O'Brien, S., 2015. The Genome 10K Project: A way forward. *Annu. Rev. Anim. Biosci.* 3, 57–111. doi:10.1146/annurev-animal-090414-014900.
- Konermann, L., Vahidi, S., Sowole, M.A., 2014. Mass spectrometry methods for studying structure and dynamics of biological macromolecules. *Anal. Chem.* 86, 213–232. doi:10.1021/ac4039306. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/24304427>.
- Koshland, D., 1958. Application of a theory of enzyme specificity to protein synthesis. *Proc. Natl. Acad. Sci. USA* 44, 98–104.
- Kozakov, D., Hall, D.R., Xia, B., et al., 2017. The cluspro web server for protein–protein docking. *Nat. Protoc.* 12, 255–278. doi:10.1038/nprot.2016.169. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/28079879>.
- Krissinel, E., Henrick, K., 2007. Inference of macromolecular assemblies from crystalline state. *J. Mol. Biol.* 372, 774–797. doi:10.1016/j.jmb.2007.05.022. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17681537>.
- Kristensen, A.R., Gsponer, J., Foster, L.J., 2012. A high-throughput approach for measuring temporal changes in the interactome. *Nat. Methods* 9, 907. doi:10.1038/nmeth.2131379. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/22863883>.
- Kufareva, I., Budagyan, L., Raush, E., Totrov, M., Abagyan, R., 2007. PIER: Protein interface recognition for structural proteomics. *Proteins* 67, 400–417. doi:10.1002/prot.21233.
- Laddach, A., Gautel, M., Fraternali, F., 2017. TITINdb—a computational tool to assess titins role as a disease gene. *Bioinformatics*. btx424. doi:10.1093/bioinformatics/btx424.
- Lengyel, J., Hnath, E., Storms, M., Wohlfarth, T., 2014. Towards an integrative structural biology approach: Combining Cryo-TEM, X-ray crystallography, and NMR. *J. Struct. Funct. Genomics* 15, 117–124. doi:10.1007/s10969-014-9179-9.
- Li, J., Liu, Q., 2009. 'Double water exclusion': A hypothesis refining the O-ring theory for the hot spots at protein interfaces. *Bioinformatics* 25, 743–750. doi:10.1093/bioinformatics/btp058.
- Lu, H.C., Herrera Braga, J., Fraternali, F., 2016. Pinsnp: Structural and functional analysis of snps in the context of protein interaction networks. *Bioinformatics* 32, 2534–2536.
- Lyskov, S., Gray, J.J., 2008. The rosettaDock server for local protein–protein docking. *Nucleic Acids Res.* 36, W233–W238. doi:10.1093/nar/gkn216. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/18442991>.
- Ma, B., Kumar, S., Tsai, C., Nussinov, R., 1999. Folding funnels and binding mechanisms. *Protein Eng.* 12, 713–720.
- Moal, I., Barradas-Bautista, D., Jimenez-Garcia, B., et al., 2017. IRAPPA: Information retrieval based integration of biophysical models for protein assembly selection. *Bioinformatics* 33, 1806–1813. doi:10.1093/bioinformatics/btx068.
- Moreira, I., Fernandes, P., Ramos, M., 2007. Hot spots – A review of the protein–protein interface determinant amino-acid residues. *Proteins* 68, 80312. doi:10.1002/prot.21396.
- Mosca, R., Ceol, A., Aloy, P., 2013. Interactome3D: Adding structural details to protein networks. *Nat. Methods* 10, 47–53. doi:10.1038/nmeth.2289.
- Mosca, R., Ceol, A., Stein, A., Olivella, R., Aloy, P., 2014. 3did: A catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Res.* 42, D374–D379. doi:10.1093/nar/gkt887.
- Mosca, R., Tenorio-Laranga, J., Olivella, R., et al., 2015. dSysMap: Exploring the edgetic role of disease mutations. *Nat. Methods* 12, 167–168. doi:10.1038/nmeth.3289.
- Murakami, Y., Mizuguchi, K., 2010. Applying the Naive Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. *Bioinformatics* 26, 1841–1848. doi:10.1093/bioinformatics/btq302.
- Ofran, Y., Rost, B., 2003. Predicted protein–protein interaction sites from local sequence information. *FEBS Lett.* 544, 236–239.
- Ofran, Y., Rost, B., 2007. ISIS: Interaction sites identified from sequence. *Bioinformatics* 23, e13–e16. doi:pmid17237081
- Orchard, S., Ammari, M., Aranda, B., et al., 2014. The mintact project – Intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* 42, D358–D363. doi:10.1093/nar/gkt1115. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/24234451>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3965093>, <http://nar.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=24234451>.
- Orchard, S., Kerrien, S., Abbani, S., et al., 2012. Protein interaction data curation: The international molecular exchange (imex) consortium. *Nat. Methods* 9, 345–350. doi:10.1038/nmeth.1931. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/22453911>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3703241>.
- Orlov, I., Myasnikov, A., Andronov, L., et al., 2017. The integrative role of cryo electron microscopy in molecular and cellular structural biology. *Biol. Cell* 109, 81–93. doi:10.1111/boc.201600042.
- Pearson, W.R., 2013. An introduction to sequence similarity (“homology”) searching. *Curr. Protoc. Bioinform.* 1. doi:10.1002/0471250953.bi0301s42. Chapter 3, Unit3.
- Perkins, J., Diboun, I., Dessailly, B., Lees, J., Orenco, C., 2010. Transient protein–protein interactions: Structural, functional, and network properties. *Structure* 18, 1233–1243. doi:10.1016/j.str.2010.08.007.
- Politis, A., Schmidt, C., 2017. Structural characterisation of medically relevant protein assemblies by integrating mass spectrometry with computational modelling. *J. Proteomics*. doi:10.1016/j.jprot.2017.04.019.
- Porollo, A., Meller, J., 2007. Prediction-based fingerprints of protein–protein interactions. *Proteins* 66, 630–645. doi:10.1002/prot.21248.
- Qin, S., Zhou, H., 2007. meta-PPISP: A meta web server for protein–protein interaction site prediction. *Bioinformatics* 23, 3386–3387. doi:10.1093/bioinformatics/btm434.
- Raghavachari, B., Tasneem, A., Przytycka, T., Jothi, R., 2008. DOMINE: A database of protein domain interactions. *Nucleic Acids Res.* 36, D656–D661. doi:10.1093/nar/gkm761.
- Rao, V.S., Srinivas, K., Sujini, G.N., Kumar, G.N., 2014. Protein–protein interaction detection: Methods and analysis. *Int. J. Proteomics* 2014, 147648. Available: <https://www.ncbi.nlm.nih.gov/pubmed/24693427>, doi:10.1155/2014/147648.
- Rodrigues, J., Bonvin, A., 2014. Integrative computational modeling of protein interactions. *FEBS J.* 281, 1988–2003. doi:10.1111/febs.12771.
- Rolland, T., Taan, M., Charlotiaux, B., et al., 2014. A proteome-scale map of the human interactome network. *Cell* 159, 1212–1226. doi:10.1016/j.cell.2014.10.050. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/25416956>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4266588>, [https://linkinghub.elsevier.com/retrieve/pii/S0092-8674\(14\)01422-6](https://linkinghub.elsevier.com/retrieve/pii/S0092-8674(14)01422-6).
- Rydevik, B., Pedowitz, R., Hargens, A., et al., 1991. Effects of acute, graded compression on spinal nerve root function and structure: An experimental Study of the Pig Cauda Equina. *Spine (Phila. PA 1976)* 16, 487–493.
- Sahni, N., Yi, S., Taipale, M., et al., 2015. Widespread macromolecular interaction perturbations in human genetic disorders. *Cell* 161, 647–660. doi:10.1016/j.cell.2015.04.013.
- Schlick, T., Collepardo-Guevara, R., Halvorsen, L.A., Jung, S., Xiao, X., 2011. Biomolecular modeling and simulation: A field coming of age. *Q. Rev. Biophys.* 44, 191–228. doi:10.1017/S0033583510000284. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/21226976>.
- Schmidt, C., Macpherson, J.A., Lau, A.M., et al., 2017. Surface accessibility and dynamics of macromolecular assemblies probed by covalent labeling mass spectrometry and integrative modeling. *Anal. Chem.* 89, 1459–1468. doi:10.1021/acs.analchem.6b02875.
- Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., Wolfson, H.J., 2005. Patchdock and symmdock: Servers for rigid and symmetric docking. *Nucleic Acids Research* 33, W363–W367. doi:10.1093/nar/gki481. Available at: <https://doi.org/10.1093/nar/gki481>.

- Sillitoe, I., Dawson, N., Thornton, J., Orengo, C., 2015. The history of the CATH structural classification of protein domains. *Biochimie* 119, 209–217. doi:10.1016/j.biochi.2015.08.004.
- Soni, N., Madhusudhan, M., 2017. Computational modeling of protein assemblies. *Curr. Opin. Struct. Biol.* 44, 179–189. doi:10.1016/j.sbi.2017.04.006.
- Sriwastava, B.K., Basu, S., Maulik, U., 2015. Protein–protein interaction site prediction in *Homo sapiens* and *E. coli* using an interaction-affinity based membership function in fuzzy SVM. *J. Biosci.* 40, 809–818. doi:10.1007/s12038-015-9564-y.
- Szklarczyk, D., Franceschini, A., Wyder, S., *et al.*, 2015. STRING v10: Protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* 43, D447–D452. doi:10.1093/nar/gku1003.
- Tamo, G., Maesani, A., Trager, S., *et al.*, 2017. Disentangling constraints using viability evolution principles in integrative modeling of macromolecular assemblies. *Sci. Rep.* 7, 235. doi:10.1038/s41598-017-00266-w.
- Telenti, A., Pierce, L., Biggs, W., *et al.*, 2016. Deep sequencing of 10,000 human genomes. *Proc. Natl. Acad. Sci. USA* 113, 11901–11906. doi:10.1073/pnas.1613365113.
- Tuncbag, N., Gursoy, A., Nussinov, R., Keskin, O., 2011. Predicting protein–protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. *Nat. Protoc.* 6, 1341–1354. doi:10.1038/nprot.2011.367.
- Vidal, M., Brachmann, R.K., Fattaey, A., Harlow, E., Boeke, J.D., 1996. Reverse two-hybrid and one-hybrid systems to detect dissociation of protein–protein and DNA–protein interactions. *Proc. Natl. Acad. Sci. USA* 93, 10315–10320. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/8816797>.
- Wan, C., Borgeson, B., Phanse, S., *et al.*, 2015. Panorama of ancient metazoan macromolecular complexes. *Nature* 525, 339–344. doi:10.1038/nature14877. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/26344197>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5036527>.
- Westermarck, J., Ivaska, J., Ivaska, G.L., 2013. Identification of protein interactions involved in cellular signaling. *Mol. Cell Proteomics* 12, 1752–1763. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/23481661>, doi:10.1074/mcp.R113.027771.
- Wo'jcikowski, M., Ballester, P., Siedlecki, P., 2017. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci. Rep.* 7, 46710. doi:10.1038/srep46710.
- Xue, L., Dobbs, D., Bonvin, A., Honavar, V., 2015. Computational prediction of protein interfaces: A review of data driven methods. *FEBS Lett.* 589, 3516–3526. doi:10.1016/j.febslet.2015.10.003.
- Xue, L., Dobbs, D., Honavar, V., 2011. HomPPI: A class of sequence homology based protein–protein interface prediction methods. *BMC Bioinform.* 12, 244. doi:10.1186/1471-2105-12-244.
- Yates, J.R., 2013. The revolution and evolution of shotgun proteomics for large-scale proteome analysis. *J. Am. Chem. Soc.* 135, 1629–1640. doi:10.1021/ja3094313. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/23294060>.
- Yi, S., Lin, S., Li, Y., *et al.*, 2017. Functional variomics and network perturbation: Connecting genotype to phenotype in cancer. *Nat. Rev. Genet.* 18, 395–410. doi:10.1038/nrg.2017.8.
- Zhang, Q., Deng, L., Fisher, M., *et al.*, 2011. Pre-dUs: A web server for predicting protein interfaces using structural neighbors. *Nucleic Acids Res.* 39, W283–W287. doi:10.1093/nar/gkr311.
- Zhao, N., Pang, B., Shyu, C., Korkin, D., 2011. Structural similarity and classification of protein interaction interfaces. *PLOS One* 6, e19554. doi:10.1371/journal.pone.0019554.
- Zhao, Z., Gong, X., 2017. Protein–protein interaction interface residue pair prediction based on deep learning architecture. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi:10.1109/TCBB.2017.2706682.

Protein–Protein Interaction Databases

Ashwini Patil, The University of Tokyo, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

The function of a protein in a cell depends on its interaction partners in the form of other proteins, nucleic acids and small molecules. These interactions change with time and cellular state allowing proteins to perform their functions. The knowledge of these interactions can be used to make interaction networks that can be analyzed to identify functional modules of proteins and their interconnections to ultimately understand cellular function (Cafarelli *et al.*, 2017). The identification of protein-protein interactions (PPIs) is thus an important means of understanding biological processes.

The detection of interactions between proteins began on a large scale in the early 2000s, prior to which PPIs were identified mainly as those between proteins of interest or those associated with a specific pathway or function. Starting in the year 2000, several studies reported a large number of protein-protein interactions identified using high-throughput screening methods (Uetz *et al.*, 2000; Ito *et al.*, 2001; Ho *et al.*, 2002; Giot *et al.*, 2003; Li *et al.*, 2004; Rual *et al.*, 2005; Stelzl *et al.*, 2005). Over the years, the number of interaction datasets has increased exponentially. This lead to the need for resources that would combine the large number of interactions, annotate them and make them interoperable. This need is fulfilled by PPI databases which organize, curate, integrate and annotate PPI data and make it available for downstream analysis.

All PPI databases collect protein-protein interactions either directly from literature, through direct deposition by authors, or from other databases. Databases use different criteria to filter, annotate and group the collected interactions and provide distinct user interfaces to allow users to access the PPI data. In some cases, methods to analyze this data are also provided. Some databases are limited to one or a few species. An important service provided by the databases is the assessment of the PPIs for reliability. It is known that PPIs identified in-vitro by existing methods have a large number of false-positives i.e. interactions that show incorrect binding or are not observed in-vivo (Von Mering *et al.*, 2002). This is now changing with increasing confirmation of the identified interactions using validation assays (Luck *et al.*, 2017). However, several databases now assign scores to interactions that help users identify those interactions that are more likely to exist in the cell. Due to limitations of interaction detection techniques, the number of interactions experimentally identified is much smaller than those existing in the cell. Therefore, a few databases also provide predicted PPIs, either as predicted interactions between proteins or functional associations identified by genetic interactions.

Thus, there is a wide variety of PPI databases available. As the number of PPIs detected increases and the databases become more specialized, it is becoming increasingly difficult to keep abreast of the various databases and their offerings. Therefore, it is important for users to understand these databases to help identify the ones that will be most appropriate in a specific study. This article provides an overview of the current major PPI databases. The article first describes the different types of interactions identified and stored in PPI databases. This is followed by a brief description of several databases, their special features and the methods they use for PPI quality assessment. Next, a description of the recent development of community-wide standards and protocols that make it easier to access database content for in-depth analyses is provided. Finally, a discussion of the strengths and weaknesses of the current PPI databases is presented and suggestions are made for their improvement.

Types of Interactions

PPIs can be broadly classified into two groups – direct or physical, and indirect or non-physical interactions. These are further divided into sub-groups (Table 1). These groups are frequently used to classify PPIs in databases.

Physical Interactions

Physical interactions are those in which the interactors physically bind to each other. These may be interactions between proteins or proteins and other molecules. Physical protein-protein interactions can further be divided into two categories depending on the type of method used to identify them. Binary interactions are direct binding events that are detected between two proteins. Several

Table 1 Classification of interactions in PPI databases

Class	Definition
Physical/direct	Direct binding event between two proteins
Non-physical/Indirect	Functional association – genetic or predicted interaction
Small-scale	Detected in small experiment with 100 or less interactions
High-throughput	Detected in proteome-scale experiment
Binary	Interaction between two proteins
Complex	Association of multiple proteins; direct interactions unknown

methods can be used to detect binary interactions, such as, yeast two hybrid (Y2H) systems (Ito *et al.*, 2001; Giot *et al.*, 2003; Li *et al.*, 2004; Rolland *et al.*, 2014), protein arrays (Yazaki *et al.*, 2016), fluorescence resonance energy transfer (FRET), and structure determination methods like X-ray crystallography and NMR Spectroscopy. PPIs can also be identified in the form of protein complexes, or associations, using methods like affinity purification followed by mass spectroscopy (AP-MS) (Hein *et al.*, 2015; Huttlin *et al.*, 2015, 2017) and co-fractionation followed by mass spectroscopy (CoFrac-MS) (Wan *et al.*, 2015). Of these methods, Y2H, AP-MS and CoFrac-MS are amenable to high-throughput experiments and have been used to identify PPIs on the proteomic scale. Methods identifying protein complexes do not provide information about direct binding events in the complex. Hence binary interactions are often predicted from protein complex data before they are stored in databases. Databases use one of two models to convert protein complex data into binary interactions – spoke model or matrix model (Bader and Hogue, 2002). The spoke model assumes that the bait protein, i.e. the protein that is tagged and used to elute other proteins that associate with it, directly binds to all the other associated proteins. The matrix model assumes that all proteins eluted together interact with each other. The spoke model is more conservative and results in fewer false positive interactions (Bader and Hogue, 2002) and hence, is the model of choice in several databases. However, neither model gives the true binding states, often resulting in the prediction of incorrect binary interactions.

An additional classification that is common for PPIs is that based on the size of the experiment in which they are detected. PPIs that are detected in small, focused studies, typically identifying less than 100 interactions, are often denoted as small-scale (Patil *et al.*, 2011). On the other hand, interactions identified in proteome-wide studies are deemed as high-throughput. The cutoff interaction count separating small-scale from high-throughput experiments varies with databases. It has long been argued that interactions identified in high-throughput experiments have larger number of false positives or non-physiological interactions (Von Mering *et al.*, 2002; Wodak *et al.*, 2013). This makes scoring interactions for reliability very important. It has also been observed that interactions from the same species identified by different methods are complementary and have very little overlap (Von Mering *et al.*, 2002; Wodak *et al.*, 2013). It has been proposed that this may be the result of the large number of potential interactions in the cell all of which cannot be identified by a single method, and the low sensitivity of the methods, rather than the high rate of false interactions identified by these methods (Wodak *et al.*, 2013; Luck *et al.*, 2017).

Non-Physical Interactions

Indirect or non-physical interactions are functional associations between proteins and/or other molecules that may or may not be direct binding events. Indirect associations of proteins may be identified as genetic interactions using synthetic lethal arrays (Costanzo *et al.*, 2010). They can also be predicted interactions or associations (Skrabaneck *et al.*, 2008). Sequence homology is often used to predict protein associations such that homologs of interacting proteins in the same or different species are also predicted to interact. This is a method that is often used to transfer interactions between closely related species. For example, interactions identified in mouse can be transferred to human through orthologous proteins (Brown and Jurisica, 2005). Structural methods can also be used to predict PPIs through binding surface prediction and docking of two proteins with known structures (Mosca *et al.*, 2013). Genomic context approaches are another commonly used method of PPI prediction. These include methods based on (1) identifying pairs of genes that are in close proximity or fused in a few species, (2) the coordinated presence or absence of genes in different genomes and, the presence of conserved regions and (3) correlated mutations in two protein/gene sequences indicating a possible role in binding (Skrabaneck *et al.*, 2008). Finally, co-occurring terms in scientific abstracts obtained through text-mining can also be used to predict functional associations between proteins (Papanikolaou *et al.*, 2015).

Protein–Protein Interaction Databases

Protein-protein interaction databases are divided into two categories: (1) primary databases, and (2) derived or consolidated databases.

Primary Databases

Primary databases collect and store interactions directly from literature or through author submissions (Table 2). They may contain data from small-scale focused studies or large-scale high-throughput experiments. Following are some of the major primary databases storing PPI data:

1. IntAct: A database containing molecular interactions curated from literature or obtained by direct submission from authors (Orchard *et al.*, 2014). Most of the interactions in IntAct are physical associations from high-throughput experiments, and the species with the highest representation is human. It is one of the few PPI databases with extensive annotations and proteins mapped to multiple database identifiers.
2. BioGRID: A primary database that collects interactions from literature, mainly from high-throughput experiments, and stores them after curation (Chatr-Aryamontri *et al.*, 2017). The interactions in BioGRID are classified into physical or genetic interactions. BioGRID also stores data about post-translational modifications in proteins. It contains the largest number of curated interactions in yeast. All interactions in BioGRID are in binary format i.e. complex associations have been converted to binary format using the spoke model in which the bait protein interacts with each prey protein.

Table 2 Primary protein-protein interaction databases

Database	Interactions	PPI type	Associations	Scoring	Focus	Reference
IntAct	Protein-molecule	Experimental	All	Yes	All	Orchard et al. (2014)
BioGRID	Protein-molecule	Experimental	All	No	All	Chatr-Aryamontri et al. (2017)
MINT	Protein-molecule	Experimental	All	Yes	All	Ceol et al. (2010)
DIP	Protein-protein	Experimental	All	No	All	Salwinski et al. (2004)
HPRD	Protein-protein	Experimental	Physical	No	Human	Prasad et al. (2009)
MatrixDB	Protein-protein	Experimental	All	No	All	Launay et al. (2015)
CORUM	Protein-protein	Experimental	Physical	No	Mammals	Ruepp et al. (2010)
MIPS	Protein-protein	Experimental	Physical	No	Mammals	Pagel et al. (2005)

Interactions: Protein-protein – interactions exclusively between proteins; Protein-molecule – interactions between proteins and proteins, DNA, RNA, drugs or other small molecules.

PPI type: Experimental – experimentally identified interactions; Predicted – predicted associations.

Associations: All – physical and indirect associations between interactors; Physical – only interactions involving physical binding.

Scoring: Presence of confidence score assessing interaction reliability.

Focus: Specific area of focus of the database eg. Species or biological process specific.

3. MINT: MINT collects and curates PPI data on a smaller scale than IntAct and BioGRID ([Ceol et al., 2010](#)). This data is now hosted by the IntAct database. However, MINT still contains some interactions that have not been transferred to IntAct.
4. DIP: DIP is a much smaller database compared to IntAct and BioGRID. It stores curated protein-protein interactions from literature ([Salwinski et al., 2004](#)).
5. HPRD: The Human Protein Reference Database (HPRD) is one of the most comprehensive resources of physical protein-protein interactions in humans ([Prasad et al., 2009](#)). The data is manually curated from literature, primarily from small-scale experiments. Interactions in HPRD are in binary format and come with isoform information. HPRD data available for download is not as detailed as that provided by some of the other primary databases.
6. MatrixDB: This is a database that curates and stores interactions of extracellular matrix proteins obtained from literature ([Launay et al., 2015](#)).
7. CORUM: A database containing manually annotated protein complexes from mammals. This database contains small-scale interactions collected and manually curated from literature ([Ruepp et al., 2010](#)). These interactions are taken only from small-scale focused experiments and are, therefore, considered to be more reliable than those present in larger databases which contain a greater number of interactions from high-throughput experiments. The data from CORUM is also frequently used as a gold standard, or known true, dataset in PPI quality evaluation.
8. MIPS: MIPS is a database of manually curated protein-protein interactions in mammals from literature ([Pagel et al., 2005](#)). MIPS is also used as a gold standard dataset for binary protein-protein interactions.

Primary databases perform a commendable task of collecting, curating and making the PPI data available. But several issues need to be addressed. At present, PPI data is distributed across multiple primary databases. As a result, getting a complete set of interactions for network or pathway analysis requires combining data from multiple sources. The adoption of standards by several primary databases has significantly eased this process. However, combining data from multiple databases and annotating them is still non-trivial for two reasons. Firstly, databases use different interactor identifiers requiring the mapping of identifiers from multiple databases onto a common identifier before merging. Secondly, the presence of interactions using obsolete or incorrect identifiers makes merging challenging. The depth of curation of data varies with each database. Thirdly, some databases are better curated and have more annotations than others. As a result, integrating information from databases requires time and resources, and derived or consolidated databases help resolve these issues.

Derived Databases

Derived databases take PPI data from primary interaction databases, integrate it, filter it and annotate it ([Table 3](#)). Many of the derived databases also provide an additional level of curation. The derived databases try to alleviate the problems users face during integration of multiple PPI datasets by mapping all the interactor identifiers from multiple databases to a common identifier followed by combining the interactions into a unique set. After integration, interactions may be selected by the databases for one or more species or a specific biological system. Along with integration and selection, several databases also provide confidence scores to assess the quality of the interactions. Some databases also provide predicted interactions and interactions with annotations of 3D structure. As a result, derived databases are often the ones from which users get their interaction data for further analysis. Following are some of the major derived PPI databases.

1. STRING: STRING is a database of functional associations ([Szklarczyk et al., 2015](#)). It is one of the largest derived databases. It collects physical and indirect PPIs from multiple primary databases. It also contains predicted functional associations between proteins. All interactions in STRING are scored using multiple evidences for their reliability. The data from STRING is now available for network analysis through a plugin in the software, Cytoscape ([Cline et al., 2007](#)).

Table 3 Derived protein-protein interaction databases. Column descriptions are same as those in **Table 2**

Database	Interactions	PPI type	Associations	Scoring	Focus	References
STRING	Molecular	Experimental + Predicted	All	Yes	All	Szklarczyk <i>et al.</i> (2015)
HitPredict	Proteins	Experimental	Physical	Yes	All	Lopez <i>et al.</i> (2015)
iRefWeb	Proteins	Experimental	All	Yes	All	Turner <i>et al.</i> (2010)
mentha	Proteins	Experimental	Physical	Yes	All	Calderone <i>et al.</i> (2013)
InnateDB	Molecular	Experimental + Predicted	All	No	Mammals (immunity)	Breuer <i>et al.</i> (2013)
IID	Proteins	Experimental + Predicted	All	No	6 organisms	Kotlyar <i>et al.</i> (2016)
HPIDB	Proteins	Experimental + Predicted	All	No	Human	Ammari <i>et al.</i> (2016)
ConsensusPathDB	Molecular	Experimental	All	No	3 organisms	Kamburov <i>et al.</i> (2013)
HAPPI	Proteins	Experimental + Predicted	All	Yes	Human	Chen <i>et al.</i> (2017)
HIPPIE	Proteins	Experimental	All	Yes	Human	Alanis-Lobato <i>et al.</i> (2017)
DroID	Molecular	Experimental + Predicted	All	Yes	<i>Drosophila</i>	Murali <i>et al.</i> (2011)
Interactome3D	Proteins	Experimental	Physical	No	All (3D structure)	Mosca <i>et al.</i> (2013)

- HitPredict: HitPredict is a derived database that includes physical protein-protein interactions from multiple primary databases (Lopez *et al.*, 2015). HitPredict is a deeply curated database using several automatic and manual checks to assess the quality of the PPI data from primary sources. Extensive checks are performed to confirm the validity of the interacting proteins and significant effort is put into assigning the correct identifier to proteins. Interactions of proteins that do not map to valid identifiers, are not in the same species, or that do not have valid experimental annotations are not included. HitPredict includes interactions from organisms that have at least 10 high-quality interactions. HitPredict also provides a reliability score for PPIs.
- iRefWeb: This is a derived database that integrates PPI data from multiple sources (Turner *et al.*, 2010). It includes physical as well as indirect interactions. iRefWeb provides detailed information about each interaction in the form of the number of primary databases it was obtained from and the annotation of the interaction in each source database. It also provides extensive mapping to multiple protein and gene identifiers including the various isoforms. iRefWeb provides a reliability score for each interaction.
- mentha: This is a derived database that automatically collects and integrates data published by multiple primary databases using a web service (Calderone *et al.*, 2013). mentha depends on the source databases to provide the curation and the correct interactor identifiers. mentha also provides a reliability score. It provides some network analysis tools to extract subnetworks of genes and identify the possible paths between two set of genes.
- InnateDB: InnateDB is a database of associations between mammalian genes and proteins that are specific to the innate immune response (Breuer *et al.*, 2013). Interactions are limited to those in human, cow and mouse. It includes a small set of curated interactions, a set of interactions validated from publications and a set of predicted interactions. Interactions are physical as well as indirect associations between proteins, DNA and RNA. InnateDB provides tools for pathway, gene ontology and network analyses. The interactions in InnateDB are not scored.
- IID: The Integrated Interaction Database combines experimentally identified interactions from primary databases and includes predicted PPIs in yeast, worm, fly, rat, mouse and human (Kotlyar *et al.*, 2016). It also provides tissue specific PPIs. IID does not provide reliability scoring for interactions.
- HPIDB: HPIDB provides interactions between proteins from pathogens and their host organisms (Ammari *et al.*, 2016). PPIs are collected from primary and some derived databases without reliability scoring.
- ConsensusPathDB: Physical and indirect interactions from human, yeast and mouse are collected from primary sources and curated (Kamburov *et al.*, 2013). Network analysis and gene set enrichment analysis is provided. This database also includes information about protein-drug interactions. A confidence score is provided for each interaction.
- HAPPI: HAPPI is a database of experimentally identified and predicted human protein-protein interactions collected from primary databases (Chen *et al.*, 2017). All interactions are curated and scored.
- HIPPIE: A human protein-protein interaction database with confidence scoring and network analysis tools (Alanis-Lobato *et al.*, 2017). Interactions are collected from multiple PPI databases.
- DroID: A database of physical and genetic interactions between proteins, DNA and RNA from the fly, *Drosophila melanogaster* (Murali *et al.*, 2011). It provides extensive links to fly specific databases such as FlyBase (Gramates *et al.*, 2017). It also includes normalized gene expression values and expression correlation values. Orthologous interactions predicted from known human, worm and yeast interactions are also part of this database.
- Interactome3D: A derived database that collects PPIs from primary source databases and provides 3D structural annotations to the interacting proteins (Mosca *et al.*, 2013). 3D structures are predicted for the complete or partial protein complex, when unknown, along with the binding mode.

Quality Assessment and Scoring

Most of the experimentally identified PPIs in databases are detected using high-throughput experimental methods on a proteome-wide scale and potentially contain spurious interactions (Von Mering *et al.*, 2002; Wodak *et al.*, 2013). In spite of the increasing

accuracy of these methods, assessing the existing interaction data for reliability is important before its use in downstream analyses. As described earlier, several PPI databases now provide a reliability score for the interactions that they contain.

Experimentally detected interactions can be scored in two ways. Firstly, they can be experimentally verified either by confirming a small set of the identified interactions or by assessing the accuracy of the experimental method to estimate the fraction of high-confidence interactions among those detected (Luck *et al.*, 2017). With the increase in experimental validation and assessment, most primary and derived databases provide information about author assigned confidence for interactions and experimental datasets. Secondly, interactions can be assessed computationally to identify the probability of the association or interaction being true and occurring in the cell.

The accuracy of the interaction detected is calculated using information about the type of association detected (physical or indirect), the method used to detect the interaction (Y2H, AP-MS), the type of experiment (small-scale or high-throughput) and the number of publications that support the interaction. Several databases including IntAct, MINT, mentha, HAPPI, HIPPIE and HitPredict incorporate these experimental details into the reliability score they calculate.

To assess the probability of an interaction occurring *in vivo*, the features of the interacting proteins or the presence of homologous interactions may be used. HitPredict uses three characteristics to calculate an annotation score for an interaction – the presence of a homologous interaction, the same gene ontology terms associated with the interacting proteins and the presence of domains that are observed to interact in 3D structural data in the interacting proteins (Patil and Nakamura, 2005). The annotation score is combined with the experiment score, calculated using information about the experimental method, to give the final confidence score in HitPredict (Lopez *et al.*, 2015). STRING calculates the probability of two random pairs of proteins binding and includes a score based on homology and co-occurrence to calculate the final score (Szklarczyk *et al.*, 2015). The reliability score in mentha is based on the number of publications supporting an interaction and the type of method used to detect the method (Calderone *et al.*, 2013). ConsensusPathDB integrates interaction assessment scores calculated using graph-based topology, literature evidence, pathway co-occurrence of the interacting proteins and the functional term similarities (Kamburov *et al.*, 2013). iRefWeb calculates the reliability score using the number of publications supporting the interaction, the method used to identify the interaction and the presence of the same interaction in other species (Turner *et al.*, 2010). HIPPIE uses a semi-automated scoring procedure based on experimental evidence of the interaction (Alanis-Lobato *et al.*, 2017). HAPPI uses a combination of heuristic scores for the source and detection method to calculate a final score (Chen *et al.*, 2017).

With each database adopting a different method to assess the interaction quality, it is not surprising that the high-confidence interactions identified by each database are different (Wodak *et al.*, 2013). Another reason for differences in the interactions deemed as high-confidence between databases is the different gold standard interactions that are used by these methods to evaluate their performance and set a threshold score that differentiates between the high and low confidence interactions. The gold standard datasets can be often biased and error-prone and tend to sometimes result in incorrect threshold scores. Therefore, it must be noted that while interaction scores are important for assessing the quality of interactions, they must be checked across multiple databases for the interactions of interest and used with caution when filtering interactions.

Community Standards

Given the large number of PPI databases and the increasing number of interactions in these databases, inter-operability is an issue that needs to be addressed. With each database having its own data model, terminology, filtering and curation criteria, combining data from multiple databases and mapping the proteins to common identifiers has become tedious and time-consuming. Community standards have been introduced by the Human Proteome Organisation Proteomics Standards Initiative – Molecular Interactions (HUPO PSI-MI) to address this issue (Orchard and Hermjakob, 2008).

PSI-MI has developed common data formats for sharing interaction data in the form of XML (PSI-MI XML) and tab-delimited (MITAB) files. Controlled vocabulary has been developed to describe interaction properties such as the molecular features of the interactors (protein, DNA, etc.), the type of interaction (direct, genetic, etc.) and the interaction detection method (biophysical, biochemical, etc.). The controlled vocabulary includes a standardized term to describe the interaction feature and an accession number that can be used to programmatically search and filter interactions based on their features (Orchard, 2012). HUPO PSI-MI also provides a standardized method for calculating the interaction score in the form of a PSIScore (Villaveces *et al.*, 2015). Most of the databases now provide downloadable files in these file formats and several follow the use of controlled vocabularies.

It is also possible to programmatically access the data within PPI databases so that it can be combined and analyzed. The PSI Common Query InterfaCe (PSICQUIC) is a query interface that allows users to programmatically access interaction databases and get data in PSI-MI XML or MITAB formats (Orchard, 2012). Depending on the database compliance, PSICQUIC also allows users to query for proteins using multiple identifiers. The user can query the most recent version of multiple databases that provide this web service. The PSICQUIC View at EMBL-EBI provides information about the databases that implement this service and their current status. Some databases like IntAct, BioGrid, MINT DIP, InnateDB, STRING and mentha are available through PSICQUIC.

Along with the definition of standardized formats and the availability of a standardized query interface, it is also important to specify common standards for data quality and curation. The International Molecular Exchange (IMEx) consortium has established common curation rules for PPI data. IntAct, MINT, DIP, MatrixDB and InnateDB are some of the members of the consortium. Member databases are committed to curating the data from published papers through a central database and web service, IMEx Central. The data is curated to standards specified by the IMEx consortium. Data curation is divided between the

members of the consortium to prevent overlapping curation efforts. The IMEx consortium provides users with a non-redundant set of interactions through the PSICQUIC service (Orchard *et al.*, 2012).

Future Directions

With the improvement in interaction detection technology and the rapidly increasing number of PPIs being detected, the role of databases in collecting, organizing and curating these interactions has become crucial. The existing PPI databases are doing an excellent job of making the interactions available to the scientific community. The established standards and web services have eased the access to interactions and their integration for downstream network analyses. It has now become possible to perform large-scale network analysis studies and to combine other omics and disease-related data in the context of the network to identify activated cellular pathways and novel functions of proteins.

Some problems associated with the data in PPI databases are the result of the properties of the techniques used to identify these interactions. For instance, there is a known functional bias in the interaction data as a result of the techniques used to detect the PPIs (Yu *et al.*, 2008). For example, Y2H data is enriched in interactions involving signal transduction while AP-MS interactions are enriched for proteins functioning in translation. This may be the result of Y2H being more suited to identify transient interactions while AP-MS being better for the identification of protein complexes. Current high-throughput methods also provide no information about the affinity of interactions, protein abundance, post-translational modifications of the proteins, tissue-specific expression patterns of genes, and the isoform participating in the interaction. All these are important properties of proteins and are known to impact the interaction landscape in the cell.

An estimate of the size of interaction universe in organisms shows that many new PPIs are yet to be discovered (Vidal and Fields, 2014). As the amount of protein-protein interaction data increases and the quality of these interactions improves, resources have to evolve to keep up with these changes. Therefore, existing databases will have to scale and improve their integration, curation and annotation methods. Firstly, integrating interaction data with other omics data needs to become easier through the use of extensive cross-referencing and identifier mapping by all databases. Secondly, the depth of curation needs to be improved in primary as well as derived databases. Obsolete identifiers, incorrect methods, interaction types, evidence types and other annotations are some of the errors that can creep into PPI data. Automatic, or semi-automatic, methods are necessary to correct these errors and reduce the task of manual curators. The scoring schemes used by different databases can also lead to different high confidence datasets and hence, greater uniformity in scoring of interactions is recommended. Finally, methods to analyze and visualize the interaction information are also important and need to be integrated into the databases. Basic network analyses and function enrichment analyses methods will dramatically increase the utility of PPI databases.

Despite these shortcomings, protein-protein interaction databases are critical for the future of large-scale biological data analyses. The next several years will be important as these databases develop new methods to deal with the large amount of data being generated as they make it easily accessible and available for analysis.

See also: Alignment of Protein-Protein Interaction Networks. Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Biological Pathway Analysis. Community Detection in Biological Networks. Data Storage and Representation. Graphlets and Motifs in Biological Networks. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Pathway Informatics. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Functional Annotation. Protein–Protein Interaction Databases. Protein-Protein Interactions: An Overview. Proteomics Data Representation and Databases. Two Decades of Biological Pathway Databases: Results and Challenges. Visualization of Biomedical Networks

References

- Alanis-Lobato, G., Andrade-Navarro, M.A., Schaefer, M.H., 2017. HIPPIE v2.0: Enhancing meaningfulness and reliability of protein–protein interaction networks. *Nucleic Acids Res* 45, D408–D414.
- Ammari, M.G., Gresham, C.R., McCarthy, F.M., Nanduri, B., 2016. HPIDB 2.0: A curated database for host–pathogen interactions. *Database: J Biol Databases Curation* 2016, baw103.
- Bader, G.D., Hogue, C.W., 2002. Analyzing yeast protein–protein interaction data obtained from different sources. *Nat Biotechnol* 20, 991–997.
- Breuer, K., Foroushani, A.K., Laird, M.R., *et al.*, 2013. InnateDB: Systems biology of innate immunity and beyond – Recent updates and continuing curation. *Nucleic Acids Res* 41, D1228–D1233.
- Brown, K.R., Jurisica, I., 2005. Online predicted human interaction database. *Bioinformatics* 21, 2076–2082.
- Cafarelli, T.M., Desbuleux, A., Wang, Y., *et al.*, 2017. Mapping, modeling, and characterization of protein–protein interactions on a proteomic scale. *Curr Opin Struct Biol* 44, 201–210.
- Calderone, A., Castagnoli, L., Cesareni, G., 2013. Mentha: A resource for browsing integrated protein–interaction networks. *Nat Methods* 10, 690–691.
- Ceol, A., Chatr Aryamontri, A., Licata, L., *et al.*, 2010. MINT, the molecular interaction database: 2009 Update. *Nucleic Acids Res* 38, D532–D539.
- Chatr-Aryamontri, A., Oughtred, R., Boucher, L., *et al.*, 2017. The BioGRID interaction database: 2017 Update. *Nucleic Acids Res* 45, D369–D379.
- Chen, J.Y., Pandey, R., Nguyen, T.M., 2017. HAPPI-2: A comprehensive and high-quality map of human annotated and predicted protein interactions. *BMC Genom* 18, 182.
- Cline, M.S., Smoot, M., Cerami, E., *et al.*, 2007. Integration of biological networks and gene expression data using cytoscape. *Nat Protoc* 2, 2366–2382.
- Costanzo, M., Baryshnikova, A., Bellay, J., *et al.*, 2010. The genetic landscape of a cell. *Science* 327, 425–431.

- Giot, L., Bader, J.S., Brouwer, C., *et al.*, 2003. A protein interaction map of *Drosophila melanogaster*. *Science* 302, 1727–1736.
- Gramates, L.S., Marygold, S.J., Santos, G.D., *et al.*, 2017. FlyBase at 25: Looking to the future. *Nucleic Acids Res* 45, D663–D671.
- Hein, M.Y., Hubner, N.C., Poser, I., *et al.*, 2015. A human interactome in three quantitative dimensions organized by stoichiometries and abundances. *Cell* 163, 712–723.
- Ho, Y., Gruhler, A., Heilbut, A., *et al.*, 2002. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* 415, 180–183.
- Huttlin, E.L., Bruckner, R.J., Paulo, J.A., *et al.*, 2017. Architecture of the human interactome defines protein communities and disease networks. *Nature* 545, 505–509.
- Huttlin, E.L., Ting, L., Bruckner, R.J., *et al.*, 2015. The BioPlex network: A systematic exploration of the human interactome. *Cell* 162, 425–440.
- Ito, T., Chiba, T., Ozawa, R., *et al.*, 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci USA* 98, 4569–4574.
- Kamburov, A., Stelzl, U., Lehrach, H., Herwig, R., 2013. The ConsensusPathDB interaction database: 2013 Update. *Nucleic Acids Res* 41, D793–D800.
- Kotlyar, M., Pastrello, C., Sheahan, N., Jurisica, I., 2016. Integrated interactions database: Tissue-specific view of the human and model organism interactomes. *Nucleic Acids Res* 44, D536–D541.
- Launay, G., Salza, R., Multedo, D., *et al.*, 2015. MatrixDB, the extracellular matrix interaction database: Updated content, a new navigator and expanded functionalities. *Nucleic Acids Res* 43, D321–D327.
- Li, S., Armstrong, C.M., Bertin, N., *et al.*, 2004. A map of the interactome network of the metazoan *C. elegans*. *Science* 303, 540–543.
- Lopez, Y., Nakai, K., Patil, J., 2015. HitPredict version 4: Comprehensive reliability scoring of physical protein-protein interactions from more than 100 species. *Database (Oxford)*. 2015:bav117.
- Luck, K., Sheynkman, G.M., Zhang, I., Vidal, M., 2017. Proteome-scale human interactomics. *Trends Biochem Sci* 42, 342–354.
- Mosca, R., Ceol, A., Aloy, P., 2013. Interactome3D: Adding structural details to protein networks. *Nat Meth* 10, 47–53.
- Murali, T., Pacifico, S., Yu, J., *et al.*, 2011. Droid 2011: A comprehensive, integrated resource for protein, transcription factor, RNA and gene interactions for *Drosophila*. *Nucleic Acids Res* 39, D736–D743.
- Orchard, S., 2012. Molecular interaction databases. *Proteomics* 12, 1656–1662.
- Orchard, S., Ammari, M., Aranda, B., *et al.*, 2014. The MIntAct project – IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res* 42, D358–D363.
- Orchard, S., Hermjakob, H., 2008. The HUPO proteomics standards initiative – Easing communication and minimizing data loss in a changing world. *Brief Bioinform* 9, 166–173.
- Orchard, S., Kerrien, S., Abbani, S., *et al.*, 2012. Protein interaction data curation: The International Molecular Exchange (IMEx) consortium. *Nat Methods* 9, 345–350.
- Pagel, P., Kovac, S., Oesterheld, M., *et al.*, 2005. The MIPS mammalian protein–protein interaction database. *Bioinformatics* 21, 832–834.
- Papanikolaou, N., Pavlopoulos, G.A., Theodosiou, T., Iliopoulos, I., 2015. Protein–protein interaction predictions using text mining methods. *Methods* 74, 47–53.
- Patil, A., Nakai, K., Nakamura, H., 2011. HitPredict: A database of quality assessed protein–protein interactions in nine species. *Nucleic Acids Res* 39, D744–D749.
- Patil, A., Nakamura, H., 2005. Filtering high-throughput protein–protein interaction data using a combination of genomic features. *BMC Bioinform* 6, 100.
- Prasad, T.S., Kandasamy, K., Pandey, A., 2009. Human protein reference database and human proteinpedia as discovery tools for systems biology. *Methods Mol Biol* 577, 67–79.
- Rolland, T., Tasan, M., Charleatoux, B., *et al.*, 2014. A proteome-scale map of the human interactome network. *Cell* 159, 1212–1226.
- Rual, J.F., Venkatesan, K., Hao, T., *et al.*, 2005. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 437, 1173–1178.
- Ruepp, A., Waaghele, B., Lechner, M., *et al.*, 2010. CORUM: The comprehensive resource of mammalian protein complexes – 2009. *Nucleic Acids Res* 38, D497–D501.
- Salwinski, L., Miller, C.S., Smith, A.J., *et al.*, 2004. The database of interacting proteins: 2004 Update. *Nucleic Acids Res* 32, D449–D451.
- Skrabaneck, L., Saini, H.K., Bader, G.D., Enright, A.J., 2008. Computational prediction of protein–protein interactions. *Mol Biotechnol* 38, 1–17.
- Stelzl, U., Worm, U., Lalowski, M., *et al.*, 2005. A human protein–protein interaction network: A resource for annotating the proteome. *Cell* 122, 957–968.
- Szklarczyk, D., Franceschini, A., Wyder, S., *et al.*, 2015. STRING v10: Protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Res* 43, D447–D452.
- Turner, B., Razick, S., Turinsky, A.L., *et al.*, 2010. iRefWeb: Interactive analysis of consolidated protein interaction data and their supporting evidence. *Database* 2010, baq023.
- Uetz, P., Giot, L., Cagney, G., *et al.*, 2000. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature* 403, 623–627.
- Vidal, M., Fields, S., 2014. The yeast two-hybrid assay: Still finding connections after 25 years. *Nat Methods* 11, 1203–1206.
- Villaveces, J.M., Jimenez, R.C., Porras, P., *et al.*, 2015. Merging and scoring molecular interactions utilising existing community standards: Tools, use-cases and a case study. *Database (Oxford)*. p. 2015.
- Von Mering, C., Krause, R., Snel, B., *et al.*, 2002. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* 417, 399–403.
- Wan, C., Borgeson, B., Phanse, S., *et al.*, 2015. Panorama of ancient metazoan macromolecular complexes. *Nature* 525, 339–344.
- Wodak, S.J., Vlasblom, J., Turinsky, A.L., Pu, S., 2013. Protein–protein interaction networks: The puzzling riches. *Curr Opin Struct Biol* 23, 941–953.
- Yazaki, J., Galli, M., Kim, A.Y., *et al.*, 2016. Mapping transcription factor interactome networks using HaloTag protein arrays. *Proc Natl Acad Sci USA* 113, E4238–E4247.
- Yu, H., Braun, P., Yildirim, M.A., *et al.*, 2008. High-quality binary protein interaction map of the yeast interactome network. *Science* 322, 104–110.

Relevant Website

<http://www.ebi.ac.uk/Tools/webservices/psicquic/view/main.xhtml>
EMBL-EBI.

Computational Approaches for Modeling Signal Transduction Networks

Zhongming Zhao, The University of Texas Health Science Center at Houston, Houston, TX, United States

Junfeng Xia, Anhui University, Hefei, China

© 2019 Elsevier Inc. All rights reserved.

Introduction

Signal transduction is an essential biological process in cellular systems. It involves numerous biological functions in cell, and its disruption may lead to various diseases, phenotypes and drug treatment outcomes. Through dynamic and refined signal transduction, molecules connect and interact in a predefined fashion in signaling pathways so that the living organisms fulfill overall function in a harmonious way. Aberrant signal transduction in the pathways may cause dysregulation of biological processes, resulting in diseases such as cancer (Sever and Brugge, 2015). This article first introduces the basic concept about the signaling transduction, biological pathways, and molecular networks. Then, we describe the signaling transduction networks (STNs) in complex diseases by using cancer and its mitogen-activated protein kinase (MAPK) pathway as a specific example. MAPK pathway plays critical roles in multiple processes associated with many types of cancer and other diseases. Next, we introduce essential structural motifs in STNs, including feedback and feed forward loops. Finally, we review the databases and analysis tools that have been used in STNs and highlight some applications of computational modeling of STNs to complex disease studies.

Basic Concept in Signal Transduction Networks

Complex cellular activities such as cell growth and division, cell death, cell fate, and cell motility are generally regulated through the refined response of cells to stimuli in their environment. The transportation of an extracellular cue through cell membrane to the nucleus and intracellular signal passing between organelles or from organelles to other cellular components in response to cell stress or metabolic status are all referred to signal transduction (Liu *et al.*, 2008). Signal transduction allows cells to respond to the cues from either neighboring cells or even distant cells and to change their behavior accordingly. Cells constantly and concordantly receive a variety of signals that will affect one or multiple intersections within a signaling transduction pathway, as well as multiple signaling transduction pathways. Although numerous cell signaling transduction events and related molecular mechanisms have been uncovered, a deep understanding of cellular signaling transduction has still remained a main challenge in biological and biomedical research. This is mainly because there are thousands of small molecules, often with different forms or isoforms, that interact in a dynamic and complex fashion (e.g., spatiotemporal dimensions). Such a complex, dynamic feature is a major limit to apply the currently available statistical or computational modeling. However, deep deciphering of the complexity of cell signaling is ultimately essential for investigators to understand the normal biological functions as well as the pathophysiology of diseases. Such knowledge builds the foundation on the development of enhanced therapeutic strategies for many diseases, including cancer (Csermely *et al.*, 2016). So far, many molecular therapeutic approaches in cancer treatment have been based on the actionable mutations in kinase domains, which are the critical components in signaling pathways (Cheng *et al.*, 2014).

Extensive signaling studies have reconstructed a great number of carefully curated canonical signaling pathways, including the Jak-STAT, NF- κ B, Wnt, Notch, Ras/ERK, and G protein-coupled receptor pathways (Housden and Perrimon, 2014). These pathways form a complex network called signal transduction network (STN) (Kolch *et al.*, 2015) that allows cells to integrate multiple environmental signals, and respond with output signals specifically tailored to the environmental change or condition. In these STNs or subnetworks, nodes are primarily the molecules such as signaling proteins and microRNAs, while edges (links between STN nodes) are activating or inhibitory physical connections of participating signaling proteins, microRNAs and other regulators. The regulation involves in binding at the critical sites of the molecules, enzymatic reaction (e.g., phosphorylation and dephosphorylation), and allosteric regulation, among others (Csermely *et al.*, 2016).

The Raf–MEK–ERK MAPK Signaling Pathway in Cancer

There are many signaling pathways in our knowledgebase. For example, one of the popularly used pathway databases, Kyoto Encyclopedia of Genes and Genomes (KEGG) database, currently curates a total of 320 human pathways (Kanehisa *et al.*, 2016). Illustration of each pathway and how it works is beyond the scope of this article. Here, we take signaling pathway in cancer as one example. Tumorigenesis has been perceived as dysfunction of STNs that regulate molecular communications and cellular processes (Kolch *et al.*, 2015). During cancer initiation and development, the STN undergoes systematic changes in terms of expression level of nodes, as well as in the sign (inhibition or activation) and weight (strength) of their edges. Dysregulation of cellular signal transduction pathways underlies most well-defined features in cancer cells (Sever and Brugge, 2015). We performed literature survey using keywords “signal transduction”, “network”, and “cancer”. As shown in Fig. 1, it clearly shows the research in signaling pathways in cancer has been growing rapidly and consistently. We expect the number of cancer STN studies to increase substantially in the near future, considering that more –omics data and better pathway and network tools will be released in the near

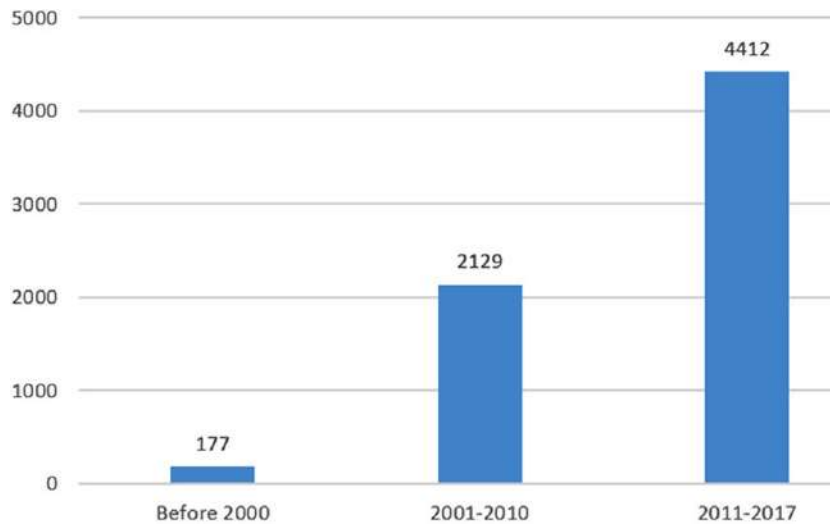


Fig. 1 Number of publications (y -axis) from PubMed title/abstract search using keywords “signal transduction” AND “network” AND “cancer”. This simple search is not to have an exhaustive list but shows a strong trend of the growing studies in cancer by using signaling transduction network approaches.

future. To cover all of the signaling molecules involved and their myriad contributions to cancer would be too much in this article. We therefore focus on mitogen-activated protein kinase (MAPK or MAP kinase) signaling pathway, which attracts most attention and research interest due to its central role in regulating cell proliferation and survival and significant impact in various types of cancer (Roberts and Der, 2007; Cheng *et al.*, 2013).

In cancer, especially in melanoma (Cheng *et al.*, 2013), MAPK signaling cascades are critical and constitutively activated by a variety of mechanisms. These activation mechanisms, as well as the key genetic alterations detected in the pathway, have enabled the investigators to successfully develop kinase inhibitors and applied to the treatment of cancer, namely kinase or pathway based, molecularly targeted therapies. MAPK signaling pathway is one of key cellular mechanisms involving in cancer through promoting cell proliferation, survival, invasion, and tumor angiogenesis. This pathway consists of four widely studied MAP kinases, and these cascades sequentially transfer proliferative signals from the cell surface receptors into the nucleus via a series of protein phosphorylation events. Each MAPK cascade is comprised of three protein kinases that act as a signaling relay: MAP kinase kinase kinase (MAPKKK, MAP3K, or MEKK), MAP kinase kinase (MAPKK, MAP2K, or MEK), and MAPK. The four alternative terminal MAP kinases are the extracellular signal-regulated kinase 1 and 2 (ERK1/2), the c-Jun amino-terminal kinases (JNK12/3), p38 kinases, and ERK5. Among these four cascades, RAS-RAF-MEK-ERK1/2 pathway (Fig. 2) has attracted most research interest due to its central involvement in the regulation of cell proliferation and cell survival (Roberts and Der, 2007). In response to binding of various activators such as growth factors and hormones, receptor tyrosine kinases (RTKs; e.g., KIT) form a dimer, which causes the activation of Ras to GTP-bound state, and facilitates the recruitment of Raf (e.g., Braf) from the cytosol to cell membrane where it becomes an activated form. The adaptor protein growth factor receptor-bound protein 2 (Grb2), which comprises of Src homology 2 domain, promotes the binding of guanine nucleotide exchange factor son of sevenless (SOS), thereby leading to the formation of active Ras-GTP. Once being activated, Raf kinases phosphorylate and activate Mek proteins (Mek1 and Mek2), which subsequently lead to the phosphorylation and activation of Erk (Erk1 and Erk2). Finally, phosphorylated Erk1/2 proteins enter nucleus and regulate the activities of a large number of substrates – an estimate of over 160 proteins including ELK-1, Fos, Myc, and others – to control multiple cellular processes (Roberts and Der, 2007). The cascade of kinase reaction from RAF to MEK to ERK is a classic example of a signaling pathway. This pathway also represents an excellent example to illustrate feedback regulation (see below) and signal amplification.

Structural Motifs in STN

In a highly clustered STN, essential modules can be detected by identifying recurring patterns of highly interlinked groups of nodes (Liu *et al.*, 2008). A number of structural motifs exist in STNs, such as feed forward loops (FFLs) (Mangan and Alon, 2003) and feedback loops (FBLs) (Brandman and Meyer, 2008). These motifs enable a STN to respond selectively to a particular stimulus at the dynamic signal level, and therefore, result in precision action as the response.

The FFL (Mangan and Alon, 2003) is composed of three signaling molecules (e.g., transcription factors or microRNAs): a regulator (X) that regulates a molecule Y and targets a molecule Z, and Z is jointly regulated by X and Y. Because each of the three regulatory interactions in the FFL can be either positive (activation) or negative (repression), there are eight possible structural types of FFL (Fig. 3). Four of these FFLs are termed “coherent”, where the sign of the direct signaling path (from X to Z) is the same

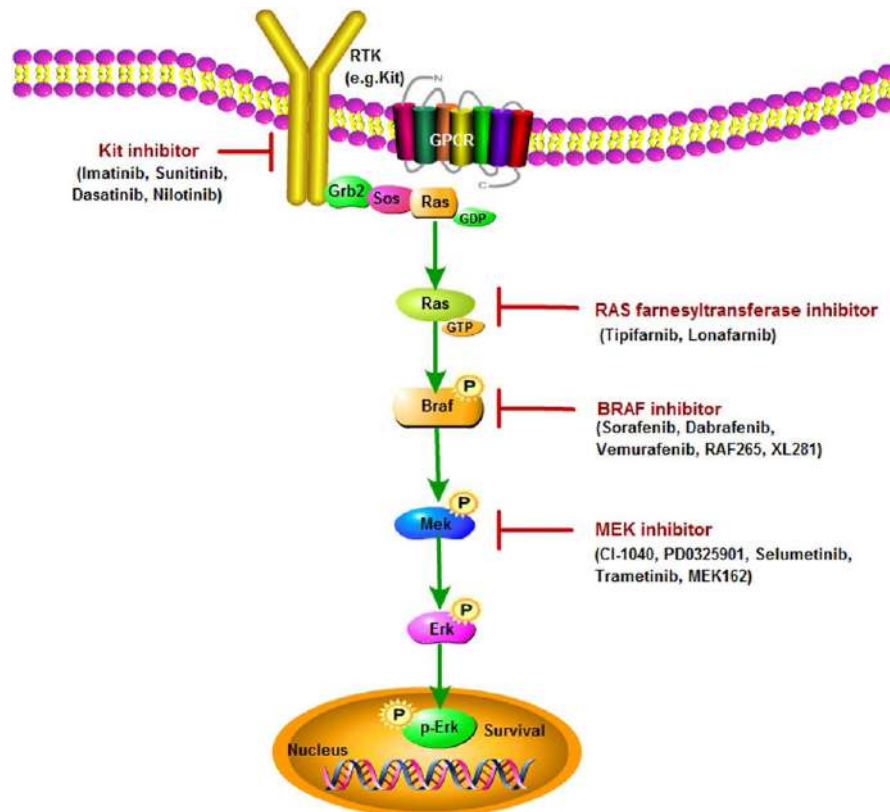


Fig. 2 Schematic diagram of the MAPK signaling pathway and selected inhibitors.

as the overall sign of the indirect signaling path (from X through Y to Z). The remaining four FFL configurations are termed “incoherent” where the signs of the direct and indirect signaling paths are opposite. In complex disease research field, the FFL approach was first applied to schizophrenia in 2010, where 32 FFLs among the compiled schizophrenia-related transcription factors (TFs), miRNAs and genes were identified (Guo *et al.*, 2010). This approach has been frequently applied to cancer and other disease or phenotype since then, and a database for TF-miRNA-gene modules has been developed.

The FBL (Brandman and Meyer, 2008) is another common network motif that connect output signals back to their inputs. In this case, a downstream signaling molecule of a pathway activates (positive feedback) or represses (negative feedback) an upstream signaling molecule (Fig. 3). Positive FBLs can be utilized to control the response of a signaling pathway to different stimulation durations by locking the system into an on or off state following a transient signal. On the contrary, negative FBLs can cause a change in the dynamics of signaling because in that scenario, an activation of the signaling pathway can subsequently lead to a repression of the same pathway. And this may have several different effects depending on the kinetics of the FBL. FBL as a motif unit has been widely used in biomedical and biological research, such as the kinase cascade introduced in the subsection ‘The Raf–MEK–ERK MAPK signaling pathway in cancer’.

STN Databases and Enrichment Analysis Tools

There have been numerous studies of STNs or the subcomponents of a STN. The rich information has been available from literatures, databases and other resources. Accordingly, knowledgebase on STNs has become one of the important efforts to summarize the previous findings. Such knowledgebase can provide us with a systematic view of the biological processes and assist us for future studies of disease mechanisms (Liu *et al.*, 2008). To meet various needs, researchers have established many databases through careful curation and storage of such data, as well as allowing users to access and visualize the biological pathways and molecular interactions (Table 1). Among these databases, KEGG database (Kanehisa *et al.*, 2016) is one of the popular, manually curated pathway resources. It includes the knowledge in the molecular interaction, reaction and relation networks, and pharmacological information. KEGG consists of seven types of network context: metabolism, genetic information processing, environmental information processing, cellular processes, organismal systems, human diseases, and drug development. The KEGG pathway information is machine-readable via KEGG Markup Language. KEGG includes many canonical signaling transduction pathways. It was initially publicly available but later the access requires fee-based subscription. Reactome (Fabregat *et al.*, 2014) is another popular, well curated, and peer reviewed pathway database. The current version (V62) comprises 11,302 reactions (e.g.,

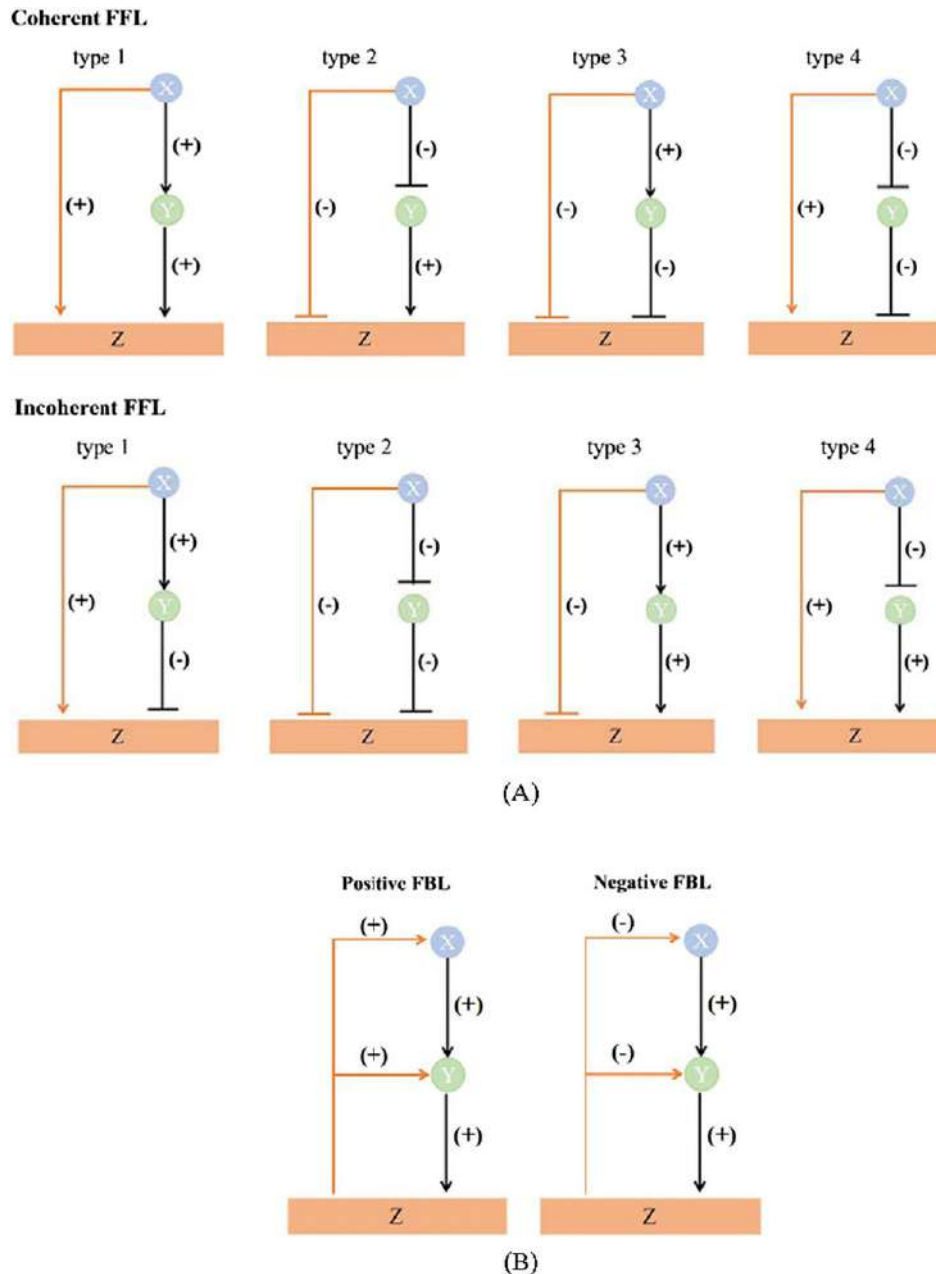


Fig. 3 Motifs of feed forward loop (FFL) and feedback loop (FBL) in signal transduction network. (A) The eight types of FFLs can be classified into four coherent and four incoherent FFLs. In coherent FFLs, the sign of the direct path from signaling molecule X to output Z is the same as the overall sign of the indirect path through signaling molecule Y. Incoherent FFLs have inconsistent or opposite signs for the two paths. (B) The two types of FBLs: a downstream molecule activates (positive FBL) or represses (negative FBL) an upstream molecule.

phosphorylation, acetylation, etc.) organized into 2176 human pathways, involved in 10,878 proteins and 1768 small molecules. These data were extracted from 27,526 publications. For machine readability, Reactome adopted the Systems Biology Markup Language (SBML) format for systematic access and analysis. In addition to those self-curated pathway databases, other databases assemble different sources of pathways. For example, the Pathway Commons database (Cerami *et al.*, 2011) contains extensive collection of pathway and interaction data from partner databases such as KEGG, Reactome, and MSigDB (the Molecular Signatures Database) (Subramanian *et al.*, 2005) and is represented in the Biological Pathway Exchange (BioPAX) standard. WikiPathways (Kutmon *et al.*, 2016) is another integrated database of biological pathways maintained by and for the scientific community. The advantage of WikiPathways lies in the scalable, community-based curation and unrestricted pathway model through accepting any pathway that researchers find useful in their work. Table 1 provides a list of pathway databases, including the description and access URLs. Each of these pathway databases include signaling transduction pathways.

Table 1 Summary of databases and tools for signaling transduction networks

Name	URLs	Description
KEGG	www.genome.jp/kegg/pathway.html	A collection of manually pathway maps representing our knowledge on the molecular interaction, reaction and relation networks
Reactome	www.reactome.org	A free, open-source, curated and peer reviewed pathway database
Gene Ontology	www.geneontology.org	A computational representation of gene products in terms of their associated biological processes, cellular components and molecular functions
Pathway Commons	www.pathwaycommons.org	An integrated resource from public pathway and interactions databases
NCI PID	https://pid.nci.nih.gov	Biomolecular interactions and cellular processes assembled into authoritative human signaling pathways
BioCarta	https://cgap.nci.nih.gov/Pathways/BioCarta_Pathways	Online maps of metabolic and signaling pathways
WikiPathways	www.wikipathways.org	A database of biological pathways maintained by and for the scientific community
UCSD Signaling Gateway	www.signalinggateway.org	A database providing essential information on the proteins involved in cell signaling
MSigDB	https://software.broadinstitute.org/gsea/msigdb	A collection of annotated gene sets for gene set enrichment analysis
NetPath	www.netpath.org	A manually curated resource of signal transduction pathways in humans
Signalink	www.signalink.org	A signaling pathway resource with multi-layered regulatory networks
Pascal	www2.unil.ch/cbg/index.php?title=Pascal	A tool for gene scoring and pathway analysis from GWAS results
DAVID	https://david.ncifcrf.gov	A comprehensive set of functional annotation tools to understand biological meaning behind large list of genes
webGestalt	www.webgestalt.org	A suite of tools for functional enrichment analysis in various biological contexts
Ingenuity	www.qiagenbioinformatics.com/products/ingenuity-pathway-analysis	A comprehensive analysis and search tool of global molecular network based on manual curation
ToppGene	https://toppgene.cchmc.org	A portal for gene list enrichment analysis and candidate gene prioritization based on functional annotations and protein interactions network

Based on the prior knowledge (pathway databases), many tools have been developed to interpret gene lists derived from the high-throughput experiments (Wadi *et al.*, 2016). These pathway analysis tools use various strategies to aggregate genes across sets of the related genes (Table 1). The most common approach used for gene set enrichment of pathway function is based on binary enrichment tests, which rely on a threshold parameter to determine which genes are significantly associated with the disease or trait. One disadvantage of such an approach is that potential contributions of the weakly associated genes that just missed the threshold would be lost and there is no clear rule on the threshold setting. To address this problem, Lamparter *et al.* (2016) presented a powerful tool called Pascal (Pathway scoring algorithm) for computing pathway scores. Pascal integrates individual gene scores without the need for a tunable threshold parameter to dichotomize gene scores for binary membership enrichment analysis. As a result, it shows better discovery of confirmed pathways compared with other popular methods. However, challenges and limitations exist in all the pathway enrichment methods currently available. These challenges include the multiple test correction, redundancy of the genes shared by different pathways, appropriate statistical tests, incomplete pathway data being available to test, and gene length bias on measuring the signals from the genomic data, among others (Jia *et al.*, 2011; Wang *et al.*, 2010).

Computational Modeling of STNs and Its Applications

In this section, we present several examples of the computational modeling of STNs.

Genetic Mutations Impact STNs

Somatic mutations may cause abnormal cell growth, resulting in tumorigenesis. These mutations confer a selective advantage to the cancer cells, which together with changes in the microenvironment, promote cancer growth and progression. Analyses of massive amounts of cancer genomics data generated from several large-scale cancer genome sequencing projects such as The

Cancer Genome Atlas (TCGA) and the International Cancer Genome Consortium (ICGC) have discovered more than 68 million simple somatic mutations and many other types of somatic mutations (as of December 11, 2017). However, most of these mutations are passenger mutations. Typically, only two to eight in a cancer cell are driver mutations that cause progression of the cancer (Vogelstein *et al.*, 2013). These may be single-nucleotide variants, small insertions or deletions, copy number variations, or structure variants.

Based on these mutation data, we can connect the genetic mutations in cancer cells within STNs that control processes associated with tumorigenesis and place them in the context of distortions of wider STNs that fuel cancer progression. Genetic mutations can cause abnormal expression of proteins involved in a STN (e.g., gene amplification) or change the individual contributions of a specific STN by producing mutant proteins whose activities are dysregulated (e.g., point mutations, truncations, and fusions). Alternatively, deletions in genes may inactivate negative regulators, which may cause the absence of the specific nodes within a STN. Moreover, multiple mutations can occur within a cell simultaneously and among cells of a specific tumor type. Recently, Zhao *et al.* (2017) proposed a computational oncoproteomics approach, namely kinome-wide network module for cancer pharmacogenomics (KNMPx), for identifying actionable mutations that rewired STNs by incorporating the somatic missense mutations into the protein phosphorylation sites. Specifically, 746,631 missense mutations in 4997 tumor samples across 16 major cancer types/subtypes from TCGA were integrated into over 170,000 carefully curated non-redundant phosphorylation sites covering 18,610 proteins. Forty-seven mutated proteins (e.g., ERBB2, TP53, and CTNNB1) had enriched missense mutations at their phosphorylation sites in pan-cancer analysis. In addition, tissue-specific kinase-substrate interaction modules altered by somatic mutations that were identified in cancer genomes were significantly associated with patient survival. Interestingly, the authors found that cell lines could highly reproduce oncogenic phosphorylation site mutations identified in primary tumors, supporting the confidence in their associations with sensitivity/resistance of inhibitors targeting EGF, MAPK, PI3K, mTOR, and Wnt signaling pathways. According to the results, the KNMPx approach is powerful for identifying oncogenic alterations in signaling molecules via rewiring phosphorylation-related STNs and for predicting drug sensitivity or resistance in the era of precision oncology.

Pan-Cancer Analysis of Dysregulated Transcription Factor – microRNA FFLs in STNs

Transcription factor and microRNA (miRNA) can mutually regulate each other and jointly regulate their shared target genes. This is an ideal example for FFLs in cellular system. Since 2010, TF-miRNA FFLs have been widely used to identify the cancer-associated genes or miRNAs in many tumor types. As one example, Jiang *et al.* examined dysregulated FFLs in pan-cancer (13 cancer types) (Jiang *et al.*, 2016). They identified 26 pan-cancer FFLs, each of which was found to be dysregulated in at least five tumor types using TCGA data. They found these pan-cancer FFLs could communicate with each other and form functionally consistent subnetworks, such as epithelial to mesenchymal transition-related subnetwork. Many proteins and miRNAs in each subnetwork belong to the same protein family and miRNA family, respectively. Importantly, cancer-associated genes and drug targets were enriched in these pan-cancer FFLs, in which the genes and miRNAs also tended to be hubs and bottlenecks. In addition, they identified potential anticancer indications for existing drugs with novel mechanism of action. Collectively, their study demonstrated the potential of pan-cancer FFLs as critical regulatory units that are useful for elucidating pathogenesis of cancer and developing anticancer drugs.

Decipher STNs for Drug Effect in Complex Disease

A drug exerts its effects typically through a STN cascade, which is non-linear and involves intertwined networks of multiple signaling pathways. Construction of such a STN may help the investigators to identify novel drug targets and further understand drug action. Sun *et al.* developed a novel computational framework, the Drug-specific Signaling Pathway Network (DSPNet), to tackle this issue (Sun *et al.*, 2015). The DSPNet amalgamates the prior drug knowledge and drug-induced gene expression via random walk algorithms. Using the drug metformin as an example, they illustrated this framework and obtained one metformin-specific SPNetwork containing 477 nodes and 1366 edges. The further examination of this subnetwork using one type 2 diabetes (T2D) genome-wide association study (GWAS) dataset, three cancer GWAS datasets, and one GWAS dataset of cancer patients with T2D on metformin indicated that the metformin network was significantly enriched with disease genes for both T2D and cancer, and that the network also included genes that may be associated with metformin-associated cancer survival. The authors also generated a subnetwork to highlight the molecules' crosstalk between T2D and cancer. The follow-up network analyses and literature mining revealed that some valuable insights into the mode of metformin action, which will facilitate our understanding of the molecular mechanisms underlying drug treatments, disease pathogenesis, and identification of novel drug targets and repurposed drugs. As many drugs may target signaling molecules, or through metabolism that links to signaling pathways, novel computational approaches for deciphering STNs will be much needed.

Applications of STNs in Other Fields

Besides cancer, STNs have been frequently used to study the molecular regulation in specific cellular conditions, cell types, epigenetic regulation, or in other complex disease or phenotype. For example, Bernabò *et al.* have applied STN approach to

investigate the cell signal transduction, which is a complex phenomenon and has a central role in cell surviving and adaptation, in non-medical systems. The authors presented two applications of STNs for cell signaling. One is to study the architecture of signaling systems for acquisition of a complex cellular function (i.e., process of activation of spermatozoa) and the other is to find the organization of specific signaling systems that are active in specific cells and/or tissues (i.e., the endocannabinoid system) (Bernabò *et al.*, 2014). In both the applications, the authors found that the STNs follow a scale free and small world topology. Such network characteristics will allow the investigators to identify molecules (e.g., hubs) or modules that play key role in signaling system. STNs approaches have also been applied to study human pluripotent stem cells. As an example, Fgf/MAPK, TGF β /SMAD2,3 and insulin/PI3K signaling pathways are found to be required for maintenance of the stem cell state (Dalton, 2013). Moreover, DNA methylation data has been integrated with other genomic data into STNs for exploring epigenetic regulation patterns for complex disease like bone mineral density (BMD) (Zhang *et al.*, 2015). In synthetic biology field, the progress of STN approaches in synthetic signal transduction implementation has been systematically reviewed in diverse biological contexts including bacteria, yeast, vertebrate, and plant cells (Hansen and Benenson, 2016), and various systems STN concepts have been implemented into metabolic engineering and synthetic biology (He *et al.*, 2016).

Closing Remarks

Complex disease like cancer is highly heterogeneous – it involves dynamic regulation and interactions of STNs at the cellular level. It is often tissue-specific and development stage specific. Identification of STN changes that are critical for cell function, development, or drug binding will substantially improve our knowledge and have potential for healthcare. As we have entered the big data era, we expect more novel, systematic studies will be conducted, leading to not only knowledge discovery in signaling pathways and networks, but also more disease mutations and drug-targetable sites. Advanced statistical and computational methods and tools will be further needed to meet such a strong demand in future. For example, novel tools that can effectively integrate the genomic and genetic data in a spatiotemporal fashion into the STNs will power the investigators to measure the genetic or biological changes in high resolution and more reflective on the cellular condition.

Acknowledgements

Dr. Zhao was partially supported by National Institutes of Health grants [R01LM012806 and R21CA196508]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

See also: Cell Modeling and Simulation. Computational Systems Biology Applications. Computational Systems Biology. Functional Genomics. Graphlets and Motifs in Biological Networks. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Network Models. Network-Based Analysis for Biological Discovery. Networks in Biology. Pathway Informatics. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Post-Translational Modification Prediction. Protein-Protein Interactions: An Overview. Visualization of Biomedical Networks

References

- Bernabò, N., Barboni, B., Maccarrone, M., 2014. The biological networks in studying cell signal transduction complexity: The examples of sperm capacitation and of endocannabinoid system. *Computational and Structural Biotechnology Journal* 11, 11–21.
- Brandman, O., Meyer, T., 2008. Feedback loops shape cellular signals in space and time. *Science* 322, 390–395.
- Cerami, E., Gross, B., Demir, E., *et al.*, 2011. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Research* 39, 685–690.
- Cheng, F., Jia, P., Wang, Q., Zhao, Z., 2014. Quantitative network mapping of the human kinome interactome reveals new clues for rational kinase inhibitor discovery and individualized cancer therapy. *Oncotarget* 5, 3697–3710.
- Cheng, Y., Zhang, G., Li, G., 2013. Targeting MAPK pathway in melanoma therapy. *Cancer and Metastasis Reviews* 32, 567–584.
- Csermely, P., Korcsmaros, T., Nussinov, R., 2016. Intracellular and intercellular signaling networks in cancer initiation, development and precision anti-cancer therapy: Ras acts as contextual signaling hub. *Seminars in Cell & Developmental Biology* 58, 55–59.
- Dalton, S., 2013. Signaling networks in human pluripotent stem cells. *Current Opinion in Cell Biology* 25, 241–246.
- Fabregat, A., Sidropoulos, K., Garapati, P., *et al.*, 2014. The reactome pathway knowledgebase. *Nucleic Acids Research* 42, 472–477.
- Guo, A.Y., Sun, J., Jia, P., Zhao, Z., 2010. A novel microRNA and transcription factor mediated regulatory network in schizophrenia. *BMC Systems Biology* 4, 10.
- Hansen, J., Benenson, Y., 2016. Synthetic biology of cell signaling. *Natural Computing* 15, 5–13.
- He, F., Murabito, E., Westerhoff, H.V., 2016. Synthetic biology and regulatory networks: Where metabolic systems biology meets control engineering. *Journal of the Royal Society Interface* 13, 20151046.
- Housden, B.E., Perrimon, N., 2014. Spatial and temporal organization of signaling pathways. *Trends in Biochemical Sciences* 39, 457–464.
- Jia, P., Wang, L., Meltzer, H.Y., Zhao, Z., 2011. Pathway-based analysis of GWAS datasets: Effective but caution required. *International Journal of Neuropsychopharmacology* 14 (4), 567–572.
- Jiang, W., Mitra, R., Lin, C., *et al.*, 2016. Systematic dissection of dysregulated transcription factor-miRNA feed-forward loops across tumor types. *Briefings in Bioinformatics* 17, 996–1008.
- Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M., 2016. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research* 44, 457–462.

- Kolch, W., Halasz, M., Granovskaya, M., Kholodenko, B.N., 2015. The dynamic control of signal transduction networks in cancer cells. *Nature Reviews Cancer* 15, 515–527.
- Kutmon, M., Riutta, A., Nunes, N., *et al.*, 2016. WikiPathways: Capturing the full diversity of pathway knowledge. *Nucleic Acids Research* 44, 488–494.
- Lamparter, D., Marbach, D., Rueedi, R., Kutalik, Z., Bergmann, S., 2016. Fast and rigorous computation of gene and pathway scores from SNP-based summary statistics. *PLOS Computational Biology* 12 (1), e1004714.
- Liu, W., Li, D., Zhu, Y., He, F., 2008. Bioinformatics analyses for signal transduction networks. *Science China-life Sciences* 51, 994–1002.
- Mangan, S., Alon, U., 2003. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences of the United States of America* 100, 11980–11985.
- Roberts, P.J., Der, C.J., 2007. Targeting the Raf-MEK-ERK mitogen-activated protein kinase cascade for the treatment of cancer. *Oncogene* 26, 3291–3310.
- Sever, R., Brugge, J.S., 2015. Signal transduction in cancer. *Cold Spring Harbor Perspectives in Medicine* 5, a006098.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* 102, 15545–15550.
- Sun, J., Zhao, M., Jia, P., *et al.*, 2015. Deciphering signaling pathway networks to understand the molecular mechanisms of metformin action. *PLOS Computational Biology* 11, e1004202.
- Vogelstein, B., Papadopoulos, N., Velculescu, V.E., *et al.*, 2013. Cancer genome landscapes. *Science* 339, 1546–1558.
- Wadi, L., Meyer, M., Weiser, J., Stein, L., Reimand, J., 2016. Impact of outdated gene annotations on pathway enrichment analysis. *Nature Methods* 13, 705–706.
- Wang, K., Li, M., Hakonarson, H., 2010. Analysing biological pathways in genome-wide association studies. *Nature Reviews Genetics* 11, 843–854.
- Zhang, J.G., Tan, L.J., Xu, C., *et al.*, 2015. Integrative analysis of transcriptomic and epigenomic data to reveal regulation patterns for BMD variation. *PLOS ONE* 10, e0138524.
- Zhao, J., Cheng, F., Zhao, Z., 2017. Tissue-specific signaling networks rewired by major somatic mutations in human cancer revealed by proteome-wide discovery. *Cancer Research* 7, 2810–2821.

Further Reading

- Alon, U., 2007. Network motifs: Theory and experimental approaches. *Nature Reviews Genetics* 8, 450–461.
- Inui, M., Martello, G., Piccolo, S., 2010. MicroRNA control of signal transduction. *Nature Reviews Molecular Cell Biology* 11 (4), 252–263.
- Logue, J.S., Morrison, D.K., 2012. Complexity in the signaling network: Insights from the use of targeted inhibitors in cancer therapy. *Genes & Development* 26, 641–650.
- Prasasya, R.D., Tian, D., Kreeger, P.K., 2011. Analysis of cancer signaling networks by systems biology to develop therapies. *Seminars in Cancer Biology* 21, 200–206.

Cell Modeling and Simulation

Ayako Yachie-Kinoshita, The Systems Biology Institute, Tokyo, Japan and University of Toronto, Toronto, ON, Canada

Kazunari Kaizu, RIKEN Quantitative Biology Center (QBiC), Osaka, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Since the 21st century has begun, the so-called post-genome era, the biggest open questions are how the phenotype is generated from the genotype, and how evolution is achieved from the phenotype. Towards this ultimate goal, systems biology has emerged as a conceptual approach to understand life as an evolvable and robust system (Kitano, 2002). Mathematical modeling and simulation of the biological systems have played the central role in the context of systems-level understanding and predictability of living organisms (Fig. 1).

As with modeling of other systems, the abstraction of the cell properties is the first key step in cell modeling. This step requires the definition of boundaries in the focused cellular system.

The cell contains highly integrated subsystems in itself, although it is often called a building unit for the structure and function of living organisms. The complexity lays in space and time. Moreover, although the cell has independent replicability, its activity is dependent not only internal state but also extrinsic perturbations, that is, its surroundings, which are orchestrated towards robust biological functions. It is thus difficult to define physical and functional boundaries in the cellular system, as is clear from the fact that the cellular behavior observed in vitro is hardly reproduced in vitro. Under these circumstances, cell simulation often requires drastic abstraction of the system to keep the balance of desired trade-offs with predictability.

The demands for cell simulations are also largely regulated by how the technology in molecular biology has evolved. The electronic kinetics emerged in the 1950s and drove the first mathematical modeling of cellular phenotypes reported by Hodgkin–Huxley (Hodgkin and Huxley, 1952), focusing on the dynamics of heart activity by calculating the ensemble of voltage-dependent channels using nonlinear ordinary differential equations, which is still a basis for studying the behavior of ion channels. The protein purification led to massive studies on enzymes in the 1970s that initiated metabolic simulations based on enzyme kinetics (Garfinkel *et al.*, 1970; Heinrich *et al.*, 1978; Tyson and Othmer, 1978). From the late 1990s to the present, with genomics as a start, comprehensive measurements of each biological layer have emerged, including metabolomics, proteomics, and epigenomics, and have changed our view of biological systems towards data-driven science (Hey *et al.*, 2009). In recent years, it has become possible to observe cell behaviors at the single cell resolution (Kalisky *et al.*, 2011; Xia *et al.*, 2013). This has made us aware of the significance of cell–cell heterogeneity at the population level (Iwamoto *et al.*, 2016), inter- and intracellular spatio-temporal dynamics considering diffusion and reaction kinetics (Takahashi *et al.*, 2005), and inherent biological noise (Mettetal *et al.*, 2006).

Simulation models where essential components in the system are appropriately defined and assembled can predict the system's response towards given perturbations. The hypothesis derived from such simulation then becomes valuable for experimental validations. The simulation–experiment cycle through mathematical models can minimize costly or impossible experiments such as sensitivity analysis, impact estimation of unknown factors, assessment of systems threshold, or finding necessary and sufficient minimum components for the focused event.

Approaches in Cell Simulation

Due to the complexity in both cellular systems and the demands for cell simulation, a variety of approaches and methods have been considerable, depending on the feature of the focused system including cell type and phenomena, the accessibility to the system, and the availability of the data (Fig. 2).

Dynamic Modeling and Static Modeling

The first decision branch on cell modeling is the definition of time in the simulation. Dynamic models typified by ordinary differential equations (ODEs) and partial differential equations (PDEs) determine the time development of the cellular system based on the precedent states and response to the environmental changes.

On the other hand, static cell models represent the analysis of the static diagrams of model components, including their attributes and relationships between components. For example, for metabolic systems, topological pathway analysis such as elementary modes (Schuster *et al.*, 2000) and extreme pathways (Schilling *et al.*, 2000) predict essential chain of processes by using stoichiometric relationships and mass conservation law. Metabolic flux analysis is also a stoichiometry-based static model to predict gross error and flux distribution of the metabolic system by introducing flux measurements into the pathway topology (Zupke and Stephanopoulos, 1994; Schmidt *et al.*, 1997). Metabolic control analysis is the static modeling to predict rate or pool controlling steps from the pathway diagram (Fell, 1992). In a broader sense, the static modeling can be expanded to include the generation of comprehensive networks of physical and functional interactions between cellular components by certain network

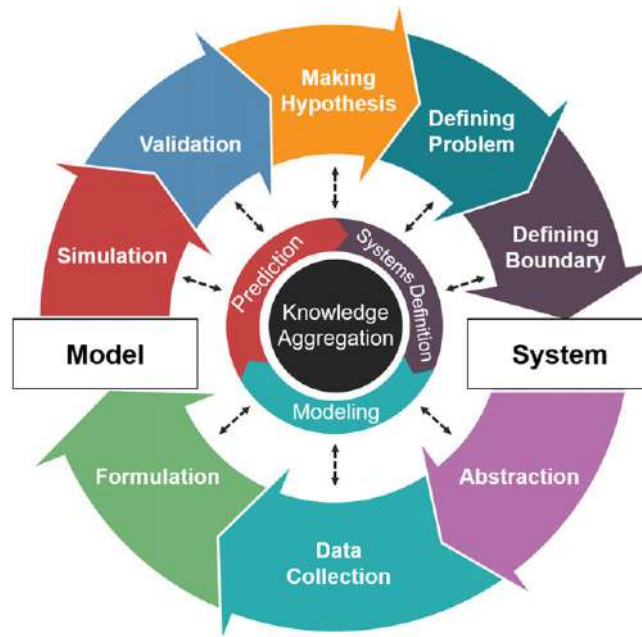


Fig. 1 A roadmap in predictive biology based on cell modeling and simulation. Once we define the cell system, the system is modeled through mathematical formulation based on data. The model is then used to predict the focused cellular phenomena to be validated. The prediction generates, strengthens, or refines the hypothesis on cell system, which drives another cycle of the road map.

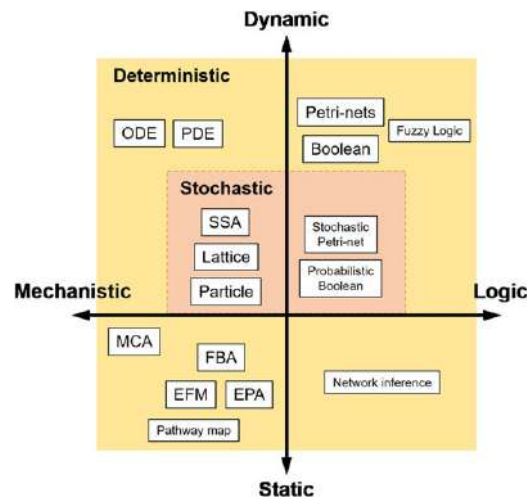


Fig. 2 Simulation approaches of cell systems. Cell modeling is starting from the selection of the suitable mathematical framework from a variety of simulation approaches displayed along several axes. Horizontal axis and vertical axis sort out simulation techniques by mechanistic or logic, dynamic or static, respectively. In addition, dynamic approaches are also split into two groups depending on stochasticity (deterministic or stochastic). Abbreviation used: ODE (ordinary differential equation), PDE (partial differential equation), FBA (flux balance analysis), SSA (stochastic simulation algorithm), EPA (extreme pathway analysis), MCA (metabolic control analysis), EFM (elementary flux mode).

inference approaches (Marbach *et al.*, 2012), or manual curation of the molecular pathway maps (Oda *et al.*, 2005; Oda and Kitano, 2006; Kaizu *et al.*, 2010; Matsuoka *et al.*, 2013).

Deterministic Simulation and Stochastic Simulation

One of the major aspects segregating approaches of cell simulation is the consideration of the stochasticity in the processes. Deterministic behavior can be defined as a consecutive chain of precedent events where no randomness is affecting the next states. In this sense, deterministic cellular models reproduce the same predictions for a given initial condition. This is applicable in case of well-stirred systems, in other words, the systems where cell behavior can be assumed as the behavior of the cell population where the cell belongs.

However, in case the target cell system is assuming small volumes or small numbers of molecules, this definition of deterministic behavior is not applicable (Rao *et al.*, 2002), and stochasticity of the process should be introduced, such as the Gillespie algorithm (Gillespie, 1977). For example, each transcription process is governed by a specific transcription factor whose number is as low as several dozen (Elowitz *et al.*, 2002), and the deterministic modeling can no longer reflect the behavior of cell systems. Moreover, recent emergence of single cell tracking techniques shed light on the stochastic fluctuations of molecules, which exert large impact on significant processes (Gillespie, 2007).

Mechanism-Based Modeling and Logic-Based Modeling

Modeling of the cell system considering biochemical properties for each process, for example, biochemical rules, requires a large amount of mechanistic and quantitative information. This shortage of data on modeling has never been solved despite the rapid spread of high-throughput technologies, as the mechanism-based modeling requires data restricted to influential range of the focused variables while the high-throughput data typically provide more macroscopic view.

Logic-based models are the alternative approaches to overcome the issue around data availability where relationships between variables are described without molecular details, such as Boolean models, fuzzy logic models, and Petri-net models (Kauffman, 1969; Reddy *et al.*, 1996; Hofestadt and Thelen, 1998; Aldridge *et al.*, 2009). Logic-based models assume that one variable respond to the others in a gate (switch-like) fashion (Watterson *et al.*, 2008; Morris *et al.*, 2010; Wynn *et al.*, 2012; Le Novère, 2015). This mathematical simplification is analogous to a steep Hill function with high cooperativity while dose-response features can't be represented. In contrast, it may be feasible to some extent, for example, to apply logic-based modeling to predict the gene expression patterns, with the generally applicable fact that gene expression is a bimodality distributed at the single cell level (Shalek *et al.* 2013).

Simulation Models in Biological Layers

Simulation models are focused on specific phenotype and thus generally focused on specific biological layers. Due to the characteristics of each layer, frequently applied approaches are varied.

Metabolism

The nature of metabolism is the well-regulated mass transfer reaction chains of small molecules called metabolites, whose number is enough large in most cases to assume well-stirred systems. Due to these characteristics of metabolic pathways, dynamic, deterministic and mechanistic approach has been dominantly applied empowered by the massive studies of enzyme kinetics. The case example of modeling of red blood cell metabolism and its application is discussed in the following section.

On the other hand, apart from the dynamic kinetic models, cellular metabolic systems have long played a central role in the tight connectivity between mathematics and wet-lab biological experiments, specifically in the area of systems control theory in metabolic engineering. Indeed, the distinguished scientists in metabolic engineering and their studies took the stage when systems biology emerged and formed as a field (see the subsection "Dynamic Modeling and Static Modeling" in the section "Approaches in Cell Simulation"). Under the mass conservation law and the steady-state assumption, which is relatively fair in metabolism given the robust metabolic homeostasis, it is possible to calculate the flux distribution of the given metabolic system, called flux balance analysis (FBA; Orth *et al.*, 2010). Further assumptions that the cell metabolism is optimized or can be engineered towards the objective cellular functions introduce optimization techniques into FBA to examine the ideal space of metabolic states (i.e., flux distribution patterns), which is called constraint-based modeling (O'Brien *et al.*, 2015; Oberhardt *et al.*, 2009). We now have information from genomes in most model organisms and their annotations with high quality, and thus have a comprehensive topology comprising metabolic reactions at whole-genome scale (Bao and Eddy, 2003). In addition to the bottom-up reconstruction of the model, top-down integration of high-throughput measurements as a further constraint has led the increasingly relevant predictions on cell metabolism (Bordbar *et al.*, 2014). For example, the constraint-models of a genome-scale metabolic pathway have been applied to engineer a strain of *Escherichia coli* producing a valuable chemical, such as L-threonine and 1,4-Butanediol (Lee *et al.*, 2007; Yim *et al.*, 2011), and those of human metabolism have predicted tissue-specific significance of disease gene activity (Shlomi *et al.*, 2008; Jerby *et al.*, 2010).

Signaling

In contrast to metabolic pathways where small molecules and their mass transfer processes are the major variables and processes, the central components of signaling pathways are proteins and the sets of chemical chain reactions based on their states, which is called a signaling cascade. In addition, the key signaling cascades often contain physical information transfer such as nuclear translocations, as the signal propagation should occur in a local and restricted environment. This nature of cellular signaling has called modelers' attention toward spatiotemporal approaches, as described in the section "Spatial Simulation Techniques."

The cellular response to biological signals is significantly varied in contrast to the limited number of known signaling molecules and their cascades. Nevertheless the biological phenomena as a final outcome of the signaling cascade are decoded and often digitized, such as cell fate decision. Due to this discreteness and the dynamic variations of the systems output, the consideration of stochasticity of each process is important in modeling of signaling systems. Remarkably, the models have predicted the significance of stochasticity in small subcellular volume for the propagation of information through certain signaling cascade (Bhalla, 2004a,b, Fujii *et al.*, 2017).

Logic-based modeling approaches have also been applied to simulate signaling cascades (Morris *et al.*, 2010) while the detailed mechanistic and kinetic models of signaling systems have given us significant insights on both fundamental topology and hidden dynamics of the signaling cascade (Kholodenko, 2006). For example, using the mitogen-activated protein kinase signaling model, it has been revealed that the frequently observed signaling cascade arrangement has unexpected consequences for the dynamics that can give an ultrasensitive response to input signal (Huang and Ferrell, 1996). It has been also revealed by the kinetic model that transiently high and sustained concentration of the upstream input signal make different outputs of extracellular-signal-regulated kinase signaling cascade (Sasagawa *et al.*, 2005). This indicated that cell systems encode the distinct dynamics of signals to specific proteins using the signaling cascade to keep the variations of cellular response with minimum set of molecules.

Gene Regulatory Networks

A gene regulatory network (GRN) is a collection of regulatory relationships between transcription factors (TFs) and TF-binding sites of specific mRNA to govern certain expression levels of mRNA and their resulted proteins. Up to 10% of the ORF-coding genomes are coding TFs (Levine and Tjian, 2003) each of which are highly interconnected in the GRN. The GRN itself is often recognized as a static model and frequently described as a large “graph” consisting of hundreds of TFs where nodes represent TFs and edges between nodes represent regulatory relationships between them. The edges in the GRN graph are assessed experimentally by the detection of protein–DNA interactions through chromatin immunoprecipitation techniques (Lee *et al.*, 2002; Ren *et al.*, 2000; Robertson *et al.*, 2007; Johnson *et al.*, 2007; Mikkelsen *et al.*, 2007) or protein–protein interactions (Fields and Song, 1989; Chien *et al.*, 1991). The computational approaches have also been massively developed (Bansal *et al.*, 2007; Ideker and Krogan, 2012; Marbach *et al.*, 2012) to infer GRNs.

Apart from static modeling, dynamic approaches to model GRNs have considerable difficulties including the time delay between transcription and translation, and the shortage of information on the essential topology to represent the focused phenotype and the kinetics. In this context, logic-based models have frequently been applied to simulate and understand the transient states of GRNs (Chai *et al.*, 2014). A recent attempt identified a minimal set of nodes (genes) and their interactions sufficient to explain the experimentally observed cell behavior, by automated reasoning using Boolean models with possible interactions in the GRN in mouse embryonic stem cells (Dunn *et al.*, 2014).

Nevertheless, cell fate transitions occur when cells switch from one GRN to another. Boundaries between these cell states are influenced both by intrinsic fluctuation expression (Eldar and Elowitz, 2010; Singer *et al.*, 2014) and by extrinsic variations in the cellular microenvironment. Moreover, cell–cell heterogeneity of GRN states is commonly observed that can yield stochastic single cell state transitions coalescing into a stable cell population (Glauche *et al.*, 2010; Herberg *et al.*, 2014). To tackle these fundamental properties of GRNs while keeping predictive power, the novel simulation framework has been developed based on asynchronous Boolean simulation followed by graph-theory analysis of the GRN state-transitions (Yachie-Kinoshita *et al.*, 2017). The simulation quantitatively depicts the dynamic changes in cellular GRN states as a function of signaling inputs and their consequential feedback loops within the GRN. The outputs from the simulation provide predictions of the probability of transitions between distinct single cell states, population-average expression frequencies, and the emergence of stable subpopulations in response to different input conditions. The simulation framework was then applied to a relatively large GRN (around 30 nodes in the graph) in mouse embryonic stem cells and five major signaling pathways around pluripotency. As a result, the model predicted a novel combination of signaling inputs that drives mESCs to a specific cell fate.

Application of Predictive Kinetic Model

Red blood cell metabolism and its enzymology have been well studied over the last four decades due to its availability as human cell samples and relative simplicity where the mature cells lack the dynamism from gene expression and the individual cells are physically and functionally compartmentalized via the cell membrane. Although the essential biochemical pathways in red blood cells to maintain the cell state are established earlier and the kinetic information corresponding to each enzymatic reaction in these pathways is available, the quantitative and physiological role of the metabolism is still an open question, because the nature of the cellular function is the complex dynamics of all metabolites.

The ODE-based model of the comprehensive red blood cell metabolism representing detailed enzyme kinetics was then constructed on the E-Cell simulation platform (Takahashi *et al.*, 2004). The model comprises nearly 100 reactions and around 50 metabolites each of which belongs to glycolysis, pentose phosphate pathway, purine salvage pathway, membrane transports, ion leak processes, ATP (adenosine triphosphate)-dependent Na^+/K^+ pump process, or binding reactions between metabolites (Kinoshita *et al.*, 2007a,b).

The initial role of this model was to predict acute hypoxic response of red blood cell metabolism to understand the cellular states in microcirculation. To this end, the hemoglobin allostery in binding and releasing oxygen was modeled in ODE formalism, as well as the binding reactions between key energy metabolites and distinct forms of hemoglobin with and without oxygen.

In the simulation of hypoxia, the oxygen pressure was set from 100 mmHg (arterial level) to 30 mmHg (venous level), then the predicted dynamics in metabolite concentrations were compared with the experimental data from metabolome analysis. The predicted hypoxia response from the model was not in agreement with the experimental data from metabolome analysis; rather no remarkable dynamics was predicted upon state transition of hemoglobins, in spite of that the model successfully reproduce the metabolic states upon other environmental or genetic changes of enzymes.

Given the discrepancy between the prediction and the validation, the model was then refined and expanded to include the other biochemical events by literature curation focusing on connectivity between hemoglobins and metabolism. It is known that both hemoglobin forms with and without oxygen bind to a membrane anion transporter protein called band 3 in distinct association constants. Separately, it is also proved that three enzymes catalyzing intermediate steps of glycolysis also bind to band 3 membrane protein. The model refinement step was carried out by the curation and implementation of these binding processes with distinct kinetic parameters, that is, binding constants.

The refined model considering band 3 membrane protein successfully predicted the measured dynamics. These results from the simulation–experiment cycle indicated the significance of this protein for the metabolic alteration during hypoxia as well as the hypothesized nature of hypoxic response of metabolism where the hemoglobin state transition is the dominant factor of acute changes of red blood cell metabolism upon hypoxia. Moreover, from the analysis of the model, it was predicted that this hypoxic response is necessary to maintain metabolic resources (ATP and 2,3-bisphosphoglycerate (2,3-BPG)) in red blood cells during hypoxia. These hypotheses were confirmed by the biochemical experiment using carbon monoxide-treated hemoglobins, which can mimic the oxygen-containing form of hemoglobin without oxygen molecule.

This precise and predictive model of red blood cell metabolism has further been applied to predict the residual content of metabolic resources in practical long-term storage (Nishino *et al.*, 2009; Nishino *et al.*, 2013). The kinetic parameters of the model are to be altered by temperature, time, and the storage solution. In the modeling studies, these unknown parameters were estimated by real number genetic algorithm to fit with experimental measurements of time course changes in key metabolite concentrations in preserved red blood cells. As a result of the parameter calibration, the model successfully reproduced the dynamics of intermediate metabolites measured in metabolome analysis. The simulation analysis of the model predicted the fates and roles of additive metabolites in the solution for red blood cell storage, which showed somewhat of a trade-off and interlocking relationships between the benefits and possible side effects.

The predictive models and the validation experiments in this way can help us to generate testable hypothesis through the model refinement cycles and give insights to design cellular behaviors (Fig. 1).

Spatial Simulation Techniques

The rapid progress in spatial simulation techniques together with the significant developments in advanced single-molecule measurement such as TIRF (total internal reflection fluorescence) microscopy, and FRET (fluorescence resonance energy transfer) (Liu *et al.*, 2015) enables various applications to study significant cellular phenomena, which are hardly assumed to be spatially uniform.

A conventional way of spatial representation of the cell system is compartmental modeling, where an intracellular space is partitioned into compartments, such as cytoplasm, nucleus, and membrane (Fig. 3(a)). Each compartment is assumed to be well-stirred and the molecules are distributed uniformly in it. In the compartmental modeling framework, the systems dynamics, that is, time-dependent changes of molecular concentrations are either deterministically formulated as ODEs or a chemical master equation for the kinetic reactions, or stochastically formulated as a stochastic simulation algorithm typed by the Gillespie method to be numerically calculated for the chemical system taking into account its intrinsic fluctuation.

On the other hand, the mesh model is widely applied to take account of spatial heterogeneity of molecular distribution in the cell (Fig. 3(b)). In the mesh models, intracellular space is divided into small mesh or subvolumes (0.1–1 μm , in general). Molecules diffuse between neighboring subvolumes, and react in each subvolume. In the deterministic formalization, the chemical reaction–diffusion system is represented as PDEs, and is solved by finite element method, a widely used and common approach in various scientific fields. Various implementations of stochastic algorithms have been proposed for the mesh model with Gillespie method-based stochasticity (see Table 1 for the mesoscopic scope). The mesh models with multiple molecules in each voxel are also known as a mesoscopic lattice model. For example, distinct spatial patterns of synaptic response have been studied by mesh modeling of dendrites and spines (Blackwell, 2013).

Furthermore, along with the improvement in spatio-temporal resolutions of measurements, single molecule-level biophysical approaches have emerged where chemical reactions and diffusion processes of individual molecules are represented rather than the number or concentration of each molecule. As contrasted with mesoscopic lattice method, this type of modeling approach where each molecule occupies a molecular-scale voxel (typically few nanometers) is called the microscopic lattice method (Fig. 3(c)). In the microscopic lattice models, each molecule diffuses across a lattice, and reacts with another molecule at the collision (Arjunan and Tomita, 2010; Arjunan and Takahashi, 2017). The successful models with this type of approach have shown good agreement with experimental observations even with single-particle tracking (SPT) microscopy (Watabe *et al.*, 2015; Lindén *et al.*, 2016).

Yet another modeling approach for spatio-temporal simulation is the particle method based on Brownian dynamics, which treats molecules explicitly as particles, enabling a nanoscale simulation in continuous space (Fig. 3(d)). The Brownian dynamics (BD) method

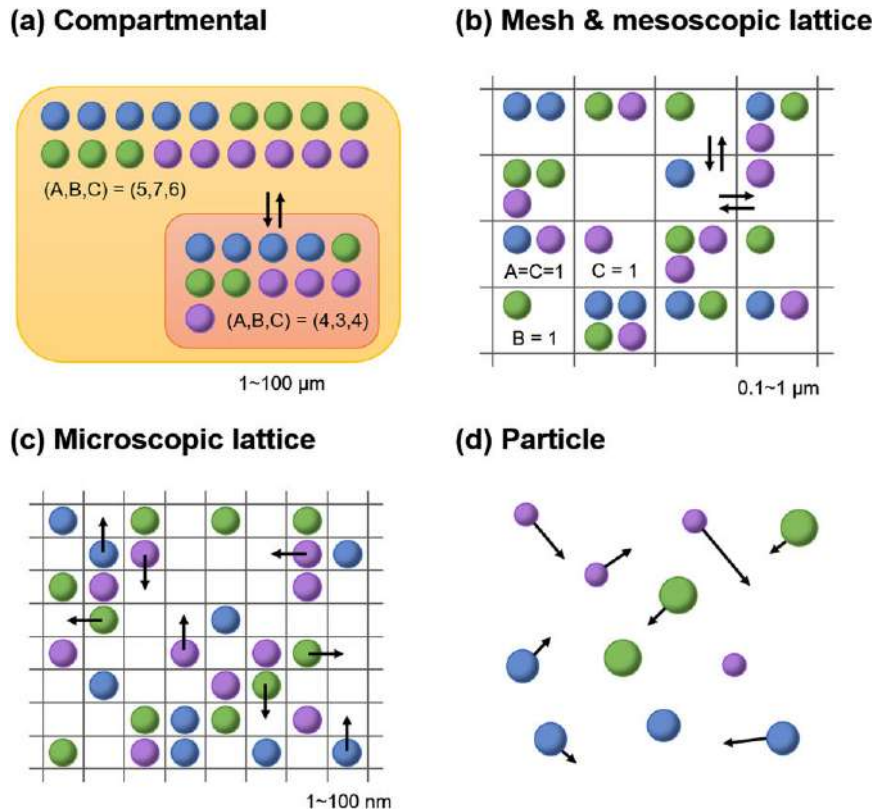


Fig. 3 Spatial representations and simulation approaches. (a) The compartmental model is based on the numbers or concentrations of molecules in well-stirred compartments. The spatial resolution is comparable to the size of cell, nucleus, and organelles (about 1–100 μm). (b) The mesh and mesoscopic lattice models discretize a space into small volumes or areas. Each subvolume can contain multiple molecules in it. Molecules diffuse between and react in subvolumes. The length scale is smaller than a cell but larger than a molecule (0.1–1 μm). (c) The microscopic lattice model consists of small voxels that have the same size with molecules (1–100 nm). A molecule occupies a voxel at the location, and other molecules cannot penetrate there. (d) In the particle model, a molecule is explicitly represented as a particle or a set of particles in continuous space.

generally takes significantly long time to simulate the models and is assumed not applicable to the practical simulations of cell behavior. However, recently developed algorithms with remarkable speed-up have pushed the field towards cellular scale simulation with particle model in several minutes (Schöneberg *et al.*, 2014). As one of the representative particle simulations, Green's function reaction dynamics (GFRD) has accomplished fast and exact simulations of the many-body problem by decomposing it into one- and two-body problems while applying the analytical solution in an event-driven manner (Van Zon and Ten Wolde, 2005; Opplestrup *et al.*, 2006). In the context of the particle simulations at the single-molecule level, various representations of molecules can be applied (Schöneberg *et al.*, 2014). The particle model describes the molecule as a point particle (Andrews, 2017; Plimpton and Slepoy, 2005; Kerr *et al.*, 2008), as a sphere (Takahashi *et al.*, 2010; Klann *et al.*, 2011), or as a collection of spheres resembling molecular structure (Schöneberg and Noé, 2013; Michalski and Loew, 2016; Vijaykumar *et al.*, 2017). For example, GFRD-based particle modeling of ligand-receptor reactions predicted sensing efficiency in bacterial chemotaxis against biological noise (Kaizu *et al.*, 2014).

Tools for Cell Modeling and Simulation

Here we provide an up-to-date list of software and tools for cellular modeling and simulation (Table 1). Due to increasing need for integration of multiple models and approaches, extensibility has become a key factor in tool selection. In this context, the license information for each software package or tool has also been provided in the comprehensive list.

Discussion

Given the emergence of high-throughput and high-resolution experimental techniques as well as the demand for predictive science in cell biology, "whole-cell simulation" is now set as an ultimate but intended goal of systems biology (Tomita *et al.*, 1999; Karr *et al.*, 2012). Whole-cell modeling is a set of challenges in various aspects including modeling approaches, software connectivity, and big data handling (Karr *et al.*, 2015; Goldberg *et al.*, 2018).

Table 1 An up-to-date list of available tools and software for cell modeling and simulation

<i>Name</i>	<i>License</i>	<i>Year</i>	<i>Usability</i>	<i>Approach</i>	<i>Access</i>
SBW	BSD License	2002	Built-in?	ODE	http://sbw.sourceforge.net/
Cell Illustrator	Proprietary?	2004	Standalone/GUI	ODE	http://www.cellillustrator.com/home
DBSolveOptimum	Proprietary?	2010	Standalone/GUI	ODE	http://insysbio.com/en/software/db-solve-optimum
JWS Online		2002	Web-based/GUI	ODE	http://jjj.biochem.sun.ac.za/
SloppyCell	BSD License	2007	Library (Python)	ODE	http://sloppycell.sourceforge.net/
PySCeS	Creative Commons Zero v1.0 Universal	2004	Python	ODE	http://pysces.sourceforge.net/
JSim (physiome)	BSD-like License	2013	Standalone/GUI/Web-based (Java)	ODE, PDE	http://www.physiome.org/jsim/
MATLAB (Mathworks Simulink)	Proprietary		Standalone/GUI/Scriptable	ODE, PDE	https://www.mathworks.com/products/matlab.html
PhysioDesigner/FLINT	MIT License	2012	Standalone/GUI	ODE, PDE	http://www.physiodesigner.org/
MOOSE	GPL v3	2007	Standalone/GUI/Python-scriptable	ODE, PDE	https://moose.ncbs.res.in/
CADLIVE	Proprietary?	2003	Standalone/GUI	ODE, S-system	http://www.cadlive.jp/
E-Cell3	GPL v2	1996	Standalone/GUI/Python-Scriptable	ODE, SSA	http://www.e-cell.org/
COPASI (Gepasi)	Artistic License 2.0	1993	Standalone/GUI/Scriptable?	ODE, SSA	http://copasi.org/
CellDesigner (with COPASI, SOSlib)	Proprietary ?	2003	Standalone/GUI	ODE, SSA	http://www.celldesigner.org/
PySB	BSD 2-clause License	2008	Python-scriptable	ODE, SSA	http://pysb.org/
BioNetGen (RuleBender, BioLab, NFSim)		2004	Standalone/GUI	ODE, SSA	http://www.csb.pitt.edu/Faculty/Faeder/?page_id=409
KaSim (Kappa, KaSa)	LGPL v3	2003	Standalone	ODE, SSA	http://dev.executableknowledge.org/
tellurium (libroadrunner, antimony, phrasedml, libsbml)	Apache 2.0	2014	Standalone/GUI/Library (Python)	ODE, SSA	http://tellurium.analogmachine.org/
BIOCHAM	GPL v2	2003	Web-based/Standalone(Java)/GUI	ODE, SSA, Boolean	https://lifeware.inria.fr/biocham/
BioUML	Open-Source?	2004	Standalone/GUI	ODE, SSA, FBA	http://wiki.biouml.org/index.php/Landing
iBioSim	Apache 2.0	2009	Standalone/GUI	ODE, SSA, FBA	www.async.ece.utah.edu/ibiosim
E-Cell4	GPL v2	2005	Python-Scriptable	ODE, SSA, Meso/ Microscopic, Particle	http://www.e-cell.org/ecell4/
The Virtual Cell	MIT License	1997	Web-based/Standalone (Java)	PDE	http://vcell.org/
FreeFEM	LGPL v2	1987	Scriptable? (own language)	PDE	http://www.freefem.org/
openCOBRA	LGPL/GPL v2	2007	Scriptable (MATLAB, Python Julia)	FBA	https://opencobra.github.io/
BetaWB (BlenX)	Proprietary?	2008	Standalone	SSA	https://sites.google.com/site/aromanel/software/bwb
Bio-PEPA (Bio-PEPA Workbench)	GPL v2	2008	Plugin(Eclipse)/Standalone	SSA	http://homepages.inf.ed.ac.uk/stg/software/biopepa/bpwb.html
SPiM	Microsoft License Agreement (Open-Source, but noncommercial use)	2004	Standalone/GUI	SSA	https://www.microsoft.com/en-us/research/project/stochastic-pi-machine/
BoolNet	Artistic License 2.0	2009	Library (R-lang)	Boolean	http://sysbio.uni-ulm.de/?Software: BoolNet
BooleanNet	MIT License	2007	Python-scriptable	Boolean	https://pypi.python.org/pypi/BooleanNet/1.2.8
ViSiBooL	Open-source?	2017	Standalone (Java)	Boolean	http://sysbio.uni-ulm.de/?Software: ViSiBooL
GINSim	GPL v3	2006	Standalone/GUI (Java)	Boolean	http://ginsim.org/
ePNK	GPL	2012	Eclipse Plugin	Petri nets	http://www.imm.dtu.dk/~ekki/projects/ePNK/index.shtml
WoPeD	LGPL v3	2003	Standalone	Petri nets	http://woped.dhbw-karlsruhe.de/woped/

Table 1 Continued

<i>Name</i>	<i>License</i>	<i>Year</i>	<i>Usability</i>	<i>Approach</i>	<i>Access</i>
MONALISA	Artistic License 2.0	2013	Standalone/GUI (Java)	Petri nets	http://www.bioinformatik.uni-frankfurt.de/tools/monalisa/
GreatSPN	Proprietary, but free for educational use?	1995	Standalone/GUI (Java)	Petri nets	http://www.di.unito.it/~greatspn/index.html
Snoopy	Proprietary, but free for non-commercial use?	2003	Standalone/GUI	Petri nets	http://www.dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy
MCell (CellBlender)	GPL v2	1996	Standalone/GUI	Agent-based	http://mcell.org/
FLAME	GNU LGPL?	2006	Standalone	Agent-based	http://flame.ac.uk/
REPASt	BSD-like License	2004	Standalone (GUI)	Agent-based	https://repast.github.io/
URDME (StochSS)	GPL v3	2008	Scriptable (Python/MATLAB)	Mesosopic	http://www.stochss.org/
MesoRD	GPL v2	2005	Standalone/GUI	Mesosopic	http://mesord.sourceforge.net/
STEPS	GPL v2	2009	Python-scriptable	Mesosopic	http://steps.sourceforge.net/STEPS/default.php
Lattice Microbes (pyLM)	University of Illinois Open Source License	2013	Python-scriptable	Mesosopic	http://www.scs.illinois.edu/schulten/lm/
Simmune	Proprietary	2006	Standalone/GUI	Mesosopic	https://www.niaid.nih.gov/research/simmune-project
CompuCell3D	Open-Source?	2012	Standalone/ Python-Scriptable/GUI	Mesosopic	http://www.compuCell3d.org/
Spatocyte	GPL v2	2009	Standalone/GUI/ Python-scriptable	Microscopic	http://spatocyte.org/
GridCell	Proprietary, but free for academic use	2007	Standalone/GUI/ Web-based	Microscopic	http://www.isip.ece.mcgill.ca/research/gridcell/start
Smoldyn	Mostly LGPL	2003	Standalone/GUI	Particle	http://www.smoldyn.org/
GFRD	GPL v2	2005	Standalone	Particle	http://gfrd.org/
ReaDDy	BSD/3-clause	2011	Python-scriptable	Particle	http://www.readdy-project.org/
ChemCell (Pizza.py)	GPL v2	2005	Standalone/Python-scriptable	Particle	http://chemcell.sandia.gov/
SpringSalad	Open-Source?	2016	Standalone	Particle	http://vcell.org/springsalad-2
CDS	Open-Source?	2010	Standalone/GUI	Particle	https://med.uth.edu/nba/cds/
SRSim	GPL v2	2010	Standalone	Particle	http://www.biosys.uni-jena.de/Members/Gerd+Gruenert/SRSim.html
dReal (dReach)	GPL v3	2012	Standalone	Model Checking	http://dreal.github.io/dReach/
Pathway Logic	GPL v2	2006	Standalone/GUI	Model Checking	http://pl.csl.sri.com/
PRISM	GPL v2	2000	Standalone	Model Checking	http://www.prismmodelchecker.org/
Neuron	The 3-Clause BSD License?	1989	GUI/Python-Scriptable	Nerve Equations	https://www.neuron.yale.edu/neuron/

For example, as discussed in the section of “Simulation Models in Biological Layers,” each biological layer has distinct scales of time, space, and quantity, all of which are orchestrated into one cell system. The integration of distinct biological scales so far cannot be done with a single simulation algorithm. In this context, for example, the E-Cell System originally launched in 1996 (Tomita *et al.*, 1997; Tomita *et al.*, 1999) endeavors to model and simulate the distinct spatio-temporal scales at the same time by developing a mathematical framework to combine multi-algorithms and multi-timescales (Takahashi *et al.*, 2004; The E-Cell System version 4).

As in the up-to-date list in Table 1, we now have a variety of tools and software, each of which has different focus on applicable biological layers, simulation algorithms, and usabilities. Under these circumstances, the need for unified platform to support multi-tools has become more apparent. From the “soft power” point of view, a standard format of simulation model would increase exchangeability among simulation tools and reproducibility and reusability of the models are secured. The Systems Biology Markup Language (SBML) has been developed and now popularized in the scientific community (Hucka *et al.*, 2003; Hucka *et al.*, 2015; Waltemath *et al.*, 2016). The movement of packaging software for cell modeling and simulation has also come out such as BioNetGen (Blinov *et al.*, 2004) and PySB for Python-based programmatic model construction (Lopez *et al.*, 2013).

The scalability of the simulation models towards whole-cell simulation is largely dependent on (generally experimental) data availability and its accessibility. The complete computational workflow including data collection, data mining, network inference, modeling, simulation, model analysis, and model validation has to be executed to complete the systems biology roadmap and that should be connected with the “big pond” of the experimental data. The complete workflow to integrate whole available knowledge from independent small sciences needs the novel computational framework and connectivity platform (Kitano *et al.*, 2011). Recent advances in the development of such platforms include Garuda (Ghosh *et al.*, 2011), Galaxy (Goecks *et al.*, 2010), GenomeSpace (Qu *et al.*, 2016), and Taverna (Oinn *et al.*, 2004).

All these bottom-up and top-down efforts contributing to whole-cell modeling have opened up opportunities for the systems-level understanding of cells to assess the mechanism of how cell behavior can advance the evolvability of life.

See also: Biological and Medical Ontologies: GO and GOA. Computational Approaches for Modeling Signal Transduction Networks. Computational Systems Biology Applications. Computational Systems Biology. Coupling Cell Division to Metabolic Pathways Through Transcription. Data Formats for Systems Biology and Quantitative Modeling. Gene Regulatory Network Review. Graphlets and Motifs in Biological Networks. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics. Molecular Dynamics and Simulation. Multi-Scale Modelling in Biology. Natural Language Processing Approaches in Bioinformatics. Network Inference and Reconstruction in Bioinformatics. Pathway Informatics. Prediction of Protein Localization. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein-Protein Interactions: An Overview. Quantitative Modelling Approaches. Visualization of Biomedical Networks

References

- Aldridge, B.B., Saez-Rodriguez, J., Muhlich, J.L., Sorger, P.K., Lauffenburger, D.A., 2009. Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/insulin-induced signaling. *PLOS Computational Biology* 5, e1000340.
- Andrews, S.S., 2017. Smoldyn: Particle-based simulation with rule-based modeling, improved molecular interaction and a library interface. *Bioinformatics* 33, 710–717.
- Arjunan, S.N.V. & Takahashi, K., 2017. Multi-algorithm particle simulations with spatioocyte. In: *Methods in Molecular Biology*, vol. 1611, New York, NY: Humana Press, pp. 219–236.
- Arjunan, S.N.V., Tomita, M., 2010. A new multicompartmental reaction-diffusion modeling method links transient membrane attachment of *E. coli* MinE to E-ring formation. *Systems and Synthetic Biology* 4, 35–53.
- Bansal, M., Belcastro, V., Ambesi-Impombato, A., Di Bernardo, D., 2007. How to infer gene networks from expression profiles. *Molecular Systems Biology* 3, 78.
- Bao, Z., Eddy, S.R., 2003. Automated de novo identification of repeat sequence families in sequenced genomes. *Genome Research* 13, 1269–1276.
- Bhalla, U.S., 2004a. Signaling in small subcellular volumes. I. Stochastic and diffusion effects on individual pathways. *Biophysical Journal* 87, 733–744.
- Bhalla, U.S., 2004b. Signaling in small subcellular volumes. II. Stochastic and diffusion effects on synaptic network properties. *Biophysical Journal* 87, 745–753.
- Blackwell, K.T., 2013. Approaches and tools for modeling signaling pathways and calcium dynamics in neurons. *Journal of Neuroscience Methods* 220, 131–140.
- Blinov, M.L., Faeder, J.R., Goldstein, B., Hlavacek, W.S., 2004. BioNetGen: Software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics* 20, 3289–3291.
- Bordbar, A., Monk, J.M., King, Z.A., Palsson, B.O., 2014. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics* 15, 107–120.
- Chai, L.E., *et al.*, 2014. A review on the computational approaches for gene regulatory network construction. *Computers in Biology and Medicine* 48, 55–65.
- Chien, C.T., Bartel, P.L., Sternglanz, R., Fields, S., 1991. The two-hybrid system: A method to identify and clone genes for proteins that interact with a protein of interest. *Proceedings of the National Academy of Sciences of the United States of America* 88, 9578–9582.
- Dunn, S.J., Martello, G., Yordanov, B., Emmott, S., Smith, A.G., 2014. Defining an essential transcription factor program for naïve pluripotency. *Science* (80-) 344, 1156–1160.
- Eldar, A., Elowitz, M.B., 2010. Functional roles for noise in genetic circuits. *Nature* 467, 167–173.
- Elowitz, M.B., Levine, A.J., Siggia, E.D., Swain, P.S., 2002. Stochastic gene expression in a single cell. *Science* (80-) 297, 1183–1186.
- Fell, D.A., 1992. Metabolic control analysis: A survey of its theoretical and experimental development. *Biochemical Journal* 286, 313–330.
- Fields, S., Song, O., 1989. A novel genetic system to detect protein–protein interactions. *Nature* 340, 245–246.
- Fujii, M., Ohashi, K., Karasawa, Y., Hikichi, M., Kuroda, S., 2017. Small-volume effect enables robust, sensitive, and efficient information transfer in the spine. *Biophysical Journal* 112, 813–826.
- Garfinkel, D., Garfinkel, L., Pring, M., Green, S.B., Chance, B., 1970. Computer applications to biochemical kinetics. *Annual Review of Biochemistry* 39, 473–498.
- Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., Kitano, H., 2011. Software for systems biology: From tools to integrated platforms. *Nature Reviews Genetics* 12, 821–832.
- Gillespie, D.T., 2007. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry* 58, 35–55.
- Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* 81, 2340–2361.
- Glauche, I., Herberg, M., Roeder, I., 2010. Nanog variability and pluripotency regulation of embryonic stem cells – Insights from a mathematical model analysis. *PLOS ONE* 5, e11238.
- Goecks, J., *et al.*, 2010. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology* 11, R86.
- Goldberg, A.P., *et al.*, 2018. Emerging whole-cell modeling principles and methods. *Current Opinion in Biotechnology* 51, 97–102.
- Heinrich, R., Rapoport, S.M., Rapoport, T.A., 1978. Metabolic regulation and mathematical models. *Progress in Biophysics & Molecular Biology* 32, 1–82.
- Herberg, M., Kalkan, T., Glauche, I., Smith, A., Roeder, I., 2014. A model-based analysis of culture-dependent phenotypes of mESCs. *PLOS ONE* 9, e92496.
- Hey T., Tansley S., and Tolle K., 2009, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, ISBN 978-0-9825442-0-4.
- Hodgkin, A.L., Huxley, A.F., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117, 500–544. (<https://doi.org/10.1113/jphysiol.1952.sp004764>).
- Hofstadter, R., Thelen, S., 1998. Quantitative modeling of biochemical networks. *In Silico Biology* 1, 39–53.
- Huang, C.Y., Ferrell, J.E., 1996. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proceedings of the National Academy of Sciences of the United States of America* 93, 10078–10083.
- Hucka, M., *et al.*, 2003. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.
- Hucka, M., *et al.*, 2015. Promoting coordinated development of community-based information standards for modeling in biology: The COMBINE initiative. *Frontiers in Bioengineering and Biotechnology* 3, 19.
- Ideker, T., Krogan, N.J., 2012. Differential network biology. *Molecular Systems Biology* 8, 565.
- Iwamoto, K., Shindo, Y., Takahashi, K., 2016. Modeling cellular noise underlying heterogeneous cell responses in the epidermal growth factor signaling pathway. *PLOS Computational Biology* 12, e1005222.
- Jerby, L., Shlomi, T., Ruppin, E., 2010. Computational reconstruction of tissue-specific metabolic models: Application to human liver metabolism. *Molecular Systems Biology* 6, 401.
- Johnson, D.S., Mortazavi, A., Myers, R.M., Wold, B., 2007. Genome-wide mapping of in vivo protein–DNA interactions. *Science* (80-) 316, 1497–1502.
- Kaizu, K., *et al.*, 2014. The berg–purcell limit revisited. *Biophysical Journal* 106, 976–985.
- Kaizu, K., *et al.*, 2010. A comprehensive molecular interaction map of the budding yeast cell cycle. *Molecular Systems Biology* 6, 415.
- Kalisky, T., Blainey, P., Quake, S.R., 2011. Genomic analysis at the single-cell level. *Annual Review of Genetics* 45, 431–445.
- Karr, J.R., *et al.*, 2012. A whole-cell computational model predicts phenotype from genotype. *Cell* 150, 389–401.
- Karr, J.R., Takahashi, K., Funahashi, A., 2015. The principles of whole-cell modeling. *Current Opinion in Microbiology* 27, 18–24.
- Kauffman, S., 1969. Homeostasis and differentiation in random genetic control networks. *Nature* 224, 177–178.
- Kerr, R.A., *et al.*, 2008. Fast Monte Carlo simulation methods for biological reaction–diffusion systems in solution and on surfaces. *SIAM Journal on Scientific Computing* 30, 3126–3149.
- Kholodenko, B.N., 2006. Cell-signalling dynamics in time and space. *Nature Reviews Molecular Cell Biology* 7, 165–176.

- Kinoshita, A., *et al.*, 2007a. Roles of hemoglobin allosterism in hypoxia-induced metabolic alterations in erythrocytes: Simulation and its verification by metabolome analysis. *Journal of Biological Chemistry* 282, 10731–10741.
- Kinoshita, A., Nakayama, Y., Kitayama, T., Tomita, M., 2007b. Simulation study of methemoglobin reduction in erythrocytes: Differential contributions of two pathways to tolerance to oxidative stress. *The FEBS Journal* 274, 1449–1458.
- Kitano, H., 2002. Systems biology: A brief overview. *Science* (80-) 295, 1662–1664.
- Kitano, H., Ghosh, S., Matsuoka, Y., 2011. Social engineering for virtual 'big science' in systems biology. *Nature Chemical Biology* 7, 323–326.
- Klann, M.T., Lapin, A., Reuss, M., 2011. Agent-based simulation of reactions in the crowded and structured intracellular environment: Influence of mobility and location of the reactants. *BMC Systems Biology* 5, 71.
- Lee, K.H., Park, J.H., Kim, T.Y., Kim, H.U., Lee, S.Y., 2007. Systems metabolic engineering of *Escherichia coli* for L-threonine production. *Molecular Systems Biology* 3, 149.
- Lee, T.I., *et al.*, 2002. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* (80-) 298, 799–804.
- Levine, M., Tjian, R., 2003. Transcription regulation and animal diversity. *Nature* 424, 147–151.
- Lindén, M., Čurić, V., Boucharin, A., Fange, D., Elf, J., 2016. Simulated single molecule microscopy with SMEagol. *Bioinformatics* 32, 2394–2395.
- Liu, Z., Lavis, L.D., Betzig, E., 2015. Imaging live-cell dynamics and structure at the single-molecule level. *Molecular Cell* 58, 644.
- Lopez, C.F., Mühlich, J.L., Bachman, J.A., Sorger, P.K., 2013. Programming biological models in Python using PySB. *Molecular Systems Biology* 9, 646.
- Marbach, D., *et al.*, 2012. Wisdom of crowds for robust gene network inference. *Nature Methods* 9, 796–804.
- Matsuoka, Y., *et al.*, 2013. A comprehensive map of the influenza A virus replication cycle. *BMC Systems Biology* 7, 97.
- Mettetal, J.T., Muzzey, D., Pedraza, J.M., Ozbudak, E.M., van Oudenaarden, A., 2006. Predicting stochastic gene expression dynamics in single cells. *Proceedings of the National Academy of Sciences of the United States of America* 103, 7304–7309.
- Michalski, P.J., Loew, L.M., 2016. SpringSaLaD: A spatial, particle-based biochemical simulation platform with excluded volume. *Biophysical Journal* 110, 523–529.
- Mikkelsen, S., *et al.*, 2007. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature* 448, 553–560.
- Morris, M.K., Saez-Rodriguez, J., Sorger, P.K., Lauffenburger, D.A., 2010. Logic-based models for the analysis of cell signaling networks. *Biochemistry* 49, 3216–3224.
- Nishino, T., *et al.*, 2009. In silico modeling and metabolome analysis of long-stored erythrocytes to improve blood storage methods. *Journal of Biotechnology* 144, 212–223.
- Nishino, T., *et al.*, 2013. Dynamic simulation and metabolome analysis of long-term erythrocyte storage in adenine–guanosine solution. *PLOS ONE* 8, e71060.
- Le Novère, N., 2015. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics* 16, 146–158.
- Oberhardt, M.A., Palsson, B., Papin, J.A., 2009. Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology* 5, 320.
- Oda, K., Kitano, H., 2006. A comprehensive map of the toll-like receptor signaling network. *Molecular Systems Biology* 2, 2006.0015.
- Oda, K., Matsuoka, Y., Funahashi, A., Kitano, H., 2005. A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology* 1, E1–E17.
- Oinn, T., *et al.*, 2004. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20. pp. 3045–3054.
- Oppelstrup, T., Bulatov, V.V., Gilmer, G.H., Kalos, M.H., Sadigh, B., 2006. First-passage Monte Carlo algorithm: Diffusion without all the hops. *Physical Review Letters* 97, 230602.
- Orth, J.D., Thiele, I., Palsson, B.O., 2010. What is flux balance analysis? *Nature Biotechnology* 28, 245–248.
- O'Brien, E.J., Monk, J.M., Palsson, B.O., 2015. Using genome-scale models to predict biological capabilities. *Cell* 161, 971–987.
- Plimpton, S.J., Slepoy, A., 2005. Microbial cell modeling via reacting diffusive particles. *Journal of Physics: Conference Series* 16, 305–309.
- Qu, K., *et al.*, 2016. Integrative genomic analysis by interoperation of bioinformatics tools in GenomeSpace. *Nature Methods* 13, 245–247.
- Rao, C.V., Wolf, D.M., Arkin, A.P., 2002. Control, exploitation and tolerance of intracellular noise. *Nature* 420, 231–237.
- Reddy, V.N., Liebman, M.N., Mavrouniotis, M.L., 1996. Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine* 26, 9–24.
- Ren, B., *et al.*, 2000. Genome-wide location and function of DNA binding proteins. *Science* (80-) 290, 2306–2309.
- Robertson, G., *et al.*, 2007. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods* 4, 651–657.
- Sasagawa, S., Ozaki, Y.I., Fujita, K., Kuroda, S., 2005. Prediction and validation of the distinct dynamics of transient and sustained ERK activation. *Nature Cell Biology* 7, 365–373.
- Schilling, C.H., Letscher, D., Palsson, B.O., 2000. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology* 203, 229–248.
- Schmidt, K., Carlsen, M., Nielsen, J., Villadsen, J., 1997. Modeling isotopomer distributions in biochemical networks using isotopomer mapping matrices. *Biotechnology and Bioengineering* 55, 831–840.
- Schöneberg, J., Noé, F., 2013. ReadDy – A software for particle-based reaction–diffusion dynamics in crowded cellular environments. *PLOS ONE* 8, e74261.
- Schöneberg, J., Ullrich, A., Noé, F., 2014. Simulation tools for particle-based reaction–diffusion dynamics in continuous space. *BMC Biophysics* 7, 11.
- Schuster, S., Fell, D.A., Dandekar, T., 2000. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology* 18, 326–332.
- Shalek, A.K., *et al.*, 2013. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature* 498, 236–240.
- Shlomi, T., Cabili, M.N., Herrgård, M.J., Palsson, B.O., Rupp, E., 2008. Network-based prediction of human tissue-specific metabolism. *Nature Biotechnology* 26, 1003–1010.
- Singer, Z.S., *et al.*, 2014. Dynamic heterogeneity and DNA methylation in embryonic stem cells. *Molecular Cell* 55, 319–331.
- Takahashi, K., Kaizu, K., Hu, B., Tomita, M., 2004. A multi-algorithm, multi-timescale method for cell simulation. *Bioinformatics* 20, 538–546.
- Takahashi, K., Tanase-Nicola, S., ten Wolde, P.R., 2010. Spatio-temporal correlations can drastically change the response of a MAPK pathway. *Proceedings of the National Academy of Sciences of the United States of America* 107, 2473–2478.
- Takahashi, K., Vel Arjunan, S.N., Tomita, M., 2005. Space in systems biology of signaling pathways – Towards intracellular molecular crowding in silico. *FEBS Letters* 579, 1783–1788.
- The E-Cell System version 4: An integrated environment for modeling, simulation and analysis of cell systems with multi-algorithms, multi-timescales and multi-spatial-representations. doi:10.5281/zenodo.1119017.
- Tomita, M., *et al.*, 1997. E-cell: Software environment for whole cell simulation. *Genome Informatics. Workshop on Genome Informatics* 8, 147–155.
- Tomita, M., *et al.*, 1999. E-cell: Software environment for whole-cell simulation. *Bioinformatics* 15, 72–84.
- Tyson, J., Othmer, H.G., 1978. The dynamics of feedback control circuits in biochemical pathways. *Progress in Theoretical Biology* 5, 2–62.
- Vijaykumar, A., Ouldrige, T.E., Ten Wolde, P.R., Bolhuis, P.G., 2017. Multiscale simulations of anisotropic particles combining molecular dynamics and Green's function reaction dynamics. *Journal of Chemical Physics* 146, 114106.
- Waltemath, D., *et al.*, 2016. Toward community standards and software for whole-cell modeling. *IEEE Transactions on Biomedical Engineering* 63, 2007–2014.
- Watabe, M., *et al.*, 2015. A computational framework for bioimaging simulation. *PLOS ONE* 10, e0130089.
- Watterson, S., Marshall, S., Ghazal, P., 2008. Logic models of pathway biology. *Drug Discovery Today* 13, 447–456.
- Wynn, M.L., Consul, N., Merajver, S.D., Schnell, S., 2012. Logic-based models in systems biology: A predictive and parameter-free network analysis method. *Integrative Biology* 4, 1323.
- Xia, T., Li, N., Fang, X., 2013. Single-molecule fluorescence imaging in living cells. *Annual Review of Physical Chemistry* 64, 459–480.
- Yachie-Kinoshita, A., *et al.*, 2017. Modeling signaling-dependent pluripotent cell states with Boolean logic can predict cell fate transitions. *bioRxiv* 14, e7952.
- Yim, H., *et al.*, 2011. Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. *Nature Chemical Biology* 7, 445–452.
- Van Zon, J.S., Ten Wolde, P.R., 2005. Simulating biochemical networks at the particle level and in time and space: Green's function reaction dynamics. *Physical Review Letters* 94, 128103.
- Zupke, C., Stephanopoulos, G., 1994. Modeling of isotope distributions and intracellular fluxes in metabolic networks using atom mapping matrices. *Biotechnology Progress* 10, 489–498.

Quantitative Modelling Approaches

Filippo Castiglione, National Research Council of Italy, Rome, Italy

Emiliano Mancini, University of Amsterdam, Amsterdam, Netherlands

Marco Pedicini, Roma Tre University, Rome, Italy

Abdul Salam Jarrah, American University of Sharjah, Sharjah, UAE

© 2019 Elsevier Inc. All rights reserved.

Glossary

Complexity science A field of study aiming at defining methods and tools to assess and predict the dynamics of a complex system using a holistic approach.

Complex system A hierarchical self-organized system with a large number of components characterized by non-linear connections. In such systems order is an emergent property of its internal microscopic dynamics.

Computation All steps necessary for a digital computer to solve a problem. A computation transforms the input, i.e., the definition of a problem in mathematical terms, into outputs, i.e., the solution of the problem in terms of values of some of the problem's variables.

Dynamics Area of classical mechanics concerned with the study of forces and torques and their effect on motion. More

in general, the dynamics of a system describes the trajectory of the constituting variables in a proper mathematical space.

Mathematical modelling Description of a natural systems by means of the mathematical language that is then suitable for transformations and other mathematical treatments to find non-trivial relations among system elements.

Networks Graphical description of the relations among entities that is then viable to mathematical treatments such as those of graph theory.

Simulation computer recreation of the dynamics of natural or artificial systems.

Stochasticity The presence of something that is not deterministically determined. In mathematics, it is used to indicate the use of a random element in the description of a system.

Introduction

The recent wide availability of experimental high-throughput omics data (genomics, proteomics, metabolomics, etc.) is fostering the development of theoretical biology. This data is analysed by established bioinformatics tools that have been developed in the past decades as well as by novel approaches devised at a great pace by the community. However, the results of such analysis are in some cases inconclusive and further investigation is necessary to make sense of it. This is when mathematical, statistical and computational approaches such as modelling and simulation come into play. This article briefly describes few but well-representative methods that have been used to this purpose in the last couple of decades.

Quantitative modelling of biological phenomena comprise all methods borrowed by other quantitative or *exact* disciplines such as physics, mathematics, computer science or engineering, that are nowadays used to tame the biological complexity. Given the enormous diversity found in biological systems, it is in general difficult to obtain quantitative estimation of parameters of interest and often the results are dependent on the modelling assumptions. Therefore, the methods range between the two extremes of being completely qualitative to perfectly quantitative. Qualitative methods provide an approximate explanation of what is going on in the studied system without being able to estimate values precisely or give clear indications about how to influence the system (Saadatpour and Albert, 2016). On the other hand quantitative methods provide answers to specific questions with numbers, that is, with something that is readily useful in real practice without the need of further (complex) transformations or assumptions (Mogilner *et al.*, 2006).

The usefulness of computational models in systems biology has been pointed out in many recent literature review articles (Bartocci and Lió, 2016; Fisher and Henzinger, 2007; Ji *et al.*, 2017; Machado *et al.*, 2011; Materi and Wishart, 2007). This article, however, aims at providing an informal and not exhaustive introduction to the main computational modelling approaches adopted in systems biology. Besides the classical differential equation approach, this paper presents other computational dynamic approaches such as Petri nets, Boolean networks, cellular automata, agent based models, and the statistical methods of Bayesian networks and formal systems such as process algebras.

Background/Fundamentals

Mathematical modelling has a long tradition and is central to most disciplines of science and engineering. It characterizes the scientific effort in describing nature in quantitative terms. Newton's work which was published in the 17th century, is probably the most striking example of application of the concepts of calculus to describe a physical system (i.e., the motion of planets in terms of differential equations). A model is, today as then, a mathematical description of the elements of a process and the ways in which they influence each other. The mathematical description, sometimes amenable of symbolic transformations, leads to a formulation of the "solution" which reveals aspects of the structure and dynamics of the system that are not plainly recognisable at a first sight. Mathematical (now computational alike) modelling is an iterative process which goes from the real-life phenomenon to the virtual mathematical object called the "model" that produces "results" in terms of how it behaves in hypothetical scenarios.

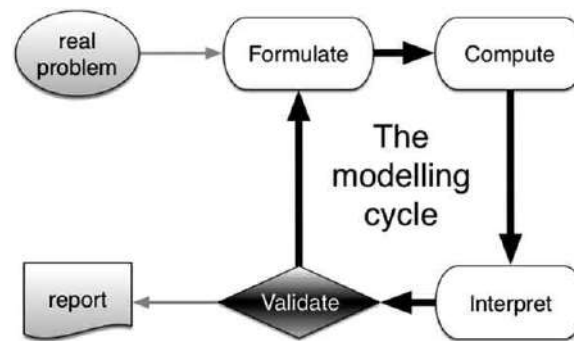


Fig. 1 Mathematical or computational modelling is an iterative process which involves the translation of the most significant characteristics of the real problem into a mathematical formalism amenable of computation (or analytical treatment) to produce results that are then interpreted to be translated back to reality. Those need then to be compared to the real system to be validated so to show the correctness and even usefulness of the model. If this is not the case, the model is improved and the whole cycle is repeated.

To be scientifically sound, these results should be compared to actual data that were produced by the real-life phenomenon and were not used to build or infer the model yet. This comparison, which is called model validation, usually identifies shortcomings or problems in the model, advocating some model adjustment, hence the iterative scheme described in Fig. 1.

This practice is not different when the system under consideration comprises life. On one side, we have a biological system we wish to understand and on the other side we have a model corresponding to our understanding of the real system that allows us to perform analysis and testing of hypothetical scenarios. This is for instance the case of a model describing the pathogenesis of a disease permitting to test different therapeutic interventions thus enabling the search for the most successful treatment options.

Models can be broadly classified according to their structure. A first important distinction to be made relates to whether or not time is one of the variables of the model. A *dynamic* model accounts for time-dependent changes in the state of the system whereas *static* models only consider the system at equilibrium. Static models are often used to describe the invariable relations among objects such as, for instance, proteins, metabolites or genes. A subfield of systems biology called network biology deals with such descriptions in terms of graphs or networks. Dynamical models allow for “forecasting” hypothetical scenarios and are quite valuable, for instance, in finding optimal clinical interventions in modelling of disease progress.

Another distinction concerns the involvement of the concept of randomness among the elements of the model. A model is *deterministic* if the current state of the model is uniquely determined by the model parameters and its previous states. That is, given an initial state, a deterministic model always follows the same “trajectory”. In contrast, the trajectories of a *stochastic* model may vary depending on the randomness present, in other words, by the variance of the stochastic variables that are therefore not described by unique values, but rather by probability distributions.

A third important classification criterion is the structure of the space from which the variables of the model take on their values. A variable is called *discrete* if it takes on values from a countable set (finite such as the set $\{0,1\}$ or infinite such as the set of integers), otherwise, it is called *continuous*, that is the case when its values can be arbitrarily close real numbers. If all variables are discrete then the model is said to be discrete. On the other hand, if all variables stand for continuous values then the model is continuous. Finally, hybrid models contain both continuous and discrete components. For instance a model of bacterial growth could represent with $n(t,x)$ the number of bacteria at time t in a certain volume at position x with time and space continuous but state n discrete.

In the following sections, we briefly illustrate different mathematical or computational modelling paradigms classifiable according to the mentioned criteria above. We leave to the reader the leisure of recognising each of them. The paragraphs also report examples of application of the methodologies in the biomedical field with the hope to convey the message of the convenience of quantitative reasoning in these fields. References of relevant works are provided and further references for interested readers are provided at the end of the article.

Differential Equations

Ordinary Differential Equations

To our knowledge one of the earliest work in the literature that used differential equations in biology was the influential and controversial paper by Daniel Bernoulli which was published in 1760. Using a simple model of differential equations, Bernoulli was able to show that the long-term benefits of inoculation outweigh the risk of immediate death as it may lead to increasing the life expectancy. For more information about the original Bernoulli’s model see Chapter 4 of Bacaër (2011), and, for recent work on Bernoulli’s model, see Dietz and Heesterbeek (2002).

For a given biological system of interacting species (substances, chemicals, ...) where their concentrations are functions of only one variable (time, age, or cell size,...), an ordinary differential equation (ODE) model of the system is basically a collection of equations, one for each species, describing the rate of change in the concentration of the species, with respect to the variable, in terms of the concentrations of other species in the system, see Fig. 2, which presents the well-known Lotka-Volterra or predator-prey model of two interacting quantities x (prey) and y (predator) with the initial concentrations being x_0 and y_0 , respectively. This

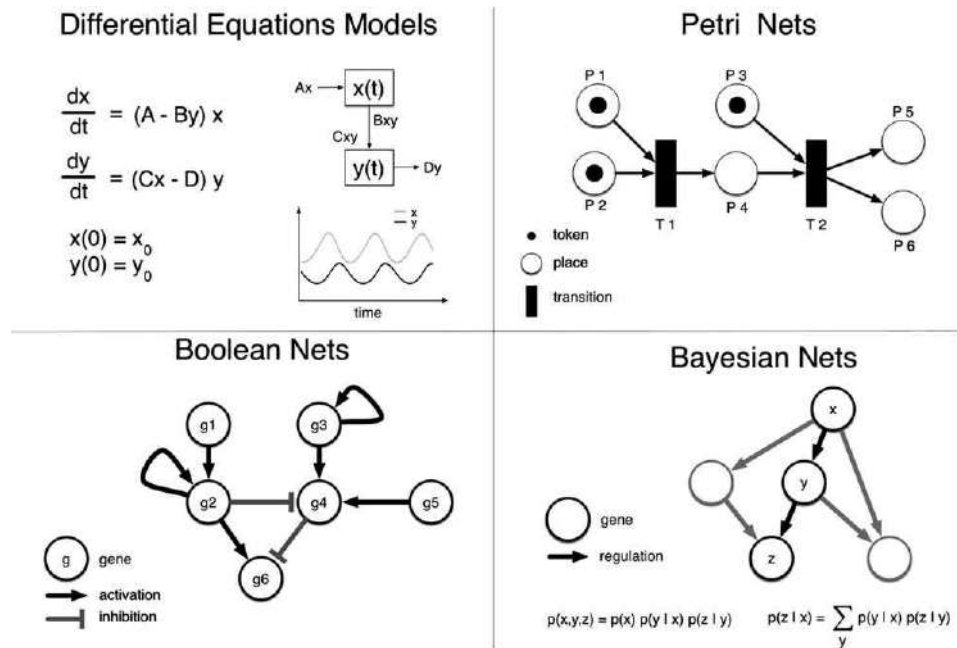


Fig. 2 Differential equations models exemplified in a predator-prey relation. The Petri-nets modelling paradigm is very generic and apt to model concurrency not just in biological systems. It consists in tokens moving among places through transitions epitomising conditions. Boolean networks have been conceived to describe gene regulation but are general enough to be applied to other field of science well. Bayesian networks are statistical methods which allow to estimate the strength (probabilistically speaking) of a relation among two objects such as genes, give a set of data.

model has four parameters, namely A , B , C , and D . For more information about predator-prey models, see Chapter 13 of [Bacaër \(2011\)](#). A similar but more specific example of ODE system applied to clinic is those in [Dell'Acqua and Castiglione \(2009\)](#) and [Jarrah et al. \(2014\)](#) about modelling muscular dystrophy.

Values of the parameters of a model are determined based on the available data and using any of several parameter estimation methods such as Genetic Algorithm, Particle Swarm, Simulated Annealing, Steepest Descent, to name few ([Moles et al., 2003](#)) and multiple shooting method for stochastic systems ([Peifer and Timmer, 2007](#)). For biochemical networks, the parameters could be assumed to be of one of two forms depending on the underlying kinetics: mass action or Hill function which includes Michaelis-Menten as a special case. While Mass Action (polynomial form) is appropriate when the reaction rate among species is proportional to the probability of the collision of the particles of the species, the Hill function in general is used in enzyme kinetics as well as protein's activation or inhibition.

Regardless of the optimization method, parameter estimation is probably the hardest step in the ODE approach for modelling biological systems ([Chou and Voit, 2009](#)) mainly due to the large number of species being modelled, their nonlinear interactions, and the fact that available quantitative data are limited. Yet the ODE approach is probably the most popular among all, mainly due to its solid and old mathematical foundation as well as its huge success in modelling a variety of biological systems at all levels; such as the famous population dynamic model predator-prey, Hodgkin-Huxley model of excitable neuron ([Hodgkin and Huxley, 1952](#)), and gene regulatory network of yeast cell cycle ([Tyson and Novák, 2015](#)) to name a few.

There are many different variations of the ODE approach, in particular, stochastic ODE ([Bachar et al., 2013](#)), delay ODE ([Parmar et al., 2015](#)). Also the S-systems ([Savageau, 1988](#)) is a special class of ODE models that assumes a specific structure of the system.

There are several general ODE solvers as well as specialized standalone software that can be used for parameter estimations and solving ODE. The standalone software CellDesigner ([Matsuoka et al., 2014](#)) and COPASI ([Hoops et al., 2006](#)) can be used for the modelling and simulation of biochemical networks and both have most of the mentioned methods and features above.

One of the main limitations of the ODE approach is the fact that it assumes that the species are well-mixed, and hence it cannot be used to properly model a system that is space-dependent. This can be easily resolved when using partial differential equations.

Partial Differential Equations

This approach enables modelling interacting species as functions of more than one variable. In particular, a partial differential equation (PDE) represents the change in the concentration of a substance with respect to not only time, as most ODE models, but also to other variables such as place (spatial coordinates), age and size of a cell.

Alan Turing developed a simple PDE model of two diffusible and interacting chemicals, and hence the name reaction-diffusion model, and he used the model to explain the pattern formation in the developing animal embryo ([Turing, 1952](#)). Since then, the reaction-diffusion model has been the basis of most PDE methods for the PDE modelling and simulation of complex biological patterns ([Kondo and Miura, 2010](#)). For a comprehensive treatment of the subject, see [Murray \(2003\)](#).

Modelling methods using the PDE approach have been developed to study various kinds of biological systems at all levels; such as ion singling (Ramay *et al.*, 2010), chemotaxis-driven migration of T-cells toward infection sites (Vroomans *et al.*, 2012), and climate change (Goosse *et al.*, 2010) as well as weather prediction (Steppeler *et al.*, 2003).

Similar to the ODE approach, solving PDE equations is not easy and is usually done using numerical techniques (Tadmor, 2012). Furthermore, methods for the parameter estimations of PDE models have recently been developed using different approaches such as least squares method (Muller and Timmer, 2002) and Bayesian methods (Xun *et al.*, 2013).

Difference Equations

In *Liber Abaci*, which was published in 1202, Fibonacci investigated how fast rabbits could breed. Assuming ideal environment, he developed a model for the number of rabbits using a difference equation and produced the famous Fibonacci's sequence: 1, 1, 2, 3, 5, 8, 13, 21, It is worth mentioning that Fibonacci's sequence has appeared in other areas within as well as outside biology. Fast forward to the 20th century, difference equations models of various biological systems are ubiquitous, in particular, the difference equations approach provides a discrete version of population biology in general (Cull *et al.*, 2005). Also, finite difference equations are used to develop numerical methods for solving ODE and PDE systems (Sewell, 2006). Furthermore, difference equations approach provide a generalization as well as a mathematical framework for cellular automata (Itoh and Chua, 2009) and other finite discrete methods. See the classical textbook (Elaydi, 2005) for more information on the theory of difference equations, and Allman and Rhodes (2004) and Cull *et al.* (2005) for examples of how the difference equations approach is used to model several biological systems such logistic growth and phylogenetic trees. Difference equations are best for studying discrete-time, population dynamics with stochastic features (Witten and de la Torre, 1984).

Petri Nets

Petri nets (PN) are directed bipartite graphs with nodes belonging to one of the two sets called states (also called places) and transitions (see Fig. 2). They were conceived by Carl Adam Petri in 1962 to describe chemical reactions (Petri and Reisig, 2008). Indeed, the states indicate substances and the transitions the reactions. Nodes identifying states are filled with tokens to indicate that the relative substance is present (alternatively, that the substance is present in sufficient concentration to allow the reaction to take place). When a transition has more than one incoming arcs (called *take arcs*) linked to corresponding states, those states all have to contain sufficient tokens to allow the transition to fire, meaning that the reaction takes place and the token is moved ahead in the states linked downward the exit arcs (called *give arcs*) (see Fig. 2). In this manner, the states code for conditions that need to be satisfied for a transition (or more generically an action) to take place. The flexibility of this modelling formalism allow its use to describe concurrent process in distributed systems. It has indeed been extensively used in computer science to describe distributed processing. In systems biology, Petri nets and its variants, developed to remove constraints such as the use of a discrete number of tokens (i.e., the coloured PN), or the use of deterministic transitions (i.e., the stochastic PN), have been employed to model biochemical networks (Chaouiya, 2007; Goss and Peccoud, 1998; Srivastava *et al.*, 2001) such as metabolic processes (Simao *et al.*, 2005) and cell signalling (Li *et al.*, 2007). The description of a biological process such as those at the molecular level just mentioned, is quite straightforward with PN. Moreover, given the intrinsic account for the temporal scale, such formalism is quite appealing for biologist that can analyse, mainly in a qualitative fashion but in principle also in a quantitative manner, complex biological phenomena by answering what-if questions such as "what will happen if the production of a molecular compound is inhibited?"

Boolean Networks

Boolean networks (BN) have been introduced by Kauffman in 1969 to model gene regulation (Kauffman, 1969, 1993). They represent genes as nodes of a graph with two types of arcs, activation or inhibition, denoting respectively an activation of, say, gene g_2 subsequent to the activation of gene g_1 , and inhibition, indicating the capacity of an express gene to block the transcription of another gene (see Fig. 2). The model describes the temporal evolution, carried out in discrete steps, of the active/inactive states of the set of genes. At each time step the state of each gene is determined by a Boolean logical rule which is a function of the state of the genes which impinge upon it (i.e., its regulators). For instance, for the network in Fig. 2, the Boolean formula for node g_4 could be $g_4(t+1) = g_3(t) \text{ AND } g_5(t) \text{ AND NOT } g_2(t)$.

The state of the genes can be updated all at once (synchronous update) or one at a time with a specified or unspecified order (asynchronous update). The two evolutions are not equivalent with the latter being much more complex to compute than the former. The space of the possible trajectories of the set of gene-expression grows exponentially with the number of genes thus its exploration to determine the dynamical properties of the BN becomes quickly unfeasible. On the other hand, the analysis of network properties such as its robustness or the steady-states of its dynamics which identify specific biological functions, are exactly the reward of using such models (Li *et al.*, 2004).

Boolean networks are generally inferred from time-series of gene expression data (D'haeseleer *et al.*, 2000) or are constructed by expert manual curation from literature data (Pedicini *et al.*, 2010). Boolean networks have also been employed to model signalling pathways (Saez-Rodriguez *et al.*, 2007). For these biological applications the limitation of the two-values logic has been removed in the multi-valued extension (Schaub *et al.*, 2007). Moreover, to account for the presence of noise and uncertainty, the stochastic extensions of probabilistic Boolean networks were introduced (Shmulevich *et al.*, 2002).

Bayesian Networks

A Bayesian network is a statistical model describing a set of variables and their conditional dependencies via a directed acyclic graph (see Fig. 2). The nodes of the network represent random variables whereas the edges stand for dependencies between couple of variables. The value of each variable is determined by a probabilistic function of the variables associated to the nodes than impinges upon it.

Bayesian networks were introduced by the work of Pearl (Pearl, 1988) as generic probabilistic methods to infer relationships among variables given a set of data. In fact, based on the formula from Bayes, there exist learning methods to infer the probability parameters also in case of incomplete data. Data coming from gene regulation activity provide a natural example of applicability of Bayesian networks. In this case the nodes-variables are the expression levels of the genes and the probabilistic relationships identified by the edges are estimated by means of the inference algorithms (Friedman, 2004). Data from signalling networks is amenable to this representation and analysis (Sachs *et al.*, 2005). The disadvantage due to the inability to model feedback loops present in biological networks has been removed in an extension called dynamic Bayesian networks (Husmeier, 2003; Kim *et al.*, 2003; Zou and Conzen, 2005).

Microscopic Ab-initio Models

Models in this category are usually discrete (both in time and space) dynamic models that aim to simulate/address biological problems for which spatial properties strongly affects the resulting dynamics. Discrete models are better suited to capture the discrete nature of individual cells than continuous models, especially when dealing with complex systems involving many variables such as sets of cells in different stages of their life cycle or distinguishable by the affinity of their receptors for different antigens. Cell-based models such as cellular automata, cellular Potts and agent-based models are able to represent the behaviour and interactions of each individual cell in a biological system. In these models, the macroscopic dynamics of a system is an emergent property of the microscopic interactions and for this reason they are considered bottom-up modelling approaches.

The Ising model (1925) is the simplest theoretical model of ferromagnetism used to study ferromagnetic phase transitions and is considered by many the precursor of cellular automata and agent based models. In the Ising model each magnetic dipole moment of atomic spins is represented as a discrete variable characterized by one of two states ($+1$ or -1) and arranged on a lattice. The interactions between spins are local and each spin is allowed to change synchronously based on the spin state of its neighbours.

Cellular Automata

Cellular automata (1940s) are a general computational model characterized by discrete space, state, and time (Ilachinski, 2001; Toffoli and Margolus, 1987; Wolfram, 2002) inspired in the first half of the 20th century by the work of John von Neumann (von Neumann, 1966) and Stanislaw Ulam (Ulam, 1952). A cellular automata model is a discrete dynamic system defined by a grid (a finite metric space), a finite set of possible cells states, a function describing the neighbourhood of a cell (the set of cells influencing the one under consideration) and a transition function determining the state of a cell in the next time step depending on the state of the cells in its neighbourhood (see Fig. 3).

Classical cellular automata are synchronous (all cells update their state simultaneously thus the grid state is independent from the order of update of the grid), local, and homogeneous. Many modifications of the classical model exist: asynchronous automata, heterogeneous automata, non-uniform cellular automata, stochastic automata and many others such as movable cellular automata (Psakhie *et al.*, 1995), Brownian cellular automata (Lee and Peper, 2010).

Cellular automata have been extensively used as a modelling paradigm in biology (Ermentrout and Edelstein-Keshet, 1993). Some applications include the study of idiotypic B cell proliferation in the human immune system (De Boer *et al.*, 1992) or the dynamics of infectious diseases (Zorzenon dos Santos and Coutinho, 2001).

Cellular Potts models, also known as Glazier Graner-Hogeweg models (Graner and Glazier, 1992; Glazier and Graner, 1993), are an evolution of cellular automata in which each cell occupies more than one lattice site or pixel. For this reason, each cell has a shape which varies over time as a function of a Hamiltonian representing the adhesion energy of the cell and other mechanical constraints such as surface-area-to-volume ratio. One of the most known framework for cellular Potts model is CompuCell3D (Chaturvedi *et al.*, 2005; Popławski *et al.*, 2008) which has been used in several different applications, from anatomical and pathological conditions of tissues and organs to models of single-species and multi-species biofilm growth. In this hybrid framework, the mechanical properties of the cell membrane can be integrated with continuous models of reaction-diffusion dynamics and models of intracellular signalling dynamics achieving a complete and realistic multi-scale model of a complex biological system.

Agent-Based Models

Agent-based models (ABM) are an evolution of the simpler cellular automata model in which each agent is an autonomous decision-making entity interacting on the basis of a given set of rules which are not necessarily identical (Bonabeau, 2002; Gilbert, 2008). Strictly speaking, ABMs are not discrete in time and space as they are exemplary of hybrid systems where discrete time variables are combined with continuous state-space ones. Nevertheless, in their applications to biology and medicine and with the purpose to speed up execution on a computer, agents are usually placed on a structure (a two dimensional or three-dimensional lattice representing real-world spatial environment such as a lymph node or a complex network representing social contacts or sexual interactions) that defines the couples or groups of agents that are going to interact at a given time. Agent-based models have

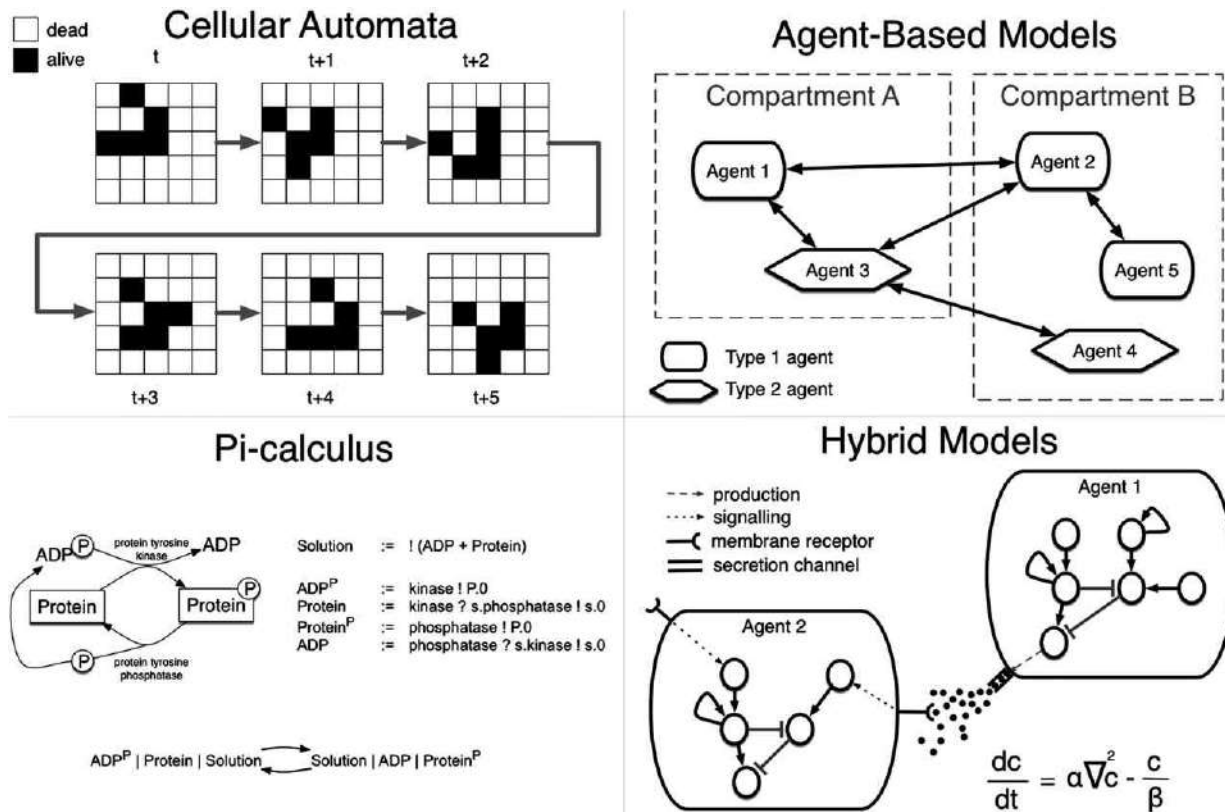


Fig. 3 Cellular automata and agent-based models represent entities individually. These evolve according to defined rules whose complexity range from rather simple (often Boolean) functions in cellular automata to sophisticated algorithms for the agents' behaviour in agent-based models. Furthermore, as an example, one can construct complex hybrid models embedding Boolean networks in agents representing biological cells to set up the rule for its differentiation. Pi-calculus is a different approach used to formally describe a complex system such as a chemical reactor and to conduct (in the case of stochastic calculi, e.g., stochastic k-calculus) simulations much like in the previously described methodologies.

been used to study many different problems. For instance, they have been used to simulate HIV or gonorrhoea epidemics where agents represent individuals part of a sexual contact network (Mei *et al.*, 2011) or to describe the population dynamics of ecosystems in which each agent is an animal or a plant. They have also being used to model the intracellular dynamics phenomena in which agents are viral particles moving within the cytoplasm of an infected cell. Finally, they have been extensively used to describe the dynamics of the immune system. In this case the agents represent all the entities involved in the immune response to a pathogen such as immune cells, epithelial cells, bacteria, viral particles etc (Castiglione and Celada, 2015).

Several frameworks can be used to implement agent based models: NETLOGO, FLAME, MASON, JASON and REPAST are some of the most versatile and supported frameworks.

Process Calculi

The design of a computational implementation of any biological system starts from the knowledge of the living system under consideration and by abstracting its main characteristics. This task is fulfilled by means of some formal method such as those describing the syntax of a computer language. The most striking feature of biological phenomena that is difficult to capture in symbolic formal methods is concurrency. Formal systems, therefore, need to be able to fully denote such biological complexity or otherwise turn out inaccurate and eventually useless.

Process calculi or process Algebras (PA) are formal systems devised to easily express parallel computations and their properties (like Church's lambda-calculus introduced to express Turing computable arithmetic functions). The most successful system was Milner's pi-calculus introduced to express interactions between concurrent processes (Milner, 1989). PAs share some similarity with formalism used in the description of chemical reactions (see Fig. 3 for the example of the phosphorylation/dephosphorylation reaction expressed in pi-calculus). This was precisely established with the introduction of the Chemical Abstract Machine or CHAM (Berry and Boudol, 1992), where the states of the parallel machine are chemical solutions and molecules can interact according to reaction rules.

Biological phenomena involve millions of moving particles in any single cell while millions of cells form tissues which transport substances, transform energy use signals, etc. Notably, biological systems process information necessary to the organization of life by the same mechanisms they use to processes any other element, like the nutrients. Process algebras are considered suitable formalisms to model these kinds of systems because the transmitted information is processed *concurrently* and

asynchronously. Moreover, part of essential transactions which happen in biological systems are inherently computational and therefore they can be expressed with formal languages used to describe concurrent processes.

While representing biological systems with PAs, several researchers realised that there was a missing dimension: quantitative aspects were neglected in the description of the interaction rules. The stochastic pi-calculus (Priami, 1995) was an enriched Milner's formalism with a quantitative context semantics giving access to the world of simulations of biochemical processes (Regev *et al.*, 2001; Priami *et al.*, 2001). Further developments were obtained by deriving new formal systems by adding missing features to PAs: bio-ambients (Cardelli and Gordon, 2000) and brane-calculi (Cardelli, 2005) allowing compartments, the kappa-calculus (Danos and Laneve, 2004) with the specific purpose of representing protein interactions.

The prominent aspect to remark regarding the convenience for using this approach in describing biological processes is that it enables, for instance, the syntactic and semantic comparison (hence prompt translation of the properties of the first to the second system) between two concurrent systems once formalised with a PA or to build incrementally complex process from a multitude of elementary processes being either independent or interrelated.

Closing Remarks

Computational methods are used to make sense of the vast amount of data produced by experimental biology high throughput technologies and, importantly, to “fill the dots” where data is incomplete in revealing the underlying structure or in describing the temporal evolution of the system under study. Mathematical or computational modelling methods are applied and even developed anew to analyse such data, to dig information out of it, to discover structures, to generalize conclusions, to provide scenarios to choose from, etc. The accuracy of these computational approaches ranges from low (that is when we call the approach qualitative) to high (i.e., that is quantitative) depending on the data availability.

The methodologies range from classical differential equations to more unconventional agent-based simulation, passing through process algebra and statistical approaches. These models have shown to be promising if not even concretely useful in some cases, revealing biological aspects of the phenomena under consideration which were impossible to see just by inspecting the data. Vice versa, biology is providing a number of challenging problems to appraise the methods themselves, encouraging computational scientists to extend and improve existing methodologies (Machado *et al.*, 2011).

One of the greatest challenge in this respect is the construction of multi-scale models embracing the wide range of time and length scales of biological phenomena. Whereas few modelling techniques though theoretical able to handle different scales (e.g., process algebras or Petri nets), their practical use is impaired by the combinatorial explosion of variants eventually leading to computational intractable problems. In this case the obvious trick is to combine two or (rarely) more approaches that have their strength in different time/length scale into a unified hybrid multi-scale system (Fisher and Henzinger, 2007; Bortolussi and Policriti, 2008; Fromentin *et al.*, 2010). This is for instance the case of agent-based models of cellular systems where each agent/cell is equipped with a system of ordinary differential equations or a Boolean network to describe the regulatory network among genes driving its behaviour (see Fig. 3).

Computational biology is a flourishing field of applied mathematics and computer science. Only in the past two decades the collective effort of computational scientists combined with the expertise of field biologist have produced a number of open-software that offer the model techniques described in this brief article to analyse experimental data and/or to simulate hypothetical scenarios. Review papers such as Bartocci and Lió (2016) offer a nice view of what is available at the time of that publication. There is no reason to believe that such vigorous effort in the systems biology community will grow smaller, but rather just the opposite, new tools will be available soon, either as improvements of existing methodologies or able to exploit and combine different formalisms and modelling paradigms to provide powerful platforms for the analysis of the ever increasing experimental and diagnostic data.

Acknowledgement

Abdul Jarrah was supported by the National Science Foundation under Grant No. DMS-1440140 in the Fall 2018 while the author was in residence at the Mathematical Sciences Research Institute in Berkeley, California, USA. Filippo Castiglione acknowledge partial support from the Institute for Advanced Studies of the University of Amsterdam. Emiliano Mancini acknowledges partial support from the SimCity project of the Dutch NWO, eScience agency under contract C.2324.0293.

See also: Cell Modeling and Simulation. Computational Systems Biology Applications. Computational Systems Biology. Multi-Scale Modelling in Biology. Natural Language Processing Approaches in Bioinformatics. Pathway Informatics

References

- Allman, E., Rhodes, J., 2004. *Mathematical Models in Biology: An Introduction*. New York: Cambridge University Press.
- Bac  r, N., 2011. *A Short History of Mathematical Population Dynamics*. London: Springer-Verlag.
- Bachar, M., Batzel, J., Ditlevsen, S., 2013. *Stochastic Biomathematical Models: With Applications to Neuronal Modeling*. Heidelberg: Springer.
- Bartocci, E., Li  , P., 2016. Computational modeling, formal analysis, and tools for systems biology. *PLOS Computational Biology* 12 (1), e1004591. doi:10.1371/journal.pcbi.1004591.
- Berry, G., Boudol, G., 1992. The chemical abstract machine. *Theoretical Computer Science* 96, 217–248.

- Bonabeau, E., 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America* 99, 7280–7287.
- Bortolussi, L., Policriti, A., 2008. Hybrid systems and biology. In: Bernardo, Marco, Degano, Pierpaolo, Zavattaro, Gianluigi (Eds.), *Formal Methods for Computational Systems Biology*, vol. 5016 of LNCS. Springer, pp. 424–448.
- Cardelli, L., Gordon, A.D., 2000. Mobile ambients. *Theoretical Computer Science* 240 (1), 177–213.
- Cardelli, L., 2005. Brane Calculi. In: Danos, V., Schachter, V. (Eds.), *Computational Methods in Systems Biology (CMSB 2004)*. *Lecture Notes in Computer Science* 3082. Berlin, Heidelberg: Springer, pp. 257–278.
- Castiglione, F., Celada, F., 2015. *Immune System Modeling and Simulation*. Boca Raton, FL, USA: CRC Press.
- Chauviya, C., 2007. Petri net modelling of biological networks. *Briefings in Bioinformatics* 8, 210–219.
- Chaturvedi, R., Huang, C., Kazmierczak, B., *et al.*, 2005. On multiscale approaches to three-dimensional modeling of morphogenesis. *Journal of the Royal Society Interface* 2, 237–253.
- Chou, C., Voit, E.O., 2009. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathematical Biosciences* 219, 57–83.
- Cull, P., Flahive, M., Robson, R., 2005. *Difference Equations: From Rabbits to Chaos*. New York: Springer.
- Danos, V., Laneve, C., 2004. Formal molecular biology. *Theoretical Computer Science* 325 (1), 69–110.
- De Boer, R.J., Segel, L.A., Perelson, A.S., 1992. Pattern formation in one- and two-dimensional shape-space models of the immune system. *Journal of Theoretical Biology* 155 (3), 295–333.
- Dell'Acqua, G., Castiglione, F., 2009. Stability and phase transitions in a mathematical model of Duchenne muscular dystrophy. *Journal of Theoretical Biology* 260 (2), 283–289. doi:10.1016/j.jtbi.2009.05.037.
- Dietz, K., Heesterbeek, J.A.P., 2002. Daniel Bernoulli's epidemiological model revisited. *Mathematical Biosciences* 180, 1–21.
- D'haeseleer, P., Liang, S., Somogyi, R., 2000. Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics* 16 (8), 707–726.
- Elaydi, S., 2005. *An Introduction to Difference Equations*. New York: Springer.
- Ermentrout, G.B., Edelstein-Keshet, L., 1993. Cellular automata approaches to biological modeling. *Journal of Theoretical Biology* 160 (1), 97–133.
- Fisher, J., Henzinger, T.A., 2007. Executable cell biology. *Nature Biotechnology* 25 (11), 1239–1249. doi:10.1038/nbt1356.
- Friedman, N., 2004. Inferring cellular networks using probabilistic graphical models. *Science* 303 (5659), 799–805.
- Fromentin, J., Eveillard, D., Roux, O., 2010. Hybrid modeling of biological networks: Mixing temporal and qualitative biological properties. *BMC Systems Biology* 4, 79. doi:10.1186/1752-0509-4-79. (PMID: 20525331).
- Gilbert, N., 2008. *Agent-Based Models*. Los Angeles: Sage Publications.
- Glazier, J.A., Graner, F., 1993. Simulation of the differential adhesion driven rearrangement of biological cells. *Physical Review E* 47, 2128–2154.
- Goosse, H., Barriat, P.Y., Lefebvre, W., Loutre, M.F., Zunz, V., 2010. Chapter 3: Modelling the Climate System of Introduction to Climate Dynamics and Climate Modeling. Online textbook available at <http://www.climate.be/textbook>.
- Goss, P.J., Peccoud, J., 1998. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proceedings of the National Academy of Sciences of the United States of America* 95, 6750–6755.
- Graner, F., Glazier, J.A., 1992. Simulation of biological cell sorting using a two-dimensional extended Potts model. *Physical Review Letters* 69 (13), 785–790.
- Hodgkin, A.L., Huxley, A.F., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117, 500–544.
- Hoops, S., Sahle, S., Gauges, R., *et al.*, 2006. COPASI: A complex pathway simulator. *Bioinformatics* 22, 3067–3074.
- Husmeier, D., 2003. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics* 19 (17), 2271–2282.
- Ilachinski, A., 2001. *Cellular Automata: A Discrete Universe*. River Edge, NJ, USA: World Scientific Publishing Company.
- Itoh, K., Chua, L., 2009. Difference equations for cellular automata. *International Journal of Bifurcation and Chaos* 19, 805–830.
- Jarrah, A.S., Castiglione, F., Evans, N.P., Grange, R.W., Laubenbacher, R., 2014. A mathematical model of skeletal muscle disease and immune response in the mdx mouse. *BioMed Research International*. 871810. doi:10.1155/2014/871810.
- Ji, Z., Yan, K., Li, W., Hu, H., Zhu, X., 2017. Mathematical and computational modeling in complex biological systems. *BioMed Research International*. 5958321. doi:10.1155/2017/5958321.
- Kauffman, S., 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* 22, 437–467.
- Kauffman, S.A., 1993. *The Origins of Order: Self-organization and Selection in Evolution*. New York: Oxford University Press.
- Kim, S., Imoto, S., Miyano, S., 2003. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics* 4 (3), 228–235.
- Kondo, S., Miura, T., 2010. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science* 329, 1616–1620.
- Lee, J., Peper, F., 2010. Efficient computation in Brownian cellular automata. In: Peper, F., Urneo, H., Matsui, N., Isokawa, T. (Eds.), *Natural Computing. Proceedings in Information and Communications Technology 2*. Tokyo: Springer.
- Li, C., Ge, Q.W., Nakata, M., Matsuno, H., Miyano, S., 2007. Modelling and simulation of signal transductions in an apoptosis pathway by using timed Petri nets. *Journal of Biosciences* 32, 113–127.
- Li, F., Long, T., Lu, Y., Ouyang, Q., Tang, C., 2004. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America* 101 (14), 4781–4786.
- Machado, D., Costa, R.S., Rocha, M., *et al.*, 2011. Modeling formalisms in systems biology. *AMB Express* 1, 45. doi:10.1186/2191-0855-1-45.
- Materi, W., Wishart, D.S., 2007. Computational systems biology in drug discovery and development: Methods and applications. *Drug Discovery Today* 12 (7–8), 295–303. doi:10.1016/j.drudis.2007.02.013.
- Matsuoka, Y., Funahashi, A., Ghosh, S., Kitano, H., 2014. Modeling and simulation using CellDesigner. In: Miyamoto-Sato, E., Ohashi, H., Sasaki, H., Nishikawa, J., Yanagawa, H. (Eds.), *Transcription Factor Regulatory Networks. Methods in Molecular Biology (Methods and Protocols)*, vol. 1164. New York: Humana Press.
- Mei, S., Quax, R., van de Vijver, D., Zhu, Y., Sloot, P.M.A., 2011. Increasing risk behaviour can outweigh the benefits of antiretroviral drug treatment on the HIV incidence among men-having-sex-with-men in Amsterdam. *BMC Infectious Diseases* 11, 118.
- Milner, R., 1989. *Communication and Concurrency*. Upper Saddle River, NJ, USA: Prentice Hall, Inc., (ISBN:0-13-115007-3).
- Mogilner, A., Wollman, R., Marshall, W.F., 2006. Quantitative modeling in cell biology: What is it good for? *Developmental Cell* 11, 279–287.
- Moles, C.G., Mendes, P., Banga, J.R., 2003. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research* 13, 2467–2474.
- Muller, T., Timmer, J., 2002. Fitting parameters in partial differential equations from partially observed noisy data. *Physical Review* 171, 1–7.
- Murray, J., 2003. *Mathematical Biology*. Berlin: Springer-Verlag.
- Parmar, K., Blyuss, K.B., Kyrychko, Y.N., Hogan, S.J., 2015. Time-delayed models of gene regulatory networks. *Computational and Mathematical Methods in Medicine* 2015, 16. doi:10.1155/2015/347273. (Article ID 347273).
- Pearl, J., 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann Publishers Inc.
- Pedicini, M., Barrenas, F., Clancy, *et al.*, 2010. Combining network modeling and gene expression microarray analysis to explore the dynamics of Th1 and Th2 cell regulation. *PLOS Computational Biology* 6 (12), e1001032. doi:10.1371/journal.pcbi.1001032.
- Peifer, M., Timmer, J., 2007. Deterministic inference for stochastic systems using multiple shooting and a linear noise approximation for the transition probabilities. *IET System Biology* 1, 78–88.
- Petri, C.A., Reisig, W., 2008. Petri net. *Scholarpedia* 3 (4), 6477. Available at: http://www.scholarpedia.org/article/Petri_net.
- Poplawski, N.J., Shrinifard, A., Swat, M., Glazier, J.A., 2008. Simulation of single-species bacterial-biofilm growth using the Glazier-Graner-Hogeweg model and the CompuCell3D modeling environment. *Mathematical Biosciences and Engineering* 5 (2), 355.
- Priami, C., 1995. Stochastic pi-calculus. *Computer Journal* 38 (7), 578–589.

- Priami, C., Regev, A., Shapiro, E., Silverman, W., 2001. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters* 80 (1), 25–31.
- Pskahie, S.G., Horie, Y., Korostelev, S.Y., *et al.*, 1995. Method of movable cellular automata as a tool for simulation within the framework of mesomechanics. *Russian Physics Journal* 38 (11), 1157–1168.
- Ramay, H., Jafri, M.S., Lederer, W.J., Sobie, E.A., 2010. Predicting local SR Ca^{2+} dynamics during Ca^{2+} wave propagation in ventricular myocytes. *Biophysical Journal* 98, 2515–2523.
- Regev, A., Silverman, W., Shapiro, E., 2001. Representation and simulation of biochemical processes using the π -calculus process algebra. In: Altman, R.B., Dunker, A.K., Hunter, L., Klein, T.E. (Eds.), *Pacific Symposium on Biocomputing 6*. Singapore: World Scientific Press, pp. 459–470.
- Saadatpour, A., Albert, R., 2016. A comparative study of qualitative and quantitative dynamic models of biological regulatory networks. *EPJ Nonlinear Biomedical Physics* 4, 5. doi:10.1140/epjnbp/s40366-016-0031-y.
- Saez-Rodriguez, J., Simeoni, L., Lindquist, J., *et al.*, 2007. A logical model provides insights into T cell receptor signaling. *PLOS Computational Biology* 3 (8), e163.
- Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., Nolan, G., 2005. Causal protein-signaling networks derived from multi-parameter single-cell data. *Science* 308 (5721), 523–529.
- Savageau, M.A., 1988. Introduction to S-systems and the underlying power-law formalism. *Mathematical and Computer Modelling* 11, 546–551.
- Schaub, M.A., Henzinger, T.A., Fisher, J., 2007. Qualitative networks: A symbolic approach to analyze biological signaling networks. *BMC Systems Biology* 1, 4.
- Sewell, G., 2006. *The Numerical Solution of Ordinary and Partial Differential Equations*. Chichester, United Kingdom: John Wiley & Sons.
- Shmulevich, I., Dougherty, E.R., Zhang, W., 2002. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE* 90 (11), 1778–1792.
- Simao, E., Remy, E., Thieffry, D., Chauviya, C., 2005. Qualitative modelling of regulated metabolic pathways: Application to the tryptophan biosynthesis in *E. coli*. *Bioinformatics* 21 (2), 190–196.
- Srivastava, R., Peterson, M.S., Bentley, W.E., 2001. Stochastic kinetic analysis of the *Escherichia coli* stress circuit using sigma(32)-targeted antisense. *Biotechnology and Bioengineering* 75, 120–129.
- Stappeler, J., Hess, R., Schattler, U., Bonaventura, L., 2003. Review of numerical methods for nonhydrostatic weather prediction models. *Meteorology and Atmospheric Physics* 82 (1–4), 287–301.
- Tadmor, E., 2012. A review of numerical methods for nonlinear partial differential equations. *Bulletin of the American Mathematical Society* 49, 507–554.
- Toffoli, T., Margolus, N., 1987. *Cellular Automata Machines: A New Environment for Modeling*. Cambridge, MA, USA: MIT Press.
- Turing, A., 1952. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London Series B Biological Sciences* 237, 37–72.
- Tyson, J., Novák, B., 2015. Models in biology: Lessons from modeling regulation of the eukaryotic cell cycle. *BMC Biology* 13, 46 doi:10.1186/s12915-015-0158-9.
- Ulam, S., 1952. Random processes and transformations. In: *Proceedings of the International Congress of Mathematicians*, pp. 264–275. Rhode Island: American Mathematical Society.
- von Neumann, J., 1966. *The Theory of Self-reproducing Automata*. Urbana, IL: University of Illinois Press.
- Vroomans, R.M.A., Marée, A.F.M., de Boer, R.J., Beltman, J.B., 2012. Chemotactic migration of T cells toward dendritic cells promotes the detection of rare antigens. *PLOS Computational Biology* 8, e1002763.
- Witten, M., de la Torre, D., 1984. Biological populations obeying difference equations: The effects of stochastic perturbation. *Journal of Theoretical Biology* 111, 493–507.
- Wolfram, S., 2002. *A New Kind of Science*. Champaign, IL: Wolfram Media, Inc.
- Xun, X., Cao, J., Mallick, B., Carroll, R.J., Maity, A., 2013. Parameter estimation of partial differential equation models. *Journal of the American Statistical Association* 108 (503), doi:10.1080/01621459.2013.794730.
- Zorzenon dos Santos, R.M., Coutinho, S., 2001. Dynamics of HIV infection: A cellular automata approach. *Physical Review Letters* 87, 168102.
- Zou, M., Conzen, S., 2005. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* 21 (1), 71–79.

Further Reading

- Castiglione, F., Celada, F., 2015. *Immune System Modeling and Simulation*. Boca Raton: CRC Press.
- Ciocchetta, F., Hillston, J., 2009. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science* 410 (33–34), 3065–3084.
- Deutsch, A., Dormann, S., 2005. *Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis*. Boston: Birkhäuser.
- Elaydi, S., 2005. *An Introduction to Difference Equations*. New York: Springer.
- Ellner, S.P., Guckenheimer, J., 2006. *Dynamic Models in Biology*. New Jersey: Princeton University Press.
- Gilbert, N., 2008. *Agent-Based Models*. Los Angeles: Sage Publications.
- Ilachinski, A., 2001. *Cellular Automata: A Discrete Universe*. River Edge, NJ, USA: World Scientific Publishing Company.
- Kauffman, S.A., 1995. *At Home in the Universe: The Search for Laws of Self-organization and Complexity*. Oxford: Oxford University Press.
- Laneve, C., Tarissan, F., 2007. Simple calculus for proteins and cells. *Electronic Notes in Theoretical Computer Science* 171, 139–154.
- Milner, R., 1999. *Communicating and Mobile Systems: The π -Calculus*. Cambridge: Cambridge University Press.
- Murray, J., 2003. *Mathematical Biology*. Berlin: Springer-Verlag.

Biographical Sketch



Filippo Castiglione is researcher at the Istituto per le Applicazioni del Calcolo of the National Research Council of Italy and adjunct professor of Computational Biology at the Department of Mathematics and Physics of Roma Tre University. He graduated in computer science at the University of Milan, Italy, and got a PhD in Scientific Computing at the University of Cologne, Germany. He has been PostDoc at the Institute for Medical Bio-Mathematics in Tel Aviv, Israel and visiting research fellow at the IBM - T.J. Watson Research Center, Yorktown Heights (NY), at the Department of Molecular Biology at Princeton University and at the Department of Cell Biology, Harvard Medical School, Boston. Filippo has published one book and about 100 reviewed research papers among journals, books and conferences proceedings. He is the main author of the C-ImmSim agent-based simulation model of the immune system. Filippo has been involved as principal investigator in two EU-funded projects of the ICT for Health of the 6th Framework Programme (FP6) and has coordinate a project of the FP7. His research interests range from the study of complex systems to the modeling of biological systems, machine learning and high-performance computing.



Emiliano Mancini is programme developer on Health Systems Complexity at the Institute for Advanced Study in Amsterdam, where he coordinates the research in various academic disciplines for interdisciplinary research projects. In this role, he also engages private companies and stakeholders for the valorisation and utilization of these academic projects as well as collaborate on the strategic plan for the growth of the Institute. Emiliano obtained his Master degree in Physics at University of Rome Tor Vergata and a PhD in Computational Science at University of Amsterdam simulating infectious diseases (HIV and H1N1) using agent-based modeling. As a postdoc Emiliano has worked between University of Amsterdam and the Nanyang Technological University in Singapore on modeling the dynamics of several complex biological systems. He worked as a consultant or team member of several start-up companies and has been involved in the development of a clinical decision support platform for HIV treatment and on a non-invasive continuous glucose meter funded by SMART-MIT. Emiliano was also responsible for the organization and supervision of several experiments on crowd dynamics for the Kumbh Mela Experiment: an Indo-Dutch project to study the Kumbh Mela, the largest religious event in the world.



Marco Pedicini is professor of Computer Science at the Department of Mathematics and Physics of Roma Tre University. He received his Ph.D. in Mathematics (Mathematical Logic and Theoretical Computer Science) from the University of Paris 7 in 1998. He was researcher at Italian National Research Council (CNR). Since his PhD thesis, Marco has developed ideas in TCS and his interests include logic in computer science, cryptography, computer security, parallel and distributed computing, computational methods for systems biology, computational number theory, and more recently quantum computing. Although his whole activity could be classifiable in TCS it finds in interdisciplinary research its own specificity. Interactions with other disciplines are the key to evaluate his scientific activity: in fact, he developed a very special kind of expertise while coping deep theoretical results with practical issues in advanced applications in collaboration with colleagues of different disciplines. Marco has published 30 scientific papers, many of them in leading international journals and selective peer reviewed conferences. In order to witness a broad area in Marco interests, these papers not only appeared on top conferences and journals in logic for computer science but also on top journals covering application of computer science to biology and medicine.



Abdul Salam Jarrah is a Professor of Mathematics and a faculty member of the Master of Science in Biomedical Engineering Programme at the American University of Sharjah (AUS) in the United Arab Emirates. Before joining the AUS in 2009, he was a senior research scientist at the Virginia Bioinformatics Institute and Affiliate Assistant Professor at the Department of Mathematics in Virginia Tech, USA. Prior to that, he was as an Assistant Professor of Applied Mathematics at East Tennessee State University, USA. In addition, Abdul was a member of the Mathematical Sciences Research Institute at Berkeley, USA from August 2016 to January 2018. He received his PhD in Mathematics from New Mexico State University in 2002. His current research interests span many areas of mathematics including discrete dynamical systems, discrete homology theory of graphs, computational algebra, and their applications in biology, especially the modeling and simulation of biological systems. He has made several contributions to these fields through many publications in international journals.

Data Formats for Systems Biology and Quantitative Modeling

Martin Golebiewski, Heidelberg Institute for Theoretical Studies (HITS gGmbH), Heidelberg, Germany

© 2019 Elsevier Inc. All rights reserved.

Standards for Computational Systems Biology

Standards support communities with a basis for mutual understanding and information exchange and thereby shape our everyday life. Common standards are indispensable for collaborative work. An example is the Baltimore fire in 1904, when more than 1500 buildings were destroyed even though fire departments from Washington, Philadelphia, New York, and other towns provided aid, as they found their hose couplings did not match the city's fire hydrants due to a lack of standardization. This reduced the impact of help significantly and led afterwards to a national standard for hose couplings and fire hydrants in the USA (Seck and Evans, 2004). A similar breakdown of collaborative work due to a lack of standardization, however, could happen in any field that requires interfacing between different parts, especially in research fields such as in the modern life sciences with their increasing flood and complexity of data and all the vast variety of technologies that have to be brought together like in a jigsaw puzzle.

Standardization of data and their documentation is all the more important for highly interdisciplinary and collaborative fields such as systems biology and systems medicine, where computational models are created that can assist with understanding, describing, analyzing, and prediction of biological systems and their functions. Over the last 20 years computational modeling has grown from a niche activity in systems biology into a standard tool in the life sciences, as documented by the increasing number of published models stored in public model repositories (Li *et al.*, 2010; Yu *et al.*, 2011). Model-driven design, testing, analysis, and optimization of biological systems are performed computationally by integrating a variety of heterogeneous data together within mathematical frameworks in order to understand and predict the complex and dynamic, often nonlinear molecular interactions within cells, as well as the interplay of cells within tissues, organs, organisms, and beyond that even the interaction of different organisms in whole biological ecosystems. Because of the interdisciplinary approach in systems biology these datasets are coming from different sources, derived from different experimental setups, and correspond to various technologies applied to collect the data.

A computational model in systems biology, thus, typically relies on a collection of varied data, which can include, for example, data from genomics, transcriptomics, proteomics, or metabolomics experimental setups, reaction kinetics information, and imaging data. This requires a consistent structuring and description of data and computer models with their underlying experimental or modeling processes, comprising the standardized description of applied methods, biological material and workflows for data processing, analysis, exchange, and integration (e.g., into computational models), as well as of the setup, handling, and simulation of the resulting models. Hence, standards for formatting and describing experimental data, applied workflows, and computer models have become important, especially for data integration across the biological scales for multiscale approaches. Such a consistent documentation based on standards ensures that the data and corresponding metadata (data describing the data and its context), as well as models, methods, and visualizations are structured and described in a "FAIR" manner: Findable, Accessible, Interoperable, and Reusable (Wilkinson *et al.*, 2016).

Systems biology standards are not static but develop and evolve with the progress of science and technology. A major challenge is to harmonize the standardization efforts in the different fields that refer to different approaches and technologies and to make the corresponding standards interoperable. The standards used for encoding and annotating models and related data are designed by experts with an understanding of what key information will comprise the outcome of an experiment, and how this information is best structured and formatted. The development of such standards for systems biology typically occur at a grassroots level driven by the corresponding scientific communities themselves with their need for exchanging data and models. The COMBINE (see "Relevant Websites section") network ("COMputational Modeling in BIOlogy NEtwork" (Hucka *et al.*, 2015b; Myers *et al.*, 2017)), for example, is a consortium of groups involved in the development of open community standards and formats used in computational modeling in biology. It was formed in 2009 following the observation that many standardization efforts shared similar goals and sometimes even involved the same individuals. The network helps foster greater interaction and awareness of the activities in different standards' development, which encourages the federated projects to develop standards that are more likely to be interoperable and less likely to overlap substantially than if the efforts proceeded separately. Building on the experience of mature standards, which already have stable specifications, software support, user bases, and community governance, it also supports emerging efforts aimed at filling gaps or addressing new needs in the overall interoperability landscape. The COMBINE initiative published the first collection of systems and synthetic biology standards as a special issue of the *Journal of Integrative Bioinformatics (JIB)* in 2015 (Schreiber *et al.*, 2015). Since then a regular special issue of *JIB* serves as both an overview of existing standards as well as an update of the current state of standards in the domain (Schreiber *et al.*, 2016, 2018).

Given the diversity of research topics in computational biology, and the different facets of modeling, multiple specific standards, with particular purposes, are needed – also beyond the COMBINE standards – in order to formalize and serialize the data (Stromback *et al.*, 2007; Klipp *et al.*, 2007). In the following, several standards for specific tasks in systems biology and related fields will be introduced, in order to provide examples for typical workflows and the standards that can be applied therein.

Standards for Data Used for Model Construction

The first step in generating a model is collating the datasets that need to be integrated into the model. This task heavily relies on the datasets being formatted and annotated correctly. Reporting and annotation checklists, or “minimum information guidelines” as they are often called, for different types of data derived from a big variety of laboratory technologies are readily cataloged for the biosciences as part of MIBBI (Minimum Information for Biological and Biomedical Investigations (Taylor *et al.*, 2008)). Such guidelines are widely accepted for some data types, for example, for the description of microarray experiments (MIAME, Minimum Information About a Microarray Experiment (Brazma *et al.*, 2001)), genome sequences (MIGS, Minimum Information about a Genome Sequence (Field *et al.*, 2008)), or proteomics experiments (MIAPE, Minimum Information About a Proteomics Experiment (Taylor *et al.*, 2007)). The FAIRsharing community makes a wide range of minimum information checklists available for researchers in the MIBBI FAIRsharing collection (see “Relevant Websites section”). Finding comprehensive lists about formatting standards for different types of data can be difficult, and not all formatting standards or annotation checklists are still maintained, or fully usable. To reduce the time required to identify standards that are maintained and usable, FAIRDOM (see “Relevant Websites section”) maintains a FAIRsharing collection of formatting standards, and checklists that are known to be actively used within the systems biology community (see “Relevant Websites section”). This collection is based on a community survey commissioned by Infrastructure for Systems Biology Europe (ISBE) (see “Relevant Websites section”) in 2015 (Stanford *et al.*, 2015).

Experimentally obtained kinetic data that describe the dynamics of biochemical reactions (e.g., metabolic or signaling reactions within a cellular network), for instance, can be obtained from databases such as SABIO-RK (Wittig *et al.*, 2012) (see “Relevant Websites section”). This manually curated database contains data that are either extracted by hand from the scientific literature (Wittig *et al.*, 2014) or directly submitted from laboratory experiments (Swainston *et al.*, 2010). The database content includes kinetic parameters in relation to the reactions and biological sources, as well as experimental conditions under which they were obtained, and is fully annotated to other resources, making the data easily accessible via web interface or web services for integration into the computational models. Through export of the data (together with its annotations and SABIO-RK identifiers for tracing back to the original dataset) in standardized data exchange formats like SBML (Systems Biology Markup Language) (Hucka *et al.*, 2003) this allows the direct data integration into models.

Standards for Structuring Models of Biological Systems

Model representation formats standardize the structured encoding of biological models. Examples mainly used for models consisting of molecular entities and describing their interactions and dynamical interplay include:

- SBML (Systems Biology Markup Language) as standardized interchange format for computer models of biological processes (Hucka *et al.*, 2003; Hucka *et al.*, 2018).
- CellML, a standard format to store and exchange reusable, modular computer-based mathematical models (Cuellar *et al.*, 2003).
- NeuroML (Neural Open Markup Language), which allows standardization of model descriptions in computational neuroscience (Gleeson *et al.*, 2010).

All three are based on XML (Extensible Markup Language) (see “Relevant Websites section”), a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. These formats represent the structure of a model (e.g., the biochemical network) and enable annotation of the model to better convey a model's intention. Here the focus will be on two of the most widely used standards, SBML and CellML.

SBML (Systems Biology Markup Language)

The Systems Biology Markup Language (SBML) (Hucka *et al.*, 2003) is a machine-readable representation format for computational models in systems biology. This format began its development in 2000 as a way of encoding models for the exchange between modeling software platforms and ensuring longevity of the models beyond the software used to create it. The evolution of SBML proceeds in stages in which each “Level” is an attempt to achieve a consistent language at a certain level of complexity. The current version of SBML is Level 3 Version 2 (Hucka *et al.*, 2018). SBML Level 3 is modular, with the core usable in its own right and Level 3 packages being additional “layers” that add features to the core. By itself, core SBML Level 3 is well suited to representing such things as classical metabolic models and cell signaling models, involving well-mixed substances and spatially homogeneous compartments where they are located. Other model types can also be expressed using SBML's core constructs, but SBML Level 3 packages that extend the core and are optional in their use, add more natural support for such types and additional model features, such as visualizations (Gauges *et al.*, 2015), constraint-based models (Olivier and Bergmann, 2015; Olivier and Bergmann, 2018), hierarchical model composition (Smith *et al.*, 2015), or grouping of elements (Hucka and Smith, 2016).

SBML models are decomposed into explicitly labeled constituent elements. A valid model may consist of various user-defined elements, for example, substances, products and modifiers involved in processes, or compartments where they are located. Models are not cast directly into a specific form such as differential equations. Many SBML models represent biochemical reaction networks by reactants, called *species* in SBML, that participate in *reactions*, as a substrate, a product, or a modifier (e.g., a *catalyst* of an enzyme catalyzed reaction or an *inhibitor* or *activator*) and that are part of a *compartment*, for example, a cellular substructure or a defined cell. Mathematical descriptions of the reactions within the model, such as the kinetic rate equations or kinetic constants,

are formed using *functions*, *units*, *parameters*, *assignments*, and *kinetic definitions*. The mathematical system can be enhanced using *constraints*, *rules*, and *events*, which can mimic biological phenomena such as feed-source constraints, fixed substrate ratios if needed (e.g., ATP, ADP, AMP balances), nutrient pulse experiments, and more. Every major element carries an identifier, and MathML (see “Relevant Websites section”) is used to encode mathematical descriptions of reactions.

In 2010 the SBML community developed an annotation scheme (Hucka *et al.*, 2015a; Hucka *et al.*, 2018) based on the Resource Description Format (RDF (Lassila and Swick, 1999)) of the guideline standard “Minimum information requested in the annotation of biochemical models” (MIRIAM (Le Novère *et al.*, 2005)) and also based on unambiguous identifiers retrieved from the corresponding registry (Juty *et al.*, 2012). This annotation framework enables users to describe their models, and include standard identifiers, so that models are more reusable and exchangeable. SBML is supported by hundreds of different software tools for reading, writing, simulating, editing, and otherwise working with models encoded in the standard, as well as by many databases and other resources (see “Relevant Websites section”).

CellML

CellML (Cuellar *et al.*, 2003) is a description language to define models of cellular and subcellular processes. At its core, CellML defines lightweight XML constructs that group mathematical relationships within modules. The variables used in mathematics are defined within each module and connections between variables in different modules can be specified. It began development in 1998, completely independently of SBML, in order to support the construction and exchange of cardiac cell models. The first stable specification for CellML 1.0 was released in 2001, at the time of writing the current stable version is CellML 1.1 (Cuellar *et al.*, 2003), but a public draft version 2.0 has been available since May 2017.

Primarily used for describing systems of differential algebraic equations, CellML provides the syntactic constructs to describe mathematical equations in a modular framework called components. One of the key defining features of CellML is its ability to support component-based modeling, allowing models to import other models, or subparts of models, therefore strongly encouraging their reuse (Lloyd *et al.*, 2004; Wimalaratne *et al.*, 2009a) and facilitating a modularized modeling approach. A CellML model typically consists of components, which may contain variables and mathematics that describe the behavior of that component. CellML provides means to reuse and group components into hierarchical structures. Similar to SBML, all entities (elements) carry an identifier and mathematical definitions are encoded using MathML. The mathematical model is considered to be the primary data and biological context is provided by annotating the variables and equations with metadata using Resource Description Format (RDF (Lassila and Swick, 1999)). All numerical values and variables used in a CellML document are required to unambiguously define their physical units that can be defined document-wide, or specifically for certain components. Due to the requirement for physical units, numerical quantities can vary between modules and software is expected to convert units automatically. A CellML metadata standard allows for biological and biophysical annotation of the models (Beard *et al.*, 2009; Wimalaratne *et al.*, 2009b). A wide range of software supports CellML (Garny *et al.*, 2008; Wimalaratne *et al.*, 2009a) and includes software for modeling, visualization, simulation, validation, and conversion. All the software tools can be viewed on the CellML website (see “Relevant Websites section”).

Both CellML and SBML facilitate semantic annotations (e.g., links to controlled vocabularies and ontologies), which further describe the biological meaning of single certain elements (Courtot *et al.*, 2011). Controlled vocabularies and ontologies are detailed later in this article. Neither CellML nor SBML encode what is done with a model nor the results of doing something with it (e.g., simulating it) – these are aspects addressed by other standards such as SED-ML (see below).

Standard for Visualizing Models: The Systems Biology Graphical Notation

When it comes to visualizing networks, each researcher has their own ideas of how the different biological components should be illustrated. The style of diagram can be affected by many factors, including the researchers’ backgrounds, and what elements of the diagram they want to emphasize for their work. This creates bias, and makes creating useful visuals challenging; particularly because each researcher also interprets these figures differently (Kitano *et al.*, 2005). In the same way that electronic circuit diagrams have been standardized in electrical engineering for many decades, consistency and uniformity can be brought to the graphical expression of network diagrams in biology by defining common standards for visualization. The Systems Biology Graphical Notation (SBGN (Le Novère *et al.*, 2009)) standardizes the visual notation used to depict biological networks and processes. The use of such a standard visual notation is vital to ensure that diagrams are unambiguous and consistent. SBGN was first developed in 2006, and is comprised of three languages, which allow biological networks to be viewed from different perspectives and at different levels of detail:

- (1) The process description (PD) variant visualizes biochemical interactions as processes over time (Moodie *et al.*, 2015). As the process map extends from left to right, time evolves, allowing the exact order of reactions to be represented. It is normal in this diagram for the same entity to appear multiple times. PD can describe each process in a network in great detail (e.g., biochemical reaction, binding/unbinding of proteins, and the like) and is useful to represent chemical kinetics models. However, some biological phenomena entail a combinatorial explosion of possible interrelated states, making them extremely difficult to depict at this level of detail (Fig. 1).

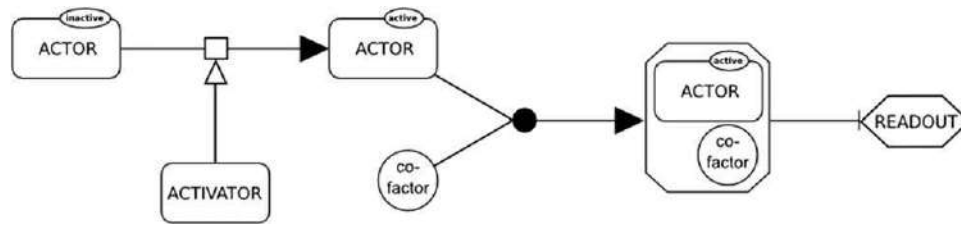


Fig. 1 Example of a SBGN process description map. provided by Nicolas Le Novère, UK.

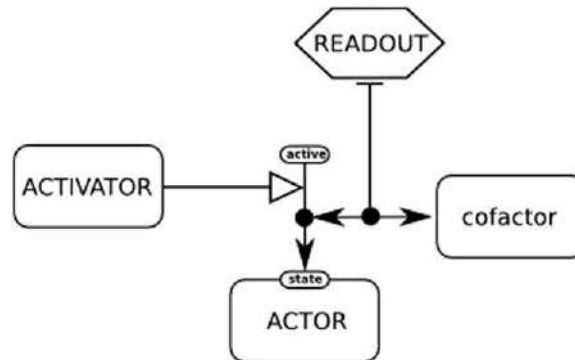


Fig. 2 Example of a SBGN entity relationship map. provided by Nicolas Le Novère, UK.



Fig. 3 Example of a SBGN activity flow map. provided by Nicolas Le Novère, UK.

- (2) The entity relationship (ER) variant visualizes all interactions and relationships between entities, for a given entity (Sorokin *et al.*, 2015). Thereby ER abstracts away the notion of time and focuses on depicting only the relationships between elements, independent of each other. It is useful to represent rule-based models, among other things (Fig. 2).
- (3) The activity flow (AF) variant visualizes the flow of information between biological entities (Mi *et al.*, 2015). So, AF maps focus on the influences between elements rather than the actual processes and are especially useful for representing qualitative models (Fig. 3).

In each instance, a standardized set of entities, and relations visualize the model in a so-called map. A standardized markup language (SBGN-ML) can be used to independently encode the map (van Iersel *et al.*, 2012). SBGN can be used to visualize data and models in SBML and CellML formats, among others.

Standard for Recording Simulation Environments and Setups: The Simulation Experiment Description Markup Language

Consistent and standardized descriptions and formatting of models are not the only requirements for improved reproducible modeling results. Published models are often validated using a set of simulations, and results published from the model are the outcome of a set of virtual experiments. The number of options available to simulate models in different ways, however, can range from diverse algorithms that can run, for example, time courses, steady-states, metabolic control analysis, parameter sensitivities and others, to fixing of various cellular concentrations as precondition, or introducing stepwise changes of variables at specific time-points. This information about a simulation setup can be captured in a method description, but mistakes can be made in recording what steps were taken, and in reimplementing the steps. The Simulation Experiment Description Markup Language (SED-ML (Waltemath *et al.*, 2011b)) was designed to record such descriptive information necessary to rerun a model, such that it can be exported from one simulation tool, and imported into another. These standardized descriptions ensure that virtual experiments, when applied to a computational model, reproduce a given result. So, SED-ML allows the exact simulation setup to be configured and rerun. The first official version was released in 2011. Similarly to SBML, SED-ML evolves in Levels and Versions. The standard is not specific to any simulation software, or modeling format. It is widely used to exchange simulation experiments in computational biology (Olivier and Bergmann, 2015).

SED-ML files record information related to the checklist that describes the minimum information required to understand and reproduce a simulation study, MIASE (Minimum Information About a Simulation Experiment (Waltemath *et al.*, 2011a)). This typically consists of five major blocks of information (Waltemath *et al.*, 2011a; Bergmann *et al.*, 2015):

- (1) Reference to the models being used in the simulation;
- (2) Descriptions of modifications applied to the model before simulation (e.g., the initialization of the variables);
- (3) Descriptions of the simulation steps, including the configuration of the software tool or numerical algorithm;
- (4) Descriptions of the post-processing of result data after simulation; and
- (5) Specifications of the results, including definition of plots and numerical reports.

Libraries to read and write SED-ML are provided by the community and some software tools already consume and export SED-ML files (see “Relevant Websites section”), for example, COPASI (Hoops *et al.*, 2006) or JWS Online (Olivier and Snoep, 2004; Snoep and Olivier, 2003), or Tellurium (Choi *et al.*, 2016). SED-ML elements can also be linked to semantic annotations (Courtot *et al.*, 2011). Simulation Experiment Description Markup Language files can be linked to model descriptions in other formats, notably SBML or CellML, to ensure reproducibility of experiments presented in scientific publications. Such links can, for example, be instantiated via the provision of files in a COMBINE archive (Bergmann *et al.*, 2014), or through provision via public model repositories such as BioModels Database (Li *et al.*, 2010) or the Physiome Repository (Yu *et al.*, 2011).

Recording Modeling Results in a Standardized Way

Similarly to models and simulation setups, communicating the results for *in silico* modeling experiments for exchange, validation, and reuse can be difficult. Results of modeling in the life sciences typically include numerical values, which may also be turned into figures. Numerical results are usually tables or matrices, encoded in CSV (comma-separated values) (Shafranovich, 2005). The flexibility of CSV files means that software and tooling that produces the data can structure and format the data in many different orientations. However, the same data are often structured and labeled differently by each software. This variation hampers ease of exchange, validation, and reuse. Early, generic approaches to providing structure and schema definitions to CSV-like files includes fielded text (Klink, 2016), which is XML-based annotations that describe the header fields of a CSV file. Whilst useful, the amount of life science-specific information that can be used to describe the data is limited. To improve the level of annotation, and to ensure consistent formatting of results data, the Systems Biology Results Mark-up Language (SBRML (Dada *et al.*, 2010)) was developed in 2010. It is specialized for encoding simulation results obtained by running an SBML model. Semantic annotations can be included, with SBRML able to include terms from any ontology that may be relevant to the results. Developing software to analyze the results is also improved by the consistent data structure that can be expected, improving validation and reuse of the data. SBRML is currently being used as a basis to develop a more general results exchanging format NuML (Numerical Markup Language) (see “Relevant Websites section”). The associated library libNUML provides software support for reading, writing, and manipulating data in NuML format on all operating systems (see “Relevant Websites section”).

Metadata, Controlled Vocabularies, and Ontologies for Adding Semantic Information

Sustainable model reuse requires a basic understanding of (1) the biological background, (2) the modeled system, and (3) possible parameterizations under different conditions (Scharm and Waltemath, 2016). This knowledge can be transferred to end-users and computers by using metadata in the form of semantic annotations. Metadata is data about data – it clarifies the intended semantics of the biological data, improving understanding of their scope and validity. Through this it improves the shareability and interoperability of the model (Gennari *et al.*, 2011) – especially computationally. Machine-readable annotations can automate exchange, validation, reuse, composition of models and to convert machine-readable code into human-readable formats (Finney *et al.*, 2006; Misirli *et al.*, 2016; Swainston and Mendes, 2009; Rodriguez *et al.*, 2016). The semantic layer can also be exploited to convert machine-readable code into human-readable formats, such as PDF documents (Dräger *et al.*, 2009; Shen *et al.*, 2010) or visualizations (Junker *et al.*, 2012; Shen *et al.*, 2010) to aid human comprehension.

The metadata can comprise of many things, from free text descriptions, through to inclusion of more formal representations of knowledge like controlled vocabularies and ontologies (Rosse and Mejino, 2003; Bard and Rhee, 2004). Controlled vocabularies are an organized arrangement of precise terms and definitions for a given research domain. They work similarly to a taxonomy system or hierarchical classifications, defining given objects by *is_a* relationships e.g., ethanol *is_a* primary alcohol. Ontologies are similar to controlled vocabularies, but define more comprehensive relationships between objects (e.g., *part_of*, *has_role*). Along with other descriptions, these comprise the information that would be used to semantically annotate data and models, promoting their reuse (Misirli *et al.*, 2016). Terms from controlled vocabularies and ontologies are linked to the entities of a project using semantic technologies, such as the Resource Description Framework (see “Relevant Websites section”) (Lassila and Swick, 1999). There is software available for embedding semantic annotations within spreadsheets, improving semantic adoption among laboratory scientists (Wolstencroft *et al.*, 2011; Maguire *et al.*, 2013).

Especially important ontologies in the domain of modeling in systems biology are (see “Relevant Websites section”):

- Systems Biology Ontology (SBO) (Courtot *et al.*, 2011) designed for models in the domain of systems biology,
- Kinetic Simulation Algorithm Ontology (KiSAO) (Courtot *et al.*, 2011) designed for specifying simulation algorithms and their parameterizations,

- Terminology for the Description of DYnamics (TEDDY) (Courtot *et al.*, 2011) designed for dynamical behaviors and results,
- Just Enough Results Model (JERM) (Wolstencroft *et al.*, 2013) designed to be a minimal descriptor of key information required for ensuring reproducibility of systems biology experiments.

There are also a number of other ontologies and controlled vocabularies that are commonly used within computational systems biology for describing the content and environmental context of models. These include for example, the Gene Ontology (Ashburner *et al.*, 2000), which provides information on genes and their molecular function; the NCBI Taxonomy (Federhen, 2012), which provides information about the nomenclature of organisms; the Protein Ontology (Natale *et al.*, 2014), which provides information about proteins; ChEBI (Degtyarenko *et al.*, 2008), which provides information about chemical compounds of biological interest. In addition, using the vCard ontology (Iannella and McKinney, 2014) it is possible to relate to people and organizations. Using the COMODI ontology it is also possible to encode knowledge about differences between computational models and versions thereof (Scharm *et al.*, 2016).

There are a number of guidelines and checklists available in order to support researchers in annotating their scientific results. For example, the initiative “Minimum Information for Biological and Biomedical Investigations” (MIBBI (Taylor *et al.*, 2008)) provides a general checklist for results in the domain of the life sciences. More specifically, the guideline on “Minimum information requested in the annotation of biochemical models” (MIRIAM (Le Novère *et al.*, 2005)) provides a checklist specifically entailed for computational models. According to MIRIAM, the description of a computational model requires:

- A valid implementation in an appropriate language,
- An initial parameterization,
- Proper metadata about the model provenance (creators, contributors, and creation and modification dates, terms of distribution, etc.) and references to corresponding publications or similar documentations,
- The reproducibility of the references publication/documentation.

Similarly, the guideline on “Minimum Information About a Simulation Experiment” (MIASE, (Waltemath *et al.*, 2011a)) provides a checklist for simulation descriptions of modeling projects. According to MIASE, a simulation experiment needs to specify:

- A comprehensive model including equations and parameterizations,
- A simulation description including precise description of the simulation steps,
- Anything that is necessary to obtain described results.

Survey and Synopsis of Modeling Formats: The NormSys Registry

We have seen that in systems biology, the consistent structuring and description of data and computer models with their underlying experimental or modeling processes is only possible by applying interoperable standards for formatting and describing data, workflows, and models, as well as the metadata describing the interconnection between all these. However, with hundreds of available standards in the field, it is cumbersome for the modelers as potential users of these standards to get an overview about the available formats and supporting standards, as well as their potential fields of application.

To survey some of the most common freely available standards for model description and exchange used in systems biology and related fields, the NormSys registry (see “Relevant Websites section”) was first released in October 2015. This publicly accessible platform provides a single access point for consistent information about these model-exchange formats and aims at listing them in a synopsis to bundle detailed and coherent information about their major features, as well as their similarities, relationships, and differences. The standards are classified by potential fields of application to provide the users guidance on the application of certain standards for a given task (e.g., a certain modeling approach that needs to be implemented) and representative use-cases give concrete model examples for these tasks (Fig. 4). Moreover, possible translation options from one standard to the other are illustrated, as well as interfacing options between the analyzed modeling standards. The standards covered by the first release of the NormSys registry include:

- SBML (Systems Biology Markup Language) as standardized interchange format for computer models of biological processes;
- SBGN (Systems Biology Graphical Notation) as standard for the graphical description of biological networks and pathways;
- CellML as standard for the exchange of mathematical models in biology and physiology;
- FieldML as declarative standard language for building hierarchical models represented by generalized mathematical fields;
- SED-ML (Simulation Experiment Description Markup Language) as standardized format for encoding simulation setups, to ensure exchangeability and reproducibility of simulation experiments;
- SBOL (Synthetic Biology Open Language) to represent genetic designs through a standardized vocabulary and standardized formats;
- NeuroML as a standardized description language that provides a common data format for defining and exchanging descriptions of neuronal cell and network models; and
- PharmML (Pharmacometrics Markup Language) as exchange format for encoding of models, associated tasks, and their annotation in pharmacometrics.

The NormSys registry also comprises an extension for validation and certification of computer models of biological systems that are described in the standard formats. This NormSys model validator, which was released at the end of 2016, includes both a formal syntax

Biological Application	SBML L3V1 Core	CellML 1.1	SBGN ER L1 V1.2	SBGN PD L1 V1.3	SBGN AF L1 V1.9	MorphML v1.8.1	NeuroML 2 beta.3	PharmML v0.6	SBOL v2.0	SBOL Visual v1.0.0	ChannelML v1.8.1	BiophysML v1.8.1	NetworkML v1.8.1
Multi-organism Process	✓	✓	-	-	-	-	-	-	-	-	-	-	-
Cell Cycle	✓	✓	✓	-	-	-	-	-	-	-	-	-	-
Signaling	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-
Single Cell Morphology	-	-	-	-	-	✓	✓	-	-	-	-	-	-
Pharmacokinetic	✓	✓	-	-	-	-	✓	-	-	-	-	-	-
Pharmacodynamics	✓	✓	-	-	-	-	✓	-	-	-	-	-	-
Izhikevich-based Neuron Models	✓	-	-	-	-	✓	-	-	-	-	-	-	-
Synthetic Gene Regulatory Network	✓	-	✓	✓	✓	-	-	✓	✓	-	-	-	-
Metabolic Process	✓	✓	-	✓	-	-	✓	-	-	-	-	-	-
Immune Response	✓	✓	-	-	✓	-	-	-	-	-	-	-	-
Circadian Rhythm	✓	✓	✓	-	-	✓	-	-	-	-	-	-	-
Regulation of Gene Expression	✓	✓	✓	✓	✓	-	-	✓	✓	-	-	-	-
Electrophysiology	✓	✓	-	-	-	✓	-	-	-	✓	✓	-	-
Synaptic Transmission	✓	-	-	✓	-	✓	✓	-	-	✓	✓	✓	✓
Neuronal Network	✓	✓	-	-	-	✓	-	-	-	-	-	-	✓

Fig. 4 The NormSys registry (<http://normsys.h-its.org>) provides details about modeling standards in systems biology and indicates potential fields of biological application.

check (XML-based) and a semantic check of the models and the integrity and consistency of their content, their entities, and their annotations. For syntax validation, a generic framework was implemented that makes use of the XML schema information on the corresponding standards stored in the NormSys registry. The consistency and validity of model entities and their corresponding annotations are checked for semantic validation. This semantic annotation test of an uploaded model is performed by a compliance check according to the MIRIAM guidelines (Le Novère *et al.*, 2005), which provide a checklist catalog for model annotations in biology ("Minimum information requested in the annotation of biochemical models").

Acknowledgment

The author received funding support for this work from the German Federal Ministry for Economic Affairs and Energy (BMWi) via the NormSys project (grant FKZ 01FS14019), from the German Federal Ministry of Education and Research (BMBF) as part of the Liver Systems Medicine Network (LiSyM, FKZ 031L0056), as well as from the Klaus Tschira Foundation (KTS).

See also: Bioinformatics Data Models, Representation and Storage. Biological and Medical Ontologies: GO and GOA. Cell Modeling and Simulation. Computational Systems Biology Applications. Computational Systems Biology. Data Cleaning. Data Integration and Transformation. Data Storage and Representation. Information Retrieval in Life Sciences. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Multi-Scale Modelling in Biology. Natural Language Processing Approaches in Bioinformatics. Pre-Processing: A Data Preparation Step. Quantitative Modelling Approaches. Standards and Models for Biological Data: BioPax. Standards and Models for Biological Data: Common Formats. Standards and Models for Biological Data: SBML. Studies of Body Systems

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* 25 (1), 25–29. doi:10.1038/75556.
- Bard, J.B., Rhee, S.Y., 2004. Ontologies in biology: Design, applications and future challenges. *Nature Reviews Genetics* 5 (3), 213–222. doi:10.1038/nrg1295.
- Beard, D.A., Britten, R., Cooling, M.T., *et al.*, 2009. CellML metadata standards, associated tools and repositories. *Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences* 367 (1895), 1845–1867. doi:10.1098/rsta.2008.0310.
- Bergmann, F.T., Adams, R., Moodie, S., *et al.*, 2014. One file to share them all: Using the COMBINE archive and the OMEX format to share all information about a modeling project. *BMC Bioinformatics* 15, 369. doi:10.1186/s12859-014-0369-z.
- Bergmann, F.T., Cooper, J., Le Novère, N., Nickerson, D., Waltemath, D., 2015. Simulation experiment description markup language (SED-ML) level 1 version 2. *Journal of Integrative Bioinformatics* 12 (2), 119–212. doi:10.1515/jib-2015-262.
- Brazma, A., Hingamp, P., Quackenbush, J., *et al.*, 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature Genetics* 29 (4), 365–371. doi:10.1038/ng1201-365.
- Choi, K., Medley, J.K., Cannistra, C., *et al.*, 2016. Tellurium: A python based modeling and reproducibility platform for systems biology. *bioRxiv*. doi:10.1101/054601.
- Courtot, M., Juty, N., Knüpfer, C., *et al.*, 2011. Controlled vocabularies and semantics in systems biology. *Molecular Systems Biology* 7, 543. doi:10.1038/msb.2011.77.
- Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., *et al.*, 2003. An overview of CellML 1.1, a biological model description language. *Simulation* 79 (12), 740–747.
- Dada, J.O., Spasic, I., Paton, N.W., Mendes, P., 2010. SBRML: A markup language for associating systems biology data with models. *Bioinformatics* 26 (7), 932–938. doi:10.1093/bioinformatics/btq069.
- Degtyarenko, K., de Matos, P., Ennis, M., *et al.*, 2008. ChEBI: A database and ontology for chemical entities of biological interest. *Nucleic Acids Research* 36 (Database issue), D344–D350. doi:10.1093/nar/gkm791.
- Dräger, A., Planatscher, H., Moutsou Woumba, D., *et al.*, 2009. SBML2L(A)T(E)X: Conversion of SBML files into human-readable reports. *Bioinformatics* 25 (11), 1455–1456. doi:10.1093/bioinformatics/btp170.
- Federhen, S., 2012. The NCBI Taxonomy database. *Nucleic Acids Research* 40 (Database issue), D136–D143. doi:10.1093/nar/gkr1178.
- Field, D., Garrity, G., Gray, T., *et al.*, 2008. The minimum information about a genome sequence (MIGS) specification. *Nature Biotechnology* 26 (5), 541–547. doi:10.1038/nbt1360.
- Finney, A., Hucka, M., Bornstein, B.J., Keating, S.M., Shapiro, B.E., 2006. Software infrastructure for effective communication and reuse of computational models. In: Szallasi, Z., Stelling, J., Periwé, V. (Eds.), *System Modeling in Cell Biology: From Concepts to Nuts and Bolts*. Cambridge, MA: MIT Press, pp. 355–378.
- Garny, A., Nickerson, D.P., Cooper, J., *et al.*, 2008. CellML and associated tools and techniques. *Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences* 366 (1878), 3017–3043. doi:10.1098/rsta.2008.0094.
- Gauges, R., Rost, U., Sahle, S., Wengler, K., Bergmann, F.T., 2015. The systems biology markup language (SBML) level 3 package: Layout, version 1 core. *Journal of Integrative Bioinformatics* 12 (2), 550–602. doi:10.1515/jib-2015-267.
- Gennari, J.H., Neal, M.L., Galdzicki, M., Cook, D.L., 2011. Multiple ontologies in action: Composite annotations for biosimulation models. *Journal of Biomedical Informatics* 44 (1), 146–154. doi:10.1016/j.jbi.2010.06.007.
- Gleeson, P., Crook, S., Cannon, R.C., *et al.*, 2010. NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail. *PLOS Computational Biology* 6 (6), e1000815. doi:10.1371/journal.pcbi.1000815.
- Hoops, S., Sahle, S., Gauges, R., *et al.*, 2006. COPASI – A complex pathway simulator. *Bioinformatics* 22 (24), 3067–3074. doi:10.1093/bioinformatics/btl485.
- Hucka, M., Bergmann, F.T., Dräger, A., *et al.*, 2018. The systems biology markup language (SBML): Language specification for level 3 version 2 core. *Journal of Integrative Bioinformatics* 15, 20170081. doi:10.1515/jib-2017-0081.
- Hucka, M., Bergmann, F.T., Hoops, S., *et al.*, 2015a. The systems biology markup language (SBML): Language specification for level 3 version 1 core. *Journal of Integrative Bioinformatics* 12 (2), 382–549. doi:10.1515/jib-2015-266.
- Hucka, M., Finney, A., Sauro, H.M., *et al.*, 2003. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19 (4), 524–531.
- Hucka, M., Nickerson, D.P., Bader, G.D., *et al.*, 2015b. Promoting coordinated development of community-based information standards for modeling in biology: The COMBINE initiative. *Frontiers in Bioengineering and Biotechnology* 3, 19. doi:10.3389/fbioe.2015.00019.
- Hucka, M., Smith, L.P., 2016. SBML level 3 package: Groups, version 1 release 1. *Journal of Integrative Bioinformatics* 13 (3), 8–29. doi:10.1515/jib-2016-290.
- Iannella, R., McKinney, J., 2014. vCard ontology – For describing people and organizations. W3c. <https://www.w3.org/TR/vcard-rdf/> (accessed 03.04.17).
- Junker, A., Rohn, H., Czaderna, T., *et al.*, 2012. Creating interactive, web-based and data-enriched maps with the systems biology graphical notation. *Nature Protocols* 7 (3), 579–593. doi:10.1038/nprot.2012.002.
- Juty, N., Le Novère, N., Laibe, C., 2012. Identifiers.org and MIRIAM registry: Community resources to provide persistent identification. *Nucleic Acids Research* 40 (Database issue), D580–D586. doi:10.1093/nar/gkr1097.
- Kitano, H., Funahashi, A., Matsuo, Y., Oda, K., 2005. Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology* 23 (8), 961–966. doi:10.1038/nbt1111.
- Klink, P. (2016) FieldedText. Available at: <http://www.fieldedtext.org/> (accessed 01.05.17).
- Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., Schaber, J., 2007. Systems biology standards – The community speaks. *Nature Biotechnology* 25 (4), 390–391. doi:10.1038/nbt0407-390.
- Lassila O., Swick R.R., 1999. Resource Description Framework (RDF) model and syntax specification. Technical report. World Wide Web Consortium.
- Li, C., Donizelli, M., Rodriguez, N., *et al.*, 2010. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology* 4, 92. doi:10.1186/1752-0509-4-92.
- Lloyd, C.M., Halstead, M.D., Nielsen, P.F., 2004. CellML: Its future, present and past. *Progress in Biophysics & Molecular Biology* 85 (2–3), 433–450. doi:10.1016/j.pbiomolbio.2004.01.004.
- Maguire, E., Gonzalez-Beltran, A., Whetzel, P.L., Sansone, S.A., Rocca-Serra, P., 2013. OntoMaton: A bioportal powered ontology widget for google spreadsheets. *Bioinformatics* 29 (4), 525–527. doi:10.1093/bioinformatics/bts718.
- Misirli, G., Cavaliere, M., Waites, W., *et al.*, 2016. Annotation of rule-based models with formal semantics to enable creation, analysis, reuse and visualization. *Bioinformatics* 32 (6), 908–917. doi:10.1093/bioinformatics/btv660.
- Mi, H., Schreiber, F., Moodie, S., *et al.*, 2015. Systems biology graphical notation: Activity flow language level 1 version 1.2. *Journal of Integrative Bioinformatics* 12 (2), 340–381. doi:10.1515/jib-2015-265.
- Moodie, S., Le Novère, N., Demir, E., Mi, H., Villeger, A., 2015. Systems biology graphical notation: Process description language level 1 version 1.3. *Journal of Integrative Bioinformatics* 12 (2), 213–280. doi:10.1515/jib-2015-263.
- Myers C.J., Bader G.D., Gleeson P., *et al.*, 2017. A brief history of COMBINE, 2017 Winter Simulation Conference (WSC), pp. 884–895. Las Vegas, NV, USA. doi: 10.1109/WSC.2017.8247840.
- Natale, D.A., Arighi, C.N., Blake, J.A., *et al.*, 2014. Protein ontology: A controlled structured network of protein entities. *Nucleic Acids Research* 42 (Database issue), D415–D421. doi:10.1093/nar/gkt1173.

- Le Novère, N., Finney, A., Hucka, M., *et al.*, 2005. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature Biotechnology* 23 (12), 1509–1515. doi:10.1038/nbt1156.
- Le Novère, N., Hucka, M., Mi, H., *et al.*, 2009. The systems biology graphical notation. *Nature Biotechnology* 27 (8), 735–741. doi:10.1038/nbt.1558.
- Olivier, B.G., Bergmann, F.T., 2015. The Systems Biology Markup Language (SBML) level 3 package: Flux balance constraints. *Journal of Integrative Bioinformatics* 12 (2), 660–690. doi:10.1515/jib-2015-269.
- Olivier, B.G., Bergmann, F.T., 2018. SBML level 3 package: Flux balance constraints version 2. *Journal of Integrative Bioinformatics* 15 (1), 20170082. doi:10.1515/jib-2017-0082.
- Olivier, B.G., Snoep, J.L., 2004. Web-based kinetic modelling using JWS online. *Bioinformatics* 20 (13), 2143–2144. doi:10.1093/bioinformatics/bth200.
- Rodriguez, N., Pettit, J.B., Dalle Pezze, P., Li, L., *et al.*, 2016. The systems biology format converter. *BMC Bioinformatics* 17, 154. doi:10.1186/s12859-016-1000-2.
- Rosse, C., Mejino Jr., J.L., 2003. A reference ontology for biomedical informatics: The foundational model of anatomy. *Journal of Biomedical Informatics* 36 (6), 478–500. doi:10.1016/j.jbi.2003.11.007.
- Scharm, M., Waltemath, D., 2016. A fully featured COMBINE archive of a simulation study on syncytial mitotic cycles in *Drosophila* embryos. *F1000 Research* 5, 2421. doi:10.12688/f1000research.9379.1.
- Scharm, M., Waltemath, D., Mendes, P., Wolkenhauer, O., 2016. COMODI: An ontology to characterise differences in versions of computational models in biology. *Journal of Biomedical Semantics* 7 (1), 46. doi:10.1186/s13326-016-0080-2.
- Schreiber, F., Bader, G.D., Gleeson, P., *et al.*, 2016. Specifications of standards in systems and synthetic biology: Status and developments in 2016. *Journal of Integrative Bioinformatics* 13, 289. doi:10.2390/biecoll-jib-2016-289.
- Schreiber, F., Bader, G., Gleeson, P., *et al.*, 2018. Specifications of standards in systems and synthetic biology: Status and developments in 2017. *Journal of Integrative Bioinformatics* 15 (1), 20180013 https://doi.org/10.1515/jib-2018-0013.
- Schreiber, F., Bader, G.D., Golebiewski, M., *et al.*, 2015. Specifications of standards in systems and synthetic biology. *Journal of Integrative Bioinformatics* 12, 258. doi:10.2390/biecoll-jib-2015-258.
- SeckM.D., Evans, D.D., 2004. Major U.S. cities using national standard fire hydrants, one century after the great Baltimore fire, vol. 7158, pp. 7–9. Gaithersburg: National Institute of Standards and Technology, NISTIR.
- ShafraiovichY., 2005. Common format and MIME type for comma-separated values (CSV) Files. IETF p. 1 (RFC 4180). The Internet Society. doi:10.17487/RFC4180.
- Shen, S.Y., Bergmann, F., Sauro, H.M., 2010. SBML2tikZ: Supporting the SBML render extension in LaTeX. *Bioinformatics* 26 (21), 2794–2795. doi:10.1093/bioinformatics/btq512.
- Smith, L.P., Hucka, M., Hoops, S., *et al.*, 2015. SBML level 3 package: Hierarchical model composition, version 1 release 3. *Journal of Integrative Bioinformatics* 12 (2), 603–659. doi:10.1515/jib-2015-268.
- Snoep, J.L., Olivier, B.G., 2003. JWS online cellular systems modelling and microbiology. *Microbiology* 149 (Pt 11), 3045–3047. doi:10.1099/mic.0.C0124-0.
- Sorokin, A., Le Novère, N., Luna, A., *et al.*, 2015. Systems biology graphical notation: Entity relationship language level 1 version 2. *Journal of Integrative Bioinformatics* 12 (2), 281–339. doi:10.1515/jib-2015-264.
- Stanford, N.J., Wolstencroft, K., Golebiewski, M., *et al.*, 2015. The evolution of standards and data management practices in systems biology. *Molecular Systems Biology* 11, 851. doi:10.15252/msb.20156053.
- Stromback, L., Hall, D., Lambrix, P., 2007. A review of standards for data exchange within systems biology. *Proteomics* 7 (6), 857–867. doi:10.1002/pmic.200600438.
- Swainston, N., Golebiewski, M., Messiha, H.L., *et al.*, 2010. Enzyme kinetics informatics: From instrument to browser. *FEBS Journal* 277, 3769–3779. doi:10.1111/j.1742-4658.2010.07778.x.
- Swainston, N., Mendes, P., 2009. libAnnotationSBML: A library for exploiting SBML annotations. *Bioinformatics* 25 (17), 2292–2293. doi:10.1093/bioinformatics/btp392.
- Taylor, C.F., Field, D., Sansone, S.A., *et al.*, 2008. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: The MIBBI project. *Nature Biotechnology* 26 (8), 889–896. doi:10.1038/nbt.1411.
- Taylor, C.F., Paton, N.W., Lilley, K.S., *et al.*, 2007. The minimum information about a proteomics experiment (MIAPE). *Nature Biotechnology* 25 (8), 887–893. doi:10.1038/nbt1329.
- van Iersel, M.P., Villeger, A.C., Czauderna, T., *et al.*, 2012. Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics* 28 (15), 2016–2021. doi:10.1093/bioinformatics/bts270.
- Waltemath, D., Adams, R., Beard, D.A., *et al.*, 2011a. Minimum information about a simulation experiment (MIASE). *PLOS Computational Biology* 7 (4), e1001122. doi:10.1371/journal.pcbi.1001122.
- Waltemath, D., Adams, R., Bergmann, F.T., *et al.*, 2011b. Reproducible computational biology experiments with SED-ML – The simulation experiment description markup language. *BMC Systems Biology* 5, 198. doi:10.1186/1752-0509-5-198.
- Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., *et al.*, 2016. Comment: The FAIR guiding principles for scientific data management and stewardship. *Scientific Data* 3. doi:10.1038/sdata.2016.18.
- Wimalaratne, S.M., Halstead, M.D., Lloyd, C.M., *et al.*, 2009a. Facilitating modularity and reuse: Guidelines for structuring CellML 1.1 models by isolating common biophysical concepts. *Experimental Physiology* 94 (5), 472–485. doi:10.1113/expphysiol.2008.045161.
- Wimalaratne, S.M., Halstead, M.D., Lloyd, C.M., Crampin, E.J., Nielsen, P.F., 2009b. Biophysical annotation and representation of CellML models. *Bioinformatics* 25 (17), 2263–2270. doi:10.1093/bioinformatics/btp391.
- Wittig, U., Kania, R., Golebiewski, M., *et al.*, 2012. SABIO-RK – Database for biochemical reaction kinetics. *Nucleic Acids Research* 40, D790–D796. doi:10.1093/nar/gkr1046.
- Wittig, U., Rey, M., Kania, R., *et al.*, 2014. Challenges for an enzymatic reaction kinetics database. *FEBS Journal* 281, 572–582. doi:10.1111/febs.12562.
- Wolstencroft, K., Owen, S., Horridge, M., *et al.*, 2011. RightField: Embedding ontology annotation in spreadsheets. *Bioinformatics* 27 (14), 2021–2022. doi:10.1093/bioinformatics/btr312.
- Wolstencroft K., Owen S., Krebs O., *et al.*, 2013. Semantic Data and Models Sharing in Systems Biology: The Just Enough Results Model and the SEEK Platform. In: Alani H., Kagal L., Fokoue A *et al.* (Eds.), *The Semantic Web – ISWC 2013. Lecture Notes in Computer Science*, vol. 8219, 212–227. Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-41338-4_14
- Yu, T., Lloyd, C.M., Nickerson, D.P., *et al.*, 2011. The physiome model repository 2. *Bioinformatics* 27 (5), 743–744. doi:10.1093/bioinformatics/btq723.

Relevant Websites

<http://co.mbine.org/>
 COMBINE.
<https://www.cellml.org/tools>
 CellML tools
 CellML.
<https://www.w3.org/XML/>
 Extensible Markup Language (XML) I World Wide Web Consortium (W3C).

<https://fairsharing.org/collection/MIBBI>
FAIRsharing MIBBI Collection.

<https://fair-dom.org/>
FAIRDOM.

<https://fairsharing.org/collection/FAIRDOM>
FAIRsharing Collection: FAIRDOM Community Standards.

<http://project.isbe.eu/>
ISBE Project Website.

<http://jermontology.org/>
JERM Ontology.

<http://co.mbine.org/standards/kisao>
Kinetic Simulation Algorithm Ontology | COMBINE.

<https://www.w3.org/Math/>
MathML | World Wide Web Consortium (W3C).

<https://github.com/numl/numl>
NuML · GitHub.

<https://github.com/NuML/NuML/tree/master/libnuml>
NuML/libnuml at master · NuML/NuML · GitHub.

<http://normsys.h-its.org>
Normsys Standard Registry | HITS gGmbH.

<https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>
RDF 1.1 Concepts and Abstract Syntax | World Wide Web Consortium (W3C).

<http://sabiork.h-its.org>
SABIO-RK Biochemical Reaction Kinetics Database | HITS gGmbH.

http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix
SBML Software Guide/SBML Software Matrix | Systems Biology Markup Language (SBML).

<https://sed-ml.github.io/showcase.html>
SED-ML Tools and Libraries.

<https://www.ebi.ac.uk/sbo/main/>
SBO (Systems Biology Ontology) | EMBL-EBI.

<http://co.mbine.org/standards/teddy/>
TEDDY-Ontology | COMBINE.

Computational Lipidomics

Josch K Pauling, Humboldt University of Berlin, Berlin, Germany

© 2019 Elsevier Inc. All rights reserved.

Introduction

Lipidomics, the study of the lipidome, is a rather new member of the omics family (Wenk, 2005; Dennis, 2009). A first European initiative with a focus on lipidomics started in 2005/2006 (van Meer *et al.*, 2007). The term *cellular lipidome* refers to the full lipid complement of a cell (van Meer, 2005). It first appeared in a publication by Han and Gross (2003) where the authors describe a mass spectrometry-based method to detect and quantify all lipids in a cell extract, the lipidome. Even though lipids had been known and measured for over three centuries, the ability to now measure the entire lipidome was a great advancement from traditional approaches that focused on efficiently extracting and measuring a single targeted lipid class or species (Hsu and Turk, 2000a,b, 2002, 2003, 2008; Liebisch *et al.*, 2006; Deems *et al.*, 2007; Krank *et al.*, 2007) often in a specific biochemical or biomedical context. Lipids display a great diversity in chemical structure and have distinct chemical properties (van Meer, 2005) and thus measuring lipidomes is a complex task that requires suitable extraction protocols (Folch *et al.*, 1957; Bligh and Dyer, 1959), ionization (Fenn *et al.*, 1989; Ho *et al.*, 2003), detection and quantification techniques (Sandhoff *et al.*, 1999; Ivanova *et al.*, 2007; Almeida *et al.*, 2015), algorithms to automatically read and process mass spectra (Niemelä *et al.*, 2009; Husen *et al.*, 2013; Herzog *et al.*, 2013; Marella *et al.*, 2015; Kochen *et al.*, 2016; Peng and Ahrends, 2016) (explained throughout this article), nomenclature rules (IUPAC-IUB (CBN), 1967, 1977; IUPAC-IUB (JCBN), 1987, 1989a,b; Moss, 1996; Liebisch *et al.*, 2013), databases to store information (Fahy *et al.*, 2005, 2007, 2009; Wishart *et al.*, 2007; Sud *et al.*, 2007; Schmelzer *et al.*, 2007; Foster *et al.*, 2013; Aimo *et al.*, 2015), and software to link and mine this information (Hadadi *et al.*, 2014). Hence, lipidomics research integrates a wide range of scientific disciplines such as biochemistry, mass spectrometry, bioinformatics, computational biology, biomedicine and biophysics.

Lipids and Functions

Cellular lipids are biomolecules with a very high structural and functional diversity (Gross and Han, 2011). Glycerophospholipids, for example, consist of a hydrophilic head group and two hydrophobic hydrocarbon tails. Together with sphingolipids and cholesterol they can create lipid bilayers (Edidin, 2003; van Meer *et al.*, 2008) that separate two aqueous environments from each other. Within a cell these lipid bilayers act as biological membranes creating cellular compartments. A membrane is the site of metabolic exchange between the organelle and the extra-organelle environment, for example, the cytosol with membrane proteins managing the actual exchange and transport across the membrane. The composition of lipids determines the specific membrane properties such as structure and fluidity. The regulation of membrane-lipid composition is very rigid and slight alterations in it can impair membrane functionality.

Another, and widely known, function of lipids is the storage and provision of cellular energy. Glycerolipids, especially triacylglycerols (TAGs), are storage lipids that serve as carriers of fatty-acids. They consist of a glycerol backbone where fatty-acyl side chains are attached. A TAG consists of three fatty-acyl side chains, which is the maximal number that can be directly attached to a glycerol. Through enzymatic activity these fatty-acyl side chains can be released from the glycerol providing non-esterified fatty-acids (NEFAs), which can serve as a source of ATP when oxidized. There are many more lipids and functions but glycerophospholipids, specifically phosphatidylcholine (PC 34:1 as shown in Fig. 1) will be used in following examples.

Computational Lipidomics

The lipidomics workflow starts with extracting lipids from a cell or tissue sample. Protocols such as the one proposed by Folch *et al.* (1957) or by Bligh and Dyer (1959) have been widely used for decades and the corresponding publications have received

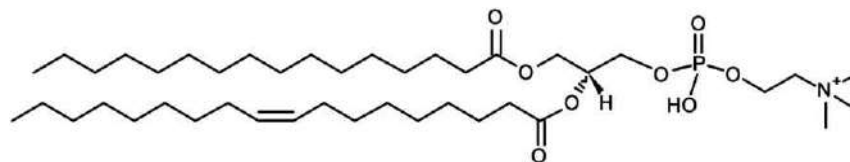


Fig. 1 Chemical structure of a phosphatidylcholine (PC). Together, both fatty-acyls consist of 34 carbon atoms and one double bond (PC 34:1). A shorthand notation of this particular species is PC 16:0/18:1 indicating the position-specific distribution of fatty-acyls. A fatty-acyl with 16 carbon atoms is bound to the stereospecific numbering (sn)-1 position of the glycerol. A fatty-acyl with 18 carbon atoms and one double bond is bound to the sn-2 position of the glycerol. Given that the depicted double bond is of the *cis*-type (Z according to the E-Z-scheme) the full name is 1-hexadecanoyl-2-(9Z-octadecenoyl)-sn-glycero-3-phosphocholine according to IUPAC nomenclature rules.

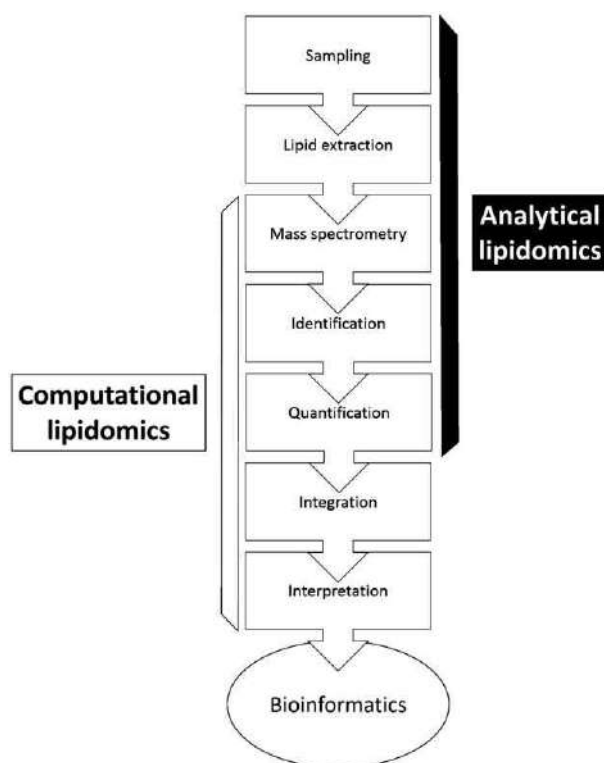


Fig. 2 The lipidomics workflow from a sample to an interpretation in the context of integrated omics analyses. This scheme is described in this article.

over 50,000 and over 40,000 citations, respectively. After extraction, a feasible mix of exogenous lipid standards is selected and added to the extract (Moore *et al.*, 2007). Finally, the sample extract is injected into a mass spectrometer for measurement. There are various mass spectrometric methods but the basic decision is whether or not to apply a chromatographic separation prior to injection, for example, liquid chromatography (LC/MS) or gas chromatography (GC/MS) (McDonald *et al.*, 2007; Sullards *et al.*, 2007; Cruz-Hernandez and Destailats, 2012). The alternative is to inject the sample extract directly into the electrospray ionization (ESI) device of a mass spectrometer. This is called “shotgun lipidomics” (Han, 2003; Han and Gross, 2005; Schwudke *et al.*, 2007, 2011; Schuhmann *et al.*, 2011, 2012).

The path from generating a set of mass spectra to algorithms mining this data and meta data by querying multiple databases is a very long and tedious one (Fig. 2). The first few steps are very technical and require detailed background knowledge of lipid biochemistry and mass spectrometry. Mass spectra representing a lipidome are generated and a readout is stored. Subsequently, the data must be analyzed and prepared in a way that meta data is quickly accessible. This includes quality control mechanisms and statistical analysis. At last there are classical bioinformatics tasks such as database development, data mining and linking the data to other omics data sets, for example, to study health and disease. At this point, the applied bioinformatics methods are not specifically tailored to deal with lipidome data and may therefore be more generally summarized as “lipid bioinformatics”. However, the earlier steps are highly specific to mass spectra of lipidomes and may thus rather be termed “computational lipidomics” which is in the focus of this article.

Mass Spectral Analysis and Peak Detection

High throughput, high resolution lipid experiments may become quite complex when measuring various different tissue samples under various conditions at different time points with biological and technical replicates. This may result in thousands of mass spectra contained in a single data set. Experts can analyze a mass spectrum quickly by looking at it but this method is not feasible for more than about one hundred mass spectra. Automated routines are necessary to analyze such a data set in an adequate time frame. The first step in such a routine is to identify peaks that each resemble lipid molecules of a specific mass per charge ratio (m/z) in a long list of data points which comprise two values (1) m/z value and (2) intensity. Ideally, these data point lists, typically one list per mass spectrum, are kept in computer memory. However, mass spectrometer vendors keep the format in which mass spectra are stored secret and instead provide software tools to convert them to regular tab- or comma-separated text files. Reading and storing the lists in memory is followed by a peak detection heuristic that maps groups of data points to one peak and determines this peak’s centroid m/z and intensity tuple. It is important to note that this is not always a straight forward process as

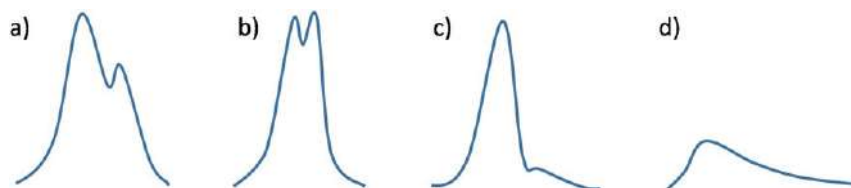


Fig. 3 Various schemata of peak appearances in a mass spectrum. (a) through (c) may depict two or more convoluted peaks. The higher the resolution of the mass spectrometer the more molecules are fully resolved as singular peaks.

shown in Fig. 3. In most of the depicted peak schemata the peak is not resembling a clean Gaussian distribution of intensities over the peak width and in some cases one may observe multiple peaks that could not be fully resolved. The challenge is to deconvolute overlapping peaks and to estimate the m/z and intensity values for each peak. The apex value of a fitted Gaussian bell curve is a good estimator to start with but more sophisticated solutions are needed when this method fails.

Identification and Quantification of Lipids

Assuming that m/z values and intensities have been assigned to each peak, the m/z value can be utilized to annotate each peak with the name of the lipid it resembles. This identification process is vastly dependent on the accuracy of the assigned m/z value and the calibration of the mass spectrometer at the time of measurement. A shift in m/z values may result in a false mapping to a lipid with a slightly higher or lower m/z value. It is therefore essential in quality control to become aware of and correct for such shifts that may occur over the time of a measurement and can only be detected by measuring control samples in between the actual samples. Additionally, there exist lipids with the same atomic composition but yet different chemical structure. These are isomeric molecules and it is not possible to distinguish one from the other just by the peak m/z value.

After mapping every peak to a lipid species the intensity value is utilized to quantify the full complement of lipids in the sample of interest. Normalization is key to guarantee comparability between several mass spectra and experiments. Prior to injection into the mass spectrometer, a mix of internal standards (non-endogenous lipid species) of specific and exact amounts is added to each sample extract to obtain a relationship between the peak signal and the actual concentration in the sample. Lipid class intensities can then be normalized to the respective lipid standard. Doing so ensures that intensity values from different mass spectra become directly comparable. At the same time, differences in ionization efficiencies between lipid classes are corrected.

In a final step after normalization all detected and identified lipids including their intensities are reassembled to formulate the lipidome of a particular sample. To acquire a complete data set, this procedure is done for each sample measured as a part of an experiment. In case one fills an entire 96-well plate with samples, controls, and blanks this is an enormous effort that requires sophisticated computational support.

The Next Dimension

The previous section provided insights into the computational challenges to solve when measuring a lipidome based on intact lipid molecules (abbreviated as MS or MS1). This measurement is not sufficient to reveal the particular fatty-acyl side chains on one of those lipid species or to resolve isomeric lipids. All one can infer from the m/z value is the atomic composition of the whole molecule but not its structural arrangement. In lipid research it is often of interest to track fatty-acid metabolism and the side chains of membrane glycerophospholipids. To achieve this information it is necessary to break a lipid into smaller fragments, for example, via collision-induced dissociation (CID) (Sleno and Volmer, 2004; Wells and McLuckey, 2005). With this technique (abbreviated as MS2, one fragmentation step) it is possible to dissociate the fatty-acyl side chains and the head group from the glycerol backbone and afterwards measure the m/z values of these fragments. This allows to infer the particular fatty-acyls. Fig. 4 guides through this process. Finally, fragments may be fragmented again to reveal even further structural detail (abbreviated MS3, MS4, etc., or simply MS n , where n denotes the number of fragmentation steps).

The lipid molecule of interest, a PC 34:1, has a m/z value of 760.58 Da. Multiple peaks are shown each representing fragments with lower m/z values. The dissociated part of the molecule did not have a charge and is therefore not represented in the mass spectrum. This is called a neutral loss (NL) while the observable charged fragment is called a product ion. The neutral loss can occur in two slightly different ways leaving the two peaks at 478 and 496 Da. The corresponding compounds are shown above. However, according to the mass difference between the complete lipid molecule and the fragments one can infer that in both cases a fatty-acyl with 18 carbon atoms and one double bond is the missing part of the fragment. Hence, the presence of the sub-species PC 16:0–18:1 can be confirmed. The same procedure can be applied to all peaks in the mass spectrum. Doing so additionally shows the neutral losses of different fatty-acyls (red annotation) revealing the presence of another sub-species PC 16:1–18:0 in which the double bond is found on the shorter fatty-acyl. Fragmentation by collisional dissociation is a powerful tool to detect lipid molecules with a high level of structural detail. However, it too has its limitations since some structural details such as double bond type and position and the position of hydroxyl group modifications cannot directly be detected.

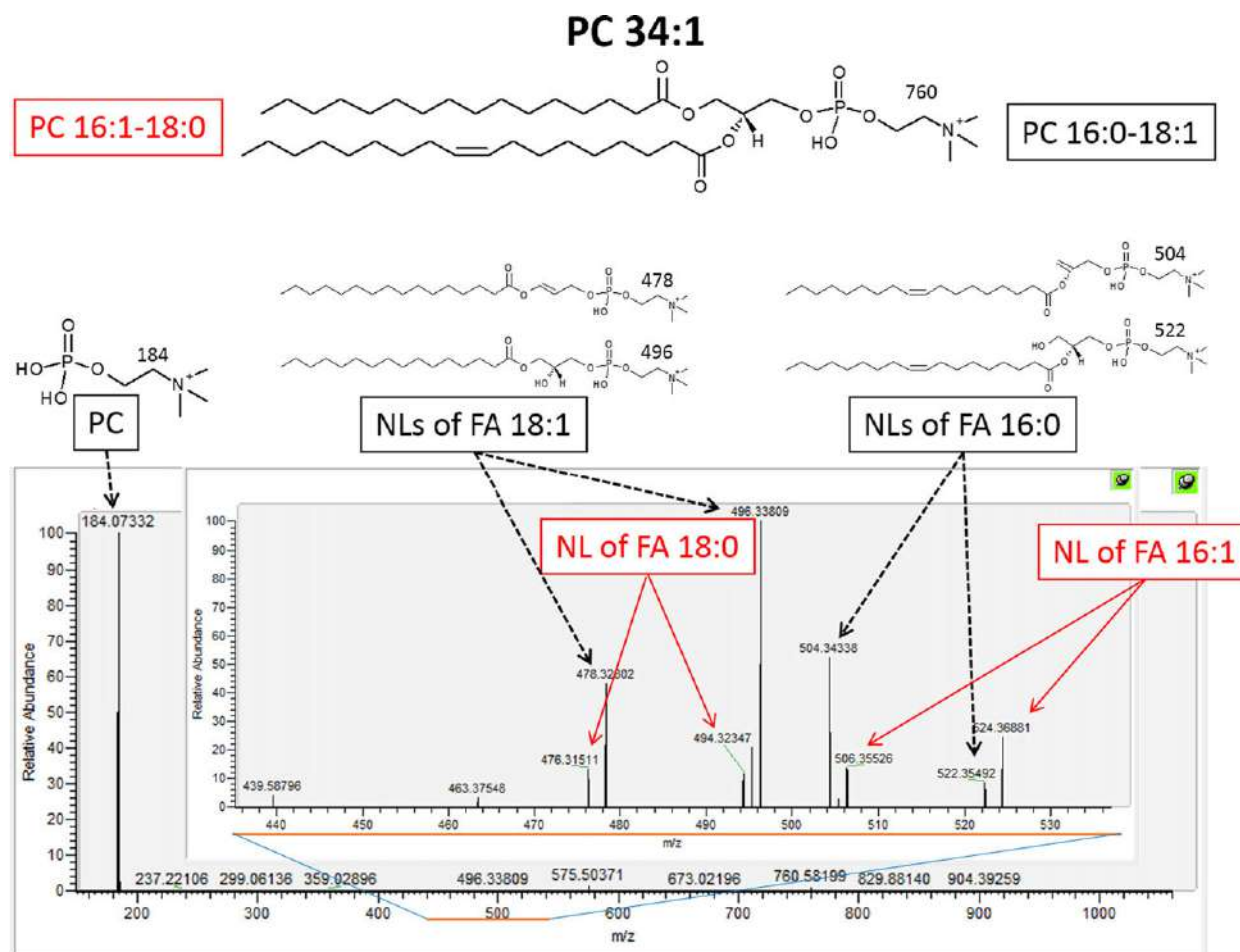


Fig. 4 A raw mass spectrum of fragmented PC 34:1 as it is generated by a mass spectrometer. The spectrum in front depicts a zoom in the m/z range between 440 and 540 Da as indicated by the horizontal, orange lines. The spectrum in the back shows the entire mass range and the prominent peak at 184, a phosphocholine. The chemical structure of this fragment is shown above. The other peaks strongly suggest that there are two detected sub-species (labels in black and red). The corresponding fragments to each peak are shown above. From the evidence one can deduce that there is not only the black labelled molecule PC 16:0–18:1 but also the red labelled PC 16:1–18:0. These are molecules with different chemical structure but with the same atomic composition and m/z value. For simplicity, the structure of the red labelled PC 16:1–18:0 is not shown. On top of the figure only the black labelled PC 16:0–18:1 is shown. Please also note that the sn-position of the fatty-acyl side chains cannot be deduced from a peak's m/z value. Therefore, the annotation "PC 16:0–18:1" is used instead of the position-specific annotation "PC 16:0/18:1" despite the position-specific graphical structure of a PC 16:0/18:1.

When fragmentation is utilized in an experiment the number of mass spectra is significantly increased from hundreds (one mass spectrum per well on a 96-well plate, double injections) to possibly several thousand. The challenge is to automate the process described in Fig. 4 and to accurately infer the various fatty-acyl side chain compositions for a particular lipid species. Fig. 5 depicts the difference between a MS measurement and a MS2 (or MSn) measurement visualized as a schematic bar plot.

With the progression and advancement of today's analytical capacities from smaller scale lipid studies to large scale high-throughput scans of full lipidomes and from whole molecule identification to a higher degree of structural detail it will continue to be a vital task to co-develop computational support routines. To develop sophisticated solutions it is essential to further expand on interdisciplinary expertise in the lipid-oriented research community. In comparison with other omics technology the range of lipid-centric informatics solutions is currently small and it is therefore not surprising that the few existing ones may not be usable free of charge. It is advisable to acquire a solid basic understanding of how computational biology support has helped to fuel the advancement of the omics landscape. Lipidomics and proteomics are largely mass spectrometry-based disciplines. Thus, computational lipidomics can profit from translational efforts. This, however, requires at least basic expertise in proteomics, lipidomics, and computational biology as well as bioinformatics and potentially additional disciplines. In today's biomedical research, there is already a need for position-specific structural characterization of lipid molecules on a lipidome-wide scale within a multi-omics setting. This underlines the importance of the emerging field of computational lipidomics research.

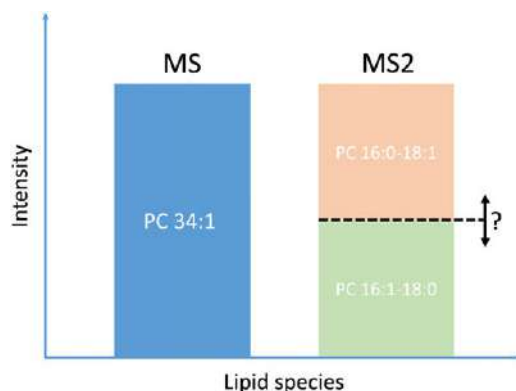


Fig. 5 The levels of detail gained from either a full MS scan or a MS2 up to MS_n fragmentation analysis. The stacked bar on the right visualizes the sub-species composition. The exact quantity of detected sub-species cannot accurately be determined solely based on fragment spectra and must be approximated.

See also: Computational Systems Biology. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Natural Language Processing Approaches in Bioinformatics

References

- Aimo, L., Liechti, R., Hyka-Nouspikel, N., *et al.*, 2015. The swisslipids knowledgebase for lipid biology. *Bioinformatics* 31, 2860–2866.
- Almeida, R., Pauling, J.K., Sokol, E., Hannibal-Bach, H.K., Eising, C.S., 2015. Comprehensive lipidome analysis by shotgun lipidomics on a hybrid quadrupole-orbitrap-linear ion trap mass spectrometer. *Journal of the American Society for Mass Spectrometry* 26, 133–148.
- Bligh, E.G., Dyer, W.J., 1959. A rapid method of total lipid extraction and purification. *Canadian Journal of Biochemistry and Physiology* 37 (8), 911–917.
- Cruz-Hernandez, C., Destailats, F., 2012. Analysis of lipids by gas chromatography. *Gas chromatography*, 529–549. doi:10.1016/B978-0-12-385540-4.00023-7a.
- Deems, R., Buczynski, M.W., Bowers-Gentry, R., Harkewicz, R., Dennis, E.A., 2007. Detection and quantitation of eicosanoids via high performance liquid chromatography-electrospray ionization-mass spectrometry. *Methods in Enzymology* 432, 59–82.
- Dennis, E.A., 2009. Lipidomics joins the omics evolution. *Proceedings of the National Academy of Sciences of the United States of America* 106 (7), 2089–2090.
- Edidin, M., 2003. Lipids on the frontier: A century of cell-membrane bilayers. *Nature Reviews. Molecular Cell Biology* 4, 414–418.
- Fahy, E., Subramaniam, S., Brown, H.A., *et al.*, 2005. A comprehensive classification system for lipids. *Journal of Lipid Research* 46, 839–861.
- Fahy, E., Subramaniam, S., Murphy, R.C., *et al.*, 2009. Update of the lipid maps comprehensive classification system for lipids. *Journal of Lipid Research* 50 (Suppl.), S9–S14.
- Fahy, E., Sud, M., Cotter, D., Subramaniam, S., 2007. Lipid maps online tools for lipid research. *Nucleic Acids Research* 35, W606–W612.
- Fenn, J.B., Mann, M., Meng, C.K., Wong, S.F., Whitehouse, C.M., 1989. Electrospray ionization for mass spectrometry of large biomolecules. *Science* 246, 64–71.
- Folch, J., Lees, M., Sloane Stanley, G.H., 1957. A simple method for the isolation and purification of total lipides from animal tissues. *The Journal of Biological Chemistry* 226, 497–509.
- Foster, J.M., Moreno, P., Fabregat, A., *et al.*, 2013. Lipidhome: A database of theoretical lipids optimized for high throughput mass spectrometry lipidomics. *PLOS ONE* 8, e61951.
- Gross, R.W., Han, X., 2011. Lipidomics at the interface of structure and function in systems biology. *Chemistry & Biology* 18, 284–291.
- Hadadi, N., Cher Soh, K., Seijo, M., *et al.*, 2014. A computational framework for integration of lipidomics data into metabolic pathways. *Metabolic Engineering* 23, 1–8.
- Han, X., Gross, R.W., 2003. Global analyses of cellular lipidomes directly from crude extracts of biological samples by ESI mass spectrometry: A bridge to lipidomics. *The Journal of Lipid Research* 44 (6), 1071–1079.
- Han, X., Gross, R.W., 2005. Shotgun lipidomics: Multidimensional MS analysis of cellular lipidomes. *Expert Review of Proteomics* 2, 253–264.
- Herzog, R., Schwudke, D., Shevchenko, A., 2013. Lipidexplorer: Software for quantitative shotgun lipidomics compatible with multiple mass spectrometry platforms. *Current protocols in bioinformatics* 43 (14), 12.1–14.1230.
- Ho, C.S., Lam, C.W.K., Chan, M.H.M., *et al.*, 2003. Electrospray ionisation mass spectrometry: Principles and clinical applications. *The Clinical Biochemist. Reviews* 24, 3–12.
- Hsu, F.F., Turk, J., 2000a. Charge-driven fragmentation processes in diacyl glycerophosphatidic acids upon low-energy collisional activation. A mechanistic proposal. *Journal of the American Society for Mass Spectrometry* 11, 797–803.
- Hsu, F.F., Turk, J., 2000b. Charge-remote and charge-driven fragmentation processes in diacyl glycerophosphoethanolamine upon low-energy collisional activation: A mechanistic proposal. *Journal of the American Society for Mass Spectrometry* 11, 892–899.
- Hsu, F.F., Turk, J., 2002. Characterization of ceramides by low energy collisional-activated dissociation tandem mass spectrometry with negative-ion electrospray ionization. *Journal of the American Society for Mass Spectrometry* 13, 558–570.
- Hsu, F.F., Turk, J., 2003. Electrospray ionization/tandem quadrupole mass spectrometric studies on phosphatidylcholines: The fragmentation processes. *Journal of the American Society for Mass Spectrometry* 14, 352–363.
- Hsu, F.F., Turk, J., 2008. Structural characterization of unsaturated glycerophospholipids by multiple-stage linear ion-trap mass spectrometry with electrospray ionization. *Journal of the American Society for Mass Spectrometry* 19, 1681–1691.
- Husen, P., Tarasov, K., Katafiasz, M., *et al.*, 2013. Analysis of lipid experiments (alex): A software framework for analysis of high-resolution shotgun lipidomics data. *PLOS ONE* 8, e79736.
- IUPAC-IUB commission on biochemical nomenclature (IUPAC-IUB (CBN)), 1967. The nomenclature of lipids. *European Journal of Biochemistry* 2, 127–131.
- IUPAC-IUB commission on biochemical nomenclature (IUPAC-IUB (CBN)), 1977. The nomenclature of lipids recommendations (1976). *Lipids* 12, 455–468.
- IUPAC-IUB joint commission on biochemical nomenclature (IUPAC-IUB (JCBN)), 1987. Prenol nomenclature recommendations 1986. *European Journal of Biochemistry* 167, 181–184.
- IUPAC-IUB joint commission on biochemical nomenclature (IUPAC-IUB (JCBN)), 1989a. The nomenclature of steroids recommendations 1989. *European Journal of Biochemistry* 186, 429–458.

- IUPAC-IUB Joint Commission on Biochemical Nomenclature (IUPAC-IUB (JCBN)), 1989b. Guidelines on eicosanoid nomenclature. *Eicosanoids* 2, 65–68.
- Ivanova, P.T., Milne, S.B., Byrne, M.O., Xiang, Y., Brown, H.A., 2007. Glycerophospholipid identification and quantitation by electrospray ionization mass spectrometry. *Methods in Enzymology* 432, 21–57.
- Kochen, M.A., Chambers, M.C., Holman, J.D., *et al.*, 2016. Greazy: Open-source software for automated phospholipid tandem mass spectrometry identification. *Analytical Chemistry* 88, 5733–5741.
- Krank, J., Murphy, R.C., Barkley, R.M., Duchoslav, E., McAnoy, A., 2007. Qualitative analysis and quantitative assessment of changes in neutral glycerol lipid molecular species within cells. *Methods in Enzymology* 432, 1–20.
- Liebisch, G., Binder, M., Schifferer, R., *et al.*, 1761. High throughput quantification of cholesterol and cholesteryl ester by electrospray ionization tandem mass spectrometry (esi-ms/ms). *Biochimica et Biophysica Acta* 121–128, 2006.
- Liebisch, G., Vizcaino, J.A., Köfeler, H., *et al.*, 2013. Shorthand notation for lipid structures derived from mass spectrometry. *Journal of Lipid Research* 54, 1523–1530.
- Marella, C., Torda, A.E., Schwudke, D., 2015. The lux score: A metric for lipidome homology. *PLOS Computational Biology* 11, e1004511.
- McDonald, J.G., Thompson, B.M., McCrum, E.C., Russell, D.W., 2007. Extraction and analysis of sterols in biological matrices by high performance liquid chromatography electrospray ionization mass spectrometry. *Methods in Enzymology* 432, 145–170.
- Moore, J.D., Caulfield, W.V., Shaw, W.A., 2007. Quantitation and standardization of lipid internal standards for mass spectroscopy. *Methods in Enzymology* 432, 351–367.
- Moss, G.P., 1996. Basic terminology of stereochemistry (IUPAC recommendations 1996). *Pure and Applied Chemistry* 68 (12).
- Niemelä, P.S., Castillo, S., Sysi-Aho, M., Oresic, M., 2009. Bioinformatics and computational methods for lipidomics. *Journal of Chromatography. B, Analytical Technologies in the Biomedical and Life Sciences* 877, 2855–2862.
- Peng, B., Ahrends, R., 2016. Adaptation of skyline for targeted lipidomics. *Journal of Proteome Research* 15, 291–301.
- Sandhoff, R., Brügger, B., Jeckel, D., Lehmann, W.D., Wieland, F.T., 1999. Determination of cholesterol at the low picomole level by nano-electrospray ionization tandem mass spectrometry. *Journal of Lipid Research* 40, 126–132.
- Schmelzer, K., Fahy, E., Subramaniam, S., Dennis, E.A., 2007. The lipid maps initiative in lipidomics. *Methods in Enzymology* 432, 171–183.
- Schuhmann, K., Almeida, R., Baumert, M., *et al.*, 2012. Shotgun lipidomics on a Itq orbitrap mass spectrometer by successive switching between acquisition polarity modes. *Journal of Mass Spectrometry: JMS* 47, 96–104.
- Schuhmann, K., Herzog, R., Schwudke, D., *et al.*, 2011. Bottom-up shotgun lipidomics by higher energy collisional dissociation on Itq orbitrap mass spectrometers. *Analytical Chemistry* 83, 5480–5487.
- Schwudke, D., Liebisch, G., Herzog, R., Schmitz, G., Shevchenko, A., 2007. Shotgun lipidomics by tandem mass spectrometry under data-dependent acquisition control. *Methods in Enzymology* 433, 175–191.
- Schwudke, D., Schuhmann, K., Herzog, R., Bornstein, S.R., Shevchenko, A., 2011. Shotgun lipidomics on high resolution mass spectrometers. *Cold Spring Harbor Perspectives in Biology* 3, a004614.
- Sleno, L., Volmer, D.A., 2004. Ion activation methods for tandem mass spectrometry. *Journal of Mass Spectrometry: JMS* 39, 1091–1112.
- Sud, M., Fahy, E., Cotter, D., *et al.*, 2007. LMSD: Lipid maps structure database. *Nucleic Acids Research* 35, D527–D532.
- Sullards, M.C., Allegood, J.C., Kelly, S., *et al.*, 2007. Structure-specific, quantitative methods for analysis of sphingolipids by liquid chromatography-tandem mass spectrometry: "Inside-out" sphingolipidomics. *Methods in Enzymology* 432, 83–115.
- van Meer, G., 2005. Cellular lipidomics. *The EMBO Journal* 24, 3159–3165.
- van Meer, G., Leeflang, B.R., Liebisch, G., Schmitz, G., Goi, F.M., 2007. The european lipidomics initiative: Enabling technologies. *Methods in Enzymology* 432, 213–232.
- van Meer, G., Voelker, D.R., Feigenson, G.W., 2008. Membrane lipids: Where they are and how they behave. *Nature Reviews. Molecular Cell Biology* 9, 112–124.
- Wells, J.M., McLuckey, S.A., 2005. Collision-induced dissociation (cid) of peptides and proteins. *Methods in Enzymology* 402, 148–185.
- Wenk, M.R., 2005. The emerging field of lipidomics. *Nature Reviews. Drug Discovery* 4, 594–610.
- Wishart, D.S., Tzur, D., Knox, C., *et al.*, 2007. Hmdb: The human metabolome database. *Nucleic Acids Research* 35, D521–D526.

Multi-Scale Modelling in Biology

Socrates Dokos, University of New South Wales, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Multi-scale modelling aims at developing computational models of physical systems characterised by phenomena spanning multiple spatial or temporal scales. Non-biological examples include modelling whole-material properties from microstructural and molecular-level features (Horstemeyer, 2009), chemical engineering processes from molecular-level interactions (Saliccioli *et al.*, 2011; Zhao *et al.*, 2012), nuclear reactor design (Mahadevan *et al.*, 2014) and astrophysics (van Elteren *et al.*, 2014). In biology and physiology, multi-scale models are often used to integrate biological processes operating from the molecular and sub-cellular scale through to cell and tissue function, up to whole-organ and organ systems (Hunter and Nielsen, 2005; Southern *et al.*, 2008; Qu *et al.*, 2011). Examples include biomechanics (Viceconti, 2012), electrophysiology (Winslow *et al.*, 2000), pulmonary gas exchange (Donovan, 2011), tissue patterning (Thorne *et al.*, 2007), tumour growth (Zangooei and Habibi, 2017), biochemical signalling pathways (Walker *et al.*, 2008) and systems biology in general (Dada and Mendes, 2011). An overview of the typical spatial scales employed in multi-scale biological models is shown in Fig. 1.

Multi-Scale Modelling Approaches

Approaches to formulating multi-scale models and their computational implementation will depend on the choice of spatial scales adopted, as well as the particular physics being coupled.

Model Formulation

Multi-scale models are generally constructed from one or more sub-component models, each characterised by a finer spatial and/or temporal scale than the overall model. Integration of these into a single coarse scale model involves some degree of macroscopic approximation, loosely termed *homogenisation* (across space) or *averaging* (across time) (Matthies, 2006). These methods include model reduction and course-graining (Gorban *et al.*, 2006), continuous field approximation (Bonilla *et al.*, 2014), compartmental and lumped-parameter modelling (Quarteroni *et al.*, 2016) and cellular automata approaches (Ermentrout and Edelstein-Keshet, 1993).

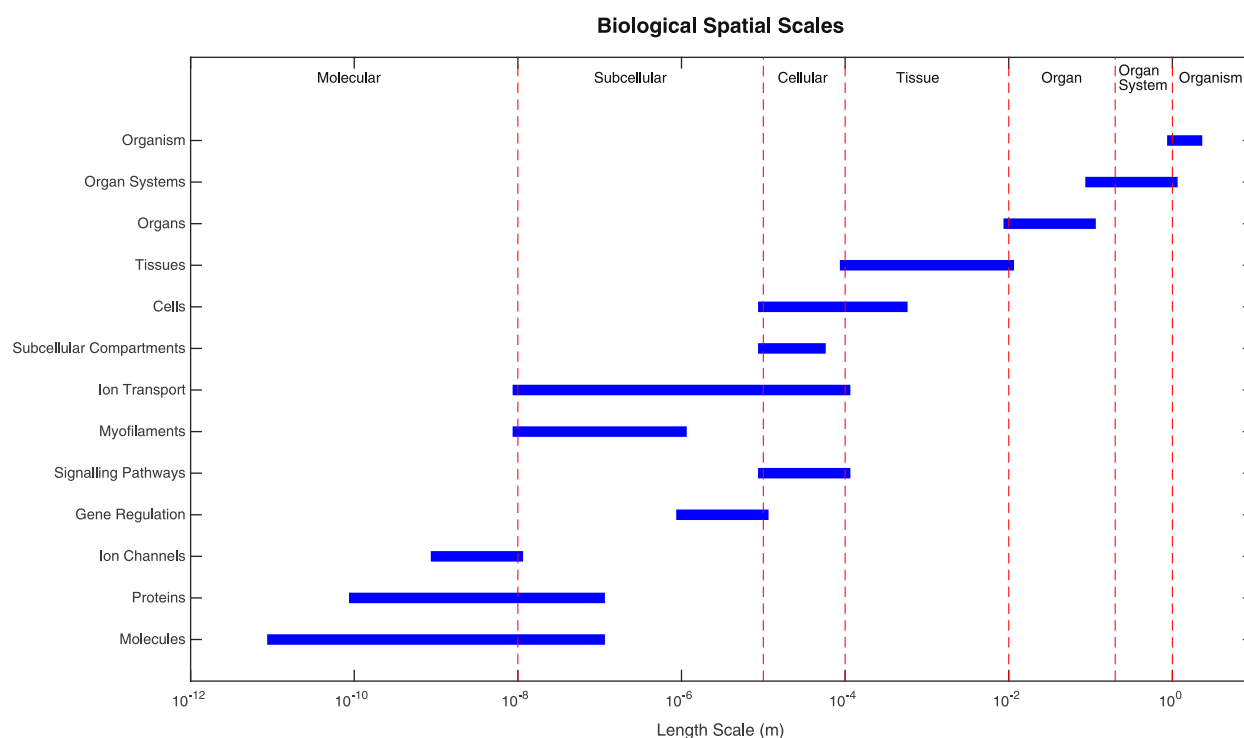


Fig. 1 Typical biological spatial scales used in multiple-scale modelling. Larger scales are shown typical for human physiology.

Course-graining methods approximate spatial heterogeneities by discretising the model's spatial domain into course regions, assigning spatially-averaged properties of the original system to each. In molecular dynamics simulations for example, instead of representing the kinetics of every atom in a system, groups of atoms are clustered together into single mesoscale entities, to provide a simplified, low-resolution approximation (Saunders and Voth, 2013). In contrast, *model reduction* seeks to lower the total degrees of freedom in a system of equations by projecting state variables onto lower-dimensional subspaces whilst preserving overall model behaviour. A simple example is Fitzhugh's (1961) reduction of the Hodgkin and Huxley (1952) equations of neural electrical activity from four state-variables to two, subsequently known as the Fitzhugh-Nagumo model.

In biology, many multi-scale models are typically formulated using *continuous field approximation*, in which reaction-diffusion equations are used to represent macroscopic properties of the system. For example, to model the propagation of electrical activity in excitable tissues such as the heart, the complex network of constituent interconnected cells can be represented using a simplified continuum approximation of cell membrane potential at each point in the tissue (Pullan et al., 2005) (see also section "Modelling Spatial Propagation of Electrical Activation in Heart Tissue" below). Another approach to multi-scale model formulation is the use of *lumped-parameter* or *compartmental* descriptions to represent separable model components such as, for example, modelling of insulin-glucose kinetics using only two discrete intercellular and blood plasma compartments (Tolić et al., 2000). Finally, *cellular-automata* models have also been used to represent multi-scale biological systems, in which simplified rule-based descriptions are employed to characterise biological function of constituent unit components. An example is the cardiac arrhythmia model of Mitchell et al. (1992), who used cellular automata to characterise single-cell electrical activity, modelling electrical conduction at the whole-tissue level.

Computational Implementation

To solve multi-scale biological models, commercial multiphysics finite-element numerical software such as COMSOL Multiphysics (COMSOL AB, Switzerland) (Dokos, 2017), or SIMULIA (Dassault Systèmes Simulia, USA) (Baillargeon et al., 2014), are increasingly being employed, along with a range of specialised open-source software platforms including OpenCMISS (see "Relevant Websites section"), Chaste (see "Relevant Websites section"), CHeart (see "Relevant Websites section") and Continuity (see "Relevant Websites section"): the latter platforms utilised within multi-scale biological modelling frameworks such as the Physiome Project (see "Relevant Websites section") (Hunter et al., 2008). Increasingly important to multi-scale biological model development are mark-up language specifications for biological/physiological systems including SBML (see "Relevant Websites section"), CellML (see "Relevant Websites section") and FieldML (see "Relevant Websites section"). Similar multi-scale computational frameworks have been described for chemical process engineering applications (Yang, 2013). An advantage of such frameworks is that, at least in principle, models of component subsystems can be readily swapped in and out from existing model libraries, providing a powerful means of formulating multi-scale biological models (Cooling et al., 2010).

Case Study – Modelling Cardiac Function

An interesting example of multi-scale modelling applied to biology is in the area of cardiac electromechanics, whereby ion channel dynamics in individual cardiomyocytes determines the pattern of electrical activation of the whole heart and its associated contractile performance (Land et al., 2012). Such models could have important applications for a range of patient-specific cardiovascular therapies in future, including development of antiarrhythmic drugs for long-QT syndrome (Hunter and Smith, 2016). A multi-scale model of the heart will incorporate a geometric description of the heart and its chambers, coupled to spatially-heterogeneous models of ion channel kinetics, local wall microstructure and active force generation, all linked to a lumped-parameter description of the circulation system acting as a load on the heart, as shown in Fig. 2.

Single-Cell Ionic Models

The basic modelling unit of multi-scale cardiac electromechanics is the single-cell ionic model of cardiac electric activity. Ion flow through voltage-gated transmembrane pores can be experimentally determined from patch-clamp data, yielding whole-cell membrane current macroscopic approximations expressed using either Hodgkin and Huxley (1952) formalism, or as Markov-type models (Hille, 2001). For example, in the human ventricular myocyte model of ten Tusscher et al. (2004), the rapid potassium ionic current I_{Kr} can be expressed in Hodgkin-Huxley form as

$$I_{Kr} = G_{Kr} \sqrt{\frac{[K]_o}{5.4}} x_{r1} x_{r2} (V_m - E_K)$$

where V_m is the membrane potential, G_{Kr} is the maximum membrane conductance of I_{Kr} channels, E_K is the equilibrium potential for potassium, $[K]_o$ is the extracellular potassium ion concentration, and x_{r1} , x_{r2} are gating variables, each governed by a first-order ordinary differential equation of the form

$$\frac{dx_{r1}}{dt} = \alpha_{r1}(1 - x_{r1}) - \beta_{r1}x_{r1}$$

where α_{r1} , β_{r1} are rate coefficients that are empirical functions of the membrane potential V_m . The same ionic current can be expressed also in Markov form, whereby several channel states are defined as either open, closed or inactive, and transition rates

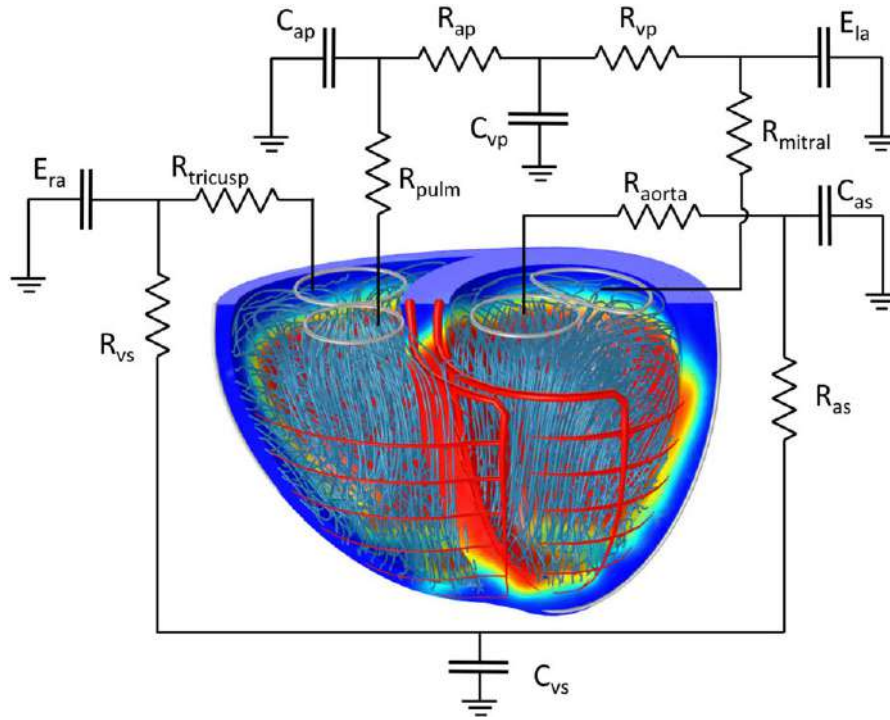


Fig. 2 Multiscale model of cardiac electromechanics, according to Bakir *et al.*, 2017 illustrating multi-physics coupling between cardiac electrical activation (activated regions shown in red, quiescent regions in blue), blood dynamics (shown as velocity streamlines within the left and right ventricular cavities), and lumped-parameter models of systemic and pulmonary circulations. Also shown are the tree-like structures corresponding to the rapidly-conducting cardiac conduction system.

between these states are given as a function of membrane potential V_m (Bett *et al.*, 2011). If the fraction of open states is given by O , then

$$I_{Kr} = G_{Kr} O (V_m - E_K)$$

Summing together all the ion currents present in the cell to form the total ionic current I_{tot} , leads to the space-clamped equation for membrane potential:

$$\frac{dV_m}{dt} = -\frac{I_{tot}}{C_m}$$

where C_m is the membrane capacitance per unit area. This formulation allows for the reconstruction of a cardiac action potential waveform, given an external stimulus of sufficient magnitude (added to I_{tot} in the above).

Several single-cell cardiac ionic models are available from the CellML online repository (see “Relevant Websites section”) (Lloyd *et al.*, 2004), which can be used to specify different expressions for I_{tot} in various regions of the heart. Many of these ionic models also include intracellular compartments for simulating calcium ion transients or other ions, second messengers and metabolites.

Modelling Spatial Propagation of Electrical Activation in Heart Tissue

Once I_{tot} has been specified for each region of the heart, propagation of electrical activation across the myocardium can be simulated using the macroscopic continuum approximation of excitable tissues known as the *monodomain formulation* (Cloherty *et al.*, 2006):

$$\beta \left(C_m \frac{\partial V_m}{\partial t} + i_{ion} \right) = \nabla \cdot (\sigma \nabla V_m)$$

where β is the tissue cell membrane surface-to-volume ratio, σ is the tissue conductivity, and ∇ is the *nabla* or *del* spatial partial derivative operator, given in 3D by

$$\nabla \equiv \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}$$

If the myocardium is *isotropic*, that is its physical properties are not dependent on spatial orientation of the tissue, then σ will be a scalar. In the general case however, myocardial tissue is anisotropic, both in its electrical and mechanical properties, with electrical conductivity dependent on local fine microstructure of the tissue. This microstructure consists of cardiac muscle *fibres* and *fibre sheets*, representing layers of bundled fibres. An orthogonal set of local coordinate axes can then be defined along the fibre, sheet, and normal-to-sheet directions, with electrical conductivity greatest along the fibre axis, followed by the sheet, then the normal-to-sheet axis (Hooks *et al.*, 2007). Defining the unit vectors of these local fibre, sheet and normal axes with respect to the global coordinate system as $\hat{\mathbf{f}}$, $\hat{\mathbf{s}}$, $\hat{\mathbf{n}}$, we can define the conductivity *tensor* for use with the monodomain equation as

$$\sigma = \sigma_f \hat{\mathbf{f}} \otimes \hat{\mathbf{f}} + \sigma_s \hat{\mathbf{s}} \otimes \hat{\mathbf{s}} + \sigma_n \hat{\mathbf{n}} \otimes \hat{\mathbf{n}}$$

where σ_f , σ_s and σ_n denote the scalar conductivities along the local fibre, sheet and normal-to-sheet orientations respectively, and $\otimes$ denotes the tensor outer product. Note that the orientation of the local microstructural axes will vary with spatial position in the heart.

Modelling Passive and Active Mechanical Properties of the Myocardium

In addition to electrical function, passive mechanical properties of the myocardium will also depend on the local microstructure (Dokos *et al.*, 2002). These properties can be expressed using a continuum mechanics approximation of the tissue, typically though use of a *hyperelastic* constitutive law, such as that of Holzapfel *et al.* (2000):

$$W = \frac{c_0}{2} (\bar{I}_1 - 3) + \frac{c_{f1}}{2c_{f2}} \left[e^{c_{f2}(\bar{I}_f - 1)^2} - 1 \right]$$

where W denotes a scalar *strain energy* function, $\bar{I}_1$ is the first invariant of the Cauchy-Green deformation tensor, c_0 , c_{f1} and c_{f2} are material parameters of the tissue, and $\bar{I}_f$ denotes the stretch along the fibre direction $\hat{\mathbf{f}}$. Detailed step-by-step implementation of this material description for myocardium using COMSOL Multiphysics software is given in Dokos (2017).

Many cardiac single-cell ionic models also contain descriptions of cytosolic calcium transients, which can be used to model calcium ion interaction with contractile proteins to generate active force (Hunter *et al.*, 1998). Other models simply link active force generation to the membrane potential itself (Nash and Panfilov, 2004). Either way, these models can be used to generate active tension T_a , which can then be incorporated into the passive strain energy function to determine the 2nd-Piola Kirchhoff stress tensor components S_{ij} within the myocardium according to

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} + T_a \hat{\mathbf{f}} \otimes \hat{\mathbf{f}} + \gamma T_a \hat{\mathbf{s}} \otimes \hat{\mathbf{s}} + \gamma T_a \hat{\mathbf{n}} \otimes \hat{\mathbf{n}}$$

where E_{ij} are the components of the Green strain tensor, and γ represents the fractional component of active tension in the cross-fibre directions due to fibre branching. Passive constitutive laws coupled with active force generation allows structural deformation of the myocardium to be determined (Holzapfel, 2000).

Modelling Fluid Dynamics and Circulatory Blood Flow

Blood flow within the ventricles can be modelled using incompressible Navier-Stokes equations (Fung, 1997):

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u}$$

$$\nabla \cdot \mathbf{u} = 0$$

where $\mathbf{u}$ is the fluid velocity field and p is its pressure, ρ is the fluid density and μ its viscosity. Boundary conditions for the fluid equations include a moving wall condition on the endocardial surface, whereby the fluid velocity is set to equal the velocity of the moving wall determined from the structural mechanics deformation. Because of the deforming fluid domain, a moving mesh approach must also be implemented, known as the Arbitrary Lagrangian-Eulerian (ALE) method (Donea *et al.*, 2004). In turn, the fluid pressure p can be used as an endocardial boundary load for the wall deformation mechanics. In practice however, the pressure will be nearly-constant across the surface at each time instant. As a result, most multi-scale models of cardiac electromechanics omit the Navier-Stokes equations, determining the endocardial fluid pressure load from the circulatory load model instead.

As shown in Fig. 2, the circulatory load on the heart can also be approximated by hydraulic circuits, analogous to electric circuits where the analogue of current is fluid flow rate, and the analogue of voltage is fluid pressure. Also shown are capacitive elements, representing the compliance of arteries and veins due to their elastic distensibility. These circuit elements are lumped parameter representations of circulatory components including tissue beds, large arteries and the lungs.

Multi-Scale Overview of Cardiac Electromechanics

From the above descriptions, it is evident that the multi-physics model of cardiac electromechanics combines several spatial scales into a single, albeit complex, model formulation. Beginning with the membrane ionic channels, whose properties can be modelled at the molecular level using molecular dynamics, a macroscopic representation of their function is adopted in the form of Hodgkin-Huxley formalism or Markov-state models. These descriptions of membrane currents are then incorporated into single-cell electrophysiology models, which can additionally include expressions for sub-cellular ionic concentrations, metabolites, contractile processes and signalling pathways. The cell models can then be combined into the monodomain macroscopic formulation of tissue electrical activity to solve for the propagating electrical activation wavefront. Similar macroscopic approximations are also made for the passive and active structural mechanics, incorporating features of the local fine tissue microstructure. Finally, the entire circulation, which sets the loading conditions on the heart, is approximated as a lumped-parameter description, representing various components of the vasculature.

Conclusion

Biological systems are inherently complex, characterised by multiple phenomena operating at various spatial and temporal scales from the molecular and sub-cellular, through to whole-cells and tissues, up to the whole-organ level and beyond. Multi-scale modelling approaches are well-suited to the study of biological systems, integrating knowledge of the various subsystems into a coherent understanding of whole-system properties and function. In future, with ever-increasing computational power becoming available, multi-scale models will undoubtedly yield fundamental insights into numerous complex biological phenomena, even playing a key role in drug design and personalised therapies.

See also: Cell Modeling and Simulation. Computational Approaches for Modeling Signal Transduction Networks. Computational Systems Biology Applications. Computational Systems Biology. Data Formats for Systems Biology and Quantitative Modeling. Integrative Bioinformatics. Molecular Dynamics and Simulation. Natural Language Processing Approaches in Bioinformatics. Pathway Informatics. Quantitative Modelling Approaches. Studies of Body Systems

References

- Baillargeon, B., Rebelo, N., Fox, D.D., Taylor, R.L., Kuhl, E., 2014. The living heart project: A robust and integrative simulator for human heart function. *European Journal of Mechanics A/Solids* 48, 38–47.
- Bakir, A.A., Al, A.A., Lovell, N.H. Dokos, S., 2017. Dokos A generic cardiac biventricular fluid-electromechanics model. In: *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3680–3683.
- Bett, G.C.L., Zhou, Q., Rasmusson, R.L., 2011. Models of HERG gating. *Biophysical Journal* 101, 631–642.
- Bonilla, L.L., Capasso, V., Alvaro, M., Carretero, M., 2014. Hybrid modeling of tumor-induced angiogenesis. *Physical Review E* 90, 062716.
- Cloherly, S.L., Dokos, S., Lovell, N.H., 2006. Modeling of electrical activity in cardiac tissue. In: Akay, M. (Ed.), *Wiley Encyclopedia of Biomedical Engineering*, vol. 2. Hoboken, NJ: John Wiley and Sons, pp. 1216–1226.
- Cooling, M.T., Rouilly, V., Misirli, G., *et al.*, 2010. Standard virtual biological parts: A repository of modular modeling components for synthetic biology. *Bioinformatics* 26, 925–931.
- Dada, J.O., Mendes, P., 2011. Multi-scale modelling and simulation in systems biology. *Integrative Biology* 3, 86–96.
- Dokos, S., 2017. *Modelling Organs, Tissues, Cells and Devices Using Matlab and Comsol Multiphysics*. Berlin: Springer-Verlag.
- Dokos, S., Smaill, B.H., Young, A.A., LeGrice, I.J., 2002. Shear properties of passive ventricular myocardium. *American Journal of Physiology: Heart and Circulatory Physiology* 283, H2650–H2659.
- Donea, J., Huerta, A., Ponthot, J.-Ph., Rodríguez-Ferran, A., 2004. Arbitrary Lagrangian-Eulerian methods. In: Stein, E., de Borst, R., Hughes, T.J.R. (Eds.), *Encyclopedia of Computational Mechanics*. Vol. 1: Fundamentals. New York: Wiley, pp. 413–437.
- Donovan, G.M., 2011. Multiscale mathematical models of airway constriction and disease. *Pulmonary Pharmacology and Therapeutics* 24, 533–539.
- van Elteren, A., Pelupessy, I., Portegies Zwart, S., 2014. Multi-scale and multi-domain computational astrophysics. *Philosophical Transactions of the Royal Society A* 372, 20130385.
- Ermentrout, G.B., Edelstein-Keshet, L., 1993. Cellular automata approaches to biological modeling. *Journal of Theoretical Biology* 160, 97–133.
- Fitzhugh, R., 1961. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal* 1, 445–466.
- Fung, Y.C., 1997. *Biomechanics: Circulation*, second ed. New York: Springer.
- Gorban, A.N., Kazantsev, N.K., Kevrekides, I.G., Öttinger, H.C., Theodoropoulos, C., 2006. *Model Reduction and Coarse-Graining Approaches for Multiscale Phenomena*. Berlin: Springer.
- Hille, B., 2001. *Ion Channels Of Excitable Membranes*, third ed. Sunderland, MA: Sinauer Associates.
- Hodgkin, A.L., Huxley, A.F., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* 117, 500–544.
- Holzappel, G.A., 2000. *Nonlinear Solid Mechanics – A Continuum Approach for Engineering*. UK: Wiley-Blackwell.
- Holzappel, G.A., Gasser, T.C., Ogden, R.W., 2000. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity and the Physical Science of Solids* 61, 1–48.
- Hooks, D.A., Trew, M.L., Caldwell, B.J., *et al.*, 2007. Laminar arrangement of ventricular myocytes influences electrical behavior of the heart. *Circulation Research* 101, e103–e112.
- Horstemeyer, M.F., 2009. Multiscale modeling: A review. In: Leszczynski, J., Shukla, M.K. (Eds.), *Practical Aspects of Computational Chemistry*. New York: Springer, pp. 87–135.

- Hunter, P.J., Crampin, E.J., Nielsen, P.M.F., 2008. Bioinformatics, multiscale modelling and the IUPS physiome project. *Briefings in Bioinformatics* 9, 333–343.
- Hunter, P.J., McCulloch, A.D., ter Keurs, H., 1998. Modelling the mechanical properties of cardiac muscle. *Progress in Biophysics and Molecular Biology* 69, 289–331.
- Hunter, P., Nielsen, P., 2005. A strategy for integrative computational physiology. *Physiology (Bethesda)* 20, 316–325.
- Hunter, P.J., Smith, N.P., 2016. The cardiac physiome project. *Journal of Physiology* 594, 6815–6816.
- Land, S., Niederer, S.A., Smith, P., 2012. Efficient computational methods for strongly coupled cardiac electromechanics. *IEEE Transactions on Biomedical Engineering* 59, 1219–1228.
- Lloyd, C.M., Halstead, M.D.B., Nielsen, P.F., 2004. CellML: Its future, present and past. *Progress in Biophysics and Molecular Biology* 85, 433–450.
- Mahadevan, V.S., Merzari, E., Tautges, T., *et al.*, 2014. High-resolution coupled physics solvers for analysing fine-scale nuclear reactor design problems. *Philosophical Transactions of the Royal Society A* 372, 20130381.
- Matthies, K., 2006. Exponential estimates in averaging and homogenisation. In: Mielke, A. (Ed.), *Analysis, Modeling and Simulation of Multiscale Problems*. Berlin: Springer, pp. 1–19.
- Mitchell, R.H., Bailey, A.H., Anderson, J., 1992. Cellular automaton model of ventricular fibrillation. *IEEE Transactions on Biomedical Engineering* 39, 253–259.
- Nash, M.P., Panfilov, A.V., 2004. Electromechanical model of excitable tissue to study reentrant cardiac arrhythmias. *Progress in Biophysics and Molecular Biology* 85, 501–522.
- Pullan, A.J., Buist, M.L., Cheng, L.K., 2005. *Mathematically Modelling the Electrical Activity of the Heart: From Cell to Body Surface and Back Again*. Singapore: World Scientific Publishing.
- Quarteroni, A., Veneziani, A., Vergara, C., 2016. Geometric multiscale modeling of the cardiovascular system, between theory and practice. *Computer Methods in Applied Mechanics and Engineering* 302, 193–252.
- Qu, Z., Garfinkel, A., Weiss, J.N., Nivala, M., 2011. Multi-scale modeling in biology: How to bridge the gaps between scales? *Progress in Biophysics and Molecular Biology* 107, 21–31.
- Saliccioli, M., Stamatakis, M., Caratzoulas, S., Vlachos, D.G., 2011. A review of multiscale modeling of metal-catalyzed reactions: Mechanism development for complexity and emergent behavior. *Chemical Engineering Science* 66, 4319–4355.
- Saunders, M.G., Voth, G.A., 2013. Course-graining methods for computational biology. *Annual Reviews of Biophysics* 42, 73–93.
- Southern, J., Pitt-Francis, J., Whiteley, J., *et al.*, 2008. Multi-scale computational modelling in biology and physiology. *Progress in Biophysics and Molecular Biology* 96, 60–89.
- Thorne, B.C., Bailey, A.M., Peirce, S.M., 2007. Combining experiments with multi-cell agent-based modeling to study biological tissue patterning. *Briefings in Bioinformatics* 8, 245–257.
- Tolić, I.M., Mosekilde, E., Sturis, J., 2000. Modeling the insulin-glucose feedback system: The significance of pulsatile insulin secretion. *Journal of Theoretical Biology* 207, 361–375.
- ten Tusscher, K.H.W.J., Noble, D., Noble, P.J., Panfilov, A.V., 2004. A model for human ventricular tissue. *American Journal of Physiology: Heart and Circulatory Physiology* 286, H1573–H1589.
- Viceconti, M., 2012. *Multiscale Modelling of the Skeletal System*. Cambridge: Cambridge University Press.
- Walker, D.C., Georgopoulos, N.T., Southgate, J., 2008. From pathway to population – A multiscale model of juxtacrine EGFR-MAPK signalling. *BMC Systems Biology* 2, 102.
- Winslow, R.L., Scollan, D.F., Holmes, A., *et al.*, 2000. Electrophysiological modeling of cardiac ventricular function: From cell to organ. *Annual Review of Biomedical Engineering* 2, 119–155.
- Yang, A., 2013. On the common conceptual and computational frameworks for multiscale modeling. *Industrial and Engineering Chemistry Research* 52, 11451–11462.
- Zangooei, M.H., Habibi, J., 2017. Hybrid multiscale modeling and prediction of cancer cell behavior. *PLOS ONE* 12, e0183810.
- Zhao, Y., Jiang, C., Yang, A., 2012. Towards computer-aided multiscale modelling: An overarching methodology and support of conceptual modelling. *Computers and Chemical Engineering* 36, 10–21.

Relevant Websites

<https://www.cellml.org>
 cellML.
<http://www.cs.ox.ac.uk/chaste>
 Chaste.
<http://cheart.co.uk>
 CHeart.
<http://continuity.ucsd.edu>
 CONTINUITY 6.
<http://opencmis.org>
 OpenCMIS.
<http://physiomeproject.org>
 Physiome Project.
<http://physiomeproject.org/software/fieldml>
 Physiome Project.
<http://sbml.org>
 SBML.org.

Computational Immunogenetics

Marta Gómez Perosanz, Complutense University of Madrid, Madrid, Spain

Giulia Russo, University of Catania, Catania, Italy

Jose Luis Sanchez-Trincado Lopez, Complutense University of Madrid, Madrid, Spain

Marzio Pennisi, University of Catania, Catania, Italy

Pedro A Reche, Complutense University of Madrid, Madrid, Spain

Adrian Shepherd, University of London, London, United Kingdom

Francesco Pappalardo, University of Catania, Catania, Italy

© 2019 Elsevier Inc. All rights reserved.

Introduction

The extraordinary availability of large sets of data produced in biotechnology through the advancement of information technology is promoting the amplification of the knowledge of biological systems. These innovations have swung the pendulum in the way biomedical research, development and applications are done. Clinical data complement biological data, enabling detailed descriptions of various healthy and diseased states, progression and responses to therapies. The amount of data representing various biological states, processes, and their time dependencies enable the study of biological systems at various levels of organization, from molecule to organism, and even at population levels. Multiple sources of data support a rapidly growing body of biomedical knowledge; however, our ability to analyse and interpret these data lags far behind data generation and storage capacity.

Mathematical and computational models are increasingly used to help interpret biomedical data produced by high-throughput genomics and proteomics projects. Advanced applications of computer models that support the simulation of biological processes are used to generate hypotheses and plan experiments. Appropriately interfaced with biomedical databases, computational models are necessary for rapid access to and sharing of knowledge through data mining and knowledge discovery approaches.

The immune system (IS) represents one of the most fascinating systems from many points of view, including biology, physics, computer science and mathematics. Together with the central nervous system, it is probably the most complex, adaptive and highly distributed system in life sciences. It constitutes the fundamental defence mechanism of the vertebrate animals, including human beings, from invading from the pathogens and harmful foreign substances. Following the evolution of multicellular organisms, the immune system developed various defence mechanisms and abilities against all kinds of pathogens, like viruses, bacteria, fungi and other parasites.

Innate Immune System

The “evolutionary-older” mechanisms belong to the class of “innate immunity” defences, and include physical barriers, soluble mediators that directly inhibit foreign micro-organisms and specialized cells able to phagocytose and kill micro-organisms.

The simplest physical barrier is represented by the skin which gives a very effective protection from infection. The mildly acidic pH of skin also inhibits bacterial proliferation. Moreover, the blood coagulation system stops the bleeding, and creates a protective clot over wounds, while the internal mucosae inhibits foreign invaders from entering in the organism. The very acidic pH of stomach juices is also able of effectively sterilize ingested materials. Even the mechanisms that regulate the host temperature (i.e. fever) represent a defence mechanism against the proliferation of some pathogens.

Many soluble molecules such as defensins (natural antibiotics), lysozyme (an enzyme that lyses the wall of some bacteria), the complement (a system of proteins activated in cascade in the cell membrane of certain pathogens that result in their lysis), opsonins (molecules that coat pathogens and facilitate phagocytosis) and some ancient cytokines such as interferons can inhibit viral infections and also kill bacteria.

Innate immunity is also composed by many cellular-driven mechanisms that include the cellular migration toward invaders and phagocytosis, followed by intracellular destruction of the ingested micro-organism. Extracellular killing is also possible if phagocytes release substances that can cause lysis. In mammals such actions are carried by granulocytes and macrophages. The former contains intracellular granules with lytic substances, whereas the latter emigrate to practically all tissues and organs to carry out their phagocyte functions, also important in the turnover of aging cell components.

Even if their evolution is relatively recent and goes in parallel with the evolution of lymphocytes, natural killer (NK) cells are commonly considered as non-phagocytic innate immunity cells, and are able to kill virus-infected cells.

In innate immunity, the recognition of pathogens is done thanks to the presence of pathogen-associated molecular patterns expressed by micro-organisms. Cells belonging to innate immunity hold receptors, globally referred to as “pattern-recognition receptors”, that are able to recognize and match with such molecular patterns. These receptors include the family of Toll-like receptors (TLR), mannose receptors (MR) and seven-transmembrane spanning receptors (TM7). TLR recognize bacterial and viral nucleic acids, flagellin, bacterial peptides, lipopolysaccharide (LPS) and other bacterial components. MR receptors bind carbohydrate moieties of several pathogens, such as bacteria, fungi, parasites, and viruses. Finally, TM7 receptors are activated by bacterial peptides or by endogenous chemokines.

Adaptive Immune System

The evolutionary process brought to the developing of newer adaptive and specific responses that couple the classical innate (or natural) immune responses. Adaptive immunity is in fact evolutionary recent. Starting from sharks, the evolution of adaptive immunity followed those of vertebrates, contributing to their evolutionary success and their long life spans. It should be said that adaptive immunity must be seen as a complement of innate immunity, as the two systems strongly interact and cooperate to eradicate pathogens.

Lymphocytes represent the principal effectors of adaptive immunity. Lymphocytes circulate in the blood, tissues, lymphatic vessels and reside in lymphoid organs, which include the thymus, the spleen and lymph nodes. Why are lymphocyte functions so different from innate immunity cells? The answer can be reassumed as follows: specificity, memory and immune tolerance. The lymphocyte population is composed by millions of clones specialized (thanks to their clonotypic antigen receptors) in recognizing a specific antigenic sequence. Only the clones that express the receptors capable of recognizing a specific micro-organism are activated and thus proliferate upon infection, while the others remain inactive. A unique random DNA rearrangement procedure is responsible to bring, from a relatively small pool of DNA sequences, billions of different receptors (a theoretical estimate is of the order of 10^{20}). The immunological meaning of memory represent the capability of lymphocytes to “remember”, thanks to specialized memory cells, the first encounter with a given antigen (primary immune response) in order to respond more promptly and more efficiently to subsequent encounters (secondary immune response). Immune tolerance is a selective mechanism used by the immune system to exclude potentially autologous (self) lymphocytes clones through their destruction or their inactivation.

Lymphocytes populations can be distinguished in many ways, starting from the different types of antigen receptors or different specific functions. The most fundamental distinction is between T and B cells. B cells are the main effectors of the so-called “humoral response” and use immunoglobulins(Ig)-like membrane antigen receptors. After recognition and stimulation by the antigen, activated B cells differentiate into plasma cells which secrete a soluble form of the receptor, called antibody. The help of CD4 T-cells (helper T-cells) is a fundamental requirement to mount an appropriate response of B-cells in producing mature antibodies. Each antibody molecule is a dimer of a heavy and a light chain, and each one has a variable and a constant part. The basic antibody molecule is Y-shaped with two independent antigen-binding sites (at the ends of the diagonal segments of the Y). The constant stem of the Y mediates the so-called effector functions of the antibody. We can distinguish among various classes of antibody types. IgM are the first antibodies produced during the primary immune response. IgG are the main class of antibodies released in blood during secondary immune responses. In humans there are four different IgG types (IgG1, IgG2, IgG3 and IgG4). IgA are specialized for functioning in secretions like milk, tears, saliva and intestinal. IgE are best known in conjunction with allergies.

Antibodies bind and inactivate their cognate antigens, and with different mechanisms they promote immunity against foreign pathogens. Antibodies binding to cell membranes may activate the complement system that can lead to cell lysis. Moreover, leukocytes that express surface receptors for the constant stem of antibodies can act with phagocytosis (opsonization) and cell lysis (antibody-dependent cell-mediated cytotoxicity, ADCC) of antibody-bound cells. In this way, macrophages or NK cells acquire the antigen specificity of adaptive immunity. Finally, insoluble antigen-antibody complexes (immunocomplexes) can be rapidly removed from the host.

T cells instead represent the main effectors of the “cell-mediated response”. There are many important differences between B and T cells. First, antibodies are at the same time the receptor and the effector molecule of B cells, while the T cell receptor (TCR) is a membrane receptor that activates a series of effector actions mediated by other molecules. Moreover, antibodies can bind any conceivable molecular species such as proteins, lipids, sugars and small organic molecules while T cells receptors are specialized in recognizing only small peptides on the surface of cells. Furthermore, antibodies recognize the antigen in its free native form, whereas TCRs recognize only specific peptides present on the host cell membrane bound to a cellular protein called Major Histocompatibility Complex (MHC), not the whole antigen structure.

There are various T cell populations. Cytotoxic T cells (Tc, also referred to as CTL) directly kill cells expressing the antigen, helper T cells (Th) promote the activities of B and Tc cells, while regulatory T cells (Treg) modulate the immune responses. The Th population can be further distinguished in Th1 and Th2 sub-populations, according to the cytokines they secrete. The former mostly release gamma-interferon and other cytokines to stimulate immune responses against virus and intracellular bacteria, whereas the latter release interleukin 4 (IL-4) against parasites.

Adaptive immune response involves a complex cooperation between both innate and adaptive cells. In general, immune responses start on the periphery of the body where antigen presenting cells (i.e., cells that commonly are able to process and endocytose antigenic sequences or proteins) capture an antigen and move through lymphatic system towards lymph nodes. Here they present the antigenic peptides exposed on their membrane in conjunction with MHC molecules to Th cells and secrete interleukins, such as Interleukin 12 (IL-12), to promote Th responses. Th cells with specialized TCR for the target antigen are activated by both recognition and co-stimulatory molecules.

Activated Th cells proliferate and secrete other cytokines activating antibody production by B cells and replication of Tc cells. Antibodies, Th and Tc cells reach the periphery where they encounter the pathogen. Antibodies bind to pathogens promoting, for example, phagocytosis by macrophages, activation of the complement system and the ADCC in conjunction with NK cells, respectively. Th cells secrete multiple cytokines with diverse functions such as attraction of other leukocytes in the place of infection, inhibition of viral replication and stimulation of haemathopoiesis to increase the number of leukocytes. Tc cells recognize and directly kill infected cells expressing peptide-MHC complexes on the cell membrane.

As previously stated, lymphocytes receptors are obtained by random rearrangements of a relatively small number of DNA molecules. This random rearrangement sometimes implies the risk of generating receptors which can bring self-reactive responses (autoimmunity). The way the immune system uses to avoid autoimmunity is to generate new cells clones and to destroy potentially harmful clones thereafter. After initial development in the bone marrow, T cell precursors migrate to the thymus. Positive selection occurs in cortex of the thymus over double positive (CD8 + CD4 +) positive thymocytes, selecting those that are capable of engaging with self peptide-MHC complexes. During positive selection, thymocytes commit to recognize either MHC class I or MHC class II molecules, developing into single positive (SP) cells that express either CD8 or CD4. Those cells can not engage with peptide-MHC complex die by apoptosis (death by neglect). Up to 95% of the cells die by neglect. Subsequently, SP cells migrate to the medulla of the thymus and in the region between the cortex and the medulla they undergo the process of negative selection. SP cells that recognized self-peptide/MHC complexes with high affinity undergo apoptosis. Positive selection guarantee MHC restriction while negative selection provides a mechanism of central tolerance.

Elimination of self-reactive T clones in the thymus (central tolerance) is not 100% efficient. Self-reactive clones that surpass the central tolerance are rendered harmless by a mechanism called "peripheral tolerance". T cell activation depends from antigen presentation of the MHC-peptide complex by APC. Only these cells are able to release the appropriate costimulatory molecules required to completely activate T clones. In lack of these costimulatory molecules, the interaction between a self-reactive T cell and a parenchymal cell brings to the inhibition of the T cell (anergy).

Sometimes a breakdown of the tolerance mechanisms may entitle the appearing of autoimmune diseases. Auto-reactive cells can be activated by foreign micro-organisms that express antigens similar to autologous molecules (antigen mimicry) or as a consequence of the presence of cytokines that have been released to activate other clones, but have the collateral effect to counterbalance the lack of costimulatory molecules needed for anergy.

Allergy is due to a pathological immune response (hypersensitivity) to specific antigens (allergens) contained in food, dust, pollens, animal components, chemical products and drugs.

Recognition of the allergen elicits a Th2 response leading to IgE antibody production and capture by receptors on the surface of mast cells and basophil granulocytes (sensitization phase). Allergen binding then triggers the release from cells of various mediators, such as histamine, prostaglandins and cytokines, which cause the allergic reaction.

The type of reaction is typically related to the mode of entry of allergens, e.g. inhaled pollens cause hay fever with sneezing and cough, while food allergens cause gastrointestinal reactions.

The most severe type of allergic reaction, anaphylactic shock, may result from allergen injection, for example bee stings.

A lack of immune response can be caused either by heritable genetic defects (primary immunodeficiency) or by extrinsic causes as, for example, the HIV infection or drug treatments (secondary immunodeficiencies). The fact that all immunodeficient individuals experience a significant increase in sensitivity to chronic and lethal infections is a clear illustration of the defensive role of the immune system toward microorganisms.

The immune system is capable of responding against anything that is considered as non-self. Transplantation of cells, organs or tissues between unrelated individuals elicits in the recipient a very strong immune response directed against the graft. The cause of tissue incompatibilities within the same species is due to the fact that individuals express unique antigens (alloantigens). Either recipient antibodies or by T cells recognizing donor MHC antigens can be cause of rejection. The solution to this problem is given by immunosuppressive drugs like cyclosporin that block T cell activation and response.

In immunology, the use of mathematical and computational methodologies revealed to be a very important support tool not only for the understanding of the immune system behaviour when dealing with pathogens, tumours and auto-immunity disorders, but also for testing, predicting and optimizing possible treatments for them.

Modelling the Immune System

A model is a mapping from a real-world domain to a mathematical/computational domain; thus, it highlights some of the essential believed properties while ignoring believed unessential ones. Mathematical modelling permits the integration of biological and clinical data at various levels and, in doing so, has the potential to provide insights into complex diseases. Modelling necessitates the statement of explicit hypotheses, a process which improves our understanding of the biological system and can uncover critical points where understanding is still poor. Subsequent simulations can reveal hidden patterns and/ or counter-intuitive mechanisms in complex systems. Theoretical thinking and mathematical modelling help generate new hypotheses that can be tested in the laboratory. Mathematical and computational models are usually designed to improve our understanding of phenomena, to validate or reject hypotheses, to shorten production costs, or to find possible points of intervention in order to drive the behaviour of a system towards given goals.

Of course, such models usually represent coarse sketch of the real biological scenario, however they often demonstrated able to capture the complexity of the biological problem, being able to reproduce the most known important immunological features and to bring out some unknown hidden features.

Models help in finding possible therapeutic targets for many diseases, in optimize existing treatments, in shorten, both in terms of costs and time, the pharmaceutical research and discovery pipeline. Furthermore, a technological revolution is nowadays beginning, as models are starting to be personalized on specific individual data. Patient specific models will allow to use such techniques to reproduce both the immunological background and future of the individual, and to personalize therapeutic and

prophylactic interventions directly on him, achieving better efficacy and minimizing the risk of undesirable effects. This will drive towards what we call a “personalized medicine” scenario.

A good model should be: (1) relevant, capturing the essential properties of the phenomenon; (2) computable, driving computational knowledge into mathematical representation; (3) understandable, offering a conceptual framework for thinking about the scientific domain; and (4) extensible, allowing the inclusion of additional real properties in the same mathematical scheme.

In the framework of the modelling of immune system dynamics, relevant means that the model should be able to capture the essential properties of the system, namely its organization and dynamic behaviour; computable refers to the model's ability to simulate both the dynamic behaviour and the evolution and interactions of system entities; understandable means that the model must reproduce concepts and ideas of immunology while opening new computational possibilities for understanding immune system interactions; finally, extensible refers to the possibility to include new immunology concepts and knowledge with limited effort using the same mathematical and computational framework.

Immunoinformatics represents a multidisciplinary field dealing with experimental immunology and computer science. In particular, it takes advantage of computational approaches that help to understand the overwhelming and complex dynamics of immune system and to face with the enormous amount of data in immunological context.

Recently, immunoinformatics has grown significantly in scientific field and it potentially has the cards to become a discipline of outstanding value. Immunoinformatics involves several subareas that include immunogenomics, immunoproteomics, epitope prediction, in silico vaccination and systems biology approaches applied to immunology. Whereas immunoinformatics aims to develop mathematical and computational methods to study the dynamics of cellular and molecular entities during the immune response (Pappalardo *et al.*, 2015), the immunological bioinformatics focuses on proposing methods to investigate big genomic and proteomic immunological-related datasets and predict new knowledge mainly by statistical inference and machine learning algorithms.

The glut of data produced by high-throughput instrumentation, notably genomics, transcriptomics, epigenetics, and proteomics methods, requires computational tools for acquisition, storage, and analysis of immunological data. The exploitation of such a huge amount of immunological data usually requires its conversion into computational problems, their solution using mathematical and computational approaches, and then the translation of the obtained results into immunologically meaningful interpretations.

Immunogenetics Modelling Approaches

Antibody Modelling

An antibody (Ab) is a protein with a shape depicting a large Y. It is a protein that is released largely by plasma cells. It's a component of the humoral immunity able to make pathogens not harmful. Specifically, the antibody recognizes the antigen expressed by the pathogen or on the cell surface. The Ab binding region is built by a paratope that is specific for one particular epitope on an antigen. As soon as the Ab binds specifically the antigen, the pathogen or the target cell can be killed directly by the Ab or signalled to other specific immune cells for killing.

The specific function of an antibody (Ab), i.e. where it binds and with what affinity, is intimately related to its structure. However, the number of solved Ab structures is tiny compared to the number of known sequences. Given that precisely solving the structure of an Ab (notably using X-ray crystallography) is typically both challenging and time consuming, modelling the structure computationally is a highly-desirable option.

At the time of writing, there are over 2600 Ab structures in the Protein Data Bank (PDB) (see Relevant Website section), of which around 1700 are bound to some kind of ligand (such as protein, peptide and glycan). Even Abs from different species are sufficiently similar that producing homology models of Ab structures with good overall similarity (measured, for example, using the root-mean-square deviation (RMSD) between the α carbons of model and target) is relatively straightforward using standard homology modelling tools, such as Modeller (Webb and Sali, 2014). But low overall RMSD is rarely the end goal of Ab modelling – what matters for nearly all practical applications is that key regions of an Ab are modelled with high accuracy, notably those parts of the Ab responsible for binding to an Ag. In most cases this requires multiple Complementarity-determining regions (CDRs) – which, depending on the definitions used, consist exclusively or predominantly of loops – to be modelled. CDRs are part of the variable chains in immunoglobulins (antibodies) and T cell receptors, generated by B-cells and T-cells respectively. There are three types of CDRs, CDR1, CDR2 and CDR3. Hypervariable regions associated with both immunoglobulins and T cell receptors are found in CDRs. In general, loop modelling is a considerable challenge and becomes increasingly intractable beyond a length of around 8 amino acids (Fiser *et al.*, 2000). CDR3 loops are usually longer than 10 residues and comprise some of V, all of diversity (D, heavy chains only) and joining (J) regions.

Thankfully there are a number of dedicated Ab modelling tools that exploit knowledge of canonical CDR conformations to produce accurate models of most Ab CDRs, at least when conditions are favourable. Example are Pigs (Marcatili *et al.*, 2014), RosettaAntibody (Weitzner *et al.*, 2017). Modelling and docking of antibody structures with Rosetta, and ABodyBuilder (Leem *et al.*, 2016). A typical strategy for such a tool is to choose appropriate template structures from the PDB (often separate templates for the heavy and light chains), graft canonical CDR loops onto the templates (typically good canonical CDR matching will be found for all but CDR-H3), and model the CDR-H3 (either from a template, or *ab initio*).

The accuracy with which tools can generate models of unknown structures has been evaluated via two Antibody Modelling Assessment initiatives, AMA-I (Almagro *et al.*, 2011) and AMA-II (Almagro *et al.*, 2014), at which research groups were given the

task of modelling Ab variable domains of unpublished high-resolution structures. The outcome of AMA-II was positive in several important respects. Accurate models within an RMSD of 1.5 Å are possible with most software for most CDRs and most structures, and higher accuracy is often achievable when someone skilled at modelling is prepared to spend time manually checking and tweaking the model. On the other hand, performance is often poor with Abs from under-represented species and models of the most variable CDR, CDR-H3, are often poor. Given that CDR-H3 is often the most important CDR for Ag binding, this represents a significant problem and an important research topic in its own right (Marks and Deane, 2017).

Even if one has a method capable of generating highly accurate models of unbound Abs, it is not clear how useful that is if one wishes to understand where that Ab binds – even if one has (as is often the case) a solved structure of its known, or likely, Ag. This introduces another challenging problem, that of Ab docking. Protein-protein docking in general has been the subject of the blinded prediction evaluation initiative, CAPRI, since 2002 (Gray *et al.*, 2003). Effective docking commonly involves two key components: the generation of lots of models (decoys) that widely sample the potential “binding space”; and the effective scoring of these models to identify the best decoy. Ab-Ag docking has its own characteristics that make it somewhat different from the more general case – in some respects easier (we know in advance that Abs bind Ags mainly via their CDRs), in other respects potentially more challenging (the CDRs are flexible), and in further respects simply different (Ab contact residues tend to be of different types to those of proteins in general (Brenke *et al.*, 2012). This has led to the development of Ab-specific approaches to docking e.g. SNUGDOCK (Sircar and Gray, 2010), although there appears ample scope for further improvement. In summary, the obstacles to generating a good model of a bound Ab should not be underestimated. However, when circumstances are favourable (which partly depends on the availability of good structural templates), and particularly when an experienced modeller is involved, success is certainly possible.

One context in which there is strong potential to build high quality models of bound Abs is where the model differs only slightly (e.g. by as little as a single residue) from that of a solved structure. Such analyses can be used to infer the impact of mutations to the epitope (e.g. in the context of antigenic escape by an evolving virus) or to the paratope (e.g. where the goal is to engineer a higher-affinity Ab). There are several general approaches for analysing the impact of mutations on the binding between two proteins that can be readily applied to the binding of Ab to Ag. These include online tools such as ANCHOR (Meireles *et al.*, 2010) and mCSM (Pires *et al.*, 2014) that predict changes in the binding affinity between proteins. A much more complex and time-consuming option, but one that has proved remarkably accurate in recent study focusing on broad-spectrum Abs binding to the stalk of influenza A (Lees *et al.*, 2017), is molecular dynamics (MD). MD involves the simulation of molecular forces between atoms and molecules, and can now be performed using relatively modest computational resources through the exploiting of cheap and fast graphical processing units (GPUs). It seems reasonable to assume that MD will make an increasingly important future contribution to the modelling of Abs and Ab/Ag complexes.

Prediction of T- and B-Cell Epitopes

B and T-cells recognize portions within their cognate antigens known as epitopes. The identification of the precise location of these epitopes in antigens is called epitope mapping and is relevant for understanding disease etiology, immune monitoring, developing diagnosis assays and for designing epitope-based vaccines. However, epitope mapping is hampered by the need for experimental testing on large arrays of potential epitope candidates. Luckily, researchers have developed *in silico* prediction methods that dramatically reduce the burden associated with epitope mapping by decreasing the list of potential epitope candidates. Here, we will review some of the most relevant tools for epitope prediction, with special focus on those that are available for free online use.

T-cell epitope prediction

T-cell epitopes are peptide antigens displayed on the surface of antigen-presenting cells (APCs) and bound to major histocompatibility complex (MHC) molecules. MHC molecules fall in two classes, class I (MHC I) and II (MHC II), that are recognized by two distinct sets of T-cells, CD8 T and CD4 T-cells, respectively (Fig. 1). Subsequently, there are two types of T-cell epitopes, CD8 and CD4.

CD8 T-cells become cytotoxic T lymphocytes (CTL) through a priming process that requires the recognition of peptide antigens presented by MHC I molecules on the cell surface of dendritic cells (DC). Thereby, CD8 T-cell epitopes are often known as CTL epitopes. Primed CTLs kill target T-cells displaying their cognate antigens bound to MHC I molecules, thus playing a key role in the clearance of infected and tumoral cells. On the other hand, DC-primed CD4 T-cells become helper (Th) or regulatory (Treg) T-cells that control the immune response through the production of cytokines and cell-to-cell contact.

T-cell epitope mapping consists of isolating the shortest amino acid sequence of an antigen that is able to stimulate either CD4 or CD8 T-cells (Ahmed and Maeurer, 2009). This capacity to stimulate T-cells is called immunogenicity and it is tested in assays requiring synthetic peptides derived from antigens (Ahmad Eweida and El-Sayed, 2016; Malherbe, 2009). There are many distinct peptides within antigens and T-cell prediction methods aim to identify those that are immunogenic. T-cell epitope immunogenicity is contingent on three basic steps: i) antigen processing, ii) peptide binding to MHC molecules and iii) recognition by cognate TCR. Of these three events, MHC-peptide binding is the most selective one at determining T-cell epitopes (Lafuente and Reche, 2009; Jensen, 2007). Therefore, prediction of peptide-MHC binding is the main basis to anticipate T-cell epitopes.

Prediction of peptide-MHC binding

MHC I and MHC II molecules have similar 3D-structures with bound peptides sitting in a groove delineated by two α -helices overlying a floor comprised of eight antiparallel β -stranded sheets (Fig. 2). However, there are also key differences between MHC I

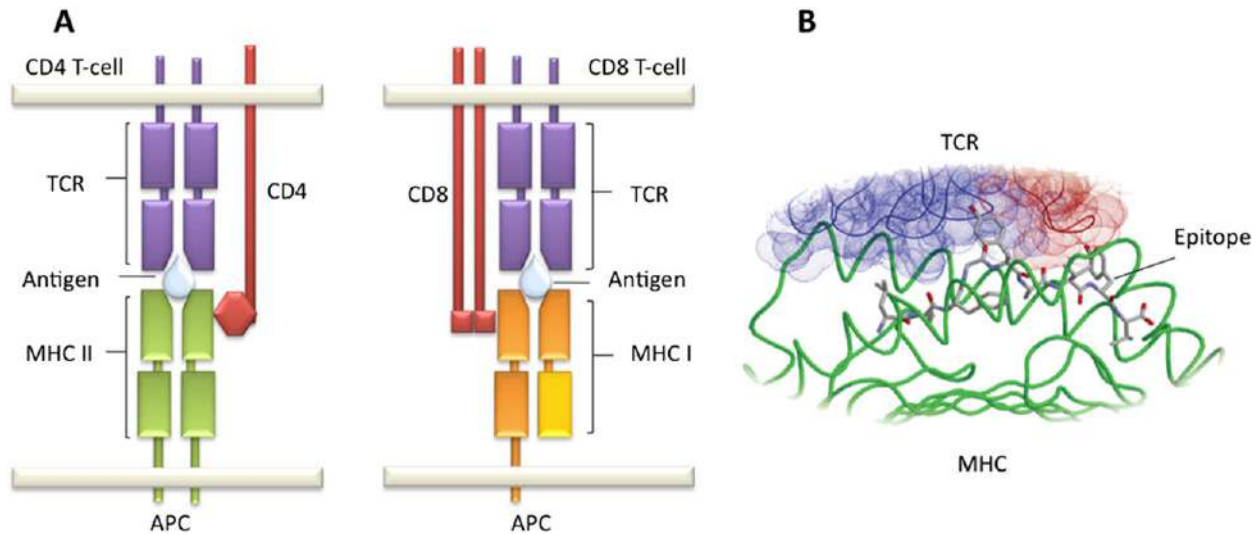


Fig. 1 T-cell epitope recognition (A) Schematic view of T-cell recognition. T cells recognize via T cell receptors (TCR) peptide antigens bound to MHC molecules displayed in the cell surface of antigen presenting cells. CD8 T-cells express CD8 co-receptor that binds to MHC I, while CD4 T-cells express the CD4 co-receptor, which binds to MHC II. (B) Close up of representative 3D-structure of a peptide-MHC I complex recognized by a TCR. MHC I molecule is depicted as green ribbons and the T cell epitope, peptide, in sticks. The TCR α and β chains are shown as blue and green ribbons, respectively, covered by atom surface mesh matching the color of the TCR chains. TCR recognition of peptide-MHC II complexes is similar to that of peptide-MHC I complexes.

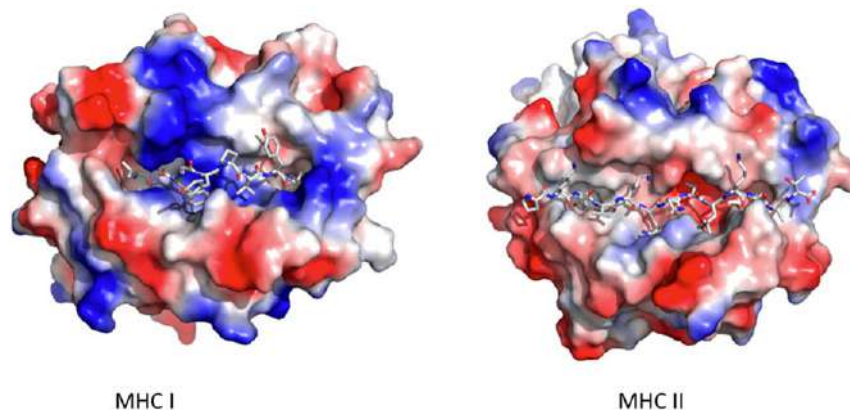


Fig. 2 Binding groove of MHC molecules. Figure depicts the molecular surface as seen by the TCR of representative MHC I and II molecules. The binding groove of MHC I molecules is closed while is open in MHC II molecules. As a result, MHC I molecules bind short peptides (8–10 amino acids), while MHC II molecules bind longer peptides (9–22 amino acids).

and II binding grooves that condition peptide-binding predictions. The peptide binding cleft of MHC I molecules is closed as it is made by a single chain. As a result, MHC I molecules can only bind short peptides ranging from 9 to 11 amino acids, whose N- and C-terminal ends remain pinned to conserved residues of the MHC I molecule through a network of hydrogen bonds (Stern and Wiley, 1994; Madden, 1995). The MHC I peptide binding groove also contains deep binding pockets with tight physico-chemical preferences that facilitate binding predictions. There is a complication however. Peptides that have different sizes and bind to the same MHC I molecule often use alternative binding pockets (Madden *et al.*, 1993). Therefore, methods predicting peptide-MHC I binding aim for a fixed peptide length; yet it is generally preferable to predict peptides with nine residues (9-mers) as most MHC I-peptide ligands have that size. In contrast, the peptide-binding groove of MHC II molecules is open, allowing the N- and C-terminal ends of a peptide to extend beyond the binding (Stern and Wiley, 1994; Madden, 1995). As a result, MHC II-bound peptides vary widely in length (9–22 residues), although only a core of nine residues (peptide binding core) sits into the MHC II binding groove. Therefore, peptide-MHC II binding prediction methods generally target to identify these peptide-binding cores. MHC II molecules binding pockets are also shallower and less demanding than those of MHC I molecules. Therefore, peptide-binding prediction to MHC II molecules is less reliable than that of MHC I molecules.

Given the relevance of the problem, there are myriad of methods to predict peptide-MHC binding (**Table 1**). They can be divided in two main categories: data-driven and structure-based methods. Structure-based approaches generally rely on modelling the peptide-MHC structure followed by evaluation of the interaction through methods such as molecular dynamics simulations (Lafuente and Reche, 2009; Desai and Kulkarni-Kale, 2014; Khan *et al.*, 2011; Khan and Ranganathan, 2011, Khan *et al.*, 2010;

Table 1 Selected T cell epitope prediction tools available online for free public use

Tool	URL	Method ^a	MHC Class	S	T	P	Ref.
EpiDOCK	http://epidock.ddg-pharmfac.net	SB	II	–	–	–	Atanasova <i>et al.</i> (2013)
MotifScan	https://www.hiv.lanl.gov/content/immunology/motif_scan/motif_scan	SM	I and II	X	–	–	(–)
Rankpep	http://imed.med.ucm.es/Tools/rankpep.html	MM	I and II	–	–	X	Reche <i>et al.</i> (2004)
SYFPEITHI	http://www.syfpeithi.de/	MM	I and II	–	–	–	Rammensee <i>et al.</i> (1999)
MAPPP	http://www.mpiib-berlin.mpg.de/MAPPP/	MM	I	X	–	X	Hakenberg <i>et al.</i> (2003)
PREDIVAC	http://predivac.biosci.uq.edu.au/	MM	II	–	–	–	Oyarzun <i>et al.</i> (2013)
PEPVAC	http://imed.med.ucm.es/PEPVAC/	MM	I	X	–	X	Reche and Reinherz (2005)
EPISOPT	http://bio.med.ucm.es/episopt.html	MM	I	X	–	–	Molero-Abraham <i>et al.</i> (2013)
Vaxign	http://www.violinet.org/vaxign/	MM	I and II	–	–	–	He <i>et al.</i> (2010)
MHCPred	http://www.ddg-pharmfac.net/mhcpred/MHCPred/	QSAR	I and II	–	–	–	Guan <i>et al.</i> (2003)
EpiTOP	http://www.pharmfac.net/EpiTOP	QSAR	II	–	–	–	Dimitrov <i>et al.</i> (2010)
BIMAS	https://www.bimas.cit.nih.gov/molbio/hla_bind/	QAM	I	–	–	–	Parker <i>et al.</i> (1994)
TEPITOPE	http://datamining-iip.fudan.edu.cn/service/TEPITOPEpan/TEPITOPEpan.html	QAM	II	–	–	–	Sturniolo <i>et al.</i> (1999)
Propred	http://www.imtech.res.in/raghava/propred/	QAM	II	X	–	–	Singh and Raghava (2001)
Propred-1	http://www.imtech.res.in/raghava/propred1/	QAM	I	X	–	X	Singh and Raghava (2003)
EpiJen	http://www.ddg-pharmfac.net/epijen/EpiJen/EpiJen.htm	QAM	I	–	X	X	Doytchinova <i>et al.</i> (2006)
IEDB-MHCI	http://tools.immuneepitope.org/mhci/	Combined	I	–	–	–	Zhang <i>et al.</i> (2008)
IEDB-MHCII	http://tools.immuneepitope.org/mhcii/	Combined	II	–	–	–	Zhang <i>et al.</i> (2008)
MULTIPRED2	http://cvc.dfci.harvard.edu/multipred2/index.php	ANN	I and II	X	–	–	Zhang <i>et al.</i> (2011)
MHC2PRED	http://www.imtech.res.in/raghava/mhc2pred/index.html	SVM	II	–	–	–	Bhasin and Raghava (2004a,b)
NetMHC	http://www.cbs.dtu.dk/services/NetMHC/	ANN	I	–	–	–	Nielsen <i>et al.</i> (2003)
NetMHCII	http://www.cbs.dtu.dk/services/NetMHCII/	ANN	II	–	–	–	Nielsen <i>et al.</i> (2007)
NetMHCpan	http://www.cbs.dtu.dk/services/NetMHCpan/	ANN	I	–	–	–	Nielsen <i>et al.</i> (2007)
NetMHCIIpan	http://www.cbs.dtu.dk/services/NetMHCIIpan/	ANN	II	–	–	–	Nielsen <i>et al.</i> (2008)
nHLApred	http://www.imtech.res.in/raghava/nhlapred/	ANN	I	–	–	X	Bhasin and Raghava (2007)
SVMHC	http://abi.inf.uni-tuebingen.de/Services/SVMHC/	SVM	I and II	–	–	–	Donnes and Elofsson (2002)
SVRMHC	http://us.accurascience.com/SVRMHCdb/	SVM	I and II	–	–	–	Liu <i>et al.</i> (2006)
NetCTL	http://www.cbs.dtu.dk/services/NetCTL/	ANN	I	X	X	X	Larsen <i>et al.</i> (2005)
WAPP	https://abi.inf.uni-tuebingen.de/Services/WAPP/index_html	SVM	I	–	X	X	Donnes and Kohlbacher (2005)

^aMethod used for prediction of peptide-MHC binding. Keys for methods: SM: sequence-motif; SB: structure-based; MM: Motif-matrix; QAM: quantitative affinity matrix; SVM: support vector machine; ANN: artificial neural network; QSAR: Quantitative structure–activity relationship model; Combined: Tool uses different methods including ANN and QAM, selecting the more appropriated for each distinct MHC molecule. The table also indicate whether the tool predict supertypes (S), TAP binding (T) and proteosomal cleavage (P); marked with an X.

Tong and Ranganathan, 2013). Structure-based methods have the great advantage of not needing experimental data. However, they are seldom used for they are computationally intensive and could exhibit a lower predictive performance than data-driven methods (Patronov and Doytchinova, 2013).

Data-driven methods for peptide-MHC binding prediction are based on peptide sequences that are known to bind to MHC molecules. These peptides sequences are generally available in specialized epitope databases such as IEDB (Vita *et al.*, 2015), EPIMHC (Molero-Abraham *et al.*, 2014), Antigen (Toseland *et al.*, 2005) and others (Singh and Mishra, 2016). Both MHC I and II binding peptides contain frequently occurring amino acids at particular peptides positions, known as anchor residues. Thereby, prediction of peptide-MHC binding was first approached using sequence motifs (SM) reflecting amino acid preferences of MHC molecules at anchor positions (D'Amaro *et al.*, 1995). However, it was soon shown that non-anchor residues also contribute to the capacity of a peptide to bind to a given MHC molecule (Bouvier and Wiley, 1994; Ruppert *et al.*, 1993). Subsequently, researchers developed motif-matrices (MM) that could evaluate the contribution of each and all peptide positions to the binding with the MHC molecule (Nielsen *et al.*, 2004; Rammensee *et al.*, 1999; Reche *et al.*, 2002; Reche and Reinherz, 2007). The most sophisticated form of motif-matrices consists of profiles (Reche *et al.*, 2002, 2004; Reche and Reinherz, 2007) that are similar to those used for detecting sequence homology (Gribbskov and Veretnik, 1996). Motif-matrices are often confused with quantitative affinity matrices (QAMs) since both produce peptide scores. However, MMs are derived without taking in consideration values of binding affinities and therefore resulting peptide scores are not suited to address binding affinity. In contrast, QAMs are trained on peptides and corresponding binding affinities, and aim to predict binding affinity. The first method based on QAMs was developed by Parker *et al.* (1994) (Table 1). Subsequently, various approaches were developed to obtain QAMs from peptide affinity data and predict peptide binding to MHC I and II molecules (Bui *et al.*, 2005; Nielsen *et al.*, 2007; Peters and Sette, 2005; Stumliolo *et al.*, 1999).

QAMs and motif-matrices assume an independent contribution of peptide-side chains to the binding. This assumption is well supported by experimental data but there is also evidence that neighbouring peptide residues interfere with others (Peters *et al.*, 2003). To account for those interferences, researchers introduced quantitative structure activity relationship (QSAR) additive models wherein the binding affinity of peptides to MHC is computed as the sum of amino acid contributions at each position plus the contribution of adjacent side chains interactions (Guan *et al.*, 2003). However, machine learning (ML) is the most popular and robust approach introduced to deal with non-linearity of peptide-MHC binding data (Lafuente and Reche, 2009). Researchers have used ML for two distinct problems: discrimination of MHC binders from non-binders and prediction of binding affinity of peptides MHC molecules.

For discrimination models, ML algorithms are trained on data sets consisting of peptides that either bind or do not bind to MHC molecules. Relevant examples of ML-based discrimination models are those based on artificial neural networks (ANNs) (Milik *et al.*, 1998; Brusic *et al.*, 1998), support vector machines (SVMs) (Donnes and Kohlbacher, 2005; Bhasin and Raghava, 2004a,b; Jacob and Vert, 2008), decision trees (DTs) (Zhu *et al.*, 2006; Savoie *et al.*, 1999) and Hidden Markov models (HMMs), that can also cope with non-linear data, have also been used to discriminate peptides binding to MHC molecules. However, unlike other ML algorithms, they have to be trained only on positive data. Three types of HMMs have been used to predict MHC-peptide binding: fully connected HMMs (Mamitsuka, 1998), structure-optimized HMMs (Zhang *et al.*, 2006) and profile HMMs (Zhang *et al.*, 2006; Lacerda *et al.*, 2010). Of those, only fully connected HMMs (fcHMMs) and structure-optimized HMMs (soHMMs) can model non-linear data, recognizing different patterns in the peptide binders. In fact, profile HMMs that are derived from sets of ungapped alignments (the case for peptides binding to MHC), are nearly identical to profile matrices (Durbin *et al.*, 1998) (Table 1).

With regard to predicting binding affinity, ML algorithms are trained on datasets consisting of peptides with known affinity to MHC molecules. Both SVMs and ANNs have been used for such purpose. SVMs were first applied to predict peptide binding affinity to MHCI molecules (Liu *et al.*, 2006) and later to MHC II molecules (Jandric, 2016) (Table 1). Likewise, ANNs were also applied first to the prediction of peptide binding to MHC I (Buus *et al.*, 2003; Nielsen *et al.*, 2003) and latter to MHC II molecules (Nielsen and Lund, 2009) (Table 1). Benchmarking of peptide-MHC binding prediction methods appears to indicate that those based on ANNs are superior to those based on QAMs and MMs. However, the differences between the distinct methods are marginal and vary for different MHC molecules (Yu *et al.*, 2002). Moreover, it has been shown that performance of peptide-MHC predictions is improved by combining several methods and providing consensus predictions (Wang *et al.*, 2008).

A major complication for predicting T-cell epitopes through peptide-MHC binding models is MHC polymorphism. In the human, MHC molecules are known as human leukocytes antigens (HLAs), and there are hundreds of allelic variants of class I (HLA I) and class II (HLA II) molecules. These HLA allelic variants bind distinct sets of peptides (Reche and Reinherz, 2003) and hence require specific models for predicting peptide-MHC binding. However, peptide-binding data is only available for a minority of HLA molecules. To overcome this limitation, some researchers have developed pan-MHC specific methods by training ANNs on input data combining MHC residues that contact the peptide with peptide-binding affinity that are capable of predicting peptide-binding affinities to uncharacterized HLA alleles (Nielsen *et al.*, 2007, 2008).

HLA polymorphism also hampers the development of T-cell epitope-based vaccines with wide population coverage as HLA variants are also expressed at vastly variable frequencies in different ethnic groups (Terasaki, 2007). Interestingly, different HLA molecules can also bind similar sets of peptides (Sette and Sidney, 1998, 1999) and researchers have devised methods to cluster them in groups, known as HLA supertypes, consisting of HLA alleles with similar peptide-binding specificities (Doytchinova and Flower, 2005; Greenbaum *et al.*, 2011; Lund *et al.*, 2004). The HLA-A2, HLA-A3, and HLA-B7 are relevant examples of supertypes; 88% of the population expresses at least an allele included in these supertypes (Reche and Reinherz, 2007; Sette and Sidney, 1998, 1999). Identification of promiscuous peptide-binding to HLA supertypes enables the development of T-cell epitope vaccines with high population coverage using a limited number of peptides. Currently, several web-based methods allow the prediction of promiscuous peptide-binding to HLA supertypes for epitope vaccine design including MULTIPRED (Zhang *et al.*, 2011) and PEPVAC (Reche and

Reinherz, 2005) (see Table 1). A method to identify promiscuous peptide-binding beyond HLA supertypes was developed and implemented by Molero-Abraham *et al.* (2013) with the name of EPISOPT. EPISOPT predicts HLA I presentation profiles of individual peptides regardless of supertypes and identifies epitope combinations providing a wider population protection coverage.

Prediction of antigen processing and integration with peptide-MHC binding predictions

Antigen processing shapes the peptide repertoire available for MHC binding and is a limiting step determining T-cell epitope immunogenicity (Zhong *et al.*, 2003). Subsequently, computational modelling of the antigen processing pathway provides a mean to enhance T-cell epitope predictions. Antigen presentation by MHC I and II molecules proceed by two different pathways. MHC II presents peptide antigens derived from endocytosed antigens that are degraded and loaded onto the MHC II molecule in endosomal compartments (Blum *et al.*, 2013). Class II antigen degradation is poorly understood and there is lack of good prediction algorithms yet (Hoze *et al.*, 2013). In contrast, MHC I molecules present peptides derive mainly from antigens degraded in the cytosol. The resulting peptides antigens are then transported to the endoplasmic reticulum by TAP where they are loaded onto nascent MHC I molecules (Blum *et al.*, 2013) (Fig. 3). Prior to loading, peptides often undergo trimming by ERAAP N-terminal amino peptidases (Hammer *et al.*, 2006).

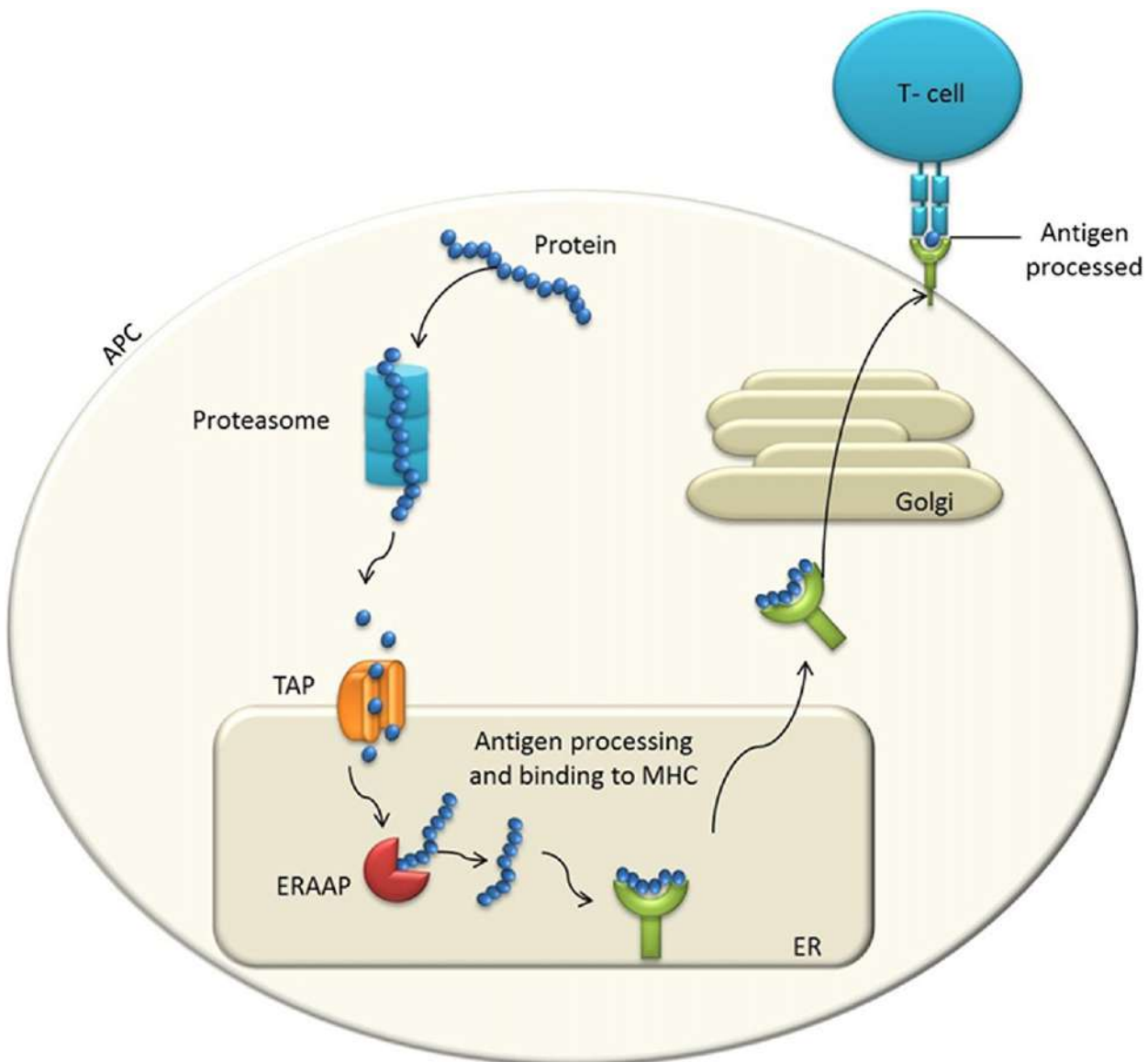


Fig. 3 Class I antigen processing. Figure depicts the major step involved in antigen presentation by MHC I molecules. Proteins are degraded by the proteasome and peptide fragments transported to the endoplasmic reticulum (ER) by TAP before being presented by MHC molecules. TAP transport peptides ranging from 8 to 16 amino acids. Long peptides often become suitable for binding to MCH I molecules after trimming at the N-terminus by ERAAP.

Proteosomal cleavage and peptide-binding to TAP have been studied in detail and there are computational methods to that predict both processes. Proteosomal cleavage prediction models have been derived from peptide fragments generated *in vitro* by human constitutive proteasomes (Nussbaum *et al.*, 2001; Holzthutter *et al.*, 1999) and from sets of MHC I restricted ligands mapped onto their source proteins (Nielsen *et al.*, 2005; Diez-Rivero *et al.*, 2010a,b; Bhasin and Raghava, 2005). On the other hand, TAP binding prediction methods have been developed by training different algorithms on peptides of known affinity to TAP (Bhasin and Raghava, 2004a,b; Diez-Rivero *et al.*, 2010a,b; Daniel *et al.*, 1998; Brusic *et al.*, 1999). Combination of proteosomal cleavage and peptide-binding to TAP with peptide-MHC binding predictions increases T-cell epitope predictive rate in comparison to just peptide-binding to MHC I (Donnes and Kohlbacher, 2005; Tenzer *et al.*, 2005; Bhasin and Raghava, 2004a,b; Doytchinova *et al.*, 2006; Larsen *et al.*, 2005). Subsequently, researchers have developed resources to predict CD8 T-cell epitopes through multistep approaches integrating proteosomal cleavage, TAP transport and peptide-binding to MHC molecules (Reche *et al.*, 2004; Donnes and Kohlbacher, 2005; Doytchinova *et al.*, 2006; Larsen *et al.*, 2005; Schubert *et al.*, 2015, 2016; Rammensee *et al.*, 1995) (Table 1).

T-cell epitope prediction concluding remarks

Current peptide-MHC binding predictions methods can readily anticipate T-cell epitopes from primary sequences. However, while it is possible to verify experimental binding data for most peptides predicted to bind to MHC molecules, only ~10% of those are shown to be immunogenic (able to elicit a T-cell response) (Zhong *et al.*, 2003). This scenario remains true when using T-cell epitope predictions methods that combine models for all the steps that determine T-cell epitope immunogenicity. Such a low T-cell epitope discovery rate is due to the fact that we do not have adequate models for antigen processing. In order to accelerate epitope identification and translational vaccine research, we must improve epitope immunogenicity prediction and define rationales and platforms for prioritizing protein antigens in epitope prediction and vaccine design (Flower *et al.*, 2010; Diez-Rivero and Reche, 2012; Molero-Abraham *et al.*, 2015). An alternative approach to overcome the lack of immunogenicity of predicted T-cell epitopes is to combine legacy experimentation, consisting of experimentally defined epitopes, with immunoinformatics predictions. This strategy was first conceived to assemble CD8 T-cell epitope vaccines (Reche *et al.*, 2006; Molero-Abraham *et al.*, 2013) and later extended to CD4 T-cell epitope vaccines (Sheikh *et al.*, 2016). Key criteria for epitope inclusion/selection are conservation and binding to multiple MHC molecules for maximum population protection coverage.

B-cell epitope prediction

B-cells recognize solvent exposed antigens through antigen receptors, B-cell receptor (BCR), consisting of membrane-bound immunoglobulins (Fig. 4). Upon activation, B-cells differentiate and secrete soluble forms of the immunoglobulin, also known as antibodies. A B-cell epitope or antigenic determinant is the antigen portion binding to the immunoglobulin or antibody. Antigens recognized by B-cells can be of different chemical nature but most of them are proteins and here will focus on them.

Any solvent exposed region in the antigen can be subject of recognition by antibodies. Nonetheless, B-cell epitopes can be divided in two main groups: linear and conformational (Fig. 5). Linear B-cell epitopes consist of sequential residues, peptides, whereas conformational B-cell epitopes consist of patches of solvent exposed atoms from residues that are not necessarily sequential (Fig. 5). Therefore, linear and conformational B-cell epitopes are also known as continuous and discontinuous B-cell epitopes, respectively. Antibodies recognizing linear B-cell epitopes can recognize denatured antigens, while denaturing the antigen results in loss of recognition for conformational B-cell epitopes. Most B-cell epitopes (approximately a 90%) are conformational and in fact only a minority of native antigens contains linear B-cell epitopes (Van Regenmortel, 2009).

B-cell epitopes are determined by structural studies that require solving the 3D-structure of antigen-antibody complexes using X-ray crystallography, nucleic magnetic resonance (NMR) or electron microscopy (EM). Alternatively, B-cell epitopes can be identified by functional assays in which the antigen is mutated and the interaction antibody-antigen is evaluated and by screening

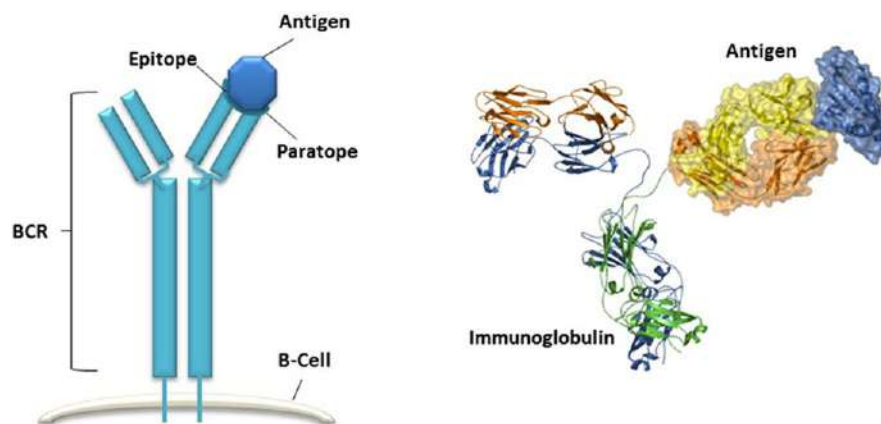


Fig. 4 Antigen-antibody interaction. B-cell epitopes are solvent exposed portions of the antigen that bind to immunoglobulins and antibodies. The portion of the antibody recognizing the epitope is known as paratope.

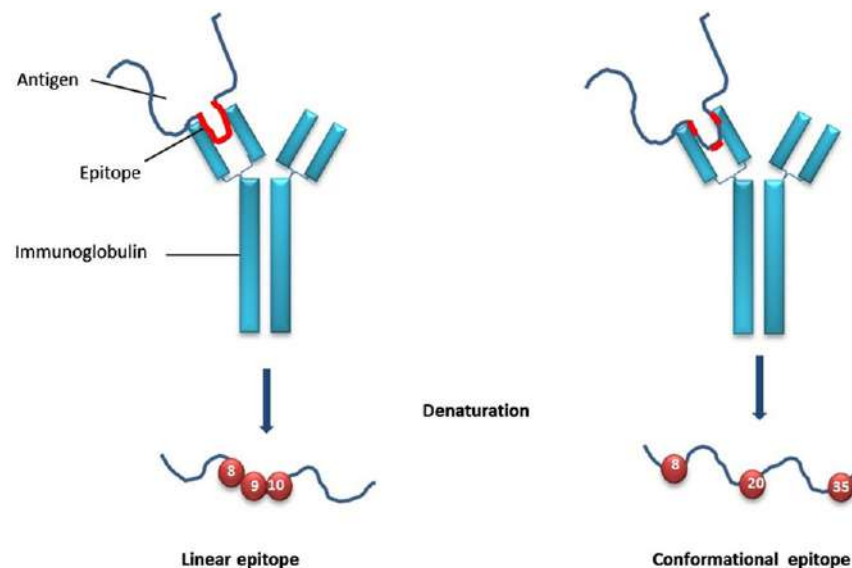


Fig. 5 Linear and conformational B-cell epitopes. Linear B-cell epitopes are composed of sequential/continuous residues, while conformational B-cell epitopes contain scattered/discontinuous residues along the sequence.

of peptide libraries for antibody binding (Potocnakova *et al.*, 2016). B-cell epitope prediction aims to facilitate B-cell epitope identification with the practical purpose of replacing the antigen for antibody production or for carrying structure-function studies.

Prediction of linear B-cell epitopes

For Linear B-cell epitopes are peptides they can easily be used to replace antigens for antibody production. Therefore, despite being a minority, prediction of linear B-cell epitopes has received major attention. Linear B-cell epitopes are predicted from the primary sequence of antigens using sequence-based methods. Early computational methods for the prediction of B-cell epitopes were based on simple amino acid propensities scales depicting physico-chemical features of B-cell epitopes. For example, Hoop and Wood applied residue hydrophilicity calculations for B-cell epitope prediction (Hopp and Woods, 1981, 1983) on the assumption that hydrophilic regions are predominantly located on the protein surface and are potentially antigenic. We know now however that protein surfaces contain roughly the same number of hydrophilic and hydrophobic residues (Lins *et al.*, 2003). Other amino acid propensity scales introduced for B-cell epitope prediction are based on flexibility (Karplus and Schulz, 1985), surface accessibility (Emini Hughes *et al.*, 1985) and β -turn propensity (Pellequer *et al.*, 1993). Current available bioinformatics tools to predict linear B-cell epitopes using propensity scales include PREDITOP (Pellequer and Westhof, 1993) and PEOPLE (Alix, 1999) (Table 2). PREDITOP (Pellequer and Westhof, 1993) uses a multi-parametric algorithm based on hydrophilicity, accessibility, flexibility and secondary structure properties of the amino acids. PEOPLE (Alix, 1999) uses the same parameters and in addition include the assessment of β -turns. A related method to predict B-cell epitopes was introduced by Kolaskar and Tongaonkar (Kolaskar and Tongaonkar, 1990) consisting of a simple antigenicity scale derived from physicochemical properties and frequencies of amino acids in experimentally determined B-cell epitopes. This index is perhaps the most popular antigenic scale for B-cell epitope prediction and it is actually implemented by GCG (Womble, 2000) and EMBOSS (Rice *et al.*, 2000) packages.

Comparative evaluations of propensity scales carried out on a dataset of 85 linear B-cell epitopes showed that most propensity scales predicted between 50%–70% of B-cell epitopes, with the β -turn scale reaching the best values (Pellequer *et al.*, 1993, 1991). It has also been shown that combining the different scales does not appear to improve the predictions (Pellequer and Westhof, 1993; Odorico and Pellequer, 2003). Moreover, Blythe and Flower (Blythe and Flower, 2005) demonstrated that single-scale amino acid propensity scales are not reliable to predict epitope location.

Poor performance amino acid scales for prediction of linear B-cell epitopes prompted the introduction of machine learning (ML)-based methods (Table 3). These methods are developed by training ML algorithms to distinguish experimental B-cell epitopes from non-B-cell epitopes. Prior to training B-cell epitopes are translated into features vectors capturing selected properties, such as those given by different propensity scales. Relevant examples of B-cell epitope prediction methods based on ML include BepiPred (Jespersen *et al.*, 2017), ABCpred (Saha and Raghava, 2006), LBtope (Singh *et al.*, 2013), BCPREDS (El-Manzalawy *et al.*, 2008) and SVMtrip (Yao *et al.*, 2012). Datasets, training features and algorithms used for developing these methods differ. BepiPred is based on random forests trained on B-cell epitopes obtained from 3D-structures of antigen-antibody complexes (Jespersen *et al.*, 2017). Both BCPREDS (El-Manzalawy *et al.*, 2008) and SVMtrip (Yao *et al.*, 2012) are based on support vector machines (SVM) but while BCPREDS was trained using various string kernels that eliminate the need for representing the sequence into fix length feature vectors, SMVtrip was trained on fix length tripeptide composition vectors. ABCpred and LBtope methods consist of artificial neural networks (ANN) trained on similar positive data, B-cell epitopes, but differ on the negative data, non B-cell epitopes. Negative data used for training ABCpred consisted on random peptides while negative data used for LBtope

Table 2 Selected B-cell epitope prediction methods available for free online use

<i>Tool</i>	<i>Method</i>	<i>Server (URL)</i>	<i>Ref.</i>
<i>Linear B cell epitope</i>			
PEOPLE	Propensity scale method	http://www.iedb.org/	Alix (1999)
BepiPred	ML (DT)	http://www.cbs.dtu.dk/services/BepiPred/	Jespersen <i>et al.</i> (2017)
ABCpred	ML (ANN)	http://www.imtech.res.in/raghava/abcpred/	Saha and Raghava (2006)
LBtope	ML (ANN)	http://www.imtech.res.in/raghava/lbtope/	Singh <i>et al.</i> (2013)
BCPREDS	ML (SVM)	http://ailab.ist.psu.edu/bcpred/	El-Manzalawy <i>et al.</i> (2008)
SVMtrip	ML (SVM)	http://sysbio.unl.edu/SVMTriP/prediction.php	Yao <i>et al.</i> (2012)
<i>Conformational B-cell epitope</i>			
CEP	Structure-based method (solvent accessibility)	http://bioinfo.ernet.in/cep.htm	Kulkarni-Kale <i>et al.</i> (2005)
DiscoTope	Structure-based method (surface accessibility and propensity amino acid score)	http://tools.iedb.org/discotope/	Haste Andersen <i>et al.</i> (2006)
ElliPro	Structure-based method (geometrical properties)	http://tools.iedb.org/ellipro/	Ponomarenko <i>et al.</i> (2008)
PEPITO	Structure-based method (physicochemical properties and geometrical structure)	http://pepito.proteomics.ics.uci.edu/	Sweredoski and Baldi (2008)
SEPPA	Structure-based method (physicochemical properties and geometrical structure)	http://lifecenter.sgst.cn/seppa/	Sun <i>et al.</i> (2009)
EPITOPIA	Structure-based method (ML- Naïve bayes)	http://epitopia.tau.ac.il/	Rubinstein <i>et al.</i> (2009)
EPSVR	Structure-based method (ML-SVR)	http://sysbio.unl.edu/EPSVR/	Liang <i>et al.</i> (2010)
EPIPREP	Structure-based method (ASEP, Docking)	http://opig.stats.ox.ac.uk/webapps/sabdab-sabpred/EpiPred.php	Krawczyk <i>et al.</i> (2014)
PEASE	Structure-based method (ASEP, ML)	http://www.ofranlab.org/PEASE	Sela-Culang <i>et al.</i> (2015a)
MIMOX	Mimotope	http://immunet.cn/mimox/helps.html	Huang <i>et al.</i> (2006)
PEPITOPE	Mimotope	http://pepitope.tau.ac.il/	Mayrose <i>et al.</i> (2007)
EpiSearch	Mimotope	http://curie.utmb.edu/episearch.html	Negi and Braun (2009)
MIMOPRO	Mimotope	http://informatics.nenu.edu.cn/MimoPro	Chen <i>et al.</i> (2011)
CBTOPE	Sequence based (SVM)	http://www.imtech.res.in/raghava/cbtope/submit.php	Ansari and Raghava (2010)

consisted of experimentally validated non B-cell epitopes form IEDB (Vita *et al.*, 2015). In general, B-cell epitope prediction methods based on ML-algorithm are reported to outperform those based on amino acid propensity scales. However, some authors have reported that ML algorithms show little improvement over single scale-based methods (Greenbaum *et al.*, 2007).

Prediction of conformational B-cell epitopes

Most B-cell epitopes are conformational and yet, prediction of conformational B-cell epitopes has lagged behind that of linear B-cell epitopes. There are two main practical reasons for that. First of all, prediction of conformational B-cell epitopes requires in general knowing the protein three-dimensional (3D)-structure and this information is only available for a fraction of the proteins (Levitt, 2009). Secondly, isolating conformational B-cell epitopes from their protein context for selective antibody production is a difficult task that requires suitable scaffold for epitope grafting (Levitt, 2009). Thereby, prediction of conformational B-cell prediction is currently of little relevance for epitope-vaccine design and antibody-based technologies. Nonetheless, prediction of conformational B-cell epitopes is relevant for carrying structure-function studies involving antibody-antigen interactions.

Table 3 A summary list of the best-known and recent databases relevant in immunology research

Databases	Description	References
AAgAtlas http://aagatlas.ncpsb.org	Human autoantigen database	Wang <i>et al.</i> (2017)
ALPSbase http://www.niaid.nih.gov/topics/alps/Pages/default.aspx	Autoimmune lymphoproliferative syndrome database	Puck (2005)
AntigenDB http://www.imtech.res.in/raghava/antigen	Sequence, structure and other data on pathogen antigens	Ansari <i>et al.</i> (2010)
AntiJen http://www.ddg-pharmfac.net/antijen/AntiJen/antijenhomepage.htm	Quantitative binding data for peptides and proteins of immunological interest	Toseland <i>et al.</i> (2005)
BCIpep http://www.imtech.res.in/raghava/bcipep/	Database of experimentally determined B-cell epitopes of antigenic proteins	Saha <i>et al.</i> (2005)
bNAber http://bnaber.org/	A database of broadly neutralizing HIV – 1 antibodies	Eroshkin <i>et al.</i> (2014)
dbMHC http://www.ncbi.nlm.nih.gov/gv/mhc/	dbMHC provides access to HLA sequences, tools to support genetic testing of HLA loci, HLA allele and haplotype frequencies of over 90 populations worldwide	Helmberg <i>et al.</i> (2004)
DIGIT http://circe.med.uniroma1.it/digit	Database of immunoglobulin variable domain sequences annotated with the type of antigen, the germline sequences and pairing information between light and heavy chains	Chailyan <i>et al.</i> (2012)
EPIMHC http://imed.med.ucm.es/epimhc/	A curated database of MHC-binding peptides for customized computational vaccinology	Reche <i>et al.</i> (2005)
Epitome http://roslab.org/services/epitome/	Database of all known antigenic residues and the antibodies that interact with them	Schlessinger <i>et al.</i> (2006)
GPX-Macrophage Expression Atlas http://gpxmea.gti.ed.ac.uk/	An online resource for expression based studies of a range of macrophage cell types following treatment with pathogens and immune modulators	Grimes <i>et al.</i> (2005)
HaptenDB http://www.imtech.res.in/raghava/haptendb	Database of hapten molecules, small molecule not immunogenic by itself, that can react with antibodies of appropriate specificity	Singh <i>et al.</i> (2006)
HPTAA http://www.bioinfo.org.cn/hptaa/	Database of potential tumor-associated antigens that uses expression data from various expression platforms, including carefully chosen publicly available microarray expression data, GEO SAGE data and Unigene expression data	Wang <i>et al.</i> (2006)
IEDB http://www.iedb.org/	The Immune Epitope Database (IEDB) provides a catalog of experimentally characterised B and T cell epitopes, as well as data on MHC binding and MHC ligand elution experiments	Vita <i>et al.</i> (2015)
EDB – 3D http://www.immuneepitope.org/bb_structure.php	Structural data within the Immune Epitope Database	Ponomarenko <i>et al.</i> (2011)
IL2Rgbase http://research.nhgri.nih.gov/scid/	X-linked severe combined immunodeficiency mutations	Puck (1996)
IMGT http://www.imgt.org/	The international ImMunoGeneTics information system (iMGT) is a high-quality integrated knowledge resource specialized in IG, T cell receptors and MHC of human and other vertebrate species and related proteins of the immune system (RPI)	Lefranc <i>et al.</i> (2005)
IMGT/GENE-DB http://www.imgt.org/IMGT_GENE-DB/GENEelect?livret=0/	IMGT/GENE-DB is part of IMGT database and allows a search of IG and TR gene entries by locus, group, subgroup, based on the classification axiom of IMGT-ONTOLOGY	Giudicelli and Lefranc (1999)
IMGT/HLA http://www.ebi.ac.uk/imgt/hla/	Database of recognized HLA alleles and their sequences for performing sequence alignments	Robinson <i>et al.</i> (2000)
IMGT/LIGM-DB http://www.imgt.org/ligmdb/	IMGT/LIGM-DB is a database of immunoglobulin and T cell receptor nucleotide sequences	Giudicelli <i>et al.</i> (2006)
IMGT/mAb-DB http://www.imgt.org	Database for therapeutic monoclonal antibodies	Lefranc <i>et al.</i> (2015)
ImmuNet http://immunet.princeton.edu	Database of immunological networks and functional relationships between molecular entities, able to predict disease-associated genes	Gorenshteyn <i>et al.</i> (2015)
InnateDB http://www.innatedb.com/	Database of genes, proteins, experimentally-verified interactions and signalling pathways involved in the innate immune response of humans, mice and bovines to microbial infection	Lynn <i>et al.</i> (2008)
Interferon Stimulated Gene Database http://www.lerner.ccf.org/labs/williams/xchip-html.cgi	A database of interferon stimulated genes (ISGs)	Samarajiwa <i>et al.</i> (2009)
IPD - Immuno Polymorphism Database http://www.ebi.ac.uk/ipd/	Databases of polymorphic genes in the immune system	Robinson <i>et al.</i> (2003)

Table 3 Continued

Databases	Description	References
IPD-ESTDAB http://www.ebi.ac.uk/ipd/estdab/	IPD-ESTDAB is a database of immunologically characterised melanoma cell lines	Robinson <i>et al.</i> (2003)
IPD-HPA - Human Platelet Antigens http://www.ebi.ac.uk/ipd/hpa/	IPD-HPA is a database of alloantigens expressed only on platelets	Robinson <i>et al.</i> (2003)
IPD-KIR - Killer-cell Immunoglobulin-like Receptors http://www.ebi.ac.uk/ipd/kir/	IPD-KIR contains the allelic sequences of Killer-cell Immunoglobulin-like Receptors	Robinson <i>et al.</i> (2003)
IPD-MHC http://www.ebi.ac.uk/ipd/mhc/	IPD-MHC is a database of sequences of the major histocompatibility complex of different species	Robinson <i>et al.</i> (2003)
MHCBN http://www.imtech.res.in/raghava/mhcbn/	MHCBN is a comprehensive database comprising over 23000 peptides sequences, whose binding affinity with MHC or TAP molecules has been assayed experimentally.	Bhasin <i>et al.</i> (2003)
MPID-T2 http://biolinfo.org/mpid-t2/	MPID-T2 is a highly curated database for sequence-structure-function information on MHC-peptide interactions	Kanguane <i>et al.</i> (2001)
MUGEN Mouse Database http://www.mugen-noe.org/database/	Murine models of immune processes and immunological diseases	Aidinis <i>et al.</i> (2008)
Protegen http://www.violinet.org/protegen/	Protective antigen database and analysis system	Yang <i>et al.</i> (2011)
SABDab http://opig.stats.ox.ac.uk/webapps/sabdab	Structural Antibody Database	Dunbar <i>et al.</i> (2014)
SYFPEITHI http://www.syfpeithi.de/	A database of MHC ligands and motifs	Rammensee <i>et al.</i> (1995)
SuperHapten http://bioinformatics.charite.de/superhapten/	Database integrating information from literature and web resources about haptens	Günther <i>et al.</i> (2007)
VBASE2 http://www.vbase2.org/	Database of germ-line V genes from the immunoglobulin loci of human and mouse	Retter <i>et al.</i> (2005)

There are several methods available to predict conformational B-cell epitopes (Table 2). The first to be introduced was CEP (Kulkarni-Kale *et al.*, 2005) which relied almost entirely on predicting patches of solvent exposed residues. It followed DiscoTope (Haste Andersen *et al.*, 2006), which in addition to solvent accessibility, considered amino acid statistics and spatial information to predict conformational B-cell epitopes. An independent evaluation of these two methods using a benchmark dataset of 59 conformational epitopes revealed that they did not exceed 40% of precision and 46% of recall (Ponomarenko and Bourne, 2007). Subsequently, more methods were developed like ElliPro (Ponomarenko *et al.*, 2008), that aims to identify protruding regions in antigen surfaces, and PEPITO (Sweredoski and Baldi, 2008) and SEPPA (Sun *et al.*, 2009) that manage to combine single physicochemical properties of amino acids and geometrical structure properties. The reported Area Under the Curve (AUC) of these methods is around 0.7, which is indicative of poor discrimination capacity yet better than random. However, in an independent evaluation SEPPA reached an AUC of 0.62 while all the mentioned methods had AUC around 0.5 (Xu *et al.*, 2010). ML has also been applied to predict conformational B-cell epitopes in 3D-structures. Relevant examples include EPITOPIA (Rubinstein *et al.*, 2009) and EPSVR (Liang *et al.*, 2010) which are based on naïve bayes and support vector regression, respectively, train on features vectors combining different scores. The reported AUC of these two methods is around 0.6.

The above methods for conformational B-cell epitope prediction identify generic antigenic regions regardless of antibodies that are ignored. However, there are also methods for antibody-specific epitope prediction (Sela-Culang *et al.*, 2015b). This approach was pioneered by Soga *et al.* (2010) who defined an antibody-specific epitope propensity (ASEP) index after analysing the interfaces of antigen-antibody 3D-structures. Using this index, they developed a novel method of predicting epitope residues for individual antibodies that worked by narrowing down candidate epitope residues predicted by conventional methods. More recently Krawczyk *et al.* (2014) developed EpiPred, a method that uses a docking like approach to matchup antibody-antigen structures and thus identify epitope regions on the antigen. A similar approach is used by PEASE (Sela-Culang *et al.*, 2015a) only that the method utilizes the sequence of the antibody and the 3D-structure of the antigen. Briefly, for each pair of antibody sequence and antigen structure, PEASE uses a machine-learning model trained on properties from 120 antibody-antigen complexes to identify pair combination of residues from CDRs of the antibody and the antigen that are likely to interact.

Another method to identify conformational B-cell epitopes in a protein with a known 3D structure is through mimotope-based methods. Mimotopes are peptides selected from randomized peptide libraries for their ability to bind to an antibody raised against a native antigen. Mimotope-based methods require as input antibody affinity-selected peptides and the 3D structure of antigen. Examples of bioinformatics tools for conformational B-cell epitope prediction using mimotopes include MIMOX (Huang *et al.*, 2006), PEPITOPE (Mayrose *et al.*, 2007), EPISEARCH (Negi and Braun, 2009), MIMOPRO (Chen *et al.*, 2011) and PEPMAPPER (Chen *et al.*, 2012) (Table 2).

Methods for conformational B-cell epitope prediction require the 3D-structure of the antigen. Exceptionally, however, Ansari and Raghava (2010) developed a method (CBTOPE) for the identification of conformational B-cell epitope from the primary sequence of the antigen. CBTOPE is based on SVM trained on physicochemical and sequence-derived features of conformational B-cell epitopes. CBTOPE reported accuracy was 86.6% in cross-validation experiments.

B-cell epitope prediction concluding remarks

Prediction B-cell epitopes is clearly relevant for epitope-vaccine design and for developing pharmaceutical and biotech applications based on antibody-antigen interactions. As a result, there are many methods available to predict both conformational and linear B-cell epitopes. However, practical application of B-cell epitope prediction has seldom arrived for different reasons. First of all, prediction of B-cell epitopes is still unreliable for both linear and conformational B-cell epitopes. Secondly, linear B-cell epitopes do usually elicit antibodies that do not cross-react with native antigens. Third, the great majority of B-cell epitopes are conformational and yet predicting conformational epitopes has few applications, as they cannot be isolated from their protein context. Under this scenario, it is key to improve current methods for B-cell epitope prediction and more importantly to develop approaches and platforms for epitope grafting onto suitable scaffolds capable of replacing the native antigen.

Modelling Signalling Pathways in Immunology

The study of immunology includes also the study of immune signalling networks (Dower and Qvarnstrom, 2003). Immunological networks oscillate from the macro to the micro level and within this range the breadth of computational tools co-adjuvate both in vitro and in vivo studies (Kidd *et al.*, 2014).

Several signalling pathway components, such as kinases, phosphatases, adaptor molecules and molecular scaffolds are essential to orchestrate extracellular signals with distinct cellular function. In immune context, cellular signalling leads to the activation of different cells owing specific immune activities (Harwood and Batista, 2010; Kawai and Akira, 2011). Specific ligands bind to a specific receptor on the cellular membrane of an immune system cell to trigger biochemical reactions and signal transduction pathways. Cytokines are secreted by immune cells in response to cellular signalling; they bind to specific membrane receptors to propagate the signal through second messengers, such as tyrosine kinase complexes, with the aim to modify cellular activity and regulate gene expression (Turnera *et al.*, 2014). Interleukins represent the largest class of cytokines and are secreted by a specific leukocyte that indirectly acts on other ones recruited as signalling ligands.

Understanding and targeting these networks may be the key for the development of novel immunomodulatory therapies and the treatment of immune dysfunction and dysregulation disorders. Graph theory (Biggs *et al.*, 1986) and dynamical modelling (Banks *et al.*, 2017) represent two major computational approaches used to study signalling networks (Janes and Lauffenburger, 2013). Both approaches result useful: specifically, network analysis and graph theory improve the understanding of signalling network architecture and organization, and predict the information-processing capabilities of the model through statistical methods, whereas dynamical modelling is helpful to determine how the entire system varies in time and space upon receiving specific and simultaneous stimuli, such as in cross-talk phenomena. Dynamic modelling approaches can be divided in two classes: deterministic and stochastic models. Deterministic models usually use ordinary differential equations (ODEs) based systems of biochemical interactions. Stochastic models use Gillespie's algorithm as the standard approach for computations (Gillespie, 2007).

A new computational approach could be represented by a combination of methodologies consisting of agent-based technique implemented with signalling metabolic pathway analysis. A combination model like this, could be considered exemplary in precision medicine due to the fact that these models own the potential to create therapeutic protocols tailored to a patient's specific biology, needs and health conditions. Indeed, it could be used to better understand the pathogenic mechanisms of autoimmune disease such as multiple sclerosis disorder and to suggest predictive information capabilities. The latter concerns the effectiveness of drugs in typical sclerosis multiple's patient scenario, according to his/her own genetic risk factors, anamnesis, average number and typologies of relapses. To facilitate the design and the simulation of the dynamics of immune system related signalling pathways, specific software like Complex Pathways Simulator (COPASI), may help to this aim (Hoops *et al.*, 2006).

Tools for Modelling Cellular Immune System Dynamics

During the years, many models and tools have been developed to mimic the immune system functions at the cellular scale. However, all the adopted strategies can be grouped in two main approaches: the top-down and the bottom-up approaches.

The first approach is represented by so-called top-down approach. The top down approach works by looking down from above to the problem and estimates the behaviour at the macroscopic level. This is usually obtained by the use of Differential Equations (DE). The DE-based models are all population-based, and the spatiality and topology which both depend on individual interactions are, in general, ignored. Historically, the first models that have been developed for representing the immune system function at the cellular scale make use of such an approach thanks to the possibility to borrow the well-established mathematical techniques widely used for engineering, mathematics and physics. The simplest models based on the top-down approach make use of Ordinary Differential Equations (ODE). For very simple ODE models it is possible to take advantage of years of mathematical studies and methods that can bring out analytical solutions, steady states and asymptotic behaviour. However, as the biological description grows, they become almost intractable and can be only solved numerically. The application of some extension, such as the use of Partial Differential Equations (PDE) can bring some spatial description, whereas the use of stochastic differential equations (SDE) and delayed differential equations (DDE) can allow to reproduce stochastic effects and temporal delays, respectively. However, all of these usually entitle instability problems, a higher computational effort and numerical simulations only. Another problem of differential equations is given by their formalism. The mathematical "language" used is not

commonly known to biologists, making the communication with mathematicians/computer scientists even hard, and damaging the interdisciplinary collaboration. Despite of that, models based on differential equations have been successfully used for years bringing out important results and helping in understanding, verifying and validating the basic concepts of immunology.

Many equation-based models have been presented in the literature. In the field of tumor immunology, a simple model for effector cells and tumor cells has been defined by Kuznetsov and co-workers (Kuznetsov *et al.*, 1994). In their model, a threshold above which there is uncontrollable tumor growth is predicted. Below such threshold, the disease can be attenuated with periodic exacerbations occurring every 3–4 months. DeLisi and Rescigno (1977) and Adam (1996) take also into account, besides of IS populations and tumor cells, the external stimulation of immune cells, showing that such a stimulation may increase survival. On the other hand, it has been possible to show that, in some particular scenario, an increase in the number of effector cells may increase the chance of tumor survival. Nani and Ögütörel (1994) include adoptive cellular immunotherapy in their model against tumor growth. This model takes into account stochastic effects (SDE) on the IS-cancer interactions, showing that success of treatment is dependent on the initial tumor burden. Kirschner and co-workers (Kirschner and Webb, 1997) used several differential equations to demonstrate the progression HIV infection by including some important characteristics of the pathology as well as the effects of a combined drug therapy. The model by Nowak *et al.* (1991) indicated that virus diversity is able to speed up the infection progress of HIV.

Gullo *et al.* (2015) developed an ODE model for predicting and improving the expansion of human cord blood CD133 + hematopoietic stem/progenitor cells. In this model, no pathology is reproduced. The combined effects (survival, duplication and differentiation) of different cytokines combinations on progenitor stem cells are instead predicted to suggest the best cytokine combination that maximizes duplication, while minimizing differentiation towards other cell populations. Another example of the immune system – tumor interaction under the actions of an external stimulus can be found in the work by Bianca *et al.* (2012). In their model the authors describe with sufficient degree of detail both the humoral and cellular response of the immune system to the tumor associated antigens and the recognition process between B cells, T cells and antigen presenting cells. The control of the tumor cells growth occurs through the definition of different vaccine protocols with good agreement with in vivo experiments on transgenic mice.

The bottom-up approach works at the opposite side by observing the entities behaviour individually, and thus representing the system at a lower, microscopic level. With the bottom-up approach, unexpected complex global properties can “emerge” as the sum of the rather simple individual behaviours (emergent properties). This approach usually requires greater computational power in order to simulate a significant number of entities, and for this reason it has been pushed only in the last years following the gains in computational performance achieved in the computer world. Cellular automata (CA) and (Multi)Agent-based methods are the most used bottom-up approaches.

CA are composed by a set of identical elements, called cells, each of one occupying a node of a regular and discrete spatial network (a lattice of n -tuples of integer number). The possible states of a cell are discrete and belong to a finite set of states. CA evolve using discrete time steps, and all cells change their states according to a local rule δ (or transition function) homogeneously applied at every step which depends on cell state itself and on the state of the cells in the neighbourhood. The CA-based approach was adopted to reproduce the IS behaviour mainly to study their ability to self-adapt and regulate. In this field, the papers by Santos (1999) and Hershberg *et al.* (2001) showed how, with CA, it was possible to study the HIV-immune dynamics in physical space and Shape Space, respectively. Grilo *et al.* (1999) combined of Genetic Algorithms and CA to represent the dynamical changes that may occur in the virus - the immune cells equilibrium. In Beauchemin *et al.* (2005), Beauchemin and co-workers present a simple two-dimensional CA model of influenza infection. CA have been also used to reproduce the IS-Tumor interactions (Mallet and De Pillis, 2006). Other CA-based models for the IS can be found in Bandini (1996), Hu and Ruan (2003) and Perelson (2002).

Agent based models can be seen as an extension of CA. ABM suppose the presence of individuals (agents) that are usually placed on a simulation space (i.e. a lattice). Such agents can be of heterogeneous nature, can have internal properties (i.e. lifetime, internal state and energy), and can act and take decisions (i.e. move, interact with other agents in their neighbourhood, modify their internal state or die) individually or as a result with the interaction with other agents. Models based on CA and ABM have various advantages. By definition, they can be stochastic and include both delays and a spatial description. Furthermore, they allow a richer description of the biological aspects. As a consequence of that, approximations are usually more biological in character than mathematical. Nonlinearities are managed easily and it is always possible to add complexity or modify the model without introducing any new difficulties in solving it. Such methods are also intrinsically numerically stable thanks to the fact that most of the variables representing the attributes of the entities are integers and very few floating point operations are required.

Guo and co-workers validated the three stages of HIV infection by using a multi-agent model (Guo *et al.*, 2005). Perrin and co-workers emphasized the diversity and mutation of HIV virus, which are important characteristics that influence the immune response latency (Perrin *et al.*, 2006). Jacob *et al.* (2004) presented a swarm-based, three-dimensional model for the human immune system, innate response and adaptive response. The model takes on a strengthening reaction to the previous encountering pathogen, giving a simple representation of immune memory mechanisms.

C-IMMSIM and SIMTRIPLEX are ABM models developed in the C programming language (Bernaschi and Castiglione, 2001; Palladini *et al.*, 2010), with focus on improved efficiency and simulation size and complexity. These simulators, initially developed for HIV and mammary carcinoma, respectively, mimic the receptor interactions by using binary strings. Many mechanism of the immune system (i.e., memory, specificity, tolerance, homeostasis etc.) are here represented. In these simulators, the IS response is designed and coded to allow simulations considering millions of cells with a very high degree of complexity.

The same computational framework developed with SimTriplex has been also used to predict the effects of candidate vaccine adjuvants derived by citrus in boosting the immune system activation (Pappalardo *et al.*, 2016), to investigate the induced immune system response against B16-melanoma (Pappalardo *et al.*, 2011), and, when combined in a hybrid fashion with a lattice Boltzmann (an equation based method), to model the avascular tumor growth under nutrient diffusion, also including the relative immune response (Alemani *et al.*, 2012). SIMMUNE investigates how context adaptive behaviour of the IS might emerge from local cell-cell and cell-molecule interactions (see Relevant Websites section). It is based on molecule interactions on a cells surface. Cells do not have states but behaviours that depend on rules based on cellular response to external stimuli. CYCELLS (Warrender *et al.*, 2006), designed for studying intercellular interactions, uses a hybrid model that represents molecular concentrations continuously and cells discretely. Each type of molecular signal (e.g. cytokines) is represented by a decay and diffusion rate.

Finally, ABM specific development tools have also been used to model the Immune system behaviour, as in the model by Pennisi *et al.* (2013). In their work the authors use an ABM development framework named NetLogo to reproduce the complex cross regulation mechanisms that occur between CD8 T cells and T regulatory cells and show how the appearing of relapsing-remitting multiple sclerosis can occur in predisposed patients that also present breakdown of such regulatory functions.

In the last few years also Petri Nets (PN) have been applied to catch up the immune system dynamics. PN are graphical modelling tool initially developed to reproduce the behaviour of concurrent distributed systems. PN include some graphical elements such as places (represented by empty circles) to represent the possible states of a system, transitions (represented by boxes) to represent possible changes in the states of the system, and arrows to connect places to transitions and vice-versa. Inside places we can find the tokens (small black spheres) to quantify the occurrences or the number of elements in given state. The happening of an event is described by the fact that tokens are taken from input places and consumed by transitions, that can then produce some new tokens in the output places according to well-defined rules. Many analytical methodologies for studying the properties of a PN model have been also presented. A brief summary of PN related methodologies can be found in Murata (1989).

PN have been increasingly applied in other fields of engineering, computer science, chemistry and, lastly, in biology for modelling signalling pathways. Their application to model the immune system function is relatively new and shows how it is possible to apply a particular extension of PN (the Fuzzy continuous PN) for the modelling of the Immune system response (Park *et al.*, 2006). However, such an approach presented some flaws; among them, the biggest one was represented by the fact that it did not allow to distinguish among different kind of cells that have different internal states, receptors, or a different position in the simulation space.

Recently, Pennisi and co-workers presented a novel methodological approach based on Coloured Petri Nets (CPN) to model the immune system function (Pennisi *et al.*, 2016). CPN associate “colours” to tokens. Colours can describe token specific properties and/or internal states (i.e., life-time, position, receptor) and then it is possible to discern among tokens with different internal states. Through a simple example model of the humoral immune system response that was able to represent some of the most complex features of the adaptive IS response like memory and specificity, such a methodology demonstrated able to allow a good level of granularity in the description of cells behaviour, without losing the possibility to have some sort of qualitative analysis, and thus positioning itself somewhat in the middle between top-down and bottom-up approaches.

Immunological Databases

Over recent years database resources have become a fundamental tool for immunological research, due to the growing explosion of extensive information from advanced biotechnology that produce large amounts of biological data. The access to immunological databases is increasingly widespread in immunological research for extracting existing information, planning experiments and analysing experimental results. Immunological data include biological sequences, antigenic and epitope structures, immunogenetic properties and specificity of immune interactions. Immunological databases include both bioinformatic resources in which immunological data are stored and computational methodologies suitable for the extraction and the analysis of information data. These tools have become increasingly sophisticated to allow a speed up of the discovery process, a quick identification of sequences of interest and an achievement of significant bibliographic, taxonomic and feature information.

Repertoire Sequencing

The advent of comparatively cheap Next Generation Sequencing (NGS) is transforming many areas of biology, and immunology is no exception. NGS is a term used to describe a number of high-throughput sequencing technologies that typically generate large numbers of fairly short reads spanning part of a longer nucleic acid sequences. There are many potential applications in this field, ranging from the identification of polymorphisms in immune-related genes, to understanding how rapidly-mutating viruses evolve within a single individual in response to host immune pressure. But many of the most interesting and challenging applications involve characterizing the properties of T-cell and/or B-cell repertoires by sequencing the genomic DNA or messenger RNA that encodes individual TCRs and Abs, a technique known as Rep-seq. The discussion here focuses on Ab repertoire sequencing in order that the additional complexities associated with somatic hypermutation can be discussed, but in other respects the similarities with T-cell repertoire sequencing are considerable.

Ideally, we would like to capture the whole repertoire – paired, full-length sequences of Abs – at sufficiently frequent intervals to elucidate the dynamic properties of interest (potential applications are considered below), but there are important practical

limitations. Firstly, there are limits to what proportion of the repertoire is present in a biological sample – only around 2% of the full human repertoire is found in peripheral blood, the commonest type of sample (much larger proportions are present in lymph nodes and spleens). Secondly, the choice of sequencing platform affects both read length and depth-of-coverage (i.e. the number reads spanning a given base in the target sequence). Here it is worth considering the specific challenges posed by Ab repertoires as targets for NGS.

Standard RNA-seq with paired-end read lengths of around 100 (2×50) or 150 (2×75) nucleotides is the norm for genomic studies. These short sequences are then assembled, often using a reference genome, into full-length sequences. However, the pattern of variability within a typical set of Ab sequences – with highly variable regions (the CDRs) separated by less variable regions (the framing regions) – means that naïve approaches to sequence assembly risk generating large numbers of chimeric artefacts, i.e., sequences that do not exist *in vivo*. For this reason, Rep-seq is generally undertaken using reads that are long enough to span all, or most, of the region of interest – which, for the heavy chain, spans all three CDRs and, for some applications, the part of the constant region that determines Ab isotype. In practice, many different read-lengths have been used with 600 (2×300) nucleotides (Illumina MiSeq) arguably the most popular current option. Although no assembly is required, this length is still shorter than one would choose, as it means that the 5' end of each read starts within a comparatively variable region of the Ab sequence. To address this problem, it is typically necessary to use a large set of V-gene-specific primers.

Given the high diversity of Ab sequences, a further challenge arises from the combination of sequencing depth constraints and sequencing error rates (approximately 2.5×10^6 reads per sample and 10^{-4} per base respectively with Illumina MiSeq). The number of reads is much lower than the actual repertoire diversity; in practice, this means that it is impossible to distinguish between a true sequence polymorphism observed only once and a sequence error. This problem can be partially solved by adding a molecular barcode to each sequence prior to PCR amplification; sequences containing errors that share the same barcode (and hence should be identical) can then be combined to form the correct consensus sequence.

One additional, and crucial, limitation is that standard sequencing platforms fail to retain information about the pairing of heavy and light Ab chains. New technologies are emerging that address this issue using microfluidic single-cell technology. More generally, it seems reasonable to assume that technologies better suited to the specific requirements of Rep-seq will emerge in the fairly near future and give an additional boost to this emerging field.

Bioinformatics plays a crucial role in the analysis of Rep-seq data. The aim of such analyses is often to probe the dynamic properties of the repertoire (e.g. to characterize the polyclonality of a response to a specific challenge as it changes over time), and also the fine details of an immune response (e.g. to elucidate the maturation pathway of a particular Ab of interest). Rep-seq analytics is an active area of research, with new tools regularly developed. A maintained list is available at OMICtools. The key steps that are undertaken in many Rep-seq analyses comprise (1) Preliminary sequence processing, including: NGS sequence quality control; primer masking; handling barcodes (if present); and the assembly of pair-end reads; (2) Junction analysis: the assignment of V, J and D genes (typically in descending order of confidence); and the identification of CDRs; (3) Clonal family inference, i.e. inferring which sequences arose from the same V(D)J rearrangement and (4) Maturation pathway reconstruction, i.e. inferring the unmutated ancestor and intermediates sequences for an Ab of interest.

Additionally, Rep-seq data is often integrated with structural and/or functional information about individual Abs, information that has been derived using a range of techniques including X-ray crystallography, hybridoma technology, or tandem mass spectrometry. Rep-seq of B- and T-cell repertoires has a wide range of applications. It is being used to characterise immune responses to different challenges, including vaccination (Galson *et al.*, 2014), infection (e.g. HIV Zhu *et al.*, 2013), West Nile virus (Tsioris *et al.*, 2015), and herpesviruses (Klarenbeek *et al.*, 2012), and to cancer (van Heijst *et al.*, 2013). It is also being used to help us gain a better understanding of the properties of 'normal' repertoires, such as the degree of similarity/dissimilarity between the repertoires of individuals (including twin studies (Zvyagin *et al.*, 2014), and to elucidate the impact of ageing (Rubelt *et al.*, 2012)). Rep-seq can also help us gain insights into the properties of immune dysfunction (e.g. acute Systemic Lupus Erythematosus (Tipton *et al.*, 2015) and rheumatoid arthritis (Tan *et al.*, 2014).

Conclusions

Computational immunogenetics involves the development and application of bioinformatics methods, mathematical models and statistical techniques for the study of immune system biology. The immune system composition is highly heterogenic and all the entities involved in its functionality play both centralized and de-centralized roles. The metaphor that depicts the immune system like an orchestra is tight-fitting as it really shows a clear image on the machinery that governs the immune function. Tens of different cell types and thousands of interconnected molecular pathways and signals act in real time to carry on the defence against pathogens and tumours. Systems approaches can be used to predict how the immune system will respond to a particular infection or therapeutic strategies, helping in the comprehension on how to act on an immunotherapy to enhance their effect and to reduce possible side-effects. Moreover, computational approaches are increasingly vital to understand the implications of the wealth of gene expression and epigenomics data being gathered from immune cells. The availability of dozens of immune database really helps in organizing and deciphering the huge amount of data that is collected day by day from the availability of new biotechnological approaches. Multi-scale methodologies are needed to tether molecular, cellular and organism levels, in order to gain a global vision on how the things really go. There are already such methodologies but they still suffer of complex issues related to the communications among them i.e., the scales speak different languages in terms of time and space. Computational

immunology is also required in the new challenge in the field of computational biomedicine: *in silico* trials. *In silico* trial platforms allow the simulation of the relevant individual human physiology and physiopathology in patients. Virtual populations of individuals aim to study the effects of treatments, allowing the simulation of the action of the vaccination strategies and predicting the treatments outcomes in order to have a personalized medicine approach. In the near future, *in silico* trial could predict, explore and inform of the reasons for failure should the vaccinations strategies against a determined pathology under testing found not efficient, which will suggest possible improvements, which can be rapidly explored on *in silico* platforms. Another possible benefit of the *in silico* trial platform could be the reduction of human testing: by providing a reliable prediction of the phase III outcomes on the basis of the data collected a phase II clinical trial, it will increase the confidence in investing in a phase III trial to demonstrate the efficacy in term of reduced recurrence and to drastically reduce the numerosity of the enrolled patients to obtain enough statistical power.

See also: Computing for Bioinformatics. Extraction of Immune Epitope Information. Immunoglobulin Clonotype and Ontogeny Inference. Immunoinformatics Databases. Natural Language Processing Approaches in Bioinformatics. Vaccine Target Discovery

References

- Adam, J.A., 1996. Effects of vascularization on lymphocyte/tumor cell dynamics: Qualitative features. *Mathematical and Computer Modelling* 23, 1–10.
- Ahmad, T.A., Eweida, A.E., El-Sayed, L.H., 2016. T-cell epitope mapping for the design of powerful vaccines. *Vaccine Reports* 6, 13–22.
- Ahmed, R.K., Mäurer, M.J., 2009. T-cell epitope mapping. *Methods in Molecular Biology* 524, 427–438.
- Aidinis, V., Chandras, C., Manoloukos, M., *et al.*, 2008. MUGEN mouse database; Animal models of human immunological diseases. *Nucleic Acids Research* 36, D1048–D1054.
- Aleman, D., Pappalardo, F., Pennisi, M., Motta, S., Brusici, V., 2012. Combining cellular automata and lattice Boltzmann method to model multiscale avascular tumor growth coupled with nutrient diffusion and immune competition. *Journal of Immunological Methods* 376, 55–68.
- Alix, A.J., 1999. Predictive estimation of protein linear epitopes by using the program people. *Vaccine* 18, 311–314.
- Almagro, J.C., Beavers, M.P., Hernandez-Guzman, F., *et al.*, 2011. Antibody modeling assessment. *Proteins: Structure, Function, and Bioinformatics* 79, 3050–3066.
- Almagro, J.C., Teplyakov, A., Luo, J., *et al.*, 2014. Second antibody modeling assessment (AMA-II). *Proteins: Structure, Function, and Bioinformatics* 82, 1553–1562.
- Ansari, H.R., Flower, D.R., Raghava, G.P., 2010. AntigenDB: An immunoinformatics database of pathogen antigens. *Nucleic Acids Research* 38, D847–D853.
- Ansari, H.R., Raghava, G.P., 2010. Identification of conformational B-cell Epitopes in an antigen from its primary sequence. *Immunome Research* 6, 6.
- Atanasova, M., Patronov, A., Dimitrov, I., Flower, D.R., Doytchinova, I., 2013. EpiDOCK: A molecular docking-based tool for MHC class II binding prediction. *Protein Engineering, Design and Selection* 26, 631–634.
- Bandini, S., 1996. Hyper-cellular automata for the simulation of complex biological systems: A model for the immune system, special issue on advances in mathematical modeling of biological processes. *International Journal of Applied Science and Computation* 3, 1076–1131.
- Banks, H.T., Hu, S., Rosenberg, E., 2017. A dynamical modeling approach for analysis of longitudinal clinical trials in the presence of missing endpoints. *Applied Mathematics Letters* 63, 109–117.
- Beauchemin, C., Samuel, J., Tuszyński, J., 2005. A simple cellular automaton model for influenza A viral infections. *Journal of Theoretical Biology* 232, 223–234.
- Bernaschi, M., Castiglione, F., 2001. Design and implementation of an immune system simulator. *Computation in Biology and Medicine* 31, 303–331.
- Bhasin, M., Raghava, G.P.S., 2004a. Analysis and prediction of affinity of TAP binding peptides using cascade SVM. *Protein Science* 13, 596–607.
- Bhasin, M., Raghava, G.P.S., 2004b. SVM based method for predicting HLADRB1*0401 binding peptides in an antigen sequence. *Bioinformatics* 20, 421–423.
- Bhasin, M., Raghava, G.P.S., 2005. Pcleavage: An SVM based method for prediction of constitutive proteasome and immunoproteasome cleavage sites in antigenic sequences. *Nucleic Acids Research* 33, W202–W207.
- Bhasin, M., Raghava, G.P., 2007. A hybrid approach for predicting promiscuous MHC class I restricted T cell epitopes. *Journal of Biosciences* 32, 31–42.
- Bhasin, M., Singh, H., Raghava, G.P.S., 2003. MHCBN: A comprehensive database of MHC binding and non-binding peptides. *Bioinformatics* 19, 665–666.
- Bianca, C., Chiacchio, F., Pappalardo, F., Pennisi, M., 2012. Mathematical modeling of the immune system recognition to mammary carcinoma antigen. *BMC Bioinformatics* 13, S21.
- Biggs, N., Lloyd, E.K., Wilson, R.J., 1986. *Graph Theory*. Oxford: Clarendon.
- Blum, J.S., Wearsch, P.A., Cresswell, P., 2013. Pathways of antigen processing. *Annual Review of Immunology* 31, 443–473.
- Blythe, M.J., Flower, D.R., 2005. Benchmarking B cell epitope prediction: Underperformance of existing methods. *Protein Science* 14, 246–248.
- Bouvier, M., Wiley, D.C., 1994. Importance of peptide amino and carboxyl termini to the stability of MHC class I molecules. *Science* 265, 398–402.
- Brenke, R., Hall, D.R., Chuang, G.Y., *et al.*, 2012. Application of asymmetric statistical potentials to antibody–protein docking. *Bioinformatics* 28, 2608–2614.
- Brusici, V., Rudy, G., Honeyman, G., Hammer, J., Harrison, L., 1998. Prediction of MHC class II-binding peptides using an evolutionary algorithm and artificial neural network. *Bioinformatics* 14, 121–130.
- Brusici, V., van Ender, P., Zelezniuk, J., *et al.*, 1999. A neural network model approach to the study of human TAP transporter. *In Silico Biology* 1, 109–121.
- Bui, H.H., Sidney, J., Peters, B., *et al.*, 2005. Automated generation and evaluation of specific MHC binding predictive tools: Arb matrix applications. *Immunogenetics* 57, 304–314.
- Buus, S., Lauemoller, S.L., Worning, P., *et al.*, 2003. Sensitive quantitative predictions of peptide-MHC binding by a 'Query by Committee' artificial neural network approach. *Tissue Antigens* 62, 378–384.
- Chailyan, A., Tramontano, A., Marcatili, P., 2012. A database of immunoglobulins with integrated tools: Digit. *Nucleic Acids Research* 40, D1230–D1234.
- Chen, W., Guo, W.W., Huang, Y., Ma, Z., 2012. PepMapper: A collaborative web tool for mapping epitopes from affinity-selected peptides. *PLoS One* 7, e37869.
- Chen, W.H., Sun, P.P., Lu, Y., *et al.*, 2011. MimoPro: A more efficient Web-based tool for epitope prediction using phage display libraries. *BMC Bioinformatics* 12, 199.
- D'Amaro, J., Houbiers, J.G., Drijfhout, J.W., *et al.*, 1995. A computer program for predicting possible cytotoxic T lymphocyte epitopes based on HLA class I peptide-binding motifs. *Human Immunology* 43, 13–18.
- Daniel, S., Brusici, V., Caillat-Zucman, S., *et al.*, 1998. Relationship between peptide selectivities of human transporters associated with antigen processing and HLA class I molecules. *The Journal of Immunology* 161, 617–624.
- DeLisi, C., Rescigno, A., 1977. Immune surveillance and neoplasia-I: A minimal mathematical model. *Bulletin of Mathematical Biology* 39, 201–221.
- Desai, D.V., Kulkarni-Kale, U., 2014. T-cell epitope prediction methods: An overview. *Methods In Molecular Biology* 1184, 333–364.
- Diez-Rivero, C.M., Chenlo, B., Zuluaga, P., Reche, P.A., 2010a. Quantitative modeling of peptide binding to TAP using support vector machine. *Proteins* 78, 63–72.

- Diez-Rivero, C.M., Lafuente, E.M., Reche, P.A., 2010b. Computational analysis and modeling of cleavage by the immunoproteasome and the constitutive proteasome. *BMC Bioinformatics* 11, 479.
- Diez-Rivero, C.M., Reche, P.A., 2012. CD8 T cell epitope distribution in viruses reveals patterns of protein biosynthesis. *PLOS One* 7, e43674.
- Dimitrov, I., Garnev, P., Flower, D.R., Doytchinova, I., 2010. EpiTOP— a proteochemometric tool for MHC class II binding prediction. *Bioinformatics* 26, 2066–2068.
- Donnes, P., Elofsson, A., 2002. Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinformatics* 3, 25.
- Donnes, P., Kohlbacher, O., 2005. Integrated modelling of the major events in the MHC class I antigen processing pathway. *Protein Science* 14, 2132–2140.
- Dower, S.K., Qvarnstrom, E.E., 2003. Signalling networks, inflammation and innate immunity. *Biochemical Society Transactions* 31, 1462–1471.
- Doytchinova, I.A., Flower, D.R., 2005. *In silico* identification of supertypes for class II MHCs. *The Journal of Immunology* 174, 7085–7095.
- Doytchinova, I.A., Guan, P., Flower, D.R., 2006. EpiJen: A server for multistep T cell epitope prediction. *BMC Bioinformatics* 7, 131.
- Dunbar, J., Krawczyk, K., Leem, J., *et al.*, 2014. SABDab: The structural antibody database. *Nucleic Acids Research* 42, D1140–D1146.
- Durbin, R., Eddy, S., Krogh, A., Mitchison, G., 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge University Press.
- El-Manzalawy, Y., Dobbs, D., Honavar, V., 2008. Predicting linear B-cell epitopes using string kernels. *Journal of Molecular Recognition* 21, 243–255.
- Emini, E.A., Hughes, J.V., Perlow, D.S., Boger, J., 1985. Induction of hepatitis A virus-neutralizing antibody by a virus-specific synthetic peptide. *Journal of Virology* 55, 836–839.
- Eroshkin, A.M., LeBlanc, A., Weekes, D., *et al.*, 2014. bNAber: Database of broadly neutralizing HIV antibodies. *Nucleic Acids Research* 42, D1133–D1139.
- Fiser, A., Do, R.K.G., Sali, A., 2000. Modeling of loops in protein structures. *Protein Science* 9, 1753–1773.
- Flower, D.R., Macdonald, I.K., Ramakrishnan, K., Davies, M.N., Doytchinova, I.A., 2010. Computer aided selection of candidate vaccine antigens. *Immunome Research* 6, S1.
- Galson, J.D., Pollard, A.J., Trück, J., Kelly, D.F., 2014. Studying the antibody repertoire after vaccination: Practical applications. *Trends in immunology* 35, 319–331.
- Gillespie, D.T., 2007. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry* 58, 35–55.
- Giudicelli, V., Ginestoux, C., Folch, G., *et al.*, 2006. IMGT/LIGM-DB, the IMGT[®] comprehensive database of immunoglobulin and T cell receptor nucleotide sequences. *Nucleic Acids Research* 34, D781–D784.
- Giudicelli, V., Lefranc, M.P., 1999. Ontology for immunogenetics: The IMGT-Ontology. *Bioinformatics* 15, 1047–1054.
- Gorenshteyn, D., Zaslavsky, E., Fribourg, M., *et al.*, 2015. Interactive big data resource to elucidate human immune pathways and diseases. *Immunity* 43, 605–614.
- Gray, J.J., Moughon, S.E., Kortemme, T., *et al.*, 2003. Protein–protein docking predictions for the CAPRI experiment. *Proteins: Structure, Function, and Bioinformatics* 52, 118–122.
- Greenbaum, J.A., Andersen, P.H., Blythe, M., *et al.*, 2007. Towards a consensus on datasets and evaluation metrics for developing B-cell epitope prediction tools. *Journal of Molecular Recognition* 20, 75–82.
- Greenbaum, J., Sidney, J., Chung, J., *et al.*, 2011. Functional classification of class II human leukocyte antigen (HLA) molecules reveals seven different supertypes and a surprising degree of repertoire sharing across supertypes. *Immunogenetics* 63, 325–335.
- Gribskov, M., Veretnik, S., 1996. Identification of sequence pattern with profile analysis. *Methods in Enzymology* 266, 198–212.
- Grilo, A., Caetano, A., Rosa, A., 1999. Immune system simulation through a complex adaptive system model. In: Dasgupta, D., Nino, F. (Eds.), *Proceedings of the 3rd Workshop on Genetic Algorithms and Artificial Life (GAAL99)*, pp. 1–2. Lisbon: CRC Press.
- Grimes, G.R., Moodie, S., Beattie, J.S., *et al.*, 2005. GPX-Macrophage Expression Atlas: A database for expression profiles of macrophages challenged with a variety of pro-inflammatory, anti-inflammatory, benign and pathogen insults. *BMC Genomics* 6, 178.
- Guan, P., Doytchinova, A., Zygouri, C., Flower, D.R., 2003. MHCPreD: A server for quantitative prediction of peptide-MHC binding. *Nucleic Acids Research* 31, 3621–3624.
- Gullo, F., van der Garde, M., Russo, G., *et al.*, 2015. Computational modeling of the expansion of human cord blood CD133+ hematopoietic stem/progenitor cells with different cytokine combinations. *Bioinformatics* 31, 2514–2522.
- Guo, Z., Han, H.K., Tay, J.C., 2005. Sufficiency verification of HIV-1 pathogenesis based on multi-agent simulation. In: Beyer, H., O'Reilly, U., Arnold, D., *et al.* (Eds.), *Proceedings of the ACM Genetic and Evolutionary Computation Conference 2005 (GECCO'05)*, pp. 305–312. Washington: ACM Press.
- Günther, S., Hempel, D., Dunkel, M., Rother, K., Preissner, R., 2007. SuperHapten: A comprehensive database for small immunogenic compounds. *Nucleic Acids Research* 35, D906–D910.
- Hakenberg, J., Nussbaum, A.K., Schild, H., *et al.*, 2003. MAPPP: MHC class I antigenic peptide processing prediction. *Appl Bioinformatics* 2, 155–158.
- Hammer, G.E., Gonzalez, F., Champsaur, M., Cado, D., Shastri, N., 2006. The aminopeptidase ERAAP shapes the peptide repertoire displayed by major histocompatibility complex class I molecules. *Nature Immunology* 7, 103–112.
- Harwood, N.E., Batista, F.D., 2010. Early events in B cell activation. *Annual Review of Immunology* 28, 185–210.
- Haste Andersen, P., Nielsen, M., Lund, O., 2006. Prediction of residues in discontinuous B-cell epitopes using protein 3D structures. *Protein Science* 15, 2558–2567.
- Helmberg, W., Dunivin, R., Feolo, M., 2004. The sequencing-based typing tool of dbMHC: Typing highly polymorphic gene sequences. *Nucleic Acids Research* 32, W173–W175.
- Hershberg, U., Louzoun, Y., Atlan, H., Solomon, S., 2001. HIV time hierarchy: Winning the war while, losing all the battles. *Physica A* 289, 178–190.
- He, Y., Xiang, Z., Mobley, H.L., 2010. Vaxign: The first web-based vaccine design program for reverse vaccinology and applications for vaccine development. *Journal of Biomedicine and Biotechnology* 2010, 297505.
- Holzhtutter, H.G., Frommel, C., Klotzel, P.M., 1999. A theoretical approach towards the identification of cleavage-determining amino acid motifs of the 20 S proteasome. *Journal of Molecular Biology* 286, 1251–1265.
- Hoops, S., Sahle, S., Gauges, *et al.*, 2006. COPASI: A COMplex PATHway Simulator. *Bioinformatics* 22, 3067–3074.
- Hopp, T.P., Woods, K.R., 1981. Prediction of protein antigenic determinants from amino acid sequences. *Proceedings of the National Academy of Sciences of the USA* 78, 3824–3828.
- Hopp, T.P., Woods, K.R., 1983. A computer program for predicting protein antigenic determinants. *Molecular Immunology* 20, 483–489.
- Hoze, E., Tsaban, L., Maman, Y., Louzoun, Y., 2013. Predictor for the effect of amino acid composition on CD4+ T cell epitopes preprocessing. *Journal of Immunological Methods* 391, 163–173.
- Huang, J., Gutteridge, A., Honda, W., Kanehisa, M., 2006. MIMOX: A web tool for phage display based epitope mapping. *BMC Bioinformatics* 7, 451. [*Immunogenetics* 41, 178–228].
- Hu, R., Ruan, X., 2003. A simple cellular automaton model for tumor-immunity system. In: *Proceedings of IEEE International Conference of Robotics, Intelligent Systems and Signal Processing*, pp. 1031–1035. Changsha, Hunan, China: IEEE Press.
- Jacob, C., Litorco, J., Lee, L., 2004. Immunity through swarms: Agent-based simulations of the human immune system. *Lecture Notes in Computer Science* 3239, 400–412.
- Jacob, L., Vert, J.P., 2008. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics* 24, 358–366.
- Jandrić, D.R., 2016. SVM and SVR-based MHC-binding prediction using a mathematical presentation of peptide sequences. *Computational Biology and Chemistry* 65, 117–127.
- Janes, K.A., Lauffenburger, D.A., 2013. Models of signalling networks – what cell biologists can gain from them and give to them. *Journal of Cell Science* 126, 1913–1921.
- Jensen, P.E., 2007. Recent advances in antigen processing and presentation. *Natural Immunology* 8, 1041–1048.
- Jespersen, M.C., Peters, B., Nielsen, M., Marcatili, P., 2017. BepiPred-2.0: Improving sequence-based B-cell epitope prediction using conformational epitopes. *Nucleic Acids Research* 2. doi:10.1093/nar/gkx346.
- Kangueane, P., Saktharkar, M.K., Kolatkar, P.R., Ren, E.C., 2001. Towards the MHC-Peptide combinatorics. *Human Immunology* 62, 539–556.
- Karplus, P.A., Schulz, G.E., 1985. Prediction of chain flexibility in proteins: A tool for the selection of peptide antigen. *Naturwissenschaften* 72, 212–213.

- Kawai, T., Akira, S., 2011. Toll-like receptors and their crosstalk with other innate receptors in infection and immunity. *Immunity* 34, 637–650.
- Khan, J.M., Cheruku, H.R., Tong, J.C., Ranganathan, S., 2011. MPID-T2: A database for sequence-structure-function analyses of pMHC and TR/pMHC structures. *Bioinformatics* 27, 1192–1193.
- Khan, J.M., Ranganathan, S., 2011. Understanding TR binding to pMHC complexes: How does a TR scan many pMHC complexes yet preferentially bind to one. *PLoS ONE* 6 (2), e17194.
- Khan, J.M., Tong, J.C., Ranganathan, S., 2010. Structural Immunoinformatics: Understanding MHC-Peptide-TR binding. In: Davies, N., Ranganathan, S., Flower, D.R. (Eds.), *Bioinformatics for Immunomics*. New York: Springer, pp. 77–93.
- Kidd, B.A., Peters, L.A., Schadt, E.E., Dudley, J.T., 2014. Unifying immunology with informatics and multiscale biology. *Nature Immunology* 15, 118–127.
- Kirschner, D.E., Webb, G.F., 1997. A mathematical model of combined drug therapy of HIV infection. *Journal of Theoretical Medicine* 1, 25–34.
- Klarenbeek, P.L., Remmerswaal, E.B.M., ten Berge, I.J.M., *et al.*, 2012. Deep sequencing of antiviral T-cell responses to HCMV and EBV in humans reveals a stable repertoire that is maintained for many years. *PLoS Pathogens* 8, e1002889.
- Kolaskar, A.S., Tongaonkar, P.C., 1990. A semi-empirical method for prediction of antigenic determinants on protein antigens. *FEBS Letters* 276, 172–174.
- Krawczyk, K., Liu, X., Baker, T., *et al.*, 2014. Improving B-cell epitope prediction and its application to global antibody-antigen docking. *Bioinformatics* 30, 2288–2294.
- Kulkarni-Kale, U., Bhosle, S., Kolaskar, A.S., 2005. CEP: A conformational epitope prediction server. *Nucleic Acids Research* 33, W168–W171.
- Kuznetsov, V.A., Makalkin, I.A., Taylor, M.A., Perelson, M.A., 1994. Non-linear dynamics of immunogenic tumors: Parameter estimation and global bifurcation analysis. *Bulletin of Mathematical Biology* 56, 295–321.
- Lacerda, M., Scheffler, K., Seoighe, C., 2010. Epitope discovery with phylogenetic hidden Markov models. *Molecular Biology and Evolution* 27, 1212–1220.
- Lafuente, E.M., Reche, P.A., 2009. Prediction of MHC-peptide binding: A systematic and comprehensive overview. *Current Pharmaceuticals Design* 15, 3209–3220.
- Larsen, M.V., Lundegaard, C., Lamberth, K., *et al.*, 2005. An integrative approach to CTL epitope prediction: A combined algorithm integrating MHC class I binding, TAP transport efficiency, and proteasomal cleavage predictions. *European Journal of Immunology* 35, 2295–2303.
- Leem, J., Dunbar, J., Georges, G., Shi, J., Deane, C.M., *et al.*, 2016. ABodyBuilder: Automated antibody structure prediction with data-driven accuracy estimation. *MAbs* 8, 1259–1268.
- Lees, W.D., Stejskal, L., Moss, D.S., Shepherd, A.J., 2017. Investigating substitutions in antibody-antigen complexes Using Molecular Dynamics: A case study with Broad-spectrum, influenza A antibodies. *Frontiers In Immunology* 8, 143.
- Lefranc, M.-P., Giudicelli, V., Duroux, P., *et al.*, 2015. IMGT[®], the international ImMunoGeneTics information system[®] 25 years on. *Nucleic Acids Research* 43, D413–D422.
- Lefranc, M.P., Giudicelli, V., Kaas, Q., *et al.*, 2005. IMGT, the international ImMunoGeneTics information system[®]. *Nucleic Acids Research* 33, D593–D597.
- Levitt, M., 2009. Nature of the protein universe. *Proceedings of the National Academy of Sciences of the USA* 106, 11079–11084.
- Liang, S., Zheng, D., Standley, D.M., *et al.*, 2010. EPSVR and EPMeta: Prediction of antigenic epitopes using support vector regression and multiple server results. *BMC Bioinformatics* 11, 381.
- Lins, L., Thomas, A., Brasseur, R., 2003. Analysis of accessible surface of residues in proteins. *Protein Science* 12, 1406–1417.
- Liu, W., Meng, X., Xu, Q., Flower, D.R., Li, T., 2006. Quantitative prediction of mouse class I MHC peptide binding affinity using support vector machine regression (SVR) models. *BMC Bioinformatics* 7, 182.
- Lund, O., Nielsen, M., Kesmir, C., *et al.*, 2004. Definition of supertypes for HLA molecules using clustering of specificity matrices. *Immunogenetics* 55, 797–810.
- Lynn, D.J., Winsor, G.L., Chan, C., *et al.*, 2008. InnateDB: Facilitating systems-level analyses of the mammalian innate immune response. *Molecular Systems Biology* 4, 218.
- Madden, D.R., 1995. The three-dimensional structure of peptide-MHC complexes. *Annual Review of Immunology* 13, 587–622.
- Madden, D.R., Garboczi, D.N., Wiley, D.C., 1993. The antigenic identity of peptide-MHC complexes: A comparison of the conformations of five viral peptides presented by HLA-A2. *Cell* 75, 693–708.
- Malherbe, L., 2009. T-cell epitope mapping. *Annals of Allergy Asthma and Immunology* 103, 76–79.
- Mallet, D., De Pillis, L., 2006. A cellular automata model of tumor immune system interactions. *Journal of Theoretical Biology* 239, 334–350.
- Mamitsuka, H., 1998. Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models. *Proteins* 33, 460–474.
- Marcatili, P., Olimpieri, P.P., Chailan, A., Tramontano, A., 2014. Antibody modeling using the prediction of ImmunoGlobulin structure (PIGS) web server. *Nature Protocols* 9, 2771–2783.
- Marks, C., Deane, C.M., 2017. Antibody H3 structure prediction. *Computational and Structural Biotechnology Journal* 15, 222–231.
- Mayrose, I., Penn, O., Erez, E., *et al.*, 2007. Pepitope: Epitope mapping from affinity-selected peptides. *Bioinformatics* 23, 3244–3246.
- Meireles, L.M.C., Dömling, A.S., Camacho, C.J., 2010. ANCHOR: A web server and database for analysis of protein-protein interaction binding pockets for drug discovery. *Nucleic Acids Research* 38, W407–W411.
- Milik, M., Sauer, D., Brunmark, A.P., *et al.*, 1998. Application of an artificial neural network to predict specific class I MHC binding peptide sequences. *Natural Biotechnology* 16, 753–756.
- Molero-Abraham, M., Glutting, J.P., Flower, D.R., Lafuente, E.M., Reche, P.A., 2015. EPIPOX: Immunoinformatic characterization of the shared T-Cell epitome between Variola virus and related pathogenic Orthopoxviruses. *Journal of Immunology Research* 2015, 738020.
- Molero-Abraham, M., Lafuente, E.M., Flower, D.R., Reche, P.A., 2013. Selection of conserved epitopes from hepatitis C virus for pan-population stimulation of T-cell responses. *Clinical and Developmental Immunology* 2013, 601943.
- Molero-Abraham, M., Lafuente, E.M., Reche, P., 2014. Customized predictions of peptide-MHC binding and T-cell epitopes using EPIMHC. *Methods In Molecular Biology* 1184, 319–332.
- Murata, T., 1989. Petri Nets: Properties, analysis and applications. *Proceedings of the IEEE* 77, 541–580.
- Nani, F.K., Ögüztörel, M.N., 1994. Modelling and simulation of Rosenberg-type adoptive cellular immunotherapy. *IMA Journal of Mathematics Applied in Medicine & Biology* 11, 107–147.
- Negi, S.S., Braun, W., 2009. Automated detection of conformational epitopes using phage display Peptide sequences. *Bioinformatics and Biology Insights* 3, 71–81.
- Nielsen, M., Lund, O., 2009. NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinformatics* 10, 296.
- Nielsen, M., Lundegaard, C., Blicher, T., *et al.*, 2007. NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. *PLoS One* 2 (8), e796.
- Nielsen, M., Lundegaard, C., Blicher, T., *et al.*, 2008. Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCpan. *PLoS Computational Biology* 4, e1000107.
- Nielsen, M., Lundegaard, C., Lund, O., 2007. Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. *BMC Bioinformatics* 8, 238.
- Nielsen, M., Lundegaard, C., Lund, O., Kesmir, C., 2005. The role of the proteasome in generating cytotoxic T-cell epitopes: Insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics* 57, 33–41.
- Nielsen, M., Lundegaard, C., Worning, P., *et al.*, 2003. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science* 12, 1007–1017.
- Nielsen, M., Lundegaard, C., Worning, P., *et al.*, 2004. Improved prediction of MHC class I and class II epitopes using a novel Gibbs sampling approach. *Bioinformatics* 20, 1388–1397.
- Nowak, M.A., Anderson, R.M., Mclean, A.R., *et al.*, 1991. Antigenic diversity thresholds and the development of AIDS. *Science* 254, 963–969.

- Nussbaum, A.K., Kuttler, C., Haderl, K.P., Rammensee, H.G., Schild, H., 2001. PAPROC: A prediction algorithm for proteasomal cleavages available on the WWW. *Immunogenetics* 53, 87–94.
- Odorico, M., Pellequer, J.L., 2003. BEPITOPE: Predicting the location of continuous epitopes and patterns in proteins. *Journal of Molecular Recognition* 16, 20–22.
- Oyarzun, P., Ellis, J.J., Boden, M., Kobe, B., 2013. PREDIVAC: CD4+ T-cell epitope prediction for vaccine design that covers 95% of HLA class II DR protein diversity. *BMC Bioinformatics* 14, 52.
- Palladini, A., Nicoletti, G., Pappalardo, F., Murgio, A., Grosso, V., *et al.*, 2010. In silico modeling and in vivo efficacy of cancer preventive vaccinations. *Cancer Research* 70, 7755–7763.
- Pappalardo, F., Fichera, E., Paparone, N., Lombardo, A., Pennisi, M., *et al.*, 2016. A computational model to predict the immune system activation by citrus derived vaccine adjuvants. *Bioinformatics* 32, 2672–2680.
- Pappalardo, F., Flower, D., Russo, G., Pennisi, M., Motta, S., 2015. Computational modelling approaches to vaccinology. *Pharmacological Research* 92, 40–45.
- Pappalardo, F., Forero, I.M., Pennisi, M., Palazon, A., Melero, I., *et al.*, 2011. SimB16: Modeling induced immune system response against B16-melanoma. *PLoS ONE* 6.
- Parker, K.C., Bednarek, M.A., Coligan, J.E., 1994. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side chains. *Journal of Immunology* 152, 163–175.
- Park, I., Na, D., Lee, D., Lee, K.H., 2006. Fuzzy continuous Petri Net-based approach for modeling immune systems. *Lecture Notes In Computer Science* 3931, 278–285.
- Patronov, A., Doytchinova, I., 2013. T-cell epitope vaccine design by immunoinformatics. *Open Biology* 3, 120139.
- Pellequer, J.L., Westhof, E., 1993. PREDITOP: A program for antigenicity prediction. *Journal of Molecular Graphics* 11, 204–210.
- Pellequer, J.L., Westhof, E., Van Regenmortel, M.H., 1991. Predicting location of continuous epitopes in proteins from their primary structures. *Methods In Enzymology* 203, 176–201.
- Pellequer, J.L., Westhof, E., Van Regenmortel, M.H., 1993. Correlation between the location of antigenic sites and the prediction of turns in proteins. *Immunology Letters* 36, 83–99.
- Pennisi, M., Cavalieri, S., Motta, S., Pappalardo, F., 2016. A methodological approach for using High-Level Petri Nets to model the adaptive immune system response. *BMC Bioinformatics* 16, 91–105.
- Pennisi, M., Rajput, A.-M., Toldo, L., Pappalardo, F., 2013. Agent based modeling of Treg-Teff cross regulation in relapsing-remitting multiple sclerosis. *BMC Bioinformatics* 14.
- Perelson, A.S., 2002. Modelling viral and immune system dynamics. *Nature Reviews Immunology* 2, 28–36.
- Perrin, D., Ruskin, H.J., Burns, J., Crane, M., 2006. An agent-based approach to immune modelling. *Lecture Notes in Computer Science* 3980, 612–621.
- Peters, B., Sette, A., 2005. Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. *BMC Bioinformatics* 6, 132.
- Peters, B., Tong, W., Sidney, J., Sette, A., Weng, Z., 2003. Examining the independent binding assumption for binding of peptide epitopes to MHC-I molecules. *Bioinformatics* 19, 1765–1772.
- Pires, D.E., Ascher, D.B., Blundell, T.L., 2014. mCSM: Predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* 30, 335–342.
- Ponomarenko, J.V., Bourne, P.E., 2007. Antibody-protein interactions: Benchmark datasets and prediction tools evaluation. *BMC Structural Biology* 7, 64.
- Ponomarenko, J., Bui, H.H., Li, W., *et al.*, 2008. ElliPro: A new structure-based tool for the prediction of antibody epitopes. *BMC Bioinformatics* 9, 514.
- Ponomarenko, J., Papangelopoulos, N., Zajonc, D.M., *et al.*, 2011. IEDB-3D: Structural data within the immune epitope database. *Nucleic Acids Research* 39, D1164–D1170.
- Potocnakova, L., Bhide, M., Pulzova, L.B., 2016. An introduction to B-Cell epitope mapping and in silico epitope prediction. *Journal of Immunology Research* 2016, 6760830.
- Puck, J.M., 1996. IL2RGbase: A database of gamma c-chain defects causing human X-SCID. *Immunology Today* 17, 507–511.
- Puck, J.M., 2005. ALPSbase: Database of mutation causing human ALPS. Available online at: <http://research.nhgri.nih.gov/alps/>.
- Rammensee, H., Bachmann, J., Emmerich, N.P., Bachor, O.A., Stevanovic, S., 1999. SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* 50, 213–219.
- Rammensee, H.G., Friede, T., Stevanovic, S., 1995. MHC ligands and peptide motifs: First listing. *Immunogenetics* 41, 178–228.
- Reche, P.A., Glutting, J.P., Reinherz, E.L., 2002. Prediction of MHC class I binding peptides using profile motifs. *Human Immunology* 63, 701–709.
- Reche, P.A., Glutting, J.P., Zhang, H., Reinherz, E.L., 2004. Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. *Immunogenetics* 56, 405–419.
- Reche, P.A., Keskin, D.B., Hussey, R.E., *et al.*, 2006. Elicitation from virus-naïve individuals of cytotoxic T lymphocytes directed against conserved HIV-1 epitopes. *Medical Immunology* 5, 1.
- Reche, P.A., Reinherz, E.L., 2003. Sequence variability analysis of human class I and class II MHC molecules: Functional and structural correlates of amino acid polymorphisms. *Journal of Molecular Biology* 331, 623–641.
- Reche, P.A., Reinherz, E.L., 2005. PEPVAC: A web server for multi-epitope vaccine development based on the prediction of supertypic MHC ligands. *Nucleic Acids Research* 33, W138–W142.
- Reche, P., Reinherz, E.L., 2007. Definition of MHC supertypes through clustering of MHC peptide-binding repertoires. *Methods In Molecular Biology* 409, 163–173.
- Reche, P.A., Zhang, H., Glutting, J.-P., Reinherz, E.L., 2005. EPIMHC: A curated database of MHC-binding peptides for customized computational vaccinology. *Bioinformatics* 21, 2140–2141.
- Retter, I., Althaus, H.H., Münch, R., Müller, W., 2005. VBASE2, an integrative V gene database. *Nucleic Acids Research* 33, D671–D674.
- Rice, P., Longden, I., Bleasby, A., 2000. EMBOSS: The European molecular biology open software suite. *Trends in Genetics* 16, 276–277.
- Robinson, J., Malik, A., Parham, P., Bodmer, J.G., Marsh, S.G.E., 2000. IMGT/HLA database – a sequence database for the human major histocompatibility complex. *Tissue Antigens* 55, 280–287.
- Robinson, J., Waller, M.J., Parham, P., *et al.*, 2003. IMGT/HLA and IMGT/MHC: Sequence databases for the study of the major histocompatibility complex. *Nucleic Acids Research* 31, 311–314.
- Rubelt, F., Sievert, V., Knaust, F., *et al.*, 2012. Onset of immune senescence defined by unbiased pyrosequencing of human immunoglobulin mRNA repertoires. *PLoS One* 7, e49774.
- Rubinstein, N.D., Mayrose, I., Martz, E., Pupko, T., 2009. Epitopia: A web-server for predicting B-cell epitopes. *BMC Bioinformatics* 10, 287.
- Ruppert, J., Sidney, J., Celis, E., *et al.*, 1993. Prominent role of secondary anchor residues in peptide binding to HLA-A2.1 molecules. *Cell* 74, 929–937.
- Saha, S., Bhasin, M., Raghava, G.P., 2005. Bcipep: A database of B-cell epitopes. *BMC Genomics* 6, 79.
- Saha, S., Raghava, G.P., 2006. Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. *Proteins* 65, 40–48.
- Samrajiwa, S.A., Forster, S., Auchetti, K., Hertzog, P.J., 2009. INTERFEROME: The database of interferon regulated genes. *Nucleic Acids Research* 37, D852–D857.
- Santos, R., 1999. Immune responses: Getting close to experimental results with cellular automata models. Singapore: World Scientific Publishing Company.
- Savoie, C.J., Kamikawaji, N., Sasazuki, T., Kuhara, S., 1999. Use of BONSAI decision trees for the identification of potential MHC class I peptide epitope motifs. In: *Pacific Symposium on Biocomputing*, pp. 182–9.
- Schlessinger, A., Ofra, Y., Yachdav, G., Rost, B., 2006. Epitome: Database of structure-inferred antigenic epitopes. *Nucleic Acids Research* 34, D777–D780.
- Schubert, B., Brachvogel, H.P., Jurgens, C., Kohlbacher, O., 2015. EpiToolKit – a web-based workbench for vaccine design. *Bioinformatics* 31, 2211–2213.
- Schubert, B., Walzer, M., Brachvogel, H.P., *et al.*, 2016. FRED 2: An immunoinformatics framework for Python. *Bioinformatics* 32, 2044–2046.
- Sela-Culang, I., Ashkenazi, S., Peters, B., Ofra, Y., 2015a. PEASE: Predicting B-cell epitopes utilizing antibody sequence. *Bioinformatics* 31, 1313–1315.
- Sela-Culang, I., Ofra, Y., Peters, B., 2015b. Antibody specific epitope prediction-emergence of a new paradigm. *Current Opinion in Virology* 11, 98–102.
- Sette, A., Sidney, J., 1998. HLA supertypes and supermotifs: A functional perspective on HLA polymorphism. *Current Opinion in Immunology* 10, 478–482.
- Sette, A., Sidney, J., 1999. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. *Immunogenetics* 50, 201–212.
- Sheikh, Q.M., Gatherer, D., Reche, P.A., Flower, D.R., 2016. Towards the knowledge-based design of universal influenza epitope ensemble vaccines. *Bioinformatics* 32, 3233–3239.

- Singh, H., Ansari, H.R., Raghava, G.P., 2013. Improved method for linear B-cell epitope prediction using antigen's primary sequence. *PLoS One* 8, e62216.
- Singh, S.P., Mishra, B.N., 2016. Major histocompatibility complex linked databases and prediction tools for designing vaccines. *Human Immunology* 77, 295–306.
- Singh, H., Raghava, G.P., 2001. ProPred: Prediction of HLA-DR binding sites. *Bioinformatics* 17 (1236–7), 2001.
- Singh, H., Raghava, G.P., 2003. ProPred1: Prediction of promiscuous MHC Class-I binding sites. *Bioinformatics* 19, 1009–1014.
- Singh, M.K., Srivastava, S., Raghava, G.P., Varshney, G.C., 2006. HaptenDB: A comprehensive database of haptens, carrier proteins and anti-hapten antibodies. *Bioinformatics* 22, 253–255.
- Sircar, A., Gray, J.J., 2010. SnugDock: Paratope structural optimization during antibody-antigen docking compensates for errors in antibody homology models. *PLoS Computational Biology* 6, e1000644.
- Soga, S., Kuroda, D., Shirai, H., *et al.*, 2010. Use of amino acid composition to predict epitope residues of individual antibodies. *Protein Engineering, Design and Selection* 23, 441–448.
- Stern, L.J., Wiley, D.C., 1994. Antigenic peptide binding by class I and class II histocompatibility proteins. *Structure* 2, 245–251.
- Sturniolo, T., Bono, E., Ding, J., *et al.*, 1999. Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. *Nature Biotechnology* 17, 555–561.
- Sun, J., Wu, D., Xu, T., *et al.*, 2009. SEPPA: A computational server for spatial epitope prediction of protein antigens. *Nucleic Acids Research*, 37. pp. W612–W616.
- Sweredowski, M.J., Baldi, P., 2008. PEPITO: Improved discontinuous B-cell epitope prediction using multiple distance thresholds and half sphere exposure. *Bioinformatics* 24, 1459–1460.
- Tan, Y.C., Kongpachith, S., Blum, L.K., *et al.*, 2014. Barcode-enabled sequencing of plasmablast antibody repertoires in rheumatoid arthritis. *Arthritis & rheumatology* 66, 2706–2715.
- Tenzer, S., Peters, B., Bulik, S., *et al.*, 2005. Modeling the MHC class I pathway by combining predictions of proteasomal cleavage, TAP transport and MHC class I binding. *Cellular and Molecular Life Sciences* 62, 1025–1037.
- Terasaki, P.I., 2007. A brief history of HLA. *Immunologic Research* 38, 139–148.
- Tipton, C.M., Fucile, C.F., Darce, J., *et al.*, 2015. Diversity, cellular origin and autoreactivity of antibody-secreting cell expansions in acute Systemic Lupus Erythematosus. *Nature immunology* 16, 755.
- Tong, J.C., Ranganathan, S., 2013. Computer-aided vaccine design. *Woodhead Publishing Series in Biomedicine*, 23. Cambridge: Woodhead, pp. 1–164.
- Toseland, C.P., Clayton, D.J., McSparron, H., *et al.*, 2005. AntiJen: A quantitative immunology database integrating functional, thermodynamic, kinetic, biophysical, and cellular data. *Immunome Research* 1, 4.
- Tsioris, K., Gupta, N.T., Ogunniyi, A.O., *et al.*, 2015. Neutralizing antibodies against West Nile virus identified directly from human B cells by single-cell analysis and next generation sequencing. *Integrative Biology* 7, 1587–1597.
- Turnera, M.D., Nedjaib, B., Hursta, T., Penningtonc, D.J., 2014. Cytokines and chemokines: At the crossroads of cell signalling and inflammatory disease. *Molecular Cell* 11, 2563–2582.
- van Heijst, J.W., Ceberio, I., Lipuma, L.B., *et al.*, 2013. Quantitative assessment of T-cell repertoire recovery after hematopoietic stem cell transplantation. *Nature medicine* 19, 372.
- Van Regenmortel, M.H., 2009. What is a B-cell epitope? *Methods in Molecular Biology* 524, 3–20.
- Vita, R., Overton, J.A., Greenbaum, J.A., *et al.*, 2015. The immune epitope database (IEDB) 3.0. *Nucleic Acids Research* 43, D405–D412.
- Wang, D., Yang, L., Zhang, P., *et al.*, 2017. AAgAtlas 1.0: A human autoantigen database. *Nucleic Acids Research* 45, D769–D776.
- Wang, P., Sidney, J., Dow, C., *et al.*, 2008. A systematic assessment of MHC class II peptide binding predictions and evaluation of a consensus approach. *PLoS Computational Biology* 4, e1000048.
- Wang, X., Zhao, H., Xu, Q., *et al.*, 2006. HPTaa database-potential target genes for clinical diagnosis and immunotherapy of human carcinoma. *Nucleic Acids Research* 34, D607–D612.
- Warrender, C., Forrest, S., Koster, F., 2006. Modeling intercellular interactions in early *Mycobacterium* infection. *Bulletin of Mathematical Biology* 68, 2233–2261.
- Webb, B., Sali, A., 2014. Protein structure modeling with MODELLER. *Protein Structure Prediction* 1137, 1–15.
- Weitzner, B.D., Jeliakov, J.R., Lyskov, S., *et al.*, 2017. Modeling and docking of antibody structures with Rosetta. *Nature Protocols* 12, 401–416.
- Womble, D.D., 2000. GCG: The Wisconsin Package of sequence analysis programs. *Methods In Molecular Biology* 132, 3–22.
- Xu, X., Sun, J., Liu, Q., *et al.*, 2010. Evaluation of spatial epitope computational tools based on experimentally-confirmed dataset for protein antigens. *Chinese Science Bulletin* 55, 5.
- Yang, B., Sayers, S., Xiang, Z., He, Y., 2011. Protegen: A web-based protective antigen database and analysis system. *Nucleic Acids Research* 39, D1073–D1078.
- Yao, B., Zhang, L., Liang, S., Zhang, C., 2012. SVMTriP: A method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity. *PLoS One* 7, e45152.
- Yu, K., Petrovsky, N., Schonbach, C., Koh, J.Y., Brusic, V., 2002. Methods for prediction of peptide binding to MHC molecules: A comparative study. *Molecular Medicine* 8, 137–148.
- Zhang, C., Bickis, M.G., Wu, F.X., Kusalik, A.J., 2006. Optimally-connected hidden markov models for predicting MHC-binding peptides. *Journal of Bioinformatics and Computational Biology* 4, 959–980.
- Zhang, G.L., DeLuca, D.S., Keskin, D.B., *et al.*, 2011. MULTIPRED2: Acomputational system for large-scale identification of peptides predicted to bind to HLA supertypes and alleles. *Journal of Immunological Methods* 374, 53–61.
- Zhang, Q., Wang, P., Kim, Y., *et al.*, 2008. Immune epitope database analysis resource (IEDB-AR). *Nucleic Acids Research* 36, W513–W518.
- Zhong, W., Reche, P.A., Lai, C.C., Reinhold, B., Reinherz, E.L., 2003. Genome-wide characterization of a viral cytotoxic T lymphocyte epitope repertoire. *The Journal of Biological Chemistry* 278, 45135–45144.
- Zhu, J., Wu, X., Zhang, B., *et al.*, 2013. De novo identification of VRC01 class HIV-1-neutralizing antibodies by next-generation sequencing of B-cell transcripts. *Proceedings of the National Academy of Sciences* 110, E4088–E4097.
- Zhu, S., Udaka, K., Sidney, J., *et al.*, 2006. Improving MHC binding peptide prediction by Incorporating binding data of auxiliary MHC molecules. *Bioinformatics* 22, 1648–1655.
- Zvyagin, I.V., Pogorelyy, M.V., Ivanova, M.E., *et al.*, 2014. Distinctive properties of identical twins' TCR repertoires revealed by high-throughput sequencing. *Proceedings of the National Academy of Sciences of the United States of America* 111, 5980–5985.

Relevant Websites

<https://www.rcsb.org/pdb/>

RCSB, Protein Data Bank.

<https://www.niaid.nih.gov/research/simmune-project>

National Institute of Allergy and Infectious Diseases.

Biographical Sketch



Marta has a bachelor's Degree in Biology (2015) and a master Degree in Biomedical Investigation (2016) both from Universidad Complutense of Madrid. She has two years of experience in the Department of Clinical Microbiology and Infectious Diseases of the Hospital General Universitario Gregorio Marañón of Madrid. After completing her master degree Marta joined the Immumedicine Group at the School of Medicine of the Complutense University where she is pursuing her PhD, which is based on computer-assisted design of epitope-based vaccines.



Giulia Russo received the MSc degree in Pharmacy from the University of Catania, in 2014. In 2015, she started her research activities at the Department of Drug Sciences, University of Catania, in the field of computational modeling in oncology and immunology in Prof. Francesco Pappalardo's research group. Her current research interests include signalling pathways analysis, systems biology and computational biology in oncology. In 2015, she won the competition for the PhD program in biomedical and biotechnological sciences, with a particular interest in computational approaches in systems biomedicine.



Jose Luis has a Bachelor's Degree in Biotechnology (2015), from the Pablo de Olavide University, followed by a Master's Degree in Advanced Genetics, from the Autonomous University of Barcelona (2016). Then, he worked as research assistant in Severo Ochoa's Center for Molecular Biology to work on Huntington's disease. Jose Luis joined the Immunomedicine group early in 2017 to complete his PhD in T-cell immunology.



Marzio Pennisi is currently a post-doc researcher at the University of Catania, Italy. He earned his MSc. and PhD. degrees from University of Catania, in 2006 and 2010 respectively. His research topics include stochastic optimization techniques, evolutionary algorithms, constrained and unconstrained optimization and computational and mathematical methods for the simulation of the immune system. Up to now, he published more than 60 reviewed research papers. He serves the scientific community as editorial board member, reviewer and program committee member for various journals and conferences in the area of computer science, computational biology and bioinformatics.



Pedro A Reche is Chemist (1990) with a Ph.D. in Molecular Biology/Biochemistry (1995) from the University of Granada, Spain. He obtained post-doctoral training at the Department of Biochemistry of the University of Cambridge (England) (1995-1998) and the DNAX Research Institute, California, USA (1998-2001). From 2001 to 2006, he worked at the Dana-Farber Cancer Institute, Boston, MA, USA, directing the Bioinformatics cores of the Molecular Immunology Foundation and the Cancer Vaccine Center. He also held the appointment of Instructor of Harvard Medical School. Late 2006, he joined the University Complutense to establish the Immunomedicine Group. Pedro is a multidisciplinary scientist with relevant contributions on the fields of Biochemistry, Molecular Biology, Bioinformatics and Immunology. He has published over 60 articles in peer review journals that have received over 4700 citations (h-index = 28) and he is in the Editorial Board of 5 international journals. Pedro's main research lines are A) prediction of immunogenicity, B) discovery of therapeutic molecules and C) study of adaptive immune conditioning by epithelial cells.



Dr Adrian Shepherd was awarded a PhD in Neural Computing at UCL in 1995, then joined the research group of Prof. Janet Thornton as a Post-Doctoral Research Fellow to design machine learning classifiers for protein structure prediction. He joined Birkbeck, University of London as a lecturer in 2002, and was a founding member of the European ImmunoGrid Consortium. Now a Reader in Computational Biology at the Institute of Structural and Molecular Biology, Birkbeck, his research focuses on human adaptive immune response using a range of computational methods - from Next Generation Sequencing of immune repertoires (Rep-seq) to molecular dynamics simulations of antibody binding. His group collaborates with clinicians, virologists and other wet-lab research scientists to address medical problems that include the design of more effective vaccines and the stratification of patient responses to therapy. He is currently working on the adaptive immune response to several pathogens (influenza A virus, hepatitis C virus and Bacillus anthracis), to cancer (hepatocellular carcinoma and breast cancer), and to therapeutic proteins (notably replacement Factor VIII used in the treatment of hemophilia A).



Francesco Pappalardo is Associate Professor of Computer Science at the University of Catania, Italy and Visiting Professor at Metropolitan College of Boston, USA. He did research in computer operating system security. He was visiting researcher at Dana-Farber Cancer Institute in Boston (USA) and at the Molecular Immunogenetics Labs, IMGT in Montpellier (France). His major research area is on computational modelling in systems biology and medicine, with a special focus on the immune system. Up to now, Francesco Pappalardo published more than 100 reviewed research papers. He serves the scientific community as editorial board member, reviewer and program committee member for major journals and conferences in the area of computer science and computational biology.

Immunoinformatics Databases

Christine LP Eng and Tin Wee Tan, National University of Singapore, Singapore

Joo Chuan Tong, National University of Singapore, Singapore and Institute of High Performance Computing, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

Immunological databases play important roles in knowledge dissemination and contribute directly to our better understanding of the mechanisms underlying the defence of human body. Without appropriate data, effective bioinformatics tools cannot be built that allow for meaningful interpretations of immunological outcomes. As of May 2017, 36 immunological databases have been described in the NAR Molecular Biology Database Collection (Galperin *et al.*, 2017). Immune sequence databases provide access to experimentally characterized immune cell sequences that are implicated in infections and autoimmunity. Immune epitope databases are a set of specialist databases that stores binding data of immune molecules. Such information is commonly used to inform the design of subunit vaccines. Here, the most important databases are reviewed.

The Immune System

The human body has developed a complex immune system against the constant threat by pathogens from the environment. The first line of defence is helmed by the innate immune system, which is a general, non-specific immunity against infection that occurs rapidly upon infection (Kimbrell and Beutler, 2001). The cells from the innate immune system recognize and respond to pathogens generically, as they have different types of receptors to recognize common features of pathogens. Hence, the effects of immunity to the host are only short-lived.

In contrast, the adaptive immune system is much more specialized, capable of a more effective elimination of the infection in the event that the pathogens escape innate immunity. The adaptive immune system, consisting of B-cells and T-cells, is shaped to recognize specific infections, resulting in long-lasting immunity (Murphy *et al.*, 2008). The lymphocytes of the adaptive immune system have a single receptor type recognizing specific chemical structures, which leads to a great diversity from the huge repertoire of immunoglobulins and T-cell receptors.

What is an Epitope?

An epitope is a localized region on the surface of an antigen that is recognized by the immune system, specifically the B- or T-cells. B-cell epitopes are antigenic determinants that interact with B-cell receptors. There are two types of B-cell epitopes – linear and non-linear. About 10% of B-cell epitopes are continuous, consisting of a linear stretch of amino acids along the polypeptide chain. Most B-cell epitopes, though, are discontinuous in nature, where distant residues are brought into spatial proximity by protein folding. Studies have shown that not all residues within a B-cell epitope are functionally important for binding, and the specificity could be reduced or eliminated by single-site amino acid substitution.

T-cell epitopes are short linear peptides that bind to T-cell receptors in association with the major histocompatibility complex (MHC) molecules. Two classes of T cells are available: 1) CD8 + T cytotoxic (Tc) cells, which recognize peptides displayed by MHC class I molecules, and 2) CD4 + T helper (Th) cells, which recognize peptides in association with MHC class II molecules (#49). Tc cells release cytotoxins which are responsible for cell lysis, and granzymes which induces apoptosis. Th1 cells produce interferon γ (IFN- γ) and tumour necrosis factor β (TNF- β) and are involved in delayed-type hypersensitivity (DTH) reactions. By contrast, Th2 cells produce interleukin 4 (IL-4), IL-5, IL-10 and IL-13, which are responsible for strong antibody responses, including the activation and recruitment of IgE antibody-producing B-cells, mast cells, eosinophils, and the inhibition of several macrophage functions. T-cell epitopes presented by MHC class I molecules are typically peptides about 8–10 amino acids long, whereas MHC class II molecules binds longer peptides of about 13–20 residues.

Major Databases

In the following section, we review the major databases that are available for immunoinformatics research. This ranges from comprehensive databases such as IEDB providing epitope data and structure information, to databases specialising in genes or MHC binding information, as well as databases focusing on sequence/structure/function with emphasis on interaction information.

IEDB

The Immune Epitope Database and Analysis Resource (IEDB; see Relevant Websites section) is a central resource for immunoinformatics, with an extensive collection of B and T cell epitopes characterized experimentally in humans, non-human primates, mice, as well as other species (Fleri *et al.*, 2017). Funded by the National Institute of Allergy and Infectious Diseases (NIAID), the database was developed in 2004 at the La Jolla Institute for Allergy and Immunology with the aim of assisting researchers in the development of novel diagnostics, therapeutics and vaccines (Fig. 1).

The database contains information on peptidic and non-peptidic epitopes, as well as T cell, B cell and MHC ligand assays, which were curated from peer-reviewed literature and researcher submitted data. Each record consists of the sequence, source organism, source antigen, and references linked to additional information which include author, title and abstract of the literature. As of June 2017, IEDB includes over 298,000 peptidic epitopes and 2500 non-peptidic epitopes from over 3600 source organisms, based on approximately 319,000 T cell assays, 393,000 B cell assays, and 601,000 MHC ligand assays. This represents over 99% of all peptidic epitope information available publicly.

IMGT

The international ImMunoGeneTics information system[®] (IMGT[®]; see Relevant Websites section) is an integrated knowledge resource for immunoglobulins, major histocompatibility, and T-cell receptors of human and other vertebrates (Lefranc *et al.*, 2015). Created at 1989 by Marie-Paule Lefranc, IMGT[®] manages the diverse and complex genes and proteins of immunoglobulin and T-cell receptors, as well as the major histocompatibility proteins polymorphism. It comprises seven databases in total: (Fig. 2).

1. IMGT/LIGM-DB is a comprehensive database of fully-annotated nucleotide sequences of immunoglobulin and T-cell receptor from humans and other vertebrates (Giudicelli *et al.*, 2006). Each sequence in the database include information on sequence identification, classification of gene and allele, constitutive and specific motif description, numbering of the codon and amino acid, and the sequence obtaining information. As of June 2017, there are over 178,000 immunoglobulin and T-cell receptor sequences from 351 species.

The screenshot displays the IEDB homepage with a navigation bar at the top containing 'Home', 'Specialized Searches', 'Analysis Resource', 'Help', and 'More IEDB'. The main content area is divided into several sections:

- Welcome:** A text box explaining that IEDB is a free resource funded by a contract from the National Institute of Allergy and Infectious Diseases, offering easy searching of experimental data characterizing antibody and T cell epitopes.
- 2017 USER WORKSHOP:** A box announcing a workshop from October 25-26, 2017, at NIAID, Rockville, MD, USA, with information available at workshop.iedb.org.
- Summary Metrics:** A table showing the following data:

Peptidic Epitopes	298,573
Non-Peptidic Epitopes	2,505
T Cell Assays	319,619
B Cell Assays	393,998
MHC Ligand Assays	601,555
Epitope Source Organisms	3,605
Restricting MHC Alleles	742
References	18,528
- START YOUR SEARCH HERE:** A central section with multiple search filters:
 - Epitope:** Radio buttons for 'Any Epitopes' (selected), 'Linear Epitope', 'Discontinuous Epitopes', and 'Non-peptidic Epitopes'. A text box for 'Exact ID' with the example 'SIINFEKL'.
 - Assay:** Checkboxes for 'Positive Assays Only', 'T Cell Assays', 'B Cell Assays', and 'MHC Ligand Assays'. A text box for 'Ex: neutralization' and a 'Find' button.
 - Antigen:** A text box for 'Organism' (example: 'influenza, peanut') and a text box for 'Antigen Name' (example: 'core, capsid, myosin').
 - MHC Restriction:** Radio buttons for 'Any MHC Restriction' (selected), 'MHC Class I', 'MHC Class II', and 'MHC Nonclassical'. A text box for 'Ex: HLA-A*02:01' and a 'Find' button.
 - Host:** Radio buttons for 'Any Host' (selected), 'Humans', 'Mice', and 'Non-human Primates'. A text box for 'Ex: dog, camel' and a 'Find' button.
 - Disease:** Radio buttons for 'Any Disease' (selected), 'Infectious Disease', 'Allergic Disease', and 'Autoimmune Disease'. A text box for 'Ex: asthma, diabetes' and a 'Find' button.
- Epitope Analysis Resource:** A section on the right with links for 'T Cell Epitope Prediction' (including MHC I Binding, MHC II Binding, MHC I Processing, and MHC I Immunogenicity), 'B Cell Epitope Prediction' (including Antigen Sequence Properties, Predict discontinuous B cell epitopes, and Discotope), and 'Epitope Analysis Tools' (including Population Coverage, Conservation Across Antigens, and Clusters with Similar Sequences).

At the bottom, there are links for 'Provide Feedback', 'Help Request', 'Solutions Center', and 'Tool Licensing Information'. A footer note states: 'Supported by a contract from the National Institute of Allergy and Infectious Diseases, a component of the National Institutes of Health in the Department of Health and Human Services. Data Last Updated: June 04, 2017'.

Fig. 1 IEDB homepage where users can start accessing data by performing queries based on epitopes, assays, antigens, MHC restriction type, hosts and diseases. IEDB. Available at: <http://www.iedb.org>.

WELCOME! to the IMGT Home page

THE INTERNATIONAL IMMUNOGENETICS INFORMATION SYSTEM®



IMGT®	IMGT®, the international ImMunoGeneTics information system® http://www.imgt.org , is the global reference in immunogenetics and immunoinformatics, created in 1989 by Marie-Paule Lefranc (Université de Montpellier and CNRS). IMGT® is a high-quality integrated knowledge resource specialized in the immunoglobulins (IG) or antibodies, T cell receptors (TR), major histocompatibility (MH) of human and other vertebrate species, and in the immunoglobulin superfamily (IgSF), MH superfamily (MhSF) and related proteins of the immune system (RPI) of vertebrates and invertebrates. IMGT® provides a common access to sequence, genome and structure Immunogenetics data, based on the concepts of IMGT-ONTOLOGY and on the IMGT Scientific chart rules. IMGT® works in close collaboration with EBI (Europe), DBJ (Japan) and NCBI (USA). IMGT® consists of sequence databases , genome database , structure database , and monoclonal antibodies database , Web resources and interactive tools .
References and News	IMGT founder and director: Marie-Paule Lefranc (Marie-Paule.Lefranc@igh.cnrs.fr), Université de Montpellier, CNRS, LIGM, IGH, SFR, Montpellier (France)
Contacts & Legal notices	The 2015 IMGT® Customer Satisfaction Survey The Quality Management System of IMGT® Montpellier France has been approved by Lloyd's Register Quality Assurance France SAS to the following Quality Management System Standard: ISO 9001:2008

IMGT databases IMGT/LIGM-DB (doc) LIGM, Montpellier, France Nucleotide sequences of IG and TR from 351 species (178 929 entries) IMGT/MH-DB ANR , BPRC, hosted at EBI Sequences of the human MH (HLA) IMGT/PRIMER-DB (doc) LIGM, Montpellier, France Oligonucleotides (primers) of IG and TR from 11 species (1 864 entries) IMGT/CLL-DB (bylaws) LIGM, Montpellier, France IG sequences from CLL, an initiative of the IMGT/CLL-DB group IMGT/GENE-DB (doc) LIGM, Montpellier, France International nomenclature for IG and TR genes from human, mouse, rat and rabbit (4 147 genes, 5 858 alleles) IMGT/3Dstructure-DB and IMGT/2Dstructure-DB (doc) LIGM, Montpellier, France 3D structures (IMGT Colliers de Perles) of IG antibodies, TR, MH and RPI (4 808 entries) Source: PDB, INN, Kabat IMGT/mAb-DB (doc) LIGM, Montpellier, France Monoclonal antibodies (IG, mAb), fusion proteins for immune applications (FPIA), composite proteins for clinical applications (CPCA), and related proteins (RPI) of therapeutic interest (722 entries)	IMGT Web resources IMGT Repertoire (IG and TR, MH and RPI) IMGT Scientific chart (Sequence and 3D structure identification and description, Numbering, Nomenclature, Representation rules) IMGT Index (FactsBook, IMGT-ONTOLOGY, Sequence submission, Taxonomy...) IMGT Bloc-notes (Interesting links, PubMed, Meeting announcements, Postdoctoral positions and jobs, Messages, Search engines...) IMGT Education (IMGT Lexique, Aide-mémoire, Tutorials, Questions and answers, Enseignements...) IMGT Posters and diaporama The IMGT Medical page The IMGT Veterinary page The IMGT Biotechnology page The IMGT Immunoinformatics page
--	--

Fig. 2 The IMGT homepage with links to the seven databases. IMGT®. Available at: <http://www.imgt.org>.

2. IMGT/MH-DB specializes in human major histocompatibility complex sequences (Robinson *et al.*, 2015). Established as a locus-specific database for the allelic sequences of the HLA genes, the database also includes official sequences specified by the World Health Organization Nomenclature Committee For Factors of the HLA System. As of June 2017, there are 12,351 HLA class I alleles, 4404 HLA class II alleles, and 178 non-HLA alleles in the database.
3. IMGT/PRIMER-DB is a database of oligonucleotides (primer) for immunoglobulin and T-cell receptor. The primers can be used for antibody single chain Fragment variable (scFv), combinatorial library design, microarray technologies and phage display. The database comprises 1864 primers from 11 species.
4. IMGT/CLL-DB is a database specializing in primary immunoglobulin sequences associated with clinical and biological data of Chronic Lymphocytic Leukemia (CLL) patients. There are current restrictions to the database for members of the IMGT/CLL-DB group.
5. IMGT/GENE-DB is a comprehensive database for immunoglobulin and T-cell receptor genes from human, mouse, rat and rabbit (Giudicelli *et al.*, 2005). The database is the international reference for the gene nomenclature of immunoglobulin and T-cell receptor, displaying gene data related to genome, allelic polymorphisms, gene expression, protein and structures. As of June 2017, the database contains 4147 genes and 5858 alleles from 24 species.
6. IMGT/3Dstructure-DB is a resource on annotated structural data of immunoglobulin, T-cell receptor, MHC and related proteins of the immune system (Ehrenmann *et al.*, 2010). With 4766 entries currently, each record provides information on the sequence, 2D structures and 3D structures.
7. IMGT/mAB-DB is a monoclonal antibodies database, providing resource on immunoglobulins or monoclonal antibodies with clinical indications, as well as fusion proteins for immune applications. As of June 2017, the database contains 722 records.

SYFPEITHI

One of the first publicly available database for MHC ligands and peptide motifs, SYFPEITHI (see Relevant Websites section) is a database of known peptide sequences bound to MHC class I and II molecules (Rammensee *et al.*, 1995). It was developed in 1999

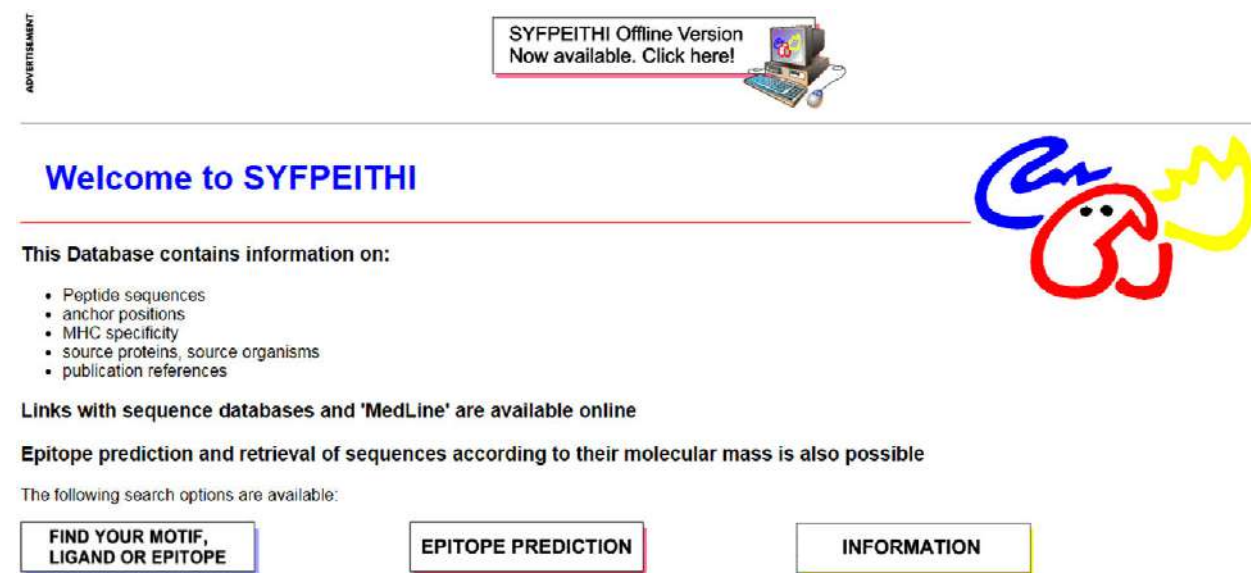


Fig. 3 Homepage of SYFPEITHI, with links to access the database. SYFPEITHI. Available at: <http://www.syfpeithi.de>.

by Hans-Georg Rammensee's group at the Institute of Cell Biology, University of Tübingen. With over 7000 MHC ligands, motifs and T-cell epitopes, the database aims to facilitate the search for peptides and aiding in the prediction of T-cell epitopes. Manually curated from published literature, each record provides information on the source and reference (Fig. 3).

AntiJen

AntiJen (see Relevant Websites section) is a large compilation of quantitative binding data (McSparron *et al.*, 2003). Developed by Darren Flower's group at the Edward Jenner Institute for Vaccine Research, the database was developed with the aim of improving computational vaccinology for building tools to accurately predict epitopes. The database includes MHC ligand molecules and kinetics, T cell epitopes, Transmembrane Peptide Transporter and B cell epitopes, protein complexes and protein-protein interactions, as well as data peptide library, diffusion coefficient and copy numbers. With over 24,000 entries based on experimentally determined data, AntiJen is among the largest immune epitope databases available (Fig. 4).

MHCBN

MHCBN (see Relevant Websites section) is a comprehensive database on MHC binding and non-binding peptides curated from existing databases and published literature (Lata *et al.*, 2009). The database was developed by Gagendra Raghava's group at the Institute of Microbial Technology in India. With over 25,800 peptide sequences, the database provides sequence and structure data on the source proteins as well as the MHC molecules (Fig. 5).

MPID-T2

The MHC-Peptide Interaction Database-TR version 2 (MPID-T2; see Relevant Websites section) is a new generation database with information on the sequence-structure-function of T-cell receptor/peptide/MHC interactions (Tong *et al.*, 2006). First developed at National University of Singapore in 2006 by Shoba Ranganathan's group, and subsequently at the Macquarie University in 2010, the MPID-T2 contains information on all known structures of T-cell receptor/peptide/MHC and peptide/MHC complexes, emphasizing on the structural characterization. Protein interaction information such as hydrogen bonds, non-hydrogen bonds, solvent accessibility, contact residues, interface area, gap index and gap volume is also included. Manually curated and populated with data from the Protein Data Bank (PDB), the database contains 415 records from humans (282), murine (127), rat (3), chicken (2), and monkey (1), which spans across 56 alleles. The database aims to facilitate the development of tools to predict peptide binding to MHC alleles (Fig. 6).

BEID

The B-cell Epitope Interaction Database (BEID) is a repository of sequence-structure-function information on interactions of immunoglobulin-antigen (Tong *et al.*, 2008). The database contains 164 antigens, 126 immunoglobulins and 189 immunoglobulin-antigen complexes extracted from PDB where they were manually verified, classified, and then analysed for the intermolecular

Full Search: [MHC Ligand](#) [MHC Kinetics](#) [T Cell Epitope](#) [TCR-MHC](#) [TAP Ligand](#) [Protein Interactions](#) [Linear B Cell Epitope](#) [Keywords](#)
[Discontinuous B Cell Epitope](#) [Diffusion Coefficient](#) [Peptide Libraries](#) [Copy Numbers](#) [Antibody binding](#) [AntiJen *BLAST](#)

Quick Search: [Help](#)

Welcome to the AntiJen Database v2.0.

AntiJen v2.0, is a database containing quantitative binding data for peptides binding to [MHC Ligand](#), [TCR-MHC Complexes](#), [T Cell Epitope](#), [TAP](#), [B Cell Epitope](#) molecules and [immunological Protein-Protein interactions](#). Most recently, AntiJen has included [Peptide Library](#), [Copy Numbers](#) and [Diffusion Coefficient](#) data. All entries are from published experimentally determined data. The database currently holds over 24,000 entries. No data in AntiJen is from prediction experiments..

[JenPep](#)^{1,2} established a basic system which has now undergone major advancements. **AntiJen v2.0** not only contains a wider spectrum of data but also demonstrates superior search capabilities. The expanded and updated AntiJen database currently accommodates data and look-up access for:

- MHC Ligand molecules and MHC Ligand kinetics,
- T Cell Epitope, TAP and B Cell Epitope molecules,
- Protein-Protein interactions and Protein complexes,
- Peptide Library, Diffusion Coefficient and Copy Numbers data.

Development of sophisticated search mechanics incorporates the flexibility to conduct a very detailed search or a broad search from a single interface. For example,

- the search engine can accept epitope strings containing variable amino acid positions,
- specific amino acid alternatives can be requested in any epitope string,
- an optional filter can be used to target experimental data of interest,
- delimiting windows can be used to narrow searches further by epitope length or data size.

Other novel search functionality includes:

- Quick Search, this provides instant cross-category database access;
- Keyword Search, AntiJen may be searched using keyword input alone .

References

1 Blythe MJ, Doytchinova IA, Flower DR. *JenPep: a database of quantitative functional peptide data for immunology*. [Bioinformatics 2002 18 434-439](#)

2 McSpanon H, Blythe MJ, Zygouri C, Doytchinova IA, Flower DR. *JenPep: A Novel Computational Information Resource for Immunobiology and Vaccinology*. [J Chem Inf Comput Sci. 2003 43:1276-1287](#)

[\[Edward Jenner Institute \]](#) [\[Drug Design Group \]](#) [\[Links \]](#) [\[Help \]](#) [\[Contact Us \]](#)

Fig. 4 Homepage of AntiJen with quick search functions for epitopes. AntiJen. Available at: <http://www.ddg-pharmfac.net/antijen/>.

interactions. Each record contains interaction information on gap index, gap volume, interface area, solvent accessibility, contact residues, hydrogen bonds and non-hydrogen bonds.

CED

The Conformational Epitope Database (CED; see Relevant Websites section) is a specialist database on conformational B cell epitopes (Huang and Honda, 2006). Manually curated from peer-reviewed literature, each entry contains information on the location and composition of the epitope, immunological property, source antigen and corresponding immunoglobulin. As of June 2017, the database contains 225 entries from protein antigens, nucleic acids, glycans and lipids, derived from various sources.

DFRMLI

The Dana-Farber Repository for Machine Learning in Immunology (DFRMLI; see Relevant Websites section) is a database specializing in standardized datasets for the application of machine learning in immunology (Zhang *et al.*, 2011). Processed specifically for the development of machine learning tools for use in epitope prediction, the repository provides standard datasets of experimentally validated binding affinities of HLA-binding peptides mapped onto a common scale.

MHCBN Version 4.0

ing with TAP. The database is cross-linked to NCBI, Swiss-Prot, PDB, OMIM and GenBank.

The MHCBN is a curated database consisting of detailed information about Major Histocompatibility Complex (MHC) Binding, Non-binding peptides and T-cell epitopes. The version 4.0 of database provides information about peptides interacting with TAP and MHC linked autoimmune diseases. This database is Developed by [Dr Raghava's Group](#), at [Bioinformatics Centre, Institute of Microbial Technology, Chandigarh, INDIA](#).

What MHCBN can do for you

Resources	The database provides information about allele specific MHC binding peptides, MHC non-binding, TAP binding, TAP non-binding peptides and T-cell epitopes reported in literature	
Hypertext Links	Antigenic Protein	Source or Antigenic proteins are linked to SWISS-PROT database
	MHC Sequence	MHC Allele sequence are hyperlinked to IMGT/HLA-DB
	3D-Structure	The hypertext link has been made to Protein Databank(PDB) if any identical/related peptide/MHC protein is available in PDB.
	Diseases	The MHC linked diseases has been hyperlinked to OMIM.
	References	For full information references are linked to PubMed
Web Tools	Query Searching	The web server has tools(General query &Peptide search) to extract all information from the fields of database.
	TAP Search	The tool has been designed for searching the part of sequence interacting with TAP.
	Peptide Mapping	This option allows mapping of binders, non-binder and T-cell epitope in query protein
	Dataset Creation	This allows Creation of Allele specific peptide datasets useful for developing prediction methods
	Similarity Search	This option allows BLAST search of query protein sequence against antigenic and MHC proteins in database

The Database can be accessed from [Mirror site at IAMS](#)

Fig. 5 MHCBN homepage with information and links to peptide query. MHCBN. Available at: <http://imtech.res.in/raghava/mhcbn>.

MPID-T2

HOME SEARCH STATISTICS PATTERNS ALIGNMENT REFERENCES CONTACT US HELP

About MPID-T2:

The MHC-Peptide Interaction Database-TR version 2 (MPID-T2) is a new generation database for sequence-structure-function information on T cell receptor/peptide/MHC interactions. It contains all known crystal structures of TR/pMHC and pMHC complexes, with emphasis on the structural characterization of these complexes.

MPID-T2 will facilitate the development of algorithms to predict whether a peptide sequence will bind to a specific MHC allele forming a pMHC agonist which will consequently be recognized by a particular TR leading to T cell activation for immune response. The database has been populated with data from the Protein Data Bank(PDB). The data from PDB is manually verified and classified, after which each structure is analysed for atomic interactions relevant to pMHC and TR/pMHC complexes.

The intermolecular hydrogen bonds are calculated using the program [HBPLUS](#), gap volume is pre-computed using the program [SURFNET](#) and change in interface area due to complex formation is calculated based on accessible surface area calculation performed by the [NACCESS](#) program for structures released until December 2006 and by the [ICM](#) program for structures released after December 2006. Gap index is calculated using the values for gap volume and interface area. Schematic diagrams of the pMHC and TR/pMHC interactions based on the plotting program [LIGPLOT](#) are also available. The peptide consensus patterns available in MPID-T2 are generated using the program [WEBLOGO](#).

MPID-T2 (November 2010 update) contains 415 entries from five MHC sources (Human:282, Murine:127, Rat:3, Chicken:2 and Monkey:1), spanning 56 alleles. [353](#) of which are pMHC structures and [62](#) TR/pMHC complexes. MHC-I (class I pMHC and TR/pMHC) complexes number 352 while 63 structures are from MHC-II (class II pMHC and TR/pMHC) complexes. On the whole, 327 entries are non-redundant (279 from MHC-I and 48 from MHC-II complexes). Among the TR/pMHC structures 50 are MHC-I and 11 are MHC-II structures. Included in this version are non-classical structures and complexes with non-standard residues. MPID-T2 is updated on a quarterly basis. To view the distribution graphically [click here](#). To visualize the structural alignment you need [Chime](#) or [Java](#) plug-in for [Jmol](#). MPID-T2 is best viewed using Internet Explorer.

MPID-T2 Team:

MPID-T2 is proudly brought to you by [Javed Mohammed Khan](#), [Harish Reddy Cheruku](#), [Joo Chuan Tong](#) and [Shoba Ranganathan](#)

Copyright 2010 Department of Chemistry and Biomolecular Sciences, Macquarie University, Sydney

Fig. 6 MPID-T2 homepage with information and links to access the database. MHCBN. Available at: <http://biolinfo.org/mpid-t2>.

IPD

The Immuno Polymorphism Database (IPD; see Relevant Websites section) is a compilation of specialist databases associated with the study of gene polymorphisms in the immune system (Robinson *et al.*, 2013). It was developed in 2003 from a collaboration between the European Bioinformatics Institute (EBI) and the HLA Informatics Group of the Anthony Nolan Research Institute. The IPD comprises four databases: (Fig. 7).

1. IPD-KIR provides a repository on human Killer-cell Immunoglobulin-like Receptors (KIR) sequences. There are over 600 alleles in the database, coding for more than 320 unique sequences.
2. IPD-MHC is a centralised database of MHC sequences from different species. Manually curated by field experts, the database now contains over 4000 alleles from 47 species of non-human primates, as well as other species including canines, felines, cattle, teleost fish, rats, sheep, and swine.

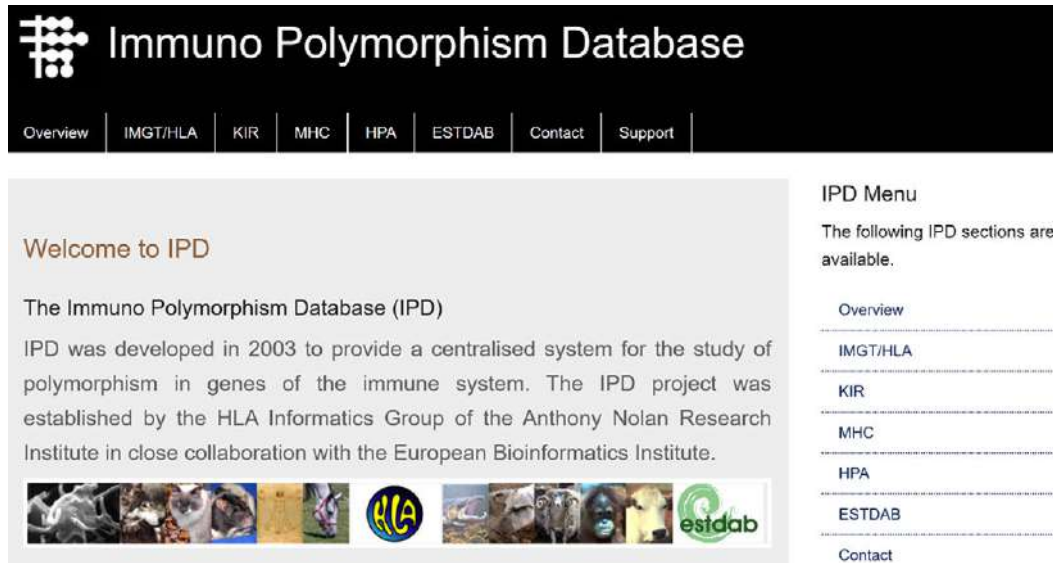


Fig. 7 The IPD homepage with links to four specialist databases.

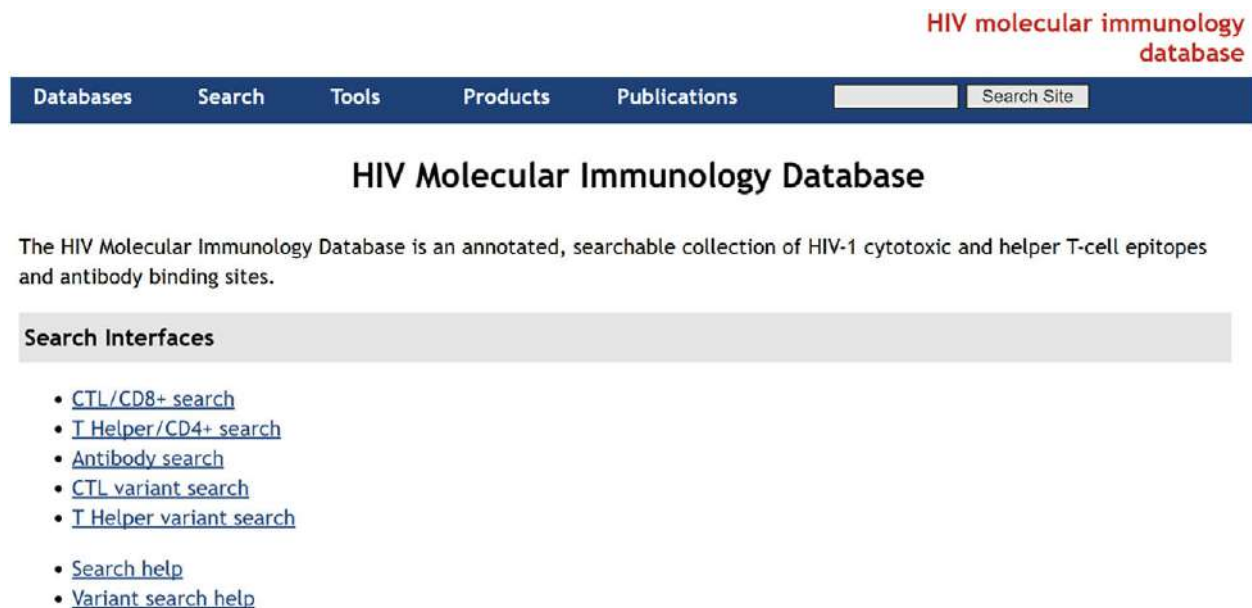


Fig. 8 Homepage of HIV Molecular Immunology Database with links to search interfaces.

3. IPD-HPA details data on the human platelet antigens (HPA). The database provides information on the HPA as well as additional background information.
4. IPD-ESTDAB is the European Searchable Tumour Line Database and Cell Bank for immunologically characterized melanoma cell lines. The database provides a search facility for HLA typed, immunologically characterized tumour cells.

HIV Molecular Immunology Database

The HIV Molecular Immunology Database (see Relevant Websites section) is an annotated database of HIV-1 cytotoxic and helper T-cell epitopes, and immunoglobulin binding sites. The database is funded by the Division of AIDS, National Institute of Allergy and Infectious Diseases and developed by the Los Alamos National Laboratory. The data is extracted from HIV immunology literature, and includes information on immunoglobulin sequence, escape mutations, cross-reactivity, T-cell receptor usage, functional domains overlapping with epitopes, immune response among others. As of June 2017, there are 905 HIV-1 cytotoxic T cell epitopes, 1023 HIV-1 helper T-cell epitopes and 1448 immunoglobulin binding sites in the database (Fig. 8).

Discussion

The immunoinformatics field has matured over the past decade, with the creation of many high-quality immunological databases available publicly. Many of the databases discussed above contain well-curated and comprehensive data on MHC ligands and immune epitopes, including both naturally processed peptides as well as synthetic peptides available on IEDB. Some of the databases such as IEDB, AntiJen, MHCBN, MPID-T2 and BEID also include information on the structure and protein-protein interactions which provides much more detail to study individual binding interactions. On the other hand, there are also specialized databases such as DFRMLI in addition to IEDB and IMGT, which provide training datasets with large number of MHC-peptide binding data to facilitate the development of machine learning approaches for epitope prediction. While there are a multitude of databases available, the challenge remains to update the database with new information regularly so that more tools would be able to utilize the rich resource to advance the field of immunoinformatics.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Computational Immunogenetics. Data Storage and Representation. Extraction of Immune Epitope Information. Immunoglobulin Clonotype and Ontogeny Inference. Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics. Vaccine Target Discovery

References

- Ehrenmann, F., Kaas, Q., Lefranc, M.P., 2010. IMGT/3Dstructure-DB and IMGT/DomainGapAlign: A database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF. *Nucleic Acids Res.* 38 (Database Issue), D301–D307.
- Flieri, W., *et al.*, 2017. The immune epitope database and analysis resource in epitope discovery and synthetic vaccine design. *Front. Immunol.* 8, 278.
- Galperin, M.Y., Fernandez-Suarez, X.M., Rigden, D.J., 2017. The 24th annual nucleic acids research database issue: A look back and upcoming changes. *Nucleic Acids Res.* 45 (9), 5627.
- Giudicelli, V., *et al.*, 2006. IMGT/LIGM-DB, the IMGT comprehensive database of immunoglobulin and T cell receptor nucleotide sequences. *Nucleic Acids Res.* 34 (Database Issue), D781–D784.
- Giudicelli, V., Chaume, D., Lefranc, M.P., 2005. IMGT/GENE-DB: A comprehensive database for human and mouse immunoglobulin and T cell receptor genes. *Nucleic Acids Res.* 33 (Database Issue), D256–D261.
- Huang, J., Honda, W., 2006. CED: A conformational epitope database. *BMC Immunol.* 7, 7.
- Kimbrell, D.A., Beutler, B., 2001. The evolution and genetics of innate immunity. *Nat. Rev. Genet.* 2 (4), 256–267.
- Lata, S., Bhasin, M., Raghava, G.P., 2009. MHCBN 4.0: A database of MHC/TAP binding peptides and T-cell epitopes. *BMC Res. Notes* 2, 61.
- Lefranc, M.P., *et al.*, 2015. IMGT(R), the international ImMunoGeneTics information system(R) 25 years on. *Nucleic Acids Res.* 43 (Database Issue), D413–D422.
- McSparron, H., *et al.*, 2003. JenPep: A novel computational information resource for immunobiology and vaccinology. *J. Chem. Inf. Comput. Sci.* 43 (4), 1276–1287.
- Murphy, K.P., *et al.*, 2008. *Janeway's Immunobiology*. New York, NY: Garland Pub.
- Rammensee, H.G., Friede, T., Stevanović, S., 1995. MHC ligands and peptide motifs: First listing. *Immunogenetics* 41 (4), 178–228.
- Robinson, J., *et al.*, 2013. IPD – The Immuno polymorphism database. *Nucleic Acids Res.* 41 (Database Issue), D1234–D1240.
- Robinson, J., *et al.*, 2015. The IPD and IMGT/HLA database: Allele variant databases. *Nucleic Acids Res.* 43 (Database issue), D423–D431.
- Tong, J.C., *et al.*, 2006. MPID-T: Database for sequence–structure–function information on T-cell receptor/peptide/MHC interactions. *Appl. Bioinform.* 5 (2), 111–114.
- Tong, J.C., *et al.*, 2008. BEID: Database for sequence–structure–function information on antigen–antibody interactions. *Bioinformatics* 3 (2), 58–60.
- Zhang, G.L., *et al.*, 2011. Dana-Farber repository for machine learning in immunology. *J. Immunol. Methods* 374 (1–2), 18–25.

Relevant Websites

- <http://www.ddg-pharmfac.net/antijen/>
AntiJen.
- <http://immunet.cn/ced>
CED.
- <http://bio.dfci.harvard.edu/DFRMLI>
DFRMLI.

<http://www.hiv.lanl.gov/content/immunology>
HIV Molecular Immunology Database.

<http://www.iedb.org>
IDEB.

<http://www.imgt.org>
IMGT.

<http://www.ebi.ac.uk/ipd>
IPD.

<http://imtech.res.in/raghava/mhcbn>
MHCBN.

<http://biolinfo.org/mpid-t2>
MPID-T2.

<http://www.syfpeithi.de>
SYFPEITHI.

Study of Human Antibody Responses From Analysis of Immunoglobulin Gene Sequences

Katherine J.L. Jackson, Garvan Institute of Medical Research, Darlinghurst, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Antibodies are a crucial component of the immune system and the genes that encode antibodies are subject to more manipulation and modification than any other human genes. The many processes that occur during antibody ontogeny are captured within the sequence of the rearranged immunoglobulin genes. The study of the sequences of these genes therefore provides a window into not only the antigen response but also into the underlying genetics of the person who produced the response. Sequencing technologies that capture millions of immune receptor gene transcripts offer a means for high resolution analysis of the antibody response in contexts ranging from responses to infection (Wu *et al.*, 2011; Parameswaran *et al.*, 2013; Liao *et al.*, 2013) and vaccination (Wang *et al.*, 2015; Jackson *et al.*, 2014; Galson *et al.*, 2015; Ellebedy *et al.*, 2016; Truck *et al.*, 2015), to autoimmunity and immune deficiency (Roskin *et al.*, 2015), in allergy (Hoh *et al.*, 2016; Wang *et al.*, 2014; Levin *et al.*, 2016) and in lymphoid cancers (Boyd *et al.*, 2009; Faham *et al.*, 2012).

Each antibody protein in the human is comprised of four polypeptide chains – two identical heavy chains encoded by an immunoglobulin heavy chain (IGH) gene and two identical light chains encoded by either an immunoglobulin kappa (IGK) or lambda light chain (IGL) gene (Fig. 1) (Tonegawa, 1983). Each polypeptide chain is broadly divided into two regions; the variable region and the constant region. The variable portion of immunoglobulin genes are formed by a series of genomic rearrangements within each B cell's genome at the heavy (chromosome 14 (Croce *et al.*, 1979)) and light chain loci (kappa: chromosome 2, lambda: chromosome 22 (McBride *et al.*, 1982)) that bring together several gene segments to form an immunoglobulin gene (Tonegawa, 1983) (Fig. 1(A)). The resulting gene is often termed a 'rearranged' immunoglobulin gene in order to differentiate it from the unrearranged or 'germline' gene segments within the immunoglobulin gene loci that are the building blocks from which functional antibodies are created. The process of genomic recombination allows for a vast diversity of antibody proteins to be generated from a relatively small number of variable (IGHV), diversity (IGHD) and joining (IGHJ) gene segments for the heavy chain and variable (IGKV/IGLV) and joining (IGKJ/IGLJ) gene segments for the light chains. The same V(D)J recombination mechanism also forms the T cell receptor (TCR) gene repertoire (Davis, 1990).

The variable region of an immunoglobulin protein provides the antigen specificity. It can be further sub-divided into four structurally important framework regions (FR1, 2, 3, 4) and three diverse complementarity determining regions (CDR1, 2, 3) that tend to contact the antigen (Chothia and Lesk, 1987). Immunoglobulin protein function is dictated by the constant region, encoded by the heavy chain constant region (IGHC) gene, which determines the isotype or subclass of the immunoglobulin (Schroeder and Cavacini, 2010). IGHG exons are associated with the variable region at the level of mRNA transcription and each isotype class and subclass has varying in its functional capabilities. Prior to antigen exposure, naïve B cells express their VDJ rearrangement in association with both IgM and IgD through alternative mRNA splicing (Fig. 1(D)). Following antigen encounter, further genomic changes, termed class switch recombination (CSR), alter the associated constant region by deletion of IGHG genes from the genome and allowing expression of IgG subclasses (IgG3, IgG1, IgG2, IgG4), IgA subclasses (IgA1, IgA2), or IgE (Fig. 1(D)). The constant region also determines if paired heavy and light chains are secreted from the B cell as antibodies, or if they are membrane bound and act as B cell receptors (BCRs). This is based on whether the constant region includes a terminal secretory (antibodies) or trans-membrane domain (BCRs). Light chains also have constant regions encoded by IGKC or IGKL genes but these do not undergo CSR.

In addition to CSR, following encounter with an antigen to which the BCR can bind, immunoglobulin genes will also undergo affinity maturation (Fig. 1(E)) (Eisen, 2014). This is driven by a mutation process termed somatic hypermutation (SHM) because the mutations are introduced within the V(D)J gene rearrangement at a rate of 10^5 to 10^6 above the normal background somatic mutation rate (Weigert *et al.*, 1970) (Bernard *et al.*, 1978). The point mutations introduced within the V(D)J may be positively selected when they improve the interaction between the antigen and the immunoglobulin and negatively selected when they are deleterious. CSR and affinity maturation occur in the context of the B cell clonal expansion whereby the initial B cell that expressed an immunoglobulin with some specificity for antigen divides and each daughter cell inherits the immunoglobulin rearrangement, but may acquire further SHM and undergo further class switch, to build a B cell clonal lineage.

Diversification of Immunoglobulin Genes

The diversity of the human antibody repertoire within a single person is predicted to be as high as 10^{11} unique antibodies (Glanville *et al.*, 2009). Given that a human genome includes fewer than 20,000 protein-coding genes (Ezkurdia *et al.*, 2014), it is not possible for a single gene for each unique antibody to be maintained within the germline genome, rather a diverse immunoglobulin repertoire is generated through processes that involve rearrangements and modifications to the genomes of single immune cells (Tonegawa, 1983). Immunoglobulin diversification processes can be broadly divided into 'combinatorial' and

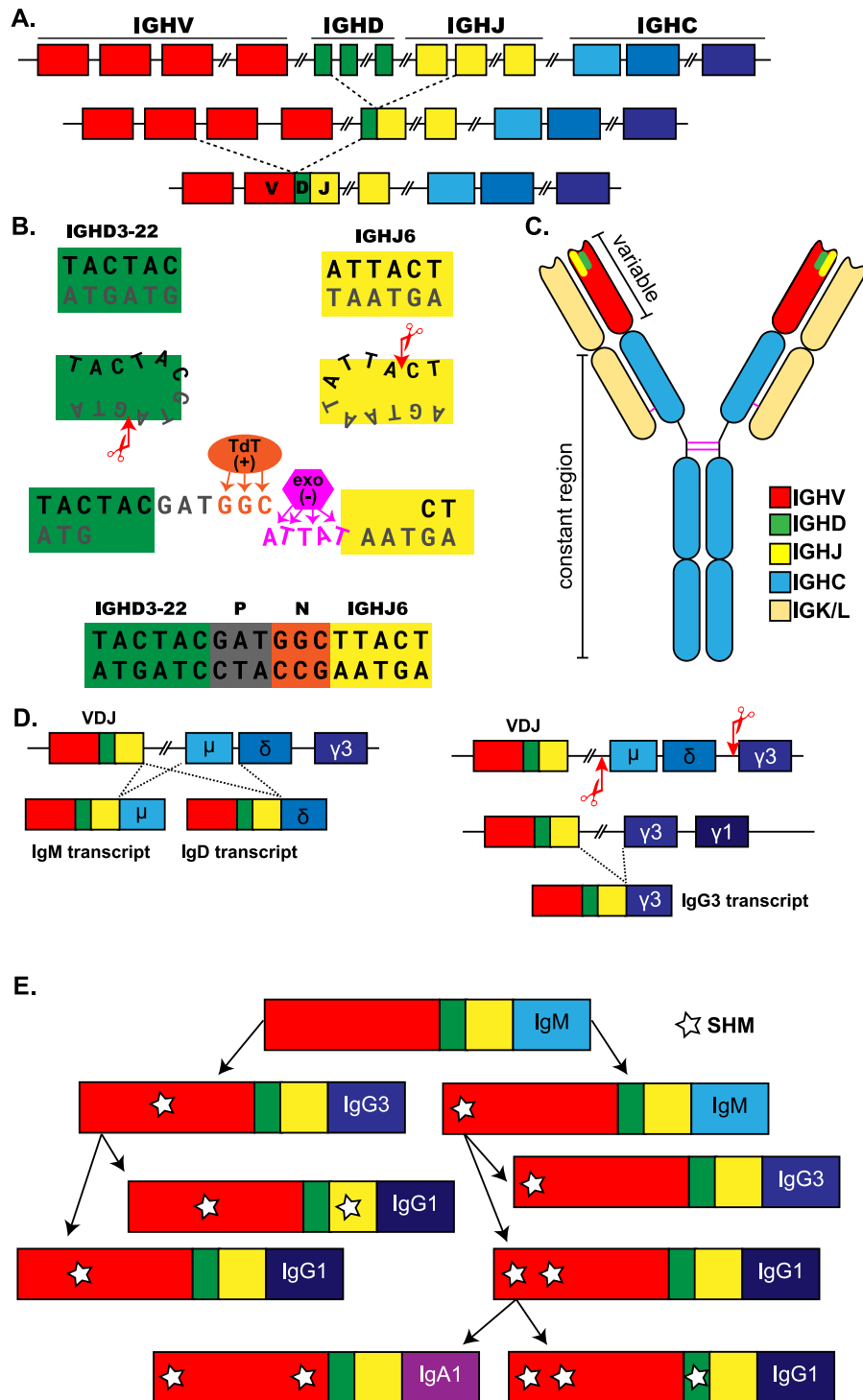


Fig. 1 Rearrangement of immunoglobulin genes and the generation of diversity in humans. (A) The human IGH locus is comprised of sets of IGHV, IGHD and IGHJ gene segments. A rearranged immunoglobulin gene is generated by genomic recombination events that first join single IGHD and IGHJ gene segments, followed by joining to a single IGHV gene segment. (B) Joining of gene segments is imprecise as hairpin structures that form following RAG cleavage at RSS sites are opened asymmetrically and the joints are acted on by exonucleases that remove nucleotides from coding ends and TdT which incorporates untemplated nucleotides, leading to diverse junctions. (C) The IGH and light chain polypeptide chains are joined by disulfide bonds (pink) to form an immunoglobulin protein which is broadly divided into variable and constant regions. (D) Rearranged immunoglobulin genes are associated with IGHC exons at the mRNA level, naïve B cells can express both IgM and IgD through alternative splicing. Genomic recombination events can alter the IGHC locus leading to expression of downstream isotypes classes and subclasses. (E) Rearranged genes undergo somatic hypermutation and class switch recombination as part of the B cell clonal expansion to give rise to B cell lineages.

'junctional' which generate the vastly diverse naïve B cell compartment (**Fig. 1(A)** and **(B)**). Naïve B cells are those that are yet to experience foreign antigen interactions. Diversification subsequent to antigen encounter is SHM driven.

Combinatorial Diversity

IGH are formed via the joining of three distinct gene segments; IGHV, IGHD and IGHJ. Sets of each type of gene segments are present within the IGH locus on chromosome 14. IGH gene segments are both polygenic and polymorphic. Within the human population each gene may exist as a number of allelic variants with any single individual carrying just one (homozygous) or two (heterozygous) of the possible alleles ([Watson et al., 2013](#)). More than two variants of some genes, for example IGHV1–69, may be carried by a single person due copy number variant haplotypes ([Watson et al., 2013](#)).

Functional IGH gene segments are flanked by recombination signal sequences (RSS) ([Tonegawa, 1983](#)). Each RSS is composed of a conserved heptamer (6 base pairs) and nonamer (9 base pair) sequence separated by a variable spacer region of either 12 ± 1 (IGHD segments) or 23 ± 1 (IGHV and IGHJ segments) base pairs (bp) in length. Only genes with different spacer lengths can efficiently recombine and this 12/23 rule promotes the preferential formation IGHD: IGHJ and IGHV: IGHDJ joins. The formation of the rearranged immunoglobulin variable gene that will eventually be transcribed as the IGH begins with the pairing of single IGHD and IGHJ segments from the approximately 25 IGHD and 6 IGHJ segments within an individuals' IGH locus. Subsequently, the IGHD: IGHJ will be joined to one of between 45 to 50 functional IGHV segments ([Boyd et al., 2010](#); [Kidd et al., 2012](#); [Watson et al., 2013](#)) to give the rearranged IGH variable region gene.

The recombination of select gene segments to generate the final gene sequence provides for approximately 22,500 unique IGH rearrangements as the IGHDs can be used in all three reading frames (RFs). The light chain provides for approximately 200 different IGKs by recombining 34 to 40 IGKV to 5 IGKJ and 165 lambda light chains combinations from joining between 29 to 33 IGLV and 5 IGLJ. The pairing of heavy and light chains therefore provides for just over 8 million unique antibodies.

A number of biases in the joining of particular segments to each other have been observed, for example, a tendency for 3' IGHD genes to preferentially pair with 5' IGHJ genes ([Kidd et al., 2016](#); [Volpe and Kepler, 2008](#); [Souto-Carneiro et al., 2005](#)). This may be the result of RSS efficiency or other mechanisms such as chromatin structure that impact gene segment accessibility.

Junctional Diversity

The approximately 8 million immunoglobulins formed from combinatorial diversity accounts for only a fraction of the predicted diversity of the immunoglobulin repertoire within a single individual's repertoire. Further diversity stems from the imprecise nature of the gene segment joining with nucleotides added and removed during the recombination process (**Fig. 1(B)**) ([Alt and Baltimore, 1982](#); [Lafaille et al., 1989](#)). Joining of two gene coding segments begins with the recruitment of the lymphocyte specific Recombination Activation Gene (RAG) complex consisting of RAG-1 and RAG-2 proteins to the RSS elements ([van Gent et al., 1995](#)). The RAG complex introduces a nick between the RSS and the adjacent gene coding segment that is converted a double-stranded break with the formation of a covalently sealed hairpin on the gene coding side of the break ([McBlane et al., 1995](#)).

The hairpin loops at the gene segment ends are opened by a complex formed between the nuclease Artemis and the catalytic subunit of the DNA-dependent kinase (DNA-PK_{CS}) ([Ma et al., 2002](#)). Artemis preferentially cleaves the hairpin loop at a position 3' to the tip generating a single stranded overhang (**Fig. 1(B)**). The single stranded overhang is an inverted repeat of the sequence of the end of the gene segment and this sequence may be incorporated into the rearranged gene as palindromic (P) nucleotides. The overhangs of the open hair pins are processed in two ways - exonuclease trimming and non-template encoded (N) addition ([Lafaille et al., 1989](#)). Exonuclease trimming may remove all P nucleotides, plus additional nucleotides from the coding sequence beyond the original break. The nuclease responsible is unknown. N nucleotide addition arises from the incorporation of free dNTPs to the 3' end of DNA by the short isoform of nuclear enzyme terminal deoxynucleotidyl transferase (TdT) within the joint ([Benedict et al., 2000](#)). These regions are termed the N-regions (N1 at the IGHV to IGHD join and N2 at the IGHD to IGHJ join). TdT preferentially incorporates G. Following processing, the joint between the two gene segment ends is resolved by non-homologous DNA end joining ([Lieber, 1999](#)).

Somatic Hypermutation

SHM expands the diversity of the immunoglobulin repertoire and tailors antibodies and BCRs for higher affinity interactions with antigen (**Fig. 1(E)**). SHM takes place within specialized micro-environments within secondary lymphoid tissue call germinal centers (GC) ([Bannard and Cyster, 2017](#)). Within the GC, B cells undergo rapid clonal expansion, during which the B cell's immunoglobulin genes mutate at the rate of approximately 10^{-3} changes per nucleotide per cell division. Point mutations are introduced into the domain that starts approximately 150 bp downstream from the IGHV promoter and extends 2kpbs downstream with a frequency that decays exponentially from the IGHV promoter ([Rada and Milstein, 2001](#)).

Substitutions are targeted to intrinsic mutational hotspots ([Jolly et al., 1996](#)). Activation induced cytosine deaminase (AID) is specifically expressed in activated B cells and undertakes targeted deamination of deoxycytidine residues ([Muramatsu et al., 2000](#)). Deamination of C by AID changes C:G base pairs to U:G mismatches and is targeted to WRC motifs with a major intrinsic hotspot being overlapping WRC motifs on the coding and non-coding strands – WGCW – which suggests that AID binds as an oligomer ([Ta et al., 2003](#)). The U:G mismatch can be resolved in a number of ways. Replication using the mismatching template can produce transition mutations where C/G are replaced by A/T. Alternatively, the U can be removed by uracil-DNA glycosylase such as UNG2 creating an apyrimidinic site (AP) which can be repaired by base excision repair (BER) ([Teng and Papavasiliou, 2007](#)). As BER

results in untemplated replication opposite the AP site any dNTP can be incorporated at the position leading to either reversion, transition or transversion mutations. The nucleotide inserted is dependent on the DNA polymerase involved in the BER as polymerases are biased in the nucleotides that they insert.

Targeting of mutations to A:T pairs may arise through several proposed mechanism (Teng and Papavasiliou, 2007). For example, the binding of MSH2-MSH6 at U:G mispairs can recruit exonuclease 1 where it can create a gap that when filled by an error prone polymerase leads to mutations at A:T pairs. Alternatively, during short patch repair of the AID U:G lesion, U may be inappropriately incorporated due to dNTP pool imbalances creating A:U mispairs in the region surrounding U:G lesion and the resolution of the A:U mispair may introduce mutations at A:T pairs.

Sequencing of Immunoglobulin Gene Repertoires

Each immunoglobulin transcript captures details of the gene segments that formed the rearranged gene, the processing the that gene segments underwent during recombination, and, if the B cell has encountered antigen, the mutations that have accumulated during affinity maturation, along with the isotype subclass that the VDJ is associated with. Amplicons from tens to hundreds of thousands of B cells may be sequenced from a single peripheral blood sample to capture the different members of clonal expansions. Furthermore, as VDJ rearrangement is an intra-chromosomal event, analysis of low mutation rearrangements can also inform on the underlying immune genotype and haplotype of the individual (Boyd *et al.*, 2010) (Kidd *et al.*, 2012). Repeated longitudinal sampling from a single subject over the course of an immune response, such as infection or vaccination, allows tracking of the origin and fate of B cell clones over the course of the response.

Several approaches for deep sequencing of immune receptor repertoires have been published (Jackson *et al.*, 2014; Galson *et al.*, 2015; Ellebedy *et al.*, 2016). The common themes to these protocols are amplification from cDNA to capture VDJ and isotype information and multiplexing of samples using nucleotide sequence barcodes to permit single libraries to be generated from many different samples. Some protocols incorporate unique molecular identifiers (UMI) at the cDNA synthesis step to tag each RNA molecule with unique identifier before PCR amplification to allow counting of single transcripts and for error correction (Turchaninova *et al.*, 2016). In addition to cDNA based protocols, the relationship between single B cells and amplicons may be maintained by genomic DNA amplification in replicate to study clonal expansions at the sacrifice of isotype information (Boyd *et al.*, 2009).

Full length VDJ amplified from leader primers, or near-full length from FR1 primers, with enough CH1 exon to discriminate IgG and IgA subclasses can be captured using paired end libraries sequences on the Illumina MiSeq platform with 600 cycle kits for in excess of 20 million reads. Longer read, lower throughput technologies, such as PacBio and Oxford Nanopore have potential to offer ability to capture full length VDJ plus constant region amplicons. Usually amplifications are from bulk cell populations, either total peripheral blood mononuclear cells or sorted cell populations from fluorescent or magnetic cell sorting to enrich to particular phenotypic cell subsets (eg. plasmablasts, memory B cells, activated B cells, or naïve B cells), however, immunoglobulin genes may also be studied from single B cells, to capture paired heavy and light chain gene transcripts (Busse *et al.*, 2014). Attempts have also been made to reconstruct rearranged immunoglobulin genes from single cell RNA transcriptome sequencing methods to permit linking of paired heavy and light chain transcripts to overall gene transcription within a cell (Rizzetto *et al.*, 2017).

Analysis of Immunoglobulin Gene Sequences

Partitioning of Rearranged Immunoglobulin Genes

The most basic challenge of the study of immune receptor genes through sequencing datasets is the partitioning of rearranged genes into their various components such as IGHV, IGHD and IGHJ segments and identification of nucleotide loss and addition of the IGHV-IGHD and IGHD-IGHJ joints to determine of SHM spectra.

A number of tools exist for this purpose (Table 1). The earliest contributions to the field treated the partitioning problem as three separate alignment problems; one for each gene segment. These include IMGT/V-QUEST (Brochet *et al.*, 2008), which appears to be based on a dynamic programming algorithm with a single recursive rule as alignment scores are re-created by a simple derivation of the Smith-Waterman local sequence alignment algorithm, and IgBLAST (Ye *et al.*, 2013) based on the Basic Local Alignment Search Tool (BLAST) algorithm (Camacho *et al.*, 2009). Contemporary tools have attempted to gain speed advantages to the alignment-approach by using modified kmer-chaining based methods (MiXCR (Bolotin *et al.*, 2015)), dynamic programming algorithms constrained by conserved motifs (HTJoinSolver (Russ *et al.*, 2015)) and fast-tag-searching algorithms (LymAnalyzer (Yu *et al.*, 2016)). IMGT/V-QUEST and IgBLAST remain perhaps the most widely used germline gene annotation tools for rearranged gene sequences. Only MiXCR identifies the associated isotype from the IGHC gene, otherwise this is undertaken a customized post-processing step using BLAST (Camacho *et al.*, 2009) or string matching.

The variety of processes that underlie the gene rearrangement however complicate the partitioning, particularly the discrimination of the components within the CDR3 that spans from the IGHV end, across the V-D junction, IGHD, D-J junction, through to the IGHJ start. Tools specific for the CDR3 were therefore developed. The first wave of such tools included IMGT/Junction Analysis (IMGT/JA (Yousfi Monod *et al.*, 2004)) and JOINSOLVER (Souto-Carneiro *et al.*, 2004). IMGT/JA uses a five-stage process to identify the P nucleotides, the N nucleotides, the IGHD, to consider the processing of IGHV and IGHJ ends, and to

Table 1 Example of immune receptor tools and pipelines available for processing and analysis of rearranged immune gene repertoire datasets

Name	Website	Purpose	Type
ALPHABETR	https://github.com/edwardslee/alphabetr	R package for inference of TCR alpha-beta pairing	Analysis tool
BASILINE	http://selection.med.yale.edu/baseline/	Antigen selection	Analysis tool
IgPhyML	https://github.com/kbhoehn/IgPhyML	Analysis of immunoglobulin phylogeny	Analysis tool
MaxSnippetModel	https://github.com/jostmey/MaxSnippetModel	Statistical classifier for immune repertoires	Analysis tool
RDI	http://bitbucket.org/cbolet1/rdicore/	Repertoire dissimilarity metric	Analysis tool
VDJSeq-Solver	http://eda.polito.it/VDJSeq-Solver/	Identification of clonal populations from neoplastic datasets	Analysis tool
Decombinator	https://github.com/innate2adaptive/Decombinator	TCR, string matching algorithm	Annotation
HTJoinSolver	https://dcb.cit.nih.gov/HTJoinSolver/	Partitioning by dynamic programming constrained by conserved motifs	Annotation
IgBLAST	https://www.ncbi.nlm.nih.gov/igblast/	BLAST-based partitioning of BCR and TCR via web-interface or stand-alone	Annotation
iHMMune-align	https://www.ncbi.nlm.nih.gov/igblast/	HMM partitioning of IGH	Annotation
IMGT/V-QUEST	http://www.imgt.org/IMGT_vquest/vquest	Partitioning tool using IMGT reference datasets via web-interface	Annotation
JOINSOLVER	https://joinsolver.niaid.nih.gov/	IGH partitioning with a focus on the confident determination of the IGH within the CDR3	Annotation
partis	https://github.com/psathyrella/partis/	HMM partitioning tool	Annotation
SoDA, SoDA2	No longer supported, succeeded to cleanalyst	Probabilistic partitioning of IGH	Annotation
TCRbiter	https://github.com/AstraZeneca-NGS/tcrbiter	TCR annotation tool	Annotation
TCRklass	https://sourceforge.net/projects/tcrklass	K-string based algorithm for TCR analysis	Annotation
VDJSolver	http://www.cbs.dtu.dk/services/VDJSolver/	IGH partitioning tool using Maximum Likelihood for model fitting	Annotation
IGoR	https://bitbucket.org/qmarcou/igor	Probabilistic alignment using a sparse Expectation-Maximization algorithm, three modes – statistics learning, sequence analysis and sequence generation.	Annotation & Simulation
IgDiscover	https://github.com/NBISweden/IgDiscover/	Allele discovery	Germline inference
IMPre	https://github.com/zhangwei2015/IMPre	Gene segment prediction	Germline inference
ARGalaxy	https://bioinf-galaxian.erasmusmc.nl/argalaxy/	Antibody analysis within the Galaxy platform	GUI
ARResT/Interrogate	https://github.com/infspiredBAT/ARResT.Interrogate	Interactive IG/TR analysis, currently down	GUI
ClonoCalc & ClonoPlot	https://bitbucket.org/ClonoSuite/clonocalc-plot	GUI front end for MiXCR with visualization tools	GUI
IGGalaxy	http://bioinformatics.erasmusmc.nl/wiki/index.php/Immunoglobulin_Galaxy	Galaxy service for detection and quantification, uses IMGT and IgBLAST for alignment	GUI
VDJServer	https://vdjserver.org/	Web-based repertoire analysis and storage	GUI
Vidjil	http://www.vidjil.org/	Interactive analysis of HTS via web-based interface with IMGT/V-QUEST, IgBLAST or MiXCR processing	GUI
Migmap	https://github.com/mikesh/migmap	IgBLAST wrapper	Other
VDJMLpy	https://vdjserver.org/vdjml/	File format for aligned sequence data	Other
AbMining Toolbox	https://sourceforge.net/projects/abmining/	Toolkit focusing on CDR3 analysis of antibody libraries	Pipeline
Cleanalyst	http://www.bu.edu/computationalimmunology/research/software/	Bayesian based approaches to gene annotation and unmutated ancestor inference	Pipeline
IgSCUEAL	https://github.com/spond/IgSCUEAL	Pipeline for B cell transcript analysis based on genetic algorithm for viral sub-typing	Pipeline
Immcantation	http://immcantation.readthedocs.io/	Pipeline, uses either IMGT or IgBLAST for align, many features including allelic inference	Pipeline
ImmuneDB	http://immunedb.com/	Storage and analysis of B- and T-cell sequence data, pipeline with custom alignment algorithm	Pipeline

ImmunediveRsity	https://bitbucket.org/ImmunediveRsity/immunediversity	Manipulation and processing of HTS reads to identify VDJ usage and clonal origin, IgBLAST	Pipeline
IMonitor	https://github.com/zhangwei2015/IMonitor	TCR/BCR pipeline using BLAST for alignment	Pipeline
IMSEQ	http://www.imtools.org/	Pipeline focused on clonotype inference	Pipeline
LymAnalyzer	https://sourceforge.net/projects/lymanalyzer	Allele discovery, fast-tag-search algorithm for alignment of datasets	Pipeline
MIXCR	https://mllaboratory.com/software/mixcr/	Pipeline for BCR and TCR analysis with focus on clonotype building	Pipeline
RTCR	http://uubram.github.io/RTCR/	TCR analysis pipeline	Pipeline
SONAR	https://github.com/scharch/SONAR	Pipeline for analysis of repertoire datasets	Pipeline
VDJfast	https://sourceforge.net/projects/vdjfasta/	BCR analysis tools including partitioning	Pipeline
clonotypeR	http://clonotyper.branchable.com	R package for the identification and analysis of clonotypes	Post-analysis
IMEX	http://bioinformatics.fh-hagenberg.at/immunexplorer/	Post-processing of IMGT/HighV-QUEST output	Post-analysis
LymphoSeq	https://bioconductor.org/packages/release/bioc/html/LymphoSeq.html	Post-analysis of Adaptive Biotechnologies ImmunoSEQ output	Post-analysis
tcR	http://imminfo.github.io/tcr/	Post processing of MiXCR, ImmunoSEQ or MiGEC	Post-analysis
TRIGS	https://github.com/williamdrees/TRIGS	Clonal lineage analysis of processed datasets	Post-analysis
VDJtools	https://github.com/mikessh/vdjtools	Toolkit for the post-analysis of datasets	Post-analysis
VDJviz	https://github.com/antigenomics/vdjviz	Visualization and analysis of datasets	Post-analysis
IgRepertoireConstructor	http://yana-safonova.github.io/ig_repertoire_constructor/	Antibody repertoire construction for MS/MS applications	Simulation
IgSimulator	http://yana-safonova.github.io/ig_simulator/	Simulation of antibody repertoires	Simulation
repenHMM	https://bitbucket.org/yuvalel/repenhmm	HMM for generating synthetic rearrangements	Simulation
sciReptor	https://github.com/b-cell-immunology/sciReptor	Single cell immunoglobulin repertoire analysis	Single cell
TraCeR	https://github.com/teichlab/tracer	Analysis of TCRs from single cell transcriptomes	Single cell
VDJPuzzle	https://bitbucket.org/kirbyvisp/vdjpuze2	TCR and BCR reconstruction from scRNA-seq data	Single cell

Note: The 'type' column provides a general classification for the tool; 'analysis' tools focus on performing targeted analysis on processed datasets; 'annotation' tools focus on the partitioning of rearranged sequences into their components; 'germline inference' tools attempt to define the genotypes from rearranged datasets; 'GUIs' provide easy-to-use interfaces for tools or pipelines; 'pipelines' are end-to-end solutions starting from raw data; 'post-analysis' tools offer general post-processing of annotated datasets, and 'single-cell' are tools for the reconstruction of genes from single cell datasets.

allow for somatic point mutations within the ends by placing predefined maximum values on the number of mutations and by excluding the possibility of consecutive mutations (Yousfi Monod *et al.*, 2004)).

The location of the IGHD amongst the N-additions means that care must be taken to delineate N additions from germline gene contributions. JOINSOLVER considers the likelihood that N-addition may give rise to sequences that mimic germline IGHDs (Souto-Carneiro *et al.*, 2004) by defining the maximum number of consecutive matches that differentiate between a true germline IGHD gene and N-additions. The minimum number consecutive matches were determined using a Monte Carlo simulation whereby large simulated datasets representative of N additions are used to calculate the frequency at which IGHD sequence motifs occur to establish probabilities. For appropriate probabilities to be calculated the simulated N additions should reflect real N-additions and models for simulation of N additions have been proposed (Jackson *et al.*, 2007).

Partitioning approaches also consider the problem of immunoglobulin partitioning in the context of the biological rearrangement process. Rather than a series of local alignments against reference sets of germline genes, models of rearrangement are created, and mathematical techniques are used to find the combination of germline genes, mutation and nucleotide loss and addition that best explain a rearranged sequence in the context of the model. SoDA for example utilized a 3D alignment algorithm for simultaneous alignments of all genes segments and to account for nucleotide loss and addition (Volpe *et al.*, 2006). JointML uses a maximum likelihood method to determine the best fit to a model of IGH rearrangement that considers all possible combinations of germline genes and N and P additions (Ohm-Laursen *et al.*, 2006), and iHMMune-align uses the veterbi algorithm to find the best path through a hidden markov model (HMM) which consists of a series of probabilistically defined states whose relationship to each other are described by transitions that are themselves also associated with probabilities (Gaeta *et al.*, 2007). These utilities were developed in the pre-deep sequencing era and generally do not scale well to very large repertoire datasets.

Contemporary tools that use model-based approaches include partis (Ralph and Matsen, 2016a), IGoR (Marcou *et al.*, 2017) and IgSCUEAL (Frost *et al.*, 2015). The partis package uses a novel HMM factorization strategy across large datasets to build a parameter rich model for partitioning of individual sequences into their germline gene and N addition contributions. It overcomes the limitations of previous HMM-based approaches by using flexible categorical distributions rather than probability distributions, implementing a new HMM compiler (ham), and by performing inferences on collections of HMMs rather than a single all-encompassing HMM (Ralph and Matsen, 2016a). IgSCUEAL attempts to annotate IGHV and IGHJ genes within rearrangements using a phylogenetic-based approach that combines a genetic algorithm with maximum likelihood fitting of evolutionary models, but IGHD identification is performed using pair-wise alignment (Frost *et al.*, 2015). IGoR uses a Sparse Expectation-Maximization algorithm to learn statistics of rearrangements for use in annotating rearranged genes in a probabilistic manner and can also generate simulated repertoire datasets based on the learned statistics (Marcou *et al.*, 2017).

Gene Reference Datasets and Allelic Inference

The major reference set utilized in the analysis of rearranged immunoglobulin genes is the IMGT reference set (Lefranc and Lefranc, 2001). Alternative collections of human immunoglobulin gene segments include NCBI (Ye *et al.*, 2013) and VBASE2 (Retter *et al.*, 2005). As increasingly diverse human populations are studied it has become apparent that existing reference datasets do not capture all polymorphisms that exist in the human population (Watson *et al.*, 2013; Boyd *et al.*, 2010; Kidd *et al.*, 2012; Watson *et al.*, 2015; Wang *et al.*, 2011; Scheepers *et al.*, 2015).

Several tools for the inference of novel gene segment polymorphisms within repertoire datasets have been developed including IgDiscover (Corcoran *et al.*, 2016), TIGGER (Gadala-Maria *et al.*, 2015), LymAnalyzer (Yu *et al.*, 2016) and IMPre (Zhang *et al.*, 2016). These tools leverage the ability of rearranged immunoglobulin genes to be used to infer set of available gene segments within the IGH locus of an individual (Boyd *et al.*, 2010). IgDiscover applies an iterative clustering, consensus building and filtering algorithm to IgM antibody libraries to define dataset specific germline gene segment sets *de novo* (Corcoran *et al.*, 2016). TIGGER utilizes datasets that have previously been aligned against a reference set, such as the IMGT reference set (Lefranc and Lefranc, 2001), and applies regression-based methods to detect distinct patterns of SHM to distinguish polymorphic positions from positions targeted by SHM (Gadala-Maria *et al.*, 2015). LymAnalyzer is a toolkit that aligns to the IMGT reference set using a fast-tag-search alignment and applies a two-step heuristic approach, adapted from (Boyd *et al.*, 2010), to the detection of novel alleles in the aligned data, testing for enrichment of a mismatch to the reference in multiple distinct rearrangements and requiring the difference to be present in at least 10% of the alignments to a reference gene (Yu *et al.*, 2016). IMPre utilizes a custom algorithm combining k-mer seed clustering, multiway tree-based assembly and optimization by filtering, merging and error correction, for *de novo* inference of germline gene segments from rearranged repertoire datasets (Zhang *et al.*, 2016).

Analysis suggests that parallel use haplotype inference, as originally proposed by (Kidd *et al.*, 2012), in conjunction with novel allele inference, provides for high sensitivity and specificity in repertoire analysis (Kirik *et al.*, 2017). The use of dataset specific reference sets does however restrict the options with respect to partitioning tools to those, such as IgBLAST (Ye *et al.*, 2013), that permit the user defined datasets, or to the application of re-assignment methods such as within TIGGER (Gadala-Maria *et al.*, 2015). Attempts to define additional allelic variants using other data sources have also been undertaken. For example, AlleleMiner (Yu *et al.*, 2017) attempts to recover immunoglobulin gene segments alleles from genomic sequence data by mining genomic variant calls from the short-read sequencing datasets such as the 1000 Genomes Project data. There are many challenges in building reference datasets from this type of single nucleotide polymorphism (SNP) data and caution must be exercised especially with respect to the mapping of the SNP data to the reference prior to inference (Watson *et al.*, 2017).

Analysis of Clonal Lineages

Once a dataset of high quality annotated gene rearrangements has been gathered it is desirable to understand the lineage structure within the dataset, or across datasets in the case of longitudinal or multiple subset or multiple site sampling from a single subject. Historically, this has been approached as a sequence clustering problem aimed at grouping highly similar immunoglobulin gene rearrangements together under the assumption that their similarity has resulted from their sharing the same initial B cell. This often involves clustering the CDR3 sequences of rearranged IGH that share the same IGHV and IGHJ segments and have equal CDR3 lengths at an identity threshold that allows for some SHM within the CDR3 but which shouldn't bring together CDR3s that were generated by distinct VDJ recombination events.

Several software packages implement sequence-based clustering for clone grouping, for example, IMSEQ's clone building using shared CDR3 sequences (Kuchenbecker *et al.*, 2015) and the clone clustering within the ImmuneDB (Rosenfeld *et al.*, 2017). The clone clustering implemented by ImmuneDB, which iterates over unique reads in descending size order, grouping smaller copy reads with higher count reads where they differ by less than 15% at the CDR3 amino acid level, was recently applied to B cells sampled from a number of different anatomical sites to develop an atlas of B-cell clonal distribution (Meng *et al.*, 2017). An attempt to improve on the results of sequence-based clustering approached is offered by the likelihood-based clonal inference using multi-HMMs implemented as part of partis (Ralph and Matsen, 2016b).

Detection of Antigen Selection

Antigen selection refers to testing whether or not the mutational spectra of immunoglobulin genes have been influenced by enrichment of mutations predicted to improve interaction with antigen. One package for testing if mutational distributions have been skewed by antigen selection is BASELINE (Yaari *et al.*, 2012). BASELINE provides a framework for quantifying the strength of antigen selection on individual sequences or within clonal lineages using Bayesian estimates of replacement frequency and has the capacity to aggregate probability density functions over multiple sequences in order to compare sets of sequences.

Additional approaches for studying antigen selection have been described, but not implemented as tools. Approaches are largely based on detecting an enrichment of mutations that have led to amino acid substitutions within the CDRs, and the avoidance of such changes in the FRs, and differ in their hypothesis testing statistics. The first of such methods was proposed by (Shlomchik *et al.*, 1987) and tested replacement (amino acid change) and silent (no amino acid change) in CDRs and FRs against a random model of mutation using the binomial distribution. This was later modified by (Chang and Casali, 1994) in an attempt to better account for the differences in codon usage between CDRs and FRs as CDRs are more intrinsically biased toward replacement mutations owing to their codon usage and (Lossos *et al.*, 2000) suggested the use of a multinomial, rather than binomial, model to determine significance. The failure to account for mutational hotspots and biases in mutational outcomes was suggested to lead to inaccuracies in these approaches (Bose and Sinha, 2005). This led to the proposed method by (Dahlke *et al.*, 2006) which extended prior literature to model hotspots using trinucleotides mutability scoring based on observed CDR and FR sequences to compare the accumulation of replacement mutations in the CDR in the context of the overall level of IGHV mutation.

Comparing Immunoglobulin Repertoires

The ability to compare repertoires, either from different samples from a single individual, or between different individuals generally relies upon comparing various metrics calculated after repertoire partitioning and annotation of clonal relationships and SHM spectra. For example, the frequency at which different gene segments are utilized, the frequency of somatic hypermutation, the isotype subclass utilization or characteristics and distributions of the clonal expansions. Frameworks for statistical comparisons of overall repertoires have been historically lacking. Two recently published frameworks to compare repertoires are the repertoire dissimilarity index (RDI) (Bolen *et al.*, 2017) and the MaxSnippetModel (Ostmeyer *et al.*, 2017). RDI is a non-parametric method to quantitatively compare repertoires by average variance in gene segment utilization through a combination of bootstrapped sub-sampling and simulation (Bolen *et al.*, 2017). In comparison, the MaxSnippetModel focusses on determining differences in CDR3 sub-motif composition between repertoires from different disease contexts by attempting to statistically classify sequences by representing CDR3 AA 'snippets' (overlapping sub-strings of fixed length) sequences as vector-based representation of AA properties using Atchely factors (Ostmeyer *et al.*, 2017). A TensorFlow implementation is used to score the snippet Atchely factors using logistic modelling to define the maximum score for each sample before model fitting using gradient optimization for classification.

Analysis Pipelines for Repertoire Data Analysis

A variety of immunoglobulin repertoire datasets are available from public repositories. A sampling of public immunoglobulin repertoire sequence datasets is provided in Table 2. There is no single approach to the analysis of repertoire data. The approach taken will often depend upon the source of the data sequenced, the sequencing technology used, the computational skills of the user, and the research question they are seeking to. Many toolkits that implement end-to-end pipelines for analysis are now available with extensive use-case documentation, for example MiXCR (see Relevant Websites section), Immcantation (see Relevant Websites section) or ImmuneDB (see Relevant Websites section).

Table 2 Examples of publically available human immunoglobulin repertoire datasets

Accession	Accession type	Technology	Description
PRJNA324093	BioProject	Illumina MiSeq	Longitudinal vaccine response with different cell subsets
PRJNA337970	BioProject	Illumina MiSeq	B cell subsets from lifelong malaria exposures
PRJNA330659	BioProject	Illumina HiSeq	Pre- and post-treatment samples for childhood B-ALL cohort
PRJNA309577	BioProject	Illumina HiSeq	5' RACE derived BCR and TCR repertoire
PRJNA291102	BioProject	Illumina HiSeq	Comparison of methodologies for repertoire amplifications
PRJNA277373	BioProject	Illumina MiSeq	Amplification from total PBMCs of a healthy donor
PRJNA260985	BioProject	Illumina MiSeq	IGH sequencing from B cells and plasmablasts in a variety of conditions including autoimmunity
PRJNA260905	BioProject	Illumina HiSeq	Immunosuppression in context of transplantation

In the interest of highlighting the general components of repertoire data analysis pipelines an example of the basic processing steps from raw Illumina MiSeq paired end sequenced library of cDNA amplicons generated from primers positioned in the IGHV and IGHC to post-processed data using stand-alone tools would be as follows:

- **Pre-processing:** Merging of read pairs (eg. PEAR (Zhang *et al.*, 2014), de-multiplexing of barcodes and extraction of UMIs (eg. UMI-tools (Smith *et al.*, 2017)) and read trimming to remove primers and adaptors and quality filtering (eg. trimmomatic (Bolger *et al.*, 2014))
- **Partitioning:** Partitioning of merged read pairs using stand-alone instance of IgBLAST (Ye *et al.*, 2013) using either the IMGT reference sets of IGHV, IGHD and IGHJ segment or customized germline datasets.
- **Isotype calling:** Using either script-based string matching (eg. perl or python) or BLAST alignment (Camacho *et al.*, 2009) to determine which isotype subclass is associated with each VDJ based on the expected CH1 exon sequence governed by the IGHC primer location.
- **Parsing:** Scripting to parse key features from the IgBLAST output such as gene segment calls, mutations, N/P nucleotides and CDR/FRs and to integrate the isotype and other metadata.
- **Clone building:** CDR3 sequence subset based on the subject, IGHV, IGHJ and CDR3 length may be clustered using tools such as cd-hit (Fu *et al.*, 2012) or uclust (Edgar, 2010) to infer clonally related sequences within an individual response. Similar clustering methods can be used across responses from different subject to define convergent IGH signatures.
- **Post-processing:** Custom databases may be used for storage to serve as a basis for analysis or custom file formats generated. Scripting languages can be used to develop analysis to explore the features of interest within the dataset.

Current Challenges and Conclusions

Like any field in rapid change, the immune receptor repertoire sequencing field has lacked standardization as it has moved into the era of deep sequencing, however, this gap is beginning to be addressed (Yaari and Kleinstein, 2015; Breden *et al.*, 2017). Challenges exist around the development and implementation of frameworks for the standardization of data submission, data formats for tool input and output and in the incorporation of new data types into traditional resources.

The integration of repertoire sequencing datasets with other 'omics data derived from the same subject is becoming increasingly popular. For example, annotation of deeply sequenced repertoire data for antigen specificity through the use of monoclonal antibodies (Jackson *et al.*, 2014), or matching of transcript datasets to proteomic (Adamson *et al.*, 2017). Tools that provide researchers with a unified analysis of their various data types will become increasingly important as these gaps are currently filled by custom, in-house pipelines that link often disparate tools and toolkits to undertake analysis. Leveraging deep sequenced dataset for deep and machine learning applications is also expected to become increasingly informative as the variety, number and depth of datasets continues to increase.

See also: Computational Immunogenetics. Extraction of Immune Epitope Information. Immunoglobulin Clonotype and Ontogeny Inference. Immunoinformatics Databases. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition. Vaccine Target Discovery

References

- Adamson, P.J., Al Kindi, M.A., Wang, J.J., *et al.*, 2017. Proteomic analysis of influenza haemagglutinin-specific antibodies following vaccination reveals convergent immunoglobulin variable region signatures. *Vaccine* 35, 5576–5580.
- Alt, F.W., Baltimore, D., 1982. Joining of immunoglobulin heavy chain gene segments: Implications from a chromosome with evidence of three D–JH fusions. *Proc Natl Acad Sci USA* 79, 4118–4122.
- Bannard, O., Cyster, J.G., 2017. Germinal centers: Programmed for affinity maturation and antibody diversification. *Curr Opin Immunol* 45, 21–30.
- Benedict, C.L., Gilfillan, S., Thai, T.H., Kearney, J.F., 2000. Terminal deoxynucleotidyl transferase and repertoire development. *Immunol Rev* 175, 150–157.

- Bernard, O., Hozumi, N., Tonegawa, S., 1978. Sequences of mouse immunoglobulin light chain genes before and after somatic changes. *Cell* 15, 1133–1144.
- Bolen, C.R., Rubelt, F., van der Heiden, J.A., Davis, M.M., 2017. The Repertoire Dissimilarity Index as a method to compare lymphocyte receptor repertoires. *BMC Bioinformatics* 18, 155.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30, 2114–2120.
- Bolotin, D.A., Poslavsky, S., Mitrophanov, I., *et al.*, 2015. MiXCR: Software for comprehensive adaptive immunity profiling. *Nat Methods* 12, 380–381.
- Bose, B., Sinha, S., 2005. Problems in using statistical analysis of replacement and silent mutations in antibody genes for determining antigen-driven affinity selection. *Immunology* 116, 172–183.
- Boyd, S.D., Gaeta, B.A., Jackson, K.J., *et al.*, 2010. Individual variation in the germline Ig gene repertoire inferred from variable region gene rearrangements. *J Immunol* 184, 6986–6992.
- Boyd, S.D., Marshall, E.L., Merker, J.D., *et al.*, 2009. Measurement and clinical monitoring of human lymphocyte clonality by massively parallel VDJ pyrosequencing. *Sci Transl Med* 1, 12ra23.
- Breden, F., Luning Prak, E.T., Peters, B., *et al.*, 2017. Perspective: Reproducibility and reuse of adaptive immune receptor repertoire data. *Front Immunol*.
- Brochet, X., Lefranc, M.P., Giudicelli, V., 2008. IMGT/V-QUEST: The highly customized and integrated system for IG and TR standardized V–J and V–D–J sequence analysis. *Nucleic Acids Res* 36, W503–W508.
- Busse, C.E., Czogiel, I., Braun, P., Arndt, P.F., Wardemann, H., 2014. Single-cell based high-throughput sequencing of full-length immunoglobulin heavy and light chain genes. *Eur J Immunol* 44, 597–603.
- Camacho, C., Coulouris, G., Avagyan, V., *et al.*, 2009. BLAST+: Architecture and applications. *BMC Bioinform* 10, 421.
- Chang, B., Casali, P., 1994. The CDR1 sequences of a major proportion of human germline Ig VH genes are inherently susceptible to amino acid replacement. *Immunol Today* 15, 367–373.
- Chothia, C., Lesk, A.M., 1987. Canonical structures for the hypervariable regions of immunoglobulins. *J Mol Biol* 196, 901–917.
- Corcoran, M.M., Phad, G.E., Vazquez Bernat, N., *et al.*, 2016. Production of individualized V gene databases reveals high levels of immunoglobulin genetic diversity. *Nat Commun* 7, 13642.
- Croce, C.M., Shander, M., Martinis, J., *et al.*, 1979. Chromosomal location of the genes for human immunoglobulin heavy chains. *Proc Natl Acad Sci USA* 76, 3416–3419.
- Dahlke, I., Nott, D.J., Ruhno, J., Sewell, W.A., Collins, A.M., 2006. Antigen selection in the IgE response of allergic and nonallergic individuals. *J Allergy Clin Immunol* 117, 1477–1483.
- Davis, M.M., 1990. T cell receptor gene diversity and selection. *Annu Rev Biochem* 59, 475–496.
- Edgar, R.C., 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26, 2460–2461.
- Eisen, H.N., 2014. Affinity enhancement of antibodies: How low-affinity antibodies produced early in immune responses are followed by high-affinity antibodies later and in memory B-cell responses. *Cancer Immunol Res* 2, 381–392.
- Ellebedy, A.H., Jackson, K.J., Kissick, H.T., *et al.*, 2016. Defining antigen-specific plasmablast and memory B cell subsets in human blood after viral infection or vaccination. *Nat Immunol* 17, 1226–1234.
- Ezkurdia, I., Juan, D., Rodriguez, J.M., *et al.*, 2014. Multiple evidence strands suggest that there may be as few as 19,000 human protein-coding genes. *Hum Mol Genet* 23, 5866–5878.
- Faham, M., Zheng, J., Moorhead, M., *et al.*, 2012. Deep-sequencing approach for minimal residual disease detection in acute lymphoblastic leukemia. *Blood* 120, 5173–5180.
- Frost, S.D., Murrell, B., Hossain, A.S., Silverman, G.J., Pond, S.L., 2015. Assigning and visualizing germline genes in antibody repertoires. *Philos Trans R Soc Lond B Biol Sci* 370.
- Fu, L., Niu, B., Zhu, Z., Wu, S., Li, W., 2012. CD-HIT: Accelerated for clustering the next-generation sequencing data. *Bioinformatics* 28, 3150–3152.
- Gadala-Maria, D., Yaari, G., Uduman, M., Kleinstein, S.H., 2015. Automated analysis of high-throughput B-cell sequencing data reveals a high frequency of novel immunoglobulin V gene segment alleles. *Proc Natl Acad Sci USA* 112, E862–E870.
- Gaeta, B.A., Malmgren, H.R., Jackson, K.J., *et al.*, 2007. iHMMune-align: Hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. *Bioinformatics* 23, 1580–1587.
- Galson, J.D., Clutterbuck, E.A., Truck, J., *et al.*, 2015. BCR repertoire sequencing: Different patterns of B-cell activation after two Meningococcal vaccines. *Immunol Cell Biol* 93, 885–895.
- Glanville, J., Zhai, W., Berka, J., *et al.*, 2009. Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *Proc Natl Acad Sci USA* 106, 20216–20221.
- Hoh, R.A., Joshi, S.A., Liu, Y., *et al.*, 2016. Single B-cell deconvolution of peanut-specific antibody responses in allergic patients. *J Allergy Clin Immunol* 137, 157–167.
- Jackson, K.J., Gaeta, B.A., Collins, A.M., 2007. Identifying highly mutated IGHD genes in the junctions of rearranged human immunoglobulin heavy chain genes. *J Immunol Methods* 324, 26–37.
- Jackson, K.J., Liu, Y., Roskin, K.M., *et al.*, 2014. Human responses to influenza vaccination show seroconversion signatures and convergent antibody rearrangements. *Cell Host Microbe* 16, 105–114.
- Jolly, C.J., Wagner, S.D., Rada, C., *et al.*, 1996. The targeting of somatic hypermutation. *Semin Immunol* 8, 159–168.
- Kidd, M.J., Chen, Z., Wang, Y., *et al.*, 2012. The inference of phased haplotypes for the immunoglobulin H chain V region gene loci by analysis of VDJ gene rearrangements. *J Immunol* 188, 1333–1340.
- Kidd, M.J., Jackson, K.J., Boyd, S.D., Collins, A.M., 2016. DJ pairing during VDJ recombination shows positional biases that vary among individuals with differing IGHD locus immunogenotypes. *J Immunol* 196, 1158–1164.
- Kirik, U., Greiff, L., Levander, F., Ohlin, M., 2017. Parallel antibody germline gene and haplotype analyses support the validity of immunoglobulin germline gene inference and discovery. *Mol Immunol* 87, 12–22.
- Kuchenbecker, L., Nienen, M., Hecht, J., *et al.*, 2015. IMSEQ – A fast and error aware approach to immunogenetic sequence analysis. *Bioinformatics* 31, 2963–2971.
- Lafaille, J.J., Decloux, A., Bonneville, M., Takagaki, Y., Tonegawa, S., 1989. Junctional sequences of T cell receptor gamma delta genes: Implications for gamma delta T cell lineages and for a novel intermediate of V-(D)-J joining. *Cell* 59, 859–870.
- Lefranc, M.-P., Lefranc, G., 2001. *The Immunoglobulin Factsbook*. San Diego: Academic Press.
- Levin, M., King, J.J., Glanville, J., *et al.*, 2016. Persistence and evolution of allergen-specific IgE repertoires during subcutaneous specific immunotherapy. *J Allergy Clin Immunol* 137, 1535–1544.
- Liao, H.X., Lynch, R., Zhou, T., *et al.*, 2013. Co-evolution of a broadly neutralizing HIV-1 antibody and founder virus. *Nature* 496, 469–476.
- Lieber, M.R., 1999. The biochemistry and biological significance of nonhomologous DNA end joining: An essential repair process in multicellular eukaryotes. *Genes Cells* 4, 77–85.
- Lossos, I.S., Tibshirani, R., Narasimhan, B., Levy, R., 2000. The inference of antigen selection on Ig genes. *J Immunol* 165, 5122–5126.
- Ma, Y., Pannicke, U., Schwarz, K., Lieber, M.R., 2002. Hairpin opening and overhang processing by an Artemis/DNA-dependent protein kinase complex in nonhomologous end joining and V(D)J recombination. *Cell* 108, 781–794.
- Marcou, Q., Mora, T. & Walczak, A.M. 2017. IGoR: A Tool For High-Throughput Immune Repertoire Analysis. *bioRxiv*.
- McBlane, J.F., van Gent, D.C., Ramsden, D.A., *et al.*, 1995. Cleavage at a V(D)J recombination signal requires only RAG1 and RAG2 proteins and occurs in two steps. *Cell* 83, 387–395.
- McBride, O.W., Hieter, P.A., Hollis, G.F., *et al.*, 1982. Chromosomal location of human kappa and lambda immunoglobulin light chain constant region genes. *J Exp Med* 155, 1480–1490.
- Meng, W., Zhang, B., Schwartz, G.W., *et al.*, 2017. An atlas of B-cell clonal distribution in the human body. *Nat Biotechnol* 35, 879–884.

- Muramatsu, M., Kinoshita, K., Fagarasan, S., *et al.*, 2000. Class switch recombination and hypermutation require activation-induced cytidine deaminase (AID), a potential RNA editing enzyme. *Cell* 102, 553–563.
- Ohm-Laursen, L., Nielsen, M., Larsen, S.R., Barington, T., 2006. No evidence for the use of DIR, D-D fusions, chromosome 15 open reading frames or VH replacement in the peripheral repertoire was found on application of an improved algorithm, JointML, to 6329 human immunoglobulin H rearrangements. *Immunology* 119, 265–277.
- Ostmeyer, J., Christley, S., Rounds, W.H., *et al.*, 2017. Statistical classifiers for diagnosing disease from immune repertoires: A case study using multiple sclerosis. *BMC Bioinformatics* 18, 401.
- Parameswaran, P., Liu, Y., Roskin, K.M., *et al.*, 2013. Convergent antibody signatures in human dengue. *Cell Host Microbe* 13, 691–700.
- Rada, C., Milstein, C., 2001. The intrinsic hypermutability of antibody heavy and light chain genes decays exponentially. *EMBO J* 20, 4570–4576.
- Ralph, D.K., Matsen, F.A.T., 2016a. Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. *PLOS Comput Biol* 12, e1004409.
- Ralph, D.K., Matsen, F.A.T., 2016b. Likelihood-based inference of B cell clonal families. *PLOS Comput Biol* 12, e1005086.
- Retter, I., Althaus, H.H., Munch, R., Muller, W., 2005. VBASE2, an integrative V gene database. *Nucleic Acids Res* 33, D671–D674.
- Rizzetto, S., Koppstein, D.N., Samir, J., *et al.*, 2017. B-cell receptor reconstruction from single-cell RNA-seq with VDJPuzzle. *bioRxiv*.
- Rosenfeld, A.M., Meng, W., Luning Prak, E.T., Hershberg, U., 2017. ImmuneDB: A system for the analysis and exploration of high-throughput adaptive immune receptor sequencing data. *Bioinformatics* 33, 292–293.
- Roskin, K.M., Simchoni, N., Liu, Y., *et al.*, 2015. IgH sequences in common variable immune deficiency reveal altered B cell development and selection. *Sci Transl Med* 7, 302ra135.
- Russ, D.E., Ho, K.Y., Longo, N.S., 2015. HTJoinSolver: Human immunoglobulin VDJ partitioning using approximate dynamic programming constrained by conserved motifs. *BMC Bioinform* 16, 170.
- Scheepers, C., Shrestha, R.K., Lambson, B.E., *et al.*, 2015. Ability to develop broadly neutralizing HIV-1 antibodies is not restricted by the germline Ig gene repertoire. *J Immunol* 194, 4371–4378.
- Schroeder Jr., H.W., Cavacini, L., 2010. Structure and function of immunoglobulins. *J Allergy Clin Immunol* 125, S41–S52.
- Shlomchik, M.J., Aucoin, A.H., Pisetsky, D.S., Weigert, M.G., 1987. Structure and function of anti-DNA autoantibodies derived from a single autoimmune mouse. *Proc Natl Acad Sci USA* 84, 9150–9154.
- Smith, T., Heger, A., Sudbery, I., 2017. UMI-tools: Modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome Res* 27, 491–499.
- Souto-Carneiro, M.M., Longo, N.S., Russ, D.E., Sun, H.W., Lipsky, P.E., 2004. Characterization of the human Ig heavy chain antigen binding complementarity determining region 3 using a newly developed software algorithm, JOINSOLVER. *J Immunol* 172, 6790–6802.
- Souto-Carneiro, M.M., Sims, G.P., Girschik, H., Lee, J., Lipsky, P.E., 2005. Developmental changes in the human heavy chain CDR3. *J Immunol* 175, 7425–7436.
- Ta, V.T., Nagaoka, H., Catalan, N., *et al.*, 2003. AID mutant analyses indicate requirement for class-switch-specific cofactors. *Nat Immunol* 4, 843–848.
- Teng, G., Papavasiliou, F.N., 2007. Immunoglobulin somatic hypermutation. *Annu Rev Genet* 41, 107–120.
- Tonegawa, S., 1983. Somatic generation of antibody diversity. *Nature* 302, 575–581.
- Truck, J., Ramasamy, M.N., Galson, J.D., *et al.*, 2015. Identification of antigen-specific B cell receptor sequences using public repertoire analysis. *J Immunol* 194, 252–261.
- Turchaninova, M.A., Davydov, A., Britanova, O.V., *et al.*, 2016. High-quality full-length immunoglobulin profiling with unique molecular barcoding. *Nat Protoc* 11, 1599–1616.
- van Gent, D.C., McBlane, J.F., Ramsden, D.A., *et al.*, 1995. Initiation of V(D)J recombination in a cell-free system. *Cell* 81, 925–934.
- Volpe, J.M., Cowell, L.G., Kepler, T.B., 2006. SoDA: Implementation of a 3D alignment algorithm for inference of antigen receptor recombinations. *Bioinformatics* 22, 438–444.
- Volpe, J.M., Kepler, T.B., 2008. Large-scale analysis of human heavy chain V(D)J recombination patterns. *Immunome Res* 4, 3.
- Wang, C., Liu, Y., Cavanagh, M.M., *et al.*, 2015. B-cell repertoire responses to varicella-zoster vaccination in human identical twins. *Proc Natl Acad Sci USA* 112, 500–505.
- Wang, Y., Jackson, K.J., Davies, J., *et al.*, 2014. IgE-associated IGHV genes from venom and peanut allergic individuals lack mutational evidence of antigen selection. *PLOS One* 9, e89730.
- Wang, Y., Jackson, K.J., Gaeta, B., *et al.*, 2011. Genomic screening by 454 pyrosequencing identifies a new human IGHV gene and sixteen other new IGHV allelic variants. *Immunogenetics* 63, 259–265.
- Watson, C.T., Matsen, F.A.T., Jackson, K.J.L., *et al.*, 2017. Comment on "A database of human immune receptor alleles recovered from population sequencing data". *J Immunol* 198, 3371–3373.
- Watson, C.T., Steinberg, K.M., Graves, T.A., *et al.*, 2015. Sequencing of the human IG light chain loci from a hydantidiform mole BAC library reveals locus-specific signatures of genetic diversity. *Genes Immunol* 16, 24–34.
- Watson, C.T., Steinberg, K.M., Huddleston, J., *et al.*, 2013. Complete haplotype sequence of the human immunoglobulin heavy-chain variable, diversity, and joining genes and characterization of allelic and copy-number variation. *Am J Hum Genet* 92, 530–546.
- Weigert, M.G., Cesari, I.M., Yonkovich, S.J., Cohn, M., 1970. Variability in the lambda light chain sequences of mouse antibody. *Nature* 228, 1045–1047.
- Wu, X., Zhou, T., Zhu, J., *et al.*, 2011. Focused evolution of HIV-1 neutralizing antibodies revealed by structures and deep sequencing. *Science* 333, 1593–1602.
- Yaari, G., Kleinstein, S.H., 2015. Practical guidelines for B-cell receptor repertoire sequencing analysis. *Genome Med* 7, 121.
- Yaari, G., Uduman, M., Kleinstein, S.H., 2012. Quantifying selection in high-throughput Immunoglobulin sequencing data sets. *Nucleic Acids Res* 40, e134.
- Ye, J., Ma, N., Madden, T.L., Ostell, J.M., 2013. IgBLAST: An immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res* 41, W34–W40.
- Yousfi Monod, M., Giudicelli, V., Chaume, D., Lefranc, M.P., 2004. IMGT/Junction Analysis: The first tool for the analysis of the immunoglobulin and T cell receptor complex V–J and V–D–J Junctions. *Bioinformatics* 20 (Suppl 1), i379–i385.
- Yu, Y., Ceredig, R., Seoighe, C., 2016. LymAnalyzer: A tool for comprehensive analysis of next generation sequencing data of T cell receptors and immunoglobulins. *Nucleic Acids Res* 44, e31.
- Yu, Y., Ceredig, R., Seoighe, C., 2017. A database of human immune receptor alleles recovered from population sequencing data. *J Immunol* 198, 2202–2210.
- Zhang, J., Kobert, K., Flouri, T., Stamatakis, A., 2014. PEAR: A fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*, 30, 614–620.
- Zhang, W., Wang, I.M., Wang, C., *et al.*, 2016. IMPre: An accurate and efficient software for prediction of T- and B-cell receptor germline genes and alleles from rearranged repertoire data. *Front Immunol* 7, 457.

Relevant Websites

<http://immcantation.readthedocs.io/>
Immccantation.

<http://immunedb.com/>
ImmuneDB.

<http://mixcr.readthedocs.io/>
MiXCR.

Biographical Sketch

Dr Katherine J.L. Jackson: Dr Jackson is currently a Senior Research Officer at the Garvan Institute of Medical Research in Sydney, Australia. She completed her graduate studies at the University of New South Wales in Sydney, Australia on the detection of antigen selection signatures in immunoglobulin sequence datasets and has completed postdoctoral training at both the University of New South Wales and Stanford University, CA, USA. Her research applies immune repertoire sequencing methodologies to the study a variety of human immune responses including primary and secondary responses to vaccination and infection, allergy and autoimmunity, while also using exploring the immunogenetics of both humans and mice.

Epitope Predictions

Roman Kogay, Nazarbayev University, Astana, Republic of Kazakhstan

Christian Schönbach, Nazarbayev University, Astana, Kazakhstan

© 2019 Elsevier Inc. All rights reserved.

Abbreviations

ACO Ant colony optimization
ANN Artificial neural network
A_{ROC} Area under receiver operating characteristic curve

MCC Matthews correlation coefficient
SVM Support vector machine
SVR Support vector regression
WEKA Waikato environment for knowledge analysis

Introduction

Hallmarks of the adaptive immune system are discrimination of self from non-self recognized as antigenic determinants or epitopes through concerted activation of antibody and cell-mediated primary or memory responses. A complex network of molecular and cellular interactions that includes antigen presenting cells, T cell and B cells regulates the steps from antigen processing, epitope presentation and recognition to memory recall (Litman *et al.*, 2010; Frank, 2002). One key question in epitope prediction is whether a predicted epitope will be immunogenic. Indeed theoretical estimates for MHC class I restricted T cell epitopes indicate that only 1 in 2000 peptides of non-self antigens may induce a dominant CTL response (Yewdell and Bennink, 1999).

T Cell Epitopes

Antigen presenting cells process proteins into peptides that if recognized by T cells are called T cell epitopes. Two distinct pathways facilitate the processing of exogenous and endogenous (self and foreign) proteins into peptides which were comprehensively reviewed by Blum *et al.* (2013). Most peptides generated by proteolysis through 26S proteasome are transported by TAP (transporter associated with antigen processing) into the endoplasmic reticulum where they bind to MHC class I molecules. If the affinity of the peptides to MHC class I molecules is sufficiently high, stable peptide-MHC-I complexes are transported through the Golgi apparatus to the cell surface where they are recognized by T-cell receptors (TCR) of CD8⁺ T cells (Fig. 1). In contrast, MHC class II molecules usually bind in the endosome to peptides derived from lysosomal proteolysis of exogeneous proteins trafficked by phago- and endocytosis. Peptide-MHC II complexes are then transported in endosomal vesicles to the cell surface where they are recognized by TCR expressed on CD4⁺ T cells (Fig. 2). Although the binding of the peptides to MHC is crucial in

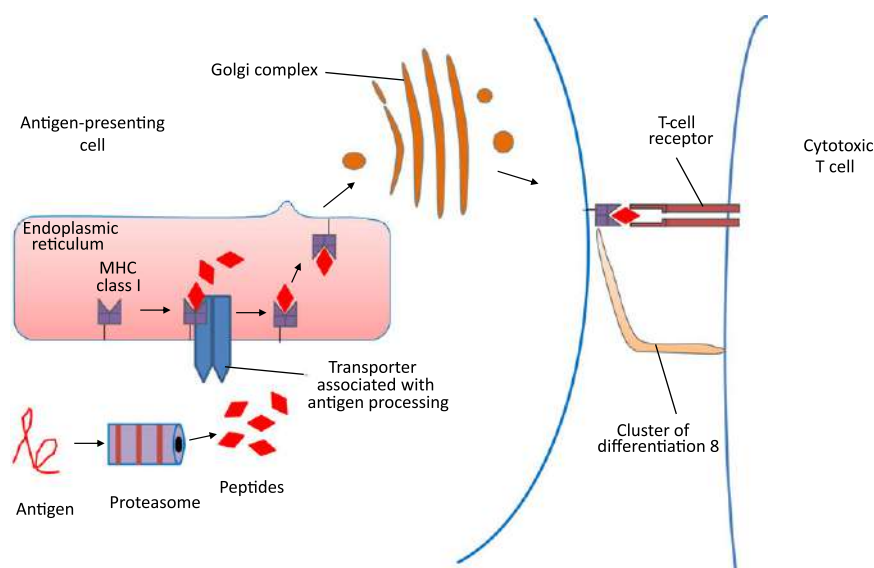


Fig. 1 Processing, presentation and recognition of MHC class I-restricted T cell epitopes. Modified from Fig 17.21 (a) in Karp, G., 2008. Cell and Molecular Biology. Concepts and Experiments, fifth ed. Asia: John Wiley & Sons (Asia) Pte. Ltd.

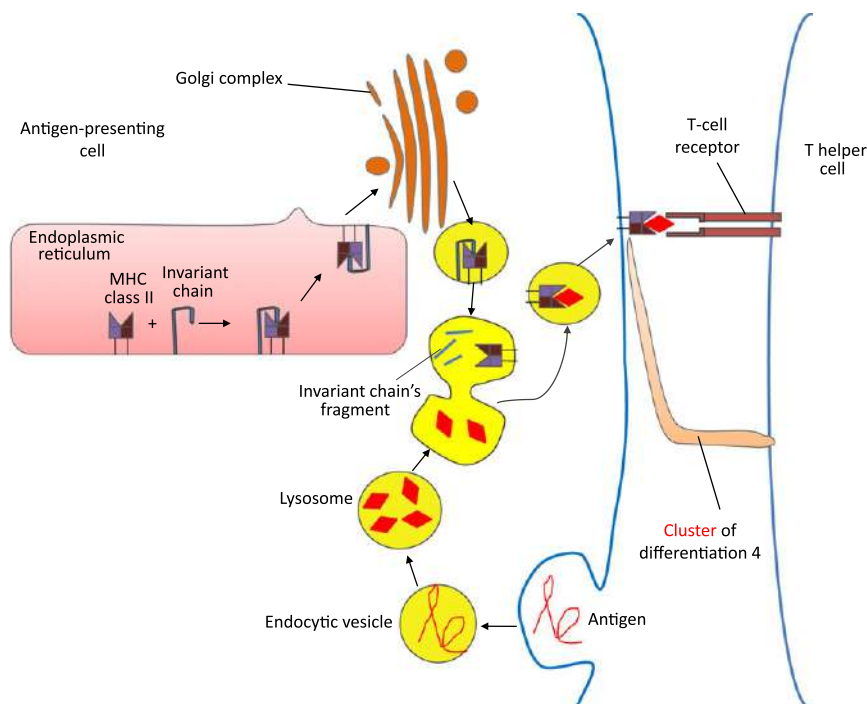


Fig. 2 Processing, presentation and recognition of MHC class II restricted T cell epitopes. Modified from Fig 17.21 (a) in Karp, G., 2008. Cell and Molecular Biology. Concepts and Experiments, fifth ed. Asia: John Wiley & Sons (Asia) Pte. Ltd.

defining whether peptides may become epitopes, cross-presentation of peptides generated from phagocytosed exogenous proteins by MHC class I or endogenous proteins processed in the lysosome by MHC class II complicates matters in epitope prediction-assisted vaccine design. Cross-priming of naïve $CD8^+$ T cells mediated by cross-presentation is just one example where the choice of epitopes in a vaccine will affect primary and secondary T-cell responses (Grotzke *et al.*, 2017). Since epitope-based vaccines are meant to mimic the natural protective immunity that activates the functions of both T and B cells (Fig. 3) we need to consider also B cell epitopes.

B Cell Epitopes

B cell epitopes do not require processing and are predominantly of conformational or discontinuous nature (Barlow *et al.*, 1986; Greenbaum *et al.*, 2007). The B cell epitope is a specific 3D surface area of an antigen that is recognized by the paratope, the antigen-binding part of the antibody (Sela-Culang *et al.*, 2013). Antibodies represent the secreted form of B-cell receptors (hence B cell epitopes) which are produced by mature B cells, the plasma cells (Fig. 3). The antibodies produced by plasma cells are the result of a somatic hypermutation process of the complementary determining regions (CDRs) that increases the antibody affinity to the antigen (affinity maturation) (Schroeder and Cavacini, 2010; Neu and Wilson, 2016). Unlike T cell epitope predictions the conformational nature of the B cell epitopes and conformational diversity of antibodies themselves, in addition to conformational changes induced in both antibodies and epitopes when they bind to each other, pose an additional challenge in identifying immunogenic candidate B cell epitopes. Unsurprisingly, advances in predicting B cell epitopes are less pronounced than for T cell epitopes although the problem of conformational epitopes has been known since 1986 when the first crystal structure of a lysozyme-antibody complex was published (Barlow *et al.*, 1986).

Background

Immunogenicity of Epitopes

In both T and B cell epitope predictions of pathogen-derived proteins the goal is to identify potential immunogenic epitopes that induce a protective immune response. In case of T cell epitopes, peptides that are predicted to bind with high affinity (good binders) to MHC molecules are indeed very often immunogenic. Yet the MHC binding affinity remains a surrogate that does not provide clues about the type of T cell response an epitope may trigger. Co-stimulatory signals mediated by expression of various members of the cluster of differentiation (CD) family on both antigen presenting cells and T cells will determine which signaling pathway downstream of the TCR-peptide-MHC complex are activated, and hence whether the epitope triggers an effector T cell response or a suppressive, regulatory T cell response. Thus future assessments of predicted epitopes towards immunogenicity,

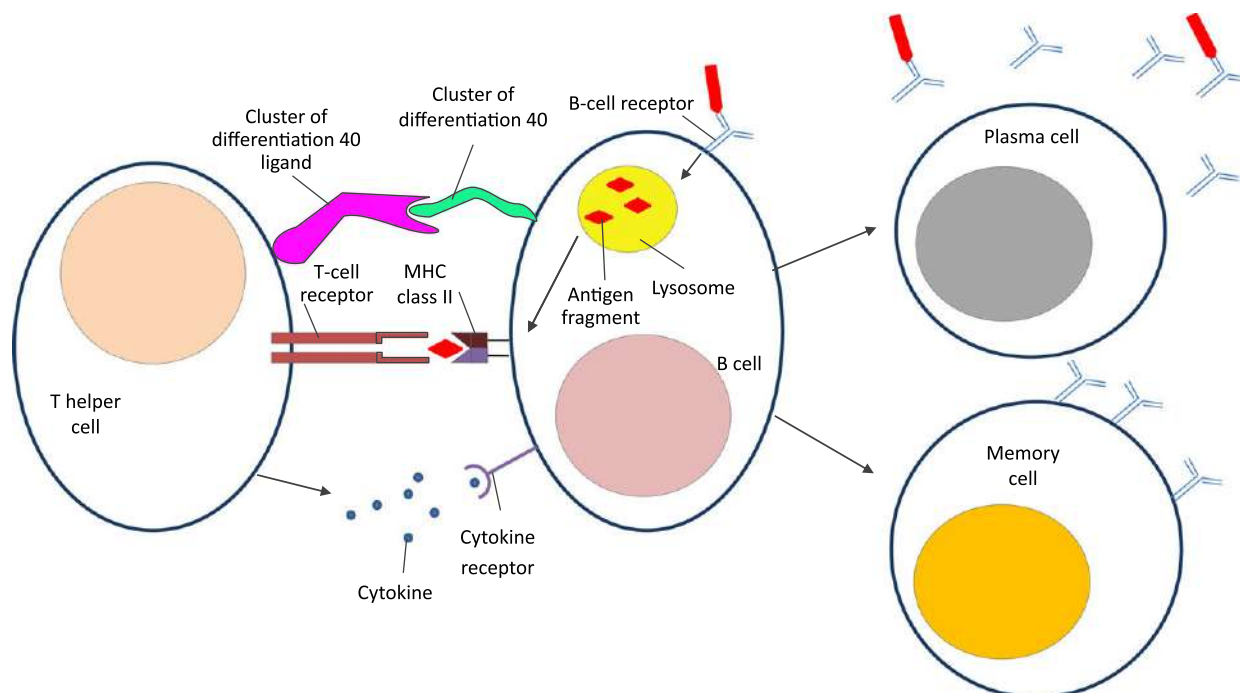


Fig. 3 Interaction of activated T helper cells activates B cells which differentiate into memory and plasma cells. The latter secrete antigen-specific antibodies. Modified from Fig 17.10 (a) in Karp, G., 2008. *Cell and Molecular Biology. Concepts and Experiments*, fifth ed. Asia: John Wiley & Sons (Asia) Pte. Ltd.

particularly in context of human vaccine development, should take into account the data derived from systems biology approaches (Querec *et al.*, 2009) and biomarker assays (Rappuoli and Aderem, 2011) in the context, dynamics and quality of a protective immune response to further boost the efficacy of data-driven predictions.

Overview of Epitope Predictions

Epitope predictions utilize statistical, machine learning or structural models derived from experimental data as reviewed by Soria-Guerra *et al.* (2015). Early epitope prediction methods BIMAS (Parker *et al.*, 1994) and EpiMatrix (Schafer *et al.*, 1998) were based on quantitative matrices (QM) or binding motifs (SYFPEITHI) (Rammensee *et al.*, 1999) that capture simple linear relationships of binding and non-binding data. Both methods are limited by the number of available experimental binding and non-binding data for a specific MHC allotype to derive a binding motif based on amino acids observed at peptide residue positions or to score a peptide's binding potential using position-dependent coefficients of amino acids. The limitations of quantitative matrices including overfitting, are balanced in practice by their easy implementation and use combined with an acceptable accuracy for selected MHC allotypes (Schönbach *et al.*, 2002; De Groot *et al.*, 2002).

The continuous growth of data enabled the application of more sophisticated methods utilizing artificial neural networks (ANN) (Adams and Koziol, 1995), hidden Markov models (HMM) ranging from fully connected HMMs to profile HMMs (Mamitsuka, 1998; Brusic *et al.*, 2002; Zhang *et al.*, 2006; Larsen *et al.*, 2006), and support vector machines (SVM) (Zhang *et al.*, 2007; Jacob and Vert, 2007; Chen *et al.*, 2007) that are adept in extracting and processing complex nonlinear relationships in peptide-MHC binding or B cell epitope data to derive predictive models. The predictive power of these models largely depends on the quantity and quality of (unbiased) annotated data. Immune Epitope Database (IEDB) (Vita *et al.*, 2014), a manually curated database of experimentally characterized immune epitopes has been instrumental in providing since 2005 high-quality data for training and testing datasets. Successful application of these data-driven methods that reduced experimental time and costs spurred further improvements towards increasing the accuracy of CTL epitope predictions by combining different individual prediction-based models followed by integrating top-predicting scores to derive a consensus score (consensus methods) (Moutafsi *et al.*, 2006). Similarly, the integration of predictions for distinct processes such as proteasomal processing, TAP binding and peptide-MHC binding while decreasing the dependency on MHC allotype-specific binding data as implemented in NetCTL-pan (Stranzl *et al.*, 2010) improved the predictive power.

Structure-based methods (Patronov and Doytchinova, 2013) yield high-quality quantitative predictions of peptide-MHC binding affinities based on free Gibbs energy calculations that take into account the interactions and distances between atoms of peptide and MHC amino acid residues. Threading of peptides predicted to bind to MHC is based on the computationally intensive free energy calculations of individual interactions (Logean and Rognan, 2002) or pairwise interactions using an energy potential

matrix (Schueler-Furman *et al.*, 2000). Quantitative structure-activity relationship (QSAR) methods that are employed in drug discovery, assuming similar molecular structures result in similar activities, analyze differences in free energy and structures (Doytchinova and Flower, 2002). The limited number of available high quality structures restricts the applicability of structure-based methods particularly for B cell epitope predictions.

The performance of prediction methods is evaluated by applying Pearson correlation, Matthews correlation coefficients or receiver operating characteristic (ROC) analyses which includes positive-predictive value (PPV), negative-predictive value (NPV), area under the receiver operating characteristic curve (A_{ROC}), sensitivity (SE), specificity (SP) or root mean squared deviations (RMSD) of distances in Å for structure predictions (Yang and Yu, 2009; Hattotuwigama *et al.*, 2007). SE provides the ratio of correctly predicted real positives, whereas SP for example, correctly predicted true negatives indicate the quality of predictions. PPV and NPV provide an overall success rate as proportions of true positives (negatives) of all positive (negative) predicted, whereas the A_{ROC} value reflects the overall quality of predictions within SE and SP value ranges. The A_{ROC} value is a robust measure as long as the number of positives and negatives in a test set are not extremely lop-sided. When no independent test data sets are available leave-one-out cross-validation (LOOC) (Sammut and Webb, 2010) is often used. LOOC may indicate an acceptable performance of a tool that becomes unacceptable when applied to a new data set.

Approaches

T Cell Epitope Prediction

T cell epitope predictions can be divided into integrated, MHC-pan and allotype-specific binding, TAP binding, proteasomal cleavage, and a few miscellaneous approaches which are represented by 38 publicly available, mostly web accessible tools shown in Table 1. Each tool is summarized by its name, URL, applicable species, and salient features including limitations, performance and methods applied to provide a convenient desk reference for choosing a T cell epitope prediction tool.

Independent of the approach categories the majority of tools rely on the application of QM, ANN, SVM, consensus or hybrid methods. An ANN is a network of interconnected nodes whose connections carry numeric data that are trained by weighting the connections (Honeyman *et al.*, 1998). Prior to using an ANN for peptide sequences the sequences must be transformed to numeric descriptors to input into the network layers. Popular ANN-based prediction tools include MULTIPRED2 (Zhang *et al.*, 2011a), NetMHC 4.0 (Andreatta and Nielsen, 2015) and NetMHCpan 3.0 (Nielsen and Andreatta, 2016) for MHC class I and NetMHCII 2.2 (Nielsen and Lund, 2009) and NetMHCIIpan 3.1 (Andreatta *et al.*, 2015) for MHC class II peptide binding.

SVMs comprise machine learning algorithms that classify complex data by placing it into a multidimensional space and constructing a hyperplane through an appropriate kernel function. Representative tools using SVM are TAP binding predictor TAPreg (Diez-Rivero *et al.*, 2010), MHC class II peptide binding predictor MHC2SKpan (Guo *et al.*, 2013) and T cell epitope immunogenicity predictor Repitope (Ogishi and Yotsuyanagi, 2017).

Several MHC-pan (PSSMHCpan (Liu *et al.*, 2017) and TEPITOPEpan (Zhang *et al.*, 2012)) and allotype-specific tools (PickPocket 1.1 (Zhang *et al.*, 2009), HLaffy (Mukherjee *et al.*, 2016)) and PREDIVAC 2.0 (Oyarzún *et al.*, 2013) use QM-based methods that assign coefficients for each amino acid in a peptide frame and discriminate potential binders from non-binders through the derived peptide score. The only QM-based tool that successfully integrated proteasomal processing, TAP, MHC class I and II binding prediction is TEpredict (Antonets and Maksyutov, 2010) whereas ANN- and QM-based NetCTLpan 1.1 (Stranzl *et al.*, 2010) do not cover MHC class II.

Integrated approaches that are based on the consensus of independent methods tend to outperform single methods. NetMHCcons 1.1 which couples NetMHC 3.4, NetMHCpan 2.8 and PickPocket 1.1 methods is a leading example (Karosiene *et al.*, 2012).

The only MHC class II allele-specific candidate epitope prediction tools that utilize structure-based methods (Yao *et al.*, 2013) are EpiDOCK (Atanasova *et al.*, 2013) and EpiTOP (Dimitrov *et al.*, 2010a; Dimitrov *et al.*, 2010b). EpiDOCK uses docking score-based quantitative matrices to evaluate the binding potential of overlapping nonamer peptides generated from an input sequence to five HLA class II DPA/DBP, six DQA/DQB and 12 DRB1 proteins. EpiTOP employs QSAR-derived quantitative matrices that encode both peptide and HLA protein pocket interactions derived from 12 HLA-DRB1 allotypes to predict peptide-HLA-DRB1 binding.

Two tools, categorized under miscellaneous approaches use hybrid SVM and motif methods to identify potential T-helper cell epitopes by predicting cytokine-inducing peptides. IL4Pred (Dhanda *et al.*, 2013a) and IFNepitope (Dhanda *et al.*, 2013b) predict interferon gamma- and interleukin 4-inducing MHC class II peptides, respectively.

B Cell Epitope Prediction

The conformational and linear B cell epitope prediction approaches comprising 21 web accessible tools are summarized according to their characteristics including performance, limitations and methods in Table 2.

Representatives of conformational epitope prediction approaches using structure-based methods are Discotope 2.0 (Kringelum *et al.*, 2012) and BEpro (Sweredoski and Baldi, 2008b). Discotope 2.0 uses amino acid residue propensity scores within a defined sphere whereas BEPro applies an amino acid propensity scale in combination with side chain orientation and solvent accessibility properties.

Table 1 T cell epitope prediction approaches

Name and URL	Species	Functionality and Performance	Limitations	Method
<i>Integrated approaches</i>				
NetCTLpan 1.1 http://www.cbs.dtu.dk/services/NetCTLpan/	Hosa, Patr, Mamu, Susc, Mumu, Gogo, Bota	Integrates prediction of proteasomal cleavage, TAP and MHC class I binding affinity; A_{ROC} : 0.920–0.977 (depending on data); (Stranzl <i>et al.</i> , 2010).	Predictions for 8–11 mer; at most 5000 sequences per submission and each sequence must be <20,000 amino acids; maximum 20 MHC allotypes per submission.	ANN and matrix (matrix for TAP).
NetTepi 1.0 http://www.cbs.dtu.dk/services/NetTepi/	Hosa	Integrates peptide-MHC binding affinity, peptide-MHC stability and TCR propensity; prediction values are weighted sum or % rank; sorted by combined, affinity, stability, TCR propensity scores; $AUC_{0.1}$: 0.9285–0.9305 (depending on the combined model) (Trolle and Nielsen, 2014).	13 HLA allotypes; 8–14mer peptides; at most 5000 sequences per submission and each sequence must be <20,000 amino acids; T cell propensity tool was validated only on 9mer peptides.	ANN and matrix.
NetMHCcons 1.1 http://www.cbs.dtu.dk/services/NetMHCcons/	Hosa, Patr, Mamu, Susc, Mumu, Gogo, Bota	Integrates methods of NetMHC, NetMHCpan and Pickpocket; predictions for 8–15mer peptides; prediction values in nM IC_{50} and % Rank; sorting by predicted binding affinity; PCC: 0.23–0.72 (depending on the group of allotypes) (Karosiene <i>et al.</i> , 2012).	101 MHC I allotypes; at most 5000 sequences per submission and each sequence must be <20,000 amino acids; maximum 20 MHC allotypes per submission.	Consensus method: ANN, pan-specific ANN, and matrix.
RANKPEP http://imed.med.ucm.es/Tools/rankpep.html	Hosa, Mumu	Cleavage prediction; immunodominance filter; molecular weight filter; predictions can be made from MSA; > 80% of CD8 ⁺ T cell epitopes are among top 2% of scoring peptides; 3–10% threshold was required to predict 80% of CD4 ⁺ T cells epitopes; (Reche <i>et al.</i> , 2004).	102 MHC I and 80 MHC II allotypes; predictions for MHC I binder are limited 8–11mer peptide; matrices are limited by the quality of sequences.	Matrix and SVM (Immuno-dominance).
NetCTL 1.2 http://www.cbs.dtu.dk/services/NetCTL/	Hosa	Integrates HLA class I binding, C terminal cleavage, TAP transport efficiency; (depending on the threshold); A_{ROC} : 0.941, sensitivity: 72% (among 5% top-scoring peptide) (Larsen <i>et al.</i> , 2007).	12 HLA supertypes; predicts only 9mer epitopes.	ANN (HLA binding and C terminal cleavage) and matrix (TAP transport efficiency).

ProPred1 http://crrdd.osdd.net/raghava/propred1/	Hosa; Mumu	Prediction of the standard proteasome and immunoproteasome cleavage sites; identification of MHC binders with cleavage sites at C terminus; output displayed in different formats; allows subsequence analysis; accuracy: 38–80% for HLA-A*0201, 70–80% for H2-Kb (depends on the threshold); (Singh and Raghava, 2003).	47 MHC I allotypes; only 9mer peptides are predicted; over-generalized – the matrices were obtained for enolase-1 protein.	Matrix.
nHLAPred http://crrdd.osdd.net/raghava/nhlapred/	Hosa, Mumu	Average accuracy for hybrid approach: 92.8%; prediction of the standard proteasome and immunoproteasome cleavage sites; identification of MHC binders with cleavage sites at C terminus; (Bhasin and Raghava, 2007).	67 MHC I allotypes; only nonamer peptides.	ANN and matrix or matrix only.
MULTIPRED 2.0 http://cvc.dfci.harvard.edu/multipred2/	Hosa	Allows input of pre-calculated viral proteomes; employs NetMHCpan and NetMHCIIpan as predictive engines; heatmaps are generated to visualize results; (Zhang <i>et al.</i> , 2011).	26 HLA class I and II supertypes; 8–11 mer peptides for HLA I and 9mer peptides for HLA II.	ANN.
TepiTool (Pipeline) http://tools.iecb.org/tepitool/	Hosa, Patr, Mamu, Susc, Mumu, Gogo, Bota and others	Integrates several predictive tools; predicts MHC I and MHC II binders; allow to select most frequent allotypes; different types of filtering, such as percentile rank, absolute rank, based on IC ₅₀ (Paul <i>et al.</i> , 2016).	Predicts 8–14mer epitopes.	Includes but not limited to consensus, ANN, and matrix methods.
MHCPred 2.0 http://www.ddg-pharmfac.net/mhcpred/MHCPred/	Hosa, Mumu	Two models of prediction: amino acids contribution; amino acids + their interactions; anchor positions (maximum 4) can be chosen; predicts affinity to TAP; (Guan <i>et al.</i> , 2006).	17 MHC class I and II allotypes; sequences are limited to 1000 residues; only plain format is supported.	Matrix.
FRED 2.0 (Pipeline) http://fred-2.github.io/	N/A	HLA typing, epitope prediction, epitope selection, and epitope assembly; implemented in Python; provides unified access to different epitope prediction tools and databases; (Schubert <i>et al.</i> , 2016).	Difficult to execute for unfamiliar users; requires separate installation of several external tools from Center for Biological Sequence Analysis, Technical University of Denmark.	Python-based framework.
TEpredict http://tepredict.sourceforge.net/downloads.html	N/A	Predicts MHC I and MHC II binders, proteasomal and immunoproteasomal processing, peptide and TAP binding;	Considers only 9mer peptides.	Matrix.

(Continued)

Table 1 Continued

Name and URL	Species	Functionality and Performance	Limitations	Method
SVMHC (works only via EpiToolKit pipeline) https://abi.inf.uni-tuebingen.de/Services/SVMHC	Hosa, Mumu	sensitivity: 50–80%; specificity: 75–99%; (Antonets and Maksyutov, 2010). Allows accessions/database identifiers as input; analysis of single amino acid polymorphism; different output views; (Donnes and Kohlbaicher, 2006).	Predictions for 8–10 mer peptides; 26 MHC I allotypes from MHCPEP, 24 MHC I allotypes from SYFPEITHI and 51 MHC II allotypes; only one sequence per submission; thresholds cannot be set by users.	SVM (MHC I) and matrices (MHC II).
PickPocket 1.1 http://www.cbs.dtu.dk/services/PickPocket/	Hosa, Patr, Mamu, Susc, Murmu, Gogo, Bota	Robust when data is scarce and when the similarity to MHC molecules with characterized binding specificities is low; PCC: 0.26–0.6 (depending on the evaluated set) (Zhang <i>et al.</i> , 2009).	8–12mer peptides; > 150 MHC I and II allotypes; at most 5000 sequences per submission and each sequence must be < 20,000 amino acids; maximum 20 MHC per submission.	Matrix.
MetaMHCpan http://datamining-iiip.fudan.edu.cn/MetaMHCpan/index.php/pages/view/info	Hosa, Mumu	Meta-server with different predictive methods (Xu <i>et al.</i> , 2016).	8–11mer peptides for MHC I and 9–25mer peptides for MHC II; 41 MHC I and > 600 MHC II allotypes.	Matrix, SVM, and multiple instance learning method.
<i>MHC I pan approaches</i> NetMHCpan 3.0 http://www.cbs.dtu.dk/services/NetMHCpan/	Hosa, Patr, Mamu, Susc, Murmu, Gogo, Bota	91% of ligands are recovered at a rank threshold of 2% with a specificity of 98%; A _{ROC} : 0.83–0.89; (depending on the epitope length); (Nielsen and Andreatta, 2016).	172 MHC I molecules; 8–14mer peptides; at most 5000 sequences per submission and each sequence must be < 20,000 amino acids; maximum 20 MHC allotypes per submission.	ANN.
PSSMHCpan https://github.com/BGI2016/PSSMHCpan	Hosa	A _{ROC} : 0.94, accuracy: 85%; able to predict neoantigens; 8–25 mer peptide length prediction; (Liu <i>et al.</i> , 2017).	87 HLA class I allotypes.	Matrix.
<i>MHC I allotype-specific approaches</i> NetMHC 4.0 http://www.cbs.dtu.dk/services/NetMHC/	Hosa, Patr, Mamu, Susc, Murmu, Bota	A _{ROC} : 0.882–0.895; (depending on the peptide epitope length); (Andreatta and Nielsen, 2015).	122 MHC I allotypes; 8–13 mer peptides; at most 5000 sequences per submission and each sequence must be < 20,000 amino acids;	ANN.

HLaffy http://proline.biochem.iisc.ernet.in/HLaffy/?tab=1	Hosa	Estimates peptide affinity for HLA class I; accuracy: 92% and correlation: 0.85, for IEDB dataset accuracy: 82.5%; provides a histogram view; representative molecular models with favorable peptide-HLA residue interactions are available (Mukherjee et al., 2016).	maximum 20 MHC allotypes per submission. Only for 9mer peptides.	Matrix.
<i>MHC II pan approaches</i> NetMHCIIpan 3.1 http://www.cbs.dtu.dk/services/NetMHCIIpan/	Hosa, Mumu	Prediction values are given as IC ₅₀ (inhibitory concentration) and as % ranks; A _{ROC} : 0.80–0.90 for the binding between peptide and MHC; (Andreatta et al., 2015).	4 MHC II isotypes; at most 5000 sequences per submission and each sequence must be <20,000 amino acids; maximum 20 MHC per submission.	ANN.
TEPITOPEpan http://datamining-iiip.fudan.edu.cn/service/TEPITOPEpan/TEPITOPEpan.html	Hosa	Shows binding cores and percentile ranks; prediction for 9–25mer peptides; average A _{ROC} : 0.717–0.833 (depending on the dataset); predicts over 700 HLA-DR allotypes; (Zhang et al., 2012).	Limited to HLA-DR.	Matrix.
MHC2SKpan http://datamining-iiip.fudan.edu.cn/service/MHC2SKpan/info.html	Hosa	Predicts MHC class II peptide binding; A _{ROC} : 0.734–0.843 (depending on the dataset); (Guo et al., 2013).	9–25mer peptides; DRB only.	SVM.
PREDIVAC – 2.0 http://predivac.biosci.uq.edu.au/	Hosa	A _{ROC} : 0.842–0.872 for HLA II binding; A _{ROC} : 0.749 for CD4-T-cell epitopes; 883 HLA II allotypes; target population epitope predictions (Oyarzún et al., 2013).	DRB only.	Matrix.
<i>MHC II allotype-specific approaches</i> NetMHCII 2.2 http://www.cbs.dtu.dk/services/NetMHCII/	Hosa, Mumu	Predictions are given as % Rank and in nM IC ₅₀ values; A _{ROC} : 0.68–0.82 (depending on the dataset and method) (Nielsen and Lund, 2009).	28 MHC II allotypes; at most 5000 sequences per submission, each sequence must be <20,000 amino acids.	ANN.
EpiDOCK http://epidock.ddg-pharmfac.net/	Hosa	Identifies 90% true binders and 76% true non-binders; overall accuracy: 83% (Atanasova et al., 2013).	23 HLA II allotypes; only 9mer peptides are predicted; no sorting mechanism.	Matrix.
EpiTOP http://www.pharmfac.net/EpiTOP/	Hosa		12 HLA-DRB1 allotypes; only 9mer peptides are predicted;	Matrix.

(Continued)

Table 1 Continued

Name and URL	Species	Functionality and Performance	Limitations	Method
MHC2Pred http://crdd.osdd.net/raghava/mhc2pred/	Hosa, Mumu	Identifies 89% of known epitopes within top 20% of predicted binders (Dimitrov <i>et al.</i> , 2010a,b). Accuracy: > 78%; (Lata <i>et al.</i> , 2007).	allotype selection did not work at the time of testing. 42 MHC II allotypes; only 9mer peptides are predicted; inconvenient output format without sorting options.	SVM.
Propred http://crdd.osdd.net/raghava/propred/	Hosa	Output displayed in different formats; (Singh and Raghava, 2001).	51 HLA-DR allotypes.	Matrix.
HLA-DR4Pred http://crdd.osdd.net/raghava/hladr4pred/	Hosa	Accuracy: 86% and 78% for SVM and ANN, respectively. (Bhasin and Raghava, 2004b)	Only for HLA-DRB1*0401.	SVM and ANN.
<i>Proteasomal cleavage approaches</i>				
Pcleavage http://crdd.osdd.net/raghava/pcleavage/	N/A	Predicts proteasome cleavage sites; MCCs: 0.54 and 0.43 for <i>in vitro</i> and MHC ligand data respectively (Bhasin and Raghava, 2005).	The C-terminal of MHC ligands represents only a subset of cleavages that occur <i>in vivo</i> .	SVM.
NetChop 3.1 http://www.cbs.dtu.dk/services/NetChop/	Hosa	Predicts cleavage sites of human proteasome; two different methods: C-terminus and 20S; specificity: 48%, sensitivity: 81% MCC: 0.31 (Nielsen <i>et al.</i> , 2005).	At most 100 sequences and 100,000 amino acids per submission.	ANN.
<i>TAP binding approaches</i>				
TAPPred http://crdd.osdd.net/raghava/tappred/	Hosa	Predicts TAP-binding peptides; correlation coefficients: 0.88 (Cascade SVM) and 0.8 (SVM); cascade SVM uses features of amino acids along with sequence whereas SVM uses only sequence (Bhasin <i>et al.</i> , 2007).	N/A	SVM or Cascade SVM.
TAPreg http://imed.med.ucm.es/Tools/tapreg/	N/A	Predicts affinity of TAP binding ligands; maximum PCC: 0.89 ± 0.03 (Diez-Rivero <i>et al.</i> , 2010).	Data training was done on 9mer peptides only.	SVM.
<i>Miscellaneous T cell epitope related prediction approaches</i>				
Repitope https://github.com/masato-ogishi/Repitope	Hosa	Epitope immunogenicity prediction; accuracy: 70–80% (Ogishi and Yotsuyanagi, 2017).	Retrospective observational study; simplified assumptions biophysical-chemical nature of the TCR-peptide-MHC	SVM.

IL4pred http://crdd.osdd.net/raghava/il4pred/	N/A	Predicts interleukin 4-inducing MHC class II binders; maximum accuracy: 75.76% and MCC: 0.51 (hybrid method) (Dhanda <i>et al.</i> , 2013a).	interactions; limited to TCR-V beta. Training set consists of only 8–22mer peptides created without species considerations.	Motif, SVM or hybrid.
IFNepitope http://crdd.osdd.net/raghava/ifnepitope/	N/A	Prediction and design of interferon- γ inducing MHC class II binding peptides; maximum prediction accuracy of hybrid approach: 81.3% (MCC: 0.57) – 82.1% (MCC: 0.62) (depends on dataset) (Dhanda <i>et al.</i> , 2013b).	Analysis of positions-specific preference of residues was done only for a small number of peptides.	Motif, SVM or hybrid.
CTLPred http://crdd.osdd.net/raghava/ctlpred/	N/A	Predicts CTL epitopes; accuracy for QM, ANN and SVM: 70.0, 72.2% and 75.2% respectively; has combined and consensus prediction approaches; (Bhasin and Raghava, 2004a).	Training data contains only 9mer epitopes; small blind dataset (63 epitopes for 2 subgroups).	ANN, SVM and matrix.
MMBPred http://crdd.osdd.net/raghava/mmbpred/	Hosa, Mamu, Mumu	Prediction of mutated promiscuous binders e.g. increase/decrease of peptide-MHC binding; analyzes position and type of mutations (Bhasin and Raghava, 2003).	67 MHC I allotypes; only 9mer peptides are predicted; promiscuous binding prediction was not supported at the time of testing.	Matrix.

Abbreviations: ANN: artificial neural network, A_{ROC} or AUC: area under receiver operating characteristic curve, Bola: *Bos taurus*, Gogo: Gorilla gorilla, Hosa: *Homo sapiens*, HLA: human leukocyte antigen, Mamu: *Macaca mulatta*, MCC: Matthews correlation coefficient, MHC: major histocompatibility complex, MSA: multiple sequence alignment, Mumu: *Mus musculus*, Patr: *Pan troglodytes*, PCC: Pearson correlation coefficient, SVM: support vector machine, Susc: *Sus scrofa*, TAP: transporter associated with antigen processing, and QM: quantitative matrix.

Table 2 B cell epitope prediction approaches

Name and URL	Functionality and Performance	Limitations	Method
<i>Discontinuous/conformational epitope prediction approaches</i>			
SEPPA 2.0 http://lifecenter.sgst.cn/seppa2/	Subcellular localization of an antigen and host species are considered; A_{ROC} : 0.785–0.823 (depending on the host and localization of antigen protein); Jmol is used for visualization (Qi <i>et al.</i> , 2014).	Antigens with less than <25 residues were not considered.	Optimized logistic regression algorithm.
SEPIa https://github.com/SEPIaTool/SEPIa	A_{ROC} : 0.65; prediction is based on the amino acid sequence; amino acid features and sequence-based features are taken into account (Dalkas and Rooman, 2017).	Only window of 9 residues is used to test whether the middle residue is an epitope or not.	Gaussian Naïve Bayes and Random Forest algorithms based on 13 features.
DiscoTope 2.0 http://www.cbs.dtu.dk/services/DiscoTope/	A_{ROC} : 0.824 and 0.727 (for training and independent data sets) (Kringelum <i>et al.</i> , 2012).	Trained on a dataset with the total number of residues per epitope 9–22 and with the longest sequential stretch 3–12 residues per epitope.	Definition of the spatial neighborhood to sum propensity scores and half-sphere exposure as a surface measure.
EPSVR http://sysbio.unl.edu/EPSVR/	A_{ROC} : 0.597; evaluates 6 different features: residue epitope propensity, conservation score, side chain energy score, contact number, surface planarity score and secondary structure composition (Liang <i>et al.</i> , 2010).	Multiple epitopes in one antigen were not considered.	SVR with six attributes.
EPMeta http://sysbio.unl.edu/EPMeta/	A_{ROC} : 0.638 (Liang <i>et al.</i> , 2010).	Works only on Linux OS.	Consensus method for EPSVR, EPCES, Epitopia, SEPPA, BEpro and Discotope 1.2.
BEpro http://pepito.proteomics.ics.uci.edu/index.html	A_{ROC} : 0.754 and 0.683 (for Discotope and Epitome datasets respectively); Jmol is used for visualization (Sweredoski and Baldi, 2008b).	Propensity scale scores averaged over a window of exactly 9 residues; only PDB file format is supported.	Linear combination of amino-acid propensity scale, side chain orientation and solvent accessibility information using half sphere exposure values.
CBTOPE http://crrdd.osdd.net/raghava/cbtope/	Accuracy: 86.59%, A_{ROC} : 0.90, MCC: 0.73; prediction is based on the amino acid sequence using different window length patterns, standard binary and physicochemical profiles of pattern (Ansari and Raghava, 2010).	Lacks 3D structural output.	SVM.
EPCES http://sysbio.unl.edu/EPCES/	Sensitivity: 47.8%, Specificity: 69.5%, A_{ROC} : 0.632; evaluates residue epitope propensity, conservation score, side chain energy score, contact number, surface planarity score and secondary structure composition (Liang <i>et al.</i> , 2009).	Low accuracy for the bound structures.	Consensus scoring utilizing six different scoring functions.

Bpredictor https://code.google.com/archive/p/my-project-bpredictor/download	A_{ROC} : 0.633 and 0.654 for bound and unbound datasets respectively; impact of interior residues and different contributions of adjacent residues were considered (Zhang <i>et al.</i> , 2011a,b,c).	Window size: 3–15 residues.	Random-forest algorithm with distance based feature.
EpiSearch http://curie.utmb.edu/episearch.html	In most cases covers >50% of experimentally validated residues; Jmol is used for visualization (Negi and Braun, 2009).	In addition to the 3D structure of the antigen the input requires the set of mimotopes (up to 12 mer).	Patch analysis that identifies cluster of residues on the surface of antigen with similar physicochemical properties as in mimotopes.
PEP – 3D-Search http://kyc.nenu.edu.cn/Pep3DSearch/	MCC: 0.176, sensitivity: 36.4%, Precision: 69.5% (Huang <i>et al.</i> , 2008).	In addition to the 3D structure of the antigen the input requires the set of mimotopes; compatible only with Windows OS.	Mimotope-based analysis via ACO algorithm.
CEP http://196.1.114.49/cgi-bin/cep.pl	Accuracy: 75%; Jmol is used for visualization (Kulkarni-Kale <i>et al.</i> , 2005).	Simplified approach with few attributes.	Algorithm uses accessibility of residues and spatial distance cut-off.
<i>Linear B cell epitope prediction approaches</i> LBtope http://crdd.osdd.net/raghava/lbtope/	Able to identify mutation(s) in peptide and convert it/ them to the epitope; accuracy: 81% (54–86% depending on model and dataset); mutation tool is available to design better epitope or for de-immunization purpose (Singh <i>et al.</i> , 2013).	5–30mer epitope prediction.	SVM, using diverse features; binary profile, di-peptide composition and amino acid pair profile.
COBEpro http://scratch.proteomics.ics.uci.edu/	A_{ROC} : 0.606–0.829 (depending on the dataset); Predicts epitopes of any length (Sweredoski and Baldi, 2008a).	Prediction is limited to sequences of 1500 amino acids length.	SVM for the similarity measure based on the total number of identical substrings.
ABCPred http://crdd.osdd.net/raghava/abcpred/	Sensitivity: 67.14%, specificity: 64.71%, accuracy: 65.93%; has overlapping filter (Saha and Raghava, 2006).	10, 14, 16, 18, 20mer predicted epitope length in plain text format; developed with a small dataset derived from database BCiPEP (Saha <i>et al.</i> , 2005)	ANN.
SVMTriP http://sysbio.unl.edu/SVMTriP/	Precision: 54.1–57.1%, sensitivity: 68.5–80.1%, A_{ROC} : 0.674–0.702 (depending on the length of epitope); prediction combines tri-peptide similarity and propensity scores (Yao <i>et al.</i> , 2012).	10, 14, 16, 18, 20mer predicted epitope length; only one sequence per time in FASTA format; waiting time at the time of testing was 10–30 min.	SVM.
IgPred http://crdd.osdd.net/raghava/igpred/	Predicts B-cell epitopes that can induce a specific class of antibody; MCC: 0.44 (IgG), 0.7 (IgE), 0.45 (IgA), accuracy is around 80%; epitope mapping and motif scan functions are available (Gupta <i>et al.</i> , 2013).	4–20 mer epitopes.	SVM and WEKA. package tools.
Bcepred http://crdd.osdd.net/raghava/bcepred/		Only plain text format is supported; developed with a small dataset derived	Evaluation of different physicochemical scales by different method for each.

(Continued)

Table 2 Continued

Name and URL	Functionality and Performance	Limitations	Method
<i>Linear and discontinuous/conformational epitope prediction approaches</i> Epitopia http://epitopia.tau.ac.il/	Prediction accuracy for various properties: 52.92–57.53%; highest accuracy via combination of properties: 58.70% (Saha and Raghava, 2004). A_{ROC}: 0.6 (conformational) and 0.59 (linear); visualization through Jmol and RasMol; calculates immunogenicity for each solvent accessible residue for 3D structure input and for every amino acid for sequence input (Rubinstein <i>et al.</i>, 2009).	from database BCIPEP (Saha <i>et al.</i>, 2005) Immunogenicity of each residue is determined by analysis of physicochemical properties of three flanking residues.	Naïve Bayes classifier.
EllipPro http://tools.iedb.org/ellipro/	A_{ROC}: 0.732, sensitivity: 60.1%; Jmol and MODELLER program are available for visualization and prediction 3D structure (Ponomarenko <i>et al.</i>, 2008).	Generalize all proteins as ellipsoids.	Modified Thornton's method with residue (pI value) clustering
BepiPred 2.0 http://www.cbs.dtu.dk/services/BepiPred/index.php	A_{ROC} for structural epitopes: 0.62, A_{ROC} for linear epitopes: 0.574 (Jespersen <i>et al.</i>, 2017).	5–25mer epitope predictions; at most 50 sequences (max. 300,000 residues per submission, < 6000 residues/sequence	Random Forest algorithm

Implementations of machine learning algorithms for conformational epitope predictions are Epitopia (Rubinstein *et al.*, 2009), EPSVR (Liang *et al.*, 2010) and Bpredictor (Zhang *et al.*, 2011c). These tools utilize naïve Bayesian classifiers, SVM and random forest algorithm, respectively. A consensus method (Liang *et al.*, 2009) was developed for EPMeta (Liang *et al.*, 2010). EPMeta combines EPSVR, EPCES, Epitopia, SEPPA, BEpro and Discotope 1.2 predictions to predict surface residues as an epitope if two or more single tools have voted for it. None of the methods applied increased the predictive performances which range between A_{ROC} of 0.597 and 0.824, depending on the datasets tested, above moderate levels.

Sequence-based methods rely on amino acid sequence-based features to evaluate the probability of each residue to be a part of a conformational epitope (Sun *et al.*, 2013). CBTOPE is a SVM-based tool that applies standard binary and physico-chemical profiles of patterns (Ansari and Raghava, 2010). SEPIa employs naïve Bayesian classifier and random forest ensemble methods to classify residues using 13 different features (Dalkas and Rooman, 2017). Although sequence-based methods can be a solution for conformational epitope predictions the performance improvement of complex methods as implemented in SEPIa is minor compared to simpler methods.

Mimotopes are peptides selected from random libraries for their ability to bind to an antibody directed against a specific antigen. Since the mimicry relies on similarities in physicochemical properties and spatial organization (Moreau *et al.*, 2006) mimotope analysis methods applied to conformational epitope predictions require both mimotopes and a 3D structure of the target antigen as an input. The mimotopes are mapped to the surface of the antigen to identify the best sequence alignment, and predict potential epitope regions (Sun *et al.*, 2016). For example PEP-3D-Search searches for matching paths on an antigen surface with respect to the query mimotopes using an ant colony optimization algorithm (Huang *et al.*, 2008).

Linear epitope prediction approaches utilize SVM, ANN and random forest machine learning methods to evaluate multiple amino acid residue properties (Potocnakova *et al.*, 2016). The machine learning approaches have substituted older, poorly performing methods that used only amino acid propensity scores (Blythe and Flower, 2005). The features assessed include for example hydrophilicity, flexibility, turns, solvent accessibility and amino acid pair antigenicity scale (Yasser and Honavar, 2010). SVMTriP (Yao *et al.*, 2012), LBtope (Singh *et al.*, 2013) and COBEpro (Sweredowski and Baldi, 2008a) are representatives of SVM-based tools. ABCpred (Saha and Raghava, 2006) and Bepipred 2.0 (Jespersen *et al.*, 2017) are ANN and random forest-based tools, respectively.

A few tools allow the prediction of both conformational and linear B cell epitopes. ElliPro (Ponomarenko *et al.*, 2008) is based on the implementation of three different algorithms that treat a protein as an ellipsoid shape (Taylor *et al.*, 1983), derive a residue protrusion index using a modified Thornton method (Thornton *et al.*, 1986), and cluster neighboring residues according to their pI (isoelectric point) values. When tested on a conformational epitope dataset constructed from antibody-protein complex 3D structures ElliPro's performance was moderate (A_{ROC} 0.732). Naïve Bayes classifier-based Epitopia (Rubinstein *et al.*, 2009) performs slightly inferior for conformational (A_{ROC} 0.60) and linear (A_{ROC} 0.59) epitopes. Similarly modest performances were reported for BepiPred 2.0 with A_{ROC} 0.62 for conformational and A_{ROC} 0.574 for linear epitopes (Jespersen *et al.*, 2017).

MHC and Epitope Databases

Sustained public accessibility of curated high-quality data on HLA allele sequences, experimentally derived epitope and non-epitope data has enabled the development and improvement of epitope prediction tools. IMGT[®] (the international ImmunoGeneTics information system[®]) established in 1989 became the global reference in immunogenetics and immunoinformatics for immunoglobulins or antibodies, T-cell receptors, human and vertebrate major histocompatibility, immunoglobulin superfamily and related proteins of the immune system of vertebrates and invertebrates (Lefranc *et al.*, 2014).

SYFPEITHI, one of the oldest databases in the field had been utilized together with IEDB (Vita *et al.*, 2014) to develop T cell epitope prediction tools. SYFPEITHY includes more than 7000 peptides that bind to MHC class I and II molecules (Rammensee *et al.*, 1999). With increasing utility of IEDB and integration of epitope data SYFPEITHY became static in 2012.

IEDB stores data on more than 300,000 peptide epitopes including epitope information derived from antigen-antibody complex of PDB, and more than 2500 non-peptide epitopes. The epitope data is not restricted to human and mouse but includes also chimpanzee, macaque, cow and swine. Integrated epitope prediction tools for linear and discontinuous B cell epitopes such as ElliPro (Ponomarenko *et al.*, 2008) or the pipeline TepiTool pipeline (Paul *et al.*, 2016) for vaccine, diagnostic, therapeutic candidate epitope discovery render IEDB by far the most user-friendly and effective resource. Yet a prediction tool that identifies in one process candidate B and T cell epitopes still awaits its implementation.

Illustrative Examples or Case Studies

The applications of epitope predictions are as multifarious as immunoinformatics the field that rose from the beginnings in theoretical immunology to make an impact on vaccine research and development. A representative example is reverse vaccinology (Sette and Rappuoli, 2010). This strategy allows the rapid design of novel vaccines based on pathogen genome information used in epitope predictions. Guttierrez and collaborators have demonstrated successfully the design of an effective epitope-based vaccine against swine Influenza A virus through MHC class I and II restricted epitope prediction using PigMatrix (Gutiérrez *et al.*, 2016). The vaccinated pigs responded to virus re-stimulation, showing that the epitope-based vaccination gave rise to T cells that were cross-reactive in vitro with epitopes present in the whole virus.

Another important application of epitope predictions are personalized therapeutic vaccines with the aim to treat epithelial cancer and melanoma as reviewed by Bobisse *et al.* (Bobisse *et al.*, 2016). Typically, epitope predictions of mutated antigens (neonantigens) involves NetMHC or NetMHCpan to identify high-affinity candidate neo-epitopes for *in vitro* validation before stimulating and expanding a patient's tumor infiltrating lymphocytes *ex vivo* for use in adoptive cell therapy. Several clinical trials are ongoing for example a phase I study on a personalized melanoma neoantigen cancer vaccine that started in 2013 (see relevant websites). Although it is too early to pass judgement on the success or failure of the approach even negative data, provided they are published and shared, can help to improve predictions, particularly prediction methods for class II restricted epitopes which perform less accurately than for class I.

In a recent report by Anagnostou *et al.* (2017) epitope prediction had an essential role to elucidate the mechanism of resistance to immune checkpoint blockade drugs in cancer patients. The study of neoantigen evolution during immune checkpoint blockade in non-small cell lung cancer required to assess the immunogenicity of somatic mutations in tumors using wild-type and mutated peptides. NetMHCpan was used to predict the class I binding potential of each peptide, and NetCTLpan to evaluate antigen processing to classify epitopes and non-epitopes. Interestingly, candidate mutation-associated neoantigens with high HLA binding affinity disappeared, probably driven by immune-mediated elimination of cancer cells, thereby depriving patients the means to mount an effective functional immune response.

A lesser known application area of epitope prediction is graft-versus-host disease (GVHD) in organ transplantation research. Monitoring GVHD onset and prevention depends critically on the knowledge of minor histocompatibility antigen (MiHA) match/mismatch between donor and recipient. Van Bergen and collaborators used NetCTLpan to evaluate amino acid polymorphisms among 19 MiHAs for the potential to be targeted by alloreactive CD8 T cells in GVHD patients (van Bergen *et al.*, 2017). The approach yielded 13 new MiHA T cell epitopes.

Numerous other case studies ranging from allergen cross-reactivity, infectious diseases, autoimmunity to therapeutic proteins could be listed here that demonstrate the applicability of current epitope prediction methods in basic, applied and clinical research accompanied by savings in time and cost.

Results and discussion

One of the challenges in epitope prediction is the high polymorphism of MHC genes. Immuno Polymorphism Database IPD-HLA/IMGT (release 3.29.0, 2017) contains 12,544 HLA class I alleles and 4622 HLA class II alleles (Robinson *et al.*, 2014). In IEDB only 272 (2.17%) HLA class I and 247 (5.34%) HLA class II allotypes are associated with experimental peptide binding or T cell epitope data. Therefore allotype-specific prediction methods (e.g. QM-based methods) are not able to deal with MHC allotypes that are uncharacterized with regard to peptide binding. The first method that satisfactorily addressed the issue for HLA-A and -B allotypes was NetMHCpan (Nielsen *et al.*, 2007; Zhang *et al.*, 2011a,b,c) by combining all HLA sequence and peptide data as input into an ANN to derive general features between HLA sequences and peptides and make inferences on binding affinities. Although the initial method has been refined and overcame the problem of variation in peptide length (Lundegaard *et al.*, 2008) and potential multiple binding frames for MHC class II (NetMHCIIpan) it has been of limited use for various non-human primate, bovine and swine MHC where dissimilarities among MHC sequences exceeded the threshold for reasonably accurate epitope predictions. Considering the effects of MHC sequence and peptide data diversity and redundancy on the accuracy of epitope predictions (Kim *et al.*, 2014) Mattsson *et al.* proposed in 2016 an MHC-pan method improvement that relies on MHC class I similarity redundancy reduction rather than peptide similarity by using pseudo-sequences derived from the binding cleft amino acid environment of each MHC class I molecule in the input space (Mattsson *et al.*, 2016). In principle, the method could be extended to all MHC class II allotypes opening the possibility to integrate OptiType HLA-typing from next-generation sequencing data (Szolek *et al.*, 2014) with epitope prediction for patient-specific therapeutic vaccine or in transplantation.

Despite the limitations of QM-based methods compared to ANN-based methods, they still yield robust results as demonstrated by De Groot and collaborators who have applied a genome-to-vaccine approach to design (not produce) an HLA class I and II epitope-based vaccine for avian H7N9 influenza using EpiMatrix in 20 h (De Groot *et al.*, 2013). The integration of EpiMatrix and JanusMatrix (Moise *et al.*, 2013), a prediction tool which allows to evaluate potential undesired cross-reactivity of predicted epitopes with human proteins or commensal gut bacteria that may trigger autoimmune responses into iVax (Moise *et al.*, 2015) improves the selection of epitope candidates to be tested for a vaccine. Unfortunately iVax is not an unrestricted-access tool which limits its spread and use, and may trigger future improvements of unrestricted accessible ANN-based tools emulating the JanusMatrix concept.

The performance of B cell epitope predictions is limited by the lack of a sufficiently accurate propensity scales for sequence-based methods and the scarcity of structural data of antigen-antibody complexes. For instance the training data sets of EPSVR and Discotope 2.0 comprised only 48 and 75 complexes, respectively. Therefore it is not too surprising that experimental validations of predicted B cell epitopes result in an average accuracy of 60% (Bergmann-Leitner *et al.*, 2013). Since NMR or X-ray crystallographic antigen-antibody and unbound structural data will increase only slowly, the development of new hybrid and meta-learning methods incorporating qualitative and quantitative features extracted from data generated by experimental methods of lower cost and effort such as phage-display library screening and "quality of antibody response" workflows that generate data of antibody binding from surface plasmon resonance (Davidoff *et al.*, 2015) and hydrogen deuterium exchange data combined with mass spectrometry (Yang *et al.*, 2016) are likely to improve the performance of B cell epitope predictions in the near future.

Epitope prediction tools are an integral part of the reverse vaccinology approach, yet a straightforward performance assessment similar to Critical Assessment of Structure Prediction (CASP) (Moult *et al.*, 2014) has not been conducted so far. CASP has helped to raise the accuracy of homology models and their acceptance as *bona fide* structural information source. In 2006 a NIAD B cell epitope prediction tool workshop (Greenbaum *et al.*, 2007) recommended the creation of annotated datasets and the development of methods and metrics to assess the prediction tools. IEDB has been very successful in assembling richly annotated T and B cell epitope data including negative non-epitope data. The latter were found to be biased towards short non-B-cell epitopes which negatively affected the performance of B cell epitope predictions when included in testing data sets (Rahman *et al.*, 2016). Yet, we are still lacking a community-based consensus procedure and minimal standards to assess the prediction tools. The establishment of a Critical Assessment of Epitope Predictions (CAEP) initiative is overdue to guide the improvement of predictions tools.

Future Directions

The success of epitope predictions particularly for HLA class I T cell epitopes has been largely driven by data and their level of integration with the current state of knowledge of immunological processes and host-pathogen interactions. Knowledge gaps in the details of immunological processes that define relations and differentiation dynamics of effector and memory T cell pools, and central and peripheral tolerance represent just two bottlenecks among others that require many more experimental data to advance predictive methods. At present we can identify immunogenic epitope candidates, but not optimal protective ones. For example, more work and data on molecular structures, TCR and BCR repertoire data are needed to predict cross-reactive and -neutralizing epitopes that will not be reduced in efficacy nor trigger autoimmunity when a pathogen is encountered two or more times.

Rational vaccine design and evaluation resembles in its temporal and mechanistic order of steps a biological pathway with epitope prediction positioned fairly upstream. Efforts to improve connectivity and feedback among individual components including epitope prediction may come to fruition on proof-of-concept level within the next five to ten year if concerted and standardized generation and collection of longitudinal data downstream of epitope predictions on gene and protein expression levels, protein-protein interactions and pathways, pathogen and antigen variation is funded and performed. The Human Vaccines Project (Koff *et al.*, 2014) a nonprofit public-private project that aims to elucidate the molecular cellular principles of vaccine-induced immunity is a promising high-impact initiative to enhance rational vaccine design. Translation into effective and affordable preventive vaccines for complex targets such as tuberculosis or HIV and therapeutic cancer vaccines may take longer because research progress on the effects and impact of natural genome and transcriptome variations in populations, and impact of environmental factors for example metals and chemicals, on the development, functioning and ageing of the immune system is slower. Areas deserving more attention include the development of algorithms to predict non-protein epitopes e.g. carbohydrates that are not unimportant for vaccines.

Closing Remarks

In the hierarchy of epitope prediction performance HLA class I binding predictions occupy the top position. Basically we can predict with current systems for any classical HLA class I allotype T cell epitope candidates. Second ranked are MHC class I-restricted T cell epitope predictions for chimpanzee, macaque, cow and swine. Therefore the development and actual use of epitope-based veterinary vaccines might precede the ones for human vaccines. Third ranked are predictions of HLA class II DR- and DQ-restricted epitope candidates. There is still room for significant improvements with regard to correctly identifying the core binding residues and increasing the number of experimental data for less studied allotypes, especially for HLA-DP. At the bottom of the performance hierarchy are B cell epitope predictions. This fact is expected to incentivize the development of both new experimental and data-driven computational methods that are anticipated to lift the accuracy and efficacy of B cell epitope predictions to acceptable levels until 2025.

Acknowledgement

R.K. acknowledges the award of an IRCMS Internship by International Research Center for Medical Sciences, Kumamoto University. C.S. acknowledges the support for the work on epitope predictions by Kumamoto University, International Research Center for Medical Sciences (#005–5700101122).

See also: Data Mining: Classification and Prediction. Data Mining: Prediction Methods. Extraction of Immune Epitope Information. Immunoinformatics Databases. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Protein Interactions: Looking Through the Kaleidoscope. Protein Post-Translational Modification Prediction. Protein Properties. Secondary Structure Prediction. Sequence Analysis. Sequence Composition. Supervised Learning: Classification. Vaccine Target Discovery

References

- Adams, H.-P., Koziol, J.A., 1995. Prediction of binding to MHC class I molecules. *Journal of Immunological Methods* 185, 181–190.
- Anagnostou, V., Smith, K.N., Forde, P.M., *et al.*, 2017. Evolution of neoantigen landscape during immune checkpoint blockade in non-small cell lung cancer. *Cancer Discovery* 7, 264–276.
- Andreatta, M., Karosiene, E., Rasmussen, M., *et al.*, 2015. Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification. *Immunogenetics* 67, 641–650.
- Andreatta, M., Nielsen, M., 2015. Gapped sequence alignment using artificial neural networks: Application to the MHC class I system. *Bioinformatics* 32, 511–517.
- Ansari, H.R., Raghava, G.P., 2010. Identification of conformational B-cell epitopes in an antigen from its primary sequence. *Immunome Research* 6, 6.
- Antonets, D., Maksyutov, A., 2010. TEPredict: Software for T-cell epitope prediction. *Molecular Biology* 44, 119–127.
- Atanasova, M., Patronov, A., Dimitrov, I., Flower, D.R., Doytchinova, I., 2013. EpiDOCK: A molecular docking-based tool for MHC class II binding prediction. *Protein Engineering, Design & Selection* 26, 631–634.
- Barlow, D., Edwards, M., Thornton, J., 1986. Continuous and discontinuous protein antigenic determinants. *Nature* 322, 747–748.
- Bergmann-Leitner, E.S., Chaudhry, S., Steers, N.J., *et al.*, 2013. Computational and experimental validation of B and T-cell epitopes of the in vivo immune response to a novel malarial antigen. *PLoS One* 8, e71610.
- Bhasin, M., Lata, S., Raghava, G., 2007. TAPPred prediction of TAP-binding peptides in antigens. *Immunoinformatics: Predicting Immunogenicity In Silico*. 381–386.
- Bhasin, M., Raghava, G., 2003. Prediction of promiscuous and high-affinity mutated MHC binders. *Hybridoma and Hybridomics* 22, 229–234.
- Bhasin, M., Raghava, G., 2004a. Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine* 22, 3195–3204.
- Bhasin, M., Raghava, G., 2004b. SVM based method for predicting HLA-DRB1* 0401 binding peptides in an antigen sequence. *Bioinformatics* 20, 421–423.
- Bhasin, M., Raghava, G., 2005. Pcleavage: An SVM based method for prediction of constitutive proteasome and immunoproteasome cleavage sites in antigenic sequences. *Nucleic Acids Research* 33, W202–W207.
- Bhasin, M., Raghava, G., 2007. A hybrid approach for predicting promiscuous MHC class I restricted T cell epitopes. *Journal of Biosciences* 32, 31–42.
- Blum, J.S., Wearsch, P.A., Creswell, P., 2013. Pathways of antigen processing. *Annual Review of Immunology* 31, 443–473.
- Blythe, M.J., Flower, D.R., 2005. Benchmarking B cell epitope prediction: Underperformance of existing methods. *Protein Science* 14, 246–248.
- Bobisse, S., Foukas, P.G., Coukos, G., Harari, A., 2016. Neoantigen-based cancer immunotherapy. *Annals of Translational Medicine* 4, 262.
- Brusic, V., Petrovsky, N., Zhang, G., Bajic, V.B., 2002. Prediction of promiscuous peptides that bind HLA class I molecules. *Immunology and Cell Biology* 80, 280.
- Chen, J., Liu, H., Yang, J., Chou, K.-C., 2007. Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. *Amino acids* 33, 423–428.
- Dalkas, G.A., Rومان, M., 2017. SEPla, a knowledge-driven algorithm for predicting conformational B-cell epitopes from the amino acid sequence. *BMC Bioinformatics* 18, 95.
- Davidoff, S.N., Ditto, N.T., Brooks, A.E., Eckman, J., Brooks, B.D., 2015. Surface plasmon resonance for therapeutic antibody characterization. *Label-Free Biosensor Methods in Drug Discovery*. 35–76.
- De Groot, A.S., Einck, L., Moise, L., *et al.*, 2013. Making vaccines “on demand” A potential solution for emerging pathogens and biodefense? *Human Vaccines and Immunotherapeutics* 9, 1877–1884.
- De Groot, A.S., Sbai, H., Saint Aubin, C., *et al.*, 2002. Immuno-informatics: Mining genomes for vaccine components. *Immunology and Cell Biology* 80, 255.
- Dhanda, S.K., Gupta, S., Vir, P., Raghava, G., 2013a. Prediction of IL4 inducing peptides. *Clinical and Developmental Immunology* 2013, 263952.
- Dhanda, S.K., Vir, P., Raghava, G.P., 2013b. Designing of interferon-gamma inducing MHC class-II binders. *Biology Direct* 8, 30.
- Diez-Rivero, C.M., Chenlo, B., Zuluaga, P., Reche, P.A., 2010. Quantitative modeling of peptide binding to TAP using support vector machine. *Proteins: Structure, Function, and Bioinformatics* 78, 63–72.
- Dimitrov, I., Garnev, P., Flower, D.R., Doytchinova, I., 2010a. EpiTOP – a proteochemometric tool for MHC class II binding prediction. *Bioinformatics* 26, 2066–2068.
- Dimitrov, I., Garnev, P., Flower, D.R., Doytchinova, I., 2010b. Peptide binding to the HLA-DRB1 supertype: A proteochemometrics analysis. *European Journal of Medicinal Chemistry* 45, 236–243.
- Dönnes, P., Kohlbacher, O., 2006. SVMHC: A server for prediction of MHC-binding peptides. *Nucleic Acids Research* 34, W194–W197.
- Doytchinova, I.A., Flower, D.R., 2002. Physicochemical explanation of peptide binding to HLA-A* 0201 major histocompatibility complex: A three-dimensional quantitative structure-activity relationship study. *Proteins: Structure, Function, and Bioinformatics* 48, 505–518.
- Frank, S.A., 2002. *Immunology and Evolution of Infectious Disease*. Princeton, NJ: Princeton University Press.
- Greenbaum, J.A., Andersen, P.H., Blythe, M., *et al.*, 2007. Towards a consensus on datasets and evaluation metrics for developing B-cell epitope prediction tools. *Journal of Molecular Recognition* 20, 75–82.
- Grotzke, J.E., Sengupta, D., Lu, Q., Cresswell, P., 2017. The ongoing saga of the mechanism (s) of MHC class I-restricted cross-presentation. *Current Opinion in Immunology* 46, 89–96.
- Guan, P., Hattotuwigama, C.K., Doytchinova, I.A., Flower, D.R., 2006. MHCpred 2.0. *Applied Bioinformatics* 5, 55–61.
- Guo, L., Luo, C., Zhu, S., 2013. MHC2SKpan: A novel kernel based approach for pan-specific MHC class II peptide binding prediction. *BMC Genomics* 14, S11.
- Gupta, S., Ansari, H.R., Gautam, A., Raghava, G.P., 2013. Identification of B-cell epitopes in an antigen for inducing specific class of antibodies. *Biology Direct* 8, 27.
- Gutiérrez, A.H., Loving, C., Moise, L., *et al.*, 2016. In vivo validation of predicted and conserved T cell epitopes in a swine influenza model. *PLoS One* 11, e0159237.
- Hattotuwigama, C.K., Doytchinova, I.A., Flower, D.R., 2007. Toward the prediction of class I and II mouse major histocompatibility complex-peptide-binding affinity: In silico bioinformatic step-by-step guide using quantitative structure-activity relationships. *Immunoinformatics: Predicting Immunogenicity In Silico*. 227–245.
- Honeyman, M.C., Brusic, V., Stone, N.L., Harrison, L.C., 1998. Neural network-based prediction of candidate T-cell epitopes. *Nature Biotechnology* 16, 966–969.
- Huang, Y.X., Bao, Y.L., Guo, S.Y., *et al.*, 2008. Pep-3D-Search: A method for B-cell epitope prediction based on mimotope analysis. *BMC Bioinformatics* 9, 538.
- Jacob, L., Vert, J.-P., 2007. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics* 24, 358–366.
- Jespersen, M.C., Peters, B., Nielsen, M., Marcotili, P., 2017. BepiPred-2.0: Improving sequence-based B-cell epitope prediction using conformational epitopes. *Nucleic Acids Research* 45, W24–W29.
- Karosiene, E., Lundegaard, C., Lund, O., Nielsen, M., 2012. NetMHCcons: A consensus method for the major histocompatibility complex class I predictions. *Immunogenetics* 64, 177–186.
- Kim, Y., Sidney, J., Buus, S., *et al.*, 2014. Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions. *BMC Bioinformatics* 15, 241.
- Koff, W.C., Gust, I.D., Plotkin, S.A., 2014. Toward a human vaccines project. *Nature Immunology* 15, 589–592.
- Kringelum, J.V., Lundegaard, C., Lund, O., Nielsen, M., 2012. Reliable B cell epitope predictions: Impacts of method development and improved benchmarking. *PLOS Computational Biology* 8, e1002829.
- Kulkarni-Kale, U., Bhosle, S., Kolaskar, A.S., 2005. CEP: A conformational epitope prediction server. *Nucleic Acids Research* 33, W168–W171.
- Larsen, J.E., Lund, O., Nielsen, M., 2006. Improved method for predicting linear B-cell epitopes. *Immunome Research* 2, 2.
- Larsen, M.V., Lundegaard, C., Lamberth, K., *et al.*, 2007. Large-scale validation of methods for cytotoxic T-lymphocyte epitope prediction. *BMC Bioinformatics* 8, 424.
- Lata, S., Bhasin, M., Raghava, G.P., 2007. Application of machine learning techniques in predicting MHC binders. *Methods in Molecular Biology* 409, 201–215.
- Lefranc, M.-P., Giudicelli, V., Duroux, P., *et al.*, 2014. IMGT[®], the international ImmunoGeneTics information system[®] 25 years on. *Nucleic Acids Research* 43, D413–D422.

- Liang, S., Zheng, D., Standley, D.M., *et al.*, 2010. EPSVR and EPMeta: Prediction of antigenic epitopes using support vector regression and multiple server results. *BMC Bioinformatics* 11, 381.
- Liang, S., Zheng, D., Zhang, C., Zacharias, M., 2009. Prediction of antigenic epitopes on protein surfaces by consensus scoring. *BMC Bioinformatics* 10, 302.
- Litman, G.W., Rast, J.P., Fugmann, S.D., 2010. The origins of vertebrate adaptive immunity. *Nature Reviews. Immunology* 10, 543.
- Liu, G., Li, D., Li, Z., *et al.*, 2017. PSSMHCpan: A novel PSSM-based software for predicting class I peptide-HLA binding affinity. *Giga Science* 6, 1–11.
- Logean, A., Rognan, D., 2002. Recovery of known T-cell epitopes by computational scanning of a viral genome. *Journal of Computer-aided Molecular Design* 16, 229–243.
- Lundegaard, C., Lund, O., Nielsen, M., 2008. Accurate approximation method for prediction of class I MHC affinities for peptides of length 8, 10 and 11 using prediction tools trained on 9mers. *Bioinformatics* 24, 1397–1398.
- Mamitsuka, H., 1998. Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models. *Proteins Structure Function and Genetics* 33, 460–474.
- Mattsson, A.H., Kringelum, J.V., Garde, C., Nielsen, M., 2016. Improved pan-specific prediction of MHC class I peptide binding using a novel receptor clustering data partitioning strategy. *HLA* 88, 287–292.
- Moise, L., Gutierrez, A., Kibria, F., *et al.*, 2015. iVAX: An integrated toolkit for the selection and optimization of antigens and the design of epitope-driven vaccines. *Human Vaccines and Immunotherapeutics* 11, 2312–2321.
- Moise, L., Gutierrez, A.H., Bailey-Kellogg, C., *et al.*, 2013. The two-faced T cell epitope: Examining the host-microbe interface with JanusMatrix. *Human Vaccines and Immunotherapeutics* 9, 1577–1586.
- Moreau, V., Granier, C., Villard, S., Laune, D., Molina, F., 2006. Discontinuous epitope prediction based on mimotope analysis. *Bioinformatics* 22, 1088–1095.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2014. Critical assessment of methods of protein structure prediction (CASP) – round x. *Proteins: Structure, Function, and Bioinformatics* 82, 1–6.
- Moutafsi, M., Peters, B., Pasquetto, V., *et al.*, 2006. A consensus epitope prediction approach identifies the breadth of murine TCD8 + -cell responses to vaccinia virus. *Nature Biotechnology* 24, 817.
- Mukherjee, S., Bhattacharyya, C., Chandra, N., 2016. HLaffy: Estimating peptide affinities for Class-I HLA molecules by learning position-specific pair potentials. *Bioinformatics* 32, 2297–2305.
- Negi, S.S., Braun, W., 2009. Automated detection of conformational epitopes using phage display peptide sequences. *Bioinformatics and Biology Insights* 3, 71.
- Neu, K.E., Wilson, P.C., 2016. Taking the broad view on B cell affinity maturation. *Immunity* 44, 518–520.
- Nielsen, M., Andreatta, M., 2016. NetMHCpan-3.0; improved prediction of binding to MHC class I molecules integrating information from multiple receptor and peptide length datasets. *Genome Medicine* 8, 33.
- Nielsen, M., Lund, O., 2009. NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinformatics* 10, 296.
- Nielsen, M., Lundegaard, C., Blicher, T., *et al.*, 2007. NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and-B locus protein of known sequence. *PLoS One* 2, e796.
- Nielsen, M., Lundegaard, C., Lund, O., Keşmir, C., 2005. The role of the proteasome in generating cytotoxic T-cell epitopes: Insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics* 57, 33–41.
- Ogishi, M., Yotsuyanagi, H., 2017. Epitope immunogenicity prediction through repertoire-wide TCR-peptide contact profiles. *bioRxiv* 155317.
- Oyarzun, P., Ellis, J.J., Bodén, M., Kobe, B., 2013. PREDIVAC: CD4 + T-cell epitope prediction for vaccine design that covers 95% of HLA class II DR protein diversity. *BMC Bioinformatics* 14, 52.
- Parker, K.C., Bednarek, M.A., Coligan, J.E., 1994. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *The Journal of Immunology* 152, 163–175.
- Patronov, A., Doytchinova, I., 2013. T-cell epitope vaccine design by immunoinformatics. *Open Biology* 3, 120139.
- Paul, S., Sidney, J., Sette, A., Peters, B., 2016. TepiTool: A pipeline for computational prediction of T cell epitope candidates. *Current Protocols in Immunology* 114, 18.19.1.
- Ponomarenko, J., Bui, H.-H., Li, W., *et al.*, 2008. ElliPro: A new structure-based tool for the prediction of antibody epitopes. *BMC Bioinformatics* 9, 514.
- Potocnakova, L., Bhide, M., Pulzova, L.B., 2016. An Introduction to B-cell epitope mapping and in silico epitope prediction. *Journal of Immunology Research* 2016, 6760830.
- Qi, T., Qiu, T., Zhang, Q., *et al.*, 2014. SEPPA 2.0 – more refined server to predict spatial epitope considering species of immune host and subcellular localization of protein antigen. *Nucleic Acids Research* 42, W59–W63.
- Querec, T.D., Akondy, R.S., Lee, E.K., *et al.*, 2009. Systems biology approach predicts immunogenicity of the yellow fever vaccine in humans. *Nature Immunology* 10, 116–125.
- Rahman, K.S., Chowdhury, E.U., Sachse, K., Kaltenboeck, B., 2016. Inadequate reference datasets biased toward short non-epitopes confound B-cell epitope prediction. *The Journal of Biological Chemistry* 29, 14585–14599.
- Rammensee, H.-G., Bachmann, J., Emmerich, N.P.N., Bachor, O.A., Stevanović, S., 1999. SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* 50, 213–219.
- Rappuoli, R., Aderem, A., 2011. A 2020 vision for vaccines against HIV, tuberculosis and malaria. *Nature* 473, 463.
- Reche, P.A., Glutting, J.-P., Zhang, H., Reinherz, E.L., 2004. Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. *Immunogenetics* 56, 405–419.
- Robinson, J., Halliwell, J.A., Hayhurst, J.D., *et al.*, 2014. The IPD and IMGT/HLA database: Allele variant databases. *Nucleic Acids Research* 43, D423–D431.
- Rubinstein, N.D., Mayrose, I., Martz, E., Pupko, T., 2009. EpiTopt: A web-server for predicting B-cell epitopes. *BMC Bioinformatics* 10, 287.
- Saha, S., Bhasin, M., Raghava, G.P., 2005. Bcipep: A database of B-cell epitopes. *BMC Genomics* 6, 79.
- Saha, S., Raghava, G., 2006. Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. *Proteins: Structure, Function, and Bioinformatics* 65, 40–48.
- Saha, S., Raghava, G.P.S., 2004. BcePred: Prediction of continuous B-Cell epitopes in antigenic sequences using physico-chemical Properties. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (Eds.), ICARIS. Berlin: Springer, pp. 197–204.
- Sammur, C., Webb, G.I., 2010. Leave-One-Out Cross-Validation. In: Sammur, C., Webb, G.I. (Eds.), *Encyclopedia of Machine Learning*, 2nd edn. Boston: Springer US.
- Schäfer, J.R.A., Jesdale, B.M., George, J.A., Kouttab, N.M., De Groot, A.S., 1998. Prediction of well-conserved HIV-1 ligands using a matrix-based algorithm, EpiMatrix. *Vaccine* 16, 1880–1884.
- Schönbach, C., Kun, Y., Brusci, V., 2002. Large-scale computational identification of HIV T-cell epitopes. *Immunology and Cell Biology* 80, 300.
- Schroeder, H.W., Cavacini, L., 2010. Structure and function of immunoglobulins. *Journal of Allergy and Clinical Immunology* 125, S41–S52.
- Schubert, B., Walzer, M., Brachvogel, H.-P., *et al.*, 2016. FRED 2: An immunoinformatics framework for Python. *Bioinformatics* 32, 2044–2046.
- Schueler-Furman, O., Altuvia, Y., Sette, A., Margalit, H., 2000. Structure-based prediction of binding peptides to MHC class I molecules: Application to a broad range of MHC alleles. *Protein Science* 9, 1838–1846.
- Sela-Culang, I., Kunik, V., Ofan, Y., 2013. The structural basis of antibody-antigen recognition. *Frontiers in Immunology* 4, 302.
- Sette, A., Rappuoli, R., 2010. Reverse vaccinology: Developing vaccines in the era of genomics. *Immunity* 33, 530–541.
- Singh, H., Ansari, H.R., Raghava, G.P., 2013. Improved method for linear B-cell epitope prediction using antigen's primary sequence. *PLoS One* 8, e62216.
- Singh, H., Raghava, G., 2001. ProPred: Prediction of HLA-DR binding sites. *Bioinformatics* 17, 1236–1237.
- Singh, H., Raghava, G., 2003. ProPred1: Prediction of promiscuous MHC Class-I binding sites. *Bioinformatics* 19, 1009–1014.
- Soria-Guerra, R.E., Nieto-Gomez, R., Govea-Alonso, D.O., Rosales-Mendoza, S., 2015. An overview of bioinformatics tools for epitope prediction: Implications on vaccine development. *Journal of Biomedical Informatics* 53, 405–414.

- Stranzl, T., Larsen, M.V., Lundegaard, C., Nielsen, M., 2010. NetCTLpan: Pan-specific MHC class I pathway epitope predictions. *Immunogenetics* 62, 357–368.
- Sun, P., Ju, H., Liu, Z., *et al.*, 2013. Bioinformatics resources and tools for conformational B-cell epitope prediction. *Computational and Mathematical Methods in Medicine* 2013, 943636.
- Sun, P., Qi, J., Zhao, Y., *et al.*, 2016. A novel conformational B-cell epitope prediction method based on mimotope and patch analysis. *Journal of Theoretical Biology* 394, 102–108.
- Sweredoski, M.J., Baldi, P., 2008a. COBEpro: A novel system for predicting continuous B-cell epitopes. *Protein Engineering, Design and Selection* 22, 113–120.
- Sweredoski, M.J., Baldi, P., 2008b. PEPITO: Improved discontinuous B-cell epitope prediction using multiple distance thresholds and half sphere exposure. *Bioinformatics* 24, 1459–1460.
- Szolek, A., Schubert, B., Mohr, C., *et al.*, 2014. OptiType: Precision HLA typing from next-generation sequencing data. *Bioinformatics* 30, 3310–3316.
- Taylor, W., Thornton, J.T., Turnell, W., 1983. An ellipsoidal approximation of protein shape. *Journal of Molecular Graphics* 1, 30–38.
- Thornton, J., Edwards, M., Taylor, W., Barlow, D., 1986. Location of 'continuous' antigenic determinants in the protruding regions of proteins. *The EMBO Journal* 5, 409.
- Trolle, T., Nielsen, M., 2014. NetTepi: An integrated method for the prediction of T cell epitopes. *Immunogenetics* 66, 449–456.
- van Bergen, C.A., Van Luxemburg-Heijs, S.A., De Wreede, L.C., *et al.*, 2017. Selective graft-versus-leukemia depends on magnitude and diversity of the alloreactive T cell response. *The Journal of Clinical Investigation* 127, 517.
- Vita, R., Overton, J.A., Greenbaum, J.A., *et al.*, 2014. The immune epitope database (IEDB) 3.0. *Nucleic Acids Research* 43, D405–D412.
- Xu, Y., Luo, C., Mamitsuka, H., Zhu, S., 2016. MetaMHCpan, a meta approach for pan-specific MHC peptide binding prediction. *Vaccine Design: Methods and Protocols, Volume 2: Vaccines for Veterinary Diseases*. 753–760.
- Yang, D., Frego, L., Lasaro, M., *et al.*, 2016. Efficient qualitative and quantitative determination of antigen-induced immune responses. *The Journal of Biological Chemistry* 291, 16361–16374.
- Yang, X., Yu, X., 2009. An introduction to epitope prediction methods and software. *Reviews in Medical Virology* 19, 77–96.
- Yao, B., Zheng, D., Liang, S., Zhang, C., 2013. Conformational B-cell epitope prediction on antigen protein structures: A review of current algorithms and comparison with common binding site prediction methods. *PLOS One* 8, e62249.
- Yao, B., Zhang, L., Liang, S., Zhang, C., 2012. SVMTriP: A method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity. *PLOS One* 7, e45152.
- Yasser, E.-M., Honavar, V., 2010. Recent advances in B-cell epitope prediction methods. *Immunome Research* 6, S2.
- Yewdell, J.W., Bennink, J.R., 1999. Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. *Annual Review of Immunology* 17, 51–88.
- Zhang, C., Bickis, M.G., Wu, F.-X., Kusalik, A.J., 2006. Optimally-connected hidden markov models for predicting MHC-binding peptides. *Journal of Bioinformatics and Computational Biology* 4, 959–980.
- Zhang, G.L., Bozic, I., Kwoh, C.K., August, J.T., Brusica, V., 2007. Prediction of supertype-specific HLA class I binding peptides using support vector machines. *Journal of Immunological Methods* 320, 143–154.
- Zhang, G.L., Deluca, D.S., Keskin, D.B., *et al.*, 2011a. MULTIPRED2: A computational system for large-scale identification of peptides predicted to bind to HLA supertypes and alleles. *Journal of Immunological Methods* 374, 53–61.
- Zhang, H., Lund, O., Nielsen, M., 2009. The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: Application to MHC-peptide binding. *Bioinformatics* 25, 1293–1299.
- Zhang, L., Chen, Y., Wong, H.-S., *et al.*, 2012. TEPITOPEpan: Extending TEPITOPE for peptide binding prediction covering over 700 HLA-DR molecules. *PLOS One* 7, e30483.
- Zhang, L., Udaka, K., Mamitsuka, H., Zhu, S., 2011b. Toward more accurate pan-specific MHC-peptide binding prediction: A review of current methods and tools. *Briefings in Bioinformatics* 13, 350–364.
- Zhang, W., Xiong, Y., Zhao, M., *et al.*, 2011c. Prediction of conformational B-cell epitopes from 3D structures by random forests with a distance-based feature. *BMC Bioinformatics* 12, 341.

Further Reading

- Belden, O.S., Baker, S.C., Baker, B.M., 2015. Citizens unite for computational immunology!. *Trends in Immunology* 36, 385–387.
- Brusica, V., Gottardo, R., Kleinstein, S.H., Davis, M.M., 2014. Computational resources for high-dimensional immune analysis from the Human Immunology Project Consortium. *Nature Biotechnology* 32, 146–148.
- De, R.K., Tomar, N., 2014. Immunoinformatics. [eds.] In: Walker, (Ed.), *Methods in Molecular Biology*, second ed., 1184. New York: Humana Press.
- De Gregorio, E., Rappuoli, R., 2014. From empiricism to rational design: A personal perspective of the evolution of vaccine development. *Nature Reviews. Immunology* 14, 505.
- Ditto, N.T., Brooks, B.D., 2016. The emerging role of biosensor-based epitope binning and mapping in antibody-based drug discovery. *Expert Opinion on Drug Discovery* 11, 925–937.
- Fleri, W., Paul, S., Dhand, S.K., *et al.*, 2017. The immune epitope database and analysis resource in epitope discovery and synthetic vaccine design. *Frontiers in Immunology* 8, 278.
- He, L., Zhu, J., 2015. Computational tools for epitope vaccine design and evaluation. *Current Opinion in Virology* 11, 103–112.
- Liljeroos, L., Malito, E., Ferlenghi, I., Bottomley, M.J., 2015. Structural and computational biology in the design of immunogenic vaccine antigens. *Journal of Immunology Research* 2015, 156241.
- Scheuermann, R.H., Sinkovits, R.S., Schenkelberg, T., Koff, W.C., 2017. A bioinformatics roadmap for the human vaccines project. *Expert Review of Vaccines* 16, 535–544.
- Sette, A., Peters, B., 2007. Immune epitope mapping in the post-genomic era: Lessons for vaccine development. *Current Opinion in Immunology* 19, 106–110.

Relevant Websites

- <https://clinicaltrials.gov/ct2/show/NCT01970358>
A phase I study with a personalized neoantigen cancer vaccine in melanoma.
- <https://www.immunospace.org/>
Enabling integrative modelling of human immunological data. The Human Immunology Project Consortium.
- <http://www.iedb.org/>
Immune Epitope Database Analysis Resource.
- <http://www.imgt.org>
IMGT[®], the international ImMunoGeneTics information system[®].
- <http://www.humanvaccinesproject.org>
The Human Vaccines Project.

Biographical Sketch

Roman Kogay is an undergraduate student at School of Science and Technology, Department of Biology, Nazarbayev University. His research interests include application of bioinformatics in metabolomics, genomics, proteomics and biomedical sciences. After graduation he hopes to pursue a doctoral degree.

Christian Schönbach is Professor at Graduate School of Medical Sciences, International Research Center for Medical Sciences, Kumamoto University. Prior to joining Kumamoto University he was Professor at School of Science and Technology, Department of Biology, Nazarbayev University (2013–2016). His research and teaching interests revolve around bioinformatics, genomics and immunology. Since 2010 he is serving the bioinformatics community of Asia-Pacific Bioinformatics Network (APBioNet) in various leadership roles.

Immunoglobulin Clonotype and Ontogeny Inference

Pazit Polak, Ramit Mehr, and Gur Yaari, Bar-Ilan University, Ramat Gan, Israel

© 2019 Elsevier Inc. All rights reserved.

Introduction

The adaptive immune response functions mainly through B and T lymphocytes. B lymphocytes express immunoglobulins (Igs, aka antibodies), which specifically bind antigens on surfaces of pathogens and neutralize them. Specificity towards such a large pool of threats is achieved by the diversity and dynamics of Ig repertoires. The DNA encoding for Igs comes from random selection of the variable (V), diversity (D), and joining (J) genes, through somatic rearrangement. In addition to the combinatorial diversity coming from the choice of V, D, and J genes, further diversity is added at the boundaries where the segments are joined, by imprecise excision and addition of random nucleotides, resulting in a staggering receptor diversity of up to 10^{11} different B cell clones in each human (see Fig. 1) (Boyd *et al.*, 2010, 2009; Murphy and Weaver, 2016; Weinstein *et al.*, 2009).

Following initial antigen recognition, B cells further undergo cycles of affinity maturation, during which their rearranged Ig gene loci are mutated in a process called somatic hypermutation (SHM), and cells with increased affinity to the antigen are selected for clonal expansion. This results in improved affinity of the Igs to the antigens, and the whole process is referred to as affinity maturation. The outcome of affinity maturation is short- or long-lived plasma cells secreting Igs, and short- or long-lived memory B cells.

Thus, the Ig repertoire of an individual stores information about current and past threats that the body has encountered. In addition to studying the development of the adaptive immune system (Pogorelyy *et al.*, 2017; Rechavi and Somech, 2017; Rechavi *et al.*, 2015) and other fundamental processes underlying the immune system in healthy individuals (Amaout *et al.*, 2011; Boyd *et al.*, 2010, 2009; Wu *et al.*, 2010b), investigation of the repertoire has the potential to reveal dysregulation in abnormal conditions such as insufficient, overly active or misdirected immune responses (von Büdingen *et al.*, 2012; Cameron *et al.*, 2009; Lehmann-Horn *et al.*, 2013; Palanichamy *et al.*, 2014; Singh *et al.*, 2013; Snir *et al.*, 2015; Stern *et al.*, 2014; Zuckerman *et al.*, 2010a), as well as infectious diseases (Laserson *et al.*, 2014; Parameswaran *et al.*, 2013; Sok *et al.*, 2013; Tsioris *et al.*, 2015), allergy (Patil *et al.*, 2015; Wu *et al.*, 2014), cancer (Fridman *et al.*, 2012; Galon *et al.*, 2006; Glanville *et al.*, 2011; Jiang *et al.*, 2015; Katoh *et al.*, 2017; Lossos *et al.*, 2000; Yahalom *et al.*, 2013), and aging (Ademokun *et al.*, 2011; Dunn-Walters and Ademokun, 2010; Dunn-Walters *et al.*, 2003; Wu *et al.*, 2012).

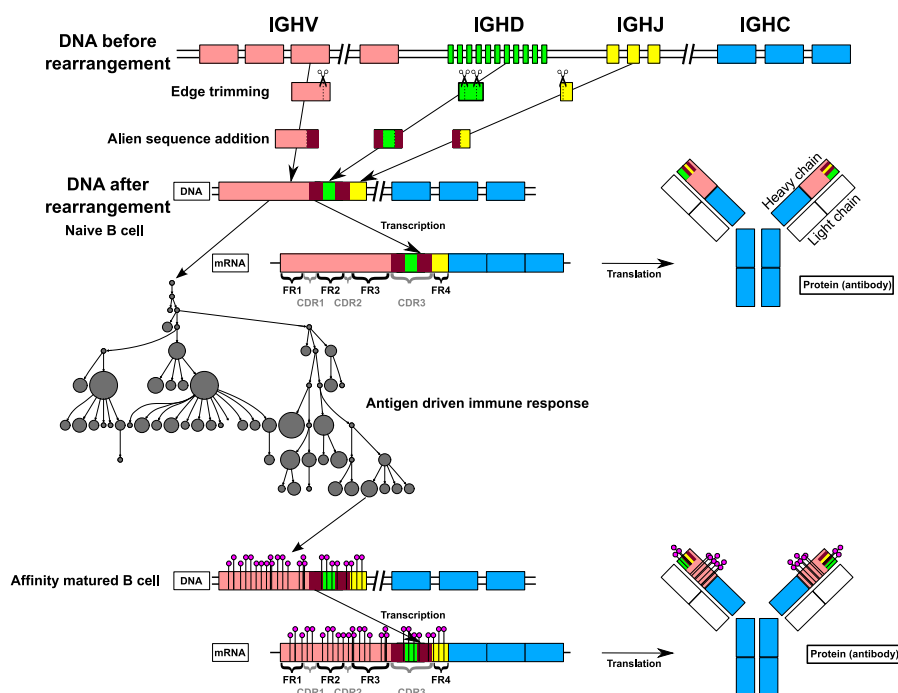


Fig. 1 B cell maturation. Stem cells DNA contains many potential genes for the V, D and J regions. During B cell development, one V gene, one D gene and one J gene are recombined to form the variable region of the Ig. Further diversity is added at the boundaries between the segments, by trimming and template-less addition of nucleotides. After encountering a pathogen, activated B cells undergo an affinity maturation process, resulting in mutated high-affinity Igs.

Ig repertoires are also starting to be utilized in several clinical settings including the detection of minimal residual disease in leukemia (Logan *et al.*, 2011) and analysis of tumor-infiltrating B cells for cancer prognosis (Zhang *et al.*, 2017). A snapshot of the immune system could serve as a natural bio-sensor for diagnostic and prognostic purposes. Indeed, recent studies have found that common Ig sequences can be found in unrelated individuals, for example, following Dengue (Parameswaran *et al.*, 2013) or Zika virus infection (Magnani *et al.*, 2017), in multiple sclerosis (Cameron *et al.*, 2009), and rheumatoid arthritis (Tak *et al.*, 2017), and therefore may be of diagnostic value. Repertoire analysis could also allow for diagnosis of diseases in early stages, and personalized medical treatment. Moreover, analysis of Ig repertoire sequential snapshots can teach us about the dynamics of the arms race between the immune system and the threat.

The diverse repertoire of B lymphocytes is constantly changing. Ig diversification endows the system with the ability to recognize any biological molecule or pathogen. However, the huge diversity and dynamic nature of Ig repertoires make their study challenging. Recent developments in high throughput sequencing (HTS) enable researchers to obtain large numbers of sequences from multiple samples simultaneously and increase the detection power (Ademokun *et al.*, 2011; Boyd *et al.*, 2010; Campbell *et al.*, 2008; Prabakaran *et al.*, 2012; Scheid *et al.*, 2009). Here, we describe protocols for construction Ig libraries for HTS, analysis methods, clonotype inference, i.e. classification of Igs into clones based on their V(D)J sequences, ontogeny inference, i.e. lineage trees of Ig evolution during antigen-driven affinity selection, and applications of clonotype and ontogeny inference.

Sequencing Approaches – Pros and Cons

Dramatic improvements in HTS technologies now enable large-scale characterization of Ig repertoires (Benichou *et al.*, 2012; Georgiou *et al.*, 2014). HTS platforms such as MiSeq by Illumina are being exploited by most researchers in the field (Arnaout *et al.*, 2011; Boyd *et al.*, 2010, 2009; von Büdingen *et al.*, 2012; Jiang *et al.*, 2011; Krause *et al.*, 2011; Weinstein *et al.*, 2009; Wu *et al.*, 2010b). MiSeq can generate 10 to 20 million paired-end 300 base-pair reads in a single run with relatively few errors (Loman *et al.*, 2012) and is expected to be extended to 400 base-pair reads in the near future. Other HTS platforms include Ion torrent, Pacbio, and minION (Goodwin *et al.*, 2016).

The choice of library preparation protocol involves several decisions that influence the experimental setup, the bioinformatic pipeline, and the type of information that can be extracted from the data. For example, single-cell sequencing is becoming more and more widespread in many biological applications, and is expected to spread also into Ig repertoire sequencing (Briggs *et al.*, 2017; DeKosky *et al.*, 2014). However, current technologies for Ig single cell sequencing are dramatically more expensive and complicated, and as a consequence far less popular. Other aspects that need to be taken into consideration when choosing a library preparation protocol are whether to use DNA or RNA as starting material, which sets of primers should be used, and if and how to include unique molecular identifiers. The pros and cons of choosing between DNA vs. RNA are summarized in Table 1.

UMI

A relatively recent major improvement to Ig repertoire library preparation protocols is the inclusion of unique molecular identifiers (UMIs), i.e. random sequences of 10–15 nucleotides that are added to the tail of the reverse transcription primer(s) (Jiang *et al.*, 2013; Mamedov *et al.*, 2013; Shiroguchi *et al.*, 2012; Shugay *et al.*, 2014). These barcodes tag each individual molecule with a specific barcode, so each amplicon can be traced to a consensus sequence, providing a means to distinguish real somatic mutations

Table 1 Pros and Cons for using DNA vs. RNA

	DNA	RNA
Number of molecules	(+) One DNA molecule encoding for one type of functional Ig in each cell, and thus the sequencing results reflect cell type distribution to some extent.	(–) Many RNA molecules in each cell, the number is highly variable between cells, thus the sequencing results reflect mRNA distribution which can be skewed due to Ig secreting plasmablasts.
Material stability	(+) Much more stable, enables working with difficult samples, e.g., formalin preserved.	(–) Less stable. Even if stored in -80°C , RNA can degrade after a few months.
Number of steps required for library preparation	(+) Ready for PCR.	(–) Requires an additional reverse transcription step.
Isotype information	(–) Lost, since the intron between the constant and variable regions is too large for PCR.	(+) One can use primers corresponding to the 5' edge of the constant region to obtain isotype information with minimal addition to the length of each molecule in the library.
Published protocols	(–) Few. None includes UMI, which makes PCR and sequencing error as well as bias correction impossible.	(+) Many, including UMI, so error and bias correction are possible.

from PCR amplification and sequencing-dependent errors. UMIs also help in correcting PCR amplification biases stemming from varied primer efficiencies. Although theoretically possible, no protocol for using UMIs on Ig DNAs has been published yet.

The Choice of Primer Sets

The high diversity of the Ig encoding region poses a challenge for reverse transcription or amplification during library preparation. In humans, for example, there are 9 possible constant genes for IgM, IgD, IgE, IgA1, IgA2, IgG1, IgG2, IgG3, IgG4, or 6 possible J genes, and there is no single primer that can capture all constant or J genes at the 3' end. At the 5' end, the challenge is even greater, with more than 200 possibilities for functional V alleles in the heavy chain (Lefranc *et al.*, 2009). To cover these sequences there is a need for a minimum of ~40 primers. Extension of the transcript into the 5'UTR reduces the minimum number of necessary primers to 12, still a considerable and bias-prone number for a single PCR reaction. Moreover, due to SHM, even if the Ig sequence originated from a sequence that was included in the primer set, the mutated Ig sequence, including its 5' UTR, can be very different and hence will not be amplified. A method to circumvent the need for so many primers is to perform the reverse transcription with primers only for the J or constant regions, using a 5' RACE polymerase. This polymerase incorporates a sequence of several free cytosine (C) nucleotides at the end of each newly synthesized molecule, followed by template switching with any primer that contains a sequence of 3 RNA guanine (rG) nucleotides at the 3' end. As a result, 5' RACE eliminates the potential amplification bias and can be used to amplify highly mutated sequences and sequences of species with limited information about the diversity of their Ig loci. To date, 5' RACE can only be used when the starting material is RNA.

Single Cell Sequencing

Each Ig molecule comprises two identical light chains and two identical heavy chains, linked by disulfide bonds. Most current experimental protocols sequence only the heavy chain gene, since this gene contains most of the diversity of the Ig molecule. In some cases, heavy and light chains are sequenced separately from the same sample, but high-throughput pairing of light and heavy chain sequences originating from a single lymphocyte remains a highly non-trivial challenge (Busse *et al.*, 2014; Dekosky *et al.*, 2013, 2014; Tan *et al.*, 2014; Zhu *et al.*, 2013). Currently, the best experimental approach for retaining the pairing of heavy and light chains is performing single cell barcoding (Briggs *et al.*, 2017) or linkage PCR (Dekosky *et al.*, 2013; McDaniel *et al.*, 2016) within emulsion droplets. Although promising, currently these methods are much more costly than bulk sequencing, typically resulting in a reduced number of sequenced cells per experiment. Another approach is to combinatorically identify and pair heavy and light chains based on their frequencies in the B cell repertoire (Zhu *et al.*, 2013), i.e. the most abundant VL with the most abundant VH, and so forth. However, this approach is noisy for large data sets and works well only for very abundant sequences.

Library Preparation

B cell isolation or enrichment can be performed, for example, by FACS sorting or ficoll gradient, respectively. Cells are lysed and RNA or DNA is extracted. Then the variable region of the Ig gene is amplified by reverse transcription (in the case of RNA) and PCR, using primers as described above (see "The choice of primer sets" subsection). The primers include overhanging extensions with UMIs, sample barcodes, annealing sites for the sequencing primers, and adaptors for the sequencer. Usually, two PCR steps are required, due to the length of overhang extensions on the primers, and sometimes the need to perform nested PCR to increase specificity. Another possibility to add the above extensions is by ligation. After verification of product size and quality, e.g. by tapestation or bioanalyzer analysis, the concentration is adjusted to fit the sequencer requirements, and the library is ready for sequencing (Fig. 2).

A specific problem for Ig repertoire HTS is the highly similar nature of the Ig sequences, which leads to failure of the Illumina sequencer to distinguish neighboring clusters. This can be alleviated by spiking the desired library with a control phiX virus library, and/or replacing each primer with a mixture of 4 primers, where the body of the primer is identical except for 0–3 extra random nucleotides that are added at the beginning of it.

Ig Repertoire Sequence Data Pre-Processing, Annotation and Clonal Assignment

The theoretical diversity of Igs is immense. Igs are combined of a light and a heavy chain, of varying lengths. For example, a typical heavy chain is between 330 and 390 base pairs, so naively, one could think that there are $4^{330-390}$ possible combinations for this sequence. Structural constraints reduce this number dramatically, but still the estimated diversity of the heavy chain only is $\sim 10^9$ (Murphy and Weaver, 2016).

Pre-Processing

Many non-Ig HTS applications assume that the sequenced amplicons can be mapped and compared to a reference genome (Michaeli *et al.*, 2012). Due to the somatic rearrangement discussed above, a B cell clone creates its unique Ig sequence, with no

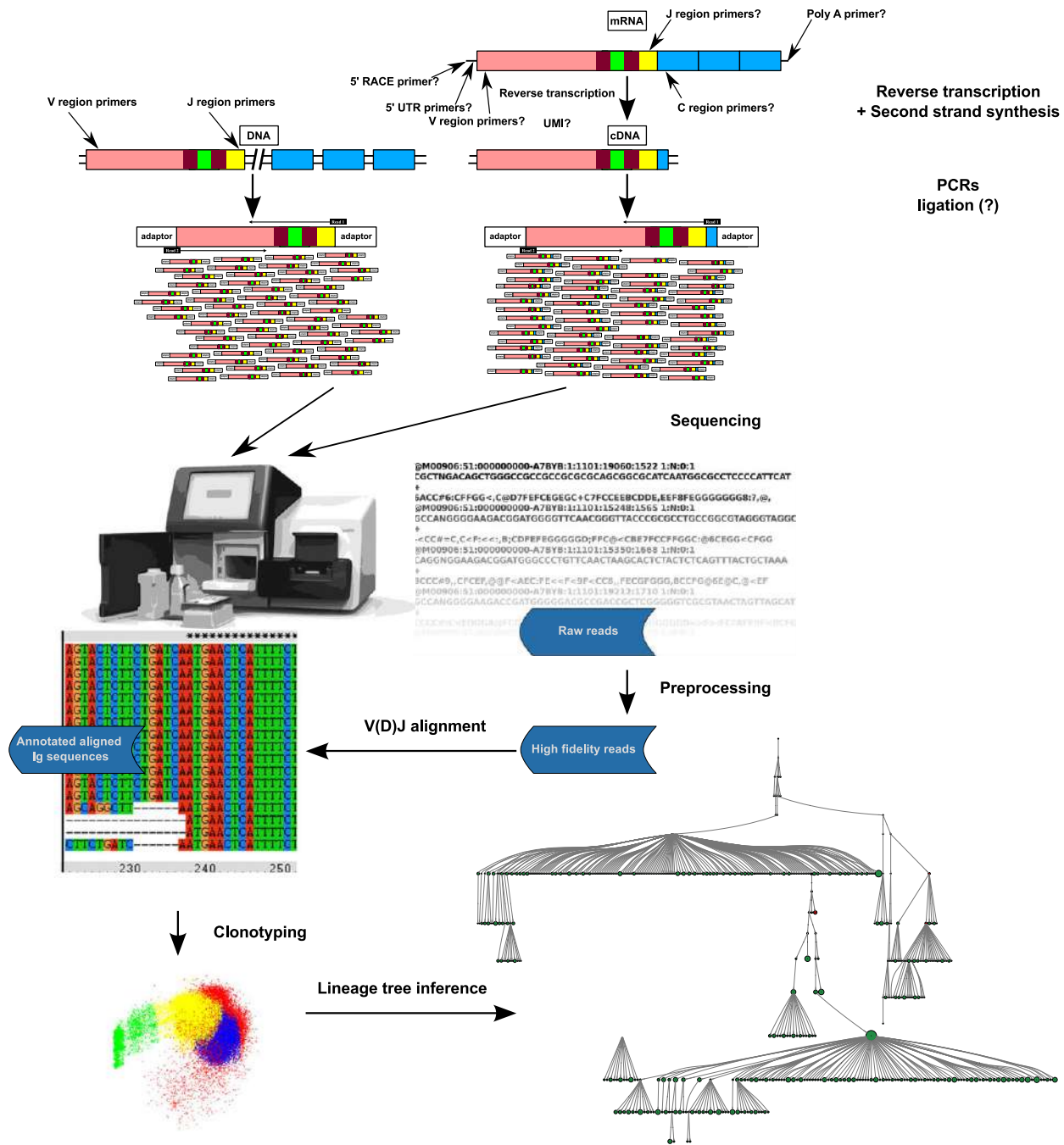


Fig. 2 The process of Ig repertoire sequencing and analysis. Starting from either RNA (top right) or DNA (top left), the variable region of the Ig gene is enriched and amplified, and adaptors are added for HTS. The raw sequencing reads undergo preprocessing to filter out irrelevant and low-quality sequences, and the high-fidelity reads are aligned to the known reference genes, annotated and clonotyped, followed by lineage tree construction.

complete reference in the genome, and further diversifies it through SHM. Thus, tailored tools for preprocessing Ig repertoires sequencing data had to be developed by the computational immunology community (see, e.g., Georgiou *et al.*, 2014; Vander Heiden *et al.*, 2014; Michaeli *et al.*, 2012, 2013; Moorhouse *et al.*, 2014; Schaller *et al.*, 2015; Wardemann and Busse, 2017; Yaari and Kleinstei, 2015). First, Ig repertoire sequencing data require a large amount of pre-processing work to clean out the data and prepare them for more specific downstream analysis related to clonal identification and mutation analysis. Specific computational toolkits tailored for Ig repertoires sequencing purposes perform pre-processing of raw sequencing reads and produce error-corrected, sorted and annotated sequence sets, along with a wealth of quality control metrics (Bolotin *et al.*, 2015; Vander Heiden *et al.*, 2014; Michaeli *et al.*, 2012). These tools include, for example: a. Removal of low-quality reads. b. Removal of reads

where the primer could not be identified or had a poor alignment score. c. Identification of sets of sequences with identical UMIs. These are collapsed into one consensus sequence per set. d. Assembly of the two consensus paired-end reads into a complete Ig sequence. e. Removal of sequences that do not appear in a single sample with at least two independent molecular IDs (see also Stern *et al.*, 2014).

Annotation

Ig repertoire analysis benefits from adequate annotations. For example, each Ig sequence is derived from a combination of V, D, and J germline genes, which could be inferred from the Ig sequence using different tools (Bolotin *et al.*, 2015; Brochet *et al.*, 2008; Gaëta *et al.*, 2007; Munshaw and Kepler, 2010; Ralph and Matsen, 2016; Ye *et al.*, 2013). These VDJ assignment tools return for each Ig an aligned germline sequence, constructed from the inferred V, D, and J genes, taking into account the possibilities for trimming the edges of these genes and inserting alien nucleotides in between them (N and P additions) ((Murphy and Weaver, 2016), Fig. 1). Germline sequence annotations reduce the dimensionality of the sequence space, and enable a more comprehensible and interpretable analysis. Furthermore, and most importantly, somatic mutations can be identified from the germline sequence, and be used for phylogenetic tree construction, selection quantification, and building somatic mutation models. To identify the V and J genes, methods based on sequence similarity are commonly used (Lefranc *et al.*, 2009; Ye *et al.*, 2013). The inference of the D gene and its flanking sequences is a more difficult task, due to the stochastic processes underlying VDJ recombination and the relatively short lengths of the D genes. Several statistical frameworks to cope with this stochastic process exist, including HMM (Gaëta *et al.*, 2007; Munshaw and Kepler, 2010) and an E-M iterative process that first estimates the probabilities for each event from the whole dataset, and then aligns the sequence to the inferred germline (Marcou *et al.*, 2017).

Germline Repertoire Inference

Another important point in VDJ assignment is the ability to a priori set the V, D, and J, germline repertoires. Studying polymorphisms in the Ig loci in human populations is an active field. Recent studies suggest that there are complex events of deletion and duplication in these loci, yielding an undiscovered wealth of unknown alleles (Kidd *et al.*, 2012; Luo *et al.*, 2016; Wang *et al.*, 2008; Watson and Breden, 2012; Watson *et al.*, 2013). IMGT has a central repository for V, D, and J allele sequences; to add new sequences to this repertoire there is a need for direct DNA sequencing of non B cells. In recent years, several independent methods to detect novel polymorphisms from Ig repertoires have been developed (Corcoran *et al.*, 2016; Gadala-Maria *et al.*, 2015), and there is an ongoing effort to define a set of rules that will allow recognition by the community of these inferred alleles (Brenden *et al.*, 2017). Comparing Ig sequences to all known alleles can result in erroneous gene assignments, because alleles might be missing in current germline gene datasets, and since highly mutated sequences may be mis-assigned. To reduce this risk, a genotype and/or a haplotype step that generates a personalized pool of possible alleles can be performed. In a genotype step, all sampled sequences from an individual are combined, and a decision is made about the set of alleles that the individual actually has. After creating such a genotype, VDJ assignment is repeated with the restricted set of available germline genes. Such a step can reduce the number of mis-assigned sequences dramatically (Gadala-Maria *et al.*, 2015). To haplotype a repertoire, one can use J (or D) gene heterozygosity and analyze the frequency of recombined J (or D) alleles with V alleles (Kidd *et al.*, 2012). After inferring a haplotype for an individual, VDJ assignment should be repeated to ensure that alleles are co-assigned to a sequence only if they are present on the same chromosome. After the VDJ assignment step, each sequence is annotated with its inferred germline genes, trimmed and added sequences, CDR3 length, charge, etc.

Clonal Assignment

One of the leading themes behind our understanding of adaptive immunity is clonal selection, and the idea that T and B cell repertoires are comprised of groups of cells of varying sizes, each of which derived from a common ancestor (Hodgkin *et al.*, 2007). While in T cells it is straightforward to identify these clonally-related sequences, as they are identical in all the cells belonging to the same clone, in B cells, due to SHM, inferring these groups from Ig repertoire sequencing data is a statistically non-trivial task (Chen *et al.*, 2010; Hershberg and Luning Prak, 2015). Since the number of sequences in a typical Ig repertoire sequencing experiment is in the order of millions, clustering these sequences requires tailored approaches from machine learning and statistics. Clones are usually inferred based on their heavy chain sequences, as light chain diversity alone is too low for this task (Saada *et al.*, 2007). When heavy and light chain sequences are paired (Dekosky *et al.*, 2013, 2014), the combined information can be used for clonal assignment.

Clonal assignment is normally performed in two steps. First, sequences are grouped based on their V-J gene calls and junction length annotations. Second, a sequence-based distance measure is used to cluster the sequences inside each group. Due to the nature of SHM, the common distance metric is based on nucleotide similarity, and in most cases is applied to the CDR3 part of the sequence. In principle, it is possible to cluster the sequences based on sequence similarity throughout the whole variable region, and also allow for insertions and deletions that can occur during SHM (Kepler *et al.*, 2014a). However, in most studies these are not considered. It is also possible to use a distance metric that is based on a mutability model that takes into account hot and cold spots of SHM (Elhanati *et al.*, 2015; Yaari *et al.*, 2013).

After constructing a distance matrix, clustering can be performed using approaches such as imposing a threshold on a hierarchical tree using either single, complete or average linkage. To determine the adequate threshold, a “distance-to-nearest” plot (Glanville *et al.*, 2011) can be utilized. Since biological clones are restricted to a single individual, inferred clones that span multiple individuals can be used to estimate specificity for clonal assignment. With these assumptions, it was shown that “hamming distance” using single linkage clustering performs better in terms of specificity and sensitivity (Gupta *et al.*, 2015; Yermanos *et al.*, 2017). Alternatives to the distance matrix clustering include cutting lineage trees (see below) to create subtrees that can be interpreted as clones (Lieberman *et al.*, 2013), and maximum likelihood approaches (Kepler *et al.*, 2014a; Wu *et al.*, 2011). Initial V(D)J assignments can be refined after clonal assignments as all sequences stemmed from the same cell have the same germline (Kepler, 2013; Stern *et al.*, 2014). Although V(D)J assignments are not required for clonal assignment (Giraud *et al.*, 2014), utilizing these annotations makes it possible to cluster much larger data sets with standard clustering methods.

Grouping similar sequences can also serve other purposes. Similar receptor sequences could indicate similarity in function. For example, the similarity of TCRs was recently shown to help in predicting epitopes (Dash *et al.*, 2017; Glanville *et al.*, 2017). Functionally similar Ig sequences across individuals indicate “public” immune responses (Glanville *et al.*, 2011). These sequences can also indicate convergent evolution where multiple clones produce independently similar Ig sequences at the amino acid level, but not necessarily using the same genes (Jackson *et al.*, 2014; Parameswaran *et al.*, 2013). For these applications, sequence similarity at the amino acid level can be used for clustering.

Repertoire Analyses

Common Repertoire Measures and Coping With Inherent Biases

Ig repertoires are characterized by many features that allow researchers to reduce their high dimensionality and describe them in biologically meaningful terms. Examples include: V-D-J gene usage distribution, isotype usage distribution, various measures of repertoire diversity and similarity, CDR3 length distribution, amino acid composition, and mutation patterns. The interpretation of these distributions and measures depends on various factors, such as the sequenced material (mRNA or DNA), whether a UMI approach was taken, and whether bulk or single cell sequencing was done (see Table 1). For example, amplification biases can dramatically influence the results if each sequence is counted independently and UMIs are not utilized. Even if UMIs are utilized, the measured distribution will describe the mRNA distribution, which can be very misleading as some cells (plasmablasts) have orders of magnitude higher numbers of mRNA molecules than B cells. A common way to confront these biases is to construct the distributions from the set of unique sequences. However, this approach still cannot overcome the experimental artifacts, such as transcription and amplification errors. Yet, a repetitive important task in Ig repertoire analysis is to calculate the above feature distributions and conclude from them something about repertoire generation and dynamics. For example, using the D-J genes distribution to infer the inherent constraints and preferences of the V-D-J recombination machinery. To do this, researchers commonly pick one sequence from every clone and compute the distributions from these representatives. On the other hand, taking only one representative from every clone leaves most of the data unanalyzed. To exploit more information from the data, it is necessary to also integrate the construction of lineage trees into the analysis, as discussed below.

In a healthy repertoire, large clones are rare, and we usually sample only one or a few cells per clone (Hershberg and Luning Prak, 2015). Even during an acute infection, only a few responding clones grow and may constitute at most a few percent of the repertoire. In contrast, if the clonal size distribution contains one or more highly frequent clones, we may suspect the presence of an autoimmune disease such as Sjogren’s syndrome (Hershberg *et al.*, 2014) systemic lupus erythematosus following rituximab therapy (Sfikakis *et al.*, 2009), or a B cell malignancy such as chronic lymphocytic leukemia (Logan *et al.*, 2011). Positive selection for independent B cell clones often results in shared amino acid sequence motifs that bind to particular antigens. Such public CDR3 motifs have been documented in many chronic disorders ranging from viral (Jackson *et al.*, 2014; Wrammert *et al.*, 2011) or bacterial (Scott *et al.*, 1989; Silverman and Lucas, 1991) infections and autoimmunity to cancer.

Repertoire Diversity

Repertoires may differ from one another in many ways. Quantifying these differences is a research field of its own, and includes many possibilities to assess diversity and similarity of repertoires. One potent approach to measure diversity is the Hill diversity index (Hill, 1973). This index can be applied to any feature distribution (e.g. V gene usage), and usually is applied to characterize the clone size distribution. It is defined as

$$qD = \left(\sum_{i=1}^{i=n} p_i^q \right)^{(1/(1-q))}$$

where the sum is over all clones and p_i is the fraction of clone i out of the entire population. This diversity index converges to all of the known diversity measurements for different values of the parameter q . For example, for $q = 0$, D is the number of species (a.k.a. richness); for $q = \infty$, D is the reciprocal of the fraction of the largest clone; for $q = -\infty$, D is the reciprocal of the fraction of the smallest clone; for $q = 1$, D is the exponent of Shannon’s entropy, and higher values of q (2, 3 etc.) give a higher weight to the larger clones. Studying the shape of this curve is much more informative than measuring a single diversity score along with an

“evenness” score (Hill, 1973). When comparing diversity profiles between samples, it is advised to subsample (with replacement) the larger repertoire to the size of the smaller one and only then to compare the two (Gupta *et al.*, 2015).

Lineage Tree Analysis & Lessons Learned From Tree Structure

During affinity maturation, Ig loci accumulate mutations via SHM. An important step in Ig repertoire analysis is the inference of each clone’s pedigree. These pedigrees are referred to as lineage trees, and methods to infer them are usually borrowed from the field of phylogeny, despite different assumptions underlying each case. For example, in B cell lineages, the trees are rooted by the rearranged, pre-mutation sequence. This sequence is not necessarily the most recent common ancestor of the tree, but is used to construct the tree.

Lineage Tree Construction

The phylogenetic methods most commonly used for analysis of Ig repertoire sequencing data are maximum likelihood, maximum parsimony, neighbor joining (Barak *et al.*, 2008) and Bayesian inference (Yermanos *et al.*, 2017). These methods utilize a distance metric to construct the trees. The most common distance metrics are the Hamming, Kimura substitution model (Kimura, 1968), and Levenshtein distances. Since SHM preferences are based on neighboring nucleotides, a more adequate distance metric should take into account these neighboring nucleotides. A first step to integrate neighboring bases effects in constructing a B cell lineage tree was taken recently (Hoehn *et al.*, 2017). It was evaluated using trees constructed from HIV patients (Wu *et al.*, 2015), and showed better performance. A method based on natural SHM metrics, such as the S5F (Yaari *et al.*, 2013) or 7mers (Elhanati *et al.*, 2015) is still lacking. Given a distance metric, the differences in performance between the methods are mild (Yermanos *et al.*, 2017). In order to use the outputs of lineage tree construction algorithms, there is a need in parsing the resulting trees and fit them to Ig repertoires (Barak *et al.*, 2008; Vander Heiden *et al.*, 2015), as most phylogenetic algorithms assume that all observed sequences are at the leaves of the tree. Also, in B cell lineages the germline sequence should be upstream of the root of the tree.

Lessons Learned From Lineage Trees

Merely constructing Ig lineage trees sheds light on fundamental processes underlying the immune system in health and disease, and may aid in vaccine and drug design. We present here a few examples, which are by no means an exhaustive set. Tabibian-Keissar *et al.* (Tabibian-Keissar *et al.*, 2008) used lineage trees to give the first direct evidence of trafficking between human colon and adjacent lymph nodes, Bergqvist *et al.* (Bergqvist *et al.*, 2013) used trees to show the spreading and synchronization of IgA responses throughout the murine gut, and Pabst *et al.* (Pabst *et al.*, 2015) used trees of sequences from serial mouse gut biopsy samples to show the development of the murine gut immune system. Hazanov *et al.* (Hazanov *et al.*, 2015) have used lineage tree structures to elucidate the relationships between switched memory, IgM memory, naïve and IgD-CD27 – (double-negative, “DN”) B cells in humans. Meng *et al.* (Meng *et al.*, 2017), with the aim of enhancing our understanding of the distribution of B cell clones in the human body, sequenced and analyzed over 38 million B cells from eight anatomic compartments. They found two major networks of large clones, one in the blood, bone marrow, spleen and lung, and another in the GI tract, with little overlap between the networks. The GI tract contained large clones with the highest levels of somatic hypermutation.

SHM and class switching

Horns *et al.* (Horns *et al.*, 2016) developed a way to reconstruct the B cell proliferation process and thereby trace the lineage of individual B cells, using somatic hypermutations as a molecular clock. They revealed that closely related B cells often switch to the same class, but lose coherence as somatic mutations accumulate. They also found that naïve classes (IgM or IgD) accumulated more mutations before undergoing class switch recombination to activated classes (IgG, IgA, or IgE), in comparison with class switch recombination between activated classes. These findings lay a foundation for developing techniques to direct Ig class switching. Kepler *et al.* (Kepler *et al.*, 2014b) analyzed paired heavy-light lineage trees of an influenza-infected individual, and showed how affinity maturation occurs stepwise with time within clones, increasing 1000-fold from the unmutated ancestor to the highest affinity observed Igs.

Aging

Jiang *et al.* (Jiang *et al.*, 2013) and the Dunn-Walters lab (Ademokun *et al.*, 2011; Tabibian-Keissar *et al.*, 2016; Wu *et al.*, 2012) determined the lineage structure of the Ig repertoire before and after influenza vaccination, and observed age related changes. Elderly subjects had fewer lineages than other age groups, both before and after vaccination, and influenza vaccination resulted in expansion of far fewer B cell lineages and a reduced B cell clonal diversity in the elderly compared to the younger age groups. The elderly had a higher percentage of somatic mutations, likely since their clonal expansions draw upon a pool of B cells having more somatic mutations to begin with.

Infection and response dynamics

Highly effective, broadly neutralizing Igs towards HIV have unusual characteristics (Burton *et al.*, 1994; Calarese *et al.*, 2003; Haynes *et al.*, 2005; Huang *et al.*, 2004; Klein *et al.*, 2013; Pancera *et al.*, 2010; Pejchal *et al.*, 2010; Scheid *et al.*, 2011; Walker *et al.*,

2009, 2011; Wu *et al.*, 2010a, 2011; Zhou *et al.*, 2007) and usually take years of chronic infection to develop (Gray *et al.*, 2011; Wu *et al.*, 2006). The best studied broadly neutralizing Ig lineages are VRC01, VRC26, and CH103 (Doria-Rose *et al.*, 2014; Liao *et al.*, 2013; Wu *et al.*, 2015). Wu *et al.* (Wu *et al.*, 2015) investigated VRC01, which targets gp120, the site of CD4 engagement on HIV-1. They identified and described Ig lineage characteristics over the course of 15 years in a single patient. Most Igs differ from the germline by ~5%, but the VRC01 lineage is over 30% different. Over the course of the study, the authors showed using lineage trees that VRC01 Igs continuously evolved, with a rate of ~2 substitutions per 100 nucleotides per year, comparable to that of HIV-1 evolution. Liao *et al.* (Liao *et al.*, 2013) followed the CH103 Ig lineage, also targeting gp120, in a single donor over 3 years. They showed the evolution of the lineage up to a mature broadly neutralizing Ig that could neutralize ~55% of HIV-1 isolates. This lineage is less mutated than other CD4-binding Igs, and may be first detectable as early as 14 weeks after infection. Therefore, the CD4-binding site may be a vaccine target. Doria-Rose *et al.* (Doria-Rose *et al.*, 2014) delineated longitudinal interactions between the development of 12 related broadly neutralizing Igs and HIV-1 within a single donor over 4 years. They found that the unmutated ancestor of the lineage emerged between weeks 30–38 post-infection, bound and neutralized the virus weakly and did not bind any other HIV strains. Later variants demonstrated increasing affinity towards several strains of the virus. Subsequent affinity maturation focused within CDR H3 and allowed for progressively greater binding and neutralization. This work defined the molecular requirements and genetic pathways leading to virus neutralization, providing a template for their vaccine elicitation. Sok *et al.* (Sok *et al.*, 2013) developed a novel method called ImmuniTree, which is an alternative approach to conventional phylogenetic analyses and is designed specifically to model Ig somatic hypermutations. They used this method to study the PGT121-134 broadly neutralizing Ig lineage, which is also characterized by high levels of somatic mutations and is among the most potent lineages described to date. They discovered Igs in the lineage with half the mutation level of PGT121-134, capable of neutralizing 40%–80% of PGT121-134 sensitive viruses. Such Igs will likely be easier to induce through vaccination.

Autoimmune diseases

Autoimmune responses also generate large clones, and various insights can be gleaned from analysis of the corresponding lineage trees. Stern *et al.* (Stern *et al.*, 2014) and Palanichamy *et al.* (Palanichamy *et al.*, 2014) analyzed B cell repertoires from the brain and lymph nodes of multiple sclerosis patients, and showed that B cells populating the multiple sclerosis brain mature in the draining cervical lymph nodes, and that the traffic between the periphery and the brain is an ongoing process, and not a one-time event as was previously thought. This provides insight into the trafficking of B cells in MS. Therefore, modulation of specific B cell subsets could provide therapeutic benefit in MS patients. Snir *et al.* (Di Niro *et al.*, 2016; Snir *et al.*, 2015) compared B cell clones in the gut and blood of celiac patients, and found that the generation of plasma cells producing autoantibodies likely occurs outside the gut. Such knowledge helps to shed light on the molecular mechanisms causing the disease, and may facilitate the development of immune-based therapies.

Mutation Analyses Relying on Lineage Tree Structure

Lineage tree graphical properties can be used by themselves to estimate the relative strength of selection, mutation and initial competitive advantage of clones (Shahaf *et al.*, 2008), and thus to characterize a specific immune response. However, the most useful feature of the trees is the mutations they are constructed from. Analyzing mutation patterns in Ig repertoires can shed light on the underlying biology of B cell adaptation, and can be used to estimate affinity dependent selection. For any analysis of mutation patterns, reliable lineage trees should be used to identify independent events of mutations. If one ignores the lineage structure of a B cell clone, mutations can be counted multiple times as an artificially larger number of independent events. This may lead to severe biases in mutation analyses. Integrating lineage tree structure in the analysis should thus be used to count each mutation event only once. In addition, tree structure enables more correct identification of each mutation (as each sequence is compared to its most recent identifiable ancestor, and not to the root), and the identification of reversion mutations.

The patterns of mutations carry the fingerprints of the SHM machinery – from AID targeting sites (Chandra *et al.*, 2015) to the various pathways to deamination site resolution (Zanotti and Gearhart, 2016). In most of our studies, whether the focus was aging, chronic or autoimmune diseases or B cell malignancies, we have found no differences from controls in mutation statistics such as targeting motifs, transition: transversion ratios, the fraction of mutations from and to each nucleotide, etc. A noteworthy exception was seen in studying ectopic GC from myasthenia gravis patient thymi. Using tree-based mutation analysis, Zuckerman *et al.* have predicted changes in the SHM mechanism (Zuckerman *et al.*, 2010b), a prediction which was verified by gene expression analysis.

Selection analysis has many applications, including identification of potentially high affinity sequences, understanding how different genetic manipulations impact affinity maturation, and investigating whether disease processes are antigen-driven. It is common to look for an increased frequency of non-synonymous mutations as evidence of antigen-driven positive selection, and a decreased frequency of non-synonymous mutations as evidence of negative selection (Hershberg *et al.*, 2008; Uduman *et al.*, 2011; Yaari *et al.*, 2012). Expectations are calculated either using a uniform targeting model in which all mutations occur with the same rate (Lossos *et al.*, 2000; Shlomchik *et al.*, 1987), or using a mutability model such as the S5F (Yaari *et al.*, 2013).

When quantifying selection in Ig sequences, mutations can be calculated in several time scales. An Ig sequence can be compared to the germline sequence, to the most recent common ancestor, or to the sequence that is one branch upstream in the tree. These different comparisons yield different selection estimations. Lineage trees were used to show how selection is stronger along the tree

vs. close to its leaves (Uduman *et al.*, 2014). It was also shown that memory cells from past infections experienced stronger selection compared to current clonal expansion (Yaari *et al.*, 2015).

Summary and Challenges for the Future

The analysis of TCR and BCR clonal repertoires differs from other bioinformatic practices due to the lack of reference genes, and with the amounts of data generated by HTS, is more complex. On the other hand, as shown by the examples shown above, it has an enormous potential for generating new insights into the structure and activity of the immune system at every stage of life and when faced with any challenge. We eagerly anticipate the day in which knowing a person's BCR and TCR "haplotypes" and having a good sample of their repertoire would give us as much knowledge of their immune status as the rest of their genome can tell about their health status and potential.

Before we reach that day, however, there are several challenges to overcome. First, we do not have a full understanding of the structure and contents of human BCR and TCR genomic loci; better knowledge of these genomic regions, in particular the identification of larger numbers of alleles, will accumulate as the repertoires from more and varied population groups are sequenced. Second, for sequencing of such magnitude to be possible, we need ways to generate longer, more reliable reads at far lower cost. Third, in order to understand the function of the BCRs and TCRs sequenced, we need a reliable, easy and low-cost protocol for sequencing paired heavy and light chains or beta and alpha chains. While emulsion-based methods are much less work-intensive and have higher throughputs than well plate-based methods, paired sequencing is still basically a single-cell method of analysis and hence its current output is not as high as single-chain sequencing. Fourth, in order to compare and evaluate studies and enable meta-analysis, we must standardize protocols and data format between labs. Research in the field will be facilitated by data and protocol sharing and adherence to meticulous standards (see Relevant Website Section). Finally, we should devote considerable efforts to the continued invention of new and creative ways to look at the data collected. We need ways to obtain a deeper view of the true repertoire, as even the new methods only sample a small portion of it.

Acknowledgement

This research was supported by the Israel Science Foundation (grant No. 832/16).

See also: Computational Immunogenetics. Epitope Predictions. Extraction of Immune Epitope Information. Immunoinformatics Databases. Natural Language Processing Approaches in Bioinformatics. Vaccine Target Discovery

References

- Ademokun, A., Wu, Y.-C., Martin, V., *et al.*, 2011. Vaccination-induced changes in human B-cell repertoire and pneumococcal IgM and IgA antibody at different ages. *Aging Cell* 10, 922–930.
- Arnaout, R., Lee, W., Cahill, P., *et al.*, 2011. High-resolution description of antibody heavy-chain repertoires in humans. *PLOS ONE* 6, e22365.
- Barak, M., Zuckerman, N.S., Edelman, H., Unger, R., Mehr, R., 2008. IgTree: Creating Immunoglobulin variable region gene lineage trees. *J. Immunol. Methods* 338, 67–74.
- Benichou, J., Ben-Hamo, R., Louzoun, Y., Efroni, S., 2012. Rep-Seq: Uncovering the immunological repertoire through next-generation sequencing. *Immunology* 135, 183–191.
- Bergqvist, P., Stensson, A., Hazanov, L., *et al.*, 2013. Re-utilization of germinal centers in multiple Peyer's patches results in highly synchronized, oligoclonal, and affinity-matured gut IgA responses. *Mucosal Immunol.* 6, 122–135.
- Bolotin, D.A., Poslavsky, S., Mitrophanov, I., *et al.*, 2015. MiXCR: Software for comprehensive adaptive immunity profiling. *Nat. Methods* 12, 380–381.
- Boyd, S.D., Gaëta, B.A., Jackson, K.J., *et al.*, 2010. Individual variation in the germline Ig gene repertoire inferred from variable region gene rearrangements. *J. Immunol.* 184, 6986–6992.
- Boyd, S.D.S., Marshall, E.L., Merker, J.D., *et al.*, 2009. Measurement and clinical monitoring of human lymphocyte clonality by massively parallel VDJ pyrosequencing. *Sci. Transl. Med.* 1, 12ra23.
- Breden, F., Luning Prak, E.T., Peters, B., *et al.*, 2017. Reproducibility and reuse of adaptive immune receptor repertoire data. *Front. Immunol.* 8, Article 1418.
- Briggs, A.W., Goldfless, S.J., Timberlake, S., *et al.*, 2017. Tumor-infiltrating immune repertoires captured by single-cell barcoding in emulsion. *BioRxiv*.
- Brochet, X., Lefranc, M.-P., Giudicelli, V., 2008. IMGT/V-QUEST: The highly customized and integrated system for IG and TR standardized V-J and V-D-J sequence analysis. *Nucleic Acids Res.* 36, W503–W508.
- Burton, D.R., Pyati, J., Koduri, R., *et al.*, 1994. Efficient neutralization of primary isolates of HIV-1 by a recombinant human monoclonal antibody. *Science* 266, 1024–1027.
- Busse, C.E., Czogiel, I., Braun, P., Arndt, P.F., Wardemann, H., 2014. Single-cell based high-throughput sequencing of full-length immunoglobulin heavy and light chain genes. *Eur. J. Immunol.* 44, 597–603.
- Calarese, D.A., Scanlan, C.N., Zwick, M.B., *et al.*, 2003. Antibody domain exchange is an immunological solution to carbohydrate cluster recognition. *Science* 300, 2065–2071.
- Cameron, E.M., Spencer, S., Lazarini, J., *et al.*, 2009. Potential of a unique antibody gene signature to predict conversion to clinically definite multiple sclerosis. *J. Neuroimmunol.* 213, 123–130.
- Campbell, P.J., Pleasance, E.D., Stephens, P.J., *et al.*, 2008. Subclonal phylogenetic structures in cancer revealed by ultra-deep sequencing. *Proc. Natl. Acad. Sci. USA.* 105, 13081–13086.
- Chandra, V., Bortnick, A., Murre, C., 2015. AID targeting: Old mysteries and new challenges. *Trends Immunol.* 36, 527–535.
- Chen, Z., Collins, A.M., Wang, Y., Gaëta, B.A., 2010. Clustering-based identification of clonally-related immunoglobulin gene sequence sets. *Immunome Res.* 6, S4.
- Corcoran, M.M., Phad, G.E., Vázquez Bernat, N., *et al.*, 2016. Production of individualized V gene databases reveals high levels of immunoglobulin genetic diversity. *Nat. Commun.* 7, 13642.
- Dash, P., Fiore-Gartland, A.J., Hertz, T., *et al.*, 2017. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* 547, 89–93.

- Dekosky, B.J., Ippolito, G.C., Deschner, R.P., *et al.*, 2013. High-throughput sequencing of the paired human immunoglobulin heavy and light chain repertoire. *Nat. Biotechnol.* 31, 166–169.
- Dekosky, B.J., Kojima, T., Rodin, A., *et al.*, 2014. In-depth determination and analysis of the human paired heavy- and light-chain antibody repertoire. *Nat. Med.* 21, 86–91.
- Di Niro, R., Snir, O., Kaukinen, K., *et al.*, 2016. Responsive population dynamics and wide seeding into the duodenal lamina propria of transglutaminase-2-specific plasma cells in celiac disease. *Mucosal Immunol.* 9, 254–264.
- Doria-Rose, N.A., Schramm, C.A., Gorman, J., *et al.*, 2014a. Developmental pathway for potent V1V2-directed HIV-neutralizing antibodies. *Nature* 509, 55–62.
- Dunn-Walters, D.K., Ademokun, A.A., 2010. B cell repertoire and ageing. *Curr. Opin. Immunol.* 22, 514–520.
- Dunn-Walters, D.K., Banerjee, M., Mehr, R., 2003. Effects of age on antibody affinity maturation. *Biochem. Soc. Trans.* 31, 447–448.
- Elhanati, Y., Sethna, Z., Marcou, Q., *et al.*, 2015. Inferring processes underlying B-cell repertoire diversity. *Philos. Trans. R. Soc. B Biol. Sci.* 370, 20140243.
- Fridman, W.H., Pagès, F., Sautès-Fridman, C., Galon, J., 2012. The immune contexture in human tumours: Impact on clinical outcome. *Nat. Rev. Cancer* 12, 298–306.
- Gadala-Maria, D., Yaari, G., Uduman, M., Kleinstein, S.H., 2015. Automated analysis of high-throughput B-cell sequencing data reveals a high frequency of novel immunoglobulin V gene segment alleles. *Proc. Natl. Acad. Sci. USA.* 112, E862–E870.
- Gaëta, B.A., Malming, H.R., Jackson, K.J.L., *et al.*, 2007. iHMM-align: Hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. *Bioinformatics* 23, 1580–1587.
- Galon, J., Costes, A., Sanchez-Cabo, F., *et al.*, 2006. Type, density, and location of immune cells within human colorectal tumors predict clinical outcome. *Science* 313, 1960–1964.
- Georgiou, G., Ippolito, G.C., Beausang, J., *et al.*, 2014. The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nat. Biotechnol.* 32, 158–168.
- Giraud, M., Salson, M., Duez, M., *et al.*, 2014. Fast multiclonal clusterization of V(DJ) recombinations from high-throughput sequencing. *BMC Genomics* 15, 409.
- Glanville, J., Kuo, T.C., Büdingen, H.-C., *et al.*, 2011. Naive antibody gene-segment frequencies are heritable and unaltered by chronic lymphocyte ablation. *Proc. Natl. Acad. Sci. USA.* 108, 20066–20071.
- Glanville, J., Huang, H., Nau, A., *et al.*, 2017. Identifying specificity groups in the T cell receptor repertoire. *Nature* 547, 94–98.
- Goodwin, S., McPherson, J.D., McCombie, W.R., 2016. Coming of age: Ten years of next-generation sequencing technologies. *Nat. Rev. Genet.* 17, 333–351.
- Gray, E.S., Madiga, M.C., Hermanus, T., *et al.*, 2011. The neutralization breadth of HIV-1 develops incrementally over four years and is associated with CD4+ T cell decline and high viral load during acute infection. *J. Virol.* 85, 4828–4840.
- Gupta, N.T., Vander Heiden, J.A., Uduman, M., *et al.*, 2015. Change-O: A toolkit for analyzing large-scale B cell immunoglobulin repertoire sequencing data: Table 1. *Bioinformatics* 31, 3356–3358.
- Haynes, B.F., Fleming, J., St Clair, E.W., *et al.*, 2005. Cardioliipin polyspecific autoreactivity in two broadly neutralizing HIV-1 antibodies. *Science* 308, 1906–1908.
- Hazanov, L., Mehr, R., Wu, Y.-C.B., Dunn-Walters, D.K., 2015. Lineage tree analysis of high throughput immunoglobulin sequencing clarifies B cell maturation pathways In: 2015 International Workshop on Artificial Immune Systems (AIS). pp. 1–6. IEEE.
- Hershberg, U., Luning Prak, E.T., 2015. The analysis of clonal expansions in normal and autoimmune B cell repertoires. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 370.
- Hershberg, U., Meng, W., Zhang, B., *et al.*, 2014. Persistence and selection of an expanded B cell clone in the setting of rituximab therapy for Sjogren's syndrome. *Arthritis Res. Ther.* 16, R51.
- Hershberg, U., Uduman, M., Shlomchik, M.J., Kleinstein, S.H., 2008. Improved methods for detecting selection by mutation analysis of Ig V region sequences. *Int Immunol* 20, 683–694.
- Hill, M.O., 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology* 54, 427.
- Hodgkin, P.D., Heath, W.R., Baxter, A.G., 2007. The clonal selection theory: 50 years since the revolution. *Nat. Immunol.* 8, 1019–1026.
- Hoehn, K.B., Lunter, G., Pybus, O.G., 2017. A phylogenetic codon substitution model for antibody lineages. *Genetics* 206, 417–427.
- Horns, F., Vollmers, C., Croote, D., *et al.*, 2016. Lineage tracing of human B cells reveals the in vivo landscape of human antibody class switching. *Elife* 5.
- Huang, C., Venturi, M., Majeed, S., *et al.*, 2004. Structural basis of tyrosine sulfation and VH-gene usage in antibodies that recognize the HIV type 1 coreceptor-binding site on gp120. *Proc. Natl. Acad. Sci. USA.* 101, 2706–2711.
- Jackson, K.J.L.L., Liu, Y., Roskin, K.M., *et al.*, 2014. Human responses to influenza vaccination show seroconversion signatures and convergent antibody rearrangements. *Cell Host Microbe* 16, 105–114.
- Jiang, N., He, J., Weinstein, J.A., *et al.*, 2013. Lineage structure of the human antibody repertoire in response to influenza vaccination. *Sci. Transl. Med.* 5, 171ra19.
- Jiang, N., Weinstein, J.A., Penland, L., *et al.*, 2011. Determinism and stochasticity during maturation of the zebrafish antibody repertoire. *Proc. Natl. Acad. Sci. USA.* 108, 5348–5353.
- Jiang, Y., Nie, K., Redmond, D., *et al.*, 2015. VDJ-Seq: Deep sequencing analysis of rearranged immunoglobulin heavy chain gene to reveal clonal evolution patterns of B cell lymphoma. *J. Vis. Exp.* e53215.
- Katoh, H., Komura, D., Konishi, H., *et al.*, 2017. Immunogenetic profiling for gastric cancers identifies sulfated glycosaminoglycans as major and functional B cell antigens in human malignancies. *Cell Rep.* 20, 1073–1087.
- Kepler, T.B., 2013. Reconstructing a B-cell clonal lineage I. Statistical inference of unobserved ancestors. *F000Research* 2, 103.
- Kepler, T.B., Liao, H.-X., Alam, S.M., *et al.*, 2014a. Immunoglobulin gene insertions and deletions in the affinity maturation of HIV-1 broadly reactive neutralizing antibodies. *Cell Host Microbe* 16, 304–313.
- Kepler, T.B., Munshaw, S., Wiehe, K., *et al.*, 2014b. Reconstructing a B-cell clonal lineage. II. Mutation, selection, and affinity maturation. *Front. Immunol.* 5, 170.
- Kidd, M.J., Chen, Z., Wang, Y., *et al.*, 2012. The inference of phased haplotypes for the immunoglobulin H chain V region gene loci by analysis of VDJ gene rearrangements. *J. Immunol.* 188, 1333–1340.
- Kimura, M., 1968. Evolutionary rate at the molecular level. *Nature* 217, 624–626.
- Klein, F., Diskin, R., Scheid, J.F., *et al.*, 2013. Somatic mutations of the immunoglobulin framework are generally required for broad and potent HIV-1 neutralization. *Cell* 153, 126–138.
- Krause, J.C., Tsibane, T., Tumpey, T.M., *et al.*, 2011. Epitope-specific human influenza antibody repertoires diversify by B cell intracolon sequence divergence and interclonal convergence. *J. Immunol.* 187, 3704–3711.
- Laserson, U., Vigneault, F., Gadala-Maria, D., *et al.*, 2014. High-resolution antibody dynamics of vaccine-induced immune responses. *Proc. Natl. Acad. Sci.* 111, 4928–4933.
- Lefranc, M.-P., Giudicelli, V., Ginestoux, C., *et al.*, 2009. IMGT, the international ImMunoGeneTics information system. *Nucleic Acids Res.* 37, D1006–D1012.
- Lehmann-Horn, K., Kronsbein, H.C., Weber, M.S., 2013. Targeting B cells in the treatment of multiple sclerosis: Recent advances and remaining challenges. *Ther. Adv. Neurol. Disord* 6, 161–173.
- Liao, H.-X., Lynch, R., Zhou, T., *et al.*, 2013. Co-evolution of a broadly neutralizing HIV-1 antibody and founder virus. *Nature* 496, 469–476.
- Lieberman, G., Benichou, J., Tsaban, L., Glanville, J., Louzoun, Y., 2013. Multi step selection in Ig H chains is initially focused on CDR3 and then on other CDR regions. *Front. Immunol.* 4, 274.
- Logan, A.C., Gao, H., Wang, C., *et al.*, 2011. High-throughput VDJ sequencing for quantification of minimal residual disease in chronic lymphocytic leukemia and immune reconstitution assessment. *Proc. Natl. Acad. Sci. USA.* 108, 21194–21199.
- Loman, N.J., Misra, R.V., Dallman, T.J., *et al.*, 2012. Performance comparison of benchtop high-throughput sequencing platforms. *Nat. Biotechnol.* 30, 434–439.
- Lossos, I.S., Okada, C.Y., Tibshirani, R., *et al.*, 2000. Molecular analysis of immunoglobulin genes in diffuse large B-cell lymphomas. *Blood* 95, 1797–1803.
- Luo, S., Yu, J.A., Song, Y.S., 2016. Genotyping allelic and copy number variation in the immunoglobulin heavy chain locus.

- Magnani, D.M., Silveira, C.G.T., Rosen, B.C., *et al.*, 2017. A human inferred germline antibody binds to an immunodominant epitope and neutralizes Zika virus. *PLoS Negl. Trop. Dis.* 11, e0005655.
- Mamedov, I.Z., Britanova, O.V., Zvyagin, I.V., *et al.*, 2013. Preparing unbiased T-Cell receptor and antibody cDNA libraries for the deep next generation sequencing profiling. *Front. Immunol.* 4, 456.
- Marcou, Q., Mora, T., Walczak, A.M., 2017. IGoR: A tool for high-throughput immune repertoire analysis.
- McDaniel, J.R., DeKosky, B.J., Tanno, H., Ellington, A.D., Georgiou, G., 2016. Ultra-high-throughput sequencing of the immune receptor repertoire from millions of lymphocytes. *Nat. Protoc.* 11, 429–442.
- Meng, W., Zhang, B., Schwartz, G.W., *et al.*, 2017. An atlas of B-cell clonal distribution in the human body. *Nat. Biotechnol.* 35, 879–884.
- Michaeli, M., Barak, M., Hazanov, L., Noga, H., Mehr, R., 2013. Automated analysis of immunoglobulin genes from high-throughput sequencing: Life without a template. *J. Clin. Bioinforma.* 3, 15.
- Michaeli, M., Noga, H., Tabibian-Keissar, H., Barshack, I., Mehr, R., 2012. Automated cleaning and pre-processing of immunoglobulin gene sequences from high-throughput sequencing. *Front. Immunol.* 3, 386.
- Moorhouse, M.J., van Zessen, D., IJsepeert, H., *et al.*, 2014. ImmunoGlobulin galaxy (IGGalaxy) for simple determination and quantitation of immunoglobulin heavy chain rearrangements from NGS. *BMC Immunol.* 15, 59.
- Munshaw, S., Kepler, T.B., 2010. SoDA2: A Hidden Markov Model approach for identification of immunoglobulin rearrangements. *Bioinformatics* 26, 867–872.
- Murphy, K., Weaver, C., 2016. *Janeway's Immunobiology*. Garland Science.
- Pabst, O., Hazanov, H., Mehr, R., 2015. Old questions, new tools: Does next-generation sequencing hold the key to unraveling intestinal B-cell responses? *Mucosal Immunol.* 8, 29–37.
- Palanichamy, A., Apeltsin, L., Kuo, T.C., *et al.*, 2014. Immunoglobulin class-switched B cells form an active immune axis between CNS and periphery in multiple sclerosis. *Sci. Transl. Med.* 6, 248ra106.
- Pancera, M., McLellan, J.S., Wu, X., *et al.*, 2010. Crystal structure of PG16 and chimeric dissection with somatically related PG9: Structure-function analysis of two quaternary-specific antibodies that effectively neutralize HIV-1. *J. Virol.* 84, 8098–8110.
- Parameswaran, P., Liu, Y., Roskin, K.M., *et al.*, 2013. Convergent antibody signatures in human dengue. *Cell Host Microbe* 13, 691–700.
- Patil, S.U., Ogunniyi, A.O., Calatroni, A., *et al.*, 2015. Peanut oral immunotherapy transiently expands circulating Ara h 2-specific B cells with a homologous repertoire in unrelated subjects. *J. Allergy Clin. Immunol.* 136, 125–134. [e12].
- Pejchal, R., Walker, L.M., Stanfield, R.L., *et al.*, 2010. Structure and function of broadly reactive antibody PG16 reveal an H3 subdomain that mediates potent neutralization of HIV-1. *Proc. Natl. Acad. Sci. USA.* 107, 11483–11488.
- Pogorelyy, M.V., Elhanati, Y., Marcou, Q., *et al.*, 2017. Persisting fetal clonotypes influence the structure and overlap of adult human T cell receptor repertoires. *PLOS Comput. Biol.* 13, e1005572.
- Prabakaran, P., Chen, W., Singarayan, M.G., *et al.*, 2012. Expressed antibody repertoires in human cord blood cells: 454 sequencing and IMGT/HighV-QUEST analysis of germline gene usage, junctional diversity, and somatic mutations. *Immunogenetics* 64, 337–350.
- Ralph, D.K., Matsen, F.A., 2016. Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. *PLOS Comput. Biol.* 12, e1004409.
- Rechavi, E., Lev, A., Lee, Y.N., *et al.*, 2015. Timely and spatially regulated maturation of B and T cell repertoire during human fetal development. *Sci. Transl. Med.* 7, 276ra25.
- Rechavi, E., Somech, R., 2017. Survival of the fetus: Fetal B and T cell receptor repertoire development. *Semin. Immunopathol.*
- Saada, R., Weinberger, M., Shahaf, G., Mehr, R., 2007. Models for antigen receptor gene rearrangement: CDR3 length. *Immunol. Cell Biol.* 85, 323–332.
- Schaller, S., Weinberger, J., Jimenez-Heredia, R., *et al.*, 2015. ImmunExplorer (IMEX): A software framework for diversity and clonality analyses of immunoglobulins and T cell receptors on the basis of IMGT/HighV-QUEST preprocessed NGS data. *BMC Bioinformatics* 16, 252.
- Scheid, J.F., Mouquet, H., Feldhahn, N., *et al.*, 2009. Broad diversity of neutralizing antibodies isolated from memory B cells in HIV-infected individuals. *Nature* 458, 636–640.
- Scheid, J.F., Mouquet, H., Ueberheide, B., *et al.*, 2011. Sequence and structural convergence of broad and potent HIV antibodies that mimic CD4 binding. *Science* 333, 1633–1637.
- Scott, M.G., Tarrand, J.J., Crimmins, D.L., *et al.*, 1989. Clonal characterization of the human IgG antibody repertoire to Haemophilus influenzae type b polysaccharide. II. IgG antibodies contain VH genes from a single VH family and VL genes from at least four VL families. *J. Immunol.* 143, 293–298.
- Stikakis, P.P., Karali, V., Lilakos, K., Georgiou, G., Panayiotidis, P., 2009. Clonal expansion of B-cells in human systemic lupus erythematosus: Evidence from studies before and after therapeutic B-cell depletion. *Clin. Immunol.* 132, 19–31.
- Shahaf, G., Barak, M., Zuckerman, N.S., *et al.*, 2008. Antigen-driven selection in germinal centers as reflected by the shape characteristics of immunoglobulin gene lineage trees: A large-scale simulation study. *J. Theor. Biol.* 255, 210–222.
- Shiroguchi, K., Jia, T.Z., Sims, P.A., Xie, X.S., 2012. Digital RNA sequencing minimizes sequence-dependent bias and amplification noise with optimized single-molecule barcodes. *Proc. Natl. Acad. Sci. USA.* 109, 1347–1352.
- Shlomchik, M.J., Marshak-Rothstein, A., Wolfowicz, C.B., Rothstein, T.L., Weigert, M.G., 1987. The role of clonal selection and somatic mutation in autoimmunity. *Nature* 328, 805–811.
- Shugay, M., Britanova, O.V., Merzlyak, E.M., *et al.*, 2014. Towards error-free profiling of immune repertoires. *Nat. Methods.*
- Silverman, G.J., Lucas, A.H., 1991. Variable region diversity in human circulating antibodies specific for the capsular polysaccharide of Haemophilus influenzae type b. Preferential usage of two types of VH3 heavy chains. *J. Clin. Invest.* 88, 911–920.
- Singh, V., Stoop, M.P., Stingl, C., *et al.*, 2013. Cerebrospinal-fluid-derived immunoglobulin G of different multiple sclerosis patients shares mutated sequences in complementarity determining regions. *Mol. Cell. Proteomics* 12, 3924–3934.
- Snir, O., Mesin, L., Gidoni, M., *et al.*, 2015. Analysis of celiac disease autoreactive gut plasma cells and their corresponding memory compartment in peripheral blood using high-throughput sequencing. *J. Immunol.* 194, 5703–5712.
- Sok, D., Laserson, U., Laserson, J., *et al.*, 2013. The effects of somatic hypermutation on neutralization and binding in the PGT121 family of broadly neutralizing HIV antibodies. *PLoS Pathog.* 9, e1003754.
- Stern, J.N.H., Yaari, G., Vander Heiden, J.A., *et al.*, 2014. B cells populating the multiple sclerosis brain mature in the draining cervical lymph nodes. *Sci. Transl. Med.* 6, 248ra107.
- Tabibian-Keissar, H., Hazanov, L., Schiby, G., *et al.*, 2016. Aging affects B-cell antigen receptor repertoire diversity in primary and secondary lymphoid tissues. *Eur. J. Immunol.* 46, 480–492.
- Tabibian-Keissar, H., Zuckerman, N.S., Barak, M., *et al.*, 2008. B-cell clonal diversification and gut-lymph node trafficking in ulcerative colitis revealed using lineage tree analysis. *Eur. J. Immunol.* 38, 2600–2609.
- Tak, P.P., Doorenspleet, M.E., de Hair, M.J.H., *et al.*, 2017. Dominant B cell receptor clones in peripheral blood predict onset of arthritis in individuals at risk for rheumatoid arthritis. *Ann. Rheum. Dis.* (annrheumdis-2017-211351).
- Tan, Y.-C., Blum, L.K., Kongpachith, S., *et al.*, 2014. High-throughput sequencing of natively paired antibody chains provides evidence for original antigenic sin shaping the antibody response to influenza vaccination. *Clin. Immunol.* 151, 55–65.
- Tsioris, K., Gupta, N.T., Ogunniyi, A.O., *et al.*, 2015. Neutralizing antibodies against West Nile virus identified directly from human B cells by single-cell analysis and next generation sequencing. *Integr. Biol.* 7, 1587–1597.

- Uduman, M., Shlomchik, M.J., Vigneault, F., Church, G.M., Kleinstein, S.H., 2014. Integrating B cell lineage information into statistical tests for detecting selection in Ig sequences. *J. Immunol.* 192, 867–874.
- Uduman, M., Yaari, G., Hershsberg, U., *et al.*, 2011. Detecting selection in immunoglobulin sequences. *Nucleic Acids Res.* 39, W499–W504.
- Vander Heiden, J.A., Yaari, G., Uduman, M., *et al.*, 2014. pRESTO: A toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires. *Bioinformatics* 30, 1930–1932.
- Vander Heiden, J., Gupta, N., Marquez, S., *et al.*, 2015. alakazam: Immunoglobulin clonal lineage and diversity analysis.
- von Büdingen, H.-C., Kuo, T.C., Sirota, M., *et al.*, 2012. B cell exchange across the blood-brain barrier in multiple sclerosis. *J. Clin. Invest.* 122, 4533–4543.
- Walker, L.M., Huber, M., Doores, K.J., *et al.*, 2011. Broad neutralization coverage of HIV by multiple highly potent antibodies. *Nature* 477, 466–470.
- Walker, L.M., Phogat, S.K., Chan-Hui, P.-Y., *et al.*, 2009. Broad and potent neutralizing antibodies from an African donor reveal a new HIV-1 vaccine target. *Science* 326, 285–289.
- Wang, Y., Jackson, K.J.L., Sewell, W.A., Collins, A.M., 2008. Many human immunoglobulin heavy-chain IGHV gene polymorphisms have been reported in error. *Immunol. Cell Biol.* 86, 111–115.
- Wardemann, H., Busse, C.E., 2017. Novel approaches to analyze immunoglobulin repertoires. *Trends Immunol.* 38, 471–482.
- Watson, C.T., Breden, F., 2012. The immunoglobulin heavy chain locus: Genetic variation, missing data, and implications for human disease. *Genes Immun.* 13, 363–373.
- Watson, C.T., Steinberg, K.M., Huddleston, J., *et al.*, 2013. Complete haplotype sequence of the human immunoglobulin heavy-chain variable, diversity, and joining genes and characterization of allelic and copy-number variation. *Am. J. Hum. Genet.* 92, 530–546.
- Weinstein, J.A., Jiang, N., White, R.A., Fisher, D.S., Quake, S.R., 2009. High-throughput sequencing of the zebrafish antibody repertoire. *Science* 324, 807–810.
- Wrammert, J., Koutsouanos, D., Li, G.-M., *et al.*, 2011. Broadly cross-reactive antibodies dominate the human B cell response against 2009 pandemic H1N1 influenza virus infection. *J. Exp. Med.* 208, 181–193.
- Wu, L., Yang, Z.-Y., Xu, L., *et al.*, 2006. Cross-clade recognition and neutralization by the V3 region from clade C human immunodeficiency virus-1 envelope. *Vaccine* 24, 4995–5002.
- Wu, X., Yang, Z.-Y., Li, Y., *et al.*, 2010a. Rational design of envelope identifies broadly neutralizing human monoclonal antibodies to HIV-1. *Science* 329, 856–861.
- Wu, X., Zhang, Z., Schramm, C.A., *et al.*, 2015a. Maturation and diversity of the VRC01-antibody lineage over 15 years of chronic HIV-1 infection. *Cell* 161, 470–485.
- Wu, X., Zhou, T., Zhu, J., *et al.*, 2011a. Focused evolution of HIV-1 neutralizing antibodies revealed by structures and deep sequencing. *Science* 333, 1593–1602.
- Wu, Y.-C.B., James, L.K., Vander Heiden, J.A., *et al.*, 2014. Influence of seasonal exposure to grass pollen on local and peripheral blood IgE repertoires in patients with allergic rhinitis. *J. Allergy Clin. Immunol.* 134, 604–612.
- Wu, Y.-C.B., Kipling, D., Dunn-Walters, D.K., 2012. Age-related changes in human peripheral blood IGH repertoire following vaccination. *Front. Immunol.* 3, 193.
- Wu, Y.-C., Kipling, D., Leong, H.S., *et al.*, 2010b. High-throughput immunoglobulin repertoire analysis distinguishes between human IgM memory and switched memory B-cell populations. *Blood* 116, 1070–1078.
- Yaari, G., Benichou, J.I.C., Vander Heiden, J.A., Kleinstein, S.H., Louzoun, Y., 2015. The mutation patterns in B-cell immunoglobulin receptors reflect the influence of selection acting at multiple time-scales. *Philos. Trans. R. Soc. B Biol. Sci.* 370, 20140242.
- Yaari, G., Kleinstein, S.H., 2015. Practical guidelines for B-cell receptor repertoire sequencing analysis. *Genome Med.* 7, 121.
- Yaari, G., Uduman, M., Kleinstein, S.H., 2012. Quantifying selection in high-throughput immunoglobulin sequencing data sets. *Nucleic Acids Res.* 40, e134.
- Yaari, G., Vander Heiden, J.A., Uduman, M., *et al.*, 2013. Models of somatic hypermutation targeting and substitution based on synonymous mutations from high-throughput immunoglobulin sequencing data. *Front. Immunol.* 4.
- Yahalom, G., Weiss, D., Novikov, I., *et al.*, 2013. An antibody-based blood test utilizing a panel of biomarkers as a new method for improved breast cancer diagnosis. *Biomark. Cancer* 5, 71–80.
- Ye, J., Ma, N., Madden, T.L., Ostell, J.M., 2013. IgBLAST: An immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res.* 41, W34–W40.
- Yermanos, A., Greiff, V., Krautler, N.J., *et al.*, 2017. Comparison of methods for phylogenetic B-cell lineage inference using time-resolved antibody repertoire simulations (AbSim). *Bioinformatics*.
- Zanotti, K.J., Gearhart, P.J., 2016. Antibody diversification caused by disrupted mismatch repair and promiscuous DNA polymerases. *DNA Repair (Amst)* 38, 110–116.
- Zhang, W., Feng, Q., Wang, C., *et al.*, 2017. Characterization of the B cell receptor repertoire in the intestinal mucosa and of tumor-infiltrating lymphocytes in colorectal adenoma and carcinoma. *J. Immunol.* 198, 3719–3728.
- Zhou, T., Xu, L., Dey, B., *et al.*, 2007. Structural definition of a conserved neutralization epitope on HIV-1 gp120. *Nature* 445, 732–737.
- Zhu, J., Ofek, G., Yang, Y., *et al.*, 2013. Mining the antibodyome for HIV-1-neutralizing antibodies with next-generation sequencing and phylogenetic pairing of heavy/light chains. *Proc. Natl. Acad. Sci. USA* 110, 6470–6475.
- Zuckerman, N.S., Hazanov, H., Barak, M., *et al.*, 2010a. Somatic hypermutation and antigen-driven selection of B cells are altered in autoimmune diseases. *J. Autoimmun.* 35, 325–335.
- Zuckerman, N.S., Howard, W.A., Bismuth, J., *et al.*, 2010b. Ectopic GC in the thymus of myasthenia gravis patients show characteristics of normal GC. *Eur. J. Immunol.* 40, 1150–1161.

Relevant Website

<http://airr.irmacs.sfu.ca/>
AIRR Community.

Quantitative Immunology by Data Analysis Using Mathematical Models

Shoya Iwanami, Kyushu University, Fukuoka, Japan

Shingo Iwami, Kyushu University, Fukuoka, Japan and Japan Science and Technology Agency, Kawaguchi, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Immunology covers many areas of research, such as the production mechanisms of diverse antibodies, the formulation and maintenance of the T-cell repertoire, the development and maturation of lymphocytes, discrimination of self and non-self, and the interactions between immune cells and viruses or cancer cells (Fig. 1). By investigating these topics from both basic and clinical viewpoints, we can understand pathology and develop treatment modalities for allergies, autoimmune diseases and immunodeficiency (Kishimoto, 2005). The broad-ranging topics in immunology are underpinned by the fundamental processes of immune systems, which dictate how cells, antibodies, and pathogens interact and the consequent effects of these interactions (Kitano, 2002). Owing to recently developed experimental techniques, we can clearly visualize the immune response process and its associated changes of signal transductions and gene expressions (Orkin and Zon, 2008; Notta *et al.*, 2016; Gewirtz *et al.*, 2001; Fehniger *et al.*, 1999). Immunologists can also exploit the large body of accumulated datasets. For example, bioinformatics and statistics have revealed several parts of homeostatic mechanisms in living bodies (Martinez *et al.*, 2006). To apply the knowledge obtained from cell or animal experiments at the clinical stage (e.g., in drug development and drug treatment strategies), we must quantitatively understand immune system dynamics such as antibody production changes and cell cycling (Perelson, 2002; Perelson and Nelson, 1999; Koizumi *et al.*, 2017). Mathematical models have become increasingly popular in experimental and clinical data analysis, and are now regarded as cutting-edge tools for quantitative interpretation in immunology (Perelson and Nelson, 1999; Perelson, 2002; Michor *et al.*, 2005; Bernitz *et al.*, 2016; De Boer and Perelson, 2013). In this paper, we introduce and discuss several important data analysis studies that quantify immune-cell differentiation, lymphocyte homeostasis, and viral infection using mathematical models.

Fundamentals

In response to potentially harmful internal and external stimuli, the immune system initiates many interactions of cells, antigens, antibodies and/or cytokines. These mechanisms can be described by “population dynamics” models (Hirsch *et al.*, 2012), which temporally evolve the concentration of cells, antigens, antibodies or cytokines throughout their interactions (Efimie *et al.*, 2011). In general, cell populations undergo four processes: “birth”, “death”, “immigration” and “emigration”. In populations of immune cells, these four processes correspond to cell proliferation, death, differentiation from other populations, and differentiation into other populations, respectively (see Fig. 1). These processes are generally formulated as follows:

$$dN_i(t)/dt = (p_i - a_i - (\sum_{j=1, j \neq i}^m d_{ij})N_i(t) + (\sum_{j=1, j \neq i}^m h_{ji})N_j(t) \quad (1)$$

where $N_i(t)$ is the number of cells in population i ($i=1, 2, \dots, m$) at time t . The parameters p_i and a_i are the proliferation and death rates of cell population i , respectively, d_{ij} is the emigration rate from population i to population j , and h_{ji} is the immigration rate from population j to population i . If cells in population i migrate into population j by differentiation, then d_{ij} and h_{ji} are equal. If cells of population i are produced from cells in population j , d_{ij} is 0 and h_{ji} is positive. The coefficients might be constants or functions of $N_h(t)$ ($h=1, 2, \dots, m$) to describe interactions with other populations. This generalized model can be tailored to

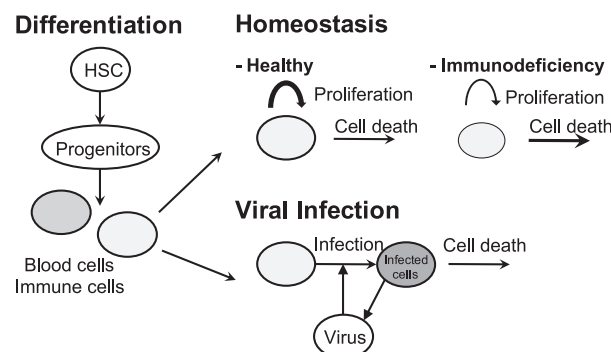


Fig. 1 Schematic of the immune system. Immune cells are derived from HSCs and the immune system is homeostatically maintained in a healthy state that protects against pathogens. Homeostasis is considered to fail in the disease state. The arrows correspond to interactions between populations of cells or antigens. The interactions among populations can be incorporated into mathematical models of dynamic population changes.

model the interactions and differentiation mechanisms of cell populations of interest to the researcher. It can also include hypothetical but experimentally testable interactions. Moreover, using statistical criteria such as Akaike's Information Criterion (AIC) and the likelihood ratio test, we can select the most suitable model among several hypothetical models, discussing empirical possibilities and guiding the next steps in experimental studies (Akaike, 1974). The generalized model can clearly define the focal areas of a study, enhancing our understanding of immune systems rather more than complex models. The following sections exemplify some of the novel insights gained from the generalized model.

Applications

Owing to recent technological developments, the snapshot comparisons in classical researches can be supplemented by novel researches that quantitatively characterize biological behaviors from the dynamical changes in time-course data. In the immunology field, mathematical modeling is increasingly used for describing and quantifying the time-course data of clinical practices and animal/cell experiments.

Cell Differentiation

Cell differentiation is the process by which dividing cells change their functional or phenotypical type. All cells presumably derive from stem cells and obtain their functions as they mature. Cellular composition is often modeled as a hierarchical scheme with stem cells at the top of the hierarchy. Mathematical models of cell differentiation consider the proliferation, death, and differentiation of an appropriately distinguished cell population.

For instance, the cells in a stem cell population differentiate into multiple cells but maintain their number by self-renewal. A progenitor cell population derives from stem cells and differentiates into a restricted lineage, whereas a matured cell population derives from progenitor cells, performs some functions, and dies a natural death. Considering these populations as cell differentiation, Michor *et al.* (2005) proposed a set of ordinary differential equations (ODEs) describing the population dynamics of four cell populations (hematopoietic stem cells, progenitors, differentiated cells and terminally differentiated cells), which explains the clinical data of leukemia. In successful cancer therapy, the number of leukemic cells exhibits a biphasic exponential decline. By fitting the model to the clinical data, the authors revealed that this behavior corresponds to the turnover rates of differentiated and progenitor cells. Furthermore, their study implied that cancer therapy strongly inhibits the production of differentiated leukemic cells, but does not deplete leukemic stem cells (Michor *et al.*, 2005). Their mathematical modeling provided valuable quantitative insights.

Abundant datasets are also available for hematopoietic cell systems, for which the cell hierarchy is well characterized (Orkin and Zon, 2008; Dingli *et al.*, 2007). In hematopoietic stem cell research, the number of divisions of stem cells can be measured by labeling techniques. Bernitz *et al.* (2016) distinguished the numbers of different hematopoietic stem cells by the numbers of cell divisions, which were determined from the intensities of fluorescent protein incorporated in the stem cells. Fitting a mathematical model to the data, they estimated the rate at which the hematopoietic stem cells leaned toward a myeloid lineage during differentiation. Thus, the mathematical modeling of cell differentiation quantifies the parameters that cannot be obtained from conventional experimental data.

Lymphoid Cell Turnover

Humans maintain a stable internal environment by special mechanisms called homeostatic mechanisms, which maintain the number of cells in the body within a certain range. Consequently, the number of human lymphocytes such as T lymphocytes is usually unchanged. Dysregulation of homeostasis leads to disease conditions. Abnormal cells that increase their numbers by uncontrolled cell division become malignant cancer cells. Immune cells can become depleted or malfunctioning (leading to immune deficiency) or unusually activated to attack the body's own tissues (causing autoimmune diseases). Such disease states cannot be understood by merely comparing the numbers of cells in healthy and diseased states.

An analog of pyrimidine, 5-Bromo-2'-deoxyuridine (BrdU), can be incorporated into DNA during cell division. Under BrdU administration, dividing cells become labeled and can be detected by a BrdU antibody for measuring the dynamics of lymphocytes (Grazner, 1982). Since the 2000s, immunology studies have combined classical BrdU labeling with mathematical modeling (De Boer and Perelson, 2013). Interestingly, BrdU labeling and mathematical modeling can together quantify the proliferation or death rates of cells, and detect differences in the BrdU-labeled lymphocyte dynamics. For example, Mohri *et al.* (1998) compared the time-course BrdU labeling data from rhesus macaques infected with simian immunodeficiency virus (SIV) and their non-infected counterparts. They found that SIV infection dramatically increases the turnover of T lymphocytes. Their assumption – a constant total number of T lymphocytes – was relaxed in a later mathematical model (Bonhoeffer *et al.*, 2000). Similar approaches have quantified the turnovers of other lymphocyte populations such as B cells, T cells, and NK cells (De Boer *et al.*, 2003b). De Boer *et al.* (2003a) proposed a new index, the "average turnover rate", defined as the cellular death rate averaged over all subpopulations in the model (i.e., $\hat{\delta} = (\Sigma)_i^m (\alpha_i \delta_i)$ ($i=1, \dots, m$), where $\hat{\delta}$, δ_i and α_i are the average death rate, a death rate of subpopulation i and a fraction of subpopulation i among whole cell population, respectively, and m is the number of subpopulations.). This value is estimated similarly by different models. In summary, mathematical modeling complements conventional BrdU labeling experiments to enhance our quantitative understanding of lymphocyte dynamics.

Virus Dynamics

The representative virus of immunology studies is human immunodeficiency virus type I (HIV-1), which causes acquired immune deficiency syndrome (AIDS). Viral infection behaviors, called “virus dynamics”, have been mathematically studied since the 1990s (Ho *et al.*, 1995; Wei *et al.*, 1995). Mathematical models of virus dynamics have analyzed many kinds of *in vivo* experimental and clinical data, and have elucidated various aspects of HIV biology, especially, the average lifespan of virus-producing cells (Simon and Ho, 2003; Perelson, 2002). Many recent studies have combined mathematical modeling with cell culture (i.e., *in vitro*) experiments (Beauchemin *et al.*, 2017; Ito *et al.*, 2017; Iwami *et al.*, 2015; Iwanami *et al.*, 2017; Mahgoub *et al.*, 2018; Ikeda *et al.*, 2015). In cell culture experiments, one can measure several different kinds of time-course experimental data, enabling robust parameter estimation for the mathematical model. For example, Ikeda *et al.* (2015) coupled mathematical modeling with cell culture experiments to quantify the antiviral effects of interferon (IFN) on HIV-1 infection. They deduced that IFN acts by inhibiting *de novo* infection rather than virus production. Thus, quantitative estimates can help to resolve unanswered questions in immunology.

Analysis and Assessment

Mathematical Modeling of Hematopoietic Stem Cell Differentiation

The type and number of the target cells can be measured in flow cytometers. The functional types of immune cells are easily distinguished by markers attached on the cells. Therefore, the differentiation rates or pathways can be theoretically estimated by analyzing the count data of cell phenotypes in mathematical models. In simple terms, differentiation occurs when either or both of the two daughter cells change(s) after cell division. To mathematically describe cell differentiation, we set the number of daughter cells as twice the number of mother cells, then immigrate some of the daughters into the next cell population at a certain rate, maintaining the others in their current precursor population. For example, Thomas-Vaslin *et al.* (2008) described the differentiation process from CD4 and CD8 double-negative T lymphocytes in the thymus to CD4 or CD8 single-positive naïve T lymphocytes in the spleen by ODEs. Fitting their mathematical model to the measured number of T lymphocytes in each cell-differentiation stage, they estimated the turnover rate of each stage and the death rate of T lymphocytes in the thymus. Schlub *et al.* (2009) theoretically explained the formulation mechanism of central- memory and effector-memory T lymphocyte repertoires. Here, the memory cells compose the immunological memory. Buchholz *et al.* (2013) mathematically identified conceivable differentiation models (from naïve T cells to central memory precursors, effector memory precursors, or effector cells) by fitting them to time-course data. The best-fit model was selected by ranking the AICs of the models. These studies are expected to elucidate the homeostasis mechanism of the immune system against various stimuli.

For a deeper understanding of immune-system homeostasis, we need to elucidate the differentiation of hematopoietic stem cells (HSCs), which sustain the production of immune cells. Although HSCs are known to differentiate into two major lineages (the lymphoid and myeloid lineages), how these lineages branch during HSC differentiation and how aging HSCs maintain their homeostasis and functions are not fully understood. In fact, diseases related to the immune system are sometimes considered to arise from disorders of aging HSCs. Crude markers and functional heterogeneity of HSCs have been reported (Yamamoto *et al.*, 2013). Therefore, by combining mathematical modeling with flow-cytometer measurements of HSC numbers, we might elucidate the mechanism of HSC differentiation. HSC differentiation is among the most fundamental processes in immune-cell differentiation, and HSC research has recently adopted mathematical modeling for quantitatively understanding the differentiation phenomena. Assuming that HSCs expand through four symmetric divisions, the increasing HSC numbers during aging can be described by the following ODEs (Fig. 2):

$$\begin{aligned}
 dx_0(t)/dt &= -k_0x_0(t) \\
 dx_1(t)/dt &= 2k_0x_0(t) - k_1x_1(t) \\
 dx_2(t)/dt &= 2k_1x_1(t) - k_2x_2(t) \\
 dx_3(t)/dt &= 2k_2x_2(t) - k_3x_3(t) \\
 dx_4(t)/dt &= 2k_3x_3(t) - k_4x_4(t)
 \end{aligned} \tag{2}$$

Here, $x_n(t)$ is the number of HSCs that have divided n times ($n=0, 1, 2, 3, 4$) and $1/k_n$ is the expected time interval to the next division for a cell that has previously divided n times. After fitting the model to the measured number of HSCs in young mice, the division rate of murine HSCs that had divided n times was best described by $k_n = k\beta^{2n}$ ($0 < \beta < 1$) (Bernitz *et al.*, 2016). Note that by using phenotypes (which roughly distinguish HSCs from other cells), and by tracking the intensity of fluorescent protein (which decreases by half after each cell division), we can measure the number of cells corresponding to each of the n division events.

As HSCs age, their differentiation ability leans toward the myeloid lineage (Notta *et al.*, 2016), and the fraction of cells expressing CD41 (a marker of increased myeloid differentiation) is lower in young mice than in adult mice (Bernitz *et al.*, 2016). Considering the bias toward the myeloid lineage, defined by CD41 expression, the HSC dynamics can be described by the following system of ODEs (Fig. 2):

4 step symmetric division model

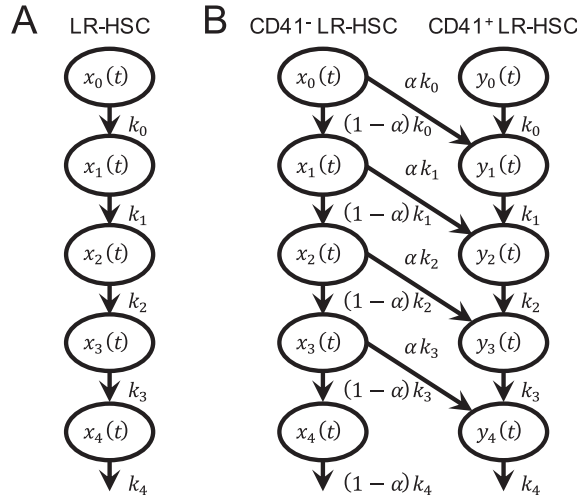


Fig. 2 Division scheme of HSCs with aging. Mathematical model describing the dynamics of aging HSCs. (A) $x_n(t)$ denotes the number of HSCs that have divided n times ($n=0, 1, 2, 3, 4$) at time t , and $1/k_n$ is the expected time interval to the next division for a cell that has previously divided n times. (B) $x_n(t)$ and $y_n(t)$ denote the number of CD41⁻ and CD41⁺ HSCs, respectively, that have divided n times at time t . A fraction α of the divided cells in the CD41⁻ HSC population migrate into the CD41⁺ HSC population.

$$\begin{aligned}
 dx_0(t)/dt &= -k_0x_0(t) \\
 dx_1(t)/dt &= 2(1-\alpha)k_0x_0(t) - k_1x_1(t) \\
 dx_2(t)/dt &= 2(1-\alpha)k_1x_1(t) - k_2x_2(t) \\
 dx_3(t)/dt &= 2(1-\alpha)k_2x_2(t) - k_3x_3(t) \\
 dx_4(t)/dt &= 2(1-\alpha)k_3x_3(t) - k_4x_4(t) \\
 dy_0(t)/dt &= -k_0y_0(t) \\
 dy_1(t)/dt &= 2k_0y_0(t) + 2\alpha k_0x_0(t) - k_1y_1(t) \\
 dy_2(t)/dt &= 2k_1y_1(t) + 2\alpha k_1x_1(t) - k_2y_2(t) \\
 dy_3(t)/dt &= 2k_2y_2(t) + 2\alpha k_2x_2(t) - k_3y_3(t) \\
 dy_4(t)/dt &= 2k_3y_3(t) + 2\alpha k_3x_3(t) - k_4y_4(t)
 \end{aligned} \tag{3}$$

Here, $x_n(t)$ and $y_n(t)$ are the number of HSCs not expressing and expressing CD41, respectively. The cells have divided n times ($n=0, 1, 2, 3, 4$). In this model, the CD41⁻ HSCs gain CD41 expression at a rate α with each division. Analyzing HSC count data, α was estimated as 0.12, meaning that the cells gain CD41 expression after every ten divisions (Bernitz et al., 2016). In this way, the differentiation rate of HSCs was quantitatively deduced from detailed cell count data and the mathematical model.

Basic Model of BrdU Labeling

Lymphocyte dynamics are commonly quantified by two DNA labeling methods, the BrdU labeling technique introduced in Section “Applications”, and deuterium labeling, which can be provided as heavy water or deuterated glucose. Under deuterium administration, some of the hydrogen atoms of the DNA that was newly synthesized during cell division are replaced with deuterium atoms, which are detectable by mass spectrometry. Deuterium glucose is non-toxic to humans, so is available for quantifying the dynamics of human lymphocytes (Macallan et al., 2003, 2005; Mohri et al., 2001; Ribeiro et al., 2002). Both methods label the DNA of dividing cells. DNA can also be labeled by an intracellular fluorescent dye called carboxy-fluorescein diacetate succinimidyl ester (CFSE) (Lyons, 2000). After measuring the fluorescence intensity of the CFSE-labeled cells, one can determine the distribution of the number of cell divisions undergone by each cell, because (as mentioned above) the fluorescence intensity halves after each cell division. CFSE-labeling data have been analyzed by several ODE models (Revy et al., 2001; Luzyanina et al., 2007), delay equations based on the cell cycle (De Boer and Perelson, 2005) and stochastic processes (Hawkins et al., 2007).

In this subsection, we discuss mathematical models for analyzing BrdU labeling data. The fraction of cells incorporating BrdU is measured by detecting the BrdU antibodies attached to the cells. During BrdU administration, all unlabeled cells become two labeled daughter cells after a cell division, and all labeled cells divide into two labeled daughter cells. After the BrdU administration, the unlabeled cells divide into two unlabeled cells but the labeled cells divide into two labeled daughter cells with diluted

BrdU. Let us assume that a cell inflowing from the outside is labeled during the BrdU administration but is not labeled after the administration is withdrawn (Fig. 3; De Boer *et al.*, 2003a,b; Kaur *et al.*, 2008). The lymphocyte dynamics during the BrdU administration are then described by the following ODEs:

$$\begin{aligned} dL(t)/dt &= s + 2pU(t) + (p-d)L(t) \\ dU(t)/dt &= -(p+d)U(t) \end{aligned} \quad (4)$$

Here, $L(t)$ and $U(t)$ are the fractions of BrdU-labeled and unlabeled cells among the activated lymphocytes, respectively. s is a constant rate of inflowing cells per unit time, and p and d are the proliferation and death rate constants of the lymphocytes, respectively. As the number of total activated cells, $A(t) = L(t) + U(t)$, usually remains constant, we obtain:

$$A(t) = A = s/(p-d) \quad (5)$$

The lymphocyte dynamics after the BrdU administration are described by the following ODEs:

$$\begin{aligned} dL(t)/dt &= (p-d)L(t) \\ dU(t)/dt &= s + (p-d)U(t) \end{aligned} \quad (6)$$

Let R be the number of unactivated lymphocytes. The total number of lymphocytes is then $T = A + R$, and the fractions of BrdU-labeled lymphocytes during and after the BrdU administration ($f_L(t) = L(t)/T$) can be analytically solved as follows:

$$\begin{aligned} f_L(t) &= 100\alpha(1 - e^{-(d+p)t})(t \leq T_{\text{end}}) \\ f_L(t) &= 100\alpha(1 - e^{-(d+p)T_{\text{end}}})e^{(p-d)(t-T_{\text{end}})}(t > T_{\text{end}}) \end{aligned} \quad (7)$$

Here T_{end} represents the end of the BrdU administration, and α is the maximum fraction of BrdU-labeled lymphocytes if the BrdU is sufficiently administrated (i.e., $\alpha = A/T$) (De Boer *et al.*, 2003a,b). Fitting Eq. (7) to the BrdU-labeled experimental data and comparing the estimated parameters α , p and d under different conditions (e.g., health and disease), one can detect the affected process in terms of parameter values. In the macaque study of (Mohri *et al.*, 1998), the replacement rates of CD3 + CD4 + / - lymphocytes (defined as the difference between the death and proliferation rates of the lymphocytes) were approximately two- to three-fold higher in SIV-infected macaques than in uninfected macaques, indicating that SIV infection is associated with heightened lymphocyte turnover (Mohri *et al.*, 1998). Quantified cell dynamics, i.e., the rates of cell division and cell death, can effectively reveal a pathophysiology or the effect of a treatment, and have been reported in many studies (De Boer *et al.*, 2003a,b; De Boer and Perelson, 2013; Kaur *et al.*, 2008; Asquith *et al.*, 2007).

Modeling the Antiviral Effect of IFN

Model (8) is a classical but well-parameterized mathematical model of virus infection, which has been especially applied in cell culture (see Fig. 4; Iwami *et al.*, 2012b; Perelson, 2002; Perelson and Nelson, 1999):

$$\begin{aligned} dT(t)/dt &= -\beta T(t)V(t) \\ dI(t)/dt &= \beta T(t)V(t) - \delta I(t) \\ dV(t)/dt &= pI(t) - cV(t) \end{aligned} \quad (8)$$

Here, $T(t)$ and $I(t)$ are the numbers of target (i.e., uninfected) cells and infected cells, respectively, and $V(t)$ is the viral load at time t . β is the rate constant of the infection. Infected cells die at rate δ per unit time. Viruses are produced from infected cells at rate p per unit time, and degrade at rate c per unit time. To quantify the antiviral effect of IFN on HIV-1-infected cells in culture, Ikeda *et al.* (2015) proposed the following extended mathematical model of target cell growth:

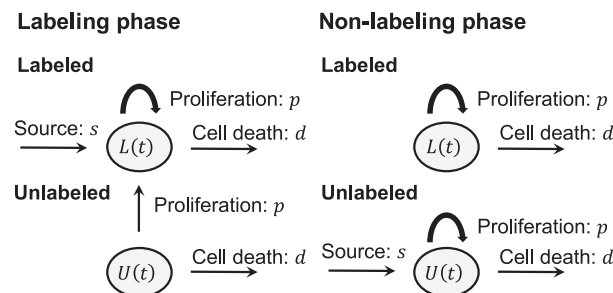


Fig. 3 BrdU-labeling scheme during and after administration. BrdU-labeling dynamics of a leukocyte population. $L(t)$ and $U(t)$ denote the numbers of labeled and unlabeled cells, respectively, at time t . In the labeling phase, unlabeled cells become labeled after cell proliferation and outside cells flow into the population of labeled cells. In the non-labeling phase, outside cells flow into the population of unlabeled cells.

$$\begin{aligned}
dT(t)/dt &= gT(t)(1 - (T(t) + I(t))/T_{\max}) - (1 - \eta)\beta T(t)V(t) \\
dI(t)/dt &= (1 - \eta)\beta T(t)V(t) - \delta I(t) \\
dV(t)/dt &= (1 - \varepsilon)pI(t) - cV(t)
\end{aligned} \tag{9}$$

In Model (9), IFN is assumed to inhibit *de novo* infection and virus production at rates η and ε , respectively ($0 < \eta, \varepsilon < 1$). By fitting Eq. (9) to the experimental cell-culture data in the presence and absence of IFN, Ikeda *et al.*, (2015) found that IFN is at least two-fold more effective against *de novo* infection (η) than against virus production (ε). As discussed in landmark papers (Neumann *et al.*, 1998; Dixit *et al.*, 2004), mathematical modeling can extract the unknown action mechanisms of antiviral drugs from experimental and clinical data.

Illustrative Examples

All models in Section “Analysis and Assessment” were derived from the generalized mathematical model Eq. (1) introduced in Section “Fundamentals”.

In Model (2), the $-k_n x_n(t)$ terms denote emigrations from cell populations that have divided n times (n th-population cells) to populations of cells that have divided $n + 1$ times ($[n + 1]$ th- population cells) by cell division (i.e., k_n corresponds to d_{ij} , $i = x_n$, $j = x_{n+1}$). Immigrations from the n th population to the $(n + 1)$ th population are described by $2k_n x_n(t)$, because cells divide into two daughter cells (i.e., k_n corresponds to d_{ij} , where $i = x_{n+1}$, $j = x_n$, and $2k_n$ corresponds to h_{ij} , where $i = x_{n+1}$, $j = x_n$). In Model (3), the $CD41^-$ cells emigrated from the n th population divide into two daughter cells, which then immigrate to the $(n + 1)$ th populations of $CD41^-$ cells at a rate of $1 - \alpha$, or to the $(n + 1)$ th population of $CD41^+$ cells while gaining $CD41$ expression at rate α . In this case, $2(1 - \alpha)k_n$ corresponds to h_{ij} , where $i = x_{n+1}$, $j = x_n$, and $2\alpha k_n$ corresponds to h_{ij} , where $i = x_{n+1}$, $j = y_n$. The emigrated $CD41^+$ cells of the n th population divide into two daughter cells and immigrate to the $(n + 1)$ th population of $CD41^+$ cells. In models (2) and (3), the proliferation and death processes are omitted because both daughter cells generated by the cell division migrate to the next population and HSCs are not assumed to die within the time scale of the experiment.

In Model (4), the emigrated unlabeled cells ($-pU(t)$), which divide into two daughter cells and become labeled by BrdU, immigrate into populations of BrdU-labeled cells ($2pU(t)$) during the BrdU administration (i.e., p corresponds to d_{ij} , where $i = U$, $j = L$, and $2p$ corresponds to h_{ij} , where $i = L$, $j = U$). Outside cells immigrating into a population of BrdU-labeled cells are described by s , because all such cells are assumed to be labeled by BrdU. The terms $-dL(t)$ and $-dU(t)$ denote a cell death event among the BrdU-labeled and unlabeled cells, respectively (i.e., d corresponds to a_i , where $i = L, U$). In Model (6), the BrdU-labeled and -unlabeled cells proliferate and die at rates p and d after BrdU administration, respectively (i.e., p corresponds to p_i , with $i = L, U$, and d corresponds to a_i , with $i = L, U$). Outside immigration into a population of unlabeled cells is described by s because all such cells are assumed to be unlabeled (i.e., s corresponds to d_{ij} , where $i = L$ and $j = \text{outside}$).

In Model (8), the term $\beta T(t) V(t)$ describes a migration event from the target cells into the virally infected cells (i.e., a new viral infection of a target cell). This migration rate $\beta V(t)$ assumes that the target cells and viruses are well-mixed and that all target cells encounter all viruses at the same frequency (i.e., $\beta V(t)$ corresponds to d_{ij} , where $i = T$, $j = I$). The infected cells die by cytopathogenesis at rate δ (i.e., δ corresponds to a_i , with $i = I$). Immigration into the virus population is denoted by $pI(t)$, which describes the newly produced viruses from infected cells (i.e., p corresponds to h_{ij} , where $i = V$, $j = I$). In Model (9), the cell-growth term $gT(t)(1 - (T(t) + I(t))/T_{\max})$ is a form of the “logistic equation” (i.e., $(1 - (T(t) + I(t))/T_{\max})$ corresponds to p_i , with $i = T$). Migrations from the target cells into the infected cells by infection, and immigrations into the virus population by viral production in infected cells, are inhibited at rates $(1 - \eta)$ and $(1 - \varepsilon)$ by the antiviral effects of IFN, respectively. In this case, $(1 - \eta)\beta V(t)$ corresponds to d_{ij} , where $i = T$, $j = I$, and $(1 - \varepsilon)p$ corresponds to h_{ij} , where $i = V$, $j = I$.

Results and Discussion

Mathematical modeling of experimental systems is important for obtaining quantitative results. However, the estimated parameters and the interpreted results of a data analysis depend on the experimental system and/or formulation of the mathematical model. Therefore, the assumptions underlying these models should be biologically reasonable. When several mathematical models can conceivably describe the analysis result, immunologists must base their choice on the precise assumptions underlying the different models (De Boer and Perelson, 2013). Moreover, when developing a mathematical model based on the generalized model (1), the modeler should include only those cell populations and interactions with a sound scientific or immunological basis.

For example, the cell population dynamics in Model (3) are distinguished by the number of cell divisions undergone by each cell, and by the presence or absence of $CD41$ expression (Ito *et al.*, 2017; Bernitz *et al.*, 2016). On the one hand, this mathematical model sufficiently estimates the rate at which $CD41^-$ HSCs gain $CD41$ expression from the cell-division number distribution in the presence or absence of $CD41$ expression, which was experimentally determined. On the other hand, Model (4) considers two subpopulations of the activated lymphocytes (BrdU-labeled and BrdU-unlabeled). Mathematical models of BrdU-labeling, regardless of their formulation, are known to robustly estimate “average turnover” (De Boer *et al.*, 2003a,b). Thus, under clear assumptions, a theoretical study is advantageous for understanding the relationships between different mathematical models. Viral infection models of target cells, infected cells and viruses are also derived from the generalized model (1). An example is Model (9), which was established for both clinical and experimental data analysis (Perelson, 2002). Other model branched from this fundamental model including the non-infectious virus population was fitted to the measured concentration of target cells,

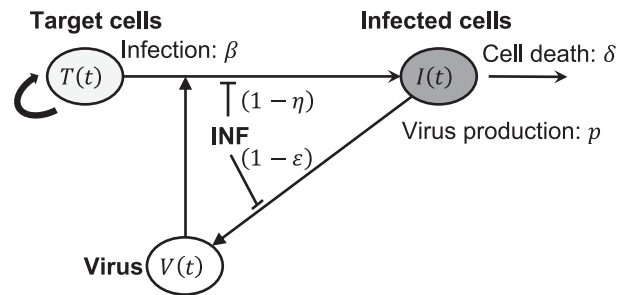


Fig. 4 Infection scheme including the effect of INF. Dynamics of HIV-1 infection, showing the anti-viral effect of INF on *de novo* infection and virus production. $T(t)$ and $I(t)$ denote the number of target (uninfected) cells and infected cells at time t , respectively. $V(t)$ denotes the viral load at time t . In this model, target cells are infected with viruses at rate β and infected cells produce viruses at rate p . The inhibition rates of infection and virus production by INF are η and ϵ , respectively.

concentration of infected cells, total viral load, and infectious viral load. From the fittings, the production efficacy of infectious virus was estimated (Iwami *et al.*, 2012a; Iwanami *et al.*, 2017).

In summary, the simple generalized model (1) can provide novel immunological insights. A deep and mutual understanding between theoreticians and experimentalists can establish a strong collaboration of mathematical models and experimental data.

Future Directions

Various experimental methods for cell counting and data analysis have been established in the last 20 years. The coupling of experimental methods with mathematical models can quantitatively reveal the dynamics of different cell types, as discussed herein. This approach should be followed by mathematical models and computer simulations that reveal the complex, hierarchical effective interactions among immune cells that maintain homeostasis. Recall that a failure of this balanced system leads to a diseased state. For example, osteoimmunology is an interdisciplinary field that tries to link immunological interactions to the mechanisms of bone formation and bone resorption (Takayanagi, 2007). Osteoblasts and osteoclasts interact through cytokines represented by RANKL, a member of the tumor necrosis factor family of cytokines. RANKL also works in the immune system. These interactions maintain the balance of bone formation and bone resorption. When this balanced interaction fails, bone diseases such as osteoporosis and fibrodysplasia ossificans progressiva will develop. The generalized model (i.e., Eq. (1)) and its derived version are appropriate and suitable for elucidating the mechanisms of these complex immune-related diseases.

Closing Remarks

Immunological research has greatly benefitted from experimental techniques in molecular and cell biology, which have rapidly advanced in recent years. However, many of these experimental techniques elucidate only one aspect of immunological phenomena. In tandem with rigorous experimental work, mathematical modeling offers an excellent opportunity for comprehensively analyzing immunological events. At one time, modeling work was essentially ignored by experimental immunologists, but in the last 20 years, it has become an important tool in biological research. In fact, almost all experimental biology groups are now collaborating with a theoretical scientist. Mathematical modeling provides quantitative insights that cannot be obtained by experimental and clinical studies alone. The examples in the present article were selected to highlight these strengths.

Acknowledgment

This work was supported in part by the JST PRESTO and CREST program (to S.I.), the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers 16H04845, 16K13777, 15KT0107 and 26287025 (to S.I.), a Grant-in-Aid for Scientific Research on Innovative Areas from the Ministry of Education, Culture, Science, Sports, and Technology (MEXT) of Japan 16H06429, 16K21723, and 17H05819 (to S.I.), J-PRIDE 17fm0208006h0001, 17fm0208019h0101, 17fm0208014h0001 (to S.I.), the Program on the Innovative Development and the Application of New Drugs for Hepatitis B 17fk0310114j0001, AMED (to S.I.), the Mitsui Life Social Welfare Foundation (to S.I.), the Shin-Nihon of Advanced Medical Research (to S.I.), a GSK Japan Research Grant 2016 (to S.I.), the Mochida Memorial Foundation for Medical and Pharmaceutical Research (to S.I.), the Suzuken Memorial Foundation (to S.I.), the SEI Group CSR Foundation (to S.I.), the Life Science Foundation of Japan (to S.I.), the SECOM Science and Technology Foundation (to S.I.), the Center for Clinical and Translational Research of Kyushu University Hospital (to S.I.), the Kyushu University-initiated venture business seed development program (to S.I.), and the Japan Prize Foundation (to S.I.). We thank Leonie Pipe, PhD, from Edanz Group (see “Relevant Website section”) for editing a draft of this manuscript.

See also: Bioinformatics Approaches for Studying Alternative Splicing. Chromatin: A Semi-Structured Polymer. Comparative and Evolutionary Genomics. Computational Immunogenetics. Exome Sequencing Data Analysis. Functional Genomics. Gene Mapping. Genome Alignment. Genome Annotation. Genome Annotation: Perspective From Bacterial Genomes. Genome Databases and Browsers. Genome Informatics. Genome-Wide Association Studies. Hidden Markov Models. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Learning Chromatin Interaction Using Hi-C Datasets. Linkage Disequilibrium. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Nucleosome Positioning. Phylogenetic Footprinting. Repeat in Genomes: How and Why You Should Consider Them in Genome Analyses?. Whole Genome Sequencing Analysis

References

- Akaike, H., 1974. A new look at the statistical model identification. *IEEE Trans. Autom. Control* 19 (6), 716–723.
- Asquith, B., Zhang, Y., Mosley, A.J., *et al.*, 2007. In vivo T lymphocyte dynamics in humans and the impact of human T-lymphotropic virus 1 infection. *Proc. Natl. Acad. Sci. USA* 104 (19), 8035–8040.
- Beauchemin, C.A., Miura, T., Iwami, S., 2017. Duration of SHIV production by infected cells is not exponentially distributed: Implications for estimates of infection parameters and antiviral efficacy. *Sci. Rep.* 7, 42765.
- Bernitz, J.M., Kim, H.S., Macarthur, B., Sieburg, H., Moore, K., 2016. Hematopoietic stem cells count and remember self-renewal divisions. *Cell* 167, 1296–1309. e10.
- Bonhoeffer, S., Mohri, H., Ho, D., Perelson, A.S., 2000. Quantification of cell turnover kinetics using 5-bromo-2'-deoxyuridine. *J. Immunol.* 164, 5049–5054.
- Buchholz, V.R., Flossdorf, M., Hensel, I., *et al.*, 2013. Disparate individual fates compose robust CD8+ T cell immunity. *Science* 340, 630–635.
- De Boer, R.J., Mohri, H., Ho, D.D., Perelson, A.S., 2003a. Estimating average cellular turnover from 5-bromo-2'-deoxyuridine (BrdU) measurements. *Proc. Biol. Sci.* 270, 849–858.
- De Boer, R.J., Mohri, H., Ho, D.D., Perelson, A.S., 2003b. Turnover rates of B cells, T cells, and NK cells in simian immunodeficiency virus-infected and uninfected rhesus macaques. *J. Immunol.* 170, 2479–2487.
- De Boer, R.J., Perelson, A.S., 2005. Estimating division and death rates from CFSE data. *J. Comput. Appl. Math.* 184 (1), 140–164.
- De Boer, R.J., Perelson, A.S., 2013. Quantifying T lymphocyte turnover. *J. Theor. Biol.* 327, 45–87.
- Dingli, D., Traulsen, A., Pacheco, J.M., 2007. Compartmental architecture and dynamics of hematopoiesis. *PLOS ONE* 2, e345.
- Dixit, N.M., Layden-Almer, J.E., Layden, T.J., Perelson, A.S., 2004. Modelling how ribavirin improves interferon response rates in hepatitis C virus infection. *Nature* 432, 922–924.
- Eftimie, R., Bramson, J.L., Earn, D.J., 2011. Interactions between the immune system and cancer: A brief review of non-spatial mathematical models. *Bull. Math. Biol.* 73, 2–32.
- Fehniger, T.A., Shah, M.H., Turner, M.J., *et al.*, 1999. Differential cytokine and chemokine gene expression by human NK cells following activation with IL-18 or IL-15 in combination with IL-12: Implications for the innate immune response. *J. Immunol.* 162 (8), 4511–4520.
- Gewirtz, A.T., Navas, T.A., Lyons, S., Godowski, P.J., Madara, J.L., 2001. Cutting edge: Bacterial flagellin activates basolaterally expressed TLR5 to induce epithelial proinflammatory gene expression. *J. Immunol.* 167, 1882–1885.
- Gratzner, H.G., 1982. Monoclonal antibody to 5-bromo- and 5-iododeoxyuridine: A new reagent for detection of DNA replication. *Science* 218, 474–475.
- Hawkins, E.D., Turner, M.L., Dowling, M.R., Van Gend, C., Hodgkin, P.D., 2007. A model of immune regulation as a consequence of randomized lymphocyte division and death times. *Proc. Natl. Acad. Sci. USA* 104, 5032–5037.
- Hirsch, M.W., Smale, S., Devaney, R.L., 2012. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic Press.
- Ho, D.D., Neumann, A.U., Perelson, A.S., *et al.*, 1995. Rapid turnover of plasma virions and CD4 lymphocytes in HIV-1 infection. *Nature* 373, 123–126.
- Ikeda, H., Godinho-Santos, A., Rato, S., *et al.*, 2015. Quantifying the antiviral effect of IFN on HIV-1 Replication in Cell Culture. *Sci. Rep.* 5, 11761.
- Ito, Y., Remion, A., Tazuin, A., *et al.*, 2017. Number of infection events per cell during HIV-1 cell-free infection. *Sci. Rep.* 7, 6559.
- Iwami, S., Holder, B.P., Beauchemin, C.A., *et al.*, 2012a. Quantification system for the viral dynamics of a highly pathogenic simian/human immunodeficiency virus based on an in vitro experiment and a mathematical model. *Retrovirology* 9, 18.
- Iwami, S., Sato, K., De Boer, R.J., *et al.*, 2012b. Identifying viral parameters from in vitro cell cultures. *Front. Microbiol.* 3, 319.
- Iwami, S., Takeuchi, J.S., Nakaoka, S., *et al.*, 2015. Cell-to-cell infection by HIV contributes over half of virus infection. *elife*. 4.
- Iwanami, S., Kakizoe, Y., Morita, S., *et al.*, 2017. A highly pathogenic simian/human immunodeficiency virus effectively produces infectious virions compared with a less pathogenic virus in cell culture. *Theor. Biol. Med. Model.* 14, 9.
- Kaur, A., Di Mascio, M., Barabasz, A., *et al.*, 2008. Dynamics of T- and B-lymphocyte turnover in a natural host of simian immunodeficiency virus. *J. Virol.* 82, 1084–1093.
- Kishimoto, T., 2005. Interleukin-6: From basic science to medicine – 40 years in immunology. *Annu. Rev. Immunol.* 23, 1–21.
- Kitano, H., 2002. Systems biology: A brief overview. *Science* 295, 1662–1664.
- Koizumi, Y., Ohashi, H., Nakajima, S., *et al.*, 2017. Quantifying antiviral activity optimizes drug combinations against hepatitis C virus infection. *Proc. Natl. Acad. Sci. USA* 114, 1922–1927.
- Luzyanina, T., Mrusek, S., Edwards, J.T., *et al.*, 2007. Computational analysis of CFSE proliferation assay. *J. Math. Biol.* 54, 57–89.
- Lyons, A.B., 2000. Analysing cell division in vivo and in vitro using flow cytometric measurement of CFSE dye dilution. *J. Immunol. Methods* 243, 147–154.
- Macallan, D.C., Asquith, B., Irvine, A.J., *et al.*, 2003. Measurement and modeling of human T cell kinetics. *Eur. J. Immunol.* 33, 2316–2326.
- Macallan, D.C., Wallace, D.L., Zhang, Y., *et al.*, 2005. B-cell kinetics in humans: Rapid turnover of peripheral blood memory cells. *Blood* 105, 3633–3640.
- Mahgoub, M., Yasunaga, J.I., Iwami, S., *et al.*, 2018. Sporadic on/off switching of HTLV-1 Tax expression is crucial to maintain the whole population of virus-induced leukemic cells. *Proc. Natl. Acad. Sci. USA* 115, E1269–E1278.
- Martinez, F.O., Gordon, S., Locati, M., Mantovani, A., 2006. Transcriptional profiling of the human monocyte-to-macrophage differentiation and polarization: New molecules and patterns of gene expression. *J. Immunol.* 177, 7303–7311.
- Michor, F., Hughes, T.P., Iwasa, Y., *et al.*, 2005. Dynamics of chronic myeloid leukaemia. *Nature* 435, 1267–1270.
- Mohri, H., Bonhoeffer, S., Monard, S., Perelson, A.S., Ho, D.D., 1998. Rapid turnover of T lymphocytes in SIV-infected rhesus macaques. *Science* 279, 1223–1227.
- Mohri, H., Perelson, A.S., Tung, K., *et al.*, 2001. Increased turnover of T lymphocytes in HIV-1 infection and its reduction by antiretroviral therapy. *J. Exp. Med.* 194, 1277–1287.
- Neumann, A.U., Lam, N.P., Dahari, H., *et al.*, 1998. Hepatitis C viral dynamics in vivo and the antiviral efficacy of interferon-alpha therapy. *Science* 282, 103–107.
- Notta, F., Zandi, S., Takayama, N., *et al.*, 2016. Distinct routes of lineage development reshape the human blood hierarchy across ontogeny. *Science* 351, aab2116.
- Orkin, S.H., Zon, L.I., 2008. Hematopoiesis: An evolving paradigm for stem cell biology. *Cell* 132, 631–644.
- Perelson, A.S., 2002. Modelling viral and immune system dynamics. *Nat. Rev. Immunol.* 2, 28–36.
- Perelson, A.S., Nelson, P.W., 1999. Mathematical analysis of HIV-1 dynamics in vivo. *SIAM Rev.* 41, 3–44.
- Revy, P., Sospedra, M., Barbour, B., Trautmann, A., 2001. Functional antigen-independent synapses formed between T cells and dendritic cells. *Nat. Immunol.* 2, 925–931.

- Ribeiro, R.M., Mohri, H., Ho, D.D., Perelson, A.S., 2002. In vivo dynamics of T cell activation, proliferation, and death in HIV-1 infection: Why are CD4⁺ but not CD8⁺ T cells depleted? *Proc. Natl. Acad. Sci. USA* 99, 15572–15577.
- Schlub, T.E., Venturi, V., Kedzierska, K., *et al.*, 2009. Division-linked differentiation can account for CD8(+) T-cell phenotype in vivo. *Eur. J. Immunol.* 39, 67–77.
- Simon, V., Ho, D.D., 2003. HIV-1 dynamics in vivo: Implications for therapy. *Nat. Rev. Microbiol.* 1, 181–190.
- Takayanagi, H., 2007. Osteoimmunology: Shared mechanisms and crosstalk between the immune and bone systems. *Nat. Rev. Immunol.* 7, 292–304.
- Thomas-Vaslin, V., Altes, H.K., De Boer, R.J., Klatzmann, D., 2008. Comprehensive assessment and mathematical modeling of T cell population dynamics and homeostasis. *J. Immunol.* 180, 2240–2250.
- Wei, X., Ghosh, S.K., Taylor, M.E., *et al.*, 1995. Viral dynamics in human immunodeficiency virus type 1 infection. *Nature* 373, 117–122.
- Yamamoto, R., Morita, Y., Ooehara, J., *et al.*, 2013. Clonal analysis unveils self-renewing lineage-restricted progenitors generated directly from hematopoietic stem cells. *Cell* 154, 1112–1126.

Relevant Website

www.edanzediting.com/ac

Edanz Editing - Expert English Editing.

Biographical Sketch

Shoya Iwanami I am a graduate student in the mathematical biology laboratory, Department of Biology, in Kyushu University, Japan. I studied biology as an undergraduate student and joined the mathematical biology laboratory in 2015. My research interest is quantifying the biological characteristics of immunology, virus dynamics and stem cells using mathematical models.

Shingo Iwami I am an Associate Professor in the mathematical biology laboratory, Department of Biology, in Kyushu University, Japan. I have been working on theoretical analyses of virus infection *in vivo* and *in vitro* since 2007, covering various infections. My main interest is mathematical modeling and data analysis based on mechanistic models of virus infection dynamics.

Introduction

Bioimaging techniques were developed, which produce images from biological and medical specimens of entire organism, to single molecules and organs (Swedlow, 2012). Common biological imaging methods include magnetic resonance imaging (MRI) computed tomography (CT), X-rays, and positron emission tomography images. These methods are used especially in medical imaging for clinical purposes for vascular, tumor, anatomical, and metabolic imaging and they have a lower resolution. Confocal or two-photon laser scanning microscopy (Pawley, 2006), scanning or transmission electron microscopy, and atomic force microscopy (Keller and Stelzer, 2010) are imaging methods that have higher resolutions commonly used for visualization of cell structures, mapping cell surface, and discerning protein structure (Schmidt *et al.*, 2013). Both the medical and the high resolution imaging techniques are important sources for bioimaging techniques. Methods based on light microscopy are also important to understand the cell structure and function, and which lie between medical imaging and high resolution techniques. Imaging techniques such as photoactivation localization microscopy, fluorescence photoactivation localization microscopy, and stochastic optical reconstruction microscopy provide locations and localization uncertainties of individual molecules accurately up to 20 nm (Betzig *et al.*, 2006; Hess *et al.*, 2006; Rust *et al.*, 2006). Advancement in imaging technologies generates quantitative measurements from images that enhance knowledge and understanding of biological phenomena.

Advances in the field of microscopy in terms of three dimensional (3D) images provides better understanding of biological images (Schmid *et al.*, 2010; Mayer *et al.*, 2012; Guo *et al.*, 2013; Allen *et al.*, 2015). In four-dimensional (4D) imaging, time is the added dimension in the existing spatial dimensions that allows studying biological phenomena in four dimensions. 4D imaging is important in the field of medical imaging such as tumor detection (Handels *et al.*, 2007), changes in tissues intensities (Gorbunova *et al.*, 2012) and cell invasion (Kelley *et al.*, 2017).

Image processing can be used to determine whether an image included specific information or contains some specific objects, features, or activities. The identification task is solved automatically by using a digital image processing field with less human effort. The surge of complex biological and biomedical images leads to the existence of numerous image data analysis and informatics methods to extract quantitative information via segmentation and motion tracking, comparison, search, and management of the biological information of the images (Kvilekval *et al.*, 2010). Bioimage informatics techniques translate image data into valuable biological information (Peng, 2008). Bioimage informatics allows us to discover answers to biological problems from biological and medical specimens of the entire organism, single molecules and organs that would require higher cost to solve (Peng, 2008; Myers, 2012). Bioimage informatics research is heading towards automated invention of models of biological systems and machine learning methods (Murphy, 2014). Obara *et al.* (2012) established an image analysis method to identify and illustrate huge compound fungal networks using a graph representation that can be used for predicting the physiological performance of the fungal system. Puniyani and Xing (2013) used images of *Drosophila* embryo using a supervised learning method to understand gene interaction networks. Liu *et al.* (2012) developed a phenotype recognition model using zebrafish embryos' development response towards HTS toxicity. Aging is studied using phenotypes described in *C. elegans* upon different levels of mitochondrial alteration to develop an automated high-content strategy to identify new potential pro-longevity interventions.

Coelho *et al.* (2015) developed an automated quantification and identification of neutrophil extracellular traps (NETs) using supervised machine learning method that is important in controlling bacterial pathogens. Deep convolutional neural networks have been used for automated segmentation of bacterial and mammalian cells and identification of wild animal species in camera-trap images (Sadanandan *et al.*, 2017; Gomez and Salazar, 2016; Villa *et al.*, 2017). Perre *et al.* (2016) applied image analysis techniques to automatically identify invasive fruit fly species based on wing and aculeus images.

Mosleh *et al.* (2016) developed a program for automated cephalometric analysis to identify important landmarks point in the human skull that can also be applied to species identification. Computerizing cephalometric is important to lessen the time required for orthodontic procedures by providing X-ray enhancement, consistent measurements, presurgical simulation. Cephalometric analysis can be extended to include species classification.

Automated image recognition has been used to identify various species of algae and harmful species of algae to access environmental condition in aquatic habitat (Mosleh *et al.*, 2012; Verikas *et al.*, 2012; Luo *et al.*, 2011; Dimitrovski *et al.*, 2012; Coltelli *et al.*, 2014). Identification of flatworms is important for preservation of marine life was conducted by Kalafi *et al.* (2016) using images of hard parts of the haptor organs of flatworms with machine learning approach for classification. Nguyen *et al.* (2016) proposed a method SLIC-MMED (simple linear iterative clustering on multichannel microscopy with edge detection) that is able to segment muscle fibers and can be applied to stem cell regeneration and cell lineage tracing. Savkare and Narote (2012) and Rosado *et al.* (2016) used image processing approach to extract red blood cell and classify them as normal or malaria infected cell. Intarapanich *et al.* (2016) developed an object-based extended depths of field method to obtain a clear focus image of the foreground properties of the image which constitutes important features. This can be applied in medical diagnostic applications

such as diagnosis of malaria infection that can considerably lessen image processing time while preserving significant image features.

Quantitative information resulting from microscopy bioimages is important to enhance understanding of biological processes. With the advancement in imaging technology, image processing techniques are in dire need. There is vast amount of studies conducted on basic image processing, however there very is little information on image processing methods that involves bioimage informatics. Furthermore the absence of high-quality precise data sets for assessment of various common bioimage analysis methods is the main obstacle. A high-quality, well-defined data set for bioimage analysis consists of benchmarked data with well-defined verified and validated structure that is the gold standard for bioimage analysis. A benchmarked dataset reduces analysis and evaluation time and is important for assessing performance, consistency, and accuracy of methods in bioimage analysis (Gelasca *et al.*, 2009).

The important procedures of bioimage informatics include bioimage acquisition, visualization, analysis, and management. Bioimage informatics also covers creation and understanding of biological information generated from biomages using image processing, data mining, machine learning, and pattern recognition methods.

In this article, image acquisition and processing, image segmentation, feature extraction, feature selection, classification registration, and annotation are discussed. Examples and an appropriate case study are provided in each section.

Image Acquisition and Preprocessing

Digital images are produced by image equipment and the process of transferring the analogue image into digital images using imaging devices is image acquisition. The main unit of digital images is pixel which represent the values correspond to the light intensity in one or more spectral bands. Image acquisition is a crucial stage in bioimage informatics related research and automated systems (Coltelli *et al.*, 2014). Captured microscopy images usually suffer from problems such as varying in colors, darkness, resolution, contrast, and brightness that is related to light source, lens, and method of capturing. Microscope images also include unavoidable scum existing beside target cells, and some holes or small objects that strongly affect image quality. Images may contain unwanted areas, shadows, and appear blurry. The preprocessing of captured images is a preparation and treatment process used to enhance the features of images to produce clearer details, remove noise, remove intelligibility of images, and improve the overall appearance of images. Image enhancement is defined as the conversion of an image quality to create a clear image and ensure the accuracy of the process of segmentation and feature extraction. Some common techniques used in image enhancement include image conversion to grayscale, filtering, histogram conversion, and color composition. Kho *et al.* (2017) converted the RGB value of a leaf image to HSV value to remove shadow and to ensure clear image contrast for image segmentation and feature extraction as depicted in Fig. 1.

Differences in light or brightness can be reduced using histogram equalization method and contrast enhancement in images can be carried out by stretching the histogram of digital image (Gonzalez and Woods, 2007). Histogram equalization can be applied to enhance the contrast of the color image intensity before the image is converted into a gray scale image. The frequency occurrence of pixel intensities given by the histogram image intensity was adjusted to increase image contrast. Histogram equalization is one type of gray scale conversion; it used to convert the histogram of the original image into an equalized histogram. An accumulated histogram was calculated from original image and then divided into a number of equal regions. The corresponding gray scale in each region was assigned to a converted gray scale. The effect of histogram equalization was the enhancement of image parts that have more frequency variation, whereas parts of an image with less frequency were neglected. Fig. 2 shows a comparison between the original and converted images after histogram equalization was applied, and shows a comparison between the histogram of the original image with the histogram of a produced algae image.

Images acquisition condition such as parameter and equipment should be consistent and images should be of high-quality with clear details and less noise. Applying resampling method, reducing sensor noise and contrast enhancement can increase the quality of image being captured. Images captured using different equipment should be labeled with scales. Problem related to using different equipment in medical images may produce images that are not alike in appearance. Images can contain noise such

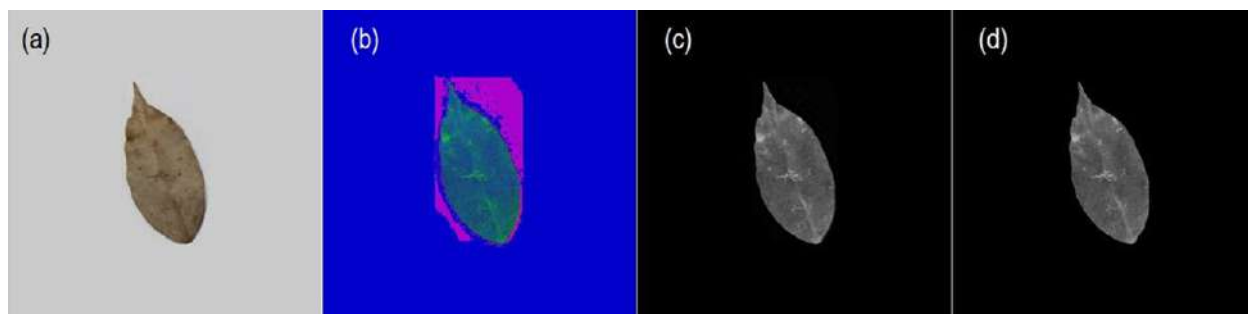


Fig. 1 Shadow removing image preprocessing. Thin streak of shadow can still be seen in (c), which is then removed in (d).

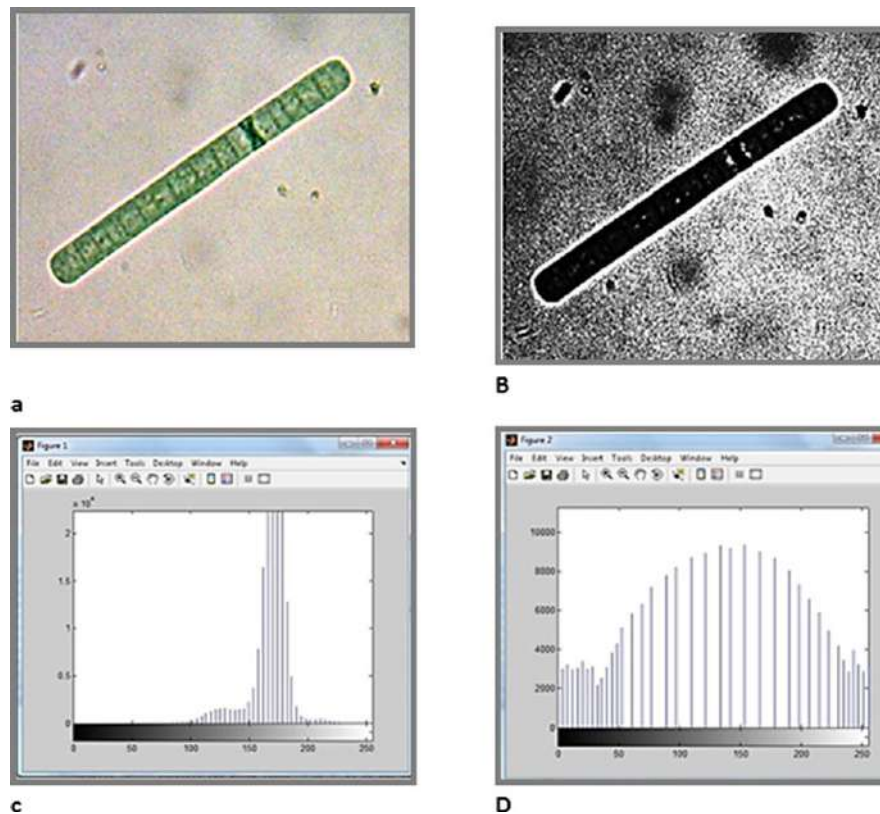


Fig. 2 Examples: Histogram equalization application process as (a) Original image of *Oscillatoria* sp., (b) Result image after equalization, (c) Accumulated histogram of original image, (d) Accumulated histogram for grayscale image.

as Gaussian film grain, nonisotropic, speckle and periodic noise, and salt and pepper noise from imaging equipment that can reduce the image quality especially salt and pepper noise in medical images (Sánchez et al., 2012). Noise is added into an image at the time of image acquisition or image capturing. Noise reduction is necessary to preserve the quality of image. Noise reduction filters are comprised of linear, nonlinear, adaptive, median, and fuzzy filters (Mythili and Kavitha, 2011). The main purpose of the filters is to produce a high-quality image by sharpening, edge highlighting, and contrast improvement. High-quality images are important for image processing especially medical images. Some of these developed filters to enhance image quality are median filter, a nonlinear filter used mostly to reduce image noise in better way than other methods such as convolution approach (Lim, 1990). Fig. 3 illustrates application of median filter to algae species. Median filters have been applied by Kho et al. (2017) to remove noise from a leaf image and Barry and Williams (2011) for fungal network extraction.

The Kalman filter an adaptive filtering approach for color noise based on the correlative method (Xinding et al., 1996). The Gaussian filter is an image-blurring filter that uses a normal distribution, also called Gaussian distribution, for calculating the transformation to apply to each pixel in the image. It is used widely to filter medical images to remove fine detail and noise (Yue et al., 2006). Gaussian filters are also used to smooth the input image (Yue et al., 2006). Gaussian filters can separate the holes from the noise in gray image region filters to smooth gray images for contrast enhancement. Gaussian filters are widely used for detection of edges and peaks in images. Unsharp filters are used to filter low spatial frequencies from the image data to enhance the image. An X-ray or clinical medical image generally suffers from low contrast quality and degradations that vary from one region to another. Unsharp filtering or masking is an edge-enhancement technique by isolating the edges in an image, amplifying them, and then adding them back into the image. The unsharp filter and Gaussian filter are the most common filters used for medical image filtering (Chalazonitis et al., 2003).

In X-rays of hard and soft tissue in a cephalometric system image quality influences the accuracy of locating landmark points automatically. High-quality images are important to identify landmark points automatically as manual identification is not accurate (Mosleh et al., 2016). X-Cephalometric images are the basis of cephalometric measurement and analysis, which have been widely applied in the fields of orthodontics and provide a scientific basis for making the diagnosis, treatment, and accurate and effective prediction. The cephalometric system analysis requires good quality X-rays. If the X-ray is poorly exposed, the lines are fragmented. The critical lines defining the landmarks may not be tracked properly. The quality of images directly influence accuracy of locating landmarks. The original image must be filtered to ensure the accuracy of locations and measurements. During sampling and transmission, images are often degraded by noises that originated from a multiplicity of sources. These noises are colorful and their variances are not known beforehand. The original images must be filtered to ensure the accuracy of location and measurement. Mosleh et al. (2016) applied unsharp and Gaussian filters to enhance image properties for feature extraction. Unsharp filters in this

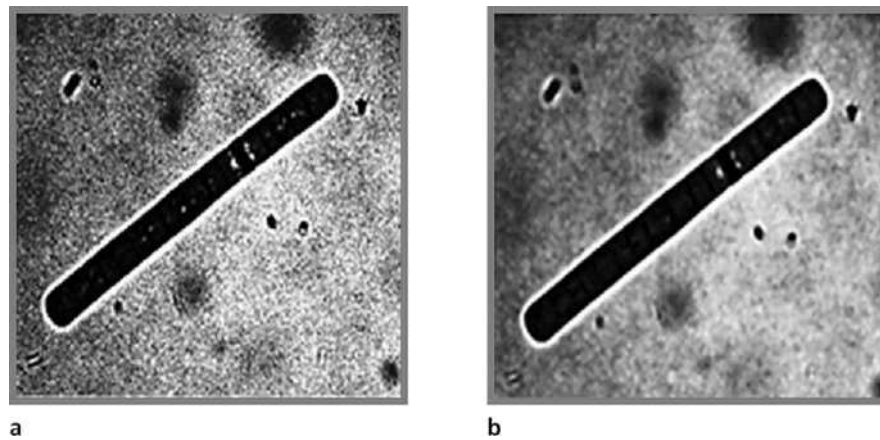


Fig. 3 Median filter application (a) *Oscillatoria* sp. image in gray scale before the process (b) *Oscillatoria* sp. after median filter was applied.

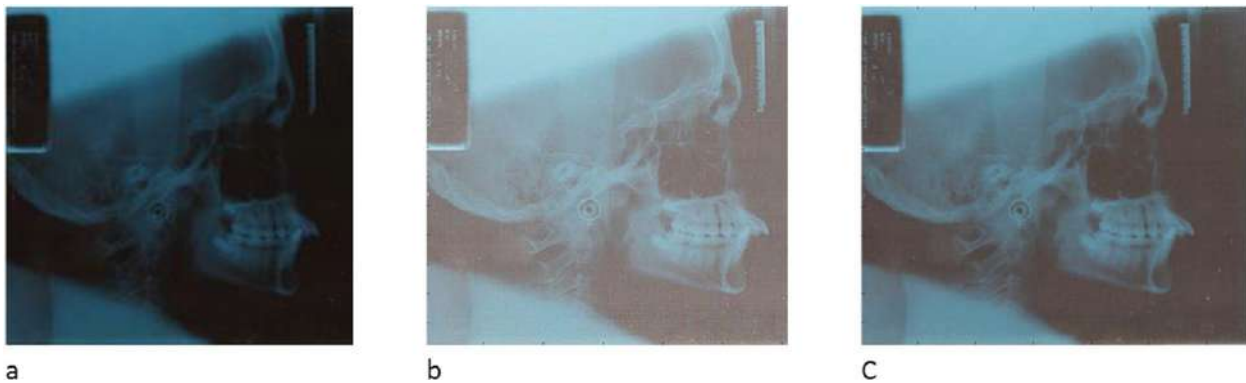


Fig. 4 Illustrates application filter for image enhancement (a) original image (b) unsharp filter, (c) Gaussian filter.

study are used for sharpening of X-ray images in the presence of low noise levels via an adaptive filtering algorithm. Adaptive filtering using these kernels can be performed by filtering the image with each kernel, in obtaining the output image with sharp filter. Gaussian filters are designed to give no overshoot to a step function input while maximizing the rise and fall time. This behavior is closely connected to the fact that the Gaussian filter has the minimum possible group delay. The gradient operators in a Gaussian filter enhance image edges and at the same time enhance noise. Emphasizing the fine details of a radiographic image while suppressing noise is possible by employing a median filter prior to applying a gradient operator. Approximations to the pill box blur and Gaussian low pass. The idea of Gaussian smoothing is to use this 2-D distribution as a point-spread function, and this is achieved by convolution. Before the convolution is performed a discrete approximation to the Gaussian function was produced. In theory, the Gaussian distribution is nonzero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero when it is more than three standard deviations from the mean (Davies, 1990).

Gaussian filter is generated with some parameters as a 2-D array because the X-ray image is a 2-D array. Input is the unfiltered X-ray image and the output is the filtered X-ray image. Both implemented filters can work in real time at system runtime.

Fig. 4(a) illustrates X-ray images without and with filtering. The low intensity contrast that is almost hidden in the background in **Fig. 4(a)** can be distinguished clearly in **Fig. 4(b)** and **Fig. 4(c)**. The image in **Fig. 4(a)** is considered a slightly overexposed cephalometric radiograph. The contrast of the bone anatomical structures is quite satisfactory, but soft tissue tends to mix with the background in some zones. After the application of unsharp filter as shown in **Fig. 4(b)** or Gaussian filter as shown in **Fig. 4(c)**, the bone contrast is greatly improved, and anatomical structures become more visible.

Image Segmentation

Microscopic images normally contain different objects, therefore image processing is required to isolate the image into different object components. Detection processes for specific image points or regions are performed to determine the object for further processing. This process of isolating and dividing the image into a subimage, where each image contains at least one object, is called segmentation. Segmentation is an essential phase in image processing, which is carried out after image processing and before feature extraction and classification (Haralick and Shapiro, 1992). The image segmentation process is used to isolate

individual objects in captured images. The segmentation process is important for quantification and analysis of bioimages. There are numerous image segmentation methods such as the thresholding based method, clustering based method, edge based method, region growing based method, watershed method contour based method, and model based method (Basavaprasad and Ravi, 2014). The thresholding based segmentation method is a common method for image segmentation that is applied to the image pixels. This method can be categorized into otsu, global, and local thresholding techniques. Rosada *et al.* (2016) reviewed numerous methods involved in automated malaria cell segmentation where the thresholding based method is most commonly applied. In bioimage segmentation the otsu based method has been applied in identifying *Drosophila* egg stages (Jia *et al.*, 2016), identification of herbs leaves and plant disease identification (Isnanto *et al.*, 2016; Akhtar *et al.*, 2013).

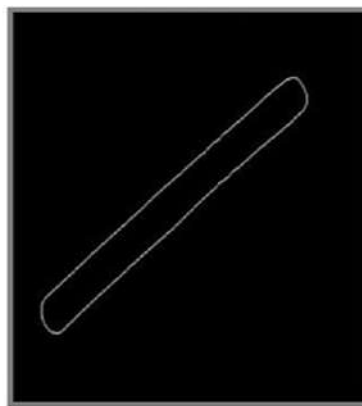
Edge based segmentation is another common method and is implemented using filters such as Canny's, Sobels, Laplacian, Robert, and gradient filters (Gonzalez and Woods, 2007). A Canny edge detector is considered as the most powerful edge detector for image segmentation (Canny, 1987). It is used to identify discontinuities in an image intensity value or the edge of the image and it has been used widely in species identification (Kalafi *et al.*, 2016; Mosleh *et al.*, 2012; Murat *et al.*, 2017). Both Canny and Sobel have been applied in automatic identification of algae due to substantial edges and shapes of the species (Santhi *et al.*, 2013).

Thresholding based method together with edge based method have been applied in Munisami *et al.* (2015) for plant leaf recognition and brain tumor detection (Manasa *et al.*, 2016). Identification of bioimage segments is often more effective when making use of boundaries and shape information extracted by segmentation methods. The Grabcut algorithm (Rother *et al.*, 2004) is a segmentation technique used in automated identification of species systems (Hernández-Serna and Jiménez-Segura, 2014) to remove background. In this technique, hard segmentation made by iterative graph-cut optimization is combined with border matting to get rid of mixed and blurred pixels on the boundaries of an object.

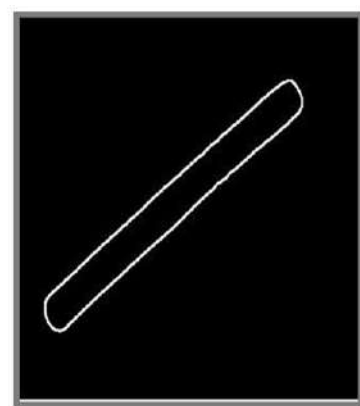
Overlapping objects in microscopic images makes object detection difficult. There have been various methods to separate overlapping objects including watershed algorithm, edge detection method, morphological erosion, active contour method, and others. The application of the edge detection method together with morphological erosion and dilation are implemented in



a-BW Image for *Oscillatoria* after Canny Edge detection.



b-Open Binary process to remove small objects



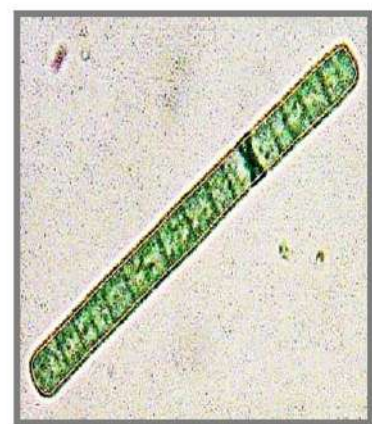
c-Dilation process Detected Edge



d-A Flood Fill Operation to fill object area



e-Erosion Operation to focus more on detected object



f-Outline same are on original image

Fig. 5 Image samples for morphological operation on *Oscillatoria* sp.

Mosleh *et al.* (2012) in automatic algae detection. In morphological operations, the value of each pixel in the output image was compared with the corresponding pixel in the input image of its neighbors. Fig. 5 illustrates application of morphological operation and Canny edge detection.

The edge detection method does not perform well in instances where the overlapping objects are intensely textured or there is no noticeable intensity variance between the overlapping. The active contour and snake methods (Kass *et al.*, 1988) require high computer processing capacity and it not suitable for large objects. It has been used by Chopin *et al.* (2016) to determine in identification of wheat leaves and Sabeena Beevi *et al.*, (2017) in identification of mitotic nuclei from microscopy images of breast histopathology slides.

The watershed method is considered as an effective and efficient method to separate overlapping objects (Savkare and Narote, 2012). Watershed method segmentation is also used to detect and extract a graph representation of complex curvilinear networks and overcomes a critical bottleneck in biological network analysis (Obara *et al.*, 2012). Segmentation of globular object can be challenging due to object irregularity. This requires additional step such as hole filling before applying watershed segmentation method (Long *et al.*, 2007). Yang and Ahuja (2014) used local density clustering to identify pixels that matter before the watershed algorithm is used to separate the overlapping objects in blob or granular object recognition for cell/nuclei segmentation. Automatic segmentation of cell nuclei in microscopic images is highly desirable, however it is difficult due to huge intensity inhomogeneities in the cell nuclei and the background. Unsupervised methods are popular for cell nuclei segmentation thresholding, *k*-means clustering, watershed, active contour, level set and graph-based models. Thresholding, *k*-means, and watershed are applicable when there is great contrast between object and background and it is not suitable with images with inhomogeneities.

Other techniques such as *k*-means clustering, which attempts to minimize the variance of intensity values within a predefined number of clusters, and Gaussian mixture models (GMM) (Permuter *et al.*, 2006), which use probability distributions to classify pixels as foreground or background. GMM has been used in the brain segmentation of magnetic resonance images (MRI) due to ease of use and high contrast between different tissue classes of the brain (Song *et al.*, 2014). Combination of *k*-means clustering and otsu method was applied in identification of plant leaf disease after RGB color transformation on leaf. *K*-means clustering was used to divide a leaf image into segments to identify disease where the leaf is infected by more than one disease (Al-Hiary *et al.*, 2011).

Chopin *et al.* (2016) applied hybrid segmentation technique for whole plant such as wheat image segmentation comprising the leaf tips and twists. Initial segmentation was done using GMM, *K*-mean, and multidimensional histogram thresholding and active contour model by Kass *et al.* (1988), and applied to enhance the segmentation after identification of control points in the image.

Before feature extraction is applied, images, especially species or plant images, need to be aligned. Mosleh *et al.* (2012) applied techniques for image objects alignments. This method also has been adopted by Kho *et al.* (2017) in *ficus* plant leaf identification. This technique performs automatic rotation for objects to be aligned with the horizontal axis, which increases the classification accuracy. A small routine is developed to obtain the angle of inclination for an object automatically. This routine is then used for rotating image objects to be aligned horizontally as shown in Fig. 6.

The angle of inclination is calculated automatically by obtaining the longest path between each two points on the object boundary. Identified point $P_1(X_1, Y_1)$, $P_2(X_2, Y_2)$, and Origin point $P(0, 0)$ are used to determine the angle of inclination using the following equation:

$$\theta = \tan^{-1}(m_1 - m_2)/(1 + m_1 * m_2) \quad (1)$$

where m_1 and m_2 are the slopes of lines that form the angle that is obtained using the following equations:

$$m_1 = (Y_2 - Y_1)/(X_2 - X_1) \quad (2)$$

$$m_2 = (Y_1 - Y_0)/(X_1 - X_0) \quad (3)$$

This routine is designed to align the rotated shape into horizontal lines, which will ease the feature extraction process, and improve the accuracy and performance of the recognition process by ensuring that all the extracted features are calculated with similar positioning of the object coordinates.

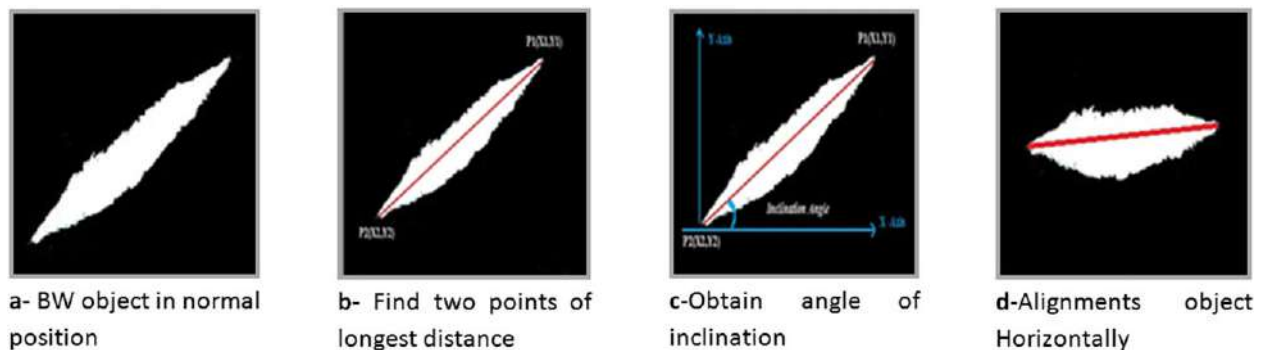


Fig. 6 Sample steps of auto-orientation method.

Feature Extraction

Features extraction and selection is a crucial step as features extracted from bioimages will be used for bioimage identification. Selection of significant features from bioimages is important as the number of features extracted can be large and this can lead to heavy computational effort. Feature selection allows an optimized number of features being selected. The success of bioimage feature extraction and selection techniques is determined by the quality of the extracted data and the type of classifiers used in bioimage identification and recognition (Chora's, 2007; Kiranyaz *et al.*, 2011).

Feature extraction is a process to obtain useful information from the segmented bioimages that can be used to classify or identify them. Feature extraction is necessary for successful bioimage classification. Features used in bioimage classification normally are a combination of several types of features (Song *et al.*, 2016). There are three distinct features: Texture, shape, and color (Islam *et al.*, 2008; Pin Tian, 2013).

Color is the most significant feature for image classification (Seyyid, 2015; Chora's, 2007). Color histograms, coherence vector, moments, correlogram, scalable color descriptor, dominant color descriptor, and structure color descriptor are various methods to extract color features in an image that are based on mean, skewness, and standard deviation of an image pixel. Color histogram is the most common method to detect color distribution in an image that meets basic requirements meanwhile color moment is the simplest method to implement (Pin Tian, 2013; Seyyid, 2015). In automated identification of zebrafish embryo development in response to toxicity, color features such as scalable color descriptor, color layout descriptor, and color histogram with texture features gave successful results (Liu *et al.*, 2012). Color variance is an important feature in skin cancer detection (Bhuiyan *et al.*, 2013). Chopin *et al.* (2016) used color of leaf together with the direction and magnitude of leaf edges in whole plant identification.

Texture features are significant as well however it is used with other features such as color and shape to enhance its effectiveness. Texture features are important in digital image identification especially in plant leaf identification (Brilhador *et al.*, 2013; Jamil *et al.*, 2015). Combination of texture and shape feature produced better identification in plant leaf identification than using combination of texture, shape and color (Jamil *et al.*, 2015). Texture features should be used as an input when color or shape features are not distinctive enough on the plant species images (Camargo and Smith 2009). Textures features are categorized into statistical and structural features such as Haralick, Gabor, and wavelet and Fourier transform, co-occurrence matrices, shift-invariant principal component analysis, and Tamura features (Chora's, 2007). The Gabor filters compared to gray level co-occurrence matrix for image segmentation and classification can be utilized as is or converted into feature vectors (Kumar *et al.*, 2015). Al-Hiary *et al.* (2011) used texture features derived from co-occurrence matrix with color features to identify plant disease. Camargo *et al.* (2009) used color features together with shape and texture features to enhance identification of visual symptoms of plant disease. Akhtar *et al.* (2013) combined Haralick texture features (Haralick *et al.*, 1973), discrete cosine transform (DCT), and discrete wavelet transform (DWT) features for automatic identification of plant disease.

Combination of Haralick texture features, Pearson correlation between channel intensities, and SURF features (Bay *et al.*, 2008) was used to automatically identify NETs images (Coelho *et al.*, 2015). Texture feature based on DWT is important for bioimage classification of cancerous cells. However in breast cancer cell diagnosis complex daubechies wavelet transform (CDWT) performed better than DWT with color features (Niwas *et al.*, 2013). In identification mitotic nuclei from images of breast histopathology shape and textures features are used (Beevi *et al.*, 2014). Morphological changes of *Caenorhabditis elegans* (*C. elegans*) is used for studying aging due to the small size, transparent body, well-characterized cell types and lineages. An automatic image processing method for both normal and abnormal structures is contrasted using geometric, intensity, and texture features that describe the properties of nuclei in *C. elegans* (Zhao *et al.*, 2017).

Shape feature is important in bioimage object identification. The shape feature extraction method can be categorized into contour based and region-based techniques. Shape features typically comprises of geometric features such as aspect ratio, circularity and irregularity. Advance method for shape features is feature moment such as Zernike moment (ZM) a region-based descriptor. Detailed descriptions of shape based features are explained in Mingqiang *et al.* (2008). Shape, size scale-invariant feature transform (SIFT) are used in identification of cell nuclei in microscopic images (Song *et al.*, 2013). Kalafi *et al.* (2016) extracted various shape features from anchors and bars of monogeneans such as Euler number, perimeter, area, density, and bounding box. Automated stage identification of the *Drosophila* egg chambers includes morphological shape features such as egg chamber size, oocyte size, egg chamber ratio and distribution of follicle cell (Jia *et al.*, 2016).

Shape feature is the most common feature used in plant leaf identification (Jamil *et al.*, 2015). Spatial interrelation shape feature such as convex hull, morphological shape feature of plant, distance features, and color histogram have been used in plant leaf recognition (Munisami *et al.*, 2015). Convex hull allows segmentation of a complex object into a polygon without holes that are convex in nature and shape is very essential in object recognition (Jayaram and Fleyeh 2016). An herbal plant leaf identification system (Isnanto *et al.*, 2016) is based on the shape of the herbal plants' leaf region-based invariant feature extraction or Hu's seven moment invariants. Hu's seven moment invariants are indifferent against translation, scaling, and rotation, as well as in mirror position of the herbal leaf. A combination of shape features using SIFT, color using color moment, and texture feature using segmentation-based fractal texture analysis (SFTA) in herbal plant identification gave better results compared to using a single feature (Jamil *et al.*, 2015). A combination of shape features, texture, and color features have also been applied successfully in classification of plant leaves and algae recognition (Kho *et al.*, 2017; Mosleh *et al.*, 2012).

Below are examples of shape feature extraction based on Mosleh *et al.* (2012). In this study, a specific feature set was selected to obtain the essential characteristics for the selected algae. Feature extraction process is implemented to extract some parameters



Fig. 7 Example of area extraction on algae species.

from both binary and color images of algae including shape index, area, perimeter, minor and major axes, centroid, equivalent perimeters, bounded box, and Fourier spectrum with principal combination analysis (PCA). Other features were derived from the main feature to obtain the relation between shape parameters such as that between area and parameters.

(1) *Area*

The area represents the actual number of white pixels in the selected region. It is calculated by using the number of white or "1" pixels inside the image object as shows in Fig. 7. Area is used as a parameter in classifying operations because it gives an indication of the size of objects.

(2) *Perimeter*

The perimeter of the object is the summation of the distance between each adjoining pair of pixels around the object border, as shown in red pixel in Fig. 8.

(3) *Major and minor axes*

Major and minor axes are extracted where two points are identified automatically by calculating the maximum distance between given points in the objects vector. The major axis represents the line segment connecting between the base points in the X axis, whereas the minor axis represents the maximum width which is perpendicular to the major axis. The major axis can be used as object length and minor axis as object width as represented on Fig. 9.

(4) *Object width factor*

Object width factor is a customized shape feature developed for algae feature extraction that can be applied for any plant images. It is calculated by slicing across the major axis and parallel to the minor axis, the feature points are normalized into a number of vertical strips, and for each strip, the ratio of strip length to the object width is calculated using the following equation: $R_c = W_c/L$, where R_c is the ratio at column c , W_c is the width of object at column c , and L is the object length as shown in Fig. 10. This feature is used to differentiate between some algae that is identical in width and length, and have some essential variations in width at different positions such as *Navicula*, which has similar value of width as *Oscillatoria* at the middle position; however values of width vary in different positions as illustrated in Fig. 10.

(5) *Equivalent diameter*

Equivalent diameter is the feature that determines the diameter of a circle with the same area of a segmented image. It is computed as:

$$\text{EquDi} = \sqrt{4 \cdot \text{Area} / \pi}$$

(6) *Euler number*

This feature is suitable to identify the number of holes inside the image object. The Euler function segments the image before it is filled by holes. The Euler number is equal to the number of objects in the region minus the number of holes in those objects. Fig. 11 shows one example of using Euler number to improve classification method of algae and plant leaf recognition.



Fig. 8 Perimeter feature extraction.

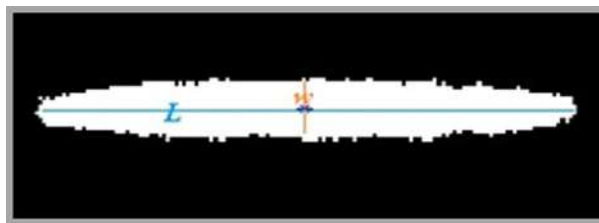


Fig. 9 Major and minor axes.



Fig. 10 Slicing process on *Navicula* and *Oscillatoria*.

(7) Bounding box

Bounding box is identified as the smallest rectangle that contains the region. The box has four feature parameters, including the left corner point of the rectangle (X_1 , Y_1), length, and width of bounding rectangle. **Fig. 12** shows an example for the bounding box object. The rectangle bounding the object is used to compute the extent parameter, which represents the proportion of pixels in the bounding box within the region. The results are obtained by dividing the object area with the area of the bounding box.

(8) Centroid

The centroid is the center of mass for the segmented region, and helps us determine the object center coordinates. The elements of the centroid are two points, the x - and y -coordinates, of the center of mass. **Fig. 13** represents the centroid of an object as small blue star in the center of the object.

The centroid as parameter does not provide a distinguishing feature for selected algae however it can be used to derive a new feature to enhance the classifications method. The centroid coordinate is used to extract the length and slope of the line that connects between the centroid coordinate points and the left corner point extracted from the bounding box; the line is shown also in **Fig. 13** in blue.

(9) Derived features

Selected feature ratios are used for improving the classifying process such as the ratio of the area and perimeter (perimeter/area), ratio of perimeter and object length (perimeter/length), ratio of perimeter and object width (perimeter/width), and ratio of minor to major axes (minor/major).



Fig. 11 Euler number for *Chroococcus* sp.



Fig. 12 Bounding box feature example.



Fig. 13 Illustrates centroid and slope lines.

(10) *Shape index*

Extracted shape index is one of the customized features constructed in this study. Algal taxonomy based on algal shapes is used to develop a new function that can categorize algal shape into three different types: Circular, spiral, and irregular. This function is designed to identify algal shape type and provided us with a new feature called shape index, which is equal to "1" if the shape is circular, "0.5" if the shape is cylindrical, "0" if the shape is spiral, and "−1" if the shape is irregular. This classifier is used to improve accuracy rate and optimize the time of recognition process (Abdullah, 2013). The results obtained from shape extraction will be included as one of the input parameters for algal classification using MLP. The shape index feature can be used as an early stage classifier for the algal taxonomy area. The measurement of the shape index is obtained by evaluating the roundness or cylinder of the objects as described in the following steps:

- Comparing the longest and shortest diameters for the object.
- Comparing the area with the formula $\pi \times r^2$.
- Comparing the perimeter with the formula $2\pi \times r$.
- Eccentricity is also used to improve the results of shape index measurements. Eccentricity refers to the ratio of the distance between the foci of the ellipse bounded the object and its major axis length.

(11) *Entropy of gray image*

Entropy is defined as the statistical measurement of randomness that can be used to characterize the texture of the input image. Entropy is used to extract texture features on the gray scale image. During segmentation processes, the detected object is used to obtain the outline of both color and gray images. Entropy is implemented in our system to obtain one output feature from the outlined gray image.

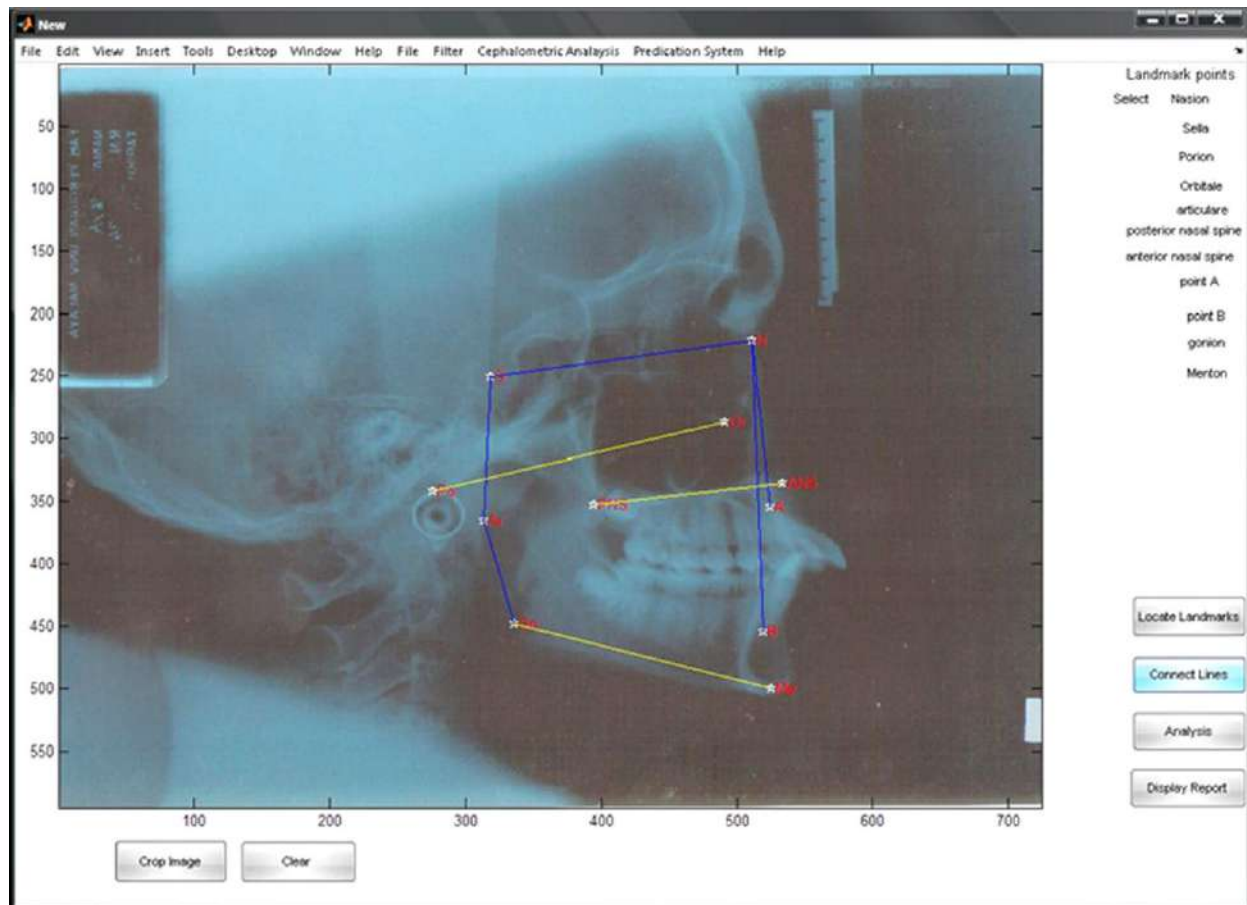


Fig. 14 Illustrates customized features extraction.

Morphological shape features and Euclidean distance are extracted from a graph representation of complex curvilinear or fungal networks. The approach is generic and can be applied to a wide range of biomedical images (Obara *et al.*, 2012). To further improve efficiency of feature descriptors for particular problem areas, tailored features have been constructed manually for feature extraction from bioimages (Song *et al.*, 2016). Some of the customized shape features for algae and plant leaf detection by Mosleh *et al.* (2012) have been discussed such as object width factor and shape index. Mosleh *et al.* (2016) extracted morphological or shape features for a cephalometric identification system. Tailored feature extraction methods have been constructed to extract line, line angles, and line length. Two points in a binary image are connected if a path can be found between them along which the characteristic function remains constant. Since every selected landmark point has a position in a coordinate system (X, Y), it can be used as parameters to create a line between two selected points as shown in Fig. 14. To perform this operation is a new matrix, which consists of all points between the two points where each generated point is represented by the shortest path in this matrix. Angle measurements are essential for any cephalometric measurement process. The angular measurements module is designed to calculate the size of these angles. Angle is a result of intersection between two lines in a plane, or via connecting all three points. In the two-dimensional space, the angle θ between two lines is the slope of a straight line in the XY plane represents changes of the ordinate Y of the point of the line per a unit-change of the abscissa X of the point. It requires that the line is not vertical. However, in this model angles are considered dimensionless, since they are defined as the ratio of lengths, the unit of measurement is radian. All angle measurements are converted to degree scale instead of radian scale because the degree scale is more standardized and easy to understand for orthodontics specialists. The line measurement of the distance between two points or pixels is constructed using trigonometry given by Pythagoras' theorem, the linear measurements are converted from pixel to inch. Cephalometric line measurements are important because they give information about the skeletal relation. Image resolution and screen resolution is used to calculate the linear measurements in inches. By considering the factor scales, all image types, whatever their resolution, can be analyzed and measured. Scale factor is calculated in our system to minimize the errors of measurement.

Some bioimage software such as BIOimage Classification and Annotation Tool (BIOCAT) and CP-CHARM are image based classification algorithms that are able to extract features without performing segmentation. CP-CHARM algorithm is based on WND-CHARM (Weighted Neighbor Distances using a Compound Hierarchy of Algorithms Representing Morphology) (Orlov *et al.*, 2008). CP-CHARM algorithm is capable of extracting morphological features of an image minus the necessities of segmentation. Features extracted from CP-CHARM are as follows: Moments features, texture features (Haralick, Tamura), edge

features, high contrast features, Gabor features, polynomial decompositions (Fourier, Haar wavelet, and discrete Chebyshev transforms), pixel statistics (histograms, moments), texture measurements, and image levels (transforms and compound transforms) (Uhlmann *et al.*, 2016). CP-CHARM is easier to use and it is an effective substitute to more comprehensive and difficult tools such as BIOCAT (Zhou *et al.*, 2013). Features extraction in BIOCAT comprises of 14 different features such as texture features using DTW, morphological features that comprises of Hu moments, ZMs, Hessian features for identifying tubular features, Gaussian derivatives, and Laplacian features and object statistics (Zhou *et al.*, 2013).

Feature Selection

Feature selection is an important method to construct a classification model that produces higher accuracy rates in bioimage recognition or classification. Feature selection eliminates features that are uninformative, noisy, or redundant, which leads to a dataset that is expressed with important information that is essential for better classification with controlled or little loss of information.

The importance of feature selection method is as follows: It can avoid overfitting of the classification model, provide better understanding into the underlying processes that generated the data, improve model performance, and lower processing time. Feature selection method can be classified into the following three methods: Filter, wrapper, and embedded methods (George and Raj, 2011).

Filter method uses ranking of features based on univariate metric. The top ranking features are utilized in model construction whereas the remaining low ranking features are eliminated. The method is only dependent on training data for feature selection, therefore the results from filter method does not affect model classification. Filter methods evaluate the significance of features by considering solely inherent properties of the data. Filter methods computational requirement are low and are independent of the classification algorithm. Therefore feature selection needs to be executed only once before carrying out classification (Saeys *et al.*, 2007). Brilhador *et al.* (2013) applied correlation feature selection (CFS) method for plant image classification with the Image-CLEF dataset. CFS is an automatic filter based method that identifies features that have no correlation with each other but are greatly correlated with a classification problem (Hall, 1999).

The wrapper method applies a learning algorithm to rank features based on their importance to classification model performance. It is a simple and convenient method of feature selection applicable irrespective of the selected learning algorithm. The learning algorithm is considered as black box. The wrapper procedure involves in using the classification model performance of a particular learning algorithm to evaluate the significant importance of the features. Specific criteria needs to be determined such as on search procedures of features subsets, assessment of machine learning algorithm performance and the type of features to be used. Performance evaluation of the learning classification algorithm is carried out using a cross validation method. Some examples of common classification models are decision trees, naive Bayes, and genetic algorithm (GA). Yusof *et al.* (2013) applied kernel based GA for feature selection from tropical wood texture features. The application of GA produced better results compared with a classification model that did not apply any feature selection. A disadvantage of the wrapper method is the high computational requirements. However this problem can be solved using effective search approaches, which do not sacrifice classification or predictive performance. Such approach involves greedy search strategies; forward selection and backward elimination are computationally beneficial and robust against overfitting. In forward selection, model performance is estimated by adding unselected features to the features subsets. Features that improve model performance are selected. Forward selection procedure is terminated when the expected performance of adding any feature is less than the performance of the feature set already selected. Whereas backward elimination model performance is accessed using all available features and gradually eliminates the least significant ones. The wrapper method is easy to use but embedded methods that incorporate feature selection in the training process might be more effective as there is no need for data splitting into the training and validation set. An example of embedded method is decision trees such as classification and regression trees (CART), which is equipped with a feature selection mechanism (Breiman *et al.*, 1984). The feature selection in an embedded method is built into the classification method. Embedded methods are explicit to a particular learning algorithm. Embedded methods comprise the interaction with the classification model. Examples of embedded methods are random forests (RFs), support vector machine (SVM) and artificial neural network (ANN) (Guyon and Elisseeff, 2003; Saeys *et al.*, 2007). The RF variable importance method has been applied in feature selection of plant leaf; variables with higher mean decrease accuracy are kept and retrained for better classification accuracy (Kumar *et al.*, 2015).

An alternative set of methods are the ones that construct new features from existing features that possibly could lead to loss of information. The methods are linear discriminant analysis (LDA) and principal component analysis (PCA) (Lewis *et al.*, 2012). Landmark localization uses PCA to extract landmarks on the surface of the human face using shape and texture information from 2D data human facial images (Guo *et al.*, 2013). Mosleh *et al.* (2012) in his study applied PCA to extract feature texture for algae classification. CP-CHARM bioimage classification software also utilizes PCA as its feature selection algorithm (Uhlmann *et al.*, 2016). Clustering based algorithms also have been used for feature selection. The most common algorithms comprise *K*-means and hierarchical clustering. Clustering is an unsupervised learning method. Kalafi *et al.* (2016) applied LDA for feature selection in fully automated classification of monogeneans at the species level. Huh *et al.* (2009) applied extension of LDA method for yeast image classification. However this method only works well when the numbers of classes are large as it tends to over reduce features. Stepwise discriminant analysis has been proven to work well in identifying protein in human cancer cells (Xu *et al.*, 2014). Bioimage classification software BIOCAT, comprises of the Fisher's linear discriminant feature selection method.

Classification

Machine learning and data mining techniques have the ability to accurately and successfully analyze substantial amounts of image data. Many studies have demonstrated the benefits of applying machine learning classification techniques to classify images using features extracted from them. Some of the common classification approaches are SVM, LDA, the K-nearest neighbor (KNN) classifier, ANNs, RF and decision tree classifier, naive Bayes (Abbas *et al.*, 2015). These classification methods for cell nuclei classification normally classify images based on variance in intensities of foreground and background histograms. The classification model performance is subjected to the variances in intensity between image background and foreground. Features such as local Fourier transform, spatial information, shape, and morphological features are normally used for cell nuclei classification (Song *et al.*, 2013).

SVM classifier separates data points of different classes to achieve largest margin between the classes. The SVM classifies the data by finding an optimal hyperplane that is the one with the largest margin between the classes. Margin is defined as the distance from the decision surface to the support vectors (Vapnik and Vapnik, 1998). RF is a nonparametric approach that builds an assembled model of decision trees during the model training. The output is the mode of the individual trees (Breiman, 2001). ANNs are a machine learning method that processes information by adopting the way in which the neurons of human brains work (Daliakopoulos *et al.*, 2005), which consists of a set of nodes that imitate the neuron and carries activation signals of different strength. ANN is a black box as it does not provide any insights on the structure of the function being approximated. There are two approaches to ANN development, which are supervised and unsupervised learning algorithm. The KNN approach comprises of input samples of k -nearest neighbors in the training set in the feature space and assigns the labels of classes that are most similar among its k -nearest neighbors. Performance of KNN is largely reliant on the value of k and the applied distance metric such as Euclidean distances or Manhattan distance. The CART algorithm uses tree structure where the roots are at the top with the leaves at the bottom (Kuhn and Johnson, 2013). A node represents a condition of an attribute, each branch signifies the result of a condition, and each leaf node holds a class label. CART involves three phases: Creation of an initial tree from examples, pruning the tree to remove branches with little statistical validity, and processing the pruned tree to improve its understandability (Alpaydin, 2004; Kotsiantis, 2007). Naive Bayes is a linear classification model that uses Bayes theorem to analyze feature probabilities and assumes feature independence. This approach is particularly useful when the dataset is large and contains many different features.

There is no one particular classification method that is superior and outperforms other available classification methods. Every method has its own advantages and disadvantages. SVM using radial basis function (RBF) classifier is superior in performance and is tolerant to irrelevant and highly correlated features as compared to decision trees, ANN, and KNN classifiers (Alpaydin, 2004; Kotsiantis, 2007).

SVM is suitable when data is not linearly separable. However the SVM algorithm is slower due to the optimization of the hyperplane parameters, the cost parameter C and γ . The selection of these parameters' value involves a grid search. The cost parameter provides an adjustment between training error and model complexity. Higher value of C indicates higher cost for nonseparable samples (Alpaydin, 2004; Kotsiantis, 2007). The SVM model depends on mapping data to a higher dimensional space by using the function of a kernel, the maximum-margin hyperplane will be selected in order to separate training data. Hence SVM improves the accuracy via optimization of the space separation. Thus, it produces a better result compared to the other classification models. ANN and SVM models are considered good at fitting functions and recognizing patterns in various datasets. Both ANN and SVM can approximate practically all types of nonlinear functions (Desai *et al.*, 2008; Gulati *et al.*, 2010). However, there are some limitations such as that standardized coefficients for variables may not be straightforwardly calculated and presented, and it is considered as a "black box" approach. The complete insight into the internal workings of the model or information for evaluating the interaction of inputs is unknown (Dayhoff and DeLeo, 2001). RF compared to ANN and SVM is relatively easier to use and RF provides insight to the variable importance.

Nuclear abnormality is a hallmark of progeria in humans. Analysis of age-dependent nuclear morphological changes in *C. elegans* is of great value to aging research. Classification of *C. elegans* images was made using several machine learning methods such as linear SVM, RF, KNN, ANN, and DT. The data used in this study is imbalanced where the data inhibits unequal amount of data for each class. In order to improve classification performance when imbalanced data set is involved each class is assigned weight based on number of total sample over number of samples in each class. Optimum number of classifiers are identified using the K-fold method and the RF algorithm outperformed all other machine learning algorithms including SVM. The reason behind this is the choice of SVM algorithm in this study, that is, linear SVM. Performance classification of SVM-linear is faster compared to SVM-RBF. However SVM-RBF demonstrates higher classification accuracy and lower misclassification compared to SVM-linear. SVM-linear is a suitable choice for variable selection during iterative training phase and SVM-RBF for all cell types' classification (Abbas *et al.*, 2014; Mao and Tao, 2017). SVM has been used as a classifier to assess the existence of malaria cell *P. falciparum* trophozoites and white blood cells in Giemsa stained thick blood smears. Images were obtained using a smartphone and manually annotated. The results were 80.5% sensitivity for malaria cell and 98.2% sensitivity for white blood cell (Rosado *et al.*, 2016).

Automated plant identification using machine learning methods can be applied to classify plants into appropriate taxonomies. Brilhador *et al.* (2013) compared KNN, ANN, SVM, RF, and naïve Bayes for plant leaf image classification using all features and selected features. SVM outperformed all other classifiers without and with using feature selection method. RF and ANN performance was slightly lower than SVM. Kho *et al.* (2017) applied SVM and ANN classifier for automated ficus leaf identification that resulted in both models having similar performance. K-means method was used for diseased plant leaf image clustering and ANN for classification resulted in accuracy of 94% (Al-Hiary *et al.*, 2011).

Plant leaf identification of 32 different species using convex hull, morphological, color histogram, and distance map features using KNN as classifier achieved an accuracy of 87.3% (Munisami *et al.*, 2015). Automated plant disease using Otsu segmentation method was carried out using four different types of classifiers, that is KNN with k value of 1, ANN, naïve Bayesian, linear SVM, and decision tree. SVM classifier produced the highest accuracy of 94.45% using discrete cosine and wavelet transform and 90% accuracy using texture features alone. Previous studies in plant disease identification (Camargo and Smith, 2009; Al-Hiary *et al.*, 2011) using SVM with texture, shape, histogram of frequency features using thresholding segmentation method, and ANN with K -means clustering using texture features produced a slightly lower classification result of 94% (Akhtar *et al.*, 2013).

In a species identification system successful results have been obtained using ANN and KNN as a classifier. A species identification system using photographic images of fish, plant, and butterfly species uses geometry, morphology, and texture feature and ANN classifier for image recognition. ANN achieved accuracy of about 90% for all species (Hernández-Serna and Jiménez-Segura, 2014). DAISY is a generic species identification system for insect groups using ANN, SVM, and a plastic self-organizing map as classifiers (O'Neill, 2007). ANN and shape features were used to classify the skulls, sex, and region of species of *Suncus murinus*. Skull classification achieved 100% and the other was about 80%. This indicates that shape alone is sufficient for species identification (Abu *et al.*, 2016). KNN was applied to classify the monogenean specimens and achieved classification accuracy of 90% using morphological shape features (Kalafi *et al.*, 2016).

In the medical domain the SVM classifier achieved higher accuracy as well compared to other methods. Comparative study on automated identification of ophthalmic images using local binary pattern and SVM as classifier and wavelet transformation and SVM attained accuracy of 87%. Combination of color and texture feature using SVMs or k NN classifier identification of ophthalmic images can be used for mobile devices application due to ease of implementation and faster processing time (Wang *et al.*, 2017). Early detection of skin cancer melanoma using morphological features of pixel intensity extracted from diseased layer of skin was classified using SVM classifier, which achieved 92% (Li *et al.*, 2013). Identification of tumor location from head and neck MRI images data is vital but very complicated. SVM classifier is used with discrete sine transform and DCT achieve better performance compared to the existing techniques (Gouid *et al.*, 2014). Cervical cell nuclei classification using morphometric and Haralick texture features with PCA for dimension reduction was conducted using linear method, KNN, Mahalanobis, and SVM. SVM classifier with PCA achieved highest classification accuracy (Lorenzo-Ginori *et al.*, 2013).

Application of feature selection and different combination features improves classifier performance in bioimage classification. Feature selection algorithm effect on the performance accuracy of plant leaf classification was studied using KNN, CART, RF, and J48. Gabor texture features and RF feature selection method was used to select significant features in plant leaf identification. RF model outperformed with and without feature selection; meanwhile CART was the worst performing classification model. Classification accuracy increased using selected features compared to using a full set of features for all classification algorithm (Kumar *et al.*, 2015). Adaboost classifiers boosting algorithm for binary classification was used with a combination of shape features using SIFT, color feature using color moment, and texture feature using SFTA in herbal plant identification. Single texture feature and combination of texture and shape feature produced almost similar classification results of 92% compared to using all three features together of 94% or using either shape or color feature alone (Jamil *et al.*, 2015). SVM with RBF kernel classifier using feature reduction method for yeast image classification for 20 classes achieved 87% and classification without feature selection is 95% (Huh *et al.*, 2009).

Deep learning applications have been gaining popularity in bioimage informatics. The deep learning method uses a convolutional neural network (CNN), a special kind of ANN. Individual neurons execute a convolution of a kernel with an input image and generate a filtered output image or a feature map. The input image can be made out of a few channels, and each layer in the neural net holds as many channels of feature maps. The feature maps in the last layer are considered the final features learned by the network and are used for classification. The main difference of CNN from conventional feature based classification methods are features, and the weights of the kernels, are learnt by the CNN algorithm. CNN performance was compared with conventional machine learning classifiers such as SVM, RF, and LDA. CNN outperformed with high accuracy rate of 97%. CNN as a features learning algorithm is as good as conventional features and supervised learning algorithm (Dürr and Sick, 2016). Related literature on CNN application include classification of red blood cells in sickle cell anemia (Xu *et al.*, 2017), breast cancer histopathological image classification (Spanhol *et al.*, 2016), and automatic plant type identification. CNN has been compared with SVM and yielded higher accuracy (Yalcin and Razavi, 2016).

Bioimage software for image analysis and classification uses various classification methods and is easy to use. Area, shape, intensity, texture, and advanced features such as Haralick, Gabor, and Zernike features are the most commonly used features for cell identification in different bioimage software packages (Abbas *et al.*, 2014). CP-CHARM uses LDA as classification and PCA for feature dimension reduction (Uhlmann *et al.*, 2016). WND-CHARM uses thresholding and weighting as feature selection algorithm and weighted nearest neighbor for classification (Orlov *et al.*, 2008). CellProfiler is a Matlab-based package for analysis of biological images using joint boosting classifier limited to two classes of classification (Lamprecht *et al.*, 2007). To enhance CellProfiler capabilities, enhanced CellClassifier (Misselwitz *et al.*, 2010) uses an images analyst by CellProfiler for multiclass classification using SVM. BIOCAT software is used for automatic classification and annotation of 2D and 3D and regions of interest on individual biological images. It is mainly used for cell biology and neuroscience. Feature selection is carried out using Fisher's criterion and supports classifiers such as SVM, KNN, naïve Bayes, DT, and RF (Zhou *et al.*, 2013). FARSIGHT is software comprised of modules of bioimage analysis and classification that uses supervised spectral clustering. CellCognition (Held *et al.*, 2010) uses a hidden Markov model algorithm as classifier. CellXpress is based on SVM and Ilastik (Sommer *et al.*, 2011) uses RF as its classifier. SVM linear and SVM RBF have proven to be superior in classification task compared to joint boosting and LDA classifier. Abbas *et al.* (2014) suggests that the SVM based classifier should be incorporated in bioimage analysis tools' fast and efficient analysis of high-throughput data.

Image Registration and Annotation

Image registration is defined as a process that overlays two or more images from various imaging equipment or sensors taken at different times and angles, or from the same scene to geometrically align the images for analysis (Zitová and Flusser, 2003). Objects taken in different imaging equipment are not alike. Existence of calcification can be identified efficiently by using computed tomography images (CT scan). In MRI uneven intravascular calcification plaque and blood disorders can create image artifacts. Application of using similar image scanning method for example MRI on different scanning equipment with different parameter can also lead to differences in acquired image. The image registration method can be applied to resolve the issues from differences in images acquired due to equipment and parameter settings (Chen *et al.*, 2017).

The basic procedure in the image registration method comprises feature detection, feature alignment from the captured image and the reference image, mapping functions parameter estimation, and image resampling and transformation. Image registration methods are unique to each problem and there is no common technique of image registration that is applicable to all registration tasks due to images' geometric radiometric distortions and noise disturbance, data characteristic, and accuracy threshold level.

Vascular registration of retinal and MRA methods proposed by Chen *et al.* (2017) comprises area and feature based registration methods. The area based method comprises of cross correlation matching and mutual information methods and correspondences estimation from the original image. The feature based method consists of a description of invariance obtained from feature extraction such as correlation coefficient, closed region, geometric features, and coherent point drift. This method proposed by the author can be applied to other tubular structures.

There is a huge amount of images of proteins with cancerous tissues, however due to absence of clear and precise annotations, images from cancerous tissues are not being utilized for developing supervised models for cancer classification (Uhlen *et al.*, 2010). Annotation of biological images has conventionally been done manually by a domain expert and it is very time consuming. Automated annotation *Drosophila* gene expression pattern images and identification of the images have been carried out using the bags-of-words (BoW) method. Images are denoted as BoW based on visual features extracted from the image to produce a visual codebook about the image. Feature of an image based on the regions of an image is extracted using SIFT descriptor for image codebook generation. The spatial BoW method on the FlyExpress image database is used to generate a histogram for each image and frequency of words appearing in an image. The BoW trails the location of visual words appearing on the image. The spatial BoW is an enhancement of the BoW method with spatial properties of an image with better accuracy compared to using nonspatial BoW or nonannotated images (Yuan *et al.*, 2012). Software such as the BIOCAT is used for automated bioimage annotation. The software is able to perform automated image recognition, classification and annotation of images and region of interests with application to cell biology and neuroscience (Zhou *et al.*, 2013).

Closing Remarks

This work discusses general aspects of bioimage informatics covering cell, organism, species, and medical images. Techniques discussed in this work can be applied to various biological images.

See also: Data Mining: Classification and Prediction. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics. Proteome Informatics

References

- Abbas, M.M., Mohie-Eldin, M.M., El-Manzalawy, Y., 2015. Assessing the Effects of Data Selection and Representation on the Development of Reliable E. coli Sigma 70 Promoter Region Predictors. *Plos One* 10 (3), doi:10.1371/journal.pone.0119721.
- Abbas, S., Dijkstra, T.M., Heskes, T., 2014. A comparative study of cell classifiers for image-based high-throughput screening. *BMC Bioinformatics* 15, 342. doi:10.1186/1471-2105-15-342.
- Abdullah, H.M., 2013. A preliminary study on an automated freshwater algae recognition and classification system.
- Abu, A., Leow, L.K., Ramli, R., Omar, H., 2016. Classification of *Suncus murinus* species complex (Soricidae: Crocidurinae) in peninsular Malaysia using image analysis and machine learning approaches. *BMC Bioinformatics* 17 (Suppl. 19), S505. doi:10.1186/s12859-016-1362-5.
- Akhtar, A., Khanum, A., Khan, S.A., Shaikat, A., (2013). Automated Plant Disease Analysis (APDA): Performance comparison of machine learning techniques. In: Proceedings of the 2013 11th International Conference on Frontiers of Information Technology. doi:10.1109/fit.2013.19.
- Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., ALRahamneh, Z., 2011. Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications* 17, 31–38.
- Allen, M.J., Kanteti, R., Riehm, J.J., El-Hashani, E., Salgia, R., 2015. Whole-animal mounts of *Caenorhabditis elegans* for 3D imaging using atomic force microscopy. *Nanomedicine: Nanotechnology, Biology and Medicine* 11, 1971–1974. doi:10.1016/j.nano.2015.07.014.
- Alpaydin, E., 2004. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN: 026201243.
- Barry, D., Williams, G., 2011. Microscopic characterisation of filamentous microbes: Towards fully automated morphological quantification through image analysis. *Journal of Microscopy* 244 (1), 1–20. doi:10.1111/j.1365-2818.2011.03506.x.
- Basavaprasad, B., Ravi, M., 2014. A comparative study on classification of image segmentation methods with a focus on graph based techniques. *International Journal of Research in Engineering and Technology* 03, 310–315. doi:10.15623/ijret.2014.0315060.

- Bay, H., *et al.*, 2008. Speeded-up robust features (SURF). *Comput. Vis. Image Understand.* 110, 346–359.
- Bevi, K.S., Nair, M.S., Bindu, G.R., 2014. Automatic segmentation and classification of mitotic cell nuclei in histopathology images based on Active Contour Model. In: *Proceedings of the 2014 International Conference on Contemporary Computing and Informatics (IC3I)*. doi:10.1109/ic3i.2014.7019762.
- Betzig, E., Patterson, G.H., Sougrat, R., *et al.*, 2006. Imaging intracellular fluorescent proteins at nanometer resolution. *Science* 313, 1642–1645. doi:10.1126/science.1127344.
- Bhuiyan, M.A.M., Azad, I., Uddin, M.U., 2013. Image processing for skin cancer features extraction. *International Journal of Scientific & Engineering Research* 4 (2).
- Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. *Classification and Regression Trees*. Taylor & Francis.
- Breiman, L., 2001. Random forests. *Machine Learning* 45, 5–32.
- Brilhador, A., Colonhezi, T.P., Bugatti, P.H., Lopes, F.M., 2013. Combining texture and shape descriptors for bioimages classification: A case of study in ImageCLEF dataset. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science*. 431–438. doi:10.1007/978-3-642-41822-8_54.
- Camargo, A., Smith, J.S., 2009. Image pattern classification for the identification of disease causing agents in plants. *Computers and Electronics in Agriculture* 66, 121–125.
- Canny, J., 1987. A Computational Approach to Edge Detection. *Readings in Computer Vision*. 184–203. doi:10.1016/b978-0-08-051581-6.50024-6.
- Chalazonitis, A.N., Koumarios, D., Tzovara, J., Chronopoulos, P., 2003. How to optimize radiological images captured from digital cameras, using the Adobe Photoshop 6.0 program. *Journal of Digital Imaging* 16, 216–229. doi:10.1007/s10278-003-1651-1.
- Chen, L., Lian, Y., Guo, Y., *et al.*, 2017. A vascular image registration method based on network structure and circuit simulation. *BMC Bioinformatics* 18 (1), doi:10.1186/s12859-017-1649-1.
- Chopin, J., Laga, H., Miklavcic, S.J., 2016. A hybrid approach for improving image segmentation: Application to phenotyping of wheat leaves. *PLOS ONE* 11, e0168496. doi:10.1371/journal.pone.0168496.
- Chora's, R.S., 2007. Image feature extraction techniques and their applications for CBIR and biometrics systems. *International Journal of Biology and Biomedical Engineering* 1 (1), 6–16.
- Coelho, L.P., Pato, C., Friães, A., *et al.*, 2015. Automatic determination of NET (neutrophil extracellular traps) coverage in fluorescent microscopy images. *Bioinformatics* 31, 2364–2370. doi:10.1093/bioinformatics/btv156.
- Coltelli, P., Barsanti, L., Evangelista, V., Frassanito, A.M., Gualtieri, P., 2014. Water monitoring: Automated and real time identification and classification of algae using digital microscopy. *Environ. Sci.: Processes Impacts* 16 (11), 2656–2665. doi:10.1039/c4em00451e.
- Daliakopoulos, I.N., Coulibaly, P., Tsanis, I.K., 2005. Groundwater level forecasting using artificial neural networks. *Journal of Hydrology* 309, 229–240.
- Davies, E.R., 1990. *Machine Vision: Theory, Algorithms and Practicalities*. London: Academic Press, pp. 42–44.
- Dayhoff, J.E., DeLeo, J.M., 2001. Artificial neural networks. *Cancer* 91, 1615–1635.
- Desai, K.M., Survase, S.A., Saudagar, P.S., Lele, S., Singhal, R.S., 2008. Comparison of artificial neural network (ANN) and response surface methodology (RSM) in fermentation media optimization: Case study of fermentative production of scleroglucan. *Biochemical Engineering Journal* 41 (3), 266–273. doi:10.1016/j.bej.2008.05.009.
- Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S., 2012. Hierarchical classification of diatom images using ensembles of predictive clustering trees. *Ecological Informatics* 7 (1), 19–29. doi:10.1016/j.ecoinf.2011.09.001.
- Dürr, O., Sick, B., 2016. Single-cell phenotype classification using deep convolutional neural networks. *Journal of Biomolecular Screening* 21, 998–1003. doi:10.1177/1087057116631284.
- Gelasca, E.D., Obara, B., Fedorov, D., Kvilekval, K., Manjunath, B., 2009. A biosegmentation benchmark for evaluation of bioimage analysis methods. *BMC Bioinformatics* 10, 368. doi:10.1186/1471-2105-10-368.
- George, G.V.S., Raj, V.C., 2011. Review on feature selection techniques and the impact of SVM for cancer classification using gene expression profile. *CoRR*. abs/1109.1062.
- Gomez, A., Diez, G., Salazar, A., Diaz, A., 2016. Animal Identification in Low Quality Camera-Trap Images Using Very Deep Convolutional Neural Networks and Confidence Thresholds. In: *Bebis, G., et al. (Eds.), Advances in Visual Computing. ISVC 2016. Lecture Notes in Computer Science*, vol 10072. Cham: Springer.
- Gonzalez, R., Woods, R., 2007. *Digital Image Processing*, third ed. Prentice Hall.
- Gorbunova, V., Sparring, J., Lo, P., *et al.*, 2012. Mass preserving image registration for lung CT. *Medical Image Analysis* 16, 786–795. doi:10.1016/j.media.2011.11.001.
- Gould, G., Nasser, A., Mostafa, M., El-Hennawi, D., 2014. Automatic identification of Head and Neck Swellings in MRI images using support vector machines based on cepstral analysis. In: *Proceedings of the 2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*. doi:10.1109/cibcb.2014.6845504.
- Gulati, T., Chakrabarti, M., Singh, A., Duvuuri, M., Banerjee, R., 2010. Comparative study of response surface methodology, artificial neural network and genetic algorithms for optimization of soybean hydration. *Food Technol Biotechnol* 48 (1), 11–18.
- Guo, J., Mei, X., Tang, K., 2013. Automatic landmark annotation and dense correspondence registration for 3D human facial images. *BMC Bioinformatics* 14, 232. doi:10.1186/1471-2105-14-232.
- Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182.
- Hall, M.A., 1999. Correlation-based feature selection for machine learning.
- Handels, H., Werner, R., Schmidt, R., *et al.*, 2007. 4D medical image computing and visualization of lung tumor mobility in spatio-temporal CT image data. *International Journal of Medical Informatics* 76 (Suppl. 3), S433–S439. doi:10.1016/j.ijmedinf.2007.05.003.
- Haralick, R.M., Shapiro, L.G., 1992. *Computer and robot vision*. Reading, MA: Addison-Wesley.
- Haralick, R.M., Shanmugam, K., *et al.*, 1973. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* 3, 610–621.
- Held, M., Schmitz, M.H., Fischer, B., *et al.*, 2010. CellCognition: Time-resolved phenotype annotation in high-throughput live cell imaging. *Nature Methods* 7, 747–754. doi:10.1038/nmeth.1486.
- Hernández-Serna, A., Jiménez-Segura, L.F., 2014. Automatic identification of species with neural networks. *PeerJ* 2, e563. doi:10.7717/peerj.563.
- Hess, S.T., Girirajan, T.P.K., Mason, M.D., 2006. Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophysical Journal* 91, 4258–4272. doi:10.1529/biophysj.106.091116.
- Huh, S., Lee, D., Murphy, R.F., 2009. Efficient framework for automated classification of subcellular patterns in budding yeast. *Cytometry Part A* 75A, 934–940. doi:10.1002/cyto.a.20793.
- Intarapanich, A., Kaewkamnerd, S., Pannarut, M., Shaw, P.J., Tongsim, S., 2016. Fast processing of microscopic images using object based extended depth of field. *BMC Bioinformatics* 17 (Suppl. 19), 516. doi:10.1186/s12859-016-1373-2.
- Islam, M.M., D. Zhang, G. Lu, 2008. A geometric method to compute directionality features for texture images. In: *Presented at the IEEE International Conference of Multimedia and Expo (ICME)*, IEEE. doi:10.1109/ICME.2008.4607736.
- Isnanto, R.R., Zahra, A.A., Julietta, P., 2016. Pattern recognition on herbs leaves using region-based invariants feature extraction. In: *Proceedings of the 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. doi:10.1109/icitacee.2016.7892491.
- Jamil, N., Hussin, N.A., Nordin, S., Awang, K., 2015. Automatic plant identification: Is shape the key feature? *Procedia Computer Science* 76, 436–442. doi:10.1016/j.procs.2015.12.287.
- Jayaram, M.A., Fleyeh, H., 2016. Convex hulls in image processing: A scoping review. *American Journal of Intelligent Systems* 6, 48–58. doi:10.5923/j.ajis.20160602.03.
- Jia, D., Xu, Q., Xie, Q., Mio, W., Deng, W., 2016. Automatic stage identification of Drosophila egg chamber based on DAPI images. *Scientific Reports* 6. doi:10.1038/srep18850.
- Kalafi, E.Y., Boon, T.W., Town, C., Dhillon, S.K., 2016. Automated identification of Monogeneans using digital image processing and k-nearest neighbour approaches. *BMC Bioinformatics* 17 (Suppl. 19), S511. doi:10.1186/s12859-016-1376-z.
- Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. *International Journal of Computer Vision* 1 (4), 321–331. doi:10.1007/BF00133570.

- Keller, P.J., Stelzer, E.H.K., 2010. Digital scanned laser light sheet fluorescence microscopy. *Cold Spring Harb Protocots* 2010 (5), [pdb.top78](#).
- Kelley, L.C., Wang, Z., Hagedorn, E.J., *et al.*, 2017. Live-cell confocal microscopy and quantitative 4D image analysis of anchor-cell invasion through the basement membrane in *Caenorhabditis elegans*. *Nature Protocols* 12, 2081–2096. [doi:10.1038/nprot.2017.093](#).
- Kho, S.J., Manickam, S., Malek, S., Mosleh, M.A., Dhillon, S.K., 2017. Automated plant identification using artificial neural network and support vector machine. *Frontiers in Life Science* 10 (1), 98–107. [doi:10.1080/21553769.2017.1412361](#). [org](#).
- Kiranyaz, S., Ince, T., Pulkkinen, J., *et al.*, 2011. Classification and retrieval on macroinvertebrate image databases. *Computers in Biology and Medicine* 41, 463–472.
- Kotsiantis, S.B., 2007. Supervised machine learning: A review of classification techniques. *Informatica* 31 (3), 249–268.
- Kuhn, M., Johnson, K., 2013. *Applied Predictive Modeling*. vol. 26. Springer.
- Kumar, A., Patidar, V., Khazanchi, D., Saini, P., 2015. Role of feature selection on leaf image classification. *Journal of Data Analysis and Information Processing* 03, 175–183. [doi:10.4236/jdaip.2015.34018](#).
- Kvilekval, K., *et al.*, 2010. Bisque: A platform for bioimage analysis and management. *Bioinformatics* 26, 544–552.
- Lamprecht, M., Sabatini, D., Carpenter, A., 2007. CellProfiler™: Free, versatile software for automated biological image analysis. *BioTechniques* 42, 71–75. [doi:10.2144/000112257](#).
- Lewis, J.M., Van Der Maaten, L., De Sa, V.R., 2012. A behavioral investigation of dimensionality reduction. In: *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, pp. 671–676.
- Lim, J.S., 1990. *Two-dimensional signal and image processing*. Englewood Cliffs, NJ: Prentice Hall, p. 710.
- Liu, R., Lin, S., Rallo, R., *et al.*, 2012. Automated phenotype recognition for zebrafish embryo based in vivo high throughput toxicity screening of engineered nano-materials. *PLOS ONE* 7 (4), e35014. [doi:10.1371/journal.pone.0035014](#).
- Li, L., Zhang, Q., Ding, Y., *et al.*, 2013. A computer-aided spectroscopic system for early diagnosis of melanoma. In: *Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence*. [doi:10.1109/ictai.2013.31](#).
- Long, F., Peng, H., Myers, E., 2007. Automatic Segmentation of Nuclei In 3D Microscopy Images of *C.elegans*. 2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro. [doi:10.1109/isbi.2007.356907](#).
- Lorenzo-Ginori, J.V., Curbelo-Jardines, W., López-Cabrera, J.D., Huergo-Suárez, S.B., 2013. Cervical cell classification using features related to morphometry and texture of nuclei. In: Ruiz-Schulcloper, J., Sanniti di Baja, G. (Eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2013. Lecture Notes in Computer Science*, vol 8259. Berlin, Heidelberg: Springer.
- Luo, Q., Gao, Y., Luo, J., *et al.*, 2011. Automatic Identification of Diatoms with Circular Shape using Texture Analysis. *Journal of Software* 6 (3), [doi:10.4304/jsw.6.3.428-435](#).
- Mao, H., Tao, L., 2017. Segmentation and classification of two-channel *C. elegans* nucleus-labeled fluorescence images. *BMC Bioinformatic* 18, 412. [doi:10.1186/s12859-017-1817-3](#).
- Mayer, J., Swoger, J., Ozga, A.J., Stein, J.V., Sharpe, J., 2012. Quantitative measurements in 3-dimensional datasets of mouse lymph nodes resolve organ-wide functional dependencies. *Computational and Mathematical Methods in Medicine* 2012. Article ID 128431, 8 pages.
- Myers, G., 2012. Why bioimage informatics matters. *Nature Methods* 9 (7), 659–660. [doi:10.1038/nmeth.2024](#).
- Mingqiang, Y., Kidiyo, K., Joseph, R., 2008. A survey of shape feature extraction techniques. *Pattern Recognition Techniques, Technology and Applications*. [doi:10.5772/6237](#).
- Misselwitz, B., Strittmatter, G., Periaswamy, B., *et al.*, 2010. Enhanced CellClassifier: A multi-class classification tool for microscopy images. *BMC Bioinformatics* 11, 30. [doi:10.1186/1471-2105-11-30](#).
- Mosleh, M.A.A., Baba, M.S., Sorayya, M., Almakari, R.A., 2016. Ceph-X: Development and evaluation of 2D cephalometric system. *BMC Bioinformatics* 17 (Suppl. 19), S499. [doi:10.1186/s12859-016-1370-5](#).
- Mosleh, M.A., Manssor, H., Malek, S., Milow, P., Salleh, A., 2012. A preliminary study on automated freshwater algae recognition and classification system. *BMC Bioinformatics* 13, S25. [doi:10.1186/1471-2105-13-25](#).
- Munisami, T., Ramsurn, M., Kishnah, S., Pudaruth, S., 2015. Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers. *Procedia Computer Science* 58, 740–747. [doi:10.1016/j.procs.2015.08.095](#).
- Murphy, R.F., 2014. A new era in bioimage informatics. *Bioinformatics* 30, 1353. [doi:10.1093/bioinformatics/btu158](#).
- Murat, M., Chang, S., Abu, A., Yap, H.J., Yong, K., 2017. Automated classification of tropical shrub species: A hybrid of leaf shape and machine learning approach. *PeerJ* 5. [doi:10.7717/peerj.3792](#).
- Mythili, C., Kaviitha, V., 2011. Efficient Technique for Color Image Noise Reduction. *The research bulletin of Jordan. ACM*.
- Nguyen B.P., Heemskerck H., So P.T.C., Tucker-Kellogg, L., 2016. Superpixel-based segmentation of muscle fibers in multi-channel microscopy. *BMC Systems Biol*. [doi:10.1186/s12918-016-0372-2](#). In: *Proceedings of the 16th International Conference on Bioinformatics (InCoB 2017)*: Available at: <http://incob.apbionet.org/incob17> (accessed 18.11.16).
- Niwas, S.I., Palanisamy, P., Sujathan, K., Bengtsson, E., 2013. Analysis of nuclei textures of fine needle aspirated cytology images for breast cancer diagnosis using Complex Daubechies wavelets. *Signal Processing* 93, 2828–2837. [doi:10.1016/j.sigpro.2012.06.029](#).
- Obara, B., Grau, V., Fricker, M.D., 2012. A bioimage informatics approach to automatically extract complex fungal networks. *Bioinformatics* 28, 2374–2381. [doi:10.1093/bioinformatics/bts364](#).
- Oneill, M., 2007. Daisy. *Systematics Association Special Volumes Automated Taxon Identification in Systematics*. 101–114. [doi:10.1201/9781420008074.ch7](#).
- Orlov, N., Shamir, L., Macura, T., *et al.*, 2008. WND-CHARM: Multi-purpose image classification using compound image transforms. *Pattern Recognition Letters* 29, 1684–1693. [doi:10.1016/j.patrec.2008.04.013](#).
- Pawley, J. (Ed.), 2006. *Handbook Of Biological Confocal Microscopy*. [doi:10.1007/978-0-387-45524-2](#).
- Peng, H., 2008. Bioimage informatics: A new area of engineering biology. *Bioinformatics* 24, 1827–1836.
- Permuter, H., Francos, J., Jermyn, I., 2006. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition* 39, 695–706.
- Perre, P., Faria, F.A., Jorge, L.R., *et al.*, 2016. Toward an automated identification of anastrepha fruit flies in the fraterculus group (Diptera, Tephritidae). *Neotropical Entomology* 45, 554–558. [doi:10.1007/s13744-016-0403-0](#).
- Pin Tian, D., 2013. A Review on Image Feature Extraction and Representation Techniques. *International Journal of Multimedia and Ubiquitous Engineering* 8 (4),
- Puniyani, K., Xing, E.P., 2013. GINI: From ISH Images to Gene Interaction Networks. *PLoS Computational Biology* 9 (10), [doi:10.1371/journal.pcbi.1003227](#).
- Rosado, L., Correia Da Costa, J.M., Elias, D., Cardoso, J.S., 2016. A review of automatic malaria parasites detection and segmentation in microscopic images. *Anti-Infective Agents* 14, 11–22. [doi:10.2174/221135251401160302121107](#).
- Rother, C., Kolmogorov, V., Blake, A., 2004. GrabCut. *ACM SIGGRAPH 2004 Papers on - SIGGRAPH 04*. [doi:10.1145/1186562.1015720](#).
- Rust, M.J., Bates, M., Zhuang, X., 2006. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature Methods* 3, 793–796. [doi:10.1038/nmeth929](#).
- Sabeena Beevi, K., Madhu, S. Nair, Bindu, G.R., 2017. A Multi-Classifer System for Automatic Mitosis Detection in Breast Histopathology Images using Deep Belief Networks. *IEEE Journal of Translational Engineering in Health and Medicine* 5 (1), 1–11.
- Sadanandan, S.K., Ranefall, P., Guyader, S.L., Wahlby, C., 2017. Automated training of deep convolutional neural networks for cell segmentation. *Scientific Reports* 7 (1), 7860. [doi:10.1038/s41598-017-07599-6](#).
- Saeyns, Y., Inza, I., Larranaga, P., 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* 23, 2507–2517. [doi:10.1093/bioinformatics/btm344](#).

- Santhi, N., Pradeepa, C., Subashini, P., Kalaiselvi, S., 2013. Automatic Identification of Algal Community from Microscopic Images. *Bioinformatics and Biology Insights* 7. doi:10.4137/bbi.s12844.
- Sánchez, M.G., Vidal, V., Verdú, G., Mayo, P., Rodenas, F., 2012. Medical image restoration with different types of noise. In: *Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4382–4385.
- Savkare, S., Narote, S., 2012. Automatic System for Classification of Erythrocytes Infected with Malaria and Identification of Parasites Life Stage. *Procedia Technology* 6, 405–410. doi:10.1016/j.protcy.2012.10.048.
- Schmidt, T., Dürr, J., Keuper, M., *et al.*, 2013. Variational attenuation correction in two-view confocal microscopy. *BMC Bioinformatics* 14, 366. doi:10.1186/1471-2105-14-366.
- Schmid, B., Schindelin, J., Cardona, A., Longair, M., Heisenberg, M., 2010. A high-level 3D visualization API for Java and ImageJ. *BMC Bioinformatics* 11, 274.
- Seyyid, A.M., 2015. A comparative study of feature extraction methods in images classification. *International Journal of Image, Graphics and Signal Processing* 3, 16–23. doi:10.5815/ijigsp.2015.03.03.
- Sommer, C., Straehle, C., Kothe, U., Hamprecht, F.A. 2011. Ilastik: Interactive learning and segmentation toolkit. In: *Proceedings of the 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. doi:10.1109/isbi.2011.5872394.
- Song, Y., Cai, W., Huang, H., *et al.*, 2016. Bioimage classification with subcategory discriminant transform of high dimensional visual descriptors. *BMC Bioinformatics* 17 (1), 465. doi:10.1186/s12859-016-1318-9.
- Song, Y., Cai, W., Huang, H., *et al.*, 2013. Region-based progressive localization of cell nuclei in microscopic images with data adaptive modeling. *BMC Bioinformatics* 14, 173. doi:10.1186/1471-2105-14-173.
- Song, Y., Ji, Z., Sun, Q., 2014. An extension Gaussian mixture model for brain MRI segmentation. In: *Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. doi:10.1109/embc.2014.6944676.
- Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L., 2016. Breast cancer histopathological image classification using Convolutional Neural Networks. In: *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*. doi:10.1109/ijcnn.2016.7727519.
- Swedlow, J.R., 2012. Innovation in biological microscopy: Current status and future directions. *Bioessays* 34, 333–340.
- Uhlen, M., *et al.*, 2010. Towards a knowledge-based human protein atlas. *Nature Biotechnology* 28, 1248–1250.
- Uhlmann, V., Singh, S., Carpenter, A.E., 2016. CP-CHARM: Segmentation-free image classification made accessible. *BMC Bioinformatics* 17, 51. doi:10.1186/s12859-016-0895-y.
- Vapnik, V.N., Vapnik, V., 1998. *Statistical Learning Theory*. vol. 1. New York: Wiley.
- Verikas, A., Gelzinis, A., Bacauskiene, M., *et al.*, 2012. Automated image analysis- and soft computing-based detection of the invasive dinoflagellate *Prorocentrum minimum* (Pavillard) Schiller. *Expert Systems with Applications* 39 (5), 6069–6077. doi:10.1016/j.eswa.2011.12.006.
- Villa, A.G., Salazar, A., Vargas, F., 2017. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. *Ecological Informatics* 41, 24–32. doi:10.1016/j.ecoinf.2017.07.004.
- Wang, L., Zhang, K., Liu, X., *et al.*, 2017. Comparative analysis of image classification methods for automatic diagnosis of ophthalmic images. *Scientific Reports* 7, 41545. doi:10.1038/srep41545.
- Xu, M., Papageorgiou, D.P., Abidi, S.Z., *et al.*, 2017. A deep convolutional neural network for classification of red blood cells in sickle cell anemia. *PLOS Computational Biology* 13 (10), doi:10.1371/journal.pcbi.1005746.
- Xu, Y., Yang, F., Zhang, Y., Shen, H., 2014. Bioimaging-based detection of mislocalized proteins in human cancers by semi-supervised learning. *Bioinformatics* 31, 1111–1119. doi:10.1093/bioinformatics/btu772.
- Xinding, S., Zhenming, X., Changsheng, X., Yan, W. (n.d.). Adaptive Kalman filtering approach of color noise in cephalometric image. *Proceedings of Third International Conference on Signal Processing (ICSP96)*. doi:10.1109/icsigp.1996.567341.
- Yang, H., Ahuja, N., 2014. Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed. *Pattern Recognition* 47 (6), 2266–2279. doi:10.1016/j.patcog.2013.11.004.
- Yalcin, H., Razavi, S., 2016. Plant classification using convolutional neural networks. In: *Proceedings of the 2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*. doi:10.1109/agro-geoinformatics.2016.7577698.
- Yuan, L., Woodard, A., Ji, S., *et al.*, 2012. Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval. *BMC Bioinformatics* 13, 107. doi:10.1186/1471-2105-13-107.
- Yue, Y., Croitoru, M., Bidani, A., Zwischenberger, J., Clark, J., 2006. Nonlinear multiscale wavelet diffusion for speckle suppression and edge enhancement in ultrasound images. *IEEE Transactions on Medical Imaging* 25 (3), 297–311. doi:10.1109/tmi.2005.862737.
- Yusof, R., Khalid, M., Khairuddin, A.S., 2013. Application of kernel-genetic algorithm as nonlinear feature selection in tropical wood species recognition system. *Computers and Electronics in Agriculture* 93, 68–77. doi:10.1016/j.compag.2013.01.007.
- Zhao, M., An, J., Li, H., *et al.*, 2017. Segmentation and classification of two-channel *C. elegans* nucleus-labeled fluorescence images. *BMC Bioinformatics* 18, 412. doi:10.1186/s12859-017-1817-3.
- Zhou, J., Lamichhane, S., Sterne, G., Ye, B., Peng, H., 2013. Biocat: A pattern recognition platform for customizable biological image classification and annotation. *BMC Bioinformatics* 14, 291.
- Zitová, B., Flusser, J., 2003. Image registration methods: A survey. *Image and Vision Computing* 21, 977–1000. doi:10.1016/s0262-8856(03)00137-9.

Bioimage Databases

Arpah Abu and Sarinder K Dhillon, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

In biology, image data can be classified as high-resolution microscopy data, medical imaging data, observed effects in cell, high content screening, molecular imaging and biodiversity (organisms) data. These data are generated via advance experimental and field work as well as from high throughput analysis using the imaging technology. Besides images of organisms, images of protein structures and structural models of proteins and RNA molecules are also available for browsing, querying as well as analysis. Besides the raw source of images obtained, the annotations that describe the related details are valuable source of information. However, till date, very little has been reported in the development of bioimage databases, particularly for storing and pooling the vast amount of scientific image data both at molecular or cellular level. Nevertheless, a substantial amount of work has been done in documenting digital pictures of organisms as in terms of technology it is much simple and affordable.

Generally, bioimage databases are developed for storage and retrieval purposes. The data formats used by these databases are specific to the type of images stored. Retrieval of data from image databases can be done in two ways; the first method accepts only text query to retrieve images (or the content of the database) and the other method accepts image query which referred as content-based image retrieval. In this article, existing work done using both of these approaches are presented. The problems concerning bioimage database are discussed and finally we present a viable solution to develop bioimage databases.

Bioimage Databases Using Text-Based Query

In this section, some of the important and widely used bioimage databases are presented. These databases are text query based, which means the input in the search function is string rather than an image.

Flybase (see “Relevant Websites section”)

FlyBase (McQuilton *et al.*, 2012) is an image database of *Drosophila* genes and genomes. One of the query tools provided in this system is ImageBrowse (see Fig. 1) for browsing the images based on the organ system, life-cycle stages, major tagma, germ layer and all species images.

The images in this database were collected from the researchers' publications such as journal articles and books. The retrieved images are listed in unranked order along with short description (see Fig. 2).

Planetary Biodiversity Inventory – PBI (see “Relevant Websites section”)

The Planetary Biodiversity Inventory – PBI (Gregorev *et al.*, 2003) for Plant Bugs provides a resource about the global plant bug subfamilies Orthotylinae and Phylinae (Insecta: Heteroptera: Miridae) to various users such as systematists, insect ecologists, conservation biologists, the general public, and students via the Internet. In the latest version 6, all users have access to a systematic catalog of plant bug taxa with more than 10,000 species, a digital library of ~30,000 pages, locality database, and updated Methods and News and Outputs. Furthermore, users can view taxon treatments and distribution maps through discoverlife.org by using the search box provided on the page as well as through a dedicated web page, provided in “Relevant Websites section”. Through these portals specimen records are available for approximately 4000 species and more than 250,000 specimens. 56 papers have so far been published with at least 15 additional papers in preparation or in press.

The Plant Bug PBI was made possible with the financial support from the National Science Foundation (NSF) from September 1, 2003 to August 31, 2011.

The images along with taxon information and distribution can be retrieved using Image Database interface in Fig. 3. Users can enter the relevant bug name such as ant, bee and wasp and the retrieved images are listed in unranked order.

Specimen Image Database – SID (see “Relevant Websites section”)

Specimen Image Database – SID (Simon and Vince, 2011) is a searchable database of high-resolution images for phylogenetic and biodiversity research. This database is intended as a reference collection of named specimens and a resource for comparative morphological research. Each image is accompanied by fully searchable annotation, and can be browsed, searched or downloaded. Public users as well can register in this database and the registered users can add, annotate or label the images. Currently, this database is devoted to the insect order Phthiraptera (lice) and contains 7650 images of 440 taxa.

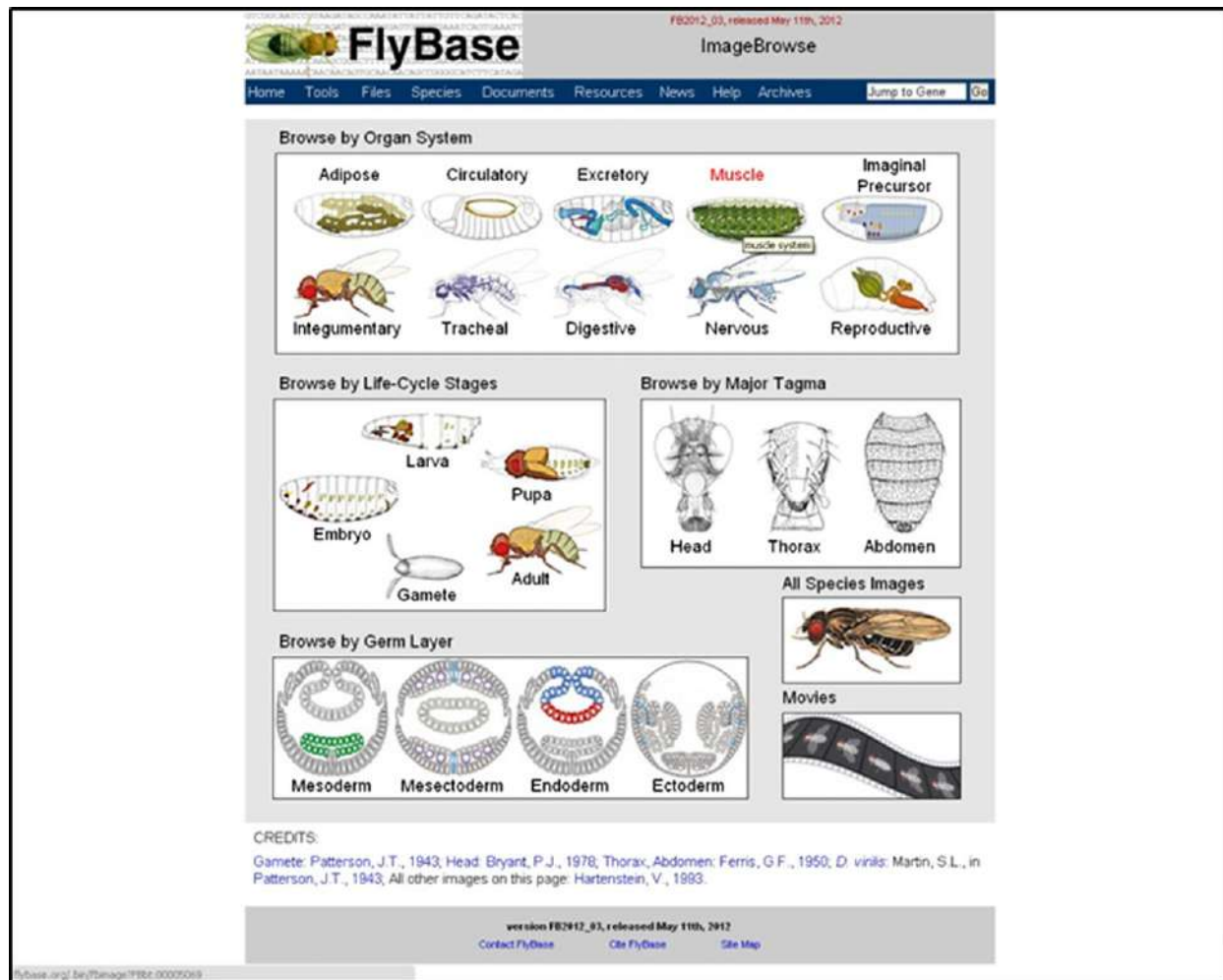


Fig. 1 ImageBrowse in FlyBase.

Key features of SID (see Fig. 4) include web upload/download of images, bulk and single image annotation via web forms, extensive browse and search options by text query, web service facility, web utility to label specific image features, taxonomy served and validated independently by the Glasgow Taxonomy Name Server, alias addresses for images by accession number and freeware which allows anyone to set up the database and serve their own images. The retrieved images are listed in unranked order with the taxon information, host as well as image properties.

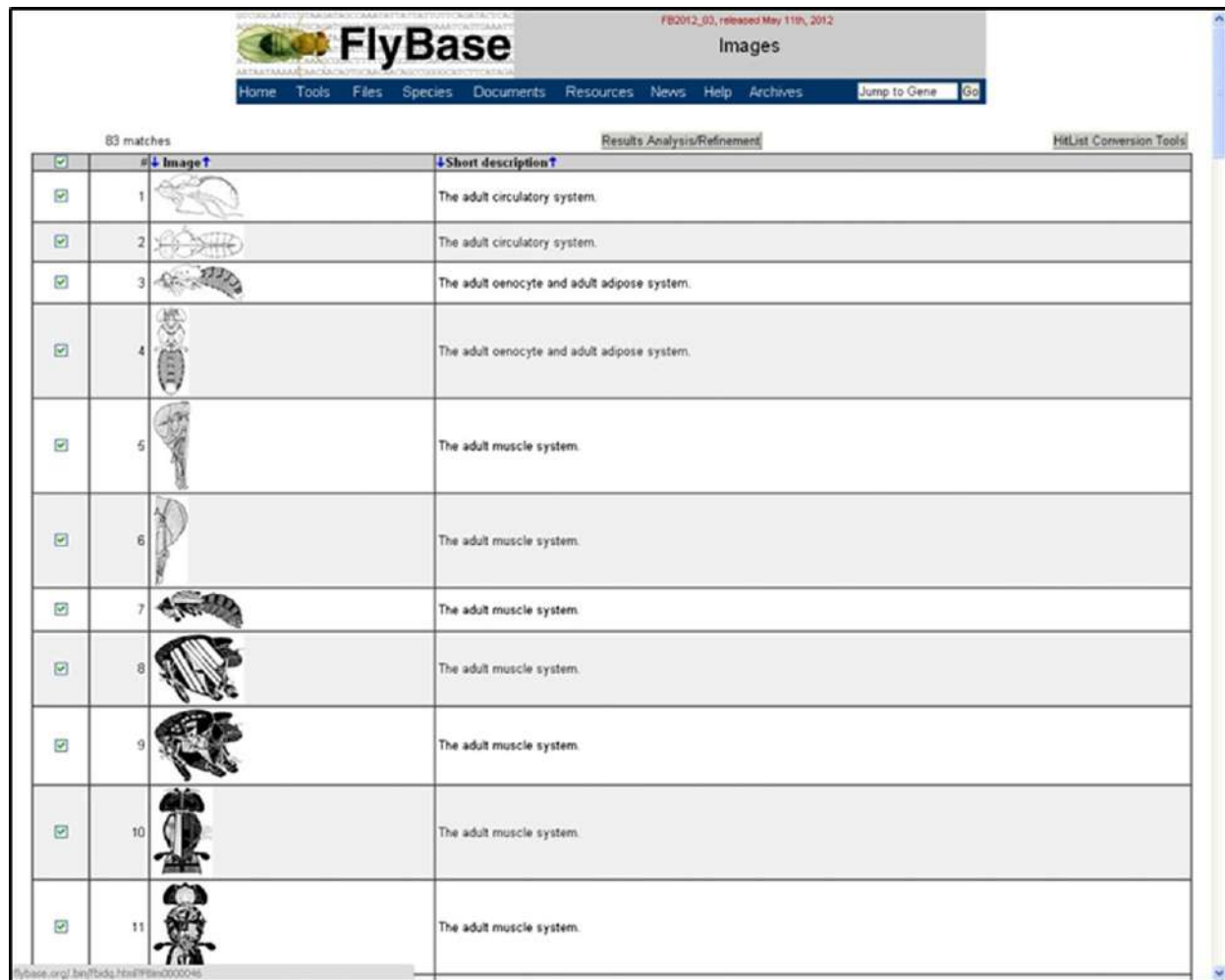
Orchidaceae of Central Africa (see “Relevant Websites section”)

Orchidaceae of central Africa (Droissart *et al.*, 2012) is an image database on orchids of Central Africa. It includes scientific names, distribution data, photos, interactive identification keys, web links, and references. The images presented in this database are from intensive fieldwork carried out by the Université Libre de Bruxelles, the Université de Yaoundé I, the Missouri Botanical Garden and the Institut de Recherche pour le Développement in several countries of Central Africa. This information will promote land conservation and sound natural resource management in tropical Africa. The information presented in this database may be suitable for academic, educational, and general purpose.

The database currently comprises information on about 300 illustrated orchid species. The images can be retrieved by using text query based on the taxon name; or browsing the entire database or with few options such as taxa by genus, country or author as shown in Fig. 5. The retrieved images are listed in unranked order with species and authors name.

MonoDb (see “Relevant Websites section”)

MonoDb (Andy and James, 2012) is another biodiversity database that provides image gallery as one of the features in the database. MonoDb is a web-host for the parasite monogenea. As mentioned in this website, the purpose of this website is to help children, adults, experts and non-experts to learn more about this fascinating group of animals. Images in this database can be



83 matches			Results Analysis/Refinement	HitList Conversion Tools
☒	# Image ↑	Short description ↑		
☒	1		The adult circulatory system.	
☒	2		The adult circulatory system.	
☒	3		The adult oenocyte and adult adipose system.	
☒	4		The adult oenocyte and adult adipose system.	
☒	5		The adult oenocyte and adult adipose system.	
☒	6		The adult muscle system.	
☒	7		The adult muscle system.	
☒	8		The adult muscle system.	
☒	9		The adult muscle system.	
☒	10		The adult muscle system.	
☒	11		The adult muscle system.	

Fig. 2 Retrieval results from FlyBase.

retrieved by browsing the entire images provided in the database. The images are listed randomly and no information attached to the images (see [Fig. 6](#)).

Cell Image Library – CIL-CCDB (see “Relevant Websites section”)

CIL-CCDB ([Orloff et al., 2013](#)) is a searchable database and archive of cellular images. As a repository for microscopy data, it accepts all forms of cell imaging from light and electron microscopy, including multi-dimensional images, Z- and time stacks in a broad variety of raw-data formats, as well as movies and animations. A CIL-CCDB was developed as a resource and a tool for the cell biology community such as for researchers, educators and the general public. Images in this database can be retrieved by browsing the entire images provided in the database and by using the simple or advance search features as shown in [Fig. 7](#).

ModBase: Database of Comparative Protein Structure Models (see “Relevant Websites section”)

MODBASE ([Ursula et al., 2014](#)) is a queryable database of annotated protein structure models. The models are derived by ModPipe, an automated modelling pipeline relying on the programs PSI-BLAST and MODELLER. The database also includes the fold assignments and alignments on which the models were based. ModBase was developed by Sali Lab. [Fig. 8](#) shows the main home page of ModBase. The data was organized into datasets in which can be used either available to the public, to the academicians or to specific users.

SWISS-MODEL Repository (see “Relevant Websites section”)

The SWISS-MODEL Repository ([Bienert et al., 2017](#)) as shown in [Fig. 9](#) is a database of annotated 3D protein structure models generated by the SWISS-MODEL homology-modelling pipeline. It was developed by Swiss Institute of Bioinformatics. The aim of

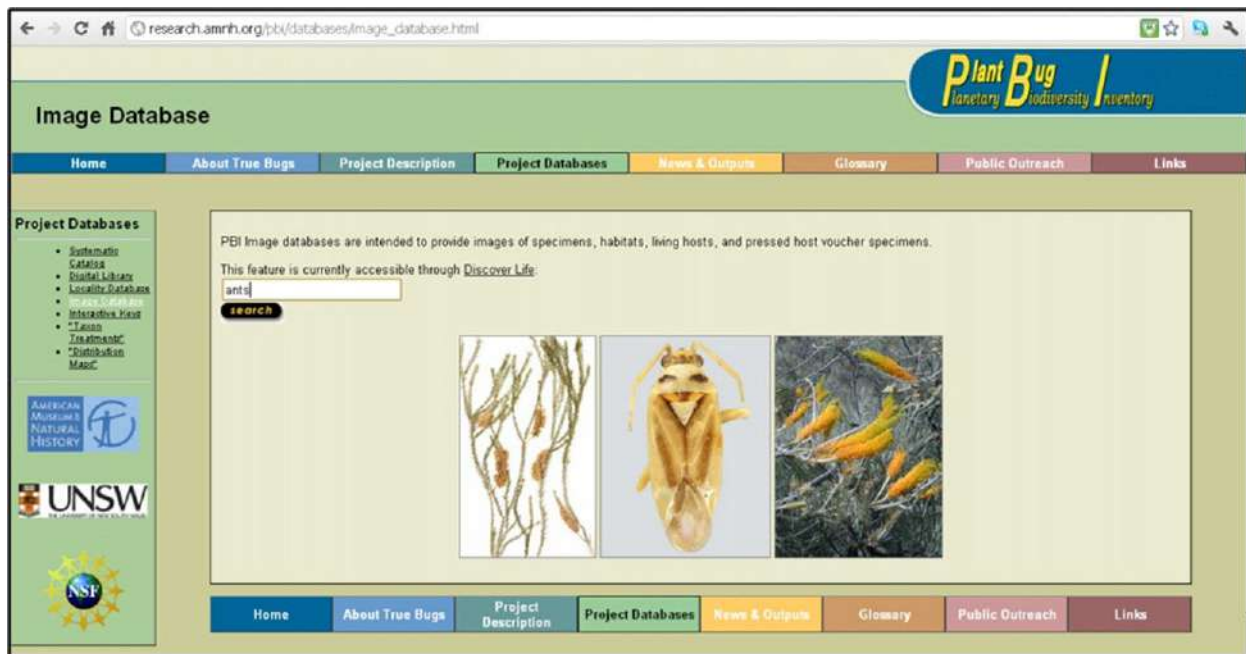


Fig. 3 PBI – Image Database interface.

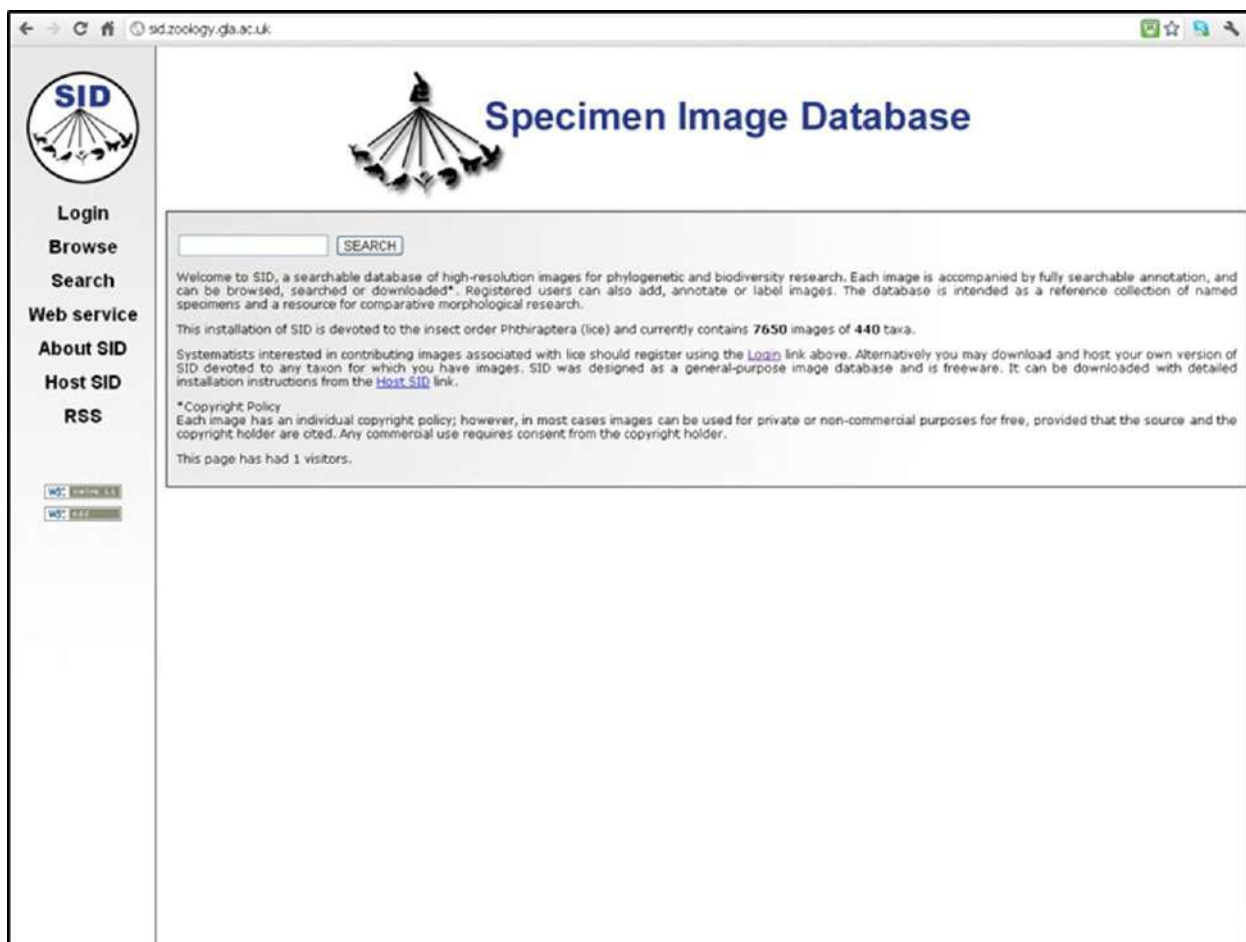


Fig. 4 SID – Search page interface.



Fig. 5 Orchidaceae of central Africa – Search page interface.

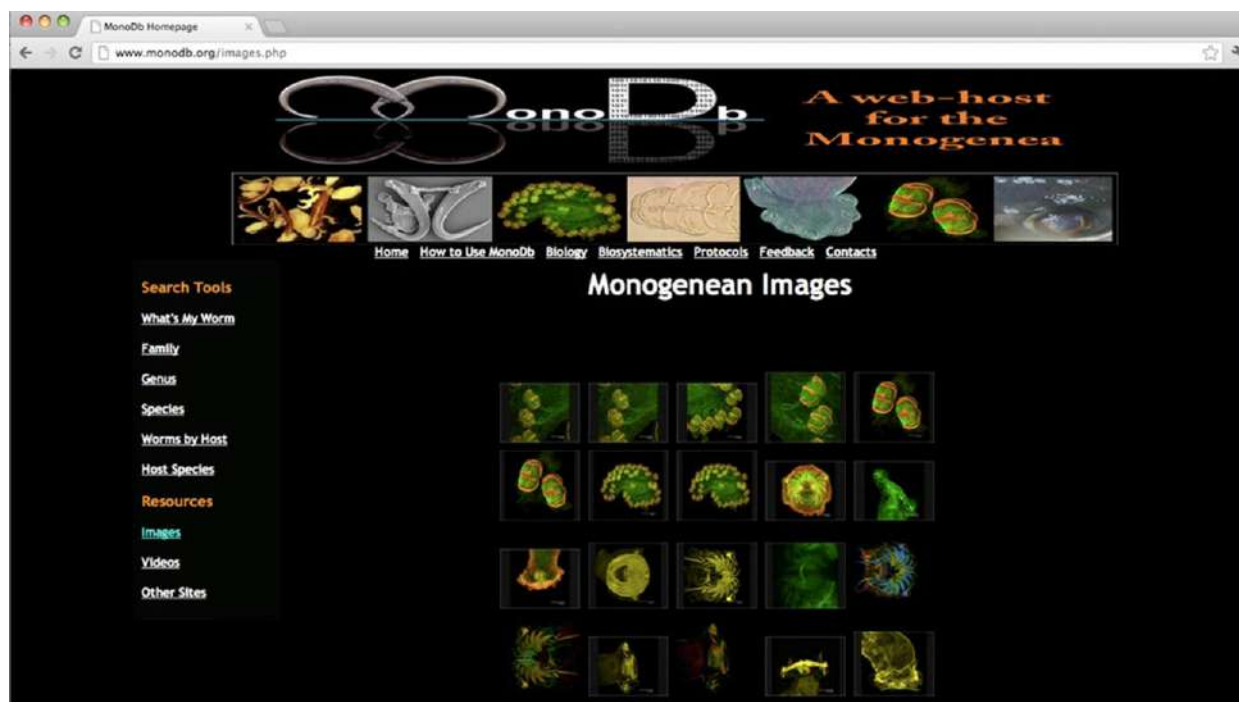


Fig. 6 Monogenean images in MonoDb.

the SWISS-MODEL Repository is to provide access to an up-to-date collection of annotated 3D protein models generated by automated homology modelling for relevant model organisms and experimental structure information for all sequences in UniProtKB. Regular updates ensure that target coverage is complete, that models are built using the most recent sequence and template structure databases, and that improvements in the underlying modelling pipeline are fully utilized. It also allows users to assess the quality of the models using the latest QMEAN results. If a sequence has not been modeled, the user can build models interactively via the SWISS-MODEL workspace.

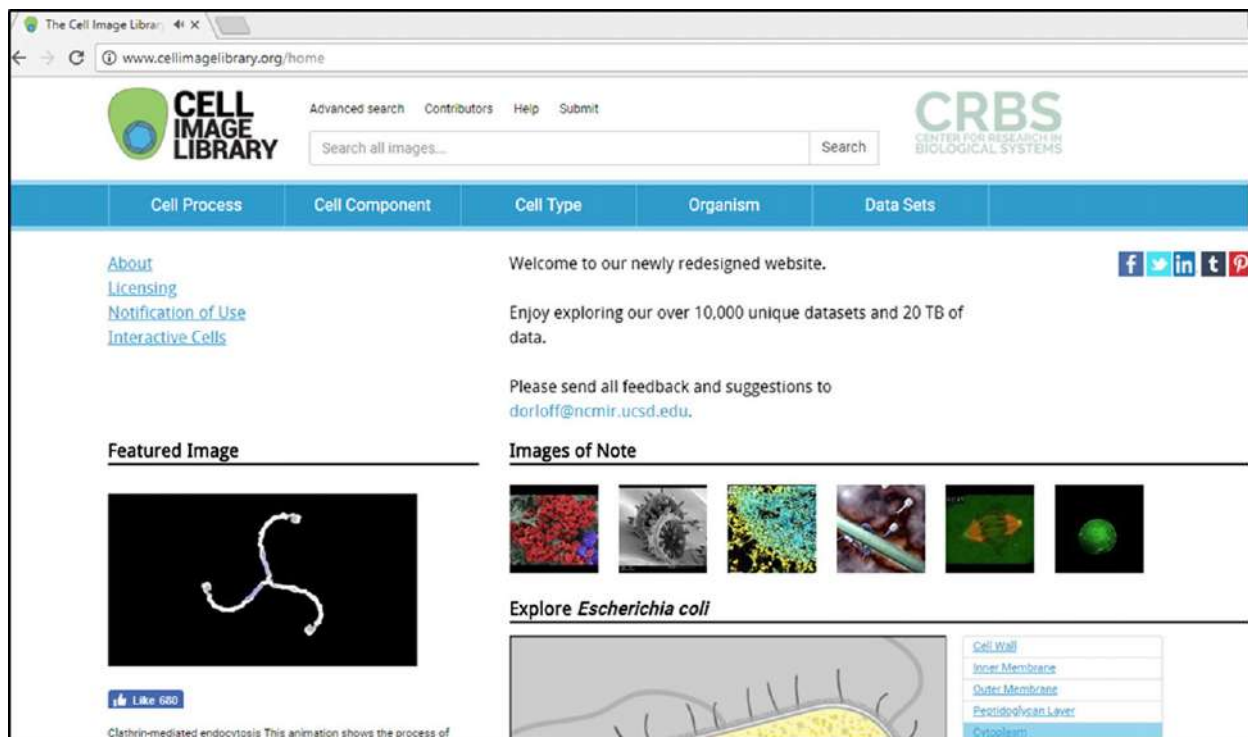


Fig. 7 Home page of CIL-CCDB.

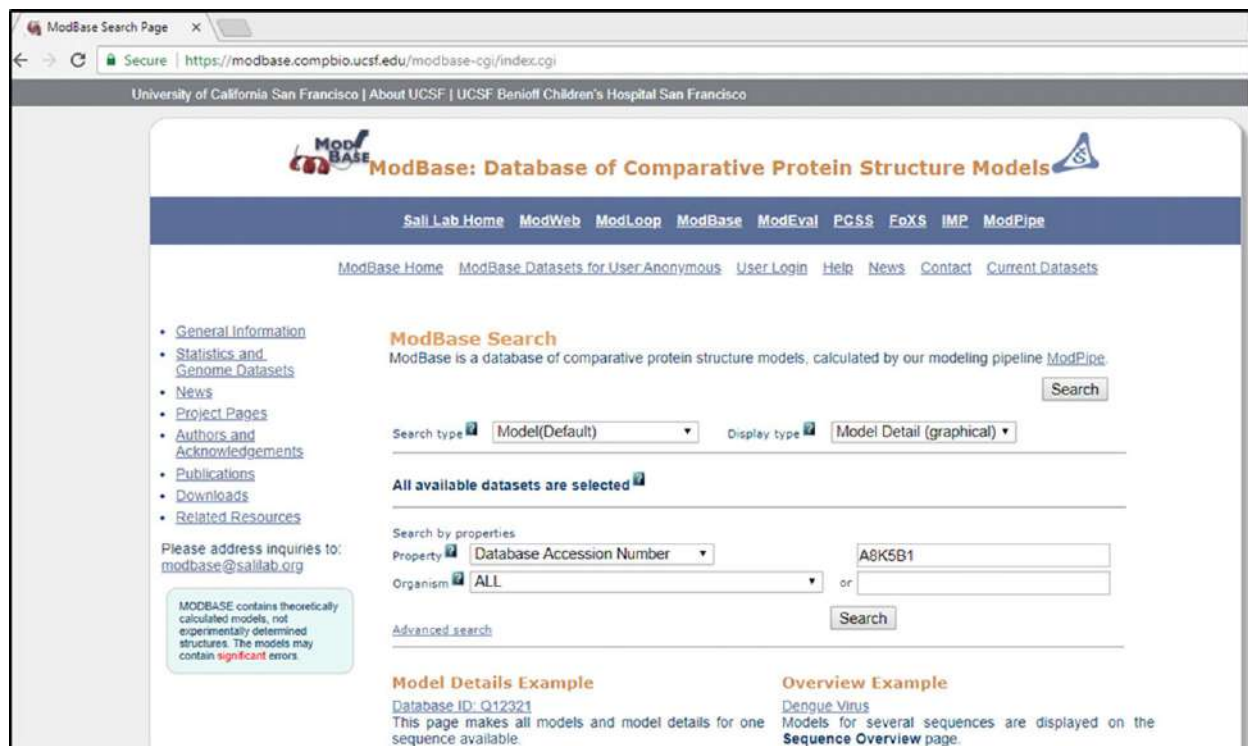
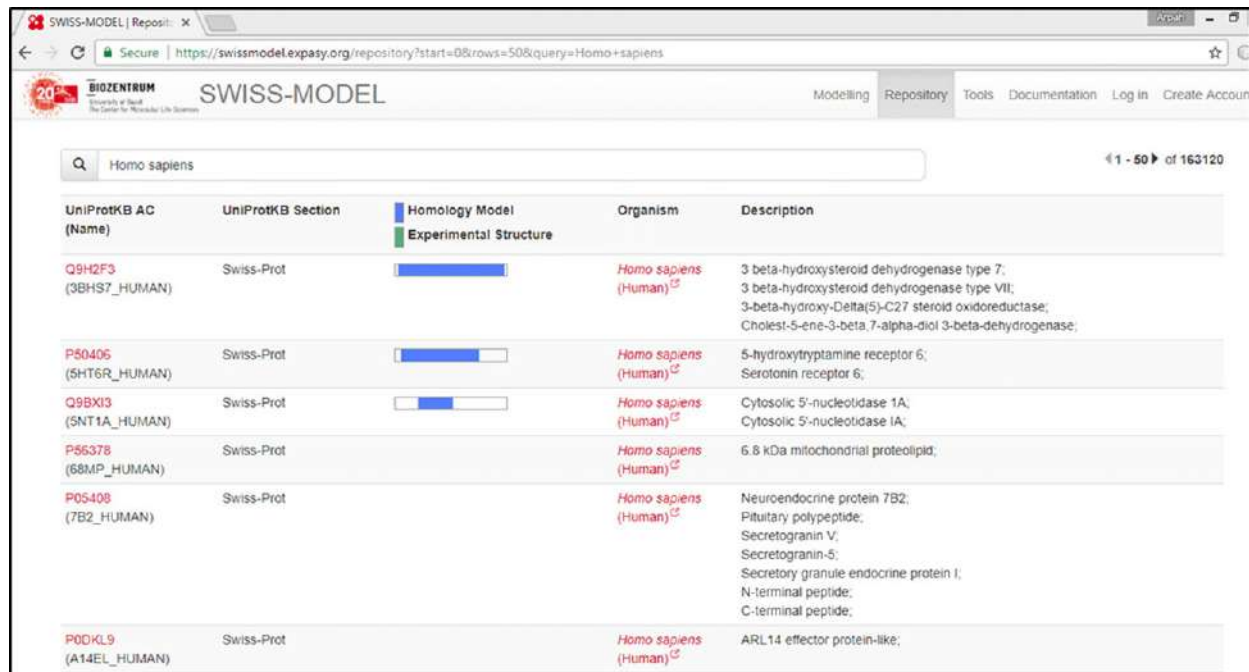


Fig. 8 ModBase home page.



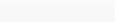
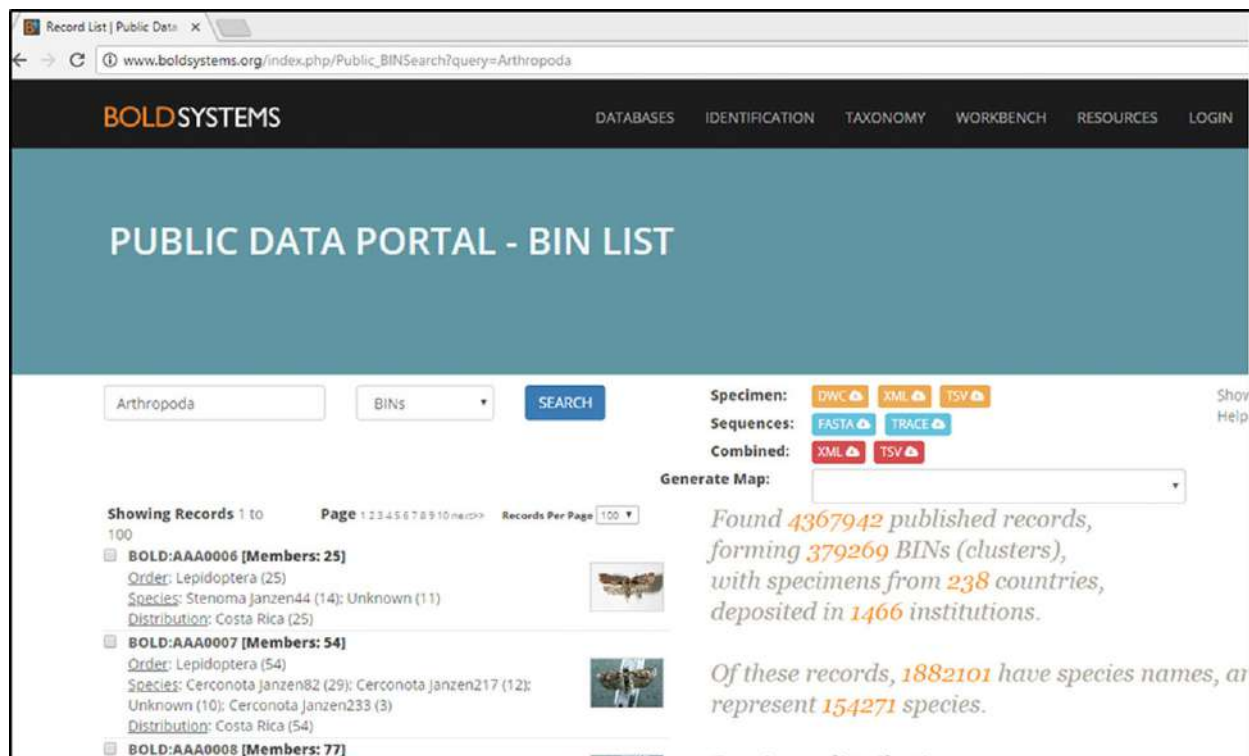
UniProtKB AC (Name)	UniProtKB Section	Homology Model Experimental Structure	Organism	Description
Q9H2F3 (3BHS7_HUMAN)	Swiss-Prot		Homo sapiens (Human)	3 beta-hydroxysteroid dehydrogenase type 7; 3 beta-hydroxysteroid dehydrogenase type VII; 3-beta-hydroxy-Delta(5)-C27 steroid oxidoreductase; Cholest-5-ene-3-beta,7-alpha-diol 3-beta-dehydrogenase;
P50406 (5HT6R_HUMAN)	Swiss-Prot		Homo sapiens (Human)	5-hydroxytryptamine receptor 6; Serotonin receptor 6;
Q9BXI3 (SNT1A_HUMAN)	Swiss-Prot		Homo sapiens (Human)	Cytosolic 5'-nucleotidase 1A; Cytosolic 5'-nucleotidase 1A;
P56378 (68MP_HUMAN)	Swiss-Prot		Homo sapiens (Human)	6.8 kDa mitochondrial proteolipid;
P05408 (7B2_HUMAN)	Swiss-Prot		Homo sapiens (Human)	Neuroendocrine protein 7B2; Pituitary polypeptide; Secretogranin V; Secretogranin-5; Secretory granule endocrine protein I; N-terminal peptide; C-terminal peptide;
P00KL9 (A14EL_HUMAN)	Swiss-Prot		Homo sapiens (Human)	ARL14 effector protein-like;

Fig. 9 Search page in SWISS MODEL.



BOLD SYSTEMS DATABASES IDENTIFICATION TAXONOMY WORKBENCH RESOURCES LOGIN

PUBLIC DATA PORTAL - BIN LIST

Arthropoda BINS **SEARCH**

Specimen: **DWC** **XML** **TSV**
Sequences: **FASTA** **TRACE**
Combined: **XML** **TSV**
Generate Map:

Showing Records 1 to 100 Page 1 2 3 4 5 6 7 8 9 10 next Records Per Page 100

- BOLD:AAA0006 [Members: 25]**
Order: Lepidoptera (25)
Species: Stenoma janzen44 (14); Unknown (11)
Distribution: Costa Rica (25)
- BOLD:AAA0007 [Members: 54]**
Order: Lepidoptera (54)
Species: Cerconota janzen82 (29); Cerconota janzen217 (12); Unknown (10); Cerconota janzen233 (3)
Distribution: Costa Rica (54)
- BOLD:AAA0008 [Members: 77]**

Found **4367942** published records, forming **379269** BINs (clusters), with specimens from **238** countries, deposited in **1466** institutions.

Of these records, **1882101** have species names, and represent **154271** species.

Fig. 10 Search the images using BIN feature.

BOLD SYSTEMS (see “Relevant Websites section”)

BOLD is a cloud-based data storage and analysis platform developed at the Centre for Biodiversity Genomics in Canada. It consists of four main modules, a data portal, an educational portal, a registry of BINs (putative species), and a data collection and analysis

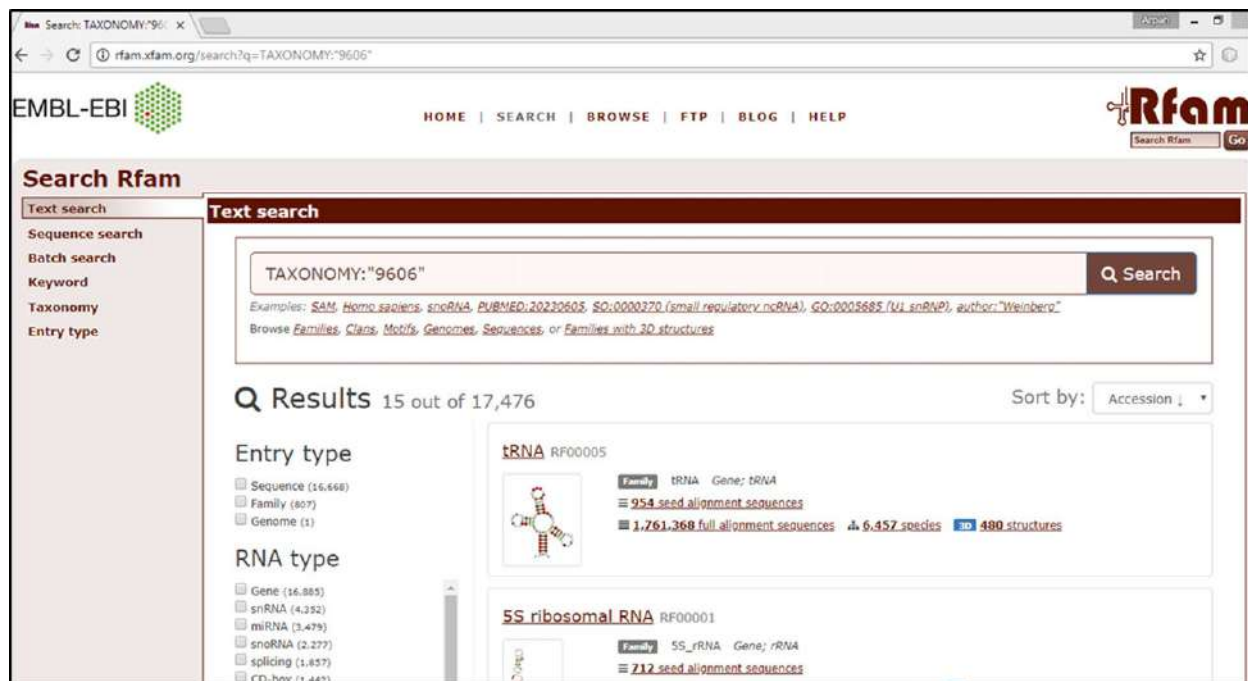


Fig. 11 Rfam searching page.

workbench. The BIN database as shown in [Fig. 10](#) is a searchable database of Barcode Index Numbers (BINs), sequence clusters that closely approximate species. The retrieved images are displayed with their description that give meaning to the images.

Rfam (see “Relevant Websites section”)

The Rfam ([Kalvari et al., 2017](#)) database is a collection of RNA families, each represented by multiple sequence alignments, consensus secondary structures and covariance models (CMs). The Rfam database is curated and maintained at the European Bioinformatics Institute in Cambridge, UK. The resource is collaboration with researchers from Sean Eddy lab at Harvard University, USA. The secondary structures images are retrieved as results with annotated information using a text-based query as shown in [Fig. 11](#).

Bioimage Databases Using Content-Based Image Retrieval

Content-based Image Retrieval (CBIR) is a concept used in image databases. The work presented in Section “Bioimage Databases Using Text-Based Query” are databases with text-based retrieval capabilities. Hence the functionality is limited in the sense that only text query can be used to retrieve images, unlike CBIR database systems, whereby the query input can be an image. Due to the escalation of biological data, new ways have been introduced to store and analyze data. The need of efficient management of the vast visual information has introduced new techniques and tools for content-based image retrieval approach.

[Fig. 12](#) shows a typical architecture of CBIR system depicted from [Torres and Falcao \(2006\)](#). The goal of this approach is to search and retrieve a set of similar images to the user query. The interface layer allows user to send a query image. The images from image database are then assigned as training set images. Both query and training set images features (such as shape, texture and color) are extracted and formed the feature vectors in the feature space. The similarity comparison (such as Euclidean distance and Mahalanobis distance) between the query and training set images are then measured, and the classifier (such as minimum distance, maximum distance and k-nearest classifier) is used to classify the retrieved images. The results are then returned to the user through user interface. As for the results, the retrieved images must be accurate, relevant and related to the user query.

[Rui et al. \(1999\)](#), [Smeulders et al. \(2000\)](#), [Feng et al. \(2003\)](#) and [Torres and Falcao \(2006\)](#) introduced some fundamental techniques as well as technical achievements in the field of CBIR.

As mentioned previously, biologists produce a vast of digital images. From these images, it can be used for identification as well as for teaching and research. [Wang et al. \(2012\)](#) presents a work on butterfly family identification using CBIR, while [Sheikh et al. \(2011\)](#) developed CBIR system for various types of marine life images. A content-based pattern analysis system for a biological specimen collection was developed by [Joyita and Gardner \(2007\)](#) and a CBIR system for fish taxonomy research was also

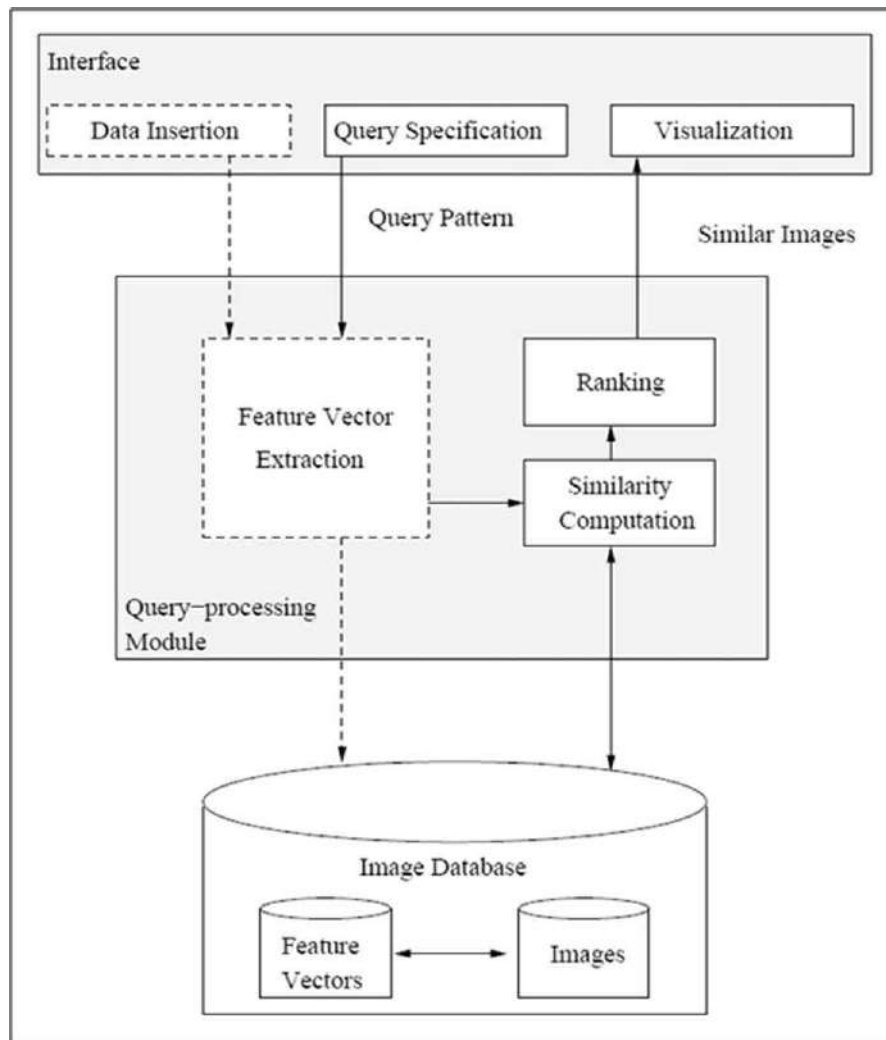


Fig. 12 A typical architecture of CBIR system.

developed by [Chen et al. \(2005\)](#). As stated in [Wang et al. \(2012\)](#), CBIR is applied because of its capacity for mass processing and operability.

On the other hand, because of the heterogeneous data, complexity of the biology images as well as the images descriptions are often ignored to attach to together with the images, there are few works such as mentioned in [EKEY \(2012\)](#), [Torres et al. \(2004\)](#) and [Murthy et al. \(2009\)](#) to enhance the CBIR capability

The CBIR approach has been applied in medical for teaching, research and diagnostics on diseases. The benefits and future directions have been discussed in [Müller et al. \(2004\)](#). In [Kak and Pavlopoulou \(2002\)](#), CBIR is used to automate retrieval from large medical image databases and presented solutions to some of them in the specific context of HRCT images of lung and liver. [Scott and Chi-Ren \(2007\)](#) presents a knowledge-driven multidimensional indexing structure for biomedical media database retrieval. While in [El-Naqa et al. \(2004\)](#) and [Rosa et al. \(2008\)](#), they use CBIR approach for digital mammographic masses. As to improve the retrieval efficiency, some of the works such as mentioned in [Demner-Fushman et al. \(2009\)](#), [Hsu et al. \(2009\)](#) and [You et al. \(2011\)](#), they have combining the metadata approach into CBIR approach.

The CBIR approach has been used as well in several applications such as face identification, digital libraries, historical research, medical and geology. It is probably the most useful application in biology. In this article, some of these applications are presented as in this section, focusing on medical and biology applications.

Google Image Search (see “Relevant Websites section”)

Google Image Search is a Google’s CBIR system. Query specification follows the query by example using external images. However, it does not work on all images. Metadata query function is also provided.

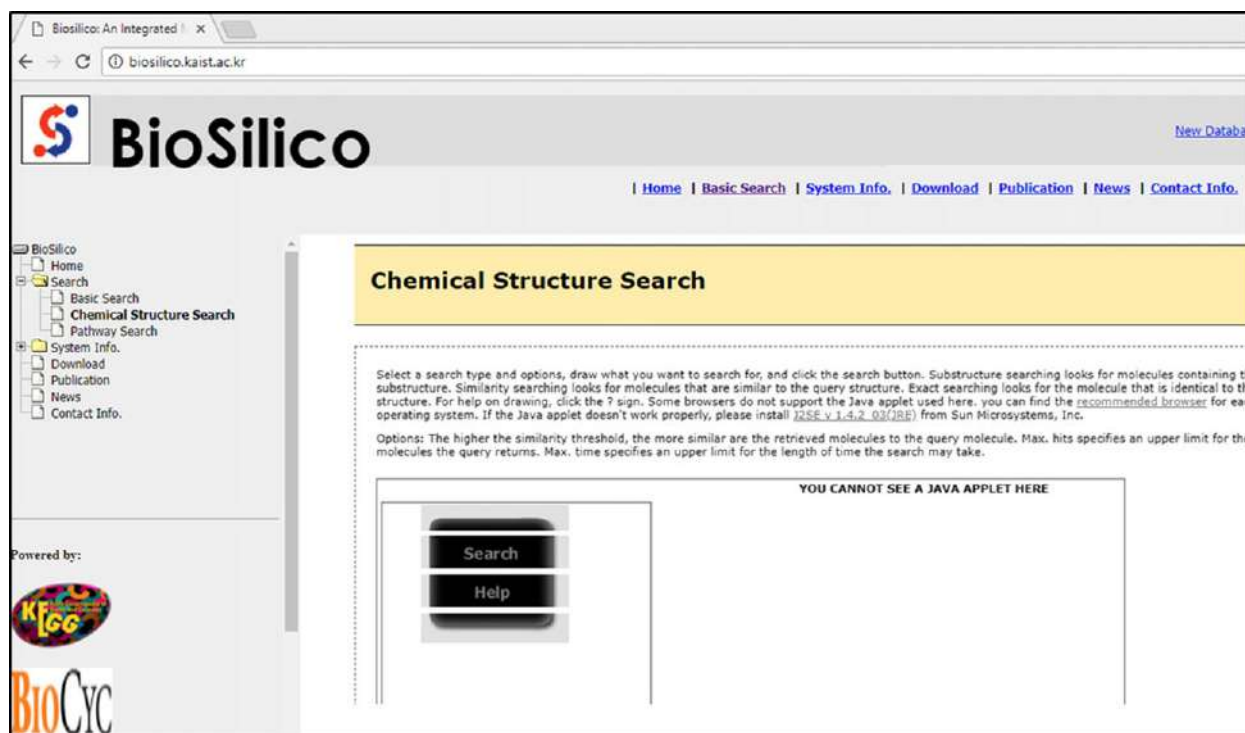


Fig. 13 Chemical structure search using CBIR in BioSilico.

Macroglossa Visual Search (see “Relevant Websites section”)

Macroglossa is a visual search engine based on the comparison of images, coming from an Italian Group. For image retrieval, query specification follows the query by example using external images or query by category such animals, biological, panoramic, artistic or botanical. Macroglossa supports all popular image extensions such jpeg, png, bmp, gif and video formats such avi, mov, mp4, m4v, 3gp, wmv, mpeg.

BioSilico (see “Relevant Websites section”)

BioSilico is a web-based database system that facilitates the search and analysis of metabolic pathways. Heterogeneous metabolic databases including LIGAND, ENZYME, EcoCyc and MetaCyc are integrated in a systematic way, thereby allowing users to efficiently retrieve the relevant information on enzymes, biochemical compounds and reactions. In addition, it provides well-designed view pages for more detailed summary information as shown in [Fig. 13](#).

CATH (see “Relevant Websites section”)

The CATH ([Sillitoe et al., 2015](#)) database is a free, publicly available online resource that provides information on the evolutionary relationships of protein domains. It was created in the mid-1990s by Professor Christine Orengo and colleagues, and continues to be developed by the Orengo group at University College London. The CATH database is a hierarchical domain classification of protein structures in the Protein Data Bank. Protein structures are classified using a combination of automated and manual procedures. There are four major levels in this hierarchy which are Class, Architecture, Topology, and Homologous superfamily. For any given structure classified in the database, CATH gives you information on the structure and function of that protein. The evolutionary relationships involving the structure of interest and other proteins in the database can also be determined. These tasks can be performed by searching using Text or ID, Sequence or Structure in PDB image format as shown in [Fig. 14](#).

IDR – Image Data Resource (see “Relevant Websites section”)

IDR ([Williams et al., 2017](#)) is online, public data repository seeks to store, integrate and serve image datasets from published scientific studies. It was developed by University of Dundee & Open Microscopy Environment. IDR allow the community to search, view, mine and even process and analyze large, complex, multidimensional life sciences image data as shown in [Fig. 15](#). Sharing data promotes the validation of experimental methods and scientific conclusions, the comparison with new data obtained by the global scientific community and enables data reuse by developers of new analysis and processing tools.

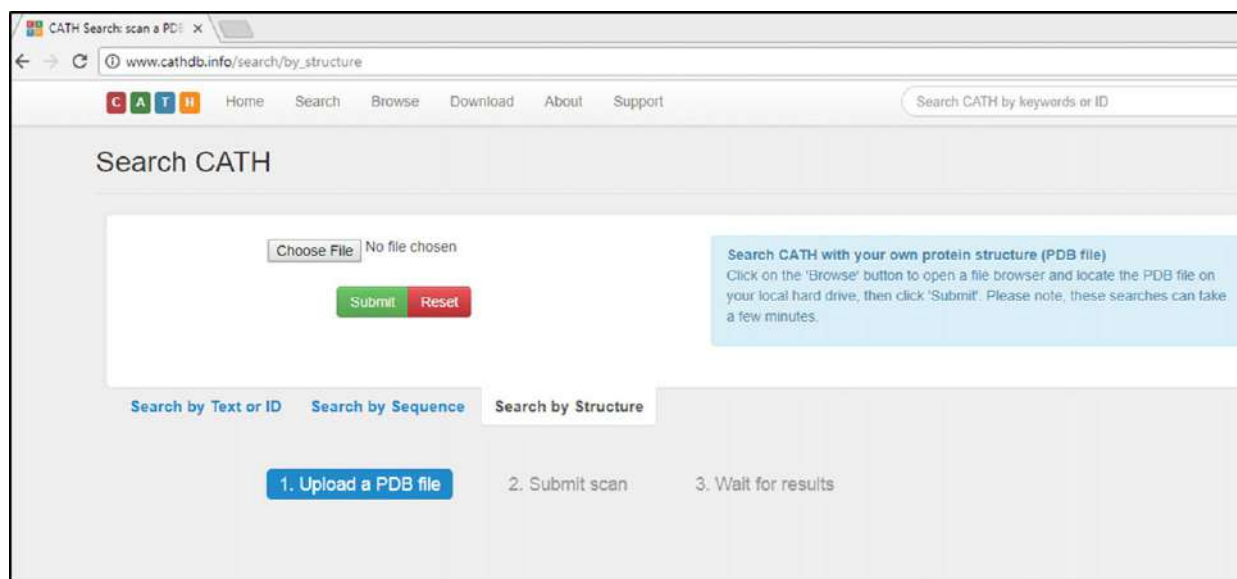


Fig. 14 CATH image database using image query.

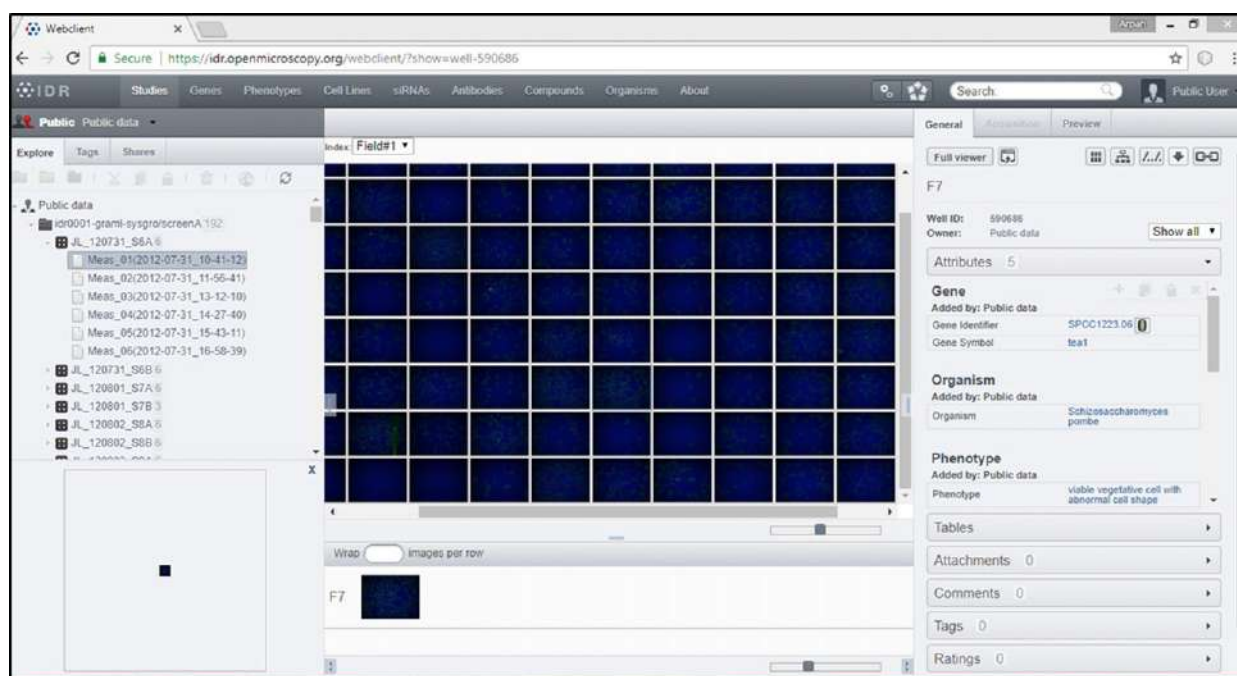


Fig. 15 Retrieved images from fungi dataset.

BioDIG (see “Relevant Websites section”)

BioDIG (Oberlin *et al.*, 2013) is a generic set of tools that allows linking of image data to genomic data. BioDIG features the following: rapid construction of web-based workbenches, community-based annotation, user management and web services. By using BioDIG to create websites, researchers and curators can rapidly annotate a large number of images with genomic information. The BioDIG includes an image module, a genome module and a user management module as shown in Fig. 16.

RNA 3D Motif Atlas (see “Relevant Websites section”)

RNA 3D Motif Atlas (Petrov *et al.*, 2013) is a comprehensive and representative collection of internal and hairpin loop RNA 3D motifs extracted from the Representative Sets of RNA 3D structures. It was developed by BGSU RNA Structural Bioinformatics at

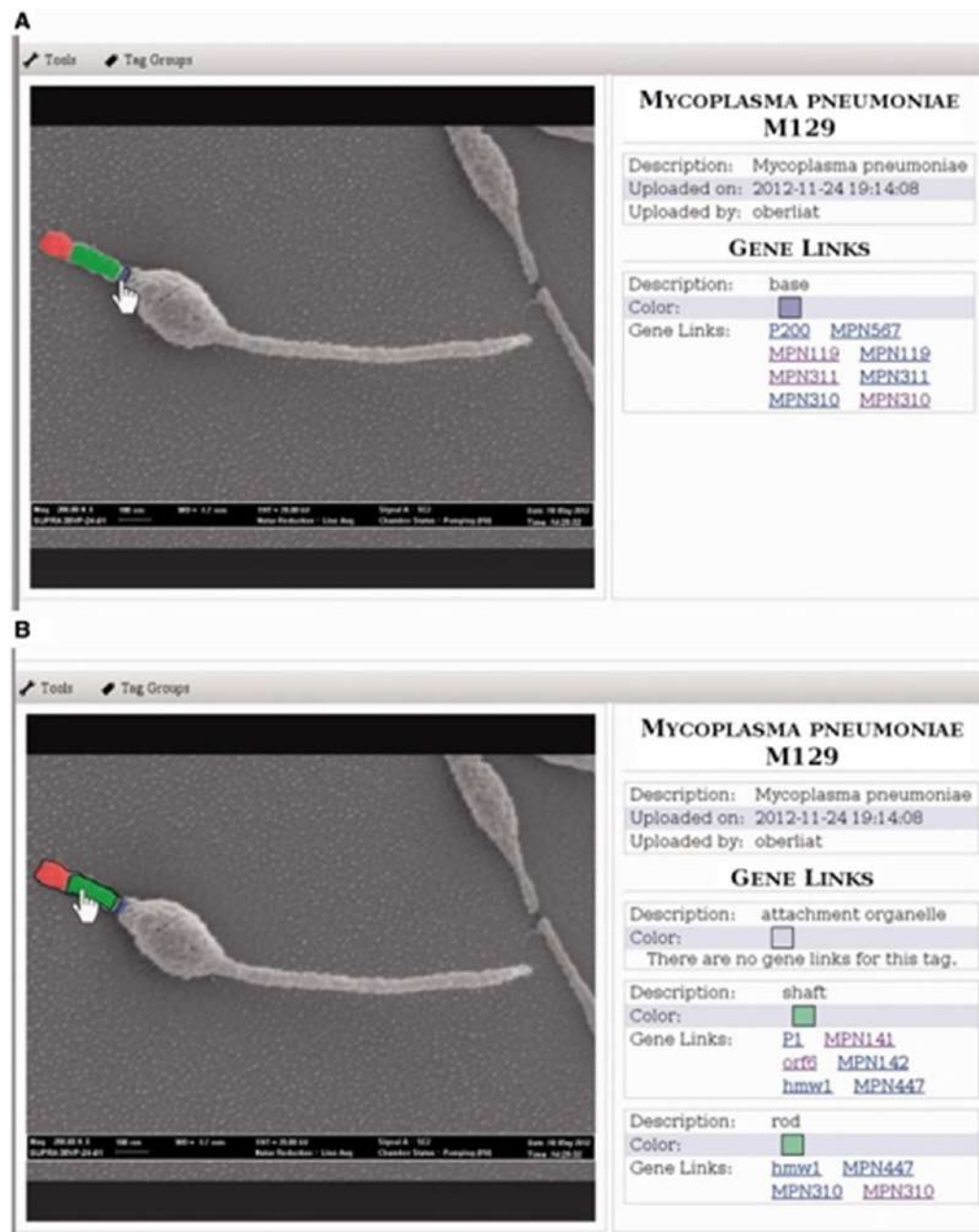


Fig. 16 The *Mycoplasma pneumonia* image annotated with genome data in BioDIG interface.

Bowling Green State University. The 3D images of RNA motif are retrieved using browsing featured motifs which are Kink-turn (as shown in Fig. 17), C-loop, Sarcin, Triple sheared, Double sheared T-loop, and GNRA.

Discussion on Bioimage Databases

Many issues have been discussed and suggested (Rui *et al.*, 1999; Smeulders *et al.*, 2000; Shandilya and Singhai, 2010) to improve the typical Content-based Image Retrieval (CBIR) based image databases such as involve user interaction, integration of multi-disciplines approach, relevance feedback and reducing semantic gap. The main purpose of these improvements is to enhance the efficiency of image retrieval.

Most previous works focused on image representation (Krishnapuram *et al.*, 2004; Wei *et al.*, 2006; Lamard *et al.*, 2007; Sergyan, 2008), classifier algorithm (Xin and Jin, 2004; Duan *et al.*, 2005; Liu *et al.*, 2008), the use of image database (Kak and Pavlopoulou, 2002), and relevance feedback (Stejić *et al.*, 2003; Zhang *et al.*, 2003; Ortega-Binderberger and Mehrotra, 2004;

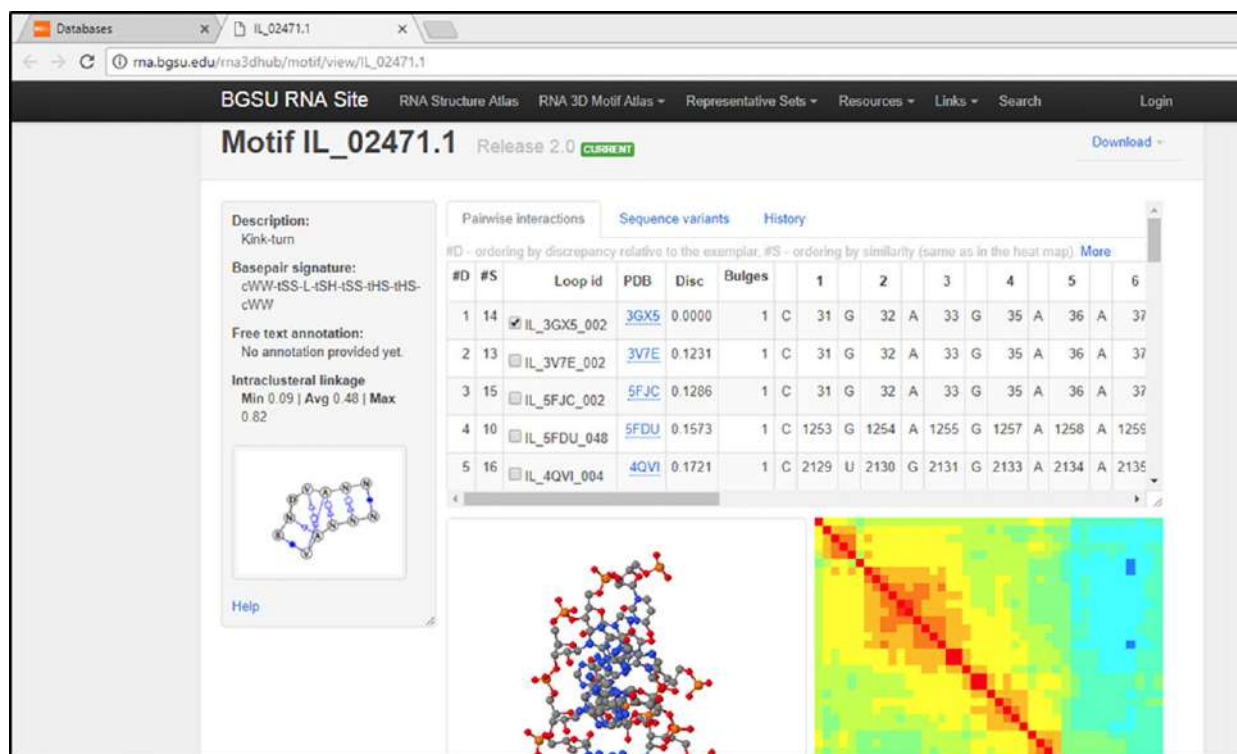


Fig. 17 Retrieved RNA 3D motifs using Kink-turn featured motif.

Wang and Ma, 2005; Wei and Li, 2006) to enhance the CBIR system. However, as mentioned in Liu *et al.* (2007), research focus has been shifted into reducing the semantic gap and identified five major categories of the state-of-the-art techniques in narrowing down the semantic gap i.e., (i) using object ontology to define high-level concepts; (ii) using machine learning methods to associate low-level features with query concepts; (iii) using relevance feedback to learn users' intention; (iv) generating semantic template to support high-level image retrieval; (v) fusing the evidences from HTML text and the visual content of images for WWW image retrieval.

On the other hand, as stated in Torres *et al.* (2004), the implementation of CBIR systems raises several research challenges such as (i) new tool for annotating needs to be developed to deal with the semantic gap presented in images and their textual descriptions; (ii) automatic tool for extracting semantic features from images; (iii) development of new data fusion algorithm to support text-based and content-based retrieval when combining information of different heterogeneous formats; (iv) text mining techniques to be combined with content-based descriptions; and (v) investigating user interfaces for annotating, browsing and searching based on image content.

Related works focusing on reducing the semantic gap between the visual (low-level) features and the richness of human semantic (high-level features) is in progress. Lin *et al.* (2007) proposes the integration of textual and visual information for cross-language image retrieval; Zhang *et al.* (2011) presents automatic image tagging automatically assigns image with semantic keyword called tag, which significantly facilitates image search and organization; Aye and Thein (2012) presents a retrieval framework can support various types of queries and can accept multimedia examples and metadata-based document; and Lee and Wang (2012) presents a utilization of text- and photo- types of location information with a novel approach of information fusion that exploit effective image annotation and location based text-mining approaches to enhance identification of geographic location and spatial cognition.

Generally, there are two approaches to image retrieval which are textual-based that retrieves images based on human-annotated metadata, and content-based that analyzes the actual image data (Avril, 2005). Many digital libraries have annotation capabilities for image digital libraries such as Imense Image Search Portal (imense, 2007), Incogna Image Search (Incogna, 2012), and Anaktisi (Chatzichristofis *et al.*, 2010). Yet, in biology, very few are providing the capabilities to integrate text- and content-based annotation and retrieval of parts of images such as EKEY (EKEY, 2012), SuperIDR (Murthy *et al.*, 2009), and teaching tool for parasitology (Kozievitch *et al.*, 2010). EKEY is a web-based system that provides taxonomic classification, dichotomous key, text-based search and combination of shape and text-based search which taking into account fish shape outlines and textual terms. For the SuperIDR, instead of provides same features as EKEY, it enables the user interaction to add content, support for working with specific parts of images, performing content-based image annotation and retrieval and has pen-input capabilities which mimicking free-hand drawing and writing on paper. While in terms of database system, the relational database architecture was used for text annotation. Both systems are used in the Ichthyology domain. As an alternative approach to teach, compare and learn concept about parasites in general, research group in Kozievitch *et al.* (2010) adapted SuperIDR.

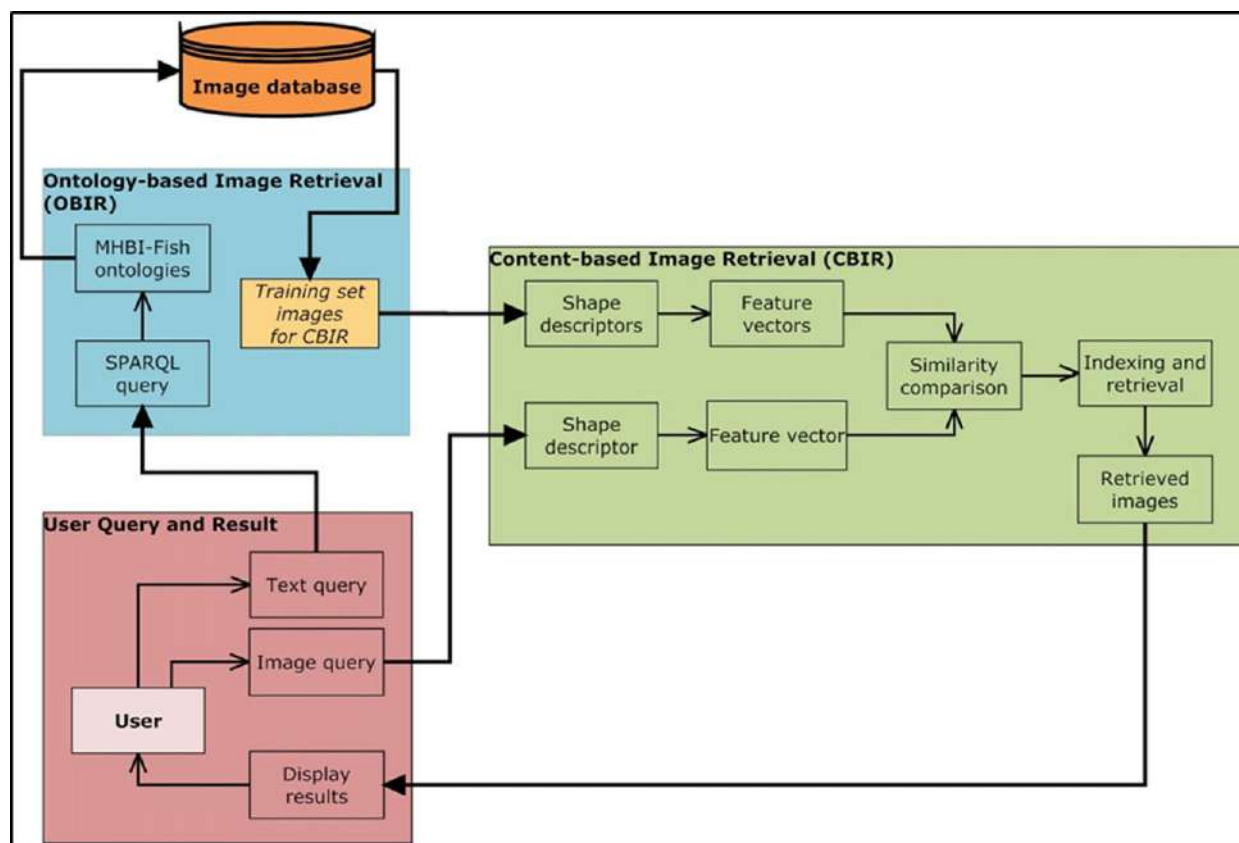


Fig. 18 The Model procedural flow.

To summarize, based on the review done, several main points are derived:-

- (1) Most of the image databases provide only text query option to retrieve the images from the database.
- (2) None of the existing systems uses ontology-based image annotation and retrieval to perform image pre-classification. The nearest are the EKEY and SuperIDR whereby each image is annotated with certain parameters such as species name. Thus, the user can customize their search according to these parameters to find the nearest match to the query image.
- (3) Biology data is heterogeneous, containing complex images and terminology to describe the data is always evolving overtime. Thus, graph data is suitable approach for text data modelling.
- (4) Both text- and content-based image retrieval approaches have their own advantages. Yet both approaches as well have a same limitation, which is, may retrieve irrelevant images.

Finally, in order to improve the efficiency of image retrieval in CBIR approach, one of the solutions to reduce the semantic gap limitation is by combining text-based image retrieval into CBIR. By using this approach, it will narrow down the most relevant images to be used for training set images. There are few aspects that are important to consider when developing the integrating text- and content-based image retrieval model, i.e., textual data representation, image annotations, query specification and the expected output of the retrieval process.

Suitable and proper vocabularies are needed in order to annotate the image because the images will be retrieved based on the vocabularies. The annotated data is required to represent in meaningful, dynamic and flexible manner so that any inclusion new vocabulary in future will be able to accommodate without changing the whole data structure. As for query specification, both textual and image query are needed. The last aspect is the output of the retrieval process, which is crucial in determining whether the retrieval process works well and in an efficient manner. Thus, to achieve this, the most relevance images must be retrieved with their annotation.

Proposed Solution for Bioimage Databases Using Ontology and CBIR Approaches

The details of the proposed solution in relation to the identified problems are presented here. The proposed solution aims to integrated both textual and image retrieval from image databases. The proposed image retrieval using ontology and CBIR approaches is shown in [Fig. 18](#). This architecture is proposed by [Arpah et al. \(2013\)](#).

In this model, the Ontology Based Information Retrieval (OBIR) layer determines the training set for Content Based Image Retrieval (CBIR). Ontology-based image retrieval is used as technique to reduce the training images for the CBIR layer by eliminating the irrelevant images using the text-based query in OBIR layer. This technique is also referred as data reduction usually used in data pre-processing to obtain a reduced representation of the dataset, which is smaller in quantity, yet closely maintains the integrity of the original data.

Conclusion

In this article, bioimage databases are presented in two classified categories which are text-based retrieval and image-based retrieval or more commonly known as content-based image retrieval. Based on the reviews presented in this article, we conclude that bioimage databases have not been able to completely harness the power of computer vision and image analytics in building the data archives. As such, a substantial amount of work in this field is required to build scientific databases, using ontologies for text queries and content based image retrieval methods for image queries. The methodology presented by [Arpah et al. \(2013\)](#) and the initiative started by [Williams et al. \(2017\)](#) are good example of research to be modeled in building bioimage databases. Finally, for the advancement of science, particularly biology, access to a wide range of image data is necessary to conduct transformative research.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Data Storage and Representation. Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics

References

- Andy, S., James, B., 2012, 2009. MonoDb homepage. Retrieved August 2011, from: <http://www.monodb.org/index.php>.
- Abu, A., Lim, L.H.S., Sidhu, A.S., Dhillon, S.K., 2013. Biodiversity image retrieval framework for monogeneans. *Systematics and Biodiversity* 11 (1), 19–33.
- Avril, S., 2005. Ontology-based image annotation and retrieval. Master of Science, University of Helsinki. Retrieved from: <http://www.cs.helsinki.fi/u/astyman/gradu.pdf>.
- Aye, K.N., Thein, N.L., 2012. Efficient indexing and searching framework for unstructured data. In: Zeng, Z.L.Y. (Ed.), *Proceedings of the Fourth International Conference on Machine Vision*, vol. 8349.
- Bienert, S., Waterhouse, A., de Beer, T.A., et al., 2017. The SWISS-MODEL repository - new features and functionality. *Nucleic Acids Research* 45 (D1), D313–D319.
- Chatzichristofis, S.A., Zagoris, K., Boutalis, Y.S., Papamarkos, N., 2010. Accurate image retrieval based on compact composite descriptors and relevance feedback information. *International Journal of Pattern Recognition and Artificial Intelligence* 24 (2), 207–244.
- Chen, Y., Henry, L., Bart, J., Teng, F., 2005. A content-based image retrieval system for fish taxonomy. In: *Paper Presented at the Proceedings of the 7th ACM SIGMM International Workshop on Multimedia information retrieval*, Hilton, Singapore.
- Demner-Fushman, D., Antani, S., Simpson, M., Thoma, G.R., 2009. Annotation and retrieval of clinically relevant images. *International Journal of Medical Informatics* 78 (12), e59–e67. doi:10.1016/j.ijmedinf.2009.05.003.
- Droissart, V., Simo, M., Sonké, B., Geerinck, D., Stévant, T., 2012. Orchidaceae of Central Africa. Retrieved August 2011, from: <http://www.orchid-africa.net/>.
- Duan, L., Gao, W., Zeng, W., Zhao, D., 2005. Adaptive relevance feedback based on Bayesian inference for image retrieval. *Signal Processing* 85 (2), 395–399. doi:10.1016/j.sigpro.2004.10.006.
- El-Naqa, I., Yongyi, Y., Galatsanos, N.P., Nishikawa, R.M., Wernick, M.N., 2004. A similarity learning approach to content-based image retrieval: Application to digital mammography. *IEEE Transactions on Medical Imaging* 23 (10), 1233–1244. doi:10.1109/tmi.2004.834601.
- EKEY. (2012). EKEY - The Electronic Key for Identifying Freshwater Fishes Retrieved July, 2009, from <http://digitalcorpora.org/corp/nps/files/govdocs1/054/054359.html>.
- Feng, D., Siu, W.C., Zhang, H.J., 2003. *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*. Springer.
- Gregorev, N., Huber, B., Shah, M.R.A.V., et al., 2003. Plant bug: Planetary biodiversity inventory, 2011, from: <http://www.monodb.org/index.php>.
- Hsu, W., Antani, S., Long, L.R., Neve, L., Thoma, G.R., 2009. SPIRS: A Web-based image retrieval system for large biomedical databases. *International Journal of Medical Informatics* 78 (Suppl. 1), S13–S24. doi:10.1016/j.ijmedinf.2008.09.006.
- imense, 2007. Imense Image Search Portal. Retrieved March 2010, from: <http://imense.com/>.
- Incogna, 2012. Incogna image search. Retrieved August 2011, from: <http://www.incogna.com/>.
- Kalvari, I., Argasinska, J., Quinones-Olvera, N., et al., 2017. Rfam 13.0: Shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Research*. doi:10.1093/nar/gkx1038.
- Mallik, J., Samal, A., Gardner, S.L., 2007. A content based pattern analysis system for a biological specimen collection. In: *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*. doi: 10.1109/ICDMW.2007.3.
- Kak, A., Pavlopoulou, C., 2002. Content-based image retrieval from large medical databases. In: *Paper Presented at Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*.
- Kozievitch, N.P., Torres, R.D., Andrade, F., et al., 2010. A teaching tool for parasitology: Enhancing learning with annotation and image retrieval. In: Lalmas, M., Jose, J., Rauber, A., Sebastiani, F., Frommholz, I. (Eds.), *Research and Advanced Technology for Digital Libraries 6273*. Berlin: Springer-Verlag Berlin, pp. 466–469.
- Krishnapuram, R., Medasani, S., Sung-Hwan, J., Young-Sik, C., Balasubramaniam, R., 2004. Content-based image retrieval based on a fuzzy approach. *IEEE Transactions on Knowledge and Data Engineering* 16 (10), 1185–1199. doi:10.1109/tkde.2004.53.
- Lamard, M., Cazuguel, G., Quellec, G., et al., 22–26 August 2007. Content based image retrieval based on wavelet transform coefficients distribution. In: *Paper Presented at the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007*.
- Lee, C.-H., Wang, S.-H., 2012. An information fusion approach to integrate image annotation and text mining methods for geographic knowledge discovery. *Expert Systems with Applications* 39 (10), 8954–8967. doi:10.1016/j.eswa.2012.02.028.
- Lin, W.-C., Chang, Y.-C., Chen, H.-H., 2007. Integrating textual and visual information for cross-language image retrieval: A trans-media dictionary approach. *Information Processing & Management* 43 (2), 488–502. doi:10.1016/j.ipm.2006.07.015.

- Liu, Y., Zhang, D., Lu, G., Ma, W.-Y., 2007. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40 (1), 262–282. doi:10.1016/j.patcog.2006.04.045.
- Liu, R., Wang, Y., Baba, T., Masumoto, D., Nagata, S., 2008. SVM-based active feedback in image retrieval using clustering and unlabeled data. *Pattern Recognition*, 41(8), 2645–2655. doi:10.1016/j.patcog.2008.01.023.
- McQuilton, P., Pierre, S.E.S., Thurmond, J., 2012. FlyBase 101 – The basics of navigating FlyBase. *Nucleic Acids Research* 40 (Database issue), D706–D714. doi:10.1093/nar/gkr1030.
- Müller, H., Michoux, N., Bandon, D., Geissbühler, A., 2004. A review of content-based image retrieval systems in medical applications – Clinical benefits and future directions. *International Journal of Medical Informatics* 73 (1), 1–23. doi:10.1016/j.ijmedinf.2003.11.024.
- Murthy, U., Fox, E.A., Chen, Y., *et al.*, 2009. Superimposed Image description and retrieval for fish species identification. In: Agnosti, M.B.J.K.S.P.C.T.G. (Ed.), *Proceedings of the Research and Advanced Technology for Digital Libraries*, vol. 5714, pp. 285–296.
- Oberlin, A.T., Jurkovic, D.A., Balish, M.F., Friedberg, I., 2013. Biological database of images and genomes: Tools for community annotations linking image and genomic information. *Database: The Journal of Biological Databases and Curation* 2013, bat016. doi:10.1093/database/bat016.
- Orloff, D.N., Iwasa, J.H., Martone, M.E., Ellisman, M.H., Kane, C.M., 2013. The cell: An image library-CCDB: A curated repository of microscopy data. *Nucleic Acids Research* 41 (Database issue), D1241–D1250. doi:10.1093/nar/gks1257.
- Ortega-Binderberger, M., Mehrotra, S., 2004. Relevance feedback techniques in the MARS image retrieval system. *Multimedia Systems* 9 (6), 535–547. doi:10.1007/s00530-003-0126-z.
- Petrov, A.I., Zirbel, C.L., & Leontis, N.B., 2013. Automated classification of RNA 3D motifs and the RNA 3D Motif Atlas. *RNA*, 19(10), 1327–1340. <http://doi.org/10.1261/rna.039438.113>.
- Rosa, N.A., Felipe, J.C., Traina, A.J.M., *et al.*, 20–25 August 2008. Using relevance feedback to reduce the semantic gap in content-based image retrieval of mammographic masses. In: Paper Presented at the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2008.
- Rui, Y., Huang, T.S., Chang, S.-F., 1999. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation* 10, 39–62.
- Scott, G., Chi-Ren, S., 2007. Knowledge-driven multidimensional indexing structure for biomedical media database retrieval. *IEEE Transactions on Information Technology in Biomedicine* 11 (3), 320–331. doi:10.1109/titb.2006.880551.
- Sergyan, S., 21–22 January 2008. Color histogram features based image classification in content-based image retrieval systems. In: Paper Presented at the 6th International Symposium on Applied Machine Intelligence and Informatics, SAMI 2008.
- Shandilya, S.K., Singhai, N., 2010. A survey on: Content based image retrieval systems. *International Journal of Computer Applications* 4 (2), 22–26.
- Sheikh, A.R., Lye, M.H., Mansor, S., Fauzi, M.F.A., Anuar, F.M., 14–17 June 2011. A content based image retrieval system for marine life images. In: Paper Presented at the IEEE 15th International Symposium on the Consumer Electronics, ISCE.
- Sillitoe, I., Lewis, T.E., Cuff, A.L., 2015. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research* 43 (Database issue), D376–D381. doi:10.1093/nar/gku947.
- Simon, R., Vince, S., 2011. SID: Specimen image database. Retrieved August 2011, from: <http://sid.zoology.gla.ac.uk/>.
- Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (12), 1349–1380. doi:10.1109/34.895972.
- Stejić, Z., Takama, Y., Hirota, K., 2003. Genetic algorithm-based relevance feedback for image retrieval using local similarity patterns. *Information Processing and Management* 39 (1), 1–23. doi:10.1016/s0306-4573(02)00024-9.
- Torres, R.D.S., Falcao, A.X., 2006. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada* 13, 161–185.
- Torres, R.D., Medeiros, C.B., Dividino, R.Q., *et al.*, 2004. Using digital library components for biodiversity systems.
- Ursula, P., Webb, B.M., Qiang Dong, G., *et al.*, 2014. MODBASE, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Research* 42, D336–D346.
- Wang, J., Ji, L., Liang, A., Yuan, D., 2012. The identification of butterfly families using content-based image retrieval. *Biosystems Engineering* 111 (1), 24–32. doi:10.1016/j.biosystemseng.2011.10.003.
- Wang, D., Ma, X., 2005. A hybrid image retrieval system with user's relevance feedback using neurocomputing. *Informatica* 29 (3), 271–280.
- Wei, J., Guihua, E., Qionghai, D., Jinwei, G., 2006. Similarity-based online feature selection in content-based image retrieval. *IEEE Transactions on Image Processing* 15 (3), 702–712. doi:10.1109/tip.2005.863105.
- Wei, C.-H., Li, C.-T., 2006. Calcification descriptor and relevance feedback learning algorithms for content-based mammogram retrieval. In: Astley, S., Brady, M., Rose, C., Zwiggelaar, R., (Eds.), *Proceedings of International Workshop on Digital Mammography Digital Mammography*, vol. 4046, pp. 307–314. Heidelberg: Springer Berlin.
- Williams, E., Moore, J., Li, S.W., *et al.*, 2017. Image data resource: A bioimage data integration and publication platform. *Nature Methods* 14, 775. doi:10.1038/nmeth.4326.
- Xin, J., Jin, J.S., 2004. Relevance feedback for content-based image retrieval using Bayesian network. In: Paper Presented at the Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing. Available at: <http://dl.acm.org/citation.cfm?id=1082137>.
- You, D., Antani, S., Demner-Fushman, D., *et al.*, 2011. Automatic identification of ROI in figure images toward improving hybrid (text and image) biomedical document retrieval. In: Agam, G., ViardGaudin, C. (Eds.), *Document Recognition and Retrieval XVIII* 7874. Bellingham: Spie-Int Soc Optical Engineering.
- Zhang, H., Chen, Z., Li, M., Su, Z., 2003. Relevance feedback and learning in content-based image search. *World Wide Web* 6 (2), 131–155. doi:10.1023/a:1023618504691.
- Zhang, X.M., Huang, Z., Shen, H.T., Li, Z.J., 2011. Probabilistic image tagging with tags expanded by text-based search. In: Yu, J.X., Kim, M.H., Unland, R. (Eds.), *Database Systems for Advanced Applications, Pt I* 6587. Berlin: Springer-Verlag Berlin, pp. 269–283.

Relevant Websites

<http://www.monodb.org/index.php>

A web-host for the monogenea.

<http://biosilico.kaist.ac.kr/>

Biosilico: An Integrated Metabolic Database System.

http://www.cathdb.info/search/by_structure

CATH Search.

<http://rfam.xfam.org/>

EMBL-EBI.

<http://sid.zoology.gla.ac.uk/>

Florida State University, Tallahassee, USA.

<http://flybase.org/>

FlyBase Homepage.

<http://biodig.org>
 Friedberg Lab, Miami University, Oxford OH, USA.
<http://images.google.com/>
 Google Images.
<http://research.amnh.org/pbi/heteropteraspeciespage/>
 Heteroptera Species Pages - Our Research.
<https://idr.openmicroscopy.org>
 Image Data Resource: IDR.
<https://modbase.compbio.ucsf.edu>
 ModBase Search Page.
<http://www.macroglossa.com/>
 MVE Photography.
<http://www.orchid-africa.net/>
 Orchids of Central Africa.
<https://research.amnh.org/pbi/>
 Plant Bug :: Planetary Biodiversity Inventory.
<http://rna.bgsu.edu/rna3dhub/motifs>
 RNA 3D Motif Atlas - BGSU RNA Structural Bioinformatics.
<https://swissmodel.expasy.org/repository>
 SWISS-MODEL - Repository.
<http://www.boldsystems.org/index.php>
 SWISS-MODEL.
<http://www.cellimagelibrary.org>
 The Cell Image Library.

Segmentation Techniques for Bioimages

Saowaluck Kaewkamnerd and Apichart Intarapanich, National Electronics and Computer Technology Center, Pathum Thani, Thailand

Sissades Tongsim, National Center for Genetic Engineering and Biotechnology, Pathum Thani, Thailand

© 2019 Elsevier Inc. All rights reserved.

Introduction

Bioimage is a combination of the two words “Biology” and “image”. Biology means the study of life and living organisms, including their physical and chemical structure, function, development and evolution ([Murphy and Davidson, 2001](#)) and thus, Bioimages broadly imply biologically related images. These images can be taken from large varieties of organisms of any size, ranging from whole organisms down to a single molecule level. There are many techniques to obtain these images, that can be categorized according to sizes into three levels, namely molecular, microscopic and whole organism levels. First, images from the molecular level include scanning or transmission electron microscopy ([Tsien, 2003](#)). For the microscopic level, Bioimages include those from confocal or two-photon laser scanning microscopy ([Tsien, 2003](#)). The last category is composed of those Bioimages of parts or organs of organisms, obtained from X-ray, magnetic resonance, ultrasound, endoscopy, thermography, positron emission tomography (PET), Single-photon emission computed tomography (SPECT) and inferred ([Jia et al., 2016](#)) and others. These Bioimages require domain expert visual instruction to interpret them. However, these manual interpretation steps are very tedious, error prone and time consuming. To overcome the limitation, the use of computer-aided systems such as image processing techniques, has been introduced. Thus, components/objects from Bioimages must be automatically extracted and converted into a format that any image-analysis algorithm can efficiently processed.

Bioimage Segmentation: Definition and Challenges?

Humans understand and identify specific parts of an image because of our recognition through memorization and retrieval of said information. To be able to do this using computer, automatic identification of useful components in an image is needed. Image segmentation is a process to partition an image into objects, which can be either meaningful or non-meaningful and therefore, it is a vital process to identify objects of interest. Bioimage segmentation plays an important role in many branches of life sciences such as, biotechnology, precision agricultures, pharmacology, advanced medicine and microbiology. These domains require large number of images to be processed. For more efficiency and automation, computer-based systems are required to identify objects or region of interest before passing them to other stages of image processing, such as feature extraction, image classification and image recognition ([Gonzalez and Woods, 2008](#)). Even though research on image segmentation has been conducted for more than 40 years ([Zhang, 2009](#)), there are no single solution for all kinds of images. Such a challenge majorly stems from a wide spectrum of image qualities. For example, in microscopic image analysis, the difficulties in segmentation are from inconsistent stained color, over-exposed or under-exposed to microscope light, poor contrast, varieties of cell types, cell overlapping and unwanted objects (artifacts) in the image ([Win and Choomchuay, 2017](#)). Another example of confounding signals is found in an analysis of intravascular ultrasound images whose poor qualities come from the presence of calcifications, shadows, transducer reflection, speckle noises and bifurcations and intensity-variant of structure ([Jodas et al., 2003](#)). Customized segmentation algorithms are yet to be devised for some specific problems ([Gupta et al., 2017](#)).

Segmentation Techniques: Review

In image analysis, the segmentation step is carried out after some pre-processing processes such as noise reduction, color conversion and enhancement processes to suppress unwilling distortions, transform data or enhance features for further processing. Segmentation techniques can be further classified into seven classes based on their methodologies ([Fig. 1](#)), including thresholding segmentation, region-based segmentation, edge-based segmentation, contour-based segmentation, clustering-based segmentation, recognition-based segmentation and computational intelligence-based segmentation ([Patil and Deore, 2013](#); [Kaur and Kaur, 2014](#); [Anjna and Kaur, 2017](#)). The underlying theoretical foundations and limitations of the methods in these classes are discussed in the following subsections.

Thresholding Segmentation

Segmentation algorithms from this class share a common scheme that is using a pixel intensity as a threshold to partition an image into two partitions. For any pixel p located at a coordinate (x,y) , the pixel p is assigned to one partition if its intensity I , at the coordinate (x,y) is greater than or equal to the threshold value Th , i.e., $I(x,y) \geq Th$. Otherwise the pixel is assigned to the other partition ([Kaur and Kaur, 2014](#)). The performance of this kind of thresholding protocol relies on its threshold value, which can be determined manually or automatically.

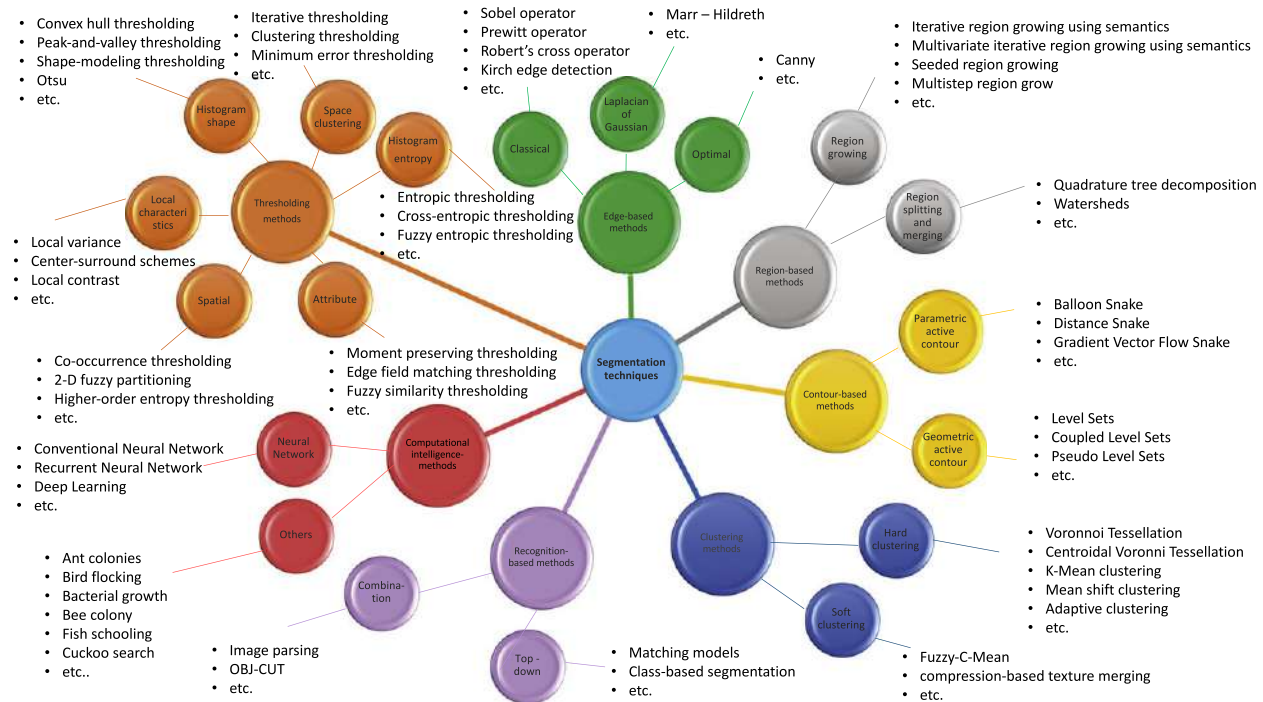


Fig. 1 Seven main classes of segmentation methods, that are classified based upon their core techniques used to partition input images: (1) thresholding, (2) region-based, (3) edge-based, (4) contour-based, (5) clustering-based, (6) recognition-based, and (7) computational intelligence-based classes.

The automatic thresholding techniques may be further divided into six sub-categories based on distinguishable characteristic at the pixel level such as histogram shape information, measurement space clustering, histogram entropy information, image attribute information, spatial information and local characteristics (Zhang, 2001). The first category uses the information of histogram shape such as peaks, valleys and curvature for thresholding. Examples of shape thresholding are Otsu, Convex hull thresholding, Peak-and-valley thresholding, Shape-modeling thresholding and others (Sezgin and Sankur, 2004). The main idea of the Otsu method, named after the inventor, Otsu (1979) is to find an optimum threshold for separating an image into two classes (object and background). This Otsu technique operates by iterating over all possible threshold values and selects the threshold that minimizes their intra-class variances or maximizes their inter-class variances. For the second sub-category, measurement of space clustering, the number of clusters is always set to two, namely the object cluster and the background cluster. The thresholding value, also called the discriminator, is used to create a mask (0 or 1 corresponding to each pixel from the original image) that can separate objects from the background. Examples of the measurement of space clustering category include Iterative thresholding, Clustering thresholding, Minimum error thresholding (Sezgin and Sankur, 2004). The entropy-based thresholding sub-category utilizes the concept of the average level of information, known as the entropy as a criteria to look for the most suitable thresholding value that best preserves the information containing in the original image. A method, called maximum entropy thresholding, was introduced by Kapur *et al.* (1985). In their approach, the threshold is selected if the sum of entropies of the object and the background reaches the maximum value. Another example of entropy-based is the cross-entropy technique (Kapur *et al.*, 1985) when the threshold value is selected by minimizing the information difference between the input and output images. Object attribute-based segmentation algorithms commonly employ similarity of attributes such as shape compactness, gray-level moments and texture between input and output images (Tsai, 1985; Pal and Rosenfeld, 1988). Thresholding methods, belonging to the spatial sub-category, uses higher-order probability distribution of gray value of pixel and its neighborhood such as context probabilities, correlation functions and co-occurrence probabilities (Cheng and Chen, 1999). Finally, those local characteristic segmentation algorithms make use of local statistics like range and variance of range and variance of a pixel and those from other pixels around it. More examples of the segmentation techniques based on thresholding are presented in Sezgin and Sankur (2004).

The above thresholding-based image segmentation techniques are computationally fast and very easy to implement. They are commonly adopted and implemented by many real-time applications (Dass *et al.*, 2012; Anjna and Kaur, 2017; Patil and Deore, 2013). However, these simple segmentation techniques are not good when processing images that have uneven illumination background with broad and flat valleys. Using a threshold to segment poor images with high degree of noise interferences results in poor segmentation outputs. Furthermore, these algorithms do not offer spatial characteristic of image. To demonstrate, how thresholding-based segmentation performs, Fiji (Schindelin *et al.*, 2012), which is a nice image processing software that bundles all popular image processing algorithms in ImageJ (Schneider *et al.*, 2012) was used. To segment an MRI image using thresholding

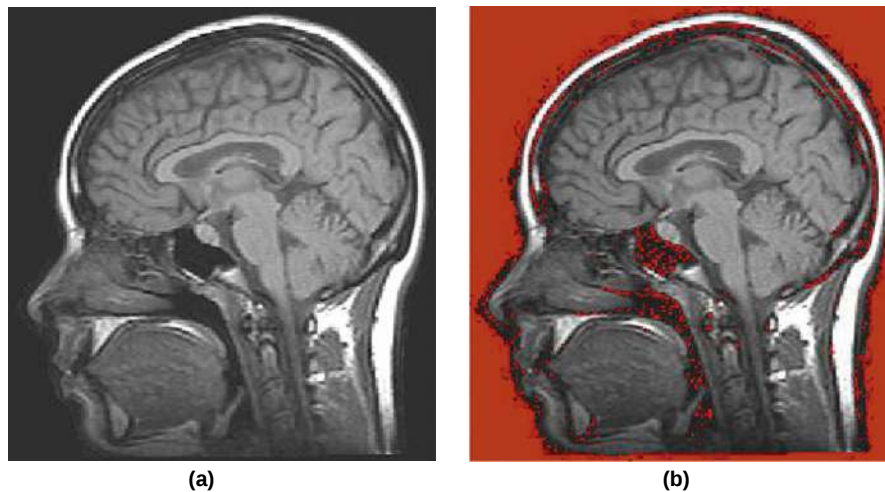


Fig. 2 An example of a brain Magnetic Resonance Imaging (MRI) image. (a) The original MRI scanned image and (b) the segmented image with a threshold set to 30, where regions highlighted in red represent detected background.

(Fig. 2), choose Fiji: image → adjust → threshold from the main menu. A pop-up window displaying the histogram profile of the selected image will be shown where the user can adjust the thresholding parameter using a slide bar.

Edge-Based Segmentation

Edge-based segmentation methods detect “discontinuity” properties such as gray change, color distinctness and texture variety that can demarcate between different regions. Particularly, edges and region boundaries are closely related because lines around an object create region boundaries. This kind of edge detection scheme is commonly used as a pre-processing step to enhance other segmentation techniques. There are different types of edges that are considered in algorithms from this edge-based segmentation class, namely ramp edge (gradually change in intensity), step edge (an abrupt change in intensity), roof edge (not instantaneous over a short distance) and spike edge (a quick change in intensity values) (Anjna and Kaur, 2017). Krishnan *et al.* (2017) classified edge-based segmentation according to three operators used to detect the changes: classical operator, Laplacian of Gaussian operator and optimal operator. A classical operator is the first order derivation with examples such as Sobel operator (Gonzalez and Woods, 2008), Prewitt operator (Prewitt, 1970), Robert’s cross operator (Bansal *et al.*, 2012), Kirsch edge detection (Kirsch, 1971) etc. Laplacian of Gaussian operator uses to find the second derivation. Finally the optimal operator uses a multi-stage algorithm to detect a wide range of edge information, e.g., Canny edge detection (Canny, 1986). Fig. 3 shows a resulting segmented fingerprint image (right panel) using ImageJ by choosing ImageJ: Plugins → Canny Edge Detector from ImageJ menu.

Edge-based methods are simple to implement and yield good results in most cases. However, for those images which contain too much noises too many traces, resulting edges identified by an edge detection method are usually discontinued and very fragmented. Therefore an edge linking algorithm to join these fragments and construct separable boundaries is needed.

Region-Based Segmentation

Region-based segmentation methods partition an image using a concept of region similarity which must be predefined. To form a region, a picture element (pixel) that is an elementary building block for bitmap (raster) images (Kaur and Kaur, 2014) is compared to the neighboring pixels. If the similarity criteria is met, these pixels will incrementally be formed as a region. These region-based segmentation methods can be categorized into two groups, (1) region growing group and (2) region splitting and merging group (Kaur and Kaur, 2014). The region growing group incrementally expands a boundary of a region by merging those similar pixels together. Examples of algorithms from the region growing group include Seeded Region Growing, Theoretical Criteria, and Multistep Region Growing (Vantaram and Saber, 2012). A recent review by Adamu and Ding (2015) further classifies these region growing segmentation techniques into Seeded Region Growing (SRG) and Unseeded Region Growing (UsRG). For SRG, the growing process defines seed points or seed pixels whose properties can be used as inclusion criteria for expanding a region. The neighboring pixels that satisfy the inclusion criteria join the existing seed pixels. This inclusion selection is done iteratively until all pixels are examined. Clearly, segmentation results from the SRG algorithms are very much subjective and dependent upon the selected seeds. Unlike the SRG protocol, UsRG techniques do not require preselected explicit seeds. UsRG algorithms arbitrarily define one single region covering some pixels that will be used to grow the region boundary iteratively by recruiting more neighboring pixels. When encountering a pixel whose property greatly differs from the current growing region, that pixel will start a new region, which in turn start growing accordingly.

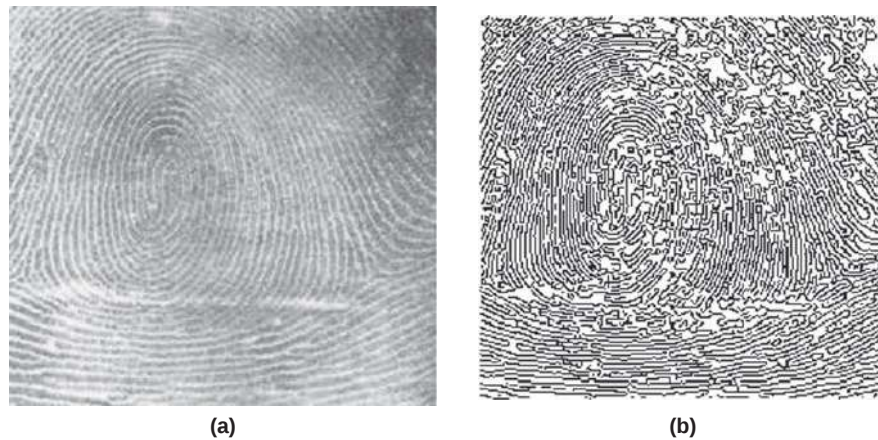


Fig. 3 A fingerprint image contains mostly traces of fingerprints presenting a challenge in segmentation based on classical edge-detection since detected edges may not be connected to form good boundaries for partitioning. (a) The original fingerprint image and (b) The segmented image using Canny edge detection, where contour traces show potential edges extracted from the original fingerprint.

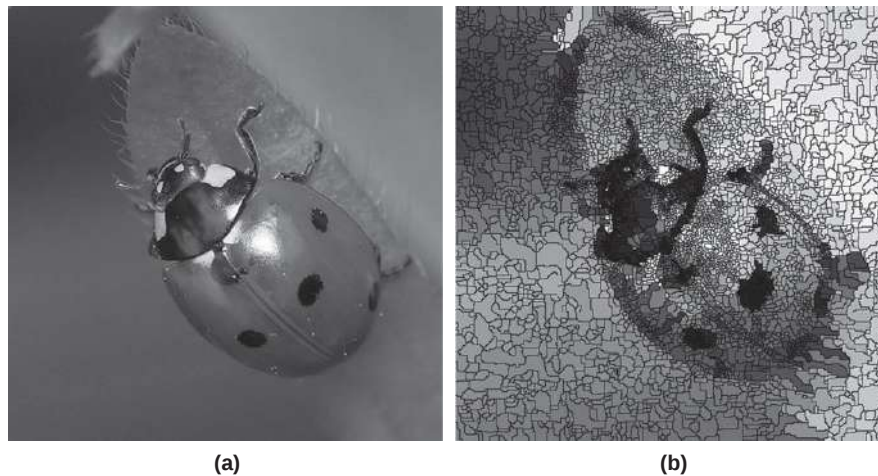


Fig. 4 Using Watershed to partition a clear region boundary image of a ladybug produces over-segmented result: (a) The original image of a ladybug and (b) the over-segmented result after using Watershed.

For the region splitting and merging class, algorithms in this class use divide-and-conquer to recursively partition (split) the whole image and then quickly merge those small yet similar regions to create bigger regions. In (Gorte, 1996), the quadtree segmentation algorithm was used by setting the whole image as a root of the quadtree. The splitting commences if all pixels in this whole image are not homogeneous. The splitting process create four child squares, each of which is carried out recursively using the aforementioned splitting procedure. Splitting terminate if all the pixels in the region are homogeneous (being sufficiently similar according to the criterion) or having only one pixel in it. The merging of similar regions grows from the leaves of this quadtree toward the root. As an example, Watershed is an efficient region-based splitting method (Courprie and Bertrand, 1997). The term “Watershed” refers to a geological watershed, which separates adjacent drainage basins. An input image is treated like a topographic map, whose pixels’ brightness represent the heights of ridges. Examples of Watershed-based algorithms used to partition regions are watershed by flooding, watershed by topographic distance, watershed by the drop of water principle, inter-pixel watershed and topological watershed (Courprie and Bertrand, 1997).

Advantages of region-based methods are having high robustness, preserving the boundary of the segmented regions, producing coherent regions (regions with high degree of pixel uniformity). Nevertheless, objects with multiple disconnected regions present greater challenges, requiring human intervention for selecting input seed pixels. Furthermore, the resulting segmentations by the region-based methods are highly dependent on the choice of input seeds for seeded region growing. Watershed-based methods are likely to produce over-segmented results. As an example, Fig. 4 presents a ladybug image with Watershed-based segmentation algorithm, provided in Fiji to partition the image. To operate this, from Fiji main menu choose: Fiji: Plugins → MorphoLibJ → Segmentation → Classic Watershed. The resulting image displays too many regions that Watershed attempted to segment.

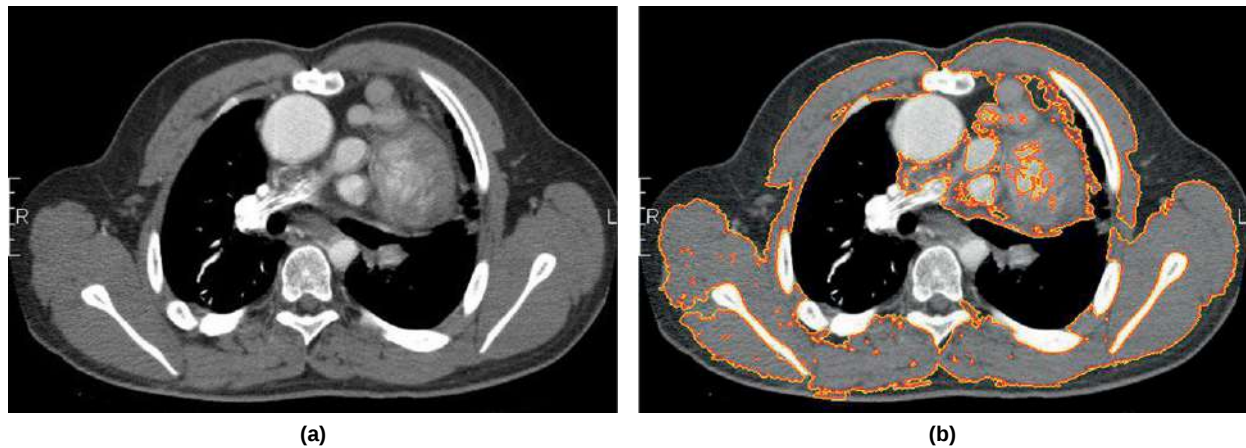


Fig. 5 A CT scan image does not have explicit edge to be detected (a). Using contour-based segmentation, important objects in this image can be detected as shown in the right panel (b).

Contour-Based Segmentation

A contour-based method is used for tracking boundaries (boundary tracing) in an image. First, a closed region (closed contour line) is designated as the starting boundary. This starting boundary is iteratively modified by either shrink/expansion operations; thus this changing boundary is also called an active contour. There are two types of active contour models, (1) the parametric active contour model such as Balloon Snake, Distance Snake and Gradient Vector Flow Snake and (2) the geometric active contour model such as level sets, coupled level-set, pseudo level-set etc. (Baswaraj *et al.*, 2012; Vantaram and Saber, 2012). Particularly, Snake (Kass *et al.*, 1988) moves a contour based on an energy minimization while level-set (Osher and Sethian, 1988) moves a contour according to a level of a function. A contour changes its location and shape based on “a predefined force function” that pulls the contours toward features of interest. In Snake, a set of snake points must be selected from an image. Those snake points will be moved to the next location by determining the local minimum energy until the points stop at the object boundary. In Level set, the algorithm partitions several objects in an image by implicit curve propagation. The curve is moved as the evolution of the level set function. This method helps estimate the geometric properties of the evolution structure (Bhaidasna and Mehta, 2013). Dynamic contour concept provides a clever way to detect objects from the input image with continuous boundaries, stability and irrelevancy with topology. A good segmentation relies on the initial locations (snake points) and having only one object in the image. Fig. 5 presents a CT scan image that does not have clear boundaries. By applying level-set from Fiji by choosing: Plugins → Segmentation → Level Sets. Significant objects can be detected by means of contour-based segmentation.

Clustering-Based Segmentation

Clustering is a method to group data into clusters based on their similarities (Dehariya *et al.*, 2010). The idea of clustering-based segmentation methods is to segment an image by assigning pixels with similar characteristics to a k number of groups (clusters) when k is given (Sharma and Suji, 2016). A membership function is used to define if a pixel belongs to the cluster. Such a function is described using a membership matrix with numbers between 0 and 1 representing a degree of membership of a pixel to a cluster. The pixel assignment continues until the maximum separation between clusters exceeds a given threshold.

There are two main types of clustering, (1) hard clustering such as Voronoi Tessellation, Centroidal Voronoi Tessellation, K-Mean clustering, Mean shift clustering, and Adaptive clustering (Vantaram and Saber, 2012) and (2) soft clustering such as Fuzzy-C-Mean (Kaur and Kaur, 2014). Hard clustering partitions an image into a given number of clusters with a restricted constraint that only one pixel belongs to one cluster. In this case, the hard clustering membership functions with the entry value of either 1 or 0 representing a membership status (being a member or not). K-Means clustering is a well-known hard clustering technique that first estimates a center for each of the K clusters, then each pixel will be assigned to those K clusters based on a similarity criterion such as distance, connectivity, intensity between pixel and cluster. The pixel assignment process attempts to minimize the inter-cluster similarity (between different clusters) and maximizing the intra-cluster similarity (within the group). Unlike hard clustering, soft clustering makes use of a flexible membership function and allows a pixel to be assigned to more than one cluster with varying degrees of membership ranging from 0 to 1. This type of flexibility is more natural than that of hard clustering. The Fuzzy-C-Mean algorithm (FCM) belongs to the soft clustering class (Christ and Parvathi, 2011). FCM is more flexible but its computational time could be expensive. The computational time of FCM can be dramatically reduced by means of using an intensity histogram of an image instead of accessing at the pixel level (Kaur and Kaur, 2014).

In summary, these clustering-based segmentation algorithms are quite robust to use in most images. However, all clustering-based segmentation methods require a predefined number of clusters. Most clustering techniques are largely sensitive to noises and other image artifacts that could interfere with the clustering assignment. Fig. 6 presents a segmentation result of a flower image

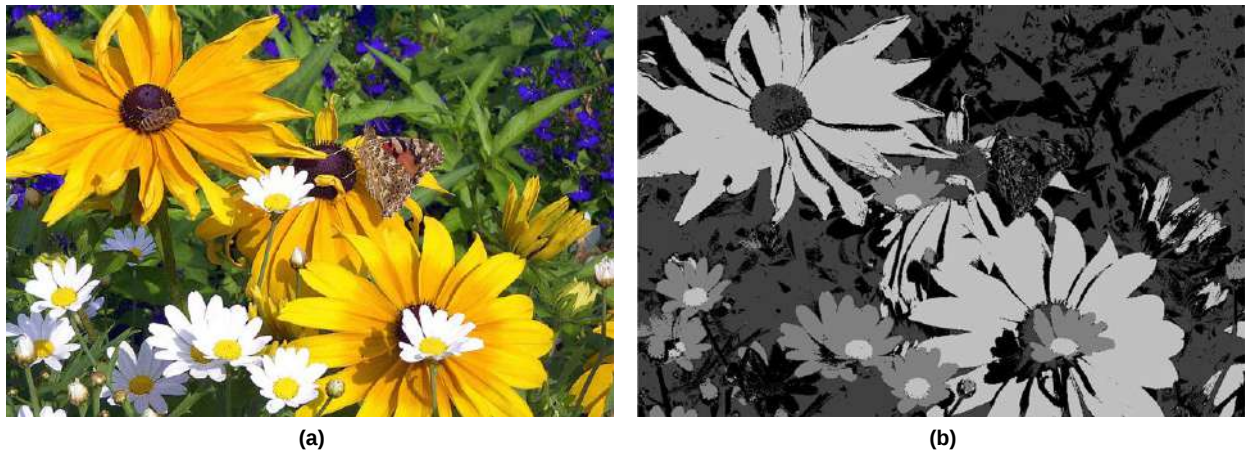


Fig. 6 K-mean clustering-based segmentation algorithm was used to detect different flowers. (a) A flower image with big (yellow) and small (white) flowers with leaves among them (b) the segmented image where both type of flowers are clearly separated.

that use K-mean clustering when $K=4$ to identify these flowers. The analysis was performed using ImageJ: Plugins → Segmentation → k-means Clustering.

Recognition-Based Segmentation

The aforementioned image segmentation techniques segment an image by grouping pixels that have something in common, like common colors, textures, intensities etc. Once the image is segmented using these techniques, a whole object or region may be oversegmented. Thus these segmented regions must be combined so that true objects can finally be extracted. The process of using fundamental image segmentation to identify uniquely primitive regions and then merge them to form the objects and background is called the bottom-up method. Bottom-up is quite useful for most images and generally produce good segmentation results; however, there are some complex images containing objects that cannot be well partitioned/detected by the former segmentation algorithms. For example, a segmentation algorithm may partition background from real object(s) but toward the final step, similar colors/textures of background and the objects may be merged. This happens because of the lack of meaningful information about the objects of interest. Human utilizes knowledge about each object in order to differentiate an object from the background or other objects. Top-down image segmentation (Borenstein and Ullman, 2002) imitates human recognition process by including a matching model to do the job by utilizing some characteristics of objects with known shapes to guide its segmentation process.

Another variant of image segmentation algorithm is the combination of both bottom-up and top-down approach (Ferrari *et al.*, 2006; Wang *et al.*, 2007; Levin and Weiss, 2009). The approach by Wang *et al.* presents an algorithm that combines both top-down recognition and bottom-up image segmentation. It consists of two steps, (1) hypothesis generation and (2) verification. In the first step, a set of object locations and figure-ground masks, called hypotheses, are constructed. Then the verification step uses these object hypotheses to compute a set of feasible segmented objects where the false positive objects are subject to be eliminated using False Positive Pruning (FPP) (Wang *et al.*, 2007). Advantages of these recognition-based segmentation algorithms include the unique ability to handle images with extensive clutter of objects, dominant occlusion, and large scale and viewpoint changes (Levin and Weiss, 2009). In particular, these algorithms can be used to track an object in an image so that no matter how the original image rotates or even shifted, the object will always be recognized. Fig. 7 shows how a red flower with a specific shape and color is tracked using a recognition-based segmentation algorithm from Fiji. From the main menu of Fiji: Plugins → Feature Extraction → Extract SIFT Correspondences.

Computational Intelligence-Based Segmentation

Human vision can efficiently recognize or identify an object from its background even if the object is not clearly or completely standing out. Clearly, gathering and using primitive information from an image to help partition is not enough. To improve this computational processing, researchers employ computational intelligence including Neural Network and other nature-inspired algorithms. Examples of Neural Network are Conventional Neural Network, Concurrent Neural Network, Deep Learning (Zheng *et al.*, 2015). The Deep Neural Networks was adopted to help detect brain tumor of various shapes, sizes and contrasts from magnetic resonance images (Havaei *et al.*, 2017). The adoption of Deep Neural Network in image segmentation creates a novel Convolutional Neural Networks (CNN) architecture that concurrently utilizes both local and global contextual features. Nature-inspired algorithms efficiently identify the best solution from a large number of finite solutions by mimicing how nature solves a combinatorial optimization problem and comes up with a good solution. Examples of these nature-inspired optimization algorithms include swarm intelligence, ant colony optimization (ACO), bird flocking, bacterial growth, bee colony, fish schooling, cuckoo search and others (Passino, 2002; Sharma *et al.*, 2015). A recent nature-inspired segmentation algorithm is the combination of ACO and genetic algorithm (Rahebi *et al.*, 2010). This work by Rahebi *et al.* is a probabilistic technique mimicking ant foraging behavior to choose an



Fig. 7 An image with many red flowers scattered and buried in the background. To track a particular object (a flower in a small window in the left panel), a recognition-based segmentation technique is used to perform the task.

optimal path by following ant pheromone to bring food back to their nest quickly. ACO segmentation models an image as a graph where nodes represent pixels. The algorithm generates artificial ants to store pixel intensities and construct a pheromone matrix that keeps track of paths (graph edges) that ants often visit. Nature-inspired algorithms can detect meaningful objects that may be incomplete or partly overlapped with other background. However, these algorithms can be very time consuming as learning stage is required to provide sufficient intelligence to these algorithms. [Fig. 8](#) presents an image containing different fishes. Using “Trainable Weka Segmentation” in Fiji: Plugins → Segmentation → Trainable Weka Segmentation, fishes can be detected.

Segmentation Techniques: Case study

In order to demonstrate the basic idea of each technique, a Giemsa-stained blood smear image with varying image conditions, namely in-focus, with noise and with uneven illumination, are used ([Fig. 9](#)). This blood smear image ([Fig. 9\(a\)](#)) was obtained from [Kaewkamnerd et al. \(2012\)](#) containing white blood cells (big black dots) and malaria parasites (small dots). Random noises, dots whose sizes are smaller than those of malaria parasite in the original image, were added to the original image ([Fig. 9\(b\)](#)). Finally the illumination scale is adjusted so that the image has illumination gradient ([Fig. 9\(c\)](#)). These three images are used as inputs for different segmentation algorithms. The image processing tool, named Fiji, is used for comparison purposes.

Thresholding-Based Method

Thresholding segmentation in Fiji was used to process the above three images in [Fig. 9](#). Thresholding is able to segment all of the white blood cells (marked by blue region around each cell) as well as most of malaria parasites ([Fig. 10\(a\)](#)). With the present of noises, the segmentation result ([Fig. 10\(b\)](#)) reveals that only few malaria parasites are identified; random noises clearly interfere with thresholding segmentation. When illumination is uneven, thresholding segmentation could neither detect both white blood cells nor malaria parasites in the area affected with dark illumination. Particularly, the whole area with dark illumination was partitioned into one big object region (blue shaded area in [Fig. 10\(c\)](#)).

Edge-Based Method

For edge-based segmentation, the Canny edge-based method in Fiji was used. Similar to thresholding, Canny was able to segment the original in-focus blood smear image ([Fig. 11\(a\)](#)). For the noisy image, Canny identified more objects ([Fig. 11\(b\)](#)), via its edge detection scheme, than that from in-focus original image ([Fig. 11\(a\)](#)) as well as from the uneven illumination image ([Fig. 11\(c\)](#)). Noises introduce discontinuities of pixel intensity that provide suitable environment for edge-based segmentation to make a decision. However, most objects detected by Canny are not connected, causing difficulties to further post-process this image.

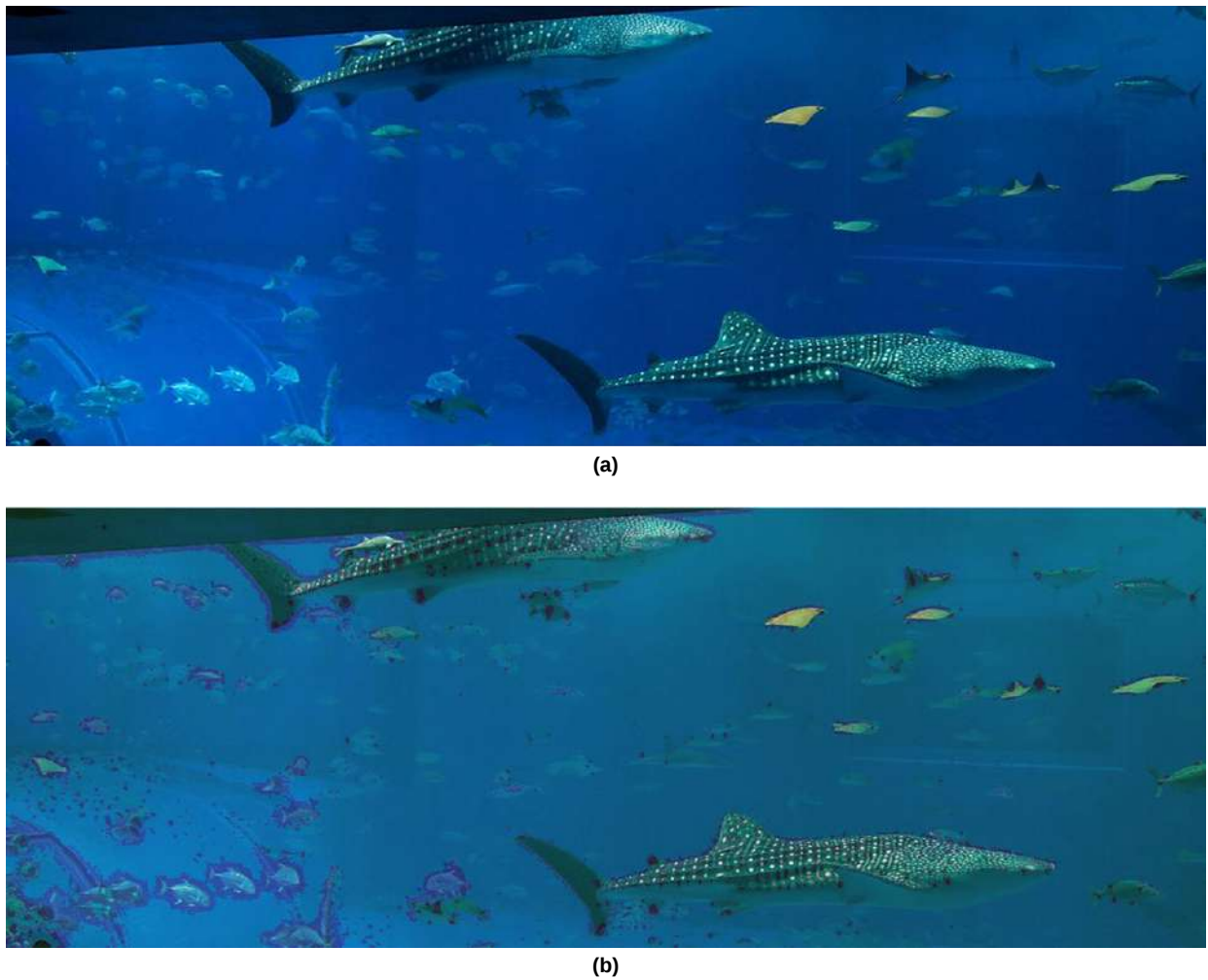


Fig. 8 An aquarium image has different kinds of fishes that may blend in with the blue background (a), presenting a challenge in image segmentation. Using a computational intelligence, fish objects can be detected (b).

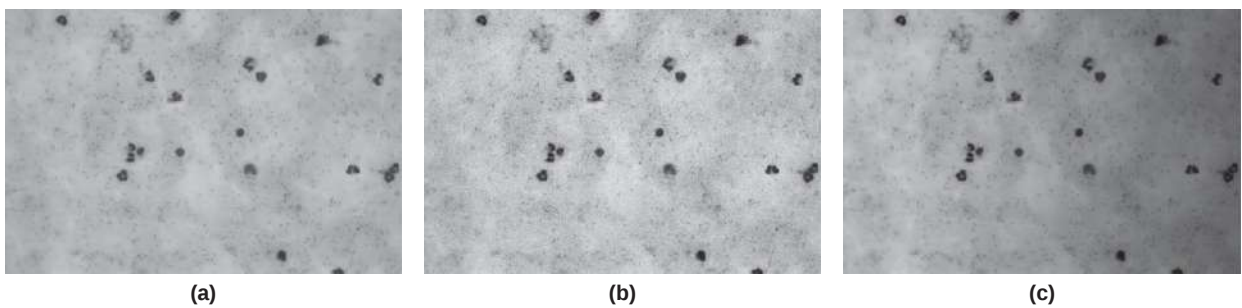


Fig. 9 Test Images (a) In-focus image. (b) Image with noises. (c) Image with uneven illumination.

Region-Based Method

The Watershed region-based method in Fiji was used to perform segmentation on the blood smear images in Fig. 9. However, Watershed over-segmented all of the three input images. Fig. 12(a) shows that the segmentation of white blood cells includes some of the malaria parasites around them. The light gray area of the background was clearly over-segmented. Such over-segmented effects appear more in the noisy image (Fig. 12(b)) which might have been influenced by those small noisy dots added to the original image. Fig. 12(c) also reveal the over-segmented result on uneven illumination blood smear image. Watershed is still over sensitive to changes of pixel illumination.

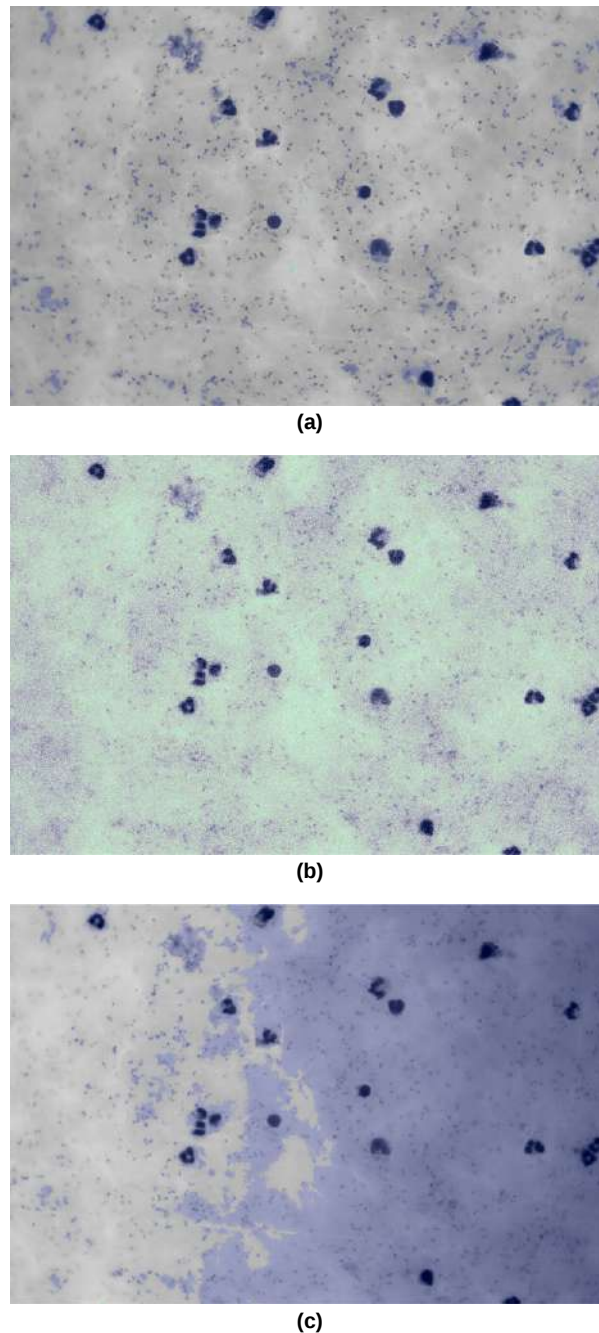


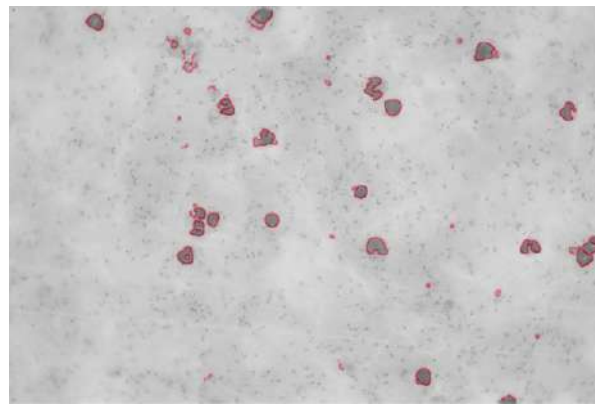
Fig. 10 Result images using thresholding method (a) In-focus image. (b) Image with noise. (c) Image with uneven illumination.

Contour-Based Method

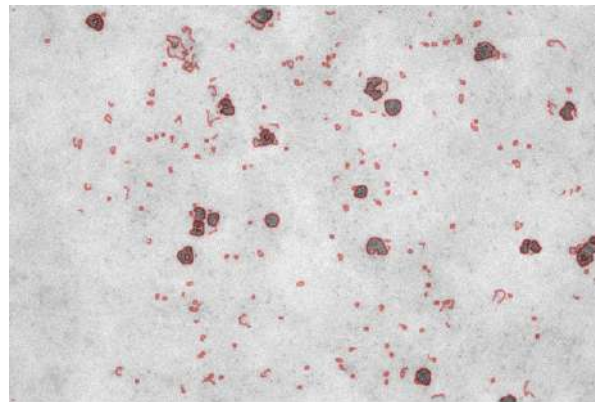
In this experiment, the Level-set contour-based segmentation was used. To demonstrate the region growing in Level-set, we assigned the initial seeds to the centers of those white blood cells in three images. Level set was able to correctly detect boundaries of all white blood cells from the original blood smear image ([Fig. 13\(a\)](#)). However, those noisy dots affect the performance of the region growing process ([Fig. 13\(b\)](#)) where fewer white blood cells were correctly traced. Uneven illumination did not interfere with the object detection ([Fig. 13\(c\)](#)).

Clustering-Based Method

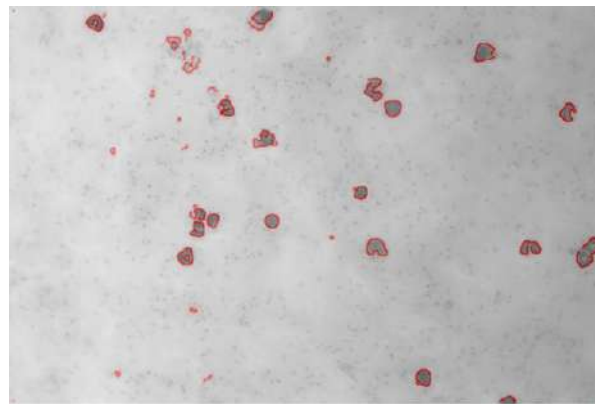
K-Mean clustering in ImageJ was used in this experiment. K-Mean clustering identifies regions in the in-focus image into two clusters ($K=2$), namely white blood cells and malaria parasites. All white blood cells are identified while only some of malaria



(a)



(b)



(c)

Fig. 11 Result images using Canny edge-based method (a) In-focus image. (b) Image with noise.(c) Image with uneven illumination.

parasites are detected ([Fig. 14\(a\)](#)). Random noises caused K-Mean clustering to be erratic and not able to properly segment the image. Only white blood cells were detected while a few of parasites were marked ([Fig. 14\(b\)](#)). Uneven illumination poses more problem to K-Mean clustering where the darker illumination was detected as one big cluster ([Fig. 14\(c\)](#)). K-Mean clustering was able to properly segment the area with lighter illumination in [Fig. 14\(c\)](#).

Recognition-Based Method

The scale-invariant feature transform (SIFT) in Fiji (JavaSIFT) was used to identify objects in the test images. SIFT attempted to extract and store key features of objects of interest from the input image; it then use these features to recognize the objects, e.g., tracking them. To demonstrate image tracking feature, a 15 degree rotated image of [Fig. 9](#) was used as a source image

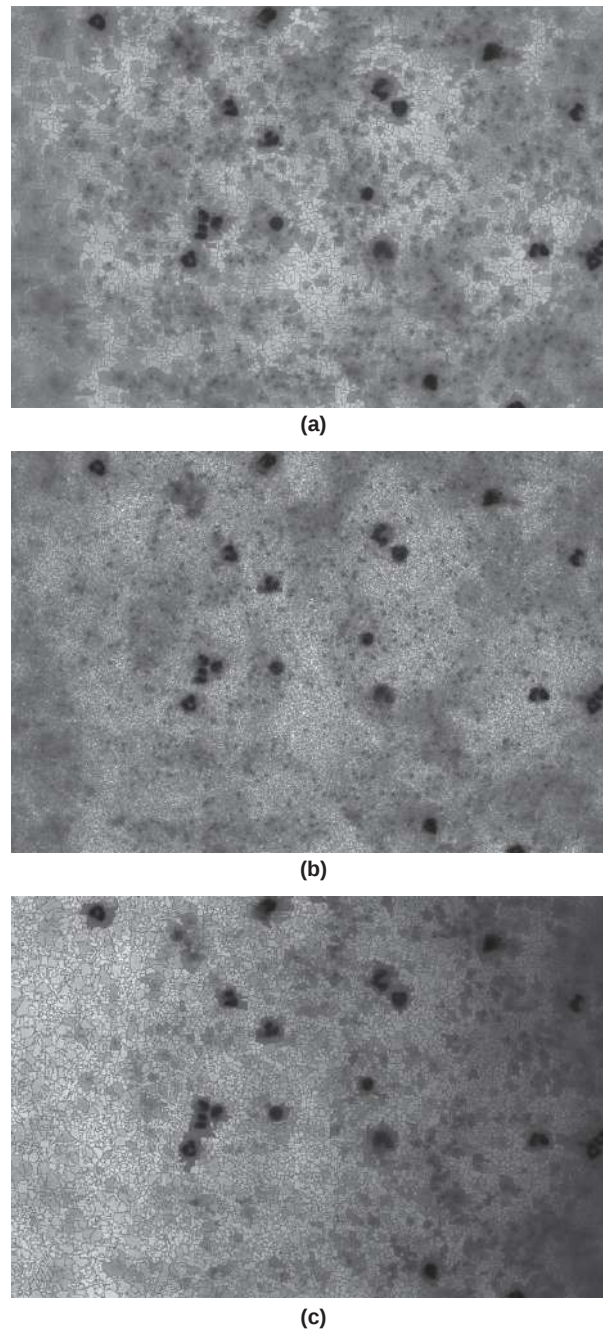


Fig. 12 Result images using Watershed region-based method (a) In-focus image. (b) Image with noise. (c) Image with uneven illumination.

(see Fig. 15(a)); then SIFT extracted object features for future looking up. Using Fig. 9(a–c) as the input SIFT tracked those objects from Fig. 15(a) and produced the tracked results in Fig. 15(b–d), respectively. Yellow marks on these resulting figures reveal the tracked objects. SIFT is able to track (recognize) the objects using the stored key features on all variant of blood smear images.

Computational Intelligence Method

The trainable Weka segmentation with Bayesian classifier in Fiji is used in to analyze the previous malaria images. Prior to performing image segmentation, we assigned two groups of the training set, namely white blood cell and malaria parasite groups. After training, Bayesian classifier was able to properly segment the original blood smear image, where majority of white blood cells and parasites were correctly segmented (Fig. 16(a)). When encountering with random noise signals, however,

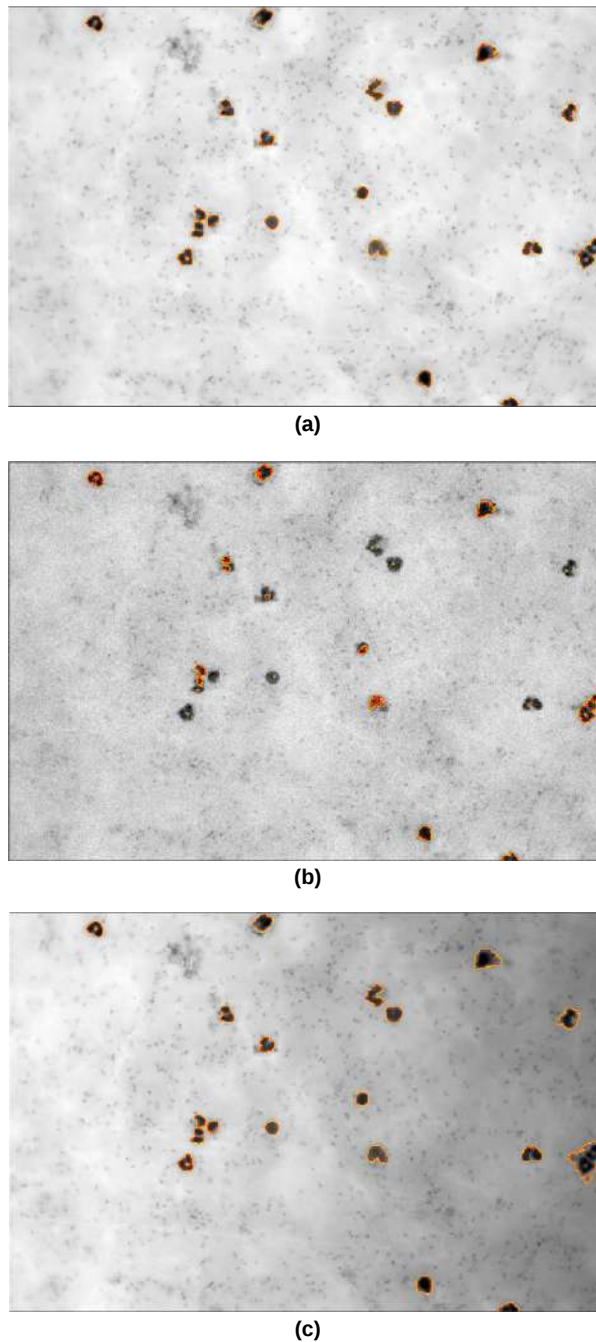


Fig. 13 Result images using Level set contour-based method (a) In-focus image. (b) Image with noise. (c) Image with uneven illumination.

the number of detected malaria parasites slightly dropped ([Fig. 16\(b\)](#)). The Bayesian classifier was able to largely detect the objects from both groups, except the region toward the rightmost end of the uneven illumination image that has the darkest shade ([Fig. 16\(c\)](#)).

Segmentation Techniques: Performance Criteria

Performance of an image processing algorithm can either be subjective (judging by feeling) evaluation or objective (judging by some measurable quality values) evaluation. Subjective evaluation is commonly conducted via human visual inspection; thus, it may be different from one evaluator to the other. In image segmentation, the performance of a segmentation method can be done

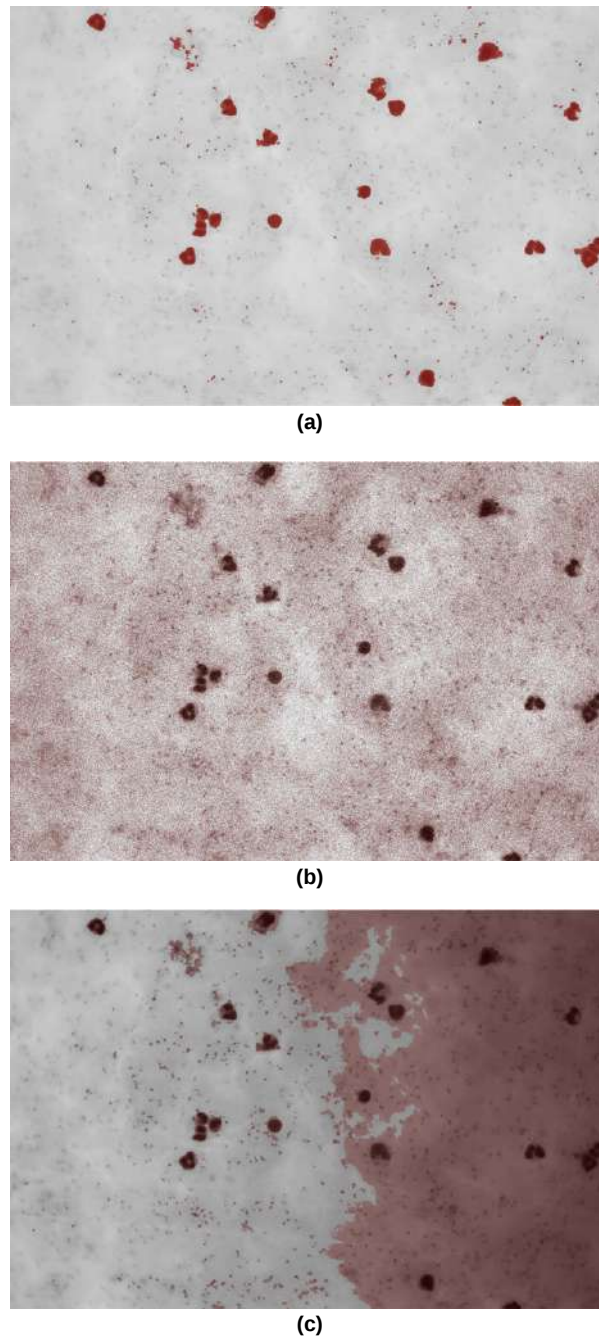
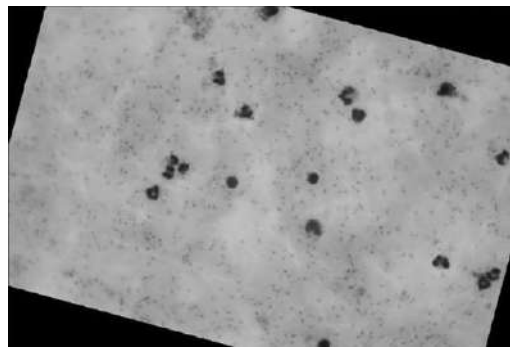
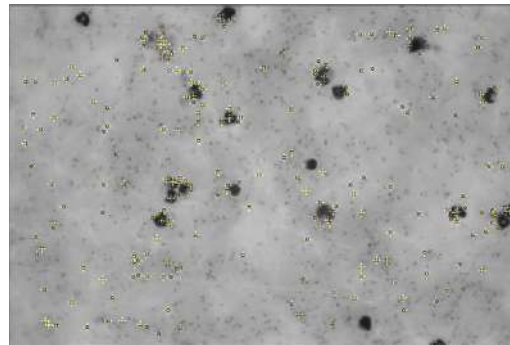


Fig. 14 Result images using K-Mean clustering-based method (a) In-focus image. (b) Image with noise. (c) Image with uneven illumination.

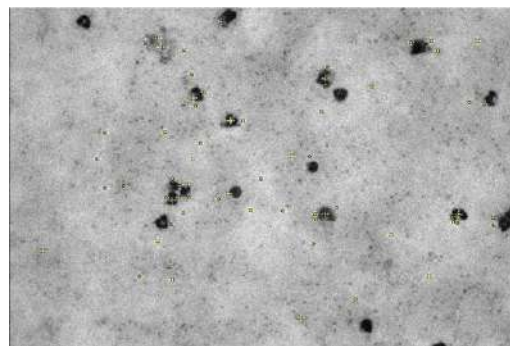
by visually comparing the original image to the segmented image. For objective evaluation, various mathematical measurement criteria are adopted for examples, misclassification error, edge mismatch, relative foreground area error, modified Hausdorff distance and region nonuniformity (Sezgin and Sankur, 2004). Misclassification error estimates the percentage of the misclassification between object and background. Edge mismatch calculates differences of the edge map between the original image and the corresponding segmented image. Relative foreground area error compares object properties in segmented image with the original one. The modified Hausdorff distance assesses the similarity of shape of segmented regions and compares with the ground-truth shapes. For the region nonuniformity measurement and comparison, the uniformity of objects and background from a segmented image are used to calculate variance of both segmented region. Since only objective evaluation may not be enough to effectively and correctly appraise a segmentation algorithm, subjective evaluation is also a common practice. Both objective and subjective evaluation.



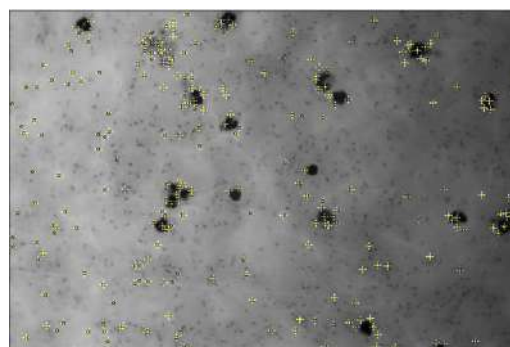
(a)



(b)



(c)



(d)

Fig. 15 Result images using SIFT recognition-based method (a) 15 degree rotated image for SIFT feature extraction, (b) tracked objects on In-focus image, (c) tracked objects on noisy image, and (d) tracked objects from the uneven illumination image.

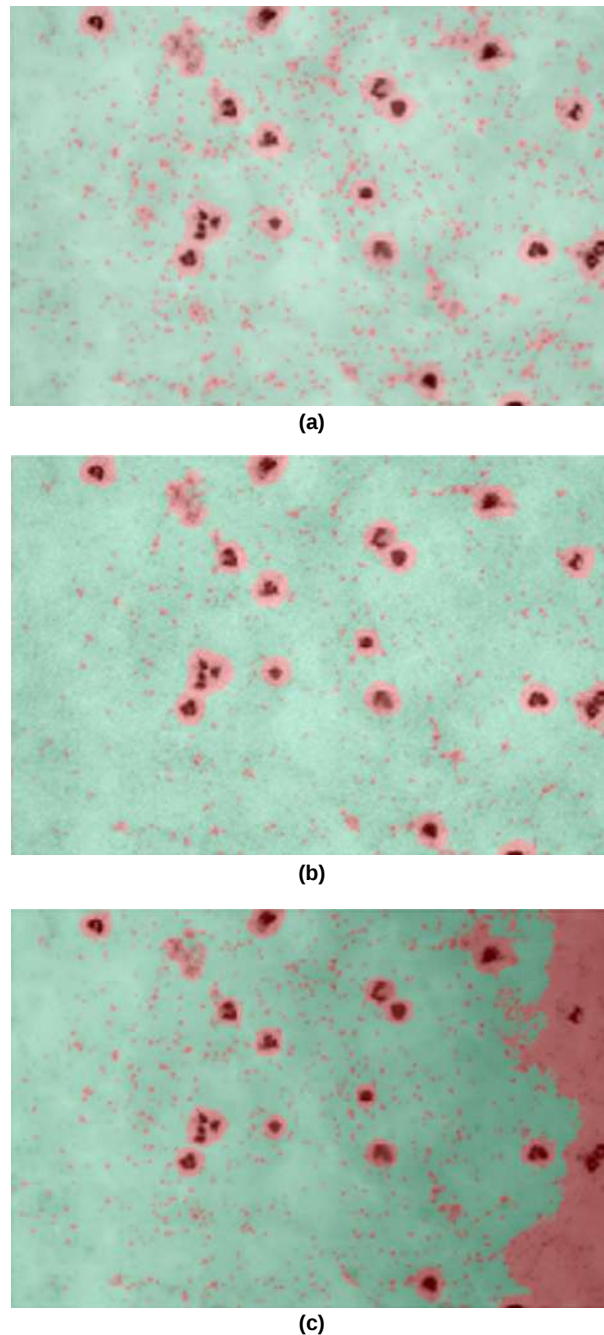


Fig. 16 Result images using Natural inspired-method (a) In-focus image. (b) Image with noise. (c) Image with uneven illumination.

Conclusion

The ultimate goal of image segmentation is to efficiently and effectively partition an image into meaningful objects or regions. Segmentation techniques have been widely studied and some of them have been implemented as tools at different levels, namely, lower level, middle level and high level. In this article, the segmentation techniques are categorized into seven classes, thresholding, edge-based, region-based, contour-based, clustering, recognition-based and computation-intelligence segmentations. Each class is evaluated based on three aspects, input constraints, output of methods and performance ([Table 1](#)).

Based on [Table 1](#), Thresholding methods, Edge-based methods, Region-based methods, Contour-based methods and clustering methods are not required prior knowledge; however, these methods are very sensitive and may fail when dealing with noisy image and uneven illumination image while the recognition methods and computational intelligence methods could perform better in both images but require a prior knowledge from user. The last two require high complexity and more stability comparing to the others. The contour-

Table 1 The evaluation of segmentation methods

Methods	Input constraints			Output of methods				Performance	
	Noise	Uneven illumination	Requiring prior knowledge	Edge	Region	Contour	Object of interest	High complexity	Stability
Thresholding	✗	✗	✗	✗	✓	✗	✗	✗	✗
Edge-based	✗	✗	✗	✓	✗	✗	✗	✗	✗
Region-based	✗	✗	✗	✗	✓	✗	✗	✗	✗
Contour-based	✗	✗	✗	✗	✓	✓	✗	✗	✓
Clustering	✗	✗	✗	✗	✓	✗	✗	✗	✓
Recognition	✓	✓	✓	✗	✓	✗	✓	✓	✓
Computational Intelligence	✓	✓	✓	✗	✓	✗	✓	✓	✓

Notes: ✗ means a method does not have an ability and ✓ means a method do have an ability.

Noises: unwanted interruptions which causes a number of pixels in the image to change in color and/or brightness.

Prior knowledge: addition data added to the system to help it identify, recognize, classify, differentiate or etc.

Uneven illumination: the level of brightness is not the same throughout the image.

Edge: where there is a sudden change of intensity between pixels. (most edges do not connect).

Region: an area of pixels which have similar properties.

Contour: boundary of regions which could be formed by a set of connected edges.

Object of interest: a meaningful object or region.

High complexity: computational complexity.

Stability: ability of a technique to remain unchanged under unexpected condition.

based method is stable as it can perform well on the uneven illumination image. Although there are many segmentation techniques have been proposed, they are not able to efficiently handle all possible image conditions as shown in the case study. Many factors such as image preparation process, acquisition process and complexity of an image itself present greater challenges for researchers to explore.

Acknowledgement

We would like to thank Napat Kaewkamnerd for preparing and processing those malaria images using various algorithms in Fiji presented in the case study section.

See also: Natural Language Processing Approaches in Bioinformatics

References

- Adamu, M.J., Ding, X., 2015. A relative study on image segmentation methods. *International Journal of Science and Research* 6, 1975–1981.
- Anjna, E., Kaur, R., 2017. Review of image segmentation technique. *International Journal of Advanced Research in Computer Science* 8, 36–39.
- Bansal, B., Saini, J.S., Bansal, V., Kaur, G., 2012. Comparison of various edge detection techniques. *Journal of Information and Operations Management* 3, 103–106.
- Baswaraj, D., Govardhan, A., Premchand, P., 2012. Active contours and image segmentation the current state of the art. *Global Journal of Computer Science and Technology Graphics & Vision* 12, 1–13.
- Bhaidasna, Z.C., Mehta, S., 2013. A review on level set method for image segmentation. *International Journal of Computer Applications* 63, 20–22.
- Borenstein, E., Ullman, S., 2002. Class-specific, top-down segmentation. In: *Proceedings of the 7th Proceeding of the European Conference on Computer Vision-Part II*, pp. 109–122. London: Springer-verlag.
- Canny, J., 1986. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 6, 679–698.
- Cheng, H.D., Chen, Y.H., 1999. Fuzzy partition of two-dimensional histogram and its application to thresholding. *Pattern Recognition* 32, 825–843.
- Christ, M.C.J., Parvathi, R.M.S., 2011. Fuzzy c-means algorithm for medical image segmentation. In: *Proceedings of the 3rd International Conference on Electronics Computer Technology*, pp. 33–36. Kanyakumari: IEEE.
- Courprie, M., Bertrand, G., 1997. Topological grayscale watershed transformation. *SPIE Vision Geometry* 3168, 136–146.
- Dass, R., Priyanka, Devi, S., 2012. Image segmentation techniques. *The International Journal of Electronics & Communication Technology* 3, 66–70.
- Dehariya, V.K., Shrivastava, S.K., Jain, R.C., 2010. Clustering of image data asset using K-Means and Fuzzy K-Means algorithms. In: *International Conference on Computational Intelligence and Communication Networks*. pp. 386–391. Bhopal, IEEE.
- Ferrari, V., Tuytelaars, T., Gool, L.V., 2006. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision* 67, 1–26.
- Gonzalez, R.C., Woods, R.E., 2008. *Digital Image Processing*, third ed. Upper Saddle River: Pearson Prentice Hall.
- Gorte, B.G.H., 1996. Multi-spectral quadtree based image segmentation. *International Archives of Photogrammetry and Remote Sensing* 31, 251–256.
- Gupta, H., Schmitter, D., Uhlmann, V., Unser, M., 2017. General surface energy for spinal cord and aorta segmentation. In: *Proceedings of the 14th International Symposium on Biomedical Imaging*, pp. 319–322. Melbourne: IEEE.
- Havaei, M., Davy, A., Warde-Farley, D., et al., 2017. Brain tumor segmentation with deep neural networks. *Medical Image Analysis* 35, 18–31.
- Jia, G., Heymsfield, S.B., Zhou, J., Yand, G., Takayama, Y., 2016. Quantitative biomedical imaging: Techniques and clinical applications. *BioMed Research International* 2016, 1–2.
- Jodas, D.S., Pereira, A.S., Tavares, R.S., 2003. Automatic segmentation of the lumen region in intravascular images of the coronary artery. *Medical Image Analysis* 40, 60–79.

- Kaewkamnerd, S., Uthaipibul, C., Intarapanich, A., *et al.*, 2012. An automatic device for detection and classification of malaria parasite species in thick blood film. *BMC Bioinformatics* 13, 1–10.
- Kapur, J.N., Sahoo, P.K., Wong, A.K., 1985. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision Graphics and Image Processing* 29, 273–285.
- Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes, active contour model. *International Journal of Computer Vision*. 321–331.
- Kaur, D., Kaur, Y., 2014. Various image segmentation techniques: A review. *International Journal of Computer Science and Mobile Computing* 3, 809–814.
- Kirsch, R., 1971. Computer determination of the constituent structure of biological images. *Computers and Biomedical Research* 4, 315–328.
- Krishnan, K.B., Ranga, S.P., Guptha, N., 2017. A survey on different edge detection techniques for image segmentation. *Indian Journal of Science and Technology* 10, 1–8.
- Levin, A., Weiss, Y., 2009. Learning to combine bottom-up and top-down segmentation. *International Journal of Computer Vision* 81, 105–118.
- Murphy, D.B., Davidson, M.W., 2001. *Fundamentals of light microscopy and electronic imaging*, second ed. Singapore: Wiley-Blackwell.
- Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computation Physics* 79, 12–49.
- Otsu, N., 1979. A threshold selection method from gray-level histogram. *IEEE Transaction on System Man Cybernet SMC-8*. 62–66.
- Pal, S.K., Rosenfeld, A., 1988. Image enhancement and thresholding by optimization of fuzzy compactness. *Pattern Recognition Letter* 7, 77–86.
- Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine* 22, 52–67.
- Patil, D.D., Deore, S.G., 2013. Medical image segmentation. *International Journal of Computer Science and Mobile Computing* 2, 22–27.
- Prewitt, J.M.S., 1970. Object enhancement and extraction. In: Rosenfeld, A. (Ed.), *Picture Processing and Psychopictorics*. New York, NY: Academic Press, pp. 75–149.
- Rahebi, J., Elmi, Z., Farzamnai, A., Shayan, K., 2010. Digital image edge detection using an ant colony optimization based on genetic algorithm. In: *Conference on Cybernetics and Intelligent Systems*, pp. 145–149. Singapore: IEEE.
- Schindelin, J., Arganda-Carreras, I., Frise, E., *et al.*, 2012. Fiji: An open-source platform for biological-image analysis. *Nature Methods* 7, 676–682.
- Schneider, C.A., Rasband, W.S., Eliceiri, K.W., 2012. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods* 7, 671–675.
- Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* 13, 146–165.
- Sharma, A., Chaturvedi, R., Dwivedi, U.K., 2015. Recent trends and techniques in image segmentation using particle Swarm optimization-a survey. *International Journal of Scientific and Research Publication* 5, 1–6.
- Sharma, P., Suji, J., 2016. A review on image segmentation with its clustering techniques. *International Journal of Signal Processing, Image Processing and Pattern Recognition* 9, 209–218.
- Tsai, W.H., 1985. Moment-preserving thresholding: A new approach. *Graphics Models Image Process* 19, 377–393.
- Tsien, R.Y., 2003. Imagining imaging's future. *Nature Review Molecular Cell Biology* 4, SS16–SS21.
- Vantaram, S.R., Saber, E., 2012. Survey of contemporary trends in color image segmentation. *Journal of Electronic Imaging* 4, 1–28.
- Wang, L., Shi, J., Song, G., Shen, L., 2007. Object detection combining recognition and segmentation. In: *Proceedings of the 8th Asian Conference on Computer Vision – Volume Part I*, pp. 189–199. Heidelberg: Springer-Verlag.
- Win, Y.K., Choomchuay, S., 2017. Automated segmentation of cell nuclei in cytology pleural fluid images using OTSU thresholding. In: *International Conference on Digital Arts, Media and Technology*, pp. 1–5.
- Zhang, Y.J., 2001. An overview of image and video segmentation in the last 40 years. In: *Proceedings of the 6th International Symposium on Signal Processing and Its Applications*, pp. 144–151. IGI Global Disseminator of Knowledge.
- Zhang, Y.J., 2009. Image segmentation in the last 40 years. In: Khosrow-Pour, M. (Ed.), *Encyclopedia of Information Science and Technology*, second ed. New York, NY: Information Science Reference, pp. 1818–1823.
- Zheng, S., Jayasumana, S., Romera-Paredes, B., *et al.*, 2015. Conditional random fields as recurrent neural networks. In: *Proceeding of the IEEE International Conference on Computer Vision*, pp. 1529–1537. Washington, DC: IEEE Computer Society.

Relevant Websites

<http://bioimages.vanderbilt.edu/Bioimage>.
<https://fiji.sc>
 Fiji.
<https://imagej.net/Welcome>
 ImageJ.
https://en.wikipedia.org/wiki/Medical_imaging
 Medical Imaging.

Biographical Sketch



Saowaluck Kaewkamnerd is currently a principal researcher and the head of Biomedical Signal Processing Laboratory at National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA), Thailand. Dr. Kaewkamnerd received her Ph.D. in Electrical Engineering from the University of Texas at Arlington, United States, in 2001. She has experienced in conducting research in biomedical image processing as well as other signal processing related projects.



Apichart Intarapanich is currently a researcher in Biomedical Signal Processing Laboratory at National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA), Thailand. He obtained his doctoral degree in Electrical Engineering from the University of Calgary, Canada in 2005. His specialized skills include Signal Processing for Telecommunication Section, implementing a real-time microphone array system for sound quality improvement, antenna design for WLAN, smart Antenna System development and PHS handset development.



Sissades Tongsimma received his Ph.D. in Computer Science and Engineering from the University of Notre Dame, United States in 1999. He worked at the National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA) until 2003. Since then he has moved to the National Center for Genetic Engineering and Biotechnology (BIOTEC), NSTDA as a senior researcher. Currently, he is now a principal researcher and head of Bioinformatics Laboratory, Genome Technology Research Unit at BIOTEC. His research interests include – omics Big data analysis as well as utilizing digital signal processing in his research. He serves as an executive committee of Asia Pacific Bioinformatic Network (APBioNET) and an associate editor of the journal of Human Genetics.

Translational and Disease Bioinformatics

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal and University of Minho, Braga, Portugal

Chakrawarti Prasun and Deepak Sharma, Indian Institute of Technology Roorkee, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

The completion of the human genome project and a huge amount of data generated from the health care system have presented us with tremendous opportunities to understand and analyze our biological system in a greater detail. The data generated during clinical practice/research is used in finding its correlation with genomic data to obtain biologically significant insights. Due to the enormous size of data, the knowledge of information technology is required to store, integrate and analyze it. Analysis of complex biological data using knowledge of computer science, statistics and mathematics has resulted in the birth of a new field of life sciences, 'bioinformatics'.

Translational bioinformatics is an essential player in bench-to-bedside research, i.e., transformation of data into diagnostics, prognostics and therapeutics (Kann, 2010; Yan, 2011). It is the development of storage-related, analytic and interpretive methods by applying informatics methodology on voluminous biomedical and genomic data. It is particularly useful to find specific knowledge and medical devices that can be utilized by various stakeholders like biomedical scientists, clinicians and patients for improving patient care system, real time health monitoring and rational drug discovery. To understand translational bioinformatics, it is imperative to know the various types of data as well as the numerous tools and techniques to integrate and interpret these data (Altman, 2012). In this article, we describe various components of translational bioinformatics (Fig. 1).

Types of Datasets

The data utilized in translational bioinformatics can be broadly classified into biological and clinical data. An effective integration of these large heterogeneous datasets from classified areas, for instance molecular biology and clinical practice, provides clinically significant information (Sorani *et al.*, 2010).

Biological Data

The biological data encompasses molecular biology databases, which includes systematic arrangement of genomics, proteomics, metabolomics, microarray gene expression and phylogenetic data (Baxevanis, 2001). It can be further categorized into sequence, structure, functional or gene ontology database. Sequence databases include nucleic acid and protein sequences. Structure database encompasses RNA and protein structure. Functional database includes the function of gene and gene products. Functional or gene ontology is classified further based on molecular function, cellular component and biological process. If the functional database relates to physiological role of gene products, it is termed as molecular function database; if it determines the subcellular structures, it is called cellular component database; and if it determines the biological pathway in combination of other gene products, it is known as biological process database (Ashburner *et al.*, 2000). Gene ontology in conjunction with anatomical data

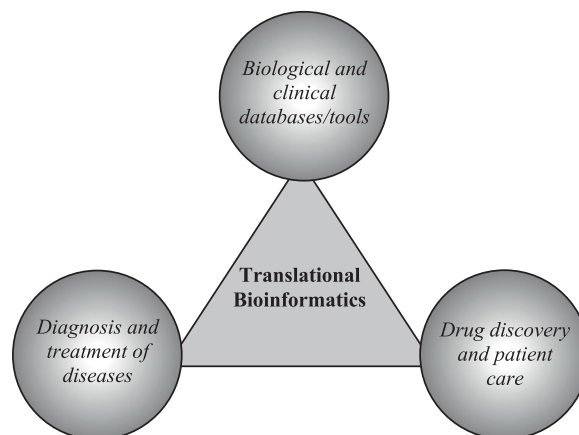


Fig. 1 Components of translational bioinformatics.

connects the parent and children phenotype. Similarly, biological process and disease progression, which are genetically determined, can be studied, and prophylactic action can be taken to slow down or avoid disease conditions from occurring.

The prominent biological databases implicated in the field of translational bioinformatics encompass data across metabolic/signaling pathways, protein-protein interactions, gene expression, polymorphisms, enzyme databases as well as drug related databases, which include drug structure, indication, pharmacological information, target structure/binding information and drug toxicity/adverse drug reactions (ADRs) data. Several databases and tools have been developed on metabolic/signaling pathways analysis, amongst them human metabolome database (HMDB), Kyoto Encyclopedia of Genes and Genomes (KEGG), Pathguide, pathDIP and XTalkDB are the most prominent and with distinct focus (Table 1).

HMDB provides information of human metabolite/metabolism, its detailed description, synonyms, structure, physico-chemical properties and reference NMR/MS spectra along with disease associations, pathway information, enzyme data, gene sequence data, SNP (single-nucleotide polymorphism) and mutation data (Wishart *et al.*, 2018). On the other hand, KEGG is a database and tool for analysis of gene function for genomes of 24 species including human, by linking the genomic information with cellular processes [such as metabolism, membrane transport, signal transduction and cell cycle] (Kanehisa and Goto, 2000). It has three databases, GENES database for genomic information, PATHWAY database for graphical representations of cellular processes and LIGAND database for the information about chemical compounds, enzyme molecules and enzymatic reactions. In contrast, Pathguide is a meta-database on

Table 1 Prominent biological and clinical databases/tools

Website	Key feature(s)	CI ^a	Access ^b	URL
BindingDB	Database of protein-ligand binding affinities	85.2	W	https://www.bindingdb.org/bind/index.jsp
BRENDA	Comprehensive enzyme information database	43.6	W	https://www.brenda-enzymes.org/
CAGE	Cap-analysis gene expression (CAGE) analysis database	27.7	W	https://cage-seq.com/
CTD	Illuminates how chemicals affect human health	22.7	W	http://ctdbase.org/
DrugBank	Comprehensive drug target database	174.6	W	https://www.drugbank.ca/
DynamProt	For tracking the levels and location of endogenous proteins in human cells	1.6	W	http://www.weizmann.ac.il/mcb/UriAlon/DynamProt/
Gene expression barcode	Barcode algorithm to estimate expressed and unexpressed genes in a microarray hybridization	20.9	W	http://barcode.luhs.org/
GeneWeaver	Cross-species data and gene entity integration database for scalable hierarchical analysis	12.4	W	https://geneweaver.org/
HIT	Herb Ingredients' Targets database	14.6	W	http://lifecenter.sgst.cn/hit/
HMDB	Human metabolite/metabolism database	142.7	W	http://www.hmdb.ca/
KEGG	Knowledge base for systematic analysis of gene functions, linking genomic information with higher order functional information	553.2	W	http://www.genome.jp/kegg/
LigAsite	Database of biologically relevant binding sites in proteins with known apo-structures	6.8	W	http://ligasite.org/
MIMIC	Repository of physiologic signals and vital signs, captured from patient monitors, and comprehensive clinical data	87.8	W	https://physionet.org/mimic2/
MINT	Experimentally verified protein-protein interactions database	71.0	W	http://mint.bio.uniroma2.it/
PANTHER	Tool to decipher the function of uncharacterized genes based on their evolutionary relationships	49.3	W	http://www.pantherdb.org/
pathDIP	Database of signaling cascades, comprising core pathways	3.3	W	http://ophid.utoronto.ca/pathdip/
Pathguide	Meta-database of biological pathways	31.3	W	http://www.pathguide.org/
PolyDoms	Whole genome database for the identification of non-synonymous coding SNPs	5.8	W	https://polydoms.cchmc.org/polydoms/
SIDER	Repository of marketed medicines and their recorded ADRs	60.0	W	http://sideeffects.embl.de/
SuperDRUG2	Database of approved/marketed drugs with chemical structures, dosage, biological targets, physicochemical properties, side-effects and pharmacokinetic data	- ^c	W	http://cheminfo.charite.de/superdrug2/
T3DB	Toxin and toxin target database	15.2	W	http://www.t3db.ca/
TTD	Therapeutic protein and nucleic acid targets database	19.2	W	http://bidd.nus.edu.sg/group/cjttd/
WeGET	Tool to find mammalian genes that strongly co-express with a human query gene set	3.8	W	http://weget.cmbi.umcn.nl/
XTalkDB	Database of signaling pathway crosstalk	8.4	W	http://www.xtalkdb.org/home
ZINC	Database of commercially-available compounds for virtual screening	215.9	W	http://zinc.docking.org/

^aCI, citation index (number of citations per year).

^bW, Web based.

^c-, no citations.

metabolic pathways, signaling pathways, transcription factor targets, gene regulatory networks, genetic interactions, protein-compound interactions and protein-protein interactions (Bader *et al.*, 2006). It is an overview of more than 190 web-accessible biological pathway and network databases, maintained by diverse groups in different locations. Recently, a resource pathDIP was developed that integrates core pathways (signaling/metabolic pathway databases) to predict biologically relevant protein-pathway associations (Rahmati *et al.*, 2017). It annotates 17070 protein-coding genes with 4678 pathways. Concurrently, a distinct tool XTalkDB was established for the analysis of signaling pathways and their crosstalk (Sam *et al.*, 2017). In addition, it covers the molecular components (e.g., proteins, hormones, microRNAs) that mediate crosstalk between a pair of pathways and the species and tissue.

Out of several protein-protein interactions databases, the most used are PANTHER (Protein ANALysis THrough Evolutionary Relationships), MINT (Molecular INTeraction Database) and DynamProt (Dynamics Proteomics) (Table 1). One of the first one to be developed was PANTHER that provides the function of uncharacterized genes (from any organism) based on their evolutionary relationships to genes with known functions (Mi *et al.*, 2005). It contains statistical models (Hidden Markov Models, or HMMs) to find out proteins (and their genes) family and subfamily that allows more precise evolutionary information. Subsequently, MINT database was established that catalogs physical interactions (obtained from large scale, genome wide experiment) between proteins stored in structured format (Chatr-aryamontri *et al.*, 2007). It also includes an integrated feature, HomoMINT for interactions between human proteins with orthologous proteins in model organisms. Another useful resource is DynamProt which is compendium of endogenously tagged human proteins expressed from its endogenous chromosomal location under its natural regulation (Frenkel-Morgenstern *et al.*, 2010). It illustrates the protein dynamics and localizations in individual living human cells following drug administration which is crucial for understanding effect of drug on cell.

Gene expression databases include CAGE, GeneWeaver, Gene Expression Barcode and WeGET (Table 1). CAGE (Cap analysis gene expression) is a highly sensitive and precise means of gene expression technique to produce high-throughput CAGE library (Kodzius *et al.*, 2006). It is primarily used to locate exact transcription start sites in the genome and allows investigation of promoter structure necessary for gene expression. It contributes to genome annotation, gene discovery and expression profiling. In comparison, GeneWeaver is a platform for integration of cross-species data and gene entity as well as scalable hierarchical analysis of investigator's experimental data with a community-built and curated data archive of gene sets and gene networks (Baker *et al.*, 2012). It provides means for data driven comparison of biological, behavioral and disease concepts. Gene Expression Barcode is the first database to provide absolute measures of expression for most annotated genes (McCall *et al.*, 2014). The expression of genes across tissues and cell types will help in generating hypothesis for gene function and clinical predictions using gene expression signatures. Recently, WeGET (Weighted Gene Expression Tool and database) was developed to predict new genes of a molecular system by correlating gene expression (Szkarczyk *et al.*, 2016). It ranks new candidate genes based on their weighted co-expression with that system and predict novel genes that co-express with a custom query set. On the other hand, BRENDA is one of the most comprehensive enzyme repositories (Schomburg *et al.*, 2004). It provides functional and molecular information of enzymes which is crucial for research of enzyme mechanisms, metabolic pathways and, furthermore, for medicinal diagnostics and pharmaceutical research. Additionally, PolyDoms is a whole genome database for polymorphism analysis and identification of non-synonymous coding SNPs (nsSNPs) with the potential to impact disease (Jegga *et al.*, 2007). It also predicts structural and functional impacts of all nsSNPs and identifies potentially harmful nsSNPs among multiple genes associated with specific diseases, anatomies, mammalian phenotypes, gene ontologies, pathways or protein domains.

The drug related databases cover drug structure database such as ZINC, DrugBank and SuperDRUG2, adverse drug reaction or toxicological databases such as CTD (Comparative Toxicogenomics Database), T3DB (Toxin and Toxin Target Database) and SIDER (Side Effect Resource) as well as drug target databases like TTD (Therapeutic Target Database) and HIT (Herb Ingredients' Targets). ZINC contains over 35 million molecules and provides their 3D structures for virtual screening against the specified target (Irwin and Shoichet, 2005) while DrugBank contains prescribing and pharmacological information including kinetics, dynamics, mechanism, target, interactions and metabolism pathway along with chemical, spectroscopic, patent information and 3D structure of a drug (Wishart *et al.*, 2008). In a similar vein, SuperDRUG2 provides indications, drug targets, side-effects, physicochemical properties, drug-drug interactions, pharmacological, chemical and regulator marketed drug/marketed drugs (Siramshetty *et al.*, 2018). In contrast, CTD provides the information about the effects of environmental chemicals on human health (Davis *et al.*, 2009). It also curates chemical – Gene interactions, chemical – Disease relationships and gene – Disease relationships to construct chemical – Gene – Disease networks by data integration. Similarly, T3DB is a database of toxic exposome (environmental exposures of human to an acutely toxic compounds) (Wishart *et al.*, 2015). It contains detailed information of toxic substances that is, chemical properties, descriptions, targets, toxic effects, toxicity thresholds, sequences (for both targets and toxins) and mechanisms. Recently, SIDER database was developed that hosts drug's side effects or adverse drug reactions [ADRs] (Kuhn *et al.*, 2016). TTD is a repository of known therapeutic protein and nucleic acid targets (Chen *et al.*, 2002). It covers sequence, 3D structure, function, nomenclature, drug/ligand binding properties and drug effects. Another distinct useful resource is HIT which is a database of herbal ingredients with protein target information (Ye *et al.*, 2011).

Some important binding site information databases are BindingDB and LigASite. BindingDB is a resource of binding affinity of protein-ligand complexes (Liu *et al.*, 2007). In comparison, LigASite is a repository of biologically relevant binding site in proteins for which at least one apo- and one holo-structure (ligand unbound and bound structure, respectively) are available (Dessailly *et al.*, 2008).

It is noteworthy to mention that the popular/useful general tools and databases of DDBJ, NCBI, EMBL, ExPASy as well as CLUSTALW and PDB have not been covered as they are beyond the focus/scope of this article.

Clinical Data

The data collected by physicians while treating patients and during clinical trials are termed as clinical databases (Blum, 1982). It includes comprehensive data pertaining to drug effects, electronic health records (EHRs), drug-drug interactions data, patients' medical histories and demographic data, pathology, diagnostic data and data of concomitant medications. The various databases where such clinical data are stored are MIMIC, ClinicalTrials.gov, EMA Clinical Data and Clinical Data Repository.

MIMIC (Multiparameter Intelligent Monitoring in Intensive Care) is a database of physiologic signals and vital signs captured from patient monitors (Saeed *et al.*, 2011). Clinical Trials (see "Relevant Websites section") is a resource of clinical trials and their outcomes, provided by the U.S. National Library of Medicine. EMA Clinical Data (see "Relevant Websites section") web portal hosts clinical data published under the European Medicines Agency. Clinical Data Repository (CDR) is a real time database that houses data from a variety of clinical sources such as 'UVA clinical data repository' (see "Relevant Websites section"), containing clinical information on over 1 million patients and 5 million clinical encounters, the clinical data repository ALLOY from Liaison (see "Relevant Websites section") and Inovalon's clinical data repository (see "Relevant Websites section").

Integration and Analysis of Data for Diagnosis and Treatment

After the collection of voluminous data on numerous diseases, the next crucial step is to store it systematically so that it can be easily retrieved/processed and analyzed. Volume, Variety and Velocity (3V's) are formidable challenges in such big data management (Sagiroglu and Sinanc, 2013). To manage these challenges, numerous tools and techniques have been developed. Big data could be structured (stored as arrays, files, records, tables or trees), unstructured (either in text-format for instance PowerPoint presentations, word documents, PDFs or non-text format like images, audios, videos) or semi structured [mixed] (Raghupathi and Raghupathi, 2014). Many approaches and technologies for storage and processing have been developed for better management and quick retrieval/response. Notable among these are Distributed File System (a server based application that allows remote data access of structured data to several clients simultaneously e.g., Apache Hadoop), NoSQL (for storage and retrieval of non-tabular data), Cloud Computing (for storage on a network of remote servers hosted on internet rather than a personal computer or local server) and Massive Parallel Processing (MPP, coordinated processing of a program by different processors working on different parts of the program) (Katal *et al.*, 2013).

Apart from storage and processing techniques, powerful and highly analytical tools are required to visualize, analyze and integrate the vast data. Some of the most popular tools toward this context include EHDViz (for data visualization), Theano (for deep learning), PreGel, Apache Giraph and Gephi (for graph processing) as well as Apache Mahout, Spark MLlib and Weka (for machine learning) (Shameer *et al.*, 2017). Machine and deep learning techniques are used to process and analyze variety of data with high velocity. Machine learning is a part of artificial intelligence (AI) that trains the computer to perform the desired task automatically and more efficiently and accurately. There are different machine learning techniques like supervised learning (viz. Naïve Bayes Classifiers, Decision Tree, Support Vector Machine), unsupervised learning, reinforcement learning, deep learning, association rule learning and numeric prediction (Tan and Gilbert, 2003). Towards this end, several web servers and programs have been developed. For instance, an algorithm like Decision Support Systems (DSS), a program which facilitates complex decision making and problem solving by compiling, either graphically or as written report, massive reams of data to provide analytical result (Somek and Hercigonja-Szekeres, 2017). KDSS (Knowledge-based Decision Support System) is an example of DSS using a knowledge based coding to provide a novel workflow for protein complex extraction (Fiannaca *et al.*, 2013).

Cardiovascular diseases, central nervous system (CNS) diseases, respiratory diseases, cancer, diabetes, tuberculosis and viral diseases (like HIV, hepatitis, dengue, influenza/flu, measles) are the major cause of deaths globally (WHO report 2017). Amongst these, computational tools and databases implicated in cancer (Blekherman *et al.*, 2011), tuberculosis (Sharma *et al.*, 2015) and viral diseases (Sharma and Surolia, 2011) have already been compiled/described in great details. Prominent tools for cardiovascular, CNS and respiratory diseases as well as diabetes research are listed in Table 2. It also enlists useful tools that are not specific to any particular set of diseases.

CADgene, T-HOD or CardioSignal are gene databases associated with cardiovascular diseases (Table 2). CADgene is a comprehensive resource and integration tool for coronary artery disease genes (Liu *et al.*, 2011). It provides detailed information about the size of case-control, population, SNP, odds ratio and P-value, for each gene. T-HOD is a gene database for hypertension, obesity and diabetes (Dai *et al.*, 2013). It has a gene/disease identification system and a disease-gene relation extraction system to affirm the association of genes with these three diseases. CardioSignal is a repository of transcriptional regulation involved in heart development and cardiac hypertrophy (Zhen *et al.*, 2007). On the other hand, PubAngioGen is a comprehensive database for exploring the connection between angiogenesis and diseases at multi-levels including protein-protein interaction, drug-target, disease-gene and signaling pathways (Li *et al.*, 2015). In contrast, miRHrt is a database of microRNA function in heart development and by computational analysis it illustrates the correlation of differentially expressed miRNAs with cellular functions and heart development (Liu *et al.*, 2010). CARFMAP is an interactive cardiac fibroblast pathway map derived from biomedical literature (Nim *et al.*, 2015). It enables identification of new genes that are relevant to cardiac research and differentially regulated in high-throughput assays. Another useful tool is ECGSYN (see "Relevant Websites section") that generates a synthesized ECG signal with user-settable mean heart rate, number of beats, sampling frequency, waveform morphology (P, Q, R, S, and T timing, amplitude and duration), standard deviation of the RR interval, and LF/HF ratio (a measure of the relative contributions of the low

Table 2 Disease-specific tools and databases

Disease	Website	Key feature(s)	CI ^a	Access ^b	URL
Cardiovascular diseases	CADgene	Coronary artery disease (CAD) gene database	9.5	W	http://www.bioguo.org/CADgene/about_us.php
	CardioSignal	Repository of transcriptional regulation in cardiac development and hypertrophy	0.8	W	http://www.cardiosignal.org/index.html
	CARFMAP	For analyses of cardiac fibroblasts	1.6	W	http://visionet.erc.monash.edu.au/CARFMAP/
	ECGSYN	ECG waveform generator with user-settable parameters	- ^c	W	https://www.physionet.org/physiotools/ecgsyn/
	miRHrt	Database of miRNAs function in heart development	2.2	W	http://cistrome.org/cr/miRHrt_for_loading/
	PubAngioGen	Resource dedicated to angiogenesis	1.4	W	http://www.megabionet.org/aspd/
	T-HOD	Gene database for hypertension, obesity and diabetes	3.0	W	http://bws.iis.sinica.edu.tw/THOD
Central Nervous System (CNS) diseases	ADHDgene	Genetic resource of Attention Deficit Hyperactivity Disorder	8.2	W	http://adhd.psych.ac.cn
	AlzGene	Database of Alzheimer disease susceptibility genes	137.0	W	http://www.alzgene.org
	AutDB	Repository of genes implicated in autism susceptibility	23.1	W	http://autism.mindspec.org/autdb/Welcome.do
	EpilepsyGene	Catalogs genes and mutations related to epilepsy	5.6	W	http://122.228.158.106/EpilepsyGene
	G2Cdb	Database for linking synaptic genes to cognition and disease	5.4	W	http://www.genes2cognition.org
	NSDNA	Nervous System Disease NcRNAome Atlas	2.4	W	http://www.bio-bigdata.net/nsdna/
	PDGene	Database to provide genetic association results in Parkinson's disease	62.5	W	http://www.pdgene.org
Diabetes	SZGR	Schizophrenia Gene Resource	10.1	W	https://bioinfo.uth.edu/SZGR/
	Islet Regulome Browser	Pancreatic islet genomic database	-	W	http://gattaca.imppc.org/isletregulome/home
	T1DBase	Molecular genetics database for T1D susceptibility and pathogenesis	3.3	W	http://T1DBase.org
	T2D Knowledge Portal	Data/tools to promote understanding and treatment of T2D and its complications	-	W	http://www.type2diabetesgenetics.org
	T2o-Db	Database of the molecular components involved in the T2D pathogenesis	3.7	W	http://t2ddb.ibab.ac.in
Respiratory diseases	IGDB.NSCLC	Repository of non-small cell lung cancer genes and microRNAs	2.4	W	http://igdb.nslc.ibms.sinica.edu.tw/
	LGEA	Resource for lung gene expression analysis	14.0	W	https://research.cchmc.org/pbge/lunggens/mainportal.html
	MyMpn	<i>Mycoplasma pneumoniae</i> database	6.3	W	http://mympn.crg.eu/
	SEGEL	For visualization of smoking effects on human lung gene expression	0.4	W	http://www.chengfeng.info/smoking_database.html
General tools	DisGeNET	Database to provide associations between genes and diseases, disorders and clinical or abnormal human phenotypes	75.5	W	http://www.disgenet.org/web/DisGeNET/menu/home
	DistiLD	Disease-gene associations database	3.1	W	http://distild.jensenlab.org
	Imprinted Gene Catalogue	Imprinted gene and parent-of-origin effect database	7.3	W	http://igc.otago.ac.nz/home.html
	LncRNADisease	Resource for long- non-coding RNA-associated diseases	66.4	W	http://cmbi.bjmu.edu.cn/lncrnadisease
	Organ System Heterogeneity DB	Repository of phenotypic effects of human diseases and drugs	0.7	W	http://mips.helmholtz-muenchen.de/Organ_System_Heterogeneity/

^aCI, citation index (number of citations per year).^bW, Web based.^c-, no citations.

and high frequency components of the RR time series to total heart rate variability). Additionally, a method for simulating atrial fibrillation has been developed using ECGSYN (Healey *et al.*, 2004). The simulator uses a timing operator to switch from generating normal ECG morphologies to atrial fibrillation.

Several genomic databases for central nervous system diseases include AlzGene, PDGene, ADHDgene, EpilepsyGene, AutDB or SZGR (Table 2). AlzGene is a resource of Alzheimer disease susceptibility genes with statistically significant allelic summary

(Bertram *et al.*, 2007). It also performs systematic meta-analyses for each polymorphism with available genotype data. Similarly, PDGene is a meta-analyses database of genes associated with Parkinson's disease (Lill *et al.*, 2012). ADHDGene is a genetic resource and analysis platform for Attention Deficit Hyperactivity Disorder [ADHD] (Zhang *et al.*, 2012). EpilepsyGene is a repository of genes and mutations related to epilepsy (Ran *et al.*, 2015). It also includes functional annotation, gene prioritization, functional analysis of prioritized genes, and overlap analysis focusing on the comorbidity. In contrast, AutDB is a well classified (as per genetic variation of candidates identified from genetic association studies, rare single gene mutations and genes linked to syndromic autism) database of genes connected to Autism Spectrum Disorders [ASD] (Basu *et al.*, 2009). It covers gene annotation for their relevance to autism, along with an in-depth view of their molecular functions. Schizophrenia Gene Resource (SZGR) catalogs genetic data from association studies, linkage scans, gene expression, literature, gene ontology (GO) annotations, gene networks, cellular and regulatory networks across all mycobacterial spe, as well as microRNAs and their target sites (Jia *et al.*, 2010). Another useful tool is Genes to Cognition database (G2Cdb) that houses sets of genes and biochemically isolated (from mammalian synapse) or experimentally elucidated proteins (Croning *et al.*, 2009). It links genes, proteins, (patho)physiology, anatomy and behavior across species. Recently, a manually curated database Nervous System Disease NcRNAome Atlas (NSDNA) was established that provides comprehensive experimentally supported associations about nervous system diseases (NSDs) and noncoding RNAs (ncRNAs) (Wang *et al.*, 2017).

The databases/tools dedicated for diabetic research and study are T1DBase, T2D-Db, Type 2 Diabetes Knowledge Portal and Islet Regulome Browser (Table 2). T1DBase is a web server of molecular genetics to study susceptibility and pathogenesis of type-1 diabetes (T1D) (Smink *et al.*, 2005). It covers annotated genome sequence for human, rat and mouse; information on genetically identified T1D susceptibility regions in human, rat and mouse, and genetic linkage and association studies pertaining to T1D; the Beta Cell Gene Expression Bank, which reports expression levels of genes in beta cells under various conditions, and annotations of gene function in beta cells; data on gene expression in a variety of tissues and organs; and biological networks across all mycobacterial spe from KEGG and BioCarta. It also contains tools like GBrowse (genome browser), site-wide context dependent search, Connect-the-Dots (for connecting gene and other identifiers from multiple data sources), Cytoscape (for visualizing and analyzing biological networks), and the GESTALT workbench (for genome annotation). On the other hand, T2D-Db (Type 2 diabetes database) provides integrated data on molecular components involved in the pathogenesis of T2D of human, mouse and rat (Agrawal *et al.*, 2008). It also contains information on candidate genes, SNPs (Single Nucleotide Polymorphism) in candidate genes or candidate regions, Genome-Wide Association Studies (GWAS), tissue specific gene expression patterns, EST (Expressed Sequence Tag) data, expression information from microarray data, pathways, protein-protein interactions and disease associated risk factors. Similarly, 'Type 2 Diabetes Knowledge Portal' is a repository of DNA sequence, functional and epigenomic information, and clinical data on T2D and its macro- and microvascular complications, and creating analytic tools to analyze these data (see "Relevant Websites section"). Islet Regulome Browser is a tool for exploring pancreatic islet epigenomic and transcriptomic data and provides interactive access to GWAS variants, different classes of regulatory elements, together with enhancer clusters, stretch-enhancers and transcription factor binding sites in pancreatic progenitors and adult human pancreatic islets (Mularoni *et al.*, 2017).

The databases involved in research of respiratory diseases are IGDB.NSCLC, LGEA, MyMpn and SEGEL (Table 2). IGDB.NSCLC is an integrated genomic resource for non-small cell lung cancer (Kao *et al.*, 2012). It covers aberrant expressed genes and microRNAs, somatic mutations and experimental evidence and clinical information of non-small cell lung cancer patients. In contrast, LGEA (Lung Gene Expression Analysis) is a gene expression database for lung diseases (Du *et al.*, 2017). It provides different tools such as LungGENS (for single cell transcriptomes analysis), LungSortedCells (for sorting lung cell population), LungDTC (for development time course analysis) and LungDisease (for disease analysis). MyMpn is an online resource for studying the human pathogen *Mycoplasma pneumonia* and hosts data obtained from gene expression profiling experiments, gene essentiality studies, protein abundance profiling, protein complex analysis, metabolic reactions and network modeling, cell growth experiments, comparative genomics and 3D tomography (Wodke *et al.*, 2015). Concurrently, a distinct tool SEGEL (Smoking Effects on Gene Expression of Lung) for visualizing smoking effects on human lung gene expression was developed (Xu *et al.*, 2015). It integrates expression microarray data sets from trachea epithelial cells, large airway epithelial cells, small airway epithelial cells and alveolar macrophages.

In addition to disease-specific tools, there are several general tools to integrate and analyze association between gene and disease progression such as DisGeNet, DistiLD, Imprinted Gene Catalogue, LncRNADisease, and 'Organ System Heterogeneity DB' (Table 2). DisGeNet integrates expert-curated databases with text-mined data, covers information on Mendelian and complex diseases and includes data from animal disease models (Pinero *et al.*, 2015). It features a score based on the supporting evidence to prioritize gene-disease associations. In comparison, DistiLD focuses on disease-associated SNPs and genes in their chromosomal context (Palleja *et al.*, 2012). Another useful tool is Imprinted Gene Catalogue that helps in examining parental origin trends for different types of spontaneous mutations (Glaser *et al.*, 2006). LncRNADisease is a database for long-non-coding RNA-associated disease diagnosis, treatment and prognosis (Chen *et al.*, 2013). In addition, 'Organ System Heterogeneity DB' displays distribution of phenotypic effects across organ systems along with the heterogeneity value (Mannil *et al.*, 2015).

Strategies and Scope of Translational Bioinformatics

Translational bioinformatics assists in bringing laboratory work to clinical world as well as helps in drug discovery, clinical research, surveillance of approved drugs and patient-specific therapy regimes.

Drug Discovery

Drug molecule development

The important aspects of drug molecule development include:

Molecular variation

The efficacy and safety of any new compound is first evaluated in an animal model under preclinical studies. We find that several of these new compounds show efficacy in animal models, but they fail in humans and vice-versa. The reason of this efficacy variation is molecular variation between humans and animals (Shanks *et al.*, 2009; Buchan *et al.*, 2011). The two models might have distinct molecular pathologies. Translational bioinformatics helps to analyze the molecular variation in data obtained from the animal models and humans, and helps to take decision whether further development should proceed or not (Cox *et al.*, 2011). The drugs which are efficacious during preclinical phase and get rejected in late clinical phase, increase the cost of drug development. Furthermore, the safety of drugs could be an issue, as molecule may be safe in animal but can be life threatening in case of human beings (Ioannidis, 2012). The cost of drug development has been increasing steadily due to drug failure, the major cause of which is lack of clinical efficacy as well as toxicity. As per the executive summary prepared by FDA in October 2017, the estimated cost of failed clinical trials ranges from \$800 million to \$1.4 billion. Translational bioinformatics can help in pre-analysis and ensure cost effectiveness and safe drug development.

Similarly considering the reverse case, several molecules which might be life saving for human beings, are discarded due to their failure in animal models, and thus are not studied in humans. So, translational bioinformatics may help in identifying drugs, which may be efficacious in humans.

Target identification/rational drug discovery

The discovery of drugs based on the knowledge of the target is termed as rational drug discovery. There are several inventive approaches to find out and characterize the target. The techniques of transcriptional bioinformatics, metabolic and signaling pathway analysis, knowledge of interacting proteins and information on enzyme activity are used as a protocol for target identification/rational drug discovery. Numerous bioinformatics tools/web servers (Table 1) are frequently used to find target structure and binding site information that can translate data into drugs.

Clinical research

Clinical research comprises of pharmacogenomics for optimal treatment and biomarker efficacy.

Pharmacogenomics for optimal treatment

The study of how genes affect individual's response to the drug is termed as pharmacogenomics, which could be studied as a part of precision medicine (Thorn *et al.*, 2010). The available genomic and phenomic data studied in relation with Electronic Medical Records (EMRs) and pharmacology help to evaluate safe and effective medicine and transplant for individuals.

Characterization of patient's genome by high throughput sequence analysis and comparative analysis with standard database gives an idea whether any gene is responsible for disease expression (Sarkar *et al.*, 2011). The inhibitor of that gene will only be effective in the patient who is affected due to overexpression of it, not by any other cause. The knowledge of pharmacogenomics helps in the selection of people with desired genetic composition for optimal treatment, with minimum side effects.

Biomarker (tracking) efficacy

A biomarker is a biomolecule that provides early diagnosis of a disease or its state of progression (Xia *et al.*, 2013). It is utilized to quantify drug efficacy, and identify drug targets and mechanism. Biomarkers that monitor specific physiological or pharmacological phenomenon can be used to select multiple therapeutic targets of a drug molecule (Kusumegi *et al.*, 2004).

Post-approval surveillance

The safety of drugs and repurposing drugs for new ailments are important considerations once drugs get approved for marketing.

Drug safety monitoring

To monitor safety of drugs, adverse drug reactions (ADRs) data are prepared. ADRs data can be easily obtained by transforming the big data (refers to electronic health records, administrative or health claims data, disease and drug monitoring registries) for pharmacovigilance system (Harpaz *et al.*, 2016). Sometimes, this data helps in finding new indications for existing drugs. A beneficial side effect could be considered as an additional therapeutic effect of the drug. For example, minoxidil, which is a potassium channel potentiator, is used as an antihypertensive drug, with the side effect of hypertrichosis (excessive growth of hair). This is considered as an additional therapeutic benefit and is used to control alopecia (hair loss or baldness).

Some ADRs are also observed in persons with specific genetic make-up. ADR databases help to identify the correct set of patients for a particular drug molecule. In addition, the analysis of these ADR database gives an idea about extra effects of the drugs and the pharmacological mechanism of drug action (Harpaz *et al.*, 2016; Abernethy *et al.*, 2011).

Drug repurposing

The reuse of already approved drugs for new diseases is termed as drug repositioning or repurposing. Since drug development and approval are costly and time consuming, this is an efficient way to bypass the early development phase and proceed directly with clinical trials for possible new diseases.

Translational bioinformatics strategies for repurposing involve finding new targets for approved drug molecules, by protein-protein, and drug-protein interactions, pathway analysis and similarity in gene expression during the progression of different diseases that may involve same molecular mechanism. Furthermore, high throughput screening of a drug's chemical structure against the various receptors allows finding new targets.

Patient-Specific Treatment Regimes

These include personalized medicine and real-time health monitoring systems.

Personalized medicine and pharmacogenomics

An individual genetic/molecular variation predicts disease susceptibility and responses to therapeutic agents. Biological data of people of various demographic conditions enable us to understand differential treatment approach required by individuals of different types of genes, lifestyle and environments (Reynolds, 2012). The data related to medical history of the patient is a key requirement for personalized medication. Apart from diagnostic reports, metagenomic analysis (environmental analysis) and genetic testing of patients are utilized to compare and analyze the class of medication that must be prescribed to the patient. The term 'personalized medicine' is used for group of people having similar gene type, lifestyle and environments, providing a basis for similar treatment strategies.

Although the term 'precision medicine' is relatively new, the concept has been a part of healthcare for more than a century. For instance, specific blood group of people can be transfused with only specific types of blood due to their antigen, which is result of their genetic composition. Presently, precision medicine is being intensively studied and, if future, could lead into new era of medicine.

Real-time health monitoring

To get the real time data of the individual medical aspects, various real time health monitoring wearable or implantable devices are available, for e.g., Fitbit Surge (heart rate, activity, sleep, and caloric burn), iHealth BP5 (blood pressure), iHealth glucometer (Blood glucose), Muse Headband (EEG), Scanadu Scanflo (urine analysis), Zephyr BioPatch (respiratory rate, ECG), Empatica E3 (photoplethysmograph, electrodermal activity). The data provided by these devices are known as Electronic Health Records (EHRs) or Electronic Medical Records (EMRs), which are integrated to clinical repositories or biobank by machine learning technology and predictive analysis offered by various tools and programs such as (i) eMERGE (see "Relevant Websites section"), a tool to combine DNA biorepositories with EMR systems for large scale, high-throughput genetic research in support of implementing genomics (Gottesman *et al.*, 2013a); (ii) CLIPMERGE PGx, a program for personalized medicine through EHRs and genomics-pharmacogenomics (Gottesman *et al.*, 2013b); (iii) i2b2 (see "Relevant Websites section"), integrating biology and bedside, a translational engine of patient's clinical data (Takai-Igarashi *et al.*, 2011).

EMRs are standard medical and clinical data gathered and stored electronically by health care provider. Analysis of these data with respect to available database helps to understand and identify the transition of the healthy state of the person to the diseased state, and the individual's disease progression (Shameer *et al.*, 2017). Most prominent databases linked with EMRs are (i) BioMe BioBank (see "Relevant Websites section"), a database of blood samples and health information, (ii) HarmoniMD (see "Relevant Websites section"), a repository of electronic clinical documents and cloud based EHR system, and (iii) HealthVault (see "Relevant Websites section"), a platform to store personal health information and generate insight from it.

Drug Interactions

Drug-drug interaction (DDI) can occur when two or more drugs are co-administered to a patient. It can lead to changed systemic exposure, resulting in variations in drug response of the co-administered drugs. DDIs generally occur due to inhibition of the metabolism for one drug by the other. It leads to a rise in plasma concentration of the drug whose metabolism is inhibited. The therapeutic index of the drug that has increased concentration in plasma could be decreased, leading to adverse reaction or increase in toxicity. The drug efficacy databases and DDI databases, play crucial roles during concomitant medication and avoid risk of adverse impact on the patient (Duke *et al.*, 2012). Evaluation of potential drug interactions also provides the idea about physiological pathways of its kinetics and dynamics.

Computational Tools Implicated in Drug Discovery

Multitude of biological and clinical databases have led to the development of tools for drug discovery (Table 3). It includes data integration software, virtual screening or structure activity relationship tool, docking, modeling or interaction visualization tool, disease progression or signaling pathway analysis tools.

Table 3 Drug discovery tools and web servers

Website	Key feature(s)	CI ^a	Access ^b	URL
CAVER analyst	For calculation, analysis and real time visualization of static or dynamic protein tunnel interaction	20.2	W	http://www.caver.cz/index.php
ConsensusPathDB	Tool to integrate protein-protein, genetic, metabolic, signaling, gene regulatory and drug-target and biochemical pathways interactions	24.1	W	http://cpdb.molgen.mpg.de/
Drug2Gene	For identifying gene targeting compound	5.4	W	http://www.drug2gene.com/
MLViS	Machine learning-based virtual screening tool	1.1	W	http://www.biosoft.hacettepe.edu.tr/MLViS/
Molsoft	Tools for cheminformatics, QSAR, mutation analysis and sequence analysis	- ^c	W	https://www.molsoft.com/index.html
MycoRegDB	For delineating regulatory/pathways in mycobacteria and in turn identify new drug targets for tuberculosis	1.8	W	http://compbio.iitr.ac.in/mycoregdb/
NNScore	For calculating receptor-ligand scoring function based on neural network	9.0	W	http://www.nbcr.net/software/nnscore/
PharmGKB	To find impact of genetic variation on drug response	21.4	W	https://www.pharmgkb.org/
PROMISCUOUS	For drug repositioning by integrating drug, target and signaling pathway	20.6	W	http://bioinformatics.charite.de/promiscuous/
SuperTarget	Web server to analyze drug-target interactions	32.0	W	http://insilico.charite.de/supertarget/index.php?site=home

^aCI, citation index (number of citations per year).^bW, Web based.^c-, no citations.

ConsensusPathDB, PharmGKB and PROMISCUOUS are database integration and analysis tools that integrate -omic (genomic, proteomic or metabolomics) data with signaling or biochemical pathway data to identify drug targets. ConsensusPathDB integrates and provides visualization of diverse functional interactions (protein – Protein interactions, biochemical reactions, gene regulatory interactions) and more than 1700 metabolic/biochemical pathways of human and provides visualization of interaction networks and substructures derived from the integrated interaction network (Kamburov *et al.*, 2009). On the other hand, MycoRegDB delineates regulatory pathways/networks across all mycobacterial species (Sharma *et al.*, 2009; Sharma and Surolia, 2013, unpublished data). This in turn will pave way for identification of new drugs for tuberculosis. In contrast, PharmGKB provides pathway and pharmacogene summary by integrating genotype, molecular as well as clinical knowledge and helps in investigating how genetic variation affects drug response (Thorn *et al.*, 2013). PROMISCUOUS is a database for understanding and prediction of off-target effects of drugs and allows a rational approach for drug-repositioning (von Eichborn *et al.*, 2011). It measures structural similarity for drugs and connects it to protein-protein interactions to establish and analyze networks responsible for its known side-effects. Based on this network-based approach, it helps in drug-repositioning. Another useful tool is MLViS that classifies molecules as drug-like and nondrug-like based on various machine-learning methods, including discriminant, tree-based, kernel-based, ensemble and other algorithms (Korkmaz *et al.*, 2015). Subsequently, it helps in performing virtual screening of drug like molecules against the target to find out active drug compounds. NNScore is a tool for scoring function (score is used during virtual screening) based on neural network for the characterization of protein-ligand complex (Durrant and McCammon, 2010). A distinct tool SuperTarget has also been developed that integrates drug-related information like medical indication, adverse drug effects, drug metabolism, pathways and Gene Ontology to find out drug-target relations (Gunther *et al.*, 2008). ‘CAVER Analyst’ is a graphic tool for interactive real-time visualization and analysis of tunnels and channels in static and dynamic protein structures (Kozlikova *et al.*, 2014). Molsoft contains a quantitative structure activity relationship (QSAR) tool, which helps in drug designing by providing the quantitative information about functional group’s position/type present in drug molecule (see “Relevant Websites section”). In addition, Drug2Gene is a database that combines the compound/drug-gene/protein information and integrates it to identify compounds targeting a given gene product or for finding all known targets of a drug (Roider *et al.*, 2014).

Discussion

Translational bioinformatics has opened new doors for churning out precious information from the existing knowledge of human health, disease progression mechanism, prophylactic conditions and human variation. The current challenges in healthcare are cost, ADRs, efficacy variance, late diagnosis and high rate of organ failures. These challenges can be addressed in the patient if the right medication can be administered at an appropriate time. Against this backdrop, it becomes important to understand the genetic makeup of an individual as well as the impact of environmental and demographic factors to provide right medication in sync with the requirement. The biological database helps us in this endeavor, provided we have scientific base for finding gene interactors and gene(s) responsible for disease progression and its molecular mechanism. Various relevant tools help us to understand the connection between genotype and phenotype via proteomic and metabolomics information.

On the other hand, the structure-function relationship provides information for drug designing, pharmacophore modeling and QSAR studies, thereby leading to drug discovery. Numerous molecules are now available in market or under development owing to this rational method of drug discovery rather than the traditional methods of study. It has not only reduced the cost of pharmaceutical industry significantly but also provided motivation for new findings or synthesizing analogs of higher efficacy and least toxicity. In addition, translational bioinformatics has proved the significance and utility of biomarkers by studying systemic interactions among genes, metabolites, tissue system, drugs and intermediate molecules, including environmental factors. Bio-markers information helps in timely diagnosis, and thereby ensures better treatment outcomes.

Similarly, the clinical information about patient's conditions, disease progression, adverse drug reactions (ADRs), pathological, diagnostics and wellness reports provide records for real time health monitoring by comparing and analyzing with the help of relevant tools and applications. It provides information about disease progression that helps to take required treatment at an appropriate time. Furthermore, the integration of these data help in the prediction of the rate of occurrence of any disease based on pollution condition or type, socioeconomic and geographic distribution. This helps to suggest further precaution or prophylactic action to the group of individuals. The total environmental exposure of human is termed as 'exposome', which indicates the disease susceptibility (e.g., lung cancer, skin cancer, asthma, allergy, baldness and deficient to specific micronutrient) based on exposure to the ambient environmental conditions (e.g., air-quality, smoke, radiations and specific nutrient chelators in food). Some devices are also there to measure exposome along with EMRs.

Summary

In the new era of data-driven healthcare evolution, translational bioinformatics is a crucial approach for drug discovery and development process. The innovative translational and computational analysis approaches are making a great impact on discovery, preclinical, clinical and post launch of a drug. The computational study saves millions in cost of drug development as compared to classical method. It gives rational approach, which reduces the chance of failure as compared to blind testing. It facilitates a better drug designing of improved efficacy since structural data of receptors gives fair idea about pharmacophore modeling and quantitative structure activity relationship (QSAR) studies. Additionally, the sequence data helps in generating model of unknown structures with the help of several tools, which are used to study its interaction with drug molecules. It also enables to understand and decipher the mechanism of drug molecules by analyzing the functional data and laboratory verification. The development of various databases in health care system is a huge resource for advancement and evolution of safe and efficacious medicinal system. As discussed above, the clinical data independently or in integration with biological data help to find out new drug or improved safety prescriptions. In addition, proper selection of patients during trial improves design of the trial and its success rate. Personalized medicine and real time health monitoring is a new era in the medical history, wherein huge efforts are being made. It has a great role in preventing life threatening disease like cancer where proper drug selection is still a challenge. Translational bioinformatics is an emerging field with lot of opportunities in medical advancement with development of algorithms, tools, programs and methods in order to find out significant information from available databases. Hopefully in the near future, translational bioinformatics with utilization of artificial intelligence will revolutionize medical landscape.

See also: Biological and Medical Ontologies: Disease Ontology (DO). Biological and Medical Ontologies: Human Phenotype Ontology (HPO). Clinical Proteomics. Comparative Transcriptomics Analysis. Disease Biomarker Discovery. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Identification and Extraction of Biomarker Information. Information Retrieval in Life Sciences. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics. Ontology in Bioinformatics. Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases

References

- Abernethy, D.R., Woodcock, J., Lesko, L.J., 2011. Pharmacological mechanism-based drug safety assessment and prediction. *Clin. Pharmacol. Ther.* 89 (6), 793–797.
- Agrawal, S., *et al.*, 2008. T2D-DB: An integrated platform to study the molecular basis of Type 2 diabetes. *BMC Genom.* 9, 320.
- Altman, R.B., 2012. Translational bioinformatics: Linking the molecular world to the clinical world. *Clin. Pharmacol. Ther.* 91 (6), 994–1000.
- Ashburner, M., *et al.*, 2000. Gene ontology: Tool for the unification of biology. *Nat. Genet.* 25 (1), 25–29.
- Bader, G.D., Cary, M.P., Sander, C., 2006. Pathguide: A pathway resource list. *Nucleic Acids Res.* 34 (Database issue), D504–D506.
- Baker, E.J., *et al.*, 2012. GeneWeaver: A web-based system for integrative functional genomics. *Nucleic Acids Res.* 40 (Database issue), D1067–D1076.
- Basu, S.N., Kollu, R., Banerjee-Basu, S., 2009. AutDB: A gene reference resource for autism research. *Nucleic Acids Res.* 37 (Database issue), D832–D836.
- Baxevaris, A.D., 2001. The molecular biology database collection: An updated compilation of biological database resources. *Nucleic Acids Res.* 29 (1), 1–10.
- Bertram, L., *et al.*, 2007. Systematic meta-analyses of Alzheimer disease genetic association studies: The AlzGene database. *Nat. Genet.* 39 (1), 17–23.
- Blekherman, G., *et al.*, 2011. Bioinformatics tools for cancer metabolomics. *Metabolomics* 7 (3), 329–343.
- Blum, R.L., 1982. Discovery, confirmation, and incorporation of causal relationships from a large time-oriented clinical data base: The RX project. *Comput. Biomed. Res.* 15 (2), 164–187.
- Buchan, N.S., *et al.*, 2011. The role of translational bioinformatics in drug discovery. *Drug Discov. Today* 16 (9–10), 426–434.
- Chatr-aryamontri, A., *et al.*, 2007. MINT: The Molecular INTERaction database. *Nucleic Acids Res.* 35 (Database issue), D572–D574.

- Chen, G., *et al.*, 2013. LncRNADisease: A database for long-non-coding RNA-associated diseases. *Nucleic Acids Res.* 41 (Database issue), D983–D986.
- Chen, X., Ji, Z.L., Chen, Y.Z., 2002. TTD: Therapeutic target database. *Nucleic Acids Res.* 30 (1), 412–415.
- Cox, B., *et al.*, 2011. Translational analysis of mouse and human placental protein and mRNA reveals distinct molecular pathologies in human preeclampsia. *Mol. Cell. Proteomics* 10 (12), doi:10.1074/mcp.M111.012526.
- Croning, M.D., *et al.*, 2009. G2Cdb: The genes to cognition database. *Nucleic Acids Res.* 37 (Database issue), D846–D851.
- Dai, H.J., *et al.*, 2013. T-HOD: A literature-based candidate gene database for hypertension, obesity and diabetes. *Database (Oxford)* 2013, bas061.
- Davis, A.P., *et al.*, 2009. Comparative toxicogenomics database: A knowledgebase and discovery tool for chemical-gene-disease networks. *Nucleic Acids Res.* 37 (Database issue), D786–D792.
- Dessailly, B.H., *et al.*, 2008. LigASite – A database of biologically relevant binding sites in proteins with known apo-structures. *Nucleic Acids Res.* 36, D667–D673.
- Duke, J.D., *et al.*, 2012. Literature based drug interaction prediction with clinical assessment using electronic medical records: Novel myopathy associated drug interactions. *PLOS Comput. Biol.* 8 (8), e1002614.
- Durrant, J.D., McCammon, J.A., 2010. NNScore: A neural-network-based scoring function for the characterization of protein-ligand complexes. *J. Chem. Inf. Model.* 50 (10), 1865–1871.
- Du, Y., *et al.*, 2017. Lung Gene Expression Analysis (LGEA): An integrative web portal for comprehensive gene expression data analysis in lung development. *Thorax* 72 (5), 481–484.
- Fiannaca, A., *et al.*, 2013. A knowledge-based decision support system in bioinformatics: An application to protein complex extraction. *BMC Bioinform.* 14 (Suppl 1), S5.
- Frenkel-Morgenstern, M., *et al.*, 2010. Dynamic proteomics: A database for dynamics and localizations of endogenous fluorescently-tagged proteins in living human cells. *Nucleic Acids Res.* 38 (Database issue), D508–D512.
- Glaser, R.L., Ramsay, J.P., Morison, I.M., 2006. The imprinted gene and parent-of-origin effect database now includes parental origin of de novo mutations. *Nucleic Acids Res.* 34 (Database issue), D29–D31.
- Gottesman, O., *et al.*, 2013a. The Electronic Medical Records and Genomics (eMERGE) network: Past, present, and future. *Genet. Med.* 15 (10), 761–771.
- Gottesman, O., *et al.*, 2013b. The CLIPMERGE PGx program: Clinical implementation of personalized medicine through electronic health records and genomics-pharmacogenomics. *Clin. Pharmacol. Ther.* 94 (2), 214–217.
- Grover, G., Sharma, D., unpublished data.
- Gunther, S., *et al.*, 2008. SuperTarget and matador: Resources for exploring drug-target relationships. *Nucleic Acids Res.* 36 (Database issue), D919–D922.
- Harpaz, R., DuMochel, W., Shah, N.H., 2016. Big data and adverse drug reaction detection. *Clin. Pharmacol. Ther.* 99 (3), 268–270.
- Healey, J., *et al.*, 2004. An open-source method for simulating atrial fibrillation using ECGSYN. *Comput. Cardiol.* 31, 425–427.
- Iannidis, J.P., 2012. Extrapolating from animals to humans. *Sci. Transl. Med.* 4 (151), 151ps15.
- Irwin, J.J., Shoichet, B.K., 2005. ZINC – A free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* 45 (1), 177–182.
- Jegga, A.G., *et al.*, 2007. PolyDoms: A whole genome database for the identification of non-synonymous coding SNPs with the potential to impact disease. *Nucleic Acids Res.* 35 (Database issue), D700–D706.
- Jia, P., *et al.*, 2010. SZGR: A comprehensive schizophrenia gene resource. *Mol. Psychiatry* 15 (5), 453–462.
- Kamburov, A., *et al.*, 2009. ConsensusPathDB – A database for integrating human functional interaction networks. *Nucleic Acids Res.* 37 (Database issue), D623–D628.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 28 (1), 27–30.
- Kann, M.G., 2010. Advances in translational bioinformatics: Computational approaches for the hunting of disease genes. *Brief. Bioinform.* 11 (1), 96–110.
- Kao, S., *et al.*, 2012. IGB.NSCLC: Integrated genomic database of non-small cell lung cancer. *Nucleic Acids Res.* 40 (Database issue), D972–D977.
- Katal, A., Wazid, M., Goudar, R.H., 2013. Big data: Issues, challenges, tools and good practices. In: *Proceedings of the 2013 Sixth International Conference on Contemporary Computing (Ic3)*, pp. 404–409.
- Kodzius, R., *et al.*, 2006. CAGE: Cap analysis of gene expression. *Nat. Methods* 3 (3), 211–222.
- Korkmaz, S., Zararsiz, G., Goksuluk, D., 2015. MLViS: A web tool for machine learning-based virtual screening in early-phase of drug discovery and development. *PLOS ONE* 10 (4), e0124600.
- Kozlikova, B., *et al.*, 2014. CAVER Analyst 1.0: Graphic tool for interactive visualization and analysis of tunnels and channels in protein structures. *Bioinformatics* 30 (18), 2684–2685.
- Kuhn, M., *et al.*, 2016. The SIDER database of drugs and side effects. *Nucleic Acids Res.* 44 (D1), D1075–D1079.
- Kusumegi, T., *et al.*, 2004. BMP7/ActRIIB regulates estrogen-dependent apoptosis: New biomarkers for environmental estrogens. *J. Biochem. Mol. Toxicol.* 18 (1), 1–11.
- Lill, C.M., *et al.*, 2012. Comprehensive research synopsis and systematic meta-analyses in Parkinson's disease genetics: The PDGene database. *PLOS Genet.* 8 (3), e1002548.
- Li, P., *et al.*, 2015. PubAngioGen: A database and knowledge for angiogenesis and related diseases. *Nucleic Acids Res.* 43 (Database issue), D963–D967.
- Liu, G., *et al.*, 2010. Computational analysis of microRNA function in heart development. *Acta Biochim. Biophys. Sin. (Shanghai)* 42 (9), 662–670.
- Liu, H., *et al.*, 2011. CADgene: A comprehensive database for coronary artery disease genes. *Nucleic Acids Res.* 39 (Database issue), D991–D996.
- Liu, T., *et al.*, 2007. BindingDB: A web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res.* 35 (Database issue), D198–D201.
- Mannil, D., *et al.*, 2015. Organ system heterogeneity DB: A database for the visualization of phenotypes at the organ system level. *Nucleic Acids Res.* 43 (Database issue), D900–D906.
- McCall, M.N., *et al.*, 2014. The gene expression barcode 3.0: Improved data processing and mining tools. *Nucleic Acids Res.* 42 (Database issue), D938–D943.
- Mi, H., *et al.*, 2005. The PANTHER database of protein families, subfamilies, functions and pathways. *Nucleic Acids Res.* 33 (Database issue), D284–D288.
- Mularoni, L., Ramos-Rodriguez, M., Pasquali, L., 2017. The pancreatic islet regulome browser. *Front. Genet.* 8, 13.
- Nim, H.T., *et al.*, 2015. CARFMAP: A curated pathway map of cardiac fibroblasts. *PLOS ONE* 10 (12), e0143274.
- Palleja, A., *et al.*, 2012. DistilD database: Diseases and traits in linkage disequilibrium blocks. *Nucleic Acids Res.* 40 (Database issue), D1036–D1040.
- Pinero, J., *et al.*, 2015. DisGeNET: A discovery platform for the dynamical exploration of human diseases and their genes. *Database (Oxford)* 2015, bav028.
- Raghupathi, W., Raghupathi, V., 2014. Big data analytics in healthcare: Promise and potential. *Health Inf. Sci. Syst.* 2, 3.
- Rahmati, S., *et al.*, 2017. pathDIP: An annotated resource for known and predicted human gene-pathway associations and pathway enrichment analysis. *Nucleic Acids Res.* 45 (D1), D419–D426.
- Ran, X., *et al.*, 2015. EpilepsyGene: A genetic resource for genes and mutations related to epilepsy. *Nucleic Acids Res.* 43 (Database issue), D893–D899.
- Reynolds, K.S., 2012. Achieving the promise of personalized medicine. *Clin. Pharmacol. Ther.* 92 (4), 401–405.
- Roider, H.G., *et al.*, 2014. Drug2Gene: An exhaustive resource to explore effectively the drug-target relation network. *BMC Bioinform.* 15, 68.
- Saeed, M., *et al.*, 2011. Multiparameter intelligent monitoring in intensive care II: A public-access intensive care unit database. *Crit. Care Med.* 39 (5), 952–960.
- Sagiroglu, S., Sinanc, D., 2013. Big data: A review. In: *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (Cts)*, pp. 42–47.
- Sam, S.A., *et al.*, 2017. XTalkDB: A database of signaling pathway crosstalk. *Nucleic Acids Res.* 45 (D1), D432–D439.
- Sarkar, I.N., *et al.*, 2011. Translational bioinformatics: Linking knowledge across biological and clinical realms. *J. Am. Med. Inform. Assoc.* 18 (4), 354–357.
- Schomburg, I., *et al.*, 2004. BRENDA, the enzyme database: Updates and major new developments. *Nucleic Acids Res.* 32 (Database issue), D431–D433.
- Sharma, D., Mohanty, D., Surolia, A., 2009. RegAnalyst: a web interface for the analysis of regulatory motifs, networks and pathways. *Nucleic Acids Res.* 37, W193–W201.
- Sharma, D., Surolia, A., 2013. Pathway targeting, antimycobacterial drug design. *Encyclopedia of Systems Biology*. Springer, pp. 1656–1659.
- Shameer, K., *et al.*, 2017. Translational bioinformatics in the era of real-time biomedical, health care and wellness data streams. *Brief. Bioinform.* 18 (1), 105–124.
- Shanks, N., Greek, R., Greek, J., 2009. Are animal models predictive for humans? *Philos. Ethics Humanit. Med.* 4, 2.

- Sharma, D., Priyadarshini, P., Vrati, S., 2015. Unraveling the web of viroinformatics: Computational tools and databases in virus research. *J. Virol.* 89 (3), 1489–1501.
- Sharma, D., Surolia, A., 2011. Computational tools to study and understand the intricate biology of mycobacteria. *Tuberculosis (Edinb.)* 91 (3), 273–276.
- Siramshetty, V.B., *et al.*, 2018. SuperDRUG2: A one stop resource for approved/ marketed drugs. *Nucleic Acids Res.* 46 (D1), D1137–D1143.
- Smink, L.J., *et al.*, 2005. T1DBase, a community web-based resource for type 1 diabetes research. *Nucleic Acids Res.* 33 (Database issue), D544–D549.
- Somek, M., Hercigonja-Szekeres, M., 2017. Decision support systems in health care - Velocity of Apriori algorithm. *Stud. Health Technol. Inform.* 244, 53–57.
- Sorani, M.D., *et al.*, 2010. Clinical and biological data integration for biomarker discovery. *Drug Dis. Today* 15 (17–18), 741–748.
- Szklarczyk, R., *et al.*, 2016. WeGET: Predicting new genes for molecular systems by weighted co-expression. *Nucleic Acids Res.* 44 (D1), D567–D573.
- Takai-Igarashi, T., *et al.*, 2011. On experiences of i2b2 (Informatics for integrating biology and the bedside) database with Japanese clinical patients' data. *Bioinformatics* 6 (2), 86–90.
- Tan, A.C., Gilbert, D., 2003. Ensemble machine learning on gene expression data for cancer classification. *Appl. Bioinform.* 2 (3 Suppl), S75–S83.
- Thorn, C.F., Klein, T.E., Altman, R.B., 2010. Pharmacogenomics and bioinformatics: PharmGKB. *Pharmacogenomics* 11 (4), 501–505.
- Thorn, C.F., Klein, T.E., Altman, R.B., 2013. PharmGKB: The pharmacogenomics knowledge base. *Methods Mol. Biol.* 1015, 311–320.
- von Eichborn, J., *et al.*, 2011. PROMISCUOUS: A database for network-based drug-repositioning. *Nucleic Acids Res.* 39 (Database issue), D1060–D1066.
- Wang, J., *et al.*, 2017. NSDNA: A manually curated database of experimentally supported ncRNAs associated with nervous system diseases. *Nucleic Acids Res.* 45 (D1), D902–D907.
- Wishart, D.S., *et al.*, 2008. DrugBank: A knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.* 36 (Database issue), D901–D906.
- Wishart, D., *et al.*, 2015. T3DB: The toxic exposome database. *Nucleic Acids Res.* 43 (Database issue), D928–D934.
- Wishart, D.S., *et al.*, 2018. HMDB 4.0: The human metabolome database for 2018. *Nucleic Acids Res.* 46 (D1), D608–D617.
- Wodke, J.A., *et al.*, 2015. MyMpn: A database for the systems biology model organism *Mycoplasma pneumoniae*. *Nucleic Acids Res.* 43 (Database issue), D618–D623.
- Xia, J., *et al.*, 2013. Translational biomarker discovery in clinical metabolomics: An introductory tutorial. *Metabolomics* 9 (2), 280–299.
- Xu, Y., *et al.*, 2015. SEGEL: A web server for visualization of smoking effects on human lung gene expression. *PLOS ONE* 10 (5), e0128326.
- Yan, Q., 2011. Toward the integration of personalized and systems medicine: Challenges, opportunities and approaches. *Personalized Med.* 8 (1), 1–4.
- Ye, H., *et al.*, 2011. HIT: Linking herbal active ingredients to targets. *Nucleic Acids Res.* 39 (Database issue), D1055–D1059.
- Zhang, L., *et al.*, 2012. ADHDgene: A genetic database for attention deficit hyperactivity disorder. *Nucleic Acids Res.* 40 (Database issue), D1003–D1009.
- Zhen, Y., *et al.*, 2007. CardioSignal: A database of transcriptional regulation in cardiac development and hypertrophy. *Int. J. Cardiol.* 116 (3), 338–347.

Relevant Websites

- <http://www.type2diabetesgenetics.org/>
ACCELERATING MEDICINES PARTNERSHIP (AMP).
- <http://icahn.mssm.edu/research/ipm/programs/biome-biobank>
BioMe.
- <https://clinicaldata.ema.europa.eu/web/cdp/home>
Clinical data.
- <https://www.physionet.org/physiotools/ecgsyn/>
ECGSYN.
- <https://emerge.mc.vanderbilt.edu/>
eMERGE.
- <http://about.harmonimd.com/>
HarmoniMD.
- <https://international.healthvault.com/in/en>
HealthVault.
- <https://www.i2b2.org/>
i2b2.
- <http://www.inovalon.com/howwehelp/cdr>
Inovalon.
- <https://www.liaison.com/healthcare-data-management/solutions/clinical-data-repository/>
LIAISON.
- <https://clinicaltrials.gov/>
NIH.
- <https://www.molsoft.com/>
Search ResultsMolsoft L.L.C.
- <http://data.hsl.virginia.edu/clinical-data-repository>
UVA Clinical Data Repository.

Translational Bioinformatics Databases

Onkar Singh, Institute of Information Science, Academia Sinica, Taipei, Taiwan, Republic of China and National Yang-Ming University, Taipei, Taiwan, Republic of China

Nai-Wen Chang, National Taiwan University, Taipei city, Taiwan, Republic of China and National Yang-Ming University, Taipei, Taiwan, Republic of China

Hong-Jie Dai, National Taitung University, Taipei, Taiwan, Republic of China

Jitendra Jonnagaddala, University of New South Wales, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Translational bioinformatics (TBI) is an emerging field with great potential to assist clinicians to achieve success in the advancement of personalized medicine. Translational bioinformatics (TBI) is very young field at the horizon of TBI comprises molecular bioinformatics, clinical informatics, genetics, biostatistics, health informatics and. The convergence of these fields into one makes TBI a prime component of biomedical research. The American Medical Informatics Association defined TBI as “the development of storage, analytic, and interpretive methods to optimize the transformation of increasingly voluminous biomedical data, and genomic data, into proactive, predictive, preventive, and participatory health” (see “Relevant Website section”). TBI integrates biological findings with the clinical information by using advanced computing environments and methodologies to bridge the gap between bench and bedside. TBI mainly focus on development of novel techniques for the integration of biological and clinical data and applying bioinformatics to translate the biological observations to practical application in a clinical setting. After the completion of the human genome project, clinicians acknowledge the potential use of the data generated, that it might give valuable information to know more about the basic mechanism of a human body. This accelerated the use of genome sequencing techniques resulting in the generation of a massive data that helps characterize the transcriptome, the proteome, and the metabolome (Altman, 2012). TBI not only created new knowledge but also gave new insights into underlying genetic mechanism of a disease (Londin and Barash, 2015). The four main foci of bioinformatics are clinical genomics, genomic medicine, pharmacogenomics, and genetic epidemiology (Fig. 1). These form the foundation of practices used by TBI to bridge the gap of research discoveries to clinical applications and their respective databases (Sarkar *et al.*, 2011).

In order to provide precision medicine-based therapies, heterogeneous data from clinical evaluations, bio-specimens and experimental investigations to characterize an individual patient’s disease progression need to be integrated. This requires systematic collection, standardized storage, and analysis of data (Jonnagaddala *et al.*, 2016, 2014). This integrated information often resides in several public and private databases. These comprehensive databases have become increasingly important among the researchers for hypothesis generation, validation, and verification in many areas such as drug discovery and clinical research (Zou *et al.*, 2015). The TBI rely on the four fundamental pillars of bioinformatics i.e., clinical genomics, genomic medicine, pharmacogenomics, and genetic epidemiology. These emerging fields laid down the foundation of approaches used in TBI to

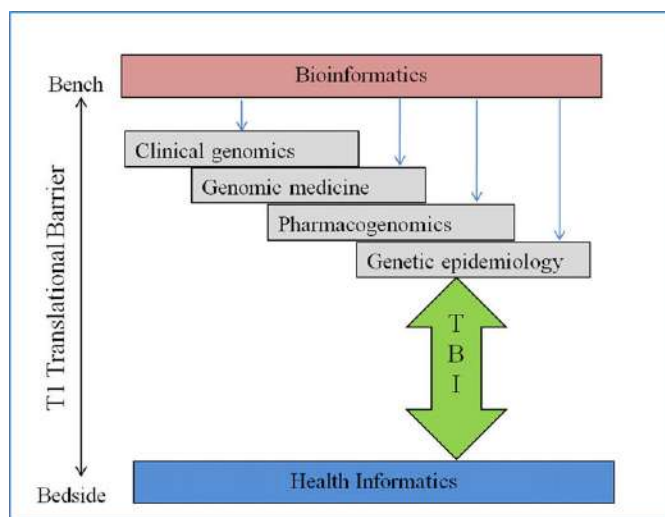


Fig. 1 Schematic illustration of four main areas of bioinformatics that laid the foundation of the approaches used by TBI to bridge the gap between bench and bedside.

bridge the gap between bench and bedside. This article discusses various TBI databases. These databases have already made a significant impact worldwide and continue to assist clinicians and other health care providers to implement personalized medicine. We also briefly discuss about the use of TBI databases in clinical genomics and in drug discovery and repurposing.

Translational Bioinformatics Databases

Over the decade, several TBI databases are developed to store biological findings, high-throughput data; scientific studies, published literature, and computational analysis. Most of the data stored in these databases contain evolving information which can give new insights to the researchers and help to facilitate the advancement of existing research. Here in this article we present few databases which are frequently used by diverse groups including from biologists to bioinformaticians.

Clinical Genomics Databases

Genomics is a study of the function, evolution, structure, and inheritance of the entire set of genetic material of an organism. Genomics involves the study of genes at all levels (e.g., DNA, mRNA, protein, cell, and tissue). The genomic studies become very efficient after the development of advanced technologies to facilitate high throughput sequencing and functional analysis. These technologies generate huge amounts of data that consist of DNA and protein sequence, data generated from various functional analysis and genetic mapping data. Most data are freely available for the public to access. Some of the popular genomic databases are: Clinical Genomic Database (CGD) (Solomon *et al.*, 2013) is a valuable resource of known genetic conditions contains the data of single gene alteration. COXPRESdb (Obayashi *et al.*, 2012) provides co-expression relationship for several animal species. COXPRESdb is a resource to discover a function of new gene candidates or to identify functional modules in metabolic and signaling pathways. Another popular database is “Database of genetic variants” (DGV) (MacDonald *et al.*, 2014) provides comprehensive information of structural variant in the human genome (healthy controls). dbVar is a publicly accessible, and freely available database that is developed and maintained by the National Center for Biotechnology Information (NCBI). The dbVar database is slightly different from DGV, as it also contains structural variation information for human but also includes clinically relevant structural variation data from ClinVar. The human genome database (GDB) is a public repository mainly focus on human genes, clones, polymorphism, and STSs (Letovsky *et al.*, 1998).

Genomic Medicine Databases

Genomic medicine is an emerging medical field which necessitates the use of an individual’s genomic information to advance preventive treatment strategies and tailor-made treatments. Genomic medicine is adequate to make a mark on the personal healthcare by understanding the underlying mechanism of a disease. Here we will briefly describe several web resources that fall under the genomic medicine category. Cancer genome interpreter (CGI) is a web interface mainly developed to identify validated oncogenic alterations and to anticipate cancer driver genes among mutations of unknown significance. This CGI also have a comprehensive catalogue of validated oncogenic alterations, biomarker of drug response, cancer genes, and cancer types (Tamborero *et al.*, 2018). ClinVar is a freely available archive which addresses the relationship of human variability and ascertained health condition with detailed interpretation knowledge (Landrum *et al.*, 2016). The database of curated mutation (DoCM) in cancer is an open resource with the collection of biological important cancer variants (Ainscough *et al.*, 2016). OncoKB, a precision-based oncology knowledge base provides evidenced based information of individual somatic mutation and structural alterations (Chakravarty *et al.*, 2017). Like OncoKB, Precision Medicine Knowledge Base (PMKB) is also precision-based knowledge base valuable resource for cancer variants and represented the interpretations in structured manner. This knowledge base is freely accessible which allows the user to submit interpretations as well as modify existing information (Huang *et al.*, 2017). Personalized cancer therapy (PCT) is a knowledge base for precision oncology that was developed to suggest a potential therapy to patients and clinicians based on particular tumor biomarkers.

Pharmacogenomics Databases

The pharmacogenomics is a combination of pharmacology and genomics where genomic information is used to examine the individual response to a drug agent. Pharmacogenomic research and discovery strongly assist health practitioners and clinicians in determining the correct drugs and accurate dosage for each individual, which may prevent adverse drug reaction and helps in the advancement of drug therapy (Zhang *et al.*, 2015). In this section, we will describe most commonly used pharmacogenomic database. The PharmacODB (Smirnov *et al.*, 2018) is the largest collection of cancer pharmacogenomic studies. This database allows researchers to explore the information of a particular drug or cell line across publicly available datasets and compare the dose-response data for a specific cell line – drug pair from any of the studies included in the database. The pharmacogenomic knowledgebase (PharmGKB) is a curated collection of genetic variants, drug labels, genes, drugs and their associations and relationships (Whirl-Carrillo *et al.*, 2012). The Clinical Pharmacogenetics Implementation Consortium (CPIC) is an international consortium formed in 2009 with the aim to address the implementation of genetic test results into an actionable form to make decision for affected drugs. This web resource is the collection of updated, detailed gene/drug clinical practice guidelines. The

DrugBank is a freely available resource containing drug and drug target related information. This database is full of both bioinformatics and chemo-informatics resources. The SCAN database is the collection of genetic and genomic data with multiple algorithms that can be used in data mining. This database consists two categories of SNPs: (1) Physical-based annotation and (2) functional annotation. CTDBase curates scientific data that describes relationships among chemicals/drugs, genes/proteins, diseases, taxa, phenotypes, GO annotations, pathways, and interaction modules. The primary goal of CTDBase is to enhance the understanding of the effects of environmental chemicals on human health on the genetic level, a field called toxicogenomic (Davis *et al.*, 2017). The drug gene interaction database (DGIdb) provides information of candidate genes or drugs against the known and potentially druggable genome (Cotto *et al.*, 2018).

Genetic Epidemiology Databases

Genetic factors or genes have all detailed information of our body; it has a crucial role in maintaining health and causing disease. Genetic epidemiology is the study of the role of genetic factors in our body for the determination of inherited disease among families. Such study can lead us to understand the underlying mechanism of genetic factors. In this topic we will discuss several genetic epidemiology databases include neuropsychiatric disorders, infectious diseases, breast cancer, kidney disease, cardiovascular disease, diabetes and aging. Neuropsychiatric disorder de novo mutation database (NPdenovo) is a huge collection of new (de novo) mutations causing mental disorders (autism spectrum disorder, intellectual disability, epileptic encephalopathy, schizophrenia and unaffected siblings). All information is identified by using whole-exome sequencing (Li *et al.*, 2016). BRCA share is a largest database containing clinical BRCA gene variants responsible for breast cancer (Beroud *et al.*, 2016). T2D@ZJU is comprehensive knowledge base contained information of type 2 database. All information are categorized in three types retrieved from pathways database, Protein-Protein Interaction (PPI) database and research articles (Yang *et al.*, 2013). Type 2 diabetes genetic association database (T2DGADB) is developed to provide valuable information to the clinicians about the genetic risk factors, which plays vital role in the development of Type 2 diabetes (Lim *et al.*, 2010). The clinical database for kidney disease (CDKD) includes physiological data of kidney patients (Singh *et al.*, 2012). CardioGenBase, a comprehensive database for major cardiovascular disease (MCVDs) and causing genes. Allgene/protein information are collected from MCVD based literatures. This database provides valuable information to the clinicians to unveil novel disease mechanism (Alexandar *et al.*, 2015). Alzforum is an information resource aiming to help researchers to accelerate discovery and development of diagnostics and treatments for Alzheimer's disease and related disorders. This

Table 1 A list of few currently available TBI databases. The databases are categorized into four different categories as discussed in this article

Database	URL
Clinical Genomic Database	
Clinical Genomic Database	https://research.nhgri.nih.gov/CGD/
COXPRESdb	http://coxpresdb.jp/
Database of Genomic Variants	http://dgv.tcag.ca/dgv/app/home
dbVar	https://www.ncbi.nlm.nih.gov/dbvar/
GDB	http://www.gdb.org
Genomic Medicine Databases	
CGI	https://www.cancergenomeinterpreter.org/home
ClinVar	https://www.ncbi.nlm.nih.gov/clinvar/
DoCM	http://docm.info/
OncokB	http://oncokb.org/#/
PMKB	https://pmkb.weill.cornell.edu/
PCT	https://pct.mdanderson.org
Pharmacogenomics Databases	
PharmGKB	https://www.pharmgkb.org/
CPIC	https://cpicpgx.org/
Drug Bank	https://www.drugbank.ca/
SCAN	http://www.scandb.org/
CTDBase	http://ctdbase.org/
DGIdb	http://www.dgidb.org/
Genetic Epidemiology Database	
NPdenovo	http://www.wzgenomics.cn/NPdenovo/
BRCA share	http://umd.be/BRCA1/
T2D@ZJU	http://tcm.zju.edu.cn/t2d
CardioGenBase	http://www.CardioGenBase.com/
CDKD	http://www.cdkd.org/
Alzforum	https://www.alzforum.org/

information resource has several databases dedicated to Alzheimer's disease. A list of few currently available TBI databases are presented in [Table 1](#).

Recent Advances

Drug discovery is a process of designing potential medicines to stop or reverse the effects of the disease. This process starts after the diagnosis and characterization of symptoms affecting human life ([Xia, 2017](#)). Drug discovery needs a large investment, enormous time and effort. The average time frame to develop a drug is 10–17 years. Despite this, Food and Drug Administration (FDA) approved a very small number of drugs every year ([Jadamba and Shin, 2016](#)). Drug repurposing is an approach that addresses these issues to some extent. It takes advantage of the fact that approved drugs and many left alone compounds have already been tested in humans and detailed information of clinical and pharmacokinetic data is available ([Astin *et al.*, 2017](#)). The applications of TBI approaches are on a high demand in the pharmaceutical industry especially in drug discovery and drug repositioning aspects. By using the new knowledge created in the TBI databases, pharmaceuticals are addressing the identification and development of potential drugs ([Buchan *et al.*, 2011](#)). Several sophisticated computational tools are employed to analyze these databases that help in decision making in several aspects of drug discovery and development.

Conclusion

In summary, in this article we discussed the role of TBI databases and also presented mostly commonly used databases by diverse group of users, including but not limited to clinicians, biologists, clinical researchers and bioinformaticians. We provided an overview and some insights on how these TBI databases are used. The TBI databases are valuable resources available to generate, test or validate new hypothesis shiftily in a short amount of period. TBI databases continue to consistently contribute to the discovery and development of novel therapeutics and treatments. We believe in future there will be more TBI databases of high quality which integrates and brings together knowledge already acquired in the current TBI databases.

Acknowledgments

This article was prepared as part of the Translational Cancer research network (TCRN) research programs. TCRN is funded by Cancer Institute of New South Wales and Prince of Wales Clinical School, UNSW Medicine.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Data Storage and Representation. Genome Annotation. Genome Databases and Browsers. Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Models for Computable Phenotyping. Natural Language Processing Approaches in Bioinformatics. Networks in Biology. Standards and Models for Biological Data: FGED and HUPO

References

- Ainscough, B.J., *et al.*, 2016. DoCM: A database of curated mutations in cancer. *Nature Methods* 13 (10), 806–807.
- Alexandar, V., *et al.*, 2015. CardioGenBase: A literature based multi-omics database for major cardiovascular diseases. *PLOS ONE* 10 (12), e0143188.
- Altman, R.B., 2012. Translational bioinformatics: Linking the molecular world to the clinical world. *Clinical Pharmacology and Therapeutics* 91 (6), 994–1000.
- Astin, J.W., *et al.*, 2017. Chapter 2 – Innate immune cells and bacterial infection in zebrafish. In: Detrich, H.W., Westerfield, M., Zon, L.I. (Eds.), *Methods in Cell Biology*. Academic Press, pp. 31–60.
- Beroud, C., *et al.*, 2016. BRCA share: A collection of clinical BRCA Gene variants. *Human Mutation* 37 (12), 1318–1328.
- Buchan, N.S., *et al.*, 2011. The role of translational bioinformatics in drug discovery. *Drug Discovery Today* 16 (9), 426–434.
- Chakravarty, D., *et al.*, 2017. OncoKB: A precision oncology knowledge base. *JCO Precision Oncology* 1, 1–16.
- Cotto, K.C., *et al.*, 2018. DGIdb 3.0: A redesign and expansion of the drug–gene interaction database. *Nucleic Acids Research* 46 (D1), D1068–D1073.
- Davis, A.P., *et al.*, 2017. The comparative toxicogenomics database: Update 2017. *Nucleic Acids Research* 45 (D1), D972–D978.
- Huang, L., *et al.*, 2017. The cancer precision medicine knowledge base for structured clinical-grade mutations and interpretations. *Journal of the American Medical Informatics Association* 24 (3), 513–519.
- Jadamba, E., Shin, M., 2016. A systematic framework for drug repositioning from integrated omics and drug phenotype profiles using pathway-drug network. *BioMed Research International* 2016, 7147039.
- Jonnagaddala, J., *et al.*, 2014. Data sharing challenges and recommendations for human biorepositories: A systematic literature review. *The International Technology Management Review* 4 (2), 68–77.
- Jonnagaddala, J., *et al.*, 2016. Integration and analysis of heterogeneous or translational. *Nursing Informatics* 2016, 387.
- Landrum, M.J., *et al.*, 2016. ClinVar: Public archive of interpretations of clinically relevant variants. *Nucleic Acids Research* 44 (D1), D862–D868.
- Letovsky, S.I., *et al.*, 1998. GDB: The human genome database. *Nucleic Acids Research* 26 (1), 94–99.
- Li, J., *et al.*, 2016. Genes with de novo mutations are shared by four neuropsychiatric disorders discovered from NPdenovo database. *Molecular Psychiatry* 21 (2), 290–297.

- Lim, J.E., *et al.*, 2010. Type 2 diabetes genetic association database manually curated for the study design and odds ratio. *BMC Medical Informatics and Decision Making* 10 (1), 76.
- Londin, E.R., Barash, C.I., 2015. What is translational bioinformatics? *Applied & Translational Genomics* 6, 1–2.
- MacDonald, J.R., *et al.*, 2014. The database of genomic variants: A curated collection of structural variation in the human genome. *Nucleic Acids Research* 42 (Database issue), D986–D992.
- Obayashi, T., *et al.*, 2012. COXPRESdb: A database of comparative gene coexpression networks of eleven species for mammals. *Nucleic Acids Research* 41 (D1), D1014–D1020.
- Sarkar, I.N., *et al.*, 2011. Translational bioinformatics: Linking knowledge across biological and clinical realms. *Journal of the American Medical Informatics Association* 18 (4), 354–357.
- Singh, S.K., *et al.*, 2012. CDKD: A clinical database of kidney diseases. *BMC Nephrology* 13, 23.
- Smirnov, P., *et al.*, 2018. PharmacDB: An integrative database for mining in vitro anticancer drug screening studies. *Nucleic Acids Research* 46 (Database issue), D994–D1002.
- Solomon, B.D., *et al.*, 2013. Clinical genomic database. *Proceedings of the National Academy of Sciences of the United States of America* 110 (24), 9851–9855.
- Tamborero, D., *et al.*, 2018. Cancer genome interpreter annotates the biological and clinical relevance of tumor alterations. *Genome Medicine* 10 (1), 25.
- Whirl-Carrillo, M., *et al.*, 2012. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology and Therapeutics* 92 (4), 414–417.
- Xia, X., 2017. Bioinformatics and drug discovery. *Current Topics in Medicinal Chemistry* 17 (15), 1709–1726.
- Yang, Z., *et al.*, 2013. T2D@ZJU: A knowledgebase integrating heterogeneous connections associated with type 2 diabetes mellitus. *Database* 2013, bat052.
- Zhang, G., *et al.*, 2015. Web resources for pharmacogenomics. *Genomics, Proteomics & Bioinformatics* 13 (1), 51–54.
- Zou, D., *et al.*, 2015. Biological databases for human research. *Genomics, Proteomics & Bioinformatics* 13 (1), 55–63.

Relevant Website

<http://www.amia.org/applications/informatics/translational-bioinformatics>
American Medical Informatics Association (AMIA).

Electronic Health Record Integration

Hong Yung Yip, Nur A Taib, Haris A Khan, and Sarinder K Dhillon, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

An Electronic Medical Record (EMR) is a computerized system that contains a patient's health record generated in a medical practice. Its primary purpose is to integrate health care information to improve quality of care (Gunter and Terry, 2005). EMR systems are designed to store data accurately and to capture and reflect the state of a patient across time which includes records such as patient personal statistics and demographics, past medical history, diagnosis information such as laboratory test results, and surgery related information. The improved access of readily available healthcare information provides a well-coordinated, collaborative, and multidisciplinary care that contribute to better quality of healthcare delivery (Cutler, 2010). Traditionally, EMR data is designed and stored in relational database management system (RDBMS), since it offers a powerful declarative query language known as Structured Query Language (SQL) which is capable of handling complex queries. The limitations of the current implementation of EMR using relational model are presented in later section.

Data visualization is visual representation of data which is indispensable in order to convey message, facilitate understanding and discovery, make sense of complex data which assists users in analyzing and reasoning evidence (Aparicio and Costa, 2015). In the case of a clinical studies of EMR data, data visualization allows one to quickly identify areas requiring attention or improvement, identify relationships and patterns of a disease outbreak, pinpoint emerging trends, etc. (SAS, 2016).

Linked data is one of the use cases in visualization model. It is a method of publishing structured data so that it can be interlinked and become informative through semantic queries (Fig. 1). This technology has revolutionized the era of semantic web and defined a new way of how information can be traversed (Bizer *et al.*, 2009).

The primary objective of this study is to propose a novel way of visualizing and linking EMR data by developing a NoSQL graph database using Neo4j. With the same set of EMR data, this study also compares and contrasts the performance, in terms of query runtime and storage requirement between relational (Microsoft SQL Server) and NoSQL databases (Neo4j and MongoDB document-oriented database). Other criteria such as scalability, availability, consistency, data models and open source availability are also subjectively measured. Additionally, this study evaluates the differences between MS SQL Server and Neo4j in terms of schema design, ease of programming language, maturity and level of support, flexibility, and security to demonstrate how graph databases can provide a viable data storage and management alternative applicable to EMR systems.

Electronic Medical Record (EMR)

A system that handles operations and stores EMR records is referred to as an EMR system. Some of the benefits of an EMR system include efficient information sharing function to improve the value and quality of integrated health care to support development of health policies, medical education and advanced research (van der Linden *et al.*, 2009). However, despite its benefits, there are many obstacles and challenges in relation to EMR systems such as standardization of vocabulary, security, privacy and data quality. Nonetheless, Orfanidis *et al.* (2004) claimed that the one of the most critical issues with the current EMR systems is the ever expanding size of healthcare data. The exponential and uncoordinated growth of information systems in healthcare industry in terms of size and heterogeneity of data has resulted in no standardised data that makes, linking, retrieval and analysis difficult. Moreover, most EMRs are not customizable enough to cater to the information needs of researchers from different backgrounds. Fortunately, data visualization or visual analytics holds the potential to address the current information overload that is becoming more and more prevalent.

Data Visualization and Linked Data

Data visualization is the science of analytic reasoning which involves the study of the visual representation of data. It has revolutionized data sciences by enabling mapping, and visualization to allow better inferences of complicated data (Huang *et al.*, 2015). Proper visualization provides an alternative approach to illustrate potential data interconnectedness and relationships. Data visualization, despite having applied in various domains of healthcare (Huang *et al.*, 2015), remain absent from EMR data visualization using NoSQL graph model.

One of many use cases in visualization model involves Linked Data, which uses uniform resource identifiers (URIs), hypertext transfer protocol (HTTP) and resource description framework (RDF/JSON) technologies. Its importance include large scale integration of data on the Web (Bizer *et al.*, 2009). One of the best example of a large Linked Dataset is DBpedia, which utilizes the content in RDF format from Wikipedia to link to other datasets on the Web (W3C, 2015).

Table 1 Categories of NoSQL databases (Abramova and Bernardino, 2013)

Category	Description
Key-value store	Data are stored as key-value pairs or hashes. The unique keys are utilized to index and retrieve the information stored in values. Examples are Redis, Azure Table Storage and DynamoDB.
Document store	Document store is similar to key-value store, however, their values are typically documents in formats such as eXtensible Markup Language (XML) or JSON. This category offers great performance and horizontal scalability. The internal storage is similar to relational databases but more flexible due to the lack of schema. It is also great for storage since it does not store empty or null values.
Column-family	Data are stored in columns. Each key is associated with one or more columns. This category is suitable for data mining and analytic systems due to high scalability. Examples are Cassandra and HBase.
Graph database	Data are stored as graph in the form of nodes (objects) and edges (relationships). The primary goal is to visualize the relationships between nodes. Examples are Neo4j and InfoGrid.

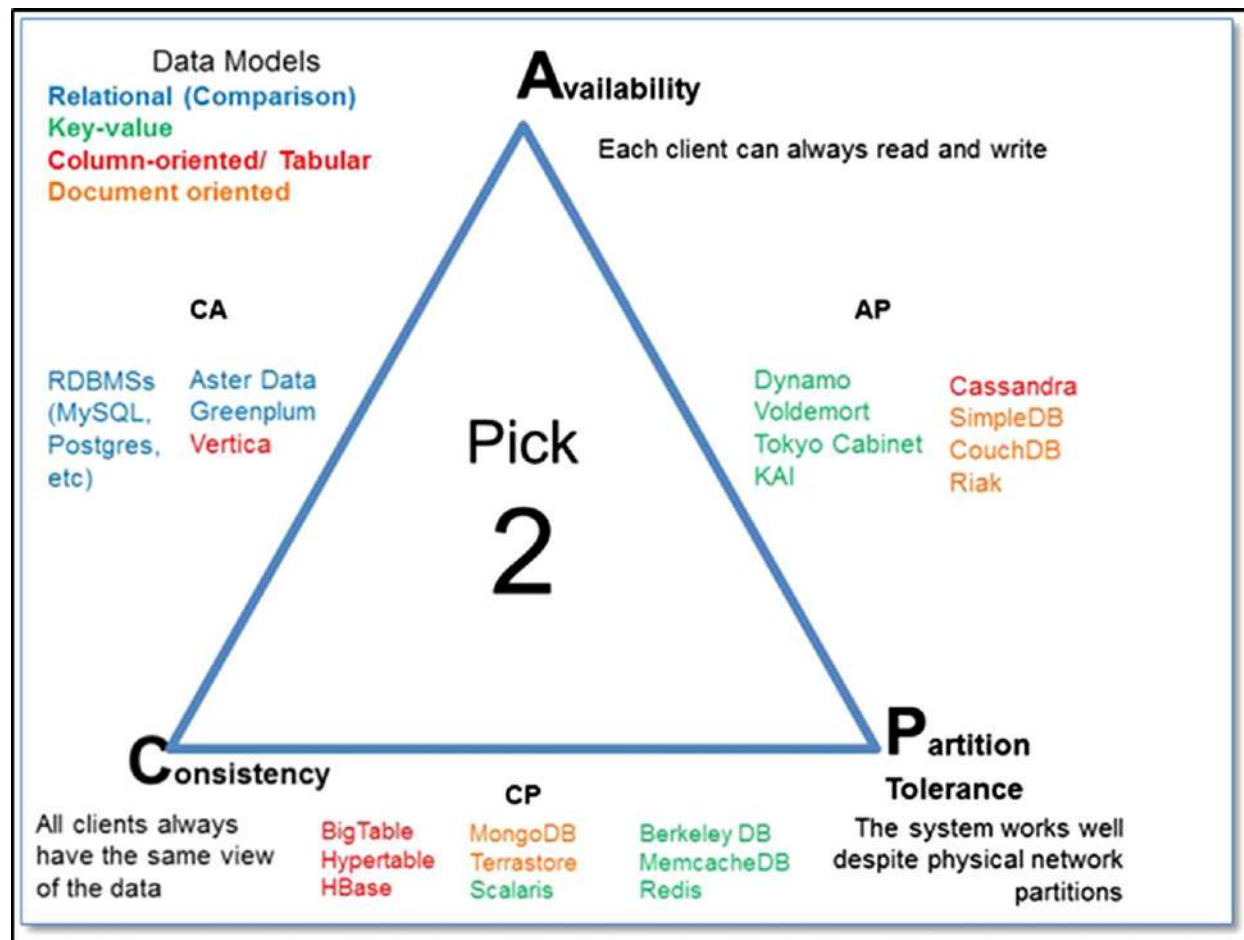


Fig. 2 Comparison of three main data model types used in NoSQL versus relational databases in terms of CAP Theorem. Reproduced from Ercan, M.Z., 2014. Evaluation of NoSQL databases for EHR systems. In: Proceedings of the 25th Australasian Conference on Information Systems, pp. 1–10. Auckland, New Zealand.

Potential benefits of NoSQL databases for EMR systems

Table 2 describes the main requirements of an EMR system and how newer NoSQL database system features can address these requirements to develop a distributed healthcare system.

NoSQL graph database

Graph databases enable queries that allow semantic web technology to function with the data defined and represented by nodes, edges and properties. They data retrieval from complex data structures easily as compared to relational systems (Neo4j, 2016a,b,c,d,e). Contrary to relational databases that operate with SQL language, the difficulty with graph

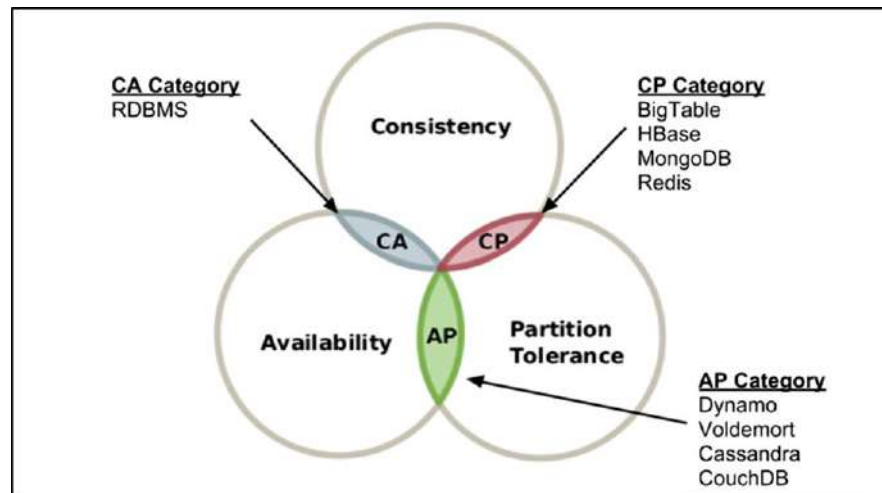


Fig. 3 Three categories of CAP Theorem. Reproduced from Ercan, M.Z., 2014. Evaluation of NoSQL databases for EHR systems. In: Proceedings of the 25th Australasian Conference on Information Systems, pp. 1–10. Auckland, New Zealand.

Table 2 Comparison of EMR requirements and corresponding NoSQL database features (Ercan, 2014)

EMR requirement	NoSQL database feature
Size of healthcare data expanded over time and eventually became a bottleneck for traditional EMR systems.	NoSQL databases are based on horizontal scalability which permits effortless and automatic scaling.
Increasing heterogeneity of healthcare data such as free-text notes, images and other complex data that are unstructured or semi-structured require new storage alternatives.	Flexible data models or schemas powered by NoSQL databases allow complex data to be stored easily.
Healthcare records should always be accessible for undisruptive and continuity of healthcare services.	NoSQL databases provide high availability due to its distributed nature and data replication.
Healthcare records are normally added, not updated.	Eventual consistency of NoSQL database architecture provides an acceptable EMR use cases.
Sharing of healthcare data requires concurrent access to EMR systems from multiple locations which requires a high-performance system to respond to data access and retrieval request in a timely manner.	Contrary to relational databases, NoSQL distributed databases offer higher performance that is capable of spanning data across server nodes, racks, or even multiple data centers with no single point of failure.
Implementation and infrastructure costs are high.	Most NoSQL database systems are open-source and can run on inexpensive commodity hardware architectures.

database is that no query language has ever been standardized. Most graph based applications make use of application programming interfaces (APIs) for querying, though there are some query languages available like SPARQL and Gremlin (Wood, 2012).

Property graph model

Graph databases employ nodes, edges, and properties for data representation (Fig. 4). A node can hold any number of attributes or properties or key-value-pairs. Edges represent graphs or relationships, which are the lines that connect nodes to other nodes to describe relationship between them (Fig. 5).

Neo4j is the implementation chosen to represent graph database. Sponsored by Neo Technology, it is an open-source NoSQL graph database implemented in Java and Scala for all non-commercial uses (Vicknair *et al.*, 2010). Neo4j is an embedded, disk-based, fully transactional Java persistence engine that stores data structured in graph rather than in tables. It implements the property graph model with native graph storage and provides full database characteristics including ACID transaction compliance, cluster support, and runtime failover, making it suitable for the development of EMR database (Neo4j, 2016a,b,c,d,e).

Database and Database Management Systems (DBMS)

A database by definition, is a set of data and definition of its organization. It is an electronic collection of schemas, tables, queries, reports, views and other objects. A general-purpose DBMS provides functionalities to define, create, query, update and administer databases. DBMSs are often classified according to the database model that they support such as relational and

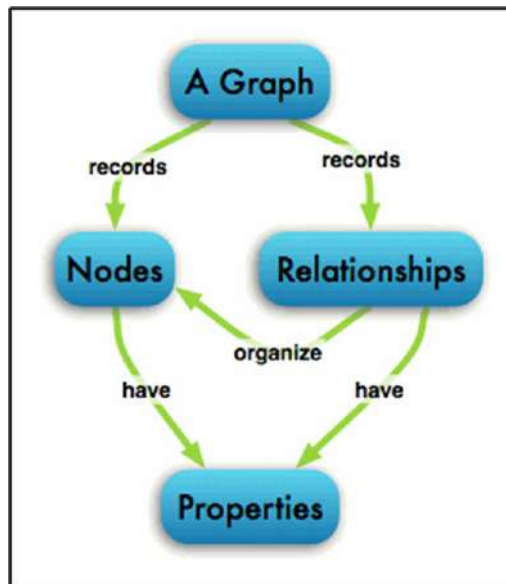


Fig. 4 The building blocks of the property graph model. Reproduced from Neo4j, 2016a,b,c,d,e. What is a graph database? Retrieved from Neo4j: <https://neo4j.com/developer/graph-database/>.

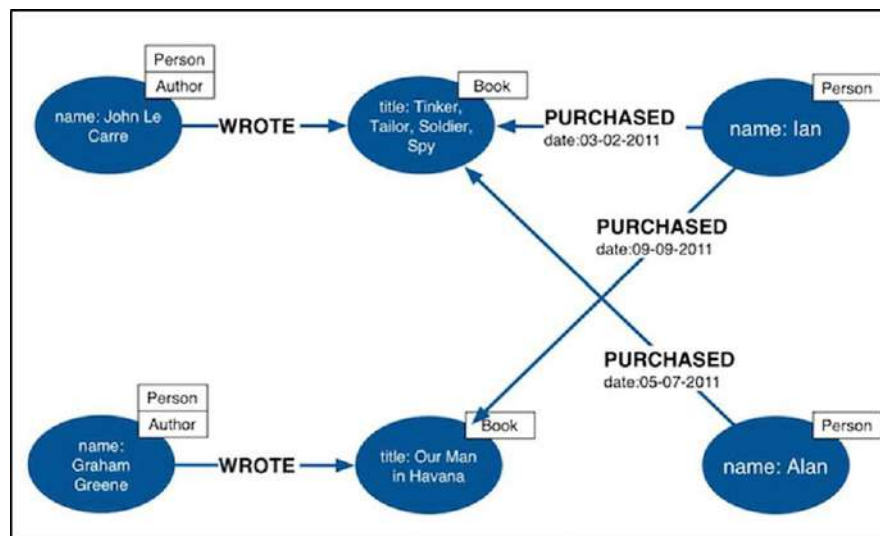


Fig. 5 Labeled property graph data model. Reproduced from Neo4j, 2016a,b,c,d,e. What is a graph database? Retrieved from Neo4j: <https://neo4j.com/developer/graph-database/>.

Table 3 Types of database languages (Beynon-Davies, 2003)

Language	Description
Data definition language (DDL)	Defines data types and their relationships.
Data manipulation language (DML)	Performs transactions such as insert, update or delete records.
Query language	Enables searching for information and computing derived information.

non-relational (NoSQL). Generally, a database is not backwards compatible through a range of different DBMSs. However, DBMS can be standardized to allow compatibility. Databases can be manipulated using one or more of the following languages (Table 3).

Case Study of a EMR Graph Model

EMR Document Database Development

The EMR NoSQL document-oriented database was developed using MongoDB. To convert and export the existing EMR relational database to MongoDB, a third party application, MongoDB SQL Server Importer (SQL2Mongo) (Tuldoklambat, 2012) was used to achieve such task. SQL2Mongo required the specification of the source destination and the target database, and then automatically converts SQL data types directly to JSON data types as JSON output. Subsequently, a database transaction was performed in MongoChef (MongoDB GUI) to ensure the imported database and tables were consistent with those created in MS SQL Server. A sample query was also executed to verify the database integrity and to ensure data reproducibility of both MS SQL Server and MongoDB.

EMR Graph Database Development

Contrary to MongoDB, to import data from MS SQL Server into Neo4j, it is essential to transform a relational database schema and model it as a graph prior implementation. Since Neo4j graph database revolves around property graph model, a graph schema (Figs. 6 and 7) was derived from the EMR relational model with the following guidelines (Neo4j, 2016a,b,c,d,e):

- Each *entity table* is a *label* on nodes
- Each *row* in the entity table is a *node*
- *Columns* on the entity table become node *properties*

Once the graph schema was established, EMR data and their appropriate tables were extracted and exported from MS SQL Server using the built-in SSMS data exporter interface in CSV format. Since Neo4j is a Shell-based application without GUI unlike MS SQL Server, all actions were performed using its native language, Cypher (Neo4j, 2016a,b,c,d,e). Cypher's LOAD CSV command was used to load the contents of the CSV file into Neo4j to generate a graph structure according to the derived graph schema (Figs. 6 and 7).

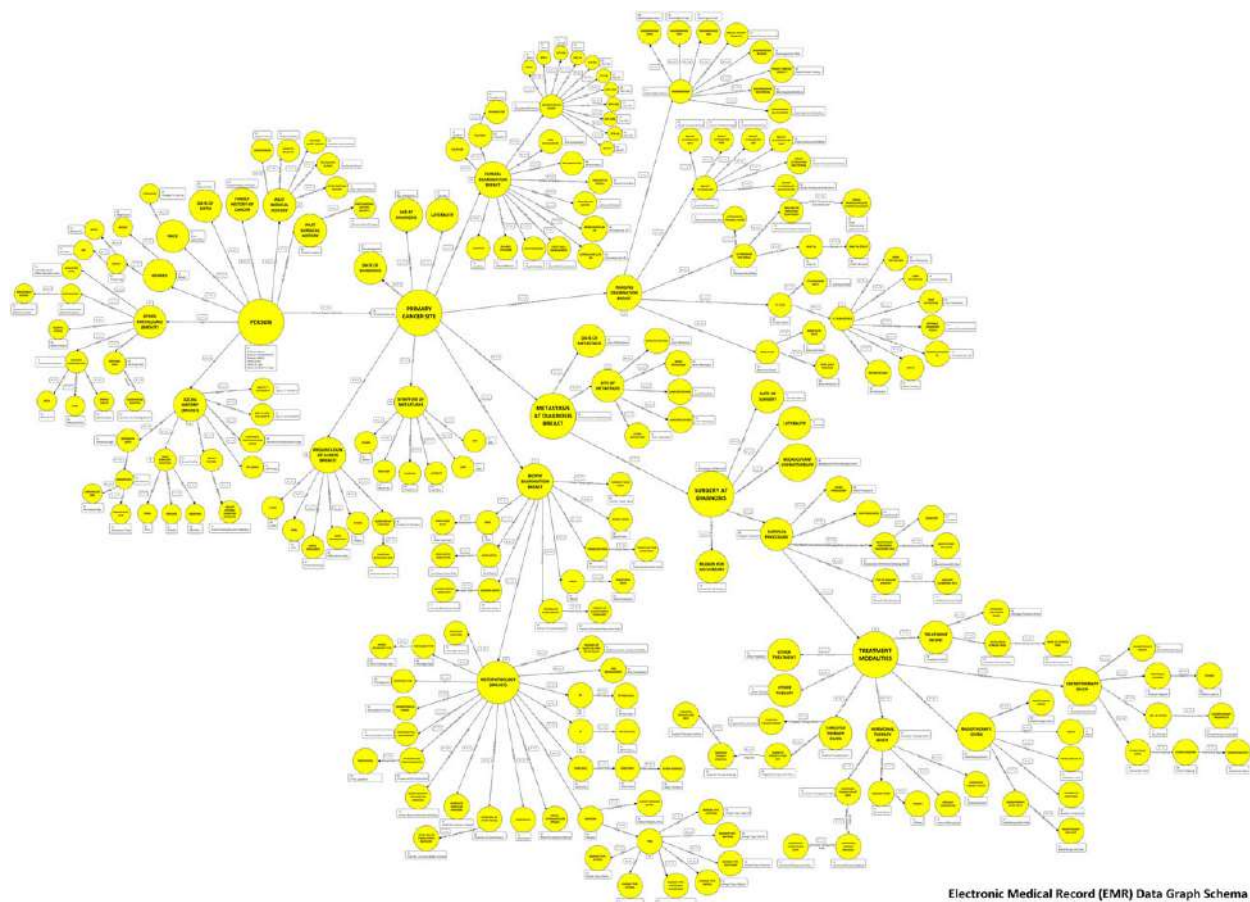


Fig. 6 Full graph schema of EMR database. A snapshot view of the red outlined graph schema is shown in Fig. 7.

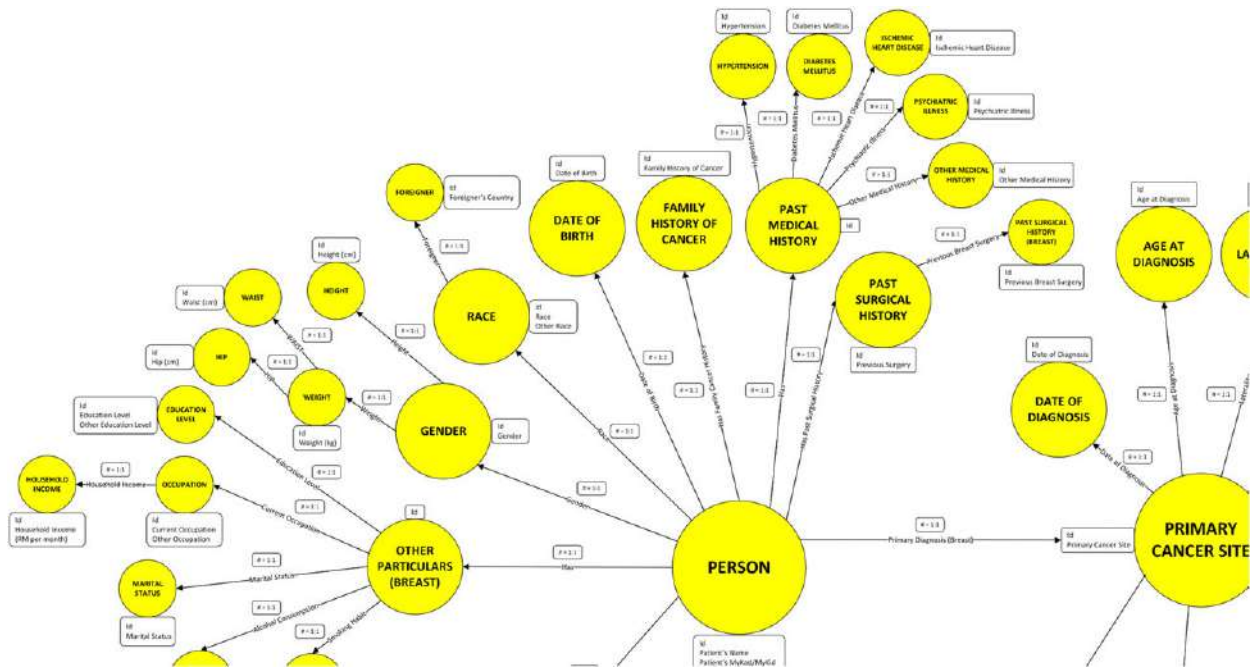


Fig. 7 Closer view of graph schema of EMR database.

The resulting graph with a total of 231 unique labels and 190 relationship types with 22,869 nodes, 22,869 relationships and 23,680 properties was generated.

Neo4j EMR Data Visualization

A single EMR record was fully visualized in Neo4j with all its adjacent relationships defined to confer semantics of EMR data. The full visualization (**Fig. 8(a–c)**) includes EMR data ranging from:

- Patient personal identifying information
- Patient family history of cancer
- Past medical and surgical history
- History of presenting illness
- Clinical examination information such as cancer location, size and description
- Imaging examination information such as mammogram, breast ultrasound, CT scan and bone scan
- Biopsy examination such as core biopsy and excision biopsy of tumor tissue
- Histopathology
- Diagnosis and metastasis information
- Surgical relation information
- Follow up therapy treatment records

Fig. 9 illustrates a simple visualization of how patient demographics information can be linked with past medical history to allow meaningful data representation.

Discussion

Data Visualization and Linked Data

In relational databases such as MS SQL Server, EMR data is retrieved by the users according to a set of defined criteria. In such cases, the results will be limited to only what the user intended to query. EMR data contain extensive amount of information about a patient and metadata on how healthcare is delivered. However, Graph databases such as Neo4j provide visual representations to facilitate effective visualization renders to assist users in analyzing (Aparicio and Costa, 2015). According to Fig. 8(a-c), EMR data were well presented as a graph consisting of nodes interconnected with relationships to confer semantics for sense-making and data analysis. Fig. 9 illustrated a simple visualization of how patient demographics data were linked with past medical history in Neo4j.

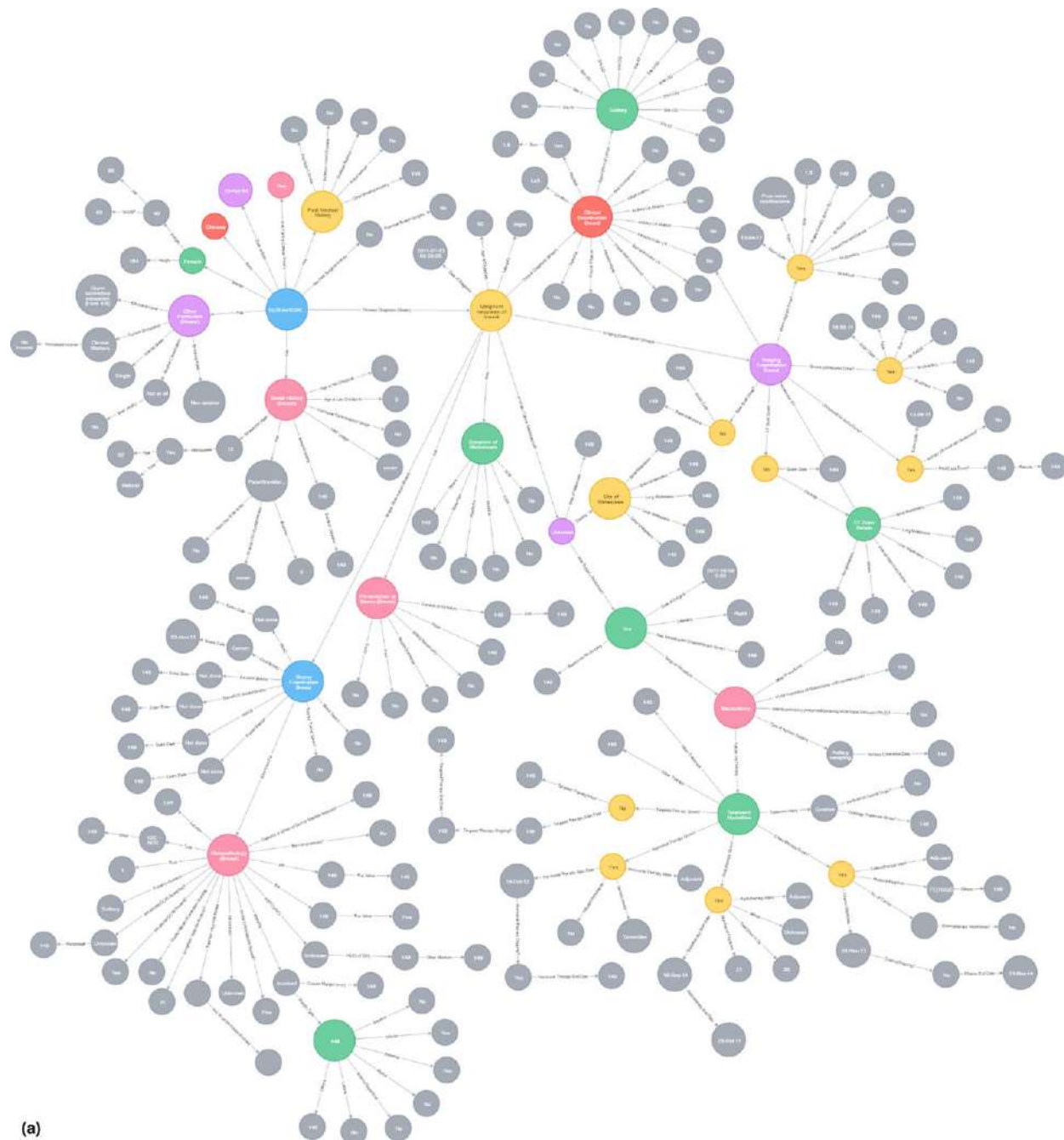


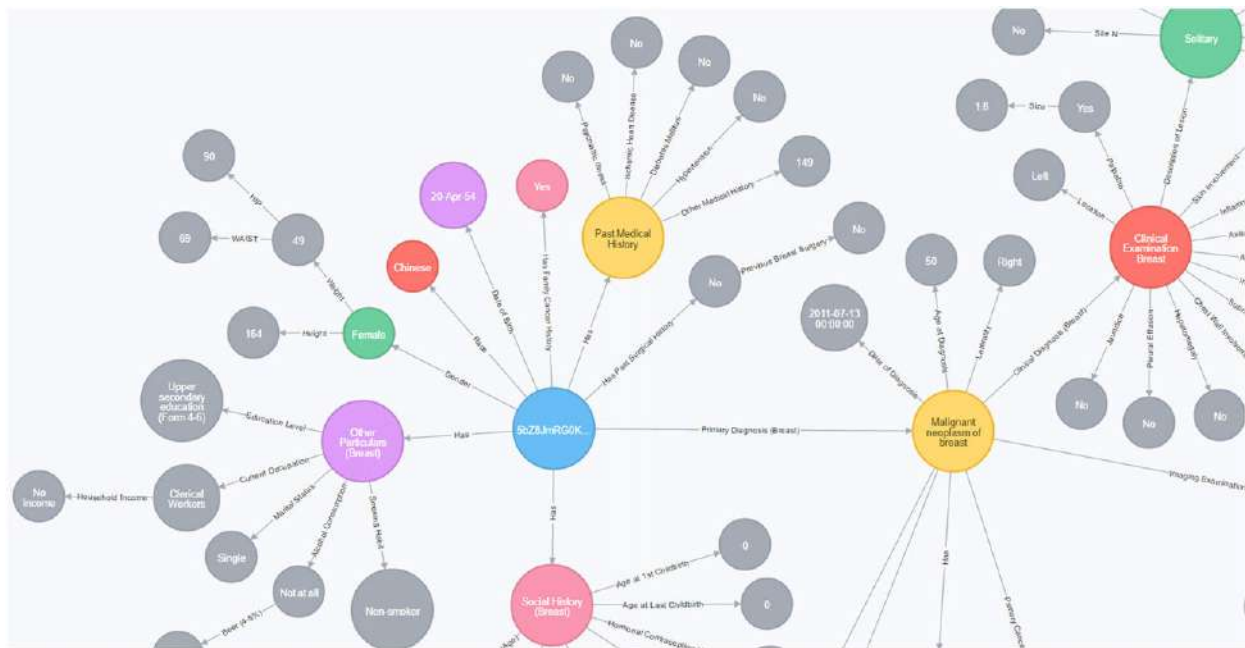
Fig. 8 (a) Full visualization of a single EMR record in Neo4j.

Relational Versus NoSQL Databases

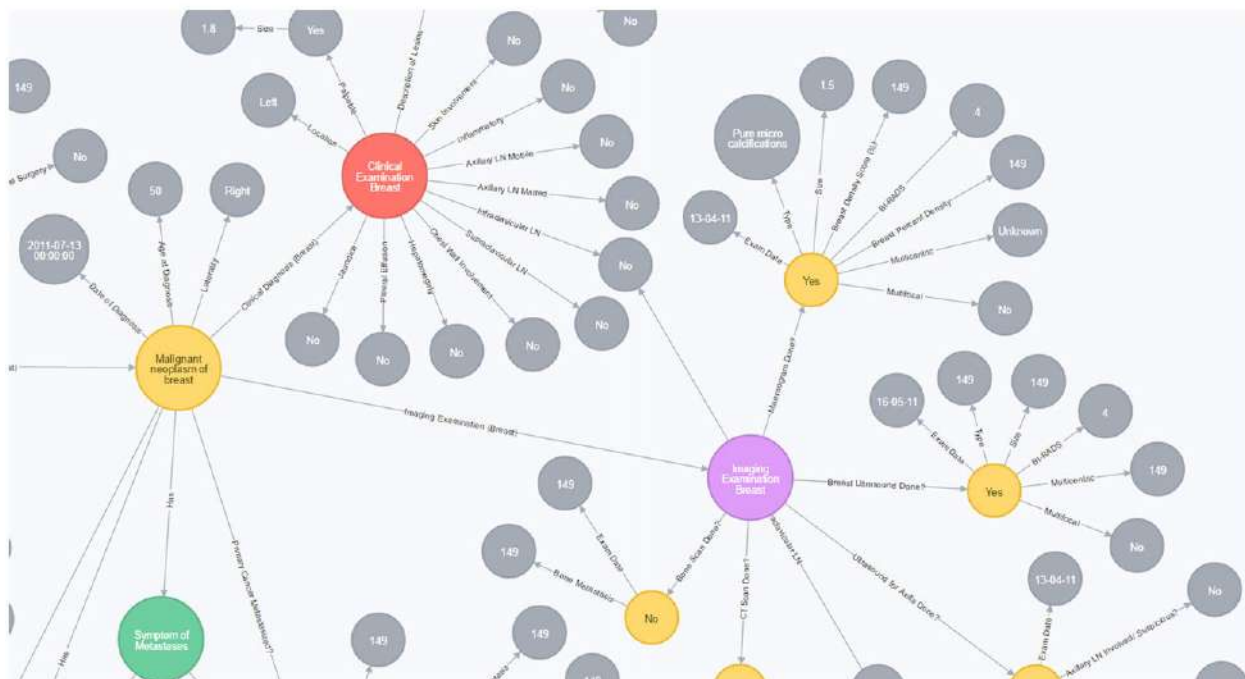
Performance of MS SQL server versus MongoDB

According to the results obtained, MongoDB was 4 times faster than MS SQL Server with an average execution time of 1ms to 4ms respectively in performing simple queries 1–4. Results were in agreement with comparative study conducted by [Parker et al. \(2013\)](#).

For the more complex query 5, involving multiple object types and nested queries, MS SQL Server had a runtime of 6 ms while MongoDB had a runtime of 11.75 ms. MongoDB was 2 times slower than MS SQL Server. MS SQL Server excels in join operations of relatively modest amount of structured data and SQL comes with native aggregate function like DATEDIFF(). In the present study, EMR records/ documents (Date of surgery) from one collection was first unwind to be copied/ joined to the second collection (Date of diagnosis), and subsequently underwent subtraction and division operations to calculate the time taken to surgery in days. This slows down the performance by a considerable amount. Unlike using SQL, MongoDB uses reference to locate



(b)



(c)

Fig. 8 (Continued) (b) The relationship between adjacent nodes in a record in Neo4j conferring semantics of EMR data. (c) Visualization of a record in Neo4j, showing the relationships between clinical symptoms and test carried out.

data in memory, additional decisions and aggregate functions such as lookup were required to determine how to implement or join relationships between collections, hence the slower performance (Parker *et al.*, 2013).

Performance of MS SQL server versus Neo4j

According to our results, MS SQL Server was 2 times faster on average compared to Neo4j with a mean execution time of 4.5–8.5 ms in performing simple queries 1–4. Results were in agreement with study by Vicknair *et al.* (2010). With regard to performance, Neo4j appears to have lower optimization features around querying for null values since it does not store empty or null values

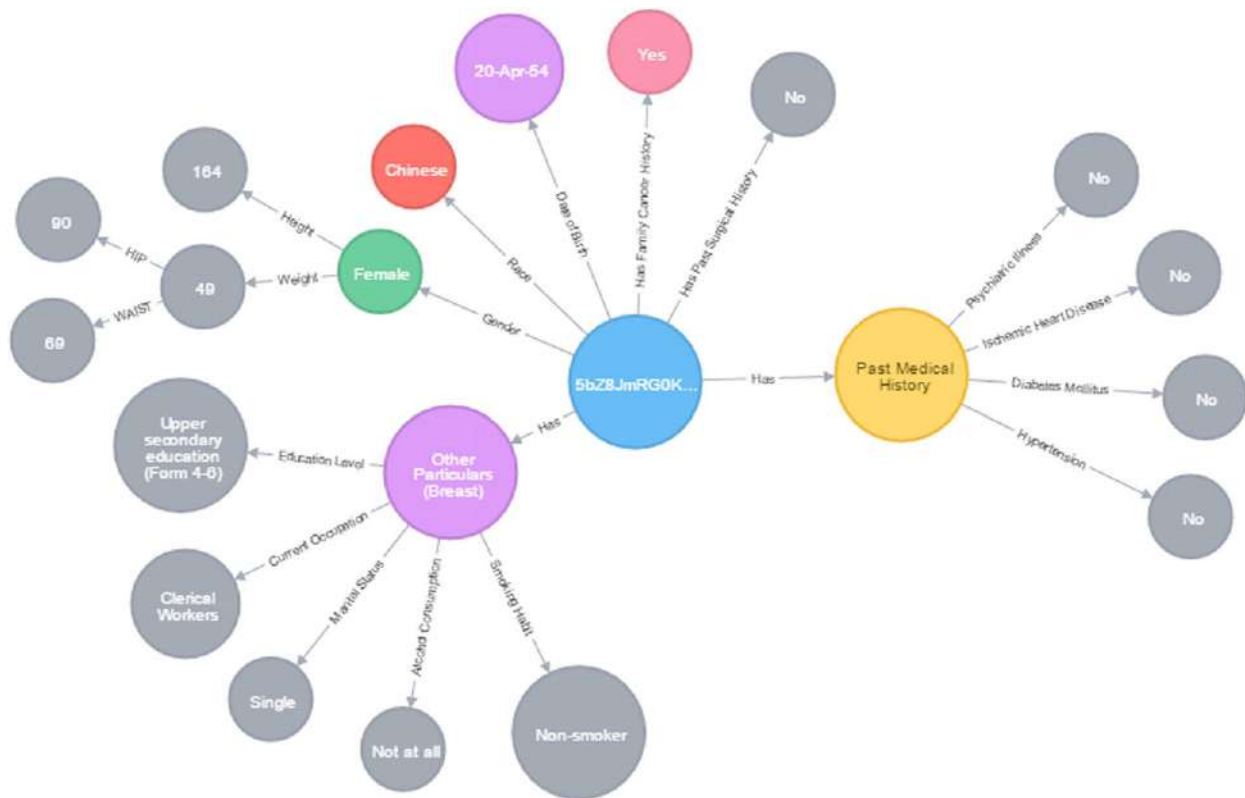


Fig. 9 Simple visualization of patient demographics linked with past medical history in Neo4j.

natively (Neo4j, 2015). Query 1–4 involve conditions using CASE statement with SUM aggregate function to search through each and every record (total 99) to query for the total number of breast cancer cases and surgeries performed respectively. However, the Patient records retrieved in this case study were not fully populated and contain some empty values, hence the slower performance from Neo4j.

Since query test 1–4 were performed on default configurations, theoretically speaking, options such as memory allocation can be adjusted, tuned and optimized to improve input/output (I/O) and read/write performance. According to Vicknair *et al.* (2010), relational database performed better than Neo4j at small scale, but scaling upward dramatically shifted the search times in favor of Neo4j.

For the more complex query 5 which involves date and time notions, MS SQL Server had a runtime of 6 ms, whereas the test was not applicable to Neo4j as Neo4j currently does not have a native date/time/timestamp value type (Neo4j, 2015). Therefore, aggregate function to determine the date/time difference was not feasible in Neo4j. Neo4j is considerably new with only seven years of production. It is still undergoing active development.

Storage requirement of MS SQL server versus MongoDB

Database storage comprises of the internal (physical) level in the database architecture as well all the information needed such as metadata to construct the conceptual and external level. According to the results obtained, with the identical set of EMR data, MongoDB was 51 times more space-efficient than MS SQL Server, occupying only 0.50 MB in disk space compared to 25.50 MB in MS SQL Server. This can be attributed to differences in storage engine and the way the data is stored in MS SQL Server and MongoDB (Nguyen, 2010).

The storage engine manages the nature of how the data is stored in a database. MS SQL Server uses SYBASE storage engine. It is a full-featured engine but is generally slower due to its ACID compliance for reliability (VCU, 2013). Most of the data are stored in disk which explains the higher storage requirement. Read and write speed can also be bottlenecked by the performance of disk I/O. On the contrary, MongoDB comes with two supported storage engines: MMAPv1 (Memory Mapped Version 1) engine and WiredTiger storage engine (MongoDB, 2016a,b,c,d,e,f). MMAPv1 is MongoDB's original storage engine based on memory mapped files. It prioritizes memory usage over disk as its cache to read and write documents. In addition, WiredTiger is a newer storage engine that scales on modern multi-CPU architectures. It offers document level concurrency control with native compression to minimize on-disk overhead and I/O, which significantly reduces the amount of storage required by up to 80%, unlocking new infrastructure efficiency and cost savings (MongoDB, 2016a,b,c,d,e,f). Hence, the observed lower disk space consumption by MongoDB compared to MS SQL Server.

Storage requirement of MS SQL server versus Neo4j

According to the results obtained, Neo4j was 1.35 times more space-efficient than MS SQL Server, consuming only 18.87 MB to 25.50 MB in disk space. Likewise, the observed difference can be attributed to the variation in storage engine and the way how the data is stored.

Neo4j storage engine is powered by Neo Technology, which behaves similarly to MongoDB, where it prioritizes memory usage rather than disk with newer compression engine to read and write data. However, Neo4j is actually a fully transactional database with ACID compliance ([Neo4j, 2016a,b,c,d,e](#)), which explains the higher disk space usage compared to MongoDB.

Performance and storage scalability

The requirement of scalability for EMR systems is considered restrictive as most of the current systems are based on relational databases which are not designed to scale, which calls for a need of newer scalable database systems to maximize output at the expense of lower storage requirement ([Jin et al., 2011](#)). Based on the objective tests (performance and storage size) conducted in this study, it was noticeable that NoSQL databases such as MongoDB and Neo4j performed better with a lower disk space requirement compared to MS SQL Server.

To scale is to either increase hardware capacity or increase the amount of data stored per capacity. For an example based on results in, 25.5 MB was required to store 99 records in MS SQL Server, whereas 99 records occupied only 0.50 MB and 18.87 MB in MongoDB and Neo4j respectively. Therefore, with an effective similar storage size of 25.5 MB, MongoDB and Neo4j can theoretically store up to 5049 and 135 records, a 51 and 1.35 time increased in the amount of data stored per same capacity respectively due to newer storage and compression technology.

On the contrary, NoSQL database systems are based on shared-nothing approach where servers have their own dedicated resources such as RAM, processor or storage, allowing scaling horizontally ([Cattell, 2011](#)).

Availability

[Schmitt and Majchrzak \(2012\)](#) suggested that the nature of EMR data holds critical patient records which therefore, requires the prerequisites of having both high availability and distributed data management to allow the system to function even a single point of failure within it. Based on CAP Theorem, NoSQL database systems trade consistency for availability and introduce eventual consistency concept ([Dede et al., 2013](#)). In addition, data replications allow failovers to enable “always-on” database systems which is crucial to the day-to-day operations in hospital settings.

Consistency

NoSQL is not ACID compliant, instead it adopts BASE principles. In NoSQL databases, data read by clients immediately after being updated may be outdated as not all nodes have been updated instantly. Due to this though there is no guarantee of consistency at any instance of time, it eventually will be consistent at some point in time ([Vogels, 2009](#)). [Bailis and Ghodsi \(2013\)](#) showed that the inconsistency is less than a second and therefore, eventual consistency model is claimed to be adequate in most use cases. Due to this, weaker consistency models can therefore be applied using NoSQL databases in healthcare environment without the fear of causing any major setback in terms of data consistency ([Frank et al., 2014](#)).

Data models

The need to store heterogeneous medical data has increased since using relational databases has proven inefficient and difficult to handle and manage such data ([Jin et al., 2011](#)). NoSQL databases support various types of data structures which allow flexible modelling of data, contrary to relational databases which support only structured model.

Relational Versus Graph Databases

Schema design

Schema refers to the organization of data as a blueprint of how the database is constructed prior implementation. A schema design in relational model adheres to the notion of theory in predicate calculus to define tables, fields, relationships, views, and indexes ([Rybiński, 1987](#)). Hence, relational databases are limited to homogenous data structure and extremely ineffective for heterogeneous, unstructured and frequently changing data.

In a clinical setting, data collected from patient can be in the form of texts or images. According to relational databases, data structure must be homogenous. On the contrary, a new label can be easily defined and linked to store both text and image records since graph databases embrace relationships over data types. Therefore, in terms of handling multiple variants of data, graph databases like Neo4j are more suitable for the task as relational schema is more rigid and less flexible.

Language

Relational databases like MS SQL Server use SQL to query and manage relational data. It is based upon relational algebra and tuple relational calculus. It consists of a DDL, DML and Data Control Language (DCL). SQL is a declarative language which includes data insert, query, update and delete, schema creation and modification, and data access control. It is sub-divided into several language elements ([Table 4](#)).

Table 4 SQL language elements (ISO.org, 2011)

Element	Description
Clauses	Constituent components of statements and queries.
Expressions	Produce either scalar values, or tables consisting of columns and rows of data.
Predicates	Specify conditions that can be evaluated to SQL three-valued logic (3 VL) (true/false/unknown) or Boolean truth values that are used to change program flow and limit the effects of statements and queries.
Queries	Retrieve data based on specific criteria.
Statements	Control transactions, program flow, connections, sessions, or diagnostics.

Neo4j uses Cypher language which is a declarative graph query language that allows for expressive and efficient querying and updating of the graph store. It is a relatively simple yet powerful. Unlike SELECT statement, Cypher uses MATCH and WHERE. WHERE is used to add additional constraints to patterns. CREATE and DELETE are used to create and delete nodes and relationships. SET and REMOVE are used to set values to properties and add labels to nodes (Neo4j, 2016a,b,c,d,e).

Maturity and level of support

Relational database has been around for decades and is one of the most used systems. They are used in almost every commercial and academic pursuits and have spawned several commercial ventures such as Oracle and Microsoft. Relational databases benefit from having a unified language, SQL. Since SQL does not differ greatly between implementations, support for one implementation is generally applicable to other implementations. Graph databases on the other hand, have less market penetration in general. Since most of the graph database systems are open source, their commercial ventures are usually smaller and lack a unified language. Neo4j however, has a reasonable amount of support. Hence in terms of maturity and level of support, relational databases have a better edge.

Flexibility

Both relational and graph databases perform equally well in the environments for which they were designed. Flexibility measures how well they perform outside of those environments (Vicknair *et al.*, 2010).

MS SQL Server is a large server application designed to perform in a large-scale multi-user environment. While it is optimized, it may be too heavy weighted for small applications users as there some overhead for features that are not necessarily beneficial for the small user.

Neo4j is small, light-weight, and efficient. It performs well at server scale and does not impose the overhead often associated with server applications (Vicknair *et al.*, 2010). Unlike MS SQL Server which only works on Microsoft ecosystems like Windows, as a Java component, Neo4j supports cross-platform deployment, meaning it can be run on Windows, Linux or Mac OS (Neo4j, 2016a,b,c,d,e). Therefore in terms of flexibility, Neo4j graph database is the clear winner.

Security

Most relational databases in general such as MS SQL Server have extensive built-in multi-user support. Different privileges can be assigned to individuals or groups to grant different user views and actions. On the contrary, many graph databases including Neo4j lack much support for multi-user environments. It forces all user management to be handled at the application level. By extension, Neo4j assumes a trusted environment and does not have any built-in security support (Vicknair *et al.*, 2010). Relational databases such MS SQL Server are better at ACL-based security.

Limitations of Current Study and Proposed Future Work

The present study illustrated a full scale visualization of EMR data in Neo4j and demonstrated a simple use case of how records can be linked among EMR data. Future work can look into importing and integrating open data in RDF/JSON format with current EMR graph database system, Neo4j to study and assess the proposed idea of Linked Data initiative.

Future work shall look into assessing the scalability of NoSQL databases in terms of performance and storage in comparison to a relational database in various configurations and scenarios by using different size of EMR data. In addition, EMR data can be distributed among different nodes to evaluate the performance and horizontal scalability of NoSQL database systems in a distributed environment compared to a single system using relational database. Future work can also look into assessing availability of NoSQL databases by deploying and maintaining a number of replications to simulate failovers in the event of power or systems failure.

Conclusion

Electronic Medical Records store and manage huge amounts of clinical data. New approaches are needed in order to achieve higher performance and to store data efficiently than is possible with traditional relational databases. In this study, we demonstrated that NoSQL databases (MongoDB and Neo4j), particularly the graph model is a powerful tool to visualize EMR data for sense-making

and data analysis. Graph model allows one to identify make predictive inferences which are beneficial in qualitative research. Graph model stores data in JSON format, which can be easily linked with open data to create new relationships and extend information as linked data uses a universal interchangeable RDF/JSON format to describe things. Nonetheless, the capacity of graph model and linked data are underutilized due to the lack of adoption in healthcare domain.

This study also demonstrated that NoSQL databases such as MongoDB and Neo4j performed better with a lower disk space consumption compared to relational MS SQL server. Although we cannot conclude the clear superiority of NoSQL databases over relational database, we however provide enough reasons such as generic suitability and technical and financial reliefs for large scale data-intensive applications that is provided by NoSQL databases. Given the importance of EMR systems for healthcare delivery and overall health systems, distributed NoSQL databases arguable has a clear potential to dominate the development of EMR applications due to the scalability, high availability, and flexibility it provides.

Overall, this study demonstrated how graph database system such as Neo4j is the way forward due to its powerful visualization model and potential for linked data. Although it fell short in terms of maturity, level of support and security compared to relational databases, its flexibility in handling heterogeneous data with dynamic schema, SQL-like Cypher programming language and advantages as a NoSQL database provide a better data storage and management alternative viable to hospital EMR systems.

See also: Biological and Medical Ontologies: Disease Ontology (DO). Biological and Medical Ontologies: Human Phenotype Ontology (HPO). Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Models for Computable Phenotyping. Natural Language Processing Approaches in Bioinformatics. Ontology: Introduction

References

- Abramova, V., Bernardino, J., 2013. NoSQL databases. In: Proceedings of the International C* Conference on Computer Science and Software Engineering – C3S2E '13, pp. 14–22. New York, New York, USA: ACM Press. Available at: <https://doi.org/10.1145/2494444.2494447>.
- Aparicio, M., Costa, C.J., 2015. Data visualization. Communication Design Quarterly Review 3 (1), 7–11. Available at: <http://doi.org/10.1145/2721882.2721883>.
- Bailis, P., Fekete, A., Ghodsi, A., Stoica, I., 2013. HAT, not CAP: Towards highly available transactions. In: Proceedings of the 14th USENIX Conference on Hot Topics in Operating Systems, pp. 24–24. Santa Ana Pueblo, New Mexico: USENIX Association.
- Bailis, P., Ghodsi, A., 2013. Eventual consistency today: Limitations, extensions, and beyond. Queue 11 (3), 20.
- Beynon-Davies, P., 2003. Database Systems, third ed. Palgrave Macmillan.
- Bizer, C., Heath, T., Berners-Lee, T., 2009. Linked data – The story so far. International Journal on Semantic Web and Information Systems 5 (3), 1–22. Available at: <http://doi.org/10.4018/jswis.2009081901>.
- Cattell, R., 2011. Scalable SQL and NoSQL data stores. ACM SIGMOD Record 39 (4), 12. Available at: <http://doi.org/10.1145/1978915.1978919>.
- Cutler, D., 2010. Analysis & commentary. How health care reform must bend the cost curve. Health Affairs 29 (6), 1131–1135. Available at: <http://doi.org/10.1377/hlthaff.2010.0416>.
- Dede, E., Govindaraju, M., Gunter, D., Canon, R.S., Ramakrishnan, L., 2013. Performance evaluation of a MongoDB and hadoop platform for scientific data analysis. In: Proceedings of the 4th ACM workshop on Scientific Cloud Computing – Science Cloud '13, p. 13. New York, New York, USA: ACM Press. Available at: <https://doi.org/10.1145/2465848.2465849>.
- Frank, L., Pedersen, R.U., Frank, C.H., Larsson, N.J., 2014. The CAP theorem versus databases with relaxed ACID properties. In: Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication – ICUIMC '14, pp. 1–7. New York, New York, USA: ACM Press. Available at: <https://doi.org/10.1145/2557977.2557981>.
- Gunter, T.D., Terry, N.P., 2005. The emergence of national electronic health record architectures in the United States and Australia: Models, costs, and questions. Journal of Medical Internet Research 7 (1), e3. Available at: <http://doi.org/10.2196/jmir.7.1.e3>.
- Huang, C.-W., Lu, R., Iqbal, U., et al., 2015. A richly interactive exploratory data analysis and visualization tool using electronic medical records. BMC Medical Informatics and Decision Making 15 (1), 92. Available at: <http://doi.org/10.1186/s12911-015-0218-7>.
- ISO.org, 2011. ISO/IEC 9075-2:2011. Retrieved from information technology: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53682.
- Yang, J., Tang, D., Zheng, X., 2011. Research on the distributed electronic medical records storage model. In: Proceedings of 2011 IEEE International Symposium on IT in Medicine and Education, pp. 288–292. IEEE. Available to: <https://doi.org/10.1109/ITIME.2011.6132041>.
- MongoDB, 2016a. Documents. Retrieved from Documentation: <https://docs.mongodb.com/manual/core/document/>.
- MongoDB, 2016b. Replication. Retrieved from Documentation: <https://docs.mongodb.com/manual/replication/>.
- MongoDB, 2016c. Sharding. Retrieved from Documentation: <https://docs.mongodb.com/manual/sharding/>.
- MongoDB, 2016d. Storage engines. Retrieved from Storage: <https://docs.mongodb.com/v3.2/core/storage-engines/>.
- MongoDB, 2016e. What is NoSQL? Retrieved from NoSQL Databases Explained: <https://www.mongodb.com/nosql-explained>.
- MongoDB, 2016f. WiredTiger storage engine. Retrieved from Storage Engines: <https://docs.mongodb.com/v3.2/core/wiredtiger/>.
- Neo4j, 2015. Neo4j: Real-world performance experience with a graph model. Retrieved from Neo4j: <https://neo4j.com/blog/neo4j-real-world-performance/>.
- Neo4j, 2016a. From relational to Neo4j. Retrieved from What is Neo4j?: <https://neo4j.com/developer/graph-db-vs-rdbms/>.
- Neo4j, 2016b. From SQL to cypher – A hands-on guide. Retrieved from Cypher Query Language: <https://neo4j.com/developer/guide-sql-to-cypher/>.
- Neo4j, 2016c. Graph database use cases. Retrieved November 15, 2016, from: <https://neo4j.com/use-cases/>.
- Neo4j, 2016d. Neo4j release candidate. Retrieved from Products Download: <https://neo4j.com/download/other-releases/>.
- Neo4j, 2016e. What is a graph database? Retrieved from Neo4j: <https://neo4j.com/developer/graph-database/>.
- Nguyen, L., 2010. SQL server database engine basics. Retrieved from SQL Server: <http://sqlmag.com/sql-server/sql-server-database-engine-basics>.
- Orfanidis, L., Bamidis, P.D., Eaglestone, B., 2004. Data quality issues in electronic health records: an adaptation framework for the Greek health system. Health informatics journal 10 (1), 23–36.
- Parker, Z., Poe, S., Vrbsky, S.V., 2013. Comparing NoSQL MongoDB to an SQL DB. In: Proceedings of the 51st ACM Southeast Conference on – ACMSE '13, p. 1. New York, New York, USA: ACM Press. Available at: <https://doi.org/10.1145/2498328.2500047>.
- Rybiński, H., 1987. On first-order-logic databases. ACM Transactions on Database Systems 12 (3), 325–349. Available at: <http://doi.org/10.1145/27629.27630>.
- SAS, 2016. Data visualization: What it is and why it matters. Retrieved November 15, 2016, from: http://www.sas.com/en_us/insights/big-data/data-visualization.html.
- Schmitt, O., Majchrzak, T.A., 2012. Using document-based databases for medical information systems in unreliable environments. In: Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management (ISCRAM). Vancouver, CA.

- Stonebraker, M., 2009. SQL databases vs NoSQL databases. Retrieved November 15, 2016, from: <http://cacm.acm.org/blogs/blog-cacm/50678-the-nosql-discussion-has-nothing-to-do-with-sql/fulltext>.
- Tuldoklambat, 2012. MongoDB SQL server importer. Retrieved from CodePlex: <https://sql2mongo.codeplex.com/>.
- van der Linden, H., Kalra, D., Hasman, A., Talmon, J., 2009. Inter-organizational future proof EHR systems. *International Journal of Medical Informatics* 78 (3), 141–160. Available at: <http://doi.org/10.1016/j.ijmedinf.2008.06.013>.
- VCU, 2013. A comparison of MySQL vs microsoft SQL server. Retrieved from Virginia Commonwealth University: <http://www.people.vcu.edu/~agnew/Misc/MySQL-MS-SQL.HTML>.
- Vicknair, C., Macias, M., Zhao, Z., *et al.*, 2010. A comparison of a graph database and a relational database. In: *Proceedings of the 48th Annual Southeast Regional Conference on – ACM SE '10*, p. 1. New York, New York, USA: ACM Press. Available at: <https://doi.org/10.1145/1900008.1900067>.
- Vogels, W., 2009. Eventually consistent. *Communications of the ACM* 52 (1), 40. Available at: <http://doi.org/10.1145/1435417.1435432>.
- W3C, 2015. Linked data. Retrieved November 15, 2016, from: <https://www.w3.org/standards/semanticweb/data>.
- Wood, P.T., 2012. Query languages for graph databases. *ACM SIGMOD Record* 41 (1), 50. Available at: <http://doi.org/10.1145/2206869.2206879>.

Genome Databases and Browsers for Cancer

Madhu Goyal, University of Technology Sydney, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Researchers over the last two to three decades supporting the fact that different type of cancers are fundamentally a disease of the genome. Genes are being studied and identified that underlie the different types of cancer. A better understanding of these genes is one of the most pressing needs in basic cancer research. Huge amount of the genome sequence data has become available in the last two decades for researchers to do research on cancer. This explosion of data has also provided opportunity for researchers to identify patterns among data and to develop understanding the way cancer is caused by changes in the DNA, RNA, and proteins of a cell that drive over growth of cells. This big data has also witnessed the development of the systematic study of the cancer genome and sophisticated sequencing technologies. The knowledge of cancer genome databases is very important to guide the development of more effective approaches for reducing cancer morbidity and its mortality. The identification of the genomic variations that arise in cancer can also help researchers decipher cancer development and improve upon the diagnosis and treatment of cancers.

Cancer genomics data (Chin *et al.*, 2011) is complex and heterogeneous just like cancer itself. But a comprehensive analysis of the cancer genome still remains an overwhelming task. This is partly due to the limitations in current technologies to visualize, integrate, compare and analyze cancer genomics data. Such limitations prevent investigators from truly appreciating the breadth and depth of these genomics and epigenomes resources. Thus careful statistical and algorithmic considerations are required to integrate the information provided by the large volume and variety of data alongside clinical annotations. The specific conclusions extracted from data must be presented in a coherent system for display and analysis should be accessible to the scientific and medical communities.

Genomes databases (Schattner, 2008; Yang *et al.*, 2015) for Cancer contain the information of DNA sequence and gene expression differences between tumor cells and normal host cells. Genome database mean a data repository that includes all or most of the genomic DNA sequence data of one more organisms. Human genome database includes “annotations” that either describes the features of the DNA sequence itself or other biological properties of humans. The multiple fundamental tasks are involved in building a genome database, which include identification of the locations of the genes within the genome sequence, aligning transcript data into the genomic sequence, sequencing the genomic DNA and assembling the fragments of DNA sequence onto the length of chromosomes.

A genome database typically also includes a web based interface referred to as ‘genome browser’. Genome browsers have been created to allow the simultaneous display of multiple annotations within a graphical interface. They provide the ability to search for markers and sequences, to extract annotations for specific regions or for the whole genome and to act as a central starting point for genomic research. Genome browsers allow researchers to navigate the genome in an comparable way to navigating the internet e.g., with Internet Explorer, Mozilla Firefox or Chrome. Nowadays the amount of available genomic data is vast or big and different open source databases aim to make these data accessible to all researchers. The number and variety of annotations has increased dramatically and thus there is huge requirement of a detailed view of many aspects of the genomes. In this paper several popular cancer genomics data repositories and browser applications are described (Table 1).

Various genome databases and browsers have provided different analysis tools, which have proven to be successful at facilitating a better understanding of the genomes. The overwhelming amount of cancer genomics data from multiple different technical platforms has provided increasingly opportunities to perform data integration, exploration, and analytics, especially for scientists without a computational background. These cancer genomics resources have specifically lowered the barriers of access to the complex data sets and thereby accelerate the translation of genomic data into new biological insights, therapies, and clinical trials. They have offered researchers comprehensive repositories, contextual information, and curated data as a method of handling the exponentially growing amount of sequence data. Although these resources have been useful and have solved many issues, but they will continue to face new types and ever-growing amounts of data that will exacerbate many more challenges.

Systems and Applications

NCI'S Genomic Data Commons (GDC)

The NCI's Genomic Data Commons (GDC) provides the cancer research community with an integrated data repository that enables data sharing across different cancer genomic studies. The National Cancer Institute (NCI)'s Genomic Data Commons was previously managed by the Cancer Genomics Hub (CGHub), which was established in August 2011 to provide a repository to promote the sharing of cancer data, collaboration between cancer researchers, and to generate effective treatments for the Cancer. CGHub rapidly grew to be the largest database of cancer genomes in the world, storing more than 2.5 petabytes of data and serving downloads of nearly 3 petabytes per month. The Cancer Genomics Hub mission project was completed in 2016. The GDC supports several cancer genome programs at the NCI's Center for Cancer Genomics (CCG), including The Cancer Genome Atlas (TCGA) and Therapeutically Applicable Research to Generate Effective Treatments (TARGET).

Table 1 Key resources for cancer genomics research and study

<i>Name</i>	<i>Website</i>	<i>Key features</i>
NCI's genomic data commons (GDC)	https://gdc.nci.nih.gov/	Big data repository
Cancer genome atlas (TCGA)	https://cancergenome.nih.gov/	Big data repository
GDC's data analysis, visualization, and exploration (DAVE) tools	https://gdc.cancer.gov/analyze-data/gdc-dave-tools	Web interface for exploring and analyzing GDC's cancer genomic data
ICGC data portal (DCC)	http://icgc.org/	Huge data knowledgebase of somatic mutations, abnormal expression of genes, epigenetic modifications.
cBio cancer genomics portal	http://cbioportal.org	Open-access resource of cancer genomics data sets
COSMIC	http://cancer.sanger.ac.uk/cosmic	Largest somatic mutation database
Cancer cell line encyclopedia	https://portals.broadinstitute.org/ccle	Analysis and visualization of DNA copy number, mRNA expression, mutation data of cancer cell lines
FireHose	http://gdac.broadinstitute.org/	Systemize the analyses from the cancer genome atlas TCGA
FireBrowse	http://firebrowse.org	Explore FireHose cancer data with graphical tools.
TumorPortal	http://tumorportal.org/	Analyzing and visualizing genes, cancers, DNA mutations and annotations
Integrative genomics viewer (IGV)	http://www.igv.org/	High-performance visualization tool for interactive exploration of large, integrated genomic datasets
Genome analysis toolkit	https://software.broadinstitute.org/gatk/	Industry standard for identifying SNPs, DNA and RNAseq data
Hail	https://hail.is/	Open-source, scalable framework for exploring and analyzing genomic data
European genome-phenome archive (EGA)	https://www.ebi.ac.uk/ega/about	Huge repository for all types of sequence and genotype experiments and data
Network of cancer genes (NCG)	http://ncg.kcl.ac.uk/index.php	Web based tool to study duplicability, orthology and evolutionary appearance of cancer genes
MutaGene	https://www.ncbi.nlm.nih.gov/research/mutagene/	Web tool for computational exploration of DNA context-dependent mutational patterns
UCSC Xena	http://xena.ucsc.edu/	Cancer genomics browser for viewing, analyzing, visualizing the public data hubs of cancer genomics and clinical data
canEvolve	http://www.canevolve.org/	Analysis of functional genomics which includes gene, mRNA, microRNA and protein expression, genome variations and protein–protein interactions
MethyCancer	http://methycancer.psych.ac.cn/	Hosts integrated data of DNA methylation, mutation and cancer information from public resources
SomamiR 2.0	http://compbio.uthsc.edu/SomamiR	Database of cancer somatic mutations in microRNAs (miRNA)
canSAR	https://cansar.icr.ac.uk/	Open source, multidisciplinary, cancer database developed to make provision for cancer translational research and drug discovery
NONCODE	http://www.biinfo.org/noncode/	Database of cancer somatic mutations in microRNAs (miRNA)
China cancer genome database (CCGD)	https://db.cngb.org/cancer/	Large comprehensive genome database of Chinese cancer patients

The NCI's Center for Cancer Genomics (CCG) was established to command the NCI effort in generating, cataloging and unification of datasets of alterations seen in human tumors. It also aims to be at the forefront of supporting the development of analytical tools and computation approaches for improving the understanding of large-scale multidimensional data. The CCG supports several comprehensive cancer genome research programs including The Cancer Genome Atlas (TCGA) and the Office of Cancer Genomics (OCG). OCG includes two programs i.e., Therapeutically Applicable Research to Generate Effective Treatments (TARGET) program and the Cancer Genome Characterization Initiative (CGCI). TCGA, TARGET, CGCI, and other CCG programs have provided a complete characterization of genomic changes in several human cancers; however these characterizations are maintained in separate repositories, in diverse formats, and with different data management infrastructures. NCI established the GDC to unify these efforts and to provide the cancer research community with a data service supporting the receipt, quality

control, integration, storage, and redistribution of standardized cancer genomic data sets derived from various legacy and active NCI programs.

Cancer genome atlas (TCGA)

The Cancer Genome Atlas (TCGA) is the result of collaboration between the National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI). It has created a genomic data analysis pipeline that can effectively collect, select, and analyze human tissues for genomic alterations on a very large scale. It has also generated complete multi-dimensional maps of the key genomic changes in 33 types of cancer. The TCGA dataset of about 2.5 petabytes describing tumor tissue of more than 11,000 patients is publically available and has been used widely by the research community. The data have contributed to more than a thousand studies of cancer by independent researchers and to the TCGA research network publications. The success of this national network of research and technology teams serves as a model for future projects and exemplifies the tremendous power of teamwork in science.

GDC's data analysis, visualization, and exploration (DAVE) tools

DAVE is a web interface for intuitively exploring and analyzing GDC's cancer genomic data. It provides an unprecedented level of flexibility in exploring the data by selecting patients with particular altered genes or other relevant biological and clinical features. Researchers can navigate from project cohorts to individual patients, to specific genes and mutations of interest. The tool can generate specialized graphs to help researchers visualize the genes in which the most somatic mutations are observed. Users can also plot patient survival curves and identify the molecular consequence of a mutation on the resultant protein. All cases of the project can be plotted and visualize the top 50 mutated genes affected by high impact mutations in the GDC's OncoGrid.

ICGC Data Portal (DCC)

The primary goal of International Cancer Genome Consortium (ICGC) has been to catalogue and coordinate a large number of research projects that have the common aim of interpreting the genomic changes present in many forms of cancers. ICGC has generated huge data knowledgebase of genomic abnormalities (somatic mutations, abnormal expression of genes, epigenetic modifications) in tumors from 50 different cancer types and/or subtypes across the globe and make the data available to the entire research community. The ICGC's Data Portal provides tools for visualizing, querying and downloading the data released quarterly by the consortium's member projects. The ICGC facilitates communication among the members and provides a forum for coordination with the objective of maximizing efficiency among the scientists working to understand, treat, and prevent the different cancers.

cBio Cancer Genomics Portal

The cBioPortal for Cancer Genomics was originally developed at Memorial Sloan Kettering Cancer Center (MSK). The public cBioPortal site is hosted by the Center for Molecular Oncology at MSK. The cBioPortal software is now available under an open source license via GitHub. The cBio Cancer Genomics Portal ([Gao et al., 2013](#)) is an open-access resource for exploration of multidimensional cancer genomics data sets. It currently provides access to data from more than 160 cancer studies. The portal provides graphical summaries of gene-level data from multiple platforms, network visualization, survival analysis and also software programmatic access. The cBio Cancer Genomics Portal significantly lowers the barriers between complex genomic data and cancer researchers who want rapid, intuitive, and high-quality access to molecular profiles and clinical attributes from large-scale cancer genomics projects.

COSMIC

COSMIC ([Forbes et al., 2017](#)), the Catalogue of Somatic Mutations in Cancer, is the world's largest and high resolution resource for exploring the genetics of human cancer. COSMIC is a database that collects these somatic mutation data from a variety of public sources into one standard repository containing all forms of human cancer, from the most frequent cancers in lung, breast and colon, to extremely rare forms of blood cancer. Currently, COSMIC contains 1335 disease descriptions across more than 5000 detailed classifications. Its public website has been custom-built to make the many annotations in it which can be easily explored in user-friendly graphical ways whilst also providing large tabulated data sets. The front page of website offers multiple ways to explore the database e.g., 'Resources', 'Tools', and a range of pages describing the database content and details on its accessibility.

Cancer Program Resource Gateway

The central component of the Broad Institute's Cancer Program is Cancer Program Resource Gateway, which addresses unanswered questions of cancer genomics through platforms, datasets and resources. The institute was founded in 2004 due to the need that arose from the Human Genome Project to successfully decipher the entire human genetic code. This prestigious and famous institute is owned by MIT and Harvard and the main goal is to improve human health by using genomics to advance understanding and treatment of human diseases. The cancer program resource gateway has two strands i.e. data and tools.

Some of the popular resources are:

Cancer cell line encyclopedia

The CCLE (Cancer Cell Line Encyclopedia) project is collaboration between the Broad Institute, and the Novartis Institutes for Biomedical Research and its Genomics Institute of the Novartis Research Foundation. Its main objective is to conduct a detailed genetic and pharmacologic characterization of a large panel of human cancer models. The Cancer Cell Line Encyclopedia (CCLE) project is an effort to conduct a detailed genetic characterization of a large panel of human cancer cell lines. The CCLE provides public access analysis and visualization of DNA copy number, mRNA expression, mutation data and more, for 1000 cancer cell lines.

FireHose

In order to systemize the analyses from The Cancer Genome Atlas TCGA and to study the remaining diseases FireHose was developed. GDAC Firehose now sits atop ~55 terabytes of analysis-ready TCGA data and reliably executes thousands of pipelines per month.

FireBrowse

FireBrowse is a simple way to explore cancer data, backed by a powerful computational infrastructure, application programming interface (API) and graphical tools. It sits above the TCGA GDAC Firehose one of the deepest and most integrated *open* cancer datasets in the world with over 80 K sample aliquots from 11,000+ cancer patients, spanning 38 unique disease cohorts.

TumorPortal

TumorPortal is for analyzing and visualizing genes, cancers, DNA mutations and annotations. In TumorPortal datasets can be explored by tumor types as well as genes.

Integrative genomics viewer (IGV)

The Integrative Genomics Viewer (IGV) is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations. Although IGV is often used to view genomic data from public sources, its primary emphasis is to support researchers who wish to visualize and explore their own data sets or those from colleagues.

Genome analysis toolkit

The GATK is the industry standard for identifying SNPs, DNA and RNAseq data. It also has the capability to find somatic variation, to tackle copy number (CNV) and structural variation (SV). GATK also includes many utilities to perform related tasks such as processing and quality control of high-throughput sequencing data. And although it was originally developed for human genetics, the GATK has since evolved to handle genome data from any organism

Hail

Hail is an open-source, scalable framework for exploring and analyzing genomic data. It can generate variant annotations like call rate, Hardy-Weinberg equilibrium p-value, and population-specific allele count. Hail can also generate new annotations from existing ones as well as genotypes, and use these to filter samples, variants, and genotypes.

European Genome-Phenome Archive

The European Genome-phenome Archive (EGA) is designed to be a repository for all types of sequence and genotype experiments, including case-control, and family studies. It includes SNP and CNV genotypes from array based methods and genotyping done with re-sequencing methods. EGA serve as a permanent archive that will archive several levels of data including the raw data (which could, for example, be re-analysed in the future by other algorithms) as well as the genotype calls provided by the submitters. Data at EGA is collected from individuals whose consent agreements to authorise data release only for specific research use to bona fide researchers. The EGA accepted data types include raw data formats from the array-based and new sequencing platforms as well as phenotype files describing study samples. EGA offers a range of tools for the file and meta-data upload like Java EgaCryptor, which encrypts and verifies all the accepted file types, and the EGA Webin portal for the metadata submission.

Network of Cancer Genes (NCG)

The network of cancer genes (NCG) is a web based tool to duplicability, orthology and evolutionary appearance of cancer genes. It introduces a more robust procedure to extract manually curated genes from published cancer mutational screenings. NCG allows to interpreting cancer sequencing by determining if a gene has already been reported as a driver gene in a given cancer type. This helps interpreting the mutational landscape of a tumor sample and to identify whether the genes mutated have a known role in cancer or if they are likely to be passengers. NCG provides new approaches to target cancer genes i.e., the systems-level properties reported for each cancer gene and allow identifying new targets and strategies to target a mutated gene.

MutaGene

MutaGene (Goncarencu *et al.*, 2017) is an online computational framework, which explores DNA context-dependent mutational patterns and underlying somatic cancer mutagenesis. It explores mutational profiles of cancer samples, identifies the combinations of underlying mutagenic processes including those related to infidelity of DNA replication and repair machinery, and various other endogenous and exogenous mutagenic factors. It also calculate expected mutability for each DNA and protein site using background mutational models.

UCSC Xena

UCSC Xena (Goldman *et al.*, 2015) is a cancer genomics browser for viewing, analyzing, visualizing the public data hubs of cancer genomics and clinical data. It is a web-based application that is developed in response to a crucial demand in the cancer research field for integrative visualization of large, complex genomic datasets arising from different technology platforms. It hosts many public databases such as TCGA, ICGC, TARGET, GTEx, CCLE, and others. These databases are normalized before use so that they can be combined, linked, filtered, explored and downloaded. Xena is very flexible with data and can load most types of genomic or phenotype data into it. Researchers can explore the relationship between genomic alterations and phenotypes by visualizing various genomic data alongside clinical and phenotypic features, such as age, subtype classifications and genomic biomarkers. It can also generate Kaplan Meier plot (KM plot) to test whether a genomic or phenotype variable significantly affects survival. Xena can group samples by any data (cell lines, xenografts, organoids, or patients) and then use these groups to compare expression or any other genomic data.

canEvolve

CanEvolve (Samur *et al.*, 2013) supports different type of analysis of information extracted from 90 cancer genomics studies comprising of more than 10,000 patients. The portal stores functional genomics and other large-scale data on cancer which includes gene, mRNA, microRNA and protein expression, genome variations and protein–protein interactions. The portal also provides stored knowledge in database (canEvolve web portal is implemented using MySQL open source system) as well as generate analysis results from oncogenomic profiles in response to user queries. It allows visualization of knowledge and analysis results in an appropriate manner and let the user download query results and related information from the portal. The querying for primary analysis includes differential gene and miRNA expression as well as changes in gene copy number measured with SNP microarrays. Finally, canEvolve provides query functionalities to fulfill most frequent analysis requirements of cancer researchers towards generating novel biological hypotheses.

MethyCancer

DNA methylation plays a vital role in the development of cancer and is associated with oncogene activation and chromosomal instability. MethyCancer (He *et al.*, 2008) hosts integrated data of DNA methylation, mutation and cancer information from public resources, and the CpG Island (CGI) clones derived from large-scale sequencing. The database also has graphical Methy-View, which shows DNA methylation in context of genomics and genetics data. It thus facilitates the research in cancer to understand genetic mechanisms that make dramatic changes in gene expression of tumor cells.

SomamiR

Genetic and somatic mutations or miRNA–mRNA interactions have been associated with various cancers. The miRNAs are small non-coding RNAs, known for their role as post-transcriptional regulators of protein-coding mRNAs. SomamiR 2.0 (Bhattacharya and Cui, 2016) is a database of cancer somatic mutations in microRNAs (miRNA). It helps researchers study the interactions between miRNAs and competing endogenous RNAs (ceRNA) including mRNAs, circular RNAs (circRNA) and long noncoding RNAs (lncRNA). It also has webserver miR2GO which is integrated with the database to provide a seamless pipeline for assessing functional impacts of somatic mutations in miRNA seed regions.

canSAR

canSAR (Tym *et al.*, 2016) is open source, multidisciplinary, cancer database developed to make provision for cancer translational research and drug discovery. It integrates comprehensive multidisciplinary knowledge i.e., annotation for genes, pharmacological, drug and chemical data with structural biology to enable target validation and drug discovery. It also applies machine learning approaches to provide drug-discovery useful predictions. It is known globally as the key resource to aid target selection and prioritization of drug discovery for cancer. canSAR thus provides unique views on genes and proteins, drugs, 3D structures, protein interaction networks, cancer cell lines and cancer clinical trials.

NONCODE

Mounting evidence shows that non-coding RNAs play key roles in various biological processes and in disease etiology. The recently reduced cost of RNA sequencing has produced an explosion of newly identified data, and as a result there has been an explosive rise in the number of newly identified non-coding RNAs. NONCODE (Yi *et al.*, 2015) is an integrated database of cancer somatic mutations in microRNAs (miRNA) from several species such as human and mouse. NONCODE provides a subset searching interface, literature support, other database support and long-read sequencing method support. The quality controls also include selection of exon numbers, the lengths of the transcripts and prediction tools support. The web interface presents the subset according to the conditions users chose and allow users to download the data.

China Cancer Genome Database (CCGD)

The China Cancer Genome Database (CCGD) is built by China National GeneBank as a first, large comprehensive genome database in China. The CCGD Data Portal stores, catalogs, and accesses the cancer related data, and provides a platform for researchers to download data sets. The types of data include raw data, clean data, alignment data, function analysis data, phenotype data. This database collects and stores cancer genome data and shares with all cancer research centers in order to promote the development of China cancer genome research. The database utilizes these data or research mechanism of the most frequently occurred cancers among Chinese population and largely promotes research of the early diagnosis and cure of cancer in China.

Conclusions

Advancement in software and information technologies has enabled software developers and bioinformatics researchers to integrate different publicly-accessible genetics data types from a large variety of sources. The coordinated efforts of many organizations have also led to the development of large-scale cancer genomics through which complete catalogs of the genomic alterations in specific cancer types can be obtained. Overall there are four different categories of genome resources discussed in this article. The first category represents very large repositories of genomics data such as TCGA, ICGC, COSMIC, EGA, cBio portal, China Cancer database with their own browsing tools. The second category resources offer tools for data analysis, data integration, and visualization (e.g., UCSC Xena, FireBrowse, TumorPortal, IGV, GATK, Hail, NCG, MutaGene). The third class includes databases that focus on inferring the association of specific biological features to cancer (e.g., MethyCancer, SomamiR, and NONCODE). Finally, databases such as CanSAR support the application of genomics to drug discovery. There are other useful resources for cancer research, but are not included in this paper due to the limitation of space and scope.

Although the methods for analyzing cancer genomes have improved at a swift rate, but still many challenges remain. Big Data management and advanced computational methods should be developed and integrated with tools to establish the clinical relevance of cancer genomic discovery. Collecting accurate clinical information on tumor samples remains an important and challenging task, but one that is necessary for the interpretation of genomic findings in depth. However, most available samples are associated with incomplete annotation or lack appropriate consent to permit the linkage of clinical information. There is thus a huge scope for personalized cancer genomic information, which once widely available and successfully implemented, holds considerable promise for improving the lives of many patients with cancer. Lastly, but importantly, the growing number of genome databases, analysis tools, and other resources available on the web has made it an unnerving task for researchers to use these resources effectively.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Data Storage and Representation. Genome Annotation. Genome Informatics. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases. Quantitative Immunology by Data Analysis Using Mathematical Models. Standards and Models for Biological Data: FGED and HUPO. Whole Genome Sequencing Analysis

References

- Bhattacharya, A., Cui, Y., 2016. SomamiR 2.0: A database of cancer somatic mutations altering microRNA–ceRNA interactions. *Nucleic Acids Research* 44 (D1), D1005–D1010. PMID: 26578591.
- Chin, L., Hahn, W.C., Getz, G., Meyerson, M., 2011. Making sense of cancer genomic data. *Genes & Development* 25, 534–555.
- Forbes, S.A., Beare, D., Boutselakis, H., *et al.*, 2017. COSMIC: Somatic cancer genetics at high-resolution. *Nucleic Acids Research* 45 (Issue D1), D777–D783.
- Gao, J., Aksoy, B.A., Dogrusoz, U., *et al.*, 2013. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Science Signaling* 6 (269), p1.
- Goldman, M., Craft, B., Swatloski, T., *et al.*, 2015. The UCSC cancer genomics browser: Update 2015. *Nucleic Acids Research* 43 (Issue D1), D812–D817.
- Gonczarenko, A., Rager, L.S., Li, M., *et al.*, 2017. Exploring background mutational processes to decipher cancer genetic heterogeneity. *Nucleic Acids Research* 45 (Issue W1), W514–W522.
- He, X., Chang, S., Zhang, J., *et al.*, 2008. MethyCancer: The database of human DNA methylation and cancer. *Nucleic Acids Research* 36, D836–D841.
- Samur, M.K., Yan, Z., Wang, X., *et al.*, 2013. CanEvolve: A web portal for integrative oncogenomics. *PLOS ONE* 8, e56228.
- Schattner, P., 2008. *Genomes, Browsers and Databases: Data-Mining Tools for Integrated Genomic Databases*. Cambridge University Press.
- Tym, J.E., Mitsopoulos, C., Coker, E.A., *et al.*, 2016. canSAR: An updated cancer research and drug discovery knowledgebase. *Nucleic Acids Research* 44 (D1), D938–D943.
- Yang, Y., Dong, X., Xie, B., *et al.*, 2015. Databases and web tools for cancer genomics study. *Genomics, Proteomics & Bioinformatics* 13, 46–50.
- Yi, Z., Hui, L., Fang, S., *et al.*, 2015. NONCODE 2016: An informative and valuable data source of long non-coding RNAs. *Nucleic Acids Research* 44 (D1), D203–D208.

Text Mining Resources for Bioinformatics

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal and University of Minho, Braga, Portugal

Priyanka Baloni and Charu K Midha, Institute for Systems Biology, Seattle, WA, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

With the digital revolution, there has been an explosion in the amount of information available, which in turn has transformed the traditional ways of data analyses. Mining important information from terabytes of data is the need of the hour. Data mining uses sophisticated algorithms and statistical measurements to extract useful information from existing data. In modern day computing, the main types of data are text, numbers and multimedia. Text mining is the process of analyzing a collection of unstructured data in the form of text and deriving useful information or generating a new hypothesis. There is a wide scope and application of this technology in risk management, cyber-crime prevention (Berry and Kogan, 2010; Witten *et al.*, 2016; Kontostathis *et al.*, 2010), fraud detection (Joudaki *et al.*, 2015), biomedical research (Rebholz-Schuhmann *et al.*, 2012), to name a few.

Text mining for bioinformatics has an overwhelming potential for translational research. There are thousands of research articles published every year showcasing the discoveries made in the scientific field, making it difficult for the researchers to follow them in great details. Due to the massive information generated, there is a need for computational tools to parse and analyze the information to make the learning process simpler. Text mining, as the term suggests, is the use of an automated method for extracting meaningful data from the wealth of information available in literature and other resources. The process of text mining comprises mainly of information retrieval, information extraction, knowledge discovery and hypothesis generation. The information extracted in the process is generally stored in database for future reference.

There are various tools and resources that are used by bioinformaticians for retrieving and extracting information from unformatted data. Some of the extensively used retrieval engines for biomedical research are PubMed, PubChem, RefMED, UK PubMed Central (Walport and Kiley, 2006), WorldWideScience, MedlinePlus, GoPubMed and Google Scholar (Ananiadou *et al.*, 2006). In order to extract information, some of the commonly used resources are iHOP, Textpresso, Open Biomedical Annotator, InAct, MedScan, Reactome as well as KEGG (Rebholz-Schuhmann *et al.*, 2012). There is a constant effort in building curated databases that contain information retrieved and extracted for specific purposes. STRING, STITCH, SIDER, HPID, HPRD, Bio-Caster, PharmGKB are few databases that have been curated and extensively used by researchers to conclude valuable inferences (Rebholz-Schuhmann *et al.*, 2012; Zhu *et al.*, 2013). Fig. 1 depicts the process of text mining involving converting unformatted text to machine-interpretable structured format.

Text mining is computationally intensive and is performed using technologies such as natural language processing, knowledge management, information extraction, machine learning as well as pattern recognition. The present text mining tools have virtually explored only the tip of the iceberg and there is still lots to be explored. The field of text mining has a wide scope for research development and has immensely assisted researchers in the advancement of biomedical research. Although, there are challenges associated with this field (discussed later), with the development of powerful algorithms and techniques, this field has received great impetus. This article discusses the fundamental concepts and various text mining resources developed for researchers.

Is Text Mining Ready to Deliver?

Information mining is a process that involves a clear understanding of various stages involved in parsing information. Some of these stages not only require bioinformatic techniques like pattern evaluation from databases and automatic detection, but scientific knowledge, analyst creativity as well as basic knowledge of the field. A better understanding of the biological process or problems help us to structure text mining projects, therefore, they are closer to accurate analysis rather than driven by chance or individual's insight.

History of Text Mining

Text mining techniques have evolved over many years with a view of expansion and improving methodologies by learning from previous attempts. Text mining was commenced in the earlier times to catalog the literature in the form of books in libraries. Soon, it was deviated to data retrieval using Natural Language Processing methods (abbreviated as NLP – it is an attempt to understand the modelling of the natural human language using computers). Indeed, text mining has evolved with time owing to improvised computational techniques. Since 2001, information on the internet has expanded from over 100 terabytes to 1500 terabytes in 2009, which is nearly 40% explosion in the information generation and storage. This exponential growth of information is mainly composed of a vast number of personal, public and corporate pages, news, electronic books, scientific data repository, databases *etc.* The Library of Congress, one of the largest library in the world contains 17 million catalog records for books, serials, manuscripts, maps, music, recordings, images, and electronic resources in its collections and is still growing exponentially. Although, this field is still in its developmental stage, it has already contributed to many scientific discoveries.

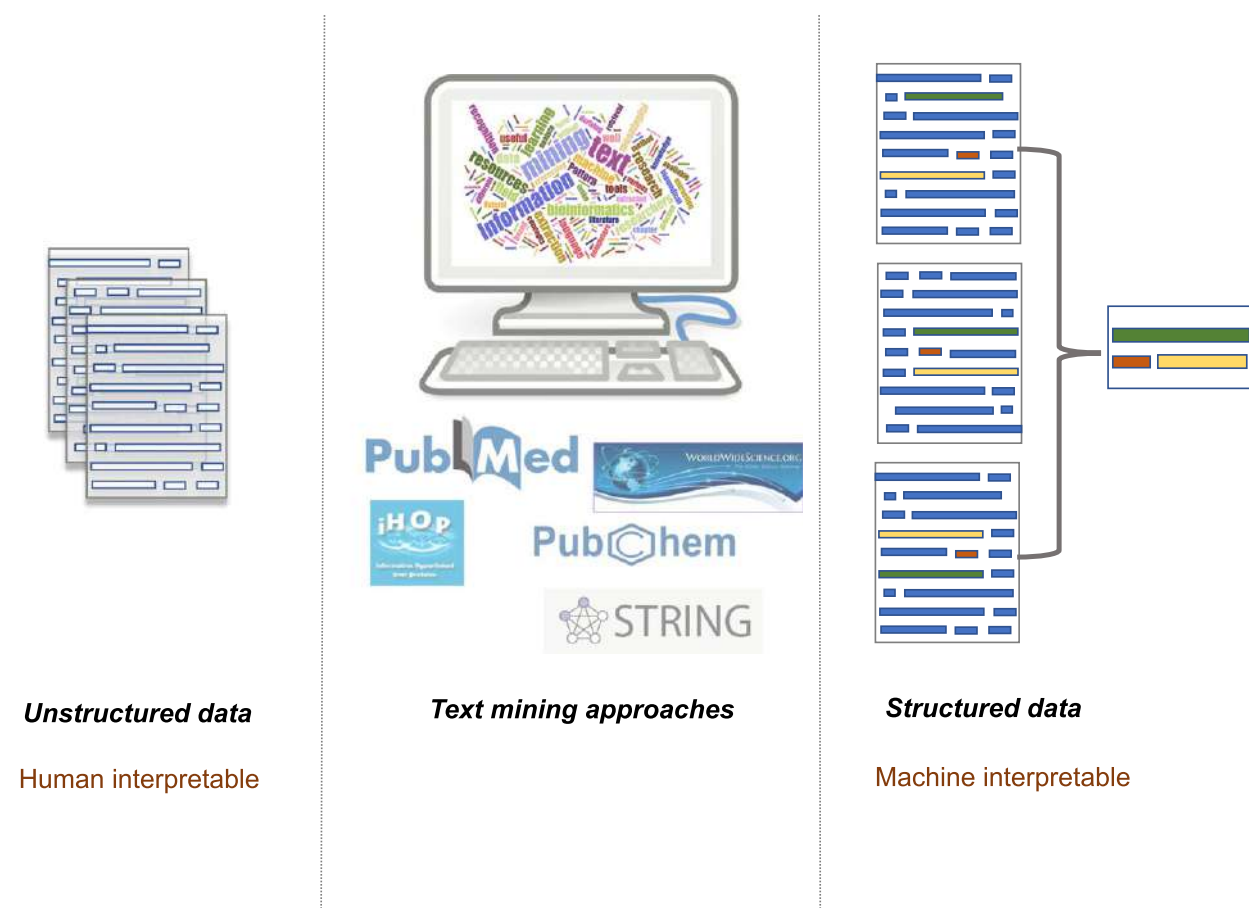


Fig. 1 Deconvolution of the unstructured data into machine-interpretable format using various text mining approaches.

Trends in Text Mining

In earlier days, library catalog involved text summarization and classification which was contributed by Thomas Hyde from University of Oxford for the Bodleian Library in 1674. Later in 1876, Melvil Dewey introduced the index card for library card catalog at Yale University. Text mining has not sprouted in one instance, rather several technological advancements, extensive data generations of different types and applications have contributed to the present modern data retrieving systems.

Later, by the year 1898, the text processing included summarization of the information for generating abstracts, majorly in Science stream (Luhn, 1958). It was as a joint collaboration between the Physical Society of London and Institute of Electrical Engineers. Luhn used computers to generate the document abstracts in 1958 by exploiting the word frequency analysis on an early IBM 701 vacuum tube computer (Luhn, 1958). It gave relative significance measurement of sentences which were combined according to their linear distances between the words to produce a metric of significance sentence contributing to form an abstract of a document. This automated procedure very well adapted in library sciences. Electronic versions of the scientific abstracts had become available since 1967 onwards.

As the technology advanced from huge room size computers to personal computers and revolution in digitization, document cataloging has improved with more access to other existing documents. Science Citation Indexing (SCI) is a citation index created by Eugene Garfield in 1964 from Institute for Scientific Information and is currently owned by Clarivate Analytics after Thomson Reuters. Its enhanced version Science Citation Index Expanded includes more than 8500 journals, across 150 disciplines (Garfield, 2006, 2007, 1964).

The searching mechanism was advanced by text manipulations, indexing, and development of Natural Language Processing (NLP). To access the information from the repository, two processes – Information Retrieval and Information extraction are used. Eventually, computational power in the 1960s invented NLP applications, clustering, information extraction and modern text mining methods. All these methods are briefly described in the next section.

Since 2000, major innovations and methods in text mining include modern information extraction engines (based on tokenization); categorization of text with Machine Learning techniques, Support Vector Machines, kernel-based learning methods, Neural Networks, automation, linking and statistical approaches (Miner, 2012). A glimpse of few of these methods are discussed in the next section.

Background

Mining information from big data repositories involves strategies and techniques that can process data in fast and reliable manner. In order to improvise scientific inferences and narrow down the search parameters, it is important to carry out pattern recognition, relation extraction, knowledge discovery as well as hypothesis generation. One of the major challenges of retrieving data from scientific databases demands the ability to think how such fundamental concepts apply to specific biological problems ([Sharma and Srivastava, 2016](#)). This section highlights the main concepts underlying textmining approaches.

Concepts

Text mining

The term denotes a process aimed at analyzing unstructured text extracted from various text resources, detecting lexical patterns useful for deriving previously unknown information ([Berry and Castellanos, 2008](#); [Sebastiani, 2002](#)).

Machine learning (ML)

It is a method to train computers to make and improve predictions based on given data without being explicitly programmed. Some of the widely used machine learning approaches are Hidden Markov Models (HMM), Support Vector Machines (SVMs), Conditional Random Fields (CRFs) as well as Maximum Entropy (ME) ([Sharma and Srivastava, 2016](#)).

Natural language processing (NLP)

It relies on machine learning algorithms to analyze and understand the context in human language that is easily interpretable by computers. NLP is used for text mining, translation, relationship extraction as well as speech recognition. This method involves processing of words in the text and relies on statistical techniques. Some of the open source libraries for NLP are Natural Language Toolkit (NLTK), Apache OpenNLP, Stanford NLP and MALLET (see “Relevant Websites section”).

Information retrieval

Information retrieval (IR) deals with searching for information as well as recovery of textual information from a collection of resources. The desired information is often posed as a search query, which in turn recovers those articles from a repository that are most relevant and matches to the given input. Google scholar, PubMed, UK PubMed Central [06] as well as Elsevier’s Science Direct are some of the widely-used document search and retrieval engines for research or publications. Also, resources like PubMed QUEST are very helpful in the rapid searching of PubMed for publications of interest ([PubMed, 1946](#)).

Information extraction

Information extraction (IE) is the task of automatically extracting meaningful semantic structures using different strategies and identify relations between concepts. Open Biomedical Annotator, iHOP, Textpresso, CoPub are examples of resources used for information extraction ([Rebholz-Schuhmann et al., 2012](#)).

Pattern recognition

It is a branch of machine learning and involves recognizing patterns in data and classify the input data into different classes based on certain features.

Document summarization

This tool takes the documents into consideration and returns a concise summary of the text that highlights key points in the longer text.

Importance of Text Mining

From Experiments to Articles to Models

The experimentalist designs and carry out the lab work (experiments) to obtain the results followed by its processing and interpretation. The significant outcomes are published in scientific journals that follow specific journal format comprising of Title, Abstract, Keywords, Text body, References, Tables and Figures.

National Center for Biotechnology Information (NCBI) developed PubMed database (see “Relevant Websites section”), which is an information retrieval system at National Library of Medicine (NLM). This database is mainly dedicated to life sciences literature. Access to the literature is through NCBI Entrez retrieval system which is a text-based search and retrieval method. It can have a search query by author name, the title of the papers, or the keywords.

Scientific community submits their citations electronically from different journals into PubMed. Presently, PubMed has more than 27 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher websites. Among these published articles, few are indexed with

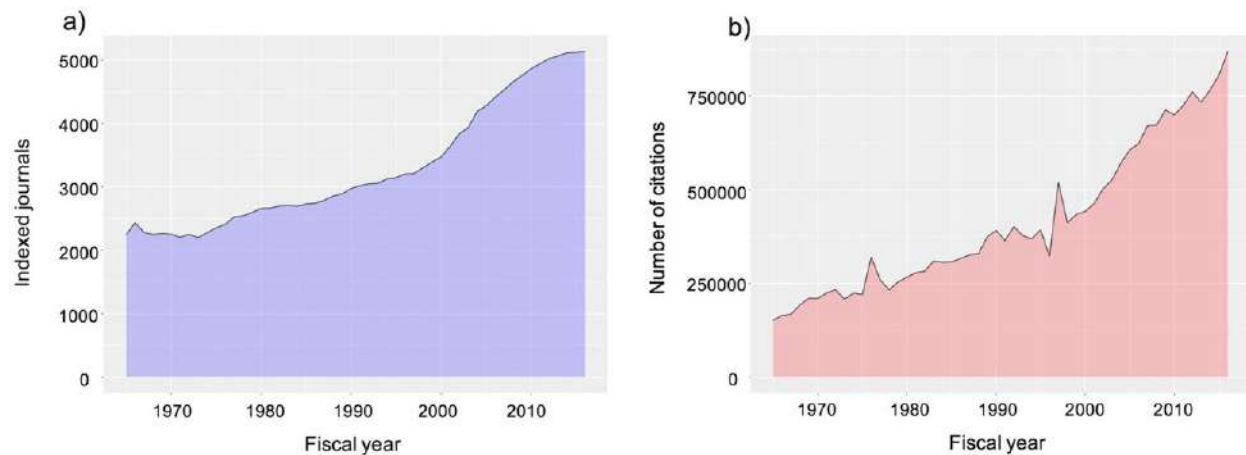


Fig. 2 The growth in indexed journals in PubMed (a) and recorded citations from 1965 to 2016 (b) is represented in plots. (https://www.nlm.nih.gov/bsd/index_stats_comp.html).

Table 1 A table illustrating the most used tools for text mining (Biological networks, Protein–Protein interactions, NMR/Crystal structure)

Database name [Reference]	Name of database	URL
NCBI-PubMed (Hanauer and Chinnaiyan, 2006)	Published biomedical literature	https://www.ncbi.nlm.nih.gov/pubmed
DIP (Salwinski <i>et al.</i> , 2004)	Database of interacting proteins	http://dip.doe-mbi.ucla.edu/dip/Main.cgi
HPID (Han <i>et al.</i> , 2004)	Human protein interaction database	http://wilab.inha.ac.kr/hpid/webforms/intro.aspx
MINT (Licata <i>et al.</i> , 2011)	Molecular interaction database	http://mint.bio.uniroma2.it/
STRING (Szkarczyk <i>et al.</i> , 2017)	Search tool for the retrieval of interacting genes	https://string-db.org/
PDB (Berman <i>et al.</i> , 2002)	Protein data bank	https://www.rcsb.org/pdb/home/home.do
UniProt (UniProt Consortium, 2011)	The universal protein resource	http://www.uniprot.org/

MeSH (expanded as Medical Subject Headings) term publication types to GenBank Accession numbers. MeSH is NLM curated medical vocabulary resource (see “Relevant Websites section”). MeSH, indexes and catalogues the terminology belonging to biomedical information (MEDLINE/PubMed and other NLM databases) in hierarchical order. The growth in the number of indexed journals in PubMed and recorded citations from 1965 to 2016 is represented in Fig. 2.

As discussed in concepts above, Google has developed Google Scholar search engine targeting academic and research users. It’s a scholarly literature that includes peer-reviewed papers, theses, books, abstracts, reports. On passing query it returns results based on full text, authors, publications types/journals and number of citations.

Biomedical literature has characteristics features in terms of usage of domain-specific terminologies (technical terms); words possessing ambiguity in their meaning e.g., *Drosophila* has genes named archipelago, capicua or ebony; new terminology and names, low frequency of certain words, typographical variants etc. to keep in mind while searching (Table 1).

Case Studies of Use of Text Mining in Bioinformatics/Biomedical Research

Accessing information from database: PlasmoDB as an example

Among the biological databases available for the scientific community, several dedicated databases exist belonging to a specific class, organism, characterization etc. For example – Eukaryotic Pathogen Database Resources, EuPathDB (see “Relevant websites section”) (Fig. 3) is a portal provided by Bioinformatics Resource Center for accessing genomic-scale datasets from wide eukaryotic microbes listed in Table 2.

This article will cover one of the databases from EuPathDB, called Plasmodium Genomics Resource, PlasmoDB (see “Relevant Websites section”) in detail (Aurrecoechea *et al.*, 2008).

Of the five species of Plasmodium- *Plasmodium vivax*, *P. falciparum*, *P. ovale*, *P. malariae* and *P. knowlesi*, mostly the first four species cause human malaria and occasionally by *P. knowlesi* which causes animal malaria. Among these *P. falciparum* is the most lethal. PlasmoDB comprises of all the updated information on each and every gene of these species/strains uploaded and submitted by the experimentalists. It has linked up the information to other relevant databases and summarized results in a single window.

The home page of PlasmoDB shows wide range of search options such as: text-based, gene models (exon count/gene type), annotations, curation and identifiers (gene IDs), genomic location, taxonomy (organism), sequence analysis (BLAST, protein

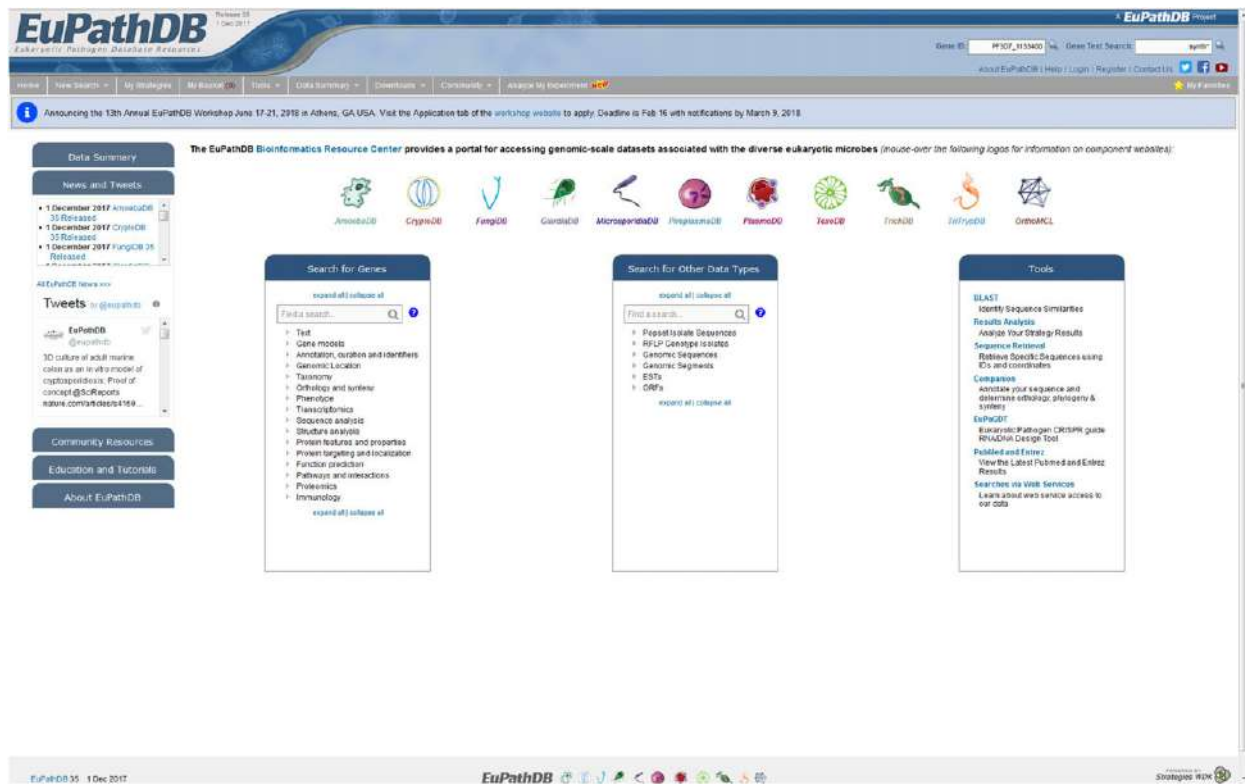


Fig. 3 Screenshot of EuPathDB.

Table 2 List of databases included in the EuPathDB consortium

Database	Url	Specification
AmoebaDB	http://amoebadb.org/amoeba/	Acanthamoeba, Entamoeba and Naegleria
CryptoDB	http://cryptodb.org/cryptodb/	Chromera, Cryptosporidium, Gregarina and Vitrella
FungiDB	http://fungidb.org/fungidb/	Agaricomycetes, Blastocladiomycetes, Chytridiomycetes, Eurotiomycetes, Leotiomyces, Oomycetes, Pneumocystidomycetes, Pucciniomycetes, Saccharomycetes, Schizosaccharomycetes, Sordariomycetes, Tremellomycetes, Ustilaginomycetes, Zygomycetes
GiardiaDB	http://giardiadb.org/giardiadb/	Giardia and Spironucleus
MicrosporidiaDB	http://microsporidiadb.org/micro/	Annaliia, Edhazardia, Encephalitozoon, Enterocytozoon, Hamiltosporidium, Nematocida, Nosema, Spraguea, Trachipleistophora, Vavraia, Vittiforma
PiroplasmaDB	http://piroplasmadb.org/piro/	Babesia and Theileria
PlasmoDB	http://plasmodb.org/plasmo/	Plasmodium
ToxoDB	http://toxodb.org/toxo/	Eimeria, Hammondia, Neospora, Sarcocystis, Toxoplasma
TrichDB	http://trichdb.org/trichdb/	Trichomonas
TriTrypDB	http://tritrypdb.org/tritrypdb/	Crithidium Endotrypanum, Leishmania, Trypanosoma
OrthoMCL	http://orthomcl.org/orthomcl/	Ortholog Groups of Protein Sequences

motif pattern, TF Binding Site Evidence), structure analysis (PDB 3D structures, predicted 3D structures, protein secondary structures), and micro-array expression based searches.

On looking in detail the results obtained after searching tyrosyl- tRNA synthetases by its Gene ID: PF11_0181, opens a page with a brief summary of the gene at the left side (**Fig. 4**): its name, type as in coding/non-coding, chromosome no. and its location, species and strain name.

Right panel reveals the experimental information, gene model- exons and transcripts with their length in pictorially; annotation, curations and identifiers – listing its previous IDs and alias notes that are specific to annotator or user comments; Link outs- external links to different databases and tools for the gene of interest (such as Entrez Gene, Gene DB, Malaria Literature Database, Ontology Based pattern Identification, PlasmoDraft, PubMed, UniProt); genomic locations on the chromosome; associated literatures to this gene/protein with their PubMed ID, DOI links, titles and authors; taxonomic classification- to which

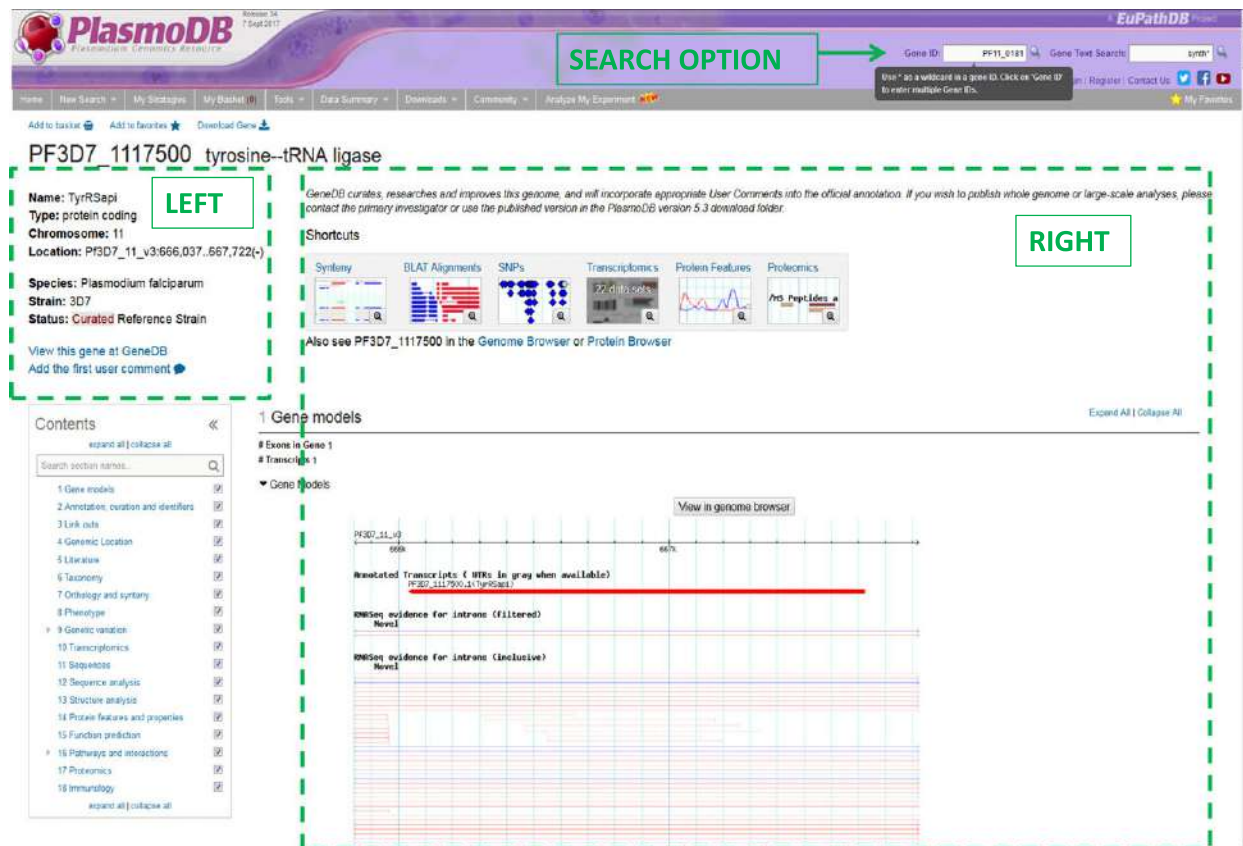


Fig. 4 Screenshot of the PlasmoDB with tyrosine-tRNA ligase as a hit.

superkingdom, phylum, class, order, family, genus and species it belongs; Orthologous and paralogous list of proteins from different species; Synteny of this gene and its domains in pictorial form, phenotypic information, and genetic variation; transcriptomics expression information; sequences -protein/mRNA/genomic sequence and their lengths, structural information-structural homologous PDB IDs from different organisms, and structural information coverage of the protein; protein family classifications, signal peptides presence and its length and relevant BLASTP hits and low complexity regions, secondary structure predictions (Helix or Beta strands); functional classification, gene ontology classification, pathways involved and protein-protein interactions, proteomic- Mass Spectrometry based expression information at different stages of *Plasmodium falciparum* life cycle.

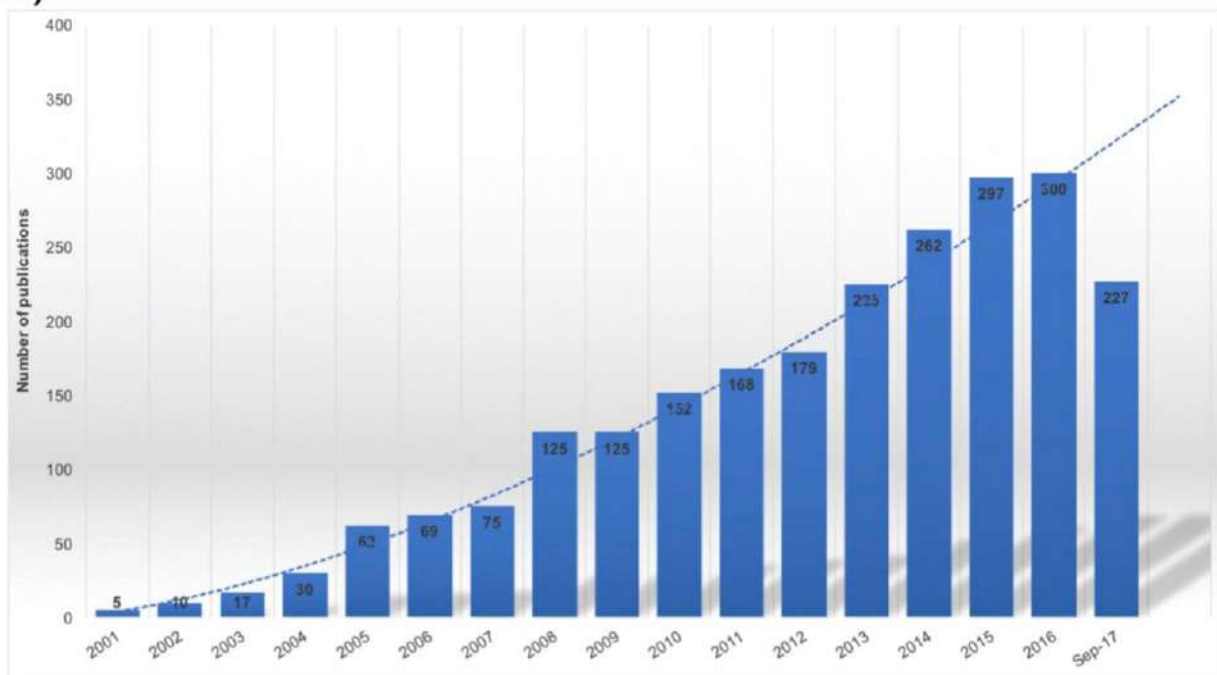
In addition, it allows to perform the multiple sequence alignment (ClustalW) or Multi-FASTA of the nucleotide sequence with any other set of sequences through external links. To conclude, this DB platform provides the best practices to retrieve all or any information of a gene/protein belonging to *Plasmodium* species.

Personalized medicine and text mining

Text mining has played a pivotal role in the development of diverse fields. The field of personalized medicine has immensely benefitted from development in text mining approaches. As shown in Fig. 5(a), the number of publications containing the term 'text mining' in title or abstract when queried in PubMed has increased greatly since 2001. We can forecast the number of publications for text mining from the trend of previous years. With rapid advances in genomics and healthcare, it has become possible to tailor the best medical intervention to an individual. With the knowledge of underlying genotypic variations responsible for a particular phenotype, we can improve predictions in turn leading to prevention and cure of disease. The field of personalized medicine is gradually becoming mainstream practice. With the options of genome sequencing and various health monitoring trackers, the field of personalized medicine has gained momentum. The field of text mining has immensely helped in deriving the relationships between various factors such as genes, proteins, metabolites and others. As can be seen in Fig. 5(b), the number of publications on personalized medicine is increasing every year.

Identifying the disease-gene relationship is a major challenge in this field. In many cases, ML-based methods are used to identify the genetic mutations described in biomedical literature related to particular disorder or disease. In one such study, the authors retrieved biomedical literature related to target disease and extracted information for mutation and disease (Singhal et al., 2016). They used several features such as statistical, distance and sentiment features for a ML-based method to predict the relationship between disease and gene mutation. The advantage of using this approach is generalizability and scalability as it is feasible to use the model to extract relationship for any disease using larger biomedical literature knowledgebase. Some of the

a)



b)

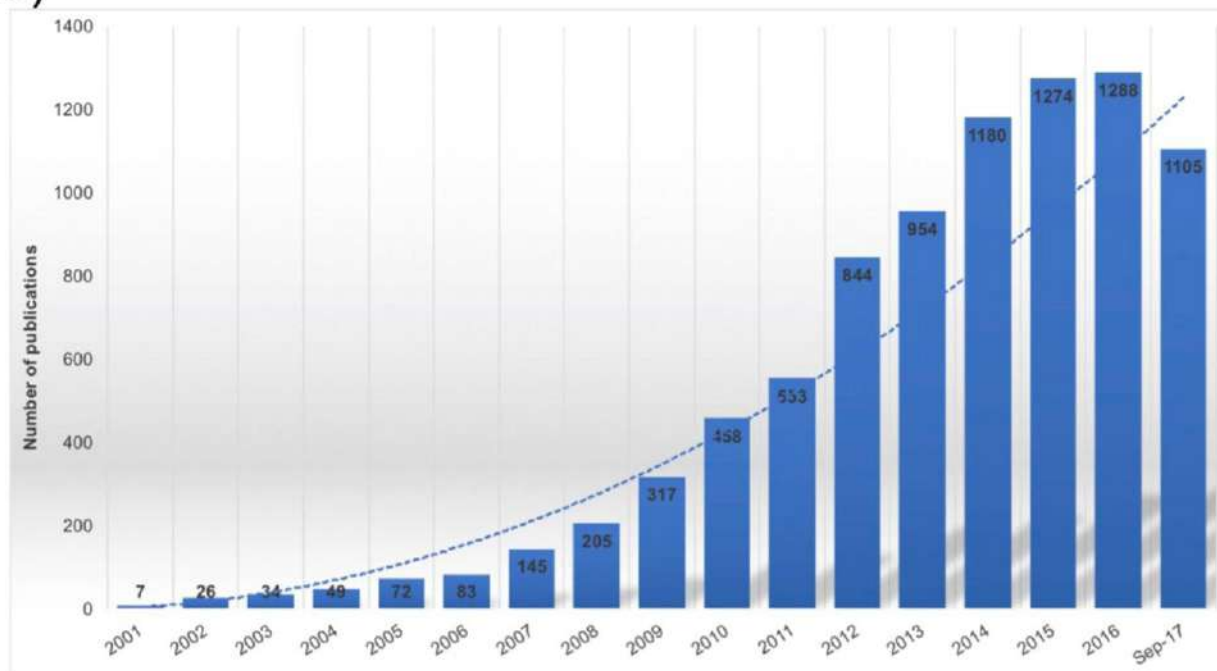


Fig. 5 Plots representing an increase in the number of publications using (a) 'text mining' and (b) 'personalized medicine' as query term in PubMed. The data from 2001 to September 2017 is plotted.

areas of improvement for text mining in personalized medicine is the availability of highly curated gene-mutation database, a crowdsourcing effort to curate existing databases, use of electronic health records and clinical trials data in addition to biomedical literature as well as robust methods for such analysis. Text mining and curation have facilitated biomedical research, development of knowledge base for precision cancer medicine being one of the relevant examples.

Cancer models and text mining

Cancer is responsible for deaths of millions of people every year in the world. There has been constant effort to find cure for cancer, and numerous publications feature cancer research each year. Biomedical text mining has been used by researchers in mining cancer-related genes and proteins and their relationship to various forms of cancer. Automatic recognition of relationship between cancer and genes or protein terms associated with the disease is one of the important aspects of bioinformatics. In order to identify genes related to prostate cancer and their relationship, prostate cancer-related abstracts from MEDLINE were used to develop maximum entropy-based system (Chun *et al.*, 2006). Other than identifying genes involved in cancer, it is also important to verify them. In one such study, the prostate cancer biomarkers obtained from experimental studies were validated by employing text mining approach with OMIM (Online Mendelian Inheritance in Man) (Deng *et al.*, 2006). One of the primary goals of cancer research is developing strategies for early detection of cancer so that it is useful for prevention and controlling the disease. Using biomedical text mining techniques, cancer risk assessment can be evaluated by using terms from literature as features, and developing classifiers for the purpose of automatic identification from text. Databases such as PubMeth, MeInfoText, and others are used for obtaining information such as genes and their association in cancer methylation (Ongenaert *et al.*, 2007; Fang *et al.*, 2011). Patient's clinical records also serves as a good source to extract relevant information. There has been great progress in creating automated systems for constructing models from free-text pathology reports. MedTAS/P is useful in mapping the pathology reports with cancer model (Codem *et al.*, 2009). This system also efficiently captures the details of grading and staging cancer. The clinical significance of such an NLP-based system is the ease in cancer practice management by sub-classifying cancer patients based on the grade or stage of cancer. This practice is slowly entering mainstream medicine and will be helpful to clinicians in deciding the course of treatment in near future. Some of the well-documented text mining systems used for clinical purposes are MedLEE (Medical Language Extraction and Encoding) system and cTAKES (clinical Text Analysis and Knowledge Extraction System) (Zhu *et al.*, 2013). Thus, to study the complex mechanisms of cancer, the need of the hour is text mining from hierarchical network view and discover new knowledge using this systems biomedicine perspective.

Text mining and drug discovery

Drug discovery is the process of identifying new candidate molecules that are selected by screening hits, their affinity, selectivity, potency, stability as well as bioavailability. The field of drug discovery has undergone a paradigm shift from using traditional methods to identify active ingredient to employing computational methods for such discovery (Bull *et al.*, 2000). Publicly-available chemical databases such as PubChem, Chemical Entities of Biological Interest (ChEBI), ChemExpr, ChemBank, Side Effect Resource (SIDER), ChemSpider, Therapeutic Target Database (TTD), and DrugBank provides information of chemical entities that can be explored for therapeutic purposes (Papanikolaou *et al.*, 2016). Text mining approach has been applied to extract information from DrugBank. DrugQuest is an information retrieval and extraction tool that uses textual information from DrugBank and clusters these records. Once the user provides query, the DrugBank records are selected on the basis of the fields such as the mechanism of action, pharmacodynamics, and indication. Named Entity Recognition techniques have been used for storing and parsing information from DrugBank. Tagging services such as Reflect and BeCAS (Papanikolaou *et al.*, 2016) help in minimizing the ambiguity in the gene, protein and chemical terms and are useful in resolving the issue of multiple synonyms while searching the text. The retrieved documents from DrugBank are clustered algorithmically on the basis of the information contained in them. DrugQuest efficiently summarizes the results of the analysis and also provides user the option of visual representation of the results. Although DrugQuest has the limitation of 5000 textual records per analysis, it takes only a few seconds to process and compute the output (Papanikolaou *et al.*, 2016). Fig. 6 shows the output obtained when 'isoniazid' and 'methotrexate' are given as query in DrugQuest.

Thus, text mining approaches are useful in mining chemical repositories to find relationship between chemical entities and leverage this information for drug repositioning. It will be even more useful if chemical information can be extracted simultaneously from various repositories such as PubChem, ChemExpr, ChEBI, and SIDER. This kind of concerted effort will boost the development of this field and provide information that was unknown till now.

Discussion

There is a surge of scientific data that is generated at an exponential rate and limited methods for comprehensive analysis of data. There is an urgent need for the development of searching methods to minimize the gap between powerful storehouses and fetching useful information in an efficient manner. User-specific problems have to be addressed primarily and in more structured way. Thus, an understanding of these facts is crucial for data scientists, recruiters, investors, or the end user scholars. A better and clear understanding of process and stages involved helps in our rationale and systematic thinking leading to less errors and their omissions.

There are challenges involved in searching methodology like human formulated questions using natural language – “What are the molecular functions of Glycogenin?”, alternative scientific terms, new terminologies derivations etc. Inferences derived using text mining resources can be of immense value to the scientific community as a whole. Indeed, there are evidences, claiming that data-oriented decisions and big data technologies improve scientific performances.

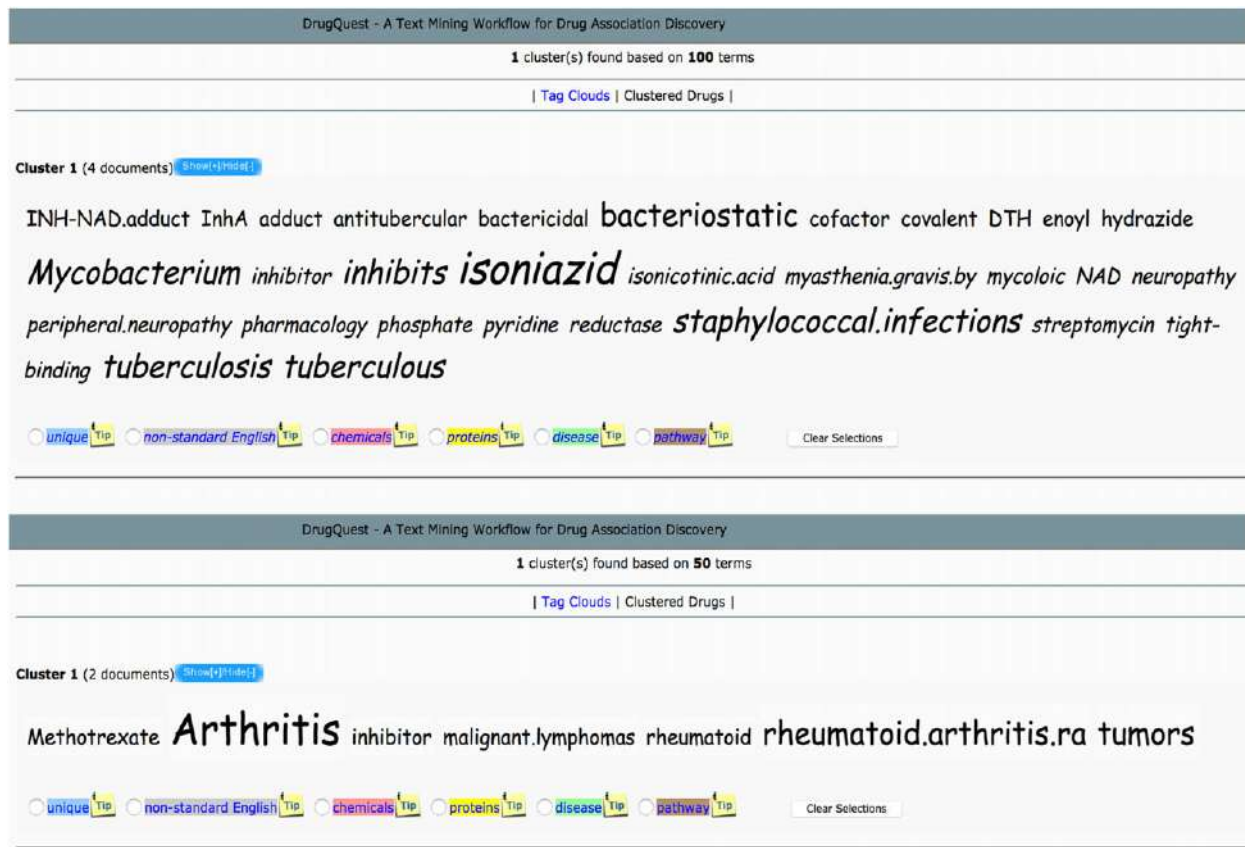


Fig. 6 The clustered output from DrugQuest for terms such as 'isoniazid' and 'methotrexate', respectively. Reproduced from Zhu, F., Patumcharoenpol, P., Zhang, C., *et al.*, 2013. Biomedical text mining and its applications in cancer research. *Journal of biomedical informatics*, 46 (2), 200–211.

Closing Remarks

Here we encourage the scholar to read listed references -published papers and books, relevant to the topic for advance information. Explore the information from given links of different bioinformatics databases and text mining tools. This will enable the learner to acquire the knowledge that is available while implementing it all together.

Acknowledgement

Sandeep Kaushik, Priyanka Baloni and Charu K Midha, designed the workflow and wrote the article.

See also: Bayes' Theorem and Naive Bayes Classifier. Data-Information-Concept Continuum From a Text Mining Perspective. Integrative Bioinformatics. Natural Language Processing Approaches in Bioinformatics. Text Mining Applications. Text Mining Basics in Bioinformatics. Text Mining for Bioinformatics Using Biomedical Literature

References

- Ananiadou, S., Kell, D.B., Tsujii, J.I., 2006. Text mining and its potential applications in systems biology. *Trends in Biotechnology* 24 (12), 571–579.
- Aurrecochea, C., Brestelli, J., Brunk, B.P., *et al.*, 2008. PlasmoDB: A functional genomic database for malaria parasites. *Nucleic Acids Research* 37 (Suppl. 1), D539–D543.
- Berman, H.M., Battistuz, T., Bhat, T.N., *et al.*, 2002. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography* 58 (6), 899–907.
- Berry, M.W., Castellanos, M., 2008. *Survey of Text Mining II*. vol. 6. New York: Springer.
- Berry, M.W., Kogan, J. (Eds.), 2010. *Text Mining: Applications and Theory*. John Wiley & Sons.
- Bull, A.T., Ward, A.C., Goodfellow, M., 2000. Search and discovery strategies for biotechnology: The paradigm shift. *Microbiology and Molecular Biology Reviews* 64 (3), 573–606.

- Chun, H.W., Tsuruoka, Y., Kim, J.D., *et al.*, 2006. Automatic recognition of topic-classified relations between prostate cancer and genes using MEDLINE abstracts. *BMC Bioinformatics* 7 (3), S4.
- Coden, A., Savova, G., Sominsky, I., *et al.*, 2009. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model. *Journal of Biomedical Informatics* 42 (5), 937–949.
- Deng, X., Geng, H., Bastola, D.R., Ali, H.H., 2006. Link test – A statistical method for finding prostate cancer biomarkers. *Computational Biology and Chemistry* 30 (6), 425–433.
- Fang, Y.C., Lai, P.T., Dai, H.J., Hsu, W.L., 2011. MeInfoText 2.0: Gene methylation and cancer relation extraction from biomedical literature. *BMC Bioinformatics* 12 (1), 471.
- Garfield, E., 1964. Science Citation Index – A new dimension in indexing. *Science* 144 (3619), 649–654.
- Garfield, E., 2006. Citation indexes for science. A new dimension in documentation through association of ideas. *International Journal of Epidemiology* 35 (5), 1123–1127.
- Garfield, E., 2007. The evolution of the science citation index. *International Microbiology* 10 (1), 65.
- Hanauer, D.A., Chinnaiyan, A.M., 2006. PubMed QUEST: The PubMed Query Search Tool. An informatics tool to aid cancer centers and cancer investigators in searching the PubMed databases. *Cancer Informatics* 2, 79.
- Han, K., Park, B., Kim, H., Hong, J., Park, J., 2004. HPID: The human protein interaction database. *Bioinformatics* 20 (15), 2466–2470.
- Joudaki, H., Rashidian, A., Minaei-Bidgoli, B., *et al.*, 2015. Using data mining to detect health care fraud and abuse: A review of literature. *Global Journal of Health Science* 7 (1), 194.
- Kontostathis, A., Edwards, L., Leatherman, A., 2010. Text mining and cybercrime. In: *Text Mining: Applications and Theory*. Chichester, UK: John Wiley & Sons, Ltd.
- Licata, L., Briganti, L., Peluso, D., *et al.*, 2011. MINT, the molecular interaction database: 2012 update. *Nucleic Acids Research* 40 (D1), D857–D861.
- Luhn, H.P., 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2 (2), 159–165.
- Miner, G., 2012. *Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications*. Academic Press.
- Ongenaert, M., Van Neste, L., De Meyer, T., *et al.*, 2007. PubMeth: A cancer methylation database combining text-mining and expert annotation. *Nucleic Acids Research* 36 (Suppl. 1), D842–D846.
- Papanikolaou, N., Pavlopoulos, G.A., Theodosiou, T., Vizirianakis, I.S., Iliopoulos, I., 2016. DrugQuest – A text mining workflow for drug association discovery. *BMC Bioinformatics* 17 (5), 182.
- PubMed, 1946. Bethesda (MD): National Library of Medicine (US). (cited 2017 Dec 21).
- Rebholz-Schuhmann, D., Oellrich, A., Hoehndorf, R., 2012. Text-mining solutions for biomedical research: Enabling integrative biology. *Nature Reviews Genetics* 13 (12), 829–839.
- Salwinski, L., Miller, C.S., Smith, A.J., *et al.*, 2004. The database of interacting proteins: 2004 update. *Nucleic Acids Research* 32 (Suppl. 1), D449–D451.
- Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)* 34 (1), 1–47.
- Sharma, S., Srivastava, S.K., 2016. Review on text mining algorithms. *International Journal of Computer Applications* 134 (8).
- Singhal, A., Simmons, M., Lu, Z., 2016. Text mining for precision medicine: Automating disease-mutation relationship extraction from biomedical literature. *Journal of the American Medical Informatics Association* 23 (4), 766–772.
- Szklarczyk, D., Morris, J.H., Cook, H., *et al.*, 2017. The STRING database in 2017: Quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research* 45 (D1), D362–D368.
- UniProt Consortium, 2011. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Research*. gkr981.
- Walport, M., Kiley, R., 2006. Open access, UK PubMed central and the wellcome trust. *Journal of the Royal Society of Medicine* 99 (9), 438–439.
- Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
- Zhu, F., Patumcharoenpol, P., Zhang, C., *et al.*, 2013. Biomedical text mining and its applications in cancer research. *Journal of Biomedical Informatics* 46 (2), 200–211.

Relevant Websites

<http://eupathdb.org/eupathdb/>
EuPathDB.

<https://www.nlm.nih.gov/mesh/>
MeSH Browser.

<http://www.phontron.com/nlptools.php>
Natural Language Processing Tools.

<https://www.ncbi.nlm.nih.gov/pubmed/>
PubMed
NCBI
NIH.

<http://plasmodb.org/plasmo/>
PlasmoDB.

Preclinical: Drug Target Identification and Validation in Human

Meena K Sakharkar and Karthic Rajamanickam, University of Saskatchewan, Saskatoon, SK, Canada

Chidambaram S Babu, JSS University, Mysuru, India

Jitender Madan, Chandigarh College of Pharmacy Landran, Mohali, India

Ramesh Chandra, University of Delhi, Delhi, India

Jian Yang, University of Saskatchewan, Saskatoon, SK, Canada

© 2019 Elsevier Inc. All rights reserved.

Introduction

Target identification is the first step in drug discovery. A target is an entity to which an endogenous ligand or a drug binds resulting in a change in its behavior or function. Enzymes, receptors and transport proteins along with DNA and RNA are the main targets for drugs at the molecular level. The pre-clinical drug development process is based on the hypothesis that modulation of the target(s) of interest will result in a therapeutic effect in a disease state. The human genome data is an excellent resource to understand the genetic factors in human disease, and one of its prime goals was to pave the way for new strategies for disease diagnosis, treatment and prevention; as it has the potential to reveal targets involved in a specific disease. However, not many new molecular entities have been discovered since the sequencing of the human genome. The paucity of translatable results can be attributed to several factors. Lack of accurate information on gene architecture and gene annotation which is essential for drug discovery as this allows insight into splice variants and drug cross-reactivity. Moreover, potential targets provided by the genome projects are not endowed with elaborate background knowledge because of our inability to infer the function of most of the DNA in the genome. Many of the computationally derived annotations in the databases are either minimal or incorrect (apart from a carefully manually-curated database such as Uniprot) (Zhao *et al.*, 2007). Also, as the annotation of genes is provided by multiple public resources using different methodologies, the resultant information may be similar but not always identical (Kangueane *et al.*, 2015). This is further confounded by our limitation to infer the number of splice variants or protein-protein interactions from gene sequences alone, thereby limiting our analyses of genome and in consequence, our estimation of the total number of targets. Lack of information on protein structure, protein function and binding of a specific drug to the target of interest adds another dimension to this puzzle. In conclusion, the increased number of targets and the lack of functional knowledge about them are generating a bottleneck in the target validation process (Ofra *et al.*, 2005). Despite the relatively limited knowledge about the complex relationship between chemical space and genomic space, drug development strategies have been influenced profoundly by the wealth of potential targets offered by genome projects (Finan *et al.*, 2017). This article highlights some of the key aspects in human pre-clinical drug target identification and validation.

Target Identification *in Silico*

Over the past two decades, several computational (*in silico*) methods have been developed and applied to generate pharmacology hypotheses. These *in silico* methods are primarily used alongside *in vitro* data to create the *in vivo* model for pre-clinical testing and optimization of novel molecules with the potential to bind to a target, and the ADMET (absorption, distribution, metabolism, excretion and toxicity) properties in addition to the physicochemical characterization.

Drugs Databases and Drug Target Databases

The first step towards drug discovery is the correct identification and validation of the drug target and its interaction with the drug. The key drug databases and drug target databases are:

Drug databases

The PubChem database contains about 35 million compounds. It is estimated that <7000 compounds have the information on their corresponding target proteins (Chen *et al.*, 2016). It has been reported that the known chemical space probably contains on the order of 100 million molecules and the estimate for Lipinski virtual chemical space is around 10^{60} compounds or a more modest 10^{20} – 10^{24} molecules if the combination of known fragments are considered (Ertl, 2003). ChEMBL contains 1.6 million distinct compounds, 14 million bioactivities, 11K biological targets, and other related data organized in 72 tables manually collected from the published literature. These data are very useful for drug discovery and include binding, functional and ADMET (i.e., assessment of *in vivo* absorption, distribution, metabolism, excretion and toxicity properties) information for a larger number of drug-like bioactive compounds (Nowotka *et al.*, 2017). ZINC contains over 120 million compounds for ligand discovery and virtual screening and allows for investigators to seek chemical matter for their biological targets (Sterling and Irwin, 2015). DCDB (Drug Combination Database) offers data on drug combinations. Its current version comprises of 1363 approved or investigational drug combinations, including 237 unsuccessful drug combinations, involving 904 individual drugs, from > 6000 references

(Liu *et al.*, 2014). The information about drugs and their targets is manually annotated based on the literature and relevant databases such as Drugbank (Wishart *et al.*, 2017), PubChem, UniProt and Drugs.com.

Thus, even with today's computational resource, the entire chemical space cannot be exhaustively enumerated computationally and prioritisation and targeted selection is essential for virtual screening.

Drug target databases

The DrugBank database is a richly annotated bioinformatics and cheminformatics resource that combines detailed drug data (e.g., chemical, pharmacological and pharmaceutical) with comprehensive target information (e.g., sequence, structure and pathway) (Wishart *et al.*, 2008). Therapeutic target database (TTD) provides the information about known and explored therapeutic protein and nucleic acid targets, the targeted diseases, pathway information and corresponding drugs directed at each of these targets (Chen *et al.*, 2002). Recently, the information of 1755 biomarkers for 365 disease conditions and 210 drug scaffolds for 714 drugs and leads has been added into this database (Qin *et al.*, 2014). PharmGKB is another pharmacogenomics knowledge resource that encompasses clinical information including dosing guidelines and drug labels, potentially clinically actionable gene-drug associations and genotype-phenotype relationships. PharmGKB also collects curates and disseminates knowledge about the impact of human genetic variation on drug responses (Thorn *et al.*, 2013).

Computational Approaches in Drug Target Identification

In order to understand the cellular processes taking place in response to a drug, it is critical to elucidate its molecular targets. This has tremendous implications for disease prevention and treatment. Here, it is important to mention that the incorrect identification of targets is a significant factor in drug failures and the low drug approval rate during drug development stemming from an incomplete knowledge on the underlying physiology of the target and incorrect biological hypothesis (Vasaikar *et al.*, 2016). Specifically, in line with this, the association of a drug with additional targets beyond its direct ones, may give rise to hazardous side effects, precluding further drug development and usage. 82% of Food and Drug Administration (FDA) approved drugs have an assigned mechanism-of-action (Overington *et al.*, 2006). However, at least half of the drugs with unassigned mechanism of action date from the pre-molecular era and have no targets defined as their action was explored against whole tissues, rarely on isolated proteins, and target identities were only inferred from tissue-based responses (Gregori-Puigjane *et al.*, 2012). Even though the total number of predicted drug targets of pharmacological interest is estimated in the range of 6000–8000 in the human genome, drugs have been approved for only a fraction of these targets (Landry and Gies, 2008; Plewczynski and Rychlewski, 2009). This is because a large number of these putative drug targets remain to be validated. Druggability is the presence of protein folds (quaternary structures) that favor interactions with drug-like chemical compounds (Hajduk *et al.*, 2005). The ability of a protein to bind a small molecule with the appropriate chemical properties and binding affinity might make it druggable, but does not necessarily make it a potential drug target. This is true, though, for proteins that are linked to human diseases (Hopkins and Groom, 2002). In this regard, Hopkins and Groom have highlighted that ~10% of the entire human genome is involved in disease onset or progression, resulting in ~3000 potential targets suitable for therapeutic intervention (Perumal *et al.*, 2009; Sakharkar and Sakharkar, 2007; Sakharkar *et al.*, 2007). An attempt to identify the number of druggable disease genes using DGIdb database (see "Relevant Websites section"), which is a compendium of 3860 druggable genes from BaderLabGenes, CarisMolecularIntelligence, FoundationOneGenes, GO, GuideToPharmacologyGenes, HopkinsGroom, MskImpact, RussLampel, and dGene resources and the DisGeNet platform which reports on the gene disease associations derived from expert curated databases and text mined data (see "Relevant Websites section") shows that there are 1645 druggable disease genes which are associated with 923 diseases. Here, it is important to mention that DisGeNet provides information on 26,522 disease gene associations reporting 7878 disease genes and 6761 unique diseases. Taking genes involved in breast cancer from BCDB database and mapping them onto the 1645 druggable genes, we found 471 genes that are involved in breast cancer and are druggable (Fig. 1).

Once the disease mechanism is understood and the target and lead compounds are identified, chemoinformatic tools present a tremendous potential to help decipher the interactions between protein targets and the ligands before expensive and time-consuming experiments. Towards this end, we have earlier shown the coffee component HHQ as a putative ligand for PPAR gamma and its implications in breast cancer *in vitro* (BMC Genomics) (Fig. 2) (permission applied).

However, one should keep in mind that the identification of novel drugs and their targets is still an extremely difficult goal owing to the limitation in mapping drugs and targets in chemical space and genomic space into a unified space, called pharmacological space.

Approaches for Target Validation *in Silico*, *in Vitro* and *in Vivo*

Validation is a crucial step in the drug discovery process. The only way to be completely certain that a protein is instrumental in a given disease is to test the idea in clinical trials. With the evolution of sequence targeted therapeutics like CRISPR and RNAi silencing, gene expression levels can be raised or lowered at transcriptional, translational or post-translational stages. The effects of these can be evaluated in preclinical models generated by the gene editing of experimental animals and human cells (2017). Additionally, the role of Genome-wide association studies (GWAS) of disease predisposition, metabolism and gene expression provides data on

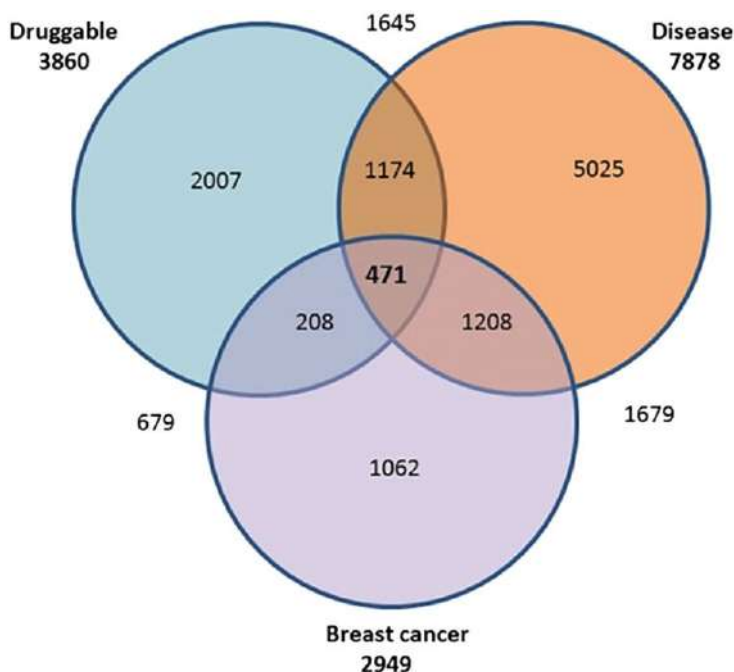


Fig. 1 Schematic representation of number of genes involved in breast cancer as provided by breast cancer data base (BCDB). Out of 1645 druggable genes, 471 genes were involved in breast cancer.

functioning of protein-targeting drugs. GWAS also use the regulatory variation in human genomes to guide the therapeutic future of targeted gene regulation (2017). Thus, the protein based concept of druggability needs to be expanded and redefined. Several resources and techniques help validate the predicted drug-target interaction. Some of them are briefly described below:

Gene and protein expression profiling datasets: Gene expression profiling through transcriptomics (RNA profiling) and proteomics (protein profiling) has contributed significantly to our efforts in understanding complex biological processes. Technologies such as microarrays, quantitative real-time polymerase chain reactions, and next-generation sequencing have helped decipher the targeted receptor(s), pathway or network through the identification of downregulated or upregulated pattern(s) of the intended drug target (s). **Connectivity Map (CMAP)** is a publicly accessible gene expression database that contains over 1.5 million gene expression profiles from ~5000 small-molecule compounds, and ~3000 genetic reagents, tested in multiple cell types (see "Relevant Websites section"). CMAP collates, compares and links all changes in gene expression ("signatures") arising from a disease or gene modulation (knockdown or overexpression of a gene), or treatment with a small molecule (Lamb *et al.*, 2006). Differential expressions that show highly similar or dissimilar, expression signatures are termed "connected". Similar physiological effects on the cells imply that the transcriptional effects are related. GEO is a database of gene expression profiles derived from over 100 organisms and more than a billion individual gene expression measurements addressing several biological issues such as disease states and stages of development. It is a public repository that freely distributes this information that is submitted by the scientific community. It can be explored, queried, and visualized using user-friendly Web-based tools. **DeSigN** is a web-based tool for predicting drug efficacy against cancer cell lines using gene expression patterns (Lee *et al.*, 2017). It can use gene expression analyses to identify candidate drugs using an input gene signature. Identifying distinct common pathways/gene modules that are shared by pathophysiological processes of multiple diseases is made easier by the integration of these diverse datasets.

Phenotypic Screening

Phenotypic screening is based on the alteration(s) of metabolically active systems such as tissues or whole animals produced at physiological or molecular structural levels upon exposure to a chemical entity. Hits are identified in phenotypic screening and followed by the leads thereof by structural activity relationship assays. Phenotypic screenings are based on the system suitability, the stimulus and the end-point readout rather than the pathological and clinical endpoints (Vincent *et al.*, 2015). These ensure robustness, repeatability and cost-effectiveness of the procedure. Phenotypic screenings are carried out using test systems which include cell lines, situ organ/tissue preparations and animal models. The animal models used in phenotypic screening include *C. elegans*, *Danio rerio* (Zebrafish), *X. laevis*, and *Drosophila melanogaster*. In disease models, phenotypic screening involves the determination of changes in structural morphology and biochemical characteristics and effects of compounds (drugs) on reversal of the changes. The screening of compounds on the neurite growth and amyloid- β , neurofibrillary tangles in Alzheimer diseases

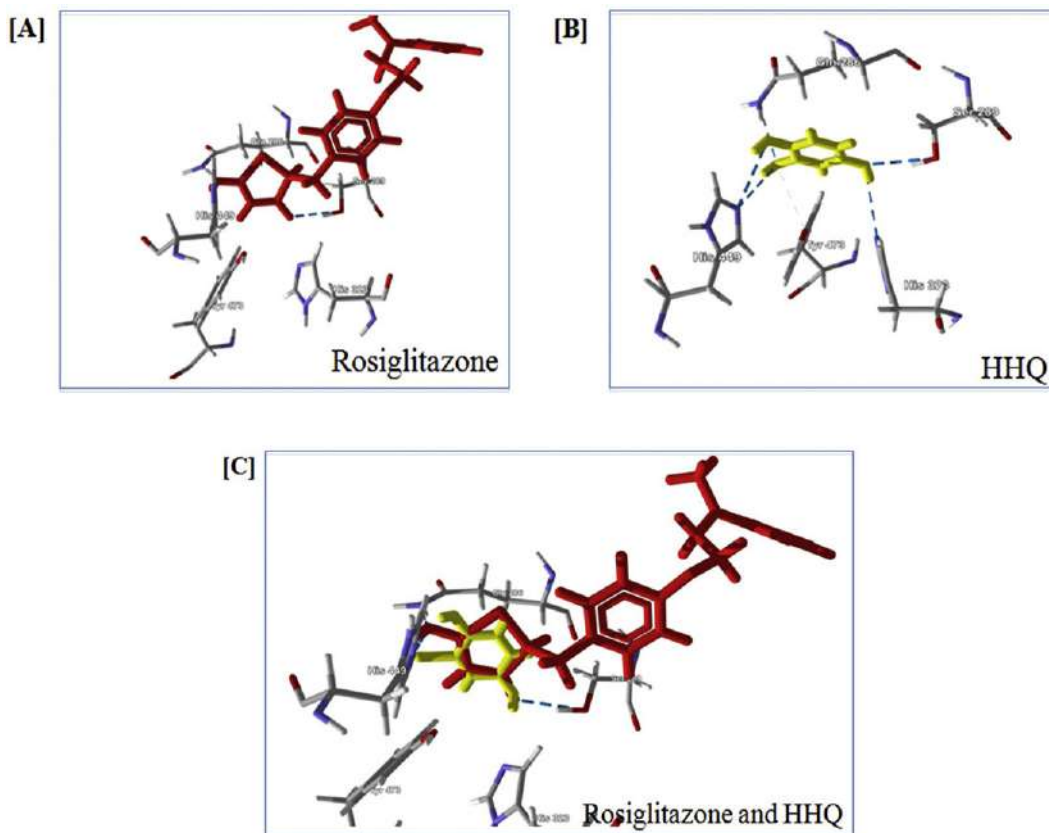


Fig. 2 HHQ docked in the ligand binding domain of PPAR γ protein crystal structure for PDB solved in conjunction with Rosiglitazone (PDBID: 2PRG). [A] Represents hydrogen bonds (in black dotted line) observed for Rosiglitazone (in red) with the active site residues in the ligand binding domain of PPAR γ (2PRG). [B] Represents hydrogen bonds (in black dotted line) observed for HHQ (in yellow) with the active site residues in the ligand binding domain of 2PRG. [C] Represents superposition of the best conformation of Rosiglitazone (in red) and HHQ (in yellow) in the ligand binding domain of 2PRG. Rosiglitazone is a known ligand of PPAR γ (published in [Shashni, B., Sharma, K., Singh, R., et al., 2013](#). Coffee component hydroxyl hydroquinone (HHQ) as a putative ligand for PPAR gamma and implications in breast cancer. BMC Genomics 14 (Suppl. 5), S6).

and measurements is one of the best examples for phenotypic screening in which both the cellular morphology and the biochemical changes and effects of drugs are studied in parallel.

One of the key limitations of phenotypic screening is that it usually provides very little information on the possible targets of the compounds. Moreover, compounds with a poor ADMET may not be active in primary screens. These factors demand cell-based assays or transgenic/knockout models to identify the target proteins and provide some information on the possible mechanism of action of the compounds/drugs.

Transgenic organisms

Genetically modified animals are widely in research to create models of human diseases. These animal models are created based on a molecular understanding of genomic/proteomic alterations in human diseases. However, the intensity of up- or down-regulation, progress and intracellular accumulation may vary phenotypically from the human disease. Animal models are genetically modified to over-express (transgenic), remove ("knock-out"), or replace ("knock-in") specific genes. These animals are widely used to understand the pathology and investigate the effects of hits/leads for their efficacy ([Bolon, 2004](#)). Furthermore, the use of genetically modified models provides information on target identification and helps establish the precise mechanism efficacy or toxicity of the new chemical entities or drugs. For example, in the transgenic mouse model of Parkinsonism, the animals were modified to over-express human α -synuclein, a major protein that aggregates in neuronal cells in Parkinson's disease (PD). This animal model mimics PD pathology and simulates it closer to the human disease. Further, the animals which over-expressed human α -synuclein showed decreased dopamine turn over in turn motor deficits. Thus, the model serves as a target for the phenotypic screening of drug candidates having effects on α -synuclein. Transgenic models also help to rule out the down- or upstream signalling pathways, which are further helpful in understanding how the disease progresses and its possible therapeutic options.

Imaging

Imaging techniques have often been used to study gross anatomy. Read outs usually show structural deformities. With advancements in technology, it became feasible to image physiological and biochemical events in metabolically active systems such as cell lines and animal models. This is called functional imaging. The live imaging of specific proteins or target proteins of interest was made possible with the help of the development of molecular probes, this is often called molecular imaging. The main advantages of molecular imaging are that it is repetitive, non-invasive, and the effects are quantifiable at cellular and sub-cellular levels. These imaging techniques use various energy forms to penetrate or interact with the endogenous proteins or fluorescence tagged proteins in in-vivo systems (Fig. 3).

With the help of sophisticated imaging techniques, the identification of marker proteins and effects of drugs on the target proteins is performed in phenotypic screening. This helps to screen huge numbers of compounds at fast rates and more economically in drug discovery processes. Molecular imaging techniques are also being used to study the pharmacokinetics of drug following its administration. On the other hand, in vivo imaging techniques serve as an important tool for studying the pathophysiology and efficacy of drug on the reversal processes in an evidence based manner. As mentioned earlier, the use of molecular imaging became is a pivotal method in preclinical oncology research, including the xenograft and orthotopic models. For example, in a mouse lymphoma xenograft model, the effect of temozolomide on tumor progression and the prolongation of survival was studied. The tumor cells were transfected with the luciferase gene, enabling the visualization of lymphoma cells using bioluminescence (BL) imaging. The tumor growths between untreated and treated groups were studied with reference to the luciferase imaging (Kadoch *et al.*, 2009).

Biomarkers: Biomarker assays not only assist in patient selection and the design of clinical trials but also act as indicators of drug efficacy, toxicity and disease progression (Ganesalingam and Bowser, 2010).

This substantially increases their importance in drug discovery processes. These assays are initiated in cell lines and primary blood cells or isolated tissue cells and are used to validate in vitro modulation of target by the drug and the effect of drugs on cells that is routine cellular screening. Typically, these assays are initiated in human cell lines, primary blood cells or isolated tissue cells. Biomarker assays are also used to identify markers of drug sensitivity that are used in selection of patient populations. Biomarker assays for pharmacokinetic/pharmacodynamic (PK/PD) models have also been developed, profiling molecules prior to testing in longer term disease models as initial proofs of concept. PK/PD models can also assist in dose-to-man scaling predictions for use in clinical trials. The biomarker assays developed during the in vitro discovery phases are frequently used as efficacy or toxicity endpoints in the clinic. Clinical trials, particularly in oncology, are frequently designed around these biomarkers.

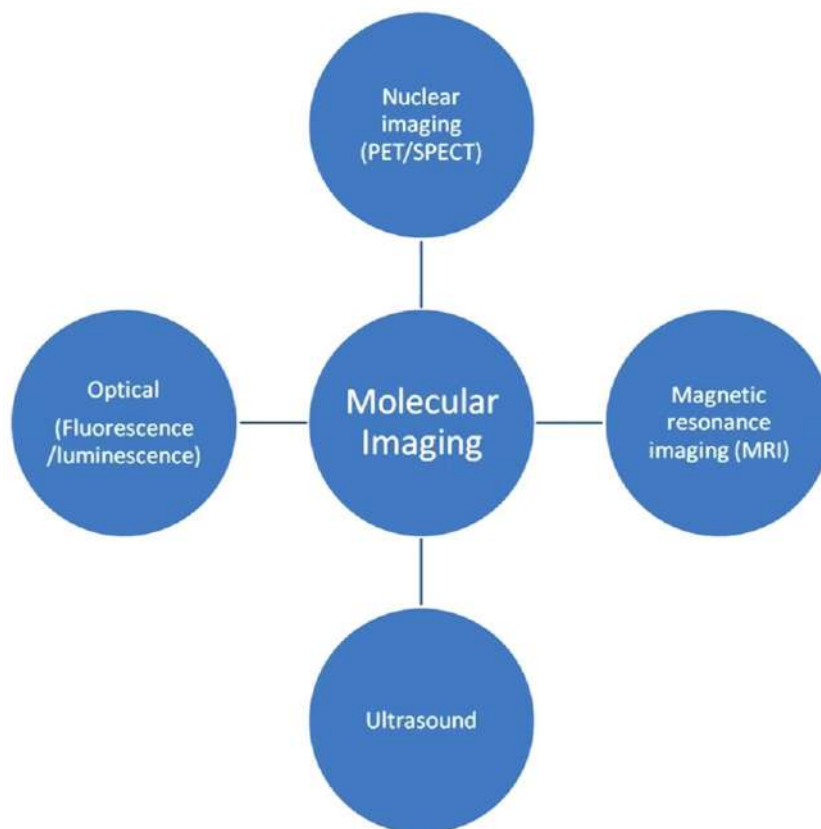


Fig. 3 Armamentarium of molecular imaging techniques employed to quantify target proteins. Nuclear imaging, magnetic resonance imaging, ultrasound imaging, and optical imaging techniques have been categorized as repetitive and non-invasive molecular imaging techniques.

Conclusions

Target identification and validation is costly and labour intensive. Failures in clinical trials are likely to have multifactorial reasons. However, selecting the most biologically plausible molecular targets that are relevant to the disease state is a critical first step to improve the probability of success. Towards this end, it is key to get a complete picture of the target(s), their role in a biological process and how they can be modulated to achieve the required output.

See also: Chemoinformatics: From Chemical Art to Chemistry in Silico. Comparative Epigenomics. Functional Enrichment Analysis. Gene Prioritization Using Semantic Similarity. Integrative Bioinformatics. Introduction of Docking-Based Virtual Screening Workflow Using Desktop Personal Computer. Molecular Mechanisms Responsible for Drug Resistance. Natural Language Processing Approaches in Bioinformatics. Rational Structure-Based Drug Design. Structure-Based Drug Design Workflow. Transmembrane Domain Prediction

References

- Bolon, B., 2004. Genetically engineered animals in drug discovery and development: A maturing resource for toxicological research. *Basic and Clinical Pharmacology and Toxicology* 95, 154–161.
- Chen, X., Ji, Z.L., Chen, Y.Z., 2002. TTD: Therapeutic target database. *Nucleic Acids Research* 30, 412–415.
- Chen, X., Yan, C.C., Zhang, X., *et al.*, 2016. Drug-target interaction prediction: Databases, web servers and computational models. *Briefings in Bioinformatics* 17, 696–712.
- Ertl, P., 2003. Cheminformatics analysis of organic substituents: Identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *Journal of Chemical Information and Modeling* 43, 374–380.
- Finan, C., Gaulton, A., Kruger, F.A., *et al.*, 2017. The druggable genome and support for target identification and validation in drug development. *Science Translational Medicine* 9.
- Ganesalingam, J., Bowser, R., 2010. The application of biomarkers in clinical trials for motor neuron disease. *Biomarkers in Medicine* 4, 281–297.
- Gregori-Puigjane, E., Setola, V., Hert, J., *et al.*, 2012. Identifying mechanism-of-action targets for drugs and probes. *Proceedings of National Academy of Science* 109, 11178–11183.
- Hajduk, P.J., Huth, J.R., Tse, C., 2005. Predicting protein druggability. *Drug Discovery Today* 10, 1675–1682.
- Hopkins, A.L., Groom, C.R., 2002. The druggable genome. *Nature Reviews Drug Discovery* 1, 727–730.
- Kadoch, C., Dinca, E.B., Voicu, R., *et al.*, 2009. Pathologic correlates of primary central nervous system lymphoma defined in an orthotopic xenograft model. *Clinical Cancer Research* 15, 1989–1997.
- Kangueane, P., Sowmya, G., Anupriya, S., *et al.*, 2015. Short peptide vaccine design and development: Promises and challenges. In: *Global Virology I-Identifying and Investigating Viral Diseases*. Springer.
- Lamb, J., Crawford, E.D., Peck, D., *et al.*, 2006. The Connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313, 1929–1935.
- Landry, Y., Gies, J.P., 2008. Drugs and their molecular targets: An updated overview. *Fundamental and Clinical Pharmacology* 22, 1–18.
- Lee, B.K.B., Tiong, J.K., Chang, J.K., *et al.*, 2017. DeSigN: Connecting gene expression with therapeutics for drug repurposing and development. *BMC Genomics* 18, 934.
- Liu, Y., Wei, Q., Yu, G., *et al.*, 2014. DCDB 2.0: A major update of the drug combination database. Database, Oxford 2014, bau124.
- Nowotka, M.M., Gaulton, A., Mendez, D., *et al.*, 2017. Using ChEMBL web services for building applications and data processing workflows relevant to drug discovery. *Expert Opinion on Drug Discovery* 12, 757–767.
- Ofran, Y., Punta, M., Schneider, R., Rost, B., 2005. Beyond annotation transfer by homology: Novel protein-function prediction methods to assist drug discovery. *Drug Discovery Today* 10, 1475–1482.
- Overington, J.P., Al-Lazikani, B., Hopkins, A.L., 2006. How many drug targets are there? *Nature Reviews Drug Discovery* 5, 993–996.
- Perumal, D., Lim, C.S., Sakharkar, M.K., 2009. A comparative study of metabolic network topology between a pathogenic and a non-pathogenic bacterium for potential drug target identification. *Summit on Translational Bioinformatics* 2009, 100–104.
- Plewczynski, D., Rychlewski, L., 2009. Meta-basis estimates the size of druggable human genome. *Journal of Molecular Modeling* 15, 695–699.
- Qin, C., Zhang, C., Zhu, F., *et al.*, 2014. Therapeutic target database update 2014: A resource for targeted therapeutics. *Nucleic Acids Research* 42, D1118–D1123.
- Sakharkar, M.K., Sakharkar, K.R., 2007. Targetability of human disease genes. *Current Drug Discovery Technology* 4, 48–58.
- Sakharkar, M.K., Sakharkar, K.R., Pervaiz, S., 2007. Druggability of human disease genes. *International Journal of Biochemistry and Cell Biology* 39, 1156–1164.
- Shashni, B., Sharma, K., Singh, R., *et al.*, 2013. Coffee component hydroxyl hydroquinone (HHQ) as a putative ligand for PPAR gamma and implications in breast cancer. *BMC Genomics* 14 (Suppl. 5), S6.
- Sterling, T., Irwin, J.J., 2015. ZINC 15 – Ligand discovery for everyone. *Journal of Chemical Information and Modeling* 55, 2324–2337.
- Thorn, C.F., Klein, T.E., Altman, R.B., 2013. PharmGKB: The pharmacogenomics knowledge base. *Methods in Molecular Biology* 1015, 311–320.
- Vasailkar, S., Bhatia, P., Bhatia, P.G., Chu Yaiw, K., 2016. Complementary approaches to existing target based drug discovery for identifying novel drug targets. *Biomedicine* 4, 27.
- Vincent, F., Loria, P., Pregel, M., *et al.*, 2015. Developing predictive assays: The phenotypic screening “rule of 3”. *Science Translational Medicine* 7, 293ps15.
- Wishart, D.S., Feunang, Y.D., Guo, A.C., *et al.*, 2017. DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Research*.
- Wishart, D.S., Knox, C., Guo, A.C., *et al.*, 2008. DrugBank: A knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research* 36, D901–D906.
- Zhao, B., Sakharkar, K.R., Lim, C.S., Kangueane, P., Sakharkar, M.K., 2007. MHC-Peptide binding prediction for epitope based vaccine design. *International Journal of Integrative Biology* 1, 127–140.

Relevant Websites

<https://clue.io/cmap>
CMap.
<http://www.disgenet.org/web/DisGeNET/menu/downloads>
DisGeNET.
<http://dgidb.org>
DGIdb.

Biomedical Text Mining

Hagit Shatkay, University of Delaware, Newark, DE, United States

© 2019 Elsevier Inc. All rights reserved.

1	Introduction	1099
2	Natural Language Processing and Information Extraction	1099
2.1	Basic Concepts and Challenges in Natural Language Processing	1100
2.2	Information Extraction: From Unstructured Text to Structured Data	1101
3	Information Retrieval and Text Classification	1101
3.1	Boolean Queries and Index Structures	1102
3.2	Similarity Queries: Documents in Vector Space	1102
3.3	Text Categorization: Classification, Clusters, and Topic Models	1103
4	Biomedical Applications, Shared Tasks, and Current/Future Challenges	1104
4.1	Information Retrieval and Extraction Tasks	1104
4.2	Community Challenges and Shared Tasks	1104
4.3	Knowledge Discovery Through Text and Future Directions	1105
	References	1106
	Further Reading	1109
	Relevant Websites	1109

1 Introduction

The vast majority of biomedical knowledge is conveyed and shared by means of written text, ranging from publications in journals and conferences, summary abstracts such as those stored in PubMed (PubMed, 2016), notes in electronic health records, drug labels, or short descriptive entries in biomedical databases (Chatr-Aryamontri *et al.*, 2015; Dowell *et al.*, 2009; Eppig *et al.*, 2015; The UniProt Consortium, 2012; Van Auken *et al.*, 2012). As text is typically provided in the form of natural language sentences, searching for and gleaning information conveyed within requires much human labor or computational tools that can sift through text and identify relevant words or passages. Developing and applying such tools for processing text and finding relevant information is typically referred to as Text Mining. This is a broad area of activity and research, involving several disciplines. A key discipline is natural language processing (NLP), and specifically information extraction (IE). It focuses on obtaining structured information from natural-language text, by identifying certain entities (e.g., genes, proteins, or diseases) and relations between them (e.g., phosphorylation, activation, repression) mentioned within the text. The entities and the relationships are typically converted into a standard form, such as ontological terms or numerical identifiers, and stored in a database. Ontological terms are terms listed within a controlled and carefully maintained vocabulary that denotes entities pertaining to a certain domain, along with relationships among these entities. The Gene Ontology (GO) is the most widely used ontology in the biological domain (Gene Ontology, 2016; The Gene Ontology Consortium, 2000), while the Medical Subject Headings (MeSH) is one of several controlled vocabularies used in medicine, covering organs, symptoms, diseases, and other conditions (MeSH, 2016).

Another major text mining task is the identification of publications or text passages that are relevant to individual users or user-communities. Examples include papers discussing gene expression in the mouse, which are of interest for curation by the Mouse Genome Database (Eppig *et al.*, 2015), or papers indicating side effects of a drug – which may be of interest to individual physicians. The latter may also be of interest for curation by the FDA (The United States Food and Drug Administration). This area of text analysis is known as information retrieval.

Another area related to information retrieval is that of text classification. It can be viewed as another way of identifying documents relevant to specific tasks; however, rather than viewing the goal as that of satisfying the different needs of an individual end user, the objective is to partition a collection of articles into individual subcategories based on topics of interest. For instance, one may partition documents into those that are relevant to breast cancer, those that pertain to prostate cancer, and those that are not relevant to any type of cancer. Each relevance-category is called a class, and the objective is to label each document by its appropriate class. Clearly, there are many ways to classify documents into different categories, and as such, text classification is applicable in a variety of contexts, and used for a wide range of text-related tasks.

Throughout this module we provide background about these different text-related tasks, along with examples of their application within the biomedical domain.

2 Natural Language Processing and Information Extraction

As noted above, biomedically-relevant text is available from many sources and repositories, ranging from sentences and comments stored as annotations, to curated genes, proteins or processes, to complete publications stored in publishers databases and in

PubMed/PubMed-central (2016). To effectively seek information within text, computational methods that perform NLP are often used (Jurafsky and Martin, 2009; Manning and Schütze, 1999).

2.1 Basic Concepts and Challenges in Natural Language Processing

Consider an application that tries to automatically seek information about a specific protein within text. Basic processing of the text's natural language requires first identifying orthographical features including word breaks, word-capitalization, and punctuation marks, which help identify words, phrases, or tokens within sentences (Tokens are character-sequences that are treated as a single unit, a word, or a term; the task of identifying them is known as tokenization). Fig. 1 shows an example of a sentence, along with the components comprising it. The example demonstrates the value of orthographic information for identifying specific entities of interest; the names of genes/proteins like *FSP27* (Fat Specific Protein 27) and *PLIN1* (prelipin 1) are typically shown as a short sequence of capital letters followed by a number.

Notably, orthographic rules are not hard-and-fast; while they serve as useful cues for identifying word boundaries in general, as well as genes and proteins in the biological domain, capitalization is often used for a variety of other purposes, indicating abbreviations, acronyms, or the start of a sentence. Similarly, hyphens, dashes, periods and other delimiters can serve a variety of purposes. The multiple roles such features play within a sentence make the identification of components a nontrivial task.

Another indicator of a word's meaning is its morphology, that is, its structure as a composition of smaller common units. For instance, the words *interaction*, *interacts*, and *interacting* all share the same root form *interact* and the same semantics, where the differences are relegated to the suffix. Conversely, certain suffixes can be used to convey a common semantic meaning, for example, the suffix “*ase*” often denotes an enzyme (e.g., *kinase*, *ligase*). Thus, morphological analysis including suffix stripping or suffix detection can be useful in identifying words that denote certain entity types or convey a similar meaning – such as enzymes, diseases, drugs, or interactions among these entities.

The meaning of a phrase or a sentence depends on relationships among words and the role of each word within a broader context. Breaking the sentence into its components – known as parsing, and identifying the role of words – known as part of speech (POS) tagging, are both fundamental steps in NLP. Fig. 1 illustrates the syntactic parsing of an example sentence, and the assignment of POS tags to its components: *FSP27* and *PLIN1* are both nouns, *interacts* is a verb, *in* and *with* are prepositions. The set of POS labels used in practice is much more comprehensive (see for instance the *Penn Treebank corpora* (Marcus et al., 1993), that include tens of thousands manually annotated sentences, employing tens of POS different tags).

Similar to the dependency of each word's meaning on its neighboring words, the meaning of a sentence as a whole also depends on surrounding sentences. That is, the meaning of any sequence of words or sentences is context-dependent. Discourse analysis (Afantenos et al., 2010; Jurafsky and Martin, 2009; Conrath et al., 2014; Dascalu, 2014) is an area within NLP focusing on the meaning conveyed within broader text regions through relations among multiple sentences.

Seeking the actual meaning conveyed in the text through NLP is a complex and challenging task. A key obstacle to understanding natural language, by a computer or by a human being, is ambiguity. Ambiguity manifests itself in every level of the language. Orthographical cues have multiple interpretations where capitalization can indicate a beginning of a sentence, a proper noun, or an acronym. At the word level, a single word may convey multiple meanings (polysemy, e.g., the word *fly* may serve as a noun denoting an animal or a verb denoting the flight action); at the sentence-level, syntactic ambiguity occurs where a sentence may be parsed in more than one way conveying different intentions. Much work in NLP is dedicated to overcoming ambiguity, providing accurate tools for identifying terms within text, tagging and parsing sentences, and identifying types of discourse (see e. g., (Jurafsky and Martin, 2009; Manning and Schütze, 1999; Shatkey and Craven, 2012) for details and examples).

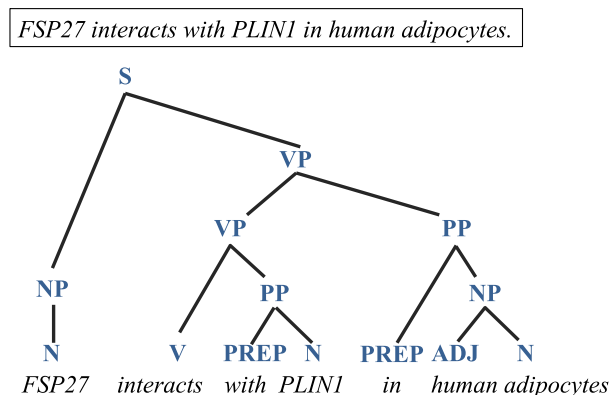
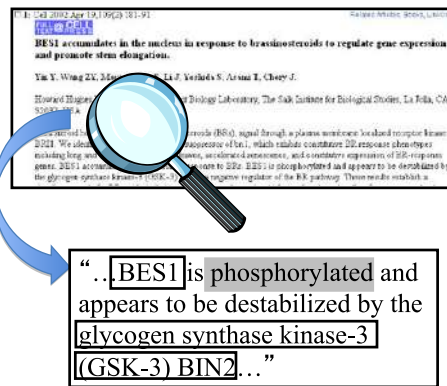


Fig. 1 A sentence (S) taken from a biomedical abstract (Grahn et al., 2014), along with its parse tree that shows parts-of-speech tags. Phrase tags, NP, VP, and PP correspond to Noun-, Verb-, and Prepositional-phrase respectively, while terminal tags – one layer above the bottom of the tree – indicate nouns (N), verbs (V), prepositions (PREP), and adjectives (ADJ). The leaves of the tree are the *terminals*, i.e., the words comprising the sentence.

A) Unstructured Data: Text from a publication



B) Structured Data: Phosphorylation Table

Kinase	Protein
<i>A-ase</i>	<i>P1</i>
<i>B-ase</i>	<i>P2</i>
...	...
...	...
...	...
<i>glycogen synthase kinase-3 (GSK-3) BIN2</i>	<i>BES1</i>
...	...

Fig. 2 (A) A biomedical paper (Yin et al., 2002) – shown as free, unstructured text (top), and a sentence within it discussing phosphorylation (bottom). (B) A structured phosphorylation table, showing a list of proteins and the kinase that phosphorylates each. The entry shown toward the bottom of the table is populated through information extraction methods applied to the sentence shown in A (bottom, left). The names of the identified protein and kinase are shown in dark frames, while the verb indicating phosphorylation is shown in gray.

The accuracy of NLP tools often depends on the domain in which they have been developed and employed (Ferraro et al., 2013; McClosky et al., 2010). Thus, as a basis for building NLP tools in the biological domain, specialized NLP resources providing annotations and tagging of biomedical sentences have been developed (Tateisi et al., 2005; Thompson et al., 2017) along with specialized tools for language analysis, for example (Cardie, 1997; Ferraro et al., 2013; Kang et al., 2011; Smith et al., 2004).

2.2 Information Extraction: From Unstructured Text to Structured Data

Two specific tasks that are often addressed within NLP and are of special interest within the biomedical domain are named entity recognition and relation extraction. The former means identifying mentions of relevant biological entities like genes, proteins, chemical compounds, drugs, diseases and many others; the latter refers to detecting relationships among these entities, such as reduction in drug efficacy due to a gene's activity, or interaction between two proteins or between two drugs. Gleaning such information from unstructured text and placing it in a database or within another structured format is known as IE (Cardie, 1997; Cowie and Lehnert, 1996).

Fig. 2 shows an illustration of IE applied to a text abstract. The task is the identification and extraction of phosphorylation relations between a kinase and a protein. The Protein and the Kinase entities are both identified, along with a verb that indicates a phosphorylation relationship. The entities are then harvested and placed in an entry of a structured phosphorylation table.

Named entity recognition is clearly needed for relation extraction (as illustrated in Fig. 2). It is also an important step toward indexing documents based on their contents so that documents discussing particular topics can be easily found, as discussed in Section 3.1. As such, much work in the field of biomedical text mining focuses on named entity recognition. It ranges from developing tools for gene- or protein-name extraction (Fluck et al., 2007; Leser and Hakenberg, 2005; Settles, 2005; Tanabe and Wilbur, 2002) to chemical- drug- or disease-identification (Batista-Navarro et al., 2015; Krallinger et al., 2015; Kuhn et al., 2016; Leaman et al., 2013; Lowe and Sayle, 2015; Savova et al., 2010). These methods rely on matching word-patterns to dictionaries containing curated entity-names, as well as on rules and machine learning methods, which are designed to distinguish named-entities from other words and terms. Extraction work is also being conducted in order to identify relations and statements pertaining to drug-interaction or other relationships among biological entities, for example (Kolchinsky et al., 2010; Kuhn et al., 2016; Savova et al., 2010; Comeau et al., 2014; Howe et al., 2016).

Notably, extraction efforts aim to identify explicit names and statements within text that contains relevant information. Identifying the publications and text passages that are indeed relevant to a certain subject matter is a task known as information retrieval, which is discussed next.

3 Information Retrieval and Text Classification

Information retrieval aims to identify a set of relevant documents in a large body of text. Commonly used search engines, like Google or PubMed, perform information retrieval known as ad hoc retrieval. The basic search mechanism in this case relies on a query specified by the user, where the query consists of a search term or a Boolean combination of terms. The search engine scans the large document collection and retrieves all documents that satisfy the query. The documents are then displayed in an order that is based on certain criteria such as temporal order, relevance, importance, or the number of citations.

Boolean term-combinations can be a cumbersome way to express a query, often resulting in many false-positives and false-negatives. A complementary form of ad hoc retrieval relies on similarity queries, within what is known as the vector space

(Manning and Raghavan, 2008; van Rijsbergen, 1979; Salton, 1989; Sparck-Jones *et al.*, 2000). Under this framework, a set of terms or words – which can even be a complete document – constitutes the query. The latter is represented as an algebraic vector in high-dimensional space; the vector is then compared against the text collection by employing a vector-similarity measure. The documents retrieved are those whose vector representation is most similar to that of the query vector.

Another aspect of information retrieval is text classification (e.g., Sebastiani, 2002; Yang and Liu, 1999). Rather than issue queries against the whole collection, the documents are split into different categories, where each category corresponds to a topic of interest. For instance, the categories may be defined a priori, where a collection of medical publications can be partitioned based on the disease discussed; here documents discussing kidney disease form a separate class from those discussing pneumonia. Categories may also be automatically uncovered by the categorization system, a process known as clustering. Throughout the rest of this section we discuss several mechanisms supporting the different types of queries, and provide examples for their use.

3.1 Boolean Queries and Index Structures

The basic mechanism that enables most information retrieval engines and Boolean queries in particular, is the index. As noted earlier, any document is initially processed by breaking it into its constituent terms, which can be individual words (also called unigrams), pairs of consecutive words (bigrams), longer sequences of consecutive words (n-grams), or grammatical phrases. An index is a data structure mapping each of the terms to all its occurrences within documents in the text collection. An illustration of an index structure is shown in Fig. 3. A Boolean query is executed by scanning through the index structure for the query terms and identifying the documents that satisfy the required Boolean combination of term occurrences (Manning and Raghavan, 2008; Witten *et al.*, 1999).

The basic index structure contains terms occurring in the document collection. However, other terms denoting fundamental concepts can be associated with the documents and included in the index. These may include, for instance, terms from the National Library of Medicine's MeSH controlled vocabulary (MeSH, 2016) assigned by human curators to documents, or terms from the GO (Gene Ontology, 2016; The Gene Ontology Consortium, 2000) that indicate properties of the proteins mentioned in the publication.

Boolean queries, expressed as terms connected by Boolean operators, (e.g., "BRCA1 AND Cancer") are supported in a straightforward way by index structures as shown in Fig. 3, albeit in their basic form can be ineffective for satisfying domain-specific information needs. Their limitations stem primarily from the inherent ambiguity of natural language as discussed in Section 1. Firstly, many terms occur often in a very large number of documents, thus the set of documents satisfying a simple Boolean query is too large for a user to practically scan through. Secondly, a query term typically has multiple meanings, varying by context (e.g., expression has a different meaning in the context of "gene expression" vs. "facial expression," while fly in the context of "fruit fly" carries a different meaning than in the context "birds fly"). Thus a large portion of documents that satisfy the literal query may end up being irrelevant to the user's actual need. Thirdly, documents that are relevant but use different words from the ones mentioned in the query (e.g., synonyms of the query terms) to denote the same idea, are not retrieved.

The vector space model, discussed next, offers an alternative mechanism, relying less on explicit or specific words and more on the whole context and the semantics denoted by words and phrases.

3.2 Similarity Queries: Documents in Vector Space

Naturally, we think about and visualize documents as sequences of words, terms, or sentences. As a step toward applying computational methods to text, we abstract away from our own use of documents, and view both documents and queries as

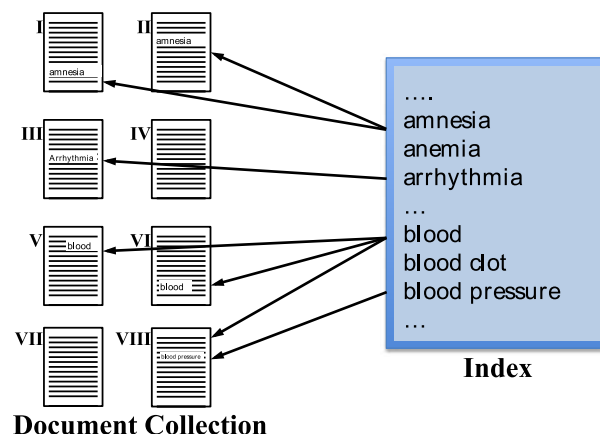


Fig. 3 An example of an index structure. The term entries in the index (right) are sorted alphabetically, and each term entry references the documents in which the term appears. Such a structure supports fast query processing, as documents containing the query terms can be quickly identified and accessed.

formal mathematical entities represented as algebraic vectors of weights, where each weight reflects the significance of an individual term. That is, every document d is represented as an N -dimensional vector of term-weights $\langle w_{t_1}^d, \dots, w_{t_N}^d \rangle$, where $t_1, \dots, t_N$ are the terms in the vocabulary and $w_{t_i}^d$ is a weight reflecting the significance of term t_i within the document d or the set of documents. A query q is likewise represented as a vector $\langle w_{t_1}^q, \dots, w_{t_N}^q \rangle$.

The retrieval task thus amounts to finding the document-vectors most similar to the query vector. Such queries are supported, for instance, by PubMed (PubMed, 2016) – where one can click on the link “Similar articles” next to a retrieved article and obtain links to all articles that share similar words or contents. Similarity-based retrieval methods vary across the specific choice of terms included in the vocabulary, the weighting scheme used to assign weights to these terms, and the similarity measure employed for comparing the vectors.

A simple weighting scheme is to assign a binary value (0 or 1) to each term, corresponding to its presence or absence in the document. Other weighting strategies are often applied, guided by several intuitive principles (Sparck-Jones *et al.*, 2000): (1) A term occurring frequently within a document, is typically important in the document’s context, and its weight should reflect this. (Local Term Frequency); (2) A term occurring in a very large number of documents typically does not characterize any specific document, and as such is less significant (Inverse Document Frequency); (3) A term occurring a few times in a short document is more significant to the document than if it occurs the same number of times in a long document (Inverse Document Length).

Weighting schemes that employ these principles thus include Local term frequency – the number of times the term occurs in the document, and variations on what is known as TF $\times$ IDF (Term Frequency $\times$ Inverse Document Frequency); the latter is a product of some function of the Local term frequency and the inverse of the number of documents within the collection that contain the term. Detailed weighting scheme strategies are discussed in the information retrieval literature (Manning and Raghavan, 2008; Salton, 1989; Sparck-Jones *et al.*, 2000; Witten *et al.*, 1999; Wilbur and Yang, 1996), where (Wilbur and Yang, 1996) is particularly relevant to the biomedical literature.

Once documents and queries are represented in vector space, vector-similarity is calculated to assess similarity between pairs of documents or between a query and each document in the collection. A commonly used vector-similarity measure is the cosine coefficient, which is the cosine of the angle between two vectors. For vectors $V_1 = \langle w_1^1, \dots, w_N^1 \rangle$ and $V_2 = \langle w_1^2, \dots, w_N^2 \rangle$, it is calculated as: $(\sum_{i=1}^N w_i^1 \cdot w_i^2) / (\|V_1\| \cdot \|V_2\|)$, where $(\|V_1\| \cdot \|V_2\|)$ are the norms of the vectors V_1, V_2 , respectively.

Several approaches, discussed next, take advantage of the vector model, using it to group together articles that share a common topic or theme. Some of these approaches use it directly, while others utilize a probabilistic interpretation of the weight-vector.

3.3 Text Categorization: Classification, Clusters, and Topic Models

One way to obtain documents that are relevant to a specific area of interest is to pre-categorize documents by such areas. For instance, as illustrated in Fig. 4, documents can be grouped according to the disease or other medical subjects of interest discussed in them. When a user looks for documents relevant to such a subject or area, all documents that are categorized as belonging to this area can be retrieved and returned.

A variety of machine learning methods are applied to address this task (Cohen and Singer, 1999; Dumais *et al.*, 1998; Joachims, 1998; Lewis, 1998; Lewis *et al.*, 2004; Sebastiani, 2002; Xu *et al.*, 2016; Yang and Liu, 1999). We distinguish between two types of categorization approaches: supervised categorization, known as classification, and unsupervised, known as clustering. In classification, the set of class labels is fixed and a set of pre-categorized documents, regarded as training examples, is available. A classifier is learned from the training examples, where the goal is to correctly assign class labels to documents outside the training set. Clustering, on the other hand, aims to identify subsets (clusters) of interrelated documents within the dataset, while separating unrelated documents into disjoint clusters – without having any predefined labels or pre-categorized training examples. Much of the categorization work done in the biomedical domain is concerned with supervised learning, i.e., classification. The methods that are often used in practice include decision trees (Mitchel, 1997; Quinlan, 1993), their extension to Random Forests (Breiman, 2001), naïve Bayes (Mitchel, 1997; Lewis, 1998; Sohn *et al.*, 2008), support vector machines (SVMs) (Joachims, 1998; Vapnik, 1995) and logistic regression (Schütze *et al.*, 1995; Vlachos and Craven, 2010).

Text categorization is based on the idea of similarity, where documents that are similar according to a certain similarity criterion are grouped together, while those that are dissimilar are placed in separate categories. In Section 3.2 we have discussed the vector representation for documents, along with the cosine similarity measure. This measure can be applied both in the

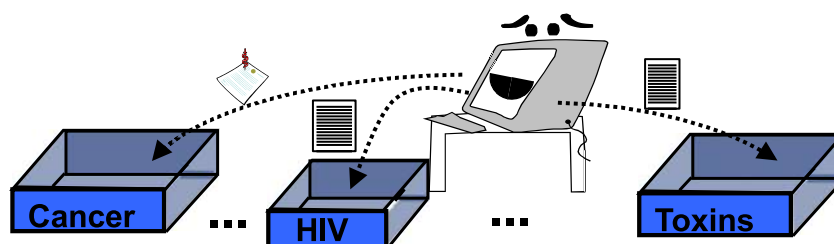


Fig. 4 Automatic text categorization. Documents are processed and assigned to a specific category based on their contents.

context of query-answering and for categorization. Another interpretation of the vector model views each weight w_i^j of a term t_i in a document d^j as reflecting the probability of the term to occur in the document. Under such probabilistic interpretation, the information retrieval task amounts to finding documents that with high probability satisfy the information need expressed by the query. Probabilistic models of retrieval have been introduced by Van Rijsbergen and others several decades ago (van Rijsbergen, 1977; Sparck-Jones *et al.*, 2000), and were further developed and formalized as language models (Ponte and Croft, 1998) and, more recently, as topic models (Hofmann, 1999; Shatkey and Wilbur, 2000; Blei *et al.*, 2003; Blei, 2012). Under a simple language model, we view a document as though it was produced by sampling terms from a language, where the language is modeled as a multinomial distribution over terms. Within the language-model framework, similarity between documents is expressed as their probability to have been generated by the same language model. A theme or a topic model (Shatkey and Wilbur, 2000; Blei *et al.*, 2003) extends the basic model, by having documents viewed as sampled from a mixture of several multinomial distributions, where the mixture defines a combination of subjects that are discussed within the same context. Some of the early work on theme models was developed specifically for analyzing biomedical literature, aiming to support gene-expression interpretation and enrichment-analysis for certain genomic functions (Shatkey and Craven, 2012; Shatkey and Wilbur, 2000). A more in-depth discussion of probabilistic, language and topic models can be found in the literature cited above as well as in more comprehensive books (van Rijsbergen, 1979; Manning and Raghavan, 2008; Shatkey and Craven, 2012).

Within the biomedical domain, much work on text categorization is concerned with supporting biomedical data-curation for public databases. For example, organism-specific databases including the Mouse Genome Informatics databases maintained by the Jackson Lab (Dowell *et al.*, 2009; Eppig *et al.*, 2015), WormBase (Van Auken *et al.*, 2012) and others, store information about gene expression or the effects of genetic mutations in the respective organisms. The information is obtained by curators who scan the relevant literature. Automatically identifying the relevant body of literature is a text-categorization task: a set of documents relevant to a specific database forms a class, while irrelevant documents form another class. Curators for databases that store information about proteins, their properties, and their interactions, such as BioGrid (Chatr-Aryamontri *et al.*, 2015), IntAct (Kerrien *et al.*, 2012) or UniProt (The UniProt Consortium, 2012) face a similar challenge. Such categorization of documents by relevance, is also known as triage, and has been the focus of work and shared tasks within the biomedical text mining community (e.g., Hersh *et al.*, 2006; Hirschman *et al.*, 2012).

4 Biomedical Applications, Shared Tasks, and Current/Future Challenges

We have already touched on several text-based methods applicable in biomedicine. This section provides a brief overview of text-related tasks that have been addressed and text-based methods that have been applied in biology and medicine – both early and recent.

4.1 Information Retrieval and Extraction Tasks

In terms of information retrieval, PubMed (2016) is the most comprehensive and widely used biomedical text-retrieval system. It supports Boolean queries, similarity queries, as well as refinement of the retrieval task utilizing pre-classification of the articles by species (e.g., Human, Other Animals), publication type (e.g., Review, Clinical Trial), and additional categories of common interest. Other large broadly-used biomedical data resources, such as the protein database UniProt (The UniProt Consortium, 2012), support text-based Boolean queries, which provide access to annotated entries based on the text description and GO terms associated with these entries.

Much other information-retrieval work involves text categorization. For example, identifying articles discussing mouse genomics is a text-categorization task supporting mouse-related curation. This task was one of the first text-related shared tasks, as further discussed below (Hersh *et al.*, 2006; Hersh, 2008). Other examples of text categorization include identifying documents and sentences discussing protein-interaction or protein-characterization (Denroche *et al.*, 2010; Donaldson *et al.*, 2003; Kolchinsky *et al.*, 2010; Krallinger *et al.*, 2011; Xu *et al.*, 2009), adverse drug reaction (Duda *et al.*, 2005; Sarker and Gonzalez, 2015), and many others.

In terms of fine-grained extraction and NLP, a lot of effort has been dedicated to identifying bio-entities – genes, proteins, drugs, chemicals and diseases, and on finding statements that link between such entities. The very first applications of text mining in biology (Leek, 1997; Blaschke *et al.*, 1999; Craven and Kumlien, 1999), specifically looked to identify protein names within sentences searching for statements of protein–protein interactions (Blaschke *et al.*, 1999), and for locations of proteins within the cell and location of protein-coding genes on the chromosome (Craven and Kumlien, 1999; Leek, 1997). This pioneering work was followed by much research and progress, improving gene- and protein-name identification (Fluck *et al.*, 2007; Leser and Hakenberg, 2005; Settles, 2005; Tanabe and Wilbur, 2002), and the recognition of other biomedical entities and relationships (Comeau *et al.*, 2014; Friedman, 2009; Hunter *et al.*, 2008; Kolchinsky *et al.*, 2010; Krallinger *et al.*, 2015; Kuhn *et al.*, 2016; Leaman *et al.*, 2013; Lowe and Sayle, 2015).

4.2 Community Challenges and Shared Tasks

The growing interest in systems that utilize text resources within biomedicine, along with the importance of assessing their utility toward meeting biologically-relevant challenges, has given rise to several shared tasks. In such shared tasks, a text-related challenge is proposed to the community by the challenge organizers, and teams come together to address the challenge. A gold standard

consisting of expert-annotated document sets is made available to the community, and the systems that are being developed are trained and tested over the provided datasets. The latter are typically formed by a well-recognized organization that maintains and curates large biomedical data resources, while the annotators are typically biologists, database-curators or physicians who have much expertise and experience in the relevant area. Performance measures that enable quantitative evaluation of the systems are also decided on as part of the challenge definition. While standard performance measures – accuracy, precision, recall, and variations over them are often used (see e.g., (Shatkey and Craven, 2012), Chapter 5.1), certain tasks require developing specific performance evaluation metrics.

The first shared task on biological text mining was introduced as part of the KDD-cup 2002 (Yeh *et al.*, 2002), where the challenge involved identifying evidence for gene expression in the fruit fly. This event was followed by several series of long-term organized tasks. One series, TREC-Genomics (Hersh *et al.*, 2006; Hersh, 2008; Roberts *et al.*, 2009) was introduced as part of the well-established text retrieval conferences (TREC) sponsored by the U.S. National Institutes of Standards and Technology (NIST) (TREC, 2016). It focused primarily on ad hoc retrieval (that is, addressing a variety of user queries), particularly searching for documents discussing specific aspects of genes and proteins (e.g., gene function, role in disease, expression within certain tissue types). TREC-Genomics was followed by additional TREC medical/clinical tracks for addressing retrieval from electronic health records and support clinical decision-making (Simpson *et al.*, 2014; Voorhees and Hersh, 2012).

Another important series of shared tasks, BioCreative (BioCreative, 2016; Hirschman *et al.*, 2005), focuses primarily on biological text mining in support of database curation. Aiming to facilitate Critical Assessment of IE in Biology, BioCreative has been running regularly since 2004, presenting challenges ranging from identifying named-entities such as genes or chemicals (Krallinger *et al.*, 2015), through finding evidence for protein–protein interactions (Krallinger *et al.*, 2011) to creating effective interfaces for textual annotation by curators (Wang *et al.*, 2016; Hirschman *et al.*, 2012). As high-quality ground-truth text annotations are essential for both training text mining systems and for assessing their performance throughout the shared challenges, using crowd-sourcing for obtaining reliable annotations is an important recent direction that is being studied as part of biological text mining (Hirschman *et al.*, 2016). Additional recent area-specific shared tasks include the i2b2 (Informatics for Integrating Biology and the Bedside) NLP tasks (i2b2, 2016), BioNLP challenges (BioNLP Shared Task, 2016) primarily affiliated with the Association for Computational Linguistics, and BioASQ (BioASQ, 2016; Tsatsaronis *et al.*, 2015) targeting question-answering and extraction of certain types of information from text records or the literature. Additional information about such shared tasks can be found recently published survey on the topic (Huang and Lu, 2016).

4.3 Knowledge Discovery Through Text and Future Directions

The basic form of automated text mining revolves around finding within the text facts and assertions that have already been uncovered and published. However, a highly useful and less conventional way of leveraging published text is by identifying indirect links that expose hitherto unknown information, giving rise to new knowledge.

Swanson's pioneering work (Swanson, 1986; Swanson, 1990; Swanson *et al.*, 2001) introduced the idea of uncovering new links between concepts or entities, by following indirect connections among them. A canonical example illustrating the idea was his showing that Raynaud's syndrome can be treated by increasing the intake of fish oil (Swanson, 1986). To do so he pointed out two separate recurring statements in the literature: (1) fish oil can reduce blood viscosity and (2) Raynaud's syndrome is characterized by increased blood viscosity. While no previous publication stated these two facts together, Swanson established the connection, noting the overlapping concept blood viscosity occurring in both the context of fish oil and the context of Raynaud's syndrome. He thus proposed that fish oil may be used for treating Raynaud's syndrome. The idea is illustrated in Fig. 5. Following Swanson's uncovering the putative connection between fish oil and Raynaud's treatment, an independent clinical study

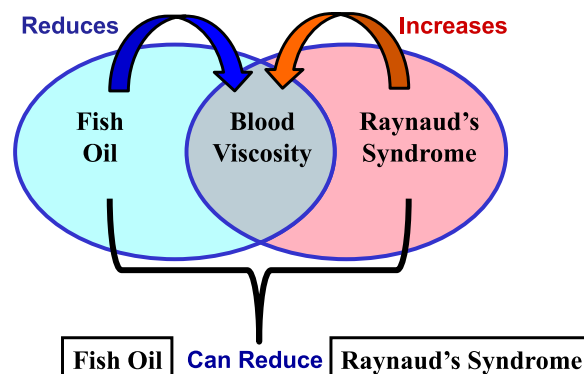


Fig. 5 Example of a relationship discovery by connecting two indirect link. Swanson established a hitherto unknown relationship between fish oil intake and reduction in Raynaud's Syndrome, by noting that the literature concerning Raynaud's syndrome (shown in pink on the right) reports an increase in blood viscosity, while the literature about fish oil (blue, left) reports reduction in blood viscosity. The gray area in the middle indicates the common concept, blood viscosity, which establishes the connection between the other two terms.

corroborated this link (DiGiacomo *et al.*, 1989). The framework has been implemented in Swanson's Arrowsmith discovery system (Swanson *et al.*, 2001), and further extended by others in later years toward literature-based knowledge discovery (Lee *et al.*, 2007; Srinivasan, 2004; Srinivasan and Libbus, 2004).

Another way of uncovering new knowledge is by treating text as raw data, while searching for common motifs and patterns within it (Shatkay *et al.*, 2015; Shatkay *et al.*, 2000). This use of text is similar to the way in which homology and recurring motifs are scanned for on genomic and proteomic sequences, aiming to reveal relationships among such biological entities. Like genomic sequences, the text is viewed as a sequence of symbols, words, terms, phrases, or grammatical objects, and the statistical properties of their distribution across documents are used to expose similarities and connections among different text passages. Unlike most biological data, text components have readily understandable semantics, making results of such statistical pattern analysis easy to interpret and to validate. Early work establishing this direction included the utilization of statistical theme models in the gene-literature in order to identify functional relations among genes (Shatkay *et al.*, 2000). Similar ideas were employed for identifying sets of related genes or proteins by grouping together text passages discussing them (Chagoyen *et al.*, 2006; Renner and Aszodi, 2000).

A distinct but similar line of work directly employs text-based features associated with biological entities in order to predict biological properties such as protein homology, function, or subcellular location (Shatkay *et al.*, 2015; Chang *et al.*, 2001; Nair and Rost, 2002; Brady and Shatkay, 2008). Moreover, the integration of both text and sequence data into a unified prediction framework has shown very effective in improving prediction accuracy (Briesemeister *et al.*, 2009; Shatkay *et al.*, 2007). Thus, a direction that shows much promise in utilizing text is its integration as a source of data with other forms of biological data obtained from expression experiments, sequencing, or mutation analysis.

Another source of data that is currently being studied in conjunction with text is image information, where figures accompanying the text within medical records and publications are being examined as a source of additional knowledge (Ahmed *et al.*, 2010; Bockhorst *et al.*, 2012; Cohen *et al.*, 2003; Demner-Fushman *et al.*, 2012; Kalpathy-Cramer *et al.*, 2014; Ma *et al.*, 2015; Shatkay *et al.*, 2006; Xu *et al.*, 2008). This new direction that takes advantage of figures in addition to text holds much promise, both for supporting reliable curation of existing knowledge and for identifying significant pieces of information within the published literature.

See also: Bayes' Theorem and Naive Bayes Classifier. Data Mining: Mining Frequent Patterns, Associations Rules, and Correlations. Data-Information-Concept Continuum From a Text Mining Perspective. Knowledge Discovery in Databases. Natural Language Processing Approaches in Bioinformatics. Text Mining Applications. Text Mining Basics in Bioinformatics. Text Mining for Bioinformatics Using Biomedical Literature

References

- Afantenos, S., Denis, P., Muller, P., Danlos, L., 2010. Learning recursive segments for discourse parsing. In: Proceedings of 7th Language Resources and Evaluation Conference (LREC'10), pp. 3578–3584.
- Ahmed, A., Arnold, A., Coelho, L.P., *et al.*, 2010. Structured literature image finder: Parsing text and figures in biomedical literature. *Web Semantics: Science, Services and Agents on the World Wide Web* 8 (2), 151–154.
- Batista-Navarro, R., Rak, R., Ananiadou, S., 2015. Optimizing chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. *Journal of Cheminformatics* 7 (Suppl 1), S6.
- BioASQ, 2016. Available at: <http://bioasq.org/>
- BioCreative, 2016. BioCreative: Critical assessment of information extraction in biology. Available at: <http://www.biocreative.org/>
- BioNLP Shared Task, 2016. Available at: <http://www.bionlp-st.org>
- Blaschke, C., Andrade, M., Ouzounis, O., Valencia, A., 1999. Automatic extraction of biological information from scientific text: Protein–protein interactions. In: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99), aaAI Press, pp. 60–67.
- Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J., 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022.
- Blei, D.M., 2012. Probabilistic topic models. *Communications of the ACM* 55 (4), 77–84.
- Bockhorst, J.P., Conroy, J.M., Agrawal, S., O'Leary, D.P., Yu, H., 2012. Beyond captions: Linking figures with abstract sentences in biomedical articles. *PLOS ONE* 7 (7), 1–15.
- Brady, S., Shatkay, H., 2008. EpiLoc: A (working) text-based system for predicting protein subcellular location. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 604–615.
- Breiman, L., 2001. Random Forests. *Machine Learning* 45 (1), 5–32.
- Briesemeister, S., Blum, T., Brady, S., *et al.*, 2009. Sherlock2: A high-accuracy hybrid method for predicting subcellular localization of proteins. *Journal of Proteome Research* 8 (11), 5363–5366.
- Cardie, C., 1997. Empirical methods in information extraction. *AI Magazine* 18 (4), 65–80.
- Chagoyen, M., Carmona-Saez, P., Shatkay, H., Caraz, J.M., Pascual-Montano, A., 2006. Discovering semantic features in the literature: A foundation for building functional associations. *BMC Bioinformatics* 7, 41.
- Chang, J.T., Raychaudhuri, S., Altman, R.B., 2001. Including biological literature improves homology search. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 374–383.
- Chatr-Aryamontri, A., Breitkreutz, B.J., Oughtred, R., *et al.*, 2015. The BioGRID interaction database: 2015 update. *Nucleic Acids Research* 43 (Database Issue), D470–D478. Available at: <http://thebiogrid.org/>
- Cohen, W., Kou, Z., Murphy, R.F., 2003. Extracting information from text and images for location proteomics. In: Proceedings of the 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD'03), pp. 2–9.
- Cohen, W.W., Singer, Y., 1999. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems* 17 (2), 141–173.
- Comeau, D.C., Liu, H., Doğan, R.I., Wilbur, W.J., 2014. Natural Language processing pipelines to annotate BioC Collections with an Application to the NCBI Disease Corpus. Database. <http://dx.doi.org/10.1093/database/bau056>

- Conrath, J., Afantenos, S., Asher, N., Muller, P., 2014. Unsupervised extraction of semantic relations using discourse cues. In: Proceedings of the International Conference on Computational Linguistics (COLING'14), pp. 2184–2194.
- Cowie, J., Lehnert, W., 1996. Information extraction. *Communications of the ACM* 39 (1), 80–91.
- Craven, M., Kumlien, J., 1999. Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99), AAAI Press, pp. 77–86.
- Dascalu, M., 2014. Computational discourse analysis. *Analyzing Discourse and Text Complexity for Learning and Collaborating: Series on Studies in Computational Intelligence* (534). Switzerland: Springer, pp. 53–77. (Chapter 4).
- Demner-Fushman, D., Antani, S., Thoma, G.R., 2012. Design and development of a multimodal biomedical information retrieval system. *Journal of Computing Science and Engineering* 6 (2), 168–177.
- Denroche, R., Madupu, R., Yooseph, S., Sutton, G., Shatkay, H., 2010. Toward computer-assisted text curation: Classification is easy (choosing training data can be hard...). In: Blaschke, C., Shatkay, H. (Eds.), *Linking Literature, Information, and Knowledge for Biology*. Berlin, Heidelberg: Springer, pp. 33–42.
- DiGiacomo, R., Kremer, J., Shah, D., 1989. Fish-oil dietary supplementation in patients with Raynaud's phenomenon: A double-blind, controlled, prospective study. *The American Journal of Medicine* 86 (2), 158–164.
- Donaldson, I., Martin, J., Wolting, C., *et al.*, 2003. PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics* 4, 11.
- Dowell, K., McAndrews-Hill, M., Hill, D., Drabkin, H., Blake, J., 2009. Integrating text mining into the MGI biocuration workflow. Database. <http://dx.doi.org/10.1093/database/bap019>
- Duda, S., Aliferis, C., Miller, R., Statnikov, A., Johnson, K., 2005. Extracting drug–drug interaction articles from MEDLINE to improve the content of drug databases. In: Proceedings of the AMIA Annual Symposium, p. 216.
- Dumais, S.T., Platt, J., Heckerman, D., Sahami, M., 1998. Inductive learning algorithms and representations for text categorization. In: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp. 148–155.
- Eppig, J.T., Blake, J.A., Bult, C.J., Kadin, J.A., Richardson, J.E., and The Mouse Genome Database Group, 2015. The mouse genome database (MGD): Facilitating mouse as a model for human biology and disease. *Nucleic Acids Research* 43 (Database Issue), D726–D736.
- Ferraro, J.P., Daumé, H., DuVall, S.L., *et al.*, 2013. Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. *Journal of the American Medical Informatics Association* 20 (5), 931–939.
- Fluck, J., Mevissen, H.T., Dach, H., Oster, M., Hofmann-Apitius, M., 2007. ProMiner: Recognition of human gene and protein names using regularly updated dictionaries. In: Proceedings of Second BioCreative Challenge Evaluation Workshop, pp. 149–151.
- Friedman, C., 2009. Discovering novel adverse drug events using natural language processing and mining of the electronic health record. In: Proceedings of the 12th Conference on Artificial Intelligence in Medicine (AIME), pp. 1–5.
- Gene Ontology, 2016. Gene ontology consortium. Available at: www.geneontology.org
- Grahn, T.H.M., Kaur, R., Yin, J., *et al.*, 2014. Fat-specific protein 27 (FSP27) interacts with Adipose Triglyceride Lipase (ATGL) to regulate lipolysis and insulin sensitivity in human adipocytes. *Journal of Biological Chemistry* 289 (17), 12029–12039.
- Hersh, W., 2008. *Information Retrieval: A Health and Biomedical Perspective*. New York, NY: Springer-Verlag, (Chapter 9.2).
- Hersh, W.R., Cohen, A., Yang, J., *et al.*, 2006. TREC 2005 genomics track overview. In: Proceedings of the 14th Text Retrieval Conference – TREC'05, NIST Special Publication, pp. 14–25.
- Hirschman, L., Burns, G.A., Krallinger, M., *et al.*, 2012. Text mining for biocuration workflow. Database. <http://dx.doi.org/10.1093/database/bas020>
- Hirschman, L., Fort, K., Boue, S., *et al.*, 2016. Crowdsourcing and curation: Perspectives from biology and natural language processing. Database. <http://dx.doi.org/10.1093/database/baw115>
- Hirschman, L., Yeh, A., Blaschke, C., Valencia, A., 2005. Overview of BioCreAtIvE: Critical assessment of information extraction for biology. *BMC Bioinformatics* 6 (Suppl 1), S1.
- Hofmann, T., 1999. Probabilistic latent semantic indexing. In: Proceedings of the 22nd International Conference on Research and Development in Information Retrieval, (SIGIR'99), pp. 50–57.
- Howe, K.L., Bolt, B.J., Cain, S., *et al.*, 2016. WormBase 2016: Expanding to enable Helminth genomic research. *Nucleic Acids Research* 44 (Database Issue), D774–D780.
- Huang, C.C., Lu, Z., 2016. Community challenges in biomedical text mining over 10 years: Success, failure and the future. *Briefings in Bioinformatics* 17 (1), 132–144.
- Hunter, L., Lu, Z., Firby, J., *et al.*, 2008. OpenDMP: An open source, ontology-driven concept analysis engine with applications to capturing knowledge regarding protein transport, protein interactions and cell-type specific gene expression. *BMC Bioinformatics* 9, 78.
- i2b2, 2016. Informatics for integrating biology & the bedside. Available at: <https://www.i2b2.org/NLP/>
- Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the Tenth European Conference on Machine Learning, pp. 137–142.
- Jurafsky, D., Martin, J., 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, second ed. Englewood Cliffs, NJ: Prentice-Hall.
- Kalpathy-Cramer, J., de Herrera, A.G.S., Demner-Fushman, D., *et al.*, 2014. Evaluating performance of biomedical image retrieval systems – An overview of the medical image retrieval task at imageCLEF 2004–2014. *Comp. Medical Imaging and Graphics* 39, 55–61.
- Kang, N., van Mulligan, E.M., Kors, J.A., 2011. Comparing and combining chunkers of biomedical text. *Journal of Biomedical Informatics* 44 (2), 354–360.
- Kerrien, S., Aranda, B., Breuza, L., *et al.*, 2012. The IntAct molecular interaction database in 2010. *Nucleic Acids Research* 40 (D1), D841–D846. Available at: <http://www.ebi.ac.uk/intact/>
- Kolchinsky, A., Abi-Haidar, A., Kaur, J., Hamed, A.A., Rocha, L.M., 2010. Classification of protein–protein interaction full-text documents using text and citation network features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 7 (3), 400–411.
- Krallinger, M., Leitner, F., Rabal, O., *et al.*, 2015. CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics* 7 (1), 1.
- Krallinger, M., Vazquez, M., Leitner, F., *et al.*, 2011. The protein–protein interaction tasks of BioCreative III: Classification/ranking of articles and linking bio-ontology concepts to full text. *BMC Bioinformatics* 12 (Suppl 8), S3.
- Kuhn, M., Letunic, I., Jensen, L.J., Bork, P., 2016. The SIDER database of drugs and side effects. *Nucleic Acids Research* 44 (D1), D1075–D1079.
- Leaman, R., Doğan, R.I., Lu, Z., 2013. DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics* 29 (22), 2909–2917.
- Leek, T., 1997. Information extraction using hidden Markov models. Master's Thesis, Department of Computer Science and Engineering, University of California.
- Lee, W.J., Raschid, L., Srinivasan, P., *et al.*, 2007. Using annotations from controlled vocabularies to find meaningful associations. In: Proceedings of the Workshop on Data Integration in the Life Sciences, Lecture Notes in Computer Science, Springer, pp. 247–263.
- Leser, U., Hakenberg, J., 2005. What makes a gene name? Named entity recognition in the biomedical literature. *Briefings in Bioinformatics* 6 (4), 357–369.
- Lewis, D.D., Yang, Y., Rose, T., Li, F., 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5, 361–397.
- Lewis, D.D., 1998. Naïve (Bayes) at forty: The independence assumption in information retrieval. In: Proceedings of the 10th European Conference on Machine Learning (ECML'98), pp. 4–15.
- Lowe, D.M., Sayle, R.A., 2015. LeadMine: A grammar and dictionary driven approach to entity recognition. *Journal of Cheminformatics* 7 (1), S5.
- Manning, C., Raghavan, P., 2008. *Introduction to Information Retrieval*. New York, NY: Cambridge University Press.
- Manning, C., Schütze, H., 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

- Marcus, M.P., Santorini, B., Marcinkiewicz, M.A., 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics* 19 (2), 313–330.
- Ma K., Jeong H., Rohith M.V., *et al.* 2015. Utilizing image-based features in biomedical document classification. In: *Proceedings of the International Conference on Image Processing (ICIP'15)*, pp. 4451–4455.
- McClosky, D., Charniak, E., Johnson M., 2010. Automatic domain adaptation for parsing. In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL'10)*, pp. 28–36.
- MeSH, 2016. Medical Subject Headings. Available at: <https://www.nlm.nih.gov/mesh/>
- Mitchel, T., 1997. *Machine Learning*. New York, NY: McGraw-Hill.
- Nair, R., Rost, B., 2002. Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics* 18 (Suppl. 1), S78–S86.
- Ponte, J.M., Croft, W.B., 1998. A language modeling approach to information retrieval. In: *Proceedings of the 21st International Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 275–281.
- PubMed, 2016. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/> (accessed Oct 2016).
- Quinlan, J., 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann.
- Renner, A., Aszodi, A., 2000. High-throughput functional annotation of novel gene products using document clustering. In: *Proceedings of the Pacific Symposium on Biocomputing*, pp. 54–65.
- Roberts, P., Cohen, A.M., Hersh, W.R., 2009. Tasks, topics and relevance judging for the TREC genomics track: Five years of experience evaluating biomedical text information retrieval systems. *Information Retrieval* 12 (1), 81–97.
- Salton, G., 1989. *Automatic Text Processing*. Boston, MA: Addison-Wesley.
- Sarker, A., Gonzalez, G., 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of Biomedical Informatics* 53, 196–207.
- Savova, G.K., Masanz, J.J., Ogren, P.V., *et al.*, 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association* 17 (5), 507–513.
- Schütze, H., Hull, D.A., Pedersen, J.O., 1995. A comparison of classifiers and document representations for the routing problem. In: *Proceedings of the 18th International Conference on Research and Development in Information Retrieval (SIGIR'95)*, ACM, pp. 229–237.
- Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34 (1), 1–47.
- Settles, B., 2005. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics* 21 (14), 3191–3192.
- Shatkay, H., Brady, S., Wong, A., 2015. Text as data: Using text-based features for proteins representation and for computational prediction of their characteristics. *Methods* 74, 54–64. **Special Issue on Text Mining of Biomedical Literature.**
- Shatkay, H., Chen, N., Blostein, D., 2006. Integrating image data into biomedical text categorization. *Bioinformatics* 22 (14), e446–e453.
- Shatkay, H., Craven, M., 2012. *Mining the Biomedical Literature*. Cambridge, MA: MIT Press.
- Shatkay, H., Edwards, S., Wilbur, W.J., Boguski, M., 2000. Genes, themes and microarrays: Using information retrieval for large scale gene analysis. In: *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, pp. 317–328.
- Shatkay, H., Höglund, A., Brady, S., *et al.*, 2007. Sherloc: High-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics* 23 (11), 1410–1417.
- Shatkay, H., Wilbur, W.J., 2000. Finding themes in MEDLINE documents: Probabilistic similarity search. In: *Proceedings of the IEEE Conference on Advances in Digital Libraries*, pp. 183–192.
- Simpson, M.S., Voorhees, E., Hersh, W., 2014. Overview of the TREC 2014 clinical decision support track. In: *Proceedings of the 23rd Text Retrieval Conference – TREC'14*, NIST Special Publication.
- Smith, L., Rindflesch, T., Wilbur, W.J., 2004. MedPost: A part-of-speech tagger for bioMedical text. *Bioinformatics* 20 (14), 2320–2321.
- Sohn, S., Kim, W., Comeau, D.C., Wilbur, W.J., 2008. Optimal training sets for Bayesian prediction of MeSH assignment. *Journal of the American Medical Informatics Association* 15 (4), 546–553.
- Sparck-Jones, K., Walker, S., Robertson, S., 2000. A probabilistic model of information retrieval: Development and status. *Information Processing and Management* 36 (6), 779–840.
- Srinivasan, P., 2004. Text mining: Generating hypotheses from MEDLINE. *Journal of the American Society for Information Science (JASIS)* 55 (5), 396–413.
- Srinivasan, P., Libbus, B., 2004. Mining MEDLINE for implicit links between dietary substances and diseases. *Bioinformatics* 20 (Suppl. 1), i290–i296.
- Swanson, D.R., 1986. Fish-oil, Raynaud's syndrome and undiscovered public knowledge. *Perspectives in Biology and Medicine* 30 (1), 7–18.
- Swanson, D.R., 1990. Somatomedin C and arginine: Implicit connections between mutually isolated literatures. *Perspectives in Biology and Medicine* 33 (2), 157–186.
- Swanson, D.R., Smalheiser, N.R., Bookstein, A., 2001. Information discovery from complementary literatures: Categorizing viruses as potential weapons. *Journal of the American Society for Information Science and Technology* 52 (10), 797–812.
- Tanabe, L., Wilbur, W.J., 2002. Tagging gene and protein names in full text articles. In: *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, vol. 3, pp. 9–13.
- Tateisi, Y., Yakushiji, A., Ohta, T., Tsujii, J., 2005. Syntax annotation for the GENIAcorpus. In: Dale, R., Wong, K.F., Su, J., Kwong, O.Y. (Eds.), *Natural Language Processing – IJCNLP 2005*. Springer, pp. 222–227.
- TREC, 2016. Text retrieval conference. Available at: <http://trec.nist.gov>
- The Gene Ontology Consortium, 2000. Gene ontology: Tool for the unification of biology. *Nature Genetics* 25, 25–29.
- The UniProt Consortium, 2012. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Research* 40, D71–D75.
- Thompson, P., Ananiadou, S., Tsujii, J., 2017. The GENIA corpus: Annotation levels and applications. In: Ide N., Pustejovsky J. (Eds.), *Handbook of Linguistic Annotation*, Springer, pp. 1421–1432.
- Tsatsaronis, G., Balikas, G., Malakasiotis, P., *et al.*, 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics* 16, 138.
- Van Auken, K., Fey, P., Berardini, T.Z., *et al.*, 2012. Text mining in the biocuration workflow: Applications for literature curation at wormbase, dicty base and TAIR. Database. <http://dx.doi.org/10.1093/database/bas040>
- van Rijsbergen, C.J., 1977. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation* 33 (2), 106–119.
- van Rijsbergen, C.J., 1979. *Information Retrieval*. London: Butterworth.
- Vapnik, V., 1995. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag.
- Vlachos, A., Craven, M., 2010. Detecting speculative language using syntactic dependencies and logistic regression. In: *Proceedings of the Conference on Computational Natural Language Learning*, pp. 18–25.
- Voorhees, E., Hersh, W., 2012. Overview of the TREC 2012 medical records track. In: *Proceedings of the 21st Text Retrieval Conference – TREC'12*, NIST Special Publication.
- Wang, Q., Abdul, S.S., Almeida, L., *et al.*, 2016. Overview of the interactive task in BioCreative V. Database. <http://dx.doi.org/10.1093/database/baw119>
- Wilbur, W.J., Yang, Y., 1996. An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology text. *Computers in Biology and Medicine* 26 (3), 209–222.
- Witten, I.H., Moffat, A., Bell, T.C., 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*, second ed. San Francisco, CA: Morgan-Kaufmann.
- Xu, G., Niu, Z., Uetz P., *et al.*, 2009. Semi-supervised learning of text classification on bacterial protein–protein interaction documents. In: *Proceedings of the International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing (IJCBS'09)*, pp. 263–270.

- Xu, R., Yang, Y., Liu, H., Hsi, A., 2016. Cross-lingual text classification via model translation with limited dictionaries. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM'16), pp. 95–104.
- Xu, S., McCusker, J., Krauthammer, M., 2008. Yale Image Finder (YIF): A new search engine for retrieving biomedical images. *Bioinformatics* 24 (17), 1968–1970.
- Yang, Y., Liu, X., 1999. A re-examination of text categorization methods. In: Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 42–49.
- Yeh, A., Hirschman, L., Morgan, A., 2002. Background and overview for KDD Cup 2002 Task 1: Information extraction from biomedical articles. *SIGKDD Explorations* 4 (2), 87–89.
- Yin, Y., Wang, Z.Y., Mora-Garcia, S., *et al.*, 2002. BES1 accumulates in the nucleus in response to brassinosteroids to regulate gene expression and promote stem elongation. *Cell* 109 (2), 181–191.

Further Reading

- Cohen, K.B., Demner-Fushman, D., 2014. *Biomedical Natural Language Processing*. Amsterdam: John Benjamin Publishing Company.
- Hersh, W., 2009. *Information Retrieval: A Health and Biomedical Perspective*. New York, NY: Springer.
- Manning, C., Raghavan, P., 2008. *Introduction to Information Retrieval*. New York, NY: Cambridge University Press.
- Manning, C., Schütze, H., 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
- Mitchel, T., 1997. *Machine Learning*. McGraw-Hill.
- Przybyła, P., Shardlow, M., Aubin, S., *et al.*, 2016. Text mining resources for the life sciences. Database. <http://dx.doi.org/10.1093/database/baw145>
- Shatkay, H., Craven, M., 2012. *Mining the Biomedical Literature*. Cambridge, MA: MIT Press.

Relevant Websites

<http://bioasq.org/>
BioASQ.

<http://www.biocreative.org/>
Biocreative.

<http://bionlp.org/>
BioNLP.

<http://www.geneontology.org/>
Gene Ontology Consortium.

<https://www.ncbi.nlm.nih.gov/pubmed/>
NCBI.

Biographical Sketch



Hagit Shatkay is an Associate Professor and Director of the Computational Biomedicine and Machine Learning Lab at the Department of Computer and Information Sciences, University of Delaware. She has a PhD in Computer Science from Brown University, and an MSc and BSc in Computer Science from the Hebrew University of Jerusalem. Her research is focused on the development and use of machine learning and data-mining methods for addressing data-intensive problems in biology and medicine. She is an active member of the biomedical informatics and bio-text research communities, has presented many invited talks and international tutorials, and has authored, with Mark Craven, the book *Mining the Biomedical Literature* (MIT Press, 2012). Among many organizational and editorial roles, she is Section Editor for Knowledge Based Analysis at BMC Bioinformatics, associate editor in several journals, a board member of the International Society for Computational Biology (ISCB), has been an Area Chair of the Text Mining area at the International Conference on Intelligent Systems for Molecular Biology (ISMB) since 2008, co-organizer of the BioLINK Special Interest Group on linking biology and literature at ISMB, on the advisory board of BioASQ since its inception (2012), the steering committee of for TREC-Genomics 2005–2008, and others.

Biodiversity Databases and Tools

Lina AM Zalani and Kho S Jye, University of Malaya, Kuala Lumpur, Malaysia
Shaarmini Balakrishnan, Manipal International University, Nilai, Negeri Sembilan, Malaysia
Sarinder K Dhillon, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Background

Biological database is a large, organized library of life sciences information, collected from scientific experiments, published literature, high-throughput experiment technology, and computational analyses. It is associated with the usage of computerized software to store, update, query, and retrieve information within the system ([Kumar, 2005](#)). Database provides a convenient method to store enormous amount of data and allows biologists or researchers to collect the data, organize them systematically, analyze the data, and share the information with other researchers efficiently.

Biological data is often comprised of complicated data. There are different types of biological data, such as DNA sequence, protein sequence, protein structure, biodiversity, and many other specialized fields. Development of a database should acknowledge the complexity of the database content the capability of the technology in managing the complex data. In this article, we focus on biodiversity databases and tools, focusing on important projects done in this field. We also discuss the importance of biodiversity databases and the future direction anticipated inline with the advancement of technology and ICT.

Biodiversity Databases

Recent developments in information and communication technology have given birth to new experiences in the integration, analysis and visualization of biodiversity information, leading to the development of biodiversity databases ([Canhos *et al.*, 2004](#)). It includes application of information technology to the management, algorithmic exploration, analysis and interpretation of primary data regarding life, particularly at the species level of organization ([Soberon and Peterson, 2004](#)).

Biodiversity databases deals with managing information of unnamed taxa that are produced by environmental samples or sequencing of samples, and also cover the computational problems specific to the names of biological entities. Algorithms are developed to cope with those problems. For example, [Ch'ng \(2009\)](#) developed a segmentation algorithm for entity interaction, which increases the efficiency of real-time simulation that models the biotic interactions of large population datasets. Besides, biodiversity databases involve development of the syntax and semantics for publishing and integrating biodiversity information, while Darwin Core archive is the data standard developed to improve interoperability ([Wieczorek *et al.*, 2012](#)).

Existing Biodiversity Databases

Antweb

[AntWeb \(2018\)](#) focuses on ants, and it features as an image database. Being the leading online database of specimen records, images, and natural history information on ants, it aims to publish to the scientific community the high-quality images of global ant species. AntWeb focuses on data at specimen level and the corresponding images of the specimens. besides distribution maps. It accepts contributions on natural history information and field images that are linked directly to taxonomic names. Its content are also provided to the Global Biodiversity Information Facility ([gbif.org](#)), the Encyclopedia of Life ([EOL.org](#)) and Wikipedia (see "Relevant Websites section").

Fishbase

Fishbase (see "Relevant Websites section") ([FishBase.org, 2017](#)) is a global text and image database on finfishes. It contains a wide range of data on all species currently known worldwide. This includes information on the taxonomy, biology, trophic ecology, life history, uses, and historical data reaching back to 250 years. The detailed information in the database, which includes online analytical and graphical tools, cater to the needs of stakeholders, consists of researchers, scientists, policy makers, fisheries managers, donor, conservationists, teachers, and students. FishBase aims to provide for sustainable fisheries management, biodiversity conservation, and environmental protection.

Georgian Biodiversity Database

The Georgian Biodiversity Database (see "Relevant Websites section") ([Georgian Biodiversity Database, 2015](#)) focuses on the biological diversity of Georgia, and to a certain extent, the Caucasus ecoregion. This database can be categorized as a text and

image database, but mostly as an image database. This resource aims to introduce its biological diversity to the worldwide scientific community and also to any users alike. The Georgian Biodiversity database contains species list from terrestrial and freshwater ecosystems of Georgia, categorized according to the taxonomic hierarchical system. It also contains brief information on the individual species and higher-order taxa, including verbal descriptions of key ecological features and range, systematic remarks, bibliographies, images, and distribution maps.

Species + Database

Species + (see “Relevant Websites section”) ([Species +, 2017](#)) is a text database developed by UN Environment World Conservation Monitoring Centre (See Relevant Websites section) and the Convention on International Trade in Endangered Species of Wild Fauna and Flora (CITES) Secretariat ([cites.org](#)). It is designed as a centralized data warehouse for accessing key information on species such as Legal, Names, Distribution, References, Documents, and many others.

Freshwater Ecoregions of the World, FEOW

The main focus of this database is Freshwater biodiversity and it is known as a text and image database. FEOW (see “Relevant Websites section”) ([FEOW.org, 2015](#)) provides a new global biogeographic regionalization of this planet’s freshwater biodiversity. It covers virtually all freshwater habitats on Earth, and is the first-ever ecoregion map. Together with associated species data, it is a useful tool for underpinning global and regional conservation planning efforts, particularly to identify outstanding and imperiled freshwater systems. It also serves as a logical framework for a large-scale conservation strategies, and a global-scale knowledge base for increasing freshwater biogeographic literacy.

Bacterial Diversity Metadatabase, Bacdive

The Bacterial Diversity Metadatabase is a Metadatabase that provides strain-linked information about bacterial and archaeal biodiversity. It features a text database. BacDive (see “Relevant Websites section”) ([BacDive, 2017](#)) represents a collection of organism-linked information covering the multifarious aspects of bacterial and archaeal biodiversity. Its content covers on taxonomy, morphology, physiology, sampling and environmental conditions, and molecular biology too. BacDive allows simple search at the portal entrance for a quick access to strain related information. As for TAXplorer, it can be used to select a strain by browsing through taxonomy.

Scalenet

ScaleNet (see “Relevant Websites section”) ([ScaleNet, 2016](#)) is a text database on scale insects (superfamily Coccoidea). It contains information about scale insects, their taxonomic diversity, summary of nomenclatural history, geographic distribution, ecological associations, and economic importance and a complete bibliography. To date it contains data from 23,708 references pertaining to 8187 valid species names.

Naturdata

Naturdata (see “Relevant Websites section”) ([Naturdata.com, 2017](#)) is a text and image database on Portuguese taxa. It contains checklist of Portuguese species with a page for each species pertaining to the taxonomy, synonyms, vernacular names, images, videos and distribution of the species. Until 2009, it was not possible to know the count and the type of species existed in Portugal. Some groups, such as the chordates, insects, arachnids were well inventoried, but in all, these groups represented a minimal percentage of Portugal’s biodiversity (with current information, about 5%). Beyond 2009, Naturdata was created as a data warehouse to collect, process and share information on Portugal’s biodiversity.

Pan-European Species-Directories Infrastructure, PESI

The Pan-European Species-directories Infrastructure, PESI (see “Relevant Websites section”) ([PESI, 2017](#)) features European taxa and functions as a text and image database. PESI is an authoritative taxonomic checklist of European species, including higher taxonomy, synonyms, vernacular names, and European distribution. It is Europe’s e-infrastructure for taxonomic information on species occurring in Europe. The data warehouse contains nearly 450,000 scientific names, 240,000 valid (sub) species names, and 140,000 vernacular names in 89 languages. Besides taxonomic information, PESI gathers relevant information on species, such as images, literature, distributions, and conservation status. PESI also offers its services to those who are interested to build their proprietary species applications.

Wikispecies

WikiSpecies (see “Relevant Websites section”) ([WikiSpecies, 2017](#)) focuses on all forms of life. This database is both a text and image database, with a free species directory on Animalia, Plantae, Fungi, Bacteria, Archaea, Protista, and all other forms of life. Currently, there are about 549,468 articles in the database. Interestingly, it is able to do taxon navigation, and includes higher taxonomy, synonyms, vernacular names, and references.

All Catfish Species Inventory, ACSI

All Catfish Species Inventory (see “Relevant Websites section”) ([ACSI, 2017](#)) is a text and image database on catfish, categorized by the genera. This inventory also estimates the total species count, aids in the discovery, description and dissemination of knowledge of all catfish species by the world consortium of taxonomists and systematists. The aim of ACSI is to discover around 1750 new species of catfish and describe between 2300 and 4600 new species of freshwater fishes, which will then result to a complete taxonomy of Siluriformes and later in the completed taxonomy of Otophysi, the clade containing over two-thirds of all freshwater fishes. Besides a completed taxonomy of catfishes with up-to-date identification guides, ACSI will include atlases, catalogues and checklists of species, phylogenetic studies of higher-level relationships among catfishes, and data analytics. It is expected to produce large samples of freshwater fishes from poorly collected regions to be added to permanent collections in US and foreign institutions, and enhanced international communication among fish taxonomists. The ACSI is expected to serve as a channel of communication among global taxonomists on topics pertaining to research, educational and outreach opportunities.

Arctos

Arctos (see “Relevant Websites section”) ([Arctos, 2017](#)) is a text and image database that features specimen holdings of several natural history museums, agencies, and accessible private collections. It consists of vertebrates, invertebrates, parasites, vascular and non-vascular plants, many with images and extensive usage data. It is both a community and a collection management information system. It provides fundamental research infrastructure for biodiversity data, and is intended for curators, collection managers, investigators, educators, and anyone interested in natural and cultural history. Besides serving individuals, Arctos is a consortium of museums that collaborate to serve data on over 3 million records from natural and cultural history collections. It is a collaborative effort to share data vocabulary and standards while curating and improving the quality of shared data such as agents and taxonomy. Arctos on its own is an information system that integrates data from a myriad of disciplines which are anthropology, botany, entomology, ethnology, herpetology, ichthyology, mammalogy, ornithology, paleontology, and parasitology. The content of Arctos includes specimen records, observations, tissues, endoparasites and ectoparasites, stomach contents, field notes and other documents, and media such as images, audio recordings, and video. Arctos displays all that is known about a museum record, and provides solutions to managing and integrating collections data with object tracking (barcodes, RFID tags), transactions (loans, borrows, accessions, permits), geospatial information (collecting events, coordinates), agents (people, organizations), and usage (publications, projects, citations).

ACB Database

The ASEAN Centre for Biodiversity (see “Relevant Websites section”) ([ACB, 2015](#)), is a text and image database consisting of information on amphibians, birds, butterflies, dragonflies, edible plants, freshwater fishes, mammals, plants, reptiles and Malesian mosses of Southeast Asia. Established in 2005, besides serving as a data store, ACB facilitates cooperation and coordination among the ten ASEAN Member States (AMS) on the conservation and sustainable use of biological diversity, and the fair and equitable sharing of benefits arising from the use of such natural treasures. ACB implements its programs and projects through five components, which are Programme Development and Implementation, Capacity Building, Biodiversity Information Management, Communication and Public Affairs, and Organizational Management and Resource Mobilization.

Vertnet

While most biodiversity databases are built as data warehouses, VertNet (see “Relevant Websites section”) ([VertNet.org, 2016](#)) uses the distributed database concept. It contains text and image specimens on vertebrates. VertNet is built upon conventional VertNET networks such as FishNet, MaNIS, HerpNET, ORNIS by combining them into a single interface for the community to discover, capture, and publish biodiversity data online. It is also an engine for training current and future professionals to use and build upon best practices in data quality, curation, research, and data publishing in taxonomy.

Database of Plant Biodiversity of West Bengal, WBPBDIVDB

The Database of Plant Biodiversity of West Bengal (see “Relevant Websites section”) ([WBPBDIVDB, 2017](#)) is a plant biodiversity database of West Bengal, and it functions as a text and image database. The database covers the richness of floral diversity of West Bengal from Terai, Duars, Darjeeling, the eastern Himalayan region and in the mangrove forests of Sundarbans. West Bengal is an important

biodiversity spot in India as it contributes around 12% of the total angiosperm diversity found all over India while the southern deltaic parts of West Bengal owns more than 60 species of the Sundarban Mangrove ecosystem, which outnumbers the total mangrove diversity in India. There are also information on the IUCN status of Plants in West Bengal, showing Family, RDB status, distribution sites and average altitudes. The Genera included are algae, bryophyte, dicotyledon, fungi, gymnosperm, lichen, monocotyledon and pteridophyta. It also shows the species and ecological statistics, by displaying total species and frequency of occurrences.

Reptiles Database

The Reptiles Database (see “Relevant Websites section”) ([The Reptiles Database, 2017](#)) is a text and image database on reptiles, containing taxonomic information, names, and photos on about 10,000 species and 2800 subspecies. This database provides a catalogue of all living reptile species and their classification. It covers all living snakes, lizards, turtles, amphisbaenians, tuataras, and crocodiles, emphasizing on taxonomic data, particularly names and synonyms, distribution and type data, and literature references. It has no commercial interest and therefore depends on contributions from volunteers. Most of the data comes from published sources, which are curated by editors.

iNaturalist

iNaturalist (see “Relevant Websites section”) ([iNaturalist.org, 2017](#)) is an online social network of nature enthusiasts sharing biodiversity knowledge. In 2014, the California Academy of Sciences acquired iNaturalist, and now serves as the home for this endeavor. iNaturalist records a life based on number of observations by the community, which is then noted by scientists and land managers, hence serves as a crowdsourced species identification system and an organism occurrence recording tool. The community in iNaturalist records observations, get help with identifications, collaborate with others to collect similar information for a common purpose, or access the observational data collected by other iNaturalist users. It is a unique database as the scientifically valuable biodiversity data is generated by the user’s personal encounters and experiences. Scientists and taxonomic experts can provide enormous value to the community by helping to identify observations.

Integrated Biodiversity Information System, IBIS

The Integrated Biodiversity Information System (see “Relevant Websites section”) ([IBIS, 2011](#)) is a text and image database focusing on Australian plants. It provides taxonomic information, collection details and photographs, incorporating the integration, retrieval and dissemination of biodiversity information via collections management of nomenclatural and taxonomic indices, information delivery, web services and federated database support. IBIS links the data held in the various collections of the Australian National Botanic Gardens, the Australian National Herbarium, the Australian Plant Image Index and the Australian Plant Name Index. IBIS is built upon open source solutions such as Apache, Perl and Java using the Oracle RDBMS for the application and SQL engine. The content uses global naming conventions such as HISPID, TDWG, and BioCase for information management and delivery.

Natureserve Explorer

NatureServe (see “Relevant Websites section”) ([NatureServe Explorer, 2017](#)) is a database that houses about 70,000 plants, animals, and ecosystems of the United States and Canada, with an in-depth coverage for rare and endangered species. NatureServe easily locates information on scientific and common names, conservation status, distribution maps and images for thousands of species, life histories and conservation needs.

Virtual Tour Database System of Rimba Ilmu Botanic Garden, University of Malaya

This system is a virtual reality (VR) application on plants in Rimba Ilmu Botanical Garden, University of Malaya (UM) (see “Relevant Websites section”) ([Rimba Ilmu Botanic Garden, 2016](#)), Kuala Lumpur, Malaysia. In this application, the VR technology was used to develop an innovative VR database system which includes visual analytics features. The interactive virtual tour functions side by side the biodiversity database, as it involves an online interactive website to be developed so that users have a virtual tour of a botanic garden and are able to select distinct hotspots throughout the garden. A biodiversity database that consists of the plant species details and information is linked to this interactive portal. The information on the plant species of Rimba Ilmu Botanic Garden such as taxonomy and locality, are stored in a plant database. Geographical representation data are also available in the portal. Visualizing multiple hotspots in a botanical garden using the concept of virtual tour is a novel method of visualizing biodiversity. There are seven hotspots which were selected according to significant plants within the hotspot. Overall, it is a portal on Rimba Ilmu that links to a plant species database and the virtual tours around the botanic garden. The application is a useful tool for biodiversity researchers, non-specialists and also for educating and engaging the public. This system is based on the effort started by [Sarinder et al. \(2007\)](#) in digitizing the local biodiversity using diverse computational tools and technology including building ontologies in biodiversity.

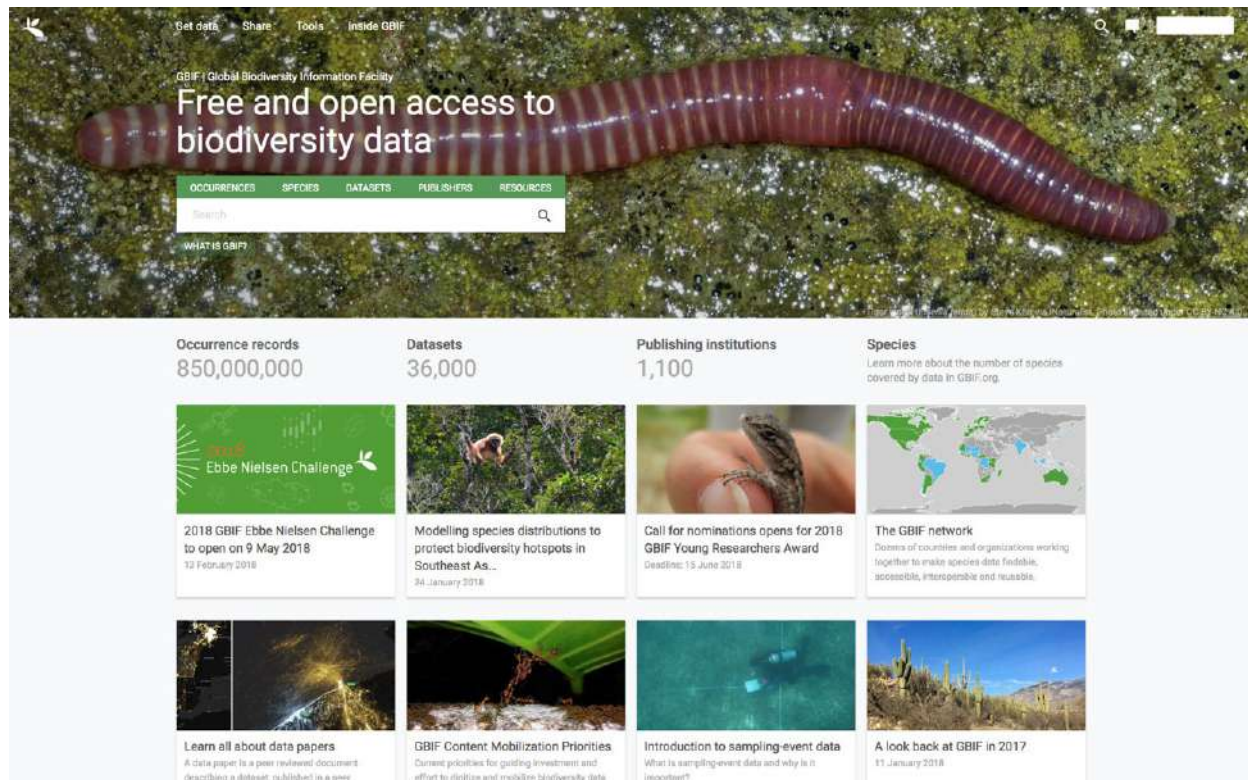


Fig. 1 The GBIF website. *Source:* GBIF.org, 2017. GBIF Home Page. Available from: <http://gbif.org> (27.02.18).

Get data Share Tools Inside GBIF					
Occurrences					
SEARCH OCCURRENCES 975,360,779 RESULTS					
TABLE GALLERY MAP TAXONOMY DATASETS DOWNLOAD					
Scientific Name	Country Or Area	Coordinates	Basis Of Record	Month & Year	
Lynx lynx (Linnaeus, 1758)	Norway	59.4N, 8.9E	human observation	2018 January	
Lynx lynx (Linnaeus, 1758)	Norway	58.9N, 7.8E	human observation	2018 January	
Phalangium opilio Linnaeus, 1758	New Zealand	40.3S, 175.6E	human observation	2018 January	
Anas zonorhynchos Swinhoe, 1866	Korea, Republic of	36.6N, 127.5E	human observation	2018 January	
Weinmannia racemosa L.f.	New Zealand	41.8S, 172.1E	human observation	2018 January	
Baeolophus bicolor (Linnaeus, 1766)	United States	38.9N, 77.1W	human observation	2018 January	
Nettion rufina (Pallas, 1773)	United Kingdom	54.9N, 1.5W	human observation	2018 January	
Canis latrans Say, 1823	United States	43.0N, 89.4W	human observation	2018 January	
Paraphidippus fertilis (Peckham & Peckham)	Mexico	23.3N, 106.4W	human observation	2018 January	
Barea exarcha Meyrick, 1884	New Zealand	41.3S, 174.7E	human observation	2018 January	
Stenellipsis bimaculata (White, 1845)	New Zealand	39.4S, 174.4E	human observation	2018 January	
Pluvialis aquatarola Linnaeus, 1758	United States	27.9N, 97.4W	human observation	2018 January	
Picea mariana Britton, Sterns & Poggenb.	United States	47.7N, 91.5W	human observation	2018 January	
Apis mellifera Linnaeus, 1758	United States	34.0N, 118.1W	human observation	2018 January	

Fig. 2 Occurrences information in GBIF. *Source:* GBIF.org, 2017. GBIF. Available from: <http://gbif.org> (27.02.18).

The figure displays two side-by-side screenshots of the GBIF Occurrences search interface. Both screenshots show a green header bar with a leaf icon and navigation links: 'Get data', 'Share', 'Tools', and 'Inside'. Below the header, a green bar contains the word 'Occurrences' and a trash icon. A search bar with the placeholder 'Search all fields' and a magnifying glass icon is present. Below the search bar, there are two tabs: 'Simple' and 'Advanced'. The left screenshot shows the 'Simple' search mode, and the right screenshot shows the 'Advanced' search mode. Both modes display a list of 20 searchable fields, each with a dropdown arrow. The fields are: Record License, Scientific Name, Basis Of Record, Location, Year, Month, Dataset, Country Or Area, Issue, Media Type, Publisher, Institution Code, Collection Code, Catalogue Number, Type Status, Recorded By, Catalogue Number, Type Status, Recorded By, Record Number, Occurrence ID, Organism ID, Publishing Country Or Area, Elevation, Depth, Locality, Water Body, State Province, Repatriated Data, Protocol, Last Interpreted, Event Date, and Establishment Means.

Fig. 3 Searchable fields in GBIF. *Source:* GBIF.org, 2017. GBIF. Available from: <http://gbif.org> (27.02.18).

The next section presents two very important and large biodiversity databases and tools widely used by scientists; the Global Biodiversity Information Facility (GBIF) and the Encyclopedia of Life (EoL) database. These databases are discussed as case studies in the next section.

Case Studies on Biodiversity Databases

This case study describes two databases, GBIF database and EOL database, which are some of the best examples of existing biodiversity databases and tools that are available on the web. It gives a description of the research and biodiversity needs motivated by a set of technologies with front end portals providing internal and external users with a comprehensive list of capabilities.

GBIF: Free and Open Access to Biodiversity Data

The Global Biodiversity Information Facility, GBIF (see "Relevant Websites section") (GBIF.org, 2017) is an open-data research infrastructure which is funded by governments around the globe, facilitating access more to over 377 million records from more than 400 data publishers (Ariño, 2010). GBIF arose from a 1999 recommendation by the Biodiversity Informatics Subgroup of the Organization for Economic Cooperation and Development's Megascience Forum. One of its report concluded that "An international mechanism is needed to make biodiversity data and information accessible worldwide", stressing the fact that this mechanism could produce many economic and social benefits and enable sustainable development by providing sound scientific

evidence. Following this mechanism, in 2001, GBIF was officially established through Memorandum of Understanding between participating governments.

The aim of GBIF is to provide easy access to text and image data on all types of life on Earth. The coordinator with its network of participating countries and organizations provide data owners around the globe with common standards and open-source tools that allows sharing of information such as the location and date of the recorded species. GBIF data is derived from multiple sources, including from museum specimens collected in the past and geotagged smartphone photos shared by amateur naturalists in recent dates (Fig. 1).

The GBIF network draws all the sources together using the Darwin Core standard, which forms the basis of millions of species occurrence records. An assessment of data count resulted in 267 million occurrences records (Fig. 2) accessible through the GBIF network in which 62% belonged to Kingdom Animalia, followed by Kingdom Plantae (23%), Fungi (1.55%), Protozoa (0.67%), and Bacteria (0.59%) (Gaiji *et al.*, 2003). Every year, hundreds of peer-reviewed publications and policy papers are written based on datasets published by GBIF, covering topics from the impacts of climate change and the spread of pests, to priorities for conservation and protected areas, food security and human health (see “Relevant Websites section”).

The role of GBIF is to provide a discovery window on the posted statistics. Such a position requires reconciling, decoding and publishing the crucial key attributes: taxonomic, temporal and geospatial (Gaiji *et al.*, 2003). The searchable fields in GBIF is presented in Fig. 3 while an example of species page is presented in Fig. 4. Fig. 5 shows example of georeferenced records.

GBIF's ultimate plan is to serve the needs of GBIF's global community and infrastructure to the next decade by empowering global network, enhancing biodiversity information infrastructure, filling data gaps, improving data quality, and also by delivering relevant data.

EOL: Free, Online Collaborative Encyclopedia

The Encyclopedia of Life (EOL) (see “Relevant Websites section”) (EOL.org, 2017) is a free, online collaborative encyclopedia intended to document all of the 1.9 million living species known to science (Fig. 1). It gives global access to knowledge about life on Earth (Fig. 6).

Under the umbrella of EOL, data previously scattered in books, journals, databases, websites, specimen collections, reports are now collated and made available to the public, in the form of articles, media, maps, data in a single website with a collection of application programming interfaces (APIs) (see “Relevant Websites section”). An example of a species page is shown in Fig. 7

The screenshot displays the GBIF species page for *Andrographis elongata* T. Anderson. The page is structured with a green header containing navigation links (Get data, Share, Tools, Inside GBIF) and a search bar. The main content area is divided into several sections:

- Classification:** A sidebar on the left lists the taxonomic hierarchy: Kingdom: Plantae, Phylum: Tracheophyta, Class: Magnoliopsida, Order: Lamiales, Family: Acanthaceae, and Genus: *Andrographis* Wall. ex Nees.
- Species:** The species name *Andrographis elongata* T. Anderson is displayed, followed by a list of synonyms: *Cryptophragmium cordifolium* Nees, *Cryptophragmium elongatum* Nees, *Justicia clandestina* Russell, *Justicia clandestina* Russell ex Wall., *Justicia cordifolia* B. Heyne, *Justicia cordifolia* B. Heyne ex Wall., and *Justicia elongata* Vahl.
- Species Page Header:** The species name *Andrographis elongata* T. Anderson is prominently displayed, along with its publication information: "Published in: Anderson T (1867) An Enumeration of the Indian Species of Acanthaceae. (continued.). Journal of the Linnean Society of London, Botany 9(40): 455-526. doi: 10.1111/j.1095-8339.1867.tb01309.x. Source: Catalogue of Life".
- Overview and Occurrences:** The page includes tabs for "OVERVIEW" and "REFERENCE TAXON". A green button indicates "94 OCCURRENCES". Below this, a section titled "15 OCCURRENCE RECORDS WITH IMAGES" shows a grid of specimen images.
- Georeferenced Records:** A map section titled "6 GEOREFERENCED RECORDS" shows a world map with yellow dots indicating the locations of the records, primarily in South America and Africa.

Fig. 4 Example of species page in GBIF. Source: GBIF.org, 2017. GBIF. Available from: <http://gbif.org> (7.02.18).

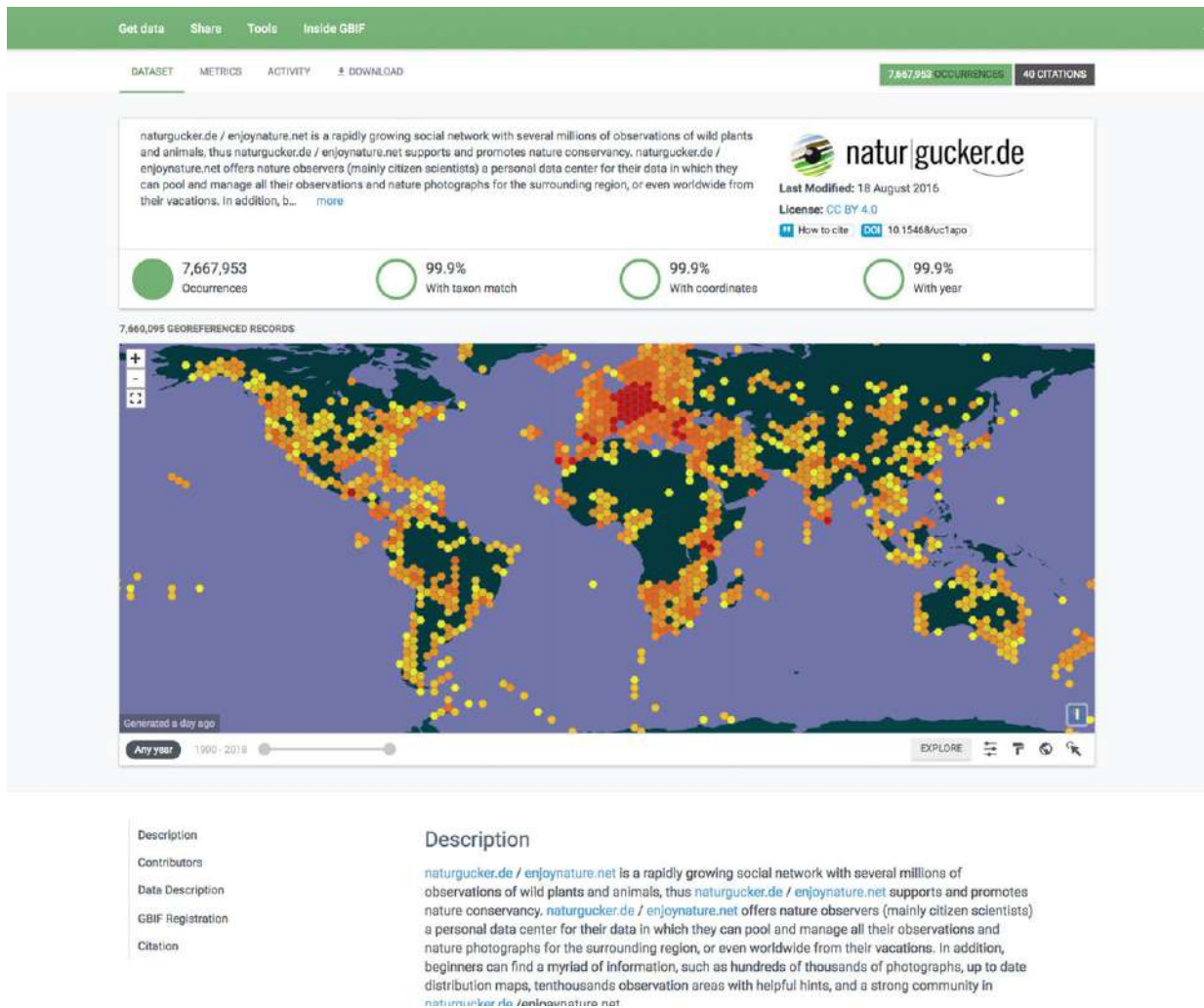


Fig. 5 GBIF's georeferenced records. *Source:* GBIF.org, 2017. GBIF. Available from: <http://gbif.org> (27.02.18).

while **Fig. 8** shows an example of textual and image content in the EOL website (see "Relevant Websites section"). The type of taxonomy content is shown in **Fig. 9**.

EOL is supported by some very important institutions, which are Atlas of Living Australia, La Comisión Nacional para el Conocimiento y Uso de la Biodiversidad (CONABIO), Harvard University, Marine Biological Laboratory, New Library of Alexandria, and also Smithsonian Institution's National Museum of Natural History. In addition, EOL collaborates with global bioinformatics projects, like the Biodiversity Heritage Library (BHL), Barcode of Life (BOLD), Catalogue of Life (COL), and Global Biodiversity Information Facility (GBIF) (see "Relevant Websites section").

Essentially, the Encyclopedia of Life serves the global community and by collaborating with its Global Partners, EOL seeks to remove the barriers of geography and language to support international professional and citizen scientists interested in biodiversity and related topics (see "Relevant Websites section"). More than just a database, EOL offers tools and resources that help users harness and apply the information found on EOL.

EOL's mission is to increase awareness and understanding of living nature through a service engine that gathers, generates, and shares knowledge in an open, freely accessible and trusted digital resource (see "Relevant Websites section"). EOL has achieved its mission as it is currently one of the biggest and widely accessible data store on biodiversity. Over time, the content in EOL will be richer as more partners joins this initiative and the current partners stay active in depositing data into the portal.

Importance of Digitizing Biodiversity Information

The collection of biodiversity information plays a critical role in understanding biodiversity and ecosystem. The significance of biodiversity databases are highlighted in this section.



Fig. 6 EOL website. Source: EOL.org, 2017. Encyclopedia of Life Home Page. Available from: <http://www.eol.org> (27.02.18).

Biodiversity and Environmental Change

Biodiversity databases and tools allows documentation of the biological diversity of life and demonstration of changes in the environment that have taken place through time (Baird, 2010). The presence or absence of a species as well as their frequency changes in a geographic region over the course of time are documented by examining the records. If a gradual decline of an important species is observed, it may serve as an indicator whether a recovery program should be carried out to restore the population of the species. The collections that are well documented and deposited can offer proof that the program was successful, provide evidence that changes has taken place, and can help to effectively monitor rare, threatened and endangered species.

Invasive Alien Species and Biosecurity

Other than monitoring endangering species, biodiversity informatics is capable of revealing the invasive alien species that are threat for biodiversity. Increasing travel, trade, and tourism associated with globalization and expansion of the human population have facilitated intentional and unintentional movement of species beyond natural biogeographical barriers (Forest Research Institute Malaysia, 2011). These species are known as alien species and many of them have become invasive. Invasive alien species are known to cause biodiversity loss and lead to a nation's economic loss. Thus, it is critical to differentiate native from alien species, and respond rapidly to these threat, either to remove the species from a region, or suppress their population growth if unable to remove it.

Besides revealing invasive alien species, biodiversity informatics allows prediction of potential of a non-native species to be invasive alien species. Faulkner *et al.* (2014) has created and applied a simple, rapid methodology for developing invasive species watch lists. The watch lists are created using three predictors of three invasion success: History of invasion, environmental suitability and propagule pressure. The authors claim that building the list may be an important step in developing biosecurity scheme, especially for resource poor regions.

Public Health and Wildlife Disease

There are many infections that manifest themselves in human and non-human populations, such as H1N1 Influenza, West Nile Virus, Lyme disease, tuberculosis, Ebola Virus, and SARS. For these and other zoonoses, approximately 70% of new important disease affecting human health are believed to have a wild animal source, and can have profound impacts on human health and economy (Blancou *et al.*, 2005).

Kissophagus hederæ [add to a collection](#)

[learn more about names for this taxon](#)

You've arrived here by searching for *Kissophagus hederæ*. [Click here to see other search results.](#)

Overview Detail 1 Media 2 Maps Names Community Resources Literature Updates




Kissophagus vicinus **TRUSTED** • *Kissophagus hederæ* **UNREVIEWED** ... [more](#)

 © Udo Schmidt

Source: Flickr: EOL Images

[see all media](#)

EOL has data for 7 traits [see all](#)

water depth	(min)	-5 m
	(max)	19 m
longitude	(min)	-0.0177 degrees
	(max)	0.24 degrees
latitude	(min)	51.39 degrees
	(max)	51.52 degrees
extinction status		extant
habitat includes		non-marine
interacts with		Kissophagus hederæ
interacts with		Curculionidae
		Curculionidae
		Kissophagus

Classification

Classification from [Species 2000 & ITIS Catalogue of Life: April 2013](#) selected by G. Michael Hogan - [see more](#).

- Animalia +
 - Arthropoda +
 - Insecta +
 - Coleoptera +
 - Curculionidae +
 - Curculionidae +
 - Kissophagus +
 - Kissophagus hederæ* Chapuis, 1869
 - Kissophagus albivittatus* Lepesme, 1942a
 - Kissophagus binodius* Reitter, 1913a
 - Kissophagus confusus* Eggers, 1935c
 - Kissophagus curtus* Eggers, 1925
 - Kissophagus erinaceus* Wichmann, H.E., 1916
 - Kissophagus fasciatus* Hagedorn, 1909n
 - Kissophagus fuscus* Schedl, 1954d
 - Kissophagus gallicus* Schedl (Eggers in), 1963h
 - Kissophagus granulatus* Lepesme, 1942a
 - 7 more... [show full tree...](#)

Brief summary

No one has contributed a brief summary to this page yet.

Explore what EOL knows about *Kissophagus hederæ*.

[add a brief summary to this page](#)

Present in 14 collections [see all](#)

 **Taxon Clean-Up To Do List**
177 other items

Belongs to 2 communities [see all](#)

 **Taxon Page Cleaning Crew**
277 other items, 23 members

Fig. 7 EOL's species page. Source: EOL.org, 2017. Encyclopedia of Life website. Available from: <http://www.eol.org> (27.02.18).

Using the occurrence data from GBIF, [Pigott et al. \(2014\)](#) mapped the area potentially at risk from outbreaks of Ebola virus, based on the environmental niche of bat species believed to act as reservoir hosts of the disease. The authors determined the national population at risk, and claimed that this would be a strong rationale for improving, prioritizing, and stratifying surveillance for EVD outbreaks and diagnostic capacity in these countries.

Economics and Regulatory Framework

The Economics of Ecosystems and Biodiversity (TEEB) (see "Relevant Websites section") is a global initiative focused on "making nature's values visible". The objective of TEEB is to introduce the importance of putting economic value to biodiversity, particularly for policies and decision making at all levels. It follows a structured approach to valuation of ecosystem in economic terms in order to recognize the benefits of biodiversity protection, which is inline with the objectives of the Convention of Biological Diversity. While economic impact studies on biodiversity is still new, sooner or later it is going to change the way humans perceive biodiversity.

Access and Benefit Sharing

The obvious significance of digitization of biodiversity information is dissemination of information globally online. Before digitization, the information is held by institutes in developed countries, whereas scientists may do their field work at

Felidae
Cats [learn more about names for this taxon](#)

[add to a collection](#)

You've arrived here by searching for *cat*. [Click here to see other search results.](#)

Overview Detail 7455 Media 3 Maps Names Community Resources Literature Updates







Lynx rufus **TRUSTED**
(cc) BY-SA Don DeBoid from San Jose, CA, USA
Source: [Wikimedia Commons](#)

[see all media](#)
[see all maps](#)

EdIV a de donades sus 20 caràcters [see all](#)

first appearance (older bound)	33.3 million years ago
habitat	carcass chaparral city mai
feeds on	Atopochen aegyptiaca Amazilia tzacati Ammodorcas clarkei mai
has predator	Felidae Felidae
interacts with	Acinonyx jubatus Atopochen aegyptiaca Amazilia tzacati mai
preys upon	Alectura lathamii Boiga irregularis Burhinus grallarius mai
interacts with	Felidae
number of articles on EOL	2,216
number of images on EOL	5,848

Comprehensive Description [read full entry](#)

[learn more about this article](#)

With the exception of Antarctica, Australia, New Zealand, Madagascar, Japan, and most oceanic islands, native populations of cats are found worldwide, and one species, domestic cats, have been introduced nearly everywhere humans currently exist. Although some authorities recognize only a few genera, most accounts of Felidae recognize 18 genera and 36 species. With the exception of the largest cats, most are adept climbers, and many are skilled swimmers. Most felids are solitary. Often, felids are separated into two distinct subgroups, large cats and small cats. Generally, small cats are those that, due to a hardening of the hyoid bone, have an inability to roar. Felidae consists of 2 subfamilies, Pantherinae (e.g., lions and tigers) and Felinae (e.g., bobcats, pumas, and cheetahs).

Felids are perhaps the most morphologically specialized hunters of all carnivores, often taking prey as

Classification

Classification from [IUCN Red List](#) selected by [Jennifer Hammock](#) - [see more](#)

- [Animalia](#) ±
- [Chordata](#) ±
- [Mammalia](#) ±
- [Carnivora](#) ±
- [Felidae](#)
- [Acinonyx](#) ±
- [Caracal](#) ±
- [Felis](#) ±
- [Lagocardius](#) ±

Fig. 8 Textual and image content in EOL. Source: EOL.org, 2017. Encyclopedia of Life website. Available from: <http://www.eol.org> (27.02.18).

underdeveloped country. Digitization allows researcher to access to the information remotely via the web. Besides, digitization allows sharing of information among researchers. The information may be integrated and analyzed to contribute to their study.

Future Directions and Closing Remarks

The production of biodiversity data has accelerated enormously in the past few years, and this has led to the creation of databases and tools to seek deeper understanding of the myriad of values it offers. While every biodiversity and conservation entity in the world is trying to collate and store their data in a digital form, without sharing it in a central repository, the actual value cannot be perceived. The engagement programs started by world class organizations like GBIF and EOL have introduced novel methodologies in managing, accessing and visualizing primary and secondary biodiversity datasets. It is hoped that, with the integration of biodiversity data globally, new discoveries in science is produced, besides creating a new economy in this field.

Apart from curated databases, we strongly belief that species identification, which is currently dependent on expertise of a few individuals, can be automated using technologies in pattern recognition and semantics, for both image and text retrieval. Future efforts in biodiversity should focus on areas such as (1) to what extent can biodiversity be linked using the current technology; (2) how does biodiversity fits into Big Data Analytics; and (3) to what extent automated systems can help in returning accurate species identifications with annotations? With these findings, we should aim to build integrated biodiversity platforms, focusing on semantic search and big data analytics, along with high throughput automated systems capable of accurate species identifications in seconds. This potential system can then be used as a benchmark for indexing millions of identified species.

The actual value of a species can only be known if sufficient data about the species is linked, in a manner when new insights can be discovered easily, through the technology of linked data. This is similar to the social media data linking people such as Facebook, Twitter, LinkedIn, ResearchGate, ResearcherID, WeChat and Hangouts. A similar idea could be implemented for specimens or species.

Table of Contents
OVERVIEW
Brief Summary
Comprehensive Description
Distribution
PHYSICAL DESCRIPTION
Morphology
ECOLOGY
Habitat
Trophic Strategy
Associations
LIFE HISTORY AND BEHAVIOR
Behavior
Life Expectancy
Reproduction
EVOLUTION AND SYSTEMATICS
Functional Adaptations
MOLECULAR BIOLOGY AND GENETICS
Molecular Biology
CONSERVATION
Conservation Status
RELEVANCE TO HUMANS AND ECOSYSTEMS
Benefits
WIKIPEDIA
RESOURCES
Partner links
Nucleotide sequences
LITERATURE
Literature references
Biodiversity Heritage Library

Fig. 9 Taxonomy data content in EOL. *Source:* EOL.org, 2017. Encyclopedia of Life website. Available from: <http://www.eol.org> (27.02.18).

Technology such as semantic web and ontologies, big data analytics, cloud computing and data science are controlling the ICT age and many new discoveries are anticipated. We strongly believe application of these technologies in biodiversity studies will help scientists in future investigation, inquiry, reasoning and analysis of biodiversity data. Moreover, students, policymakers and stakeholders will gain insight to biodiversity data captured and interpreted precisely.

See also: Bioinformatics Data Models, Representation and Storage. Biological Database Searching. Challenges in Creating Online Biodiversity Repositories with Taxonomic Classification. Data Storage and Representation. Ecosystem Monitoring Through Predictive Modeling. Information Retrieval in Life Sciences. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Population Genetics

References

- ACB, 2015. ASEAN Centre for Biodiversity (ACB) Home Page. Available from: <https://aseanbiodiversity.org> (06.09.17).
- ACSI, 2017. All Catfish Species Inventory (ACSI) Home Page. Available from: <http://silurus.acnatsci.org> (06.09.17).
- AntWeb, 2018. Available from <http://www.antweb.org>. (25.04.18).
- Arctos, 2017. Arctos Home Page. Available from: <https://arctosdb.org> (06.09.17).
- Ariño, A.H., 2010. Approaches to estimating the universe of natural history collections data. *Biodiversity Informatics* 7, 81. Available from: <https://journals.ku.edu/jbi/article/view/3991>.
- BacDive, 2017. The Bacterial Diversity Metadatabase Home Page. Available from: <https://bacdive.dsmz.de> (23.08.17).
- Baird, R., 2010. Leveraging the fullest potential of scientific collections through digitization. *Biodiversity Informatics* 7, 130–136.
- Blancou, J., Chomel, B.B., Belotto, A., Meslin, F.X., 2005. Emerging or re-emerging bacterial zoonoses: Factors of emergence, surveillance and control. *Veterinary Research* 36, 507–522. doi:10.1051/vetres:2005008.
- Canhos, V.P., Souza, S., Giovannia, R., Canhos, D.A.L., 2004. Global biodiversity informatics: Setting the scene for “new world” of ecological modeling. *Biodiversity Informatics* 1, 1–13.
- Ch’ng, E., 2009. An efficient segmentation algorithm for entity interaction. *Biodiversity Informatics* 6, 5–17.
- EOL.org, 2017. Encyclopedia of Life Home Page. Available from: <http://www.eol.org> (31.12.17).
- Faulkner, K.T., Robertson, M.P., Rouget, M., Wilson, J.R.U., 2014. A simple, rapid methodology for developing invasive species watch lists. *Biological Conservation* 179, 25–32. doi:10.1016/j.biocon.2014.08.014.
- FEOW.org, 2015. Freshwater Ecoregions of the World (FEOW) Home Page. Available from: <http://www.feow.org> (14.08.17).

- FishBase.org, 2017. FishBase Search Page. Available from: <http://www.fishbase.org> (01.01.18).
- Gaiji, S., Chavan, V., Arino, A.H., 2013. Content assessment of the primary biodiversity data published through GBIF network: Status, challenges and potentials. *Biodiversity Informatics* 8, 94–172. Available from: <https://www.jcel-pub.org/jbi/article/view/4124/4201>.
- GBIF.org, 2017. GBIF Home Page. Available from: <http://gbif.org> (31.12.17).
- Georgian Biodiversity Database, 2015. Georgian Biodiversity Database Home Page. Available from: <http://www.biodiversity-georgia.net> (14.08.17).
- IBIS, 2011. IBIS (Integrated Biodiversity Information System) Introduction Page. Available from: <http://www.anbg.gov.au/ibis> (07.09.17).
- iNaturalist.org, 2017. iNaturalist Home Page. Available from: <http://www.inaturalist.org> (07.09.17).
- Kumar, S., 2005. Molecular biology database/biological database/bioinformatics database – An overview. Available from: <http://bioinformaticsweb.net/>
- Naturdata.com, 2017. Naturdata Home Page. Available from: <http://naturdata.com> (23.08.17).
- NatureServe Explorer, 2017. NatureServe Explorer Home Page. Available from: <http://explorer.natureserve.org> (07.09.17).
- PESI, 2017. Pan-European Species-directories Infrastructure (PESI) Home Page. Available from: <http://www.eu-nomen.eu/portal> (23.08.17).
- Pigott, D.M., Golding, N., Mylne, A., *et al.*, 2014. Mapping the zoonotic niche of Ebola virus disease in Africa. *eLife* 3, e04395. doi:10.7554/eLife.04395.
- Rimba Ilmu Botanic Garden, 2016. Rimba Ilmu Botanic Garden Home Page. Available from: <http://rimba.um.edu.my> (28.02.18).
- Sarinder, K.K.S., Lim, L.H.S., Dimiyati, K., Merican, A.F., 2007. An indigenous integration system for biodiversity databases. *Online Journal of Bioinformatics* 8 (1), 56–60.
- ScaleNet, 2016. ScaleNet Home Page. Available from: <http://scalenet.info> (23.08.17).
- Soberon, J., Peterson, A.T., 2004. Biodiversity informatics: Managing and applying primary biodiversity data. *Philosophical Transactions of the Royal Society B: Biological Sciences* 359, 689–698. doi:10.1098/rstb.2003.143.
- Species +, 2017. Species + Home Page. Available from: <https://speciesplus.net> (14.08.17).
- The Reptiles Database, 2017. The Reptiles Database Home Page. Available from: <http://www.reptile-database.org> (07.09.17).
- VertNet.org, 2016. VertNet Home Page. Available from: <http://www.vertnet.org/index.html> (07.09.17).
- WBPBDIVB, 2017. Database of Plant Biodiversity of West Bengal Introduction Page. Available from: http://thebiome.in/databases/wbpbdivb_intro.php (07.09.17).
- Wieczorek, J., Bloom, D., Guralnick, R., Blum, S., *et al.*, 2012. Darwin core: An evolving community-developed biodiversity data standard. *PLOS ONE* 7 (1), e29715. doi:10.1371/journal.pone.0029715.
- WikiSpecies, 2017. WikiSpecies Main Page. Available from: https://species.wikimedia.org/wiki/Main_Page (23.08.17).

Relevant Websites

- <http://silurus.acnatsci.org/>
All Catfish Species Inventory.
- <http://www.antweb.org>
Antweb.org - AntWeb.
- <https://arctosdb.org/>
Arctos.
- <https://aseanbiodiversity.org/>
ASEAN Centre for Biodiversity.
- <https://bacdive.dsmz.de/>
BacDive.
- <http://eol.org/>
Encyclopedia of Life.
- <http://www.eu-nomen.eu/portal/>
EU-nomen.
- <http://www.feow.org/>
FEOW.
- <http://www.fishbase.org>
FishBase.
- <https://www.gbif.org/>
GBIF.
- <http://www.biodiversity-georgia.net>
Georgian Biodiversity Database.
- <http://www.anbg.gov.au/ibis/>
IBIS.
- <http://www.inaturalist.org/>
iNaturalist.
- <http://naturdata.com/>
NATURDATA.
- <http://explorer.natureserve.org/>
NatureServe Explorer.
- <http://rimba.um.edu.my>
Rimba Ilmu.
- <http://scalenet.info/>
Scale Net.
- <https://speciesplus.net/>
Species +.
- <http://www.teebweb.org/>
TEEB.
- http://thebiome.in/databases/wbpbdivb_intro.php
The Biome.
- <http://www.reptile-database.org/>
THE REPTILE DATABASE.
- <https://www.unep-wcmc.org/>
Unep-WCMC.

<http://www.vertnet.org/index.html>

VertNet.

<https://www.wikipedia.org/>

Wikipedia.

https://species.wikimedia.org/wiki/Main_Page

Wikispecies.

Challenges in Creating Online Biodiversity Repositories With Taxonomic Classification

Mohd NM Ali, Amy Y Then Hui, and Sarinder K Dhillon, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Presently many biodiversity databases with species information annotations are available online. Although these databases tend to focus on specific taxa, accurate and updated taxonomic classification remain challenging given that new species are being discovered from time to time while extant species may become or have gone extinct. According to the United Nations Environment Programme, scientists estimate that 150–200 species of plants, insects, birds, and mammals become extinct every 24 h and there are around 15% of mammal species and 11% of bird species that are classified as threatened with extinction (Vidal, 2010). On the other hand, new species are being described every three days over last decade (Tobias, 2010). This situation depicts the challenges taxonomists and biologists have to face in order to keep up with updated, accurate information on biodiversity profiles and classification. Most of the online taxonomic databases or portals contain information on authoritative source(s), namely who is managing the data, how it is curated and credibility of the data. Most of these authoritative sources are fellow scientists or researchers, who form a group that manages the taxonomic information shared to the public.

Taxonomy is a field of classifying living things through defining and naming based on shared characteristics. Over time, there are varying definitions of taxonomy within the scientific community, but the core of its practice is revolved around the conception, naming, and classification of groups of organisms. The Linnaean Taxonomy, a system for categorizing organisms and naming living things through a binomial nomenclature, is regarded as the origin of the practice. This system is named after Carl Linnaeus, a Swedish botanist and the father of taxonomy, who introduced the science of naming organisms (either living or extinct) through the usage of two descriptors, namely the genus that the organism belong to, followed by the species modifier. For example, a Chinook salmon or better known as king salmon, has the scientific name of *Oncorhynchus tshawytscha* where *Oncorhynchus* is the genus and *tshawytscha* is the species modifier.

In modern biological classification, taxonomic hierarchy is used to place organisms in increasingly exclusive groupings or ranks, starting with the broadest rank of domain, followed by kingdom, phylum, class, order, family, genus, and species. Within these categorizations, there are more specific groups for them known as subdivisions, which further clarify their classification ranks. For example, superclass, subclass, infraclass, or parvclass form taxonomic subdivisions of the class rank.

Taxonomic classification is one of the main branches of biology, practiced by experts known as taxonomists. The roles of trained taxonomists are clear, which include describing new species and conducting taxonomic revisions to described species as needed. Such exercises are not easy as prior knowledge of the taxa of interest is required and formal taxonomic training on the subject of study is crucial. Incorrect identification of biological entities will introduce misleading information and errors in subsequent work pertaining to that particular information, e.g., worldwide distribution of a species. As such, competent taxonomic work is fundamentally important for the understanding of biology and conservation of biodiversity.

Taxonomy is separated into several sub-branches, each with their unique approach of classifying organisms. Some of the popular sub-branches are alpha taxonomy, beta taxonomy, cytotaxonomy, chemotaxonomy, and cladistics taxonomy. Alpha taxonomy and beta taxonomy both deal with external morphology; however the latter includes internal features such as anatomical characteristics and organ studies. Cytotaxonomy deals with plant classification through the use of cytology (cell) studies, which greatly aid plant classification. Chemotaxonomy also focuses on plants, but utilizes information on chemical constituents of the plants for classification. A widely used taxonomic approach in the modern age is the cladistic taxonomy which groups organisms into clades based on shared, derived characteristics from the most recent common ancestor.

Taxonomy Related Issues

It is evident that even in the age of technology, where information can be accessed and shared anywhere in the world via the internet, dedicated online databases have proven to be useful for the scientific community to publish their findings. The use of an online database is a necessity for taxonomic research; however there are a number of issues related to biodiversity conservation and data management which will be discussed. The first issue, which is quite alarming, is insufficiency of taxonomists or taxonomic research capacity. This is essentially a global issue affecting biology in general. A study showed that although systematic research as a whole has been on the increase, traditional taxonomists, a unique breed of scientists with specific skill sets that are difficult to train and replace, are dwindling in numbers (Lee and Palci, 2015). Another study on the other hand showed that the number of taxonomists has actually been increasing; however the number of species described per taxonomist has not seen decline for poorly studied taxa such as beetles and parasitic wasps (Bacher, 2012). This suggests that much remains to be done for new species discovery and description. One recent study pointed out that while taxonomy is still very much alive, taxonomists are struggling to obtain sufficient funding and appropriate recognition for their field (Bik, 2017).

The second issue, which is linked to the previous, is about species unknown to science going extinct before they are discovered and described. It is commonly accepted in the scientific community that at the current rate of extinction, many species will die before they are managed to be described by science (Dirzo and Raven, 2003; Gaston and May, 1992). In a research done in the Cape Floristic Region (one of the smallest and popular biodiversity hotspots in South Africa), it was reported that there is a finite number of species that may be processed and identified by a taxonomist during his/her working life (Treurnicht *et al.*, 2017). The same research further claimed that the current state of technology and infrastructure does not support an increase in individual taxonomic output; improper operational concepts practiced by taxonomists may also affect a taxonomist's output.

The third issue is regarding the correct identification of species by biologists in general and by taxonomists, which is fundamental for any taxonomy-based studies. The field of taxonomy is thwarted by inconsistent taxonomy literature and vague species descriptions containing incorrect identifications due to work by amateur taxonomists, or inadequate access to original works on the species. Also as mentioned earlier, there are various sub-branches of taxonomy. The use of integrated taxonomic methods, i.e., other branches in conjunction with traditional morphological methods, has led to taxonomic revisions for many known species; this indicates that accurate classification of species remains an issue. However, in modern taxonomy practice, traditional morphology tends to be overlooked in favor of genetic studies. Decoupling of traditional and modern taxonomy affects the type of data shared on online databases, which usually have groups or consortium of sorts regulating these information. Given the present age of Big Data, it is impossible to review and regulate all taxonomic information needed for correct species identification. This is an issue which both taxonomists and data scientists need to tackle in order to provide reliable data for and to facilitate accurate classification of living things.

Impact of the Issues to Online Taxonomy Databases

The issues raised above will create an impact on online taxonomy databases, particularly pertaining to users' expectation to have updated content available to them whenever needed. The decline in the number of expert taxonomists in the near future will impede progress in taxonomic research (Lim and Gibson, 2010). This situation is keenly felt in developing countries, which are often areas of high biodiversity and where not only the number of taxonomists is low but funding for basic taxonomic research is scarce; the net result is often inadequate data quality and analysis and poor data management.

Another impact is bias in research funding. Traditional taxonomy, a fundamental branch of biology, is not perceived as popular as molecular studies, the latter usually preferred by funding bodies. For any research community, funding availability is critical to the progress of the research. In this day and age, dissemination of updated information is done primarily via online means; insufficient funds will also impact dedicated curation of online taxonomic data. Creative funding opportunities must be sought to alleviate the biodiversity crisis, and to promote progress of research on of online taxonomy databases.

Benefits and Risks of Putting Data Online

The necessity for scientific research, taxonomy notwithstanding, to be merged with computational technology is evident (Bik, 2017); this includes making research information publicly available and updated. One benefit of having research data online is to increase visibility of the research, to make them accessible for research continuity and to avoid loss of invaluable data. Even though 'old data' might not have a direct or immediate value, such data might generate new findings after recalibration. Another benefit for putting research data online is to provide public documentation of a topic of research even though it has not gone through a peer-reviewed process. This minimizes the possibility of research hijacking or another party claiming authority over the same research topic as their own. Besides sharing research data online, it is possible to link the data to other information sources via data sharing portals. This enables new analysis and research while increasing the value of the previous research. Current online databases have already started this practice where taxonomic data are linked to genetic databases pertaining to a taxonomy rank. This is discussed further in the next section.

However, there are valid reasons why scientists may not prefer to share their data online. The human factor is perhaps the main stumbling block to online publishing. Publishing taxonomy data exposes information into the public domain. Unlike peer reviewed publications where researchers get scientific credit via the citation process (an important academic performance measure), taxonomists publishing raw data tend to remain anonymous while the web data curators governing the information are known. Another risk is plagiarism. Scientists fear sharing data openly, especially from research in the initial stages, as unethical researchers might illegally use these data and represent them as their original work. Data published online is prone to security threats. There is a risk of data being hacked and manipulated. Data owners who are not computationally savvy or bogged down by other research work may not be able to manage the online data content, which involves creating a change log, and notifying users about the change. Finally, there is a risk of inappropriate use of data whereby they are used to conduct inappropriate or unethical research.

Comparison Between Databases

Taxonomic classification data contain information on scientific names, taxon ranks, morphological features, distributions, and other related information; all these information provided in species profile databases, species lineage or classification information

are valuable. The lineage or classification data in most databases are usually linked to sequence data repositories, such as GenBank (see “Relevant Websites section”). Online taxonomic databases have clearly defined ways to manage taxonomic information. Here we discuss some of the popular databases, with emphasis on fish taxonomy.

FishBase (see “Relevant Websites section”) is a portal managed by the FishBase Consortium group and is one of the trusted online resources on fish. This portal holds a huge amount of information on fish but here we are concerned only with taxonomic classification. FishBase displays species classification data on individual species profile page, such as common names, climate, distribution, and size parameters; the portal presents relevant information specific to the species without covering the species lineage or phylogenetic tree. Nevertheless, FishBase provides a link to the Deep Fin Classification for every species stored in its database.

Another popular taxonomic database is the NCBI Taxonomy Browser managed by the NCBI group (see “Relevant Websites section”). It provides some of the most comprehensive information for a species. In addition to displaying a full lineage of any species of interest, it also provides links to its mitochondrial genetic code, nucleotide sequences and protein sequences. The coverage is elaborated with external information sources which include origin of the name, author, abbreviated taxonomy, and biotic interactions.

FishBase is a database made to handle solely fish data while the NCBI Taxon is developed for classification of living organisms. As such there is a difference in the way data are displayed and organized. FishBase displays information related to the fish species, leaving out information about the lineage of the species. NCBI Taxon, on the other hand may increase the coverage by elaborating on the genetic information of the classification for interested users; when any doubt arises, they may challenge the findings. The possibilities of using these data to produce new research avenues are endless and it is up to the scientific community on how to use it.

Next, we discuss how Wikipedia (see “Relevant Websites section”) handles species information and compare it to more credible sources such as FishBase and NCBI Taxon discussed above. Wikipedia, although not necessarily an authoritative source, is often a preferred choice among students or even researchers wanting to get a quick overview on any topic. Data in Wikipedia are generated mostly by voluntary contributors; thus, available taxonomic information provided on each species page may vary depending on the popularity and extent of knowledge of the species. Nevertheless, there are some standards set by Wikipedia for publishing species information such as full scientific names, scientific classification, conservation status, binomial name, synonyms (if available), and references used to write the species article. Information on species classification, however, is basic and linked to a Wikipedia article page, not another authoritative source. [Table 1](#) presents the comparison between popular online taxonomic databases, including those discussed above.

Ontology and Taxonomic Classification

With the birth of a new Web commonly referred to as the Semantic Web, it is imperative for scientists to embrace the world of data sharing and Big Data. This shift in scientific operation and paradigm requires that the wealth of scientific data, in all manners of format and complexity, be captured and mapped correctly especially in the context of species classification. A preferred way to do this for species data is in the form of an ontology, a formal naming convention that defines any domain of interest and handles information about the types, properties, and interrelationships of these entities.

Ontology is different from a database in terms of their data structure and the way to query data. The most important difference is the triple data structure of an ontology, which creates relationships between a class and properties. As such there are differences in the purpose of creating an online database versus an ontology. The latter is created primarily to give definition to a certain domain; an example is the Gene Ontology that is created specifically to describe genes. It is generally advisable to create lower level domain ontologies with a defined focus, such as gene sequences and skeletal structure, since it is easier to handle and limit the amount of information and to reuse it for other ontologies or databases. Creation of ontologies for large-scale domains such as human or flowers would result in huge amount of data that need to be managed which may contain information that are vague.

In 2017, we published a Fish Ontology (FISHO) (see “Relevant Websites section”), with the aim of performing automated identification of fish using taxonomy ([Ali et al., 2017](#)). The FISHO consists of 652 classes (terms), and 27 object properties (relationships) (can be viewed at “Relevant Websites section”). There are 10 main classes which act as the core classes covering fish-related and non-related terms within the FISHO structure. The FISHO provides terms related to fish and infers species related information based on data that are fed to the ontology. The current version of the FISHO is able to classify jawless fish, early jawed fish and living fossil fish. The FISHO contains 253 classes dedicated to fish studies and 38 classes related to fish sampling processes. [Table 2](#) shows the adoption of the terms in the FISHO using 3 sources which include an authoritative book on fish ([Helfman et al., 2009](#)), the Vertebrate Taxonomy Ontology ([Midford et al., 2013](#)), and the NCBI Taxon ([Federhen, 2016](#)) ([Fig. 1](#)).

The FISHO is created using Protégé ([Protégé, 2018](#)), an open source ontology editor provided by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine. This software contains all the tools needed to develop the FISHO with many supporting features that can assist in the development and visualization of ontology. One of the feature provided by Protégé is the inferring capabilities. In [Fig. 2](#), we show the results of the inferring process within the FISHO. Inferred results of a sample specimen with certain parameters are shown with bright orange color. Once it is inferred, more results can be gained from the inferred results; as an example, Specimen5 was recognized not only as a whale in Infer Result A, the FISHO also inferred that it is not a fish, but a mammal. For Sample2, it is recognized as a longtail carpet shark after the inferring process, and furthermore recognized as an early jawed fish in Infer Result B.

Table 1 Differences between online databases in handling and displaying taxonomic classification information

Database name	Purpose	Classification tree	Link to taxonomic information source	Related Data complementing the taxonomy data	External Links related to the taxonomy data
FishBase (www.fishbase.org)	Fish profile	Partial	Yes	Names, Synonyms, Climates, Size, Distribution	Minimal. (Catalog of Fishes, Deep Fin Classification)
NCBI taxon (www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html)	Taxonomy	Complete including ranks subdivisions	Yes	Taxon Id, Inherited Blast Name, Genetic Code, Mitochondrial Genetic Code, Entrez records	Moderate. (Encyclopedia of Life, Global Biotic Interaction, Dryad Digital Repositories, Arctos Specimen Database)
Wikipedia (www.wikipedia.org)	General knowledge	Partial	Yes	Description, History (if available), Elaborated Information (based on necessity and availability), conservation status, binomial name, synonym	Minimal. (IUCN Redlist, and random links, depending on the author)
WoRMS (www.marinespecies.org)	Marine species registry	Partial	Yes	Status, Ranks, Parents, Original name, Synonyms, Vernacular names, Environment, and Distribution	Comprehensive. (Barcode of Life, Biodiversity Heritage Library, Encyclopedia of Life, and Fishbase to name a few)
The IUCN red list of threatened species (www.iucnredlist.org)	Red list of threatened species	Full	Yes	Names, synonyms, assessment information, population, usage, threats level, and conservation actions	None.
Barcode of life data system (www.boldsystems.org)	Generation & application of DNA barcode data	Full	No	Sample ID, Museum ID, Collection data, Sequencing records (sequence ID, GenBank Accession, Genome, Locus, Nucleotides, Amino Acids, and Illustrative Barcode)	Minimal. (Genbank)

Table 2 Term adoption in the Fish Ontology

Example of terms	Sources			Implementations in the Fish Ontology
	Helfman (2009)	Vertebrate Taxonomy Ontology (VTO)	NCBI Taxon	
Furcacaudiformes (order)	Classified as Subclass of Thelodonti (superclass)	Classified as subclass of Agnatha (class)	Not classified	Follows and reuses the VTO terms
Jawless fish	Contains species and information for jawless fish species	No classes and annotations found, but related species are classified	No classes and annotations found, but related species are classified	Follows Helfman (2009) for labeling
Lobe finned fish	Classified as Actinopterygii (page 4)	No classes and annotations found, but related species are classified	Classified as Coelacanthiformes	Follow Helfman (2009) for classification and labeling
Gobiidae (family)	Listed and classified as family	Listed and classified as family	Listed and classified as family	Follows and reuses the VTO terms
Oxudercinae (subfamily)	Not listed or classified	Not listed or classified	Classified as a subclass of Gobiidae (family)	Follows and reuses the VTO classification up to the lowest existing taxonomic terms covered (Family Gobiidae). Adopts NCBI Taxon terms for subfamily Oxudercinae onwards

The FISHO was reviewed by several experts, one of them being a museum curator who raised concerns over taxonomic classification within the ontology. From a cladistics standpoint, mammals are derived from fish and the inclusion of mammals (and other tetrapods) under Sarcopterygii (an ancient fish group with lobed fins) allows it to be a monophyletic group. However, the main point of creating the ontology is not primarily for taxonomic classification, but to create an automated fish recognition using taxonomy. Presently in the FISHO, the class Mammalia is not placed within the general fish classification structure but under the Chordata phylum. Inclusion of the Mammalia class in the FISHO is due to the existence of some aquatic mammals, such as

BioPortal Login Tools Support

Fish Ontology

Summary Classes Properties Notes Mappings Widgets

Details

ACRONYM	FISHO
VISIBILITY	Public
BIOPORTAL PURL	http://purl.bioontology.org/ontology/FISHO

Metrics

NUMBER OF CLASSES:	386
NUMBER OF INDIVIDUALS:	8
NUMBER OF PROPERTIES:	28

Fig. 1 The Fish Ontology (FISHO) hosted at Bioportal.

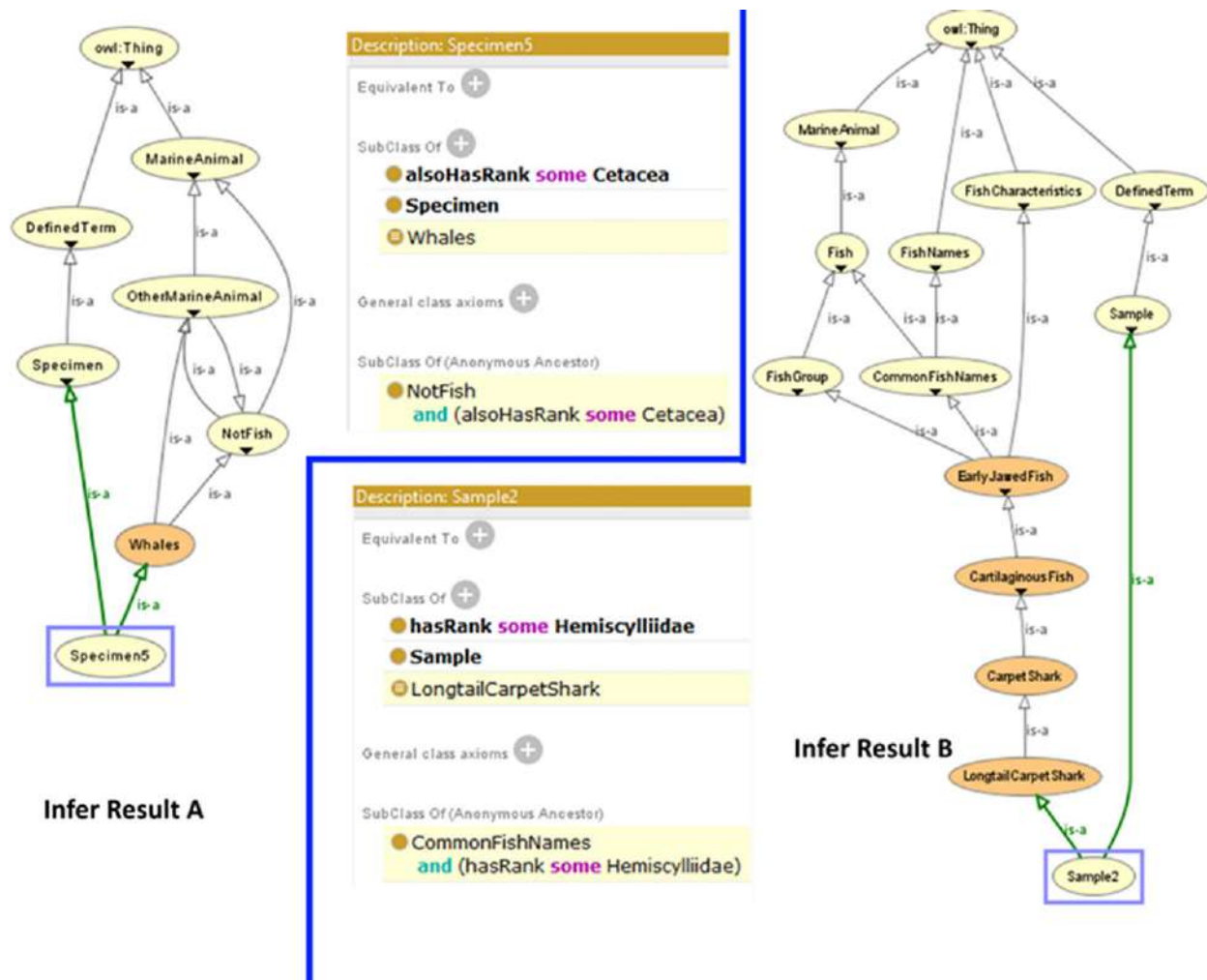


Fig. 2 Inferring process in the FISHO.

whales and dolphins, which may be erroneously classified as fish. Under the ideal FISHO classification (in our opinion), whales would not be recognized as a fish but a mammal. As such, the consequence of explicitly classifying mammals under Sarcopterygii in the FISHO is such that a whale would then be recognized as both a mammal and a fish. Further explanation of this situation is shown in Fig. 3, where we show the specific use case of other sources compared to the FISHO.

Another comment from the reviewer highlighted the confusion about jawless fishes in the FISHO. From a phylogenetic point of view, Agnatha (jawless fish group that includes lampreys, hagfish and ostracoderms) is not a monophyletic group. However, our

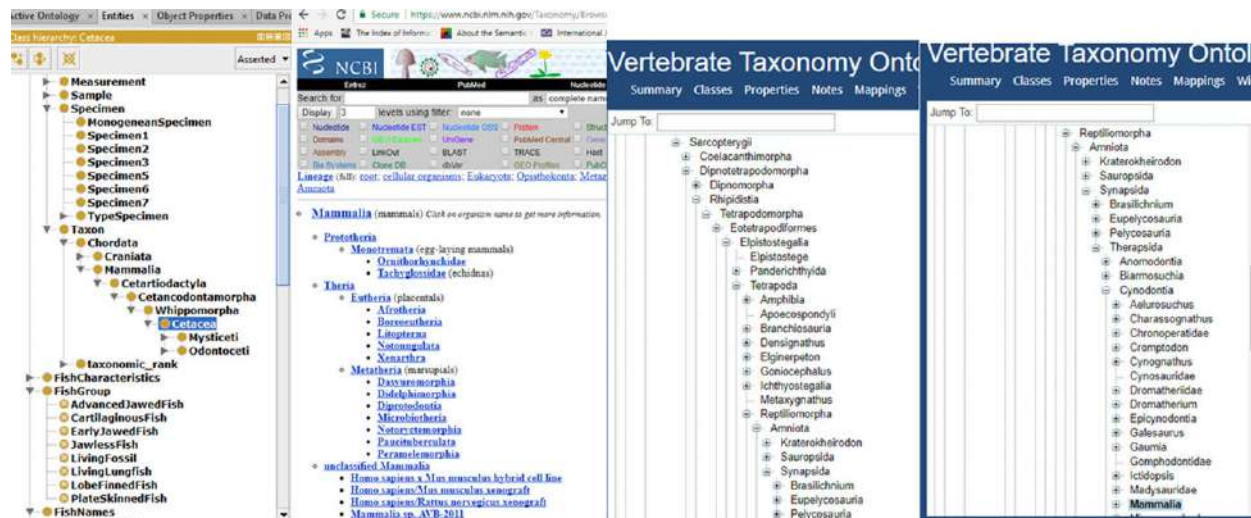


Fig. 3 Different usage of other classification sources compared to the FISHO. The leftmost figure shows how the FISHO classify mammals compared to the NCBI Taxon and the Vertebrate Taxonomy Ontology (VTO).

purpose of including jawless fishes in the FISHO was to capture it as one type of fish grouping or characteristic. For example, a hagfish would be characterized as (or having the attribute of) a jawless fish in the FISHO. This grouping term is not intended as a form of taxonomic classification (i.e., this term is not a class under taxonomic rank, but under Fish Group in the FISHO). The nature of the ontology is such that broad inclusion of relevant fish-related terms such as 'jawless fish' would improve the inference capacity of the FISHO reasoner, thus allowing for relevant and comprehensive search automation and classification of fish.

With these two examples of the challenges faced during the development of the FISHO, one solution that may be adopted to further upgrade the FISHO is to create 2 types of classification, one based on traditional morphology-based taxonomic classification and the other based on modern of phylogenetic taxonomy. Through this approach, the ontology might have the ability to recognize fish specimens using both species and DNA characteristics. This solution may also be applied to other databases or recognition softwares having similar issues. Finally, we propose that future development of computational systems should utilize a variety of species specific data types to speed up the taxonomic classification processes.

Conclusions

An enduring and elusive challenge in biodiversity classification is to determine the numbers of species inhabiting the earth. It had been estimated that there are around 8.7 million species on land and sea, but many are still awaiting description (Mora *et al.*, 2011). This staggering number demonstrates the amount of effort that has been and still needs to be invested to properly classify and capture this biodiversity. As such, the pressure is on for taxonomists to expedite the process and for taxonomic databases to manage the voluminous amount of data accurately and with timely updates.

Scientific data, which refer to taxonomic information in this context, have limited usefulness if they are not read or discovered by others. It has been more than 200 years now since Linnaeus founded taxonomy. His research, and that of his counterparts, had resulted in many natural history drawings, identification keys, species descriptions, and monographs. However, such important records are almost untouched in this digital age. The way forward is to integrate technology with these traditional taxonomic knowledge base. Availability of such data in the online medium will promote secondary research using modern data science techniques, which may give birth to new knowledge and facilitate reproducible research. With the help of semantic data and ontology, digitized taxonomic resources could be made immediately accessible to the public and facilitate seamless integration of new taxonomic information.

See also: Bioinformatics Data Models, Representation and Storage. Data Storage and Representation. Ecosystem Monitoring Through Predictive Modeling. Information Retrieval in Life Sciences. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Molecular Phylogenetics. Natural Language Processing Approaches in Bioinformatics. Population Genetics

References

- Ali, N.M., Khan, H.A., Then, A.Y.-H., *et al.*, 2017. Fish Ontology framework for taxonomy-based fish recognition. *PeerJ* 5, e3811.
 Bacher, S., 2012. Still not enough taxonomists: Reply to Joppa *et al.* *Trends in Ecology & Evolution* 27 (2), 65–66.
 Bik, H.M., 2017. Let's rise up to unite taxonomy and technology. *PLOS Biology* 15 (8), e2002231. Available at: <https://doi.org/10.1371/journal.pbio.2002231>.

- Dirzo, R., Raven, P.H., 2003. Global state of biodiversity and loss. *Annual Review of Environment and Resources* 28 (1), 137–167.
- Federhen, S., 2016. NCBI organismal classification – An ontology representation of the NCBI organismal taxonomy. Retrieved from <http://www.obofoundry.org/ontology/ncbitaxon.html>.
- Gaston, K.J., May, R.M., 1992. Taxonomy of taxonomists. *Nature* 356 (6367), 281–282. <https://doi.org/10.1038/356281a0>.
- Helfman, G.S., Collette, B.B., Facey, D.E., Bowen, B.W., 2009. *The Diversity of Fishes: Biology, Evolution, and Ecology*. vol. 2. Atlantic: John Wiley & Sons.
- Lim, L.H.S., Gibson, D.I., 2010. Taxonomy, taxonomists & biodiversity. In: Manurung, R., Zaliha, C.A., Fasihuddin, B.A., kuek, C. (Eds.), *Biodiversity-Biotechnology: Gateway to Discoveries, Sustainable Utilization and Wealth Creation*. Kuching, Sarawak, Malaysia, pp. 33–43.
- Lee, M.S., Palci, A., 2015. Morphological phylogenetics in the genomic age. *Current Biology* 25 (19), R922–R929.
- Midford, P., Dececchi, T., Balhoff, J., *et al.*, 2013. The vertebrate taxonomy ontology: A framework for reasoning across model organism and species phenotypes. *Journal of Biomedical Semantics* 4 (1), 34.
- Mora, C., Tittensor, D.P., Adl, S., Simpson, A.G.B., Worm, B., 2011. How many species are there on earth and in the ocean? *PLOS Biology* 9 (8), e1001127..
- Protégé. (2018). Stanford Center for Biomedical Informatics Research. Retrieved from <http://protege.stanford.edu/>.
- Tobias, J., 2010. Amazon alive. In: World Wildlife Fund. Retrieved from <https://www.worldwildlife.org/stories/amazon-alive>.
- Treurnicht, M., Colville, J.F., Joppa, L.N., Huyser, O., Manning, J., 2017. Counting complete? Finalising the plant inventory of a global biodiversity hotspot. *PeerJ* 5, e2984.
- Vidal, J., 2010. Protect nature for world economic security, warns UN biodiversity chief. *The Guardian*. Retrieved from <https://www.theguardian.com/environment/2010/aug/16/nature-economic-security>.

Relevant Websites

- www.worldwildlife.org/stories/amazon-alive
Amazon Alive.
- www.boldsystems.org
Barcode of Life Data System.
- <https://bioportal.bioontology.org/ontologies/FISHO>
Fish Ontology.
- www.fishbase.org
FishBase.
- www.ncbi.nlm.nih.gov/genbank
GenBank.
- <https://mohdnajib1985.github.io/FOWebPage/>
mohdnajib1985 · GitHub.
- www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html
NCBI Taxonomy Browser.
- www.theguardian.com
The Guardian.
- www.iucnredlist.org
The IUCN Red List of Threatened Species.
- www.obofoundry.org/ontology/vto.html
Vertebrate Taxonomy Ontology (VTO).
- www.wikipedia.org
Wikipedia.
- www.marinespecies.org
WoRMS.

Ecological Networks

Kazuhiro Takemoto and Midori Iida, Kyushu Institute of Technology, Fukuoka, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the natural world, many species complexly interact with one another via various types of relationships (e.g., commensalism, amensalism, competition, mutualism, prey-predator relationship), and they compose ecosystems (Allesina and Tang, 2012). It is important to understand the function and stability against environmental perturbations (e.g., climate change) of ecosystems in the context of biodiversity maintenance and environmental assessment (Bascompte, 2010; Evans *et al.*, 2013; Pocock *et al.*, 2012). In computational biology, up until now, ecological networks have been mainly examined using theory (mathematical models such as Lotka–Volterra equations (Jansen and Kokkoris, 2003) describing the time evolutions of species populations). For example, May's pioneering theoretical works on the relationship between ecosystem complexity and stability (May, 1976; May, 1972) are interesting because ecosystem stability is related to biodiversity maintenance.

More recently, ecosystems have also been studied from a data analysis perspective. The development of field observation techniques and the improvement of infrastructure such as databases have increased the ecological data availability and have enabled large-scale data analyses of real-world ecosystems. In this context, *network science* plays an important role. Network science in ecology is called *network ecology*. In this article, we will overview the network ecology topic and the approaches to investigate ecological networks.

Network Ecology

Network science is a research area in which complex networks are studied, and it originates from graph theory (Barabási, 2013). Networks describe the relationships among elements, and are, thus, simple and powerful tools for describing complicated systems. The concept of networks is universal and can be applied to a wide range of fields (e.g., mathematics, computer science, economy, sociology, chemistry, biology). In recent years, considerable data on interaction has been accumulated. Thus, networks have become quite important for understanding real-world systems and extracting knowledge of complex systems. For example, network science has been applied to biology. The biomolecules of the living organisms such as proteins and metabolites undergo several interactions and chemical reactions which lead to the occurrence of various life phenomena. To understand the biological processes, it is important to obtain an understanding of the networks; thus, network biology (Barabási and Oltvai, 2004) and network medicine (Barabási *et al.*, 2011) have already attracted attention. Ecological communities are also represented as networks or graphs (so-called *ecological networks* in which nodes and edges correspond to species and inter-specific interactions, respectively). As the availability of ecological data has increased, network science has also been applied to ecology (Proulx *et al.*, 2005).

Ecological networks are often classified according to interaction types (e.g., mutualism and prey–predator relationship). The representative ecological networks are *food webs* and *mutualistic networks*.

Food webs indicate who eats whom, and they have discussed in ecology for a very long time (Kondoh *et al.*, 2010; Thompson *et al.*, 2012). Food webs are represented as networks in which nodes and edges correspond to organisms and trophic links (prey–predator relationships). It should be noted that food webs are generally represented as unipartite directed networks (Fig. 1(A)) because of predator–prey relationships which are direction-oriented. Food webs are often also called *antagonistic networks* and *trophic networks*. However, plant–herbivore food webs are represented as bipartite networks (Muto-Fujita *et al.*, 2017) defined as graphs having two different node sets.

Mutualistic networks are generally mentioned in the context of plant–animal mutualism. The representative mutualistic networks are *pollination networks* (plant–pollinator relationships) and *seed-dispersal networks* (the relationships between plants and animals that transport the plant seeds). The mutualistic networks are represented as bipartite networks (Fig. 1(B)) because mutualistic links are only found between two types of organisms (i.e., plants and animals) (Bascompte *et al.*, 2003; Olesen *et al.*, 2007).

Approaches

Searching for Non-Random Structural Patterns

Which factors determinate ecosystem stability is a long-standing question in ecology (see also Section Structure–Stability Relationship). Earlier theoretical studies (e.g., May, 1976, 1972) indicated that more complex ecosystems (i.e., larger and/or denser ecological networks) are less stable. However, this prediction is inconsistent with the fact that real-world ecosystems are complex. This is known as May's paradox. Resolving the paradox is a central topic in theoretical ecology. The paradox is expected to result from an assumption, in particular, the studies that considered that ecological networks are random for simplicity.

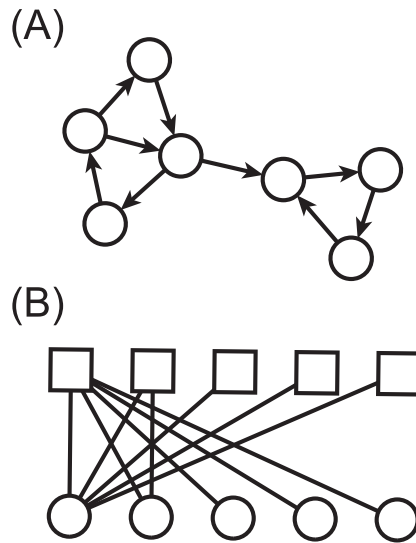


Fig. 1 A network representation of ecological networks. (A) A food web represented as a directed network. Nodes and directed edges correspond to species and prey–predator relationships, respectively. (B) A mutualistic network represented as a bipartite network. Square nodes and circle nodes are plants and animals. Edges are mutualistic relationships (pollination between plants and animals).

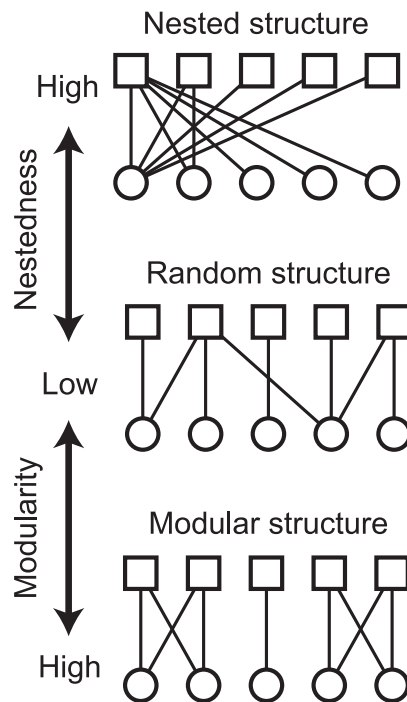


Fig. 2 Schematic diagram of nested structure and modular structure in bipartite networks.

However, is this assumption really sufficient in real-world networks? Network analyses of real-world ecological networks have found that the networks are not random. Specifically, real-world ecological networks are known to display two non-random structural patterns (**Fig. 2**). One structural pattern is the nested architecture (*nestedness*) (Bascompte *et al.*, 2003), a hierarchical structure in which the interaction pairs of a specialist species are included in those of another (generalist) species. In other words, the interaction partners of a specialist species are the subset of those of another generalist species. Another structural pattern is the modular structure (*modularity*) (Olesen *et al.*, 2007), a compartmentalized structure in which a number of dense sub-networks (modules) are weakly interconnected. Despite the correlation between them, these two structural patterns can provide complementary information on how interactions are organized in communities (Fortuna *et al.*, 2010). The degree of nestedness and modularity differ between food webs and mutualistic networks, in particular, the modularity of mutualistic

networks is typically lower than that of food webs, whereas the nestedness of mutualistic networks is generally higher than that of food webs (Bascompte *et al.*, 2003; Thébault and Fontaine, 2010). However, food web subnetworks are significantly nested (Kondoh *et al.*, 2010).

Structure–Stability Relationship

Non-random structural patterns (e.g., nestedness and modularity) are expected to influence ecosystem stability. In fact, nestedness plays important roles in increasing ecosystem stability in mutualistic networks (Bastolla *et al.*, 2009; Saavedra *et al.*, 2016, 2011), and it emerges as a result of an optimization principle aimed at maximizing species abundance (Suweis *et al.*, 2013). Modularity is a particularly important property because it is related to the robustness (Hartwell *et al.*, 1999). Thus, modularity is a significant property of biomolecular networks such as signaling networks (Takemoto and Kihara, 2013) and metabolic networks (Takemoto and Borjigin, 2011; Takemoto and Oosawa, 2012). According to these previous studies, modularity is expected to affect ecosystem stability. In fact, modularity can have moderate stabilizing effects, while anti-modularity can greatly destabilize ecological networks (Grilli *et al.*, 2016). Moreover, both nestedness and modularity influence ecosystem stability (Thébault and Fontaine, 2010). The contributions of nestedness and modularity to ecosystem stability differ between food webs and mutualistic networks. For example, increasing nestedness and/or decreasing modularity enhance the stability of mutualistic networks but reduce the stability of food webs. However, note that several recent studies have cast doubt on the importance of nestedness. For example, nested architecture can be more easily acquired than previously thought (Takemoto and Arita, 2010). A study based on numerical simulation indicated that biodiversity in mutualistic communities is described by the number of mutualistic partners a species has (i.e., node degree) rather than nestedness (James *et al.*, 2012). A theoretical study (Feng and Takemoto, 2014) supports this conclusion.

Effects of Environmental Factors on Ecological Networks

In the context of the structure–stability relationship, the effects of environmental or external factors on ecological networks are also important. Given that environmental factors can be sources of perturbation (e.g., rainfall, seasonal variation of climate), it would be expected that ecological networks have an optimal structure that maximizes the ecosystem stability against such perturbations. Macroecological studies are useful for evaluating the effects of environments on ecological networks (Trojelsgaard and Olesen, 2013), and they have enabled by the combined use of global-scale environmental data (see Section Impact of Global Warming on Ecological Networks for details). Such studies have demonstrated the climate change effects and human impacts on ecological network structure/ecosystem stability.

In the context of the structure–stability relationship, environmental factors influence the nestedness and modularity. Climatic parameters are linked to nestedness and modularity in mutualistic networks and food webs. For example, nestedness in pollination networks decreased with annual precipitation (Trojelsgaard and Olesen, 2013), whereas modularity in seed-dispersal networks and food webs increased with temperature seasonality (Schleuning *et al.*, 2014), and precipitation seasonality (Takemoto *et al.*, 2014), respectively. The type of climatic seasonality influencing the network structure differs among ecosystems. For example, network properties were mainly affected by rainfall seasonality in freshwater ecosystems but primarily by temperature seasonality in terrestrial ecosystems (Takemoto *et al.*, 2014).

The effects of climate change and human activities are more interesting in the context of biodiversity maintenance and environmental assessment. For instance, modularity and nestedness in pollination networks correlated with the historical rate of warming (Dalsgaard *et al.*, 2013) and human impacts (e.g., human population density, land use change, infrastructure development, and so forth) (Takemoto and Kajihara, 2016). Modularity declined and nestedness increased in seed-dispersal networks in response to human impacts (Sebastián-González *et al.*, 2015) and warming velocity (Takemoto and Kajihara, 2016). Food web nestedness increased and modularity declined in response to the global warming (Takemoto and Kajihara, 2016).

Finding Keystone Species

It is also important to characterize local properties (i.e., the characteristics of each node in a complex network) in addition to the global features of ecological networks such as non-random structural patterns. Specifically, finding important (keystone) species in ecological networks is challenging (Jordan, 2009). In this context, *Centrality* analysis is useful which is an important concept in network analysis because it helps in finding central (important) nodes in complex networks (Takemoto and Oosawa, 2012). It plays an important role in network biology and network medicine (Vidal *et al.*, 2011). In general, it is expected that hub species (i.e., species that interact with many other species) are important. However, recent studies encourage a reconsideration of the importance of hubs. For example, hubs can be classified into 2 types (Han *et al.*, 2004): party hubs that coordinate a specific functional modules and date hubs that play a role in intermediates between different specific functional modules. Moreover, they found that the effect of hub removals on complex networks is different between party hubs and date hubs. In particular, the removal of date hubs leads to a more immediate collapse of the networks than that by the removal of party hubs. This finding implies the importance of hubs bridging between different network modules. Therefore, module decomposition or community detection is also useful. In particular, a functional cartography method (Guimerà and Amaral, 2005), revealing patterns of

intra- and inter-module connections in complex networks, and the concept of bottlenecks (Yu *et al.*, 2007) is more useful to find important nodes. The bottlenecks are not always hubs. Importantly, the method can also evaluate the importance of inconspicuous (i.e., low-degree) nodes. In fact, a study examined pollination networks using this method, and it concluded that species serving as hubs and connectors should receive high conservation priorities (Olesen *et al.*, 2007). To detect important nodes based on the concept of bottlenecks, a number of centrality measures have been proposed (Takemoto and Oosawa, 2012). For example, an application of the PageRank algorithm for a website ranking of ecological networks is interesting. Such a centrality analysis is also useful for finding the importance of species for coextinction (Allesina and Pascual, 2009).

Illustrative Examples

In this section, we present some illustrative examples of network ecology approaches along with available databases and analysis tools.

Databases and Network Analysis Tools

Databases on ecological networks

Several databases on ecological networks are available, for example, 359 food webs are available in the GlobalWeb database (Thompson *et al.*, 2012; see Relevant Websites section). The Interaction Web DataBase (see Relevant Websites section) includes a number of ecological networks types, such as 27 food webs, 42 pollination networks, 12 seed-dispersal networks, 4 plant–herbivore networks, 4 plant–ant networks, 7 host–parasite networks, and 2 anemone–fish networks. The Web-of-Life Database (see Relevant Websites section) provides the data on a number of ecological networks: 143 pollination networks, 34 seed-dispersal networks, 34 host–parasite networks, 4 plant–herbivore networks, and 4 plant–ant networks. The Web-of-Life Database is useful for macro-ecological studies because the locations (i.e., latitude and longitude) of observation sites of the networks are also available in the database. Note that there are data duplications among the databases and the statistics as of October 4, 2017.

Network analysis tools

A number of network analysis tools are available to accelerate ecological network studies, in particular, the packages of the R-software (see Relevant Websites section). For example, the package *igraph* (igraph.org) is used for general network analysis (i.e. centrality analysis, community detection). A tutorial of *igraph* is available online at “Network Analysis and Visualization with R and *igraph*” link (see Relevant Websites section). The package *vegan* (see Relevant Websites section) and *bipartite* (see Relevant Websites section) are useful for analyzing mutualistic networks, represented as bipartite networks. The *nested* and *computeModules* functions in the *bipartite* package are used to calculate nestedness and modularity of bipartite networks, respectively. The functional cartography method is also available.

Environmental data

To explore the relationship between environments and ecological networks, environmental data at observation sites of ecological networks are needed. Several useful databases are available for this purpose. For example, the WorldClim database (Hijmans *et al.*, 2005; see Relevant Websites section) provides the gridded climate data with a spatial resolution of about 1 km². These data include annual mean temperature, temperature seasonality, annual precipitation, and rainfall seasonality (see Relevant Websites section). The data can be easily obtained using the R-package *raster*. In addition, the WorldClim database provides the past climate data (e.g., last glacial maximum climate conditions) estimated by the paleoclimate modeling intercomparison project. Climate change velocity is estimated by a comparison between the current and past climate (Dalsgaard *et al.*, 2013; Takemoto and Kajihara, 2016).

For human activities, the human footprint (HF) score from the global human footprint dataset compiled by the last of the wild project is available (Sanderson *et al.*, 2002). The HF score is provided with a spatial resolution of 1–km grid cells. According to the descriptions in the last of the wild database (see Relevant Websites section), HF scores are calculated by normalizing the human influence index defined as the sum of eight human activity-related variables: human population density, human land use change and infrastructure (built-up areas, nighttime lights, land use/land cover), and human access (coastlines, roads, railroads, navigable rivers).

The National Aeronautics and Space Act (NASA) Earth Data website (see Relevant Websites section) may be useful for obtaining other global environmental data.

Evaluating Significance of a Structural Pattern

As mentioned in Section Searching for Non-Random Structural Patterns, it is important to find non-random structural patterns of ecological networks. A comparison with randomized model networks was performed to evaluate the statistical significance of an observed network measure or structural pattern. Randomized networks are generated from a real-world network using edge-switching algorithms (Milo *et al.*, 2002) and configuration models (Catanzaro *et al.*, 2005; see also Takemoto and Oosawa, 2012). The significance of the network measure X was evaluated based on the Z-score: $Z_X = (X_{\text{real}} - X_{\text{rand}})/\text{SD}_{\text{rand}}$, where X_{real} is the

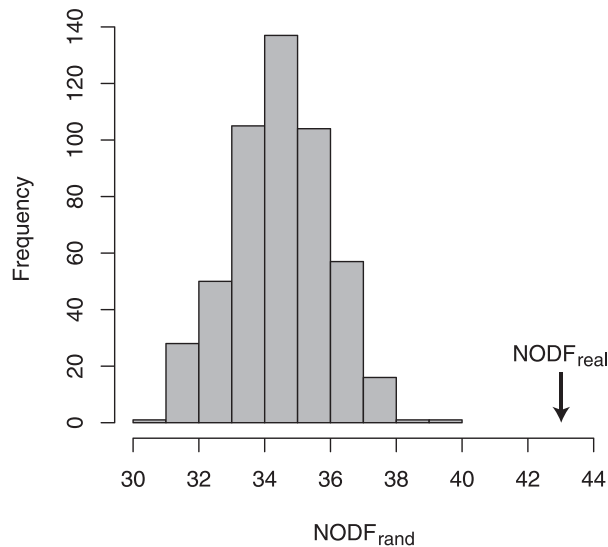


Fig. 3 The distribution of $NODF_{rand}$ obtained from 500 randomized networks compared to $NODF_{real}$.

network measure of a real-world network, $\bar{X}_{rand}$ is the average value of the network measure, and SD_{rand} is the standard deviation obtained from a large number of randomized model networks. Moreover, Z-scores are also used in the context of standardization (i.e., to allow comparisons among matrices).

As an example, we indicated the significance of nestedness in a pollination network (Mommott, 1999) (Fig. 3). Although several definitions of nestedness are proposed, we used the Nestedness metric based on Overlap and Decreasing Fill (NODF) metrics for evaluating nestedness (Almeida-Neto *et al.*, 2008). The NODF metrics is related to the proportion of shared interactions between species pairs over a bipartite network, and it ranges from 0 to 100. A null model based on configuration models (Bascompte *et al.*, 2003) was used to generate 500 randomized networks.

Comparison of Structural Patterns Between Different Types of Ecological Networks

Are structural patterns different between different types of ecological networks? Using a dataset (Takemoto and Kajihara, 2016), we compared nestedness and modularity among food webs, pollination networks and seed-dispersal networks (Fig. 4). We used the standardized NODF (Z_{NODF}) and standardized M (Z_M) for evaluating nestedness and modularity, respectively. M is a well-used modularity score, and it is defined as the fraction of edges that lie within, rather than between, modules relative to that expected by chance (Fortunato, 2010). M ranges from 0 to 1; a network with a higher M indicates a higher modular structure. We used the BIPARTMOD software (Guimerà *et al.*, 2007) (see Relevant Websites section) to calculate the modularity (M) of directed networks and bipartite networks because the BIPARTMOD software finds the maximum M based on simulated annealing to minimize the resolution-limit problem in community detection (Fortunato, 2010; Fortunato and Barthélemy, 2007).

Impact of Global Warming on Ecological Networks

As mentioned in Section Effects of Environmental Factors on Ecological Networks, it is hypothesized that non-random structural patterns are associated with environmental perturbations, given the structure–stability relationship. To test this hypothesis, using a dataset (Takemoto and Kajihara, 2016), we examined the relationship between warming velocity and nestedness/modularity in food webs (Fig. 5). We used the standardized NODF and standardized M to evaluate nestedness and modularity, respectively. Warming velocity (temperature change) is defined as the temporal climate gradient divided by the spatial climate gradient, with the temporal gradient in turn defined as the absolute difference between current and the last glacial maximum (LGM) climate conditions (see Takemoto and Kajihara, 2016 for details).

Results and Discussion

In the pollination network, the observed NODF score (42.8) was significantly higher than the average NODF score (34.5) obtained from the randomized networks (Fig. 3; $Z_{NODF} = 5.8$, $p = 7.2 \times 10^{-9}$ using the Z-test). This result indicated that the pollination network was nested than expected by chance. The significance of nestedness was generally concluded in pollination networks and seed-dispersal networks (i.e., mutualistic networks) (Fig. 4(A)). However, the significance of structural patterns was different between food webs and mutualistic networks. Food webs were less nested than expected by chance, while mutualistic networks

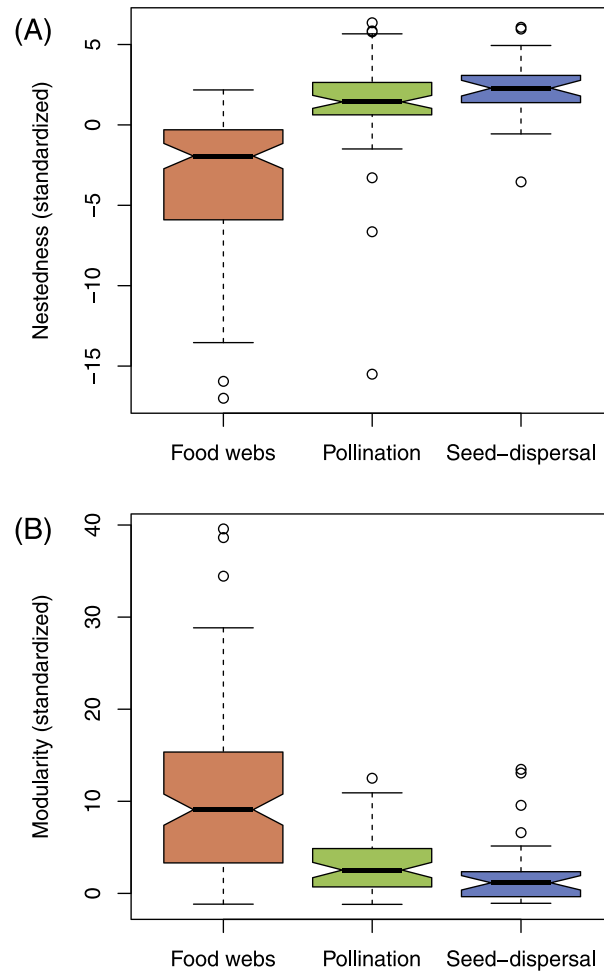


Fig. 4 Differences in nestedness (A) and modularity (B) among food webs, pollination networks, and seed-dispersal networks.

(pollination and seed-dispersal networks) are highly nested (**Fig. 4(A)**; $p < 2.2 \times 10^{-16}$ using the Kruskal–Wallis rank sum test). On the other hand, food webs were highly modularized (compartmentalized) while mutualistic networks were less modularized (**Fig. 4(B)**; $p = 8.2 \times 10^{-13}$ using the Kruskal–Wallis rank sum test). This result may be due to the fact that the contributions of nestedness and modularity to ecosystem stability differ between mutualistic networks and food webs, in that increasing nestedness and/or decreasing modularity enhance the stability of mutualistic networks but reduce the stability of food webs (Thébault and Fontaine, 2010).

A positive correlation between warming velocity and nestedness was observed (Spearman's rank correlation coefficient $r_s = 0.44$, $p = 2.7 \times 10^{-7}$). On the other hand, modularity was negatively correlated with warming velocity ($r_s = -0.30$, $p = 6.1 \times 10^{-4}$) (**Fig. 5**). Increasing nestedness and/or decreasing modularity reduced ecosystem stability in food webs (Thébault and Fontaine, 2010); as such, this result suggested that food-web stability decreases in response to environmental changes. However, more careful examinations are required. In macroecological studies, we need to remove any inherent spatial autocorrelation. In this context, a spatial eigenvector mapping modeling approach is useful (Takemoto and Kajihara, 2016).

Future Directions

Network ecology has scientific and societal impacts; however, it remains controversial.

Weighted Network Analysis

Ideally, ecological networks should be represented as weighted networks because of interaction weights, in particular, a different conclusion may be derived from comparisons between weighted and binary networks. For example, nestedness is statistically significant in binary networks but not in weighted networks (Staniczenko *et al.*, 2013). Temperature seasonality was correlated with the weighted version of modularity, but not with the binary version of modularity (Schleuning *et al.*, 2014). However, many

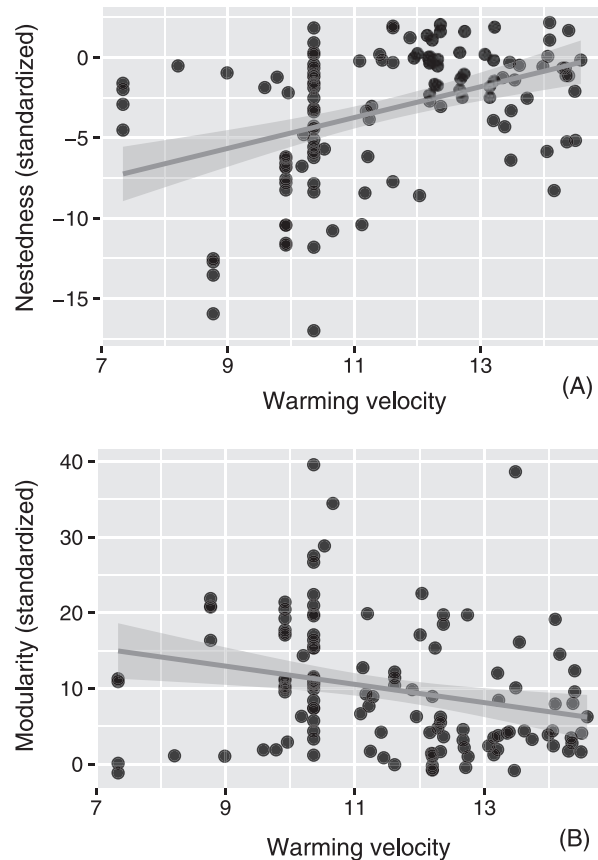


Fig. 5 Scatter plots of a network parameter versus warming velocity in food webs. Nestedness (A) and modularity (B).

of ecological network studies have considered binary networks (i.e., presence: 1, or absence: 0, of a given link) because the datasets include numerous binary data, and many software programs for network analysis generally assume binary networks. This is also because the definition of interaction weight is not uniform throughout the ecological-network datasets. The interaction weight assigned to a given species pair is based on the number of contacts they share. However, the weight need to be corrected (or normalized) for factors such as sampling effort (e.g., observation area and observation time) and species abundance. Such normalization methods differ among studies. Thus, database normalizations are needed (see Section Construction of Highly Normalized Databases).

Multiplex Network Analysis

A mixture of interaction types is essential for more realistic interactions although ecological networks have been generally classified according to the interaction types. In particular, the mixture of interaction types may contribute to increasing ecosystem stability (Allesina and Tang, 2012; Mougi and Kondoh, 2012). Network measures for multiplex networks should be considered in the future to evaluate ecological network structure under more realistic conditions. Multiplex networks consist of a set of nodes connected by different link types, and have been studied in the context of social network analysis (Hoang and Antoncic, 2003). Recently, multiplex ecological network analyses have also begun (Baggio *et al.*, 2016; Pilosof *et al.*, 2017).

Construction of Highly Normalized Databases

More highly normalized databases should be constructed. In particular, information on ecological networks (e.g., species taxonomy, ecosystem type, observed site, observed area, and time) should be summarized. For example, it is possible that sampling effort affects network parameters (e.g., nestedness and modularity) when considering the species–area relationship (McGuinness, 1984) which states that the number of observed species increases with the observed area. However, the relevant information on sampling effort is not often obtained because it is not always clearly delineated in the literature and databases. In addition, the effects of phylogenetic signals should be examined because several studies have reported that phylogenetic signals are weak in ecological networks (Rezende *et al.*, 2007; Schleuning *et al.*, 2014). However, ecological networks partly consist of species whose descriptions are unknown (i.e., they are expressed as species A and species B) or ambiguous. Moreover, a restricted understanding

of interspecific reactions (i.e., missing links) is a more serious limitation. To avoid these limitations, larger-scale and more highly normalized databases should be constructed, and it is especially important that data on weighted networks be expanded. In this context, data sharing (Parr and Cummings, 2005) is also important.

Linking to Dynamics

The relationship between network structure and dynamics (e.g., ecosystem stability) remains unclear. Alternative approaches are needed. Recently, the novel analytical framework revealed the universal relationship between the system's dynamics (coextinctions or cascading failures in ecological networks, in particular) and network topology (Gao *et al.*, 2016). In particular, the authors indicated that a single universal resilience function and parameter can characterize the behavior of different networks. Stochastic models for coextinctions in ecological networks are also useful (Vieira and Almeida-Neto, 2014). Such models are capable of simulating extinction cascades far more complex than those observed in network topology-based models. For example, the models produce complex extinction cascades in which species losses in one trophic level may lead to indirect additional species losses at the same trophic level, and they predicted extinction cascades to be more likely in highly connected ecological communities. However, the validity of these approaches is still debatable; thus, more careful examinations are required. The development of such novel network-based analytical frameworks remains an important topic.

Closing Remarks

This article has described the current understanding of ecological networks revealed through data analysis and theoretical studies. Networks are useful for extracting knowledge of complex ecosystems. Real-world ecological networks are non-random, and such non-random structural patterns are different among the ecological network types. Moreover, the non-random structural patterns influence ecosystems stability, and they are linked to environmental changes. Network ecology approaches enhance our understanding of ecosystem stability and the effects of environmental change on ecosystems. Considering that ecological networks have not been fully understood yet, network ecology is a research field with high future growth potential.

Acknowledgement

KT was partly supported by a Grant-in-Aid for Young Scientists (A) from the Japan Society for the Promotion of Science (no. 17H04703).

See also: Challenges in Creating Online Biodiversity Repositories with Taxonomic Classification. Community Detection in Biological Networks. Ecosystem Monitoring Through Predictive Modeling. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Network Models. Network-Based Analysis for Biological Discovery. Transcriptome Analysis

References

- Allesina, S., Pascual, M., 2009. Googling food webs: Can an eigenvector measure species' importance for coextinctions? *PLOS Comput. Biol.* 5, e1000494. doi:10.1371/journal.pcbi.1000494.
- Allesina, S., Tang, S., 2012. Stability criteria for complex ecosystems. *Nature* 483, 205–208. doi:10.1038/nature10832.
- Almeida-Neto, M., Guimarães, P., Guimarães, P.R., Loyola, R.D., Ulrich, W., 2008. A consistent metric for nestedness analysis in ecological systems: Reconciling concept and measurement. *Oikos* 117, 1227–1239. doi:10.1111/j.0030-1299.2008.16644.x.
- Baggio, J.A., BurnSilver, S.B., Arenas, A., *et al.*, 2016. Multiplex social ecological network analysis reveals how social changes affect community robustness more than resource depletion. *Proc. Natl. Acad. Sci. USA* 113, 13708–13713. doi:10.1073/pnas.1604401113.
- Barabási, A.-L., 2013. Network science. *Philos. Trans. R. Soc. A* 371, 20120375. doi:10.1098/rsta.2012.0375.
- Barabási, A.-L., Oltvai, Z.N., 2004. Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.* 5, 101–113. doi:10.1038/nrg1272.
- Barabási, A.L., Gulbahce, N., Loscalzo, J., 2011. Network medicine: A network-based approach to human disease. *Nat. Rev. Genet.* 12, 56–68. doi:10.1038/nrg2918.
- Bascompte, J., 2010. Structure and dynamics of ecological networks. *Science* 329, 765–766. doi:10.1126/science.1194255.
- Bascompte, J., Jordano, P., Melián, C.J., Olesen, J.M., 2003. The nested assembly of plant-animal mutualistic networks. *Proc. Natl. Acad. Sci. USA* 100, 9383–9387. doi:10.1073/pnas.1633576100.
- Bastolla, U., Fortuna, M.A., Pascual-García, A., *et al.*, 2009. The architecture of mutualistic networks minimizes competition and increases biodiversity. *Nature* 458, 1018–1020. doi:10.1038/nature07950.
- Catanzaro, M., Boguñá, M., Pastor-Satorras, R., 2005. Generation of uncorrelated random scale-free networks. *Phys. Rev. E* 71, 27103. doi:10.1103/PhysRevE.71.027103.
- Dalsgaard, B., Trøjelsgaard, K., Martín González, A.M., *et al.*, 2013. Historical climate-change influences modularity and nestedness of pollination networks. *Ecography (Cop.)* 36, 1331–1340. doi:10.1111/j.1600-0587.2013.00201.x.
- Evans, D.M., Pocock, M.J.O., Memmott, J., 2013. The robustness of a network of ecological networks to habitat loss. *Ecol. Lett.* doi:10.1111/ele.12117. [n/a-n/a].
- Feng, W., Takemoto, K., 2014. Heterogeneity in ecological mutualistic networks dominantly determines community stability. *Sci. Rep.* 4, 5912. doi:10.1038/srep05912.

- Fortuna, M.A., Stouffer, D.B., Olesen, J.M., *et al.*, 2010. Nestedness versus modularity in ecological networks: Two sides of the same coin? *J. Anim. Ecol.* 79, 811–817. doi:10.1111/j.1365-2656.2010.01688.x.
- Fortunato, S., 2010. Community detection in graphs. *Phys. Rep.* 486, 75–174. doi:10.1016/j.physrep.2009.11.002.
- Fortunato, S., Barthélemy, M., 2007. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* 104, 36–41. doi:10.1073/pnas.0605965104.
- Gao, J., Barzel, B., Barabási, A.-L., 2016. Universal resilience patterns in complex networks. *Nature* 530, 307–312. doi:10.1038/nature16948.
- Grilli, J., Rogers, T., Allesina, S., 2016. Modularity and stability in ecological communities. *Nat. Commun.* 7, 1–10. doi:10.1038/NCOMMS12031.
- Guimerà, R., Amaral, L.A.N., 2005. Functional cartography of complex metabolic networks. *Nature* 433, 895–900. doi:10.1038/nature03288.
- Guimerà, R., Sales-Pardo, M., Amaral, L., 2007. Module identification in bipartite and directed networks. *Phys. Rev. E* 76, 36102. doi:10.1103/PhysRevE.76.036102.
- Han, J.-D.J., Bertin, N., Hao, T., *et al.*, 2004. Evidence for dynamically organized modularity in the yeast protein–protein interaction network. *Nature* 430, 88–93. doi:10.1038/nature02555.
- Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W., 1999. From molecular to modular cell biology. *Nature* 402, C47–C52. doi:10.1038/35011540.
- Hijmans, R.J., Cameron, S.E., Parra, J.L., Jones, P.G., Jarvis, A., 2005. Very high resolution interpolated climate surfaces for global land areas. *Int. J. Climatol.* 25, 1965–1978. doi:10.1002/joc.1276.
- Hoang, H., Antoncic, B., 2003. Network-based research in entrepreneurship. *J. Bus. Ventur.* 18, 165–187. doi:10.1016/S0883-9026(02)00081-2.
- James, A., Pitchford, J.W., Plank, M.J., 2012. Disentangling nestedness from models of ecological complexity. *Nature* 487, 227–230. doi:10.1038/nature11214.
- Jansen, V.A.A., Kokkoris, G.D., 2003. Complexity and stability revisited. *Ecol. Lett.* 6, 498–502. doi:10.1046/j.1461-0248.2003.00464.x.
- Jordan, F., 2009. Keystone species and food webs. *Philos. Trans. R. Soc. B Biol. Sci.* 364, 1733–1741. doi:10.1098/rstb.2008.0335.
- Kondoh, M., Kato, S., Sakato, Y., 2010. Food webs are built up with nested subwebs. *Ecology* 91, 3123–3130.
- May, R.M., 1972. Will a large complex system be stable? *Nature* 238, 413–414. doi:10.1038/238413a0.
- May, R.M., 1976. Thresholds and breakpoints in ecosystems with a multiplicity of stable states. *Nature* 260, 471–477. doi:10.1038/269471a0.
- McGuinness, K.A., 1984. Species-area curves. *Biol. Rev.* 59, 423–440. doi:10.1111/j.1469-185X.1984.tb00711.x.
- Mommott, J., 1999. The structure of a plant–pollinator food web. *Ecol. Lett.* 2, 276–280. doi:10.1046/j.1461-0248.1999.00087.x.
- Milo, R., Shen-Orr, S., Itzkovitz, S., *et al.*, 2002. Network motifs: Simple building blocks of complex networks. *Science* 298, 824–827. doi:10.1126/science.298.5594.824.
- Mougi, A., Kondoh, M., 2012. Diversity of interaction types and ecological community stability. *Science* 337, 349–351. doi:10.1126/science.1220529.
- Muto-Fujita, A., Takemoto, K., Kanaya, S., *et al.*, 2017. Data integration aids understanding of butterfly–host plant networks. *Sci. Rep.* 7, 43368. doi:10.1038/srep43368.
- Olesen, J.M., Bascompte, J., Dupont, Y.L., Jordano, P., 2007. The modularity of pollination networks. *Proc. Natl. Acad. Sci. USA* 104, 19891–19896. doi:10.1073/pnas.0706375104.
- Parr, C., Cummings, M., 2005. Data sharing in ecology and evolution. *Trends Ecol. Evol.* 20, 362–363. doi:10.1016/j.tree.2005.04.023.
- Pilosof, S., Porter, M.A., Pascual, M., Kéfi, S., 2017. The multilayer nature of ecological networks. *Nat. Ecol. Evol.* 1, 101. doi:10.1038/s41559-017-0101.
- Pocock, M.J.O., Evans, D.M., Mommott, J., 2012. The robustness and restoration of a network of ecological networks. *Science* 335, 973–977. doi:10.1126/science.1214915.
- Proulx, S.R., Promislow, D.E.L., Phillips, P.C., 2005. Network thinking in ecology and evolution. *Trends Ecol. Evol.* 20, 345–353. doi:10.1016/j.tree.2005.04.004.
- Rezende, E.L., Jordano, P., Bascompte, J., 2007. Effects of phenotypic complementarity and phylogeny on the nested structure of mutualistic networks. *Oikos* 116, 1919–1929. doi:10.1111/j.2007.0030-1299.16029.x.
- Saavedra, S., Rohr, R.P., Olesen, J.M., Bascompte, J., 2016. Nested species interactions promote feasibility over stability during the assembly of a pollinator community. *Ecol. Evol.* 6, 997–1007. doi:10.1002/ece3.1930.
- Saavedra, S., Stouffer, D.B., Uzzi, B., Bascompte, J., 2011. Strong contributors to network persistence are the most vulnerable to extinction. *Nature* 478, 233–235. doi:10.1038/nature10433.
- Sanderson, E.W., Jaiteh, M., Levy, M.A., *et al.*, 2002. The human footprint and the last of the wild. *Bioscience* 52, 891–904. doi:10.1641/0006-3568(2002)052[0891:THFATL]2.0.CO;2.
- Schleuning, M., Ingmann, L., Strauß, R., *et al.*, 2014. Ecological, historical and evolutionary determinants of modularity in weighted seed–dispersal networks. *Ecol. Lett.* doi:10.1111/ele.12245. [n/a–n/a].
- Sebastián-González, E., Dalsgaard, B., Sandel, B., Guimarães, P.R., 2015. Macroecological trends in nestedness and modularity of seed–dispersal networks: Human impact matters. *Glob. Ecol. Biogeogr.* 24, 293–303. doi:10.1111/geb.12270.
- Staniczenko, P.P.A., Kopp, J.C., Allesina, S., 2013. The ghost of nestedness in ecological networks. *Nat. Commun.* 4, 1391. doi:10.1038/ncomms2422.
- Suweis, S., Simini, F., Banavar, J.R., Maritan, A., 2013. Emergence of structural and dynamical properties of ecological mutualistic networks. *Nature* 500, 449–452. doi:10.1038/nature12438.
- Takemoto, K., Arita, M., 2010. Nested structure acquired through simple evolutionary process. *J. Theor. Biol.* 264, 782–786. doi:10.1016/j.jtbi.2010.03.029.
- Takemoto, K., Borjigin, S., 2011. Metabolic network modularity in Archaea depends on growth conditions. *PLOS ONE* 6, e25874. doi:10.1371/journal.pone.0025874.
- Takemoto, K., Kajihara, K., 2016. Human impacts and climate change influence nestedness and modularity in food-web and mutualistic networks. *PLOS ONE* 11, e0157929. doi:10.1371/journal.pone.0157929.
- Takemoto, K., Kanamaru, S., Feng, W., 2014. Climatic seasonality may affect ecological network structure: Food webs and mutualistic networks. *Biosystems* 121, 29–37. doi:10.1016/j.biosystems.2014.06.002.
- Takemoto, K., Kihara, K., 2013. Modular organization of cancer signaling networks is associated with patient survivability. *Biosystems* 113, 149–154. doi:10.1016/j.biosystems.2013.06.003.
- Takemoto, K., Oosawa, C., 2012. Introduction to complex networks: Measures, statistical properties, and models. *Stat. Mach. Learn. Approaches Netw. Anal.* 45–75. doi:10.1002/9781118346990.ch2.
- Thébault, E., Fontaine, C., 2010. Stability of ecological communities and the architecture of mutualistic and trophic networks. *Science* 329, 853–856. doi:10.1126/science.1188321.
- Thompson, R.M., Brose, U., Dunne, J.A., *et al.*, 2012. Food webs: Reconciling the structure and function of biodiversity. *Trends Ecol. Evol.* 27, 689–697. doi:10.1016/j.tree.2012.08.005.
- Trøjelsgaard, K., Olesen, J.M., 2013. Macroecology of pollination networks. *Glob. Ecol. Biogeogr.* 22, 149–162. doi:10.1111/j.1466-8238.2012.00777.x.
- Vidal, M., Cusick, M.E., Barabási, A.-L., 2011. Interactome Networks and Human Disease. *Cell* 144, 986–998. doi:10.1016/j.cell.2011.02.016.
- Vieira, M.C., Almeida-Neto, M., 2014. A simple stochastic model for complex coextinctions in mutualistic networks: Robustness decreases with connectance. *Ecol. Lett.* 18, 144–152. doi:10.1111/ele.12394.
- Yu, H., Kim, P.M., Sprecher, E., Trifonov, V., Gerstein, M., 2007. The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLOS Comput. Biol.* 3, e59. doi:10.1371/journal.pcbi.0030059.

Further Reading

- Takemoto, K., Oosawa, C., 2012. Introduction to complex networks: Measures, statistical properties, and models. *Stat. Mach. Learn. Approaches Netw. Anal.* 45–75. doi:10.1002/9781118346990.ch2.
- Hastings, A., Gross, L. (Eds.), 2012. *Encyclopedia of Theoretical Ecology*. University of California Press. Available at: <http://www.jstor.org/stable/10.1525/j.ctt1pp0s7>.

Relevant Websites

[CRAN.R-project.org/package=bipartite](https://cran.r-project.org/package=bipartite)

Bipartite: Visualising Bipartite Networks and Calculating Some (Ecological) Indices

www.globalwebdb.com

GlobalWeb.

<https://www.globalwebdb.com>

GlobalWeb: An Online Collection of Food Webs.

www.nceas.ucsb.edu/interactionweb

Interaction Web DataBase.

earthdata.nasa.gov

NASA.

<http://kateto.net/networks-r-igraph>

Network Analysis and Visualization With R and Igraph.

<http://networksciencebook.com>

Network Science.

seeslab.info/downloads/bipartite-modularity

SEESlab.

sedac.ciesin.columbia.edu/data/collection/wildareas-v2/methods

Socioeconomic Data and Applications Center (SEDAC)

www.R-project.org

The R Project for Statistical Computing.

[CRAN.R-project.org/package=vegan](https://cran.r-project.org/package=vegan)

Vegan: Community Ecology Package

www.web-of-life.es

Web of Life.

<http://www.web-of-life.es>

Web of Life: Ecological Networks Database.

www.worldclim.org

WorldClim

Global Climate Data.

www.worldclim.org/bioclim

WorldClim

Global Climate Data: Bioclimatic Variables.

Biographical Sketch



Kazuhiro Takemoto received his doctoral degree in Informatics from Kyoto University in 2008 after earning his bachelor's and master's degrees in Computer Science from Kyushu Institute of Technology (Kyutech) in 2004 and 2006, respectively. After serving as a JSPS research fellow (2007–2009), a postdoctoral fellow at University of Tokyo (2009), Japan Science Technology Agency PRESTO researcher (2009–2013) and assistant professor at Kyutech (2012–2015), he is currently an associate professor at Kyutech (2015–). His research interests are network science, computational and integrative biology.



Midori Iida received her doctoral degree in Science from Ehime University in 2013 after earning her bachelor's and master's degrees in the university in 2008 and 2010, respectively. She is currently a postdoctoral fellow at Kyushu Institute of Technology (2013–). Her research interests are ecotoxicology, aquatic life science, and bioinformatics.

Dedicated Bioinformatics Analysis Hardware

Bertil Schmidt and Andreas Hildebrandt, Institut für Informatik, Mainz, Germany

© 2019 Elsevier Inc. All rights reserved.

Introduction

Recent years have seen a tremendous increase in the volume of data generated in the life sciences, especially propelled by the rapid progress of next-generation sequencing (NGS) technologies. For example, the sequence read archive (SRA) from NCBI which stores sequence data obtained from NGS technology now contains well over 10^{16} base-pairs (bps) of DNA sequence data and doubles in size roughly every 18 months. It is estimated that between one hundred million and two billion human genomes could be sequenced within the next decade (Stephens *et al.*, 2015) which will have a major impact on many areas of life in general with many applications in precision and personalized medicine as well as drug discovery (Schmidt and Hildebrandt, 2017).

This review provides an overview of the usage of GPUs, FPGAs, and dedicated hardware in bioinformatics. We provide some background information in Section Background/Fundamentals. In Section Applications we discuss a number of typical bioinformatics applications that can benefit from massive parallelism while Section Illustrative Examples focuses on two illustrative examples. We finish with a discussion, upcoming trends, and closing remarks in Sections Discussion, Future Directions, and Closing Remarks, respectively.

Bioinformatics is a wide field. We focus on sequence-based and structure-based applications. Further popular areas such as the usage of GPU/FPGA computing in systems biology, genome-wide analysis, or mass spectrometer data processing are outside the scope of our review.

Background/Fundamentals

Inherent to sequence-based approaches is a very common abstraction in bioinformatics – the representation of biopolymers (proteins, DNA, and RNA) as one-dimensional strings or sequences of characters. While this abstraction works surprisingly well for a wide variety of application scenarios, more detailed insight into bio-molecular function often requires more complex representations.

In structural or structure-based bioinformatics, molecules are represented as collections of atoms with properties such as element type, partial charge, covalent bonds, radius, and three-dimensional position and velocity in classical or semi-classical models, or by models of their quantum mechanical wave functions in quantum chemistry.

Even though sequences and structures are different molecular abstractions, both sequence-based and structure-based bioinformatics have the need for efficient implementations of core algorithms in common, albeit generally for different reasons (while sequence-based bioinformatics needs to scale to exploding experimental data set sizes, structure-based approaches generate tremendous amounts of floating-point computations). The traditional approach to implement bioinformatics applications is based on standard microprocessors (CPUs). Unfortunately, the compute performance provided by CPUs is often insufficient to meet current and future requirements. For example, it is projected that variant calling from NGS data alone requires around two trillion CPU hours by 2025 (Stephens *et al.*, 2015). Thus, bioinformatics data analysis shows the increased importance of computational techniques and the usage of high performance computing (HPC) infrastructures (Schatz, 2015).

Technology relevant for HPC applications comes in different variants, each with their own characteristics. Accelerator technologies such as Field Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs) can be an attractive option compared to large-scale compute clusters and clouds, especially in terms of price-performance ratio and energy-efficiency.

GPUs can provide around one order-of-magnitude higher peak performance compared to CPUs through massive fine-grained parallelism at a highly competitive price-performance ratio (Owens *et al.*, 2008). Using programming languages such as CUDA or OpenCL they can be used for general-purpose applications. While applications in structural bioinformatics are typically floating point-based and highly compute-intensive, sequence analysis algorithms are usually integer-based and more data-intensive. In order to realize an efficient GPU-parallelization, both careful algorithm design and optimized implementation are required.

Field programmable gate arrays (FPGAs) are programmable hardware chips mainly consisting of configurable logic gates and memory blocks (Compton and Hauck, 2002). FPGA configurations are generally specified using a hardware description languages such as VHDL or Verilog. FPGAs are particularly attractive for applications in cryptography and sequence analysis where relatively simple highly regular operations are performed on short integers. Thus, using FPGAs for analyzing large-scale sequence data offers a great opportunity for algorithm acceleration.

Finally, if monetary cost is not an issue, dedicated hardware can be designed for the problem at hand in the form of so-called application-specific integrated circuits (ASICs). These hand-tailored processors can achieve unparalleled efficiency and performance, but suffer from inflexibility with respect to alternative workloads.

Applications

BLAST

The most popular software to search a library of sequences with a query sequence is BLAST. BLAST is a family programs of which BLASTN (Nucleotide-nucleotide BLAST) and BLASTP (Protein-protein BLAST) are most widely used. The heuristics used consist of a pipeline of several stages, which have different characteristics (e.g., two-hit method in BLASTP vs. word-matching in BLASTN) but also contain common components.

Acceleration of BLAST on FPGAs and GPUs has received much attention. Mercury BLASTP (Jacob *et al.*, 2008) and Mercury BLASTN (Lancaster *et al.*, 2009) implement all stages of NCBI BLASTP and NCBI BLASTN. They achieve speedups of around one order-of-magnitude on a system with two Xilinx Virtex-II 6000-6 FPGAs compared to NCBI BLAST executed on two Opteron CPUs. The more recent CAAD BLAST (Mahram and Herbordt, 2015) reports a 5x speedup on a single Virtex-6 FPGA over a fully parallel implementation of the reference code on a modern multi-core CPU. Wienbrandt (2014) presents a BLASTP implementation that scales to 128 Spartan3-5000 FPGAs. Timelogic's Tera-BLAST (2017) is a commercial FPGA implementation of various BLAST algorithms. GPU-BLASTP (Vouzis and Sahinidis, 2010), CUDA-BLASTP (Liu *et al.*, 2011a,b), and cuBLASTP (Zhang *et al.*, 2015) are implementations of BLASTP on CUDA-enabled GPUs. G-BLASTN (Zhao and Chu, 2014) and HS-BLASTN (Chen *et al.*, 2015) report massively parallel GPU accelerations of BLASTN and MegaBLAST. The recent H-BLAST (Ye *et al.*, 2017) achieves speedups ranging between 4 and 10 on one K20x GPU over the sequential NCBI-BLASTP.

NGS Read Mapping

The *mapping* of NGS reads to a reference genome sequence is usually one of the first steps in many NGS data processing pipelines. Most existing read mappers (or aligners) are based on a seed-and-extend approach where short exact matches (seeds) are first identified using an index data structure. These seeds are then verified whether they can be extended to a full alignment. Approaches to accelerate read mapping can be categorized by the utilized indexing data structure.

Early GPU approaches including CUSHAW (Liu *et al.*, 2012) and SOAP3-dp (Luo *et al.*, 2013) are based on the Burrows Wheeler transform (BWT) with FM-index. This approach has a low memory footprint and thus can fit into the limited GPU global memory. Arioc (Wilton *et al.*, 2015) and PEANUT (Koster and Rahmann, 2014) are examples of more recent GPU-accelerated read mapper based on hash tables. nvBowtie is a GPU-accelerated of the popular Bowtie2 software built on top of the NVBio (NVIDIA, see Relevant Websites section) library achieving a speedup of up to 8 on a K80 GPU compared to Bowtie2 running with 20 CPU threads. BowMapCL (Nogueira *et al.*, 2016) is based on OpenCL instead of CUDA in order to support multiple heterogeneous accelerators reporting speedups between 2 and 7.5 on a single GeForce GTX Ti GPU compared to multi-threaded Bowtie and BWA. Chen *et al.* (2013) proposed a hash-based short read mapper for FPGAs. Houtgast *et al.* (2015) implemented an FPGA-accelerated as well as a GPU-accelerated (Houtgast *et al.*, 2017) version of BWA-MEM, which achieve roughly the same performance. Arram *et al.* (2017) leverage FPGAs to accelerated Bowtie2 and achieve a 28x speedup on a Maxeler MPC-X2000 node consisting of eight Altera Stratix-V FPGAs compared to multi-threaded Bowtie2 running with 16 CPU threads. Fernandez *et al.* (2015) reported a 12-fold speed gain on a Conway HC-2ex system compared to Bowtie running eight CPU threads.

Variant Calling

Variant calling is an important operation performed on the alignments returned by mapping NGS reads to a reference genome. GSNP (Lu *et al.*, 2011) is a GPU-accelerated version of SOAPsnv reporting speedups of around 40. Luo *et al.* (2014) leverages the power of GPUs to implement a pipeline consisting of read mapping, re-alignment and variant calling called BALSA. It is able to process 90 samples from the 1000 genomes project in around 3 days on a cluster with five GTX680 GPUs. The commercial DRAGEN platform provides FPGA-accelerated implementations of genome and transcriptome NGS analysis pipelines. Miller *et al.* (2015) report whole genome sequencing (WGS) on the DRAGEN system within 26-h time from blood sample to provisional diagnosis.

Denovo Assembly and Error Correction

While approaches for accelerating denovo genome assembly have been limited, the time-consuming pre-processing step for correcting errors in NGS reads has received considerable more attention. CUDA-EC (Shi *et al.*, 2010) was the first approach to speedup error correction (EC) on GPUs based on Bloom filters. DecGPU (Liu *et al.*, 2011a,b) improved this approach to GPU clusters. nvLighter is a GPU-accelerated re-engineering of the Lighter error corrector using the NVBio library (NVIDIA, see Relevant Websites section). Ramachandran *et al.* (2015) perform FPGA-accelerated EC achieving speedups of around 40x on a Stratix V FPGA compared to the CPU-based BLESS algorithm.

Multiple Sequence Alignment

Progressive alignment is a widely used but time-consuming approach for computing multiple sequence alignments (MSAs). MSA-CUDA (Liu *et al.*, 2009b) was the first GPU-implementation of all three stages of the ClustalW pipeline. CUDA ClustalW (Hung *et al.*, 2015) is an efficient GPU implementation of ClustalW v2 on multiple GPUs. G-MSA (Blazewicz *et al.*, 2013) accelerates the

T-Coffee algorithm to achieve around two orders-of-magnitude speedup. QuickProbs (Gudys and Deorowicz, 2014) is a variant of the highly accurate MSAProbs (Liu *et al.*, 2010b) algorithm suited for GPUs. A reconfigurable computing approach proposed by Oliver *et al.* (2005b) accelerates the first stage of ClustalW by computing pairwise alignments with a linear systolic array. Lloyd and Snell (2011) mapped the third phase onto FPGAs reporting a speedup of up to 150 versus a single core. Mahram and Herbordt (2012) accelerate ClustalW through pipelined prefiltering on a FPGA.

Molecular Force Fields

A fundamental abstraction in structural bioinformatics is the notion of a molecular force field. In principle, simulating molecular behaviour at atomic resolution would require modelling quantum mechanics, as non-classical effects are crucial at such length scales. Such models, however, have very large computational demands. Molecular force fields address this problem by prescribing collections of interatomic interaction energies that would lead to a behaviour compatible with quantum mechanics and experimental evidence. For example, a quadratic energy component can be used to model covalent bonds, as long as the expected length deviations are not too large. More realistic models for larger deviations can be obtained using higher order polynomials or exponential models, albeit at greater computational cost.

Force fields contain two different classes of interaction: *bonded* and *non-bonded*. Bonded interactions occur only between atoms sharing a covalent bond, while non-bonded ones (typically van-der-Waals interactions and electrostatics) can occur between any pair of atoms. Since the number of bonds per atom is bounded by a small number, the number of bonded interactions is linear in the number of atoms in the system. Non-bonded interactions, on the other hand, grow quadratically and, hence, typically dominate computational cost of force field evaluations.

Depending on the system under consideration and the required accuracy, more sophisticated types of force fields may be needed, such as polarizable (Baker, 2015) or reactive (Farah *et al.*, 2012). On the other hand, force fields may be coarse-grained over collections of atoms to save on computational cost (Barnoud and Monticelli, 2015).

Molecular Docking

Molecules exert influence on one another through physical interactions. To regulate a system as complex as a living cell, or even a multi-cellular organism, at the molecular level requires a high degree of specificity: a molecule intended to activate or suppress a particular target can not be allowed to blindly interact with any other protein in the vicinity. The physical origin of this specificity is the short range and relatively weak strength of the pair-wise interactions: to exert a noticeable influence, many pairs of atoms on the surfaces of both molecules need to come into close contact. This implies a degree of geometric compatibility. To use Emil Fischer's famous image, one molecule fits like a key into the other molecule's lock (Fischer, 1894), or more appropriately using Koshland's induced fit-model (Koshland, 1958), like a hand into a glove that is slightly too small, modifying its shape along the way.

Predicting whether a given molecule could bind to an active site of another, and with what binding strength, is of great interest to many fields of science and industry, such as drug design. In principle, the structure and binding strength of such molecular complexes could be studied using molecular dynamics (MD), given suitable force fields. The computational effort required for this task, though, is currently out of reach outside of dedicated MD hardware such as Anton 2. However, in many cases it might not be necessary to study the full dynamics of molecular complexes to predict molecular binding. Empirically, we often find that one geometric arrangement of such a complex has significantly lower energy than others and, hence, dominates the partition function of the system (Rarey *et al.*, 1996). Predicting this structure and approximating its binding energy is the task of docking algorithms.

In essence, docking methods are optimization procedures for some scoring function approximating the binding strength with respect to the molecular degrees of freedom. In rigid docking, only six degrees of freedom are optimized: the translation and rotation of one of the partners. In reality, though, molecules are non-rigid and can undergo flexible rearrangements upon binding, which greatly increases the dimensionality of the optimization problem as well as the number of local minima.

Most work on the GPU-based acceleration of flexible docking algorithms was devoted to porting heuristic global optimization approaches, such as genetic algorithms, simulated annealing, or differential evolution, or to vectorize the interaction energy calculations required for these methods (Korb *et al.*, 2011). Autodock (Morris *et al.*, 2009), for instance, has seen several user-contributed ports to the GPU, some of which are quite popular. In addition, Autodock has been ported to FPGAs (Pechan and Feher, 2011), yielding significant speed-up compared to a CPU-implementation, but being more or less on par with a GPU variant.

Illustrative Examples

Pairwise Sequence Alignment

For pairwise sequence similarity computation, the Smith-Waterman (SW) algorithm, the Needleman Wunsch (NW) algorithm, and their variants are widely used. They compute the optimal local, global, or semi-global alignment of two sequences under a given scoring scheme by means of dynamic programming (DP). The associated time complexity proportional to the product of sequence lengths makes this method time-consuming in many application scenarios. Consequently, parallelization approaches

have been proposed for a variety of hardware platforms. We compare the achieved performance in terms of billion cell updates per second (GCUPS).

Early solutions based on dedicated hardware were already proposed in the 1980s. [Lipton and Lopresti \(1985\)](#) proposed one of the first designs based on a systolic array implemented on an ASIC. Systolic array approaches are typically based on a linear array of simple processing elements (PEs) working in lock-step. The DP matrix is typically calculated using a *wavefront* scheme, whereby in every clock cycle a minor diagonal of the DP matrix can be calculated in parallel.

Consider two sequences S_1 and S_2 of length l_1 and l_2 . Assuming there are l_1 PEs each storing one character of S_1 , then we can compute the whole DP matrix in $l_1 + l_2$ time steps instead of $\mathcal{O}(l_1 \cdot l_2)$ on a single processor. If the number of PEs is less than the sequence length, a suitable partitioning scheme needs to be implemented.

FPGA-based solutions typically implement the systolic array approach on reconfigurable hardware instead of an ASIC. An early solution for DNA sequences using linear gap scoring by [Hoang and Lopresti \(1992\)](#) used a SPLASH board consisting of 32 Xilinx XC3090 FPGA chips to implement a linear of 248 PEs to achieve 0.27 GCUPS. [Oliver et al. \(2005a\)](#) proposed a systolic array of 252 PEs to scan a database of protein sequences with a query sequence using affine gap penalties and achieved 5.8 GCUPS on a single Virtex II XC2V6000 FPGA. [Zhang et al. \(2007\)](#) further improved the PE design to achieve 25.6 GCUPS on an Altera Stratix II FPGA. [Benkrid et al. \(2009\)](#) improved the flexibility with respect to supporting different sequence types and scoring schemes by implementing a parameterized SW algorithm FPGA-based accelerator. Recent work by [Xia et al. \(2017\)](#) extends the design from a score-only alignment computation to performing backtracking on a linear systolic array to achieve a performance of 106 GCUPS using 512 PEs on a Xilinx XC7VX1140T FPGA.

SW has also been implemented on a number of commercial FPGA systems. [Wienbrandt \(2014\)](#) reports 6020 GCUPS for aligning DNA sequences on the SciEngines RIVYERA S6-LX150 platform equipped with 128 FPGAs of type Xilinx Spartan6-LX150. [Vermij \(2011\)](#) achieves a performance of 460 GCUPS on the Convey HC1 system composed of four XC5VLX330 FPGA chips for SW without backtracking. Another commercial solution is from Timelogic (see Relevant Websites Section).

The first implementation of SW using CUDA was proposed by [Manavski and Valle \(2008\)](#) who achieved a performance of up to 3.6 GCUPS on a GeForce 8800 GTX. CUDASW++ 1.0 ([Liu et al., 2009a](#)) considerably improved performance to 16 GCUPS on a GeForce GTX 295 by a more careful consideration of the GPU memory hierarchy and the introduction of the inter-task parallelization approach where each CUDA thread computes a distinct alignment of a protein sequence database search task. The successor versions CUDASW++ 2.0 ([Liu et al., 2010a](#)) and CUDASW++ 3.0 ([Liu et al., 2010a](#)) further improved performance to 29.7 GCUPS on a GeForce GTX 295 and 185.6 GCUPS on a dual-GPU GeForce GTX 690, respectively. SWHybrid ([Lan et al., 2017](#)) is currently fastest CUDA implementation of SW-based protein database search achieving close to 300 GCUPS on a GeForce GTX 1080.

CUDAlign focuses on the computation of a single pairwise alignment of long (chromosome-length) DNA sequences. CUDAlign 1.0 ([Sandes and Melo, 2010](#)) calculates the DP matrix with the wavefront method and achieved a performance of 20.4 GCUPS on a GeForce GTX 280. CUDAlign 2.1 ([Sandes and Melo, 2013](#)) incorporates the retrieval of optimal local alignments in linear space with a performance 58.2 GCUPS on a GeForce 560Ti. The latest version ([Sandes et al., 2016](#)) achieves 10,370 GCUPS on a cluster with 384 GPUs for computing exact chromosome-wide DNA alignments.

Molecular Dynamics

Today, molecular dynamics (MD) simulations clearly belong to the most important computational techniques to study molecular properties ([Karplus and McCammon, 2002](#)). Starting from an initial configuration (i.e., position and velocity vectors for every atom in the system), a chemical parametrization (i.e., information about the chemical type for each atom) and a molecular force field, it solves for the classical equations of motion of the involved particles, with suitable modifications to achieve, e.g., constant temperature or pressure ([Rapaport, 2004](#)).

Integrating the equations of motion requires very small time steps, leading to large numbers of iterations to simulate biologically relevant time scales. Since system sizes can become quite large and since the number of non-bonded interaction evaluations grows quadratically, computation of such pairwise interaction energies typically dominates MD running times. Using increasingly long simulations of increasingly large systems with increasingly complex force fields hence leads to a growing demand for computational resources. Consequently, MD codes made use of GPU accelerators almost immediately after NVIDIA introduced CUDA in 2007 ([Stone, 2007](#); [Liu et al., 2008](#)). Today, all practically relevant MD implementations are GPU-accelerated, and in many cases scale well to multi-GPU setups.

A fully featured MD application is a complex, often modular, system with components that cover the whole MD pipeline from input preparation up to trajectory analysis. Porting the entirety of this pipeline to the GPU would hardly be suitable. Hence, modern MD codes provide GPU-accelerated implementations for time-critical parts of the pipeline that vectorize well and typically cover the majority of the computational work. Most vectorization efforts focus on the computation of non-bonded interactions, which are further decomposed into short-range and long-range interactions depending on the distance between the two interacting atoms.

Considering the functional form of the van-der-Waals interaction, which very quickly saturates to zero with distance, it is often possible to neglect all van-der-Waals terms between atoms at a distance beyond a user-defined threshold (optionally, the terms can be smoothly pulled to zero in a small region just below the threshold using a switch-off function) and hence treat these as short-ranged interactions only. For electrostatics, though, this approximation is often unsuitable. The most popular method to handle these interactions computationally is the so-called *particle mesh Ewald* (PME) summation ([Darden et al., 1993](#)), which separates the sum of all pairwise electrostatic interactions into a short-ranged and a long-ranged part as described above. The first part converges

quickly in real space and can hence be directly computed. To parallelize the second part, three different strategies can be employed (Rovigatti *et al.*, 2015):

1. Atom-decomposition, where each compute unit receives a block of atoms and computes all interactions of each atom in the block,
2. Force-decomposition, where all potentially interacting atom pairs are distributed over compute units, and
3. Space-decomposition, where a compute unit calculates the interactions of atoms within a certain spatial domain.

Atom-decomposition is currently the most popular approach on the GPU by far, implemented, e.g., in GROMACS, NAMD, or HOOMD-blue, even though force-decomposition might be preferable in some cases (Rovigatti *et al.*, 2015).

The second term of the above decomposition (long-ranged contributions to non-bonded interactions) converges slowly in real space, but quickly in reciprocal- or Fourier space. For periodic boundary conditions, this suggests to use fast Fourier transform algorithms on charge- and potential grids to convert them to Fourier space where suitable cut-offs enable efficient computation of the interaction contributions. While PME reduces computational effort on long-range interaction from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log(n))$, it is harder to parallelize than pairwise interaction summation due to non-negligible communications overhead. Hence, many MD packages, such as GROMACS (Pall and Hess, 2013; Pall *et al.*, 2015) or YASARA (Krieger and Vriend, 2015) use GPU acceleration only for the short-range contributions while keeping the rest (including PME) on the CPU which, incidentally, allows to keep CPU and GPU busy simultaneously. Other packages such as AMBER PMEMD (Götz *et al.*, 2012; Salomon-Ferrer *et al.*, 2013), ACEMD (Harvey *et al.*, 2009a,b), or HOOMD-blue (Anderson *et al.*, 2008; Glaser *et al.*, 2015) report that more than 90% of their computational workload is offloaded to the GPU.

When porting MD codes to accelerators, a further challenging complication arises from inherent numerical instabilities (e.g., through cancellation effects in summing over many small interactions). As shown by Colberg and Höfling (2011), single precision floating point calculations are insufficient even in simple situations using only Lennard-Jones potentials. While modern GPUs can handle double precision floating point computations reasonably well, albeit at the cost of greatly reduced performance. Hence, modern MD code often uses what is known as *mixed* precision or SPDP-mode (Götz *et al.*, 2012), where only sensitive operations are computed in double precision. Alternatively, fixed-point integer arithmetic or even extended precision can be used as a replacement for double precision in so-called SPFP-mode (Le Grand *et al.*, 2013) or SPXP-mode (Thall, 2006).

All this effort seems to be justified in practice. According to a white paper by NVIDIA, running at least part of the code on the GPU typically leads to speed-ups of 3x to 8x when compared to multi-threaded CPU implementations.

Considering the enormous compute requirements of MD simulations, it is not surprising that other kinds of accelerator hardware besides commodity graphics cards have been and are still being actively investigated. FPGA-based solutions such as Yang *et al.* (2007), Khan *et al.* (2013), Waidyasooriya *et al.* (2016) have repeatedly proven to yield significantly better performance than CPU-only code, but seem to suffer from sub-optimal floating point performance, in particular for those parts where double precision is required. The importance of MD as an indispensable tool for molecular studies even justifies the enormous cost of designing special-purpose MD ASIC solutions. RIKEN's MDGRAPE-4 system (Ohmura *et al.*, 2014) consists of 512 specially designed systems-on-chip (SoC), each of which has 64 pipelines for the computation of non-bonded interactions, running at 0.8 GHz and 65 Ten-silica Xtensa LX configurable processor cores with single-precision floating point units, clocked at 0.6 GHz, for the remaining calculations. Each SoC has a peak performance of 51.2 billion interaction computations per second. Optical transmitters and receivers are used for internode connections.

D.E. Shaw Research's (DESRES) Anton-architecture (Shaw *et al.*, 2008, 2014) in its second iteration is the first platform that was able to simulate several microseconds per day on systems composed of millions of atoms. All parts of the simulation pipeline are handled by the ASICs without resorting to general purpose commodity host processors. To fully leverage the system, DESRES also developed new MD algorithms adapted to the strengths of the system. The Anton 2 systems in operation consist of 512 ASIC nodes, each of which dedicates about one quarter of its die area to the computation of pair-wise interactions and additionally contains 66 general purpose programmable cores. Each ASIC is clocked at 1.65 GHz.

As demonstrated by the Anton system, the ability to routinely study protein behaviour at the micro- or even millisecond range is truly transformative, both for understanding general principles of biochemistry and protein organization and for dedicated biological or pharmaceutical applications (Chung *et al.*, 2015; Hu *et al.*, 2015; Pan *et al.*, 2017).

Discussion

In sequence-based bioinformatics the usage of GPU/FPGA/dedicated hardware is driven by the enormous progress of NGS technologies. Typical algorithms often rely on combinations of integer-intensive computations (such as dynamic programming for alignments) or big data techniques (such as large-scale index data structures). As many applications involving a single genome have already been parallelized, we are starting to see first acceleration approaches for analyzing *metagenomic* NGS data. MEGAHIT (Li *et al.*, 2016) assembles large and complex metagenomic datasets using a parallel algorithm for constructing succinct de Bruijn Graphs on GPUs while (Kobus *et al.*, 2017) perform metagenomic read classification on GPUs.

On the other hand, structural bioinformatics models biomolecules as three-dimensional objects with non-trivial and often dynamic geometry. It relies on concepts from computational physics, such as interaction energies and force fields, and from computational geometry, such as geometric queries and comparisons. Such models are typically based on floating-point

computations, often even at double precision. With increasing system sizes, simulation time steps, and model accuracy, the required amount of floating point operations grows immensely. Hence, structural bioinformatics quickly adopted accelerator technologies. Here, we have discussed some of the typical application scenarios focusing on MD and docking.

Compared to MD, using accelerators for docking has received considerably less attention. In fact, not all docking methods seem to map well to GPUs or FPGAs at all. One area that does profit significantly from GPU acceleration is rigid docking, which today is typically only considered suitable as part of a larger pipeline including flexibilization steps. Many rigid docking algorithms are based on computing correlations through multiplication in Fourier space, as pioneered by Katchalski-Katzir *et al.* (1992). FFT computation can easily be offloaded to the GPU, which has been used to speed up docking based on spherical harmonics (Ritchie and Venkatraman, 2010) or in the classical Katchalski-Katzir sense (Ohue *et al.*, 2014). In addition, the code has been ported to Intel's MIC architecture, where it was found to perform less well than on GPUs (Ohue *et al.*, 2014).

Future Directions

Programming of massively parallel accelerators can still be cumbersome. Implementing hand-optimized algorithms from scratch using languages like CUDA can be time-consuming since it requires a deep understanding of GPU architectures. However, this might change with the availability of highly optimized libraries and the advancements of pragma-based languages such as OpenACC. Furthermore, a major hurdle in taking up FPGA-based systems can be their programming complexity, leading to long software development cycles. Furthermore, application software is often not compatible across different FPGA generations. However, the recent development of higher level programming languages such as OpenCL for FPGAs and the availability of associated libraries offer an opportunity to write more portable yet efficient code.

Bioinformatics is becoming increasingly data-intensive which also implies an increased application of *deep learning* techniques (Ching *et al.*, 2017). Central to these techniques is large performance gains on parallel hardware such as GPUs or FPGAs. A recent example is *DeepVariant* (DePristo and Poplin, 2017) a tool for variant calling from NGS data that trains a highly complex neural network from MSAs.

However, increasingly complex neural networks and data sets require an even higher performance which motivates the recent development of ASICs (such as Google's Tensor Processing Unit (TPU)) or special functional components (such as the integration of Tensor cores in Volta GPUs). Thus, we can expect the regular usage of such hardware in bioinformatics analysis in the near future.

Closing Remarks

In this article we have surveyed the usage of parallel accelerator solutions for sequence-based and structure-based bioinformatics. State-of-the-art tools based on GPUs, FPGAs, or even ASICs can provide significant speedups compared to traditional CPU-based solutions. Since biological data sets continue to increase rapidly this approach provides a promising future direction to address the *big data deluge* challenge in the 'omics' sciences.

See also: Computing for Bioinformatics. Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Infrastructures for High-Performance Computing: Cloud Infrastructures. Models and Languages for High-Performance Computing. Natural Language Processing Approaches in Bioinformatics. Parallel Architectures for Bioinformatics

References

- Anderson, J.A., Lorenz, C.D., Travesset, A., 2008. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics* 227 (10), 5342–5359.
- Arram, J., *et al.*, 2017. Leveraging FPGAs for accelerating short read alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 14 (3), 668–677.
- Baker, C.M., 2015. Polarizable force fields for molecular dynamics simulations of biomolecules. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 5 (2), 241–254.
- Barnoud, J., Monticelli, L., 2015. Coarse-grained force fields for molecular simulations. *Methods in Molecular Biology* 1215, 125–149.
- Benkrid, K., Liu, Y., Benkrid, A., 2009. A highly parameterized and efficient FPGA-based skeleton for pairwise biological sequence alignment. *IEEE Transactions on VLSI* 17 (4), 561–570.
- Blazewicz, J., *et al.*, 2013. G-MSA – A GPU-based, fast and accurate algorithm for multiple sequence alignment. *Journal of Parallel and Distributed Computing* 73 (1), 32–41.
- Chen, Y., *et al.*, 2015. High speed BLASTN: An accelerated MegaBLAST search tool. *Nucleic Acids Research* 43 (16), 7762–7768.
- Chen, Y., Schmidt, B., Maskell, D., 2013. A hybrid short read mapping accelerator. *BMC Bioinformatics* 14 (1), 67.
- Ching, T., *et al.*, 2017. Opportunities and obstacles for deep learning in biology and medicine. *bioRxiv*. 142760.
- Chung, H.S., *et al.*, 2015. Structural origin of slow diffusion in protein folding. *Science* 349 (6255), 1504–1510.
- Colberg, P.H., Höfling, F., 2011. Highly accelerated simulations of glassy dynamics using GPUs: Caveats on limited floating-point precision. *Computer Physics Communications* 182 (5), 1120–1129.
- Compton, K., Hauck, S., 2002. Reconfigurable computing: A survey of systems and software. *ACM Computing Surveys* 34 (2), 171–210.
- Darden, T., York, D., Pedersen, L., 1993. Particle mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems. *The Journal of Chemical Physics* 98 (12), 10089–10092.
- DePristo, M., Poplin, R., 2017. *DeepVariant: Highly accurate genomes with deep neural networks*. Available at: <https://research.googleblog.com/2017/12/deepvariant-highly-accurate-genomes.html>

- Farah, K., Müller-Plathe, F., Böhm, M.C., 2012. Classical reactive molecular dynamics implementations: State of the art. *ChemPhysChem* 13 (5), 1127–1151.
- Fernandez, E.B., *et al.*, 2015. FFAST: Fpga-based acceleration of Bowtie in hardware. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12 (5), 973–981.
- Fischer, E., 1894. Einfluss der Konfiguration auf die Wirkung der Enzyme. *Berichte Der Deutschen Chemischen Gesellschaft* 27 (3), 2985–2993.
- Glaser, J., *et al.*, 2015. Strong scaling of general-purpose molecular dynamics simulations on GPUs. *Computer Physics Communications* 192, 97–107.
- Götz, A.W., Williamson, M.J., Xu, D., *et al.*, 2012. Routine microsecond molecular dynamics simulations with AMBER on GPUs. *Journal of Chemical Theory and Computation* 8 (5), 1542–1555.
- Gudy, A., Deorowicz, S., 2014. QuickProbs – A fast multiple sequence alignment algorithm designed for graphics processors. *PLOS ONE* 9 (2), e88901.
- Harvey, M.J., De Fabritiis, G., 2009a. An implementation of the smooth particle-mesh Ewald (PME) method on GPU hardware. *Journal of Chemical Theory and Computation* 5, 2371–2377.
- Harvey, M.J., Giupponi, G., De Fabritiis, G., 2009b. ACEMD: Accelerating biomolecular dynamics in the microsecond time scale. *Journal of Chemical Theory and Computation* 5 (6), 1632–1639.
- Hoang, D.Z., Lopresti, D., 1992. FPGA implementation of systolic sequence alignment. *International Workshop on Field Programmable Logic and Applications*. 183–191.
- Houtgast, E.J., *et al.*, 2017. An efficient GPU-accelerated implementation of genomic short read mapping with BWA-MEM. *ACM SIGARCH Computer Architecture News* 44 (4), 38–43.
- Houtgast, E.J., *et al.*, 2015. An FPGA-based systolic array to accelerate the BWA-MEM genomic mapping algorithm. In: 2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), IEEE.
- Hung, C., *et al.*, 2015. CUDA ClustalW: An efficient parallel algorithm for progressive multiple sequence alignment on Multi-GPUs. *Computational Biology and Chemistry* 58, 62–68.
- Hu, X., *et al.*, 2015. The dynamics of single protein molecules is non-equilibrium and self-similar over thirteen decades in time. *Nature Physics* 12 (2), 171–174.
- Jacob, A., *et al.*, 2008. Mercury BLASTP: Accelerating protein sequence alignment. *ACM Transactions on Reconfigurable Technology and Systems* 1 (2), 9.
- Karplus, M., McCammon, J.A., 2002. Molecular dynamics simulations of biomolecules. *Nature Structural Biology* 9 (9), 646–652.
- Katchalski-Katzir, E., *et al.*, 1992. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences of the United States of America* 89 (6), 2195–2199.
- Khan, M.A., Chiu, M., Herbordt, M.C., 2013. FPGA-accelerated molecular dynamics. In: Benkrid, K., Vanderbauwhede, W. (Eds.), *High-Performance Computing Using FPGAs*. New York, NY: Springer, p. 105135.
- Kobus, R., *et al.*, 2017. Accelerating metagenomic read classification on CUDA-enabled GPUs. *BMC Bioinformatics* 18 (1), 11.
- Korb, O., Stutzle, Exner, T.E., 2011. Accelerating molecular docking calculations using graphics processing units. *Journal of Chemical Information and Modeling* 51 (4), 865–876.
- Koshland, D.E., 1958. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences of the United States of America* 44 (2), 98–104.
- Koster, J., Rahmann, S., 2014. Massively parallel read mapping on GPUs with the q-group index and PEANUT. *PeerJ* 2, e606.
- Krieger, E., Vriend, G., 2015. New ways to boost molecular dynamics simulations. *Journal of Computational Chemistry* 36 (13), 996–1007.
- Lancaster, J., Buhler, J., Chamberlain, R.D., 2009. Acceleration of ungapped extension in Mercury BLAST. *Microprocessors and Microsystems* 33 (4), 281–289.
- Lan, H., Liu, W., Liu, Y., Schmidt, B., 2017. SWHybrid: A hybrid-parallel framework for large-scale protein sequence database search. *IEEE IPDPS 2017*, 42–51.
- Li, D., *et al.*, 2016. MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods* 102, 3–11.
- Le Grand, S., Gtz, A.W., Walker, R.C., 2013. SPFP: Speed without compromise – A mixed precision model for GPU accelerated molecular dynamics simulations. *Computer Physics Communications* 184 (2), 374–380.
- Lipton, R.J., Lopresti, D., 1985. A systolic array for rapid string comparison. In: *Proceedings of the Chapel Hill Conference on VLSI*. 363–376.
- Liu, Y., Schmidt, B., Maskell, D., 2010a. CUDASW ++ 2.0: Enhanced Smith-Waterman protein database search on CUDA-enabled GPUs based on SIMD and virtualized SIMD abstractions. *BMC Research Notes* 3 (1), 93.
- Liu, W., *et al.*, 2008. Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA. *Computer Physics Communications* 179 (9), 634–641.
- Liu, W., Schmidt, B., Müller-Wittig, K.W., 2011a. CUDA-BLASTP: Accelerating BLASTP on CUDA-enabled graphics hardware. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8 (6), 1678–1684.
- Liu, Y., Maskell, D., Schmidt, B., 2009a. CUDASW ++: Optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. *BMC Research Notes* 2 (1), 73.
- Liu, Y., Schmidt, B., Maskell, D., 2009b. MSA-CUDA: Multiple sequence alignment on graphics processing units with CUDA. In: *Proceedings of the 20th IEEE International Conference Application-specific Systems, Architectures and Processors*.
- Liu, Y., Schmidt, B., Maskell, D., 2010b. MSAProbs: Multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. *Bioinformatics* 26 (16), 1958–1964.
- Liu, Y., Schmidt, B., Maskell, D., 2011b. DecGPU: Distributed error correction on massively parallel graphics processing units using CUDA and MPI. *BMC Bioinformatics* 12 (1), 85.
- Liu, Y., Schmidt, B., Maskell, D., 2012. CUSHAW: A CUDA compatible short read aligner to large genomes based on the BurrowsWheeler transform. *Bioinformatics* 28 (14), 1830–1837.
- Lloyd, S., Snell, Q.O., 2011. Accelerated large-scale multiple sequence alignment. *BMC Bioinformatics* 12 (1), 466.
- Lu, M., *et al.*, 2011. GSNP: A DNA single-nucleotide polymorphism detection system with GPU acceleration. In: 2011 International Conference on Parallel Processing (ICPP), IEEE.
- Luo, R., *et al.*, 2013. SOAP3-dp: Fast, accurate and sensitive GPU-based short read aligner. *PLOS ONE* 8 (5), e65632.
- Luo, R., *et al.*, 2014. BALS: Integrated secondary analysis for whole-genome and whole-exome sequencing, accelerated by GPU. *PeerJ* 2, e421.
- Mahram, A., Herbordt, M.C., 2012. FMSA: FPGA-accelerated ClustalW-based multiple sequence alignment through pipelined prefiltering. In: *Proceedings of the 20th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, IEEE.
- Mahram, A., Herbordt, M.C., 2015. NCBI BLASTP on high-performance reconfigurable computing systems. *ACM Transactions on Reconfigurable Technology and Systems* 7 (4), 33.
- Manavski, S., Valle, G., 2008. CUDA compatible GPU cards as efficient hardware accelerators for Smith Waterman sequence alignment. *BMC Bioinformatics* 9 (S2),
- Miller, N.A., *et al.*, 2015. A 26-h system of highly sensitive whole genome sequencing for emergency management of genetic diseases. *Genome Medicine* 7 (1), 100.
- Morris, G.M., *et al.*, 2009. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry* 30 (16), 2785–2791.
- Nogueira, D., Tomas, P., Roma, N., 2016. BowMapCL: Burrows-wheeler mapping on multiple heterogeneous accelerators. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13 (5), 926–938.
- Ohmura, I., *et al.*, 2014. MDGRAPE-4: A special-purpose computer system for molecular dynamics simulations. *Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences* 372 (2012),
- Owens, J.D., *et al.*, 2008. GPU computing. *Proceedings of the IEEE* 96 (5), 879–899.
- Ohue, M., *et al.*, 2014. MEGADOCK 4.0: An ultra-high-performance protein-protein docking software for heterogeneous supercomputers. *Bioinformatics* 30 (22), 3281–3283.
- Oliver, T., *et al.*, 2005a. Using reconfigurable hardware to accelerate multiple sequence alignment with ClustalW. *Bioinformatics* 21 (16), 3431–3432.
- Oliver, T.F., Schmidt, B., Maskell, D.L., 2005b. Reconfigurable architectures for bio-sequence database scanning on FPGAs. *IEEE Transactions on Circuits and Systems II* 52 (12), 851–855.
- Pall, S., Abraham, M.J., Kutzner, C., Hess, B., Lindahl, E., 2015. Tackling Exascale Software Challenges in Molecular Dynamics Simulations With GROMACS. *Springer*. pp. 3–27.
- Pall, S., Hess, B., 2013. A flexible algorithm for calculating pair interactions on SIMD architectures. *Computer Physics Communications* 184 (12), 2641–2650.
- Pan, A.C., *et al.*, 2017. Quantitative characterization of the binding and unbinding of millimolar drug fragments with molecular dynamics simulations. *Journal of Chemical Theory and Computation* 13 (7), 3372–3377.

- Pechan, I., Feher, B., 2011. Molecular docking on FPGA and GPU platforms. In: 2011 Proceedings of the 21st International Conference on Field Programmable Logic and Applications, pp. 474–477. IEEE.
- Ramachandran, A., *et al.*, 2015. FPGA accelerated DNA error correction. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE.
- Rapaport, D.C., 2004. The Art of Molecular Dynamics Simulation. Cambridge University Press.
- Rarey, M., *et al.*, 1996. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology* 261 (3), 470–489.
- Ritchie, D.W., Venkatraman, V., 2010. Ultra-fast FFT protein docking on graphics processors. *Bioinformatics* 26 (19), 2398–2405.
- Rovigatti, L., *et al.*, 2015. A comparison between parallelization approaches in molecular dynamics simulations on GPUs. *Journal of Computational Chemistry* 36 (1), 1–8.
- Salomon-Ferrer, R., Gtz, A.W., Poole, D., Le Grand, S., Walker, R.C., 2013. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. explicit solvent particle mesh ewald. *Journal of Chemical Theory and Computation* 9 (9), 3878–3888.
- Sandes, E.F.O., Melo, A., 2010. CUDAlign: Using GPU to accelerate the comparison of megabase genomic sequences. *ACM SIGPLAN Notices* 45 (5), 137–146.
- Sandes, E.F.O., Melo, A., 2013. Retrieving Smith–Waterman alignments with optimizations for megabase biological sequences using GPU. *IEEE Transactions on Parallel and Distributed Systems* 24 (5), 1009–1021.
- Sandes, E.F.O., Miranda, G., Ayguade, E., *et al.*, 2016. CUDAlign 4.0: Incremental speculative traceback for exact chromosome-wide alignment in GPU clusters. *IEEE Transactions on Parallel and Distributed Systems* 27 (10), 2838–2850.
- Schmidt, B., Hildebrandt, A., 2017. Next-generation sequencing: Big data meets high performance computing. *Drug Discovery Today* 22 (4), 712–717.
- Schatz, M.C., 2015. Biological data sciences in genome research. *Genome Research* 25 (10), 1417–1422.
- Shaw, D.E., *et al.*, 2014. Anton 2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In: SC14: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 41–53. IEEE.
- Shi, H., *et al.*, 2010. A parallel algorithm for error correction in high-throughput short-read data on CUDA-enabled graphics hardware. *Journal of Computational Biology* 17 (4), 603–615.
- Stephens, Z.D., *et al.*, 2015. Big Data: Astronomical or genomic? *PLOS Biology* 13, e1002195.
- Stone, J.E., Phillips, J.C., Freddolino, P.L., *et al.*, 2007. Accelerating molecular modeling applications with graphics processors. *Journal of Computational Chemistry* 28 (16), 2618–2640.
- Thall, A., 2006. Extended-precision floating-point numbers for GPU computation. *ACM SIGGRAPH 2006 Research Posters*, pp. 1–12.
- Vermij, E., 2011. Genetic sequence alignment on a supercomputing platform. MS Thesis, TU Delft, Netherlands.
- Vouzis, P.D., Sahinidis, N.V., 2010. GPU-BLAST: Using graphics processors to accelerate protein sequence alignment. *Bioinformatics* 27 (2), 182–188.
- Waidyasooriya, H.M., Hariyama, M., Kasahara, K., 2016. Architecture of an FPGA accelerator for molecular dynamics simulation using OpenCL. In: 2016 IEEE/ACIS Proceedings of the 15th International Conference on Computer and Information Science (ICIS), p. 15.
- Wienbrandt, L., 2014. The FPGA-based high-performance computer RIVY-ERA for applications in bioinformatics. *Conference on Computability in Europe*. 383–392.
- Wilton, R., *et al.*, 2015. Arioc: High-throughput read alignment with GPU-accelerated exploration of the seed-and-extend search space. *PeerJ* 3, e808.
- Xia, F., *et al.*, 2017. FPGASW: Accelerating large-scale Smith–Waterman sequence alignment application with backtracking on FPGA linear systolic array. *Interdisciplinary Sciences: Computational Life Sciences*. 1–13.
- Yang, X., Mou, S., Dou, Y., 2007. FPGA-accelerated molecular dynamics simulations: An overview. *Reconfigurable Computing: Architectures, Tools and Applications*. 293–301.
- Ye, W., *et al.*, 2017. H-BLAST: A fast protein sequence alignment toolkit on heterogeneous computers with GPUs. *Bioinformatics* 33 (8), 1130–1138.
- Zhang, J., Wang, H., Feng, W., 2015. cublastp: Fine-grained parallelization of protein sequence search on cpu + gpu. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- Zhang, P., Tan, G., Gau, G.R., 2007. Implementation of the Smith–Waterman algorithm on a reconfigurable supercomputing platform. In: *Proceedings of the 1st international workshop on high-performance reconfigurable computing technology and applications*, pp. 39–48.
- Zhao, K., Chu, X., 2014. G-BLASTN: Accelerating nucleotide alignment by graphics processors. *Bioinformatics* 30 (10), 1384–1391.

Further Reading

- Anderson, J.A., Lorenz, C.D., Travesset, A., 2008. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics* 227 (10), 5342–5359.
- Compton, K., Hauck, S., 2002. Reconfigurable computing: A survey of systems and software. *ACM Computing Surveys* 34 (2), 171–210.
- Le Grand, S., Gtz, A.W., Walker, R.C., 2013. SPFP: Speed without compromise – A mixed precision model for GPU accelerated molecular dynamics simulations. *Computer Physics Communications* 184 (2), 374–380.
- Liu, Y., Wirawan, A., Schmidt, B., 2013. CUDASW ++ 3.0: Accelerating Smith–Waterman protein database search by coupling CPU and GPU SIMD instructions. *BMC Bioinformatics* 14, 117.
- Miller, N.A., *et al.*, 2015. A 26-h system of highly sensitive whole genome sequencing for emergency management of genetic diseases. *Genome Medicine* 7 (1), 100.
- Oliver, T.F., Schmidt, B., Maskell, D.L., 2005b. Reconfigurable architectures for bio-sequence database scanning on FPGAs. *IEEE Transactions on Circuits and Systems II* 52 (12), 851–855.
- Owens, J.D., *et al.*, 2008. GPU computing. *Proceedings of the IEEE* 96 (5), 879–899.
- Schatz, M.C., 2015. Biological data sciences in genome research. *Genome Research* 25 (10), 1417–1422.
- Schmidt, B., Hildebrandt, A., 2017. Next-generation sequencing: Big data meets high performance computing. *Drug Discovery Today* 22 (4), 712–717.
- Shaw, D., *et al.*, 2008. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM* 51 (7), 91.
- Stephens, Z.D., *et al.*, 2015. Big Data: Astronomical or genomic? *PLOS Biology* 13, e1002195.
- Stone, J.E., Phillips, J.C., Freddolino, P.L., *et al.*, 2007. Accelerating molecular modeling applications with graphics processors. *Journal of Computational Chemistry* 28 (16), 2618–2640.

Relevant Websites

- <https://developer.nvidia.com/nvbio>
NVIDIA.
- <http://www.timelogic.com/catalog/757>
TimeLogic.
- <http://www.timelogic.com/catalog/758/decyphersw>
TimeLogic.

Biographical Sketch

Bertil Schmidt is tenured Full Professor and Chair for Parallel and Distributed Architectures at the University of Mainz, Germany. Prior to that he was a faculty member at Nanyang Technological University (Singapore) and at University of New South Wales (UNSW). His research group has designed a variety of algorithms and tools for Bioinformatics mainly focusing on the analysis of large-scale sequence and read datasets. For his research work, he has received a GPU Research Center award, GPU Education Center Award, CUDA Academic Partnership award, CUDA Professor Partnership award and Best Paper Awards at IEEE ASAP 2015 and IEEE ASAP 2009.

Andreas Hildebrandt is tenured Full Professor and Chair for Bioinformatics and Software Engineering at Johannes Gutenberg University Mainz in Germany. Previously, he was the head of the independent junior research group on protein–protein interactions and computational proteomics at the Center for Bioinformatics of Saarland University. His main research interests include structural bioinformatics, computational proteomics and computational systems biology, where his group develops and applies novel techniques and program packages. He is also one of the core developers of the BALL library.

Computational Pipelines and Workflows in Bioinformatics

Jeremy Leipzig, Drexel University, Philadelphia, PA, United States

© 2019 Elsevier Inc. All rights reserved.

Background and Fundamentals

Bioinformatic analyses aim to extract biological insights from raw data. Tenable approaches to such goals must be composed of discrete ordered steps, performed either in serial or in parallel. These steps must be communicated to other scientists, and generalized to accept both new biological samples and parameters. In many cases these steps, or *transformations*, have traditionally been performed by compiled software designed for Unix-compatible operating systems on distributed grid or cluster hardware at large academic and commercial institutions. As scientific computing has matured and borrowed from the greater technology community, this approach is evolving to further abstract out both hardware and operating infrastructure using portable, scalable, and expressive software.

Modern bioinformatics has been largely dominated by sequence analysis, whose processing and organizational demands has catalyzed the growth of pipelines and workflows in a field that has witnessed an explosion in throughput over the last two decades – from capillary gel electrophoresis systems (ABI 3700) capable of sequencing 300 kbp per day in 1999 (Mullikin and McMurry, 1999) to a sequencing by synthesis (SBS) machines capable of sequencing 16 whole human genomes (49.6 bpb; Illumina NovaSeq 6000) at a depth of 30x – a 165,333% increase. However, virtually every -omics subfield, as well as other life sciences and physical sciences, deal with the same need to formalize ad-hoc processes for the purposes of robustness, growth, transparency, and reproducibility.

The terms “pipelines” and “workflows” are used somewhat interchangeably, although conventionally a pipeline is a narrower description of a file transformation process, while a workflow can refer to any set of processes that encompass multiple goals and sometimes manual steps, even those in the laboratory.

Several disparate and changing elements – tools, pipeline frameworks, data and metadata standards, programming languages, notebooks, computational infrastructure – make up the “workflow landscape”. In particular three developments – the cloud, containerization, and semantic encoding – are rapidly changing how modern bioinformatic workflows are created. These are continue to evolve with the aim of improved scalability, accessibility, transparency, and perhaps most importantly, reproducibility.

The process of converting existing analyses into workflows catalyzes the modularization of steps common to many analyses, such that individual steps or entire pipelines can be reused for similar experiments with minimal modifications. A tangentially related aspect of this modularization is the formal introduction of common steps in machine learning workflows such as feature engineering, training, validation, testing, and optimization. The efficient dispatch of processing necessary for high-throughput analysis, processing thousands of samples and millions of data points, is an evolving focus of modern pipeline frameworks, and will be discussed in terms of scalability and computational runtime as well as in development time and handholding.

Scalability in workflows has been tackled from two sides of the high performance computing – one side the maturing use of cloud platforms to quickly deploy jobs on rented on-demand virtualized computing infrastructure using a variety of traditional cluster batch queueing systems, cloud-aware batch schedulers, container management systems, and serverless computing services. The other side is the development of distributed computing environments for big data, such as Apache Spark which can leverage large clusters while presenting a seamless interactive programming session. While these efforts are not incompatible, cloud computing has engaged the evolution of workflows and workflow frameworks more than Spark by allowing existing executables and algorithms to scale without reimplementing, basically running unaware of the workflow that is driving them.

The primary goal of a workflow, regardless of its formal structure, is “narrow-sense reproducibility” – the automated replication of results given a set of input data and parameters. A secondary or auxiliary goal is broad-sense reproducibility (aka “reviewable research,” Stodden *et al.*, 2013), the transparent communication of a scientific experiment or train of thought which encompasses both the intent of an analysis and its underlying source code. Workflows connect the versioning, dependencies, and literate programming, to reproducibly produce results from data and metadata (Sandve *et al.*, 2013) in the narrow repeatable sense. Workflows will also play larger roles in broader senses of reproducibility – confirmable, generalizable, reviewable, auditable, verifiable and validatable – though these goals are dependent on semantic representations of workflows and integration with the larger scientific process, namely publishing (Hrynaskiewicz *et al.*, 2014). Ideally, future article reviewers will swap out both individual tools and data in a workflow attached to a submission to test its underlying scientific validity.

In the information sciences community, the concept of reproducibility is often discussed in terms of “provenance” (Kanwal *et al.*, 2017; Gil *et al.*, 2007). Provenance is defined by the W3C as “a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing” (Moreau *et al.*, 2013). In workflow circles provenance refers to the origins of both input data, tools, results, and intermediates.

An interesting review by Cohen-Boulakia *et al.* (2017) examines workflow systems in the context of *reproducibility-friendliness*, using real-world use cases as a means to explore the needs of researchers in terms of reproducibility, and the strengths and limitations of pipeline frameworks. The workflow specification, provenance tracking, and dependency management are compared in terms of scientific demands related to reproducibility. These include rapid integration of new tools, tool version tracking, workflow execution provenance trace, and workflow segment reuse. This group also distinguishes “levels of reproducibility” from

the increasingly broad senses of repeat, replicate, and reproduce, each introducing progressively more variation from the initial *in silico* experiment. The article concludes that more tools are needed to assist in semantic annotation of workflows and to take advantage of the provenance being generated in terms of both validation and discovery; many workflows are not modular enough to work in an interoperable manner (accepting intermediates from other workflows, for example). Other areas highlighted as having potential for improvement are integration with publications, and the development of more user-friendly and abstracted workbenches.

Accessibility and shareability of workflows is key to realizing the vision of enabling “open analysis” or “open research” – transmission of an entire workflow to another individual to be evaluated or reused without significant technical hurdles. This goes hand-in-hand with the goal of analytical democratization, allowing biologists and other non-specialists to build and use pipelines with their own data. In the life sciences, many of these goals have been formalized in manifests such as the FAIR principles (Wilkinson *et al.*, 2016) and are being actively pursued by various international consortiums such as GA4GH (The Global Alliance for Genomics and Health*, 2016), as well as government-funded efforts such as the NIH’s Big Data to Knowledge (BD2K) program, and NCI’s Cancer Genomics Cloud (Davis-Dusenbery, 2015). These initiatives and other like them seek to build shared workflow platforms on the massive data coordination that has taken place over the past two decades.

Systems and Applications

Toolkits

Bioinformatics tools that cater to novel experimental research methods tend to be composed of simple tools aggregated into “toolkits” with common data formats and syntax conventions that focus on a domain. To maintain flexibility, tools within toolkits can be used “a la carte”, enabling tools from other toolkits or user-developed tools to be swapped in, and can take on a variety of software language implementations. Popular toolkits include the Genomic Analysis Toolkit for variant analysis (McKenna *et al.*, 2010) as of version 3.6 composed mostly of Java .jar files, the Tuxedo suite, comprised of tools including Bowtie, TopHat, and Cufflinks, compiled executables for DNA-seq and RNA-Seq analysis (Trapnell *et al.*, 2012). Bioconductor, which is currently composed of 1383 R packages, supports many interrelated tools organized by “Views” and “Common Workflows” such as Differential Expression and Flow Cytometry, with an emphasis on recipes and use-case driven documentation (Gentleman *et al.*, 2004). Bioconductor is supported by a backbone of key classes and methods such as GenomicRanges, SummarizedExperiment (Huber *et al.*, 2015), and VariantAnnotation (Obenchain *et al.*, 2014).

Ready-made pipelines

Ready-made pipelines such as bcbio-nextgen (Guimera, 2012), omicspipe (Fisch *et al.*, 2015) and Churchill (Kelly *et al.*, 2015) are tightly bound collections of existing third-party tools. Such pipelines are typically driven by large configuration files and are designed to provide optimal settings and high performance for typical workflows, however they are not extensible in the sense that user-developed tools cannot be easily integrated which can limit their application in experimental settings. Some ready-made pipelines, such as recount2 (Collado-Torres *et al.*, 2017), are composed of purely of libraries used within programming environments such as R, which offer the advantage of remaining in an interactive session for the duration of a pipeline, although this limits scalability.

Some pipelines are built on existing pipeline frameworks (discussed below). These include Knime4Bio (Lindenbaum *et al.*, 2011) and KNIME4NGS (Hastreiter *et al.*, 2017), based on the Konstanz Information Miner (KNIME). META-Pipe (Robertson *et al.*, 2016) and deepTools (Ramírez *et al.*, 2014) are popular toolkits built for use within Galaxy server workbench.

Plummer and Twin (2015) compared three popular ready-made pipelines for processing 16s rRNA gene sequences to study bacterial phylogeny and taxonomy – MG-RAST (Meyer *et al.*, 2008), QIIME (Caporaso *et al.*, 2010), and Mothur (Schloss *et al.*, 2009). This study found large differences in usability and performance, as well as tertiary measures such as user support, visualizations, and overall robustness in terms of being able to stably accommodate a range of experimental scenarios and varying input quality as encountered in the field. Robustness is arguably the primary criterion that determines the longevity of a pipeline in the scientific community, and together with reproducibility are the two most important abstract properties for judging pipeline quality (Sztromwasser, 2014).

Pipeline frameworks

Pipeline frameworks, also known as workflow management systems (WMS), or scientific workflow management systems (SWfMS), provide structured means of expressing file transformations that are common to bioinformatics analyses. These provide several key advantages over their predecessors in terms of robustness, reusability, extensibility and scalability (Goble and De Roure, 2009).

Until recently, most basic bioinformatic pipelines were developed using scripting languages, either in a Unix shell or a high-level scripting language such as Perl or Python. In such a scheme, variables are used to represent files or parameters and loops to perform iterative processing. An alternative to used the build automation tool, Make (Stallman and McGrath, 2002, described below), which provides a means of representing a dependency graph in terms of filename suffix wildcard recipes. Parameterization and dependency graphs, the descendants of scripts and Makefiles, have been universal components of all modern pipeline frameworks.

The dependency graphs described above, more precisely understood as directed acyclic graphs (DAGs), can represent both serial and parallel tasks and file dependency order. DAGs serve as the conceptual logic underpinning all workflows and are useful

as a visual representation as well as a functional map. The means by which DAGs are created distinguishes modern bioinformatic pipeline frameworks. DAGs are created either by the workflow itself, inferred from filename extensions or explicit inline task dependencies, or by the user, who may use a formal markup such as XML, a domain-specific shorthand, a full-featured language, or using a graphical user interface. Regardless of their origins, DAGs enable the internal workflow engine to determine which unfinished dependencies are needed to complete a task and which can be run in parallel.

A review by Leipzig categorizes frameworks using three criteria: using an implicit or explicit syntax, using a configuration, convention or class-based design paradigm and offering a command line or workbench interface (Leipzig, 2016). While these categories are broad and do not address myriad novel features developed by individual projects, the taxonomy is useful for understanding the major types of pipeline frameworks.

Implicit and explicit DSLs

Implicit frameworks can trace their lineage to the Make, a compile build tool designed in the 1970s as a domain-specific language. Make offers a set syntax consisting of symbols which relate files with different filename suffixes into implicit wildcard pattern rules, which define how these files can be transformed into one another. As such, Make can produce any file for which a rule chain exists. Make is highly dependent on file modification dates in order to assess necessary tasks. Crucial to the success of Make and other implicit frameworks is *reentrancy* – the ability to stop and resume the progress of a pipeline at any stage without penalty. Files that are already completed are not remade, but new samples introduced as input are processed. Reentrancy is indispensable during pipeline development, when errors are frequent, but is also useful for quickly recovering from unexpected events in production, as well as bypassing redundant tasks as projects that grow both in scale and scope. Implicit frameworks such as Snakemake (Köster and Rahmann, 2012) and Nextflow (Di Tommaso et al., 2017) augment the Make approach with support for full featured programming languages, Python and Groovy respectively.

Snakemake remains true to the conventions introduced by Make, in that pipelines are composed of explicit rules with target inputs (usually lists of files), and implicit wildcard rules which describe file transformations. While Python functions can be used as inputs, allowing databases or other actors to be sources of filenames, the general approach is file-centric. This reliance on file naming can necessitate a number of lookup functions when there is no clear suffix chain to associate inputs and outputs, as may happen when processing files named with universally unique identifiers (UUIDs). Nextflow introduces *channels* which hold file types shared between tasks, obviating the need to carefully tag intermediates with complex file suffix name schemes. Inputs and outputs in Nextflow are typed values, offering greater flexibility in terms of holding results in memory while still retaining reentrancy through caching.

Rather than rely on inputs, outputs, and generalized rules, *explicit frameworks* link *tasks* in a set order specified by the user. This approach is more intuitive to some whose conception of a pipeline does not deviate from a set script, but limits the flexibility of the pipeline to produce target files not initially anticipated by the user.

Configuration frameworks

We consider domain-specific languages (DSL) to use a “conventional” design paradigm, whereby the DAG is formed by following syntax conventions formed by the DSL and the underlying language, with an aim to reduce the overall typing load on the developer. A more structured approach is to demand a configuration file formatted in a markup language such as XML or JSON, with little or no embedded code.

Configuration frameworks are always explicit. In addition, they also strongly distinguish workflows from provenance. The respective terms assigned are “workflow templates” and “workflow runs” – a workflow run having all variables assigned and serving as a log of steps that are performed in the creation of an output.

Of course, workflow configurations can themselves be generated by scripts, various wizards both graphical and command line, or APIs, but they must be validated and loaded using strict parsers. This leaves little room for inline on-the-fly conditional logic based on input itself, often called “dynamic branching”, for example skipping corrupt files or switching sensitivity/specificity parameters.

Established configuration frameworks such as Pegasus (Deelman et al., 2005), are often tightly coupled to an existing execution engine, such as Condor (Thain et al., 2005), which enables efficient job scheduling but at the expense of portability. Modern configuration frameworks, such as Cromwell from the Broad Institute (Voss et al., 2017), support a number of execution engines and schedulers, both for in-house clusters and the cloud.

Workbenches

Workbenches typically offer a graphical canvas that allows users to connect nodes (representing tasks) with edges (representing parallel or serial data flow). Open source server workbenches can be installed locally or in the cloud. With over 1500 citations, Galaxy (Goecks et al., 2010) is arguably the most popular bioinformatics workbench. Galaxy is designed to be run on a local cluster or in the cloud. Knime (Berthold et al., 2009), Taverna (Wolstencroft et al., 2013), Vistrails (Callahan et al., 2006), and Kepler (Ludäscher et al., 2006) can be installed on a desktop computer but offer server and cloud-based execution. Commercial cloud-based workbenches such as DNANexus, SevenBridges, and Illumina’s BaseSpace are remotely managed, Software-as-a-Service (SaaS) installations, which provide private accounts and abstract the provisioning of underlying cloud platforms for usability. Developments in object storage such as Amazon Web Services (AWS) Simple Storage Service (S3) buckets have allowed researchers some autonomy in how their data is owned and managed.

Some *collaborative analysis platforms* combine the mechanics of canvas workbenches with an existing suite of reference data, highly customized reports, and comparative samples, typically focused to maximize relevant results in a specific domain. One

example is Cavatica, a joint project of Seven Bridges Genomics and Children's Hospital of Philadelphia to accelerate pediatric cancer research by encouraging data sharing with the open use of standards-compliant tools and workflows and visualizations developed for the cBioPortal (Gao *et al.*, 2013).

For high throughput analyses there is usually a need to automate data management away from the graphical user interface using local command line dispatch. This is normally supported using APIs. Projects such as BioBlend (Sloggett *et al.*, 2013) provide tools to remotely manage CloudMan (Afgan *et al.*, 2010) cloud instances which host Galaxy workflows. Other platforms such as Arvados (Amstutz *et al.*, 2016) and Cyverse (Devisetty *et al.*, 2016) provide support for API-driven workflows and a large data collection that can be browsed using dashboard tools.

Common workflow language

Because of the highly linked nature of these operations, workbenches require strongly typed tool definitions and workflow descriptions to perform in a robust manner. The sheer number of bioinformatic tools and frameworks warranted a level of standardization in order to reduce redundancy and enhance portability of tools and workflows. The proliferation of pipeline frameworks, each with its own idiosyncratic syntax for defining tool inputs, outputs, and parameters, presents frustrations for both users and developers. An effort spearheaded by developers at commercial cloud workbench platforms has developed a standard format for defining both workflows and tool definitions that can be used by any configuration-based framework. The Common Workflow Language (CWL) is a multi-vendor consortium that has developed a configuration-based standard for interfacing with command-line software, describing workflows, and adding semantic annotation to these elements.

Two important technologies being used by CWL are JSON-LD, a linked human-readable data format that provides semantic encoding to elements such as tools, inputs, and outputs, and Docker (see below), a containerization layer that provides isolation and a consistent environment for tools. This effort has spawned an entirely new schema validation system supporting JSON-LD (SALAD), the cwl-viewer for visualization, and a number of APIs for producing CWL programmatically (Figs 1 and 2).

Class-based frameworks

Class-based frameworks dispense with the symbolism of domain-specific language clauses and rely on pure-language annotators and methods to introduce workflow operations into existing code.

A number of class-based frameworks have been developed in e-commerce settings, and these tend to provide better support than scientific pipeline frameworks for event-triggered executions, live monitoring, and database operations common to real-time applications. On such framework, Luigi, is similar in spirit to implicit DSLs, but its syntax is implemented as Python classes and methods. Airflow is a pure-python DAG generator – its implementation is closer to a configuration file API than a means to annotate existing code.

Toil is a pure-Python framework developed to take full advantage of cloud infrastructure, featuring a “pluggable backend APIs for machine provisioning, job scheduling and file management” to enable execution on heterogeneous computing environments and sophisticated “caching and streaming” to reduce file system contention (Vivian *et al.*, 2017). Toil received a great deal of attention after a group at UCSF reported it the framework to process an RNA-Seq run of 200,000 samples performed over 90 h on Amazon Web Services, utilizing roughly 32,000 compute cores. The group was able to reduce the cost of analysis by using a combination of Toil optimizations and kmer-based transcript quantification tools. While Toil can ingest CWL, it also includes a number of methods to leverage the Spark-based ADAM toolset developed by the Big Data Genomics (Paten *et al.*, 2015).

Future Directions

Workflows do not exist in a vacuum but are an integral part of the scientific publishing cycle. A major driving factor in workflow standards and innovation has been responding to developments in the cloud, containerization, notebooks, and the semantic web.

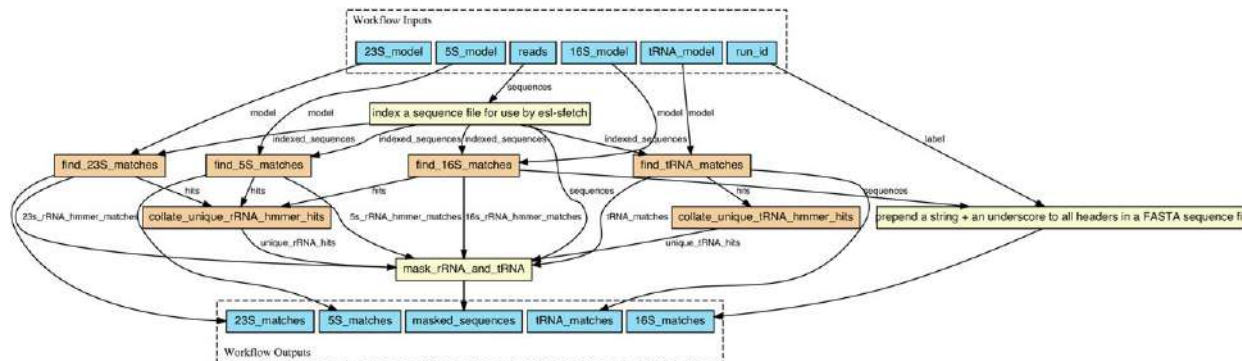


Fig. 1 A workflow using rRNA selector as viewed through the cwl-viewer. Reproduced from Robinson, M., Soiland-Reyes, S., Crusoe, M.R., Goble, C., 2017. CWL viewer: The common workflow language viewer. F1000Research 6, 1075. Available at: <http://dx.doi.org/10.7490/f1000research.1114375.1> (accessed 25.06.17); Lee, J.-H., Yi, H., Chun, J., 2011. rRNASelector: A computer program for selecting ribosomal RNA encoding sequences from metagenomic and metatranscriptomic shotgun libraries. Journal of Microbiology 49 (4), 689–691.

```

inputs:
  reads:
    type: File
    format: edam:format_1929 # FASTA
...
outputs:
  16S_matches:
    type: File
    outputSource: prepend_header_for_QIIME/16S_sequences
...
index_reads:
  run: ../tools/esl-sfetch-index.cwl
  in:
    sequences: reads
  out: [ sequences_with_index ]
...
$namespaces:
  edam: http://edamontology.org/
  s: http://schema.org/
$schemas:
  - http://edamontology.org/EDAM\_1.16.owl
  - https://schema.org/docs/schema\_org\_rdfa.html

```

Fig. 2 Snippets of the CWL code show input, output semantically-defined inputs and outputs, and a step in the workflow. Reproduced from Crusoe, M., 2017. Available at: <https://github.com/ProteinsWebTeam/ebi-metagenomics-cwl/blob/master/workflows/rna-selector.cwl>.

The cloud

While the cloud can be considered a rented timeshare of local computing infrastructure, the cloud has introduced many new opportunities for scalability and the sharing of workflows than was previously possible. Several technologies that are relatively rare in academic computing environments have also become de rigueur among workflows, namely object storage. Workflows must accommodate staging files from object store to local volumes. This would be highly impractical without frameworks.

Mixed-use schedulers

Apache Mesos and Apache YARN are cluster management platforms that are designed to conduct job scheduling across distributed heterogeneous networks. While these projects have different origins, they both compete with traditional batch engines such as SGE, LSF, and SLURM.

Cloud-native batch execution engines

Proprietary cloud execution engines such as Microsoft's Azure Batch and Amazon's AWS Batch assume job scheduling and auto-scaling as an integrated feature of the Platform-As-A-Service (PaaS) model. Because they are designed from the ground up to work in the cloud, these engines are equipped to better profile existing CPU, memory, and cost-structures to complete jobs than batch schedulers designed for traditional grids, and they require less setup time. As of this writing Nextflow has already introduced support for AWS Batch, which suggests widespread support for cloud-native engines is likely in the near future.

Serverless computing

Some cloud providers offer a class of services aimed at users who wish to dispense with all of the setup and overhead associated with servers. Serverless platforms such as AWS Lambda, Microsoft Azure Functions, and Google Cloud Functions allow simple code to run without the need to configure and deploy servers. While this "microservices" approach is quite familiar to software developers in e-commerce, rapid adoption in bioinformatics will likely coalesce around specific high-load query cases. Because serverless functions lack the basic process exit status logic of traditional DSL-based frameworks, process status checkpoints need to be manually built to work with file-centric sequence analysis programs.

Dependency management and containerization

The installation and communication of a corpus of versioned compatible tools is a major point of frustration for developers and a concern for reproducible research. Because workflows consist of many tools chained together, the management and harmony of these dependencies is of paramount importance to workflow sharing. Two developments, universal package management and containerization, have addressed this problem using different but potentially compatible approaches.

Universal package management

A package manager that originated from Python, Conda provides an elegant solution by offering a number of dependencies – including Python and R packages and compiled binary executables – in a single versioned requirements file. Conda packages specify other Conda dependencies such as linked libraries, cross-platform binaries that are distributed within the Conda ecosystem through channels. One such channel is Bioconda, which contains over 2700 bioinformatics-specific packages submitted by a wide group of developers and validated using an automated continuous testing and integration scheme.

Containerization

Docker is an open source project that provides a lightweight and configurable operating system virtualization layer called a container. Containers allow software with exact dependencies and requirements to run isolated from the external underlying operating system. This approach ensures software operates consistently and identically on different operating systems, both locally and in the cloud. By isolating programs from each other and the OS, Docker guarantees the replicable execution of scientific software within a workflow regardless of conflicting dependencies or versions with other software in that workflow or software installed by the user (Boettiger, 2015), with no appreciable loss of performance (Di Tommaso *et al.*, 2015).

The Docker architecture utilizes configured scripts called Dockerfiles that are processed to build “images”, ready-to-run files used to instantiate containers. Dockerfiles can themselves be specified with base images, enabling a certain level of templating. The process of building images is far more time consuming than instantiating containers, so it is common to cache these images in a service such as Quay (quay.io). The proliferation of bioinformatics Docker containers has been aided by repositories such as Dockstore (O'Connor *et al.*, 2017) and bioShaDock (Moreews *et al.*, 2015).

In terms of distributing the underlying software behind workflows, dependency management and containers have significant overlap in terms of the goal of providing consistent software across platforms. Conda attempts to do this by providing compiled software and a compatible ecosystem within custom environments. While Conda provides environments and cross-platform dependency management, it does not provide process isolation and depends on a compatible cohort of programs to coexist within an environment, often making exact version specifications impossible. Docker isolates individual programs to eliminate conflict from each other and the underlying operating system. Both approaches are less than ideal for scientific computing, but do provide working solutions, especially when used synergistically. Dockerfiles consist mostly of dependency installation commands and a few exposed ports and paths. Some Docker images use Conda as their primary dependency installation step, and there are now projects, such as Muled, a project within the BioContainers repository (da Veiga Leprevost *et al.*, 2017), that auto-create Docker images based on Conda packages.

The potential for Docker containers to provide a missing technical link in terms of reproducible computational research in bioinformatics has been explored by projects including evaluation of *de novo* assemblers (Belmann *et al.*, 2015). Beaulieu-Jones *et al.* introduced the application of Docker containers using the Drone configuration pipeline engine, allowing the automated evaluation of code changes into the analytical pipeline, which the authors call “continuous analysis” (Beaulieu-Jones and Greene, 2017). While containers can be run in a similar fashion to any command-line software, some pipeline frameworks, such as Nextflow and the CWL-enabled workbenches, offer first-class support for Docker containers.

Containers present an atomic unit of doing work. Docker Swarm and Kubernetes are frameworks that orchestrate hundreds of Docker containers, such as Docker Swarm and Kubernetes, and also offer some level of queueing and resource management, but they compete in a crowded field of existing schedulers.

Workflow sharing

Provided a data set that fits an existing approach, workbenches provide a smooth path for workflow reuse. myExperiment collates workflows from Taverna, Galaxy, and other frameworks into a searchable portal (Goble *et al.*, 2010).

Research Objects aims to bundle the entire stack of data and analysis into a cohesive stack, with a research object bundle (see “Relevant Website” section) and associated workflow ontology (see “Relevant Website” section) which together with a manifest allow researchers to understand an entire analysis from raw data to results (Bechhofer *et al.*, 2010). The motivations and rationale for packaging research objects is described in the highly-cited “Why Linked Data is Not Enough for Scientists” (Bechhofer *et al.*, 2013), which explains why bundling linked data with workflows, results, and publications is preferable to linked data without context.

Notebooks

Notebooks are web-based interactive programming environments. Like Sweave (Leisch, 2002) or knitr (Xie, 2014) literate reports, notebooks support a mix of code and markup, but they are a significant advance in that they allow code to be evaluated in live chunks which retain state. This allows for an easier transition from the type of organic “conversations” typical of analyses in a statistical environment such as R to an automated reproducible report. However, the transition from data preparation steps to pipeline to a notebook is not graceful and currently involves handoffs in the form of intermediate files to store metadata.

The Jupyter notebook is a polyglot derivation of iPython (Perez and Granger, 2007) which supports over 40 languages. The JSON-based .ipynb format recognized by Jupyter been the de facto standard for distributing analyses in the data science community and is gaining acceptance within bioinformatics for rapid dissemination of reproducible analyses (Wang and Ma’ayan, 2016). A static viewer, nbviewer, is available for reading these files in a non-interactive fashion, but few canned solutions exist for instantiating live sessions. A proof of concept demonstrating live notebooks was published in Nature (Shen, 2014) using tmpnb, a Docker-driven temporary notebook engine. An effort from the Freeman neuroscience lab, mybinder.org, accepts Conda requirements files for dependency management and leverages Docker, the Kubernetes container orchestration tool, and Google Cloud Platform to host live repository-driven notebooks.

While they provide an interactive analysis session, notebooks themselves are not integrated development environments (IDEs), and most lack features such as code completion and debugging that IDEs provide. NextflowWorkbench is a project that augments authoring Nextflow pipelines (Kurs *et al.*, 2016) and Dockerfiles using supported JetBrains IDEs.

Although they lack reentrancy frameworks and native support for parallelization, in many ways notebooks are more appropriate for the final steps of analyses than typical workflows frameworks, especially for analysis that rely on statistical tests conducted in

scripting languages or statistical environments. Some notebooks such as Beaker are polyglot – supporting multiple languages simultaneously. Collaborative notebook servers such as CoCalc (previous SageMathCloud), aim to provide live editing for highly collaborative research, but self-hosted collaborative notebooks are still in development. Most cloud workbenches do not yet offer significant support for notebooks and integrating notebooks and workflows is a major area of research (Grüning *et al.*, 2017).

Semantically encoded workflows

Extending semantic encoding to workflows has long been envisioned as a way to make them understandable, discoverable, and more reproducible. For example, semantic encoding may allow researchers to find workflows that use a certain reference data set, a combination of similar tools, or even suggest a biological interaction. All of these scenarios are dependent on semantically encoded linked data. They are also highly dependent on linking data between all the various components and stages of a workflow – input data, tools, pipelines, statistical reports and publication. Fortunately, the theoretical (ontologies) and technical (RDF/OWL) foundations for this endeavor have been established by efforts to create the Semantic Web.

An ontology is a set of terms that provides meaning to data. It builds on controlled vocabularies, by defining formal relationships between concepts. Ontologies have been used extensively to describe biological concepts. Building on ontologies is the concept of linked data, which leverages remote but unambiguous addresses to point to items of data. The combinations of ontologies and linked data forms the theoretical foundation for the semantic web, a set of technologies that allow computers to gain insight into the meaning of data, giving different services an ability to both locate and relate linked data using common web Uniform Resource Identifiers (URIs). The underpinning of the semantic web is the Resource Description Framework (RDF), which allows entities to be related to each other and to attributes using triplets in the form of entity: attribute:value. The relationships themselves are defined by a standard called Web Ontology Language (OWL), in which domain-specific ontologies, including those specific to science, can be expressed.

While RDF and OWL were designed to give computers the ability to transverse the web intelligently, these technologies have been used extensively to describe biological systems in implementations such as Uniprot (Jain *et al.*, 2009), EBI (Jupp *et al.*, 2014), DisGeNet (Queralt-Rosinach *et al.*, 2016), KEGG (Kanehisa and Goto, 2000), Reactome (Joshi-Tope *et al.*, 2005), OpenLifeData (González *et al.*, 2014) and others. A popular compendium of linked biological data resources is Bio2RDF (Callahan *et al.*, 2013).

Linked data can be queried using the SPARQL language. Cytoscape and ChEMBL were early proponents of this strategy. Providing SPARQL endpoints has become especially desirable for complex data such as that hosted by the Cancer Genomics Cloud. Tools such as SADI have been extended to Galaxy to enable complex data queries within workflows (Aranguren *et al.*, 2014). For queries on workflows themselves, Taverna and myExperiment support SPARQL endpoints, as well as the Research Object Hub. This means, in theory, one can use a single query for workflows that rely on certain data, use certain tools, and were published within a certain time period by certain analysts. It is also been posited that semantic representations can be used a high-level standard to translate workflows into other frameworks (Garijo *et al.*, 2014).

A semantic workflow platform, Workflow Instance Generation and Specialization (WINGS) is a project that allows semantically encoded workflows, which are then executed using existing pipeline frameworks (Gil *et al.*, 2011). A key feature of WINGS is the ability to semantically enforce outputs, allowing technical and scientific-based sanity checks to be integrated into workflows.

WINGS is built on existing semantic web standards and workflow-specific ontologies, including PROV, a generic semantic model for describing provenance, consisting of data models (PROV-DM) and ontology (PROV-O) capable of representing typical data lineage (Missier *et al.*, 2013). The Open Provenance Model (OPM) extends PROV for use in workflows. A metadata ontology called Open Provenance Model for Workflows (OPMW) extends OPM, PROV and P-Plan in order to encompass a large range of actors, intents, and concepts. Motivations include the ability to enforce “analytical validity” (Zheng *et al.*, 2015) to encode expert knowledge within workflows.

In configuration frameworks such as workbenches there is a clear distinction between workflow templates and workflow runs, and these are reflected respectively in the ontologies wfdesc and wfprov developed by the wf4ever consortium (Corcho *et al.*, 2012), supported as exports from Taverna via TavernaPROV (Soiland-Reyes, 2016) and CWL-based frameworks, and implemented in Research Objects. CWL itself offers a level of semantic encoding using JSON-LD which should be able to leverage existing ontologies, although existing examples have been mostly focused on file formats.

An important ontology describing the components of workflows is provided by EDAM (Ison *et al.*, 2013), an OWL ontology that can describe a diverse set of tools, datasets, and operations common to bioinformatic workflows. At the analysis level, statistical ontologies such as OBCS (Zheng *et al.*, 2016) may be useful for promoting reproducibility and discovery.

Challenges and Opportunities for Pipelines and Workflows

Unruly provenance

Because workflow performance is highly dependent on both parameters and high-performance computing configurations, this should ideally be encoded into provenance metadata, and some attempts have been made to standardize this in a vendor-agnostic fashion (Santana-Perez *et al.*, 2014). Despite extensive work toward the semantic markup of workflows, simply attaching workflow runs to data may not suffice to provide a transparent provenance for data contained within results. Efforts such as LabelFlow (Alper *et al.*, 2014) and PoeM (Gaignard *et al.*, 2016) aim at providing higher-level context by identifying common usage motifs and combining domain-specific ontologies. Another problem with complex provenance is the crowding of workflow visualizations, making them incomprehensible, suggesting some lightweight annotations are needed to distinguish major and minor steps (Missier *et al.*, 2008). Binding experimental metadata to results is an under-examined step in reproducible computational research, and has been identified as a key element in the sustainability of highly variable data such as metagenomics where sample

preparation and *in silico* choices have a large impact on results (Hoopen *et al.*, 2017). It is possible blockchain technology may also be useful in this area (Neisse *et al.*, 2017).

Workflow decay

The development of large multi-institutional data repositories that characterize “big science” and remote web services that support both remote data usage and the vision of “bringing the tools to the data” make the cloud an appealing replacement for local computing resources (Stein, 2010). This dependence on data and services hosted by others, however, introduces the threat of “workflow decay” (De Roure *et al.*, 2011) that requires extensive provenance tracking to freeze inputs and tools in order to ensure reproducibility at a later date.

Integration of the web

Due to the increase in the size of datasets combined with limitations of networks, web browsers, and web application frameworks, web applications as the main interface for high-throughput analyses have been less than ideal. This prompted Nucleic Acids Research to add a “Stand-Alone Programs for High-Throughput Data Analysis” to their popular web-server issue. However, there is a large gap between pipelines and notebooks, which are the primary tools used by bioinformaticians and GUI-driven interactive analysis which allows researchers to explore data in real-time. The latter has splintered into smaller “single-page” applications, which have been made popular due to easy-to-use web application frameworks such as RShiny which have spawned many convenience utilities (Mattielo *et al.*, 2016; Zhang and Taylor, 2017; Fouillet *et al.*, 2017). The interactive nature of these sites often demand technologies such as AJAX which dispense with the advantages of stable URLs and linked data enjoyed by earlier generations of bioinformatics web applications. Web applications may also disrupt reproducibility by introducing manual and unrecorded steps into an analysis.

Cloud-based workbenches offer some solutions to this problem, because of their scalability and built-in faculties for queues and the ability to spawn runs locally using APIs. But these tools tend to drop off at the point of target file generation, and even report generation and sharing is not a strong suit for workbenches. There is clearly a great deal to be done to bridge the gap from pipelines to open, linked analyses.

A related challenge, in particular for cloud-based commercial workbenches, is providing support for highly customized or interactive web reports as reporting mechanisms. Web-based QC reporting tools such as MultiQC (Ewels *et al.*, 2016) and literature programming outputs such as knitr/R Markdown pose challenges both in terms of security and navigation within hosted infrastructure.

Notebook and Spark integration

Integrating notebooks into a workflow context, so that it can access the workflow and the notebook components can be reused, parameterized, and queried, remains an open challenge. Ideally statistical reports are tied intimately to their respective workflows (i.e., metadata-driven analysis), so changes such as addition of samples are done in one sample table, and these changes are automatically propagated through the pipeline to a notebook or static report. The approach to this propagation differs significantly between DSLs and configuration-based frameworks, the former being more likely to use variables and functions to drive the pipeline and the report. Breaking notebook sessions into file-centric state engines of DSLs, much less the verbose markup of workbenches and other configuration frameworks, completely destroys their powerful interactive and exploratory nature. This approach is precisely the opposite direction of that taken by the Spark community, which intends to bring executable components into distributed data structures that can be used in live notebook sessions. The YesWorkflow project allow developers to use ontology-based lightweight tags to annotate popular scripting languages with provenance metadata (McPhillips *et al.*, 2015). A non-coincidentally named project, noWorkflow, reconstructs provenance through profiling Python code (Murta *et al.*, 2014). These can be used together to generate semi-automated annotations (Belhajjame *et al.*, 2016; Dey *et al.*, 2015). A similar project, Data Narratives (Gil and Garijo, 2017), offers human-readable annotations of workflows.

Apache Spark is a fast, in-memory data processing engine, released as an Scala API but with support for a number of scripting languages. Spark provides a consistent API to perform data analysis without leaving a notebook session and hides many of the inner workings of the distributed memory. The basic abstraction in Spark that allows distributed data to be manipulated as though it were centralized is the Resilient Distributed Dataset (RDD). RDDs themselves implement a DAG to describe changes and provenance, including their dependence on other RDDs. Therefore, Spark itself subsumes some of the functions of pipeline frameworks, provided an entire bioinformatics analysis can be implemented in Spark. Because Spark is not itself a DSL or configured DAG evaluated at runtime, the reentrancy provided by Spark through caching RDDs is not automatic but must be anticipated by the user. Though Spark can certainly behave in concert with workflows – there are extensive example of ADAM in Toil workflows – these are largely still file-centric in nature, using Spark as a distributed replacement for an compiled executable. Spark jobs are deployed on Apache YARN, and this is supported by Cromwell and Toil. However it does not appear there is yet a framework that leverages RDDs as in-memory inputs and outputs, and supports files within the same workflow context.

Limitations of technologies borrowed from elsewhere

Many technologies for bioinformatic pipelines and workflows have been adopted from other areas of scientific computing encompassing the physical and mathematical sciences, but also computer science, data science, information science, e-commerce, finance, and media. As a result, the fit of these tools and infrastructure is often not ideal for bioinformatics. Containers are an excellent example of this phenomenon.

Docker itself offers no dependency management and its approach is often contrary to the natural flow of an analysis – Docker does not enable the easy configured addition of new R packages within an existing environment, for example. Docker was originally designed to address concerns of rapidly deploying e-commerce apps, which rely heavily on databases and web application servers. These web installations do not resemble a typical executable file-transformation pipeline common to scientific computing, and so frameworks for multi-container installations (e.g., Docker Compose) or orchestration (e.g., Kubernetes) are not geared for the start/stop nature of bioinformatic workflows. Docker is also not particularly good at accepting runtime parameters. The Docker daemon requires administrative (either sudo or Docker-group) to run containers, making Docker unfeasible for some shared compute cluster environments (Silver, 2017). A related project developed at Lawrence Berkeley National Laboratory, Singularity, allows more user-centric computing environments composed of many containers to be created and distributed without these administrative privileges.

Continued need for incentives

Despite the interest in semantic encoding from workflow developers, widespread use of semantic markup on workflows remain elusive. There is also no common API for biological file format validation, which would be low-hanging fruit for encouraging semantically defined outputs. Although considerable effort has been put into WINGS, it is built to leverage Pegasus as an execution engine. However, Pegasus is not yet CWL compliant, and the syntax used for semantic enforcement, Apache Jena, is quite verbose. The move toward standardization suggests a good candidate semantic integration moving forward is through the Common Workflow Language.

Almost 3000 workflows are available in myExperiment, although submissions have lagged in the NGS era (only 10 results were obtained for the search term “RNA-Seq”, for instance). It seems likely the introduction of CWL-based workflows will re-energize this repository, but a reinvention of the publishing structure to encourage open workflows as an essential prerequisite to the peer-review process would be a more effective catalyst.

Closing Remarks

Pipelines and workflows are the primary river between scientific data and publications. With the growth of high-throughput data, this river has been made wider by the standardization of workflow languages, deeper by the recognition for the semantic encoding, and is flowing faster thanks to technical advances in cloud computing and containerization. One of the main challenges to future workflow development is keeping the key components of *in silico* scientific processes – data, tools, notebooks – fluid enough to support groundbreaking work and advances from the greater computing community while maintaining standards for reproducible research.

See also: Cloud-Based Bioinformatics Platforms. Cloud-Based Bioinformatics Tools. Cloud-Based Molecular Modeling Systems. Cloud-Based Molecular Modeling Systems. Computational Pipelines and Workflows in Bioinformatics. Computing for Bioinformatics. Constructing Computational Pipelines. DNA Barcoding: Bioinformatics Workflows for Beginners. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome Annotation: Perspective From Bacterial Genomes. Infrastructure for High-Performance Computing: Grids and Grid Computing. Infrastructures for High-Performance Computing: Cloud Computing. Integrative Analysis of Multi-Omics Data. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. MapReduce in Computational Biology via Hadoop and Spark. Natural Language Processing Approaches in Bioinformatics. Network-Based Analysis for Biological Discovery. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Prediction of Coding and Non-Coding RNA. Profiling the Gut Microbiome: Practice and Potential. Protocol for Protein Structure Modelling. Standards and Models for Biological Data: Common Formats. Vaccine Target Discovery. Whole Genome Sequencing Analysis

References

- Afgan, E., *et al.*, 2010. Galaxy CloudMan: Delivering cloud compute clusters. *BMC Bioinformatics* 11 (Suppl. 12), S4.
- Alper, P., *et al.*, 2014. LabelFlow: Exploiting workflow provenance to surface scientific data provenance. In: *Provenance and Annotation of Data and Processes. Lecture Notes in Computer Science. International Provenance and Annotation Workshop*. Cham: Springer, pp. 84–96.
- Amstutz, P., *et al.*, 2016. Using the common workflow language (CWL) to run portable workflows with Arvados and toil. *F1000Research* 5. Available at: <http://dx.doi.org/10.7490/f1000research.1112523.1> (accessed 21.08.17).
- Aranguren, M.E., González, A.R., Wilkinson, M.D., 2014. Executing SADI services in galaxy. *Journal of Biomedical Semantics* 5 (1), 42.
- Beaulieu-Jones, B.K., Greene, C.S., 2017. Reproducibility of computational workflows is automated using continuous analysis. *Nature Biotechnology* 35 (4), 342–346.
- Bechhofer, S., *et al.*, 2010. Research objects: Towards exchange and reuse of digital knowledge. In: *The Future of the Web for Collaborative Science (FWCS 2010)*. Available at: <https://eprints.soton.ac.uk/268555/> (accessed 10.08.17).
- Bechhofer, S., *et al.*, 2013. Why linked data is not enough for scientists. *Future Generations Computer Systems: FGCS* 29 (2), 599–611.
- Belhajjame, K., *et al.*, 2016. Yin & Yang: Demonstrating complementary provenance from noWorkflow & YesWorkflow. In: *Provenance and Annotation of Data and Processes: Proceedings of the 6th International Provenance and Annotation Workshop, IPAW 2016, McLean, VA, USA, June 7–8, 2016*, Springer, p. 161.
- Belmann, P., *et al.*, 2015. Bioboxes: Standardised containers for interchangeable bioinformatics software. *GigaScience* 4, 47.
- Berthold, M.R., *et al.*, 2009. KNIME – The Konstanz information miner: Version 2.0 and beyond. *SIGKDD Explorations Newsletter* 11 (1), 26–31.
- Boettiger, C., 2015. An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review* 49 (1), 71–79.
- Callahan, A., *et al.*, 2013. Bio2RDF release 2: Improved coverage, interoperability and provenance of life science linked data. *The Semantic Web: Semantics and Big Data. Berlin; Heidelberg: Springer*, pp. 200–212. *Extended Semantic Web Conference*.

- Callahan, S.P., *et al.*, 2006. VisTrails: Visualization meets data management. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, SIGMOD '06. New York, NY: ACM, pp. 745–747.
- Caporaso, J.G., *et al.*, 2010. QIIME allows analysis of high-throughput community sequencing data. *Nature methods* 7 (5), 335–336.
- Cohen-Boulakia, S., *et al.*, 2017. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Future Generations Computer Systems: FGCS*. Available at: <http://dx.doi.org/10.1016/j.future.2017.01.012>.
- Collado-Torres, L., *et al.*, 2017. Reproducible RNA-seq analysis using recount2. *Nature Biotechnology* 35 (4), 319–321.
- Corcho, O., *et al.*, 2012. Workflow-centric research objects: First class citizens in scholarly discourse. In: Proceedings of Workshop on the Semantic Publishing. Proceedings of the 9th Extended Semantic Web Conference Hersonissos. Facultad de Informática (UPM), p. 12.
- da Veiga Leprevost, F., *et al.*, 2017. BioContainers: An open-source and community-driven framework for software standardization. *Bioinformatics* 33 (16), 2580–2582.
- Davis-Dusenbery, B.N., 2015. Petabyte-scale cancer genomics in the cloud. *Cancer Genetics* 208 (6), 360.
- Deelman, E., *et al.*, 2005. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming* 13 (3), 219–237.
- De Roure, D., *et al.*, 2011. Towards the preservation of scientific workflows. In: Proceedings of the 8th International Conference on Preservation of Digital Objects (iPRES 2011). ACM. Available at: <http://www.amiga.iaa.csic.es/FCKeditor/UserFiles/File/wfpreserv.pdf>.
- Devisetty, U.K., *et al.*, 2016. Bringing your tools to CyVerse Discovery Environment using Docker. *F1000Research* 5, 1442.
- Dey, S., *et al.*, 2015. Linking prospective and retrospective provenance in scripts. *Theory and Practice of Provenance (TaPP)*. Available at: <https://www.usenix.org/system/files/tapp15-dey.pdf>.
- Di Tommaso, P., *et al.*, 2015. The impact of Docker containers on the performance of genomic pipelines. *PeerJ* 3, e1273.
- Di Tommaso, P., *et al.*, 2017. Nextflow enables reproducible computational workflows. *Nature Biotechnology* 35 (4), 316–319.
- Ewels, P., *et al.*, 2016. MultiQC: Summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* 32 (19), 3047–3048.
- Fisch, K.M., *et al.*, 2015. Omics pipe: A community-based framework for reproducible multi-omics data analysis. *Bioinformatics* 31 (11), 1724–1728.
- Fouillet, A., *et al.*, 2017. User-friendly Rshiny web applications for supporting syndromic surveillance analysis. *Online Journal of Public Health Informatics* 9 (1), Available at: <http://journals.uic.edu/ojs/index.php/ojphi/article/view/7628> (accessed 22.06.17).
- Gaignard, A., Skaf-Molli, H., Bihouée, A., 2016. From scientific workflow patterns to 5-star linked open data. In: Proceedings of the 8th USENIX Conference on Theory and Practice of Provenance, USENIX Association, pp. 44–48.
- Gao, J., *et al.*, 2013. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Science Signaling* 6 (269), I1.
- Garijo, D., Gil, Y., Corcho, O., 2014. Towards workflow ecosystems through semantic and standard representations. In: Proceedings of the 9th Workshop on Workflows in Support of Large-Scale Science, WORKS '14. Piscataway, NJ: IEEE Press, pp. 94–104.
- Gentleman, R.C., *et al.*, 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology* 5 (10), R80.
- Gil, Y., *et al.*, 2007. Examining the challenges of scientific workflows. *Computer* 40 (12), 24–32.
- Gil, Y., *et al.*, 2011. Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems* 26 (1), 62–72.
- Gil, Y., Garijo, D., 2017. Towards automating data narratives. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces, ACM, pp. 565–576.
- Goble, C., De Roure, D., 2009. The impact of workflow tools on data-centric research. Available at: <https://eprints.soton.ac.uk/267336/1/workflows-submitted.pdf>.
- Goble, C.A., *et al.*, 2010. myExperiment: A repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research* 38 (Web Server Issue), W677–W682.
- Goecks, J., *et al.*, 2010. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology* 11 (8), R86.
- González, A.R., *et al.*, 2014. Automatically exposing OpenLifeData via SADI semantic Web Services. *Journal of Biomedical Semantics* 5, 46.
- Grüning, B.A., *et al.*, 2017. Jupyter and galaxy: Easing entry barriers into complex data analyses for biomedical researchers. *PLOS Computational Biology* 13 (5), e1005425.
- Guimera, R.V., 2012. Bcbio-nextgen: Automated, distributed next-gen sequencing pipeline. *EMBnet. Journal* 17 (B), 30.
- Hastreiter, M., *et al.*, 2017. KNIME4NGS: A comprehensive toolbox for next generation sequencing analysis. *Bioinformatics* 33 (10), 1565–1567.
- Hoopen, P.T., *et al.*, 2017. The metagenomic data life-cycle: Standards and best practices. *GigaScience*. Available at: <https://academic.oup.com/gigascience/article-abstract/doi/10.1093/gigascience/gix047/3869082/The-metagenomic-data-lifecycle-standards-and-best> (accessed 21.06.17).
- Hrynaskiewicz, I., Li, P., Edmunds, S., 2014. Open science and the role of publishers in reproducible research. *Implementing Reproducible Research*. 383–410. ISBN: 1826028603.
- Huber, W., *et al.*, 2015. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature methods* 12 (2), 115–121.
- Ison, J., *et al.*, 2013. EDAM: An ontology of bioinformatics operations, types of data and identifiers, topics and formats. *Bioinformatics* 29 (10), 1325–1332.
- Jain, E., *et al.*, 2009. Infrastructure for the life sciences: Design and implementation of the UniProt website. *BMC Bioinformatics* 10, 136.
- Joshi-Tope, G., *et al.*, 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research* 33 (Database Issue), D428–D432.
- Jupp, S., *et al.*, 2014. The EBI RDF platform: Linked open data for the life sciences. *Bioinformatics* 30 (9), 1338–1339.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Kanwal, S., *et al.*, 2017. Investigating reproducibility and tracking provenance – A genomic workflow case study. *BMC Bioinformatics* 18 (1), 337.
- Kelly, B.J., *et al.*, 2015. Churchill: An ultra-fast, deterministic, highly scalable and balanced parallelization strategy for the discovery of human genetic variation in clinical and population-scale genomics. *Genome Biology* 16 (1), 6.
- Köster, J., Rahmann, S., 2012. Snakemake – A scalable bioinformatics workflow engine. *Bioinformatics* 28 (19), 2520–2522.
- Kurs, J.P., Simi, M., Campagne, F., 2016. NextflowWorkbench: Reproducible and reusable workflows for beginners and experts. *bioRxiv*. 041236. Available at: <http://biorxiv.org/content/early/2016/02/24/041236.abstract> (accessed 25.06.17).
- Leipzig, J., 2016. A review of bioinformatic pipeline frameworks. Briefings in Bioinformatics. Available at: <http://dx.doi.org/10.1093/bib/bbw020>.
- Leisch, F., 2002. Sweave: Dynamic generation of statistical reports using literate data analysis. In: Härdle, P.D.W., Rönz, P.D.B. (Eds.), *Compstat. Heidelberg: Physica-Verlag*, pp. 575–580.
- Lindenbaum, P., *et al.*, 2011. Knime4Bio: A set of custom nodes for the interpretation of next-generation sequencing data with KNIME. *Bioinformatics* 27 (22), 3200–3201.
- Ludäscher, B., *et al.*, 2006. Scientific workflow management and the Kepler system: Research Articles. *Concurrency and Computation: Practice & Experience* 18 (10), 1039–1065.
- Mattiello, F., *et al.*, 2016. A web application for sample size and power calculation in case-control microbiome studies. *Bioinformatics* 32 (13), 2038–2040.
- McKenna, A., *et al.*, 2010. The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20 (9), 1297–1303.
- McPhillips, T., *et al.*, 2015. YesWorkflow: A user-oriented, language-independent tool for recovering workflow information from scripts. *arXiv [cs.SE]*. Available at: <http://arxiv.org/abs/1502.02403>.
- Meyer, F., *et al.*, 2008. The metagenomics RAST server – A public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics* 9 (1), 386.
- Missier, P., *et al.*, 2008. Data lineage model for taverna workflows with lightweight annotation requirements. *Provenance and Annotation of Data and Processes. Lecture Notes in Computer Science. International Provenance and Annotation Workshop. Berlin; Heidelberg: Springer*, pp. 17–30.
- Missier, P., Belhajjame, K., Cheney, J., 2013. The W3C PROV family of specifications for modelling provenance metadata. In: Proceedings of the 16th International Conference on Extending Database Technology, EDBT '13. New York, NY: ACM, pp. 773–776.
- Moreau, L., *et al.*, 2013. PROV-DM: The PROV data model. Retrieved July 30, 2013.
- Moreews, F., *et al.*, 2015. BioShaDock: A community driven bioinformatics shared Docker-based tools registry. *F1000Research* 4, 1443.

- Mullikin, J.C., McMurry, A.A., 1999. Techview: DNA sequencing. Sequencing the genome, fast. *Science* 283 (5409), 1867–1869.
- Murta, L., *et al.*, 2014. noWorkflow: Capturing and analyzing provenance of scripts. In: *Provenance and Annotation of Data and Processes. Lecture Notes in Computer Science. International Provenance and Annotation Workshop*. Cham: Springer, pp. 71–83.
- Neisse, R., Steri, G., Nai-Fovino, I., 2017. A blockchain-based approach for data accountability and provenance tracking. *arXiv [cs.CR]*. Available at: <http://arxiv.org/abs/1706.04507>.
- Oberchain, V., *et al.*, 2014. VariantAnnotation: A bioconductor package for exploration and annotation of genetic variants. *Bioinformatics* 30 (14), 2076–2078.
- O'Connor, B.D., *et al.*, 2017. The dockstore: Enabling modular, community-focused sharing of Docker-based genomics tools and workflows. *F1000Research* 6, 52.
- Paten, B., *et al.*, 2015. The NIH BD2K center for big data in translational genomics. *Journal of the American Medical Informatics Association: JAMIA* 22 (6), 1143–1147.
- Perez, F., Granger, B.E., 2007. IPython: A system for interactive scientific computing. *Computing in Science Engineering* 9 (3), 21–29.
- Plummer, E., Twin, J., 2015. A comparison of three bioinformatics pipelines for the analysis of preterm gut microbiota using 16S rRNA gene sequencing data. *Journal of Proteomics & Bioinformatics* 8 (12), Available at: <https://www.omicsonline.org/open-access/a-comparison-of-three-bioinformatics-pipelines-for-the-analysis-of-preterm-gut-microbiota-using-16s-rna-gene-sequencing-data-jpb-1000381.php?aid=65142..>
- Queralt-Rosinach, N., *et al.*, 2016. DisGeNET-RDF: Harnessing the innovative power of the Semantic Web to explore the genetic basis of diseases. *Bioinformatics* 32 (14), 2236–2238.
- Ramírez, F., *et al.*, 2014. deepTools: A flexible platform for exploring deep-sequencing data. *Nucleic Acids Research* 42 (Web Server Issue), W187–W191.
- Robertson, E.M., *et al.*, 2016. META-pipe – Pipeline annotation, analysis and visualization of marine metagenomic sequence data. *arXiv [cs.DC]*. Available at: <http://arxiv.org/abs/1604.04103>.
- Sandve, G.K., *et al.*, 2013. Ten simple rules for reproducible computational research. *PLOS Computational Biology* 9 (10), e1003285.
- Santana-Perez, I., *et al.*, 2014. A semantic-based approach to attain reproducibility of computational environments in scientific workflows: A case study. In: *Euro-Par 2014: Parallel Processing Workshops. Lecture Notes in Computer Science. European Conference on Parallel Processing*. Cham: Springer, pp. 452–463.
- Schloss, P.D., *et al.*, 2009. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology* 75 (23), 7537–7541.
- Shen, H., 2014. Interactive notebooks: Sharing the code. *Nature* 515 (7525), 151–152.
- Silver, A., 2017. Software simplified. *Nature* 546 (7656), 173–174.
- Sloggett, C., Goonasekera, N., Afgan, E., 2013. BioBlend: Automating pipeline analyses within Galaxy and CloudMan. *Bioinformatics* 29 (13), 1685–1686.
- Soiland-Reyes, S., 2016. 2016-provweek-tavernapro, Github. Available at: <https://github.com/stain/2016-provweek-tavernapro> (accessed 16.08.17).
- Stallman, R., McGrath, R., 2002. GNU make: A program for directing recompilation: GNU Make Version 3.79.1, Free Software Foundation.
- Stein, L.D., 2010. The case for cloud computing in genome informatics. *Genome Biology* 11 (5), 207.
- Stodden, V., Borwein, J., Bailey, D.H., 2013. Setting the default to reproducible computational science research. *SIAM News* 46, 4–6.
- Sztromwasser, P., 2014. Throughput and robustness of bioinformatics pipelines for genome-scale data analysis. Available at: <http://bora.uib.no/bitstream/handle/1956/7906/dr-thesis-2014-Pawe%C5%82-Sztromwasser.pdf?Sequence=3>.
- Thain, D., Tannenbaum, T., Livny, M., 2005. Distributed computing in practice: The Condor experience. *Concurrency and Computation: Practice & Experience* 17 (2–4), 323–356.
2016. A federated ecosystem for sharing genomic, clinical data. *Science* 352 (6291), 1278–1280.
- Trapnell, C., *et al.*, 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols* 7 (3), 562–578.
- Vivian, J., *et al.*, 2017. Toil enables reproducible, open source, big biomedical data analyses. *Nature Biotechnology* 35 (4), 314–316.
- Voss, K., Van der Auwera, G., Gentry, J., 2017. Full-stack genomics pipelining with GATK4 + WDL + Cromwell. *F1000Research* 6. Available at: <http://dx.doi.org/10.7490/f1000research.1114634.1> (accessed 21.08.17)..
- Wang, Z., Ma'ayan, A., 2016. An open RNA-Seq data analysis pipeline tutorial with an example of reprocessing data from a recent Zika virus study. *F1000Research* 5, 1574.
- Wilkinson, M.D., *et al.*, 2016. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data* 3, 160018.
- Wolstencroft, K., *et al.*, 2013. The Taverna workflow suite: Designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Research* 41 (Web Server Issue), W557–W561.
- Xie, Y., 2014. Knitr: A comprehensive tool for reproducible research in R. *Implementing Reproducible Research* 1, 20.
- Zhang, Z., Taylor, D., 2017. GSA-Genie: A web application for gene set analysis. *bioRxiv*. 125443. Available at: <http://biorxiv.org/content/early/2017/04/07/125443.abstract> (accessed 22.06.17)..
- Zheng, C.L., *et al.*, 2015. Use of semantic workflows to enhance transparency and reproducibility in clinical omics. *Genome Medicine* 7, 73.
- Zheng, J., *et al.*, 2016. The ontology of biological and clinical statistics (OBCS) for standardized and reproducible statistical analysis. *Journal of Biomedical Semantics* 7 (1), 53.

Relevant Websites

<https://w3id.org/ro/Wf4Ever>
<http://wf4ever.github.io/ro/Wf4Ever>

Biographical Sketch

Jeremy Leipzig is a bioinformatics software developer at Cytovus LLC in Philadelphia. He received his MS in Computer Science from North Carolina State University. He is pursuing a PhD in Information Studies at Drexel University.

**ENCYCLOPEDIA OF
BIOINFORMATICS AND
COMPUTATIONAL BIOLOGY**

ENCYCLOPEDIA OF BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

EDITORS IN CHIEF

Shoba Ranganathan

Macquarie University, Sydney, NSW, Australia

Michael Gribskov

Purdue University, West Lafayette, IN, United States

Kenta Nakai

The University of Tokyo, Tokyo, Japan

Christian Schönbach

*Nazarbayev University, School of Science and Technology, Department of Biology,
Astana, Kazakhstan*

VOLUME 3

Applications

Mohammad Asif Khan

Centre for Bioinformatics, Perdana University, Selangor, Malaysia



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge MA 02139, United States

Copyright © 2019 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers may always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-12-811414-8

For information on all publications visit our website at <http://store.elsevier.com>

		Working together to grow libraries in developing countries
www.elsevier.com • www.bookaid.org		

Publisher: Oliver Walter
Acquisition Editor: Sam Crowe
Content Project Manager: Paula Davies
Associate Content Project Manager: Ebin Clinton Rozario
Designer: Greg Harris

Printed and bound in the United States

EDITORS IN CHIEF



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. She hosted the Macquarie Node of the ARC Centre of Excellence in Bioinformatics (2008–2013). She was elected the first Australian Board Director of the International Society for Computational Biology (ISCB; 2003–2005); President of Asia-Pacific Bioinformatics Network (2005–2016) and Steering Committee Member (2007–2012) of Bioinformatics Australia. She initiated the Workshops on Education in Bioinformatics (WEB) as an ISMB2001 Special Interest Group meeting and also served as Chair of ICSB's Education Committee. Shoba currently serves as Co-Chair of the Computational Mass Spectrometry (CompMS) initiative of the Human Proteome Organization (HuPO), ISCB and Metabolomics Society and as Board Director, APBioNet Ltd.

Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books as well as articles for the 2013 Encyclopedia of Systems Biology. She is currently an Editor-in-Chief of the Encyclopedia of Bioinformatics and

Computational Biology and the Bioinformatics Section Editor of the Reference Module in Life Science as well as an editorial board member of several bioinformatics journals.



Dr. Gribskov graduated from Oregon State University in 1979 with a Bachelors of Science degree (with Honors) in Biochemistry and Biophysics. He then moved to the University of Wisconsin-Madison for graduate studies focused on the structure and function of the sigma subunit of *E. coli* RNA polymerase, receiving his Ph.D. in 1985. Dr. Gribskov studied X-ray crystallography as an American Cancer Society post-doctoral fellow at UCLA in the laboratory of David Eisenberg, and followed this with both crystallographic and computational studies at the National Cancer Institute. In 1992, Dr. Gribskov moved to the San Diego Supercomputer Center at the University of California, San Diego where he was lead scientist in the area of computational biology and an adjunct associate professor in the department of Biology. From 2003 to 2007, Dr. Gribskov was the president of the International Society for Computational Biology, the largest professional society devoted to bioinformatics and computational biology. In 2004, Dr. Gribskov moved to Purdue University where he holds an appointment as a full professor in the Biological Sciences and Computer Science departments (by courtesy). Dr. Gribskov's interests include genomic and transcriptomic analysis of model and non-model

organisms, the application of pattern recognition and machine learning techniques to biomolecules, the design and implementation of biological databases to support molecular and systems biology, development of methods to study RNA structural patterns, and systems biology studies of human disease.



Kenta Nakai received the PhD degree on the prediction of subcellular localization sites of proteins from Kyoto University in 1992. From 1989, he has worked at Kyoto University, National Institute of Basic Biology, and Osaka University. From 1999 to 2003, he was an Associate Professor at the Human Genome Center, the Institute of Medical Science, the University of Tokyo, Japan. Since 2003, he has been a full Professor at the same institute. His main research interest is to develop computational ways for interpreting biological information, especially that of transcriptional regulation, from genome sequence data. He has published more than 150 papers, some of which have been cited more than 1,000 times.



Christian Schönbach is currently Department Chair and Professor at Department of Biology, School of Science and Technology, Nazarbayev University, Kazakhstan and Visiting Professor at International Research Center for Medical Sciences at Kumamoto University, Japan. He is a bioinformatics practitioner interfacing genetics, immunology and informatics conducting research on major histocompatibility complex, immune responses following virus infection, biomedical knowledge discovery, peroxisomal diseases, and autism spectrum disorder that resulted in more than 80 publications. His previous academic appointments included Professor at Kumamoto University (2016–2017), Nazarbayev University (2013–2016), Kazakhstan, Kyushu Institute of Technology (2009–2013) Japan, Associate Professor at Nanyang Technological University (2006–2009), Singapore, and Team Leader at RIKEN Genomic Sciences Center (2002–2006), Japan. Other prior positions included Principal Investigator at Kent Ridge Digital Labs, Singapore and Research Scientist at Chugai Institute for Molecular Medicine, Inc., Japan. In 2018 he became a member of International Society for Computational Biology (ISCB) Board of Directors. Since 2010 he is serving Asia-Pacific Bioinformatics Network (APBioNet) as Vice-President (Conferences 2010–2016) and President (2016–2018).

VOLUME EDITORS



Mario Cannataro is a Full Professor of Computer Engineering and Bioinformatics at University “Magna Graecia” of Catanzaro, Italy. He is the director of the Data Analytics research center and the chair of the Bioinformatics Laboratory at University “Magna Graecia” of Catanzaro. His current research interests include bioinformatics, medical informatics, data analytics, parallel and distributed computing. He is a Member of the editorial boards of Briefings in Bioinformatics, High-Throughput, Encyclopedia of Bioinformatics and Computational Biology, Encyclopedia of Systems Biology. He was guest editor of several special issues on bioinformatics and he is serving as a program committee member of several conferences. He published three books and more than 200 papers in international journals and conference proceedings. Prof. Cannataro is a Senior Member of IEEE, ACM and BITS, and a member of the Board of Directors for ACM SIGBIO.



Bruno Gaeta is Senior Lecturer and Director of Studies in Bioinformatics in the School of Computer Science and Engineering at UNSW Australia. His research interests cover multiple areas of bioinformatics including gene regulation and protein structure, currently with a focus on the immune system, antibody genes and the generation of antibody diversity. He is a pioneer of bioinformatics education and has trained thousands of biologists and trainee bioinformaticians in the use of computational tools for biological research through courses, workshops as well as a book series. He has worked both in academia and in the bioinformatics industry, and currently coordinates the largest bioinformatics undergraduate program in Australia.



Mohammad Asif Khan, PhD, is an associate professor and the Dean of the School of Data Sciences, as well as the Director of the Centre for Bioinformatics at Perdana University, Malaysia. He is also a visiting scientist at the Department of Pharmacology and Molecular Sciences, Johns Hopkins University School of Medicine (JHUSOM), USA. His research interests are in the area of biological data warehousing and applications of bioinformatics to the study of immune responses, vaccines, inhibitory drugs, venom toxins, and disease biomarkers. He has published in these areas, been involved in the development of several novel bioinformatics methodologies, tools, and specialized databases, and currently has three patent applications granted. He has also led the curriculum development of a Postgraduate Diploma in Bioinformatics programme and an MSc (Bioinformatics) programme at Perdana University. He is an elected ExCo member of the Asia-Pacific Bioinformatics Network (APBioNET) since 2010 and is currently the President of Association for Medical and Bio-Informatics, Singapore (AMBIS). He has donned various important roles in the organization of many local and international bioinformatics conferences, meetings and workshops.

CONTENTS OF VOLUME 3

Editors in Chief	v
Volume Editors	vii
List of Contributors for Volume 3	xv
Preface	xxiii

VOLUME 3

Ecosystem Monitoring Through Predictive Modeling <i>Sorayya Malek, Cham Hui, Nanyonga Aziida, Song Cheen, Sooh Toh, and Pozi Milow</i>	1
Large Scale Ecological Modeling With Viruses: A Review <i>Vijayaraghava S Sundararajan</i>	9
Mapping the Environmental Microbiome <i>Lena Takayasu, Wataru Suda, and Masahira Hattori</i>	17
Biological Database Searching <i>Nor A Nor Muhammad</i>	29
Extraction of Immune Epitope Information <i>Guang Lan Zhang, Derin B Keskin, and Lou Chitkushev</i>	39
Characterizing and Functional Assignment of Noncoding RNAs <i>Pradeep Tiwari, Sonal Gupta, Anuj Kumar, Mansi Sharma, Vijayaraghava S Sundararajan, Shanker L Kothari, Sandeep K Mathur, Krishna M Medicherla, Babita Malik, and Prashanth Suravajhala</i>	47
Identification and Extraction of Biomarker Information <i>Vijayaraghava S Sundararajan, Shweta Shrotriya, Monisha Singhal, Pragya Chaturvedi, and Prashanth Suravajhala</i>	60
Computational Systems Biology Applications <i>Ayako Yachie-Kinoshita, Sucheendra K Palaniappan, and Samik Ghosh</i>	66
Coupling Cell Division to Metabolic Pathways Through Transcription <i>Petter Holland, Jens Nielsen, Thierry DGA Mondeel, and Matteo Barberis</i>	74
Studies of Body Systems <i>Amir Feisal Merican, Hoda Mirsafian, Adiratna Mat Ripen, and Saharuddin Bin Mohamad</i>	94
Host-Pathogen Interactions <i>Dean Southwood and Shoba Ranganathan</i>	103
Computational Pipelines and Workflows in Bioinformatics <i>Yosvany López, Piotr J Kamola, Ronesh Sharma, Daichi Shigemizu, Tatsuhiko Tsunoda, and Alok Sharma</i>	113
Constructing Computational Pipelines <i>ChuangKee Ong and Russell S Hamilton</i>	135
Pipeline of High Throughput Sequencing <i>ChuangKee Ong, Qi Bin Kwong, Ai Ling Ong, Huey Ying Heng, and Wai Yee Low</i>	144
Genome Annotation: Perspective From Bacterial Genomes <i>Alan Christoffels and Peter van Heusden</i>	152

Next Generation Sequencing Data Analysis <i>Ranjeev Hari and Suhanya Parthasarathy</i>	157
Exome Sequencing Data Analysis <i>Anita Sathyanarayanan, Srikanth Manda, Mukta Poojary, and Shivashankar H Nagaraj</i>	164
Whole Genome Sequencing Analysis <i>Rui Yin, Chee Keong Kwoh, and Jie Zheng</i>	176
Metagenomic Analysis and its Applications <i>Arpita Ghosh, Aditya Mehta, and Asif M Khan</i>	184
Integrative Analysis of Multi-Omics Data <i>Lokesh P Tripathi, Tsuyoshi Esaki, Mari N Itoh, Yi-An Chen, and Kenji Mizuguchi</i>	194
Profiling the Gut Microbiome: Practice and Potential <i>Toral Manvar and Vijay Lakhujani</i>	200
Functional Enrichment Analysis <i>Srikanth Manda, Daliah Michael, Sudhir Jadhao, and Shivashankar H Nagaraj</i>	218
Prediction of Coding and Non-Coding RNA <i>Ranjeev Hari and Suhanya Parthasarathy</i>	230
Vaccine Target Discovery <i>Li C Chong and Asif M Khan</i>	241
Protocol for Protein Structure Modelling <i>Amara Jabeen, Abidali Mohamedali, and Shoba Ranganathan</i>	252
Structure-Based Drug Design Workflow <i>Ari Hardianto, Muhammad Yusuf, Fei Liu, and Shoba Ranganathan</i>	273
Network-Based Analysis for Biological Discovery <i>Lokesh P Tripathi, Yoichi Murakami, Yi-An Chen, and Kenji Mizuguchi</i>	283
Sequence Analysis <i>Andrey D Prjibelski, Anton I Korobeynikov, and Alla L Lapidus</i>	292
Sequence Composition <i>Bryan T Li, Jin Xing Lim, and Maurice HT Ling</i>	323
Codon Usage <i>Raimi M Redwan, Suhanya Parthasarathy, and Ranjeev Hari</i>	327
Identification of Sequence Patterns, Motifs and Domains <i>Michael Gribskov</i>	332
Epigenetics: Analysis of Cytosine Modifications at Single Base Resolution <i>Malwina Prater and Russell S Hamilton</i>	341
Comparative Epigenomics <i>Yutaka Saito</i>	354
Genetics and Population Analysis <i>Fotis Tsetsos, Petros Drineas, and Peristera Paschou</i>	363
Population Analysis of Pharmacogenetic Polymorphisms <i>Zen H Lu, Nani Azman, Mark IR Petalcorin, Naeem Shafqat, and Lie Chen</i>	379
Detecting and Annotating Rare Variants <i>Jieming Chen, Akdes S Harmanci, and Arif O Harmanci</i>	388

Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases <i>Marwa S Hassan, AA Shaalan, MI Dessouky, Abdelaziz E Abdelnaiem, Dalia A Abdel-Haleem, and Mahmoud ElHefnawi</i>	400
Genome Analysis – Identification of Genes Involved in Host-Pathogen Protein-Protein Interaction Networks <i>Fransiskus Xaverius Ivan, Jie Zheng, Vincent TK Chow, and Chee-Keong Kwoh</i>	410
Comparative Genomics Analysis <i>Hui San Ong</i>	425
Single Nucleotide Polymorphism Typing <i>Srilakshmi Srinivasan and Jyotsna Batra</i>	432
Genome-Wide Haplotype Association Study <i>Yongshuai Jiang and Jing Xu</i>	441
Prediction of Protein-Binding Sites in DNA Sequences <i>Kenta Nakai</i>	447
Genome-Wide Scanning of Gene Expression <i>Sung-J Park</i>	452
Metabolome Analysis <i>Camila Pereira Braga and Jiri Adamec</i>	463
Disease Biomarker Discovery <i>Tiratha R Singh, Ankita Shukla, Bensellak Taoufik, Ahmed Moussa, and Brigitte Vannier</i>	476
Investigating Metabolic Pathways and Networks <i>Md Altaf-Ul-Amin, Zeti-Azura Mohamed-Hussein, and Shigehiko Kanaya</i>	489
Biomolecular Structures: Prediction, Identification and Analyses <i>Prasun Kumar, Swagata Halder, and Manju Bansal</i>	504
Engineering of Supramolecular RNA Structures <i>Effirul I Ramlan and Mohd Firdaus-Raih</i>	535
Predicting RNA-RNA Interactions in Three-Dimensional Structures <i>Hazrina Y Hamdani, Zatil H Yahaya, and Mohd Firdaus-Raih</i>	546
Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights <i>Chyan Leong Ng and Mohd Firdaus-Raih</i>	554
Computational Design and Experimental Implementation of Synthetic Riboswitches and Riboregulators <i>Munyati Othman, Siuk M Ng, and Mohd Firdaus-Raih</i>	568
Genome-Wide Probing of RNA Structure <i>Xiaojing Huo, Jeremy Ng, Mingchen Tan, and Greg Tucker-Kellogg</i>	574
Assessment of Structure Quality (RNA and Protein) <i>Nicolas Palopoli</i>	586
Study of the Variability of the Native Protein Structure <i>Xusi Han, Woong-Hee Shin, Charles W Christoffer, Genki Terashi, Lyman Monroe, and Daisuke Kihara</i>	606
Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures <i>Mohd Firdaus-Raih and Nur Syatila Abdul Ghani</i>	620

Computational Protein Engineering Approaches for Effective Design of New Molecules <i>Jaspreet Kaur Dhanjal, Vidhi Malik, Navaneethan Radhakrishnan, Moolchand Sigar, Anjani Kumari, and Durai Sundar</i>	631
Protein Design <i>Ragothaman M Yennamalli</i>	644
Molecular Dynamics Simulations in Drug Discovery <i>Sy-Bing Choi, Beow Keat Yap, Yee Siew Choong, and Habibah Wahab</i>	652
Protein-Carbohydrate Interactions <i>Adeel Malik, Mohammad H Baig, and Balachandran Manavalan</i>	666
Computational Prediction of Nucleic Acid Binding Residues From Sequence <i>Nur SA Ghani, Shandar Ahmad, and Mohd Firdaus-Raih</i>	678
Protein-Peptide Interactions in Regulatory Events <i>Upadhyayula S Raghavender and Ravindranath S Rathore</i>	688
Homologous Protein Detection <i>Xuefeng Cui and Yaosen Min</i>	697
Mutation Effects on 3D-Structural Reorganization Using HIV-1 Protease as a Case Study <i>Biswa R Meher, Megha Vaishnavi, Venkata SK Mattaparthi, Seema Patel, and Sandeep Kaushik</i>	706
Structural Genomics <i>Shuhaila Mat-Sharani, Doris HX Quay, Chyan L Ng, and Mohd Firdaus-Raih</i>	722
Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4) <i>Teo Chian Ying, Zalikha Ibrahim, Mohd Basyaruddin Abd Rahman, and Bimo A Tejo</i>	729
Small Molecule Drug Design <i>Vartika Tomar, Mohit Mazumder, Ramesh Chandra, Jian Yang, and Meena K Sakharkar</i>	741
In Silico Identification of Novel Inhibitors <i>Beow Keat Yap, Chong-Yew Lee, Sy Bing Choi, Ezatul E Kamarulzaman, Maywan Hariono, and Habibah A Wahab</i>	761
Drug Repurposing and Multi-Target Therapies <i>Ammu P Kumar, Minh N Nguyen, and Suryani Lukman</i>	780
Transcriptome Analysis <i>Anuj Srivastava, Joshy George, and Radha KM Karuturi</i>	792
Regulation of Gene Expression <i>Y-h Taguchi</i>	806
Comparative Transcriptomics Analysis <i>Y-h Taguchi</i>	814
Analyzing Transcriptome-Phenotype Correlations <i>Bryan T Li, Jin X Lim, and Maurice HT Ling</i>	819
Transcriptome During Cell Differentiation <i>Dwi A Pujianto</i>	825
Characterization of Transcriptional Activities <i>Adison Wong and Maurice HT Ling</i>	830
Survey of Antisense Transcription <i>Maurice HT Ling</i>	842
Statistical and Probabilistic Approaches to Predict Protein Abundance <i>Rubbiya A Ali, Ali Naqi, Sarah Ali, Mashal Fatima, Musarat Ishaq, and Ahmed M Mehdi</i>	847

Identification of Proteins From Proteomic Analysis <i>Zainab Noor, Abidali Mohamedali, and Shoba Ranganathan</i>	855
Quantification of Proteins From Proteomic Analysis <i>Zainab Noor, Subash Adhikari, Shoba Ranganathan, and Abidali Mohamedali</i>	871
Utilising IPG-IEF to Identify Differentially-Expressed Proteins <i>David I Cantor and Harish R Cheruku</i>	891
Clinical Proteomics <i>Marwenie F Petalcorin, Naeem Shafqat, Zen H Lu, and Mark IR Petalcorin</i>	911
Molecular Mechanisms Responsible for Drug Resistance <i>Mun Fai Loke and Aimi Hanafi</i>	926
Network-Based Analysis of Host-Pathogen Interactions <i>Lokesh P Tripathi, Eiji Morita, Yi-An Chen, and Kenji Mizuguchi</i>	932
Phylogenetic Analysis: Early Evolution of Life <i>Himakshi Sarma, Sushmita Pradhan, Sandeep Kaushik, and Venkata SK Mattaparthi</i>	938
Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material <i>Suhaila Sulaiman, Nur S Yusoff, Ng S Mun, Haslina Makmur, and Mohd Firdaus-Raih</i>	953
Gene Duplication and Speciation <i>Tiratha R Singh and Ankush Bansal</i>	965
Epidemiology: A Review <i>Vijayaraghava S Sundararajan</i>	975
Identification of Homologs <i>William R Pearson</i>	980
DNA Barcoding: Bioinformatics Workflows for Beginners <i>John-James Wilson, Kong-Wah Sing, and Narong Jaturas</i>	985
Text Mining Applications <i>Raul Rodriguez-Esteban</i>	996
Index	1001

LIST OF CONTRIBUTORS FOR VOLUME 3

Dalia A. Abdel-Haleem
National Research Centre, Cairo, Egypt

Abdelaziz E. Abdelnaiem
Zagazig University, Zagazig, Egypt

Jiri Adamec
University of Nebraska-Lincoln, Lincoln, NE, United States

Subash Adhikari
Macquarie University, Sydney, NSW, Australia

Shandar Ahmad
Jawaharlal Nehru University, New Delhi, India

Rubbiya A. Ali
University of Queensland, Brisbane, QLD, Australia

Sarah Ali
Hamdard Institute of Engineering and Technology, Islamabad, Pakistan

Md. Altaf-Ul-Amin
Nara Institute of Science and Technology, Ikoma, Japan

Nanyonga Aziida
University of Malaya, Kuala Lumpur, Malaysia

Nani Azman
PAPRSB Institute of Health Sciences, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Mohammad H. Baig
Yeungnam University Gyeongsan, Gyeongbuk, South Korea

Manju Bansal
Indian Institute of Science, Bangalore, India

Ankush Bansal
Jaypee University of Information Technology, Solan, India

Matteo Barberis
University of Amsterdam, Amsterdam, The Netherlands

Jyotsna Batra
Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD, Australia; and Australian Prostate Cancer Research Centre – Queensland, Translational Research Institute, Woolloongabba, QLD, Australia

Saharuddin Bin Mohamad
University of Malaya, Kuala Lumpur, Malaysia

David I. Cantor
Macquarie University, Sydney, NSW, Australia

Ramesh Chandra
University of Delhi, Delhi, India

Pragya Chaturvedi
Birla Institute of Scientific Research, Jaipur, India

Song Cheen
University of Malaya, Kuala Lumpur, Malaysia

Lie Chen
PAPRSB Institute of Health Sciences, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Yi-An Chen
National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Jieming Chen
Genentech Inc., South San Francisco, CA, United States

Yi-An Chen
National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Harish R. Cheruku
Preston, VIC, Australia

Lou Chitkushev
Boston University, Boston, MA, United States

Sy-Bing Choi
Universiti Sains Malaysia, Pulau Pinang, Malaysia

Sy Bing Choi
Universiti Sains Malaysia, Pulau Pinang, Malaysia

Li C. Chong
Universiti Putra Malaysia, Serdang, Malaysia

Yee Siew Choong
Universiti Sains Malaysia, Pulau Pinang, Malaysia

Vincent T.K. Chow
National University of Singapore, Singapore

Alan Christoffels
University of the Western Cape, Cape Town, South Africa

Charles W. Christoffer
Purdue University, West Lafayette, IN, United States

Xuefeng Cui
Tsinghua University, Beijing, China

M.I. Dessouky
Menoufia University, Menouf, Egypt

Jaspreet Kaur Dhanjal
DBT-AIST International Laboratory for Advanced Biomedicine (DAILAB), Indian Institute of Technology Delhi, New Delhi, India

Petros Drineas
Purdue University, West Lafayette, IN, United States

Mahmoud ElHefnawi
Nile University, Giza, Egypt and National Research Centre, Cairo, Egypt.

Tsuyoshi Esaki
National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Mashal Fatima
Bahauddin Zakariya University, Multan, Pakistan

Antonino Fiannaca
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Mohd Firdaus-Raih
Universiti Kebangsaan Malaysia, Bangi, Malaysia

Joshy George
The Jackson Laboratory, CT, United States

Nur Syatila Abdul Ghani
Universiti Kebangsaan Malaysia, Bangi, Malaysia

Samik Ghosh
The Systems Biology Institute, Tokyo, Japan

Arpita Ghosh
Eurofins Genomics India Pvt. Ltd., Bangalore, India; and Centre for Bioinformatics, Perdana University, Selangor, Malaysia

Michael Gribskov
Purdue University, West Lafayette, IN, United States

Massimo Guarascio
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Sonal Gupta
Birla Institute of Scientific Research, Jaipur, India; and Amity University, Jaipur, India

Swagata Halder
Indian Institute of Science, Bangalore, India

Hazrina Y. Hamdani
Universiti Sains Malaysia, Kepala Batas, Malaysia

Russell S. Hamilton
University of Cambridge, Cambridge, United Kingdom

Xusi Han
Purdue University, West Lafayette, IN, United States

Aimi Hanafi
University of Malaya, Kuala Lumpur, Malaysia

Ari Hardianto
Macquarie University, Sydney, NSW, Australia; and Universitas Padjadjaran, Jatinangor, West Java, Indonesia

Ranjeew Hari
Perdana University, Selangor, Malaysia

Maywan Hariono
Sanata Dharma University, Sleman, Indonesia

Akdes S. Harmanci
The University of Texas Health Science Center at Houston, Houston, TX, United States

Marwa S. Hassan
National Research Centre, Cairo, Egypt

Masahira Hattori
RIKEN Center for Integrative Medical Sciences, Yokohama City, Japan; and Waseda University, Tokyo, Japan

Huey Ying Heng
Sime Darby Plantation R&D Centre, Selangor, Malaysia

Petter Holland
Chalmers University of Technology, Gothenburg, Sweden

Cham Hui
University of Malaya, Kuala Lumpur, Malaysia

Xiaoqing Huo
National University of Singapore, Singapore

Zalikha Ibrahim
International Islamic University Malaysia, Kuantan, Malaysia

Musarat Ishaq
St. Vincent's Institute of Medical Research, Melbourne, VIC, Australia

Mari N. Itoh
National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Fransiskus Xaverius Ivan
Nanyang Technological University, Singapore

Amara Jabeen
Macquarie University, Sydney, NSW, Australia

Sudhir Jadhao
*Institute of Health and Biomedical Innovation,
 School of Biomedical Sciences, Queensland University
 of Technology, Brisbane, QLD, Australia*

Narong Jaturas
Naresuan University, Phitsanulok, Thailand

Yongshuai Jiang
Harbin Medical University, Harbin, China

Ezatul E. Kamarulzaman
*Universiti Sains Malaysia, Minden, Pulau Pinang,
 Malaysia*

Piotr J. Kamola
*RIKEN Center for Integrative Medical Sciences,
 Yokohama, Kanagawa, Japan; and Japan Science
 and Technology Agency, Tokyo, Japan*

Shigehiko Kanaya
Nara Institute of Science and Technology, Ikoma, Japan

Radha K.M. Karuturi
The Jackson Laboratory, CT, United States

Sandeep Kaushik
*European Institute of Excellence on Tissue Engineering
 and Regenerative Medicine, Guimaraes, Portugal*

Derin B. Keskin
*Harvard Medical School, Boston, MA, United States;
 and Broad Institute, Boston, MA, United States*

Asif M. Khan
*Perdana University, Selangor, Malaysia; and John
 Hopkins University, Baltimore, MD, United States*

Asif M. Khan
Perdana University, Selangor, Malaysia

Daisuke Kihara
Purdue University, West Lafayette, IN, United States

Anton I. Korobeynikov
St. Petersburg State University, St Petersburg, Russia

Shanker L. Kothari
Amity University, Jaipur, India

Anuj Kumar
*Uttarakhand Council for Biotechnology, Dehradun,
 India*

Prasun Kumar
University of Bristol, Bristol, United Kingdom

Ammu P. Kumar
*Khalifa University of Science and Technology, Abu
 Dhabi, United Arab Emirates*

Anjani Kumari
*DBT-AIST International Laboratory for Advanced
 Biomedicine (DAILAB), Indian Institute of Technology
 Delhi, New Delhi, India*

Chee Keong Kwoh
Nanyang Technological University, Singapore

Chee-Keong Kwoh
Nanyang Technological University, Singapore

Qi Bin Kwong
Sime Darby Plantation R&D Centre, Selangor, Malaysia

Massimo La Rosa
*Universiti Brunei Darussalam, Bandar Seri Begawan,
 Brunei Darussalam*

Yosvany LÃ³pez
Genesis Healthcare Co., Tokyo, Japan

Vijay Lakhujani
Xcelris Labs Ltd., Ahmedabad, Gujarat, India

Alla L. Lapidus
St. Petersburg State University, St Petersburg, Russia

Chong-Yew Lee
*Universiti Sains Malaysia, Minden, Pulau Pinang,
 Malaysia*

Bryan T. Li
Temasek Polytechnic, Singapore

Jin X. Lim
Temasek Polytechnic, Singapore

Maurice H.T. Ling
Colossus Technologies, LLP, Singapore

Maurice H.T. Ling
*Colossus Technologies, LLP, Singapore; and University of
 Melbourne, Melbourne, VIC, Australia*

Fei Liu
Macquarie University, Sydney, NSW, Australia

Mun Fai Loke
University of Malaya, Kuala Lumpur, Malaysia

Wai Yee Low
*Davies Research Centre, University of Adelaide,
 Roseworthy, SA, Australia*

Zen H. Lu
*Universiti Brunei Darussalam, Bandar Seri Begawan,
 Brunei Darussalam*

Suryani Lukman
*Khalifa University of Science and Technology, Abu
 Dhabi, United Arab Emirates*

Haslina Makmur
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Sorayya Malek
University of Malaya, Kuala Lumpur, Malaysia

Babita Malik
Manipal University, Jaipur, India

Vidhi Malik
DBT-AIST International Laboratory for Advanced Biomedicine (DAILAB), Indian Institute of Technology Delhi, New Delhi, India

Adeel Malik
Chungnam National University, Daejeon, South Korea

Balachandran Manavalan
Ajou University School of Medicine, Suwon, Republic of Korea

Giuseppe Manco
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Srikanth Manda
Institute of Bioinformatics, Bangalore, India

Toral Manvar
Xcelris Labs Ltd., Ahmedabad, Gujarat, India

Sandeep K. Mathur
SMS Medical College, Jaipur, India

Shuhaila Mat-Sharani
Universiti Kebangsaan Malaysia, Selangor, Malaysia

Venkata S.K. Mattaparthi
Tezpur University, Tezpur, Assam, India

Mohit Mazumder
Jawaharlal Nehru University, Delhi, India; and University of Saskatchewan, Saskatoon, Canada

Krishna M. Medicherla
Birla Institute of Scientific Research, Jaipur, India

Ahmed M. Mehdi
University of Queensland, Brisbane, QLD, Australia

Biswa R. Meher
Berhampur University, Berhampur, India

Aditya Mehta
Eurofins Genomics India Pvt. Ltd., Bangalore, India

Amir Feisal. Merican
University of Malaya, Kuala Lumpur, Malaysia

Daliah Michael
Indian Institute of Science, Bangalore, India

Pozi Milow
University of Malaya, Kuala Lumpur, Malaysia

Yaosen Min
Tsinghua University, Beijing, China

Hoda Mirsafian
University of Malaya, Kuala Lumpur, Malaysia; and University of Southern California, Los Angeles, CA, United States

Kenji Mizuguchi
National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Abidali Mohamedali
Macquarie University, Sydney, NSW, Australia

Zeti-Azura Mohamed-Hussein
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Thierry D.G.A. Mondeel
University of Amsterdam, Amsterdam, Netherlands

Lyman Monroe
Purdue University, West Lafayette, IN, United States

Eiji Morita
Hirosaki University, Hirosaki, Japan

Ahmed Moussa
École Nationale Des Sciences Appliquées de Tanger, Tangier, Morocco

Ng S. Mun
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Yoichi Murakami
Tokyo University of Information Sciences, Wakaba-ku, Chiba, Japan

Shivashankar H Nagaraj
Nanyang Technological University, Singapore

Shivashankar H. Nagaraj
Perdana University, Selangor, Malaysia

Kenta Nakai
The University of Tokyo, Tokyo, Japan

Ali Naqi
Queensland University of Technology, Brisbane, QLD, Australia

Siuk M. Ng
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Jeremy Ng
National University of Singapore, Singapore

Chyan L. Ng
Universiti Kebangsaan Malaysia, Selangor, Malaysia

Minh N. Nguyen
Agency for Science, Technology and Research, Singapore

Jens Nielsen
Chalmers University of Technology, Gothenburg, Sweden

Zainab Noor
Macquarie University, Sydney, NSW, Australia

Nor A. Nor Muhammad
Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia

ChuangKee Ong
European Bioinformatics Institute (EMBL-EBI), Hinxton, United Kingdom

Ai Ling Ong
Sime Darby Plantation R&D Centre, Selangor, Malaysia

Hui San Ong
Perdana University, Selangor, Malaysia

Munyati Othman
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Sucheendra K. Palaniappan
The Systems Biology Institute, Tokyo, Japan

Nicolas Palopoli
Universidad Nacional de Quilmes & CONICET, Bernal, Buenos Aires, Argentina

Sung-J. Park
The University of Tokyo, Tokyo, Japan

Suhanya Parthasarathy
Perdana University, Selangor, Malaysia

Suhanya Parthasarathy
Monash University Malaysia, Segamat, Malaysia

Peristera Paschou
Purdue University, West Lafayette, IN, United States

Seema Patel
San Diego State University, San Diego, CA, United States

William R. Pearson
University of Virginia School of Medicine, Charlottesville, VA, United States

Camila Pereira Braga
University of Nebraska-Lincoln, Lincoln, NE, United States

Mark I.R. Petalcorin
PAPRSB Institute of Health Sciences, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Marwenie F. Petalcorin
Queen Mary University of London, London, United Kingdom

Mark I.R. Petalcorin
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Mukta Poojary
Institute of Bioinformatics, Bangalore, India

Sushmita Pradhan
Tezpur University, Tezpur, Assam, India

Malwina Prater
University of Cambridge, Cambridge, United Kingdom

Andrey D. Prijibelski
St. Petersburg State University, St Petersburg, Russia

Dwi A. Pujianto
Universitas Indonesia, Jakarta, Indonesia

Doris H.X. Quay
Universiti Kebangsaan Malaysia, Selangor, Malaysia

Navaneethan Radhakrishnan
DBT-AIST International Laboratory for Advanced Biomedicine (DAILAB), Indian Institute of Technology Delhi, New Delhi, India

Upadhyayula S. Raghavender
Birla Institute of Scientific Research (BISR), Jaipur, India

Mohd Basyaruddin Abd Rahman
Universiti Putra Malaysia, Serdang, Selangor, Malaysia

Effirul I. Ramlan
Malaysia Genome Institute (MGI-NIBM), Kajang, Selangor, Malaysia; and University of Malaya, Kuala Lumpur, Malaysia

Shoba Ranganathan
Macquarie University, Sydney, NSW, Australia

Ravindranath S. Rathore
Central University of South Bihar, Patna, India

Valentina Ravi
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Raimi M. Redwan
Faculty of Agro-based Industry, University of Malaysia, Kelantan, Malaysia

Adiratna Mat Ripen
Institute of Medical Research, Kuala Lumpur, Malaysia

Raul Rodriguez-Esteban
F. Hoffmann-La Roche Ltd., Basel, Switzerland

Yutaka Saito
National Institute of Advanced Industrial Science and Technology (AIST), Koto-ku, Japan; and National Institute of Advanced Industrial Science and Technology (AIST), Shinjuku-ku, Japan

Meena K. Sakharkar
University of Saskatchewan, Saskatoon, SK, Canada

Himakshi Sarma
Tezpur University, Tezpur, India

Anita Sathyanarayanan
Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD, Australia.

A.A. Shaalan
Zagazig University, Zagazig, Egypt

Naeem Shafqat
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Mansi Sharma
Uttarakhand Council for Biotechnology, Dehradun, India

Ronesh Sharma
University of the South Pacific, Suva, Fiji; and Fiji National University, Suva, Fiji

Alok Sharma
RIKEN Center for Integrative Medical Sciences, Yokohama, Japan, University of the South Pacific, Suva, Fiji; and Griffith University, Brisbane, QLD, Australia

Daichi Shigemizu
RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; Japan Science and Technology Agency, Tokyo, Japan; RIKEN Cluster for Science and Technology Hub, Yokohama, Japan; National Center for Geriatrics and Gerontology, Obu, Japan; and Tokyo Medical and Dental University, Tokyo, Japan

Woong-Hee Shin
Purdue University, West Lafayette, IN, United States

Shweta Shrotriya
Birla Institute of Scientific Research, Jaipur, India

Ankita Shukla
Jaypee University of Information Technology, Solan, India

Moolchand Sagar
DBT-AIST International Laboratory for Advanced Biomedicine (DAILAB), Indian Institute of Technology Delhi, New Delhi, India

Kong-Wah Sing
Kunming Institute of Zoology, Chinese Academy of Sciences, Yunnan, P.R. China

Tiratha R. Singh
Jaypee University of Information Technology, Solan, India

Monisha Singhal
Birla Institute of Scientific Research, Jaipur, India

Dean Southwood
Macquarie University, Sydney, NSW, Australia

Srilakshmi Srinivasan
Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD,, Australia

Anuj Srivastava
The Jackson Laboratory, CT, United States

Wataru Suda
RIKEN Center for Integrative Medical Sciences, Yokohama City, Japan

Suhaila Sulaiman
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

Durai Sundar
DBT-AIST International Laboratory for Advanced Biomedicine (DAILAB), Indian Institute of Technology Delhi, New Delhi, India

Vijayaraghava S. Sundararajan
Ministry of Environmental Agency, Singapore; and Bioclues.org, Hyderabad, India

Prashanth Suravajhala
Birla Institute of Scientific Research, Jaipur, India

Y.-h. Taguchi
Chuo University, Tokyo, Japan

Lena Takayasu
RIKEN Center for Integrative Medical Sciences, Yokohama City, Japan

Mingchen Tan
National University of Singapore, Singapore

Bensellak Taoufik
École Nationale Des Sciences Appliquées de Tanger, Tangier, Morocco

Bimo A. Tejo
UCSI University, Cheras, Kuala Lumpur, Malaysia

Genki Terashi
Purdue University, West Lafayette, IN, United States

Pradeep Tiwari
Manipal University, Jaipur, India; Birla Institute of Scientific Research, Jaipur, India; and SMS Medical College, Jaipur, India

Sooh Toh
University of Malaya, Kuala Lumpur, Malaysia

Vartika Tomar
University of Delhi, Delhi, India

Lokesh P. Tripathi
National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Fotis Tsetsos
Democritus University of Thrace, Alexandroupolis, Greece

Tatsuhiko Tsunoda
RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; Japan Science and Technology Agency, Tokyo, Japan; and Tokyo Medical and Dental University, Tokyo, Japan

Greg Tucker-Kellogg
National University of Singapore, Singapore

Alfonso Urso
Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Megha Vaishnavi
Central University of Jharkhand, Ranchi, India

Peter van Heusden
University of the Western Cape, Cape Town, South Africa

Brigitte Vannier
University of Poitiers, Poitiers France

Habibah Wahab
Universiti Sains Malaysia, Pulau Pinang, Malaysia

Habibah A. Wahab
Universiti Sains Malaysia, Pulau Pinang, Malaysia

John-James Wilson
University of South Wales, Pontypridd, United Kingdom; and Naresuan University, Phitsanulok, Thailand

Adison Wong
Singapore Institute of Technology, Singapore

Jin Xing Lim
Temasek Polytechnic, Singapore

Jing Xu
Harbin Medical University, Harbin, China

Ayako Yachie-Kinoshita
The Systems Biology Institute, Tokyo, Japan

Zatil H. Yahaya
Universiti Kebangsaan Malaysia, Bangi, Malaysia

Jian Yang
University of Saskatchewan, Saskatoon, SK, Canada

Beow Keat Yap
Universiti Sains Malaysia, Pulau Pinang, Malaysia

Ragothaman M. Yennamalli
Jaypee University of Information Technology, Warknaghat, Himachal Pradesh, India

Rui Yin
Nanyang Technological University, Singapore

Teo Chian Ying
International Medical University, Bukit Jalil, Kuala Lumpur, Malaysia

Nur S. Yusoff
Universiti Kebangsaan Malaysia, Bangi, Malaysia

Muhammad Yusuf
Universitas Padjadjaran, Jatinangor, Indonesia

Guang Lan Zhang
Boston University, Boston, MA, United States and Harvard Medical School, Boston, MA, United States

Jie Zheng
Nanyang Technological University, Singapore, Singapore

PREFACE

Bioinformatics and Computational Biology (BCB) combine elements of computer science, information technology, mathematics, statistics, and biotechnology, providing the methodology and *in silico* solutions to mine biological data and processes, for knowledge discovery. In the era of molecular diagnostics, targeted drug design and Big Data for personalized or even precision medicine, computational methods for data analysis are essential for biochemistry, biology, biotechnology, pharmacology, biomedical science, and mathematics and statistics. Bioinformatics and Computational Biology are essential for making sense of the molecular data from many modern high-throughput studies of mice and men, as well as key model organisms and pathogens. This Encyclopedia spans basics to cutting-edge methodologies, authored by leaders in the field, providing an invaluable resource to students as well as scientists, in academia and research institutes as well as biotechnology, biomedical and pharmaceutical industries.

Navigating the maze of confusing and often contradictory jargon combined with a plethora of software tools is often confusing for students and researchers alike. This comprehensive and unique resource provides up-to-date theory and application content to address molecular data analysis requirements, with precise definition of terminology, and lucid explanations by experts.

No single authoritative entity exists in this area, providing a comprehensive definition of the myriad of computer science, information technology, mathematics, statistics, and biotechnology terms used by scientists working in bioinformatics and computational biology. Current books available in this area as well as existing publications address parts of a problem or provide chapters on the topic, essentially addressing practicing bioinformaticists or computational biologists. Newcomers to this area depend on Google searches leading to published literature as well as several textbooks, to collect the relevant information.

Although curricula have been developed for Bioinformatics education for two decades now (Altman, 1998), offering education in bioinformatics continues to remain challenging from the multidisciplinary perspective, and is perhaps an NP-hard problem (Ranganathan, 2005). A minimum Bioinformatics skill set for university graduates has been suggested (Tan *et al.*, 2009). The Bioinformatics section of the Reference Module in Life Sciences (Ranganathan, 2017) commenced by addressing the paucity of a comprehensive reference book, leading to the development of this Encyclopedia. This compilation aims to fill the “gap” for readers with succinct and authoritative descriptions of current and cutting-edge bioinformatics areas, supplemented with the theoretical concepts underpinning these topics.

This Encyclopedia comprises three sections, covering Methods, Topics and Applications. The theoretical methodology underpinning BCB are described in the Methods section, with Topics covering traditional areas such as phylogeny, as well as more recent areas such as translational bioinformatics, cheminformatics and computational systems biology. Additionally, Applications will provide guidance for commonly asked “how to” questions on scientific areas described in the Topics section, using the methodology set out in the Methods section. Throughout this Encyclopedia, we have endeavored to keep the content as lucid as possible, making the text “... as simple as possible, but not simpler,” attributed to Albert Einstein. Comprehensive chapters provide overviews while details are provided by shorter, encyclopedic chapters.

During the planning phase of this Encyclopedia, the encouragement of Elsevier’s Priscilla Braglia and the constructive comments from no less than ten reviewers lead our small preliminary editorial team (Christian Schönbach, Kenta Nakai and myself) to embark on this massive project. We then welcomed one more Editor-in-Chief, Michael Gribskov and three section editors, Mario Cannataro, Bruno Gaeta and Asif Khan, whose toils have results in gathering most of the current content, with all editors reviewing the submissions. Throughout the production phase, we have received invaluable support and guidance as well as milestone reminders from Paula Davies, for which we remain extremely grateful.

Finally we would like to acknowledge all our authors, from around the world, who dedicated their valuable time to share their knowledge and expertise to provide educational guidance for our readers, as well as leave a lasting legacy of their work.

We hope the readers will enjoy this Encyclopedia as much as the editorial team have, in compiling this as an ABC of bioinformatics, suitable for naïve as well as experienced scientists and as an essential reference and invaluable teaching guide for students, post-doctoral scientists, senior scientists, academics in universities and research institutes as well as pharmaceutical, biomedical and biotechnological industries. Nobel laureate Walter Gilbert predicted in 1990 that “In the year 2020 you will be able to go into the drug store, have your DNA sequence read in an hour or so, and given back to you on a compact disk so you can analyze it.” While technology may have already arrived at this milestone, we are confident one of the readers of this Encyclopedia will be ready to extract valuable biological data by computational analysis, resulting in biomedical and therapeutic solutions, using bioinformatics to “measure” health for early diagnosis of “disease.”

References

- Altman, R.B., 1998. A curriculum for bioinformatics: the time is ripe. *Bioinformatics*. 14 (7), 549–550.
Ranganathan, S., 2005. Bioinformatics education—perspectives and challenges. *PLoS Comput Biol* 1 (6), e52.
Tan, T.W., Lim, S.J., Khan, A.M., Ranganathan, S., 2009. A proposed minimum skill set for university graduates to meet the informatics needs and challenges of the “-omics” era. *BMC Genomics*. 10 (Suppl 3), S36.
Ranganathan, S., 2017. *Bioinformatics. Reference Module in Life Sciences*. Oxford: Elsevier.

Shoba Ranganathan

Ecosystem Monitoring Through Predictive Modeling

Sorayya Malek, Cham Hui, Nanyonga Aziida, Song Cheen, Sooh Toh, and Pozi Milow, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Water bodies represent significant accumulations of water on the earth's surface. These accumulations of water can include oceans, seas, lakes, ponds and wetlands. They can be still (or contained) or mobile. Rivers, streams, and canals are some examples of natural water. There are also manmade artificial water bodies such as reservoirs and wetlands. In order to maintain aqueous systems capable of sustaining life forms, preserving water quality is very important. However, water bodies all over the world are facing a few common problems, namely eutrophication, sedimentation, and weed infestation. Eutrophication is the result of water bodies' enrichment, increased growth of microscopic floating plants, algae, and the formation of dense mats of floating plants. Algae have long been used to assess environmental conditions in aquatic habitats throughout the world. Algae respond to a wide range of pollutants. They provide an early caution signal of worsening ecological conditions. They are highly sensitive to changes in their environment and therefore a good indicator. Shifts in abundance of algal species can be used to detect environmental changes, and to indicate trophic status and nutrient problems in the lake. Nutrient stimulation of algal growth made algae part of the problem in the eutrophication of lakes, and trophic status of lakes can be monitored by algal taxa found in them (Mogeeb *et al.*, 2012). Chlorophyll-a has been used to estimate algal biomass in aquatic ecosystems as it is common in most algae. Algal biomass indicates the trophic status of a water body. Chlorophyll- a therefore is an effective indicator for monitoring eutrophication, which is a common problem of lakes and reservoirs all over the world (Malek *et al.*, 2011a,b).

Eutrophication can bring the effects of anoxia, which kills fish and invertebrates and results in release of unpleasant and injurious gases. Algae will bloom and other aquatic plants' growth will be uncontrolled. Species and diversity of plants and animals has decreased in number. Polluted water resources can cause severe effects on human as well as aquatic life. Hence, water quality monitoring is essential in exploitation of aquatic resources conservation. Water quality can be defined as a complex of chemical, physical, microbiological, and radiological properties with respect to its suitability for particular purposes (Kambourova, 2006; Boyd, 2015) or simply the physical, chemical, and biological characteristics of water (Orouji *et al.*, 2013). A water quality variable refers to any physical, chemical, or biological property that influences the suitability of water for natural ecological systems or use by humans (Boyd, 2015).

The quality of water helps in regulating the biotic diversity and biomass, energy, and rate of succession. Various factors can affect water quality parameters that are essential in preserving water quality. Quality of water is divided into three main categories, which are physical, chemical, and biological. Physical measurement involves turbidity, color, odor, and taste. Chemical parameters include the organic and inorganic forms of metals and nutrients. Measurement of ammonia nitrogen and iron are used for organic parameters. Meanwhile, inorganic parameters were measured by pH, iron, sulfate, and chloride. The biological parameters include total and fecal coliform, heterotrophic plate count, chlorophyll concentration, and *Escherichia coli* counts (Ngadiman *et al.*, 2016).

Various factors affect water quality parameters. Surface water temperature for example is affected by urbanization and industrial waste that effects fluctuation in dissolved oxygen (DO) concentration in water. DO is important for living organisms in water. The amount of DO in water bodies is dependent on water temperature, sedimentation, amount of oxygen used by respiring and decaying organisms, and amount of oxygen produced from photosynthesizing plants. Water high in organic matter and nutrients can lead to decrease in DO level as a result of increased microbial activity occurring during the degradation of organic matter. Oxygen is an important parameter as it is essential in the metabolism of all aerobic aquatic organisms. Two other important factors are chemical oxygen demand (COD) and biochemical oxygen demand (BOD). COD and BOD are measurements of the susceptibility to oxidation of the organic and inorganic materials and amount of biodegradable organic matter present in water sample. Water pH can be affected by waste or pollutant from human activities. Salinization process leads to reduction on pure drinking water. Turbidity is defined as clarity of water; the clearer the water the higher the water quality. During the raining season turbidity values of water reservoirs are higher due to the sedimentation process. Nutrients such as nitrates, phosphorus, and ammonia are also important for maintaining water quality and aquatic life.

Databases for water bodies contain a vast amount of information that is not possible to be analyzed using conventional statistical methods. Predictive water quality modeling can help to capture relationships among various factors for water quality monitoring. Predictive modeling can be defined as a process by which a model is created or chosen to try to best predict the probability of an outcome (Geisser, 1993). Kuhn and Johnson (2013) modified this definition "to the process of developing a mathematical tool or model that generates an accurate prediction." Studies to develop predictive models for environmental monitoring began by the use statistical approach. A cursory literature search indicates that computational approach to predictive modeling for environmental models is culminated in a volume that was edited by Racknaegel (2003). The computational approach and procedure involve machine learning (ML), a term that literally means learning how to predict from data. ML can be defined as the capability to acquire knowledge by computers without being explicitly programmed. ML algorithms serve as a common tool for data analysis tasks including data imputation, clustering, classification, and regression. However with the

emergence of visualization using Google Earth for water quality monitoring, the investigation and monitoring of water quality has improved. Application of Google Earth gives intuitive visualization and can reflect the relationship between water quality and geographical position. Google Earth Engine enables water resource scientists to overcome tasks involved with remote sensing data analysis such as locating and downloading the data from various providers, data processing, model applications, results interpretation, and accessing current or updated data.

This article presents an overview of ML approaches, followed by application of these methods to water quality modeling, using three representative case studies. Visualization of the results of water quality monitoring is discussed in the final section.

Introduction to Machine Learning Methods in Water Quality Modeling

ML methods are categorized into supervised and unsupervised learning, and optimization techniques. Supervised learning is where a mapping function is used to learn from input variables to an output variable. It is done using a training dataset where the output variable value is already known. The aim of the mapping functions is to adjust the parameters of a particular model such that the outputs of the system approximate the outputs as closely as possible. Supervised learning can be further categorized into classification and regression. Common methods for supervised learning are artificial neural networks (ANN), random forest (RF), and support vector machine (SVM). In unsupervised learning the output variable is not presented during the learning process. The algorithm discovers on its own structures or patterns in data. Unsupervised learning can be categorized into clustering and association. An example for unsupervised learning is Kohonen self organizing feature maps (SOM). Optimization based methods identify the best set of variables that will minimize a predefined cost function. An example includes genetic algorithms (GAs).

Artificial Neural Network

ANNs gather their knowledge by detecting the patterns and relationships in data and learns (or is trained) through experience. An ANN consists of an input layer of neurons (nodes), one or two or even three hidden layers of neurons, and a final layer of output neurons. Each connection is associated with a numeric number called weight ([Wang et al., 2003](#)). [Liu et al. \(2015\)](#) applied decision tree, back propagation neural networks, radial basis neural network, and logistic regression for water quality modeling. The RBF neural network outperformed all three supervised models for water quality prediction with the highest accuracy. [Palani et al. \(2008\)](#) applied an ANN to predict and forecast quantitative characteristics of water bodies such as salinity, temperature, DO, and chlorophyll-alpha. The ANN model was provided with unseen data and was able to simulate values with high accuracy for the desired locations at which measured data are unavailable yet required for water quality models hence making the ANN algorithm a good model when it comes to monitoring of water quality.

Random Forest

[Breiman et al. \(2001\)](#) defined RF, which add an additional layer of randomness to bagging. In addition to constructing each tree using a different bootstrap sample of the data, RFs change how the classification or regression trees are constructed. In standard trees, each node is split using the best split among all variables and is usually not very sensitive to their values ([Liaw and Wiener, 2002](#)). The efficiency, accuracy, and association between trees in the forest improve with increasing number of selected variables. RF model advantages have a high accuracy of classification and variable importance function. The RF model is also nonparametric, does not require data transformation, and is able to handle a large number of predictors ([Cutler et al., 2007](#)). [Hollister et al. \(2015\)](#) applied RF for regression and RF for classification to determine chlorophyll-a concentration and lake trophic status. Both regression and classification models were developed using a combination of in situ and universally available GIS data and only universally available GIS data resulted in higher performance and better accuracy. This indicates that RF can work well for both regression and classification problem. In another study thirteen ML models such as ANN, SVM, RF, K-nearest neighbors, generalized boosted models were compared for the prediction of groundwater dissolved organic nitrogen by [Wang et al. \(2016\)](#). RF was selected as one of the tree based models with optimal model performance illustrating high generalization ability to different data conditions as well as high interpretability. The study identified factors that affect dissolved organic nitrogen using a limited dataset, which can be cost saving in groundwater monitoring. [Yajima and Derot \(2017\)](#) applied a RF model with a sliding window strategy using multivariate historical water quality records of over 10 years to forecast the concentration of chlorophyll-a in fresh and saline water bodies in Japan. The RF model was able to predict trends of the chlorophyll-a in the water bodies with high accuracy and the study concluded that the most dominant parameters for water quality assessment for chlorophyll-a was BOD, COD, pH, and TN/TP.

Support Vector Machine

SVM is a supervised training algorithm that can be useful for the purpose of classification and regression ([Vapnik, 1998](#)). SVM can be used to analyze data for classification and regression using algorithms and kernels in SVM ([Cortes and Vapnik, 1995](#)). Support vector classification (SVC) also is an algorithm that searches for the optimal separating surface. SVC is outlined first for the linearly

separable case (Burbidge and Buxton, 2001). SVM kernel methods are used when complete separations of the two classes are not possible. Some widely used kernels in SVM are polynomial, quadratic, and radial basis function.

Applied SVC and regression (SVR), which was used in monitoring of surface water quality data by predicting biochemical oxygen demand (BOD) of water. The SVR model predicted water BOD values with high correlation value of 0.952, and RMSE of 1.53, hence providing a tool for the prediction of BOD of surface water quality using set of a few measurable variables. Proposed a smooth support vector machine (SSVM) to predict water quality. SSVM is an algorithm that is used for solving nonlinear function estimation problems. The RMSE reported for aquaculture water quality prediction was very low with a value of 0.0275. This value shows that SSVM is proven to be an effective approach to predict aquaculture water quality.

Fuzzy Logic

Fuzzy logic (FL) is an approach to computing based on “degrees of truth” rather than the usual “true or false” (1 or 0) Boolean logic on which the modern computer is based. Fuzzy reasoning is the process in which fuzzy rules are used to transform input into output and consists of four steps: (1) the input variables are fuzzified, (2) the firing strength of each rule is determined, (3) the consequence of each rule is resolved, and (4) the consequences are aggregated. FL rules consist of a premise and consequence and are believed to be able to capture the reasoning of a human working in an environment with uncertainty and imprecision (Shrestha *et al.*, 1996). Used FL to make their complex reservoir optimization models more appealing to operators, who are reluctant to use procedures they do not fully understand. They use optimization techniques, such as deterministic and stochastic dynamic programming. Water quality index (WQI) is determined using a statistical approach and it is used to communicate information on river water quality. Raman *et al.* (2009) used FL method to calculate WQI value and compared the findings with conventional WQI value. The FL base WQI model generated almost similar results with the conventional method. FL is developed using natural language and it is more understandable for river water classification compared to conventional methods and is tolerant towards imprecise data.

Self-Organizing Feature Maps

SOM, also called Kohonen map, is a popular neural network method based on unsupervised learning (Kohonen, 2013). U-matrix is a visualization technique used in SOM. U-matrix visualizes the distances between the neighboring map units and thus portrays the cluster structure of the map. Darker color with the high values denotes the cluster separator while uniform areas of low values indicate clusters (Stefanovic and Kurasova, 2011). In order to extract the most important parameter in assessing high and low flow period variations in terms of river water quality, Sengorur *et al.* (2015) used SOM and ANN to investigate the nonlinear relationship between the input and output variables in complex data for river and water quality management. Chang *et al.* (2008) applied SOM and K-Means arithmetic model on water quality evaluation. An *et al.* (2016) used a combination of principal component analysis (PCA) for dimension reduction and SOM for pattern identification. The author considered that both PCA and SOM are efficient tools to evaluate and analyze the behavior of multivariable, complex, and nonlinear related surface water quality data to unravel patterns, relationship, and the behaviors of water quality parameters used in the study.

Genetic Algorithm

GAs are a type of optimization algorithm, used to find the optimal solution(s) to a given computational problem that maximizes or minimizes a particular function. GAs represent one branch of the field of study called evolutionary computation (Kinnear, 1994) in that they imitate the biological processes of reproduction and natural selection to solve for the “fittest” solutions. GA was applied to determine plant efficiency to reduce wastewater treatment cost in a river basin to improve water quality with high accuracy (Aras *et al.*, 2007). Lee *et al.* (2016) combined GA algorithm with ANN, called a neurogenetic algorithm (NGA). The NGA was designed to determine the effective number of nodes and the optimal activated functions for the ANN structure. NGA and correlation analysis was used to identify input parameters for chlorophyll-a concentrations prediction in lakes used as drinking water sources. The best NGA model architecture was using double hidden layers and logistic sigmoid function that resulted in a high accuracy in training and testing data in comparison to single hidden layer and linear and hyperbolic tangent function.

Case Study: Application of ML Methods in Water Quality Prediction

Common steps involved in water quality analysis are data preprocessing, data splitting model training and testing, and results evaluation. These are the common steps involved in development in almost all ML methods. Finlay (2014) provided steps for building predictive models, which are outlined as follows:

- (1) Exploring the data landscape
- (2) Sampling and shaping the development sample
- (3) Data preparation (data cleaning)
- (4) Creating derived data

- (5) Understanding the data
- (6) Preliminary variable selection (data reduction)
- (7) Preprocessing (data transformation)
- (8) Model construction (modeling)
- (9) Validation

The steps illustrated by Finlay (2014) and Chapman *et al.* (2000) are illustrated in the following case studies for water quality modeling using ML methods.

Case Study 1: Supervised and Unsupervised Method in Water Quality Modeling

Malek *et al.* (2012) applied supervised and unsupervised learning method to identify the relationship between algal abundance with water quality parameters in a tropical lake. The diatom is an algae type that can be used to distinguish water quality status. Two types of ML methods have been applied in this study: Supervised method (Recurrent artificial neural network (RANN)) and unsupervised method (Kohonen self organizing feature maps (SOM)).

RANN is suitable for water quality model prediction involving dynamic time series data. Applied RANN to determine total nitrogen, phosphorus, and DO at Lake Taihu during water diversion. Shim & Tollner applied RANN and sensitivity analysis to predict and identify significant water quality variables at a composting pond. Sensitivity analysis, an analytical approach, was used to select model inputs for RANN. Malek *et al.* (2012) also applied sensitivity analysis where input variables having greater sensitivity value are deemed important and those with smaller values are discarded. Variable selection or dimension reduction is an important step in water quality modeling. This enables effective water quality management by effectively managing limited resources for water quality management. Fig. 1 below illustrates results obtained from sensitivity analysis against output variable, that is, diatom abundance. It can be seen that input variables such as water temperature, pH, DO, and turbidity play an important role in diatom abundance in a lake. RANN model performance in this study was measured using root mean square error (RMSE) and *r* value, which are standard performance evaluators for regression models.

Unsupervised method SOM was applied in this study to classify and cluster diatom abundance using significant variables identified from sensitivity analysis. The SOM generated in this study was confirmed against the sensitivity curve produced from RANN as illustrated in Fig. 2. SOM cluster map and component plane in Fig. 2 illustrates that diatom is highly abundant at lower temperature less than 30° C; this was conformed with results from sensitivity analysis graph.

SOM has also been used for pattern discovery to extract rules governing alga abundance in terms of propositional IF...else rules (Malek *et al.*, 2009, 2010a,b, 2011a,b, 2012). SOM component planes together with SOM cluster map can be used to extract rules to generate a rule-based system prediction.

Case Study 2: Assessment of Predictive Models for Chlorophyll-a Concentration of a Tropical Lake

Malek *et al.* (2011a,b) assesses four predictive water quality models: FL, RANN, hybrid evolutionary algorithm (HEA) and multiple linear regressions (MLR). RANN, FL, and HEA models are suitable for data that represents a nonlinear relationship such as in water quality data. The application of MLR model in water quality modeling in this study was deemed oversimplified as water pollution or eutrophication process represents complex nonlinear relationships between various water quality variables that are not possible to be explained using simplified models.

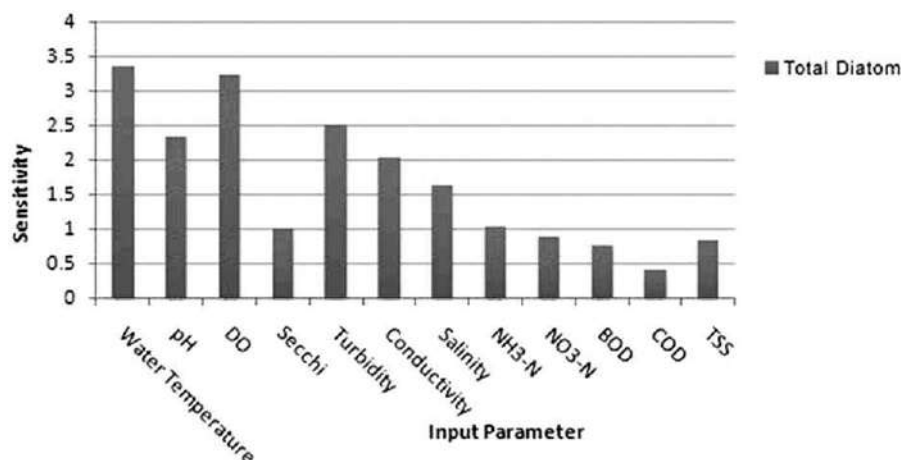


Fig. 1 Sensitivity analysis graph. Reproduced from Malek, S., Salleh, A., Milow, P., Baba, M.S., Sharifah, S., 2012. Applying artificial neural network theory to exploring diatom abundance at tropical Putrajaya Lake, Malaysia. *Journal of Freshwater Ecology* 27 (2), 211–227. doi:10.1080/02705060.2011.635883.

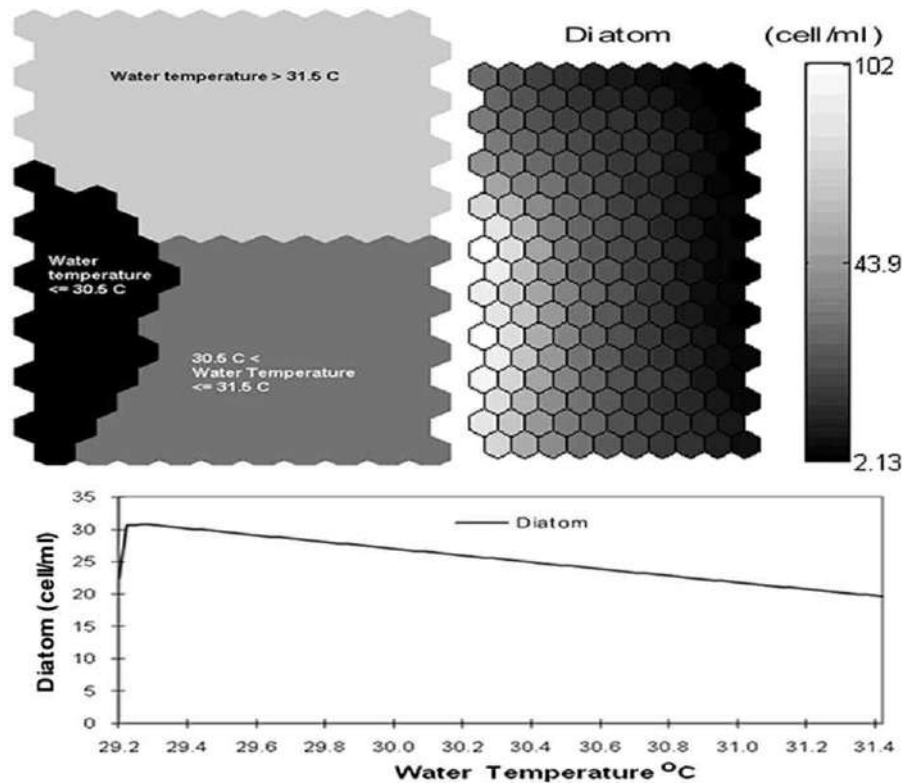


Fig. 2 SOM showing relationship between diatom density and water temperature.

FL based models on the other hand could provide insight into their own operation because the fuzzy rules provide an easily understood and common sense description of the action of the FL system. FL models have been noted to outperformed feed forward ANN in determining algal concentration and RANN in determining chlorophyll-a concentration (Malek *et al.*, 2010b, 2011a,b).

The study by Malek *et al.* (2011a,b) also uses genetic programming to generate the structure of the rule set and GA for parameter optimization for an HEA based model. The HEA model outperformed FL, RANN, and MLR. In raw datasets, some variables have large variation or spread. Normalization of the raw datasets for predictive modeling approach is an important step to ensure all the variables are within the same range. Variable selection is another important step as presenting a large number of variables to ML algorithms may compromise processing speed. Hence reducing dimension of variables used in model development reduces the size of the ML model architecture, which reduces the processing time, decreases the cost of data collection, and improves accuracy of ML models. Reduction in variable dimension for FL based models provides a simpler formulation of inference rules. In formulating FL membership and inference rules the SOM based approach has been applied by the author. SOM is deemed as an effective approach in formulating FL rules. The preferred method of variable selection should comprise prior knowledge and analytical approaches such as sensitivity analysis, backward elimination, stepwise selection, and genetic algorithm based feature selection.

Model performance evaluation adopted in the study was RMSE, r for regression type of ML models and AUROC for binary classification. RMSE is a measure of the average level of prediction error. It indicates the absolute fit of the model to the data or how close the observed data points are to the model's predicted values. The r value is a measure of correlation between the predicted and observed values of the independent variable. The r value indicates agreement between predicted and observed values but it does not indicate the performance of the models. Models are considered reliable when their predicted values correlate with observed values at r value of 0.5 or above. AUROC curve is a graphical plot of sensitivity or true positive rate versus false positive rate. It is based on binary classification task.

The higher performance of FL model reported by the author over RANN and HEA in this study is due to categorizing continuous response (chlorophyll-a concentration) into two classes. AUROC is a better measure for model evaluation than accuracy, which is measured using RMSE and is suitable for characterizing responses that are dichotomous such as lake eutrophication.

ANN models are "black-box" in nature unlike explanatory methods such as FL and HEA. An FL approach according to the author proves to be a practical and successful technique when dealing with semiquantitative knowledge and semiquantitative data. Generation of membership function and inference rules associated with FL model is dependent on knowledge and expertise of a domain expert, which can be overcome by using the HEA approach, which allows discovery of predictive rule set in complex data. The author noted that there are however drawbacks and advantages associated with each model.

Case Study 3: Support Vector Application

Malek *et al.* (2014) applied SVM for classification to predict categorized values of DO from two freshwater lakes in Malaysia. SVM is a ML method based on statistical theory and derived from instruction risk minimization, which can enhance the generalization ability and minimize the upper limit of generalization error. SVM is able to deal with complex and highly nonlinear data. SVM also works well on a limited dataset and has higher accuracy with better prediction for water quality modeling if a kernel method was used compared to ANN. DO concentration in the study was classified into 3 different category (high, medium, and low) based on guidelines by Department of Environment Malaysia. Classification of output for ML methods should be based on valid and verified categories.

Data preprocessing such as normalization and splitting was carried out to increase efficiency of the model. Data was divided into training and testing datasets. Variable selection method was based on linear SVM kernel in this study. Input parameters were ranked using linear kernel in ascending order based on the cross validation errors. Forward elimination method was then used based on ranked variables for SVM model developed using RBF, ANOVA, and Laplace kernel. Optimum parameters that yield the lowest errors, and highest accuracy were determined. Forward elimination method was used for each SVM model to determine most optimum parameters for water quality modeling. In this study SVM using ANOVA kernel resulted more accurate predication results compared with RBF and Laplacian kernels. The parameters were pH, conductivity, and water temperature. SVM models to predict DO using categorized values of DO yields higher prediction accuracy than using continuous DO value.

Visualization and Google Earth

Visualization is important in investigating and monitoring the quality of water. This is because water quality data collection involves capturing geographic features that allow visualization using maps. Visualization displays the relationship between water quality and geographical position and can aid in effective water quality monitoring.

Malek *et al.* (2015) used visualization and ML algorithm HEA for eutrophication modeling and lake management. The integration of computational data mining using HEA and visualization using thematic maps promises practical solutions and better techniques for eutrophication modeling, especially when datasets have complex relationships without clear distinction between various variables. Thematic maps are considered as an effective method of data visualization and are widely used for coastal management, and detection of the toxic algae and eutrophication. The data visualization approach in this study combines thematic cartography with volume visualization. Volume visualization is a set of techniques that present and show an object without mathematical representation. The study uses a choropleth map, which is a type of thematic map. Choropleth maps represent quantitative data such as percentage, density, and average value of an event in a geographical area using color. The rules generated using HEA are integrated with thematic map visualization over time as depicted in Fig. 3. The

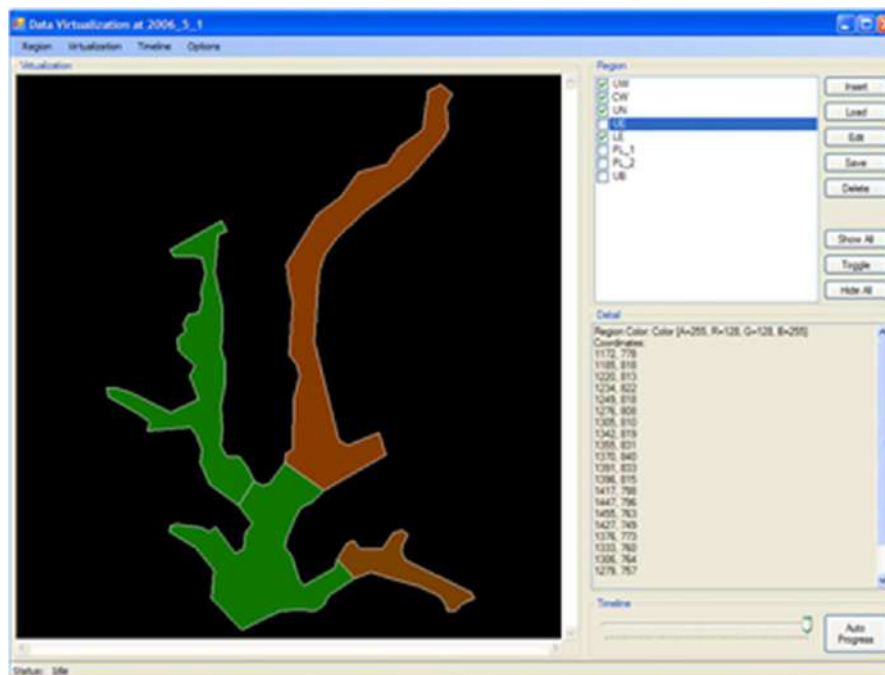


Fig. 3 Illustrates integration thematic map and HEA rules set. Reproduced from Malek, S., Hui, C., Fong, L.C., *et al.*, 2015. Ecological data prediction and visualization system. *Frontiers in Life Science* 8 (4), 387–398. doi:10.1080/21553769.2015.1041167.

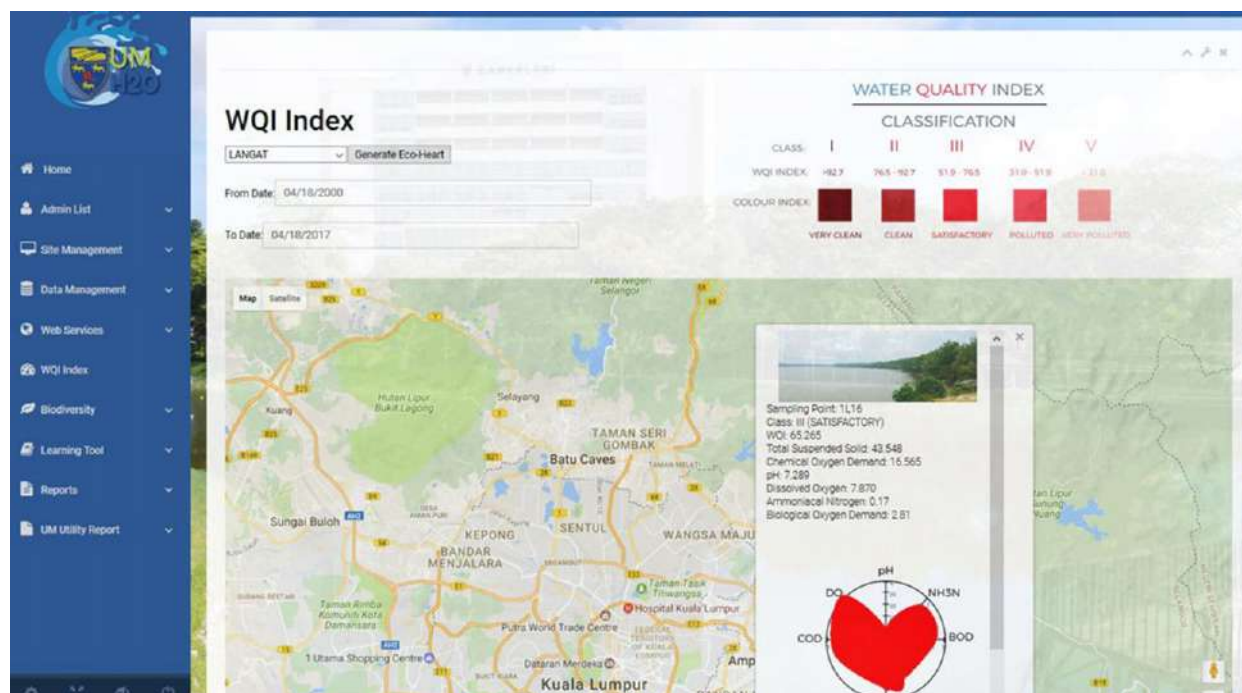


Fig. 4 Google Earth for visualization of WQI index. Source: <http://umlivinglabsystem.com>.

visualization system was evaluated by its capability in representing the output data on a map, and by predicting the abundance of chlorophyte based on other water quality parameters. Rules for predicting chlorophyte abundance had a success rate of almost 90%.

The emergence of Google Earth has added a visualization element for water quality monitoring. Google Earth was used for water quality monitoring using climate and hydrology data in South Africa (Silberbauer and Geldenhuys, 2009). Remote sensing via Google Earth is used to evaluate river ecosystem service, which is applicable to any ecoregion and any river size based on the degree of modification and nature (Large and Gilvear, 2014). Assessing water quality on a large scale using remote sensing models can be costly. Applied Google Earth Engine, a petabyte-scale data archive and remote sensing inquiry atmosphere in order to solve the main problems in designing and implementing water quality models on a large scale. Problems related to large amount of data visualization on Google Earth using the heat map was solved using Hadoop, an open source software, to store and process water quality data set effectively for visualization on Google Earth (Huang *et al.*, 2017). In an unpublished study Google Earth application was used to visualize WQI. Visualization of WQI using varying color and shape allows effective communication to the general public regarding water quality status at a particular location. This can assist in decision making for recreational activities as well as water quality monitoring. The WQI index was represented using series of varying color and shapes to indicate water quality status at a particular site as depicted in Fig. 4.

Conclusion

Water quality forecast and management comprises a substantial amount of uncertainty. There are uncertainties involved in water quality measurement methods as well as models used to predict the water quality and actions required to maintain the water quality status. These models however are necessary to assist water quality personnel involved in monitoring and managing water quality as deteriorating quality of natural water resources such as lakes, streams, and estuaries is a problem faced by mankind. Therefore, monitoring and effectively managing water resources is vital in order to optimize the quality of water. The problems associated with water contamination can be handled if water quality is predicted early using predictive models. This concern has been addressed in many prior studies, however, extensive work is still required in terms of applicability, efficiency, and reliability of existing water quality modeling. The methods presented here can be applied to other ecosystems, in order to sustain a healthy environment.

Acknowledgment

The authors would like to thank university of Malaya research grant LL023-16SUS for allowing to use unpublished materials from "see Relevant Website section".

See also: Natural Language Processing Approaches in Bioinformatics

References

- An, Y., Zou, Z., Li, R., 2016. Descriptive characteristics of surface water quality in Hong Kong by a self-organising map. *International Journal of Environmental Research and Public Health* 13 (1), 115. doi:10.3390/ijerph13010115.
- Aras, E., Toğan, V., Berkun, B., 2007. River water quality management model using genetic algorithm. *Environmental Fluid Mechanics* 7, 439–450. doi:10.1007/s10652-007-9037-4.
- Boyd, C.E., 2015. *Water Quality: An Introduction*, second ed. Heidelberg, New York, Dordrecht, London: Springer Cham.
- Breiman, L., Last, M., Rice, J., 2001. Random forests: Finding quasars. *Statistical Challenges in Astronomy*. 243–254. doi:10.1007/0-387-21529-8_16.
- Burbidge, R., Buxton, B., 2001. An introduction to support vector machines for data mining. *Keynote Papers, Young OR* 12, pp. 3–15.
- Chang, K., Gao, J., Yuan, Y., Li, N., 2008. Research on water quality comprehensive evaluation index for water supply network using SOM. In: *Proceeding of the International Symposium on Information Science and Engineering*. doi:10.1109/isis.2008.103.
- Cortes, C., Vapnik, V., 1995. Support-vector networks. *Machine Learning* 20 (3), 273–297.
- Cutler, D.R., Edwards Jr, T.C., Beard, K.H., et al., 2007. Random forests for classification in ecology. *Ecology* 88, 2783–2792.
- Finlay, S., 2014. *Predictive Analytics, Data Mining and Big Data: Myths, Misconceptions and Methods*. Basingstoke: Palgrave Macmillan.
- Geisser, S., 1993. *Predictive inference*, vol. 55.
- Hollister, J.W., Milstead, W.B., Kreakie, B.J., 2015. Modelling lake trophic state: A random forest approach. doi:10.7287/peerj.preprints.1319.
- Huang, W., Zhao, X., Han, Y., Du, W., Cheng, Y., 2017. Characterizing water quality monitoring visualization with Hadoop and Google Maps. *Water Practice and Technology* 12 (4), 882–893. doi:10.2166/wpt.2017.093.
- Kambourova, V., 2006. Potential water quality problems posed by intentional/accidental interventions. *NATO Security Through Science Series Management of Intentional and Accidental Water Pollution*, pp. 1–10. doi:10.1007/1-4020-4800-9_1.
- Kinney, K.E., 1994. Fitness landscapes and difficulty in genetic programming. In: *Proceedings of the First IEEE Conference on Evolutionary Computation*, 1994. *IEEE World Congress on Computational Intelligence*, pp. 142–147. IEEE.
- Kohonen, T., 2013. Essentials of the self-organizing map. *Neural Networks* 3, 52–65.
- Kuhn, M., Johnson, K., 2013. A summary of grant application models. *Applied Predictive Modeling*. 415–418. doi:10.1007/978-1-4614-6849-3_15.
- Large, A.R., Gilvear, D.J., 2014. Using google earth, a virtual-globe imaging platform, for ecosystem services-based river assessment. *River Research and Applications* 31 (4), 406–421. doi:10.1002/ra.2798.
- Lee, G., Bae, J., Lee, S., Jang, M., Park, H., 2016. Monthly chlorophyll-a prediction using neuro-genetic algorithm for water quality management in Lakes. *Desalination and Water Treatment* 57 (55), 26783–26791. doi:10.1080/19443994.2016.1190107.
- Liaw, A., Wiener, M., 2002. Classification and regression by randomForest. *R News* 2 (3), 18–22.
- Liu, B., Wan, X., Wu, X., Li, Y., Zhu, H., 2015. Application of decision tree and neural network algorithm in water quality assessment forecast. In: *Proceedings of the 2015 International Conference on Material Science and Applications*. doi:10.2991/icmsa-15.2015.137.
- Malek, S., Ahmad, S.S., Singh, S.K., Milow, P., Salleh, A., 2011b. Assessment of predictive models for chlorophyll-a concentration of a tropical lake. *BMC Bioinformatics* 12 (Suppl. 13), doi:10.1186/1471-2105-12-s13-s12.
- Malek, S., Hui, C., Fong, L.C., et al., 2015. Ecological data prediction and visualization system. *Frontiers in Life Science* 8 (4), 387–398. doi:10.1080/21553769.2015.1041167.
- Malek, S., Mogeeb, M., Ahmad, S.M., 2014. Dissolved oxygen prediction using support vector machine. *International Journal of Bioengineering and Life Sciences* 8, 1.
- Malek, S., Salleh, A., Ahmad, S.M., 2009. Analysis of algal growth using Kohonen self organizing feature map (SOM) and its prediction using rule based expert system. In: *Proceeding of the 2009 International Conference on Information Management and Engineering*. doi:10.1109/icime.2009.63.
- Malek, S., Salleh, A., Baba, M.S., 2010a. Analysis of selected algal growth (Pyrrophyta) in tropical lake using Kohonen self organizing feature map (SOM) and its prediction using rule based system. In: *Proceedings of the International Conference and Workshop on Emerging Trends in Technology – ICWET 10*. doi:10.1145/1741906.1742083.
- Malek, S., Salleh, A., Baba, M.S., 2010b. A comparison between neural network based and fuzzy logic models for chlorophyll-a estimation. In: *Proceedings of the 2010s International Conference on Computer Engineering and Applications*. doi:10.1109/iccea.2010.217.
- Malek, S., Salleh, A., Baba, M.S., Ahmad, S.M., 2011a. A self organizing map (SOM) guided rule based system for freshwater tropical algal analysis and prediction. *Scientific Research and Essays* 6 (25), 5279–5284. doi:10.5897/SRE10.866. Available online at: <http://www.academicjournals.org/SRE>.
- Malek, S., Salleh, A., Milow, P., Baba, M.S., Sharifah, S., 2012. Applying artificial neural network theory to exploring diatom abundance at tropical Putrajaya Lake, Malaysia. *Journal of Freshwater Ecology* 27 (2), 211–227. doi:10.1080/02705060.2011.635883.
- Mogeeb, A.A., Hayat, Sorayya, Pozi, Aishah, 2012. A preliminary study on automated freshwater algae recognition and classification system. *BMC Bioinformatics* 2012–13 (Suppl. 17), S25.
- Ngadiman, N., Bahari, N.I., Kaamin, M., et al., 2016. Water quality of hills water, supply water and RO water machine at Ulu Yam Selangor. *IOP Conference Series: Materials Science and Engineering*, vol. 136. IOP Publishing. p. 012081. No. 1.
- Orouji, H., Bozorg Haddad, O., Fallah-Mehdipour, E., Mariño, M.A., 2013. Modeling of water quality parameters using data-driven models. *Journal of Environmental Engineering* 139, 947–957.
- Palani, S., Liong, S., Tkach, P., 2008. An ANN application for water quality forecasting. *Marine Pollution Bulletin* 56 (9), 1586–1597. doi:10.1016/j.marpolbul.2008.05.021.
- Raman, B.V., Bouwmeester, R., Mohan, S., 2009. Fuzzy logic water quality index and importance of water quality parameters. *Air, Soil and Water Research* 2. doi:10.4137/aswr.s215.
- Sengorur, B., Koklu, R., Ates, A., 2015. Water quality assessment using artificial intelligence techniques: SOM and ANN – A case study of Melen River Turkey. *Water Quality, Exposure and Health* 7 (4), 469–490. doi:10.1007/s12403-015-0163-9.
- Shrestha, B.P., Duckstein, L., Stakhiv, E.Z., 1996. Fuzzy rule-based modeling of reservoir operation. *Journal of Water Resources Planning and Management* 122 (4), 262–269. doi:10.1061/(asce)0733-9496(1996)122:4(262).
- Silberbauer, M., Geldenhuys, W., 2009. Google Earth-a spatial interface for SA water resource data. *PositionIT*, pp. 42–47.
- Stefanovic, P., Kurasova, O., 2011. Visual analysis of self-organizing maps. *Nonlinear Analysis: Modelling and Control* 16 (4), 488–504.
- Vapnik, V., 1998. *Statistical Learning Theory*. New York: Wiley, p. 1998.
- Wang, H., Fan, W., Yu, P.S., Han, J., 2003. Mining concept-drifting data streams using ensemble classifiers. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge discovery and data mining*, pp. 226–235. ACM.
- Wang, B., Oldham, C., Hipsley, M.R., 2016. Comparison of machine learning techniques and variables for groundwater dissolved organic nitrogen prediction in an Urban Area. *Procedia Engineering* 154, 1176–1184. doi:10.1016/j.proeng.2016.07.527.
- Yajima, H., Derot, J., 2017. Application of the Random Forest model for chlorophyll-a forecasts in fresh and brackish water bodies in Japan, using multivariate long-term databases. *Journal of Hydroinformatics* 20 (1), 206–220. doi:10.2166/hydro.2017.010.

Relevant Website

<http://umlivinglabssystem.com>
LivingLab.

Large Scale Ecological Modeling With Viruses: A Review

Vijayaraghava S Sundararajan, Ministry of Environmental Agency, Singapore; Bioclues.org, Hyderabad, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Ecology is the study of organisms and their environments. Viruses are not usually categorized as free living organisms and merely treated as part of the environment. From late 1980s, several researchers started validating through molecular, experimental, micrographic, and model-based research on how viruses should be considered as ecological actors, and it was established that viruses are regulators of planetary biogeochemistry. A study by OMalley (2016) suggests that viruses should be considered as ecological actors that are comparably equal to organismal actors (OMalley, 2016).

In this review article, we will describe previous research studies in modeling large scale data with Dengue virus, in and around Singapore. These research methodologies, statistical analysis, and steps will give researchers an idea in how to deal with the large scale virus data and create models on virus locally and internationally to take appropriate steps to prevent the diseases spread.

1. Dengue viruses (DENV) cause 500,000 infections (dengue hemorrhagic fever, DHF, shock syndrome, and DSS) (WHO, 2009). DENV is a single stranded positive sense RNA virus with an 11.8 kb genome flanked by two untranslated regions, a single coding region, the precursor of membrane (prM), and seven nonstructural proteins (Chambers *et al.*, 1990). Being a RNA virus, DENV undergoes a rapid evolutionary process (Holmes and Burch, 2000; Holmes and Twiddy, 2003) and selects strains with enhanced adaptation, resulting in mutant variants that differ in their ability to spread and cause disease (Holmes and Burch, 2000; Twiddy *et al.*, 2002) and all four DENV serotypes are capable of causing severe dengue clinically (Whitehorn and Simmons, 2011). Previous studies implicated host genetics (Coffey *et al.*, 2009; Khor *et al.*, 2011) immune response (Green and Rothman, 2006), infection status, and viremia levels (Wang *et al.*, 2006; Vaughn *et al.*, 2000) as predisposing factors for severe dengue. Mild to severe dengue outbreaks subsequent to emergence of a DENV-2 genotype of Asian origin in Latin and Central America (Rico-Hesse *et al.*, 1997a) were seen and disease severity could be even strain-dependent (Holmes and Burch, 2000). Viral genetic differences correlate with clinical severity (Pandey and Igarashi, 2000; Tuiskunen *et al.*, 2011; Chen *et al.*, 2008; Leitmeyer *et al.*, 1999; Guzman *et al.*, 1995) and virus attenuation (Whitehead *et al.*, 2003) to support the strain-dependent concept of virulence.
2. Dengue reemergence is increasingly associated with a shift in epidemiology to older cohorts with different clinical manifestations and severity compared with very young children (Chen and Vasilakis, 2011). The dengue serotypes circulating all year round in Singapore are DENV-1, DENV-2, and DENV-3 with sporadic reports of DENV-4 (Lee *et al.*, 2010). There were major outbreaks in 2005 by DENV-1 and by DENV-2 in 2007 (Ler *et al.*, 2011). DENV-1 and DENV-2 are the main circulating serotypes out of 4 types (Ministry of Health, 2012). Infections with different serotypes may cause nearly identical clinical syndrome (Halstead, 2008), but some differences in clinical manifestations have been reported, although conclusions are usually based on limited serotype comparisons and small sample sizes (Chan *et al.*, 2009; Kumaria, 2009). The exception is a recent large cross-sectional study from the Americas comprising 1716 children and adults (Halsey *et al.*, 2012). Multiple factors have been suggested to contribute to severe dengue (SD) such as secondary infections, age, viral load, as well infecting serotype and genotype (Burke *et al.*, 1988; Guzman and Kouri, 2003; Rico-Hesse *et al.*, 1997b; Sangkawibha *et al.*, 1984). Infection with secondary DENV-2 is more likely to result in severe disease compared with other serotypes (Balmaseda *et al.*, 2006; Fried *et al.*, 2010; Nisalak *et al.*, 2003; Vaughn *et al.*, 2000). Primary DENV-1 cases were more overt whereas primary DENV-2 and DENV-3 cases were usually silent (Guzman *et al.*, 2012, 2000). DENV-2 genotype in is the Asian genotype rather than the Cosmopolitan genotype circulating in Singapore (Rabaa *et al.*, 2010; Zhang *et al.*, 2006) and extensive diversity resulting in epidemic potential (Rico-Hesse, 2003).
3. The “United in Tackling Epidemic Dengue (UNITEDengue)” portal was launched on August 28, 2012, to initiate cross-border dengue case and virus surveillance (Ng, 2011). In Singapore, the economic impact of dengue during 2000–2009 was estimated about US\$1.15 billion (Carrasco *et al.*, 2011) and in Klang Valley, Malaysia during 2005–2009 is about US\$56 million (Shepard *et al.*, 2012; Suaya *et al.*, 2009). In 2013, Malaysia reported a total of 43,348 cases (peak in 2008 with 49,335) and Singapore reported a historical high of 22,170 cases (Anon, 2013; Anon, 2014a), which was almost 50% more than the worst dengue epidemic in 2005 with 14,209 cases, over previous years (2008–2012).
4. In the Western Pacific Region (such as, ...), dengue is the main viral infection that occurs yearly in multiple countries due to complex interplay of virus, vector, and host biology, as well as climatic and socioeconomic factors (Arima *et al.*, 2015; Bhatt *et al.*, 2013; Lee *et al.*, 2012; Ritchie, 2014; Banu *et al.*, 2011; Wilder-Smith and Gubler, 2008). A 2014 study showed that DENV-serotype switches from previous years (Governments of Malaysia and Singapore, 2014; Anon, 2014b) and secondary heterotypic infection is believed because larger numbers equal severe dengue cases (Guzman *et al.*, 2013). External quality assessment (EQA) is used to compare laboratory performance and reveal potential problems associated with diagnostic kits or procedures (Anon, 2011a). The WHO Regional Office for the Western Pacific launched an EQA for dengue diagnostics testing in 2013, under the Asia Pacific Strategy for Emerging Diseases (APSED) 2010 (Anon, 2012a).
5. *Aedes aegypti* vectors acquire the dengue virus when they feed on viremia hosts (the dengue infected human patient) (Kramer and Ebel, 2003) and the role of mosquito vector in the transmission of dengue fever was first recognized by Bancroft (1906). In 1906, feeding

mosquitoes on dengue patients was further studied by Cleland *et al.*, (1919, 1918). Comprehensive human to mosquito transmission studies were conducted in the Philippines (Siler *et al.*, 1926; Simmons *et al.*, 1931), and several epidemiologically important observations (Nguyet *et al.*, 2013) in Vietnam involve feeding mosquitoes with vertebrate blood spiked with cell cultured virus, using a device, such as a “double-jacketed” glass cylinder (Rutledge *et al.*, 1964) or a Hemotek membrane feeding system (Diehlmann, 1999). Earlier studies may not accurately mimic that of the natural setting (Rodhain and Rosen, 1997) and infection rates were lower in mosquitoes (Jupp, 1976; Meyer *et al.*, 1983), and use of virus stock that had been frozen and thawed, or phenotypic virus adaptation due to numerous passages in cell culture (Richards *et al.*, 2007; Chen *et al.*, 2003; Miller, 1987).

6. In the Western Pacific Region during the 2014 outbreaks, there were 1513 dengue cases in the Solomon Islands (Anon, 2014c) 45,171 cases in China, and 108,698 cases in Malaysia (Anon, 2015), and Japan reported its first autochthonous outbreak in over 70 years (Arima *et al.*, 2014; Kutsuna *et al.*, 2015). DENV-serotype 3 was found to be circulating in the Pacific after an absence of 18 years (Cao-Lormeau *et al.*, 2014). CHIKV has probably had an unappreciated circulation in the region due to its cocirculation with DENV (Weaver and Lecuit, 2015; Horwood *et al.*, 2013). Zika virus, a flavivirus that was detected in Asia in the 1960s but has recently emerged in the Pacific and the Americas (Kindhauser *et al.*, 2016), is linked to clusters of microcephaly and other neurological disorders that WHO declared on 1-Feb-2016 as of international concern (Anon, 2016). *A. aegypti* and *A. albopictus* mosquito vectors require different control strategies (Yap *et al.*, 2010) and clinicians require specialized training to treat severe dengue cases (Anon, 2012b). Genotype data is useful for tracking the movement of viruses and for risk assessment (Hapuarachchi *et al.*, 2016), and point-of-care tests for dengue diagnosis (Prat *et al.*, 2014) was introduced. EQA in the WHO Western Pacific Region (Pok *et al.*, 2015b) was studied and APSED (Anon, 2011b) was initiated.

Research Studies

Clinical Outcome and Genetic Differences Within a Monophyletic Dengue Virus Type 2 Population (Hapuarachchi *et al.*, 2015)

The new findings showed that primary and secondary infections that progressed to DHF and DSS were likely due to host rather than virus factors.

Data: DENV-2 with 89 whole genomes were obtained from patients diagnosed with dengue fever (DF, $n=58$), dengue hemorrhagic fever (DHF, $n=30$) and dengue shock syndrome (DSS, $n=1$) during July 2010 to January 2013 in Singapore. 70% of DENV-2 serotyped cases in Singapore (2007–2011) (Han *et al.*, 2012) were noted. Primary and secondary infection cases (Gan *et al.*, 2014) were based on serum IgM and IgG in patient sera, by using Panbio Dengue Capture IgM ELISA and Dengue Capture IgG, and Indirect IgG ELISA kits.

Method: Viral RNA was extracted from patient sera/plasma by using the QIAamp Viral RNA Mini Kit and crossing point values from the real-time PCR assay results were used as a surrogate indicator of virus titers (Lai *et al.*, 2007). DENV was isolated from sera/plasma using the *Ae. albopictus* clone C6/36 mosquito cell line (ATCC CRL-1660). The presence of DENV in cell supernatants was confirmed by an immunofluorescent assay (IFA) using DENV group-specific and serotype-specific monoclonal antibodies derived from hybridoma cultures (ATCC: HB112, HB114, HB47, HB46, HB49, and HB48).

Packages: Contiguous sequences were assembled using the Lasergene package version 8.0 and were aligned in the BioEdit Sequence Alignment Editor version 7.0.9.0 (Hall, 1999). The phylogeny of study sequences was inferred by the maximum likelihood (ML) method based on the general time reversible substitution model and gamma distributed rates with invariant sites in MEGA 6.06 (Tamura *et al.*, 2013). The median joining network was constructed using the whole genome sequences in Network version 4.6.1.2 (Bandelt *et al.*, 1999). Genome-wide dN/dS ratios were computed using Hyphy package (Pond *et al.*, 2005). The single likelihood ancestor counting, fixed effects likelihood, internal fixed effects likelihood, and the mixed effect model of evolution methods (Murrell *et al.*, 2012) were used to detect the molecular signatures of selection within an alignment of 272 complete coding sequences (Asian, American Asian, and American genotypes of DENV-2) consisting of all study isolates ($n=89$) and those obtained from the GenBank database. Significance levels were set at $p<0.05$ and the secondary structures of the variable region of 3' UTR and complete 5' UTR of wild type (NS2A-V83) and mutant (NS2A-V83I) variants of DENV-2 cosmopolitan clade III were predicted using the mfold web server under standard conditions (37°C) (Zuker, 2003). The statistical significance, using R software (RCoreTeam, 2013) on association of primary and secondary infections within the DHF and DSS categories was calculated using the exact binomial test. Probability values less than 0.05 were considered statistically significant.

Conclusions: 89 patients were modeled as DF=58 and DHF=30, DSS=1. 45 patients had secondary dengue infections and 27 patients had primary infections and the primary/secondary status of 17 patients could not be classified. Findings indicated that subtle genetic differences in a highly homogenous, monotypic DENV-2 population potentially modified the clinical outcome of study subjects, regardless of primary and secondary infection status.

Dengue Serotype-Specific Differences in Clinical Manifestation, Laboratory Parameters, and Risk of Severe Disease in Adults, Singapore (Yung *et al.*, 2015)

Serotype-specific features and disease severity on patients from April 2005 to December 2011 were studied on dengue hemorrhagic fever (DHF) and severe dengue (SD). 469 dengue-confirmed patients comprising 22.0% dengue virus serotype 1 (DENV-1), 57.1% DENV-2, 17.1% DENV-3, and 3.8% DENV-4 were studied.

Patients: The study included dengue cohorts recruited from two service settings: Primary care clinics and Singapore's Communicable Disease Centre, an infectious disease center that provides an outpatient walk-in service supporting Tan Tock Seng Hospital, a university teaching hospital with 1500 beds.

Methods: Phylogenetic analysis on DENV sequences was conducted by using the maximum-likelihood method as implemented in PAUP* software and compared with sequence data obtained from GenBank. Serology testing was carried out using: Platelia™ NS1 ELISA, Panbio® Dengue IgG Indirect, IgG Capture, and IgM Capture ELISAs.

Data: Headache, drowsiness, eye pain, muscle pain, joint pain, rash, bleeding, anorexia, nausea, vomiting, red eyes (conjunctival injection), and abdominal pain are used as clinical manifestations in the analysis. The WHO (2009) warning signs of lethargy/drowsiness, severe abdominal pain, and mucosal bleeding were assessed as one variable defined as patients who fulfilled any of the three warning signs (WS). Plasma viral RNA level, temperature, systolic blood pressure, pulse rate, hematocrit, platelet, and leukocyte count at the first day of presentation were analyzed. Between April 2005 and December 2011, a total of 3468 patients were enrolled into study cohorts.

Analysis: For descriptive analyses, number and percentage were used for categorical variables; median and range were used for continuous variables. The χ^2 test was used to compare univariate categorical data whereas Fisher's exact test was used if expected cell sizes <5. Adjustment was done for clinically relevant potential confounders: Age, gender, year of infection, recruitment site, fever day, and primary/secondary infection status unless stated otherwise. All analysis was done using R and in-house SAS software.

Results: Dengue was confirmed in 617 (18.2%) patients. Serotype information was available for 469 (76%) of dengue-confirmed patients comprising 103 (22.0%) DENV-1, 268 (57.1%) DENV-2, 80 (17.1%) DENV-3, and 18 (3.8%) DENV-4.

Conclusion: DENV-1 infection may be more severe compared with DENV-2 infection. A findings model that included a prospective adult dengue cohort found that DENV-1 cases were more likely to present with red eyes whereas absence of red eyes but presence of joint pain and lower platelet count was associated with DENV-2 cases. The differences in severity may be attributable to variations in plasma viral RNA levels between serotypes. At a molecular level, our findings may be associated with DENV-1 genotype 1 and DENV-2 cosmopolitan genotype.

Dengue Outbreaks in Singapore and Malaysia Caused by Different Viral Strains (Ng *et al.*, 2013)

A consortium of ASEAN countries formulated UNITEDengue with 14,079 circulating DENV in a cross-border surveillance program revealed that in the 2013 outbreaks in Singapore and Malaysia, the predominant virus in Singapore switched from DENV2 to DENV1, with DENV2 becoming predominant in neighboring Malaysia. Dominance of DENV2 was most evident in the southern states where higher fatality rates were observed. A shared secured web-portal UNITEDengue (see "Relevant Website section") features weekly incidence, monthly Dengue virus (DENV), serotype distribution, and sequences of the envelope (E) gene of viruses.

DENV serotype surveillance revealed that outbreaks in both countries were associated with a switch in predominant serotypes. In Malaysia, DENV2 overtook a preexisting DENV3 and DENV4 dominance pattern in 2013 (Fig. 1(A) and (B)). DENV2 was most dominant in the southernmost states of Johor and Malacca (Fig. 1(C)), where the proportion of DENV2 rose to 70%–90% after August 2013. Incidentally, the fatality rates of 0.5% in those two states were unusually high when compared with those of the whole of Malaysia and Selangor (0.18% and 0.08%, respectively), where surveillance protocols were the same. The association of DENV2 dominance with high fatality is of concern although the reasons remain unclear. Hypothetically, it could be because of the inherent virulence of DENV2 strain or the antibody-dependent enhancement, which causes sequential infection of DENV1 (or DENV3) followed by DENV2 to be more severe.

Envelope gene-based virus surveillance: DENV1 strains circulating in Singapore during the 2013 epidemic were first detected in November 2012 and belonged to genotype III. From April 2013 onward, more than 50% of all viruses sampled in Singapore clustered in this clade. Interestingly, only 42 (12%) of 349 DENV1 samples sequenced in Malaysia belonged to this clade. The predominant DENV2 strain found in Johor and Malacca belonged to the cosmopolitan genotype (named as cladeIb) and was distinct from those that were predominant in the northern part of Malaysia. Though closely related to the cosmopolitan virus that caused Singapore's outbreak in 2007, and to strains that continue to circulate in Singapore, DENV-2 cladeIb in Johor and Malacca clustered distinctively (supported by a bootstrap value of 98%) from DENV-2 strains in Singapore. Out of five dengue-related death cases in Johor, four were due to cladeIb viruses. Close monitoring in Singapore initially detected sporadic cases (40 (11%) out of 363 DENV2 sequenced) due to cladeIb, and seven of those 40 sequences were identical to the cladeIb strains in Johor. Incidentally, the first death case in Singapore in early 2014 was due to a virus from the same clade.

First Round of External Quality Assessment of Dengue Diagnostics in the World Health Organization Western Pacific Region, 2013 (Pok *et al.*, 2015a)

Nineteen national-level public health laboratories from 18 countries and areas (two in Vietnam) in the WHO Western Pacific Region where dengue is endemic or where imported cases have been detected were invited to participate in the EQA. All 19 agreed and the EQA panel was dispatched to these laboratories between May and July 2013, and the labs performed routine dengue diagnostic assays on a proficiency testing panel consisting of two modules.

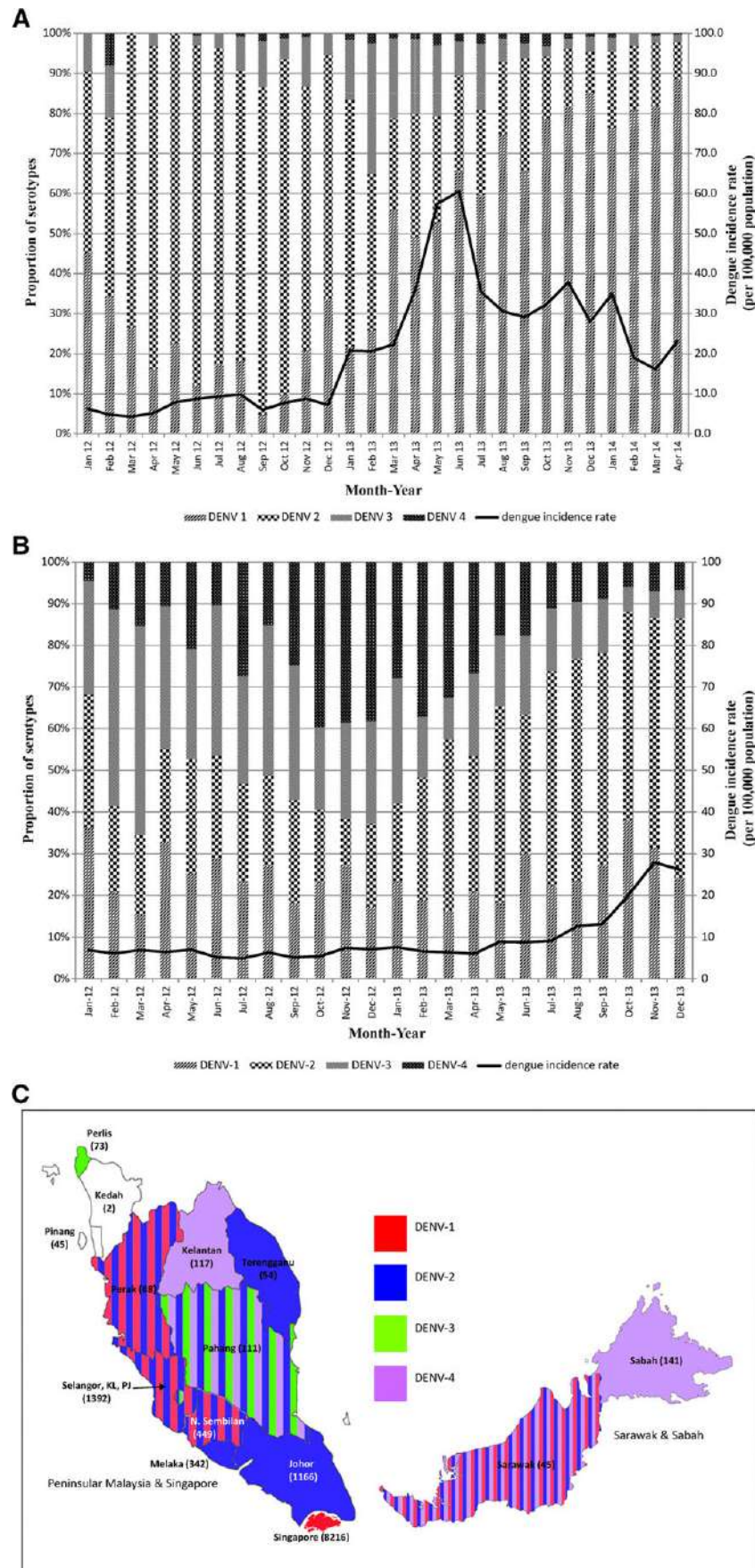


Fig. 1 (A) Monthly distribution of DENV serotypes in Singapore 2012–2014. (B) Monthly distribution of DENV serotypes in Malaysia 2012–2013. Predominant serotype switched from DENV2 to DENV1 in Singapore, while it switched from DENV3/4 to DENV2 in Malaysia. (C) Spatial distribution of predominant DENV serotypes in Malaysia and Singapore (2013). Reproduced from Ng, L.-C., Chem, Y.-K., Koo, C., *et al.*, 2013. Dengue outbreaks in Singapore and Malaysia caused by different viral strains. *American Journal of Tropical Medicine and Hygiene* 92, 1150–1155.

Module A: Samples A2013-V01 and A2013-V03 contained at least 10^6 RNA copies/mL of in vitro-cultured DENV of different serotypes: DENV-1 genotype III and DENV-2 cosmopolitan clade II, deposited in Genbank with accession numbers KP685233 and KP685236, respectively.

Module B: Samples B2013-S01 and B2013-S02 were split serum samples from a convalescent dengue patient included to assess reproducibility of testing by the participating laboratories. These samples were confirmed by PRNT to contain neutralizing antibodies against DENV 1–4 ($> 1:1000$) and confirmed DENV-negative.

Analysis: In Module A, two points each were awarded for the correct detection of DENV by RT-PCR or NS1 assay and accurate serotyping of DENV. In Module B, two points each were awarded for the correct detection of anti-DENV IgG and IgM antibodies. The final score was the proportion of points earned out of the possible awardable points. Accuracy for each assay (e.g., serotyping) was defined as the proportion of laboratories scoring 100% for that assay. Quantitative data (RT-PCR cycle threshold values and ELISA values) submitted were used for reference and for assessing reproducibility of laboratory results. For ELISA assays, the percentage coefficient of variation was calculated from values recorded in triplicate runs to evaluate the reproducibility of results.

Module A: (Viral RNA and NS1 antigen): For laboratories using RT-PCR in Module A, the majority (11/16) used the QIAmp Viral RNA Mini Kit for extraction and purification of DENV RNA and commercial kits to perform RT-PCR. More than half (56.3%) used real-time RT-PCR, while the remainder used conventional RT-PCR methods. The laboratory reporting equivocal results for the negative sample in Module A used the real-time methodology.

Module B: (Serology): All 18 laboratories that requested Module B chose the ELISA methodology to detect anti-DENV IgM. Half used the Panbio Dengue IgM Capture ELISA with two also using a rapid diagnostic test. The two laboratories detecting anti-DENV IgM in only one of the split samples used an in-house IgM MAC-ELISA protocol and a commercial Dengue IgM ELISA kit, respectively. This first round of EQA in the Western Pacific Region strengthens the regional public health laboratory system for detecting emerging infectious diseases, in line with APSED (2010).

Results: Module A was performed by 16 laboratories with reverse transcriptase polymerase chain reaction (RT-PCR) for both RNA and serotype detection. Of these, 15 had correct results for RNA detection and all 16 correctly serotyped the viruses. All nine laboratories performing NS1 antigen detection obtained the correct results. Sixteen of the 18 laboratories using IgM assays in Module B obtained the correct results as did the 13 laboratories that performed IgG assays.

Membrane Feeding of Dengue Patient's Blood as a Substitute for Direct Skin Feeding in Studying Aedes-Dengue Virus Interaction (Tan *et al.*, 2016)

Understanding the interaction between Aedes vectors and DENV has significant implications in determining the transmission dynamics of dengue. The absence of an animal model and ethical concerns regarding direct feeding of mosquitoes on patients has resulted in most infection studies using blood meals spiked with laboratory-cultured DENV.

Patients: The patient criteria were: (1) Adult, ≥ 21 years of age, (2) ≤ 5 days of fever, and (3) positive for DENV by point-of-care dengue NS1 Ag rapid test kit (Standard Diagnostic Inc., Korea) and written consent was obtained. A total of 26 dengue patients, 20 (80%) were found to be infected with DENV-1, three (12%) were infected with DENV-3, and two (8%) with DENV-2. The viremia level (DENV Log₁₀ RNA copies/ml) of patients ranged from 4.75 to 9.41 (DENV-1), 3.73 to 5.6 (DENV-2), and 3.69 to 8.69 (DENV-3).

Methods: The number of RNA copies in patient's serum was measured using a TaqMan® one-step qRT-PCR assay targeting a highly conserved 3'UTR region of DENVs. Oligonucleotide sequences used were DENVF2: 5'-AAACAGCATATTGACGCTGGGA-3' and DENVR3: 5'-GGCGYTCTGTGCTGGAATGATG-3', with a probe sequence of 5'-FAM-AGACCAGAGATCCTGCTGTCTC-MGB-3'. PCR reactions were carried out using the TaqMan® Fast Virus 1-step Master Mix. Larvae were reared in 25 cm $\times$ 30 cm $\times$ 9 cm enamel pans containing 800 mL of water and fed with Plecomin® fish food (Tetra, Germany). Pupae were placed in 30 cm $\times$ 30 cm $\times$ 30 cm (H $\times$ W $\times$ L) cages. 17–34 mosquitoes (4–6 days old) were transferred into paper cups covered with net and were starved for at least 24 h before exposure to dengue patients. After the EDTA blood was collected, the cup was placed on patient's forearm and the mosquitoes were exposed to feed through the net for 15 min. Total RNA was amplified from mosquito midguts using the QIAamp Viral Mini. The number of RNA copies in mosquito midguts was estimated using the same TaqMan® one-step qRT-PCR assay.

Feeding rates: Overall, DSFA consistently resulted in higher mosquito feeding rates when compared to membrane feeding. In 11 out of 25 feeding events, a significantly higher number of mosquitoes were found to be fully engorged when they fed directly on a patient's arm as compared to those fed on the EDTA blood through a membrane ($P \leq 0.05$). Direct skin feeding consistently achieved more than 90% feeding rates, while the rates for membrane feeding were between 48.2% and 98.2%.

Results: Direct skin feeding assay (DSFA) consistently showed higher mosquito feeding rates (93.3%–100%) when compared with the membrane feeding assay (MFA) (48%–98.2%). Pair-wise comparison between methods showed no significant difference in midgut infection rates between mosquitoes exposed via each method and a strong correlation was observed in midgut infection rates for both feeding methods ($r = 0.89$, $P < 0.0001$). A strong correlation between salivary gland infection and patient serum viremia were observed for both DSFA ($r = 0.80$, $P < 0.0001$) and MFA ($r = 0.79$, $P < 0.0001$) methods. Unlike midgut infection, salivary gland infection was only observed in mosquitoes that fed on patients with viremia of more than 5.5 Log₁₀copies/ml.

External Quality Assessment of Dengue and Chikungunya Diagnostics in the Asia Pacific Region, 2015 (Soh *et al.*, 2016)

Twenty-four national-level public health laboratories performed routine diagnostic assays on a proficiency testing panel consisting of two modules. Module A contained serum samples spiked with cultured DENV or chikungunya virus for the detection of nucleic acid and DENV nonstructural protein 1 (NS1) antigen. Module B contained human serum samples for the detection of anti-DENV antibodies.

Analysis: The appropriate dengue diagnostic tools must be used at the correct time for the correct diagnosis of dengue. While 19/24 (79.2%) laboratories employed assays for both acute (RT-PCR) and recent (anti-DENV IgM) DENV infection, four performed antibody testing for dengue but lacked assays for early detection (preantibody immunological response) of dengue such as RT-PCR or NS1 kits and one could perform RT-PCR but had no serology capacity. With an incomplete set of diagnostics, these laboratories may be unable to diagnose a proportion of DENV infections and should consider quickly strengthening their capacity through the use of commercial ELISA assays for the detection of NS1 antigen or anti-DENV IgM antibodies. Accuracy was high ($\geq 85\%$) for DENV and CHIKV detection and DENV serotyping by RT-PCR. The few errors in DENV detection (false negatives) appeared to be clustered in samples A2015-V01 and V02, which were identical, high-titer DENV-1 samples. Most of the inaccuracies in Module A were in NS1 testing. Specifically, 87.5% of NS1 testing errors were derived from a single DENV-positive sample, A2015-V03, being reported as negative or equivocal (7/8 laboratories), particularly when using the Platelia Dengue NS1 Ag kit. This suggests that the NS1 levels in the sample may have been at the threshold of detection for the kit, making complementary assays for virus detection, such as RT-PCR, highly relevant.

RESULTS: Among 20 laboratories testing Module A, 17 (85%) correctly detected DENV RNA by reverse transcription polymerase chain reaction (RT-PCR), 18 (90%) correctly determined serotype, and 19 (95%) correctly identified CHIKV by RT-PCR. Ten of 15 (66.7%) laboratories performing NS1 antigen assays obtained the correct results. In Module B, 18/23 (78.3%) and 20/20 (100%) of laboratories, correctly detected anti-DENV IgM and IgG, respectively. Detection of acute/recent DENV infection by both molecular (RT-PCR) and serological methods (IgM) was available in 19/24 (79.2%) participating laboratories.

Acknowledgments

We thank Elsevier for giving us an opportunity to write this article. We sincerely thank Dr. Asif Khan for the invitation to write a book article.

See also: Ecosystem Monitoring Through Predictive Modeling. Natural Language Processing Approaches in Bioinformatics

References

- Anon, 2011a. Laboratory Quality Management System. Geneva: World Health Organization.
- Anon, 2011b. Asia Pacific Strategy for Emerging Diseases (2010). Manila: World Health Organization Regional Office for the Western Pacific.
- Anon, 2012a. WHO External Quality Assessment Project for the detection of influenza Virus Type A by PCR. Manila: World Health Organization Regional Office for the Western Pacific.
- Anon, 2012b. World Health Organization and Special Programme for Research and Training in Tropical Diseases: Handbook for clinical management of dengue. Geneva: World Health Organization.
- Anon, 2013. Dengue Situation Update – 25 December 2013. Manila: World Health Organization Regional Office for the Western Pacific.
- Anon, 2014a. Communicable Disease Surveillance in Singapore 2013. Singapore: Ministry of Health.
- Anon, 2014b. Dengue Situation Update – 22 April 2014. Manila: World Health Organization Regional Office for the Western Pacific.
- Anon, 2014c. Dengue Situation Update – 3 June 2014. Manila: World Health Organization Regional Office for the Western Pacific.
- Anon, 2015. Dengue Situation Update – 13 January 2015. Manila: World Health Organization Regional Office for the Western Pacific.
- Anon, 2016. WHO Director-General summarizes the Outcome of the Emergency Committee Regarding Clusters of Microcephaly and Guillain-Barré Syndrome. Geneva: World Health Organization.
- Arima, Y., *et al.*, 2014. Ongoing local transmission of dengue in Japan, August to September 2014. *Western Pacific Surveillance and Response Journal* 5 (4), 27–29. doi:10.5365/wpsar.2014.5.3.007.
- Arima, Y., *et al.*, 2015. Epidemiologic update on the dengue situation in the Western Pacific Region, 2012. *Western Pacific Surveillance and Response Journal* 6 (2), doi:10.5365/wpsar.2014.5.4.002.
- Balmaseda, A., Hammond, S.N., Perez, L., *et al.*, 2006. Serotype-specific differences in clinical manifestations of dengue. *The American Journal of Tropical Medicine and Hygiene* 74, 449–456.
- Bancroft, T.L., 1906. On the etiology of dengue fever. *The Australasian Medical Gazette* 25, 17.
- Bandelt, H.J., Forster, P., Rohl, A., 1999. Median-joining networks for inferring intraspecific phylogenies. *Molecular Biology and Evolution* 16 (1), 37–48. PMID:10331250.
- Banu, S., *et al.*, 2011. Dengue transmission in the Asia-Pacific region: Impact of climate change and socio-environmental factors. *Tropical Medicine & International Health* 16, 598–607.
- Bhatt, S., *et al.*, 2013. The global distribution and burden of dengue. *Nature* 496, 504–507. doi:10.1038/nature12060.
- Burke, D.S., Nisalak, A., Johnson, D.E., Scott, R.M., 1988. A prospective study of dengue infections in Bangkok. *The American Journal of Tropical Medicine and Hygiene* 38, 172–180.
- Cao-Lormeau, V.M., Roche, C., Musso, D., *et al.*, 2014. Dengue virus type 3, South Pacific Islands, 2013. *Emerging Infectious Diseases* 20 (6), 1034–1036.
- Carrasco, L.R., Lee, L.K., Lee, V.J., *et al.*, 2011. Economic impact of dengue illness and the cost-effectiveness of future vaccination programs in Singapore. *PLOS Neglected Tropical Diseases* 5, e1426.
- Chambers, T.J., Hahn, C.S., Galler, R., Rice, C.M., 1990. Flavivirus genome organization, expression, and replication. *Annual Review of Microbiology* 44, 649–688. PMID: 2174669.
- Chan, K.S., Chang, J.S., Chang, K., *et al.*, 2009. Effect of serotypes on clinical manifestations of dengue fever in adults. *Journal of Microbiology, Immunology and Infection* 42, 471–478.
- Chen, H.L., Lin, S.R., Liu, H.F., *et al.*, 2008. Evolution of dengue virus type 2 during two consecutive outbreaks with an increase in severity in southern Taiwan in 2001–2002. *The American Journal of Tropical Medicine and Hygiene* 79 (4), 495–505. PMID:18840735.
- Chen, R., Vasilakis, N., 2011. Dengue – Quo tu et quo vadis? *Viruses* 3, 1562–1608.
- Chen, W.J., Wu, H.R., Chiou, S.S., 2003. E/NS1 modifications of dengue 2 virus after serial passages in mammalian and/or mosquito cells. *Intervirology* 46 (5), 289–295.
- Cleland, J.B., Bradley, B., McDonald, W., 1918. Dengue fever in Australia. *The Journal of Hygiene* 16 (4), 319–418.

- Cleland, J.B., Bradley, B., McDonald, W., 1919. Further experiments in the etiology of dengue fever. *The Journal of Hygiene* 18 (3), 217–254.
- Coffey, L.L., Mertens, E., Brehin, A.C., *et al.*, 2009. Human genetic determinants of dengue virus susceptibility. *Microbes and Infection* 11 (2), 143–156. pmid:19121645.
- Diehlmann, H., 1999. Laboratory Rearing Of Mosquitoes Using a Hemotek Feeding System. Hronov: Czech Republic Grafické závody.
- Fried, J.R., Gibbons, R.V., Kalayanaraj, S., *et al.*, 2010. Serotype-specific differences in the risk of dengue hemorrhagic fever: An analysis of data collected in Bangkok, Thailand from 1994 to 2006. *PLOS Neglected Tropical Diseases* 4, e617.
- Gan, V.C., Tan, L.K., Lye, D.C., *et al.*, 2014. Diagnosing dengue at the point-of-care: Utility of a rapid combined diagnostic kit in Singapore. *PLOS ONE* 9 (3), e90037. pmid:24646519.
- Governments of Malaysia and Singapore, 2014. Joint Media Release: UNITEDengue Cross-Border Data Sharing Provides Countries With Timely Risk Alerts. Singapore: National Environment Agency.
- Green, S., Rothman, A., 2006. Immunopathological mechanisms in dengue and dengue hemorrhagic fever. *Current Opinion in Infectious Diseases* 19 (5), 429–436. pmid:16940865.
- Guzman, M.G., Alvarez, M., Halstead, S.B., 2013. Secondary infection as a risk factor for dengue hemorrhagic fever/dengue shock syndrome: An historical perspective and role of antibody-dependent enhancement of infection. *Archives of Virology* 158, 1445–1459.
- Guzman, M.G., Alvarez, A., Vazquez, S., *et al.*, 2012. Epidemiological studies on dengue virus type 3 in Playa municipality, Havana, Cuba, 2001–2002. *International Journal of Infectious Diseases* 16, e198–e203.
- Guzman, M.G., Deubel, V., Pelegrino, J.L., *et al.*, 1995. Partial nucleotide and amino acid sequences of the envelope and the envelope/non-structural protein-1 gene junction of four dengue-2 virus strains isolated during the 1981 Cuban epidemic. *The American Journal of Tropical Medicine and Hygiene* 52 (3), 241–246. pmid:7694966.
- Guzman, M.G., Kouri, G., 2003. Dengue and dengue hemorrhagic fever in the Americas: Lessons and challenges. *Journal of Clinical Virology* 27, 1–13.
- Guzman, M.G., Kouri, G., Valdes, L., *et al.*, 2000. Epidemiologic studies on dengue in Santiago de Cuba, 1997. *American Journal of Epidemiology* 152, 793–799. discussion 804.
- Hall, T.A., 1999. BioEdit: A user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. *Nucleic Acids Symposium Series* 41, 95–98.
- Halsey, E.S., Marks, M.A., Gotuzzo, E., *et al.*, 2012. Correlation of serotype-specific dengue virus infection with clinical manifestations. *PLOS Neglected Tropical Diseases* 6, e1638.
- Halstead, S.B., 2008. Dengue virus-mosquito interactions. *Annual Review of Entomology* 53, 273–291.
- Han, H.K., Ang, L.W., Tay, J., 2012. Review of Dengue Serotype Surveillance Programme in Singapore. *Epidemiological News Bulletin*.
- Hapuarachchi, H.C., Chua, R.C.R., Shi, Y., *et al.*, 2015. Clinical outcome and genetic differences within a monophyletic Dengue virus type 2 populations. *PLOS ONE* 10 (3), e0121696.
- Hapuarachchi, H.C., Koo, C., Kek, R., *et al.*, 2016. Intra-epidemic evolutionary dynamics of a Dengue virus type 1 population reveal mutant spectra that correlate with disease transmission. *Scientific Reports* 6, 22592.
- Holmes, E.C., Burch, S.S., 2000. The causes and consequences of genetic variation in dengue virus. *Trends in Microbiology* 8 (2), 74–77. pmid:10664600.
- Holmes, E.C., Twiddy, S.S., 2003. The origin, emergence and evolutionary genetics of dengue virus. *Infection, Genetics and Evolution* 3 (1), 19–28. pmid:12797969.
- Horwood, P., Bande, G., Dagina, R., *et al.*, 2013. The threat of chikungunya in Oceania. *Western Pacific Surveillance and Response* 4 (2), 8–10.
- Jupp, P.G., 1976. The susceptibility of four South African species of *Culex* to West Nile and Sindbis viruses by two different infecting methods. *Mosquito News* 36, 166–173.
- Khor, C.C., Chau, T.N., Pang, J., *et al.*, 2011. Genome-wide association study identifies susceptibility loci for dengue shock syndrome at MICB and PLCE1. *Nature Genetics* 43 (11), 1139–1141. pmid:22001756.
- Kindhauser, M.K., Allen, T., Frank, V., Santhana, R.S., Dye, C., 2016. Zika: The origin and spread of a mosquito-borne virus. *Bull World Health Organization*. (submitted) 10.2471/BLT.16.171082.
- Kramer, L.D., Ebel, G.D., 2003. Dynamics of flavivirus infection in mosquitoes. *Advances in Virus Research* 60, 187–232.
- Kumaria, R., 2009. Correlation of disease spectrum among four dengue serotypes: A five years hospital based study from India. *Brazilian Journal of Infectious Diseases* 14, 141–146.
- Kutsuna, S., Kato, Y., Moi, M.L., *et al.*, 2015. Autochthonous dengue fever, Tokyo, Japan, 2014. *Emerging Infectious Diseases* 21 (3), 517–520.
- Lai, Y.L., Chung, Y.K., Tan, H.C., *et al.*, 2007. Cost-effective real-time reverse transcriptase PCR (RT-PCR) to screen for Dengue virus followed by rapid single-tube multiplex RT-PCR for serotyping of the virus. *Journal of Clinical Microbiology* 45 (3), 935–941. pmid:17215345.
- Lee, K.S., *et al.*, 2012. Dengue virus surveillance in Singapore reveals high viral diversity through multiple introductions and in situ evolution. *Infection, Genetics and Evolution* 12, 77–85.
- Lee, K.S., Lai, Y.L., Lo, S., *et al.*, 2010. Dengue virus surveillance for early warning, Singapore. *Emerging Infectious Diseases* 16, 847–849.
- Leitmeyer, K.C., Vaughn, D.W., Watts, D.M., *et al.*, 1999. Dengue virus structural differences that correlate with pathogenesis. *Journal of Virology* 73 (6), 4738–4747. pmid:10233934.
- Ler, T.S., Ang, L.W., Yap, G.S.L., *et al.*, 2011. Epidemiological characteristics of the 2005 and 2007 dengue epidemics in Singapore – similarities and distinctions. *Western Pacific Surveillance and Response* J2, 24–29.
- Meyer, R.P., Hardy, J.L., Presser, S.B., 1983. Comparative vector competence of *Culex tarsalis* and *Culex quinquefasciatus* from the Coachella, Imperial, and San Joaquin Valleys of California for St. Louis encephalitis virus. *The American Journal of Tropical Medicine and Hygiene* 32 (2), 305–311.
- Miller, B.R., 1987. Increased yellow fever virus infection and dissemination rates in *Aedes aegypti* mosquitoes orally exposed to freshly grown virus. *Transactions of the Royal Society of Tropical Medicine and Hygiene* 81 (6), 1011–1012.
- Ministry of Health, 2012. Communicable Disease Surveillance in Singapore 2011. Singapore: Ministry of Health.
- Murrell, B., Wertheim, J.O., Moola, S., *et al.*, 2012. Detecting individual sites subject to episodic diversifying selection. *PLOS Genetics* 8 (7), e1002764.
- Ng, L.-C., Y-k, C., Koo, C., *et al.*, 2013. dengue outbreaks in Singapore and Malaysia caused by different viral strains. *The American Journal of Tropical Medicine and Hygiene* 2015 (92), 1150–1155.
- Ng, L.C., 2011. Challenges in dengue surveillance and control (Editorial). *Western Pacific Surveillance and Response Journal* 2, 1–3.
- Nguyet, M.N., Duong, T.H., Trung, V.T., *et al.*, 2013. Host and viral features of human dengue cases shape the population of infected and infectious *Aedes aegypti* mosquitoes. *Proceedings of the National Academy of Sciences of the United States of America* 110 (22), 9072–9077.
- Nisalak, A., Endy, T.P., Nimmannitya, S., *et al.*, 2003. Serotype-specific dengue virus circulation and dengue disease in Bangkok, Thailand from 1973 to 1999. *The American Journal of Tropical Medicine and Hygiene* 68, 191–202.
- OMalley, M.A., 2016. The ecological virus. *Studies in History and Philosophy of Biological and Biomedical Sciences* 59, 71–79.
- Pandey, B.D., Igarashi, A., 2000. Severity-related molecular differences among nineteen strains of dengue type 2 viruses. *Medical Microbiology and Immunology* 44 (3), 179–188. pmid:10789505.
- Pok, K.Y., Squires, R.C., Tan, L.K., *et al.*, 2015a. First round of external quality assessment of dengue diagnostics in the WHO Western Pacific Region, 2013. *Western Pacific Surveillance and Response Journal* 6 (2), 73–81.
- Pok, K.Y., Squires, R.C., Tan, L.K., *et al.*, 2015b. First round of external quality assessment of dengue diagnostics in the WHO Western Pacific Region, 2013. *Western Pacific Surveillance and Response* 6 (2), 73–81. doi:10.5365/wpsar.2015.6.1.017.
- Pond, S.L., Frost, S.D., Muse, S.V., 2005. HyPhy: Hypothesis testing using phylogenies. *Bioinformatics* 21 (5), 676–679. pmid:15509596.
- Prat, C.M., Flusin, O., Panella, A., *et al.*, 2014. Evaluation of commercially available serologic diagnostic tests for chikungunya virus. *Emerging Infectious Diseases* 20 (12), 2129–2132. doi:10.3201/eid2012.141269.
- Rabaa, M.A., Ty Hang, V.T., Wills, B., *et al.*, 2010. Phylogeography of recently emerged DENV-2 in southern Vietnam. *PLOS Neglected Tropical Diseases* 4, e766.
- RCoreTeam, 2013. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Austria: Vienna, Available at: <http://www.R-project.org/>.
- Richards, S.L., Pesko, K., Alto, B.W., Mores, C.N., 2007. Reduced infection in mosquitoes exposed to blood meals containing previously frozen flaviviruses. *Virus Research* 129, 224–227.

- Rico-Hesse, R., Harrison, L.M., Salas, R.A., *et al.*, 1997a. Origins of dengue type 2 viruses associated with increased pathogenicity in the Americas. *Virology* 230 (2), 244–251. pmid:9143280.
- Rico-Hesse, R., Harrison, L.M., Salas, R.A., *et al.*, 1997b. Origins of dengue type 2 viruses associated with increased pathogenicity in the Americas. *Virology* 230, 244–251.
- Rico-Hesse, R., 2003. Microevolution and virulence of dengue viruses. *Advances in Virus Research* 59, 315–341.
- Ritchie, S.A., 2014. Dengue vector bionomics: Why *Aedes aegypti* is such a good vector. In: Gubler, D.J., Ooi, E.E., Vasudevan, S., Farrar, J. (Eds.), *Dengue and Dengue Hemorrhagic Fever*. Oxfordshire: CABI, pp. 455–480.
- Rodhain, F., Rosen, L., 1997. Mosquito vectors and dengue virus-vector relationships. In: Gubler, D.J., Kuno, G. (Eds.), *Dengue and Dengue Hemorrhagic Fever*. Cambridge, MA: CABI Publishing, pp. 47–60.
- Rutledge, L.C., Ward, R.A., Gould, D.J., 1964. Studies on the feeding response of mosquitoes to nutritive solutions in a new membrane feeder. *Mosquito News* 24, 407–419.
- Sangkawibha, N., Rojanasuphot, S., Ahandrik, S., *et al.*, 1984. Risk factors in dengue shock syndrome: A prospective epidemiologic study in Rayong, Thailand. I. The 1980 outbreak. *American Journal of Epidemiology* 120, 653–669.
- Shepard, D.S., Undurraga, E.A., Lees, R.S., *et al.*, 2012. Use of multiple data sources to estimate the economic cost of dengue illness in Malaysia. *The American Journal of Tropical Medicine and Hygiene* 87, 796–805.
- Siler, J.E., Hall, M.W., Hitchens, A.P., 1926. Dengue: Its history, epidemiology, mechanism of transmission, etiology, clinical manifestations, immunity and prevention. *Philippine Journal of Science* 29, 1–302.
- Simmons, J.S., St John, J.H., Reynolds, F.H.K., 1931. Experimental studies of dengue. *Philippine Journal of Science* 44, 1–247.
- Soh, L.T., Squires, R.C., Tan, L.K., *et al.*, 2016. External quality assessment of dengue and chikungunya diagnostics in the Asia Pacific region, 2015. *Western Pacific Surveillance and Response Journal: WPSAR* 7 (2), 26–34.
- Suaya, J.A., Shepard, D.S., Siqueira, J.B., *et al.*, 2009. Cost of dengue cases in eight countries in the Americas and Asia: A prospective study. *The American Journal of Tropical Medicine and Hygiene* 80, 846–855.
- Tamura, K., Stecher, G., Peterson, D., Filipski, A., Kumar, S., 2013. MEGA6: Molecular Evolutionary Genetics Analysis version 6.0. *Molecular Biology and Evolution* 30 (12), 2725–2729. pmid:24132122.
- Tan, C.H., Wong, P.S., Li, M.Z., *et al.*, 2016. Membrane feeding of dengue patient's blood as a substitute for direct skin feeding in studying *Aedes*-dengue virus interaction. *Parasites & Vectors* 9, 211.
- Tuiskunen, A., Monteil, V., Plumet, S., *et al.*, 2011. Phenotypic and genotypic characterization of dengue virus isolates differentiates dengue fever and dengue hemorrhagic fever from dengue shock syndrome. *Archives of Virology* 156 (11), 2023–2032.
- Twiddy, S.S., Woelk, C.H., Holmes, E.C., 2002. Phylogenetic evidence for adaptive evolution of dengue viruses in nature. *Journal of General Virology* 83 (Pt 7), 1679–1689. pmid:12075087.
- Vaughn, D.W., Green, S., Kalayanarooj, S., *et al.*, 2000. Dengue viremia titer, antibody response pattern, and virus serotype correlate with disease severity. *The Journal of Infectious Diseases* 181 (1), 2–9. pmid:10608744.
- Vaughn, D.W., Green, S., Kalayanarooj, S., *et al.*, 2000. Dengue viremia titer, antibody response pattern, and virus serotype correlate with disease severity. *The Journal of Infectious Diseases* 181, 2–9.
- Wang, W.K., Chen, H.L., Yang, C.F., *et al.*, 2006. Slower rates of clearance of viral load and virus-containing immune complexes in patients with dengue hemorrhagic fever. *Clinical Infectious Diseases* 43 (8), 1023–1030. pmid:16983615.
- Weaver, S.C., Lecuit, M., 2015. Chikungunya virus and the global spread of a mosquito-borne disease. *The New England Journal of Medicine* 372 (13), 1231–1239. 10.1056/NEJMr1406035.
- Whitehead, S.S., Hanley, K.A., Blaney Jr, J.E., *et al.*, 2003. Substitution of the structural genes of dengue virus type 4 with those of type 2 results in chimeric vaccine candidates which are attenuated for mosquitoes, mice, and rhesus monkeys. *Vaccine* 21 (27–30), 4307–4316. pmid:14575763.
- Whitehorn, J., Simmons, C.P., 2011. The pathogenesis of dengue. *Vaccine* 29 (42), 7221–7228. pmid:21781999.
- WHO, 2009. *Dengue: Guidelines for Diagnosis, Treatment, Prevention and Control*, New ed. Geneva: World Health Organization.
- Wilder-Smith, A., Gubler, D.J., 2008. Geographic expansion of dengue: The impact of international travel. *The Medical Clinics of North America* 92, 1377–1390.
- Yap, G., Pok, K.Y., Lai, Y.L., *et al.*, 2010. Evaluation of Chikungunya diagnostic assays: Differences in sensitivity of serology assays in two independent outbreaks. *PLOS Neglected Tropical Diseases* 4 (7), e753. doi:10.1371/journal.pntd.0000753.
- Yung, C.F., Lee, K.S., Thein, T.L., *et al.*, 2015. Dengue serotype-specific differences in clinical manifestation, laboratory parameters and risk of severe disease in adults, Singapore. *The American Journal of Tropical Medicine and Hygiene*. <https://www.ncbi.nlm.nih.gov/pubmed/25825386>.
- Zhang, C., Mammen Jr, M.P., Chinawirotpisan, P., *et al.*, 2006. Structure and age of genetic diversity of dengue virus type 2 in Thailand. *Journal of General Virology* 87, 873–883.
- Zuker, M., 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research* 31 (13), 3406–3415. pmid:12824337.

Relevant Website

www.uniteddengue.org
UniteDengue.

Biographical Sketch



Dr. Vijayaraghava Seshadri Sundararajan has been an advisor to Bioclues.org, and has organized international conferences/workshops.

Mapping the Environmental Microbiome

Lena Takayasu and Wataru Suda, RIKEN Center for Integrative Medical Sciences, Yokohama City, Japan

Masahira Hattori, RIKEN Center for Integrative Medical Sciences, Yokohama City, Japan and Waseda University, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

On the earth, highly diverse bacteria adaptively reside in various environments from animal and plant body sites to extreme environments such as the deep ocean and hot springs. The total number of bacterial cells on the earth are estimated to be $4\text{--}6 \times 10^{30}$, suggesting that the total amount of cellular carbon from bacteria exceeds that of plants and animals (Whitman *et al.*, 1998). Bacteria inhabiting natural environments such as oceans, fresh water, and soil play important roles in global resource renewal cycles such as the carbon cycle and nitrogen cycle by making organics and inorganics available to other living things. On the other hand, animal- and plant-related microbiomes are highly dense compared with environmental microbiomes (Whitman *et al.*, 1998), although total cell numbers are less than those of environmental microbiomes (Table 1). Among them, human-associated microbiomes are attracting attention in the medical field, as current studies strongly suggest the profound relationship between the human gut microbiome and host health and disease.

The human microbiomes such as the gut, oral, and skin microbiome are known to be highly body-site-specific and highly variable between individuals (The Human Microbiome Project Consortium, 2012). These characteristic features of human microbiomes are associated with various internal and external factors such as host genetic background, dietary intake, antibiotic use, pregnancy, age, and geography (Benson *et al.*, 2010; Wu *et al.*, 2011; Manichanh *et al.*, 2010; DiGiulio *et al.*, 2015; Yatsunenkov *et al.*, 2012; Goodrich *et al.*, 2014). In addition, microbial imbalance, called dysbiosis, was also observed in patients with various diseases such as inflammatory bowel disease (IBD), type 2 diabetes, and obesity (Frank *et al.*, 2007; Qin *et al.*, 2012; Karlsson *et al.*, 2013; Turnbaugh *et al.*, 2009), demonstrating that the structural change in normal microbiomes is linked with diseases and the host homeostasis. Furthermore, in 2013, fecal microbiota transplantation (FMT), in which feces from healthy donors are transplanted to disease-afflicted patients, was first reported to be an effective therapy for chronic colitis caused by *Clostridium difficile*, and since then FMT has been tried for various diseases (Van Nood *et al.*, 2013; Moayyedi *et al.*, 2015). These studies implied that the human gut microbiome is involved in both disease onset and health maintenance. In fact, it was reported that some human gut species regulate differentiation and proliferation of pro- and antiinflammatory T cells (Treg and Th17) in the gut (Atarashi *et al.*, 2013, 2015). However, since the microbial community is composed of numerous species that interact with each other and with host cells with complex mechanisms, elucidating the feature of bacterial community through time and space is required to exactly and deeply understand how the microbial ecosystem is shaped and how it maintains our health by controlling the microbiomes.

Background/Fundamentals

In the 1800s, knowledge on bacterial fermentation and infectious diseases accumulated for the agricultural and medical fields by development of methods for isolating and culturing the bacterial species from the natural environments and host animals/humans. Today, the technologies for DNA analysis have revolutionarily advanced, enabling us to comprehensively evaluate ecological and functional features of the microbial communities and ecosystems, which is represented by a culture-independent metagenomic sequencing of the microbial community using next-generation sequencing (NGS) technologies. We herein present the development and improvement of analytical methods in microbiome research.

Table 1 Order of bacterial cell number on the earth

	Location	Order of Bacterial cell number
Environment	Soil	$10^{29}\text{--}10^{30}$
	Fresh water	10^{26}
	Sea water	10^{29}
Host-related	Human	10^{23}
	Animal	10^{24}
	Plant	10^{26}

Source: Reproduced from Morris, C. E., Kinkel, L.L., 2002. Fifty years of phyllosphere microbiology: Significant contributions to research in related fields. In: Lindow, S.E., Hecht-Poinar, E.I., Elliott, V. (Eds.), *Phyllosphere Microbiology*. St. Paul, Minn: APS Press, pp. 365–375. *Phyllosphere microbiology*.

Source: Reproduced from Whitman, W.B., Coleman, D.C., Wiebe, W.J., 1998. Prokaryotes: The unseen majority. *Proceedings of the National Academy of Sciences of the United States of America* 95, 6578–6583. PNAS.

Culture-Based Approach

Pure culture technology by which the specific microbial species are isolated and cultured on agar was established in the 1800s by Pasteur and Koch and others, and since then, microbiologists have intensively performed culture-based studies of pathogenic bacteria causative for human infectious diseases, and microorganisms useful for fermentation such as wine production in agriculture. In 1885, the commensal bacterium *Escherichia coli* was isolated from human feces by Theodor Escherich, and the early anaerobic culturing method succeeded in the pure culture of *Bifidobacterium* in 1899. In the mid-1980s, culture-independent DNA-based technologies such as bacterial 16S ribosomal RNA (rRNA) gene analysis were developed, and revealed that many more bacteria existed in environments including the human gut than those isolated by the conventional cultural methods. For example, only about 0.1% of bacteria in the ocean and 0.3% in the soil could be cultivated (Amann *et al.*, 1995). Thus, naturally occurring microbes include many species difficult to culture in the laboratory (Staley and Konopka, 1985).

DNA-Based Approach

In 1978, Carl Woese proposed three domains of life (bacteria, archaea, and eukaryotes) based on ribosomal RNA genes of which the gene products RNA molecules are essential components of ribosome in the cell. In the microbial ecological study, the 16S rRNA gene sequence has been extensively used to be a molecular marker in the taxonomy of prokaryotes (Woese and Fox, 1977; Woese, 1987). The first culture-independent 16S rRNA gene analysis of the environmental microbiome was reported in 1990 (Giovannoni *et al.*, 1990). The paper showed that the bacterial diversity estimated by the 16S gene data far exceeded that by the culture-dependent approach.

Another incredible advance in the culture-independent approach is use of the NGS technologies in place of the traditional Sanger method-based sequencing coupled with the recombinant DNA technology, which began around 2005, leading to the high-throughput analysis of 16S rRNA gene sequences from community samples with low cost. The NGS technologies have also determined numerous 16S rRNA gene and genome sequences of tens of thousands of microbes isolated individually, of which the databases are useful for precise assignment of the observed 16S sequences to known bacterial taxa by sequence similarity search, establishing a solid way to elucidate the overall microbial community structure.

However, it is difficult for the 16S rRNA gene analysis to obtain or infer the community's functional features such as metabolism of various biological compounds. To uncover the functionality of the microbial community, the metagenomic analysis based on whole genome shotgun sequencing was developed as the third method, in which many of the microbial gene sequences in the community can be randomly and comprehensively determined (Venter *et al.*, 2004; Tyson *et al.*, 2004). For the human gut microbiome, metagenomic sequencing was performed by the traditional Sanger method in the early days (Gill *et al.*, 2006; Kurokawa *et al.*, 2007), and thereafter by NGS technologies in 2009 and 2010 (Turnbaugh *et al.*, 2009; Qin *et al.*, 2010). Gene sequences can be assigned to the function by similarity search to public databases such as KEGG (Kyoto Encyclopedia of Genes and Genomes) and COG (Clusters of Orthologous Groups) (Kanehisa *et al.*, 2002; Tatusov *et al.*, 1997). In addition, the NCBI genome database composed of numbers of the individual microbial genome sequences has also been developed (see "Relevant Website section"). Particularly, the genome database of human-derived cultured microbes allowed us to analyze not only the gene composition but also the taxonomic composition in human microbiomes by directly mapping metagenomic sequences to the microbial genomes.

Application

We herein showed two popular methods using 16S rRNA gene and metagenomic sequences produced by NGS for human microbiome research.

16S rRNA Gene Analysis

The 16S rRNA gene (16S) analysis may be used most frequently for investigating the microbial diversity and composition in microbial communities coupled with the subsequent computational analysis of 16S sequences. The 16S rRNA gene is about 1500 bp long and contains conserved regions (functionally important) and nine alternatively located variable regions (V1–V9; functionally less important) among bacterial species (Andersson *et al.*, 2008; Fig. 1). The sequence similarity or diversity in variable regions can be used to discriminate taxonomically different species and to identify the particular species. In the 16S analysis, DNA fragments containing variable regions of almost all bacterial species in the community are simultaneously amplified by universal PCR primer sets annealed to the conserved regions.

The overall process of typical 16S analysis using the MiSeq sequencing system was shown in Fig. 1. MiSeq is the NGS that produces reads of ~350-bp covering at least two variable regions (e.g., V1–V2 in Fig. 1) by paired-end sequencing. Since MiSeq and other NGS sequencers generate tens of millions of 16S reads with one sequencing run, it is possible to sequence 16S amplicons from multiple different samples simultaneously in one sequencing run by using different sets of barcoded universal primers for the samples. The 16S reads of several thousands to ten thousands per sample are then subjected to quality check to remove low-quality reads including ones less than threshold of the average quality value (QV < 20) of base-calling, lacking PCR

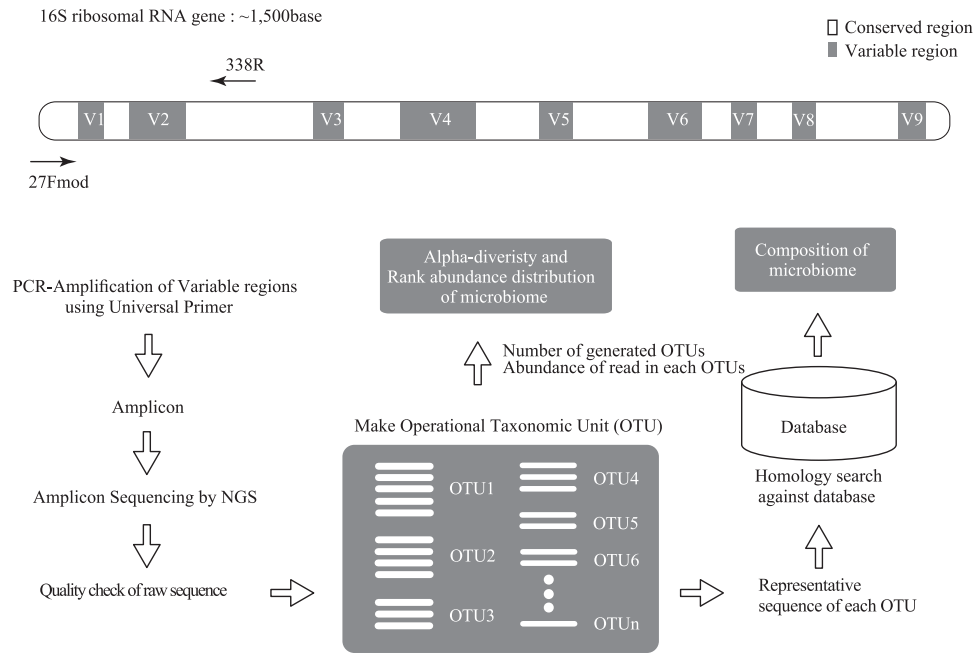


Fig. 1 Outline flowchart of 16S rRNA gene analysis.

primer sequences, and possible chimerisms. After trimming PCR primer sequences from the filter-passed reads, high-quality reads are clustered with a 97% sequence identity to obtain taxonomically different groups called operational taxonomic units (OTUs), which correspond to the bacterial “species.” The number of observed OTUs represents the community’s species richness (called α diversity), the number of 16S reads in an OTU approximates the abundance of the OTU, and their ratio represents the relative composition of all the OTUs. It is noted that overestimation of the OTU number is generally observed in the clustering of NGS reads with the sequencing error rate as high as $\sim 0.5\%$. Therefore, clustering of more 16S reads with the high error rate generates OTUs greater than the actual species number (Kim *et al.*, 2013). The bacterial taxa are assigned for the OTUs by similarity search of the OTU-representative 16S reads to the public 16S databases (Ribosomal Database Project (Cole *et al.*, 2008) and Green genes (DeSantis *et al.*, 2006)) and the NCBI genome database (also see below). Several software programs such as QIIME are useful for 16S analysis (Caporaso *et al.*, 2010). Direct mapping of all the 16S reads to the 16S and genome databases without making OTUs can also be used to analyze the species diversity and richness in the microbial community (Miyake *et al.*, 2015).

For evaluation of the overall structural similarity between two or more bacterial communities (called β diversity), the UniFrac distance analysis was developed for the 16S data (Lozupone and Knight, 2005). In this analysis, the similarity of phylogenetic trees constructed from the OTU sequences between the communities is calculated as UniFrac distance based on the common branch length between the samples. The UniFrac distance ranging from 0 to 1 represents 100% and 0% similarity between the communities. Furthermore, principle coordinate analysis (PCoA) based on UniFrac distance is also a useful tool to visualize the overall structural similarity between the samples (also see below). Multivariate analysis such as permutational multivariate analysis of variance (PERMANOVA) is used to test the statistical significance of the similarity and dissimilarity between different communities.

Metagenomic Analysis

As described above, the 16S analysis provides the microbial profiles in the communities, but it is not suitable to obtain functional features of the communities. On the other hand, metagenomic sequencing can collect many gene sequences representing the function of microbial communities.

The process of metagenomic analysis is shown in Fig. 2. Microbial DNA prepared from the community is subjected to whole-genome shotgun sequencing by NGS. Tens of millions of metagenomic reads of ~ 300 bp accounting for a total of several gigabases per sample are obtained and filtered with the quality check to remove low-quality reads similar to the 16S analysis. There are two processes for microbial and functional profiling in the metagenomic analysis. For the functional analysis, metagenomic reads are assembled with appropriate assemblers such as Newbler to obtain nonredundant microbial sequences/contigs, from which protein-coding genes can be predicted by several prediction software programs such as MetaGeneAnnotator (Noguchi *et al.*, 2008). The predicted genes are then subjected to similarity search to functional databases of genes and metabolic pathways such as COG and KEGG to assign and classify the genes into functions. The quantification of the identified genes is performed by mapping metagenomic reads to the genes. For the microbial analysis, taxonomic assignment and quantification of the microbial abundance

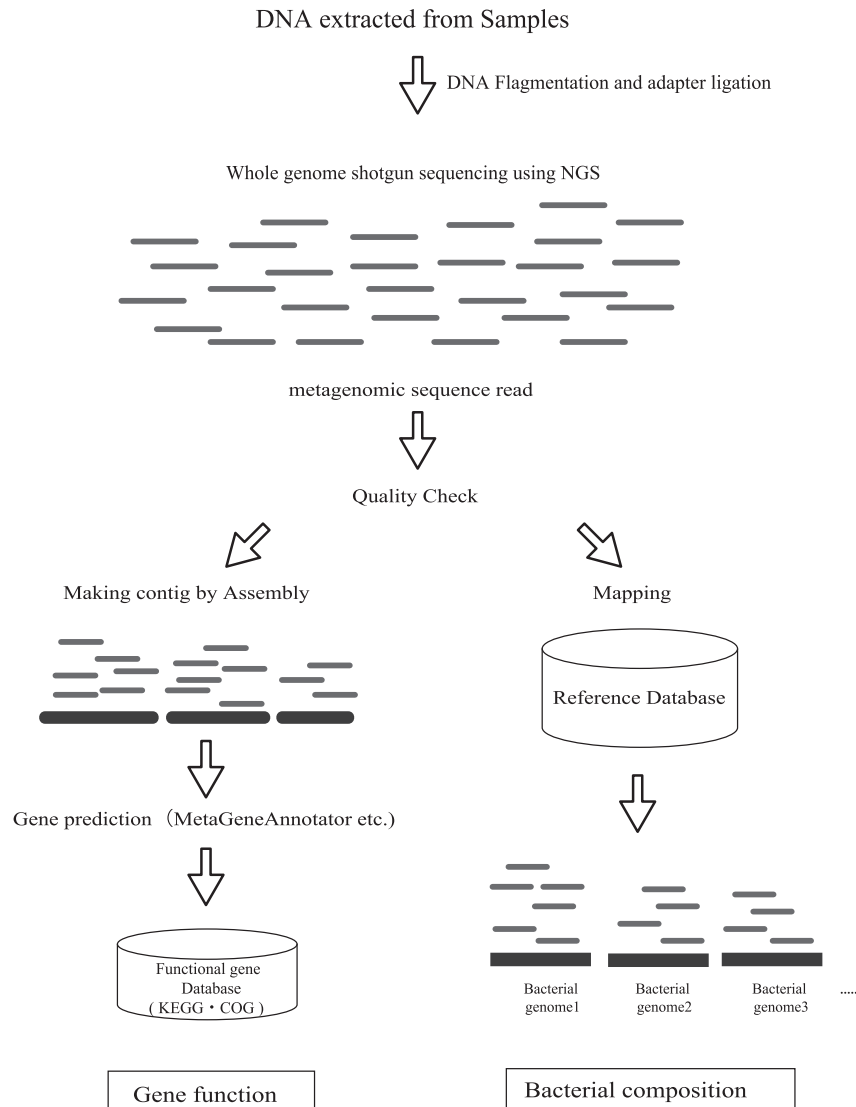


Fig. 2 Outline flowchart of metagenomic analysis.

can be conducted by mapping metagenomic reads to the NCBI genome databases compiled with more than 20,000 individual genomes including ~3000 human microbes. Metagenomic analysis usually requires a high-spec computation system to handle larger datasets than the 16S analysis. For the high-performance computational analysis, cloud server systems such as MG-RAST have been developed, in which the quality check and calculation of the community's structures regarding microbial and functional profiles can be carried out automatically (Meyer *et al.*, 2008).

Most of metagenomic sequencing analyses have used NGS sequencers such as Illumina HiSeq and MiSeq because those rapidly collect large amounts of sequence data with low cost. However, since these sequencers produce short reads of ~300 bp, and the assembly of them generates relatively short contigs possibly containing misassembled contigs mainly due to the presence of high-similarity sequences such as 16S rRNA genes and horizontally transferred genes within and between microbial genomes in the community. Therefore, short-read assembly includes some ambiguity in reconstruction of individual genomes and contigs. To improve these problems encountered in standard short-read sequencing, several computational tools have been developed for reconstruction of the high-quality genomes by binning short contigs based on coabundance across samples (Nielsen *et al.*, 2014; Alneberg *et al.*, 2014; Kang *et al.*, 2015; Cleary *et al.*, 2015), and by incorporating independently preassembled synthetic reads of ~10 kb in assembly of short reads, which improved the assembly accuracy and the contig length (Sharon *et al.*, 2015; Kuleshov *et al.*, 2016). Use of long-read NGS sequencers such as PacBio and MinION that produce longer reads (>10 kb) than the short-read sequencers is also an alternative approach to improve the outcomes in metagenomic analysis (Leonard *et al.*, 2014; Tsai *et al.*, 2016). Actually, long-read sequencing facilitated accurately assembled long contigs for a wide range of individual microbial and eukaryotic genomes (Rasko *et al.*, 2011; Chin *et al.*, 2013; Chaisson *et al.*, 2015).

Illustrative Example(s) or Case Studies

Diversity Index

There are several indices to evaluate ecological profiles of the microbial community structure. The simplest diversity index is the number of distinct OTUs/species detected in 16S analysis and the number of genomes mapped by metagenomic reads in metagenomic analysis (see section “Application”). In 16S analysis, unseen novel species can be included as unassigned OTUs that have <70% identity with known species, while only known species/genomes having high similarity to metagenomic reads are identified but original genomes/species of unmapped reads are still unknown in metagenomic analysis. Also, estimation of the microbial abundance may be more or less influenced by the PCR bias at the amplification step of the target 16S rRNA gene using universal primers in 16S analysis, while metagenomic analysis includes no PCR step at all. These differences may cause some discrepancy in the outcomes between these two methods. Other diversity indices, Chao1 and ACE, are also used for estimating the maximum OTU/species number by extrapolation of the observed OTU number, respectively (Chao, 1987; Chazdon *et al.*, 1998).

The Shannon index is also a popular metric for evaluation of the community structure, in which both the species richness and the abundance evenness are considered. The Shannon index is calculated based on Shannon’s information entropy, that is, the more equally distributed the microbial abundance, the higher the score of the index when the species richness is the same.

We compared the OTU (see section “16S rRNA Gene Analysis”) number, the Shannon index, and the number of phyla based on our in-house 16S datasets among 39 microbiomes including 23 gut microbiomes of orangutans (domestic zoos), humans (Japanese healthy adults), pigs (domestic farms), experimental mice (conventional), and experimental rats (conventional); foliar microbiomes of *Cerasus* and *Ginkgo*, and four soil microbiomes from a potato field; and five samples from river fresh water (Saitama, Japan) and five samples from oceanic water (Saga, Japan) (Fig. 3). The soil communities showed the highest diversity in the three indices followed by the sea and river communities. The diversity in the sea communities was slightly higher than that of river in all three indices. The three indices also revealed the tendency of the diversity to be relatively lower in the host-related microbiomes than the environmental microbiomes, suggesting less complex community structure of the host-related microbiomes than the environmental ones. These differences also suggest that the host-related microbiomes have selective pressures on the colonization by host physiology including immunity compared with the environmental microbiomes.

Cumulative Relative Abundance Distribution

We observed the cumulative relative abundance distribution (CRAD) shared among various types of human gut microbiomes from healthy adults, infants, and disease-afflicted patients where the taxonomic composition was significantly varied (Takayasu *et al.*, 2017b). The CRADs showed linear relationship between the population size of bacterial species (X) and the existing bacterial species number holding more than X in a log-log plot, which can be approximated by power law function.

CRADs of the 39 microbiomes described above also indicated the linearity in a log-log plot (Fig. 4), suggesting that all of the microbial communities from environments, animal guts, and plants also follow the power law. However, the slopes of the CRADs corresponding to the power exponent of power law function were distinct ranging from 0.6 and 1.5 among the microbiomes, in which the soil community was characterized by the highest power exponent, and the human microbiome by the smallest power exponent. These data suggested that the exponent of CRAD has the tendency of correlation with the α diversity indices represented by the species richness and the Shannon index.

Taxonomic Assignment

We showed the average phylum-level microbial composition of 10 communities from the different hosts and environments described in 4.1 based on 16S data in Fig. 5, respectively. The species assignment of the 16S OTUs was performed with similarity search with a 70% identity to the databases. The soil community was dominated by higher number of phyla than the others, as

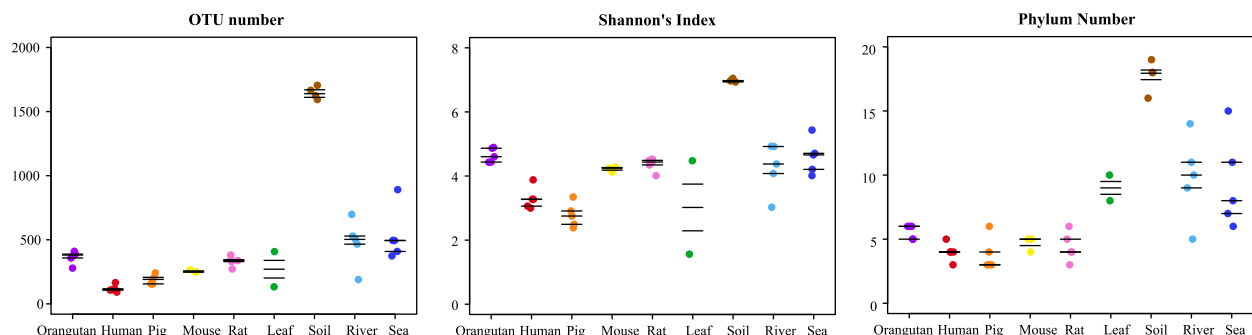


Fig. 3 Comparison of diversity indices of various microbiomes.

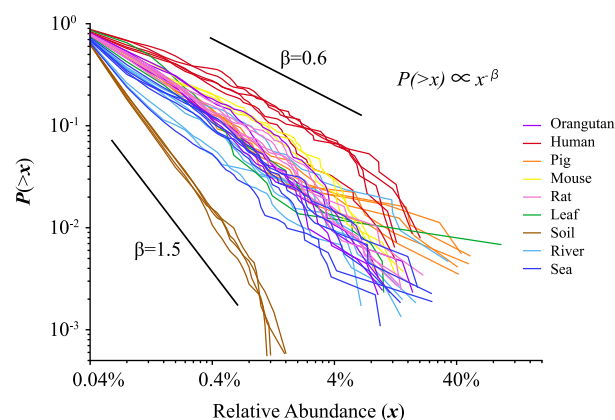


Fig. 4 Cumulative relative abundance distributions of various microbiomes.

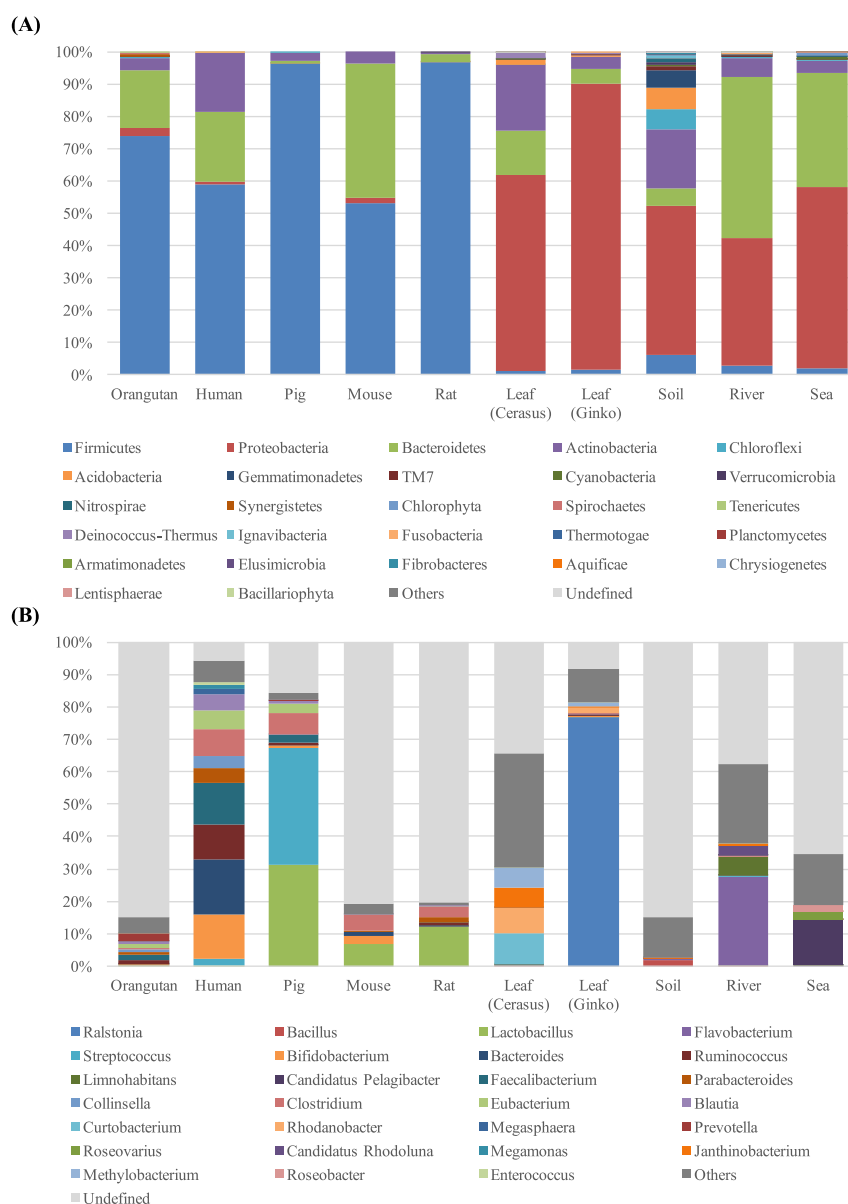


Fig. 5 The average bacterial composition of nine microbiomes based on 16S data. (A) Nine microbiomes at the phylum level. (B) Nine microbiomes at the genus level.

shown in Fig. 3. The environmental microbiomes were prominently dominated by the species belonging to the phylum Proteobacteria, while the host-related microbiomes were dominated by the species in Firmicutes. Four major phyla, Proteobacteria, Bacteroidetes, Actinobacteria, and Firmicutes occupied up to ~92% of the total abundance in all microbiomes except for soil, for which the combined abundance of the top four phyla (Proteobacteria, Actinobacteria, Chloroflexi, and Acidobacteria) was around ~76% of the total abundance. Several phyla such as Gemmatimonadetes and Nitrospirae were characteristically detected in the soil microbiomes. The hydrosphere microbiomes (river and sea) were abundant in two phyla, Proteobacteria and Bacteroidetes, different from the leaf and soil microbiomes. On the other hand, the animal gut microbiomes were dominated by the four phyla, Firmicutes, Bacteroidetes, Actinobacteria, and Proteobacteria up to 99% or more with the strong selective pressure on the colonization as described above.

However, at the genus-level species assignment, many of the microbiomes were composed of many of the unidentified species distinct from known genera, particularly for orangutans, mice, rats, soil, and sea (Fig. 5(B)). This poor genus-level assignment may be largely due to an insufficient number of bacterial strains isolated from these environments and the high abundance of phylogenetically distant species from known species in these communities. Thus, taxonomic analysis based on the 16S and metagenomic data largely depends on the quality of the databases, although the overall structural or functional features can be evaluated from these data. In contrast, the database of human microbes has been systematically constructed by an effort of the Human Microbiome Project for the current 10 years, in which genomes of ~3000 human microbes were determined. Similarity search of human microbiome 16S data to the public 16S and genome databases was performed with the range from 97% to 70% identities. Almost all of the OTUs can be assigned to known phyla with a $\geq 70\%$ identity, 60% of them to genera with a $\geq 94\%$ identity, and 30% of them to species with a $\geq 97\%$ identity, while less than 20% of murine gut OTUs can be assigned to known genera and species (Fig. 6(A)). The combined abundance of the assigned OTUs in the human gut microbiome accounted for over 80% of the total abundance at the genus and species levels, indicating that the majority of dominant species in the human gut microbiome were uncovered and the species yet-unidentified might be minor members (Fig. 6(B)). The sufficient taxonomic assignment of pig microbiomes may be because the highly abundant species were coincidentally shared between human and pig microbiomes. Again, poor species assignment of mouse, rat, and orangutan microbiomes at the genus level may be due to insufficient 16S and genome databases and phylogenetic differences in the constituted taxa between gut microbiomes of these animals and humans, similar to the environmental microbiomes mentioned above. For taxonomic assignment in metagenome analysis, metagenomic reads are directly mapped to known genomes in the genome databases with a $\geq 95\%$ identity and a $\geq 90\%$ length coverage. The reads mapped to known genomes account for ~70% of the total reads in human gut microbiomes (Nishijima *et al.*, 2016).

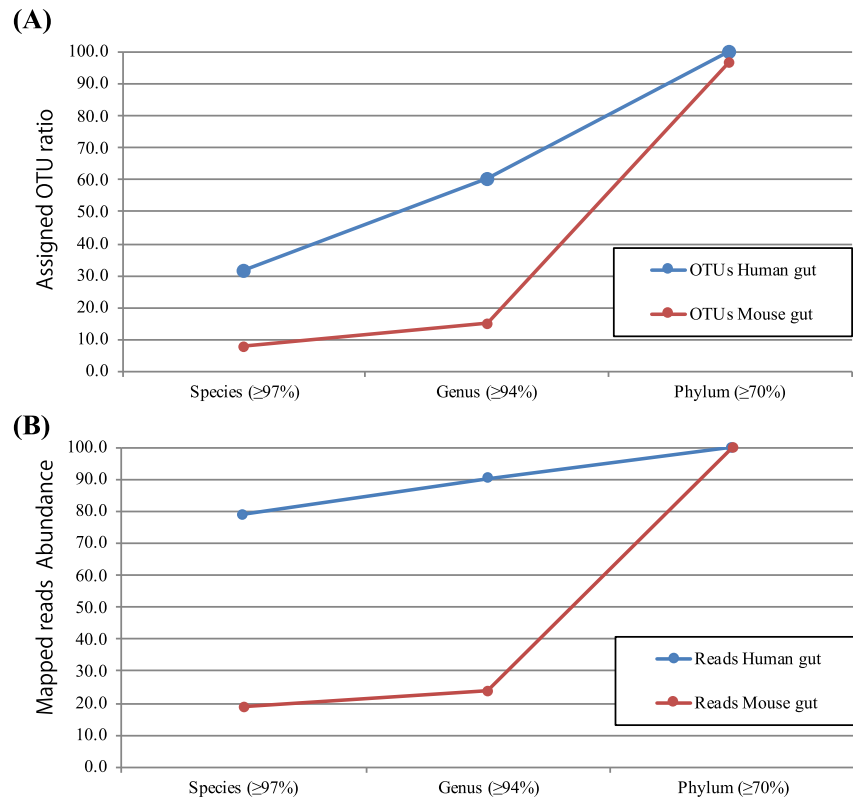


Fig. 6 Taxonomic assignment of 16S data of human and mouse gut microbiomes. (A) Ratios of OTUs taxonomically assigned at the species, genus, and phylum levels. (B) Ratios of 16S reads taxonomically assigned at the species, genus, and phylum levels.

Multivariate Analysis Based on Unifrac Distance

We herein show three examples of multivariate analysis based on the Unifrac distance calculated from 16S data. The Unifrac distance-based PCoA of the nine microbiome samples is shown in Fig. 7(A). The results revealed that the gut samples from the same hosts tended to aggregate to make independent clusters both in the weighted and unweighted Unifrac distance analyses, respectively, in which clusters of the human, orangutan, and pig gut samples were close to each other, but distant from those of the rodent (mouse and rat) gut samples, which were close to each other in the unweighted Unifrac analysis. All of the four environmental samples clearly form a cluster distinct from the gut samples both in the weighted and unweighted Unifrac analyses. The unweighted Unifrac distance analysis considers the presence or absence of the phylogenetically close OTUs/species between the samples, while the weighted Unifrac distance considers differences in the abundance of commonly existing OTUs/species between the samples (Lozupone and Knight, 2005). The present data suggested that the human, orangutan, and pig gut microbiomes were relatively similar to each other in the species members compared with the rodent gut microbiomes, both of which had similar species members. The human and orangutan

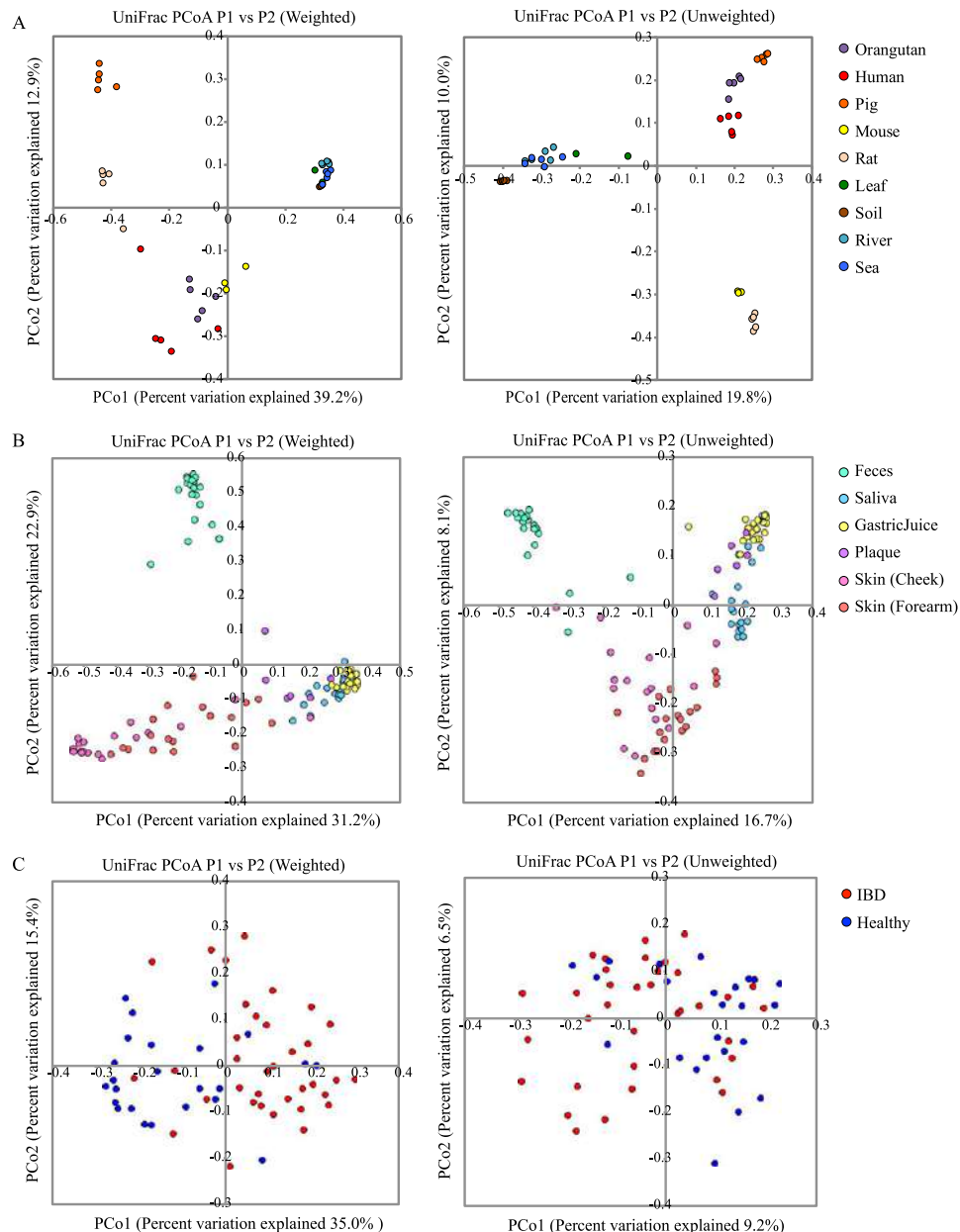


Fig. 7 Unifrac distance-based PCoA of various microbiomes. (A) Weighted and unweighted Unifrac distance-based PCoA of nine microbiomes (orangutans, humans, pigs, mice, rats, leaves, soil, river water, and oceanic water). (B) Weighted and unweighted Unifrac distance-based PCoA of human microbiomes from feces, saliva, gastric juice, plaque, skin (cheek), and skin (forearm). (C) Weighted and unweighted Unifrac distance-based PCoA of human salivary microbiomes from IBD patients and healthy subjects.

gut microbiomes also had relatively similar species abundance, but differed from the pig gut microbiomes in the species abundance. Similarly, although the mouse and rat gut microbiomes had relatively similar species members, the species abundance was different between them. The four environmental microbiomes had relatively similar species members and abundance compared with the different gut microbiomes. In addition, the high similarity between the human and orangutan microbiome structures suggested that the host's genetics more or less contributed to shaping of the gut microbiomes (Goodrich *et al.*, 2014).

We also showed an example for the UniFrac-based PCoA of human microbiomes from different body sites including gut, saliva, plaque, gastric juice, and two skins (cheek and forearm) in Fig. 7(B). The results revealed that the gut samples formed a distinct cluster from the other samples both in the weighted and unweighted UniFrac distance. The saliva, plaque, and gastric juice samples tended to form clusters relatively similar to each other, suggesting that the two oral microbiomes had relatively close community structure with the gastric microbiomes (Tsuda *et al.*, 2015). The two skin samples also formed a distinct cluster from each other though the inter-individual diversities were relatively high comparing to non-skin samples.

Fig. 7(C) further shows an example for the UniFrac-based PCoA analysis of the salivary microbiomes between healthy individuals and patients with IBD. The results found that both samples were segregated from each other both in the weighted and unweighted UniFrac analyses, indicating the typical dysbiosis of the IBD salivary microbiomes (Said *et al.*, 2013). The differences between the two sample sets showed the statistical significance with P -values < 0.01 by Student's t -test.

Periodic Analysis

Several studies have reported the long and short-term periodic changes in the microbial composition of host-related and environmental microbiomes (Kuipers *et al.*, 2000; Thaïss *et al.*, 2014; Takayasu *et al.*, 2017a). A few computational tools have been developed to determine the diurnal periodicity of the microbial abundance in the gut and salivary microbiomes across samples longitudinally collected at time intervals (Hughes *et al.*, 2010; Takayasu *et al.*, 2017a). We herein introduce a method called the periodicity discrimination method (PDM), which uses the Fisher exact test, and can be effective even for small datasets often containing the "zero read" abundance across the samples (Takayasu *et al.*, 2017a). In the PDM, the time series abundance data of each day were first subjected to the min-max normalization, respectively, and the average normalized abundance of each time zone was calculated. Next, the time zones of a day were divided into two groups, and the Fisher exact test p -value was calculated for every combination of the two groups for every possible relative abundance threshold.

We showed diurnal changes in the human salivary microbiome as an example using the PDM. We obtained 16S data from the salivary samples collected from six healthy adults with 4-h intervals for three consecutive days. The periodicity of species abundance in longitudinally collected samples for each individual was evaluated by the PDM. The results revealed that the higher abundant genera tended to have the stronger circadian periodicity, and the genera accounting for 68.4%–89.6% of the total abundance significantly oscillated (Fig. 8(A)). The two major phyla, Bacteroidetes and Firmicutes, were found to have the periodicity of ~ 24 h by auto-correlation coefficient analysis (Fig. 8(B)). The analysis also revealed that the two highly abundant genera *Prevotella* and *Streptococcus* had the opposite oscillation patterns in which the abundance of *Prevotella* increased exclusively in the morning, while that of *Streptococcus* increased exclusively in the evening in the all six subjects, respectively (Fig. 8(C)). We further showed the oscillated patterns in the gene functions of the salivary microbiomes by metagenomic analysis (Fig. 8(D)). The functions of salivary microbiomes were categorized by assigning the metagenomic genes to functions in the KEGG database. The results showed that the circadian oscillation appeared to enrich the functions of environmental responses such as various transporters and two-component regulatory systems in the evening, and those of metabolisms such as the biosynthesis of vitamins and fatty acids in the morning.

Discussion

We presented several analytical methods and tools useful for evaluation of the 16S and metagenomic data of various types of microbiomes. The analysis of a community's diversity suggested the strong selective pressure of host physiologies on the colonization of host-related microbiomes compared with the environmental microbiomes. The taxonomic analysis revealed differences in the dominant species between host-related and environmental microbiomes at the phylum level, and poor species assignment at the genus level for many of the environmental microbiomes. Multivariate analysis based on the UniFrac distance showed that the microbiomes were highly variable according to their habitats in the host-related and the environmental microbiomes. Significant differences in the overall community structure of the salivary microbiome between the IBD patients and healthy individuals suggested the profound association between the salivary microbiome and the host's physiological state. The response of the salivary microbiome to the host's signals may also be the case for the circadian oscillation of the human salivary microbiome, implying the concerted regulation of the human microbiome with the host's homeostasis.

As described above, various microbiomes exhibited similar CRADs, which follow the power-law-like long-tail distribution, and the power exponents are likely to be characteristic to the animal and environmental microbiomes, respectively. The power-law-like long-tailed cumulative distribution is distinct from the Gaussian distribution generally observed in stature and manufacturing quality distributions. In the economic sciences, it has been reported that the richest, accounting for 20% of the world's population, contribute to $\sim 80\%$ of the world's total income (United Nations Development Program, 1992; Table 3.1). Interestingly, this distribution in human society structure is analogous to the CRADs representing the microbial community structure where a small numbers of species predominate the abundance in the community over a vast number of other members. For example, $\sim 20\%$ of the species contribute to 80%–90% of the total abundance in the adult human gut microbiomes as shown in Figs. 3–5. It is of

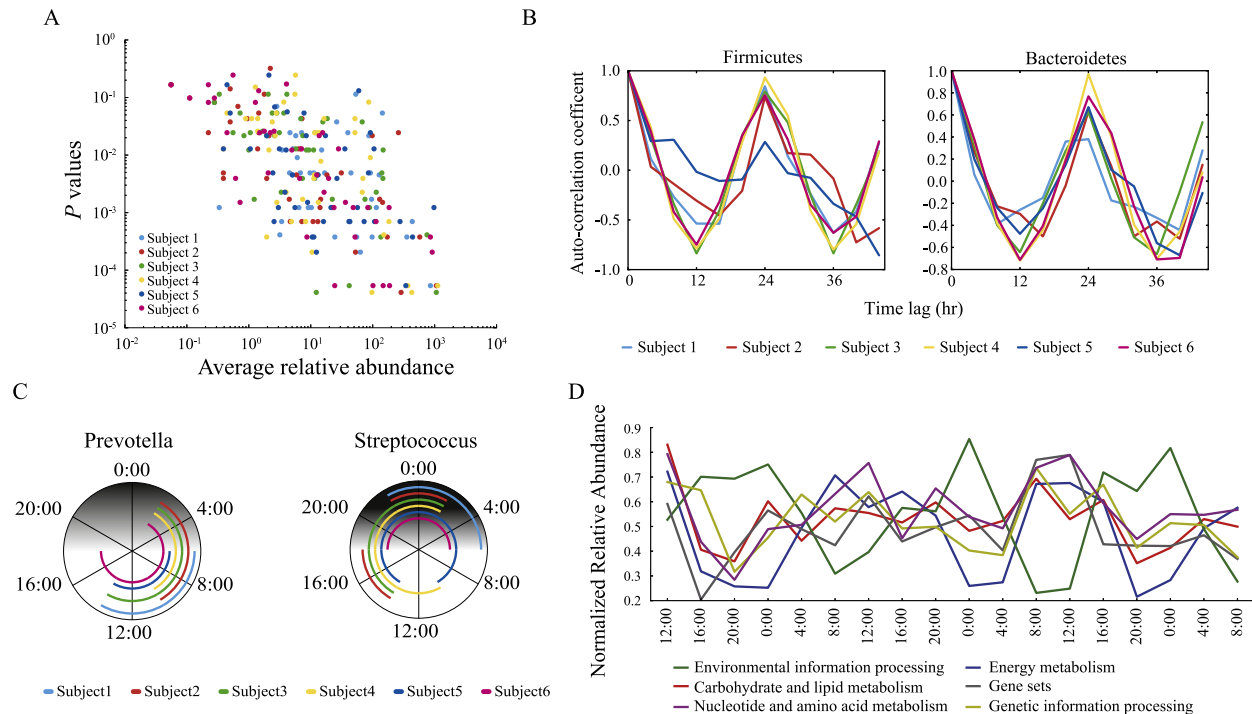


Fig. 8 Circadian rhythm in human salivary microbiomes. (A) Correlations between the relative abundance and circadian periodicity of the genera. (B) Autocorrelation coefficients of the relative abundance of the two major phyla Firmicutes and Bacteroidetes in all the subjects. (C) Circadian proliferation time zones of the two genera *Prevotella* and *Streptococcus*. (D) Diurnal oscillation patterns of the normalized relative gene abundance of several KEGG modules. Reproduced from Takayasu, L., Suda, W., Takanashi, K., *et al.* (2017a). Circadian oscillations of microbial and functional composition in the human salivary microbiome. *DNA Research* 24(3), 261–270.

interest to explore whether there is a similar mechanism for development and establishment of the community structure between human society and gut microbial community.

We also proposed a mathematical model for the power-law-like CRADs in the microbial communities, in which the observed CRADs were reproduced by incorporation of a parameter of spatial competition for the nutrient source between bacterial species (Takayasu *et al.*, 2017b). When the bacteria in the compact assemblage like biofilm or particulate organic matters compete with each other for common nutrient sources, the species with higher proliferation rate could access more resources than other groups with lower rates, resulting in a long-tail structure with large differences in the abundance between these species after repeating the spatial competition. This simple mathematical mechanism could produce power law CRADs similar to those observed in many of the gut microbiomes.

Future Directions

In the environmental microbiomes, it is anticipated to manipulate and improve the environment by controlling bacterial community, which is deeply associated with resource renewal cycles, while the host-related microbiomes are expected to be keys for developing therapies for diseases. Practically, FMT showed the feasibility of medical treatments for several diseases such as *Clostridium difficile* associated diarrhea, irritable bowel syndrome, and acute graft-versus-host disease (Van Nood *et al.*, 2013; Mizuno *et al.*, 2017; Kakihana *et al.*, 2016). However, since mechanisms for successful FMT have been largely unknown, it will be required to explore how donor microbiomes change and shape the alternative form contributing to healing in the patient gut. Our mathematical model for proliferation and restoration of the human gut microbiome will be useful for these FMT studies.

In addition, microbiomes may also be potential biomarkers that respond to changes in host physiologies. This is because the human salivary microbiome is sensitive to systemic changes in a host's physiological states such as circadian rhythms and lesions in the IBD gut as shown here, but likely to show little sensitivity to external stimuli such as dietary intake and administration of medicines including antibiotics (Cabral *et al.*, 2017). The human salivary microbiome was also reported to be a potential marker for pancreatic cancer, which is difficult to diagnose in the early stage (Farrell *et al.*, 2011).

Closing Remarks

OTU number, the Shannon index, and the observed phylum number are shown for five gut microbiomes from orangutans (domestic zoos), humans (Japanese healthy adults), pigs (domestic farms), experimental mice (conventional), and experimental

rats (conventional), foliar microbiomes of *Cerasus* and *Ginkgo*, and three natural environmental microbiomes from soil (a potato field), river fresh water (Saitama, Japan), and oceanic water (Saga, Japan). The data are obtained from 16S data.

CRADs of nine microbiomes indicated in Fig. 3 are shown. The sample numbers are five for orangutans, five for humans, five for pigs, three for mice, five for rats, one for leaf *Cerasus*, one for leaf *Ginkgo*, four for soil, five for river fresh water, and five for oceanic water, respectively. The exponents (β s) of CRADs calculated from the given equation are shown.

Nine microbiomes are from feces of orangutans, humans, pigs, mice, and rats, and from leaves, soil, river water, and oceanic water, respectively.

See also: Ecosystem Monitoring Through Predictive Modeling. Large Scale Ecological Modeling With Viruses: A Review. Natural Language Processing Approaches in Bioinformatics

References

- Aleberg, J., Bjarnason, B.S., De Bruijn, I., *et al.*, 2014. Binning metagenomic contigs by coverage and composition. *Nature Methods* 11, 1144–1146.
- Amann, R.L., Ludwig, W., Schleifer, K.-H., 1995. Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiological Reviews* 59, 143–169.
- Andersson, A.F., Lindberg, M., Jakobsson, H., *et al.*, 2008. Comparative analysis of human gut microbiota by barcoded pyrosequencing. *PLOS ONE* 3, e2836.
- Atarashi, K., Tanoue, T., Ando, M., *et al.*, 2015. Th17 cell induction by adhesion of microbes to intestinal epithelial cells. *Cell* 163, 367–380.
- Atarashi, K., Tanoue, T., Oshima, K., *et al.*, 2013. T reg induction by a rationally selected mixture of Clostridia strains from the human microbiota. *Nature* 500, 232–236.
- Benson, A.K., Kelly, S.A., Legge, R., *et al.*, 2010. Individuality in gut microbiota composition is a complex polygenic trait shaped by multiple environmental and host genetic factors. *Proceedings of the National Academy of Sciences of the United States of America* 107, 18933–18938.
- Cabral, D.J., Wurster, J.I., Flokas, M.E., *et al.*, 2017. The salivary microbiome is consistent between subjects and resistant to impacts of short-term hospitalization. *Scientific Reports* 7, 11040.
- Caporaso, J.G., Kuczynski, J., Stombaugh, J., *et al.*, 2010. QIIME allows analysis of high-throughput community sequencing data. *Nature Methods* 7, 335–336.
- Chaisson, M.J., Huddleston, J., Dennis, M.Y., *et al.*, 2015. Resolving the complexity of the human genome using single-molecule sequencing. *Nature* 517, 608–611.
- Chao, A., 1987. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics* 783–791.
- Chazdon, R.L., Colwell, R.K., Denslow, J.S., *et al.*, 1998. Statistical methods for estimating species richness of woody regeneration in primary and secondary rain forests of northeastern Costa Rica. *Man and the Biosphere Series No. vol. 20*. In: Dallmeier, F., Corniskey, J.A. (Eds.), *Forest Biodiversity Research, Monitoring and Modeling: Conceptual Background and Old World Case Studies*, pp. 285–309.
- Chin, C.-S., Alexander, D.H., Marks, P., *et al.*, 2013. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature Methods* 10, 563–569.
- Cleary, B., Brito, I.L., Huang, K., *et al.*, 2015. Detection of low-abundance bacterial strains in metagenomic datasets by eigengene partitioning. *Nature Biotechnology* 33, 1053–1060.
- Cole, J.R., Wang, Q., Cardenas, E., *et al.*, 2008. The Ribosomal Database Project: Improved alignments and new tools for rRNA analysis. *Nucleic Acids Research* 37, D141–D145.
- DeSantis, T.Z., Hugenholtz, P., Larsen, N., *et al.*, 2006. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology* 72, 5069–5072.
- DiGiulio, D.B., Callahan, B.J., McMurdie, P.J., *et al.*, 2015. Temporal and spatial variation of the human microbiota during pregnancy. *Proceedings of the National Academy of Sciences of the United States of America* 112, 11060–11065.
- Farrell, J.J., Zhang, L., Zhou, H., *et al.*, 2011. Variations of oral microbiota are associated with pancreatic diseases including pancreatic cancer. *Gut* 61 (4), 582–588. doi:10.1136/gutjnl-2011-300784.
- Frank, D.N., Amand, A.L.S., Feldman, R.A., *et al.*, 2007. Molecular-phylogenetic characterization of microbial community imbalances in human inflammatory bowel diseases. *Proceedings of the National Academy of Sciences of the United States of America* 104, 13780–13785.
- Gill, S.R., Pop, M., DeBoy, R.T., *et al.*, 2006. Metagenomic analysis of the human distal gut microbiome. *Science* 312, 1355–1359.
- Giovannoni, S.J., Britschgi, T.B., Moyer, C.L., *et al.*, 1990. Genetic diversity in Sargasso Sea bacterioplankton. *Nature* 345, 60–63.
- Goodrich, J.K., Waters, J.L., Poole, A.C., *et al.*, 2014. Human genetics shape the gut microbiome. *Cell* 159, 789–799.
- Hughes, M.E., Hogenesch, J.B., Kornacker, K., 2010. JTK_CYCLE: An efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets. *Journal of Biological Rhythms* 25, 372–380.
- Kakihana, K., Fujioka, Y., Suda, W., *et al.*, 2016. Fecal microbiota transplantation for patients with steroid-resistant acute graft-versus-host disease of the gut. *Blood* 128, 2083–2088.
- Kanehisa, M., Goto, S., Kawashima, S., *et al.*, 2002. The KEGG databases at GenomeNet. *Nucleic Acids Research* 30, 42–46.
- Kang, D.D., Froula, J., Egan, R., *et al.*, 2015. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ* 3, e1165.
- Karlsson, F.H., Tremaroli, V., Nookaew, I., *et al.*, 2013. Gut metagenome in European women with normal, impaired and diabetic glucose control. *Nature* 498, 99–103.
- Kim, S.-W., Suda, W., Kim, S., *et al.*, 2013. Robustness of gut microbiota of healthy adults in response to probiotic intervention revealed by high-throughput pyrosequencing. *DNA Research* 20, 241–253.
- Kuipers, B., van Noort, G.J., Vosjan, J., *et al.*, 2000. Diel periodicity of bacterioplankton in the euphotic zone of the subtropical Atlantic Ocean. *Marine Ecology Progress Series* 201, 13–25.
- Kuleshov, V., Jiang, C., Zhou, W., *et al.*, 2016. Synthetic long-read sequencing reveals intraspecies diversity in the human microbiome. *Nature Biotechnology* 34, 64–69.
- Kurokawa, K., Itoh, T., Kuwahara, T., *et al.*, 2007. Comparative metagenomics revealed commonly enriched gene sets in human gut microbiomes. *DNA Research* 14, 169–181.
- Leonard, M.T., Davis-Richardson, A.G., Ardisson, A.N., *et al.*, 2014. The methylome of the gut microbiome: Disparate Dam methylation patterns in intestinal *Bacteroides* *doi*. *Frontiers in Microbiology* 5, 361.
- Lozupone, C., Knight, R., 2005. UniFrac: A new phylogenetic method for comparing microbial communities. *Applied and Environmental Microbiology* 71, 8228–8235.
- Manichanh, C., Reeder, J., Gibert, P., *et al.*, 2010. Reshaping the gut microbiome with bacterial transplantation and antibiotic intake. *Genome Research* 20, 1411–1419.
- Meyer, F., Paarmann, D., D'Souza, M., *et al.*, 2008. The metagenomics RAST server – A public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics* 9, 386.
- Miyake, S., Kim, S., Suda, W., *et al.*, 2015. Dysbiosis in the gut microbiota of patients with multiple sclerosis, with a striking depletion of species belonging to clostridia XIVa and IV clusters. *PLOS ONE* 10, e0137429.

- Mizuno, S., Masaoka, T., Naganuma, M., *et al.*, 2017. Bifidobacterium-rich fecal donor may be a positive predictor for successful fecal microbiota transplantation in patients with irritable bowel syndrome. *Digestion* 96, 29–38.
- Moayyedi, P., Surette, M.G., Kim, P.T., *et al.*, 2015. Fecal microbiota transplantation induces remission in patients with active ulcerative colitis in a randomized controlled trial. *Gastroenterology* 149, 102–109. e106.
- Nielsen, H.B., Almeida, M., Juncker, A.S., *et al.*, 2014. Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nature Biotechnology* 32, 822–828.
- Nishijima, S., Suda, W., Oshima, K., *et al.*, 2016. The gut microbiome of healthy Japanese and its microbial and functional uniqueness. *DNA Research* 23, 125–133.
- Noguchi, H., Taniguchi, T., Itoh, T., 2008. MetaGeneAnnotator: Detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. *DNA Research* 15, 387–396.
- Qin, J., Li, R., Raes, J., *et al.*, 2010. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature* 464, 59–65.
- Qin, J., Li, Y., Cai, Z., *et al.*, 2012. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature* 490, 55–60.
- Rasko, D.A., Webster, D.R., Sahl, J.W., *et al.*, 2011. Origins of the *E. coli* strain causing an outbreak of hemolytic–uremic syndrome in Germany. *The New England Journal of Medicine* 365, 709–717.
- Said, H.S., Suda, W., Nakagome, S., *et al.*, 2013. Dysbiosis of salivary microbiota in inflammatory bowel disease and its association with oral immunological biomarkers. *DNA Research* 21, 15–25.
- Sharon, I., Kertesz, M., Hug, L.A., *et al.*, 2015. Accurate, multi-kb reads resolve complex populations and detect rare microorganisms. *Genome Research* 25, 534–543.
- Staley, J.T., Konopka, A., 1985. Measurement of in situ activities of nonphotosynthetic microorganisms in aquatic and terrestrial habitats. *Annual Review of Microbiology* 39, 321–346.
- Takayasu, L., Suda, W., Takanashi, K., *et al.*, 2017a. Circadian oscillations of microbial and functional composition in the human salivary microbiome. *DNA Research* 24 (3), 261–270.
- Takayasu, L., Suda, W., Watanabe, E., *et al.*, 2017b. A 3-dimensional mathematical model of microbial proliferation that generates the characteristic cumulative relative abundance distributions in gut microbiomes. *PLOS ONE* 12, e0180863.
- Tatusov, R.L., Koonin, E.V., Lipman, D.J., 1997. A genomic perspective on protein families. *Science* 278, 631–637.
- Thaiss, C.A., Zeevi, D., Levy, M., *et al.*, 2014. Transkingdom control of microbiota diurnal oscillations promotes metabolic homeostasis. *Cell* 159, 514–529.
- The Human Microbiome Project Consortium, 2012. Structure, function and diversity of the healthy human microbiome. *Nature* 486, 207–214.
- Tsai, Y.-C., Conlan, S., Deming, C., *et al.*, 2016. Resolving the complexity of human skin metagenomes using single-molecule sequencing. *mBio* 7, e01948–01915.
- Tsuda, A., Suda, W., Morita, H., *et al.*, 2015. Influence of proton-pump inhibitors on the luminal microbiota in the gastrointestinal tract. *Clinical and Translational Gastroenterology* 6, e89.
- Turnbaugh, P.J., Hamady, M., Yatsunenko, T., *et al.*, 2009. A core gut microbiome in obese and lean twins. *Nature* 457, 480–484.
- Tyson, G.W., Chapman, J., Hugenholtz, P., *et al.*, 2004. Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428, 37–43.
- United Nations Development Program, 1992. Human Development Report. New York: Oxford University Press.
- Van Nood, E., Vrieze, A., Nieuwdorp, M., *et al.*, 2013. Duodenal infusion of donor feces for recurrent *Clostridium difficile*. *The New England Journal of Medicine* 368, 407–415.
- Venter, J.C., Remington, K., Heidelberg, J.F., *et al.*, 2004. Environmental genome shotgun sequencing of the Sargasso Sea. *Science* 304, 66–74.
- Whitman, W.B., Coleman, D.C., Wiebe, W.J., 1998. Prokaryotes: The unseen majority. *Proceedings of the National Academy of Sciences of the United States of America* 95, 6578–6583.
- Woese, C.R., 1987. Bacterial evolution. *Microbiological Reviews* 51, 221–271.
- Woese, C.R., Fox, G.E., 1977. Phylogenetic structure of the prokaryotic domain: The primary kingdoms. *Proceedings of the National Academy of Sciences of the United States of America* 74, 5088–5090.
- Wu, G.D., Chen, J., Hoffmann, C., *et al.*, 2011. Linking long-term dietary patterns with gut microbial enterotypes. *Science* 334, 105–108.
- Yatsunenko, T., Rey, F.E., Manary, M.J., *et al.*, 2012. Human gut microbiome viewed across age and geography. *Nature* 486, 222–227.

Relevant Website

<https://www.ncbi.nlm.nih.gov/>
NCBI.

Biological Database Searching

Nor A Nor Muhammad, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological databases are databases which store information from biological experiments, especially high throughput molecular experiments. There are many types of biological databases, including sequence databases, protein domain databases, protein structure databases, and non-coding RNA databases, among others. These databases house biological information that can be used for annotation and characterization of new biological data. New inferences may also be obtained from processing data from these databases using bioinformatics.

A simple analysis pipeline that uses biological databases would consist of four main steps (Fig. 1). First would be the data collection step. This can either be from data mining or from generating your own data from sequencing or other omics experiments. Once collected, things to note regarding the data includes the data type, what is known regarding the data and what do you want to know about the data. Hypotheses and objectives of the bioinformatic analysis can then be constructed. These will determine the bioinformatic pipeline that needs to be used. The second stage of the analysis is the selection of relevant databases that needs to be incorporated. For example, annotation of protein sequences would require the UniProt database while the annotation of gene sequences would require the GenBank database. Next, we do the homology searching. The tools used for this step depends on the data collected and the database chosen. For sequence-based data, NCBI BLAST would be the go to choice (Madden, 2013). HMM profile data on the other hand can be processed using the HMMER package (Mistry *et al.*, 2013). Often, these tools are embedded in the databases. Finally, manual curation might be necessary when interpreting the results. The use of multiple databases may either verify results or may produce contradicting results. Results from gene annotation can be mapped onto reference pathways to identify pathways that might be in the data collected.

Biological Databases and Types of Data Stored

Sequence Databases

The largest and main databases for biological data are sequence databases. They house digital versions of nucleotide or amino acid sequences. The nucleotide sequences may correspond to whole genomes, genes, expressed sequence tags (ESTs) or ribonucleic acids (RNAs), while amino acid sequences correspond to protein sequences.

The National Center for Biotechnology Information (NCBI) (Acland *et al.*, 2014) houses one of the biggest nucleotide sequence database named GenBank. GenBank is a collection of annotated nucleotide sequences that are publicly available. It supports bibliographic and biological annotation of each nucleotide sequence. NCBI builds GenBank mainly from data submitted by authors, whole-genome shotgun (WGS) and other data from high-throughput sequencing centers. Over 370,000 formerly

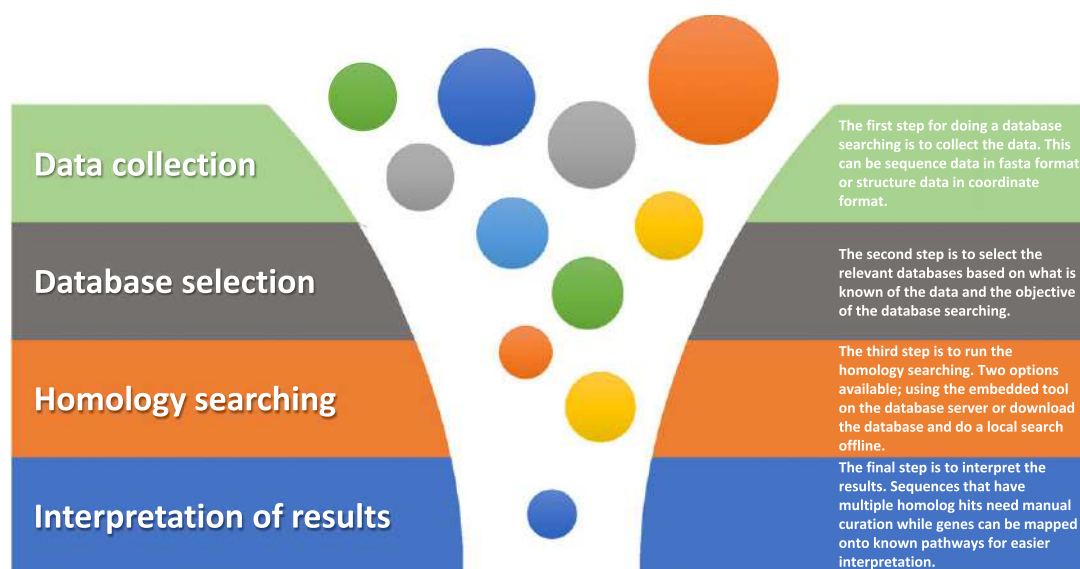


Fig. 1 Overview of analysis pipeline that uses biological databases.

described species are being covered by the GenBank database (Benson *et al.*, 2017). Thus, this makes GenBank one of the main reference for nucleotide sequence analyses.

The European Nucleotide Archive (ENA) is a nucleotide sequence repository developed by The European Bioinformatics Institute (EMBL-EBI) (Leinonen *et al.*, 2010). Compared with GenBank, ENA is broader in the types of nucleotide sequences being stored from raw reads to annotated sequences. Data came from both small-scale research laboratories and large-scale sequencing centers. As of 2016, ENA had collected data from 192,000 studies, with 770 million sequence records linked to 200,000 literature publications. ENA are friendly to both users that look for single sequences as well as bioinformaticians that intend to incorporate the ENA database in their analyses pipeline. Both simple and advanced search tools are provided for users at ENA's website. For bioinformaticians that need programmatic access to ENA, the search and download function are being provided by RESTful services (Toribio *et al.*, 2017).

The National Institute of Genetics (NIG) established the DNA Data Bank of Japan (DDBJ) database. Like GenBank and ENA, DDBJ stores nucleotide sequences that were submitted from laboratories and sequencing centers. Submission of next-generation sequencing data is handled by DDBJ Sequence Read Archive (DRA). There is also BioProject to handle the sequencing project metadata and BioSample for handling sample information. In 2013, DDBJ with their partner, the National Bioscience Database Center (NBDC) launched the Japanese Genotype-phenotype Archive (JGA). This database stores the genotype and phenotype data of individuals. Data from JGA is restricted to certain research based on the individuals consent. The DDBJ center is primarily a supercomputing center that provide services to the research community. Their services include web services, submission systems, data retrieval systems, WebApi and DDBJ Read Annotation Pipeline (Mashima *et al.*, 2016). The latest release of DDBJ was on June 2017, Release 109.0 with 874,923,909 entries.

Together with GenBank and ENA, they form the International Nucleotide Sequence Database Collaboration (INSDC; see "Relevant Websites section"). These three databases shares data submitted to them with each other. They all contribute to one pool of nucleotide sequences under the INSDC initiative. They also work together in the development of the database platform and tools. INSDC champions the work of establishing standards, formats and protocols for nucleotide data and metadata. They have become the model for data sharing in the field of life sciences (Cochrane *et al.*, 2016).

UniProt Knowledgebase or UniProtKB is the main protein sequence resource, which is developed and maintained by EMBL-EBI. It consists of two main components, Swiss-Prot and TrEMBL. Swiss-Prot houses over 500 thousand manually annotated and reviewed protein sequences while TrEMBL has over 60 million computationally annotated protein sequences (Consortium, 2017). The annotation of proteins in Swiss-Prot depends on experimental characterization of proteins, thus each entry has a publication attached to it. UniProtKB also houses a non-redundant database of all known protein sequences called UniParc. It currently has over 120 million sequences (Consortium, 2017). UniProtKB is linked to over 150 databases, thus acting as a central hub for organizing protein information. Its accession number plays a key role as a fixed reference for protein sequences, allowing tagging of proteins in informatics applications (Consortium, 2014).

These sequence databases are used as point of reference for either finding your gene or protein sequence of interest. They can also be used for annotating sequences of interest using sequence alignment tools such as NCBI BLAST (Basic Local Alignment Search Tool) (Camacho *et al.*, 2009). In any characterization of sequence, this will be the first method in finding annotation for your sequence of interest. The web links for all mentioned and other sequence databases is in Table 1.

Protein Domain Databases

Protein domain databases are also called protein families databases. They house information on conserved domains in protein sequences. Each domain may be responsible for a specific function, thus it can also be called a protein functional domain (Finn *et al.*, 2016b). Conserved domains may also be structural features of protein sequences. There are many public protein domain databases, of which eleven are listed in Table 2. The databases differ in the method of identifying the conserved domains. We will be exploring some of the features of each database.

Pfam, a protein family database, is essentially a database of multiple sequence alignments. Each protein family entry in Pfam consists of a set of protein sequences called seed sequences (Sonnhammer *et al.*, 1997). The seed sequences are manually curated to make sure they are true members of the protein family. These seed sequences are then aligned to produce the motif logo of the family/domain. The multiple sequence alignment and motif shows which residues are more conserved than others in the domain. Some regions of the domain will be able to tolerate insertions and deletions while some might not. The multiple sequence alignment is profiled using HMMER, a Hidden Markov Model software for the analyses of biological sequences (Finn *et al.*, 2011). A full

Table 1 List of publicly available sequence database with their names, URLs and hosts/maintainers.

Database name	Database URL	Database host
DNA Data Bank of Japan (DDBJ)	www.ddbj.nig.ac.jp	National Institute of Genetics (NIG)
European Nucleotide Archive (ENA)	www.ebi.ac.uk/ena	European Bioinformatics Institute (EMBL-EBI)
GenBank	www.ncbi.nlm.nih.gov/genbank	National Center for Biotechnology Information (NCBI)
Reference Sequence (RefSeq)	www.ncbi.nlm.nih.gov/refseq	National Center for Biotechnology Information (NCBI)
Universal Protein Resource (UniProt)	www.uniprot.org	European Bioinformatics Institute

Table 2 List of public protein domain/family database with their names, URLs and hosts/maintainers.

Database name	Database URL	Database host
CDD (Conserved Domain Database)	www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml	National Center for Biotechnology Information (NCBI)
Gene3d	gene3d.biochem.ucl.ac.uk	University College London
HAMAP (High-Quality Automated and Manual Annotation of Proteins)	hamap.expasy.org	Swiss Institute of Bioinformatics
Panther (Protein Analysis Through Evolutionary Relationship)	www.pantherdb.org	University of Southern California
Pfam	pfam.xfam.org	European Bioinformatics Institute
PIRSF (Protein Information Resource Super Family)	pir.georgetown.edu/pirwww/dbinfo/pirsf.shtml	Georgetown University Medical Center
PRINTS	130.88.97.239/PRINTS/index.php	The University of Manchester
PROSITE (Database of protein domains, families and functional sites)	prosite.expasy.org	Swiss Institute of Bioinformatics
ProDom	prodom.prabi.fr/prodom/current/html/home.php	French National Institute for Agricultural Research (INRA)
SMAT (Simple Modular Architecture Research Tool)	smart.embl-heidelberg.de	European Molecular Biology Laboratory
SUPERFAMILY (HMM library and genome assignments server)	supfam.org/SUPERFAMILY	MRC Laboratory of Molecular Biology
TIGRFAM	www.jcvi.org/cgi-bin/tigrfams/index.cgi	J. Craig Venter Institute (JCVI)

alignment of the protein family is built by aligning all members with the HMM profile (Eddy, 1998). This full alignment is useful for identifying features of the protein family such as the distribution of the protein family in living species, the size of the protein family, the level of conservation of the domain of interest as well as identifying homologs of the protein sequence being analyzed. The relationship between the protein family and taxonomic hierarchy can also be studied by generating the taxonomic tree of all the members of the protein family. Each Pfam entry is also cross referenced with literatures and UniProtKB in order to access annotations of the protein family (Finn *et al.*, 2016b). Current version of Pfam is version 31.0, released in March 2017 with 16,712 entries.

New publications that describe protein characterization usually provide one or two proven homologs of the protein. No rules or definitions are provided for the identification of protein homologs from all other organisms. Furthermore, the grouping of protein into functional groups differ from case to case and no fixed heuristics are available that work in most cases. Thus, TIGRFAM was developed to facilitate automated annotations of proteins by providing a library of protein family definitions. Like Pfam, TIGRFAM is a manually curated database. However, TIGRFAM has additional features which make it suitable for automated genome annotation. The annotation of a protein family is always traced back to its original source that is not linked to any computational annotation pipeline. Stringent rules are applied in the selection of seed sequences. Multiple sequence alignment of the seed sequences is checked for misalignments, inconsistent domain architecture and altered active or binding sites. Hidden Markov Models of high quality are then produced from the alignments. The aim is that each HMM produced by TIGRFAM pipeline can be the representative of an expert curator who operates with BLAST-like speeds (Haft *et al.*, 2012). The current release version is 15.0 with 4488 families.

Biologist turn to protein family databases, such as Pfam and TIGRFAM to annotate protein sequences that do not have any matched homologs from protein sequence databases. Both Pfam and TIGRFAM uses HMM profiles, but they differ in their intended use cases. Pfam tries to be as broad as possible to allow for detection of distant homologs to a protein family in the Pfam database. Thus, each protein family in Pfam encodes a small part of the functional domain, with a conservation profile indicative of high sensitivity. The small conserved domain allows the generated HMM to match distant homologs. TIGRFAM on the other hand was developed based on the concept of equivalogs, which are genes or encoded proteins that have a common conserved function since their last common ancestor. This contrasts with the definition of orthologs where only the evolutionary history is concerned (Haft *et al.*, 2003); as such equivalogs may include more than orthologs (i.e., xenologs) and orthologs may not be equivalogs (because of neofunctionalization).

The Conserved Domain Database (CDD) was developed to accelerate the annotation of new protein sequences. Due to the exponential increase in the size of public sequence data, it is impractical to keep aligning new sequences to each sequence in the database for characterization purposes. The strategy took by the developers of CDD was to identify representative sequences for conserved domains. This representative set is unlikely to grow at the same exponential rate as new protein sequences (Marchler-Bauer *et al.*, 2014). CDD collects its protein and protein domain models from Pfam (Sonnhammer *et al.*, 1997), SMART (Letunic *et al.*, 2014), COG Database (Tatusov *et al.*, 2003), TIGRFAM (Haft *et al.*, 2012), NCBI Protein Cluster Database (Klimke *et al.*, 2008) and in house curations efforts (Marchler-Bauer *et al.*, 2003). Each CDD entry is crosslinked to other databases for annotation and search purposes. They include NCBI Entrez Search, Entrez/protein, Entrez/gene, molecular modelling database (MMDB), NCBI Biosystems, PubMed and PubChem (Marchler-Bauer *et al.*, 2014). Current live version of CDD is v3.16 with 56,066 entries, where 12,805 entries are by CDD curation efforts. The entries are grouped into 5697 multi-model superfamilies.

PROSITE is the first database created for identification of protein families or domains. Each entry in the database is a motif descriptor dedicated for identification of protein families. The description can be of two types, patterns or profiles (Sigrist *et al.*, 2002). Patterns are regular expression of short sequence motif that often relate to functionally or structurally important residues. Profiles on the other hand are weight matrices that describe the protein family or domain (Sigrist *et al.*, 2013). Annotation rules in PROSITE are called ProRules. ProRules define annotations relating to protein sequence such as the active site, ligand binding residues and the conditions relating to the annotations (Sigrist *et al.*, 2005). To search protein sequence of interest against PROSITE, the ScanProsite tool can be used. ScanProsite can also search for matches in UniProtKB and Protein Data Bank (PDB) structure database (De Castro *et al.*, 2006). The latest PROSITE release was on 5th July 2017 with 1309 patterns, 1189 profiles and 1208 ProRules.

ProDom is a protein domain database that uses automation instead of relying on manual curation like in Pfam (Finn *et al.*, 2016b) and SMART (Letunic *et al.*, 2014). ProDom algorithm applies automated clustering towards a database of non-redundant sequences from Swiss-Prot, TrEMBL, complete proteomes on the ExPASy server and on the Proteome Analysis pages. The clustering program used is MKDOM2. It uses PSI-BLAST to look for homologous domains amongst the sequences in the database. Sequences that belong to a cluster become a ProDom family and are removed from the database. This is iterated until all the sequences in the database have been exhausted (Servant *et al.*, 2002). In the latest version of ProDom, 3D structure information is used during the generation of ProDom families. 3D structure information is obtained from the Structural Classification of Proteins (SCOP) database (Andreeva *et al.*, 2004). ProDom-CG is a subset of the ProDom database; members of ProDom-CG are restricted to protein sequences from completely sequenced genomes. ProDom also hosts the ProDom-SG, a server for structural genomics. This server identifies protein candidates suitable for crystallization that have potential in having new folds that are yet to be seen in proteins (Bru *et al.*, 2005). The live version of ProDom currently has 1,718,157 domain families (that have at least two members); 426,997 families have links to PDB, 128,882 families have links to PROSITE and 823,801 families have links to Pfam.

PRINTS is a protein 'fingerprint' database. Where other domain databases have entries of single motif representing a conserved domain, 'fingerprint' consist of multiple short motifs that represent a protein family or structural features (Attwood and Beck, 1994). Thus, 'fingerprint' can be more powerful compared to single motif approaches as it depends on multiple motifs. 'Fingerprint' can also annotate the relationships between protein families (domains) as matches to a 'fingerprint' will have multiple domains. Given this, the hierarchical or evolutionary relationship between matches of a 'fingerprint' may also be inferred (Attwood *et al.*, 2012). PRINTS database is currently at version 39.0 with 1950 entries and 11,625 motifs.

SUPERFAMILY is a library of Hidden Markov Models (HMMs) representing protein of known structure. The structural information is obtained from the SCOP database, where proteins are grouped together at several levels; one such level is 'superfamily'. Proteins in a superfamily group shares a common ancestor, thus having functional, structural and sequence similarity. The SUPERFAMILY database was derived from this level because this is the level with most distantly related domains. The framework behind SUPERFAMILY database is HMMs, similar to Pfam (Sonnhammer *et al.*, 1997), SMART (Letunic *et al.*, 2014) and TIGRFAM (Haft *et al.*, 2012) databases. The difference is that SUPERFAMILY only covers proteins with known structure where else the other databases covers all protein sequences (Gough and Chothia, 2002). The SUPERFAMILY database has also been integrated with InterPro (Finn *et al.*, 2016a) and Gene Ontology terms. Abstract from InterPro entries have been imported into SUPERFAMILY. SUPERFAMILY database also provides the users with taxonomic visualization of the families in the database. Users can also search for domain architectures with similar function to find related domains to domains of interest (Wilson *et al.*, 2009). Current SUPERFAMILY database has 1962 entries based on SCOP 1.75 superfamilies.

The Protein Analysis Through Evolutionary Relationships or PANTHER is a database resource of protein families, subfamilies, phylogenetic trees and functions. PANTHER aims to practically infer functions of genes and proteins in large sequence databases. It uses phylogenetic tree approach to extrapolate experimental information from model organisms onto other sequences. Trees generated in PANTHER describes the evolutionary events in gene family histories. One tree is generated for each family. Nodes on the tree are annotated with gene attributes. There are three types of gene attributes annotated in PANTHER. They are 'subfamily membership', 'protein class' and 'gene function'. The trees also include information on speciation and gene duplication events (Mi *et al.*, 2013). Current live version of the database is PANTHER12.0, with 103 genomes, 1,378,820 genes, 14,710 families, and 76,032 subfamilies.

Gene3D derives its domain entries from the CATH (class, architecture, topology & homology) database. CATH is a protein domain structure database. Structures from the Protein Data Bank (PDB) are broken down into smaller domain structures. These domains are then grouped into superfamilies if evidences of common ancestry were found (Sillitoe *et al.*, 2015). From these superfamilies, profile HMMs are generated and put into the Gene3D library.

Protein Structure Databases

Protein structure databases stores coordinate data from protein and other biomolecules structures. The main source for the data are X-ray crystallography experiments. These databases also provide tools for searching proteins that have their structure solved, tools for comparing input proteins with protein structures in the database, visualization tools for generating 3D images of the protein structures in the databases and annotations for each protein structure entry. Advance users may also download raw coordinate data for custom computational biology works. Main databases in this field include the Protein Data Bank, CATH database and SCOP2 database.

The Protein Data Bank (PDB) was founded in 1971 for archiving biomolecular macromolecular crystal structures (Berman *et al.*, 2014). The emergence of new structural biology technologies has forced PDB to evolve into what it is today. Today, after over 46 years, PDB is the largest public database that hosts structural data of biomolecules. PDB employs the best of current data

science technologies and practices. Everything is centralized and various checkpoints are implemented in the PDB pipeline to maintain data integrity and consistency. In 2013, more than 400 million coordinate sets have been downloaded from PDB partner sites. The well-developed data structure and data processing practices has enabled the development of resources for small molecules and ligands such as ChEMBL (Bento *et al.*, 2014), DrugBank (Law *et al.*, 2013), BindingDB (Gilson *et al.*, 2016), BindingMOAD (Ahmed *et al.*, 2014) and PDBind (Liu *et al.*, 2014a). There are also resources developed for protein structure classification and annotation such as CATH (Cuff *et al.*, 2008), SCOP2 (Andreeva *et al.*, 2013) and PDBsum (De Beer *et al.*, 2013). Last but not least, there are specialty annotation resources developed from PDB data such as Protein Data Bank of Transmembrane Proteins (PDBTM) (Kozma *et al.*, 2012), ArchDB (Bonet *et al.*, 2013) and 3did (Mosca *et al.*, 2013).

The CATH (class, architecture, topology & homology) database is a hierarchical database of domains. The lowest level of the hierarchy is the H-level or the homologous superfamily. At this level, the domains share significant sequence similarity, structural similarity and/or functional similarity. The next level up is the T-level or the fold or topology groups. Domains in a group at this level share only structure similarity. Topology groups that have similarities in the arrangement of their secondary structures are then put into the same architecture group. This is also called the A-level. The top level of the hierarchy is the C-level or class level. The groupings at this level depends on the percentage composition of α -helices and β -strands in the domains (Pearl *et al.*, 2003).

SCOP2 database organizes protein structures based on their structure and evolutionary relationships. The main feature of SCOP2 compared to other protein structure database is its focus on protein evolution. The relationship between proteins are represented as a complex network of nodes. There are four major categories of relationships in SCOP2, protein types, evolutionary events, structural classes and protein relationships which have three subtypes: structural, evolutionary and other. SCOP2 uses five levels of evolutionary classification: species, protein, family, superfamily and hyperfamily. At the species level, each gene product is considered as one entity and is represented by its full sequence length. At the protein level, orthologous proteins are grouped together. Conserved residues between proteins forms the next level, which is the family level. Sharing of common structural regions then forms the fourth level, i.e., the superfamily level. The highest level is the hyperfamily level. A domain at this level are shared between superfamilies. These can be smaller than structural domains and was introduced to cater highly populated superfamilies (Andreeva *et al.*, 2013). There are also other protein/biomolecule structure related databases as listed in Table 3.

Non-Coding RNA Databases

Non-coding RNAs or ncRNA are RNAs that do not code for proteins. They are abundant and have important roles in cellular processes. There are several known types of ncRNA including snoRNAs, scaRNAs, miRNAs, siRNAs, snRNAs, exRNAs, piRNAs and long ncRNAs. Small nucleolar RNAs (snoRNAs) are a class of small RNA that assist in chemical modifications of other RNAs such as ribosomal RNAs. A subset of snoRNAs are located in Cajal bodies, thus sometimes called scaRNAs. Micro RNA (miRNA) is a class of small non-coding RNA that can be found in plants, animals and viruses. They are responsible for RNA silencing and post transcriptional regulation of gene expression. Small interfering RNAs (siRNAs) also called short interfering RNAs or silencing RNAs, are nucleotides up to 21 residues long that can interfere with protein translation. They do this by binding to and promoting the degradation of messenger RNAs (mRNAs). Small nuclear RNAs (snRNAs) are confined to the nucleus. They are involved in splicing and other RNA processing (Mattick and Makunin, 2006). Extracellular RNA (exRNA) is a class of RNA found extracellularly. In humans, it has been found in bodily fluids such as blood and saliva. They have many roles depending on types but generally involved in cell-to-cell communication. When the RNA reaches the target cell, it initiates certain biological process

Table 3 List of public protein/biomolecules structure databases as well as related resources.

Database name	Database URL	Database host
3did (Three-Dimensional Interacting Domains)	3did.irbbarcelona.org	Institute for Research in Biomedicine
ArchDB	sbi.imim.es/archdb	Structural Bioinformatics Lab
BindingDB	www.bindingdb.org	University of California
BindingMOAD	bindingmoad.org	University of Michigan
CATH	www.cathdb.info	University College London
ChEMBL	www.ebi.ac.uk/chembl	European Molecular Biology Laboratory
DrugBank	www.drugbank.ca	University of Alberta
ModBase: Database of Comparative Protein Structure Models	modbase.compbio.ucsf.edu	University of California
PDB (The Protein Data Bank)	www.rcsb.org	San Diego Supercomputer Center
PDBe (Protein Data Bank in Europe)	www.ebi.ac.uk/pdbe	European Bioinformatics Institute
PDBind	www.pdbbind.org	University of Michigan
PDBind-CN	www.pdbbind.org.cn	Shanghai Institute of Organic Chemistry
PDBj (Protein Data Bank Japan)	pdbj.org	Osaka University
PDBsum	www.ebi.ac.uk/pdbsum	European Bioinformatics Institute
PDBTM: Protein Data Bank of Transmembrane Proteins	pdbtm.enzim.hu	Institute of Enzymology
SCOP2 (Structural Classification of Proteins 2)	scop2.mrc-lmb.cam.ac.uk	MRC Laboratory of Molecular Biology
SWISS-MODEL Repository	swissmodel.expasy.org/repository	Swiss Institute of Bioinformatics
Protein Model Portal	www.proteinmodelportal.org	Swiss Institute of Bioinformatics

(Benner, 1988). Piwi-interacting RNA (piRNA) is a class of non-coding RNA abundant in animal cells. They form RNA-protein complexes by binding to piwi proteins and are associated with silencing of genetic elements (Seto *et al.*, 2007). lncRNA or long non-coding RNA are nucleotides with over 200 residues but do not encode proteins. They form majority of the human transcriptome but their function remain largely unknown (Mattick and Rinn, 2015).

Generally, ncRNA databases and resources can be grouped into two types, an integrated type which houses annotations of ncRNAs regardless of the subtype and another group which focuses on a specific type of ncRNA. Examples of centralized ncRNA databases are RNAcentral and NONCODE. They provide annotations of the different types of ncRNA by doing data mining and merging of information from other databases. RNAcentral for example, combined information from 42 external databases providing a one stop solution for annotation of ncRNA sequences. There are three main tools in RNAcentral that can be used for finding ncRNA of interest. The tools are text-based searching, sequences-based searching and genome browser. In the text-based searching, one can search based on gene, species, publication and author. The sequence-based searching accepts RNA sequence in FASTA format or an RNAcentral ID. This is similar to searching with web NCBI BLAST (see “Relevant Websites section”). Lastly is the genome browser tool. Currently, there are 12 genomes available in the genome browser. They are *Arabidopsis thaliana*, *Bos taurus*, *Bombyx mori*, *Caenorhabditis elegans*, *Canis familiaris*, *Danio rerio*, *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus*, *Pan troglodytes*, *Rattus norvegicus* and *Schizosaccharomyces pombe*. With the genome browser, one can go to the chromosome of interest to see the genes, transcripts and RNAcentral entries at that location in the genome. The current 8th release has over 13.1 million sequences.

NONCODE is another ncRNA knowledgebase, like the RNAcentral. However, NONCODE has 17 genomes in its genome browser and also does a lot of literature mining to gather ncRNA data. It houses most ncRNA types but focuses on lncRNA (long non-coding RNA). Apart from BLAST and genome browser, NONCODE also provides an ID conversion tool and a pipeline for identification of lncRNA. Other sources of ncRNA can be seen in Table 4.

Applications/Use Cases

The mass amount of biological data housed in various databases is an invaluable resource for fields related to biology: biotechnology, systems biology, bioinformatics, genomics, transcriptomics, proteomics and metabolomics to name a few. In the Fig. 2 below are examples of the databases described in the previous section grouped into their respective types.

Characterization of an Unknown Sequence

Due to the wealth of databases and tools, characterization of unknown sequences, both nucleotides and amino acid sequences can be done relatively quickly. The limitation is the information in the databases itself. If the unknown sequence is truly novel and has

Table 4 List of non-coding RNA databases and resources with their corresponding URLs and hosts.

Database name	Database URL	Database host
5SrrNAdb	combio.pl/rRNA	Adam Mickiewicz University
ATTRACT (A Database of RNA Binding Proteins and Associated Motifs)	https://attract.cnice.es/	The Centro Nacional de Investigaciones Cardiovasculares Carlos III (CNIC)
CRW (Comparative RNA Web Site and Project)	www.rna.cccb.utexas.edu	The University of Texas at Austin
lncRNAdb	www.lncrnadb.org	Garvan Institute of Medical Research
lncBase	http://carolina.imis.athena-innovation.gr/diana_tools/web/index.php?r=lncbasev2%2Findex	University of Thessaly
LNCipedia 4.1	lncipedia.org	
miRBase	www.mirbase.org	University of Manchester
NONCODE	www.noncode.org	Chinese Academy of Sciences
RBPDB (The database of RNA-binding protein specificities)	http://rbpdb.ccb.utoronto.ca/	University of Toronto
Rfam	rfam.xfam.org	European Bioinformatics Institute
RNAcentral	rnacentral.org	European Bioinformatics Institute
RNApathwaysDB	www.genesilico.pl/rnapathwaysdb	International Institute of Molecular and Cell Biology in Warsaw
SILVA (High Quality Ribosomal RNA Databases)	www.arb-silva.de	Max Planck Institute for Marine Microbiology and Jacobs University
snOPY (snoRNA Orthological Gene Database)	snoopy.med.miyazaki-u.ac.jp	University of Miyazaki
sRNAMap	srnamap.mbc.nctu.edu.tw	National Chiao-Tung University
Tarbase	http://andrea.imis.athena-innovation.gr/diana_tools/web/index.php?r=tarbasev8%2Findex	University of Thessaly

Sequence Databases	Protein Domain	Protein Structure	ncRNA Databases
• GenBank • ENA • DDBJ • RefSeq • UniProt	• Pfam • TIGRFAM • CDD • PROSITE • ProDom • SUPERFAMILY • Panther	• PDB • CATH • SCOP2 • ModBase • SWISS-MODEL • ArchDB	• NONCODE • lncRNAdb • RNAcentral • Rfam • LncBase • ATTRACT

Fig. 2 Example databases for different types of biological data.

not be seen before, *in silico* characterization might then be limited. Given an unknown sequence, the first instinct would be to figure out whether there are similar sequences to the sequence in the databases. Using tools based on homology searching such as BLAST (Camacho, 2015), homologs of the unknown sequence can be identified from the databases. If the homologs are well known, their annotations can be ascribed as putative annotations of the unknown sequence.

Exploring Sequence Patterns and Profiles

Now biologist and researchers in general can explore vast number of domain databases with ease. Each database is equipped with search function and various filters for finding domains of interest. Once the domain of interest is found, users can explore the features and annotations of the domain. Most domain databases provide the motif logo, the multiple sequence alignment and the description of the domain. We can infer a lot from the motif logo; we can observe which residues at which position in the domain are conserved. We can also see alternative residues for positions that are not absolutely conserved in the members of the domain. Databases such as Pfam and TIGRFAM also allows the download of the HMM profile of the domain. With the HMM profile, users can search for sequences that contain the domain or align sequences of interest with the domain. Generating our own HMM profile is also easily done using HMMER. Firstly, we generate a multiple sequence alignment containing the domain of interest. Then using HMMER (hmmbuild), a HMM profile can be generated. This profile can then be searched against sequence databases to find members having the same domain (Mistry *et al.*, 2013).

Mining for Immune Epitopes

There are now databases and resources that provides information on immune epitopes that have been found. From these resources, tools that can predict the existence of immune epitopes in new data have been generated. The main resource for analyzing immune epitopes is the Immune Epitope Database and Analysis Resource (IEDB) (Peters *et al.*, 2005). This resource provides both simple search engine for finding immune epitopes that have been deposited in the database and analysis tools and scripts for prediction of immune epitopes in sequence data. The analysis tools can be found at the website provided in “Relevant Websites section”. There are four categories of analysis tools being provided at IEDB. They are T Cell Epitope Prediction Tools, B Cell Epitope Prediction Tools, Analysis Tools and IEDB Analysis Resource Labs. In the T Cell Epitope Prediction Tools category, there are six tools provided. The use of these tools allows for the identification of peptides that may bind to MHC molecules, peptides that may be T-cell epitopes and the ability of peptides to induce an immune response. In the B Cell Prediction Tools category, there are five tools provided. They allow the prediction of linear B cell epitopes from amino acid sequences, prediction of B cell epitopes from protein structures and structure prediction of antibodies. Tools for analysis of known epitopes have also been developed. The tools can do population coverage analysis, epitope conservation analysis, epitope cluster analysis and mapping of mimotope (Kim *et al.*, 2012).

Global Identification of Protein Post-Translational Modifications

The development of protein sequence databases has accelerated the field of global post-translational modifications identification. There are many types of protein post-translational modifications including phosphorylation, glycosylation, amidation, hydroxylation, methylation and others. Due to the wealth of post-translational modification types, many resources and tools have been developed for the identification of post-translational modifications. As an example, there are 17 or more databases that are related to protein phosphorylation. Choosing appropriate database or resource will depend on the analysis intended or biological question to be answered. For prediction or annotation purposes, there are also databases and tools that can predict multiple post-translational modifications. Some of them include DbPTM 3.0 (Lu *et al.*, 2012), PhosphoSitePlus (PSP) (Hornbeck *et al.*, 2014), PTMcode v2 (Minguez *et al.*, 2014), Compendium of Protein Lysine Modifications (CPLM) (Liu *et al.*, 2014b) and ProteomeScout (Matlock *et al.*, 2014). For doing custom advance bioinformatics, one can download data from post-translational modifications databases and use as training data for machine learning algorithms. Once trained, the machine learning (e.g., neural network or support vector machine) algorithm can identify putative post-translational modifications when input with a new protein sequence data (Gnad *et al.*, 2010).

Characterizing and Functional Assignment of Non-Coding RNAs

Non-coding RNAs from sequencing experiments or data mining efforts can be characterized via several methods depending on what is known about the dataset and the objective of the characterization. If none is known regarding the ncRNA sequences dataset, a good strategy would be to start with a BLAST search against the RNAcentral or NONCODE database. Sequences with identified homologs are ascribed with functions and type of the homologs. Those with identified homologs can be grouped into their putative ncRNA types. Further analysis can be done by doing a sequence search against curated database of a specific ncRNA type. The search will either confirm the previous annotation using RNAcentral (or NONCODE) database or give a different annotation. Assuming the curated database has less errors compared to computationally curated databases, the new function can be assigned to those sequences in the dataset.

Additionally, it may be worthwhile to characterize the ncRNA dataset with all available ncRNA databases. This can be done through at least two strategies. The first strategy is to download all the sequences from the databases. Once downloaded, the sequences can be appended into a single flat file. This flat file can then be converted into a BLAST database using makeblastdb, which is part of the NCBI BLAST software package. Details on generating a BLAST database can be found in the official BLAST guide (see “Relevant Websites section”). The ncRNA sequence dataset can then be searched using the local install of the NCBI BLAST against the custom BLAST database created. The best homologs for each query sequence across all ncRNA databases (source of the custom database) can then be found.

The second strategy is to search each database separately then download all the results. For easier processing, the results can be input into database applications such as mysql or MS Access. With the pooled results, one can now proceed to reduce the results into a non-redundant list. This can be done easily using remove redundant tools in those applications. Some sequences might have different annotations from different database. Manual curation is best at this stage where the investigator can select the best annotation based on what is known regarding the sequences. Once ncRNA sequence databases have been exhaustively searched, sequences that are yet to be annotated can be searched against the Rfam RNA domain database (Daub *et al.*, 2015) or any other RNA motif databases including ATtRACT (Giudice *et al.*, 2016) and RBPBD (Cook *et al.*, 2011).

Future Directions

There are many classes of biological databases today and in each class, there is often more than one database. The protein domain database class for example, twelve databases have been listed in this article. The challenge in having so many public databases of the same class is how to control for overlap of information; this is from the perspective of the database developers. On the other hand, from the user’s perspective, how can one integrate the results from so many databases into his or her analysis pipeline. Each database often has its own ID and set of annotations making the removal of redundant results between databases a not trivial task. Thus, the effort of building a centralized database must continue. In the case of domain databases, InterProScan is one of the efforts for doing search against multiple domain databases (Jones *et al.*, 2014).

Another area of gap in the field of biological databases is the integration of different database class. An integrated database would consist of DNA sequences, RNA sequences, protein sequences, protein structures, protein domains, RNA domains etc. The reason this has yet to be seen is due to the challenges that comes with integration of biological data. The biggest challenge is figuring out the relationships between the different molecular entities. Biological data is big and after integration, it will be even bigger. Thus, there is also the challenge of navigating through the data and data visualization. Finally, is the issue of storage and management due to the sheer size of these integrated databases.

Closing Remarks

Biological databases have been and will always be an essential part of any molecular biology research. The wealth of information gathered since the start of the genomic era is priceless and have led to many new discoveries since then. Developments in biological databases must continue in line with the generation of new biological data.

See also: Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics

References

- Acland, A., Agarwala, R., Barrett, T., *et al.*, 2014. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 42 (Database issue), D7.
- Ahmed, A., Smith, R.D., Clark, J.J., Dunbar Jr, J.B., Carlson, H.A., 2014. Recent improvements to Binding MOAD: A resource for protein – Ligand binding affinities and structures. *Nucleic Acids Research* 43 (D1), D465–D469.
- Andreeva, A., Howorth, D., Brenner, S.E., *et al.*, 2004. SCOP database in 2004: Refinements integrate structure and sequence family data. *Nucleic Acids Research* 32 (Suppl. 1), D226–D229.

- Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., Murzin, A.G., 2013. SCOP2 prototype: A new approach to protein structure mining. *Nucleic Acids Research* 42 (D1), D310–D314.
- Attwood, T., Beck, M., 1994. PRINTS – A protein motif fingerprint database. *Protein Engineering, Design and Selection* 7 (7), 841–848.
- Attwood, T.K., Coletta, A., Muirhead, G., *et al.*, 2012. The PRINTS database: A fine-grained protein sequence annotation and analysis resource – Its status in 2012. *Database* 2012, bas019.
- Benner, S.A., 1988. Extracellular 'communicator RNA'. *FEBS Letters* 233 (2), 225–228.
- Benson, D.A., Cavanaugh, M., Clark, K., *et al.*, 2017. GenBank. *Nucleic Acids Research* 45 (Database issue), D37–D42.
- Bento, A.P., Gaulton, A., Hersey, A., *et al.*, 2014. The ChEMBL bioactivity database: An update. *Nucleic Acids Research* 42 (D1), D1083–D1090.
- Berman, H.M., Kleywegt, G.J., Nakamura, H., Markley, J.L., 2014. The Protein Data Bank archive as an open data resource. *Journal of Computer-Aided Molecular Design* 28 (10), 1009–1014.
- Bonet, J., Planas-Iglesias, J., Garcia-Garcia, J., *et al.*, 2013. ArchDB 2014: Structural classification of loops in proteins. *Nucleic Acids Research* 42 (D1), D315–D319.
- Bru, C., Courcelle, E., Carrère, S., *et al.*, 2005. The ProDom database of protein domain families: More emphasis on 3D. *Nucleic Acids Research* 33 (Suppl. 1), D212–D215.
- Camacho, C., 2015. BLAST+ Release Notes.
- Camacho, C., Coulouris, G., Avagyan, V., *et al.*, 2009. BLAST+: Architecture and applications. *BMC Bioinformatics* 10, 421.
- Cochrane, G., Karsch-Mizrachi, I., Takagi, T., 2016. The International nucleotide sequence database collaboration. *Nucleic Acids Research* 44 (D1), D48–D50.
- Consortium, U., 2014. UniProt: A hub for protein information. *Nucleic Acids Research*. gku989.
- Consortium, U., 2017. UniProt: The universal protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Cook, K.B., Kazan, H., Zuberi, K., Morris, Q., Hughes, T.R., 2011. RBPDB: A database of RNA-binding specificities. *Nucleic Acids Research* 39 (Suppl. 1), D301–D308.
- Cuff, A.L., Sillitoe, I., Lewis, T., *et al.*, 2008. The CATH classification revisited – Architectures reviewed and new ways to characterize structural divergence in superfamilies. *Nucleic Acids Research* 37 (Suppl. 1), D310–D314.
- Daub, J., Eberhardt, R.Y., Tate, J.G., Burge, S.W., 2015. Rfam: Annotating families of non-coding RNA sequences. *RNA Bioinformatics*. 349–363.
- De Beer, T.A., Berka, K., Thornton, J.M., Laskowski, R.A., 2013. PDBsum additions. *Nucleic Acids Research* 42 (D1), D292–D296.
- De Castro, E., Sigrist, C.J., Gattiker, A., *et al.*, 2006. ScanProsite: Detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. *Nucleic Acids Research* 34 (Suppl. 2), W362–W365.
- Eddy, S.R., 1998. Profile hidden Markov models. *Bioinformatics (Oxford)* 14 (9), 755–763.
- Finn, R.D., Attwood, T.K., Bablitt, P.C., *et al.*, 2016a. InterPro in 2017 – Beyond protein family and domain annotations. *Nucleic Acids Research* 45 (D1), D190–D199.
- Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Research*. gkr367.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016b. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research* 44 (D1), D279–D285.
- Gilson, M.K., Liu, T., Baitaluk, M., *et al.*, 2016. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research* 44 (D1), D1045–D1053.
- Giudice, G., Sánchez-Cabo, F., Torroja, C., Lara-Pezzi, E., 2016. ATIRACT – A database of RNA-binding proteins and associated motifs. *Database: The Journal of Biological Databases and Curation* 2016, baw035.
- Gnad, F., Ren, S., Choudhary, C., Cox, J., Mann, M., 2010. Predicting post-translational lysine acetylation using support vector machines. *Bioinformatics* 26 (13), 1666–1668.
- Gough, J., Chothia, C., 2002. SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. *Nucleic Acids Research* 30 (1), 268–272.
- Haft, D.H., Selengut, J.D., Richter, R.A., *et al.*, 2012. TIGRFAMs and genome properties in 2013. *Nucleic Acids Research* 41 (D1), D387–D395.
- Haft, D.H., Selengut, J.D., White, O., 2003. The TIGRFAMs database of protein families. *Nucleic Acids Research* 31 (1), 371–373.
- Hornbeck, P.V., Zhang, B., Murray, B., *et al.*, 2014. PhosphoSitePlus, 2014: Mutations, PTMs and recalibrations. *Nucleic Acids Research* 43 (D1), D512–D520.
- Jones, P., Binns, D., Chang, H.-Y., *et al.*, 2014. InterProScan 5: Genome-scale protein function classification. *Bioinformatics* 30 (9), 1236–1240.
- Kim, Y., Ponomarenko, J., Zhu, Z., *et al.*, 2012. Immune epitope database analysis resource. *Nucleic Acids Research* 40 (W1), W525–W530.
- Klimke, W., Agarwala, R., Badretdin, A., *et al.*, 2008. The national center for biotechnology information's protein clusters database. *Nucleic Acids Research* 37 (Suppl. 1), D216–D223.
- Kozma, D., Simon, I., Tusnády, G.E., 2012. PDBTM: Protein Data Bank of Transmembrane Proteins after 8 years. *Nucleic Acids Research* 41 (D1), D524–D529.
- Law, V., Knox, C., Djoumbou, Y., *et al.*, 2013. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Research* 42 (D1), D1091–D1097.
- Leinonen, R., Akhtar, R., Birney, E., *et al.*, 2010. The European nucleotide archive. *Nucleic Acids Research* 39 (Suppl. 1), D28–D31.
- Letunic, I., Doerks, T., Bork, P., 2014. SMART: Recent updates, new developments and status in 2015. *Nucleic Acids Research* 43 (D1), D257–D260.
- Liu, Z., Li, Y., Han, L., *et al.*, 2014a. PDB-wide collection of binding data: Current status of the PDBbind database. *Bioinformatics* 31 (3), 405–412.
- Liu, Z., Wang, Y., Gao, T., *et al.*, 2014b. CPLM: A database of protein lysine modifications. *Nucleic Acids Research* 42 (D1), D531–D536.
- Lu, C.-T., Huang, K.-Y., Su, M.-G., *et al.*, 2012. DbPTM 3.0: An informative resource for investigating substrate site specificity and functional association of protein post-translational modifications. *Nucleic Acids Research* 41 (D1), D295–D305.
- Madden, T., 2013. The BLAST Sequence Analysis Tool.
- Marchler-Bauer, A., Anderson, J.B., DeWeese-Scott, C., *et al.*, 2003. CDD: A curated Entrez database of conserved domain alignments. *Nucleic Acids Research* 31 (1), 383–387.
- Marchler-Bauer, A., Derbyshire, M.K., Gonzales, N.R., *et al.*, 2014. CDD: NCBI's conserved domain database. *Nucleic Acids Research* 43 (D1), D222–D226.
- Mashima, J., Kodama, Y., Kosuge, T., *et al.*, 2016. DNA data bank of Japan (DDBJ) progress report. *Nucleic Acids Research* 44 (D1), D51–D57.
- Matlock, M.K., Holehouse, A.S., Naegle, K.M., 2014. ProteomeScout: A repository and analysis resource for post-translational modifications and proteins. *Nucleic Acids Research* 43 (D1), D521–D530.
- Mattick, J.S., Makunin, I.V., 2006. Non-coding RNA. *Human Molecular Genetics* 15 (Suppl. 1), R17–R29.
- Mattick, J.S., Rinn, J.L., 2015. Discovery and annotation of long noncoding RNAs. *Nature Structural & Molecular Biology* 22 (1), 5–7.
- Mi, H., Muruganujan, A., Thomas, P.D., 2013. PANTHER in 2013: Modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Res* 41 (Database issue), D377–D386.
- Minguez, P., Letunic, I., Parca, L., *et al.*, 2014. PTMcode v2: A resource for functional associations of post-translational modifications within and between proteins. *Nucleic Acids Research* 43 (D1), D494–D502.
- Mistry, J., Finn, R.D., Eddy, S.R., Bateman, A., Punta, M., 2013. Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. *Nucleic Acids Research* 41 (12), e121.
- Mosca, R., Céol, A., Stein, A., Olivella, R., Aloy, P., 2013. 3did: A catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Research* 42 (D1), D374–D379.
- Pearl, F.M.G., Bennett, C., Bray, J.E., *et al.*, 2003. The CATH database: An extended protein family resource for structural and functional genomics. *Nucleic Acids Research* 31 (1), 452–455.
- Peters, B., Sidney, J., Bourne, P., *et al.*, 2005. The immune epitope database and analysis resource: From vision to blueprint. *PLOS Biology* 3 (3), e91.
- Servant, F., Bru, C., Carrère, S., *et al.*, 2002. ProDom: Automated clustering of homologous domains. *Briefings in Bioinformatics* 3 (3), 246–251.
- Seto, A.G., Kingston, R.E., Lau, N.C., 2007. The coming of age for PwI proteins. *Molecular Cell* 26 (5), 603–609.
- Sigrist, C.J., Cerutti, L., Hulo, N., *et al.*, 2002. PROSITE: A documented database using patterns and profiles as motif descriptors. *Briefings in Bioinformatics* 3 (3), 265–274.

- Sigrist, C.J., de Castro, E., Cerutti, L., *et al.*, 2013. New and continuing developments at PROSITE. *Nucleic Acids Research* 41 (Database issue), D344–D347.
- Sigrist, C.J., De Castro, E., Langendijk-Genevaux, P.S., *et al.*, 2005. ProRule: A new database containing functional and structural information on PROSITE profiles. *Bioinformatics* 21 (21), 4060–4066.
- Sillitoe, I., Lewis, T., Orengo, C., *et al.*, 2015. Using CATH-Gene3D to Analyze the Sequence, Structure, and Function of Proteins. *Current protocols in bioinformatics*. 1.28. 1–1.28. 21.
- Sonnhammer, E.L., Eddy, S.R., Durbin, R., 1997. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function and Genetics* 28 (3), 405–420.
- Tatusov, R.L., Fedorova, N.D., Jackson, J.D., *et al.*, 2003. The COG database: An updated version includes eukaryotes. *BMC Bioinformatics* 4 (1), 41.
- Toribio, A.L., Alako, B., Amid, C., *et al.*, 2017. European Nucleotide Archive in 2016. *Nucleic Acids Research* 45 (D1), D32–D36.
- Wilson, D., Pethica, R., Zhou, Y., *et al.*, 2009. SUPERFAMILY – Sophisticated comparative genomics, data mining, visualization and phylogeny. *Nucleic Acids Research* 37 (Suppl. 1), D380–D386.

Relevant Websites

- <http://tools.iedb.org/main>
IEDB Analysis Resource.
- <http://www.insdc.org>
INSDC: International Nucleotide Sequence Database Collaboration.
- <https://www.ncbi.nlm.nih.gov/books/NBK279688/>
NCBI BLAST User Manual.
- <https://blast.ncbi.nlm.nih.gov/Blast.cgi>
NCBI BLAST Web Server.

Extraction of Immune Epitope Information

Guang Lan Zhang, Boston University, Boston, MA, United States and Harvard Medical School, Boston, MA, United States

Derin B Keskin, Harvard Medical School, Boston, MA, United States and Broad Institute, Boston, MA, United States

Lou Chitkushev, Boston University, Boston, MA, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

The adaptive immune system is composed of highly specialized cells and processes that recognize “non-self” antigens, mount immune responses to eliminate the antigens or antigen-infected cells, and develop immunological memory through memory B cells and memory T cells that leads to an enhanced response to subsequent encounters with the same antigens. Two arms of adaptive immunity, the humoral immunity mediated by B cells and the cell-mediated immunity mediated by T cells, work closely to combat infection and malignancy in an antigen specific manner. Major histocompatibility complex (MHC) proteins play a vital role in the regulation of cell-mediated immune responses. MHC class I molecules are found on every nucleated cell of the body, while MHC class II molecules are expressed on specialized cells, including professional APCs such as B cells, macrophages and dendritic cells (Janeway, 2005). They bind short peptides derived from protein antigens through proteolytic mechanisms and subsequently present MHC-peptide complexes on the cell surface for recognition by T cells. T cells identify foreign antigens through their T-cell receptors (TCRs), which interacts with MHC-peptide complexes in conjunction with CD4 or CD8 co-receptors (Meuer *et al.*, 1982; Wang and Reinherz, 2002). MHC class I antigen processing pathway steps include proteasome cleavage of proteins into shorter peptides, translocation of peptides into the endoplasmic reticulum (ER) by TAP (the transporter associated with antigen processing), optional ER trimming by aminopeptidases, insertion of peptides, typically 8–11 amino acid residues in length, into the binding groove of MHC molecules, and transport of peptide-MHC complexes to the cell surface for presentation to CD8 + T cells (Purcell and Gorman, 2004; Strehl *et al.*, 2005). In humoral immunity, B cells produce antibodies to eliminate extracellular microorganisms and prevent the spread of infection. T lymphocytes expressing the CD4 co-receptor recognize MHC class II molecules complexed to peptides generated from endogenous cellular proteins that are typically 8–30 amino acids in length and are often nested sets of peptides with a core sequence shared among them (Rudensky *et al.*, 1991; Engelhard, 1994; Bozzacco *et al.*, 2011). The recognition of peptide-MHC class II complexes by CD4 + T cells, stimulating them to make proteins that stimulate the B cells to differentiate into antibody secreting cells. Thus the activation of B cells is triggered by antigens and usually requires helper T cells (Janeway, 2005). B cells identify antigens through B-cell receptors, which recognize discrete sites on the surface of target antigens called B-cell epitopes (Van Regenmortel, 2009). Peptides that are presented by MHC and recognized by T-cells are termed T cell epitopes. They are important targets of adaptive immune responses and rational vaccine design. This article focuses on the MHC binding peptides and T cell epitopes.

Identification of T cell epitopes is essential in the development of epitope-based vaccines. One of the key steps in T-cell epitope prediction is the prediction of MHC binding, as it is considered the most selective step in T cell recognition. The challenge in the identification of broadly protective vaccine targets stems from two principal sources – the diversity of pathogens and the diversity of human immune system. HLA (Human Leukocyte Antigen, human MHC) are among the most polymorphic human proteins, with more than 12,000 known HLA Class I alleles and 4700 HLA class II variants as of Dec 2017 (Robinson *et al.*, 2015). To handle HLA polymorphism, the concept of HLA supertypes was proposed to group HLA alleles showing similar peptide binding specificity into supertypes (Sidney *et al.*, 1996; Sette and Sidney, 1999). The ideal vaccine targets are conserved immune epitopes that are broadly cross-reactive to viral subtypes and protective of a large human population (Zhang *et al.*, 2014). In this article we reviewed *in silico* and *in vitro* methods for immune epitope discovery.

In Vitro Methods for Identification of MHC Ligands and T Cell Epitopes

In Vitro Discovery of MHC Ligands

MHC binding to a potential peptide can be tested *in vitro* using binding assays or Mass spectrometry (MS) based methods. Note that MHC binding does not guarantee T cell reactivity. In binding assays, purified MHC proteins or TAP deficient cells lines are used to test peptide binding to specific MHC molecules. These assays are only useful in testing MHC/peptide binding, a prerequisite for T-cell activation. It does not guarantee T cells will recognize the binding peptide as antigen. For example, TAP-deficient T2 cell HLA stabilization assay was used to confirm 10 HPV (human papillomavirus) E6 and E7 peptides that bind to HLA-A*0201 (Riemer *et al.*, 2010).

MHC-peptide complexes may be immunoprecipitated from target cells. Peptides may be released from MHC using fractionation (e.g., using HPLC, the physical separation capabilities of liquid chromatography). These peptides can be analyzed by MS analysis that produces detailed information of the MHC ligandome. MS based ligand discovery physically detect the peptides that are naturally processed and presented on the cell surface. We refer to these peptides as “eluted peptides”. The immunogenicity of these peptides needs further *in vivo* and *in vitro* testing. MS based assays require extensive optimization for pure MHC-Peptide complex isolation. MS analysis also requires rare expertise, which are not directly applicable to all laboratories (Bozzacco *et al.*, 2011; Abelin *et al.*, 2017).

In Vitro Discovery of T Cell Epitopes

T cell epitope discovery without the help of bioinformatics prediction algorithms applies T cell immunogenicity assays on overlapping peptides. These synthesized peptides are typically 15–16 amino acids (aa) in length with 10aa overlap and cover the protein sequence of the whole pathogen. They are pooled together to be tested. T cells can be tested *ex-vivo* from previously infected, immune donors or after a few rounds of stimulation in cell culture from naïve donors. Generation of an antigen specific T cell line may take up to four weekly antigen stimulations. Cloning of a T cell may require further culture for up to two months of hands-on work.

In T cell reactivity testing, multiple experimental assays including intra-cellular cytokine staining, proliferation assays and Elispot assays are available to test peptide epitope reactivity. In intra-cellular cytokine staining, T cells produce cytokines such as IFN γ , TNF α and IL-2 after epitope recognition and activation. After a brief *ex-vivo* culture with possible reactive peptides or peptide pools, T cells are stained for these cytokines *in vitro* using specific antibodies linked to fluorochromes. Cell membranes need to be permeabilized for antibody access using detergents, which makes the assay less sensitive than other available assays. Golgi blocking agents can be used briefly in culture for cytokine deposition in the cytoplasm. After antibody staining the cells are analyzed using a flow-cytometer (Ott *et al.*, 2017).

In proliferation assays, T cells start proliferating 2–3 days after antigen recognition. T cell proliferation can be measured using tritiated Thymidine incorporation or dilution of cell permeable marker dyes such as CFSE (Carboxyfluorescein succinimidyl ester) by flow cytometry. These assays are sensitive however require long term cell culture (Mellor *et al.*, 2002).

Elispot assays utilize cytokine capture antibody coated plates. T cells produce cytokines such as IFN γ , TNF α and IL-2 after epitope recognition and activation. T cells can be stimulated in this cytokine capture antibody coated plates up to 48 h. After removal of T cells, the captured cytokines on plates are detected by a sandwich assay utilizing a biotinylated secondary cytokine specific antibody and streptavidin enzyme conjugate. These assays produce spots, which represents each cytokine producing T cells. Spot forming cells can be calculated from each assay and will give accurate antigen specific T cell frequency in a sample (Keskin *et al.*, 2015).

In Silico Methods for Epitope Identification

The advancement in high throughput methods such as whole-genome sequencing and in bioinformatics led to the identification of potential vaccine targets without the need for growing the pathogen. The concept of reverse vaccinology supports identification of vaccine targets by large-scale bioinformatics screening of entire pathogenic genomes followed by experimental validation (Rappuoli, 2000; Mora *et al.*, 2003; Rappuoli *et al.*, 2016).

Publicly available bioinformatics resources include biological databases that organize and catalog immune epitopes and their related information as well as computational systems that model the biological processes involved in antigen processing and presentation, i.e., proteasome cleavage, TAP transportation, and MHC binding.

Databases Hosting Immune Epitopes

Table 1 summarizes immune epitopes databases that are publically available. Some are general-purpose databases, including SYFPEITHI, the Immune Epitope Database and Analysis Resource (IEDB), MCHBN, and Dana-Farber Repository for Machine Learning in immunology (DFRMLI), hosting MHC binding peptides and immune epitopes from various types of antigens. Some host immune epitopes from tumor antigens, such as the Cancer Immunome database and Tantigen. Some host immune epitopes from specific viruses, such as EBVdb (Epstein-Barr virus T cell antigen database), FLAVIdB (Flaviviruses database), FluKB (Influenza virus knowledge-base), HPVdb (Human Papillomavirus T cell Antigen Database), Los Alamos hepatitis C immunology database, and Los Alamos HIV Molecular Immunology Database.

DFRMLI hosts standardized data sets of HLA-binding peptides with all binding affinities mapped onto a common scale and a list of experimentally validated naturally processed eluted peptides and non-binding peptides derived from tumor or virus antigens (Zhang *et al.*, 2011). Some of the data in DFRMLI were aggregated from publically available databases such as MHCPEP (Brusic *et al.*, 1994), IEDB, CBS (Center for Biological Sequence Analysis). Some data are our in-house data and the validation data sets were made of carefully selected independent data. These data sets were used to support the 1st and 2nd machine learning competitions in immunology in 2011 and 2012 (Zhang *et al.*, 2011). IEDB established in 2004 catalogs experimentally characterized B and T cell epitopes, MHC binding peptides, and data on corresponding experimental assays. It also hosts computational tools for immune epitope prediction and analysis (Vita *et al.*, 2015; Fleri *et al.*, 2017). Domain experts manually curated the data in IEDB to ensure quality and data completeness. The IEDB intentionally does not contain HIV (human immunodeficiency virus) epitopes as they are archived in the Los Alamos HIV Molecular Immunology Database (Altman and Safritz, 1998). MCHBN is a database hosting experimentally validated T cell epitopes, binding peptides to MHC and TAP, and non-binding peptides to MHC and TAP. MCHBN also provides bioinformatics tools for the analysis and retrieval of information like mapping of antigenic regions, creation of allele specific dataset, BLAST search, various diseases associated with MHC alleles etc (Lata *et al.*, 2009). SYFPEITHI is among the earliest online database that host immune epitopes (Rammensee *et al.*, 1999; Schuler *et al.*, 2007). It

Table 1 Online databases hosting immune epitopes listed in alphabetical order. The information is up to date as of Dec 2017

Database	Description of content	URL
Cancer Immunome Database	Four data tables containing over 400 T cell epitopes identified in tumor antigens	www.cancerresearch.org/scientists/events-and-resources/peptide-database
Dana-Farber Repository for Machine Learning in Immunology (DFRMLI)	MHC class I and II binding peptides, eluted peptides, non-binding peptides	projects.met-hilab.org/DFRMLI/
EBVdb, Epstein-Barr virus (EBV) T cell antigen database	610 T cell epitopes and 26 HLA ligands from EBV antigens	projects.met-hilab.org/ebv
FLAVIdb, Flaviviruses database	184 verified T-cell epitopes, 201 verified B-cell epitopes	cvc.dfci.harvard.edu/flavi/
FluKB, a knowledge-base for influenza viruses	357 T-cell epitopes, 685 HLA binding peptides, 16 naturally processed MHC ligands, and 28 influenza antibodies and their structurally defined B-cell epitopes	research4.dfci.harvard.edu/cvc/flukb/
HPVdb, Human Papillomavirus (HPV) T cell Antigen Database	191 T cell epitopes and 45 HLA ligands from HPV antigens	projects.met-hilab.org/hpv
IEDB	422,219 peptide epitopes, 2553 non-peptidic epitopes, T cell and B cell assays, MHC binding assays etc.	www.iedb.org/
Los Alamos hepatitis C immunology database	383 CD8 + T cell epitopes, 222 CD4 + T cell epitopes, and 60 antibody binding sites from HCV proteins for human, mouse, chimpanzee, and rat	hcv.lanl.gov/content/immuno/immuno-main.html
Los Alamos HIV Molecular Immunology Database	9821 CD8 + T cell epitopes, 1578 CD4 + T cell epitopes, and 3306 antibody binding sites from HIV proteins for human, macaque, chimpanzee, mouse etc.	www.hiv.lanl.gov/content/immunology/
MHCBN	6722 T cell epitopes, 4022 non-binding peptides, 1022 TAP binding peptides	crdd.osdd.net/raghava/mhcbn/
SYFPEITHI	>7000 MHC class I and II binding peptides	www.syfpeithi.de/
Tantigen	>1000 HLA class I and II binding peptides, eluted peptides, and T cell epitopes from tumor antigens	projects.met-hilab.org/tadb or cvc.dfci.harvard.edu/tantigen/

allows users to search for MHC ligands, peptide binding motifs, natural ligands, and T cell epitopes by specifying MHC alleles, source proteins/organisms, etc., It also supports MHC binding prediction.

The identification of tumor antigens is a high priority in cancer research and is an essential component in developing immune-based strategies to combat cancer. The following databases introduced are useful resources for the study of immune responses against tumors. The Cancer Immunome database is a continuation of the SEREX database, maintained by the Ludwig Institute for Cancer Research since 1997, in a more organized form. The database contains T cell epitopes in tumor antigens that are classified based on their tumor specificity into four data tables (Jongeneel, 2001; Vigneron *et al.*, 2013). All immune epitopes in the database have been documented to elicit immune responses in cancer patients. Tantigen is a database of human tumor T cell antigens that catalogued more than 1000 tumor peptides. The peptides are labelled as one of four categories: (1) peptides measured *in vitro* to bind the HLA, (2) peptides found to bind the HLA and to elicit an *in vitro* T cell response, (3) peptides shown to elicit *in vivo* tumor rejection, and (4) peptides processed and naturally presented as defined by physical detection. The Tumor T Cell Antigen Database (TANTIGEN) is a data source and analysis platform for cancer vaccine target discovery focusing on human tumor-derived HLA ligands and T cell epitopes. It contains 4006 curated antigen entries representing 251 unique proteins, information on experimentally validated T cell epitopes and HLA ligands, antigen isoforms, antigen sequence mutations, and tumor antigen classification. TANTIGEN provides a rich data resource and tailored analysis tools for tumor associated epitope and neoepitope discovery studies (Olsen *et al.*, 2017).

Some viruses such as Epstein-Barr virus (EBV) and human papillomaviruses (HPVs) are the causes of many cancers, which some such as influenza viruses, flaviviruses, and HIV cause various infectious disease. To facilitate diagnosis, prognosis and characterization of these diseases, it is necessary to make full use of the immunological data on the viruses. Epstein-Barr virus (EBV) T cell antigen database is a data source and analysis platform for EBV immune target discovery by focusing on EBV antigens that contain T cell epitopes (Zhang *et al.*, 2015). It hosts 2622 curated EBV antigen sequences, 610 T cell epitopes and 26 HLA ligands. FLAVIdb combines antigenic data of flaviviruses, specialized analysis tools, and workflows for automated complex analyses focusing on applications in immunology and vaccinology (Olsen *et al.*, 2011). It contains 12,858 entries of flavivirus antigen sequences, 184 T-cell epitopes, 201 B-cell epitopes, and four representative molecular structures of the dengue virus envelope protein. FluKB provides access to more than 400,000 curated influenza protein sequences, known epitope data (357 verified T-cell epitopes, 685 HLA binders, and 16 naturally processed MHC ligands), and a collection of 28 influenza antibodies and their structurally defined B-cell epitopes (Simon *et al.*, 2015). HPVdb is a data mining system for knowledge discovery in HPV with applications in T cell immunology and vaccinology. It contains 2781 curated antigenic proteins derived from 18 genotypes of high-risk HPV and 18 genotypes of low-risk HPV, 191 T cell epitopes and 45 HLA ligands (Zhang *et al.*, 2014). Los Alamos hepatitis C immunology database contains T-cell epitopes and antibody binding sites in hepatitis C virus (HCV) for human and multiple animal models, as well as relevant retrieval

and analysis tools (Yusim *et al.*, 2005). The HIV Molecular Immunology Database provides an annotated collection of HIV-1 cytotoxic and helper T-cell epitopes and antibody binding sites (Immunology, 2016).

Prediction Systems for Proteasome Cleavage, TAP Binding, and MHC Binding

Computational tools that model the proteolytic activities of the proteasome and TAP binding can be integrated with prediction systems for MHC class I binding peptides to improve the performance of *in silico* prediction of T-cell epitopes. NetChop predicts cleavage sites of the human proteasome using artificial neural network (ANN) models trained on data from *in vitro* digestion experiments with constitutive proteasomes and MHC Class I ligand data that reflect immunoproteasome specificity (Kesmir *et al.*, 2002; Nielsen *et al.*, 2005). NetChop team maintains the tool and updated it periodically. The latest version is NetChop v3.1. MAPPP models two essential steps in endogenous antigen processing pathway, the proteasome cleavage and MHC binding. FRAGPREDICT that predicts the proteasome cleavage sites using motifs encoding cleavage-enhancing and -inhibiting amino acids is the part that models proteasome cleavage in MAPPP (Holzhutter *et al.*, 1999). PAPProC (Prediction Algorithm for Proteasomal Cleavages) is an online prediction system for cleavages by constitutive proteasomes of human and yeast based on experimental cleavage data (Nussbaum *et al.*, 2001). Pcleavage is a prediction system for constitutive as well as immunoproteasome cleavage sites using a support vector machine (SVM) based method (Bhasin and Raghava, 2005). Saxova and co-authors evaluated the performance of PAPProC, FRAGPREDICT and two versions of NetChop, v1.0 and v2.0 and found that NetChop v2.0 performed the best (Saxova *et al.*, 2003) (Tables 2).

TAP is a transmembrane protein that transports antigenic peptides into the ER. It has been shown that the efficiency of TAP-mediated translocation of a peptide correlates positively to its TAP binding affinity (Brusic *et al.*, 1999). Peptides of 8–16 amino acids and have sufficient TAP binding affinity are efficiently translocated by TAP into the ER, while longer peptides may be transported but with lower efficiency (Saveanu *et al.*, 2005). SVMTAP employs a matrix-based method and support vector regression to predict peptides transported by TAP (Donnes and Kohlbacher, 2005). TAPPred is based on cascade SVM using sequence and properties of the amino acids (Bhasin and Raghava, 2004) (Table 3).

MHC binding is the most selective step in the antigen presentation pathway. Quite a few online prediction systems have been developed for the prediction of peptide binding to MHC molecules based on algorithms including binding motifs, quantitative matrices, and machine learning methods such as ANN models and support vector machines. In addition there has been effort to develop a prediction algorithm based on naturally processed and presented MHC eluted peptide epitopes detected by MS together with gene expression and protein processing (Abelin *et al.*, 2017). While the prediction systems listed in Table 4 were designed for different purposes with various methods, all of them have peptide binding predictions implemented as specific modules. MAPPP, ProPred-I, and RANKPEP predict proteasomal cleavage sites and MHC binding. MULTIPRED2 predicts peptide binding to HLA supertypes, and BIMAS predicts peptide binding as half-time dissociation (off-rate).

BIMAS is one of the first online tools that predicts 8-mer, 9-mer, or 10-mer binding peptides using class I mouse and human MHC binding motifs (Parker *et al.*, 1994). SYFPEITHI predicts MHC class I and class II binding peptides in human, mouse and rat based on allele-specific peptide motifs (Schuler *et al.*, 2007). The MHC binding prediction part of MAPPP identifies potential 8-mer, 9-mer and 10mer peptides that bind human, mouse and bovine MHC class I alleles with the option to use either BIMAS matrix or SYFPEITHI matrix for prediction (Hakenberg *et al.*, 2003). Note that MAPPP (BIMAS) and MAPPP (SYFPEITHI) showed identical predictions to BIMAS and SYFPEITHI, respectively (Lin *et al.*, 2008). RANKPEP predicts binding peptides of 88 MHC class I and 50 MHC class II alleles in human and mouse using position specific scoring matrices (PSSMs) (Reche *et al.*, 2002, 2004). There is option to add in the proteasome cleavage step in the prediction.

Table 2 Online prediction tools for proteasome cleavage listed in alphabetical order. The information is up to date as of Dec 2017

Online tool	URL
FRAGPREDICT	www.mpiib-berlin.mpg.de/MAPPP/cleavage.html
NetChop	www.cbs.dtu.dk/services/NetChop/
PAPProC	www.paproc.de/
Pcleavage	crdd.osdd.net/raghava/pcleavage/

Table 3 Online prediction tools for TAP binding listed in alphabetical order. The information is up to date as of Dec 2017

Online tool	URL
SVMTAP	abi.inf.uni-tuebingen.de/Services/SVMTAP
TAPPred	crdd.osdd.net/raghava/tappred/

Table 4 Online tools for predicting MHC binding peptides listed in alphabetical order. The information is up to date as of Dec 2017

Online tool	Description	URL
BIMAS	Predicts MHC class I binding peptides in human and mouse	www-bimas.cit.nih.gov/molbio/hla_bind/
IEDB MCH class I tool	Predicts MHC class I binding peptides in human and multiple animal models	tools.immuneepitope.org/mhci/
IEDB MCH class II tool	Predicts MHC class II binding peptides in human and mouse	tools.iedb.org/mhcii/
MAPPP	Predicts MHC class I binding peptides in human and mouse	www.mpiib-berlin.mpg.de/MAPPP/binding.html
Multipred2	Predicts binding peptides of multiple alleles in HLA class I and class II supertypes and of alleles belonging to an individual's genotype	projects.met-hilab.org/multipred2/
NetMHC	Predicts MHC class I binding peptides in human, monkey, cattle, pig, and mouse	www.cbs.dtu.dk/services/NetMHC/
NetMHCcons	Predicts MHC class I binding peptides in human, chimpanzee, monkey, gorilla, pig, and mouse	www.cbs.dtu.dk/services/NetMHCcons/
NetMHCpan	Predicts binding peptides of any MHC class I molecule of known sequence	www.cbs.dtu.dk/services/NetMHCpan/
NetMHCII	Predicts MHC class II binding peptides in human and mouse	www.cbs.dtu.dk/services/NetMHCII/
NetMHCIIpan	Predicts binding peptides of HLA-DR, -DP, -DQ molecules and H2 class II molecules	www.cbs.dtu.dk/services/NetMHCIIpan/
PickPocket	Predicts MHC class II binding peptides in human and mouse	www.cbs.dtu.dk/services/PickPocket/
ProPred	Predicts binding peptides of HLA class II alleles	crdd.osdd.net/raghava/propred/
ProPred-I	Predicts peptides that bind human, mouse and bovine MHC class I alleles	crdd.osdd.net/raghava/propred1/
RANKPEP	Predicts MHC class I and class II binding peptides in human and mouse	imed.med.ucm.es/Tools/rankpep.html
SVMHC	Predicts binding peptides of MHC class I and class II alleles in human and mouse	abi.inf.uni-tuebingen.de/Services/SVMHC
SYFPEITHI	Predicts MHC class I and Class II binding peptides in human, mouse, and rat	www.syfpeithi.de/bin/MHCServer.dll/EpitopePrediction.htm
TEPITOPEpan	Predicts binding peptides to HLA-DR alleles	datamining-iip.fudan.edu.cn/service/TEPITOPEpan/TEPITOPEpan.html

NetMHC v4.0 predicts peptides that bind 81 different HLA-A, -B, -C and -E alleles as well as 41 MHC class I alleles of monkey, cattle, pig, and mouse. It employs a sequence alignment method based on ANN that allows insertions and deletions in the alignment (Nielsen *et al.*, 2003; Andreatta and Nielsen, 2016). NetMHCpan v4.0 predicts peptides that bind any MHC class I molecule of known sequence based on an ANN model trained using pseudo-sequences representing peptide/MHC interactions (Nielsen *et al.*, 2007a,b; Jurtz *et al.*, 2017). The pseudo-sequences that capture the specific HLA-peptide interaction by combining each amino acid of the peptide with the variable amino acids of its positional environment was first proposed in Brusci *et al.* (2002). NetMHCpan v4.0 was trained on a combination of quantitative MHC binding data and MS derived MHC eluted ligands (Hoof *et al.*, 2009; Jurtz *et al.*, 2017). PickPocket v1.1 predicts binding peptides of any known MHC class I molecule using position specific weight matrices that capture receptor-pocket similarities between MHC molecules. Predictions can be made for HLA-A, -B, -C, -E and -G alleles as well as MHC class I alleles in mouse, cattle and pig (Zhang *et al.*, 2009). NetMHCcons a consensus method for MHC class I binding predictions integrating three methods, NetMHC, NetMHCpan and PickPocket (Karosiene *et al.*, 2012). NetMHCII v2.2 predicts binding peptides for 14 HLA-DR alleles, six HLA-DQ, six HLA-DP, and two mouse H2 class II alleles. It employs an ANN-based alignment method, NN-align, trained using an algorithm that incorporates information on the residues flanking the peptide-binding core (Nielsen *et al.*, 2007a,b; Nielsen and Lund, 2009). NetMHCIIpan v3.1 predicts binding peptides of MHC class II molecules in human (HLA-DR, HLA-DP and HLA-DQ) and mouse (H2) (Karosiene *et al.*, 2013; Andreatta *et al.*, 2015).

IEDB MCH class I tool predicts binding peptides of MHC class I alleles in human and multiple animals. It allows users to choose from a number of prediction methods including Stabilized matrix method (SMM) (Peters and Sette, 2005), NetMCHpan, NetMHC, NetMHCcons, PickPocket and so on. IEDB MCH class II tool predicts binding peptides of MHC class I alleles in human and multiple animals. It allows users to choose from a number of prediction methods including the consensus method (Wang *et al.*, 2008), NetMHCII, NetMHCIIpan and so on. MULTIPRED2 screens for peptide binding to multiple alleles belonging to HLA class I and class II supertypes as well as to alleles belonging to an individual's genotype. The prediction engines used are NetMHCpan and NetMHCIIpan. In total MULTIPRED2 performs binding predictions on 1077 alleles belonging to 26 HLA supertypes (Zhang *et al.*, 2011).

ProPred-I predicts binding peptides of 47 MHC class-I alleles in human, mouse and cow using quantitative matrices. ProPred-I provides the option to filter out the peptides with predicted proteasomal cleavage sites (Singh and Raghava, 2003). The ProPred1 predictions were found highly correlated with BIMAS predictions for five HLA-I alleles (HLA-A*02:01, -A*03:01, -A*11:01, -A*24:01, and -B*08:01,) as some of the BIMAS matrices were adopted by ProPred1 (Lin *et al.*, 2008). ProPred predicts binding peptides of 51 HLA-DRB1 and -DRB5 alleles using quantitative matrices (Singh and Raghava, 2001). SVMHC predicts of 9-mer and 10-mer peptides that bind multiple MHC class I alleles in human and mouse using SVM based methods. It also predicts binding peptides to HLA-DRB1 alleles using PSSMs from TEPITOPE, a widely used method containing PSSMs that cover 51 different HLA-DR alleles (Sturniolo *et al.*, 1999; Donnes and Kohlbacher, 2006). TEPITOPEpan was built by extrapolating from the

HLA-DR alleles with known binding specificities in TEPITOPE to the HLA-DR molecules with unknown binding specificities based on pocket similarity (Zhang *et al.*, 2012). TEPITOPEpan is able to predict binding peptides of more than 700 HLA-DR alleles.

With many MHC binding prediction systems available, the next question is how to choose the most suitable one for predicting binding to a given MHC molecule. Several groups have reported their effort in assessing performance of computational methods that predict peptide binding to several common MHC class I (Peters *et al.*, 2006; Lin *et al.*, 2008; Gowthaman *et al.*, 2010; Zhang *et al.*, 2011; Trolle *et al.*, 2015) and II alleles (Lin *et al.*, 2008). Machine learning competition in immunology organized by Dana-Farber Cancer Institute evaluated the performance of 20 computational methods in predicting 9-mer and 10-mer peptides binding to HLA-A*01:01, A*02:01, and B*07:02. The evaluation data sets used in the competition are available at DFRMLI. It was found that NetMHC, NetMHCpan, and NetMHCcons demonstrated the best performance overall (Zhang *et al.*, 2011). Lin *et al.* (2008) compare the performance of 30 computational methods in predicting peptides binding to seven HLA class I alleles, A*0201, 0301, 1101, 2402, B*0702, 0801, and 1501. In general they found the prediction methods for A*0201, 0301, 1101, B*0702, 0801, and 1501 have excellent, and for A*2402 moderate classification accuracy (Lin *et al.*, 2008). Gowthaman *et al.* evaluated eight prediction systems in predicting peptides binding to seven HLA class I alleles, A*0101, 0201, 0301, 1101, 2402, B*0702, and 0801. It was found that NetMHC, NetMHCpan, and IEDB showed better overall efficiency (Gowthaman *et al.*, 2010). In Lin *et al.* (2008), 21 HLA class II prediction methods were evaluated for their performance in predicting binding peptides of seven common HLA-DR allele, DRB1*0101, 0301, 0401, 0701, 1101, 1301, and 1501. It was found that the HLA class II predictors do not match prediction capabilities of HLA class I ones. It is more difficult to predict MHC class II binding as, unlike MHC class I molecules, binding grooves of a MHC class II molecules have open ends allowing them to accommodate the 9-mer binding core of the peptides inside while peptide termini protrude outside of the grooves (Stern *et al.*, 1994). Not a single method is consistently the best for all the alleles and users have to select the suitable predictors by referring to the detailed findings in the papers. A framework that supports automated benchmarking of MHC binding prediction tools was developed. It runs weekly benchmarks on data that are newly entered into IEDB to provide up-to-date performance evaluations available at the website provided in "Relevant Website section" (Trolle *et al.*, 2015; Andreatta *et al.*, 2017).

Discussion

Immune epitopes are the targets of adaptive immune responses and rational vaccine design. We reviewed *in silico* and *in vitro* methods for immune epitope discovery. *In silico* methods include databases of immune epitopes, prediction systems for proteasome cleavage, TAP binding, and MHC binding. These *in silico* tools support downstream *in vitro* and *in vivo* studies. Publically available immune epitope databases allow both scientists and clinicians to utilize target genome information to support rational vaccine design and development. The availability of curated MHC binding data and T cell epitopes in turn improves target selection. These technologies are finding real life applications, for example the vaccines against HIV, the causative agent of AIDS (Stephenson *et al.*, 2016). The vast genetic diversity of HIV type 1 (HIV-1) envelope glycoprotein poses challenges in developing a global vaccine. Currently a mosaic HIV virus vaccine that contains mosaic HIV virus components designed utilizing Los Alamos HIV database to provide global virus coverage is in clinical trial (Ad26-env mosaic vaccine) (Nkolola *et al.*, 2014). Similar efforts are underway to develop universal Influenza vaccines utilizing FluKB and vaccines against cancer causing viruses such as HPV utilizing HPVdb (Riemer *et al.*, 2010; Keskin *et al.*, 2011; Keskin *et al.*, 2015). Computational tools and databases also play a role in support of developing personalized neo-antigen based cancer vaccines. Neo-antigens arise from tumor specific mutations, these epitopes are exclusively tumor specific and they circumvent T cell tolerance against self-epitopes (Fritsch *et al.*, 2013). Next generation sequencing and utilization of T cell epitope prediction algorithms provide tools to design cancer vaccines against an individual's own tumor from the Tumor gene sequencing data, one patient at a time (Melief, 2017; Ott *et al.*, 2017; Sahin *et al.*, 2017). These personalized cancer vaccines provide a leap in cancer treatment in the direction of personalized immunotherapy.

Future Directions

Identification of peptide presentation and processing rules through MS analysis of HLA binding peptides is currently perfecting epitope prediction algorithms (Abelin *et al.*, 2017). Next-generation sequencing (NGS) enables rapid and accurate deduction of the genomics of viruses, bacteria and cancers, in which we can identify mutated new epitopes (neoepitopes) (Ott *et al.*, 2017). Many existing prediction systems predict MHC binding peptides and MS analysis of these peptides detect the presentation of them on cell surface, however not all of these presented peptides are seen by T cells as antigens. Those silent, stealth epitopes complicate reverse vaccinology. There are efforts to improve the prediction of antigenicity of epitopes, however it requires large amount of validated antigen data. "T cell receptor (TCR) sequences are very diverse, with many more possible sequence combinations than T cells in any one individual" (Davis and Bjorkman, 1988; Shortman *et al.*, 1990; Qi *et al.*, 2014). Technologies to rapidly detect T cell epitopes are currently in development (Gee *et al.*, 2018). The next advancement in rapid vaccine design would be algorithms that can predict the immunogenicity of HLA binding peptides. Combinations of all three of these technologies, (i) rapid NGS sequencing to deduce protein coding sequences, (ii) HLA allele specific, predictive epitope processing and presentation algorithms and (iii) development of TCR specific antigenicity prediction would finally enable rapid vaccine design through reverse vaccinology.

Conflict of Interest

The authors declare that they have no conflict of interest.

See also: Biological Database Searching. Information Retrieval in Life Sciences. Natural Language Processing Approaches in Bioinformatics

References

- Abelin, J.G., Keskin, D.B., Sarkizova, S., *et al.*, 2017. Mass Spectrometry Profiling of HLA-Associated Peptidomes in Mono-allelic Cells Enables More Accurate Epitope Prediction. *Immunity* 46 (2), 315–326.
- Altman, J.D., Safrit, J.T., 1998. MHC tetramer analyses of CD8+ T-cell responses to HIV and SIV. *HIV Molecular Immunology*. Database 1998.
- Andreatta, M., Karosiene, E., Rasmussen, M., *et al.*, 2015. Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification. *Immunogenetics* 67, 641–650.
- Andreatta, M., Nielsen, M., 2016. Gapped sequence alignment using artificial neural networks: Application to the MHC class I system. *Bioinformatics* 32, 511–517.
- Andreatta, M., Trolle, T., Yan, Z., *et al.*, 2017. An automated benchmarking platform for MHC class II binding prediction methods. *Bioinformatics*.
- Bhasin, M., Raghava, G.P., 2004. Analysis and prediction of affinity of TAP binding peptides using cascade SVM. *Protein Science* 13, 596–607.
- Bhasin, M., Raghava, G.P., 2005. Pcleavage: An SVM based method for prediction of constitutive proteasome and immunoproteasome cleavage sites in antigenic sequences. *Nucleic Acids Research* 33 (Web Server issue), W202–W207.
- Bozzacco, L., Yu, H., Zebroski, H.A., *et al.*, 2011. Mass spectrometry analysis and quantitation of peptides presented on the MHC II molecules of mouse spleen dendritic cells. *Journal of Proteome Research* 10, 5016–5030.
- Brusic, V., Petrovsky, N., Zhang, G., Bajic, V.B., 2002. Prediction of promiscuous peptides that bind HLA class I molecules. *Immunology and Cell Biology* 80, 280–285.
- Brusic, V., Rudy, G., Harrison, L.C., 1994. MHCPEP: A database of MHC-binding peptides. *Nucleic Acids Research* 22, 3663–3665.
- Brusic, V., van Ender, P., Zelezniok, J., *et al.*, 1999. A neural network model approach to the study of human TAP transporter. *In Silico Biology* 1, 109–121.
- Davis, M.M., Bjorkman, P.J., 1988. T-cell antigen receptor genes and T-cell recognition. *Nature* 334 (6181), 395–402.
- Donnes, P., Kohlbacher, O., 2005. Integrated modeling of the major events in the MHC class I antigen processing pathway. *Protein Science* 14, 2132–2140.
- Donnes, P., Kohlbacher, O., 2006. SVMHC: A server for prediction of MHC-binding peptides. *Nucleic Acids Research* 34 (Web Server issue), W194–W197.
- Engelhard, V.H., 1994. Structure of peptides associated with class I and class II MHC molecules. *Annual Review of Immunology* 12, 181–207.
- Fleri, W., Paul, S., Dhanda, S.K., *et al.*, 2017. The immune epitope database and analysis resource in epitope discovery and synthetic vaccine design. *Frontiers in Immunology* 8, 278.
- Gee, M.H., Han, A., Lofgren, S.M., *et al.*, 2018. Antigen Identification for Orphan T Cell Receptors Expressed on Tumor-Infiltrating Lymphocytes. *Cell* 172 (3), 549–563. e516.
- Gowthaman, U., Chodisetti, S.B., Parihar, P., Agrewala, J.N., 2010. Evaluation of different generic in silico methods for predicting HLA class I binding peptide vaccine candidates using a reverse approach. *Amino Acids* 39, 1333–1342.
- Hacohen, N., Fritsch, E.F., Carter, T.A., Lander, E.S., Wu, C.J., 2013. Getting personal with neoantigen-based therapeutic cancer vaccines. *Cancer Immunology Research* 1, 11–15.
- Hakenberg, J., Nussbaum, A.K., Schild, H., *et al.*, 2003. MAPPP: MHC class I antigenic peptide processing prediction. *Applied Bioinformatics* 2, 155–158.
- Holzthutter, H.G., Frommel, C., Kloetzel, P.M., 1999. A theoretical approach towards the identification of cleavage-determining amino acid motifs of the 20 S proteasome. *Journal of Molecular Biology* 286, 1251–1265.
- Hoof, I., Peters, B., Sidney, J., *et al.*, 2009. NetMHCpan, a method for MHC class I binding prediction beyond humans. *Immunogenetics* 61, 1–13.
- Immunology, H.M., 2016. *HIV Molecular Immunology* 2016. Los Alamos, New Mexico: Los Alamos National Laboratory, Theoretical Biology and Biophysics.
- Janeway, C., 2005. *Immunobiology: The Immune System in Health and Disease*. New York, NY, USA: Garland Science.
- Jongeneel, V., 2001. Towards a cancer immunome database. *Cancer Immunology* 1, 3.
- Jurtz, V., Paul, S., Andreatta, M., *et al.*, 2017. NetMHCpan-4.0: Improved peptide-MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. *Journal of Immunology* 199, 3360–3368.
- Karosiene, E., Lundegaard, C., Lund, O., Nielsen, M., 2012. NetMHCcons: A consensus method for the major histocompatibility complex class I predictions. *Immunogenetics* 64, 177–186.
- Karosiene, E., Rasmussen, M., Blicher, T., *et al.*, 2013. NetMHCIIpan-3.0, a common pan-specific MHC class II prediction method including all three human MHC class II isotypes, HLA-DR, HLA-DP and HLA-DQ. *Immunogenetics* 65, 711–724.
- Keskin, D.B., Reinhold, B., Lee, S.Y., *et al.*, 2011. Direct identification of an HPV-16 tumor antigen from cervical cancer biopsy specimens. *Frontiers in Immunology* 2, 75.
- Keskin, D.B., Reinhold, B.B., Zhang, G.L., *et al.*, 2015. Physical detection of influenza A epitopes identifies a stealth subset on human lung epithelium evading natural CD8 immunity. *Proceedings of the National Academy of Sciences of the United States of America* 112, 2151–2156.
- Kesmir, C., Nussbaum, A.K., Schild, H., Detours, V., Brunak, S., 2002. Prediction of proteasome cleavage motifs by neural networks. *Protein Engineering* 15, 287–296.
- Lata, S., Bhasin, M., Raghava, G.P., 2009. MHCBN 4.0: A database of MHC/TAP binding peptides and T-cell epitopes. *BMC Research Notes* 2, 61.
- Lin, H.H., Zhang, G.L., Tongchusak, S., Reinherz, E.L., Brusic, V., 2008. Evaluation of MHC-II peptide binding prediction servers: Applications for vaccine research. *BMC Bioinformatics* 9, S22.
- Lin, H.H., Ray, S., Tongchusak, S., Reinherz, E.L., Brusic, V., 2008. Evaluation of MHC class I peptide binding prediction servers: Applications for vaccine research. *BMC Immunology* 9, 8.
- Melief, C.J.M., 2017. Cancer: Precision T-cell therapy targets tumours. *Nature* 547, 165–167.
- Mellor, A.L., Keskin, D.B., Johnson, T., Chandler, P., Munn, D.H., 2002. Cells expressing indoleamine 2,3-dioxygenase inhibit T cell responses. *Journal of Immunology* 168, 3771–3776.
- Meuer, S.C., Schlossman, S.F., Reinherz, E.L., 1982. Clonal analysis of human cytotoxic T lymphocytes: T4+ and T8+ effector T cells recognize products of different major histocompatibility complex regions. *Proceedings of the National Academy of Sciences* 79, 4395–4399.
- Mora, M., Veggi, D., Santini, L., Pizzi, M., Rappuoli, R., 2003. Reverse vaccinology. *Drug Discovery Today* 8, 459–464.
- Nielsen, M., Lundegaard, C., Blicher, T., *et al.*, 2007a. NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. *PLOS ONE* 2, e796.
- Nielsen, M., Lundegaard, C., Lund, O., 2007b. Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. *BMC Bioinformatics* 8, 238.
- Nielsen, M., Lundegaard, C., Lund, O., Kesmir, C., 2005. The role of the proteasome in generating cytotoxic T-cell epitopes: Insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics* 57, 33–41.
- Nielsen, M., Lundegaard, C., Worning, P., *et al.*, 2003. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science* 12, 1007–1017.
- Nielsen, M., Lund, O., 2009. NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinformatics* 10, 296.
- Nkolola, J.P., Bricalot, C.A., Cheung, A., *et al.*, 2014. Characterization and immunogenicity of a novel mosaic M HIV-1 gp140 trimer. *Journal of Virology* 88, 9538–9552.
- Nussbaum, A.K., Kuttler, C., Hader, K.P., Rammensee, H.G., Schild, H., 2001. PAPProC: A prediction algorithm for proteasomal cleavages available on the WWW. *Immunogenetics* 53, 87–94.
- Olsen, L.R., Tongchusak, S., Lin, H., Reinherz, E.L., Brusic, V., Zhang, G.L., 2017. TANTIGEN: A comprehensive database of tumor T cell antigens. *Cancer Immunology, Immunotherapy* 66, 731–735.

- Olsen, L.R., Zhang, G.L., Reinherz, E.L., Brusic, V., 2011. FLAVIdB: A data mining system for knowledge discovery in flaviviruses with direct applications in immunology and vaccinology. *Immunome Research* 7 (3).
- Ott, P.A., Hu, Z., Keskin, D.B., *et al.*, 2017. An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature* 547 (7662), 217–221.
- Parker, K.C., Bednarek, M.A., Coligan, J.E., 1994. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *Journal of Immunology* 152, 163–175.
- Peters, B., Sette, A., 2005. Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. *BMC Bioinformatics* 6, 132.
- Peters, B., Bui, H.H., Frankild, S., *et al.*, 2006. A community resource benchmarking predictions of peptide binding to MHC-I molecules. *PLOS Computational Biology* 2, e65.
- Purcell, A.W., Gorman, J.J., 2004. Immunoproteomics: Mass spectrometry-based methods to study the targets of the immune response. *Molecular & Cellular Proteomics* 3, 193–208.
- Qi, Q., Liu, Y., Cheng, Y., *et al.*, 2014. Diversity and clonal selection in the human T-cell repertoire. *Proc. Natl. Acad. Sci. USA* 111 (36), 13139–13144.
- Rammensee, H., Bachmann, J., Emmerich, N.P., Bacher, O.A., Stevanovic, S., 1999. SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* 50, 213–219.
- Rappuoli, R., 2000. Reverse vaccinology. *Current Opinion in Microbiology* 3, 445–450.
- Rappuoli, R., Bottomley, M.J., D'Oro, U., Finco, O., De Gregorio, E., 2016. Reverse vaccinology 2.0: Human immunology instructs vaccine antigen design. *Journal of Experimental Medicine* 213, 469–481.
- Reche, P.A., Glutting, J.P., Reinherz, E.L., 2002. Prediction of MHC class I binding peptides using profile motifs. *Human Immunology* 63, 701–709.
- Reche, P.A., Glutting, J.P., Zhang, H., Reinherz, E.L., 2004. Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. *Immunogenetics* 56, 405–419.
- Riemer, A.B., Keskin, D.B., Zhang, G., *et al.*, 2010. A conserved E7-derived cytotoxic T lymphocyte epitope expressed on human papillomavirus 16-transformed HLA-A2 + epithelial cancers. *Journal of Biological Chemistry* 285, 29608–29622.
- Robinson, J., Halliwell, J.A., Hayhurst, J.D., *et al.*, 2015. The IPD and IMGT/HLA database: Allele variant databases. *Nucleic Acids Research* 43 (Database issue), D423–D431.
- Rudensky, A., Preston-Hurlburt, P., Hong, S.C., Barlow, A., Janeway Jr., C.A., 1991. Sequence analysis of peptides bound to MHC class II molecules. *Nature* 353, 622–627.
- Sahin, U., Derhovanessian, E., Miller, M., *et al.*, 2017. Personalized RNA mutanome vaccines mobilize poly-specific therapeutic immunity against cancer. *Nature* 547, 222–226.
- Saveanu, L., Carroll, O., Hassaniya, Y., van Ender, P., 2005. Complexity, contradictions, and conundrums: Studying post-proteasomal proteolysis in HLA class I antigen presentation. *Immunological Reviews* 207, 42–59.
- Saxova, P., Buus, S., Brunak, S., Kesmir, C., 2003. Predicting proteasomal cleavage sites: A comparison of available methods. *International Immunology* 15, 781–787.
- Schuler, M.M., Nastke, M.D., Stevanovic, S., 2007. SYFPEITHI: Database for searching and T-cell epitope prediction. *Methods in Molecular Biology* 409, 75–93.
- Sette, A., Sidney, J., 1999. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. *Immunogenetics* 50, 201–212.
- Shortman, K., Egerton, M., Spangrude, G.J., Scollay, R., 1990. The generation and fate of thymocytes. *Semin Immunol* 2 (1), 3–12.
- Sidney, J., Grey, H.M., Kubo, R.T., Sette, A., 1996. Practical, biochemical and evolutionary implications of the discovery of HLA class I supermotifs. *Immunology Today* 17, 261–266.
- Simon, C., Kudahl, U.J., Sun, J., *et al.*, 2015. FluKB: A knowledge-based system for influenza vaccine target discovery and analysis of the immunological properties of influenza viruses. *Journal of Immunology Research* 2015, 380975.
- Singh, H., Raghava, G.P., 2001. ProPred: Prediction of HLA-DR binding sites. *Bioinformatics* 17, 1236–1237.
- Singh, H., Raghava, G.P., 2003. ProPred1: Prediction of promiscuous MHC Class-I binding sites. *Bioinformatics* 19, 1009–1014.
- Stephenson, K.E., D'Couto, H.T., Barouch, D.H., 2016. New concepts in HIV-1 vaccine development. *Current Opinion in Immunology* 41, 39–46.
- Stern, L.J., Brown, J.H., Jardetzky, T.S., *et al.*, 1994. Crystal structure of the human class II MHC protein HLA-DR1 complexed with an influenza virus peptide. *Nature* 368, 215–221.
- Strehl, B., Seifert, U., Kruger, E., *et al.*, 2005. Interferon-gamma, the functional plasticity of the ubiquitin-proteasome system, and MHC class I antigen processing. *Immunological Reviews* 207, 19–30.
- Sturniolo, T., Bono, E., Ding, J., *et al.*, 1999. Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. *Nature Biotechnology* 17, 555–561.
- Trolle, T., Metushi, I.G., Greenbaum, J.A., *et al.*, 2015. Automated benchmarking of peptide-MHC class I binding predictions. *Bioinformatics* 31, 2174–2181.
- Van Regenmortel, M.H., 2009. What is a B-cell epitope? *Methods in Molecular Biology* 524, 3–20.
- Vigneron, N., Stroobant, V., Van den Eynde, B.J., van der Bruggen, P., 2013. Database of T cell-defined human tumor antigens: The 2013 update. *Cancer Immunology* 13, 15.
- Vita, R., Overton, J.A., Greenbaum, J.A., *et al.*, 2015. The immune epitope database (IEDB) 3.0. *Nucleic Acids Research* 43 (Database issue), D405–D412.
- Wang, J.H., Reinherz, E.L., 2002. Structural basis of T cell recognition of peptides bound to MHC molecules. *Molecular Immunology* 38, 1039–1049.
- Wang, P., Sidney, J., Dow, C., *et al.*, 2008. A systematic assessment of MHC class II peptide binding predictions and evaluation of a consensus approach. *PLOS Computational Biology* 4, e1000048.
- Yusim, K., Richardson, R., Tao, N., *et al.*, 2005. Los alamos hepatitis C immunology database. *Applied Bioinformatics* 4, 217–225.
- Zhang, G.L., Ansari, H.R., Bradley, P., *et al.*, 2011. Machine learning competition in immunology – Prediction of HLA class I binding peptides. *The Journal of Immunological Methods* 374, 1–4.
- Zhang, G.L., DeLuca, D.S., Keskin, D.B., *et al.*, 2011. MULTIPRED2: A computational system for large-scale identification of peptides predicted to bind to HLA supertypes and alleles. *The Journal of Immunological Methods* 374, 53–61.
- Zhang, G.L., Keskin, D.B., Chitkushhev, L., Reinherz, E.L., Brusic, V., 2015. EBVdb: A data repository and analysis platform for knowledge discovery in Epstein-Barr virus with applications in T cell immunotherapy. In: *Proceedings of the International Conference on Swarm Intelligence (ICSI3)*. Taormina, Italy.
- Zhang, G.L., Lin, H.H., Keskin, D.B., Reinherz, E.L., Brusic, V., 2011. Dana-Farber repository for machine learning in immunology. *The Journal of Immunological Methods* 374, 18–25.
- Zhang, G.L., Riemer, A.B., Keskin, D.B., *et al.*, 2014. HPVdb: A data mining system for knowledge discovery in human papillomavirus with applications in T cell immunology and vaccinology. *Database (Oxford)* 2014, bau031.
- Zhang, G.L., Sun, J., Chitkushhev, L., Brusic, V., 2014. Big data analytics in immunology: A knowledge-based approach. *BioMed Research International* 2014, 437987.
- Zhang, H., Lund, O., Nielsen, M., 2009. The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: Application to MHC-peptide binding. *Bioinformatics* 25, 1293–1299.
- Zhang, L., Chen, Y., Wong, H.S., Zhou, S., Mamitsuka, H., Zhu, S., 2012. TEPITOPEan: Extending TEPITOPE for peptide binding prediction covering over 700 HLA-DR molecules. *PLoS One* 7, e30483.

Relevant Website

http://tools.iedb.org/auto_bench/mhcii/weekly/

MHC II Automated Server Benchmarks – IEDB Analysis Resource.

Characterizing and Functional Assignment of Noncoding RNAs

Pradeep Tiwari, Manipal University, Jaipur, India; Birla Institute of Scientific Research, Jaipur, India; and SMS Medical College, Jaipur, India

Sonal Gupta, Birla Institute of Scientific Research, Jaipur, India and Amity University, Jaipur, India

Anuj Kumar and Mansi Sharma, Uttarakhand Council for Biotechnology, Dehradun, India

Vijayaraghava S Sundararajan, Ministry of Environmental Agency, Singapore and Bioclues.org, Hyderabad, India

Shanker L Kothari, Amity University, Jaipur, India

Sandeep K Mathur, SMS Medical College, Jaipur, India

Krishna M Medicherla and Prashanth Suravajhala, Birla Institute of Scientific Research, Jaipur, India

Babita Malik, Manipal University, Jaipur, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Noncoding RNAs (ncRNAs) are a family of RNA molecules that do not code for proteins. A large number of studies on cell proliferation, metabolism, epigenetic, and physiological processes (Gao *et al.*, 2017) have shown their role specific to regulation leading to a number of molecular mechanisms (Wang and Chang, 2011). Essentially ncRNAs are classified into small ncRNAs, which include microRNAs (miRNAs) and circular RNAs (CiRNAs); and long intergenic noncoding RNAs (lincRNAs) and long noncoding RNAs (lncRNAs) (Shah *et al.*, 2016). Recent human whole transcriptomic data have revealed that less than 2% of the genome encodes protein-coding transcripts, even though the majority of the genome is actively transcribed into ncRNAs under multiple physiological conditions (Liz and Esteller, 2016). Diseases that are associated with an altered transcriptome are not only restricted to the abnormal production of protein-coding RNAs, but also in the expression of many noncoding RNAs. Although there has been an increase in evidence for the link between ncRNAs and diverse human diseases, studies on ncRNA–protein association studies have caught an immense interest and have opened a new possibility in identifying cause for diseases (Wapinski and Chang, 2011). This has allowed us to improve next generation sequencing (NGS) tools for identifying functional moieties, for example, pseudogenes in making novel proteins (Shidhi *et al.*, 2015).

Background

With transcriptome and high-throughput sequencing (HITS) analysis revealing a large number of ncRNAs in various organisms, studies on their functional characterization have resulted in data on interactions with their RNA peers, DNA, or proteins (Novikova *et al.*, 2013). While the ncRNAs are believed to be associated with a wide number of human diseases including development, imprinting, mental and psychiatric disorders, and tumor growth (Liu *et al.*, 2012), they are also associated with plant growth, environmental stress, biological processes and development, besides capacity to encode small peptides (Lv *et al.*, 2016). The miRNAs play an important role in posttranscriptional gene regulation by either translation repression or directing the degradation of complementary mRNA targets. The miRNA molecules are derived from the primary miRNAs (pri-miRNAs) and negatively regulate the gene expression in various plants and animals. It was also known that the plant pri-miRNAs containing small open reading frames (ORFs) have the capacity to encode regulatory peptides (Lauressergues *et al.*, 2015). It has been reported that primary transcripts of plant miRNAs encode peptides (miPEPs) and increases the transcription of other associated miRNAs (Couzigou *et al.*, 2015). Unlike miRNAs, which are short and ca. 22–23 bp, lncRNAs are more than 200 nucleotides without ORFs. Most of these transcripts are coded by RNA polymerase II, either spliced or poly-adenylated. Whereas the ncRNAs play a role as key regulators in life cycle of the organism, this concept of regulation driven by ncRNAs has begun to be understood in mitochondria and chloroplasts as well (Lung *et al.*, 2006). For example, nuclear-encoded miRNAs were found to be associated with mitochondrial transcription and translation in eukaryotes (Rackham *et al.*, 2011). In addition, ncRNAs of various sizes are thought to play a role in RNA interference or antisense pathways (Kaczmarek *et al.*, 2017). Like the traditional proteins piggy-backing from nucleus/cytoplasm to mitochondria, the ncRNAs also operate in the nucleus or cytoplasm while targeting the mitochondria. With regards to numerous ncRNAs identified in chloroplasts, they have been known to map to the plastid genome (Dietrich *et al.*, 2015) and play a role in various functions of the plant cell. A broad classification of ncRNAs based on their mode of function is presented in Fig. 1.

While transcription of ncRNAs is done by RNA polymerase II, the transcripts are capped and polyadenylated (Kim, 2005). Classes of small RNAs such as short interfering RNAs (siRNAs), piwi-interacting RNAs (piRNAs), and miRNAs found in eukaryotes regulate endogenous genes and defend the genome from invasive nucleic acids. For example, in 1993, the first miRNA, *lin-4* from *Caenorhabditis elegans* was discovered by Ambros, which is known to control developmental timing (Bartel, 2004). In addition, their role in genome defense is documented (Mello and Conte, 2004).

The pri-miRNA transcribed from DNA is excised by Drosha, to produce the pre-miRNA (Fig. 2). The pre-miRNA is exported to the cytoplasm by exportin-5 and spliced by Dicer to generate a miRNA duplex, which is then assembled into the RNA-induced silencing complex (Bartel, 2004). The miRNAs modulate target mRNA translation by complementary

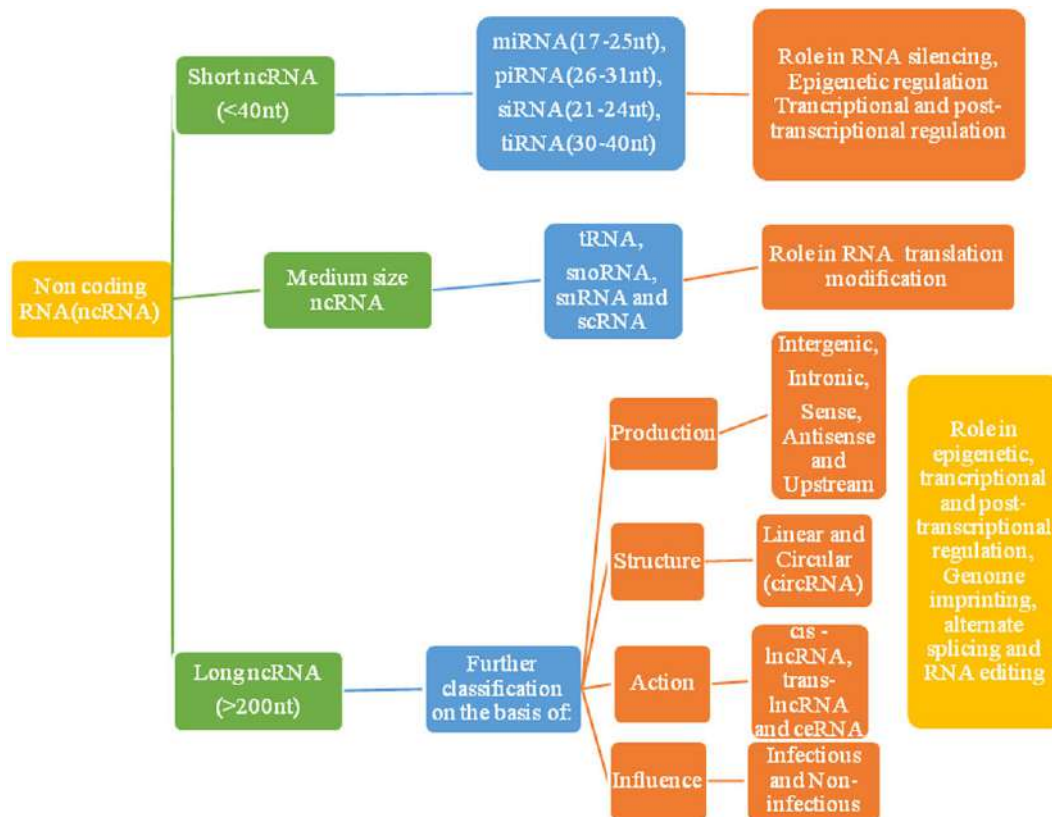


Fig. 1 Classification of ncRNAs based on different modes of action and function. Conceptualized from Katsarou *et al.* (2015), Hou and Bonkovsky, 2013. Non-coding RNAs in hepatitis C-induced hepatocellular carcinoma: Dysregulation and implications for early detection, diagnosis and therapy. *World Journal of Gastroenterology* 19(44), 7836–7845. Dey *et al.*, 2012. Non-micro-short RNAs: The new kids on the block. *Molecular Biology of the Cell* 23(24), 4664–4667. Abbreviations: ncRNA: noncoding RNA, miRNA: micro RNA, piRNA: piwi RNA, siRNA: small interfering RNA, tiRNA: transcription initiation RNA, tRNA: transfer RNA, snoRNA: small nucleolar RNA, snRNA: small nuclear RNA, scRNA: small cytoplasmic RNA, *cis*-lncRNA: *cis*-acting long noncoding RNA, *trans*-lncRNA: *trans*-acting long noncoding RNA, ceRNA: competing endogenous RNA.

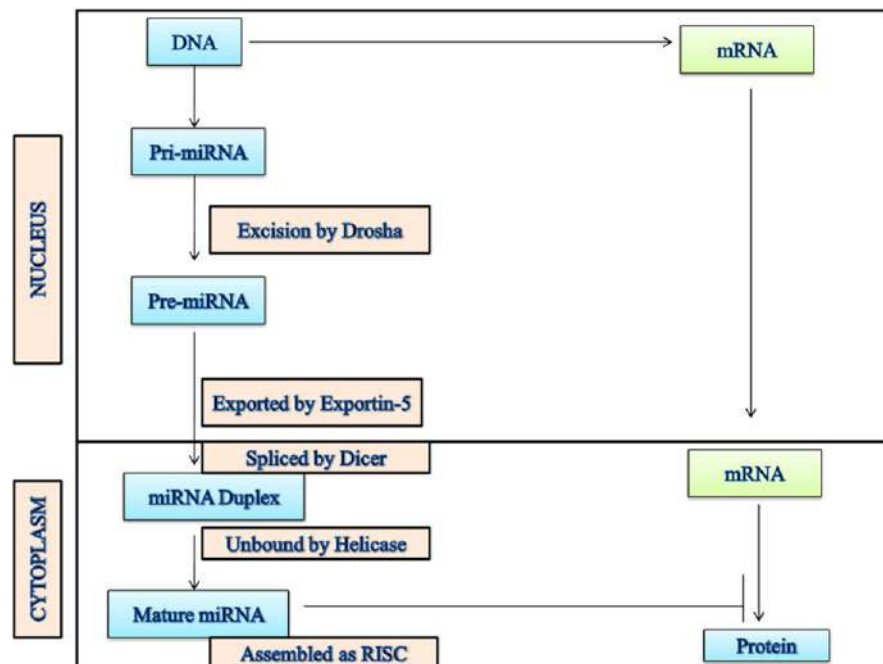


Fig. 2 Illustrative representation of formation of mature miRNAs. The primary miRNAs synthesized in nucleus are exported to cytoplasm by exportin later spliced by Dicer before assembly.

binding, within the 3' UTR, 5' UTR, or coding regions. The miRNAs also function by binding to DNA promoters, acting as RNA decoys or through direct binding of target mRNAs. Small RNAs bounded with effector proteins target the nucleic acids and sets of macromolecules, such as Dicer, Argonaute protein and ~21–23 nucleotide duplex-derived RNAs are recognized as main components in RNA silencing (Meister and Tuschl, 2004). While Dicer plays a role in RNA gene silencing, Dicer-1 is required for miRNA biogenesis and Dicer-2 in the siRNA pathway (Tomari and Zamore, 2005). The second core component of the effector machinery is a member of the Argonaute protein superfamily that functions in eukaryotic gene silencing, which is also present in bacterial and archaeal species. The argonaute protein superfamily can be divided into three subgroups: The Piwi clade that binds to piRNAs, the Ago clade binds to miRNAs and siRNAs, and a third clade described only in nematodes thus far (Yigit *et al.*, 2006). While it is known that the eukaryotic genes are regulated by ncRNAs, validating miRNAs from their progenitor mRNAs is needed for assigning putative function (Singh *et al.*, 2013). However, it is not so clear how RNA regulatory networks have existed with the emergence of rRNAs, tRNA, and other progenitors. Theories on their presence in earlier stages of life in archaea and bacteria have been documented (Daly *et al.*, 2013) as well as questions on how classes of ncRNAs play a role in human diseases (Esteller, 2011). The ncRNAs in bacteria serve as one of the functional elements and play an important role in gene regulation, predominantly at the post-transcription level. Primarily, they can be divided into cis-encoded and trans-encoded ncRNAs, RNA thermometers, and riboswitches, small noncoding RNAs that influence the translation and stability of mRNAs by binding to the base-pairing sites in their target transcripts. In pathogenic bacteria, numerous ncRNAs are involved in the coordinated expression of virulence determinants to facilitate the pathogenicity (Han *et al.*, 2013). With recent reports revealing that there are around 15,000 lincRNAs encoded by the human genome (Deniz and Eрман, 2017), the lincRNAs in eukaryotes have also been studied in establishing genotype–phenotype associations, which has led to advancements in prognosis, diagnosis, and treatment of human disorders.

In all living organisms, molecular machinery is activated by long double-stranded RNAs to transcribe the RNAs into short miRNAs. The miRNAs are generated from long double-stranded RNAs where they are produced by their own genes or specific part of sequences of the protein-coding genes (Yao, 2016). The biogenesis of these miRNA molecules starts with the synthesis of pri-miRNAs (Bartel, 2004; Ha and Kim, 2014; Yao, 2016). While pri-miRNAs are transcribed in the form of polycistronic transcripts, they also arise from individual transcripts constituting exonic, intergenic, or intronic sequences (Kim and Nam, 2006). Based on the location of the miRNA generation, miRNAs can be classified into two classes (Wang *et al.*, 2009): (1) intergenic miRNAs generated from transcripts of miRNA genes located inside protein-coding genes; (2) intragenic miRNAs generated from transcripts of intronic or exonic sequences of the protein-coding genes. Whereas features of ncRNAs, such as TATA box, CpG islands, TBIIF recognition, histone modifications, and initiator elements are much similar to protein-coding genes (Ozsolak *et al.*, 2008; Corcoran *et al.*, 2009), it is believed that the same regulation mechanisms of gene expression are applicable to miRNA molecules as well. RNA polymerases II and III facilitate a wide range of regulatory options and the miRNAs encoded as clusters are transcribed in the form of long polycistronic primary transcripts (Melo and Melo, 2014) or regulated autonomously (Diederichs and Haber, 2006; Yanaihara *et al.*, 2006). For example, the tumor-suppressor p53 is responsible to activate the entire miR-34 family, which plays a critical role in transactivation or repression in cell cycle mediated apoptosis (He *et al.*, 2005; Chang *et al.*, 2008). The transcription activation found in response to various growth factors such as transforming growth factor- β , platelet-derived growth factor, and bone-derived neurotrophic factor have been well studied in lieu of miRNA regulation (Corcoran *et al.*, 2009; Chan *et al.*, 2010). Further studies on epigenetics significant to miRNA gene regulation for understanding the methylation of CpG islands have shown to contribute to silence their transcription (Lujambio *et al.*, 2007, 2008; Davalos *et al.*, 2012). Specific promoters of miRNA genes are also regulated by the histone modifications process during various developmental stages and disease (Scott *et al.*, 2006).

Similarly, the three forms of lincRNAs, such as antisense lincRNAs, intronic lincRNAs and long intergenic noncoding RNAs (lincRNAs), which are the classes of lincRNAs, are spliced from multiexonic precursors (Ma *et al.*, 2013). Their role on regulation and evolutionary conservation and predicting secondary structures have been well documented (Johnsson *et al.*, 2014; Yan *et al.*, 2016). With the advent of NGS technologies, several novel lincRNAs have been identified using various approaches. How the lincRNA–protein interactions play a role in immunomodulation further taking part in the gene regulatory networks has been shown (Suravajhala *et al.*, 2015; Weikard *et al.*, 2013; Kung *et al.*, 2013). Furthermore, as lincRNAs are identified from whole transcriptome or RNA-Seq methods, interestingly a few lincRNAs have also been reported from whole exome sequencing as well (Pan *et al.*, 2016a,b). The nuclear lincRNAs in particular are known to play a major role in regulation especially serving as both *cis*- and *trans*-regulators. For example, HOTAIR, a class of lincRNAs, is well transcribed from regulatory regions and further plays a role in regulatory networks. A myriad of other ncRNAs in the form of enhancer RNAs (eRNAs) and complementary enhancer RNAs (ceRNAs) play a role in such processes. On the other hand, cytoplasmic lincRNAs are known to serve as endogenous “sponges” for miRNAs, which in turn repress miRNA repression on targeting genes (Rashid *et al.*, 2016; Thomson and Dinger, 2016). Many lincRNA–protein interactions have been shown to be associated with coactivation, repression, sequestration and organelle formation (Ferre *et al.*, 2016) while showing expression at multiple levels in the cell cycle (Paralkar and Weiss, 2013). In addition, the transposable elements (TE) have contributed widely to genome evolution and are known to play a role in maturity of noncoding transcripts (Hadjigryrou and Delihias, 2013). Conversely, the TEs playing a role in formation of ncRNAs especially lincRNA transcripts have accounted for about 30% portion of total sequences in human. While these are rarely found in protein-coding transcripts (Kapusta *et al.*, 2013), they are known to help cause signals essential for the biogenesis of many lincRNAs such as transcription initiation, splicing, or polyadenylation in human. Their role, however, in development of biomarkers has recently begun to be understood (Xi *et al.*, 2017).

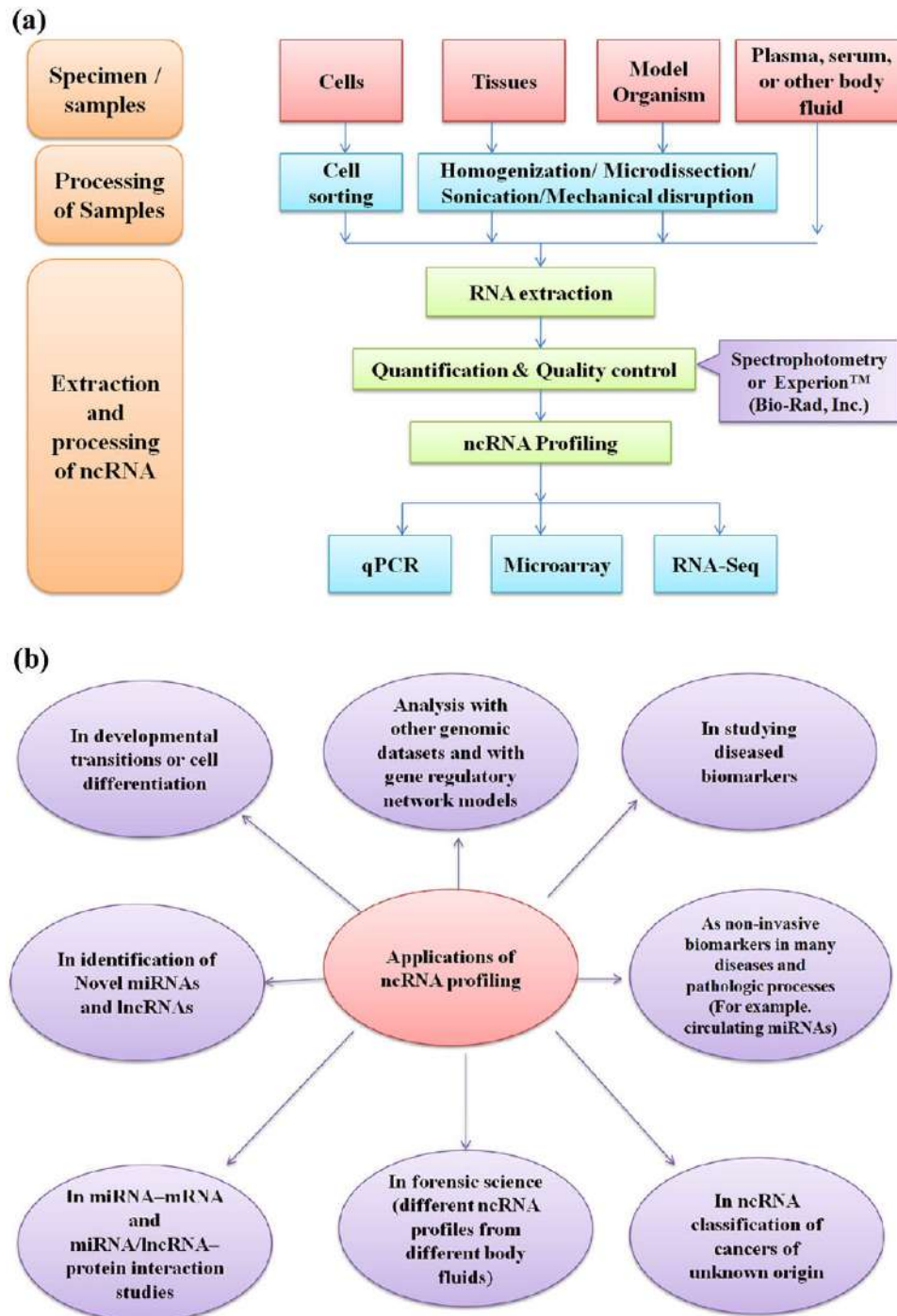


Fig. 3 (a) The ncRNAs can be extracted from various specimen types such as cultured cells, model organisms, tissue and body fluids such as plasma and serum. Homogenization, laser capture microdissection (LCM), and cell sorting can be used to purify specific cell types. The ncRNA populations can be purified by gel electrophoresis and AGO2 immunoprecipitation (AGO2-IP) along with or without ultraviolet crosslinking and immunoprecipitation (CLIP). The ncRNA isolation can be done using commercially available extraction kits. miRNA quality can be assessed after using various methods that includes spectrophotometry, automated capillary electrophoresis with Bioanalyzer or Experion. (b) Applications of ncRNA profiling. Abbreviations: qPCR, quantitative PCR; RNA-Seq, RNA sequencing. Image conceptualized from Pritchard *et al.*, 2012. *MicroRNA profiling: approaches and considerations*. *Nature Reviews Genetics* 13, 358–369.

Comparing the ncRNA profiles between different stages of development has facilitated in developmental transitions or cellular differentiation. Whole transcriptome or RNA sequencing (RNA-Seq) is an important approach for both identifying and profiling of novel ncRNAs. The miRNAs linked with mRNA or with RNA-binding protein can be analyzed using crosslinking immunoprecipitation (CLIP) using ultraviolet rays to create covalent bonds between RNA-binding proteins and RNA (Fig. 3).

Table 1 A gist of ncRNA databases and tools along with their association with other small molecules

Type	Database name	Function	Strengths	Weaknesses, if any	Reference	URL
Micro RNA databases	miRBASE	Database of published miRNA sequences and annotation	User-friendly with good number of options for analyzing the miRNA. A new option to identify the novel miRNAs	miRNA-miRNA interactions cannot be ascertained completely	Griffiths-Jones <i>et al.</i> (2006)	http://www.mirbase.org/
	miRwalk	Database on predicted and validated microRNA targets	A whole set of miRNA targets including pathway information is made accessible through this. A comprehensive list of targets including other ncRNAs is detailed. Users have the flexibility to study any custom miRNAs or target genes of interest, related to functional miRNA annotations	Targeted regions are many. A careful double check is required to analyze them, Not so user-friendly	Dweep <i>et al.</i> (2011)	http://zmf.umm.uni-heidelberg.de/apps/zmf/mirwalk/
	miRDB	MicroRNA target prediction and functional study database	Users have the flexibility to study any custom miRNAs or target genes of interest, related to functional miRNA annotations	–	Wong and Wang (2015)	http://www.mirdb.org/
	miRTarBase	Experimentally validated microRNA-target interactions database	Comprehensively annotated, experimentally validated miRNA-target interactions database		Hsu <i>et al.</i> (2011)	http://mirtarbase.mbc.nctu.edu.tw/
	VirMIRDB	Predicts viral microRNA candidate hairpins	Beneficial for viral miRNAs and host-gene interaction studies	Not so user-friendly, Execution of the entire bioinformatics' pipeline is tough to scan for putative miRNA hairpins	Li <i>et al.</i> (2008)	http://alk.ibms.sinica.edu.tw/cgi-bin/miRNA/miRNA.cgi
LncRNA-protein interaction databases	NONCODE	An integrated knowledge database dedicated to ncRNAs, especially lncRNAs	Provides both basic and advance information about LncRNA such expression profile, conservation info, predicted function and disease relation	Does not give information about tRNAs and rRNAs	Zhao <i>et al.</i> (2016)	http://www.noncode.org/
	LncRNAwiki	Community-curated lncRNA knowledgebase	Integrates information on human lncRNAs from multiple different resources	Relies on community intelligence to curate a wide range of lncRNA-related topic, improvement is needed to link with other existing relevant databases	Ma <i>et al.</i> (2015)	http://lncrna.big.ac.cn/index.php/Main_Page
	LncRNAdb	The reference database for functional long noncoding RNAs	User-friendly and thoroughly compiled and updated information of lncRNAs in a variety of systems	Limited to the location of lncRNA only	Amaral <i>et al.</i> (2011)	http://www.lncrnadb.org/
LncRNA-protein interaction prediction databases	Interaction Pattern (IP) Miner	Predicts ncRNA-protein interactions from protein sequences	Good performance for mining the ncRNA-protein interactions	Bash/command line tools and applicable for commandline user interface (GUI). Uses Python scripts	Pan <i>et al.</i> (2016a, b)	http://www.csbio.sjtu.edu.cn/bioinf/IPMiner/IPMiner.tar.gz

(Continued)

Table 1 Continued

Type	Database name	Function	Strengths	Weaknesses, if any	Reference	URL
Binding site databases	IncPro	Predicts the interaction between long noncoding RNAs and proteins	Computational friendly	Interaction data is still considered unsatisfactory because of the large number of proteins. <i>Bona fide</i> ity is at risk	Lu <i>et al.</i> (2013)	http://bioinfo.bjmu.edu.cn/incpro/
	ncRNAs and protein related bio macromolecules interaction database	Integrates experimentally verified functional interactions between noncoding RNAs and other biomolecules (proteins, RNAs and genomic DNAs)	Novel integrated functional interactions between ncRNAs and protein related macromolecules (PRM) into a single, easily accessible database	Interactions are manually collected from publication, however, they are curated	Wu <i>et al.</i> (2006)	http://www.bioinfo.org/NP/Inter
	RNA-protein Association and Interaction Networks	A database of ncRNA-RNA and ncRNA-protein interactions	Integrated with the STRING database	Does not give information about RNA-protein binding site integration	Junge <i>et al.</i> (2017)	https://rth.dk/resources/rain/
	Protein-RNA interaction prediction	Predicts structural fragments information of given protein and RNA	User-friendly with support-vector machine-based method. Predicts protein-RNA interaction pairs, based on both the sequences and structures	Does not give information and determines the binding partners for other types of proteins which are able to interact with both DNA and RNA	Suresh <i>et al.</i> (2015)	http://ctsb.is.wfubmc.edu/projects/rpi-pred
	RNA-associated interaction database	Predicts RNA-associated (RNA-Protein/RNA-RNA) interactions	Users can query, browse, analyze, and manipulate RNA-associated (RNA-RNA/RNA-protein) interaction		Zhang <i>et al.</i> (2014)	http://www.rna-society.org/raid/
	Protein-RNA Interface Database (PRIDB)	Database of RNA-Protein interfaces	It displays interfacial amino acids and ribonucleotides within the primary sequences of the interacting protein and RNA chains. PRIDB also identifies ProSite motifs in protein chains and FR3D motifs in RNA chains and provides links to external databases	Highly reliant on Java. Interfaces in the RNA-protein complexes can be visualized using an integrated JMol applet or downloaded in a machine-readable format	Lewis <i>et al.</i> (2011)	http://bindr.gdcb.iastate.edu/PRIDB
Bacteria	RNA-Binding Protein Database (RBPDB)	Repository of experimental observation of RNA-binding sites	Freely accessible by a web interface. Can also use to scan sequences for RBP-binding sites	Currently populated only with data from metazoans	Cook <i>et al.</i> (2011)	http://rbpdb.ccbr.utoronto.ca
	RsiteDB	Describes, classifies and predicts the interactions between RNA and protein binding pockets	Provides information about the protein binding pockets that interact with single-stranded RNA nucleotide bases	Uses Java to visualize the results through Java applets	Shulman-Peleg <i>et al.</i> (2009)	http://bioinfo3d.cs.tau.ac.il/RsiteDB
	NOCO RNAc	A Java based tool for the prediction and characterization of ncRNA transcripts in bacteria	Characterizes noncoding RNAs in prokaryotes. Gives genome-wide prediction of ncRNA transcripts in bacteria	Does not predict, for example, 23S ribosomal RNA	Herbig and Nieselt (2011)	http://www.swmath.org/software/17423

RNA sequences bound by a protein of interest are then detected by high-throughput sequencing in combination with HITS-CLIP or photoactivatable ribonucleoside-enhanced CLIP (PAR-CLIP) using photoreactive analogs. Integrative analyses of miRNAs and lncRNAs are done *in silico* using MAGIA95 and mirConnX96, Cytoscape, a network visualization tool, as well as with other databases such as miRbase, lncRNAdb, and NONCODE (see [Table 1](#)). Such profiling has enabled study of miRNAs as biomarkers for a wide variety of diseases such as cancer, autoimmune, psychiatric, and neurological disorders. While it has been applied successfully for the cancer classification of unknown origin, various clinical diagnostic assays those including specific cancer-derived miRNAs and different body fluids have been characterized ([Pritchard et al., 2012](#)). In profiling lncRNAs, they can be functionally impaired by blocking their molecular interactions by applying small molecule inhibitors that mask the binding sites and further help in diagnosis of diseases ([Sanchez and Huarte, 2013](#)). For example, preventing the interaction of HOTAIR with the PRC2 or LSD1 complexes using small molecular inhibitors may limit the metastatic potential of breast cancers ([Tsai et al., 2011](#)). In this process, oligonucleotides have also been used to target a specific ncRNA region and block its correct folding ([Colley and Leedman, 2009](#)).

Illustrative Example

The role of ncRNAs in various diseases, especially cancers, has been well documented ([Esteller, 2011](#)) with less focus on stem cells. The cancer stem cells (CSC) are a small subset of cancer cells that are associated with normal stem cells and are responsible for drug/treatment resistance, tumor recurrence, and metastasis ([Garg, 2015](#)). While they were recognized few years ago, their identification and characterization have already been done continuously for 15 years in some malignancies, such as leukemia and other tumors ([Bonnet and Dick, 1997](#)). Due to self-renewal and tumor generation capacity, CSCs have become potential targets for various cancer treatments including prostate, breast, colon, head and neck, colorectal, and brain ([Huang and Rofstad, 2017](#)). It has been reported that miRNAs play a vital role in the regulation of CSCs in various types of malignant tumors at the posttranscriptional level ([Ali et al., 2013](#)) while it is widely accepted that CSCs play a significant role in cell biology of cancers. Further, they are known to work via signaling pathways that mediate the expression through self-renewal, such as Notch and Wnt ([Prokopi et al., 2015](#)). In several studies many miRNAs showed different expression profiles in the various types of cancer ([Ma et al., 2015](#)) and their role as potential targets for cancer therapy is widely known ([Fig. 4](#)) ([Bimonte et al., 2016](#)). Based on their important roles in progression of human cancer and expression profile miRNAs are classified into two groups: (1) oncogenic miRNAs, which are reported to upregulate in tumor cells ([Bao et al., 2011](#); [Albulescu et al., 2011](#)); (2) tumor suppressor miRNAs, which are reported to be downregulated in pancreatic cancer ([Ji et al., 2009](#); [Haselmann et al., 2014](#)). Reports on downregulation of brain specific neuromiR-124 in glioblastoma has shown to improve the number of CSCs and oncogenic capacity ([Bian and Sun, 2011](#)) and let-7 downregulation in lung cancer linked with poor survival ([Calin and Croce, 2006](#)). Likewise, pneumomiR-29, a lung specific miRNA, is shown to suppress the tumorigenicity in non-small cell lung cancer cells while two other, such as miR-143 and miR-145, have been reported to be downregulated in

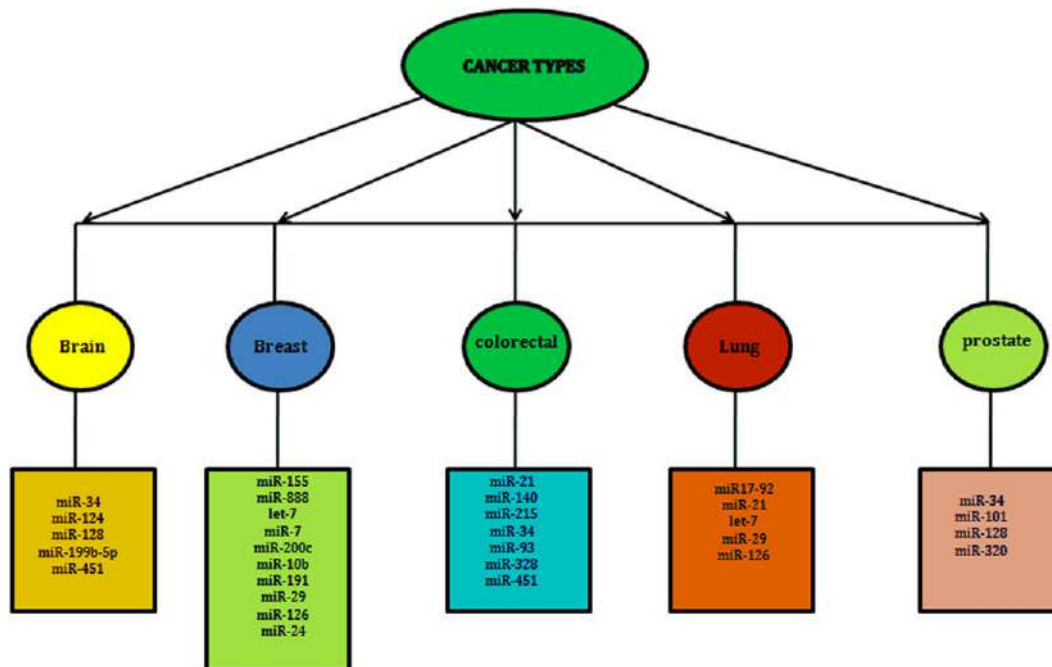


Fig. 4 The miRNAs targeted to various major cancer types.

various cancers including breast, cervical, and colorectal (Prokopi *et al.*, 2015). Despite these results, more studies on the function of ncRNAs in biology of CSCs will be needed in the future in order to discover novel oncology therapeutics (Parasramka *et al.*, 2016).

Technological developments enabled us to get volumes of data using the high-throughput NGS technologies. Evidence suggests that large amount of transcripts especially miRNAs have been modeled (Friedlander *et al.*, 2008; Lu *et al.*, 2009). The first reports on analyzing the RNA data to search new ncRNA gene sequences and predicting their targets came in the year 2003 (Lim *et al.*, 2003). While computational modeling was employed to use precursors as input and functional end products as ncRNAs, previous approaches used finding new miRNAs based on sequence data (Mendes *et al.*, 2009). Xie *et al.* (2005) reported artificial intelligence using learning algorithms used with sets of test known data and targets are predicted. Since then, many new ncRNA annotations were reported and put in use for the public domain. With high-throughput deep sequencing creating much more roadmaps for the ncRNA discovery (Lu *et al.*, 2005), hairpin structures are used to mimic miRNA byproducts (Friedlander *et al.*, 2008) and least-square classification algorithm for miRNAs (Lu *et al.*, 2009). While earlier methods were used to predict ncRNAs understanding ncRNA biogenesis and target specificity, HITS approaches involving bioinformatics simulations/predictions are needed. Even though constraints in using sequence data would be hard to predict ncRNA computationally, complex diseases affected by a number of ncRNAs rather than a single ncRNAs have been documented (Kargul and Laurent, 2011). Characterizing transcript structures and understanding expression profiles mediating regulatory roles are in agreement with good number of databases and repositories documented for ncRNAs (Table 1). Nevertheless, with big data revolution enabling us to handle huge exponential volume of ncRNA data, developing robust mathematical models for prediction of ncRNAs would be beneficial to the research community.

Future Directions

The list of ncRNAs in the genomes has consistently grown without worthy annotation with many of them associated with diseases. Although the role of ncRNAs towards their regulation in important diseases, such as cancer, obesity, immunomodulatory diseases etc. are well reported, the larger diversity of their function, explosive growth in the known functional relationships between ncRNAs and diseases such as cancer can be categorically approached. These can be characterized based on in silico predictions carried out in the laboratory keeping in view of the following documentation:

- (1) While finding ncRNAs linked to immunomodulatory disease types, finding the various interaction partners of ncRNAs especially those that can affect protein function and localization needs to be attempted. As far as computational methods to the problem are concerned, this focus could not only allow us to process such data but also ensure we build a healthy community annotated wiki catalog of ncRNAs. This resource can serve for wet-lab validation or develop a machine learning intensive web server in predicting the interactions among the unknown ncRNA–protein pairs.
- (2) Could we expect to find ncRNAs to control protein complexes, for example whether or not any specific targets are involved in this process leading to immunomodulation. With the larger diversity of function of ncRNAs, explosive growth in functional relationships of ncRNAs and proteins causal to diseases would be categorically approached. We expect that a majority of ncRNAs might be the direct descendants from the known unknown regions or the domains of unknown function, which possibly would provide clues on ncRNAs binding to disordered regions of the proteins. If it were the case, the bound residues intrinsic to the immunomodulated state can be targeted for downstream experiments in the laboratory to identify prognostic biomarkers.
- (3) Identifying regulatory elements associated with a disease and validating them post exome sequencing using Sanger sequencing has been a routine task. In this process, distinct signatures are discovered among various populations across disease spectrum. What remains as a challenge is to identify those that are most previously uncharacterized and find if they are specific to population. Notable among such identified mutations or variants could be the one that pops up from ncRNAs where these mutation hotspots are seen. If by the advent of sequencing chemistry going wrong, a study defining exome mutational spectrum would highlight discovery of ncRNA mutations playing an important role in therapeutic targets.
- (4) Can CRISPR/Cas9 delete a ncRNA gene? Can these be used as targets for them, for example lncRNAs? As lncRNAs make an effect on its neighboring genes, it would be interesting to see the mutations from these genes that are associated with it. If they modulate the expression, that would be an interesting case to further explore which will allow lncRNA sequences to serve as protospacer adjacent motifs in the genome. Additionally, molecular causes especially studying regulatory interactions between classes of ncRNAs and proteins would have a significant influence on the organism and diseases remains a challenge. A hope that the NGS technologies have an answer subtly proves this.

Acknowledgments

PS thanks Asif Khan and Shobha Ranganathan for inviting us to write a book article.

Authors' contributions

PT, SG, and PS wrote the entire initial draft with AK and MS focusing on miRNAs and diseases. SM, SLK, BM, and KM wrote the parts on regulation and biogenesis of ncRNAs, and VSS on big data challenges. PS proofread the manuscript. All authors read and approved the book article.

See also: Natural Language Processing Approaches in Bioinformatics

References

- Albulescu, R., Neagu, M., Albulescu, L., Tanase, C., 2011. Tissular and soluble miRNAs for diagnostic and therapy improvement in digestive tract cancers. *Expert Review of Molecular Diagnostics* 11, 101–120.
- Ali, A.S., Ahmed, A., Ali, S., *et al.*, 2013. The role of cancer stem cells and miRNAs in defining the complexities of brain metastasis. *Journal of Cellular Physiology* 228, 36–42.
- Amaral, P.P., Clark, M.B., Gascoigne, D.K., Dinger, M.E., Mattick, J.S., 2011. lncRNAdb: A reference database for long noncoding RNAs. *Nucleic Acids Research* 39, D146–D151.
- Bao, B., Wang, Z., Ali, S., *et al.*, 2011. Notch-1 induces epithelial-mesenchymal transition consistent with cancer stem cell phenotype in pancreatic cancer cells. *Cancer Letters* 307, 26–36.
- Bartel, D.P., 2004. MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell* 116, 281–297.
- Bian, S., Sun, T., 2011. Functions of noncoding RNAs in neural development and neurological diseases. *Molecular Neurobiology* 44, 359–373.
- Bimonte, S., Barbieri, A., Leongito, M., *et al.*, 2016. The role of miRNAs in the regulation of pancreatic cancer stem cells. *Stem Cells International* 2016, 8352684.
- Bonnet, D., Dick, J.E., 1997. Human acute myeloid leukemia is organized as a hierarchy that originates from a primitive hematopoietic cell. *Nature Medicine* 3, 730–737.
- Calin, G.A., Croce, C.M., 2006. MicroRNA signatures in human cancers. *Nature Reviews Cancer* 6, 857–866.
- Chan, M.C., Wu, C., Davis, B.N., *et al.*, 2010. Molecular basis for antagonism between PDGF and the TGF beta family of signalling pathways by control of miR-24 expression. *EMBO Journal* 29, 559–573.
- Chang, T.C., Chang, T.C., Yu, D., *et al.*, 2008. Widespread microRNA repression by Myc contributes to tumorigenesis. *Nature Genetics* 40, 43–50.
- Colley, S.M., Leedman, P.J., 2009. SRA and its binding partners: An expanding role for RNA-binding coregulators in nuclear receptor-mediated gene regulation. *Critical Reviews in Biochemistry and Molecular Biology* 44, 25–33.
- Cook, K.B., Kazan, H., Zuberi, K., *et al.*, 2011. RBPDB: A database of RNA-binding specificities. *Nucleic Acids Research* 39, D301–D308.
- Corcoran, D.L., Pandit, K.V., Gordon, B., *et al.*, 2009. Features of mammalian microRNA promoters emerge from polymerase II chromatin immunoprecipitation data. *PLOS ONE* 4, e5279.
- Couzigou, J.M., Laressergues, D., Becard, G., Combier, J.P., 2015. miRNA-encoded peptides (miPEPs): A new tool to analyze the roles of miRNAs in plant biology. *RNA Biology* 12, 1178–1180.
- Daly T., Chen S.X., Penny D., 2013. How old are RNA networks? Available at: <https://www.ncbi.nlm.nih.gov/books/NBK53775/>.
- Davalos, V., Moutinho, C., Villanueva, A., *et al.*, 2012. Dynamic epigenetic regulation of the microRNA-200 family mediates epithelial and mesenchymal transitions in human tumorigenesis. *Oncogene* 31, 2062–2074.
- Deniz, E., Erman, B., 2017. Long noncoding RNA (lincRNA), a new paradigm in gene expression control. *Functional & Integrative Genomics* 17, 135–143.
- Dey, B.K., Mueller, A.C., Dutta, A., 2012. Non-micro-short RNAs: The new kids on the block. *Molecular Biology of the Cell* 23, 4664–4667.
- Diederichs, S., Haber, D.A., 2006. Sequence variations of microRNAs in human cancer: Alterations in predicted secondary structure do not affect processing. *Cancer Research* 66, 6097–6104.
- Dietrich, A., Wallet, C., Iqbal, R.K., Gualberto, J.M., Lotfi, F., 2015. Organellar non-coding RNAs: Emerging regulation mechanisms. *Biochimie* 117, 48–62.
- Dweep, H., Sticht, C., Pandey, P., Gretz, N., 2011. miRWalk-database: Prediction of possible miRNA binding sites by “walking” the genes of three genomes. *Journal of Biomedical Informatics* 44, 839–847.
- Esteller, M., 2011. Non-coding RNAs in human disease. *Nature Reviews Genetics* 12, 861.
- Ferre, F., Colantoni, A., Helmer-Citterich, M., 2016. Revealing protein–lncRNA interaction. *Briefings in Bioinformatics* 17, 106–116.
- Friedlander, M.R., Chen, W., Adamidi, C., *et al.*, 2008. Discovering microRNAs from deep sequencing data using miRDeep. *Nature Biotechnology* 26, 407–415.
- Gao, J., Xu, W., Wang, J., Wang, K., Li, P., 2017. The role and molecular mechanism of non-coding RNAs in pathological cardiac remodeling. *International Journal of Molecular Sciences* 18, 608.
- Garg, M., 2015. Emerging role of microRNAs in cancer stem cells: Implications in cancer therapy. *World Journal of Stem Cells* 7, 1078–1089.
- Griffiths-Jones, S., Grocock, R.J., Dongen, S., Alex Bateman, A., Enright, A.J., 2006. miRBase: MicroRNA sequences, targets and gene nomenclature. *Nucleic Acids Research* 34, D140–D144.
- Ha, M., Kim, V.N., 2014. Regulation of microRNA biogenesis. *Nature Reviews Molecular Cell Biology* 15, 509–524.
- Hadjiargyrou, M.I., Delias, N., 2013. The intertwining of transposable elements and non-coding RNAs. *International Journal of Molecular Sciences* 14, 13307–13328.
- Han, Y., Liu, L., Fang, N., Yang, R., Zhou, D., 2013. Regulation of pathogenicity by noncoding RNAs in bacteria. *Future Microbiology* 8, 579–591.
- Haselmann, V., Kurz, A., Bertsch, U., *et al.*, 2014. Nuclear death receptor TRAIL-R2 inhibits maturation of let-7 and promotes proliferation of pancreatic and other tumor cells. *Gastroenterology* 146, 278–290.
- He, L., Thomson, J.M., Hemann, M.T., *et al.*, 2005. A microRNA polycistron as a potential human oncogene. *Nature* 435, 828–833.
- Herbig, A., Nieselt, K., 2011. nocoRNAc: Characterization of non-coding RNAs in prokaryotes. *BMC Bioinformatics* 12, 40.
- Hsu, S.D., Lin, F.M., Wu, W.Y., *et al.*, 2011. miRTarBase: A database curates experimentally validated microRNA-target interactions. *Nucleic Acids Research* 39, D163–D169.
- Huang, R., Rofstad, E.K., 2017. Cancer stem cells (CSCs), cervical CSCs and targeted therapies. *Oncotarget* 8, 35351–35367.
- Ji, Q., Hao, X., Zhang, M., *et al.*, 2009. MicroRNA miR-34 inhibits human pancreatic cancer tumor-initiating cells. *PLOS ONE* 4, e6816.
- Johnsson, P., Lipovich, L., Grandér, D., Morris, K.V., 2014. Evolutionary conservation of long noncoding RNAs; sequence, structure, function. *Biochimica et Biophysica Acta* 1840, 1063–1071.
- Junge, A., Refsgaard, J.C., Garde, C., *et al.*, 2017. RAIN: RNA–protein association and interaction networks. Database (Oxford), 2017. p. baw167.
- Kaczmarek, J.C., Kowalski, P.S., Anderson, D.G., 2017. Advances in the delivery of RNA therapeutics: From concept to clinical reality. *Genome Medicine* 27 (1). 60. 9.
- Kapusta, A., Kronenberg, Z., Lynch, V.J., *et al.*, 2013. Transposable elements are major contributors to the origin, diversification, and regulation of vertebrate long noncoding RNAs. *PLoS Genetics* 9 (4).

- Kargul, J., Laurent, G.J., 2011. Liver growth, development, and disease – New research revealing new horizons. *The International Journal of Biochemistry & Cell Biology* 43, 171.
- Kim, V.N., 2005. MicroRNA biogenesis: Coordinated cropping and dicing. *Nature Reviews Molecular Cell Biology* 6, 376–385.
- Kim, V.N., Nam, J.W., 2006. Genomics of microRNA. *Trends in Genetics* 22, 165–173.
- Kung, J.T.Y., Colognori, D., Lee, J.T., 2013. Long noncoding RNAs: Past, present, and future. *Genetics* 193, 651–669.
- Lauressergues, D., Couzigou, J.M., Clemente, H.S., *et al.*, 2015. Primary transcripts of microRNAs encode regulatory peptides. *Nature* 520, 90–93.
- Lewis, B.A., Walla, R.R., Terrililini, M., *et al.*, 2011. PRIDB: A protein-RNA interface database. *Nucleic Acids Research* 39, D277–D282.
- Li, S.C., Shiau, C.K., Lin, W.C., 2008. Vir-Mir db: Prediction of viral microRNA candidate hairpins. *Nucleic Acids Research* 36, D184–D189.
- Lim, L.P., Glasner, M.E., Yekta, S., Burge, C.B., Bartel, D.P., 2003. Vertebrate microRNA genes. *Science* 299, 1540.
- Liu, C.H., Wu, D., Pollock, J.D., 2012. Bioinformatic challenges of big data in non-coding RNA research. *Frontiers in Genetics* 3, 178.
- Liz, J., Esteller, M., 2016. lncRNAs and microRNAs with a role in cancer development. *Biochimica et Biophysica Acta* 1859, 169–176.
- Lu, C., Tej, S.S., Luo, S., *et al.*, 2005. Elucidation of the small RNA component of the transcriptome. *Science* 309, 1567–1569.
- Lu, Q., Ren, S., Lu, M., *et al.*, 2013. Computational prediction of associations between long non-coding RNAs and proteins. *BMC Genomics* 14, 651.
- Lu, Y.C., Smielewska, M., Palakodeti, D., *et al.*, 2009. Deep sequencing identifies new and regulated microRNAs in *Schmidtea mediterranea*. *RNA* 15, 1483–1491.
- Lujambio, A., Calin, G.A., Villanueva, A., *et al.*, 2008. A microRNA DNA methylation signature for human cancer metastasis. *Proceedings of the National Academy of Sciences of the United States of America* 105, 13556–13561.
- Lujambio, A., Ropero, S., Ballestar, E., Fraga, M.F., *et al.*, 2007. Genetic unmasking of an epigenetically silenced microRNA in human cancer cells. *Cancer Research* 67, 1424–1429.
- Lung, B., Zemmann, A., Madej, M.J., *et al.*, 2006. Identification of small non-coding RNAs from mitochondria and chloroplasts. *Nucleic Acids Research* 34, 3842–3852.
- Lv, S., Pan, L., Wang, G., 2016. Commentary: Primary transcripts of microRNAs encode regulatory peptides. *Frontiers in Plant Science* 7, 1436.
- Ma, C., Huang, T., Ding, Y.C., *et al.*, 2015. MicroRNA-200c overexpression inhibits chemoresistance, invasion and colony formation of human pancreatic cancer stem cells. *International Journal of Clinical and Experimental Pathology* 8, 6533–6539.
- Ma, L., Bajic, V.B., Zhang, Z., *et al.*, 2013. On the classification of long non-coding RNAs. *RNA Biol.* 10 (6), 924–933.
- Ma, L., Li, A., Zou, D., Xu, X., *et al.*, 2015. LncRNAWiki: Harnessing community knowledge in collaborative curation of human long non-coding RNAs. *Nucleic Acids Research* 43, D187–D192.
- Meister, G., Tuschl, T., 2004. Mechanisms of gene silencing by double-stranded RNA. *Nature* 431, 343–349.
- Mello, C.C., Conte, D., 2004. Revealing the world of RNA interference. *Nature* 431, 338–342.
- Melo, C.A., Melo, S.A., 2014. MicroRNA biogenesis: Dicing assay *RNA mapping*. *Methods in Molecular Biology* 1182, 219–226.
- Mendes, N.D., Freitas, A.T., Sagot, M.F., 2009. Current tools for the identification of miRNA genes and their targets. *Nucleic Acids Research* 37, 2419–2433.
- Novikova, I.V., Hennelly, S.P., Tung, C.S., Sanbonmatsu, K.Y., 2013. Rise of the RNA machines: Exploring the structure of long non-coding RNAs. *Journal of Molecular Biology* 425, 3731–3746.
- Ozsolak, F., Poling, L.L., Wang, Z., *et al.*, 2008. Chromatin structure analyses identify miRNA promoters. *Genes & Development* 22, 3172–3183.
- Pan, W., Zhou, L., Ge, M., *et al.*, 2016a. Whole exome sequencing identifies lncRNA GAS8-AS1 and LPAR4 as novel papillary thyroid carcinoma driver alterations. *Human Molecular Genetics* 25, 1875–1884.
- Pan, X., Fan, Y.X., Yan, J., Shen, H.B., 2016b. IPMiner: Hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction. *BMC Genomics* 17, 582.
- Paralkar, V.R., Weiss, M.J., 2013. Long noncoding RNAs in biology and hematopoiesis. *Blood* 121, 4842–4846.
- Parasramka, M.A., Maji, S., Matsuda, A., Yan, I.K., Patel, T., 2016. Long non-coding RNAs as novel targets for therapy in hepatocellular carcinoma. *Pharmacology & Therapeutics* 161, 67–78.
- Pritchard, C.C., Cheng, H.H., Tewari, M., 2012. MicroRNA profiling: Approaches and considerations. *Nature Reviews Genetics* 13, 358–369.
- Prokopi, M., Kousparou, C.A., Epenetos, A.A., 2015. The secret role of microRNAs in cancer stem cell development and potential therapy: A notch-pathway approach. *Frontiers in Oncology* 4, 389.
- Rackham, O., Shearwood, A.M., Mercer, T.R., *et al.*, 2011. Long noncoding RNAs are generated from the mitochondrial genome and regulated by nuclear-encoded proteins. *RNA* 12, 2085–2093.
- Rashid, F., Shah, A., Shan, G., 2016. Long non-coding RNAs in the cytoplasm. *Genomics Proteomics Bioinformatics* 14, 73–80.
- Sanchez, Y., Huarte, M., 2013. Long non-coding RNAs: Challenges for diagnosis and therapies. *Nucleic Acid Therapeutics* 23, 15–20.
- Scott, G.K., Berger, C.E., Benz, S.C., Benz, C.C., 2006. Rapid alteration of microRNA levels by histone deacetylase inhibition. *Cancer Research* 66, 1277–1281.
- Shah, M.Y., Ferrajoli, A., Sood, A.K., Lopez-Berestein, G., Calin, G.A., 2016. MicroRNA therapeutics in cancer – An emerging concept. *EBioMedicine* 12, 34–42.
- Shidhi, P.R., Suravajhala, P., Nayeema, A., *et al.*, 2015. Making novel proteins from pseudogenes. *Bioinformatics* 31 (1), 33–39.
- Shulman-Peleg, A., Nussinov, R., Wolfson, H.J., 2009. RsiteDB: A database of protein binding pockets that interact with RNA nucleotide bases. *Nucleic Acids Research* 37, D369–D373.
- Singh, T.R., Gupta, A., Suravajhala, P., 2013. Challenges in the miRNA research. *International Journal of Bioinformatics Research and Applications* 9, 576–583.
- Suravajhala, P., Kogelmeide, L.J.A., Mazzoni, G., Kadarmideen, H.N., 2015. Potential role of lncRNA cyp2c91–protein interactions on diseases of the immune system. *Frontiers in Genetics* 6, 255.
- Suresh, V., Liu, L., Adjero, D., Zhou, X., 2015. RPI-Pred: Predicting ncRNA-protein interaction using sequence and structural information. *Nucleic Acids Research* 43, 1370–1379.
- Thomson, D.W., Dinger, M.E., 2016. Endogenous microRNA sponges: Evidence and controversy. *Nature Reviews Genetics* 17, 272–283.
- Tomari, Y., Zamore, P.D., 2005. Perspective: Machines for RNAi. *Genes & Development* 19, 517–529.
- Tsai, M.C., Spitale, R.C., Chang, H.Y., 2011. Long intergenic noncoding RNAs: New links in cancer progression. *Cancer Research* 71, 3–7.
- Wang, K.C., Chang, H.Y., 2011. Molecular mechanisms of long noncoding RNAs. *Molecular Cell* 43, 904–914.
- Wang, X., Xuan, Z., Zhao, X., Li, Y., Zhang, M.Q., 2009. High-resolution human core-promoter prediction with CoreBoost_HM. *Genome Research* 19, 266–275.
- Wapinski, O., Chang, H.Y., 2011. Long noncoding RNAs and human disease. *Trends Cell in Biology* 21, 354–361.
- Weikard, R., Hadlich, F., Kuehn, C., 2013. Identification of novel transcripts and noncoding RNAs in bovine skin by deep next generation sequencing. *BMC Genomics* 14, 789.
- Wong, N., Wang, X., 2015. miRDB: An online resource for microRNA target prediction and functional annotations. *Nucleic Acids Research* 43, D146–D152.
- Wu, T., Jie Wang, J., Liu, C., *et al.*, 2006. NPInter: The noncoding RNAs and protein related biomacromolecules interaction database. *Nucleic Acids Research* 34, D150–D152.
- Xi, X., Li, T., Huang, Y., *et al.*, 2017. RNA biomarkers: Frontier of precision medicine for cancer. *Non-Coding RNA* 3, 9.
- Xie, X., Lu, J., Kulbokas, E.J., *et al.*, 2005. Systematic discovery of regulatory motifs in human promoters and 3′ UTRs by comparison of several mammals. *Nature* 434, 338–345.
- Yan, K., Arfat, Y., Li, D., *et al.*, 2016. Structure prediction: New insights into decrypting long noncoding RNAs. *International Journal of Molecular Sciences* 17, 132.
- Yanaihara, N., Caplen, N., Bowman, E., *et al.*, 2006. Unique microRNA molecular profiles in lung cancer diagnosis and prognosis. *Cancer Cell* 9, 189–198.
- Yao, S., 2016. MicroRNA biogenesis and their functions in regulating stem cell potency and differentiation. *Biological Procedures Online* 18, 8.

Yigit, E., Batista, P.J., Bei, Y., *et al.*, 2006. Analysis of the *C. elegans* Argonaute family reveals that distinct Argonautes act sequentially during RNAi. *Cell* 127, 747–757.
Zhang, X., Wu, D., Chen, L., *et al.*, 2014. RAID: A comprehensive resource for human RNA-associated (RNA–RNA/RNA–protein) interaction. *RNA* 20, 989–993.
Zhao, Y., Li, H., Fang, S., *et al.*, 2016. NONCODE 2016: An informative and valuable data source of long non-coding RNAs. *Nucleic Acids Research* 44, D203–D208.

Biographical Sketch



Pradeep Tiwari is a PhD fellow in the Department of Chemistry at School of Basic Sciences, Manipal University Jaipur. He is working for his PhD on Systems Genomics Approach for understanding noncoding RNAs in Asian Indian Type 2 Diabetes Mellitus (AITDM).



Sonal Gupta is a PhD fellow in Department of Biotechnology at Amity University, Jaipur. She is working for her PhD on long noncoding RNAs (lncRNA) and proteins specific to congenital pouch colon.



Anuj Kumar is a Bioinformatician at Advance Center for Computational and Applied Biotechnology, Uttarakhand Council for Biotechnology (UCB), Silk Park, Prem Nagar, Dehradun.

Mansi Sharma is a Senior Research fellow at Uttarakhand Council for Biotechnology (UCB), Silk Park, Prem Nagar, Dehradun.



Dr. Vijayaraghava Seshadri Sundararajan is a Scientist at Environmental Health Institute, National Environment Agency, Singapore.



Professor S.L. Kothari is the Director, Amity Institute of Biotechnology, Amity University Rajasthan, Jaipur.



Dr. Sandeep Kumar Mathur is a Senior Medical Professor and Head, Department of Endocrinology, S.M.S. Medical college and hospital, Jaipur.



Dr. Krishna Mohan Medicherla is the head of Research and Development at Birla Institute of Scientific Research, Jaipur.



Dr. Babita Malik is the Professor and Head, Department of Chemistry, School of Basic Sciences, Manipal University, Jaipur.



Dr. Prashanth Suravajhala is a Research Scientist in Systems Biology at Department of Biotechnology and Bioinformatics, Birla Institute of Scientific Research, Jaipur.

Identification and Extraction of Biomarker Information

Vijayaraghava S Sundararajan, Ministry of Environmental Agency, Singapore and Bioclues.org, Hyderabad, India
Shweta Shrotriya, Monisha Singhal, Pragya Chaturvedi, and Prashanth Suravajhala, Birla Institute of Scientific Research, Jaipur, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

The development of biomarker discovery has revolutionized the field of disease diagnosis and prognosis. Bioinformatics based approaches for biomarker identification are growing with the advances in genome sequencing resulting in a huge amount of sequences. Various bioinformatics based techniques, such as machine learning, network-based approach, and text-mining for identification and characterization of biomarkers are widely in use (Cohen and Hersh, 2005). As per the World Health Organization, biomarkers reflect an interaction between a biological system and a potential hazard, which may be chemical, physical, or biological (WHO International Programme on Chemical Safety Biomarkers and Risk Assessment: Concepts and Principles, 1993). While biomarkers can help in the care of patients with disease or no disease, on the basis of this clinical perspective, biomarkers are classified into prognostic, diagnostic, predictive and therapeutic (Carlomagno *et al.*, 2017). Prediction of disease is not the only application of biomarkers but it helps in monitoring toxicity for the development and discovery of new drugs. With bioinformatics applications important in filling this gap between bioanalytics and initial discovery phase, semantic searches and biomedical literature by and large have allowed us to efficiently identify and characterize biomarkers (Bundschus *et al.*, 2008). The main focus of this review is to provide information on extraction and identification of biomarker information.

Exploring Contextual Genes

A complex cellular system of higher organisms consists of huge number of processes running simultaneously. Maintaining a particular cellular state is one of the major tasks regulated synchronously in a cell. Transition of cellular state from one state to another requires the alterations in regulatory pathways as well, which consists of a large number of genes functioning to get a particular state. Such genes with functional relatedness are called contextual genes (Ferrer *et al.*, 2010). Genome context methods were used to predict this functional relatedness between genes using the patterns of existence and relative locations of the homologs. While annotating functional and structural information of genes in newly sequenced genomes, homology is used a reliable tool; we also explore contextual genes in targeted genomes to find out their functional relatedness. Widely used context methods in literature are phylogenetic profiles, gene neighbor, gene cluster, and gene fusion (Fig. 1) (Ferrer *et al.*, 2010). The important criteria for these methods are presence of homologous sequences for the genes in the target genome where in this case the degree of sequence similarity is checked. The contextual gene methods are classified by and large into (1) full coverage and (2) restricted coverage with the former generating scores for all possible gene pairs from a genome, while the other essentially generates scores only for some pairs. One widely employed full coverage method is the phylogenetic profile (Fig. 1) and gene neighbor methods (Fig. 1), which are used to compare sequences based on the degree of full coverage, whereas the gene cluster and the gene fusion are a part of restricted coverage methods (Fig. 1). However, the contexts used to infer functional relationships between genes with or without sequence similarity based on evolutionary relationships can be biased in some cases (Krawczuk and Łukaszuk, 2016). To check this, crude statistics and normalization are done to enhance the performance and genome dependence (Lehmann and Romano, 2005). In addition, combining scores from two different methods, machine learning and classification with parameter selection and tuning, would help achieve better results. Among these methods, the gene neighbor methods are known to outperform the phylogenetic profile method by as much as 40% in sensitivity when compared with the gene cluster method at low sensitivities.

Methods Used for Biomarker Identification

Identifying biomarkers is one of the greatest challenges as different genotype–phenotype correlation from large-scale biological data is assessed. There are a good number of text mining, knowledge-based and network-based methods to discover biomarkers from literature (Hall and Holmes, 2003). These can be better made by constructing a resource based on indexed text or dictionary used from a finite state machine. One of the key components of identifying contextual genes is condition-specific interactions in biological networks. Valuable mechanistic insights can be derived from functional analyses of genomic data. A list of commonly used data mining methods is given below.

- Unpaired null hypothesis testing: This is a univariate filter method where the calculated probability (P -value) serves as the evaluation measure to rate variables on their discriminatory ability (Lehmann and Romano, 2005).
- Principal component analysis: This technique is an unsupervised projection method, which employs linear combinations of variables based on the variance of the original data (Ringnér, 2008).

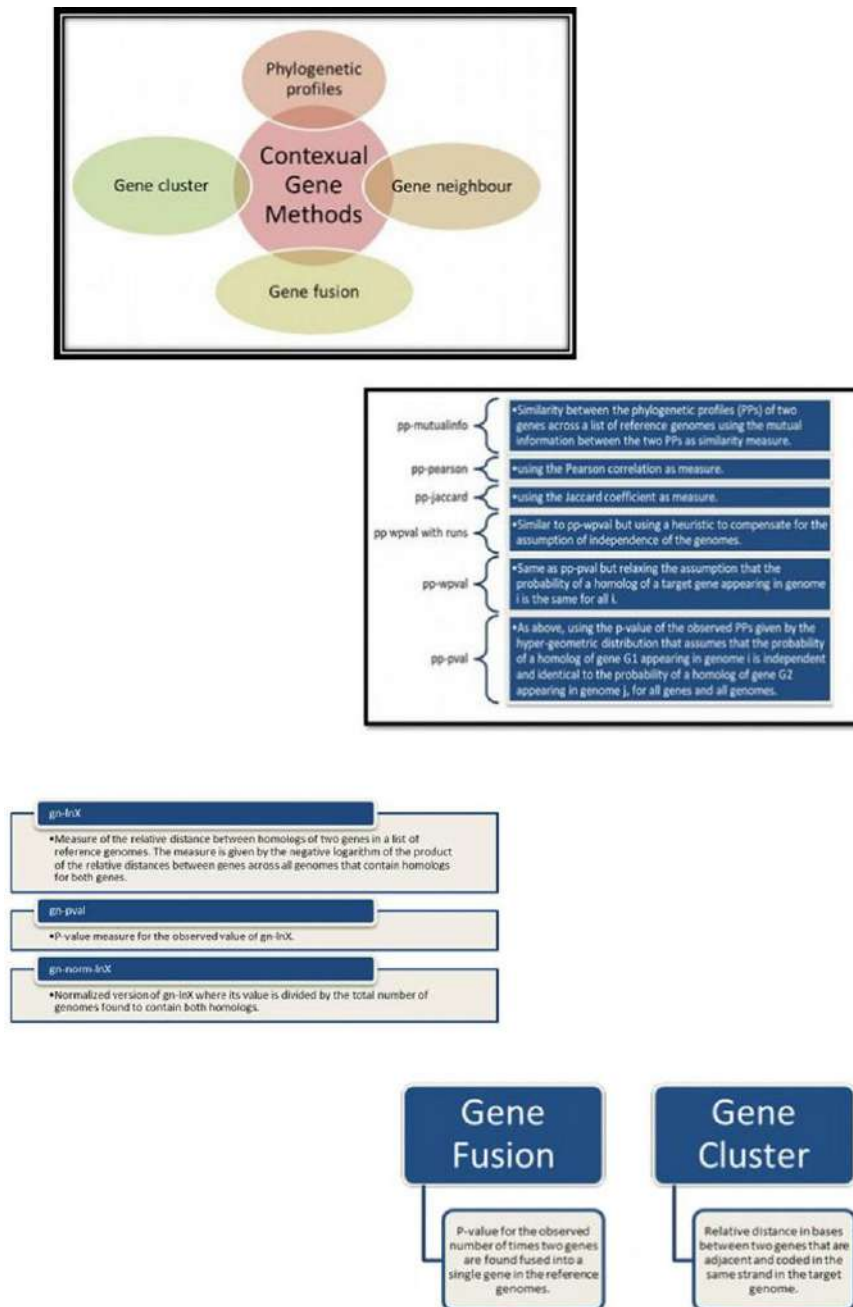


Fig. 1 Widely used context methods. (a) Contextual gene methods and their classification, (b) an overview of phylogenetic profiles methods, (c) gene neighbor methods and (d) gene fusion and cluster methods.

- Information gain: Is another univariate filter method, which evaluates entropy reduction of data due to feature separation (Hall and Holmes, 2003).
- ReliefF (RF): This is a multivariate filtration method, in which the RF score relies on the concept that values of a significant feature are correlated with the feature values of an instance of the same class and uncorrelated with the feature values of an instance of the other class (Robnik-Sikonja, 2016).
- Associative voting: This is another multivariate filter method and uses a rule-based evaluation criterion by a special form of association rules; considers interaction among features (Osl et al., 2008).
- Unpaired biomarker identifier: Is a univariate filter method, in which a statistical score can be determined by combining a discriminant measure with a biological effect term appropriate for binary classifications alone (Baumgartner et al., 2010).
- Support vector machine recursive feature elimination is an embedded selection method that optimizes the weights of SVM classifiers to rank features (Guyon et al., 2002).

- Random forest models is an embedded selection method, using bagging and random subspace methods to construct a collection of decision trees to identify a complete set of significant features (Enot *et al.*, 2006).
- Aggregating feature selection is an ensemble selection method that aggregates multiple feature selection results to a consensus ranking (Saeys *et al.*, 2008).
- Stacked feature ranking is an ensemble selection method that has stacked learning architecture to construct a consensus feature ranking by combining multiple feature selection methods (Netzer *et al.*, 2009).
- Wrapper approach evaluates the merit of a feature subset by accuracy estimates using a classifier and produces a subset of very few features that are dominated by stronger and uncorrelated attributes (Hall and Holmes, 2003). There are a few guilt-by-association studies to check these features (Shin *et al.*, 2008).
- Paired null hypothesis testing is a univariate filter method where *P*-value serves as evaluation measure for the discriminatory ability of variables (Lehmann and Romano, 2005).
- Paired biomarker identifier is an univariate filter method that uses a statistical evaluation score by combining biological effects (Baumgartner *et al.*, 2010) (Fig. 2).

Other Feature Selection Methods

Next-generation sequencing has given us tremendous impetus in identifying the biomarkers and with the help of feature selection/classification methods, there arose an interest to discern them using deep learning approaches (Lee *et al.*, 2018). Several combinatorial based methods are known for identification of biomarkers, for example heuristic searches, hybrid methods, and exhaustive methods, which have been reviewed (Wang *et al.*, 2016). There is still apaucity of resources employing these features. For example, a database of mutation-gene-drug relations for such classification features could prove to be a valuable resource for researchers interested in personalized medicine and genealogy (Wu *et al.*, 2014). For example, Wu *et al.* (2014) have developed a large-scale text-mining system to generate molecular profiles of thyroid cancer using literature based classification scoring schemes. In this process, they are able to identify key genes and pathways for easy prioritizing diagnostic biomarkers for therapeutic use. Similarly, Li *et al.* (2014) prioritized cancer-related miRNAs using network-based approaches by employing the random-walk algorithm to elucidate miRNA-disease networks. In the recent past knowledge-driven text-mining approaches have shown tremendous applications for extracting biomarker information covering all therapeutic areas (Bravo *et al.*, 2014). These are preferentially used by taking Medline publications containing the MeSH terms and other bibliometric analysis. Automated extraction of biomarkers using semantic graphs has also been attempted recently (Vlietstra *et al.*, 2017). Within the last few years, systems biology approaches clubbed with NGS became more amenable in extracting the useful information (Mayer *et al.*, 2012), biomarker identification and computationally expensive and challenging task methods such as knowledge graphs prove to be very useful in bringing exponential growth of biomedical literature and databases (Bravo *et al.*, 2014).

Structured knowledge is better represented in the form of a knowledge graph where unique relationships are defined using the Unified Medical Language System (UMLS), which integrates relationships extracted from Medline abstracts (Bodenreider, 2004). The knowledge graph runs on a 1.8.3 Neo4j graph database, the 3,527,423 biomedical concepts are represented as vertices, with 68,413,238 relationships between them. The individual concepts represent units of thought, which are atomic and unique. Two concepts can be connected to each other with one or more semantic relationships (also referred to as predicates), such as “causes” or “inhibits,” thereby forming a triple. In addition, ranking and filtering methods are applied by checking recall, precision, and cumulative gains. The list of potential biomarker compounds generated by the method could be retrieved by the aforementioned processes and later assigned to the error categories. This will also allow us to recognize and apply normalization methods for the disease and biomarker entities in biomedical publications by means of the biomedical named entity recognition system. A gene dictionary constituting a huge collection of terms referring to human genes and proteins is integrated with data from three biological databases. This is complemented by a disease dictionary with UMLS metathesaurus (UMLS Methathesaurus, 2018), a large, multipurpose, and multilingual thesaurus that contains millions of biomedical and health-related concepts, their synonymous names, and their known relationships. Both dictionaries are curated and extended semiautomatically using different rules to facilitate the matching task.

Methodology focused on the extraction of disease–biomarker associations reported in the literature is not just limited to the methods as mentioned above but a knowledge-driven approach takes advantage of the annotation of MEDLINE publications pertaining to biomarkers with MeSH terms, narrowing the search for specific publications and therefore minimizing the false positive ratio. The application of this methodology is shown using a neural network-based biomarker association extraction approach for cancer classification (Bravo *et al.*, 2014; Kim *et al.*, 2007). These are complemented by a linear biomarker association network (LN), a fully connected neural network algorithm that assumes that the expression level of a biomarker is associated with some other biomarkers and can be estimated using a linear combination. This is briefly described below.

Consider a set of p biomarkers. The input of the LN represents the observed levels of the p biomarkers, denoted as $x = [x_1, x_2, \dots, x_p]^T$, and the output represents the corresponding estimated results, denoted as $y = [y_1, y_2, \dots, y_p]^T$. Let $A = \{a_{ij} \in \mathbb{R}, a_{ii} = 0; i, j = 1, 2, \dots, p\}$ denotes the connection matrix of the network, the operation of the LN can be represented as follows:

$$y = Ax + B, \quad B = [b_1, b_2, \dots, b_p]^T$$

where $b_i, i = 1, 2, \dots, p$, represents the expression baseline of the i th biomarker.

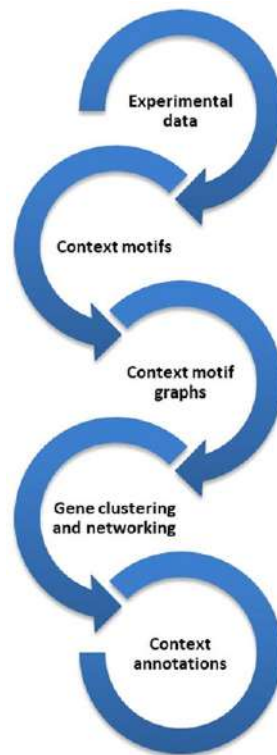


Fig. 2 An overview of context mining process flow. The experimental data is measured from contextual information, before annotations are drawn from various methods.

The classification features using conventional methods include Fisher discriminant analysis, k -nearest neighborhood, Bayesian network classifier (BN), and support vector machines with both linear kernels (linear-SVM) and radial basis function kernels (rbf-SVM). Among these conventional methods, the BN classifier, linear-SVM, and RBF-SVM require their parameters to be tuned for best performance. Wang *et al.* (2018) have developed a new cancer classification approach based on the concept of a biomarker association network (BAN) and a neural network structure to model the biomarker association patterns responsible for cancer. Using publicly available datasets it has been shown that the BAN-based classification approaches, in particular, the NLN classifier, achieve excellent classification performances on the four datasets (Wang *et al.*, 2018). On the other hand, extracting semantic biomedical relations with a sequence labeling approach based on conditional random fields is studied with general free text searchers containing concise phrases. However, these are bound by limitations because an effective search using gene cards and other secondary repesitive database searchers are necessitated to study them (Wang *et al.*, 2009).

Conclusions

We have presented an overview of extraction and identification of biomarker information using different methods, primarily based on feature selection, knowledge-derived methods, and graphs. Identification of biomarkers not only serves as a distinct identity for understanding molecular, physiological, and structural variations but also helps disseminate a contextual flow for future methods in accurately classifying the biomarkers. An integrated embedded approach to identify biomarkers through graph theories, biological networking, and modeled through learning algorithms would be of much use to the research community. There is a vivid hope that classification feature selection methods through machine learning approaches would bring identification of the biomarkers more accurately.

See also: Biological Database Searching. Natural Language Processing Approaches in Bioinformatics

References

- Baumgartner, C., Lewis, G.D., Netzer, M., Pfeifer, B., Gerszten, R.E., 2010. A new data mining approach for profiling and categorizing kinetic patterns of metabolic biomarkers after myocardial injury. *Bioinformatics* 26, 1745–1751.

- Bodenreider, O., 2004. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research* 32, 267D–270D.
- Bravo, À., Cases, M., Queralt-Rosinach, N., Sanz, F., Furlong, A., 2014. Knowledge-driven approach to extract disease-related biomarkers from the literature. *BioMed Research International* vol. Article ID 253128.
- Bundschuh, M., Dejori, M., Stetter, M., Tresp, V., Kriegl, H.P., 2008. Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*. 9.207-10.1186/1471-2105-9-207.
- Carlomagno, N., Incollingo, P., Tammaro, V., *et al.*, 2017. Diagnostic, predictive, prognostic, and therapeutic molecular biomarkers in third millennium: A breakthrough in gastric cancer. *BioMed Research International* 2017, 7869802.
- Cohen, A.M., Hersh, W.R., 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics* 6 (1), 57–71.
- Enot, D.P., Beckmann, M., Overy, D., Draper, J., 2006. Predicting interpretability of metabolome models based on behavior, putative identity, and biological relevance of explanatory signals. *Proceedings of the National Academy of Sciences of the United States of America* 103, 14865–14870.
- Ferrer, L., Dale, J.M., Karp, P.D., 2010. A systematic study of genome context methods: Calibration, normalization and combination. *BMC Bioinformatics*. 11, 493.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422.
- Hall, M.A., Holmes, G., 2003. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering* 15, 1437–1447.
- Kim, S., Sen, I., Bittner, M., 2007. Mining molecular contexts of cancer via in-silico conditioning. In: *Proceedings of the IEEE Computational Systems Bioinformatics Conference*, vol. 6, pp. 169–179.
- Krawczuk, J., Łukaszyk, T., 2016. The feature selection bias problem in relation to high-dimensional gene data. *Artificial Intelligence in Medicine* 66, 63–71.
- Lee, K., Kim, B., Choi, Y., *et al.*, 2018. Deep learning of mutation-gene-drug relations from the literature. *BMC Bioinformatics*. 19 (1), 21.
- Lehmann, E.L., Romano, J.P., 2005. *Testing Statistical Hypotheses*, third ed. New York, NY: Springer Verlag.
- Li, L., Hu, X., Yang, Z., *et al.*, 2014. Establishing reliable miRNA-cancer association network based on text-mining method. *Computational and Mathematical Methods in Medicine*. 746979.
- Mayer, P., Mayer, B., Mayer, G., 2012. Systems biology: Building a useful model from multiple markers and profiles. *Nephrology Dialysis Transplantation* 27 (11), 3995–4002.
- Netzer, M., Millionig, G., Osl, M., *et al.*, 2009. A new ensemble-based algorithm for identifying breath gas marker candidates in liver disease using ion molecule reaction mass spectrometry. *Bioinformatics* 25, 941–947.
- Osl, M., Dreiseitl, S., Pfeifer, B., *et al.*, 2008. A new rule-based data mining algorithm for identifying metabolic markers in prostate cancer using tandem mass spectrometry. *Bioinformatics* 24, 2908–2914.
- Ringnér, M., 2008. What is principal component analysis? *Nature Biotechnology* 26, 303–304.
- Robnik-Sikonja, M., 2016. Data generators for learning systems based on RBF networks. *IEEE Transactions on Neural Networks and Learning Systems* 27 (5), 926–938.
- Saeys Y., Abeel T., Van de Peer Y., 2008. Robust feature selection using ensemble feature selection techniques. In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, vol. 5212, pp. 313–325. Part II Berlin, Heidelberg: Springer-Verlag. (Lecture Notes in Artificial Intelligence).
- Shin, H., Sheu, B., Joseph, M., Markey, M.K., 2008. Guilt-by-association feature selection: Identifying biomarkers from proteomic profiles. *Journal of Biomedical Informatics* 41, 124–136.
- UMLS Metathesaurus: https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/ last accessed: April 28, 2018.
- Vlietstra, W.J., Zielman, R., van Dongen, R.M., *et al.*, 2017. Automated extraction of potential migraine biomarkers using a semantic graph. *Journal of Biomedical Informatics* 71, 178–189.
- Wang, L., Wang, Y., Chang, Q., 2016. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods* 111, 21–31.
- Wang, H.Q., Wong, H.S., Zhu, H., Yip, T.T., 2009. A neural network-based biomarker association information extraction approach for cancer classification. *Journal of Biomedical Informatics* 42, 654–666.
- Wang, X., Yang, C., Guan, R., 2018. A comparative study for biomedical named entity recognition. *International Journal of Machine Learning and Cybernetics* 9 (3), 373–382.
- WHO International Programme on Chemical Safety Biomarkers and Risk Assessment: Concepts and Principles. 1993. Retrieved from: <http://www.inchem.org/documents/ehc/ehc/ehc155.htm>. (accessed: 04.28.2018).
- Wu, C., Schwartz, J.M., Brabant, G., Nenadic, G., 2014. Molecular profiling of thyroid cancer subtypes using large-scale text mining. *BMC Medical Genomics* 7 (Suppl. 3), S3.

Biographical Sketch



Dr. Vijayaraghava Seshadri Sundararajan is on the board of advisers for Bioclues, a not-for-profit virtual organization in India.



Shweta Shrotriya is a Research fellow at Department of Biotechnology and Bioinformatics, Birla Institute of Scientific Research, Jaipur.



Monisha Singhal is a Research fellow at Department of Biotechnology and Bioinformatics, Birla Institute of Scientific Research, Jaipur.



Pragya Chaturvedi is pursuing a doctorate and working as Research fellow at Department of Biotechnology and Bioinformatics, Birla Institute of Scientific Research, Jaipur.



Dr. Prashanth Suravajhala is a Research Scientist in Systems Biology at Department of Biotechnology and Bioinformatics, Birla Institute of Scientific Research, Jaipur.

Computational Systems Biology Applications

Ayako Yachie-Kinoshita, Sucheendra K Palaniappan, and Samik Ghosh, The Systems Biology Institute, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Since its emergence in the mid-1990s, Systems Biology has laid the foundations for developing systems level understanding of living systems beyond the paradigms of single gene/protein/molecular view of biology (Kitano, 2002a,b; Ideker *et al.*, 2001). With the rapid development of large scale omics experimental systems in tandem with computational tools for molecular pathway visualization, modeling, data management and analysis, the field has contributed to deeper understanding of fundamental mechanisms of biological processes as well as their applications in various branches of medicine and biotechnology (Tyson *et al.*, 2001; Novak and Tyson, 1993; Chen *et al.*, 2004; Aoki *et al.*, 2011; Schoeberl *et al.*, 2010, 2009).

Particularly, modeling and simulation efforts have been applied in early approaches to study cell cycle dynamics (Tyson *et al.*, 2001; Novak and Tyson, 1993; Chen *et al.*, 2004) and in analysis of signaling pathways implicated in cancer, for example, the mitogen-activated protein kinase (MAPK) dynamics (Aoki *et al.*, 2011). Such modeling efforts have also found application in drug discovery and clinical trial designs as demonstrated by the development of MM-121 an ErbB3 antibody as an investigational drug candidate for various cancer sub-types (e.g., heregulin positive Non-small cell lung cancer) (Schoeberl *et al.*, 2010, 2009).

Further, approaches from systems biology have been integrated with traditional pharmacodynamics/pharmacokinetic (PKPD) and ADME-Tox (drug administration, distribution, metabolism and excretion and toxicology) modeling techniques to develop novel analysis pipelines in systems toxicology as reviewed in (Ghosh *et al.*, 2013b). Particularly, multiple spatio-temporal modeling efforts have been developed as part of the High Definition (HD) Physiology project in Japan and the Virtual Physiological Human (VPH) project in Europe (Ghosh *et al.*, 2013b). In recent years, systems approach to the study of chemical toxicology have gained increasing attention with the restriction on animal testing for toxicity studies in various industries like cosmetics and food. Concurrently, new analytical constructs, like the Adverse Outcome Pathways (AOP) have been mandated by the OECD which “...describe sequential chain of causally linked events at different levels of biological organization that lead to an adverse health or ecotoxicological effect” (see “Relevant Websites section”). Various *in silico* pipelines are been developed for multiscale modeling of safety assessments based on *in vitro* studies as well as toxicogenomics data on chemical compound screens (Bois *et al.*, 2017; Vinken, 2015).

While hypothesis-driven modeling approaches have yielded practical applications of systems biology as noted above, data-driven techniques are gaining traction in recent years (Webb, 2018). Wide-ranging applications of deep learning models have been developed in cheminformatics, translational biomarker and drug discovery and clinical applications (comprehensive reference available at (Webb, 2018)). Particularly, success of such deep architecture models has been demonstrated in genomics field including the DeepVariant model for variant calling, enhancer predictions, and population genomics (Poplin *et al.*, 2017). With the ability to generate omics scale data at various biological layers (transcriptomics, proteomics, metabolomics), novel analytics and data management are being developed for obtaining systems level insights from multi-omics data (Yi *et al.*, 2014; Shehzad *et al.*, 2014; Imam *et al.*, 2015; Schulz *et al.*, 2014; Yugi *et al.*, 2016). Specifically, *Trans-Omics* approach has been proposed such as in (Yugi *et al.*, 2016) to reconstruct biochemical and regulatory networks across multiple layers of data.

Such methods which bridge multiple dimensions of data and analysis/modeling approaches herald a new direction for systems biology. Specifically, with the advent of *big data* in biology together with the rapid pace of development of machine learning (ML)/artificial intelligence (AI), it is increasing evident that new horizons need to be explored to leverage systems level approaches on a broader scale spanning basic biology, drug discovery to translational and clinical applications and moving to patient reported phenotypic data, as shown schematically in Fig. 1.

In this article, to overview the new horizons in systems biology-guided approaches in biology and medicine, we identify the digital drivers shaping the current landscape and explore the need for next generation platforms to connect the drivers. Some case studies of connecting multi-dimensional data and analysis for different workflows are also discussed. We then conclude with perspectives on the vision driving next-generation research in biology and healthcare.

Digital Drivers at the Intersection of Technology and Medicine

This section outlines some of the digital drivers which are playing a pivotal role in the changing landscape of systems biology and its applications in biology and medicine. Highlighting the key areas as identified in (Hird *et al.*, 2016), the section captures impact of the digital drivers before presenting the role of potential new approaches in the next section.

Role of Data

With increasing volumes of data been generated in multiple dimensions, the role of data is becoming center stage in biology and medicine. High-throughput biotechnologies including DNA sequencing, transcriptomics, proteomics and metabolomics comprise

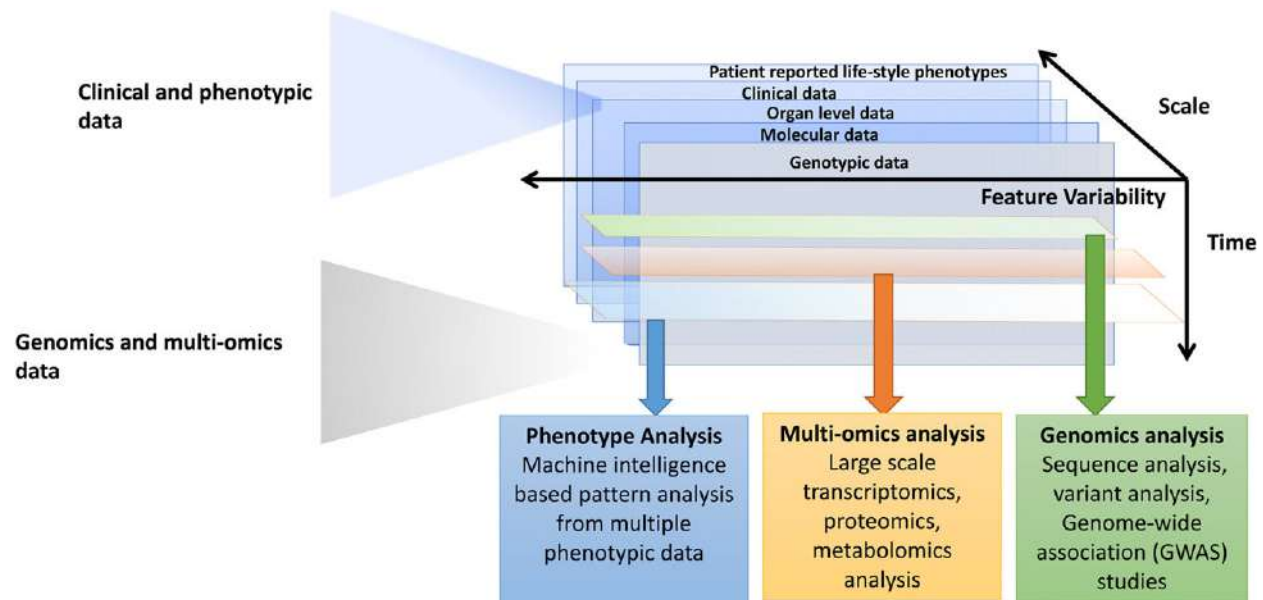


Fig. 1 New Horizons of data and analytics in biology and medicine. Multiple dimensions of data are being generated in the new era of medicine having distinct features- from genomics and molecular data to clinical and life-style phenotypes); such high dimensional (HD) data needs connection to novel HD analytics with different analytics techniques to obtain insights at different scales. Reproduced from Hird, N., Ghosh, S., Kitano, H., 2016. Digital health revolution: Perfect storm or perfect opportunity for pharmaceutical R&D? Drug Discovery Today. Available at: <http://doi.org/10.1016/j.drudis.2016.01.010>.

"Big data", characterized by the 4 'V's – data having high Volume, Variety, Velocity and Variability, are being generated in next generation experimental systems at the levels of single cells, tissues, organs to individuals. The most prevalent data at individual level is genotype, and genotype-phenotype correlations. A number of systems biology approaches have emerged to fill the gap between genotype and phenotype with the use of molecular big data such as protein-protein interaction network (Ben-Hamo and Efroni, 2013). On the other hand, a variety of "Small data" for each individual is being collected by multiple digital sources in hospital (Hood and Stephen, 2011) as well as home settings, such as personalized and time or event-dependent physiological parameters, social network and patterns in behavior and its geographical areas.

The integration of the multi-level, multi-scope and multi-scale data has given rise to the concepts of "extended phenotype" (Jain *et al.*, 2015). Such a paradigm shift in data in systems biology will enable to assess the open question how the phenotype is generated from the genotype, bringing the most important 'V' of Value to data.

Role of Devices

Another key driver in digital technologies is the rapid evolution in computers, sensors, and robotics, both in power and performance. Wearable devices with embedded sensors for measuring vital signs, to telemedicine devices and robotic automation of large scale laboratory experiments are now being developed. Moreover, together with the progress in Information and Communication Technologies (ICT) like the internet ubiquity, high network intensity and substantial data security, these hardware devices can be connected with smart devices such as wearable monitors, smart watches, cell phones or even household electrical appliances, generally defined under the umbrella term of Internet of Things (IoT). In the sense of extended phenotype, these "off the shelf" technologies bring about a transition from the world of a collection of scattered small data to a new future of integrated big data – enabling specialists from diverse domains, such as biologists and data analysts to access and analyze the data from different perspectives.

Role of Advanced Analytics and Machine Intelligence

The flood of data coming from multiple device source far exceeds the cognitive threshold of researchers. The development of advanced analytics techniques for such High-dimension (HD) data is ranging from inference techniques (Schadt *et al.*, 2005) to so-called Machine Intelligence (Chouard and Venema, 2015) including the methodologies of machine learning and artificial intelligence (AI). These techniques provide powerful predictions to identify patterns in the data and insights for deep phenotyping of the target systems. For example, the community challenges of applying machine learning techniques in inference of gene regulatory network (GRN) have established community-based methodologies as a powerful approach to predict GRN from accumulative transcriptome data (Marbach *et al.*, 2012). Genome-wide association studies (GWAS)-guided or targeted metabolomics technology-driven weighted co-expression network analysis (WGCNA) of HD transcriptome data have provided insights on

potential therapeutic strategy for this subset of breast cancer and the potential etiology of psychiatric disorders, respectively (O'Dushlaine *et al.*, 2015; Camarda *et al.*, 2016). Recent studies in healthcare have applied these techniques such as for disease biomarkers identification, drug discovery and trend analysis in disease spread (Ginsberg *et al.*, 2009; Libbrecht and Noble, 2015).

Role of Platforms

These key trends are opening up new challenges and opportunities in systems biology for solving systems complexity represented in data and approaches. The application of systems biology in domains such as systems drug design or personalized medicine requires sophisticated data-handling tools, modeling, integrated computational analysis and knowledge aggregation in a connected manner, as presented in Fig. 2. On the other hand, every workflow in systems biology must be specialized to the target system and the problem. In reality, researchers are collecting and connecting data and tools for each building block in each workflow in an ad-hoc manner. Also, as no single method/data universally applicable to all workflows, the choice of whether and how to utilize them will rely on individual researchers' ability and accessibility.

Under these circumstances, an integrated platform is required where researchers can find the right data and suitable tools and devices, build the problem-specific workflow, and execute the entire workflow. Several efforts have begun to develop such integrated platforms in the domain of systems biology, healthcare and beyond (Table 1). For example, Galaxy has involved several bioinformatics researchers to build a robust and reusable data analytic pipelines around next-generation sequencing (NGS) data (Goecks *et al.*, 2010). Genome-space (Qu *et al.*, 2016) involves another established analytic platform such as Gene Pattern (Reich *et al.*, 2006) and geWorkBench5 (Floratos *et al.*, 2010), which is also focusing on NGS data analytics. Another effort in this direction is the Garuda Platform (Ghosh *et al.*, 2011), a connectivity and automation platform which aims to provide a gateway of tools (called gadgets in Garuda) on different domains and dynamic construction of workflows (called Garuda recipes) by discovering and connecting different gadgets in a system agnostic manner (web-based applications and standalone software). Garuda is built as a light-weight programming language independent framework which allows seamless connection of software, devices in a social community network (called gadget graphs).

Role of Community – Human in the Loop Research

Finally, we identify and discuss on the critical role of as a key driver in digital technologies-driven systems biology and its applications. It is becoming increasingly apparent that for large-scale projects which involve massive amounts of data, diverse ICT needs and analytics techniques, cross and inter-disciplinary skills are the need of the future – it is impossible for a single researcher or even a single laboratory to harness all the different dimensions of a research project.

Community-driven data collection through data repositories has grown in its popularity and necessity, especially for the omics data (Perez-Riverol *et al.*, 2017), while the application of mass spectrometry analysis such as metabolomics and proteomics have certain difficulties to standardize the needs and usability with regards to the type of data and analysis (Kind *et al.*, 2009; Wang *et al.*, 2016). Apart from the data, notably, the concept of Wisdom of Crowd, where the collective knowledge of a community is greater than the knowledge of any individual, has become proven in both network inference of living organisms (Marbach *et al.*, 2012). Under these circumstances, consistency and the interoperability of various algorithms, tools and devices are the first step towards the goal.

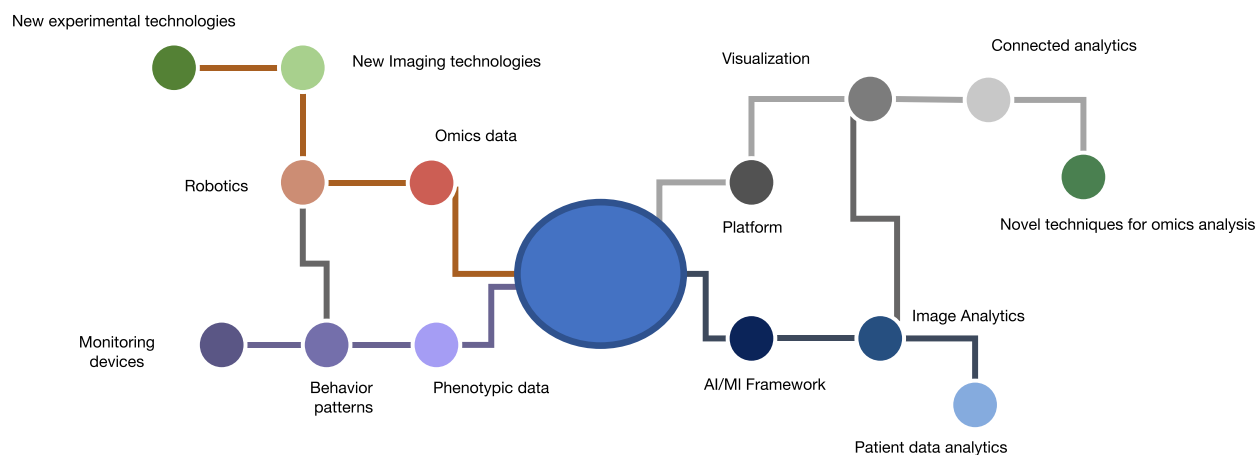


Fig. 2 Digital drivers at the intersection of technology and medicine. Various dimensions of big data, novel high-throughput experimental technologies, smart devices coupled with rapid development in ML/AI is driving the frontiers of computational systems biology and its application; such digital drivers highlight the need of next generation connected platforms to leverage the multi-dimensional drivers. Reproduced from Hird, N., Ghosh, S., Kitano, H., 2016. Digital health revolution: Perfect storm or perfect opportunity for pharmaceutical R&D? Drug Discovery Today. Available at: <http://doi.org/10.1016/j.drudis.2016.01.010>.

Table 1 Integrated platforms applicable in systems biology and beyond

Platform	Focused domain	Environment		Automation	Device connectivity	Availability	Development community	Web link
		Local	Cloud					
Galaxy	NGS analytics	○	○	○	×	Free	Open	https://galaxyproject.org/
Garuda	Data analytics	○	Δ[v2.0 or later]	Δ[v2.0 or later]	○	Free/ Commercial	Open	https://www.garuda-alliance.org/
Genomespace	NGS analytics	×	○	×	×	Free	Open	http://www.genomespace.org/
Knime	Workflow builder	○	○	○	×	Free	Open	https://www.knime.org/
Pipeline Pilot	Workflow builder	○	×	○	×	Commercial	Developer	http://accelrys.com/products/collaborative-science/biovia-pipeline-pilot/
Synapse	Data management	×	○	×	×	Free/ Commercial	Developer	https://www.synapse.org/
Taverna	Data analytics	Δ	○	○	×	Free	Open	http://www.taverna.org.uk/

Note: Adapted from Yachie, A., Matsuoka, Y., 2016. Systems drug and therapy design. *Igaku No Ayumi* 259 (8), 853–858 (in Japanese).

The standardization of the software, i.e., a common language for inter-operability, is the key to store and share information in a seamless and unambiguous fashion. One successful case is The Systems Biology Markup Language (SBML: see “Relevant Websites section”), which is set of standardization for mathematical modeling and simulation executable through various modeling and simulation software in systems biology. Importantly, the SBML community was evolved to structure themselves and keeps growing themselves to engage, reach out and educate the scientific community. Over 200 modeling and simulation software in biology domain comply with SBML enabling the community to share and reuse the models across them (Klipp *et al.*, 2007). Further, open source, open development software and environments have significantly moved the field forward, which is mostly based on specific programming language including R: Bioconductor (Huber *et al.*, 2015) and python (such as PySB; Systems biology modeling in Python (Lopez *et al.*, 2013)).

On the other hand, data, tools and platforms may not be enough to frame and motivate open-ended collaboration to solve the big problems in science. Bringing humans in the loop, empowered by advanced devices, data and analytics are important to drive new horizons of research in computational systems biology. For example, as a community organization level, the study group at NIH on quantitative systems pharmacology (QSP) proposed the establishment of inter-disciplinary programs for research and training at various levels, ranging from individual teams to multi-investigator, multi-center programs (Ghosh *et al.*, 2013a). Similarly, Kitano *et al.* (2011) have suggested the needs of virtual big science (Kitano *et al.*, 2011), and the community of mathematical modeling and simulation has started identification of the required tools and concepts and possible approaches of integration of existing ones towards whole cell modeling (Goldberg *et al.*, 2018; Karr *et al.*, 2015).

While digital drivers, as identified in this section, do indeed play an important role for the next generation of computational systems biology, social engineering efforts for community engagement will be equally vital to the success of systems biology.

Case Studies in Multi-Dimensional Data Analysis in Systems Biology and Healthcare

Data and computational methodologies in systems biology have become horizontally and vertically extended and no single analytics can handle all of them. When models and tools are independently developed without an explicit strategy on how they can be integrated, they are likely to be inconsistent in their use of data formats as well as operating procedures. Researchers spend a lot of time and frequently face problems of converting data formats, learning tool operations, and even adjusting operating environments. This hinders productivity, counteracts the flexibility in individual workflows and are liable to errors.

How can we ensure the connectivity among multiple tools in one workflow while securing reusability? As discussed in the former section, several integrated platforms have been developed aiming to have a high-level interoperability between tools in a language-agnostic manner. These platforms strive to offer a consistent user experiences and a broader access to a large plethora of tools and resources. The intention of such platform is to host substantial numbers of analytic tools, data resources or knowledge bases such as domain-specific omics database or molecular database.

There are a wide variety of research motivations to participate in various roles to move the Systems Biology field forward (Fig. 3). In this section, we illustrate a case study of multi-dimensional data analysis based on the Garuda platform (Ghosh *et al.*, 2011). We will begin with a brief introduction of the Garuda platform and its features before moving to specific research workflows.

As discussed before, the Garuda platform is a connectivity, discoverability and automation platform striving to seamlessly and intuitively build analytics workflows. One of the key components of the Garuda Platform is the Garuda Protocol which are a series of messages encoded in JSON format that facilitate applications (gadgets) to exchange information. Comprehensive Application

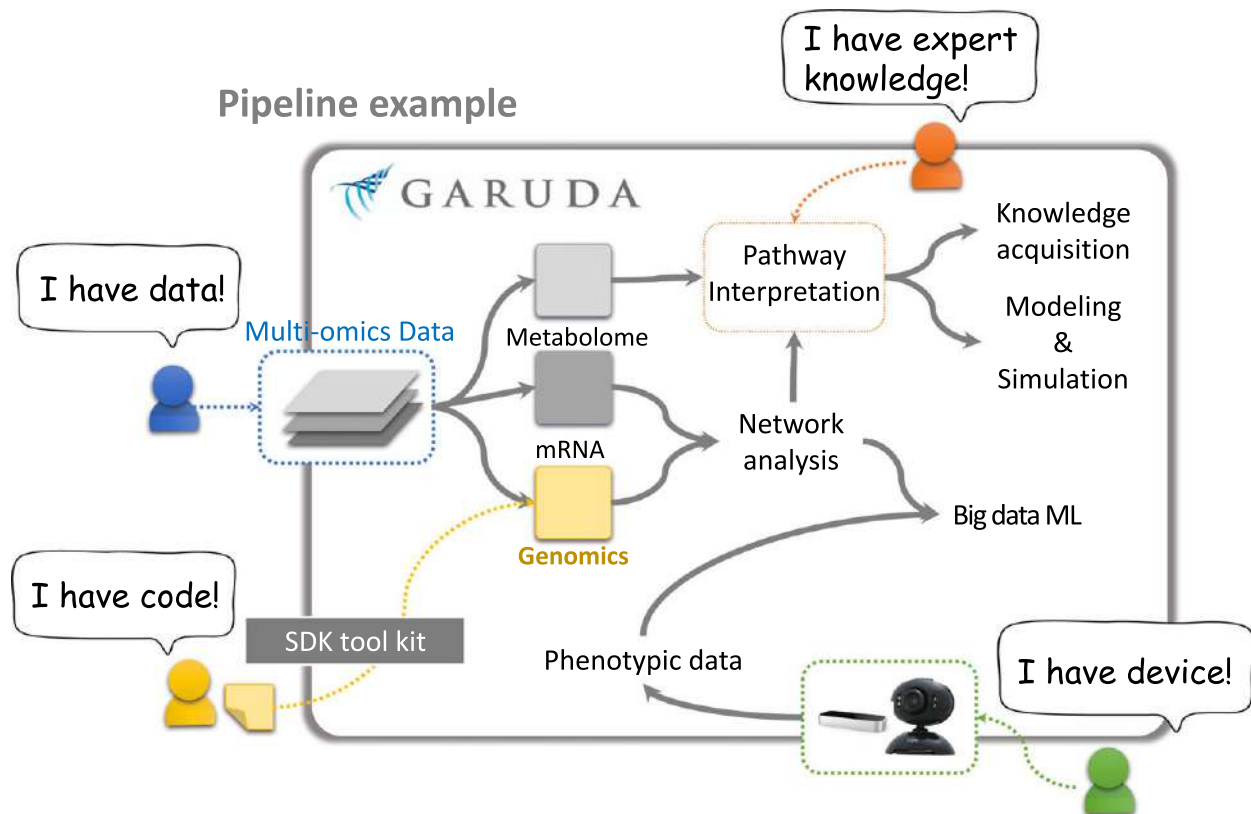


Fig. 3 A wide variety of research motivations for integrated platform-based workflow.

Programming Interfaces (APIs) power connectivity in Garuda. Any application that has been configured to understand these core messages and communicate using them to other tools is said to be “Garudified”. Garudified tools communicate with each other using messages which are coordinated by the Garuda Core. APIs implement the Garuda Protocol in language specific bindings and provide the hooks to communicate with the core. The Garuda core also maintains a gadget graph which depicts the global connectivity of all gadgets in the platform. In addition to connecting existing software tools, one of the attractive feature of the Garuda ecosystem is the ease with which even scripts written in languages like R, python or Perl can be effortlessly wrapped and made into standalone Garudified tools. This facilitates easy sharing and reuse of scripts developed by researchers by making it more visible and available to the community.

Gadgets are distributed using the Garuda Gateway (gateway.garuda-alliance.org). Developers who have garudified their software can list their tools on the gateway with an appropriate license term for the usage of their software. This points to another interesting aspect of the Garuda Platform, the BYOL (bring your own license) model where by while the platform itself is open by design, gadget providers can enforce their own license terms for gadget usage.

There are many interesting use cases of Garuda. For instance, assume that computational analysis was performed on some gene expression data in Garuda. Many times, domain-expert knowledge is required to interpret these analytics results to first gain an understanding of the underlying mechanism as well to point to any gaps. For such cases, deep curated molecular pathway maps (Oda *et al.*, 2005; Oda and Kitano, 2006; Kaizu *et al.*, 2010; Matsuoka *et al.*, 2013) have been made available as a part of Garuda Platform. The results of analytics (which can be a set of gene symbols) can now be mapped onto these maps seamlessly using the Garuda protocol. Practical use cases for Garuda platform are found in a diverse set of applications including clustering, network-based analysis, visualization, machine-learning and mathematical modeling of gene regulatory networks (Hu *et al.*, 2016; Caron *et al.*, 2017; Yachie-Kinoshita *et al.*, 2018). Even for non-specialists, Garuda-compliant software tools offer fast adoptability owing to uniform user-interface guidelines. Garuda platform also provides access to a wide panel of software tools from unrestricted domains without the need for additional learning.

In an example research workflow of oncogenic signaling pathway, researchers may first try to find the mutations associated with a focused cancer subgroup by accessing the sequences of genes encoding the pathway members. They may next predict how the mutations affect the protein structures by searching a three-dimensional protein structure database. Virtual docking simulations can be used to explore possible candidate of kinase inhibitors for the cancer specific signaling. Taking the name of the top-rated inhibitor candidate, they could search for articles of experimental or clinical research reporting the potential effects of the compound or similar compounds on the cell line of interest.

In another scenario, researchers may be eager to know the molecular mechanism of action (MOA), based on the output from the docking simulation. They might want to predict the impact of the kinase inhibitor candidate. In this scenario, they first explore the curated molecular map of the pathway and find the possible influential path from focusing molecule to the molecular end-phenotype such as apoptosis. Next they explore how the candidate inhibitor changes the static path whether a bypass exist. The search for a molecular MOA needs understanding of the dynamics of cell behavior. The researcher may develop or modify existing dynamic simulation model of the pathway to incorporate the mutation-driven changes in the network to predict the cellular dynamics in cancer upon the compounds.

Currently, these entire workflows require a series of separate software tools without information transfer among the tools. A platform successfully integrate these software tools would make such workflows executable with fewer actions, so that researchers can focus on their science rather than on tool operations.

Another key advantage of Garuda is that it can also provide device connectivity and subsequent analytics. Any measurement devices including mass-spectrometers, or any physical devices such as motion sensors and cameras can also be a part of workflow which enables one-stop analysis by connecting the hardware seamlessly with downstream analytics tools (see “Relevant Websites section”). This way experts of various domain other than systems biology, including machine operators, chemist, biologists, data analysts can share the existing and frequently-used workflow in systems biology.

Another interesting use case is when a web-camera takes a picture and using the Garuda protocol automatically sends the data to the next tool where mood detection algorithm is implemented. Such a workflow can be used in clinical settings to monitor patients. Assuming yet another tool downstream of these tools that analyzes the correlation and dynamics between the patient mode and lab results such as blood test and genotype, now the user can have an insight by the integration of phenotypic-level, molecular-level and genetic-level.

Notably, this device connectivity works not only when the device is used as a data source, but also when used as an output. This is realized in the scenario when data analytic tool detects unusual behavior, and can send the alert to doctor's or patient's cell phone in real-time if it is implemented in the platform. Similarly, the connectivity with the device of motion detection, data analytics tools, one can assume the possibility of game-therapy supported by multi-level integrative data such as in diagnosis or monitoring of dementia.

As a whole, the discoverability, navigability and connectivity of multi-dimensional data and analysis are key ingredients that will drive the digital era in biology and medicine (Fig. 4).

Perspectives

Computational systems biology is poised for a paradigm shift standing the cusp of rapidly advancing technologies in experimental systems, data analytics and visualization, machine learning and artificial intelligence (AI) and device technologies.

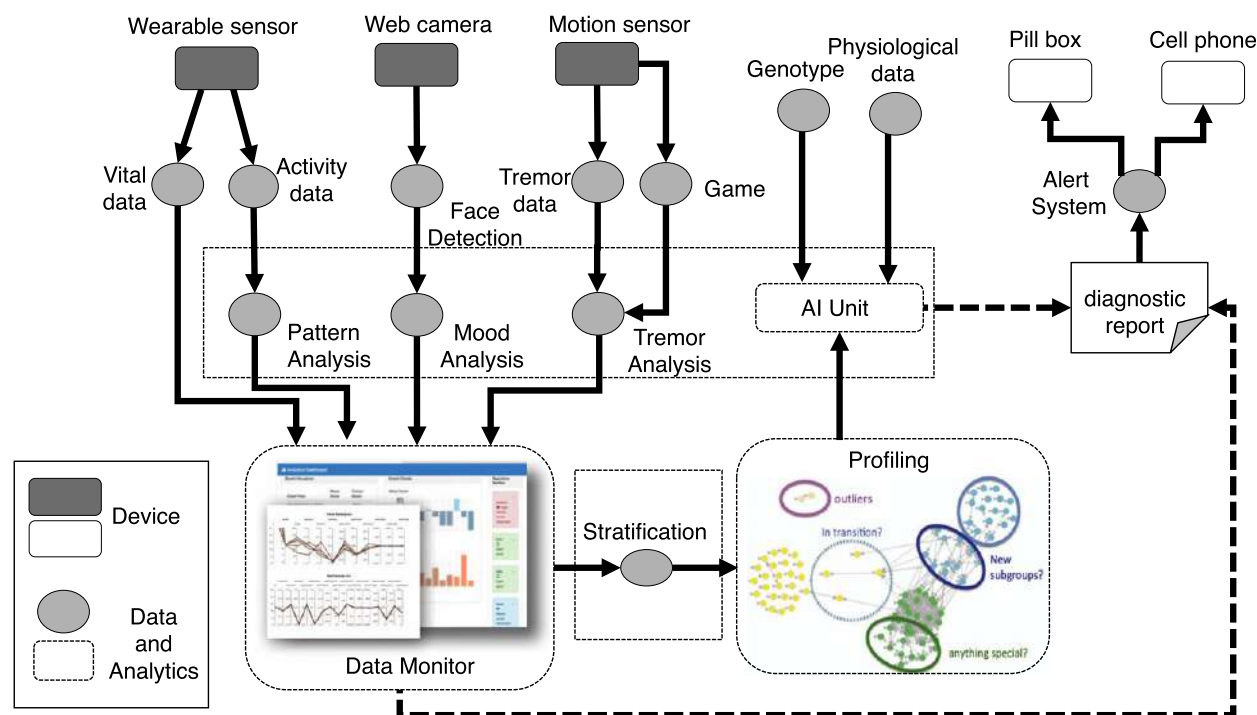


Fig. 4 Case study of connectivity platform for the integration of phenotypic data and data analytics. Adapted from Yachie, A., Matsuoka, Y., 2016. Systems drug and therapy design. Igaku No Ayumi 259 (8), 853–858 (in Japanese).

As outlined in this article, these developments underscore the importance of combining the digital drivers of research and technologies in multi-disciplinary, community-driven approaches to research in biology and medicine.

Novel experimental techniques to study biological processes at multiple levels in exquisite details have been developed. Similarly, advanced computing and machine learning are allowing researchers to extract hitherto unknown patterns in the data and generate novel hypotheses. In order to connect these diverse tools and methodologies, as highlighted in this article, computational systems biology needs to move towards platforms – large-scale, cloud-enabled frameworks which allow flexible connectivity of these tools to build, execute and re-use research pipelines.

At the same time, the inherent complexity of biological systems, particularly, as manifested in multi-omics scale projects, push the cognitive boundaries of the human mind. As outlined in (Kitano, 2016), several issues emerge in large-scale systems level study of biological processes – information horizon problem, information gap, phenotyping inaccuracy, cognitive bias, and minority report problem. Developments in machine learning, text analytics and robotic automation technologies hold the promise of alleviating some of the problems. It is pertinent to note that unlike other fields where advanced AI have shown unprecedented success, their application in biology and medicine needs to be predicated on explainability and interpretability – the ability to trace the model-based decision-making criteria and interpret the results in the context of the biology.

Intelligent platforms, with humans in the loop and powered by high-precision robotic systems, big data management, analysis and highly explainable and interpretable models will usher in new horizons in computational systems biology towards deeper understanding of biology and its applications in medicine and healthcare.

See also: Cloud-Based Molecular Modeling Systems. Natural Language Processing Approaches in Bioinformatics

References

- Aoki, K., Yamada, M., Kunida, K., Yasuda, S., Matsuda, M., 2011. Processive phosphorylation of ERK MAP kinase in mammalian cells. *Proceedings of the National Academy of Sciences of the United States of America* 108, 12675–12680.
- Ben-Hamo, R., Efroni, S., 2013. Network as biomarker: Quantifying transcriptional co-expression to stratify cancer clinical phenotypes. *Systems Biomedicine* 1 (1), 35–41.
- Bois, F.Y., et al., 2017. Multiscale modelling approaches for assessing cosmetic ingredients safety. *Toxicology* 392, 130–139. doi:10.1016/j.tox.2016.05.026. Epub 2016 Jun 4.
- Camarda, R., Alicia, Y.Z., Kohn, R.A., et al., 2016. Inhibition of fatty acid oxidation as a therapy for MYC-overexpressing triple-negative breast cancer. *Nature Medicine* 22 (4), 427.
- Caron, E., Roncagalli, R., Hase, T., et al., 2017. Precise temporal profiling of signaling complexes in primary cells using SWATH mass spectrometry. *Cell Report* 18 (13), 3219–3226.
- Chen, K.C., et al., 2004. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell* 15, 3841–3862.
- Chouard, T., Venema, L., 2015. *Machine Intelligence*. 435.
- Floratos, A., Smith, K., Ji, Z., Watkinson, J., Califano, A., 2010. geWorkbench: An open source platform for integrative genomics. *Bioinformatics* 26 (14), 1779–1780.
- Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.-Y., Kitano, H., 2011. Software for systems biology: From tools to integrated platforms. *Nature Reviews Genetics* 12 (12), 821.
- Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., Kitano, H., 2013a. Toward an integrated software platform for systems pharmacology. *Biopharmaceutics & Drug Disposition* 34 (9), 508–526.
- Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., Kitano, H., 2013b. Toward an integrated software platform for systems pharmacology. *Biopharmaceutics & Drug Disposition* 34 (9), 508–526. doi:10.1002/bdd.1875.
- Ginsberg, J., Matthew, H.M., Patel, R.S., et al., 2009. Detecting influenza epidemics using search engine query data. *Nature* 457 (7232), 1012.
- Goecks, J., Nekrutenko, A., Taylor, J., 2010. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology* 11 (8), R86.
- Goldberg, A.P., Balázs Szegedi, Y.H., Chew, J.A.P., et al., 2018. Emerging whole-cell modeling principles and methods. *Current Opinion in Biotechnology* 51, 97–102.
- Hird, N., Ghosh, S., Kitano, H., 2016. Digital health revolution: Perfect storm or perfect opportunity for pharmaceutical R&D? *Drug Discovery Today*. <https://doi.org/10.1016/j.drudis.2016.01.010>.
- Hood, L., Stephen, H.F., 2011. Predictive, personalized, preventive, participatory (P4) cancer medicine. *Nature Reviews Clinical Oncology* 8 (3), 184.
- Huber, W., Vincent, J.C., Gentleman, R., et al., 2015. Orchestrating high-throughput genomic analysis with bioconductor. *Nature Methods* 12 (2), 115.
- Hu, Y., Huang, K., An, Q., et al., 2016. Simultaneous profiling of transcriptome and DNA methylome from a single cell. *Genome Biol.* 17 (1), 88.
- Ideker, T., Galitski, T., Hood, L., 2001. A new approach to decoding life: Systems biology. *Annual Review of Genomics and Human Genetics* 2, 343–372.
- Imam, S., et al., 2015. Data-driven integration of genome-scale regulatory and metabolic network models. *Frontiers in Microbiology* 6, 409.
- Jain, S.H., Powers, B.W., Hawkins, J.B., Brownstein, J.S., 2015. The digital phenotype. *Nature Biotechnology* 33 (5), 462.
- Kaizu, K., Ghosh, S., Matsuoka, Y., et al., 2010. A comprehensive molecular interaction map of the budding yeast cell cycle. *Molecular Systems Biology* 6 (1), 415.
- Karr, J.R., Takahashi, K., Funahashi, A., 2015. The principles of whole-cell modeling. *Current Opinion in Microbiology* 27, 18–24.
- Kind, T., Fiehn, O., 2009. What are the obstacles for an integrated system for comprehensive interpretation of cross-platform metabolic profile data? *Bioanalysis* 1 (9), 1511–1514.
- Kitano, H., 2000. Perspectives on systems biology. *New Generation Computing* 18, 199–216.
- Kitano, H., 2002a. Systems biology: A brief overview. *Science* 295, 1662–1664.
- Kitano, H., 2002b. Computational systems biology. *Nature* 420, 206–210.
- Kitano, H., 2016. Artificial intelligence to win the nobel prize and beyond: Creating the engine for scientific discovery. *AI Magazine* 37.
- Kitano, H., Ghosh, S., Matsuoka, Y., 2011. Social engineering for virtual big science in systems biology. *Nature Chemical Biology* 7 (6), 323.
- Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., Schaber, J., 2007. Systems biology standards – The community speaks. *Nature Biotechnology* 25 (4), 390.
- Libbrecht, M.W., Noble, W.S., 2015. Machine learning applications in genetics and genomics. *Nature Reviews Genetics* 16 (6), 321.
- Lopez, C.F., Jeremy, L.M., Bachman, J.A., Sorger, P.K., 2013. Programming biological models in Python using PySB. *Molecular Systems Biology* 9 (1), 646.
- Marbach, D., Costello, J.C., Küffner, R., et al., 2012. Wisdom of crowds for robust gene network inference. *Nature Methods* 9 (8), 796.
- Matsuoka, Y., Matsumae, H., Katoh, M., et al., 2013. A comprehensive map of the influenza A virus replication cycle. *BMC Systems Biology* 7 (1), 97.

- Novak, B., Tyson, J.J., 1993. Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos. *Journal of Cell Science* 106, 1153–1168.
- Oda, K., Kitano, H., 2006. A comprehensive map of the toll-like receptor signaling network. *Molecular Systems Biology* 2 (1).
- Oda, K., Matsuoka, Y., Funahashi, A., Kitano, H., 2005. A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology* 1 (1).
- O'Dushlaine, C., Rossin, L., Lee, P.H., *et al.*, 2015. Psychiatric genome-wide association study analyses implicate neuronal, immune and histone pathways. *Nature Neuroscience* 18 (2), 199.
- Perez-Riverol, Y., Bai, M., da Veiga Leprevost, F., *et al.*, 2017. Discovering and linking public omics data sets using the Omics Discovery Index. *Nature Biotechnology* 35 (5), 406.
- Poplin, R., Newburger, D., Dijamco, J., *et al.*, 2017. Creating a universal SNP and small indel variant caller with deep neural networks. *BioRxiv*. 092890.
- Qu, K., Zheng, Y., Dai, S., Qiao, S.Z., 2016. Graphene oxide-polydopamine derived N, S-codoped carbon nanosheets as superior bifunctional electrocatalysts for oxygen reduction and evolution. *Nano Energy* 19, 373–381.
- Reich, M., Liefeld, T., Gould, J., *et al.*, 2006. GenePattern 2.0. *Nature Genetics* 38 (5), 500.
- Schadt, E.E., *et al.*, 2005. An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics* 37 (7), 710.
- Schoeberl, B., *et al.*, 2009. Therapeutically targeting ErbB3: A key node in ligand-induced activation of the ErbB receptor-PI3K axis. *Science Signaling* 2, ra31.
- Schoeberl, B., *et al.*, 2010. An ErbB3 antibody, MM-121, is active in cancers with ligand-dependent activation. *Cancer Research* 70, 2485–2494.
- Schulz, J.C., *et al.*, 2014. Large-scale functional analysis of the roles of phosphorylation in yeast metabolic pathways. *Science Signaling* 7, rs6.
- Shehzad, Z., *et al.*, 2014. A multivariate distance-based analytic framework for connectome-wide association studies. *Neuroimage* 93, 74–94.
- Tyson, J.J., Chen, K., Novak, B., 2001. Network dynamics and cell physiology. *Nature Reviews Molecular Cell Biology* 2, 908–916.
- Vinken, M., 2015. Adverse outcome pathways and drug-induced liver injury testing. *Chemical Research in Toxicology* 28 (7), 1391–1397.
- Wang, M., *et al.*, 2016. Sharing and community curation of mass spectrometry data with GNPS. *Nature Biotechnology* 34, 828–837.
- Webb, S., 2018. Deep Learning for Biology. 555.
- Yachie-Kinoshita, A., Onishi, K., Ostblom, J., *et al.*, 2018. Modeling signaling-dependent pluripotency with Boolean logic to predict cell fate transitions. *Molecular Systems Biology* 14 (1), e7952.
- Yi, T., *et al.*, 2014. Quantitative phosphoproteomic analysis reveals system-wide signaling pathways downstream of SDF-1/CXCR4 in breast cancer stem cells. *Proceedings of the National Academy of Sciences of the United States of America* 111, E2182–E2190.
- Yugi, K., 2016. Trans-Omics: How to reconstruct biochemical networks across multiple 'Omic' layers. *Trends in Biotechnology* 34 (4), 276–290.

Relevant Websites

- <http://www.oecd.org/chemicalsafety/testing/adverse-outcome-pathways-molecular-screening-and-toxicogenomics.htm>
OECD.org.
- www.sbml.org
SBML.caltech.edu.
- <http://www.garuda-alliance.org/gadgetpack/shimadzu>
The Garuda Alliance.

Coupling Cell Division to Metabolic Pathways Through Transcription

Petter Holland¹ and Jens Nielsen, Chalmers University of Technology, Gothenburg, Sweden

Thierry DGA Mondeel¹ and Matteo Barberis, University of Amsterdam, Amsterdam, The Netherlands

© 2019 Elsevier Inc. All rights reserved.

Introduction: The Right Timing of Cell Division

Living organisms have the ability to survive in (un)favourable environmental conditions due to the capacity of their individual cells to grow and divide. The regulation of cell growth and division is realized by the cell cycle, which is set in motion by molecules that regulate and coordinate these events at the right timing.

The cell cycle is defined by four different phases, which are accompanied by strictly ordered and irreversible biochemical reactions. In the G1 phase, the cell grows to a critical cell size that allows it to duplicate its genetic material (DNA) in the S phase, before to continue growing in the G2 phase and, finally, dividing into two daughter cells in the M phase. Because of its crucial role in cell survival, the cell cycle and its component have been conserved across evolution.

In the budding yeast *Saccharomyces cerevisiae*, the maintenance of strictly alternating cycles of DNA duplication and cell division requires a regulator. Dimeric complexes formed by an enzyme, called cyclin-dependent kinase (Cdk1), and a differential pool of phase-specific, regulatory subunits (cyclins) represent the driving force behind cell cycle progression. Cyclin/Cdk1 kinase complexes must be active for entry into S phase and passage through metaphase, and must be inactivated to allow cytokinesis (cell division) in M phase and licensing (activation) in G1 phase of DNA sequences at which new rounds of DNA duplication are initiated. This regulator ensures that each cell cycle phase is completed before the next one initiates, thus guaranteeing alternation between DNA duplication and cell division with a specific temporal delay.

In order to complete these events, periodic waves of cyclin/Cdk1 activities regulate, and are regulated by DNA-binding proteins, called transcription factors (TFs). These proteins stimulate four waves of gene expression (i) at the transition from G1 phase to S phase (G1/S transcription), (ii) in S phase (S phase transcription), (iii) at the transition from G2 phase to M phase (G2/M transcription), and (iv) at the transition from M phase to G1 phase (M/G1 transcription) (Cho *et al.*, 1998; Spellman *et al.*, 1998; Orlando *et al.*, 2008). TFs are activated by cyclin/Cdk activity through biochemical modifications, i.e., phosphorylation, and regulate expression of phase-specific genes throughout the cell cycle. In turn, a transcriptional network drives Cdk activity by stimulating a timely expression of specific cyclin genes. Thus, Cdk activity and the transcriptional network are interlocked to tightly control the timing of cell cycle progression (Bähler, 2005; Wittenberg and Reed, 2005; McNerny, 2011; Haase and Wittenberg, 2014).

The coupling between these cellular layers of regulation has consequences at the system-level. That is, cyclin/Cdk1-mediated waves of gene expression do not just modulate the transcription of cyclin genes at the right timing, but also of a large number of genes involved in the cell's wellbeing. In this context, identification of TFs that may convert a cell cycle response to metabolic switches, and/or that may initiate metabolic reactions whose end products fuel cell cycle reactions, is pivotal to understand how the precise timing of a cell's response is achieved. Ideally, TFs may bridge multiple spatial, temporal and functional scales within various cellular layers of regulation. Thus, they may be hubs (nodes in a network characterized by a high connectivity) at the interface of cellular networks in multi-scale models, which aim to understand how a function emerges from a network of interactions (Castiglione *et al.*, 2014).

By using a Systems Biology strategy, which integrates predictive computer models with biochemical experimentation, we have recently discovered a dynamic coupling of cyclin/Cdk1 complexes to the Forkhead (Fkh) TFs Fkh1 and Fkh2 to guarantee a timely cell cycle progression. Specifically, we have demonstrated that the waves of cyclin/Cdk1 complexes active from S through M phases are synchronized by Fkh2 (and, perhaps, to a lesser extent by Fkh1): S and M phase-specific cyclin/Cdk complexes phosphorylate Fkh2, which in turn promotes expression of S and M phase cyclins (Linke *et al.*, 2017).

The discovery that DNA duplication and cell division are temporally coordinated by Fkh, leads us to ask what is their specific role in the different cell cycle phases. In this study, we aim to understand whether or not Fkh may act as hubs to bridge regulatory processes spanning cell cycle/division, DNA replication, signal transduction and metabolic networks. To this end, we identify targets of Fkh1 and Fkh2 across the budding yeast genome, and indicate whether, and to which extent, a Fkh-mediated transcriptional program leading to the activation of cellular pathways may contribute to the timely cell's cycling. Specifically, we show (i) how definite metabolic pathways may be possibly activated by Fkh, and (ii) their relevance in the synchronization of the cyclin/Cdk1 kinase activities that guarantee alternation of DNA duplication and cell division.

Background: The Interplay Between DNA and Proteins

For the DNA to serve its role as carrier of information in designing proteins in the cell, it is essential to retrieve this information timely and accurately. While protein designs are carried within the DNA sequence, regulation of when and how much of a given protein is produced is achieved by an interplay between the DNA and associated proteins, such as TFs, within the nucleus of eukaryotic cells.

¹Petter Holland and Thierry DGA Mondeel contributed equally to this work.

Corresponding Author: Matteo Barberis, E-mail: matteo@barberislab.com.

One of the most abundant proteins in the cell are histones, which were first discovered as structural components that DNA wraps around. Chemical modifications of histones were later found to regulate DNA functions, affecting how tightly the DNA-histone complexes are packed and interacting with other proteins to modulate gene expression (Tessarz and Kouzarides, 2014).

The signals that regulate gene expression are integrated at the Transcription Start Site (TSS) of genes by a complex of proteins that includes the RNA polymerase. How different signals can lead to various levels of transcription is not well understood; however, a wide range of cellular components and reactions have been demonstrated to influence this process. In addition to the aforementioned histone modifications, the relative position of a gene in the nucleus has been shown to influence gene expression, introducing the concept of transcriptional factories whereby the expression level of a gene may be influenced by the proximity to an active factory (Papantonis and Cook, 2013).

Another group of proteins influencing transcription through a direct interaction with the DNA sequence is represented by TFs. TFs are defined as proteins that bind to a specific 'motif', i.e., a well-defined DNA sequence, with a typical length of 5–12 base pairs. In budding yeast, TFs are thought to influence transcription, first, by binding relatively close to the TSS, and then by interacting with histones and other TFs on the DNA, thereby influencing the assembly of a productive TSS protein complex. In multicellular eukaryotes, enhancer elements exist that can influence transcription at a long distance from the TSS.

Fundamentals: Methodologies to Investigate DNA–TF Interaction

Understanding how, when and why TFs to DNA has become a major undertaking to understand transcriptional regulation. The first genome-wide method realized to experimentally map TF binding across the genome in living cells was Chromatin Immunoprecipitation (ChIP) with microarray detection of bound DNA (ChIP-chip) (Solomon *et al.*, 1988). This technique relies on: (i) Using formaldehyde to chemically link proteins and DNA together, (ii) sonicating the DNA to fragment it, (iii) purifying a selected protein by antibody, and (iv) detecting the DNA fragments that are bound to the purified protein. Studies that have mapped the DNA binding sites of many TFs provided the first integrated view of the proteins and regulatory mechanisms that control eukaryotic gene expression (Harbison *et al.*, 2004).

With the emergence of next generation sequencing, further development of the ChIP methodology led to ChIP-seq (Robertson *et al.*, 2007), which could achieve higher resolution as compared to ChIP-chip because of not being limited by the amount of probes on the chip. Multiplexing with barcoded adaptors and increased availability of commercial sequencing services have lowered the costs of ChIP-seq experiments, to a point where it has now replaced ChIP-chip as the dominant method of genome-wide determination of TF binding to the DNA (Park, 2009). Importantly, ChIP-seq was used as the method of choice for the ENCODE consortia to map a large number of human TFs with a consistent set of experimental and data processing protocols (Landt *et al.*, 2012).

The latest development of the ChIP methodology is called ChIP-exo, which further ameliorate ChIP-seq by using an exonuclease to degrade one strand of the isolated DNA (Rhee and Pugh, 2011). The exonuclease cleaves one DNA strand until the TF – DNA crosslink, and adaptors are then ligated to the cleaved ends, leading to a high resolution information of the TF – DNA contact surface. The general ChIP protocol and differences among the three ChIP methods are illustrated in Fig. 1, whereas the main advantages between ChIP-seq and ChIP-exo are listed in Table 1. An impressive demonstration of the increased resolution of ChIP-exo was the mapping of components of the pre-initiation complex in budding yeast. In this study, the area of DNA that was protected by the exonuclease was a good match for how a specific protein was predicted to cross-link to the DNA from a model of DNA and protein structures (Rhee and Pugh, 2012).

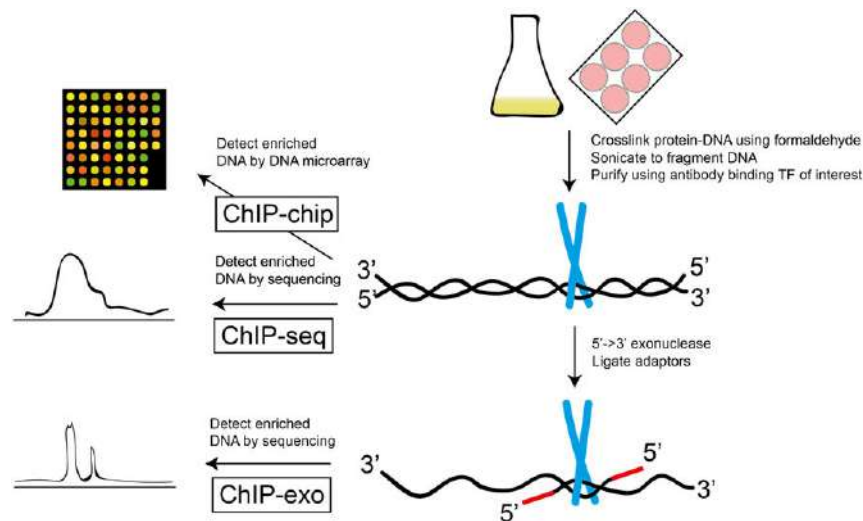


Fig. 1 Summary of the ChIP experimental protocol and overview of the main differences between three ChIP methods: ChIP-chip, ChIP-seq and ChIP-exo (see text for details).

Table 1 Advantages of ChIP-seq and ChIP-exo when compared to each other

<i>ChIP-seq</i>	<i>ChIP-exo</i>
– Easier and quicker experimental protocol – Fewer costly reagents – More developed tools for data analysis – Immunoprecipitation control allows for the analysis of unspecific antibody binding	– Higher resolution mapping of TF–DNA binding – Footprinting several TFs can provide structural insights of protein complexes – Less background binding to the DNA due to exonuclease treatment

The choice of which ChIP methodology to use when investigating a given TF is not obvious, and several factors should be considered. Although ChIP-exo is the most recent development and seems to be the most attractive choice if the resources are available, new challenges are introduced when using this ChIP methodology as compared to ChIP-seq. Notably, a treatment with the exonuclease makes it impossible to have an immunoprecipitation (IP) control, as typically used in ChIP-seq experiments. Recently, an interesting solution to this problem, called protein attached chromatin capture (PATCh-Cap) (Terooatea *et al.*, 2016), was proposed. PATCh-Cap seems to increase the confidence in peak identification; however, it remains untested at a large scale to exclude any introduction of bias. The lack of an IP control means that the most common ChIP peak quantification metric, i.e., ChIP peak signal/IP control signal, is no longer possible and other metrics have to be used. While the lack of an IP control complicates the measurement of a ChIP peak signal, ChIP-exo seems to have a reduced level of background binding as compared to ChIP-seq (Rhee and Pugh, 2011), resulting in a potentially more accurate quantitative measurement of TF binding to the DNA. In addition, an increased resolution given by the exact position of the TF binding event can potentially be used in a system-level analysis to describe gene regulation in detail.

The network of cellular signals that regulate expression of a gene or of a group of genes is often referred to as Transcriptional Regulatory Network (TRN). While a variety of signals affects gene transcript levels, much of the current studies focus on TFs. Importantly, in recent years, the power of system-level analyses that integrate TF binding information with transcriptome analysis has been demonstrated, for example to understand the bacterial TRN of gene expression (Arrieta-Ortiz *et al.*, 2015; Fang *et al.*, 2017). Although many details are known about eukaryotic gene regulation, a TRN with demonstrated predictive power is lacking. This is partially due to the fact that the eukaryotes genome comprehends more genes and TFs, and that additional complexity exists that is introduced by histone occupancy and modifications, TF subcellular localization and a variety of post-translational modifications. These additional levels of regulation mean that more information needs to be taken into account in a system-level analysis to understand how eukaryotic TRNs regulate gene expression.

ChIP data provide a list of targets of TFs that may be subsequently analyzed for their specific function within cellular networks. Given the wealth of data that is available about genes and the proteins they encode, ChIP data can predict novel functions of TFs. When retrieving these data, a number of challenges are faced: (i) False-positives binding events have to be excluded, and (ii) a comprehensive picture of the TF functions from the wealth of data generated shall be drawn. In this study, we present and discuss our approach to dealing with these challenges when investigating the Fkh-mediated transcriptional program. A picture of Fkh's functions within the multi-scale network of the budding yeast cell demonstrates the power of the ChIP methodology to generate hypotheses and suggest definite experiments that may shed light on novel regulatory events underlying timely cellular responses upon stimulation.

Application: Chip Data Formats

Choosing the ChIP methodology to study TF binding events should take into account the differences in data processing between these, in addition to those in the experimental protocols. ChIP-seq is currently well established, and a major resource is represented by the detailed experimental protocols and data processing pipelines that have been established for the ENCODE consortium experiments (Landt *et al.*, 2012). In the following, we will describe briefly the main established protocols currently available for the identification of a ChIP-seq signal, and the alternative approaches that we have developed for analyzing ChIP-exo data.

Established Peak Detection Protocols: GEM and MACE

One of the data analysis tools selected for the ENCODE ChIP-seq experiments is called GEM (Genome wide Event finding and Motif discovery) (Guo *et al.*, 2012). GEM (i) determines the shape of peaks corresponding to TF binding events on the DNA (peak detection), and motifs enriched in the most common peaks in a given sample, and (ii) iterates on both peak shape and motif to enrich the dataset for peaks that generate a weaker signal with a similar peak shape and motifs. This strategy can also be used to analyze ChIP-exo data; however, the main limitation is the need for a large number of peaks to reliably iterate the procedure.

Another peak detection tool is called MACE (Model based Analysis of ChIP-Exo) (Wang *et al.*, 2014). MACE separates the reads of different DNA strands and searches for a definite distance (called 'border pair') between the start of reads that is enriched across the sample. The enriched border pair distance should correspond to the size of the TF – DNA interaction surface. A set of clear (strong) peaks defined as 'elite border pairs' is required for the analysis, in order to have a reliable starting point to define the remaining weaker peaks. Thus, the limitations of MACE are similar to GEM, with a requirement for a relatively large amount of strong peaks.

Alternative Approaches: *maxPeak* and *loessPeak*

While studies of human TFs binding to DNA typically provide a sufficient number of peaks to be analyzed by iteration-based peak detection methods, our experience with budding yeast indicates that not all TFs will have enough targets to achieve a reliable iteration procedure when using GEM or MACE. This led us to develop alternative tools to define peaks that are not dependent on iteration, thus that can process data containing fewer peaks. To define what a peak is, relative to an uneven background signal, is not a trivial task when only relatively few peaks are detected.

According to the principle of ChIP-exo, the positions inside the TF – DNA binding surface should be highly enriched for overlapping reads from each DNA strand, while unspecific reads from experimental background or technical noise may be on only one DNA strand. One approach that we have developed is to reduce the background signal by using the strand-specific information and counting up reads only where there were reads overlapping on each DNA strand; we will denote this approach by OL, for overlapping DNA strand, as opposed to AD, for additive DNA strand approach. The term ‘overlap’ is here to be interpreted as having, at a minimum, one read on each DNA strand covering the same single nucleotide position. Hence, these two reads overlap on that nucleotide, albeit on different DNA strands.

To further reduce the background noise, we determine a read length to be used where a TF binding site is covered by reads on each strand, but not extending out on each side. Reads coming from our experimental setup have a length of 75 base pairs, and choosing a read length of 10 would correspond to taking the left-most 10 base pairs into account. The exact read length to be used is tested by quantifying the highest point of overlapping reads per gene for read lengths from 1 to 75 base pairs, and by selecting the read length where the increase in highest peak counts is most increased over the previous three read lengths. For the Fkh TFs investigated in this study, this read length was determined to be equal to seven. Given the read length, each read is shortened to correspond to the read length (starting to count at the left-most base pair). Given the read length, either through the optimisation procedure indicated above or by choosing it manually, and the approach for dealing with the strands, OL or AD, we will hereafter denote the combination of these two as e.g., OL7 or AD10, for read lengths of 7 or 10 base pairs, respectively.

When investigating TF binding events, a choice has to be made with respect to the part of the genome to be explored. Two options may be considered: (i) Use of the known TSS for the subset of genes for which this information is available, and search 1000 base pairs upstream of TSS, or (ii) use the known locations of the ORF (Open Reading Frame) for each gene, and search 1000 base pairs upstream of the start codon. The choice of a region of 1000 base pairs is somewhat arbitrary, but is limited by the fact that a too large region might end up in the ORF of an upstream gene. The advantage of the former approach is that transcription starts from the TSS so it defines a more accurate region as the promoter of a gene, whereas the advantage of the latter is that information can be retrieved from a higher number of genes even if the TSS location is unknown. In practice, the difference of using the two approaches is minimal for the majority of genes that have a defined TSS, which have a distance from the TSS to the start codon of about 20–80 base pairs. In a parallel study, we have considered TSS from the study of Iyer and colleagues (Park *et al.*, 2014) as well as 166 manually added approximate TSSs based on RNAseq data, for a total of 5373 TSSs (unpublished). Here, we instead search upstream of the start codon of the ORF for each gene.

At this point, the ChIP-exo dataset consists of short reads, overlapping on one (AD) or both (OL) DNA strands, which we can count at each base pair. Consequently, a series of peaks of different heights may be retrieved along the genome. In this work, we present two novel approaches to determine TF binding locations on the DNA, which we refer to as ‘maxPeak’ and ‘loessPeak’.

In the maxPeak approach, we locate the base pair with the highest count of reads for each gene’s promoter region. This means that only one TF binding location for any given gene may be found. As previously mentioned, a challenge with ChIP-exo as compared to ChIP-seq is the lack of an antibody immunoprecipitation (IP) control that can be used to measure quantitatively a specific TF – DNA enrichment relative to an unspecific antibody – DNA enrichment. In order to have a relative measurement of the peak strength (signal/noise) and to remove the background signal, we detect peaks by using peak quantiles. Quantiles are calculated from the maximum signal per gene promoter of all promoters that have any signal. Our experience from ChIP-exo analysis of other budding yeast TFs is that the 65% quantile (the 65th percentile) is a threshold that results in a good representation of peaks as well as removal of the majority of the background and noisy signal (unpublished observations). In contrast, the signal/noise for GEM is calculated by dividing the IP signal of the peak by the ‘expected binding strength’ that calculates the background signal in the local context around the detected peak. In MACE, peak significance is reported as a p-value. A subtlety to take into account when considering the 65th percentile threshold is that a similar background noise may be assumed across experiments based on different conditions (e.g., different growth) for the same TF. In these cases, it may be useful to average the threshold for the conditions and use the new value as the normalizing factor. We will return to this specific point in the section of the ChIP data analysis.

After applying the maxPeak approach, and before applying the loessPeak approach, TF binding sites may be ordered by the height of their signal. In the maxPeak approach, this corresponds to the number of reads at the peak binding position (i.e., zero or one location per gene). For the loessPeak approach, this corresponds to potentially many peaks per gene. For either approach, TF binding locations may first be sorted in ascending order and then reduced by applying the threshold of the 65th percentile, which is chosen as the quantile threshold to consider a binding position non-random. For the maxPeak approach, the signal of each identified peak is then normalized (divided) by this quantile, yielding a Signal-to-Noise Ratio (SNR) below 1 if the location has a lower signal than the threshold and a SNR > 1 if its signal is above the threshold. For the loessPeak approach, any signal in the promoter that is below the 65th percentile is set to that value.

In the loessPeak approach, after leveling any signal below the 65th percentile, the signal at the promoter is smoothed with a loess function using a span of 0.05. This is only a mild amount of smoothing where the main goal is to get an accurate position of the

highest point of the peak while also having a small differential effect on strong and weak peaks. The smoothing causes weak peaks that are, for example, very narrow to be relatively reduced as compared to wider and stronger peaks. Subsequently, peaks are identified by searching for a pattern signal change in the smoothed signal containing minimum 7 consecutive increasing signal values and then minimum 7 consecutive decreasing signal values. For a gene, all identified peaks are listed by order of signal strength and, if a lower peak signal is within 100 base pairs of a stronger peak, the weaker one is discarded. All peaks are then normalized by the 65th percentile. Generally, peaks with a remaining, smoothed and normalized, $\text{SNR} < 1$ are discarded.

Biological duplicates are explicitly averaged in the maxPeak and loessPeak approaches when the number of reads for each nucleotide is quantified. GEM and MACE internally deal with the biological duplicates.

The workflow of our experimental protocol, starting from ChIP-exo sequencing reads, is summarized in [Fig. 2](#), whereas a visualization of the maxPeak and loessPeak approaches is shown in [Fig. 3](#).

Case Study: ChIP-exo Data Analysis of Fkh1 and Fkh2 TFs

The Forkhead (Fkh) TFs Fkh1 and Fkh2 modulate a timely cell's cycling in budding yeast, by coordinating DNA duplication with cell division. Fkh2 is expressed from the G1 phase until the M phase, with Fkh1 overlapping during S and G2 phases ([Lee et al., 2002](#)).

In the G1 phase of the cell cycle, Fkh guarantee the right spatial organization of chromatin within the nucleus of yeast cells. Specific DNA binding sites, called origins of replication or replication origins, timely cluster prior DNA duplication and are subsequently recognized by enzymes (DNA polymerases) that initiate this process. Fkh1 and Fkh2 were shown to be involved in this mechanism by binding to, and regulating the transcription timing of a number of replication origins ([Knott et al., 2012](#); [Ostrow et al., 2014](#)) by acting as rate-limiting activators ([Peace et al., 2016](#)). We have also demonstrated that Fkh proteins are responsible for the clustering of replication origins in the S phase, thereby of the timing of DNA duplication, through a structural motif that allows dimerization to bring distal DNA binding sites into close proximity ([Ostrow et al., 2017](#)).

Fkh-dependent activity also drives transcriptional regulation of genes at the G2/M transition, in particular of the *CLB2* cyclin gene promoting cell division ([Koranda et al., 2000](#); [Kumar et al., 2000](#)). Although Fkh1 and Fkh2 have overlapping functions, only Fkh2 can form a complex, together with the scaffold protein Mcm1 and the co-activator Ndd1, that binds to and modulate the activity of both *CLB2* and *CLB3* promoters ([Koranda et al., 2000](#); [Kumar et al., 2000](#); [Pic et al., 2000](#); [Reynolds et al., 2003](#); [Linke et al., 2017](#)). Conversely, Fkh1 binds less efficiently to the *CLB2* promoter and represses *CLB2* transcription ([Hollenhorst et al., 2000, 2001](#); [Sheriff et al., 2007](#)).

Fkh expression and function throughout cell cycle progression suggests that these TFs may have the potential to establish regulatory interactions with cellular networks beside the cell cycle. Therefore, in this study, we have explored the capability of Fkh1 and Fkh2 to bind promoters of genes across the budding yeast genome through ChIP-exo. ChIP data have been gathered under two different experimental conditions (cells grown exponentially, or logarithmic phase, and in stationary phase) to compare the data analysis tools described above, and their challenges and advantages will be highlighted in the following. Fkh1 and Fkh2 have an intermediate amount of peaks as compared to other budding yeast TFs (unpublished observations), thus may serve as good case studies to test the workflow and challenges in working with ChIP-exo in this organism. The workflow shown in [Fig. 2](#) has been implemented in this study for the Fkh1 and Fkh2 data analysis.

Challenges With GEM and MACE

GEM did reliably iterate on Fkh1 data (two different experimental conditions, two biological replicates of each condition), but not on Fkh2 data. Considering that Fkh1 and Fkh2 are proposed to have similar binding motifs ([de Boer and Hughes, 2012](#)), we ran Fkh1 and Fkh2 data together in the GEM analysis, treating them as different conditions of the same TF. As a result, a reliable iteration was possible for both data. Similarly, MACE also presented challenges when analyzing Fkh2 data; the detected amount of elite border pairs was only 25, and the MACE python script by default fails with less than 30 elite border pairs. Therefore, the script was modified to reduce the existing threshold, thus allowing us to analyze the data for Fkh2 as well.

Determination of the Read Length

As described above, the OL and AD approaches applied to the maxPeak and loessPeak methods of peak detection do not require iteration or minimal numbers of peaks of a certain quality. Data analysis was performed four times: the optimal read length was determined to be 7 base pairs for both Fkh1 and Fkh2; however, in order to test the sensitivity to the read length setting, we have also performed the analysis for a manually determined read length of 10 base pairs. For both read lengths we performed the analysis by using the overlapping DNA strand approach (OL) and the additive DNA strand approach (AD). In [Fig. 4](#) the raw reads and peak detection results are visualized for the four data analysis methods (maxPeak, loessPeak, GEM and MACE,) for the genes *PES4* (coding for a poly(A) binding protein) and *CLB2* (coding for a mitotic cyclin that promotes the transition from the G2 to M phase), which have the highest SNR for Fkh1 and Fkh2, respectively.

In [Fig. 5](#) the signal of the peaks against the percentiles of raw data for both Fkh1 and Fkh2 is plotted. As previously mentioned, based on prior experience, the 65th percentile represents a good threshold, and it also seems to be a good choice to analyze Fkh1 and Fkh2 data. In fact, this value sits in a relatively flat area of the curves where the difference between both TFs is not too large.

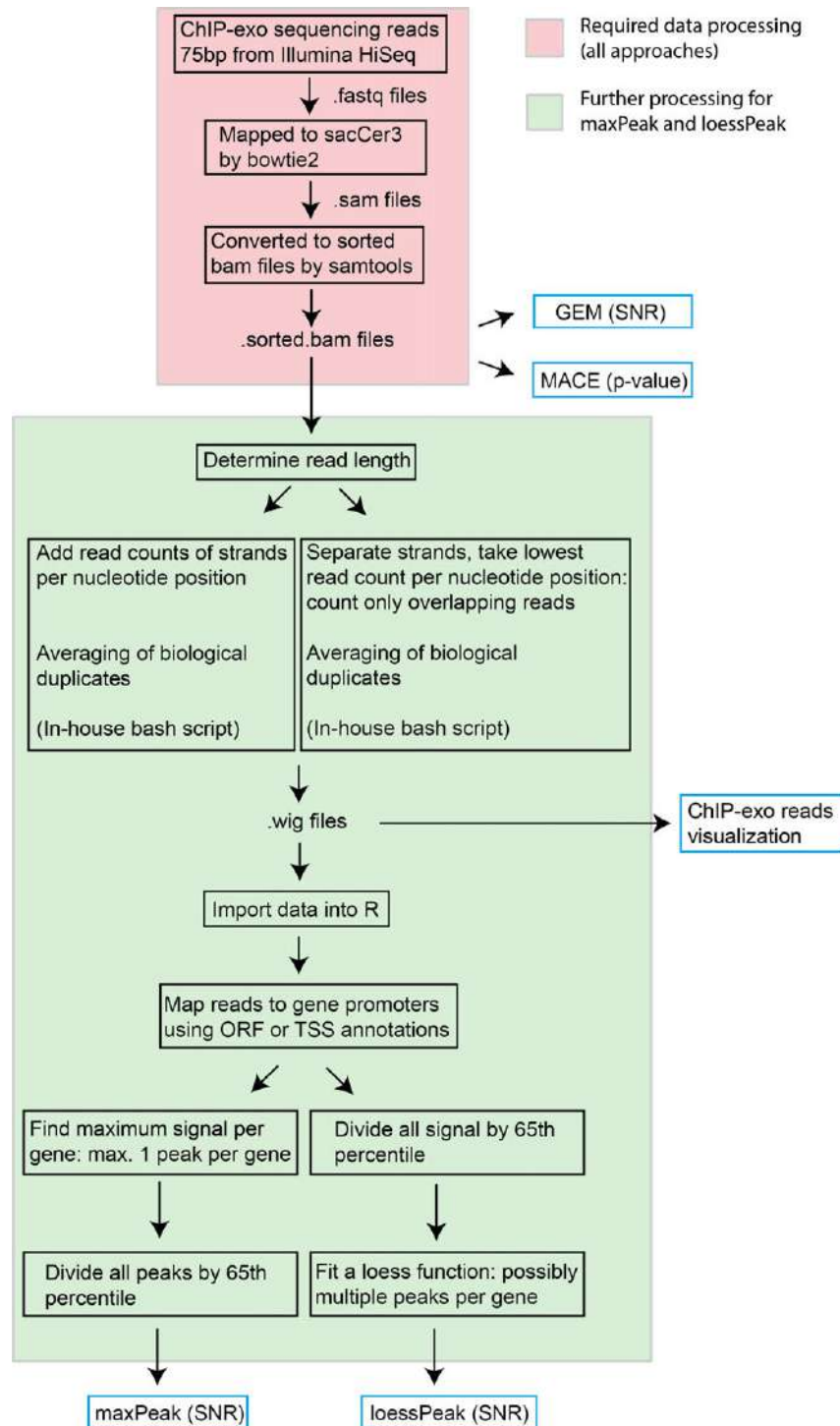


Fig. 2 Overview of the ChIP-exo data analysis workflow. To map reads to the budding yeast genome, raw reads were mapped to the SacCer3 genome (S288C_reference_sequence_R64-2-1_20150113.fsa, downloaded from the Saccharomyces Genome Database (SGD) website (see “Relevant Websites Section”) with Bowtie 2. (Langmead and Salzberg, 2012). SAM files were converted to BAM files, sorted and indexed using SAMtools (Li *et al.*, 2009). Output indicated as blue rectangles is shown in Fig. 5.

Data Integration and Annotation

After ChIP data processing was completed, we have integrated and annotated the data to retrieve biological information. We have recently developed GEMMER (see “Relevant Websites section”), a web-based visualization tool of multi-scale interaction networks that integrates information from various databases for budding yeast (Mondeel *et al.*, 2018). The ChIP data were merged with the

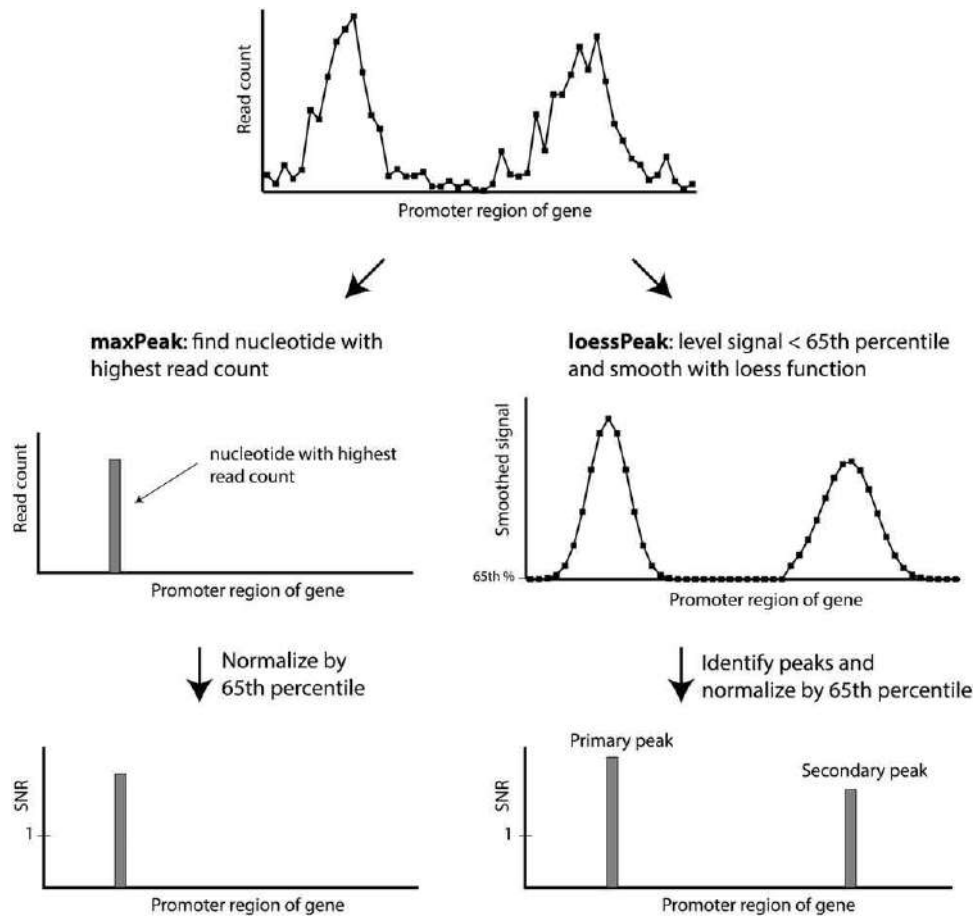


Fig. 3 Illustration of the maxPeak and loessPeak approaches. The workflow depicted applies to each gene in the genome. Reads are counted across all base pairs within the promoter region of a gene. The maxPeak approach (left) identifies a single nucleotide with the highest read count in each promoter region. The genome-wide 65th percentile is calculated, and used to normalize the read count of each peak, generating a Signal-to-Noise Ratio (SNR). In contrast, the loessPeak approach (right) starts by calculating and normalizing the peak signals by the genome-wide 65th percentile of the highest point of all genes that are > 0 , and then applies a loess smoothing function; finally, any remaining peaks with a $\text{SNR} \geq 1$ are identified.

gene annotations contained in GEMMER. This process has extended the experimental data available for each protein-coding gene in SGD, with information regarding functional annotation, localization, abundance, and both timing and cell cycle phase of peak occurrence of RNA transcript levels. Thus, this step allows for a number of novel analyses of the ChIP-exo data. Of note, the vast majority of genes reported by the data analysis are protein-coding genes; however, some exceptions exist.

Comparing Data Formats and Peak Detection Methods' Performance

We have explored the number of genes and peaks detected for the four different peak detection methods introduced above, and we found significant differences in the number and locations of peaks detected (see Fig. 4). Moreover, as previously mentioned, the quantile determination may be performed independently for each experimental condition, or determined as average for multiple conditions for each TF. We will therefore briefly explore the difference for the Fkh1 and Fkh2 datasets with respect to these different determinations.

Table 2 reports the number of TF binding locations above threshold in unique genes and the (equal or larger) number of distinct peaks for GEM ($\text{SNR} \geq 1$), MACE ($p\text{-value} \leq 0.01$), maxPeak (OL7 and AD10 with $\text{SNR} \geq 1$) and loessPeak (OL7 and AD10 with $\text{SNR} \geq 1$) when determining the average quantile for both experimental conditions (growth of cells in logarithmic phase or in stationary phase) for each TF. Similarly, Table 3 reports the number of peaks and genes retrieved when the quantiles are determined independently for each experimental condition. Of note, only the maxPeak and loessPeak approaches are affected by the choice of the quantiles (see workflow in Fig. 2). In Table 2 a significant increase in genes/peaks may be observed in stationary phase for both Fkh1 and Fkh2 as compared to the logarithmic phase, when using maxPeak and loessPeak. This evidence suggests that the assumption of an equal background noise across these conditions may be not realistic. In Table 3 a reduced ratio of genes/

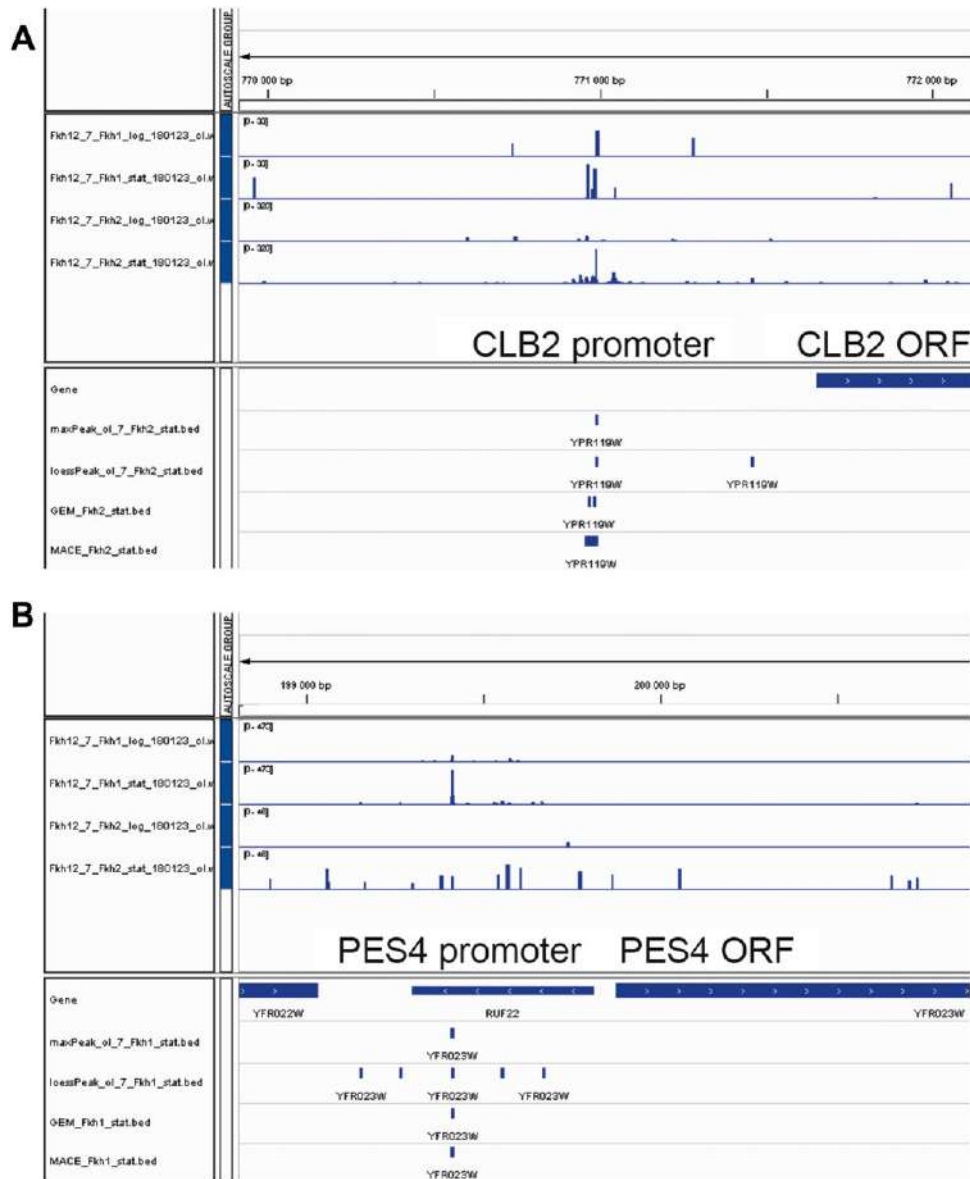


Fig. 4 Analysis of ChIP-exo reads (top part of panels) by different peak detection methods for two genes. (A) *CLB2*, selected for having the highest peak for Fkh2. (B) *PES4*, selected for having the highest peak for Fkh1. The maxPeak (OL7) approach returns a single peak for in the promoter region of both genes. GEM and loessPeak may return multiple peaks, whereas MACE returns a region of several consecutive nucleotides as the peak.

peaks across experimental conditions may be observed, although a large increase in the signal above threshold still occurs in stationary phase experiments. For example, the ratio of target genes identified by maxPeak OL7 between the two conditions for Fkh1 changes from $747/1422=0.525$, to $816/1363=0.599$. We have proceeded to analyze ChIP data based on quantiles determined independently for each experimental condition.

Investigating the Discrepancy Between OL7 and AD10

So far we have been considering the AD10 dataset as a simultaneous illustration of the additive approach to counting reads on both DNA strands, and of the change of the read length. Tables 2 and 3 clearly point out that large differences exist in the analysis between the OL7 and AD10 datasets, even within one approach, e.g., maxPeak. In Fig. 6(A) the fold-change in the SNR among three sets of genes is shown when using the maxPeak OL7 and AD10 datasets for Fkh1 in logarithmic phase: (i) The 100 genes with the highest signal in OL7, (ii) all genes in OL7 with a $SNR \geq 1$, and (iii) all genes in OL7 with a $SNR > 0$. Large fold changes are observed in all three groups. This difference is more apparent in Fig. 6(B), where the difference in rank for genes in each of the three groups is plotted. The term 'rank' here indicates the index of a given gene when the dataset is sorted on the SNR in descending

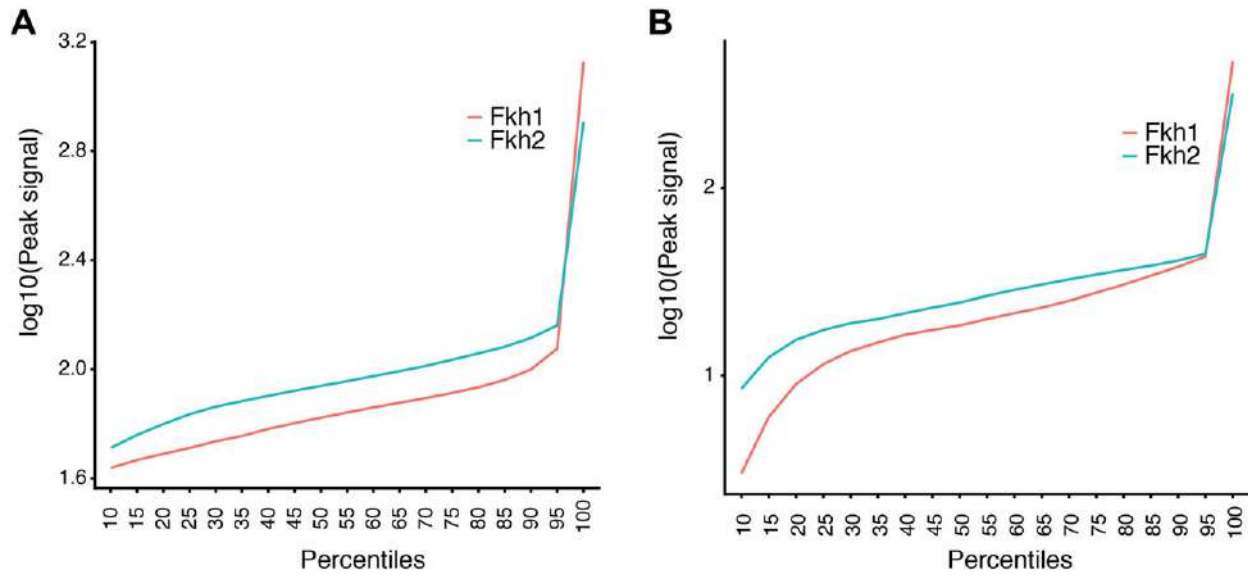


Fig. 5 Peak percentiles across all genes with signal >0 for two data formats explored. (A) Summing up all read counts per nucleotide position with 10 base pairs long reads (AD10). (B) Counting only overlapping reads per nucleotide position with 7 base pairs long reads (OL7). It can be observed that the signal is lower in the OL7 approach, due to the more stringent approach of only taking overlapping reads on both DNA strands.

Table 2 Unique target genes and total peak counts for TF normalized data (averaged across experimental conditions) analyzed by four peak detection methods. A low threshold was used for considering target genes significant (see text), and for maxPeak and loessPeak both overlapping (OL) and additive (AD) DNA strand approaches were applied using read lengths of 7 and 10 base pairs, respectively. Of note, for the maxPeak approach the number of genes and peaks is always equal

	<i>maxPeak OL7</i>		<i>maxPeak AD10</i>		<i>loessPeak OL7</i>		<i>loessPeak AD10</i>		<i>GEM</i>		<i>MACE</i>	
	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>
Fkh1 mid-log	747	747	1880	1880	726	779	1786	2116	215	221	439	451
Fkh1 stationary	1422	1422	3029	3029	1372	1564	2923	3937	248	259	814	841
Fkh2 mid-log	686	686	755	755	628	655	713	783	132	139	691	699
Fkh2 stationary	2944	2944	4170	4170	2800	3497	4055	5940	170	184	1142	1196

order. A negative index indicates a lower rank in the AD10 dataset, whereas a positive rank indicates a higher rank in the OL7 dataset. Strikingly, there are large shifts in rank that correspond with the fold changes; even among the top 100 OL7 targets there are genes that shift down hundreds of places in AD10.

What is the reason for the observed difference? In [Fig. 6\(C\)](#) and [\(D\)](#) the same fold change and rank displacement analysis is reported when comparing OL7 and OL10. Here, the differences are almost insignificant, especially among the genes with a higher signal. It appears that changing the read length from 7 to 10 base pairs has only a little consequence to the data analysis. To drive the point home, in [Fig. 6\(E\)](#) and [\(F\)](#) the same analysis is performed among OL7 and AD7 datasets. The similarity with [Fig. 6\(A\)](#) is remarkable. Therefore, we conclude that the maxPeak and loessPeak (the latter not shown) approaches are relatively insensitive to the read length setting, but very sensitive to the choice of how to deal with reads on different DNA strands. Given that the OL approach reduces the background noise – as it is intuitively apparent and reflected in the number of genes/peaks reported in [Tables 2](#) and [3](#) –, we hereafter proceed considering only the OL approach with a read length of 7.

Increasing the Stringency for TF Binding Significance

Analyzing in more detail [Table 3](#), it can be observed that GEM is the most stringent among the peak detection methods by far in terms of the number of genes and peak locations it returns as being significant. Following the results shown in [Fig. 6](#), we have realized that especially the maxPeak and loessPeak approaches may overestimate the number of significant TF binding events when using a threshold of $\text{SNR} \geq 1$. Therefore, we have repeated the data analysis for more stringent thresholds with the aim of uncovering the more reliable Fkh target genes (see [Table 4](#)).

Table 3 Unique target genes and total peak counts for condition-normalized data analyzed by four peak detection methods. A low threshold was used for considering target genes significant (see text), and for maxPeak and loessPeak both OL and AD DNA strand approaches were applied using read lengths of 7 and 10 base pairs, respectively. Of note, for the maxPeak approach the number of genes and peaks is always equal

	<i>maxPeak OL7</i>		<i>maxPeak AD10</i>		<i>loessPeak OL7</i>		<i>loessPeak AD10</i>		<i>GEM</i>		<i>MACE</i>	
	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>
Fkh1 mid-log	816	816	2460	2460	796	858	2349	2910	215	221	439	451
Fkh1 stationary	1363	1363	2473	2473	1319	1499	2365	3062	248	259	814	841
Fkh2 mid-log	1292	1292	2463	2463	1182	1283	2362	2911	132	139	691	699
Fkh2 stationary	2361	2361	2470	2470	2239	2672	2367	2919	170	184	1142	1196

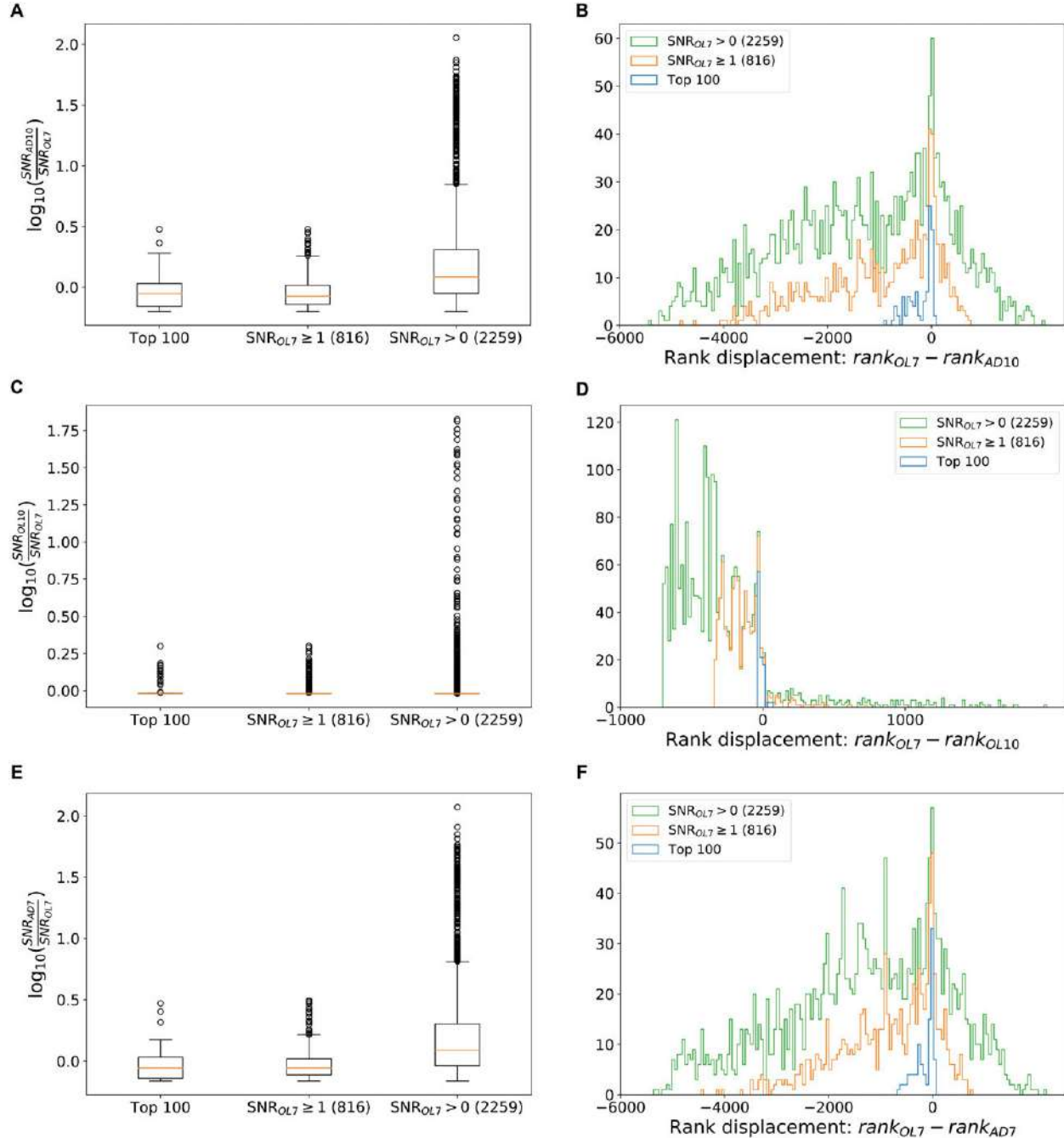


Fig. 6 Comparison of the rank displacement and SNR log10 fold-change between the maxPeak OL7 and AD10 datasets for Fkh1 in logarithmic phase.

Table 4 Unique target genes and total peak counts for condition-normalized data analyzed by four peak detection methods. A higher threshold was used for considering target genes significant (see text), and for maxPeak and loessPeak only the OL DNA strand approach was applied using a read lengths of 7 base pairs. Numbers highlighted in red color indicate the condition for a particular dataset that approximately reaches the number of genes retrieved by GEM for definite stringency settings

	<i>maxPeak OL7</i>		<i>loessPeak OL7</i>		<i>GEM</i>		<i>MACE</i>	
	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>	<i>Genes</i>	<i>Peaks</i>
Fkh1 mid-log	348	348	287	301	215	221	219	223
Fkh1 stationary	530	530	467	496	248	259	403	408
Fkh2 mid-log	452	452	360	370	132	139	191	193
Fkh2 stationary	167	167	169	169	170	184	281	282

To this aim, we used a heuristic approach: We leave GEM stringency as is, and we increase the thresholds for maxPeak, loessPeak and MACE independently, until one of the four experimental conditions reaches the same or similar number of target genes as GEM returns. Importantly, these are not necessarily the same genes, but just the same number of genes. We therefore retained GEM as the ‘upper bound’ on the stringency and tried to match it with the other peak detection methods. In the end, we decided on thresholds of $\text{SNR} > 1.5$ for maxPeak, $\text{SNR} > 1.123$ for loessPeak and $p < 0.0052$ for MACE. The number of genes retrieved using these thresholds is reported in Table 4.

System-Level Insight From the ‘Common Core’ of Retrieved Target Genes

After having reduced the number of ‘significant’ Fkh target genes as shown in Table 4, we may ponder how divergent are the genes identified by each of the four peak detection methods with regard to the target genes retrieved by all methods. In Fig. 7 Venn diagrams are shown that indicate the overlap of the number of target genes among the four peak detection methods, across the four different experiments.

Strikingly, only a small overlap is observed among the four peak detection methods. Even among the published tools, GEM and MACE, the number of target genes that are not retrieved by the other methods is remarkable. Apparently, even among the targets that show a relatively high signal there is a large variation among the four methods. A number of reasons may be considered for explaining the discrepancy: (i) GEM and MACE did have issues when analyzing the Fkh2 ChIP dataset; (ii) the four peak detection methods might recognize different signal signatures; (iii) the thresholds used might be not sufficiently stringent, i.e., when looking only at the top scoring targets in each method, these might retrieve the same genes. The latter point does not seem to hold, based on our attempts to increase the binding stringency further: We did not observe an increase in the ratio of overlapping genes to total genes reported as significant.

In Table 5 the genes retrieved as significant Fkh targets by the four peak detection methods are listed by their standard name (if available). We refer to these genes as the ‘common core’. Importantly, the known main target (i.e., positive control) of Fkh2, the *CLB2* gene, is among the genes within the common core, and it is retrieved by all peak detection methods in both logarithmic and stationary experimental conditions. Interestingly, the common core of Fkh1 targets includes a large subset of metabolic enzymes (10 out of 77), whereas only two metabolic enzymes (*PMA1* and *IDI1*) out of 19 are retrieved by all four methods for Fkh2. Four previous ChIP studies have reported target genes of both Fkh1 and Fkh2 (MacIsaac *et al.*, 2006; Hu *et al.*, 2007; Venters *et al.*, 2011; Ostrow *et al.*, 2014); we have taken this information into account, and we have indicated in Table 5, in red color, those target genes that have not been reported by these studies.

In a previous study 1082 genes have been identified as being cell cycle-regulated, and the time of their expression peak as well as the cell cycle phase where it occurs was reported for each such gene (Kudlicki *et al.*, 2007; Rowicka *et al.*, 2007). The common core of target genes can be verified for the presence of cell cycle regulated genes as well as for their functional annotation through the merge with GEMMER. In GEMMER, the GO (Gene Ontology) term annotations – which provide the logical structure of the biological functions (‘terms’) and their relationships to one another – are retrieved for each gene in SGD, and traced back through the hierarchical tree of GO terms to one of the following high-level GO terms: *Cellular metabolic process* (GO:0044237), *Cell cycle* (GO:0007049), *Cell division* (GO:0051301), *Signal transduction* (GO:0007165), and *DNA replication* (GO:0006260). These terms fall under the GO term *Cellular process*, with *DNA replication* falling under *Cellular metabolic process*. In GEMMER each such GO term annotation is assigned to one of the high-level terms listed above. For each gene, GEMMER adds up the number of annotations that fall under each high-level GO term. The GO term with the highest count is then assigned as the gene’s primary function.

In Fig. 8 all cell cycle-regulated genes in the common core (44 out of the 88 unique genes listed in Table 5) are displayed for all four experimental conditions, and grouped across the cell cycle phase where their expression peak occurs. It can be observed that the majority of the cell cycle-regulated genes that are Fkh targets fall within the GO terms *Metabolism*, suggesting their function especially in the S phase for DNA duplication, and primarily activated by Fkh1. A large number of the common core genes are Fkh1 targets, which exhibit their expression peak throughout all cell cycle phases, in particular in G1(P) (pre-replicative G1), G1/S, S and M phases. Conversely, Fkh2 targets are mostly concentrated in G2 and G2/M phases, such as *CLB2*. Several targets of Fkh1

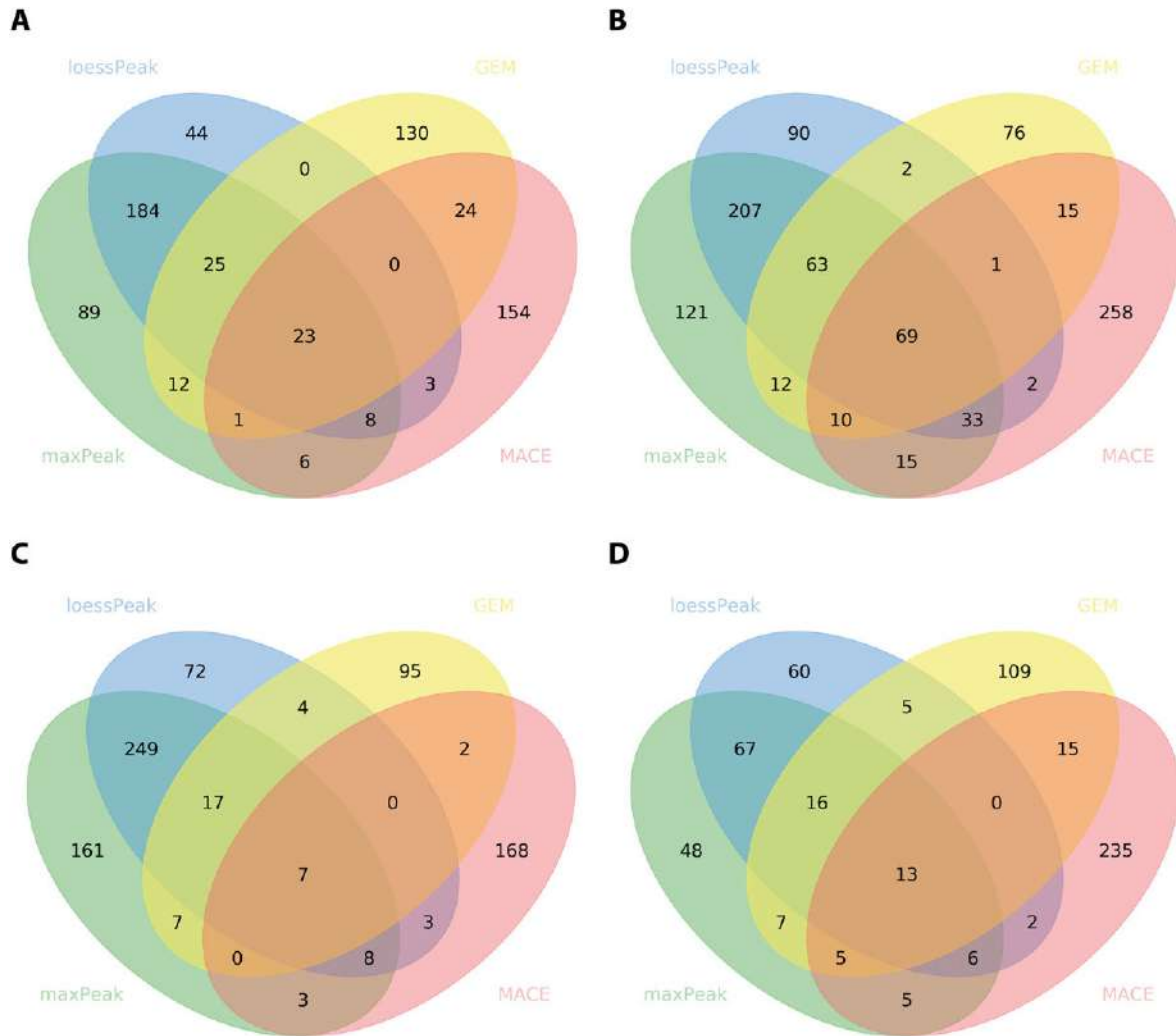


Fig. 7 4-way Venn diagrams of the overlap of the number of target genes among the four peak detection methods: maxPeak, loessPeak, GEM and MACE. (A) Fkh1 logarithmic phase. (B) Fkh1 stationary phase. (C) Fkh2 logarithmic phase. (D) Fkh2 stationary phase.

Table 5 Target genes in the ‘common core’, i.e., target genes retrieved by each of the four peak detection methods discussed in this work. Genes encoding a metabolic enzyme are indicated in bold. Genes that have not previously been observed as Fkh targets in previous ChIP studies are indicated in red color

Experiment	Target genes retrieved by maxPeak, loessPeak, GEM and MACE
Fkh1 logarithmic	ALG5, BRN1, DIN7, EGO2, ERS1 , ESP1, EXG1 , FLR1, FRK1 , HOS3, IDI1 , KIP2, NEW1, OSH7, PES4, RNR1 , RPS1B, SUN4, TEL2, TEM1, VIK1, YLR299C-A , YPL251W
Fkh1 stationary	ABP140, ADH4 , AIM46, ALG5, ARK1, BRN1, BUD3, BUD8, CIK1, CIT2 , CSI1, CSN9, DCC1, DIT1 , DIT2 , DSE2 , ECM10, EDC3, FHL1, FIN1, FIR1, FLR1, FRK1, GAS3, HOS3, HXT5 , IDI1 , KIP2, MKK2, MNT2, MUM3, NBL1, NEW1, NPP2, NUP2, NVJ3, OSH7, PDS5, PES4, PMA1 , RNR1 , RPL39, RPL8A, RPN10, RPN11, RPS1B, RPS22A, SAS3, SFG1 , SGO1, SPC24, TDA7, TEM1, TIM18, TMN2, TRS85, VIK1, VTI1, WTM1, YCS4, YGL007C-A, YGR111W, YLR334C, YMC2, YNL174W, YOR314W-A , YPI1, YPL251W, YPT31
Fkh2 logarithmic	CIS3, CLB2, ENV9, PMA1 , SRL1, YGL007C-A, YOR248W
Fkh2 stationary	ASE1, BUD4, CLB2, CLN1, HOS3, IDI1 , JSN1, KIP2, MTC6, OSH7, SFG1 , SPO12, YOR314W-A

and Fkh2 have their primary function in the cell cycle, e.g., *KIP2*, *CIK1*, *ESP1*, *VIK1* (Fkh1 targets) and *CLN1*, *CLB2* (Fkh2 targets). Noteworthy, a non-cell cycle target for Fkh1 is *RNR1*, the large subunit of the ribonucleoside-diphosphate reductase (RNR) that catalyzes the rate-limiting step in dNTP synthesis.

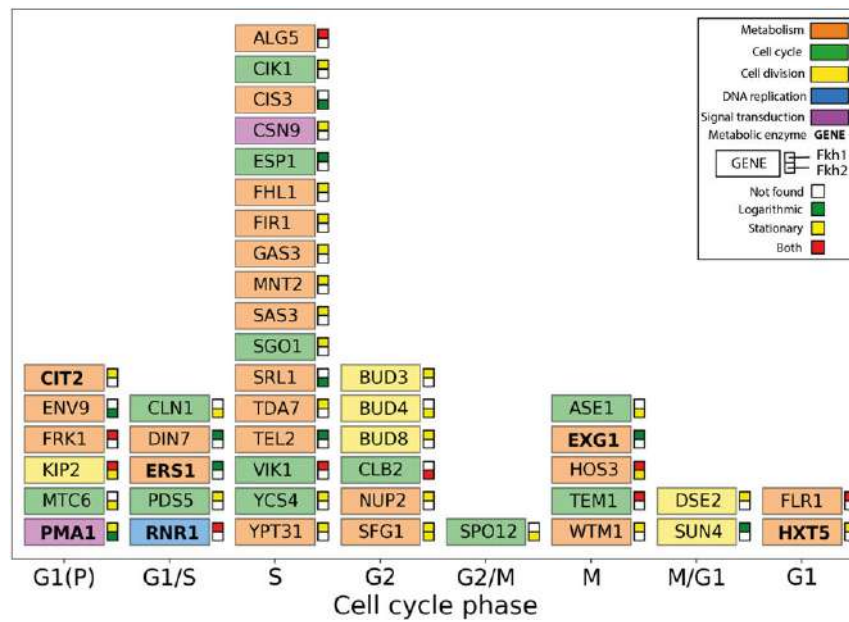


Fig. 8 Stack plot representing cell cycle phase of peak expression for the common core of Fkh target genes. Genes are retrieved by all four peak detection methods, for four experimental conditions, and are grouped according to the cell cycle phase of peak expression, as reported (Rowicka *et al.*, 2007). Each gene is indicated as a rectangle, and the colors of each rectangle indicate a specific gene function. The colors (white, red, yellow and green) of the smaller rectangles on the side of a gene indicate the experimental condition in which a gene was found as Fkh1 target (upper) and/or Fkh2 target (lower), respectively. Genes encoding a metabolic enzyme are indicated in bold. Vertical ordering is based on alphabet of the standard gene name.

Table 6 Enrichment of global function of Fkh target genes compared to the set of all protein-coding genes (genome-wide %). The percentages in the last four columns (Fkh1 logarithmic, Fkh1 stationary, Fkh2 logarithmic and Fkh2 stationary) are calculated with respect to the “Genome-wide %” column. The genes considered for the last four columns are considered to be significant at least by *one* of the four peak detection methods for the chosen thresholds (see text)

Functional category	Genome-wide %	Fkh1 logarithmic %	Fkh1 stationary %	Fkh2 logarithmic %	Fkh2 stationary %
Metabolism	75.65	−6.90	−6.21	−6.74	−6.93
Cell cycle	11.99	1.33	0.85	0.71	2.69
Signal transduction	6.85	−0.66	−0.57	−1.14	−1.58
Cell division	4.54	0.66	−0.38	1.22	1.22
DNA replication	0.96	−0.4	0.26	−0.34	−0.13

Exploration of Unique Sets of Fkh Targets Unravels Regulatory Links With Metabolism

The relatively small number of common core target genes retrieved by all four peak detection methods provides an insufficiently large dataset to systematically compare enrichment in function and timing. Therefore, we have considered the larger set of genes that, for each independent experiment and experimental condition, is retrieved as a Fkh target using the stringent thresholds by at least *one* method. In other words, this set of genes is the set of unique genes indicated in Fig. 7, for each Fkh and for each experimental condition: 703 (Fkh1 logarithmic), 974 (Fkh1 stationary), 796 (Fkh2 logarithmic) and 593 (Fkh2 stationary). Subsequently, we calculated the enrichment of functions of target genes relative to the whole genome (see Table 6).

From the analysis, we have observed that both Fkh1 and Fkh2 show an increased percentage of target genes that fall within the GO terms *Cell cycle* and *Cell division*, both in logarithmic and stationary phases, with the exception of Fkh1 in stationary phase. This result supports the earlier finding that Fkh targets are primarily cell cycle genes (Spellman *et al.*, 1998). Interestingly, we observed that a fraction of genes with a metabolic function is present among the targets. Specifically, we identified 73 and 101 metabolic enzymes targets of Fkh1 and Fkh2, respectively, in logarithmic phase, and 101 and 71 metabolic enzymes, respectively, in stationary phase. This provides a clear indication of the potential role of Fkh as hubs connecting cell cycle and metabolism.

Similarly to the calculated functional enrichment shown in Table 6, we calculated the enrichment of cell cycle-regulated target genes that peak in specific cell cycle phases (Fig. 9). When comparing the distributions of the identified target genes in our four ChIP-exo experiments to the genome-wide distribution (Kudlicki *et al.*, 2007; Rowicka *et al.*, 2007), we observed that Fkh1 targets are shifted mainly towards S, G2 and M/G1, and away from G1/P, G1/S and M, in both logarithmic and stationary phases. There are instead mixed results for G1 and G2/M. Fkh2 targets are shifted mainly towards G2, G2/M and M/G1, and away from G1, G1(P), G1/S and M, in both logarithmic and stationary phases. There are instead mixed results for S phase. Between G2 and M/G1, Fkh2 has an

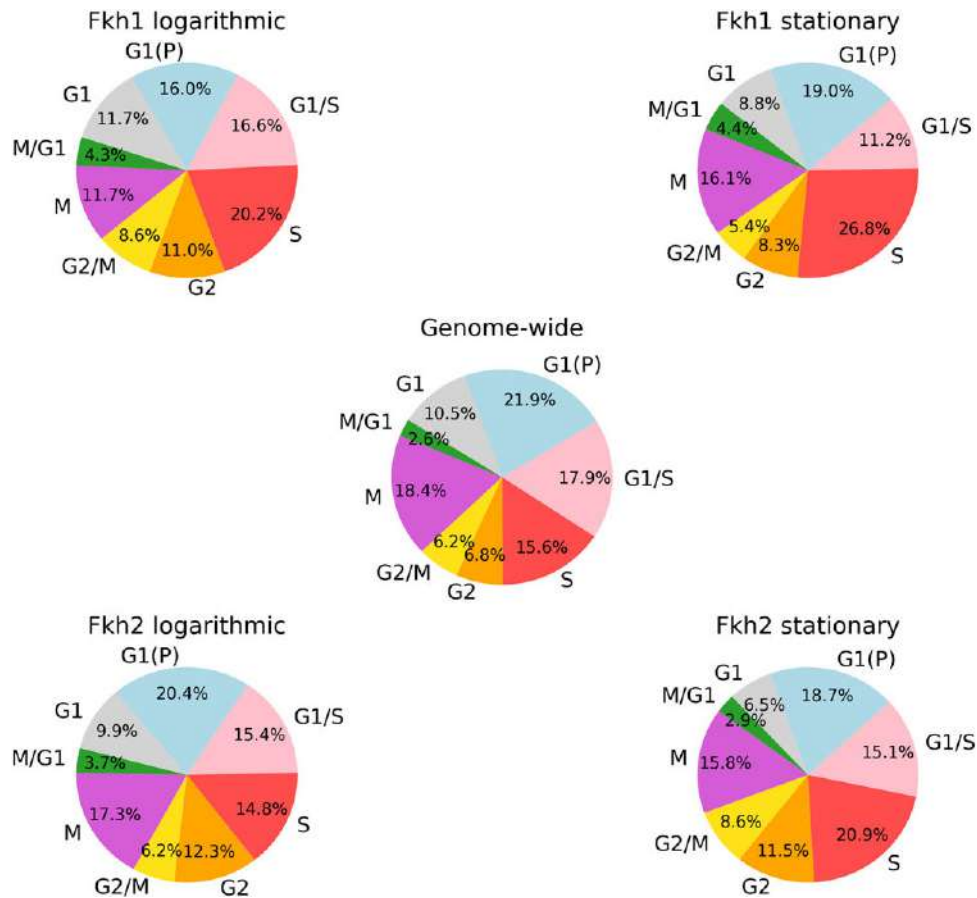


Fig. 9 Distribution of the expression peak for cell cycle-regulated genes. A genome-wide dataset was compared to the ChIP-exo Fkh dataset to identify target genes that are cell cycle-regulated. The distribution of the cell cycle-regulated genes (Rowicka *et al.*, 2007) is shown in the center, whereas the other four pies show the distribution of Fkh1 and Fkh2 targets, in both logarithmic and stationary phases. The set of genes considered for the latter four pies is the subset of cell cycle-regulated genes that are retrieved as being significant at least by *one* of the four peak detection methods.

enrichment of 5.5% and 4.8% in target genes, in logarithmic and stationary phase, respectively, whereas Fkh1 of 1.6% and 0.2%, respectively, consistent with data showing that Fkh are mainly expressed during late cell cycle phases (Lee *et al.*, 2002).

In order to visualize the cell cycle-regulated target genes of Fkh1 and Fkh2 with transcription peaks across different cell cycle phases, stack plots were generated for the larger sets of retrieved significant genes, similarly to the information presented in Fig. 8. As an example, we show here the result for Fkh2 in logarithmic phase (Fig. 10). It becomes apparent that the majority of target genes, for this experimental condition, falls primarily within the GO terms *Metabolism*, followed by *Cell cycle* and *Cell division*. This evidence further supports that the Fkh-dependent transcriptional program may be also activate metabolic pathways to guarantee a timely cell cycle progression.

To investigate the network effects of the Fkh targets, these have been annotated to include the list of KEGG Pathways each gene is mapped on. As an example, we visualize the target genes of both Fkh1 and Fkh2 in central carbon metabolism, which enzymes are retrieved as significant by each of the four peak detection methods (Fig. 11). Alternatively, by using the Pathview library for R (Luo and Brouwer, 2013), the target genes may be superimposed on KEGG maps in an automated way. Strikingly, identified reliable target genes of Fkh1 in central carbon metabolism are *HXT5*, *ADH4* and *CIT2*; conversely, Fkh2 generally lacks potential reliable targets, based on our data.

Discussion

As compared to previous ChIP-based methodologies, which have been conducted in logarithmic phase, our ChIP-exo analyses were performed in two experimental conditions, logarithmic and stationary phases. In either or both conditions we observed that targets of Fkh1, but not Fkh2, are mainly found in metabolic processes (see Fig. 11), with a substantial number of metabolic enzymes (see Table 5 and Fig. 8). Therefore, we were interested to identify which specific KEGG Pathways are populated by the common core genes, which are retrieved as significant by each of the four peak detection methods used in this study.

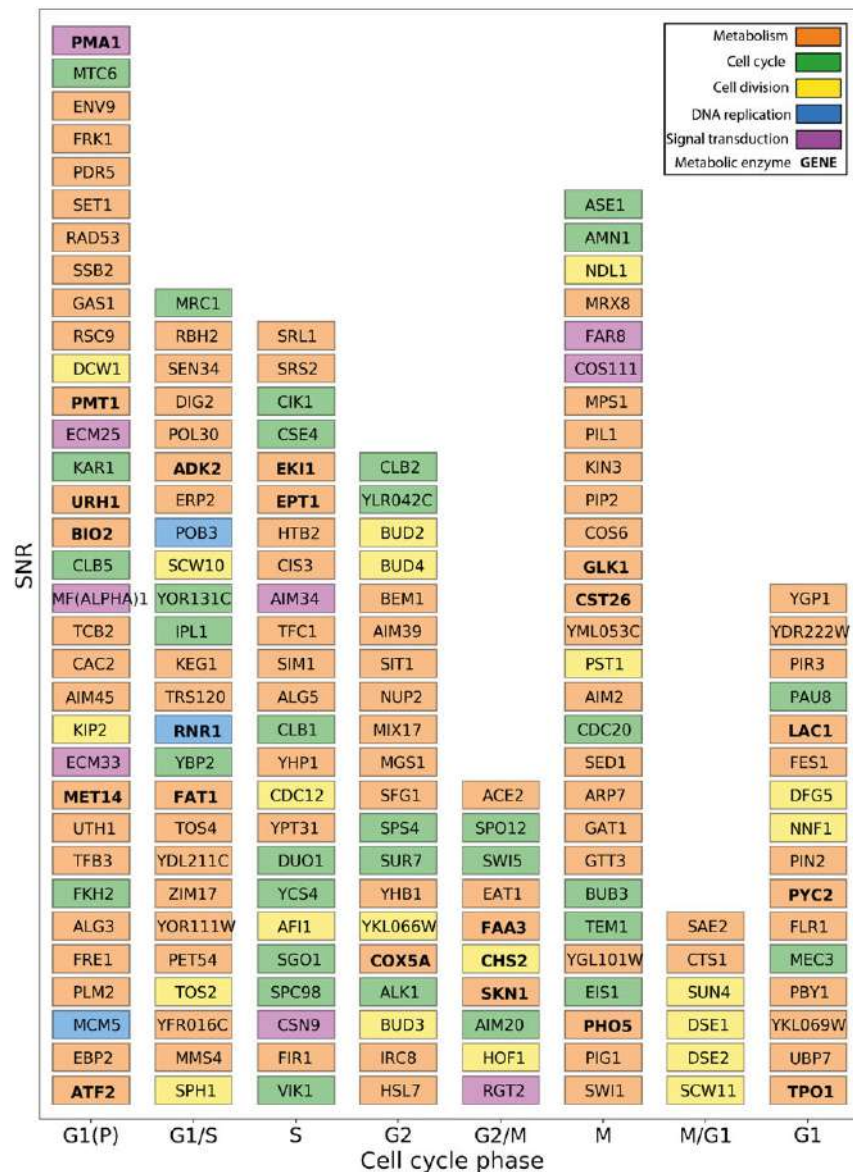


Fig. 10 Stack plot representing the Fkh2 target genes in logarithmic phase retrieved by any of the four peak detection methods. Genes are grouped according to their cell cycle phase of peak expression (Rowicka et al., 2007). Vertical ordering is based on the SNR of target genes in the maxPeak dataset. Each gene is colored according to its specific function. Genes encoding a metabolic enzyme are indicated in bold.

In Table 7 the Fkh target genes retrieved by all methods for each experimental condition were categorized according to the KEGG Pathways they occur in. A large difference between the target genes of Fkh1 and Fkh2 may be observed, due to the larger number of Fkh1 targets considered as significant. Fkh1 appears to target many different cellular processes including fatty acid metabolism, purine and pyrimidine metabolism, and central carbon metabolism. Strikingly, four Fkh1 target genes relate to the ribosome (*RPL39*, *RPL8A*, *RPS1B*, *RPS22A*). *EXG1*, a novel Fkh1 target never reported before in the literature, is a metabolic enzyme, i.e. exo-1,3-beta-glucanase, involved in cell wall beta-glucan assembly.

In the smaller set of target genes retrieved for Fkh2, only one connection with metabolism is observed. This regards the metabolic enzyme *PMA1*, plasma membrane ATPase which pumps protons out of the cell, and major regulator of cytoplasmic pH and plasma membrane potential. This gene has been previously reported as a Fkh target by a ChIP-chip study (Ostrow et al., 2014). However, the other Fkh2 target genes are functional to cell cycle and signaling (*CLN1*, *CLB2*, *SPO12*), consistently with the current view of the Fkh2 role in cell cycle progression. This evidence indicates that Fkh1 and Fkh2 appear to have divergent functions, and that these TFs may serve as hubs that integrate multi-scale regulatory networks to achieve proper timing of cell division.

The latter aspect is particularly relevant, as a number of biochemical reactions requires to be completed for a timely cell division. Coordination of these reactions requires cyclin/Cdk1 kinase complexes, which are activated upon sequential transcription of cyclin genes with a characteristic staggered behavior known as waves of cyclins (Nasmyth, 1993; Futcher, 1996). Progressive

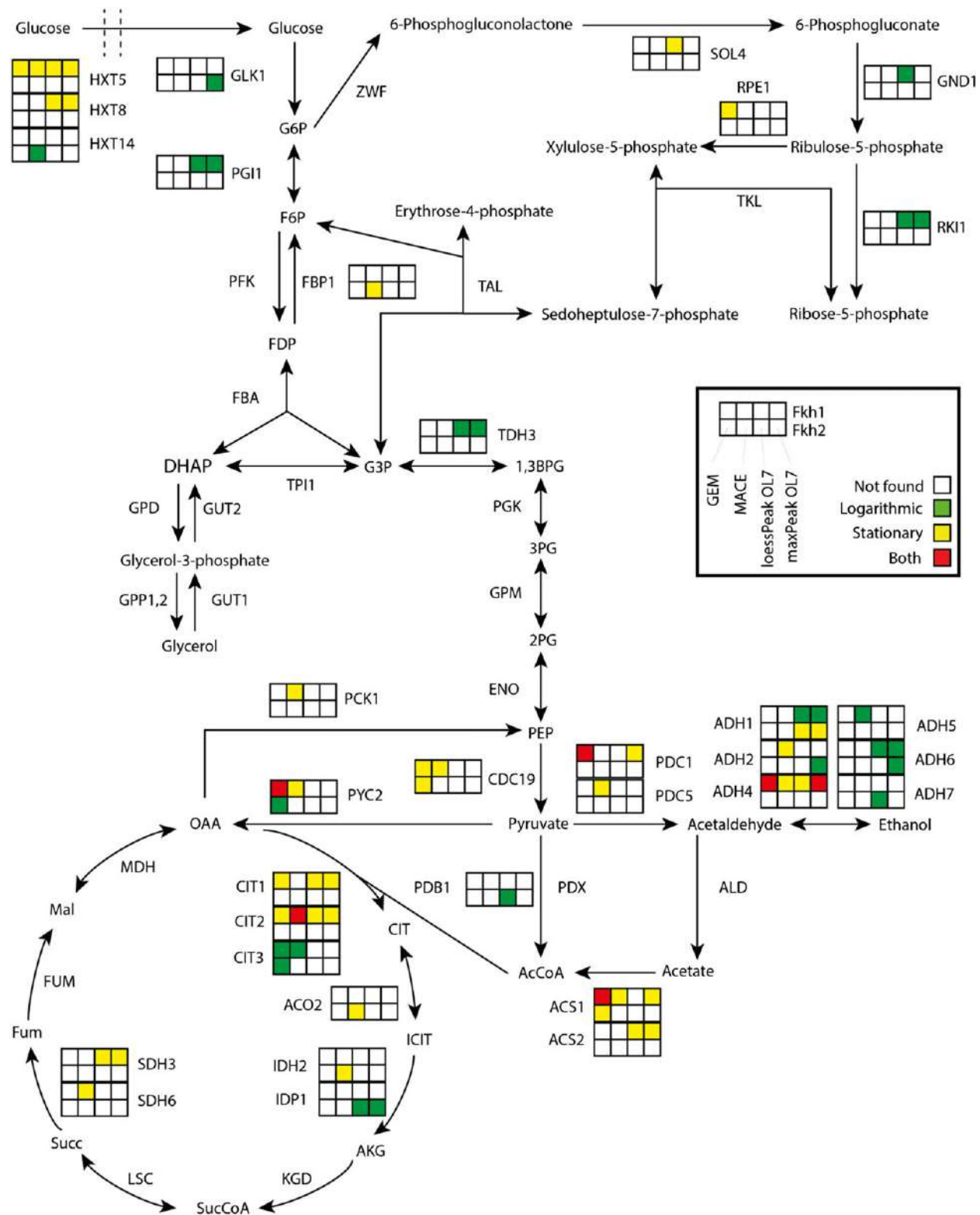


Fig. 11 Overview of metabolic enzymes in central carbon metabolism that are targets of Fkh1 and Fkh2. Each enzyme is associated with eight squares divided in two rows (Fkh1, top row; Fkh2, bottom row) representing data analysis with four peak detection methods. Empty squares indicate that the gene was not retrieved as a significant target, whereas colored squares indicated positive evidence. A distinction between the results in logarithmic and stationary phases is visualized through the color of the squares (see the figure insert). Isoenzymes that have no available evidence in any of the four studies were neglected. In some cases, metabolic enzymes may have no associated squares when no isoenzyme is available with an experimental validation. In this latter case, we sometimes referred to a generic name of the isoenzyme, e.g., *GPM* as opposed to *GPM1*.

Table 7 Categorization according to KEGG Pathway occurrence of Fkh target genes retrieved to be significant by all four peak detection methods. Genes encoding a metabolic enzyme are indicated in bold. Genes that have not previously been observed as Fkh targets in previous ChIP studies (MacIsaac *et al.*, 2006; Hu *et al.*, 2007; Venters *et al.*, 2011; Ostrow *et al.*, 2014) are indicated in red color. The KEGG Pathways listed are not complete, but rather a representative subset of pathways that contain at least one of the Fkh target genes

<i>Kegg pathways</i>	<i>Fkh1 logarithmic</i>	<i>Fkh1 stationary</i>	<i>Fkh2 logarithmic</i>	<i>Fkh2 stationary</i>
Cell cycle – Yeast	BRN1, ESP1, TEM1	BRN1, TEM1, YCS4	CLB2	CLB2, CLN1, SPO12
MAPK signaling pathway – Yeast		MKK2	CLB2	CLB2, CLN1
Meiosis – Yeast	ESP1	HXT5, SGO1		SPO12
Citrate cycle (TCA cycle)		CIT2		
Glycolysis/Gluconeogenesis		ADH4		
Glutathione metabolism	RNR1	RNR1		
Oxidative phosphorylation		PMA1	PMA1	
Starch and sucrose metabolism	EXG1			
Mitophagy – Yeast		MKK2		
Ribosome	RPS1B	RPL39, RPL8A, RPS1B, RPS22A		
Purine metabolism	RNR1	RNR1		
Pyrimidine metabolism	RNR1	RNR1		
Terpenoid backbone biosynthesis	IDI1	IDI1		IDI1
Endocytosis		YPT31		
N-Glycan biosynthesis	ALG5	ALG5		
Fatty acid degradation		ADH4, DIT2		
RNA degradation		EDC3		
Proteasome		RPN10, RPN11		

oscillations of cyclin levels, thereby of cyclin/Cdk1 activity, ensure a robust timing of cell cycle progression and of transcriptional regulation. Fkh modulate transcription of cyclin genes in late cell cycle phases (Breedon, 2000; Jorgensen and Tyers, 2000), in particular of Clb2 and Clb3 (Koranda *et al.*, 2000; Kumar *et al.*, 2000; Pic *et al.*, 2000; Reynolds *et al.*, 2003; Linke *et al.*, 2017). Considering the relevance of Fkh in the progressive activation of the cyclin/Cdk1 complexes, we have explored the relevance of our ChIP-exo findings for Fkh1 and Fkh2 on cell cycle dynamics (Fig. 12).

In Fig. 12(A) the regulatory cascade driving cell cycle progression is shown: In G1 phase, the cyclin Cln2 (and Cln1), together with the kinase Cdk1, inhibits the cyclin/Cdk1 inhibitor Sic1. When Clb5/Cdk1 is released from Sic1 inhibition, it promotes DNA duplication in S phase. Consequently, a Clb/Cdk1 cascade is activated, involving waves of Clb5, Clb3 and Clb2 cyclins (all bound to Cdk1). In Fig. 12(B) the evidence of Fkh binding at promoters of target genes is highlighted in the Clb/Cdk1 cascade. In our ChIP-exo study, CLB2 appears to be the major target of Fkh2, as reported (Sherriff *et al.*, 2007; Reynolds *et al.*, 2003; Pic *et al.*, 2000; Kumar *et al.*, 2000; Koranda *et al.*, 2000; Hollenhorst *et al.*, 2000, 2001), in both logarithmic and stationary phases. SWI5 is also confirmed to be Fkh2 target (Zhu *et al.*, 2000; Pic *et al.*, 2000; Kumar *et al.*, 2000) as well as FKH2 (although only when using the loessPeak peak detection method), as reported by ChIP-chip and ChIP-seq, genome-wide studies (Venters *et al.*, 2011; Ostrow *et al.*, 2014; MacIsaac *et al.*, 2006).

In addition, our findings of FKH2 being target of Fkh1 and of CLB5 being target of Fkh2 are supported by a previous ChIP-chip study (Ostrow *et al.*, 2014). An interesting scenario regards the CLB3 gene. Fkh2 binding to CLB3 promoter was shown by a ChIP-chip study (Ostrow *et al.*, 2014). Furthermore, we have recently demonstrated that Fkh2 binds to the CLB3 promoter and regulates Clb3 expression, thus synchronizing the temporal expression of mitotic CLB genes (Linke *et al.*, 2017). However, in our ChIP-exo data for Fkh2, CLB3 has a SNR below threshold in stationary phase and it shows no signal in logarithmic phase when using the OL7 DNA strand approach. Interestingly, when the maxPeak detection method is used (with either AD7 or AD10), SNR values between ~0.62–1 were observed for the binding of Fkh1 and Fkh2 in both logarithmic and stationary phases, whereas using the loessPeak method with AD7 for CLB3 results in a SNR=1.003 (above noise but not above stringent threshold) for the binding of Fkh1 in stationary phase. Thus, although the SNR values indicate relatively low levels of binding of Fkh to the CLB3 promoter, we can conclude that CLB3 identification as target is sensitive to the DNA strand approach used. This specific case indicates the possible limitation of the peak detection methods when analyzing ChIP-exo data, which are not able to retrieve CLB3 although a positive evidence from an independent experimental validation exists.

Further analyses are required in order to validate the connection between Fkh and a number of metabolic pathways, and to investigate the specific role of Fkh in the Clb/Cdk1 cascade. However, our findings highlight possible novel regulatory mechanisms contributing to cell's viability, which impact on proper cell cycle dynamics. Thus, our study provides evidence of the role of Fkh as hubs connecting multi-scale cellular networks in the budding yeast cell.

Future Directions

The study that we have presented highlights the divergence in retrieving ChIP data by using a number of data analysis methods, and stimulates further research to (i) identify the more stable method(s), (ii) investigate in depth the advantages and shortcomings of each method, and (iii) measure the accuracy of the data analysis methods with regard to identifying functional targets.

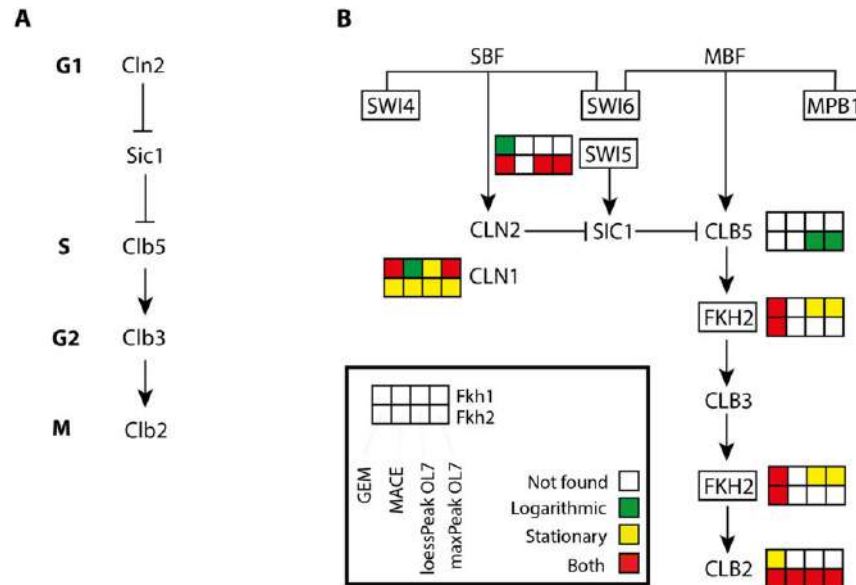


Fig. 12 Fkh1 and Fkh2 target genes in the molecular cascade regulating dynamics of cell cycle progression. (A) Molecular players driving cell cycle progression (see text for details). (B) Overview of cell cycle regulators that are Fkh targets. The transcription factors *SWI4*, *SWI6*, *MPB1*, *SWI5* and *FKH2* are shown within rectangles. Genes without associated rectangles were not identified as significant Fkh targets in the various dataset.

Closing Remarks

Various ChIP-based methodologies exist, with ChIP-exo being the newest instantiation. In this study, we have presented the main differences between ChIP-exo and currently in-use ChIP techniques, and discussed two existing (GEM and MACE) and two novel (maxPeak and loessPeak) methods for analyzing ChIP-exo data. We have described a pipeline to be followed when analyzing the data, and applied it to two Forkhead (Fkh) transcription factors (Fkh1 and Fkh2) that regulate cell division in budding yeast. The result of our analysis indicates a large diversity of the target genes retrieved by the four methods. This may be explained, in part, by failure of the existing algorithms (GEM and MACE) to properly analyze the data considered here, and may point to the fact that these approaches might be less reliable when the number of high-quality peaks is relatively low. However, importantly, a small common set of retrieved target genes exists, pointing to an accurate, reliable output that is insensitive to the method used.

In this study, we have also highlighted how to integrate ChIP data with existing literature data from a number of databases containing information of genes in budding yeast, such as functional annotation, localization, abundance, and both timing and cell cycle phase of peak occurrence of RNA transcript levels. This integration has been made possible through GEMMER, a web-based visualization tool that we have recently developed.

Altogether, we have shown that our pipeline of analyzing ChIP data and integrating these with literature information allows for predicting and understanding the function of transcription factors in a multi-scale, networked, cellular system.

Author Contributions

M.B. conceived and designed the study. P.H., T.D.G.A.M., J.N. and M.B. designed the experimental analysis, which was performed by P.H.. M.B. and T.D.G.A.M. designed the computational analysis, which was performed by T.D.G.A.M.. T.D.G.A.M. and M.B. analyzed the data. M.B. provided the biological interpretation of data. P.H., T.D.G.A.M. and M.B. wrote the paper, with contribution from J.N.. M.B. provided scientific leadership and supervised the study. Petter Holland and Thierry DGA Mondeel contributed equally to this work.

Acknowledgement

This work was supported by the Swammerdam Institute for Life Science Starting Grant of the University of Amsterdam to M.B., and by the Novo Nordisk Foundation, Vetenskapsrådet, and Knut and Alice Wallenberg Foundation to J.N.

See also: Computational Systems Biology Applications. Natural Language Processing Approaches in Bioinformatics

References

- Arrieta-Ortiz, M.L., Hafemeister, C., Bate, A.R., *et al.*, 2015. An experimentally supported model of the *Bacillus subtilis* global transcriptional regulatory network. *Molecular Systems Biology* 11, 839.
- Bähler, J., 2005. Cell-cycle control of gene expression in budding and fission yeast. *Annual Review of Genetics* 39, 69–94.
- Breeden, L.L., 2000. Cyclin transcription: Timing is everything. *Current Biology* 10, R586–R588.
- Castiglione, F., Pappalardo, F., Bianca, C., Russo, G., Motta, S., 2014. Modeling biology spanning different scales: An open challenge. *BioMed Research International* 2014, 902545.
- Cho, R.J., Campbell, M.J., Winzler, E.A., *et al.*, 1998. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell* 2, 65–73.
- de Boer, C.G., Hughes, T.R., 2012. YeTFaSCO: A database of evaluated yeast transcription factor sequence specificities. *Nucleic Acids Research* 40, D169–D179.
- Fang, X., Sastry, A., Mih, N., *et al.*, 2017. Global transcriptional regulatory network for *Escherichia coli* robustly connects gene expression to transcription factor activities. *Proceedings of the National Academy of Sciences of the United States of America* 114, 10286–10291.
- Futcher, B., 1996. Cyclins and the wiring of the yeast cell cycle. *Yeast* 12, 1635–1646.
- Guo, Y., Mahony, S., Gifford, D.K., 2012. High resolution genome wide binding event finding and motif discovery reveals transcription factor spatial binding constraints. *PLOS Computational Biology* 8, e1002638.
- Haase, S.B., Wittenberg, C., 2014. Topology and control of the cell-cycle-regulated transcriptional circuitry. *Genetics* 196, 65–90.
- Harbison, C.T., Gordon, D.B., Lee, T.I., *et al.*, 2004. Transcriptional regulatory code of a eukaryotic genome. *Nature* 431, 99–104.
- Hollenhorst, P.C., Bose, M.E., Mielke, M.R., *et al.*, 2000. Forkhead genes in transcriptional silencing, cell morphology and the cell cycle. Overlapping and distinct functions for FKH1 and FKH2 in *Saccharomyces cerevisiae*. *Genetics* 154, 1533–1548.
- Hollenhorst, P.C., Pietz, G., Fox, C.A., 2001. Mechanisms controlling differential promoter-occupancy by the yeast forkhead proteins Fkh1p and Fkh2p: Implications for regulating the cell cycle and differentiation. *Genes and Development* 15, 2445–2456.
- Hu, Z., Killion, P.J., Iyer, V.R., 2007. Genetic reconstruction of a functional transcriptional regulatory network. *Nature Genetics* 39, 683–687.
- Jorgensen, P., Tyers, M., 2000. The forked path to mitosis. *Genome Biology* 1, 1022.1–1022.4.
- Knott, S.R., Peace, J.M., Ostrow, A.Z., *et al.*, 2012. Forkhead transcription factors establish origin timing and long-range clustering in *S. cerevisiae*. *Cell* 148, 99–111.
- Koranda, M., Schleiffer, A., Endler, L., Ammerer, G., 2000. Forkhead-like transcription factors recruit Ndd1 to the chromatin of G2/M-specific promoters. *Nature* 406, 94–98.
- Kudlicki, A., Rowicka, M., Otwinowski, Z., *et al.*, 2007. SCEPTRANS: An online tool for analyzing periodic transcription in yeast. *Bioinformatics* 23, 1559–1561.
- Kumar, R., Reynolds, D.M., Shevchenko, A., Goldstone, S.D., Dalton, S., 2000. Forkhead transcription factors, Fkh1p and Fkh2p, collaborate with Mcm1p to control transcription required for M-phase. *Current Biology* 10, 896–906.
- Landt, S.G., Marinov, G.K., Kundaje, A., *et al.*, 2012. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Research* 22, 1813–1831.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nature Methods* 9, 357–359.
- Lee, T.I., Rinaldi, N.J., Robert, F., *et al.*, 2002. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* 298, 799–804.
- Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25, 2078–2079.
- Linke, C., Chasapi, A., González-Novo, A., *et al.*, 2017. A Clb/Cdk1-mediated regulation of Fkh2 synchronizes CLB expression in the budding yeast cell cycle. *Nature Partner Journals Systems Biology and Applications* 3, 7.
- Luo, W., Brouwer, C., 2013. Pathview: An R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics* 29, 1830–1831.
- MacIsaac, K.D., Wang, T., Gordon, D.B., *et al.*, 2006. An improved map of conserved regulatory sites for *Saccharomyces cerevisiae*. *BMC Bioinformatics* 7, 113.
- McInerney, C.J., 2011. Cell cycle regulated gene expression in yeasts. *Advances in Genetics* 73, 51–85.
- Mondeel, T.D.G.A., Crémazy, F., Barberis, M., 2018. GEMMER: GENome-wide tool for Multi-scale Modeling data Extraction and Representation for *Saccharomyces cerevisiae*. *Bioinformatics* bty052, <https://doi.org/10.1093/bioinformatics/bty052>.
- Nasmyth, K., 1993. Control of the yeast cell cycle by the Cdc28 protein kinase. *Current Opinion in Cell Biology* 5, 166–179.
- Orlando, D.A., Lin, C.Y., Bernard, A., *et al.*, 2008. Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature* 453, 944–947.
- Ostrow, A.Z., Kalhor, R., Gan, Y., *et al.*, 2017. Conserved forkhead dimerization motif controls DNA replication timing and spatial organization of chromosomes in *S. cerevisiae*. *Proceedings of the National Academy of Sciences of the United States of America* 114, E2411–E2419.
- Ostrow, A.Z., Nellimoottil, T., Knott, S.R., *et al.*, 2014. Fkh1 and Fkh2 bind multiple chromosomal elements in the *S. cerevisiae* genome with distinct specificities and cell cycle dynamics. *PLOS ONE* 9, e87647.
- Papantonis, A., Cook, P.R., 2013. Transcription factories: Genome organization and gene regulation. *Chemical Reviews* 113, 8683–8705.
- Park, D., Morris, A.R., Battenhouse, A., Iyer, V.R., 2014. Simultaneous mapping of transcript ends at single-nucleotide resolution and identification of widespread promoter-associated non-coding RNA governed by TATA elements. *Nucleic Acids Research* 42, 3736–3749.
- Park, P.J., 2009. ChIP-seq: Advantages and challenges of a maturing technology. *Nature Reviews Genetics* 10, 669–680.
- Peace, J.M., Villwock, S.K., Zeytounian, J.L., Gan, Y., Aparicio, O.M., 2016. Quantitative BrdU immunoprecipitation method demonstrates that Fkh1 and Fkh2 are rate-limiting activators of replication origins that reprogram replication timing in G1 phase. *Genome Research* 26, 365–375.
- Pic, A., Lim, F.L., Ross, S.J., *et al.*, 2000. The forkhead protein Fkh2 is a component of the yeast cell cycle transcription factor SFF. *EMBO Journal* 19, 3750–3761.
- Reynolds, D., Shi, B.J., McLean, C., *et al.*, 2003. Recruitment of Thr319-phosphorylated Ndd1p to the FHA domain of Fkh2p requires Clb kinase activity: A mechanism for CLB cluster gene activation. *Genes and Development* 17, 1789–1802.
- Rhee, H.S., Pugh, B.F., 2011. Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution. *Cell* 147, 1408–1419.
- Rhee, H.S., Pugh, B.F., 2012. Genome-wide structure and organization of eukaryotic pre-initiation complexes. *Nature* 483, 295–301.
- Robertson, G., Hirst, M., Bainbridge, M., *et al.*, 2007. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods* 4, 651–657.
- Rowicka, M., Kudlicki, A., Tu, B.P., Otwinowski, Z., 2007. High-resolution timing of cell cycle-regulated gene expression. *Proceedings of the National Academy of Sciences of the United States of America* 104, 16892–16897.
- Sheriff, J.A., Kent, N.A., Mellor, J., *et al.*, 2007. The Isw2 chromatin-remodeling ATPase cooperates with the Fkh2 transcription factor to repress transcription of the B-type cyclin gene CLB2. *Molecular and Cellular Biology* 27, 2848–2860.
- Solomon, M.J., Larsen, P.L., Varshavsky, A., 1988. Mapping protein-DNA interactions in vivo with formaldehyde: Evidence that histone H4 is retained on a highly transcribed gene. *Cell* 53, 937–947.
- Spellman, P.T., Sherlock, G., Zhang, M.Q., *et al.*, 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell* 9, 3273–3297.
- Terroate, T.W., Pozner, A., Buck-Koehntop, B.A., 2016. PATCh-Cap: Input strategy for improving analysis of ChIP-exo data sets and beyond. *Nucleic Acids Research* 44, e159.
- Tessarz, P., Kouzarides, T., 2014. Histone core modifications regulating nucleosome structure and dynamics. *Nature Reviews Molecular Cell Biology* 15, 703–708.

- Venters, B.J., Wachi, S., Mavrich, T.N., *et al.*, 2011. A comprehensive genomic binding map of gene and chromatin regulatory proteins in *Saccharomyces*. *Molecular Cell* 41, 480–492.
- Wang, L., Chen, J., Wang, C., *et al.*, 2014. MACE: Model based analysis of ChIP-exo. *Nucleic Acids Research* 42, e156.
- Wittenberg, C., Reed, S.I., 2005. Cell cycle-dependent transcription in yeast: Promoters, transcription factors, and transcriptomes. *Oncogene* 24, 2746–2755.
- Zhu, G., Spellman, P.T., Volpe, T., *et al.*, 2000. Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature* 406, 90–94.

Relevant Websites

<http://gemmer.barberislab.com/>
GEMMER.

https://downloads.yeastgenome.org/sequence/S288C_reference/genome_releases/
Reference genome sequence R.64.1.1.

Studies of Body Systems

Amir Feisal Merican, University of Malaya, Kuala Lumpur, Malaysia

Hoda Mirsafian, University of Malaya, Kuala Lumpur, Malaysia and University of Southern California, Los Angeles, CA, United States

Adiratna Mat Ripen, Institute of Medical Research, Kuala Lumpur, Malaysia

Saharuddin Bin Mohamad, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The human body is made up of a combination of specialized biological systems with specific functions. These body systems represent groups of organs, tissues, and cells that act together to perform various tasks for growth, reproduction, and survival of the body. Some organs may be part of more than one body system, if they serve more than one function. The systems work and interact with each other to form a functioning human body. The human body system comprises the skeletal system (bones, cartilage, ligaments), muscular system (cardiac or heart muscle, skeletal muscle, smooth muscle, tendons), nervous system (brain, spinal cord, nerves), respiratory system (trachea, larynx, pharynx, lungs), lymphatic system (lymph nodes, lymph vessels), immune system (bone marrow, spleen, white blood cells), endocrine system (pituitary gland, hypothalamus, adrenal glands, thyroid, parathyroids, pancreas, ovaries, testes), cardiovascular or circulatory system (heart, blood vessels, blood), digestive or excretory system (esophagus, stomach, small intestine, large intestine), urinary or renal system (kidneys, urinary bladder), integumentary or exocrine system (skin, hair, nails), and reproductive system (female: uterus, vagina, fallopian tubes, ovaries; male: penis, testes, seminal vesicles) (McDowell, 2010; Taylor, 2017). Although these body systems are all very different, they do have one thing in common, of which all of them begin with cells. During the process of cellular differentiation, a cell changes from one cell type to another with specialized function forming tissues and organs. Different cell types express characteristic sets of proteins that are encoded by respective genes within a cell's DNA (Slack, 2013).

Cells can dynamically access and translate specific information through gene expression by selectively switching on and off particular genes. In the selected genes, the information encoded is transcribed into RNA molecules, which consequently can be translated into proteins or can be directly used to control gene expression (Finotello and Di Camillo, 2015). The investigation and analysis of any individual dysregulations occurring within the genomic, transcriptomic, proteomic, and metabolomic levels could utilize advanced high-throughput technologies (Likić *et al.*, 2010; Ayers and Day 2015; Krauss *et al.*, 2017).

The transcriptome is a set of all RNA transcripts existing in a cell or tissue at a certain point of time under specific conditions (Sirri *et al.*, 2008). The transcriptome encompasses multiple types of coding and noncoding RNA species. Thus, transcriptomic study is essential to identify the current state of the cells and fundamental pathogenic mechanisms of diseases. In addition, differential gene expression study facilitates the comparison of gene expression profiles from different cells and conditions to characterize genes that are responsible in the determination of phenotypes. For example, the comparison of healthy versus diseased cells can present new insights on genetic aspects involved in pathology (Finotello and Di Camillo, 2015). Previously, microarray has been the most important and commonly used method for transcriptome analysis (Baldi and Hatfield, 2002), however in recent years, high-throughput RNA sequencing (RNA-Seq) has become a powerful alternative approach for transcriptome profiling studies. RNA-Seq is able to qualitatively and quantitatively investigate any RNA molecules including messenger RNAs (mRNAs), long noncoding RNAs (lncRNAs), microRNAs (miRNAs), and small interfering RNAs (siRNA) (Dong and Chen, 2013, Kukurba and Montgomery, 2015).

The immune system is one of the most complex body systems that the human has. The immune system comprises the innate and the adaptive systems. Monocytes, a type of leukocyte or white blood cells, are key elements of the innate immune-mediated processes that become the first line of defense response against pathogens (Janeway, 2001). They can differentiate into macrophages and myeloid lineage dendritic cells. They are mononuclear cells and play a central role in the clearance of microbial infections and cellular debris, secretion of immunoregulatory bioactive factors including interleukins, interferons, chemokines, and growth factors, and control of cancer progression (Kraft-Terry and Gendelman, 2011).

Recent studies have applied RNA-Seq technology for transcriptome profiling of several tissues and cell types such as endometrium (Zieba *et al.*, 2015), spleen (Dang *et al.*, 2016), T cells (Mitchell *et al.*, 2015), B cells (Toung *et al.*, 2011), and macrophages (Beyer *et al.*, 2012). In addition, by using the RNA-Seq approach, the global gene transcription changes that occur during the differentiation of monocyte to macrophage have been reported (Ancuta *et al.*, 2009; Dong *et al.*, 2013; Ziegler-Heitbrock *et al.*, 2010). Furthermore, the RNA-Seq data are valuable for postgenomic study leading to the development of diagnostic and therapeutic applications for precision medicine. Recently, Fuchs *et al.* (2016) established an integrative tool for postgenomic data analysis utilizing next-generation sequencing, RNA-Seq, and microarray data. However, there is limited or no data exist on the genome-wide transcriptome expression profile of human primary monocytes under healthy and disease states.

We have used bioinformatics approaches to catalogue the entire transcriptome of primary monocytes from healthy subjects using deep polyadenylated (Poly(A) +) paired-end RNA sequencing (RNA-Seq) technique. Firstly, we integrated this dataset with other publicly available RNA-Seq datasets for human monocytes to provide a comprehensive gene reference catalogue of human primary monocytes (Mirsafian *et al.*, 2016b). The reference gene catalogue of immune cells based on their transcriptome expression profiles would be useful for immune-related research, particularly for understanding disease states, pathogenesis, and developing therapeutic biomarkers. As a second step, we characterized long noncoding RNAs (lncRNAs) in human monocytes and

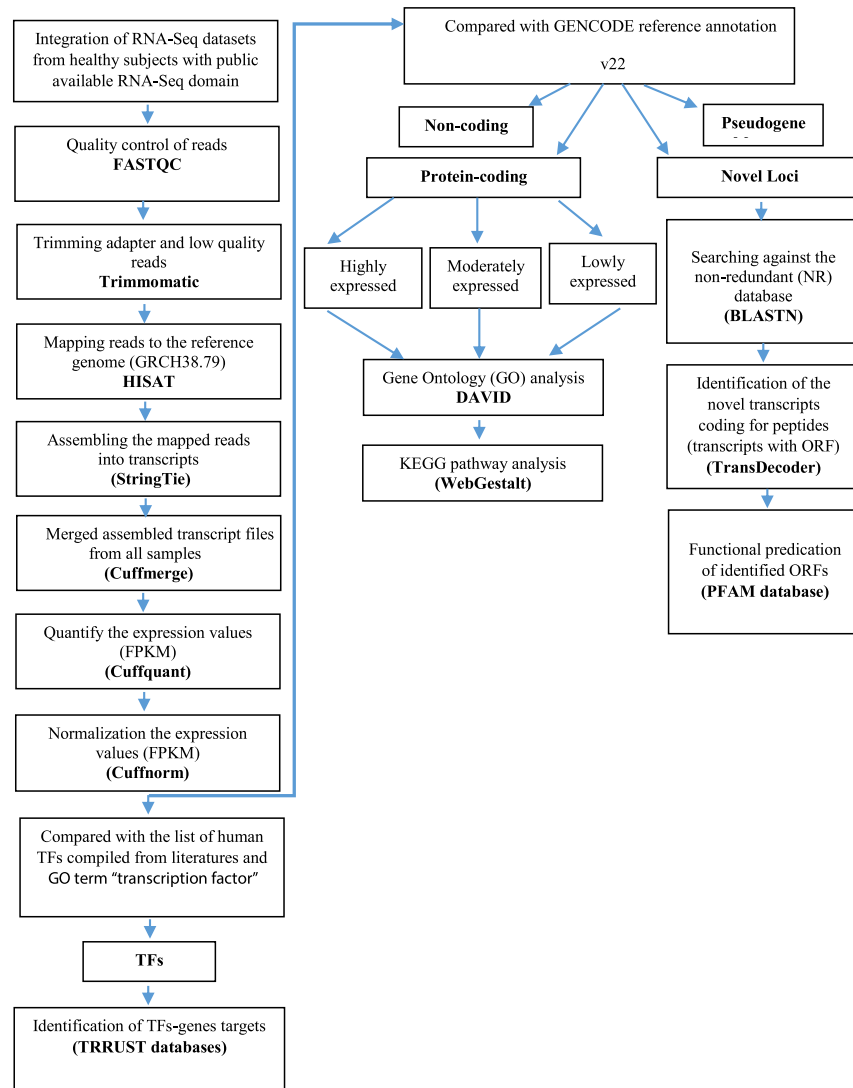


Fig. 1 Schematic representation of workflow to generate the gene catalogue of primary monocytes from healthy subjects.

identified unannotated lncRNAs that were not yet annotated in the public databases, of which this could facilitate future experimental studies to understand the functions of these molecules in the innate immune system (Mirsafian *et al.*, 2016a).

As an example, we present the analysis of deep RNA-Seq datasets of primary monocytes of patients with X-linked agammaglobulinemia (XLA), which is a rare X-linked genetic disorder that affects the male (Mirsafian *et al.*, 2017). XLA is caused by mutations in gene coding for *BTK* (Bruton's tyrosine kinase) located in the Xq21.3-q22 region, which is essential for B cell development and function (Vetrie *et al.*, 1993). XLA disorder is characterized by few or absent of peripheral B cells leading to a decrease in all serum immunoglobulin levels and recurrent infections with encapsulated bacteria and enteroviruses (Ochs and Smith, 1996). The expression of *BTK* is not limited to B cells. It is also expressed in other immune cells such as NK cells (Bao *et al.*, 2012), neutrophils (Honda *et al.*, 2012), and monocytes/macrophages (Koprulu and Ellmeier, 2009). It is well known that *BTK* mutations affected B cell development and functions in XLA patients (Lee *et al.*, 2010; Mohamed *et al.*, 2009; Ochs and Smith, 1996). However, the effect of *BTK* deficiency in primary monocytes of XLA patients is not fully understood and there is no or limited data available on transcriptome profiles of primary monocytes from XLA patients. Moreover, despite the evidence for a role of lncRNAs in the immune system regulation (Derrien *et al.*, 2012; Karapetyan *et al.*, 2013; Guttman *et al.*, 2009), the functions of lncRNAs in primary monocytes of XLA have not been studied yet. Through this study, a comparative gene expression analysis of primary monocytes was conducted between RNA-Seq datasets of XLA patients and healthy male subjects to look into the possible differences in expression patterns of protein-coding genes and lncRNAs in XLA patients compared to healthy individuals, which may affect the function of primary monocytes in these patients. The high-resolution genome-wide transcriptome expression profile of primary monocytes based on the RNA-Seq approach would provide a better understanding of monocytes' characterization and function in healthy and XLA states. It also assists the detailed analyses of innate immune system abnormalities and novel pathomechanisms concerning XLA.

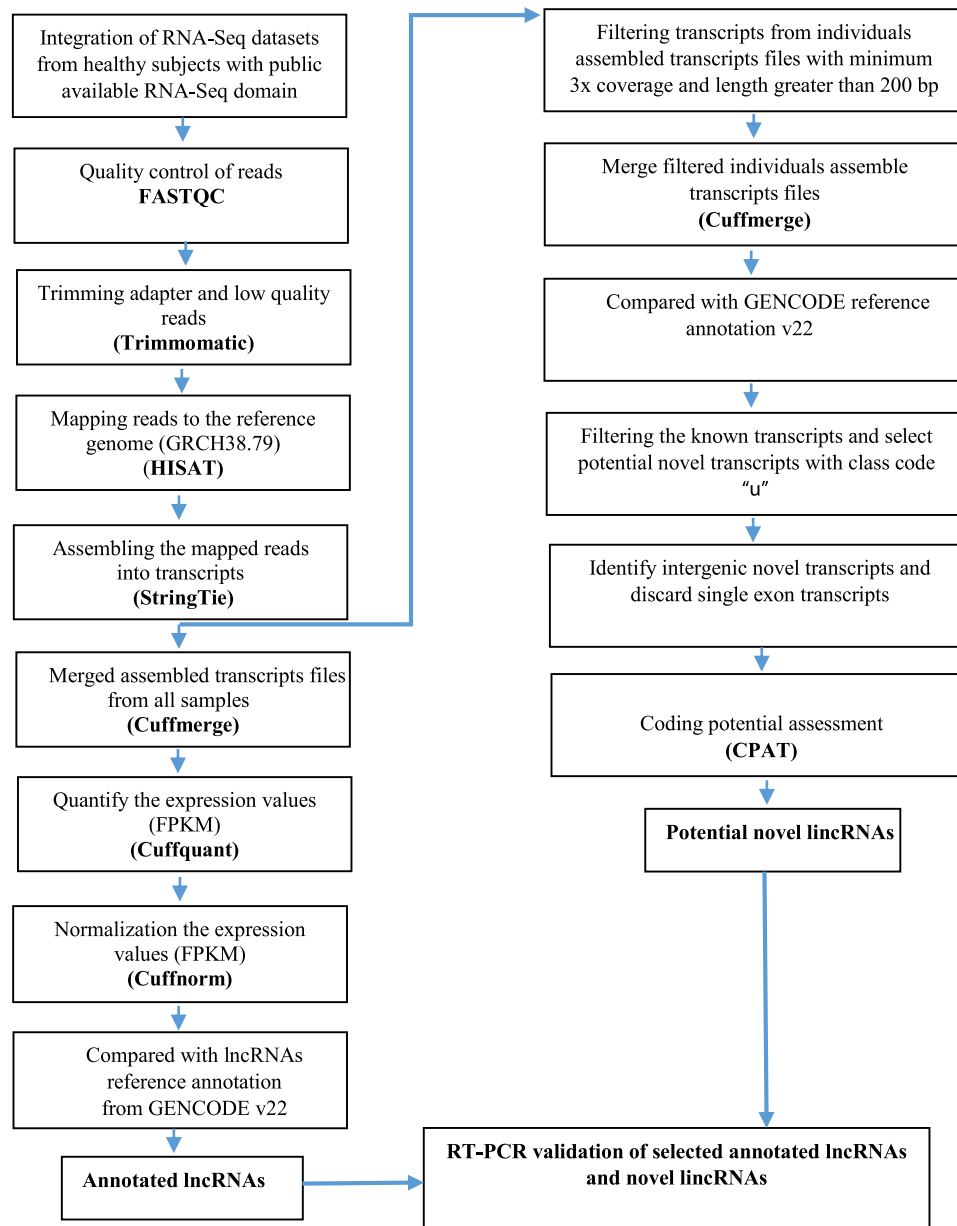


Fig. 2 Schematic representation of workflow to profile the lncRNAs expression landscape of primary monocytes from healthy subjects.

Step 1: Techniques for Cell Preparation and Transcriptome Profiling

The techniques used for monocyte isolation and transcriptome profiling have been described in [Mirsafian et al. \(2016a,b, 2017\)](#). Briefly, the peripheral blood samples from 6 healthy individuals (3 male and 3 female) and 3 male patients with XLA were collected. The classical monocytes (CD14 + + CD16 –) were isolated from peripheral blood mononuclear cells, which are a mixed population of different cell types including lymphocytes, monocytes, and macrophages) using a negative selection technique. Using this method, the nonmonocyte cells including T cells, B cells, dendritic cells, NK cells, and basophils were indirectly magnetically labeled and the untouched monocytes were then isolated by depletion of the magnetically labeled cells. The purity of isolated monocytes from all samples were checked separately by flow cytometry analysis and were found to be >90% for all samples. The total RNA was extracted from monocytes and the quality, quantity, and integrity of the extracted RNA from all samples were checked separately. The purified RNA was used to perform deep poly(A) + paired-end RNA sequencing. All the reads were mapped to the reference genome (Ensembl GRCH38.79) and assembled into a transcriptome. The transcripts were reconstructed for all aligned reads separately and then merged together to form a single nonredundant set of transcripts. The abundance of assembled transcripts was estimated using fragments per kilobase of exon per million fragments mapped value. Detailed information on the bioinformatics analysis workflows and pipeline is described in [Figs. 1–3](#).

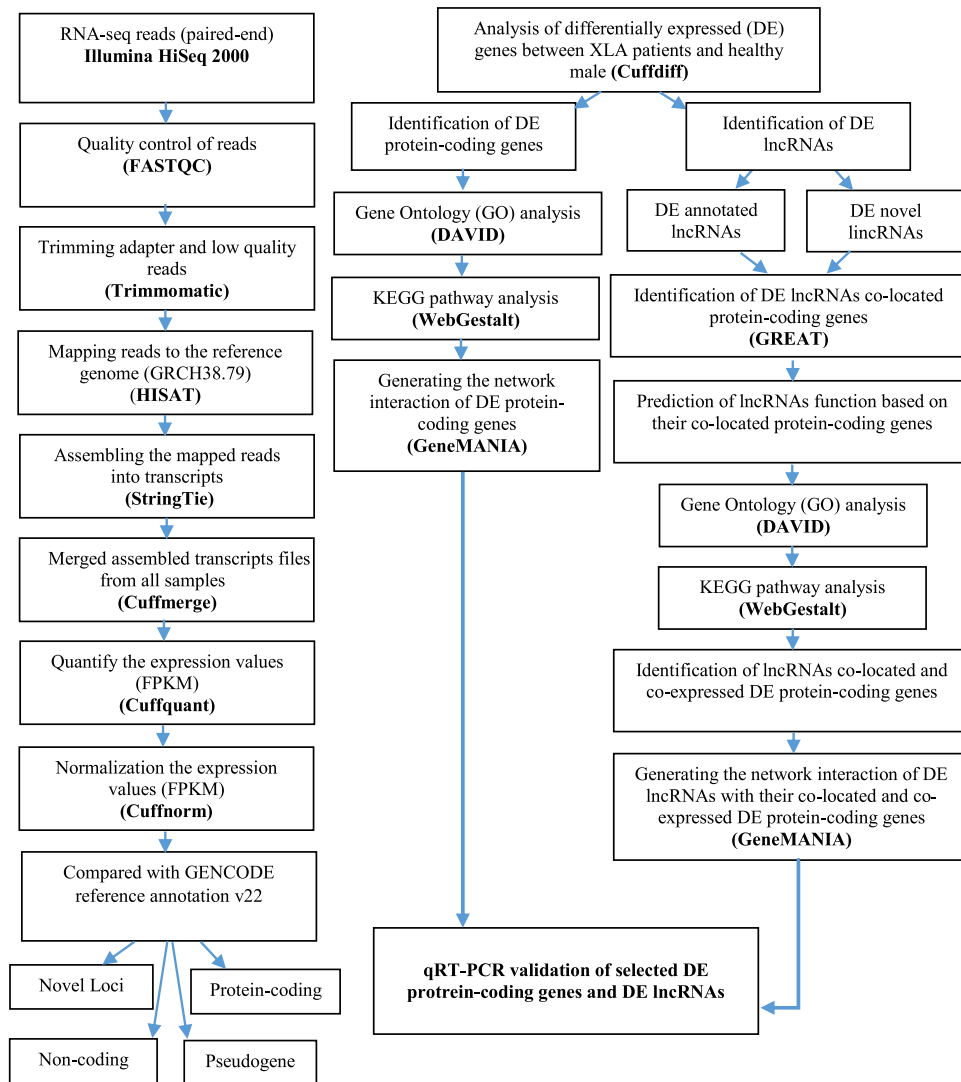


Fig. 3 Schematic representation of workflow of analysis of RNA-Seq dataset of primary monocytes from XLA patients.

Step 2: Establishment of a Reference Gene Catalog of Human Primary Monocytes

RNA-Seq datasets from 6 healthy individuals were integrated with other publicly available RNA-Seq datasets for monocytes. Based on these datasets, we were able to capture most of the genes transcribed in human monocytes including: 11,994 protein-coding genes, 5558 noncoding genes, 2820 pseudogenes, and 7034 putative novel transcripts. The summary of identified genes and transcripts with regard to their biotype is presented in [Table 1](#).

The functional analysis of identified protein-coding genes expressed in monocytes revealed that highly expressed genes were mainly involved in several processes belonging to the immune system while the moderately expressed genes and lowly expressed genes were mainly involved in metabolic processes and biological regulations ([Fig. 4](#)). These results showed that genes within a particular process are expressed at similar levels, which is in agreement with a previous study ([Toung et al., 2011](#)). It has been reported that in B cells, the high-expressing genes were involved in translation, RNA processing, and splicing transcription, and the medium- and low-expressing genes were related to cell adhesion and ion transport and transcription, respectively ([Toung et al., 2011](#)).

Using these datasets, the expression pattern of 1155 transcription factors (TFs) in human primary monocytes were also profiled. TFs are key molecules that control gene transcriptions ([Vaquerizas et al., 2009](#)). Over the past 30 years, several TFs involved in the immune system have been discovered and their mechanisms of action were studied ([Smale, 2014](#)). The interaction network between the top 20 highly expressed identified TFs with their targeted genes is presented in [Fig. 5](#).

Addition of publicly available RNA-Seq datasets for monocytes to our RNA-Seq dataset from healthy individuals also allowed us to provide a landscape of lncRNAs in human primary monocytes. Several recent studies have shown the role of lncRNAs in relation to immune regulation and their role in several autoimmune diseases like SLE ([Shi et al., 2014](#)) and rheumatoid arthritis ([Müller et al., 2014](#)). Recent studies have also indicated the possible role of lncRNA expression and its correlation to immune cells' differentiation and

Table 1 Summary of identified genes and transcripts in human monocytes

Gene biotype	Protein-coding	Noncoding			Pseudogenes	Novel
		Long noncoding	Pre-miRNAs	miRNA, snRNA, and snoRNA)		
Number of genes	11,994	4,799	165	593	2,820	N/A
Number of transcripts	63,515	5,608	186	601	3,233	7,034
Distribution across chromosomes	All chromosomes	All chromosomes	All chromosomes, except: Y	All chromosomes	All chromosomes	All chromosomes
Top 10 highly expressed genes (FPKM > 20)	<i>ANKRD28</i>	<i>CH507-513H4.4</i>	<i>hsa-miR5188</i>	<i>RN7SK</i>	<i>AC007881.4</i>	<i>XL0C_038138</i>
	<i>PTPRD</i>	<i>RMRP</i>	<i>hsa-miR6805</i>	<i>RN7SL5P</i>	<i>USP8P1</i>	<i>XL0C_029167</i>
	<i>B2M</i>	<i>SNHG5</i>	<i>hsa-miR7705</i>	<i>RN7SL4P</i>	<i>TMSB4XP6</i>	<i>XL0C_019828</i>
	<i>TMSB4X</i>	<i>SNHG1</i>	<i>hsa-miR181b</i>	<i>RN7SKP203</i>	<i>RP11-649E7.8</i>	<i>XL0C_029358</i>
	<i>FTH1</i>	<i>LINC00824</i>	<i>hsa-miR4709</i>	<i>RN7SKP255</i>	<i>EEF1A1P5</i>	<i>XL0C_047644</i>
	<i>S100A9</i>	<i>NEAT1</i>	<i>hsa-miR-129</i>	<i>RNU1-2</i>	<i>HCG4B</i>	<i>XL0C_011102</i>
	<i>CROCC</i>	<i>RP11-72M17.1</i>	<i>hsa-miR155</i>	<i>MALAT1</i>	<i>CTD-2031P19.4</i>	<i>XL0C_067917</i>
	<i>EEF1A1</i>	<i>ADAMTSL4-AS1</i>	<i>hsa-miR6723</i>	<i>RNU2-2P</i>	<i>HLA-S</i>	<i>XL0C_020955</i>
	<i>LYZ</i>	<i>CH507-513H4.6</i>	<i>hsa-miR6873</i>	<i>SNORA31</i>	<i>RP41P1</i>	<i>XL0C_069963</i>
	<i>HFM1</i>	<i>RP11-342K6.3</i>	<i>hsa-miR6852</i>	<i>RNU1-1</i>	<i>EEF1A1P6</i>	<i>XL0C_006796</i>

Note: Reproduced from Mirsafian, H., Ripen, A.M., Manaharan, T., Mohamad, S.B., Merican, A.F., 2016. Towards a reference gene catalogue of human primary monocytes. OMICS, 20 (11), 627–634.

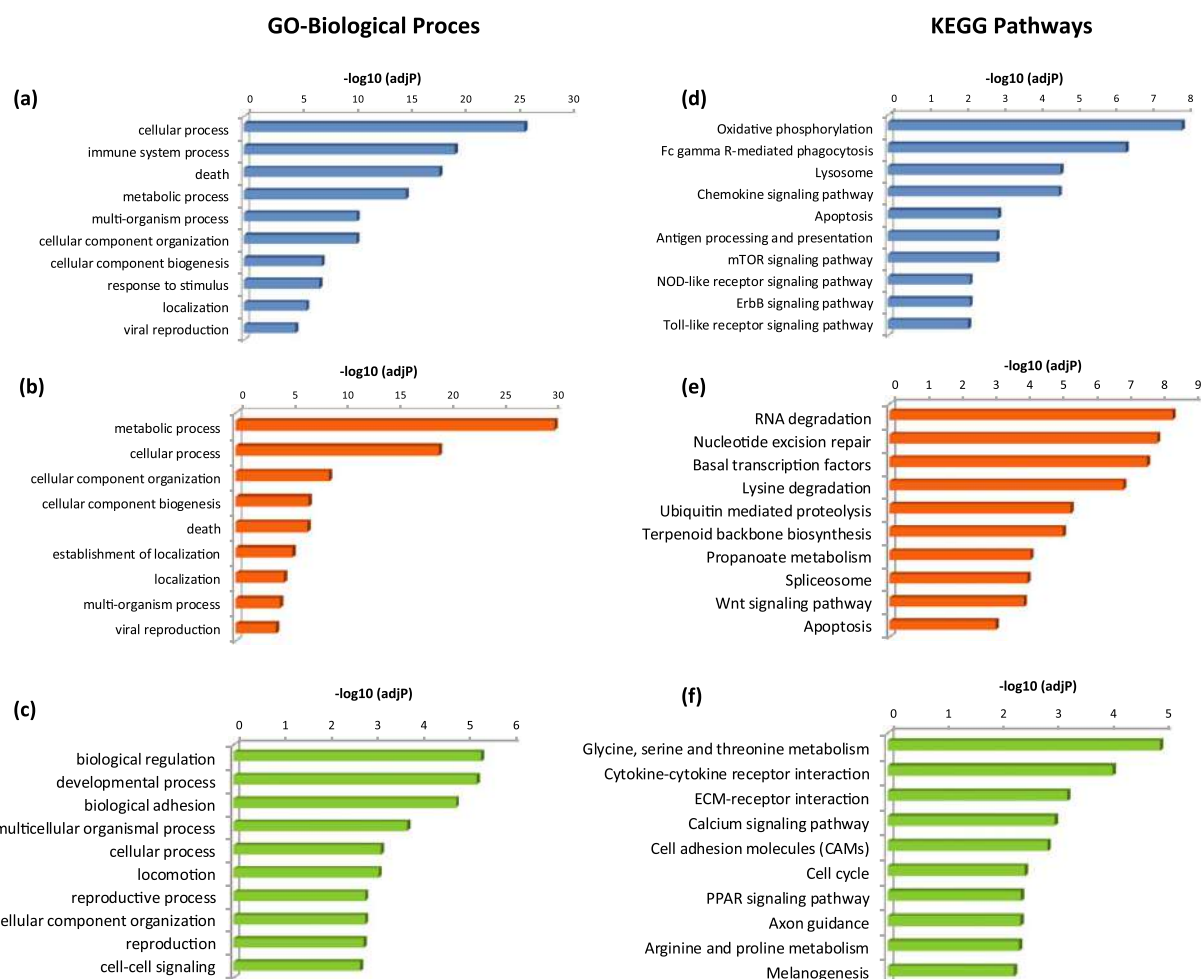


Fig. 4 The GO and KEGG pathway analysis of protein-coding genes in human monocytes. (a) The significant GO and KEGG pathway terms ($\text{adjP} < 0.01$) of highly expressed protein-coding genes. (b) The significant GO and KEGG pathway terms ($\text{adjP} < 0.01$) of moderately expressed protein-coding genes. (c) The significant GO and KEGG pathway terms ($\text{adjP} < 0.01$) of low expressed protein-coding genes. Reproduced from Mirsafian, H., Ripen, A.M., Manaharan, T., Mohamad, S.B., Merican, A.F., 2016. Towards a reference gene catalogue of human primary monocytes. OMICS, 20 (11), 627–634.

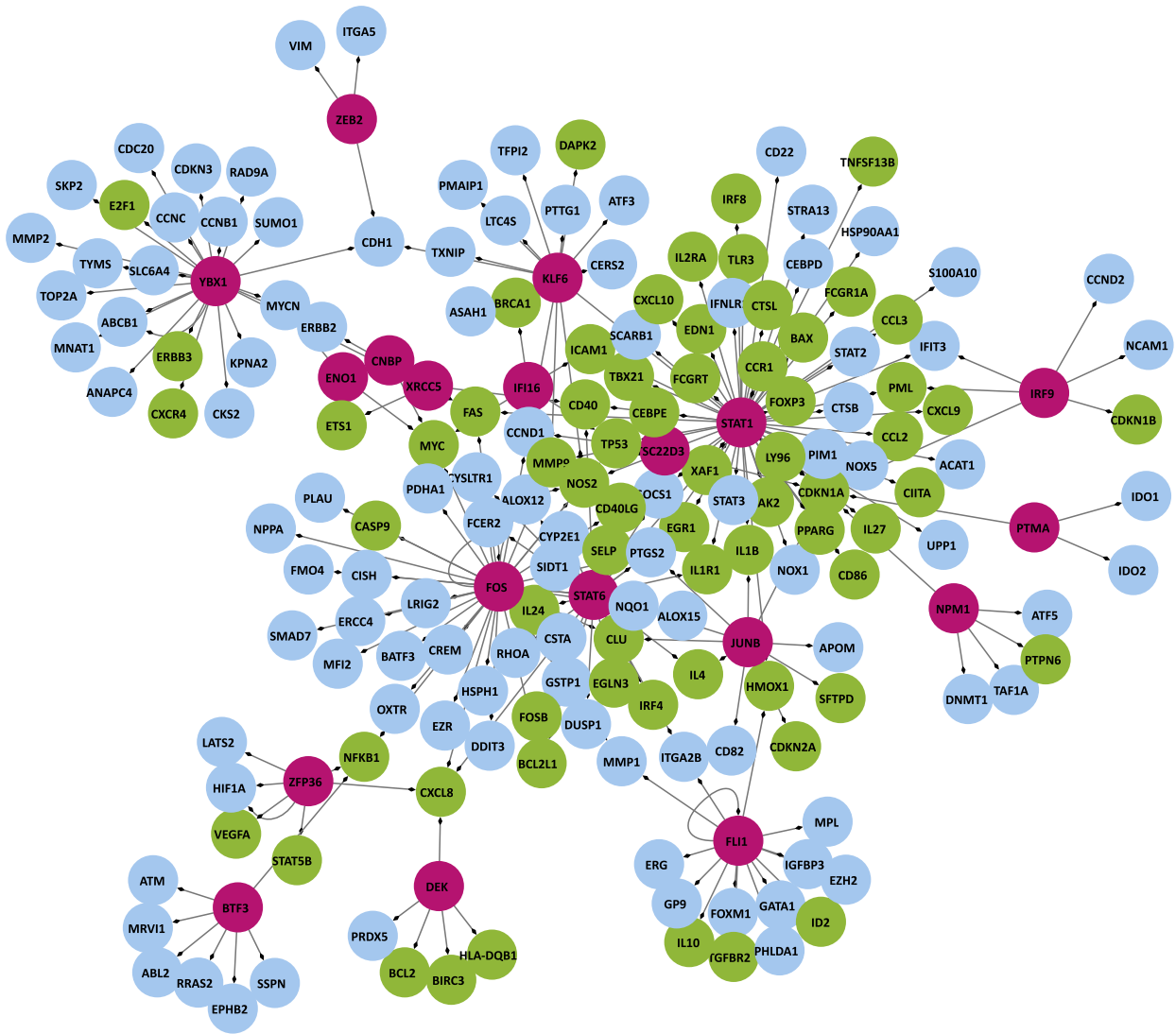


Fig. 5 Interaction network analysis of the top 20 highly expressed TFs with their target in human monocytes. The network contains 226 interactions between 20 TFs and 146 targeted genes. The pink circles represented TFs, while the blue and green circles represented TF-target genes. The green circles represent the genes that are involved in immune system process and death. Reproduced from Mirsafian, H., Ripen, A.M., Manaharan, T., Mohamad, S.B., Merican, A.F., 2016a. Towards a reference gene catalogue of human primary monocytes. OMICS, 20 (11), 627–634.

maturation (Stachurska *et al.*, 2014). However, most of the lncRNAs transcribed in the innate immune system remain unknown. From this study, the expression of 6382 lncRNAs (40% of the known lncRNAs reported in GENCODE database (version 22)) were characterized in human primary monocytes. Using a computational pipeline developed in-house, a total of 1032 unannotated lincRNAs in monocytes were identified that have not been annotated in the public databases (Mirsafian *et al.*, 2016a).

The study described above resulted in a comprehensive reference gene catalog of human primary monocytes under healthy states, which could be used as an important resource for postgenomics and system biology research on human monocytes under healthy and diseased states (Mirsafian *et al.*, 2016b).

Step 3: Comparative Analysis of Transcriptome Profile of Healthy Individuals and XLA Patients

Deep RNA-Seq approach on primary monocytes from 3 XLA patients generated approximately 477 million reads of 100 bp read length. This led to the profiles of 17,510 genes (including 11,788 protein-coding genes, 3681 noncoding genes, 2041 pseudo-genes) and 62,367 transcripts (including 58,136 annotated and 4231 novel transcripts) in the primary monocytes of XLA patients. In addition, the lncRNAs expression patterns in primary monocytes of XLA patients were analyzed and a total of 3363 annotated lncRNAs were identified. By using multistep mapping and filtering criteria, the expression of 430 potential novel lincRNAs in the patients were also identified.

A comparative analysis on gene expression profiles of primary monocytes between healthy individuals and XLA patients was performed to examine possible differences in the gene expression patterns of primary monocytes in XLA patients. The comparative analysis showed the total of 1827 DE protein-coding genes in XLA patients compared to healthy subjects. Out of 1827 DE protein-coding genes, 859 genes were upregulated and 968 genes were downregulated in XLA patients compared to the healthy subjects. Based on the GO and KEGG pathways analysis, the detailed information on the biological functions and potential mechanisms of detected DE protein-coding genes were obtained. The GO enrichment analysis showed that downregulated genes were mainly involved in regulation of immune response. Pathway analysis also revealed that downregulated genes mainly enriched in several pathways belonged to innate immune system such as "Fc gamma R-mediated phagocytosis," "Chemokine signaling pathways," "Toll like receptors signaling pathway," and "MTOR signaling pathway," which reflected the deficiencies of innate immune function in primary monocytes of XLA patients (Fig. 6).

Evidence suggests that in addition to protein-coding genes, lncRNAs can act as key regulators of various biologic processes (Clark and Mattick, 2011; Kim and Sung, 2012). The lncRNAs function via DNA–DNA, DNA–RNA, or other kinds of interactions (Satpathy and Chang, 2015). The lncRNAs dysregulated expression has been reported in many human diseases (Hrdlickova *et al.*, 2014; Wapinski and Chang, 2011). However, little is known about lncRNAs' role in XLA patients. Through the comparative analysis of lncRNA expressions in primary monocytes between XLA patients and healthy subjects, 95 DE annotated lncRNAs including 56 upregulated and 39 downregulated and 20 DE novel lncRNAs including 5 upregulated and 15 downregulated were obtained in XLA patients compared to healthy subjects.

Overall the results as presented above indicated the inclusive dysregulation of monocytes' immune functions and the increase of susceptibility to apoptosis in monocytes of XLA patients. This suggests that BTK mutations are not only affecting the B cell development and differentiation, they would be also contributing to dysregulation of innate immune system in XLA patients. This study also revealed the differentially expression patterns of lncRNAs in primary monocytes of XLA patients that would suggest the potential role of lncRNAs in regulation of immune functions in primary monocytes of XLA patients (Mirsafian *et al.*, 2017).

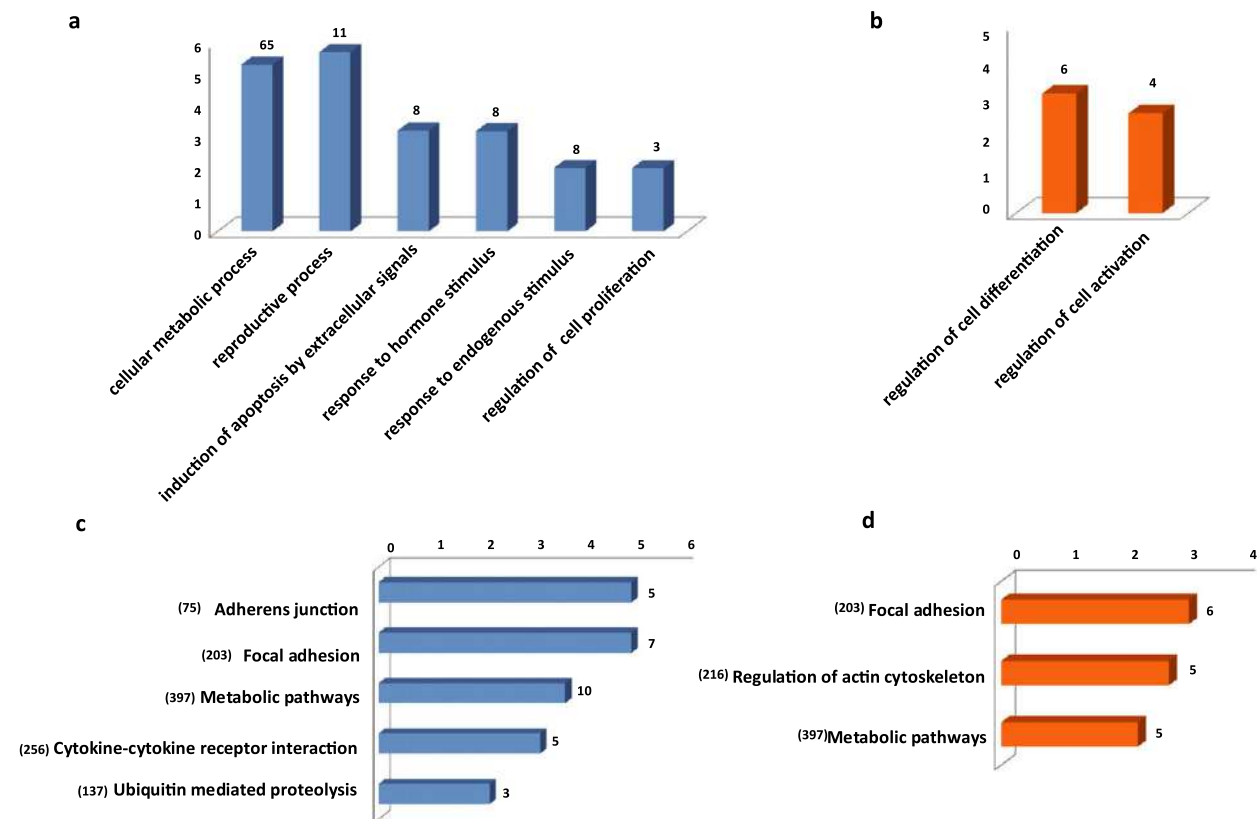


Fig. 6 The GO and KEGG pathway analysis of DE lncRNAs colocated genes in primary monocytes of XLA patients compared to healthy subjects. The significant GO biological process terms enriched for (a) DE annotated lncRNAs colocated genes, and (b) DE novel lncRNAs colocated genes. The number of DE lncRNAs colocated genes enriched in each GO terms is depicted above the bars in the figure. The significant KEGG pathway terms enriched for (c) DE annotated lncRNAs colocated genes, and (d) DE novel lncRNAs colocated genes. The numbers in the brackets indicated the total numbers of genes available in the KEGG database for each pathway terms. The number of identified DE lncRNAs colocated genes enriched in each KEGG pathways is depicted above the bars in the figure. Reproduced from Mirsafian, H., Ripen, A.M., Leong, W.M., *et al.*, 2017. Transcriptome profiling of monocytes from XLA patients revealed the innate immune function dysregulation due to the BTK gene expression deficiency. *Scientific Reports* 7 (1), 6836.

Conclusions

Transcriptomic data based on deep RNA-Seq approach can provide valuable information on differential gene and transcript expression patterns in specific cell types. Herein, we described the analysis of transcriptomic datasets from monocytes of the innate immune system, which resulted in a comprehensive expression profile of human primary monocytes under healthy and XLA disease states. This also includes the establishment of a reference gene catalog of human primary monocytes. A set of differentially expressed (DE) protein-coding genes and DE lncRNAs identified in XLA patients compared to the healthy individuals opens exciting and several potential avenues of research that will help us to better understand the complex pathophysiology in XLA disease. We believe that transcriptomic profiling based on the RNA-Seq approach offers significant promise towards precision medicine, systems diagnostics, immunogenomics, and the development of genomic markers as well as innovative therapeutic monitoring strategies.

Closing Remarks

The availability of high coverage and greater resolution transcriptomic datasets generated from high-throughput RNA-Seq approaches have made possible the applications of systems bioinformatics to further elucidate the functional complexity and provide better understanding of the human body systems towards advancement in healthcare and biomedical research. One of the big challenges that lie ahead is to integrate different types of “omics” datasets (i.e., multiomics datasets comprising, for example, genomics, transcriptomics, proteomics, and metabolomics data from similar or identical biological background) generated from multiomics measurement platforms in order to provide meaningful insights and in-depth understanding on how a given genotype affects the phenotypic features of a given individual through various molecular mediators as well as interaction between a genotype and the environment. We believe that as technologies mature and computational data analysis pipelines and methodologies become more robust and sophisticated, such challenges will be overcome in the very near future.

Acknowledgements

The authors acknowledged the financial support for the research on the transcriptome profiling of the monocytes using RNA-Seq approach from the UM-MOHE High Impact Research (HIR) Grant Scheme: UM.S/P/HIR/MOHE/30; Ministry of Higher Education Malaysia Fundamental Research Grant Scheme (FRGS): FP050–2016; University of Malaya Research Grant (UMRG): RP004C: 13AFR and University of Malaya PPP Postgraduate Grant: PG086–2013B. The authors would also like to thank the Director General of Health, Malaysia for supporting the work carried out by the authors on this subject.

See also: Computational Systems Biology Applications. Coupling Cell Division to Metabolic Pathways Through Transcription. Natural Language Processing Approaches in Bioinformatics

References

- Ancuta, P., Liu, K.-Y., Misra, V., *et al.*, 2009. Transcriptional profiling reveals developmental relationship and distinct biological functions of CD16+ and CD16- monocyte subsets. *BMC Genomics* 10 (1), 403.
- Ayers, D., Day, P.J., 2015. Systems medicine: The application of systems biology approaches for modern medical research and drug development. *Molecular Biology International* 2015, 698169.
- Baldi, P., Hatfield, G.W., 2002. *DNA microarrays and Gene Expression: From Experiments To Data Analysis And Modeling*. Cambridge: Cambridge University Press, Paperback reissue.
- Bao, Y., Zheng, J., Han, C., *et al.*, 2012. Tyrosine kinase Btk is required for NK cell activation. *Journal of Biological Chemistry* 287 (28), 23769–23778.
- Beyer, M., Mallmann, M.R., Xue, J., *et al.*, 2012. High-resolution transcriptome of human macrophages. *PLOS ONE* 7 (9), e45466.
- Clark, M.B., Mattick, J.S., 2011. Long noncoding RNAs in cell biology. *Seminars in Cell & Developmental Biology* 22 (4), 366–376.
- Dang, Y., Xu, X., Shen, Y., *et al.*, 2016. Transcriptome analysis of the innate immunity-related complement system in spleen tissue of ctenopharyngodon idella infected with *aeromonas hydrophila*. *PLOS ONE* 11 (7), e0157413.
- Derrien, T., Johnson, R., Bussotti, G., *et al.*, 2012. The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. *Genome Research* 22 (9), 1775–1789.
- Dong, C., Zhao, G., Zhong, M., *et al.*, 2013. RNA sequencing and transcriptomal analysis of human monocyte to macrophage differentiation. *Gene* 519 (2), 279–287.
- Dong, Z., Chen, Y., 2013. Transcriptomics: Advances and approaches. *Science China. Life Sciences* 56 (10), 960–967.
- Finotello, F., Di Camillo, B., 2015. Measuring differential gene expression with RNA-seq: Challenges and strategies for data analysis. *Briefings in Functional Genomics* 14 (2), 130–142.
- Fuchs, S.B.A., Liedler, I., Stelzer, G., *et al.*, 2016. GeneAnalytics: An integrative gene set analysis tool for next generation sequencing, RNAseq and microarray data. *OMICS* 20, 139–151.
- Guttman, M., Amit, I., Garber, M., *et al.*, 2009. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature* 458 (7235), 223–227.
- Honda, F., Kano, H., Kanegane, H., *et al.*, 2012. The kinase Btk negatively regulates the production of reactive oxygen species and stimulation-induced apoptosis in human neutrophils. *Nature Immunology* 13 (4), 369–378.

- Hrdlickova, B., Kumar, V., Kanduri, K., *et al.*, 2014. Expression profiles of long non-coding RNAs located in autoimmune disease-associated regions reveal immune cell-type specificity. *Genome Medicine* 6 (10).
- Immunobiology: The Immune System in Health and Disease ; [Animated CD-ROM Inside]. Janeway, C.A. (Ed.), New York, NY: Garland Publications. [u.a.].
- Karapetyan, A., Buiting, C., Kuiper, R., Coolen, M., 2013. Regulatory roles for long ncRNA and mRNA. *Cancers* 5 (2), 462–490.
- Kim, E.-D., Sung, S., 2012. Long noncoding RNA: Unveiling hidden layer of gene regulatory networks. *Trends in Plant Science* 17 (1), 16–21.
- Koprlu, A.D., Ellmeier, W., 2009. The role of Tec family kinases in mononuclear phagocytes. *Critical Reviews in Immunology* 29 (4), 317–333.
- Kraft-Terry, S.D., Gendelman, H.E., 2011. Proteomic biosignatures for monocyte–macrophage differentiation. *Cellular Immunology* 271 (2), 239–255.
- Krauss, M., Hofmann, U., Schafmayer, C., *et al.*, 2017. Translational learning from clinical studies predicts drug pharmacokinetics across patient populations. *NPJ Systems Biology and Applications* 3, 11.
- Kukurba, K.R., Montgomery, S.B., 2015. RNA sequencing and analysis. *Cold Spring Harbor Protocols* 2015 (11), 951–969.
- Lee, P.P.W., Chen, T.-X., Jiang, L.-P., *et al.*, 2010. Clinical Characteristics and Genotype-phenotype Correlation in 62 Patients with X-linked Agammaglobulinemia. *Journal of Clinical Immunology* 30 (1), 121–131.
- Likić, V.A., McConville, M.J., Lithgow, T., Bacic, A., 2010. Systems Biology: The Next Frontier for Bioinformatics. *Advances in Bioinformatics* 2010, 268925.
- McDowell, J., ed. 2010. *Encyclopedia of Human Body Systems*. vol. 1 Santa Barbara, California: Greenwood.
- Mirsafian, H., Manda, S.S., Mitchell, C.J., *et al.*, 2016a. Long non-coding RNA expression in primary human monocytes. *Genomics* 108 (1), 37–45.
- Mirsafian, H., Ripen, A.M., Leong, W.M., *et al.*, 2017. Transcriptome profiling of monocytes from XLA patients revealed the innate immune function dysregulation due to the BTK gene expression deficiency. *Scientific Reports* 7 (1), 6836.
- Mirsafian, H., Ripen, A.M., Manaharan, T., Mohamad, S.B., Merican, A.F., 2016b. Towards a reference gene catalogue of human primary monocytes. *OMICS* 20 (11), 627–634.
- Mitchell, C.J., Getnet, D., Kim, M.-S., *et al.*, 2015. A multi-omic analysis of human naïve CD4+ T cells. *BMC Systems Biology* 9 (1),
- Mohamed, A.J., Yu, L., Bäckesjö, C.-M., *et al.*, 2009. Bruton's tyrosine kinase (Btk): Function, regulation, and transformation with special emphasis on the PH domain. *Immunological Reviews* 228 (1), 58–73.
- Müller, N., Döring, F., Klapper, M., *et al.*, 2014. Interleukin-6 and tumour necrosis factor- α differentially regulate lincRNA transcripts in cells of the innate immune system in vivo in human subjects with rheumatoid arthritis. *Cytokine* 68 (1), 65–68.
- Ochs, H.D., Smith, C.I., 1996. X-linked agammaglobulinemia. A clinical and molecular analysis. *Medicine* 75 (6), 287–299.
- Satpathy, A.T., Chang, H.Y., 2015. Long noncoding RNA in hematopoiesis and immunity. *Immunity* 42 (5), 792–804.
- Shi, L., Zhang, Z., Yu, A.M., *et al.*, 2014. The SLE transcriptome exhibits evidence of chronic endotoxin exposure and has widespread dysregulation of non-coding and coding RNAs. *PLOS ONE* 9 (5), e93846.
- Sirri, V., Urcuqui-Inchima, S., Roussel, P., Hernandez-Verdun, D., 2008. Nucleolus: The fascinating nuclear body. *Histochemistry and Cell Biology* 129 (1), 13–31.
- Slack, J.M.W., 2013. *Essential Developmental Biology*. Oxford: Wiley-Blackwell.
- Smale, S.T., 2014. Transcriptional regulation in the immune system: A status report. *Trends in Immunology* 35 (5), 190–194.
- Stachurska, A., Zorro, M.M., van der Sijde, M.R., Withoff, S., 2014. Small and long regulatory RNAs in the immune system and immune diseases. *Frontiers in Immunology* 5.
- Taylor, C., 2017. *The Kingfisher Science Encyclopedia*, fourth ed. London, England: Kingfisher Publications Plc.
- Toung, J.M., Morley, M., Li, M., Cheung, V.G., 2011. RNA-sequence analysis of human B-cells. *Genome Research* 21 (6), 991–998.
- Vaquerizas, J.M., Kummerfeld, S.K., Teichmann, S.A., Luscombe, N.M., 2009. A census of human transcription factors: Function, expression and evolution. *Nature Reviews Genetics* 10 (4), 252–263.
- Vetrie, D., Vořechovský, I., Sideras, P., *et al.*, 1993. The gene involved in X-linked agammaglobulinaemia is a member of the src family of protein-tyrosine kinases. *Nature* 361 (6409), 226–233.
- Wapinski, O., Chang, H.Y., 2011. Long noncoding RNAs and human disease. *Trends in Cell Biology* 21 (6), 354–361.
- Zieba, A., Sjöstedt, E., Olovsson, M., *et al.*, 2015. The human endometrium-specific proteome defined by transcriptomics and antibody-based profiling. *OMICS: A Journal of Integrative Biology* 19 (11), 659–668.
- Ziegler-Heitbrock, L., Ancuta, P., Crowe, S., *et al.*, 2010. Nomenclature of monocytes and dendritic cells in blood. *Blood* 116 (16), e74–e80.

Host-Pathogen Interactions

Dean Southwood and Shoba Ranganathan, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

"You must not fight too often with one enemy, or you will teach him all your art of war."

– Plutarch

Humanity's war against infectious disease is an epic and gruelling conflict, one which has not yet been won. In the last two centuries, advances in science and technology have turned the tide on many fronts, through the development of treatments to help the body subdue, eradicate, or prevent infections. This is a war where espionage and intelligence go a long way – the more knowledge we have about pathogens and their interactions with their respective host systems, the easier it is to find new ways of defeating them; or at the very least, subduing them. Substantial improvements made in recent decades in genomic and proteomic technology have allowed for vast amounts of data to be created. The challenge now is to find elegant and efficient methods of analysing this data to work out the fine details of host-pathogen interactions – how pathogens invade, evade, and proliferate, and the damage they do to their hosts in the process. Such information is key to the development of new therapeutics, treatment strategies, methods of diagnosis, and vaccines to prevent and mediate infection.

The field of bioinformatics has been essential in keeping up with the flood of new data in this area. Strategies for storing, handling, and analysing such data have been developed, and are continuing to be improved on. The post-genomic era has seen an explosion of databases and methodologies for investigating the different aspects of host-pathogen interactions. Each omics field has developed its own approaches to tackling the problem, through processing, analysis, and prediction. There are also current efforts to synthesise the information obtained from a variety of different omics fields to create a global approach to host-pathogen interaction determination.

This article is divided into four main sections. The first section sets up the fundamentals of host-pathogen interactions and current methods to investigate them. This includes a discussion of the terminology of pathogenesis, and the contention around how it is defined, as well as discussion of infection, and current bioinformatics tools to probe it. The second section focusses on how these fundamentals have been applied in the design of drugs and vaccines. The third section discusses the current issues and unresolved contentions surrounding host-pathogen interactions, and the role of bioinformatics in solving them. The last section deals with possible outlooks to the future, with an emphasis on open questions and areas requiring a synthesis of ideas.

Fundamentals

Defining Pathogens

Before we can begin to discuss current approaches to elucidating the details of host-pathogen interactions, we need to clarify what it is we mean by pathogens, what kinds of hosts they invade, and the nature of possible interactions between the two. Historically, as our understanding of disease and its causative agents has increased, the emphasis has shifted from an invader-centric view to a more holistic, host-invader view, where factors such as the type of host and its immune defences, as well as the interaction between the invader and the host are important to defining what a pathogen is. In the last two decades, there has been significant discussion on redefining terms; such as, pathogen, pathogenicity, and virulence; to match this equal emphasis (Casadevall and Pirofski, 1999, 2000, 2001; Methot and Alizon, 2014). These are summarised in [Table 1](#), where the common thread concerns the ability to cause damage in a particular host, and the possible extent of this damage. Such an idea makes host-pathogen interactions the centrepiece in investigating disease as they form the crucial connection between the intricate details of a pathogen and the intricate details of the host, which taken individually do not give us a good picture of disease itself.

If we take a pathogen as being something which causes damage in a host, the next questions to ask are: what kind of pathogens do we know about already, and what hosts do they infect? At present, it is useful to categorise pathogens into six types: viruses, bacteria, fungi, protozoa, parasites, and prions. Most work concerning host-pathogen interactions has been done with respect to the first four categories. The fifth category, parasites, is essentially a catch-all for infective organisms which do not fit well into any of the other categories, helminths being prominent examples. The final category, prions, is unique in that the pathogen is (in some cases hypothesised to be) a misfolded protein, rather than an organism. Key representatives of each category (with humans as hosts) are given in [Table 2](#), along with the diseases they cause.

Not every possible invasive species is pathogenic in humans – how pathogenic a particular organism is depends strongly on the host it is trying to invade. The defences that must be evaded and the conditions that must be adapted to vary widely in their details from host to host, and will often require a unique set of machinery. Therefore, most pathogens evolve to target a particular host almost exclusively. It is important to recognise that interactions pathogens may have with non-human hosts are hints rather than concrete answers as to their interactions with humans. However, there are several good reasons to investigate pathogens in organisms aside from humans. Firstly, the study of pathogenesis in model organisms is often more achievable than in humans, due to well-established methods and comprehensive databases. Organisms; such as, the nematode *Caenorhabditis elegans* (Kurcz and

Table 1 Definitions of common terms associated with host-pathogen interactions, emphasising the dual sources of pathogenicity – pathogen and host

Term	Definition (as per Casadevall and Pirofski, 1999)
Pathogen	A microbe capable of causing host damage; host damage can result from either direct microbial action or the host immune response
Pathogenicity	The capacity of a microbe to cause damage in a host
Virulence	The relative capacity of a microbe to cause damage in a host
Virulence factor	A component of a pathogen that damages the host; this can include components essential for viability

Casadevall, A., Pirofski, L.A., 1999. Host-pathogen interactions: Redefining the basic concepts of virulence and pathogenicity. *Infection and Immunity* 67(8), 3703–3713.

Table 2 Common examples of each type of pathogen in humans, including the corresponding diseases they cause

Pathogen Type	Pathogens	Diseases
Viruses	Hepatitis (A-E) virus	Hepatitis (A-E)
	Herpes simplex virus (1–2)	Herpes
	Human immunodeficiency virus	HIV/AIDS
	Influenza virus A	Influenza
	Measles virus	Measles
	Dengue virus	Dengue fever
	Varicella zoster virus	Chicken pox
	Zika virus	Zika fever
Bacteria	<i>Clostridium tetani</i>	Tetanus
	<i>Vibrio cholerae</i>	Cholera
	<i>Mycobacterium tuberculosis</i>	Tuberculosis
	<i>Streptococcus pneumoniae</i>	Pneumonia
	<i>Salmonella</i> Typhi	Typhoid
	<i>Salmonella</i> Typhimurium	Gastroenteritis
	<i>Borrelia burgdorferi</i>	Lyme disease
	<i>Treponema pallidum</i>	Syphilis
Fungi	<i>Yersinia pestis</i>	Plague
	<i>Aspergillus fumigatus</i>	Aspergillosis
	<i>Candida albicans</i>	Candidiasis (thrush)
	<i>Trichophyton rubrum</i>	Ringworm
	<i>Histoplasma capsulatum</i>	Histoplasmosis
	<i>Cryptococcus neoformans</i>	Cryptococcus and fungal meningitis
	<i>Plasmodium falciparum</i>	Malaria
	<i>Entamoeba histolytica</i>	Amoebic dysentery
Protozoa	<i>Trypanosoma brucei</i>	African sleeping sickness
	<i>Giardia lamblia</i>	Giardiasis (Beaver fever)
	<i>Ancylostoma duodenale</i>	Hookworm
	<i>Schistosoma mansoni</i>	Schistosomiasis (Swamp fever)
	<i>Taenia solium</i>	Tapeworm infection
	CJD prion	Creutzfeldt-Jakob disease
	vCJD prion	Variant Creutzfeldt-Jakob disease
	Kuru prion	Kuru

Ewbank, 2000), the fruit fly *Drosophila melanogaster* (Vodovar *et al.*, 2004), and the zebrafish *Danio rerio* (Meijer and Spaink, 2011; Tobin *et al.*, 2012); are significant non-human hosts in the study of host-pathogen interactions. Secondly, by analysing data obtained from other host organisms, we can investigate tools that pathogens have to invade, evade, and proliferate, which provide leads to finding key interactions with humans. Thirdly, animal and plant diseases are economically and ecologically significant due to their role in food production and conservation. Lastly, the possibility of diseases undergoing a zoonotic jump from animals to humans means the study of animal diseases is of preventative significance (Baigent and McCauley, 2003).

Due to the current directions of the field, as well as pressing medical relevance, the primary focus of the current chapter will be on human hosts and their interactions with pathogens, with only secondary mentions of non-human hosts.

Stages of Infection

In broad terms, there are four stages of host-pathogen interactions: the pathogen invading the host system, the pathogen evading the host system's defences, the pathogen proliferating in the host system, and the host system's immune response to the invader.

Different pathogens will possibly have different machinery at each stage, with some mechanisms developing in response to very specific host system defences. The four stages are not disjoint; there is some overlap between them, with some stages being more infectious for certain pathogens than others. An overview of our current knowledge of host-pathogen interactions at each stage is provided below; state-of-the-art methods used to gain knowledge in these areas are discussed in more detail in the next section.

Invading the host system

Initial infection can occur *via* a number of pathways – the pathogen may enter the body through the mouth or nose, the bloodstream (possibly through a contaminated needle), or through mucosal membranes, as well as other more specialised means. However, once the pathogen has entered, it must interact with the host cells in order to remain within the host and continue its invasion. The form this interaction takes depends on the type of pathogen. Viral pathogens invade host cells by merging their envelope with the membrane of either a cell or an endosome. For other pathogens, including Gram-negative bacteria, specialised protein secretion systems are used to release so-called effector proteins into host cells which trigger a particular response mechanism in the host, allowing the invader to migrate to a preferable environment. Such secretion and injection machinery is not always required, especially in fungal pathogens. Instead, effector proteins are expressed on the surface of the invader, and interact with host cell receptors. The host's response to these proteins typically allows the pathogen to invade deeper into the host system.

In most cases, viral pathogens invade by merging with the host cell membrane itself. This is aided through the use of fusion proteins provided by the virus, promoting attachment and fusing of the host and pathogen membranes. In general, the fusion proteins used by each type of virus are distinct, meaning that vaccines and drug therapies must be designed specifically for each type (Cohen, 2016).

So far, six distinct secretion systems have been discovered in Gram-negative bacteria: Type I (Welch *et al.*, 1981), Type II (d'Enfert *et al.*, 1987), Type III (Galán and Curtiss, 1989), Type IV (Kuldau *et al.*, 1990), Type V (Pohlner *et al.*, 1987), and Type VI (Pukatzki *et al.*, 2006; Mougous *et al.*, 2006). In mycobacteria, there is the Type VII secretory system, also known as the ESX system (Stanley *et al.*, 2003). In general, these systems act as injectors, transporting effector proteins from the invading pathogen to the host cell. The particular proteins secreted by each type of system are still being determined; recent methods for such investigations are discussed in Section "Bioinformatic Methods of Discovery". The details of the delivery mechanisms are also still being discovered; for an in-depth discussion of our current knowledge of secretion system machinery itself, see Further Reading 1 (Costa *et al.*, 2015).

Fungal invasion mechanisms in many ways lie between those of viruses and bacteria. Instead of effector proteins being injected into the host cell, they are only located on the hyphal surface of the pathogen. These are then presented to host cell receptors, which induce a change that the pathogen can exploit. In the case of *Candida albicans*, the changes can lead to induced endocytosis or cellular uptake of the pathogen, as well as active penetration, where the pathogen exploits the interaction to penetrate the epithelial lining (Yang *et al.*, 2014).

As most known invasion mechanisms proceed *via* protein-protein interactions on the surface of cells, glycosylation plays a significant role in determining pathogenicity (Hooper and Gordon, 2001). If the glycosylation of surface proteins in one part of the body is not compatible with the pathogen, the interaction is weakened and invasion is also significantly reduced. Either the presence or the absence of glycosylation can be essential to pathogenicity depending on the type of invader, as some effector proteins selectively bind to certain glycoproteins (Hooper and Gordon, 2001).

Evading the host system

Once the pathogen has successfully gained a foothold inside the host system, it will in most cases attempt to evade the host's defences in order to survive. There are a number of mechanisms through which pathogens try to evade defence systems, including affecting essential pathways in the host to increase or decrease defence activity, inhibiting macrophage activity, and sabotaging crosstalk in pathways using miRNA and short linear motifs (SLiMs).

Immune responses are triggered through a variety of essential pathways, resulting in the activation of both intra- and extra-cellular mechanisms to tackle the invader. In order to evade these mechanisms, pathogens have adapted to target specific pathways in order to optimise their survival chances. One such target is the NF- κ B pathway, which plays a role in the regulation of apoptosis (Tato and Hunter, 2002). A variety of bacterial, viral and parasitic pathogens have all been found to attack the NF- κ B pathway, either through activation or inhibition (Rahman and McFadden, 2011), and it is therefore crucial to early infection. Other possible targets for pathogens are so-called 'hub' proteins, which connect with a variety of signal pathways (Dyer *et al.*, 2008), in particular in the case of the Epstein-Barr virus (Calderwood *et al.*, 2007).

Upon activating the immune defences of the host, the invader is in most cases hunted down and consumed by the macrophage. Therefore, many pathogens attempt to avoid detection by the macrophage, or inhibit its activity. In the case of *Mycobacterium tuberculosis*, this is possibly achieved by inhibiting interleukin-12 production, in turn inhibiting macrophage activation (Nau *et al.*, 2002). In the case of viruses, between 26 and 37 HIV-1 proteins have been associated with monocyte-derived macrophages (Chertova *et al.*, 2006), indicating this is a significant target of the host defence system.

Invaders also attempt to sabotage the molecular crosstalk of the host *via* microRNA regulation (Scaria *et al.*, 2007). This is particularly the case for viral pathogens, where the invading microRNA can modulate the expression of viral and antiviral genes, therefore playing an essential role in staying undetected in the host (Ghosh *et al.*, 2009). Fungal pathogens have also been shown to suppress immunity in plants by hijacking RNA interference pathways using sRNAs (Weiberg *et al.*, 2013).

Another possible avenue for sabotage is through protein-protein interaction, by use of short linear motifs (SLiMs). By mimicking the protein interactions of the host, the pathogen can regulate essential pathways to its benefit (Via *et al.*, 2015). There have been recent attempts to predict such mimicry in the HIV-1-human interaction using disorder prediction algorithms (Becerra *et al.*, 2017).

Proliferating in the host system

It is not normally sufficient for a pathogen to invade and evade a host; there must be some benefit to the pathogen once it has taken root in the host system. This is largely due to adaptations by pathogens to more easily proliferate in the host system, rather than on their own. In many pathogens, there is a substantial difference in the number of genes required to survive inside the host as opposed to outside, with the mechanisms for proliferating inside the host requiring significantly less genes (Raghunathan *et al.*, 2009). The methods through which the pathogen co-opts the host machinery also depend on the type of pathogen. Host factors, or proteins produced by the host which end up playing a significant role in the proliferation of the pathogen, also tend to be significant players in establishing pathogen replication.

The hijacking of host cells by viruses (Davey *et al.*, 2011) and bacteria (Bhavsar *et al.*, 2007) is a key part of the pathogen not only surviving, but thriving in the host. How a particular pathogen has adapted to exploit the inherent machinery of the host is often a significant difference between pathogen types. This is particularly essential in the case of viruses, where the extremely compact nature of their genome requires the host cell machinery for replication. Indeed, the relationship between viruses and their host is so intimate that the host can act as vehicle for gene transfer between viruses (Rappoport and Linial, 2012).

The host can also unknowingly provide essential components for the replication and survival of pathogens. Such host factors are the subject of intense study as possible drug targets which do not depend as heavily on the specific genetic details of the invading pathogen. As many as 295 human host factors have been identified as necessary in the replication of the influenza virus (König *et al.*, 2010), while 213 host factors have been associated with HIV-1 replication (Brass *et al.*, 2008).

Immune control of invader

An invader is unlikely to escape the immune response of the host indefinitely; therefore, at some point the host will seek to control the threat. What methods does a host have at its disposal to control the pathogen? At the fundamental level, the host can trigger gene pathways to create an immune response to destroy the invader. Within a cell, autophagy can be used to remove the threat. Extracellularly, the use of dendritic cells, especially in the intestine, is important in suppressing pathogens. In many immune processes, iron regulation is essential and must be carefully controlled. Therefore, all of the above processes are potential targets for pathogens to prevent immune control and subsequent destruction.

The initial response of a host to the detection of invaders is the activation of immune signalling cascades. Depending on the type of pathogen, particular genes may also be highly expressed; these genes are significant both as targets for pathogens, as well as targets for possible drug therapies in activating the immune system to combat invaders. In the particular case of *Mycobacterium tuberculosis* infection in a mouse host, 67 genes were identified as being highly expressed in immune-competent mice, but not in immune-compromised mice (Talaat *et al.*, 2004). This suggests that these 67 genes are significant in combatting tuberculosis infection in mice, and give potential leads to similar genes in humans.

Autophagy is an intracellular lysosomal degradation process which is usually used to break down damaged organelles and other hazardous cellular products. It is also essential to the degradation of *Mycobacterium tuberculosis* during infection (Vergne *et al.*, 2006). However, the precise mechanisms by which *Mycobacterium tuberculosis* does or does not avoid autophagy in the macrophage is still under discussion, with strong indications that microRNA plays a role (Kim *et al.*, 2017).

Dendritic cells are antigen-presenting cells which act as regulators of both the innate and adaptive immune systems (Kelsall *et al.*, 2002). While they are essential in generating an appropriate immune response from the host, they are not immune to manipulation by invaders. In particular, dendritic cells are able to migrate from sites of infection to the lymph nodes, in turn activating T cells. However, this also allows for the possible migration of pathogens around the body; in the event of build-up of migratory dendritic cells, this also creates persistent inflammatory responses (Worbs *et al.*, 2017). For a thorough discussion of the role of dendritic cells in health and disease, see Worbs *et al.* (2017).

Iron is an essential part of both the responses of hosts to invaders, as well as pathogen survival. During infection, host responses tend to withdraw iron from pathogen sites; however, pathogens have developed methods of harvesting iron from other sources, such as heme, transferrin, ferritin and lactoferrin (Doherty, 2007). Therefore, in iron-deficient patients presenting infections such as tuberculosis, the decision to supplement iron must be weighed against aiding the pathogen (Doherty, 2007).

Bioinformatic Methods of Discovery

Bioinformatics has played, and will continue to play, an extremely large role in determining host-pathogen interactions. In the post-genomic scientific landscape, the emphasis on 'big data' is extremely high; methods which can fully utilise the wealth of data available have an edge on traditional, small-sample methods. Therefore, there has been more recent emphasis on machine learning and network methods in predicting and analysing the connections between hosts and pathogens. However, there is also still a strong contribution from more biochemistry-based approaches which utilise homology and structure to predict novel host-pathogen interactions.

The analysis of current methods in bioinformatics is divided into two broad, and by no means strictly exclusive, categories: biological methods, which arise from traditional biologically-inspired methodologies informed by structure and homology; and computational methods, which arise from more computationally-inspired methodologies such as network analysis and machine learning.

Biological methods

Computational methods of predicting host-pathogen interactions can make significant use of the structural information of proteins, as well as previously known interactions with other proteins, in order to narrow results. This can be done by searching for homologous proteins in the pathogen or host which may also interact with known proteins (so-called interologues). Alternatively, it can be done by comparing structural features such as domains or motifs to those present in known interactions. There is also the possibility of using the 3D structural information of proteins, such as those with known crystal structures, to inform analysis methods. A more detailed recent review of structure-based methods can be found in Further Reading 2 (Mariano and Wuchty, 2017), but highlights are presented in this section to capture the prevailing methods.

Homology-based prediction methods only work well in situations where there is a large degree of similarity or evolutionary conservation between the organisms or proteins under discussion. However, this is often the case in related pathogens, for example between different bacterial serotypes. Such an approach has been taken in analysing malaria in the *Homo sapiens-Plasmodium falciparum* host-pathogen pairing (Krishnadev and Srinivasan, 2008). The method analyses the genomic data of both organisms, and identifies homologous proteins with known protein-protein interactions. These are then compared between the two organisms to determine the likelihood of any protein-protein host-pathogen interactions occurring. A significant problem with homology-based methods is the high false positive rate (Mariano and Wuchty, 2017) – protein pairs identified may not be expressed at the same time, in the same location, and may not ever come into contact, *let alone* interact. Therefore, successful homology approaches require filters that account for these criteria, such as the use of random forest classifiers in again analysing the *Homo sapiens-Plasmodium falciparum* pairing (Dyer *et al.*, 2007; Wuchty, 2011). There has also been prediction of protein-protein interactions through comparative modelling (Davis *et al.*, 2007), which was applied across 10 pathogens, including species of *Mycobacterium*.

The analysis of smaller sections of candidate proteins from either the host or the pathogen can also provide good indications of host-pathogen interactions. By determining the domains of these proteins, and comparing to databases of known domain-domain interactions, lists of possible candidates can be narrowed down (König *et al.*, 2008). There is also the potential for domain-motif and motif-motif interactions through smaller sections of the protein. Such short linear motifs or SLiMs have also been used to improve the accuracy of predictive methods (Becerra *et al.*, 2017), and in HIV-1 (Via *et al.*, 2015). Conserved peptide motifs in human proteins have been used to predict possible human-HIV-1 protein-protein interactions (Evans *et al.*, 2009).

Computational methods

In order to find new host-pathogen interactions, a vast pool of data must be processed, sorting good predictors in the data from bad predictors, and weighing up their contributions to give an overall predictive strategy. This is precisely the domain in which machine learning and network analysis excel. There are two possible approaches within machine learning to solve such problems concerning host-pathogen interactions. Supervised learning is the most prevalent set of methods in analysing host-pathogen interactions, wherein possible predictors are identified beforehand, leaving their weights and relative importance to be determined by a training set of data. This is because in many cases we think we know what features we want to analyse through, and like to rely on curated datasets. For example, some common chosen features include gene expression profiles, gene ontology annotations, topological properties of known proteins, functional motifs and post-translational modifications. However, semi-supervised learning methods are also gaining traction through the prevalence of partially-labelled datasets, allowing other features to be identified throughout the model construction process that may not have been identified as inputs beforehand. Due to the highly interdependent nature of intracellular interactions, network analysis methods have also seen significant use, utilising the vast connections and possible interactions between proteins to identify possible intersections and new host-pathogen interactions.

Classifier-based methods have been a popular choice in identifying host-pathogen interactions, including random forest methods (Tastan *et al.*, 2009; Wuchty, 2011). In these cases, the interactions between *Homo sapiens* and HIV-1, and *Homo sapiens* and *P. falciparum*, respectively, were analysed, using appropriate feature choices in each case to encompass the most reliable known information about the pairing, such as known feature motifs, gene expression profiles, and amino acid profiles. These methods have had some success in predicting known (and recently experimentally verified) interactions, as well as previously unknown ones, with varying accuracy. Other methods, such as Naïve Bayes (Arnold *et al.*, 2009), Support Vector Machines (SVMs) (Wang *et al.*, 2011) and group lasso methods (Kshirsagar *et al.*, 2012) have also had varying levels of success in identifying known host-pathogen interactions and predicting new host-pathogen interactions from big data sets. Many of the features used across models include the amino acid profile of proteins, relative positions of residues in the structures, as well as overarching secondary features that can be identified from the primary sequence. Other methods take information such as known protein-protein interactions to predict new ones using structural similarities. Often, these models require regularizers in order to minimise the bias introduced from the biological assumption that similar proteins will interact in similar ways, which would otherwise limit the power to predict novel interactions.

Multi-task learning and semi-supervised auxiliary tasks, including multi-layer perceptron methods (supervised) and perceptron methods with partial label reference (semi-supervised), are also being used to predict novel host-pathogen interactions from current data

Table 3 Databases for analysing host-pathogen interactions; links current as of September 2017

Name	ID	URL	References
PATRIC	1	http://www.patricbrc.org/	Wattam <i>et al.</i> (2014), Driscoll <i>et al.</i> (2009)
ViPR	2	https://www.viprbrc.org/brc/home.spg?decorator=vipr/	Pickett <i>et al.</i> (2012)
EuPathDB	3	http://eupathdb.org/	Aurrecoechea <i>et al.</i> (2017), Aurrecoechea <i>et al.</i> (2013)
VectorBase	4	http://www.vectorbase.org/	Giraldo-Calderón <i>et al.</i> (2015)
IRD	5	https://www.fludb.org/brc/home.spg?decorator=influenza/	Zhang <i>et al.</i> (2017)
VirHostNet 2.0	6	http://virhostnet.prabi.fr/	Guirmand <i>et al.</i> (2015), Navratil <i>et al.</i> (2009)
VirusMentha	7	http://virusmentha.uniroma2.it/	Calderone <i>et al.</i> (2015)
ViralZone	8	http://viralzone.expasy.org/	Hulo <i>et al.</i> (2011)
ViTa	9	http://vita.mbc.nctu.edu.tw/index.php	Hsu <i>et al.</i> (2007)
HPIDB 2.0	10	http://hpidb.igbb.msstate.edu/	Ammari <i>et al.</i> (2016), Kumar and Naduri (2010a, b)
GPS-Prot	11	http://gpsprot.org/	Fahey <i>et al.</i> (2011)
PHI-base	12	http://phi-base.org/	Urban <i>et al.</i> (2017), Winnenbergh <i>et al.</i> (2006)
PID	13	http://www.ndexbio.org/#/user/301a91c6-a37b-11e4-bda0-000c29202374	Schaefer <i>et al.</i> (2009)
HIV-1, Human Protein Interaction Database	14	http://www.ncbi.nlm.nih.gov/genome/viruses/retroviruses/hiv-1/interactions/	Fu <i>et al.</i> (2009)
HIV Database Tools LANL	15	https://www.hiv.lanl.gov/content/index/	N/A
HCV Database Tools LANL	16	https://hcv.lanl.gov/content/index/	N/A
HFV Database Tools LANL	17	https://hfv.lanl.gov/content/index/	N/A
hpvPDB	18	http://www.bicjbt-drc-mgims.in/hpvPDB/index.html	Kumar <i>et al.</i> (2013)
PHIDIAS	19	http://www.phidias.us/	Xiang <i>et al.</i> (2007)
PHISTO	20	http://www.phisto.org/	Durmuş Tekir <i>et al.</i> (2013)
HoPaCI-DB	21	http://mips.helmholtz-muenchen.de/HoPaCI	Bleves <i>et al.</i> (2014)
VFDB	22	http://www.mgc.ac.cn/VFs/main.htm	Chen <i>et al.</i> (2016), Chen <i>et al.</i> (2005)
IntAct	23	http://www.ebi.ac.uk/intact/	Aranda <i>et al.</i> (2010)
BioGRID	24	http://thebiogrid.org/	Chatr-aryamontri <i>et al.</i> (2017)
Mentha	25	http://mentha.uniroma2.it/	Calderone <i>et al.</i> (2013)

(Qi *et al.*, 2010; Kshirsagar *et al.*, 2013). In this case, fully labelled reference data has been used to train a (supervised) perceptron model, while partially labelled reference data has been used afterwards to improve the feature determination and weighting (semi-supervised) in the case of the *Homo sapiens*-HIV-1 host-pathogen system. Indeed, there has been significant recent emphasis on HIV-1 analysis due to the development of the HIV-1, Human protein interaction database (Fu *et al.*, 2009). Previous machine learning methods have been well-characterized and summarised (Pinney *et al.*, 2009), while more recent *in silico* approaches to *Homo sapiens*-HIV-1 protein-protein interactions have also been thoroughly reviewed (Bandyopadhyay *et al.*, 2015). Network analysis-based methods have also seen use in this area, with interactome-based network analysis of interaction between HIV-1 proteins and signal transduction pathways identifying possible alternatives around the proteins targeted by HIV-1 (Balakrishnan *et al.*, 2009).

For further details and methods of machine learning approaches, Further Reading 3 (Carvalho Leite *et al.*, 2017) has a substantial recent review of methods in this area. For a more thorough recent review of network methods, see Further Reading 4 (Pan *et al.*, 2016), as well as a review of computational systems biology methods in Further Reading 5 (Durmuş *et al.*, 2015).

Databases for Host-Pathogen Interaction

The need for comprehensive, well-curated data sources is a persistent problem in the field of bioinformatics. In some areas this has been solved through large-scale collaborative efforts, such as with the RCSB Protein Data Bank for structural data of proteins (Berman *et al.*, 2000). However, in the field of host-pathogen interactions, there is still a lot of discussion and contention over good data sources, and several curated databases to choose from, varying in the type of data, the quality, and the level of detail in the curation. This section in conjunction with Table 3, aims to provide a survey of currently available sources of large datasets related to host-pathogen interactions. The IDs provided in brackets correspond to those in the ID column of Table 3.

The PATRIC (1), ViPR (2), EuPathDB (3), VectorBase (4), and IRD (5) databases are currently funded by the National Institute of Allergy and Infectious Diseases (NIAID). PATRIC is a bacterial pathogen database which is extremely comprehensive, and has available a number of tools such as genome annotation and comparative pathway analysis. ViPR is a viral pathogen database which is also quite comprehensive in terms of its scope. EuPathDB is a eukaryotic pathogen database, including fungal pathogens, protozoal pathogens, and other parasitic pathogens. VectorBase is a database of interactions between pathogens and invertebrate vectors, such as mosquitoes. The Influenza Research Database (IRD) is a database focussed on interactions between influenza viruses and various hosts. While very specific, it is very comprehensive. VirHostNet 2.0 (6) is a virus-virus, virus-host, and host-host interaction network database. Other virus databases include VirusMentha (7), ViralZone (8), and ViTa (9). Specific databases for particular viruses also exist, such as the HIV-1, Human Protein

Interaction Database (14) and HIV Database Tools LANL (15) (for human immunodeficiency virus), HCV Database Tools LANL (16) (for hepatitis C virus), HFV Database Tools LANL (17) (for haemorrhagic fever virus or Ebola), and hpvPDB (18) (for human papilloma virus). Protein-protein interaction databases such as HPIDB 2.0 (10), PHI-base (12), PID (13), PHIDIAS (19), and PHISTO (20) all provide differing levels of detail, as well as a variety of search tools. More specific databases are available for certain uses, such as HoPaCI-DB (21) for investigations into *Pseudomonas aeruginosa* and *Coxiella*, and the VFDB (22) for virulence factors. Databases such as IntAct (23), BioGRID (24), and Mentha (25) are primarily useful for molecular interactions, and often have more protein-focussed or organism-specific equivalents, such as VirusMentha (7).

Applications

The successful design of drugs and therapeutics to treat infection relies heavily on the knowledge of how the host and pathogen interact. In times where antibiotic resistance is increasing, new strategies are required to treat and prevent infection. There are two main possible strategies for using host-pathogen interactions in drug design: targeting machinery used by the pathogen to disable or reduce its virulence, and targeting machinery used by the host to bolster immune response. Vaccine design typically proceeds *via* the identification of target proteins from the pathogen which can be used as antigens by the immune system. By streamlining the process required to identify new protein targets and immune system mechanisms, drug and vaccine design can be sped up. Therefore, the prediction of key host-pathogen interactions is highly significant to clinical outcomes.

Recent trends in pharmacological design have focussed on both virulence factor disabling and host system boosting (Munguia and Nizet, 2017). The identification of key virulence factors allows for the design of potential inhibitors to neutralize their activity, and possibly render them vulnerable to attack by the immune system, reducing and possibly preventing serious infection. On the other side, therapies can be designed against host mechanisms that are implicated in infection. Strategies such as boosting phagocyte activity as well as reducing immune suppression by pathogens require significant knowledge of how the host and pathogen interact, but present promising leads that allow for the development of non-traditional antibiotics and other therapeutics. Host-pathogen interactions have been used recently in the design of antimalaria drugs *in silico* (Samant *et al.*, 2016), where protein targets were identified through the analysis of protein-protein interaction networks, and the targets subsequently tested for potential inhibitors.

Vaccine design proceeds by targeting the particular features of organisms that interact with the host. This requires knowledge of effector proteins and mechanisms, available through machine learning approaches as well as more traditional structural approaches, followed by experimental validation. Successful prediction of key effector proteins is essential for streamlining the vaccine design process (Zagursky *et al.*, 2003). In the case of viruses, there has been significant progress in recent years towards designing new and more successful vaccines. However, there is still much work to be done, such as in the case of HIV-1 (Haynes *et al.*, 2016). In the case of bacteria, there has also been significant work done in assembling methods to predict protein vaccine candidates (Jaiswal *et al.*, 2013). The study of host-pathogen interactions presents a promising avenue to continue to make progress in this area. Further perspectives can be found in Further Reading 6 (Marston *et al.*, 2014).

Discussion and Future Directions

While much progress has been made in recent years in the investigation of host-pathogen interactions, there are still a number of pressing issues that must be addressed. In broad terms, the main issues can be classed as a lack of detailed understanding of the relationship between pathogens, a lack of variety of different host-pathogen pairs under analysis, and the quality, type, and curation of data available to the community.

There is still uncertainty in how different or similar mechanisms of virulence are across pathogens of related strains, or even related types. As related strains may target different hosts, there can be significant differences between them in terms of how they invade, proliferate, and damage their hosts. However, in the case of *Mycobacteria*, there are distinct similarities between key mechanisms of virulence (Converse and Cox, 2005). More investigation is required into how similar such key pieces of machinery are across strains, species and types, and how possible similarities can be exploited to improve prediction of host-pathogen interactions in newly identified pathogens.

Previous studies from both biological and computational perspectives have focussed on a few key host-pathogen pairs. However, with the variety of infectious diseases already identified, broader and more numerous studies are required to fill in important gaps. Therefore, as the field progresses, it is important that more novel host-pathogen pairs are investigated in the detail that more prominent examples (such as HIV-1, *Mycobacterium tuberculosis*, *Yersinia pestis* and *Plasmodium falciparum*) are.

Computational models are only as good as the data on which they are built. Therefore, it is imperative that the data available for analysing host-pathogen interactions is of the highest possible quality, collected with detailed documentation, labelled and stored in easily accessible ways, and indexed or collated with other data to enable broad analysis across studies and organisms. With a wide variety of different databases available of varying quality which do not contain all current host-pathogen interaction data (as discussed in Section “Databases for Host-Pathogen Interaction” above), there are significant advances to be made in this area. Improvements in text mining methods are a promising avenue for filling in gaps in knowledge; however, they are still lacking in the accuracy required (Thieu *et al.*, 2012). There is also a lack of indexing, maintenance, and reviewing of current host-pathogen

interaction databases, making efforts to find and use already-collected data more laborious than should be required. Therefore, future efforts to curate and review currently available databases, as well as identify areas which are not adequately serviced by current databases, would be extremely valuable.

Closing Remarks

Host-pathogen interactions are vital to our understanding of infectious disease, as well as its treatment and prevention. Through the investigation and analysis of the different stages of infection, the mechanisms by which pathogens invade and proliferate in their hosts can be elucidated. Bioinformatics has and will continue to play a large role in expanding our knowledge in this field, especially in the post-genomic era with the advent of extremely large datasets. Traditional alignment methods, structural analysis, and machine learning methods are all essential to furthering our understanding of host-pathogen interactions, and how such interactions can be exploited to treat disease. The development of novel drugs, vaccines and other therapeutics going into the future will be highly dependent on the knowledge gained from investigating host-pathogen interactions. There is still work to be done in collating and curating data in the field despite significant recent efforts, and this presents an ongoing challenge which must be addressed.

See also: Computational Systems Biology Applications. Extraction of Immune Epitope Information. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics

References

- Ammari, M.G., Gresham, C.R., McCarthy, F.M., Naduri, B., 2016. HPIDB 2.0: A curated database for host-pathogen interactions. *Database*. 1–9. baw103.
- Aranda, B., Achuthan, P., Alam-Farouque, Y., *et al.*, 2010. The IntAct molecular interaction database in 2010. *Nucleic Acids Res.* 38, D525–D531.
- Arnold, R., Brandmaier, S., Kleine, F., *et al.*, 2009. Sequence-based prediction of type III secreted proteins. *PLOS Pathog.* 5 (4).
- Aurrecochea, C., Barreto, A., Basenko, E.Y., *et al.*, 2017. EuPathDB: The eukaryotic pathogen genomics database resource. *Nucleic Acids Res.* 45, D581–D591.
- Aurrecochea, C., Barreto, A., Brestelli, J., *et al.*, 2013. EuPathDB: The eukaryotic pathogen database. *Nucleic Acids Res.* 41, D684–D691.
- Baigent, S.J., McCauley, J.W., 2003. Influenza type A in humans, mammals and birds: Determinants of virus virulence, host-range and interspecies transmission. *Bioessays* 25 (7). 657–671.
- Balakrishnan, S., Tastan, O., Carbonell, J., Klein-Seetharaman, J., 2009. Alternative paths in HIV-1 targeted human signal transduction pathways. *BMC Genom.* 10 (Suppl. 3). S30.
- Bandyopadhyay, S., Ray, S., Mukhopadhyay, A., *et al.*, 2015. A review of in silico approaches for analysis and prediction of HIV-1-human protein-protein interactions. *Brief. Bioinform.* 16 (5). 830–851.
- Becerra, A., Bucheli, V.A., Moreno, P.A., 2017. Prediction of virus-host protein-protein interactions mediated by short linear motifs. *BMC Bioinform.* 163 (18). 1–11.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Res.* 28 (1). 235–242.
- Bhavsar, A.P., Guttman, J.A., Finlay, B.B., 2007. Manipulation of host-cell pathways by bacterial pathogens. *Nature* 449 (7164). 827–834.
- Blevess, S., Dunger, I., Walter, M.C., *et al.*, 2014. HoPaCI-DB: Host-*Pseudomonas* and *Coxiella* interaction database. *Nucleic Acids Res.* 42, D671–D676.
- Brass, A.L., Dykxhoorn, D.M., Benita, Y., *et al.*, 2008. Identification of host proteins required for HIV infection through a functional genomic screen. *Science* 319 (5865). 921–926.
- Calderone, A., Castagnoli, L., Cesareni, G., 2013. Mentha: A resource for browsing integrated protein-interaction networks. *Nat. Methods* 10 (8). 690–691.
- Calderone, A., Licata, L., Cesareni, G., 2015. VirusMentha: A new resource for virus-host protein interactions. *Nucleic Acids Res.* 43, D588–D592.
- Calderwood, M.A., Venkatesan, K., Xing, L., *et al.*, 2007. Epstein-Barr virus and virus human protein interaction maps. *Proc. Natl. Acad. Sci. USA* 104 (18). 7606–7611.
- Casadevall, A., Pirofski, L.A., 1999. Host-pathogen interactions: Redefining the basic concepts of virulence and pathogenicity. *Infect. Immun.* 67 (8). 3703–3713.
- Casadevall, A., Pirofski, L.A., 2000. Host-pathogen interactions: Basic concepts of microbial commensalism, colonization, infection, and disease. *Infect. Immun.* 68 (12). 6511–6518.
- Casadevall, A., Pirofski, L.A., 2001. Host-pathogen interactions: The attributes of virulence. *J. Infect. Dis.* 184, 337–344.
- Chatr-aryamontri, A., Oughtred, R., Boucher, L., *et al.*, 2017. The BioGRID interaction database: 2017 update. *Nucleic Acids Res.* 45, D369–D379.
- Chen, L., Yang, J., Yu, J., *et al.*, 2005. VFDB: A reference database for bacterial virulence factors. *Nucleic Acids Res.* 33, D325–D328.
- Chen, L., Zheng, D., Liu, B., Yang, J., Jin, Q., 2016. VFDB 2016: Hierarchical and refined dataset for big data analysis – 10 years on. *Nucleic Acids Res.* 44, D694–D697.
- Chertova, E., Chertov, O., Coren, L.V., *et al.*, 2006. Proteomic and biochemical analysis of purified human immunodeficiency virus type 1 produced from infected monocyte-derived macrophages. *J. Virol.* 80 (18). 9039–9052.
- Cohen, F.S., 2016. How viruses invade cells. *Biophys. J.* 110 (5). 1028–1032.
- Converse, S.E., Cox, J.S., 2005. A protein secretion pathway critical for *Mycobacterium tuberculosis* virulence is conserved and functional in *Mycobacterium smegmatis*. *J. Bacteriol.* 187 (4). 1238–1245.
- Davey, N.E., Travé, G., Gibson, T.J., 2011. How viruses hijack cell regulation. *Trends Biochem. Sci.* 36 (3). 159–169.
- Davis, F.P., Barkan, D.T., Eswar, N., McKerrow, J.H., Sali, A., 2007. Host-pathogen protein interactions predicted by comparative modelling. *Protein Sci.* 16, 2585–2596.
- Doherty, C.P., 2007. Host-pathogen interactions: The role of iron. *J. Nutr.* 137 (5). 1341–1344.
- Driscoll, T., Dyer, M.D., Murali, T.M., Sobral, B.W., 2009. PiG – The pathogen interaction gateway. *Nucleic Acids Res.* 37, 647–650.
- Durmuş Tekir, S., Çakır, T., Ardiç, E., *et al.*, 2013. PHISTO: Pathogen-host interaction search tool. *Bioinformatics* 29 (10). 1357–1358.
- Dyer, M.D., Murali, T.M., Sobral, B.W., 2007. Computational prediction of host-pathogen protein-protein interactions. *Bioinformatics* 23, i159–i166.
- Dyer, M.D., Murali, T.M., Sobral, B.W., 2008. The landscape of human proteins interacting with viruses and other pathogens. *PLOS Pathog.* 4 (2).
- d'Enfert, C., Ryter, A., Pugsley, A.P., 1987. Cloning and expression in *Escherichia coli* of the *Klebsiella pneumoniae* genes for production, surface localization and secretion of the lipoprotein pullulanase. *EMBO J.* 6 (11). 3531–3538.
- Evans, P., Dampier, W., Ungar, L., Tozeren, A., 2009. Prediction of HIV-1 virus-host protein interactions using virus and host sequence motifs. *BMC Med. Genom.* 2 (27).
- Fahey, M.E., Bennett, M.J., Mahon, C., *et al.*, 2011. GPS-Prot: A web-based visualization platform for integrating host-pathogen interaction data. *BMC Bioinform.* 298 (12).
- Fu, W., Sanders-Beer, B.E., Katz, K.S., *et al.*, 2009. Human immunodeficiency virus type 1, human protein interaction database at NCBI. *Nucleic Acids Res.* 37, D417–D422.

- Galán, J.E., Curtiss, R., 1989. Cloning and molecular characterization of genes whose products allow *Salmonella typhimurium* to penetrate tissue culture cells. *Proc. Natl. Acad. Sci. USA* 86, 6383–6387.
- Ghosh, Z., Mallick, B., Chakrabarti, J., 2009. Cellular versus viral microRNAs in host-virus interaction. *Nucleic Acids Res.* 37 (4), 1035–1048.
- Giraldo-Calderón, G.I., Emrich, S.J., MacCallum, R.M., *et al.*, 2015. VectorBase: An updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases. *Nucleic Acids Res.* 43, D707–D713.
- Guirimand, T., Delmotte, S., Navrátil, V., 2015. VirHostNet 2.0: Surfing on the web of virus/host molecular interactions data. *Nucleic Acids Res.* 43, D583–D587.
- Haynes, B.F., Shaw, G.M., Korber, B., *et al.*, 2016. HIV-host interactions: Implications for vaccine design. *Cell Host Microbe* 19 (3), 292–303.
- Hooper, L.V., Gordon, J.I., 2001. Glycans as legislators of host-microbial interactions: Spanning the spectrum from symbiosis to pathogenicity. *Glycobiology* 11 (2), 1R–10R.
- Hsu, P.W., Lin, L., Hsu, S., Hsu, J.B., Huang, H., 2007. ViTa: Prediction of host microRNAs targets on viruses. *Nucleic Acids Res.* 35, D381–D385.
- Hulo, C., de Castro, E., Masson, P., *et al.*, 2011. ViralZone: A knowledge resource to understand virus diversity. *Nucleic Acids Res.* 39, D576–D582.
- Jaiswal, V., Channumolu, S.K., Gupta, A., Chauhan, R.S., Rout, C., 2013. Jenner-predict server: Prediction of protein vaccine candidates (PVCs) in bacteria based on host-pathogen interactions. *BMC Bioinform.* 211 (14).
- Kelsall, B.L., Biron, C.A., Sharma, O., Kaye, P.M., 2002. Dendritic cells at the host-pathogen interface. *Nat. Immunol.* 3, 699–702.
- Kim, J.K., Kim, T.S., Basu, J., Jo, E.K., 2017. MicroRNA in innate immunity and autophagy during mycobacterium infection. *Cell. Microbiol.* 19, e12687.
- König, R., Stertz, S., Zhou, Y., *et al.*, 2010. Human host factors required for influenza virus replication. *Nature* 463 (7282), 813–817.
- König, R., Zhou, Y., Elleder, D., *et al.*, 2008. Global analysis of host-pathogen interactions that regulate early-stage HIV-1 replication. *Cell* 135, 49–60.
- Krishnadev, O., Srinivasan, N., 2008. A data integration approach to predict host-pathogen protein-protein interactions: Application to recognize protein interactions between human and a malarial parasite. *In Silico Biol.* 8 (3), 235–250.
- Kshirsagar, M., Carbonell, J., Klein-Seetharaman, J., 2012. Techniques to cope with missing data in host-pathogen protein interaction prediction. *Bioinformatics* 28 (18), i466–i472.
- Kshirsagar, M., Carbonell, J., Klein-Seetharaman, J., 2013. Multi-task learning for host-pathogen protein interactions. *Bioinformatics* 29 (13), i217–i226.
- Kuldau, G.A., De Vos, G., Owen, J., McCaffrey, G., Zambryski, P., 1990. The *virB* operon of *Agrobacterium tumefaciens* pTiC58 encodes 11 open reading frames. *Mol. Gen. Genet.* 221, 256–266.
- Kumar, S., Jena, L., Daf, S., *et al.*, 2013. hvpPDB: An online proteome reserve for human papillomavirus. *Genom. Inform.* 11 (4), 289–291.
- Kumar, R., Naduri, B., 2010a. HPIDB – A unified resource for host-pathogen interactions. *BMC Bioinform.* 11, 1–6.
- Kumar, R., Naduri, B., 2010b. HPIDB – A unified resource for host-pathogen interactions. *BMC Bioinform.* 11 (Su), S16.
- Kurz, C.L., Ewbank, J.J., 2000. *Caenorhabditis elegans* for the study of host-pathogen interactions. *Trends Microbiol.* 8 (3), 142–144.
- Meijer, A.H., Spaink, H.P., 2011. Host-pathogen interactions made transparent with the zebrafish model. *Curr. Drug Targets* 12, 1000–1017.
- Methot, P.O., Alizon, S., 2014. What is a pathogen? Toward a process view of host-parasite interactions. *Virulence* 8 (5), 775–785.
- Mougous, J.D., Cuff, M.E., Raunser, S., *et al.*, 2006. A virulence locus of *Pseudomonas aeruginosa* encodes a protein secretion apparatus. *Science* 312 (5779), 1526–1530.
- Munguia, J., Nizet, V., 2017. Pharmacological targeting of the host-pathogen interaction: Alternatives to classical antibiotics to combat drug-resistant superbugs. *Trends Pharmacol. Sci.* 38 (5), 473–488.
- Nau, G.J., Richmond, J.F.L., Schlesinger, A., *et al.*, 2002. Human macrophage activation programs induced by bacterial pathogens. *Proc. Natl. Acad. Sci. USA* 99 (3), 1503–1508.
- Navrátil, V., de Chassey, B., Meyniel, L., *et al.*, 2009. VirHostNet: A knowledge base for the management and the analysis of proteome-wide virus-host interaction networks. *Nucleic Acids Res.* 37, D661–D668.
- Pickett, B.E., Sadat, E.L., Zhang, Y., *et al.*, 2012. ViPR: An open bioinformatics database and analysis resource for virology research. *Nucleic Acids Res.* 40, D593–D598.
- Pinney, J.W., Dickerson, J.E., Fu, W., *et al.*, 2009. HIV-host interactions: a map of viral perturbation of the host system. *AIDS* 23 (5), 549–554.
- Pohlner, J., Halter, R., Beyreuther, K., Meyer, T.F., 1987. Gene structure and extracellular secretion of *Neisseria gonorrhoeae* IgA protease. *Nature* 325, 458–462.
- Pukatzki, S., Ma, A.T., Sturtevant, D., *et al.*, 2006. Identification of a conserved bacterial protein secretion system in *Vibrio cholerae* using the *Dictyostelium* host model system. *Proc. Natl. Acad. Sci.* 103 (5), 1528–1533.
- Qi, Y., Tastan, O., Carbonell, J.G., *et al.*, 2010. Semi-supervised multi-task learning for prediction interactions between HIV-1 and human proteins. *Bioinform.* 26, i645–i652.
- Raghunathan, A., Reed, J., Shin, S., Palsson, B., Daefler, S., 2009. Constraint-based analysis of metabolic capacity of *Salmonella typhimurium* during host-pathogen interaction. *BMC Syst. Biol.* 3 (1).
- Rahman, M.M., McFadden, G., 2011. Modulation of NF- κ B signalling by microbial pathogens. *Nat. Rev. Microbiol.* 9 (4), 291–306.
- Rappoport, N., Linial, M., 2012. Viral proteins acquired from a host converge to simplified domain architectures. *PLOS Comput. Biol.* 8 (2).
- Samant, M., Chadha, N., Tiwari, A.K., *et al.*, 2016. In silico designing and analysis of inhibitors against target protein identified through host-pathogen interactions in malaria. *Int. J. Med. Chem.* 2016, 2741038.
- Scaria, V., Hariharan, M., Pillai, B., Maiti, S., Brahmachari, S.K., 2007. Host-virus genome interactions: Macro roles for microRNAs. *Cell. Microbiol.* 9 (12), 2784–2794.
- Schaefer, C.F., Anthony, K., Krupa, S., *et al.*, 2009. PID: The pathway interaction database. *Nucleic Acids Res.* 37, D674–D679.
- Stanley, S.A., Raghavan, S., Hwang, W.W., *et al.*, 2003. Acute infection and macrophage subversion by *Mycobacterium tuberculosis* require a specialized secretion system. *Proc. Natl. Acad. Sci. USA* 100 (22), 13001–13006.
- Talaat, A.M., Lyons, R., Howard, S.T., Johnston, S.A., 2004. The temporal expression profile of *Mycobacterium tuberculosis* infection in mice. *Proc. Natl. Acad. Sci. USA* 101 (13), 4602–4607.
- Tastan, O., Qi, Y., Carbonell, J.G., Klein-Seetharaman, J., 2009. Prediction of interactions between HIV-1 and human proteins by information integration. *Pac. Symp. Biocomput.* 2009, 516–527.
- Tato, C.M., Hunter, C.A., 2002. Host-pathogen interactions: Subversion and utilization of the NF- κ B pathway during infection. *Infect. Immun.* 70 (7), 3311–3317.
- Thieu, T., Joshi, S., Warren, S., Korkin, D., 2012. Literature mining of host-pathogen interactions: Comparing feature-based supervised learning and language-based approaches. *Bioinformatics* 28 (6), 1357–1358.
- Tobin, D.M., May, R.C., Wheeler, R.T., 2012. Zebrafish: A see-through host and fluorescent toolbox to probe host-pathogen interaction. *PLOS Pathog.* 8 (1).
- Urban, M., Cuzick, A., Rutherford, K., *et al.*, 2017. PHI-base: A new interface and further additions for the multi-species pathogen-host interactions database. *Nucleic Acids Res.* 45, D604–D610.
- Vergne, I., Singh, S., Roberts, E., *et al.*, 2006. Autophagy in immune defense against *Mycobacterium tuberculosis*. *Autophagy* 2 (3), 175–178.
- Via, A., Uyar, B., Brun, C., Zanzoni, A., 2015. How pathogens use linear motifs to perturb host cell networks. *Trends Biochem. Sci.* 40 (1), 36–48.
- Vodovar, N., Acosta, C., Lemaître, B., Boccad, F., 2004. *Drosophila*: A polyvalent model to decipher host-pathogen interactions. *Trends Microbiol.* 12 (5), 235–242.
- Wang, Y., Zhang, Q., Sun, M., Guo, D., 2011. High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. *Bioinformatics* 27 (6), 777–784.
- Wattam, A.R., Abraham, D., Dalay, O., *et al.*, 2014. PATRIC, the bacterial bioinformatics database and analysis resource. *Nucleic Acids Res.* 42, 581–591.
- Weiberg, A., Wang, M., Lin, F., *et al.*, 2013. Fungal small RNAs suppress plant immunity by hijacking host RNA interference pathways. *Science* 342 (6154), 118–123.
- Welch, R.A., Dellinger, E.P., Minshew, B., Falkow, S., 1981. Haemolysin contributes to virulence of extra-intestinal *E. coli* infections. *Nature* 294, 665–667.
- Winnenberg, R., Baldwin, T.K., Urban, M., *et al.*, 2006. PHI-base: A new database for pathogen-host interactions. *Nucleic Acids Res.* 34, D459–D464.
- Worbs, T., Hammerschmidt, S.I., Förster, R., 2017. Dendritic cell migration in health and disease. *Nat. Rev. Immunol.* 17, 30–48.
- Wuchty, S., 2011. Computational prediction of host-parasite protein interactions between *P. falciparum* and *H. sapiens*. *PLOS ONE* 6 (11).

- Xiang, Z., Tian, Y., He, Y., 2007. PHIDIAS: A pathogen-host interaction data integration and analysis system. *Genome Biol.* 8 (7). R150.
- Yang, W., Yan, L., Wu, C., Zhao, X., Tang, J., 2014. Fungal invasion of epithelial cells. *Microbiol. Res.* 169, 803–810.
- Zagursky, R.J., Olmsted, S.B., Russell, D.P., Wooters, J.L., 2003. Bioinformatics: How it is being used to identify bacterial vaccine candidates. *Expert Rev. Vaccines* 2 (3). 417–436.
- Zhang, Y., Aebermann, B.D., Anderson, T.K., *et al.*, 2017. Influenza Research Database: An integrated bioinformatics resource for influenza virus research. *Nucleic Acids Res.* 45, D466–D474.

Further Reading

- Carvalho Leite, D.M., Brochet, X., Resch, G., *et al.*, 2017. Computational prediction of host-pathogen interactions through omics data analysis and machine learning. In: Rojas, I., Ortuño, F. (Eds.), *Bioinformatics and Biomedical Engineering. IWBBIO 2017. Lecture Notes in Computer Science 10209*. Springer.
- Costa, T.R.D., Felisberto-Rodrigues, C., Meir, A., *et al.*, 2015. Secretion systems in Gram-negative bacteria: Structural and mechanistic insights. *Nat. Rev. Microbiol.* 13, 343–359.
- Durmuş, S., Çakır, T., Özgür, A., Guthke, R., 2015. A review on computational systems biology of pathogen-host interactions. *Front. Microbiol.* 6, 235.
- Mariano, R., Wuchty, S., 2017. Structure-based prediction of host-pathogen protein interactions. *Curr. Opin. Struct. Biol.* 44, 119–124.
- Marston, H.D., Folkers, G.K., Morens, D.M., Fauci, A.S., 2014. Emerging viral diseases: Confronting threats with new technologies. *Sci. Transl. Med.* 6 (253). 253ps10.
- Pan, A., Lahiri, C., Rajendiran, A., Shanmugham, B., 2016. Computational analysis of protein interaction networks for infectious diseases. *Brief. Bioinform.* 17 (3). 517–526.

Biographical Sketch



Dean Southwood is a PhD student in the Department of Molecular Sciences at Macquarie University. His background is in theoretical quantum physics, quantum computation, and bioinformatics. His current research interests involve determining novel biomarkers for cancer and disease using machine learning methods.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books in as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.

Computational Pipelines and Workflows in Bioinformatics

Yosvany López, Genesis Healthcare Co., Tokyo, Japan

Piotr J Kamola, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan and Japan Science and Technology Agency, Tokyo, Japan

Ronesh Sharma, University of the South Pacific, Suva, Fiji and Fiji National University, Suva, Fiji

Daichi Shigemizu, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; Japan Science and Technology Agency, Tokyo, Japan; RIKEN Cluster for Science and Technology Hub, Yokohama, Japan; National Center for Geriatrics and Gerontology, Obu, Japan; and Tokyo Medical and Dental University, Tokyo, Japan

Tatsuhiko Tsunoda, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; Japan Science and Technology Agency, Tokyo, Japan; and Tokyo Medical and Dental University, Tokyo, Japan

Alok Sharma, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; University of the South Pacific, Suva, Fiji; and Griffith University, Brisbane, QLD, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The research field of bioinformatics is gaining momentum due to its role in the examination of huge volumes of biological data. Nowadays, technological advancements in sequencer and microscope technology have opened many novel and interesting avenues to study organisms at the molecular level. These advances should be accompanied by efficient computational tools and pipelines, geared towards processing and analysis of the data. This combination will allow for more accurate quantification of molecular interaction, and lead to better understanding of the underlying mechanisms. In the field of molecular biology, next-generation sequencing (NGS) technology is playing a key role already. For instance, studies aimed at assessing DNA–protein interactions often make use of chromatin immunoprecipitation-sequencing (ChIP-seq) data, whereas those focused on trying to understand the genetics behind rare diseases or cancers take advantage of whole genome (WGS) or whole exome sequencing (WES) data. Studies that focus on profiling coding and noncoding RNA on a larger scale have the choice of DNA-chip and RNA-sequencing (RNA-seq) technologies. Complex cellular structures can be studied with microscopes that are capable of generating high-quality images. Given the above circumstances, computational algorithms and methodologies are necessary to face such challenges (Roy *et al.*, 2016, 2018; Metzker, 2010). Bioinformatics pipelines combine a set of software tools, each designed to deal with a specific task, to take full advantage of the biological information.

In this article, we discuss the computational pipelines and workflows used for analyzing a wide range of sequencing and biomedical image information. Accordingly, it is hereafter divided into five sections. Section “Genome Analysis” covers genome analysis pipelines, especially tools for dealing with ChIP-seq and genomic variation. Section “Transcriptome Analysis” describes transcriptome strategies, including differential expression analysis, pathway and network approaches, and machine learning (ML) methods for sample subclassification and biomarker discovery. Section “Workflows in Proteomics” focuses on proteomics, specifically on molecular recognition of features in intrinsically disordered proteins (IDPs). Section “Bioimage Analysis Pipelines” briefly introduces the standard workflow for processing bioimages from cellular structures, while Section “Conclusions” highlights the challenges and future directions.

Genome Analysis

The function of molecular structures is ultimately encoded in the genome sequence, thereby bioinformatics pipelines designed for its analysis are of utmost importance. In this section, we present the workflow and tools available for effectively analyzing ChIP-seq and genomic variation data.

ChIP-Seq Pipeline

A great number of studies have been currently focusing on deciphering the molecular consequences of DNA–protein interactions. In this case, NGS data has played a key role because it allows us to sequence the DNA sequence around binding sites. Subsection “ChIP-Seq Pipeline” introduces the standard workflow (Fig. 1) used by many laboratories for such analyses.

Alignment

Because read quality is often affected by artifacts such as adapter contamination and base calling errors, the quality check of reads is sometimes the first procedure before mapping the raw reads to a reference genome. It is here where biases or sequencing errors are identified and low-quality reads are filtered out. Given its flexibility and applicability to many sequencing platforms, the software FastQC (Babraham-Bioinformatics, 2018) is frequently utilized. A more detailed explanation and additional tools can be found in Pabinger *et al.* (2014).

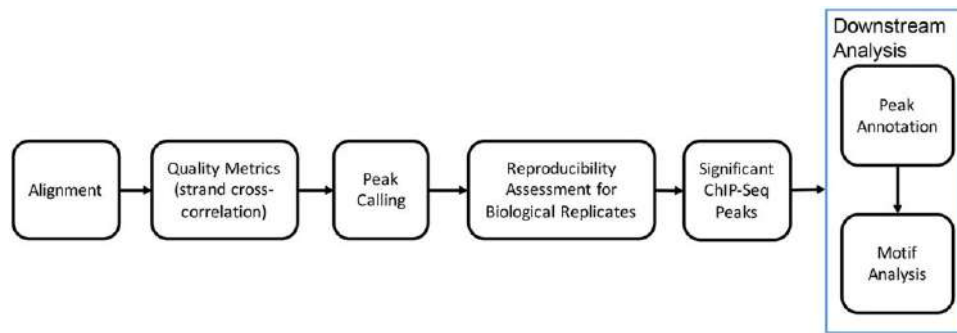


Fig. 1 Standard workflow for the analysis of ChIP-seq data.

After conducting the above filtering, high-quality reads are aligned to the corresponding reference genome using aligners such as Bowtie2 (Langmead and Salzberg, 2012), SOAP (Li *et al.*, 2009b), BWA (Li and Durbin, 2009) and MAQ (Li *et al.*, 2008). It is worth noting that unlike transcriptome analysis pipelines, aligners capable of detecting splice sites are not required whatsoever. The alignment results (the number of uniquely mapped reads) should be carefully checked to validate the success of the mapping procedure.

Quality metrics

When processing ChIP-seq data, the assessment of immunoprecipitation enrichment, sequencing depth or fragment-size selection is a must. This is often referred to as the signal-to-noise ratio and assessed with metrics like strand cross-correlation (Landt *et al.*, 2012) or immunoprecipitation enrichment estimation (Landt *et al.*, 2012). Although the package CHANCE (Diaz *et al.*, 2012) is widely utilized, current peak callers such as MACS (Zhang *et al.*, 2008) already include strand cross-correlation analysis.

Peak calling

Peak calling is one of the most important steps in a ChIP-seq analysis pipeline. It is here where bound DNA regions are detected. Of note, if the improvement of specificity is prioritized, duplicate reads should be removed before peak calling. Moreover, the success of this step will depend on the peak caller and the type of protein being analyzed. For the purpose of analysis, DNA-binding proteins (DBPs) are divided into three groups depending on their binding signals: point-source DBPs, broadly enriched DBPs, and DBPs with mixed characteristics.

Point-source DBPs

These proteins are the most common and therefore many peak callers have been designed to deal with their signals. For instance, peak callers like MACS (Zhang *et al.*, 2008) and SPP (Kharchenko *et al.*, 2008) are able to accurately detect the size of the genomic gap between reads aligned to plus and minus strands. Other software such as BEADS (Cheung *et al.*, 2011) use background models from control samples or GC content for reducing noise. To assess enriched peaks, software such as MACS (Zhang *et al.*, 2008) and CisGenome (Jiang *et al.*, 2010), which rely on statistical distributions, or BayesPeak (Spyrou *et al.*, 2009) and HPeak (Qin *et al.*, 2010), which implement hidden Markov models, are often utilized.

Broadly enriched DBPs

These proteins are usually involved in epigenetic regulation. For their analysis, peak callers such as SICER (Xu *et al.*, 2014) and ZINBA (Rashid *et al.*, 2011) can be utilized to detect broad regions, whereas others such as MACS (Zhang *et al.*, 2008), PeakRanger (Feng *et al.*, 2011), and SPP (Kharchenko *et al.*, 2008) can be used as long as their bandwidth is increased and their peak cut-off decreased.

DBPs with mixed signals

For analyzing molecules with mixed characteristics, for example the RNA Polymerase II, peak callers such as MACS (Zhang *et al.*, 2008), ZINBA (Rashid *et al.*, 2011), and PeakRanger (Feng *et al.*, 2011) can be employed.

Assessment of reproducibility

Because the quality of detected peaks could be affected by the peak caller, sequencing depth, or the number of binding sites, pipelines should always include an assessment of reproducibility. This procedure is intended to check whether the peaks are really reproducible, and should comprise at least two replicates of the same experiment. To do this, it is advisable to filter out those regions with high ChIP signals and then compute the Pearson correlation coefficient of mapped reads at genomic positions. The consistency of peak sets detected in biological replicates is evaluated with the irreproducible discovery rate, and those peaks passing a specific cut-off are finally retrieved. This assessment is often conducted as described in Li *et al.* (2011).

Differential binding analysis

A typical ChIP-seq experiment aims to detect binding differences between biological conditions. Current studies follow two strategies: quantitative and qualitative. The quantitative approach takes into consideration the differential binding between conditions by using the number of reads or the density of reads in peaks. For this, tools such as DBChIP (Liang and Keleş, 2012), which relies on read counts or MANorm (Shao *et al.*, 2012), which uses read densities, can be utilized. The qualitative approach considers overlapping peaks and should only be regarded for exploration purposes. It is important to point out that reproducible peaks should be computed per condition before regarding one of the above strategies.

Downstream analysis

This step comprises important tasks such as peak annotation and motif analysis. The detection of peaks is often not the ultimate goal, but it is followed by the discovery of functional genomic regions. To do this, the peaks should be first formatted in BED or GFF files so that they can be easily analyzed with a genome browser, or annotated with in-house scripts. Current packages such as BEDTools (Quinlan and Hall, 2010) include functions for annotation purposes and calculate the genomic distance of each peak to relevant genomic features. Other libraries like ChIPpeakAnno (Zhu *et al.*, 2010) can be further used for associating binding sites with gene expression activity. On the other hand, motif analysis could also help to identify those genomic sites bound by a specific regulatory protein, thereby facilitating the validation of the ChIP experiment. To detect motif sequences, the genomic regions of the detected peaks are analyzed with motif-discovery software. These algorithms complement each other so that it is advisable to consider different methods. For instance, algorithms such as MEME-ChIP (Machanick and Bailey, 2011) and peak-motifs (Thomas-Chollier *et al.*, 2012) are often employed. Detailed information on motif analysis can be found in Zambelli *et al.* (2013).

Mutation and Genomic Variation

The low cost of NGS technology has made possible international collaborations such as the 1000 Genomes (Genomes Project *et al.*, 2015) and HapMap (International Hapmap Consortium, 2005) Projects. These efforts have provided detailed catalogs of human genetic variation, which are of vital importance to identify disease-associated variants. Consequently, recent studies have aimed at pinpointing genetic risk factors for common diseases, including rheumatoid arthritis and type 2 diabetes, cancers, and Mendelian diseases such as long QT syndrome and Huntington's disease. Because the accurate detection of variants proves critical for clinical treatments and novel drugs, the main challenges are no longer in the sequencing technology but in the development of suitable bioinformatics pipelines. This section addresses two impacted areas: genome-wide association studies (GWAS), and whole genome and exome analysis.

Genome-wide association studies

The genetics behind complex diseases has been long studied by focusing on disease-related genes. However, the massive amounts of genotype data have paved the way for broader approaches like GWAS, which regard linkage disequilibrium at the population level. GWAS has proven absolutely necessary for identifying the genetic risk factors for common diseases (Ozaki *et al.*, 2002; Newton-Cheh *et al.*, 2009), and investigating associations between single nucleotide polymorphisms (SNPs) from case/control studies. The most common tool for GWAS analysis is PLINK (Purcell *et al.*, 2007). PLINK is an open-source and freely available software, which also provides quality controls of the data. It requires two standard formats: PED and MAP files, though binary files such as BED, BIM, and FAM are acceptable as well. The PED file is a space or tab delimited file, composed of a family identifier, individual identifier, father and mother identifiers, gender, phenotype, and genotypes. Of note, when creating a PED file, strand orientation should be checked for allele calls (i.e., forward or reverse complement). The TOP/BOT strand and A/B allele coding, developed by Illumina, along with the database of genetic variation dbSNP (Sherry *et al.*, 2001) are widely used for uniformly designating SNP entries. The MAP file consists of a chromosome, identifier, genetic distance, and physical position for each marker. As for the experimental dataset, it should pass different quality checks in which the samples and markers are carefully examined. The samples should be checked for (1) sex inconsistencies (`-check-sexa`), (2) inbreeding coefficient (`-heta 0.1`), (3) genotype missingness (`-missinga 0.05`), (4) kinship coefficient (`-genomea 0.2`), and (5) population stratification. Similarly, the markers should be checked for (1) genotyping efficiency or call rate (`-genoa 0.99`), (2) minor allele frequency (`-freqa 0.01`), and (3) Hardy-Weinberg equilibrium (`-hwea 0.001`). The remaining dataset can now be used for association analysis (`-assoca`) whose significance cut-off should be set at $p < 5 \times 10^{-8}$. These association results can be further visualized with Q-Q and Manhattan plots included in the R package *qqman*.

Often, the number of markers in an experimental dataset is not enough for accurate analysis, limiting the power and resolution of the GWAS strategy. To overcome this limitation, genotype imputation at ungenotyped loci is strongly recommended. The most common imputation software include IMPUTE2 (Marchini *et al.*, 2007), BEAGLE (Browning and Browning, 2009), and minimac (Neumann *et al.*, 2010). Of note, reference panels from the HapMap (International Hapmap Consortium, 2005) or 1000 Genomes (Genomes Project *et al.*, 2015) Projects should be used.

Thus far, we have described a typical SNP-based association analysis. However, gene-based association approaches can sometimes turn out effective. These statistical methods are classified into three categories: (1) the burden test (Li and Leal, 2008; Madsen and Browning, 2009; Morgenthaler and Thilly, 2007; Price *et al.*, 2010), (2) the variance component test (Wu *et al.*, 2011),

^aPLINK parameter

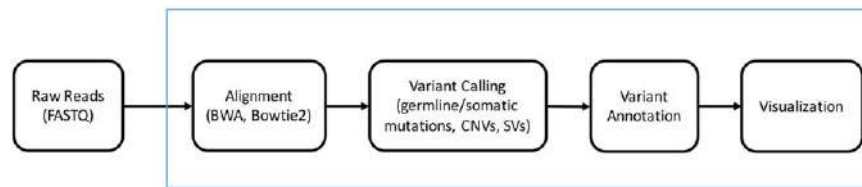


Fig. 2 Standard workflow for whole genome/exome sequencing data analysis.

and (3) the combination methods (Lee *et al.*, 2012). Recently, a method called SMR, which integrates summary-level data of GWAS and expression quantitative trait locus, was developed (Zhu *et al.*, 2016). Additional examples of risk prediction models for several diseases, which regarded disease-associated genes, can be found in Imamura *et al.* (2013); Shigemizu *et al.* (2014).

Genome and exome analysis pipelines

For rare Mendelian diseases, NGS technology (Metzker, 2010; Rusk and Kiermer, 2008) has made WGS and WES possible at an individual level. Extensive studies have revised the pipelines aimed to detect genetic variations by analyzing NGS data (Hwang *et al.*, 2015; Pabinger *et al.*, 2014). A standard pipeline consists of five main steps: quality check of raw reads, alignment to a reference genome, variant identification, variant annotation, and visualization (Fig. 2). The first two steps are discussed in Section “Alignment” so that we will focus on the remaining steps.

Variant identification

The identification of genomic variants is the main step of a variant detection pipeline. This step is intended to detect genomic variations between biological samples. Most of the available software require the aligned reads in BAM format and sorted by genomic coordinates. Both requirements can be easily achieved with SAMtools (Li *et al.*, 2009a).

The current variant identification tools can be used for detecting (1) germline mutations, (2) somatic mutations, (3) copy number variations (CNVs), and (4) structural variations (SVs). Among the variant callers for discovering germline mutations are the Genome Analysis Tool Kit (GATK) HaplotypeCaller (McKenna *et al.*, 2010), SAMtools (Li *et al.*, 2009a) and VarScan 2 (Koboldt *et al.*, 2012). The GATK includes a statistical model based on Bayesian genotype likelihood for computing genotypes and allele frequencies. Nevertheless, one disadvantage is that it produces many false positives which have to be filtered for downstream analysis (McKenna *et al.*, 2010). SAMtools can be easily integrated into one single pipeline because of its additional functions for aligning reads, as well as sorting and indexing alignment results (Li *et al.*, 2009a). VarScan 2 implements heuristic and statistical approaches for detecting single nucleotide variants, indels, somatic mutations, and copy number alterations between cancer and control samples (Koboldt *et al.*, 2012). SAMtools and VarScan 2, along with SomaticSniper (Larson *et al.*, 2012) can be also employed for identifying somatic mutations. SomaticSniper statistically compares the genotype likelihoods of cancer and control samples, which could turn out useful for cancer-related studies (Larson *et al.*, 2012). Tools such as CNVnator (Abyzov *et al.*, 2011) and ExomeCNV (Sathirapongsasuti *et al.*, 2011) were mainly developed for identifying CNVs. CNVnator combines the mean-shift approach with multiple-bandwidth partitioning and GC correction for detecting atypical CNVs like *de novo* and multiallelic events. However, it often misses CNVs generated by retrotransposable elements (Abyzov *et al.*, 2011). On the other hand, ExomeCNV was specifically created for identifying loss of heterozygosity in addition to CNVs. This software is able to correctly discover small indels due to the inclusion of B-allele frequencies and depth-of-coverage (Sathirapongsasuti *et al.*, 2011). For detecting SVs such as inversions, translocations, or large indels, the software CLEVER is still widely utilized because of its efficient use of internal segment sizes (Marschall *et al.*, 2012).

According to one study that revised several pipelines for whole exome analysis, the combined use of the aligner BWA and SAMtools, or of any aligner with Freebayes, turns out suitable for SNP calling. This study also claims the software GATK (HaplotypeCaller) as the best variant caller for indels, and Freebayes as the most convenient tool when low-quality variants are previously filtered out (Hwang *et al.*, 2015).

Variant annotation

After identifying variants, the detected variants are annotated to understand their biological implications. This step aims to filter out irrelevant variants and retrieve those disease-related ones. Although the current tools are able to annotate SNPs and indels, the annotation of CNVs has not been fully implemented. The available software includes ANNOVAR (Wang *et al.*, 2010), Vcfanno (Pedersen *et al.*, 2016), and VarMatch (Sun and Medvedev, 2017). ANNOVAR can be utilized for annotating SNVs and indels. It analyzes the function of those genes harboring variants, detects variants in conserved genomic stretches, and predicts cytogenetic bands (Wang *et al.*, 2010). Vcfanno includes a “chromosome sweeping” approach, making possible the annotation of a large number of variants (Pedersen *et al.*, 2016). VarMatch can be used when dealing with dense areas of variants or low complexity genomic regions. It implements an optimization strategy known as edit distance, which contributes to improving robustness (Sun and Medvedev, 2017). Furthermore, the detected variants should be matched against those in mutation databases such as dbSNP (Sherry *et al.*, 2001) or COSMIC (Forbes *et al.*, 2008). Consequently, the variants could be classified as deleterious or accepted variants.

Visualization

Variant visualization is also an essential step because it offers better interpretations of the experiment. There are many genome browsers that facilitate the visualization of genomic regions. Some are useful for visualization and annotation, for example, ABrowse (Kong *et al.*, 2012), Apollo (Lee *et al.*, 2013), Ensembl (Ruffier *et al.*, 2017), and GenomeView (Abeel *et al.*, 2012). Others are specifically designed for visualization, including BamView (Carver *et al.*, 2013), JBrowse (Buels *et al.*, 2016), and MapView (Bao *et al.*, 2009). An extensive review of visualization tools can be found in Pabinger *et al.* (2014).

Transcriptome Analysis

Alterations at the genome level, which can be detected using DNA-sequencing approaches such as WGS, are responsible for a wide range of adverse health phenotypes. Such diseases can often be linked to a single mutation within a crucial gene or accumulation of changes that, when combined, can lead to cancer or age-related illnesses (Podolskiy *et al.*, 2016). However, most complex biological mechanisms arise through the interplay of hundreds of interactome elements, with the disease often being a result of perturbation to those systems and networks (Vidal *et al.*, 2011). The most accessible genomic layer, in terms of easiness of profiling and basic examination, is the transcriptome. A global view of gene activity, which takes place at the RNA level, provides means to perform a wide variety of studies. This includes analysis of cellular functions, identification of predictive biomarkers and diseases subtypes, or pathway and network-oriented studies that can more comprehensively explain changes in the state of biological systems (Brodie *et al.*, 2014). Here we will present several computational approaches to study the transcriptome, and avenues to explore when linking such profiles to actionable results. Emphasis will be placed on popular, open-source tools and libraries and on the fundamentals, which can be easily built upon and extended.

Profiling Platforms

The two most widespread platforms for studying global gene changes are NGS and DNA microarrays. The former, a high-throughput approach that superseded Sanger sequencing, relies on “reading” millions of small RNA fragments in parallel. As each RNA transcript is represented by multiple overlapping fragments, it is possible to combine the sequence pieces together by mapping them to a specie-specific reference (Behjati and Tarpey, 2013). DNA microarray (or DNA-chip) technology, on the other hand, uses a collection of DNA probes attached to a solid surface, with each oligonucleotide probe cluster designed to match a specific short region from a gene of interest. Binding a complementary sequence produces a signal (generated by a fluorophore-labeled antibody), whose strength is used to quantify the abundance of the corresponding target gene (Shalon *et al.*, 1996). While the choice between platforms is often based on personal preference and technical capabilities available within a given group, there are several factors that should be considered when making the decision. Microarrays contain a set of predefined target genes, which makes the data processing simple and straightforward. The methodology has matured to a point where the process can be accomplished using a single package – an overview using a representative tool is provided in Section “DNA-Chip Data Processing.” While convenient, the technology does not offer as much flexibility as NGS alternatives (Hurd and Nelson, 2009). If a project would benefit from detecting novel or rare transcripts, fusion genes, or splice variant analysis, or requires higher confidence in the findings (which can be achieved by increasing the sequencing depth of coverage), NGS methods offer an advantage. The analysis of RNA-seq data does, however, require some understanding of the process on the molecular level, as well as familiarity with additional software tools. Each step of the process is described in Section “RNA-Seq Data Processing,” and a graphical overview of the whole transcriptome section is shown in Fig. 3.

DNA-Chip Data Processing

In this section, we will provide a basic overview of initial DNA microarray data processing using limma package (Ritchie *et al.*, 2015) within R software environment. While these steps can be accomplished using manufacturer provided software solutions, limma offers more built-in functionalities and can be easily integrated with other bioinformatics tools. Furthermore, a similar methodology can be utilized for other types of arrays, such as those profiling protein expression levels. R is a statistical computing environment that is widely used for exploratory data analytics and visualization. It can be installed on any operating systems by following the instructions found on the project’s website (see “Relevant Websites” section). Subsequently, a wide range of genomic packages can be downloaded from Bioconductor, a repository for high-throughput genomic analysis tools (see “Relevant Websites” section).

The initial chip processing step is automatically performed by the image analysis software, where the TIFF scan of the array is translated into a list of intensities for each of the small “spots” (i.e., areas with multiple copies of a unique DNA probe). limma provides a read function that loads and converts data (together with sample annotation) from a wide range of microarray formats. Probe annotation, usually stored in GenePix Array List (GAL) format, should be provided separately, in case it is not included in the output files.

Issues with the sample quality can be introduced at any stage of preparation, from low-quality biological material to problems during scanning. It is thus important to detect such problems early during analysis. There are two common quality control

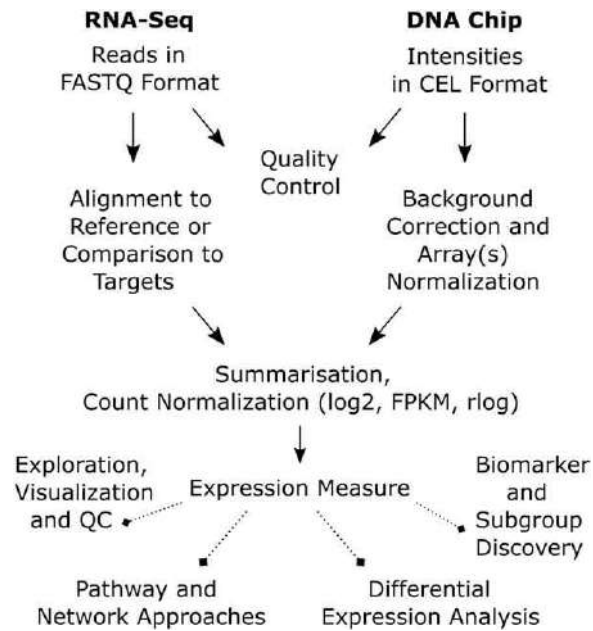


Fig. 3 Graphical overview of the standard pipeline for transcriptome data analysis.

strategies that are applicable to both individual spots within arrays (much higher complexity) and to cross-array comparison – assigning quality weights that are used in a downstream analysis or removing problematic cases (Kauffmann and Huber, 2010). limma has built-in functions to cover the former, which calculate spot weight quality as well as array variance (Ritchie *et al.*, 2006). Lower scores are assigned to spots with quality issues and to arrays that are less reproducible, ensuring that results are not affected by bad samples. Alternatively, arrayQualityMetrics package (Kauffmann, 2009) can be used to calculate and visualize the distance between arrays and exclude those that are significantly different from the rest.

When measuring the fluorescence intensities there are several factors that affect the scan, by introducing ambient noise, both within and between the arrays (Silver *et al.*, 2009). Such signals are adjusted with background correction algorithms, which estimate and remove nonspecific binding or spatial heterogeneity effects (Ritchie *et al.*, 2007). Plotting the foreground against the background intensities can guide the choice of optimal correction algorithm. Lastly, the samples should be normalized to ensure that the biological signal is not “diluted” by technical variation or biases. Depending on the type (single- or two-color) and technology of the array, limma provides several approaches to normalize (i.e., remove some of the variations) within and between the arrays. This ensures that all samples are on the same measuring scale and that technical differences are not influencing the results. The choice of algorithms for background correction and normalization is the only challenging step in the processing of DNA microarrays, and both should be guided by the assessment of the data. As the subsequent steps in the analysis are common between sequencing and DNA-chip platforms, they will be described together in Section “Differentially Expressed Genes and Visualization.”

RNA-Sequencing Data Processing

The field of NGS profiling encompasses a wide range of protocols, though from the analysis standpoint the main difference lies in the type of sequence that is enriched during library preparation. These can include small regulatory RNA such as miRNA, total or coding RNA, degraded RNA (German *et al.*, 2008), or RNA amplified from a single cell rather than a population of cells (Eberwine *et al.*, 2014). While there are differences in how each of the types is analyzed, the initial data processing steps and general principles are similar. The section below will focus on processing data from total or polyA selected RNAs, though with small changes to the reference the methodology can be applied to other types of protocols. For the purpose of this section, we will presume that the sequencing was performed on Illumina’s HiSeq or MiSeq machine, and that demultiplexing (the process of separating individual libraries that were sequenced on the same lane) was performed during data export to FASTQ files. FASTX-Toolkit (Hannon, 2010) and EMBOS seqret (Cock *et al.*, 2010) are two useful tools for checking, manipulating, and converting FASTQs to match the input format requirements of other programs.

The quality of reads, or short sequences obtained at the end of NGS protocols, depends on the integrity of the initial sample, proficiency in library construction, and handling during final sequencing stage. FastQC (Babraham-Bioinformatics, 2018) is a quality control tool that generates comprehensive graphical reports on several important characteristics of any sequence data. It should be noted that modern alignment algorithms can deal with cases such as adapter (short DNA sequences used in library preparation and sequencing) contamination or when part of the reads are of lower quality. However, in more severe cases,

trimming the sequences or removing the adapters can increase the quality and number of aligned reads. This can be achieved using tools such as Cutadapt (Martin, 2012), FASTX (Hannon, 2010) or Trimmomatic (Bolger *et al.*, 2014). It is important to repeat the quality control step after each manipulation to ensure that the overall quality of the library has improved. If the entire length of the reads is of very low quality (below the score of 20 for Illumina's version 1.3 or later encoding), adapters make up most of the reads, or there is a significant overexpression of few "background" genes (such as ribosomal RNAs), it is usually advisable to repeat the sample and/or library preparation. The aims of the project and type of data will naturally influence the software stack and parameters used in the data processing. Differential splicing and metatranscriptomics assembly require a more complex setup, though the principles are the same once the fundamentals are understood. For the purpose of this and the next three subsections, we will focus on transcriptome profile comparison between two phenotypes or conditions.

The RNA-Seq reads are small sections of the transcriptome, hence the traditional way of processing them was to align the reads to a reference genome or transcriptome. From there, the reads were "stitched" together, and the overlap with regions of interest was used for abundance estimation. Abundance can be calculated for exons, transcripts, or genes, though gene-level analysis is generally recommended when transcript information is not required. Transcript quantification is more challenging due to a combination of short sequencing length and lack of unique sequences in many splice variants. Increasing the read length through gradual improvements in the technology will solve this problem in the near future. A prime example of the "traditional" approach is STAR aligner (Dobin *et al.*, 2013), where the genome index created prior to mapping is used during alignment. STAR gained widespread adaptation due to high-performance relative to other methods available at its launch. After the alignment is completed, programs such as featureCounts (Liao *et al.*, 2014) are run to count the mapped reads to genomic features, and provide a summary of the alignment. The postalignment report contains information on the percentage of mapped, unique, and duplicated reads, and serves as a final quality check before downstream analysis. A more detailed overview of the alignment can be generated using RNA-SeQC (DeLuca *et al.*, 2012). A recently developed alternative to this software stack is the so-called "pseudoaligners," which instead of aligning to the reference, determine read compatibility with defined targets (Bray *et al.*, 2016). kallisto is an example of such approach, and the algorithm has not only been shown to be very fast but also accurate and more robust to errors (Bray *et al.*, 2016). The transcript level abundance estimates produced by kallisto can be converted to counts using the R package tximport (Soneson *et al.*, 2015), after which they can be analyzed in the same fashion as any other alignment result. Lastly, and similarly to DNA-chip data, the reads should be normalized so that a comparison can be made between samples from different lanes or sequencing runs. Fragments per kilobase of transcript per million mapped reads (FPKM) is a popular expression unit that normalizes for sequencing depth and corresponding gene length. A similar type of normalization can be achieved in tximport using the lengthScaledTPM parameter, which is the final step in the processing of raw sequencing reads.

Differentially Expressed Genes and Visualization

The downstream analysis of normalized counts varies greatly depending on the experimental design, type of data and sample characteristics. While biological replicates and appropriate controls are crucial in many studies, in settings such as clinical bioinformatics, the lack of controls or a low number of samples is still common. Whenever paired samples are available (e.g., control and treated, or normal and diseased tissue from the same patient), differential expression analysis is usually performed first. It is a quantitative measure of expression change that can be used to rank genes based on a degree of under- or overexpression. It is a good practice to remove lowly expressed genes beforehand, which can be accomplished either using sophisticated software models or more simply by removing genes with a count ≤ 10 .

Fold change, or the magnitude of change between conditions, is a preferred method for quantification of differential expression. It can be calculated using a variety of packages such as limma, DESeq2 (Love *et al.*, 2014), or sleuth (Pimentel *et al.*, 2017), all of which offer technically advanced though slightly different implementations. For a simpler manual calculation, the genes should first be transformed to account for the presence of extreme values and mean-variance dependency. This is most commonly achieved using log2 transformation, though alternatives such as rlog or VST are also used (Lin *et al.*, 2008). Subsequently, the log2(mean) of all replicates from a healthy control/untreated samples are subtracted from the log2(mean) of all replicates from a disease phenotype/treated samples. For datasets with no paired samples, a comparison can be made between groups with and without a certain condition or phenotype.

Visualization is an important component in bioinformatics as it is challenging to identify patterns in tens of thousands of genes based on text data alone. The most common graphics libraries used for high-quality plotting are ggplot2 in R (Ginestet, 2011) and matplotlib in Python 3 (Hunter, 2007). The built-in statistics and graphics capabilities in R are usually sufficient for initial data exploration purposes. Bioinformatics investigation is often initiated with a thorough check on how closely the samples match the experimental design. This is performed to identify population structures in the data that are previously unknown, or subgroups of samples with different characteristics. Furthermore, outliers, or samples with abnormally low or high values (which could represent technical issues), can be identified and removed. Principal component analysis (PCA) helps to find such characteristics by obtaining a combination of variables, in this case, gene expression values, that best differentiate the samples (Abraham and Inouye, 2014; Sharma and Paliwal, 2007). Any aberrant samples should be removed at this stage so that they do not influence the results. Subsequently, two-group expression comparison can be visualized using MA plots, which show the relationship between logged fold change and mean expression. Such plots are equally suited for diagnosing data-related issues as well as checking the number, magnitude, and pattern of differentially expressed genes. Another way of exploring global changes in gene expression is

the Kolmogorov-Smirnov nonparametric test. It compares the distribution of each gene between two defined groups and is a good approach to visualize changes in mean expression and variability. The above-mentioned plots show general patterns of expression and should be supplemented with more direct visualizations. A very popular and information-rich example is heatmaps, which are two-dimensional grids where a range of values is represented by colors. They are often paired with hierarchical clustering, which sorts rows (which represent gene expression) and columns (which represent samples) in a way that most clearly shows patterns in the data. The combination can reveal commonly regulated genes or sets of samples with distinct expression profiles and characteristics (such as disease subtypes). Another way of using heatmaps is correlation matrices, which show a correlation of expression between multiple genes. More advanced plotting methods often contain a higher density of information within a single figure, utilizing colors, shapes, and opacity in a third-dimensional space.

Functional, Pathway, and Network Analysis

While global profiling of gene (or protein) expression provides a significant amount of information, it is often challenging to link such results to tangible biological insights. Employing annotation regarding a gene's (or its product's) involvement in a certain biological process, signaling pathway, or molecular function is one of the most effective approaches to study RNA-Seq data. Gene set enrichment analysis is one of the most widely used solutions, which verifies if the differentially expressed gene results show an enrichment or depletion of genes associated with a certain biological aspect (Subramanian *et al.*, 2005). This can include genes connected with cancer development, drug response, a wide range of signaling pathways, and more (Curtis *et al.*, 2005). There are numerous approaches for performing gene set analysis (e.g., overrepresentation analysis, pathway topology), most of which are implemented in R or are available as online tools (Tarca *et al.*, 2013). Pathway information can be easily retrieved from online databases such as KEGG (Kanehisa *et al.*, 2016), Reactome (Fabregat *et al.*, 2016), or WikiPathways (Slenter *et al.*, 2018), either manually or using the associated application programming interfaces. A different approach makes use of topological properties of genes within molecular interaction networks. Based on the direct interaction partners, or close proximity neighbors, it is possible to infer novel insights such as biological function (Sharan *et al.*, 2007) or prioritize candidate disease genes (Yu *et al.*, 2013). Cytoscape is the most popular environment for network studies (Shannon *et al.*, 2003), though it is also possible to perform a wide variety of network analyses using igraph in R (Csardi and Nepusz, 2006) or networkx in Python 3 (Hagberg *et al.*, 2008). Weighted gene coexpression network analysis is a great example of a network-based exploratory tool that provides additional data reduction and feature selection capabilities (Langfelder and Horvath, 2008). It is based on correlation patterns between gene expression, which forms the basis for finding clusters of potentially associated genes. It also identifies relationships between multiple such modules. Network approaches are also very useful for integrating different types of data, such as multiomics profiles (e.g., mutation, expression, or methylation datasets) or electronic medical records. Similarity network fusion is a popular example – it constructs a network of samples for each data layer and fuses them into a combined structure (Wang *et al.*, 2014). This approach strengthens strong signals (i.e., similarities between layers) and allows the samples to be grouped into distinctive subtypes.

Machine Learning Approaches in Genomics

The unprecedented growth in volume and complexity of biomedical datasets and literature has made bioinformatics analyses more challenging than ever. The very high ratio of features (e.g., mutation or expression) to the number of samples, and the considerable time investment needed to achieve expert-level knowledge on a particular biological phenomenon translate into the additional level of complexity. This has led to growing popularity of ML approaches, algorithms designed to automatically learn from data without an explicit set of rules to follow. While genomic datasets pose many unique challenges that need to be considered, ML algorithms can be extremely useful, particularly when combined with standard bioinformatics tools and databases. There are three main types of algorithms that are of direct use to genomic studies – unsupervised, supervised, and semisupervised. They are all easily accessible through Python, R, or Java libraries, although it requires familiarity with statistical analysis and programming.

Unsupervised learning is geared towards inferring “hidden structures” in unlabeled data – labels being the output values that in biological context can mean disease subtypes, months of patient survival or type of response to a treatment. Given the previously mentioned high number of features, dimensionality reduction is the first area of interest. PCA has traditionally been used to reduce the dimensionality of the data by geometrically projecting them onto lower dimensions (Lever *et al.*, 2017). This transformation allows for noise removal and easier classification, though recently developed t-distributed stochastic neighbor embedding was shown to outperform it in some biological applications (Fonville *et al.*, 2013). Such methods can be combined with pathway annotation to better explain variations in the phenotype (Ma and Kosorok, 2009). One of the primary aims of personalized medicine efforts is finding subgroups of patients with distinct phenotypes, which would allow for a more tailored prognosis, care, and treatment. To this end, a plethora of clustering approaches are routinely used on genomic data, mainly in hopes of finding novel disease subtypes (Van Rooden *et al.*, 2010). Hierarchical clustering, k-means, and spectral clustering are popular algorithms, though many specialized methods have been created specifically for biomedical application (Sharma *et al.*, 2017a,b,c, 2016a).

When label data is available, supervised learning approaches can be used to identify significant features or build models to predict categorical classes (classification) or continuous values (regression). While the identification of biomarkers for

detecting and monitoring diseases has long been a focus in medicine, in many cases more complex models are needed to make an accurate diagnosis (Kidd *et al.*, 2015). Overfitting is a common challenge when building prediction models – given the high number of features in genomic datasets, it is relatively easy to find a combination of genes that will correlate with outcome labels by pure chance. Such results will not generalize well when tested with an independent dataset. It is thus important to separate the data into training and testing datasets (a popular split is 70%–30%), and only use the latter for sporadic validation of the model. Models with fewer features and lower complexity are always preferred, especially if the features are already known to be predictive of the outcome of interest, or are connected to a biologically relevant mechanism or pathway. The baseline prediction performance can be established using simple annotation (patient information, sample characteristics) or using biomarkers already reported in the literature. Such model can be further expanded or replaced entirely depending on its performance and complexity. Subsequently, feature selection approaches (Sharma *et al.*, 2012a,b,c,d, 2014, 2011; Sharma and Paliwal, 2011) can be applied to find biological components that best explain the phenotype of interest. Simple univariate methods such as t-test or Pearson’s correlation were shown to be the optimal methods for feature selection in genomic data (Haury *et al.*, 2011). Alternatively, tree-based algorithms or recursive feature elimination paired with an estimator are popular choices as they allow to rank features based on their importance to the model. Once a subset of features is selected, a model can be constructed using a variety of algorithms – LASSO, ElasticNet, and random forest regressor are popular choices for predicting quantity while support vector machine (SVM), random forest classifier, or k-nearest neighbors (kNN) are popular for predicting categories. More advanced strategies such as boosting, bagging, or stacking can be used to decrease variance or increase accuracy – xgboost, being a representative example, usually provides satisfactory out-of-the-box (i.e., using default parameters) results (Chen and Guestrin, 2016). Neural networks, while popular in many fields such as medical image recognition, are used more sporadically in biomedical applications where decision-making transparency is often crucial. An iterative process of constructing, tuning (i.e., tweaking parameters), and testing the model is usually performed on the training data, using a 10-fold cross-validation (CV) technique. The process partitions the data into 10 equal parts and uses nine parts for training and one for evaluation. This is repeated 10 times, each time using different parts of the data for training and testing, and the final result is calculated as an average from all the runs. The performance metric depends on the type of the label, with accuracy, precision, and recall being popular for classification and root mean squared error and coefficient of determination (R^2) for regression. Once satisfactory results are achieved, the model should be trained on the entire training data (70% in our example) and tested on the remaining test data (the leftover 30%). The performance of a good model should show comparable or better results relative to the 10-fold CV evaluation. While some improvements can be achieved using different ML algorithms or by tweaking the parameters, poor performance usually means that the selected features are not enough to predict the outcome. Feature engineering, or creation of novel features using domain knowledge, is often the best approach to improve the model (Ang *et al.*, 2016). A more recent development in the ML field is semisupervised algorithms, which take advantage of the information contained within unlabeled data in classification problems (Scholkopf and Zien, 2006). Mixing unlabeled and labeled data has been shown to improve the supervised learning accuracy and is of particular interest to biology, where unlabeled data is abundant but labeled data is difficult to obtain.

Workflows in Proteomics

Proteomics is an extensive field of research where a plethora of studies are being conducted. These studies cover the following areas: protein fold recognition, structural class prediction, function analysis, posttranslational modification, and molecular recognition of features (Dehzangi *et al.*, 2018, 2017; López *et al.*, 2018, 2017; Chou, 2017). Various bioinformatics tools and packages have thus been developed for advancing the above research areas. The implementation of modern pipeline frameworks requires detailed guidelines and often offers a command line or workbench interface. In this section, we survey on the pipeline frameworks for identifying molecular recognition features (MoRFs) in IDPs. Specifically, we emphasize the design strategy of MoRF prediction systems and provide practical recommendations based on the requirements of high-throughput bioinformatics analyses.

The Necessity of Molecular Recognition Feature Prediction

In the traditional view of protein structure–function paradigm, proteins fold into a stable three-dimensional structure that ultimately determines their function. However, recent progress in computational and experimental methods has revealed a lack of this stable tertiary structure in certain protein regions (Dyson and Wright, 2005, 2015; Lee *et al.*, 2014; Uversky, 2014). Proteins comprising such regions are called IDPs and reportedly have important biological functions related to signal transduction and cell regulation (Wright and Dyson, 2015; Lee *et al.*, 2014). IDPs often achieve their functions through the loosely structured MoRF region. This region binds to a structured partner and consequently undergoes a disorder-to-order transition to adopt a well-defined conformation (Lee *et al.*, 2014). MoRF length varies up to 70 amino acids. Although there are many documented experimental techniques for identification and analysis of MoRF regions, they are still expensive and time-consuming. To overcome this

challenge, computational approaches have become absolutely necessary for predicting these MoRFs in disordered protein sequences (Sharma *et al.*, 2018a,b, 2016b; Malhis *et al.*, 2016; Disfani *et al.*, 2012; Dosztányi *et al.*, 2009).

State-of-the-Art Molecular Recognition Feature Predictors

An abundance of computational methods and predictors have been already developed to predict MoRFs. These include ANCHOR (Dosztányi *et al.*, 2009), MoRFPred (Disfani *et al.*, 2012), MoRFchibi (Malhis and Gsponer, 2015), MoRFPred-plus (Sharma *et al.*, 2018a), MoRFchibi-web (Malhis *et al.*, 2016), PROMIS, and OPAL (Sharma *et al.*, 2018b). OPAL, the latest proposed predictor, has also proven the most accurate predictor in the literature. The above-mentioned predictors are currently available as online web servers and downloadable software packages. The subsequent sections illustrate the guidelines of MoRFs prediction and how to evaluate the pipeline of MoRF prediction.

Fundamentals of Protein Sequence Analysis

Sequence

The protein sequence is retrieved from an experimental setup where all the amino acids of the protein or peptide are determined. For computational analyses, a large number of protein sequences are at present available in different databanks, which were generated and annotated through high-throughput technologies.

Feature source and feature extraction technique

The features of protein sequences can be captured from different sources of information. These can be chemical, physical and physicochemical properties of amino acids, evolutionary and structural information of protein sequences, functional domains, and gene ontology information of protein sequences.

Some of the existing feature extraction techniques have been used to create feature vectors from protein attributes such as occurrence, composition, transition and distribution, pairwise frequencies, amino acid composition, autocorrelation, and bigram (Sharma *et al.*, 2013, 2015; Liu *et al.*, 2015; Dehzangi *et al.*, 2015; Du *et al.*, 2014). Early studies used to focus on sequential, syntactical, and physicochemical features. However, more recent studies have considered evolutionary-based features because they tend to provide good prediction accuracy for protein-related problems (Sharma *et al.*, 2018a; Yang *et al.*, 2017; Lyons *et al.*, 2016; Dehzangi *et al.*, 2015).

Classifiers

For classification purposes, several classifiers have been widely explored, including SVMs, artificial neural networks, kNN, as well as ensembles of classifiers. Amongst the aforementioned classifiers, SVMs is a well-known classifier that has delivered promising results in recent studies (Sharma *et al.*, 2018b; Malhis and Gsponer, 2015).

Design of a Molecular Recognition Feature Predictor

This section will cover the benchmark dataset and the framework of MoRF prediction.

Benchmark dataset

A wide collection of benchmark databases has been introduced, assembled, and used for developing MoRF predictors. These repositories include MoRFPred, MoRFchibi, MoRFPred-plus, MoRFchibi-web, and OPAL (Table 1).

The sequences in TRAIN and TEST464 sets were obtained from Protein Data Bank and filtered to retain protein-peptide regions of 5 to 25 residues in size (Disfani *et al.*, 2012). The sequences in TEST464 share a less than 30% sequence identity with sequences in TRAIN (Disfani *et al.*, 2012). MoRF predictors are always trained using the sequences in TRAIN set and assessed using the sequences of TEST464 set (Sharma *et al.*, 2018a,b; Malhis and Gsponer, 2015; Malhis *et al.*, 2016). According to the principles used to assemble TRAIN and TEST464 sequences, it is not verified whether identified protein-peptides are disordered in isolation and also TEST464 is not free of redundancy as its sequences share more than 30% sequence identity to each other (Disfani *et al.*, 2012). In order to address the above and validate a MoRF predictor, the EXP53 set is often used. The EXP53 set contains 53 nonredundant protein sequences with MoRFs which have been experimentally verified to be disordered in isolation (Malhis *et al.*, 2016; Sharma *et al.*, 2018b).

Table 1 Description of OPAL benchmark dataset

Datasets	No. of sequences	Total residues	No. of molecular recognition features (MoRF) residues	No. of non-MoRF residues
Training set	TRAIN	421	245,984	5396
Test sets	TEST464	464	296,362	5779
	EXP53	53	25,186	2432
				240,588
				290,583
				22,754

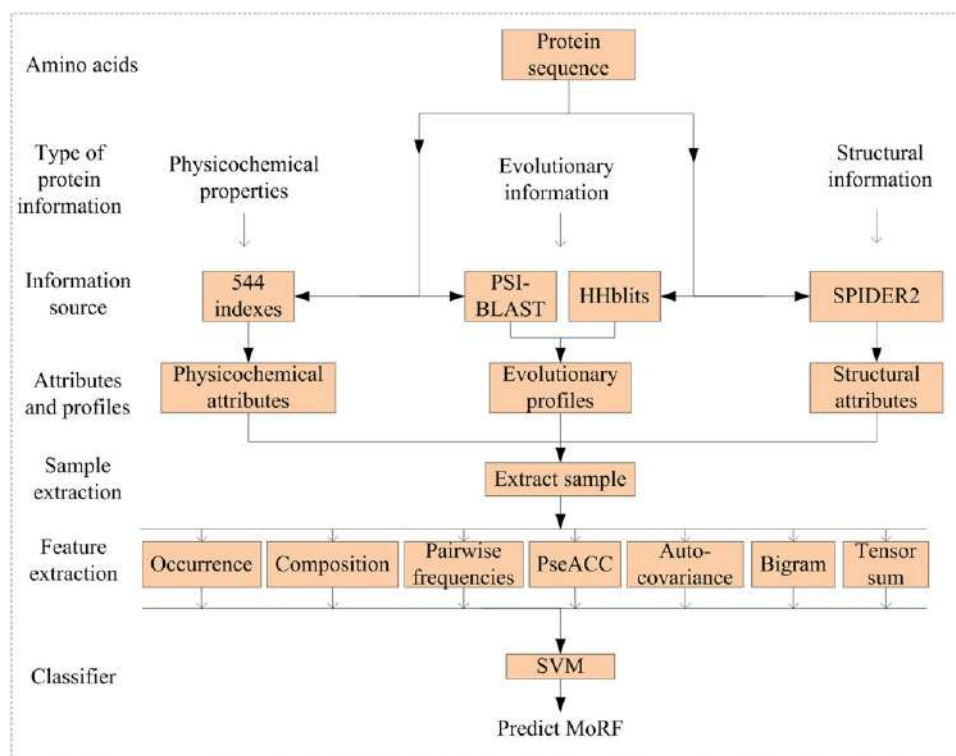


Fig. 4 Framework of molecular recognition feature (MoRF) prediction.

Framework of Molecular Recognition Feature Prediction

Fig. 4 shows the framework of MoRF prediction, whose input is the protein sequence. For prediction purposes, protein sequences are commonly represented using different features. These characteristics comprise physicochemical properties of amino acids such as 544 common physicochemical indexes (Kawashima *et al.*, 2008), evolutionary profiles such as position specific scoring matrix generated with PSI-BLAST (Altschul *et al.*, 1997) and hidden Markov model profiles generated with HHblits (Remmert *et al.*, 2011), as well as structural attributes such as secondary structure, accessible surface area and dihedral torsion angles predicted by tools as SPIDER2 (Yang *et al.*, 2017). To predict each residue in the protein as MoRF and non-MoRF, a sample is first extracted, described as a feature vector, and finally predicted with a suitable classifier.

Fig. 5(A) outlines the bioinformatics tools/scripts and the data format required for implementing a framework of MoRF prediction. These tools are generally run on a Linux operating system and protein sequences are organized in a FASTA format file. Tools such as PSI-BLAST, HHblits, and SPIDER2 can be installed and run via command line. On the other hand, scripts are written with software tools such as MATLAB, OCTAVE, and C-code. Finally, **Fig. 5(B)** shows the procedure of training and scoring a MoRF predictor.

Implementation Steps of a Molecular Recognition Feature Predictor

To identify MoRFs in a protein sequence, each residue should be initially scored. This score will help predict the residues as MoRF or non-MoRF. Two of the methods often used in a MoRF classification scheme are ResidueMoRF and RegionMoRF. These methods are used to extract samples from protein sequences during the training and test steps. The following subsections will discuss the training and test steps, which are of utmost importance to construct a MoRF predictor.

Training step

In this step, features are extracted from MoRFs and non-MoRFs and grouped in a training set. Let us assume the segment F , which represents a MoRF with flanks, and the segment G , which describes a non-MoRF with flanks. Both segments F and G are from a protein sequence in the training set. Let us also suppose that a protein sequence P_i is given as

$$P_i = A_1 A_2 \dots A_j \dots A_{n_i} \quad (i = 1, 2, \dots, T) \quad (1)$$

where A_j is the j th amino acid in the sequence, T is the total number of protein sequences in the training set, and n_i is the length of the protein sequence P_i . For instance, let us consider a protein sequence P_1 whose length $n_1 = 150$ and which contains one MoRF located between A_{30} and A_{40} . The segments F and G for this protein P_1 are illustrated in **Fig. 6**. For illustration purposes, a flank size of 20 amino acids has been considered. Of note, the flank size should be selected during training and evaluation.

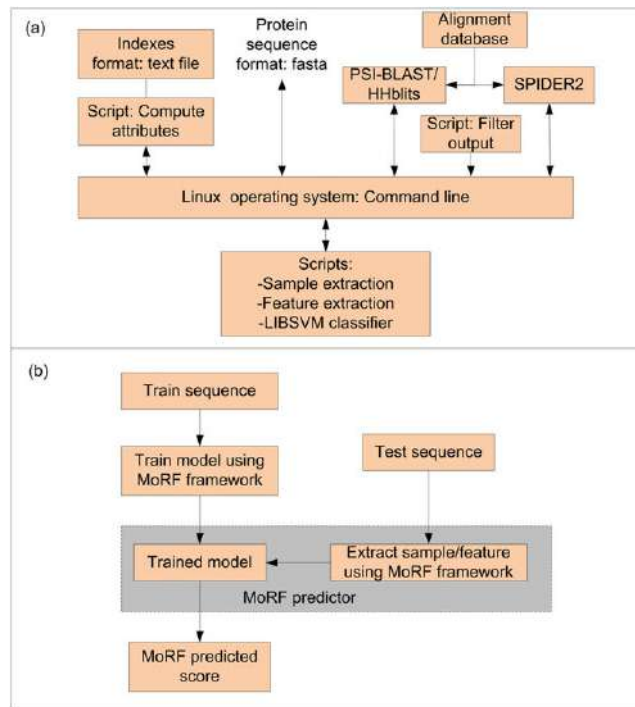


Fig. 5 Pipeline of molecular recognition feature (MoRF) predictor implementation. (A) details of bioinformatics tools/scripts and data formats, (B) procedure of training and testing a MoRF predictor.

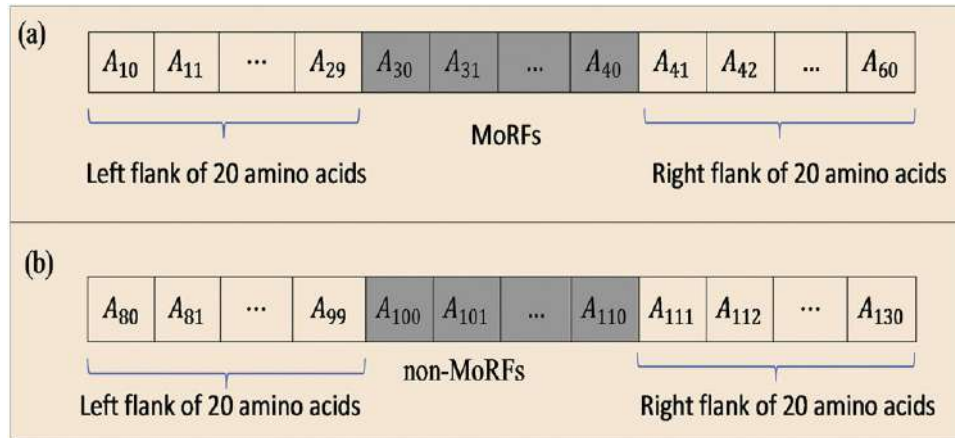


Fig. 6 Schematic illustration of (a) segment F and (b) segment G.

MoRFs can have a variable length of 5 to 25 amino acids. Additionally, zeros are padded to create flanks of 20 amino acids if the MoRF is present at the beginning or end of the protein sequence. Positive and negative samples representing MoRFs and non-MoRFs are extracted from segments F and G, respectively.

For sample extraction, two methods named ResidueMoRF and RegionMoRF, which are detailed below, are commonly used.

ResidueMoRF method

With this method, a window size of 41 ((flank size $\times$ 2) + 1) amino acids is regarded for extracting samples from a segment. For extracting positive samples, the center of the window is placed on the MoRF residue of segment F, with amino acids upstream and downstream of the MoRF residue. For example, the sample S_x for a MoRF residue is defined as

$$S_x = \{A_{x-20}, \dots, A_{x-2}, A_{x-1}, A_x, A_{x+1}, A_{x+2}, \dots, A_{x+20}\} \quad (2)$$

where A_x is the MoRF residue (i.e., in Fig. 7(A), $x=30$). Positive samples extracted from a segment F using Eq. (2) are interpreted as

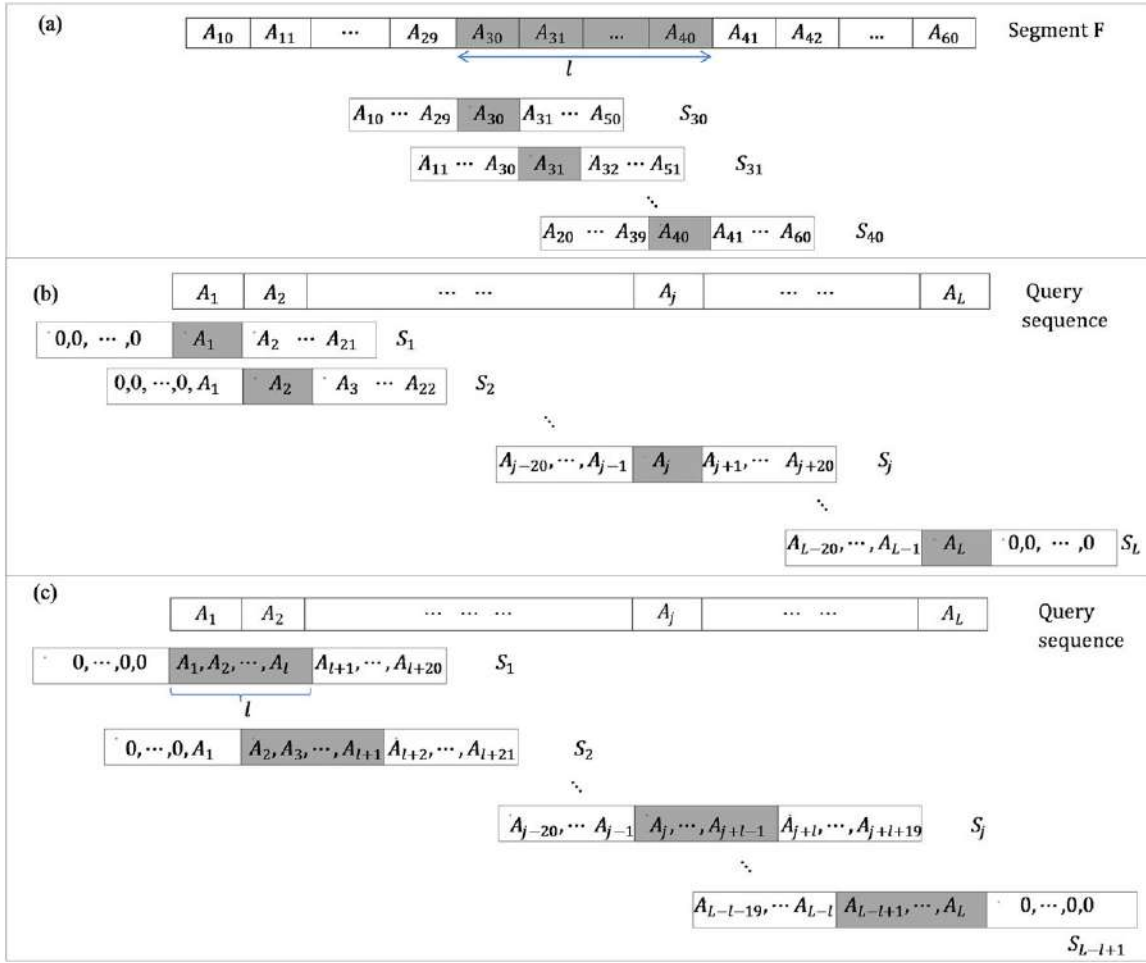


Fig. 7 Graphical illustration of (a) sample extraction from segment F , (b) sample extraction for scoring a query protein sequence of length L (A_j is the j th amino acid in the sequence), and (c) sample extraction from a query sequence of length L .

$$\gamma_{tr} = \begin{cases} S_x \\ S_{x+1} \\ \vdots \\ S_{x+l-1} \end{cases} \text{ for } 5 \leq l \leq 25 \quad (3)$$

where l is the MoRF length (i.e., in Fig. 7(A), $l=11$). Fig. 7(A) shows the graphical illustration of sample extraction from the segment F . In a similar way, negative samples are extracted from segment G .

After applying either of the above models, the feature vector is finally computed from the sample and used for model training.

RegionMoRF method

With this method, the entire segment F with a MoRF region of length l (as depicted in Fig. 7(A)) is taken as a positive sample. Likewise, the entire segment G is taken as a negative sample.

Test step

The scoring of each residue in a query protein sequence also requires the extraction of samples. In this case, the sample extraction procedure makes use of the above methods ResidueMoRF and RegionMoRF as well.

ResidueMoRF method

Like the procedure described in the above section, a window size of 41 ((flank size $\times 2$) + 1) amino acids is used to extract a sample for each query residue. The sample S_j represents a query residue defined as

$$S_j = \{A_{j-20}, \dots, A_{j-2}, A_{j-1}, A_j, A_{j+1}, A_{j+2}, \dots, A_{j+20}\} \quad (4)$$

where A_j is the residue in the query sequence and $j=1,2,\dots,L$. Samples extracted using Eq. (4) for a query sequence of length L are described as

$$\gamma_{ts} = \begin{cases} S_1 \\ S_2 \\ \vdots \\ \vdots \\ \vdots \\ S_L \end{cases} \quad (5)$$

Fig. 7(B) shows the graphical illustration of sample extraction from a query sequence. Like the procedure followed for training, zeros are padded to make the length of flanks equal to 20 if query residues are at the beginning or end of the sequence. The feature vector is then computed from the sample and used for scoring.

RegionMoRF method

With this method, a window of size $(2 \times \text{flank size} + l)$ is used to extract samples from a query sequence. The sample S_j for a query sequence is defined as

$$S_j = \{A_{j-20}, \dots, A_{j-2}, A_{j-1}, A_j, A_{j+1}, A_{j+2}, \dots, A_{j+(l-1)+20}\} \quad (6)$$

where A_j is the residue in the query sequence and $j=1,2,\dots,L-l+1$. Samples extracted using Eq. (6) for a query sequence of length L are interpreted as

$$\gamma_{ts} = \begin{cases} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_{L-l+1} \end{cases} \quad (7)$$

where $l=6,7,\dots,30$. A graphical illustration of sample extraction can be observed in Fig. 7(C). Eq. (8) defines the samples used for scoring each of the residues as

$$A_i \rightarrow \begin{cases} \{S_1, S_2, \dots, S_i\}, 1 \leq i < l \\ \{S_{i-l+1}, S_{i-l+2}, \dots, S_i\}, l \leq i \leq L-l+1 \\ \{S_{i-l+1}, S_{i-l+2}, \dots, S_{L-l+1}\}, i > L-l+1 \end{cases} \quad (8)$$

where $i=1,2,\dots,L$. By varying l from 6 to 30, 450 $(6+7+\dots+29+30)$ samples are obtained for each residue, except for the residues at the beginning and end of the query sequence. The feature vector is computed from the sample and used for scoring. For each residue, the maximum score of the samples defined in Eq. (8) is set as final propensity score.

Summary of Molecular Recognition Feature Prediction

For the prediction of MoRFs in protein sequences, state-of-the-art predictors such as ANCHOR, MoRFPred, MoRchibi, MoRFchibi-web, and OPAL are currently available as web servers. Thus, end users or scientists can easily use a graphical web interface to predict MoRF and non-MoRF residues in a protein sequence. Except for MoRFPred, the other predictors can be further installed and configured locally (in personal computers). The performance of the above-mentioned predictors is shown in Table 2. As it clearly shows, the OPAL predictor outperforms the other approaches, achieving a significant accuracy and confirming its practical use in real scenarios.

To select a target protein for further experimental analysis, the sequences of large databases need to be scored. To do this, the end user can write SQL query scripts to obtain such scores from the MoRF predictor (Fig. 8(A)).

As Table 2 clearly indicates, the prediction accuracies are still limited when it comes to selecting a target protein for further experimental analysis for drug design. However, these limitations can be further improved by considering certain strategies. There is a need for high-performance workbenches, which could be accessible via the web to obtain protein sequence alignments. In the current MoRF web servers, the alignment tools are locally installed and run instead of using available alignment web servers. This is partly because alignment tools require a huge amount of time to provide alignment results. In addition, future MoRF pipelines similar to the one in Fig. 8(B) are expected. In other words, a core database server should store the experimental annotation from research labs whereas the computational annotation should be guaranteed by high-throughput bioinformatics analyses.

Table 2 Performance of state-of-the-art MoRF predictors. The results are reported for the TEST464 and EXP53 datasets using the area under the curve (AUC) as the performance metric

Predictor/method	TEST464	EXP53
ANCHOR	0.605	0.615
MoRFPred	0.675	0.620
MoRFchibi	0.743	0.712
PROMIS	0.790	0.818
MoRFchibi-web	0.805	0.797
OPAL	0.816	0.836

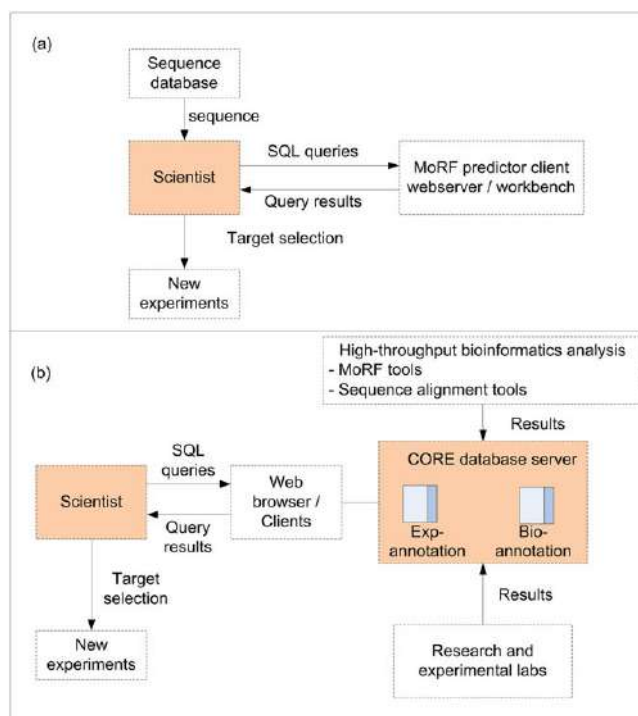


Fig. 8 Requirements for an improvement to molecular recognition feature (MoRF) prediction. (a) procedure of target protein selection for further experimentation, and (b) future pipeline for MoRF analysis.

Bioimage Analysis Pipelines

One of the areas in bioinformatics which is experiencing an enormous transformation is related to bioimage analysis. This is because a great number of bioinformatics applications are relying on image data for assessing different cellular and molecular mechanisms and thus validating the biological hypothesis. More and more images of biological mechanisms are generated by microscopes every single day, making necessary the design of accurate computational approaches.

In this direction, distinct imaging techniques such as PALM (Betzig *et al.*, 2006), STORM (Rust *et al.*, 2006), and STED (Hell, 2003) are currently able to detect individual proteins separated a few nanometers. Moreover, images related to biological mechanisms like RNA interference and chemical compounds (Echeverri and Perrimon, 2006; Moffat *et al.*, 2006; Sepp *et al.*, 2008) are absolutely necessary to shed light on scientific problems such as the differentiation of cancer cell phenotypes (Long *et al.*, 2007a).

However, the complexity of bioimages makes it extremely difficult to apply conventional medical image analysis workflows because the objects of interest sometimes possess different morphologies and intensities. Therefore, it is necessary to start paying attention to this emerging subfield of bioinformatics, which is not widely covered in the scientific literature.

The development of bioimage analysis pipelines often consists of the following steps (Fig. 9): preprocessing in which noise is reduced, image registration (Qu *et al.*, 2015) where images are aligned for pixel-by-pixel comparisons, image segmentation (Meijering, 2012) in which the objects of interest are separated from the background, feature extraction where object descriptors such as shape (Pincus and Theriot, 2007) and texture (Depeursinge *et al.*, 2014) are computed, classification where objects are detected by ML techniques (Shamir *et al.*, 2010), and visualization (Walter *et al.*, 2010).

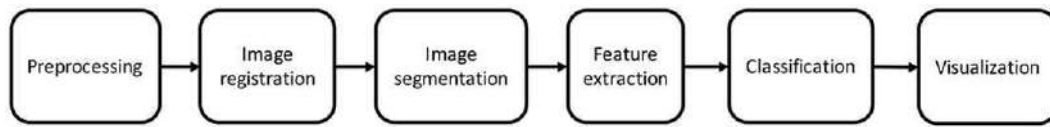


Fig. 9 Bioimage analysis workflow.

In this section, we will go through each of the above steps and explain the different methods.

Preprocessing

Preprocessing should be the first step of any pipeline. Its main objective is to improve the appearance of a bioimage or highlight additional information about the objects of interest, which might not be clearly observable. Many enhancement techniques and filters are often used to enhance the contrast and borders of the different objects, accentuate space frequencies, and remove or attenuate noise. One of these contrast enhancement techniques is histogram adjust, which produces a histogram with a specific form for the output image. Histogram equalization achieves good approximations towards a uniform distribution of the values of gray scale levels, giving the same occurrence probability to all gray levels (Pratt, 2014). To overcome the limitations of standard histogram equalization, the contrast-limited adaptive histogram equalization (CLAHE) technique is sometimes used. It divides the images into contextual regions and applies the histogram equalization to each of them (Pratt, 2014). This evens out the distribution of used gray values, making the hidden features of the image more visible. Another technique is the wavelet transform, which increases the image contrast and facilitates well-localized decomposition schemes (Salvado and Roque, 2005). In addition, digital filters are also used for enhancing bioimages by reducing noise and preserving the borders of those objects of interest. Some of these filters are Prewitt and Canny operators, as well as median, Gaussian, and average filters.

Image Registration

This step is often included in pipelines intended to compare images of different biological conditions such as those for building brain atlas (Ng *et al.*, 2007), and for comparing cell morphology and gene expression patterns. Here, methods such as mutual information registration (Viola and Wells, 1997), spline-based elastic registration (Rohr *et al.*, 2003), and congealing registration (Learned-Miller, 2006) can be utilized. Image registration has been used for blastoderm embryos in *D. Melanogaster* where each nucleus is described using a point in the 3D space (Fowlkes *et al.*, 2008).

Image Segmentation

The main aim of the segmentation step is to separate the objects of interest from the background. It is sometimes considered as a classification process of the objects present in the image because it can easily locate the different objects. Segmentation methods can vary depending on the specific application and image type. However, a segmentation method able to reach acceptable results for all types of bioimages has not been developed thus far. For a better explanation, we will present the segmentation methods in two groups: classical segmentation and clustering methods.

Classical segmentation methods

Thresholding is a method that segments scalar images, creating a binary partition of the image intensities. Once thresholding determines an intensity value, segmentation is achieved by grouping all the pixels based on intensity into two different classes. Its main limitation is that it only considers the intensity instead of other relationships between the pixels. One of the most reported methods is the Otsu's method, which selects the optimal threshold by maximizing the between-class variance with an exhaustive search. Although there are different methods to find a threshold, most of them are not satisfactory when analyzing real images, such as bioimages, due to the presence of noise, plain histograms, or an inadequate illumination.

Another classical method is deformable models, which are based on physical motivations, and used to delineate the borders of regions using curves or closed parametric surfaces deformed under the influence of external and internal forces. Two important deformable models are snakes (Kass *et al.*, 1988) which segments objects whose morphology is variable, and intelligent scissors (Liang *et al.*, 2006) which allows us to segment with minimal user interaction.

Clustering methods

Clustering techniques are also important in biomedical analysis pipelines because they achieve a correct separation of the regions of interest from the image background. Two of the frequently used clustering techniques are fuzzy C-means (Dulyakam and Rangsanseri, 2001; Yang *et al.*, 2004) and hierarchical clustering, including single-linkage, complete-linkage, and average-linkage.

The segmentation of bioimages depends on which features are needed. For instance, texture features can be extracted for chromatin compositions whereas concavity features could be employed for nuclear morphology. When the images are of cell-based assays where nuclear compartments are colored, methods such as globular template segmentation, watershed segmentation, or active contour/snake methods are recommended.

Nevertheless, when the images contain nonglobular objects such as neurons, the neuroanatomical analysis software NeuroLucida (Glaser and Glaser, 1990) is often used. Alternatively, directional kernels are also considered for searching neuronal structures in confocal images (Al-Kofahi *et al.*, 2003). Finally, the tool ImageJ plugin NeuronJ (Meijering *et al.*, 2004) is advisable as well.

Feature extraction/selection

The feature extraction/selection step is of vital importance when we aim to classify objects in bioimages. In this step, representative feature vectors are usually obtained. To do this, dimensionality reduction techniques like PCA are necessary. In addition, in-depth knowledge of the research domain is needed because it will allow us to select those features that can help discriminate between objects of interest. For example, if the aim is to analyze dynamic fluorescent images, texture features should be used (Dorn *et al.*, 2008). For clustering gene expression patterns, global decomposition by eigen-embryo analysis is recommended (Peng *et al.*, 2006) whereas wavelet features able to detect global and local properties are advisable to recognize and annotate gene expression patterns (Zhou and Peng, 2007). When a collection of image characteristics is extracted, we would then recommend the use of minimum-redundant maximum-relevant feature selection algorithms (Ding and Peng, 2005; Peng *et al.*, 2005b) so that redundant features can be eliminated.

Classification

The classification step is also of vital importance in bioimage analysis pipelines because it is here where the features of the object of interest will be used for classification. Standard ML techniques are often employed to achieve this goal. Some of these classifiers are the following:

- Bayesian classifiers, which rely on the Bayes' theorem, can be used in binary classification problems. The naïve Bayes classifier is often employed for classifying objects of interest because it requires a small amount of training data for parameter estimation.
- The kNN rule (Kuncheva, 2014) does not require knowledge of a priori probabilities whereas it assigns an unknown object to the class of its nearest neighbor in the measurement space. Moreover, it does not require patterns to be represented in a suitable vector space but only a similarity measure or distance function.
- Decision trees (Kuncheva, 2014) do not use comparison functions to infer general properties of the training matrix. Nor do they use all the features for object classification but each class can be inferred using its own feature set.
- SVMs (Hastie *et al.*, 2001b) search the hyperplane that separates a set of training objects while maximizing the margin between their classes. This method aims to minimize the bound on the generalization error rather than minimize the mean squared error over the dataset.

When the training set consists of few bioimages, an ensemble of machines is advisable. This strategy builds several basic classifiers and combines their outputs for improving classification. In such cases, a diverse ensemble in which the classifiers have adequately different decision boundaries is needed. To achieve this diversity, strategies such as bagging, boosting, AdaBoost, stacked generalization, and a mixture of experts are often considered (Kuncheva, 2014; Polikar, 2006). Bagging (Kuncheva, 2014; Polikar, 2006) draws different training subsets with replacement and is particularly recommended when the number of images is limited. Boosting (Kuncheva, 2014; Polikar, 2006) makes new machines pay more attention to those examples in which previous machines produced errors. AdaBoost (Kuncheva, 2014; Polikar, 2006) generates a set of hypotheses by training weak machines and making sure that the cases misclassified by previous classifiers are included in the training data of the next classifier.

One of the methods used for clustering mRNA expression patterns of coexpressed genes was the minimum spanning tree cut (Peng *et al.*, 2006) whereas for determining cell identities, the absolute three-dimensional location of the cell along with its location patterns was employed (Long *et al.*, 2008). However, for assigning bioimage patterns to different annotation terms, parallel classifiers are highly recommended (Zhou and Peng, 2007).

Visualization

As in other bioinformatics pipelines, visualization is vital for result verification. Among the techniques often used in bioimage analyses are surface and flow visualization. Tools for the visualization of images of protein surfaces and gene expression can be utilized in addition to immersive visualization systems such as NCMIR's ATLAS and ImmersaDeskTM (Ai *et al.*, 2005).

Conclusions

The spectrum of bioinformatics tools and methods is currently too wide for an exhaustive review. The computational approaches grow in sophistication alongside technological improvements in the corresponding experimental protocols. New requirements, which become more evident with increase in volume and complexity of biological data, will further push the development of

efficient and comprehensive solutions. This article covers four main research areas: genomics, transcriptomics, proteomics, and bioimage analysis.

Within the genomics area, we presented current efforts that focus on accurate mapping of sequences to the reference and identifying regions that show variation. As the number of quantifiable changes will continue to rise, new frameworks geared towards integrating this information and linking them to phenotypic changes will be required.

In the transcriptome section, we examined several computational data processing approaches and covered linking expression profiles to actionable results. We summarized popular tools and libraries that can be easily built upon and extended. As our understanding of pathways, networks and omics interaction grows, so will the scope of the analyses. Future solutions will look at RNA changes more in the context of global interaction network, rather than few genes with higher or lower expression.

For proteomics, we have provided a detailed guideline for MoRF prediction, including data description, examples of state-of-the-art tools, MoRF classification schemes, and pipelines of MoRF downstream analysis. Any reader can easily follow the steps to develop a new MoRF predictor, and thus analyze MoRFs in protein sequences using the presented methodology.

Finally, within the bioimage analysis area, we described the standard workflow for analyzing digital images, which includes preprocessing, image registration and segmentation, feature extraction, classification, and visualization. With the capture of high-resolution images of different cellular mechanisms, future bioinformatics approaches will definitely be necessary to keep up with the technological advances.

Author Contributions

AS and PJK defined the structure of the article. YL, AS and PJK wrote the abstract, introduction, and conclusions. YL and DS covered Section “Genome Analysis” (Genome). PJK covered Section “Transcriptome Analysis” (Transcriptome). RS and AS covered Section “Workflows in Proteomics” (Proteome). YL covered Section “Bioimage Analysis Pipelines” (Bioimages). AS and TT oversaw and managed article creation.

See also: Computing for Bioinformatics. Natural Language Processing Approaches in Bioinformatics

References

- Abeel, T., Parys, T.V., Saeys, Y., Galagan, J., Peer, Y.V.D., 2012. GenomeView: A next-generation genome browser. *Nucleic Acids Research* 40, e12.
- Abraham, G., Inouye, M., 2014. Fast principal component analysis of large-scale genome-wide data. *PLoS ONE* 9, e93766.
- Abyzov, A., Urban, A.E., Snyder, M., Gerstein, M., 2011. CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Research* 21, 974–984.
- Ai, Z., Chen, X., Rasmussen, M., Folberg, R., 2005. Reconstruction and exploration of three-dimensional confocal microscopy data in an immersive virtual environment. *Computerized Medical Imaging and Graphics* 29, 313–318.
- Al-Kofahi, K.A., Can, A., Lasek, S., *et al.*, 2003. Median-based robust algorithms for tracing neurons from noisy confocal microscope images. *IEEE Transactions on Information Technology in Biomedicine* 7, 302–317.
- Altschul, S.F., Madden, T.L., Schaffer, A.A., *et al.*, 1997. Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Research* 17, 3389–3402.
- Ang, J.C., Mirzal, A., Haron, H., Hamed, H.N.A., 2016. Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13, 971–989.
- Babraham-Bioinformatics, 2018. A quality control tool for high throughput sequence data. Available at: <http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc>.
- Bao, H., Guo, H., Wang, J., *et al.*, 2009. MapView: Visualization of short reads alignment on a desktop computer. *Bioinformatics* 25, 1554–1555.
- Behjati, S., Tarpey, P.S., 2013. What is next-generation sequencing? *Archives of Disease in Childhood - Education and Practice* 98, 236–238.
- Betzig, E., Patterson, G.H., Sougrat, R., *et al.*, 2006. Imaging intracellular fluorescent proteins at nanometer resolution. *Science* 313, 1642–1645.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30, 2114–2120.
- Bray, N.L., Pimentel, H., Melsted, P., Pachter, L., 2016. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology* 34, 525–527.
- Brodie, A., Tovia-Brodie, O., Ofra, Y., 2014. Large scale analysis of phenotype-pathway relationships based on GWAS results. *PLoS ONE*. e100887.
- Browning, B.L., Browning, S.R., 2009. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American Journal of Human Genetics* 84, 210–223.
- Buels, R., Yao, E., Diesh, C.M., *et al.*, 2016. JBrowse: A dynamic web platform for genome visualization and analysis. *Genome Biology*, 17, p. 66.
- Carver, T., Harris, S.R., Otto, T.D., *et al.*, 2013. BamView: Visualizing and interpretation of next-generation sequencing read alignments. *Briefings in Bioinformatics* 14, 203–212.
- Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: *KDD'16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California pp. 785–794.
- Cheung, M.-S., Down, T.A., Latorre, I., Ahninger, J., 2011. Systematic bias in high-throughput sequencing data and its correction by BEADS. *Nucleic Acids Research* 39, e103.
- Chou, K.C., 2017. An unprecedented revolution in medicinal chemistry driven by the progress of biological science. *Current Topics in Medicinal Chemistry* 17, 2337–2358.
- Cock, P.J.A., Fields, C.J., Goto, N., Heuer, M.L., Rice, P.M., 2010. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research* 38, 1767–1771.
- Csardi, G., Nepusz, T., 2006. The igraph software package for complex network research. *Inter Journal Complex Systems*. 1695.
- Curtis, R.K., Oresic, M., Vidal-Puig, A., 2005. Pathways to the analysis of microarray data. *Trends in Biotechnology* 23, 429–435.
- Dehzangi, A., Hefterman, R., Sharma, A., *et al.*, 2015. Gram-positive and Gram-negative protein subcellular localization by incorporating evolutionary-based descriptors into Chou's general PseAAC. *Journal of Theoretical Biology* 364, 284–294.

- Dehzangi, A., López, Y., Lal, S.P., *et al.*, 2017. PSSM-Suc: Accurately predicting succinylation using position specific scoring matrix into bigram for feature extraction. *Journal of Theoretical Biology* 425, 97–102.
- Dehzangi, A., López, Y., Lal, S., *et al.*, 2018. Improving succinylation prediction accuracy by incorporating the secondary structure via helix, strand and coil, and evolutionary information from profile bigrams. *PLOS ONE* 13, e0191900.
- DeLuca, D.S., Levin, J.Z., Sivachenko, A., *et al.*, 2012. RNA-SeQC: RNA-seq metrics for quality control and process optimization. *Bioinformatics* 28, 1530–1532.
- Depeursinge, A., Foncubierta-Rodríguez, A., Ville, D.V.D., Müller, H., 2014. Three-dimensional solid texture analysis in biomedical imaging: Review and opportunities. *Medical Image Analysis* 18, 176–196.
- Diaz, A., Nellore, A., Song, J.S., 2012. CHANCE: Comprehensive software for quality control and validation of ChIP-seq data. *Genome Biology*, 13. . p. R98.
- Ding, C., Peng, H., 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology* 3, 185–205.
- Distani, F.M., Hsu, W.L., Mizianty, M.J., *et al.*, 2012. MoRFPred, a computational tool for sequence-based prediction and characterization of short disorder-to-order transitioning binding regions in proteins. *Bioinformatics* 28, i75–i83.
- Dobin, A., Davis, C.A., Schlesinger, F., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29, 15–21.
- Dosztányi, Z., Mészáros, B., Simon, I., 2009. ANCHOR: Web server for predicting protein binding regions in disordered proteins. *Bioinformatics* 25, 2745–2746.
- Dulyakarn, P., Rangsanerai, Y., 2001. Fuzzy C-means clustering using spatial information with application to remote sensing. In: *Proceedings of the 22nd Asian Conference on Remote Sensing*.
- Du, P., Gu, S., Jiao, Y., 2014. PseAAC-General: Fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. *International Journal of Molecular Sciences* 15, 3495–3506.
- Dyson, H.J., Wright, E.P., 2005. Intrinsically unstructured proteins and their functions. *Nature Reviews Molecular Cell Biology* 6, 197–208.
- Eberwine, J., Sul, J.Y., Bartfai, T., Kim, J., 2014. The promise of single-cell sequencing. *Nature Methods* 11, 25–27.
- Echeverri, C.J., Perrimon, N., 2006. High-throughput RNAi screening in cultured cells: A user's guide. *Nature Reviews Genetics* 7, 373–384.
- F.Dorn, J., Danuser, G., Yang, G., 2008. Computational processing and analysis of dynamic fluorescence image data. *Methods in Cell Biology* 85, 497–538.
- Fabregat, A., Sidropoulos, K., Garapati, P., *et al.*, 2016. The reactome pathway knowledgebase. *Nucleic Acids Research* 44, D481–D487.
- Feng, X., Grossman, R., Stein, L., 2011. PeakRanger: A cloud-enabled peak caller for ChIP-seq data. *BMC Bioinformatics* 12, 139.
- Fonville, J.M., Carter, C.L., Pizarro, L., *et al.*, 2013. Hyperspectral visualization of mass spectrometry imaging data. *Analytical Chemistry* 85, 1415–1423.
- Forbes, S.A., Bhamra, G., Bamford, S., *et al.*, 2008. The Catalogue of Somatic Mutations in Cancer (COSMIC). In: HAINES, J.L. (Ed.), *Current Protocols In Human Genetics*.
- Fowlkes, C.C., Hendriks, C.L.L., Keränen, S.V.E., *et al.*, 2008. A quantitative spatiotemporal atlas of gene expression in the *Drosophila* blastoderm. *Cell* 133, 364–374.
- Auton, A., Brooks, L.D., Durbin, R.M., *et al.*, 2015. A global reference for human genetic variation. *Nature* 526, 68–74.
- German, M.A., Pillay, M., Jeong, D.H., *et al.*, 2008. Global identification of microRNA-target RNA pairs by parallel analysis of RNA ends. *Nature Biotechnology* 26, 941–946.
- Ginestet, C., 2011. ggplot2: Elegant graphics for data analysis. *Journal of the Royal Statistical Society Series A* 174, 245.
- Glaser, J.R., Glaser, E.M., 1990. Neuron imaging with neuroLucida – A PC-based system for image combining microscopy. *Computerized Medical Imaging and Graphics* 14, 307–317.
- Hagberg, A., Swart, P.J., Chult, D.S., 2008. Exploring network structure, dynamics, and function using NetworkX. In: *Proceedings of the 7th Python in Science Conference*.
- Hannon, 2010. FASTX-Toolkit: FASTQ/A short-reads pre-processing tools. Available at: http://hannonlab.cshl.edu/fastx_toolkit/.
- Hastie, T., Friedman, J., Tibshirani, R., 2001b. Support vector machines and flexible discriminants. *The Elements of Statistical Learning*. New York: Springer.
- Haury, A.C., Gestraud, P., Vert, J.P., 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS ONE* 6, e28210.
- Hell, S.W., 2003. Toward fluorescence nanoscopy. *Nature Biotechnology* 21, 1347–1355.
- Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 90–95.
- Hurd, P.J., Nelson, C.J., 2009. Advantages of next-generation sequencing versus the microarray in epigenetic research. *Briefings in Functional Genomics & Proteomics* 8, 174–183.
- Hwang, S., Kim, E., Lee, I., Marcotte, E.M., 2015. Systematic comparison of variant calling pipelines using gold standard personal exome variants. *Scientific Reports* 5, 17875.
- Imamura, M., Shigemizu, N., Tsunoda, T., *et al.*, 2013. Assessing the clinical utility of a genetic risk score constructed using 49 susceptibility alleles for type 2 diabetes in a Japanese population. *The Journal of Clinical Endocrinology and Metabolism* 98, E1667–E1673.
- International HapMap Consortium, 2005. A haplotype map of the human genome. *Nature* 437, 1299–1320.
- Jiang, H., Wang, F., Dyer, N.P., Wong, W.H., 2010. CisGenome Browser: A flexible tool for genomic data visualization. *Bioinformatics* 26, 1781–1782.
- Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M., 2016. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research* 44, D457–D462.
- Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. *International Journal of Computer Vision* 1, 321–331.
- Kauffmann, A., 2009. arrayQualityMetrics – A bioconductor package for quality assessment of microarray data. *Bioinformatics* 25, 415–416.
- Kauffmann, A., Huber, W., 2010. Microarray data quality control improves the detection of differentially expressed genes. *Genomics* 95, 138–142.
- Kawashima, S., Pokarowski, P., Pokarowska, M., *et al.*, 2008. AAindex: Amino acid index database, progress report 2008. *Nucleic Acids Research* 36, D202–D205.
- Kharchenko, P.V., Tolstourov, M.Y., Park, P.J., 2008. Design and analysis of ChIP-seq experiments for DNA-binding proteins. *Nature Biotechnology* 26, 1351–1359.
- Kidd, B.A., Readhead, B.P., Eden, C., Parekh, S., Dudley, J.T., 2015. Integrative network modeling approaches to personalized cancer medicine. *Personalized Medicine* 12, 245–257.
- Koboldt, D.C., Zhang, Q., Larson, D.E., *et al.*, 2012. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research* 22, 568–576.
- Kong, L., Wang, J., Zhao, S., *et al.*, 2012. ABrowse – a customizable next-generation genome browser framework. *BMC Bioinformatics* 13, 2.
- Kuncheva, L.I., 2014. *Combining Pattern Classifiers: Methods and Algorithms*. New Jersey: John Wiley & Sons.
- Landt, S.G., Marinov, G.K., Kundaje, A., *et al.*, 2012. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Research* 22, 1813–1831.
- Langfelder, P., Horvath, S., 2008. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* 9, 559.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nature Methods* 9, 357–359.
- Larson, D.E., Harris, C.C., Chen, K., *et al.*, 2012. SomaticSniper: Identification of somatic point mutations in whole genome sequencing data. *Bioinformatics* 28, 311–317.
- Learned-Miller, E., 2006. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 236–250.
- Lee, R.V.D., Buljan, M., Lang, B., *et al.*, 2014. Classification of intrinsically disordered regions and proteins. *Chemical Reviews* 114, 6589–6631.
- Lee, S., Emond, M.J., Bamshad, M.J., *et al.*, 2012. Optimal unified approach for rare-variant association testing with application to small-sample case-control whole-exome sequencing studies. *American Journal of Human Genetics* 91, 224–237.
- Lee, E., Helt, G.A., Reese, J.T., *et al.*, 2013. Web Apollo: A web-based genomic annotation editing platform. *Genome Biology* 14, R93.
- Lever, J., Krzywinski, M., Altman, N., 2017. Points of Significance: Principal component analysis. *Nature Methods* 14, 641–642.
- Liang, K., Keleg, S., 2012. Detecting differential binding of transcription factors with ChIP-seq. *Bioinformatics* 28, 121–122.
- Liang, J., Mcinerney, T., Terzopoulos, D., 2006. United Snakes. *Medical Image Analysis* 10, 215–233.
- Liao, Y., Smyth, G.K., Shi, W., 2014. featureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* 30, 923–930.
- Lin, S.M., Du, P., Huber, W., Kibbe, W.A., 2008. Model-based variance-stabilizing transformation for Illumina microarray data. *Nucleic Acids Research* 36, e11.
- Liu, B., Liu, F., Wang, X., Chen, J., 2015. Pse-in-One: A web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Research* 43, W65–W71.

- Li, Q., Brown, J.B., Huang, H., Bickel, P.J., 2011. Measuring reproducibility of high-throughput experiments. *The Annals of Applied Statistics* 5, 1752–1779.
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25, 1754–1760.
- Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009a. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25, 2078–2079.
- Li, B., Leal, S.M., 2008. Methods for detecting associations with rare variants for common diseases: Application to analysis of sequence data. *American Journal of Human Genetics* 83, 311–321.
- Li, H., Ruan, J., Durbin, R., 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18, 1851–1858.
- Li, R., Yu, C., Li, Y., *et al.*, 2009b. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics* 25, 1966–1967.
- Long, F., Peng, H., Liu, X., Kim, S., Myers, G., 2008. Automatic Recognition of Cells (ARC) for 3D Images of *C. elegans*. In: Vingron, M., Wong, L. (Eds.), *Research in Computational Molecular Biology*. Berlin, Heidelberg: Springer.
- Long, F., Peng, H., Sudar, D., Lelièvre, S.A., Knowles, D.W., 2007a. Phenotype clustering of breast epithelial cells in confocal images based on nuclear protein distribution analysis. *BMC Cell Biology* 8, S3.
- López, Y., Dehzangi, A., Lal, S.P., *et al.*, 2017. SucStruct: Prediction of succinylated lysine residues by using structural properties of amino acids. *Analytical Biochemistry* 527, 24–32.
- López, Y., Sharma, A., Dehzangi, A., *et al.*, 2018. Success: Evolutionary and structural properties of amino acids prove effective for succinylation site prediction. *BMC Genomics* 19, 923.
- Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* 15, 550.
- Lyons, J., Paliwal, K.K., Dehzangi, A., *et al.*, 2016. Protein fold recognition using HMM–HMM alignment and dynamic programming. *Journal of Theoretical Biology* 393, 67–74.
- Machanic, P., Bailey, T.L., 2011. MEME-ChIP: Motif analysis of large DNA datasets. *Bioinformatics* 27, 1696–1697.
- Madsen, B.E., Browning, S.R., 2009. A groupwise association test for rare mutations using a weighted sum statistic. *PLoS Genetics* 5, e1000384.
- Malhis, N., Gsponer, J., 2015. Computational identification of MoRFs in protein sequences. *Bioinformatics* 31, 1738–1744.
- Malhis, N., Jacobson, M., Gsponer, J., 2016. MoRFchibi SYSTEM: Software tools for the identification of MoRFs in protein sequences. *Nucleic Acids Research* 44, W488–W493.
- Marchini, J., Howie, B., Myers, S., Mcvean, G., Donnelly, P., 2007. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics* 39, 906–913.
- Marschall, T., Costa, I.G., Canzar, S., *et al.*, 2012. CLEVER: Clique-enumerating variant finder. *Bioinformatics* 28, 2875–2882.
- Martin, M., 2012. Cutadapt removes adapter sequences from high-throughput sequencing reads. *Bioinformatics in Action* 17, 10–12.
- Ma, S., Kosorok, M.R., 2009. Identification of differential gene pathways with principal component analysis. *Bioinformatics* 25, 882–889.
- Mckenna, A., Hanna, M., Banks, E., *et al.*, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20, 1297–1303.
- Meijering, E., 2012. Cell segmentation: 50 years down the road. *IEEE Signal Processing Magazine* 29, 140–145.
- Meijering, E., Jacob, M., Sarria, J.-C.F., *et al.*, 2004. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A* 58A, 167–176.
- Metzker, M.L., 2010. Sequencing technologies – the next generation. *Nature Reviews Genetics* 11, 31–46.
- Moffat, J., Grueneberg, D.A., Yang, X., *et al.*, 2006. A lentiviral RNAi library for human and mouse genes applied to an arrayed viral high-content screen. *Cell* 124, 1283–1298.
- Morgenthaler, S., Thilly, W.G., 2007. A strategy to discover genes that carry multi-allelic or mono-allelic risk for common diseases: A cohort allelic sums test (CAST). *Mutation Research* 615, 28–56.
- Neumann, B., Walter, T., Hériché, J.-K., *et al.*, 2010. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature* 464, 721–727.
- Newton-Cheh, C., Johnson, T., Gateva, V., *et al.*, 2009. Genome-wide association study identifies eight loci associated with blood pressure. *Nature Genetics* 41, 666–676.
- Ng, L., Pathak, S., Kuan, C., *et al.*, 2007. Neuroinformatics for genome-wide 3-D gene expression mapping in the mouse brain. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 4, 382–393.
- Ozaki, K., Ohnishi, Y., Iida, A., *et al.*, 2002. Functional SNPs in the lymphotoxin-alpha gene that are associated with susceptibility to myocardial infarction. *Nature Genetics* 32, 650–654.
- Pabinger, S., Dander, A., Fischer, M., *et al.*, 2014. A survey of tools for variant analysis of next-generation genome sequencing data. *Briefings in Bioinformatics* 15, 256–278.
- Pedersen, B.S., Layer, R.M., Quinlan, A.R., 2016. Vcfanno: fast, flexible annotation of genetic variants. *Genome Biology* 17, 118.
- Peng, H., Long, F., Ding, C., 2005b. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1226–1238.
- Peng, H., Long, F., Eisen, M.B., Myers, E.W., 2006. Clustering gene expression patterns of fly embryos. In: *Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pp. 1144–1147.
- Pimentel, H., Bray, N.L., Puente, S., Melsted, P., Pachter, L., 2017. Differential analysis of RNA-seq incorporating quantification uncertainty. *Nature Methods* 14, 687–690.
- Pincus, Z., Theriot, J.A., 2007. Comparison of quantitative methods for cell-shape analysis. *Journal of Microscopy* 227, 140–156.
- Podolskiy, D.I., Lobanov, A.V., Kryukov, G.V., Gladyshev, V.N., 2016. Analysis of cancer genomes reveals basic features of human aging and its role in cancer development. *Nature Communications* 7, 12157.
- Polikar, R., 2006. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine* 6, 21–45.
- Pratt, W.K., 2014. *Introduction to Digital Image Processing*. CRC Press Taylor & Francis Group.
- Price, A.L., Kryukov, G.V., De Bakker, P.I., *et al.*, 2010. Pooled association tests for rare variants in exon-resequencing studies. *American Journal of Human Genetics* 86, 832–838.
- Purcell, S., Neale, B., Todd-Brown, K., *et al.*, 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *American Journal of Human Genetics* 81, 559–575.
- Qin, Z.S., Jianjun, Y., Shen, J., *et al.*, 2010. HPeak: An HMM-based algorithm for defining read-enriched regions in ChIP-Seq data. *BMC Bioinformatics* 11, 369.
- Quinlan, A.R., Hall, I.M., 2010. BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics* 26, 841–842.
- Qu, L., Long, F., Peng, H., 2015. 3-D Registration of biological images and models: Registration of microscopic images and its uses in segmentation and annotation. *IEEE Signal Processing Magazine* 32, 70–77.
- Rashid, N.U., Giresi, P.G., Ibrahim, J.G., Sun, W., Lieb, J.D., 2011. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. *Genome Biology* 12, R67.
- Remmert, M., Biegert, A., Hauser, A., Söding, J., 2011. HHblits: Lightning-fast iterative protein sequence searching by HMM–HMM alignment. *Nature Methods* 9, 173–175.
- Ritchie, M.E., Diyagama, D., Neilson, J., *et al.*, 2006. Empirical array quality weights in the analysis of microarray data. *BMC Bioinformatics* 7, 261.
- Ritchie, M.E., Phipson, B., Wu, D., *et al.*, 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* 43.
- Ritchie, M.E., Silver, J., Oshlack, A., *et al.*, 2007. A comparison of background correction methods for two-colour microarrays. *Bioinformatics* 23, 2700–2707.
- Rohr, K., Fornefeld, M., Stiehl, H.S., 2003. Spline-based elastic image registration: Integration of landmark errors and orientation attributes. *Computer Vision and Image Understanding* 90, 153–168.
- Van Rooden, S.M., Heiser, W.J., Kok, J.N., *et al.*, 2010. The identification of Parkinson's disease subtypes using cluster analysis: A systematic review. *Movement Disorders* 25, 969–978.

- Roy, S., Coldren, C., Karunamurthy, A., *et al.*, 2018. Standards and guidelines for validating next-generation sequencing bioinformatics pipelines. *The Journal of Molecular Diagnostics* 20, 4–27.
- Roy, S., Laframboise, W.A., Nikiforov, Y.E., *et al.*, 2016. Next-generation sequencing informatics challenges and strategies for implementation in a clinical environment. *Archives of Pathology & Laboratory Medicine* 140, 958–975.
- Ruffier, M., Kähäri, A., Komorowska, M., *et al.*, 2017. Ensembl core software resources: Storage and programmatic access for DNA sequence and genome annotation. *Database* 2017.bax020.
- Rusk, N., Kiermer, V., 2008. Primer: Sequencing – the next generation. *Nature Methods* 5, 15.
- Rust, M.J., Bates, M., Zhuang, X., 2006. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature Methods* 3, 793–796.
- Salvado, J., Roque, B., 2005. Detection of calcifications in digital mammograms using wavelet analysis and contrast enhancement. In: *Proceedings of the IEEE International Workshop on Intelligent Signal Processing*.
- Sathirapongsasuti, J.F., Lee, H., Horst, B.A.J., *et al.*, 2011. Exome sequencing-based copy-number variation and loss of heterozygosity detection: ExomeCNV. *Bioinformatics* 27, 2648–2654.
- Schölkopf, B., Zien, A., 2006. Introduction to semi-supervised learning. In: Chapelle, O., Schölkopf, B., Zien, A. (Eds.), *Semi-Supervised Learning*. MIT Press.
- Sepp, K.J., Hong, P., Lizarraga, S.B., *et al.*, 2008. Identification of neural outgrowth genes using genome-wide RNAi. *PLoS Genetics* 4, e1000111.
- Shalon, D., Smith, S.J., Brown, P.O., 1996. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research* 6, 639–645.
- Shamir, L., Delaney, J.D., Orlov, N., Eckley, D.M., Goldberg, I.G., 2010. Pattern Recognition software and techniques for biological image analysis. *PLoS Computational Biology* 6, e1000974.
- Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research* 13, 2498–2504.
- Shao, Z., Zhang, Y., Yuan, G.-C., Orkin, S.H., Waxman, D.J., 2012. MANorm: A robust model for quantitative comparison of ChIP-Seq data sets. *Genome Biology* 13, R16.
- Sharan, R., Ulitsky, I., Shamir, R., 2007. Network-based prediction of protein function. *Molecular Systems Biology* 3, 88.
- Sharma, R., Bayarjargal, M., Tsunoda, T., Patil, A., Sharma, A., 2018a. MoRFPred-plus: Computational identification of MoRFs in protein sequences using physicochemical properties and HMM profiles. *Journal of Theoretical Biology* 437, 9–16.
- Sharma, A., Boroevich, K., Shigemizu, D., *et al.*, 2017a. Hierarchical maximum likelihood clustering approach. *IEEE Transactions on Biomedical Engineering* 64, 112–122.
- Sharma, R., Dehzangi, A., Lyons, J., *et al.*, 2015. Predict Gram-positive and Gram-negative subcellular localization via incorporating evolutionary information and physicochemical features into Chou's general PseAAC. *IEEE Transactions on Nanobioscience* 14, 915–926.
- Sharma, A., Imoto, S., Miyano, S., 2012a. A between-class overlapping filter-based method for transcriptome data analysis. *Journal of Bioinformatics and Computational Biology* 10, 1250010.
- Sharma, A., Imoto, S., Miyano, S., 2012b. A filter based feature selection algorithm using null space of covariance matrix for DNA microarray gene expression data. *Current Bioinformatics* 7, 289–294.
- Sharma, A., Imoto, S., Miyano, S., 2012c. A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9, 754–764.
- Sharma, A., Imoto, S., Miyano, S., Sharma, V., 2012d. Null space based feature selection method for gene expression data. *International Journal of Machine Learning and Cybernetics* 3, 269–276.
- Sharma, A., Kamola, P.J., Tsunoda, T., 2017b. 2D-EM clustering approach for high-dimensional data through folding feature vectors. *BMC Bioinformatics* 18, 547.
- Sharma, A., Koh, C.H., Imoto, S., Miyano, S., 2011. Strategy of finding optimal number of features on gene expression data. *Electronics Letters* 47, 480–482.
- Sharma, R., Kumar, S., Tsunoda, T., Patil, A., Sharma, A., 2016b. Predicting MoRFs in protein sequences using HMM profiles. *BMC Bioinformatics* 17, 504.
- Sharma, A., López, Y., Tsunoda, T., 2017c. Divisive hierarchical maximum likelihood clustering. *BMC Bioinformatics* 18, 546.
- Sharma, A., Lyons, J., Dehzangi, A., Paliwal, K.K., 2013. A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. *Journal of Theoretical Biology* 320, 41–46.
- Sharma, A., Paliwal, K.K., 2007. Fast principal component analysis using fixed-point algorithm. *Pattern Recognition Letters* 28, 1151–1155.
- Sharma, A., Paliwal, K.K., 2011. A gene selection algorithm using Bayesian classification approach. *American Journal of Applied Sciences* 9, 127–131.
- Sharma, A., Paliwal, K.K., Imoto, S., Miyano, S., 2014. A feature selection method using improved regularized linear discriminant analysis. *Machine Vision and Applications* 25, 775–786.
- Sharma, R., Raicar, G., Tsunoda, T., Patil, A., Sharma, A., 2018b. OPAL: prediction of MoRF regions in intrinsically disordered protein sequences. *Bioinformatics* 34, 1850–1858.
- Sharma, A., Shigemizu, D., Boroevich, K.A., *et al.*, 2016a. Stepwise iterative maximum likelihood clustering approach. *BMC Bioinformatics* 17, 319.
- Sherry, S.T., Ward, M.H., Kholodov, M., *et al.*, 2001. dbSNP: The NCBI database of genetic variation. *Nucleic Acids Research* 29, 308–311.
- Shigemizu, D., Abe, T., Morizono, T., *et al.*, 2014. The construction of risk prediction models using GWAS data and its application to a type 2 diabetes prospective cohort. *PLoS ONE* 9, e92549.
- Silver, J.D., Ritchie, M.E., Smyth, G.K., 2009. Microarray background correction: Maximum likelihood estimation for the normal-exponential convolution. *Biostatistics* 10, 352–363.
- Slenter, D.N., Kutmon, M., Hanspers, K., *et al.*, 2018. WikiPathways: A multifaceted pathway database bridging metabolomics to other omics research. *Nucleic Acids Research* 46, D661–D667.
- Soneson, C., Love, M.J., Robinson, M.D., 2015. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research* 4, 1521.
- Spyrou, C., Stark, R., Lynch, A., Tavaré, S., 2009. BayesPeak: Bayesian analysis of ChIP-seq data. *BMC Bioinformatics* 10, 299.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* 102, 15545–15550.
- Sun, C., Medvedev, P., 2017. VarMatch: Robust matching of small variant datasets using flexible scoring schemes. *Bioinformatics* 33, 1301–1308.
- Tarca, A.L., Bhatti, G., Romero, R., 2013. A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity. *PLoS ONE* 8, e79217.
- Thomas-Chollier, M., Darbo, E., Herrmann, C., *et al.*, 2012. A complete workflow for the analysis of full-size ChIP-seq (and similar) data sets using peak-motifs. *Nature Protocols* 7, 1551–1568.
- Uversky, V., 2014. Introduction to Intrinsically Disordered Proteins (IDPs). *Chemical Reviews* 114, 6557–6560.
- Vidal, M., Cusick, M.E., Barabasi, A.L., 2011. Interactome networks and human disease. *Cell* 144, 986–998.
- Viola, P., Wells, W., 1997. Alignment by maximization of mutual information. *International Journal of Computer Vision* 24, 137–154.
- Walter, T., Shattuck, D.W., Baldock, R., *et al.*, 2010. Visualization of image data from cells to organisms. *Nature Methods* 7, S26–S41.
- Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38, e164.
- Wang, B., Mezzini, A.M., Demir, F., *et al.*, 2014. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods* 11, 333–337.
- Wright, P.E., Dyson, H.J., 2015. Intrinsically disordered proteins in cellular signalling and regulation. *Nature Reviews Molecular Cell Biology* 16, 18–29.
- Wu, M.C., Lee, S., Cai, T., *et al.*, 2011. Rare-variant association testing for sequencing data with the sequence kernel association test. *American Journal of Human Genetics* 89, 82–93.

- Xu, S., Grullon, S., Ge, K., Peng, W., 2014. Spatial clustering for identification of ChIP-Enriched Regions (SICER) to map regions of histone methylation patterns in embryonic stem cells. In: Kidder B. (Eds.), *Stem Cell Transcriptional Networks*, Humana Press, New York.
- Yang, Y., Chongxun, Z., Lin, P., 2004. A novel fuzzy C-means clustering algorithm for image thresholding. *Measurement Science Review* 4, 11–19.
- Yang, Y., Heffernan, R., Paliwal, K., *et al.*, 2017. SPIDER2: A package to predict secondary structure, accessible surface area and main-chain torsional angles by deep neural networks. *Methods in Molecular Biology* 1484, 55–63.
- Yu, D., Kim, M., Xiao, G., Hwang, T.H., 2013. Review of biological network data and its applications. *Genomics & Informatics* 11, 200–210.
- Zambelli, F., Pesole, G., Pavesi, G., 2013. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Briefings in Bioinformatics* 14, 225–237.
- Zhang, Y., Liu, T., Meyer, C.A., *et al.*, 2008. Model-based analysis of ChIP-Seq (MACS). *Genome Biology* 9, R137.
- Zhou, J., Peng, H., 2007. Automatic recognition and annotation of gene expression patterns of fly embryos. *Bioinformatics* 23, 589–596.
- Zhu, L.J., Gazin, C., Lawson, N.D., *et al.*, 2010. ChIPpeakAnno: a Bioconductor package to annotate ChIP-seq and ChIP-chip data. *BMC Bioinformatics* 11, 237.
- Zhu, Z., Zhang, F., Hu, H., *et al.*, 2016. Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets. *Nature Genetics* 48, 481–487.

Relevant Websites

www.r-project.org
R Project.
www.bioconductor.org
Bioconductor.

Constructing Computational Pipelines

ChuangKee Ong, European Bioinformatics Institute (EMBL-EBI), Hinxton, United Kingdom

Russell S Hamilton, University of Cambridge, Cambridge, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

The processing and analysis of data, such as from next-generation sequencing experiments requires multiple steps, many of which have complex sets of requirements. Running these steps in an ad-hoc manner can be very repetitive, time consuming and prone to human error. These shortcomings can be overcome through the utilization of computational pipeline tools, designed to streamline multiple complex analysis steps by linking the individual components into a single robust, efficient and reproducible workflow.

Dedicated pipelines remove the repetitiveness of the analysis and a single command can run a multiple-step pipeline on thousands of samples, whilst at the same time eliminating human error in the manual specification of the parameters required by the pipeline. As the pipeline runs the expectation is that a log file is generated with the version of the tools used, specified parameters, genome version, warning and error messages. These logs record the provenance of the analysis permitting the exact same analysis to be performed by researchers without access to the same hardware or pipeline tool.

In this article, we outline the features of several pipeline frameworks in terms of their job running efficiency and reproducibility. However, the choice of pipeline tools should also take into account the accompanying information provided by the pipeline tool project, for example, user documentation and resources provided by the user community. The development of pipeline frameworks and the defining of individual pipelines have many of the same requirements as the bioinformatics tools the pipelines call. For example, good documentation including manuals, quick start guides and code commenting make implementing the pipelines more manageable (Karimzadeh and Hoffman, 2017). Consideration should also be made to the reproducibility of the software, support for standard file formats, log file and version tracking (List *et al.*, 2017). The future adoption and support of the pipeline tool/framework will depend on its ease of use and ability for the users to write their own new pipelines.

Four key requirements we consider to be essential for a computational pipeline tool are:

1. Log provenance of analysis (e.g., version, parameters, genome release).
2. Standardized analysis across experiments.
3. Customizable, user created pipelines.
4. Utilize job scheduling where available.

Other important pipeline tool features include job dependencies where jobs await the execution of the steps upstream in the pipeline producing the required input. Non-dependent jobs can be run in parallel. In combination with a job scheduler this allows the individual steps in a pipeline to run as efficiently as possible. Check-pointing is also a very desirable feature, where pipelines can be restarted from the failure point rather than having to execute the entire pipeline from the start (Leipzig, 2016).

The primary take home message from this article is that to achieve robust and reproducible bioinformatics analyses it is vital to make use of a dedicated pipeline tool. Furthermore, the most appropriate tool should be selected based on the available computer infrastructure, user experience and project size.

Definitions of Hardware Levels, Project Sizes and User Experience Levels

One of the first considerations for choosing a pipeline framework to implement is the hardware available to run the pipelines (Fig. 1). This may be at the modest level of a single laptop for small scale project or prototyping right up to large, many thousand central processing unit (CPU) high performance computing (HPC) clusters. Cloud based solutions can scale beyond typical HPC installations if resources are available. Here we define four levels of hardware to aid selecting the most appropriate framework for the hardware available.

- Level 1: Laptop. Multicore (2+), RAM (8 GB+).
- Level 2: Node. Multi-processor (1–2), Multicore (4+), RAM (64 GB+).
- Level 3: HPC. Multi-processor, Multi-nodes (2+, typically 100 s), RAM (64 GB+).
- Level 4: Cloud. Multi-processor, Multi-node (2+) RAM (64 Gb+).

A further consideration is the project size which could range from a small RNA-Seq experiment consisting of six samples (e.g., 3 control and 3 case), up to a large-scale cohort study including many thousands of samples. Unless there are dedicated systems administrators available to install and maintain a pipeline environment the ease of install and usage are important factors in the choice of pipeline framework. User experience can range from a bench biologist wanting to analyze their data in a robust and reproducible manner up to seasoned bioinformaticians running pipelines on a production level scale for example at the EBI working towards the Ensembl releases on a three-month cycle.

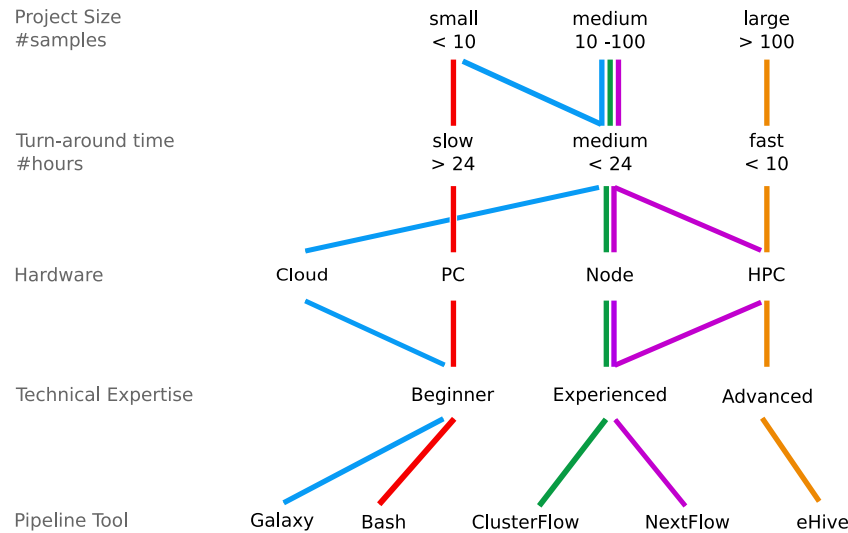


Fig. 1 Decision flowchart for choosing a suitable pipeline tool. There are many routes through the decision chart as each project or situation will have unique and changing requirements. We have highlighted five routes we consider to lead to the most appropriate choice for some typical environments. The red route is the least optimal as no dedicated pipeline framework is utilized.

Pipeline Tools

In this section, we will discuss a selection of popular pipeline tools and highlight their strengths and weaknesses. We will also introduce more generally applicable resources such as virtualisation and cloud based frameworks which are particularly well suited to developing bioinformatics pipelines.

Each pipeline tool uses its own terminology for the elements comprising a fully functional workflow or pipeline. To simplify the terminology, we define some common terms and concepts.

- Module/process: an independent, self-contained task in a pipeline, for example, the adapter and quality trimming of fastq files.
- Pipeline/workflow: constructed by linking together modules/processes to achieve an objective. [Fig. 3](#) depicts a simple RNA-seq data processing pipeline.
- Pipeline tool/framework: an implementation of multiple pipelines or workflows with necessary supporting libraries. A pipeline framework can also include other features such as viewing and creating figures from the analyses.
- Job scheduler: software to control and manage unattended background execution of jobs. A job scheduler will queue jobs until the requested resources (processor and memory) are available. Examples of job scheduling software are Simple Linux Utility for Resource Management (SLURM), Grid Engine (SGE), Portable Batch System (PBS) and Load Sharing Facility (LSF).
- Serial/parallel job processing: a job can run as serial processes, where each instruction is performed in turn; or in parallel where instructions are performed at the same time on different or multi-node processors.

Ad-Hoc/Shell Scripts

At the simplest level, a collection of tools can be combined into a pipeline using shell scripts, a set of commands to be run by the Linux Shell (e.g., BASH). These are often simple to create and require customisation for each project. There are many advanced features that can be incorporated into shell scripts. However, scripts lack the two main features of an efficient data processing pipeline: the ability to launch jobs on the completion of dependent jobs and check-pointing ([Leipzig, 2016](#)). With this in mind, and the difficulty in writing and maintaining scripts can be avoided by using dedicated workflow/pipeline tools.

ClusterFlow

ClusterFlow was developed as an easy to use and install pipelining system for bioinformatics applications ([Ewels et al., 2016](#)), and can be run on any level of hardware, with and without a dedicated job scheduling system. Internally, ClusterFlow uses pre-defined modules to build pipelines which are run as shell scripts tailored to the system hardware. For example, on a laptop the pipeline would run in serial for a set number of CPU cores. On a full cluster the bash script would launch the pipeline steps in parallel with queuing controlled by the job scheduling system. There are currently over 40 modules defined in ClusterFlow, using over 25 bioinformatics tools and there are over thirty predefined pipelines, an example of a simple RNA-Seq workflow is shown in [Fig. 3](#). Writing custom modules involves the tool command and options to be defined in a Perl module, requiring a moderate knowledge

of coding. Once created, modules can be included in pipelines defined as a nested list of tools to be run. The indentation and nesting defines the order the tools must be run and their dependencies. For example, a trimming step must complete before an alignment is started. As a pipeline completes a log file is generated capturing any log output from the tools in the pipeline as well as version information for the tools run. This is harvested at run time so produces the actual tool and genome version used. An email notification system runs when the job has completed, or of any issues that occurred during the run. ClusterFlow is unable to re-queue failed jobs, but does report the errors in log files.

Nextflow

NextFlow is a domain independent pipeline tool with strong support for bioinformatics applications (Di Tommaso *et al.*, 2017) with tasks such as operations on common bioinformatics file formats are included as part of the core distribution. Pre-made bioinformatics pipelines are available from the Nextflow user community and are curated on the NextFlow website. The installation and initial set-up are non-trivial and require an experienced bioinformatician with systems administration skills. However, once set up NextFlow provides a very robust system for running pipelines with check-pointing allowing pipelines to be restarted from any point if failures are encountered. Modules can be written in any language (e.g., Perl, Python, Bash) and can use existing scripts, this has an advantage of not requiring new modules to be specifically written for Nextflow. Modules are then incorporated into pipelines in the NextFlow 'dataflow programming model' build as an operating system and hardware independent Java Virtual Machine. Nextflow is also compatible with Docker, permitting pipelines to be scaled up to a full HPC system or to cloud instances.

Bpipe

Bpipe (Sadedin *et al.*, 2012) is a Java/Groovy based pipeline tool for bioinformatics, designed to take existing command line tools and shell scripts and wrap them into robust pipelines. The syntax has been selected to ease the transition from existing shell script based pipelines into Bpipe. Each task of the pipeline is written as a separate module allowing new pipelines to easily be built from existing tasks. Check-pointing is implemented so jobs can be restarted from a failure, in addition to the generation of log files as the pipelines proceeds. Non-interdependent tasks can be run in parallel and are compatible with job scheduling tools to fully optimise running jobs on different scale hardware. The tasks and pipelines are hardware independent so can be easily shared across different architectures.

Snakemake

Snakemake (Köster and Rahmann, 2012) is a scalable pipeline system based on the Python language with in-built check-pointing. The specification of the number of cores/threads on the hardware and the job scheduling system enables snakemake to scale from single core platforms to large multi-node clusters. For each snakemake run a directed acyclic graph (DAG) is generated with nodes representing the specified jobs to be run. A useful feature is the ability to export the DAG as an image for the pipeline to allow visualisation of the structure of the pipeline (Similar to Fig. 3). In this way, non-dependent jobs can be run in parallel, and with further configuration of the multi-threaded nature of some programs CPU usage can be optimised. The check-pointing system permits pipelines to be restarted, and with an output file check, pipelines can be re-started from the last successfully completed step rather than having to restart the entire pipeline. If a step of the pipeline fails then snakemake performs an automatic cleanup of output files from these incomplete jobs. Users can create and distribute snakemake workflows via code repositories such as GitHub, however to date no central repository for user-contributed pipelines exists.

eHive

eHive (Severin *et al.*, 2010) is a Perl workflow management system, utilising job schedulers such as LSF to submit jobs to the compute clusters. Large compute clusters work on the concept of a centralised job queue and a job controller. Bottlenecks arise in the job controller as cluster nodes need explicit instructions for each and every job execution. A few seconds latency between job submission and execution can translate into a significant overhead when the scheduling system needs to handle millions of jobs. The core concept of *eHive* is that jobs are not scheduled by central controller, instead are modelled on a honeybee queen in a hive with her worker bees. The queen will create workers using a job creation/selection algorithm by observing the central blackboard. It is designed with the aim of moving job scheduling away from a centralised controller yet retaining the ability to monitor jobs as they progress. Eventually a system emerges with very small (3 ms) job overhead, fault tolerance and optimised utilization of computing clusters with more than 1000 cores.

The main console for *eHive*, the *beekeeper*, creates *workers* as required and submits them to the compute cluster, reporting their progress. In this model, workers need not be aware of the existence of other workers, rather only the steps prior and downstream of itself in the pipeline. Each worker generates and stores partial solutions to problems communicated indirectly through the central blackboard. Workers are autonomous and have the ability to create additional jobs, analyses, dataflow and control rules as they run. A MySQL database is used as the central blackboard detailing the jobs and processes to be run in the pipeline. In a centralised pipeline tool, a job will retrieve input data from a shared resource and in some cases store results into a common database. This

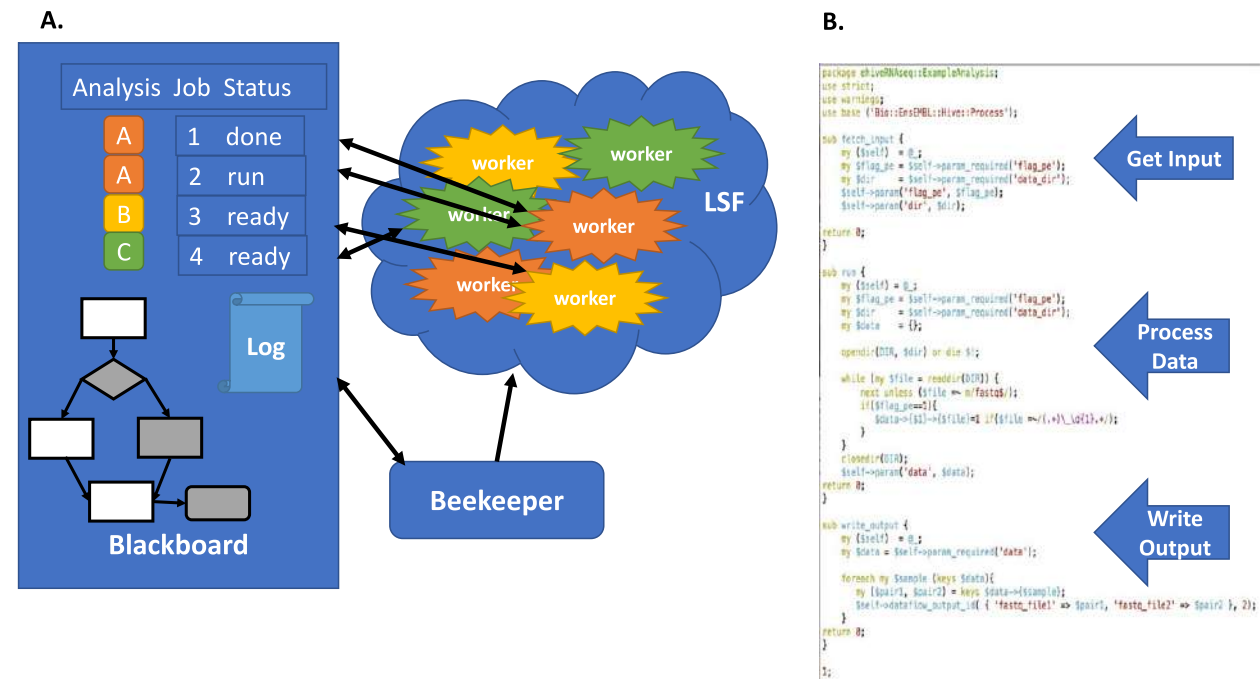


Fig. 2 Architecture of *eHive* Pipeline Tool. (A) Different components of *eHive*, the Blackboard (MySQL database) containing job list and the data flow rules; *beekeeper*, console that create workers and submit jobs to compute clusters; Workers, which run in the LSF job scheduling system. (B) A typical *eHive* Analysis module. Three key stages in a typical *eHive* analysis module lifecycle.

has the potential to create bottlenecks if there are a large number of jobs running at once. In *eHive*, the database can anticipate significant numbers of concurrent write requests via the implementation of an InnoDB database engine to enable row-level-locking supporting concurrent table updates. Jobs can also be processed in batches to mitigate this issue. Once a worker is created, it will register its choice of analysis on the central blackboard and load the modules needed to execute the analysis. Subsequent workers will grab and execute jobs available from the central blackboard. As the jobs progress, workers log their process status on the blackboard, finally setting the job status to done once completed without errors (Fig. 2).

Through these optimisations *eHive* can easily handle tens of millions of jobs and has been successfully tested with SGE, PBS and the default LSF job scheduler. *eHive* has been used in several large scale international bioinformatics projects as a key infrastructure of the data coordination centre, for example, the Ensembl (Aken *et al.*, 2017) and Ensembl Genomes Project (Kersey *et al.*, 2016), the Functional Annotation of Animal Genomes project (FAANG) (Andersson *et al.*, 2015) and the International Genome Sample Resource (ISGR) (Clarke *et al.*, 2017), as well various other high throughput genomics and epigenomics projects. To illustrate *eHive*'s industry scale capacity; the *eHive* InterProScan pipeline (Jones *et al.*, 2014) is run every three months in Ensembl and Ensembl Genomes, where protein domain annotations are run for over 1000 genomes, against 11 million protein sequences. *eHive* chunks the sequences into smaller sets and runs the InterProScan protein domain identification. The resulting XML files of identified signatures are then parsed and stored in the respective genome databases. The pipeline usually runs for 7–10 days, requiring regular monitoring via *guiHive* (Cunningham *et al.*, 2015) or a MySQL interface to ensure the beekeeper is still alive, and creating beekeeper jobs are typically due to memory limits or missing files, but can be easily restarted with the check-pointing system.

Virtualization

Rather than run pipelines directly on the available hardware via a dedicated pipelining system such as those outlined above, virtualisation provides a layer of abstraction with benefits such as being agnostic to the local hardware and operating system (OS). In essence, virtualisation is the ability to run multiple virtual computers on a single computer (e.g., laptop). Containers are stripped down virtual machines just containing the required parts of the operating system to run the required programs. The main advantage of virtualisation is the portability and the popularity of tools such as Docker in the wider computing community have created a large user base with extensive online resources.

Docker

Docker containers are a method of packaging together the minimal required elements required to run a piece of software or pipeline such as the source code, runtime environment, system tools, and system libraries. This makes an efficient, lightweight,

self-contained systems ensuring software will always run reproducibly, regardless of where it is deployed. Containers and Virtual Machines have a shared heritage so are similar as in having resource isolation and allocation benefits, however architectural differences have given containers the advantage of being both portable and efficient. Virtual Machines typically comprise the main application, binaries and libraries as well as the entire guest OS and can be tens of GBs, running independently of the host OS. Contrary to this, containers have the same capability of a Virtual Machine, excluding the full OS in favour of sharing a kernel with other containers, running as isolated processes in user space on the host operating system. A further advantage of containers is their ability to isolate applications from one another as well as the underlying hardware, providing an additional layer of protection for the application a.k.a secure by default.

Docker is utilized across many disciplines, not just bioinformatics, however there are several projects providing domain specific resources for bioinformatics pipelines. These range from tutorials and documentation to repositories of community created pipelines and containers for bioinformatics software (see Relevant Websites Section). Bio-Docklets ([Kim et al., 2017](#)) are an effort to ease the creation of bioinformatics pipelines using Docker. Each tool in the pipeline is containerised in an individual *docklet*, with wrapper scripts combining the docklets into a pipeline.

Pipeline Cloud Environments

In a next generation sequencing project with a high number of samples and/or read depth is likely to require the availability of high performance computing hardware. If this is not available, or a rapid scale-up is required, then cloud based environments are an attractive alternative to local hardware. However, it is worth stating that as with all cloud based storage, the data may be located in a different country potentially violating the policies of data storage for many government organisations and charities. A key advantage of many of the available pipeline cloud environments is a graphical interface to run and create pipelines, rather than relying on command-line execution or coding.

Galaxy

Galaxy ([Afgan et al., 2016](#)) is targeted at researchers who do not want to design and run pipelines via the command line as pipelines can be designed graphically via the Galaxy interface accessed via an Internet browser. There is a large user community providing implementations of individual bioinformatics tools as well as complete pipelines. The Galaxy team also present regular user and developer meetings ensuring a thriving community supporting the platform. Galaxy can be accessed via public servers, local installs or via cloud instances.

Seven Bridges and DNANexus

There are several commercial cloud-based bioinformatics environments, with Seven Bridges (see Relevant Websites Section) and DNANexus (see Relevant Websites Section) being the most widely known. Both provide a large range of pipelines for the most common types of NGS analysis. However, a completely novel/custom pipeline would require working closely with the commercial teams. An open source alternative, Arvados (see Relevant Websites Section) can be installed and run on private cloud instances, or via a commercial partner Curoverse (see Relevant Websites Section). The main advantage of cloud solutions is the ability to instantly scale up analysis without having to worry about the underlying hardware, the cost of cloud deployments is usually on a cost-effective pay-per-usage model. The commercial solutions also allow users to interact via web-based graphical interfaces or the command line for more advanced users.

Pipeline Languages

Each of the pipeline tools introduced in this article require the pipelines to be defined and coded in a tool specific manner. While some effort has gone into making pipelines portable, e.g., the ability to import shell scripts into NextFlow, it is a very attractive prospect to define pipelines in a truly environment agnostic manner. To address this the common workflow language was, and continues to be developed.

Common Workflow Language

The Common Workflow Language (CWL) aims to standardise the language used to define pipelines and allow the sharing of pipelines across the different pipeline systems ([Amstutz et al., 2016](#)). The vision is for a researcher with a pipeline developed in for example Clusterflow to be able to share the code with a researcher using Arvados without needing to re-implement the pipeline from scratch. This inherent portability from using CWL is also advantageous if a pipeline on local hardware requires a rapid scale up to run on a cloud environment for a specific project. CWL is supported in NextFlow, Galaxy, Seven Bridges, DNANexus and Arvados.

Worked Example

To highlight the differences of the implementations of workflow in a selection of pipeline tools we provide a worked example for a simple RNA-Seq pipeline as a bash script (Fig. 3), in *ClusterFlow* and in *eHive*. We provide example code and documentation for each pipeline tool on Github (see Relevant Websites Section). In our simple paired-end RNA-Seq example there are six main steps, with requirements for jobs to complete before the next can proceed (i.e., dependencies). The initial quality control is performed with FastQC and trim_galore. The alignment with HiSat2 is dependent on trim_galore completing. Post-alignment gene quantification using htseq-counts and quality assessment using Qualimap require the alignment step to complete. Finally, once all jobs in the pipeline are complete for a sample then a summary report is generated with MultiQC. In the bash script example, each step in the pipeline is executed sequentially as the job dependencies are not defined. Each step can run as multithreaded jobs, but the overall efficiency of the pipeline is far from optimal. In the *ClusterFlow* and *eHive* examples the jobs are executed in a dependency aware manner, allowing efficient deployment of the pipeline jobs across samples. *eHive*'s check-pointing system will automatically restart failed jobs, and with the further optimisations detailed above is particularly suited to very large scale projects.

With the recent development and rapid adoption of single cell RNA sequencing (scRNA-Seq) it is worth highlighting the differences to the worked example above in terms of the creation of a reproducible and robust pipeline. There are several scRNA-Seq tool suites available, for example: Seurat, Scater and Monocle (Zappia *et al.*, 2017), however these could benefit from tighter integration with high performance computing environments and the robustness of pipeline tools for reproducible analysis. As the number of single cells in a typical experiment can easily be over 100,000 depending on the sequencing protocol, particular attention must be given to the quality control steps including filtering, normalisation, and drop-out modelling so are a critical component of any single cell pipeline (Svensson *et al.*, 2017).

Discussion

When it comes to pipeline design, there are various tools available and considerations to be taken into account. Here we illustrate a selection of tools and outline some key metrics (Table 1), which we feel that can aid in the process of decision making. Depending on the importance of a certain metric(s) to the project, one could then decide on the most appropriate tool to use. For instance, due to a short delivery time-frame for a small size project, a tool that is easy to install, new modules and pipeline can be easily built are of paramount importance. With this in mind, one could then look at *ClusterFlow* as potential tool solution as it satisfies the requirements for logging and the use of a job scheduler. At the other end of the spectrum, when analysis reproducibility, tool robustness and re-queuing capabilities are essential, *eHive* or *Galaxy* are strong candidates. It is absolutely critical from the outset to have a clear understanding of the project characteristics, available resources and output delivery in order to make the optimal decision on the pipeline tool to use. The use of pipeline frameworks in designing computational pipelines will ease the management of multiple pipelines, reducing the need for manual intervention, therefore minimizing errors and increasing the robustness and reproducibility of

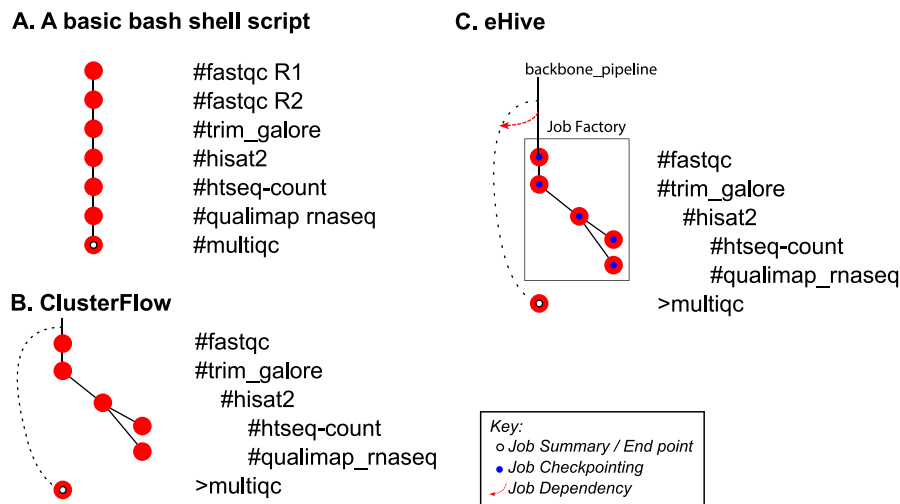


Fig. 3 A schematic highlighting the different job architectures for three pipeline tools. (A) Simple bash shell script with each of the pipeline steps executed in series requiring each job to be completed before launching the next. (B) *ClusterFlow* implements job dependencies, where jobs wait for their dependent job to be completed before launching. Non-dependent jobs can launch and run in parallel. A final step can be specified, in this case a MultiQC report, to run once all other jobs have completed. (C) *eHive* also has job dependencies, but with additional error checking and checkpointing. In this way pipeline jobs can be restarted from the last completed step. In all cases multithreading can be implemented in the specification of the tools being run, allowing within job parallelisation. Example code for each of the pipelines (A,B,C) can be freely accessed. Available at: <https://github.com/darogan/ConstructingComputationalPipelines/>.

Table 1 Comparison table for pipeline tool features

<i>Metric</i>	<i>BASH</i>	<i>ClusterFlow</i>	<i>NextFlow</i>	<i>eHIVE</i>	<i>Galaxy</i>
Ease of install	Easy	Easy	Moderate	Advanced	Moderate
Pre-made modules	No	Yes	Yes	Yes	Yes
Ease of writing modules	Easy	Moderate	Moderate	Advanced	Easy
Ease of writing pipelines	Easy	Easy	Moderate	Advanced	Easy
Use of queueing system	Partial	Yes	Yes	Yes	Yes
Checkpoints/re-queueing	No	No	Yes	Yes	Yes
Tool version logging	No	Yes	Yes	Yes	Yes
Pre-defined genome locations	No	Yes	Yes	Yes	Yes
Hardware level compatibility	1,2,3	1,2,3	1,2,3	2,3	2,3,4
Cloud compatibility	No	No	Yes	Yes	Yes
Inbuilt Docker support	No	No	Yes	No	Yes
Large scale projects	No	No	Yes	yes	Yes
Common Workflow language compatible	No	No	No	No	Yes
Under active development	N/A	Yes	Yes	Yes	Yes

Five options for running pipelines are compared for features supported by the tool as default. In many cases the feature compatibility is possible with customisation and an advanced level of expertise. Hardware levels are defined in the Introduction (1. Laptop, 2. Node, 3. HPC, 4. Cloud).

the analyses. The RNA-Seq worked example illustrated here is a basic implementation to demonstrate how a pipeline is constructed within three different frameworks. For examples of more comprehensive analysis pipelines, we recommend the GATK best practices for variant calling on whole genome and exome high-throughput sequencing projects (Auweru *et al.*, 2013). The variant calling pipeline includes many quality control steps such as marking duplicates, base quality recalibrations and filtering for variants. GATK also produce their own workflow data language (WDL) for constructing pipelines.

Future Directions

One of the key issues that must be addressed to ensure the widespread adoption of robust pipeline frameworks is the lack of interoperability of defined pipelines (e.g., RNA-Seq) between the different pipeline frameworks. Currently, it is not trivial to share a defined pipeline between frameworks, however with the standardization of languages for defining pipelines (e.g., CWL or WDL) and platform independent frameworks for running software (e.g., Biocontainers) there is a clear route to achieve this in the near future (Leprevost *et al.*, 2017).

Closing Remarks

The primary aim of this article has been to guide users into selecting the most suitable pipeline tool for their project and available resources. With all the advantages of automating robust bioinformatics pipelines it should be stated that this should not be at the expense of understanding the underlying processing and analyses being performed. Bioinformatics should never be thought of as a black-box, and careful thought must be applied to ensuring the most appropriate tools are being used.

Acknowledgements

Russell Hamilton is funded by the Centre for Trophoblast Research (University of Cambridge). ChuangKee Ong is funded by Open Targets, European Bioinformatics Institute (EMBL-EBI). We would also like to thank Joseph Gardner and Malwina Prater for the critical reading of this article.

Appendix

Code and documentation to support our implementation of the worked example in Bash, *Clusterflow* and *eHive* are freely available from GitHub (see Relevant Websites Section).

See also: Computational Pipelines and Workflows in Bioinformatics. Computing for Bioinformatics. MapReduce in Computational Biology via Hadoop and Spark. Natural Language Processing Approaches in Bioinformatics

References

- Afgan, E., Baker, D., van den Beek, M., *et al.*, 2016. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.* 44, W3–W10. doi:10.1093/nar/gkw343.
- Aken, B.L., Achuthan, P., Akanni, W., *et al.*, 2017. Ensembl 2017. *Nucleic Acids Res.* 45, D635–D642. doi:10.1093/nar/gkw1104.
- Amstutz, P., Crusoe, M.R., Tijanic, N., *et al.*, 2016. Common Workflow Language. v1.0. doi:10.6084/M9.FIGSHARE.3115156.V2.
- Andersson, L., Archibald, A.L., Bottema, C.D., *et al.*, 2015. Coordinated international action to accelerate genome-to-phenome with FAANG, the Functional Annotation of Animal Genomes project. *Genome Biol.* 16, 57. doi:10.1186/s13059-015-0622-4.
- Auwerda, G.A., Van Der, Carneiro, M.O., Hartl, C., *et al.*, 2013. From FastQ data to high-confidence variant calls: The genome analysis toolkit best practices pipeline. *Curr. Protoc. Bioinform.* 43. doi:10.1002/0471250953.bi1110s43.
- Clarke, L., Fairley, S., Zheng-Bradley, X., *et al.*, 2017. The international Genome sample resource (IGSR): A worldwide collection of genome variation incorporating the 1000 Genomes Project data. *Nucleic Acids Res.* 45, D854–D859. doi:10.1093/nar/gkw829.
- Cunningham, F., Amode, M.R., Barrell, D., *et al.*, 2015. Ensembl 2015. *Nucleic Acids Res.* 43, D662–D669. doi:10.1093/nar/gku1010.
- Di Tommaso, P., Chatzou, M., Floden, E.W., *et al.*, 2017. Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* 35, 316–319. doi:10.1038/nbt.3820.
- Ewels, P., Krueger, F., Käller, M., *et al.*, 2016. Cluster flow: A user-friendly bioinformatics workflow tool. *F1000Research* 5, 2824. doi:10.12688/f1000research.10335.1.
- Jones, P., Binns, D., Chang, H.Y., *et al.*, 2014. InterProScan 5: Genome-scale protein function classification. *Bioinformatics* 30, 1236–1240. doi:10.1093/bioinformatics/btu031.
- Karimzadeh, M., Hoffman, M.M., 2017. Top considerations for creating bioinformatics software documentation. *Brief. Bioinform.* bbw134. doi:10.1093/bib/bbw134.
- Kersey, P.J., Allen, J.E., Armean, I., *et al.*, 2016. Ensembl Genomes 2016: More genomes, more complexity. *Nucleic Acids Res.* 44, D574–D580. doi:10.1093/nar/gkv1209.
- Kim, B., Ali, T., Lijeron, C., Afgan, E., Krampis, K., 2017. Bio-Docklets: Virtualization containers for single-step execution of NGS pipelines. *bioRxiv*. 0–8.
- Köster, J., Rahmann, S., 2012. Snakemake – A scalable bioinformatics workflow engine. *Bioinformatics* 28, 2520–2522. doi:10.1093/bioinformatics/bts480.
- Leipzig, J., 2016. A review of bioinformatic pipeline frameworks. *Brief. Bioinform.* bbw020. doi:10.1093/bib/bbw020.
- Leprevost, F., da, V., Grüning, B.A., *et al.*, 2017. BioContainers: An open-source and community-driven framework for software standardization. *Bioinformatics* 33, 1–3. doi:10.1093/bioinformatics/btx192.
- List, M., Ebert, P., Albrecht, F., 2017. Ten simple rules for developing usable software in computational biology. *PLOS Comput. Biol.* 13, e1005265. doi:10.1371/journal.pcbi.1005265.
- Sadedin, S.P., Pope, B., Oshlack, A., 2012. Bpipe: A tool for running and managing bioinformatics pipelines. *Bioinformatics* 28, 1525–1526. doi:10.1093/bioinformatics/bts167.
- Severin, J., Beal, K., Vilella, A.J., *et al.*, 2010. eHive: An artificial intelligence workflow system for genomic analysis. *BMC Bioinform.* 11, 240. doi:10.1186/1471-2105-11-240.
- Svensson, V., Vento-Tormo, R., Teichmann, S.A., 2017. Exponential scaling of single-cell RNA-seq in the last decade. *arXiv:1704.01379*.
- Zappia, L., Phipson, B., Oshlack, A., 2017. Exploring the single-cell RNA-seq analysis landscape with the scRNA- tools database. *bioRxiv*. doi:10.1101/206573.

Further Reading

- Beaulieu-Jones, B.K., Greene, C.S., 2017. Reproducibility of computational workflows is automated using continuous analysis. *Nat. Biotechnol.* 35, 342–346. doi:10.1038/nbt.3780.
- Ewels, P., Magnusson, M., Lundin, S., Käller, M., 2016. MultiQC: Summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* 32, btw354. doi:10.1093/bioinformatics/btw354.
- Hothorn, T., Leisch, F., 2011. Case studies in reproducibility. *Brief. Bioinform.* 12, 288–300. doi:10.1093/bib/bbq084.
- Kim, Y., Poline, J., Dumas, G., 2017. Experimenting with reproducibility in bioinformatics. *bioRxiv*. 0–5. doi:10.1101/143503.
- Merelli, I., Pérez-Sánchez, H., Gesing, S., D'Agostino, D., 2014. Latest advances in distributed, parallel, and graphic processing unit accelerated approaches to computational biology. *Concurr. Comput. Pract. Exp.* 26, 1699–1704. doi:10.1002/cpe.3111.
- Pawlik, A., van Gelder, C.W.G., Nenadic, A., *et al.*, 2017. Developing a strategy for computational lab skills training through Software and Data Carpentry: Experiences from the ELIXIR Pilot action. *F1000Research* 6, 1040. doi:10.12688/f1000research.11718.1.
- Perez-riverol, Y., Wang, R., Sachsenberg, *et al.*, Ten simple rules for taking advantage of GitHub. *PLOS Comput. Biol.* 5–11. Available at: <https://doi.org/10.1371/journal.pcbi.1004947>.
- Piccolo, S.R., Frampton, M.B., 2016. Tools and techniques for computational reproducibility. *Gigascience* 5, 30. doi:10.1186/s13742-016-0135-4.
- Sandve, G.K., Nekrutenko, A., Taylor, J., Hovig, E., 2013. Ten simple rules for reproducible computational research. *PLOS Comput. Biol.* 9, 1–4. doi:10.1371/journal.pcbi.1003285.
- Wilson, G., Aruliah, D.A., Brown, C.T., *et al.*, 2014. Best practices for scientific computing. *PLOS Biol.* 12. doi:10.1371/journal.pbio.1001745.

Relevant Websites

- <https://arvados.org/>
Arvados.
- <https://github.com/BioContainers/containers>
BioContainers.
- <https://github.com/helios/bio-docker>
BioDocker.
- <https://curoverse.com/>
Curoverse.
- <https://github.com/sjackman/docker-bio>
DockerBio.
- <https://www.dnanexus.com>
DNANexus.
- <https://github.com/darogan/ConstructingComputationalPipelines>
Github.
- <https://github.com/pditommaso/awesome-pipeline>
Pditommaso/awesome-pipeline.
- <https://www.sevenbridges.com/>
Seven Bridges.

Biographical Sketch

ChuangKee is a proven research and informatics leader in the life science and pharma drug discovery space with hands on computational and informatics experience. In his role as Data Coordinator/Manager at OpenTargets (collaboration between EBI, GSK, Sanger Institute & Biogen), He manages data providers across various different groups, and drives data integration efforts. ChuangKee joined OpenTargets from Ensembl, one of the most successful large scale bioinformatics projects in history. Most recently he was the Senior Technical Officer in the Ensembl's production team, responsible for large scale processing pipelines, production infrastructure development, data coordination among sub-teams to ensure delivery of releases consisting of thousands genomes. Before Ensembl, ChuangKee was the Principal Scientist, VP at Sime Darby Technology Centre (SDTC). He headed the bioinformatics, data analysis department whose efforts were geared toward enhancing palm oil yield using high throughput assays such as NGS. Prior to joining SDTC, ChuangKee was a Senior Associate Scientist at Eli Lilly. He lead various cross-functional, -discipline drug discovery informatics projects to support target identification and validation in various therapeutic areas. Before Eli Lilly he was with Medical Research Council in Edinburgh. ChuangKee has a Masters in Bioinformatics from Chalmers University of Technology in Sweden.

Russell Hamilton heads the bioinformatics core facility at the Centre for Trophoblast Research (CTR), University of Cambridge. The focus of the CTR is the study of the placenta and maternal-fetal interactions during pregnancy and brings together over 25 Principal Investigators based in different departments within the University of Cambridge and Babraham Institute. He has previously worked in industry at Cambridge Epigenetix (CEGX) developing epigenetics tools to enable single base resolution sequencing of methyl-cytosine (mC) and hydroxymethyl-cytosine (hmC). Prior to this was a senior postdoc in the Department of Biochemistry at University of Oxford, working on the RNA structural motifs required for mRNA localization and their interaction with trans-acting protein factors. Russell received an MSc in Intelligent Systems from the University of Aberdeen and his PhD in bioinformatics from the University of Edinburgh. At each of his previous positions, Russell has set up bioinformatics infrastructure including pipelines for next-generation sequencing, RNA structure searches, protein structural modelling and a microscope imaging facility.

Pipeline of High Throughput Sequencing

ChuangKee Ong, European Bioinformatics Institute (EMBL-EBI), Hinxton, United Kingdom

Qi Bin Kwong, Ai Ling Ong, and Huey Ying Heng, Sime Darby Plantation R&D Centre, Selangor, Malaysia

Wai Yee Low, Davies Research Centre, University of Adelaide, Roseworthy, SA, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Advancements in sequencing technologies in recent years have created massive amount of sequenced data, which necessitate the development of efficient computational pipeline to mine the data. Preprocessing of raw next generation sequencing (NGS) data of large genomes such as that of humans is compute intensive. The bioinformatics bottleneck can be more time consuming than the time it takes to sequence the DNA and hence, the application of the right pipeline with features such as ease of setup and use, parallelization, job scheduling and tracking of compute processes is pivotal to move the field forward. To demonstrate and give readers a glimpse on the pros and cons of applying a pipeline to NGS data, this article will take as an example of five human exomes and preprocess them for single nucleotide variant (SNV) discovery. It starts with showing the painstaking process involved in running some common preprocessing software without stringing them together as a pipeline. Then, the application of pipeline for preprocessing in three different ways are discussed: Shell scripting, Galaxy ([Afgan et al., 2016](#)) and eHive ([Severin et al., 2010](#)). Among these three different methods, eHive can be considered as a frontier pipeline or workflow management system that has been deployed for use in big projects such as The Functional Annotation of Animal Genomes (FAANG) ([Andersson et al., 2015](#)). The dataset and software used for this study are given in the Appendix.

NGS Preprocessing

One of the common tasks in resequencing studies is to align raw sequencing data (e.g., provided in FASTQ format) to a genome and then perform variant calling. This process is so common that the literature is replete with examples of methodology that involves preprocessing and there are many different tools that can be used to achieve this purpose. To provide as a sample case study, consider a scenario where researchers have sequenced a human exome using the Illumina technology; what then would be the steps required to go from raw reads to variant calls?

To start the preprocessing, a quality control check on the raw sequenced data is mandatory. A software tool known as FastQC ([Andrews, 2010](#)) can be used for this purpose. A sample command line format is as below:

```
$ fastqc <PE1> <PE2>
```

where <PE1> and <PE2> represents the front paired end and back paired end FASTQ files, respectively. Key aspects such as per base sequence quality, per base sequence content and Kmer content are given as HTML reports for users to evaluate sequence quality ([Andrews, 2015](#)). The presence of adapters is unwanted because only the insert sequences need to be analyzed. Adapters can be removed using Trim Galore! software ([Krueger, 2015](#)).

```
$ trim_galore -paired <PE1> <PE2>
```

The default output from Trim Galore! will be <PE1>_val_1.fq and <PE2>_val_2.fq. For this command with default parameters, TrimGalore! will auto-detect some common adapters such as Illumina universal, Nextera transposase and Illumina small RNA. Users may provide their own specific adapter sequences too. Additionally, by default sequences with quality score below Q20 will also be removed.

Prior to mapping, the reference genome <REF> needs to be indexed for quick access of its sequences, which is done using the command below.

```
$ bwa index <REF>
$ samtools faidx <REF>
```

The mapping of raw FASTQ reads to the reference genome can be done using BWA aligner ([Li, 2013](#)). BWA mem is used in this case because it is more suitable, faster and more accurate for sequences longer than 70 bp. Users can explore other options given in BWA and a myriad of other published aligners but such details are beyond the scope of this article.

```
$ bwa mem <REF> <PE1>_val_1.fq <PE2>_val_2.fq > mapped.sam
```


This is followed by the variant calling step where mapped.sam is the input. SNPs are called using samtools (Li *et al.*, 2009) and bcftools (Narasimhan *et al.*, 2016). The samtools view command converts the alignment file from SAM to BAM format, which represents a more compact way to store the data. Next, the samtools sort command will arrange the mapped alignment according to coordinate in ascending order along the chromosomes in the reference genome. The combination of samtools mpileup and bcftools view is to gather SNPs and visualize in vcf format.

```
$ samtools view -bT <REF> mapped.sam -o mapped.bam
$ samtools sort mapped.bam mapped.sort
$ samtools mpileup -uf <REF> mapped.sort.bam > output.mpileup
$ bcftools view -vcg output.mpileup > result.vcf
```

The vcf output contains all the SNPs found in the sample. However, it is not meaningful if it cannot be translated into biological significance. ANNOVAR (Wang *et al.*, 2010) is a collection of Perl scripts designed for functional annotation of the SNPs. The script converts the vcf format into input recognizable by ANNOVAR. Next, annotate_variation.pl performs annotations on the functions of the SNPs.

```
$ convert2annovar.pl -format vcf4 -includeinfo result.vcf > input.annovar
$ annotate_variation.pl -buildver hg19 input.annovar annovar/humandb -outfile result
```

Upon running the final command, two files will be generated, which are “result.variant_function” and the “result.exonic_variant_function”. The first file contains all the potential gene-based functional annotation of the detected SNPs, and the second file gives additional information regarding the exonic SNPs detected.

Readers should take note that the steps outlined above illustrate some of the important steps for preprocessing but the gold standard version has additional steps such as base quality recalibration. For those interested in a tutorial that details these additional steps, refer in Low and Tammi (2017).

Application of a Pipeline for NGS Preprocessing Automation

Case Study With Shell Scripting

If one would like to perform large-scale analysis of multiple resequenced data, such as human exomes, instead of running commands one by one, users can write a Shell script (*.sh) that encompasses all the above commands. Shell script is a text file that serves as an overall controller for the pipeline of interest. Upon running the script, the written code will be interpreted and the pipeline will then be executed. Some of the usual Shell-script/command-line interpreters are the Bourne shell (sh), C Shell (csh) and the Bourne Again Shell (bash). In the example in Box 1 below, the interpreter used is Bourne shell, as shown by the #!/bin/sh line on top.

This Shell script basically runs the pipeline for the preprocessing steps detailed in the preceding section. The difference is that a script is now being used to chain together a series of commands to process FASTQ files to get variant calls.

In Shell scripting, it is common to add an echo command, which prints out what the next step is, before running a new process. This allows for tracking of analysis progress or errors, and is particularly useful when a long Shell script is used. The input files for the Shell script can be passed from the command line through the usage of arguments/parameters. In this case, the script takes in two arguments, the FASTQ file and the reference genome file. The first argument is passed into the individual commands in the script as \$1, whereas the second argument is passed on as \$2. To ensure that two arguments are always passed into the script, the “if” statement starting at the second line of the script explains the usage, and that the script will automatically quits if there are lesser than two input arguments.

The main advantage of having a Shell script is that it reduces retyping of commands every time similar analysis needs to be carried out, and it can be set to run automatically at scheduled interval using cron (Keller, 1999). As compared to other workflow programs, setting up an analysis pipeline using Shell script has far lesser dependencies as compared to other workflow programs. Additionally, it is easier to customize. However, the disadvantage of this method is that there is no Graphical User Interface (GUI) and some programming skills are required. Furthermore, it is sometimes difficult to troubleshoot errors but this can be helped by writing code that can catch potential errors or trace for errors from log files but the script is not easy to interpret for novices.

Case Study With Galaxy

In the previous section, automation of various preprocessing steps using a Shell script is shown for one human exome dataset. In a real study, it is likely that multiple exome datasets will be used and as such some sort of looping is required to run the same analysis using different FASTQ input and saving the corresponding output. Although a Shell script can still be used for running multiple samples, the complexity of its use may have grown beyond what most can comfortably handle. In this section, a dedicated platform known as Galaxy will be used to demonstrate the same analysis and extending it to include multiple human exome samples.

Galaxy is an open, web-based platform for bioinformatics analysis that is implemented in Python programming language, which can be deployed on any UNIX system, from local computers to clusters to computer clouds. The web-based GUI of Galaxy

Box 1 Sample shell scripting for running bioinformatics analysis workflow.

```
#!/bin/sh
if [[ $# -lt 3 ]] ; then
    echo "USAGE: sh script.sh fastq_PE1 fastq_PE2 reference.fa"
    exit 1
fi
#This runs FastQC to check on quality of the FASTQ input file
echo "Running FastQC"
fastqc $1 $2
#Indexing the reference genome prior to mapping and SNP calling
echo "Indexing the reference genome"
bwa index $3
samtools faidx $3
#Set output prefix
prefix=`expr match "$2" '\(.*\)_'`
#Running trim_galore filter
echo "Running sequence trimming"
trim_galore --paired $1 $2
#Mapping of the raw reads to the reference genome was done using BWA, followed by
SNP calling with samtools
echo "Running mapping and SNP calling"
bwa mem $3 $prefix\_1\_val\_1.fq $prefix\_2\_val\_2.fq | samtools view -bT $3 - | samtools
sort - tmp1.sort
samtools mpileup -uf $3 tmp1.sort.bam | bcftools view -vcg - > $prefix.vcf
#Functional annotation of the detected SNPs
echo "Running ANNOVAR"
annovar/convert2annovar.pl --format vcf4 --includeinfo $prefix.vcf > $prefix.annovar
annovar/annotate_variation.pl --buildver hg19 $prefix.annovar annovar/humandb -outfile
$prefix
echo "Analysis completed"
```

uses simple HTML markup that works with most web browsers. It can be accessed via the main public web server (see section Relevant Websites), or other publicly accessible Galaxy servers listed (see section Relevant Websites). It also can be installed locally by following the tutorial: see Relevant Websites section. Galaxy development team has created a portal named Galaxy Community Hub (see section Relevant Websites) that is a forum for discussion and allows researchers to share their analysis workflows.

One of the great features of Galaxy is the point-and-click interface that allows users without programming skills to manipulate data interactively. In addition, Galaxy supports for data uploads from a local computer and also import from external sources (Fig. 1). There are more than 650 tools available in the Galaxy server that enable users to perform a wide range of analyses on their data (Afgan *et al.*, 2016). The Galaxy Tool Shed (Blankenberg *et al.*, 2014) provides a platform where tool developers can share their tool configuration files and documentation (Fig. 2).

Users can create a workflow to string together the flow of output as input in other processes, either by using existing history or from scratch. This is done by using the Galaxy's graphical workflow editor. However, it is recommended that users should plan their analyses in advance if they would like to create the workflow from scratch. Fig. 3 shows the workflow for preprocessing human exome datasets created in Galaxy workflow editor. When the workflow is properly setup, users can run the analysis using point-and-click on the interface. In this workflow, once the exome dataset is uploaded, Galaxy will run FastQC and Trim Galore! concurrently to check the quality and trim the sequences. Next, the output from trimming step will be automatically used as input for mapping to reference genome using bwa, and then call for SNP variants.

Galaxy has the feature of capturing the information of each analysis step, including the datasets, tools and parameters used. Its user interface makes it simple to monitor the status of each job. If a job has failed, the error details will be provided to help in

The screenshot displays the Galaxy web interface. On the left is a 'Tools' panel with categories like 'Get Data', 'Send Data', 'Collection Operations', 'Text Manipulation', 'Filter and Sort', 'Join, Subtract and Group', 'Convert Formats', 'Extract Features', 'Fetch Sequences', 'Fetch Alignments', 'Statistics', 'Graph/Display Data', 'NGS: Mapping', 'NGS: SAMtools', 'NGS: Variant Analysis', and 'NGS: QC and manipulation'. The center panel shows a dataset viewer for a FASTQ file, with a warning that the dataset is large and only the first megabyte is shown. The right panel shows a 'History' panel with a list of jobs, including 'subset10K.SRR5306316.2.fastq'.

Fig. 1 A screenshot of Galaxy interface that shows the uploaded human exome datasets from a local computer. The left panel is a tool panel that contains the list of tools available in Galaxy. The center panel is a detail panel, showing the details of tools and data. The right panel is a history panel, showing the executed jobs and data.

Valid Repositories

fastqc search repository name, description				
Name	Synopsis	Type	Installable Revisions	Owner
fastqc	Read QC reports using FastQC	Unrestricted	11 (2017-04-20)	devteam
fastqc_workflow	FastQC workflow designed for use with Refinery Platform	Unrestricted	2 (2016-04-28)	stemcellcommons
package_fastqc_0_10_1	Contains a tool dependency definition that downloads and compiles version 0.10.1 of FastQC.	Tool dependency definition	0 (2014-01-27)	devteam
package_fastqc_0_11_2	fastqc v 0.11.2	Tool dependency definition	0 (2014-11-11)	devteam
package_fastqc_0_11_4	FastQC v 0.11.4	Tool dependency definition	2 (2016-01-19)	iuc

Fig. 2 Galaxy Tool Shed allows the administrator to easily install a tool and its dependencies.

troubleshooting (Fig. 4). The feature of automatically tracked records allows for reproducible research (Sandve *et al.*, 2013), where other users can reproduce the same analysis.

Case Study With eHive

In situations where a large number of NGS datasets need to be processed in a short amount of time, for example, the 1000 Genomes project, a pipeline that is more suitable for heavy tasks is needed. One example is *eHive*, which is a workflow management system designed in Perl. It comes with features such as fault tolerant and distributed processing, which uses schedulers such as Load Sharing Facility (LSF) to submit jobs to the compute clusters (Severin *et al.*, 2010). *eHive* works on the principle that central controller does not perform job scheduling. Rather a queen creates autonomous “worker”, which then create jobs by monitoring the central blackboard.

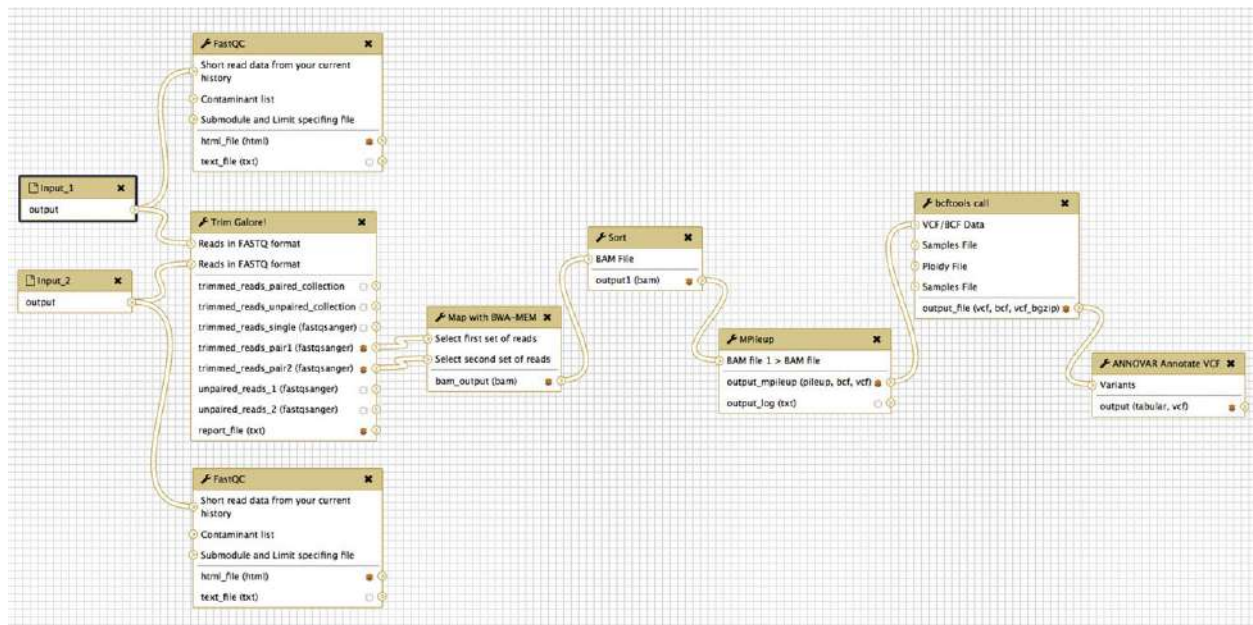


Fig. 3 Exome dataset preprocessing workflow in Galaxy.

Dataset generation errors

Dataset 3: FastQC on data 1: Webpage

Tool execution generated the following error message:

```
Fatal error: Exit code 1 ()
/bin/sh: fastqc: command not found
Traceback (most recent call last):
  File "/usr/share/conda/conda-envs/galaxy-dev/bin/fastqc.py", line 165, in <module>
    fastqc_runner.run_fastqc()
  File "/usr/share/conda/conda-envs/galaxy-dev/bin/fastqc.py", line 137, in run_fastqc
    subprocess.check_call(cmd, shell=True)
  File "/usr/lib/python2.7/subprocess.py", line 540, in check_call
    raise CalledProcessError(retcode, cmd)
subprocess.CalledProcessError: Command 'fastqc --outdir /galaxy/database/job_directory/000/493/dataset_650_files --quiet --extract -f fastqc subseq10K.SRR866988.1' returned non-zero exit status 1
```

Report this error to the local Galaxy administrators

Usually the local Galaxy administrators regularly review errors that occur on the server. However, if you would like to provide additional information (such as what you were trying to do when the error occurred) and a contact e-mail address, we will be better able to investigate your problem and get back to you.

Error Report

Your email
galaxyworkflow@gmail.com

Message

Report

Fig. 4 A screenshot of Galaxy that shows an example of error messages due to a failed run. The status of jobs is marked by coloured boxes for a quick visualization of the overall run: Grey=in queue, Yellow=currently running, Green=completed, and Red=failed.

As a result, the system “behaves” like an insect hive with workers working on jobs. The advantage of the system is that it has very small (~ 3 ms) job submission overhead. The worker is central to the *eHive* model, once they are created by the *eHive* console, they are registered with the central blackboard. The blackboard will have information about various pipeline analysis, statistics of workers

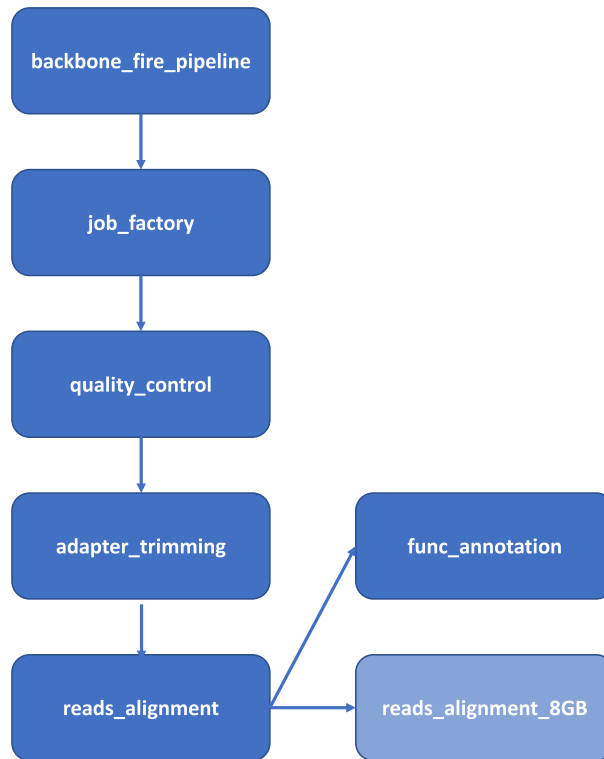


Fig. 5 A snapshot of an eHive workflow for processing of five human exomes.

registered to different analysis and jobs available for processing. Subsequently, the worker will load the code for a chosen analysis, register it on central blackboard and finally run jobs. Once completed, a worker can create new jobs and post it back to the blackboard for other workers. As workers progress through the processing modules within a job, the processing status is logged on the blackboard. Once a job is completed without errors, the worker will set the job status to “DONE”. The primary strength of the blackboard model is that it provides more flexibility as no central controller is required. Though new workers are created centrally by *beekeeper*, it does not perform task distribution and has no visibility about the structure of the pipeline thus order of which the tasks should be solved. Workers have the liberty to alter structure of the pipeline at run-time and as they are fault tolerant, jobs can be re-executed up to a configurable number in the event of a failure before exiting the pipeline. Moreover, it allows automatic flow of jobs to cluster nodes with higher memory, if there is such a requirement.

Fig. 5 shows data flow of an *eHive* pipeline for the same five chosen human exome datasets mentioned in earlier sections. The “job_factory” is the analysis module, which will discover the raw sequencing data to be processed and create jobs with corresponding parameters (e.g., filenames). The “quality_control” analysis module will run the FASTQC steps for all the jobs in parallel, which upon completion of each will create new jobs for the “adapter_trimming” analysis module. Jobs of “adapter_trimming” analysis module, will start immediately when their corresponding “quality_control” jobs is completed. In other words, the dependency is at the job level instead of the analysis module level, which speeds up the whole processing pipeline. The “reads_alignment” analysis module will perform short reads alignment using bwa and should the computing node runs out of memory due to large dataset, the job will be “passed_on” to the next analysis module doing the same function but executed on a computing node with a larger memory, which in this case is 8 GB. The “func_annotation” module executes the ANNOVAR scripts similar to the above Shell scripting section. By default, each job will have a maximum retry attempt of 4 (configurable), until which the job status will be set to “FAILED.” If a significant number of jobs are failing, the console will be terminated and users can investigate the failed jobs looking at the log messages to fix and resume the jobs as needed. The pipeline can be resumed from the state where the jobs failed using intermediate results if any, and hence removes the need to restart a pipeline from scratch, which is a great time saver especially when failure happens at later stage of the pipeline.

Discussion

The true list of potential ways to run a pipeline is not limited to the three methods outlined. A myriad of other workflow management systems exist and some of them are proprietary software that is beyond the scope here. This article focuses more on the usefulness of a computational pipeline to preprocess raw NGS data that nowadays comes in millions of reads. Whether users prefer to use the

Table 1 Comparison of different methods to automate a pipeline

Feature	Platform		
	Shell Scripting	Galaxy	eHive
Graphical User Interface (GUI) feature	No	Yes	Yes
Ease of setup	Easy, with pre-installed software or tools	Medium, with the advantage of publicly available tools and full integration	Medium
Error logging	Yes but indirect	Yes	Yes
Customization	Flexible	Need more time to incorporate customized scripts into Galaxy	Yes
Suitability for simple tasks	Yes	Yes	Not recommended
Scalable	Yes. Meant for high throughput and large scale analysis	Yes. Assumption is the users manage to set up the run on a cluster or cloud computing infrastructure	Yes. It has been used on projects with large datasets such as FAANG

ubiquitously applied Shell scripting method or resort to a workflow management system, such as Galaxy or eHive, depends on various factors that include complexity of the tasks, users' programming skills, ease of setup, errors troubleshooting and customization. **Table 1** details a comparison of the three pipeline methods and features that users will typically need to consider before choosing any pipeline.

The main reason to use a computational pipeline is to avoid time wastage. In a real biological experiment setting that involves NGS data, such as the human exome example given above, a computational pipeline will likely be designed or utilized by bioinformaticians who can understand instructions from biologists on what analyses need to be carried out. However, bioinformaticians are usually not the system administrator and as such, the pipeline designed is shared with those who understand and manage computer infrastructure to determine efficient ways to run the pipeline. In some projects, such as those that involve proprietary software, a part of the pipeline is essentially a 'black box' to the users and critical time bottleneck may not necessarily be in the pipeline itself but rather in the communication on how the data was processed. Sometimes even data transfer itself is time consuming, such those involving large alignment file in BAM format generated by a company that is in a country different than that of the user, who needs to be a custodian of such files for subsequent analyses. Efficiency from raw data to outcome requires consideration of external factors, such as data communication and transfer.

Conclusion

A computational pipeline is necessary to achieve efficiency in preprocessing of high throughput sequencing data, especially when dealing with multiple samples. The application of the right pipeline depends on complexity of the tasks, users' programming skills, ease of setup, error troubleshooting, customization and scalability.

Acknowledgement

We thank M. Farhan Sjaugi for his help on eHive installation.

Appendix

Sample Exome Datasets and Software

The human exome data was downloaded from the NCBI Sequence Read Archive (SRA). Here is a list of the SRA IDs: SRR3270888.sra, SRR5182389.sra, SRR5280052.sra, SRR5306316.sra, SRR866988.sra.

The FASTQ files can be obtained using NCBI SRAtoolkit as per sample commands:

```
$ ./sra toolkit.2.8.1-3-centos_linux64/bin/prefetch -v SRR5215189
$ ./sra toolkit.2.8.1-3-centos_linux64/bin/fastq-dump SRR3270888.sra -split-files
```

The software included in this study are given below:

- FastQC version 0.11.4 for quality control check of FASTQ files. The software is available at Babraham Bioinformatics website (see section Relevant Websites).

- Trim Galore! Version 0.4.4, downloadable at Babraham Bioinformatics website (see section Relevant Websites). This software served as automation of adapter and quality trimming of raw FASTQ sequences.
- BWA version 0.7.12 (see Section Relevant Websites). This software maps the raw NGS reads against a large reference genome (e.g., GCRH37, Hg19). BWA could map not only short reads (up to 100 bp) but also long reads (up to 1 Mbp). It uses the Burrow-Wheeler Transformation (BWT) algorithm for mapping reads. The input of BWA is a FASTQ file and the output is either a SAM or BAM file.
- Samtools version 0.1.18 (see section Relevant Websites). This software provides useful utilities to work with SAM and BAM files. It allows users to view, sort and make index of the BAM/SAM files. In addition, it is also possible to call variants by using mpileup and bcftools.
- ANNOVAR (version 2016 Feb01) is a program built for functional annotation of genetic variants acquired from NGS data and it is written in Perl. To download it, users need to register at ANNOVAR website.
- IGV version 2.3.72 is a visualization tool for SNVs data developed by the Broad Institute, which can be obtained at Integrative Genomics Viewer website.

See also: Computational Pipelines and Workflows in Bioinformatics. Constructing Computational Pipelines. Natural Language Processing Approaches in Bioinformatics

References

- Afgan, E., *et al.*, 2016. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research* 44 (W1), W3–W10.
- Andersson, L., *et al.*, 2015. Coordinated international action to accelerate genome-to-phenome with FAANG, the Functional Annotation of Animal Genomes project. *Genome Biology* 16 (1), 57.
- Andrews, S., 2010. FastQC: A quality control tool for high throughput sequence data.
- Andrews, S., 2015. FastQC: A quality control tool for high throughput sequence data. Available at: <http://www.bioinformatics.babraham.ac.uk/projects/fastqc>.
- Blankenberg, D., *et al.*, 2014. Dissemination of scientific software with Galaxy ToolShed. *Genome Biology* 15 (2), 403.
- Keller, M.S., 1999. Take command: Cron: Job scheduler. *Linux Journal* 1999 (65es), 15.
- Krueger, F., 2015. Trim Galore!: A wrapper tool around Cutadapt and FastQC to consistently apply quality and adapter trimming to FastQ files.
- Li, H., *et al.*, 2009. The sequence alignment/map format and SAMtools. *Bioinformatics* 25 (16), 2078–2079.
- Li, H., 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*.
- Low, L., Tammi, M., 2017. *Bioinformatics: A Practical Handbook of Next Generation Sequencing and Its Applications*. World Scientific.
- Narasimhan, V., *et al.*, 2016. BCFtools/RoH: A hidden Markov model approach for detecting autozygosity from next-generation sequencing data. *Bioinformatics*. btw044.
- Sandve, G.K., *et al.*, 2013. Ten simple rules for reproducible computational research. *PLOS Computational Biology* 9 (10), e1003285.
- Severin, J., *et al.*, 2010. eHive: An artificial intelligence workflow system for genomic analysis. *BMC Bioinformatics* 11 (1), 240.
- Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38 (16), e164.

Relevant Websites

- http://www.openbioinformatics.org/annovar/annovar_download_form.php
ANNOVAR.
- <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
Babraham Bioinformatics.
- http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
Babraham Bioinformatics.
- <http://bio-bwa.sourceforge.net/>
Burrows Wheeler Aligner.
- <https://usegalaxy.org>
Galaxy.
- <https://galaxyproject.org/public-galaxy-servers/>
Galaxy Community Hub.
- <http://getgalaxy.org>
Galaxy Community Hub.
- <https://galaxyproject.org/>
Galaxy Community Hub.
- <https://www.broadinstitute.org/igv/>
Integrative Genomics Viewer.
- <http://samtools.sourceforge.net/>
SAMtools.

Genome Annotation: Perspective From Bacterial Genomes

Alan Christoffels and Peter van Heusden, University of the Western Cape, Cape Town, South Africa

© 2019 Elsevier Inc. All rights reserved.

Introduction

Advances in sequencing technologies have accelerated the generation of genomic data for a myriad of species in record time. The Genomes Online Database (<https://gold.jgi.doe.gov/>) GOLD provides comprehensive access to information regarding genome and metagenome sequencing projects, and their associated metadata. For example, as of 1st May 2018 the GOLD database contained 10,204 completely sequenced bacterial genomes. Access to these genomic resources allow scientists to compare genomes in different biomedical, environmental and/or geographical contexts and driving the development of comparative and visualization tools that underpin the field of comparative genomics. Therefore, it is important to accurately define the location and structure of genes in a genome and map these at different levels to proteins, function and pathways.

Genome annotation refers to the extraction of biological information from nucleotide sequencing data. Medigue and Moszer (2007) define a two-component hierarchy for genome annotation; namely, (i) a static view of genome annotation versus (ii) a dynamic view of genome annotation.

Static View of Genome Annotation

A range of gene prediction tools is used to predict the location of a gene (i.e., protein coding region or functional RNA). The predicted genes are used in sequence similarity searches against databases to assign a cellular function to the gene product. Functional assignment includes chemical and structural properties of proteins, sub-cellular localization, biological functions in a cell and protein domain identification.

Caveats to this approach

Gene prediction tools can miss small genes or genes with unusual nucleotide composition. For example the smallest gene identified is 39 nucleotides long PatS peptide (Yoon and Golden, 1998), yet gene prediction algorithms avoid such a short gene length parameter setting to optimize its performance (Tripp *et al.*, 2015). Furthermore, genome sequencing projects could be incomplete i.e., draft assembly and contain sequencing errors that will impact the accuracy of gene prediction tools.

Protein function assignment through sequence similarity search uses a similarity threshold score to limit spurious results. Accuracy of the annotations is improved by using annotations of model organisms that are manually curated; such as, data in RefSeq (Pruitt *et al.*, 2007) and UniProt (The UniProt Consortium, 2007).

Protein domain identification could lead to a single domain being shared by two unrelated proteins. The overall protein domain organisation is required to remove false positive functional assignment.

Dynamic View of Genome Annotation

The annotations obtained in Section “1” are enriched for biological context by integrating information at the level of regulatory and metabolic networks, and protein-protein interactions. Chromosome context is also used to improve genome annotations especially when sequence identity is low. For example, co-location genes in different genomes, gene order and gene fusion analysis allows for more accurate assignment of orthologues. These co-localized genes tend to be co-expressed and can provide evidence for protein interactions that feed into reconstruction of protein interaction networks. Annotations stemming from integrated information can be accessed through warehouses that facilitate exploration of the data via queries against a range of annotation attributes (O'Malley and Soyer, 2012; Triplet and Butler, 2011). Intermine (Smith *et al.*, 2012), is a framework that allows construction of such warehouses. More recently this framework was used as the bases for INDIGO – an Integrated data warehouse for microbial genomes (Alam *et al.*, 2013).

Genome Annotation Software

Genome annotation pipelines vary in their degree to which they are automated. Initial development of annotation pipelines included MAGPIE (Gaasterland and Sensen, 1996), GENEQUIZ (Hoersch *et al.*, 2000), AUTOFACT (Koski *et al.*, 2005), and BASys (Van Domselaar *et al.*, 2005). The latter is a web-based system for annotating bacterial chromosomes. Many of these tools do not allow users to edit the underlying annotations. Often editing applications are independent of the annotation database. For example, annotation and editing browsers such as ARTEMIS (Berriman and Rutherford, 2003) and Apollo (ref) provide a platform for reviewing and editing annotations but do not generate the annotations. Instead the annotations are imported from a GFF formatted file or database.

A combination of automated and manual annotation platforms include commercial systems such as ERGO (Overbeek *et al.*, 2003) or Pedant-Pro (successor of PEDANT (Frishman *et al.*, 2001)), and open-source systems, such as GenDB (Meyer *et al.*, 2003), Manatee (TIGR, unpublished), SABIA (Almeida *et al.*, 2004) and AGMIAL (Bryson *et al.*, 2006). Many of these annotation tools have other dependencies or the annotation pipeline requires a range of tools to be stitched together so that inputs and outputs are clearly defined. The installation of an array of tools (Fig. 1) required for annotation pipelines have been simplified with the availability of the conda package manager and the bioconda package channel. A package channel is a collection of precompiled software packages that can be installed by conda. The bioconda project creates conda packages for bioinformatics software, released through the conda package channel of the same name.

Curated Data Repositories for Genome Annotation

Data collections provide the basis for comprehensive genome annotation. These databases can be categorized as follows:

- (1) Nucleotide Resources
The International nucleotide sequence database collection (see “Relevant Websites section”) is a collaboration among DDBJ, EMBL and GENBANK. Other resources such as RefSeq (Pruitt *et al.*, 2007) at NCBI provide a non-redundant integrated set of nucleotides and proteins for a range of organisms.
- (2) Protein Resources
UniProt (The UniProt consortium, 2007) originated from a merger between SwissProt and PIR protein databanks. This resource represents an expertly curated annotation resource, inclusion of the relevant literature and numerous cross-references. The unprecedented volume of data that required curation had to be further sub-divided into two collections namely:
 - (i) UniProtKB/SwissProt – manually validated protein entries, and
 - (ii) UniProtKB/TrEMBL – computationally annotated records
- (3) Protein domain Resources
InterPro (Finn *et al.*, 2017) provides a unified resource that links databases such as PFAM (Finn *et al.*, 2016) and Prosite (Sigrist *et al.*, 2013). These proteins are organized into protein families based on their protein domain signatures.
- (4) Cluster of orthologous groups
Proteins are clustered based on BLASTP comparison of all proteins against all. Clusters are generated if a minimum of three distantly related organisms are present (Tatusov *et al.*, 2003).
- (5) Metabolic Pathways
These are provide by KEGG (Kanehisa *et al.*, 2017) and BioCyc (Karp *et al.*, 2015).

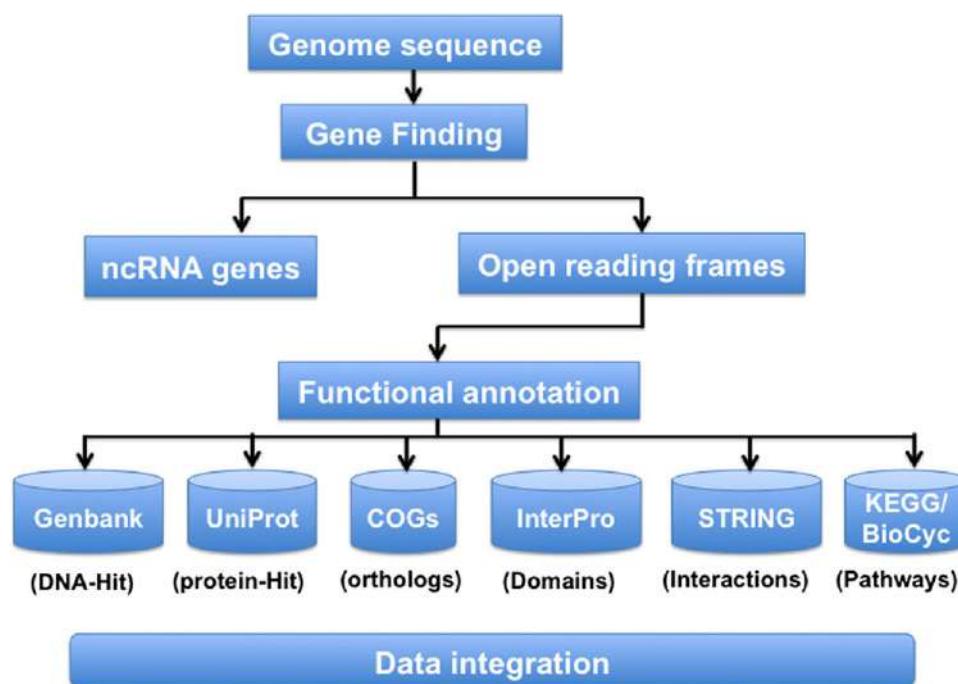


Fig. 1 Integrated Genome Annotation Pipeline.

- (6) Protein interactions
STRING database integrates predicted and known protein interaction data (von Mering *et al.*, 2007).
- (7) Structural databases
PDB (Berman *et al.*, 2007) stores 3-D protein structures.

Stepwise Approach to Genome Annotation

Genome annotation is preceded by a process of genome assembly using a reference genome-based method or de novo approach. The annotation of the assembled genome (Fig. 1) starts with identifying and masking RNA genes using RNAmmer (Lagesen *et al.*, 2007) and tRNAScanSE (Schattner *et al.*, 2005). Gene finding tools; such as, Prodigal (Hyatt *et al.*, 2010), GeneMark (Besemer *et al.*, 2001) and MetageneAnnotator (Noguchi *et al.*, 2008); are used to identified open reading frames (ORFs) in the genome sequence. These ORFs are BLAST searched against databases such as GENBANK and UniProt to identify putative functions and protein evidence. The ORFs are mapped to metabolic pathways using a KEGG database. Protein domains are identified through InterProScan searches. This search assigns GO terms to each of the protein domains and these features are later used to carry out functional enrichment analyses. ORFs are searched against the conserved domain database (Marchler-Bauer *et al.*, 2013) that includes COGs to identify corresponding orthologs.

Future Directions

In 2017 the NCBI announced that they would be re-annotating the 89732 bacterial genomes available in the Genbank repository using a new version of their automated prokaryotic genome annotation pipeline (PGAP). Analysis of the set of non-redundant proteins derived from this annotation exercise yielded a set of automated annotation rules that allow for rapid application of the best possible annotation for novel proteins (Richter and Rosselló-Móra, 2009). The future of prokaryotic genome annotation will almost certainly involve the application of curated biological knowledge for the development of heuristics for such multi-level annotation rules. Even these rules, however, only allow annotation of some 53% of the prokaryotic proteins in ReqSeq.

One obstacle facing annotators of de-novo assembled prokaryotic genomes is the difficulty in accessing expertise of biocuration communities, such as the one contributing to the NCBI rule engine described above. Historically the worlds of automated genome annotation and manual curation of annotation have been far apart. The emergence of rule engines (and possible in the future machine learning (Yip *et al.*, 2013) promises to accelerate the work of curators and bring the two worlds of annotation closer together.

Improved automated pipelines (e.g. Prokka (Seemann, 2014) and DFAST (Tanizawa, 2018)) and acceleration of pipeline components such as homology search tools (e.g. RapSearch (Ye *et al.*, 2011), DIAMOND (Buchfink 2014) and GHOSTX (Suzuki, 2014)) are addressing the need for faster annotation which in turn is driven by the decline in cost and increase in volume of DNA sequencing.

While there have been web-based portals for prokaryotic genome annotation available for more than a decade (e.g. MaGe (Vallejo *et al.*, 2006), RAST (Aziz *et al.*, 2008) and the many others mentioned in Siezen *et al.* (2010) advances in package deployment (e.g. conda) and the Galaxy web-based science platform (Afgan *et al.*, 2016) now make it possible to make prokaryotic genome annotation tools available “in house” to users who might not be familiar with the command line.

Closing Remarks

The decreasing cost of sequencing drives the production of genomic sequencing data such that genome annotation continues to be a requirement to provide biological context. Improving the accuracy of genome annotation, the process of extracting biological information from nucleotide sequencing data, requires a combination of ab-initio gene finding tools, supporting experimental evidence and manual curation. Adhering to such an integrated genome annotation strategy will allow scientific enquiry based on a rich source of meta data.

Acknowledgement

The authors are funded under the DST/NRF Research Chairs Initiative of South Africa.

See also: Natural Language Processing Approaches in Bioinformatics. Pipeline of High Throughput Sequencing

References

- Afgan, E., Baker, D., van den Beek, M., *et al.*, 2016. Nucleic Acids Res. 44: Web Server issue. W3–W10. doi:10.1093/nar/gkw343.
- Alam, I., Antunes, A., Kamau, A.A., Baalawi, W., Kalkatawi, M., *et al.*, 2013. INDIGO – Integrated data warehouse of microbial genomes with examples from the red sea extremophiles. PLOS ONE 8 (12), e82210. doi:10.1371/journal.pone.0082210.
- Almeida, L.G., Paixao, R., Souza, R.C., *et al.*, 2004. A system for automated bacterial (genome) integrated annotation- SABIA. Bioinformatics 20, 2832e2833.
- Aziz, R.K., Bartels, D., Best, A.A., *et al.*, 2008. The RAST Server: Rapid Annotations using Subsystems Technology. BMC Genomics 9, 75. doi:10.1186/1471-2164-9-75.
- Berman, H., Henrick, K., Nakamura, H., Markley, J.L., 2007. The worldwide Protein Data Bank (wwPDB): Ensuring a single, uniform archive of PDB data. Nucleic Acids Res. 35, D301eD303.
- Berriman, M., Rutherford, K., 2003. Viewing and annotating sequence data with Artemis. Brief. Bioinform. 4, 124e132.
- Besemer, J., Lomsadze, A., Borodovsky, M., 2001. GeneMarkS: A selftraining method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. Nucleic Acids Res. 29, 2607–2618.
- Bryson, K., Loux, V., Bossy, R., *et al.*, 2006. AGMIAL: Implementing an annotation strategy for prokaryote genomes as a distributed system. Nucleic Acids Res. 34, 3533e3545.
- Finn, R.D., Attwood, T.K., Babbitt, P.C., *et al.*, 2017. InterPro in 2017 – Beyond protein family and domain annotations. Nucleic Acids Res. 45, D190–D199.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. Nucleic Acids Res. 44, D279–D285.
- Frishman, D., Albermann, K., Hani, J., *et al.*, 2001. Functional and structural genomics using PEDANT. Bioinformatics 17, 44e57.
- Gaasterland, T., Sensen, C.W., 1996. Fully automated genome analysis. that reflects user needs and preferences. A detailed introduction to the MAGPIE system architecture. Biochimie 78, 302e310.
- Hoersch, S., Leroy, C., Brown, N.P., Andrade, M.A., Sander, C., 2000. The GeneQuiz web server: Protein functional analysis through the Web. Trends Biochem. Sci. 25, 33e35.
- Hyatt, D., Chen, G.L., Locascio, P.F., Land, M.L., Larimer, F.W., *et al.*, 2010. Prodigal: Prokaryotic gene recognition and translation initiation site identification. BMC Bioinform. 11, 119.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K., 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 45, D353–D361.
- Karp, P.D., Latendresse, M., Paley, S.M., *et al.*, 2015. Pathway tools version 19.0: Integrated Software for pathway/genome informatics and systems biology. Brief. Bioinform..
- Koski, L.B., Gray, M.W., Lang, B.F., Burger, G., 2005. AutoFACT: An automatic functional annotation and classification tool. BMC Bioinform. 6, 51.
- Lagesen, K., Hallin, P., Rødland, E.A., Staerfeldt, H.H., Rognes, T., *et al.*, 2007. RNAmmer: Consistent and rapid annotation of ribosomal RNA genes. Nucleic Acids Res. 35, 3100–3108.
- Marchler-Bauer, A., Zheng, C., Chitsaz, F., *et al.*, 2013. CDD: Conserved domains and protein three-dimensional structure. Nucleic Acids Res. 41, D348–D352.
- Medigue, C., Moszer, I., 2007. Annotation, comparison and databases for hundreds of bacterial genomes. Res. Microbiol. 158, 724–736.
- Meyer, F., Goesmann, A., McHardy, A.C., *et al.*, 2003. GenDBan open source genome annotation system for prokaryote genomes. Nucleic Acids Res. 31, 2187e2195.
- Noguchi, H., Taniguchi, T., Itoh, T., 2008. MetaGeneAnnotator: Detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. DNA Res. 15, 387–396.
- O'Malley, M.A., Soyer, O.S., 2012. The roles of integration in molecular systems biology. Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences 43, 58–68. doi:10.1016/j.shpsc.2011.10.006.
- Overbeek, R., Larsen, N., Walunas, T., *et al.*, 2003. The ERGO genome analysis and discovery system. Nucleic Acids Res. 31, 164e171.
- Pruitt, K.D., Tatusova, T., Maglott, D.R., 2007. NCBI reference sequences (RefSeq): A curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res. 35, D61eD65.
- Richter, M., Rosselló-Móra, R., 2009. Shifting the genomic gold standard for the prokaryotic species definition. PNAS 106 (45), 19126–19131. doi:10.1073/pnas.0906412106.
- Schattnr, P., Brooks, A.N., Lowe, T.M., 2005. The tRNAscan-SE, snoscan and snoGPS web servers for the detection of tRNAs and snoRNAs. Nucleic Acids Res. 33, W686–W689.
- Seemann, T., 2014. Prokka: Rapid prokaryotic genome annotation. Bioinformatics 30 (14), 2068–2069. doi:10.1093/bioinformatics/btu153.
- Siezen, R.J., van Hijum, S.A.F.T., 2010. Genome (re-)annotation and open-source annotation pipelines. Microbial Biotechnology 3 (4), 362–369. doi:10.1111/j.1751-7915.2010.00191.x.
- Sigrist, C.J.A., de Castro, E., Cerutti, L., *et al.*, 2013. New and continuing developments at PROSITE. Nucleic Acids Res. 41, D344–D347.
- Smith, R.N., Aleksic, J., Butano, D., *et al.*, 2012. InterMine: A flexible data warehouse system for the integration and analysis of heterogeneous biological data. Bioinformatics 28, 3163–3165. doi:10.1093/bioinformatics/bts577.
- Suzuki, S., Kakuta, M., Ishida, T., Akiyama, Y., 2014. GHOSTX: An Improved Sequence Homology Search Algorithm Using a Query Suffix Array and a Database Suffix Array. PLoS ONE 9 (8), e103833. doi:10.1371/journal.pone.0103833.
- Tatusov, R.L., Fedorova, N.D., Jackson, J.D., *et al.*, 2003. The COG database: An updated version includes eukaryotes. BMC Bioinform. 4, 41.
- Tanizawa, Y., Fujisawa, T., Nakamura, Y., 2017. DFAST: A flexible prokaryotic genome annotation pipeline for faster genome publication. Bioinformatics 34 (6), 1037–1039. doi:10.1093/bioinformatics/btx713.
- The UniProt Consortium, 2007. The universal protein resource (UniProt). Nucleic Acids Res. 35, D193eD197.
- Triplet T., Butler G., 2011. Systems biology warehousing: Challenges and strategies toward effective data integration. In: Proceedings of the 3rd International Conference on Advances in Databases, Knowledge, and Data Applications, St. Maarten. IARIA. pp 34–40.
- Vallenet, D., Labarre, L., Rouy, Z., *et al.*, 2006. MaGe: A microbial genome annotation system supported by synteny results. Nucleic Acids Res 34 (1), 53–65. doi:10.1093/nar/gkj406.
- Van Domselaar, G.H., Stothard, P., Shrivastava, S., *et al.*, 2005. BASys: A web server for automated bacterial genome annotation. Nucleic Acids Res. 33, W455eW459.
- Von Mering, C., Jensen, L.J., Kuhn, M., *et al.*, 2007. STRING 7- recent developments in the integration and prediction of protein interactions. Nucleic Acids Res. 35, D358eD362.
- Yip, K.Y., Cheng, C., Gerstein, M., 2013. Machine learning and genome annotation: a match meant to be? Genome Biology 14, 205. doi:10.1186/gb-2013-14-5-205.
- Yoon, H.S., Golden, J.W., 1998. Heterocyst pattern formation controlled by a diffusible peptide. Science 282, 935–938. doi:10.1126/science.282.5390.935.

Relevant Websites

<https://www.climb.ac.uk/>

Cloud infrastructure for microbial bioinformatics.

<https://gold.jgi.doe.gov/>

Genomes Online Database.

<https://bioconda.github.io/recipes.html#recipes>

Packages available in the Bioconda channel.

<http://www.insdc.org>

INSDC.

<https://bioconda.github.io/recipes.html%23recipes>

Packages available in the Bioconda channel.

Biographical Sketch

Alan Christoffels After completing my studies in South Africa in 2001, I moved to Singapore for a postdoc with a focus on genome evolution. During this time I contributed to the annotation and genome analysis of the Pufferfish genome and developed method for detecting genome duplication events. This methodology was later used in other international genome projects after I established my laboratory in Singapore in 2004. In 2007 I returned to South Africa and established my group at the University of the Western Cape at the South African National Bioinformatics Institute. For the following 8 years, I was part of an international execute team driving the genome sequencing and analysis of the Tsetse genome. As of 2013, the focus of the lab has shifted to bacterial genomes where most of my funding is directed at *M. tuberculosis*. My bioinformatics laboratory focuses on building tools and analyzing data that leads to a better understanding of human-M. tuberculosis interaction with a focus on functional non-coding RNA, drug target identification and genome evolution. Underpinning these Tuberculosis projects are methods to analyze next generation sequencing data.

Peter van Heusden I first joined the South African National Bioinformatics Institute as a computer systems administrator in 1996 and was soon involved in assisting with the bioinformatics research of the Institute. In the past decade I have assisted with annotating the genome of the African coelacanth (*Latimeria chalumnae*) and that of the Asian sea bass (*Lates calcarifer*). Since 2014 I have led a research software engineering team at SANBI with a focus on automating scientific workflows, specifically those related to analysis *Mycobacterium tuberculosis* genomic data.

Next Generation Sequencing Data Analysis

Ranjeev Hari and Suhanya Parthasarathy, Perdana University, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The era of next generation sequencing (NGS) began following the first reports of this innovative technique in 2005 (Margulies *et al.*, 2005; Shendure and Ji, 2008). The high-throughput NGS technology, which uses parallel amplification and sequencing yields shorter read lengths, giving an average raw error rates of 1%–1.5% (Shendure and Ji, 2008) when compared to conventional Sanger sequencing protocols, which can generate to 1000 bp of 99.999% per base accuracies. While automation attempts of traditional dideoxy DNA sequencing by Sanger method improved the efficiency of DNA sequencing; however, in terms of cost and time, NGS technology was regarded as superior. An early method called massively parallel sequencing (MPS) that was introduced by Lynxgen Therapeutics set the stage for high throughput sequencing (Brenner *et al.*, 2000). The first NGS machine, the GS20, that was made available for researchers in 2005 by 454 Life Sciences (Basel, Switzerland) is based on large-scale parallel pyrosequencing by microbeads in micro-droplets of water in oil emulsion (Henson *et al.*, 2012).

The major NGS technology platforms in the market for whole-genome sequencing are primarily from brand names such as Illumina, Roche 454, Solid and IonTorrent. Each platform has its advantages and disadvantages and cost implications in terms of reliability, time and money (Table 1). Depending on each NGS platform, they offer values that may be attractive for specific purposes. For instance, the Ion Torrent is often positioned as a general purpose sequencer as well as in diagnostic protocols due to the quicker turnaround time (Tarabeux *et al.*, 2014). However, longer reads technology offered by Illumina and Roche are desirable but the cost involved for Roche 454 FLX is very steep rendering it impractical for large scale genome projects. Reads from Pacific Bioscience machine, PacBio, are generally not used in large genome projects for direct sequencing but can be useful in resolving repetitive regions and ambiguous regions due to capability of generating very long read lengths (Table 1).

NGS Process Workflow

The common processes involved in general NGS platforms are library preparation, library amplification, and sequencing (Figure 2.3). The starting material for library preparation can be from either RNA or DNA (genomic source or PCR-amplified). In the case of RNA, it has to be transcribed into cDNA because as of now, NGS machines sequence only DNA directly. Since target library molecules sequenced on each NGS platform are required to be in specific lengths, genomic DNA requires fractionation and size selection, which is performed by sonication, nebulization, or enzymatic techniques followed by gel electrophoresis and excision. For instance, Illumina NGS platform's standard fragment size is in the range of 300 and 550 bp including adapters. Generally, libraries are built by adding NGS platform-specific DNA adapters to the DNA molecules. These adapters facilitate the binding of the library fragments to a surface such as a microbead (454, Ion PGM, SOLiD) or a glass slide (Illumina, SOLiD).

However, depending on the specific NGS platform, the library construction step has to be customised to fit the sequencing protocol. Generally, DNA library construction directly depend on its applications and can be divided mainly into fragment libraries and mate-paired libraries. In fragment libraries, target genomic sequences are fragmented to smaller sizes typically up to 5 times the NGS platform's read length capabilities. Subsequently, sequencing adapters are attached to these fragments allowing the NGS platform to sequence from the adaptor tags. Typically, fragment libraries are of single end and paired-end. With adaptor tags at both forward and reverse sites, NGS platforms are able to sequence from both ends in paired-end libraries. The applications of fragment libraries are mainly for variant calling, copy number detection and genome reconstruction. While other types of NGS

Table 1 Comparison between different NGS platforms

	454 FLX	HiSeq2000	Solid 5500XL	IonTorrent (318 Chip)	PacBio
Company	Roche	Illumina	Life technologies	Life technologies	Pacific biosciences
Nucleotides/Run	700 Mbp	540–600 Gbp	180 Gbp	800 Mbp	0.5–1 Gbp
Maximum read length	700 bp	2 × 100 bp	75 + 35 bp	200 bp	10 kbp–> 40 kbp
Pairs	2 × 150 bp	2 × 150 bp	2 × 60 bp	N/A	
Run time	23 hours	11 days	12–16 days	4.5 hours	30 m–4 hours
Reagent cost per Mbp (USD)	7	0.04	0.07	1	0.13–0.60
Advantages	–Read length –Fast	–High throughput –Read length	–Low cost/base –Accuracy	–Less expensive –Fast	–Read length –Fast
Disadvantages	–Expensive –Homopolymer errors –Low throughput	–High DNA concentration –Short reads	–Slow –Palindromic errors –Short reads	–Homopolymer errors	–Error rate

sequencing techniques may exist, the fundamental principles are similar in fragment library construction. Other techniques may involve intermediary steps such as fragment capture, immune-hybridizations or reverse transcription.

In mate-pair library construction, target DNA sequences are fragmented to size more than 1000 bp that are often longer than the read length capacities of the NGS platforms. The required fragment size is excised from a gel electrophoresis run of the sheared DNA corresponding to the intended library size usually 2 kbp, 5 kbp or 10 kbp. These isolated DNA fragments are circularised by ligation through addition of biotinylated adapters to the ends to promote sequence specific ligation instead of ligating across fragments. Once again, the circularised DNA are fragmented and biotinylated fragments are purified by affinity capture. Sequencing adapters are added to the size selected sequences. The manipulation performed during the library construction step allows identification of the terminal sequences of the circularised DNA by sequencing. Having known the sequence length *a priori*, computational algorithms can help differentiate large scale genomic rearrangements from a reference sequence besides typically being used to guide the scaffolding process in a genome assembly.

In the sequencing step, having known adaptor sequences allows the amplification of library fragments either by emulsion PCR (emPCR) or bridge PCR which are methods specific to the NGS platforms. Commonly, irrespective of platforms, NGS exploits parallelism in a factor of ten thousand to billions of library fragments. Generally, this is achieved by reiterated cycles of nucleotide addition using DNA polymerases or ligases (SOLiD), detection of incorporated nucleotides and washing steps. The extensive washing and repetitive steps, albeit automatic, may thus take sequencing to complete from hours to days. Base calling is the process of algorithmically deciding the incorporated nucleotide from the signal intensities that are detected during sequencing process.

Every NGS platform relies on slightly different strategies to generate and detect signals (Luo *et al.*, 2012; Merriman *et al.*, 2012; Ronaghi *et al.*, 1998). For instance, the chemistry used by Illumina and SOLiD sequencing is similar to the principles of sequencing by synthesis. Using four separate fluorescent-labelled nucleotides, its flushed over the glass slide and fluorescence signals are repetitively captured when nucleotides become incorporated to target sequences. In contrast, flooding of non-labelled nucleotides in 454 and Ion PGM sequencing are detected by pyrophosphate or proton release as part of its unique chemistry. Typically, the proton release is measured as a pH change by the semiconductor chip of the Ion Torrent instrument (Merriman *et al.*, 2012). On the other hand, pyrophosphate signal in the 454 system relies on chemistry of the enzyme luciferase to induce light signals (Ronaghi *et al.*, 1998). Altogether, these signals respective to each platform will be converted into basecalls and converted into raw sequencing data that is generally output in FASTQ format that includes the basecall qualities along with the DNA sequence.

While many platforms exist, markedly there are many advantages of NGS, compared to the conventional Sanger sequencing, such as: (i) NGS permits a considerably higher degree of parallelism than Sanger's conventional capillary-based sequencing. (ii) Removal of bottlenecks and limitations in older methods (*E. coli* transformation and colony picking) by *in vitro* construction of a sequencing library, followed by *in vitro* clonal amplification further enforcing parallelism in the workflow (iii) NGS system usually uses reagents that operate on immobilized target on arrays in microliter quantities, which costs are gained back over the full set of sequencing features on the array because the volume per feature are in the range of picoliters to femtoliters. Overall, these advantages translate into intensely lower costs for DNA sequence production using NGS technologies (Shendure and Ji, 2008). Therefore, in any NGS projects, the high-throughput nature of the method is favoured (Figs. 1 and 2).

Typical Data Analysis Workflow for NGS

NGS analysis utilize bioinformatics approaches in order to convert signals from the machine to meaningful information that involve signal conversion to data, annotations or catalogued information, and actionable knowledge. Primarily, the NGS bioinformatics analysis is divided into three distinct phases as primary, secondary and tertiary analyses. In the primary analyses, raw data from the sequencers are converted into nucleotide base and short read data. Secondary analyses apply detailed bioinformatics methodology specific to the NGS technique that was employed which may involve read alignments or read assembly. Usually, secondary analyses have the most complex analyses workflow and is usually run in a sequence as a pipeline. Besides, depending on the type of NGS technique that are employed, the analysis pipeline may differ greatly. For example, RNA-seq data, which characterizes transcriptomes, differ in secondary analysis approach compared to ChIP-seq data, which investigates genome-wide epigenetic mechanisms. Lastly, by tertiary analyses, the previous results obtained can be associated and understood in a biological context. However, tertiary bioinformatics analyses can be an iterative process that involve rigorous statistical and computational biology methods.

Sequence Generation

The initial primary analysis is usually transient as the sequencing machines detectors receives the signals obtained from the high throughput reactions. Thus, the base-calling and recording process is tightly integrated with the sequencing instruments resulting in quality scores corresponding to the short reads nucleotide sequence being output parallelly. The primary analysis software is installed by machine vendors on the workstation supporting the sequencing instrument. The software can be run in high-performance cluster systems for faster results. Besides converting raw signals to base calls, some software tools include demultiplexing of multiple samples into a single pooled and indexed run (Dodt *et al.*, 2012).

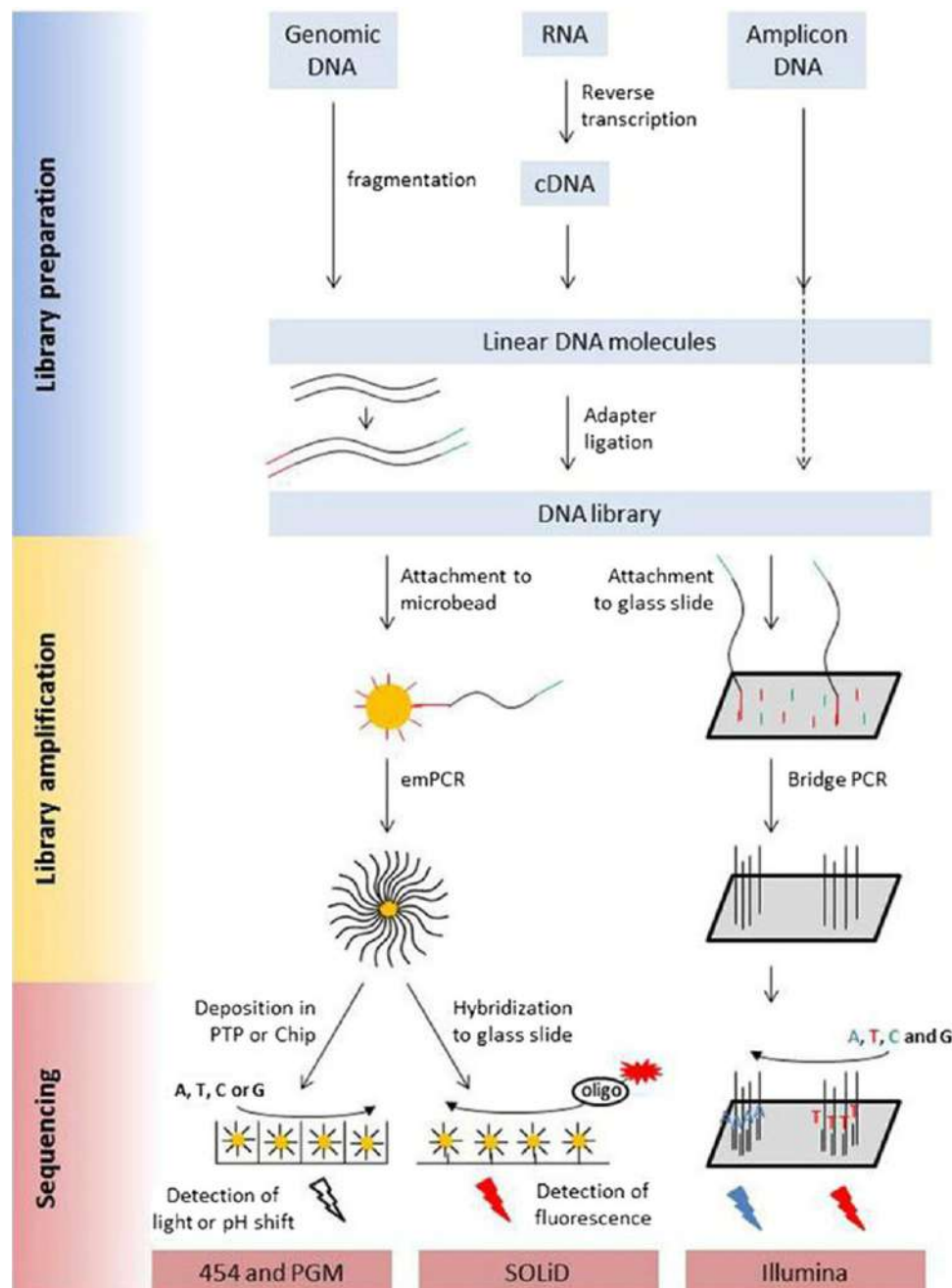


Fig. 1 Schematic diagram of the process involved in common NGS platforms. Adapted from Knief, C., 2014. Analysis of plant microbe interactions in the era of next generation sequencing technologies. *Plant Genet. Genom.* 5, 216. Available at: <https://doi.org/10.3389/fpls.2014.00216>.

NGS Raw Data Pre-Processing

As described earlier, NGS platforms suffer from higher error rates compared to Sanger sequencing (Nakamura *et al.*, 2011; Shendure and Ji, 2008). However, as part of primary analysis, different approaches and algorithms have been developed to compensate and spot these errors (Margulies *et al.*, 2005). Moreover, bases that have inaccuracy less than 0.1% can be carefully chosen algorithmically. As a simple approach, error-rates can be decreased by performing the DNA sequencing with high coverage, of at least 20–60-fold, depending on the sequencing project's goal (Luo *et al.*, 2012; Margulies *et al.*, 2005; Voelkerding *et al.*, 2009). Notably, each sequencing read can be categorised as a distinct genotype but in fact could be the result of sequencing error. Thus, it is very important to use established methods in differentiating these two causes of variation as it may lead to inaccurate results when flawed.

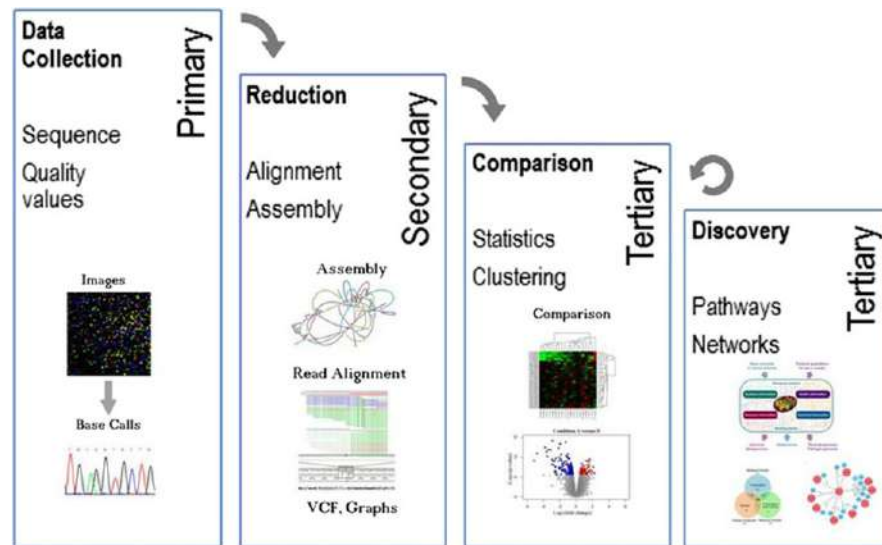


Fig. 2 Workflow of NGS data analysis in three phases: primary, secondary and tertiary. The tertiary phase of Comparison and Discovery is indicated as an iterative process.

To improve the quality of the data after base-calling, Phred-based filtering algorithms can be used to filter or remove low quality sequencing reads (Margulies *et al.*, 2005). These filters discard reads with low-quality, uncalled, and ambiguous bases besides clipping the lower quality 3'-ends of reads. All such filters use the quality information contained in the FASTQ file that are computed by the NGS platform at each base during the base calling procedure. Previous studies (Minoche *et al.*, 2011) have shown the effect of different filtering approaches on Illumina data and suggests that it can reduce error rates to less than 0.2% by eliminating around 15%–20% of the low-quality bases, mostly via 3'-end trimming that are prone to errors. Another study has supported the findings that a 5-fold decrease of error rate can be observed by applying a filter (Phred score of Q30, with 0.1% likelihood of a false basecall) that eliminated reads with low quality bases (Nguyen *et al.*, 2011). It may be useful to note that low quality bases are sometimes localised in specific regions of a genome. It is important to note that removal of these reads may introduce potential bias in the quantitative studies undertaken (Minoche *et al.*, 2011; Nakamura *et al.*, 2011). Therefore, read clipping strategy can be used to remove the erroneous bases from the left or right edges of the reads without filtering the whole reads in order to address errors that are usually present in the reads edges alone.

Apart from read clipping and filtering methods, several error correction tools (e.g., Coral, HiTEC, Musket, Quake, RACER, Reptile, or SHREC) could be used as a complementary strategy to reduce sequencing error rates in reads (Knief, 2014). Generally, these error correction methods make use of high sequencing coverage in order to identify and correct errors using the laws of probability and statistics. Moreover, these algorithms often consider quality scores of the examined bases besides looking at neighbouring base quality values. For instance, some of these tools are able to correct substitution errors in Illumina sequencing data (Ilie and Molnar, 2013; Liu *et al.*, 2013; Yang *et al.*, 2010) while others (Coral, HSHREC, KEC, and ET) are designed to include indel correction algorithms that are available for the analysis of Roche's 454 and IonTorrent data (Salmela, 2010; Salmela and Schröder, 2011; Skums *et al.*, 2012). The relevance of error correction tools is seen as a very useful strategy in *de novo* genome sequencing, resequencing and amplicon sequencing projects with benefits ranging from finding more optimal assembly in the DBG and reducing overall memory footprint to perform the assembly stage (Skums *et al.*, 2012; Yang *et al.*, 2010).

Genome Assembly

After the pre-processing of sequence reads, we can assemble the pre-processed reads into contigs. The process of assembly of sequencing reads generated from NGS technology involves the reduction of redundant data by contiguously placing reads by overlapping them adjacent to each other in an optimal way (Miller *et al.*, 2010). Instead of reads, when contigs (assembled set of reads) undergo the previously described process with long length information, it is known as scaffolding. In other words, it is a process of reconstructing the target as such to groups reads into contigs and contigs into scaffolds.

Generally, the size and accuracy of the contigs and scaffolds are important statistics in genome assemblies (Miller *et al.*, 2010). The quality of genome assemblies is usually described by maximum length, average length, combined total length, and N50. The contig N50 is the length of the smallest contig in the set that contains the fewest (of the largest) contigs whose joint length represents at least 50% of the assembly (Miller *et al.*, 2010). Generally, larger N50 values imply a higher quality genome assembly that describes lesser overall fragments. Typical high-coverage genome projects have N50 values that range in megabases; however, they are dependent on the genome size and is not a good measure to compare between unrelated assemblies instead of the same. Assembly accuracy is tough to quantify. Nevertheless, mapping the assembled contigs/scaffolds to reference genomes is useful to examine its quality if the references exist.

As outlined earlier, an assembly is an ordered data construction that maps the sequencing data to a supposed reconstruction of the target (He *et al.*, 2013). Contigs are reconstructed sequences from sequence alignment of reads which give rise to a consensus sequence. The scaffolds is a higher order organisation of sequences which define the contig order and orientation and the sizes of the gaps between contigs. The scaffolds represent more contiguous sequences mimicking the physical genome composition. Scaffold sequences could have N's in the gaps between contigs. The number of consecutive N's may show the gap length estimate during the assembly process based on the bridging of mate pair reads (Miller *et al.*, 2010).

There are many well-established software for assembling sequencing reads into contigs/scaffolds. In general, these genome assemblers can be grouped into three categories based on their approaches (Miller *et al.*, 2010): (1) The Overlap/Layout/Consensus (OLC) approaches depend on an overlap graph; (2) the *de Bruijn* Graph (DBG) use some form of *k*-mer graph; and, (3) the greedy graph algorithms can use OLC or DBG (Table 2).

There are many factors that need to be considered when choosing the most appropriate genome assembler especially when considering for large whole genome sequencing project. Among these factors are the choice of algorithm, compatibility with the NGS platform, the support of the assembly of large genomes, and parallel-computing support for speeding-up the assembly (Zerbino and Birney, 2008). Generally, the choice of algorithm and software will directly determine the memory requirements and speed of assembly. In general, DBG assemblers are faster but require large amount of memory compared to OLC assemblers.

Read Mapping

Whenever a reference genome becomes available, instead of a *de novo* assembly strategy, reads are mapped or aligned to the reference genome prior to subsequent analysis steps. The goal of mapping is to realign the vast number of reads back to the respective regions it likely originated from. The mapping of the reads to the reference genome typically involves the alignment of millions of short reads to the genome using fast algorithms. The algorithms are able to function parallelly while taking into account mutations such as polymorphisms, insertions and deletions in order to produce the alignment. In well-known aligners, for example BLAST, the individual query sequence is searched against a reference database using hash tables and seed and extend approaches. With NGS data, often similar methods are adapted to scale the alignment of short query sequences that are in the millions against a single reference genome of large sequences. Advances in mapping algorithms using various other techniques has improved alignment speed while reducing memory and space requirements. Examples of mapping software that are well known and used in NGS data includes SOAP2 (Short Oligonucleotide Alignment Program), BWA (Burrow-Wheels Alignment), NovoAlign and Bowtie 2.

The widely used format of storing mapping information of the reads to the genome is SAM (Sequence Alignment Map) format or its compressed binary form called BAM. While the BAM file is smaller and optimized for machine reading, the SAM file is human readable albeit slower for computer operations. There are 11 mandatory fields in the SAM format specification. Commonly, SAMtools software is used to manipulate and read both BAM and SAM formats.

Typically, mapping of reads to the reference genome is followed by collecting data about the mapping statistics. The summary statistics that is mainly of interest is the percentage of aligned reads, or the mapping rate. The mapping rate of reads to the reference genome are usually only 60%–75%. Besides the limitation due to the intrinsic properties of NGS data and technique that it was generated from, the inability to map to the reference genome can be ascribed to challenging regions in the genome, such as repeat rich regions, that aligners are not able to map to. Moreover, short read lengths in most high throughput NGS technology limits the alignment in mapping to span a small region hence limiting its coverage to convenient areas of the genome. Besides that, limitations such as NGS sequencing error, algorithmic robustness, mutational load and variation contributes to the low mapping rate.

The mapping file generated can be further inspected by regional in-depth using visualization tools such as genome viewers which plot pileups (the stacked alignment of the reads). Visualization of reads mapped, for instance, can be important in diagnosing problems in read alignment in certain regions, detecting duplicates, and visualizing variations. Commonly used genome browsers that enable the reading of SAM files include Integrated Genome Viewer (IGV), and Tablet while some web-based browser that achieve similar visualizations include JBrowse, NGB and UCSC genome browser.

Table 2 Summary of assembly software used for *de novo* assembly of genomes

Assembler	Algorithm	Preferred data	Multithreading	Target genome
SGA	OLC	Illumina	Yes	Large
SOAPdenovo2	DBG	Illumina	Yes	Large
ALLPATHS-LG	DBG	Illumina	No	Large
CLC	DBG	Mixed	No	Large
SSAKE, SSHARCGS and VCAKE	Greedy	Illumina	No	Small
Edena	OLC	Illumina	No	Small
Newbler	OLC	454	No	Large
CABOG	OLC	Mixed	No	Large
Euler	DBG	454 + Sanger	No	Small
Velvet	DBG	Illumina	No	Small
ABYSS	DBG	Illumina	Yes	Large

Tertiary Analysis

The tertiary process of analysing NGS data can be quite diverse depending on the scenario and context of a study. Generally, the reads that are representative of the underlying annotations will characterize the functional aspects of the study. As such, the corresponding statistics used are usually descriptive. In other cases, where a comparison is being made to a reference or a control, rigorous statistical tests are employed taking into account read counts in the target regions representative in each treatment groups. Applying statistical models to identify bias, accounting covariates and testing for significant difference are common steps in comparative analysis. The resulting outcomes of the analysis may provide a collection of target annotations or genes. Such a gene list can be subsequently analysed for enrichment in regards to gene ontology (GO) terms to infer the collective function, biological process and cellular compartmentalization. With such a gene list, the pathways being affected can also be mapped to grasp a better biological understanding of the process. In other derivative NGS techniques such as ChIP-seq or Hi-C, tertiary analysis may additionally involve deriving profiles that are commonly occurring in the interactions being observed. Thus, new motifs, structural interactions and regulatory signals that are being generated for a particular condition can be characterized. NGS techniques when applied to a population could derive meaningful interaction of evolutionary forces besides characterizing the differences in genetic composition either in terms of polymorphisms when observing the same species or taxonomy when studying metagenomics.

Conclusion

Approaches to NGS data analysis are diverse and dependant on the technology and methods being employed. Nevertheless, among the multiple stages that are involved in NGS data analysis, steps in primary and secondary analysis is generally a prerequisite in all NGS projects. Therefore, primary and secondary analyses must be carefully performed in order to prevent errors being carried over to tertiary analysis. In tertiary analysis, insights can be generated from the inclusion of annotation, network, and interaction information from external databases to expand on gene lists and profiles found in earlier steps. Collectively, these steps are required frequently, hence, independent bioinformatics labs create analysis pipelines for their in-house routine analyses. However, as NGS technology grows and improves, the methods for analyses may require further evaluation and integration with latest technologies.

See also: Computational Pipelines and Workflows in Bioinformatics. Constructing Computational Pipelines. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Genome Annotation: Perspective From Bacterial Genomes. Natural Language Processing Approaches in Bioinformatics. Pipeline of High Throughput Sequencing

Reference

- Brenner, S., Johnson, M., Bridgham, J., *et al.*, 2000. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nat. Biotechnol.* 18, 630–634. Available at: <https://doi.org/10.1038/76469>.
- Dodd, M., Roehr, J.T., Ahmed, R., Dieterich, C., 2012. FLEXBAR – Flexible barcode and adapter processing for next-generation sequencing platforms. *Biology* 1, 895–905. Available at: <https://doi.org/10.3390/biology1030895>.
- Henson, J., Tischler, G., Ning, Z., 2012. Next-generation sequencing and large genome assemblies. *Pharmacogenomics* 13, 901–915. Available at: <https://doi.org/10.2217/pgs.12.72>.
- He, Y., Zhang, Z., Peng, X., Wu, F., Wang, J., 2013. De novo assembly methods for next generation sequencing data. *Tsinghua Sci. Technol.* 18, 500–514. Available at: <https://doi.org/10.1109/TST.2013.6616523>.
- Ilie, L., Molnar, M., 2013. RACER: Rapid and accurate correction of errors in reads. *Bioinformatics* 29, 2490–2493. <https://doi.org/10.1093/bioinformatics/btt407>.
- Knief, C., 2014. Analysis of plant microbe interactions in the era of next generation sequencing technologies. *Plant Genet. Genom.* 5, 216. Available at: <https://doi.org/10.3389/fpls.2014.00216>.
- Liu, Y., Schröder, J., Schmidt, B., 2013. Muskiet: a multistage k-mer spectrum-based error corrector for Illumina sequence data. *Bioinformatics* 29. <https://doi.org/10.1093/bioinformatics/bts690>.
- Luo, C., Tsementzi, D., Kypides, N., Read, T., Konstantinidis, K.T., 2012. Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. *PLOS ONE* 7, e30087. Available at: <https://doi.org/10.1371/journal.pone.0030087>.
- Margulies, E.H., Maduro, V.V.B., Thomas, P.J., *et al.*, 2005. Comparative sequencing provides insights about the structure and conservation of marsupial and monotreme genomes. *Proc. Natl. Acad. Sci. USA* 102, 3354–3359. Available at: <https://doi.org/10.1073/pnas.0408539102>.
- Merriman, B., Rothberg, J.M., Ion Torrent R&D Team, 2012. Progress in ion torrent semiconductor chip based sequencing. *Electrophoresis* 33, 3397–3417. Available at: <https://doi.org/10.1002/elps.201200424>.
- Miller, J.R., Koren, S., Sutton, G., 2010. Assembly algorithms for next-generation sequencing data. *Genomics* 95, 315–327. Available at: <https://doi.org/10.1016/j.ygeno.2010.03.001>.
- Minoche, A.E., Dohm, J.C., Himmelbauer, H., 2011. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. *Genome Biol.* 12, R112. Available at: <https://doi.org/10.1186/gb-2011-12-11-r112>.
- Nakamura, K., Oshima, T., Morimoto, T., *et al.*, 2011. Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Res.* 39, e90. Available at: <https://doi.org/10.1093/nar/gkr344>.
- Nguyen, P., Ma, J., Pei, D., *et al.*, 2011. Identification of errors introduced during high throughput sequencing of the T cell receptor repertoire. *BMC Genom.* 12, 106. Available at: <https://doi.org/10.1186/1471-2164-12-106>.

- Ronaghi, M., Uhlén, M., Nyrén, P., 1998. A sequencing method based on real-time pyrophosphate. *Science* 281, 363–365. Available at: <https://doi.org/10.1126/science.281.5375.363>.
- Salmela, L., 2010. Correction of sequencing errors in a mixed set of reads. *Bioinformatics* 26, 1284–1290.
- Salmela, L., Schröder, J., 2011. Correcting errors in short reads by multiple alignments. *Bioinformatics* 27, 1455–1461.
- Shendure, J., Ji, H., 2008. Next-generation DNA sequencing. *Nat. Biotechnol.* 26, 1135–1145. Available at: <https://doi.org/10.1038/nbt1486>.
- Skums, P., Dimitrova, Z., Campo, D.S., *et al.*, 2012. Efficient error correction for next-generation sequencing of viral amplicons, in: *BMC Bioinformatics*. BioMed Central. p. S6.
- Tarabeux, J., Zeitouni, B., Moncoutier, V., *et al.*, 2014. Streamlined ion torrent PGM-based diagnostics: BRCA1 and BRCA2 genes as a model. *Eur. J. Hum. Genet.* 22, 535–541. Available at: <https://doi.org/10.1038/ejhg.2013.181>.
- Voelkerding, K.V., Dames, S.A., Durtschi, J.D., 2009. Next-generation sequencing: From basic research to diagnostics. *Clin. Chem.* 55, 641–658. Available at: <https://doi.org/10.1373/clinchem.2008.112789>.
- Yang, X., Dorman, K.S., Aluru, S., 2010. Reptile: representative tiling for short read error correction. *Bioinformatics* 26, 2526–2533.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 18, 821–829. Available at: <https://doi.org/10.1101/gr.074492.107>.

Exome Sequencing Data Analysis

Anita Sathyanarayanan, Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD, Australia.

Srikanth Manda and Mukta Poojary, Institute of Bioinformatics, Bangalore, India

Shivashankar H Nagaraj, Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QD, Australia and Translational Research Institute, Brisbane, QLD, Australia

© 2019 Elsevier Inc. All rights reserved.

Abbreviations

BAM	Binary Alignment/Map	SAM	Sequence Alignment/Map
BWT	Burrows Wheeler Transform	SNP	Single Nucleotide Polymorphism
CNV	Copy Number Variation	SNV	Single Nucleotide Variation
dbSNP	Single Nucleotide Polymorphism database	SRA	Sequence Read Archive
ExAC	Exome Aggregation Consortium	SV	Structural Variation
GATK	Genome Analysis ToolKit	VCF	Variant Call Format
Indel	Insertion or Deletion	WGS	Whole Genome Sequencing
OMIM	Online Mendelian Inheritance in Man	WES	Whole Exome Sequencing

Glossary

Alignment The process of mapping reads to a reference genome.

Complex diseases Otherwise known as multifactorial disorders. Contrast to Mendelian diseases, these diseases are caused due to multiple genes or mutations as well as environmental risk factors.

Germline variants Variants or mutations present in the germ cells.

Mendelian diseases Disorders caused due to mutation at a single locus and are inherited according to Mendel's law.

Reads The short sequence of nucleotide bases from a single molecule of DNA detected by the sequencers.

Somatic variants Variants exclusive to somatic cells and not present in germ cells.

Variant calling Variant calling refers to identification of bases in the sequencing reads which are different from the reference genome at that position.

Whole exome sequencing Sequencing only the protein coding region of an organism's genome.

Whole genome sequencing Sequencing the complete genome to identify the full DNA sequence of an organism.

Introduction

The human genome is composed of three billion base pairs, comprising both protein coding and non-coding regions. The advancement in sequencing techniques from traditional Sanger sequencing to massively parallel high-throughput methods has made genome sequencing faster, cheaper, and more informative, allowing for a more comprehensive understanding of human genetics. Genome sequencing has become a vital clinical tool for the identification of the genetic basis of diseases, facilitating the management and treatment of many previously intractable conditions. There are two primary methods used to identify particular allelic variants in a given individual: Whole Genome Sequencing (WGS) and Whole Exome Sequencing (WES). WGS involves the sequencing of the entire genome, while WES focuses specifically on sequencing only the coding (exonic) regions of the genome in a more targeted approach. Both methods provide an unbiased means of variant identification to identify variants making them invaluable tools for precise personalized medicine.

The human exome comprises the entirety of the protein coding portion of the genome, which is just 1%–2% of the entire genome. The term exon derives from EXpressed regiON, i.e., the regions that are translated into protein products. These are generally the best studied and understood regions of the genome, and any mutation therein has the potential to disrupt normal protein function and to thereby cause disease. Given its more focused approach to genomic characterization, exome sequencing has emerged as a powerful tool in the recent years and has been utilized successfully to identify several severe and rare diseases associated with specific genetic variants in protein coding regions. The list of such diseases includes Miller Syndrome (Ng *et al.*, 2010b), Schinzel-Giedion syndrome (Hoischen *et al.*, 2010), Nonsyndromic hearing loss (Walsh *et al.*, 2010), Perrault syndrome (Pierce *et al.*, 2010), Sensenbrenner syndrome (Gilissen *et al.*, 2010), Kabuki syndrome (Ng *et al.*, 2010a), Progeroid syndrome (Puente *et al.*, 2011) and many others, each of which is caused by distinct mutations in specific proteins with serious physiological consequences. Exome sequencing thus provides a revolutionary means of characterizing the genomic landscapes of individuals with single base resolution, facilitating the identification of disease-associated mutations that allow for a more in-depth understanding of the root causes of a given disease, ultimately facilitating its treatment and management.

Background

There are two unbiased sequencing-based approaches for detecting genetic variations in an individual: Whole Genome Sequencing (WGS) and Whole Exome Sequencing (WES). While the human genome contains 3 billion bases, the human exome contains only 30 million bases (30 Mb). Despite phenomenal improvements in the accuracy and economic viability of genomic sequencing technologies, sequencing whole human genomes to a depth sufficient to identify novel variants associated with a given disease phenotype remains prohibitively resource-intensive both in terms of monetary and computational costs. WGS approaches are thus primarily employed for variant identification and related discoveries only in large genome sequencing laboratories and companies where such an approach is financially viable.

Advantages of Whole Exome Sequencing

WES technologies, in contrast, have helped to significantly reduce the time and resources required for actionable data – 20 human exomes can be sequenced in the same time required to sequence a single human genome. WES analysis significantly reduces the computational burden relative to a WGS analysis owing to the 99% reduction in the size of the sequenced regions within a given individual. Protein coding regions are better conserved between individuals (and across species) relative to intronic and non-coding regions, as the vital role of proteins in all aspects of human physiology renders these regions significantly more sensitive to the potentially deleterious effects of mutations. Indeed, more than 85% of disease-causing mutations in Mendelian diseases are located in coding regions of the genome. Given this enrichment for disease-associated mutations within the exome, WES provides an unbiased yet targeted approach to detect these variants, facilitating personalized and precision medicine at a fraction of the cost of WGS.

Exome Sequencing Analysis Workflow

The complete exome sequencing workflow begins with the extraction of DNA from a donor or patient of interest, and ends with the biological interpretation of identified variants through both experimental and computational approaches. The initial wet lab procedures involved in exome sequencing include genomic DNA extraction, DNA fragmentation, adaptor ligation, targeted exome capture, enrichment, and finally sequencing. The subsequent computational protocol involves the processing of raw sequencing reads to obtain a Variant Call Format (VCF) file that contains the list of identified variants within the given individual's exome. Intermediary steps include read quality control, alignment of sequenced data to a reference genome, post alignment clean-up (such as removal of duplicate reads), indel realignment, and base quality score recalibration, followed by variant calling, filtering, and annotation of the identified variants. Several in-depth guides regarding library preparation and sequencing are available in various published articles, and are an ideal resource for individuals performing WES (Parla *et al.*, 2011; Teer and Mullikin, 2010; Zhang *et al.*, 2015).

To begin a whole exome sequencing-based experiment, following the development of a careful experimental design which takes into account optimal patient/donor numbers and means of enriching for individuals with novel disease-associated variants, samples (blood, cheek swabs, etc.) must be collected from relevant individuals. After DNA is extracted from these samples, it is subjected to random fragmentation either via ultrasonication or with the help of transposons. Ensuring random fragmentation is important to avoid introducing sequencing artifacts. Artificial DNA linkers are added to the ends of all fragments to generate a shotgun library. Fragments containing coding sequences are then captured and enriched by commercially available hybridization-based capture kits. This target enrichment can be performed using in-solution or array-based capture techniques. For in-solution enrichment, a variety of probe definitions are available from different commercial companies such as Agilent, Illumina and NimbleGen. These enrichment strategies primarily differ in their specific genomic target regions covered, size, number of probes, and in their workflow design (see Table 1). The fragments of interest are hybridized to biotinylated DNA or RNA baits (depending on the exome capture kit used). The hybridized fragments are later recovered and enriched via affinity-based pull down, and are amplified prior to sequencing in order to generate a library of only captured exons.

Sequencing of the enriched fragments can be performed using a range of different sequencer platforms commercially available from Illumina, Life Technologies, or PacBio's SMRT. The information regarding the base identified at a particular position is stored in a raw data output file, the format of which varies based on the sequencer chosen. Regardless of the utilized sequencer, these files can be processed and analyzed using the data analysis pipelines tailored to answer experimental questions of interest. The steps of data analysis, starting from raw sequencer file outputs, are described further below.

Exome Sequencing Analysis

Quality Control/Pre-Processing

Sequencer outputs are known as nucleotide base calls, and are stored in the form of short nucleotide sequences called reads in file formats that vary based on the sequencing platform chosen. Generally accepted formats for read files are FASTQ and SRA (Sequence Read Archive). A number of different criteria can be used to assess read quality, including by measuring average read quality scores, GC content, presence of contaminating sequences and/or adaptor sequences, overrepresented k-mers, duplicated reads, and PCR artifacts.

Table 1 Exome capture platforms for human exome sequencing

	<i>NimbleGen's SeqCap EZ exome v3.0 kit</i>	<i>Agilent's sure select human all Exon V6 kit</i>	<i>Illumina's TruSeq exome prep kit</i>	<i>Illumina's Nextera rapid capture exome and expanded kit</i>
Probe size (bp)	55–105	114–126	95	95
Probe type	Biotinylated DNA	Biotinylated cRNA	Biotinylated DNA	Biotinylated DNA
Coverage strategy	High-density, overlapping probes	Adjacent probes	Gaps between probes	Gaps between probes
Fragmentation method	Ultrasonication	Ultrasonication	Ultrasonication	Transposons
Target region size (Mb, human)	64	60	45	62
Genomic DNA input required	1 ug	100 ng	100 ng	50 ng
Hybridization time (hours)	72	16	16	16
Reads remaining after filtering	66%	71.7%	54.8%	40.1%
Major strengths	● High sensitivity and specificity ● Most uniform coverage in difficult regions	● Better coverage of indels ● High alignment rate ● Fewer duplicate reads than other platforms	● Good coverage of UTRs and miRNAs	● Good coverage of UTRs and miRNAs
Major weaknesses	● More duplicate reads than Agilent ● Lower alignment rate than Agilent	● Fewer high-quality reads than NimbleGen	● High off-target enrichment	● High off-target enrichment ● Coverage bias for high GC content areas reducing uniformity
Human support	Not required	Not required	Required	Required

The quality of a given base call is measured as a Phred quality score and indicates the probability of the base being called correctly. The scores generally range from 2 to 40 with higher scores indicating greater confidence in the call. A common practice is to filter out bases with Phred scores below 20, although individual preferences may lead some researchers to modulate this threshold. The removal of redundant reads and undesired sequences (such as 3' adaptors, contaminants, or primers) is essential as contaminating sequences will not align to the genome resulting in poor alignment quality. Trimming low quality ends from reads is important as current sequencers are highly error-prone, with a significant dip in base quality at the 3' ends of reads. With time it is likely that such technical issues will be resolved, allowing for even more cost-effective whole exome sequencing.

Other pre-processing alternatives include assessing the read GC content. A uniform GC content across reads signifies good sequencing without artifacts or contaminants. Contaminating sequences are identifiable by checking if the frequency of a k -mer (a short sequence of bases of length k) is unusually high relative to levels that would be predicted due to chance. Sequences matching known adaptor sequences should be trimmed before aligning the reads to a reference genome, as mentioned above.

It is necessary to carefully examine the quality of raw reads prior to alignment in order to obtain a high percentage of properly aligned reads. FastQC ([Andrews, 2010](#)) is a commonly used application that can examine the quality of read sequences. It reports several parameters such as sequence quality scores, GC content, length distribution, and duplication levels. Cutadapt ([Martin, 2011](#)) and Trimmomatic ([Bolger et al., 2014](#)) are readily available programs that allow for the detection and trimming of adaptor sequences in reads, facilitating downstream alignment and analysis.

Reference-Based Alignment

The next step following quality control is alignment of reads to a reference genome. For human samples, this process involves the matching of short sequences of bases to a published reference genome using an alignment program. This is a complex and computationally intensive task, as the software ultimately matches millions of short sequences to 3 billion possible positions within the reference genome. Successful alignment requires the utilization of an effective alignment algorithm that is both efficient at searching for perfect base matches as well as tolerant of imperfect matches. This latter consideration is particularly important, as the goal of whole exome sequencing is to identify novel single nucleotide polymorphisms (SNPs), which by definition will be distinct from those incorporated into the human reference genome. Such considerations also allow the algorithm to identify insertions or deletions within the genome that could also have physiological consequences. Computationally the algorithm must be fast, memory efficient, and strike an effective balance between accuracy and speed. These tools must also be capable of handling single end and paired end reads, as paired end reads are the most common approach to sequencing data generation. Based on the computational approach used for quick and memory efficient string matching, aligners can be broadly classified into hash-based method or Burrows-Wheeler Transform (BWT) method ([Table 2](#)). Hash-based methods create hash table either for the read sequences or for the reference genome and use them to identify the matches. Aligners such as SOAP ([Li et al., 2008b](#)), MAQ

Table 2 Commonly used alignment algorithms

Aligner	Description
Borrows-Wheeler Aligner (BWA)	Gapped alignment based on Borrows-Wheeler Transform. Consists of three algorithms: BWA-backtrack, BWA-SW, and BWA-MEM. BWA-MEM is the most recent and efficient algorithm among the three.
Bowtie (1 and 2)	Both Bowtie 1 and 2 are ultra-fast memory efficient aligners based on Borrows-Wheeler Transform. Bowtie 1 is short read (≤ 50 bp) and ungapped aligner while Bowtie 2 allows gaps and has no restriction on read length.
SOAP (1,2 and 3)	While SOAP 1 is hash-based method, SOAP 2 and 3 are based on Borrows-Wheeler Transform. SOAP 2 is more memory efficient and utilizes lesser time than SOAP 1. SOAP 3 is the recent version which uses GPU processors.
MAQ	It is an ungapped short read aligner. It creates hash indexes of reads and scans the reference against the hashes to detect matches.
ELAND	Developed by Illumina company, it one of the first short read aligners. It performs ungapped alignment for reads of size 32bp. However, additional Perl scripts are provided to overcome the limitations. The source code is freely available for the machine buyers.
SHRiMP (1 and 2)	Performs Smith-Waterman local alignment algorithm for both letter space and colour space reads. While SHRiMP 1 indexes the reads, SHRiMP2 indexes the genome and is much faster. SHRiMP 2 also allows paired-end reads alignment and utilizes multi-threaded computation.
Novoalign	A commercial tool from Novocraft. It performs alignment by hashing the reference genome and is also capable of adaptor trimming and base quality trimming. It allows gaps and mismatches up to 50% of read length.

(Li *et al.*, 2008a), ELAND, and SHRiMP (Rumble *et al.*, 2009) are hash-based. BWT-based methods create a data structure called FM index for the genome sequence using data transformed using BWT and suffix array of the sequence. Popular aligners based on this approach are BWA (Li and Durbin, 2009), Bowtie2 (Langmead and Salzberg, 2012), and SOAP3 (Liu *et al.*, 2012).

Once generated, alignment results containing the read sequences and their respective mapped locations on the reference genome can be stored in the human readable format known as Sequence Alignment/Map (SAM) or in the Binary Alignment/Map (BAM) binary format (Li *et al.*, 2009). These alignment results then undergo further quality control, assessed based on a number of parameters including percentage of aligned reads, mapping quality, alignment scores, number of mismatches, and percentage of uniquely aligned reads. Successful mapping of 70%–90% of reads to the reference genome indicates good overall sequencing accuracy and absence of substantial amounts of contamination, thereby allowing for proper downstream analyses.

Several groups have reported on the comparative advantages of distinct aligners (Li and Homer, 2010; Fonseca *et al.*, 2012; Flicek and Birney, 2009). These tools differ based on their sensitivity, speed, and indexing approach. Choosing the right aligner and optimal parameters based on the sample is vital for high quality alignments, as errors could be carried downstream to variant calling. However, depending on the specific experimental application being employed there may be more than one effective alignment algorithm available. For this reason it is important that individual investigators discuss the advantages and disadvantages of each aligner prior to choosing the one which they ultimately utilize.

Post Alignment Processing

To ensure the quality of identified genetic variants in sequenced and aligned samples, several optional post alignment processing steps can be performed. These include removal of PCR and other duplicates, local alignment around indels, and recalibration of base quality scores.

Removal of Duplicates

The initial steps of library preparation involve the shearing of DNA, ligation of adaptors to both ends of the sheared fragments, and amplification of these fragments via PCR. PCR amplification is performed in order to obtain enough target DNA for sequencing with good depth and coverage. This approach can, however, selectively amplify certain DNA fragments leading to PCR duplicates and an overall misrepresentation of the initial genetic sample. Such a biased selection can lead to a disproportionate amplification of a single allele while ignoring the other, resulting in incorrect allele identification. As a consequence it is necessary to remove these PCR duplicates before variant calling in order to avoid any bias in calculation of variant frequency. While polymerases used in modern PCR reactions are of high fidelity, there is inevitably a chance that any PCR reaction may also introduce novel mutation errors during amplification. These PCR artifacts are difficult to distinguish from true DNA content solely on the basis of sequence and alignment information, as it can be difficult to tell a novel variant apart from an *in vitro* PCR-induced mutation.

Programs such as SAMtools (rmdup) (Li *et al.*, 2009) and Picard Tools (MarkDuplicates) can be used to remove the duplicates *in silico* so that they do not adversely affect sequence analyses or variant identification. These programs identify duplicates using information on the orientation and exact start site location on the forward and reverse read at their 5' and 3' positions respectively to properly establish whether a read is a duplicate or not. While SAMtools retains only the read with the highest mapping quality score, discarding their duplicates, Picard Tools simply sets a different flag for the highest scoring read pair and does not discard any

reads from the file (Ebbert *et al.*, 2016; Li *et al.*, 2009). As with other software packages, experimental need and investigator preferences will inevitably drive appropriate software selection.

Local Realignment (Indel Realignment)

Sequencing reads are mapped to the reference genome independently of one another during alignment. As a result, there are a large number of erroneous alignments that typically accumulate around the locations of indels (insertions/deletions) or low complexity regions. The next step is to locate genomic regions containing indels and to improve their alignment quality. Indel alignments to the reference genome create gaps due to their novelty, and as a consequence of poor alignment they are prone to noise. Local realignment is one solution that can help improve the accuracy of these reads. In this procedure, the reads around the indels are realigned based on the alignment information of other reads present in the location or by local *de novo* assembly of all reads spanning the indel region along with construction of a consensus sequence for indel discovery. This provides a better chance to accurately align indels than does aligning them solely based on a reference genome, which will not be able to account for these novel sequences. Although this is an optional procedure, it is still advisable to perform realignment to obtain high quality variants, given that these are the desired outcome of most whole exome sequencing studies. Programs that perform indel realignment include IndelRealigner (GATK) (Mckenna *et al.*, 2010), Dindel (Albers *et al.*, 2011), and Novoalign.

Base Quality Score Recalibration (BQSR)

Each base called by the sequencing platform is assigned a Phred score which indicates the confidence of the base call to be error-free, as discussed above. This automated system, however, generates scores that are error-prone as they can be over- or underestimated due to systematic errors. Although imperfect, Phred scores remain an important criteria for accurate variant calling, as bases with a low score still have a high probability of being called incorrectly. As variant callers depend greatly on the confidence of the base call, it is necessary to reconfirm the accuracy of the quality scores. BQSR is a process that applies machine learning to empirically model the errors and correct them accordingly, thereby markedly improving the accuracy of quality scores and boosting the quality of subsequent variant identification.

During recalibration, a covariation model based on the data and a set of known variants is constructed. Feeding in information on known variants will prevent the calling of real variant bases as base call errors. This covariation model is built based on quality score, position within the reads, and neighbouring nucleotides. Second, the base quality of the remaining mismatched bases are corrected based on this newly built model. It is common practice for investigators to generate before and after BQSR plots in order to visualise changes in base scoring for quality control purposes. Some commonly used tools are Recab (NGSUtils suite) (Breese and Liu, 2013), BaseRecalibrator (GATK) (Mckenna *et al.*, 2010), and ReQON (Bioconductor package) (Cabanski *et al.*, 2012).

Variant Analysis

A variant can be defined as a base in an individual's genome that is not the same as that in the reference genome. Variant calling refers to the identification of these mutated bases in the aligned reads, highlighting points that differ from the reference genome. To be called as a variant the base must have a good quality score and be well supported by multiple reads, thus minimizing the chance that the read is the result of poor alignment or other technical issues. Genomic variants or mutations can be of different types such as Single Nucleotide Variants (SNVs), insertion and deletion variants (indels), Copy Number Variants (CNVs), and large Structural Variants (SVs). The origin of these mutations may be either germline or somatic in nature. Germline variants, present in germ cells, are inherited and often related to familial diseases. Somatic mutations are often associated with tumour development, and are not heritable. The output of variant calling programs is a file containing several records, each of which represents a specific variant. Variant analysis consists of three major steps: Variant calling, annotation of variants, and prioritization of variants of interest depending on the constraints of a given study.

Variant calling

Variant calling refers to identification of bases in the sequencing reads which are different from the reference genome at that position. Variant calling can be performed using either single samples or multiple samples simultaneously depending on the experimental design and availability of computational resources/time constraints.

Germline variant calling

Germline variants correspond to the SNVs and indels in germ line cells. Briefly, germline variant calling tools compare each position of the sample genome to the reference genome and mathematically model the allele counts to obtain the genotype likelihood measures. Programs for identifying germline variants include UnifiedGenotyper (GATK), Haplotype caller (GATK) (Depristo *et al.*, 2011), FreeBayes (Garrison and Marth, 2012), SAMTools (Li *et al.*, 2009), Atals2 (Evani *et al.*, 2012), and VarScan 2 (Koboldt *et al.*, 2012).

Somatic variant calling

Somatic variants refer to the acquired mutations in somatic cells and are not transmitted to the progeny. As somatic variants play an important role in tumour initiation, development, and migration, somatic variant calling is widely used in cancer research and personalized medicine. It is highly recommended to have samples of both tumour and normal tissue/cells from a given patient in order to precisely identify only somatic variants while discarding the germline variants present in a given individual. Somatic variant calling programs detect variants using joint diploid genotype likelihoods or shared allele frequency between samples. Tools for somatic variants identification include MuTect (Cibulskis *et al.*, 2013), Strelka (Saunders *et al.*, 2012), deepSNV (Beerenwinkel *et al.*, 2015), and VarScan 2 (Koboldt *et al.*, 2012).

Variant annotation

Once variants are identified, they are annotated with attributes such as associated genomic features including tissue-specific expression, transcription factor binding sites, eQTLs, gene symbol, exonic function, and the specific amino acid change in question to facilitate the interpretation of their potential functional impact. Several tools are available for annotation of variant positions using information compiled from publicly available databases. They can also be used to calculate minor allele frequency (MAF) in normal populations, mine experimental evidence from clinical assays, predict potential deleterious variant functions, and analyze the frequency of the variant in oncogenic genes. Tools for annotating the variants are ANNOVAR (Wang *et al.*, 2010), MuTect (GATK), SnpEff (Cingolani *et al.*, 2012b), SnpSift (Cingolani *et al.*, 2012a), and VAT (Wang *et al.*, 2014), with ANNOVAR and MuTect being the most commonly used tools.

Variant filtration and prioritization

Typically a variant calling analysis will produce a list of thousands or millions of variants. This means that in addition to annotation, it is necessary to filter this list to obtain high confidence target variants that contribute to the disease or biological process under study. Variants are selected based on population frequency, coverage, supporting reads, and alignment score. They can also be prioritized according to their predicted coding effect. Mutations that are likely to cause a loss of protein function such as nonsense mutations in conserved regions, splice mutations that lead to frameshift, and mutations that alter protein structure are often disease-causing variants. Disease-causing mutations can be identified based on experimental evidence from previous studies or by observing specific mutational patterns present in the pedigree of affected and unaffected individuals. Some tools available for prioritization include VAAST2 (Hu *et al.*, 2013), VarSifter (Teer *et al.*, 2011), KGGseq (Li *et al.*, 2012), gNOME, PLINK/SEQ, and Ingenuity Variant Analysis.

Complete Automated Pipelines

A major challenge for most laboratories after sequencing is the management of data in a structured, systematic manner for targeted computational analysis. The WES analysis pipeline requires the use of many different tools for different analysis steps, and its complex parameters need to be optimized for every run. As a consequence, advanced skills and a complete understanding of the tools available, their usage, as well as their relative advantages and disadvantages, is required. Data manipulation and interpretation of the results after each step in the processing pipeline is a potentially overwhelming task for inexperienced users.

The complexity of exome sequencing data analysis can be simplified by software suites which provide all tools required for each step of this processing pipeline in a unified package. Several WES data analysis pipelines are available in the public domain. SIMPLEX (Fischer *et al.*, 2012) is a cloud-based pipeline for single-end (SE) and paired-end (PE) exome sequencing data analysis available from Illumina and ABI SOLiD. SeqMule (Guo *et al.*, 2015) is a Linux-based, user-friendly pipeline for analyzing exome and genome sequencing data. It integrates five alignment and five variant calling algorithms to provide comprehensive analyses. It can be used for identifying both somatic and germ line variants and allows for comparison of results from the different included algorithms. This platform is very flexible, as it allows users to modify the configuration file in order to fine tune its parameters. Fastq2vcf (Gao *et al.*, 2015) is another open-source pipeline that incorporates several open-source tools that allow users to perform analyses from quality control of reads to variants annotation. Galaxy (Afgan *et al.*, 2016), an open source online platform, has a wide collection of tools integrated. Customizable WES pipelines can be created and executed online. Many commercial tools are available with automated WES analysis pipelines, such as CLC Genomics Workbench 9.5.3 by Qiagen, STRAND NGS by Strand Life Sciences Pvt. Ltd which can be installed locally and their modules can be updated by connecting to their servers as per user requirements. Ingenuity Variant Analysis software Ingenuity Systems provides analysis toolkit which directly connects to its database for analysis. Other commercial software programs include Partek and DNA nexus.

Repositories for Variant Storage and Annotation

The critical step in linking mutations to disease is the identification of rare pathogenic variants which are responsible for a disease of interest. This is generally achieved by filtering out known variants present in normal healthy individuals. Such filtering is done with the help of repositories which maintain databases of such variants. Some of the widely used repositories for such common variants are the Single Nucleotide Polymorphism database (dbSNP) (Sherry *et al.*, 2001) and the Exome Aggregation Consortium (ExAC) (Lek *et al.*, 2016). Also of note, the Online Mendelian Inheritance in Man (OMIM) (Hamosh *et al.*, 2005) is the largest repository of known pathogenic variants reported for human.

dbSNP database

dbSNP is the largest repository for simple single nucleotide variants, containing both disease causing mutations and neutral polymorphisms. The Reference SNP (refSNP) Cluster Report on each variant contains the location, sequence, alleles, experimental conditions, population frequency, and variant view. Links to several external databases such as HGBASE, TSC (The SNP Consortium, Ltd), variation initiative, and NCI CGAP-GAI are also available. The ANNOVAR tool can be used to annotate the list of variants with dbSNP and filter out the neutral variants and those with minor allele frequencies greater than 1% to obtain a comprehensive list of target variants.

ExAC database

A key challenge when identifying actionable mutations is that every individual carries several thousand mutations which may not be responsible for disease and are generally seen in a subset of healthy individuals. The ExAC database maintains the largest publicly available repository of human exomes representing populations around the world. Exomes of 60,706 individuals have been deeply sequenced and mined for variants in a uniform fashion. The ExAC browser can be used to mine actionable mutations by eliminating commonly occurring background mutations in the variant dataset.

OMIM database

OMIM is a continuously updated repository of human genes and their associated phenotypes, with a focus on the molecular relationships between genetic variation and phenotypic expression. The information available at OMIM is sourced from literature and is used to identify phenotypes associated with specific human genes. The data uses Mendelian Inheritance in Man (MIM) identifiers and provides concise information on a given gene's location, structure, function, available animal models, and relevant references.

Applications of Exome Sequencing**Discovery of Rare Variants in Mendelian Diseases and Complex Disorders**

Exome sequencing has emerged as one of the most efficient strategies for deciphering the genetic causes of both simple Mendelian diseases and complex disorders. It has helped to link several diseases with their causal genes or variants, providing a cost-effective and high-throughput approach to relevant variant discovery. This approach can thus lead to deeper understanding of the mechanisms of disease, leading to improved diagnosis, and the development of superior preventive strategies or targeted therapeutics.

To identify causal variants, two types of exome sequencing sampling strategies are used: family-based sequencing and extreme phenotype sequencing. When multiple members of the same family are affected with a trait, then exome sequencing is performed for the most distantly related individuals. Distant relatives share fewer common genetic variants and hence allow identification of pathogenic variants with a greater amount of confidence. Sequencing a parent-offspring trio in which only the offspring is affected can also be conducted to identify variants responsible for complex diseases. The extreme phenotype sampling strategy involves sequencing the extreme phenotype samples of a trait i.e., samples on the either ends of a phenotype distribution (such as individuals with the most severe forms of a disease and normal controls). Alleles contributing to the extreme phenotypes are enriched, facilitating identification of causal variants.

Mendelian disorders arise from single mutations or a small number of mutations in select causal genes. Many studies have shown that exome sequencing can be successfully applied to discover causal genes in rare Mendelian disorders. The first such successful application of exome sequencing was the identification of the genetic root of the rare Mendelian Miller syndrome (Ng *et al.*, 2010b). Exome sequencing of four affected individuals was performed, and the mutations on the responsible gene, DHODH, were discovered after filtering the exonic variants of the affected individuals against dbSNP and eight HapMap exomes.

Complex disorders are thought to be linked with single or multiple genes and/or a confluence of genetic and environmental risk factors. This results in a heterogeneity of both phenotypes and of the genes and factors involved, necessitating additional scrutiny when performing WES. Exome sequencing of large numbers of patient samples still makes such variant identification possible owing to its reduced costs relative to WGS, and its concomitant reduced analysis burdens. A successful application of WES for a complex phenotype was employed in the case of peripheral neuropathy in two male siblings born to healthy parents exhibiting this disease phenotype. Upon WES analysis, the causal mutation was revealed to be the deletion or substitution of arginine in the 1070th position on the PRX gene. This mutation is known to cause Dejerine-Sottas syndrome, or peripheral neuropathy (Williams *et al.*, 2016). WES has also identified a number of causal genes for complex traits such as mental retardation (Vissers *et al.*, 2010) and sporadic autism spectrum disorders (O'Roak *et al.*, 2011).

Cancer is a heterogeneous disease with each tumour carrying a range of distinct somatic mutations in a number of different oncogenes and tumour suppressor genes including SNV's, short indels, copy number variants, and gene translocations. This makes it a challenge to identify causal genes in cancer. The success of the Human Genome Project bolstered interest in understanding cancer exomes. The first cancer exome study analyzed 140 human acute myeloid leukemia (AML) WES samples and identified six known and seven novel mutations relevant for AML pathogenesis (Ley *et al.*, 2003). Another early cancer exome study cataloged the landscape of genetic alterations that occur during tumorigenesis by studying the exomes of 11 colorectal and 11 breast tumors. They observed few recurrently mutated genes and several rarely mutated genes (Wood *et al.*, 2007). This demonstrates the powerful

capabilities of WES-based approaches in the context of complex disease, enabling the identification of potential novel driver mutations of cancer and underscoring the clinical utility of this tool.

Clinical Diagnosis

The success of WES in identifying genetic causes of disease has paved the way for its application as a diagnostic tool utilized for unbiased personalized medicine. Currently in a clinical setting WES is being used as a diagnostic tool for identifying casual genes and variants in patients afflicted with diseases that are suspected to be Mendelian in nature. It can also be used to identify dormant risk variants in patients even when symptoms are unobservable, particularly in patients with a family history of a given genetic disease. WES combined with pedigree information, through literature and relevant database searches, has been critical for diagnosis and variant or casual gene discovery in cases related to congenital heart disease, autism, epilepsy, brain malformation, neurodevelopmental deformities, and other neurological or congenital conditions.

Discussion Limitations of Exome Sequencing

As with every technology, exome sequencing also has limitations, and failure to identify disease-associated variants can occur for many reasons. One potential impediment to such analyses is the lack of knowledge of the full human transcriptome, due to an incomplete understanding of the exact locations of all protein coding regions. This makes comprehensively defining targets for WES challenging. Some of the technical reasons underlying potential assay failure include poor efficiency of capture probes, insufficient sequencing efficacy, and poor selection of target for sequencing. Analytical challenges include inadequate coverage at the casual variant location, identification of variants in low complexity regions, false positives due to misaligned reads, and poor separation of casual and non-casual alleles. Although the majority of known diseases are caused by mutations in the protein coding regions of genes, intronic regions are important and can also contribute to disease in certain instances. WES by definition precludes the possibility of identifying structural variants that arise in intergenic or intronic regions or on mitochondrial chromosomes.

Despite these limitations, WES remains a vital tool in clinical diagnostics due to its relatively low monetary and time costs relative to other sequencing approaches. As sequencing costs continue to decrease, computational resources become more available, and algorithms become more efficient it will inevitably become standard to perform WGS to establish gene-phenotype relations. Until technological advances permit for this approach, WES is a powerful tool that captures much of the utility and functionality of whole genome sequencing.

Future Directions

Advancements pertaining to Sequencing technology and large scale data analysis and management will significantly push the limits of exome sequencing applications. Improvements in sequencing methodologies will lead to production of longer reads, detection of longer indels, fewer errors during base call, high quality sequences and reliable reference genomes. With the decreasing costs and turnaround time for sequencing, usage of reference genomes specific to geographical population, ethnicity, etc., will become feasible thereby eliminating the use of a single reference genome during variant discovery. Additionally, integrated approach of WES and WGS will be increasingly applied to complement the short comings of the individual methods. Apart from WES and WGS methods, NGS is applied to other omics of cell such as the transcriptome, epigenome, proteome, and metabolome. It has already been recognised that integration of multi-omics data is crucial to get a holistic understanding of the disease state and its systems biology and multi-omics data integration will slowly become a routine in the foreseeable clinical space. With advent of NGS, there has been an enormous influx of data, making omic data handling and management undoubtedly one of the major challenges. The increasing data has led to the creation of global repositories for easier access. Efforts are also being made to create repositories with reliable multidimensional annotation and curation of phenotype. Parallel advancements in the fields such as artificial intelligence/machine learning, high performance computing and large scale data analysis will play major roles in data mining and knowledge extraction of sequencing data. Accordingly, we can look forward to a future where the current challenge of identifying of new Mendelian disorders and their causative genes will be easily resolved due to the collective effort, immense technological developments and progressive interdisciplinary research.

Closing Remarks

The field of genetics has witnessed remarkable progress in the past decade owing to the advancement in sequencing technology and computer science. With decreasing costs and increasing application of sequencing, whole exome sequencing has become an irreplaceable tool in the clinical space, giving faster, cheaper, and reliable results. However, translating the acquired knowledge in the clinical space is one of the biggest challenges being faced. A collective effort is required from researchers all around the globe to solve all Mendelian disorders, decipher the phenotype-genotype relationship, understand the disease mechanism and its systems biology which will ultimately translate the knowledge into better diagnostics and therapeutics.

See also: Computational Pipelines and Workflows in Bioinformatics. Constructing Computational Pipelines. Genome Annotation: Perspective From Bacterial Genomes. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing

References

- Afgan, E., Baker, D., Van Den Beek, M., *et al.*, 2016. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research* 44, W3–W10.
- Albers, C.A., Lunter, G., MacArthur, D.G., *et al.*, 2011. Dindel: Accurate indel calls from short-read data. *Genome Research* 21, 961–973.
- Andrews, S. 2010. FastQC: A quality control tool for high throughput sequence data.
- Beerenwinkel, N., Jones, D., Martincorena, I., Gerstung, M. 2015. Package 'deepSNV'.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for illumina sequence data. *Bioinformatics* 30, 2114–2120.
- Breese, M.R., Liu, Y., 2013. NGSUtils: A software suite for analyzing and manipulating next-generation sequencing datasets. *Bioinformatics* 29, 494–496.
- Cabanski, C.R., Cavin, K., Bizon, C., *et al.*, 2012. ReQON: A bioconductor package for recalibrating quality scores from next-generation sequencing data. *BMC Bioinformatics* 13, 221.
- Cibulskis, K., Lawrence, M.S., Carter, S.L., *et al.*, 2013. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology* 31, 213–219.
- Cingolani, P., Patel, V.M., Coon, M., *et al.*, 2012a. Using *Drosophila melanogaster* as a model for genotoxic chemical mutational studies with a new program, SnpSift. *Frontiers in Genetics* 3.
- Cingolani, P., Platts, A., Wang, L.L., *et al.*, 2012b. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly* 6, 80–92.
- Depristo, M.A., Banks, E., Poplin, R., *et al.*, 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics* 43, 491–498.
- Evani, U.S., Challis, D., Yu, J., *et al.*, 2012. Atlas2 cloud: A framework for personal genome analysis in the cloud. *BMC Genomics* 13, S19.
- Ebbert, M.T., Wadsworth, M.E., Staley, L.A., *et al.*, 2016. Evaluating the necessity of PCR duplicate removal from next-generation sequencing data and a comparison of approaches. *BMC bioinformatics* 17 (7), 239.
- Fischer, M., Snajder, R., Pabinger, S., *et al.*, 2012. SIMPLEX: Cloud-enabled pipeline for the comprehensive analysis of exome sequencing data. *PLOS ONE* 7, e41948.
- Flicek, P., Birney, E., 2009. Sense from sequence reads: Methods for alignment and assembly. *Nature Methods* 6, S6–S12.
- Fonseca, N.A., Rung, J., Brazma, A., Marioni, J.C., 2012. Tools for mapping high-throughput sequencing data. *Bioinformatics* 28, 3169–3177.
- Gao, X., Xu, J., Starmer, J., 2015. Fastq2vcf: A concise and transparent pipeline for whole-exome sequencing data analyses. *BMC Research Notes* 8, 72.
- GARRISON, E., MARTIN, G., 2012. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*.
- Gilissen, C., Arts, H.H., Hoischen, A., *et al.*, 2010. Exome sequencing identifies WDR35 variants involved in Sensenbrenner syndrome. *The American Journal of Human Genetics* 87, 418–423.
- Guo, Y., Ding, X., Shen, Y., Lyon, G.J., Wang, K., 2015. SeqMule: Automated pipeline for analysis of human exome/genome sequencing data. *Scientific Reports* 5.
- Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A., McKusick, V.A., 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research* 33, D514–D517.
- Hoischen, A., Van Bon, B.W., Gilissen, C., *et al.*, 2010. De novo mutations of SETBP1 cause Schinzel-Giedion syndrome. *Nature Genetics* 42, 483.
- Hu, H., Huff, C.D., Moore, B., *et al.*, 2013. VAAST 2.0: Improved variant classification and disease-gene identification using a conservation-controlled amino acid substitution matrix. *Genetic Epidemiology* 37, 622–634.
- Koboldt, D.C., Zhang, Q., Larson, D.E., *et al.*, 2012. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research* 22, 568–576.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nature Methods* 9, 357–359.
- Lek, M., Karczewski, K.J., Minikel, E.V., *et al.*, 2016. Analysis of protein-coding genetic variation in 60,706 humans. *Nature* 536, 285–291.
- Ley, T.J., Minx, P.J., Walter, M.J., *et al.*, 2003. A pilot study of high-throughput, sequence-based mutational profiling of primary human acute myeloid leukemia cell genomes. *Proceedings of the National Academy of Sciences* 100, 14275–14280.
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25, 1754–1760.
- Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009. The sequence alignment/map format and SAMtools. *Bioinformatics* 25, 2078–2079.
- Li, H., Homer, N., 2010. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics* 11, 473–483.
- Li, H., Ruan, J., Durbin, R., 2008a. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18, 1851–1858.
- Li, M.-X., Gui, H.-S., Kwan, J.S., Bao, S.-Y., Sham, P.C., 2012. A comprehensive framework for prioritizing variants in exome sequencing studies of Mendelian diseases. *Nucleic Acids Research* 40, e53.
- Li, R., Li, Y., Kristiansen, K., Wang, J., 2008b. SOAP: Short oligonucleotide alignment program. *Bioinformatics* 24, 713–714.
- Liu, C.-M., Wong, T., Wu, E., *et al.*, 2012. SOAP3: Ultra-fast GPU-based parallel alignment tool for short reads. *Bioinformatics* 28, 878–879.
- Martin, M., 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet Journal* 17, 10–12.
- McKenna, A., Hanna, M., Banks, E., *et al.*, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20, 1297–1303.
- Ng, S.B., Bigham, A.W., Buckingham, K.J., *et al.*, 2010a. Exome sequencing identifies MLL2 mutations as a cause of Kabuki syndrome. *Nature Genetics* 42, 790–793.
- Ng, S.B., Buckingham, K.J., Lee, C., *et al.*, 2010b. Exome sequencing identifies the cause of a mendelian disorder. *Nature Genetics* 42, 30–35.
- O’Roak, B.J., Deriziotis, P., Lee, C., *et al.*, 2011. Exome sequencing in sporadic autism spectrum disorders identifies severe de novo mutations. *Nature Genetics* 43, 585–589.
- Parla, J.S., Iossifov, I., Grabill, I., *et al.*, 2011. A comparative analysis of exome capture. *Genome Biology* 12, R97.
- Pierce, S.B., Walsh, T., Chisholm, K.M., *et al.*, 2010. Mutations in the DBP-deficiency protein HSD17B4 cause ovarian dysgenesis, hearing loss, and ataxia of Perrault syndrome. *The American Journal of Human Genetics* 87, 282–288.
- Puente, X.S., Quesada, V., Osorio, F.G., *et al.*, 2011. Exome sequencing and functional analysis identifies BANF1 mutation as the cause of a hereditary progeroid syndrome. *The American Journal of Human Genetics* 88, 650–656.
- Rumble, S.M., Lacroute, P., Dalca, A.V., *et al.*, 2009. SHRIMP: Accurate mapping of short color-space reads. *PLOS Computational Biology* 5, e1000386.
- Saunders, C.T., Wong, W.S., Swamp, S., *et al.*, 2012. Strelka: Accurate somatic small-variant calling from sequenced tumor – Normal sample pairs. *Bioinformatics*, 28. . pp. 1811–1817.
- Sherry, S.T., Ward, M.-H., Kholodov, M., *et al.*, 2001. dbSNP: The NCBI database of genetic variation. *Nucleic Acids Research* 29, 308–311.
- Teer, J.K., Green, E.D., Mullikin, J.C., Biesecker, L.G., 2011. VarSifter: Visualizing and analyzing exome-scale sequence variation data on a desktop computer. *Bioinformatics* 28, 599–600.
- Teer, J.K., Mullikin, J.C., 2010. Exome sequencing: The sweet spot before whole genomes. *Human Molecular Genetics* 19, R145–R151.
- Vissers, L.E., De Ligt, J., Gilissen, C., *et al.*, 2010. A de novo paradigm for mental retardation. *Nature Genetics* 42, 1109–1112.

- Walsh, T., Shahin, H., Elkan-Miller, T., *et al.*, 2010. Whole exome sequencing and homozygosity mapping identify mutation in the cell polarity protein GPM2 as the cause of nonsyndromic hearing loss DFNB82. *The American Journal of Human Genetics* 87, 90–94.
- Wang, G.T., Peng, B., Leal, S.M., 2014. Variant association tools for quality control and analysis of large-scale sequence and genotyping array data. *The American Journal of Human Genetics* 94, 770–783.
- Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38, e164.
- Williams, H.J., Hurst, J.R., Ocaña, L., *et al.*, 2016. The use of whole-exome sequencing to disentangle complex phenotypes. *European Journal of Human Genetics* 24, 298.
- Wood, L.D., Parsons, D.W., Jones, S., *et al.*, 2007. The genomic landscapes of human breast and colorectal cancers. *Science* 318, 1108–1113.
- Zhang, G., Wang, J., Yang, J., *et al.*, 2015. Comparison and evaluation of two exome capture kits and sequencing platforms for variant calling. *BMC Genomics* 16, 581.

Further Reading

- Bamshad, M.J., Ng, S.B., Bigham, A.W., *et al.*, 2011. Exome sequencing as a tool for Mendelian disease gene discovery. *Nature Reviews Genetics* 12, 745–755.
- Biesecker, L.G., Green, R.C., 2014. Diagnostic clinical genome and exome sequencing. *New England Journal of Medicine* 370, 2418–2425.
- Clark, M.J., Chen, R., Lam, H.Y., *et al.*, 2011. Performance comparison of exome DNA sequencing technologies. *Nature biotechnology* 29, 908–914.
- Goodwin, S., McPherson, J.D., McCombie, W.R., 2016. Coming of age: Ten years of next-generation sequencing technologies. *Nature Reviews Genetics* 17, 333–351.
- Laurie, S., Fernandez-Callejo, M., Marco-Sola, S., *et al.*, 2016. From Wet-Lab to variations: Concordance and speed of bioinformatics pipelines for whole genome and whole exome sequencing. *Human Mutation* 37, 1263–1271.
- Petersen, B.-S., Fredrich, B., Hoepfner, M.P., Ellinghaus, D., Franke, A., 2017. Opportunities and challenges of whole-genome and -exome sequencing. *BMC Genetics* 18, 14.
- Rabbani, B., Tekin, M., Mahdieh, N., 2014. The promise of whole-exome sequencing in medical genetics. *Journal of Human Genetics* 59, 5–15.
- Tennissen, J.A., Bigham, A.W., O'Connor, T.D., *et al.*, 2012. Evolution and functional impact of rare coding variation from deep sequencing of human exomes. *Science* 337, 64–69.

Relevant Websites

- <http://annovar.openbioinformatics.org/en/latest/>
ANNOVAR.
- <https://www.hgsc.bcm.edu/software/atlas-2>
Atlas.
- <http://bio-bwa.sourceforge.net/>
Borrows-Wheeler Aligner.
- <http://bowtie-bio.sourceforge.net/index.shtml>
Bowtie.
- <https://www.qiagenbioinformatics.com/products/clc-genomics-workbench/>
CLC Genomics Workbench.
- <https://pypi.python.org/pypi/cutadapt>
Cutadapt.
- <https://www.ncbi.nlm.nih.gov/SNP/>
dbSNP.
- <https://bioconductor.org/packages/release/bioc/html/deepSNV.html>
deepSNV.
- <http://www.sanger.ac.uk/science/tools/dindel>
Dindel.
- <https://www.dnanexus.com/>
DNA Nexus.
- <http://exac.broadinstitute.org/>
ExAC.
- <https://sourceforge.net/projects/fastq2vcf/>
Fastq2vcf.
- <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
FastQC.
- <https://github.com/ekg/freebayes>
FreeBayes.
- https://software.broadinstitute.org/gatk/documentation/tooldocs/current/org_broadinstitute_gatk_tools_walkers_haplotypecaller_HaplotypeCaller.php
HaplotypeCaller.
- https://software.broadinstitute.org/gatk/documentation/tooldocs/current/org_broadinstitute_gatk_tools_walkers_indels_IndelRealigner.php
IndelRealigner.
- <https://www.qiagenbioinformatics.com/products/ingenuity-variant-analysis/>
Ingenuity Variant Analysis.
- <http://grass.cgs.hku.hk/limx/kggseq/>
KGGSeq.
- <http://maq.sourceforge.net/maq-man.shtml>
MAQ.
- https://software.broadinstitute.org/gatk/documentation/tooldocs/current/org_broadinstitute_gatk_tools_walkers_cancer_m2_MuTect2.php
MuTect.
- <http://www.novocraft.com/products/novoalign/>
NovoAlign.
- <https://www.omim.org/>
OMIM.
- <http://www.partek.com/pgs>
Partek Genomics Suite.

<http://broadinstitute.github.io/picard/>

Picard Tools.

<http://atgu.mgh.harvard.edu/plinkseq/download.shtml>

Plink/Seq.

<http://samtools.sourceforge.net/>

SamTools.

<http://wglab.org/software/14-segmule>

SeqMule.

<http://compbio.cs.toronto.edu/shrimp/>

SHRiMP.

<http://icbi.at/software/simplex/simplex.shtml>

Simplex.

<http://snpeff.sourceforge.net/>

SnEff.

<http://snpeff.sourceforge.net/SnpSift.html>

SnpSift.

<http://soap.genomics.org.cn/>

SOAP.

<http://www.strand-ngs.com/>

StrandNGS.

<https://github.com/Illumina/strelka>

Strelka.

<http://www.usadellab.org/cms/?page=trimmomatic>

Trimmomatic.

https://software.broadinstitute.org/gatk/documentation/tooldocs/current/org_broadinstitute_gatk_tools_walkers_genotyper_UnifiedGenotyper.php

UnifiedGenotyper.

<http://www.yandell-lab.org/software/vaast.html>

VAAST2.

<http://dkoboldt.github.io/varscan/>

VarScan.

<https://research.nhgri.nih.gov/software/VarSifter/>

VarSifter.

Biographical Sketch



Anita Sathyanarayanan is pursuing her PhD in Bioinformatics at Queensland University of Technology, Australia. Her current area of research is application of computational methods to multi-omics data to understand cancer development and progression. Her research interests centres around the intersection of machine learning and high-throughput omics.



Srikanth Manda has completed his Masters in Biotechnology from Indian Institute of Technology, Bombay, thereafter, pursued a PhD in Bioinformatics from Institute of Bioinformatics, Bangalore and Johns Hopkins University, USA. During his PhD work he has analyzed datasets from Whole Genome (WGS), Whole Exome (WES), Transcriptome, epigenome, miRNAome, proteome and phosphoproteome. His research interests involve multi-omics data analysis, integration and interpretation from high-throughput platforms using available tools and cloud based workflows.



Mukta Poojary is a graduate student at CSIR-Institute of Genomics and Integrative Biology (CSIR-IGIB), New Delhi. She is a recipient of Bioinformatics National Certification (BINC). She works in the area of Genome informatics. Her academic interests spans RNA biology, Population genomics, Pharmacogenomics and Multi-omic data analysis.



Shivashankar H Nagaraj is Advanced Queensland Research Fellow at Queensland University of Technology. The focus of his group is on the development and application of bioinformatics approaches that utilize large amounts of data to solve problems in Genomics and Proteomics. He has developed multiple Bioinformatics solutions (e.g., ESTExplorer, EST2Secretome, PGTools) that are valuable and of practical utility to the wider scientific community.

Whole Genome Sequencing Analysis

Rui Yin, Chee Keong Kwoh, and Jie Zheng, Nanyang Technological University, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

The development of DNA sequencing has revolutionized the biological sciences and facilitated the discovery of gene functions. The first widely used DNA sequencing method was Sanger sequencing (Sanger *et al.*, 1977; Slatko *et al.*, 2001) developed in the 1980s in a manual way. Rapid shifts in the techniques of DNA sequencing made it possible for automated sequencing in the 1990s, which allowed the sequencing of whole genomes (Heather and Chain, 2016). Whole genome sequencing (WGS) is a laboratory process that determines the entire DNA sequence of an organism's genome at once. It involves uncovering the order of bases in a complete genome of an organism, which is backed by automatic DNA sequencing methods and computer techniques to assemble the tremendous biological sequence data (Sarawathy and Ramalingam, 2011). Whole genome shotgun sequencing was already in use in 1979 for small genomes ranging from 4000 to 7000 base pairs (Staden, 1979). In 1995, the first genome of *Haemophilus influenzae* was sequenced (Fleischmann *et al.*, 1995). Later, the sequencing of nearly an entire human genome was completed in 2000 (Lander *et al.*, 2001). However, early techniques for WGS were slow, labor-intensive, and expensive, especially in the era of extensive growth of genomic data (Kwong *et al.*, 2015).

The advent and widespread use of next generation sequencing (NGS) techniques have greatly brought about the availability of whole genome analysis and significantly enhanced the capacity to perform efficient and low-cost WGS. Generally, NGS processes contain massively parallel sequencing, where millions of fragments of DNA from a single sample are sequenced (Moorthie *et al.*, 2011). At the cellular level, it has been applied to the resequencing of previous reference strains and allows for the identification of all the potential mutations in an organism at the genomic level (Schuster, 2008). The genes in a genome usually don't work independently and their functions are not directly controlled by the promoter but rather many other regulatory elements such as the response elements, enhancers, and silencers (Sarawathy and Ramalingam, 2011). Therefore, the implementation of WGS will not only present better understanding of the function of all the genes in the cell, but also characterize the related information of DNA that is involved in gene expression. In addition, compared with whole exome sequencing (WES), WGS provides much more comprehensive information on genes by sequencing the noncoding DNA, which captures 98%–99% of the genome; most of the information about the function of noncoding DNA is still a mystery.

The power of NGS has broadened the scale of many applications in current work, including genetic variation (Dalca and Brudno, 2010; Alkan *et al.*, 2011a,b), assembling new genomes or transcriptomes (Flück and Birney, 2009), quantitative RNA-sequencing analysis (Pepke *et al.*, 2009), epigenetic change identification (Meaburn and Schulz, 2012), etc. In this article, we only discuss the aspect of WGS application for processing genomic data related to genetic variation. The types of genetic variants mainly consist of single nucleotide polymorphisms (SNPs), indels (insertions or deletions), structural variants (duplications, deletions, and inversions), and copy number variants (CNVs) (Zafar *et al.*, 2016; Hillier *et al.*, 2008; Kidd *et al.*, 2008). The inherited variation raises a great deal of risk for individual health that may cause both common and rare disease (Andrew *et al.*, 2017). WGS renders multiple potential solutions by providing full-scale collection of genetic variants. Though the expense to conduct a large scale has decreased considerably, there is still much space for the reduction of the cost that could popularize the techniques and applications of WGS. Consequently, it is necessary to develop more advanced and efficient methodology and pipelines for processing and analyzing whole genomes.

Typical Computational Pipeline of Whole Womene Sequencing

The exponential growth of NGS data has promoted the blossom and release of tools for genomic analysis in recent years. Numerous different workflows are now available to deal with high-throughput genome sequencing data in the distinct types (Leipzig, 2017), for example, shell script, tool-specific interface, and graphical workflow environment (Wang *et al.*, 2009; Yoo *et al.*, 2011; Goecks *et al.*, 2010; Chard *et al.*, 2008). A typical computational pipeline for WGS usually contains components such as data preparation, alignment and assembly, variant calling, annotation, and analysis. Fig. 1 shows the detailed steps of the pipeline for WGS analysis.

Data Preparation

The first step in the pipeline of WGS is data preparation, more specifically, sample preparation. The samples of nucleotide vary greatly from source to quality, which requires professional kits that help to acquire the nucleic acid after purification processing. To obtain the sequences, the genomic materials need to be managed into a sample library that contains short fragments after purification (Van Dijk *et al.*, 2014). The sample libraries usually require high quality and intact sequences at a sufficient amount that meets the requirements for performing the sequencing by instruments (Wong *et al.*, 2012). The general procedures of library preparation consist of several steps. The first step is to split the source genomic material into required lengths and then conduct the end repair and ligation of oligonucleotide adapters to the ends of the fragments. Next is the enrichment of the adapter ligated

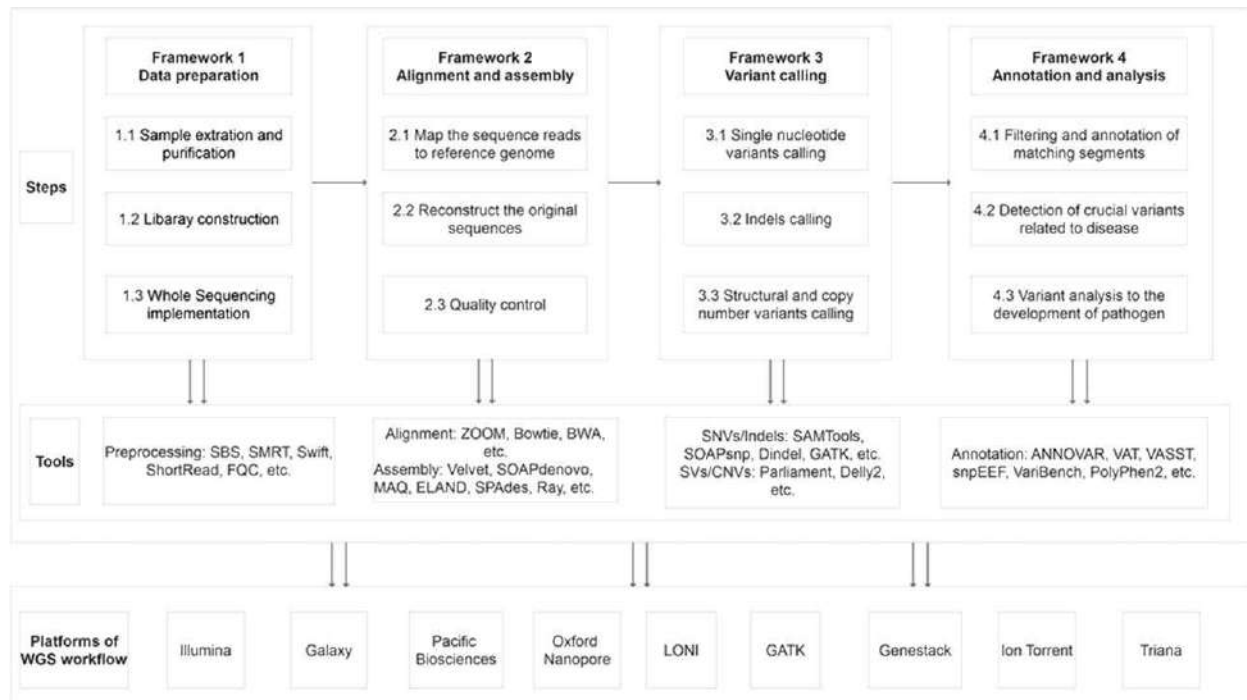


Fig. 1 A typical computational pipeline for whole genome sequencing.

library, and the final process is the validation and quantification of the results. Once the library construction is completed, large volumes of samples are sequenced in parallel by sequencing tools. The output of sequencing will be regulated in a standardized format such as FASTQ for alignment.

Alignment and Assembly

Once WGS data has been prepared, the alignment is set up that maps the reads of these samples to a reference genome. The length of reference genome could alter from thousands to billions of possible positions based on the species, so comparing plenty of short nucleotide reads to the reference genome is computationally intensive and time consuming (Kwong *et al.*, 2015). The implementation of alignment is usually using one or more specific algorithms aiming at the sequence reads characteristics. The hash-based (Flicek and Birney, 2009) and Burrows–Wheeler transform (BWT) (Li and Durbin, 2009) methods are the two most popular algorithms utilized as the core algorithm for many alignment software tools. The hash-based method builds either the hash table or input reads to scan the genome reference or in reverse. The software that uses hash table includes mrFAST, SHRIMP (Alkan *et al.*, 2011a,b; Rumble *et al.*, 2009). Another widely used method is the BWT, which creates a suffix array by performing BWT to obtain transformed sequences as a new reference genome for mapping. The BWT method achieves a faster algorithm compared with the hash-based method and is adopted by BOWTIE/BOWTIE2, SOAP2, BWA (Li *et al.*, 2009a,b; Langmead *et al.*, 2009). In addition to these software packages, many more tools can be found in Table 1.

Whole genome assembly follows the step of sequence alignment. The purpose of assembly is to reconstruct the original sequences into large contiguous segments that can be ordered and oriented along each chromosome (Flicek and Birney, 2009). De novo assembly and reference-based assembly are the two main approaches to form longer continuous sequences. De novo assembly involves utilizing computational methods to align overlapping WGS reads for assembling known reads as contigs and order the contigs into the framework of the sequenced genome (Robertson *et al.*, 2010). Referenced-based assembly maps each read to a reference genome sequence and builds a consensus sequence, which is similar but not necessarily identical to the backbone reference (Ng and Kirkness, 2010). Many tools are released online for assembling based on the two algorithms. De novo assemblers include two types: Overlap graphs (Myers, 1995) and de Bruijn graphs (Pevzner *et al.*, 2001). The overlap graph method is used by Canu (Myers *et al.*, 2000) and Arachne (Jaffe *et al.*, 2003), which calculates all the pairwise overlaps between reads and presents the information in the graph. The de Bruijn graph splits reads into k-mers to form the nodes of a network, where k describes the length in bases of these subsequences. Commonly used software based on this method includes Velvet (Zerbino and Birney, 2008), SOAPdenovo (Luo *et al.*, 2012), and ABySS (Simpson *et al.*, 2009). Compared with de novo assemblers, referenced-based assemblers require fewer computational resources but are more challenging in dealing with novel sequences. Examples of referenced-based assemblers include MAQ, SeqMap, and RMAP (Flicek and Birney, 2009; Jiang and Wong, 2008; Smith *et al.*, 2009). We have summarized some frequently used assemblers in Table 2.

Table 1 A list of software for alignment of short reads to the reference genome

Software	Simple description	Resources
MAQ	Build mapping assemblies from short reads generated by the NGS machines	http://maq.sourceforge.net/
ELAND	Hash-based computational alignment program for short oligonucleotides	https://www.illumina.com/
Bowtie/	A short-read aligner that aligns short DNA sequences (reads) to the human	http://bowtiebio.sourceforge.net/index.shtml
Bowtie2	genome at a rate of over 25 million 35-bp reads per hour	http://bowtiebio.sourceforge.net/bowtie2/index.shtml
SHRiMP	Sequence aligner developed with the multitudinous short reads of NGS	http://compbio.cs.toronto.edu/shrimp/
mrFAST	Seed-and-extend alignment tool to map short reads with the Illumina platform	http://mrfast.sourceforge.net/
SOAP	A GPU-based software for aligning short reads with a reference sequence	http://soap.genomics.org.cn/index.html
BWA	A software package that maps low-divergent sequences against a large reference genome	https://github.com/lh3/bwa
SEAL	A suite of distributed applications for aligning short DNA reads, manipulating and analyzing short-read alignments	http://biodoop-seal.sourceforge.net/
BLAST	A suite of programs that align query sequences in a selected target database	https://blast.ncbi.nlm.nih.gov/Blast.cgi

Table 2 Review of most commonly used software in genome assembly

Type	Software	Description	Resources
De novo assembly	Canu	Reconstruct long sequences of genomic DNA from fragmentary data	http://wgs-assembler.sourceforge.net/wiki/index.php?title=Main_Page
	Arachne	Package for assembling genome sequence using paired-end whole-genome shotgun reads	https://github.com/cseed/arachne-pnr
	Velvet	Sequence assembler for very short reads using de Bruijn graphs	https://www.ebi.ac.uk/~zerbino/velvet/
	SOAPdenovo	Short-read assembly method for the human-sized genomes	http://soap.genomics.org.cn/soapdenovo.html
	ABYSS	Assembles the very large data sets produced by sequencing individual human genomes	http://www.bcgsc.ca/platform/bioinfo/software/abyss
Reference-based	SeqMap	Map large amount of oligonucleotide to the genome	http://www-personal.umich.edu/~jianghui/seqmap/
	MAQ	Build mapping assemblies from short reads generated by NGS machines	http://maq.sourceforge.net/
	RMAP	Map reads with or without error probability information and supports paired-end reads	http://rulai.cshl.edu/rmap/
Other	SPAdes Assembly	Standard isolates and single-cell MDA bacteria assembler Unique and stable identifier for the set of sequences that comprise a specific genome assembly	http://cab.spbu.ru/software/spades/ https://www.ncbi.nlm.nih.gov/assembly/
	GAML	Systematic combination of diverse sequencing datasets into a single assembly	http://compbio.fmph.uniba.sk/gaml/

Quality control of sequence reads must be executed during the whole workflow, from sample preparation to alignment and assembly, as well in variant calling and annotation. Many issues can occur when processing genomic data. For example, the reads could be misaligned when performing alignment and assembly, causing systematic error of the device in sequencing and machine cycle (Torri *et al.*, 2012). These problems could lead to the quality score per base and affect the accuracy of variant calling. Unifying the aligned reads into Sequence Alignment/Map or Binary Sequence Alignment/Map format is essential for quality control (Wang *et al.*, 2012). This process could provide us a clean, well sorted, and indexed file as output that could be used in the subsequent analysis. In addition, in some WGS workflows, advanced quality control is also required in recalibrating the base quality scores of sequence reads for variation.

Variant Calling

After the alignment and assembly of short reads, the next step is variant calling, which is a very important step in the WGS pipeline. Variant calling is a conclusion that there is a difference between target sequence and reference at a given position in an individual genome. It is implemented after genome sequence alignment and assembly and the output of variant calling is stored in a standardized generic format called variant call format. These aligned reads differ from genome reference and the variants occurring are probably related to a certain kind of disease or only simple natural mutations without any functional effect in the cell. Based on the characteristics of variants, several different variations are shown in Table 3, which include SNPs/variants, indels, larger structural variants, and CNVs. We give an overview of these four variant calling types and provide the corresponding software packages for the implementation of variant calling.

Table 3 Variant calling tooling for the detection of SNVs, indels, SVs, and CNVs

Calling Type	Software	Simple description	Resources
SNP/Indel	GATK	A collection of command line for the analysis of variant discovery	https://software.broadinstitute.org/gatk/
	SAMtools	A suite of programs for interacting with high-throughput sequencing data	http://www.htslib.org/
	Isaac Variant Caller	Detects SNVs/indels from the aligned sequencing reads of a single diploid sample	https://github.com/sequencing/isaac_variant_caller
	SOAPsnp	Software package for detecting somatic mutation of SNVs by resequencing	http://soap.genomics.org.cn/soapsnp.html
	Dindel	A Bayesian method to call indels from short-read sequence data in individuals	https://github.com/genome/dindel-tgi
SV/CNV	Breakdancer	Detect indels ranging from 10 base pairs to 1 megabase pair via a single conventional approach	http://gmt.genome.wustl.edu/packages/breakdancer/
	iSV	A pipeline that applies multiple tools for detecting structural variants	http://nagasakilab.csmi.org/en/isvp
	Pindel	Detects breakpoints of large deletions, medium sized insertions, inversions	https://github.com/genome/pindel
	SvABA	Detects SVs from short-read sequencing data using genome-wide local assembly	https://github.com/walaj/svaba
	SVMerge	A pipeline detects SVs by integrating calls from existing SV callers	http://svmerge.sourceforge.net/

SNP calling aims to determine in which positions there are polymorphisms or at least one of the bases differs from a reference sequence to describe the genetic relationship between isolates (Nielsen *et al.*, 2011). Accurate calling of SNPs needs high-quality sequencing data, high coverage, and a thorough bioinformatics approach to identify the SNPs in a statistically relevant manner. Indel calling is also a very common form of variant calling that corresponds to the insertion or deletion of base pairs in the sequence of an organism (Hasan *et al.*, 2015). Compared with SNP calling, accurate indel calling is more challenging, because the occurrence rate of indels is almost 8 times lower than SNPs. Various approaches have been applied to perform SNP and indel calling. Bayesian probabilistic procedure is the most frequently used method in the software packages to implement SNP or indel calling. It calculates the score of probability on each genotype and selecting the genotype in the position with highest probability (Li, 2012). The tools include GATK, SOAPsnp, SAMtools, and Isaac Variant Caller (McKenna *et al.*, 2010; Li *et al.*, 2009a,b, 2008; Raczy *et al.*, 2013). In addition, the pattern growth approach is another common method for indel calling by computing the fragments compared to the reference genome from paired-end reads, such as Dindel (Albers *et al.*, 2011).

Moreover, SV/CNV calling is similar to SNP and indel calling. The structural variants have the potential to impact large stretches of sequence, disrupting genes and regulatory elements that are associated with genetic disorders and used as disease markers in clinical diagnosis of diseases. To call large-scale structural variants, which are the source of sequence variation, involving deletion, insertion, and rearrangement of genomic material, it uses discordant fragments and depends on the paired-end reads. By resolving discordant fragments into some breakpoints between fragments through mapping of each end of the paired-end read, the category of a structural variant can be deduced (Fiegler *et al.*, 2006). For example, chromosomal translocations are a type of large-scale SV involving rearrangement of chromosomes. They can disrupt sequences and create new fusion products. Several signature translocations have been identified as mutations that drive the development of disease. The methods that call the relevant variants mainly involve read pair, split read, read depth, and de novo assembly (Alkan *et al.*, 2011a,b). However, there is still a lack of computational framework for SV/CNV calling. Up to now, only a few computational tools have been available, such as Breakdancer, iSV, Pindel, SvABA, and SVMerge (Chen *et al.*, 2009; Mimori *et al.*, 2013; Ye *et al.*, 2009; Wong *et al.*, 2010). Details of the software used for variant calling are listed in Table 3.

Annotation and Analysis

Once the variant calling is completed, the next crucial step is the annotation between the target genome and reference genome to find the potential differences. Genome annotation includes the identification of genome segments of known and probable open reading frames and matching the identified segments to the detected gene sequences from the existing databases (Kwong *et al.*, 2015). It needs to be annotated with biological information. These biological related data range from gene models to gene functions including the Kyoto Encyclopedia of Genes and Genomes (Kanehisa and Goto, 2000) and Gene Ontology (Primmer *et al.*, 2013). A typical genome annotation constitutes a significant effort recorded on the human genome sequences in the reports, which present considerably detailed genome information. The unit record for each genome annotation describes an individual gene and its protein product. Meanwhile, simple evidence of the assigned function for the variants may also be stored in the file, for example, boundary of domains and functionally characterized homolog, but the details of the information need further exploration. The process of annotation conceptually contains two parts, namely the computational part and the annotation part. The computational part consists of the data from genomes or specific transcriptomes that are used in parallel to create initial gene and transcript information. The annotation part is used to synthesize the information into a gene annotation by a set of rules in an annotation pipeline. Usually, the success of genome annotation strongly relies on the quality of genome assembly so that it will produce satisfying results in terms of near-complete genomes broken by small gaps.

The purpose of genome analysis is to explore the effect of genomic variants on wild-type gene function and further on overall individual health. The progress of WGS has been yielding an increasing number of genetic variants. Identification of the disease-associated variants and seeking their specific types, locations, potential effects, and functions is complicated and challenging because of issues such as complex genomic regions, inaccurate variant calling, detection of the phase of the locus, etc. The causation between genomic variants and disease association is the key point during the process of genomic annotation and analysis, which could help us acquire explicit understanding on variation effects of individuals (Albert and Kruglyak, 2015). Many excellent resources and programmable workflows are usable for the annotation and analysis of identified variants that differ by type. Almost all the tools can process SNPs and indels, but for SV annotations, only a few tools are available like ANNOVAR (Wang *et al.*, 2010) and VAT (Habegger *et al.*, 2012). Also, these tools could achieve distinct functions such as reference database annotation, evolutionary information, prediction of identified variant effect, allele frequency, chromosomal position regarding to the nearby site of interest (e.g., gene, segmental duplication, splice site, regulatory region, etc.) (Bromberg, 2013). All these findings or results are the outcomes of individual variants that can be used for the diagnosis of disease. Some common tools for the execution of annotation and analysis are shown in Table 4.

The frameworks above present the steps and provide available tools for the implementation in each process. Such processes sequentially linked together constitute the complete computational WGS workflow. A number of platforms have been developed for the processing of genome sequencing analysis that integrate and manage pipelines. The most widely used WGS platform may be Illumina, which achieves DNA colony amplification by bridge PCR, sequencing-by-synthesis, and fluorescence imaging (Lunter and Goodson, 2011). The types of Illumina instruments vary a lot in terms of the function and cost, including MiSeq, NextSeq, HiSeq, HiSeq X Ten, etc. The Ion Torrent platform, a competitor to Illumina, is similar to cyclic array platforms in amplifying DNA colonies on beads that are immobilized within wells on a patterned surface. This system utilizes a novel and nonoptical detection method, measuring the release of a hydrogen ion (Merriman *et al.*, 2012). Pacific Biosciences is another platform that offers comprehensive solutions and advanced applications of WGS for complete gene discovery. Though, the cost is more expensive than alternatives, the combination of long reads and minimal GC bias for large-scale WGS by Pacific Biosciences allows access to a substantially larger portion of human genetic variation, an intrinsic advantage over short-read sequencing technology (Carneiro *et al.*, 2012). Another noteworthy platform is Oxford Nanopore, which uses low quantities of input nucleotides and long reads without manipulation (Mikheyev and Tin, 2014). Besides, there are many other companies and communities that are keen on exploiting novel and advanced WGS technologies including Galaxy, LONI, GATK, Taverna, etc. (Giardine *et al.* 2005; Rex *et al.*, 2003; McKenna *et al.*, 2010; Oinn *et al.*, 2004), so we are optimistic for the prospects of WGS platform development in the future.

Clinical Workflow and Application

The application of WGS shows a broad range across different fields, including agriculture, food, public health, even in space for a variety of settings and reasons. In this article, we only discuss its clinical applications, which are closely related to human health and life. The clinical application of WGS represents the elucidation of the genomic determinants of humans' heritable makeup. In fact, many genome projects have been released concentrating on the use of WGS for diagnosing those with rare and common diseases, cancer, etc. We anticipate that this description will continue to advance our understanding of pathophysiology with WGS and realize its value in discovering genes and variants for underlying disease and improving public health.

Current clinical applications mainly consist of Mendelian diseases, complex diseases, and cancer (Wu *et al.*, 2016). The diagnosis of Mendelian disease is the most common application of genome sequencing of individual patients that has been successfully applied in the identification of causal mutations (Lupski *et al.*, 2010; Roach *et al.*, 2010). The largest collection of Mendelian disease contains 7000 distinct diseases that are characterized based on the OMIM database (Hamosh *et al.*, 2002).

Table 4 A list of available software in genome annotation

Software	Simple description	Sources
ANNOVAR	Utilizes up-to-date information to annotate genetic variants detected from diverse genomes	http://annovar.openbioinformatics.org/en/latest/
VAT	A computational framework to functionally annotate variants in personal genomes	http://vat.gersteinlab.org/
VARIANT	Provides annotation of variants from NGS based on several different databases and repositories	http://variant.bioinfo.cipf.es/
VAAST	Identifies damaged genes and their disease-causing variants in personal genome sequences	http://www.yandell-lab.org/software/vaast.html
snpEFF	Annotates and predicts the effects of SNP, indel, and MNP variants in genomic sequences	http://snpeff.sourceforge.net/
VariBench	A benchmark database for testing and training methods for variation effect prediction	http://structure.bmc.lu.se/VariBench/
PolyPhen/ PolyPhen-2	Algorithms that predict possible impact of an AAS on the structure and function of human protein based on sequence, phylogenetics, and structure	http://genetics.bwh.harvard.edu/pph2/
SeattleSeq	Annotate novel and known SNVs and small indels with biological functions, protein positions, etc	http://snp.gs.washington.edu/SeattleSeqAnnotation150/

Almost half of them are caused by unknown genetic elements. The diagnosis of Mendelian disease by WGS workflow includes the combination of gene information, variant function prediction, variant frequency, and evolutionary conservation. The rationale underneath is the assumption that the variants related to the Mendelian diseases that change the protein function on corresponding genes tend to be rare. Common or complex diseases are another type of clinical application affected by more than one variant. Genome-wide association studies explore complex disease due to the variants that contribute to disease susceptibility with modest effect size. However, genome-wide association only genotypes common variants. The technique of WGS makes it possible to annotate all variants, including rare and common variants, and guarantee the sequencing of real functional genes. The identification of cancer, regarded as an important type of complex disease, has been in great demand in recent years. Much attention has been paid using whole exome sequencing to detect multiple types of cancers such as breast cancer, gastric cancer, and prostate cancer (Barbieri *et al.*, 2012; Wang *et al.*, 2011; Thompson *et al.*, 2012). Although certain types of cancer detection have obtained immense success, it still exposes the issue that whole exome sequencing only focuses on the protein coding region and ignores the noncoding region, which blocks the wider and deeper study of cancer. With increasing development of sequencing technology, WGS is solving the issue of identifying noncoding regions that are frequently mutated across cancer types, which creates opportunities for personalized medicine in the future (Tsai *et al.*, 2016).

We provide the MedSeq Project, the first randomized clinical trial that integrating WGS into clinical medicine, as an example to draw the outline of WGS clinical workflow for healthy individuals and individuals with disease (Vassy *et al.*, 2014). The holistic view of clinical WGS workflow is depicted in Fig. 2; this mainly consists of five components, namely patients, hospitals, lab experiments, reports and feedback, and solutions.

Limitations and Future Development

With the rapid progress of genome sequencing techniques and powerful tools and platforms, there are still limitations that obstruct the widespread use of WGS. The first challenge is that WGS may not detect all of the variations in a genome. For example, variations that cause DNA repetitions or rearrangements can hardly be detected by WGS, which means you may have variants that could deceive you to a certain genetic condition that escapes detection. Also, occasionally a portion of a gene may not be readable by WGS and the variants could be missed. The second is the interpretation of genetic variants in genes, especially related to human disease. The lack of a test and comparison in clinical research is still an immense issue, although many relevant studies are frequently published on the meaning of biological variants (Krier *et al.*, 2016). The development of a standardized evaluation of

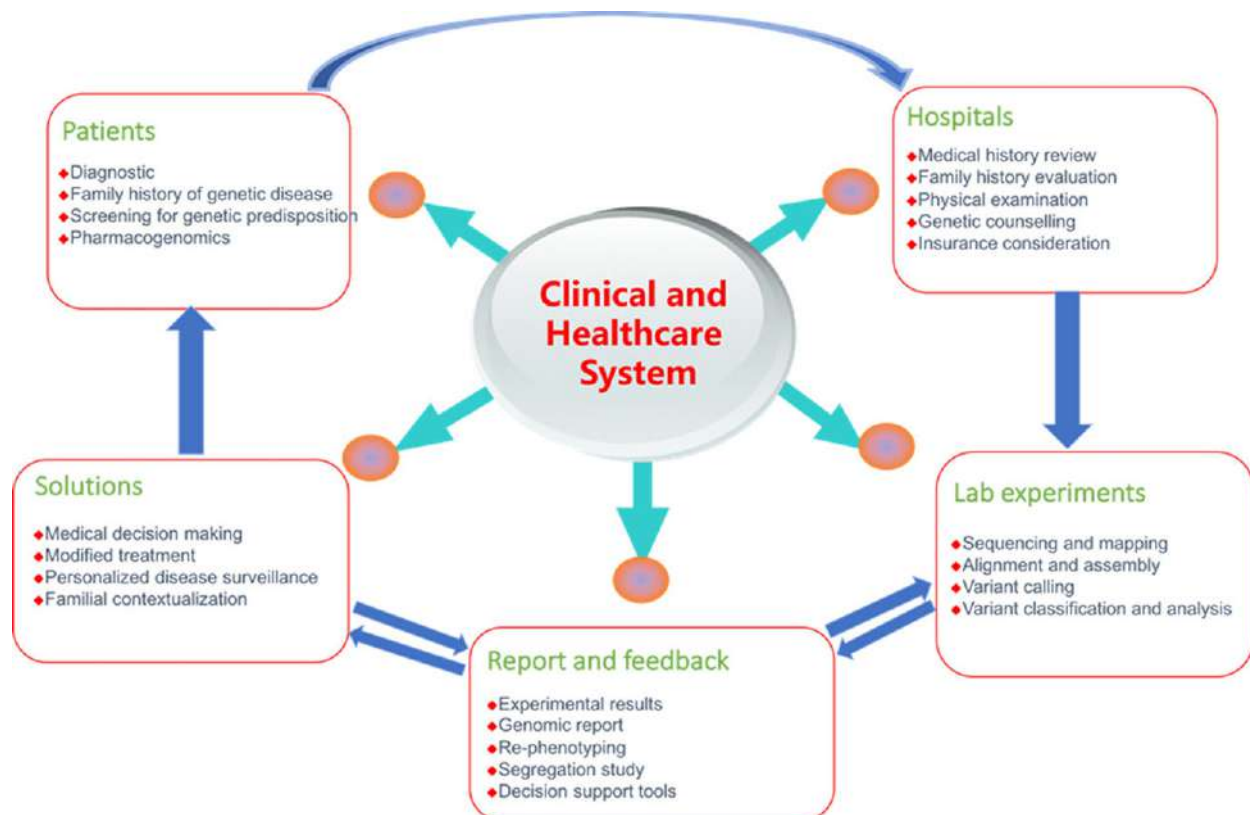


Fig. 2 The outline of clinical workflow of whole genome sequencing.

WGS pipelines for validation in clinics is of great significance to public health. The third remains the further improvement of methods and techniques for pipeline construction. WGS has extensively detected SNVs and indels, but for the identification of SVs and CNVs as well as genomic rearrangements, it remains in its infancy. Therefore, more innovative methods and advanced analysis techniques (such as deep WGS), which have not yet been applied as part of common workflow in most of the existing research and practical projects, need to be further explored.

Despite these challenges, we are still confident and looking forward to the bright future of WGS. We can expect continuous improvement on the steps of the WGS pipeline by sequencing technology and computational methods. New library construction enables sequencing by producing libraries with longer and more precise sizes and generating longer reads with less error rate. More efficient algorithms and stronger computation power could further reduce computing time and cost for genome sequencing, which will make WGS become routine in both genetics research and clinical application. After the enhancement in techniques and methods, there is no doubt that user-friendly and advanced workflows are the key to facilitate more widespread use of WGS. The unification of the pipeline with WGS in a shared and specified language, as well as common tools, will bring much convenience with the expanding community of users and producers of genomic data. This transition makes it possible that even small research teams or individuals with limited resources will be able to develop genomic studies and strengthen the use of WGS techniques in biology and related fields. As more multidimensional data and more-populated databases are generated (e.g., genetic pathways, functions, biomedical information) through research institutions and hospital networks, abundant variants will be detected, which will bring us a great chance for new discoveries to figure out the relation between variations and effects in certain diseases. In addition, the methods for storing and sharing WGS data seamlessly, construction of databases containing millions of individuals with diverse health networks using WGS, as well as the integration of genotypic and phenotypic information are also critical steps for the prosperity of future directions in WGS development.

See also: Computational Pipelines and Workflows in Bioinformatics. Constructing Computational Pipelines. Exome Sequencing Data Analysis. Genome Annotation: Perspective From Bacterial Genomes. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing

References

- Albers, C.A., Lunter, G., MacArthur, D.G., *et al.*, 2011. Dindel: Accurate indel calls from short-read data. *Genome Research* 21 (6), 961–973.
- Albert, F.W., Kruglyak, L., 2015. The role of regulatory variation in complex traits and disease. *Nature Reviews Genetics* 16 (4), 197–212.
- Alkan, C., Coe, B.P., Eichler, E.E., 2011a. Genome structural variation discovery and genotyping. *Nature Reviews Genetics* 12 (5), 363–376.
- Alkan, C., Sajjadian, S., Eichler, E.E., 2011b. Limitations of next-generation genome sequence assembly. *Nature Methods* 8 (1), 61–65.
- Dervan, A., Shendure, J., 2017. Chapter 3 – The state of whole-genome sequencing. In: Ginsburg, G.S., Willard, H.F. (Eds.), *Genomic and Precision Medicine*, third ed. Boston: Academic Press, pp. 45–62. ISBN 9780128006818.
- Barbieri, C.E., Baca, S.C., Lawrence, M.S., *et al.*, 2012. Exome sequencing identifies recurrent SPON, FOXA1 and MED12 mutations in prostate cancer. *Nature Genetics* 44 (6), 685–689.
- Bromberg, Y., 2013. Building a genome analysis pipeline to predict disease risk and prevent disease. *Journal of Molecular Biology* 425 (21), 3993–4005.
- Carneiro, M.O., Russ, C., Ross, M.G., *et al.*, 2012. Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC Genomics* 13 (1), 375.
- Chard, K., Onyuksel, C., Tan, W., *et al.*, 2008. Build grid enabled scientific workflows using gRAVI and taverna. In: *Proceeding of the IEEE Fourth International Conference on eScience*, 2008, pp. 614–619. IEEE.
- Chen, K., Wallis, J.W., McLellan, M.D., Larson, *et al.*, 2009. BreakDancer: An algorithm for high-resolution mapping of genomic structural variation. *Nature Methods* 6 (9), 677–681.
- Dalca, A.V., Brudno, M., 2010. Genome variation discovery with high-throughput sequencing data. *Briefings in Bioinformatics* 11 (1), 3–14.
- Fiegler, H., Redon, R., Andrews, D., *et al.*, 2006. Accurate and reliable high-throughput detection of copy number variation in the human genome. *Genome Research* 16 (12), 1566–1574.
- Fleischmann, R.D., Adams, M.D., White, O., *et al.*, 1995. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* 269, 496–512.
- Flicek, P., Birney, E., 2009. Sense from sequence reads: Methods for alignment and assembly. *Nature Methods* 6, S6–S12.
- Giardine, B., Riemer, C., Hardison, R.C., *et al.*, 2005. Galaxy: A platform for interactive large-scale genome analysis. *Genome Research* 15 (10), 1451–1455.
- Goecks, J., Nekrutenko, A., Taylor, J., 2010. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology* 11 (8), R86.
- Habegger, L., Balasubramanian, S., Chen, D.Z., *et al.*, 2012. VAT: A computational framework to functionally annotate variants in personal genomes within a cloud-computing environment. *Bioinformatics* 28 (17), 2267–2269.
- Hamosh, A., Scott, A.F., Amberger, J., *et al.*, 2002. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research* 30 (1), 52–55.
- Hasan, M.S., Wu, X., Zhang, L., 2015. Performance evaluation of indel calling tools using real short-read data. *Human Genomics* 9 (1), 20.
- Heather, J.M., Chain, B., 2016. The sequence of sequencers: The history of sequencing DNA. *Genomics* 107 (1), 1–8.
- Hillier, L.W., Marth, G.T., Quinlan, A.R., *et al.*, 2008. Whole-genome sequencing and variant discovery in *C. elegans*. *Nature Methods* 5 (2), 183–188.
- Jaffe, D.B., Butler, J., Gnerre, S., *et al.*, 2003. Whole-genome sequence assembly for mammalian genomes: Arachne 2. *Genome Research* 13 (1), 91–96.
- Jiang, H., Wong, W.H., 2008. SeqMap: Mapping massive amount of oligonucleotides to the genome. *Bioinformatics* 24 (20), 2395–2396.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28 (1), 27–30.
- Kidd, J.M., Cooper, G.M., Donahue, W.F., *et al.*, 2008. Mapping and sequencing of structural variation from eight human genomes. *Nature* 453 (7191), 56–64.
- Krier, J.B., Kalia, S.S., Green, R.C., 2016. Genomic sequencing in clinical practice: Applications, challenges, and opportunities. *Dialogues in Clinical Neuroscience* 18 (3), 299.
- Kwong, J.C., McCallum, N., Sintchenko, V., Howden, B.P., 2015. Whole genome sequencing in clinical and public health microbiology. *Pathology* 47 (3), 199–210.
- Lander, E.S., Linton, L.M., Birren, B., *et al.*, 2001. Initial sequencing and analysis of the human genome. *Nature* 409 (6822), 860–921.
- Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L., 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10 (3), R25.

- Leipzig, J., 2017. A review of bioinformatic pipeline frameworks. *Briefings in Bioinformatics* 18 (3), 530–536.
- Li, H., 2012. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics* 28 (14), 1838–1844.
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25 (14), 1754–1760.
- Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009a. The sequence alignment/map format and SAMtools. *Bioinformatics* 25 (16), 2078–2079.
- Li, R., Li, Y., Kristiansen, K., Wang, J., 2008. SOAP: Short oligonucleotide alignment program. *Bioinformatics* 24 (5), 713–714.
- Li, R., Yu, C., Li, Y., *et al.*, 2009b. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics* 25 (15), 1966–1967.
- Lunter, G., Goodson, M., 2011. Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Research* 21 (6), 936–939.
- Luo, R., Liu, B., Xie, Y., *et al.*, 2012. SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. *Gigascience* 1 (1), 18.
- Lupski, J.R., Reid, J.G., Gonzaga-Jauregui, C., *et al.*, 2010. Whole-genome sequencing in a patient with Charcot–Marie–Tooth neuropathy. *New England Journal of Medicine* 362 (13), 1181–1191.
- McKenna, A., Hanna, M., Banks, E., *et al.*, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20 (9), 1297–1303.
- Meaburn, E., Schulz, R., 2012. Next generation sequencing in epigenetics: Insights and challenges. In: *Seminars in Cell & Developmental Biology*, vol. 23 (2), pp. 192–199. Academic Press.
- Merriman, B., Torrent, I., Rothberg, J.M., R&D Team, 2012. Progress in ion torrent semiconductor chip based sequencing. *Electrophoresis* 33 (23), 3397–3417.
- Mikheyev, A.S., Tin, M.M., 2014. A first look at the Oxford Nanopore MiniON sequencer. *Molecular Ecology Resources* 14 (6), 1097–1102.
- Mimori, T., Nariai, N., Kojima, K., *et al.*, 2013. iSVF: An integrated structural variant calling pipeline from high-throughput sequencing data. *BMC Systems Biology* 7 (6), S8.
- Moorhith, S., Mattocks, C.J., Wright, C.F., 2011. Review of massively parallel DNA sequencing technologies. *The HUGO Journal* 5 (1–4), 1–12. <http://doi.org/10.1007/s11568-011-9156-3>.
- Myers, E.W., 1995. Toward simplifying and accurately formulating fragment assembly. *Journal of Computational Biology* 2 (2), 275–290.
- Myers, E.W., Sutton, G.G., Delcher, A.L., *et al.*, 2000. A whole-genome assembly of *Drosophila*. *Science* 287 (5461), 2196–2204.
- Ng, P.C., Kirkness, E.F., 2010. Whole genome sequencing. In: *Genetic Variation*. Humana Press, pp. 215–226.
- Nielsen, R., Paul, J.S., Albrechtsen, A., Song, Y.S., 2011. Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics* 12 (6), 443–451.
- Oinn, T., Addis, M., Ferris, J., *et al.*, 2004. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20 (17), 3045–3054.
- Pepke, S., Wold, B., Mortazavi, A., 2009. Computation for ChIP-seq and RNA-seq studies. *Nature Methods* 6, S22–S32.
- Pevzner, P.A., Tang, H., Waterman, M.S., 2001. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences* 98 (17), 9748–9753.
- Primmer, C.R., Papakostas, S., Leder, E.H., Davis, M.J., Ragan, M.A., 2013. Annotated genes and nonannotated genomes: Cross-species use of Gene Ontology in ecology and evolution research. *Molecular Ecology* 22 (12), 3216–3241.
- Raczy, C., Petrovski, R., Saunders, C.T., *et al.*, 2013. Isaac: Ultra-fast whole-genome secondary analysis on Illumina sequencing platforms. *Bioinformatics* 29 (16), 2041–2043.
- Rex, D.E., Ma, J.Q., Toga, A.W., 2003. The LONI pipeline processing environment. *Neuroimage* 19 (3), 1033–1048.
- Roach, J.C., Glusman, G., Smit, A.F., *et al.*, 2010. Analysis of genetic inheritance in a family quartet by whole-genome sequencing. *Science* 328 (5978), 636–639.
- Robertson, G., Schein, J., Chiu, R., *et al.*, 2010. De novo assembly and analysis of RNA-seq data. *Nature Methods* 7 (11), 909–912.
- Rumble, S.M., Lacroute, P., Dalca, A.V., *et al.*, 2009. SHRiMP: Accurate mapping of short color-space reads. *PLOS Computational Biology* 5 (5), e1000386.
- Sanger, F., Nicklen, S., Coulson, A.R., 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences* 74 (12), 5463–5467.
- Saraswathy, N., Ramalingam, P., 2011. *Concepts and Techniques in Genomics and Proteomics*. Elsevier.
- Schuster, S.C., 2008. Next-generation sequencing transforms today's biology. *Nature Methods* 5 (1), 16–18.
- Simpson, J.T., Wong, K., Jackman, S.D., *et al.*, 2009. ABySS: A parallel assembler for short read sequence data. *Genome Research* 19 (6), 1117–1123.
- Slatko, B.E., Albright, L.M., Tabor, S., Ju, J., 2001. DNA sequencing by the dideoxy method. *Current Protocols in Molecular Biology*. 4–7.
- Smith, A.D., Chung, W.Y., Hodges, E., *et al.*, 2009. Updates to the RMAP short-read mapping software. *Bioinformatics* 25 (21), 2841–2842.
- Staden, R., 1979. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research* 6 (7), 2601–2610.
- Thompson, E.R., Doyle, M.A., Ryland, G.L., *et al.*, 2012. Exome sequencing identifies rare deleterious mutations in DNA repair genes FANCC and BLM as potential breast cancer susceptibility alleles. *PLOS Genetics* 8 (9), e1002894.
- Torri, F., Dinov, I.D., Zamanyan, A., *et al.*, 2012. Next generation sequence analysis and computational genomics using graphical pipeline workflows. *Genes* 3 (3), 545–575.
- Tsai, E.A., Shakhbatyan, R., Evans, J., *et al.*, 2016. Bioinformatics workflow for clinical whole genome sequencing at partners healthcare personalized medicine. *Journal of Personalized Medicine* 6 (1), 12.
- Van Dijk, E.L., Jaszczyszyn, Y., Thermes, C., 2014. Library preparation methods for next-generation sequencing: Tone down the bias. *Experimental Cell Research* 322 (1), 12–20.
- Vassy, J.L., Lautenbach, D.M., McLaughlin, H.M., *et al.*, 2014. The MedSeq Project: A randomized trial of integrating whole genome sequencing into clinical medicine. *Trials* 15 (1), 85.
- Wang, D.L., Zender, C.S., Jenks, S.F., 2009. Efficient clustered server-side data analysis workflows using SWAMP. *Earth Science Informatics* 2 (3), 141–155.
- Wang, K., Kan, J., Yuen, S.T., *et al.*, 2011. Exome sequencing identifies frequent mutation of ARID1A in molecular subtypes of gastric cancer. *Nature Genetics* 43 (12), 1219–1223.
- Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38 (16), e164.
- Wang, L., Wang, S., Li, W., 2012. RSeQC: Quality control of RNA-seq experiments. *Bioinformatics* 28 (16), 2184–2185.
- Wong, K., Keane, T.M., Stalker, J., Adams, D.J., 2010. Enhanced structural variant and breakpoint detection using SVMerge by integration of multiple detection methods and local assembly. *Genome Biology* 11 (12), R128.
- Wong, P.B., Wiley, E.O., Johnson, W.E., *et al.*, 2012. Tissue sampling methods and standards for vertebrate genomics. *GigaScience* 1 (1), 8.
- Wu, J., Wu, M., Chen, T., Jiang, R., 2016. Whole genome sequencing and its applications in medical genetics. *Quantitative Biology* 4 (2), 115–128.
- Ye, K., Schulz, M.H., Long, Q., Apweiler, R., Ning, Z., 2009. Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* 25 (21), 2865–2871.
- Yoo, J., Ha, I.C., Chang, G.T., *et al.*, 2011. CNVAS: Copy Number Variation Analysis System – The analysis tool for genomic alteration with a powerful visualization module. *BioChip Journal* 5 (3), 265–270.
- Zafar, H., Wang, Y., Nakleh, L., Navin, N., Chen, K., 2016. Monovar: Single-nucleotide variant detection in single cells. *Nature Methods* 13 (6), 505–507.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18 (5), 821–829.

Metagenomic Analysis and its Applications

Arpita Ghosh, Eurofins Genomics India Pvt. Ltd., Bangalore, India and Centre for Bioinformatics, Perdana University, Selangor, Malaysia

Aditya Mehta, Eurofins Genomics India Pvt. Ltd., Bangalore, India

Asif M Khan, Perdana University, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Metagenomic analysis is commonly used to investigate complex microbial communities sampled directly from the environment, without culturing or isolating a single organism. Microbes have important roles to play in various ecosystems, however, many remain to be characterized in detail. The term metagenomics was coined in 1998 ([Handelsman et al., 1998](#)); it helps to understand various aspects of a sample, and allows one to characterize the microbes in the given environmental sample. It helps identify the species present in the community and also provides insights about the metabolic activities and functional roles of the microbes in the environmental sample ([Langille et al., 2013](#)).

Current approaches allow one to understand the complex properties of the microbiota, their dynamics and function in the natural system. Various metagenomic approaches answer fundamental questions such as which organisms are present (Taxonomic diversity), and what roles they play (Functional metagenomics) ([Vieites et al., 2008](#)). The amplification of specific targeted genes such as (V1-V9) of 16S rRNA, 18S rRNA, ribosomal ITS, NifH, among others, by PCR before sequencing permit diversity analysis ([Morgan and Huttenhower, 2012](#)). The relative abundance of the most abundant microorganisms of a particular group is not necessarily associated with the importance of that group in the functioning of the community. The most abundant organisms may not always play the most critical role in a community, while organisms constituting only 0.1% of the community (e.g., nitrogen fixers) can have very important functions ([Dinsdale et al., 2008](#)).

Types of Metagenomics Studies

The two most commonly used methods of pathogen identification for high throughput data are (1) Amplicon based method which includes 16S ribosomal RNA for bacteria, internal transcribed spacer (ITS) and 18S region for fungi and eukaryotes, respectively, and (2) whole metagenomic shotgun sequencing.

Shotgun Metagenomics

Shotgun metagenomic analysis has the ability to identify the majority of the organisms (culturable and unculturable bacteria) in the environmental sample. A community biodiversity profile is created, which is further associated with functional composition analysis of organism lineages (i.e., genera or taxa) ([Tringe et al., 2005](#)). Shotgun metagenomic studies can be divided into two types: (1) Sequence-based screens, which describe the microbial diversity and genomes of a particular environmental sample, and (2) Functional screens, which identify the functional gene products, but do not determine from what species the genetic material originated ([Madhavan et al., 2017](#)). Before initiating a whole metagenomic study, an understanding of the potential microbial diversity and the relative abundance of species in the environmental sample is very important. For example, the metagenome of soil samples will consist of a more complex microbial community, than human skin. Hence, for proper coverage, more data must be generated in case of soil than for human skin. A higher sequencing depth also allows the detection of rare taxa ([Sharpton, 2014](#)). This makes shotgun metagenomic sequencing much more expensive than 16S sequencing, in order to achieve the coverage and depth needed for species identification ([Quail et al., 2012](#)).

Amplicon Based 16S

16S sequencing is a widely used technique that relies on the variable regions (V1-V9) of the bacterial 16S rRNA gene to make community-wide taxonomic assignments ([Chakravorty et al., 2007](#)). It is also used for microbial diversity analysis and has been used for various environmental samples, such as soil ([Chong et al., 2012](#)) and human gut ([Dethlefsen et al., 2008](#)), among others. Some degree of divergence is allowed during the sequence similarity assessment stage of the analysis; typically, nearly identical sequences (> 97%) are clustered into Operational Taxonomical Units (OTU) ([Morgan and Huttenhower, 2012](#)). The limitation of this method is that, if any two organisms have the same 16S rRNA gene sequence, they may be classified as the same species in a 16S analysis, even if they are from different species. Because 16S analysis is based on the 16S rRNA gene, with OTUs defined as taxa, it is generally not possible to distinguish strains, nor, in some cases, closely related species. For example, *Escherichia coli* O157: H7 and *E. coli* K-12 cannot be differentiated based on 16S analysis ([Weinstock, 2012](#)), nor can *Shigella flexneri* be distinguished from *E. coli* ([Hilton et al., 2016](#)). The OTUs are analyzed at each taxonomic level, but are less precise at the species level ([Ranjan et al., 2016](#)).

Amplicon Based 18S/ITS

The 18S rRNA is one of the basic components of fungal cells and comprises both conserved and hypervariable regions. The internal transcribed spacer region, ITS, is located between the 18S and 5.8S rRNA genes and has a high degree of sequence variation. The 18S rRNA is mainly used for high resolution taxonomic studies of fungi, while the ITS region is widely used for analysing fungal diversity in environmental samples (Bromberg *et al.*, 2015). Taxonomic studies of fungi are often based on the nuclear ribosomal gene cluster, which includes the 18S or small subunit (SSU), 5.8S subunit, and 28S or large subunit (LSU) rRNA genes. ITS1 and ITS2 have been found to be the most suitable markers for fungal phylogenetic analysis due to their variable sequences, conserved primers and multicopy nature (Cuadros-Orellana *et al.*, 2013). Various pipelines, such as QIIME (Caporaso *et al.*, 2010), MG-RAST (Glass *et al.*, 2010) and Mothur (Schloss *et al.*, 2009) are used to perform taxonomic and functional analysis (Table 1). In QIIME, the UNITE database is used as it includes fungal rDNA ITS sequences (Kõljalg *et al.*, 2005). In addition to rRNA genes, other amplicon-based studies are performed in order to focus on specific functions such as nitrogen fixation activity, and diversity analysis of nitrogenase reductase (nifH) genes (Igai *et al.*, 2016). Arbuscular mycorrhizal fungi have been found in a symbiotic relation with the roots of plants (Smith and Read, 2008) based on analyses of the fungal SSU rRNA gene (Vasar *et al.*, 2017).

Metatranscriptomics and Metaproteomics

RNA-Seq analysis of microbial communities in a complex ecosystem is known as Metatranscriptomics (Zhang *et al.*, 2017). Co-expression of gene clusters and transcript abundance, followed by functional annotation, can be studied in environmental samples (Oyserman *et al.*, 2016). The quantitation of mRNA and pathway expression can be carried out using metatranscriptomics. The challenge associated with metatranscriptomic approaches is to get high quality RNA from environmental samples; given this, it is an efficient approach to elucidate gene expression, and to discover novel genes in the microbial community (Frias-Lopez *et al.*, 2008; Tartar *et al.*, 2009).

Table1 List of tools and databases

Category	Tools	References
Assembly	IDBA-UD	Peng <i>et al.</i> (2012)
	MEGAHIT	Li <i>et al.</i> (2015)
	MetaVelvet	Namiki <i>et al.</i> (2012)
	MetaVelvet-SL	Sato and Sakakibara (2014)
	Ray Meta	Boisvert <i>et al.</i> (2012)
	SOAPdenovo2	Luo <i>et al.</i> (2012)
	Omega	Haider <i>et al.</i> (2014)
	metaSPAdes	Nurk <i>et al.</i> (2017)
	MetaAMOS	Treangen <i>et al.</i> (2013)
	SCIMM and PHYSCIMM	Kelley and Salzberg (2010)
Binning	Phymm and PhymmBL	Brady and Salzberg (2009)
	CONCOCT	Alneberg <i>et al.</i> (2014)
	IMG/MER 4	Markowitz <i>et al.</i> (2013)
	MG-RAST	Glass <i>et al.</i> (2010)
	MEGAN	Huson and Weber (2013)
	MetaCluster	Wang <i>et al.</i> (2014)
	MetaGeneMark	Zhu <i>et al.</i> (2010)
Gene Prediction	Prodigal	Hyatt <i>et al.</i> (2010)
	FragGeneScan	Rho <i>et al.</i> (2010)
	Glimmer-MG	Kelley <i>et al.</i> (2011)
	Prokka	Seemann (2014)
	QIIME	Caporaso <i>et al.</i> (2010)
OUT Picking & Clustering	Mothur	Schloss <i>et al.</i> (2009)
	PICRUSt	Langille <i>et al.</i> (2013)
Functional 16S analysis Databases	SILVA	Quast <i>et al.</i> (2012)
	Greengenes	DeSantis <i>et al.</i> (2006)
	Ribosomal database (RDP)	Cole <i>et al.</i> (2006)
	KEGG	Ogata <i>et al.</i> (1999)
	SEED	Overbeek <i>et al.</i> (2005)
	egglog	Powell <i>et al.</i> (2014)
	COG/KOG	Tatusov <i>et al.</i> (2000)
	PFAM	Bateman <i>et al.</i> (2004)
	TIGRFAM	Haft <i>et al.</i> (2003)
	UNITE	Kõljalg <i>et al.</i> (2013)

The study of the proteome expressed in the microbial community is known as Metaproteomics. This has been used to investigate microbial activities along with complex metabolic pathways in soil ecosystems (Zampieri *et al.*, 2016). Community metaproteomics are emerging as complementary approaches to metagenomics and can provide large-scale characterization of proteins in the microbiota, such as in the human gut (Petriz and Franco, 2017). Metagenomics, along with metatranscriptomics and metaproteomics, provide insights into functional dynamics, prediction of the in situ microbial responses/activities, and the production capabilities of microbial communities (Simon and Daniel, 2011).

Analysis of the metagenome

In this section, we discuss the various analysis pipelines for the types of metagenomics studies shown in Fig. 1 (credit to Oulas *et al.*, 2015).

Shotgun Metagenomic Analysis

Metagenomics studies are carried out to investigate both known and unknown species, both culturable and unculturable, in the environmental sample studied. A *de novo* assembly of shorter reads may be carried out to obtain genomic contigs, followed by scaffolding, which is often performed to get a more compact and concise view of the sequenced environmental samples. Achieving full assemblies of the complete genomes present in the community is difficult, and is rarely possible except in simple communities, or with very deep sequencing (Tyson *et al.*, 2004; Venter *et al.*, 2004). In metagenomic studies, insight into the metabolic functions active in the environmental sample can be obtained based on the complete sequences of protein coding genes, as well as from the full operons present in the sequenced genomes (Oulas *et al.*, 2015).

Pre-processing of sequence reads for metagenomics

The raw reads generated from the next generation sequencing platform are subjected to adapter trimming, quality filtration, and de-replication. If the metagenomic sample is isolated from a host organism, then host contamination is typically removed by aligning to the reference genome of the host organism using bowtie2 or other short-read mapper (Oulas *et al.*, 2015).

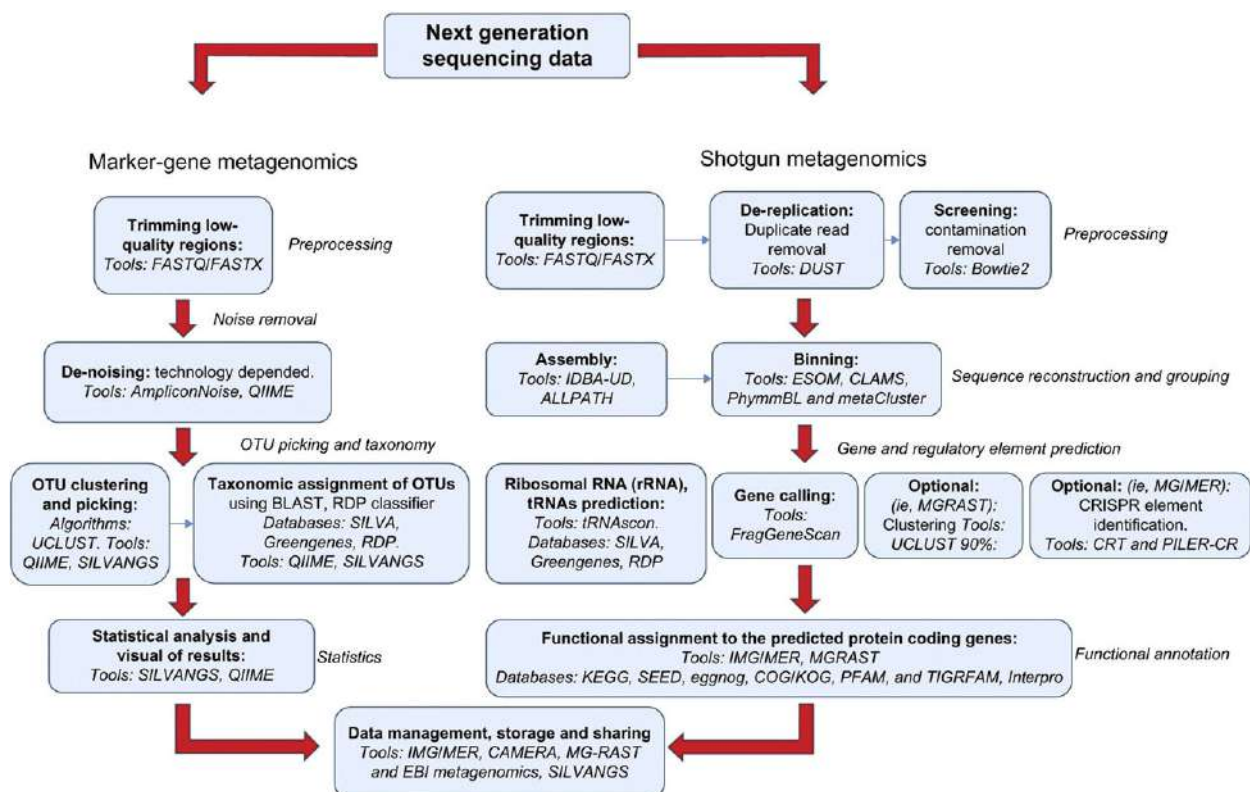


Fig. 1 The overview of the whole metagenomics and 16S analysis (marker-gene metagenomics). Reproduced from Oulas, A., Pavloudi, C., Polymenakou, P., *et al.*, 2015. Metagenomics: Tools and insights for analyzing next-generation sequencing data derived from biodiversity studies. *Bioinformatics and Biology insights* 9, 75.

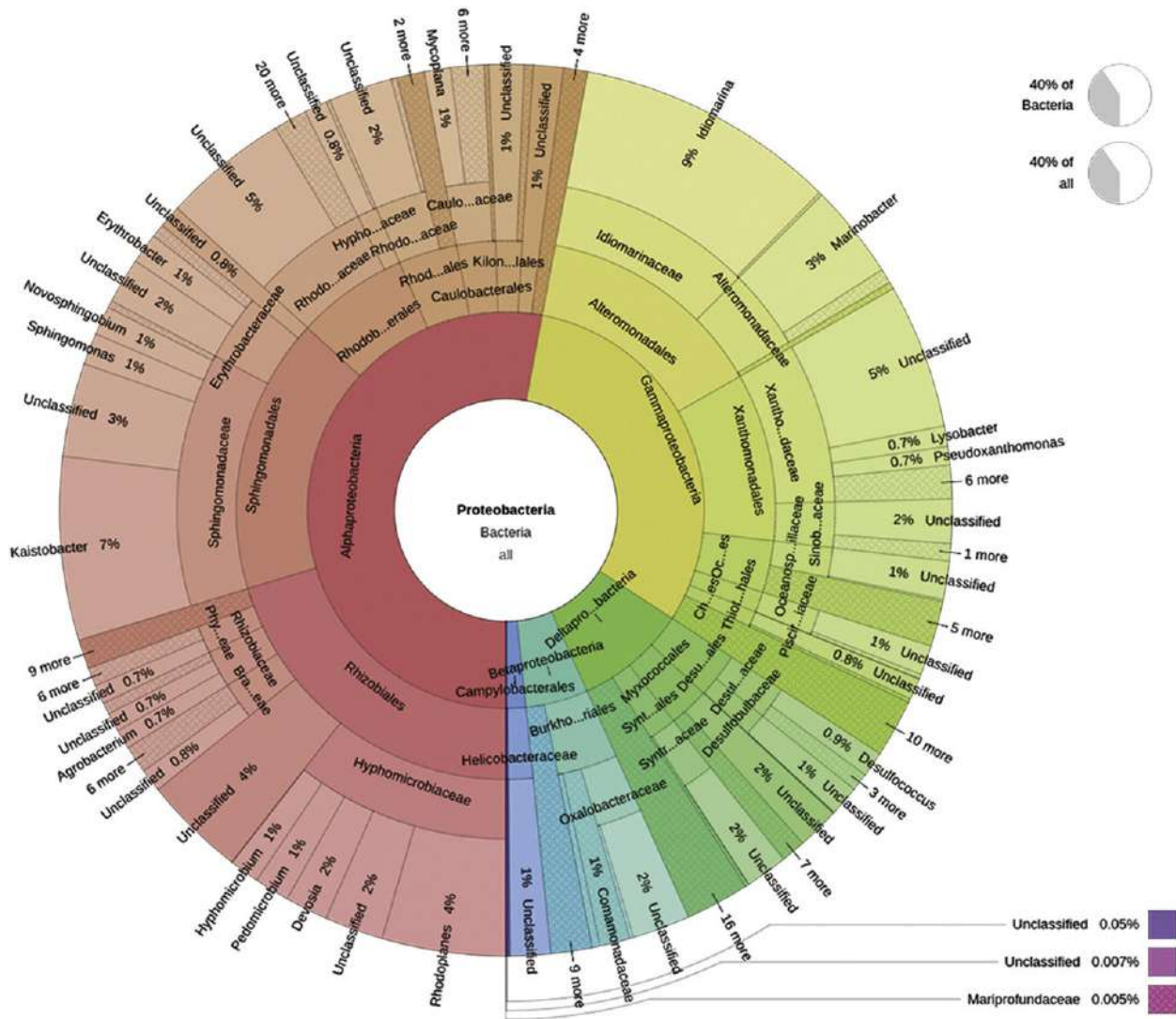


Fig. 2 Representation of Krona. Note: The 16S metagenome sample was analyzed using QIIME and displayed using Krona. The Taxonomy nodes are arranged from the top of the hierarchy at the center and progressing outward. The chart is zoomed to place the domain "Proteobacteria" at the root followed by the taxon.

De novo assembly

Assembly is computationally expensive and requires sophisticated algorithms, such as those based on de Bruijn graphs. Tools that are specifically designed for metagenomic applications are mainly built on de Bruijn graph algorithms. A few common metagenomics assembly tools include CLC workbench, Meta-Ray, MetaVelvet-SL, MetaVelvet, Meta-IDBA SOAP, and metaSPAdes (Nurk *et al.*, 2017; Zerbino and Birney, 2008; Luo *et al.*, 2012). A reference-guided assembly may be carried out when an appropriate reference genome is available (Nagarajan *et al.*, 2010).

Binning

The process of clustering reads or contigs into highly similar groups, and assigning the groups to specific species, subspecies or genera, is called binning. Two types of algorithms are available: (1) Composition-based binning and (2) Similarity-based binning. Certain binning tools deploy hybrid approaches which implement both kinds of algorithms (Oulas *et al.*, 2015). Composition-based binning groups, in a supervised or semi-supervised manner, the DNA fragments with similar composition. Similarity-based methods align the DNA fragments to NR database or reference sequences (Leung *et al.*, 2011).

Annotation

Prediction of CDS (coding DNA sequences), followed by functional assignment is carried out using similarity-based searches of query sequences against databases containing known functional and/or taxonomic information. The taxonomic information can be displayed using Krona, which displays hierarchical data as an interactive multilayered pie chart (Ondov *et al.*, 2011). The

predicted genes are annotated to identify homologous genes (Wheeler *et al.*, 2007), Gene ontology terms (Ashburner *et al.*, 2000), KEGG pathways (Kanehisa *et al.*, 2009), protein families using Pfam (Bateman *et al.*, 2004) or TIGRFams (Haft *et al.*, 2003), clusters of orthologous genes (COGs/KOGs) (Tatusov *et al.*, 2003) or orthologous families (EggNOG, Jensen *et al.*, 2007), and functional motifs using InterPro (Hunter *et al.*, 2008). Sequence similarity-based algorithms such as BLAST require high-performance computer clusters. Some tools, such as Kaiju (Menzel *et al.*, 2016), assign taxonomy using a reference database, and also integrate the Krona tool for visualization as shown in Fig. 2, and COGNIZER (Bose *et al.*, 2015) can be used for functional annotation, which employs a new approach of search strategy that helps in reducing the computational requirements.

Amplicon Based Metagenomic Analysis

The study of microbial diversity in natural environments can be carried out by targeting specific genes. For diversity analysis of bacteria and archaea community, the 16S rRNA gene is commonly used as it contains one or more variable regions (Woese, 1987). For fungi and eukaryotes, the internal transcribed spacer (ITS) and 18S rRNA gene are used, respectively. Commonly used tools for 16S rRNA analysis are QIIME (Quantitative Insights Into Microbial Ecology) (Caporaso *et al.*, 2010), Mothur (Schloss *et al.*, 2009), SILVAngs (Quast *et al.*, 2013), RDPipeline (Ribosomal Database Project Pipeline) (Cole *et al.*, 2013), MEGAN (Huson *et al.*, 2007), and MG-RAST (Metagenomics - Rapid Annotation using Subsystems Technology) (Meyer *et al.*, 2008) as listed in Table 1. Despite the availability of so many tools and databases for 16S rRNA analysis, QIIME is considered to be the “gold standard” (Nilakanta *et al.*, 2014). Widely used rRNA databases include Greengenes (16S) (DeSantis *et al.*, 2006), Ribosomal Database Project (16S) (Cole *et al.*, 2006, 2008; Maidak *et al.*, 1996), Silva (16S & 18S) (Pruesse *et al.*, 2007; Quast *et al.*, 2012) and UNITE (ITS) (Köljalg *et al.*, 2013).

Pre-processing of reads for amplicon analysis

The raw files generated from the next generation sequencing platform are subjected to demultiplexing, adapter trimming and quality filtration (Plummer *et al.*, 2015). PCR product chimera detection and removal is carried out using UCHIME algorithm (Sinclair *et al.*, 2015).

OTU picking and taxonomic assignment

OTU picking groups similar sequences by clustering or a similarity-based method. OTU picking in the most popular tool QIIME is performed using the UCLUST program. The UCLUST program uses the USEARCH algorithm to assign the sequences to clusters (Edgar, 2010). Each OTU represents a cluster of sequences with similarity greater than a threshold, typically 97%–98%, which is then assigned to a corresponding taxonomic group. There are various OTU picking strategies: (1) *De novo*, wherein the reads are clustered without reference to known sequences. (2) Closed-reference, in which the reads are clustered based on the alignment to a reference database or (3) Open reference method, which clusters reads against a reference database and also clusters unaligned reads using a *de novo* approach. All these methods are incorporated in QIIME (Oulas *et al.*, 2015). The taxonomical diversity can be represented as an abundance histogram. Further analysis can be done with decomposition methods such as Principal Coordinate Analysis (PCoA) or Canonical Correlation Analysis (Johnson and Wichern, 2014). The taxonomical diversity can be represented using Krona, Fig. 2 shows one layer of a Krona plot at the Proteobacteria level.

Statistical Analysis

The taxonomic tree in Newick format can be obtained from QIIME, and can be visualised using any tree display tool, for instance FigTree (FigTree is available at: <http://tree.bio.ed.ac.uk/software/figtree/>). Alpha diversity measures variability within a single population, which measures richness, dominance, and evenness. Other diversity metrics include, for example, Phylogenetic Diversity (PD), Chao (1984) Shannon entropy (Gorelick, 2006), among others. Rarefaction analysis is used to assess the coverage of the microbial community in the sample. Rarefaction curves plot the sample size versus the estimated number of genera (Jaenicke *et al.*, 2011). Beta diversity is the diversity across many populations or samples, which is calculated using various matrices, such as unweighted and weighted UniFrac (Lozupone *et al.*, 2006) and PCoA (Principal Coordinate Analysis). It includes absolute or relative overlap between samples to estimate the taxa shared among them. Calculation of alpha and beta diversity is supported by QIIME.

Metagenomics Phylogenetic Analysis

The taxonomic methods can provide information at specific taxonomic hierarchy, such as phylum, class, order, family, genus and species (Darling *et al.*, 2014), whereas phylogenetic approaches help in identifying species and novel lineages at taxonomic levels (Darling *et al.*, 2014). Different tools used for the phylogenetic analysis of metagenomes are AmphoraNet (Kerepesi *et al.*, 2014), TIPP (taxonomic identification and phylogenetic profiling) (Nguyen *et al.*, 2014), and Phylosift (Darling *et al.*, 2014), among others.

The PhyloSift database includes a set of “elite” gene families of Bacteria and Archaea, and also includes additional four sets of gene families *i.e.* 16S and 18S ribosomal RNA genes, mitochondrial gene families, eukaryote-specific gene families, and viral gene families. The metagenome reads are searched against the defined set of gene database to predict taxonomy. The query sequences are aligned with the reference genes in the database. The sequences are assigned on a phylogenetic tree using Pplacer and taxonomic assignment is carried out (Darling *et al.*, 2014). This approach is based on statistical phylogenetic models *i.e.* Bayesian hypothesis testing and OTU-free analyses (Darling *et al.*, 2014). The AmphoraNet webserver uses the AMPHORA2 workflow for phylogenetic diversity analyses of metagenomic data (Wu and Eisen, 2008), where query sequences are aligned to bacterial and archaeal protein

coding marker genes. Alignment to marker genes is carried out using a hidden Markov model trained on a reference database of fully sequenced bacterial genomes (Wu M and Eisen J 2008). The metagenomic reads are used as an input in AmphoraNet to search, align and mask the data against the HMM trained model, followed by phylogenetic analysis using RaxML on the masked metagenomics sequence (Kembel *et al.*, 2011). Phylogenetic analysis is one of the critical steps in analysing metagenomics data, with applications in improved classification of sequences using phylogenetic methods, functional prediction of genes, and improved identification of OTUs, among others (Darling *et al.*, 2014).

Functional analysis

To predict the functional composition of microbial communities from the 16S profile, PICRUSt (Langille *et al.*, 2013) can be used. It uses an extended ancestral-state reconstruction algorithm, which predicts the gene families and then combines gene families to estimate the composite metagenome. The annotation of the predicted gene family counts are obtained from orthologous groups of gene families, KOGs, COGs, NOGs, or Pfam families (Langille *et al.*, 2013).

Metatranscriptomics/Metaproteomics Analysis

Metatranscriptomics allows the characterization and functionality analysis of the microbiome under study. It provides RNASeq-based measures of gene expression and regulation in the microbial community. The analysis used for metatranscriptomics is the same as in metagenomics (Simon and Daniel, 2011). The analysis can be carried in two approaches (1) *de novo* and (2) reference based. There are many *de novo* assembly tools, as listed in Table 1. Read alignment is carried out using standard short-read mapping tools such as bowtie2, or BWA. To study differentially expressed genes, statistical tools are used such as DESeq2 and edgeR (Bikel *et al.*, 2015).

Metaproteomics is the study of the total proteome expressed in the microbial community. This can be used to study metabolic pathways, and to identify enzymes present in unculturable microbes and communities (Simon and Daniel, 2011; Madhavan *et al.*, 2017).

The metaproteomic approach has four steps: sample collection, protein extraction, purification and fractionation; mass spectrometric (MS) analysis; and finally protein identification and further bioinformatics analyses. Raw data (SDS PAGE, MALDI-TOF-TOF, MS (LC-ESI-MS/MS)) are interpreted using software packages such as Mascot, SEQUEST and *de novo* analysis software such as PEAKS to identify peptides and proteins. There are two approaches for this: direct mass spectra based or *de novo* peptide sequence based, each of which is followed by a quantitative step, along with visualization of the complex functional information (Wang *et al.*, 2014a,b).

Application

Metagenomics has a wide range of applications from clinical to environmental samples, from food safety to industrial waste, and also has the ability to identify pathogens. For example: Lettuce has an immense number of viruses on it, which can infect various hosts, such as humans and animals. Metagenomics can detect human and animal viruses on lettuce samples, which can identify viral contamination of the green leaves in the field (Aw *et al.*, 2016). These approaches help to determine the source of microbial and viral contamination, and can be used to identify the critical steps where such contamination occurs, and to implement improved processes to ensure food quality and safety (Doyle *et al.*, 2017).

Metagenomics provides information about the diversity of organisms in environmental samples and has provided insights in industrial research. Functional metagenomics has been used for identification of several biocatalysts, which are available in the market, for example: laccases, esterases and nitrile hydratase (Fernández-Arrojo *et al.*, 2010). Novel cellulases with improved enzymatic characteristics have been identified; cellulose is an important component of plant biomass, and is the most abundant polymer in nature (Fang *et al.*, 2010). The use of metagenomics, metatranscriptomics and metaproteomics approaches enhance enzyme discovery and can be used to efficiently screen for highly active enzymes (Madhavan *et al.*, 2017). Recent studies (Devarapalli and Kumavath, 2015), have stated that metagenomics can be used as a bioremediation tool; in comparison with other approaches of bioremediation, metagenomics gave better degrading ratios. Metagenomics study help in identifying different widespread microorganism and their respective functions in polluted environment; these microorganisms are the best tools in nature to degrade toxic pollutants (Devarapalli and Kumavath, 2015).

Metagenomics has been used in clinical diagnostics. For example, metagenomic approaches have been used for identifying novel noninvasive diagnostic bacterial biomarkers from fecal samples, for diagnosis of colorectal cancer (Liang *et al.*, 2017). The 16S rRNA is a powerful approach for developing diagnostics, for example, for detecting bacterial bloodstream infections (bBSI) (Decuyper *et al.*, 2016). Viral metagenomics has the power to identify the root cause of novel epidemic diseases (Mokili *et al.*, 2012). Metagenomics is also used in medical or forensic investigations, and to solve challenges in the field of medicine, agriculture and ecology. Major applications of metaproteomics have included investigation of acid mine drainage biofilms, activated sludge, soil, human gut microbiota, and other environmental samples (Wilmes *et al.*, 2015).

Discussion and Future Direction

Microbes survive in a vast range of environmental conditions and are found throughout nature. Most of them are unculturable, so they are difficult to identify by traditional methods. Metagenomics is a molecular tool used to analyse DNA acquired from

environmental samples, in order to study the community of microorganisms present, without the necessity of obtaining pure cultures. It is most often applied to bacteria, but can also be applied to archaea, viruses, and eukaryotes (Morgan and Huttenhower, 2012). Metagenomics provides a platform for studying the physiology and genetics of uncultured organisms. It can refine the approach of genomics and accelerates the rate of gene discovery. Metagenomics has a future in diagnostics, in which it could be applied to any specimen (for example plant, human, and *etc.*) to detect pathogens, such as bacteria, viruses, fungi, and parasites. Metagenomics in diagnostics has various challenges as there is wide range of abundance of different taxa in particular samples (Pallen, 2014). Identifying the causative agent of a new epidemic is one of the most important steps for effective response to disease outbreaks. Traditionally, virus discovery required propagation of the virus in cell culture, which has resulted in limited knowledge of viruses. Study of viral metagenomes suggests that the field of virology has explored much less than 1% of viral taxonomic diversity (Mokili *et al.*, 2012). The study of gene expression and protein production in the microbial community complement metagenomic analysis. Metatranscriptomics and metaproteomics approaches provide powerful tools for understanding the functional dynamics of microbial communities (Simon and Daniel, 2011).

Metaproteomics studies have not been used to their full potential in comparison to metagenomics and metatranscriptomics. Metaproteomics studies can provide deep insights into diverse biological questions regarding health and disease. Emerging technologies should improve the ability of metaproteomic approaches to handle the wide dynamic ranges of different metaproteomic samples, and also help in understanding protein modification (Wilmes *et al.*, 2015). Metaproteomics has emerged as complementary approach to metagenomics, which aims at the characterization of proteins from environmental microbiota (Petriz and Franco, 2017).

See also: Comparative Genomics Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Sequence Analysis. Sequence Composition. Whole Genome Sequencing Analysis

References

- Aineberg, J., Bjarnason, B.S., De Bruijn, I., *et al.*, 2014. Binning metagenomic contigs by coverage and composition. *Nature Methods* 11 (11), 1144–1146.
- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene Ontology: Tool for the unification of biology. *Nature Genetics* 25 (1), 25.
- Aw, T.G., Wengert, S., Rose, J.B., 2016. Metagenomic analysis of viruses associated with field-grown and retail lettuce identifies human and animal viruses. *International Journal of Food Microbiology* 223, 50–56.
- Bateman, A., Coin, L., Durbin, R., *et al.*, 2004. The Pfam protein families database. *Nucleic Acids Research* 32 (suppl_1), [D138–D141].
- Bikel, S., Valdez-Lara, A., Cornejo-Granados, F., *et al.*, 2015. Combining metagenomics, metatranscriptomics and viromics to explore novel microbial interactions: Towards a systems-level understanding of human microbiome. *Computational and Structural Biotechnology Journal* 13, 390–401.
- Boisvert, S., Raymond, F., Godzaridis, É., Laviolette, F., Corbeil, J., 2012. Ray Meta: Scalable de novo metagenome assembly and profiling. *Genome Biology* 13 (12), R122.
- Bose, T., Haque, M.M., Reddy, C.V.S.K., Mande, S.S., 2015. COGNIZER: A framework for functional annotation of metagenomic datasets. *PLoS One* 10 (11), e0142102.
- Brady, A., Salzberg, S.L., 2009. Phymm and PhymmBL: Metagenomic phylogenetic classification with interpolated Markov models. *Nature Methods* 6 (9), 673–676.
- Bromberg, J.S., Fricke, W.F., Brinkman, C.C., Simon, T., Mongodin, E.F., 2015. Microbiota [mdash] implications for immunity and transplantation. *Nature Reviews Nephrology* 11 (6), 342–353.
- Caporaso, J.G., Kuczynski, J., Stombaugh, J., *et al.*, 2010. QIIME allows analysis of high-throughput community sequencing data. *Nature Methods* 7 (5), 335–336.
- Chakravorty, S., Helb, D., Burday, M., Connell, N., Alland, D., 2007. A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria. *Journal of Microbiological Methods* 69 (2), 330–339.
- Chao, A., 1984. Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of Statistics*, 265–270.
- Chong, C.W., Pearce, D.A., Convey, P., Yew, W.C., Tan, I.K.P., 2012. Patterns in the distribution of soil bacterial 16S rRNA gene sequences from different regions of Antarctica. *Geoderma* 181, 45–55.
- Cole, J.R., Chai, B., Farris, R.J., *et al.*, 2006. The ribosomal database project (RDP-II): Introducing myRDP space and quality controlled public data. *Nucleic Acids Research* 35 (suppl_1), D169–D172.
- Cole, J.R., Wang, Q., Cardenas, E., *et al.*, 2008. The Ribosomal Database Project: Improved alignments and new tools for rRNA analysis. *Nucleic Acids Research* 37 (suppl_1), D141–D145.
- Cole, J.R., Wang, Q., Fish, J.A., *et al.*, 2013. Ribosomal Database Project: Data and tools for high throughput rRNA analysis. *Nucleic Acids Research* 42 (D1), D633–D642.
- Cuadros-Orellana, S., Leite, L.R., Smith, A., *et al.*, 2013. Assessment of fungal diversity in the environment using metagenomics: A decade in review. *Fungal Genomics & Biology* 3 (2), 1.
- Darling, A.E., Jospin, G., Lowe, E., *et al.*, 2014. PhyloSift: phylogenetic analysis of genomes and metagenomes. *PeerJ* 2, e243.
- Decuyper, S., Meehan, C.J., Van Puyvelde, S., *et al.*, 2016. Diagnosis of bacterial bloodstream infections: A 16S metagenomics approach. *PLoS Neglected Tropical Diseases* 10 (2), e0004470.
- DeSantis, T.Z., Hugenholtz, P., Larsen, N., *et al.*, 2006. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology* 72 (7), 5069–5072.
- Dethlefsen, L., Huse, S., Sogin, M.L., Relman, D.A., 2008. The pervasive effects of an antibiotic on the human gut microbiota, as revealed by deep 16S rRNA sequencing. *PLoS Biology* 6 (11), e280.
- Devarapalli, P., Kumavath, R.N., 2015. Metagenomics – A Technological Drift in Bioremediation. *Advances in Bioremediation of Wastewater and Polluted Soil*. Intech.
- Dinsdale, E.A., Edwards, R.A., Hall, D., *et al.*, 2008. Functional metagenomic profiling of nine biomes. *Nature* 452 (7187), 629.
- Doyle, C.J., O'Toole, P.W., Cotter, P.D., 2017. Metagenome-based surveillance and diagnostic approaches to studying the microbial ecology of food production and processing environments. *Environmental Microbiology*.
- Edgar, R.C., 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26 (19), 2460–2461.
- Fang, Z., Fang, W., Liu, J., *et al.*, 2010. Cloning and characterization of a β -glucosidase from marine microbial metagenome with excellent glucose tolerance. *Journal of Microbiology Biotechnology* 20 (9), 1351–1358.
- Fernández-Arrojo, L., Guazzaroni, M.E., López-Cortés, N., Belouqui, A., Ferrer, M., 2010. Metagenomic era for biocatalyst identification. *Current Opinion in Biotechnology* 21 (6), 725–733.

- Frias-Lopez, J., Shi, Y., Tyson, G.W., *et al.*, 2008. Microbial community gene expression in ocean surface waters. *Proceedings of the National Academy of Sciences* 105 (10), 3805–3810.
- Glass, E.M., Wilkening, J., Wilke, A., Antonopoulos, D., Meyer, F., 2010. Using the metagenomics RAST server (MG-RAST) for analyzing shotgun metagenomes. *Cold Spring Harbor Protocols* 2010 (1), pdb-prot5368.
- Gorelick, R., 2006. Combining richness and abundance into a single diversity index using matrix analogues of Shannon's and Simpson's indices. *Ecography* 29 (4), 525–530.
- Haft, D.H., Selengut, J.D., White, O., 2003. The TIGRFAMs database of protein families. *Nucleic Acids Research* 31 (1), 371–373.
- Haider, B., Ahn, T.H., Bushnell, B., *et al.*, 2014. Omega: An Overlap-graph de novo Assembler for Metagenomics. *Bioinformatics* 30 (19), 2717–2722.
- Handelsman, J., Rondon, M.R., Brady, S.F., Clardy, J., Goodman, R.M., 1998. Molecular biological access to the chemistry of unknown soil microbes: A new frontier for natural products. *Chemistry & Biology* 5 (10), R245–R249.
- Hilton, S.K., Castro-Nallar, E., Pérez-Losada, M., *et al.*, 2016. Metatranscriptomic and metagenomic approaches vs. culture-based techniques for clinical pathology. *Frontiers in Microbiology* 7.
- Hunter, S., Apweiler, R., Attwood, T.K., *et al.*, 2008. InterPro: The integrative protein signature database. *Nucleic Acids Research* 37 (suppl_1), D211–D215.
- Huson, D.H., Auch, A.F., Qi, J., Schuster, S.C., 2007. MEGAN analysis of metagenomic data. *Genome Research* 17 (3), 377–386.
- Huson, D.H., Weber, N., 2013. Microbial community analysis using MEGAN. *Methods in Enzymology* 531, 465–485.
- Hyatt, D., Chen, G.L., LoCascio, P.F., *et al.*, 2010. Prodigal: Prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* 11 (1), 119.
- Igai, K., Itakura, M., Nishijima, S., *et al.*, 2016. Nitrogen fixation and nifH diversity in human gut microbiota. *Scientific Reports* 6.
- Jaenicke, S., Ander, C., Bekel, T., *et al.*, 2011. Comparative and joint analysis of two metagenomic datasets from a biogas fermenter obtained by 454-pyrosequencing. *PLoS one* 6 (1), e14519.
- Jensen, L.J., Julien, P., Kuhn, M., *et al.*, 2007. eggNOG: Automated construction and annotation of orthologous groups of genes. *Nucleic Acids Research* 36 (suppl_1), D250–D254.
- Johnson, R.A., Wichern, D.W., 2014. *Applied Multivariate Statistical Analysis*, 4. Piscataway, NJ: Prentice-Hall.
- Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M., 2009. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research* 38 (suppl_1), D355–D360.
- Kelley, D.R., Liu, B., Delcher, A.L., Pop, M., Salzberg, S.L., 2011. Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research* 40 (1), [e9–e9].
- Kelley, D.R., Salzberg, S.L., 2010. Clustering metagenomic sequences with interpolated Markov models. *BMC Bioinformatics* 11 (1), 544.
- Kembel, S.W., Eisen, J.A., Pollard, K.S., Green, J.L., 2011. The phylogenetic diversity of metagenomes. *PLoS One* 6 (8), e23214.
- Kerepesi, C., Banky, D., Grolmusz, V., 2014. AmphoraNet: the webserver implementation of the AMPHORA2 metagenomic workflow suite. *Gene* 533 (2), 538–540.
- Kõljalg, U., Larsson, K.H., Abarenkov, K., *et al.*, 2005. UNITE: A database providing web-based methods for the molecular identification of ectomycorrhizal fungi. *New Phytologist* 166 (3), 1063–1068.
- Kõljalg, U., Nilsson, R.H., Abarenkov, K., *et al.*, 2013. Towards a unified paradigm for sequence-based identification of fungi. *Molecular Ecology* 22 (21), 5271–5277.
- Langille, M.G., Zaneveld, J., Caporaso, J.G., *et al.*, 2013. Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nature Biotechnology* 31 (9), 814–821.
- Leung, H.C., Yiu, S.M., Yang, B., *et al.*, 2011. A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. *Bioinformatics* 27 (11), 1489–1495.
- Liang, Q., Chiu, J., Chen, Y., *et al.*, 2017. Fecal bacteria act as novel biomarkers for noninvasive diagnosis of colorectal cancer. *Clinical Cancer Research* 23 (8), 2061–2070.
- Li, D., Liu, C.M., Luo, R., Sadakane, K., Lam, T.W., 2015. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 31 (10), 1674–1676.
- Lozupone, C., Hamady, M., Knight, R., 2006. UniFrac—an online tool for comparing microbial community diversity in a phylogenetic context. *BMC Bioinformatics* 7 (1), 371.
- Luo, R., Liu, B., Xie, Y., *et al.*, 2012. SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. *Gigascience* 1 (1), 18.
- Madhavan, A., Sindhu, R., Parameswaran, B., Sukumaran, R.K., Pandey, A., 2017. Metagenome Analysis: A Powerful Tool for Enzyme Bioprospecting. *Applied Biochemistry and Biotechnology*. 1–16.
- Maidak, B.L., Olsen, G.J., Larsen, N., *et al.*, 1996. The ribosomal database project (RDP). *Nucleic Acids Research* 24 (1), 82–85.
- Markowitz, V.M., Chen, I.M.A., Chu, K., *et al.*, 2013. IMG/M 4 version of the integrated metagenome comparative analysis system. *Nucleic Acids Research* 42 (D1), D568–D573.
- Menzel, P., Ng, K.L., Krogh, A., 2016. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications* 7.
- Meyer, F., Paarmann, D., D'Souza, M., *et al.*, 2008. The metagenomics RAST server – a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics* 9 (1), 386.
- Mokili, J.L., Rohwer, F., Dutilh, B.E., 2012. Metagenomics and future perspectives in virus discovery. *Current opinion in Virology* 2 (1), 63–77.
- Morgan, X.C., Huttenhower, C., 2012. Human microbiome analysis. *PLOS Computational Biology* 8 (12), e1002808.
- Nagarajan, N., Cook, C., Di Bonaventura, M., *et al.*, 2010. Finishing genomes with limited resources: Lessons from an ensemble of microbial genomes. *BMC Genomics* 11 (1), 242.
- Namiki, T., Hachiya, T., Tanaka, H., Sakakibara, Y., 2012. MetaVelvet: An extension of Velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research* 40 (20), [e155–e155].
- Nguyen, N.P., Mirarab, S., Liu, B., Pop, M., Warnow, T., 2014. TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics* 30 (24), 3548–3555.
- Nilakanta, H., Drews, K.L., Firrell, S., Foulkes, M.A., Jablonski, K.A., 2014. A review of software for analyzing molecular sequences. *BMC Research Notes* 7 (1), 830.
- Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A., 2017. metaSPAdes: A new versatile metagenomic assembler. *Genome Research* 27 (5), 824–834.
- Ogata, H., Goto, S., Sato, K., *et al.*, 1999. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic acids Research* 27 (1), 29–34.
- Ondov, B.D., Bergman, N.H., Phillippy, A.M., 2011. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics* 12 (1), 385.
- Oulas, A., Pavloudi, C., Polymenakou, P., *et al.*, 2015. Metagenomics: Tools and insights for analyzing next-generation sequencing data derived from biodiversity studies. *Bioinformatics and Biology Insights* 9, 75.
- Overbeek, R., Begley, T., Butler, R.M., *et al.*, 2005. The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Research* 33 (17), 5691–5702.
- Oyserman, B.O., Noguera, D.R., del Rio, T.G., Tringe, S.G., McMahon, K.D., 2016. Metatranscriptomic insights on gene expression and regulatory controls in *Candidatus Accumulibacter phosphatis*. *The ISME Journal* 10 (4), 810.
- Pallen, M.J., 2014. Diagnostic metagenomics: Potential applications to bacterial, viral and parasitic infections. *Parasitology* 141 (14), 1856–1862.
- Peng, Y., Leung, H.C., Yiu, S.M., Chin, F.Y., 2012. IDBA-UD: A de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* 28 (11), 1420–1428.
- Petritz, B.A., Franco, O.L., 2017. Metaproteomics as a Complementary Approach to Gut Microbiota in Health and Disease. *Frontiers in Chemistry* 5.
- Plummer, E., Twin, J., Bulach, D.M., Garland, S.M., Tabrizi, S.N., 2015. A comparison of three bioinformatics pipelines for the analysis of preterm gut microbiota using 16S rRNA gene sequencing data. *Journal of Proteomics & Bioinformatics* 8 (12), 283.
- Powell, S., Forslund, K., Szklarczyk, D., *et al.*, 2014. eggNOG v4.0: Nested orthology inference across 3686 organisms. *Nucleic Acids Research* 42 (D1), D231–D239.
- Pruesse, E., Quast, C., Knittel, K., *et al.*, 2007. SILVA: A comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Research* 35 (21), 7188–7196.
- Quail, M.A., Smith, M., Coupland, P., *et al.*, 2012. A tale of three next generation sequencing platforms: Comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics* 13 (1), 341.

- Quast, C., Pruesse, E., Yilmaz, P., *et al.*, 2012. The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Research* 41 (D1), D590–D596.
- Quast, C., Pruesse, E., Yilmaz, P., *et al.*, 2013. The 658 SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Research* 41, D590–D596. [660].
- Ranjan, R., Rani, A., Metwally, A., McGee, H.S., Perkins, D.L., 2016. Analysis of the microbiome: Advantages of whole genome shotgun versus 16S amplicon sequencing. *Biochemical and Biophysical Research Communications* 469 (4), 967–977.
- Rho, M., Tang, H., Ye, Y., 2010. FragGeneScan: Predicting genes in short and error-prone reads. *Nucleic Acids Research* 38 (20), [e191–e191].
- Sato, K., Sakakibara, Y., 2014. MetaVelvet-SL: An extension of the Velvet assembler to a de novo metagenomic assembler utilizing supervised learning. *DNA Research* 22 (1), 69–77.
- Schloss, P.D., Westcott, S.L., Ryabin, T., *et al.*, 2009. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology* 75 (23), 7537–7541.
- Seemann, T., 2014. Prokka: Rapid prokaryotic genome annotation. *Bioinformatics* 30 (14), 2068–2069.
- Sharpton, T.J., 2014. An introduction to the analysis of shotgun metagenomic data. *Frontiers in Plant Science* 5.
- Simon, C., Daniel, R., 2011. Metagenomic analyses: Past and future trends. *Applied and Environmental Microbiology* 77 (4), 1153–1161.
- Sinclair, L., Osman, O.A., Bertilsson, S., Eiler, A., 2015. Microbial community composition and diversity via 16S rRNA gene amplicons: Evaluating the illumina platform. *PIOS One* 10 (2), e0116955.
- Tartar, A., Wheeler, M.M., Zhou, X., *et al.*, 2009. Parallel metatranscriptome analyses of host and symbiont gene expression in the gut of the termite *Reticulitermes flavipes*. *Biotechnology for Biofuels* 2 (1), 25.
- Tatusov, R.L., Fedorova, N.D., Jackson, J.D., *et al.*, 2003. The COG database: An updated version includes eukaryotes. *BMC Bioinformatics* 4 (1), 41.
- Tatusov, R.L., Galperin, M.Y., Natale, D.A., Koonin, E.V., 2000. The COG database: A tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research* 28 (1), 33–36.
- Treangen, T.J., Koren, S., Sommer, D.D., *et al.*, 2013. MetAMOS: A modular and open source metagenomic assembly and analysis pipeline. *Genome Biology* 14 (1), R2.
- Tringe, S.G., Von Mering, C., Kobayashi, A., *et al.*, 2005. Comparative metagenomics of microbial communities. *Science* 308 (5721), 554–557.
- Tyson, G.W., Chapman, J., Hugenholtz, P., Allen, E.E., 2004. Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428 (6978), 37.
- Vasar, M., Andreson, R., Davison, J., *et al.*, 2017. Increased sequencing depth does not increase captured diversity of arbuscular mycorrhizal fungi. *Mycorrhiza*. 1–13.
- Venter, J.C., Remington, K., Heidelberg, J.F., *et al.*, 2004. Environmental genome shotgun sequencing of the Sargasso Sea. *Science* 304 (5667), 66–74.
- Vieites, J.M., Guazzaroni, M.E., Belouqui, A., Golyshin, P.N., Ferrer, M., 2008. Metagenomics approaches in systems microbiology. *FEMS Microbiology Reviews* 33 (1), 236–255.
- Wang, D.Z., Xie, Z.X., Zhang, S.F., 2014a. Marine metaproteomics: Current status and future directions. *Journal of Proteomics* 97, 27–35.
- Wang, Y., Leung, H.C.M., Yiu, S.M., Chin, F.Y.L., 2014b. MetaCluster-TA: Taxonomic annotation for metagenomic data based on assembly-assisted binning. *BMC Genomics* 15 (1), S12.
- Weinstock, G.M., 2012. Genomic approaches to studying the human microbiota. *Nature* 489 (7415), 250.
- Wheeler, D.L., Barrett, T., Benson, D.A., *et al.*, 2007. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 36 (suppl_1), D13–D21.
- Wilmes, P., Heintz-Buschart, A., Bond, P.L., 2015. A decade of metaproteomics: Where we stand and what the future holds. *Proteomics* 15 (20), 3409–3417.
- Woese, C.R., 1987. Bacterial evolution. *Microbiological Reviews* 51 (2), 221.
- Wu, M., Eisen, J., 2008. A simple, fast, and accurate method of phylogenomic inference. *Genome Biology* 9, R151.
- Zampieri, E., Chiappello, M., Daghino, S., Bonfante, P., Mello, A., 2016. Soil metaproteomics reveals an inter-kingdom stress response to the presence of black truffles. *Scientific Reports* 6, 25773.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18 (5), 821–829.
- Zhang, Y., Sun, J., Mu, H., Lun, J.C., Qiu, J.W., 2017. Molecular pathology of skeletal growth anomalies in the brain coral *Platygyra carnosus*: A meta-transcriptomic analysis. *Marine Pollution Bulletin*.
- Zhu, W., Lomsadze, A., Borodovsky, M., 2010. Ab initio gene identification in metagenomic sequences. *Nucleic Acids Research* 38 (12), [e132–e132].

Further Reading

- Blatter, M.C., Gerritsen, V.B., Palagi, P.M., Bougueleret, L., Xenarios, I., 2016. The Metagenomic Pizza: A simple recipe to introduce bioinformatics to the layman. *EMBnet journal* 22, e864.
- Darling, A.E., Jospin, G., Lowe, E., *et al.*, 2014. PhyloSift: Phylogenetic analysis of genomes and metagenomes. *PeerJ* 2, e243.
- DeLong, E., 2013. *Microbial Metagenomics, Metatranscriptomics, and Metaproteomics*, 531. Academic Press.
- Dudhagara, P., Bhavsar, S., Bhagat, C., *et al.*, 2015. Web resources for metagenomics studies. *Genomics, Proteomics & Bioinformatics* 13 (5), 296–303.
- Howe, A., Chain, P.S., 2015. Challenges and opportunities in understanding microbial communities with metagenome assembly (accompanied by IPython Notebook tutorial). *Frontiers in Microbiology* 6.
- Kembel, S.W., Eisen, J.A., Pollard, K.S., Green, J.L., 2011. The phylogenetic diversity of metagenomes. *PLoS One* 6 (8), e23214.
- Kerepesi, C., Banky, D., Grolmusz, V., 2014. AmphoraNet: The webserver implementation of the AMPHORA2 metagenomic workflow suite. *Gene* 533 (2), 538–540.
- Kuczynski, J., Stombaugh, J., Walters, W.A., *et al.*, 2012. Using QIIME to analyze 16S rRNA gene sequences from microbial communities. *Current Protocols in Microbiology*. 1E–5.
- Marco, D. (Ed.), 2011. *Metagenomics: Current Innovations and Future Trends*. Horizon Scientific Press.
- Wu, M., Eisen, J., 2008. A simple, fast, and accurate method of phylogenomic inference. *Genome Biology* 9, R151.
- Nguyen, N.P., Mirarab, S., Liu, B., Pop, M., Warnow, T., 2014. TIPP: Taxonomic identification and phylogenetic profiling. *Bioinformatics* 30 (24), 3548–3555.

Relevant Websites

- <https://bioinformatics.ca/workshops/2015/analysis-metagenomic-data-2015>
Analysis of Metagenomic DataS.
- https://ngs.csr.uky.edu/sites/default/files/Class_9_QIIME.pdf
Analyzing Metagenomic Data with QIIME.
- <http://tree.bio.ed.ac.uk/software/figtree/>
FigTree.
- <http://qiime.org/1.2.1/tutorials/tutorial.html>
Qiime overview Tutorial.

Biographical Sketch



Arpita Ghosh is a Manager of Bioinformatics at Eurofins Genomics India Pvt Ltd/ Eurofins Clinical Genetics India Pvt. Ltd. She is also a PhD student at Perdana University, Malaysia. She has a rich research experience of more than seven years in various fields of genomics and clinical genetics data analysis using Next Generation Sequencing technology. She has published more than 15 international publications in different areas of genomics and clinical genetics. She has been working on cutting edge applications of genomics like amplicon based metagenomics (16S, ITS), whole metagenome, WGS, RNASeq, small RNA, GBS and Tilling, among others. Her current research is on virus metagenomics and to identify novel genes from viral genomes. Her primary research interest is in the field of genomics, application heat of which can be felt invarious fields such as molecular diagnostics and the study of the environment.



Aditya Mehta has more than five years of Industrial research experience in various fields of applied genomics and translational genomics data analysis using various next generation sequencing platforms. He has worked on large scale data analysis projects like RNASeq, whole genome, Metagenomics and 16S/18S analysis using high throughput sequencing data. He has extensive experience in analysing metagenomics and 16S rRNA data generated on Illumina platform. He has carried out large scale data analysis of various metagenomics samples like soil, water and the human gut and identified microbial diversity, their relative abundance in sample and also deciphering the gene functions. He has in depth knowledge of various metagenomics tools, software and databases that are widely used for microbial data analysis and taxonomical identification from experimental sample. He has designed new strategic approaches and workflow steps for microbial data analysis, including further downstream analysis.



Dr. Mohammad Asif Khan is currently appointed as the Director of the Perdana University Center for Bioinformatics (PU-CBi), Malaysia, and a Visiting Scientist for the Department of Pharmacology and Molecular Sciences, Johns Hopkins University School of Medicine (JHUSOM), USA. His research interests are in the area of biological data warehousing and applications of bioinformatics to the study of immune responses, vaccines, venom toxins, drug design, and disease biomarkers. He has contributed in the development of several novel bioinformatics methodologies, tools, and specialized databases, and currently has three patent applications granted.

Integrative Analysis of Multi-Omics Data

Lokesh P Tripathi, Tsuyoshi Esaki, Mari N Itoh, Yi-An Chen, and Kenji Mizuguchi, National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biomolecular interactions among elementary units of a living system such as DNA, RNA, proteins, lipids and also small molecule metabolites make up the different organisational levels of the cellular networks such as microRNA (miRNA)-target interactions (MTIs), protein-protein interactions (PPIs), transcription factor (TF)-target gene interactions and protein-chemical compound (small molecule) interactions (PCIs); such interactions underlie the complexity and functioning of all biological processes. The key step in biology is to understand how these different components come together and how they interact with each other and define the emergent behaviour of the living systems.

The quest to uncover the mechanisms underlying complex biological processes and the proliferation of various high-throughput “omics” experiments have resulted in an unprecedented surge in the diversity, volume and complexity of *genomics*, *transcriptomics*, *proteomics* and *metabolomics* data among others. *Genomics* data provide an overview of the complete set of genetic instructions provided by the DNA including the variants within; *transcriptomics* data provide insights into gene expression patterns; *proteomics* data examine the cellular dynamics of proteins and their interactions and *metabolomics* can be used to determine the differences between the levels of thousands of molecules between healthy and diseased conditions. The availability of such data have, therefore, opened up a myriad of opportunities for new biological discoveries.

Typically, omics datasets are analysed separately using varied statistical and computational methods and approaches. The analysis of PPI Networks (PPIN) (Huttlin *et al.*, 2017, 2015; Rolland *et al.*, 2014), genetic interactions (Boucher and Jenna, 2013), gene co-expression networks (van Dam *et al.*, 2017; Serin *et al.*, 2016) and metabolic interaction networks (Krumisiek *et al.*, 2016), among others, have provided valuable biological insights into the set-up of the cellular machinery. However, the single-datatype analyses only offer glimpses into one of the many layers that govern the cellular machinery; they do not take into account the relationships and interactions between different biological entities across different organisational layers. Therefore, combining different types of omics data into a single analytical framework provides an opportunity to simultaneously examine the multiple cellular organisational layers, understand the complex interactions between them and gain deeper insights into how their combined influence determines the behaviour of the biological systems (Yan *et al.*, 2017; Civelek and Lusi, 2014; Zhu *et al.*, 2012). Such integration can also facilitate a better understanding of how living systems adapt and modulate their responses to different environments and also help reduce the noise and false positives emanating from single source data sets.

A systems biology-based multi-omics data analysis approach seeks to integrate disparate omics data with bioinformatics tools for data manipulation and pattern discovery into an integrative data analysis pipeline to derive conceptual systems level models of genes and biological processes. Such integrative models can empower the researchers to predict complex traits, understand genotype-phenotype correlations, gain the knowledge of biological pathways involved in different traits and diseases and therefore, help prioritise biologically and therapeutically relevant genes and proteins (Krumisiek *et al.*, 2016; Koczynski *et al.*, 2017; Sun and Hu, 2016; van Kampen and Moerland, 2016; Fukushima *et al.*, 2014; Berger *et al.*, 2013) (Fig. 1).

In this review we will discuss the different types of omics data and how they have been individually analysed. We will then discuss how the analysis of large data sets from different omics platforms has been performed using different approaches such as network modelling and pathway analysis. We finally discuss the future trends in the field.

High Throughput Omics Datatypes and Methodologies

Genomics

Every biological function has its basis in the genetic information of an organism. A better understanding of the genetic determinants of cellular processes including diseases is therefore necessary to gain the knowledge of biological systems, unravel disease mechanisms and develop effective therapeutic approaches. To achieve this, researchers have for long sought to determine the sequences of individual genes of interest. With the advent of high-throughput DNA sequencing technologies, it became possible to sequence the whole genome that is the complete set of DNA within a cell or an organism. Thus, the focus has shifted from investigating individual genes to mapping, annotation and characterisation of the whole genome, a study called *genomics*. The availability and analysis of whole genome sequence data have contributed immensely to the understanding and evolution of gene function in different species and disease mechanisms. Comparative genomics refers to the comparison and analysis of genomes from different species to gain a better understanding of the genomic organisation and conservation and evolution of gene and genomic functions that would have contributed to speciation, adaptation and evolution and to similarity and differences across species (Meadows and Lindblad-Toh, 2017). The advances in genomics technologies, especially the proliferation of low-cost NGS platforms (Levy and Myers, 2016) have led to an abundance of high-quality sequence data that have enhanced the identification of

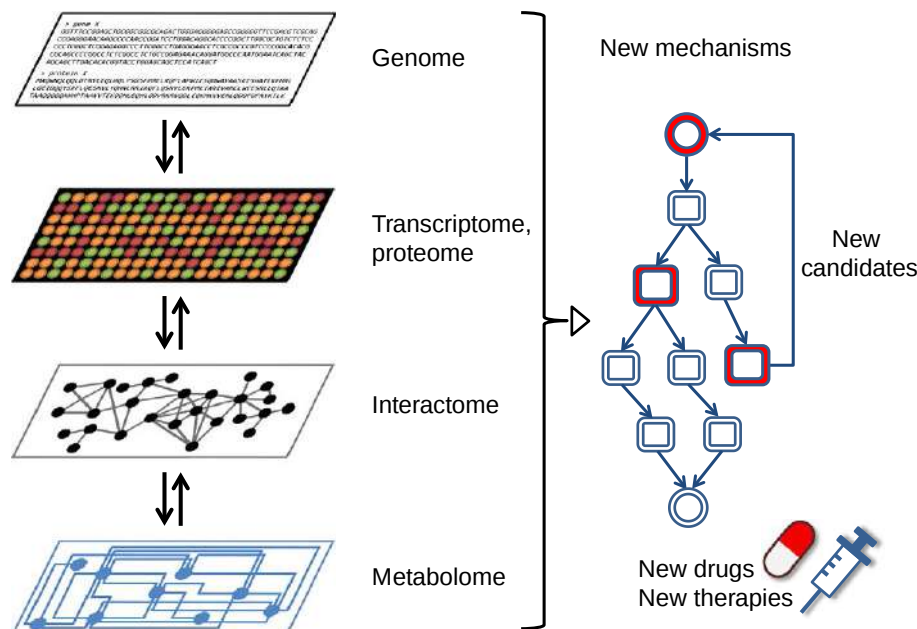


Fig. 1 High-throughput omics technologies provide large scale mapping of heterogenous molecular-level data. Computational methods for integrative multi-omics data analyses aim to uncover relationships across different cellular organisation layers represented by different omics datatypes to hypothesise the pathways and mechanisms underlying complex biological processes, which can then be leveraged to pinpoint biologically and therapeutically relevant genes and proteins (such as potential drug targets).

a large number of sequence and structural variants and are rapidly redefining the understanding of genotype-phenotype relationships (Shameer *et al.*, 2016).

Although genomics sequences have provided unprecedented insights into the gene function and evolution, they do not provide the context of these functional aspects such as spatio-temporal expression of different genes and changes in gene expression in response to different biological cues. To address these questions, genomics revolution was thus, soon followed by an increasing number of high-throughput studies that started mapping genome-wide gene expression profiles in specific cells and tissues in different biological contexts.

Transcriptomics

The genetic blueprint of an organism is encoded within its genome, but is expressed via transcription i.e., copying of the information from DNA into RNA that forms the key intermediate between the DNA and the protein. The sum total of RNA transcript population of a cell is known as Transcriptome and the study of transcriptome is known as *transcriptomics*. The two major and most widely used technologies for transcriptomics analysis are microarray and RNA-seq (Lowe *et al.*, 2017). Microarrays determine the abundance of mRNAs from a given sample, cell or a tissue via their hybridisation to an array of oligonucleotide DNA sequences or ‘probes’ that are immobilised on a solid surface (Lowe *et al.*, 2017; Schulze and Downward, 2001; Slonim and Yanai, 2009). DNA microarrays have emerged as a powerful medium for genomic and biomedical research to investigate the gene expression states underlying different physiological processes, for the characterisation of gene function and biomarker identification (Trevino *et al.*, 2007) and also for whole genome comparisons to investigate genomic variations (Gresham *et al.*, 2008). Another widely used technology to quantify the transcriptome is RNA-Seq that combines the usage of NGS with computational analytical tools to determine the presence and abundance of transcripts within a biological sample (Lowe *et al.*, 2017). Lately, RNA-Seq has emerged as the method of choice for large scale transcriptome mapping and for the identification of allele-specific expression, gene fusions and detecting alternatively spliced variants and non-coding RNAs even at the single-cell level (Hrdlickova *et al.*, 2017; Spies and Ciaudo, 2015).

With the rapid advances in transcriptomics technologies, there have been proliferating attempts at surveying the gene expression landscape across multiple tissues, cells and physiological processes such as mapping of genetic variation and its impact on the gene expression landscape in different tissues by the Genotype-Tissue Expression (GTEx) project (Consortium, 2013) and the Functional annotation of the mammalian genome (FANTOM) project (Consortium *et al.*, 2014). The availability of huge amounts of gene expression profiles from diverse physiological states have permitted the researchers to construct gene co-expression networks to assign functional annotations to different genes and to prioritise candidates for biological and clinical research (van Dam *et al.*, 2017).

Transcriptomics technologies measure the changes in the abundance of mRNA. However, most of the cellular functions are typically regulated at the level of proteins and thus, transcriptomic analyses often provides only an indirect measure of the cellular physiology. This limitation in turn has led to the development of high-throughput methodologies to simultaneously measure and understand the activities of a large number of proteins within a cell or an organism.

Proteomics

Proteins are responsible for a multitude of tasks within the cell. PPIs, for instance, are central to the organisation of macromolecular complexes and many enzymatic reactions that underlie nearly every cellular process. The goal of *proteomics* is to understand how the entire protein population of a cell or organism, i.e., the proteome, their dynamics and their interactions together contributes to and regulates different biological processes, including disease phenotypes. Proteomics can be used to identify and quantify cellular protein levels, characterise PPIs to generate proteome-scale PPI Network (PPIN) maps or to profile post-translational modifications (PTMs) within the cells. Proteomics approaches have, thus, emerged as effective tools in studies involving the investigation of biomedical, nutritional sciences, pathogenic infection, host-pathogen interactions, and molecular biology (Jean Beltran *et al.*, 2017; Moore and Weeks, 2011; Lum and Cristea, 2016).

With rapid advances in analytical capabilities, proteomics is being increasingly used to identify the components of biological systems (Sabido *et al.*, 2012; Aebersold and Mann, 2016). For instance, yeast two-hybrid system (Y2H), where the proteins of interest are fused to the fragmented TF domains and their interaction reconstitutes the functional TF and can be assayed by the downstream expression of a reporter gene (Fields and Song, 1989), is one of the most widely used techniques to detect binary PPIs on a proteome-wide scale. Another versatile experimental technology for proteomics is mass spectrometry (MS), which can detect variable proteins with higher sensitivity and measure the values in higher quantification. Studying temporal proteome alterations has become a popular approach due to the availability of well-established protocols and modern MS instrumentation. The accumulated proteomics data have enabled the researchers to carefully examine the temporally and spatially distinct phases of dynamic cellular processes. However, the data obtained from MS studies are typically huge. A typical cell reportedly contains over 20,000 different proteins, isoforms, and post-translational modifications. To extract useful information from this array of data, powerful protein quantification approaches have emerged, which are collectively classified as *targeted proteomics*; *targeted proteomics* approaches are becoming increasingly popular to address wide-ranging questions in many different systems biology and clinical studies (Ebhardt *et al.*, 2015). As the instrumentation has improved, the new bioinformatics tools to analyse the big data of proteome have also been developed (Calderon-Gonzalez *et al.*, 2016). Integrated MS approaches, antibody-based affinity purification of protein complexes and cross-linking and protein array techniques have contributed towards elucidating complex networks of virus-host protein associations during infection for a wide range of RNA and DNA viruses (Lum and Cristea, 2016). The labelling method is also one of the most useful methods to quantify protein levels, through the use of stable isotope amino acid labelled with C_{13} , H_2 , or N_{16} (Bantscheff *et al.*, 2012). These methods have immensely contributed to important biological discoveries and to the design of new therapeutics.

Metabolomics

Metabolomics is the newest of the 'omics' sciences. The metabolome refers to the sum total of low molecular weight compounds or metabolites such as amino acids, sugars, lipids etc. within a biological system (Krumsiek *et al.*, 2016). Metabolic compounds are typically substrates and by-products of biological processes and enzymatic reactions; they are widely involved in feedback regulatory processes in the cell and being the downstream products often directly influence the cellular or organismal phenotype and thus, metabolome is often regarded as the link between genotype and phenotype (Krumsiek *et al.*, 2016). Metabolomics aims to characterise the type and abundance of metabolites within a biological system at a specified time and under specific environmental conditions, thereby helping gain an understanding of varied metabolic processes (van Rijswijk *et al.*, 2017) and how their dysregulation may be correlated with the onset and progression of diseases (Tolstikov, 2016). *Metabolomics* is being increasingly used to investigate and understand diverse traits such as genetic variation and their correlation with metabolite concentrations (Suhre *et al.*, 2011; Kastenmuller *et al.*, 2015), gene expression patterns (Cavill *et al.*, 2016), biomarker and drug discovery (Wishart, 2016) and assessment of drug safety and efficacy (Ramirez *et al.*, 2013) and even personalised medicine (Tolstikov, 2016; Wishart, 2016).

As with proteomics, rapid strides in MS technologies have facilitated the detection and characterisation of a wide range of metabolites from various biological and clinical samples (Krumsiek *et al.*, 2016; Tolstikov, 2016; Wang *et al.*, 2015; Zampieri *et al.*, 2017) and also the development of new statistical and computational methods to store and analyse the huge amounts of emerging metabolomics data (Krumsiek *et al.*, 2016; Tolstikov, 2016; Xia *et al.*, 2015).

Integrative Analysis of High Throughput Omics Data

The availability of huge amounts of different types of omics data and their analyses have significantly contributed to advance our fundamental understanding of biological systems. However, the analyses of individual datasets offer limited and often disparate insights into the functions of different biological entities, with only a limited insight into how the different entities communicate with each other across different organisational layers. The first step towards an integrative multi-omics data analysis is the integration of different omics datatypes onto a single platform.

Integration of High-Throughput Data

Biological databases have emerged as invaluable resources for compiling, storing and distributing ever increasing amounts of omics data. To achieve multi-omics data analysis, the data from different omics databases must be linked together onto a single

platform. However, the heterogeneity and often inconsistencies of the data generated from the different high-throughput technologies and the formats used to store, manage and query them in different repositories makes the integration of multi-omics data and their analyses a non-trivial challenge. Despite the challenges, integrative data analyses promise holistic and more informative insights into disease biology and drug discovery (Ritchie *et al.*, 2015). Therefore, there have been innumerable efforts to develop new computational tools and approaches to integrate and analyse diverse biological data types that often differ in scope, design, content, analytical abilities as well as the targeted research audience (Chen *et al.*, 2011, 2016; Ge *et al.*, 2003; Gerstein *et al.*, 2002; Smith *et al.*, 2012; Huang *et al.*, 2009; Stein, 2003; Triplet and Butler, 2014).

Typically, the integration of different biological datatypes involves first identifying common elements among them that can be used to build relationships between the different types of biological entities. Among the many different frameworks for integrating and storing biological data, are the data warehouses that compile different biological datatypes onto a common platform, facilitate a wide range of queries linking different datatypes and different functional attributes and can produce a unified output for the user (Stein, 2003; Triplet and Butler, 2014). For instance, TargetMine is an integrative data analysis platform based on the Intermine framework (Smith *et al.*, 2012) that features different omics data types and analytical functions for gene set analysis and biological knowledge discovery (Chen *et al.*, 2011, 2016).

However, most integrated databases are equipped with limited data analyses capabilities and therefore there have been many attempts to develop more sophisticated standalone tools for integrative multi-omics data analyses that can be combined into an overall pipeline to link data generation to storage, analysis and prioritisation of biologically and therapeutically relevant candidates such as genes, transcripts, proteins and metabolites.

Analysis of Integrated Omics Data

A wide range of computational approaches have been developed for integrative analysis of multi-omics data to reduce the dimensionality of the large omics data sets, to identify biologically meaningful patterns and to prioritise candidate gene, proteins and biomolecules of interest (Ritchie *et al.*, 2015; Huang *et al.*, 2017). Broadly, such methods fall into unsupervised approaches such as clustering (Ronan *et al.*, 2016) and matrix factorisation (Li *et al.*, 2012) and supervised integrative analysis methods such as those based on network-based modelling (Robinson and Nielsen, 2016) or machine learning based algorithms (Lin and Lane, 2017). We discuss some of these approaches below.

Unsupervised Integration and Analysis of Omics Data

Unsupervised methods for data integration do not require a training dataset, rather they aim to analyse the data to uncover hidden patterns and/or underlying data structures and to gain and understand the relationships between different biological entities (Ritchie *et al.*, 2015; Huang *et al.*, 2017; Ronan *et al.*, 2016; Kirk *et al.*, 2012; Lock and Dunson, 2013). Clustering is one of the most widely used unsupervised methods for exploring patterns in large omics datasets and to identify subsets of correlated biological entities such as co-expressed or co-functional genes that are relevant to specific disease and cellular processes (Ronan *et al.*, 2016). Clustering approaches for integrative omics data analysis typically fall in the category of joint clustering approaches that leverage shared data features and relationships across diverse biological datatypes to analyse multi-dimensional biological data. As an example, the iCluster method simultaneously performs data integration and dimensionality reduction via a joint latent variable to capture biologically and clinically relevant sub-clusters (Shen *et al.*, 2012).

However, methods that exploit shared data structures to integrate and correlate biological data types may often overlook patterns and features that are unique to individual datatypes. This problem has led to the development of methods that are based on the Bayesian approaches to integrative omics data analysis that simultaneously take into account correlations across different datasets and datatypes as well as source-specific features for integrative clustering of biological datatypes (Kirk *et al.*, 2012; Lock and Dunson, 2013; Ray *et al.*, 2014).

Network-Based Models of Integrative Omics Data Analysis

Network models form an important component of supervised integrative biological data analysis. A network consists of *nodes* that represent different biological entities (gene, RNA molecules, proteins, metabolites etc.) that are connected with each other via *edges* that represent the biological connectivities (physical and/or genetic) between the individual entities, the edges can include PPIs, TF-TG interactions, MTIs and PCIs among others. Typically, network models lie at the heart of the systems biology approaches since they closely reflect genetic, physical and chemical relationships that exists between biological entities within and across different cellular organisations layers. Consequently, a wide range of network-based approaches and tools have been developed to collate and analyse large amounts of biological data to obtain a deeper understanding of cellular physiology and pathogenesis of various diseases (Yan *et al.*, 2017; Ritchie *et al.*, 2015; Huang *et al.*, 2017; Robinson and Nielsen, 2016).

Conclusions

The rapid strides in experimental technologies that can map and quantify the different organisational layers of the living system have contributed to the rapidly increasing scale and sensitivity of omics data. A better understanding of the living systems can only

come from a combined approach that seeks to integrate and analyse the different omics datatypes to unravel the complex networks that link different cellular organisational layers and drive the functioning of the cellular machinery. Consequently a large number of computational tools and pipelines using both supervised and unsupervised approaches have been developed for integrative analysis of multi-dimensional omics data.

However, several challenges persist in extracting biologically meaningful information from the analysis of multi-omics data.. For instance, the experimental platforms used to generate omics data and biological databases that are used to annotate them are far from comprehensive in their coverage of the biological systems. Additionally, the protocols for sample preparation, performing omics experiments and processing the raw omics data and their normalisation can vary substantially among researchers, which may often lead to difficulties with data sharing, analysis and reproducibility. Nevertheless, further advances in the field will facilitate more holistic approaches that can be applied to obtain deeper insights into the dynamics of complex relationships across multiple cellular organisational layers to gain more knowledge of the mechanisms underlying key biological processes including onset and pathogenesis of diseases.

See also: Biological Database Searching. Computational Pipelines and Workflows in Bioinformatics. Computational Systems Biology Applications. Constructing Computational Pipelines. Coupling Cell Division to Metabolic Pathways Through Transcription. Exome Sequencing Data Analysis. Genome Annotation: Perspective From Bacterial Genomes. Host-Pathogen Interactions. Information Retrieval in Life Sciences. Integrative Bioinformatics. Integrative Bioinformatics of Transcriptome: Databases, Tools and Pipelines. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Studies of Body Systems. Whole Genome Sequencing Analysis

References

- Aebersold, R., Mann, M., 2016. Mass-spectrometric exploration of proteome structure and function. *Nature* 537, 347–355.
- Bantscheff, M., *et al.*, 2012. Quantitative mass spectrometry in proteomics: Critical review update from 2007 to the present. *Anal. Bioanal. Chem.* 404 (4). 939–965.
- Berger, B., Peng, J., Singh, M., 2013. Computational solutions for omics data. *Nat. Rev. Genet.* 14, 333–346.
- Boucher, B., Jenna, S., 2013. Genetic interaction networks: Better understand to better predict. *Front. Genet.* 4, 290.
- Calderon-Gonzalez, K.G., *et al.*, 2016. Bioinformatics tools for proteomics data interpretation. *Adv. Exp. Med. Biol.* 919, 281–341.
- Cavill, R., *et al.*, 2016. Transcriptomic and metabolomic data integration. *Brief. Bioinform.* 17, 891–901.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2011. TargetMine, an integrated data warehouse for candidate gene prioritisation and target discovery. *PLOS ONE* 6 (3). e17844.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2016. An integrative data analysis platform for gene set analysis and knowledge discovery in a data warehouse framework. *Database (Oxford)* 2016.
- Civelek, M., Lusis, A.J., 2014. Systems genetics approaches to understand complex traits. *Nat. Rev. Genet.* 15, 34–48.
- Consortium, G.T., 2013. The Genotype-Tissue Expression (GTEx) project. *Nat. Genet.* 45, 580–585.
- Consortium, F., *et al.*, 2014. A promoter-level mammalian expression atlas. *Nature* 507, 462–470.
- Ehardt, H.A., *et al.*, 2015. Applications of targeted proteomics in systems biology and translational medicine. *Proteomics* 15, 3193–3208.
- Fields, S., Song, O., 1989. A novel genetic system to detect protein-protein interactions. *Nature* 340, 245–246.
- Fukushima, A., Kanaya, S., Nishida, K., 2014. Integrated network analysis and effective tools in plant systems biology. *Front. Plant Sci.* 5, 598.
- Ge, H., Walhout, A.J., Vidal, M., 2003. Integrating 'omic' information: A bridge between genomics and systems biology. *Trends Genet.* 19, 551–560.
- Gresham, D., Dunham, M.J., Botstein, D., 2008. Comparing whole genomes using DNA microarrays. *Nat. Rev. Genet.* 9, 291–302.
- Gerstein, M., Lan, N., Jansen, R., 2002. Proteomics. Integrating interactomes. *Science* 295, 284–287.
- Hrdlickova, R., Toloue, M., Tian, B., 2017. RNA-seq methods for transcriptome analysis. *Wiley Interdiscip. Rev. RNA* 8 (1).
- Huang da, W., Sherman, B.T., Lempicki, R.A., 2009. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* 4, 44–57.
- Huang, S., Chaudhary, K., Garmire, L.X., 2017. More is better: Recent progress in multi-omics data integration methods. *Front. Genet.* 8, 84.
- Huttlin, E.L., *et al.*, 2017. Architecture of the human interactome defines protein communities and disease networks. *Nature* 545, 505–509.
- Huttlin, E.L., *et al.*, 2015. The BioPlex network: A systematic exploration of the human interactome. *Cell* 162, 425–440.
- Jean Beltran, P.M., *et al.*, 2017. Proteomics and integrative omic approaches for understanding host-pathogen interactions and infectious diseases. *Mol. Syst. Biol.* 13, 922.
- Kastenmuller, G., *et al.*, 2015. Genetics of human metabolism: An update. *Hum. Mol. Genet.* 24, R93–R101.
- Kirk, P., *et al.*, 2012. Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics* 28, 3290–3297.
- Kopczynski, D., *et al.*, 2017. Multi-OMICS: A critical technical perspective on integrative lipidomics approaches. *Biochim. Biophys. Acta* 1862, 808–811.
- Krumsiek, J., Bartel, J., Theis, F.J., 2016. Computational approaches for systems metabolomics. *Curr. Opin. Biotechnol.* 39, 198–206.
- Levy, S.E., Myers, R.M., 2016. Advancements in next-generation sequencing. *Annu. Rev. Genom. Hum. Genet.* 17, 95–115.
- Li, W., *et al.*, 2012. Identifying multi-layer gene regulatory modules from multi-dimensional genomic data. *Bioinformatics* 28, 2458–2466.
- Lin, E., Lane, H.Y., 2017. Machine learning and systems genomics approaches for multi-omics data. *Biomark. Res.* 5, 2.
- Lock, E.F., Dunson, D.B., 2013. Bayesian consensus clustering. *Bioinformatics* 29, 2610–2616.
- Lowe, R., *et al.*, 2017. Transcriptomics technologies. *PLOS Comput. Biol.* 13 (5). e1005457.
- Lum, K.K., Cristea, I.M., 2016. Proteomic approaches to uncovering virus-host protein interactions during the progression of viral infection. *Expert Rev. Proteom.* 13, 325–340.
- Meadows, J.R.S., Lindblad-Toh, K., 2017. Dissecting evolution and disease using comparative vertebrate genomics. *Nat. Rev. Genet.* 18, 624–636.
- Moore, J.B., Weeks, M.E., 2011. Proteomics and systems biology: Current and future applications in the nutritional sciences. *Adv. Nutr.* 2, 355–364.
- Ramirez, T., *et al.*, 2013. Metabolomics in toxicology and preclinical research. *ALTEX* 30, 209–225.
- Ray, P., *et al.*, 2014. Bayesian joint analysis of heterogeneous genomics data. *Bioinformatics* 30, 1370–1376.
- Ritchie, M.D., *et al.*, 2015. Methods of integrating data to uncover genotype-phenotype interactions. *Nat. Rev. Genet.* 16, 85–97.
- Robinson, J.L., Nielsen, L., 2016. Integrative analysis of human omics data using biomolecular networks. *Mol. Biosyst.* 12, 2953–2964.
- Rolland, T., *et al.*, 2014. A proteome-scale map of the human interactome network. *Cell* 159, 1212–1226.
- Ronan, T., Qi, Z., Naegle, K.M., 2016. Avoiding common pitfalls when clustering biological data. *Sci. Signal.* 9 (432). re6.
- Sabido, E., Selevsek, N., Aebersold, R., 2012. Mass spectrometry-based proteomics for systems biology. *Curr. Opin. Biotechnol.* 23, 591–597.

- Schulze, A., Downward, J., 2001. Navigating gene expression using microarrays – A technology review. *Nat. Cell Biol.* 3, E190–E195.
- Serín, E.A., *et al.*, 2016. Learning from co-expression networks: Possibilities and challenges. *Front. Plant Sci.* 7, 444.
- Shameer, K., *et al.*, 2016. Interpreting functional effects of coding variants: Challenges in proteome-scale prediction, annotation and assessment. *Brief. Bioinform.* 17, 841–862.
- Shen, R., *et al.*, 2012. Integrative subtype discovery in glioblastoma using iCluster. *PLOS One* 7 (4). e35236.
- Slonim, D.K., Yanai, I., 2009. Getting started in gene expression microarray analysis. *PLOS Comput. Biol.* 5 (10). e1000543.
- Smith, R.N., *et al.*, 2012. InterMine: A flexible data warehouse system for the integration and analysis of heterogeneous biological data. *Bioinformatics* 28, 3163–3165.
- Spies, D., Ciaudo, C., 2015. Dynamics in transcriptomics: Advancements in RNA-seq time course and downstream analysis. *Comput. Struct. Biotechnol. J.* 13, 469–477.
- Stein, L.D., 2003. Integrating biological databases. *Nat. Rev. Genet.* 4, 337–345.
- Suhre, K., *et al.*, 2011. Human metabolic individuality in biomedical and pharmaceutical research. *Nature* 477, 54–60.
- Sun, Y.V., Hu, Y.J., 2016. Integrative analysis of multi-omics data for discovery and functional studies of complex human diseases. *Adv. Genet.* 93, 147–190.
- Tolstikov, V., 2016. Metabolomics: Bridging the gap between pharmaceutical development and population health. *Metabolites* 6 (3).
- Trevino, V., Falciani, F., Barrera-Saldana, H.A., 2007. DNA microarrays: A powerful genomic tool for biomedical and clinical research. *Mol. Med.* 13, 527–541.
- Triplet, T., Butler, G., 2014. A review of genomic data warehousing systems. *Brief. Bioinform.* 15, 471–483.
- van Dam, S., *et al.*, 2017. Gene co-expression analysis for functional classification and gene-disease predictions. *Brief. Bioinform.*
- van Kampen, A.H., Moerland, P.D., 2016. Taking bioinformatics to systems medicine. *Methods Mol. Biol.* 1386, 17–41.
- van Rijswijk, M., *et al.*, 2017. The future of metabolomics in ELIXIR. *F1000Res* 6.
- Wang, Y., *et al.*, 2015. Current state of the art of mass spectrometry-based metabolomics studies – A review focusing on wide coverage, high throughput and easy identification. *RSC Adv.* 5, 78728–78737.
- Wishart, D.S., 2016. Emerging applications of metabolomics in drug discovery and precision medicine. *Nat. Rev. Drug Discov.* 15, 473–484.
- Xia, J., *et al.*, 2015. MetaboAnalyst 3.0 – Making metabolomics more meaningful. *Nucleic Acids Res.* 43, W251–W257.
- Yan, J., *et al.*, 2017. Network approaches to systems biology analysis of complex disease: Integrative methods for multi-omics data. *Brief. Bioinform.*
- Zampieri, M., *et al.*, 2017. Frontiers of high-throughput metabolomics. *Curr. Opin. Chem. Biol.* 36, 15–23.
- Zhu, J., *et al.*, 2012. Stitching together multiple data dimensions reveals interacting metabolomic and transcriptomic networks that modulate cell regulation. *PLOS Biol.* 10 (4). e1001301.

Profiling the Gut Microbiome: Practice and Potential

Toral Manvar, Xcelris Labs Ltd., Ahmedabad, India and Gujarat University, Ahmedabad, India

Vijay Lakhujani, Xcelris Labs Ltd., Ahmedabad, India

© 2019 Elsevier Inc. All rights reserved.

Acronyms

BLAST	Basic Local Alignment Search Tool	PacBio	Pacific Biosciences
COPE	Connecting Overlapping Paired End reads	PcoA	Principal Coordinates Analysis
DGGE	Denaturing Gradient Gel Electrophoresis	PCR	Polymerase Chain Reaction
FLASH	Fast Length Adjustment of Short Reads	PICRUSt	Phylogenetic Investigation of Communities by Reconstruction of Unobserved States
FMT	Fecal Microbiota Transplantation	QIIME	Quantitative Insights Into Microbial Ecology
GI	Gastrointestinal	qPCR	Quantitative Polymerase Chain Reaction
GO	Gene Ontology	R	R Programming
HMM	Hidden Markov Model	RDP	Ribosomal Database Project
HMP	Human Microbiome Project	SAS	Statistical Analysis System
IBD	Inflammatory Bowel Diseases	SCFA	Short Chain Fatty Acid
KEGG	Kyoto Encyclopedia of Genes and Genomes	SMRT	Single-Molecule Real-Time
MetaHit	Metagenome Human Intestinal Tract Project	SOLiD	Sequencing by Oligo/Ligation and Detection
MG-RAST	Metagenomics-Rapid Annotations Using Subsystems Technology	SSCP	Single-Strand Conformation Polymorphism
NAST	Nearest Alignment Space Termination	T2D	Type 2 Diabetes
NGS	Next Generation Sequencing	TGGE	Temperature Gradient Gel Electrophoresis
OTU	Operational Taxonomic Unit	T-RFLP	Terminal Restriction Fragment Length Polymorphism

Introduction

Gut profiling is the study of unique microbial fingerprints found inside the human gut with the objective of understanding their associations with well-being and disorder. There seems to be a never ending debate on the actual number of microorganisms in the human body, specifically in the gastrointestinal (GI) tract (Sender *et al.*, 2016). The latter accounts for more than a thousand (~1000–1150) different bacterial species (Sánchez *et al.*, 2017; Qin *et al.*, 2010) with Firmicutes, Bacteroidetes, Actinobacteria, and Proteobacteria as common occurring phyla (Arumugam *et al.*, 2011). It is believed that there are around 3.3 million unique genes in human gut which is 150 times more than the genes in human genome (Zhu *et al.*, 2010a, b). Researchers are intrigued to know that no two individuals have exactly the same composition of bacterial species in their guts just as the fingerprints. The gut microflora have a vital role to play in wide variety of metabolic, trophic and protective functions which includes: (i) Facilitating withdrawal of energy from non digestible polysaccharides producing short chain fatty acid (SCFAs), vitamins and antioxidants (ii) detoxification of the deleterious xenobiotics (iii) modulation and development of immune-system (iv) Epithelial cell growth and differentiation (v) secretion of anti-microbial products, providing the barrier effect against the pathogenic bacteria through the development of colonization resistance (vi) nerve regulation, controlling mood and behavior (Guarner and Malagelada, 2003; Ly *et al.*, 2011; Sun and Chang, 2014; Vernocchi *et al.*, 2016; Mayer *et al.*, 2014).

Microbiome research suggest that dysbiotic shifts i.e., the changes in the make-up of gut microbiota in terms of number or categorization may imbalance the homeostatic harmony causing conditions such as obesity (Ly *et al.*, 2011; Alard *et al.*, 2016), inflammatory bowel diseases (IBD) (Knights *et al.*, 2013; Ferreira *et al.*, 2014; Norman *et al.*, 2015), allergy and asthma (Abrahamsson *et al.*, 2014; Ly *et al.*, 2011), chronic brain diseases (Mayer *et al.*, 2014), colorectal cancer (Sobhani *et al.*, 2011), type II diabetes (Qin *et al.*, 2012; Karlsson *et al.*, 2013; Moreno-Indias *et al.*, 2014; Musso *et al.*, 2010), rheumatoid arthritis (Scher and Abramson, 2011), fatty liver disease (Miele *et al.*, 2009), lesional skin (Ganju *et al.*, 2016) and numerous other diseases (Fig. 1). Thus, to understand the basis of dysbiotic shifts, it is important to first discover and characterize these communities by answering the following key questions: (1) Who is there in the gut? (2) What roles are they playing? (3) How they affect the gut functionally? (4) How their abundance changes with disease conditions? Because of the scale, complexity and the impact gut microbes has on an individual, the ecosystem of the colon has been the most diligently studied body habitat (Lloyd-Price *et al.*, 2016).

Background

The time since the microscopic organisms have been discovered during 1665–1683 by Robert Hooke and Antoni Van Leeuwenhoek (Gest, 2004), mankind has travelled for more than 300 years to understand the integral relationship of them with us. There is a deep-seated alliance of human beings with the cohort of microorganisms, often called the microbiome (Lederberg, 2000), and hence it will be not be an exaggeration to call the microbiome as the “second genome” (Grice and Segre, 2012; Zhu *et al.*, 2010a,b). While human beings share 99.9% similarity with each other at the genome level, our gut microbes can be very

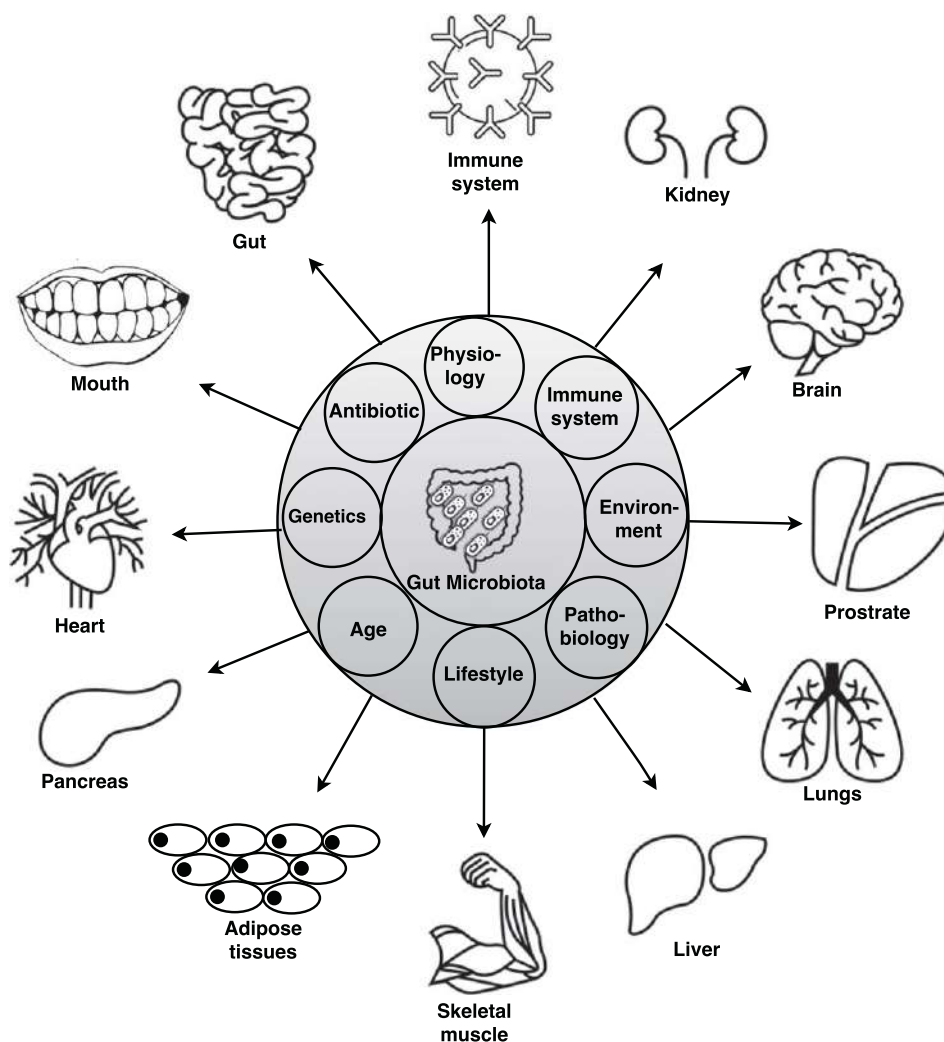


Fig. 1 The ecosystem of gut microbiome. At the centre lies the core gut microbiota. The inner small circles represent the factors that affect the composition and diversity of the gut microbes. The figures projecting out from the outer circle represent the diverse class of body organs affected by the gut microbiome interactions with human host.

much different providing us different phenotypes. The gastrointestinal tract which is sterile at birth (Ly *et al.*, 2011) starts the very first colonization right after the birth and in the following few days after delivery. Studies have shown that this colonization is largely dependent on the way of delivery (natural or c-section) and the kind of food provided in infancy (mother's milk or formula) and adulthood (vegetarian or non-vegetarian) (Jandhyala *et al.*, 2015).

A number of large scale collaborative projects have been initiated reinforcing our understanding of functional prospective of microbiota on human health. It includes "Metagenome Human Intestinal Tract project" (MetaHit project: see "Relevant Websites section") initiated in year 2010 having considered 124 European adults stool samples most of which were healthy individuals. This project was funded by the European Seventh Framework Program until 2012. Its objective was to understand the link between the human intestinal microbiota and two disorders of increasing incidence in Europe namely inflammatory bowel disease and obesity. Later in 2012, the "Human Microbiome Project" (HMP: see "Relevant Websites section") was launched to define microbes existing in 5 major body sites of 242 healthy adult from United States using 16S and shotgun metagenome approach. By end of 2013, the study of European population has begun under MyNewGut project (see "Relevant Websites section") with an aim to identify specific dietary strategies to improve the long-term health of the population. One more project entitled "American Gut Project" was conceived in 2012 which is the world's largest crowd-funded citizen science project. The participants of this project are American citizens who learn about their own microbiome and also contribute for samples and money required for executing the project smoothly.

Although there are number of molecular techniques ranging from culturing to fingerprinting, DNA microarray and clone sequencing which have been used since last many years, Next Generation Sequencing (NGS) has given a new insight into study of gut microbes. Several sequencing approaches have been deployed so far including shotgun metagenomics, microbial transcriptomics, whole genome microbial sequencing and marker gene sequencing. Additionally, NGS has also lead to the generation of giga bytes or even terabytes of data. Hence, it becomes essential to process, manipulate, analyze, visualize and store the sequencing data in a

structured and meaningful manner. It is at this point where bioinformatics approaches are playing a pivotal role and various scientific methods, tools and protocols have been formulated to study the microbiome and achieve the desired outputs.

Approaches for Gut Microbiome Study

Molecular Techniques From Pre-ngs to NGS

Before the advent of NGS technologies, early researchers have employed a number of techniques which can be broadly divided into two categories (Rastogi and Sani, 2011) as given in Fig. 2:

1. Culture dependent,
2. Culture-independent.

Culture dependent

The basis of culture dependent methods is isolation of microbes followed by inferring their composition on standard commercial growth media under controlled laboratory conditions (Handelsman, 2004; Kirk *et al.*, 2004). To maximize the cultivable fraction of microbial communities, many improvements have been made in cultivation method and culture media so far to mimic the natural environments (Rastogi and Sani, 2011). This methodology remained as golden standard for many years for characterization of microbes (Fraher *et al.*, 2012). However, culture dependent approach is not preferred for gut microbiota study as this ecosystem is considered as most diverse and it contains large number of unculturable bacteria (70%–80%) (Eckburg *et al.*, 2005;

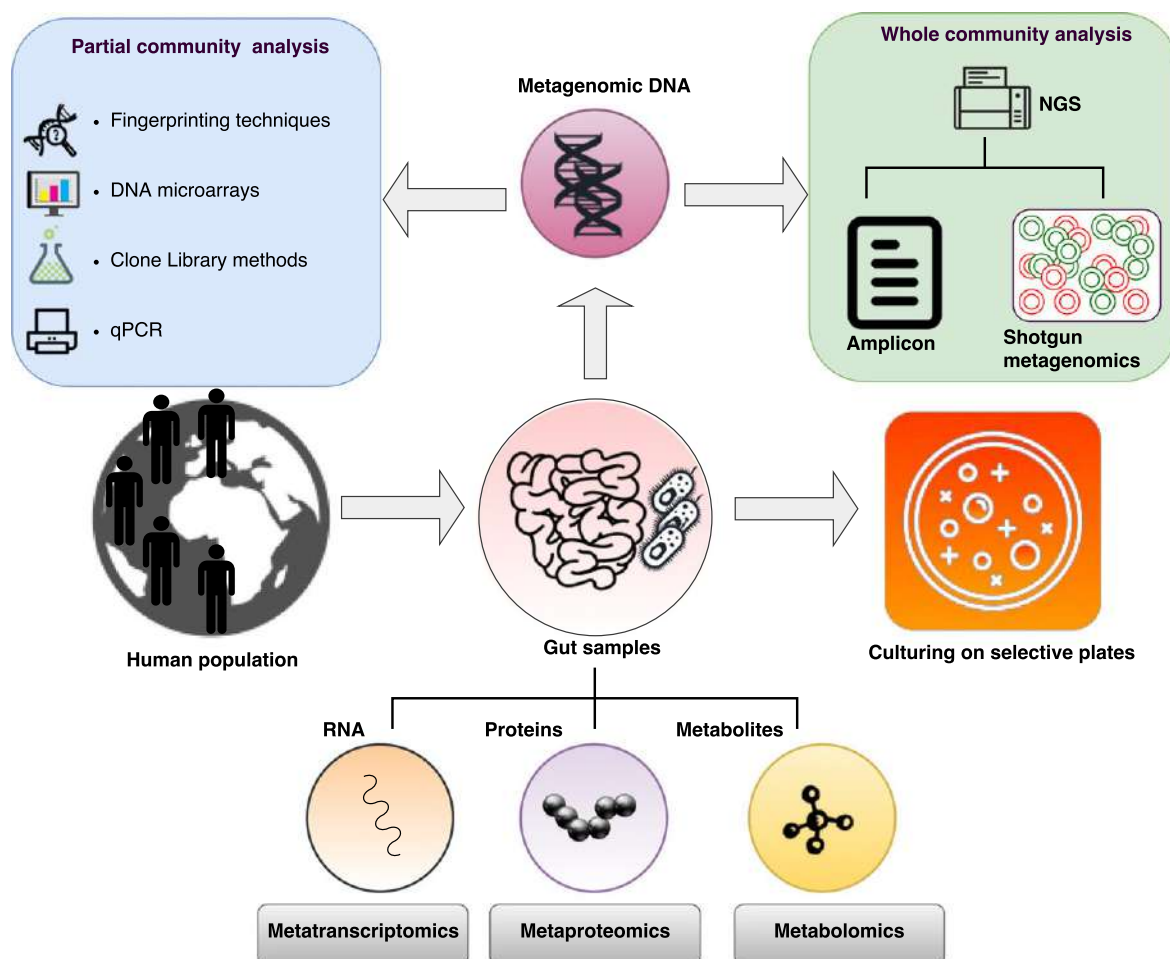


Fig. 2 Approaches for gut microbiome study. Pictorial representation of the different approaches (partial/whole community analysis) to study gut microbiome. Apart from DNA samples (metagenomics), RNA (metatranscriptomics study), protein (metaproteomics study) and metabolite (metabolomics study) samples are often useful to get a complete picture of the gut microbial interactions. Note: Modified from Rousseau, C., Butel, M.J., 2015. High-throughput techniques for studying of gut microbiota. *Advances in Probiotic Technology* 146.

Hayashi *et al.*, 2002; Suau *et al.*, 1999). Due to this shortcoming there was paradigm shift from culture dependent to culture independent methods.

Culture independent

The basis of culture-independent method is isolation of nucleic acid to analyze selected marker genes such as 16S rRNA or any other functional genes (for example, gyrase beta subunit or recombinase (A) to characterize prokaryotes community. In general, this molecular technique can be divided into (1) partial community analysis approaches (fingerprinting, clone library method DNA microarray, qPCR) (2) whole community analysis approaches (whole genome sequencing, metagenomics, metatranscriptomics, metaproteomics and metabolomics) depending upon the strength of techniques to exhibit the microbial diversity and their function (Rastogi and Sani, 2011).

Partial community analysis approaches

(a) Fingerprinting techniques:

Fingerprinting techniques namely denaturing or temperature gradient gel electrophoresis (DGGE/TTGE), terminal restriction fragment length polymorphism (T-RFLP), automated rRNA intergenic Spacer Analysis (ARISA), Amplified ribosomal DNA restriction analysis (ARDRA) and single-strand conformation polymorphism (SSCP) are commonly implemented to study complex microbial ecosystem like GI (Zoetendal *et al.*, 2004). These are cheap and rapid methods and thus are preferred even today when large number of samples are involved and aim of the study is just to identify dominant bacterial group (more than 1%) (Hamady and Knight, 2009). Studies implementing these techniques have revealed that species richness increases along the length of the gut (Rousseau and Butel, 2015). However, the main drawback of these techniques is that data from different studies cannot be combined and thus compared.

(b) Diversity microarrays:

Diversity microarray is one of the promising techniques to study gut microbes (Zoetendal *et al.*, 2004). It works on the principle of hybridization of probes (representing known genes attached to the glass surface) with DNA or RNA present in sample detected by means of fluorescence. HuGChip and PhyloChip are commonly used microarrays for microbiome studies (Tottey *et al.*, 2013; Palmer *et al.*, 2007). In 2003, for the first time (Call *et al.*, 2003) scrutinized the antibiotic resistance genes in intestinal microbiota using microarray. IBD, a disorder of gut microbiota was also studied in Kang *et al.* (2010). using phylogenetic microarray. Main shortcoming of this technique is that the sequences to be used in probe designing have to be known, though if not at specific level but at upper level (phylum or family). Thus, this technique cannot be used for undiscovered microbes (DeSantis *et al.*, 2007). However, its potential to process number of samples simultaneously makes it favorable method for many studies.

(c) Clone Library Method:

Here 16S rRNA gene is amplified from the samples and then cloned followed by its sequencing. 16S rRNA database for instance Ribosomal Database Project (RDP), Greengenes, GenBank are then used to identify microbes represented by cloned sequences (DeSantis *et al.*, 2007). Suau *et al.* were the first to characterize microbes in gut utilizing Sanger sequencing in the year 1999. Though, it is considered as “golden standard” for preliminary diversity study, the drawback of using this technique is that it requires large number of clones to be prepared and sequenced even to capture half of the diversity of sample making it labor-intensive and expensive technique (Rousseau and Butel, 2015).

Whole community analysis approaches

Since the completion of human genome project in 2003, a significant progress has been made in the field of human health and bioinformatics which was further fueled by the emergence of next generation sequencing (NGS) techniques. NGS has enabled quick analysis of huge number of samples with falling costs. This characteristic made it the most feasible and convincing method for studying uncultured gut microbial communities with no prior knowledge of the organism (Mandal *et al.*, 2015; Frey *et al.*, 2014). Gradually, this revolutionary technology resulted in development of new field of genomics called “metagenomics”, defined as direct genetic analysis of genomes present in particular ecology eliminating the need for cultivating clonal cultures. Based on the aim of study, metagenomics can be divided into (i) Marker gene amplification metagenomics or meta-genetics (Handelsman, 2009) which is used for identification and abundance analysis of bacterial community. It is primarily based on the diversity of one gene (e.g., 16S rRNA or functional genes like amoA/B, nirK, nirS and nosZ etc.) obtained after PCR amplification. (ii) Full shotgun metagenomics (Sun and Xia, 2015) which provides access to the functional gene composition of microbial communities giving much broader description than phylogenetic surveys.

(i) Marker gene amplification metagenomics

Since Woese and Fox (1977) first pioneered the use 16S rRNA gene in classifying microbes, it is regarded as the most versatile phylogenetic marker for microbial classification as (i) it is ubiquitous in bacteria (ii) its function is conserved over time and (iii) its length (~1500 bp) is appropriate for isolation, sequencing and subsequent bioinformatics analysis (Janda and Abbott, 2007). The 16S rRNA gene includes 9 hyper variable regions labeled as V1–V9 ranging from about 30–100 base pairs. Though, this gene is mostly conserved, the hyper variable regions can change in a very short evolutionary period and thus provides an advantageous and efficient way for microbe identification. Most 16S studies operate on one of these hyper variable regions at any given time due to technical constraints. Marker gene metagenomics is the preferable choice for diversity analysis as it targets specific region and thus, it is cost effective and computationally easy to analyze. However it restricts the

knowledge to only particular region studied, not providing any information of functional and metabolic aspects of microbial community present in the same.

(ii) *Full shotgun metagenomics*

As the word suggests, in full shotgun metagenomics, complete DNA including functional genes and phylogenetic markers are studied as a whole, thereby in addition to identification of microbes, it can provide information about the function of this microbial community. It is a perfect substitute to cumbersome cloning method requiring large amount of genomic DNA to sequence.

In the last decade, many studies have been carried out to analyze gut microbiota diversity using both the functional and phylogenetic approach using different NGS platforms. Metagenomics (DNA based) can be further combined with metatranscriptomics (mRNA based), metaproteomics (protein based) and metabolomics (metabolism based) to analyze composition, function, expressed activities to provide insights into bacteria-host interactions that may lead to disease (Wang *et al.*, 2015; Franzosa *et al.*, 2015). Currently five high throughput sequencing platforms are used for studying the gut microbiome diversity namely Illumina (e.g., GA II, HiSeq, MiSeq, NextSeq500), 454 Life Sciences Roche (GS FLX Titanium sequencer), Applied Biosystems SOLiD, life technologies' ion torrent personal genome machine (PGM) and Pacific Bioscience's (PacBio) Single-Molecule Real-Time (SMRT) (Medini *et al.*, 2008; Hodkinson and Grice, 2015). Out of these, Illumina is the choice of most of the researchers as it gives high throughput data and advantage of read length up to 300 bp. All these NGS technologies have already been reviewed pre-eminently by Mardis (2008) and Metzker (2010). The impact of different sequencing platforms for microbial analysis has also been reviewed and discussed in detail by Allali *et al.* (2017) and Clooney *et al.* (2016).

Bioinformatics Approaches to Study the Gut Microbiome

Computational tools, techniques and algorithms are frequently employed to understand and interpret biological data. The complexity and amount of data generated through latest sequencing technologies make it difficult to manually analyze and visualize data to extract meaningful results. Therefore, bioinformatics methodologies have been constantly evolving in various aspects and microbiome study is no exception. Broadly, bioinformatics approaches could be categorized into (i) Marker gene analysis (using a specific and ubiquitous gene across the microbial world, for e.g., 16S rRNA and ITS region) and (ii) Metagenomic analysis (Oulas *et al.*, 2015; Thomas *et al.*, 2012), as discussed briefly below:

Marker gene analysis (e.g., : rRNA or ITS region)

Popular bioinformatics tools employed for analysis of phylogenetic markers like rRNA amplicon and internal transcribed spacer (ITS) are MEGAN (Huson *et al.*, 2007), Mothur (Schloss *et al.*, 2009), MG-RAST (Glass *et al.*, 2010) and QIIME (Caporaso *et al.*, 2010). Among these, QIIME is the most widely used and established pipeline due to its continuous support and development, well written documentation and ease of use. Though, every tool has its unique core algorithm, there is a basic set of analysis steps (Fig. 3) which most of the tools use and is described below:

(1) *Pre-processing of sequencing data:*

Pre-processing includes trimming of sequencing reads according to quality, stitching the reads to generate longer stretches and removal of chimeras. FastQC (see "Relevant Websites section") and NGS Toolkit (Patel and Jain, 2012) are some of the most commonly used fastq quality assessment tools. After surveying the data, reads need to be trimmed for adapter sequences and filtered on the basis of quality or length as per experimental requirements. For this purpose, few tools like Cutadapt (Martin *et al.*, 2011), Trimmomatic (Bolger *et al.*, 2014) and FASTX Toolkit (see "Relevant Websites section") are frequently used. Some of the tools like NGS Toolkit and Trim Galore (Krueger, 2015) come with dual functionality performing quality assessment along with trimming, thus reducing an extra step. Studies suggest that longer sequence could capture much more information than the shorter ones which are typically 100–300 base pairs. Hence, paired reads could be stitched generating a single long stretch by exploiting the overlap between the paired reads. Several software are available to achieve the same including PEAR (Zhang *et al.*, 2013), FLASH (Magoč and Salzberg, 2011), bbmerge tool from BBMAP (Bushnell), PANDASeq (Masella *et al.*, 2012), COPE (Liu *et al.*, 2012a,b), UPARSE merge (Edgar, 2013) and fastq-join (Aronesty, 2011). All of them vary on the basis of sensitivity, performance and stitching percentage and could be used according to the experimental needs. Though, some of these tools may have high merging rate at default parameters, however, the rate of false positives may be high. Out of them, BBMerge has lower false positive rate than PEAR, FLASH and fastq-join. COPE is quite comparable or slightly better than FLASH in terms of overall performance, however, it is unstable at N=0. Following stitching, the subsequent pre-processing step is the removal of chimeric sequences which are PCR artifacts. Two or more biological DNA sequences may join incorrectly during PCR cycle to generate artificial novel sequence and may further lead to misinterpretation of data. For chimera identification, mainly two approaches are employed viz. reference based and de novo (absence of reference database having chimera free sequences). In case of reference based analysis, databases such as Gold (see "Relevant Websites section") and RDP (Maidak *et al.*, 1996) are generally used. However, de novo analysis is preferred method when no information is present for particular type of metagenome sample. Pintail (Ashelford *et al.*, 2005) and DECIPHER (see "Relevant Websites section") work using reference based approach and others such as USEARCH and UCHIME (Edgar, 2010; Edgar *et al.*, 2011), ChimeraSlayer (Haas *et al.*, 2011) and Perseus (Quince *et al.*, 2011) can work either way. The advantage of UCHIME is that it works equally well with both ITS sequences and 16S

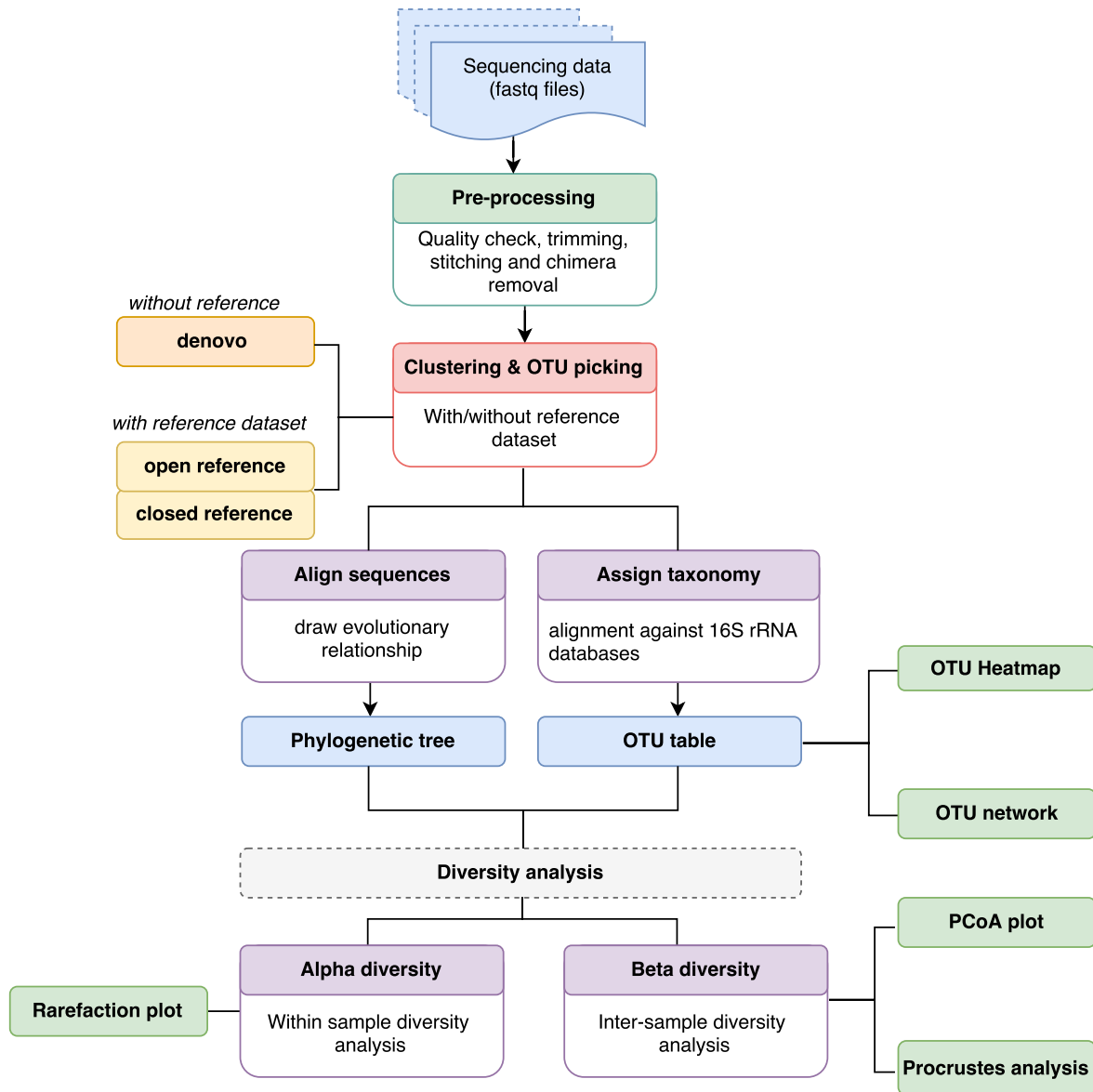


Fig. 3 Bioinformatics workflow for 16S rRNA marker gene studies. The analysis starts from pre-processing of raw sequencing reads followed by clustering and OTU picking with suitable strategies. Taxonomy is assigned by alignment methods and generating phylogenetic tree to draw evolutionary relationships. Diversity analysis (alpha and beta) is useful to study inter and intra sample diversity.

amplicon. It is also the default method when USEARCH is used in QIIME (Navas-Molina *et al.*, 2013). Chimera identification is a crucial step and the tool to remove it should be carefully chosen.

(2) Clustering and OTU picking

The word OTU stands for Operational Taxonomic Unit and there are different views on the origin and the exact meaning of the same which is out of the scope of this discussion. However, in context to metagenomics, OTU can represent an individual organism, a named taxonomic group i.e. species or genus, or a group with undetermined evolutionary relationships sharing common characters. Hence, it is often up to a scientist to put a proper context to OTU with the study. Generally, gut samples are obtained in batches and hence are often multiplexed. Such samples are first de-multiplexed (reads are assigned to samples based on unique barcode attached to the sequences) followed by OTU picking step which is basically clustering of similar sequences at (typically) 97% identity or more. However, the identity threshold of 97% may be considered for revision taking into account the increased number of high quality 16S sequences till date (Edgar, 2017).

Sequences thus clustered are considered to represent the same bacterial species (Janda and Abbott, 2007). Methods for OTU picking may vary widely and the most commonly used tool QIIME supports three different approaches, viz (i) de novo, (ii) open reference and (iii) closed reference, all of them uses UCLUST as default algorithm. As the name suggests, denovo method is solely based on pair wise identity among sequences in the absence of a reference dataset. While open and closed

reference methods relies on reference dataset for OTU picking. But the primary difference between them is, in case of open reference method the sequences are clustered based on hits obtained against reference dataset followed by denovo clustering of reference unaligned reads, whereas in case of closed reference method, reads after alignment to reference database are discarded. When a little is known about sample, the best approach is to select is open reference method as it takes advantage of reference as well as denovo method, thereby giving an adequate chance for identification of novel species. The next step is to pick representative sequence of each OTU generated, so as to reduce the time and resources in further downstream analysis. Again, different strategies are employed for picking up the representative sequence. The longest or the first or the most abundant sequence could be picked up as a representative.

(3) *Taxonomic assignment of OTUs and alignment*

Given a set of sequences and reference dataset, in the next step, taxonomy, associated with the reference sequences, can be assigned to each sequence using a variety of algorithms. There are a number of rRNA sequence databases containing well annotated and curated sequences of bacterial/archaeal and fungal origin. Some of the widely used rRNA databases include RDP (Maidak *et al.*, 1996), SILVA (Pruesse *et al.*, 2007), GreenGenes (DeSantis *et al.*, 2006a) and EzTaxon-e (Kim *et al.*, 2012). However, reference databases for ITS data analysis are quite few; most widely accepted being Unite (Kõljalg *et al.*, 2005). A number of different algorithms are available in QIIME itself, such as uclust (Edgar, 2010), BLAST (Altschul *et al.*, 1997), RDP classifier (Wang *et al.*, 2007), Mothur (Schloss *et al.*, 2009) and tax2tree (McDonald *et al.*, 2012) which can be employed to search for the best possible match for an OTU in reference database assisting in inferring the possible taxonomic lineage. However, RDP classifier has an added advantage of providing bootstrap confidence score along with the taxonomy assignment (Lan *et al.*, 2012). Higher the score, higher is the confidence that read belongs to a particular reference lineage. A threshold of 0.8 is used as default bootstrap confidence score in QIIME (Navas-Molina *et al.*, 2013).

In addition to taxonomic assignment, it is important to represent evolutionary distance between the sequences in form of phylogenetic tree wherein the distance can be visualized in FigTree (Rambaut, 2009) or TreeView X (Page, 2002) software. Result of multiple sequence alignment is used as an input for phylogenetic analysis which in turn can be obtained using tools like clustalW2 (McWilliam *et al.*, 2013). However, a more recent method to achieve the same is to use NAST algorithm (DeSantis *et al.*, 2006b) to align the OTU representative sequences to a pre-aligned reference database of 16S sequences (often called a “template”). For instance, QIIME uses PyNAST (Caporaso *et al.*, 2009) which is a Python implementation of the NAST. Another tool called Infernal makes use of an HMM approach to incorporate secondary structure information. Due to incorporation of secondary structure information, it is better at identifying variable rRNAs sequences conserved at structural level though, it is a bit slower.

(4) *Diversity analysis:*

The basic components to estimate the microbial diversity are alpha, beta and gamma diversity. Within sample diversity is represented by of alpha diversity which is indicative of “richness” (total number of species in the sample) and “evenness” (distribution of relative abundance of species). To remove the bias introduced by sequencing depth while capturing the true diversity, rarefaction curves are often drawn to determine whether the sequencing depth is adequate to represent the total diversity within the sample. Several metrics are available in both Mothur and QIIME to calculate alpha diversity. The metrics are broadly divided into 2 categories, viz. phylogenetic and non-phylogenetic, depending upon their ability to utilize phylogenetic information of microbes. Shannon index, Simpson’s index, Good’s coverage, chao1, observed species are non-phylogenetic bases metrics. Chao1 and observed OTUs/species metrics are used to get total diversity in sample, while metrics like Shannon index and Simpson index are used to get information on relative abundance of each taxon identified as well as diversity captured found in sample. On the other hand, phylogenetic diversity metric like Phylogenetic diversity (PD) whole tree takes phylogeny of microbes into account to provide diversity in the sample. Thus it has the added advantage of providing divergence-based analysis (Lozupone and Knight, 2008) for sequences which are different in base composition but similar at metabolite level.

Beta diversity is a term for the inter-sample comparison or across environmental sample comparison which is estimated by calculating the distance between samples using popular metrics like Unifrac (Weighted and UnWeighted), Bray-Curtis, Euclidean and Abundance-Jaccard. Like alpha diversity, beta diversity can also be divided as phylogenetic and non-phylogenetic based methods. Metric such as Jaccard provides richness estimation and Bray-Curtis, Euclidean can provide richness as well as evenness estimation among samples thereby giving better idea about the differences shared by different samples. However, Unifrac metrics like Unweighted and Weighted are phylogenetic based metrics for qualitative as well as quantitative beta diversity analysis respectively (Lozupone and Knight, 2008). Gamma diversity provides the total species diversity found across all samples.

(5) *Statistical analysis and visualization:*

The final step in broad characterization of complex microbial population is correlating gut microbiome with diseased or normal metabolic profile in human using different statistics and visual aids. MG-RAST (Glass *et al.*, 2010) performs a comprehensive analysis and at the end provides a detailed report which is publication ready. Similarly QIIME provides python based programs for 3D visualization of the correlation between the samples using principal coordinates analysis (PcoA). Further, procrustes analysis (Gower, 1975) is often performed to compare weighted and unweighted UniFrac PCoA plots generated by the same processing pipeline adding another dimension. PICRUSt (Langille *et al.*, 2013) analysis could also be carried out to derive functional aspect of sample using 16s rRNA gene information. The visualizations and further statistical tests are not limited to the popularly available tools. There are in fact, a number of open sources and commercial tools/programs/packages like R, SAS and MATLAB available to extend the understanding. Finally, programming in high level languages like Python cannot be ignored and

in fact the complete QIIME pipeline is being written in Python. Advanced, Python packages like NumPy and SciPy are often used parallel.

Metagenomic analysis

(1) Metagenomic assembly:

Unlike, marker gene study involving analysis of specific region of genome, metagenomics is study of complete set of genomes from a cohort of microbes within an environmental sample. Though the third generation sequencing techniques (PacBio and Oxford Nanopore) have come up with longer reads, the assembly of overlapping reads into continuous or semi-continuous genome fragments also known as contigs or scaffolds is essential. Assembly of the reads allows an exhaustive analysis by reliably assigning specific functions or taxa as compared to gene fragments or unassembled reads. This step can be accomplished using any of the two approaches viz. *de novo* or reference based/guided. While a reference based approach is suitable in the presence of already sequenced reference genome to be used as a road map for genome reconstruction, a *de novo* assembly is preferred in the absence of the same. Assemblers are further categorized based on the type of algorithm they implement (i) de-bruijn graph based & (ii) Overlap Consensus Layout (OLC) (Li *et al.*, 2012). De-bruijn graph works by dividing reads into subsequences (kmers), making it appropriate approach for short reads. While, OLC works best with long reads (>300 bp). Tools such as Newbler (Chaisson and Pevzner, 2008), MIRA (Chevreux *et al.*, 1999) or are mostly employed for performing referenced based assemblies. Traditionally, for *de novo* assembly, assemblers such as SOAPdenovo2 (Luo *et al.*, 2012), SPAdes (Bankevich *et al.*, 2012), Velvet (Zerbino and Birney, 2008) and commercial tool like CLC genomics workbench are popularly used. However, for metagenomic samples, assembly poses challenges due to various reasons. Relative abundance of microorganisms in a sample is not always even, which results in some genomes being sequenced at a greater depth and coverage as compared to the rest. This problem usually boils down to either misassembling of metagenome or entirely missing out the less abundant species. This scenario worsens when sample contains microbes of related strain sharing most of their genomic content. Another challenge for metagenomic assembly is the computational constraint that is inherent with the diversity of microbes (Howe and Chain, 2015). To tackle these problems, the improved class of assembly tools, such as metaSPAdes (Nurk *et al.*, 2017), MEGAHIT v1.0 (Li *et al.*, 2016), IDBA-UD (Peng *et al.*, 2012), MetaVelvet (Namiki *et al.*, 2012), Meta Ray (Boisvert *et al.*, 2012) and Omega (Haider *et al.*, 2014) were developed. Recently developed tool, metaSPAdes address all these issues and provides competent result capturing low abundance microbes but, it cannot handle multiple libraries at a time. Whereas IDBA-UD takes uncompressed fasta files as input and doesn't support fastq format which is most common output from the latest NGS platforms. Similar is the issue with Omega which accepts uncompressed fasta and fastq format, but neither supports its compressed version nor aid with multithreading. On the other hand, Omega works on OLC approach preferred by researchers working with long reads. Few tools like Metavelvet and Megahit, are required to specify maximum kmer to support during or before source code compilation. Assembler results vary greatly depending upon the sample and its microbe content (simple or complex). However, recent study by Vollmers *et al.* (2017) shows that tools like IDBA-UD, Megahit and metaSPAdes employing multi kmer approach provide better results in terms of diversity captured when dealing with sample having high complexity. Still, metagenome assembly has scope for improvements in terms of computational performance and accuracy of assembling reads.

(2) Binning

Some of the most common challenges to metagenomic assembly are the uneven coverage across the sample, errors introduced during sequencing and stretches of repeated sequences (Brubach *et al.*, 2017). Hence, another approach called "binning" has been introduced which essentially means to group reads or contigs into so called "bins" and assigning them to operational taxonomic units (OTUs). The primary objective of binning is to group together the fragments belonging to same organism based on information such as sequence similarity, sequence composition or read coverage (Dröge and McHardy, 2012). Clustering and data representation are the two components of binning; where clustering involves bringing together the fragments like contigs or scaffolds followed by grouping these clusters with data representation approaches into taxonomic bins (Sangwan *et al.*, 2016). This approach is considered better for capturing novel microbes in the sample. However, precision of binning highly depends upon the quality of metagenome assembly. According to the availability of pre-existing bins derived from the genomics sequences, the binning algorithms are further classified into (i) supervised binning, when bins are already available & (ii) unsupervised binning, when no pre-existing bins are available. There are a number of tools and pipeline that implements binning process such as MG-RAST (Glass *et al.*, 2010), TACAO (Diaz *et al.*, 2009), MetaPhyler (Liu *et al.*, 2010), PhyloPythia (Patil *et al.*, 2012), SOrt-ITEMS (Monzoorul Haque *et al.*, 2009), IMG/M (Markowitz *et al.*, 2013), CARMA3 (Gerlach and Stoye, 2011), S-GSOM (Chan *et al.*, 2008), MEGAN (Huson *et al.*, 2007) and PhymmBL (Brady and Salzberg, 2009) etc. Use of long reads generated from 3rd generation sequencing platform like Pacbio RS II for metagenome analysis can result in improvement of assembly which in turn can help achieving accurate binning and taxonomy assignment (Frank *et al.*, 2016). Tools like CARMA3 and PhymmBL uses combination of two approaches for the process of binning from which one is BLAST, thus it provides added advantage in taxonomic classification (Teeling and Glöckner, 2012).

(3) Annotation:

One of the most important steps in gut microbiome study is the identification and subsequent annotation of the coding regions within the metagenome. A number of tools are available for gene identification such as MetaGeneMark (Zhu *et al.*,

2010a,b), Prodigal (Hyatt *et al.*, 2010) and Metagene (Noguchi *et al.*, 2006). Another tool called FragGeneScan (Rho *et al.*, 2010) has been implemented in MG/RAST and IMG-ER which is considered better as compared to Metagene because it can handle sequencing error introduced by NGS technology. Additionally, it can work at read as well as assembly level. Non-coding RNAs such as tRNAs are commonly identified using the tools like tRNA-Scan (Lowe and Eddy, 1997). However, rRNAs can be predicted by comparing sequences to curated rRNA sequence database like Greengenes, SILVA and RDP. Functional annotation is performed by aligning the coding sequences against reference databases to perform homology searches usually with BLAST program. A recently developed alternative for blastx/blastp is Diamond (Buchfink *et al.*, 2015) which is 20,000 times faster than BLAST and is very often used for larger datasets. Of course, high performance computing clusters or commercial cloud computing services like Amazon EC2 (Walker, 2008) have significantly speeded up the time taken to perform these tasks. In the list of standalone programs, COGNIZER (Bose *et al.*, 2015) is an impressive addition since it enables direct derivation of KEGG, Pfam and GO from COG categorizations. Additionally clustered regularly interspaced short palindromic repeats (CRISPRs), a regulatory element can also be identified using specific tools like CRISPR recognition tool (CRT) (Bland *et al.*, 2007). The main limitation of most of these annotation tools is that they assume metagenome data to contain more of bacterias and archeas and thus eukaryotes like fungus having complex genomic structure interspersed with intron are ignored to some extent.

Case Study: A Core Gut Microbiome in Obese and Lean Twins

As represented by Mandal *et al.* (2015), many publications have demonstrated the association of gut microbes with obesity as compared to other diseases. Thus, here we present a case study carried out by Turnbaugh *et al.*, in 2008 considering obese and lean twins to show the effect of gut microbiota on obesity/leanness. Two set of individuals were examined in this study (i) 154 individuals (31 monozygotic twin pairs, 23 dizygotic twin pairs and 46 samples of their mothers (if available) at two time points and (ii) 18 individuals (3 lean and 3 obese European ancestry monozygotic twin pairs along with their mothers). Monozygotic, dizygotic co-twins and parent-offspring pairs were examined for research as they provide an appealing model for determining the core gut microbes and influence of genotype and shared environment on microbe composition.

Molecular technique and sequencing:

Set (i) of 154 individual samples: 16S rRNA sequencing approach using ABI 3730xl capillary sequencer as well as 16S rRNA V2 variable region13 and V6 hypervariable region1 using Roche 454 FLX instrument.

Set (ii) of 18 individual samples: Shotgun pyrosequencing using Roche 454 FLX/Titanium.

Findings:

- Not a single phylotype was found among all the sample whose abundance was more than 0.5%.
- Samples taken from the same individual at different time point's subsequently exhibited stable phylotypes. However variation was witness in relative abundance of the principal gut bacterial phyla.
- Obesity was marked by notable decrease in the level of diversity. Besides lower proportion of Bacteroidetes and a higher proportion of Actinobacteria in obese compared with lean individuals of both ancestries.
- High level of similarity at functional metabolic level was observed in contrary to excessive diversity concerning relative abundance of bacterial phylum.

This study revealed that, rather than at the phylotype level, the core gut microbiome prevail at the level of shared genes and metabolites.

Application of Gut Microbiome in Healthcare and Medicine

With the increased incidence of use/misuse of antibiotics as a treatment for different diseases, some major health issues have been raised. It is well established fact that bacteria are gradually developing resistance against many antibiotics. In Salyers *et al.* (2004) revealed that gut microbiota may act as a reservoir of antibiotic resistance when antibiotics pass through the colon. Widely used broad spectrum antibiotics as a side effect kills commensal and 'beneficial bacteria' along with pathogens thus influencing complete microbial community in gut (Shetty *et al.*, 2013). In adults, disruption of gut microbes may lead to pathogenic infection (Robinson and Young, 2010). Antibiotic like ciprofloxacin, when used in excess can lead to reduction in diversity and evenness of bacterial community in gut (Dethlefsen *et al.*, 2008). Hence knowing the fact that antibiotics can derange the protective microbes and increase susceptibility to infection, highlighted the importance of so called 'microbial interference treatment' (Bengmark, 1998). In last decades, our insight of gut microbes and their role in health and pathophysiology has refined a lot. With increased research, it is known that some disease condition can even be treated or improved using specific types of bacteria as a food component or supplement known as "probiotics" (AFRC, 1989). This term is derived from the Greek word 'biotikos' meaning 'for life' i.e., it is used to denote microbes that upon ingestion in requisite amount promotes well being beyond those of inherent fundamental nutrition, boosting gastrointestinal (GI) system (Conly and Johnston, 2004). They are likely to show positive impact upon a range of acute and chronic pathologies (Table 1). The health

Table 1 Diseases and their associated bacteria.

Disease	Name of prevalent bacteria	Reference	Probiotics
Type 2 diabetes	Akkermansia muciniphila	Qin <i>et al.</i> (2012)	Lactobacillus acidophilus, Lactobacillus rhamnosus, Lactobacillus casei, Bacillus longum, Bacillus breve, Lactobacillus sporogenes (Zhang, Q., Wu, Y. and Fei, X., 2016)
	Bacteroides intestinalis	Mandal <i>et al.</i> (2015)	
	Bacteroides sp. 20_3		
	Clostridium botteae		
	Clostridium ramosum		
	Clostridium sp. HGF2		
	Clostridium symbiosum		
	Clostridium hathewayi		
	Desulfovibrio sp. 3_1_syn3		
	Eggerthella lenta		
Obesity/IBD/CD	Escherichia coli		Bacteroides spp. (Walker and Parkhill, 2013)
	Acidimicrobiae ellin 7143	Frank <i>et al.</i> (2007)	
	Actinobacterium GWS-BW-H99	Joossens <i>et al.</i> (2011)	
	Actinomyces oxydans	Zhang <i>et al.</i> (2015)	
	Bacillus licheniformis	Mandal <i>et al.</i> (2015)	
	Drinking water bacterium Y7		
	Gamma proteobacterium DD103		
	Nocardioides sp. NS/27		
	Novosphingobium sp. K39		
	Pseudomonas straminea		
Colorectal cancer	Sphingomonas sp. A01		Bifidobacterium longum (Singh <i>et al.</i> , 1997)
	Ruminococcus gnavus		
	Acinetobacter johnsonii	Mandal <i>et al.</i> (2015)	
	Anaerococcus murdochii	Weir <i>et al.</i> (2013)	
	Bacteroides fragilis	Hemrajata and Versalovic (2013)	
	Bacteroides vulgatus	Wang <i>et al.</i> (2012)	
	Butyrate-producing bacterium A2-166	Wu <i>et al.</i> (2009)	
	Dialister pneumosintes	Balamurugan <i>et al.</i> (2008)	
	Enterococcus faecalis		
	Fusobacterium nucleatum E9_12		
	Fusobacterium periodonticum		
	Gemella morbillorum		
	Lachnospira pectinoschiza		
	Parvimonas micra ATCC 3.3270		
	Peptostreptococcus stomatis		
	Shigella sonne		
	Acidaminobacter		
	Phascolarctobacterium		
	Citrobacter farmer		
	Akkermansia muciniphila		

(Continued)

Table 1 Continued

Disease	Name of prevalent bacteria	Reference	Probiotics
Prostate cancer	<i>Helicobacter hepaticus</i>	Poutahidis <i>et al.</i> (2013)	<i>Lactobacillus rhamnosus</i> GR-1 (Hemsworth <i>et al.</i> , 2011)
HIV	<i>Escherichia coli</i>	Zajac <i>et al.</i> (2007)	
	<i>Proteus mirabilis</i>	Vujkovic-Cvijin <i>et al.</i> (2013)	
	<i>Citrobacter freundii</i>	Gori <i>et al.</i> (2008)	
	<i>Staphylococcus</i> spp.	Wolf <i>et al.</i> (1998)	
	<i>Enterobacter aerogenes</i>		
	<i>Salmonella</i> spp.		
	<i>Serratia</i> spp.		
	<i>Shigella</i> spp.		
	<i>Klebsiella</i> spp.		
Autism	<i>Campylobacter</i> spp.		
	<i>Candida albicans</i>		
	<i>Pseudomonas aeruginosa</i>		
	<i>Clostridium</i> spp.	Heberling <i>et al.</i> (2013)	
	<i>Bacteroides vulgatus</i>	De Angelis <i>et al.</i> (2013)	
	<i>Desulfovibrio desulfuricans</i>	Finigold (2011)	
	<i>Desulfovibrio fairfieldensis</i>	Song <i>et al.</i> (2004)	
	<i>Desulfovibrio piger</i>		
	<i>Caloramator</i> spp.		
	<i>Sarcina</i> spp.		
	<i>Alistipes</i> spp.		
	<i>Akkermansia</i> spp.		
Chronic heart disease	<i>Escherichia coli</i>	Krack <i>et al.</i> (2005)	
	<i>Klebsiella pneumonia</i>		
	<i>Streptococcus viridians</i>		
Symptomatic atherosclerosis	<i>Escherichia coli</i>	Karlsson <i>et al.</i> (2012)	
	<i>Eubacterium rectale</i>	Mandal <i>et al.</i> (2015)	
	<i>Eubacterium siraeum</i>		
	<i>Faecalibacterium prausnitzii</i>		
	<i>Ruminococcus bromii</i>		
	<i>Ruminococcus</i> sp. 5_1_39BFAA		
Liver disease (Nonalcoholic steatohepatitis-NASH)	<i>Escherichia coli</i>	Wu <i>et al.</i> (2008)	
Infectious diarrhea (children)	<i>Escherichia coli</i>	Sánchez <i>et al.</i> (2017)	<i>Lactobacillus rhamnosus</i> GG
	<i>Salmonella</i> spp.		<i>Saccharomyces boulardii</i> and <i>Lactobacillus reuteri</i> DSM 1,7938
	<i>Campylobacter</i> spp.		(Szajewska <i>et al.</i> , 2014)
	<i>Shigella</i> spp.		

Note: Modified from Mandal, R.S., Saha, S., 2015. Metagenomic surveys of gut microbiota. Genomics, proteomics & bioinformatics 13 (3),148–158.

benefits of these bacteria are strain specific and thus even a change of strain can affect the outcome (Guarner and Malagelada, 2003). The evidence that, probiotics can be used as effective treatment came after studying patients suffering from diarrhoea. It was observed that in case of acute gastroenteritis in children, the use of either *Lactobacillus rhamnosus* or *Lactobacillus reuteri*, or *Lactobacillus casei* significantly decreased the duration of diarrhoea (Raza *et al.*, 1995; Shornikova *et al.*, 1997; Pedone *et al.*, 1999). It is assumed that diet induced obesity might also be reversed with use of specific diet or by manipulation of gut microbiota by mean of probiotics. Ait-belgnaoui *et al.* (2009) have shown that *Lactobacillus farciminis* taken as a probiotics can affect spinal neuronal activation. Supply of specific lactobacillus along with oat fiber was observed to significantly lower infections associated with pancreatitis, liver transplants and abdominal surgery (Rayes *et al.*, 2005; Rayes, 2004; Olah *et al.*, 2002). Consequently, probiotics can either be used as a replacement or to augment antibiotic effect to re-impose and promote the beneficial microbes in our bodies against diseases condition (Reid, 2006). Deciphering all these advantageous effects, probiotics can be considered as important part of overall care of the patient. Therefore, the study of gut microbiome in different population is vital to apprehend the mechanisms of action and basal of our microbiota for the development of indigenous probiotic strains to be further used for improvement of disease condition.

Gibson and Roberfroid (1995), first introduced the term “prebiotic” explained as nondigestible food ingredient when ingested in body promotes host well-being by nourishing the growth and/or activity of one or a confined number of bacteria in the colon. Prebiotics may include oligosaccharides (like oligofructose, galacto-oligosaccharides) peptides and lipids which could alter endogenous biochemical pathways, by remodeling the specific intestinal microbial communities residing in the colon (Versalovic and Wilson, 2008). List of prebiotics may extend to use of some type resistant starches and sugar alcohols. They may work by blocking the pathogen binding and providing nutrients to useful gut microbes. To infer the mechanism of working of prebiotics, altered activities can be evaluated by structural to functional studies using metagenomics, metaproteomics and metabolomics approach. It was observed that, when transgalactosylated disaccharides was used as a prebiotics, there was increase in number of Bifidobacteria and Lactobacillus whereas decrease in Bacteroides sp. and Candida sp. (Ito *et al.*, 1993). Only side effect of use of prebiotics is increased gas production. However when dealing with prebiotics one need to focus more on understanding its sufficiency, gut passage time and osmotic regularity. Similar to probiotics, studies to determine effective dosage and duration of probiotic exposure showing its effect is necessitate (Reid *et al.*, 2003).

When combination of probiotics with prebiotics is used together for management of microflora, they are known as synbiotics which exemplifies synergism. Theoretically, if prebiotics are taken together with suited probiotics, they may boost the activity of probiotics and their survival along with GI inhabitant microbiota. It is known that synbiotics elevate the number of balanced gut microbiota and also improve immunomodulating ability (Zhang *et al.*, 2010).

However, currently the process of Fecal Microbiota Transplantation (FMT) is in trend. It implicates transplantation of fecal microbiont from healthy individual to recipient in order to refurbish healthy microbiota community and thus deliberate its health benefits (Bakken *et al.*, 2011; Smits *et al.*, 2013). Although FMT was first introduced in year 1958 by Eiseman *et al.* for the treatment of pseudomembranous colitis (Eiseman *et al.*, 1958), but recently it received attention when it offered promising solution against recurring *Clostridium difficile* infection (Smits *et al.*, 2013; Bakken *et al.*, 2011). Due to some trepidations associated with FMT like patient acceptance, infectious potential and treatment standardization, its use is restricted only in case of failure of traditional therapies (Ji and Nielsen, 2015). Being natural and considerably economical treatment, it is favorable to use (Kelly *et al.*, 2015). It is anticipated that in near future, FMT may come-up as a fruitful treatment for other conditions like IBD, obesity, metabolic syndrome and functional gastrointestinal disorders (Gupta *et al.*, 2016).

Future of Gut Microbiome Research

Following the discussion so far, it is evident that studying the gut microbiome has significantly changed our views on how we associate health conditions with their root causes (Zhang *et al.*, 2015). The deeper we have dug into it, the better we have understood the striking correlations of gut microbiota with chronic (Guinane and Cotter, 2013) as well as acute disorders (Shukla *et al.*, 2017). With a detailed understanding of the human microbiome landscape, the possibility of synthetic microbiome modulating the health should not come as a surprise (Waldor *et al.*, 2015). New attempts have already been taken for microbiome replacement therapies, both from natural (Vrieze *et al.*, 2012) and synthetic source (Petrof *et al.*, 2013) with promising results. However, further research is of course required for it to become a routine medical practice and overcoming the inherent limitations. Researchers are further narrowing down stool microbiota ensemble to scientifically validated set of beneficial or rather remedial microorganisms which may be a paradigm shift in medical practices. A recent UNICEF survey (Source: UNICEF/WHO/World Bank Joint Child Malnutrition Estimates, May 2017 edition) suggests that nearly half of all deaths in children under 5 years attributes to under nutrition which translates into the unnecessary loss of about 3 million young lives a year. One can imagine the outcome of such studies to repair the defective gut of such undernourished children along with the food restoring health. Finally, we may also redefine the formal definition of the food we consume by essentially adding the details of gut microbiome associations. With all that said, there are certain aspects which need to be thoroughly considered while drawing conclusions. Firstly, gut studies are mostly carried out by fecal sampling which only represents the luminal microbiota which considerably differs from the mucosa-associated microbiota. Secondly, it is very crucial for viral and fungal species to be taken into consideration along

with the bacterial species while studying the gut microbiome. And finally, a large numbers of varying human populations needs to be considered for sampling to establish microbiome-health relationships using combination of techniques to screen transcripts, proteins and metabolites along with the genomic sequences. On the technical side, there is a need for fast paced high performance yet sensitive bioinformatics tools to capture the information as efficiently as possible. The good news is that bioinformatics tools and methods to analyze the microbiome data are evolving in parallel with next generation wet-lab techniques to extract required samples. Hence, there is no doubt that computational techniques will continue to play a major role in the extraction of meaningful information from constantly piling up data.

Closing Remarks

With a number of human gut profiling studies being carried out, it is evident that microbes present in GI has major role to play in human well-being and disease condition. A large number of collaborative projects have been initiated to dig deeper into the understanding of function of these microbes and how their composition is impacted by diet, age, environment etc. Advancement of sequencing methodology, dropping sequencing cost and availability of easy to use bioinformatics tools has encouraged analysis of large number of gut metagenome samples simultaneously. Phylogenetic marker gene studies using next generation sequencing are routinely undertaken as a first choice to get taxonomy profile of a gut sample, answering basic question “who is there”. But owing to the limitation of 16S rDNA gene study to answer some of the crucial question like ‘how they function’ and ‘how they interact with gut ecosystem’, approaches including whole metagenome, metatranscriptome, metaproteome and metabolomic are required. Thus to get a bigger and clear picture of overall activities of microbes and their involvement in human health, it is necessary to understand which genes are expressed and which metabolites are produced by particular microbe community. Advance bioinformatics analysis and statistical tools and software have made it possible to decode biological information from large number of samples to understand mechanism and mode of action of these microbes. With the ever increasing amount of data generation for gut profiling, certain limitations and issues of existing tools need to be overcome to meet the ever increasing demand. More and more information collected for these microbes is also helping to extend the dimension of their potential and use in health care industries. Thereby, it has given a strong hint that gut microbes study cannot be ignored while moving towards the era of personalized medicine.

Acknowledgment

We would like to thank Xcelris Labs Ltd., Ahmedabad, India for providing the resources and support. We would also like to thank the bioinformatics team at Xcelris Labs for their constant support and motivation.

See also: Computational Pipelines and Workflows in Bioinformatics. Constructing Computational Pipelines. Ecosystem Monitoring Through Predictive Modeling. Functional Enrichment Analysis. Information Retrieval in Life Sciences. Integrative Analysis of Multi-Omics Data. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Metagenomic Analysis and its Applications. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Sequence Analysis. Sequence Composition. Studies of Body Systems. Whole Genome Sequencing Analysis

References

- Abrahamsson, T.R., Jakobsson, H.E., Andersson, A.F., *et al.*, 2014. Low gut microbiota diversity in early infancy precedes asthma at school age. *Clinical & Experimental Allergy* 44 (6), 842–850.
- AFRC, R.F., 1989. Probiotics in man and animals. *Journal of Applied Microbiology* 66 (5), 365–378.
- Ait-belgnaoui, A., Eutamene, H., Houdeau, E., *et al.*, 2009. *Lactobacillus farciminis* treatment attenuates stress-induced overexpression of Fos protein in spinal and supraspinal sites after colorectal distension in rats. *Neurogastroenterology & Motility* 21 (5), 567.
- Alard, J., Lehrter, V., Rhimi, M., *et al.*, 2016. Beneficial metabolic effects of selected probiotics on diet-induced obesity and insulin resistance in mice are associated with improvement of dysbiotic gut microbiota. *Environmental Microbiology* 18 (5), 1484–1497.
- Allali, I., Arnold, J.W., Roach, J., *et al.*, 2017. A comparison of sequencing platforms and bioinformatics pipelines for compositional analysis of the gut microbiome. *BMC Microbiology* 17 (1), 194.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic acids Research* 25 (17), 3389–3402.
- Aronesty, E., 2011. *ea-utils: Command-Line Tools for Processing Biological Sequencing Data*. Durham, NC: Expression Analysis.
- Arumugam, M., Raes, J., Pelletier, E., *et al.*, 2011. Enterotypes of the human gut microbiome. *Nature* 473 (7346), 174.
- Ashelford, K.E., Chuzhanova, N.A., Fry, J.C., Jones, A.J., Weightman, A.J., 2005. At least 1 in 20 16S rRNA sequence records currently held in public repositories is estimated to contain substantial anomalies. *Applied and Environmental Microbiology* 71 (12), 7724–7736.
- Bakken, J.S., Borody, T., Brandt, L.J., *et al.*, 2011. Treating *Clostridium difficile* infection with fecal microbiota transplantation. *Clinical Gastroenterology and Hepatology* 9 (12), 1044–1049.

- Balamurugan, R., Rajendiran, E., George, S., Samuel, G.V., Ramakrishna, B.S., 2008. Real-time polymerase chain reaction quantification of specific butyrate-producing bacteria, *Desulfovibrio* and *Enterococcus faecalis* in the feces of patients with colorectal cancer. *Journal of Gastroenterology and Hepatology* 23 (8pt1), 1298–1303.
- Bankovich, A., Nurk, S., Antipov, D., *et al.*, 2012. SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology* 19 (5), 455–477. BBMap - Bushnell B. - sourceforge.net/projects/bbmap/.
- Bengmark, S., 1998. Ecological control of the gastrointestinal tract. The role of probiotic flora. *Gut* 42 (1), 2–7.
- Bland, C., Ramsey, T.L., Sabree, F., *et al.*, 2007. CRISPR recognition tool (CRT): A tool for automatic detection of clustered regularly interspaced palindromic repeats. *BMC Bioinformatics* 8 (1), 209.
- Boisvert, S., Raymond, F., Godzaridis, É., Laviolette, F., Corbeil, J., 2012. Ray Meta: Scalable de novo metagenome assembly and profiling. *Genome Biology* 13 (12), R122.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30 (15), 2114–2120.
- Bose, T., Hague, M.M., Reddy, C.V.S.K., Mande, S.S., 2015. COGNIZER: A framework for functional annotation of metagenomic datasets. *PLOS ONE* 10 (11), e0142102.
- Brady, A., Salzberg, S.L., 2009. Phymm and PhymmBL: Metagenomic phylogenetic classification with interpolated Markov models. *Nature Methods* 6 (9), 673–676.
- Brubach, B., Ghurye, J., Pop, M., Srinivasan, A., 2017. Better greedy sequence clustering with fast banded alignment. In: *Proceedings of the LIPIcs-Leibniz International Proceedings in Informatics* (vol. 88). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Buchfink, B., Xie, C., Huson, D.H., 2015. Fast and sensitive protein alignment using DIAMOND. *Nature Methods* 12 (1), 59–60.
- Call, D.R., Borucki, M.K., Loge, F.J., 2003. Detection of bacterial pathogens in environmental samples using DNA microarrays. *Journal of Microbiological Methods* 53 (2), 235–243.
- Caporaso, J.G., Bittinger, K., Bushman, F.D., *et al.*, 2009. PyNAST: A flexible tool for aligning sequences to a template alignment. *Bioinformatics* 26 (2), 266–267.
- Caporaso, J.G., Kuczynski, J., Stombaugh, J., *et al.*, 2010. QIIME allows analysis of high-throughput community sequencing data. *Nature Methods* 7 (5), 335–336.
- Chaisson, M.J., Pevzner, P.A., 2008. Short read fragment assembly of bacterial genomes. *Genome Research* 18 (2), 324–330.
- Chan, C.K.K., Hsu, A.L., Halgamuge, S.K., Tang, S.L., 2008. Binning sequences using very sparse labels within a metagenome. *BMC Bioinformatics* 9 (1), 215.
- Chevreaux, B., Wetter, T., Suhai, S., 1999. Genome sequence assembly using trace signals and additional sequence information. *German Conference on Bioinformatics* 99, 45–56.
- Clooney, A.G., Fouhy, F., Sleator, R.D., *et al.*, 2016. Comparing apples and oranges?: Next generation sequencing and its impact on microbiome analysis. *PLOS ONE* 11 (2), e0148028.
- Conly, J.M., Johnston, B.L., 2004. Coming full circle: From antibiotics to probiotics and prebiotics. *Canadian Journal of Infectious Diseases and Medical Microbiology* 15 (3), 161–163.
- De Angelis, M., Piccolo, M., Vannini, L., *et al.*, 2013. Fecal microbiota and metabolome of children with autism and pervasive developmental disorder not otherwise specified. *PLOS ONE* 8 (10), e76993.
- DeSantis, T.Z., Brodie, E.L., Moberg, J.P., *et al.*, 2007. High-density universal 16S rRNA microarray analysis reveals broader diversity than typical clone library when sampling the environment. *Microbial Ecology* 53 (3), 371–383.
- DeSantis, T.Z., Hugenholtz, P., Larsen, N., *et al.*, 2006a. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology* 72 (7), 5069–5072.
- DeSantis, T.Z., Hugenholtz, P., Keller, K., *et al.*, 2006b. NAST: A multiple sequence alignment server for comparative analysis of 16S rRNA genes. *Nucleic Acids Research* 34 (Suppl_2), W394–W399.
- DeThlefsen, L., Huse, S., Sogin, M.L., Relman, D.A., 2008. The pervasive effects of an antibiotic on the human gut microbiota, as revealed by deep 16S rRNA sequencing. *PLOS Biology* 6 (11), e280.
- Diaz, N.N., Krause, L., Goesmann, A., Niehaus, K., Nattkemper, T.W., 2009. TACO – Taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach. *BMC Bioinformatics* 10 (1), 56.
- Dröge, J., McHardy, A.C., 2012. Taxonomic binning of metagenome samples generated by next-generation sequencing technologies. *Briefings in Bioinformatics* 13 (6), 646–655.
- Eckburg, P.B., Bik, E.M., Bernstein, C.N., *et al.*, 2005. Diversity of the human intestinal microbial flora. *Science* 308 (5728), 1635–1638.
- Edgar, R.C., 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26 (19), 2460–2461.
- Edgar, R.C., 2013. UPARSE: Highly accurate OTU sequences from microbial amplicon reads. *Nature Methods* 10 (10), 996–998.
- Edgar, R.C., 2017. Updating the 97% identity threshold for 16S ribosomal RNA OTUs. *bioRxiv*. 192211.
- Edgar, R.C., Haas, B.J., Clemente, J.C., Quince, C., Knight, R., 2011. UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics* 27 (16), 2194–2200.
- Eiseman, Á., Silen, W., Bascom, G.S., Kauvar, A.J., 1958. Fecal enema as an adjunct in the treatment of pseudomembranous enterocolitis. *Surgery* 44 (5), 854–859.
- Ferreira, C.M., Vieira, A.T., Vinolo, M.A.R., *et al.*, 2014. The central role of the gut microbiota in chronic inflammatory diseases. *Journal of Immunology Research*. 2014.
- Finelgold, S.M., 2011. State of the art; microbiology in health and disease intestinal bacterial flora in autism. *Anaerobe* 17 (6), 367–368.
- Fraher, M.H., O’toole, P.W., Quigley, E.M., 2012. Techniques used to characterize the gut microbiota: A guide for the clinician. *Nature Reviews Gastroenterology and Hepatology* 9 (6), 312–322.
- Frank, D.N., Amand, A.L.S., Feldman, R.A., *et al.*, 2007. Molecular-phylogenetic characterization of microbial community imbalances in human inflammatory bowel diseases. *Proceedings of the National Academy of Sciences* 104 (34), 13780–13785.
- Frank, J.A., Pan, Y., Tooming-Klunderud, A., *et al.*, 2016. Improved metagenome assemblies and taxonomic binning using long-read circular consensus sequence data. *Scientific Reports* 6, 25373.
- Franzosa, E.A., Hsu, T., Sirota-Madi, A., *et al.*, 2015. Sequencing and beyond: Integrating molecular ‘omics’ for microbial community profiling. *Nature Reviews Microbiology* 13 (6), 360.
- Frey, K.G., Herrera-Galeano, J.E., Redden, C.L., *et al.*, 2014. Comparison of three next-generation sequencing platforms for metagenomic sequencing and identification of pathogens in blood. *BMC Genomics* 15 (1), 96.
- Ganju, P., Nagpal, S., Mohammed, M.H., *et al.*, 2016. Microbial community profiling shows dysbiosis in the lesional skin of Vitiligo subjects. *Scientific Reports* 6, 18761.
- Gerlach, W., Stoye, J., 2011. Taxonomic classification of metagenomic shotgun sequences with CARMA3. *Nucleic Acids Research* 39 (14), e91.
- Gest, H., 2004. The discovery of microorganisms by Robert Hooke and Antoni Van Leeuwenhoek, fellows of the Royal Society. *Notes and Records of the Royal Society* 58 (2), 187–201.
- Gibson, G.R., Roberfroid, M.B., 1995. Dietary modulation of the human colonic microbiota: Introducing the concept of prebiotics. *The Journal of Nutrition* 125 (6), 1401.
- Glass, E.M., Wilkening, J., Wilke, A., Antonopoulos, D., Meyer, F., 2010. Using the metagenomics RAST server (MG-RAST) for analyzing shotgun metagenomes. *Cold Spring Harbor Protocols* 2010 (1), pp.pdb-prot5368.
- Gori, A., Tincati, C., Rizzardini, G., *et al.*, 2008. Early impairment of gut function and gut flora supporting a role for alteration of gastrointestinal mucosa in human immunodeficiency virus pathogenesis. *Journal of Clinical Microbiology* 46 (2), 757–758.
- Gower, J.C., 1975. Generalized procrustes analysis. *Psychometrika* 40 (1), 33–51.
- Grice, E.A., Segre, J.A., 2012. The human microbiome: Our second genome. *Annual Review of Genomics and Human Genetics* 13, 151–170.
- Guarner, F., Malagelada, J.R., 2003. Gut flora in health and disease. *The Lancet* 361 (9356), 512–519.
- Guinane, C.M., Cotter, P.D., 2013. Role of the gut microbiota in health and chronic gastrointestinal disease: Understanding a hidden metabolic organ. *Therapeutic Advances in Gastroenterology* 6 (4), 295–308.
- Gupta, S., Allen-Vercoe, E., Petrof, E.O., 2016. Fecal microbiota transplantation: In perspective. *Therapeutic Advances in Gastroenterology* 9 (2), 229–239.

- Haas, B.J., Gevers, D., Earl, A.M., *et al.*, 2011. Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Research* 21 (3), 494–504.
- Haider, B., Ahn, T.H., Bushnell, B., *et al.*, 2014. Omega: An overlap-graph de novo assembler for metagenomics. *Bioinformatics* 30 (19), 2717–2722.
- Hamady, M., Knight, R., 2009. Microbial community profiling for human microbiome projects: Tools, techniques, and challenges. *Genome Research* 19 (7), 1141–1152.
- Handelsman, J., 2004. Metagenomics: Application of genomics to uncultured microorganisms. *Microbiology and Molecular Biology Reviews* 68 (4), 669–685.
- Handelsman, J., 2009. Metagenetics: Spending our inheritance on the future. *Microbial Biotechnology* 2 (2), 138–139.
- Hayashi, H., Sakamoto, M., Benno, Y., 2002. Phylogenetic analysis of the human gut microbiota using 16S rDNA clone libraries and strictly anaerobic culture-based methods. *Microbiology and Immunology* 46 (8), 535–548.
- Heberling, C.A., Dhurjati, P.S., Sasser, M., 2013. Hypothesis for a systems connectivity model of autism spectrum disorder pathogenesis: Links to gut bacteria, oxidative stress, and intestinal permeability. *Medical Hypotheses* 80 (3), 264–270.
- Hemarajata, P., Versalovic, J., 2013. Effects of probiotics on gut microbiota: Mechanisms of intestinal immunomodulation and neuromodulation. *Therapeutic Advances in Gastroenterology* 6 (1), 39–51.
- Hemsworth, J., Hekmat, S., Reid, G., 2011. The development of micronutrient supplemented probiotic yogurt for people living with HIV: Laboratory testing and sensory evaluation. *Innovative Food Science & Emerging Technologies* 12 (1), 79–84.
- Hodkinson, B.P., Grice, E.A., 2015. Next-generation sequencing: A review of technologies and tools for wound microbiome research. *Advances in Wound Care* 4 (1), 50–58.
- Howe, A., Chain, P.S., 2015. Challenges and opportunities in understanding microbial communities with metagenome assembly (accompanied by IPython Notebook tutorial). *Frontiers in Microbiology* 6.
- Huson, D.H., Auch, A.F., Qi, J., Schuster, S.C., 2007. MEGAN analysis of metagenomic data. *Genome Research* 17 (3), 377–386.
- Hyatt, D., Chen, G.L., LoCascio, P.F., *et al.*, 2010. Prodigal: Prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* 11 (1), 119.
- Ito, M., Kimura, M., Deguchi, Y., *et al.*, 1993. Effects of transgalactosylated disaccharides on the human intestinal microflora and their metabolism. *Journal of Nutritional Science and Vitaminology* 39 (3), 279–288.
- Janda, J.M., Abbott, S.L., 2007. 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: Pluses, perils, and pitfalls. *Journal of Clinical Microbiology* 45 (9), 2761–2764.
- Jandhyala, S.M., Talukdar, R., Subramanyam, C., *et al.*, 2015. Role of the normal gut microbiota. *World Journal of Gastroenterology* 21 (29), 8787.
- Ji, B., Nielsen, J., 2015. From next-generation sequencing to systematic modeling of the gut microbiome. *Frontiers in Genetics* 6, 219.
- Joossens, M., Huys, G., Cnockaert, M., *et al.*, 2011. Dysbiosis of the faecal microbiota in patients with Crohn's disease and their unaffected relatives. *Gut*, pp.gut-2010.
- Kang, S., Denman, S.E., Morrison, M., *et al.*, 2010. Dysbiosis of fecal microbiota in Crohn's disease patients as revealed by a custom phylogenetic microarray. *Inflammatory Bowel Diseases* 16 (12), 2034–2042.
- Karlsson, F.H., Fåk, F., Nookaew, I., *et al.*, 2012. Symptomatic atherosclerosis is associated with an altered gut metagenome. *Nature Communications* 3, 1245.
- Karlsson, F., Tremaroli, V., Nielsen, J., Bäckhed, F., 2013. Assessing the human gut microbiota in metabolic diseases. *Diabetes* 62 (10), 3341–3349.
- Kelly, C.R., Kahn, S., Kashyap, P., *et al.*, 2015. Update on fecal microbiota transplantation 2015: Indications, methodologies, mechanisms, and outlook. *Gastroenterology* 149 (1), 223–237.
- Kim, O.S., Cho, Y.J., Lee, K., *et al.*, 2012. Introducing EzTaxon-e: A prokaryotic 16S rRNA gene sequence database with phylotypes that represent uncultured species. *International Journal of Systematic and Evolutionary Microbiology* 62 (3), 716–721.
- Kirk, J.L., Beaudette, L.A., Hart, M., *et al.*, 2004. Methods of studying soil microbial diversity. *Journal of Microbiological Methods* 58 (2), 169–188.
- Knights, D., Lassen, K.G., Xavier, R.J., 2013. Advances in inflammatory bowel disease pathogenesis: Linking host genetics and the microbiome. *Gut* 62 (10), 1505–1510.
- Köljal, U., Larsson, K.H., Abarenkov, K., *et al.*, 2005. UNITE: A database providing web-based methods for the molecular identification of ectomycorrhizal fungi. *New Phytologist* 166 (3), 1063–1068.
- Krack, A., Sharma, R., Figulla, H.R., Anker, S.D., 2005. The importance of the gastrointestinal system in the pathogenesis of heart failure. *European Heart Journal* 26 (22), 2368–2374.
- Krueger, F., 2015. Trim galore. In: *Proceedings of a Wrapper Tool Around Cutadapt and FastQC to Consistently Apply Quality and Adapter Trimming to FastQ Files*.
- Langille, M.G., Zaneveld, J., Caporaso, J.G., *et al.*, 2013. Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nature Biotechnology* 31 (9), 814–821.
- Lan, Y., Wang, Q., Cole, J.R., Rosen, G.L., 2012. Using the RDP classifier to predict taxonomic novelty and reduce the search space for finding novel organisms. *PLOS ONE* 7 (3), e32491.
- Lederberg, Joshua., 2000. Infectious history. *Science* 288 (5464), 287–293.
- Li, D., Luo, R., Liu, C.M., *et al.*, 2016. MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods* 102, 3–11.
- Liu, B., Gibbons, T., Ghodsi, M., Pop, M., 2010. MetaPhyler: Taxonomic profiling for metagenomic sequences. In: *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 95–100. IEEE.
- Liu, B., Yuan, J., Yiu, S.M., *et al.*, 2012a. COPE: An accurate k-mer-based pair-end reads connection tool to facilitate genome assembly. *Bioinformatics* 28 (22), 2870–2874.
- Liu, L., Li, Y., Li, S., *et al.*, 2012b. Comparison of next-generation sequencing systems. *BioMed Research International*. 2012.
- Li, Z., Chen, Y., Mu, D., *et al.*, 2012. Comparison of the two major classes of assembly algorithms: Overlap – Layout – Consensus and de-bruijn-graph. *Briefings in Functional Genomics* 11 (1), 25–37.
- Lloyd-Price, J., Abu-Ali, G., Huttenhower, C., 2016. The healthy human microbiome. *Genome Medicine* 8 (1), 51.
- Lowe, T.M., Eddy, S.R., 1997. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Research* 25 (5), 955–964.
- Lozupone, C.A., Knight, R., 2008. Species divergence and the measurement of microbial diversity. *FEMS Microbiology Reviews* 32 (4), 557–578.
- Luo, R., Liu, B., Xie, Y., *et al.*, 2012. SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. *Gigascience* 1 (1), 18.
- Ly, N.P., Litonjua, A., Gold, D.R., Celedón, J.C., 2011. Gut microbiota, probiotics, and vitamin D: Interrelated exposures influencing allergy, asthma, and obesity? *Journal of Allergy and Clinical Immunology* 127 (5), 1087–1094.
- Magoč, T., Salzberg, S.L., 2011. FLASH: Fast length adjustment of short reads to improve genome assemblies. *Bioinformatics* 27 (21), 2957–2963.
- Maidak, B.L., Olsen, G.J., Larsen, N., *et al.*, 1996. The ribosomal database project (RDP). *Nucleic Acids Research* 24 (1), 82–85.
- Mandal, R.S., Saha, S., Das, S., 2015. Metagenomic surveys of gut microbiota. *Genomics, Proteomics & Bioinformatics* 13 (3), 148–158.
- Mardis, E.R., 2008. The impact of next-generation sequencing technology on genetics. *Trends in Genetics* 24 (3), 133–141.
- Markowitz, V.M., Chen, I.M.A., Palaniappan, K., *et al.*, 2013. IMG 4 version of the integrated microbial genomes comparative analysis system. *Nucleic Acids Research* 42 (D1), D560–D567.
- Martin, M., 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet Journal* 17 (1), 10–12.
- Masella, A.P., Bartram, A.K., Truszkowski, J.M., Brown, D.G., Neufeld, J.D., 2012. PANDAseq: Paired-end assembler for illumina sequences. *BMC Bioinformatics* 13 (1), 31.
- Mayer, E.A., Knight, R., Mazmanian, S.K., Cryan, J.F., Tillisch, K., 2014. Gut microbes and the brain: Paradigm shift in neuroscience. *Journal of Neuroscience* 34 (46), 15490–15496.
- McDonald, D., Price, M.N., Goodrich, J., *et al.*, 2012. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *The ISME Journal* 6 (3), 610–618.
- McWilliam, H., Li, W., Uludag, M., *et al.*, 2013. Analysis tool web services from the EMBL-EBI. *Nucleic Acids Research* 41 (W1), W597–W600.

- Medini, D., Serruto, D., Parkhill, J., *et al.*, 2008. Microbiology in the post-genomic era. *Nature Reviews Microbiology* 6 (6), 419.
- Metzker, M.L., 2010. Sequencing technologies – The next generation. *Nature Reviews Genetics* 11 (1), 31.
- Miele, L., Valenza, V., La Torre, G., *et al.*, 2009. Increased intestinal permeability and tight junction alterations in nonalcoholic fatty liver disease. *Hepatology* 49 (6), 1877–1887.
- Monzoorul Haque, M., Ghosh, T.S., Komanduri, D., Mande, S.S., 2009. SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences. *Bioinformatics* 25 (14), 1722–1730.
- Moreno-Indias, I., Cardona, F., Tinahones, F.J., Queipo-Ortuño, M.I., 2014. Impact of the gut microbiota on the development of obesity and type 2 diabetes mellitus. *Frontiers in Microbiology* 5.
- Musso, G., Gambino, R., Cassader, M., 2010. Obesity, diabetes, and gut microbiota. *Diabetes Care* 33 (10), 2277–2284.
- Namiki, T., Hachiya, T., Tanaka, H., Sakakibara, Y., 2012. MetaVelvet: An extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research* 40 (20), e155.
- Navas-Molina, J.A., Peralta-Sánchez, J.M., González, A., *et al.*, 2013. Advancing our understanding of the human microbiome using QIIME. *Methods in Enzymology* 531, 371.
- Noguchi, H., Park, J., Takagi, T., 2006. MetaGene: Prokaryotic gene finding from environmental genome shotgun sequences. *Nucleic Acids Research* 34 (19), 5623–5630.
- Norman, J.M., Handley, S.A., Baldridge, M.T., *et al.*, 2015. Disease-specific alterations in the enteric virome in inflammatory bowel disease. *Cell* 160 (3), 447–460.
- Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A., 2017. metaSPAdes: A new versatile metagenomic assembler. *Genome Research* 27 (5), 824–834.
- Olah, A., Belagyi, T., Issekutz, A., Gamal, M.E., Bengmark, S., 2002. Randomized clinical trial of specific lactobacillus and fibre supplement to early enteral nutrition in patients with acute pancreatitis. *British Journal of Surgery* 89 (9), 1103–1107.
- Oulas, A., Pavloudi, C., Polymenakou, P., *et al.*, 2015. Metagenomics: Tools and insights for analyzing next-generation sequencing data derived from biodiversity studies. *Bioinformatics and Biology Insights* 9, 75.
- Page, R.D., 2002. Visualizing phylogenetic trees using TreeView. In: *Proceedings of the Current Protocols in Bioinformatics*, pp. 6.2.1–6.2.15.
- Palmer, C., Bik, E.M., DiGiulio, D.B., 2007. Development of the human infant intestinal microbiota. *PLoS biology* 5 (7), p.e177.
- Patel, R.K., Jain, M., *et al.*, 2012. NGS QC toolkit: A toolkit for quality control of next generation sequencing data. *PLOS ONE* 7 (2), e30619.
- Patil, K.R., Roun, L., McHardy, A.C., 2012. The PhyloPythiaS web server for taxonomic assignment of metagenome sequences. *PLOS ONE* 7 (6), e38581.
- Pedone, C.A., Bernabeu, A.O., Postaire, E.R., Bouley, C.F., Reinert, P., 1999. The effect of supplementation with milk fermented by *Lactobacillus casei* (strain DN-114 001) on acute diarrhoea in children attending day care centres. *International Journal of Clinical Practice* 53 (3), 179–184.
- Peng, Y., Leung, H.C., Yiu, S.M., Chin, F.Y., 2012. IDBA-UD: A de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* 28 (11), 1420–1428.
- Petrof, E.O., Gloor, G.B., Vanner, S.J., *et al.*, 2013. Stool substitute transplant therapy for the eradication of *Clostridium difficile* infection: 'RePOOPulating' the gut. *Microbiome* 1 (1), 3.
- Poutahidis, T., Cappelle, K., Levkovich, T., *et al.*, 2013. Pathogenic intestinal bacteria enhance prostate cancer development via systemic activation of immune cells in mice. *PLOS ONE* 8 (8), e73933.
- Pruesse, E., Quast, C., Knittel, K., *et al.*, 2007. SILVA: A comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Research* 35 (21), 7188–7196.
- Qin, J., Li, Y., Cai, Z., *et al.*, 2012. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature* 490 (7418), 55–60.
- Qin, J., Li, R., Raes, J., *et al.*, 2010. A human gut microbial gene catalog established by metagenomic sequencing. *Nature* 464 (7285), 59.
- Quince, C., Lanzén, A., Davenport, R.J., Turnbaugh, P.J., 2011. Removing noise from pyrosequenced amplicons. *BMC Bioinformatics* 12 (1), 38.
- Rambaut, A., 2009. FigTree. v1. 3.1. Available at: <http://tree.bio.ed.ac.uk/software/figtree>.
- Rastogi, G., Sani, R.K., 2011. Molecular techniques to assess microbial community structure, function, and dynamics in the environment. In: *Proceedings of the Microbes and Microbial Technology*, pp.29–57. New York: Springer.
- Rayes, N., 2004. Lactobacilli and Fibers – A strong couple against bacterial infections in patients with major abdominal surgery. *Nutrition* 20 (6), 579–580.
- Rayes, N., Seehofer, D., Theruvath, T., *et al.*, 2005. Supply of pre-and probiotics reduces bacterial infection rates after liver transplantation – A randomized, double-blind trial. *American Journal of Transplantation* 5 (1), 125–130.
- Raza, S., Graham, S.M., Allen, S.J., *et al.*, 1995. Lactobacillus GG promotes recovery from acute nonbloody diarrhea in Pakistan. *The Pediatric Infectious Disease Journal* 14 (2), 107–111.
- Reid, G., 2006. Probiotics to prevent the need for, and augment the use of, antibiotics. *Canadian Journal of Infectious Diseases and Medical Microbiology* 17 (5), 291–295.
- Reid, G., Sanders, M.E., Gaskins, H.R., *et al.*, 2003. New scientific paradigms for probiotics and prebiotics. *Journal of Clinical Gastroenterology* 37 (2), 105–118.
- Rho, M., Tang, H., Ye, Y., 2010. FragGeneScan: Predicting genes in short and error-prone reads. *Nucleic Acids Research* 38 (20), e191.
- Robinson, C.J., Young, V.B., 2010. Antibiotic administration alters the community structure of the gastrointestinal microbiota. *Gut Microbes* 1 (4), 279–284.
- Rousseau, C., Butel, M.J., 2015. High-throughput techniques for studying of gut microbiota. *Advances in Probiotic Technology*. 146.
- Salys, A.A., Gupta, A., Wang, Y., 2004. Human intestinal bacteria as reservoirs for antibiotic resistance genes. *Trends in microbiology* 12 (9), 412–416.
- Sánchez, B., Delgado, S., Blanco-Míguez, A., *et al.*, 2017. Probiotics, gut microbiota, and their influence on host health and disease. *Molecular Nutrition & Food Research* 61 (1),
- Sangwan, N., Xia, F., Gilbert, J.A., 2016. Recovering complete and draft population genomes from metagenome datasets. *Microbiome* 4 (1), 8.
- Scher, J.U., Abramson, S.B., 2011. The microbiome and rheumatoid arthritis. *Nature Reviews Rheumatology* 7 (10), 569–578.
- Schloss, P.D., Westcott, S.L., Ryabin, T., *et al.*, 2009. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology* 75 (23), 7537–7541.
- Sender, R., Fuchs, S., Milo, R., 2016. Revised estimates for the number of human and bacteria cells in the body. *PLOS Biology* 14 (8), e1002533.
- Shetty, S.A., Marathe, N.P., Shouche, Y.S., 2013. Opportunities and challenges for gut microbiome studies in the Indian population. *Microbiome* 1 (1), 24.
- Shornikova, A.V., Casas, I.A., Isolauri, E., Mykkänen, H., Vesikari, T., 1997. Lactobacillus reuteri as a therapeutic agent in acute diarrhea in young children. *Journal of Pediatric Gastroenterology and Nutrition* 24 (4), 399–404.
- Shukla, S.D., Budden, K.F., Neal, R., Hansbro, P.M., 2017. Microbiome effects on immunity, health and disease in the lung. *Clinical & Translational Immunology* 6 (3), e133.
- Singh, J., Rivenson, A., Tomita, M., *et al.*, 1997. Bifidobacterium longum, a lactic acid-producing intestinal bacterium inhibits colon cancer and modulates the intermediate biomarkers of colon carcinogenesis. *Carcinogenesis* 18 (4), 833–841.
- Smits, L.P., Bouter, K.E., de Vos, W.M., Borody, T.J., Nieuwdorp, M., 2013. Therapeutic potential of fecal microbiota transplantation. *Gastroenterology* 145 (5), 946–953.
- Sobhani, I., Tap, J., Roudot-Thoraval, F., *et al.*, 2011. Microbial dysbiosis in colorectal cancer (CRC) patients. *PLOS ONE* 6 (1), e16393.
- Song, Y., Liu, C., Finegold, S.M., 2004. Real-time PCR quantitation of clostridia in feces of autistic children. *Applied and Environmental Microbiology* 70 (11), 6459–6465.
- Suau, A., Bonnet, R., Sutren, M., *et al.*, 1999. Direct analysis of genes encoding 16S rRNA from complex communities reveals many novel molecular species within the human gut. *Applied and Environmental Microbiology* 65 (11), 4799–4807.
- Sun, F., Xia, L.C., 2015. Accurate genome relative abundance estimation based on shotgun metagenomic reads. In: *Proceedings of the Encyclopedia of Metagenomics*, pp. 21–25. US: Springer
- Sun, J., Chang, E.B., 2014. Exploring gut microbes in human health and disease: Pushing the envelope. *Genes & Diseases* 1 (2), 132–139.

- Szajewska, H., Guarino, A., Hojsak, I., *et al.*, 2014. Use of probiotics for management of acute gastroenteritis: A position paper by the ESPGHAN working group for Probiotics and Prebiotics. *Journal of Pediatric Gastroenterology and Nutrition* 58 (4), 531–539.
- Teeling, H., Glöckner, F.O., 2012. Current opportunities and challenges in microbial metagenome analysis – A bioinformatic perspective. *Briefings in Bioinformatics* 13 (6), 728–742.
- Thomas, T., Gilbert, J., Meyer, F., 2012. Metagenomics—a guide from sampling to data analysis. *Microbial Informatics and Experimentation* 2 (1), 3.
- Totter, W., Denonfoux, J., Jaziri, F., *et al.*, 2013. The human gut chip “HuGChip”, an explorative phylogenetic microarray for determining gut microbiome diversity at family level. *PLOS ONE* 8 (5), e62544.
- Vernocchi, P., Del Chierico, F., Putignani, L., 2016. Gut microbiota profiling: Metabolomics based approach to unravel compounds affecting human health. *Frontiers in Microbiology* 7.
- Versalovic, J., Wilson, M., 2008. *Therapeutic Microbiology: Probiotics and Related Strategies*. ASM Press.
- Vollmers, J., Wiegand, S., Kaster, A.K., 2017. Comparing and evaluating metagenome assembly tools from a microbiologist's perspective—not only size matters!. *PLOS ONE* 12 (1), e0169662.
- Vrieze, A., Van Nood, E., Holleman, F., *et al.*, 2012. Transfer of intestinal microbiota from lean donors increases insulin sensitivity in individuals with metabolic syndrome. *Gastroenterology* 143 (4), 913–916.
- Vujkovic-Cvijin, I., Dunham, R.M., Iwai, S., *et al.*, 2013. Dysbiosis of the gut microbiota is associated with HIV disease progression and tryptophan catabolism. *Science Translational Medicine* 5 (193), 193ra91.
- Waldor, M.K., Tyson, G., Borenstein, E., *et al.*, 2015. Where next for microbiome research? *PLOS Biology* 13 (1), e1002050.
- Walker, A.W., Parkhill, J., 2013. Fighting obesity with bacteria. *Science* 341 (6150), 1069–1070.
- Walker, E., 2008. Benchmarking amazon EC2 for high-performance scientific computing. *USENIX Login* 33 (5), 18–23.
- Wang, Q., Garrity, G.M., Tiedje, J.M., Cole, J.R., 2007. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology* 73 (16), 5261–5267.
- Wang, T., Cai, G., Qiu, Y., *et al.*, 2012. Structural segregation of gut microbiota between colorectal cancer patients and healthy volunteers. *The ISME Journal* 6 (2), 320.
- Wang, W.L., Xu, S.Y., Ren, Z.G., *et al.*, 2015. Application of metagenomics in the human gut microbiome. *World Journal of Gastroenterology* 21 (3), 803.
- Weir, T.L., Manter, D.K., Shefflin, A.M., *et al.*, 2013. Stool microbiome and metabolome differences between colorectal cancer patients and healthy adults. *PLOS ONE* 8 (8), e70803.
- Woese, C.R., Fox, G.E., 1977. Phylogenetic structure of the prokaryotic domain: The primary kingdoms. *Proceedings of the National Academy of Sciences* 74 (11), 5088–5090.
- Wolf, B.W., Wheeler, K.B., Ataya, D.G., Garleb, K.A., 1998. Safety and tolerance of *Lactobacillus reuteri* supplementation to a population infected with the human immunodeficiency virus. *Food and Chemical Toxicology* 36 (12), 1085–1094.
- Wu, S., Rhee, K.J., Albesiano, E., *et al.*, 2009. A human colonic commensal promotes colon tumorigenesis via activation of T helper type 17 T cell responses. *Nature Medicine* 15 (9), 1016–1022.
- Wu, W.C., Zhao, W., Li, S., 2008. Small intestinal bacteria overgrowth decreases small intestinal motility in the NASH rats. *World Journal of Gastroenterology* 14 (2), 313.
- Zajac, V., Stevurkova, V., Matelova, L., Ujhazy, E., 2007. Detection of HIV-1 sequences in intestinal bacteria of HIV/AIDS patients. *Neuroendocrinology Letters* 28 (5), 591–595.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18 (5), 821–829.
- Zhang, J., Kobert, K., Flouri, T., Stamatakis, A., 2013. PEAR: A fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics* 30 (5), 614–620.
- Zhang, M.M., Cheng, J.Q., Lu, Y.R., *et al.*, 2010. Use of pre-, pro-and synbiotics in patients with acute pancreatitis: A meta-analysis. *World Journal of Gastroenterology* 16 (31), 3970.
- Zhang, Y.J., Li, S., Gan, R.Y., *et al.*, 2015. Impacts of gut bacteria on human health and diseases. *International Journal of Molecular Sciences* 16 (4), 7493–7519.
- Zhang, Q., Wu, Y., Fei, X., 2016. Effect of probiotics on glucose metabolism in patients with type 2 diabetes mellitus: a meta-analysis of randomized controlled trials. *Medicina* 52 (1), 28–34.
- Zhu, B., Wang, X., Li, L., 2010a. Human gut microbiome: The second genome of human body. *Protein & Cell* 1 (8), 718–725.
- Zhu, W., Lomsadze, A., Borodovsky, M., 2010b. Ab initio gene identification in metagenomic sequences. *Nucleic Acids Research* 38 (12), e132.
- Zoetendal, E.G., Collier, C.T., Koike, S., Mackie, R.I., Gaskins, H.R., 2004. Molecular ecological analysis of the gastrointestinal microbiota: A review. *The Journal of Nutrition* 134 (2), 465–472.

Further Reading

- de Groot, P.F., Frissen, M.N., de Clercq, N.C., Nieuwdorp, M., 2017. Fecal microbiota transplantation in metabolic syndrome: History, present and future. *Gut Microbes*. 1–15.
- Gonzalez, C.G., Zhang, L., Elias, J.E., 2017. From mystery to mechanism: Can proteomics build systems-level understanding of our gut microbes?.
- Ley, R.E., Hamady, M., Lozupone, C., *et al.*, 2008. Evolution of mammals and their gut microbes. *Science* 320 (5883), 1647–1651.
- Li, M., Wang, B., Zhang, M., *et al.*, 2008. Symbiotic gut microbes modulate human metabolic phenotypes. *Proceedings of the National Academy of Sciences* 105 (6), 2117–2122.
- Naggal, R., Yadav, H., Marotta, F., 2014. Gut microbiota: The next-gen frontier in preventive and therapeutic medicine? *Frontiers in Medicine* 1, 15.
- O'Sullivan, D.J., 2000. Methods for analysis of the intestinal microflora. *Current Issues in Intestinal Microbiology* 1 (2), 39–50.
- Roumpeka, D.D., Wallace, R.J., Escalettes, F., Fotheringham, I., Watson, M., 2017. A review of bioinformatics tools for bio-prospecting from metagenomic sequence data. *Frontiers in Genetics* 8, 23.
- Santamaria, M., Fosso, B., Consiglio, A., *et al.*, 2012. Reference databases for taxonomic assignment in metagenomics. *Briefings in Bioinformatics* 13 (6), 682–695.
- Tuohy, K., Del Rio, D., 2014. *Diet-Microbe Interactions in the Gut: Effects on Human Health and Disease*. Academic Press.
- Wu, H.J., Wu, E., 2012. The role of gut microbiota in immune homeostasis and autoimmunity. *Gut Microbes* 3 (1), 4–14.
- Yatsunenkov, T., Rey, F.E., Manary, M.J., *et al.*, 2012. Human gut microbiome viewed across age and geography. *Nature* 486 (7402), 222–227.
- Zoetendal, E.G., Rajilic-Stojanovic, M., De Vos, W.M., 2008. High-throughput diversity and functionality analysis of the gastrointestinal tract microbiota. *Gut* 57 (11), 1605–1615.

Relevant Websites

<http://www.bioinformatics.babraham.ac.uk/projects/fastqc>
 Babraham Bioinformatics.
<http://DECIPHER.cce.wisc.edu>
 DECIPHER.

http://hannonlab.cshl.edu/fastx_toolkit/index.html
FASTX-Toolkit.
<http://www.metahit.eu/>
MetaHIT.
<http://www.mynewgut.eu>
MyNewGut.
<http://hmpdacc.org/>
NIH Human Microbiome Project.
http://sourceforge.net/project/showfiles.php?group_id=262346
SOURCEFORGE.

Biographical Sketch



Toral Manvar is a registered doctorate student at Gujarat University, Ahmedabad, India. She has more than 7 years of experience in CRO research and bioinformatics. She is working as team leader in NGS bioinformatics division at Xcelris Labs Ltd., Gujarat, India. She has completed her Master's in Bioinformatics in 2009 from Sardar Patel University, V.V Nagar, Gujarat, India. She has special interest in Next Generation Sequencing data analysis involving different DNA and RNA based applications and particularly metagenomics. She has experience in teaching and is currently involved in mentoring and tutoring bioinformatics to graduate and undergraduate students.



Vijay Lakhujani is a certified bioinformatician (Dept. of Biotechnology, Govt. Of India) with deep interest in Genomics and next generation sequencing technologies. He has 6.5+ years of industry experience working on design, implementation and analysis of microarray data and high-throughput NGS data. He is working as Project Scientist in NGS bioinformatics division at Xcelris Labs Ltd., Gujarat, India. He has completed his Masters in Bioinformatics in 2011 from University Of Pune, Maharashtra, India. He has special interest in computational process automation, programming and visualization. He has been formally certified as "lecturer" (CSIR-NET-JRF + LS, Govt. of India) and is involved in providing bioinformatics knowledge sharing sessions and bioinformatics trainings to graduate and undergraduate students.

Functional Enrichment Analysis

Srikanth Manda, Institute of Bioinformatics, Bangalore, India

Daliah Michael, Indian Institute of Science, Bangalore, India

Sudhir Jadhao, Institute of Health and Biomedical Innovation, School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD, Australia

Shivashankar H Nagaraj, Institute of Health and Biomedical Innovation, School of Biomedical Sciences, Queensland University of Technology, Brisbane and Australia Translational Research Institute, Woolloongabba, QLD, Australia

© 2019 Elsevier Inc. All rights reserved.

Abbreviations

BH	Benjamin Hochberg	GSEA	Gene set enrichment analysis
BLAST	Basic local alignment search tool	JSP	Java Server Pages
CRAN	Comprehensive R Archive Network	MEA	Modular enrichment analysis
DAVID	Database for Annotation, Visualization and Integration discovery	NCBI	National Center for Biotechnology Information
FDR	False discovery rate	NGS	Next-generation sequencing
GO	Gene ontology	SEA	Singular enrichment analysis
		SQL	Server Query Language

Introduction

Biological systems are complex entities with several levels of complexity inter-linked and integrated to perform diverse tasks which are monitored and coordinated closely by complex regulatory systems. The organismal level dynamics are result of complex interactions between network of genes, proteins and other biological macro and micro-molecules. These interactions across different omic levels regulate the organism's form and functions and any rewiring leads to disruption of the entire system leading to different disorders depending upon the nature of change. A comprehensive understanding of the different regulatory systems including genomics, proteomics, and metabolomics coupled with the integration of this information into a unified model will lead to better understanding of the resultant phenome at an organismal level. Hence, it is increasingly evident that a systems approach is needed to gain a deeper insight into any biological state of interest. A holistic systems approach allows us to integrate the information collected from different omic studies in order to visualize and understand the complete landscape at the organismal level (Rhee *et al.*, 2008).

Systems level probing is now possible owing to the tremendous technological advances in high throughput experimentation. Studies yielding complex proteomic and genomic datasets help us gain insight into the dynamic changes in the landscape including differential expression and activation of specific pathways depending on the biological situation. The cost of these high throughput omic or multi-omic studies has also fallen dramatically over recent years, making them increasingly affordable and sensitive. This has resulted in a huge deluge of data generated from these high throughput omic studies being performed by investigators who are increasingly motivated to gain systems-level insights into their experimental systems. While generating omic data has become far easier, the difficulty in interpreting the results of such experiments has risen dramatically with the size of datasets, at times proving to be overwhelming for unexperienced researchers.

The analysis of any high throughput experiment ultimately converges upon the identification of all any significantly dysregulated molecules (e.g., genes, proteins, etc.) which are speculated to serve as possible leads that may shed light on the cellular dynamics in the particular biological context of experimental interest. A common analytical approach often hinges on studying differentially expressed genes to identify biomarkers or unique signatures that characterize a specific phenotype. Understanding how individual biomolecules influence a specific phenotype can be invaluable in the advancement of scientific understanding, but understanding the complex interactions and cumulative effects of all biomolecules present within a given system can be just as important despite being a more difficult analytical task. Manual annotation and examination of the roles of each of the dysregulated molecules along with interpretation of their collective roles in a given system is very tedious and time-consuming. It requires thorough study of both extant literature and available databases in order to gather information regarding the functional roles of all the molecules of interest (Hill *et al.*, 2010; Hoehndorf *et al.*, 2015).

Establishing the functional roles of each of the molecules of interest in a particular experimental context is essential in order to draw any conclusions or inferences from the data. When attempting to identify molecules of interest in a complex high throughput dataset, a functional enrichment analysis is the most commonly used approach for analysis and interpretation of biological. Functional enrichment analysis is possible due to the development of unified biological ontologies by the Gene Ontology (GO) Consortium which remains same across eukaryotes. The aim of the Gene Ontology Consortium was to develop, maintain and update well structured, precisely defined terms – also known as ontologies – that articulate the roles of biological molecules across different species (Gene Ontology Consortium, 2015). The GO Consortium is discussed in detail later in the article.

Functional enrichment analysis using the GO database was rendered possible by the emergence of bioinformatics tools that can retrieve information from the GO database. With the advent of these tools it became possible to summarize the information regarding all biological activities in which a particular molecule is known to participate. This information can then be condensed into a succinct

enrichment summary. This analysis involves the use of a statistical approach to investigate abundantly represented classes of biological macromolecules such as genes, proteins or metabolites actively involved in different biological processes, molecular functions and cellular compartments. This approach allows us to generate a more holistic view of an experimental system, letting us visualize the biological roles of the differentially expressed biological macromolecules and deduce the significantly enriched biological themes. The different enrichment techniques used by different tools to assess enrichment are described below.

Types of Enrichment

There are three algorithms for performing enrichment analysis: Singular enrichment analysis (SEA), Gene set enrichment analysis (GSEA), Modular enrichment analysis (MEA) (Tipney and Hunter, 2010).

SEA

SEA is a relatively simple, easy-to-use approach to enrichment that is the most traditionally utilized. An SEA analysis takes a list of genes with particular values (e.g., Fold-change values) as the inputs and iteratively procures the associated annotation terms from the GO database for each of these genes. This approach estimates a p-value for each of the retrieved annotation terms by comparing the observed frequency of a term with the frequency expected by random chance. All the terms beyond a threshold (usually p-value ≤ 0.05) are considered to be significantly enriched. Statistical tests that are used to facilitate the identification of significantly enriched GO terms include the Chi-square test, Fischer's Exact Test, Binomial Probability or Hypergeometric distribution. A more detailed discussion of the different statistical methods used for enrichment analyses can be found in the later sections of this article. Once compiled, each of the significantly enriched annotation terms associated with each of the genes in the input gene list are then listed in form of a table. One caveat to this approach is that it ignores the hierarchical relationships between the annotation terms, and it can thus lead to redundancy within the enriched terms owing to association of multiple terms with a single biological process. Some of the tools that use this approach are: Onto-Express, FuncAssociate 2.0 and BiNGO (Berriz *et al.*, 2009; Draghici *et al.*, 2003; Maere *et al.*, 2005).

GSEA

The GSEA approach is best-suited to a comparison-based analysis between two classes of data, multiple scenarios of data points or a single dataset across a series of timepoints. This approach requires a biological metric (e.g., Fold-change or expression) for each biomolecule of interest in order to rank them. A maximum enrichment score (MES) for all genes in a particular category is generated as an output of the GSEA analysis. Once MES scores are compiled, a p-valuebased assessment is done by comparing the ranked MES to the randomly generated MES distributions to assess for enrichment for particular terms beyond what would be expected due to chance. Unlike SEA, GSEA analyses determine whether the ranked list of genes share a particular annotation and form gene sets, reducing the redundancy that can arise from similar GO terms in SEA analyses.

Tools such as GSEA/P-GSEA, GeneTrail use this approach for enrichment analysis (Keller *et al.*, 2008; Mootha *et al.*, 2003; Stöckel *et al.*, 2016; Subramanian *et al.*, 2005).

Modular Enrichment Analysis (MEA)

The MEA approach considers the relationships between the different annotation terms during enrichment. Thus, this approach has better sensitivity and specificity owing to the reduction in the redundancy inherent in SEA/GSEA analyses. This modular approach allows the analysis software to deduce the biological theme in a much more refined manner as it considers the relationship between the terms, rather than just processing individual terms in the two above mentioned processes. Tools such as Ontologizer and GeneCodis use this approach (Bauer *et al.*, 2008; Carmona-Saez *et al.*, 2007).

Steps for Enrichment Analysis

Functional enrichment analysis can be performed using a range of tools that make use of different enrichment strategies. While the details of each protocol differ, these tools share common basic principles that are outlined below.

The starting point of any enrichment analysis is the generation of raw input data to be analyzed in the form of a list of biomolecules. Depending on the nature of study, enrichment analysis can be performed using lists of genes, proteins or metabolites. The list of molecules used can be ranked or unranked depending upon the enrichment algorithm employed by the tool. The tool to be used also should be selected carefully on the basis of the expected results from the analysis (Yang *et al.*, 2008).

Step 1: Calculation of Enrichment Score

The selection of genes to be used for downstream analysis using any tool and algorithm needs to be carefully compiled, as the entire functional analysis will be based on this list. The choice of using a ranked or unranked list depends upon the desired outcome expected

and the tool being used for analysis. To ensure that meaningful and specific results can be obtained, a background dataset should be included in the analyses. This background dataset is the list of all the molecules identified in the study, including the significantly dysregulated or important as well as the identified molecules of interest. The background database serves to improve the accuracy of the enrichment score and nullify any biases introduced in capturing the molecules due to experimental procedures.

Once this gene list has been generated and provided to the analysis software, ontology terms for each molecule are fetched from the GO database. These terms encompass a range of molecule descriptors including their functions, the processes/pathways in which they participate or the cellular compartments where they are localized. Ontology terms for the query and background lists are compared to assess the enrichment of different terms and processes. The enriched processes/pathways are those that are overrepresented in the query dataset relative to the background dataset. The enrichment score is a quantified value for each of the ontology term found to be associated by the query molecules.

Step 2: Estimation of Enrichment Score Significance

The statistical significance of the enrichment score can be calculated by using Binomial, Fischer's exact, Hypergeometric or Chi-square tests. The Binomial probability is used when the list of background molecules is large. The Fischer exact, Chi-square and Hypergeometric distributions are better suited to analyses with smaller background datasets. The statistical significance of the enrichment score is quantified with the help of a p-value denoting the extent to which the occurrence of the enrichment of a term is likely to be due to random chance. All terms below a chosen p-value threshold, typically 0.05 (representing a 5% chance of a given enrichment being due to chance), are considered to be significantly enriched. For a more stringent analysis, the p-value threshold can be further decreased to 0.01 or lesser. Many tools also implement alternative mathematical approaches. The user should decide the best suitable statistical method for the dataset.

Step 3: Adjustment for Multiple Hypothesis Testing

As multiple comparisons are being made simultaneously, adjusting p-values to account for false discovery rate (FDR) is necessary in enrichment analyses. FDR is used to control the family-wise error rate, and, an FDR threshold of 1% allows only 1% of the values to be observed by chance when drawn according to the null hypothesis. To account for this FDR, multiple hypothesis correction is applied to the results of the above statistical analysis. The majority of functional enrichment analysis tools perform such corrections using Bonferroni, Benjamini-Hochberg and also Holm, Q-value, and Permutation correction algorithms (de Leeuw *et al.*, 2016).

Gene Ontology Consortium

The Gene Ontology Consortium ("see Relevant Websites section") was developed to provide a comprehensive resource for functional enrichment analysis. It is a collaborative effort that annotates biological macromolecules on the basis of experimental evidence to describe their different biological roles.

GO annotations describes the molecular functions contributing to different biological processes and cellular compartments where these occur. It also includes references to the experimental evidence supporting these associations. The GO consortium has painstakingly addressed the constant need for the development, maintenance and improvement of consistent annotations across species. This group has successfully compiled functional information from over 460,000 species including plants, animals and microorganisms (Gene Ontology Consortium, 2015).

Several commercial and open-source tools have been developed to facilitate quicker functional enrichment analyses, allowing researchers to probe the complexity of their large datasets. In the following sections we describe several tools which can be used for such analysis. Table 1 provides the list of widely used tools for functional enrichment.

DAVID

The Database for Annotation, Visualization and Integration Discovery (DAVID) was released in 2003 with the aim of providing users with a simple user interface for quick biological interpretation of large datasets. The DAVID knowledgebase integrates a broad range of publically available annotation resources in a centralized location, thus enhancing the quality of high-throughput functional annotation analysis. The entire DAVID is freely available for download in simple text format ("see Relevant Websites section") and is updated frequently. It also provides a number of powerful tools which researchers can use to comprehensively annotate large gene sets from different biological angles. The DAVID Bioinformatics Resources is available for public access at website provided in "Relevant Websites section" (Huang *et al.*, 2007, 2008, 2009).

AmiGO

AmiGO ("see Relevant Websites section") is an open source web-based application developed and maintained by the GO Consortium to query, browse and visualize ontologies and related gene product annotation (association) data. It features a BLAST search, Term Enrichment and GO Slimmer tools and the GO Online SQL Environment. It displays the distribution of annotations

Table 1 List of widely used Bioinformatics tools in the enrichment analysis. Each tool has their distinct algorithm and features; the general procedure of the tools is take query gene list processed it using algorithm and statistics against their annotation database to generate resulting enriched gene set

<i>Tools</i>	<i>Year of publication</i>	<i>Citation</i>	<i>Statistical Method</i>	<i>Tool output</i>
DAVID	2003	6078	Fisher's exact test	P-value with Enrichment score for each cluster in HTML
BINGO	2005	2475	Hypergeometric test/ binomial	Network of enriched biological process
GOrilla	2009	1359	Hypergeometric test	Colored hierarchical structure of enriched biological process
Blast2GO	2008	997	Fisher's exact test	Annotation in CSV file
WebGestalt	2017	833	Fisher's exact test	Network of enriched biological process
ToppGene Suite	2009	816	Hypergeometric test with Bonferroni correction	P-value with Enriched category
GENECODIS	2007	439	Hypergeometric test/ χ^2 test of independence	P-value with Enriched category
Ontologizer	2008	396	Fisher's exact test/ the novel parent-child method	Colored hierarchical structure of enriched biological process
Enrichr	2016	273	Fuzzy Enrichment analysis	Enriched category with cluster grams
MetPA	2010	220	Centrality measure	Network of chemical compounds with metabolites
DIANA-miRPath	2015	212	Fisher's exact test	Annotation in CSV file
FunRich	2015	153	Hypergeometric test/ Bonferroni	Enriched biological process with Venn diagrams, column, bar, pie and doughnut charts as well as interaction network and heatmap images.
INRICH	2012	148	Permutation approach	P-value with Enriched category
EasyGO	2007	133	Hypergeometric Test	Colored hierarchical structure of enriched biological process
GO-Elite	2012	127	Hypergeometric test/ Fisher's exact test	Network of enriched biological process
KEA	2009	97	Fisher's exact test	Network of enriched biological process
GeneTrailExpress	2008	91	Hypergeometric test/Fisher's exact test	Network of enriched biological process
HTSanalyzeR	2011	53	Hypergeometric test	Enrichment map and Network of enriched biological process

associated with a term and its children in both graphical and tabular formats via a bar chart viewer. It is a server-based Perl application that retrieves information from a database using go-perl and go-db-perl. It is publicly available at website provided in "Relevant Websites section" (Carbon *et al.*, 2009).

GOrilla

GOrilla is a web-based application developed using Java. It is publicly available at: "See Relevant Websites Section". It currently supports 8 organisms: *Homo sapiens*, *Mus musculus*, *Rattus norvegicus* and other model organisms such as *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster* and *Arabidopsis thaliana*. Functional analysis can be performed using this tool with either of two approaches. The first approach involves using a ranked list of genes enriched via the mHG statistics employed by GOrilla. The second approach involves the generally followed comparison of target and customized background dataset which is analyzed using a hypergeometric model employed by the tool. Thus, the input for GOrilla can be either a single ranked list of genes or two sets of genes (target and background). The output consists of color-coded trimmed Directed acyclic graph (DAG) of all annotated GO terms generated with the help of GraphViz tool. The color codes denote the level of significance of enriched GO term and the nodes are linked to the corresponding entry in a table. This table includes the enriched GO terms, p-values for enrichment and associated annotated genes. The results can be exported in excel format (Eden *et al.*, 2009).

FunRich

FunRich is a user-friendly, Windows-based, standalone application allowing users to perform enrichment analysis on their desktops. It is publicly available at: "See Relevant Websites Section" It allows users to perform functional enrichment analyses and interaction

network analyses on background databases without any restrictions on genomic, proteomic or metabolomics data. Users are given an option to select the database. It provides three database options: the FunRich database containing only human annotations, the UniProt database equipped with support for 20 different taxonomic levels and a Custom database that can be customized according to the user's requirements. This tool was developed in C# language using Microsoft .NET library. A hypergeometric distribution test is used to evaluate the statistical significance of enriched and depleted terms. False Discovery Rate (FDR) is implemented to correct for multiple testing by Bonferroni and Benjamini-Hochberg (BH) method. It provides a graphical representation of the data in the form of scalable Venn diagrams, column graphs, heatmap, bar graphs, pie chart and doughnut charts with customizable font, scale and color. Results can be saved in publication quality images and excel sheets (Pathan *et al.*, 2015).

GOplot

R is a commonly used platform by the community for data analysis and for the generation of high quality graphical representations of data. GOplot imports several R packages and is based on the graphical package, ggplot2, enabling users to generate a variety of multilayered graphs. This approach works with two types of input: Molecule lists with their expression levels, and results of functional enrichment analyses. It generates outputs offering different levels of detail: Bar plots or Bubble plots can be constructed to visualize the most enriched categories and Circle, Chord or Cluster plots can offer a more detailed representation of the analyzed results. The Circle plot has two layers: The inner layer is a bar plot representing the statistical significance of enriched terms, and the Outer layer displays the expression levels of particular molecules. The Chord plot has GO terms and molecules connected with ribbons wherever applicable. The Cluster plot displays a circular dendrogram clustered on the basis of their expression profile along with another layer of the associated GO terms. The R package GOplot is available via CRAN-The Comprehensive R Archive Network provided in "Relevant Websites section".

BiNGO

Biological Networks Gene Ontology tool (BiNGO) is a flexible, extendable, open-source Java tool to determine significantly overrepresented GO terms in a set of genes or a subgraph of a given biological network. It maps the functional annotations of gene sets on a GO hierarchy to produce a customizable visual representation in Cytoscape. It provides annotations for a wide range of organisms and these annotations are parsed from GO information available at NCBI. It provides two statistical tests for evaluating the significance of the associated GO terms: hypergeometric and binomial tests. Both of these tests generate p-values as a score for the statistical significance of the enrichment of the GO term. Multiple testing corrections are then applied to discard Type I errors. Multiple testing can be done by applying Bonferroni correction or FDR by BH correction. BiNGO results can be visualized in Cytoscape or can be saved in a tab-delimited text file. It is distributed under GNU General Public License and can be downloaded as a Cytoscape plugin (Maere *et al.*, 2005).

Blast2GO

Blast2GO ("see Relevant Websites section") is a biologist-oriented, high throughput, quality data-mining tool which can be used for functional annotation of a wide range of organisms. It is an operating system independent Java Application made available in Java Web Start (JWS).

GO annotation in Blast2GO proceeds through three basic steps: search for homologues by BLAST, GO term mapping and actual annotation. Blast2GO uses the Basic Local Alignment Search Tool (BLAST) to find sequences similar to your query set. A local BLAST or Web-based BLAST can be performed with parameters adjusted according to user's requirement. The BLAST results can be saved in a file in different formats.

Mapping is done to retrieve the GO annotations for high scoring BLAST hits. Resultant accessions retrieved after BLAST are then used to retrieve gene names, symbols and other IDs. These are then searched against the GO database.

Annotation is done to select the GO terms from the GO pool obtained by the mapping step and assigning them to the query sequences. GO annotation is carried out by applying an annotation rule (AR) to the found ontology terms. This rule seeks to find the most specific annotations with a particular selected level of reliability. For each candidate GO an annotation score (AS) is computed. The annotation rule provides a general framework for annotation but depends completely on how the different parameters at the AS are set. The functionality of InterPro annotations in Blast2GO allows retrieval domain/motif information in a sequence-wise manner. Corresponding GO terms are then transferred to the sequences and merged with already compiled GO terms. Blast2GO provides EC annotation through the direct mapping available at the GO website. Additionally, the KEGG map module allows for the display of enzymatic functions in the context of the metabolic pathways in which they participate. Visualization of the Blast2GO results can be constructed with the help of interactive directed acyclic graphs (DAG), pie charts and bar graphs which can be customized and filtered.

Most of its functionalities are available for free access at website provided in "Relevant Websites section" (Gotz *et al.*, 2008).

Case Study

Having seen the importance and application of functional enrichment analysis, the challenge now lies in the selection of the right tool for a particular analysis. There are plethora of tools that have been developed in the past to perform enrichment analyses, and

many more are emerging at a considerable rate. While most of the tools employ the same general approach, they differ in the way they represent the output and the interpretation of this enrichment dataset (Khatri and Draghici, 2005). Each of these tools has their own advantages and limitations. Thus, at the time of use, one should also have a clear understanding of the disparity between tools and choose the most appropriate tool that would best serve the user's research requirements. There are two excellent review articles (Khatri and Draghici, 2005; Huang *et al.*, 2009) on these tools, that should be consulted when selecting the analytical approach to employ for a particular dataset.

One of the most popular tools used worldwide is DAVID (Huang *et al.*, 2007, 2009), discussed above. For the convenience of the readers we present here a demonstration of the tool using publically available sample input data (i.e., the set of genes) derived from an earlier study (Gardina *et al.*, 2006), primarily to exemplify the use of the tool. DAVID can be accessed at the URL provided in "Relevant Websites section".

The study by Gardina *et al.*, identified 160 differentially expressed genes in 20 pairs of tumor and normal colon cancer samples. This case study uses this set of 160 differentially expressed genes to demonstrate functional enrichment analysis using DAVID (Huang *et al.*, 2007, 2009).

The 'Home' page of DAVID (Huang *et al.*, 2007, 2009) has two main options – 'Functional Annotation' and 'Gene Functional Classification'. The steps to follow are outlined below:

- (1) To begin, click on the 'Functional Annotation' option. A page opens up which will prompt the user to either paste a list of genes or upload a file containing the list of genes. The 'Select Identifier' asks the users to choose the appropriate type of gene ID that the input gene list uses. In the 'List Type', the user has to select whether the input gene list is a gene list or is to be used as the background gene set. Then click on 'Submit' (Fig. 1). The default database set as the background is the *Homo sapiens* genome database. However, the users may select different databases, according to the nature of their search and study. Choosing and setting up the right background for the analysis is a very crucial step. The general rule of thumb is to set a large background database such as the whole genome, as the results obtained will have more significant p-values as opposed to those resulting from concise background datasets (Huang *et al.*, 2009).
- (2) This analysis will produce an 'Annotation Summary Results' page that displays categories indicates that the genes have been annotated in Fig. 2. Users may choose to see additional results there.
- (3) Upon clicking the 'Functional Annotation Chart', the user can view a report that displays the annotation terms and the annotation database where the associated genes were mapped. It also displays the number and percentage of genes involved out of the total genes in the input list and the related terms list. The p-value of each result is displayed: the smaller the p-value the more enriched (Fig. 3a) a term is. Clicking on each annotation result will provide a list of genes from the input gene list that are associated with that particular annotation term (Fig. 3b).
- (4) The 'Functional Annotation Table' provides, for each input gene, the annotation terms, the database in which it is oriented, related genes and the species of interest (Fig. 4).
- (5) The annotation of the genes based on the Enrichment score is provided upon clicking the 'Functional Annotation Clustering' option. Here, the genes are clustered under an enrichment score, with a higher enrichment being the one with the higher

Functional Annotation Tool
DAVID Bioinformatics Resources 6.8, NCBI/NIH

Home Start Analysis Shortcut to DAVID Tools Technical Center Downloads & APIs Terms of Service Why DAVID? About Us

*** Welcome to DAVID 6.8 ***
*** If you are looking for DAVID 6.7, please visit our development site. ***

Functional Annotation Tool

Submit your gene list to start the tool!

Tell us how you like the tool
Read technical notes of the tool
Contact us for questions

Key Concepts:

Terms/Genes Co-Occurrence Reliability
Finding functional categories based on co-occurrence with sets of genes in a gene list can rapidly aid in unraveling new biological processes associated with cellular functions and pathways. DAVID 6.8 allows investigators to sort gene categories from diverse annotation systems. Sorting can be based either the number of genes within each category or by the EASE score. [More](#)

Gene Similarity Search
Any given gene is associating with a set of annotation terms. If genes share similar set of those terms, they are most likely involved in similar biological mechanisms. The algorithm tries to group those related genes based on the agreement of sharing similar annotation terms by kappa statistics. [More](#)

Term Similarity Search
Typically, a biological process/term is done by a cooperation of a set of genes. If two or more biological processes are done by similar set of genes, the processes might be related in the biological network. This search function is to identify the related biological processes/terms by quantitatively measuring the degree of the agreement how terms share the similar participating genes. [More](#)

Integrated Solutions

Protein Annotation	Numerous public sources of protein and gene annotation have been parsed and integrated into DAVID 6.8. DAVID 6.8 contains information on over 1.5 million genes from more than 65,000 species. A list of protein or gene identifiers can be uploaded at all once to extract and summarize functional annotation associated with group of genes or with each individual gene. Data can be displayed in chart or table format or downloaded to the user's hard drive.
Nucleotide Data Sources	
Co-occurrence Probability	
Gene Homology Annotation	
Dynamic Pathway Maps	
Disease Annotations	

Upload Gene List

Demolish 1 Demolish 2
Upload File

Step 1: Enter Gene List

A: Paste a List

FXSD00000240479
FXSD00000240484
FXSD00000240528
FXSD00000240461
FXSD00000240481

Clear

Or

B: Choose From a File

Browse... Download...

Multi List File

Step 2: Select Identifier

ENSEMBL_GENE_ID

Step 3: List Type

Gene List Background

Submit List

Either paste a list of gene IDs or Upload a file containing the list of gene IDs

Choose the appropriate Gene ID type

Select the gene IDs list as a Gene List or as the Background List

Fig. 1 Input page.

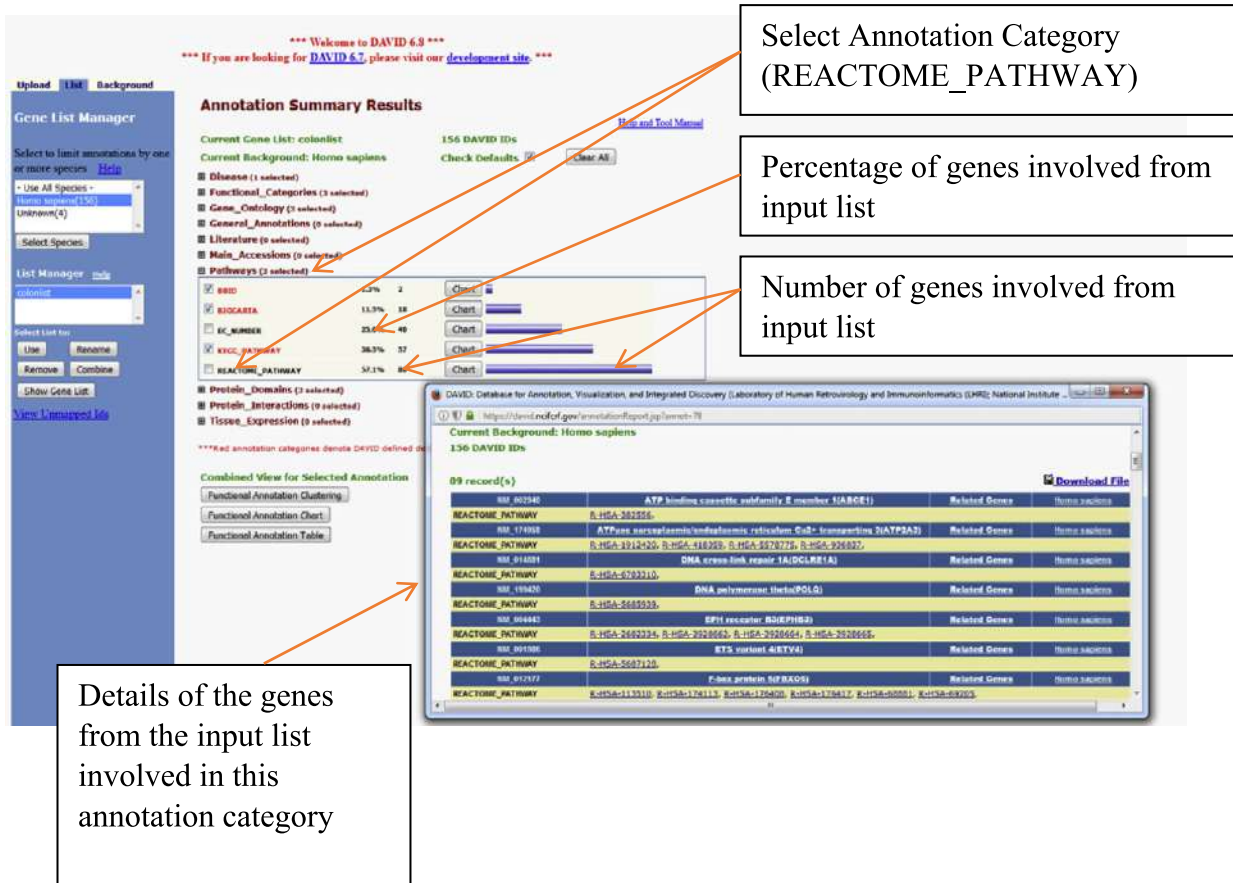


Fig. 2 The Annotation Summary page provides results on various functional annotation categories that the input genes have mapped with. Here, the 'Pathways' category shows the results of all the genes involved in the mentioned pathways, from the input gene list (i.e., 89 genes out of 156 genes).

value. The database associated as well as the number of genes involved, related terms and the p-values are also displayed. Clicking on each annotation result will provide a list of genes from the input gene list that fall under that particular cluster, with the particular enrichment score and p-value (Fig. 5).

- (6) Note: Although DAVID (Huang *et al.*, 2007, 2009) accepts a very large input gene list, the Functional Annotation Clustering option requires the users to input a maximum of 3000 genes only. Thus, users should split their input gene list into separate files, such that each file contains no more than 3000 genes.
- (7) Each result page provides an option to download the results in an easily documented manner.
- (8) In some cases, the user may not be aware of the gene ID type or may use a type that is not listed out in the Gene Identifier. DAVID has an in-built tool called the Gene ID Conversion Tool (Huang *et al.*, 2007, 2009), which enables the conversion of each gene entry into an acceptable Gene ID type.

Results

In the 'Functional Annotation Chart' (Figs. 3a,b), on selecting the first annotation result with the lowest p-value (i.e., $8.9E-8$), the list of the most enriched genes is displayed. In addition, from the 'Functional Annotation Clustering' result, on selecting the first result with enrichment score 3.12, a list of highly enriched genes is displayed. As an example, let us take the first gene which appears in this cluster, NM_014750. This represents DLG (Discs Large Homolog)-associated protein 5, which is a hepatoma up-regulated protein known to be a cell cycle regulator that plays a significant role in carcinogenesis. Similarly, users can gain invaluable volumes of information about genes and their functions through the various links provided by DAVID's results.

Future Directions

Users should test some of these and other tools in parallel or in an integrated mode in order to get relevant biological interpretations that best encapsulate their datasets. The databases described in this text are not comprehensive or exhaustive, and

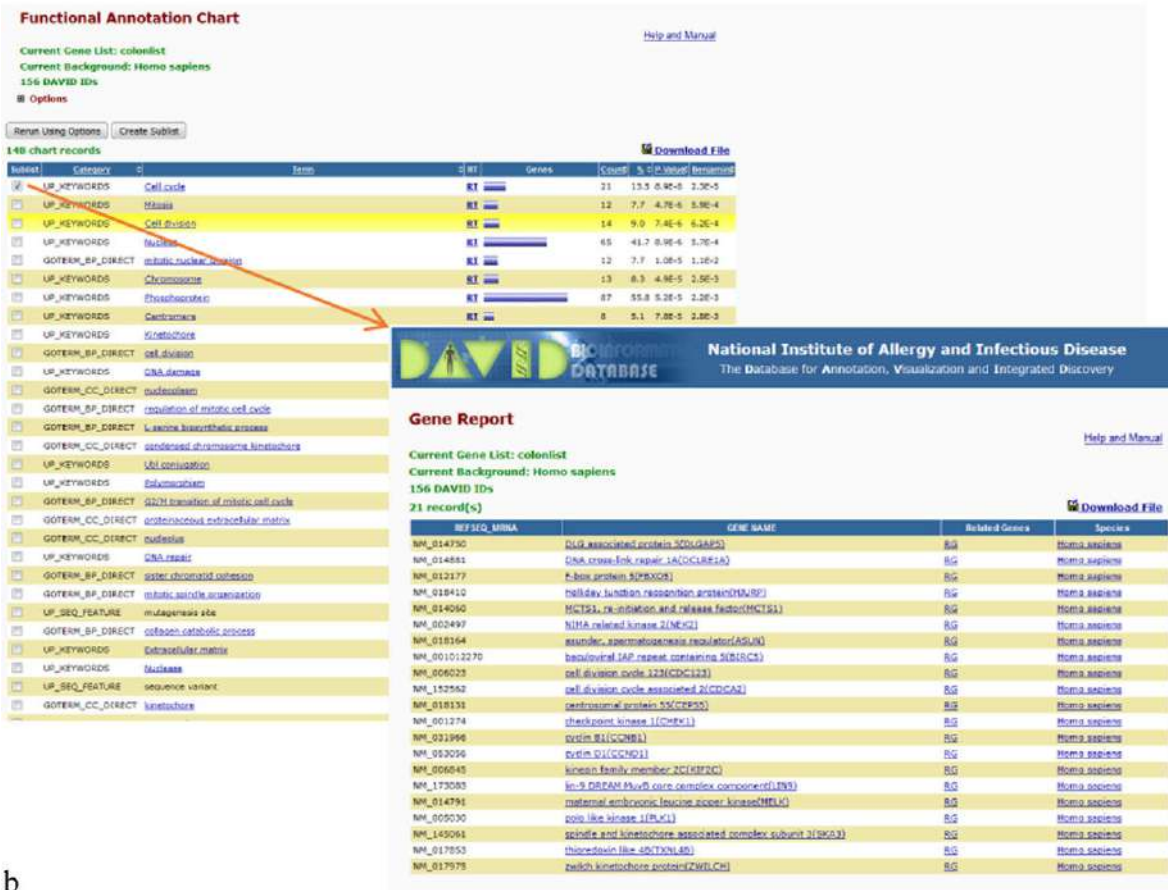
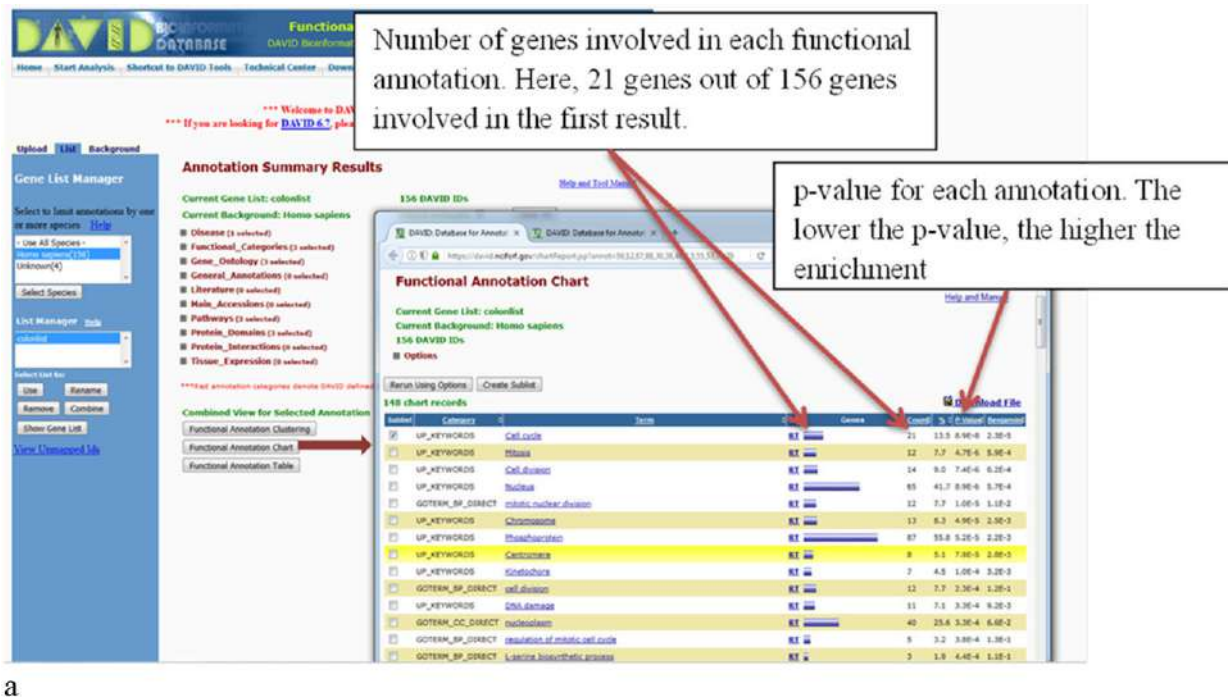


Fig. 3 (a). Functional Annotation Chart. (b). Gene Report generated from the first result on the Functional Annotation Chart page. The list of gene IDs for the particular functional annotation "Cell Cycle" along with the Gene names and related genes is displayed here.

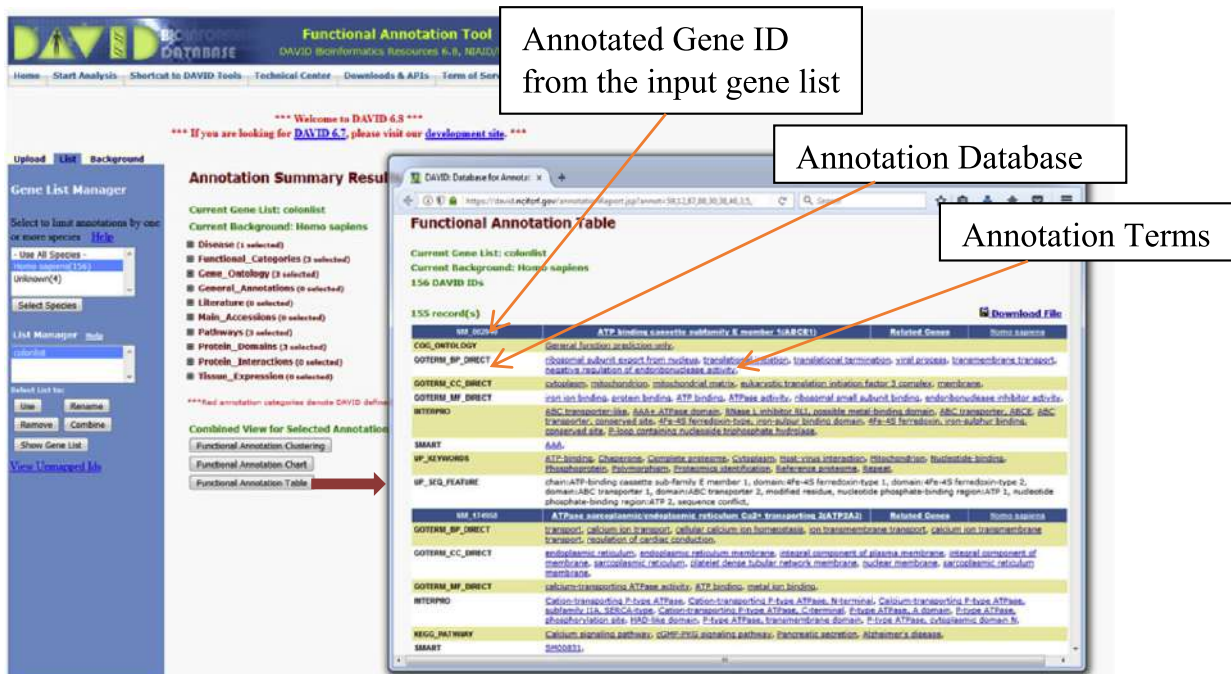


Fig. 4 The Functional Annotation Table page. Here, for each annotated Gene ID from the input gene list, details of the respective annotation database, functions/annotation terms, related genes and species are given.

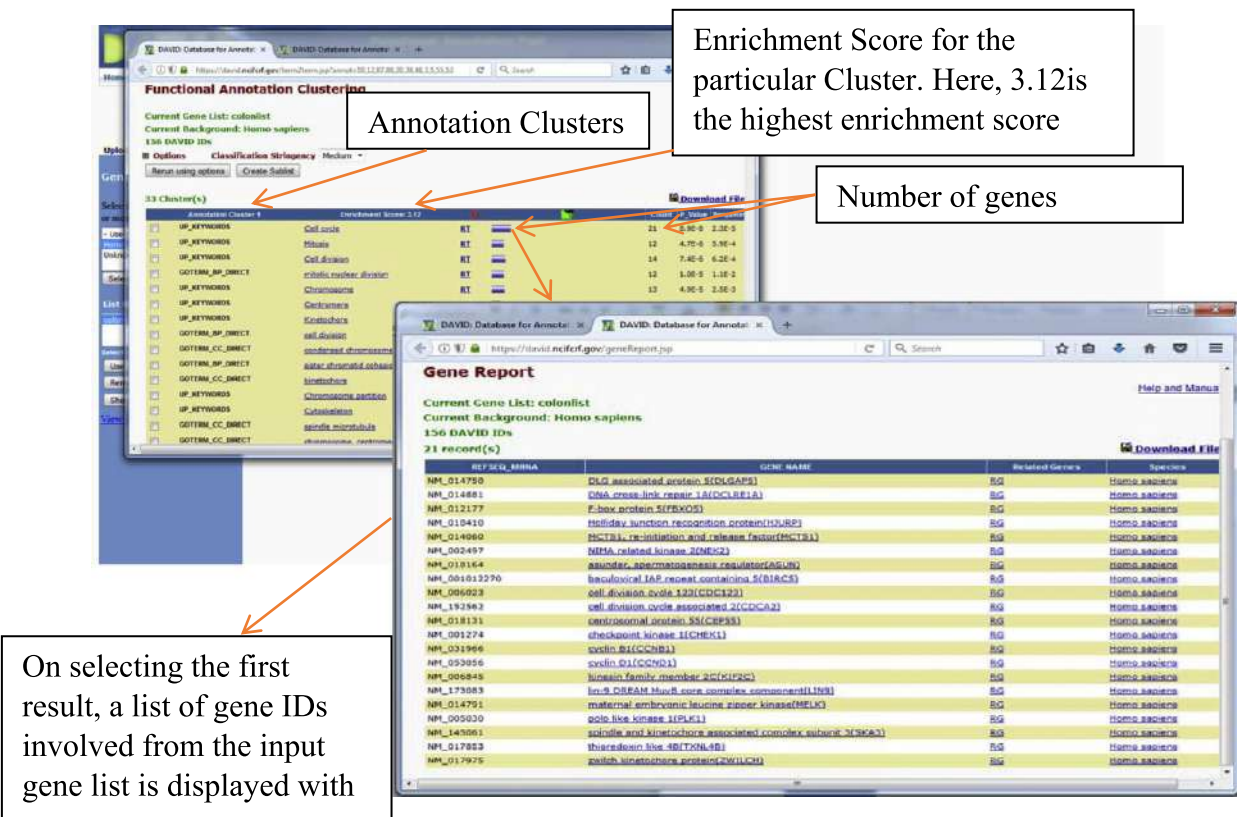


Fig. 5 The Functional Annotation Clustering page. On selecting the first result, a list of gene IDs involved from the input gene list is displayed with all details.

Table 2 Link of Enrichment analysis tools

<i>Tools</i>	<i>Link</i>
DAVID	https://david.ncifcrf.gov/
GOrilla	http://cbl-gorilla.cs.technion.ac.il/
FunRich	http://www.funrich.org/
BiNGO	https://www.psb.ugent.be/cbd/papers/BiNGO/Home.html
Blast2GO	https://www.blast2go.com/
WebGestalt	http://www.webgestalt.org/option.php
EasyGO	http://bioinformatics.cau.edu.cn/easygo/
ToppGene Suite	https://toppgene.cchmc.org/
Enrichr	http://amp.pharm.mssm.edu/Enrichr/
INRICH	https://atgu.mgh.harvard.edu/inrich/
GO-Elite	http://www.genmapp.org/go_elite/
HTSanalyzeR	https://bioconductor.org/packages/release/bioc/html/HTSanalyzeR.html
MetPA	http://www.metaboanalyst.ca/
KEA	http://amp.pharm.mssm.edu/X2K/
DIANA-miRPath	http://snf-515788.vm.okeanos.grnet.gr/
Ontologizer	http://ontologizer.de/
GeneTrailExpress	https://genetrail2.bioinf.uni-sb.de/
GENECODIS	http://genecodis.cnb.csic.es/

should instead serve as a starting point for users to develop their own analytical workflows. Users should also keep in mind that GO annotations and databases are constantly being updated, so using different versions of the same resource may result in different outputs depending on temporal constraints. As new technologies continue to emerge and the cost of high-throughput experiments continues to fall, there will be a constant refinement of existing annotations to fit new advancements in scientific understanding. As such, using the most up-to-date resources is always recommended.

Closing Remarks

This article focuses on the various methods used for enrichment analysis and some of the popular tools and their properties **Table 2**. Using a test-case dataset, the steps involved in performing a functional enrichment analysis using the DAVID databases is depicted. A functional enrichment analysis is the initial step in any high-throughput experiment, allowing us to narrow down to specific set of genes/functions/pathways of relevance to the problem in hand. The information provided in this article could be considered as a fundamental resource which can be built upon to facilitate the functional enrichment analysis of any datasets.

See also: Computational Pipelines and Workflows in Bioinformatics. Exome Sequencing Data Analysis. Functional Enrichment Analysis Methods. Gene Prioritization Tools. Gene Prioritization Using Semantic Similarity. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Whole Genome Sequencing Analysis

References

- Bauer, S., Grossmann, S., Vingron, M., Robinson, P.N., 2008. Ontologizer 2.0 – A multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics* 24, 1650–1651.
- Berriz, G.F., Beaver, J.E., Cenik, C., Tasan, M., Roth, F.P., 2009. Next generation software for functional trend analysis. *Bioinformatics* 25, 3043–3044.
- Carbon, S., Ireland, A., Mungall, C.J., *et al.*, 2009. AmiGO: Online access to ontology and annotation data. *Bioinformatics* 25, 288–289.
- Carmona-Saez, P., Chagoyen, M., Tirado, F., Carazo, J.M., Pascual-Montano, A., 2007. GENECODIS: A web-based tool for finding significant concurrent annotations in gene lists. *Genome Biol.* 8, R3.
- Draghici, S., Khatri, P., Bhavsar, P., *et al.*, 2003. Onto-Tools, the toolkit of the modern biologist: Onto-Express, Onto-Compare, Onto-Design and Onto-Translate. *Nucleic Acids Res.* 31, 3775–3781.
- Eden, E., Navon, R., Steinfeld, I., Lipson, D., Yakhini, Z., 2009. GOrilla: A tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinform.* 10, 48.
- Gardina, P.J., Clark, T.A., Shimada, B., *et al.*, 2006. Alternative splicing and differential gene expression in colon cancer detected by a whole genome exon array. *BMC Genomics* 7, 325.
- Gene Ontology Consortium, 2015. Gene Ontology Consortium: Going forward. *Nucleic Acids Res.* 43, D1049–D1056.
- Gotz, S., Garcia-Gomez, J.M., Terol, J., *et al.*, 2008. High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic Acids Res.* 36, 3420–3435.
- Hill, D.P., Berardini, T.Z., Howe, D.G., Van Auken, K.M., 2010. Representing ontogeny through ontology: A developmental biologist's guide to the gene ontology. *Mol. Reprod. Dev.* 77, 314–329.
- Hoehndorf, R., Schofield, P.N., Gkoutos, G.V., 2015. The role of ontologies in biological and biomedical research: A functional perspective. *Brief. Bioinform.* 16, 1069–1080.
- Huang, D.W., Sherman, B.T., Lempicki, R.A., 2008. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* 4, 44–57.

- Huang, D.W., Sherman, B.T., Lempicki, R.A., 2009. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* 37, 1–13.
- Huang, D.W., Sherman, B.T., Tan, Q., *et al.*, 2007. DAVID Bioinformatics Resources: Expanded annotation database and novel algorithms to better extract biology from large gene lists. *Nucleic Acids Res.* 35, W169–W175.
- Keller, A., Backes, C., Al-Awadhi, M., *et al.*, 2008. GeneTrailExpress: A web-based pipeline for the statistical evaluation of microarray experiments. *BMC Bioinform.* 9, 552.
- Khatri, P., Draghici, S., *et al.*, 2005. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics* 21, 3587–3595.
- de Leeuw, C.A., Neale, B.M., Heskes, T., Posthuma, D., 2016. The statistical properties of gene-set analysis. *Nat. Rev. Genet.* 17, 353–364.
- Maere, S., Heymans, K., Kuiper, M., 2005. BiNGO: A Cytoscape plugin to assess overrepresentation of Gene Ontology categories in Biological Networks. *Bioinformatics* 21, 3448–3449.
- Mootha, V.K., Lindgren, C.M., Eriksson, K.-F., *et al.*, 2003. PGC-1 [alpha]-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.* 34, 267.
- Pathan, M., Keerthikumar, S., Ang, C.-S., *et al.*, 2015. FunRich: An open access standalone functional enrichment and interaction network analysis tool. *Proteomics* 15, 2597–2601.
- Rhee, S.Y., Wood, V., Dolinski, K., Draghici, S., 2008. Use and misuse of the gene ontology annotations. *Nat. Rev. Genet.* 9, 509–515.
- Stöckel, D., Kehl, T., Trampert, P., *et al.*, 2016. Multi-omics enrichment analysis using the GeneTrail2 web service. *Bioinformatics* 32, 1502–1508.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci.* 102, 15545–15550.
- Tipney, H., Hunter, L., 2010. An introduction to effective use of enrichment analysis software. *Hum. Genomics* 4, 202.
- Yang, D., Li, Y., Xiao, H., *et al.*, 2008. Gaining confidence in biological interpretation of the microarray data: The functional consistence of the significant GO categories. *Bioinformatics* 24, 265–271.

Relevant Websites

<http://amigo.geneontology.org/amigo/>
AmiGO Gene Ontology.

<https://www.blast2go.com/>
Blast2GO.

<http://cran.r-project.org/web/packages/GOplot>
CRAN-R - R Project.

<https://david.ncifcrf.gov/>
DAVID Functional Annotation Bioinformatics Microarray Analysis.

<http://david.abcc.ncifcrf.gov/knowledgebase>
DAVID Knowledgebase.

<http://www.funrich.org/>
FunRich.

<http://www.geneontology.org>
Gene Ontology Consortium.

<http://cbl-gorilla.cs.technion.ac.il/>
GOrrilla.

Biographical Sketch



Srikanth Manda has completed his Masters in Biotechnology from Indian Institute of Technology, Bombay, thereafter, pursued a PhD in Bioinformatics from Institute of Bioinformatics, Bangalore and Johns Hopkins University, USA. During his Ph.D. work he has analyzed datasets from Whole Genome (WGS), Whole Exome (WES), Transcriptome, epigenome, miRNAome, proteome and phosphoproteome. His research interests involve multi-omics data analysis, integration and interpretation from high-throughput platforms using available tools and cloud based workflows.



Sudhir Jadhao is pursuing PhD in Genome Informatics at Queensland University of Technology. He specializes in NextGen sequencing and have analyzed data from diverse platforms (Illumina, PacBio, and Ion Torrent) and approaches (WGS, RNA-seq, miRNA, ChIP-seq, Microarray, Exome Sequencing, CNV). In his previous role as Research Assistant, Sudhir worked on understanding the role of non-coding RNA in regulation of the prostate cancer at the University of Macau.



Shivashankar H. Nagaraj is Advanced Queensland Research Fellow at Queensland University of Technology. The focus of his group is on the development and application of bioinformatics approaches that utilize large amounts of data to solve problems in Genomics and Proteomics. He has developed multiple Bioinformatics solutions (e.g., ESTExplorer, EST2Secretome, PGTools) that are valuable and of practical utility to the wider scientific community.

Prediction of Coding and Non-Coding RNA

Ranjeev Hari, Perdana University, Selangor, Malaysia

Suhanya Parthasarathy, Monash University Malaysia, Segamat, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Early biologists who primarily focused on prokaryotes noted that the majority (85%–90%) of genes were protein-coding and had them accounted for the major cellular processes. However, while complexity of organisms as driven by evolution became higher, the proportion of coding genes reduced gradually in relation to genome size (Mattick and Makunin, 2006). In one of the largest efforts to characterise human deoxyribonucleic acid (DNA), the Encyclopaedia of DNA Elements (ENCODE) project delineates that only about 2%–3% of the human genome are responsible for coding proteins. These genes which are transcribed appear minute in quantity when compared to the transcription of the non-coding regions which are at least 75% of the total genome (Djebali *et al.*, 2012).

The Central Dogma of Molecular Biology states that the genes from the genome (DNA) shall be transcribed into ribonucleic acid (RNA) molecules as per the quantum and mechanistic laws of physics and chemistry. Proteins are required to catalyse various cellular processes, and they are made when these RNAs called messenger RNAs (mRNA) are translated into amino acid sequence. The sudden discovery of non-coding RNAs (ncRNAs) in 1950s, however, were used to dispute the centrality of the Central Dogma. The infamous terms used for these non-coding DNA regions in the genome were “junk DNA” and is coupled with transcriptional “noise” for its RNA counterpart. By the 1970, there were key developments which suggested the role of ncRNAs in gene expression regulation of eukaryotes (Britten and Davidson, 1969; Kohne, 1970; Tomkins *et al.*, 1969). Till date, several possible functions are known for ncRNAs namely as epigenetic, transcriptional and post-transcriptional regulators often participating in DNA replication, splicing, translation, chromosome structure stability, and genome defence (Cheng *et al.*, 2005; ENCODE, 2007; Morris, 2012; Washietl *et al.*, 2007). Moreover, ncRNAs are also thought to have complex roles in cancer and other diseases. As a result, huge efforts are ongoing in trying to understand better the various classes of ncRNAs of its roles and functions in living systems.

RNA Universe

RNA is one of the three essential biopolymers apart from DNA and protein for biological life and are intermediaries to DNA and protein function as per the central dogma. It is an important catalyst for various biological processes and in some instances, once phosphorylated it can even become a subunit of ATP, NADH, and other metabolic compounds. Being commonly found in single-stranded form, it is flexible to take on a 3-dimensional structure better than the double-stranded DNA. Consequently, its structure has protein-like hierarchy that are defined into secondary and tertiary level of structure. Some of the diverse types of secondary structural elements are hairpin loops, 180° bend backbone, internal loops where sequences cannot form base pairs, multibranch loops where two or more regions form a close structure, and other bulge loops. These secondary structural elements can display distinct functions in a cell and are usually structurally conserved (Mathews *et al.*, 2004).

With diverse functions, and as new RNA are being discovered and studied there are different types of RNA classification systems that emerged. Most commonly, RNAs are classified by function and size. Functionally; as examples, RNAs; such as, messenger RNA (mRNA); better known as coding gene or coding RNA are involved in protein synthesis, ribosomal RNA (rRNA) that produce ribosome and aid mRNA by interacting with tRNA during translation, transfer RNA (tRNA) that are in triplet base carrying amino acid precursors for protein anabolism, RNA that involved in post-transcriptional activity are small nuclear RNA (snRNA) that are important in introns removal of heterogenous nuclear RNA (hnRNA), small nucleolar RNA (snoRNA) are essential in RNA mutation, ribonuclease P (RNase P) acts as a ribozyme that cleaves precursor sequences of tRNA, telomerase RNA (TER) are involved in eukaryotic RNA function of extending telomeres and small-interfering RNA (siRNA) are gene silencers and regulators. RNAs are also divided based on size into three groups; which is small ncRNAs, medium ncRNAs, and long ncRNAs (Akman and Erson Bensen, 2014; Brosius and Raabe, 2016; Gomes *et al.*, 2013a; Ma *et al.*, 2013; Santosh *et al.* 2015). An instance of the organisation of the RNA landscape by size is described in Fig. 1.

Coding RNA

Central Dogma proposed by Francis Crick explained the conversion flow from DNA which carries genetic information to protein which is a functional product (Crick, 1970). It is also a 2-key steps gene expression process, where transcription produces mRNA by RNA polymerase and translation convert information from mRNA into protein sequence. While RNA mediates majority of the transfer of information based on the proposed model by Crick (Table 1), wherever RNA is being expressed with no translation to protein, it is then largely labelled simply as RNA or non-coding RNA (ncRNA).

In coding genes, DNA will be transcribed into pre-mRNA; an immature single strand of mRNA, which will be processed into mature mRNA transcripts that are functional. In eukaryotes, 3'-end cleavage of transcripts generated by RNA polymerase II (pol II)

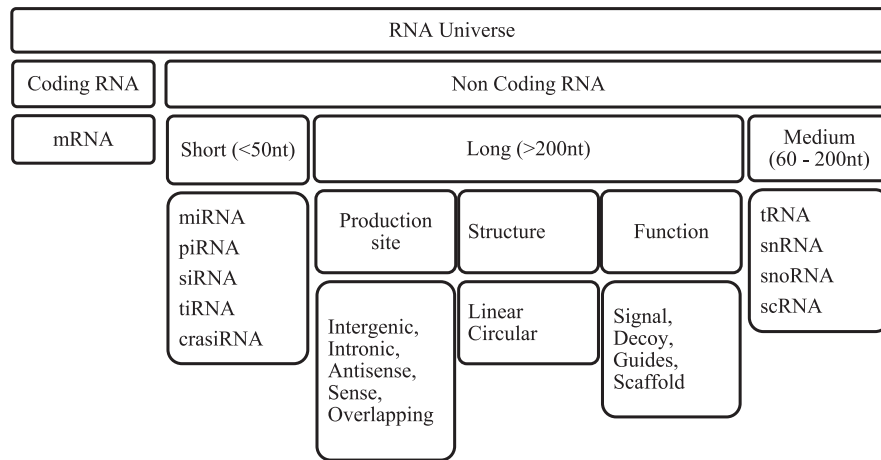


Fig. 1 RNA universe classification based on coding and non-coding RNA.

Table 1 Nine possible transfers in three classes as per Crick's central dogma

Class	Possible transfer
General transfer	DNA replication
	Transcription
	Translation
Special transfer	RNA replication
	Reverse transcriptase
	Protein synthesis without mRNA template
Unknown transfer	

Source: Reproduced from Crick, F., 1970. Central dogma of molecular biology. Nature 227, 561–563.

is a universal step of gene expression that proceeds through the recognition of cis-acting elements of the pre-messenger RNA (mRNA). It will then undergo 5'-capping, followed by intron removal and methylation, as well as 3'-cleavage and polyadenylation (Fig. 2; Millevoi and Vagner, 2009). A mature mRNA, depending on the host organism, can be translated to protein based on a triplet sequence called codon for each amino acid by a specific codon to amino acid translation table.

Gene prediction programs makes use of the molecular and structural signatures of genes as important features in software algorithms predicting protein coding mRNA transcripts. Other approaches based on sequence similarity (homology) search by using BLAST searches are commonly used especially when high quality data is available in a sequence database of genes, EST or protein. However, since only half of genes being discovered have significant homology to databases, *ab initio* gene prediction methods are used instead. Gene prediction programs that are using algorithms such as dynamic programming, Hidden Markov Models (HMM), and neural networks includes GeneID, GENSCAN, FGENSESH, and Grail (Blanco *et al.*, 2007; Burge and Karlin, 1997; Salamov and Solovyev, 1998; Xu *et al.*, 1996).

Non-Coding RNA

Essentially, RNA or ncRNA is a functional RNA molecule that are functional without having to translate into protein. Common functional RNA that are involved in cellular housekeeping includes the transfer RNA (tRNA) and ribosomal (rRNA) which are known to be involved with mRNA during translation. The many classes of ncRNAs have diverse yet essential roles in biological processes, including ribonucleoprotein (RNP) in translation, snoRNA in alternative RNA splicing regulation, Y RNAs in DNA replication, enhancer RNAs (eRNAs) which promote gene expression in gene regulation and other epigenetic ncRNAs such as miRNA, siRNA, piRNA and lncRNA involved in modifications to various DNA processes (Collins *et al.*, 2011; Storz, 2002). However, of the many ncRNAs being newly identified, most have no known function and are simply identified as junk RNA (Palazzo and Lee, 2015). Nevertheless, large bodies of evidence are being accrued that implicate ncRNA such as miRNAs and snoRNAs in diseases; such as, Autism, Alzheimer's and even some type of cancers (Ding *et al.*, 2008; Nakatani *et al.*, 2009; Shen *et al.*, 2009) (Fig. 3).

In previous studies, efforts in identifying ncRNAs often centred in experiments involving complementary DNA (cDNA) cloning and genomic tiling arrays (Lund *et al.*, 2014; Yazaki *et al.*, 2007). Some experimental strategies utilize *in vitro* translation to distinguish between ncRNAs and mRNAs where positive translation indicate that a transcript is an mRNA (Borsani *et al.*, 1991;

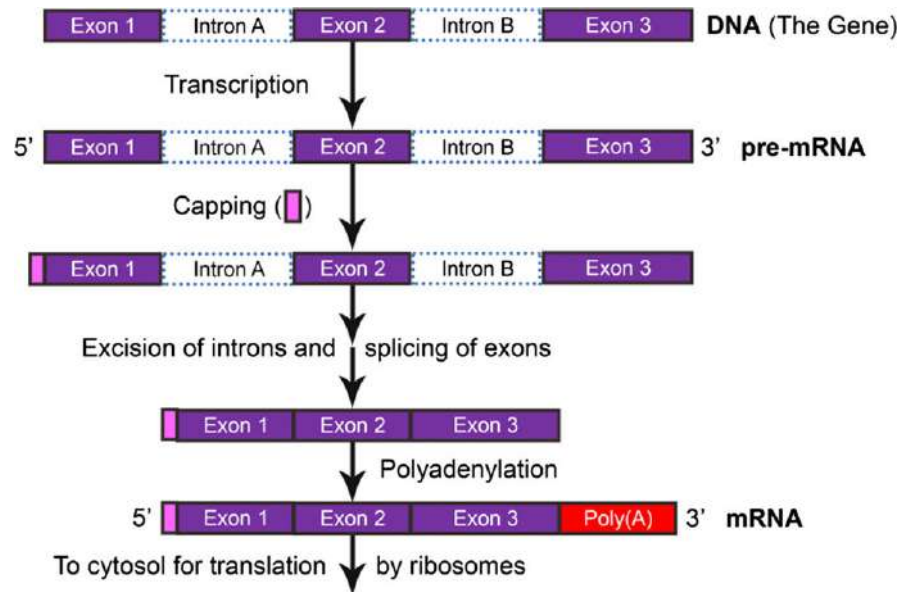


Fig. 2 The transcription process of information from DNA to RNA.

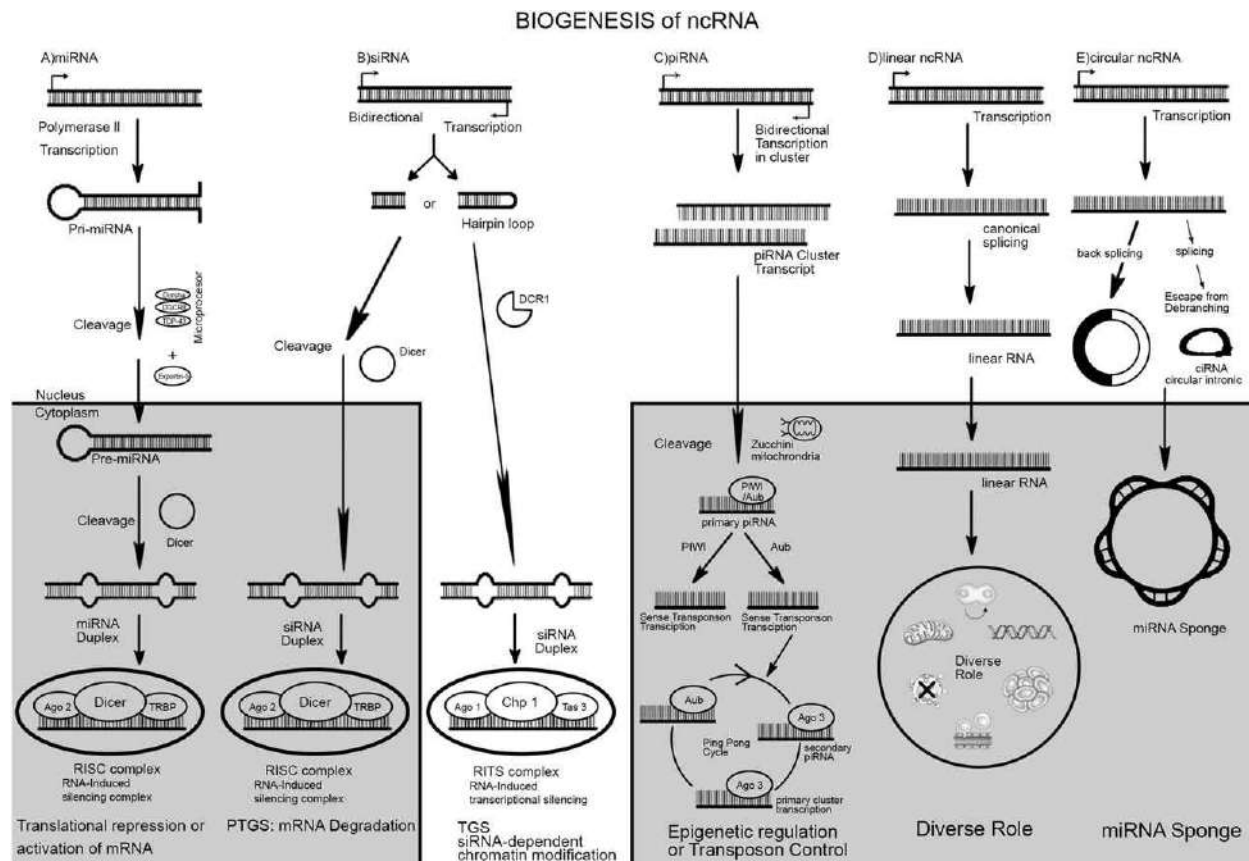


Fig. 3 Biogenesis of miRNAs, siRNAs, piRNAs and lncRNAs.

Galindo *et al.* 2007; Tupy *et al.* 2005). These experiments are often time-consuming and costly, requiring enough RNA samples while being limited to identifying long ncRNAs (lncRNAs). Researchers are increasingly using computational methods to offset these limitations by incorporating them in their experimental methods especially with the reduction in cost of high-throughput technologies such as next generation sequencing (NGS).

Coding and Non-Coding Prediction Approaches

Differentiating non-coding and coding regions in a genome can be approached computationally, moreover, it may be viewed as a classic type of statistical categorisation problem. Computationally, identifying and cataloguing vast information from genomic and transcriptomic data can be mainly divided into *ab initio* and evidence-based prediction methods while some are based on a hybrid of both methods. Generally, in *ab initio* prediction, the algorithm distinguishes from coding and non-coding genes relatively quickly from the genomic and transcript sequences provided as input. The prediction is derived upon known features of the coding gene such as the start and stop codons, open reading frame, splice and polyadenylation sites. Such *a priori* information will be applied as features in machine learning algorithms for better prediction accuracy. In contrast, evidence-based prediction method would use known information from DNA/RNA and protein sequences to generate predictions by mapping and comparison. In order to curtail the high false positive rates in *ab initio* prediction methods, supported by dwindling costs of NGS, evidence-based prediction is made practical for application in larger datasets. Furthermore, the application of both *ab initio* and evidence-based prediction into a hybrid approach is a natural progression in the field of gene prediction where, for instance, the use of gene expression evidence and subsequent genome mapping in generating better *ab initio* gene models is having widespread attention. Given transcript RNA sequences, many prediction software can predict its class of coding potential with relatively high-accuracy.

Several tools are available to discriminate mRNA and RNA which work on diverse set of features which are often shared in its approaches (Fan and Zhang, 2015). Often some software tools; such as, CNCTDiscriminator which utilizes machine learning as an ensemble of several statistical features of base composition, open-reading frame (ORF), and secondary structures in generating its predictions (Biswas *et al.*, 2013). Other example of software widely used in distinguishing coding and non-coding transcripts are COME (Hu *et al.*, 2017), CPC (Kang *et al.*, 2017; Kong *et al.*, 2007), CNCI (Sun *et al.*, 2013), CPAT (Wang *et al.*, 2013b), PhyloCSF (Lin *et al.*, 2011), PLEK (Li *et al.*, 2014), FEELnc (Wucher *et al.*, 2017), Lncrna-ID (Achawanantakun *et al.*, 2015) and PORTRAIT (Arrial *et al.*, 2009). These tools attempt to achieve similar goals of classifying mRNA and RNA, however, the focus for some are in specific organisms and RNA classes while others are general purpose classifiers.

Besides differentiating coding and non-coding RNAs, these computational methods could also be used in identifying and cataloguing functional RNAs by its class (Fig. 4). Evidence-based methods utilize homology information of the nucleotide sequences and sometimes its structural properties as well. Homology-based software tools using sequence homology includes Basic Local Alignment Search Tool (BLAST) (Johnson *et al.*, 2008), SSearch (Goujon *et al.*, 2010), BLAST-like alignment tool (BLAT) (Kent, 2002); while structural homology tools that usually have better prediction accuracy include INFERNAL (Nawrocki *et al.*, 2009) and RSearch (Klein and Eddy, 2003). Apart from homology-based software, ncRNA can be identified using *ab initio* algorithms such as CRITICA (Badger and Olsen, 1999) and ESTscan (Iseli *et al.*, 1999) based on sequence features; MiPred (Jiang *et al.*, 2007), Mfold (Zuker, 2003) and RNAfold (Hofacker, 2003) based on structure while miRDeep (Friedländer *et al.*, 2011) and snoSeeker (Yang *et al.*, 2006) based on sequence features from deep sequencing data (Wang *et al.*, 2013a). Moreover, hybrid integration of both homology and *ab initio* algorithms in RNA prediction is also applied in some software such as MASTR (Lindgreen *et al.*, 2007), pTRNAPred (Gupta *et al.*, 2014), and Infernal (Nawrocki *et al.*, 2009).

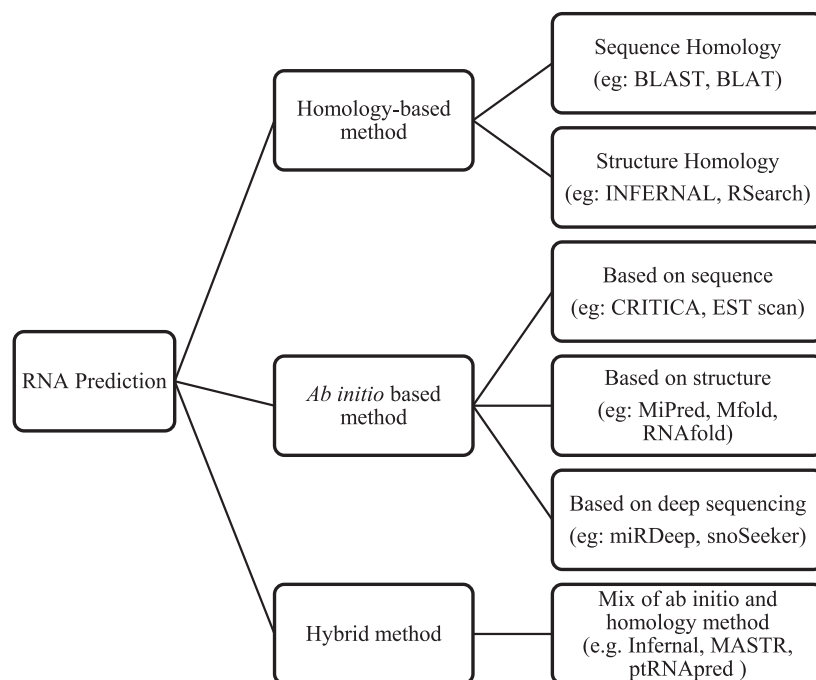


Fig. 4 Software tools for ncRNA prediction is divided into three methods: homology-based, *ab initio*, and hybrid method.

Small RNA

Sequences of RNAs within 50 nucleotides are classified as small non-coding RNA and are known to primarily control stability of mRNA translation by repressing or activating binding of ribosome to mRNAs (Storz *et al.*, 2004). Example of small ncRNAs that are commonly found in eukaryotes are microRNA (miRNA), piwi-interacting RNA (piRNA), short-interfering RNA (siRNA), tRNA-derived stress-induced RNA (tiRNA), and centromere repeat-associated small RNA (crasiRNA). The three major types of small ncRNAs play important role in gene silencing pathways especially in eukaryotes and protecting cell/genome against viruses, mobile repetitive DNA sequences, retro-elements and transposons are miRNAs, siRNAs and piRNAs (Gomes *et al.*, 2013b; Moazed, 2009).

One of the general purpose bioinformatics software that can be used to predict small ncRNA is RNAz (Gruber *et al.*, 2010; Washietl *et al.*, 2005). RNAz uses a combined approach of support vector machines (SVM) regression with binary classification technique and applies minimum free energy (MFE) structural prediction algorithms. Two other important features used by RNAz in this approach are RNA's intrinsic unusual thermodynamic stability (by comparing MFE of native sequence to large number of random identical in length and base composition sequences) and structural conservation (by using RNAalifold) (Gruber *et al.*, 2010; Washietl *et al.* 2007, 2005). Fig. 5 is given to better illustrate an example of sRNA prediction workflow using RNAz.

Microna (Mirna)

As one of the most studied class of ncRNA, microRNA (miRNA) are small, endogenous, single-stranded RNAs approximately 22 nucleotides/bases in size. They are found in both animals and plants and are evolutionary conserved units (Baulcombe, 2004; Zhang *et al.*, 2007). As small regulatory molecules, they are expressed independently, possess their own promoters and can be regulated by a distinctive set of transcription factors (Gulyaeva and Kushlinskiy, 2016). The target of miRNA commonly is messenger RNA (mRNA).

MicroRNAs functionally govern cell proliferation (Hwang and Mendell, 2006), differentiation and apoptosis (Xia *et al.*, 2017). They are sometimes related in regulation and reprogramming of cellular metabolism in cancer. In some instances, deregulated miRNAs have been correlated with cancers. For example several types of miRNAs that have been proved to be up regulated in cancerous tissues such as let-7e, miR-151p, miR-222, miR-21, miR-155, and miR-221 (Sarkar *et al.*, 2010). This RNA class is well known in affecting transcription levels in animals exerted by major silencing mechanism from destabilization through a cleavage-independent process (MacFarlane and Murphy, 2010). Moreover, miRNAs are noted in diverse functions in cellular life including cell cycle, development, signal transduction, metabolism, metastasis, cell proliferation, differentiation and apoptosis (Mahmoudi and Cairns, 2016).

Computationally, several prediction algorithms have been developed in identifying miRNAs. For instance, miRscan compares the identified putative structures with known miRNA features like 3' and 5'-stem conservation while another tool, miRseeker, attempts to select hairpins sharing similar nucleotide divergence patterns to a reference set (Lai *et al.*, 2003; Lim *et al.* 2003; Terai *et al.*, 2007). These two main tools identify sequences that can form hairpin structures based on RNAfold and Mfold based on the conserved intragenic sequences. Besides that, there are several tools that have been developed to predict miRNAs for a variety of species, for example, the HMM-based tool ProMir, and improved ProMir II (Nam *et al.*, 2006, 2005), MiRRim is another HMM-based to achieve high-performance identification of new human miRNAs (Terai *et al.*, 2007), while HHMMiR predicts de novo miRNA hairpins in the absence of evolutionary conservation (Kadri *et al.*, 2009).

Piwi-Interacting RNA (Pirna)

Piwi-interacting RNAs (piRNAs) are a distinct class of miRNAs which are mainly expressed in germline cells with approximately 24–32 nucleotides in length (Aravin *et al.*, 2006). Setting it apart from miRNAs, it has a bias for 5'-uridine common to piRNAs in vertebrates and invertebrates and known to possess lesser secondary structure conservation. They are generally detected with Piwi proteins as they get processed into piRNAs from longer precursor transcripts (Aravin *et al.*, 2006; Lau *et al.*, 2006). Piwi-interacting RNAs are important in genetic and epigenetic regulatory factors for germ cell maintenance, genome integrity, mRNA stability, DNA

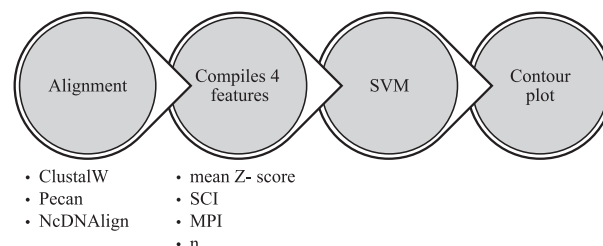


Fig. 5 The workflow of the RNAz non-coding RNA prediction pipeline. Legend: mean z-score: average folding energy z-score, SCI: Structure conservation index, MPI: Sequence divergence of the alignment, and n: Number of aligned sequences. Adapted from Gruber, A.R., Findeiss, S., Washietl, S., Hofacker, I.L., Stadler, P.F., 2010. RNAz 2.0: Improved noncoding RNA detection. *Biocomputing* 2010, 69–79.

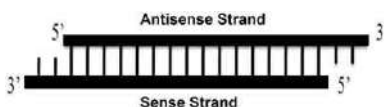


Fig. 6 General structure of siRNA.

methylation and retrotransposon control (Zuo *et al.*, 2016). It has some role in gene silencing and are involved in cancer pathways such as piRNAs piR-651 found to be overexpressed in many of cancer cell lines, piR-823 seen in many types of cancer development, and piR-Hep1 is up regulated in hepatic cell carcinoma (Cordeiro *et al.*, 2016; Ng *et al.*, 2016). Having such significance, computational methods offer quick identification of putative piRNA sequences.

Generally, a few of the computational prediction tools for piRNAs include Luo's method, Betel's method, piRNApredictor and Piano. Each attempt to differentiate transposon-derived piRNAs from non-piRNAs from different sequence and physiochemical features such as spectrum profile, mismatch profile, subsequence profile, position-specific scoring matrix, pseudo dinucleotide composition, and local structure-sequence triplet elements (Betel *et al.*, 2007; Luo *et al.*, 2016; Wang *et al.*, 2014; Zhang *et al.*, 2011). Recently, IpiRID webserver tool improved on several of the previous software tools prediction accuracy on human, mouse and fly datasets (Boucheham *et al.*, 2017).

Short Interfering RNA (Sirna)

Short interfering RNAs are generated from dsRNAs (double strand RNAs) and consist of two RNA strands which is antisense and a sense strand that form a duplex 19–25 bp in length with 3' dinucleotide overhangs as shown in Fig. 6. The antisense strand is a perfect reverse complement of the intended target mRNA.

Some of the functions and roles of siRNAs are mainly in post-transcriptional gene silencing (PTGS) or translational inhibition, protection from exogenous DNA, participation in epigenetic mechanisms and to maintain genome integrity via transcriptional silencing. Due to its ability to knockout genes, it has been exploited for commercial applications to rapidly study gene function *in vivo*. Thus, much of the computational applications of siRNA is in the design of optimal targeting of siRNA sequences to knockout genes. Thereafter, prediction of siRNAs can be used to develop screening protocols that can be used to find novel pathway, validating targets of cellular processes associated to diseases such as cancer, HIV infection, or hepatitis.

Computational prediction for siRNA is Sylamer, method to find significantly over or under-represented words in sequences according to a sorted gene list. Usually used to find significant enrichment or depletion of microRNA or siRNA seed sequences from microarray expression data.

Medium Non-Coding RNA

Mid-size RNA class have lengths that range from 60 to 200 nt which include transfer RNA (tRNA), small nucleolar RNA (snoRNA), small nuclear RNA (snRNA) and small cytoplasmic RNA (scRNA) (Bhan and Mandal, 2014).

Transfer RNA (tRNA)

The transfer RNA (tRNA) is a common RNA molecule 76–90 nt in length and serves as an adaptor molecule between mRNA and protein. During translation, tRNA acts as a carrier of amino acids to growing polypeptide chain. Besides that, tRNA is also known to be involved in non-protein synthetic processes in retrovirus life cycle. (Sharp *et al.*, 1985).

The tRNA has a cloverleaf structure which consists of four helices and three hairpin loops. The four helices are the acceptor (AA) helix (or stem) carrying charged amino acid by the cognate aminoacyl synthetase, the dihydrouridine (D) hairpin containing the modified base dihydrouracil; the anticodon (AC) hairpin having the apical loop that presents the anticodon triplets; and the thymine (T) hairpin containing the thymine base unusual in RNAs (Westhof and Auffinger, 2001). By having such highly conserved structural characteristics, established computational tools to predict tRNA from sequences are available such as tRNAscan SE and ARAGORN (Laslett and Canback, 2004; Lowe and Eddy, 1997).

As tRNA perform essential biological functions, mutation of tRNA may cause serious human disease, including pathological stress injuries, cancer and neurodegenerative disease. A special bioinformatics tool for tRNA sequence modification investigation is tRNAmopred. The tool allows the prediction of post-transcriptional modifications of tRNAs in regards to position and modification type (Machnicka *et al.*, 2016).

Small Nucleolar RNA (Snorna)

Small Nucleolar RNA (snoRNA) is one of the medium-sized RNA which is approximately 60 to 170 nt in size and prominent in both Archae and Eukaryotes, however, absent in bacteria (Jorjani *et al.*, 2016; Bhattacharya *et al.*, 2016). The locus of snoRNA biosynthesis in yeast originates from a monocistron while for plants in a polycistron. Mainly, snoRNAs play an important role in guiding other

RNA's chemical modification particularly in rRNAs and snRNAs, which are divided into three motif-based classes of antisense C/D box with 2'-O-methylation of ribonucleotides, antisense H/ACA box with pseudouridylation and small Cajal body-specific RNAs (Bhattacharya *et al.*, 2016). Loss of copy of snoRNAs leads to some debilitating human diseases such as Prader-Willi (PWS) and Angelman (AS) syndrome. Implicating complex neurogenetic disorders, PWS is characterized by intellectual impairment, obesity and type 2 diabetes, and short stature while AS symptoms show mental retardation or small head and specific appearance (Ding *et al.*, 2008; Skryabin *et al.*, 2007). In PWS, the cause is attributed to gene imprinting epigenetics on chromosome 15q11.2 and micro-deletion of the paternally inherited 29 copies of SNORD116. On the other hand, AS is attributed to the loss of maternal region chromosome 15 of Ube3A gene. Conversely, duplication of MBII52 snoRNA closely related to common neuropsychiatric disorder with impaired communication disorder, autism by affecting the serotonin 2c receptor (Nakatani *et al.*, 2009).

A number of bioinformatics tool are available for snoRNA detection such as snoScan for C/D box methylation-guide snoRNAs and snoGPS servers for H/ACA pseudouridylation-guide snoRNAs. (Schattner *et al.*, 2005). Besides that, snoSeeker includes two programs (CDseeker and ACAseeker) for screening alignments of whole genome putative snoRNA candidates in order to identify snoRNAs (Yang *et al.*, 2006). Apart from this, snoReport, which is one of the bioinformatics snoRNAs approach with unknown targets, allow to recognize two major classes snoRNAs with combination of RNA secondary structure prediction and snoRNABase collection (Hertel *et al.*, 2007).

Small Nuclear RNA (Snrna)

Small nuclear RNA (snRNA) is one of the small RNA with an average size of 150 nt. Eukaryotic genomes code for a variety of non-coding RNAs and snRNA is a class of highly abundant RNA, localized in the nucleus with important functions in intron splicing and other RNA processing (Maniatis and Reed, 1987). Generally, in transcript splicing, snRNA presents as a ribonucleoprotein particles (snRNPs) along with additional proteins that form a large particulate complex (spliceosome) bound to the unspliced primary RNA transcripts in order to mediate the process. Besides splicing, additional evidence indicate snRNPs function in nuclear maturation of primary transcripts in mRNAs, gene expression regulation, splice donor in non-canonical systems and in 3'-end processing of replication-dependent histone mRNAs (Buratti and Baralle, 2010; Ideue *et al.*, 2012; Lasda and Blumenthal, 2011).

Long Non-Coding RNA

Transcribed RNAs with more than 200 nt long are generally classified as long non-coding RNA (lncRNA). This class of RNA is large and diverse and could be further classified by position of transcription (Intergenic, Intronic, Antisense, Sense and Overlapping ncRNAs) (Fig. 7), structure of RNA (Linear, Circular) and functionality (Bhan and Mandal, 2014; Peschansky and Wahlestedt, 2014). When functionally classified, there are four subcategories divided by mode of action; such as, lncRNAs, which act as either signals, decoys, guides, or scaffolds (Nie *et al.*, 2015).

Alteration in lncRNAs is closely related to human neurological disease, cancer as well as aging. Over expressed of lncRNAs which located on chromosome 2q32 causes prostate cancer apart from abnormal of lncRNAs splicing event allow tumour-suppressor lncRNAs, metastasis associated lung adenocarcinoma transcript 1 (MALAT1) widely involved of lung cancer, pancreatic cancer and cervical cancer. Besides, MEG3, HOTAIR, and lincRNA-p21 act as oncogenic ncRNAs to allow them associated with disease susceptibility. For instance, loss and imprinted MEG3 may contribute to tumour progression; such as meningiomas, colorectal carcinoma, and type 1 diabetes, respectively. (Bhan and Mandal, 2014; Chen *et al.*, 2016). Autosomal dominant facioscapulohumeral muscular dystrophy (FSHD) is caused by deletion of chromosome 4q35; yet, Duchenne muscular dystrophy occur commonly and severely due to deregulation of lncRNAs expression. (Neguembor *et al.*, 2014). In addition, elevated antisense lncRNA transcript of beta secretase 1 (BACE1) is correlated to Alzheimer's disease (Faghihi *et al.*, 2008).

As outlined previously, the identification and study of lncRNAs is of major relevance to human biology and disease since they represent an extensive, largely unexplored, and functional component of the genome (Mattick and Makunin, 2006; Ponting *et al.*, 2009). For instance, although some lncRNAs have been shown to affect in human diseases, these studies are limited by the lack of lncRNA annotations. Therefore, the identification of high-quality catalogues of lncRNAs and its expression in tissues is important work. On the contrary, similar information for protein-coding genes has long been obtainable.

In order to facilitate the growing interest in ncRNA and particularly lncRNA, generating high-confidence catalogues of lncRNA for further study is crucial. Genome sequencing efforts in association with RNA-seq in several studies were able to catalogue lncRNA of various genomes of the sponge, mouse, bovine, and human. These studies have shown that a large portion of these genomes are actually transcribed and could aid the discovery of many more non-coding transcripts (Carninci *et al.*, 2005; Harrow *et al.*, 2006). Furthermore, some studies have focused on large intergenic non-coding RNAs (lincRNAs) (Ponjavic *et al.*, 2007; Guttman *et al.*, 2009; Khalil *et al.*, 2009; Orom *et al.*, 2010), the non-coding transcripts which are away from protein-coding regions.

Recent advances in whole-transcriptome RNA sequencing (RNA-seq) and computational methods for transcriptome reconstruction now provide an opportunity to expansively annotate and characterize lncRNA transcripts (Garber *et al.*, 2011; Guttman *et al.*, 2010; Trapnell *et al.*, 2010). Certainly, an initial application of this method in three mouse cell types characterized the gene structure of more than a thousand mouse lincRNAs, most of which were not identified before (Guttman *et al.*, 2010). With logical frameworks that have

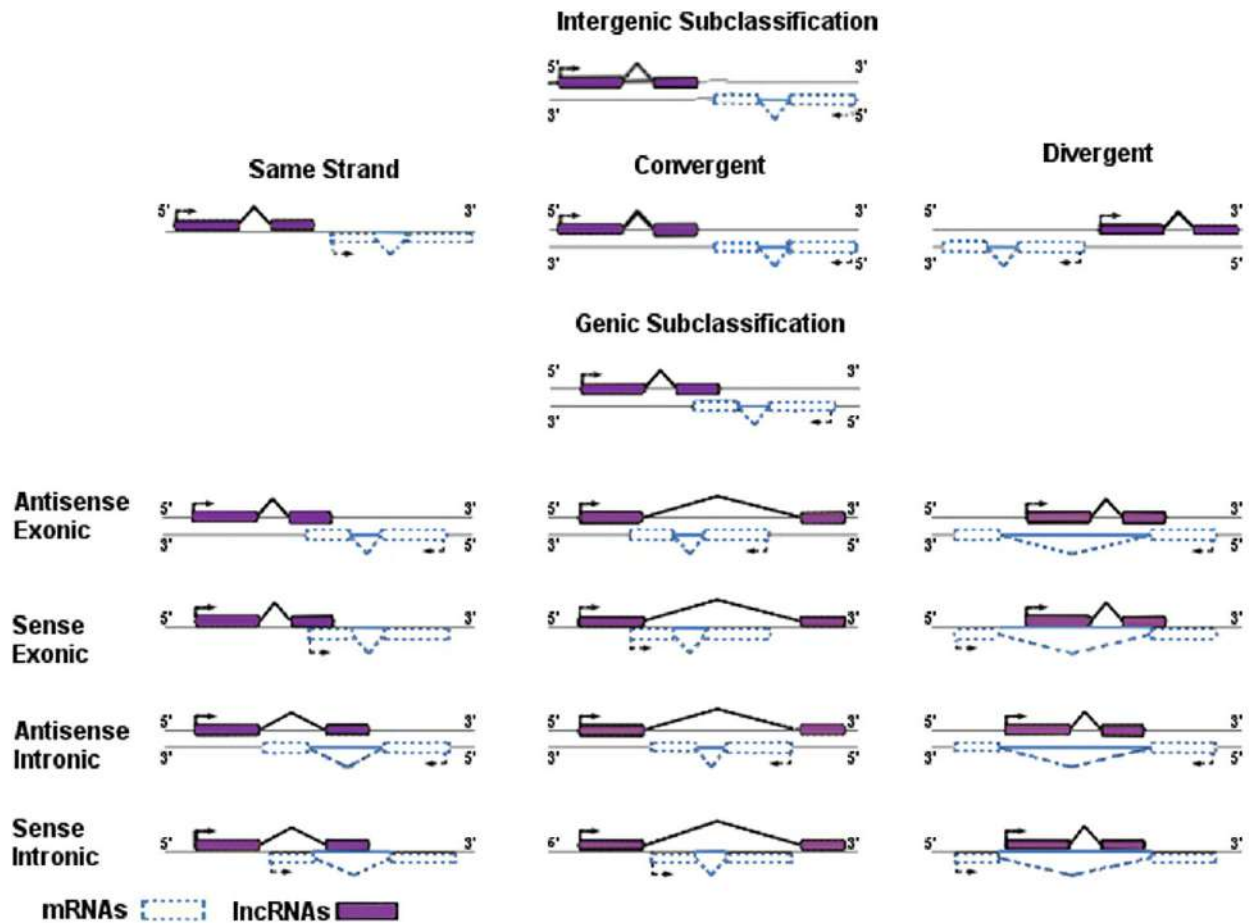


Fig. 7 Classification of lncRNA by transcript expression position in relation to gene and strand origin.

been thoroughly established in previous studies, it is now possible to catalogue and curate high-confidence lncRNA (Pauli *et al.*, 2012). Generally, a well-designed filtering pipeline can be used to arrive at putative lncRNA that can be further assessed for protein-coding capacity. The final set that exhibit poor protein-coding potential will remain as high-confidence lncRNA.

Many studies employ their own pipeline and criteria for filtering their transcript dataset from RNAseq. However, lately there has been software programs that execute the workflow systematically to catalogue the lncRNA; such as, lncrna-screen, FEELnc, and Annocript (Gong *et al.*, 2017; Peng *et al.* 2017; Wucher *et al.*, 2017). Generally, the lncRNA identification pipeline will require the assembly of reads into transcripts models which will then be merged, low-confidence transcripts removed, both potential and actual protein-coding transcripts removed, and having the final set of lncRNA transcripts classified (Ulitsky, 2016).

Besides computational identification, since lncRNAs are known to contribute widely in human diseases, computational methods to predict association with diseases are also available. LDAP (lncRNA-disease association prediction) is proposed to predict interaction between lncRNA with susceptibility disease and uses Smith-Waterman algorithm to calculate sequence similarity and several methods to calculate disease similarity (Lan *et al.*, 2016). Apart from LDAP, LRLSLDA (Laplacian Regularized Least Squares for lncRNA-Disease Association) that associates functionally similar disease with lncRNA as well as LNCSIM (LRLSLDA-lncRNA Functional Similarity Calculation Model) that calculate semantic similarity between associated disease are powerful computational approach which could help find probable relationship of lncRNA to diseases (Chen *et al.*, 2015).

See also: Characterizing and Functional Assignment of Noncoding RNAs. Natural Language Processing Approaches in Bioinformatics

References

- Achawanantakun, R., Chen, J., Sun, Y., Zhang, Y., 2015. lncRNA-ID: Long non-coding RNA IDentification using balanced random forests. *Bioinform. Oxf. Engl.* 31, 3897–3905. doi:10.1093/bioinformatics/btv480.
- Akman, H.B., Erson Bensan, A.E., 2014. Noncoding RNAs and cancer. *Turk. J. Biol.* 38, 817–828. doi:10.3906/biy-1404-104.

- Aravin, A., Gaidatzis, D., Pfeffer, S., *et al.*, 2006. A novel class of small RNAs bind to MILI protein in mouse testes. *Nature* 442, 203–207. doi:10.1038/nature04916.
- Arrial, R.T., Togawa, R.C., Brigido, M., de, M., 2009. Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: Case study of the pathogenic fungus *Paracoccidioides brasiliensis*. *BMC Bioinform.* 10, 239. doi:10.1186/1471-2105-10-239.
- Badger, J.H., Olsen, G.J., 1999. CRITICA: Coding region identification tool invoking comparative analysis. *Mol. Biol. Evol.* 16, 512–524.
- Baulcombe, D., 2004. RNA silencing in plants. *Nature* 431, 356.
- Betel, D., Sheridan, R., Marks, D.S., Sander, C., 2007. Computational analysis of mouse piRNA sequence and biogenesis. *PLOS Comput. Biol.* 3, e222. doi:10.1371/journal.pcbi.0030222.
- Bhan, A., Mandal, S.S., 2014. Long noncoding RNAs: Emerging stars in gene regulation, epigenetics and human disease. *ChemMedChem* 9, 1932–1956. doi:10.1002/cmdc.201300534.
- Biswas, A.K., Zhang, B., Wu, X., Gao, J.X., 2013. CNCTDiscriminator: Coding and noncoding transcript discriminator – An excursion through hypothesis learning and ensemble learning approaches. *J. Bioinform. Comput. Biol.* 11, 1342002. doi:10.1142/S021972001342002X.
- Blanco, E., Parra, G., Guigó, R., 2007. Using geneid to identify genes. *Curr. Protoc. Bioinform.* 3–4.
- Borsani, G., Tonlorenzi, R., *et al.*, 1991. Characterization of a murine gene expressed from the inactive X chromosome. *Nature* 351, 325.
- Boucheham, A., Sommar, V., Zehraoui, F., *et al.*, 2017. IpiRId: Integrative approach for piRNA prediction using genomic and epigenomic data. *PLOS ONE* 12, e0179787. doi:10.1371/journal.pone.0179787.
- Britten, R.J., Davidson, E.H., 1969. Gene regulation for higher cells: A theory. *Science* 165, 349–357.
- Brosius, J., Raabe, C.A., 2016. What is an RNA? A top layer for RNA classification. *RNA Biol.* 13, 140–144.
- Buratti, E., Baralle, D., 2010. Novel roles of U1 snRNP in alternative splicing regulation. *RNA Biol.* 7, 412–419.
- Burge, C., Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 268, 78–94.
- Carninci, P., Kasukawa, T., Katayama, S., *et al.*, 2005. The transcriptional landscape of the mammalian genome. *Science* 309 (5740), 1559–1563.
- Chen, X., Yan, C.C., Luo, C., *et al.*, 2015. Constructing lncRNA functional similarity network based on lncRNA-disease associations and disease semantic similarity. *Sci. Rep.* 5, 11338.
- Chen, X., Yan, C.C., Zhang, X., *et al.*, 2016. WBSMDA: Within and between score for MiRNA-disease association prediction. *Scientific Reports* 6, 21106.
- Cheng, J., Kapranov, P., Drenkow, J., *et al.*, 2005. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science* 308, 1149–1154. doi:10.1126/science.1108625.
- Collins, L.J., Schönfeld, B., Chen, X.S., 2011. The Epigenetics of Non-Coding RNA. In: Tollefsbol, T. (Ed.), *Handbook of Epigenetics: The New Molecular and Medical Genetics*. Academic, pp. 49–61.
- Cordeiro, A., Navarro, A., Gaya, A., *et al.*, 2016. PiwiRNA-651 as marker of treatment response and survival in classical Hodgkin lymphoma. *Oncotarget* 7, 46002–46013. doi:10.18632/oncotarget.10015.
- Crick, F., 1970. Central dogma of molecular biology. *Nature* 227, 561–563.
- Ding, F., Li, H.H., Zhang, S., *et al.*, 2008. SnoRNA Snord116 (Pwcr1/MBII-85) deletion causes growth deficiency and hyperphagia in mice. *PLOS ONE* 3. doi:10.1371/journal.pone.0001709.
- Djebali, S., Davis, C.A., Merkel, A., *et al.*, 2012. Landscape of transcription in human cells. *Nature* 489, 101.
2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 447, 799–816.
- Faghihi, M.A., Modarresi, F., Khalil, A.M., *et al.*, 2008. Expression of a noncoding RNA is elevated in Alzheimer's disease and drives rapid feed-forward regulation of β -secretase. *Nature Medicine* 14 (7), 723.
- Fan, X.-N., Zhang, S.-W., 2015. lncRNA-MFDL: Identification of human long non-coding RNAs by fusing multiple features and using deep learning. *Mol. Biosyst.* 11, 892–897. doi:10.1039/c4mb00650j.
- Friedländer, M.R., Mackowiak, S.D., Li, N., Chen, W., Rajewsky, N., 2011. miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. *Nucleic Acids Res.* 40, 37–52.
- Galindo, M.I., Pueyo, J.I., Fouix, S., Bishop, S.A., Couso, J.P., 2007. Peptides encoded by short ORFs control development and define a new eukaryotic gene family. *PLOS Biol.* 5, e106.
- Garber, M., Grabherr, M.G., Guttman, M., Trapnell, C., 2011. Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat. Methods* 8, 469–477.
- Gomes, A.Q., Nolasco, S., Soares, H., 2013a. Non-coding RNAs: Multi-tasking molecules in the cell. *Int. J. Mol. Sci.* 14, 16010–16039.
- Gomes, C.P.D.C., Cho, J.-H., Hood, L.E., *et al.*, 2013b. A Review of computational tools in microRNA discovery. *Front. Genet.* 4. doi:10.3389/fgene.2013.00081.
- Gong, Y., Huang, H.-T., Liang, Y., *et al.*, 2017. lncRNA-screen: An interactive platform for computationally screening long non-coding RNAs in large genomics datasets. *BMC Genom.* 18. doi:10.1186/s12864-017-3817-0.
- Goujon, M., McWilliam, H., Li, W., *et al.*, 2010. A new bioinformatics analysis tools framework at EMBL – EBI. *Nucleic Acids Res.* 38, W695–W699.
- Gruber, A.R., Findeiss, S., Washietl, S., Hofacker, I.L., Stadler, P.F., 2010. RNAz 2.0: Improved noncoding RNA detection. *Bioinformatics* 2010, 69–79.
- Gulyaeva, L.F., Kushlinskiy, N.E., 2016. Regulatory mechanisms of microRNA expression. *J. Transl. Med.* 14. doi:10.1186/s12967-016-0893-x.
- Gupta, Y., Witte, M., Möller, S., *et al.*, 2014. ptRNAPred: Computational identification and classification of post-transcriptional RNA. *Nucleic Acids Res.* 42, e167.
- Guttman, M., Amit, I., Garber, M., *et al.*, 2009. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature* 458 (7235), 223.
- Guttman, M., Garber, M., Levin, J.Z., *et al.*, 2010. Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nat. Biotechnol.* 28, 503–510.
- Harrow, J., Denoeud, F., Frankish, A., *et al.*, 2006. GENCODE: Producing a reference annotation for ENCODE. *Genome Biology* 7 (1), S4.
- Hertel, J., Hofacker, I.L., Stadler, P.F., 2007. SnoReport: Computational identification of snoRNAs with unknown targets. *Bioinformatics* 24 (2), 158–164.
- Hofacker, I.L., 2003. Vienna RNA secondary structure server. *Nucleic Acids Res.* 31, 3429–3431.
- Hu, L., Xu, Z., Hu, B., Lu, Z.J., 2017. COME: A robust coding potential calculation tool for lncRNA identification and characterization based on multiple features. *Nucleic Acids Res.* 45, e2. doi:10.1093/nar/gkw798.
- Hwang, H.-W., Mendell, J.T., 2006. MicroRNAs in cell proliferation, cell death, and tumorigenesis. *Br. J. Cancer* 94, 776–780. doi:10.1038/sj.bjc.6603023.
- Ideue, T., Adachi, S., Naganuma, T., *et al.*, 2012. U7 small nuclear ribonucleoprotein represses histone gene transcription in cell cycle-arrested cells. *Proc. Natl. Acad. Sci. USA* 109, 5693–5698. doi:10.1073/pnas.1200523109.
- Iseli, C., Jongeneel, C.V., Bucher, P., 1999. ESTScan: A program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. *ISMB*. 138–148.
- Jiang, P., Wu, H., Wang, W., *et al.*, 2007. MiPred: Classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res.* 35, W339–W344.
- Johnson, M., Zaretskaya, I., Raytselis, Y., *et al.*, 2008. NCBI BLAST: A better web interface. *Nucleic Acids Res.* 36, W5–W9.
- Jorjani, H., Kehr, S., Jedlinski, D.J., *et al.*, 2016. An updated human snoRNAome. *Nucleic Acids Res.* 44, 5068–5082.
- Kadri, S., Hinman, V., Benos, P.V., 2009. HHMMiR: Efficient de novo prediction of microRNAs using hierarchical hidden Markov models. *BMC Bioinform.* 10 (Suppl 1), S35. doi:10.1186/1471-2105-10-S1-S35.
- Kang, Y.-J., Yang, D.-C., Kong, L., *et al.*, 2017. CPC2: A fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic Acids Res.* 45 (W1), W12–W16. doi:10.1093/nar/gkx428.
- Kent, W.J., 2002. BLAT – The BLAST-like alignment tool. *Genome Res.* 12, 656–664.

- Khalil, A.M., Guttman, M., Huarte, M., *et al.*, 2009. Many human large intergenic noncoding RNAs associate with chromatin-modifying complexes and affect gene expression. *Proceedings of the National Academy of Sciences* 106 (28), 11667–11672.
- Klein, R.J., Eddy, S.R., 2003. RSEARCH: Finding homologs of single structured RNA sequences. *BMC Bioinform.* 4, 44.
- Kohne, D.E., 1970. Evolution of higher-organism DNA. *Q. Rev. Biophys.* 3, 327–375.
- Kong, L., Zhang, Y., Ye, Z.-Q., *et al.*, 2007. CPC: Assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res.* 35, W345–W349. doi:10.1093/nar/gkm391.
- Lai, E.C., Tomancak, P., Williams, R.W., Rubin, G.M., 2003. Computational identification of Drosophila microRNA genes. *Genome Biol.* 4, R42. doi:10.1186/gb-2003-4-7-r42.
- Lan, W., Li, M., Zhao, K., *et al.*, 2016. LDAP: A web server for lncRNA-disease association prediction. *Bioinformatics* 33, 458–460.
- Lasda, E.L., Blumenthal, T., 2011. Trans-splicing. *Wiley Interdiscip. Rev. RNA* 2, 417–434.
- Laslett, D., Canback, B., 2004. ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Res.* 32, 11–16. doi:10.1093/nar/gkh152.
- Lau, N.C., Seto, A.G., Kim, J., *et al.*, 2006. Characterization of the piRNA complex from rat testes. *Science* 313, 363–367.
- Li, A., Zhang, J., Zhou, Z., 2014. PLEK: A tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. *BMC Bioinform.* 15, 311. doi:10.1186/1471-2105-15-311.
- Lim, L.P., Lau, N.C., Weinstein, E.G., *et al.*, 2003. The microRNAs of *Caenorhabditis elegans*. *Genes Dev.* 17, 991–1008. doi:10.1101/gad.1074403.
- Lindgreen, S., Gardner, P.P., Krogh, A., 2007. MASTR: Multiple alignment and structure prediction of non-coding RNAs using simulated annealing. *Bioinformatics* 23, 3304–3311.
- Lin, M.F., Jungreis, I., Kellis, M., 2011. PhyloCSF: A comparative genomics method to distinguish protein coding and non-coding regions. *Bioinforma. Oxf. Engl.* 27, i275–i282. doi:10.1093/bioinformatics/btr209.
- Lowe, T.M., Eddy, S.R., 1997. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.* 25, 955–964.
- Lund, S.H., Gudbjartsson, D.F., Rafnar, T., *et al.*, 2014. A method for detecting long non-coding RNAs with tiled RNA expression microarrays. *PLOS ONE* 9, e99899. doi:10.1371/journal.pone.0099899.
- Luo, L., Li, D., Zhang, W., *et al.*, 2016. Accurate prediction of transposon-derived piRNAs by integrating various sequential and physicochemical features. *PLOS ONE* 11. doi:10.1371/journal.pone.0153268.
- MacFarlane, L.-A., Murphy, P.R., 2010. MicroRNA: Biogenesis, function and role in cancer. *Curr. Genom.* 11, 537–561. doi:10.2174/138920210793175895.
- Machnicka, M.A., Dunin-Horkawicz, S., de Crécy-Lagard, V., Bujnicki, J.M., 2016. tRNAmodyn: A computational method for predicting posttranscriptional modifications in tRNAs. *Methods* 107, 34–41.
- Mahmoudi, E., Cairns, M.J., 2016. MiR-137: An important player in neural development and neoplastic transformation. *Mol. Psychiatry* 22, 44–55. doi:10.1038/mp.2016.150.
- Ma, L., Bajic, V.B., Zhang, Z., 2013. On the classification of long non-coding RNAs. *RNA Biol.* 10, 924–933. doi:10.4161/rna.24604.
- Maniatis, T., Reed, R., 1987. The role of small nuclear ribonucleoprotein particles in pre-mRNA splicing. *Nature* 325, 673–678. doi:10.1038/325673a0.
- Mathews, D.H., Disney, M.D., Childs, J.L., *et al.*, 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 101 (19), 7287–7292.
- Mattick, J.S., Makunin, I.V., 2006. Non-coding RNA. *Hum. Mol. Genet.* 15, R17–R29.
- Millevoi, S., Vagner, S., 2009. Molecular mechanisms of eukaryotic pre-mRNA 3' end processing regulation. *Nucleic Acids Res.* 38, 2757–2774. doi:10.1093/nar/gkp1176.
- Moazed, D., 2009. Small RNAs in transcriptional gene silencing and genome defence. *Nature* 457, 413–420. doi:10.1038/nature07756.
- Morris, K.V., 2012. Non-Coding RNAs and Epigenetic Regulation of Gene Expression: Drivers of Natural Selection. Norfolk, UK: Caister Academic Press.
- Nakatani, J., Tamada, K., Hatanaka, F., *et al.*, 2009. Abnormal behavior in a chromosome-engineered mouse model for human 15q11-13 duplication seen in autism. *Cell* 137, 1235–1246. doi:10.1016/j.cell.2009.04.024.
- Nam, J.-W., Kim, J., Kim, S.-K., Zhang, B.-T., 2006. ProMiR II: A web server for the probabilistic prediction of clustered, nonclustered, conserved and nonconserved microRNAs. *Nucleic Acids Res.* 34, W455–W458. doi:10.1093/nar/gkl321.
- Nam, J.-W., Shin, K.-R., Han, J., *et al.*, 2005. Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Res.* 33, 3570–3581. doi:10.1093/nar/gki668.
- Nawrocki, E.P., Kolbe, D.L., Eddy, S.R., 2009. Infernal 1.0: Inference of RNA alignments. *Bioinformatics* 25, 1335–1337.
- Negembor, M.V., Jothi, M., Gabellini, D., 2014. Long noncoding RNAs, emerging players in muscle differentiation and disease. *Skeletal Muscle* 4 (1), 8.
- Ng, K.W., Anderson, C., Marshall, E.A., *et al.*, 2016. Piwi-interacting RNAs in cancer: Emerging functions and clinical utility. *Mol. Cancer* 15. doi:10.1186/s12943-016-0491-9.
- Nie, M., Deng, Z.-L., Liu, J., *et al.*, 2015. Noncoding RNAs, emerging regulators of skeletal muscle development and diseases. *BioMed Res. Int.* 2015, e676575. doi:10.1155/2015/676575.
- Orrom, U.A., Derrien, T., Beringer, M., *et al.*, 2010. Long noncoding RNAs with enhancer-like function in human cells. *Cell* 143 (1), 46–58.
- Palazzo, A.F., Lee, E.S., 2015. Non-coding RNA: What is functional and what is junk? *Front. Genet.* 5, 1–11. doi:10.3389/fgene.2015.00002.
- Bhattacharya, D.P., Canzler, S., Kehr, S., *et al.*, 2016. Phylogenetic distribution of plant snoRNA families. *BMC Genomics* 17, 969. doi:10.1186/s12864-016-3301-2.
- Pauli, A., Valen, E., Lin, M.F., *et al.*, 2012. Systematic identification of long noncoding RNAs expressed during zebrafish embryogenesis. *Genome Res.* 22, 577–591.
- Peng, X., Sun, K., Zhou, J., Sun, H., Wang, H., 2017. Bioinformatics for novel long intergenic noncoding RNA (lincRNA) identification in skeletal muscle cells. In: *Proceedings of the Muscle Stem Cells, Methods in Molecular Biology*, pp.355–362. New York, NY: Humana Press.
- Peschansky, V.J., Wahlestedt, C., 2014. Non-coding RNAs as direct and indirect modulators of epigenetic regulation. *Epigenetics* 9, 3–12. doi:10.4161/epi.27473.
- Ponjavic, J., Ponting, C.P., Lunter, G., 2007. Functionality or transcriptional noise? Evidence for selection within long noncoding RNAs. *Genome Research* 17 (5), 556–565.
- Ponting, C.P., Oliver, P.L., Reik, W., 2009. Evolution and functions of long noncoding RNAs. *Cell* 136, 629–641.
- Salamov, A., Solovyev, V., 1998. Fgenesh multiple gene prediction program.
- Santosh, B., Varshney, A., Yadava, P.K., 2015. Non-coding RNAs: Biological functions and applications. *Cell Biochem. Funct.* 33, 14–22. doi:10.1002/cbf.3079.
- Sarkar, F.H., Li, Y., Wang, Z., Kong, D., Ali, S., 2010. Implication of microRNAs in drug resistance for designing novel cancer therapy. *Drug Resist. Update* 13, 57–66. doi:10.1016/j.drug.2010.02.001.
- Schattnner, P., Brooks, A.N., Lowe, T.M., 2005. The tRNAscan-SE, snoscan and snoGPS web servers for the detection of tRNAs and snoRNAs. *Nucleic Acids Research* 33 (suppl_2), W686–W689.
- Sharp, S.J., Schaack, J., Cooley, L., Burke, D.J., Söhl, D., 1985. Structure and transcription of eukaryotic tRNA genes. *CRC Crit. Rev. Biochem.* 19, 107–144. doi:10.3109/10409238509082541.
- Shen, J., Ambrosone, C.B., Zhao, H., 2009. Novel genetic variants in microRNA genes and familial breast cancer. *Int. J. Cancer* 124, 1178–1182. doi:10.1002/ijc.24008.
- Skrayabin, B.V., Gubar, L.V., Seeger, B., *et al.*, 2007. Deletion of the MBII-85 snoRNA gene cluster in mice results in postnatal growth retardation. *PLOS Genet.* 3, 2529–2539. doi:10.1371/journal.pgen.0030235.
- Storz, G., 2002. An expanding universe of noncoding RNAs. *Science* 296, 1260–1263. doi:10.1126/science.1072249.
- Storz, G., Opdyke, J.A., Zhang, A., 2004. Controlling mRNA stability and translation with small, noncoding RNAs. *Curr. Opin. Microbiol.* 7, 140–144. doi:10.1016/j.mib.2004.02.015.
- Sun, L., Luo, H., Bu, D., *et al.*, 2013. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic Acids Res.* 41, e166. doi:10.1093/nar/gkt646.
- Tera, G., Komori, T., Asai, K., Kin, T., 2007. miRRim: A novel system to find conserved miRNAs with high sensitivity and specificity. *RNA N.Y.N* 13, 2081–2090. doi:10.1261/rna.655107.

- Tomkins, G.M., Gelehrter, T.D., Granner, D., *et al.*, 1969. Control of specific gene expression in higher organisms. *Science* 166, 1474–1480.
- Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28, 511–515. doi:10.1038/nbt.1621.
- Tupy, J.L., Bailey, A.M., Dailey, G., *et al.*, 2005. Identification of putative noncoding polyadenylated transcripts in *Drosophila melanogaster*. *Proc. Natl. Acad. Sci. USA* 102, 5495–5500.
- Ulitsky, I., 2016. Evolution to the rescue: Using comparative genomics to understand long non-coding RNAs. *Nat. Rev. Genet.* 17, 601–614. doi:10.1038/nrg.2016.85.
- Wang, C., Wei, L., Guo, M., Zou, Q., 2013a. Computational approaches in detecting non-coding RNA. *Curr. Genom.* 14, 371–377. doi:10.2174/13892029113149990005.
- Wang, K., Liang, C., Liu, J., *et al.*, 2014. Prediction of piRNAs using transposon interaction and a support vector machine. *BMC Bioinform.* 15, 419. doi:10.1186/s12859-014-0419-6.
- Wang, L., Park, H.J., Dasari, S., *et al.*, 2013b. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. *Nucleic Acids Res.* 41, e74. doi:10.1093/nar/gkt006.
- Washietl, S., Hofacker, I.L., Stadler, P.F., 2005. Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA* 102, 2454–2459. doi:10.1073/pnas.0409169102.
- Washietl, S., Pedersen, J.S., Korbelt, J.O., *et al.*, 2007. Structured RNAs in the ENCODE selected regions of the human genome. *Genome Res.* 17, 852–864. doi:10.1101/gr.5650707.
- Westhof, E., Auffinger, P., 2001. Transfer RNA Structure. *eLS.* 1–11. doi:10.1002/9780470015902.a0000527.pub2.
- Wucher, V., Legeai, F., Hédan, B., *et al.*, 2017. FEELnc: A tool for long non-coding RNA annotation and its application to the dog transcriptome. *Nucleic Acids Res.* 45, e57. doi:10.1093/nar/gkw1306.
- Xia, W., Zhou, J., Luo, H., *et al.*, 2017. MicroRNA-32 promotes cell proliferation, migration and suppresses apoptosis in breast cancer cells by targeting FBXW7. *Cancer Cell Int.* 17. doi:10.1186/s12935-017-0383-0.
- Xu, Y., Mural, R.J., Einstein, J.R., Shah, M.B., Uberbacher, E.C., 1996. GRAIL: A multi-agent neural network system for gene identification. *Proc. IEEE* 84, 1544–1552.
- Yang, J.H., Zhang, X.C., Huang, Z.P., *et al.*, 2006. snoSeeker: An advanced computational package for screening of guide and orphan snoRNA genes in the human genome. *Nucleic Acids Res.* 34, 5112–5123. doi:10.1093/nar/gkl672.
- Yazaki, J., Gregory, B.D., Ecker, J.R., 2007. Mapping the genome landscape using tiling array technology. *Curr. Opin. Plant Biol.* 10, 534–542. doi:10.1016/j.pbi.2007.07.006.
- Zhang, B., Wang, Q., Pan, X., 2007. MicroRNAs and their regulatory roles in animals and plants. *J. Cell. Physiol.* 210, 279–289.
- Zhang, Y., Wang, X., Kang, L., 2011. A k-mer scheme to predict piRNAs and characterize locust piRNAs. *Bioinform. Oxf. Engl.* 27, 771–776. doi:10.1093/bioinformatics/btr016.
- Zuker, M., 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31, 3406–3415.
- Zuo, L., Wang, Z., Tan, Y., Chen, X., Luo, X., 2016. piRNAs and their functions in the brain. *Int. J. Hum. Genet.* 16, 53–60.

Vaccine Target Discovery

Li C Chong, Universiti Putra Malaysia, Serdang, Malaysia

Asif M Khan, Perdana University, Selangor, Malaysia and John Hopkins University, Baltimore, MD, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Vaccination is one of the most efficacious medical interventions that have decreased human morbidity and mortality in all regions of the world. It not only has dramatically reduced the incidence of numerous diseases (Hilleman, 1985; Andre *et al.*, 2008), such as measles, diphtheria, mumps, rubella, tetanus, yellow fever, pertussis, and poliomyelitis, but also eradicated the dreaded smallpox viral infection (Strassburg, 1980). Efforts are currently also focused on developing vaccines for treatment of other diseases (Ada, 2003). This triumph over infectious diseases has been achieved by using either killed or attenuated conventional vaccines. A conventional or prophylactic vaccination is based on deliberate exposure to non-virulent form of the pathogen to establish immunity from subsequent exposure to the virulent form of the pathogen (Lee *et al.*, 2012). This approach requires little knowledge of the molecular nature of the individual pathogen antigens or the immune responses they elicit. To date, this approach has undoubtedly been the most successful.

There is, however, an ongoing trend towards emerging infectious diseases (Fauci, 2001; Fauci *et al.*, 2005) against humankind in different parts of the world. These diseases are caused by the emergence of new pathogens, resurgence of old ones, constantly mutating pathogens, drug-resistant pathogens, and even include pathogens used as agents of bioterrorism. Previously undescribed pathogens, such as severe acute respiratory syndrome (SARS), bird flu (avian influenza), mad cow, and HIV/AIDS, are appearing at an increasing frequency. On the other hand, old diseases such as ebola, hanta, dengue and cholera, among others, are invading populations from which they have disappeared or cross species barrier to invade new species. Occasionally, pathogens mutate (or recombine) and then they adapt and show up as new variants that evade the host's acquired immunity. The annual influenza epidemics are generally due to genetically drifting strains of influenza that differ slightly from previous strains (Bouvier and Palese, 2008). The development of drug-resistance pathogens, such as malaria, pneumococci, enterococci, and tuberculosis, has increased over the years, partly due to the widespread and inappropriate use of drugs (Knobler *et al.*, 2003). Moreover, the pathogens responsible for the emerging infectious diseases are also of potential use as bioterrorist weapon (Ryan, 2008). For these reasons, emerging infectious diseases continue to pose threats to public health (Morens and Fauci, 2013). Therefore, there is a need for more effective vaccines to help reduce human morbidity and mortality from emerging infectious diseases.

The last decade has seen significant advances in new technologies for the development of new vaccines. These technologies, combined with our understanding of host response to foreign antigens, have laid the foundation for rapid advances in vaccinology. Few potential candidate approaches for this new family of vaccines (Arnon and Ben-yedidia, 2003; Minichiello, 2002; Babiuk, 1999; Nandy and Basak, 2016) are subunit vaccines, genetically engineered live vaccines, and polynucleotide (DNA or genetic) vaccines. All these new directions in vaccine development have something in common; their most important challenge is the discovery of key antigens from the array of proteins encoded by the pathogen genome that are able to elicit protective immune response against these pathogens and are effective for majority of the human population. The presence of genetic variation in the genes of the host immune system across the human population and the genomes of the pathogen variants make this a multi-dimensional and a combinatorial problem.

Since 1980s, the focus of vaccine design has been on the pathogen's variable domains, mutants, multivalent coverage and not so much on conserved regions (Plotkin, 2005). However, a large body of data is building up in the field of vaccine design and epitope prediction that points to the neglected aspect of conserved epitopes as important targets of vaccine development. In fact, evidence is accumulating that while many variable regions are highly antigenic, sufficiently large number of them are actually non-immunoprotective and have been exploited by viruses and other pathogens for immune escape and may lead to immunopathology (Haydon and Woolhouse, 1998). Conserved epitopes were thought to be unimportant due to their lack of immunogenicity (Bona *et al.*, 1998; Li *et al.*, 2011). However, the immunogenicity of such conserved epitopes especially those which are immune-protective and conserved amongst many variants and mutant strains, can be boosted using adjuvants, which are chemicals or approved drug molecules. Moreover, conserved epitopes also help address the issues of ethnicity ("pan-haplotype responsive") compared to variable epitopes (Khan *et al.*, 2008). In addition, the conserved epitopes can circumvent vaccine design issues of escape mutants and the need of using the latest strains. In particular, if a set of conserved, immunogenic, and immunoprotective epitopes are suitably boosted, they will not elicit mutations in the pathogen, and thus can be reasonably predicted to remain unchanged in the next season of infection. The possibility of extended efficacy of vaccines would be a tremendous advance for the vaccine industry and will potentially serve the global population with protection against infections (Sylvester-Hvid *et al.*, 2002; De Groot *et al.*, 2004; Sette *et al.*, 2001; Raman *et al.*, 2014; Khan *et al.*, 2017).

Vaccine informatics, a fledgling sub-field of reverse vaccinology, has the potential to develop effective vaccines (Hegde *et al.*, 2017; Khan *et al.*, 2017). With the rapid expansion of vaccine related data (host and pathogen) stemming from both classical and high-throughput genomic/proteomic approaches, identifying conserved, robust, immunogenic, and immunoprotective epitopes manually from this large data pool is inefficient. Vaccine informatics is a practical science for designing new vaccines with a focus

on bioinformatics-driven acquisition, manipulation and analysis of data related to the immune system and disease agents (Raman *et al.*, 2014). It provides a means for systematic study of big data, pre-screening of targets, and facilitates experimental design for validation by a small number of key experiments. The bioinformatics support can be divided into two, the standard bioinformatics support and the more specialised immunoinformatics support (Petrovsky *et al.*, 2003). The standard support includes basic bioinformatics functions, such as sequence comparison and alignment, database searching, hunting for patterns and profiles, 3D-structure analysis and modeling, and data annotation (reviewed in Brusic and Petrovsky (2003)). Immunoinformatics is a more targeted bioinformatics support with an emphasis on data-warehousing and mining of immunological data, such as prediction of immunogenicity (Soria-Guerra *et al.*, 2015; Brusic and Petrovsky, 2003). Vaccine researchers are taking advantage of these bioinformatics approaches, in combination with experimental validation, to discover and facilitate better understanding of the components of the human immunome, which then aid in the design of new vaccines. The immunome can be defined as the complete set of genes and proteins of the immune system. Highly accurate target predictions can diminish discovery cost by 10–20 folds (De Groot *et al.*, 2002; Kast *et al.*, 1994).

Background

In the human and higher vertebrate host, the major functions of the immune system are the maintenance of homeostasis, surveillance and tolerance to self-structures, and defence followed by immunity against pathogens (Yatim and Lakkis, 2015). The immune system is widely distributed in the body and comprises of immune organs, tissues, and cells, connected as a complex, but tightly regulated network (Jerne, 1993; NIH, 2003; Nicholson, 2016). In general, the processes that take place at the molecular level and cellular level largely initiate and regulate the function of the immune system. A healthy immune system will discriminate 'non-self' or foreign antigenic proteins from those that are normally present ('self') in an organism and will raise appropriate responses. The number of self-structures is large, but finite; the number of non-self structures is practically infinite. In the cell, both self and non-self proteins are digested by the proteasomes into short peptide fragments, which are then bound by major histocompatibility complex (MHC) molecules to form peptide/MHC complexes and are displayed on the surface of host cells (Vigneron and Van den Eynde, 2014). These peptides are recognition labels, which display the contents of host cells to T cells of the immune system. The presence of non-self peptides is a prerequisite for the initiation of immune responses. Peptides produced by degradation of intracellular proteins bind MHC class I molecules and are recognised by CD8⁺ T cell receptor (Shastri *et al.*, 2002; Chowell *et al.*, 2015; Blum *et al.*, 2013). MHC class II molecules present peptides, produced by degradation of proteins of extracellular origin, on the surface of antigen-presenting cells to CD4⁺ T cells (reviewed in Lennon-Duménil *et al.* (2002)). A major function of CD8⁺ T cells is to recognize and destroy cells infected by pathogens (Nicholson, 2016). Peptides displayed by the MHC class II molecules mainly serve to regulate immune responses; they are crucial for the initiation, enhancement and suppression of immune responses.

Driven by methods of molecular and cell biology, significant advances in the understanding of immunological processes have been made during the last two decades. This progress over the years resulted in the continuous accumulation of huge amount of immunological data obtained experimentally. This growing number of immunological data and the high complexity of the functional and structural foundation of the immune processes created a need for improved data management to enable advance data analysis. The need to manage and analyse this growing amount of complex data has led to the development of a number of immunological databases (Brusic *et al.*, 2000), such as SYFPEITHI and MCHPEP, IMGT and FIMM, and complex computational models (Petrovsky and Brusic, 2002). The purpose of immunological databases is to facilitate the collection of, access to, and use of immunologically relevant data. One of the applications of the databases in immunology is for vaccine research and development by using complex computational models in combination with experimental approaches to help precise our understanding of antigen presentation and recognition by the immune system.

The combined effort between experimental and computational immunology provided us with a new perspective to designing vaccines that will be effective across demographic boundaries. Previously, the extreme degree of polymorphism observed in the MHC posed limitations for the development of such a vaccine because the ability to trigger an effective T-cell response is partly determined by the MHC phenotype of the individual and different individuals have different MHC allele (Macdonald *et al.*, 2001; Marrack *et al.*, 2017). The MHC genes are the most polymorphic of all human genes, with more than 10,000 alleles known (as of Feb 2018), and are important in increasing the range of responses that different individuals can mount. In humans, this polymorphism results from concentrated amino acid substitutions in the peptide-binding groove of human leukocyte antigen (HLA, the human MHC system) molecules that produce variability in peptide binding and presentation to T cells (MacDonald *et al.*, 2000). Over the years, few groups have investigated the possibility of a functional classification of HLA polymorphism based on peptide-binding specificities. It was found that majority of HLA alleles (both class I and II) could be grouped into 18 or more different 'supertypes', purely on the basis of similarities in their peptide binding specificity (Lund *et al.*, 2004). It has been suggested that the majority of all major human populations can be covered by only few HLA supertypes, where the different members of each supertype bind similar peptides ('promiscuous peptides') for presentation to T-cell (Sette *et al.*, 1999). Further, latest developments show evidence for presence of "immunological hot-spots" (Srinivasan *et al.*, 2004) in antigens. Immunological hot-spots are defined as antigenic regions possessing multiple promiscuous peptides that are supertype specific. This area of research raises the prospect of identifying key pathogenic antigens that possess immunological hot-spots as best candidates for vaccine design as it will provide protection at the population level, irrespective of ethnicity.

The humoral response involves antibodies produced by B-cells, which recognize both linear and conformational B-cell epitopes on the surface of the pathogen. Conformational neutralizing epitopes are the primary focus of various vaccine research for protective humoral responses. However, unlike linear B-cell epitopes and T-cell epitopes, reliable computational tools for prediction of conformational epitopes are limited (Kulkarni-Kale *et al.*, 2005; Zhang *et al.*, 2011).

As for the pathogens, the last decade witnessed the rapid expansion of sequence data at our disposal, stemming from both genomic and proteomic approaches, enabling analysis to map key antigens that are potential targets for protective immune responses. The genomic sequences of a large number of pathogens that threaten public health have been completed or are impending completion. For example, the whole genome of over 7,475 viruses have been sequenced and deposited in the major public database Entrez Genome (see "Relevant Website section"). The data derived from the genome/proteome sequencing and related projects for a particular species of pathogen, such as West Nile Virus, is the missing gap towards the development of vaccines effective against the majority of the variants currently known within that pathogenic species and probably against novel ones that are yet to emerge (Koo *et al.*, 2009). Such vaccines will be much superior to the current generation of vaccines, which are solely based on a single or few antigens providing protection only against certain variants of a pathogen and therefore might elicit too narrow a breadth of response to provide protection from the remaining diverse variants of the same pathogen species (Doolan, 2003).

We have made significant progress in understanding the processes in the host that are involved in mounting an immune response. However, we are still far from having a good understanding of the natural complexity of the pathogens. A good starting point would be by utilizing the pathogens genomic/proteomic data to study their sequence diversity, and identify antigens containing conserved and variable immunological hot-spots. To fully realize the promise of the available datasets for such study, we would require the development of appropriate technologies for systematically converting genomic/proteomic data into protective vaccines. Recently, systematic genome-wide approach to identify the key antigens of a pathogenic species from the numerous variant sequences of the same pathogen have been reported (Dhanda *et al.*, 2017; Rizwan *et al.*, 2017; Goodswen *et al.*, 2014; Vivona *et al.*, 2006; Del Tordello *et al.*, 2016; Maria *et al.*, 2017; Doolan, 2003; Raman *et al.*, 2014; Koo *et al.*, 2009; Khan *et al.*, 2017). The selected key antigens will represent the minimal representative sets of target sequences required to provide immunity against the majority of the existing variants of a particular pathogen species.

Bioinformatics is an inter-disciplinary field that is essential for the analysis and interpretation of complex and large quantity of biological data generated by functional studies and high throughput technologies. It is used to propose the next sets of experiments and, most importantly, to derive better understanding of biological processes. As stated, the number of pathogen sequence data in public databases is increasing rapidly, however, experimental approaches to study this large data pool for the development of immune interventions are time-consuming, costly and almost impractical. Through combination of bioinformatics and experimental approaches, it is possible to select key experiments and help optimize experimental design. Computer algorithms are increasingly used to speed-up the process of knowledge discovery by helping to identify critical experiments for testing hypothesis built upon the result of computational screening. A number of successful examples for application of computer models to study immunological problems have been described in Brusci *et al.* (2005). Such examples illustrate the power of computational approach to complex problems involving potentially vast datasets with potential biases, errors and discrepancies.

A Computational Framework for Vaccine Target Discovery

Reverse vaccinology, a bottom-up genomic approach, has been successfully applied to the development of vaccines against pathogens that were previously not suited to such development (Vernikos, 2008; Rappuoli and Covacci, 2003; Rappuoli, 2001; Del Tordello *et al.*, 2016). The pre-requisite for this approach is the sequence data of the target pathogen, which acts as input to various bioinformatics algorithms for prediction of putative antigens that are likely to be successful vaccine targets. These candidates can then be validated by a small number of key experiments in the lab. The approach has been successfully applied to the development of universal vaccines against group B Streptococcus (Maione *et al.*, 2005) and vaccine candidates against MenB (Pizza *et al.*, 2000), among others (Rappuoli and Covacci, 2003). Reverse vaccinology is a promising method for the high-throughput discovery of candidate vaccine targets that have the potential to mirror the dynamics and antigenic diversity of the target pathogen population, which includes the diversity of the interacting partner, the immune system. However, a big challenge to this end is the need to understand how vaccine developers can cover antigenic diversity and develop a systematic approach to rationally screen pathogen data to select candidate vaccine targets that cover the diversity.

Over the years, a number of bioinformatics pipelines have been designed that predict vaccine candidates both rapidly and efficiently. VacSol (Rizwan *et al.*, 2017), NERVE (Vivona *et al.*, 2006), and Vacceed (Goodswen *et al.*, 2014) are examples of such pipelines for proteomes of bacterial or eukaryotic pathogens and these pipelines are highly configurable and scalable (Zaharieva *et al.*, 2017). They include multiple steps and integrate various algorithms for analysis and comparison. Shortlisted candidate vaccine targets are ranked for prioritization towards experimental validation. These pipelines are expected to improve the vaccine target discovery process.

The general characteristics desired for a candidate vaccine target are (i) highly conserved; (ii) pathogen-specific; (iii) important for structure/function; (iv) immune-relevant; and (v) antigenically similar to circulating strains. Highly conserved targets are less likely to mutate and escape immune recognition. This is particularly so if they are important for structure and function, suggesting a robust historical conservation. High conservation also reduces the possibility of altered-peptide ligand (APL) effect from variant epitopes of the same pathogen species (Sloan-Lancaster and Allen, 1996; Evavold *et al.*, 1993; Rothman, 2004). Variants may also

originate from other pathogens, in particular those that co-circulate or co-infect with the pathogen of interest and if they belong to the same family. In vaccine design, epitopes common to other pathogens could either be useful by inducing cross-protection, or detrimental by inducing altered-ligand effect. Thus, potential vaccine targets should be analyzed for specificity to the target pathogen. The definition of virus species-specific vaccine targets can be further expanded to exclude those with one amino acid mismatch to human sequences in order to avoid possibility of molecular mimicry. Antigenic mismatch between a vaccine and circulating strains has been shown to increase the risk of disease outbreak by 1000-fold compared to immunization using identical strains (Park *et al.*, 2004). Thus, it is important to assess the extent of antigenic identity between the candidate vaccine targets and the circulating strains.

Typically, a generic semi-automated computational framework comprises of three key components: data collection, data processing, and data analysis (Khan, 2005, 2009; Khan *et al.*, 2006, 2017). Fig. 1 illustrates the workflow of the framework and Fig. 2 provides a non-comprehensive list of commonly used tools. Data collection would involve the user providing a set of sequences (aligned or unaligned) of the pathogen of interest for analysis. The sequences could be a single protein dataset, multiple or the complete proteome. Additionally, comparative analyses of the sequences can be performed between subtypes/groups of the pathogen. The sequences can be retrieved from public repositories, primary or specialist databases, such as the NCBI Entrez Protein database (NCBI Resource Coordinators, 2017) or Influenza Research Database (IRD) (Zhang *et al.*, 2017), respectively. User can also provide sequences derived from their experimental work, not available in public databases. Typically, downloaded data would comprise full-length or partial sequences, and the corresponding metadata. Data processing would involve removal of duplicate sequences from the dataset, and for comparative analysis, the merging of the input sequences will be required prior to the alignment step. Multiple sequence alignment will be carried out using an existing tool that is robust in dealing with large sequence data, such as (Sievers *et al.*, 2011; Katoh *et al.*, 2017; Edgar, 2004; Do *et al.*, 2005) Clustal Omega, MAFFT, MUSCLE or PROBCONS. The output alignment quality will be manually inspected for any errors and/or misalignments, which are common when dealing with partial sequences. Henceforth, the data would be ready for analyses, which can involve performing a diversity analysis, such as by measuring entropy values (Heiny *et al.*, 2007; Hu *et al.*, 2013; Khan *et al.*, 2008; Koo *et al.*, 2009) and quantifying variant motifs for each, user-defined, k-mer positions in the alignment. The results of these analyses will be plotted as an output for the user, providing a holistic view of the diversity, including variant distribution. The user can define a preferred conservation threshold for selection of highly conserved sequences. These selected sequences are then analyzed for distribution of variants in nature (Khan *et al.*, 2008; Koo *et al.*, 2009), including matches to human proteins, enabling the identification of pathogen specific, highly conserved sequences. The robust nature of the historical conservation can be assessed by performing functional and structural analysis (Sprenger *et al.*, 2008; Hung and Link, 2011; Wizemann *et al.*, 1999; Sachdeva *et al.*, 2005; Monterrubio-López *et al.*, 2015; He, 2014; Tang *et al.*, 2014; Maria *et al.*, 2017). The relevance of the identified potential candidate vaccine targets for use against current circulating strains can be assessed by measuring the incidence of the candidate targets in the corresponding sequences of recent strains of the virus of interest (Khan *et al.*, 2017). The antigenicity/immunogenicity of the candidate sequences are predicted to assess their immune relevance. Experimental validation includes matching the predicted epitopes with reported epitopes in public databases, such as the Immune Epitope Database (IEDB) or SYFPEITHI, or performing quantitative measurements of the pre-selected candidate peptides by generating synthetic constructs and testing for their immunogenicity, such as by use of HLA transgenic animal models that express specific HLA alleles (Rosloniec *et al.*, 1997; Lefranc *et al.*, 2009; Gourlay *et al.*, 2017; Khan, 2005).

Immunoinformatics

Bioinformatics tools can facilitate the process of epitope mapping by identifying peptides that can potentially elicit T-cell responses. Binding of epitopes to HLA antigens is highly allele-specific; core peptide-binding motifs (usually between 8- and 11-mers, most often 9-mers) have been defined experimentally for a number of HLA class I and class II alleles and incorporated into computational algorithms, allowing to predict candidate HLA-binding epitopes *in silico* from protein sequences (Parker *et al.*, 1994; Rammensee *et al.*, 1999; Sturmiolo *et al.*, 1999; Zhang *et al.*, 2005; Nielsen *et al.*, 2010; Karosiene *et al.*, 2013; Paul *et al.*, 2013, 2015a,b; Andreatta *et al.*, 2015; Andreatta and Nielsen, 2015; Pro *et al.*, 2015; Trolle *et al.*, 2015; Abelin *et al.*, 2017; Jurtz *et al.*, 2017; Fleri *et al.*, 2017). More recently, quantifiable predictive features of TCR $\alpha\beta$ binding to HLA/epitope complexes have been also described (Birnbaum *et al.*, 2014; Dash *et al.*, 2017; Glanville *et al.*, 2017; Gee *et al.*, 2017).

Two main categories of specialized immunoinformatics tools are available for prediction of MHC binding peptides – methods based on identifying patterns in sequences of binding peptides, and those that employ three-dimensional (3D) structures to model peptide/MHC interactions (Tong *et al.*, 2007; Liljeroos *et al.*, 2015; Gourlay *et al.*, 2017). Pattern-based methods includes binding motifs, quantitative matrices, decision trees, artificial neural networks (ANNs), hidden Markov models (HMMs) and support vector machines (SVMs), among others. In contrast, the structure based methods are theoretically rooted and include homology modeling, docking and 3D threading techniques (Dominguez *et al.*, 2003; De Vries *et al.*, 2010; Agostino *et al.*, 2016; Khan and Ranganathan, 2010; Liljeroos *et al.*, 2015; Gourlay *et al.*, 2017). Although less accurate, pattern based approaches are over-represented in the literature due to higher complexity in development and longer computational time of the more accurate structure-based approaches (Ranganathan and Tong, 2007; Maria *et al.*, 2017), including the sheer difference in the availability of linear versus structural data.

For a given sequence, typically all possible overlapping 9-mer (and later, 8- to 11- mer) peptide sequences are extracted. Epitope prediction for HLA class I and class II alleles are performed using benchmarked prediction models, including, for HLA

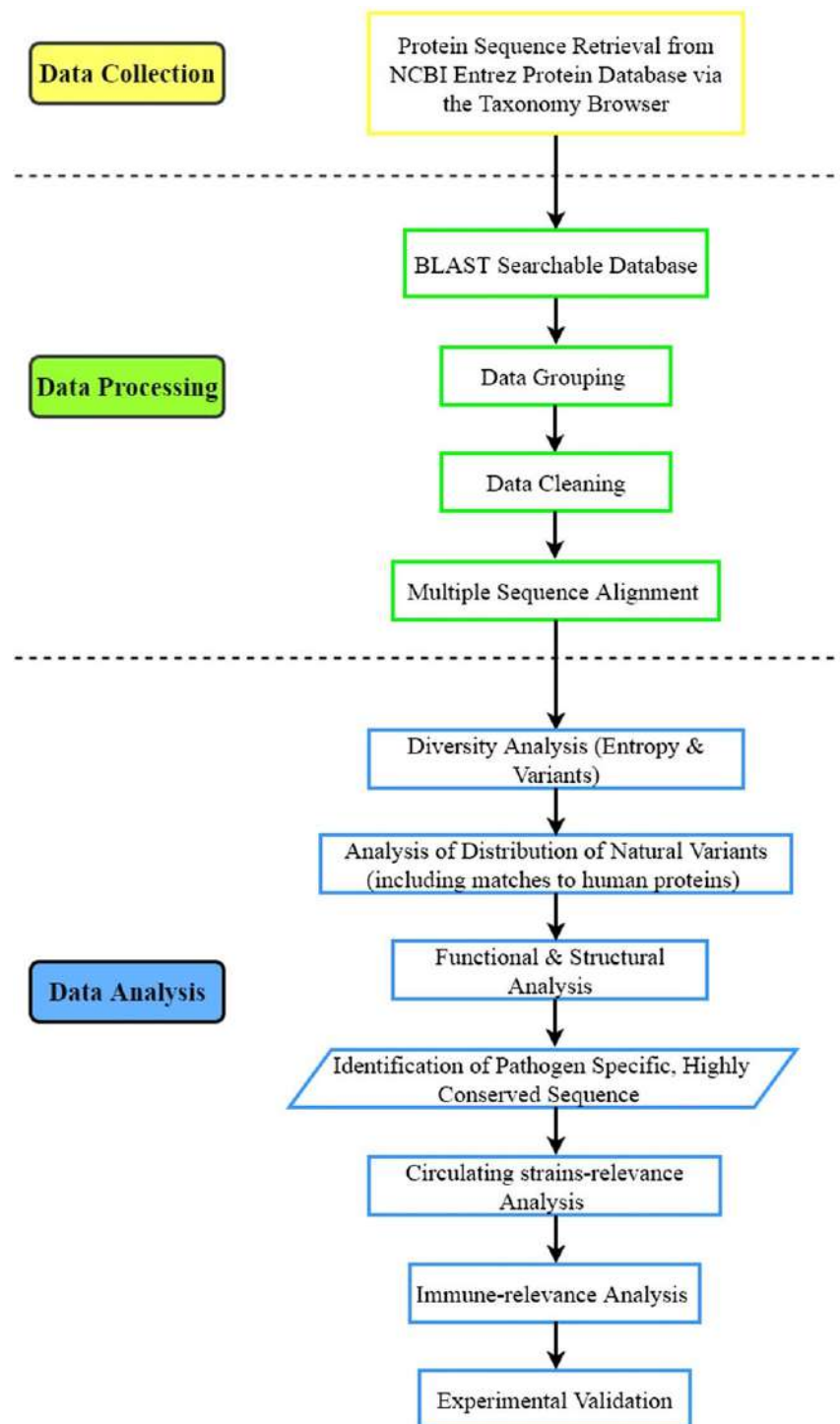


Fig. 1 Computational framework for vaccine target discovery.

class I epitopes, the artificial neural network (ANN)-trained NetMHCpan (version 4.0) (Andreatta *et al.*, 2015; Trolle *et al.*, 2015; Jurtz *et al.*, 2017), and for HLA class II epitopes, NetMHCIIpan (Karosiene *et al.*, 2013; Andreatta and Nielsen, 2015), and the allele-specific consensus percentile ranks of all algorithms queried by the Immune Epitope Database and Analysis Resource (IEDB) tools (combination of NN-align, SMM-align, and ComLib/Sturniolo) (Paul *et al.*, 2015a,b). Additionally, proteasome cleavage (Hakenberg *et al.*, 2003; Nussbaum *et al.*, 2001; Nielsen *et al.*, 2005) and TAP binding predictions (Zhang *et al.*, 2006; Bhasin *et al.*, 2007; Bhasin and Raghava, 2004) will be performed for priority ranking. Several tools, such as NetCTL (Larsen *et al.*, 2005), integrate these various predictions into one and predict for HLA supertypes, which are groups of HLA alleles with similar peptide

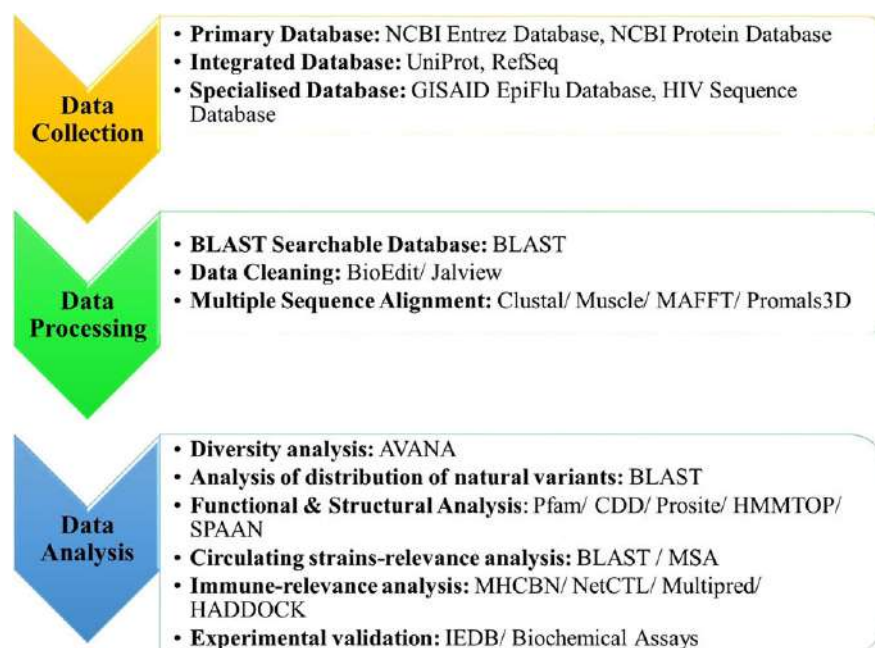


Fig. 2 An example of commonly used tools and databases for vaccine target discovery.

binding specificity. Potentially cross-reactive self epitopes may be searched within the human proteome using BLAST/BLAT alignment tools (Kent, 2002; Altschul *et al.*, 1990).

Case Studies: Vaccine on Infectious Diseases and Non-Infectious Diseases

Infectious Diseases

Reverse vaccinology immunoinformatics approaches are widely applied for viral vaccine design, such as for influenza virus, chikungunya virus, zika virus and others (Gupta *et al.*, 2016; María *et al.*, 2017), including parasites (Damfo *et al.*, 2017) and bacteria (Mistry and Flower, 2017; Zahroh *et al.*, 2016; Rappuoli, 2001). Khan *et al.* developed a bioinformatics pipeline for DENV, which proved generic as it was successfully applied to several viruses, such as WNV (Koo *et al.*, 2009), a close relative of DENV (Khan *et al.*, 2008), and a number of other viruses, such as HIV-1 (Hu *et al.*, 2013), among others. It provides a novel and generalized approach to the formulation of peptide-based vaccines targeting a broad diversity of pathogens and applicable to the human population at large. This methodology is a significant contribution to the field of reverse vaccinology as it enables the systematic screening and analyses of pathogen data which would otherwise be impossible to carry out experimentally, due to too many pathogen sequences (high viral diversity) and variations in immune system among individuals (extensive polymorphism of HLA). This approach therefore significantly reduces the efforts and cost of experimentation, while providing for systematic screening and analyses of pathogen proteomes (Raman *et al.*, 2014).

Khan *et al.* (2008), Koo *et al.* (2009) and Hu *et al.* (2013) analyzed a large number of dengue (DENV), West Nile virus (WNV) and clade B HIV-1, sequences, respectively, retrieved from the NCBI Entrez protein database (Table 1). The sequences were aligned and the overlapping nonamer amino acid positions of the viral proteome, each a possible core binding domain for human leukocyte antigen molecules and T-cell receptors, were quantitatively analyzed. The mean entropy of DENV nonamer sequences was low, with a range of 0.2–1.0 for within and 1.6–2.6 for between serotypes. This was even lower for WNV, ranging from 0.2–0.5, with the highest for HIV-1 clade B subtype, 1.9–4.2. Entropy is a general measure of diversity, and the data provided a holistic overview viral diversity across the proteome. Accordingly, the incidence of variants to the most prevalent, index sequence at the aligned nonamer positions was the lowest for WNV (intra: $\leq 10\%$) and the highest for HIV-1 clade B (intra: $\sim 80\%$ – 99%); the variants incidence within each DENV serotype was comparable to WNV, but between serotypes ($\sim 60\%$ – 80%) was closer to HIV-1 clade B subtype. Forty-four (44) sequences (pan-DENV sequences) identical in 80% or more of all recorded DENV sequences represented 15% of the DENV polyprotein length. The proportion (34%) was much higher for WNV and at complete conservation (100% incidence). Notably, at similar incidence level ($\geq 80\%$ incidence) to DENV, although pan-clade, $\sim 35\%$ of the intra HIV-1 clade B proteome was highly conserved. The proportion of these conserved sequences that were immune-relevant showed an inverse relationship: DENV (59% matched 9aa or more of 45 class I and II reported epitopes), WNV (50% matched 9aa or more of 57 class I and II reported epitopes), and HIV-1 clade B (37% matched 9aa or more of 73 class I and II reported epitopes). Khan *et al.* (2008) highlighted that conservation analysis should go beyond the species of interest, extending to all those

Table 1 Key summary results of vaccine target discovery study on Dengue virus (DENV), West Nile virus (WNV) and clade B HIV-1.

Category	DENV (Khan <i>et al.</i> , 2008)	WNV (Koo <i>et al.</i> , 2009)	HIV-1 clade B (Hu <i>et al.</i> , 2013)
Mean nonamer entropy	Intra: 0.2–1.0 Inter: 1.6–2.6	Intra: 0.23–0.51	Intra: 1.9–4.2
General variants incidence trend	Inter: ~60%–80%	Intra: <10%	Intra: ~80%–99%
% of the proteome covered by conserved sequences	Pan-clade: 15% (>=80% incidence)	Intra: 34% (=100% incidence)	Intra: 35% (>=80% incidence)
Immune relevance	(26/44) 59% matched 9aa or more of 45 class I and II reported epitopes	(44/88) 50% matched 9aa or more of 57 class I and II reported epitopes	(29/78) 37% matched 9aa or more of 73 class I and II reported epitopes
No. of other viral species matched by the conserved sequences	(27/44) 61% matched 64 flaviviruses	(67/88) 76% of the sequences matched 68 flaviviruses	(74/78) 95% matched 9 deltaviruses
No. of pathogen specific conserved sequences	17	21	4

other species that are evolutionarily related as they may act as variants to the conserved epitopes identified. This step is necessary to identify conserved epitope sequences that are pathogen specific, with none or minimal number of variant sequences within or across other pathogen species. Variant epitopes are hypothesized to cause deleterious immune responses. Many of the conserved sequences matched nine consecutive amino acids of many (flaviviruses; family of WNV and DENV) to few (*deltaviruses*; genus of HIV-1) other related viruses, leaving only 17, 21 and 4 pathogen specific conserved sequences for DENV, WNV and HIV-1 clade B subtype, respectively.

Non-Infectious Diseases

According to World Health Organization (WHO), non-infectious diseases, especially chronic diseases will lead the disability by 2020. Hence, diseases such as cancers, obesity, neurodegenerative disease addictions and others have become a recent focus of vaccine development (Barrett, 2016). Cancer is the most common non-infectious disease that leads death world-wide. Due to the advanced technology and success of *in silico* methods in infectious disease vaccine design, computational approaches have been applied in study of cancer vaccine design. For example, VacImm was developed as a bioinformatics approach to simulate peptide vaccination in cancer therapy (von Eichborn *et al.*, 2013). In addition, modeling approaches using computational biology, such as Sim Triplex and MetastaSim model are important to understand the molecular interactions at the cellular and molecular level (Pappalardo *et al.*, 2013; Sankar *et al.*, 2013). Adekiya *et al.* (2017) reports a recent example of a study that included bioinformatics analysis in cancer vaccine development.

Conclusion: Vaccine Informatics and Future Vaccines

Future vaccines will be minimalistic in approach by focusing on key parts of the pathogen, such as regions containing epitopes that cover antigenic diversity and, thus, will target immunologically similar subgroups of the human population and multiple pathogen variants. This is evident from the trend observed in evolution of vaccine strategies, which has seen a shift from whole organisms to recombinant proteins, and further towards the ultimate in minimalist vaccinology, the peptide/epitope/multi-epitope based vaccines. The minimalist approach is also expected to cover the safety concerns that are associated with the traditional vaccine approach of using whole organism (Sette and Fikes, 2003; Dertzbaugh, 1998). Vectored vaccines, suitable for 'combination immunization' that are produced by recombinant DNA technology and contain multivalent minimal antigens to protect against multiple infections, are considered to be the future of vaccinology (Kutzler and Weiner, 2008). The future will bring increased integration of vaccine research with advances in immunology, molecular biology, genomics, proteomics, informatics, and high-throughput instrumentation, collectively defined as the emerging field of "vaccinomics", which is hailed to be responsible for the next 'golden age' in vaccinology (Poland *et al.*, 2008). Awareness of the novel technological possibilities in vaccine research is also expected to grow. Future vaccinology will be based on detailed understanding of immune function, optimal stimulation of immune responses (using adjuvants) and precise mapping and rational selection of immune targets (Brusic *et al.*, 2005). To achieve this, vaccine development will routinely be conducted through large-scale functional studies supported by genomics, proteomics, and informatics techniques prior to clinical trials. This will provide an increased range of immune targets for vaccine design. The author expects the emergence of new generation of vaccines to be personalised to both the genetic make-up of the human population and of the disease agents. In summary, vaccinology will experience rapid progress and will eventually deliver benefits to patients from improved diagnosis, treatment and prevention of diseases.

See also: Computational Pipelines and Workflows in Bioinformatics. Extraction of Immune Epitope Information. Host-Pathogen Interactions. Natural Language Processing Approaches in Bioinformatics

References

- Abelin, J.G., *et al.*, 2017. Mass spectrometry profiling of HLA-associated peptidomes in mono-allelic cells enables more accurate epitope prediction. *Immunity* 46 (2). 315–326.
- Ada, G., 2003. Progress towards achieving new vaccine and vaccination goals. *Internal Medicine Journal* 33, 297–304.
- Adekiya, T., *et al.*, 2017. Structural analysis and epitope prediction of MHC class-I-chain related protein-a for cancer vaccine development. *Vaccines* 6 (1). 1.
- Agostino, M., *et al.*, 2016. Optimization of protein-protein docking for predicting Fc-protein interactions. *Journal of Molecular Recognition* 29 (11). 555–568.
- Altschul, S.F., *et al.*, 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3). 403–410.
- Andre, F.E., *et al.*, 2008. Vaccination greatly reduces disease, disability, death and inequity worldwide. *Bulletin of the World Health Organization* 86, 140–146.
- Andreatta, M., *et al.*, 2015. Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification. *Immunogenetics* 67 (11–12). 641–650.
- Andreatta, M., Nielsen, M., 2015. Gapped sequence alignment using artificial neural networks: Application to the MHC class I system. *Bioinformatics* 32 (4). 511–517.
- Arnon, R., Ben-yedidia, T., 2003. Old and new vaccine approaches. *International Immunopharmacology* 3, 1195–1204.
- Babiuk, L.A., 1999. Broadening the approaches to developing more effective vaccines. *Vaccine* 17, 1587–1595.
- Barrett, A.D.T., 2016. Vaccinology in the twenty-first century. *npj Vaccines* 1 (1). 16009.
- Bhasin, M., Lata, S., Raghava, G.P.S., 2007. TAPPred prediction of TAP-binding peptides in antigens. *Methods in Molecular Biology* 409, 381–386.
- Bhasin, M., Raghava, G.P.S., 2004. SVM based method for predicting HLA-DRB1*0401 binding peptides in an antigen sequence. *Bioinformatics* 20 (3). 421–423.
- Birnbaum, M.E., *et al.*, 2014. Deconstructing the peptide-MHC specificity of t cell recognition. *Cell* 157 (5). 1073–1087.
- Blum, J.S., Wearsch, P.A., Cresswell, P., 2013. Pathways of antigen processing. *Annual Review of Immunology* 31, 443–473.
- Bona, C.A., Casares, S., Brumeanu, T.D., 1998. Towards development of T-cell vaccines. *Immunology Today* 19 (3). 126–133.
- Bouvier, N., Palese, P., 2008. The biology of influenza viruses. *Vaccine* 26, D49–D53.
- Brusic, V., August, J.T., Petrovsky, N., 2005. Information technologies for vaccine research. *Expert Review of Vaccines* 4 (3). 407–417.
- Brusic, V., Petrovsky, N., 2003. Immunoinformatics—The new kid in town. *Novartis Foundation Symposium* 254, 3–22. 98–101, 250–252.
- Brusic, V., Zeleznikow, J., Petrovsky, N., 2000. Molecular immunology databases and data repositories. *Journal of Immunological Methods* 238, 17–28.
- Chowell, D., *et al.*, 2015. TCR contact residue hydrophobicity is a hallmark of immunogenic CD8⁺ T cell epitopes. *Proceedings of the National Academy of Sciences of the United States of America* 112 (14). E1754–E1762.
- Damfo, S.A., *et al.*, 2017. In silico design of knowledge-based Plasmodium falciparum epitope ensemble vaccines. *Journal of Molecular Graphics and Modelling* 78, 195–205.
- Dash, P., *et al.*, 2017. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* 547 (7661). 89–93.
- Dertzbaugh, M.T., 1998. Genetically engineered vaccines: An overview. *Plasmid* 39 (2). 100–113.
- Dhanda, S.K., *et al.*, 2017. Novel in silico tools for designing peptide-based subunit vaccines and immunotherapeutics. *Briefings in Bioinformatics* 18 (3). 467–478.
- Do, C.B., Mahabhashyam, M.S., Brudno, M., Batzoglou, S., 2005. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.* 15 (2). 330–340. PubMed PMID: 15687296; PubMed Central PMCID: PMC546535.
- Dominguez, C., Boelens, R., Bonvin, A.M.J.J., 2003. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society* 125 (7). 1731–1737.
- Doolan, D.L., 2003. Utilization of genomic sequence information to develop malaria vaccines. *Journal of Experimental Biology* 206 (21). 3789–3802.
- Edgar, R.C., 2004. MUSCLE: A multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*. 5, 113. PubMed PMID: 15318951; PubMed Central PMCID: PMC517706.
- von Eichborn, J., *et al.*, 2013. Vacclmm: Simulating peptide vaccination in cancer therapy. *BMC Bioinformatics* 14.
- Evavold, B.D., Sloan-Lancaster, J., Allen, P.M., 1993. Tickling the TCR: Selective T-cell functions stimulated by altered peptide ligands. *Immunology Today* 14 (12). 602–609.
- Fauci, A.S., 2001. Infectious diseases: Considerations for the 21st Century. *Clinical Infectious Diseases* 32, 675–685.
- Fauci, A.S., Touchette, N.A., Folkers, G.K., 2005. Emerging infectious diseases: A 10-year perspective from the National Institute of Allergy and Infectious Diseases. *Emerging Infectious Diseases* 11 (4). 519–525.
- Fleri, W., *et al.*, 2017. The immune epitope database: How data are entered and retrieved. *Journal of Immunology Research* 2017.
- Gee, M.H., *et al.*, 2017. Antigen identification for orphan T cell receptors expressed on tumor-infiltrating lymphocytes. *Cell*.
- Glanville, J., *et al.*, 2017. Identifying specificity groups in the T cell receptor repertoire. *Nature* 547 (7661). 94–98.
- Goodswen, S.J., Kennedy, P.J., Ellis, J.T., 2014. Vacceed: A high-throughput in silico vaccine candidate discovery pipeline for eukaryotic pathogens based on reverse vaccinology. *Bioinformatics* 30 (16). 2381–2383.
- Gourlay, L., *et al.*, 2017. Structure and computation in immunoreagent design: From diagnostics to vaccines. *Trends in Biotechnology* 35 (12). 1208–1220.
- De Groot, A.S., *et al.*, 2002. Immuno-informatics: Mining genomes for vaccine components. *Immunology and Cell Biology* 80 (3). 255–269.
- De Groot, A.S., *et al.*, 2004. Engineering immunogenic consensus T helper epitopes for a cross-clade HIV vaccine. *Methods* 34, 476–487.
- Gupta, A.K., *et al.*, 2016. ZikaVR: An integrated Zika virus resource for genomics, proteomics, phylogenetic and therapeutic analysis. *Scientific Reports* 6.
- Hakenberg, J., *et al.*, 2003. MAPPP: MHC class I antigenic peptide processing prediction. *Applied Bioinformatics* 2 (3). 155–158.
- Haydon, D.T., Woolhouse, M.E., 1998. Immune avoidance strategies in RNA viruses: Fitness continuums arising from trade-offs between immunogenicity and antigenic variability. *J. Theor. Biol.* 193, 601–612.
- He, Y., 2014. Bacterial whole-genome determination and applications. *Molecular Medical Microbiology: Second Edition*. 357–368.
- Hegde, N.R., *et al.*, 2017. The use of databases, data mining and immunoinformatics in vaccinology: Where are we? *Expert Opinion on Drug Discovery* 13 (2). 117–130.
- Heiny, A., *et al.*, 2007. Evolutionarily conserved protein sequences of influenza A viruses, avian and human, as vaccine targets. *PLOS ONE* 2 (11). e1190. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18030326>.
- Hilleman, M.R., 1985. Newer directions in vaccine development and utilization. *Journal of Infectious Diseases* 151 (3). 407–419.
- Hu, Y., *et al.*, 2013. Dissecting the dynamics of HIV-1 protein sequence diversity. *PLOS ONE* 8 (4).
- Hung, M.-C., Link, W., 2011. Protein localization in disease and therapy. *Journal of Cell Science* 124 (20). 3381–3392.
- Jerne, N.K., 1993. The generative grammar of the immune system. *Scandinavian Journal of Immunology* 38 (1). 2–8.
- Jurtz, V., *et al.*, 2017. NetMHCpan-4.0: Improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. *The Journal of Immunology* 199 (9). 3360–3368.
- Katoh, K., Rozewicki, J., Yamada, K.D., 2017. MAFFT online service: Multiple sequence alignment, interactive sequence choice and visualization. *Brief Bioinform.* doi:10.1093/bib/bbx108. [Epub ahead of print] PubMed PMID: 28968734.

- Karosiene, E., *et al.*, 2013. NetMHCIIpan-3.0, a common pan-specific MHC class II prediction method including all three human MHC class II isotypes, HLA-DR, HLA-DP and HLA-DQ. *Immunogenetics* 65 (10): 711–724.
- Kast, W.M., *et al.*, 1994. Role of HLA-A motifs in identification of potential CTL epitopes in human papillomavirus type 16 E6 and E7 proteins. *Journal of Immunology* 152 (8): 3904–3912.
- Kent, W.J., 2002. BLAT – The BLAST-like alignment tool. *Genome Research* 12, 656–664.
- Khan, A.M., 2005. Mapping targets of immune responses in complete dengue viral genomes. MSc Thesis, National University of Singapore. Available at: [http://www.scholarbank.nus.edu.sg/bitstream/handle/10635/15081/MohammadAsifKhan\(KHAN_AM_Thesis.PDF\).pdf?sequence=1](http://www.scholarbank.nus.edu.sg/bitstream/handle/10635/15081/MohammadAsifKhan(KHAN_AM_Thesis.PDF).pdf?sequence=1).
- Khan, A.M., 2009. Antigenic diversity of dengue virus: Implications for vaccine design. PhD Thesis, National University of Singapore. Available at: <http://scholarbank.nus.edu.sg/handle/10635/17142>.
- Khan, A.M., *et al.*, 2006. A systematic bioinformatics approach for selection of epitope-based vaccine targets. *Cellular Immunology* 244 (2): 141–147. Available at: <http://www.sciencedirect.com/science/article/B6WCF-4NH6NCY-2/2/081b67baf4af7b5e9da5c0dd3ea404f>.
- Khan, A.M., *et al.*, 2008. Conservation and variability of dengue virus proteins: Implications for vaccine design. *PLOS Neglected Tropical Diseases* 2 (8): e272. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18698358>.
- Khan, A.M., *et al.*, 2017. Analysis of viral diversity for vaccine target discovery. *BMC Medical Genomics* 10 (Suppl. 4): S78.
- Khan, J.M., Ranganathan, S., 2010. PDock: A new technique for rapid and accurate docking of peptide ligands to major histocompatibility complexes. *Immunome Research* 6 (1): S2. doi: 10.1186/1745-7580-6-S1-S2.
- Knobler, S.L., *et al.*, 2003. The resistance phenomenon in microbes and infectious disease vectors: Implications for human health and strategies for containment. *National Academies Press* 336 (9): 309.
- Koo, Q.Y., *et al.*, 2009. Conservation and variability of West Nile virus proteins. *PLOS ONE* 4 (4): e5352.
- Kulkarni-Kale, U., Bhosle, S., Kolaskar, A.S., 2005. CEP: A conformational epitope prediction server. *Nucleic Acids Research* 33 (Suppl. 2).
- Kutzler, M.A., Weiner, D.B., 2008. DNA vaccines: Ready for prime time? *Nature Reviews Genetics* 9 (10): 776–788.
- Larsen, M.V., *et al.*, 2005. An integrative approach to CTL epitope prediction: A combined algorithm integrating MHC class I binding, TAP transport efficiency, and proteasomal cleavage predictions. *European Journal of Immunology* 35 (8): 2295–2303.
- Lee, N.H., *et al.*, 2012. A review of vaccine development and research for industrially animals in Korea. *Clinical and Experimental Vaccine Research* 1, 18–34.
- Lefranc, M.P., *et al.*, 2009. IMGT®, the international ImmunoGeneTics information system®. *Nucleic Acids Research* 37 (Suppl. 1).
- Lennon-Dumenil, A.-M., *et al.*, 2002. Analysis of protease activity in live antigen-presenting cells shows regulation of the phagosomal proteolytic contents during dendritic cell activation. *The Journal of Experimental Medicine* 196 (4): 529–540.
- Li, F., Finnefrock, A.C., Dubey, S.A., *et al.*, 2011. Mapping HIV-1 vaccine induced T-cell responses: Bias towards less-conserved regions and potential impact on vaccine efficacy in the Step study. *PLoS One*. 6 (6): e20479. doi:10.1371/journal.pone.0020479. Epub 2011 Jun 10. PubMed PMID: 21695251; PubMed Central PMCID: PMC3112144.
- Liljeroos, L., *et al.*, 2015. Structural and computational biology in the design of immunogenic vaccine antigens. *Journal of Immunology Research* 2015.
- Lund, O., *et al.*, 2004. Definition of supertypes for HLA molecules using clustering of specificity matrices. *Immunogenetics* 55 (12): 797–810.
- MacDonald, K.S., *et al.*, 2000. Influence of HLA supertypes on susceptibility and resistance to human immunodeficiency virus type 1 infection. *The Journal of Infectious Diseases* 181 (5): 1581–1589.
- MacDonald, K.S., *et al.*, 2001. Human leucocyte antigen supertypes and immune susceptibility to HIV-1, implications for vaccine design. *DNA Sequence* 9, 151–157.
- Maione, D., *et al.*, 2005. Immunology: Identification of a universal group B streptococcus vaccine by multiple genome screen. *Science* 309 (5731): 148–150.
- Maria, R.R., *et al.*, 2017. The impact of bioinformatics on vaccine design and development. In: Afrin, F. (Ed.), *Vaccines*. Place: IntechOpen, pp. 123–145. doi:10.5772/intechopen.69273. Available from: <https://www.intechopen.com/books/vaccines/the-impact-of-bioinformatics-on-vaccine-design-and-development>.
- Marrack, P., *et al.*, 2017. The somatically generated portion of T cell receptor CDR3 α contributes to the MHC allele specificity of the T cell receptor. *eLife* 6.
- Minichiello, V., 2002. New vaccine technology—what do you need to know? *Journal of the American Academy of Nurse Practitioners* 14 (2): 73–81.
- Mistry, J., Flower, D.R., 2017. Designing epitope ensemble vaccines against TB by selection: Prioritizing antigens using predicted immunogenicity. *Bioinformation* 13 (7): 220–223.
- Monterrubio-López, G.P., González-Y-Merchand, J.A., Ribas-Aparicio, R.M., 2015. Identification of novel potential vaccine candidates against tuberculosis based on reverse vaccinology. *BioMed Research International* 2015.
- Morens, D.M., Fauci, A.S., 2013. Emerging infectious diseases: Threats to human health and global stability. *PLOS Pathogens* 9 (7): e1003467.
- Nandy, A., Basak, S.C., 2016. A brief review of computer-assisted approaches to rational design of peptide vaccines. *International Journal of Molecular Sciences* 17 (5): 2017. Database resources of the national center for biotechnology information. *Nucleic Acids Research* 45 (D1): D12–D17.
- Nicholson, L.B., 2016. The immune system. *Essays in Biochemistry* 60 (3): 275–301.
- Nielsen, M., *et al.*, 2005. The role of the proteasome in generating cytotoxic T-cell epitopes: Insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics* 57 (1–2): 33–41.
- Nielsen, M., *et al.*, 2010. MHC Class II epitope predictive algorithms. *Immunology* 130 (3): 319–328.
- NIH, 2003. Understanding the Immune System How It Works. NIH. pp. 1–7.
- Nussbaum, A.K., *et al.*, 2001. PProC: A prediction algorithm for proteasomal cleavages available on the WWW. *Immunogenetics* 53 (2): 87–94.
- Pappalardo, F., Chiacchio, F., Motta, S., 2013. Cancer vaccines: State of the art of the computational modeling approaches. *BioMed Research International* 2013, 106407.
- Park, A.W., *et al.*, 2004. The effects of strain heterology on the epidemiology of equine influenza in a vaccinated population. *Proceedings of the Royal Society B: Biological Sciences* 271 (1548): 1547–1555.
- Parker, K.C., Bednarek, M.A., Coligan, J.E., 1994. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *Journal of Immunology* 152 (1): 163–175.
- Paul, S., Dillon, M.B.C., *et al.*, 2015a. A population response analysis approach to assign class II HLA-Epitope restrictions. *The Journal of Immunology* 194 (12): 6164–6176.
- Paul, S., Lindestam Arleham, C.S., *et al.*, 2015b. Development and validation of a broad scheme for prediction of HLA class II restricted T cell epitopes. *Journal of Immunological Methods* 422, 28–34.
- Paul, S., *et al.*, 2013. HLA class I alleles are associated with peptide-binding repertoires of different size, affinity, and immunogenicity. *The Journal of Immunology* 191 (12): 5831–5839.
- Petrovsky, N., Brusic, V., 2002. Computational immunology: The coming of age. *Immunology and Cell Biology* 80 (3): 248–254.
- Petrovsky, N., Schönbach, C., Brusic, V., 2003. Bioinformatic strategies for better understanding of immune function. *In Silico Biology* 3 (4): 411–416.
- Pizza, M., *et al.*, 2000. Identification of vaccine candidates against serogroup B meningococcus by whole-genome sequencing. *Science* 287 (5459): 1816–1820.
- Plotkin, S.A., 2005. Vaccines: Past, present and future. *Nature Medicine* 11 (4 Suppl.): S5–S11.
- Poland, G.A., Ovsyannikova, I.G., Jacobson, R.M., 2008. Personalized vaccines: The emerging field of vaccinomics. *Expert Opinion on Biological Therapy* 8 (11): 1659–1667.
- Pro, S.C., *et al.*, 2015. Automatic generation of validated specific epitope sets. *Journal of Immunology Research* 2015, 763461.
- Raman, H., *et al.*, 2014. Bioinformatics for vaccine target discovery. *Asia Pacific Biotech News* 18 (9): 25–53.
- Rammensee, H.-G., *et al.*, 1999. SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* 50 (3–4): 213–219.

- Ranganathan, S., Tong, J.C., 2007. A practical guide to structure-based prediction of MHC-binding peptides. *Methods in Molecular Biology* 409, 301–308.
- Rappuoli, R., 2001. Reverse vaccinology, a genome-based approach to vaccine development. *Vaccine* 19 (17–19), 2688–2691.
- Rappuoli, R., Covacci, A., 2003. Reverse vaccinology and genomics. *Science* 302 (5645), 602.
- Rizwan, M., *et al.*, 2017. VacSol: A high throughput in silico pipeline to predict potential therapeutic targets in prokaryotic pathogens using subtractive reverse vaccinology. *BMC Bioinformatics* 18 (1).
- Rosloniec, E.F., *et al.*, 1997. An HLA-DR1 transgene confers susceptibility to collagen-induced arthritis elicited with human type II collagen. *The Journal of Experimental Medicine* 185 (6), 1113–1122.
- Rothman, A.L., 2004. Dengue: Defining protective versus pathologic immunity. *Journal of Clinical Investigation* 113 (7), 946–951.
- Ryan, C.P., 2008. Zoonoses likely to be used in bioterrorism. *Public Health Reports* 123 (3), 276–281.
- Sachdeva, G., *et al.*, 2005. SPAAN: A software program for prediction of adhesins and adhesin-like proteins using neural networks. *Bioinformatics* 21 (4), 483–491.
- Sankar, S., Nayanar, S., Balasubramanian, S., 2013. Current trends in cancer vaccines – a bioinformatics perspective. *Asian Pacific Journal of Cancer Prevention* 14 (7), 4041–4047.
- Sette, A., *et al.*, 2001. The development of multi-epitope vaccines: Epitope identification, vaccine design and clinical evaluation. *Biologicals*, 271–276.
- Sette, A., Fikes, J., 2003. Epitope-based vaccines: An update on epitope identification, vaccine design and delivery. *Current Opinion in Immunology* 15 (4), 461–470.
- Sette, A., Sidney, J., 1999. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. *Immunogenetics* 50, 201–212.
- Shastri, N., Schwab, S., Serwold, T., 2002. Producing nature's gene-chips: The generation of peptides for display by MHC class I molecules. *Annual Review of Immunology* 20 (1), 463–493. Available at: <http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.immunol.20.100301.064819>.
- Sievers, F., Wilm, A., Dineen, D., *et al.*, 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* 7, 539. doi:10.1038/msb.2011.75. PubMed PMID: 21988835; PubMed Central PMCID: PMC3261699.
- Sloan-Lancaster, J., Allen, P.M., 1996. Altered peptide ligand-induced partial T cell activation: Molecular mechanisms and role in T cell biology. *Annual Review of Immunology* 14, 1–27.
- Soria-Guerra, R.E., *et al.*, 2015. An overview of bioinformatics tools for epitope prediction: Implications on vaccine development. *Journal of Biomedical Informatics* 53, 405–414.
- Sprenger, J., *et al.*, 2008. LOCATE: A mammalian protein subcellular localization database. *Nucleic Acids Research* 36 (Suppl. 1), D230–D233.
- Srinivasan, K.N., *et al.*, 2004. Prediction of class I T-cell epitopes: Evidence of presence of immunological hot spots inside antigens. *Bioinformatics* 20 (Suppl. 1).
- Strassburg, M.A., 1980. The global eradication of smallpox. *American Journal of Infection Control* 10 (2), 53–59.
- Sturniolo, T., *et al.*, 1999. Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. *Nature Biotechnology* 17 (6), 555–561.
- Sylvester-Hvid, C., *et al.*, 2002. Establishment of a quantitative ELISA capable of determining peptide – MHC class I interaction. *Tissue Antigens* 59, 251–258.
- Tang, Y.W., Sussman, M., Liu, D., Poxton, I., Schwartzman, J. (Eds.), 2014. *Molecular Medical Microbiology*, second edn. Academic Press.
- Tong, J.C., Tan, T.W., Ranganathan, S., 2007. Methods and protocols for prediction of immunogenic epitopes. *Briefings in Bioinformatics* 8 (2), 96–108.
- Del Tordello, E., Rappuoli, R., Delany, I., 2016. Reverse vaccinology: Exploiting genomes for vaccine design. *Human Vaccines: Emerging Technologies in Design and Development*, 65–86.
- Trolle, T., *et al.*, 2015. Automated benchmarking of peptide-MHC class I binding predictions. *Bioinformatics* 31 (13), 2174–2181.
- Vernikos, G.S., 2008. Genome watch: Overtake in reverse gear. *Nature Reviews Microbiology* 6 (5), 334–335.
- Vigneron, N., Van den Eynde, B.J., 2014. Proteasome subtypes and regulators in the processing of antigenic peptides presented by class I molecules of the major histocompatibility complex. *Biomolecules* 4 (4), 994–1025.
- Vivona, S., Bernante, F., Filippini, F., 2006. NERVE: New enhanced reverse vaccinology environment. *BMC Biotechnology* 6, 35.
- De Vries, S.J., Van Dijk, M., Bonvin, A.M.J.J., 2010. The HADDOCK web server for data-driven biomolecular docking. *Nature Protocols* 5 (5), 883–897.
- Wizemann, T.M., Adamou, J.E., Langermann, S., 1999. Adhesins as targets for vaccine development. *Emerging Infectious Diseases* 5 (3), 395–403.
- Yatim, K.M., Lakkis, F.G., 2015. A brief journey through the immune system. *Clinical Journal of the American Society of Nephrology* 10 (7), 1274–1281.
- Zaharieva, N., *et al.*, 2017. Immunogenicity prediction by VaxiJen: A ten year overview. *Journal of Proteomics & Bioinformatics* 10 (11), 298–310.
- Zahroh, H., *et al.*, 2016. Immunoinformatics approach in designing epitopebased vaccine against meningitis-inducing bacteria (*Streptococcus pneumoniae*, *Neisseria meningitidis*, and *Haemophilus influenzae* type b). *Drug Target Insights* 10, 19–29.
- Zhang, G., *et al.*, 2005. MULTIPRED: A computational system for prediction of promiscuous HLA binding peptides. *Nucleic Acids Res* 33 (Web Server issue), W172–W179. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15980449>.
- Zhang, G.L., *et al.*, 2006. PRED(TAP): A system for prediction of peptide binding to the human transporter associated with antigen processing. *Immunome Research* 2 (1), 3.
- Zhang, W., *et al.*, 2011. Prediction of conformational B-cell epitopes from 3D structures by random forests with a distance-based feature. *BMC Bioinformatics* 12.
- Zhang, Y., *et al.*, 2017. Influenza research database: An integrated bioinformatics resource for influenza virus research. *Nucleic Acids Research* 45 (D1), D466–D474.

Further Reading

- Aparicio, R., *et al.*, 2017. World's largest science, technology & medicine open access book publisher. The impact of bioinformatics on vaccine design and development.
- Barrett, A.D.T., 2016. Vaccinology in the twenty-first century. *NPJ Vaccines* 1 (1), 16009.
- Brusic, V., August, J.T., Petrovsky, N., 2005. Information technologies for vaccine research. *Expert Review of Vaccines* 4 (3), 407–417.
- Brusic, V., Petrovsky, N., 2003. Immunoinformatics – The new kid in town. *Novartis Foundation Symposium* 254, 13–22. 98–101, 250–252.
- Brusic, V., Zeleznikow, J., Petrovsky, N., 2000. Molecular immunology databases and data repositories. *Journal of Immunological Methods* 238, 17–28.
- Dhanda, S.K., *et al.*, 2017. Novel in silico tools for designing peptide-based subunit vaccines and immunotherapeutics. *Briefings in Bioinformatics* 18 (3), 467–478.
- Gourlay, L., *et al.*, 2017. Structure and computation in immunoreagent design: From diagnostics to vaccines. *Trends in Biotechnology* 35 (12), 1208–1220.
- Hegde, N.R., *et al.*, 2017. The use of databases, data mining and immunoinformatics in vaccinology: Where are we? *Expert Opinion on Drug Discovery* 13 (2), 117–130.
- Khan, A.M., *et al.*, 2017. Analysis of viral diversity for vaccine target discovery. *BMC Medical Genomics* 10 (Suppl. 4), S78.
- Mistry, J., Flower, D.R., 2017. Designing epitope ensemble vaccines against TB by selection: Prioritizing antigens using predicted immunogenicity. *Bioinformatics* 13 (7), 220–223.
- Tong, J.C., Ranganathan, S., 2013. Computer-Aided Vaccine Design, Woodhead Publishing Series In Biomedicine No. 23. Cambridge, UK: Woodhead, pp. 1–164.

Relevant Website

www.ncbi.nlm.nih.gov/genomes/VIRUSES/viruses.html
Viral Genomes.

Biographical Sketch



Chong Li Chuin is a Biomedical Science undergraduate student from Faculty of Medicine and Health Sciences, UPM. As part of her undergraduate studies, she completed an internship at Perdana University Centre for Bioinformatics (PU-CBi), Malaysia. She continued to be involved with the Centre to build on her passion for bioinformatics.



Associate Professor Dr. Mohammad Asif Khan is currently the Dean of the Perdana University School of Data Sciences, the Director of Perdana University Centre for Bioinformatics (PU-CBi), Malaysia, and a Visiting Scientist at Johns Hopkins University School of Medicine (JHUSOM), USA. He obtained his PhD in Bioinformatics from the National University of Singapore (NUS), and did his postdoctoral fellowships at NUS and JHUSOM. Dr. Khan's research interests are in the area of biological data warehousing and applications of bioinformatics to the study of immune responses, vaccines, venom toxins, drug design, and disease biomarkers.

Protocol for Protein Structure Modelling

Amara Jabeen, Abidali Mohamedali, and Shoba Ranganathan, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Protein structure has four levels of organization. The primary structure is represented by sequence of amino acids bound together by peptide bonds. When backbone atoms of one amino acid residue make hydrogen bonds with the backbone atoms of other residues, α -helices and β -pleated sheets are formed that comprise the secondary structure of a protein. These secondary structural elements are connected by loop regions. The folding of secondary structure elements into 3-dimensional (3D) space yields tertiary structure of a protein. In some proteins, folded chains of more than one polypeptide interact with each other to form quaternary structure. Protein structure dictates its function (Voet *et al.*, 2016). Mutations that alter the structure of a protein can lead to alteration of function which can either be detrimental or beneficial, therefore a thorough understanding of protein structure is mandatory to have insight into the biology of life (Starr *et al.*, 2011). Moreover, a 3D structure serves as an important starting point for insights into essential features of an evolutionary protein family, thereby establishing a possibility of identification of even remotely related proteins through structural similarities (Lacapere, 2017).

Protein structures that are determined through experimental methodologies (primarily by X-ray crystallography (XRC), nuclear magnetic resonance spectroscopy (NMR) and electron microscopy) are often submitted to the Protein databank (PDB) which is the largest repository for experimentally determined structures of proteins, nucleic acids and other complex assemblies (Berman *et al.*, 2000). There are numerous derived databases representing PDB files for specific families of proteins and their related information: for instance, PDBTM has transmembrane proteins selection of PDB (Kozma *et al.*, 2012).

Despite the rapid increase in the percentage of structures determined from different experimental methods (Table 1), there is still a vast gap between protein sequence (99,261,416 UniProt entries as on December 2017) and structure (125,799 protein PDB entries as on December 2017) entries. Some classes of proteins are significantly under-represented in the PDB: for instance, membrane proteins represent 40% of the human proteome (Lacapere, 2017) but only 702 structures (as on July 2017) have been solved experimentally. This is primarily due to the difficulty in crystallizing these proteins. Olfactory receptors, which represent the largest family of G protein coupled receptors (GPCRs) do not have a single experimentally solved structure. The lack of solved structures of proteins have long been recognized as a bottleneck in rational drug design; for instance, casein kinase II (CK2) has been extensively studied due to its implications in cancer, with potential inhibitors searched by testing against CK2. The CK2 XRC structure (PDB ID: 3AT2) has paved the way for discovery and optimization of its inhibitors through structure-based drug designing (Cozza, 2017).

Computational approaches can be used to predict the structure of the protein at a fraction of the cost and time compared to the experimental methods. Though these methods are not as accurate as experimental methods, the quality and efficiency of structure prediction in the past few years has demonstrated that it can be a viable complement to experimental methods (Khor *et al.*, 2015). The majority of structure-prediction methods provide structural templates closely matching the query protein sequence. This article provides the protocol for structure prediction of a protein through template-based modelling, with the basic aim of providing practical guide to readers to use computational approaches for structure prediction.

Methods for Protein Structure Prediction

Protein structure prediction is the method of inference of protein's 3D structure from its amino acid sequence through the use of computational algorithms. The 3D structure of a protein is predicted on the basis of two principles; the laws of physics and evolution. According to physical principles, protein folds into a stable and well-formed structure by minimizing the energy, while according to evolutionary principles, a protein molecule is a consequence of gradual changes in sequence and structure over the

Table 1 Experimental methods for protein entries in the Protein Databank

Experimental method	Number of structures resolved ^a
X-ray crystallography	111,712
Nuclear magnetic resonance spectroscopy	10,476
Electron microscopy	1,233
Hybrid	102
Other	199
Total	123,722

^aAs on 4 September 2017.

course of time. These two principles set up the foundation of the two protein structure prediction methods that are template-based modelling (homology modelling and threading method) and template free modelling (*ab initio* modelling), respectively (Illergård *et al.*, 2009). Hybrid methods utilize both of these methods. The selection of the methods depends primarily upon the availability of the templates structures in PDB.

Protein structure prediction took its ground with the thermodynamics hypothesis proposed by an American biochemist Christian Anfinsen in 1962. This hypothesis was based on his work conducted in 1954 on folding of bovine pancreatic ribonuclease into 3D conformation, i.e., native conformation (Anfinsen *et al.*, 1954), and then in 1961 he reported refolding of unfolded bovine pancreatic ribonuclease A into its native conformation (Scheraga, 2014). During the Anfinsen's Nobel acceptance speech in 1972 he talked about the dependence of 3D conformation upon amino acid sequence, which motivated different groups to work on protein structure prediction problem. One of the earliest efforts was done by Gibson and Scheraga in 1967 in finding thermodynamically stable conformation of protein by using different computer searching techniques (Wooley and Ye, 2007). The first success in the area of protein fold prediction from its amino acid sequence came in 1975 when Levitt and Warshel folded bovine pancreatic trypsin inhibitor (58 residues long) using energy minimization. In 1969 Browne and his co-workers exploited the observation that proteins that share sequence similarities share similar structure, to predict the structure of α -lactalbumin using the X-ray structure of lysozyme as a template, and this laid the foundation for comparative modelling (Guo *et al.*, 2008). After the prediction of the first homology model, continuous improvements have been made, from semi-automated to fully automated homology modelling. Homology (Greer, 1981) was the first semi-automated program and Modeller (Šali and Blundell, 1993) was the first fully automated program. The first ever fully automated server with web interface was then built in 1993, named SWISS-MODEL (Guex *et al.*, 2009). Great number of programs and fully automated servers are now available for solving protein structure prediction problem.

To bridge the gap between protein sequences and structures and to encourage the development of fast, efficient and accurate structure prediction methods, a community-wide global protein structure prediction experiment known as the Critical Assessment of protein Structure Prediction, or CASP was started in 1994, with the sequences of protein structures about to be experimentally solved offered as targets for prediction. Participants then submit their predicted models which are then compared to experimentally determined structures (Moult *et al.*, 2016). The main classes of protein structure prediction methods are briefly described below.

Homology Modelling

Homology modelling has proven to be the most successful approach for protein structure prediction but it requires that a template exists in PDB (Fiser, 2010), such that, the better the template, the more accurate the prediction. This approach is greatly benefited by the fact that structure of at least one member of almost all the families of proteins have been deduced experimentally which can be used as a template for modelling the other members of that particular family (Song *et al.*, 2013).

Threading

Threading relies on the observation that a limited number of folds are present in nature due to which even remotely homologous proteins have similar structures (Kelley and Sternberg, 2009). Protein threading methods are based on a *Z-score* which tests the possibility of target protein sequence to fold in the structure that is similar to the template structure. The *Z-score* represents the distance between optimal alignment score and mean alignment score that is obtained by random shuffling of a target sequence. This method involves building up databases for representative, non-redundant template structures, followed by devising a fitness scoring function which is based on the knowledge of known sequence to structure relationships. The scoring function is then minimized by finding the optimal alignment between the target and template and in the end, best template is selected based on all the template sequence alignments (Markstein and Xu, 2006).

Ab initio Methods

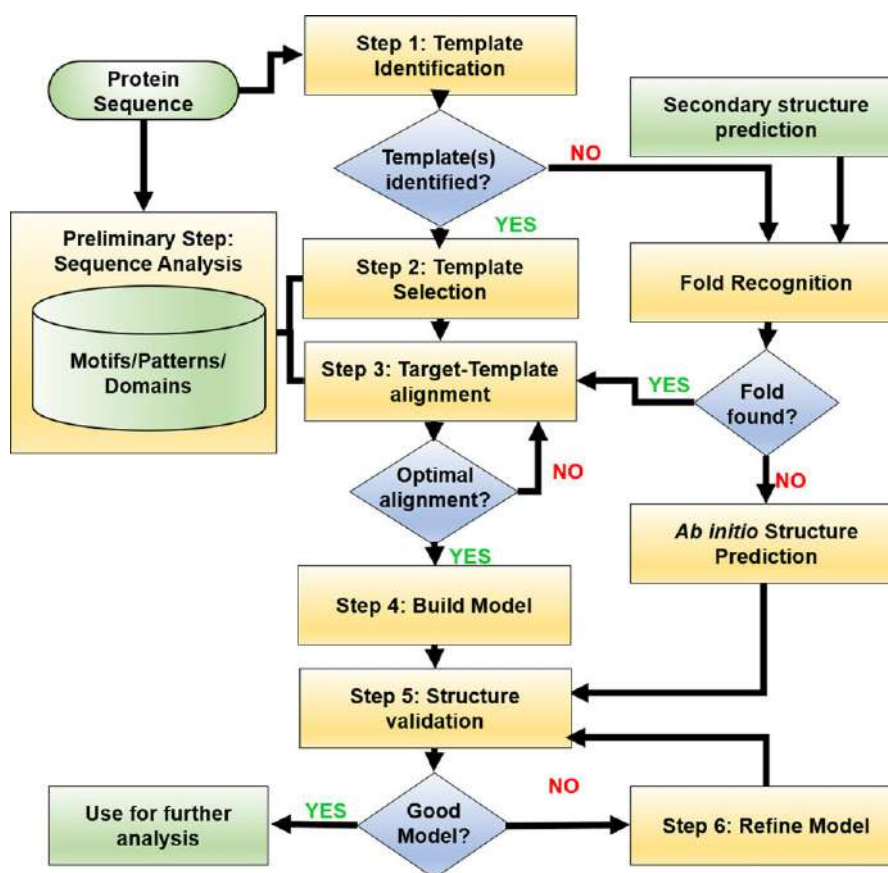
When a matching template cannot be found for a query sequence by use of homology or threading approaches, the *ab initio* methods are used to model the structure. These template-free methods rely on a set of core principles that are based on folding energetics and statistical tendencies of conformational characteristics demonstrated by experimental structures (Lee *et al.*, 2017). The general steps include devising an accurate energy function to estimate which model structures are thermodynamically stable, using an efficient search method that can identify lowest energy conformations, and selecting the near native conformation from a pool of structure decoys.

Ab initio methods are the least accurate of all the structure prediction methods and have limitations of protein size, and therefore is only used to model smaller proteins/peptides (Lee *et al.*, 2017).

Over the past decade substantial improvement has been made in the performance of different protein structure prediction servers. Table 2 lists the top ten automated structure prediction servers according to the recent CASP 12 ranking.

Table 2 Top 10 automated structure prediction servers according to CASP 12 ranking^a

Server	Rank (based on sum Z-score)	Protein size limit	URL
Zhang server	1	1500	http://zhanglab.ccmb.med.umich.edu/I-TASSER/
QUARK	2	150	http://zhanglab.ccmb.med.umich.edu/QUARK/
BAKER-ROSETTA SERVER	3	650	http://rosetta.bakerlab.org/
GOAL	4		Not available publicly
RAPTORX	5	2500	http://RaptorX.uchicago.edu/
ToyPred_email	6		Not available publicly
MULTICOM-CONSTRUCT	7		Not available publicly
MULTICOM-CLUSTER	8	No limit	http://sysbio.mnet.missouri.edu/multicom_cluster/
MULTICOM-NOVEL	9		Not available publicly
IntFOLD4	10	2000	http://www.reading.ac.uk/bioinf/IntFOLD/

^aRanking is based on analysis of Z-score for GDT_TS.**Fig. 1** Flow diagram for protein structure prediction.

Once a 3D structure for a protein sequence has been predicted using any of the three methods described above, the quality of the predicted structure needs to be thoroughly checked. CASP has been instrumental in developing methods for evaluating the quality and accuracy of structural models (Schwede, 2013).

Given a protein sequence with unknown structure, predicting its 3D structure is a laborious task, especially due to continuous improvements in structural knowledge and prediction methods as a result of advances in the area of structural bioinformatics. We present a comprehensive protocol for structure modelling, exemplified by a case study.

Protocol for Protein Structure Modelling

Given a protein sequence with unknown structure, the following protocol can be used to predict its 3D structure (Fig. 1), with each step elaborated in the following sub-sections.

Preliminary Step: Sequence Analysis

Analyzing the target sequence assists in providing useful biological information which can be further utilized in template selection and sequence alignment. The features that can be examined within the target sequence are as follows:

Protein family analysis

It has been established that two protein sequences of a length of over 100 amino acids that share more than 30% identity are predicted to have a similar structure and function (Pearson, 2013). Such proteins are called homologues (“siblings”) and are assumed to have evolved from a common ancestor. When the sequence identity between two proteins is within and below 20%–35%, they fall in the twilight (“cousins”) zone (Bahar *et al.*, 2017). The structure relatedness is not obvious from sequence relationship in this case and therefore can be inferred from structural alignment. Structure is three to 10 times more conserved than sequence because amino acid changes in the primary sequence are reflected in the structure in terms of type and location of the changes in 3D space. Even a single amino acid mutation could affect the structure completely, but if physicochemical properties remain conserved, the structure remains unaffected (Illergård *et al.*, 2009). For example, haemoglobin from a trematode which binds and transports oxygen (PDB: 1H97) has only 12.1% sequence identity with the haemoglobin of an annelid (PDB: 2GTL), which is part of a larger protein, erythrocrutorin (3.6 million Da) and performs other functions in addition to binding and transporting oxygen. However, both share similar structures (RMSD: 2.3 Å), are part of the globin-like superfamily and have similar function (Sousounis *et al.*, 2012). Despite the low sequence identity, proteins with a similar structure and function have remarkable conservation in the active site residues such that if an accurate model of the active site can be obtained, it can be utilized to elucidate the protein’s function and ligands.

Conservation of residues important for structure and function

The function of a protein depends on the localization in space of a few key residues that are conserved through evolution and occupy identical positions in corresponding structures, usually in a solvent accessible cavity. Some residues are important for the stability of the protein fold or for formation of quaternary structures. If a protein contains buried charged residues that are conserved among homologues, such residues are also usually important for maintaining function (Isom and Dohlman, 2015).

Motifs and patterns

Identification of patterns and motifs can be done through databases (listed in Section Useful Resources) and literature searches. These patterns can then be mapped onto a template to have optimal alignment (Taylor, 1992).

Domain analysis

Protein domains represent the autonomously folding functional units of a protein, which can be present as a repeated unit or in combination with other domains (Marchler-Bauer *et al.*, 2012). Protein sequence of length greater than 150 must be checked for the occurrence of multiple domains.

Template Identification

Template structure defines the framework for 3D structure prediction of a target protein. Templates are identified by aligning the target (query) sequence with sequences of available structures in PDB. The three available methods for template identification by alignment are summarized in Table 3.

Sequence-based methods are more specific but less sensitive than profile-based methods, which are more sensitive but less specific than structure-based methods. Basic Local Alignment Search Tool (BLAST) search results give clues of templates and region of high similarity but for model building we must align the target and template sequences globally. Every domain of the query protein must have a template to model it. If for any domain a template does not exist, it cannot be modelled by homology or comparative modelling. Homologue identification can be improved by splitting the long sequences into domains as determined by comparison with domain databases, and then searching a homologue for each domain.

Criteria for defining a homologue

The following parameters should be considered while defining the homology between two proteins:

Table 3 Methods for template identification

Alignment methodology	Description	Example
Sequence-based	Suitable for high identity	Protein BLAST
Profile-based	Suitable twilight-zone proteins	psi-BLAST, SAM, HHSEARCH, FFAS
Structure-based	Incorporates structural features for instance, secondary structure and solvent accessibility	RAPTOR, SPARKS, MUSTER, LOMETS

1. Alignment score and e-value: high alignment scores and low e-values (rule of thumb: <0.005) are usually desired.
2. Domain coverage: the aligned region between the two proteins must cover at least 60% of the structural domain involved.
3. Gaps: fewer gaps are better as they result in a higher quality model. Long and contiguous insertion/deletion ("indels") regions are preferred to several short indels, for generating a better structural model. Templates with fewer gaps and reasonable similarity scores are preferred over the templates with high scores and but too many gaps.

Template identification for small proteins

Small proteins lack a hydrophobic core and are stabilized by disulphide bridges, between cysteine side chains (Thangudu *et al.*, 2008) or by binding metal ions. Thus, while identifying templates for small proteins, the conservation should be restricted to the cysteine residues involved in making the disulphide bridges, and to residues that are involved in binding metal ions without which the proteins would not be able to fold properly. The size of the loops between the conserved cysteine residues is also crucial as gaps in such regions can affect the predicted model. The right modelling approach for small proteins must be able to produce a model with disulphide bridges or metal ions present intact, to make it biologically relevant.

Targets without homologues

If no considerable homologue is retrieved for the target by any of above tools, then its secondary structure can be predicted, which can give clues about class of the protein and order of occurrence of secondary structure elements that can then be used for matching with a fold, predicted by fold recognition methods (Section Useful Resources). If this also does not help, then *ab initio* methods can be used for model building. The regions of target sequence which does not have any template can be built up using an *ab initio* approach.

Template Selection

If only one template is identified, then model should be built with this template, otherwise appropriate template(s) may be selected from the pools of templates. For selection of templates a number of factors need to be considered, which include:

Sequence similarity

Multiple sequence alignment of all the domains of potential templates and target sequence can be performed and utilized in building a phylogenetic tree which can then depict the most similar structure that can be used for model building.

Similarity in secondary structure

A template with similar secondary structure as the predicted secondary structure of the target can be prioritised.

Bounded ligands

If the ultimate goal of structure prediction is receptor and ligand interactions studies, then a template that has the same binding ligand (if any) as of the protein of interest should be preferred.

Resolution of template structure

High resolution XRC structures are preferred over low resolution structures. Similarly, XRC structures are preferred over NMR. But if only NMR structures are available, the "average" NMR structure must be chosen for model building as NMR protein structures are often submitted in PDB as ensembles of models.

Presence of missing residues

Missing residues (if any) information is present in REMARK 465 in the header section of a PDB file. Missing residues information can also be seen by using most PDB viewer. If missing residues are present, then an alternate template should be selected, otherwise more than one template should be used to fill in the gaps. If the ultimate goal of structure prediction is to study protein interactions, then no missing residues should be present in the binding sites of the template.

Multiple templates

Optimal use of multiple templates can increase the accuracy of the developed model. If every template is providing unique information and there are minimal overlapping residues between templates, they can be selected for building a model. Superimposing C α atoms can reveal the unique information each template is contributing. Other features to look at are insertions, variation in the length of secondary structure elements and different conformational loops. Few of the programs can build model by using multiple templates like Modeller, IntFOLD and M4T (Fiser, 2010). If sequence identity falls below 40%, multiple template usage is favorable but above this identity a single template will usually suffice (Fiser, 2010).

Target-Template Alignment

The most crucial task in modelling that guides the model building and determines the quality of a model is target to template alignment. When sequence identity between target and template drops below 40%, accurate alignment is critical. Incorrect alignment of even a single residue can result in error of approximately 4 Å in the resulting model (Fiser, 2010).

For optimal alignment, the profile of target sequence can be generated by aligning it with its close homologues through multiple sequence alignment (MSA) through any MSA tool like T-COFFEE or ClustalW, which can show the conserved and variable parts of the target sequence. As structure-based alignment is more significant than sequence-based alignment, it is good to align all the selected templates structurally by using UCSF Chimera. Structure-based sequence alignment (SSA) can then be extracted from already aligned structures in UCSF Chimera and aligned with multiple sequence alignment (MSA) of target sequence using profile alignment without any modification of alignment within SSA or MSA. The target-template alignment can now be extracted from this profile. The following checks must be performed to improve the quality of the alignment.

Residue conservation

- Analysis of positions of identical and similar residues in terms of chemical similarity, charge, size and nature.
- Presence of functional regions, motifs and patterns must be analyzed preferably by mapping available experimental information.
- Histidine-rich sites often binds metal ions that are necessary for proper folding of small proteins (Section Template identification for small proteins) while proline has a structural role in guiding the protein folding. Therefore, the conservation of cysteine, proline and histidine residues must be checked for small proteins.

Gaps in alignment

Small gaps (indels) separating few residues should be combined into a long gap. Different sequence alignment methods (e.g., T-COFFEE or ClustalW) can provide clues regarding aggregation of gaps.

Gaps in secondary structures

If there are any indels present within α -helices or β -strands, they must be moved to the closest loop. Indels are normally placed at the end of the loop where the backbone is turning around to have a better quality model. These indels can be mapped onto the template by visualizing through any PDB viewer.

If multiple templates are chosen, then they all must be superimposed for structure alignment using any structure alignment tool such as UCSF Chimera. Structure-based sequence alignment can be extracted and can be used for model building, this alignment is different from sequence based alignment and provides improved results especially in case of low homology. Final target-template alignment is then passed to the next step of model construction.

Model Building

A number of tools are available for constructing a model on the basis of target-template alignment. Table 4 summarizes the features of different programs used for modelling. Programs such as Modeller and I-TASSER provide a model refinement feature, in addition to model building. Others like WhatIF and MOE include the structure validation feature as well. Numerous webserver are also there that take in query sequence as an input and pass it through a pipeline of different programs for template identification, sequence alignment, model building, structure validation and refinement. The webserver used for automated structure prediction are summarized in Table 5. The most popular ones are I-TASSER, Robetta and Swiss model. A few servers such as I-TASSER and Swiss model even allow the user to select the template of their choice. Webserver like Bhageerath-H follow a hybrid approach of modelling, i.e., modelling is based on a template but if template is not spanning the query sequence somewhere, *ab initio* modelling will then be used to provide a complete model.

Structure Quality Analysis

It is essential to first evaluate the quality of the predicted model before using it for further analysis. The main features of the predicted model that can be assessed include bond lengths, bond angles, torsion angles, atomic clashes, distribution of hydrophobic and polar residues (Feig, 2016). Numerous webserver and tools are available to validate the protein structural model (Table 6). Each server validates different features like stereo-chemical correctness or knowledge-based statistical measures that have been derived from experimentally solved high resolution structures and has defined threshold for validation of the model (Schwede, 2013). For instance, the PSVS server calculates *Z-score* for protein structure validation that is based on a set of 252 X-ray structures <500 residues, of resolution ≤ 1.80 Å, R-factor ≤ 0.25 and R-free ≤ 0.28 and models with positive scores are considered as good models. Many webserver also integrate several tools into one platform, such as protein structure analysis and validation (ProTSAV) and protein structure validation software suite (PSVS). To evaluate the performance of quality validation tools, quality validation category was introduced in CASP7 and from then onwards performance of quality validation servers have been evaluated in CASP every 2 years (McGuffin *et al.*, 2013).

Table 4 Programs available for protein structure modelling

Method	OS	Freely available	Single/multiple templates	Method used	Key features
Modeller Webb and Sali (2014)	Windows, Mac, UNIX/Linux	Free for academia	Both	Modelling by satisfaction of spatial restraints	1. Most frequently used 2. Protein sequences and/or structures comparison 3. Clustering of proteins 4. Sequence databases searching 5. Model refinement 6. Model Loops
WHATIF Vriend (1990)	Windows, Linux	No	Single	CONCOORD (generation of geometric bonds from template and modelling on the basis of geometric bonds) Global optimization	1. Display, manipulate and analyze small molecules, proteins, nucleic acids, and their interactions. 2. Cannot model indels 3. Structure validation
ICM Abagyan <i>et al.</i> (1994)	Windows, Mac, UNIX/Linux	Under academic licence	Single		1. Interface for molecular modelling, fully-flexible ligand and receptor docking, virtual ligand screening
MOE Molecular Operating Environment (MOE) (2017)	Windows, Mac, Linux	No	Both	Segment matching procedure (Levitt, 1992)	1. Highly interactive 2. Homology detection 3. Structure- based alignment facility 4. Side chain and loop modelling 5. Structure validation
I-TASSER Yang <i>et al.</i> (2015)	Linux	Free for academia	Both	Replica-exchange Monte-Carlo simulation technique	1. Integrated package for protein structure and function prediction 2. Use multiple threading technique to identify template 3. Model refinement
Nest Petrey <i>et al.</i> (2003)	Linux	Free for academia	Both	Artificial evolution algorithm	1. Side chain modelling 2. Loop modelling 3. Rapid model building

Table 5 Webservers available for protein structure modelling

Server	Domain boundary prediction	Template identification	Method for Sequence-template alignment	Method for model generation	Model refinement	Quality assessment
I-TASSER Zhang (2008)	Built-in process	LOMETS (User can also provide template) v	PPA Decoy-based optimized potential	Replica-exchange Monte carlo simulation technique	REMO	C-score ^a
Robetta Chivian <i>et al.</i> (2003)	Ginzu	BLAST, Psi-BLAST, FFAS03 and 3o-Jury	K*Sync	Rosetta fragment-insertion technique ¹	–	DISOPRED
RaptorX Peng and Xu (2011)	Pfam database searching	Statistical learning methods ^b	CNFpred	MODELLER	–	DISOPRED
Seok server Ko <i>et al.</i> (2012)	GalaxyDom	HHsearch	Promals3D	GalaxyCassiopeia	GalaxyRefine	GalaxyQA
InfOLD4 McGuffin <i>et al.</i> (2015)	DomFOLD3	SP3, SPARKS2, HHsearch, COMA, SPARKSX, CNFsearch and LOMETS	SP3, SPARKS2, COMA and HHSEARCH	Modellerv9.8	–	ModFOLDclust2 (for single template models), ModFOLD6_rank QA and DISO-clust3
Bhagheerath-H Jayaram <i>et al.</i> (2014)	None	pGentheader, ffas spark-x, HHSearch	substitution scoring matrix based on chemical properties of amino acids	Modeller for targets having templates and bhagheerath <i>ab initio</i> modelling method for stretches without template	Quantum mechanics (PM6) based loop bond angle optimization, Scwrl4 and Sander module of AMBER10	PROTSAV
Swiss model Biasini <i>et al.</i> (2014)	–	BLAST and HHblits (User can also provide template)	BLAST and HHblits	ProMod-II and MODELLER	–	QMEAN

^aThis is based on threading alignments and convergence of I-TASSER's structural assembly refinement simulations.^bThese methods incorporate profile information, predicted secondary structure, solvent accessibility, amino acid physico-chemical properties.

Table 6 Webservers for protein structure validation

Webserver	Tools integrated	Output	Threshold	URL
ProTSAV (Singh <i>et al.</i> , 2016)	Procheck ProSA-web ERRAT Verify 3D dDFire Naccess MolProbity D2N ProQ PSN-QA	Quality assessment plot ProTSAV score	Green area: 0-2 Å. Yellow area: 2-5 Å. Orange area: 5-8 Å. Red area: > 8 Å. Range: 0 to 1 0: Good quality 1: low quality	http://www.scfbio-iitd.res.in/software/teomics/protsav.jsp
PSVS (Bhattacharya <i>et al.</i> , 2007)	PDBStat RPF DSSP PROCHECKG-factors MolProbity Verify 3D Prosall PDB validation software	Ramachandran plot summary PROCHECKG-factors MolProbity clashscore RMS deviation for bond lengths RMS deviation for bond angles	> 85% in conformationally allowed region Z-score > -3 Z-score > -3 < 0.02 Å < 4°	http://psvs-1_5-dev.nesg.org/
Molprobity (Davis <i>et al.</i> , 2007)	None	Clash score C β deviations > 0.25 Å Bad bonds Bad angles Ramachandran favored Favored rotamers	Range : 0 (worst) – 100 (best) 0 0 < 0.1% > 98% > 98%	http://molprobity.biochem.duke.edu/

Structure Refinement

Structure refinement is based on the quality report. The goal of refinement is to improve the predicted model in a way that it has accurate side chain orientations and high stereo-chemical quality so that it might get closer to the native structure (Carlsen and Røgen, 2015). The following are two strategies used for structure refinement:

Potential energy minimization and molecular dynamics

Potential energy minimization (PEM) techniques are used to eliminate high energies in the predicted model and achieve local minima that is closer to native structure. Molecular dynamics (MD) is comparatively computationally expensive among all refinement techniques, but it is the most prevalent strategy for refinement (Feig, 2017). FG-MD (Zhang *et al.*, 2011) is one of the examples of MD-based refinement server.

Structure optimization

Poor quality regions reported by quality checks can specifically be targeted and optimized iteratively to improve the model. Secondary structure elements and hydrogen bonding networks can be optimized. One example of refinement method using structure optimization approach and energy minimization is 3DRefine (Bhattacharya and Cheng, 2013). Editing target-template alignment and rebuilding a model iteratively can often improve the built model. Structure optimization after the MD is more useful for improving overall quality of the fold (Feig, 2017).

Success Stories of Protein Structure Modelling

Protein structure prediction methods are successfully bridging up the sequence-structure gaps by complementing experimental methods. Few examples of successful model predictions are given below to show how these methods are revolutionizing the research in structural biology.

Homology Modelling Using a Single Template

A structural model for DORN1 (plant purinoreceptor) was predicted through homology modelling (Tanaka *et al.*, 2016) and used to predict the extracellular ATP (eATP) binding pocket in DORN1, and key residues involved in binding. Blind docking and different tools for ligand pocket prediction were then used to predict the eATP binding pocket in the predicted model. Predicted binding pocket was then validated by site directed mutagenesis of key residues. Results showed the accuracy of modelled ligand binding pocket.

Homology Modelling Using Multiple Templates

The structure for integrin $\alpha\beta6$ was predicted through homology modelling by using multiple templates (Sowmya *et al.*, 2014). The specific interaction of integrin $\alpha\beta6$ and uPAR is known to be associated with cancer progression and was studied using the predicted structure. The crystal structure of the extracellular domains was subsequently determined (PDB ID: 4UM9) and the RMSD between the predicted and experimental structures is $<1 \text{ \AA}$, demonstrating the accuracy of the predicted model.

Ab initio Modelling

The structure of CT296 protein was predicted by *ab initio* modelling through the I-TASSER server (Kemege *et al.*, 2011). CT296 is present in *Chlamydia trachomatis*, which is a bacterial pathogen responsible for non-heritable blindness. It was previously suggested that CT296 shows similarity to the ferric uptake regulator and has DNA binding motifs. The predicted model was not consistent with this information. The XRC of CT296 was then determined at a resolution of $1.8 \text{ \AA}$, supporting CT296 not having functional similarity to the ferric uptake regulator and also having DNA binding sites. When compared, the predicted model had significant similarity to the XRC (RMSD: $2.72 \text{ \AA}$).

Threading-Based Homology Modelling

The β_1 -adrenergic receptor (β_1 -AR) belongs to GPCR superfamily and is known to play an important role in cardiovascular function and disease, serving as a drug target for the management of cardiovascular diseases. Through threading-based homology modelling approach, a 3D model for β_1 -AR was constructed that was further utilized in docking four agonists (Carmoterol, Dobutamine, Isoprenaline and Salbutamol) to study the binding modes of β_1 -AR (Ul-Haq *et al.*, 2015). The human β_1 -AR structural model assisted the authors to understand its binding mechanism, for future targeted structure-based drug molecule development.

Let's Start to Model:

Given the poor representation of membrane proteins in the PDB and the complications involved in building a model for protein with multiple transmembrane segments, the example chosen to illustrate the steps involved in building a 3D model of a protein is that of a membrane protein.

Case Study

GPCRs represent the largest family of membrane proteins and are targets for almost half of the drugs (Tautermann, 2014). There are a total 130,314 GPCR protein sequences from multiple organisms, including *Homo sapiens*, available in UniProt (as on August 2017), but only 436 crystal structures of GPCRs are present in GPCRdb, which includes 44 unique crystallized receptors and 43 unique ligand-receptor complexes, indicating a large sequence structure gap. GPCRs consist of seven transmembrane (Schmiedeberg *et al.*, 2007) helices that are connected through three extracellular loops (ECLs) and three intracellular loops (ICLs) with extracellular N-terminus and intracellular C-terminus (Clark, 2017). Short wave sensitive opsin is one of the GPCR whose structure is predicted through homology modelling, with consideration of motifs and patterns, and in this case study is used as an example.

Obtaining Target Sequence

Download the amino acid sequence of the target, i.e., short wave sensitive opsin (OPSB_BOVIN) in FASTA Format from UniProt (UniProt ID: P51490, see Relevant Website section).

Sequence

- Use TMSEG (Bernhofer *et al.*, 2016) to predict transmembrane regions of OPSB_BOVIN. Enter query sequence in FASTA format and select transmembrane helices option.

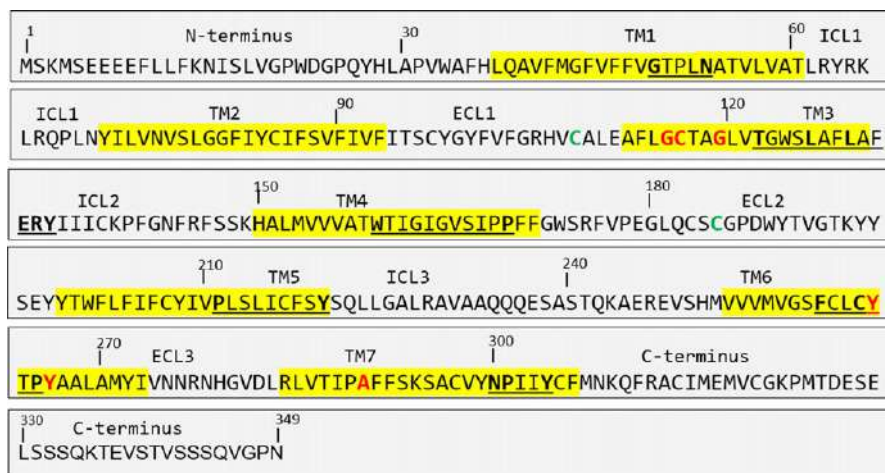
- b. Use Predictprotein (see Relevant Websites section) to confirm transmembrane regions and to predict secondary structure of non-membrane segments.
- c. Motifs and patterns can be predicted by number of available webserver and their knowledge can also be gained from available literature. Each TM of GPCR contains at least one motif that can be very helpful for generating optimal alignment (Van Der Kant and Vriend, 2014). Table 7 shows the motifs retrieved from the literature. Patterns/motifs predicted from different webserver are shown in Table 8 (only strong matches are shown).
- d. Extracellular part of GPCRs are involved in ligand recognition and binding. The ligand binding pockets are extremely diverse among GPCR members in terms of shape and features for recognition of varied ligands (Katritch *et al.*, 2012). Despite the diverse ligands, residues 3.32, 3.33, 3.36, 6.48, 6.51 and 7.39 (numbered according to Ballesteros–Weinstein numbering (Ballesteros and Weinstein, 1995)) are known to make consensus interactions with ligands among class A GPCRs. Residues from other transmembrane regions tend to interact with specific ligands to a varied extent. Therefore, TM3, TM6 and TM7 serve as a consensus cradle for the ligand binding pocket. Alteration of residues in these specific topological positions leads to ligand specificity in different receptors (Venkatakrishnan *et al.*, 2013). OPSB_BOVIN sequence is represented in Fig. 2 with predicted TM regions, motifs, consensus ligand binding pocket and cysteine residues involved in the formation of disulphide bridges.

Table 7 Motifs of GPCR sequences retrieved from the literature

TM	Motif
ECL2	Disulphide bridge between TM3 (Cys ^{3.25}) (Ballesteros and Weinstein, 1995) and ECL2 (Venkatakrishnan <i>et al.</i> , 2013)
1	GXXXN (de March <i>et al.</i> , 2015)
2	NLXXD (Miguel <i>et al.</i> , 2017)
3	DRY (Deupi <i>et al.</i> , 2007) SX ₃ LX ₂ IX ₂ D(E,H)RY (Costanzi, 2012)
4	WX _{8,9} P (Costanzi, 2012)
5	FX ₂ PX ₇ Y (Costanzi, 2012)
6	CWXP (Miguel <i>et al.</i> , 2017) FX ₂ CW(Y,F)XP (Costanzi, 2012)
7	NPXXY (Sato, 2016)

Table 8 Predicted motifs and patterns

Webserver used	Predicted pattern/feature	Sequence/location
ScanProsite	Disulphide bridge	108 & 185
	G_Protein_Recep_F1_1	VTGwSLAFLAFERYIil (121–137)
	Opsin Retinal Binding Site	IpaFFSKSACvyNPiY (288–304)
MotifScan	G_PROTEIN_RECEP_F1_1	121–137
	OPSIN Visual pigments	288–304
	7tm_1	52–304

**Fig. 2** OPSB_BOVIN sequence features. Predicted TM regions (highlighted in yellow), motifs (bold text) and patterns (underlined), cysteine residues involved in disulphide bridge formation (green) and consensus ligand binding cradle of class A GPCRs (red) are shown.

Template Identification and Selection

- Run BLAST against Protein databank to retrieve the potential templates. OPSB_BOVIN has 44% sequence identity and 92% query coverage with bovine rhodopsin (PDB ID: 3OAX) according to BLAST results. Due to high sequence identity, query coverage and completeness of structure, 3OAX can be selected as a template, but there is another structure of bovine rhodopsin deposited in PDB (PDB ID: 1U19) with a resolution of 2.2 Å while 3OAX has a resolution of 2.6 Å. Due to the better resolution than 3OAX, 1U19 is selected as template for modelling OPSB_BOVIN.
- Download the PDB file of 1U19 from Protein databank (see Relevant Websites section).

Target Template Alignment

- Before alignment, the PDB file must be edited, if needed. For example, if the template has multiple chains and one of the chains is selected to serve as a template, then the rest of the chains must be deleted by use of the PDB visualizer such as UCSF Chimera. The PDB file 1U19 has two chains A and B, since we only need chain A, chain B is deleted by selecting the chain B in UCSF Chimera and then deleting it. If the PDB file contains any protein that is co-crystallized with the template protein, then it must also be removed. Additionally, all non-standard residues must be removed except the ligands (if required) and metals ions as they are important for folding specially for small proteins. Non-standard residues are present in the form of HETATM record in the PDB file and can be deleted using any text editor. There is one modified residue (HETATM) in the PDB file of 1U19 (line number 871–873), labelled as X in the sequence, it should also be removed. There might be some missing residues in the template, so the structure derived sequence of the template will be used which can be retrieved by clicking sequence in the tools tab in UCSF Chimera and saving it into a separate file. The gaps will be inserted in place of missing residues before alignment. There are no missing residues in 1U19.
- The MSA webserver, T-COFFEE, is used to align OPSB_BOVIN and 1U19. The membrane protein option is selected before submitting the sequence.
- Knowledge of motifs, patterns and ligand binding cradle gained during sequence analysis step can now be used to optimize the alignment between query and template. Motifs in OPSB_BOVIN are well aligned to 1U19. Due to the high sequence identity, there are no insertions and deletions in the predicted transmembrane regions. The only gaps present in alignment are within the N- and C-termini, which can be ignored, as these regions are usually not conserved in membrane proteins. According to the alignment between 1U19 and OPSB_BOVIN (Fig. 3), the last five residues of OPSB_BOVIN do not have a corresponding residue from the template, and thus these residues will not be modelled due to lack of matching template residues. These residues can be removed from alignment before model building.

Model Building, the Manual Way

Build the model of OPSB_BOVIN using Modeller, by following the tutorial provided (see Relevant Website section). It is better to build multiple models (say 5) and then select the final model on the basis of the lowest objective function (Fig. 4).

OPSB_BOVIN	MSKMSEEEFFLLFKNISL--VGPWDGPQYHLAPVWAFHLQAVFMGFVFVFGTPLNATVLVATLRYRKLQRPLNYILVNV	78
1U19	MNGTEGPNFYVPFSNKTGVVRSPFEAPQYYLAEPWQFSMLAAYMFLIMLGFPINFLTLYVTQVHKLRTPLNIIILLNLA	80
	* . . : :: * * : . * : : ** : * * : * : * : : : * * * . * : : : ** * : : : * : :	
OPSB_BOVIN	LGFIYICFVSFIVFITSCYGYFVGRHVCALAEFLGCTAGLVGTWGLAFLAFERYIICKPFGNFRFSSKHALMVVAT	158
1U19	VADLFMVVGGFTTTLTYSLHGYFVFGPTGCNLGGFFATLGGETALWSLVLAIERVYVVVKPMSNFRFGENHAIMGVAFT	160
	: : : : : : . . . : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : *	
OPSB_BOVIN	WTIGIGVSIPPFEGWSRFVPEGLQCSCGPDWYTVGTYKYYSEYVTWFLFIFCYIVPLSLICFSYSQLLGALRAVAQAQQES	238
1U19	WVMALACAAPPLVGVWSRYIPEGMQCSGIDYYTPHEETNNESEFVIYMFVVHFIIPLIVIFFCYQQLVFTVKEAAAQQQES	240
	* : : : : : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : *	
OPSB_BOVIN	ASTQKAEREVSVMVVMVGSEFCLCYTPYAALAMYIVNNRNHGVDLRLVTIPAFFSKSACVYNPIIYCFMNMKQFRACIMEM	318
1U19	ATTQKAEKEVTRMVIIMVIAFLICWLPYAGVAFYIFTHQGSDFGPIMFTIPAFFAKTSAVYNPVIYIMNMKQFRNCMVTT	320
	* : : : : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : * : : *	
OPSB_BOVIN	VC-GK-PMTDESELSSSQKTEVSTVSSSDVGPR	349
1U19	LCCGNPLGDDEASTTVSKTETSQVAPA-----	348
	* * * * * . . * * * *	

Fig. 3 Alignment of Bovine rhodopsin (1U19: A) and OPSB_BOVIN sequence given as an input to Modeller. TM regions of 1U19 and predicted TM regions of OPSB_BOVIN are highlighted in yellow with disulphide bridge forming cysteine residues (green), motifs of both sequences (bold text), patterns (underlined), ligand binding cradle of OPSB_BOVIN (red), ligand binding pocket of 1U19 (blue) and residues without a template are highlighted in red. Disulphide bridge is indicated by a dashed line.

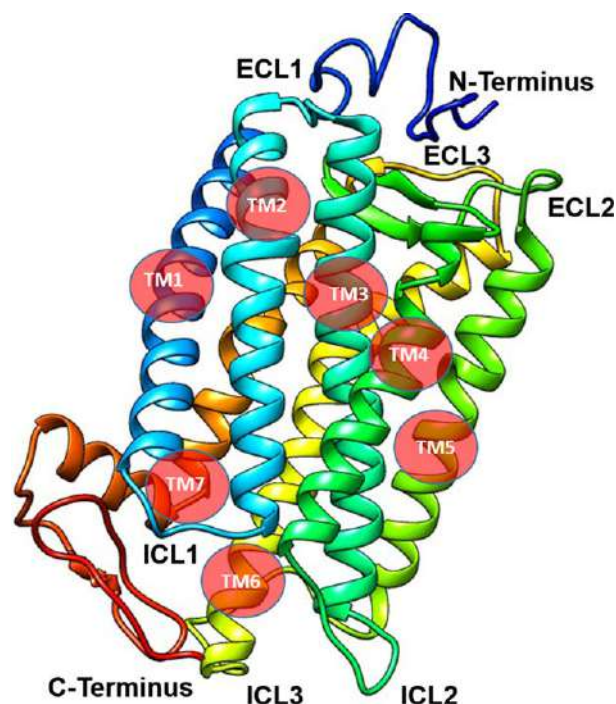


Fig. 4 OPSB_BOVIN model, from manual model building.

Structure Validation

- Use PSVS server to validate the structure. PROCHECK program present in PSVS server reports the conformationally allowed regions of the residues. High quality X-ray structures have at least 85% of the residues in conformationally allowed regions (Sowmya *et al.*, 2014). For the predicted model, 99.7% of the residues are in conformationally allowed regions as reported by PROCHECK, including 88.3% of which are favorable and 11.4% are in allowed regions. RMSD for bond lengths and bond angles is 0.019 Å and 2.1° respectively, which should be lower than 0.02 Å and 4° for a good quality structure.
- Closeness of the template and query determines the accuracy of the predicted structure, which is often measured by either Root Mean Square Deviation (RMSD) or global distance test (GDT_TS). The greater the similarity between the template and the query, the higher is the resolution. Sequence similarity over 50% and $\text{RMSD} \leq 1$ Å corresponds to high resolution structure. If similarity is between 30% and 50% and $\text{RMSD} \leq 1.5$ Å the resolution is medium (Carlsen and Røgen, 2015). The RMSD between the OPSB_BOVIN and 1U19 is 0.267 Å. RMSD can be retrieved by superimposing the predicted structure and template structure in UCSF Chimera (PDB visualizer).
- Use the Molprobiy server to look for favored and poor rotamers which are the low energy conformations of side-chain dihedral angles. The correct assignment of rotamers to each amino acid is essential for a thermodynamically stable structure (Bhuyan and Gao, 2011). The quality of predicted side chain can be analyzed by identifying the favored rotamers in the 3D structure of a protein (Harder *et al.*, 2010).
- Use the ProTSAV server that gives a normalized average *Z-score* based on ten diversified structure validation programs. A *Z-score* closer to 0 represents a good quality structure. The manual model for OPSB_BOVIN has a *Z-score* of 0.5846 (Table 9).
- Literature: It is observed that TM3 and ECL2 are connected together by a disulphide bridge (Fig. 5), which is the characteristic of majority of class A GPCRs (Venkatakrishnan *et al.*, 2013).

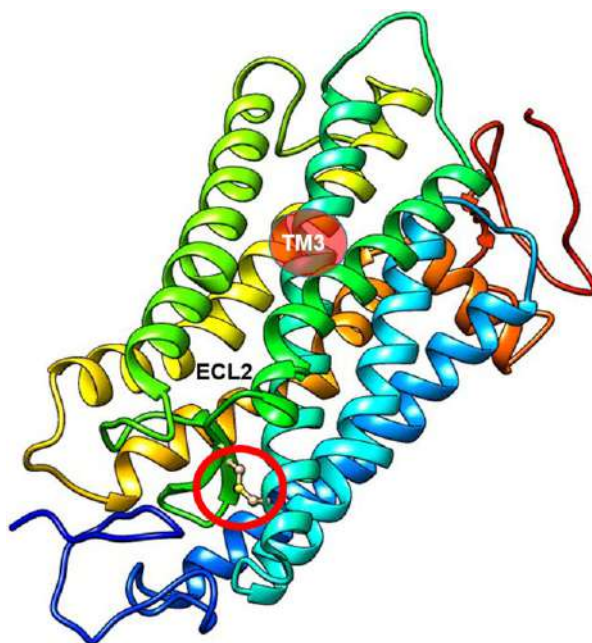
Automated Structure Prediction

Plenty of online servers are available that take a target sequence as an input and pass it to a pipeline of various programs to predict the 3D structure of the target protein. Almost all such webserver have a user-friendly interface. It is important that one should select the server carefully to have a good model. Top ranked servers (according to CASP ranking) were used in this case study to predict the model of OPSB_BOVIN, namely I-TASSER, RaptorX, and InfFOLD, among others. It should be noted that in this case, we allowed the servers to select the best template(s). The comparisons of structure validation results for predicted structures according to the steps in Section Structure Validation by use of the automated servers and the manual modelling is shown in Table 9.

Table 9 Comparison of structure validation results of manual modelling and automated modelling

Description		Manual model	RaptorX	IntFOLD	I-TASSER
Ramachandran plot (PSVS)	Favored (%)	88.3	92.5	85.4	82.5
	Allowed (%)	11.4	6.9	13.0	17.5
	Outliers (%)	0.3	0.6	1.6	0.0
RMSD for bond lengths (Å) (PSVS)		0.019	0.020	0.019	0.014
RMSD for bond angles (PSVS)		2.2°	3.0°	2.2°	2.3°
Favored rotamers (%) (Molprobit)		91.06	87.75	87.75	78.15
ProTSAV score (ProTSAV)		0.5846	0.5907	0.6042	0.5593
RMSD (Å)		0.260	1.605	0.203	1.177
Template(s) used		1U19 (Bovine Rhodopsin)	4ZWJ (Human rhodopsin)	2G87 (Bathorhodopsin)	Multiple ^a

^a4ZWJ (Human Rhodopsin); 1GZM (Bovine Rhodopsin); 1U19 (Bovine Rhodopsin); 2KS9 (Substance P receptor); 2G87 (Bathorhodopsin).

**Fig. 5** Disulphide bridge between ECL2 and TM3.

For all the predicted models by the automated servers and the manual modelling, more than 90% of the residues were in the allowed regions according to PROCHECK embedded in PSVS server. The RMSD for bond lengths and bond angles from native structures for all the models were good but there were not enough favored rotamers, which should be more than 98%, suggesting that the side chains need refinement. In the manually predicted model, the favored rotamers were comparatively better than the automated models.

N- and C-Termini

The automated RaptorX server predicted long N- and C-termini as compared to the other predictions, it is because of the long indels present in the alignment of both termini between the OPSB_BOVIN and the template 4ZWJ selected by the server. Only few residues in N-terminus are modelled on the basis of template (4ZWJ) as seen in the alignment (Fig. 6).

Helical Bundle

According to both the manual and automated models, the predicted TM1, TM2 and TM4 are modelled in the same manner and have the same positions in each predicted model. There are slight differences in the start and end positions of TM3 between the manually predicted model and the I-TASSER model. There was a variation in the prediction of TM5, where I-TASSER and RaptorX had predicted a long TM5 as compared to the manual and IntFOLD server predictions. This is because the template (4ZWJ) chosen by the automated servers had a long TM5 (Fig. 6). The start and the end of TM6 was the same across all the predictions, except for I-TASSER which had only

OPSB_BOVIN 4ZWJ	-----MSKMSEEEFLFKNISLVGPWDGPQYHLAPVWAFHLQAV FSNATGVVRS-----PFEYPQYYLAEPWQFSMLAA
OPSB_BOVIN 4ZWJ	FMGFVFFVGTPLNATVLVATLRYRKLRLQPLNYILNVSLGGFIYCFVSF YMFLLIVLGFPINFLTLVYTVQHKKLRTPLNYILLNLAVADLFMVLGGFT
OPSB_BOVIN 4ZWJ	IVFITSCYGYFVFGRHVCALEAFGLCTAGLVTGWSLAFALAFERYIIICKP STLYTSLHGYFVFGPTGCNLQGGFATLGGEIALWSLVVLAIERVWVCKP
OPSB_BOVIN 4ZWJ	FGNFRFSSKHALMWWATWTIGIGVSIPFFGWSRFVPEGLQCSCGPDWY MSNFRFGENHAIMGVAFTWVMALACAAPLAGWSRYIPEGLQCSCGIDYY
OPSB_BOVIN 4ZWJ	TVGTKYYSEYTWFLFIFCYIVPLSLICFSYSQLLGALRAVAAQQQESAS TLKPEVNNESFVIYMFVVHFTIPMIIFFCYGQLVFTVKEAAAQQQESAT
OPSB_BOVIN 4ZWJ	TQKAEREVSHMVMVVGFSFCLCYTPYAALAMYIVNNRHGVDLRLVTIPA TQKAEKEVTRMVIIVIAFLICWVPYASVAFYIFTHQGSCFGPIFMTIPA
OPSB_BOVIN 4ZWJ	FFSKSACVYNPIIYCFMNKQFRACIMEMVCG-----KPMTDESEL FFAKSAAIYNPVIYIMMNKQFRNCMLTTICCGKNVIFKKVS-----
OPSB_BOVIN 4ZWJ	SSSQKTEVSTVSSSQVGPN -----

Fig. 6 Alignment of OPSB_BOVIN with 4ZWJ used by RaptorX to generate the 3D model of OPSB_BOVIN. N-terminus, C-terminus and TM5 of both OPSB_BOVIN and 4ZWJ are highlighted in yellow.

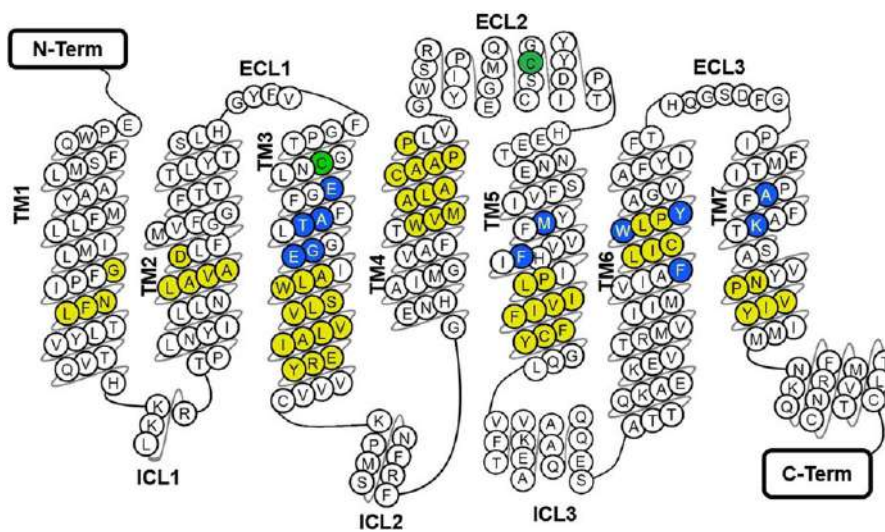


Fig. 7 Snake plot for 1U19, showing motifs/patterns (yellow), cysteine residues involved in disulphide bonding (green) and residues in ligand binding site (blue).

one residue difference. The start position of TM7 was the same for all the predictions, but there was a disagreement in the end position. TM boundaries of each predicted model can be seen clearly through snake plots in Figs. 7–9. Snake plots for 1U19 and OPSB_BOVIN are downloaded from GPCRdb (Munk *et al.*, 2016) and edited according to the predicted models.

Intracellular Loops

The structure of the predicted intracellular loop was nearly similar in all the models, except intracellular loop 3 (ICL3 in Fig. 9) of models predicted by RaptorX and I-TASSER, where in these models the TM5 was longer, and thus the ICL3 was shorter, compared to the other two predicted models (Fig. 10).

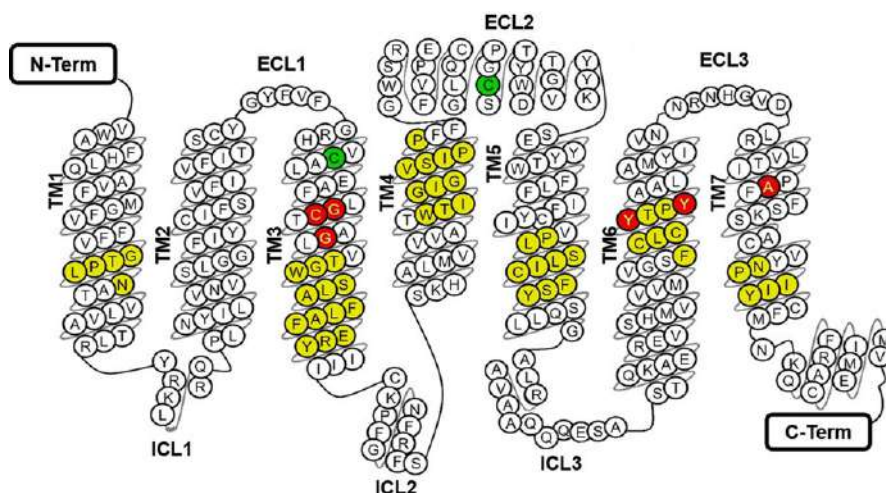


Fig. 8 Snake plot for manual model, showing motifs/patterns (yellow), cysteine residues involved in disulphide bonding (green) and residues in ligand binding site (red).

Disulphide Bridges

All of the predicted models had a single disulphide bridge between cysteine residues at positions 108 and 185. Thus, all models were consistent in building a viable disulphide bridge between the correct cysteine residues.

Secondary Structure Elements

All predicted models contained 7TM helices. Two β -strands were present in ECL2 of all the predicted models. According to the I-TASSER prediction, two more β -strands were present in the N-terminus. One small helix was present in ECL2 in a model predicted by RaptorX.

Automated Versus Manual Modelling

Automated modelling is convenient when high sequence identity exists between the query and the template, but for low sequence identity cases, manual modelling is preferable for the following reasons:

- Alignment editing is required in most of the cases. For instance, in a model predicted by RaptorX, the C-terminus and N-terminus can be improved by adjusting the gaps in alignment. Many automated servers do not allow alignment editing.
- A careful study of template is required for its selection (Section Template Selection). We can see from the case study that the manual *versus* automated selection of the template affected the model; for example, resulting difference in the length of predicted TM5 by RaptorX and I-TASSER automated servers due to the different templates. Most of the automated servers focus on sequence identity between the target and template for selection; only some incorporate additional features, such as secondary structure and solvent accessibility, among others. Some automated servers do allow the user to specify the templates, but this might be risky as it can often result in a bad model if the template selection criteria are not well applied (Fasnacht *et al.*, 2014).
- Many servers do not allow use of multiple templates.
- Servers normally take one to several days for predicting the structure depending upon the load on the server, but through manual prediction, models can be built quickly as compared to automated servers.

Despite the above facts, automated servers do provide models that can give clues about the structure of a protein and in many instances, can serve as a first guess, before careful manual model building.

Conclusion

Enormous advancement has been witnessed in the area of protein structure prediction methods over the past decade. The nor-epinephrine transporter (NET), a drug target for various mood related and behavioral disorders, such as depression, is a notable success story in the application of such prediction methods. There was not a single good resolution atomic structure available for NET.

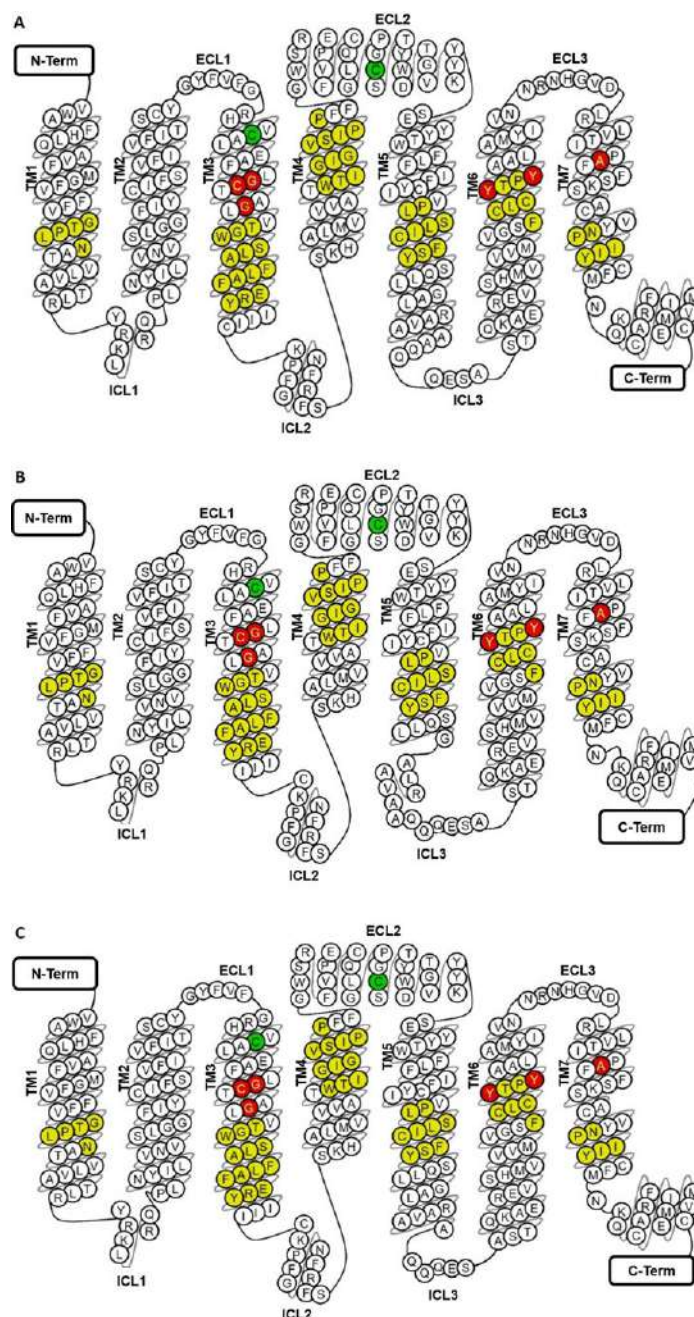


Fig. 9 Snake plots for RaptorX (A), IntFold (B) and I-TASSER (C) models, showing motifs/patterns (yellow), cysteine residues involved in disulphide bonding (green) and residues in ligand binding site (red).

Computational model of NET was constructed by use of the X-ray structure of leucine transporter LeuT from *Aquifex aeolicus* as a template. The built model was then used for virtual ligand screening against the binding site, which resulted in the identification of 10 already tested NET inhibitors and five novel ligands. Predicted ligands were then validated experimentally to be the potential inhibitors of NET (Schlessinger *et al.*, 2011). Despite the continuous progress in this field, there are cases where the proteins can be difficult to model. For example, *HpUreI*, a proton gated urea channel from *Helicobacter pylori* known to be involved in gastric infection, was one of the targets for CASP round 10. The X-ray structure of *HpUreI* showed a novel fold which was not predicted with accuracy by any of the template based modelling approach; and was predicted with little accuracy by the template free approach I-TASSER, but less than one third of the residues were in proximity to the corresponding X-ray structure (Kryshtafovych *et al.*, 2014). With continuous improvement in protein structure prediction methods, these challenges are hoped to be addressed. For example, numerous methods are now available for modelling membrane proteins; RosettaMP (Alford *et al.*, 2015) is one such method.

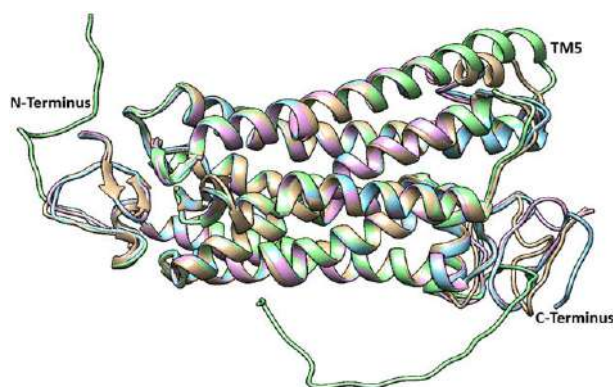


Fig. 10 Superimposition of model structures from IntFOLD (blue), I-TASSER (brown), RaptorX (green) and manual mode (pink).

Table 10 Useful resources for protein structure prediction and modelling

Step	Resource	URL
Sequence analysis resources	HHMM TOP (Topology prediction)	http://www.enzim.hu/hmmtop/
	TMHMM (Topology prediction)	http://www.cbs.dtu.dk/services/TMHMM/
	Consurf (Conserved buried/exposed residue prediction)	http://consurf.tau.ac.il/2016/
	Interpro (Domain prediction)	https://www.ebi.ac.uk/interpro/
	PRINTS (Protein families database)	http://130.88.97.239/PRINTS/index.php
	MOTIFSCAN (Motif prediction)	http://myhits.isb-sib.ch/cgi-bin/motif_scan
	SCANProsite (Motif prediction)	http://prosite.expasy.org/scanprosite/
	JPRED (Secondary structure prediction)	http://www.compbio.dundee.ac.uk/jpred/
Template identification	PSiPred (Secondary structure prediction)	http://bioinf.cs.ucl.ac.uk/psipred/
	BLAST	https://blast.ncbi.nlm.nih.gov/Blast.cgi
	Psi-BLAST	https://www.ebi.ac.uk/Tools/sss/psiblast/
	LOMETS	http://zhanglab.ccmb.med.umich.edu/LOMETS/
	MUSTER	http://zhanglab.ccmb.med.umich.edu/MUSTER/
	COMPASS	http://prodata.swmed.edu/compass/compass.php
	HMMER	http://hmmer.org/
	RAPTORX	http://RaptorX.uchicago.edu/
Multiple sequence alignment/profile construction	MAFT	https://www.ebi.ac.uk/Tools/msa/mafft/
	CLUSTAL Omega	https://www.ebi.ac.uk/Tools/msa/clustalo/
	MUSCLE	https://www.ebi.ac.uk/Tools/msa/muscle/
	T-COFFEE	http://tcoffee.crg.cat/
	SALIGN	https://modbase.compbio.ucsf.edu/salign/
	Promals3D	http://prodata.swmed.edu/promals3d/promals3d.php
3D model building	Homology Modelling	See Tables 4, 5
	PHYRE2	https://modbase.compbio.ucsf.edu/modweb/
	(Fold recognition)	
	pGenThreader	http://www.chemogenomix.com/chemogenomix/GenThreader.html
	(Fold recognition)	
	ORION	http://www.dsimb.inserm.fr/ORION/
	(Fold recognition)	
	Robetta	http://rosetta.bakerlab.org/
	(<i>Ab initio</i>)	
	QUARK	https://zhanglab.ccmb.med.umich.edu/QUARK/
Structure refinement	(<i>Ab initio</i>)	
	GalaxyRefine	http://galaxy.seoklab.org/cgi-bin/submit.cgi?type=REFINE
	ModRefiner	http://zhanglab.ccmb.med.umich.edu/ModRefiner/
	3DRefine	http://sysbio.rnet.missouri.edu/3Drefine/
	REFINEpro	http://sysbio.rnet.missouri.edu/REFINEpro/
	PREFMD	http://feiglab.org/prefmd
Structure validation	FG-MD	http://zhanglab.ccmb.med.umich.edu/FG-MD/
	See Table 6	
PDB visualizers	Pymol	https://www.pymol.org/
	RASMOL	http://www.openrasmol.org/
	Discovery studio	http://accelrys.com/products/collaborative-science/biovia-discovery-studio/
	SwissPDB viewer	http://spdbv.vital-it.ch/
	Chimera	https://www.cgl.ucsf.edu/chimera/

Useful Resources

A list of resources for each step of the protein structure modelling protocol is listed in [Table 10](#).

See also: Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Biomolecular Structures: Prediction, Identification and Analyses. Cloud-Based Molecular Modeling Systems. Computational Protein Engineering Approaches for Effective Design of New Molecules. Drug Repurposing and Multi-Target Therapies. In Silico Identification of Novel Inhibitors. Natural Language Processing Approaches in Bioinformatics. Protein Design. Small Molecule Drug Design. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Study of The Variability of The Native Protein Structure

References

- Abagyan, R., Totrov, M., Kuznetsov, D., 1994. ICM – A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. *Journal of Computational Chemistry* 15, 488–506.
- Alford, R.F., Leman, J.K., Weitzner, B.D., *et al.*, 2015. An integrated framework advancing membrane protein modeling and design. *PLOS Computational Biology* 11, e1004398.
- Anfinsen, C.B., Redfield, R.R., Choate, W.L., Page, J., Carroll, W.R., 1954. Studies on the gross structure, cross-linkages, and terminal sequences in ribonuclease. *Journal of Biological Chemistry* 207, 201–210.
- Bahar, I., Jernigan, R.L., Dill, K.A., 2017. *Protein Actions: Principles and Modeling*. New York: Garland Science.
- Ballesteros, J., Weinstein, H., 1995. Integrated methods for modeling G-protein coupled receptors. *Methods in Neuroscience* 25, 366–428.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28, 235–242.
- Bernhofer, M., Kloppmann, E., Reeb, J., Rost, B., 2016. TMSEG: Novel prediction of transmembrane helices. *Proteins: Structure, Function, and Bioinformatics* 84, 1706–1716.
- Bhattacharya, D., Cheng, J., 2013. 3Drefine: Consistent protein structure refinement by optimizing hydrogen bonding network and atomic-level energy minimization. *Proteins: Structure, Function, and Bioinformatics* 81, 119–131.
- Bhattacharya, A., Tejero, R., Montelione, G.T., 2007. Evaluating protein structures determined by structural genomics consortia. *Proteins: Structure, Function, and Bioinformatics* 66, 778–795.
- Bhuyan, M.S.I., Gao, X., 2011. A protein-dependent side-chain rotamer library. *BMC Bioinformatics* 12, S10.
- Biasini, M., Bienert, S., Waterhouse, A., *et al.*, 2014. SWISS-MODEL: Modelling protein tertiary and quaternary structure using evolutionary information. *Nucleic Acids Research* 42, W252–W259.
- Carlsen, M., Røgen, P., 2015. Protein structure refinement by optimization. *Proteins: Structure, Function, and Bioinformatics* 83, 1616–1624.
- Chivian, D., Kim, D.E., Malmström, L., *et al.*, 2003. Automated prediction of CASP-5 structures using the Robetta server. *Proteins: Structure, Function, and Bioinformatics* 53, 524–533.
- Clark, T., 2017. G-Protein coupled receptors: Answers from simulations. *Beilstein Journal of Organic Chemistry* 13, 1071–1078.
- Costanzi, S., 2012. Homology modeling of class A G protein-coupled receptors. *Methods in Molecular Biology* 857, 259–279.
- Cozza, G., 2017. The development of CK2 inhibitors: From traditional pharmacology to in silico rational drug design. *Pharmaceuticals (Basel)* 10, 26.
- Davis, I.W., Leaver-Fay, A., Chen, V.B., *et al.*, 2007. MolProbity: All-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Research* 35, W375–W383.
- Deupi, X., Dolker, N., LUZ Lopez-Rodriguez, M., *et al.*, 2007. Structural models of class AG protein-coupled receptors as a tool for drug design: Insights on transmembrane bundle plasticity. *Current Topics in Medicinal Chemistry* 7, 991–998.
- Fasnacht, M., Butenhof, K., Goupil-Lamy, A., *et al.*, 2014. Automated antibody structure prediction using Accelrys tools: Results and best practices. *Proteins: Structure, Function, and Bioinformatics* 82, 1583–1598.
- Feig, M., 2016. Local protein structure refinement via molecular dynamics simulations with locPREMD. *Journal of Chemical Information and Modeling* 56, 1304–1312.
- Feig, M., 2017. Computational protein structure refinement: Almost there, yet still so far to go. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 7, e1307.
- Fiser, A., 2010. Template-based protein structure modeling. *Methods in Molecular Biology* 673, 73–94.
- Greer, J., 1981. Comparative model-building of the mammalian serine proteases. *Journal of Molecular Biology* 153, 1027–1042.
- Guex, N., Peitsch, M.C., Schwede, T., 2009. Automated comparative protein structure modeling with SWISS-MODEL and Swiss-PdbViewer: A historical perspective. *Electrophoresis* 30 Suppl 1, S162–S73.
- Guo, J.-T., Ellrott, K., Xu, Y., 2008. A historical perspective of template-based protein structure prediction. *Methods in Molecular Biology* 413, 3–42.
- Harder, T., Boomsma, W., Paluszewski, M., *et al.*, 2010. Beyond rotamers: A generative, probabilistic model of side chains in proteins. *BMC Bioinformatics* 11, 306.
- Illergård, K., Ardell, D.H., Elofsson, A., 2009. Structure is three to ten times more conserved than sequence – a study of structural response in protein cores. *Proteins: Structure, Function, and Bioinformatics* 77, 499–508.
- Isom, D.G., Dohlman, H.G., 2015. Buried ionizable networks are an ancient hallmark of G protein-coupled receptor activation. *Proceedings of the National Academy of Sciences* 112, 5702–5707.
- Jayaram, B., Dhingra, P., Mishra, A., *et al.*, 2014. Bhageerath-H: A homology/ab initio hybrid server for predicting tertiary structures of monomeric soluble proteins. *BMC Bioinformatics* 15, S7.
- Katritch, V., Cherezov, V., Stevens, R.C., 2012. Diversity and modularity of G protein-coupled receptor structures. *Trends in Pharmacological Sciences* 33, 17–27.
- Kelley, L.A., Sternberg, M.J., 2009. Protein structure prediction on the Web: A case study using the Phyre server. *Nature Protocols* 4, 363–371.
- Kemege, K.E., Hickey, J.M., Lovell, S., *et al.*, 2011. Ab initio structural modeling of and experimental validation for Chlamydia trachomatis protein CT296 reveal structural similarity to Fe(II) 2-oxoglutarate-dependent enzymes. *Journal of Bacteriology* 193, 6517–6528.
- Khor, B.Y., Tye, G.J., Lim, T.S., Choong, Y.S., 2015. General overview on structure prediction of twilight-zone proteins. *Theoretical Biology and Medical Modelling* 12, 15.
- Kozma, D., Simon, I., Tusnády, G.E., 2012. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. *Nucleic Acids Research* 41, D524–D529.
- Ko, J., Park, H., Heo, L., Seok, C., 2012. GalaxyWEB server for protein structure prediction and refinement. *Nucleic Acids Research* 40 (Webserver issue), W294–W297.
- Kryshtafovych, A., Moult, J., Bales, P., *et al.*, 2014. Challenging the state of the art in protein structure prediction: Highlights of experimental target structures for the 10th critical assessment of techniques for protein structure prediction experiment CASP10. *Proteins: Structure, Function, and Bioinformatics* 82, 26–42.
- Lacapere, J.J., 2017. *Membrane Protein Structure and Function Characterization: Methods and Protocols*. New York: Springer.
- Lee, J., Freddolino, P.L., Zhang, Y., 2017. Ab Initio protein structure prediction. In: J. RIGDEN, D. (Ed.), *From Protein Structure to Function With Bioinformatics*. Dordrecht, Netherlands: Springer.
- Levitt, M., 1992. Accurate modeling of protein conformation by automatic segment matching. *Journal of Molecular Biology* 226, 507–533.
- Marchler-Bauer, A., Zheng, C., Chitsaz, F., *et al.*, 2012. CDD: Conserved domains and protein three-dimensional structure. *Nucleic Acids Research* 41, D348–D352.

- de March, C.A., Kim, S.K., Antonczak, S., *et al.*, 2015. G protein-coupled odorant receptors: From sequence to structure. *Protein Science* 24, 1543–1548.
- Markstein, P., Xu, Y. 2006. *Computational Systems Bioinformatics*. London: Imperial College Press.
- Mcguffin, L.J., Atkins, J.D., Salehe, B.R., Shuid, A.N., Roche, D.B., 2015. IrfOLD: An integrated server for modelling protein structures and functions from amino acid sequences. *Nucleic Acids Research* 43, W169–W173.
- Mcguffin, L.J., Buenavista, M.T., Roche, D.B., 2013. The ModFOLD4 server for the quality assessment of 3D protein models. *Nucleic Acids Research* 41, W368–W372.
- Miguel, R.N., Sanders, J., Furmaniak, J., Smith, B.R., 2017. Structure and activation of the TSH receptor transmembrane domain. *Autoimmunity Highlights* 8, 2.
- Molecular Operating Environment (MOE), 2017. 2013.08; Chemical Computing Group ULC. 1010 Sherbooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7.
- Moult, J., Fidelis, K., Krysztofowicz, A., Schwede, T., Tramontano, A., 2016. Critical assessment of methods of protein structure prediction: Progress and new directions in round XI. *Proteins: Structure, Function, and Bioinformatics* 84, 4–14.
- Munk, C., Isberg, V., Mordalski, S., *et al.*, 2016. GPCRdb: The G protein-coupled receptor database – An introduction. *British Journal of Pharmacology* 173, 2195–2207.
- Pearson, W.R., 2013. An introduction to sequence similarity (“homology”) searching. *Current Protocols in Bioinformatics*. 3.1.1–3.1.8.
- Peng, J., Xu, J., 2011. RaptorX: Exploiting structure information for protein alignment by statistical inference. *Proteins: Structure, Function, and Bioinformatics* 79, 161–171.
- Petrey, D., Xiang, Z., Tang, C.L., *et al.*, 2003. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. *Proteins: Structure, Function, and Bioinformatics* 53, 430–435.
- Šali, A., Blundell, T.L., 1993. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology* 234, 779–815.
- Sato, T., Kawasaki, T., Mine, S., Matsumura, H., 2016. Functional role of the C-terminal amphipathic helix 8 of olfactory receptors and other G protein-coupled receptors. *International Journal of Molecular Sciences* 17, 1930.
- Scheraga H.A., 2014. Simulations of the folding of proteins: A historical perspective In: LIWO, A. (ed.) *Computational Methods to Study the Structure and Dynamics of Biomolecules and Biomolecular Processes: From Bioinformatics to Molecular Quantum Mechanics*. Berlin, Heidelberg: Springer.
- Schlessinger, A., Geier, E., Fan, H., *et al.*, 2011. Structure-based discovery of prescription drugs that interact with the norepinephrine transporter, NET. *Proceedings of the National Academy of Sciences*, 108. pp. 15810–15815.
- Schmiedberg, K., Shirokova, E., Weber, H.-P., *et al.*, 2007. Structural determinants of odorant recognition by the human olfactory receptors OR1A1 and OR1A2. *Journal of Structural Biology* 159, 400–412.
- Schwede, T., 2013. Protein modeling: What happened to the “protein structure gap”? *Structure* 21, 1531–1540.
- Singh, A., Kaushik, R., Mishra, A., Shanker, A., Jayaram, B., 2016. ProTSAV: A protein tertiary structure analysis and validation server. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* 1864, 11–19.
- Song, Y., Dimaio, F., Wang, R.Y.-R., *et al.*, 2013. High-resolution comparative modeling with RosettaCM. *Structure*, 21. pp. 1735–1742.
- Sousounis, K., Haney, C.E., Cao, J., Sunchu, B., Tsonis, P.A., 2012. Conservation of the three-dimensional structure in non-homologous or unrelated proteins. *Human Genomics* 6, 10.
- Sowmya, G., Khan, J.M., Anand, S., *et al.*, 2014. A site for direct integrin $\alpha v \beta 6$ · uPAR interaction from structural modelling and docking. *Journal of Structural Biology* 185, 327–335.
- Starr, C., Taggart, R., Evers, C., Starr, L., 2011. *Biology: The Unity and Diversity of Life*. 12th ed. Belmont: Brooks/Cole.
- Tanaka, K., Cao, Y., Cho, S.-H., Xu, D., Stacey, G., 2016. Computational analysis of the ligand binding site of the extracellular ATP receptor, DORN1. *PLOS ONE* 11, e0161894.
- Tautermann, C.S., 2014. GPCR structures in drug design, emerging opportunities with new structures. *Bioorganic & Medicinal Chemistry Letters* 24, 4073–4079.
- Taylor W., *Patterns in Protein Sequence and Structure* 1992, Berlin, Heidelberg, Springer-Verlag.
- Thangudu, R.R., Manoharan, M., Srinivasan, N., *et al.*, 2008. Analysis on conservation of disulphide bonds and their structural features in homologous protein domain families. *BMC Structural Biology* 8, 55.
- Ul-Haq, Z., Saeed, M., Halim, S.A., Khan, W., 2015. 3D structure prediction of human $\beta 1$ -adrenergic receptor via threading-based homology modeling for implications in structure-based drug designing. *PLOS ONE* 10, e0122223.
- Van Der Kant, R., Vriend, G., 2014. Alpha-bulges in G protein-coupled receptors. *International Journal of Molecular Sciences* 15, 7841–7864.
- Venkatakrishnan, A., Deupi, X., Lebon, G., *et al.*, 2013. Molecular signatures of G-protein-coupled receptors. *Nature* 494, 185–194.
- Voet, D., Voet, J.G., Pratt, C.W., 2016. *Fundamentals of biochemistry: Life at the molecular level*, Life at the Molecular level, fifth ed. Wiley.
- Vriend, G., 1990. WHAT IF: A molecular modeling and drug design program. *Journal of Molecular Graphics* 8, 52–56.
- Webb, B., Sali, A., 2014. Protein structure modeling with MODELLER. *Methods in Molecular Biology* 1137, 1–15.
- Wooley, J.C., Ye, Y., 2007. A historical perspective and overview of protein structure prediction In: XU, Y., XU, D. & LIANG, J. (eds.) *Computational Methods for Protein Structure Prediction and Modeling: vol. 1: Basic Characterization*. New York: Springer.
- Yang, J., Yan, R., Roy, A., *et al.*, 2015. The I-TASSER Suite: Protein structure and function prediction. *Nature Methods* 12, 7–8.
- Zhang, J., Liang, Y., Zhang, Y., 2011. Atomic-level protein structure refinement using fragment-guided molecular dynamics conformation sampling. *Structure* 19, 1784–1795.
- Zhang, Y., 2008. I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics* 9, 40.

Further Reading

- Childers, M.C., Daggett, V., 2017. Insights from molecular dynamics simulations for computational protein design. *Molecular Systems Design & Engineering* 2, 9–33.
- Bujnicki, J.M. (Ed.), 2008. *Prediction of Protein Structures, Functions, and Interactions*. John Wiley & Sons.
- Dakal, T.C., Kumar, R., Ramotar, D., 2017. Structural modeling of human organic cation transporters. *Computational Biology and Chemistry* 68, 153–163.
- Eswar, N., Eramian, D., Webb, B., *et al.*, 2008. Protein structure modeling with MODELLER. *Structural Proteomics: High-Throughput Methods*. 145–159.
- França, T.C.C., 2015. Homology modeling: An important tool for the drug discovery. *Journal of Biomolecular Structure and Dynamics* 33 (8), 1780–1793.
- Mackenzie, C.O., Grigoryan, G., 2017. Protein structural motifs in prediction and design. *Current Opinion in Structural Biology* 44, 161–167.
- Platform, A.E., 2005. *Introduction to the Discovery Studio Visualizer*. San Diego, California, USA: Accelrys Software Inc.
- Rangwala, H., Karypis, G. (Eds.), 2011. *Introduction to Protein Structure Prediction: Methods and Algorithms* 18. John Wiley & Sons.
- Schmidt, T., Bergner, A., Schwede, T., 2014. Modelling three-dimensional protein structures for applications in drug design. *Drug Discovery Today* 19, 890–897.
- Sheehan, D., O’Sullivan, S., 2011. Online homology modelling as a means of bridging the sequence-structure gap. *Bioengineered Bugs* 2 (6), 299–305.
- Stansfeld, P.J., 2017. Computational studies of membrane proteins: From sequence to structure to simulation. *Current Opinion in Structural Biology* 45, 133–141.
- Venko, K., Choudhury, A.R., Novič, M., 2017. Computational approaches for revealing the structure of membrane transporters: Case study on Bili-translocase. *Computational and Structural Biotechnology Journal* 15, 232–242.

Relevant Websites

<https://salilab.org/modeller/tutorial/Modeller>.

<https://www.predictprotein.org>

Predictprotein.

<https://www.rcsb.org/pdb/explore/explore.do?structureId=1u19>

Protein Data Bank.

<http://www.UniProt.org/UniProt/P51490>

Uniprot.

Biographical Sketch



Amara Jabeen is a PhD student in the Department of Chemistry and Biomolecular sciences at Macquarie University. She is working on functional annotation of odorant receptors using bioinformatics approaches. She is predicting the structures of olfactory receptors through computational methods. She has worked on genome wide analysis of T6SS proteins in *Acidovorax avenae* subsp. *avenae*, *Acidovorax avenae* subsp. *citruilli* AACOO1 and *Acidovorax avenae* subsp. *avenae* ATCC19860 through bioinformatics approaches. She has also worked on structure prediction of Lipase H and bioinformatics analysis of Lipase H and Extracellular matrix protein 1 (ECM1).



Abidali Mohamedali is Lecturer at Macquarie University in the Faculty of Science and Engineering. He completed his PhD in 2010 studying, using proteomics, the effects of single mutations in mouse model on brain development in the autism spectrum disorder, Rett syndrome. He then returned to Macquarie University undertaking a post-doc to study the biology of metastasis in colorectal cancer using proteomics and to determine novel therapeutic targets and develop novel technologies to examine the plasma proteome.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, United States, Singapore and Australia, as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books, as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology, as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.

Structure-Based Drug Design Workflow

Ari Hardianto, Macquarie University, Sydney, NSW, Australia and Universitas Padjadjaran, Jatinangor, Indonesia

Muhammad Yusuf, Universitas Padjadjaran, Jatinangor, Indonesia

Fei Liu and Shoba Ranganathan, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Drug discovery in pharmaceutical companies was dominated by the process of generating synthetic compounds through combinatorial chemistry where many trials and errors occur, with a hope obtaining new active compounds (Li, 2013). It was a long way with a lot of efforts which spends time, money, and labor. For an illustration, pharmaceutical companies typically required 10–15 years with expenditure around US\$800 to US\$1.8 billion to release a drug to the market (Macalino *et al.*, 2015). With advances in high-throughput crystallography, computer hardware, and molecular modeling software, a new approach was introduced in the drug development by employing the three-dimensional (3D) structure of the biological target to design active molecules rationally. This method is called structure-based drug design (SBDD). SBDD started to flourish after the discovery of an HIV protease inhibitor (saquinavir) which exploited the 3D structure of the target macromolecule in its rational design (Li, 2013). Since then, SBDD has become an established tool which accelerates the process of drug development. Nowadays, SBDD is an integral part of the drug discovery and development in many modern pharmaceutical companies (Macalino *et al.*, 2015; Muegge *et al.*, 2017). Fig. 1 summarizes the workflow for SBDD which includes target selection, target structure evaluation, binding site identification, molecular docking and scoring, and lead compound optimization to generate new drug.

Steps Involved in the Process of SBDD

Drug Target Selection

Selection of drug targets is the first step in SBDD. It is an essential part because SBDD relies on the 3D of target macromolecules in the process of drug design. Drug targets can be enzymes, receptors, ion channels, transport proteins, DNA/RNA and the ribosome, or targets of monoclonal antibodies (Imming *et al.*, 2006). Drug targets are identified and validated through direct biochemical approaches, genetic interaction, computational inference techniques or combination of these methods. These methods are comprehensively reviewed by Schenone (Schenone *et al.*, 2013). A thorough literature survey can assists in the selection of drug targets associated with the small molecules of interest.

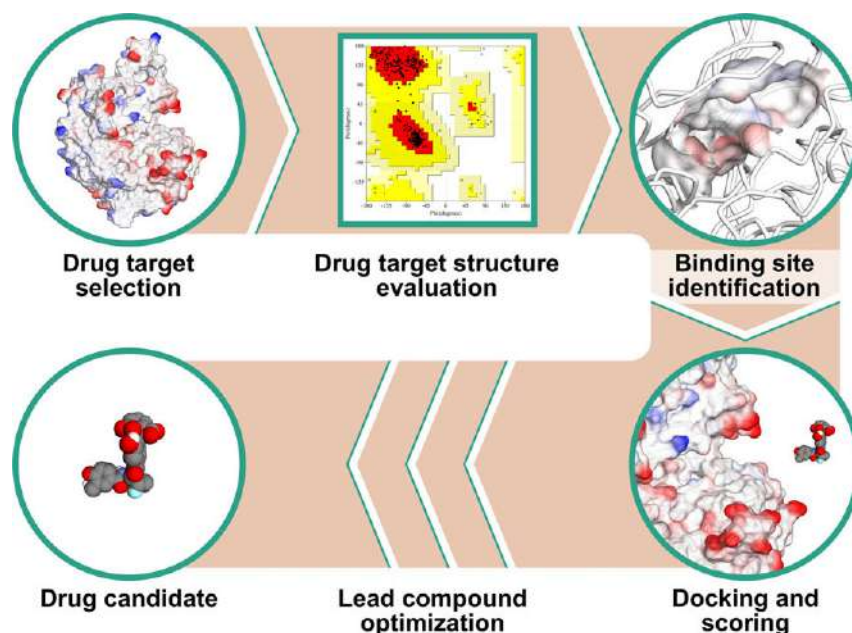


Fig. 1 Workflow of structure-based drug design (SBDD).

Evaluation of Drug Target Structures

Once the drug target has been identified, the next step is obtaining its 3D-structure from the Protein Data Bank (PDB; see “Relevant Websites section”) or Cambridge Crystallographic Data Center (see “Relevant Websites section”). Experimental methods for determining the drug target structures are X-ray crystallography, nuclear magnetic resonance (NMR), and cryo-electron microscopy (EM) approaches. Crystal structures are the widely used source of structural information for drug design (Anderson, 2003; Petsko and Ringe, 2010). The quality of crystal structures is usually assessed from their resolutions, R factors, coordinate errors, temperature factors, and chemical correctness (Anderson, 2003). These parameters are usually reported in the ‘REMARK’ part of the PDB file. Crystal structure with resolution up to 2.5 Å is acceptable for drug design. The R factor and R_{free} , which reflect the correlation between the model and experiment, should be below 25 and 28%, respectively. The coordinate error must be between 0.2 and 0.3 Å. Temperature factors of atoms, which indicate an atomic position error, are less than the average temperature factor for the molecule. Stereochemical deviations of bond angles and lengths are less than 3° and 0.015 Å, respectively, whereas planar atoms should be less than 0.015 Å out of the plane. Ramachandran plot is then used to assess the chemical correctness of backbone ϕ and ψ angles where the percentage of at least 90% in the most favored regions are preferred (Laskowski *et al.*, 2001).

However, if experimental structures are unavailable, comparative modeling, threading, or *ab initio* protein modeling is used to generate 3D-models for drug targets. The same evaluation parameters are applied to the generated models. A great number of programs are available for protein structure validation, for instance, PROCHECK (Laskowski *et al.*, 2001) and WHATIF (Rodriguez *et al.*, 1998).

Binding Site Identification

The binding site is usually a hollow part or pocket of the drug target where the small molecule or ligand binds (Macalino *et al.*, 2015). The features of the binding site that are important for ligand binding include potential H-bond donors and acceptors, unique hydrophobic surfaces and molecular surface size (Anderson, 2003; Macalino *et al.*, 2015). The knowledge of ligand binding site and its features are essential in SBDD and can be extracted from experimental structures and mutation studies. However, the information of ligand binding site may be unavailable. Fortunately, *in silico* approaches are there that can be used to predict potential binding sites. These approaches are implemented in many programs or online servers. A list of binding site prediction servers and programs are summarized in Table 1.

Molecular Docking

Once the drug target and its binding site are identified, structures of small molecules (ligands) are modified *in silico*. The modified ligands are usually subjected to molecular docking to predict binding modes and affinities of ligands (Forli *et al.*, 2016). Molecular docking methodologies (listed in Table 2) consist of searching algorithms and scoring functions. These searching algorithms explore possible conformations of small molecules while interacting with residues in the binding site. The scoring functions then rank the possible conformations and are crucial for SBDD. The scoring functions are classified into force field-based, empirical-based, and knowledge-based functions (Macalino *et al.*, 2015). Force field-based functions use physical atomic contacts in the target-ligand complex to compute binding affinity, where solvation and entropy terms can be evaluated. In empirical-based functions, the calculation is performed by fitting fewer energy terms, such hydrophobic and electrostatic interactions, to experimental binding affinity data. Knowledge-based functions employ a statistical method to derive binding energy from a training set of protein-ligand.

Table 1 Popular servers and programs for binding site prediction

Program/ server	Availability	Prediction method	URL
Cavicator	Free as standalone program	Analysis of grid-based geometric (Gao and Skolnick, 2013)	http://cssb.biology.gatech.edu/Cavicator
PocketFinder	Free as a PyMOL plugin	Shape descriptors (Weisel <i>et al.</i> , 2007)	http://www.modeling.leeds.ac.uk/pocketfinder/
fpocket	Free as a standalone program	Alpha sphere theory (Guilloux <i>et al.</i> , 2009)	http://fpocket.sourceforge.net/
ConCavity	Free as a standalone program and webserver	Evolutionary sequence conservation and 3D structure (Capra <i>et al.</i> , 2009)	http://compbio.cs.princeton.edu/concavity/
ProBis	Free as a web server	Local structural alignments (Konc and Janežič, 2010)	http://probis.cmm.ki.si/index.php
3DLigandSite	Free as a web server	Structures similarity (Wass <i>et al.</i> , 2010)	http://www.sbg.bio.ic.ac.uk/~3dligandsite/
eFindSite	Free as a standalone program and webserver	Meta-threading, machine learning, and auxiliary ligands (Brylinski and Feinstein, 2013)	http://brylinski.cct.lsu.edu/efindsite
ConSurf	Free as a web server	Surface-mapping (Ashkenazy <i>et al.</i> , 2016)	http://consurf.tau.ac.il/2016/

Table 2 Available molecular docking packages

Program	Availability	Website
AutoDock	Free	http://autodock.scripps.edu/
AutoDock Vina	Free	http://vina.scripps.edu/
DOCK	Free	http://dock.compbio.ucsf.edu/
ICM	Commercial	http://www.molsoft.com/
cDocker	Commercial	http://accelrys.com
LigandFit	Commercial	http://accelrys.com
LibDock	Commercial	http://accelrys.com
FLIPDock	Free	http://flipdock.scripps.edu/

Table 3 ADMET prediction programs and webserver

Program/webserver	Availability	Websites
SwissADME	Free as a webserver	http://www.swissadme.ch/
admetSAR	Free as a standalone software	http://lmmd.ecust.edu.cn/admetSar1/
FAF-Drugs4	Free as a webserver	http://fafdrugs4.mti.univ-paris-diderot.fr/
ALOGPS	Free as a webserver	http://www.vcclab.org/lab/alogps/
QikProp	Commercial	https://www.schrodinger.com/
ADMET Predictor	Commercial	http://www.simulations-plus.com/software/admet-property-prediction-qsar/
ADMET and predictive toxicology	Commercial	http://accelrys.com/products/collaborative-science/biovia-discovery-studio/qsar-admet-and-predictive-toxicology.html

Lead Compound Optimization

The lead compound suggested by molecular docking is then subjected to a cycle of optimization. The compound must be synthesized and assessed by both *in vivo* and *in vitro* experiments. After these experiments conducted, the step returns to the computational stages. Another analog can be developed and estimated for their binding affinity through molecular docking. Since molecular docking roughly estimates the binding affinity of binding for ligands to the target macromolecule, another sophisticated approach like molecular dynamics or free energy perturbation may be required to obtain more realistic conformations of new analogs or binding affinity predictions (Forli *et al.*, 2016). Subsequently, the analogs are synthesized in the laboratory and assayed for their binding affinities. These computational and experiment are iteratively performed until the desired binding affinity of the ligand is reached (Macalino *et al.*, 2015).

At the final step of lead compound optimization, analyses of absorption, distribution, metabolism, excretion, and toxicity (ADMET) are highly required (Macalino *et al.*, 2015). After being administered to the body, drugs may need to absorb properly by crossing various barriers like a gastrointestinal tract to enter the bloodstream. Through bloodstream, drugs are then distributed to their effector sites in muscles or organs where they are metabolized and then excreted. Few of the drug metabolisms may result in toxic compounds which can be hazardous. There are many web servers and programs available for predicting ADMET, for instance, QikProp, SwissADME, and ADMET Predictor. Thus, ADMET estimation can be optionally performed before synthesis step for prioritizing compounds suggested by molecular docking (Table 3).

Practical Applications of SBDD in Pharmaceutical Industry

HIV Anti-Viral Drug

SBDD has engendered the pharmaceutical industry with many successes for instance drugs for HIV, influenza, and cancer. The discovery of an antiviral drug for HIV was the leading example and the milestone for the establishment of SBDD. Advancements in crystallography technique lead to the determination of structure for HIV protease in the late 1980s. Saquinavir, the first anti-HIV drug was developed using the crystal structure of the protease. Since then the further development assisted by SBDD have yielded nine FDA-approved anti-HIV drugs the latest one is darunavir (Li, 2013).

Influenza Anti-Viral Drug

SBDD has also contributed to the discovery of influenza antiviral drugs. By using the crystal structure of neuraminidase (NA), zanamivir was designed from 2-deoxy-2,3-didehydro-N-acetylneuraminic acid (DANA) as the lead compound. The knowledge

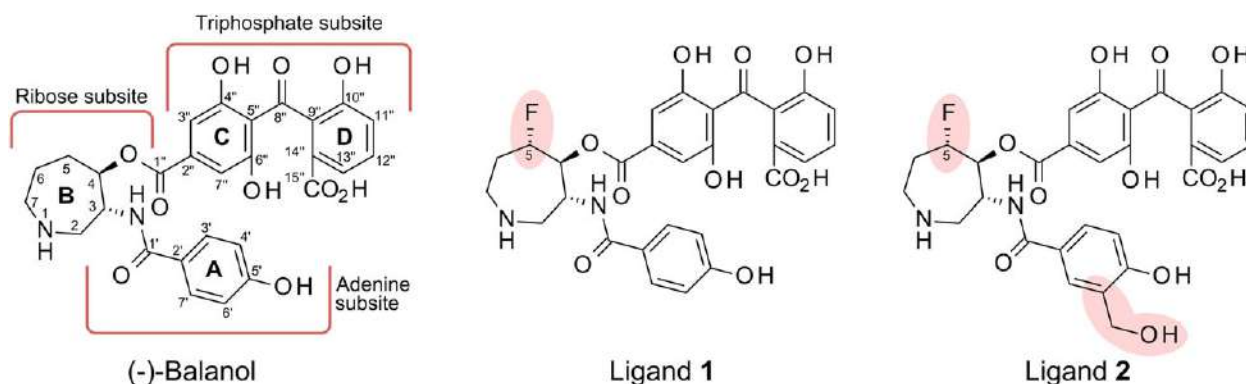


Fig. 2 Structures of balanol and its fluorinated analogs. The balanol structure has been divided into three moieties that correspond to different subsites based on its structural overlay with ATP.

extracted from the crystal structures of NA in complex with DANA and zanamivir also helped in the design of oseltamivir which has a more improved bioavailability than zanamivir. After the discovery of oseltamivir (OTV), the overlaid crystal structure of a furanose isomer with DANA in complex with N9 subtype revealed that the four functional groups (side chains) in both complexes had similar orientation and interactions in the binding site. Therefore, it was concluded that the central ring is not an absolute factor in the development of potential NA inhibitors. Instead, it is the relative position of the functional groups in the active site which is important (Babu *et al.*, 2000). For this reason, cyclopentane-based compounds (different central ring from the previous compounds) were developed and as a result peramivir was discovered. In 2006, FDA approved the usage of peramivir in the injectable formulation (Itzstein and Thomson, 2009).

Case Study

To illustrate SBDD, protein kinase C ϵ (PKC ϵ) is employed as an example. Unlike other PKC isoforms that possess ambiguous roles in cancer development, PKC ϵ shows clear oncogenic activities and is implicated in invasion and metastasis of cancer cells, such as human breast, glioma, and renal carcinoma (Garg *et al.*, 2014). A fungal natural product, called balanol, was discovered to competitively inhibit PKC ϵ targeting its ATP site (Kulanthaivel *et al.*, 1993). However, the compound also unselectively inhibits other PKC isoforms. Interestingly, a stereospecific fluorination on balanol yielded an analog (hereafter mentioned as ligand 1) with enhanced activity and selectivity to PKC ϵ (Patel *et al.*, 2017).

The structure of balanol is an ATP mimic and divided into benzamide (ring A), azepane (ring B), and benzophenone (ring C and D) moieties. It completely occupies the ATP binding site (Taylor *et al.*, 2004). In particular, the benzamide moiety replaces the adenine subsite, whereas the azepane and the benzophenone moieties fill the ribose and the triphosphate subsites, respectively (Fig. 2).

Here, SBDD is utilized to design another new analog of balanol with improved binding activity to PKC ϵ . Since the experimental structure of PKC ϵ is unavailable, its 3D model was generated using homology modeling. To build the model, two crystal structures, a mouse protein kinase A (PDB ID 1BX6) and a human PKC θ (3TXO), were used as templates. The details about homology modeling method are presented in another article of this book.

The following part is dedicated to demonstrate how SBDD, using a molecular docking approach assists in predicting binding affinity of ligand 1 and another analog of fluorinated balanol (ligand 2). A widely used program AutoDock (Forli *et al.*, 2016) is used in this case study. Autodock is freely available and can be downloaded from “see Relevant Websites section”. Discovery Studio Visualizer and AutoDockTools are utilized in the molecular docking preparation and analysis. Discovery Studio Visualizer is freely distributed at “see Relevant Websites section”, whereas AutoDockTools (ADT) which is part of MGLTools can be downloaded from “see Relevant Websites section”. Coordinate files of the ligands and PKC ϵ are provided as SBDD.zip.

Ligand Preparation

1. Create the coordinate files for ligands. The coordinates for the two ligands used in this case study were generated using homology modeling, which included a ligand mapping from the template 1BX6 to the resulting homology model of PKC ϵ . The balanol structure was modified in Discovery Studio Visualizer to yield two fluorinated balanol analogs as ligands for PKC ϵ . The first ligand is balanol with a fluorine atom on the position of C(5S) which is provided as ligand_1.pdb. It was prepared by the following steps (Fig. 3): open the balanol structure (bal.pdb) in Discovery Studio Visualizer program, select C5, on the menu bar choose ‘Chemistry $\rightarrow$ Hydrogens $\rightarrow$ Add’, then select the hydrogen atom on the position of C(5S), on the menu bar select ‘Chemistry $\rightarrow$ Element $\rightarrow$ F’ (Fig. 3). Ligand 2 is the further modification of the first ligand with an additional hydroxymethyl on the benzamide moiety. The coordinate file of this ligand is provided as ligand_2.pdb. Both ligands have negative charges on

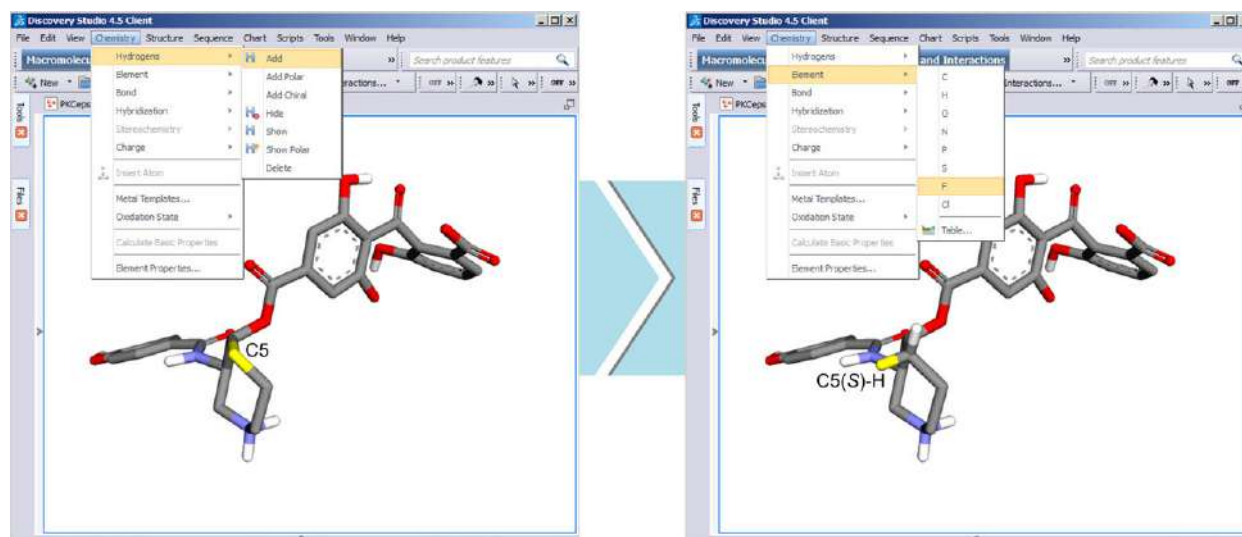


Fig. 3 Steps in incorporating the fluorine substituent to the azepane ring of balanol. Fluorine was added to the position C5(S).

the phenolic C6''OH and carboxylic C15''O₂H and a positive charge on the secondary amine N1 as suggested by our previous investigation (Hardianto *et al.*, 2017).

2. Load the coordinate file of the first ligand (ligand_1.pdb) to ADT. On ADT menu, choose 'Ligand → Input → Open' and select 'PDB files' for the file type, then browse to the folder where ligand_1.pdb is located, select the file and click 'Open'. ADT will add charges if required, merge nonpolar hydrogens, and assign suitable atom types. The ligand will appear in the viewer window. Click 'OK' on the popup window.
3. Prepare the PDBQT file for the first ligand (ligand_1.pdb). To do this, select 'Ligand → Torsion Tree → Detect Root' which will identify the center of the torsion tree. For this study case, we use the default torsional degrees of freedom. However, if a customization of such degree of freedom is needed, select 'Ligand → Torsion Tree → Choose Torsions'. Rotatable bonds are denoted in green and rigid ones are in the red, whereas non-rotatable bonds that are potentially rotatable are in magenta. Click the bonds to change the rotation flexibility. Click 'Done' to complete this step. Select 'Ligand → Output → Save as PDBQT', and then choose 'Save' to write the file ligand_1.pdbqt.

Receptor Preparation

Prepare the coordinate file of the receptor, i.e., PKCε. The coordinate file for receptor was generated using a homology modeling approach and is provided as PKCepsilon.pdb. Open the file by clicking 'Grid → Macromolecule → Open', and choose 'PDB files (.pdb)' for the file type. Click on file PKCepsilon.pdb and click 'Open'. A popup window will appear, click 'Save' to write the file PKCepsilon.pdbqt (Fig. 4).

Grid Maps Preparation and Generation

1. Prepare a grid parameter for generating the atomic affinity maps. Select 'Grid → Macromolecule → Choose' and choose 'PKCepsilon' in the popup window. Select 'Grid → Set Map Types → Choose Ligand' and choose 'bal_1c'. Set the grid box of the search space by selecting 'Grid → Grid Box', which prompts a window for defining the center and size of the box. Choose 'Center → Center on Ligand' to define a minimal box. To adjust the size and center of the box, drag left the thumbwheels to decrease or right to increase the value. Choose 'File → Close Saving Current'. Select 'Output → Save GPF' to save the grid parameter file as 'ligand_1.gpf'.
2. Run AutoGrid from the command line. Open a terminal window and go to the directory where the coordinate files and grid parameter file (ligand_1.gpf) are located. Make sure executable file of AutoGrid (autogrid4) is in the same directory. Type the following command:

```
autogrid4.exe -p ligand_1.gpf -l ligand_1.glg
```

Molecular Docking

1. Create a docking parameter file. Select the receptor PDBQT file by clicking 'Docking → Macromolecule → Set Rigid Filename' and choose 'PKCepsilon.pdbqt' in the directory where it is located. For the ligand, click 'Docking → Ligand → Choose' and

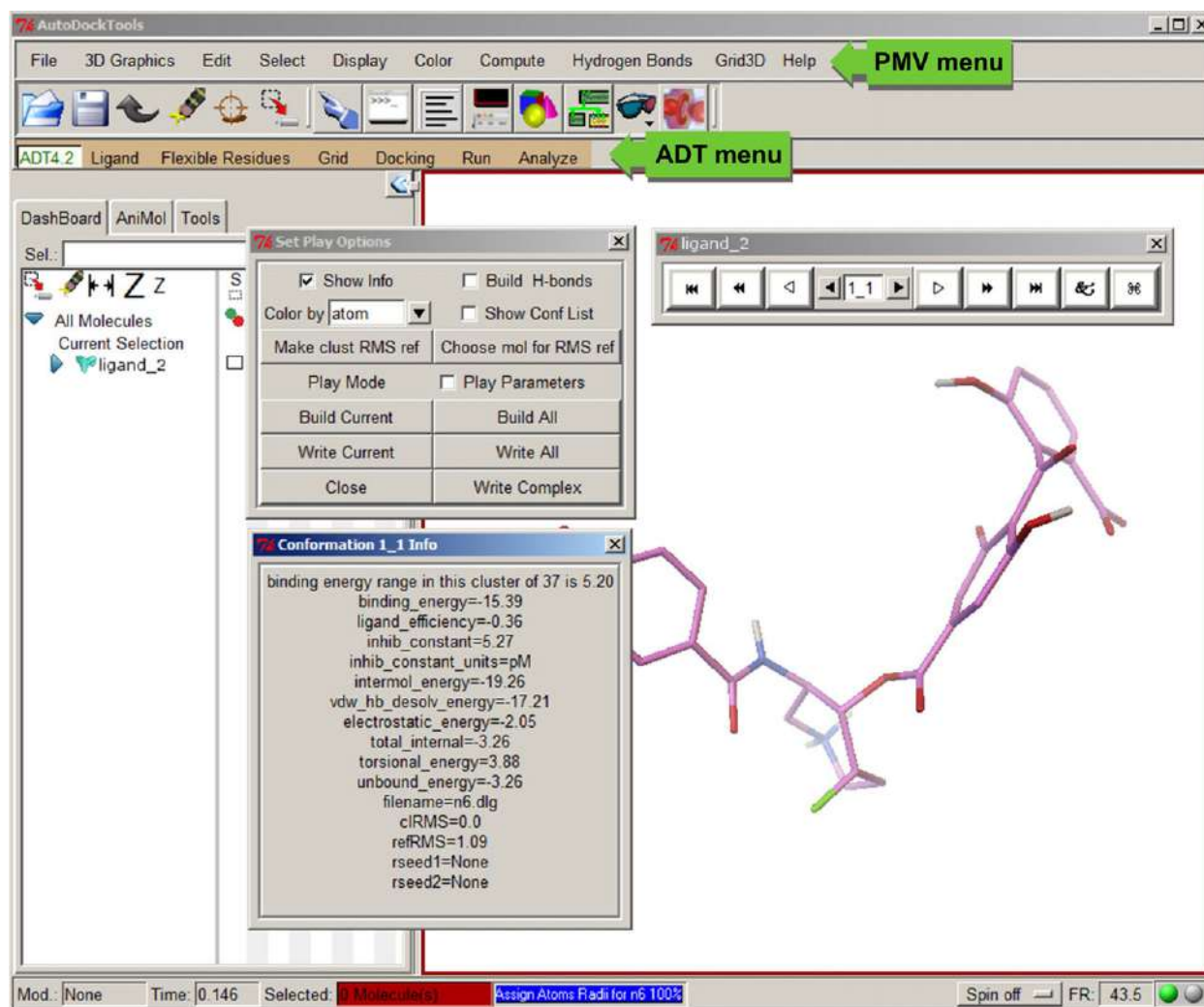


Fig. 4 Display of AutoDockTools.

select 'ligand_1', and accept the default ligand docking parameters. To set searching parameters, click 'Docking → Search Parameters → Genetic Algorithm Parameters', change 'Number of GA Runs' to 50, and accept the rest default parameters. Then click 'Docking → Docking Parameters' and use default parameters. Select 'Docking → Output → Lamarckian GA' to write the docking parameter file as 'ligand_1.dpf'.

- Run AutoDock from the command line. In a terminal window, go to the directory containing the coordinate files and docking parameter (ligand_1.dpf). By assuming AutoDock (autodock4.exe) is in the same directory with the coordinate files and docking parameter, type the following command
`autodock4.exe -p ligand_1.dpf -l ligand_1.dlg`

Analysis

- Visualize the molecular docking results. Load the results by selecting 'Analyze → Dockings → Open' and choose the ligand_1.dlg file. Select 'Analyze → Conformations → Play Ranked By Energy' to visualize the docking result. Click '&'-like button to open 'Set Play Options' panel, then check 'Show info' to display 'conformation info' dialog box which contains predicted binding energy and its components. When displaying the conformation 1, on the Set Play Options dialog box, click 'Build Current' and save the conformation of docked ligand 1 as conf1_ligand_1.pdbqt in your current working directory.
- Visualize docking pose cluster based on binding energy values. To do this, select 'Analyze → Clustering → Show'.
- Visualize ligand-protein interaction in Discovery Studio Visualizer. On Discovery Studio Visualizer, click 'File → Open' and select conf1_ligand_1.pdbqt in your working directory. On the left side of Discovery Studio Visualizer window, click 'View Interactions' and then 'Ligand interactions'.

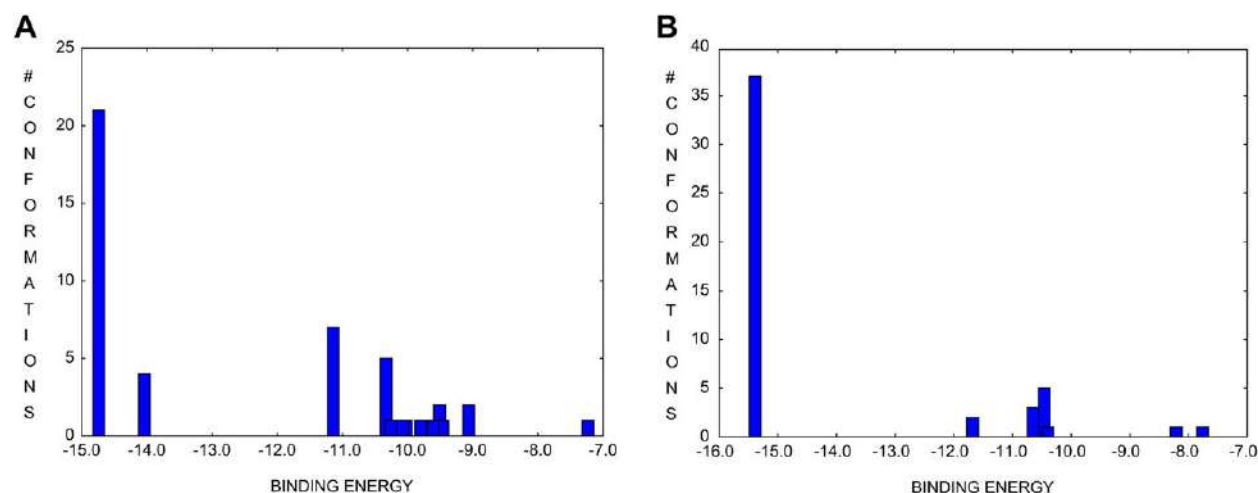


Fig. 5 Binding-energy-based clustering of docking poses of ligands (A) **1** and (B) **2** in the ATP site of PKC ϵ .

For ligand **2**, steps in sections “Ligand Preparation” to “Molecular Docking” are applied to predict the binding affinity of this balanol analog, except Section “Ligand Preparation” step 1.

Discussion

Ligand **1** has an experimental binding affinity of 0.40 nM to PKC ϵ . Molecular docking using AutoDock suggests 15 pose clusters based on predicted binding energy for this ligand when it is bound to the ATP site of PKC ϵ . Binding energy values of the clusters span between -7.0 and -15 kcal/mol. The main cluster contains 21 poses with predicted binding energy values near -15 kcal/mol (**Fig. 5(A)**), which are the most negative among the others. In this cluster, the strongest docking pose has a binding energy of -14.74 kcal/mol. According to AutoDock result, the docking pose shows an intermolecular energy of -18.02 kcal/mol, a van der Waals and hydrogen bond (H-bond) desolvation energy of -15.90 kcal/mol, electrostatic energy of -2.12 kcal/mol, total internal energy of -1.96 kcal/mol, torsional energy of 3.28 kcal/mol, and unbound energy of -1.96 kcal/mol.

Visual inspection of this docking pose in the ATP site of PKC ϵ suggests that the hydroxyl group on the benzamide moiety (C5'OH) interacts via two H-bonds with Glu489 and Val491 in the adenine subsite (**Fig. 6(A)**). The benzamide moiety may interact more tightly by introducing an additional hydroxyl group adjacent to C5'OH. The presence of such additional hydrogen group may strengthen the binding of balanol analog to PKC ϵ . However, introducing such hydroxyl group may be problematic, since the presence of two adjacent hydroxyl groups in the benzamide moiety may lead to the unstable compound. Thus the additional hydroxyl group is incorporated as a hydroxymethyl substituent on the benzamide moiety to form the second ligand. The coordinate file of the second ligand is provided as ligand_2.pdb.

Molecular docking of the second ligand to PKC ϵ predicted that the incorporation of the hydroxymethyl group results in an improved binding affinity to this protein kinase. Binding-energy-based grouping in ADT categorizes the resulting docking poses into seven clusters (**Fig. 5(B)**). The major cluster contains 37 docking poses with predicted binding energy values negative than -15 kcal/mol, which shows stronger interaction than the ligand **1**. Binding energy components of the second ligand are stronger than those of the first ligand such as intermolecular and desolvation energy which is -18.02 and -15.90 kcal/mol, respectively. Furthermore, visual assessment using Discovery Studio Visualizer shows that the presence of an additional hydroxyl group, as a hydroxymethyl substituent, enhances the H-bond interactions between the benzamide moiety and the ATP site of PKC ϵ (**Fig. 6(B)**). This increased interaction resulted in greater binding affinity of the second ligand to PKC ϵ .

Future Directions

In SBDD, molecular docking is a relatively fast and computationally cheap method in the rational drug design process to predict binding modes and affinities of small molecules to the drug targets of interest. However, with the continuous improvement of hardware, other approaches that were previously considered as computationally expensive and demanding are becoming feasible and cheaper. For instance, molecular dynamics simulation (MD), which is supported by the advance development of graphical processing unit (GPU) cards, is now more affordable (*Le Grand et al., 2013*). Thus, MD can more be routinely applied in SBDD. MD can accommodate protein flexibility and solvation effects, which normally cause problems in scoring and ranking of small molecules while using docking approach (*Macalino et al., 2015*). Additionally, the development of the accelerated MD versions alleviates the energy barrier issue in conventional one (*Miao et al., 2015*).

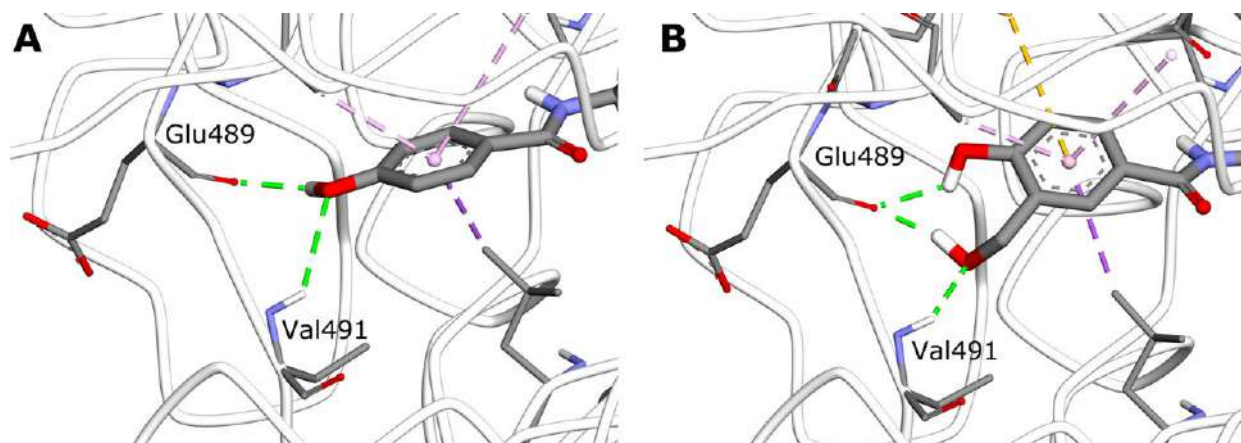


Fig. 6 H-bond interactions made by ligands (A) **1** and (B) **2** with residues of Glu489 and Val491 in the ATP site of PKC ϵ . H-bonds are indicated by green dashed lines.

Another approach in computer-aided drug design is ligand-based drug design (LBDD) which utilizes a set of small molecules with a particular activity and does not need the prior knowledge of the drug target structure. Combining SBDD and LBDD may generate a better outcome in finding novel drug candidates (Macalino *et al.*, 2015).

Closing Remarks

SBDD is a great tool which complements experimental method in the discovery of promising drug candidates. It is now revolutionizing the discovery and development of therapeutic agents by employing the protein structure in the rational drug design. The method reduces the time required as well as failure rate in drug discovery and development. Here, SBDD workflow which includes drug target selection, binding site identification, molecular docking, and ADMET has been described. The practical illustration of molecular docking as a standard tool in SBDD has also been demonstrated.

See also: Algorithms for Structure Comparison and Analysis: Docking. Computational Pipelines and Workflows in Bioinformatics. Host-Pathogen Interactions. Natural Language Processing Approaches in Bioinformatics

References

- Anderson, A.C., 2003. The process of structure-based drug design. *Chemistry & Biology* 10, 787–797.
- Ashkenazy, H., Abadi, S., Martz, E., *et al.*, 2016. ConSurf 2016: An improved methodology to estimate and visualize evolutionary conservation in macromolecules. *Nucleic Acids Research* 44, W344–W350.
- Babu, Y.S., Chand, P., Bantia, S., *et al.*, 2000. Discovery of a novel, highly potent, orally active, and selective influenza neuraminidase inhibitor through structure-based drug design. *Journal of Medicinal Chemistry* 43, 482–486.
- Brylinski, M., Feinstein, W., 2013. eFindSite: Improved prediction of ligand binding sites in protein models using meta-threading, learning and auxiliary ligands. *Journal of Computer-Aided Molecular Design* 27, 551–567.
- Capra, J.A., Laskowski, R.A., Thornton, J.M., Singh, M., Funkhouser, T.A., 2009. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLOS Computational Biology* 5, e1000585.
- Forli, S., Huey, R., Pique, M.E., *et al.*, 2016. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nature Protocols* 11, 905–919.
- Gao, M., Skolnick, J., 2013. APoc: Large-scale identification of similar protein pocket. *Bioinformatics* 29, 597–604.
- Garg, R., Benedetti, L.G., Abera, M.B., *et al.*, 2014. Protein kinase C and cancer: What we know and what we do not. *Oncogene* 33, 5225–5237.
- Guilloux, V., Le, Schmidtke, P., Tuffery, P., 2009. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics* 10, 168.
- Hardianto, A., Yusuf, M., Liu, F., Ranganathan, S., 2017. Exploration of charge states of balanol analogues acting as ATP-competitive inhibitors in kinases. *BMC Bioinformatics* 18, 18 (Suppl 16), 572. <http://dx.doi.org/10.1186/s12859-017-1955-7>.
- Imming, P., Sinning, C., Meyer, A., 2006. Drugs, their targets and the nature and number of drug targets. *Nature Reviews Drug Discovery* 5, 821–834.
- Itzstein, M., Thomson, R., 2009. Anti-influenza drugs: The development of sialidase inhibitors. In: Kräusslich, H.-G., Bartenschlager, R. (Eds.), *Antiviral Strategies*. Berlin: Springer, pp. 111–154.
- Konc, J., Janežič, D., 2010. ProBis: A web server for detection of structurally similar protein binding sites. *Nucleic Acids Research* 38, W436–W440.
- Kulanthaivel, P., Hallock, Y.F., Boros, C., *et al.*, 1993. Balanol: A novel and potent inhibitor of protein kinase C from the fungus *Verticillium balanoides*. *Journal of the American Chemical Society* 115, 6452–6453.
- Laskowski, R.A., MacArthur, M.W., Thornton, J.M., 2001. PROCHECK: Validation of protein structure coordinates. In: Rossmann, M.G., Arnold, E.D. (Eds.), *International Tables of Crystallography*, vol. F. Crystallography of Biological Macromolecules. Netherlands: Kluwer Academic Publishers, pp. 722–725.
- Le Grand, S., Götz, A.W., Walker, R.C., 2013. SPFP: Speed without compromise – A mixed precision model for GPU accelerated molecular dynamics simulations. *Journal of Chemical Theory and Computation* 13, 374–380.
- Li, J.J., 2013. History of drug discovery. In: *ELs*. Chichester: John Wiley & Sons, Ltd., pp. 1–42.

- Macalino, S.J.Y., Gosu, V., Hong, S., Choi, S., 2015. Role of computer-aided drug design in modern drug discovery. *Archives of Pharmacol Research* 38, 1686–1701.
- Miao, Y., Feher, V.A., McCammon, J.A., 2015. Gaussian accelerated molecular dynamics: Unconstrained enhanced sampling and free energy calculation. *Journal of Chemical Theory and Computation* 11, 3584–3595.
- Muegge, I., Bergner, A., Kriegl, J.M., 2017. Computer-aided drug design at Boehringer Ingelheim. *Journal of Computer-Aided Molecular Design* 31, 275–285.
- Patel, A.R., Hardianto, A., Ranganathan, S., Liu, F., 2017. Divergent response of homologous ATP sites to stereospecific ligand fluorination for selectivity enhancement. *Organic & Biomolecular Chemistry* 15, 1570–1574.
- Petsko, G.A., Ringe, D., 2010. X-ray crystallography in the service of structure-based drug design. In: Merz, K.M., Ringe, D., Reynolds, C.H. (Eds.), *Drug Design: Structure- and Ligand-Based Approaches*. New York: Cambridge University Press, pp. 17–29.
- Rodriguez, R., Chinae, G., Lopez, N., Pons, T., Vriend, G., 1998. Homology modeling, model and software evaluation: Three related resources. *Bioinformatics* 14, 523–528.
- Schenone, M., Dancik, V., Wagner, B.K., Clemons, P., 2013. Target identification and mechanism of action in chemical biology and drug discovery. *Nature Chemical Biology* 9, 232–240.
- Taylor, S.S., Yang, J., Wu, J., *et al.*, 2004. PKA: A portrait of protein kinase dynamics. *Biochimica et Biophysica Acta – Proteins and Proteomics* 1697, 259–269.
- Wass, M.N., Kelley, L.A., Sternberg, M.J., 2010. 3DLigandSite: Predicting ligand-binding sites. *Nucleic Acids Research* 38, W469–W473.
- Weisel, M., Proschak, E., Schneider, G., 2007. PocketPicker: Analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal* 1.

Further Reading

- Forli, S., Huey, R., Pique, M.E., *et al.*, 2016. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nature Protocols* 11, 905–919.
- Li, J.J., 2013. History of drug discovery. In: *eLs*. Chichester: John Wiley & Sons, Ltd., pp. 1–42.
- Macalino, S.J.Y., Gosu, V., Hong, S., Choi, S., 2015. Role of computer-aided drug design in modern drug discovery. *Archives of Pharmacol Research* 38, 1686–1701.
- Merz, K.M., Ringe, D., Reynolds, C.H., 2010. *Drug Design: Structure- and Ligand-Based Approaches*. New York: Cambridge University Press.
- Muegge, I., Bergner, A., Kriegl, J.M., 2017. Computer-aided drug design at Boehringer Ingelheim. *Journal of Computer-Aided Molecular Design* 31, 275–285.

Relevant Websites

- <http://autodock.scripps.edu/>
AutoDock.
- <http://accelrys.com/products/collaborative-science/biovia-discovery-studio/visualization-download.php>
Discovery Studio Visualization.
- <http://mgltools.scripps.edu>
MGLTools Website.
- <http://www.rcsb.org/>
RCSB PDB.
- <http://www.ccdc.cam.ac.uk/>
The Cambridge Crystallographic Data Centre (CCDC).

Biographical Sketch



Ari Hardianto is a Lecturer in the Department of Chemistry, Universitas Padjadjaran, Indonesia. Ari Hardianto received his PhD from the Department of Chemistry and Biomolecular Sciences at Macquarie University. He has worked on the development of protein kinase C ϵ inhibitors using structural bioinformatics methods such as homology modeling, molecular docking, and molecular dynamics simulation. He also has research interest in immunoinformatics, chemometrics, and applications of data science in chemistry and molecular biology.



Muhammad Yusuf is a Lecturer in the Department of Chemistry, Universitas Padjadjaran, Indonesia. He is currently the head of Bioinformatics Division at the Research Center of Molecular Biotechnology and Bioinformatics, Universitas Padjadjaran. Muhammad Yusuf received his PhD from the School of Pharmaceutical Sciences, Universiti Sains Malaysia. He has worked on homology modeling, molecular docking, and molecular dynamics simulation to study the behavior of the oseltamivir-resistance neuraminidase, a drug target of influenza virus. He also has a research interest in the molecular modeling study of protein engineering, diagnostics, therapeutics, natural compounds, and rare-earth metal separation.



Fei Liu is a Senior Lecturer in the Department of Molecular Sciences, Macquarie University. She did her PhD studies in Organic Chemistry at Yale University, USA. After her PhD, she moved to Boston, Massachusetts as an NIH postdoctoral fellow to work on new biosynthetic approaches in molecular medicine at the Harvard Medical School. In 2004, Fei moved from Boston to Sydney and is currently a teaching/research academic at Macquarie. The current phase of her research focuses on developing efficient synthetic methods for accessing small molecules with useful properties in chemistry and biology. Her long-term interest in chemical proteomics along with its applications in basic biological discovery and medicine and synthesis of novel candidate cancer drugs.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books in as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.

Network-Based Analysis for Biological Discovery

Lokesh P Tripathi, Yi-An Chen, and Kenji Mizuguchi, National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan

Yoichi Murakami, Tokyo University of Information Sciences, Wakaba-ku, Chiba, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Biological processes are complex systems, involving many interactions among elementary units of a living system; these include biological macromolecules such as DNA, RNA, proteins, lipids and also small molecule metabolites. To describe such processes, a common representation is a network model in which, the participating biomolecules are represented as *nodes* and the relationships or links between them as *edges*. Proteins are the most important biological building blocks and they carry out their functions in the cells by interacting with each other. Therefore, it is not surprising that the largest amount of biomolecular interaction data is available for PPIs and consequently, the substantial chunk of biological network analysis encompasses the construction and analysis of PPI networks (PPINs). PPIs are crucial to the formation of macromolecular structures and enzymatic complexes that form the basis of nearly every cellular process ranging from signal transduction and cellular transport to catalysing metabolic reactions, activating or inhibiting other proteins and biomolecular synthesis. PPIs are thus essential to homeostasis and their dysregulation typically leads to cellular dysfunction and is often associated with various diseases. A systematic mapping of protein interactomes, i.e., the entirety of PPIs in a cell or an organism, is necessary to gain a deeper understanding of the roles of PPIs and PPINs in fundamental cellular processes. It also enables a better understanding of the genotype-phenotype relationships and the perturbations that are involved with the onset of complex diseases.

Owing to their high specificity, PPIs are also promising targets to develop drugs that attuned to specific disease-related pathways (Jubb *et al.*, 2015; Wells and McClendon, 2007; Murakami *et al.*, 2017). However, in this review, we will focus on reconstruction and analysis of PPINs and their applications in interpreting available biological data to gain a deeper understanding of cellular processes and disease mechanisms. We first discuss how experimentally defined PPIN data have been generated and utilised for knowledge discovery. Next, we will discuss why *in silico* methods for PPI characterisation are important in a PPIN-based biological research. We will conclude with how future mapping efforts focussing on a more dynamic analysis of PPINs will shape the field.

Experimental Methods to Generate Proteome-Scale Interaction Maps

A variety of powerful experimental techniques are now available to characterise PPIs. Initially, however, interactions between protein pairs were described by independent studies that employed small-scale biochemical or genetic experiments (Koh *et al.*, 2012). The steady improvements in experimental methodologies and the development of new technologies that were more amenable to PPI mapping on a larger scale, such as yeast two-hybrid system (Y2H) (Fields and Song, 1989) (see below), have allowed rapidly increasing amounts of PPIs to be characterised. However, the advent of high-throughput genome sequencing technologies and the genomics breakthrough at the turn of the millennium was a milestone that paved the way for the PPI characterisation on a proteome-wide scale. The appearance of the first draft model organism genome sequences and the accompanying collection of genome-wide open reading frames (ORFs), coupled with the availability of robust, high-throughput PPI detection methods allowed the PPI mapping to truly take off (Luck *et al.*, 2017). Thus, proteome-scale interaction maps have been generated for different proteomes using available experimental techniques that are amenable to large-scale interactome mapping (Vidal *et al.*, 2011; Huttlin *et al.* 2015; Rolland *et al.* 2014).

The experimental methods for PPI mapping can be broadly classified into two groups. The first group consists of methods that are capable of simultaneously investigating a large number of PPIs, often on an interactome scale, whereas the second group consists of methods that are more suited to deeply examine individual PPIs (Tunchag *et al.*, 2009). The Y2H system is one of the most widely used methods to map binary PPIs. Y2H is an *in vivo* method based on the reconstitution of a functional transcription factor (TF) following an interaction between two proteins and the subsequent activation of reporter genes controlled by the TF (Fields and Song, 1989). It is a scalable and relatively inexpensive method that is well suited to detecting binary interactions between proteins and therefore facilitates the characterisation of physiologically relevant PPIs. Unsurprisingly, Y2H has been the method of choice for generating proteome-wide binary interaction maps for a number of model organisms such as *E. coli* (Rajagopala *et al.*, 2014), Yeast (Uetz *et al.*, 2000; Ito *et al.* 2001; Yu *et al.*, 2008; Vo *et al.*, 2016), *C. elegans* (Li *et al.*, 2004), *Drosophila* (Formstecher *et al.*, 2005), *Arabidopsis thaliana* (Arabidopsis Interactome Mapping Consortium, 2011) and human (Vidal *et al.*, 2011; Huttlin *et al.*, 2015; Rolland *et al.*, 2014). However, Y2H suffers from a few notable shortcomings; it is less amenable to capturing PPIs involving extracellular or membrane proteins, PPIs that require proper folding as a part of protein complex subunits, or PPIs that require post-translational modifications (PTMs). To overcome the limitations of Y2H and to study different types of PPIs, several Y2H variants such as the mammalian cell-based two-hybrid assay (Luo *et al.*, 1997), the membrane-anchored two-hybrid assay (Snider *et al.*, 2010), and the three-hybrid assay (Maruta *et al.*, 2016) have been developed.

PPIs can also be mapped on an interactomic scale by Affinity Purification – Mass Spectrometry (AP-MS) that involves biochemical purification of the epitope-tagged target proteins from the cells, followed by the identification of the components of the purified protein complexes (including proteins interacting with the target protein) using mass-spectrometry analysis (Dunham *et al.*, 2012). AP-MS method has been widely used to characterise protein complexes on a large scale in different species including yeast, *Drosophila* and human (Guruharsha *et al.*, 2011; Krogan *et al.*, 2006; Ewing *et al.*, 2007). To overcome non-specific detection of co-purified proteins, two-step tandem affinity protein purification systems have been developed (Burckstummer *et al.*, 2006). This approach allows the preparation of a substantially pure target protein complex and reduces the background signals. The quantitative mass-spectrometry analysis also has been used to identify different contaminants (Trinkle-Mulcahy *et al.*, 2008). However, AP-MS data may not always detect binary interactions and often reflects only steady state PPI dynamics, thereby, potentially missing weak and transient interactions.

Another approach is to employ co-fractionation and mass spectrometry to characterise protein complexes. In this method, cellular extracts are subject to intense co-fractionation by using biochemical separation methods such as chromatography and a precise co-elution of proteins is used to determine PPIs. A distinct advantage of this method over AP-MS is that it allows a mapping of dynamic PPIs and the determination of the size of the protein complexes owing to the use of size-exclusion chromatography (Yang *et al.*, 2015). Thus, this method has been used to map protein complexes on a proteome-wide scale in different organisms such as yeast and human (Doerr, 2012; Havugimana *et al.*, 2012; Phanse *et al.*, 2016).

Eventually, a combination of different methods will provide a more comprehensive protein interaction map, since each method will likely lead to the exploration of different aspects of the interactome.

PPIN-Based Network Analysis

PPINs assembled from the experimentally characterised PPI data are an important avenue to understand the organisation of cellular events and the biology of the living systems and complex diseases. PPI data extracted from the proteomic literature and compiled within the expert-curated resources are highly useful in uncovering functional PPINs and for guiding subsequent research. However, such data are scattered across multiple databases that differ in scope and content, i.e., the type and number of PPIs they contain, the number of organisms that are covered and the experimental and computational methods that were used for PPI characterisation. Therefore, combining different PPI maps is necessary to obtain a complete view of protein interactomes (Razick *et al.*, 2008; Chen *et al.*, 2011; Chen *et al.*, 2016).

However, the combined PPI datasets will likely be noisy and include many false positives that are inherent in experimentally characterised PPIs and therefore, the assembled PPI data must be carefully assessed before using them for PPIN analyses. A relatively simple and commonly used approach is to consider only those PPIs that are determined by at least two different experimental methods or reported in the literature in two different publications (Chen *et al.*, 2016). Biophysical data from the experimentally determined structures of interacting proteins can be useful since they offer detailed insights into how PPIs are formed at the atomic scale (De Las Rivas and Fontanillo, 2010; Erijman *et al.*, 2014; Moal *et al.*, 2011, 2013), but such data are available for very few protein complexes. It is also important to view and analyse the PPIN data in the proper spatiotemporal context such as cellular/tissue specificity, protein subcellular localisation, gene expression patterns, homologous associations and PTMs (Schaefer *et al.*, 2013). A number of studies have employed PPIN analysis to probe a broad spectrum of biomolecular processes and seek answers to key biological questions (Fig. 1).

Network Topology

A typical PPIN is an undirected graph with each protein represented as a *node* and each interaction between two proteins represented as an *edge* (Fig. 1). This wiring or the connectivity of the different proteins within a PPIN is referred to as network topology. There is a strong correlation between the topological properties of a network and its functioning. Therefore, graph theory concepts such as *node degree distribution*, *betweenness centrality* and *shortest path length* have been used to identify key determinants of network function (Raman, 2010) and also to better understand network perturbations. Network ‘hubs’ are highly connected proteins with many PPIs (that is, they have a high *node degree*); they are therefore likely to have a greater influence on network functioning via multiple interactions. Network ‘bottlenecks’ are proteins with high *betweenness centrality*; they regulate the flow of signalling information across the network and therefore, represent key nodes for communication (Yu *et al.*, 2007) (Fig. 1(a)). Thus, analysing network topologies can be a source of new discoveries such as identifying novel biomarkers and potential drug targets (Csermely *et al.*, 2013; Kotlyar *et al.*, 2012; Charitou *et al.*, 2016; Gebicke-Haerter, 2016; Hakes *et al.*, 2008). Network topologies have been employed to identify novel disease-associated genes (Vidal *et al.*, 2011; Feldman *et al.*, 2008; Sarajlic *et al.*, 2013) and also to better understand the organisation of localised cellular networks. For example, Gupta *et al.* (2015) mapped the centriole-cilium protein interaction landscape by generating a PPIN consisting of > 7000 interactions and using network topology analysis, which led to the discovery of novel insights into human centrosome and cilia biology.

Understanding Disease Mechanisms Based on Network Neighbourhood

An important outcome of using PPIN topology to investigate disease mechanisms is the disease module hypothesis, which is based on the observation that genes associated with the same disease preferentially interact with each other and tend to form

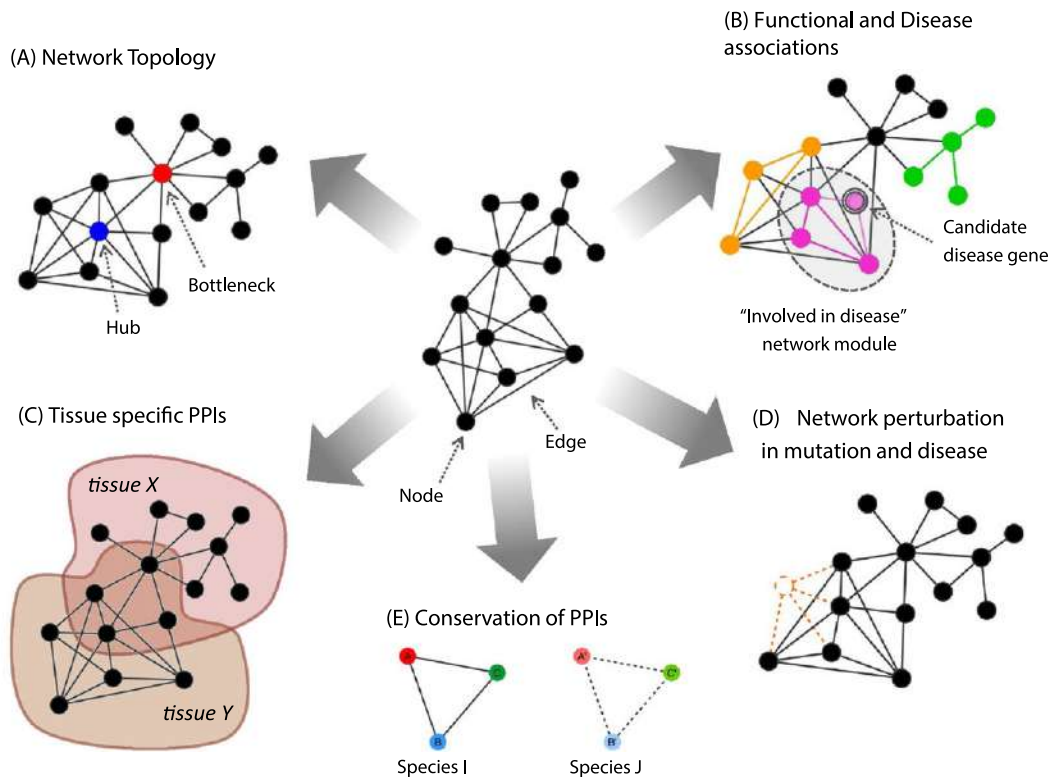


Figure 1 Applications of PPIN-based analysis using large scale protein interactome maps. A reference interactome network is usually illustrated as proteins represented as *nodes* (solid circles) interacting with other proteins via PPIs represented as *edges* (solid lines). Applications of PPINs include (a) Investigating protein function and essentiality using network topology, (b) Understanding disease mechanisms by network neighbourhood, (c) Investigating the context specificity of PPIs such as in the context of cell/tissue-specific interactions (d) PPIN rewiring caused by mutation and/or disease and (e) Leveraging orthologous gene associations to map PPIs across species.

well-connected clusters in the same network neighbourhood (Barabasi *et al.*, 2011; Ideker and Sharan, 2008; Menche *et al.*, 2015) (Fig. 1(b)). The disease module hypothesis has attracted much interest from the researchers, since a given set of disease-causing genes can provide a deeper insight into related diseases. This is carried out by collating other disease-causing genes, by defining tightly interconnecting communities of functionally related or disease-related proteins and then retrieving the uncharacterised neighbouring genes connected with the initial “seed” genes by *shortest paths*. For instance, Huttlin and colleagues constructed BioPlex, a network of experimentally derived human PPIs and defined many protein communities and subnetworks that enabled functional characterisation of poorly characterised human proteins, including many with novel roles in human diseases such as cancer and hypertensive disease (Huttlin *et al.*, 2015, 2017). Rolland and co-workers also demonstrated that known cancer-associated genes are highly interconnected in the human protein interactome (Rolland *et al.*, 2014).

Context-Specificity of Protein Interactomes

Complex diseases are usually very site-specific and mostly impact specific cells and/or tissues (Goh *et al.*, 2007; Magger *et al.*, 2012). Therefore, it is necessary to examine the PPI data, in cellular and tissue context, such as cell/tissue-specific expression of proteins, the relative abundance of alternatively spliced isoforms and PTMs and their impact on the interactome of different cells and tissues. Magger and co-workers, for instance, observed that using tissue-specific PPI networks, greatly enhanced the prioritisation of candidate disease-causing genes compared with generic PPINs and also highlight novel tissue-disease associations (Magger *et al.*, 2012).

PPIN Rewiring During Disease, Mutation and Evolution

The rapid proliferation of high-quality sequence data generation using low-cost Next Generation Sequencing (NGS) experimental platforms coupled with speedy bioinformatics methods have contributed to the mapping of a large number of sequence and structural variants associated with clinically relevant phenotypes and diseases. Although there is a limited understanding of causal relationships between various mutations and diseases, it has been well established that functional variants may often impact overall protein functions including PPIs (Shameer *et al.*, 2016). Comparative PPIN analysis, therefore, offers a promising avenue to examine the genotype-phenotype relationships underlying key biological processes and the causative mechanisms of disease-

causing mutations emanating from the gain and/or loss of specific PPIs (Fig. 1(d)). Different studies involving the analysis of global PPINs in human have highlighted mutation-induced network perturbation and loss of specific PPIs that can be reliably linked with specific diseases (Rolland *et al.*, 2014; Sahni *et al.*, 2015).

PPIN rewiring has also been examined in the context of evolution and conservation of PPIs and interactions across species (Fig. 1(e)). Vo and co-workers (Vo *et al.*, 2016) constructed a high-quality binary protein interactome for *S. pombe* and implemented a framework to compare the organisation and evolution of PPINs across yeast and human. Their findings revealed extensive species-specific network rewiring and novel paradigms on network co-evolution and conservation of interacting proteins.

Despite the wealth of knowledge emerging from the analysis of increasingly available large-scale PPI data, the known protein interactomes are incomplete. This is not only due to the challenges associated with the experimental determination of PPIs, it was speculated that to obtain a comprehensive coverage of the human protein interactome, ~200 million protein pairs would need to be experimentally tested (Rolland *et al.*, 2014). Moreover, because of systemic bias, well studied genes and proteins are screened more frequently than the others and are thereby, disproportionately represented in the literature and PPI databases. This has led to other proteins, potentially the causative agents of diseases, remaining under-represented (Edwards *et al.*, 2011). This issue is particularly visible in model organisms such as rat and mouse (Murakami *et al.*, 2017) that are key for biomedical research and it is estimated that only ~10% of the human protein interactome have been characterised so far (Kotlyar *et al.*, 2015). Therefore, to generate a complete interactome, it is necessary to develop computational methods for PPI prediction to expand the coverage of PPI space and mine the protein interactomes for knowledge discovery.

***In Silico* Prediction of PPIs for PPI Network Analysis and Assessment of PPI Quality**

A wide range of *in silico* methods has been proposed to predict new PPIs and to assess the quality of existing PPIs using various features obtained from known PPIs, such as gene co-localisation, phylogenetic profiling, gene fusion, domain-domain interaction (DDI), homologous interactions, and also using various computational methods such as machine learning and text mining (Murakami *et al.*, 2017; Peng *et al.*, 2017; Keskin *et al.*, 2016). A list of the features and techniques used for the prediction and assessment of PPIs are summarised in Table 1.

In silico methods can be broadly classified into two types: Low-resolution methods that offer a simple binary classification to determine whether a given pair of proteins interact or not, and the high-resolution methods that can predict the detailed interatomic interactions between proteins (Vakser, 2014). The former can generally predict a large number of PPIs at lower cost and in much less time than experimental methods, and are also applicable to the assessment of a large number of existing PPIs. The latter can predict PPIs based on their structural and physicochemical complementarities, i.e., protein docking (Tuncbag *et al.*, 2009; Keskin *et al.*, 2016), and require protein structural information, and therefore, are expensive approaches for characterising the entire interactome.

Although recent advances in docking methodologies have produced reliable protein complex models, docking proteins with large conformational changes and/or without prior knowledge of the PPI sites (*ab initio* docking) remains a non-trivial task (Janin *et al.*, 2003; Janin and Wodak, 2007). To achieve this task, molecular dynamics (MD) simulations, which take into consideration the physical movements of atoms in proteins, have been used to elucidate the precise positions of atoms involved in the interaction; MD simulations, however, are computational resource intensive and therefore, unsuitable for whole interactome modelling.

Consequently, most of the existing *in silico* methods applicable to interactome modelling and PPI assessment rely heavily on previously known PPIs and on primary protein sequence information, which is more widely available than structural information. Below, we discuss the underlying principles of different low-resolution widely applicable *in silico* PPI prediction methods for PPIN analysis.

Table 1 A list of the features and techniques used for the prediction and assessment of PPIs

Utilised feature	Fundamental concepts
Protein structure	Structural and physicochemical complementarities.
Interologue	Orthologous PPIs among different species: If two proteins are known to interact in a species X and are conserved in a species Y, these proteins may likely interact in the species Y.
Domain information	Domain-domain interactions, since domains are directly involved in the intermolecular interactions.
Gene neighbourhood	Two proteins are likely to interact when their genes are located in the same region of genome.
Phylogenetic similarity	Functionally related proteins, i.e., interacting proteins, tend to be inherited together during evolution and co-evolve through the interaction. Thus, they have similar topological phylogenetic profiles.
Gene fusion (Domain fusion)	Separate genes can be fused to form a single chimeric gene, which is called as “Rosetta Stone”.
Function annotation	Interacting proteins tend to have shared functional annotations since they are often involved in the same biological processes.
Text mining	Grammatical rules to retrieve the co-occurrence of predefined genes or proteins, and the relationship between these entities in repositories such as literature and various databases.
Machine learning	Training a classifier on a set of known PPI data to predict whether or not a given pair of proteins interact, using informative features extracted from the training PPI dataset.

Interologue-Based Methods

Orthologous proteins are descended from a common ancestral gene as a consequence of speciation, and they are believed to retain similarity in structure and function (Fitch, 2000; Koonin, 2005; Watson *et al.*, 2005; Webber and Ponting, 2004), including PPIs. *Interologue-based methods* predict PPIs and assess the quality of existing PPIs based on the biological principle of orthologous PPIs (*interologue*) across different species, that is, if two or more proteins are known to interact in a species *I* and if they have identifiable orthologues in species *J*, the orthologous proteins may also potentially interact in species *J* (Fig. 1(e)). This approach is similar to those used for gene function annotation, where a gene function is inferred from the function of homologous genes in other species. A large amount of PPI data are currently available in public databases (Salwinski *et al.*, 2004; Licata *et al.*, 2012; Aranda *et al.*, 2010; Szklarczyk *et al.*, 2015; Chatr-Aryamontri *et al.*, 2017), where orthologous PPIs can be identified. This approach is useful in transferring the annotation of PPIs from one species to another species of interest; for example, it has been applied to predict PPIs in human cancer proteins (Jonsson and Bates, 2006). The accuracy of this approach depends on the reliability of the interactions, so it is considered to be inappropriate for the prediction of transient interactions because such interactions are poorly conserved across species (Keskin *et al.*, 2016). Thus, other features, such as domain co-occurrences, gene co-expression or functional similarity, can be integrated into this approach to assess PPIs and the co-localisation of the proteins predicted to interact, as implemented in PPI-Search (Chen *et al.*, 2009), BIPS (Garcia-Garcia *et al.*, 2012), I2D (Brown and Jurisica, 2005) and PSOPIA (Murakami and Mizuguchi, 2014) (Table 2).

Domain-Based Methods

Protein domains are independent evolutionary units, which define protein function. Multiple studies have demonstrated that domain-domain interactions (DDIs) are useful for predicting PPIs, since domains are directly involved in the intermolecular interactions (Memisevic *et al.*, 2013; Shoemaker and Panchenko, 2007). The *domain-based methods* can identify PPIs without relying on homologous interactions that exist in public databases, unlike the *interologue-based approaches*. To use DDIs for the prediction of new PPIs, most methods annotate protein sequences using domain databases such as Pfam (Finn *et al.*, 2016), SCOP2 (Andreeva *et al.*, 2014) and CATH (Sillitoe *et al.*, 2015). There are two types of *domain-based approaches* (Shoemaker and Panchenko, 2007). First type consists of the *association approach-based methods* that are based on the idea that certain domains are frequently observed in interacting proteins and therefore can be used as markers to predict new PPIs (Sprinzak and Margalit, 2001). However, this approach does not consider the relationships of all possible domain pairs in interacting pairs, and also the missing domain pairs not observed in known interacting pairs. The second type is the *Bayesian network approach*, where the interaction probabilities of all possible domain pairs are estimated using the Maximum Likelihood Estimation (Burger and van Nimwegen, 2008; Deng *et al.*, 2002). The accuracy of this approach depends on the reliability of the domain assignments, so a sufficient coverage of domain databases is necessary to obtain sufficient true positives and negatives. This approach can also be used to assess the quality of PPIs since an interaction can be deemed more reliable if they contain domain pairs found in known PPIs in the database (Ng *et al.*, 2003).

In recognition of the limitations of the *domain-based approaches*, a new set of PPI prediction methods have been developed, which are based on the principle of short co-occurring polypeptide regions as mediators of PPIs (Pitre *et al.*, 2012; Schoenrock *et al.*, 2014). A distinct advantage of these methods is that unlike the classical *domain-based approaches*, they are designed to predict PPIs solely on the basis of primary sequence and are thus, not handicapped by the absence of characterised protein domains; these methods are therefore much useful for large-scale PPI prediction. For instance, Schoenrock and colleagues (Schoenrock *et al.*,

Table 2 A selection of *in silico* prediction web servers that are useful for PPI network analysis

Available in silico prediction web servers for PPI network analysis

Web server	Method, URL (http://)
BIPS: Biana Interolog Prediction Server (Garcia-Garcia <i>et al.</i> , 2012)	Interologue (across multiple species)/domain/functional annotation http://sbi.imim.es/web/index.php/research/servers/bips
I2D: Interolog Interaction Database (Brown and Jurisica, 2007)	Interologues (across seven species; human, rat, mouse, fly, worm, yeast and hhv8) http://ophid.utoronto.ca/ophidv2.204
InterologFinder (Wiles <i>et al.</i> , 2010)	Interologues (across five species; human, mouse, fly, worm and yeast) http://interologfinder.org/simpleIIFS
PPI-Search: Protein-Protein Interaction Search (Chen <i>et al.</i> , 2009)	Interologues (across multiple species)/domain/functional annotation http://140.113.15.76/ppisearch*
PSOPIA: Prediction Server of Protein-Protein Interactions (Murakami and Mizuguchi, 2014)	Homologues (within the human genome)/domain/the shortest path between two homologous proteins http://mizuguchilab.org/PSOPIA*
MirrorTree Server (Ochoa and Pazos, 2010)	Phylogenetic similarity http://csbg.cnb.csic.es/mtserver/

*These servers accept only a single protein pair per submission.

2014) designed a tool to predict human PPINs and validated their prediction results experimentally. They further employed their computationally predicted human PPINs for the prediction of gene functions and formations of PPI complexes in human diseases, some of which were validated by follow-up experimental assays (Schoenrock *et al.*, 2014).

Gene Neighbourhood-Based Methods

Gene co-localisation-based methods are based on the idea that two proteins are likely to interact when their genes are located in the same region of the genome (Tamames *et al.*, 1997). This approach requires a number of genome sequences to predict and assess PPIs using information about the conservation of gene locations, and confidence increases with increasing genome sequences. Although this approach can predict new PPIs without relying on known PPIs reported in the literature or available in databases, it would not be applicable to the eukaryotic genomes, since there is no tangible evidence that two genes which encode for interacting proteins are always co-localised within a genome. Although this approach is simple in comparison with other *in silico* approaches, it often fails to detect interactions between distantly located genes and often generates a high amount of false negatives in the eukaryotic genomes (Zahiri *et al.*, 2013).

Phylogenetic Similarity-Based Methods

There are two types of *phylogenetic similarity-based approaches*. One is the *phylogenetic tree-based approach* (also known as the *mirror tree* approach), which is based on the idea that interacting proteins tend to co-evolve through the interaction and thus have similar topological phylogenetic tree profiles (Sato *et al.*, 2005; Pazos and Valencia, 2001; Goh and Cohen, 2002), as implemented in MirrorTree (Ochoa and Pazos, 2010) (Table 2). However, when a pair of proteins co-evolve through the speciation events even if they do not interact, a large number of false positives are created due to the generation of similar mirror trees (Ochoa and Pazos, 2014). The elimination of non-specific tree similarities have been attempted by different methods, for example, the 16S rRNA tree is used as a representation of the speciation process to normalise the non-specific similarities by subtracting their phylogenetic distances from the distance matrices for a pair of proteins (Sato *et al.*, 2005; Pazos *et al.*, 2005). The second one is the *phylogenetic profile-based approach* that is based on the assumption that functionally related proteins tend to be inherited together during evolution. A phylogenetic profile represents the conservation of a certain protein in various species. Thus, if two proteins are functionally related, they are more likely to have similar phylogenetic profiles (Juan *et al.*, 2008). However, the outcomes of this approach are largely dependent on the number of the species used to construct phylogenetic profiles. Thus, this approach is not very suitable for eukaryotic proteins since there are comparatively fewer eukaryotes with complete genomic sequences than prokaryotes (Muley and Ranjan, 2012).

Gene Fusion-Based Methods

Gene fusion (domain fusion)-based approaches are based on the genetic observation that independent genes can combine or “fuse” together to form a single chimeric gene, known as “Rosetta Stone”. The method is based on the observation that two separate proteins are functionally related and are likely to interact if certain proteins in a given species consist of co-localised domains otherwise mapped to different proteins in another species (Enright *et al.*, 1999; Marcotte *et al.*, 1999; Chia and Kolatkar, 2004). Although the gene fusion is an informative feature of the functional relationships between different proteins, it requires a mapping of domain architecture across different genomes and is usually only applicable to those proteins corresponding to well-characterised protein domain families.

Function Annotation-Based Methods

Function annotation-based methods are based on the observation that interacting proteins tend to significantly share function annotations since they are involved in the same biological processes (Peng *et al.*, 2017). This method is often used to assess the quality of existing PPIs and also evaluate the reliability of different sources with experimentally determined PPIs, for example, the reliability is evaluated by computing the fraction of interacting proteins that have at least one identical function (Nabieva *et al.*, 2005). Gene Ontology (GO) can be used to define functional similarity between two proteins to assess the quality of existing PPIs (Cho *et al.*, 2007). However, this approach cannot reliably evaluate the quality of PPIs where the interacting proteins are annotated with many different GO terms (Peng *et al.*, 2017).

Text Mining-Based Methods

Text mining-based approaches use grammatical rules to retrieve the co-occurrence of predefined entities, i.e., in biology, genes or proteins, and the relationship between these entities in repositories such as literature and various databases (Papanikolaou *et al.*, 2015). An interaction between two proteins (A, B) can be ascertained if the grammatical rules, such as “A interaction verb B” or “interaction between A and B”, are used in the repositories. Although such approaches cannot retrieve PPIs not described in the repositories, they may be able to infer potentially novel PPIs in a given species based on homologous PPIs in another species. For example, this

approach has been used to automatically extract host-pathogen interactions from the biomedical literature (Thieu *et al.*, 2012), and also used in the STRING database to retrieve predicted interactions from the literature (Szklarczyk *et al.*, 2017).

Machine Learning-Based Methods

Machine learning-based methods train a classifier on a set of known PPI data to predict whether a given pair of proteins is likely to interact or not. In this approach, various types of features (descriptors) are combined and used to train the classifier, such as the positions of amino acids, the degree of the conservation and the physicochemical characteristics of proteins, the localisation of proteins, domains within proteins or the phylogenetic profile, and so on. Many different machine learning techniques, such as support vector machine (SVM), Naïve Bayes (NB), neural networks (NN), K-nearest neighbors (KNN) and random forest (RF) have been used to imbibe the informative features that can distinguish between true and false interactions. For example, the NB integrating protein domain data, gene expression data and functional annotation data has been applied to the human interactome network analysis to identify PPIs and subnetworks relevant to human cancer (Rhodes *et al.*, 2005). The KNN has been applied to identify human hereditary disease-genes based on topological features, which describe a protein in PPINs (Xu and Li, 2006). Furthermore, an ensemble learning approach, which combines multiple scores obtained from different classifiers trained on different machine learning techniques can be effective (Peng *et al.*, 2017). For example, an ensemble learning method which utilises four classifiers trained using RF, NB, SVM and multilayer perceptron (MLP, a type of NN with multi layers), has been applied to the prediction of PPIs between human and the hepatitis C virus (HCV) proteins (Emamjomeh *et al.*, 2014). The high quality of the training dataset, i.e., informative and unbiased, is crucial for accurate assessments and predictions, as well as for the evaluation of the machine learning models. The testing dataset, for example, is classified into three types by examining if the interacting partner proteins in the dataset are similar to the proteins in the training dataset or not (Hamp and Rost, 2015; Park and Marcotte, 2012). Such a classification offers an effective mechanism not only to evaluate the models but also to prepare high-quality datasets.

Different *in silico* methods based on various types of protein sequence information can be used to predict new PPIs and assess the quality of existing PPIs. Although some methods are more suitable for prokaryotes than eukaryotes, the confidence scores assigned to existing PPIs or to pairs of potentially interacting proteins can be useful to ascertain the reliability of the inferred interactomes for the subsequent PPIN analysis.

Conclusions and Future Prospects

A proteome-wide mapping of PPIs and leveraging them in PPIN-based analyses can help to gain knowledge of the genotype-phenotype relationships and the functioning of complex biological systems. Publicly available PPI data are expected to grow substantially and more comprehensive interactome maps are likely to become available in the near future. Consequently, the accuracy and efficacy of the various *in silico* PPI prediction and scoring methods will also likely improve with the increasing amounts of genomic and proteomics data that are likely to become available in the near future. As PPI mapping transitions from capturing steady state associations to dynamic interactions, it will become increasingly necessary to take additional parameters such as protein isoforms, spatiotemporal expression, localisation and interaction and perhaps even the “strength” of the PPIs into consideration to maximise the robustness of biological insights obtained from the PPIN-based analyses.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Natural Language Processing Approaches in Bioinformatics. Network-Based Analysis of Host-Pathogen Interactions

References

- Andreeva, A., *et al.*, 2014. SCOP2 prototype: A new approach to protein structure mining. *Nucleic Acids Res.* 42 (Database issue), D310–D314.
- Arabidopsis Interactome Mapping Consortium, C., 2011. Evidence for network evolution in an Arabidopsis interactome map. *Science* 333 (6042), 601–607.
- Aranda, B., *et al.*, 2010. The IntAct molecular interaction database in 2010. *Nucleic Acids Res.* 38 (Database issue), D525–D531.
- Barabasi, A.L., Gulbahce, N., Loscalzo, J., 2011. Network medicine: A network-based approach to human disease. *Nat. Rev. Genet.* 12 (1), 56–68.
- Brown, K.R., Jurisica, I., 2005. Online predicted human interaction database. *Bioinformatics* 21 (9), 2076–2082.
- Brown, K.R., Jurisica, I., 2007. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biol.* 8 (5), R95.
- Burckstummer, T., *et al.*, 2006. An efficient tandem affinity purification procedure for interaction proteomics in mammalian cells. *Nat. Methods* 3 (12), 1013–1019.
- Burger, L., van Nimwegen, E., 2008. Accurate prediction of protein-protein interactions from sequence alignments using a Bayesian method. *Mol. Syst. Biol.* 4, 165.
- Charitou, T., Bryan, K., Lynn, D.J., 2016. Using biological networks to integrate, visualize and analyze genomics data. *Genet. Sel. Evol.* 48, 27.
- Chat-Aryamontri, A., *et al.*, 2017. The BioGRID interaction database: 2017 update. *Nucleic Acids Res.* 45 (D1), D369–D379.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2011. TargetMine, an integrated data warehouse for candidate gene prioritisation and target discovery. *PLOS ONE* 6 (3), e17844.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2016. An integrative data analysis platform for gene set analysis and knowledge discovery in a data warehouse framework. *Database (Oxf.)* 2016.

- Chen, C.C., *et al.*, 2009. PPISearch: A web server for searching homologous protein-protein interactions across multiple species. *Nucleic Acids Res.* 37 (Web Server issue), W369–W375.
- Chia, J.M., Kolatkar, P.R., 2004. Implications for domain fusion protein-protein interactions based on structural information. *BMC Bioinform.* 5, 161.
- Cho, Y.R., *et al.*, 2007. Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC Bioinform.* 8, 265.
- Csermely, P., *et al.*, 2013. Structure and dynamics of molecular networks: A novel paradigm of drug discovery: A comprehensive review. *Pharmacol. Ther.* 138 (3), 333–408.
- De Las Rivas, J., Fontanillo, C., 2010. Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLOS Comput. Biol.* 6 (6), e1000807.
- Deng, M., *et al.*, 2002. Inferring domain-domain interactions from protein-protein interactions. *Genome Res.* 12 (10), 1540–1548.
- Doerr, A., 2012. Interactomes by mass spectrometry. *Nat. Methods* 9 (11), 1043.
- Dunham, W.H., Mullin, M., Gingras, A.C., 2012. Affinity-purification coupled to mass spectrometry: Basic principles and strategies. *Proteomics* 12 (10), 1576–1590.
- Edwards, A.M., *et al.*, 2011. Too many roads not taken. *Nature* 470 (7333), 163–165.
- Emamjomeh, A., *et al.*, 2014. Predicting protein-protein interactions between human and hepatitis C virus via an ensemble learning method. *Mol. Biosyst.* 10 (12), 3147–3154.
- Enright, A.J., *et al.*, 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402 (6757), 86–90.
- Erijman, A., Rosenthal, E., Shifman, J.M., 2014. How structure defines affinity in protein-protein interactions. *PLOS ONE* 9 (10), e110085.
- Ewing, R.M., *et al.*, 2007. Large-scale mapping of human protein-protein interactions by mass spectrometry. *Mol. Syst. Biol.* 3, 89.
- Feldman, I., Rzhetsky, A., Vitkup, D., 2008. Network properties of genes harboring inherited disease mutations. *Proc. Natl. Acad. Sci. USA* 105 (11), 4323–4328.
- Fields, S., Song, O., 1989. A novel genetic system to detect protein-protein interactions. *Nature* 340 (6230), 245–246.
- Finn, R.D., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Res.* 44 (D1), D279–D285.
- Fitch, W.M., 2000. Homology a personal view on some of the problems. *Trends Genet.* 16 (5), 227–231.
- Formstecher, E., *et al.*, 2005. Protein interaction mapping: A *Drosophila* case study. *Genome Res.* 15 (3), 376–384.
- Garcia-Garcia, J., *et al.*, 2012. BIPS: BIANA Interolog Prediction Server. A tool for protein-protein interaction inference. *Nucleic Acids Res.* 40 (Web Server issue), W147–W151.
- Gebicke-Haerter, P.J., 2016. Systems psychopharmacology: A network approach to developing novel therapies. *World J. Psychiatry* 6 (1), 66–83.
- Goh, C.S., Cohen, F.E., 2002. Co-evolutionary analysis reveals insights into protein-protein interactions. *J. Mol. Biol.* 324 (1), 177–192.
- Goh, K.I., *et al.*, 2007. The human disease network. *Proc. Natl. Acad. Sci. USA* 104 (21), 8685–8690.
- Gupta, G.D., *et al.*, 2015. A dynamic protein interaction landscape of the human centrosome-cilium interface. *Cell* 163 (6), 1484–1499.
- Gururharsha, K.G., *et al.*, 2011. A protein complex network of *Drosophila melanogaster*. *Cell* 147 (3), 690–703.
- Hakes, L., *et al.*, 2008. Protein-protein interaction networks and biology – What's the connection? *Nat. Biotechnol.* 26 (1), 69–72.
- Hamp, T., Rost, B., 2015. Evolutionary profiles improve protein-protein interaction prediction from sequence. *Bioinformatics* 31 (12), 1945–1950.
- Havugimana, P.C., *et al.*, 2012. A census of human soluble protein complexes. *Cell* 150 (5), 1068–1081.
- Huttlin, E.L., *et al.*, 2015. The BioPlex network: A systematic exploration of the human interactome. *Cell* 162 (2), 425–440.
- Huttlin, E.L., *et al.*, 2017. Architecture of the human interactome defines protein communities and disease networks. *Nature* 545 (7655), 505–509.
- Ideker, T., Sharan, R., 2008. Protein networks in disease. *Genome Res.* 18 (4), 644–652.
- Ito, T., *et al.*, 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA* 98 (8), 4569–4574.
- Janin, J., *et al.*, 2003. CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins* 52 (1), 2–9.
- Janin, J., Wodak, S., 2007. The third CAPRI assessment meeting Toronto, Canada, April 20–21, 2007. *Structure* 15 (7), 755–759.
- Jonsson, P.F., Bates, P.A., 2006. Global topological features of cancer proteins in the human interactome. *Bioinformatics* 22 (18), 2291–2297.
- Juan, D., Pazos, F., Valencia, A., 2008. High-confidence prediction of global interactomes based on genome-wide coevolutionary networks. *Proc. Natl. Acad. Sci. USA* 105 (3), 934–939.
- Jubb, H., Blundell, T.L., Ascher, D.B., 2015. Flexibility and small pockets at protein-protein interfaces: New insights into druggability. *Prog. Biophys. Mol. Biol.* 119 (1), 2–9.
- Keskin, O., Tuncbag, N., Gursay, A., 2016. Predicting protein-protein interactions from the molecular to the proteome level. *Chem. Rev.* 116 (8), 4884–4909.
- Koh, G.C., *et al.*, 2012. Analyzing protein-protein interaction networks. *J. Proteome Res.* 11 (4), 2014–2031.
- Koonin, E.V., 2005. Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.* 39, 309–338.
- Kotlyar, M., *et al.*, 2015. In silico prediction of physical protein interactions and characterization of interactome orphans. *Nat. Methods* 12 (1), 79–84.
- Kotlyar, M., Fortney, K., Jurisica, I., 2012. Network-based characterization of drug-regulated genes, drug targets, and toxicity. *Methods* 57 (4), 499–507.
- Krogan, N.J., *et al.*, 2006. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature* 440 (7084), 637–643.
- Li, S., *et al.*, 2004. A map of the interactome network of the metazoan *C. elegans*. *Science* 303 (5657), 540–543.
- Licata, L., *et al.*, 2012. MINT, the molecular interaction database: 2012 update. *Nucleic Acids Res.* 40 (Database issue), D857–D861.
- Luck, K., *et al.*, 2017. Proteome-scale human interactomics. *Trends Biochem. Sci.* 42 (5), 342–354.
- Luo, Y., *et al.*, 1997. Mammalian two-hybrid system: A complementary approach to the yeast two-hybrid system. *Biotechniques* 22 (2), 350–352.
- Magger, O., *et al.*, 2012. Enhancing the prioritization of disease-causing genes through tissue specific protein interaction networks. *PLOS Comput. Biol.* 8 (9), e1002690.
- Marcotte, E.M., *et al.*, 1999. Detecting protein function and protein-protein interactions from genome sequences. *Science* 285 (5428), 751–753.
- Maruta, N., Trusov, Y., Botella, J.R., 2016. Yeast three-hybrid system for the detection of protein-protein interactions. *Methods Mol. Biol.* 1363, 145–154.
- Memisevic, V., Wallqvist, A., Reifman, J., 2013. Reconstituting protein interaction networks using parameter-dependent domain-domain interactions. *BMC Bioinform.* 14, 154.
- Menche, J., *et al.*, 2015. Disease networks. Uncovering disease-disease relationships through the incomplete interactome. *Science* 347 (6224), 1257601.
- Moal, I.H., *et al.*, 2013. Scoring functions for protein-protein interactions. *Curr. Opin. Struct. Biol.* 23 (6), 862–867.
- Moal, I.H., Agius, R., Bates, P.A., 2011. Protein-protein binding affinity prediction on a diverse set of structures. *Bioinformatics* 27 (21), 3002–3009.
- Muley, V.Y., Ranjan, A., 2012. Effect of reference genome selection on the performance of computational methods for genome-wide protein-protein interaction prediction. *PLOS ONE* 7 (7), e42057.
- Murakami, Y., *et al.*, 2017. Network analysis and in silico prediction of protein-protein interactions with applications in drug discovery. *Curr. Opin. Struct. Biol.* 44, 134–142.
- Murakami, Y., Mizuguchi, K., 2014. Homology-based prediction of interactions between proteins using Averaged One-Dependence Estimators. *BMC Bioinform.* 15, 213.
- Nabieva, E., *et al.*, 2005. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21 (Suppl 1), i302–i310.
- Ng, S.K., *et al.*, 2003. InterDom: A database of putative interacting protein domains for validating predicted protein interactions and complexes. *Nucleic Acids Res.* 31 (1), 251–254.
- Ochoa, D., Pazos, F., 2010. Studying the co-evolution of protein families with the Mirrortree web server. *Bioinformatics* 26 (10), 1370–1371.
- Ochoa, D., Pazos, F., 2014. Practical aspects of protein co-evolution. *Front. Cell Dev. Biol.* 2, 14.
- Papanikolaou, N., *et al.*, 2015. Protein-protein interaction predictions using text mining methods. *Methods* 74, 47–53.
- Park, Y., Marcotte, E.M., 2012. Flaws in evaluation schemes for pair-input computational predictions. *Nat. Methods* 9 (12), 1134–1136.
- Pazos, F., *et al.*, 2005. Assessing protein co-evolution in the context of the tree of life assists in the prediction of the interactome. *J. Mol. Biol.* 352 (4), 1002–1015.
- Pazos, F., Valencia, A., 2001. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Eng.* 14 (9), 609–614.
- Peng, X., *et al.*, 2017. Protein-protein interactions: Detection, reliability assessment and applications. *Brief Bioinform.* 18 (5), 798–819.
- Phanse, S., *et al.*, 2016. Proteome-wide dataset supporting the study of ancient metazoan macromolecular complexes. *Data Brief.* 6, 715–721.
- Pitre, S., *et al.*, 2012. Short Co-occurring polypeptide regions can predict global protein interaction maps. *Sci. Rep.* 2, 239.
- Rajagopala, S.V., *et al.*, 2014. The binary protein-protein interaction landscape of *Escherichia coli*. *Nat. Biotechnol.* 32 (3), 285–290.

- Raman, K., 2010. Construction and analysis of protein-protein interaction networks. *Autom. Exp.* 2 (1), 2.
- Razick, S., Magklaras, G., Donaldson, I.M., 2008. iRefIndex: A consolidated protein interaction database with provenance. *BMC Bioinform.* 9, 405.
- Rhodes, D.R., *et al.*, 2005. Probabilistic model of the human protein-protein interaction network. *Nat. Biotechnol.* 23 (8), 951–959.
- Rolland, T., *et al.*, 2014. A proteome-scale map of the human interactome network. *Cell* 159 (5), 1212–1226.
- Sahni, N., *et al.*, 2015. Widespread macromolecular interaction perturbations in human genetic disorders. *Cell* 161 (3), 647–660.
- Salwinski, L., *et al.*, 2004. The Database of interacting proteins: 2004 update. *Nucleic Acids Res.* 32 (Database issue), D449–D451.
- Sarajlic, A., *et al.*, 2013. Network topology reveals key cardiovascular disease genes. *PLOS ONE* 8 (8), e71537.
- Sato, T., *et al.*, 2005. The inference of protein-protein interactions by co-evolutionary analysis is improved by excluding the information about the phylogenetic relationships. *Bioinformatics* 21 (17), 3482–3489.
- Schaefer, M.H., *et al.*, 2013. Adding protein context to the human protein-protein interaction network to reveal meaningful interactions. *PLOS Comput. Biol.* 9 (1), e1002860.
- Schoenrock, A., *et al.*, 2014. Efficient prediction of human protein-protein interactions at a global scale. *BMC Bioinform.* 15, 383.
- Shameer, K., *et al.*, 2016. Interpreting functional effects of coding variants: Challenges in proteome-scale prediction, annotation and assessment. *Brief. Bioinform.* 17 (5), 841–862.
- Shoemaker, B.A., Panchenko, A.R., 2007. Deciphering protein-protein interactions. Part II. Computational methods to predict protein and domain interaction partners. *PLOS Comput. Biol.* 3 (4), e43.
- Sillitoe, I., *et al.*, 2015. CATH: Comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res.* 43 (Database issue), D376–D381.
- Snider, J., *et al.*, 2010. Detecting interactions with membrane proteins using a membrane two-hybrid assay in yeast. *Nat. Protoc.* 5 (7), 1281–1293.
- Sprinzak, E., Margalit, H., 2001. Correlated sequence-signatures as markers of protein-protein interaction. *J. Mol. Biol.* 311 (4), 681–692.
- Szklarczyk, D., *et al.*, 2015. STRING v10: Protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* 43 (Database issue), D447. v52.
- Szklarczyk, D., *et al.*, 2017. The STRING database in 2017: Quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res.* 45 (D1), D362–D368.
- Tamames, J., *et al.*, 1997. Conserved clusters of functionally related genes in two bacterial genomes. *J. Mol. Evol.* 44 (1), 66–73.
- Thieu, T., *et al.*, 2012. Literature mining of host-pathogen interactions: Comparing feature-based supervised learning and language-based approaches. *Bioinformatics* 28 (6), 867–875.
- Trinkle-Mulcahy, L., *et al.*, 2008. Identifying specific protein interaction partners using quantitative mass spectrometry and bead proteomes. *J. Cell Biol.* 183 (2), 223–239.
- Tuncbag, N., *et al.*, 2009. A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. *Brief Bioinform.* 10 (3), 217–232.
- Uetz, P., *et al.*, 2000. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature* 403 (6770), 623–627.
- Vakser, I.A., 2014. Protein-protein docking: From interaction to interactome. *Biophys. J.* 107 (8), 1785–1793.
- Vidal, M., Cusick, M.E., Barabasi, A.L., 2011. Interactome networks and human disease. *Cell* 144 (6), 986–998.
- Vo, T.V., *et al.*, 2016. A Proteome-wide fission yeast interactome reveals network evolution principles from yeasts to human. *Cell* 164 (1–2), 310–323.
- Watson, J.D., Laskowski, R.A., Thornton, J.M., 2005. Predicting protein function from sequence and structural data. *Curr. Opin. Struct. Biol.* 15 (3), 275–284.
- Webber, C., Ponting, C.P., 2004. Genes and homology. *Curr. Biol.* 14 (9), R332–R333.
- Wells, J.A., McClendon, C.L., 2007. Reaching for high-hanging fruit in drug discovery at protein-protein interfaces. *Nature* 450 (7172), 1001–1009.
- Wiles, A.M., *et al.*, 2010. Building and analyzing protein interactome networks by cross-species comparisons. *BMC Syst. Biol.* 4, 36.
- Xu, J., Li, Y., 2006. Discovering disease-genes by topological features in human protein-protein interaction network. *Bioinformatics* 22 (22), 2800–2805.
- Yang, J., Wagner, S.A., Beli, P., 2015. Illuminating spatial and temporal organization of protein interaction networks by mass spectrometry-based proteomics. *Front. Genet.* 6, 344.
- Yu, H., *et al.*, 2007. The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLOS Comput. Biol.* 3 (4), e59.
- Yu, H., *et al.*, 2008. High-quality binary protein interaction map of the yeast interactome network. *Science* 322 (5898), 104–110.
- Zahiri, J., Bozorgmehr, J.H., Masoudi-Nejad, A., 2013. Computational prediction of protein-protein interaction networks: Algorithms and resources. *Curr. Genom.* 14 (6), 397–414.

Sequence Analysis

Andrey D Prijbelski, Anton I Korobeynikov, and Alla L Lapidus, St. Petersburg State University, St Petersburg, Russia

© 2019 Elsevier Inc. All rights reserved.

Glossary

Adjusted p-value The lowest familywise significance level, at which a certain comparison will be considered as statistically significant as part of the multiple comparison testing.

Alignment score A measure that reflects the level of similarity between sequences.

Contig Long genomic fragment, usually generated by the genome assembly software.

Copy number variation (CNV) An alteration between the number of the same genomic fragment being repeated in the genome within the population.

de Bruijn graph An approach used for *de novo* sequence assembly from short reads.

De novo genome assembly The process of constructing the genome sequence directly from the sequencing reads without any supplementary information.

Differential expression analysis A process of studying changes in expression of the genes between samples, usually under different conditions.

Duplication A rearrangement event in which a genomic fragment is copied into the same chromosome.

Edit distance The minimum number of operations (insertions, deletions or substitutions) needed to turn one sequence into another.

FASTA format A text-based file format for storing biological sequences.

GC content The percentage of cytosine and guanine nucleotides relative to the total length of nucleic sequence.

Genome rearrangement A large genomic mutation, which alternates the whole chromosomes or their long fragments.

Graph simplification The process of removing erroneous edges from the graph (usually referred to de Bruijn graph).

Hamming distance The number of positions in two strings of the same length, in which the symbols do not match.

Hybrid assembly An approach for *de novo genome assembly* using multiple different sequencing technologies simultaneously.

Indel Insertion or deletion.

Inversion A rearrangement event in which a genomic fragment is reversed within the chromosome.

k-mer A sequence of length k.

Metagenomics A field of genomics, which studies genomes of the entire bacterial communities (or other samples containing different species).

Misassembly An error in the assembly when two distant genomic fragments are joined into a single contigs/scaffold.

N50 The maximum length X for which the collection of all contigs of length $\geq X$ covers at least 50% of the assembly.

Overlap-Layout-Consensus approach (OLC) An *de novo* genome assembly method used for long sequencing reads.

Paired-end reads A couple of reads sequenced from the different ends of the same genomic fragment (typically from different strands); also can be referred to as left and right reads, or *mate reads*.

Phylogenetic tree A tree that demonstrates evolutionary relationships between selected biological species.

Read alignment (read mapping) The process of aligning sequencing reads to a longer sequence (typically reference genome) to detect the position, from which the read was originally sequences.

Read mapping See *read alignment*.

Repeat resolution The process resolving ambiguities during the assembly, which are caused by the genomic repeats.

SAM/BAM formats Formats for storing read alignment data. SAM is a text-based format and BAM is a compressed binary format.

Scaffold An ordered set of contigs, typically separated by regions with unknown bases (N).

Scaffolding The process of ordering *contigs* into *scaffolds*.

Sequence alignment The process of detecting similarities between biological sequences.

Single nucleotide polymorphism (SNP) A variation in a single position in the genome that appears in appreciable part of the population.

Strand-specific RNA-Seq A method for sequencing RNA molecules, from which it is possible to deduce the strand on which the gene is located.

String graph An approach used for *de novo* sequence assembly from short reads.

Synteny block A set of consecutive genes on the chromosome, which are typically inherited together and preserved by rearrangement events.

Translocation A rearrangement event in which a genomic fragment is exchanged with a fragment from another chromosome.

Universal single copy-ortholog A gene, which does not have paralogs and is typical for a given set of organisms.

VCF/BCF formats Formats for storing variations and mutations. VCF is a text-based format and BCF is a compressed binary format.

Introduction to Sequence Analysis

Sequence analysis is a term that comprehensively represents computational analysis of a DNA, RNA or peptide sequence, to extract knowledge about its properties, biological function, structure and evolution. To carry out sequence analysis efficiently, it is important to first understand the source of the data, i.e., the different experimental methods used for determining the biological

sequence. We then need to follow analytical strategies, depending on whether the sequence is genomic, transcriptomic or proteomic. Databases currently warehousing the enormous data on these biomolecules will need to be first checked for the presence of similar sequences, which might direct experimental assays for functional investigations. Software tools and web services are often used for carrying of the bioinformatics analysis. After analysis of DNA, RNA and protein sequences, it is important to understand how they are connected by protein to genome mapping. The small organic molecules or metabolites that are essential for organisms to live and grow also need to be studied in the context of their interaction with genes and proteins, *via* metabolic pathways. This article aims to provide an overview of sequence analysis, at the DNA, RNA and proteins levels, with metabolic pathways describing their interplay.

Nucleic Acid Sequencing

DNA Sequencing Technologies

DNA sequencing is the process of determining the order of nucleotide bases of molecules of DNA. If the DNA of the whole genome of an organism is used as the DNA of interest, then the process is referred to as genomic sequencing. This process is currently widely used both in fundamental biological research and in many applications. Information about the primary DNA sequence is a crucial piece of data used in such areas of research as medical studies, gene therapy, drug development, evolutionary studies, agriculture, improvement of nutritional quality, biotechnology, forensic and many other.

First-generation sequencing

The era of DNA sequencing began about 30 years ago, when two methods of primary DNA structure determination appeared almost simultaneously. The one developed by A. Maxam and W. Gilbert at Harvard University was based on the use of chemical modifications of DNA. The other method was published by F. Sanger and A. Coulson from Cambridge and is called the chain-termination method. These methods were named 1st generation sequencing (Heather and Chain, 2016).

And although over time, both the Gilbert and the Sanger DNA sequencing methods have gone through a number of significant changes (one of which was the automation of the Sanger DNA method that assured its wider application), first generation sequencing methods continued to remain labor intensive, had a low throughput speed and were very expensive. To overcome these limitations and in response to the National Human Genome Research Institute (NHGRI) call to lower the cost of sequencing of an individual human genome to \$1000, with an intermediate goal of reducing the cost of sequencing a mammalian-sized genome to \$100,000, next generation sequencing technologies (NGS) were rapidly developed and began to be used in pyrosequencing, base-by-base sequencing by synthesis (SBS), sequencing by ligation, nanopore sequencing, and single-molecule sequencing by synthesis in real time strategies.

These newest sequencing technologies that are currently at different stages of development and implementation can be split into two groups: Second generation sequencing and the third generation approaches.

Second-generation sequencing

Compared to the traditional Sanger sequencing, the greatest advantage of the second generation sequencing is a very high level of parallelization: up to 10^7 reactions in one experiment. In addition, the minuscule volume of the individual wells (picoliters), within which the sequencing reactions are run and the high data density (10,000 times higher, than in case of the latest microelectrophoresis-based capillary instruments) significantly decrease the volume of reagents needed to perform the reactions. Another positive aspect of the second generation methods is the ability to circumvent DNA cloning and propagation in *Escherichia coli* cells, thus avoiding the problems of biased genome coverage due to the presence of genomic areas, which are hard or even impossible to clone in *E. coli* (so called unclonable areas). The downside of these technologies, as compared with the Sanger approach, however, is the length of the produced reads.

The 454 Genome Sequencer 20 System instrument (the first commercial NGS platform) produced reads of about 110 bp. The latest GS FLX Titanium chemistry (Margulies *et al.*, 2005) has managed to increase read length up to 1 kb (454 Life Sciences was shut down in 2013 and 454 sequencing instruments are not supported any longer). Meanwhile, technologies using SBS and sequencing by ligation (Applied Biosystem SOLiD; polony-based approaches) produced reads of only 25–50 bp long (Shendure *et al.*, 2005; Valouev *et al.*, 2008). This meant that although 454 sequencing has been successfully applied to de novo genome sequencing resulting in a gapped assembly with an error rate of about 1/3000–1/5000 bp, the ultra-short reads produced by Solexa technology appeared to be of use for resequencing purposes only, in this case a high-quality reference sequence would be present for alignment. The use of 25–50 bp reads for de novo genome sequencing appeared to be very problematic.

After being acquired by Illumina, the Solexa technology was renamed to Illumina technology. This approach uses the SBS sequencing to generate massively parallel genomic data and detect single bases as they are incorporated into growing DNA strands. Read lengths vary from 100 bp to 300 bp depending on the sequencing instrument used (MiSeq, NextSeq 500, different HiSeq). The low error rate of Illumina reads ($\leq 0.1\%$) makes this sequencing technology very attractive for a number of applications including de novo genome assembly.

Ion Torrent is yet another NGS approach (Rothberg *et al.*, 2011). It uses the semiconductor sequencing technology of SBS. The Ion PGM sequencer detects changes in pH when a nucleotide is incorporated into the DNA molecules and a proton is released.

This technology produces reads of 400 bp long with about 1% error rate, allowing single nucleotide polymorphism (SNP) detection as well as whole genome assembly.

The second Ion Torrent instrument released in 2012, the Ion Proton, produces shorter reads (200 bp), but provides 10 times larger (~10 Gb) output than its predecessor. The read length provided by recently developed Ion S5 and Ion S5 XL Systems (www.thermofisher.com) is ranging from 200 bp to 600 bp depending on the chip type and the output can be as high as 10–15 Gb.

Third-generation sequencing

Among the numerous methods on the drawing board, special attention should be given to real-time sequencing of DNA molecules. Such methods use very small amounts of genomic DNA and require supersensitive detection methods. The success of these methods wholly depends on the quality of the genomic DNA used in the experiment: if the DNA is damaged while being isolated, the resulting sequence will be corrupt. In theory, the read length produced by this type of technology should be equal to that of a full genome.

PacBio (SMRT) sequencing is one of the first real-time single-molecule sequencing techniques capable of generating long reads (10–15 Kb) without requiring any amplification steps. Additionally, its unique ability to distinguish methylated bases from normal nucleotides (Levene *et al.*, 2003) opens up the possibility of studying the epigenetic potential of different organisms. Read error rate is high, but errors are random and thus can be corrected via an increase in coverage. Long reads play a crucial role in the gap closing step of the whole genome assembly and PacBio gave genome finishing, a step that for several years was set aside as time-consuming and not cost efficient, a much needed boost.

The UK company Oxford Nanopore Technologies developed a next generation sequencing approach based on nanopores. The MinION system allows to produce more than 3 Gbp of data from a single run. The resulting reads are of a variable length and can be up to 900,000 bases long. The current error rate of nanopore reads is 5%–15%. It was observed that this technology produces systematic sequencing errors, which end up being more challenging to correct bioinformatically than random ones.

The list of companies and university-based groups of researchers working on improving sequencing technology goes on and on. Some of them have already managed to achieve commercial success, others are still only working on approaching this goal and many have either already failed to do so or will end up failing eventually (Lapidus, 2015). In most cases a combination of different sequencing approaches allows to produce a high-quality sequence fairly quickly and at very low cost.

Rna-Seq

RNA molecules play a significant role in all living cells. Messenger RNA (mRNA) for example is responsible for translating the genetic information stored in DNA into proteins composed of different combinations of 20 amino acids, each of which has its own type of transfer RNA (tRNA) that binds and transports them to the growing polypeptide chain when needed. Ribosomal RNA (rRNA) is an integral part of both large and small subunits of ribosomes – A complex molecule responsible for protein synthesis in cells.

Transcriptomics allows to study the nature and amount of each of the RNA molecules in a given cell under a specific condition and at a given moment. The RNA sequencing (RNA-seq) approach uses next-generation sequencing (NGS) to perform direct sequencing of cDNA fragments (Hrdlickova *et al.*, 2017).

RNA-seq can detect: (i) Gene expression level; (ii) transcript abundance; (iii) alternative splicing that causes transcriptional diversity in eukaryotes; (iv) different isoform usage; (v) single nucleotide variations; (vi) fusion genes; (vii) post-transcriptional modifications. It can identify genes, exons, splicing events, ncRNAs, novel genes or transcripts.

A typical RNA-Seq experiment consists of RNA isolation and purification, assessment of the RNA purity and yield, enrichment of target RNAs (through polyA selection, size selection, or removal of non-target sequences through hybridization to probes), fragmentation of RNA, amplification, sequencing, followed by data analysis that in its turn includes quality control, read alignment to a reference genome or known transcripts, transcript quantification and *differential expression analysis* (Kukurba, 2015).

Most of the methods used for RNA-seq data analysis rely on well-annotated reference genomes and on accurate gene models of model organisms. The *de novo* transcriptome assembly approach helps to reconstruct the sequence of the expressed transcripts and is used when reference genomes are not available.

Sequencing Data Formats

The simplest and one of the most common formats for storing nucleic and amino acid sequences is the *FASTA format* (see “Relevant Websites section”). Typically FASTA files have “.fasta” or “.fa” extensions, but other extensions, such as “.fna” for nucleic sequences and “.faa” for proteins are also sometimes used. A single entry in a FASTA file contains the sequence name (sequence ID) and the sequence itself. Sequence names may also contain some meta information, such as for example sequence length. When the sequence is long, it is stored in multiple lines for added legibility. FASTA files can contain an unlimited number of sequence entries.

Sequencing reads can also be stored in the *FASTQ format* (see “Relevant Websites section”) that contains information about reads quality in addition to read sequence and read ID. In the process of sequencing, the sequencer can estimate the probability of each nucleotide being detected incorrectly. The decimal logarithm of the error probability is then rounded to the nearest integer,

added to a fixed offset and written to the FASTQ file as the corresponding ASCII character. However, there are several different formats for storing quality values, which vary by offset values and the way the error probability is converted to an integer number. Read ID in FASTQ file may contain additional information about the reads. For example, Illumina sequencer stores flowcell lane number, tile number, coordinates of the cluster and the information about read pairing (if reads are *paired*).

There are two possible ways to store *paired-end reads* in FASTQ files. The most common method stores left and right reads in the two separate files. The order of the reads in both files should be the same, i.e., first read from the file with left reads is a mate for the first read from the file with right reads. Paired reads can also be saved into a single file (called *interlaced FASTQ file*), in which left and right reads are placed one after another.

Since both FASTA and FASTQ formats are text formats, they typically require a huge amount of disk space. To save space, FASTQ files are usually either compressed using standard archiver software, or converted to SRA or unmapped BAM binary formats.

Beside FASTA/FASTQ, there exists a large variety of file formats suitable for storing supplementary information, such as sequence alignments, genomic variations and genome annotation. Some of these formats will be described further throughout the article.

Analyzing Genomic Sequences

Sequence Alignment

Classic alignment algorithms

Sequence alignment is the process of comparing and detecting similarities between biological sequences. What “similarities” are being detected will depend on the goals of the particular alignment process. Sequence alignment appears to be extremely useful in a number of bioinformatics applications.

For example, the simplest way to compare two sequences of the same length is to calculate the number of matching symbols. The value that measures the degree of sequence similarity is called the *alignment score* of two sequences. The opposite value, corresponding to the level of dissimilarity between sequences, is usually referred to as the *distance* between sequences. The number of non-matching characters is called the *Hamming distance*. [Fig. 1](#) shows an example of two sequences with Hamming distance ([Bookstein et al., 2002](#)) equal to 3.

It is, however, worth noting that comparing sequence characters position by position as described above can barely be referred to as alignment process, since it does not take into account such typical biological events as deletions and insertions. The classical notion of sequence alignment includes calculating the so called *edit distance*, which generally corresponds to the minimal number of substitution, insertions and deletions needed to turn one sequence into another. [Fig. 2](#) demonstrates an example of two sequences with edit distance equal to 3.

The problems of computing edit distance and various types of sequence alignment have exact solutions, e.g., ([Smith and Waterman, 1981](#)) and ([Needleman and Wunsch, 1970](#)) algorithms. Since these algorithms were initially developed for protein-protein alignment and later adapted for DNA sequence alignment, they are described in the section ‘Protein-protein alignment’. In most real-life cases, however, these algorithms appear to be impractical for DNA alignment due to their running time and memory requirements.

BLAST and similar database search methods

To overcome the limitations of the classic alignment algorithms, which are usually used for rather short sequences, new methods and heuristics were developed. By heuristic algorithm we imply a non-exact solution, which works faster than exact algorithms (sometimes much faster) and provides biologically meaningful results.

Undoubtedly, the most popular sequence alignment tool in bioinformatics is the BLAST algorithm ([Altschul et al., 1990](#)) and its variations, such as BLASTN ([Acland et al., 2014](#)) and MegaBLAST ([Morgulis et al., 2008](#)). The paper that describes the first version of the BLAST algorithm now has more than 60 thousand citations and the count keeps going up. Various modifications of the BLAST algorithms are used for aligning amino acid sequences to nucleic sequences (tblastn) and vice versa (blastx). BLAST is

```
ATTCGGATCTCA
ATTCGGCACTGA
```

Fig. 1 Example of two sequences with Hamming distances equal to 3.

```
ATTC-GATCTCA
ATTCGGA-CT-A
```

Fig. 2 Example of two sequences with edit distances equal to 3.

widely used on the NCBI web site for searching sequences in various databases, such as 16s rRNA sequences and the nucleotide collection (all assembled genomes).

Beside the BLAST family algorithms, multiple alignment methods for various goals were developed during the past 20 years. Depending on the biological problem and alignment objective, different methods employ different algorithms and heuristic ideas, such as suffix trees and suffix arrays, k-mer indexing, seed-and-extend approaches and Hidden Markov Models. More information about specific sequence alignment algorithms can be found in Section “De Novo Transcriptome Assembly” (protein sequence alignment) and Section “Sequencing Data Formats” (mapping sequencing reads to the reference genome).

Comparative genomics

Nucleic sequence alignment algorithms are widely used in comparative genomics and phylogenetic studies. Comparative genomics studies similarities between two or more genomes at the level of large *rearrangement* events, such as *inversions*, *duplications*, *translocations*, large insertions and deletions. Since the comparative genomics usually does not take into account small structural variations and *single nucleotide polymorphisms* (SNPs), it requires a specific kind of alignment software. These methods typically detect *synteny blocks* – long sequences shared between genomes being compared. Indeed, those sequences may have differences at the nucleotide level, but are still highly similar overall. The genomes are then represented as a sequence of synteny blocks and rearrangements are detected (Hannenhalli and Pevzner, 1999).

Phylogenetic studies use various multiple sequence alignment methods to detect the level of sequence dissimilarity. The distance between compared sequences is used to construct *phylogenetic trees*, in which the length of the branches typically correspond to the distance between analyzed sequences. To construct a biologically meaningful and realistic tree, various clustering methods can be used as well as different sequences may be provided as input (Felsenstein, 1981; Kumar et al., 1994). Phylogenetic studies can be done using whole genomes and rearrangement events, genes and proteins sequences, or even SNPs for closely related organisms.

Preprocessing Sequencing Data

Quality control of sequencing data

Regardless of what technology, protocol or sample was used to generate sequencing data, quality control remains an integral part of every experiment. When performed correctly during the early stages of a project, quality control helps save time and thus, money. There have been many cases when false conclusions were made due to the poor quality of initial data, and, as a known saying states, “garbage in – garbage out”, meaning that great results do not come from low-quality data.

FastQC is one of the most popular tools for basic quality control of different kinds of sequencing data (Andrews, 2010). FastQC does not require any additional data, such as a reference genome, and the QC is performed based on just a FASTQ file with sequences and corresponding quality values. Below we will take a look at several important statistics produced by FastQC, how to interpret them and what the differences between high- and low- quality data are.

The first and one of the most important plots is the per base sequence quality plot. It demonstrates the average quality value amongst the reads. Illumina reads typically have lower quality and a higher number of sequencing errors towards the end of the read. Fig. 3 demonstrates the plot of a per base sequence quality distribution of a normal Illumina dataset, while Fig. 4 gives an example of low-quality Illumina reads and shows the significant drop in the average quality towards reads’ end.

One of the basic, but very important nucleic sequence properties is the *GC content*. It is calculated as the total number of guanine and cytosine nucleotides relative to the total sequence length. The per sequence GC content plot is usually used to detect the presence of contamination in the sample. In case of non-contaminated sequencing data, the GC content distribution is typically bell-shaped and matches the theoretical distribution (Fig. 5). In the presence of contamination this distribution may contain several separate peaks, each of which would correspond to a different genome with different GC content (Fig. 6).

The following group of statistics and plots is typically used to detect a very common problem – The presence of sequencing adapters in the reads. These statistics include per base sequence content, adapter content, k-mer content and overrepresented sequences. The figures below demonstrate how sequence content plots look when working with clean data (Fig. 7) and with reads containing adapters (Fig. 8).

Adapter and quality trimming

As was demonstrated, sequencing adapters and low-quality ends (for Illumina reads) are the common problems that often arise when analyzing sequencing data. Both of them may significantly affect performance and the quality of the results generated during the next steps of the analysis, such as, for example, genome assembly. Therefore, during the preprocessing step low-quality ends are clipped and the adapters are removed.

One of the tools used for read trimming is called Trimmomatic (Bolger et al., 2014). This tool is capable of clipping reads using quality values and removing adapters from both single-end and paired-end reads. One of the most useful methods for trimming reads is the detection of low-quality ends based on the average quality in the sliding window of a fixed size. A read is cut once the average quality in a window drops below a given threshold. Fig. 9 demonstrates what the per base sequence quality plot looks like once the reads have been trimmed (plot for original reads is shown on Fig. 4). To remove the adapters from the reads a FASTA file with the adapter sequences needs to be fed into Trimmomatic. Other adapter-removing tools, such as cutadapt (Martin, 2011) or NxTrim (O’Connell et al., 2015), can also be used.

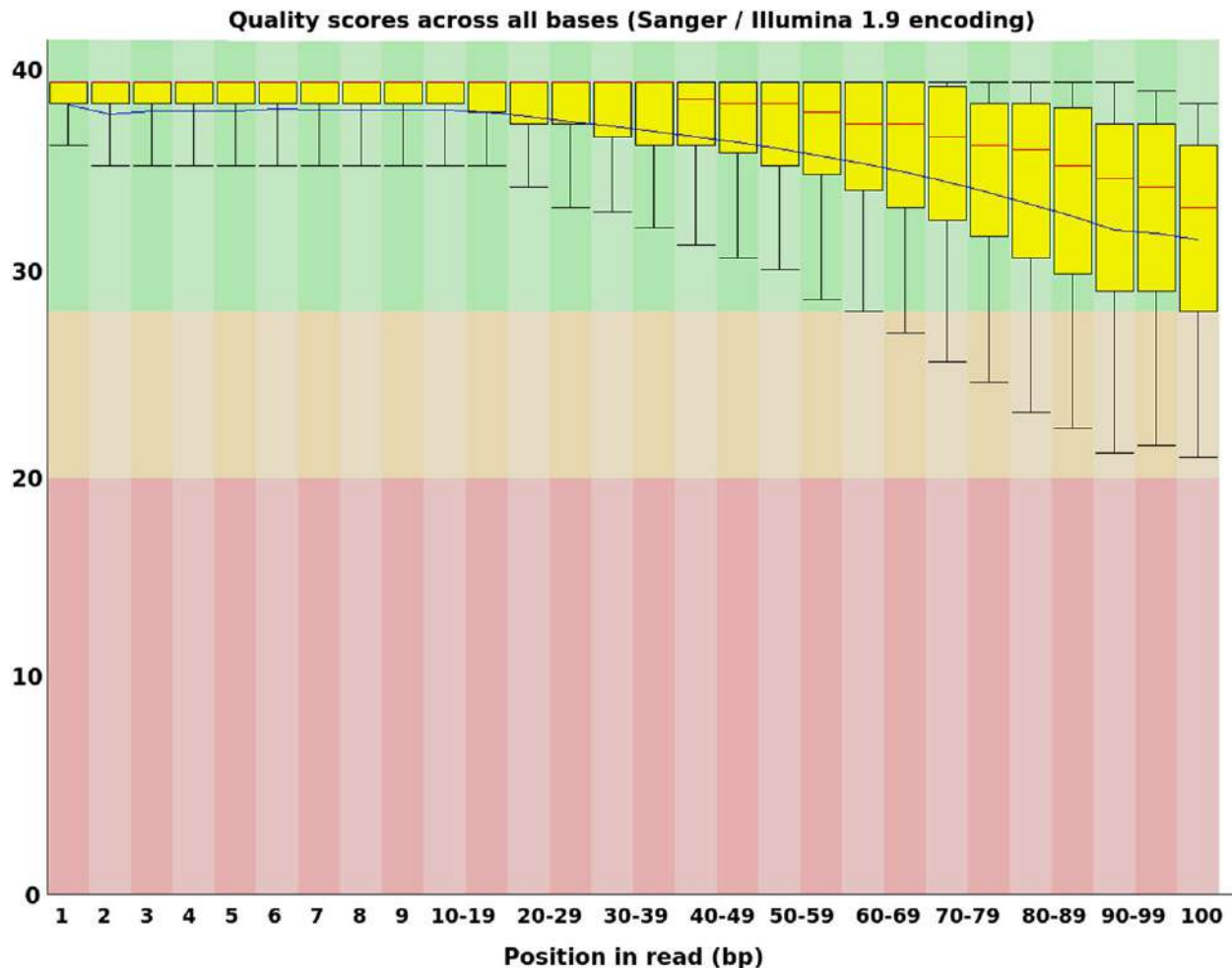


Fig. 3 Per-base sequence quality plot for normal Illumina reads.

Filtering contamination

Another typical problem for sequencing data is contamination – the presence of reads from the genomes of species you did not intend to sequence. Read contamination usually occurs when the sample has not been completely purified and contains the DNA of other organisms. As mentioned above, in cases when the unwanted genome contaminant has a different GC content, its presence can be easily detected using the per sequence GC content plot in FastQC. Otherwise, contamination may be detected using *read alignment* (see next section) or once the genome has been assembled.

In order to filter out the contaminants, one needs to have a database to screen the reads against. In most cases, a reference genome of the suspected contaminant can be used for this purpose. Contaminant reads can be cleaned using read alignment or using a simple tool called *bbduk* from the *bbtools* package (Bushnell, 2014).

Mapping Sequencing Data

Goals and problems of short reads alignment

The problem of aligning a read to a reference genome can be reformulated as an attempt to find the position, from which the read was originally sequenced. *Read alignment* (or *read mapping*) is essential for many biological and medical applications, such as detecting genomic variations in the sequenced organism and phylogenetic studies.

Although both the local and the global alignments have exact solutions, these algorithms have a quadratic running time and memory consumption, which makes them inapplicable for read alignment due to the large size of genomic sequences (e.g., human genome is almost 3 Gbp long). Additionally, a single sequencing library may contain up to 100 million reads. Thus, in order to perform read alignment for the entire dataset in an observable amount of time, every separate read alignment has to be performed extremely efficiently. Several techniques used to speed up read alignment are discussed in the next section.

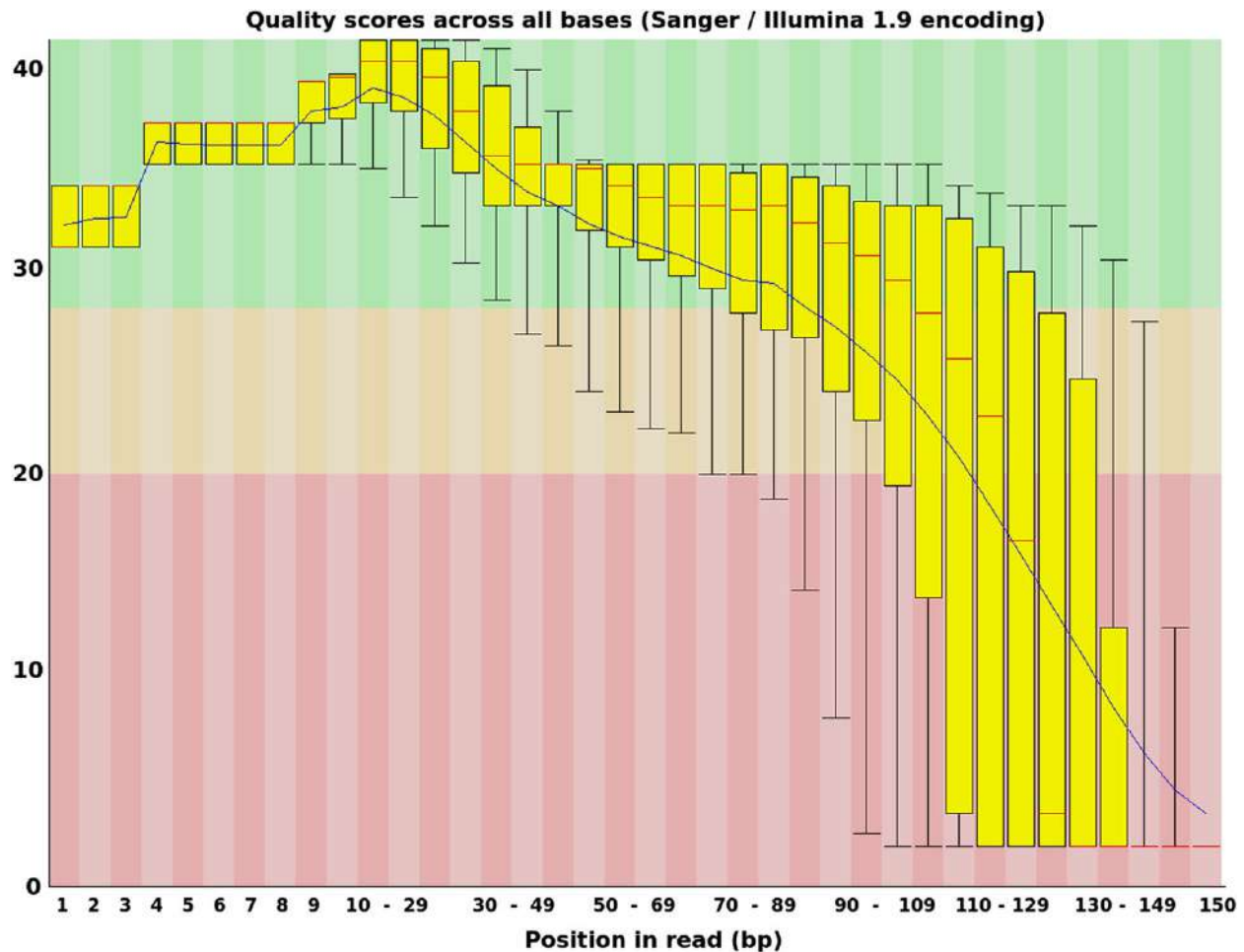


Fig. 4 Per-base sequence quality plot for low-quality Illumina reads.

The problem of read alignment is further complicated by the differences between the reference genome and the genome being analyzed, sequencing errors and genomic repeats. Reads corresponding to repetitive regions can be mapped to several genomic locations and are typically ignored.

In this section we discuss several approaches and tools that are designed for the purpose of aligning reads obtained with different sequencing technologies.

Alignment file formats

Files containing information about the alignment of reads to the reference sequence are typically stored using a specific file format called SAM. SAM files start with a header, which contains all the necessary supplementary information (e.g., lengths of chromosomes in the reference genome). Information about each read alignment is written as a separate line and contains such information as: The position and the strand of the read on the chromosome, name of the chromosome, read ID, the read sequence itself and the corresponding quality string, the information about the mate read (for paired reads) and other supplementary data. Complete information can be found in the SAM format specification.

Since SAM is a text format, it may require a huge amount of disk space, especially when the input files contain hundreds of millions of reads. To reduce disk usage, the binary BAM format was implemented. Typically, BAM files occupy up to 5 times less spaces than SAM files with the same amount of information. SAM/BAM files can be manipulated, converted and processed using a popular package called samtools (Li *et al.*, 2009).

Short read alignment

Modern short read alignment software is based on the same principles as internet search engines – Indexing the content to enable faster query search. Preprocessing of long genomic sequence allows to rapidly search for any of the reads. The genome is often indexed using a suffix array, the Burrows-Wheeler transform, the *FM-index* (Ferragina *et al.*, 2004) or a variation of these algorithms. Since the size of a genome may be large, the preprocessing step may take a significant amount of time, but on the up side, it

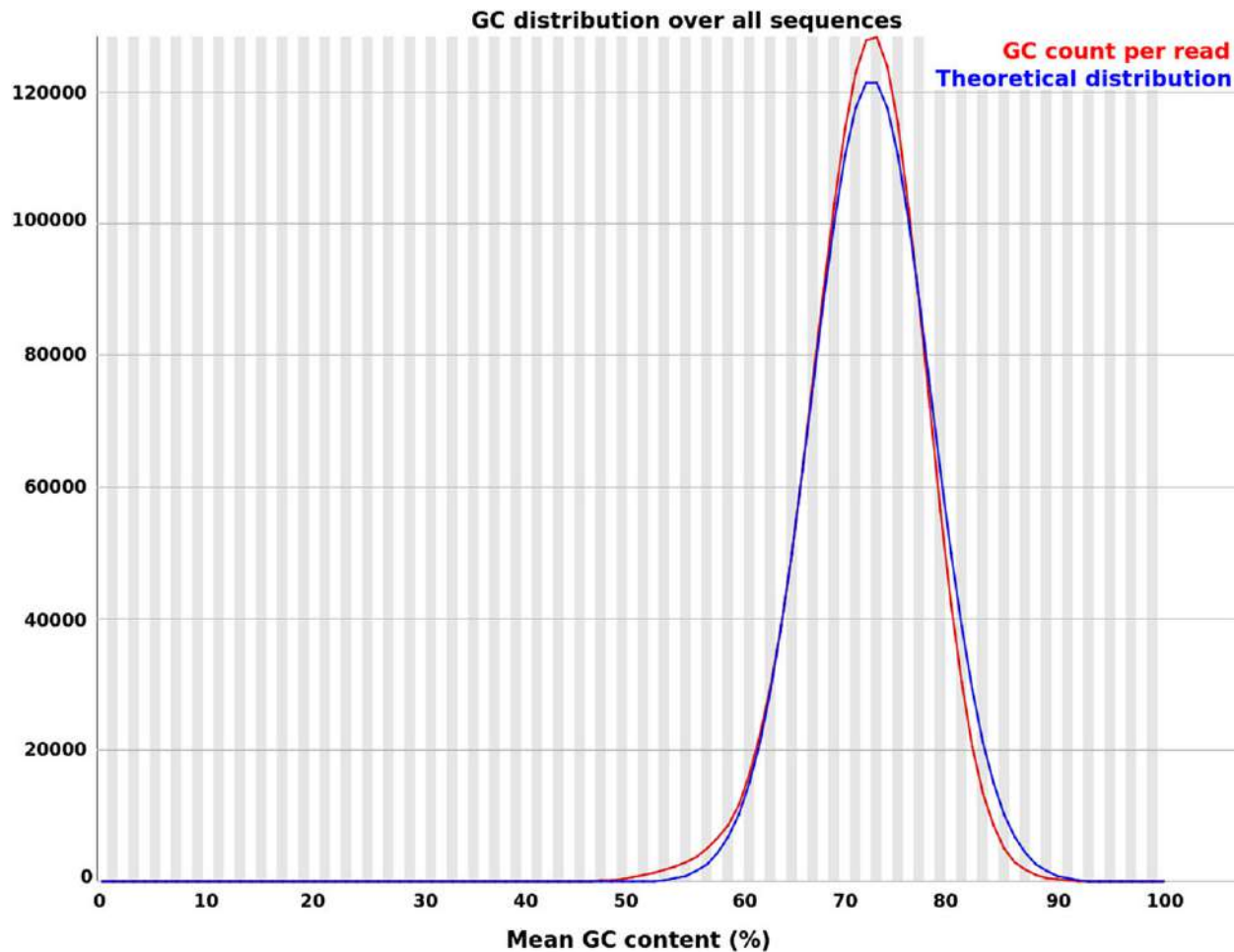


Fig. 5 GC content plot for normal Illumina reads.

needs to be computed only once. Once the index is computed, the read alignment itself can be performed quickly and in parallel, which allows to process reads simultaneously using multiple threads.

The most popular tools for aligning genomic reads are Bowtie2 (Langmead and Salzberg, 2012), various bwa algorithms (Li and Durbin, 2009; Li, 2013), bbmap (Bushnell, 2014) and others. In the case of Ion Torrent sequencing technology TMap can also be used as an alternative. Although comparisons demonstrating the differences in speed and accuracy between a number of popular aligners were published, the final selection of the read alignment software typically comes down to a personal preference based on individual experience.

Variation calling

The main application of short read alignment is detecting different types of genomic variations: *single nucleotide polymorphism* (SNP) sites, small insertions and deletions (*indels*), *copy number variations* (CNV). These kinds of events may have significant biological meaning and thus are of high interest as topics of investigation. For example, a change of single nucleotide in *rpoB* gene of *Mycobacterium tuberculosis* makes it resistant to Rifampicin.

To perform a variation calling, read alignments are sorted, the BAM files are then indexed, the reads are piled up and the variations are detected by comparing them to the reference sequence. Discovered variations are then filtered based on the specific parameters of the final objective. Filtering criteria may depend on such factors as, for example, the average coverage depth of the sequencing library or the organism ploidy. Details on the different variation calling pipelines can be found in the documentation for the Genome Analysis Toolkit (McKenna *et al.*, 2010).

Discovered genomic variations are typically stored in the VCF file format, or in the compressed BCF binary format. The vcftools toolkit is often used to filter and process these files (Danecek *et al.*, 2011). However, the process of calling SNPs on its own does not provide any biologically meaningful insights. In order to understand the effect of the discovered mutation, SNPs need to be annotated. Annotation of the genomic variations can be performed, using several existing databases (e.g., dbSNP, Ensembl Variation) and tools (e.g., SNPeff (Cingolani *et al.*, 2012), ANNOVAR (Wang *et al.*, 2012)), the selection of which will depend on the types of mutation discovered and its expected effect. *De novo* annotation is also possible, but typically requires a large number of samples to be examined in order to determine the associations between the particular variation and the studied effect (e.g., antibiotic resistance).

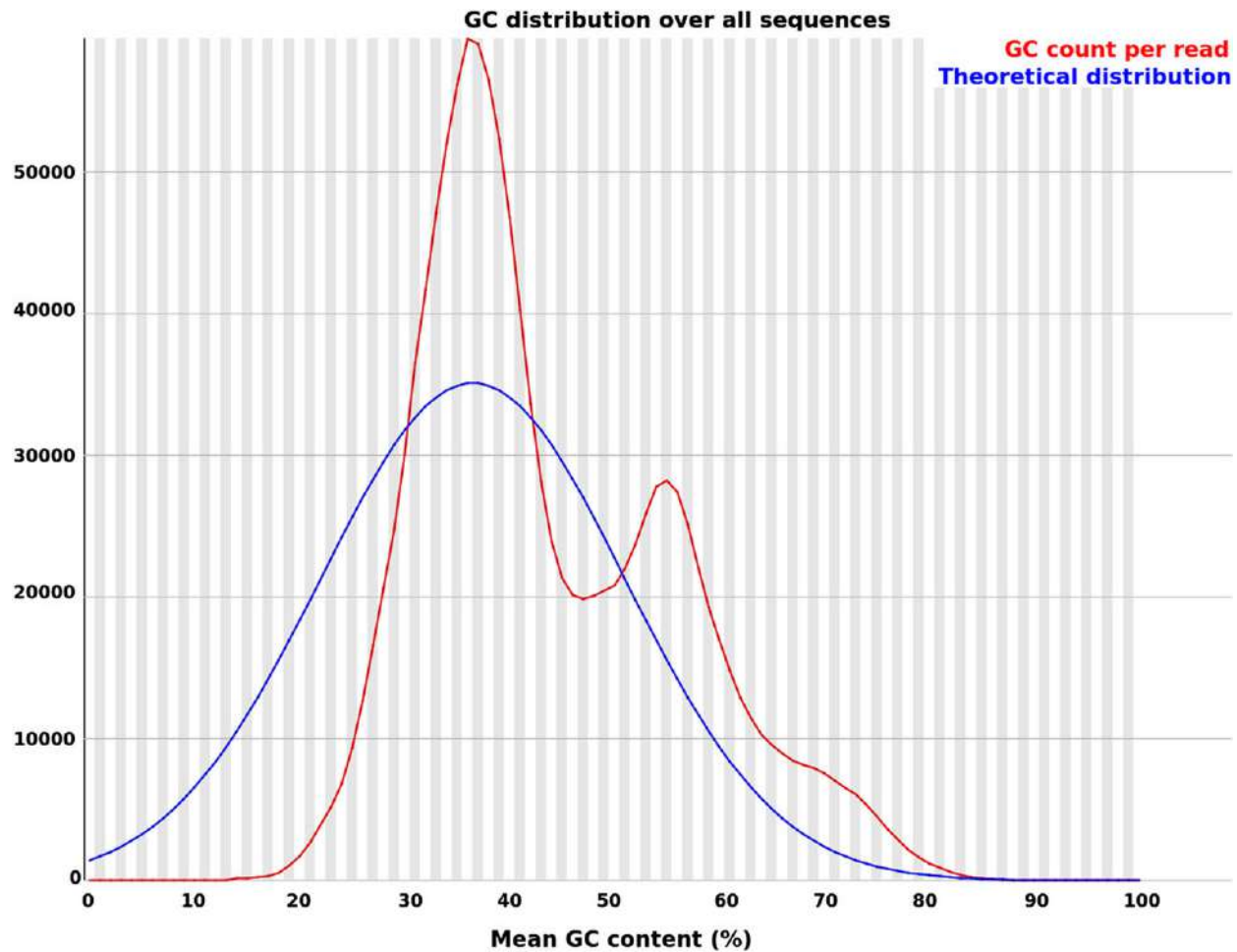


Fig. 6 GC content plot for contaminated Illumina reads.

In addition, once the variations are called, it may be useful to visualize them in a genome browser. Genome browsers are designed to display various genomic features along the chromosomes, such as genes and isoforms. Each genome browser is typically developed to address one particular need and is not universal. For example, to view and examine SNP sites a simple browser called Tablet ([Milne et al., 2009](#)) may be used. Alternatively, more sophisticated browsers, such as IGV ([Thorvaldsdóttir et al., 2013](#)) or UCSC ([Kent et al., 2002](#)) genome browsers can be used.

QC using read alignment

Another possible application of read mapping is an additional QC of sequencing library and calculation of various statistics that cannot be computed without a reference genome. For, example, the number of unaligned, uniquely aligned and multiply aligned reads, fraction of the genome covered and substitution error frequencies.

Read alignments may also be used to estimate coverage depth along the genome. [Figs. 10](#) and [11](#) show two coverage plots for different E.coli sequencing datasets. On both plots each dot represents an average coverage depth in the window of 1 kbp. [Fig. 10](#) demonstrates the coverage distribution for a conventional isolate dataset (uniform coverage distribution and more than 99% of genome covered), while plot on [Fig. 11](#) is constructed using the single-cell MDA-amplified sequencing data (only 96% of genome is covered and coverage depth is highly uneven due to non-uniform amplification process).

Another useful statistic that can be obtained via read alignment is the information about the insert size distribution. [Fig. 12](#) demonstrates the insert size distribution of an E.coli dataset with an average insert size of 215 bp and a standard deviation of 10 bp.

Plots and statistics shown above may be used to identify problems that cannot be detected using basic QC analysis, such as, for example, problems with size selection and coverage gaps.

Long error-prone read alignment

Reads generated by PacBio and Oxford Nanopore technologies significantly differ from those generated by the previous NGS technologies in terms of read length and accuracy. Alignment tools listed in the previous sections were designed especially for short reads with a rather low error rate and did not perform well out-of-the-box.

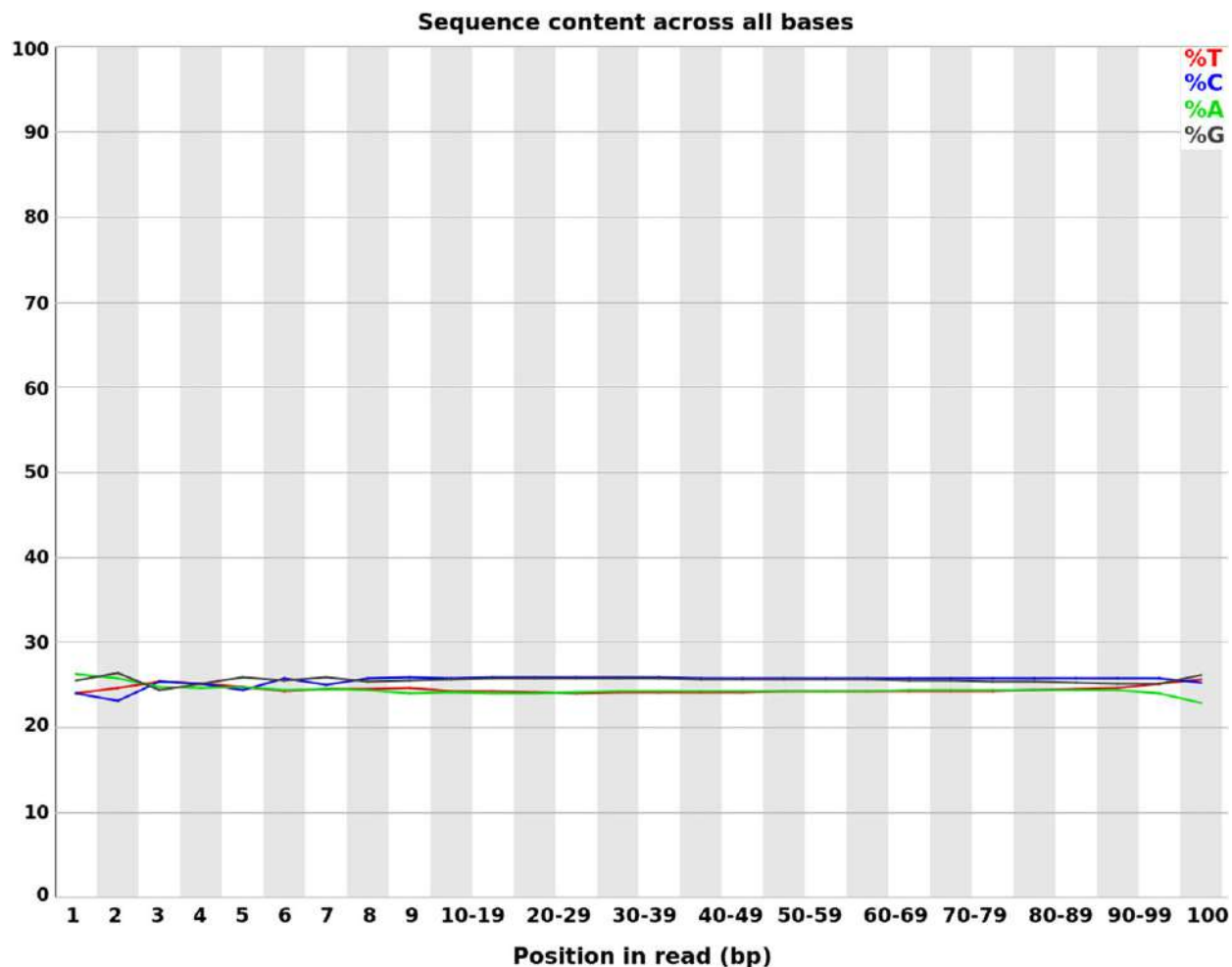


Fig. 7 Per-base sequence content for normal Illumina reads.

Therefore, several new tools were developed for alignment of long error-prone reads, e.g., GraphMap (Sović *et al.*, 2016) and minimap (Li, 2016). The later one is known for mapping reads without outputting actual alignments, which dramatically improves its performance. In addition, the bwa mem algorithm was modified and tuned to support the new family of sequencing technologies. However, due to an extremely high error rate, PacBio and Oxford Nanopores are not the best pieces of technology for calling SNPs and other small variations.

De Novo Genome Assembly

One of the first applications that genome sequencing was intended for is the *de novo genome assembly*. Currently, the problem of assembling a genome from scratch still cannot be considered as completely solved. Even now, in the age of biotechnological and computational progress, a path from the raw biological sample to the finished genome may require years of work and substantial financial resources.

The problem of *de novo* assembly is one of the most popular and algorithmically challenging in computational biology. Dozens of tools were developed during the sequencing era, and yet none of those tools can be named as the best genome assembler in the world for all cases. It is hardly possible to create a universal genome assembler due to the differences in the genomes being sequences, research goals and sequencing technologies.

Despite all recent advances in biotechnology and bioinformatics, it is rarely possible to generate a complete assembly of all chromosomes from raw reads. The key reasons are (i) the presence of genomic repeats and (ii) various sequencing errors and biases. Assembly tools typically generate long genomic regions called *contigs* (stored in FASTA format). The contigs can be further ordered and joined into *scaffolds*, even when the exact genomic sequence and distance between them is unknown. The unspecified nucleotide is usually denoted as an N symbol in the output FASTA file. The scaffolding step is typically performed by long-range sequencing libraries, e.g., mate-pairs.

De novo assembly using long reads

The most intuitive way of solving *de novo* genome assembly problem is to join overlapping reads with each other one by one – in the same way as jigsaw puzzles are assembled. The approach that implements this simple idea is called *Overlap-Layout-Consensus*

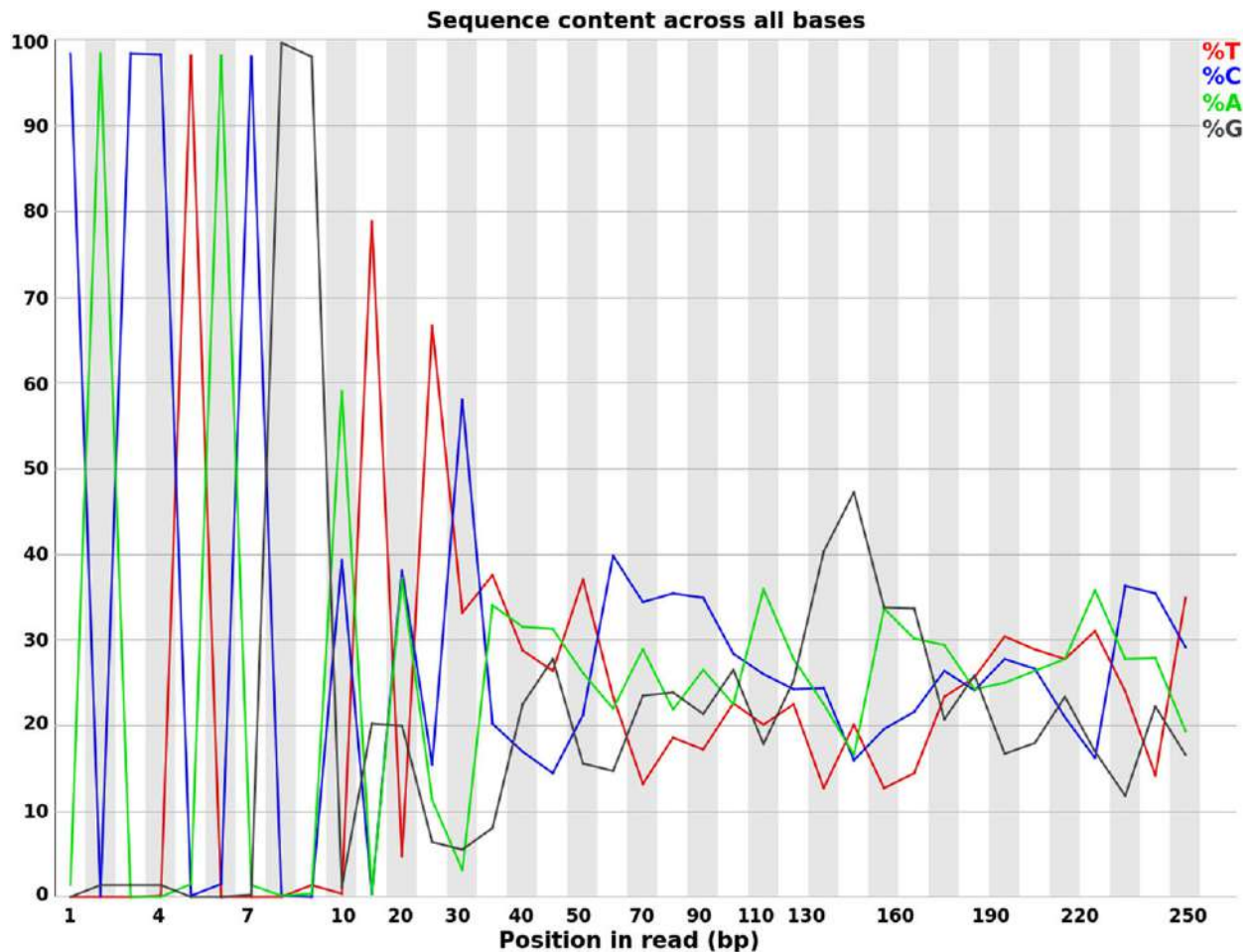


Fig. 8 Per-base sequence content for Illumina reads containing adapters.

(OLC), because it consists of the three consecutive stages. During the first step possible overlaps between all reads are detected, which is typically done using BLAST or similar algorithms. Afterwards, the algorithm constructs an overlap graph in order to determine the layout of the reads, which is then used to compute the consensus sequence and to obtain the resulting contigs. The OLC approach was implemented in multiple popular assembly tools, e.g., Celera (Myers *et al.*, 2000).

As was previously described, Sanger sequencing yields accurate reads up to 1 kbp, typically with low coverage. These characteristics allow to assemble Sanger reads using the OLC approach, which was successfully carried out in multiple groundbreaking genome assembly projects, such as the human genome project. Currently, genomes assembled from Sanger reads with the OLC approach remain the golden standard of genome assembly.

However, this algorithm requires significant computational resources for detecting overlaps between all reads. Although several optimizations can be done to speed up this step, it is still hardly possible to calculate all overlaps for high-covered NGS datasets.

The OLC approach was brought back to life by the recent emergence of the third-generation sequencing technologies – PacBio and Oxford Nanopore. Although Sanger sequencing differs significantly from these novel technologies in terms of error rate and read length, a modified OLC approach appeared to be very useful for their assembly. The main modification compared to the previous implementations of OLC was introduced in the overlap stage, which now implies the alignment of the reads with an extremely high error rate. OLC was successfully reused in such *de novo* assemblers as Canu (Koren *et al.*, 2017) and miniasm (Li, 2016).

De novo assembly using short reads

During the NGS era, sequencing became cheaper, and therefore available to more researchers worldwide. At the same time, it brought multiple algorithmic challenges to computational biology. The problem of *de novo* genome assembly was not an exception. The OLC approach appeared to be impractical for NGS data due to the extreme amount of reads and their short length.

Pevzner *et al.* (2001) proposed to apply the *de Bruijn graph* to the problem of *de novo* genome assembly from short reads. To construct a de Bruijn graph the reads are shredded into a set of smaller sequences of a fixed length k (called *k-mers*), which represent vertices of the graph. In comparison to the overlap graph, the process of constructing a de Bruijn graph does not include

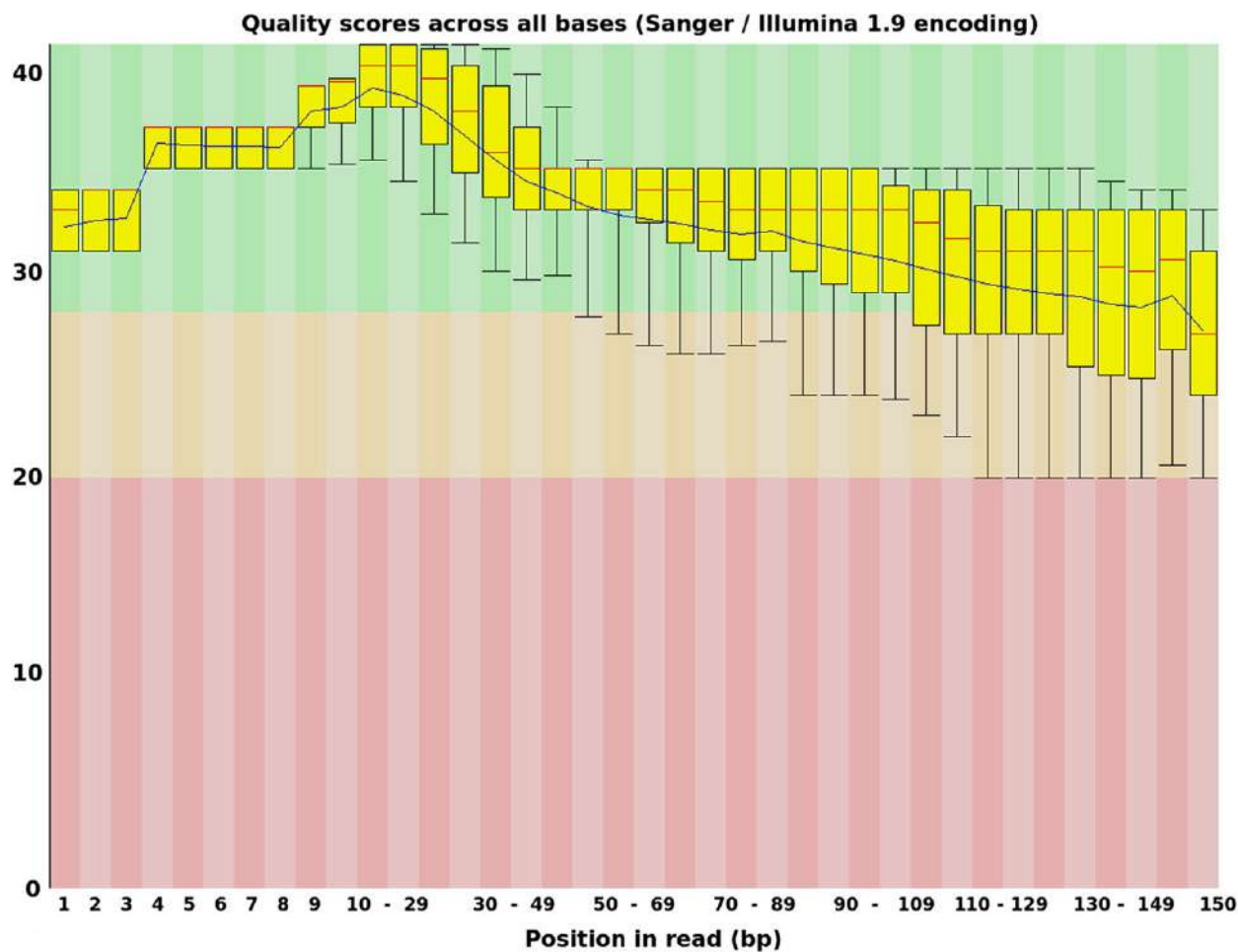


Fig. 9 Per-base sequence quality plot for quality-trimmed Illumina reads.

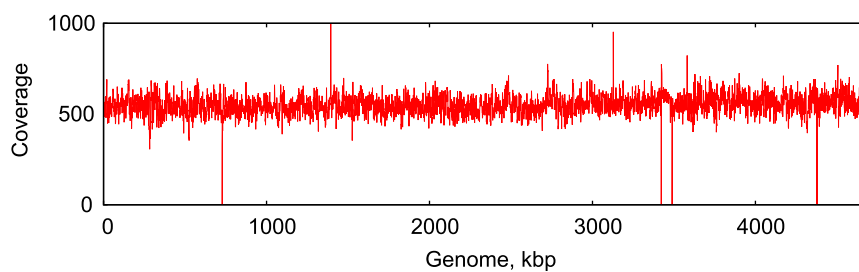


Fig. 10 Coverage depth along the genome for isolate *E.coli* dataset.

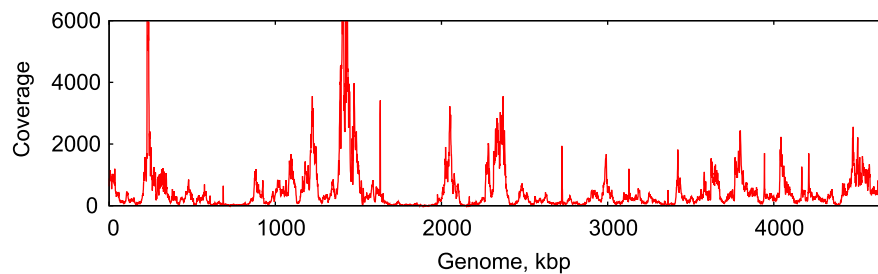


Fig. 11 Coverage depth along the genome for MDA-amplified *E.coli* dataset.

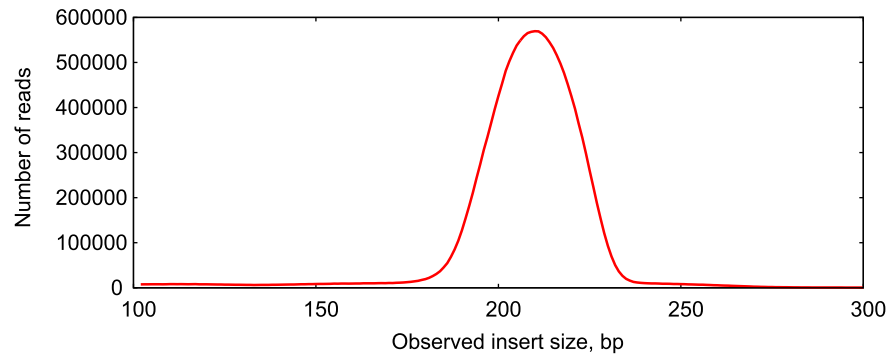


Fig. 12 Insert size distribution for isolate *E.coli* dataset.

read alignment stage, which greatly reduces running time. Since each k-mer is stored only once, the de Bruijn graph approach appears to be more memory-efficient than OLC and therefore suitable for high-coverage datasets with millions of reads.

Although the idea of using the de Bruijn graph was a breakthrough and revolutionized the field of *de novo* genome assembly, its application to real data necessitated the development of several additional algorithms and heuristics. For example, the de Bruijn graph approach appeared to be sensitive to sequencing errors. Each substitution error in a single read creates a number of erroneous non-genomic k-mers, which have to be removed. The process of cleaning the graph from these erroneous k-mers is called *graph simplification*. Once the graph is simplified, *repeat resolution* and *scaffolding* steps are performed to extract contigs and scaffolds from the graph with the help of, for example, read-pairs (Bresler *et al.*, 2012; Pribelski *et al.*, 2014). The de Bruijn graph approach is implemented in many state-of-the-art genome assembly tools, such as Velvet (Zerbino and Birney, 2008), ABySS (Simpson *et al.*, 2009), SOAPdenovo (Luo *et al.*, 2012) and SPAdes (Nurk *et al.*, 2013).

It is worth noting, that later, in 2005, another similar concept called *String graph* was proposed by Myers (2005). In 2010 it was successfully adapted for use with real data and implemented in the SGA tool (Simpson and Durbin, 2012).

Hybrid assembly

Emergence of novel sequencing technologies also brought the possibility to perform *hybrid assembly* – A process of assembling genomes using reads from different sequencing technologies simultaneously. Hybrid assembly may be useful when long-read libraries have low coverage depth (e.g., less than 20x), which may be insufficient for constructing a reliable consensus string using the OLC approach due to the high error-rate of the reads.

Currently, there are two distinct approaches for utilizing short and long reads simultaneously. In the first approach, short accurate NGS reads are assembled using existing assembly tools to obtain accurate contigs with a small number of substitution and indel errors. Afterwards, long error-prone reads are mapped to these contigs and used for repeat resolution and scaffolding. Since PacBio and Oxford Nanopore technologies are capable of yielding reads up to hundreds of kilobases long, they can be useful for resolving long repeats and closing gaps between Illumina contigs. Some algorithms for hybrid assembly also consider aligning long reads directly to the edges of the de Bruijn graph and perform repeat resolution without explicitly outputting the contigs (Antipov *et al.*, 2015; Wick *et al.*, 2017).

Another approach consists of mapping short accurate reads (typically Illumina) onto long error-prone reads in order to correct sequencing errors in the long reads. Once errors are corrected, the long reads can be fed into a conventional OLC assembler to generate long and accurate contigs (Koren *et al.*, 2012).

Genome assembly evaluation

Assessing the quality of the assembled contigs and scaffolds is an important step for both assembly software developers and their users. By evaluating assemblers on various datasets the developers may gain a better understanding of the disadvantages and weak points of their software, improve it and benchmark against other tools to convince the scientific community that their assembler performs better on a specific variety of datasets. This also means that researchers involved in projects requiring genome assembly now face the problem of having to choose the best assembly tools for their particular needs.

One of the first attempts to benchmark genome assembly tools was made by the organizers of the Assemblathon competitions, who suggested the idea of assembling several datasets and then comparing the results submitted by the different teams. Assemblathon 1 offered simulated data, while Assemblathon 2 provided real sequencing data from three vertebral genomes. The resulted contigs were compared using multiple different metrics and the results were then published (Earl *et al.*, 2011; Bradnam *et al.*, 2013).

Benchmarks of various assemblers were also performed by the developers of the assembly tools. However, since until recently no standard tool for evaluating genome assembly quality existed, most of these comparisons were made using custom in-house scripts. In addition, to justify the research, in most cases dataset selection was subjective and the benchmarks typically revealed that the assembler described in the paper outperforms all other tools. However, several benchmarks were performed and published by independent research groups with the goal of identifying the strong and weak points of different genome assembly software (Salzberg *et al.*, 2012; Magoc *et al.*, 2013).

In 2012 GAGE (Salzberg *et al.*, 2012) and QUAST (Gurevich *et al.*, 2013) tools were developed to allow researchers to perform genome assembly quality evaluation on their own. These tools are mostly designed for assessing assemblies of model organisms, i.e., when the reference genome is known. Reference-based quality evaluation allows to perform an excessive analysis of the assemblies by providing a number of informative metrics, such as the fraction of the genome covered, *misassemblies* statistics as well as mismatch and indel error rates.

While reference-based assembly evaluation is useful for testing and comparing genome assemblers, *de novo* evaluation is an integral part of the genome assembly projects. In general, there are three different approaches for reference-free assembly quality assessment. The first one includes calculation of basic statistics (e.g., total assembly length, *N50*) and is useful for initial quality control of the assembly. Another method relies on mapping the raw reads back to the assembly and estimating the likelihood of this particular assembly being generated by the given reads (Hunt *et al.*, 2013). The last group of methods involve counting genes in the assembled sequences. The BUSCO tool (Simão *et al.*, 2015) is designed for detecting the presence of *universal single-copy orthologs*, typical for the specified kingdom, phylum or class. Such tools as GeneMark (Lukashin and Borodovsky, 1998) and Glimmer (Delcher *et al.*, 1999) identify genes *de novo* using Hidden Markov Models (see Section “Genome Annotation” for more information).

Assembly of microbial genomes

Microbial genomes are known for their rather simple repeat structures (compared to the eukaryotic genomes) and therefore are easier to assemble. Since most of the repeats in bacterial genomes do not exceed 7 kbp in length, it is usually possible to assemble a complete bacterial genome with the help of either a mate-pair library (Vasilinets *et al.*, 2015) with a large insert size or long-read sequencing data (Antipov *et al.*, 2015; Wick *et al.*, 2017). Current Illumina Nextera Mate-Pair protocol allows to obtain a complete bacterial genome using only a single short-read sequencing library (Vasilinets *et al.*, 2015). At the same time, a single PacBio or Oxford Nanopore library with a decent coverage depth (e.g., higher than 20x–30x) also allows to obtain a complete bacterial genome in most cases.

To generate a sequencing library for a prokaryotic organism millions of identical cells are needed. However, nowadays most of known bacteria cannot be cultivated in the lab. To generate enough genomic material for the sequencing experiment, bacterial genome from a single cell can be amplified using for example the Multiple Displacement Amplification method (Lizardi and Yale University, 2000). Compared to the conventional isolate datasets, the reads obtained via the MDA technology typically have a highly uneven coverage depth across the genome, which complicates the assembly process. To address the problem of single-cell bacterial assembly SPAdes (Nurk *et al.*, 2013) and IDBA-UD (Peng *et al.*, 2012) assemblers were developed. Recently, a first bacterial genome was completely assembled and finished using only single-cell sequencing data (Woyke, 2010).

Assembling metagenomes

In real life, bacteria typically exist within large communities that may contain thousands of different species. Sequencing and studying the whole community at once is called *metagenomics*. Until recently, metagenomic studies mostly included such research goals as detecting the variety of species (e.g., using 16s rRNA genes) and their abundances within the specific community. The development of MEGAHIT (Li *et al.*, 2015) and metaSPAdes (Nurk *et al.*, 2017) assemblers allows to obtain decent bacterial assemblies from metagenomic datasets.

Metagenome assembly is characterized by such challenges as the presence of related strains within the community, conservative genomic regions shared between different species and uneven coverage depth due to variations in the abundances of the different bacteria in the sample. In addition, metagenomic datasets typically contain enormous amount of reads to capture genomic information for underrepresented bacteria. Since the total length of all genomes within the community may be similar to the length of a mammalian genome, metagenome assembly appears to be an extremely challenging problem not only regarding the quality of the resulting contigs, but in terms of computational resources required as well.

Assembly of eukaryotes

Eukaryotic genomes are known for their complex repeat structures, which complicate the problem of genome assembly by a margin. For example, the human genome contains various repeat families, such as ALUs (short 300 bp repeats with over a million copies in the genome), other short interspersed nuclear elements (SINEs), long interspersed nuclear elements (LINEs, typically about 7 kbp long), terminal repeats and various tandem repeats. To resolve such repeats eukaryotic genomes are assembled using a large variety of sequencing data: mate-pair libraries with different insert sizes, fosmid mate-pairs and genomic maps.

Genome Annotation

Assembled from short sequenced fragments genomic DNA contains all of the information about the metabolism of the organism, its development and responses to external stimuli. Most of these functions are carried out by proteins encoded within the genomic sequence. This means that the primary goal of any genome analysis is to identify protein encoding genes, determine their functions and then define and characterize the sections of DNA that regulate the amounts of these proteins being synthesized based on an exact point in time and conditions. This process is called *genome annotation*. At the start of the annotation process all possible start and end codons that delimit the parameters of protein encoding genes are identified. This stage is called *ORF calling* and will be described in more detail in the “Markov models and gene prediction” section.

It is important to keep in mind that ORF calling methods used for bacterial genomes cannot be applied to eukaryotic genomes, since the latter contain both coding (exons) and non-coding regions (introns). Introns are also significantly longer than exons (e.g., exons take up about 2% of the entire length of the human genome). Information, such as the frequency of encountering particular amino acids and synonymous codons and the knowledge of the length of the exons and introns is used in combination with other statistical data to identify genes in eukaryotic genomes. Despite the fact that genome annotation using statistical identification rules yields only very approximate data, it is still of great value as part of the whole process.

The best way to tackle this task is to compare the sequences of several different, but related genomes. Since exons and introns evolve with different speed (exons, evolve comparatively slower than non-coding segments, aka introns), we can use this information to highlight similar regions by comparing genome sequences to each other and as such, identify the overall location of the exons. The delimitations of these coding regions are then further refined via statistical methods.

Comparative analysis is also of help when trying to predict the function of the particular proteins encoded by the genes. If the level of similarity between the new protein and a previously researched and well described one (studied in a laboratory setting) is high, they can be considered as similar and are likely to have identical functions. The lower the degree of similarity between proteins, the more different their functions will be. To make sure that two protein coding genes from two different genomes are in fact the same gene, we must make sure that neither of these genes has a closer relative in a different genome. This stands true even if the function of these proteins is extremely similar. The quality and accuracy of such analysis is directly dependent on the availability of a large amount of high quality whole genome assembly reference data.

High quality predictions of protein functions play a crucial role in a large number of theoretical and experimental studies, such as for example research aiming to gain a deeper understanding of the underlying biochemical conditions related to numerous genetic diseases.

Not all genes in a genome encode proteins. Thus, instead of synthesizing protein, non-coding RNA genes are responsible for the production of functional RNA products. Non-coding RNA genes can be predicted by using various programs tRNAscan-SE (Lowe and Eddy, 1997), RNAMmer (Lagesen *et al.*, 2007) and BLAST (Altschul *et al.*, 1990) and data sources, as well as performing homology-based searches in such databases as Rfam (Burge *et al.*, 2013), the plant snoRNA database (Brown *et al.*, 2003) (for plant genomes), GenBank (Benson *et al.*, 2004) and ASRG (Wang and Brendel, 2004). RNA sequencing reads (if available) can be further mapped against the appropriate databases to strengthen the generated predictions.

The comparative sequence analysis approach coupled with functional genomics experiments allows annotating non-protein coding genomic regions.

A number of comprehensive annotation systems are available for carrying out functional annotation (see Further reading) for the interested reader to consider and use.

Markov models and gene prediction

Knowledge of the intrinsic gene structure is the main mechanism used in gene prediction. Indeed, the majority of genes have ATG (methionine) as a starting codon, though sometimes GTG and TTG are used as alternative starting codons. Certainly, there may be multiple ATG, GTG, or TGT codons in a frame, which means that the presence of these codons at the beginning of the frame does not necessarily give a proper indication of the translation initiation site. Therefore, another type of DNA structure, also associated with the translation initiation event, has to be used in addition to nucleotide content. One of such features is the ribosomal binding site (RBS) located upstream of the translation start codon. Knowing the ribosome binding site can help to determine the start codon. Each protein coding region ends in a stop codon that causes translation to stop. There are three possible stop codons, namely, TAG, TAA, and TGA, and all three of them are fairly easy to identify.

Manual prokaryotic gene identification relies on brute-force determination of ORFs and major features related to prokaryotic genes. As a part of this approach the DNA is first translated in all six possible frames (three frames on one strand and three frames on the opposite strand). It is worth mentioning that in a noncoding region a stop codon occurs on average every twenty codons, which means that a frame longer than thirty codons without any stop codons could be considered as putative gene coding region. The putative frame is further manually confirmed by the presence of other features such as a start codon and the RBS sequence.

Markov models can be very helpful in providing a probabilistic description of a gene sequence. A *Markov model* describes the probability distribution of nucleotides in a DNA sequence, where the conditional probability of a nucleotide occurring in a particular sequence position depends only on the nucleotides at k previous positions. The value k is called the *order* of a Markov model. This means that a zero-order Markov model implies that each of the bases occur independently with a given probability. This is a typical situation for noncoding parts of a genome. A first-order Markov model (or *Markov chain*) stipulates that the probability of the occurrence of a particular base will depend only on the base preceding it. A second-order model looks at the preceding two bases to determine the probability of occurrence of the following base. This model is a better fit for the codons in a coding sequence.

The use of the Markov models in gene finding exploits the idea that oligonucleotide distributions in the coding regions are different from those for the noncoding regions. Indeed, having the Markov models that describes the structure of coding and non-coding regions (these can be represented with Markov models of different orders) we can calculate the probabilities of occurrence of a particular sequence under these models. Comparing these probabilities or, better yet, calculating the likelihood ratio we could decide whether the particular sequence is a part of a coding region, or not. Since a fixed-order Markov model describes the probability of a particular nucleotide that depends on previous k nucleotides, the longer the nucleotide sequence, the more accurately the internal structure can be described for the coding region. Therefore, the higher the order of a Markov model, the better it can predict genes.

A protein-encoding gene is formed by nucleotides organized in triplets as codons, hence the more effective Markov models are constructed from the sets of three nucleotides, explaining the observed distributions of trimers or hexamers, and so on. The parameters of a Markov model have to be trained using a set of sequences with known gene locations. Once the parameters of the model are established, the model can be used to estimate the probabilities of trimers or hexamers in a new sequence.

Under normal circumstances the pairs of codons tend to correlate, and as a result, the frequency of a particular six nucleotides appearing together in a coding region can be much higher than occurring by random chance. As a result, a fifth-order Markov model, which calculates the probability of hexamer bases, can detect nucleotide correlations typical for coding regions more accurately and is in fact most often used (Wang *et al.*, 2004).

A potential problem of using a high-order Markov chain is that if there are not enough hexamers, which happens in short gene sequences, the method's efficacy might be limited. One way to deal with this limitation is a variable-length Markov model also known as the *interpolated Markov model* (IMM). The IMM samples the largest number of sequence patterns with Markov model orders (k) ranging from 1 to 8 and uses a weighting scheme, placing less weight on rare k -mers (outlining the worse accuracy of corresponding probability estimation) and more weight on more frequent k -mers. The probability produced by the final model is the sum of probabilities of all weighted k -mers. In other words, this method has more flexibility in using Markov models depending on the amount of data available. Higher-order models are used when there is an abundant amount of data and lower-order models are used when the amount of data is smaller (Salzberg *et al.*, 1998).

Surprisingly, the gene content and length distribution of prokaryotic genes can vary a lot. The majority of genes are in the 100–500 amino acids range with a nucleotide distribution derived from the GC content of the organism. However, there are atypical genes that are shorter or longer and have a variety of nucleotide frequency profiles. These genes tend to escape detection using the gene models trained on the typical data. Moreover, the models trained on the whole spectrum of the genes tend to produce worse results due to reduced precision. Ideally, to describe all genes in a genome fully more than one Markov model is needed. The standard way to combine different Markov models (that would represent typical and atypical nucleotide distributions) is via *Hidden Markov Models* (HMM).

A HMM combines two or more Markov models within only one consisting of observed states and others representing unobserved (or “hidden”) states that influence the outcome of the observed states (such unobserved state could be, for example, the type of the gene, or a particular part in the gene the nucleotide composition of which would differ from the other parts). In an HMM, as in a Markov model, the probability of going from one state to another state is the transition probability. Each state may be composed of a number of elements. For nucleotide sequences, there are four possible symbols, namely A, T, G, and C in each state. For amino acid sequences, there are twenty possible symbols. The probability value associated with the occurrence of each of the symbols in each state is called the emission probability. To calculate the total probability of a particular path of the model, both transition and emission probabilities linking all the “hidden” as well as observed states need to be taken into account.

The most popular prokaryotic gene prediction/ORF calling software are: Glimmer is based on interpolated Markov models (Salzberg *et al.*, 1998). GeneMark (Lukashin and Borodovsky, 1998) uses inhomogeneous three-periodic Markov chain models of protein-coding DNA sequences and could be viewed as an approximation of an HMM approach (Azad and Borodovsky, 2004).

Analyzing Transcriptomic Sequences

Quality Control of RNA-Seq

Initial quality control of RNA-Seq data can be performed using the same FastQC tool (Andrews, 2010), as used for genomic data. However, the FastQC reports for RNA-Seq data should be interpreted somewhat differently, compared to the genomic data. The main differentiating criteria for RNA-Seq data is the uneven coverage depth and the presence of ribosomal RNA sequences. Due to their high abundance, rRNA reads may introduce additional peaks into the GC content plot (Fig. 13), as well as create non-uniform k -mer content (Fig. 14).

The presence of unwanted rRNA reads is a typical problem for RNA-Seq experiments. Some sequencing protocols aim to filter out rRNAs during sample preparation, but such approach, however, may only remove a fraction of the rRNA reads from the dataset. Reads from rRNAs can be removed as contamination using, for example, the bbdut script from the bbtools package (see Section “Sequencing Data Formats” for the details), with known rRNA sequences provided as the database.

Aligning RNA Sequences

While alignment of prokaryotic RNA sequences to the reference genome can be performed with the same algorithms and methods used for DNA sequence alignment, the process of mapping eukaryotic RNA differs significantly due to the presence of splicing events. To support long insertions and skip intronic sequences splicing alignment algorithms are required. The most popular software tools for RNA to DNA mapping are BLAT (Kent, 2002), GMAP (Wu and Watanabe, 2005) and Splign (Kapustin *et al.*, 2008).

The problem of mapping eukaryotic RNA-Seq reads to the reference genome is further exacerbated by their short length and the presence of sequencing errors. One of the first tools for mapping RNA-Seq reads – TopHat – was based on the Bowtie aligner for genomic reads (Trapnell *et al.*, 2009). Later, STAR (Dobin *et al.*, 2013) and Hisat (Kim *et al.*, 2015) aligners were developed specifically to address the problem of rapid and accurate mapping of RNA-Seq data. All listed tools can take the gene database in

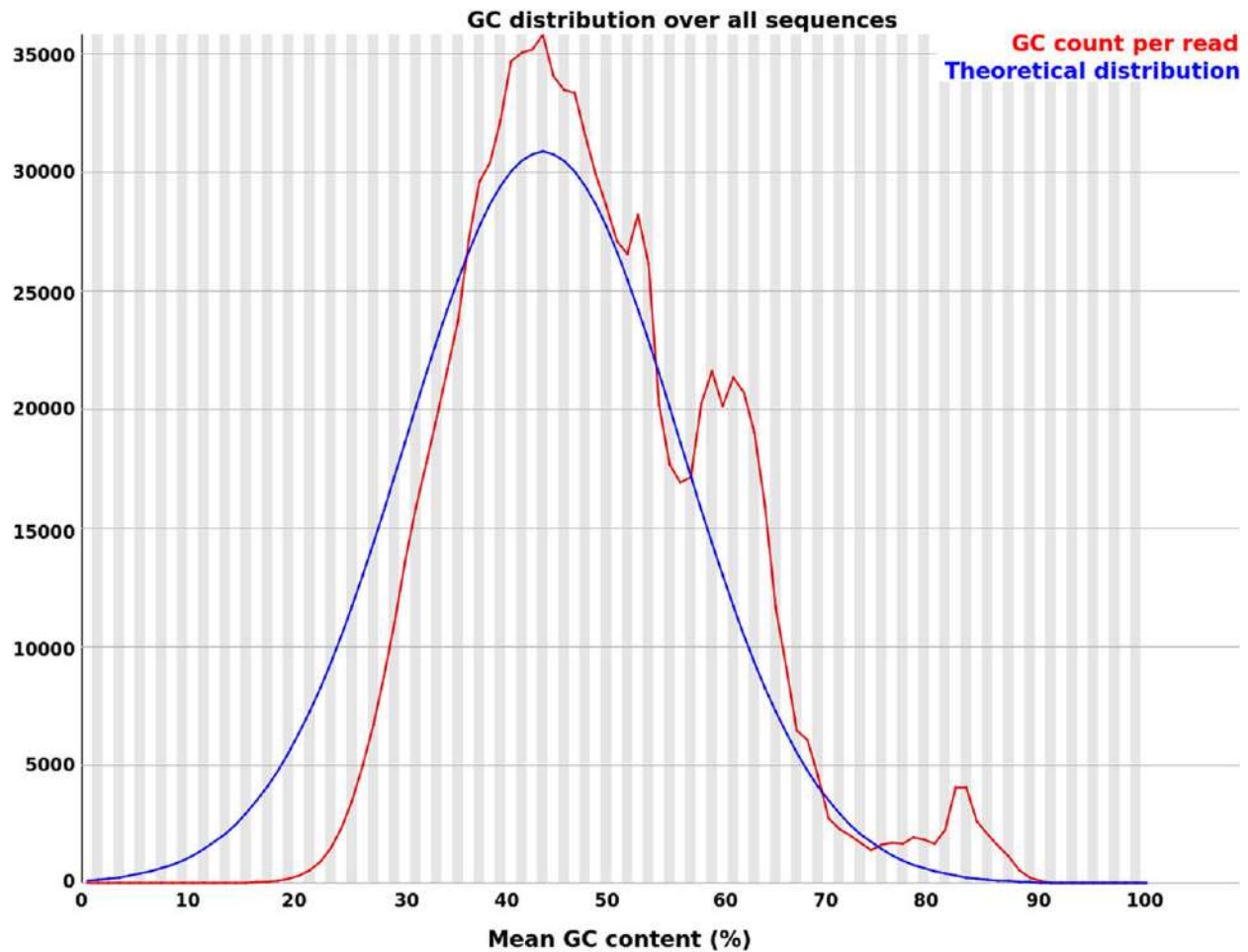


Fig. 13 GC content plot for normal Illumina RNA-Seq reads.

GFF format as input in addition to the reference genome, which increases the accuracy of the alignments and is especially true in the case of reads containing the splice junctions.

Mapping RNA-Seq reads can be of use as an additional reference based quality control. For example, the qualimap ([Okonechnikov et al., 2015](#)) tool calculates such statistics as fractions of exonic, intronic and intergenic reads, gene coverage profile and splice junction distribution. The RSeQC ([Wang et al., 2012](#)) can also be used as an alternative tool. In addition to the statistics listed above, it is capable of determining the insert size for paired-end RNA-Seq reads, estimate mismatch content and detect whether RNA-Seq data is *strand-specific* or not. In strand-specific RNA-Seq experiment the reads are sequenced either from the same or from the opposite strand of the gene that produces the RNA, which then allows to detect actual gene strand. RNA-Seq alignment can also be used to detect fusion genes – chimeric genes that can appear during genomic rearrangements in cancer cells ([Kim and Salzberg, 2011](#)).

Gene Counting and Differential Expression

The main application of RNA-Seq is the estimation of gene expression levels and *differential expression analysis* between multiple samples. Differential expression analysis implies the comparison of expression levels of the same genes/isoforms between samples under different conditions, such as cells under different treatments, healthy and ill patients, patients before and after medical treatment. The number of samples analyzed is the key to a successful analysis as only a correct choice will fully take advantage of RNA-Seq possibilities. To perform the differential expression analysis, a study requires at least three biological samples for each condition analyzed. This is because, when comparing gene expression between the different conditions, sample specific differences may affect the analysis and thus lead to inadequate conclusions, since these differences may not be related to the subject of research. Having an appropriate number of samples for each condition will reduce the impact of sample specific variability on the analysis. Both biological and technical variabilities within the samples can be reduced by normalization. The number of samples analyzed is also essential for the statistical tests required by any differential expression analysis. Most of them need at least three or more samples per condition to provide a statistically sound result. Less samples will suffice only for obtaining exploratory results, while more samples will increase the power of the analysis.

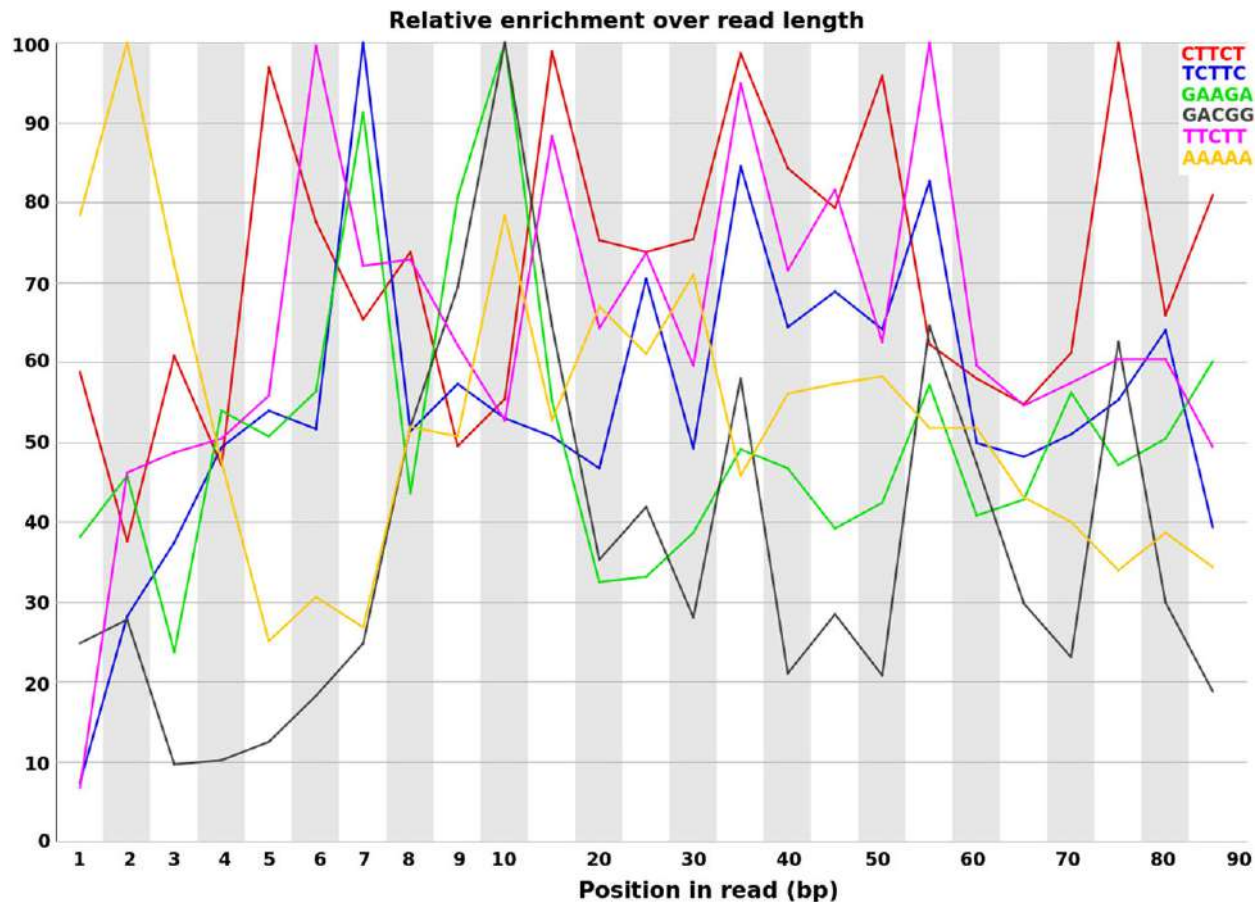


Fig. 14 k-mer distribution for normal Illumina RNA-Seq reads.

Since the isoforms of a gene contain the same exons, assigning each read to a specific isoform is much more complicated task than assigning those reads to the corresponding gene. Therefore, most RNA-Seq pipelines focus on gene level analysis, consisting of several steps.

Firstly, RNA-Seq reads are mapped to the reference genome ignoring reads that map equally well to more than one genomic location (multi-mappers) and assigned to the genes from the database. Afterwards, the number of reads assigned to each gene are counted, e.g., using `htseq-count` (Anders *et al.*, 2015). Some tools also allow to perform quantification without actually mapping the reads to the reference genome (Bray *et al.*, 2016). The gene counts can be normalized, usually by sequencing depth and gene length. The normalized counts are used to estimate relative change of the expression levels between different samples with a statistical test, e.g., a t-test or a test created specifically for RNA-Seq data. State-of-the-art software tools such as DESeq2 (Love *et al.*, 2014) and edgeR (Robinson *et al.*, 2010) require unnormalized raw counts and perform themselves the normalization step.

The results of the differential analysis are typically represented as a table with genes as rows and several values for each gene. Among these numbers, the most important are the measure of the change in gene expression between the conditions (usually a log fold-change) and the result of the statistical test, usually an *adjusted p-value*. Since thousands of tests are made, the adjusted p-value indicates the significance of the difference in expression (p-values are calculated for null-hypotheses stating that the expression level is the same for all conditions). The decision for calling a gene differentially expressed is not straightforward: it depends on arbitrary threshold values for both log fold-change (how much the expression has to change) and adjusted p-value (how many false positives can be tolerated).

Another way of exploring differential expression analysis results is using a heat map (see example in Fig. 15). Table rows correspond to the genes being examined, and each column corresponds to a single sample. Cells color represents estimated genes expression levels. The table is usually sorted according to the change in expression level between different conditions.

One of the possible biases that can significantly affect the results of differential expression analysis is the so called batch effect. The batch effect takes place when the difference between samples is caused by sample preparation or sequencing protocols, flowcell inconsistencies, variations between sequencer runs (Dündar *et al.*, 2015). These factors may lead to inappropriate calculation of expression levels and therefore, misleading interpretation of the results. *In silico* removal of the batch effect is implemented in various differential expression analysis toolkits, e.g., Deseq2 (Love *et al.*, 2014).

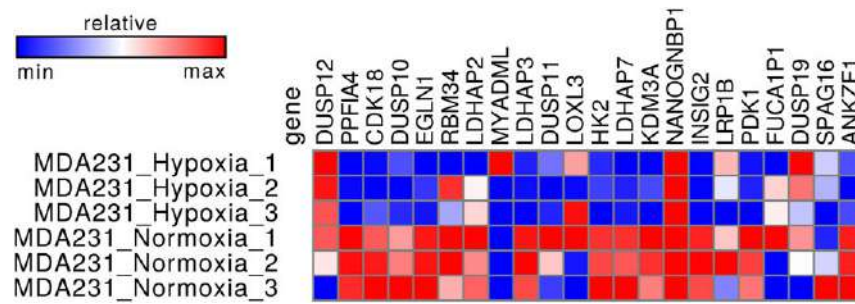


Fig. 15 An example of gene expression heat map for 6 samples under 2 different conditions and 22 genes.

De Novo Transcriptome Assembly

The *de novo* transcriptome assembly may appear to be a simpler problem than genome assembly due to the absence of long complex repeats and a smaller transcriptome total length. However, uneven coverage depth caused by varying expression levels, presence of isoforms from the same gene and paralogous genes complicate the process of transcriptome assembly. Most of the modern *de novo* transcriptome assembly tools, such as Trans-ABYSS (Robertson *et al.*, 2010) and SOAPdenovo-Trans (Xie *et al.*, 2014), are based on the existing de Bruijn graph genome assemblers. Trinity is the only assembler that was initially developed as stand-alone RNA-Seq assembler (Grabherr *et al.*, 2011).

As metagenomic studies became more popular, metatranscriptomics sequencing experiments have also become more frequent. Sequencing RNA of the bacterial community helps gain an understanding of its functionality. In some studies the usage of metatranscriptomic data is limited to the estimation of expression levels for known genes and tracking the signal pathways in the community under different conditions. In some projects, however, the *de novo* metatranscriptome assembly is also performed, although only a few tools are available now (Leung *et al.*, 2015; Ye and Tang, 2015).

Since the result of the *de novo* transcriptome assembly is completely different from the genomic contigs, other methods for quality evaluation are also needed. However, the same three distinct approaches can be applied for the assessment of assembled transcripts: Reference-based, based on gene content and *de novo*.

Reference-based approach includes alignment of the transcripts either to the reference genome with a known gene database via a spliced alignment method or to the isoform sequences themselves. The resulting alignments can be used to compute various statistics and metrics, such as *misassembled* transcripts, number of assembled genes and isoforms, duplication ratio along with mismatch and indel error rate. Recently developed rnaQUAST (Bushmanova *et al.*, 2016) and Transrate (Smith-Unna *et al.*, 2016) tools are designed specifically for the reference-based quality evaluation of *de novo* transcriptome assemblies.

Similar to the genome assemblies, the gene content can be estimated using BUSCO (Simão *et al.*, 2015), which allows to search for single-copy orthologs that are universal for the specific group of organisms. *De novo* gene identification can be performed with a modified version of a popular tool GeneMark, called GeneMarkS-T (Tang *et al.*, 2015). *De novo* quality evaluation based on raw reads for RNA-Seq assemblies is implemented in such tools as Transrate (Smith-Unna *et al.*, 2016) and Detonate (Li *et al.*, 2014).

Protein Data Analysis

All living cells contain proteins that are of a great importance for all biological objects. The structure of proteins is very complex and includes several levels of organization. The primary structure of a protein is usually represented as a string of amino acids starting from the amino-terminal (N-terminal) to the carboxyl-terminal (C-terminal) of the protein. It can be obtained as a result of the direct sequencing of the protein or deduced from the corresponding DNA or mRNA, since the order of bases on a DNA strand specifies the order in which the amino acids are linked together during translation. The amino acid sequence of proteins or peptides is basic information to understand proteins or peptides and predict its post-translational modifications and functions.

Protein Sequencing Technologies

The process of determination of the amino acid sequence is known as protein sequencing. Amino acids may be represented by a single- or three-letter codes. It is worth to mention that sequencing methods were originally developed specifically with proteins in mind, since people thought that proteins were the main genetic material. Researchers believed that the structure of DNA was too simple (composed of only four nucleotides) to play this role comparing to proteins, that can be built of 20 different amino-acids and thus have a higher heterogeneity as molecules. All protein sequencing methods can be divided into two groups: one group provides N-terminus sequence of a protein, while the second group of methods is used to sequence and identify the entirety of the protein. A label-cleavage method for protein sequencing was developed by Pehr Edman in 1950s (Edman *et al.*, 1950).

Edman degradation procedure (Fig. 16) is based on a three-stage reaction that labels and removes the N-terminal residue of a polypeptide, which can then be identified as a phenylthiohydantoin (PTH – Edman reagent) derivative.

Phenyl isothiocyanate, $C_6H_5N=C=S$, is a reagent that permits the progressive removal and identification of the N-terminal amino acid, leaving the rest of the chain unaffected. The truncated peptide after the first degradation has a new N-terminal amino acid that can again react with phenyl isothiocyanate and thus makes progressive removal of the subsequent amino acids possible. This technique cannot be applied if the N-terminus is modified. Removed amino-acid residues are then identified by chromatography (Berg *et al.*, 2002).

Edman sequencing is automated (Niall, 1973) and uses a protein sequenator that allows sequencing peptides of about 5–50 amino acids long. Polypeptides longer than 50–70 amino acids must be cleaved into smaller peptides before sequencing. Use of multiple endopeptidases for the same protein to break up long peptides allows obtaining overlapping fragments of peptide that can be used to produce final sequence. To be sequenced the peptide is adsorbed onto a solid surface.

Edman approach is known for impurities that commonly occur in the first round of degradation. Despite this fact, it remains a valuable tool for characterizing a protein's N-terminus and is still the most reliable, and accurate protein and peptide sequencing method that provides an unambiguous sequence read and is widely used for quality control purposes.

Another method for peptide end-group analysis was developed by Fred Sanger (Ryle *et al.*, 1955) who was awarded the 1958 Nobel Prize for Chemistry for sequencing insulin that was performed with the developed approach. Sanger method uses chemical derivatives to selectively label the N-terminus of a protein with the yellow dye fluorodinitrobenzene, followed by hydrolysis, and electrophoretic or chromatographic separation of the labeled N-terminal amino acid residue. Multiple rounds of partial protein hydrolysis, fractionation, and terminal amino acid determination lead to complete sequence of a protein. There is no robust method to sequence a protein or peptide from its C-terminus because carboxypeptidases – enzymes that are used as part of these methods – can remove only specific individual C-terminal amino acids (Bergman *et al.*, 2003).

Reconstructing Proteins From Mass-Spectra

Mass spectrometry (MS) methods are now the most widely used for protein sequencing and identification. Mass spectrometry is an analytical method that measures mass-to-charge ratio for ions in gas phase. MS breaks up peptides into individual amino acids using electric current, collects the released amino acids in mass spectrometer detector to be individually identified by their unique mass. The development of MS technology allowed to sequence the entire set of proteins of a living organism and thus became the trigger for the emergence of proteomics. (Tran *et al.*, 2017).

There are currently two major types of MS approaches to protein sequencing (Fig. 17): the *bottom-up* approach for the analysis of peptide mixtures from digested proteins and *top-down* mass spectrometry that is used to analyze intact proteins.

The bottom-up approach starts from the proteolytic digestion of proteins that creates complex peptide samples, which are then used in combination with high-throughput liquid chromatography (LC) and tandem MS* (LC-MS/MS; LC-MALDI MS/MS,

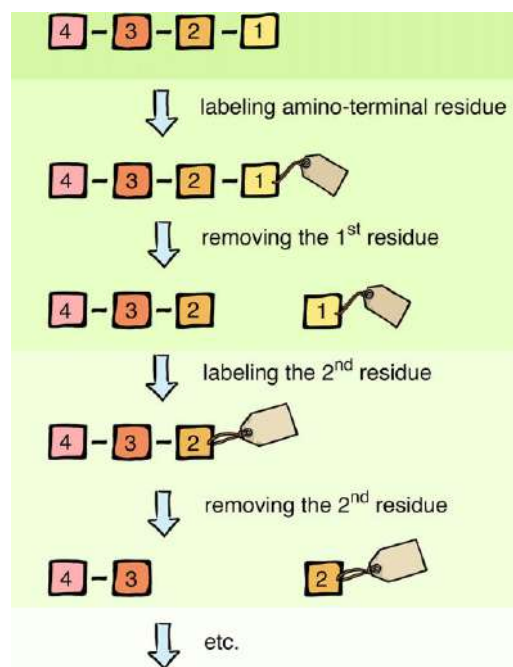


Fig. 16 Edman degradation procedure.

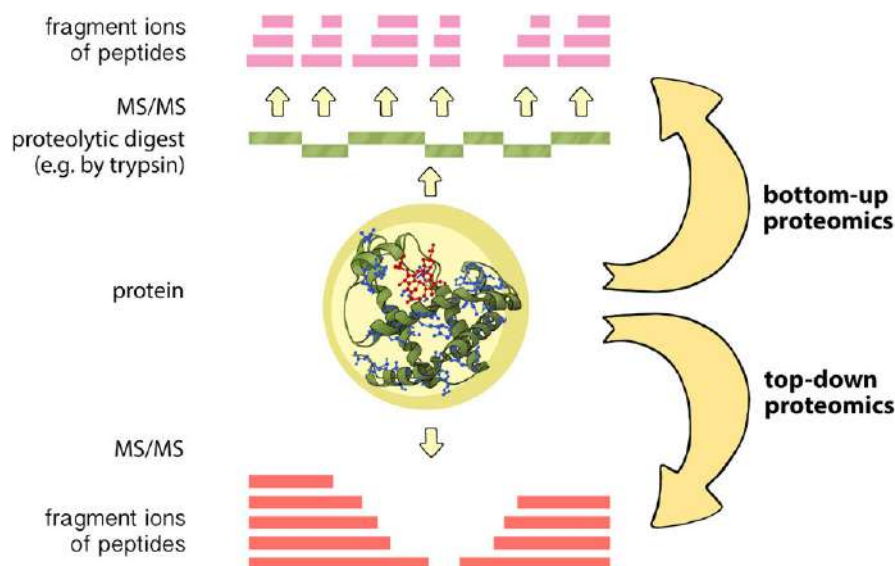


Fig. 17 MS approaches to protein sequencing.

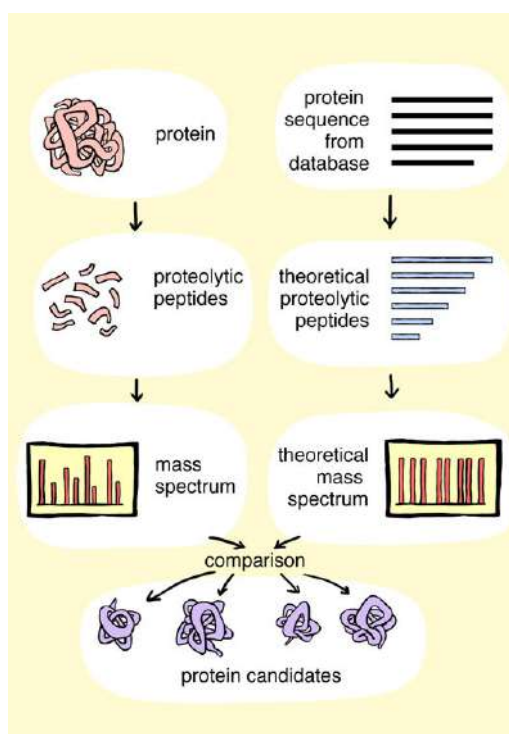


Fig. 18 Theoretical and experimental spectra for the peptide sequence.

MALDI TOF/TOF) to enable large-scale protein characterization in proteomics (Fernández-Puente *et al.*, 2014). This approach should not be applied for small proteins, as they have less cleavage sites, which will lead to an insufficient amount of peptides.

The top-down MS uses liquid chromatography or 2-D gel electrophoresis to separate intact proteins from complex samples. This approach is successfully used to characterize the structures of large proteins and to detect post-translational modifications (Aken *et al.*, 2016).

To analyze an experimental MS spectrum it needs to be compared with theoretical spectra for each of the peptides in a database (see “Relevant Websites” section). Theoretical peptides with the best fit help reconstruct the sequence of the experimental peptide (Fig. 18). Numerous post-translational modifications of proteins make them hard to be identified in databases and create a challenging computational problem.

The main goal of peptide sequencing is to reconstruct the amino acid sequence of a peptide. This goal can be achieved by using data of MS/MS spectrum and the peptide mass. Each tandem mass spectrum covers only a short part of the protein and to reconstruct the primary structure one has to assemble tandem mass spectra of overlapping peptides generated from multiple proteolytic digestions of the protein. Sounds like a DNA assembly task. Doesn't it? Here too we are trying to reconstruct a long sequence from many small fragments. In the case of protein assembly this approach is called top-down MS. To continue the analogy with genomic assembly, we should mention that top-down tandem mass spectra rarely provide full coverage of a protein. Bottom-up tandem mass spectra are used to increase sequence coverage (Liu *et al.*, 2014).

Experiments using MS technologies produce huge amount of very complex data and require specialised software for their analysis (for example: PepNovo, PEAKS, NovoHMM, MSNovo, pNovo, UniNovo, Novor and DeepNovo (Tran *et al.*, 2017)). Noisy and ambiguous MS/MS spectra cause additional computational challenges and require improved algorithms.

To find out more about the current tools that help retrieve long sequence fragments of the target proteins from sets of bottom-up and top-down MS/MS spectra see publications in further readings.

Mapping Proteins to Genome

Proteomics is a science that studies the protein composition of biological objects, as well as the modifications and the structural and functional properties of proteins. The uniqueness of the mass spectrometry pictures of products produced by proteolytic protein hydrolysis for each protein is used to analyze them against theoretical mass spectrum collected in databases.

One of the main goals when studying protein is to match the protein to the gene its coding. Knowledge of the mass of the whole protein does not provide reliable information due to the insufficient accuracy of the measurement or the presence of modifications. Therefore, researches use a mass spectrometry peptide map of a protein that consists of mass values of peptides obtained as a result of highly specific hydrolysis (chemical or enzymatic) of a protein. Such sets are called "MS-peptide fingerprints".

The main approach of protein identification by mass-spectrometry is to compare the real, experimentally obtained mass spectrum with the theoretical ones from the database that correspond to the known proteins.

Existing algorithms and corresponding programs that allow the identification of proteins by their mass spectra can be divided into three main groups:

1. Programs using the protein proteolytic peptide fingerprint (MASCOT (Perkins *et al.*, 1999). ProFound (Zhang and Chait, 2000)),
2. Programs that work with the peptide fingerprint and MS/MS spectra (MASCOT, MS-Fit (see "Relevant Websites" section),
3. Programs that work with the MS/MS spectra only (SEQUEST (see "Relevant Websites" section), PepFrag (see "Relevant Websites" section), MS-Tag (see "Relevant Websites" section), Sherpa (Taylor *et al.*, 1996).

All programs mentioned above use publicly available databases of amino acid sequences of proteins or nucleotide sequences of genes. Some of the most popular programs that contain protein sequences are: UniProt, a powerful database of annotated proteins (see "Relevant Websites" section); UniProtKB (see "Relevant Websites" section), a database that consists of Swiss-Prot (a collection of manually annotated and curated sequences) and TrEMBL (automatically annotated protein sequences translated from the nucleic acid sequences stored in EMBL) (Bairoch and Apweiler, 2000); UniRef, which contains sequence clusters for fast sequence similarity searches (see "Relevant Websites" section) and UniParc, a sequence archive of sequences and their identifiers (see "Relevant Websites" section); Protein Sequence Database (PSD) in the Protein Identification Resource (PIR, Barker *et al.*, 1999), containing annotated protein sequences; NCBI nr protein database, including entries from the non-redundant GenBank translations (see "Relevant Websites" section), UniProt, PIR (Barker *et al.*, 1999), Protein Research Foundation (PRF) in Japan (see "Relevant Websites" section), and the Protein Data Bank (PDB, see "Relevant Websites" section). Only entries with absolutely identical sequences are merged (Xu, 2004). Most of these databases also provide a sequence search tools.

Protein-Protein Alignment

Pairwise sequence alignment is a fundamental component of many bioinformatics problems. It is extremely useful in structural, functional, and evolutionary analyses of sequences since it provides information about the level of similarity between two sequences.

The overall goal of pairwise sequence alignment is to find the best "pairing" of two sequences of DNA, RNA, or protein, maximizing the number of similar characters. To achieve this, one sequence needs to be spread relative to the other to find the positions where maximal correspondence can be found. There are two main alignment strategies that are most often used: global alignment and local alignment.

In *global alignment*, the two sequences to be aligned are assumed to be generally similar over the entirety of their length. Alignment is carried out from the start and till the end of both sequences to find the best possible alignment across the entire length between the two sequences. This method is more applicable for aligning two closely related sequences of (roughly) the same length. Note that this method is not suitable for identifying the highly similar local regions between the two sequences.

Local alignment, on the other hand, does not assume that the two sequences in question are similar over their entire length. In fact, it only determines regions with the highest level of similarity between the two sequences and aligns only these regions without taking into account the alignment of the rest of the sequences. This approach can be used for aligning more divergent sequences with the goal of searching for conserved patterns in DNA or protein sequences.

The way the alignment algorithms (both global and local) are implemented is fundamentally similar, with the exception of the optimization strategy used in aligning similar residues. Typically the alignment is performed by means of *dynamic programming*.

Dynamic programming is a method that determines the optimal alignment by matching two sequences against each other to identify for all possible pairs of matching characters between them. During this process a two-dimensional matrix is created and the two sequences to be compared are both laid out of the two axes. The symbols are then matched in according to the particular *scoring matrix*. Alignment scores are calculated one row at a time. The process starts with the first row of one sequence, which is used to scan through the entire length of the other sequence. The matching scores are calculated. The scanning of the second row takes into account the scores previously obtained during the first round. The best score is put into the bottom right corner of an intermediate matrix. This process is iterated until values for all the cells are filled. The scores are thus accumulated along a diagonal going from the upper left corner to the lower right corner. Once the matrix with the scores has been computed, the next step is to find the path that represents the optimal alignment. This is done by tracing back through the matrix in reverse order from the lower right-hand corner toward the origin of the matrix in the upper left-hand corner. The best matching path is the one that has the maximum total score. If two or more paths reach the same highest score, one is chosen arbitrarily to represent the best alignment. The path can also move horizontally or vertically at a certain point, which corresponds to the introduction of a gap or an insertion or deletion for one of the two sequences.

Gap penalties

Alignment between sequences often involves adding *gaps* that represent insertions and deletions. Note that special care should be taken when allowing gaps within the scoring scheme: if the gap penalty values are set too low, the gaps can become too numerous to allow even non-related sequences to be matched up with high similarity scores. If the penalty values are set too high, gaps may become too difficult to appear, which would stand in the way of the creation of a reasonable alignment.

Another factor to consider is the cost difference between opening a gap and extending an existing gap. Normally a gap opening should have a much higher penalty than a gap extension. This is based on the rationale that if insertions and deletions ever occur, several adjacent residues are likely to have been inserted or deleted together. These differential gap penalties are also referred to as *affine gap penalties*. Under normal circumstances separate gap penalty values are assigned for introducing and extending gaps. For example, one may use a $-12/-1$ scheme, in which the gap opening penalty is equal to -12 and the gap extension penalty is given the value of -1 . The total gap penalty W is a linear function of gap length, which is calculated as $W = \gamma + \delta \times (l-1)$, where γ is the gap opening penalty, δ is the gap extension penalty, and l is the length of the gap (Gotoh, 1982). Besides the affine gap penalty, a constant gap penalty is sometimes also used, which assigns the same score for each gap position regardless whether it is an opening or an extension. In addition to that, end gaps can be allowed to be free (their inclusion will have zero penalty) to avoid getting unrealistic alignments.

Dynamic programming for global alignment

The classical global pairwise alignment algorithm using dynamic programming is known as the Needleman – Wunsch algorithm (Needleman and Wunsch, 1970). The aim this algorithm is to identify the optimal alignment is obtained over the entire lengths of the two sequences. It must extend from the beginning to the end of both sequences to achieve the highest total score. In other words, the alignment path has to go from the bottom right corner of the matrix to the top left corner. The possible drawback of focusing on trying to get a maximal score for the full-length sequence alignment is the risk of missing the best local similarities, which means that this strategy is only suitable for aligning two closely related sequences that are of the same length.

Dynamic programming for local alignment

In a regular sequence alignment, the level of divergence between the two sequences is not known in advance. The lengths of the two sequences may also differ. In cases such as these, it is of more value to identify the regional similarities between the two sequences than to try finding a match that would include all of the symbols. The dynamic programming approach to local alignment is known as Smith – Waterman algorithm (Smith and Waterman, 1981). In this algorithm, positive scores are assigned to matching symbols and zeros to mismatches. No negative scores are used. Occasionally, several optimally aligned segments with best scores are obtained. As in the global alignment, the final result is influenced by the choice of the scoring systems used. The goal of a local alignment is to get the highest alignment score locally, which may be at the expense of the highest possible overall score for a full-length alignment. This approach may be suitable for aligning divergent sequences or sequences with multiple domains that may be of different origins.

Scoring matrices

In the dynamic programming approach the alignment procedure has to make use of a scoring system, which represents a set of values used for the purpose of quantifying the likelihood of one symbol being substituted by another in an alignment. The scoring system is called a *substitution matrix* and is derived from the statistical analysis of symbol substitution data from sets of reliable alignments of highly related sequences. Scoring matrices for nucleotide sequences are relatively simple: a positive value or high score is given for a match and a negative value or low score for a mismatch. This assignment is based on the assumption that the frequencies of mutation are equal for all bases. However, this assumption may not be realistic; typically transitions (substitutions between purines and purines or between pyrimidines and pyrimidines) occur more frequently than transversions (substitutions

between purines and pyrimidines). Therefore, a more sophisticated statistical model with different probability values to reflect the two types of mutations is needed.

Scoring matrices for amino acids are much more complicated because the scoring systems have to reflect both the properties of amino acid residues, as well as the likelihood of certain residues being substituted among the homologous sequences. Certain amino acids with similar chemical properties can be more easily substituted than those having vastly different characteristics. Substitutions among similar residues are likely to preserve the essential functional and structural features. Substitutions between residues of different chemical properties, however, are more likely to cause changes (and even disruptions) to the structure and function. Such type of substitution is less likely to be favored by evolution since it is more likely to yield nonfunctional proteins.

For example, phenylalanine, tyrosine, and tryptophan all share aromatic ring structures. Because of their chemical similarities, they can be easily substituted for one another without changing the regular function and structure of the protein. Similarly, arginine, lysine, and histidine are all large basic residues and there is a high probability of them being freely interchanged. The hydrophobic residue group includes methionine, isoleucine, leucine, and valine. Small and polar residues include serine, threonine, and cysteine. Residues within these groups have high a likelihoods of being substituted for each other. However, cysteine contains a sulfhydryl group that plays a role in metal binding, as well as active site, and disulfide bond formation. Substitution of cysteine with other residues therefore often reduces the enzymatic activity or destabilizes the protein structure. This residue is thus very rarely substituted. Small and nonpolar residues such as glycine and proline are also unique since their presence often disrupts regular protein secondary structures and therefore the substitutions with these residues do not frequently occur.

Amino acid scoring matrices

Amino acid substitution matrices are 20×20 and have been developed to reflect the likelihood of residue substitutions. There are two possible types of amino acid substitution matrices: one type is based on the interchangeability of the genetic code or amino acid properties, and the other is derived from empirical studies of amino acid substitutions. Although the two different approaches have a certain degree of overlap, surprisingly, the first approach, based on the genetic code or the chemical features of amino acids, has been shown to be less accurate than the second approach, which is based on the comparison of actual amino acid substitutions among related proteins.

The empirical amino acid scoring matrices, which include PAM (Dayhoff *et al.*, 1978) and BLOSUM (Henikoff and Henikoff, 1992) matrices, are derived from actual alignments of highly similar sequences. By analyzing the frequencies of amino acid substitutions in these alignments, a scoring system can be developed by giving a higher score to more likely substitutions and correspondingly lower scores for rare ones.

Statistical significance of a sequence alignment

Once a sequence alignment showing some degree of similarity is achieved, it is important to double check whether the observed sequence alignment is indeed sound or could have been occurred by pure chance. A truly statistically significant sequence alignment might be able to provide some results connected with the homology between the sequences being aligned.

To solve this problem one must make use of a procedure designed to assist with the estimation of the probabilistic distribution of the alignment scores of the two unrelated sequences of the same length. By calculating alignment scores of a large number of unrelated pairs of sequences, a distribution model of the randomized sequence scores can be derived. It turns out that the distribution in question is the so-called Gumbel distribution (or Generalized extreme value distribution of type I) for which a mathematical expression is available. This means that given a sequence similarity value the statistical significance can be accurately estimated (Altschul *et al.*, 1990).

Multiple sequence alignment

A natural extension of pairwise alignment is multiple sequence alignment, which aims to align multiple related sequences to achieve their optimal matching. Usually such related sequences are identified through a database similarity searching. As the process generates multiple matching sequence pairs, it is often necessary to somehow turn separate pairwise alignments into a single alignment, which arranges sequences in such a way that evolutionarily equivalent positions across all sequences are matched.

The multiple sequence alignment has the unique advantage of being able to reveal more biological information than many of the pairwise alignments. For example, it allows to identify conserved sequence patterns and motifs in the whole sequence family, which would not be easy to detect when comparing only two of the sequences. Many conserved and functionally critical amino acid residues can be identified in a protein multiple alignment. Multiple sequence alignment is also an essential prerequisite to carrying out phylogenetic analysis of sequence families and prediction of protein secondary and tertiary structures.

It is theoretically possible to use dynamic programming to align any number of sequences in the similar way as for pairwise alignment. However, it is worth keeping in mind that the amount of computing time and memory required will increase exponentially the more sequences are being compared. As a consequence, full dynamic programming usually cannot be applied for datasets of more than 10–25 sequences. In practice, heuristic approaches are most often used that provide sub-optimal, but still extremely useful solutions. There are different approaches that could be used to perform multiple sequence

alignment. For the sake of the simplicity we will describe only one, namely, the *progressive alignment* approach (Feng and Doolittle, 1996).

Progressive alignment depends on the stepwise assembly of multiple alignment and is heuristic in nature. It speeds up the alignment of multiple sequences through a multistep process. It first conducts pairwise alignments for each possible pair of sequences using the Needleman–Wunsch global alignment method and records these similarity scores from the pairwise comparisons. The scores can be, for example, a percent identity or similarity scores based on a particular substitution matrix. Both do correlate with the evolutionary distances between the sequences. The scores are then converted into evolutionary distances to generate a distance matrix for all the sequences involved. A simple phylogenetic analysis is then performed based on the distance matrix to group sequences based on pairwise distance scores. As a result, a phylogenetic tree is generated using the neighbor-joining method. The tree reflects the evolutionary proximity among all the sequences.

Despite the fact that the resulting tree will only be an approximation, it still can be used as a guide for directing realignment of the sequences. For that reason, it is often referred to as a *guide tree*. According to the guide tree, the two most closely related sequences are first re-aligned using the Needleman – Wunsch algorithm. To align additional sequences, the two already aligned sequences are converted to a consensus sequence with fixed gap positions. The consensus is then treated as a single sequence in the subsequent step. In the following step, the next closest sequence based on the guide tree is aligned with the consensus sequence using dynamic programming. More distant sequences or sequence profiles are subsequently added one at a time in accordance with their relative positions on the guide tree. After realignment with a new sequence using dynamic programming, a new consensus is derived, which is then used for the next round of alignment. The process is repeated until all the sequences are aligned.

See sections “Further readings” and “Software” for the list and details of software tools and web portals used in nucleotide and protein sequence alignment.

Metabolic Pathways Analysis

The functions of about half of the genes in a bacterium can be reliably insinuated by comparing them to reference databases. This approach allows to compile an overall approximation of the metabolic map of the organism, which describes the substances the bacterium is able to synthesize on its own, the substances it needs to absorb from its environment, what constitutes its source of energy etc. These maps can then be further refined and cleaned of any possible errors by adjusting the similarity and other criteria.

The results of these studies can be used in such fields as biotechnology, agronomy, pharmaceuticals and medicine. The analysis of the metabolic pathways of commercial strains helps optimize the release of the intended product of interest. For example, enzymes that trigger key reactions in pathogens can be used as potential targets for new types of antibiotics.

Metabolic Pathways

Series of interactions between molecules in a cell that lead to assembly of new molecules in a cell, turn genes on and off, maintain and control the flow of information, energy and biochemical compounds are called biological pathways.

The most common types of biological pathways are involved in the metabolism of a cell (metabolic pathways), gene expression regulation (genetic pathways) and signal transmission (signal transduction pathways).

Metabolism can be defined as a series of chemical transformations of matter and energy taken from the environment aimed to maintain an organism’s life. Metabolic pathways in their turn are a series of biochemical reactions that convert substrates into products.

Metabolism can be conditionally divided into two opposite streams of transformations: anabolism and catabolism.

Anabolism refers to all processes involved in the creation of new substances, cells and tissues in the body. Examples of anabolism include synthesis of proteins and hormones, creation of new cells, accumulation of fats, and creation of new muscle fibers. The sum of all processes in the body that lead to the creation of any new substances and tissues is called anabolism.

On the flip side we have catabolism, which encompasses the processes involved in splitting complex substances into simpler ones, as well as decay of old cell parts and tissues of the body. An example of catabolism is the breakdown of fats and carbohydrates to produce energy, which in turn can be used for the synthesis of the substances of use to the organism, the creation of cells and the overall renewal of the body. It follows that both anabolism and catabolism are very closely interrelated to each other. The most important metabolic pathways in humans are: glycolysis, citric acid cycle (Krebs’ cycle), oxidative phosphorylation, pentose phosphate pathway, urea cycle, fatty acid β -oxidation, gluconeogenesis.

The most convenient way to investigate metabolic pathways is by analyzing metabolic models. They contain all the information about the reactions that can occur in a cell, how the reactions are related to each other, the activity of which genes can regulate the reactions, which of the reactions relate to standard metabolic pathways, and which are the most important, how they are combined into metabolic pathways and how they interact with each other.

Regulation of metabolic pathways is achieved through complex interactions of a large number of factors. As a result of the proper functioning of all metabolic pathways, the cell can maintain a stable homeostasis and properly respond to changes in the external environment. The availability of a large volume of genomic information (assembled genomes) facilitates the study of metabolism regulation. Based on this information, whole genome metabolic models generalizing the available ideas about

metabolism and its regulation are constructed. Such models include a list of all possible reactions, a set of rules for the regulation of individual reactions, thermodynamic limitations that determine possible directions of reactions.

Methods of metabolic models analysis can be subdivided into two groups: qualitative (topological analysis, flux balance analysis, structural kinetic models) and quantitative (structural kinetic models, kinetic models) (Tomar and De, 2014). A number of methods also use information about the atomic structure of metabolites. They allow to analyze the transition from individual metabolites to biochemical reactions.

The development of bioinformatics methods to interpret data led to the development of metabolic pathway analysis methodologies, including structural and stoichiometric analysis, metabolic flux analysis, metabolic control analysis, and several kinetic modeling based analysis. The article written by Dr. Namrata Tomar and Dr. Rajat K. (Tomar and De, 2014), provides the comprehensive survey on the existing metabolic pathway analysis methodologies.

A review of the most common approaches for the construction of computational models of metabolic systems can be found in (Rice *et al.*, 2008).

Metabolic Databases

Metabolic models can be obtained from different sources.

The KEGG database contains a large amount of information on metabolic reactions. It includes several separate databases, such as: KEGG REACTION – with information on biochemical reactions; KEGG COMPOUND and KEGG GLYCAN – about metabolites; KEGG ENZYME – about enzymes; KEGG GENES – about the genes; KEGG PATHWAY and KEGG MODULES – about metabolic pathways.

This database was developed for bioinformatics analysis in genomics, metagenomics, metabolomics, modeling and simulation in systems biology, and translational research in drug development. It contains about 4000 organisms, including 123 animals. The database is curated and requires a paid license for full access. Some information is available for free via the web interface and the program interface (Kanehisa *et al.*, 2008, 2012, 2014, 2016, 2017). One can analyze new sequences (DNA or protein) against metadata stored in the KEGG database by mapping them to the known pathways, models, orthologs and so on (See “Relevant Websites” Section) by linking new genomes to well curated known pathways.

The BioCyc database is similar to the KEGG database. It contains about 7600 organism-specific bases. Some of them are manually curated, including the base containing human data. Another ones, including the mouse DB, are partially supervised.

The metabolic models are also stored in the EBI BioModels Database. This free to use DB contains about 2500 whole genome metabolic models in SBML format, but all of them are generated automatically from KEGG and MetaCyc databases.

The Reactome database is another free database with information on metabolic and molecular pathways. This carefully curated database focuses mainly on human biology. Data from the database can be downloaded in SBML and other formats.

There are several end to end data pipelines dedicated to help with the complex task of metabolic pathways analysis (Poskar *et al.*, 2014; Abubucker *et al.*, 2016).

Gene Ontology

Gene ontology (GO) is a major bioinformatics initiative to unify the representation of gene and gene product attributes across all species (see “Relevant Websites” section).

Because the complexity of biological systems and the sizes of the datasets that need to be analyzed continue to grow, biomedical research is becoming increasingly dependent on knowledge stored in computable form. The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene functions and gene products across different databases. The GO knowledgebase is composed of three primary components:

1. *Gene Ontology*, which provides the logical structure of the biological functions (‘terms’). It is structured as a directed acyclic graph where each term has defined relationships to one or more other terms in the same domain and sometimes to other domains. The ontology includes data on: *cellular components*; *molecular functions* and *biological processes* (see “Relevant Websites” section).

Molecular function corresponds to activities that can be performed by the individual gene products or by the assembled complexes of gene products.

A biological process in GO is not the same as a pathway since GO does not include information about the dynamics or the dependencies, both of which would be required to fully describe a pathway.

The GO vocabulary is species-neutral, and includes terms applicable to prokaryotes and eukaryotes, single or multicellular organisms.

2. *GO annotations* are evidence-based statements relating a specific gene product (a protein, non-coding RNA, or macromolecular complex, which we referred as ‘genes’ for simplicity) to a specific ontology term.
3. *Tools* that facilitate the creation, maintenance and use of ontologies. Software being developed aims to help edit and perform logic based analysis of ontologies, provide web access to the ontology and its annotations, and analyze data using the information in the GO knowledgebase to support biomedical research.

Ontologies can be browsed using a range of web-based browsers (see “Relevant Websites” section).

The overall goal of GO is to create a comprehensive model of biological systems. Currently, the GO knowledgebase includes experimental findings from almost 140,000 published papers (see “Relevant Websites” section), represented by over 600,000 experimentally-supported GO annotations, on the basis of which an additional 6 million functional annotations for a diverse set of organisms spanning the tree of life can be inferred.

That said, it is worth mentioning that GO does not store gene sequences and is not a catalog of gene products. GO describes how gene products behave in a cellular/organism context. The use of GO terms by the collaborating databases provides controlled vocabularies that are structured to facilitate querying at different levels and also to allow annotators to assign properties to genes or gene products at different levels, depending on the depth of knowledge about the particular entity.

Conclusions

Due to the rapid evolution of new methodologies in molecular and computational biology evolve, the aim of this article was to only provide a general overview of sequence analysis at the DNA, RNA, and protein and metabolic pathways level.

We have attempted to briefly cover both the general approaches, and the most popular and powerful tools that are currently being used to analyze large amounts of data from labs around the world.

Acknowledgement

Authors are grateful to our colleague Elena Strelnikova for her help with **Figs. 16, 17** and **18**. This work was supported by Russian Science Foundation (grant number 14–50–00069).

See also: Algorithms for Strings and Sequences: Multiple Alignment. Algorithms for Strings and Sequences: Pairwise Alignment. Natural Language Processing Approaches in Bioinformatics

References

- Abubucker, S., Segata, N., Goll, J., *et al.*, 2016. Metabolic reconstruction for metagenomic data and its application to the human microbiome. *PLOS Comput. Biol.* 8 (6), e1002358.
- Acland, A., Agarwala, R., *et al.*, 2014. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 42, D7–D17.
- Aken, B.L., Ayling, S., Barrell, D., *et al.*, 2016. The Ensembl gene annotation system. *Database J. Biol. Databases Curation* 2016, baw093.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.
- Anders, S., Pyl, P.T., Huber, W., 2015. HTSeq – A python framework to work with high-throughput sequencing data. *Bioinformatics* 31 (2), 166–169.
- Andrews S., 2010. FastQC: A quality control tool for high throughput sequence data. Available online at: <http://www.bioinformatics.babraham.ac.uk/projects/fastqc>.
- Antipov, D., Korobeynikov, A., McLean, J.S., Pevzner, P.A., 2015. hybridSPAdes: An algorithm for hybrid assembly of short and long reads. *Bioinformatics* 32 (7), 1009–1015.
- Azad, R.K., Borodovsky, M., 2004. Probabilistic methods of identifying genes in prokaryotic genomes: Connections to the HMM theory. *Brief. Bioinform.* 5, 118–130.
- Barker, W.C., Garavelli, J.S., McGarvey, P.B., *et al.*, 1999. The PIR-international protein sequence database. *Nucleic Acids Res.* 27 (1), 39–43.
- Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L., 2004. GenBank: Update. *Nucleic Acids Res.* 32, D23–D26.
- Berg, J., Tymoczko, J., Stryer, L., 2002. *Biochemistry*, 5th ed. New York: W.H. Freeman.
- Bergman, T., Cederlund, E., Jörnvall, H., Fowler, E., 2003. Current protocols in protein science. (Chapter 11, Unit 11.8).
- Bairoch, A., Apweiler, R., 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.* 28 (1), 45–48.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for illumina sequence data. *Bioinformatics* 30 (15), 2114–2120.
- Bookstein, A., Kulyukin, V.A., Raita, T., 2002. Generalized hamming distance. *Inform. Retr.* 5, 353.
- Bradnam, K.R., Fass, J.N., Alexandrov, A., *et al.*, 2013. Assemblathon 2: Evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience* 2 (1), 10.
- Bray, N.L., Pimentel, H., Melsted, P., Pachter, L., 2016. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* 34 (5), 525–527.
- Bresler, M.A., Sheehan, S., Chan, A.H., Song, Y.S., 2012. Telescope: De novo assembly of highly repetitive regions. *Bioinformatics* 28 (18), i311–i317.
- Brown, J.W.S., Echeverria, M., Qu, L.-H., *et al.*, 2003. Plant snoRNA database. *Nucleic Acids Res.* 31, 432–435.
- Burge, S.W., Daub, J., Eberhardt, R., *et al.*, 2013. Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.* 41, D226–D232.
- Bushnell, B., 2014. BBTools: A suite of fast, multithreaded bioinformatics tools designed for analysis of DNA and RNA sequencedata. Available online at: <https://jgi.doe.gov/data-and-tools/bbtools/>.
- Bushmanova, E., Antipov, D., Lapidus, A., Suvorov, V., Prijbelski, A.D., 2016. rnaQUAST: A quality assessment tool for de novo transcriptome assemblies. *Bioinformatics* 32 (14), 2210–2212.
- Cingolani, P., Platts, A., Wang, L.L., *et al.*, 2012. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly* 6 (2), 80–92.
- Danecek, P., Auton, A., Abecasis, G., *et al.*, 2011. The variant call format and VCFtools. *Bioinformatics* 27 (15), 2156–2158.
- Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model for evolutionary change in proteins. In: Dayhoff, Margaret O. (Ed.), *Atlas of Protein Sequence and Structure*, vol. 5. Washington DC: National Biochemical Research Foundation, pp. 345–352.
- Delcher, A.L., Harmon, D., Kasif, S., White, O., Salzberg, S.L., 1999. Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.* 27 (23), 4636–4641.
- Dobin, A., Davis, C.A., Schlesinger, F., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29 (1), 15–21.
- Dündar, F., Skrabanek, L., Zumbo, P., 2015. Applied Bioinformatics Core/Weill Cornell Medical College. Applied Bioinformatics Core/Weill Cornell Medical College. pp. 1–67.
- Earl, D., Bradnam, K., John, J.S., *et al.*, 2011. Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome Res.* 21 (12), 2224–2241.
- Edman, P., Högfeldt, E., Sillén, L.G., Kinell, P.-O., 1950. Method for determination of the amino acid sequence in peptides. *Acta Chem. Scand.* 4, 283–293.

- Felsenstein, J., 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.* 17 (6), 368–376.
- Feng, D.-F., Doolittle, R.F., 1996. Doolittle progressive alignment of amino acid sequences and construction of phylogenetic trees from them. In: *Proceedings of the Methods in Enzymology*, 266, pp. 368–382. Academic Press.
- Fernández-Puente, P., Mateos, J., Blanco, F.J., Ruiz-Romero, C., 2014. LC-MALDI-TOF/TOF for shotgun proteomics. *Methods Mol. Biol.* 2014 (1156), 27–38.
- Ferragina, P., Manzini, G., Mäkinen, V., Navarro, G., 2004. An alphabet-friendly FM-index. In: *Proceedings of the String Processing and Information Retrieval*, p. 228. Berlin/Heidelberg: Springer.
- Gotoh, O., 1982. An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 705–708.
- Grabherr, M.G., Haas, B.J., Yassour, M., *et al.*, 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biotechnol.* 29 (7), 644–652.
- Gurevich, A., Saveliev, V., Vyahhi, N., Tesler, G., 2013. QUAST: Quality assessment tool for genome assemblies. *Bioinformatics* 29 (8), 1072–1075.
- Hannenhalli, S., Pevzner, P.A., 1999. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *J. ACM (JACM)* 46 (1), 1–27.
- Heather, J.M., Chain, B., 2016. The sequence of sequencers: The history of sequencing DNA. *Genomics* 107 (1), 1–8. doi:10.1016/j.ygeno.2015.11.003.
- Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. *PNAS* 89, 10915–10919.
- Hrdlickova, R., Toloue, M., Tian, B., 2017. RNA-Seq methods for transcriptome analysis. *WIREs RNA* 8 (1), doi:10.1002/wrna.1364. Jan.
- Hunt, M., Kikuchi, T., Sanders, M., *et al.*, 2013. REAPR: A universal tool for genome assembly evaluation. *Genome Biol.* 14 (5), R47.
- Kanehisa, M., Araki, M., Goto, S., *et al.*, 2008. KEGG for linking genomes to life and the environment. *Nucleic Acids Res.* 36, D480–D484.
- Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., Tanabe, M., 2012. KEGG for integration and interpretation of large-scale molecular datasets. *Nucleic Acids Res.* 40, D109–D114.
- Kanehisa, M., Goto, S., Sato, Y., *et al.*, 2014. Data, information, knowledge and principle: Back to metabolism in KEGG. *Nucleic Acids Res.* 42, D199–D205.
- Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M., 2016. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.* 44, D457–D462.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K., 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* 45, D353–D361.
- Kapustin, Y., Souvorov, A., Tatusova, T., Lipman, D., 2008. Splein: Algorithms for computing spliced alignments with identification of paralogs. *Biol. Direct* 3 (1), 20.
- Kent, W.J., 2002. BLAT – The BLAST-like alignment tool. *Genome Res.* 12 (4), 656–664.
- Kent, W.J., Sugnet, C.W., Furey, T.S., *et al.*, 2002. The human genome browser at UCSC. *Genome Res.* 12 (6), 996–1006.
- Kim, D., Salzberg, S.L., 2011. TopHat-Fusion: An algorithm for discovery of novel fusion transcripts. *Genome Biol.* 12 (8), R72.
- Kim, D., Langmead, B., Salzberg, S.L., 2015. HISAT: A fast spliced aligner with low memory requirements. *Nat. Methods* 12 (4), 357–360.
- Koren, S., Schatz, M.C., Walenz, B.P., *et al.*, 2012. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol.* 30 (7), 693–700.
- Koren, S., Walenz, B.P., Berlin, K., *et al.*, 2017. Canu: Scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* 27 (5), 722–736.
- Kuruba, K.R., Montgomery, S.B., 2015. RNA Sequencing and Analysis. *Cold Spring Harb Protoc.* 11, 951–969.
- Kumar, S., Tamura, K., Nei, M., 1994. MEGA: Molecular evolutionary genetics analysis software for microcomputers. *Bioinformatics* 10 (2), 189–191.
- Lagesen, K., Hallin, P., Rødland, E.A., *et al.*, 2007. RNAmmer: Consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Res.* 35, 3100–3108.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* 9 (4), 357–359.
- Lapidus, A.L., 2015. *Genome Sequence Databases: Sequencing and Assembly*. Reference Module in Biomedical Sciences. Oxford: Elsevier, Available at: <http://www.sciencedirect.com/science/article/pii/B9780128012383024958>.
- Leung, H.C., Yiu, S.M., Chin, F.Y., 2015. IDBA-MTP: A hybrid metatranscriptomic assembler based on protein information. *J. Comput. Biol.* 22 (5), 367–376.
- Levene, M.J., Korlach, J., Turner, S.W., *et al.*, 2003. Zero-mode waveguides for single-molecule analysis at high concentrations. *Science* 299, 682–686.
- Li, B., Fillmore, N., Bai, Y., *et al.*, 2014. Evaluation of de novo transcriptome assemblies from RNA-Seq data. *Genome Biol.* 15 (12), 553.
- Li, D., Liu, C.M., Luo, R., Sadakane, K., Lam, T.W., 2015. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 31 (10), 1674–1676.
- Li, H., 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Available from: <http://arxiv.org/abs/1303.3997>.
- Li, H., 2016. Minimap and miniasm: Fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* 32 (14), 2103–2110.
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows – Wheeler transform. *Bioinformatics* 25 (14), 1754–1760.
- Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009. The sequence alignment/map format and SAMtools. *Bioinformatics* 25 (16), 2078–2079.
- Liu, X., Dekker, L.J., Wu, S., *et al.*, 2014. De novo protein sequencing by combining top-down and bottom-up tandem mass spectra. *J. Proteome Res.* 13 (7), 3241–3248.
- Lizardi, P.M., 2000. Multiple displacement amplification. Yale University, U.S. Patent 6,124,120.
- Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 15 (12), 550.
- Lowe, T.M., Eddy, S.R., 1997. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.* 25, 955–964.
- Lukashin, A.V., Borodovsky, M., 1998. GeneMark.hmm: New solutions for gene finding. *Nucleic Acids Res.* 26 (4), 1107–1115.
- Luo, R., Liu, B., Xie, Y., *et al.*, 2012. SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. *Gigascience* 1 (1), 18.
- Magoc, T., Pabinger, S., Canzar, S., *et al.*, 2013. GAGE-B: An evaluation of genome assemblers for bacterial organisms. *Bioinformatics* 29 (14), 1718–1725.
- Margulies, M., Egholm, M., Altman, W.E., *et al.*, 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437, 376–380.
- Martin, M., 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet J.* 17 (1), 10.
- McKenna, A., Hanna, M., Banks, E., *et al.*, 2010. The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* 20 (9), 1297–1303.
- Milne, I., Bayer, M., Cardle, L., *et al.*, 2009. Tablet – Next generation sequence assembly visualization. *Bioinformatics* 26 (3), 401–402.
- Morgulis, A., Coulouris, G., Raytselis, Y., *et al.*, 2008. Database indexing for production MegaBLAST searches. *Bioinformatics* 24 (16), 1757–1764.
- Myers, E., 2005. The fragment assembly string graph. *Bioinformatics* 21 (S2), ii79–ii85.
- Myers, E.W., Sutton, G.G., Delcher, A.L., *et al.*, 2000. A whole-genome assembly of *Drosophila*. *Science* 287 (5461), 2196–2204.
- Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48 (3), 443–453.
- Niall, H.D., 1973. Automated Edman degradation: The protein sequenator. *Methods Enzymol.* 27, 942–1010.
- Nurk, S., Bankevich, A., Antipov, D., *et al.*, 2013. Assembling single-cell genomes and mini-metagenomes from chimeric MDA products. *J. Comput. Biol.* 20 (10), 714–737.
- Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A., 2017. metaSPAdes: A new versatile metagenomic assembler. *Genome Res.* 27 (5), 824–834.
- O’Connell, J., Schulz-Trieglaff, O., Carlson, E., *et al.*, 2015. NxTrim: Optimized trimming of illumina mate pair reads. *Bioinformatics* 31 (12), 2035–2037.
- Okonechnikov, K., Conesa, A., García-Alcalde, F., 2015. Qualimap 2: Advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics* 32 (2), 292–294.
- Peng, Y., Leung, H.C., Yiu, S.M., Chin, F.Y., 2012. IDBA-UD: A de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* 28 (11), 1420–1428.
- Perkins, D.N., Pappin, D.J.C., Creasy, D.M., Cottrell, J.S., 1999. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 20, 3551–3567.

- Pevzner, P.A., Tang, H., Waterman, M.S., 2001. An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci.* 98 (17), 9748–9753.
- Poskar, C.H., Huege, J., Krach, C., Shachar-Hill, Y., Junker, B.H., 2014. High-throughput data pipelines for metabolic flux analysis in plants. *Methods Mol. Biol.* 1090, 223–246.
- Prijbelski, A.D., Vasilinets, I., Bankevich, A., *et al.*, 2014. ExSPAnder: A universal repeat resolver for DNA fragment assembly. *Bioinformatics* 30 (12), i293–i301.
- Rice, S.A., Steuer, R., Junker, B.H., 2008. Computational models of Metabolism: Stability and regulation in metabolic. *Networks*. doi:10.1002/9780470475935.ch3.
- Robertson, G., Schein, J., Chiu, R., *et al.*, 2010. De novo assembly and analysis of RNA-seq data. *Nat. Methods* 7 (11), 909–912.
- Robinson, M.D., McCarthy, D.J., Smyth, G.K., 2010. edgeR: A bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26 (1), 139–140.
- Rothberg, J.M., Hinz, W., Rearick, T.M., *et al.*, 2011. An integrated semiconductor device enabling non-optical genome sequencing. *Nature* 475, 348–352.
- Ryle, A.P., Sanger, F., Smith, L.F., Kitai, R., 1955. The disulphide bonds of insulin. *Biochem. J.* 60 (4), 541–556.
- Salzberg, S., Delcher, A., Kasif, S., White, O., 1998. Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.* 26 (2), 544–548.
- Salzberg, S.L., Phillippy, A.M., Zimin, A., *et al.*, 2012. GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res.* 22 (3), 557–567.
- Shendure, J., Porreca, G.J., Reppas, N.B., *et al.*, 2005. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* 309, 1728–1732.
- Simão, F.A., Waterhouse, R.M., Ioannidis, P., Kriventseva, E.V., Zdobnov, E.M., 2015. BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics* 31 (19), 3210–3212.
- Simpson, J., Durbin, R., 2012. Efficient de novo assembly of large genomes using compressed data structures. *Genome Res.* 22, 549–556.
- Simpson, J.T., Wong, K., Jackman, S.D., *et al.*, 2009. ABySS: A parallel assembler for short read sequence data. *Genome Res.* 19 (6), 1117–1123.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147 (1), 195–197.
- Smith-Unna, R., Boursnell, C., Patro, R., Hibberd, J.M., Kelly, S., 2016. TransRate: Reference-free quality assessment of de novo transcriptome assemblies. *Genome Res.* 26 (8), 1134–1144.
- Sović, I., Šikić, M., Wilm, A., *et al.*, 2016. Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nat. Commun.* 7, 11307.
- Tang, S., Lomsadze, A., Borodovsky, M., 2015. Identification of protein coding regions in RNA transcripts. *Nucleic Acids Res.* 43 (12), e78.
- Taylor, J.A., Walsh, K.A., Johnson, R.S., 1996. Sherpa: A macintosh-based expert system for the interpretation of electrospray ionization LC/MS and MS/MS data from protein digests. *Rapid Commun. Mass Spectrom.* 10, 679–687.
- Thorvaldsdóttir, H., Robinson, J.T., Mesirov, J.P., 2013. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Brief. Bioinform.* 14 (2), 178–192.
- Tomar, N., De, R.K., 2014. A comprehensive view on metabolic pathway analysis methodologies. *Curr. Bioinform.* 9, 295–305.
- Tran, N.H., Zhang, X., Xin, L., Shan, B., Li, V., 2017. De novo peptide sequencing by deep learning. *Proc. Natl. Acad. Sci.* 114 (31), 8247–8252.
- Trapnell, C., Pachter, L., Salzberg, S.L., 2009. TopHat: Discovering splice junctions with RNA-Seq. *Bioinformatics* 25 (9), 1105–1111.
- Valouev, A., Ichikawa, J., Tonthat, J., *et al.*, 2008. A high-resolution, nucleosome position map of *C. elegans* reveals a lack of universal sequence-dictated positioning. *Genome Res.* 18, 1051–1063.
- Vasilinets, I., Prijbelski, A.D., Gurevich, A., Korobeynikov, A., Pevzner, P.A., 2015. Assembling short reads from jumping libraries with large insert sizes. *Bioinformatics* 31 (20), 3262–3268.
- Wang, B.-B., Brendel, V., 2004. The ASRG database: Identification and survey of *Arabidopsis thaliana* genes involved in pre-mRNA splicing. *Genome Biol.* 5, R102.
- Wang, L., Wang, S., Li, W., 2012. RSeQC: Quality control of RNA-seq experiments. *Bioinformatics* 28 (16), 2184–2185.
- Wang, Z., Chen, Y., Li, Y., 2004. A brief review of computational gene prediction methods. *Genom. Prot. Bioinform.* 4, 216–221.
- Wick, R.R., Judd, L.M., Gorrie, C.L., Holt, K.E., 2017. Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLOS Comput. Biol.* 13 (6), e1005595.
- Woyke, T., Tighe, D., Mavromatis, K., *et al.*, 2010. One Bacterial Cell, One Complete Genome. *PLoS ONE* 5 (4), e10314.
- Wu, T.D., Watanabe, C.K., 2005. GMAP: A genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* 21 (9), 1859–1875.
- Xie, Y., Wu, G., Tang, J., *et al.*, 2014. SOAPdenovo-Trans: De novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics* 30 (12), 1660–1666.
- Xu, D., 2004. Protein Databases on the Internet. *Curr. Protoc. Mol. Biol.* 19.4, (CHAPTER: Unit–19.4. *PMC*. Web. 2 Nov. 2017).
- Ye, Y., Tang, H., 2015. Utilizing de Bruijn graph of metagenome assembly for metatranscriptome analysis. *Bioinformatics* 32 (7), 1001–1008.
- Zhang, W., Chait, B.T., 2000. ProFound: An expert system for protein identification using mass spectrometric peptide mapping information. *Analyt. Chem.* 72, 2482–2489.
- Zerbino, D.R., Birney, E., 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 18 (5), 821–829.

Further Reading

- Brudno, M., Malde, S., Poliakov, A., *et al.*, 2003. Glocal alignment: Finding rearrangements during alignment. *Bioinformatics* 19 (Suppl. 1), i54–i62. 90001.
- Dohrmann, J., Puchin, J., Singh, R., 2015. Global multiple protein-protein interaction network alignment by combining pairwise network alignments. *BMC Bioinform.* 16 (Suppl. 13), S11.
- Dündar, F., Skrabanek, L., Zumbo, P., 2015. Introduction to Differential Gene Expression Analysis Using RNA-Seq. *Applied Bioinformatics Core/Weill Cornell Medical College*.
- Faisal, F.E., Zhao, H., Milenkovic, T., 2015. Global Network Alignment in the Context of Aging. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 12 (1), 40–52.
- Jones, N.C., Pevzner, P., 2004. An introduction to bioinformatics algorithms. MIT press.
- Peris, G., Marzal, A., 2014. Statistical significance of normalized global alignment. *J. Comput. Biol.* 21 (3), 257–268.
- Vyatkina, K., 2017. De novo sequencing of top-down tandem mass spectra: A next step towards retrieving a complete protein sequence. *Proteomes* 5 (1), 6.

Software

Alignment

- MUMmer <http://mummer.sourceforge.net/>
- Minimap2 <https://github.com/lh3/minimap2>
- NCBI BLAST <https://blast.ncbi.nlm.nih.gov/>
- Bushnell, B., 2014. BBMap: a fast, accurate, splice-aware aligner. <https://sourceforge.net/projects/bbmap/>
- List of sequence alignment software https://en.wikipedia.org/wiki/List_of_sequence_alignment_software

Pairwise alignment

- CLUSTALW/CLUSTALW2/CLUSTALO <https://www.ebi.ac.uk/Tools/msa/clustalo/>
- T-COFFEE <http://www.tcoffee.org/>
- MAFFT <https://www.ebi.ac.uk/Tools/msa/mafft/>
- MUSCLE <https://www.ebi.ac.uk/Tools/msa/muscle/>

Reads QC

- a. Bushnell, B., 2014. BBTools software package <https://sourceforge.net/projects/bbtools/>
- b. Andrews, S., 2010. FastQC: a quality control tool for high throughput sequence data. <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.

Functional annotation systems

- a. BioConductor - <https://www.bioconductor.org/>
- b. DAVID - <https://david.ncifcrf.gov/summary.jsp>
- c. Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., Lopez, R. 2005. InterProScan: protein domains identifier. Nucleic Acids Res. 33(Web Server issue):W116–20. <https://github.com/ebi-pf-team/interproscan>
- d. Ensembl: <http://uswest.ensembl.org/index.html>

Relevant Websites

<https://biocyc.org/>
BioCyc database.

<https://www.ebi.ac.uk>
EBI BioModels Database.

https://en.wikipedia.org/wiki/FASTA_format
FASTA format.

https://en.wikipedia.org/wiki/FASTQ_format
FASTQ format.

<http://www.geneontology.org/page/introduction-go-resource>
Gene Ontology Consortium.

<ftp://ftp.geneontology.org>
Gene Ontology Consortium.

<http://www.genome.jp/kegg/ko.html>
KEGG.

<www.genome.jp>
KEGG database.

<http://www.ncbi.nlm.nih.gov>
NCBI - NIH.

<https://www.nanoporetech.com/>
Oxford Nanopore Technologies.

<https://omictools.com/pepfrag-tool>
PepFrag.

https://en.wikibooks.org/wiki/Proteomics/Protein_Identification_-_Mass_Spectrometry/Databases
Proteomics/Protein Identification.

<http://prospector.ucsf.edu/>
ProteinProspector.

<http://prospector.ucsf.edu/>
ProteinProspector.

<http://www.rcsb.org/pdb>
RCSB PDB.

<sbml.org>
SBML format.

<https://omictools.com/sequest-tool>
SEQUEST.

<http://www.illumina.com>
Solexa technology.

<https://www.proteinresearch.net/>
The Protein Research Foundation.

<http://www.uniprot.org/>
UniProt.

<http://www.uniprot.org/uniprot/>
UniProtKB.

<http://www.uniprot.org/>
UniProt.

<http://www.uniprot.org/uniprot/>
UniProt.

<https://www.uniprot.org/help/uniref>
UniProt.

<https://www.uniprot.org/help/uniparc>
UniProt.

Biographical Sketch



Andrey Prijibelski (formal transliteration Przhevalsky) received BSc degree in informatics from St. Petersburg State Technical University in 2010 and MSc in applied mathematics and physics from St. Petersburg Academic University in 2012. In 2011 he started an internship at Algorithmic Biology Lab led by Prof. Pavel Pevzner in St. Petersburg Academic University and then continued to work there as a junior research fellow. In 2014 the whole laboratory moved to St. Petersburg University and became Center for Algorithmic Biotechnologies (CAB). Andrey participated in development of one of the main lab projects – SPAdes de novo genome assembler and in 2014 received outstanding student paper award at ISMB 2014, Boston. Later he participated developing SPAdes-based de novo transcriptome assembler and improving SPAdes for large genome assembly. Andrey also teaches “NGS data analysis” for master students St. Petersburg Academic University and participates in various bioinformatics workshops and schools as a teacher and co-organizer.



Anton Korobeynikov received the MSc and PhD degrees in applied mathematics from St. Petersburg State University, Russia, in 2007 and 2010, respectively. He started to work at St. Petersburg State University in 2007, where he currently holds the position of Associate Professor of Statistical Modeling Department, School of Mathematics and Mechanics. The main areas of research interests of Dr. Korobeynikov are computational and applied statistics, including time series analysis, efficient implementation of algorithms of computation statistics and probabilistic methods in bioinformatics.

In 2012 he joined Algorithmic Biology Laboratory at St. Petersburg Academic University. The laboratory led by Prof. P. Pevzner is developing SPAdes – a novel de novo assembler suitable for single cell and multi cell data among other tools. In 2014 the whole laboratory moved to St. Petersburg University and became Center for Algorithmic Biotechnologies. Currently Anton is a SPAdes team leader and oversees the technical questions related to SPAdes development.



Professor Alla Lapidus holds a PhD in Molecular Biology from the Institute for Genetics and Selection of Industrial Microorganisms (IGSIM) and an MS Degree in Physics from the Moscow Engineering Physics Institute. Her primary areas of expertise include high-throughput DNA sequencing, genome structure analysis, reference quality genome assembly, bioinformatics. Currently the deputy-director of the Center for Algorithmic Biotechnology, St. Petersburg State University, Dr. Lapidus was previously an Associate professor at Fox Chase Cancer Center. From 2003 Dr. Lapidus has been applying high-throughput DNA sequencing, genome-wide analysis and bioinformatics to the study of microbial and fungal genomes and metagenomic communities by optimizing data QC, data analysis and de-novo assemblies at Joint Genome Institute (JGI). She has spearheaded multiple projects using NGS to optimize data QC, analysis and de-novo assemblies, created microbial finishing pipeline from the ground up and was the mind behind the development of the next generation de-novo assembly algorithms.

Previously she served as the Director of Sequencing Center, Integrated Genomics, as a Senior Research Scientist at the INRA, France, at The University of Chicago and at the IGSIM, Moscow. Professor Lapidus has authored several on-line courses on Bioinformatics and is a lead of the MS program “Bioinformatics” at SPbU. She also authored and co-authored over 350 papers and patents.

Sequence Composition

Jin Xing Lim and Bryan T Li, Temasek Polytechnic, Singapore

Maurice HT Ling, Colossus Technologies, LLP, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

Since the sequencing of *Escherichia coli* (Moxon and Higgins, 1997) and *Plasmodium falciparum* (Cowman and Crabb, 2002) genome, DNA sequence has been commonly known as the “blueprint of life” as DNA is the hereditary molecule for cellular life. Hence, logically should contain all the information necessary to infer the activities of life even though such inference may not be obvious. Despite so, a lot of information about the regulation of a gene, as well as the function of the gene product may be derived from its sequence. For example, common response elements across multiple genes at the promoter region suggest induction by the same external stimulus (Domenech *et al.*, 2001). In this article, we will look at various DNA sequence features at the DNA and review several recent applications where sequence features are used. Through these applications, we can appreciate that sequence composition is an integral aspect of sequence analysis.

DNA Sequence Composition

In this section, we will look at 6 major DNA sequence composition features: promoter, enhancer, ribosome binding site, coding sequence, terminator, methylation sites and CpG islands.

Promoter is a short DNA region of 100–1000 base-pairs upstream of the coding sequence that triggers gene transcription. Hence, identifying the promoter indicates the start of transcriptional activity leading to the start of a potential mRNA molecule. A desktop tool capable of finding promoter is ProSOM (Abeel *et al.*, 2008). The core promoter has exclusive features that cannot be found in other sequences when comparing with the average structural profile based on based stacking energy of transcribed. The program that is used for ProSOM is able to distinctly differentiate between the structural profiles of promoter sequences and other genomic sequences. It is done based on unsupervised clustering of physical properties of DNA by using self-organising maps. Additional tests on the ENCODE regions of the human genome yield high precision, showing 98% of all predictions made by ProSOM can be associated with transcriptionally active regions (Abeel *et al.*, 2008). Another tool that we can use is CoreBoost_HM (Wang *et al.*, 2009). It is a web-based tool that is an upgraded version of CoreBoost (Zhao *et al.*, 2007), which predicts core-promoters by applying a boosting technique with stumps, using both the small-scale and large-scale DNA sequence features. CoreBoost_HM uses the same technique as CoreBoost but with the addition of histone modification features as well. CoreBoost_HM can be used to identify and characterise the core-promoters of both coding and noncoding genes with high sensitivity and specificity at high resolution (Wang *et al.*, 2009). ChromHMM (Ernst and Kellis, 2012) characterises chromatin states, it can facilitate the biological characterization of each state by automatically computing state enrichments for large-scale functional and annotation datasets (Ernst and Kellis, 2012). Hence, ChromHMM is also able to locate promoter.

Enhancer is a short DNA region of 50–1500 base-pairs that increases the chance of occurrence of a particular gene’s transcription through binding by a type of proteins called activators. One tool that can help us locate enhancers is ChromHMM (Ernst and Kellis, 2012). It is a desktop software that discovers and characterises chromatin states from multiple chromatin datasets through automation. Based on a multivariate Hidden Markov Model, ChromHMM accurately models the existence of each chromatin mark. As a result, it can then be used to annotate a genome in one or more cell types orderly; hence locating sequence features such as enhancers (Ernst and Kellis, 2012). Another tool that can locate enhancers is EnhancerPred (Jia and He, 2016). It is a web-based tool that differentiates between enhancers and non-enhancers and measures enhancer’s strength with overall accuracies of 77.39% and 68.19% respectively. From combination of bi-profile Bayes and pseudo-nucleotide composition, it presents as a binary classification problem and EnhancerPred solves it using a machine learning algorithm, which is a two-step rapper-based feature selection method (Jia and He, 2016).

Ribosome binding site (RBS) is a nucleotide sequence upstream of an mRNA transcript’s start codon, which engages a ribosome during protein translation’s initiation. One web-based tool we can use to locate RBS is the Ribosome Binding Site Calculator (Salis, 2011). It is a design method that predicts and controls bacteria’s translation initiation and protein expression. For each start codon in a mRNA transcript, the method can foresee the translation initiation’s rate. A targeted translation initiation rate of a synthetic RBS sequence can be achieved through optimization method. Through the RBS calculator, a protein coding sequence’s translation rate may be logically controlled across a 100,000 fold range (Salis, 2011; Salis *et al.*, 2009). Another desktop tool that we can use is Estimated Locations of Pattern Hits (ELPH; see “Relevant Websites sections”), which takes a set of DNA or protein sequences, up to thousands of sequences, as input and locates motifs using Gibbs sampling. The program seeks through the set of sequences for the most common motif, assuming that each sequence contains one copy of the motif. ELPH can also be used to find patterns such as RBSs and exon splicing enhancers (ESEs).

Coding sequence (CDS) of a gene is the portion of a gene’s DNA, which comprised of exons, that is used to code for protein. One web tool that we can use to find CDS is ORFfinder by NCBI (see “Relevant Websites sections”). An open reading frame (ORF)

is the portion of a DNA sequence, in continuous, non-overlapping triplets called codons, that has the potential to be translated to protein. By finding ORFs, it is equivalent of finding the potential CDSs of a gene. ORF finder is a graphical analysis tool which seeks for ORFs in your input DNA sequence. Other than returning the range of each ORF, it gives its protein translation as well. The predicted protein is verified using SMART BLAST or regular BLASTP. However, the web version of the ORF finder is limited to sequences up to 50 kb long. Another web tool that we can use is ORFPredictor (Min *et al.*, 2005). It is designed to locate protein-coding segments in expressed sequence tag (EST)-derived sequences. For query sequences with a hit in BLASTX, the program predicts the CDS based on the translation reading frames classified in BLASTX alignments. Otherwise, it predicts the most probable CDS based on the inherent signals of the query sequences. The output includes the predicted peptide sequences in the FASTA format and a definition line that consists of the query ID, the translation reading frame and the range of the CDS. However, the ORFPredictor web server is limited to data of file size up to 20 Mb. For datasets exceeding the limitations, there is a standalone desktop version available for download (Min *et al.*, 2005).

Transcription terminator is a DNA segment that indicates the end of a gene or operon during transcription. Together with the preceding promoter, they delimit the transcriptional boundaries. This segment produces signals in the newly synthesized mRNA that initiate processes which break the mRNA off from the transcriptional complex to intervene transcriptional termination. One web-based tool we can use to locate transcription terminal is ARNold (Naville *et al.*, 2011). Rho-independent termination is an important mechanism in bacteria that causes RNA transcription to terminate and release the newly transcribed RNA. ARNold is able to locate rho-independent terminators in DNA sequences. This search method incorporates two complementary prediction programs, Erpin and RNAmotif, and performs approximately as well as more complex methods while being available to the non-specialist. Erpin (Gautheret and Lambert, 2001) generates a lod-score profile and seeks high scoring instances of the profile in the user's sequence with a structure-annotated alignment of 1200 terminator sequences from *Bacillus subtilis* and *Escherichia coli* as training set. On the other hand, RNAmotif (Macke *et al.*, 2001) uses a descriptor developed by Lesnik *et al.* (2001) to recognise terminators. RNAmotif matches are scored using the sequence contents of T-rich region and stability of the stem loop region, with an empirical score cutoff is defined to accept or reject matches. The free energy of the predicted terminator stem-loop structure using RNAfold (Hofacker *et al.*, 1994) is generated to provide a uniform scoring scheme for Erpin and RNAmotif hits. Another way to locate transcription terminators is through WebGeSTer DB (Mitra *et al.*, 2011). It is one of the largest database of intrinsic transcription terminators. The database consists of a million terminators found from 1060 bacterial genome sequences and 798 plasmids. Both graphic and tabular results on putative terminators can be generated and are arranged in tiers to allow easy retrieval. An interactive map has been integrated to visualise the distribution of terminators across the whole genome (Mitra *et al.*, 2011).

DNA methylation happens when a methyl group is added to either one of the DNA's bases, cytosine or adenine. Methylation can modify the activity of a DNA region without changing the sequence. The process commonly is used to repress gene transcription when located in a gene promoter. Methylation is necessary for normal development and gene expression control such as genomic imprinting and X-chromosome inactivation (for males). One desktop tool that we can use is BS-Seeker2 (Guo *et al.*, 2013), an updated version of BS-Seeker (Chen *et al.*, 2010). It is used to map the bisulfite-treated reads and generate DNA methylomes – nucleic acids that undergo methylation modifications. Together with high-throughput sequencing, bisulfite treatment provides a useful approach for studying DNA methylation at base resolution across the genome. The use of libraries such as whole genome bisulfite sequencing (WGBS) and reduced represented bisulfite sequencing (RRBS) to generate DNA methylomes improves mappability, efficiency and accuracy of the reads. BS-Seeker2 also is equipped with additional function to filter out reads with incomplete bisulfite conversion. This will minimise the overestimation of DNA methylation levels (Guo *et al.*, 2013). Another desktop tool that we can use is MethGo (Liao *et al.*, 2015). Similar to BS-Seeker2, this software tool analyses bisulfite sequencing (BS-Seq) data from WGBS and RRBS and provides 9 analyses (both genomic and epigenomic) in 5 major modules to profile (epi) genome. The 9 analyses include coverage distribution of each cytosine, global cytosine methylation level, cytosine methylation level distribution, cytosine methylation level of genomic elements, chromosome-wide cytosine methylation level distribution, gene-centric cytosine methylation level, cytosine methylation levels at transcription factor binding sites (TFBSs), single nucleotide polymorphism (SNP) calling and copy number variation (CNV) calling (Liao *et al.*, 2015).

Besides adenine, cytosine can also be methylated. CpG islands are segments of DNA where a cytosine nucleotide is followed by a guanine nucleotide linearly along its 5' to 3' direction. Cytosine methylation often occurs in CpG islands; hence, finding CpG islands is synonymous to finding cytosine methylation sites. One accurate and fast desktop tool that we can use to identify CpG islands is CpGTLBO (Yang *et al.*, 2017). First, CpGTLBO uses clustering approach to identify CpG island candidates, which effectively cut down the large amount of redundant DNA fragments. Then it uses teaching-learning-based-optimization (TLBO) to accurately predict CpG islands among promising CpG island candidates. CpGTLBO performs more accurate and faster than many CpG island detection methods, that detect based on sliding window and clustering technology, as the accuracy of these methods is proportional to the time required (Yang *et al.*, 2017). Another desktop tool we can use is CpGcluser (Hackenberg *et al.*, 2006), which CpGTLBO (Yang *et al.*, 2017) is built on. This algorithm predicts directly CpG clusters based on the physical distance between neighbouring CpGs on the chromosome. A p-value is given to each of these clusters and the most statistically significant ones can be predicted as CpG islands. As CpGcluser only uses integer arithmetic method, it is a fast and computationally efficient algorithm to predict statistically significant clusters of CpGs. Furthermore, all CpG islands predicted by the algorithm start and end with a CpG dinucleotides, which should be relevant for a genomic feature whose functionality is based directly on CpG dinucleotides (Hackenberg *et al.*, 2006).

Secondary structure is the set of interactions between nucleotide bases, i.e., which parts of strands are likely to bind to each other. In DNA double helix, secondary structure is responsible for the 3-dimensional shape of the nucleic acids. One web-based

tool that we can use is Kinefold (Xayaphoummine *et al.*, 2005). It simulates stochastic folding of nucleic acids on second to minute time scales. Simulations of renaturation or co-transcriptional folding paths are done at the level of helix formation and dissociation in agreement with the seminal experimental results. Efficient predictions of pseudoknots and topologically “entangled” helices are done, taking into considerations due to simple geometrical and topological constraints. Simulations launched are automatically stopped from time to time to allow users to have the freedom to choose to continue incomplete simulations or modify recommended options provided by the server in their initial query. The web server provides detailed output that include a series of low free energy structures, an online animated folding path and a programmable trajectory plot focusing on a few helices of interest to each user (Xayaphoummine *et al.*, 2005). We can use another desktop tool called Unified Nucleic Acid Folding or abbreviated to UNAFold (Markham and Zuker, 2008). This software package combines different programs to simulate folding, hybridization, and melting pathways for one or two-stranded nucleic acid sequences (DNA or RNA). Folding prediction for single-stranded RNA or DNA merges free energy minimization, partition function calculations and stochastic sampling. Images of secondary structures, hybridizations, and dot plots may be computed using common formats. The program computes not only melting temperatures, but also full melting profiles. UV absorbance at 260 nm, heat capacity change and different molecular species’ mole fractions are generated as a function of temperature. The software is “command line” driven. Underlying complied programs may be used independently, or in special combinations with the use of Perl scripts (Markham and Zuker, 2008).

Applications of Sequence Composition

The application of sequence composition is extensive. In this section, four recent studies are highlighted to showcase the applications of sequence composition analysis. Eukaryotic RNA often undergo RNA processing to form mature mRNA, and one of the important steps is splicing, which is to excise the introns from the immature RNA and join the resulting exons. Hence, identifying the intron/exon boundaries on a sequence is useful to computing the resulting peptide from genomic sequence. A study (Zafri and Tuller, 2015) using 4 fungi; *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe*, *Aspergillus nidulans*, and *Candida albicans*; found that intron/exon boundaries often exhibit weak RNA folding and a decrease in GC content. This suggests that current RNA sequence analysis tools, such as RNAfold (Hofacker *et al.*, 1994) and ViennaRNA (Lorenz *et al.*, 2011), may be useful for other purposes.

The ability to predict gene expression from sequence is useful to have (Ling and Poh, 2014), especially when experimentally derived expression data is lacking. A study (Su *et al.*, 2016) proposed to use DNA sequence features; such as histone modifications, DNA methylation, DNA accessibility, transcription factors, and trinucleotide composition; in a support vector machine to predict gene expression. Su *et al.* (2016) achieved a predictive accuracy of 95.96% on human embryonic stem cell line H1, presenting the work as a predictive model for binary classification into high or low expression when experimentally derived expression data is lacking.

Assembly of reads from next generation sequencing experiments often results in large sequence fragments, known as contigs, rather than entire genome. Each contig is considered as an operational taxonomic unit and related contigs needs to be binned together for further analysis. Several software for binning contigs exist and some tools; such as CONCOCT (Alneberg *et al.*, 2014), employed sequence composition in its computation. In this case, sequence composition is used a data point for computation, which acts like a compression of the sequence has the effect of reducing the feature space compared to using raw sequences. Recently, a new tool known as COCACOLA (Lu *et al.*, 2017) that incorporates sequence composition had been proposed, and demonstrated a precision of 99.78% and a recall of 99.93%, compared to precision of 93.43% and recall of 99.6% from CONCOCT.

Many eukaryotes contains accessory chromosomes, known as B chromosomes, in contrast to standard chromosomes or A chromosomes (Houben, 2017). Sequence analysis of several B chromosomes revealed B chromosomes contains DNA originating from standard chromosomes, such as in rye (Klemme *et al.*, 2013). This suggests that B chromosomes may serve an evolutionary purpose and further demonstrates that sequence analysis is an important aspect of evolutionary genetics (Ruban *et al.*, 2017).

See also: Natural Language Processing Approaches in Bioinformatics. Sequence Analysis

References

- Abeel, T., Saeyns, Y., Rouzé, P., Van de Peer, Y., 2008. ProSOM: Core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics* 24, i24–i31.
- Alneberg, J., Bjarnason, B.S., de Bruijn, I., *et al.*, 2014. Binning metagenomic contigs by coverage and composition. *Nat. Methods* 11, 1144–1146.
- Chen, P.-Y., Cokus, S.J., Pellegrini, M., 2010. BS seeker: Precise mapping for bisulfite sequencing. *BMC Bioinform.* 11, 203.
- Cowman, A.F., Crabb, B.S., 2002. The *Plasmodium falciparum* genome – A blueprint for erythrocyte invasion. *Science* 298, 126–128.
- Domenech, V.S., Nylen, E.S., White, J.C., *et al.*, 2001. Calcitonin gene-related peptide expression in sepsis: Postulation of microbial infection-specific response elements within the calcitonin I gene promoter. *J. Investig. Med.* 49, 514–521.
- Ernst, J., Kellis, M., 2012. ChromHMM: Automating chromatin-state discovery and characterization. *Nat. Methods* 9, 215–216.

- Gautheret, D., Lambert, A., 2001. Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *J. Mol. Biol.* 313, 1003–1011.
- Guo, W., Fiziev, P., Yan, W., *et al.*, 2013. BS-Seeker2: A versatile aligning pipeline for bisulfite sequencing data. *BMC Genom.* 14, 774.
- Hackenberg, M., Previti, C., Luque-Escamilla, P.L., *et al.*, 2006. CpGcluster: A distance-based algorithm for CpG-island detection. *BMC Bioinform.* 7, 446.
- Hofacker, I.L., Fontana, W., Stadler, P.F., *et al.*, 1994. Fast folding and comparison of RNA secondary structures. *Monatshfte Chem./Chem. Mon.* 125, 167–188.
- Houben, A., 2017. B chromosomes – A matter of chromosome drive. *Front. Plant Sci.* 8, 210.
- Jia, C., He, W., 2016. EnhancerPred: A predictor for discovering enhancers based on the combination and selection of multiple features. *Sci. Rep.* 6, 38741.
- Klemme, S., Banaei-Moghaddam, A.M., Macas, J., *et al.*, 2013. High-copy sequences reveal distinct evolution of the rye B chromosome. *New Phytol.* 199, 550–558.
- Lesnik, E.A., Sampath, R., Levene, H.B., *et al.*, 2001. Prediction of rho-independent transcriptional terminators in *Escherichia coli*. *Nucleic Acids Res.* 29, 3583–3594.
- Liao, W.-W., Yen, M.-R., Ju, E., *et al.*, 2015. MethGo: A comprehensive tool for analyzing whole-genome bisulfite sequencing data. *BMC Genom.* 16, S11.
- Ling, M.H., Poh, C.L., 2014. A predictor for predicting *Escherichia coli* transcriptome and the effects of gene perturbations. *BMC Bioinform.* 15, 140.
- Lorenz, R., Bernhart, S.H., Höner zu Siederdissen, C., *et al.*, 2011. ViennaRNA package 2.0. *Algorithms Mol. Biol.*: AMB 6, 26.
- Lu, Y.Y., Chen, T., Fuhrman, J.A., Sun, F., 2017. COCACOLA: Binning metagenomic contigs using sequence composition, read coverage, co-alignment and paired-end read linkage. *Bioinformatics* 33, 791–798.
- Macke, T.J., Ecker, D.J., Gutell, R.R., *et al.*, 2001. RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res.* 29, 4724–4735.
- Markham, N.R., Zuker, M., 2008. UNAFold: Software for nucleic acid folding and hybridization. *Methods Mol. Biol.* 453, 3–31.
- Min, X.J., Butler, G., Storms, R., Tsang, A., 2005. OrfPredictor: Predicting protein-coding regions in EST-derived sequences. *Nucleic Acids Res.* 33, W677–W680.
- Mitra, A., Kesarwani, A.K., Pal, D., Nagaraja, V., 2011. WebGeSTer DB – A transcription terminator database. *Nucleic Acids Res.* 39, D129–D135.
- Moxon, E.R., Higgins, C.F., 1997. *E. coli* genome sequence. A blueprint for life. *Nature* 389, 120–121.
- Naville, M., Ghuilliot-Gaudeffroy, A., Marchais, A., Gautheret, D., 2011. ARNold: A web tool for the prediction of Rho-independent transcription terminators. *RNA Biol.* 8, 11–13.
- Ruban, A., Schmutzer, T., Scholz, U., Houben, A., 2017. How next-generation sequencing has aided our understanding of the sequence composition and origin of B chromosomes. *Genes (Basel)* 8.
- Salis, H.M., 2011. The ribosome binding site calculator. *Methods Enzymol.* 498, 19–42.
- Salis, H.M., Mirsky, E.A., Voigt, C.A., 2009. Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol.* 27, 946–950.
- Su, W.-X., Li, Q.-Z., Zhang, L.-Q., *et al.*, 2016. Gene expression classification using epigenetic features and DNA sequence composition in the human embryonic stem cell line H1. *Gene* 592, 227–234.
- Wang, X., Xuan, Z., Zhao, X., Li, Y., Zhang, M.Q., 2009. High-resolution human core-promoter prediction with CoreBoost_HM. *Genome Res.* 19, 266–275.
- Xayaphoummine, A., Bucher, T., Isambert, H., 2005. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic Acids Res.* 33, W605–W610.
- Yang, C.-H., Chiang, Y.-C., Chuang, L.-Y., Lin, Y.-D., 2017. A CpGcluster-teaching-learning-based optimization for prediction of CpG islands in the human genome. *J. Comput. Biol.*
- Zafir, Z., Tuller, T., 2015. Nucleotide sequence composition adjacent to intronic splice sites improves splicing efficiency via its effect on pre-mRNA local folding in fungi. *RNA* 21, 1704–1718.
- Zhao, X., Xuan, Z., Zhang, M.Q., 2007. Boosting with stumps for predicting transcription start sites. *Genome Biol.* 8, R17.

Relevant Websites

- <http://www.cbcb.umd.edu/software/ELPH/>
ELPH User Manual.
- <https://www.ncbi.nlm.nih.gov/orffinder/>
National Center for Biotechnology Information.

Codon Usage

Raimi M Redwan, Faculty of Agro-based Industry, University of Malaysia, Kelantan, Malaysia
Suhanya Parthasarathy and Ranjeev Hari, Perdana University, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

A genome contains all the necessary information required for a system to function optimally at any given condition. The central dogma skeleton dictates the flow of genetic information from genome to protein via RNA. The 20 amino acids are decoded within the genome using the combination of the four letters of DNA language in three base pair combinations (i.e., triplets of bases), known as DNA codons. One feature of the DNA codons is that the code is degenerate as one amino acid can be coded by more than one codon. Redundancy of the codons can be explained from the excessive 64 possible combination of the four nucleotide base pairs in a triplet base ($4^3 = 64$) to represent the 20 amino acids. As a consequence, only methionine and tryptophan are encoded by a single codon, whereas the other 18 amino acids are encoded by more than one codon. The genetic code table showed that the degeneracy mostly occurred due to variation in the third base of the codons. It is due to this redundancy that different DNA sequences can still produce identical protein sequences, a situation known to be synonymous mutation.

As the DNA sequences become accessible through sequencing, information regarding the codon usage in the coding sequences can be obtained. This data led to the discovery that these codons are not being used equally uniform throughout the genome. Earlier studies have identified the presence of preferred codons to encode for specific amino acids. Variation of choice of codons to represent amino acids is not only observed among species from different taxonomic group, but also showed significant variation among individuals of the same species, across different genes in the same genome and even across regions in the same gene. Nevertheless, the codon bias is most prominent in species from different taxonomic groups even in proteins with identical function. This phenomenon of species-specific codon choice is known as “codon dialect,” which signifies the codon-usage bias observed across different organisms (Ikemura, 1985).

GC Content and Codon-Usage Bias

Earlier, the study of codon usage focused on the DNA composition at the last base pair of the codons, the position where degeneracy mostly occurred. In relation to that, preference of the base content at this location also distinguished the codon bias observed in monocots and dicots. In monocots, 16 amino acids favor G + C at the third degenerate base, whereas in dicots only 7 out of 18 amino acids prefer G + C at this base (Murray *et al.*, 1989; Tyson and Dhindsa, 1995). Nonetheless, when there is C at the second codon's base, both groups of plants avoid G in their last codon's base as dinucleotide combination of CG may result in methylation of the DNA. Dinucleotide combination of CG is not only avoided in plants but also other higher eukaryotes, where methylation plays a role in gene regulation. Preference of base in choice of synonymous codons is known to be related to its GC content. In *Mycoplasma*, which is known to be AT rich at 60%–75%, has codon preferences toward A and T in various coding region (Muto *et al.*, 1986). Furthermore, homologous cDNA of human gene sequences derived from different tissue sample contained 80% amino acid sequence similarity but with significant G + C content variation. In consequences, the sequences were also identified to have significant variation of codon bias, especially at the third codons (Kudla *et al.*, 2009). This observation supports that the total G + C content in genome or within genes is the determinants of the variation in codon usage. This variation is hypothesized to be attributed by the mutational mechanism that directly controlled the GC content throughout the genome (Chen *et al.*, 2004). Positive correlation between the GC content and the codon usage is advantageous as the GC content data can be exploited to predict the reading frame and the coding strands of DNA sequences (Bibb *et al.*, 1984). Recently, a study highlighted the important role of the genome's GC content to influence the genome-wide codon-usage bias in the mammalian cell (Rudolph *et al.*, 2016).

Influence of tRNA Abundance on Codon Bias

Another determinant of the codon-usage variation is the tRNA abundance. As an adapter molecule to link the mRNA to the amino acid in protein synthesis, the abundance of the complementary tRNA molecules are directly related to the frequency of codons (Ikemura, 1985). The investigation was based on the finding that codon bias is the most extreme in highly expressed genes where translational efficiency would be a priority. The finding established a selection-based hypothesis that the selection of optimal codons in a certain gene is a function of the gene's adaptation to the complementary tRNA molecule available to increase efficiency of the protein synthesis. Following this observation, a metric known as tRNA adaptation index (tAI) was developed to analyze the correlation between the abundance of tRNAs and translational efficiency (dos Reis *et al.*, 2004). The metric measures the copy number of tRNA genes, which is assumed to be correlated with tRNA abundance in cells.

It also takes into consideration the efficiency of the binding between the codon–anticodon and the possibility of the imperfect match between the codon–anticodon at the third position (i.e., Crick’s wobble rules) (Crick, 1966). Most importantly, the metric enables genome-wide estimation of translational efficiency through the abundance of tRNAs, which is directly related to codon bias. The metric has been used successfully to predict translation efficiency of 2800 yeast orthologous genes based on the measures of tRNA adaptation to conform the codon usage (Man and Pilpel, 2007). Moreover, the metric has also been shown to be in correlation with the ratio of protein to mRNA level in *Saccharomyces cerevisiae*, which has been able to support the tRNA adaptation hypothesis (Tuller et al., 2010b).

The tRNA adaptation theory to shape the codon composition in genes is not applicable universally across all kingdoms. The hypothesis could explain variation of codon composition in most of the prokaryotes tested (Andersson and Kurland, 1990) but not in eukaryotes. Kanaya et al. (2001) showed that different eukaryotes have different selectional forces to determine their codon composition. In his study, only *Schizosaccharomyces pombe* and *Caenorhabditis elegans* showed positive correlation patterns of codon usage and tRNA gene number. On the other hand, *Drosophila melanogaster* showed the important influence of C or G base in the third codons to shape its codon usage. *Xenopus laevis* and *Homo sapiens* showed the role of isochore structure of mRNA produced to influence their codon composition (Kanaya et al., 2001). It was suggested that multicellular organisms of eukaryotes have higher tRNA gene redundancy, which posits lower selection pressure to compose their codons based on the tRNA availability (Quax et al., 2015). In addition, the disparity of determinants for codon composition may also be related to the different strength of optimal codon bias in eukaryotic genome as was shown in Dos Reis and Wernisch (2009). Nevertheless, recent evidence showed that translational selection (i.e., mechanism of translational optimization based on the natural selection of the codon usage) was identified to be the positive factor contributing to codon bias in housekeeping-genes of human (Ma et al., 2014), contradicting the previous findings (Kanaya et al., 2001). Moreover, the hypothesis is favored by the finding that there is a preference to employ the same tRNA leading to repetitive use of a similar set of optimal codons as a way to optimize translation speed especially in regions of highly expressed genes (Canarozzi et al., 2010). Preferential optimal codons identified in highly expressed genes were also identified to be limited by the availability of tRNAs to increase its translational efficiency (Qian et al., 2012).

Codon Adaptation in Composition of Codon

Composition of codon in genes is known to affect the fate of the protein being synthesized as it influences level of expression, protein folding, and the regulation of protein expression. This observation led to the same common question as the translational selection as discussed above and that is the role of codon bias in determination of the proteins’ expression level. As a consequence, each species has its own set of “preferred codons” identified within the highly expressed genes that is assumed to translate efficiently to ensure optimal protein synthesis. Relative to the finding, a matrix known as codon adaptation index (CAI) was developed to assess the relative adaptation of individual codons encoding a certain amino acid. The index measures the ratio of the codon’s frequency to the frequency of its other synonymous codons. The index was measured from the highly expressed genes as the set of reference and the CAI for a gene can then be derived through the geometric mean of the relative adaptiveness values of all the codon within the gene (Sharp and Li, 1987). Similar to tAI, CAI also enables prediction of gene expression level but on the basis of the choice of codons encoding the gene (Sen et al., 2007; Wu et al., 2005). The index was also shown to be highly correlated to mRNA concentration of most of the genes tested in Coghlan and Wolfe (2000) and in microarray study of Martín-Galiano et al. (2004). Nonetheless, the CAI index has its own caveats to its predictive power. This is due to the fact that the index relies on a set of reference genes to derive its index, which might imposed limited value to predict expression for genes not reflected in the reference set (Martín-Galiano et al., 2004). The index also failed to include other positive factors influencing the expression value of the composed codons (Ermolaeva, 2001). Consequently, few studies showed a disparity of CAI index and the actual expression value of the genes tested (dos Reis et al., 2003; Kudla et al., 2009). Nonetheless, the index is still widely used as a prediction tool especially in de novo gene synthesis, whereby the CAI value is taken into consideration in the codon optimization algorithm (Chung and Lee, 2012; Condon and Thachuk, 2012; Nandagopal and Elowitz, 2011). The algorithm is developed based on several known factors to influence the expression level of genes based on the composition of codons, other than just the CAI. In addition, there are also other indexes developed to predict expression value based on the codon usage. Some of the alternatives to CAI are relative codon-usage bias (Roymondal et al., 2009), expression measure $E(g)$ (Roymondal et al., 2009), relative codon adaptation (Fox and Erill, 2010) and modified relative codon bias strength (Das et al., 2017).

Positive correlation of the codon composition and its expression level implied that synonymous mutation among orthologous gene products, be it across different individuals or tissues, may no longer be considered as silent (Chamary et al., 2006; Kimchi-Sarfaty et al., 2007; Shields et al., 1988). The choice of codons, even though they are synonymous with respect to amino acids they encode, the information determines the mRNA structure and stability and affecting the protein structure, and its folding kinetics. This was depicted in a study when 154 synthetic orthologous genes with various synonymous variants in their codon showed different level of mRNA level and degradation rates (Kudla et al., 2009). It is important to note that the variant was not entirely caused by the composition of preferred codons throughout the genes but due to the variation in the ribosomal binding site that changed the stability of mRNA folding.

Codon Ramp at Translational Initiation

In separate studies, investigation of coding sequences from both prokaryotes and eukaryotes revealed a presence of evolutionarily conserved codon composition known as “ramp” at the 5' end of the sequences (Tuller *et al.*, 2010a). The author highlights the role of codon composition in controlling translation efficiency through codon and tRNA adaptation and also via the presence of codon ramp as a mean to reduce ribosomal traffic jams in the beginning of translation. Even though the feature is evolutionarily conserved across domains of life, the ramp may not be universally present in all genes in every genome but only in highly expressed genes as inspected in the study. It was also suggested that the ramp functions to sense the abundance of tRNAs and changes its ramp's design in response to the different conditions based on the observation that different functional genes contained different design of the ramp. The condition-specific ramp is still an open question and thus far no study has yet validated the claim.

Study of translational efficiency and codon usage advance forward with the emerging new techniques of ribosome profiling. The technique used deep sequencing technologies to profile *in vivo* translation (Ingolia, 2014). By using the approach, a study was able to prove slower elongation rate upon reduction of heavily used tRNA and at the beginning of a message (i.e., ramp) (Pop *et al.*, 2014). Moreover, another two studies were also able to validate the slower translational elongation rate of the nonoptimal or the rare codons that have low respective tRNA population (Dana and Tuller, 2014; Gardin *et al.*, 2014).

Bioinformatics Tools to Analyze Codon Usage

As described earlier, CAI is one of the indexes used apart from several other measures of codon usage. There are a large number of indices that cover a wide area of underlying biological features (Roth *et al.*, 2012). Typically, these indices are used to denote a measure of departure from an expected value, for instance, of codon distribution based on nucleotide frequencies. Similarly, some indices measure the closeness to a hypothetical optimal codon state. In turn, such measures get compared to preferred codon usage of reference set of genes that may be optimal, highly expressed, in a specifically defined group or, on the contrary, genes encompassing the whole genome.

While a large amount of codon measures has been developed, the choice of the index will depend on the goal being achieved as every index has its specific aspect of codon usage being measured. The CAI has been widely used and known for its usage of measuring codon-usage bias (Cannarozzi *et al.*, 2010; Friberg *et al.*, 2004). Several other complementary indices provide a measure of understanding diversity and recommend using an ensemble of features to better classify aspects such as translational efficiency (Roth *et al.*, 2012; Tuller *et al.*, 2010a). Having diverse methods of measuring codon usage requires benchmarks in order to test its statistical performance in achieving the goals comparatively. Some studies have described comparisons across measures, however, the coverage across the many indices available is still lacking (Comeron and Aguadé, 1998; Supek and Vlahoviček, 2005; Suzuki *et al.*, 2008). Partly, this is due to the scarce availability of reproducible computational implementation of those measures by the authors in the form of programming scripts or software.

Nevertheless, there are a few tools that have the ability to calculate the desired indices easily based on DNA sequences. INCA provides the ability to calculate CAI and effective Nc (ENC) besides the ability to apply principal component analysis (PCA), self-organizing maps, and 3D scatterplot to visualize any correlations in the high-dimensional data being presented (Supek and Vlahoviček, 2004). Moreover, Dnasp, a multipurpose population genetics software program, can compute relative synonymous codon usage, ENC, codon bias index, and the scaled chi square (Morton, 1993; Sharp *et al.*, 1986; Shields *et al.*, 1988; Wright, 1990). Other software tools that offer similar calculation of indices are GCUA, ACUA, and CodonW (McInerney, 1998; Peden, 1997; Vetrivel *et al.*, 2007). Most of this software includes some statistical analysis method such as multivariate analysis, PCA, and correspondence analyses (CA). Comparison of statistical methods for CA showed that within-group CA performed well in identifying new factors contributing to variation and horizontal transfers of genes more accurately (Suzuki *et al.*, 2008). To support the use of the analysis, an online resource to perform such analysis is available Pôle Bioinformatique Lyonnais server (Charif *et al.*, 2005).

Bioinformatics Tools to Optimize Codon Usage in DNA

Besides serving the specific purposes to better understand codon variation, researchers may use the principle behind the finding to synthesize artificial DNA. Codon bias does not affect the amino acid sequence itself, nevertheless, it may have impact in protein production. Directly, the biotech application to optimize the DNA sequence is crucial for optimal production of target protein in bioreactors. For such purposes, there are software programs that aid in returning the optimal form of codon usage for a DNA sequence such as DNAworks and OPTIMIZER (Hoover and Lubkowski, 2002; Puigbò *et al.*, 2007). DNAWorks has a standard threshold that it uses in the optimization step to pick the two highest frequency codons for encoding the amino acid. In contrast, OPTIMIZER server provides precomputed tables for around 150 prokaryotic genomes that are found to be under strong translational selection while optimizing all codons while introducing the most minimum changes to the DNA submitted.

Applications in Systems and Synthetic Biology

Through natural selection, optimal codon usages for diverse types of genes and organisms were formed. While the codon optimization field is inevitably moving away from older ideas that synthetic genes should contain mostly of high-frequency codons, the real challenge remains on methods in replicating high protein production for robust production systems in biotechnology. There could be many features that are important for synthetic gene design such as synonymous codon cooccurrence bias, nonsynonymous codon pair bias, tRNA-associated properties, and expression (Quax *et al.*, 2015). However, there is a lack of attention of incorporating these features in synthetic biology. For instance, recording of tRNA-associated properties such as charged tRNA and regular tRNA abundance levels in high protein expression conditions will provide hints for better codon optimization to maintain tRNA levels for increased production of target protein. Besides that, application of the “codon harmonization” principles could be better suited in achieving efficiency in robust yet demanding heterologous production environments (Angov *et al.*, 2008; Quax *et al.*, 2015).

Some of the exciting implications of better rational gene design in synthetic biology are the creation of synthetic biosynthetic pathways, gene circuits, or even entirely new genomes. Realistically, timely expression of functionally networked genes is crucial in the design of such biocircuitry. Some progress made in differential codon bias and differentially expressed prokaryotic cistrons may provide blueprints for designing synthetic operons to express these circuits or pathways (Li *et al.*, 2014; Quax *et al.*, 2013). Moreover, rapid advances in DNA assembly recently allowed the assembly and transplantation of complete synthetic genomes (Gibson, 2014). Essentially, such a protocol enables the redesign of genomes from scratch and allows its reemergence from another compatible host. Nevertheless, better understanding of codon bias could be helpful in the systematic design of an optimally functional synthetic genome by incorporating sensible modules that consider codon usage, tRNA gene levels, and tRNA-modifying enzymes.

Conclusion

The study of translational efficiency and codon usage is still an open debate. Several studies are in support of the selection theory in codon composition while others refuted and claim other features of the mRNAs are more significant to contribute in its efficiency. Nonetheless as technologies advance forward and more techniques prevail, the question demands more evidence to fulfill the answer. The understanding of codon composition is important not only for de novo gene synthesis but to understand the regulation of genes in different conditions, information that seems to be hidden in what used to be a silent mutation. Furthermore, in the future, codon usage principles might find relevance in synthetic biology, for instance in optimizing the production of target proteins.

See also: Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition

References

- Andersson, S.G., Kurland, C.G., 1990. Codon preferences in free-living microorganisms. *Microbiology Reviews* 285, 198–210.
- Angov, E., Hillier, C.J., Kincaid, R.L., Lyon, J.A., 2008. Heterologous protein expression is enhanced by harmonizing the codon usage frequencies of the target gene with those of the expression host. *PLOS ONE* 3, e2189.
- Bibb, M.J., Findlay, P.R., Johnson, M.W., 1984. The relationship between base composition and codon usage in bacterial genes and its use for the simple and reliable identification of protein-coding sequences. *Gene* 30, 157–166.
- Cannarozzi, G., Schraudolph, N.N., Faty, M., *et al.*, 2010. A role for codon order in translation dynamics. *Cell* 141, 355–367. Available at: <https://doi.org/10.1016/j.cell.2010.02.036>.
- Chamary, J.V., Parmley, J.L., Hurst, L.D., 2006. Hearing silence: Non-neutral evolution at synonymous sites in mammals. *Nature Reviews Genetics* 7, 98–108.
- Charif, D., Thioulouse, J., Lobry, J.R., Perrière, G., 2005. Online synonymous codon usage analyses with the *ade4* and *seqinR* packages. *Bioinformatics* 21, 545–547. Available at: <https://doi.org/10.1093/bioinformatics/bti037>.
- Chen, S.L., Lee, W., Hottes, A.K., Shapiro, L., McAdams, H.H., 2004. Codon usage between genomes is constrained by genome-wide mutational processes. *Proceedings of the National Academy of Sciences of the United States of America* 101, 3480–3485. Available at: <https://doi.org/10.1073/pnas.0307827100>.
- Chung, B.K.-S., Lee, D.-Y., 2012. Computational codon optimization of synthetic gene for protein expression. *BMC Systems Biology* 6, 134. Available at: <https://doi.org/10.1186/1752-0509-6-134>.
- Coghlan, A., Wolfe, K.H., 2000. Relationship of codon bias to mRNA concentration and protein length in *Saccharomyces cerevisiae*. *Yeast* 16, 1131–1145. Available at: [https://doi.org/10.1002/1097-0061\(20000915\)16:12<1131::AID-YEA609.3.0.CO;2-F](https://doi.org/10.1002/1097-0061(20000915)16:12<1131::AID-YEA609.3.0.CO;2-F).
- Comeron, J.M., Aguadé, M., 1998. An evaluation of measures of synonymous codon usage bias. *Journal of Molecular Evolution* 47, 268–274.
- Condon, A., Thachuk, C., 2012. Efficient codon optimization with motif engineering. *Journal of Discrete Algorithms* 16, 104–112. Available at: <https://doi.org/https://doi.org/10.1016/j.jda.2012.04.017>.
- Crick, F.H., 1966. Codon–anticodon pairing: The wobble hypothesis. *Journal of Molecular Evolution* 19, 548–555.
- Dana, A., Tuller, T., 2014. The effect of tRNA levels on decoding times of mRNA codons. *Nucleic Acids Research* 42, 9171–9181. Available at: <https://doi.org/10.1093/nar/gku646>.
- Das, S., Chottopadhyay, B., Sahoo, S., 2017. Comparative analysis of predicted gene expression among crenarchaeal genomes. *Genomics Information* 15, 38–47.
- Ermolaeva, M.D., 2001. Synonymous codon usage in bacteria. *Current Issues in Molecular Biology* 3, 91–97.

- Fox, J.M., Erill, I., 2010. Relative codon adaptation: A generic codon bias index for prediction of gene expression. *DNA Research: An International Journal for Rapid Publication of Reports on Genes and Genomes* 17, 185–196. Available at: <https://doi.org/10.1093/dnares/dsq012>.
- Friberg, M., von Rohr, P., Gonnet, G., 2004. Limitations of codon adaptation index and other coding DNA-based features for prediction of protein expression in *Saccharomyces cerevisiae*. *Yeast* 21, 1083–1093.
- Gardin, J., Yeasmin, R., Yurovsky, A., *et al.*, 2014. Measurement of average decoding rates of the 61 sense codons in vivo. *eLife*. 3. Available at: <https://doi.org/10.7554/eLife.03735>.
- Gibson, D.G., 2014. Programming biological operating systems: Genome design, assembly and activation. *Nature Methods* 11, 521–526.
- Hoover, D.M., Lubkowsky, J., 2002. DNAWorks: An automated method for designing oligonucleotides for PCR-based gene synthesis. *Nucleic Acids Research* 30, e43.
- Ikemura, T., 1985. Codon usage and tRNA content in unicellular and multicellular organisms. *Molecular Biology and Evolution* 2, 13–34. Available at: <https://doi.org/10.1093/oxfordjournals.molbev.a040335>.
- Ingolia, N.T., 2014. Ribosome profiling: New views of translation, from single codons to genome scale. *Nature Reviews Genetics* 15, 205–213. Available at: <https://doi.org/10.1038/nrg3645>.
- Kanaya, S., Yamada, Y., Kinouchi, M., Kudo, Y., Ikemura, T., 2001. Codon usage and tRNA genes in eukaryotes: Correlation of codon usage diversity with translation efficiency and with CG-dinucleotide usage as assessed by multivariate analysis. *Journal of Molecular Evolution* 53, 290–298. Available at: <https://doi.org/10.1007/s002390010219>.
- Kimchi-Sarfaty, C., Oh, J.M., Kim, I.-W., *et al.*, 2007. A “silent” polymorphism in the MDR1 gene changes substrate specificity. *Science* 315, 525–528. Available at: <https://doi.org/10.1126/science.1135308>.
- Kudla, G., Murray, A.W., Tollervey, D., Plotkin, J.B., 2009. Coding-sequence determinants of gene expression in *Escherichia coli*. *Science* 324, 255 LP–255258. Available at: <https://doi.org/10.1126/science.1170160>.
- Li, G.-W., Burkhardt, D., Gross, C., Weissman, J.S., 2014. Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell* 157, 624–635.
- Man, O., Pilpel, Y., 2007. Differential translation efficiency of orthologous genes is involved in phenotypic divergence of yeast species. *Nature Genetics* 39, 415–421. Available at: <https://doi.org/10.1038/ng1967>.
- Martin-Galiano, A.J., Wells, J.M., de la Campa, A.G., 2004. Relationship between codon biased genes, microarray expression values and physiological characteristics of *Streptococcus pneumoniae*. *Microbiology* 150, 2313–2325. Available at: <https://doi.org/10.1099/mic.0.27097-0>.
- Ma, L., Cui, P., Zhu, J., Zhang, Z., Zhang, Z., 2014. Translational selection in human: More pronounced in housekeeping genes. *Biology Direct* 9, 17. Available at: <https://doi.org/10.1186/1745-6150-9-17>.
- McInerney, J.O., 1998. GCUA: General codon usage analysis. *Bioinformatics* 14, 372–373.
- Morton, B.R., 1993. Chloroplast DNA codon use: Evidence for selection at the psb A locus based on tRNA availability. *Journal of Molecular Evolution* 37, 273–280.
- Murray, E.E., Lotzer, J., Eberle, M., 1989. Codon usage in plant genes. *Nucleic Acids Research* 17, 477–498.
- Muto, A., Yamao, F., Hori, H., Osawa, S., 1986. Gene organization of *Mycoplasma capricolum*. *Advances in Biophysics* 21, 49–56.
- Nandagopal, N., Elowitz, M.B., 2011. Synthetic biology: Integrated gene circuits. *Science*. 333. Available at: <https://doi.org/10.1126/science.1207084>.
- Peden, J., 1997. CodonW: Correspondence Analysis of Codon Usage. United Kingdom: Nottingham University.
- Pop, C., Rouskin, S., Ingolia, N.T., *et al.*, 2014. Causal signals between codon bias, mRNA structure, and the efficiency of translation and elongation. *Molecular Systems Biology* 10, 770. Available at: <https://doi.org/10.15252/msb.20145524>.
- Puigbó, P., Guzmán, E., Romeu, A., Garcia-Vallvé, S., 2007. OPTIMIZER: A web server for optimizing the codon usage of DNA sequences. *Nucleic Acids Research* 35, W126–W131. Available at: <https://doi.org/10.1093/nar/gkm219>.
- Qian, W., Yang, J.-R., Pearson, N.M., Maclean, C., Zhang, J., 2012. Balanced codon usage optimizes eukaryotic translational efficiency. *PLOS Genetics* 8, e1002603.
- Quax, T.E.F., Claessens, N.J., Söhl, D., van der Oost, J., 2015. Codon bias as a means to fine-tune gene expression. *Molecular Cell* 59, 149–161. Available at: <https://doi.org/10.1002/aur.1474>. Replication.
- Quax, T.E.F., Wolf, Y.I., Koehorst, J.J., *et al.*, 2013. Differential translation tunes uneven production of operon-encoded proteins. *Cell Reports* 4, 938–944. Available at: <https://doi.org/10.1016/j.celrep.2013.07.049>.
- dos Reis, M., Savva, R., Wernisch, L., 2004. Solving the riddle of codon usage preferences: A test for translational selection. *Nucleic Acids Research* 32, 5036–5044. Available at: <https://doi.org/10.1093/nar/gkh834>.
- Dos Reis, M., Wernisch, L., 2009. Estimating translational selection in eukaryotic genomes. *Molecular Biology and Evolution* 26, 451–461. Available at: <https://doi.org/10.1093/molbev/msn272>.
- dos Reis, M., Wernisch, L., Savva, R., 2003. Unexpected correlations between gene expression and codon usage bias from microarray data for the whole *Escherichia coli* K-12 genome. *Nucleic Acids Research* 31, 6976–6985.
- Roth, A., Anisimova, M., Cannarozzi, G.M., 2012. Measuring codon usage bias. In: *Codon Evolution: Mechanisms and Models*. NY: Oxford University Press Inc., pp. 189–217.
- Roymondal, U., Das, S., Sahoo, S., 2009. Predicting gene expression level from relative codon usage bias: An application to *Escherichia coli* genome. *DNA Research* 16, 13–30. Available at: <https://doi.org/10.1093/dnares/dsn029>.
- Rudolph, K.L.M., Schmitt, B.M., Villar, D., *et al.*, 2016. Codon-driven translational efficiency is stable across diverse mammalian cell states. *PLOS Genetics* 12, 1–23. Available at: <https://doi.org/10.1371/journal.pgen.1006024>.
- Sen, G., Sur, S., Bose, D., *et al.*, 2007. Analysis of codon usage patterns and predicted highly expressed genes for six phytopathogenic *Xanthomonas* genomes shows a high degree of conservation. In *Silico Biology* 7, 547–558.
- Sharp, P.M., Li, W.H., 1987. The codon Adaptation Index – A measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research* 15, 1281–1295.
- Sharp, P.M., Tuohy, T.M., Mosurski, K.R., 1986. Codon usage in yeast: Cluster analysis clearly differentiates highly and lowly expressed genes. *Nucleic Acids Research* 14, 5125–5143.
- Shields, D.C., Sharp, P.M., Higgins, D.G., Wright, F., 1988. “Silent” sites in *Drosophila* genes are not neutral: Evidence of selection among synonymous codons. *Molecular Biology and Evolution* 5, 704–716.
- Supek, F., Vlahoviček, K., 2004. INCA: Synonymous codon usage analysis and clustering by means of self-organizing map. *Bioinformatics* 20, 2329–2330.
- Supek, F., Vlahoviček, K., 2005. Comparison of codon usage measures and their applicability in prediction of microbial gene expressivity. *BMC Bioinformatics* 6, 182. Available at: <https://doi.org/10.1186/1471-2105-6-182>.
- Suzuki, H., Brown, C.J., Forney, L.J., Top, E.M., 2008. Comparison of correspondence analysis methods for synonymous codon usage in bacteria. *DNA Research: An International Journal for Rapid Publication of Reports on Genes and Genomes* 15, 357–365. Available at: <https://doi.org/10.1093/dnares/dsn028>.
- Tuller, T., Carmi, A., Vestsigian, K., *et al.*, 2010a. An evolutionarily conserved mechanism for controlling the efficiency of protein translation. *Cell* 141, 344–354. Available at: <https://doi.org/10.1016/j.cell.2010.03.031>.
- Tuller, T., Waldman, Y.Y., Kupiec, M., Ruppén, E., 2010b. Translation efficiency is determined by both codon bias and folding energy. *Proceedings of the National Academy of Sciences of the United States of America* 107, 3645–3650. Available at: <https://doi.org/10.1073/pnas.0909910107>.
- Tyson, H., Dhindsa, R., 1995. Codon usage in plant peroxidase genes. *DNA sequencing* 5, 339–351. Available at: <https://doi.org/10.3109/10425179509020865>.
- Vettrivel, U., Arunkumar, V., Dorairaj, S., 2007. ACUA: A software tool for automated codon usage analysis. *Bioinformation* 2, 62–63.
- Wright, F., 1990. The “effective number of codons” used in a gene. *Gene* 87, 23–29.
- Wu, G., Culley, D.E., Zhang, W., 2005. Predicted highly expressed genes in the genomes of *Streptomyces coelicolor* and *Streptomyces avermitilis* and the implications for their metabolism. *Microbiology* 151, 2175–2187. Available at: <https://doi.org/10.1099/mic.0.27833-0>.

Identification of Sequence Patterns, Motifs and Domains

Michael Gribskov, Purdue University, West Lafayette, IN, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

The development of protein and nucleic acid sequencing has revolutionized our understanding of macromolecular function. Sequence-based analyses are so powerful that it has been estimated that over 95% of our biochemical understanding of the protein function comes from such analyses. The power of sequence analysis is intimately tied to the evolutionary process. While the mutation process is not chemically random, it is random with respect to whether mutations occur in non-coding regions, genes, or regions that encode functionally important parts of proteins (such as active site residues). The process of selection is decidedly non-random; mutations that disrupt vital cellular processes, such as changes in transcriptional regulatory regions, structurally or functionally important parts of proteins, or epigenetic regulation are generally eliminated from the population (i.e., the organism bearing these mutation shows reduced fitness). The result is that structurally and functionally important sequence residues in proteins, the corresponding nucleotides that encode these regions in DNA, and important regulatory factor binding sites in DNA and RNA, all accumulate mutations at a slower rate than less critical regions. These regions thus appear *conserved*, and it is this conservation that is exploited by sequence-based methods for identifying patterns, motifs, and domains. Furthermore, conserved regions can often be directly associated with the specific molecular function that causes their conservation, making them extremely useful for predicting molecular properties from sequence.

In this article we will use the term *motif*, to cover all kinds of motifs, patterns, and domains, and focus on motifs that can be identified in macromolecular sequences. The definition of *motif* in computational biology is essentially the same as in common English usage – “A dominant or recurring idea in an artistic work (see “Relevant Websites section”)”. Sequence motifs can be found because they repeatedly occur, with limited sequence variation, many times in the same molecule or in different molecules. A critical sequence region that occurs only once is very difficult to define and associate with functional information.

Nucleic acid and protein motifs have very different properties. One of the most salient differences is due to the difference in the size of the nucleic acid and protein alphabets, four bases versus twenty amino acid residues. The difference in alphabet size means that, for any specific motif, the background of randomly matching (false positive) sequences will be higher in nucleic acids than in proteins. In addition, the pitch of the DNA helix, 10.4 bases/turn in A-form DNA, suggests that a protein interacting with the base-pairs in the major groove of the DNA can interrogate only five to six consecutive bases (assuming it does not wrap around the DNA); Not coincidentally, many transcription factor binding sites are about six bases long (or dimeric sites with two six base regions). The small alphabet and the short length of nucleic acid motifs makes the identification of such motifs challenging. Proteins, on the other hand have a larger alphabet, and the size of protein motifs is often larger. Structurally, conserved regions of proteins tend to lie in the folded core of the protein; typically a protein chain requires 6–12 residues to cross the protein core, where there are many constraints on acceptable mutations due to the very tight packing of the protein interior. Upon reaching the surface, the chain typically makes a bend or turn and reenters the protein core. The residues at turns and other surface exposed segments tend to show much weaker evolutionary constraints because they are free to move in the solvent. Surface residues also tend to be *hydrophilic* whereas core residues tend to be *hydrophobic*. This pattern of protein structure produces conserved regions that are much longer than in DNA, but which include a certain amount of internal length variation (commonly called *gaps*), due to the less conserved turn and surface regions. It is noteworthy that at surface turns, very large insertions of additional sequence often occur without disrupting the folded structure of the protein. The twenty letter protein alphabet and larger size of protein motifs makes it easier to confidently identify them in practice.

Transcription factor binding sites (TFBS) are the most important class of DNA motifs, due to their importance in gene regulation. Other DNA motifs of interest include transcription and translation start sites, splicing signals, terminator regions (although some of these actually function at the RNA level). Broader sequence patterns such as CG rich “islands”, patterns associated with 3D structure (bending and twisting of the helix), and sequence isochores, local DNA regions with different base composition, have also received attention. Motifs in RNA have received less attention in general, and can be either sequence-based, as in DNA, or related to the formation of internally base-paired structures.

In proteins, sequence motifs have been derived on many different scales ranging from the sites of post-translational modifications (single amino acid residues), active sites (several residues), to domains (dozens to hundreds of residues) and whole proteins (hundreds of residues).

Measuring Classification Quality

Pattern/motif/domain recognition is usually treated as a two-class supervised learning problem where the goal is to label each letter of a sequence, either nucleic acid or protein, as either belonging to a motif (positive), or belonging to a background model (negative). After classification, each letter in the sequence is considered to be true positive (TP; labeled as motif, part of motif), false negative (FN; labeled as background, part of motif), false positive (FP; labeled as motif, part of background), or true negative (TN; labeled as background, part of background), based on comparison to correctly labeled (gold-standard) training data. The success of the classification is measured in the same way as most machine learning methods, i.e., in terms of precision

($TP/TP + FP$) and recall ($TP/TP + FN$). The terms sensitivity, which is the same as recall, and specificity are also used. The definition of specificity is inconsistent in the literature, sometimes being defined as the same as precision, and sometimes defined as the correct negative fraction ($TN/TN + FN$). Precision and recall, since they are unambiguous, are therefore preferred. Usually there are many more negative examples in the training data than positive examples, for example, a gene region may contain only one TFBS (covering 10–12 bases), and tens of thousands of bases of non-TFBS bases. This *class imbalance* means that even methods with very high recall (sensitivity), may not work well in practice because the false positive rate is too high (i.e., low precision). Indeed, a method that only misclassifies 1/1000 of the negative bases, may easily generate far more false positives than true positives.

There is usually a tradeoff between precision and recall. For instance, one may achieve very high recall by simply predicting all bases as belonging to the motif. One may also achieve high precision by predicting only the strongest sites as belonging to a motif, but at the cost of misclassifying many true, but weaker, sites as non-motif. Approaches that consider both precision and recall in evaluating classification quality are preferred over simply looking at the recall since they cannot be “gamed” by tuning a method to be high recall/low precision. The most widely used method that considers both precision and recall is the Receiver Operating Characteristic, ROC (Zweig and Campbell, 1993; Gribskov and Robinson, 1996), and the area under the ROC curve, or AUC. This statistic has an attractive statistical interpretation; AUC is equivalent to a Wilcoxon test and is the probability that a random positive will score higher than a random negative. It is well known that ROC and AUC tend to perform poorly when there is a large degree of class imbalance; in this case, Mean Average Precision, MAP, is often preferred. Another commonly reported measure of classification quality is the F_1 score (also referred to as F-score, F-measure, accuracy, or Sørensen–Dice similarity coefficient), which is the harmonic mean of precision and recall ($F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$); F_1 considers false positives and false negatives to be equally costly, which is usually not true when class imbalance is large, and can be misleading in many practical situations.

The precision and recall seen on the gold standard training data often overestimates the efficacy of the classification due to *overfitting*. An unbiased estimate of classification quality therefore requires *cross validation*, a procedure in which a portion of the data is not used in training, and held out for use only in testing. It is important that the training and test data be independent; in the sequence motif context this requires that highly similar sequences (probable *homologs*) not occur in both training and test sets. Such sequences are usually identified by screening with sequence comparison programs such as Blast (Altschul *et al.*, 1990, 1997).

Models

When considered as a two-class problem, motif identification considers two feature distributions: the motif, or *foreground*, distribution and the non-motif or *background*, distribution. In a Bayesian context, the background model is the prior probability of the features, and the foreground model is the conditional probability of the features in the model class. The discrimination between these distributions can be measured in many ways, among the most common are log-odds ($\log(\text{foreground}/\text{background})$), and Bayesian posterior probability ($\frac{P_{\text{back}}P_{\text{for}}}{\sum P_{\text{back}}P_{\text{for}}}$). Log-odds statistics are particularly useful as the sum of the values over a series of sequence positions gives the relative probability that the sequence arises from the foreground or background model. The scoring matrices used in sequence alignment and database searching, such as the Dayhoff PAM (Dayhoff *et al.*, 1978) and the BLOSUM (Henikoff and Henikoff, 1992) matrices, are log-odds matrices allowing alignment scores to be interpreted in terms of the relative probability that the aligned sequences are homologous (foreground model) or unrelated (background model).

Background Model

The features are most commonly single amino acid residues, but may be physical-chemical characteristics, probabilities estimated by a prior analysis, or higher order combinations such as words (substrings). For protein sequences, the background model usually assumes that there is no positional correlation between amino acid residues, i.e., that each position is *iid* with respect to all others. While this assumption is known to be incorrect, there are in fact strong correlations between amino acid residues in proteins (in fact these are the basis of many secondary structure prediction methods), it often is sufficient in practice. Another option is to use the observed non-motif sequences to generate the background distribution as in the FastA (Pearson, 1996) sequence comparison program. For nucleic acids, it is well known that there are pronounced correlations between adjacent bases, most notably in the case of the avoidance of CG dinucleotides in eukaryotic sequences. The encoding of protein sequences imposes additional correlations. It is therefore common to use an order 3 to 5 Markov model for the background model. Use of such models is also common in *ab initio* gene modeling where exons, introns, and intergenic regions are separately modeled (most frequently with a 5th order Markov model).

Motif (foreground) Model

Consensus Sequence

Consensus sequence models date to the earliest days of sequencing and are still in use today. Consensus sequences represent a motif as a single sequence where only one base or amino acid residue is permitted at each position. Many restriction enzyme cut sites are effectively modeled in this way (consider, e.g., the *EcoRI* site: $\hat{G}AATC$). As more sequences became available, it was common to use consensus sequences to model promoters and other DNA signals by aligning examples of the signal, and using the

most common base at each position as the consensus sequence (Fig. 1). Such consensus sequences were often referred to as “boxes”, for instance the Pribnow Box (Pribnow, 1975) (now known as the -10 region of the *E. coli* promoter), TATAAT, CAAT (Graves *et al.*, 1986) box, TATA (Lifton *et al.*, 1978) box, and many others.

Because they are concise, consensus sequences are often shown in the literature to mark the position of a motif in a sequence. However, they are a very weak method for motif identification. First, they discard all information about the frequency of the letters at different positions of the motif. For instance, in the consensus sequence for the *E. coli* promoter, TATAAT, the final T is almost 100% conserved, indicating it has very high functional significance. On the other hand, the T at the third position, is not even present in a majority promoters – A, C, G, and T occur almost equally, indicating that this position is essentially irrelevant. Second, consensus sequences do not include a background model – overprediction of promoters in regions that are simply A/T rich is common – consensus sequences do not take into account that a sequence such as TATAAT occurs much more frequently at random in A/T rich regions. Third, consensus sequence do not generalize well. Only a fraction of known sequences precisely match the consensus sequence, and allowing mismatches or *gaps* greatly increases the number of false positive predictions, a problem that is exacerbated by the loss of information discussed, above.

Regular Expression

Regular expressions address some of the inadequacies of consensus sequences by admitting the possibility of several different letters matching at each position of the motif, and by allowing a limited number of variable spaces within the motif. The most important collection of regular expression patterns is found in the Prosite (Xenarios, 2013) database. Because of its high visibility, the Prosite syntax is probably the most widely used for protein sequences (Fig. 2), and is often used for nucleic acid sequences, as well. For the promoter sequence described above, we could write a regular expression such as [TC]-A-X-[AT]-{G}-T, and the combined -35 and -10 regions of the promoter could be written as T-T-[GT]-[CA]-{G}-[AT]-X(16,18)-[TC]-A-X-[AT]-{G}-T (compare to Fig. 3).

One of the advantages of regular expression representations is that efficient algorithms for identifying matches to regular expressions in text corpora are available in most programming languages, and in the UNIX operating system (*grep*). In principle, disregarding setup time, regular expression matching can be $O(n)$, where n is the length of the corpus.

In spite of their superiority to consensus sequence representations, regular expression representations suffer from similar problems. Although multiple characters can be allowed (or forbidden) at each position, there is no way to distinguish between letters that are allowed at very different frequencies (for instance the difference in the frequency of C and T in the first position of the -10 element in Fig. 3 (Oliphant and Struhl, 1988)). Regular expressions also do not incorporate a background model, and permitting incomplete matching typically gives rise to huge numbers of false positive matches. In spite of these drawbacks, the excellent compendium of functional motifs found in the Prosite database (ProDoc) has provided a fundamental resource for both protein annotation and validation of sequence matching algorithms.

ATATTATG	sequence 1
GTACTTTG	sequence 2
TCACAGTA	sequence 3
TTAGTCTC	sequence 4
CTAACTTC	sequence 5
TACTTT	consensus

Fig. 1 Consensus sequence. Five sequences from the *E. coli* -10 promoter region and the majority-rule consensus sequence. Bases matching the consensus are shown in bold. Note that no sequence in the training set exactly matches the consensus.

X	indicates a match to any letter
[]	enclose sets of allowed matching letters
{}	enclose sets of forbidden letters
(min,max)	indicates the preceding letter must occur a minimum of <i>min</i> times and a maximum of <i>max</i> times. (<i>min</i>) indicates a letter must occur <i>min</i> or more times.
-	separates sequence positions
<	marks the beginning of the sequence
>	marks the end position of the sequence

Fig. 2 Prosite regular expression (pattern) syntax.

A. -35										-10					
G	1	4	42	2	2	8	...	3	1	11	7	6	2	G	
C	0	4	3	15	32	11	...	15	2	14	8	24	1	C	
A	0	3	1	33	11	16	...	1	47	13	21	12	2	A	
T	57	47	12	8	13	23	...	33	2	14	16	10	47	T	

B.

15	16	17	18	19	spacing
2	18	59	26	5	count

Fig. 3 PSSM of *E. coli* Promoter elements. A. Numbers indicate counts of each base at each position. B. The numbers in the lower row indicate the counts of promoters with the indicated spacing (top row) between the -10 and -35 elements.

Regular expression matches are usually evaluated as simply a match or non-match, and the regular expressions in databases such as Prosite are manually tuned to provide high recall and precision. Calculating the exact probability of matching to regular expressions is complex (see [Sewell and Durbin \(1995\)](#)), but it is clear that allowing even a single unmatched position often dramatically increases the number of false positive matches, causing a great reduction in match precision.

Position Specific Scoring Matrix

PSSMs, also known as Positional Weight Matrices, PWM, and Position Specific Weight Matrices, PSWM, represent motifs as a vector of values for every possible character at each position of the motifs. In literature, PSSMs are often shown with the columns corresponding to positions and rows corresponding to the possible letters; computationally they are more often transposed so that the rows correspond to sequence positions and the columns to letters. The values in the PSSM can be of several kinds – among the most common are: simple counts, probabilities (i.e., each column sums to 1), or log-odds scores (log of the relative probabilities of the character in positive examples of the motif divided by the probabilities of the character in a background model). PSSMs retain all position independent information present in the training data, but do not incorporate information about correlated positions. PSSMs can be matched with sequences using dynamic programming alignment approaches [EPS30004], with complexity $O(nm)$ time, where n is the length of the database, and m is the number of positions in the motif PSSM. Empirical analysis suggests that the score distribution for PSSMs follows an extreme value distribution, similarly to local dynamic programming alignments ([Altschul and Gish, 1996](#)). The term profile (see next section) is sometimes used for PSSMs, although this usage is confusing.

Profile

Profile ([Gribskov et al., 1987](#)) and Profile Hidden Markov Models, PHMMs, are PSSMs that have additional information representing position-specific the weights/probabilities of insertion/deletion. For PHMMs, this additional information specifies the transition probabilities between the match states (which have the position specific probabilities of the letters) and to and from insert and delete states. Some kinds of profiles/HMMs contain additional information describing probabilities of insertion and deletions, resulting in additional columns. In order to keep the file human-readable, these motif representations may use more than one line per position in the sequence. Profiles and PHMMs can be matched using the same kind of dynamic programming algorithms used for PSSMs ([Wang and Dunbrack, 2004](#)), but it is customary to discuss PHMMs in terms of the Viterbi ([Viterbi, 1967](#); [Rabiner, 1989](#)) and Baum-Welch ([Baum et al., 1970](#)) algorithms.

Large molecular structures, such as protein domains, are most often defined using the profile approach because of the incorporation of positions specific indels in such models. In general, whole proteins and proteins domains may have insertions at any sequence position exposed on the surface of the protein – such large indels are seen in many families (see, for instance, for instance the ubiquitin specific protease family ([Ye et al., 2009](#))).

Sequence Logos

Examining and discerning the pattern of conservation in a PSSM or profile is difficult due to the large number of values. Sequence Logos ([Schneider and Stephens, 1990](#)) are a widely used graphical representation in which the letters at each position of the motif are displayed with different heights depending on their conservation. Conservation can be measured in a variety of ways such as letter frequency or information content ($H = -\sum_i p_i \log p_i$).

Learning Approaches

Multiple Alignment Based

Many motif learning approaches rely on multiple sequence alignments [EPS30005], MSAs, as a starting point. Several issues with MSAs affect one's ability to extract motif descriptions from MSAs.

One of the most important concerns is sampling. Even today, when thousands of complete genomes have been sequenced, the motif training data (i.e., the known sequences) is often very uneven with respect to the range of taxa from which sequences are drawn, and with respect to the presence of many close homologs (or even exact duplicate sequences). If not dealt with, these sampling issues tend to bias the motifs towards certain subclasses of sequences, and may compromise the ability of the learning algorithm to generalize. There are two main approaches to address sampling bias: selecting a less biased subset of sequences for training, or weighting sequences according to the amount of independent information they contain. Selecting a representative subset of sequences with less bias can be done manually, usually relying on taxonomic or other external information (for example, biochemical or structural properties of proteins), or automatically based on pairwise sequence comparisons. One of the most popular automatic methods is CD-HIT (Li *et al.*, 2001), see Fig. 4. Clearly this approach does not produce a completely unbiased representative set since the original distribution of sequences still has a strong effect.

Many methods have been successfully used for sequence weighting (Sibbald and Argos, 1990; Henikoff and Henikoff, 1994; Thompson *et al.*, 1994a,b). Many approaches rely on a distance-based tree [EPS10149], calculated using alignment scores from pairwise alignments or the MSA. The method introduced by Thompson *et al.* (1994a,b) in the Clustal V multiple alignment program, and Henikoff and Henikoff's position-based sequence weights (Henikoff and Henikoff, 1994), which do not require a tree, have been widely used (Fig. 5).

Position specific weights are first calculated for the letters at each aligned position in each sequence. The weight is

$$w = \sum_{p=1}^{LENGTH} \sum_i^{alphabet} \frac{1}{k_p n_{i,p}}$$

where k_p is the number of distinct letters at position p , and $n_{i,p}$ is the number of occurrences of letter i at position p

While the use of weighting to account for sampling bias is widely accepted, it does not appear that the weights need to be highly accurate, probably due to the high degree of biological variation in evolutionary rates.

In addition to bias, sequence training sets may also suffer from undersampling – the total number of sequences may be small, and letters/words that occur at very low frequencies are not observed. This problem is often treated by replacing the observed letter counts with a modified count that includes a pseudocount or prior. One common adjustment, commonly called a “plus one prior” is to add one to every letter count. Unless the total counts are very large, such a pseudocount is probably an overcorrection. Another common approach is to add one letter count distributed according to the overall letter distribution (i.e., the pseudocount sums to one across all letters).

Another concern with MSAs is the presence of errors in the alignment, especially near to insertions and deletions. In the case of regular expression representations, misalignment can be a severe problem and causes serious degradation of both precision and recall. The misalignment issue is exacerbated by the inclusion of sequences with errors, very distantly related sequences, or even sequences that are unrelated, in the training set. The latter problem is more frequent than one might think because training sequences are often selected by noisy biological criteria (such as co-expression) – the presence of incorrectly labeled training examples is therefore very common in biological contexts.

Consensus sequences and regular expression motif descriptions usually focus on short, highly conserved sequences. Often these are selected manually (see “Relevant Websites section”), based on an initial multiple alignment and/or expert knowledge, followed by

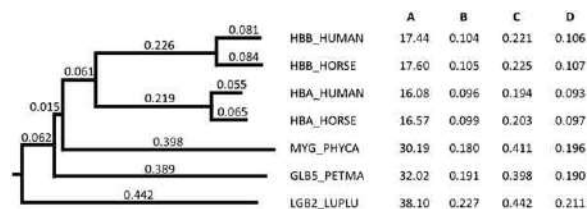


Fig. 4 Sequence weights. The tree shows a group of globin proteins (Thompson *et al.*). A. position-based weights, un-normalized. B. position-based weights, normalized to sum to 1. C. Clustal tree-based weights, un-normalized. D. Tree-based weights normalized to sum to 1.

1. Measure similarities between sequences using Blast or a dynamic programming alignment
2. For each sequence, determine the number of neighbors. Sequences are neighbors if their alignment score is higher than a user selected threshold
3. Find the available sequence with the most neighbors and add it to the representative set.
4. Mark all of its neighbors as unavailable
5. Repeat 3-4 until no sequences remain available

Fig. 5 CD-Hit method of selecting unbiased representative sequences.

iterative database searching to validate that the pattern separates positive and negative examples. A problem with this manual approach is that it may not generalize well because only letters that occur at a position, in the training data, are included in the pattern. A new sequence that contains a chemically similar but novel amino acid residue may therefore not be recognized. In addition, as true positive examples accumulate, with continuing growth in the number of sequences, one typically finds that an increasing number of letters are allowed at each position leading to a concomitant increase in false positive matches (i.e., lower precision). This problem is typically solved by splitting the pattern into two more specific patterns that each recognize a subset of the training examples.

Generalization, the ability to detect examples of a motif that do not precisely match any of the training examples is one of the primary goals of motif identification. Generalization is particularly difficult when the number of training examples is small, or drawn from a limited taxonomic range so that the sequence diversity is low. A common approach to this is to use external knowledge about the chemical/mutational similarity of the amino acid residues to augment the training observations. This approach has been called *regularization* (Karplus, 1995), although it differs significantly from the L_1 and L_2 regularization approaches typically used in machine learning to restrict the complexity of the model parameterization [EPS30012]. In the case of regular expression patterns, regularization often augments the set of allowed letters observed at a position with additional letters that are chemically similar, but were not observed. For instance, it is common to predefine classes of amino acid residues such as hydrophobic = [ILMV], basic = [HKR], aromatic = [FWY], acid/amide = [DNEQ], hydrophilic = [HKRDNEQSTWY], etc. A column in a multiple sequence that contains, for instance, Isoleucine (I), Methionine (M) and Valine (V), might be defined as belonging to the hydrophobic class [ILMV], even though Leucine (L) was not observed. A more sophisticated approach, incorporated in the EMotif program (Huang and Brutlag, 2001; Nevill-Manning *et al.*, 1998) matches multiple overlapping classes to the observed set of letters and chooses the best matching class. These approaches are regularizations in the sense that they reduce the original 5×10^{18} possible character classes to around 10 classes.

For PSSMs, mixture models based on empirical observations of amino acid substitutions are commonly used. Fitting the observed data to a Dirichlet mixture distribution is a common approach. A mixture distribution is a weighted combination of simpler base models, in this case the base models are Dirichlet distributions. The probability of the “true” letter distribution at a position $P(\vec{x})$ is given by

$$P(\vec{x}) = \frac{\sum_j w_j P_j(\vec{x})}{\sum_j w_j}$$

where the $P_j(\vec{x})$ are the probabilities predicted by the base models and w_j are weights that indicate the importance of the model in the mixture. $P_j(\vec{x})$ is a Dirichlet distribution.

$$P(\vec{x}) = \frac{\sum_{i=1}^{20} p_i^{x_i-1}}{Z}$$

where the p_i and w_j are empirically determined based on known “good” multiple alignments drawn from the BLOCKS database, HSSP, or expert protein family multiple alignments (Brown *et al.*, 1993; Sjölander *et al.*, 1996).

For a vector of observed letter counts, $\vec{n}$, and M base Dirichlet distributions, the estimated true frequency of letter i , $\hat{P}(x_i)$

$$\hat{P}(x_i) = \sum_{j=1}^k P(\vec{\alpha}_j | \vec{n}) \frac{n_i + \alpha_{ij}}{|\vec{n}| + |\vec{\alpha}_j|} = \sum_{j=1}^M \left(\frac{w_j P(\vec{n} | \alpha_j)}{\sum_k w_k P(\vec{n} | \alpha_k)} \right) \frac{n_i + \alpha_{ij}}{|\vec{n}| + |\vec{\alpha}_j|}$$

Another approach to learning a PSSM is the evolutionary profile (Gribskov and Veretnik, 1996), in which the Dayhoff PAM evolutionary model is used as a source of known biological patterns of residue substitution. Briefly, for each column in the alignment, the probability of the observed letter frequencies expected for each possible ancestral residue, at different evolutionary distance from 1 PAM to 1024 PAM, are used to choose the components of a mixture model, weighted by the probabilities. The mixture probabilities are then used to form log-odds scores using expected random frequencies of the letters as the background. This approach has also been generalized to use a continuous estimate of the evolutionary distance rather than a fixed set of PAM matrices (Eskin *et al.*, 2001).

In some cases, either due to a poor multiple sequence alignment, or because of biological or functional divergence within the set of training sequences, an aligned position may represent two or more distinct patterns of conserved letters. When this occurs, it has been argued that is desirable to replace the actual observed letter frequencies (or pseudocount) with the probabilities that are dominated by the probabilities derived from a known, biologically relevant, prior distribution. This approach was termed the megaprior heuristic (Bailey and Gribskov, 1996), and has been implemented in MEME, evolutionary profiles, and HMMs.

Non-Alignment Approaches

Protein motifs are most commonly extracted from an initial multiple sequence alignment, but sometimes the training sequences are not strictly homologous, or the sequences contain repeated sequences, rearrangements, or other common situations that disrupt alignment approaches. For nucleic acid sequences, it is usually difficult or impossible to construct an accurate multiple alignment due to the small nucleic acid alphabet, four letters, and the rapid evolution of non-coding regions (such as promoter or enhancer regions in which one might want to find transcription factor binding sites). In these cases, it is necessary to use a learning approach that simultaneously learns both the motif locations and the motif pattern itself. The most widely used approaches, expectation maximization and Gibbs sampling, learn a PSSM motif, without gaps, based on a set of training sequences.

Expectation-Maximization is a general purpose optimization approach that learns a locally optimal pattern that maximizes the probability of the data given the pattern, i.e., the likelihood. With sequences, the pattern is a PSSM giving the probability of each letter at each position of the motif for a pre-determined motif length. EM, as the name implies, is an iterative two-stage algorithm. In the first step, given the current pattern, the probability that each position of each sequence was generated by the pattern is calculated. In the second, the pattern is re-estimated based on the probabilities determined in the first step. The probability weighted frequencies of the letters at each position of the PSSM represent the maximum likelihood estimate of the pattern. The updated pattern is then used to recalculate the probabilities that each position is a site, and the two step process iterated to convergence. The total likelihood is guaranteed to remain equal or increase at each iteration so this can be recognized as a greedy, local hill-climbing, process. EM has a number of advantages: it needs little or no prior information about the pattern to be found (except its length), it typically converges rapidly, and it is adaptable to many kinds of patterns including repeated patterns and patterns that are not present in every sequence. It also has disadvantages: it is a locally optimal method, it is sensitive to initial conditions, and the width of the pattern must typically be known in advance. Typically EM approaches overcome the issues of local optimality and sensitivity to initial conditions by repeating the training multiple times using randomized initial conditions each time. Patterns with the highest total likelihood, or that are found repeatedly, are assumed to be the globally optimal solution (although obviously one cannot be sure).

The MEME program (Bailey and Elkan, 1994) incorporates several clever enhancements that improve the performance of this general EM approach with biological sequence motifs. The first insight is that the desired conserved motif must be present in the training set. Rather than using randomized initial PSSMs, MEME comprehensively samples sequences from the training set and creates the initial PSSMs by mixing the sequence with the background frequency distribution. Then several cycles of EM are performed and the partially trained PSSMs are inspected to identify which ones are converging, taking advantage of the rapid convergence of EM; only these PSSMs are iterated to convergence. The second issue has to do with the motif length. Since this is an input parameter for the EM, the only option is to sample a large number of lengths and to choose the length that maximizes the likelihood (including a correction for model complexity). Finally, especially when considering gap-free motifs, most training sets contain multiple motifs – for instance, two conserved motifs with a variable spacing, such as the *E. coli* promoter, cannot be learned as a single PSSM. To solve this problem, MEME iteratively identifies multiple motifs, and uses probabilistic erasing to remove each motif from the training data after it is discovered. In probabilistic erasing, after the PSSM has converged, the probability, $P(S_{ij}/\text{PSSM})$, that each position, j , of each of the training sequences, S_i , was generated by the motif is evaluated, and the weight of that position is decreased to $1 - P(S_{ij}/\text{PSSM})$.

Another method for learning both the motif and the motif locations is the Gibbs sampler (Lawrence *et al.*, 1993) (GS), which uses a Markov chain Monte Carlo (MCMC) sampling approach. GS can be looked on as a stochastic version of EM. As with EM, a motif width, W , is defined. Each width W word in each sequence is a possible location of the motif, and the probability of each of these positions is calculated identically to the E-step in EM. The probability of generating each word assuming some background distribution, for instance the random expected frequencies of the bases or amino acids, is also calculated. The ratio of these probabilities can be considered to be weights on each possible width- W word. The initial PSSM is constructed by randomly selecting words according to these weights, and at each step, one word is randomly replaced, the probabilities and weights recalculated. This is very similar to the process used in EM, but rather than using the probabilities of all words across the sequences, only a specific group of sampled positions is used. GS is typically slow to converge, often requiring thousands of cycles for even modest numbers and lengths of sequences. Extensions to the original GS procedure that allow multiple motifs to be learned, and that allow the motif width to be optimized have been developed (Neuwald *et al.*, 1995).

Because of the small nucleotide alphabet, motif finding in nucleic acids often incorporates the identification of kmer words that are statistically overrepresented in a training set, where k is typically in the range of 5 to 8 bases. These approaches are widely used, for instance, in defining promoter elements and transcription factor binding sites (TFBS). Because all words are evaluated, word based methods have the theoretical advantage of discovering a globally optimal pattern rather than the locally optimal patterns found by EM and GS. In practice, the advantages are less clear due to the pronounced word preferences of nucleic acid sequences, and the weak conservation of many DNA motifs. Because of the relatively strong word-preference of DNA sequences, even in non-conserved regions, it is critical to include a background model; typically a 5th or 6th order Markov model is used (Sinha and Tompa, 2002). An important issue with word probability based methods is that any ambiguity in the motif, such as a position which is not highly conserved, splits the occurrences of the motif into multiple words. Clearly, this weakens the distinction between the conserved words and the background, reducing the chance that the motif will be detected as significantly overrepresented. In principle one can identify words that represent ambiguous images of a conserved site (Pavesi *et al.*, 2004) but it remains difficult to distinguish such ambiguous patterns. Because of their short length, and the small alphabet, it is common to try to use additional information to constrain the patterns that are identified. The most common constraint has been to require an internal dyad symmetry (van Helden *et al.*, 2000), since many DNA binding proteins possess such symmetry. Most word-based approaches, similarly to PSSM-based approaches, also assume that the positions of a motif are independent which is not necessarily correct.

Validation

Assessment of the classification ability of a program has been one of the most difficult areas in both computational and systems biology. Gold standard data are, themselves, often calculated by a computational learning method. A new method that is better, appears to make many false positive label assignments when compared to an older inferior standard. Still, just assigning more class

labels, or predicting more positives, measures only the recall of a method and says nothing about precision. As discussed above, this leads to serious overestimation of the usefulness of a method. With protein motifs it has been common to use two kinds of standards. The first approach relies on the documentation associated with the PROSITE database, much of which is carefully hand curated. Although PROSITE signatures are based on these data, the data include annotations of false negatives, that is, sequences that should be positive but did not match the PROSITE signature of pattern. This inclusion of both the easy positives (that can be identified with regular expressions), and the difficult positives, which cannot, has made the PROSITE annotation very useful. Another approach is to use the three-dimensional structure databases CATH (Dawson *et al.*, 2017) and SCOP (Fox *et al.*, 2014) to define groups of related proteins. While it is still not clear that all proteins with similar three-dimensional structures are actually related, or whether some protein folds are somehow just energetically favorable, a classification method with the highest recall/precision vs the structural classification is still likely to be the best. For nucleic acid motifs, validation vs gold standard data is more problematic. While a number of compendia of promoter sites and TFBS exist, the counts for particular motifs are often low, and the motifs themselves have often been identified by the multiple alignment or overrepresented word approaches described above. In a few cases, laboratory experiments have been conducted to find short oligonucleotides that physically bind to specific proteins, or that can be pulled down by their bind (as in ChIP-Seq). These datasets are still somewhat difficult to interpret since the exact binding site is usually not determined, and because most sequence-specific nucleic acid binding-proteins have also have high non-sequence specific binding, as well.

See also: Biological Database Searching. DNA Barcoding: Bioinformatics Workflows for Beginners. Functional Enrichment Analysis. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition

References

- Altschul, S.F., Gish, W., 1996. Local alignment statistics. *Methods Enzymol.* 266, 460–480.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *J. Mol. Biol.* 5 (215), 403–410.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.
- Bailey, T.L., Elkan, C., 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In: *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pp. 28–36. AAAI Press.
- Bailey, T.L., Gribskov, M., 1996. The megaprior heuristic for discovering protein sequence patterns. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 4, 15–24.
- Baum, L., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.* 41, 164–171.
- Brown, M., Hughey, R., Krogh, A., *et al.*, 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 1, 47–55.
- Dawson, N.L., Lewis, T.E., Das, S., *et al.*, 2017. CATH: An expanded resource to predict protein function through structure and sequence. *Nucleic Acids Res.* 45, D289–D295.
- Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model of evolutionary change in proteins. In: Dayhoff, M.O. (Ed.), *Atlas of Protein Sequence and Structure*, vol. 5(3). Washington, DC: National Biomedical Research Foundation, pp. 345–352.
- Eskin, E., Grundy, W.N., Singer, Y., 2001. Using mixtures of common ancestors for estimating the probabilities of discrete events in biological sequences. *Bioinformatics* 17 (Suppl. 1), S65–S73.
- Fox, N.K., Brenner, S.E., Chandonia, J.M., 2014. SCOPe: Structural Classification of Proteins—Extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.* 42, D304–D309.
- Graves, B.J., Johnson, P.F., McKnight, S.L., 1986. Homologous recognition of a promoter domain common to the MSV LTR and the HSV tk gene. *Cell* 44, 565–576.
- Gribskov, M., McLachlan, A.D., Eisenberg, D., 1987. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA* 84, 4355–4358.
- Gribskov, M., Robinson, N.L., 1996. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.* 20, 25–33.
- Gribskov, M., Veretnik, S., 1996. Identification of sequence patterns with profile analysis. *Methods Enzymol.* 266, 198–212.
- Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* 89, 10915–10919.
- Henikoff, S., Henikoff, J.G., 1994. Position-based sequence weights. *J. Mol. Biol.* 243, 574–578.
- Huang, J.Y., Brutlag, D.L., 2001. The EMOTIF database. *Nucleic Acids Res.* 29, 202–204.
- Karplus, K., 1995. Evaluating regularizers for estimating distributions of amino acids. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 3, 188–196.
- Lawrence, C.E., Altschul, S.F., Boguski, M.S., *et al.*, 1993. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* 262, 208–214.
- Lifton, R.P., Goldberg, M.L., Karp, R.W., Hogness, D.S., 1978. The organization of the histone genes in *Drosophila melanogaster*: Functional and evolutionary implications. *Cold Spring Harb. Symp. Quant. Biol.* 42, 1047–1051.
- Li, W., Jaroszewski, L., Godzik, A., 2001. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* 17, 282–283.
- Neuwald, A.F., Liu, J.S., Lawrence, C.E., 1995. Gibbs motif sampling: Detection of bacterial outer membrane protein repeats. *Protein Sci.* 4, 1618–1632.
- Nevill-Manning, C.G., Wu, T.D., Brutlag, D.L., 1998. Highly specific protein sequence motifs for genome analysis. *Proc. Natl. Acad. Sci. USA* 95, 5865–5871.
- Olipphant, A.R., Struhl, K., 1988. Defining the consensus sequences of *E. coli* promoter elements by random selection. *Nucleic Acids Res.* 16, 7673–7683.
- Pavesi, G., Mereghetti, P., Mauri, G., Pesole, G., 2004. Weeder Web: Discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res.* 32, W199–W203.
- Pearson, W.R., 1996. Effective protein sequence comparison. *Methods Enzymol.* 266, 227–258.
- Pribnow, D., 1975. Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. *Proc. Natl. Acad. Sci. USA* 72, 784–788.
- Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 257–286.
- Schneider, T.D., Stephens, R.M., 1990. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.* 18, 6097–6100.
- Sewell, R.F., Durbin, R., 1995. Method for calculation of probability of matching a bounded regular expression in a random data string. *J. Comput. Biol.* 2, 25–31.
- Sibbald, P.R., Argos, P., 1990. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *J. Mol. Biol.* 216, 813–818.
- Sinha, S., Tompa, M., 2002. Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.* 30, 5549–5560.

- Sjölander, K., Karplus, K., Brown, M., *et al.*, 1996. Dirichlet mixtures: A method for improved detection of weak but significant protein sequence homology. *Comput Appl. Biosci.* 12, 327–345.
- Thompson, J.D., Higgins, D.G., Gibson, T.J., 1994a. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Comput. Appl. Biosci.* 10, 19–29.
- Thompson, J.D., Higgins, D.G., Gibson, T.J., 1994b. CLUSTAL W – Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* 22, 4673–4680.
- van Helden, J., Rios, A.F., Collado-Vides, J., 2000. Discovering regulatory elements in noncoding sequences by analysis of spaced dyads. *Nucleic Acids Res.* 28, 1808–1818.
- Viterbi, A.J., 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* 13, 260–269.
- Wang, G., Dunbrack, R.L., 2004. Scoring profile-to-profile sequence alignments. *Protein Sci.* 13, 1612–1626.
- Xenarios, I., 2013. New and continuing developments at PROSITE. *Nucleic Acids Res.* 41 (Database issue), D344–D347.
- Ye, Y., Scheel, H., Hofmann, K., Komander, D., 2009. Dissection of USP catalytic domains reveals five common insertion points. *Mol. Biosyst.* 5, 1797–1808.
- Zweig, M.H., Campbell, G., 1993. Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine. *Clin Chem.* 39, 561–577.

Relevant Websites

<https://en.oxforddictionaries.com/definition/motif>
motif.

<http://prosite.expasy.org/prosuser.html>
PROSITE user manual ExPASy.

Epigenetics: Analysis of Cytosine Modifications at Single Base Resolution

Malwina Prater and Russell S Hamilton, University of Cambridge, Cambridge, United Kingdom

© 2019 Elsevier Inc. All rights reserved.

Introduction

All DNA nucleotides can undergo covalent modifications *in vivo*, with 5-methylcytosine (5mC) being the most investigated and understood of the modifications. In eukaryotes, 5mC is most commonly methylated in a CpG context, however it can also be modified in non-CpG contexts (CHH and CHG methylation) (Guo *et al.*, 2014). Other known DNA nucleotide modifications include adenine that can be converted to N6-methyladenine (6mA), and thymine to 5-hydroxymethyluracil (5hmU) (Sood *et al.*, 2016).

With the advent of next generation sequencing, it is possible to interrogate DNA modifications on a genomic scale and at single base resolution. In this article, we outline methods for interrogating cytosine modifications with particular emphasis on the important and emerging interest in 5-hydroxymethylcytosine (5hmC), 5-formylcytosine (5fC) and 5-carboxylcytosine (5caC) and the downstream bioinformatics analysis.

Some of the existing methylation data including 5mC, 5hmC and 5caC generated by large epigenetic-focused consortia can be viewed and downloaded. Among these, BLUEPRINT is a key player in sharing DNA methylation data, providing a database to browse and download raw data, analysis files or explore data on the BLUEPRINT Data Analysis Portal. Genome tracks are also provided for visualizing the data in UCSC or Ensembl. Other epigenetic data consortia that can provide DNA methylation data (mostly WGBS-seq) include NIH Roadmap, ENCODE and International Human Epigenome Consortium (IHEC) (Adams *et al.*, 2012; Bernstein *et al.*, 2010; Consortium *et al.*, 2012). DeepBlue Epigenomic Data Server enables centralized access to these and more databases (Albrecht *et al.*, 2017).

It is worth noting that nucleotide modifications are also found in RNA molecules (Frye *et al.*, 2016). Many of the analyses outlined in this article will be applicable to RNA, however we focus our attention to the cytosine modifications in DNA.

Cytosine Modifications

The covalent modifications of cytosine of greatest biological interest are to the 5' position. These modifications are of particular interest as they have been shown to regulate development and disease, and more recently aging. Correlative DNA methylation levels at specific loci proved to be valuable tool for predicting the speed of aging in human and mouse (Horvath, 2013; Stubbs *et al.*, 2017). The concept of epigenetic clock was taken a step further into development of commercial test, that is now available to individuals (see Section Relevant Websites).

Starting from unmodified cytosine, a series of enzyme driven reactions lead to 4 distinct modifications and can be returned to an unmodified cytosine (Fig. 1). 5mC is established *de novo* by DNA methyltransferase enzymes (DNMTs): DNMT3a, DNMT3b and DNMT3L (Goll and Bestor, 2005), and its maintenance is controlled by DNMT1 (Delgado-Olguín *et al.*, 2012; Liang *et al.*, 2002). 5mC can be reversed to unmodified C via two alternative mechanisms: passive demethylation as a consequence of lack of maintenance as the cells replicate; or active demethylation, which involves oxidative DNA demethylation mediated by ten-eleven translocation (TET) family of enzymes (Kohli and Zhang, 2013). TET enzymes (5mC hydroxylases) can catalyze the stepwise oxidation of 5mC to 5hmC, 5fC and 5caC (He *et al.*, 2011; Ito *et al.*, 2011; Tahiliani *et al.*, 2009). The conversion of 5fC and 5caC to unmethylated C is thought to be facilitated by excision of modified cytosine by thymine DNA glycosylase (TDG), followed by base excision repair mechanisms (BER) (Weber *et al.*, 2016).

5-methylcytosine: 5mC

The addition of methyl group to the 5' position of cytosine in 5mC is stable and has been shown to play important role in development, tissue specificity, aging and disease. The majority of methylation in mammals occurs in a CpG context, which are distributed non-randomly (Guo *et al.*, 2013, 2014). Most of the mammalian genomes are methylated (Li and Zhang, 2014), but CpG island regions (CGIs) with high density of CpGs usually remain unmethylated and are predominantly associated with gene promoters. Methylation of cytosine is associated with silenced chromatin, but the role of DNA methylation depends on its location within the transcriptional unit. In general, highly methylated DNA in close proximity to the transcription start site is associated with long-term silencing of genes (Raynal *et al.*, 2012). In contrast, methylation of the gene body is associated with transcriptional activity and alternative splicing (Ball *et al.*, 2009; Hahn *et al.*, 2011). Interestingly, 5mC can inhibit binding of methylation-sensitive transcription factors to their target sites, and also promote binding of others, especially those involved in embryonic and organism development (Yin *et al.*, 2017). A further biological role of 5mC is the regulation of imprinted genes (Paulsen and Ferguson-Smith, 2001). The 5mC is also thought to have protective role for genome stability and to act as a defense mechanism against transposons (Bourc'his and Bestor, 2004; Lavie *et al.*, 2005; Turelli *et al.*, 2014; Walsh *et al.*, 1998). It is also clear that 5mC and histone modifications crosstalk, but in mammals this relationship is complex and interdependent (Du *et al.*, 2015; Li and Zhang, 2014).

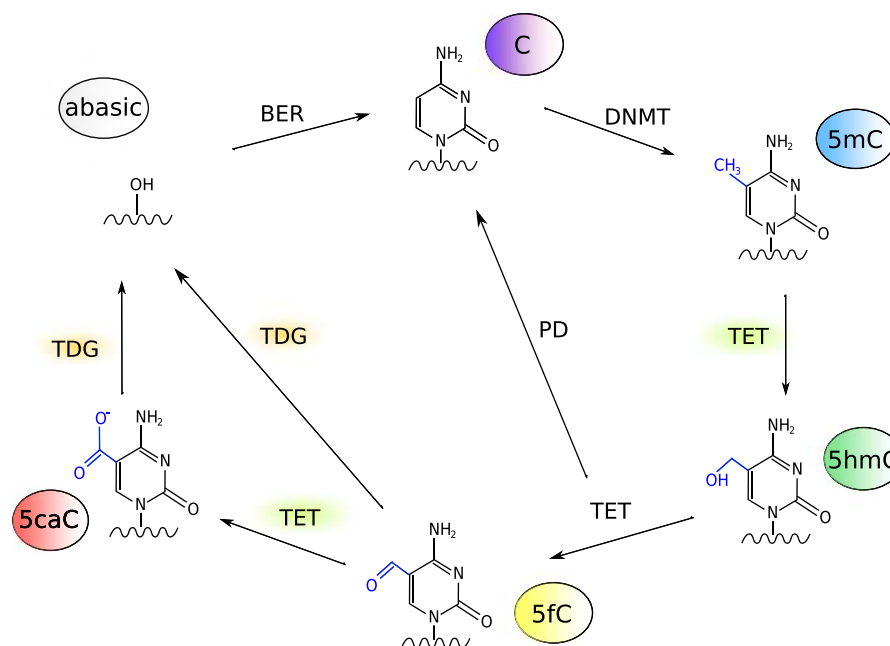


Fig. 1 Cytosine can be methylated at the 5th position by DNA methyltransferases: Dnmt1 as a maintenance process, or *de novo* by Dnmt3a or Dnmt3b. Resulting 5mC bases can be iteratively oxidised by TET enzymes in the process of active demethylation, resulting in 5hmC, 5fC and finally 5caC products. In the process of passive dilution (PD), 5hmC (and similarly 5fC and 5caC, not shown) can be demethylated to C during the course of DNA replication. 5caC, and also 5fC, can be deaminated to thymine and excised by thymine DNA glycosylase (TDG). The abasic site can be filled with unmodified deoxycytidine triphosphate (dCTP) by the base excision machinery (BER).

5-hydroxymethylcytosine: 5hmC

Recent years have brought evidence that 5hmC is not only an oxidation product of 5mC, but is chemically stable, with a distinct distribution in the genome, forming a stable epigenetic mark in its own right (Bachman *et al.*, 2014). In Embryonic Stem Cells (ESC) cells, 5hmC is enriched at differentiation-associated enhancers (Madzo *et al.*, 2014; Sérandour *et al.*, 2012; Stroud *et al.*, 2011; Szulwach *et al.*, 2011) and bivalent promoters of genes repressed at ESC cells, but activated upon differentiation (Wu *et al.*, 2011). 5hmC is also hypothesized to have a role in protection of genome integrity by marking sites for DNA damage (Kafer *et al.*, 2016). Levels of 5hmC are very tissue specific, with highest prevalence in central nervous system (brain). The differences in 5hmC levels within brain itself are pronounced enough to distinguish between specific brain regions (Lunnon *et al.*, 2016; Spiers and Hannon, 2017).

5-formylcytosine: 5fC and 5-carboxylcytosine: 5caC

The functions of 5fC and 5caC are still unclear, but recent research hints that they may be more than mere oxidation products. In mouse ESCs, 5fC is enriched in enhancer regions marked by H3K4me1 and H3K27ac (Iurlaro *et al.*, 2016). Furthermore, 5fC and 5caC have been shown to affect the rate and substrate specificity of RNA polymerase II processivity (Kellinger *et al.*, 2012; Wang *et al.*, 2015). Chromatin modifiers and transcriptional regulators have been identified as proteins binding to 5fC, suggesting a potential role of 5fC in gene regulation (Iurlaro *et al.*, 2013). Although the role of 5caC still needs to be elucidated, it has been hypothesized to be involved in regulating alternative splicing via CTCF preferential binding to 5caC (Marina and Oberdoerffer, 2016; Spruijt *et al.*, 2013).

Experimental Methods to Assay DNA Modifications in Genome

Bisulfite conversion alone can no longer claim to be a gold standard in 5mC detection as it co-detects 5mC and 5hmC modifications, leaving a confounded signal. To address this limitation of bisulfite-only conversion, several methods have been developed to distinguish between the distinct cytosine modifications (with focus on 5mC and 5hmC). A recent review compares a selection of 5hmC specific profiling techniques (Skvortsova *et al.*, 2017), in this article we focus on techniques compatible with 5hmC, 5fC and 5caC profiling (Table 1).

Bisulfite and oxidative bisulfite sequencing is able to discriminate between 5mC and 5hmC. In standard bisulfite conversion (BS), both 5mC and 5hmC are protected and are read in sequencing as C. The oxidant *potassium perruthenate* is used to convert 5hmC to 5fC, which is read as T in sequencing after BS conversion (same as unmethylated C, 5fC and 5caC) (oxBS). 5hmC levels

Table 1 Comparison of methods for assaying cytosine modifications using array or next-generation sequencing platforms

Description	Detection of										Modified base sequenced as					Depth of seq required	Limitations	Comments/Strengths					
	5mC					5hmC					5fC								5caC				
	Y	Y	Y	Y*	N	T	C	BS: C, oxBS: T	T	T	Y	low	850k	N	oxBS-redBS does not differentiate between 5caC and unmodified C; human only				cost-effective				
EPIC array	Hybridisation of BS/oxBS DNA to arrays	Y	Y	Y*	N	T	C	BS: C, oxBS: T	T	T	Y	low	850k	N	oxBS-redBS does not differentiate between 5caC and unmodified C; human only	cost-effective							
RRBS-seq/RRHP	MspI digestion, Library prep amplification, NGS	Y	Y	Y*	N	T	C	BS: C, oxBS: T	T	T	Y	mid	~12 M	Y	oxBS-redBS does not differentiate between 5caC and unmodified C; Position limited by proximity to enzymes' recognition sites	cost-effective							
WG	Evaluates methylation levels of almost all CpGs	Y	Y	Y*	N	T	C	BS: C, oxBS: T	T	T	Y	high	28 M	Y	oxBS-redBS does not differentiate between 5caC and unmodified C; high cost	Compatible with single cell as input (not for OxBS due to coverage limitations)							
Targeted	Enrichment/library prep -> NGS	Y	Y	Y*	N	T	C	BS: C, oxBS: T	T	T	Y	mid	Kit defined	Y	oxBS-redBS does not differentiate between 5caC and unmodified C	cost-effective							
MeDIP-seq	Anti - 5(h)mC antibody to methylated CpG sites followed by NGS	Y	Y	Y	Y	-	-	-	-	-	N	mid		Y	Biased toward hypermethylated regions, not suitable for very low 5hmC, 100 bp resolution; Limited by antibody specificity	bisulfite-free							
TAB-Seq	5hmC labelling with b-GT, TET oxidation, BS conversion, NGS	Y	Y	N	N	T	BS: C, TAB: T	C	T	T	N	high		Y	Accuracy limited by efficiency of Tet enzyme	limitations of using enzymes							
PacBio SMRT	Direct detection of modified bases by assessing activity of DNA polymerase	Y	Y	Y	Y	C	5mC	5hmC	5fC	5caC	N	high		Y	High cost, throughput limitations	bisulfite-free; no DNA amplification, long reads, multiple modifications detected							
Nanopore	Direct detection of modified bases via ion current as passes through pore	Y	Y	Y	Y	C	5mC	5hmC	5fC	5caC	N	high		Y	High error rate, processivity/throughput, cost	bisulfite-free; long reads, multiple modifications detected							

*Indicates with redBS.

can be quantified by subtracting BS from oxBS signals for each cytosine position (Booth *et al.*, 2012). An associated method to map 5fC is redBS-seq, which utilises chemical reduction of 5fC to 5hmC using *sodium borohydride* (Booth *et al.*, 2014). As a result, 5fC is read as C after BS conversion (same as 5hmC and 5mC). Thus, 5fC levels can be quantified by subtracting redBS from the BS signal. Whole genome sequencing of oxBS and redBS provide the most accurate and least biased 5hmC distribution across the genome, but the associated sequencing costs are very high due to high coverage requirements. OxBS and redBS treatments are also compatible with arrays (e.g., Illumina 450K and EPIC arrays) or in combination with Reduced Representation Bisulfite Sequencing (RRBS) and enrichment methods (e.g., Roche Nimblegen SeqCap) (See [Table 1](#)) (Gross *et al.*, 2016; Spiers and Hannon, 2017; Stewart *et al.*, 2015). The enrichment methods provide a more cost-effective way of assaying methylation levels than for whole genome. BS/oxBS sequencing, however, is not the only method to interrogate 5hmC distribution at a genome scale. TAB-seq, detects 5hmC directly by converting 5hmC to 5-glucosylmethylcytosine with β -glucosyltransferase first (read as C), followed by oxidation of 5mC and 5fC to 5caC by TET enzymes, and BS-seq (Yu *et al.*, 2012a,b). A subtraction of the 5hmC from the BS signal is required to call the 5mC. Several methods to map 5fC and 5caC have been proposed, including *M.SssI* methylase-assisted BS-seq (MAB-seq and caMAB-seq) and chemical-modification-assisted bisulfite sequencing (fCAB-seq and caCAB-seq). Single base resolution methods are mainly bisulfite or enzyme conversion based, but recently novel technologies with the ability to directly detect nucleotide modifications have been developed (e.g. Pacific Biosciences SMRT and Oxford Nanopore) (Rand *et al.*, 2017; Simpson *et al.*, 2017; Song *et al.*, 2012a). These methods currently do not have the per cytosine accuracy of, for example, bs/oxBS, however are likely to be the methods of choice in the future and are introduced further in the Discussion. Although bisulfite-based methods to identify and quantify DNA methylation are the current gold standard, their usage has some caveats. Bisulfite conversion itself introduces a range of systematic sequencing biases by DNA degradation. For example, DNA degradation induced by bisulfite conversion depletes more regions with unmethylated cytosines, while incomplete conversion can lead to overestimation of methylated cytosines. On top of these, methylation status of specific CpG-rich regions may cause their under- or over-estimation in WGBS experiments. These biases are even more pronounced during library amplification step (Olova *et al.*, 2017).

If single base resolution is not required, several other 5hmC detection methods could be considered: antibody immunoprecipitation (h)MeDIP-seq (Bock *et al.*, 2010); or selective labelling and pulldown (e.g., hMe-Seal) (Han *et al.*, 2016; Song *et al.*, 2012b). Analysis packages such as MeDIPS (Lienhard *et al.*, 2014) provide a suite of tools to analyse (h)MeDIP-seq data including the identification of differentially methylated regions. Although not offering base resolution and lacking sensitivity for regions with low enrichment, antibody immunoprecipitation can enrich for each of DNA modifications with specific antibody, ensuring high specificity at low cost (Tan *et al.*, 2013). Independent of the chosen method it is good practice to validate methylation calls at a locus-specific level with methods such as bisulfite pyrosequencing (Strogantsev *et al.*, 2015).

Analysis

Quality Control and Mapping to the Genome

With all the methods detailed in [Table 1](#), the first step is to map the experimental output to a reference genome. In the case of array based methods, the reported individual probes are mapped to the genome via a lookup table of locations provided by the manufacturer. With second and third generation sequencing methods, the read sequences have to be aligned to a reference genome.

Array methods

Mapping of probes to their genomic locations doesn't require an explicit alignment step as is required with NGS data; instead the probe genomic locations are provided in the manufacturer supplied manifests for the arrays. The quality control steps required for the analysis of the array data is centered around removing poorly performing probes and corrections to batch effects due to bead detection on the chip. There are also control probes on the chip which should be interrogated for a variety of QC metrics. Illumina's Infinium HumanMethylation 450K and MethylationEPIC array chips are based on two distinct probe designs for assaying the methylation status of CpGs requiring normalization to match the differing methylation profiles. Illumina methylation arrays can be analysed using *missMethyl* (Phipson *et al.*, 2015) or *Chip Analysis Methylation Pipeline* (ChAMP) (Morris *et al.*, 2014). Both *missMethyl* and ChAMP R packages accept raw IDAT format files as input and integrate a selection of normalization methods (e.g. SWAN or BMIQ), differential methylation analysis and gene set analysis. Batch effects are corrected using *Combat* (Johnson *et al.*, 2007) in ChAMP and *RUVm* (Maksimovic *et al.*, 2015) in *missMethyl* which is designed to correct unknown source of variation as well as batch effects. ChAMP can additionally identify copy number alterations present in a dataset. Both packages provide the methylation states of the probes passing the QC steps.

Next generation sequencing

Standard steps in BS-seq data analysis are: adapter and sequence trimming, assurance of reads quality, alignment of the reads to the reference genome and methylation calling. We discuss each of these steps in the sections below. [Fig. 2](#) outlines a typical pipeline for the analysis of BS-seq data.

Trimming

Bisulfite conversion itself can introduce biases, as less efficient bisulfite conversion at 5' ends is commonly observed, and this could be misinterpreted as high methylation calls. Also, in any paired-end directional BS-Seq, 5' overhangs created by sonication are

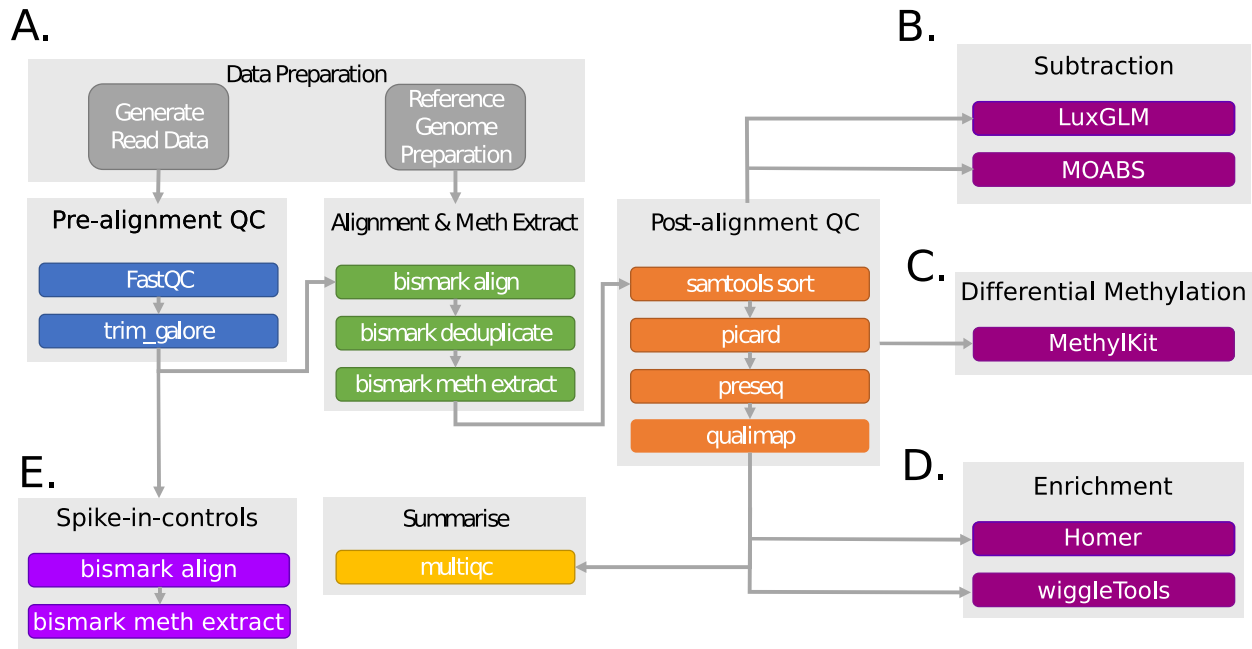


Fig. 2 A typical analysis pipeline for bisulfite based assays for detecting modified cytosine positions. A. Quality control and alignments steps. B. LuxGLM and MOABS are able to perform the required subtraction between different bisulfite based assays to call, e.g., 5hmC. C. Example software for calculating differential methylation between sample groups (e.g., case and control). D. Enrichment for cytosine modifications and histone marks can be performed with programs such as Homer and wiggleTools.

filled with unmodified C during end-repair step in library preparation, which can introduce biases in the genuine methylation state of cytosine at both ends of sequenced reads. Similarly, in RRBS, enzymatic reaction of *MspI* also creates 5' overhangs, and the missing positions are filled with unmodified C, but not of the true genomic cytosine. Additional consideration is also required for the sequencing of Post-Bisulfite Adapter Tagging (PBAT) libraries. Random priming step can introduce biases in base composition and thus in detecting methylation levels, which are affected mostly at the 5' end of all reads. Therefore, it's important to trim affected bases (length of random oligo primers) from the libraries before the alignment step to remove methylation biases. With tools such as Trim Galore! (see Relevant Websites) and Cutadapt (Martin, 2011) all of the biases can be efficiently removed.

Aligners and methylation callers

Mapping of the bisulfite converted reads to a reference genome is challenging due to C to T conversion and therefore reduced sequence complexity. In the course of bisulfite treatment, the top and bottom strands become non-complementary and their PCR amplification products differ, resulting in 4 different sequences: original top (OT), original bottom (OB), complementary top (CT) and complementary bottom (CB) strands. Different aligners approach these issues slightly differently, but arguably the most widely used is Bismark (Krueger and Andrews, 2011). Bismark is considered to be a conservative aligner, with expected alignment rates of around 80% (human), and may be a good choice as its very well supported by the developers. Bismark also has tools to extract methylation calls and distinguishes between the different cytosine contexts: CpG, CHH and CGH. methylKit can perform the methylation calls directly from aligned BAM files as well as import methylation calls from Bismark (Akalın et al., 2012). This is particularly convenient for creating pipelines as methylKit is primarily utilised for differential methylation analysis (see Section Differential Methylation). Alternatively, BWA-Meth offers more lenient aligning with rates (e.g., upwards of 90% (human) as it performs soft-clipping). This tool can be very useful if investigating genomic variants and may allow more reads to be aligned, but also some reads may be incorrectly mapped to genomic regions like repetitive regions.

The presence of SNPs can contribute to the miscalling of cytosine modifications, especially through C to T substitutions. Therefore, knowing the positions of common SNPs, for example, from dbSNP (Sherry et al., 2001), can be utilised to mask these positions in SNP aware alignment/calling software such as Bis-SNP (Liu et al., 2012).

In order for 5hmC, 5fC and 5caC to be accurately called in bisulfite based sequencing, there is a requirement for high read coverage to separate the true methylation signal from background noise (Discussed below). The high cost of sequencing to achieve the required coverage can be ameliorated with tools such as BSmooth (Hansen et al., 2012). BSmooth provides an entire pipeline from the raw reads, through alignment to differential methylation calling. However, the feature of particular interest is the smoothing of the methylation signal across regions to improve the precision of calling the methylation status of cytosine modifications. In this way, low coverage regions can be included in the differential methylation calling, where they may have been excluded in Fisher's Exact test based methods (Discussed below).

Spike in controls

It is important to monitor conversion rates of the bisulfite and oxidation reactions to ensure they are within recommended limits. Deviations from these limits will have a significant negative impact during the downstream calling of the cytosine modifications. Spike-in-controls can be artificial sequences or from a different species to the samples, and are added prior to bisulfite conversion to assay the efficiency of conversion in analysed samples. As the sequence of the spike-in-controls are known they can be aligned to a control reference and analysed in a similar manner to the sample sequences. The absolute quantification of spike-in-controls may also prove even more valuable in circumstances when global changes in methylation occur ([Hardwick et al., 2017](#)). In addition to the usage of spike-in-controls, methylation callers should also provide reports including the proportions of cytosine and methylated cytosines in the aligned reads.

Determining the Cytosine Modification Proportions

Once a sample's detectable cytosines have been mapped to the genome, regardless of the bisulfite- or enzyme-based assay used, in order to determine the relative proportions of the assayed cytosine modifications, subtractions must be performed. For example, to infer the 5hmC level at a specific cytosine, the number of methylated cytosines in a pileup from oxBS (5mC) must be subtracted from BS (5mC & 5hmC) ([Table 2](#)). Inherent in the subtraction of data is the accumulation of noise from the separate assays. Great care must be taken to perform read filtering and apply statistical tests to minimize the effect of the noise which can lead to biologically meaningless negative values.

The original publications for the single-base resolution sequencing of 5hmC used a minimum read coverage filter for CpG sites and for statistical significance of the difference between the bisulfite and oxidation results; Benjamini-Hochberg and Fisher's tests were used ([Booth et al., 2014, 2013, 2012](#)). A further recommendation was to pool CpG sites from within CpG Islands where five or more CpG site methylation values did not vary more than 20% for 5mC and 10% for 5hmC. The pooling of the CpG sites reduces the high coverage requirements for calling cytosine modifications at single CpG positions. Coverage of over 500 reads would be required to reliably detect below 5% hmC, where pooling enables detection of hmC down to 1% ([Booth et al., 2013](#)). This protocol, which we refer to as the Booth method, for read coverage and statistical testing should be considered the minimum requirement for performing a subtraction and is applicable to both array and NGS approaches.

Methylation arrays

The analysis results from MissMethyl and ChAMP, as detailed above, provide the methylation status of each of the assayed CpG site on the array. However, neither method performs the subtractions required to call the proportions of 5hmC, 5fC or 5caC modifications. To address this, two related methods were developed for the validation of the compatibility of the oxBS treatment on the 450K array ([Field et al., 2015; Stewart et al., 2015](#)). Subtractions between BS and oxBS to call 5hmC are performed for CpG sites with a methylation difference over a threshold ($\Delta\beta > 0.3$). This filtering has the effect of minimizing the number of negative values in a similar manner to the Booth approach for NGS. A more recent package, OxyBS ([Houseman et al., 2016](#)), utilises maximum likelihood estimates to call 5hmC from matched BS and oxBS samples. This method goes beyond naïve threshold filtering by modelling the methylation states (5C, 5mC, 5hmC) in the BS and oxBS paired samples.

Next-generation sequencing

There are several tools available for performing the subtractions required to make 5hmC calls ([Table 2](#)). MLML ([Qu et al., 2013](#)) is one of the earliest methods for estimating 5mC and 5hmC levels and has been developed for BS/oxBS, TAB-Seq and a combination of BS/oxBS and TAB-Seq. Maximum likelihood estimates are used to model the methylation states, and at sites with negative values the latent variables are used to approximate the MLE. oxBS-MLE ([Xu et al., 2016](#)) uses binomial modelling of paired BS and oxBS samples and is applicable to both array and sequencing methods. The method is related to OxyBS, but overcomes the CPU time constraints enabling studies to be performed on much larger sample sizes. MOABS ([Sun et al., 2014](#)) was

Table 2 Comparison table of a selection of subtraction methods for calling cytosine modifications. A variety of modelling approaches are used: Maximum Likelihood Estimates (MLE), Hidden Markov Models (HMM), General Linear Models (GLM). None of the methods have been applied and published for 5fC, and 5caC, however, they should be in theory, with optimizations applicable

	Statistical significance	Negative value filtering/modelling	Utilises biological replicates	Array/NGS/Both	5mC/5hmC/5fC/5caC/All
Booth Methods	Y	Filtering	No	NGS	All
OxBS-Array	Y	Filtering	No	Array	5mC/5hmC
Oxy-BS	Y	Modelling (MLE)	No	Array	5mC/5hmC
oxBS-MLE	Y	Modelling (MLE)	No	Both	5mC/5hmC
MLML	Y	Modelling (MLE)	No	Both	5mC/5hmC
MOABS	Y	Modelling (Beta-binomial)	Yes	NGS	5mC/5hmC
H(O)TA	Y	Modelling (hMM)	Time points	NGS	5mC/5hmC
LuxGLM	Y	Modelling (GLM)	Yes	NGS	All

primarily designed to improve the calling of 5mC sites in traditional bisulfite sequencing, however an extension to the beta-binomial method permits 5hmC calling. MOABS also takes into account biological replicates in methylation calls to account for biological variation. H(O)TA (Kyriakopoulos *et al.*, 2017) employs hidden Markov Models (hMM) of the methylation states of the cytosines in BS and oxBS paired samples across experimental time points. LuxGLM (Äijö *et al.*, 2016a,b) is one of the more recently published methods and was designed to work with a variety of chemical and enzymatic treatments followed by sequencing. A general linear model (GLM) is able to capture all cytosine modification methylation states across replicates to estimate the proportions of the modifications assayed.

Differential Methylation

Differentially Methylated Positions (DMPs) or Regions (DMRs) are genomic regions exhibiting distinct methylation states between biological samples, and are thought to have the potential to regulate gene expression. Depending on the samples interrogated, DMRs can be further classified into, e.g., tissue-specific, allele-specific, developmental stage or disease-associated (such as cancer-specific DMRs). Genome wide association studies have shown specific DMRs to be associated with disease states, while locus-specific studies confirmed their functional roles in regulating gene expression. Clear links have been made between DMRs and diseases such as systemic lupus erythematosus, cancer, aging, obesity, inflammatory bowel disease (Baba *et al.*, 2010; Jeffries *et al.*, 2011; McKinney *et al.*, 2015; Soubry *et al.*, 2015; Stirzaker *et al.*, 2016; Venthram *et al.*, 2016). Several DMRs have proved to be valuable epigenetic biomarkers with clinical applications (Barault *et al.*, 2015; Church *et al.*, 2014; Dietrich *et al.*, 2012; Imperiale *et al.*, 2014; Kristensen *et al.*, 2016). Patterns of 5hmC are tissue specific and associate with tissue, organ and developmental stage functionality. For example, in placenta, maternal DNA methylation regulates development of early trophoblast, while unique 5mC and 5hmC patterns of placenta associate with birth weight (Branco *et al.*, 2016; Fogarty *et al.*, 2015; Green *et al.*, 2016; Piyasena *et al.*, 2015; Zhu *et al.*, 2015). The latter example points a valid point how important is to discriminate between 5mC and 5hmC and that bisulfite-seq data (non-discriminative between 5mC and 5hmC) should be treated very carefully.

Determining the locations of the cytosine modifications on the genome is of particular interest in development, however it is often desirable to perform a comparison across conditions. For example, comparisons across tissues, investigating the effect of a gene knockout with CRISPR or in biomarker discovery in a disease case and control. There are a wealth of methods for finding differentially methylated genomic regions. Typically, these require more hands-on tailoring for the most appropriate comparisons in the study in hand and are often supplied as R packages. All the methods highlighted below perform multiple testing correction. BiSeq is designed for RRBS data analysis and performs DMR finding using the Wald statistic from the beta-binomial regression fit R package (Hebestreit *et al.*, 2013). BSmooth searches for bumps on smoothed t-like scores of methylation profiles to find DMRs (Hansen *et al.*, 2012). methylSig is an R package for calling DMRs with likelihood ratio test (and also a beta-binomial approach) (Park *et al.*, 2014). methylKit, an R package, identifies DMRs with logistic regression and Fisher's exact tests (Akalin *et al.*, 2012). methylKit can also directly input methylation calls from Bismark making it suitable for integrating into a pipeline. methylPipe employs Wilcoxon or Kruskal–Wallis paired non-parametric test to identify DMRs and the predicted DMRs can be *in silico* tested for validation with compEpiTools (Kishore *et al.*, 2015).

It is important to state that the identification of DMRs is only a discovery step; further experimental validation is essential. Once a large whole genome case/control study has identified and validated a DMR/biomarker it can be incorporated into a targeted panel to greatly reduce the cost for diagnostic purposes.

Interpretation

Genomic Distribution of Cytosine Modifications

Once the detectable cytosines modifications have been mapped and processed to provide relative proportions of the measurable modifications, it is then desirable to look for where they accumulate in the genome and interpret the biological significance in the context of the experiment being performed. Genome tracks such as RefSeq genes (intron/exon), promoters, enhancers, repeats (inc Alu LINE elements), CpG Islands, Shores, Shelves can be downloaded in BED format from the UCSC Genome Browser (Raney *et al.*, 2014). Tools such as bedTools2 can then be used to perform intersections with the elements of choice to address specific biological questions such as: are 5hmC sites enriched in promoters as compared to enhancers? (Quinlan and Hall, 2010).

Correlation With ChIP-Seq

Chromatin immunoprecipitation followed by high throughput sequencing (ChIP-seq) provides genomic locations of histone marks such as H3K4me3 found in promoters under active transcription and H3K27ac marking active over poised enhancers. Studies performing methylation sequencing are often coupled with ChIP-Seq for the specific tissue or cell type of interest (Guo *et al.*, 2017). In order to determine the co-localization of cytosine modifications with the histone marks tools such as wiggleTools (Zerbino *et al.*, 2014), Homer (Heinz *et al.*, 2010), bedTools2 (Quinlan and Hall, 2010), and GIGGLE (Layer *et al.*, 2017) can be used.

Correlation With RNA-Seq

DNA methylation analysis alone may be difficult to interpret, as DNA modifications can exert different effects depending on their position relative to the genes. To make a compelling case for the role of the assayed cytosine modifications on gene expression, it is recommended to perform a matched sample RNA-Seq experiment. Bulk RNA-Seq or transcription start site (TSS) specific CAGE-Seq enable the effect on gene expression in differentially methylated regions to be quantified (Steyaert *et al.*, 2016). The integrative transcriptional-epigenomic studies and their bioinformatics analyses are far more challenging in the single-cell approaches, but promising methods are under development (Angermueller *et al.*, 2016; Clark *et al.*, 2016; Hu *et al.*, 2016).

Allelic Expression

Not all genes are expressed from both alleles. Some genes are expressed in a parent-of-origin specific manner and do not follow classical Mendelian inheritance. These are known as imprinted genes and since both active and repressed copies of imprinted genes can co-exist in the same nucleus, they are particularly suited as a model system to study epigenetic gene regulation. Imprinted genes have been shown to play a role in the growth of the placenta and the development of the embryo, as well as having postnatal behavioral and physiological functions (Frost and Moore, 2010; Garfield *et al.*, 2011; Li *et al.*, 1999).

As imprinted genes are expressed from one allele, they are more susceptible to mutations and dysregulation, coinciding with their abnormal expression in diseases and cancer. A common feature found in imprinted gene clusters are imprinting control regions (ICRs) that are differentially methylated on both alleles (and so they are differentially methylated regions, DMRs). Parent of origin specific methylation is established in germline and contributes to allele specific expression of imprinted genes. Also, some loci exhibit non-imprinted allelic methylation that occurs at random. This phenomenon can be partially explained by existence of SNPs with CpGs (Rouhi *et al.*, 2006; Shoemaker *et al.*, 2010; Strogantsev *et al.*, 2015). To computationally determine the parental origin of an allele, the reads must be aligned to the genome of each parent. In human samples, genotyping permits the parental origin of the allele to be resolved. It is also possible to use triploidies to take advantage of parental genomic contributions (Yuen *et al.*, 2011). In mouse, the use of hybrid crosses allows the parental origin of the gene of interest to be resolved (Kobayashi *et al.*, 2009).

Discussion

In this article, competing methods for assaying DNA modifications were reviewed, many of which are able to assay cytosine modifications at single base resolution. However, there are limitations to the current methods, such as the requirement for chemical or enzymatic treatments, PCR amplification and high depth sequencing, all requiring computational tools to account for these sources of deviation from the true biological signal.

In mammals, cytosine modifications are predominantly found at CpG sites, and often clustered in CpG islands. These regions are therefore of low complexity due to the enrichment of CpG sites making the unique mapping of short-reads difficult, or impossible, leaving gaps in assaying the methylome. To cover these gaps, new sequencing technologies such as long-read sequencing (see Section Future Directions) are required, and accompanying bioinformatics tools to assay the nucleotide modifications need to be developed. Tool development will be key to unlocking these previously un-mappable regions of genomes.

Importantly, cytosine modifications at single base resolution need to be carefully interpreted, ideally with orthogonal methods such as ChIP-Seq or RNA-Seq, as well as with allelic expression. Together, they make powerful tools to dissect roles of DNA modifications in epigenetic regulation of gene expression.

Future Directions

Emerging technologies, such as the long-read sequencing technologies, are rapidly approaching the accuracy of the short-read based methods for methylation sequencing. A further development expected over the next few years is the ability to detect all cytosine modifications within single cells. Methods have recently been developed to assay 5mC in single cells and further developments have permitted the simultaneous sequencing the methylome and transcriptome from the same cell.

Long-Read Sequencing

Cytosines undergoing modifications are often located in clusters (e.g., CpG islands) which are therefore repetitive in nature. Short-read based methods are unable to uniquely map short reads to repetitive regions, resulting in the under representation of these important regions. Long-read based methods such as Pacific Biosciences single-molecule real-time sequencing (SMRT) and Oxford Nanopore do not suffer from this limitation as the reads can be 10's or 100's of kilobases in length and can therefore span and uniquely map to these CpG dense regions. A further key advantage is the ability to directly assess the methylation status of cytosines without the need for treatments such as bisulfite, or enzymatic modification. SMRT sequencing measures the

incorporation of nucleotides in real time and is able to distinguish between each of the modification states of cytosine and adenine (Fang *et al.*, 2012; Suzuki *et al.*, 2015), while Nanopore technology detects DNA bases, including modified cytosines, by measuring electrostatic charge as a DNA strand passes through a protein nanopore (Rand *et al.*, 2017; Simpson *et al.*, 2017). The challenge now is to develop new bioinformatics methods for long-read sequencing such as high accuracy base calling of methylated bases

Single Cell Sequencing

Methods have been developed to assay 5mC in single cells using bisulfite treatments (Smallwood *et al.*, 2014). However, to date, only TAB-Seq is able to assay 5hmC in single cell treatments as the bisulfite methods are too harsh and unable to achieve the required read depths for the subtractions (Clark *et al.*, 2016). More recent methods, e.g., G&T-Seq, have been developed to simultaneously sequence the genome and transcriptome (Macaulay *et al.*, 2015). In the future, it will hopefully be possible to simultaneously assay all the levels of epigenetic detail of a single cell such as cytosine modifications, histone marks, chromatin structure and 3D genome arrangements in combination with the transcriptome, genome and proteome (Clark *et al.*, 2016). As with long-read sequencing methods the challenge for the bioinformatics community is to develop new multi-modal methods to simultaneously analyse the multi-ome data sets derived from single cells.

Closing Remarks

The current state-of-the-art methods for interrogating cytosine modifications rely on chemical or enzymatic treatments accompanied with PCR amplification. Future gold standard methods will undoubtedly forgo these treatments and directly infer the methylation status of DNA and RNA nucleotides from the native molecules. The current, and future methods, will continue to further our understanding of the biological role of nucleotide modifications and their impact in human disease. Perhaps, most excitingly, these modification marks could be edited with systems such as CRISPR-CAS9 to treat a wide range of diseases.

Acknowledgements

Russell S. Hamilton and Malwina Prater are funded by the Centre for Trophoblast Research (University of Cambridge).

See also: Natural Language Processing Approaches in Bioinformatics

References

- Adams, D., Altucci, L., Antonarakis, S.E., *et al.*, 2012. BLUEPRINT to decode the epigenetic signature written in blood. *Nat. Biotechnol.* 30, 224–226. doi:10.1038/nbt.2153.
- Äijö, T., Huang, Y., Mannerström, H., *et al.*, 2016a. A probabilistic generative model for quantification of DNA modifications enables analysis of demethylation pathways. *Genome Biol.* 17, 49. doi:10.1186/s13059-016-0911-6.
- Äijö, T., Yue, X., Rao, A., Lähdesmäki, H., 2016b. LuxGLM: A probabilistic covariate model for quantification of DNA methylation modifications with complex experimental designs. *Bioinformatics* 32, i511–i519. doi:10.1093/bioinformatics/btw468.
- Akalın, A., Kormaksson, M., Li, S., *et al.*, 2012. methylKit: A comprehensive R package for the analysis of genome-wide DNA methylation profiles. *Genome Biol.* 13, R87. doi:10.1186/gb-2012-13-10-r87.
- Albrecht, F., List, M., Bock, C., Lengauer, T., 2017. DeepBlueR: Large-scale epigenomic analysis in R. *Bioinformatics* 33, 2063–2064. doi:10.1093/bioinformatics/btx099.
- Angermueller, C., Clark, S.J., Lee, H.J., *et al.*, 2016. Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nat. Methods* 13, 229–232. doi:10.1038/nmeth.3728.
- Baba, Y., Noshio, K., Shima, K., *et al.*, 2010. Hypomethylation of the IGF2 DMR in colorectal tumors, detected by bisulfite pyrosequencing, is associated with poor prognosis. *Gastroenterology* 139, 1855–1864. doi:10.1053/j.gastro.2010.07.050.
- Bachman, M., Uribe-Lewis, S., Yang, X., *et al.*, 2014. 5-Hydroxymethylcytosine is a predominantly stable DNA modification. *Nat. Chem.* 6, 1049–1055. doi:10.1038/nchem.2064.
- Ball, M.P., Li, J.B., Gao, Y., *et al.*, 2009. Targeted and genome-scale strategies reveal gene-body methylation signatures in human cells. *Nat. Biotechnol.* 27, 361–368. doi:10.1038/nbt.1533.
- Barault, L., Amatu, A., Bleeker, F.E., *et al.*, 2015. Digital PCR quantification of MGMT methylation refines prediction of clinical benefit from alkylating agents in glioblastoma and metastatic colorectal cancer. *Ann. Oncol.* 26, 1994–1999. doi:10.1093/annonc/mdv272.
- Bernstein, B.E., Stamatoyannopoulos, J.A., Costello, J.F., *et al.*, 2010. The NIH roadmap epigenomics mapping consortium. *Nat. Biotechnol.* 28, 1045–1048. doi:10.1038/nbt1010-1045.
- Bock, C., Tomazou, E.M., Brinkman, A.B., *et al.*, 2010. Quantitative comparison of genome-wide DNA methylation mapping technologies. *Nat. Biotechnol.* 28, 1106–1114. doi:10.1038/nbt.1681.
- Booth, M.J., Branco, M.R., Ficiz, G., *et al.*, 2012. Quantitative sequencing of 5-methylcytosine and 5-hydroxymethylcytosine at single-base resolution. *Science* 336 (80-), 934–937. doi:10.1126/science.1220671.
- Booth, M.J., Marsico, G., Bachman, M., Beraldi, D., Balasubramanian, S., 2014. Quantitative sequencing of 5-formylcytosine in DNA at single-base resolution. *Nat. Chem.* 6, 435–440. doi:10.1038/nchem.1893.

- Booth, M.J., Ost, T.W.B., Beraldi, D., *et al.*, 2013. Oxidative bisulfite sequencing of 5-methylcytosine and 5-hydroxymethylcytosine. *Nat. Protoc.* 8, 1841–1851. doi:10.1038/nprot.2013.115.
- Bourchis, D., Bestor, T.H., 2004. Meiotic catastrophe and retrotransposon reactivation in male germ cells lacking Dnmt3L. *Nature* 431, 96–99. doi:10.1038/nature02886.
- Branco, M.R., King, M., Perez-garcia, V., *et al.*, 2016. Maternal DNA methylation regulates early trophoblast development article maternal DNA methylation regulates. *Dev. Cell* 36, 152–163. doi:10.1016/j.devcel.2015.12.027.
- Church, T.R., Wandell, M., Lofton-Day, C., *et al.*, 2014. Prospective evaluation of methylated SEPT9 in plasma for detection of asymptomatic colorectal cancer. *Gut* 63, 317–325. doi:10.1136/gutjnl-2012-304149.
- Clark, S.J., Lee, H.J., Smallwood, S.A., Kelsey, G., Reik, W., 2016. Single-cell epigenomics: Powerful new methods for understanding gene regulation and cell identity. *Genome Biol.* 17, 72. doi:10.1186/s13059-016-0944-x.
- Consortium, E.N.C.O.D.E.P., Bernstein, B.E., Birney, E., *et al.*, 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74. doi:10.1038/nature11247.
- Delgado-Olguin, P., Huang, Y., Li, X., *et al.*, 2012. Epigenetic repression of cardiac progenitor gene expression by Ezh2 is required for postnatal cardiac homeostasis. *Nat. Genet.* 44, 343–347. doi:10.1038/ng.1068.
- Dietrich, D., Kneip, C., Raji, O., *et al.*, 2012. Performance evaluation of the DNA methylation biomarker SHOX2 for the aid in diagnosis of lung cancer based on the analysis of bronchial aspirates. *Int. J. Oncol.* 40, 825–832. doi:10.3892/ijo.2011.1264.
- Du, J., Johnson, L.M., Jacobsen, S.E., Patel, D.J., 2015. DNA methylation pathways and their crosstalk with histone methylation. *Nat. Rev. Mol. Cell Biol.* 16, 519–532. doi:10.1038/nrm4043.
- Fang, G., Munera, D., Friedman, D.I., *et al.*, 2012. Genome-wide mapping of methylated adenine residues in pathogenic *Escherichia coli* using single-molecule real-time sequencing. *Nat. Biotechnol.* 30, 1232–1239. doi:10.1038/nbt.2432.
- Field, S.F., Beraldi, D., Bachman, M., *et al.*, 2015. Accurate measurement of 5-methylcytosine and 5-hydroxymethylcytosine in human cerebellum DNA by oxidative bisulfite on an array (OxBS-Array). *PLOS ONE* 10, 1–12. doi:10.1371/journal.pone.0118202.
- Fogarty, N.M.E., Burton, G.J., Ferguson-Smith, A.C., 2015. Different epigenetic states define syncytiotrophoblast and cytotrophoblast nuclei in the trophoblast of the human placenta. *Placenta* 36, 796–802. doi:10.1016/j.placenta.2015.05.006.
- Frost, J.M., Moore, G.E., 2010. The importance of imprinting in the human placenta. *PLOS Genet.* 6, 1–9. doi:10.1371/journal.pgen.1001015.
- Frye, M., Jaffrey, S.R., Pan, T., Rechavi, G., Suzuki, T., 2016. RNA modifications: What have we learned and where are we headed? *Nat. Rev. Genet.* 17, 365–372. doi:10.1038/nrg.2016.47.
- Garfield, A.S., Cowley, M., Smith, F.M., *et al.*, 2011. Distinct physiological and behavioural functions for parental alleles of imprinted Grb10. *Nature* 469, 534. doi:10.1038/nature09651. [U101].
- Goll, M.G., Bestor, T.H., 2005. Eukaryotic cytosine methyltransferases. *Annu. Rev. Biochem.* 74, 481–514. doi:10.1146/annurev.biochem.74.010904.153721.
- Green, B.B., Houseman, E.A., Johnson, K.C., *et al.*, 2016. Hydroxymethylation is uniquely distributed within term placenta, and is associated with gene expression. *FASEB J.* 30, 1–11. doi:10.1096/fj.201600310R.
- Gross, J.A., Lefebvre, F., Lutz, P.-E., *et al.*, 2016. Variations in 5-methylcytosine and 5-hydroxymethylcytosine among human brain, blood, and saliva using oxBS and the Infinium MethylationEPIC array. *Biol. Methods Protoc.* bpw002. doi:10.1093/biomethods/bpw002.
- Guo, H., Hu, B., Yan, L., *et al.*, 2017. DNA methylation and chromatin accessibility profiling of mouse and human fetal germ cells. *Cell Res.* 27, 165–183. doi:10.1038/cr.2016.128.
- Guo, J.U., Su, Y., Shin, J.H., *et al.*, 2013. Distribution, recognition and regulation of non-CpG methylation in the adult mammalian brain. *Nat. Neurosci.* 17, 215–222. doi:10.1038/nn.3607.
- Guo, W., Chung, W.Y., Qian, M., Pellegrini, M., Zhang, M.Q., 2014. Characterizing the strand-specific distribution of non-CpG methylation in human pluripotent cells. *Nucleic Acids Res.* 42, 3009–3016. doi:10.1093/nar/gkt1306.
- Hahn, M.A., Wu, X., Li, A.X., Hahn, T., Pfeifer, G.P., 2011. Relationship between gene body DNA methylation and intragenic H3K9ME3 and H3K36ME3 chromatin marks. *PLOS ONE* 6. doi:10.1371/journal.pone.0018844.
- Hansen, K.D., Langmead, B., Irizarry, R.A., *et al.*, 2012. BSmooth: From whole genome bisulfite sequencing reads to differentially methylated regions. *Genome Biol.* 13, R83. doi:10.1186/gb-2012-13-10-r83.
- Han, D., Lu, X., Shih, A.H., *et al.*, 2016. A highly sensitive and robust method for genome-wide 5hmC profiling of rare cell populations. *Mol. Cell* 63, 711–719. doi:10.1016/j.molcel.2016.06.028.
- Hardwick, S.A., Deveson, I.W., Mercer, T.R., 2017. Reference standards for next-generation sequencing. *Nat. Rev. Genet.* 18, 473–484. doi:10.1038/nrg.2017.44.
- Hebestreit, K., Dugas, M., Klein, H.U., 2013. Detection of significantly differentially methylated regions in targeted bisulfite sequencing data. *Bioinformatics* 29, 1647–1653. doi:10.1093/bioinformatics/btt263.
- Heinz, S., Benner, C., Spann, N., *et al.*, 2010. Article simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol. Cell* 38, 576–589. doi:10.1016/j.molcel.2010.05.004.
- He, Y.-F., Li, B.-Z., Li, Z., *et al.*, 2011. Tet-Mediated Formation of 5-Carboxylcytosine and Its Excision by TDG in Mammalian DNA. *Science* 333 (80-), 1303–1307. doi:10.1126/science.1210944.
- Horvath, S., 2013. DNA methylation age of human tissues and cell types. *Genome Biol.* 14, R115. doi:10.1186/gb-2013-14-10-r115.
- Houseman, E.A., Johnson, K.C., Christensen, B.C., 2016. OxyBS: Estimation of 5-methylcytosine and 5-hydroxymethylcytosine from tandem-treated oxidative bisulfite and bisulfite DNA. *Bioinformatics* 32, 2505–2507. doi:10.1093/bioinformatics/btw158.
- Hu, Y., Huang, K., An, Q., *et al.*, 2016. Simultaneous profiling of transcriptome and DNA methylome from a single cell. *Genome Biol.* 17, 88. doi:10.1186/s13059-016-0950-z.
- Imperiale, T.F., Ransohoff, D.F., Itzkowitz, S.H., *et al.*, 2014. Multitarget stool DNA testing for colorectal-cancer screening. *N. Engl. J. Med.* 370, 1287–1297. doi:10.1056/NEJMoa1311194.
- Ito, S., Shen, L., Dai, Q., *et al.*, 2011. Tet proteins can convert 5-methylcytosine to 5-formylcytosine and 5-carboxylcytosine. *Science* 333 (80-), 1300–1303. doi:10.1126/science.1210597.
- Iurlaro, M., Fic, G., Oxley, D., *et al.*, 2013. A screen for hydroxymethylcytosine and formylcytosine binding proteins suggests functions in transcription and chromatin regulation. *Genome Biol.* 14, R119. doi:10.1186/gb-2013-14-10-r119.
- Iurlaro, M., McInroy, G.R., Burgess, H.E., *et al.*, 2016. In vivo genome-wide profiling reveals a tissue-specific role for 5-formylcytosine. *Genome Biol.* 17, 141. doi:10.1186/s13059-016-1001-5.
- Jeffries, M.A., Dozmorov, M., Tang, Y., *et al.*, 2011. Genome-wide DNA methylation patterns in CD4+ T cells from patients with systemic lupus erythematosus. *Epigenetics* 6, 593–601. doi:10.4161/epi.6.5.15374.
- Johnson, W.E., Li, C., Rabinovic, A., 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 8, 118–127. doi:10.1093/biostatistics/kxj037.
- Kafer, G.R., Li, X., Horii, T., *et al.*, 2016. 5-Hydroxymethylcytosine marks sites of DNA damage and promotes genome stability. *Cell Rep.* 14, 1283–1292. doi:10.1016/j.celrep.2016.01.035.
- Kellinger, M.W., Song, C.-X., Chong, J., *et al.*, 2012. 5-formylcytosine and 5-carboxylcytosine reduce the rate and substrate specificity of RNA polymerase II transcription. *Nat. Struct. Mol. Biol.* 19, 831–833. doi:10.1038/nsmb.2346.
- Kishore, K., de Pretis, S., Lister, R., *et al.*, 2015. methylPipe and compEpiTools: A suite of R packages for the integrative analysis of epigenomics data. *BMC Bioinform.* 16, 1–11. doi:10.1186/s12859-015-0742-6.

- Kobayashi, H., Yamada, K., Morita, S., *et al.*, 2009. Identification of the mouse paternally expressed imprinted gene Zdbf2 on chromosome 1 and its imprinted human homolog ZDBF2 on chromosome 2. *Genomics* 93, 461–472. doi:10.1016/j.ygeno.2008.12.012.
- Kohli, R.M., Zhang, Y., 2013. TET enzymes, TDG and the dynamics of DNA demethylation. *Nature* 502, 472–479. doi:10.1038/nature12750.
- Kristensen, L.S., Hansen, J.W., Kristensen, S.S., *et al.*, 2016. Aberrant methylation of cell-free circulating DNA in plasma predicts poor outcome in diffuse large B cell lymphoma. *Clin. Epigenetics* 8, 95. doi:10.1186/s13148-016-0261-y.
- Krueger, F., Andrews, S.R., 2011. Bismark: A flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics* 27, 1571–1572. doi:10.1093/bioinformatics/btr167.
- Kyriakopoulos, C., Giehr, P., Wolf, V., 2017. H(O)TA: Estimation of DNA methylation and hydroxylation levels and efficiencies from time course data. *Bioinformatics* 33, btx042. doi:10.1093/bioinformatics/btx042.
- Lavie, L., Kitova, M., Maldener, E., Meese, E., Mayer, J., 2005. CpG methylation directly regulates transcriptional activity of the human endogenous retrovirus family HERV-K (HML-2). *J. Virol.* 79, 876. doi:10.1128/JVI.79.2.876.
- Layer, R.M., Pedersen, B.S., DiSera, T., *et al.*, 2017. GIGGLE: A search engine for large-scale integrated genome analysis. *bioRxiv*. doi:10.1101/157735.
- Liang, G., Chan, M.F., Tomigahara, Y., *et al.*, 2002. Cooperativity between DNA methyltransferases in the maintenance methylation of repetitive elements cooperativity between DNA methyltransferases in the maintenance methylation of repetitive elements. *Mol. Cell. Biol.* 22, 480. doi:10.1128/MCB.22.2.480.
- Lienhard, M., Grimm, C., Morkel, M., Herwig, R., Chavez, L., 2014. MEDIPS: Genome-wide differential coverage analysis of sequencing data derived from DNA enrichment experiments. *Bioinformatics* 30, 284–286. doi:10.1093/bioinformatics/btt650.
- Li, E., Zhang, Y., 2014. DNA methylation in mammals. *Cold Spring Harb. Perspect. Biol.* 6, a019133. doi:10.1101/cshperspect.a019133. [a019133].
- Li, L., Keverne, E.B., Aparicio, S. a., *et al.*, 1999. Regulation of maternal behavior and offspring growth by paternally expressed Peg3. *Science* 284, 330–333. doi:10.1126/science.284.5412.330.
- Liu, Y., Siegmund, K.D., Laird, P.W., Berman, B.P., 2012. Bis-SNP: Combined DNA methylation and SNP calling for Bisulfite-seq data. *Genome Biol.* 13, R61. doi:10.1186/gb-2012-13-7-r61.
- Lunnon, K., Hannon, E., Smith, R.G., *et al.*, 2016. Variation in 5-hydroxymethylcytosine across human cortex and cerebellum. *Genome Biol.* 17, 27. doi:10.1186/s13059-016-0871-x.
- Macaulay, I.C., Haerty, W., Kumar, P., *et al.*, 2015. G&T-seq: Parallel sequencing of single-cell genomes and transcriptomes. *Nat. Methods* 12, 519–522. doi:10.1038/nmeth.3370.
- Madzo, J., Liu, H., Rodriguez, A., *et al.*, 2014. Hydroxymethylation at gene regulatory regions directs stem/early progenitor cell commitment during erythropoiesis. *Cell Rep.* 6, 231–244. doi:10.1016/j.celrep.2013.11.044.
- Maksimovic, J., Gagnon-Bartsch, J.A., Speed, T.P., Oshlack, A., 2015. Removing unwanted variation in a differential methylation analysis of Illumina HumanMethylation450 array data. *Nucleic Acids Res.* 43, e106. doi:10.1093/nar/gkv526.
- Marina, R.J., Oberdoerffer, S., 2016. Epigenomics meets splicing through the TETs and CTCF. *Cell Cycle* 15, 1397–1399. doi:10.1080/15384101.2016.1171650.
- Martin, M., 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet J.* 17, 10. doi:10.14806/ej.17.1.200.
- McKinney, B.C., Lin, C.-W., Oh, H., *et al.*, 2015. Hypermethylation of BDNF and SST genes in the orbital frontal cortex of older individuals: A putative mechanism for declining gene expression with age. *Neuropsychopharmacology* 40, 2604–2613. doi:10.1038/npp.2015.107.
- Morris, T.J., Butcher, L.M., Feber, A., *et al.*, 2014. ChAMP: 450k chip analysis methylation pipeline. *Bioinformatics* 30, 428–430. doi:10.1093/bioinformatics/btt684.
- Olova, N., Krueger, F., Andrews, S., *et al.*, 2017. Comparison of whole-genome bisulfite sequencing library preparation strategies identifies sources of biases affecting DNA methylation data. *BioRxiv*. doi:10.1101/165449.
- Park, Y., Figueroa, M.E., Rozek, L.S., Sartor, M. a., 2014. MethySig: A whole genome DNA methylation analysis pipeline. *Bioinformatics* 30, 1–8. doi:10.1093/bioinformatics/btu339.
- Paulsen, M., Ferguson-Smith, A., 2001. DNA methylation in genomic imprinting, development, and disease. *J. Pathol.* 195, 97–110. doi:10.1002/path.890.
- Phipson, B., Maksimovic, J., Oshlack, A., 2015. missMethyl: An R package for analyzing data from Illumina's HumanMethylation450 platform. *Bioinformatics* 32, btv560. doi:10.1093/bioinformatics/btv560.
- Piyasena, C., Reynolds, R.M., Khulan, B., *et al.*, 2015. Placental 5-methylcytosine and 5-hydroxymethylcytosine patterns associate with size at birth. *Epigenetics* 10, 692–697. doi:10.1080/15592294.2015.1062963.
- Quinlan, A.R., Hall, I.M., 2010. BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics* 26, 841–842. doi:10.1093/bioinformatics/btq033.
- Qu, J., Zhou, M., Song, Q., Hong, E.E., Smith, A.D., 2013. MLML: Consistent simultaneous estimates of DNA methylation and hydroxymethylation. *Bioinformatics* 29, 2645–2646. doi:10.1093/bioinformatics/btt459.
- Rand, A.C., Jain, M., Eizenga, J.M., *et al.*, 2017. Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods*. 1–6. doi:10.1038/nmeth.4189.
- Raney, B.J., Dreszer, T.R., Barber, G.P., *et al.*, 2014. Track data hubs enable visualization of user-defined genome-wide annotations on the UCSC genome browser. *Bioinformatics* 30, 1003–1005. doi:10.1093/bioinformatics/btt637.
- Raynal, N.J.M., Si, J., Taby, R.F., *et al.*, 2012. DNA methylation does not stably lock gene expression but instead serves as a molecular mark for gene silencing memory. *Cancer Res.* 72, 1170–1181. doi:10.1158/0008-5472.CAN-11-3248.
- Rouhi, A., Gagnier, L., Takei, F., Mager, D.L., 2006. Evidence for epigenetic maintenance of Ly49a monoallelic gene expression. *J. Immunol.* 176, 2991–2999. doi:10.4049/jimmunol.176.5.2991.
- Sérandour, A.A., Avner, S., Oger, F., *et al.*, 2012. Dynamic hydroxymethylation of deoxyribonucleic acid marks differentiation-associated enhancers. *Nucleic Acids Res.* 40, 8255–8265. doi:10.1093/nar/gks595.
- Sherry, S.T., Ward, M.H., Kholodov, M., *et al.*, 2001. dbSNP: The NCBI database of genetic variation. *Nucleic Acids Res.* 29, 308–311. doi:10.1093/nar/29.1.308.
- Shoemaker, R., Deng, J., Wang, W., Zhang, K., 2010. Allele-specific methylation is prevalent and is contributed by CpG-SNPs in the human genome. *Genome Res.* 20, 883–889. doi:10.1101/gr.104695.109.
- Simpson, J.T., Workman, R.E., Zuzarte, P.C., *et al.*, 2017. Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods*. 1–7. doi:10.1038/nmeth.4184.
- Skvortsova, K., Zotenko, E., Luu, P.-L., *et al.*, 2017. Comprehensive evaluation of genome-wide 5-hydroxymethylcytosine profiling approaches in human DNA. *Epigenetics Chromatin* 10, 16. doi:10.1186/s13072-017-0123-7.
- Smallwood, S.A., Lee, H.J., Angermueller, C., *et al.*, 2014. Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity. *Nat. Methods* 11, 817–820. doi:10.1038/nmeth.3035.
- Song, C.-X., Clark, T.A., Lu, X.-Y., *et al.*, 2012a. Sensitive and specific single-molecule sequencing of 5-hydroxymethylcytosine. *Nat. Methods* 9, 75–77. doi:10.1038/nmeth.1779.
- Song, C.-X., Yi, C., He, C., 2012b. Mapping recently identified nucleotide variants in the genome and transcriptome. *Nat. Biotechnol.* 30, 1107–1116. doi:10.1038/nbt.2398.
- Sood, A.J., Viner, C., Hoffman, M.M., 2016. DNAmdb: The DNA modification database. *bioRxiv* 1–13. doi:10.1101/071712.
- Soubry, A., Murphy, S.K., Wang, F., *et al.*, 2015. Newborns of obese parents have altered DNA methylation patterns at imprinted genes. *Int. J. Obes. (Lond)* 39, 650–657. doi:10.1038/ijo.2013.193.
- Spiers, H., Hannon, E., 2017. 5-hydroxymethylcytosine is highly dynamic across human fetal brain development Helen. *BioRxiv*. doi:10.1101/126169.
- Sprijt, C.G., Gnerlich, F., Smits, A.H., *et al.*, 2013. Dynamic readers for 5-(Hydroxy)methylcytosine and its oxidized derivatives. *Cell* 152, 1146–1159. doi:10.1016/j.cell.2013.02.004.
- Steyaert, S., Diddens, J., Galle, J., *et al.*, 2016. A genome-wide search for epigenetically regulated genes in zebra finch using MethylCap-seq and RNA-seq. *Sci. Rep.* 6, 20957. doi:10.1038/srep20957.
- Stewart, S.K., Morris, T.J., Guilhamon, P., *et al.*, 2015. OxBS-450K: A method for analysing hydroxymethylation using 450K BeadChips. *Methods* 72, 9–15. doi:10.1016/j.ymeth.2014.08.009.

- Stirzaker, C., Zotenko, E., Clark, S.J., 2016. Genome-wide DNA methylation profiling in triple-negative breast cancer reveals epigenetic signatures with important clinical value. *Mol. Cell. Oncol.* 3, e1038424. doi:10.1080/23723556.2015.1038424.
- Strogantsev, R., Krueger, F., Yamazawa, K., *et al.*, 2015. Allele-specific binding of ZFP57 in the epigenetic regulation of imprinted and non-imprinted monoallelic expression. *Genome Biol.* 16, 112. doi:10.1186/s13059-015-0672-7.
- Stroud, H., Feng, S., Morey Kinney, S., Pradhan, S., Jacobsen, S.E., 2011. 5-Hydroxymethylcytosine is associated with enhancers and gene bodies in human embryonic stem cells. *Genome Biol.* 12, R54. doi:10.1186/gb-2011-12-6-r54.
- Stubbs, T.M., Bonder, M.J., Stark, A., *et al.*, 2017. Multi-tissue DNA methylation age predictor in mouse. *Genome Biol.* 18, 68. doi:10.1186/s13059-017-1203-5.
- Sun, D., Xi, Y., Rodriguez, B., *et al.*, 2014. MOABS: Model based analysis of bisulfite sequencing data. *Genome Biol.* 15, R38. doi:10.1186/gb-2014-15-2-r38.
- Suzuki, Y., Korch, J., Turner, S.W., *et al.*, 2015. AgIn: Measuring the Landscape of CpG methylation of individual repetitive elements. *bioRxiv* 32, 18531. doi:10.1101/018531.
- Szulwach, K.E., Li, X., Li, Y., *et al.*, 2011. Integrating 5-hydroxymethylcytosine into the epigenomic landscape of human embryonic stem cells. *PLOS Genet.* 7. doi:10.1371/journal.pgen.1002154.
- Tahiliani, M., Koh, K.P., Shen, Y., *et al.*, 2009. Conversion of 5-methylcytosine to 5-hydroxymethylcytosine in mammalian DNA by MLL partner TET1. *Science* 324 (80-), 930–935. doi:10.1126/science.1170116.
- Tan, L., Xiong, L., Xu, W., *et al.*, 2013. Genome-wide comparison of DNA hydroxymethylation in mouse embryonic stem cells and neural progenitor cells by a new comparative hMeDIP-seq method. *Nucleic Acids Res.* 41, 1–12. doi:10.1093/nar/gkt091.
- Turelli, P., Castro-diaz, N., Marzetta, F., *et al.*, 2014. Interplay of TRIM28 and DNA methylation in controlling human endogenous retroelements Interplay of TRIM28 and DNA methylation in controlling human endogenous retroelements. *Genome Res.* 1260–1270. doi:10.1101/gr.172833.114.
- Venham, N.T., Kennedy, N.A., Adams, A.T., *et al.*, 2016. Integrative epigenome-wide analysis demonstrates that DNA methylation may mediate genetic risk in inflammatory bowel disease. *Nat. Commun.* 7, 13507. doi:10.1038/ncomms13507.
- Walsh, C.P., Chaillet, J.R., Bestor, T.H., 1998. No title. *Nat. Genet.* 20, 116–117. doi:10.1038/2413.
- Wang, L., Zhou, Y., Xu, L., *et al.*, 2015. Molecular basis for 5-carboxycytosine recognition by RNA polymerase II elongation complex. *Nature*. doi:10.1038/nature14482.
- Weber, A.R., Krawczyk, C., Robertson, A.B., *et al.*, 2016. Biochemical reconstitution of TET1–TDG–BER-dependent active DNA demethylation reveals a highly coordinated mechanism. *Nat. Commun.* 7, 10806. doi:10.1038/ncomms10806.
- Wu, H., D'Alessio, A.C., Ito, S., *et al.*, 2011. Genome-wide analysis of distribution reveals its dual function in transcriptional regulation in mouse embryonic stem cells. *Genes Dev.* 25, 679–684. doi:10.1101/gad.2036011.GENES.
- Xu, Z., Taylor, J.A., Leung, Y.-K., Ho, S.-M., Niu, L., 2016. oxBS-MLE: An efficient method to estimate 5-methylcytosine and 5-hydroxymethylcytosine in paired bisulfite and oxidative bisulfite treated DNA. *Bioinformatics* 32, btw527. doi:10.1093/bioinformatics/btw527.
- Yin, Y., Morgunova, E., Jolma, A., *et al.*, 2017. Impact of cytosine methylation on DNA binding specificities of human transcription factors. *Science* 356 (80-), eaaj2239. doi:10.1126/science.aaj2239.
- Yuen, R.K., Jiang, R., Penaherrera, M.S., McFadden, D.E., Robinson, W.P., 2011. Genome-wide mapping of imprinted differentially methylated regions by DNA methylation profiling of human placentas from triploidies. *Epigenetics. Chromatin* 4, 10. doi:10.1186/1756-8935-4-10.
- Yu, M., Hon, G.C., Szulwach, K.E., *et al.*, 2012a. Tet-assisted bisulfite sequencing of 5-hydroxymethylcytosine. *Nat. Protoc.* 7, 2159–2170. doi:10.1038/nprot.2012.137.
- Yu, M., Hon, G.C., Szulwach, K.E., *et al.*, 2012b. Base-resolution analysis of 5-hydroxymethylcytosine in the mammalian genome. *Cell* 149, 1368–1380. doi:10.1016/j.cell.2012.04.027.
- Zerbino, D.R., Johnson, N., Juettemann, T., Wilder, S.P., Flicek, P., 2014. WiggleTools: Parallel processing of large collections of genome-wide datasets for visualization and statistical analysis. *Bioinformatics* 30, 1008–1009. doi:10.1093/bioinformatics/btt737.
- Zhu, L., Lv, R., Kong, L., *et al.*, 2015. Genome-wide mapping of 5mC and 5hmC identified differentially modified genomic regions in late-onset severe preeclampsia: a pilot study. *PLOS ONE* 10, 1–15. doi:10.1371/journal.pone.0134119.

Further Reading

- Bird, A., 2007. Perceptions of epigenetics. *Nature* 447, 396–398. doi:10.1038/nature05913.
- Bock, C., Halbritter, F., Carmona, F.J., *et al.*, 2016. Quantitative comparison of DNA methylation assays for biomarker development and clinical applications. *Nat. Biotechnol.* doi:10.1038/nbt.3605.
- Chen, D.-P., Lin, Y.-C., Fann, C.S.J., 2016. Methods for identifying differentially methylated regions for sequence- and array-based data. *Brief. Funct. Genomics* 15, elw018. doi:10.1093/bfpg/elw018.
- Guy, J., Gan, J., Selfridge, J., Cobb, S., Bird, A., 2007. Reversal of neurological defects in a mouse model of Rett syndrome. *Science* 315 (80-), 1143–1147. doi:10.1126/science.1138389.
- Inoue, A., Jiang, L., Lu, F., Suzuki, T., Zhang, Y., 2017. Maternal H3K27me3 controls DNA methylation-independent imprinting. *Nature*. doi:10.1038/nature23262.
- Kriaucionis, S., Heintz, N., 2009. The nuclear DNA base 5-hydroxymethylcytosine is present in Purkinje neurons and the brain. *Science* 324, 929–930. doi:10.1126/science.1169786.
- Miska, E.A., Ferguson-smith, A.C., 2016. Transgenerational inheritance: models and mechanisms of non-DNA sequence-based inheritance. *Science* 354, 778–782. doi:10.1126/science.aaf4945.
- Robinson, M.D., Kahraman, A., Law, C.W., *et al.*, 2014. MINI REVIEW: Statistical methods for detecting differentially methylated loci and regions. *bioRxiv* 5, 7120. doi:10.1101/007120.
- Wreczycka, K., Gosdschan, A., Yusuf, D., Platform, B., Group, B., 2017. Strategies for analyzing bisulfite sequencing data. *BioRxiv*. doi:10.1101/109512.

Relevant Websites

- <http://www.cegx.co.uk>
Cambridge Epigenetix.
- <https://www.pmggenomics.ca/hoffmanlab/proj/dnamod/>
DNAmoD.
- <http://ihcc-epigenomes.org/>
International Human Epigenome Consortium.
- <https://sequencing.qcfail.com/>
QCFail.
- <https://swiftbiosci.com/>
Swift Biosciences.

<https://www.wisegeneusa.com/5hmc>

WiseGene.

<http://www.zymoresearch.com/services/epigenetic-aging-clock>

Zymo Research.

Biographical Sketch

Malwina Prater is a bioinformatician at the Centre for Trophoblast Research, University of Cambridge. Her previous work at Cambridge Epigenetix (CEGX) involved developing technologies that enable accurate quantification of methylcytosine (mC) and hydroxymethylcytosine (hmC) at a single base resolution. Prior to this, she was undertaking her PhD at Cancer Research UK Cambridge Institute, University of Cambridge. Her work focused on long non-coding RNAs and epigenetic regulation of gene expression. She has also been involved in projects investigating control of protein expression in yeast (University of Edinburgh) and ribosome biogenesis in plants (Goethe University in Frankfurt).

Russell Hamilton heads the bioinformatics core facility at the Centre for Trophoblast Research (CTR), University of Cambridge. The focus of the CTR is the study of the placenta and maternal-fetal interactions during pregnancy and brings together over 25 Principal Investigators based in different departments within the University of Cambridge and Babraham Institute. He has previously worked in industry at Cambridge Epigenetix (CEGX) developing epigenetics tools to enable single base resolution sequencing of methylcytosine (mC) and hydroxymethylcytosine (hmC). Prior to this was a senior postdoc in the Department of Biochemistry at University of Oxford, working on the RNA structural motifs required for mRNA localization and their interaction with trans-acting protein factors. Russell received an MSc in Intelligent Systems from the University of Aberdeen and his PhD in bioinformatics from the University of Edinburgh. At each of his previous positions, Russell has set up bioinformatics infrastructure including pipelines for next-generation sequencing, RNA structure searches, protein structural modelling and a microscope imaging facility.

Comparative Epigenomics

Yutaka Saito, National Institute of Advanced Industrial Science and Technology (AIST), Koto-ku, Japan and National Institute of Advanced Industrial Science and Technology (AIST), Shinjuku-ku, Japan

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

ChIP	Chromatin immunoprecipitation
DEG	Differentially expressed gene
DMC	Differentially methylated cytosine
DMR	Differentially methylated region
ENCODE	Encyclopedia of DNA elements
FPKM	Fragments per kilobase million
GEO	Gene expression omnibus

HiC

High-throughput chromosome conformation capture

IHEC

International human epigenome consortium

NGS

Next-generation sequencing

SRA

Sequence read archive

TPM

Transcripts per million

TSS

Transcription start site

Introduction

The analysis of epigenomic data often involves the comparison with other epigenomic data. For example, consider the analysis of DNA methylation data where a certain genomic locus is detected to have a methylation level lower than its flanking regions. Such a locus is regarded as a candidate of cis-regulatory elements. However, further insights into its regulatory role are difficult to obtain solely from DNA methylation data. In such cases, it is useful to compare DNA methylation data with histone modification data. If the locus has H3K4me3, it is inferred as a promoter that regulates the expression of neighboring genes. Alternatively, the locus may have H3K4me1, suggesting that it is an enhancer regulating the expression of distal genes. Epigenomic data can also be compared between different cell types. For example, consider that the DNA methylation data were collected from stem cells, and are compared with those from differentiated cells. If the locus shows an increased methylation level in the differentiated cells, its function is predicted to be related to the differentiation from the stem cells. These examples clearly demonstrate that the comparison of epigenomic data is a powerful approach in epigenome research.

This article aims to describe conventional methods for comparing epigenomic data. We show practical issues that arise in real-life analysis settings, and explain how to address these problems. A review of widely-used software for various kinds of epigenomic data is provided. We also provide a review of public epigenomic databases that can be used as a reference for the comparison with in-house epigenomic data. Finally, we present a case study where mesenchymal stem cells and mature fat cells are compared using DNA methylation, histone modification, transcription factor binding, and gene expression data. The study provides an example of how to combine each analysis task in constructing the whole analysis for addressing a specific research question.

Background

Epigenetic Modification

The knowledge on epigenetic modification is necessary to interpret the results of epigenomic data analysis. [Table 1](#) summarizes representative epigenetic modifications and their known functions in gene regulation.

Table 1 Epigenetic modifications and their functions in gene regulation

Epigenetic modification	Indication	Experimental technology
H3K4me3	Promoter	ChIP-seq
H3K4me1	Enhancer	
H3K27ac	Activated	
H3K9me3	Heterochromatin	
H3K27me3	Repressed	Bisulfite-seq, Infinium BeadChip
H3K36me3	Active gene body	
DNA methylation, promoter	Repressed	
DNA methylation, gene body	Activated (in some cell types)	

Several kinds of histone modification have established regulatory roles. H3K4me3 and H3K4me1 are histone modifications that indicate promoters and enhancers, respectively. H3K27ac is a hallmark of active chromatin. Genomic loci that harbor both H3K4me1 and H3K27ac are called active enhancers, suggesting that genes regulated by these enhancers are activated. On the other hand, genomic loci that have H3K4me1 but lack H3K27ac are sometimes called “primed” enhancers, meaning that these loci have the potential to function as enhancers although their target genes are still not activated in the cell type of interest. H3K9me3 and H3K27me3 are hallmarks of repressed chromatin. H3K9me3 tends to be observed at heterochromatic regions consistently repressed in a broad range of cell types, while H3K27me3 indicates a temporal repression in specific cell types. As a last example, H3K36me3 is observed at actively transcribed gene bodies.

The function of DNA methylation is more complicated and context-dependent in comparison to histone modification. In general, DNA methylation is considered to modulate the binding affinity of transcription factors either positively or negatively, and its regulatory outcome depends on the function of transcription factors such as activators or repressors. For example, if a transcription factor with an activator function preferentially binds to unmethylated DNA, an increase in DNA methylation will decrease the transcription factor's affinity for that region, thereby repressing gene expression. Despite this complexity, some genome-wide associations are known between DNA methylation and gene expression. DNA methylation at promoters is associated with the repression of gene expression. Inversely, DNA methylation at gene bodies is associated with the activation of gene expression in some cell types.

There are many other kinds of epigenetic modification not included in [Table 1](#). For histone modification, there are a variety of “non-standard” modifications such as phosphorylation, ubiquitination, and sumoylation. See [Zentner and Henikoff \(2013\)](#) for a recent review. While DNA methylation usually refers to cytosine methylation in the CpG context (cytosines followed by guanines), other kinds of DNA methylation are known to be important in some cell types and/or organisms. These include cytosine methylation in the non-CpG context ([Jang et al., 2017](#)) and hydroxymethylation ([Guibert and Weber, 2013](#)). In addition, it is important to be aware that the current knowledge on epigenetic modification has mainly been obtained from studies on model organisms such as mammals and yeasts. Therefore, one should be careful when analyzing epigenomic data from non-model organisms where epigenetic modifications may have unknown functions.

It is common to incorporate transcription factor binding data into epigenomic data analysis even though they are not strictly considered to be epigenomic data. This is because transcription factor binding and epigenetic modification are closely related to each other. For example, the binding affinity of some transcription factors is known to be modulated by DNA methylation, and some transcription factors are known to induce histone modification at the binding sites. Gene expression data are also regularly utilized in epigenomic data analysis. Since a major function of epigenetic modification is gene regulation, gene expression data are useful to analyze the regulatory outcome of epigenetic modification in the cell type of interest.

Experimental Technology and Raw Data Processing

The understanding of experimental technologies for measuring epigenomic data is a prerequisite for selecting the appropriate methods for analyzing the data. [Table 1](#) includes experimental technologies used for various epigenetic modifications. Most kinds of epigenomic data are measured by next-generation sequencing (NGS) or microarray-based technologies. Histone modification and transcription factor binding are measured by chromatin-immunoprecipitation sequencing (ChIP-seq) where DNA fragments bound by a specific protein are extracted by an antibody prior to sequencing. DNA methylation is measured by bisulfite sequencing (bisulfite-seq) where unmethylated cytosines are chemically converted and sequenced as thymines. DNA methylation can also be measured by a microarray-based technology called Illumina Infinium BeadChip. Gene expression is measured either by NGS (RNA-seq) or microarrays.

Raw data generated by these experimental technologies are fastq files for NGS, and platform-dependent files for microarrays (e.g., tab-delimited text files for Agilent microarrays, and CEL files for Affymetrix microarrays). With appropriate data processing, they can be transformed into signal values along genomic positions. For example, DNA methylation data are represented as a methylation level (also referred to as a methylation ratio, a methylation rate, or a beta value) for each cytosine position, which is a value ranging from 0 to 1 that indicates the fraction of methylated cytosines in a cell population. Similarly, gene expression data are represented as an expression level for each gene that is linked to its genomic position. In this article, we focus our discussion on the comparison of epigenomic data, assuming that input data are already represented as signal values. Therefore, the detail in raw data processing such as read mapping falls outside the scope of this article. However, it is still important to be aware of the experimental technology used for measuring the epigenomic data since some downstream analyses need care depending on different experimental technologies. For example, conventional methods for the detection of differentially expressed genes (DEGs) are different for RNA-seq and microarray data. In such cases, we explain the analyses separately for each experimental technology.

Application

In this section, we first describe practical issues in the comparative analysis of epigenomic data. Then, we provide a review of widely-used software for each kind of epigenomic data (The list of all software discussed in this article is given in Appendix). Finally, we provide a review of public epigenomic databases.

Comparative Analysis: Correlation and Difference

A useful method for comparing epigenomic data is to evaluate the correlation between two epigenomic datasets. For example, consider two DNA methylation datasets, each of which represents methylation levels measured from a certain cell type. The correlation coefficient between the methylation levels can be used as a similarity measure between the two cell types.

Another method for the comparative analysis is to detect differences between two epigenomic datasets at specific genomic loci. For example, one can calculate the difference in methylation levels between two cell types at each cytosine position. Cytosines with large differences in methylation levels are called differentially methylated cytosines (DMCs). Methylation differences can be detected not only for a cytosine position but also for a window of a certain length using methylation levels averaged over the region. In such cases, detected regions are called differentially methylated regions (DMRs).

Defining Windows

As shown in the above example of DNA methylation, signal values computed from epigenomic data are often averaged or summed over the windows of genomic regions. Therefore, how windows are defined is an important practical issue. The most general method for defining windows is a sliding-window approach where windows are defined as fixed-length regions (e.g., 1000 bp) with a certain step size (e.g., 500 bp). Another conventional method is to define windows based on their positions relative to genes. For example, fixed-length windows around transcription start sites (TSSs) are often used for analyzing promoters. Windows are sometimes defined using other genomic elements (e.g., retrotransposons) to specifically analyze their epigenomic profiles.

Windows can also be defined using epigenomic data themselves. Some tools are designed to detect DMRs without pre-specifying regions of interest by determining the boundaries of DMRs solely from DNA methylation data. Detected DMRs can be used as windows for analyzing other kinds of epigenomic data such as transcription factor binding. Similarly, one can detect transcription factor binding sites from ChIP-seq data, and use these sites as windows for analyzing DNA methylation.

Biological Replicates

Biological replicates refer to parallel measurements of biologically distinct samples produced from the same experimental condition. It is well known that the comparative analysis of omics data becomes more accurate and reproducible by using replicates. A simple method for utilizing replicates is to average signal values from individual replicates. However, this method fails to incorporate the information of variance among replicates. Most tools for detecting DEGs utilize the variance information for evaluating a statistical significance (i.e., p-value). For DMC and DMR detection, some tools support replicates for evaluating p-values while others do not.

Biological replicates should be distinguished from technical replicates that refer to repeated measurements from the same biological sample. While averaging technical replicates can reduce the experimental noise of measurements, their variance information is not effective to increase the power of statistical testing. This is because the variance among technical replicates does not represent biological variation. See [Blainey et al. \(2014\)](#) for a detailed discussion on handling biological and technical replicates.

Although replicates are useful, their availability is sometimes limited in real-life analysis settings. This is especially true for epigenomic data measured by newer technologies with higher experimental costs. For example, while replicates are quite common for microarrays, they are relatively limited for DNA methylation data measured by bisulfite-seq (e.g., few or no replicates are available). In such cases, one needs to resort to replicate-free analysis where the difference in methylation levels between single samples is used for detecting DMCs and DMRs.

DNA Methylation

Raw data processing of bisulfite-seq is performed by software such as Bismark ([Krueger and Andrews, 2011](#)), BSMAP ([Xi and Li, 2009](#)), and Bisulfighter ([Saito et al., 2014](#)). These tools map bisulfite-seq reads to a reference genome, and calculate a methylation level for each cytosine position based on the mapping results. The detection of DMCs and DMRs is conducted with software such as BSmooth ([Hansen et al., 2012](#)) and ComMet ([Saito and Mituyama, 2015](#)). These tools are capable of *de novo* DMR detection without pre-specifying windows. BSmooth supports replicate-aware analysis but is not applicable to data without replicates, while ComMet supports replicate-free analysis but cannot utilize replicate information. We refer to the review ([Tsuji and Weng, 2016](#)) for other tools available for bisulfite-seq data.

Raw data processing of Infinium BeadChip is performed by software such as IMA ([Wang et al., 2012](#)) and Minfi ([Aryee et al., 2014](#)). These tools compute methylation levels at individual cytosine positions (often referred to as beta values), and average methylation levels for various gene-related windows including promoters and gene bodies. The detection of differential methylation for Infinium BeadChip is usually conducted with these windows using methods originally developed for gene expression microarrays. For example, IMA detects DMCs and DMRs using the limma method ([Ritchie et al., 2015](#)) developed for detecting DEGs. DMR detection without pre-specified windows is not common for Infinium BeadChip data, perhaps due to a limited number of measurable cytosine positions, although some tools are developed for this purpose ([Morris et al., 2014](#)). See the review ([Dedeurwaerder et al., 2014](#)) for other Infinium BeadChip data analysis tools.

DNA methylation data are commonly compared with gene expression data. Since DNA methylation at promoters is associated with gene repression, the inverse correlation between methylation levels and expression levels are used in many studies. Similarly,

DMCs and DMRs between cell types can be compared with DEGs. However, it is important to be aware that the association between DNA methylation and gene expression is evident but not so strong. Therefore, it is often the case that one fails to detect an inverse correlation between methylation levels and expression levels. A practical tip here is to focus on the top-ranked genes. For example, the comparison of methylation levels between the top 10% of genes with the highest expression levels and the bottom 10% of genes with the lowest expression levels will result in a clear difference between the two groups. A similar issue also applies when comparing DMCs and DMRs with DEGs between different cell types.

Histone Modification and Transcription Factor Binding

Read mapping for ChIP-seq is performed by software such as Bowtie (Langmead *et al.*, 2009) and BWA (Li and Durbin, 2009). Signal values for ChIP-seq data are the fraction of mapped read counts in the ChIP sample (i.e., immunoprecipitated by an antibody) divided by those in the input sample (i.e., sequenced without immunoprecipitation). This task is conducted by software such as MACS (Zhang *et al.*, 2008), SICER (Zang *et al.*, 2009), and DROMPA (Nakato *et al.*, 2013). Other tools for ChIP-seq data are reviewed in Nakato and Shirahige (2017).

An important task in ChIP-seq data analysis is peak detection where genomic regions enriched with signal values are determined by comparing read mapping results between ChIP samples and input samples. The above tools for computing signal values also support peak detection. However, it is worth noting that peak detection is a difficult task, and it is often the case that peaks detected by different tools show a strong disagreement in their numbers and positions. This is partly because the characteristics of peaks are different for various DNA-binding proteins: some proteins produce sharp peaks with signal values highly enriched in short regions, while others produce broad peaks showing an intermediate enrichment throughout long regions. Due to this difficulty, signal values are sometimes directly used for downstream analyses without peak detection. Nonetheless, peaks are useful as windows for comparing ChIP-seq data with other epigenomic data.

Histone modification data are frequently compared with DNA methylation data. For example, one can calculate the methylation levels at H3K4me1 peaks for measuring their enhancer activity. Similarly, methylation levels at transcription factor peaks are used for surveying the interplay between transcription factor binding and DNA methylation. The detection of DMRs can also be conducted using ChIP-seq peaks as windows. Since peak detection involves the technical difficulties as explained above, sliding windows are also used instead of peaks. For this purpose, sliding windows with the highest enrichment of ChIP signals (e.g., the top 10%) are used for calculating methylation levels.

Gene Expression

Read mapping for RNA-seq is performed by software such as TopHat2 (Kim *et al.*, 2013) and STAR (Dobin *et al.*, 2013). Expression levels are measured as fragments per kilobase million (FPKM) computed by Cufflinks (Trapnell *et al.*, 2010), or transcripts per million (TPM) computed by RSEM (Li and Dewey, 2011). DEGs can be detected by tools like Cuffdiff (Trapnell *et al.*, 2013), edgeR (Robinson *et al.*, 2010) and DEGseq (Wang *et al.*, 2010), all of which support replicates. Compared to bisulfite-seq and ChIP-seq data, there are numerous tools available for RNA-seq data. See Conesa *et al.* (2016) for the recent review.

Raw data processing of gene expression microarrays is performed by software such as limma (Ritchie *et al.*, 2015) and affy (Gautier *et al.*, 2004). limma supports various types of microarrays produced by Agilent, while affy is a widely-used package for Affymetrix microarrays. limma is also a conventional tool for detecting DEGs with support for replicates.

Gene expression data are used in epigenomic data analysis for evaluating the regulatory outcome of epigenetic modification. For example, when a genomic locus is detected as an enhancer candidate with a high H3K4me1 signal and a low DNA methylation level, its influence on gene expression can be evaluated by RNA-seq or microarray data.

Chromatin State Estimation

One of the goals of epigenomic data analysis is to annotate genomic loci with some “chromatin states”. For example, one can annotate loci with high H3K9me3 signals and DNA methylation levels as the “heterochromatin state”, and loci with high H3K4me1 and H3K27ac signals accompanied by low DNA methylation levels as the “active enhancer state”. Chromatin state estimation is a task for doing such annotations by integrating various kinds of epigenomic data. ChromHMM (Ernst and Kellis, 2012) is a software that is commonly used for chromatin state estimation.

Public Databases

It is effective to incorporate public epigenomic data into the analysis of in-house epigenomic data. For example, even if you only have DNA methylation data, you may find histone modification data measured from the same cell type in public databases, allowing the comparative analysis between the datasets. Epigenomic data published in previous studies are deposited in the Gene Expression Omnibus (GEO) database where raw NGS data (fastq files) can be found by following a hyperlink to the Sequence Read Archive (SRA) database. In practice, it is important to be aware that the word search functionality in GEO and SRA sometimes fails to retrieve relevant data. Thus, it is highly recommended that general search engines (e.g., Google) are used in combination

with GEO and SRA. In addition, it is useful to visit the web portals of international epigenome projects, including International Human Epigenome Consortium (IHEC), Encyclopedia of DNA Elements (ENCODE), Roadmap Epigenomics, and BLUEPRINT Epigenome. These web portals organize various kinds of epigenomic data as a form of the “experiment matrix” where rows and columns represent cell types (e.g., embryonic stem cells) and experimental technologies (e.g., bisulfite-seq), respectively, so that one can easily find epigenomic data of their interest. The URLs of these databases and web portals are given in Relevant Websites.

Case Study

In the previous sections, many tasks in the comparative analysis of epigenomic data were explained separately. In practice, however, these tasks need to be combined for addressing a specific research question. Here, we show such an example using a case study. The study is the re-analysis of the data from [Takada *et al.* \(2014\)](#) where human adipose-derived stem cells (ADSCs) are compared to mature fat cells differentiated from ADSCs (FatCs) based on various types of epigenomic data. The purpose of the study is to find epigenetic markers that characterize the differentiation from ADSCs to FatCs.

Data

For ADSCs and FatCs, DNA methylation and gene expression data were measured by bisulfite-seq and RNA-seq, respectively. These were in-house data produced by the authors of the original study, and deposited in the SRA database with the accession number SRP003529. The authors also used histone modification and transcription factor binding data from FatCs. These were public data measured by ChIP-seq, and were downloaded from the SRA database with the accession number SRP002343. Replicates were not available for these datasets. For bisulfite-seq data, read mapping and the calculation of methylation levels were performed by Bisulfighter ([Saito *et al.*, 2014](#)). Read mapping for RNA-seq data was performed by TopHat2 ([Kim *et al.*, 2013](#)), and expression levels (FPKM) were calculated by Cufflinks ([Trapnell *et al.*, 2010](#)). Read mapping for ChIP-seq data was conducted by Bowtie ([Langmead *et al.*, 2009](#)), and peaks were detected by MACS ([Zhang *et al.*, 2008](#)).

Comparison of DNA Methylation and Gene Expression

We first compared DNA methylation with gene expression in ADSCs. The methylation levels of promoters and gene bodies were calculated for all genes. Promoter windows were defined as regions around the TSSs from -1000 bp upstream to $+500$ bp downstream, while gene body windows were defined as regions from $+2000$ bp downstream of the TSSs to the corresponding transcription termination sites. We then compared FPKM between the top 10% of genes with the highest methylation levels, and the bottom 10% of genes with the lowest methylation levels ([Fig. 1](#)). As expected, genes with highly methylated promoters tended to have low FPKM values, while genes with highly methylated gene bodies tended to have high FPKM values.

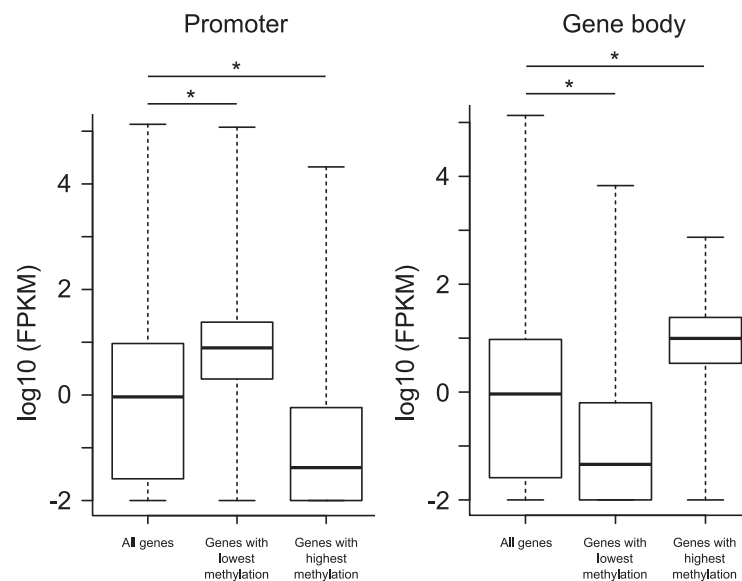


Fig. 1 Comparison of expression levels between the genes with the highest and the lowest methylation levels in ADSCs. Expression levels (FPKM) were compared for all genes, the top 10% of genes with the highest methylation levels, and the bottom 10% of genes with the lowest methylation levels. The analysis was conducted separately for promoter methylation and gene body methylation. *: $P < 1e-100$ (Mann-Whitney U test).

Comparison of DNA Methylation Between Cell Types

We next compared DNA methylation between ADSCs and FatCs using promoter methylation levels. Only small differences in promoter methylation levels were observed between ADSCs and FatCs (**Fig. 2**). For reference, when a similar analysis was conducted between ADSCs and induced pluripotent stem cells (iPSCs), promoter methylation levels were more diverse. This was surprising because when we compared gene expression using FPKM, similar levels of correlation were detected for the two comparisons (Pearson correlation coefficients: 0.79 for ADSCs versus FatCs, and 0.88 for ADSCs versus iPSCs). These results suggest that differential methylation between ADSCs and FatCs occurs at specific genomic loci other than promoters.

Comparison of DNA Methylation at Transcription Factor Binding Sites

To find specific loci harboring differential methylation between ADSCs and FatCs, we used ChIP-seq data for PPAR γ , a transcription factor known to function as a master regulator of fat cell differentiation. PPAR γ peaks were detected in FatCs, and then these peaks were used as windows for differential methylation analysis between ADSCs and FatCs. Interestingly, specific hypomethylation was found at PPAR γ peaks (**Fig. 3**). A similar analysis was also conducted using ChIP-seq data for H3K4me3 that indicates general promoters, but the hypomethylation was not observed. Therefore, these results demonstrated that specific hypomethylation at PPAR γ binding sites represents epigenetic markers for fat cell differentiation.

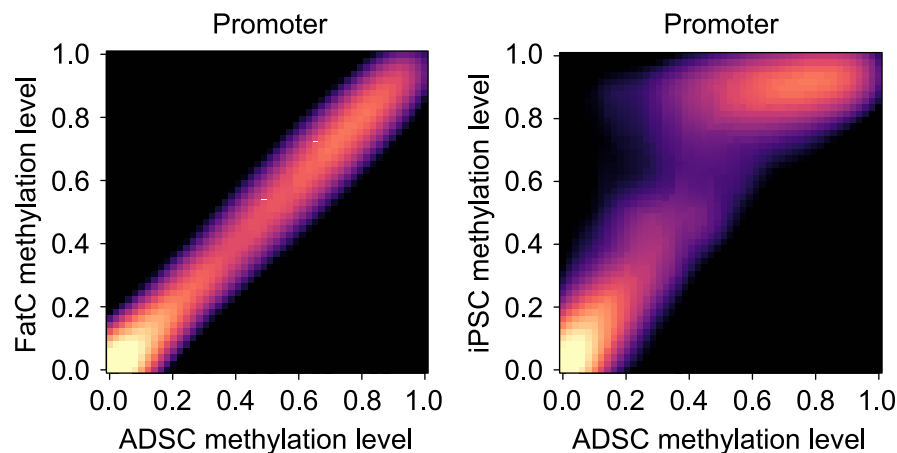


Fig. 2 Comparison of DNA methylation between cell types. Two-dimensional density plot where brighter colors indicate larger numbers of promoters with corresponding methylation levels. (Left) Promoter methylation levels were compared between ADSCs and FatCs. (Right) A similar analysis was conducted between ADSCs and iPSCs.

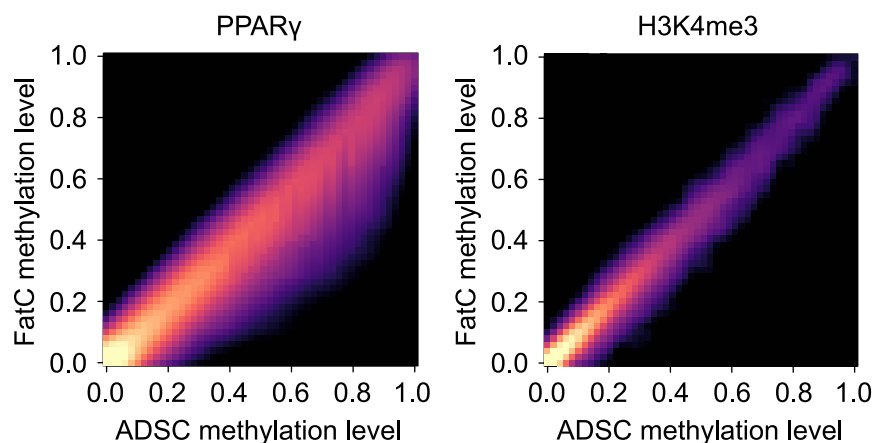


Fig. 3 Comparison of DNA methylation in light of transcription factor binding and histone modification. Two-dimensional density plot similar to **Fig. 2**. (Left) Methylation levels at PPAR γ peaks. The bright colors in the bottom-right of the plot region indicate the hypomethylation of PPAR γ binding sites in FatCs compared to ADSCs. (Right) Methylation levels at H3K4me3 peaks. Hypomethylation was not observed.

Discussion

The comparative analysis of epigenomic data involves several practical issues including how to define windows and how to handle biological replicates. As for replicates, some tools do not support replicates, especially for epigenomic data from relatively new experimental technologies. However, there is a general framework called IDR (Li *et al.*, 2011) that can extend replicate-free methods to replicate-aware versions. Indeed, IDR is extensively used for ChIP-seq data in the ENCODE project where peaks detected from individual replicates are integrated so that reliable peaks showing an agreement among all replicates are retained in the final result.

Another practical issue yet to be addressed in this article is the problem of batch effects. Batch effects refer to the difference detected between data due to artifacts such as those introduced from the laboratory environments (e.g., the difference in the dates of experiments, and/or the difference in the persons who conducted experiments). Batch effects are problematic because they may cover up true biological differences between epigenomic data. ComBat (Johnson *et al.*, 2007) and SVA (Leek and Storey, 2007) are widely-used methods for the detection and the correction of batch effects. The consideration of batch effects is necessary not only for the comparison between in-house and public data but also for the comparison of in-house data measured in different laboratory environments.

Future Directions

The advent of new experimental technologies provides new kinds of epigenomic data. An important example of new epigenomic data is three-dimensional genome structures measured by high-throughput chromosome conformation capture (HiC). See the review (de Wit and de Laat, 2012) for HiC and related experimental technologies. HiC data are especially useful for associating distal enhancers to their target genes. The utilization of such emerging epigenomic data will make comparative epigenomics more powerful in future studies.

Closing Remarks

In this article, we described conventional methods for comparing epigenomic data with a review of widely-used software and databases. There are many possible ways to compare epigenomic data depending on what kind of epigenomic data and which cell type are used in the comparison. An important point is to combine individual analysis tasks to construct the whole analysis in order to reach a specific research goal. The case study using ADSCs and FatCs provided an example of such analyses conducted in a research setting.

Appendix

The summary of software described in this article

Name	Analysis	URL
Bismark	Bisulfite-seq, read mapping, methylation level	http://www.bioinformatics.babraham.ac.uk/projects/bismark
BSMAP	Bisulfite-seq, read mapping, methylation level	https://code.google.com/archive/p/bsmap
Bisulfighter	Bisulfite-seq, read mapping, methylation level	https://github.com/yutaka-saito/Bisulfighter
BSmooth	Bisulfite-seq, DMC/DMR detection	http://bioconductor.org/packages/release/bioc/html/bsseq.html
ComMet	Bisulfite-seq, DMC/DMR detection	https://github.com/yutaka-saito/ComMet
IMA	Infinium BeadChip, methylation level, DMC/DMR detection	https://www.rforge.net/IMA
Minfi	Infinium BeadChip, methylation level, DMC/DMR detection	http://bioconductor.org/packages/release/bioc/html/minfi.html
Bowtie	ChIP-seq, read mapping	http://bowtie-bio.sourceforge.net/index.shtml
BWA	ChIP-seq, read mapping	http://bio-bwa.sourceforge.net/
MACS	ChIP-seq, signal value, peak detection	https://github.com/taoliu/MACS
SICER	ChIP-seq, signal value, peak detection	http://home.gwu.edu/~wpeng/Software.htm
DROMPA	ChIP-seq, signal value, peak detection	http://www.iam.u-tokyo.ac.jp/chromosomeinformatics/rnakato/drompa
TopHat2	RNA-seq, read mapping	http://ccb.jhu.edu/software/tophat/index.shtml
STAR	RNA-seq, read mapping	https://github.com/alexdobin/STAR
Cufflinks	RNA-seq, expression level	https://github.com/cole-trapnell-lab/cufflinks
RSEM	RNA-seq, expression level	http://deweylab.github.io/RSEM

Cuffdiff	RNA-seq, DEG detection	https://github.com/cole-trapnell-lab/cufflinks
edgeR	RNA-seq, DEG detection	https://bioconductor.org/packages/release/bioc/html/edgeR.html
DEGseq	RNA-seq, DEG detection	https://bioconductor.org/packages/release/bioc/html/DEGseq.html
limma	Microarray, expression level, DEG detection, Agilent	https://bioconductor.org/packages/release/bioc/html/limma.html
affy	Microarray, expression level, DEG detection, Affymetrix	https://bioconductor.org/packages/release/bioc/html/affy.html
ChromHMM	Chromatin state estimation from various epigenomic data	http://compbio.mit.edu/ChromHMM
IDR	To extend replicate-free methods to replicate-aware versions	https://sites.google.com/site/anshulkundaje/projects/idr
ComBAT/SVAT	To remove batch effects	http://bioconductor.org/packages/release/bioc/html/sva.html

See also: Epigenetics: Analysis of Cytosine Modifications at Single Base Resolution. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Whole Genome Sequencing Analysis

References

- Aryee, M.J., Jaffe, A.E., Corrada-Bravo, H., *et al.*, 2014. Minfi: A flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays. *Bioinformatics* 30 (10), 1363–1369. doi:10.1093/bioinformatics/btu049.
- Blainey, P., Krzywinski, M., Altman, N., 2014. Points of significance: Replication. *Nat. Methods* 11 (9), 879–880. doi:10.1038/nmeth.3091.
- Conesa, A., Madrigal, P., Tarazona, S., *et al.*, 2016. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17, 13. doi:10.1186/s13059-016-0881-8.
- Dedeurwaerder, S., Defrance, M., Bizet, M., *et al.*, 2014. A comprehensive overview of Infinium HumanMethylation450 data processing. *Brief Bioinform.* 15 (6), 929–941. doi:10.1093/bib/bbt054.
- de Wit, E., de Laat, W., 2012. A decade of 3C technologies: Insights into nuclear organization. *Genes Dev.* 26 (1), 11–24. doi:10.1101/gad.179804.111.
- Dobin, A., Davis, C.A., Schlesinger, F., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29 (1), 15–21. doi:10.1093/bioinformatics/bts635.
- Ernst, J., Kellis, M., 2012. ChromHMM: Automating chromatin-state discovery and characterization. *Nat. Methods* 9 (3), 215–216. doi:10.1038/nmeth.1906.
- Gautier, L., Cope, L., Bolstad, B.M., Irizarry, R.A., 2004. Affy - analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20 (3), 307–315. doi:10.1093/bioinformatics/btg405.
- Guibert, S., Weber, M., 2013. Functions of DNA methylation and hydroxymethylation in mammalian development. *Curr. Top. Dev. Biol.* 104, 47–83. doi:10.1016/B978-0-12-416027-9.00002-4.
- Hansen, K.D., Langmead, B., Irizarry, R.A., 2012. BSmooth: From whole genome bisulfite sequencing reads to differentially methylated regions. *Genome Biol.* 13 (10), R83. doi:10.1186/gb-2012-13-10-r83.
- Jang, H.S., Shin, W.J., Lee, J.E., Do, J.T., 2017. CpG and Non-CpG methylation in epigenetic gene regulation and brain function. *Genes* 8 (6), doi:10.3390/genes8060148. pii: E148.
- Johnson, W.E., Li, C., Rabinovic, A., 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 8 (1), 118–127. doi:10.1093/biostatistics/kxj037.
- Kim, D., Pertea, G., Trapnell, C., *et al.*, 2013. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.* 14 (4), R36. doi:10.1186/gb-2013-14-4-r36.
- Krueger, F., Andrews, S.R., 2011. Bismark: A flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics* 27 (11), 1571–1572. doi:10.1093/bioinformatics/btr167.
- Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L., 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10 (3), R25. doi:10.1186/gb-2009-10-3-r25.
- Leek, J.T., Storey, J.D., 2007. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLOS Genet.* 3 (9), 1724–1735. doi:10.1371/journal.pgen.0030161.
- Li, B., Dewey, C.N., 2011. RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinform.* 12, 323. doi:10.1186/1471-2105-12-323.
- Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25 (14), 1754–1760. doi:10.1093/bioinformatics/btp324.
- Li, Q., Brown, J.B., Huang, H., Bickel, P.J., 2011. Measuring reproducibility of high-throughput experiments. *Ann. Appl. Stat.* 5 (3), 1752–1779. doi:10.1214/11-aos466.
- Morris, T.J., Butcher, L.M., Feber, A., *et al.*, 2014. ChAMP: 450k chip analysis methylation pipeline. *Bioinformatics* 30 (3), 428–430. doi:10.1093/bioinformatics/btt684.
- Nakato, R., Itoh, T., Shirahige, K., 2013. DROMPA: Easy-to-handle peak calling and visualization software for the computational analysis and validation of ChIP-seq data. *Genes Cells* 18 (7), 589–601. doi:10.1111/gtc.12058.
- Nakato, R., Shirahige, K., 2017. Recent advances in ChIP-seq analysis: From quality management to whole-genome annotation. *Brief Bioinform.* 18 (2), 279–290. doi:10.1093/bib/bbw023.
- Ritchie, M.E., Phipson, B., Wu, D., *et al.*, 2015. Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* 43 (7), e47. doi:10.1093/nar/gkv007.
- Robinson, M.D., McCarthy, D.J., Smyth, G.K., 2010. edgeR: A bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26 (1), 139–140. doi:10.1093/bioinformatics/btp616.
- Saito, Y., Mituyama, T., 2015. Detection of differentially methylated regions from bisulfite-seq data by hidden Markov models incorporating genome-wide methylation level distributions. *BMC Genomics* 16 (Suppl 12), S3. doi: 10.1186/1471-2164-16-S12-S3.
- Saito, Y., Tsuji, J., Mituyama, T., 2014. Bisulfighter: Accurate detection of methylated cytosines and differentially methylated regions. *Nucleic Acids Res.* 42 (6), e45. doi:10.1093/nar/gkt1373.
- Takada, H., Saito, Y., Mituyama, T., *et al.*, 2014. Methylome, transcriptome, and PPARγ cistrome analyses reveal two epigenetic transitions in fat cells. *Epigenetics* 9 (9), 1195–1206. doi:10.4161/epi.29856.
- Trapnell, C., Hendrickson, D.G., Sauvageau, M., *et al.*, 2013. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nat. Biotechnol.* 31 (1), 46–53. doi:10.1038/nbt.2450.
- Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28 (5), 511–515. doi:10.1038/nbt.1621.

- Tsuji, J., Weng, Z., 2016. Evaluation of preprocessing, mapping and postprocessing algorithms for analyzing whole genome bisulfite sequencing data. *Brief Bioinform.* 17 (6), 938–952. doi:10.1093/bib/bbv103.
- Wang, D., Yan, L., Hu, Q., *et al.*, 2012. IMA: An R package for high-throughput analysis of Illumina's 450K Infinium methylation data. *Bioinformatics* 28 (5), 729–730. doi:10.1093/bioinformatics/bts013.
- Wang, L., Feng, Z., Wang, X., Wang, X., Zhang, X., 2010. DEGseq: An R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics* 26 (1), 136–138. doi:10.1093/bioinformatics/btp612.
- Xi, Y., Li, W., 2009. BSMAP: Whole genome bisulfite sequence MAPPING program. *BMC Bioinform.* 10, 232. doi:10.1186/1471-2105-10-232.
- Zang, C., Schones, D.E., Zeng, C., *et al.*, 2009. A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics* 25 (15), 1952–1958. doi:10.1093/bioinformatics/btp340.
- Zentner, G.E., Henikoff, S., 2013. Regulation of nucleosome dynamics by histone modifications. *Nat. Struct. Mol. Biol.* 20 (3), 259–266. doi:10.1038/nsmb.2470.
- Zhang, Y., Liu, T., Meyer, C.A., *et al.*, 2008. Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* 9 (9), R137. doi:10.1186/gb-2008-9-9-r137.

Further Reading

- Blainey, P., Krzywinski, M., Altman, N., 2014. Points of significance: Replication. *Nat. Methods* 11 (9), 879–880. doi:10.1038/nmeth.3091.
- Conesa, A., Madrigal, P., Tarazona, S., *et al.*, 2016. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17, 13. doi:10.1186/s13059-016-0881-8.
- Dedeurwaerder, S., Defrance, M., Bizet, M., *et al.*, 2014. A comprehensive overview of Infinium HumanMethylation450 data processing. *Brief Bioinform.* 15 (6), 929–941. doi:10.1093/bib/bbt054.
- de Wit, E.I., de Laat, W., 2012. A decade of 3C technologies: Insights into nuclear organization. *Genes Dev.* 26 (1), 11–24. doi:10.1101/gad.179804.111.
- Guibert, S., Weber, M., 2013. Functions of DNA methylation and hydroxymethylation in mammalian development. *Curr. Top. Dev. Biol.* 104, 47–83. doi:10.1016/B978-0-12-416027-9.00002-4.
- Jang, H.S., Shin, W.J., Lee, J.E., Do, J.T., 2017. CpG and Non-CpG methylation in epigenetic gene regulation and brain function. *Genes* 8 (6), doi:10.3390/genes8060148. [pii: E148].
- Nakato, R., Shirahige, K., 2017. Recent advances in ChIP-seq analysis: From quality management to whole-genome annotation. *Brief Bioinform.* 18 (2), 279–290. doi:10.1093/bib/bbw023.
- Tsuji, J., Weng, Z., 2016. Evaluation of preprocessing, mapping and postprocessing algorithms for analyzing whole genome bisulfite sequencing data. *Brief Bioinform.* 17 (6), 938–952. doi:10.1093/bib/bbv103.
- Zentner, G.E., Henikoff, S., 2013. Regulation of nucleosome dynamics by histone modifications. *Nat. Struct. Mol. Biol.* 20 (3), 259–266. doi:10.1038/nsmb.2470.

Relevant Websites

<http://www.blueprint-epigenome.eu/>
BLUEPRINT Epigenome.

<https://www.encodeproject.org/>
ENCODE.

<http://ihec-epigenomes.org/>
IHEC.

<https://www.ncbi.nlm.nih.gov/geo/>
GEO.

<https://www.ncbi.nlm.nih.gov/sra>
SRA.

<http://www.roadmapepigenomics.org/>
Roadmap Epigenomics.

Biographical Sketch



Yutaka Saito received his Ph.D. in 2012 from Keio University, Japan. He joined the National Institute of Advanced Industrial Science and Technology (AIST), Japan, as a post-doctoral researcher. Currently, he is a tenure-track researcher at AIST. His research interest is the development of bioinformatics methods for analyzing the function of noncoding genomic regions such as noncoding RNAs and epigenetic modifications.

Genetics and Population Analysis

Fotis Tsetsos, Democritus University of Thrace, Alexandroupolis, Greece

Petros Drineas and Peristera Paschou, Purdue University, West Lafayette, IN, United States

© 2019 Elsevier Inc. All rights reserved.

Nomenclature

Common Variant-Common Disease (CDCV) A hypothesis that common variants must be in the genetic background of disease of high prevalence in a population.

Copy Number Variant (CNV) A structural variation within DNA where sections of the genome have been deleted or duplicated in comparison to a reference genome.

Genetic architecture A collective term encompassing the number, frequencies, and effect sizes of causal alleles for a complex phenotype.

Genome Wide Association Study (GWAS) A study that evaluates the genomes of a large population, usually in search of disease-linked genes.

Haplotype An array of alleles that are transmitted together, hence highly-correlated, in a population.

Hardy-Weinberg Equilibrium (HWE) The law that states that allele and genotype frequencies in a population will remain constant from generation to generation in the absence of other evolutionary influences.

Heterozygosity The proportion of heterozygous individuals in each population simply by counting alleles.

Identity by Descent (IBD) The state where two individuals carry the same genetic segment because they have inherited it from a common ancestor.

Linkage Disequilibrium (LD) The non-random association of alleles at different loci in a given population.

Mendelian Following Mendelian inheritance, namely single-gene characteristics that often run in families.

Microarray A sequencing technique involving attaching DNA or RNA fragments to a slide or membrane.

Next Generation Sequencing (NGS) The collective term for modern DNA sequencing technology that can sequence millions of DNA fragments at once in a short amount of time.

Principal Component Analysis (PCA) A technique used to emphasize variation and bring out strong patterns in a dataset by reducing its dimensions.

Quality control (QC) Procedure that examines the data and ascertains its validity, by filtering out substandard data.

Reticulate evolution The concept of the partial merging of lineages that leads to genetic relationships better represented by a network.

Single Nucleotide Polymorphisms (SNP) A type of SNV defined by occurring with high frequency within a population without occurring in the majority of individuals.

Single Nucleotide Variants (SNV) Single Nucleotide Variant. A variant from the reference genome that only differs by a single base.

Whole Genome Sequencing (WGS) A type of NGS which sequences the entirety of the sample genome, both coding and non-coding regions.

Introduction

Population genetics can be defined as the study of patterns of genetic variation to explain genetic divergence within and between different groups of organisms. Its principal methodology includes genetic assessment both across populations and among individuals, and investigating the mechanics of the influence exerted by evolutionary factors on genetic variation.

The early attempts at studying genetic variation were limited to the phenotypes of traits that were clearly heritable, with the most popular trait being the blood groups. ABO, the most prominent blood group classification system, was identified by [Landsteiner \(1900\)](#), leading to a Nobel Prize award in 1930 for this discovery. Incidentally, Landsteiner did not recognize the heritable identity of the trait, which was identified by [Dungern and Hirschfeld \(1910\)](#). [Hirschfeld and Hirschfeld \(1919\)](#) performed the first ever study of human genetic variation using the ABO grouping system as a variable. Later on, molecular techniques enabled the use of genetic markers to identify genetic variation, albeit at a very limited scale. The technological advancements of the last decade have enabled us to efficiently examine large amounts of genetic markers using microarrays or by means of Whole Genome Sequencing (WGS), so that we can affordably and efficiently assess the genome-wide variation patterns to investigate the distribution of variation within and between populations. With these recent staggering technological leaps, we have also witnessed a shift towards the examination of rare variants. Furthermore, the incorporation of ancient genetic data at a large scale has led to the formation of the newly created fields of archaeogenomics and palaeogenomics.

The current scientific era is inundated with genomic data, aimed to be used towards the identification of the genetic background of disease. The alleles that get associated with disease are an important part of the total genetic variation distributed in and across populations, and as such, are influenced by demographic events and natural genetic forces. Population genetics can also supply robust models for hypotheses concerning disease pathogenesis, which can effectively steer medical efforts towards disease prevention and treatment. Studies of linkage disequilibrium patterns in disease gene identification, combined with our knowledge of the human genetic variation, can effectively aid us in our search for disease-causing genes, providing us with deep insight for the investigation of genetic trait mechanics and heritability.

In this article, we provide an overview of the software tools most commonly used in population genetics that are backed by robust algorithms, a description of the steps used when applying them to avoid pitfalls and analytic biases, along with assessment cautions to avoid interpretation biases, but also a brief recital of the fundamental concepts and perspectives of population genetics.

Quantifying Genetic Variation and Perspectives

Population genetics is a discipline that aims to address the issue of inter- and intra-population differences. Hence, it is important to set some boundaries of perspective, by citing some cases that portray the spectrum of genetic differences.

On the one end of the spectrum, identical twins differ, at least at the conception level, at practically none of their DNA base pairs. Any pair of unrelated humans differs at approximately one in a thousand base pairs, being approximately 99.9% identical at the DNA level. From this rate, it can be extrapolated that for any haploid sequence, there are approximately 3 million variants, as indeed evidenced by [Prado-Martinez et al. \(2013\)](#). The latest estimates by the [1000 Genomes Project Consortium \(2015\)](#) have revised that number to the range of 4–5 million, depending on the internal variation of the studied population.

Compared with the chimp, humans are approximately 99% similar for aligned DNA-bases, and by including structural variants, the percentage is reduced to approximately 95% similarity. Interestingly, humans present significantly reduced variation than the other major grade ape species, and can be considered to be relatively lacking in genetic diversity as showcased by [Prado-Martinez et al. \(2013\)](#).

So naturally, our focus shifts next to inquiring about the percentage of the variation shared among major human groups in different geographical continents. [Xing et al. \(2009\)](#) addressed this inquiry by studying four major well-populated regions of the world, namely Africa, Europe, East Asia, and India. They reported that approximately 80% of relatively common variants are seen in all four of these groups, while 88% are seen in at least three groups, and more than 90% are shared in at least two groups. More variation is present in the African samples, with almost seven percent of the common variants seen only in the African subset, while only half percent were seen in any non-African group, which is a commonly observed feature in population genetics.

Genetic variation is most commonly partitioned based on the nature and the allelic frequency of the variants. The [1000 Genomes Project Consortium \(2015\)](#) released the final version of their very dense public WGS panel of world-wide reference populations, cataloguing an impressive amount of genetic variants present in the human genome. The overwhelming majority of the catalogued variants affect one nucleotide, hence called Single Nucleotide Variants (SNVs). Designation of a variant as common or rare is not solidly consistent, with the most common threshold of classification being an allelic frequency of 1%.

Common variants are an invaluable tool for genetic investigative efforts, since they facilitate the detection of constraints in their flow. Rare variation proves to be more challenging to assess properly. The study by [Sudmant et al. \(2015\)](#) exploring the distribution of rare variants, showed that for major continental populations, the great majority of SNVs were found to be population-specific. For frequencies less than two percent, fewer than five percent of alleles are actually shared between any two pairs of continents, which is intuitively sensible because these rare variants arose relatively recently. In the case of rare allele analysis in our genetic studies and our disease-related studies, we need to consider that most rare alleles will tend to be population-specific rather than shared across populations.

Fundamental Concepts Overview

In order to analyze and interpret genomic data, a deep understanding of the structure of the genome is needed, as well as the forces shaping its variation and its function.

In the middle of the previous century, [Wright \(1955\)](#) identified four major factors of evolution, namely mutation, natural selection, genetic drift and gene flow. Mutation is the main force that creates variation in the genome. Natural selection acts by selecting in favor of variants that are beneficial, and selecting against variants that are harmful. Genetic drift introduces the stochastic element in evolution, with populations of limited size presenting significant changes over time in allelic frequencies. And lastly there is gene flow, the transmission of genetic material from one population to another which can be regarded as the homogenizing force. The result of these factors is the complex landscape of individual-level, population-level, and species-level variation.

Landmark studies by [Roach et al. \(2010\)](#) and [Conrad et al. \(2011\)](#) have recently been able to directly estimate the human mutation rate by investigating WGS in a human family, comparing parents and offspring and discovering that the human mutation rate from generation-to-generation is in the order of approximately one in a hundred million base pairs per generation, meaning we transit with each gamete approximately 30–35 new DNA variants. That estimate has recently been confirmed in a number of subsequent studies so we can assert that we have an accurate estimate of the rate at which new variation enters the human genomes. The rest of the factors are heavily influenced by geography and demography, and their elucidation requires population genetics analysis.

Linkage Disequilibrium

Given the trans-generational dynamic state of the genome, it is important to infer loci that tend to accompany each other across generations and shape heredity. That inference can be performed by investigating the correlation between alleles and their associations. An array of alleles in different loci that are non-randomly associated with each other is termed a haplotype.

Linkage Disequilibrium (LD) is a term coined to describe the non-random association between alleles of genetic loci that lie in proximity to each other. Initially described by [Lewontin and Kojima \(1960\)](#) the term was an obscure part of genetics, until its resurface in modern genetics with the analysis of large-scale genomic data. The LD between two loci, A and B, can be quantified by the D_{AB} metric. If we define as $P_{A_1B_1}$ the frequency of haplotype A_1B_1 , P_{A_1} as the frequency of allele A_1 , and P_{B_1} as the frequency of allele B_1 , then D_{AB} is defined as:

$$D_{AB} = P_{A_1B_1} - P_{A_1}P_{B_1} \quad (1)$$

Based on this equation, when D_{AB} equals to 0, then the alleles are in linkage equilibrium, thus are in statistical independence, much similar to Hardy-Weinberg Equilibrium (HWE) premises. For its correct interpretation for non-zero values, the D_{AB} metric should then be normalized to accurately portray LD, since D_{AB} possible values are heavily influenced by the allelic frequencies. [Lewontin and Kojima \(1960\)](#) used the D'_{AB} statistic, which is the ratio of D_{AB} to its maximum possible absolute value. This metric does provide accurate results in the case of similar allelic frequencies, but could be inflated in the case of small sample number or if one of the alleles is rare.

Another very popular normalization method is the transformation of D_{AB} to the correlation coefficient r^2 , by dividing the square of D_{AB} by the allelic frequencies in both loci. This equation is more sensitive for rare alleles, but could prove to be underpowered when detecting haplotypes that may contain rare alleles.

$$r^2 = \frac{D_{AB}^2}{P_{A_1}P_{A_2}P_{B_1}P_{B_2}} \quad (2)$$

Geneticists most often rely on both metrics to get a more complete reading on the linkage disequilibrium between two loci. It is worth stating that these descriptive statistics offer no explanation on the reason for these non-random associations between alleles, and a meaningful interpretation should include at least these two statistics, since each statistic has its own advantages. These equations can also be extended to include multiple loci for investigation, instead of only two. However, their practicality is limited, and regional LD is most commonly calculated with a pairwise model.

Haplotypic Phase

Regional LD analyses through D_{AB} reveal the extent of regions with LD-clustered SNVs, which are commonly referred to as haplotypic blocks. Haplotypic blocks are in essence the estimates of the haplotypic phase of the corresponding real genomes.

The use of haplotypic data is an excellent approach to maximize the information gained from genotyping and sequencing experiments, since the mechanics of heredity are largely powered through the transit of a single chromosome out of every chromosomal pair from each parent. Experimental methods to produced haplotypic phased data are still not adequately efficient and cost-effective, thus the haplotypes have to be estimated through statistical procedures from the genotypic data. The genotypic data produced are unphased, meaning that at heterozygosity there is no direct assumption on the parental derivation of each allele.

Haplotypic phase estimation can be performed by some well-established algorithms. Because of the statistical nature of the estimation, there are limits to the confidence levels for the haplotypic phase of the genotypes. The accurate estimation of haplotypes from high-throughput genomic data is an active field of computational and genetic research that is constantly improving. We will outline the methods used to identify haplotypes in a subsequent section of the article.

Analytic Methods and Strategies

The technological advents of the modern era have produced a wealth of data in the field of genetics. As with other big data fields, a main goal is the statistical exploration of the data and the inference of a meaningful interpretation. In this section we will outline the methods currently employed by population geneticists to investigate population genetics data.

Exploratory data analysis techniques are well established in genetics studies. In the case of population genetics, we mostly focus on graphical techniques, with one of the most important techniques being the ones pertaining to dimensionality reduction. Model-based clustering methods are equally well represented, with a variety of established algorithms used extensively. Of course, for the techniques to be used successfully, the datasets should be well maintained, with appropriate quality control steps that will minimize false-positive results and guide towards a meaningful interpretation of the data.

Quality Control

For quality control purposes, PLINK is an invaluable suite of software tools for the manipulation of genetic data, originally developed by [Purcell et al. \(2007\)](#) and further refined and augmented by [Chang et al. \(2015\)](#). It introduced the most popular format for storing and analyzing genotypic data, the binary PED format, providing high compression along with great computational efficiency.

In genome-wide population genetics, it is usually not feasible to generate all data from a specific institution, so in the majority of cases the bulk of the data derives from a variety of sources that are publically available, or available upon request. Specific care should be applied when merging datasets of different origin, since platform and batch differences could lead to false-positives in the analyses that could compromise our results. This necessitates specific quality control steps to avoid batch effects.

Merging multiple-origin datasets requires a streamlined procedure, since the merging process attends to two datasets at a time. The recommended routine is to first aim to merge together all the datasets that have derived from the same technology, e.g., datasets that have been genotyped on the same microarray chip, or datasets that have been sequenced using the same method. Even when merging datasets generated from the same technology, there are bound to be some markers that are overlapping, but with different annotation details, such as genomic mapping position. In that case, the recommended procedure is their removal, since this error most commonly is a result of a specific marker mapping to multiple regions in the genome. Then a batch effect quality control should be performed on these supersets separately, and then repeat the previous steps, with this step aiming to merge together supersets of each technology.

Avoiding batch effects can be a thorny endeavor, in a field that aims to identify cross-population variation. The study of the genetic background of population *de facto* includes the assumption of genetic flow barriers, so instinctively HWE cannot be utilized in studying batch effects, as the case would be in a common GWAS analytic protocol. In turn, we employ genome-wide missingness patterns to infer batch effects. First, a genome-wide dataset-spanning call rate threshold of 95% should be used for all markers, removing samples that fail to cross that threshold, in order to achieve a basic level of homogenization in our data. Then, an Identity-by-Missingness (IBM) matrix should be constructed, which is very similar to the Identity-by-Descent (IBD) matrix but differs in that the metric used is similarity in missingness patterns. Its visualization by using a heatmap then enables the identification of missingness clusters that could bias results in downstream analyses. In the case of cluster formation, the recommended routine is to perform of the same call rate threshold of 95% to the clustered sample subgroup and the removal of the culprit markers.

In population genetics, geographical and qualitative regularity in sampling is crucial. Even though researchers apply extensive care into avoiding irregularities, it is possible that sampling mishaps do occur. The construction of an IBD matrix enables the investigation of cryptic close genetic relationships between samples, e.g., relatives, that could confound the results, especially in the case of IBD segment detection. Another common phenomenon is the appearance of genetic outliers. These outliers should be examined carefully to validate their outlier status. Population genetic analysis is based on the average of the genetic background in a specific region, and it is possible that some rare variation could create outliers out of otherwise fine samples that just happen to have an inversion that distinguish them from other samples. Specific areas that are prone to structural variation are relatively well known, such as the 17q21 inversion, which population genetics is a separate subject in its own. Inclusion or exclusion of such areas from the analysis depends on the aim of the analysis.

It is also worth considering that the type of analyses that will be performed is highly dependent on the density and the number of markers. For dimensionality reduction methods, as well as modelling-based clustering methods, it has been shown by [Patterson *et al.* \(2006\)](#) that the number of markers to distinguish between different populations is inversely proportional to the genetic distance between the population. A useful metric for genetic distance is the F_{ST} , which will be described in the next sections.

Quality control is a very important aspect of population genetics, but its specifics are versatile, based on the aim of the analysis and the regional characteristics of the samples. Thus, it is strongly recommended that the aims of a population genetics analysis are extensively laid out before formulating a quality control strategy.

Dimensionality Reduction Methods

Traditional methods of phylogenetic analysis relied on data of low dimensionality. Current technology has since enabled the output of high-dimensional data, that are projected to scale upwards as time and technology progress. Dimensionality reduction methods facilitate modern genetic analysis and interpretation by converting large-scale genotypic data to a low number of elements. In this section we will cover the Principal Component Analysis (PCA) method, the most robust and representative method for dimensionality reduction, as well as some of its modifications that suit more specialized analytic protocols.

PCA has been in use in genetics for a long time, since the middle of the century, with [Cavalli-Sforza *et al.* \(1994\)](#) using it extensively in their landmark genetic studies. Notably, its earliest uses were on low dimensional data, condensing phenotypic, as well as genetic information. The technological advances of the past recent years has since shifted to analyzing large scale individual-level data. PCA was shown to be an excellent method to reduce the dimensionality of high throughput genetic data by [Patterson *et al.* \(2006\)](#), who developed the EIGENSOFT software suite to enable the easy application of the method on various popular genomics data formats.

PCA on genomic data can be affected by highly correlated markers and long-range high LD genetic regions. An efficient way to overcome such hurdles is to perform a pruning step to only keep SNPs that are relatively independent from each other in the genome. To avoid highly correlated markers, a simple implementation of a pruning step by specifying a window size of 200 SNPs, a window step of 5 SNPs, and a LD correlation r^2 value of 0.1 is sufficient.

Usually the genomic matrices that are used as input for PCA are nearly complete, with very low missing data. There are instances in population genetics, where specific samples cannot be fully covered, such as in the case of samples that are genotyped on specific markers, or in the case either of ancient and partially degraded nucleic samples. In such cases, these samples can be projected onto the eigenvectors calculated from the complete datasets. The best practice for PCA projection is using a least-squares approach to project the samples with missing data onto the calculated eigenvectors, so that the missing data will not be replaced by the average of the allelic frequencies in the reference data. This approach presents one drawback, referred to as shrinkage. Shrinkage is a notable phenomenon that occurs when solving by using the least-squares approach. An approximate method to efficiently address this problem was also implemented in the latest version of EIGENSOFT, which enables the more accurate projection of data when using the least squares approach.

The popularity of PCA for genetic analysis spawned many algorithms and methods to assess genetic data. Some notable modifications of PCA were designed to run on haplotypic phased data to also infer local ancestry. Haplotype resolving and haplotype-based methods will be described in the following sections, but it is worth mentioning such methods in this section, as the algorithms behind them are based on PCA. This different approach of PCA was utilized by [Brisbin *et al.* \(2012\)](#), in their software package PCAdmix. It is of note that PCA is only a part of this approach, and another step using a forward-backward HMM algorithm to assign ancestry is implemented. They proposed an algorithm that identifies LD-blocks of genetic data, used them as a guide to create genetic bins and perform PCA on every bin to identify patterns of shared inheritance. Using the forward-backward algorithm, PCAdmix is able to calculate the posterior probability for the shared ancestry for each bin by calculating the distance between populations.

PCA is a fast, non-parametric method of identifying population structure and substructure. However, PCA by itself does not assign ancestry to samples and is mostly used for a graphical representation of the data. Ancestry assignment can be achieved by using statistical clustering methods on the output of PCA.

Modelling-Based Methods

Genetic clustering methods can be roughly subdivided into two major groups, the modelling-based and the distance-based methods. Modelling-based methods use population genetic models to classify individuals into potential ancestry groups. Their inherent assumption is that the observations in each cluster are randomly drawn in a parametric model, and they aim to jointly infer the model parameters along with the cluster membership coefficients. Since they are based on population genetic models, they are traditionally more computationally expensive than PCA. Each computational run is based on the premise that each individual is the result of admixture from a number of different hypothetical ancestral populations. Based on this model, they produce an array of coefficients for each individual. Each coefficient is a quantified estimate for a calculated genetic contribution of a hypothetical ancestral population. The number of hypothetical ancestral populations is finite for each run of the algorithm, and will henceforth be referred to as the K value.

These individual membership coefficients for each ancestral population are probabilistic estimates and are translated into admixture proportions. These admixture proportion estimates for each individual are stored in matrices, referred to as Q matrices. These Q matrices can then be used to visualize and identify clusters of individual ancestries. Many approaches have been developed for this type of analysis, mainly differing on the statistical method employed, or their approximations. The choice of software depends on the nature of the constraints of the analysis, be it efficiency, resources, or depth of genome coverage.

[Pritchard *et al.* \(2000\)](#) developed STRUCTURE, the first software reported to use a model-based clustering method to identify population structure. STRUCTURE uses a bayesian approach and a Markov Chain Monte Carlo (MCMC) algorithm to sample the posterior distributions. The last extensions of the original STRUCTURE software were described by [Hubisz *et al.* \(2009\)](#), leading to the current version of STRUCTURE. The models employed by this version also allowed STRUCTURE to account for LD between markers. STRUCTURE was a novel approach to identifying population structure, and was met with great success when investigating strong population structure, though it presented significant flaws when population structure was weak. That success and shortcomings of STRUCTURE created a surge of ideas to increase accuracy and to expedite computational speeds, since the procedures followed by STRUCTURE were computationally intensive, and the number of the markers available for analysis were ever-increasing.

ADMIXTURE was developed by [Alexander *et al.* \(2009\)](#) as an alternative computational method, based on the likelihood approach employed by STRUCTURE. ADMIXTURE uses a maximum-likelihood approach instead of sampling the posterior distribution, employing a block relaxation algorithm to update the Q matrix. By combining a fast block relaxation strategy and a quasi-Newton convergence acceleration, while opting to compute standard error estimates rather than confidence intervals, ADMIXTURE achieves computational speeds up to two magnitudes greater than those of STRUCTURE, while being more accurate, especially in the case of weak structure.

Also responding to the need for better computational speeds and accuracy, [Raj *et al.* \(2014\)](#) described a new approximation method to accelerate and improve STRUCTURE, in a new software package named fastSTRUCTURE. FastSTRUCTURE introduces two new computational solutions that serve different analytic needs, the simple prior and the logistic prior. The simple prior can be used as a first investigative step, as computations with it are significantly faster, on par with those of ADMIXTURE. The logistic prior can then be used on specific K s to identify subtle and weak population structure.

ADMIXTURE succeeded in significantly speeding up the runtime, in the order of $O(K^2Ln)$, where L is the number of loci and n the number of individual samples. A newer method, sNMF gained tract in the recent years, due to reducing the complexity to the order of $O(KLn)$, while being competitively accurate. sNMF is a method developed by [Frichot *et al.* \(2014\)](#) who used a sparse non-negative matrix factorization (NMF) algorithm and a least-squares optimization approach, an idea based on the theoretical connections between PCA, likelihood and NMF approaches. This method is able to compute admixture proportion estimates with similar accuracy to ADMIXTURE and STRUCTURE, while also being ten to thirty times faster. Additionally, sNMF appears to be adept at analyzing inbred lineages of samples, achieving more accurate results than its competitors. The rewards of using sNMF's algorithm in large datasets can be fully reaped when large numbers of K are considered.

Notably, these algorithms infer global scale ancestry, that is, they produce whole-genome estimates of ancestry, and do not identify the ancestry of particular genomic segments. To that end, the Chromopainter/fineSTRUCTURE approach was developed by [Lawson *et al.* \(2012\)](#). This approach involves the painting the chromosomes of each individual, creating a coancestry matrix by

estimating the genetic relationships between all pairs, and using the matrix to infer groups. Practically, it requires the input of haplotypes instead of markers, which can be calculated by specialized software, such as BEAGLE developed by Browning and Browning (2007), SHAPEIT by Delaneau *et al.* (2013), fastPHASE by Scheet and Stephens (2006) or EAGLE by Loh *et al.* (2016). FineSTRUCTURE has two methods of operation, based on utilizing LD information or not. The unlinked method can outperform STRUCTURE in some cases when datasets with a large number of markers are used, but the core of fineSTRUCTURE's strength lies on its linked method, which has the power to identify subtle differences in population groups with finer population structure.

As with PCA-based methods, strong LD could be detrimental for these analyses. It is recommended to remove strongly linked markers, by specifying a genomic window, in the same fashion that was described for the PCA-based methods, albeit with a more permissive r^2 threshold. Care should be taken when setting LD thresholds, since most of those methods do not explicitly account for LD between markers. The authors of the methods often opt to conceptually partition LD based on its demographic implications. Hence, the LD attributable to ancestry differences between individuals is termed as *mixture LD*, the LD differences caused by recent admixture *admixture LD*, and the rest that is influenced by population history *background LD*. LD pruning can limit background LD, but again, if the aim is the intra-population investigation of genetic history, then it could prove to be detrimental for the analysis.

In the case of missing or masked genotypic data, both ADMIXTURE and sNMF fare very well when dealing with missing or masked genotypes, even in the presence of relatively high (<20%) genotypic missingness. Nevertheless, keeping the missingness thresholds to GWAS standards (<2%) is recommended for all of the aforementioned model-based methods.

Identity by Descent-Based Methods

IBD describes identity through common descent, although the IBD segments are not necessarily identical through the rise of mutations during transgenerational transmission. Since the number of mutations and events amassed in each segment is directly proportional to the generational distance to the common ancestor, that information can be leveraged to infer genetic distance. Analyzing IBD sharing patterns can provide insights into past demographic events, migrations and population size fluctuations.

The longer the IBD segments identified between individuals, the more recent the genetic relatedness among them. Shared IBD segments tend to be rare, but when analyzing large datasets, there is a higher probability of detecting them, and those segments can be relatively long. In a dataset with n individuals, the number of possible IBD sharing pairs is $\binom{n}{2}$. The number of samples needed for that type of analysis require fast and computationally efficient algorithms, so that these analyses can become feasible.

The IBD calculation method implemented by Purcell *et al.* (2007) has the advantage of being a fast method, that does not require phasing of the genotypic data. It implements a hidden Markov model for IBD to determine the posterior probabilities. However, it does not account for LD between markers and requires a previous LD pruning of the data, which can lead to a severe loss of information.

GERMLINE was developed by Gusev *et al.* (2009) and used a novel hashing and extension algorithm to detect IBD segments. This haplotypic dictionary approach greatly speeds up IBD detection, with the runtime being analogous to the number of samples, far reducing the amount of computations required by the theoretical model described above.

Browning and Browning (2013) developed a series of algorithms for efficient and accurate IBD detection, with the latest, most improved one being BEAGLE's RefinedIBD algorithm. The algorithms preceding RefinedIBD were computationally intensive and solely relied on hidden Markov models and posterior estimations. RefinedIBD investigates IBD sharing in two steps, one that incorporates GERMLINE's method, and a second refined step that performs the evaluation of the candidate segments produced by the previous step through a probabilistic approach. Due to its nature, RefinedIBD can deliver the highest accuracy of the aforementioned algorithms, while also being computationally efficient on large datasets.

In the case of datasets comprising of unrelated individuals, IBD sharing detection can produce a wealth of information about the demographics of the included populations as well as the evolutionary forces that act on specific areas in different population groups.

Distance-Based Methods

Distance-based methods attempt to measure descent, and they are based on the concepts behind IBD. These methods can be utilized to calculate the distances between all pairs of individuals to produce a population pairwise distance matrix, that can then be further manipulated into a graphical tree-based output. The phylogenetic concepts behind the distance-based methods can be difficult to grasp by an aspiring population geneticist, but they are the fundamentals of population genetics. Here we attempt a brief outline of the methods and their history.

Wright (1921) proposed the first method for distance measurement, the inbreeding coefficient. This measure was later extended to become the F -statistics, a well-established family of statistical methods for measuring population differentiation and the calculation of genetic distances. Introduced in the middle of the previous century independently by Wright (1949) and Malécot (1948), F -statistics are measures of differentiation, statistics used to calculate the amount of variation in the entire population that is attributable to population differences and subdivision. F -statistics can provide valuable insights into population differentiation and due to that they became the most applied methods in population genetics, leading to the spawning of an abundance of related statistics.

Several definitions exist for the F -statistics, and can be partitioned into three main coefficients. The most popular one is the F_{ST} , also referred to commonly as θ . It is defined as the proportion of genetic diversity due to cross-population difference in allelic frequencies, and is dependent on the deviation of the genotypic frequencies from the HWE, assuming linkage equilibrium.

The computational estimation of F_{ST} is complicated, and has been the subject of many published theoretical studies. A simplification of the estimator of F_{ST} was described by [Weir and Cockerham \(1984\)](#). For this simplification to be theoretically valid, a large number of populations are needed, with each population having an large sample size equal to that of the other populations. These assumptions lets us consider that $1/r$ and $1/\bar{n}$ can be ignored from the full formula, leading to this simplified form:

$$F_{ST} = \frac{\sigma_p^2}{\bar{p}(1 - \bar{p})} \quad (3)$$

where σ_p^2 is the variance of the allelic state and $\bar{p}$ is the total average allelic frequency. This simplified form is rarely the one used, with most published estimates of F_{ST} using the full estimation formula. The estimator by [Weir and Cockerham \(1984\)](#) is the most commonly used estimator out of a series of published estimators by different authors, with each possessing different strengths and shortcomings. Markedly, the EIGENSOFT package implementation uses the estimator proposed by [Hudson *et al.* \(1992\)](#), with [Bhatia *et al.* \(2013\)](#) arguing in favor of its robustness to sample size fluctuations and it not leading to overestimations of F_{ST} .

These F_{ST} distances can then be incorporated and fitted to a tree. The inception of the tree-based representation can be traced to the studies of ([Cavalli-Sforza and Edwards, 1967](#)), that used F_{ST} distances to infer phylogenetic trees.

[Patterson *et al.* \(2012\)](#) built on the ideas of the F -statistics to develop a new distinct set of statistics, confusingly named also f -statistics -albeit with a lower-case f for notation- published as part of the ADMIXTOOLS package. This set includes a host of methods for distance measurement, the two population test f_2 , the three-population test f_3 , the four population test f_4 and the qpGraph. They were developed in a quest to infer treeness, admixture and its proportions, founder populations, and complex past demographic events. These methods were extremely successful, inspiring the publication of a series of high-impact papers on the study of modern and ancient population genetics. In brief, f_2 can be considered as a genetic drift measure, f_3 as an admixture test and f_4 as a test of admixture proportions and directionality. Before analyzing for admixture, it is important to test our populations with a treeness test, because in the case of complex admixture events, it is often very difficult to describe past demography with a phylogenetic tree. Even in the case of treeness, the proposed tree should be fit to the data with a certain confidence.

The qpGraph is an admixture graph fitting method that shares similarities with the TreeMix method. TreeMix was developed by [Pickrell and Pritchard \(2012\)](#) and constructs phylogenetic trees by incorporating migratory models. The main difference between those two methods is that while TreeMix explores the possible model space to identify the one that best describes and fits the data, qpGraph uses a proposed model and tests it to investigate its fit on the data. Those migratory models produced by TreeMix, can provide insights into possible admixture events, but provide no info for a proposed date, they are only suggested based on the allowances set by the researcher. Methods that can estimate and date migratory events have been also developed and methodologically constitute a group of their own.

Admixture Dating Methods

Admixed individuals carry the haplotypes of their ancestors, which can be identified by the investigation of LD patterns, namely ancestry tracts. Throughout the generations, the ancestry tracts in the genome can be broken and intercepted by recombination. Thus, in the case of admixed populations or individuals, it is plausible to infer admixture dates by investigating the length distributions of the ancestry tracts.

Dating admixture events between populations was traditionally practiced by studying historical evidence and subsequently reverse engineered to investigate genetic patterns in the data. High density genome-wide genetic data enabled the investigation of LD-blocks, enabling the study of breaks in haplotypes between different populations. This knowledge can be leveraged to measure the transgenerational decay of LD-blocks and infer relative admixture dates.

ROLLOFF is the first reported software to infer admixture dates based on LD decay. The method was first reported by [Moorjani *et al.* \(2011\)](#) and was later incorporated in the ADMIXTOOLS software package developed by [Patterson *et al.* \(2012\)](#) to enable admixture studies across human history. It aggregates pairwise LD measurements using a weighting scheme, to produce a weighted LD curve as a function of genetic distance. This results in an exponential LD decay, and, given a rate constant, it can be further used to infer admixture dates.

[Loh *et al.* \(2013\)](#) built on the ideas of ROLLOFF to develop ALDER, a more sensitive algorithm that incorporates the information of the amplitude of the weighted LD curve to interpret further admixture parameters, enabling inferences on history, such as admixture dates and admixture fractions, or even to infer phylogenetic relationships. The latest extension of the method, MALDER, is able to additionally infer multiple admixture events. ALDER performs very well for two-way admixture hypotheses, but is limited to admixture events less than five hundred generations ago.

The aforementioned methods are used on genotypic data and rely on LD calculations to produce the decay curves. However, they are susceptible to performance issues when dealing with large scale datasets. Newer approaches suggest that utilizing haplotype information can produce more efficient and accurate results. GLOBETROTTER is a method described by [Hellenthal *et al.* \(2014\)](#) that is based on the Chromopainter output to infer admixture dates. Chromopainter is an efficient method that can paint chromosomes based on an ancestry estimation model using large scale data of the tested population and surrogate populations. The resulting haplotypes are then used by GLOBETROTTER to estimate the pairwise likelihood of these haplotypes being painted by these surrogate populations. These haplotypes are analyzed at a range of genetic distances to produce coancestry curves, to an exponential decay curve output similar of the previous methods. The advantage of this method is that the admixing sources are modelled as a linear mixture of surrogate populations, which enables the non-exact sampling of the source populations and the

use of larger geographical groups as surrogates. This method is best used to infer later dates of admixture, and produces a wealth of information, but can underperform when using less than 300,000 markers.

Instead of fitting an exponential LD-decay curve, [Pugach *et al.* \(2011\)](#) developed stepPCO by following a PCA-based approach. StepPCO applies PCA stepwise to identify block-like admixture signals and computes discrete wavelet transform to calculate frequencies and position from the sum of waves in each signal. The dominant frequencies are then compared to the calculated ones to generate the admixture rates. [Sanderson *et al.* \(2015\)](#) provided some extensions to the stepPCO method, accelerating the runtime and improving the statistical power when differentiating between separate admixture events, but limiting it to only two-way admixture hypotheses. StepPCO also requires haplotypes as input, which can be estimated with the appropriate software that were described in the *Modelling-based methods* section.

Each of these admixture dating methods possess various advantages and disadvantages, thus the use of multiple methods is recommended. A limitation that all these algorithms share is their inability to explore genetic events over 5000 years ago, which makes those algorithms better suited for more recent admixture events.

Network Analysis

Network theory, as the study of graphs for related discrete objects, can be applied to genetic data to infer relationships between the samples. The results of the previously mentioned analytic methods can be integrated towards the construction of nodes and edges to create interpretable graphs. Network analysis provides in depth visualizations for multi-dimensional data.

In population genetics, in the case of the aforementioned analytic strategies, we often resort to a specific series of assumptions, a major one of them being that populations emerge in the form of a tree and deriving from a set ancestral population and collectively refer to every other genetic influence as admixture. Network theory can aid the exploration of past demography, by employing the principles of reticulate evolution into our data. As described earlier, in many cases, there are no specific trees that can explain the data, with multiple lineages merging or converging to form new populations. In such cases, reticulate patterns of demography can occur. The integration of network theory into population genetics and phylogenetics can act as a valuable tool to address such issues.

Phylogenetic networks can be divided into two main categories, the unrooted and the rooted networks. Numerous methods have been developed for both types of networks, some of which have achieved popularity. Software packages have been constructed in an effort to encompass the most mainstream of them for the ease of the user. In this section we will briefly describe some of these approaches and the theory behind them.

SplitsTree is an interactive application developed by [Huson and Bryant \(2006\)](#), that infers unrooted networks from genetic data. It also includes the NeighborNet algorithm, which was introduced by [Bryant \(2003\)](#) to construct split networks from measured genetic distances. Dendroscope was also developed by [Huson and Scornavacca \(2012\)](#), with the goal of inferring rooted phylogenetic networks, integrating algorithms such as the CASS by [Iersel *et al.* \(2010\)](#), the cluster network by [Huson and Rupp \(2008\)](#), and the galled network by [Huson *et al.* \(2009\)](#). These programs can accept pairwise genetic distance matrices or sequence alignments to infer reticulated demography. A commonly chosen metric for these calculations is F_{ST} , as described in a previous section.

NetStruct is a newer package developed by [Greenbaum *et al.* \(2016\)](#) that draws from the community and modularity methods included in the *igraph* software package, described by [Csárdi and Nepusz \(2006\)](#), to create networks and infer population structure. For input it requires genotypes, in contrast to the previous methods, but it does not account for linkage disequilibrium, which is one of its shortcomings.

[Paschou *et al.* \(2014\)](#) and [Stamatoyannopoulos *et al.* \(2017\)](#) developed a different approach into network inference. Instead of using F_{ST} or traditional phylogenetic methods, they opted to infer networks from the output of PCA and ADMIXTURE. To form the networks, they identified the top few nearest neighbors of each sample by representing each sample with respect to the top K coefficients returned by PCA or ADMIXTURE, and then computing the distance of each sample to all other samples that do not belong in the same population. In the resulting network, an edge between two populations shows that the two populations share genetically related individuals, with thicker edges indicating a larger number of genetically close individuals, which gives insights into the pathways of inter-population genetic flow.

A newer method was introduced by [Zhang *et al.* \(2017\)](#), who deployed the ideas behind chromosome painting and NeighborNet to combine them using a Markov clustering approach. This approach features the investigation of recombination events, an increased resolution of shared ancestry, and the determination of the number of clades.

Networks are an important part of population genetics and phylogenetics, since reticulate evolution is the rule, which most times in phylogenetics we tend to disregard because of the model assumptions and its complications. Networks deviate from the rigid tree-like structure, and lead to a broader concept of population emergence and admixture. Of course, no method escapes all biases, so careful examination, along with a global exploration of the data is recommended.

Scanning for Selection

The methods explored in the previous section have a limit to their detection timescale. When investigating events in the far past, the effect of the evolutionary selective forces becomes increasingly evident. In this section we describe microevolutionary theory in conjunction with its applications on population genetics.

Natural selection creates regions of strong LD, raising the possibility of reverse engineering information about natural selection in populations. Considering that a new variant arises on a chromosomal background, there will be SNPs nearby in strong LD that are highly probable to be shuffled by recombination into increasingly smaller haplotypes through generations, which are going to be associated with that variant. Under a neutral model, this variant is expected to increase in frequency gradually so that by the time it attains higher frequencies, it will be on a relatively small associated SNP haplotype background.

If there has been rapid positive selection for that variant, it will essentially drag the nearby SNPs along with it and we will be able to identify a region of high linkage disequilibrium because this variant has evolved quickly to high frequency, so quickly that recombination was late to reshuffle these nearby SNPs. This is one of the signatures that indicates a region that underwent recent and rapid selection. To avoid false positives, it is suggested to remove low complexity regions, as well as the highly-complex HLA region, due to our limited knowledge into their genetic mechanics.

The detection signatures of selection is a vast category of analytic methods that is further divided into three subcategories, the frequency-based methods, the linkage disequilibrium-based methods, and the population differentiation-based methods. It is of note that, when selecting the analytic method, the timescale over which we hypothesize that selection has acted upon is considered to have a major effect in our analysis. Each method by its inception has fundamental advantages and disadvantages. Methods that rely on population differentiation and allelic frequencies have an advantage in large timescales, due to their inherent assumptions.

By Allelic Frequency

Since a selected genomic region can affect the allelic distributions in populations, statistical tests investigating allelic frequency patterns were developed to detect deviations from the neutral theory of evolution. Tajima (1989) was the first to introduce such a selection metric, referred to as Tajima's D . It operates by comparing the average pairwise difference between samples to the number of segregating polymorphisms. It is the most popular metric to detect selection signals, and spawned an array of derivatives.

The H metric by Fay and Wu (2000), building on Tajima's D , compares the pairwise differences to the samples that are homozygous for the derived allele. H measures the deviation from neutrality as they manifest between high and intermediate allelic frequencies, while D between low and intermediate allelic frequencies. Continuing this line of thought, Zeng *et al.* (2006) introduced E , which measures between high and low allelic frequencies, while putting less weight on high allelic frequencies. The combination of all of these metrics can act as an indicator of population expansion and distinguish it from purifying selection.

Fu and Li (1993) developed two metrics, F^* and D^* for the examination of singleton mutations. The D^* metric compares the average pairwise differences to the singleton mutations, and the F^* compares the average nucleotide differences to the singleton mutations. Of these two statistics, F^* is considered to be the one with the greatest power. The R^2 , introduced by Ramos-Onsins and Rozas (2002), also utilizes singleton information and improves upon the previous method by comparing the differences between the average nucleotide differences to the singletons in each sequence, and also accounting for ascertainment bias in the use of genotype microarrays.

The timescale on which these methods operate is indeed vast. Tajima's D can detect evidence of selection in the past 10,000 generations of human populations, while Fay and Wu's H can detect events up to 3000 generations in the past.

By Linkage Disequilibrium

Selective pressure can create extended regions of strong LD, with haplotypic segments that are consistently inherited through generations. By utilizing the information gained by extended haplotypes, it is possible to infer positive selective pressure.

Sabeti *et al.* (2002) proposed a statistic for the measurement of the decay of LD, called extended haplotype homozygosity (EHH), defined as the probability that two random haplotypes in the sample set are identical by descent. This statistic inspired the development of several influential methods. In the same study, Sabeti *et al.* (2002) developed the long-range haplotype (LHR) test to investigate EHH. Based on the assumption that recent events should have a clearer signal when they extend over 10 times that of the background LD segments (0.02cM), it can detect recent positive selection, typically up to 400 generations in the past. Expanding on EHH, Sabeti *et al.* (2007) also developed the cross-population extended haplotype homozygosity (XP-EHH) test, which examines variation in recombination rates by comparing haplotype lengths among populations to probe for variation in recombination rates. Voight *et al.* (2006) introduced the integrated haplotype score (iHS), a variation of EHH, which attained popularity. It calculates the integral of the EHH decay curve for the ancestral and the derived variants, to provide a measure of the unexpectedness of the haplotypes around a marker relative to the whole genome. These methods are designed to identify alleles that have attained high frequencies in such a rapid fashion that the LD around them has not decayed. While XP-EHH is more suitable with selective sweeps nearing fixation in one, but not all populations, iHS has the power to detect partial sweeps.

Leveraging the information gained by haplotypes, Fu (1997) based the calculation of his F_S on the infinite-sites mutation model to estimate the probability that a random individual would possess an equal or smaller number of alleles compared to the observed values. Fu's F_S is more powerful in detecting population expansions than Tajima's D , and also performs very well in large sample sizes. It is sensitive in the presence of recombination, which could also be used as a test for evidence of recombination. D_h is another measure that uses haplotypic information, introduced by Nei (1987). It is an unbiased haplotype diversity estimate, which computes the probability that two random haplotypes in the sample are different. In a more brute fashion, identified IBD segments, which can be obtained through the methodology described in the previous section, can also

be utilized for the detection of selective sweeps. If a particular IBD segment is enriched in a specific population, then it could be evidence for selection.

These statistics can be significantly affected by erroneous recombination estimates, and as such, when recombination rates are unknown, it is preferred to rely on the statistics based on allelic frequency, preferably Tajima's D or the R^2 . Additionally, as these methods are essentially based on LD, their investigative timescales are much lower than those based on allelic frequency.

By Population Differentiation

Selection patterns can be investigated through the direct measurement of the frequencies of the derived allele between populations. Given a common ancestor, we define the allele that was passed onto its descendants as ancestral, while the allele that was created by mutation as derived. The derived allele is subject to selection, either positive or negative. This difference between the frequencies observed between populations, the δDAF , can be utilized to leverage derived allele frequency differences to infer whether selection has occurred.

F_{ST} , described earlier, can also indicate if diversifying selection has occurred by cross-examining between populations. A genome-wide scan of single marker estimates of F_{ST} can identify regions where natural selection has favored specific alleles in specific populations, with a detection range of up to 3000 generations into the past. In order to properly identify selection signatures using this method, a proper genome-wide background should be specified, depending on the properties of the study. When identifying adaptation to a specific environment, a very similar population that resides in a different habitat should be selected as the background for comparison. F_{ST} can be highly variable when applied only to single loci to study selection, and counter-measure approaches propose utilizing sliding windows to report average or highest values, which can lead to the discarding of valuable information. As with the phylogenetic usage of F_{ST} , scrutiny of the appropriate estimators, as well as the markers to be used is recommended.

By Composite Scoring

Each of the previous categories and tests is designed to detect a particular type of signal. However, the selective process is not limited to a specific signal type, generating a multitude of signal types. The study of a single type of signal can limit the power and the resolution of our inquiries. To that end, there have been efforts to combine multiple metrics based off different approaches into one composite method to attain greater power and resolution.

The composite likelihood ratio test (CLR) is a statistical method introduced by [Kim and Stephan \(2002\)](#) to probe the reduction of variation in local regions, which divides the maximum composite likelihood under a model without selection by the maximum composite likelihood under selection. This approach was successful in utilizing the spatial variation patterns in the frequency spectrum, with the downside of being inefficient for large scale datasets and sensitive to recombination and mutation rate biases. Building on these ideas, [Nielsen et al. \(2005\)](#) modified the CLR method by opting to calculate the null hypothesis from the background of the input data. However, this method does not account for multiple populations, and as such [Nielsen et al. \(2009\)](#), with G2D, and [Chen et al. \(2010\)](#), with XP-CLR, provided extensions to account for cross-population comparisons. The XP-CLR is based on a model-based extension of F_{ST} to compare multilocus allelic frequencies between populations, and is able to capture the spatial patterns of frequency skews that emerge around given markers. XP-CLR is able to better avoid ascertainment bias and demographic uncertainties, while G2D is better suited for when detailed demographic models are available. Additionally, a computationally efficient implementation by [Pavlidis et al. \(2013\)](#), enables the application of the CLR method on very high density data, such as the ones produced by WGS.

Another strategy into composite scoring is the combination of many tests for a single site. [Pybus et al. \(2014\)](#) opted to use combine the results of a variety of methods, combining them using a ranking algorithm to evaluate their significance of the findings. Their method utilizes supervised algorithms in a hierarchical classification scheme, by maximizing the difference between different evolutionary scenarios.

The methods described in this section can be found in standalone executables by their authors, integrated in a variety of community packages of programming languages (e.g., R, python), or in software packages, for example selscan by [Szpiech and Hernandez \(2014\)](#), Sweep by [Sabeti et al. \(2007\)](#), Arlequin by [Excoffier and Lischer \(2010\)](#), and DnaSP by [Rozas et al. \(2017\)](#).

Assessment and Considerations

Assessment of PCA and PCA-derived methods mostly relies on the visualization of the components that capture the most variance. Notably, approximately ninety per cent of the total genotypic variance pertains to intra-population variance, which means that only a fraction of the top principal components will be useful for the correct interpretation of inter-population differences. In most cases, the top three principal components are indicative of population structure, which can be visualized either in a three-dimensional scatter plot, or a combination of two-dimensional scatter plots.

The assessment of model-based clustering methods is based on the use bar plots to visualize the ancestry coefficients that are contained in the Q matrices, which was also one of the major factor that contributed to their immense popularity. This in turn translates to multiple analysis runs for different K s of hypothetical ancestral populations. ADMIXTURE's cross-validation

procedure produces an error estimate that aids ancestry modelling. The lowest cross-validation error value is proposed to be the most sensible K value to be used for ancestry modelling. STRUCTURE and its derivatives rely on the calculation of the marginal likelihood for each K , where the greater the value, the better the fit of the data to the model.

Because of the model that they operate on and their inherent assumptions, model-based clustering methods are prone to misinterpretation. To address this issue, several points must be stressed. Since the K value is an estimate, there is no evidence that it is the true one. The true value of K is highly dependent on sampling and its geographical regularity. In the case that important population groups have been under-, over-, or even unsampled, K estimates will not lead to meaningful interpretation. Apart from its numerical value, the qualities of the K ancestral populations should also be considered. The K ancestral populations could, but could equally not have existed at all in history. These ancestral populations should not be considered homogeneous and unadmixed, and they may as well have been related to each other. The primary assumption of those methods is a unidirectional admixture model, which works in the case of simple population histories but can prove problematic in complex population histories. In such cases, they select the most prominent drift components as they are detected and regard them as ancestral, which can lead to populations close to ancestral frequencies to be regarded as admixed. These methods cannot discern between admixture, retrograde admixture, or population bottlenecks, which can heavily influence the resulting output.

The practicality of distance-based methods is sufficiently multifaceted. They can produce the branches of a phylogenetic tree, they can showcase the shared genetic drift between groups, and they can deduce complex demographic scenarios based on the coalescence times and the internal genealogical branch lengths. The complex nature of this range of interpretations though also raises some points of consideration. The choice of markers is critical for the type of analysis aimed for, with a notable example being the inclusion of genomic areas undergoing strong selection which can confound investigations into past demography. The presence of LD can also influence the analysis, with current software packages offering elegant solutions to this issue, such as the bootstrapping and the block-jackknife that can produce error measurements. Because the main method of interpretation is through eye examination, the distance measure and the choice of graphical representation may significantly affect the identified clusters. Because of their complex statistical nature, distance-based methods can occasionally be counter-intuitive, and as such, require a deep understanding of the concepts behind them.

In the case of the admixture dating methods, it needs to be stressed that they operate with specific hypotheses. One is that admixture happens in pulses, as single admixture events, and not at prolonged periods of rates. The second is that they take into account only unilateral admixture between pairs of populations. Third, since they require the use of surrogate populations to test their hypotheses, they can open the gates to a flood of biases, especially when selecting the surrogate populations, since most populations are already admixed. In that case, the modern populations should be carefully selected as surrogates and potential biases clearly stated. Their investigative past timeline is also limited, with 5000 years being the lowest bound in the range of inferred dates.

When probing selection sweeps, it is probable that while the neutrality tests can lead to the rejection of the null, the results could prove to be falsely positive. Demographic events may well interject in the performance of the algorithms, and to combat such possibilities, many methods that take into account demographic models have been implemented. Strong LD between a neutral and a selected variant can also create false-positives in the interpretation of the results. Confounding patterns of LD can also arise due to systematic biases. Using genotype microarray data often cause the oversight of markers with lower frequency, leading to ascertainment biases in the analysis.

In all cases, effective and regular sampling is very strongly recommended. All the cited methods are particularly sensitive on the samples used, the genetic differentiation between the populations, and, equally important, on the number of markers examined. Also of note is that the analytic choice of populations should be coupled with an appropriate density in markers. Genetic discrimination between subgroups of an ethnic group, requires a much higher density in markers than discriminating between different ethnic groups. As always, the global use of a variety of different methods is advised. As in the case of model-based clustering algorithms, different software use different estimation or approximation procedures to achieve computational efficiency. Different methods can capture mishaps and erroneous estimates produced by other methods, or can be particularly sensitive in specific genetic signals.

The Peopling of Europe as an Illustrative Case

The genetics of the European continent have provided prime illustrations of the capabilities that population genetics has when applied to study historical demographics. Archeogenetics and population genetics have been able, supported by palaeological and archaeological evidence, to give insight to the population movements that occurred from the transition of the Middle Paleolithic into the Upper Paleolithic and into the modern structure of anatomically modern humans.

Since the landmark studies by Cavalli-Sforza *et al.* (1994), it was evident that genetic components seem to follow continuous clines in Europe, a notion further reinforced by the whole-genome genotype analysis of Novembre *et al.* (2008). Cavalli-Sforza *et al.* (1994) proposed that these clines were the indications for the mass population movements of the Neolithic expansion into Europe, over the Mesolithic and Epipaleolithic populations. Present day knowledge now confirms these notions, albeit in a much more complicated manner than the one originally proposed.

The genetic ancestries of the modern Europeans can be traced to three major genetic influences as showcased by Lazaridis *et al.* (2014): the Mesolithic hunter-gatherers, the anatolian Neolithics, and the migrants from the steppe. To elaborate on these ancestries, a recapitulation of the current state of knowledge is in order.

There has been evidence of a wide distribution of anatomically modern humans by 43,000 BCE in Europe, as suggested by Benazzi *et al.* (2011). This Aurignacian culture is thought to have led to the disappearance of the Neanderthals in a multitude of ways.

The oldest genomic evidence available originates from this exact era, an individual from the geographic area of modern Romania named Oase 1. This individual was reported by [Fu et al. \(2015\)](#) to have 6%–9% Neanderthal ancestry, with genomic tracts extending over 50cM, indicating a Neanderthal ancestor 4–6 generations past. However, the genome of this individual suggests that this population did not contribute in an identifiable way to the genetics of the later humans. The Aurignacian culture was most likely succeeded by the Gravettian culture during the Upper Paleolithic era at approximately 35–30,000 BCE, which was most probably the last unified culture of Europe. [Fu et al. \(2016\)](#), analyzing 51 Eurasians that ranged from the Upper Paleolithic to the Mesolithic era (45,000–7000 BCE), reported that even though there was no detectable genetic influence by these early humans over the modern humans, these people were the likely descendants of a single founder population, with a 35,000 year-old human specimen representing the earliest discovered branch of this population. [Fu et al. \(2016\)](#) also identified a major shift in the peopling of Europe after the last Ice Age and during the major warming period, with the disappearance of the Gravettian culture, and migrants with a significant genetic component attributed to modern Near Eastern populations. This era was most likely the subject of intense population shifts, as it is proposed that these recolonizers were replaced by the eventual future Mesolithic cultures. These Mesolithic cultures are proposed to be a medley of at least two ancestral sources after the last Ice Age period, though with no clear evidence of their origin. Analysis of genomic data from Mesolithic and later populations, such as the studies by [Lipson et al. \(2017\)](#), [Lazaridis et al. \(2014\)](#), [Skoglund et al. \(2012\)](#), [Sánchez-Quinto et al. \(2012\)](#) suggest a genomic diversity background that deviates from the one observed in modern humans, but similarly showing clinal patterns, following an east to west genetic cline.

The rise of the Neolithic Age in the fertile crescent, brought about a major population migration into Europe, initiating during the 8th millennium BCE, leading to the first agrarian societies in Europe by the 7th millennium BCE. These Neolithic migrants conveyed the agricultural technology and culture to the whole mainland of the European continent. Many studies, such as the ones by [Gamba et al. \(2014\)](#), [Haak et al. \(2015\)](#), [Hofmanová et al. \(2016\)](#), demonstrated the genetic entanglement of the Mesolithic and Neolithic ancestries into admixed populations with a 10%–25% Neolithic ancestral component. The exact route followed by the Neolithics was a matter of historical debate, with genetics supplying insights into the migrational routes, as [Paschou et al. \(2014\)](#) demonstrated genetic signatures of a maritime route of colonization that was followed by the Neolithics from the fertile crescent into Europe. This finding has also been supported by [Hughey et al. \(2013\)](#), who inferred from mitochondrial data that the Minoan population was indeed a population closely resembling the Neolithic and modern European populations.

The last genetic component of the modern Europeans derives from a genetic influence from the Eurasian steppe, around the 5th millennium BCE. The studies by [Allentoft et al. \(2015\)](#) and [Haak et al. \(2015\)](#), identified an ancestral component associated with the Yamnaya and the Afanasievo cultures, mostly appearing in the Northeastern and Central Europe. These steppe populations also show distinct signatures of admixture, themselves being an amalgamation of Mesolithic cultures from East Europe and the Caucasus, the Neolithic and later age Iranian fields, as well as modern population components. These suggest that the genetic flow followed a bidirectional route, instead of a unidirectional one.

The nature of the genetic influence over other populations does not always follow the same mechanics, with some genetic signatures involving abrupt genetic influences, and others gradual genetic assimilation. Especially in the case of the steppe genetic influx, the transitions appear to have occurred between relatively short timescales. [Goldberg et al. \(2017\)](#) and [Saag et al. \(2017\)](#) demonstrated that while the Neolithic migrations involved large numbers of males and females, the migrations from the steppe, as well as their genetic influence, were mostly of male origin.

Modern day European population structure is the result mainly of these major genetic influences, with the magnitude of these influences varying by region. The patterns of these genetic components shift based on the genetic flow from these major or from minor sources. [Haak et al. \(2015\)](#) reported a more prevalent Anatolian Neolithic ancestry in southern Europeans than in northern Europeans, and higher percentages of steppe ancestry in north-central Europeans than southern Europeans. The genetic clines initially observed by Cavalli-Sforza, led to the reveal of an intriguingly intricate genetic past of Europe, branching off to the study of the human history over all continents, enabling the exploration of the rise of the modern anatomical humans and their expansion all over the world. Population genetics contributed immensely to that end, supplying an overwhelming amount of data and analytic methods for the investigation and inference of past human demographics.

The study of population genetics has also been exercised on later generations of genetic history, into the medieval and more recent time periods, to investigate and elucidate historical controversies. [Stamatoyannopoulos et al. \(2017\)](#) tested the hypothesis set by Jakob Philipp Fallmerayer, a strong proponent of the theory of the eradication of medieval Peloponnesean Greek populations by Slavic and Avar invaders and the genetic discontinuity between medieval and modern Greeks. The results refuted those claims, indicating a strong genetic similarity of the modern Greeks with modern Sicilians and Italians and a clear differentiation from the genetics of the Slavic homeland. Population genetics also has the remarkable ability to produce historical controversies, such as the case of the remains of King Richard III of England, which were genetically investigated by [King et al. \(2014\)](#).

The study substantiated that, while the maternal royal line was continuous, the paternal one was surprisingly discontinuous, which laid doubts on the right of patrilineal succession of medieval English kings even into the modern royal family.

Application of Population Genetics in Disease

The exact mechanisms that link genetic variation to disease have been a major subject of debate, stemming from the era of classical genetics and reaching to modern genetics. [Fisher \(1919\)](#) laid the foundations for considering polygenic disorders and continuous phenotypes as a combination of multiple Mendelian loci. Expanding on these notions, Common Disease-Common Variant is a

well established hypothesis, tracing its origins on multiple articles but mainly to the landmark paper by Reich and Lander (2001). In essence, it states that for a disease of high prevalence in a population, the genetic components that lead to it should be relatively common in that population.

Following the Common Disease-Common Variant hypothesis and its extensions, when investigating a multifactorial disease, a significant amount of its genetic background should rely on common variance. Since common variance is heavily influenced by demography and evolutionary forces, the implementation of population genetics becomes critical for the avoidance of a result riddled with false positives that can be attributed to population structure in the data.

When the analysis spans an extensive amount of the genome, such as in the case of a GWAS, the use of PCA or model-based clustering methods can sufficiently correct for inflation related to population stratification. Price *et al.* (2006) demonstrated that incorporating the output of the principal components for each individual as coefficients in an association model can alleviate the inflation effects. Alexander *et al.* (2009) also designed ADMIXTURE as a method competing with EIGENSOFT in the calculation of ancestry coefficient for association modelling. Nowadays, it is commonplace to use the output of PCA, or ADMIXTURE or related methods, as coefficients in a logistic regression association model to correct for inflation in the statistics.

However, a critical number of association studies are not performed in a genome-wide frame, especially in the case of replication studies that seek to validate results from GWAS in various populations. In that case, the use of PCA to identify SNPs that carry the inter- and intra-population variance, and their subsequent genotyping along with the investigated markers, is fundamental for the avoidance of bias and confounding related to population structure.

To that end, Paschou *et al.* (2007) developed a PCA-based methodology to infer ancestry informative markers (AIMs), that was able to identify the prime markers capturing the intercontinental and cross-population variance, efficiently reducing the number of markers required for structure identification to a mere handful, though these markers were in general not overlapping among regions. Paschou *et al.* (2008) applied the method in European-American datasets, effectively demonstrating that by using this method, 150–200 AIMs were sufficient to capture the fine details of stratification in individuals that derived their ancestry from Europe. Drineas *et al.* (2010) applied the method to successfully infer geographic coordinates of European population data by using a small subset of the markers, further showcasing the effectiveness of the method and the bridging between population and disease genetics. Through an extension of the method, Paschou *et al.* (2010) demonstrated its applicability in a portable worldwide setting, resulting in a significant reduction of required markers to capture broad and fine-scale stratification to less than 0.1% of the original dataset. These methods were very successful in denoting the power of PCA to decrease study expenditure and increase computational efficiency when controlling for stratification effects in association analyses.

Discussion

Population genetics has a deep impact on the whole field of genetics, shaping many of its fundamentals, and branching off into different subfields. Nowadays we can discern between population genetics, palaeogenetics, evolutionary genetics and spatial genetics. Palaeogenetics, or archeogenetics, as the genetic study of samples that predate the current era, since modern technology has enabled the isolation of DNA of appreciable quality from bone samples and the sequencing of their bases, and spatial genetics as the joint analysis of genetics and geographical space and the investigation of their interaction and co-dependence. We refrained from referring to spatial genetics methods, as we believe that they constitute a subfield of their own, and they would be outside the scope of population genetic analysis. We also skimmed the surface of palaeogenetics and evolutionary genetics, by mentioning some of the methods used for these analyses, that are overlapping with population genetics, but not delving into the complex theory required for their interpretation.

Originally, when drafting an article about population genetics analyses, a dilemma arises faced on the decision on how to classify and showcase the algorithms and the protocols, whether the criterion should be the depth level required of the genetic data, the form of the genetic data, or the algorithmic method on which the software is based on. In this guide, we opted for the latter. We focused mainly on the nature of the base method, than how many variants should be included in the dataset.

High-density data enable what is termed as local ancestry inference, namely the ability to infer the ancestry of specific genetic tracts all over the genome. Global ancestry inference is still very popular, and for a good reason, since the majority of variants are associated through LD, which means that global ancestry inference methods can work on thinner datasets, which is most commonly the case when conglomerating datasets.

Oftentimes, in interdisciplinary research, it is not uncommon that specific terms get misused or mixed. A notable example is referring to a method as “supervised”. Pritchard *et al.* (2000) refer to the term to describe a method that incorporates information to the data that relates to their prior groupings. Other researchers refer to the term as the need to specify their own number of K ancestral populations. There is no correct or erroneous use of the term, as in both cases its use is justified, and we are just using this cautionary example to prevent misgivings when interpreting published analytic protocols.

We can apply the information gained from our studies of genetic variation that pertains to the use of LD in disease-gene mapping. LD essentially reflects the patterns of recombination occurring over many hundreds of generations, especially for closely linked loci, and we can use it to infer the distance between closely linked loci. Populations that were founded a long time ago had more evolutionary time for recombination to occur, which leads to less LD. Specific groups of populations have had a more rapid decay of LD between SNP pairs (1000 Genomes Project Consortium, 2015), which necessitates a population genetics view when pursuing and interpreting disease-associated genomic loci. Since many markers in the genome are in LD with each other, they can

be considered redundant for our analyses, and we can get relatively complete coverage of a genome in a GWAS with approximately 1.5 million SNPs for African derived populations, and approximately a half million to a million SNPs for non-African populations. Thus, studies of population history can inform our design of these GWAS and our design of SNP microarrays.

It is important to note, that the most significant and impactful current scientific advances in human population genetics have been achieved through the study of SNVs. Such variants comprise the overwhelming majority of genetic variation, enabling the capture of a significant amount of variation for study. Another category of variation is the structural variation, with a substantial amount of variation existing at the structural level. These structural variants are more difficult to identify than SNVs, but, as [Sudmant *et al.* \(2015\)](#) reported in their study, we can estimate that in the average haploid human sequence, at least 9mb are affected by structural variants. When compared with the approximately 3.5mb in the average genome that are affected by SNVs, these structural variants, even though occurring less frequently in the genome, account for more differences than SNVs. And if we also consider Copy Number Variations (CNVs), where genetic segments can differ by multiple copies, each human is heterozygous for approximately 150 of those CNVs.

As rare variants tend to be population specific, we can investigate a lot about the functional significance of these genetic variants, because functional regions in the genome, whether coding or non-coding, tend to show more evidence of purifying selection. And we can actually use this information to more effectively identify functional regions of the genome.

The examples of CNVs and rare variation showcase the amount of genetic information that still eludes us. However, the modern technological advances have overcome that limitation, producing more global maps of genetic variation. With the avalanche of genetic data being generated daily, we can await for a wealth of knowledge to be generated in the following years, along with the development of methods that will be able to handle that level of data.

See also: Natural Language Processing Approaches in Bioinformatics

References

- 1000 Genomes Project Consortium, 2015. A global reference for human genetic variation. *Nature* 526 (7571), 68–74. doi:10.1038/nature15393.
- Alexander, D.H., Novembre, J., Lange, K., 2009. Fast model-based estimation of ancestry in unrelated individuals. *Genome Research* 19 (9), 1655–1664. doi:10.1101/gr.094052.109.
- Allentoft, M.E., Sikora, M., Sjögren, K.G., *et al.*, 2015. Population genomics of Bronze Age Eurasia. *Nature* 522, 167.
- Benazzi, S., Douka, K., Fornai, C., *et al.*, 2011. Early dispersal of modern humans in Europe and implications for Neanderthal behaviour. *Nature* 479 (7374), 525–528. doi:10.1038/nature10617.
- Bhatia, G., Patterson, N., Sankararaman, S., Price, A.L., 2013. Estimating and interpreting FST: The impact of rare variants. *Genome Research* 23 (9), 1514–1521. doi:10.1101/gr.154831.113.
- Brisbin, A., Bryc, K., Byrnes, J., *et al.*, 2012. PCAdmix: Principal components-based assignment of ancestry along each chromosome in individuals with admixed ancestry from two or more populations. *Human Biology* 84 (4), 343–364. doi:10.3378/027.084.0401.
- Browning, B.L., Browning, S.R., 2013. Improving the accuracy and efficiency of identity-by-descent detection in population data. *Genetics* 194 (2), 459–471. doi:10.1534/genetics.113.150029.
- Browning, S.R., Browning, B.L., 2007. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *The American Journal of Human Genetics* 81 (5), 1084–1097. doi:10.1086/521987.
- Bryant, D., 2003. Neighbor-Net: An agglomerative method for the construction of phylogenetic networks. *Molecular Biology and Evolution* 21 (2), 255–265. doi:10.1093/molbev/msh018.
- Cavalli-Sforza, L., Edwards, A., 1967. Phylogenetic analysis. Models and estimation procedures. *The American Journal of Human Genetics* 19 (3 Pt 1), 233–257. doi:10.1073/pnas.85.16.6002.
- Cavalli-Sforza, L., Menozzi, P., Piazza, A., 1994. *The History and Geography of Human Genes*. Princeton University Press.
- Chang, C.C., Chow, C.C., Tellier, L.C., *et al.*, 2015. Second-generation PLINK: Rising to the challenge of larger and richer datasets. *Gigascience* 4 (1), 7. doi:10.1186/s13742-015-0047-8.
- Chen, H., Patterson, N., Reich, D., 2010. Population differentiation as a test for selective sweeps. *Genome Research* 20 (3), 393–402. doi:10.1101/gr.100545.109.
- Conrad, D.F., Keebler, J.E.M., DePristo, M.A., *et al.*, 2011. Variation in genome-wide mutation rates within and between human families. *Nature Genetics* 43 (7), 712. doi:10.1038/ng.862.
- Csárdi, G., Nepusz, T., 2006. The igraph software package for complex network research. *InterJournal, Complex Systems* 1695 (5), 1–9.
- Delaneau, O., Zagury, J.F., Marchini, J., 2013. Improved whole-chromosome phasing for disease and population genetic studies. *Nature Methods* 10 (1), 5–6. doi:10.1038/nmeth.2307.
- Drineas, P., Lewis, J., Paschou, P., 2010. Inferring geographic coordinates of origin for Europeans using small panels of ancestry informative markers. *PLOS ONE* 5, 8. doi:10.1371/journal.pone.0011892.
- Dungern, E von., Hirschfeld, L., 1910. Über vererbung gruppenspezifischer strukturen des blutes. *Zeitschrift für Immunitätsforschung und Experimentelle Therapie* 6, 284–292.
- Excoffier, L., Lischer, H.E.L., 2010. Arlequin suite ver 3.5: A new series of programs to perform population genetics analyses under Linux and Windows. *Molecular Ecology Resources* 10 (3), 564–567. doi:10.1111/j.1755-0998.2010.02847.x.
- Fay, J.C., Wu, C.I., 2000. Hitchhiking under positive Darwinian selection. *Genetics* 155 (3), 1405–1413.
- Fisher, R.A., 1919. XV. – The correlation between relatives on the supposition of Mendelian inheritance. *Transactions of the Royal Society of Edinburgh* 52 (2), 399–433.
- Frichot, E., Mathieu, F., Trouillon, T., Bouchard, G., Francois, O., 2014. Fast and efficient estimation of individual ancestry coefficients. *Genetics* 196 (4), 973–983. doi:10.1534/genetics.113.160572.
- Fu, Y.X., 1997. Statistical tests of neutrality of mutations against population growth, hitchhiking and background selection. *Genetics* 147 (2), 915–925.
- Fu, Q., Hajdinjak, M., Moldovan, O.T., *et al.*, 2015. An early modern human from Romania with a recent Neanderthal ancestor. *Nature* 524 (7564), 216–219. doi:10.1038/nature14558.
- Fu, Y.X., Li, W.H., 1993. Statistical tests of neutrality of mutations. *Genetics* 133 (3), 693–709. evolution.
- Fu, Q., Posth, C., Hajdinjak, M., *et al.*, 2016. The genetic history of Ice Age Europe. *Nature* 534 (7606), 200–205. doi:10.1038/nature17993.
- Gamba, C., Jones, E.R., Teasdale, M.D., *et al.*, 2014. Genome flux and stasis in a five millennium transect of European prehistory. *Nature Communications* 5. doi:10.1038/ncomms6257.

- Goldberg, A., Gunther, T., Rosenberg, N.A., Jakobsson, M., 2017. Ancient X chromosomes reveal contrasting sex bias in Neolithic and Bronze Age Eurasian migrations. *Proceedings of the National Academy of Sciences* 114 (10), 2657–2662. doi:10.1073/pnas.1616392114.
- Greenbaum, G., Templeton, A.R., Bar-David, S., 2016. Inference and analysis of population structure using genetic data and network theory. *Genetics* 202 (4), 1299–1312. doi:10.1534/genetics.115.182626.
- Gusev, A., Lowe, J.K., Stoffel, M., et al., 2009. Whole population, genome-wide mapping of hidden relatedness. *Genome Research* 19 (2), 318–326. doi:10.1101/gr.081398.108.
- Haak, W., Lazaridis, I., Patterson, N., et al., 2015. Massive migration from the steppe was a source for Indo-European languages in Europe. *Nature* 522 (7555), 207–211. doi:10.1038/nature14317.
- Hellenthal, G., Busby, G.B.J., Band, G., et al., 2014. A genetic atlas of human admixture history. *Science* 343 (6172), 747–751. doi:10.1126/science.1243518.
- Hirschfeld, L., Hirschfeld, H., 1919. Serological differences between the blood of different races. The result of researches on the Macedonian front. *The Lancet* 194 (5016), 675679.
- Hofmanová, Z., Kreutzer, S., Hellenthal, G., et al., 2016. Early farmers from across Europe directly descended from Neolithic Aegeans. *Proceedings of the National Academy of Sciences* 113 (25), 6886–6891. doi:10.1073/pnas.1523951113.
- Hubisz, M.J., Falush, D., Stephens, M., Pritchard, J.K., 2009. Inferring weak population structure with the assistance of sample group information. *Molecular Ecology Resources* 9 (5), 1322–1332. doi:10.1111/j.1755-0998.2009.02591.x.
- Hudson, R.R., Slatkin, M., Maddison, W.P., 1992. Estimation of levels of gene flow from DNA sequence data. *Genetics* 132 (2), 583–589. PMC1205159.
- Hughey, J.R., Paschou, P., Drineas, P., et al., 2013. A European population in minoan Bronze age crete. *Archives of Hellenic Medicine* 30 (4), 456–466. doi:10.1038/ncomms2871.
- Huson, D.H., Bryant, D., 2006. Application of phylogenetic networks in evolutionary studies. *Molecular Biology and Evolution* 23 (2), 254–267. doi:10.1093/molbev/msj030.
- Huson, D.H., Rupp, R., 2008. Summarizing multiple gene trees using cluster networks. In: *Proceedings of the International Workshop on Algorithms in Bioinformatics*, pp. 296–305. doi:10.1007/978-3-540-87361-7_25.
- Huson, D.H., Rupp, R., Berry, V., Gambette, P., Paul, C., 2009. Computing galled networks from real data. *Bioinformatics* 25 (12), doi:10.1093/bioinformatics/btp217.
- Huson, D.H., Scornavacca, C., 2012. Dendroscope 3: An interactive tool for rooted phylogenetic trees and networks. *Systematic Biology* 61 (6), 1061–1067. doi:10.1093/sysbio/sys062.
- Iersel, L., van, Kelk, S., Rupp, R., Huson, D., 2010. Phylogenetic networks do not need to be complex: Using fewer reticulations to represent conflicting clusters. *Bioinformatics* 26 (12), doi:10.1093/bioinformatics/btq202.
- Kim, Y., Stephan, W., 2002. Detecting a local signature of genetic hitchhiking along a recombining chromosome. *Genetics* 160 (2), 765–777.
- King, T.E., Fortes, G.G., Balaresque, P., et al., 2014. Identification of the remains of King Richard III. *Nature Communications* 5. doi:10.1038/ncomms6631.
- Landsteiner, K., 1900. "Zur Kenntnis der antifermentativen, lytischen und agglutinierenden Wirkungen des Blutserums und der Lymphe". *Zentralblatt Für Bakteriologie* 27, 357–362.
- Lawson, D.J., Hellenthal, G., Myers, S., Falush, D., 2012. Inference of population structure using dense haplotype data. *PLOS Genetics* 8 (1), 11–17. doi:10.1371/journal.pgen.1002453.
- Lazaridis, I., Patterson, N., Mittnik, A., et al., 2014. Ancient human genomes suggest three ancestral populations for present-day Europeans. *Nature* 513 (7518), 409–413. doi:10.1038/nature13673.
- Lewontin, R.C., Kojima, K., 1960. The evolutionary dynamics of complex polymorphisms. *Evolution* 14 (4), 458–472.
- Lipson, M., Szécsényi-Nagy, A., Mallick, S., et al., 2017. Parallel palaeogenomic transects reveal complex genetic history of early European farmers. *Nature* 551 (7680), 368–372. doi:10.1038/nature24476.
- Loh, P.R., Danecek, P., Palamara, P.F., et al., 2016. Reference-based phasing using the haplotype reference consortium panel. *Nature Genetics* 48 (11), 14431448. doi:10.1038/ng.3679.
- Loh, P.R., Lipson, M., Patterson, N., et al., 2013. Inferring admixture histories of human populations using linkage disequilibrium. *Genetics* 193 (4), 1233–1254. doi:10.1534/genetics.112.147330.
- Malécot, G., 1948. *Les Mathématiques de l'hérédité*. Barneoud freres.
- Moorjani, P., Patterson, N., Hirschhorn, J.N., et al., 2011. The history of african gene flow into Southern Europeans, Levantines, and Jews. *PLOS Genetics* 7 (4), 1–13. doi:10.1371/journal.pgen.1001373.
- Nei, M., 1987. *Molecular Evolutionary Genetics*. Columbia University Press.
- Nielsen, R., Hubisz, M.J., Hellmann, I., et al., 2009. Darwinian and demographic forces affecting human protein coding genes. *Genome Research* 19 (5), 838–849. doi:10.1101/gr.088336.108.
- Nielsen, R., Williamson, S., Kim, Y., et al., 2005. Genomic scans for selective sweeps using SNP data. *Genome Research* 15 (11), 1566–1575. doi:10.1101/gr.4252305.
- Novembre, J., Johnson, T., Bryc, K., et al., 2008. Genes mirror geography within Europe. *Nature* 456 (7218), 98–101. doi:10.1038/nature07331.
- Paschou, P., Drineas, P., Lewis, J., et al., 2008. Tracing sub-structure in the European American population with PCA-informative markers. *PLOS Genetics* 4 (7), doi:10.1371/journal.pgen.1000114.
- Paschou, P., Drineas, P., Yannaki, E., et al., 2014. Maritime route of colonization of Europe. *Proceedings of the National Academy of Sciences* 111 (25), 9211–9216. doi:10.1073/pnas.1320811111.
- Paschou, P., Lewis, J., Javed, A., Drineas, P., 2010. Ancestry informative markers for fine-scale individual assignment to worldwide populations. *Journal of Medical Genetics* 47 (12), 835–847. doi:10.1136/jmg.2010.078212.
- Paschou, P., Ziv, E., Burchard, E.G., et al., 2007. PCA-correlated SNPs for structure identification in worldwide human populations. *PLOS Genetics* 3 (9), 1672–1686. doi:10.1371/journal.pgen.0030160.
- Patterson, N., Moorjani, P., Luo, Y., et al., 2012. Ancient admixture in human history. *Genetics* 192 (3), 10651093. doi:10.1534/genetics.112.145037.
- Patterson, N., Price, A., Reich, D., 2006. Population structure and eigenanalysis. *PLOS Genetics* 2 (12), 2074–2093. doi:10.1371/journal.pgen.0020190.
- Pavlidis, P., Živković, D., Stamatakis, A., Alachiotis, N., 2013. SweepD: Likelihood-based detection of selective sweeps in thousands of genomes. *Molecular Biology and Evolution* 30 (9), 2224–2234. doi:10.1093/molbev/mst112.
- Pickrell, J.K., Pritchard, J.K., 2012. Inference of population splits and mixtures from genome-wide allele frequency data. *PLOS Genetics* 8 (11), e1002967. doi:10.1371/journal.pgen.1002967.
- Prado-Martinez, J., Sudmant, P.H., Kidd, J.M., et al., 2013. Great ape genetic diversity and population history. *Nature* 499 (7459), 471–475. doi:10.1038/nature12228.
- Price, A., Patterson, N., Plenge, R., et al., 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* 38 (8), 904–909. doi:10.1038/ng1847.
- Pritchard, J.K., Stephens, M., Donnelly, P., 2000. Inference of population structure using multilocus genotype data. *Genetics* 155 (2), 945–959. doi:10.1111/j.1471-8286.2007.01758.x.
- Pugach, I., Matveyev, R., Wollstein, A., Kayser, M., Stoneking, M., 2011. Dating the age of admixture via wavelet transform analysis of genome-wide data. *Genome Biology* 12 (2), doi:10.1186/gb-2011-12-2-19.
- Purcell, S., Neale, B., Todd-Brown, K., et al., 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics* 81 (3), 559–575. doi:10.1086/519795.
- Pybus, M., Dall'Olio, G.M., Luisi, P., et al., 2014. 1000 genomes selection browser 1.0: A genome browser dedicated to signatures of natural selection in modern humans. *Nucleic Acids Research* 42 (D1), 903–909. doi:10.1093/nar/gkt1188.
- Raj, A., Stephens, M., Pritchard, J.K., 2014. FastSTRUCTURE: Variational inference of population structure in large SNP data sets. *Genetics* 197 (2), 573–589. doi:10.1534/genetics.114.164350.
- Ramos-Onsins, S.E., Rozas, J., 2002. Statistical properties of new neutrality tests against population growth. *Molecular Biology and Evolution* 19 (12), 2092–2100. doi:10.1093/oxfordjournals.molbev.a004034.

- Reich, D., Lander, E., 2001. On the allelic spectrum of human disease. *Trends in Genetics* 17 (9), 502–510.
- Roach, J.C., Glusman, G., Smit, A.F.A., *et al.*, 2010. Analysis of genetic inheritance in a family quartet by whole-genome sequencing. *Science* (New York, NY) 328 (5978), 636–639. doi:10.1126/science.1186802.
- Rozas, J., Ferrer-Mata, A., Sanchez-DelBarrio, J.C., *et al.*, 2017. DnaSP 6: DNA sequence polymorphism analysis of large data sets. *Molecular Biology and Evolution* 32, 3299–3302. doi:10.1093/molbev/msx248.
- Saag, L., Varul, L., Scheib, C.L., *et al.*, 2017. Extensive farming in Estonia started through a sex-biased migration from the Steppe. *Current Biology* 27 (14), 2185–2193. doi:10.1016/j.cub.2017.06.022. e6.
- Sabeti, P.C., Reich, D.E., Higgins, J.M., *et al.*, 2002. Detecting recent positive selection in the human genome from haplotype structure. *Nature* 419 (6909), 832–837. doi:10.1038/nature01027.1.
- Sabeti, P.C., Varilly, P., Fry, B., *et al.*, 2007. Genome-wide detection and characterization of positive selection in human populations. *Nature* 449 (7164), 913–918. doi:10.1038/nature06250.
- Sánchez-Quinto, F., Schroeder, H., Ramirez, O., *et al.*, 2012. Genomic affinities of two 7,000-year-old Iberian hunter-gatherers. *Current Biology* 22 (16), 1494–1499. doi:10.1016/j.cub.2012.06.005.
- Sanderson, J., Sudoyo, H., Karafet, T.M., Hammer, M.F., Cox, M.P., 2015. Reconstructing past admixture processes from local genomic ancestry using wavelet transformation. *Genetics* 200 (2), 469–481. doi:10.1534/genetics.115.176842.
- Scheet, P., Stephens, M., 2006. A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *The American Journal of Human Genetics* 78 (4), 629–644. doi:10.1086/502802.
- Skoglund, P., Malmström, H., Raghavan, M., *et al.*, 2012. Origins and genetic legacy of neolithic farmers and hunter-gatherers in Europe. *Science* 336 (6080), 466–469. doi:10.1126/science.1216304.
- Stamatoyannopoulos, G., Bose, A., Teodosiadis, A., *et al.*, 2017. Genetics of the Peloponnesean populations and the theory of extinction of the medieval Peloponnesean Greeks. *European Journal of Human Genetics* 25, 637–645. doi:10.1038/ejhg.2017.18.
- Sudmant, P.H., Rausch, T., Gardner, E.J., *et al.*, 2015. An integrated map of structural variation in 2,504 human genomes. *Nature* 526 (7571), 75–81. doi:10.1038/nature15394.
- Spiech, Z.A., Hernandez, R.D., 2014. Selscan: An efficient multithreaded program to perform EHH-based scans for positive selection. *Molecular Biology and Evolution* 31 (10), 2824–2827. doi:10.1093/molbev/msu211.
- Tajima, F., 1989. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics* 123 (3), 585–595. PMC1203831.
- Voight, B.F., Kudaravalli, S., Wen, X., Pritchard, J.K., 2006. A map of recent positive selection in the human genome. *PLOS Biology* 4 (3), 0446–0458. doi:10.1371/journal.pbio.0040072.
- Weir, B.S., Cockerham, C.C., 1984. Estimating F-statistics for the analysis of population structure. *Evolution* 38 (6), 1358–1370.
- Wright, S., 1921. Systems of mating. *Genetics* 6 (2), 111–178.
- Wright, S., 1949. The genetical structure of populations. *Annals of Eugenics* 15 (1), 323–354. doi:10.1111/j.1469-1809.1949.tb02451.x.
- Wright, S., 1955. Classification of the factors of evolution. In: *Proceedings of the Cold Spring Harbor Symposia on Quantitative Biology*, vol. 20, pp. 16–24. Cold Spring Harbor Laboratory Press.
- Xing, J., Watkins, W.S., Witherspoon, D.J., *et al.*, 2009. Fine-scaled human genetic structure revealed by SNP microarrays. *Genome Research* 19 (5), 815–825. doi:10.1101/gr.085589.108.
- Zeng, K., Fu, Y.X., Shi, S., Wu, C.I., 2006. Statistical tests for detecting positive selection by utilizing high-frequency variants. *Genetics* 174 (3), 1431–1439. doi:10.1534/genetics.106.061432.
- Zhang, J., Khan, A., Kennard, A., Grigg, M.E., Parkinson, J., 2017. PopNet: A Markov clustering approach to study population genetic structure. *Molecular Biology and Evolution* 34 (7), 1799–1811. doi:10.1093/molbev/msx110.

Further Reading

- Casillas, S., Barbadilla, A., 2017. Molecular population genetics. doi:10.1534/genetics.116.196493.
- Cavalli-Sforza, L., Menozzi, P., Piazza, A., 1994. *The History and Geography of Human Genes*. Princeton University Press.
- Charlesworth, B., Charlesworth, D., 2017. Population genetics from 1966 to 2016. doi:10.1038/hdy.2016.55.
- Dyer, R.J., 2015. Population Graphs and Landscape Genetics. *Annual Review of Ecology, Evolution, and Systematics* 46 (1), 327–342. doi:10.1146/annurev-ecolsys-112414-054150.
- Holsinger, K.E., Weir, B.S., 2009. Genetics in geographically structured populations: Defining, estimating and interpreting FST. *Nature Reviews Genetics* 10 (9), 639–650. doi:10.1038/nrg2611.
- Lawson, D.J., Falush, D., 2012. Population identification using genetic data. *Annual Review of Genomics and Human Genetics* 13 (1), 337–361. doi:10.1146/annurev-genom-082410-101510.
- Nielsen R., Akey J.M., Jakobsson M., *et al.*, 2017. Tracing the peopling of the world through genomics. doi:10.1038/nature21347.
- Ségurel, L., Bon, C., 2017. On the evolution of lactase persistence in humans. *Annual Review of Genomics and Human Genetics* 18 (1), 297–319. doi:10.1146/annurev-genom-091416-035340.

Population Analysis of Pharmacogenetic Polymorphisms

Zen H Lu and Naeem Shafqat, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Nani Azman, Mark IR Petalcorin, and Lie Chen, PAPRSB Institute of Health Sciences, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

© 2019 Elsevier Inc. All rights reserved.

Overview

The practice of modern pharmacological treatment is very much a balancing act between the therapeutic benefit and toxicity or adverse effects of the prescribed drugs. While pharmacokinetic/-dynamic properties of drugs are usually balanced carefully with many exogenous and endogenous factors such as gender, age, weight, comorbid diseases, drug–drug interaction, etc., rarely are impacts of human genetic variation being taken into consideration when it comes to conventional prescription of medications. This is despite the fact that individuals' variation in genes involving in drug reactions/responses (pharmacogenes) within a population and/or among different ethnic groups have well been established to exert a direct impact on the metabolism and hence, the pharmacological efficacy and probable toxicity of drugs (Chung *et al.*, 2004; Hung *et al.*, 2006; Kaniwa *et al.*, 2013). In addition, manifestation of adverse drug reactions (ADRs) and drug efficacy can also vary geographically (Mizzi *et al.*, 2016).

To date, approximately 19,000 genetic variants from about 1600 human genes have been annotated in various databases to play either a direct or an indirect role in drug responses (Pharmgkb, 2017; Whirl-Carrillo *et al.*, 2012). As of December 2017, 255 of these variants are included in the labels of about 200 marketable drugs approved by the U.S. Food and Drug Administration (FDA) (Fda, 2017). This represents a mere 12% of the total number (1607) of FDA-approved drugs. In fact, population-wide genetic variation means that most of the medications are beneficial to only a subset of the treated patients; with the rest remaining either untreated or subsequently developing mild-to-serious ADRs (Schork, 2015; Spear *et al.*, 2001). The National Health Service in the UK has recently reported only 30%–60% “pharmaceutical effectiveness” for drug treatments (Hill, 2015).

“Imprecision” medicine is not only a leading cause of mortality and morbidity globally, the economic burden is enormous too. In the US alone, ADRs were estimated to cost up to approximately 30 billion US dollars annually (Sultana *et al.*, 2013). Until recently, the high costs and technological complexity associated with the precise identification, analysis, and interpretation of genetic variants have hindered the inclusion of these data in the wider pharmacological practices. However, with the advent of genomics technologies, especially that of sequencing and bioinformatics, a vast amount of data on population genetic polymorphism can now be generated. Together with the increasing availability of clinical phenotypes and pharmacosurveillance data, association between these variants and their impacts on drug efficacy and/or toxicity can now be resolved with a much higher resolution at a much lower cost.

This article aims to give a general overview on the population-wide approach to the fields of pharmacogenetics and pharmacogenomics. A particular focus is paid to the different bioinformatics methodologies used in the analysis of population genetic data.

Genetic Basis of Variability in Drug Response

Pharmacogenetics & Pharmacogenomics

The terms *pharmacogenetics* and *pharmacogenomics* are sometimes used interchangeably. The former describes the influence of genetic variants of a single gene or a number of genes on drug response. However, it is evident that many of these genes do interact, either directly or indirectly, and work within a network of drug pathways to contribute to the interindividual differences observed in drug response. Consequently, a comprehensive genome-wide approach, i.e., pharmacogenomics, involving not only the pharmacogenes but also their up- and downstream pathway genes is necessary.

In the simplest sense, for any drug to exert its therapeutic effect after entering the systemic circulation of the body, it has to bind to a certain receptor molecule before being delivered across the cellular membrane to reach and act upon its intended target; and finally with the unwanted end metabolites removed from the body. The cellular transport processes are carried out by the so-called pharmacokinetics (PK) genes, which can affect how a drug is being absorbed into the body and cells, distributed throughout its target sites, metabolized to either active or inactive forms, and excreted out of the body (Fig. 1). On the other hand, the effects of a drug on its targets and associated downstream pathways are mediated by pharmacodynamics (PD) genes (Fig. 1). Some of these PK and PD genes include ATP-binding cassette (ABC) and solute carrier (SLC) transporters, membrane targeting PDZ proteins, drug receptor families such as the G protein-coupled receptors (GPCRs) and various ion channels, and Phase I and II metabolizing enzymes. Among them, allelic variation in the different member genes of the Phase I enzyme cytochrome P450 (CYP450) family have alone been reported to affect the metabolism of up to approximately 25% of all drugs (Ingelman-Sundberg *et al.*, 2007).

Therefore the desirable therapeutic response to a drug is only elicited when both PK and PD genes are functioning as intended. However, variation in any of these genes may also result in “off target” effects that can cause adverse reactions (Karczewski *et al.*,

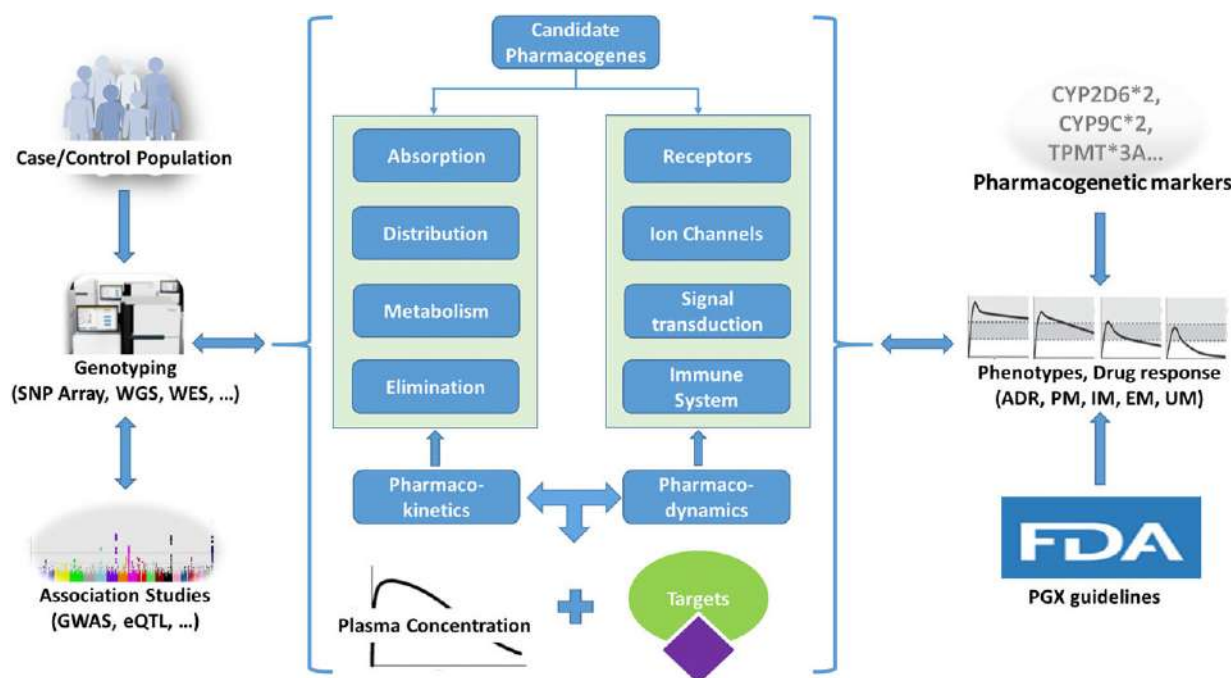


Fig. 1 A workflow of pharmacogenetics/-genomics. The workflow depicts the common steps typically taken to identify a pharmacogenetic allele responsible for a drug response in a population. Variants discovered through various genotyping platforms in patient and control groups are analyzed using the GWAS approach to identify probable causal candidate(s) in PK and/or PD genes. Functional details of these candidates need to be established and clinically validated before any clinical guidelines can be implemented.

2012). Such an “off target” effect may also be induced by a variant in a pharmacogene, which although itself is not involved in direct interaction with the drug, is a member gene of the drug metabolizing pathway.

In general, pharmacogenetic variability can be broadly classified into three groups:

1. Inherited Mendelian monogenic traits, disorders, or ADRs that are affected by usually single rare coding variants;
2. Oligogenic traits affected by a few PK and/or PD genes; and
3. Complex multifactorial traits affected by not only many small-effect pharmacogenetic variants but possibly also epigenetic and environmental factors (Zhang and Nebert, 2017).

The complexity in predicting and identifying the three categories of pharmacogenetics increases as the number of genes and factors increases, ranging from simple statistical association between genotypes and pharmacological phenotypes to massive whole genome- or exome-wide association studies.

Workflow in Pharmacogenetics

Increasingly, pharmacogenomics research involving large number of patients across different population groups is believed to hold the key to the resolution of low pharmacological efficacy observed in many of today’s drug treatments. Identification of pharmacogenetic variants usually adopts either the candidate gene or genome-wide approach. The former involves genotyping of either one or a few (sometimes up to hundreds of) genes using techniques ranging from PCR or SNP array to targeted next-generation sequencing (NGS). This is a workable approach when prior functional knowledge of the pharmacogenes and their related diseases or phenotypes is known. The latter, which sometimes is also known as the hypothesis-generating approach, uses various NGS methodologies to screen for genetic variants that include single-nucleotide variants (SNVs), insertion/deletions (indels), and structural variants (SVs) in the entire genome of patient (e.g., drug responders, ADR patients) and control (e.g., nonresponder, extensive, or normal drug metabolizers) groups (Fig. 1). No prior knowledge about any of the genetic variants is necessary for this approach. Although discovery in nature, the genome-wide approach is especially useful for studying complex traits/diseases that are influenced by multiple genetic factors. Compared with the candidate gene approach, this latter approach allows the systematic testing of more hypotheses in order to discover novel associations. Discussion on the advantages and disadvantages of the two approaches and the technologies involved is provided elsewhere (Peters et al., 2010; Yang et al., 2016; Cohn et al., 2017).

NGS technologies have the potential to discover millions of genetic variants in both diseased and healthy control individuals. Characterization of the effects of these variants on drug response forms the main objective of pharmacogenetic/-genomic studies. Whilst the number of methods and tools available for the analysis of genetic variants are too many to be adequately reviewed

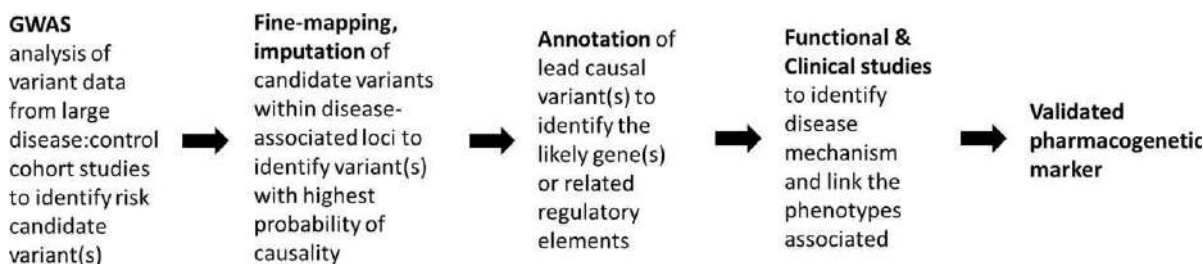


Fig. 2 A workflow of pharmacogenetics GWAS. True causal candidate variants identified during initial GWAS analysis are usually difficult to distinguish as they cluster with neutral variants in LD. They may also be missing from the list. Fine-mapping process, including filling in the variant gaps using imputation tools, is necessary to refine this list. Functional annotation and studies are next performed to decipher the biological mechanisms behind the association of the variants and drug response.

here (Omick, 2018; Henry *et al.*, 2014), some basic steps illustrated in Fig. 2 are commonly adopted, especially for the genome-wide approach.

In the last decade, genome-wide association studies (GWAS) have been widely and successfully used in the identification of single-nucleotide polymorphisms (SNPs) in genes that are associated with a variety of human phenotypes or traits, including that of pharmacogenetic significance (Cooper *et al.*, 2008; Takeuchi *et al.*, 2009; Giacomini *et al.*, 2017). A total of 333 entries on SNP-trait associations for “Response to drug” are now held in the GWAS Catalog (MacArthur *et al.*, 2017).

Statistically, GWAS analysis is basically a straightforward null hypothesis test to identify variant data that lies at the far ends of the null distribution. These are the SNPs in which associations with the tested pharmacological outcome are considered statistically significant. A few factors should be taken into consideration when a pharmacogenomics GWAS is planned. They include the allelic frequency of the SNPs, effect size (small- or large-) of SNPs on the phenotypes, the need of large sample sizes or cohorts to provide the necessary statistical power to identify the candidate SNPs, characteristics (*e.g.*, treatment regimen, patient features such as ethnicity, etc.) of the population studied, and study design (either observational study or randomized controlled trial) (Motsinger-Reif *et al.*, 2013). Furthermore, it is also important to realize that pharmacogenomics GWAS may differ from most other diseased traits in such a way that (1) due to the presence of clinically well-defined phenotypes with known underlying mechanisms, the pharmacogenomic outcomes are often clear; and (2) association can be established with greater statistical confidence since larger genetic effects may exist in some of the pharmacogenomic phenotypes.

Although many tools have been developed over the years to deal with different forms of GWAS data, some widely used analytical features include distribution of allelic frequencies, Hardy-Weinberg equilibrium, analysis of association between single SNPs and haplotypes with different traits, etc. The results of GWAS analyses are typically presented in a graph as either a QQ or a Manhattan plot of *P*-values representing the association test of each genetic variant across a chromosomal region (Uitterlinden, 2016; Reed *et al.*, 2015).

However, one of the biggest drawbacks of GWAS analysis, including that for pharmacogenomics, is the large number of resulting candidate genetic variants; many of which are clustered in linkage disequilibrium (LD) in the implicated loci. Here, the haplotype structure of these loci makes the distinction between neutral and causal variants difficult (Spain and Barrett, 2015). In addition, rarely do any of the GWAS variants fall within protein-coding sequences. Instead, the vast majority of them are found in the noncoding segments of the human genome where functional details may be incomplete or lacking (Farh *et al.*, 2015); despite efforts such as ENCODE (Encode_Project Consortium, 2011) and Fantom5 (Kawaji *et al.*, 2017) to annotate these regulatory regions. As a result, direct translation of GWAS variants into targetable pharmacogenetic markers is not always possible and further refinement of the list of candidate variants using methods such as fine-mapping and/or genotype is necessary.

In essence, fine-mapping aims to pinpoint the causal variants and/or genes that statistically are most likely to have a direct effect on the observed phenotype. However, before this can be done, many of the common variants in the loci should be genotyped or imputed with high confidence. This is especially important with GWAS summary data derived from medium- to low-coverage genotyping or whole genome sequencing datasets (Chan *et al.*, 2016). Using comprehensive genotype data such as the 1000 Genomes Project reference panels (1000_Genomes_Project Consortium *et al.*, 2012), imputation allows variant gaps missing from the genotyping arrays or NGS to be filled in. This much improved dataset can then undergo either a step-wise conditional analysis in which SNPs with the lowest *P*-value or LD are conditioned iteratively till no more SNP is found to reach the threshold of the preassigned *P*-value or analysis using Bayesian methods to assign each of the SNPs with posterior probabilities of causality (Spain and Barrett, 2015).

However, the value of a GWAS/fine-mapping exercise is not fully realized until the underlying functional basis of the refined causal genetic variants and/or the target gene can be resolved with informed biological inference through some post-GWAS analysis. Among them are expression quantitative trait loci analysis, which attempts to correlate genetic variants on regulatory regions with quantitative changes in the expression of genes that are under the control of affected elements (Schroder *et al.*, 2013; GTEx Consortium, 2015); genome-wide gene enrichment and pathway analysis to link a trait or phenotype to a pathway(s) instead of looking at single genes only (Postmus *et al.*, 2014; Di Iulio and Rotger, 2011); and protein-protein interaction analysis to predict drug repurposing (Pritchard *et al.*, 2017).

Both *in vitro* and *in vivo* assays can subsequently be run to validate and characterize the predicted functional mechanisms and roles of the candidate causal variants in the drug response. Clinical studies are finally done before any such variants can be confidently established as a pharmacogenetics marker in the population.

Some of the major bioinformatics tools used in the pharmacogenomics workflow outlined above are described in the next section.

Bioinformatics Resources for Pharmacogenetics/-Genomics

Bioinformatics Analysis Tools

GWAS are currently seen as one of the best methodologies for population analysis of pharmacogenetic polymorphisms. Advances in NGS technologies and the accompanying bioinformatics tools have allowed GWAS to be conducted with higher statistically precision. It is therefore not unexpected that nearly 500 tools have been developed to analyze various aspects of GWAS (Omicx, 2018). A few of the widely used ones are discussed below.

General GWAS analysis

PLINK: A free open-source whole genome GWAS toolset capable of performing analyses ranging from the very basic to the large-scale computationally demanding ones (Chang *et al.*, 2015; Purcell *et al.*, 2007). PLINK does not do any pre-GWAS analysis, such as genotype or CNV calling. It is a pure genotype/phenotype analytical tool with support for visualization, annotation, and management of results.

GenABEL: A software package based on the R statistical programming language; it is used for the statistical analyses of genome variation data (Aulchenko *et al.*, 2007). It contains subprograms to do genome-wide association (GWA) analysis (GenABEL, ProbABEL, MixABEL, OmicABEL), meta-analysis (MetABEL), parallelization of GWA analyses (ParallABEL), management of very large files (DatABEL), and evaluation of prediction (PredictABEL). It also provides tools for visualizing the statistical output.

Bioconductor: Another R open-source and open development effort to provide a comprehensive collection of tools to analyze different types of high-throughput genomic data (*e.g.*, microarray, NGS genotypes) (Huber *et al.*, 2015). A number of R programs are included in Bioconductor to deal with association studies, including GWASTools (Gogarten *et al.*, 2012) and traseR (Chen and Qin, 2016).

Imputation

IMPUTE: The program or its latest version IMPUTE2 is a tool used for genotype imputation and phasing in GWAS/fine-mapping studies. It is based on dense marker sets such as haplotypes from the 1000 Genomes Project (Howie *et al.*, 2009). IMPUTE2 can achieve a higher accuracy and allow information across multiple reference panels to be combined while remaining computationally feasible.

BEAGLE: A software package that performs “genotype calling, genotype phasing, imputation of ungenotyped markers, and identity-by-descent segment detection” (Browning and Browning, 2016; Browning and Browning, 2007). It is capable of analyzing big genetic datasets containing hundreds of thousands of genotypes on thousands of samples.

MACH: The core function of this package is to perform genotype imputation and haplotype phasing in samples of unrelated individuals within a variety of populations (Li *et al.*, 2010). The software achieves this by using the HapMap sample or other densely genotyped individuals (*e.g.*, from the 1000 Genomes Project) as a reference. Simple tests of association can also be run using MACH.

Functional annotation of variants

Variant Effect Predictor: A powerful toolset developed by ENSEMBL to analyze, annotate, and prioritize genomic variants (SNPs, insertions, deletions, CNVs, or SVs) in both coding and noncoding regions (McLaren *et al.*, 2016).

SnEff: A toolbox containing different programs to annotate and predict the effects of genetic variants on genes and proteins (Cingolani *et al.*, 2012). It also allows annotation using the GWAS catalog (Nhgri-Ebi, 2018).

ANNOVAR: A tool to efficiently annotate genetic variants (SNVs and indels) (Wang *et al.*, 2010). Using update-to-date information, it assigns “functional consequences of genes, inferring cytogenetic bands, reporting functional importance scores, finding variants in conserved regions, or identifying variants reported in the 1000 Genomes Project and dbSNP.”

Databases

For a long time, genetic variants associated with drug response have been deposited together with other data in various large-scale databases such as the dbSNP of NCBI and European Genome-Phenome Archive of EBI. However, the ever-increasing genetic and genomic data generated from population-wide sequencing and GWAS studies has seen the growth of dedicated pharmacogenomics-centric web resources. Some of these have been reviewed elsewhere (Zhang *et al.*, 2015; Potamias *et al.*, 2014; Sim *et al.*, 2011) and a few of the prominent ones are discussed here (Table 1).

PharmGKB (The Pharmacogenomics Knowledge Base) is a curated knowledge base providing organized information on the impacts of human genetic variations on drug response. It currently holds annotation for about 19,000 genetic variants (SNP, indel, repeat, haplotype). They are integrated into nearly 3500 genotype-phenotype relationships; 500 clinical prescriptions in the form of drug labelings, 98 genotype-based dosing recommendations, and 128 pathways showing the association of genotypes with PK

Table 1 Web-based Databases of Pharmacogenomics

Database (url)	Features	Reference
PharmGKB The Pharmacogenomics Knowledge base (https://www.pharmgkb.org/)	A curated knowledge base providing organized information on the impacts of human genetic variations on drug response. Among the vast resources, the database also contains clinical prescribing information detailing gene–drug associations, genotype–phenotype relationships, and dosing guidelines.	Whirl-Carrillo <i>et al.</i> (2012)
PharmVar Pharmacogene Variation Consortium (https://www.pharmvar.org/)	A central repository of haplotype structure and allelic variants of pharmacogenes. The database is a transition from the CYP-allele Database (http://www.cypalleles.ki.se/), which contains the nomenclature of human <i>CYP450</i> alleles.	Gaedigk <i>et al.</i> (2018); Sim <i>et al.</i> (2010)
DrugBank (https://www.drugbank.ca/)	A comprehensive bioinformatics and cheminformatics database containing information on drugs and drug targets. It also contains resources on pharmaco-genomics, -metabolomics, -transcriptomics, and -proteomics.	Wishart <i>et al.</i> (2018, 2006)
CPIC The Clinical Pharmacogenetics Implementation Consortium (https://cpicpgx.org/)	A detailed collection of regularly updated, peer-reviewed, and evidence-based clinical practice guidelines to facilitate the implementation of pharmacogenetic tests for patient care.	Relling and Klein (2011), Caudle <i>et al.</i> (2014)
SPHINX Sequence and Phenotype Integration Exchange (https://www.emergesphinx.org/)	A searchable catalog of inherited variants found in 82 important pharmacogenes of a large population cohort. In addition to the lists of genes, drugs, and pathways, SPHINX also contains such information as the implication of the genetic variations, drug interactions, and allelic frequencies of variants.	Rasmussen-Torvik, <i>et al.</i> (2014)
ePGA electronic Pharmacogenomics Assistant (http://www.epga.gr/)	A collection of public datasets linking the interactions between drug metabolizing status and the related haplotype information. The genotype-to-phenotype translation tool matches individual genotype profiles with known pharmacogenetics variation to give an indicative drug efficacy/toxicity summary.	Lakiotaki <i>et al.</i> (2016)
FDA Pharmacogenomic Biomarkers (https://www.fda.gov/Drugs/ScienceResearch/ucm572698.htm)	A table listing the pharmacogenomic biomarkers of FDA-approved drugs. Information includes clinical response, drug exposure variability, and dosing recommendation due either to germline or somatic variants of the pharmacogenes.	
PharmaADME Pharmacogenetics of Absorption, Distribution, Metabolism & Excretion genes (http://pharmaadme.org/)	An industry–academia effort that provides lists of standardized “evidence based” pharmacogenetics biomarkers that affect the absorption, distribution, metabolism, and excretion of drugs.	
PhID Pharmacology Interactions Database (http://phid.ditad.org/)	A database that allows visualization of complex network relationships among diseases, targets, drugs, genes, side effects, and pathways.	Deng <i>et al.</i> (2017)
GPCRdb G protein-coupled receptors (GPCRs) Database (http://gpcrdb.org/)	A comprehensive database containing data on genes, proteins, tertiary structures, ligand-receptors, and drug targets of GPCRs. The site also provides tools to select and study the impact of natural genetic variation for the drug targets of GPCRs.	Isberg <i>et al.</i> (2016), Hauser <i>et al.</i> (2018)
The NATs database The database of arylamine N-acetyltransferases (NATs) (http://nat.mbg.duth.gr/)	This database set up by The <i>NAT</i> Gene Nomenclature Committee covers the nomenclature and alleles of the polymorphic phase II xenobiotic metabolizing enzymes, arylamine NATs, from both eukaryotes, including those from the human, and prokaryotes.	Hein, <i>et al.</i> (2008)
PMT database The Pharmacogenetics of Membrane Transporters (http://pharmacogenetics.ucsf.edu/)	A database that provides information on both coding and noncoding genetic variants of solute carrier (SLC) transporters and ATP-binding cassette (ABC) transporters. Locations of the variants on the gene and secondary protein structure are also mapped. In addition, allelic frequencies in major racial and ethnic populations are also given.	Kroetz <i>et al.</i> (2010)
UGT-allele database The UDP-glucuronosyltransferase Allele Nomenclature (https://www.pharmacogenomics.pha.ulaval.ca/ugt-alleles-nomenclature/)	A catalogue of human <i>UGT1A</i> and <i>UGT2B</i> haplotypes and SNPs. Information of pharmacogenetic relevance include allelic names, nucleotide changes, protein effects, genotype–phenotypes.	Mackenzie <i>et al.</i> (2005)

and PD of drugs; and 64 Very Important Pharmacogenes (VIP) where there is considerable evidence on the pharmacogenomics of each of the genes.

PharmVar (Pharmacogene Variation Consortium) is a central repository of haplotype structure and allelic variants of pharmacogenes. This updated database is a transition from the Human Cytochrome P450 (CYP) Allele Nomenclature Database (Table 1), which housed the nomenclature and genotypes of the human CYP450 alleles.

DrugBank is a comprehensive bioinformatics and cheminformatics database that encompasses information such as PK, PD, ADRs, ADEMT, etc., of drugs and drug targets. In addition, it also contains multiomics (pharmacogenomics, -metabonomics, -transcriptomics and -proteomics) resources on the drug-gene interactions.

CPIC (*The Clinical Pharmacogenetics Implementation Consortium*) is a consortial effort to address the difficulty in translating pharmacogenetic findings into actionable prescribing information for clinicians. The database provides detailed collection of regularly updated, peer-reviewed, and evidence-based clinical practice guidelines to facilitate the implementation of pharmacogenetic tests for patient care. A total of 34 guidelines and updates have been published to date (January 2018).

SPHINX (*Sequence and Phenotype Integration Exchange*) is a searchable catalogue of inherited variants in 82 very important pharmacogenes identified from large population studies. In addition, the database also contains such information as the implication of the genetic variations, drug interactions, allelic frequencies of variants, and pathways of metabolism and diseases.

ePGA (*electronic Pharmacogenomics Assistant*) is a collection of public datasets used to provide associations between the haplotypes/alleles and possible ADME reactions. The genotype-to-phenotype translation tool matches individual genotype profiles with known pharmacogenetics variation to give an indicative drug efficacy/toxicity summary.

FDA Pharmacogenomic Biomarkers is a table listing FDA-approved drugs with labelings containing information on the associated pharmacogenomic biomarkers. Other information on the labeling includes clinical response, drug exposure variability, and dosing recommendation due either to germline or somatic variants of pharmacogenes.

e-PKGene integrates information from different resources to allow quantitative analysis of how genetic variants of drug metabolizing enzymes and transporters impact on pharmacokinetic responses to drugs and metabolites. Where available, summaries of the impact of variants on drugs and metabolites in different populations are also provided.

PharmaADME (*Pharmacogenetics of Absorption, Distribution, Metabolism & Excretion genes*) is an industry-academia effort that provides a consensus core list of standardized "evidence based" pharmacogenetic biomarkers that have been shown to affect the absorption, distribution, metabolism, and excretion of drugs.

PhID (*Pharmacology Interactions Database*) is a database that allows visualization of complex network relationships among diseases, targets, drugs, genes, side effects, and pathways.

SuperCYP is a Cytochrome P450 database containing genetic variants of CYP450 genes. The site also provide a tool to analyze interactions between CYP450 enzymes and drugs.

GPCRdb (*GPCRs Database*) is a comprehensive database containing data on genes, proteins, tertiary structures, ligand-receptor, and drug targets of GPCRs. The site also provides tools to select and study the impact of natural genetic variation for the drug targets of GPCRs.

The NATs database (*The database of arylamine N-acetyltransferases (NATs)*), set up by the NAT Gene Nomenclature Committee, covers the nomenclature and alleles of the polymorphic phase II xenobiotic metabolizing enzymes, arylamine NATs, from both eukaryotes, including those from the human, and prokaryotes.

PMT database (*the Pharmacogenetics of Membrane Transporters*) provides information on both coding and noncoding genetic variants of SLC transporters and ABC transporters. Locations of the variants on the gene and secondary protein structure are also mapped. In addition, allelic frequencies in major ethnic population groups are also given.

UGT-allele database (*The UDP-glucuronosyltransferase Allele Nomenclature*) is a catalogue of human UGT1A and UGT2B haplotypes and SNPs. Information of pharmacogenetic relevance includes allelic names, nucleotide changes, protein effects, genotype-phenotypes.

Pharmacogenetics in Practice

Although clinical implementation of pharmacogenomics is yet to be fully realized, there have been some success stories thus far. To date, the Clinical Pharmacogenetics Implementation Consortium has issued detailed practical guidelines for the application of pharmacogenetics in the therapeutic usage of 35 drugs (Relling and Klein, 2011; Drew, 2016). Among them is the genotyping of the highly variable major histocompatibility complex class I allele (HLA) (Becquemont, 2010) for abacavir (Watson *et al.*, 2009; Young *et al.*, 2008), carbamazepine (Phillips *et al.*, 2018), and allopurinol (Saito *et al.*, 2016; Hershfield *et al.*, 2013); the drugs prescribed for the treatment of HIV, epilepsy, and gout respectively. Approximately 5%–8% Caucasian and 0.26%–3.6% African American, Asian, and Hispanic HIV/AIDS patients with the HLA-B*57:01 allele are affected by the abacavir hypersensitivity syndrome during the first 6 weeks of drug administration (Ma *et al.*, 2010). Screening for this allele prior to drug therapy is therefore recommended by various healthcare authorities worldwide (Martin *et al.*, 2012). On the other hand, patients harboring the HLA-B*15:02 pharmacogenetic marker of Southeast Asian and Chinese origin are known to suffer from severe and potentially fatal carbamazepine-induced cutaneous adverse reactions such as toxic epidermal necrolysis (TEN) and Stevens–Johnson syndrome (SJS). This allele is present in approximately 10%–15% of the population groups in this region but <1% in the Japanese and Korean population; and it is largely absent in other non-Asian population groups (Chung *et al.*, 2004; Hung *et al.*, 2006; Kaniwa *et al.*, 2013). Similarly, allopurinol-induced TEN/SJS is more prevalent in Taiwanese and Southeast Asian patients with the HLA-B*58:01 allele (Hung *et al.*, 2005; Tassaneeyakul *et al.*, 2009).

It is estimated that almost 60%–70% of all breast cancers are estrogen receptor (ER)-positive and therefore, inhibitors of ER signaling, such as tamoxifen (TAM), remain as one of the most widely prescribed drug treatment options for patients (Hertz *et al.*, 2017). TAM has been shown to reduce the recurrence and mortality rates by one third in both pre- and postmenopausal women (Goetz *et al.*, 2008). However, approximately 40% of patients are unresponsive to the treatment due to interindividual genetic

variability (Hertz *et al.*, 2017). The highly polymorphic drug metabolizing gene *CYP2D6* (with > 160 alleles (Gaedigk *et al.*, 2018; Pharmgkb, 2017) has been implicated in this unsatisfactory treatment outcome (Rofaie *et al.*, 2010). For instance, *CYP2D6**10 allele, which corresponds to a decreased enzyme activity and intermediate drug metabolism, is more prevalent amongst Asians and Africans; with the phenotype amounting to about 38%–70% (Rofaie *et al.*, 2010). On the hand, the allele *CYP2D6**4, which corresponds to a nonfunctional variant and hence a poor metabolism (PM) phenotype, is more prevalent amongst Caucasians (Fernandez-Santander *et al.*, 2013) than Asians (Lim *et al.*, 2011). Due to its obvious medical benefit, the US FDA has in fact approved pharmacogenetic testings for *CYP2D6* variants prior to treatment with tamoxifen (Rofaie *et al.*, 2010).

In addition, pharmacogenomics research on a number of other drugs are currently undertaken by various large international consortia (Giacomini *et al.*, 2017). Translation of such efforts into more pharmacogenomic guidelines and clinical implementation should further expand the success stories.

Conclusions

The ability to interrogate genetic variation at a population scale over the last decade has seen a rapid paradigm shift in medicine, in particular the field of drug treatment, from the “one size fits all” approach to a more targeted and precise treatment tailored according to an individual’s genetic and genomic profile. Although most patients have been shown to harbor at least one actionable pharmacogenetic variant (Van Driest *et al.*, 2014), clinical prediction of drug response due to these genotypes has only been applied to relatively few drugs, which exclude even many of the most commonly prescribed ones (Relling and Evans, 2015; Caudle *et al.*, 2014; Janssens and Van Duijn, 2008). In most of these cases, pharmacogenetic guidelines are only provided to drugs whereby a small number of large-effect contributing genetic variants can be confidently associated with a clearly defined set of phenotypes such as efficacy effects and ADRs.

Pharmacogenetic association between drugs and complex traits, such as type 2 diabetes (Florez, 2017) and irritable bowel syndrome (Rufini *et al.*, 2017), usually involves multiple genetic factors and accurate deciphering of such relationship remains challenging. Not only are the genotype–phenotype association marginal in many of the complex diseases, some pharmacogenetic variants also work in combination to exert their effects (Zhang and Nebert, 2017). In addition, many of pharmacological phenotypes can further be complicated by changes in gene expression (Gamazon *et al.*, 2010), epigenetic regulation (Cacabelos and Torrellas, 2015; Pisanu *et al.*, 2018), and protein–protein interactions (Shiota *et al.*, 2008) brought about by genetic variation in either the pharmacogenes or their associated pathway genes.

Although advances in NGS and GWAS technologies have enabled many more pharmacogenetic variants to be identified through the running of large-scale cohort studies and some even at a whole-nation level (Gudbjartsson *et al.*, 2015), this deluge of data is also accompanied by increases in biases and noise of the data. Therefore, improved computational methodologies that are capable of handling veracity of big genomic data are needed.

More importantly, variants identified in many of these large sample size studies are not always going to benefit pharmacogenomic GWAS due to the simple fact that the prevalence, and hence the availability, of large sample size is usually not possible with multidrugs induced diseases having clear defined adverse pharmacological phenotypes. This much needed statistical resolution provided by the sample size is further reduced when the interethnic genetic variation is also taken into consideration. Although some progresses have recently been made to provide a glimpse into the pharmacogenomics of various population groups across the world (Ramos *et al.*, 2014; Giacomini *et al.*, 2017), the number of individuals with ADRs sequenced within each of the different ethnic groups remains small. The trend of sequencing and fine-mapping of genetic variants in other non-European and ancestral populations is, therefore, expected to continue. Technologies such as improved GWAS, as well as whole genome and exome NGS, will aid in better translation of ethnicity-specific pharmacogenetic/-genomic findings into clinical practice (Chan *et al.*, 2015; Popejoy and Fullerton, 2016).

The field of pharmacogenomics is developing rapidly. However, continuous national-scale genotyping of population groups and whole genome/exome GWAS alone are not likely to result in widespread clinical application of pharmacogenomics. Further technological and methodological advancement in areas such as long read sequencing, imputation, and precision fine-mapping are needed to increase the quantity and quality of the pharmacogenomics variants. Coupled with a more integrated approach encompassing multiomics system biology, healthcare records, and artificial intelligence, pharmacogenomic data derived from population-wide analysis is expected to ultimately speed up the introduction of stratified, if not personalized, drug treatment in the near future.

See also: Genetics and Population Analysis. Natural Language Processing Approaches in Bioinformatics

References

- Abecasis, G.R., Auton, A., Brooks, L.D., *et al.*, 2012. An integrated map of genetic variation from 1092 human genomes. *Nature* 491, 56–65.
- Aulchenko, Y.S., Ripke, S., Isaacs, A., Van Duijn, C.M., 2007. GenABEL: An R library for genome-wide association analysis. *Bioinformatics* 23, 1294–1296.
- Becquemont, L., 2010. HLA: A pharmacogenomics success story. *Pharmacogenomics* 11, 277–281.
- Browning, B.L., Browning, S.R., 2016. Genotype imputation with millions of reference samples. *American Journal of Human Genetics* 98, 116–126.

- Browning, S.R., Browning, B.L., 2007. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *American Journal of Human Genetics* 81, 1084–1097.
- Cacabelos, R., Torrellas, C., 2015. Epigenetics of aging and Alzheimer's disease: Implications for pharmacogenomics and drug response. *International Journal of Molecular Sciences* 16, 30483–30543.
- Caudle, K.E., Klein, T.E., Hoffman, J.M., *et al.*, 2014. Incorporation of pharmacogenomics into routine clinical practice: The Clinical Pharmacogenetics Implementation Consortium (CPIC) guideline development process. *Current Drug Metabolism* 15, 209–217.
- Chan, A.W., Hamblin, M.T., Jannink, J.L., 2016. Evaluating imputation algorithms for low-depth Genotyping-By-Sequencing (GBS) data. *PLOS ONE* 11, e0160733.
- Chan, S.L., Jin, S., Loh, M., Brunham, L.R., 2015. Progress in understanding the genomic basis for adverse drug reactions: A comprehensive review and focus on the role of ethnicity. *Pharmacogenomics* 16, 1161–1178.
- Chang, C.C., Chow, C.C., Tellier, L.C., *et al.*, 2015. Second-generation PLINK: Rising to the challenge of larger and richer datasets. *Gigascience* 4, 7.
- Chen, L., Qin, Z.S., 2016. traseR: An R package for performing trait-associated SNP enrichment analysis in genomic intervals. *Bioinformatics* 32, 1214–1216.
- Chung, W.H., Hung, S.I., Hong, H.S., *et al.*, 2004. Medical genetics: A marker for Stevens-Johnson syndrome. *Nature* 428, 486.
- Cingolani, P., Platts, A., Wang Le, L., *et al.*, 2012. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly (Austin)* 6, 80–92.
- Cohn, I., Paton, T.A., Marshall, C.R., *et al.*, 2017. Genome sequencing as a platform for pharmacogenetic genotyping: A pediatric cohort study. *NPJ Genomic Medicine* 2, 19.
- Cooper, G.M., Johnson, J.A., Langae, T.Y., *et al.*, 2008. A genome-wide scan for common genetic variants with a large influence on warfarin maintenance dose. *Blood* 112, 1022–1027.
- Deng, Z., Tu, W., Deng, Z., Hu, Q.N., 2017. PhD: An open-access integrated pharmacology interactions database for drugs, targets, diseases, genes, side-effects, and pathways. *Journal of Chemical Information and Modelling* 57, 2395–2400.
- Di Iulio, J., Rotger, M., 2011. Pharmacogenomics: What is next? *Frontiers in Pharmacology* 2, 86.
- Drew, L., 2016. Pharmacogenetics: The right drug for you. *Nature* 537, S60–S62.
- 2011.A user's guide to the encyclopedia of DNA elements (ENCODE). *PLOS Biology* 9, e1001046.
- Farh, K.K., Marson, A., Zhu, J., *et al.*, 2015. Genetic and epigenetic fine mapping of causal autoimmune disease variants. *Nature* 518, 337–343.
- Fda, 2017. *Table of Pharmacogenomic Biomarkers in Drug Labeling* [Online]. Available at: <https://www.fda.gov/Drugs/ScienceResearch/ucm572698.htm> (accessed 06.12.2017).
- Fernandez-Santander, A., Gaibar, M., Novillo, A., *et al.*, 2013. Relationship between genotypes Sult1a2 and Cyp2d6 and tamoxifen metabolism in breast cancer patients. *PLOS ONE* 8, e70183.
- Florez, J.C., 2017. The pharmacogenetics of metformin. *Diabetologia* 60, 1648–1655.
- Gaedigk, A., Ingelman-Sundberg, M., Miller, N.A., *et al.*, 2018. The Pharmacogene Variation (PharmVar) Consortium: Incorporation of the Human Cytochrome P450 (CYP) allele nomenclature database. *Clinical Pharmacology Therapeutics* 103 (3), 399–401.
- Gamazon, E.R., Huang, R.S., Cox, N.J., Dolan, M.E., 2010. Chemotherapeutic drug susceptibility associated SNPs are enriched in expression quantitative trait loci. *Proceedings of the National Academy of Sciences of the United States of America* 107, 9287–9292.
- Giacomini, K.M., Yee, S.W., Mushiroda, T., *et al.*, 2017. Genome-wide association studies of drug response and toxicity: An opportunity for genome medicine. *Nature Reviews Drug Discovery* 16, 1.
- Goetz, M.P., Kamal, A., Ames, M.M., 2008. Tamoxifen pharmacogenomics: The role of CYP2D6 as a predictor of drug response. *Clinical Pharmacology Therapeutics* 83, 160–166.
- Gogarten, S.M., Bhargale, T., Conomos, M.P., *et al.*, 2012. GWASTools: An R/Bioconductor package for quality control and analysis of genome-wide association studies. *Bioinformatics* 28, 3329–3331.
- 2015.Human genomics. The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science* 348, 648–660.
- Gudbjartsson, D.F., Helgason, H., Gudjonsson, S.A., *et al.*, 2015. Large-scale whole-genome sequencing of the Icelandic population. *Nature Genetics* 47, 435–444.
- Hauser, A.S., Chavali, S., Masuho, I., *et al.*, 2018. Pharmacogenomics of GPCR drug targets. *Cell* 172 (1–2), 41–54.
- Hein, D.W., Boukouvala, S., Grant, D.M., Minchin, R.F., Sim, E., 2008. Changes in consensus arylamine N-acetyltransferase gene nomenclature. *Pharmacogenet Genomics* 18, 367–368.
- Henry, V.J., Bandrowski, A.E., Pepin, A.S., Gonzalez, B.J., Desfeux, A., 2014. OMICtools: An informative directory for multi-omic data analysis. *Database (Oxford)* 2014, bau069.
- Hershfield, M.S., Callaghan, J.T., Tassaneeyakul, W., *et al.*, 2013. Clinical Pharmacogenetics Implementation Consortium guidelines for human leukocyte antigen-B genotype and allopurinol dosing. *Clinical Pharmacology and Therapeutics* 93, 153–158.
- Hertz, D.L., Kidwell, K.M., Hilsenbeck, S.G., *et al.*, 2017. CYP2D6 genotype is not associated with survival in breast cancer patients treated with tamoxifen: Results from a population-based study. *Breast Cancer Research and Treatment* 166, 277–287.
- Hill, S., 2015. Taking a more personalised approach to diagnosis and care [Online]. Available: <https://www.england.nhs.uk/blog/sue-hill/> (accessed 20.07.2017).
- Howie, B.N., Donnelly, P., Marchini, J., 2009. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLOS Genetics* 5, e1000529.
- Huber, W., Carey, V.J., Gentleman, R., *et al.*, 2015. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* 12, 115–121.
- Hung, S.I., Chung, W.H., Jee, S.H., *et al.*, 2006. Genetic susceptibility to carbamazepine-induced cutaneous adverse drug reactions. *Pharmacogenet Genomics* 16, 297–306.
- Hung, S.I., Chung, W.H., Liou, L.B., *et al.*, 2005. HLA-B*5801 allele as a genetic marker for severe cutaneous adverse reactions caused by allopurinol. *Proceedings of the National Academy of Sciences of the United States of America* 102, 4134–4139.
- Ingelman-Sundberg, M., Sim, S.C., Gomez, A., Rodriguez-Antona, C., 2007. Influence of cytochrome P450 polymorphisms on drug therapies: Pharmacogenetic, pharmacoeconomic and clinical aspects. *Pharmacology and Therapeutics* 116, 496–526.
- Isberg, V., Mordalski, S., Munk, C., *et al.*, 2016. GPCRdb: An information system for G protein-coupled receptors. *Nucleic Acids Research* 44, D356–D364.
- Janssens, A.C., Van Duijn, C.M., 2008. Genome-based prediction of common diseases: Advances and prospects. *Human Molecular Genetics* 17, R166–R173.
- Kaniwa, N., Sugiyama, E., Saito, Y., *et al.*, 2013. Specific HLA types are associated with antiepileptic drug-induced Stevens-Johnson syndrome and toxic epidermal necrolysis in Japanese subjects. *Pharmacogenomics* 14, 1821–1831.
- Karczewski, K.J., Daneshjou, R., Altman, R.B., 2012. Chapter 7: Pharmacogenomics. *PLOS Computational Biology* 8, e1002817.
- Kawaji, H., Kasukawa, T., Forrest, A., Carninci, P., Hayashizaki, Y., 2017. The FANTOM5 collection, a data series underpinning mammalian transcriptome atlases in diverse cell types. *Scientific Data* 4, 170113.
- Kroetz, D.L., Yee, S.W., Giacomini, K.M., 2010. The pharmacogenomics of membrane transporters project: Research at the interface of genomics and transporter pharmacology. *Clinical Pharmacology and Therapeutics* 87, 109–116.
- Lakiotiaki, K., Kartsaki, E., Kanterakis, A., *et al.*, 2016. ePGA: A Web-Based Information System for Translational Pharmacogenomics. *PLOS ONE* 11, e0162801.
- Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R., 2010. MaCH: Using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic Epidemiology* 34, 816–834.
- Lim, J.S., Chen, X.A., Singh, O., *et al.*, 2011. Impact of CYP2D6, CYP3A5, CYP2C9 and CYP2C19 polymorphisms on tamoxifen pharmacokinetics in Asian breast cancer patients. *British Journal of Clinical Pharmacology* 71, 737–750.
- MacArthur, J., Bowler, E., Cerezo, M., *et al.*, 2017. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). *Nucleic Acids Research* 45, D896–D901.

- Mackenzie, P.I., Bock, K.W., Burchell, B., *et al.*, 2005. Nomenclature update for the mammalian UDP glycosyltransferase (UGT) gene superfamily. *Pharmacogenet Genomics* 15, 677–685.
- Martin, M.A., Klein, T.E., Dong, B.J., *et al.*, 2012. Clinical pharmacogenetics implementation consortium guidelines for HLA-B genotype and abacavir dosing. *Clinical Pharmacology and Therapeutics* 91, 734–738.
- Ma, J.D., Lee, K.C., Kuo, G.M., 2010. HLA-B*5701 testing to predict abacavir hypersensitivity. *PLOS Currents* 2, RRN1203.
- McLaren, W., Gil, L., Hunt, S.E., *et al.*, 2016. The Ensembl Variant Effect Predictor. *Genome Biology* 17, 122.
- Mizzi, C., Dalabira, E., Kumuthini, J., *et al.*, 2016. A European spectrum of Pharmacogenomic biomarkers: Implications for clinical pharmacogenomics. *PLOS ONE* 11, e0162866.
- Motsinger-Reif, A.A., Jorgenson, E., Relling, M.V., *et al.*, 2013. Genome-wide association studies in pharmacogenomics: Successes and lessons. *Pharmacogenet Genomics* 23, 383–394.
- Nhgri-Ebi. 2018. GWAS Catalog [Online]. Available at: <http://www.ebi.ac.uk/gwas>. (accessed).
- Omick, 2018. OMICtools [Online]. Available at: <https://omictools.com/whole-genome-resequencing-category>. (accessed 17.02.2018).
- Peters, B.J., Rodin, A.S., De Boer, A., Maitland-Van Der Zee, A.H., 2010. Methodological and statistical issues in pharmacogenomics. *Journal of Pharmacy and Pharmacology* 62, 161–166.
- Pharmgkb, 2017. PharmGKB: The Pharmacogenomics Knowledgebase [Online]. Available: <https://www.pharmgkb.org/>. (accessed 28.12.2017).
- Phillips, E.J., Sukasem, C., Whirl-Carrillo, M., *et al.*, 2018. Clinical Pharmacogenetics Implementation Consortium guideline for HLA genotype and use of carbamazepine and oxcarbazepine: 2017 update. *Clinical Pharmacology Therapeutics* 103 (4), 574–581.
- Pisanu, C., Katsila, T., Patrinos, G.P., Squassina, A., 2018. Recent trends on the role of epigenomics, metabolomics and noncoding RNAs in rationalizing mood stabilizing treatment. *Pharmacogenomics* 19, 129–143.
- Popejoy, A.B., Fullerton, S.M., 2016. Genomics is failing on diversity. *Nature* 538, 161–164.
- Postmus, I., Trompet, S., Deshmukh, H.A., *et al.*, 2014. Pharmacogenetic meta-analysis of genome-wide association studies of LDL cholesterol response to statins. *Nature Communications* 5, 5068.
- Potamias, G., Lakiotaki, K., Katsila, T., *et al.*, 2014. Deciphering next-generation pharmacogenomics: An information technology perspective. *Open Biology* 4, 140071.
- Pritchard, J.E., O'mara, T.A., Glubb, D.M., 2017. Enhancing the Promise of Drug Repositioning through Genetics. *Frontiers in Pharmacology* 8, 896.
- Purcell, S., Neale, B., Todd-Brown, K., *et al.*, 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *American Journal of Human Genetics* 81, 559–575.
- Ramos, E., Doumatey, A., Elkahoun, A.G., *et al.*, 2014. Pharmacogenomics, ancestry and clinical decision making for global populations. *The Pharmacogenomics Journal* 14, 217–222.
- Rasmussen-Torvik, L.J., Stallings, S.C., Gordon, A.S., *et al.*, 2014. Design and anticipated outcomes of the eMERGE-PGx project: A multicenter pilot for preemptive pharmacogenomics in electronic health record systems. *Clinical Pharmacology Therapy* 96, 482–489.
- Reed, E., Nunez, S., Kulp, D., *et al.*, 2015. A guide to genome-wide association analysis and post-analytic interrogation. *Statistics in Medicine* 34, 3769–3792.
- Relling, M.V., Evans, W.E., 2015. Pharmacogenomics in the clinic. *Nature* 526, 343–350.
- Relling, M.V., Klein, T.E., 2011. CPIC: Clinical Pharmacogenetics Implementation Consortium of the pharmacogenomics research network. *Clinical Pharmacology Therapeutics* 89, 464–467.
- Rofaïel, S., Muo, E.N., Mousa, S.A., 2010. Pharmacogenetics in breast cancer: Steps toward personalized medicine in breast cancer management. *Pharmacogenomics and Personalized Medicine* 3, 129–143.
- Rufini, S., Ciccacci, C., Novelli, G., Borgiani, P., 2017. Pharmacogenetics of inflammatory bowel disease: A focus on Crohn's disease. *Pharmacogenomics* 18, 1095–1114.
- Saito, Y., Stamp, L.K., Caudle, K.E., *et al.*, 2016. Clinical Pharmacogenetics Implementation Consortium (CPIC) guidelines for human leukocyte antigen B (HLA-B) genotype and allopurinol dosing: 2015 update. *Clinical Pharmacology and Therapeutics* 99, 36–37.
- Schork, N.J., 2015. Personalized medicine: Time for one-person trials. *Nature* 520, 609–611.
- Schroder, A., Klein, K., Winter, S., *et al.*, 2013. Genomics of ADME gene expression: Mapping expression quantitative trait loci relevant for absorption, distribution, metabolism and excretion of drugs in human liver. *The Pharmacogenomics Journal* 13, 12–20.
- Shiota, M., Kusakabe, H., Hikita, Y., *et al.*, 2008. Pharmacogenomics of cardiovascular pharmacology: Molecular network analysis in pleiotropic effects of statin – An experimental elucidation of the pharmacologic action from protein-protein interaction analysis. *Journal of Pharmacological Sciences* 107, 15–19.
- Sim, S.C., Altman, R.B., Ingelman-Sundberg, M., 2011. Databases in the area of pharmacogenetics. *Human Mutation* 32, 526–531.
- Sim, S.C., Ingelman-Sundberg, M., 2010. The Human Cytochrome P450 (CYP) Allele Nomenclature website: A peer-reviewed database of CYP variants and their associated effects. *Human Genomics* 4, 278.
- Spain, S.L., Barrett, J.C., 2015. Strategies for fine-mapping complex traits. *Human Molecular Genetics* 24, R111–R119.
- Spear, B.B., Heath-Chiozzi, M., Huff, J., 2001. Clinical application of pharmacogenetics. *Trends in Molecular Medicine* 7, 201–204.
- Sultana, J., Cutroneo, P., Trifiro, G., 2013. Clinical and economic burden of adverse drug reactions. *Journal of Pharmacology and Pharmacotherapeutics* 4, S73–S77.
- Takeuchi, F., McGinnis, R., Bourgeois, S., *et al.*, 2009. A genome-wide association study confirms VKORC1, CYP2C9, and CYP4F2 as principal genetic determinants of warfarin dose. *PLOS Genetics* 5, e1000433.
- Tassaneeyakul, W., Jantararoungtong, T., Chen, P., *et al.*, 2009. Strong association between HLA-B*5801 and allopurinol-induced Stevens-Johnson syndrome and toxic epidermal necrolysis in a Thai population. *Pharmacogenet Genomics* 19, 704–709.
- Uitterlinden, A.G., 2016. An Introduction to Genome-Wide Association Studies: GWAS for Dummies. *Seminars in Reproductive Medicine* 34, 196–204.
- Van Driest, S.L., Shi, Y., Bowton, E.A., *et al.*, 2014. Clinically actionable genotypes among 10,000 patients with preemptive pharmacogenomic testing. *Clinical Pharmacology & Therapeutics* 95, 423–431.
- Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38, e164.
- Watson, M.E., Patel, L.G., Ha, B., *et al.*, 2009. A study of HIV provider attitudes toward HLA-B 5701 testing in the United States. *AIDS Patient Care and STDs* 23, 957–963.
- Whirl-Carrillo, M., McDonagh, E.M., Hebert, J.M., *et al.*, 2012. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics* 92, 414–417.
- Wishart, D.S., Feunang, Y.D., Guo, A.C., *et al.*, 2018. DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Research* 46 (D1), D1074–D1082.
- Wishart, D.S., Knox, C., Guo, A.C., *et al.*, 2006. DrugBank: A comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research* 34, D668–D672.
- Yang, W., Wu, G., Broeckel, U., *et al.*, 2016. Comparison of genome sequencing and clinical genotyping for pharmacogenes. *Clinical Pharmacology & Therapeutics* 100, 380–388.
- Young, B., Squires, K., Patel, P., *et al.*, 2008. First large, multicenter, open-label study utilizing HLA-B*5701 screening for abacavir hypersensitivity in North America. *AIDS* 22, 1673–1675.
- Zhang, G., Nebert, D.W., 2017. Personalized medicine: Genetic risk prediction of drug response. *Pharmacology and Therapeutics* 175, 75–90.
- Zhang, G., Zhang, Y., Ling, Y., Jia, J., 2015. Web resources for pharmacogenomics. *Genomics Proteomics Bioinformatics* 13, 51–54.

Detecting and Annotating Rare Variants

Jieming Chen, Genentech Inc., South San Francisco, CA, United States

Akdes S Harmanci and Arif O Harmanci, The University of Texas Health Science Center at Houston, Houston, TX, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

For the past decade, genome-wide association studies (GWASes) have been the focus of large-scale genetic studies that are used to characterize the genomic architecture of traits and diseases (Visscher *et al.*, 2012). These GWASes have been largely brought about by the use of single nucleotide polymorphism (SNP) microarrays, coupled with a wealth of computational tools to efficiently process and analyze the high throughput data on a large scale. SNP arrays have enabled the easy genotyping of millions of variants in the genomes of hundreds of individuals (International HapMap Consortium *et al.*, 2007; The International HapMap 3 Consortium, 2010). As they typically need to know the sequence of these genomic locations to design the array probes, the variants detected are mainly common polymorphisms, which are defined to be at a minor allele frequency (MAF) of at least 5% in the human population. In general, less common polymorphisms are between MAF 1% and 5%, and rare variants are variation that occur with MAF of less than 1% in the human population.

As of 2016, the GWAS catalog from the National Human Genome Research Institute (NHGRI) houses published GWASes from about 2500 publications, for over 24000 SNP-trait associations (MacArthur *et al.*, 2017). Despite this large number of trait and disease associations with common variants, much of the genetic contributions to these traits and diseases remain unexplained. Due to the initial focus on the common variants, variants with lower MAF in the human population have been thought to account for at least part of this “missing heritability” (Zuk *et al.*, 2014). The 1000 Genomes Project sequenced 2504 healthy individuals from 26 ethnic populations. They found that while rare variation comprises only 5% of the total variation per individual, collectively, rare germline variants are in abundance in the general human population (1000 Genomes Project Consortium *et al.*, 2015, 2012). Previously, rare variants have already been shown to play significant roles in human diseases and traits. Highly penetrant rare variants are known to cause Mendelian disorders; such as, cystic fibrosis (Castaldo *et al.*, 1999; Soe and Gregoire-Bottex, 2017) and sickle cell anemia (Higgs and Wood, 2008); and common (and complex) diseases; such as, autism (Griswold *et al.*, 2015) and Type 2 Diabetes (Steinthorsdottir *et al.*, 2014). There is, however, an ascertainment bias that partially hindered the research of rare variants. This bias is largely due to the difficulty in detecting and identifying these variants for genotyping probe creation in microarrays, coupled with the appeal of finding and publishing the “low-hanging fruits” in association studies; that is, the associations of traits with more easily detected common variants. For these reasons, the community have, for a while, largely neglected this lower end of the allele frequency spectrum.

In more recent years, the advent and maturation of short-read DNA sequencing technologies have engendered the sequencing of entire genomes of thousands of individuals that can be interrogated multiple times (more deeply). Consequently, rare genomic variants can now be detected per individual more easily. An entire ecosystem of bioinformatic algorithms have also been developed to meet the challenges of automation, and facilitating the procedures of data collection, processing, visualization, and computational analyses. In this article, we will introduce a non-exhaustive array of ways that bioinformatics and computational biology have been harnessed to create tools and methodologies, specifically in detecting and annotating rare variation.

Detection of Rare Variation

Over the years, there has been evolving connotations and meanings of the use of terminologies to define ‘nucleotide changes’, such as ‘mutation’ and ‘polymorphism’ (Condit *et al.*, 2002; Karki *et al.*, 2015). This created confusion depending on the context such as the effect (disease-causing or neutral), allelic frequency (rare or polymorphic), and inheritance pattern (somatic or germline). In order to prevent further confusion, the American College of Medical Genetics and Genomics (ACMG), the Human Variome Project, the Human Genome Variation Society (HGVS) and the Human Genome Organization (HUGO) have published recommendations and guidelines in favor of the use of more neutral terms such as ‘variant’, ‘alteration’, and ‘allelic variant’, to refer more generically to nucleotide changes in the genome, which include both natural variations and mutations (Richards *et al.*, 2015; den Dunnen *et al.*, 2016; Li *et al.*, 2017). Henceforth, we will be referring to any generic nucleotide change as a ‘variant’.

Broadly, there are three types of variants in the human genome. These are single nucleotide substitutions (SNVs), small insertions/deletions (indels) of nucleotides, and structural variants that comprise translocation and/or inversion of long chunks of DNA sequences (SVs). In addition, SVs may cause amplification or loss of large chunks of DNA sequences. These are referred to as copy number variants (CNVs). Each variant type is then detected (or “called”) using different bioinformatics methodologies. While the computational tools for the detection of rare variants are mostly generic variant calling methodologies for arrays and sequencing technologies, they have to be customized with appropriate considerations to cater to the specific underlying array and sequencing technologies, in order to more accurately detect variants with lower frequencies. Additionally, the use of *in silico* genotype imputation has also been proven to be extremely accurate in many instances, and thus provides a more cost-effective complement to array, and targeted and exome sequencing options. Hence, we will first review some of the challenges and strategies

in rare variant detection. We will then discuss, in turn, the application of bioinformatics in microarray, sequencing and imputation technologies. Finally, there are non-germline variants that are also rare, for instance, somatic mutations. Hence, we include a section on somatic mutation calling in cancer research.

Challenges and Strategies in Rare Variant Detection

Rare variants occur in very low frequency in the population. Various human genetics efforts have defined the threshold of rarity slightly differently. For example, the 1000 Genomes Project defines $<0.5\%$ to be rare ([The 1000 Genomes Project Consortium, 2012, 2015](#)), while others have used $<1\%$ as the threshold ([Agarwala et al., 2013](#); [Chouraki and Seshadri, 2014](#); [Bomba et al., 2017](#)). The low occurrence of these variants in the population is, in itself, a challenge that initially hampered the high throughput detection of rare and novel variants using genotyping arrays. The advent of sequencing enabled *de novo* rare variant detection on a large scale. In order to detect rare variants, population scale sequencing is typically performed, followed by using efficient bioinformatic pipelines to pool all the sequenced reads and perform variant calling on the pooled data. In this approach, the sequencing errors (such as polymerase chain reaction amplifications) do not affect the variant calling because they are mostly independent between individuals, and using pooled data alleviates the severity of these errors. The computational tools for the pooled variant calling procedure are parametrized to have high sensitivity. Thus, pooled variant calling increases power to detect rare variants that have very low frequencies, and also help decrease the false positive rates of detection ([1000 Genomes Project Consortium et al., 2015](#)). The cost of sequencing can be decreased by performing a low coverage sequencing in each individual. This may, however, come at the cost of reduced power to detect rare variants.

In many cases, in order to circumvent the issue of cost and coverage, it may be useful to perform targeted sequencing; for example, exome sequencing. In such a way, the cost is decreased but only a particular region or regions of interest are sequenced; using the example of exome sequencing, only the exons are sequenced. This approach increases the power to detect variants (e.g., on the exons) by trading read depth coverage for genomic coverage. One downside of this approach is that the target capture technologies have non-uniform efficiency among different parts of the genome, for example, across different exons ([Wang et al., 2017](#); [Clark et al., 2011](#)). This causes non-uniform genomic coverage, and complicates computational detection of some variants, such as in the detection of CNVs, which look for duplication or deleted regions ([Wu et al., 2012](#)). This approach is also not easily applicable for identifying complex SVs because of the low concordance between different methodologies and high false positive error rates. However, some studies have found that whole genome sequencing is less biased in identifying variants than exome sequencing ([Belkadi et al., 2015](#)).

An important aspect of detecting rare variants is also the characterization of low frequency haplotypes in the human genome. In order to identify rare haplotypes, it is necessary to have deeply catalogued population-specific genetic panels. Using these panels, the bioinformatics pipelines for variant calling can be combined with genetic imputation to detect rare variants more accurately (please refer to Section ‘Genotype Imputation’). This greatly depends on our ability to build high quality population panels ([The International HapMap 3 Consortium, 2010](#)). Also, this approach may not work well for SVs and CNVs. CNVs do not correlate highly with other variants in the linkage disequilibrium blocks. For complex SVs, the understanding of their recombination rates and LD block is not yet very well-developed ([Wineinger et al., 2011](#)). Altogether, these render the imputation of SV and CNV genotypes less accurate than SNVs and indels.

Nonetheless, the use of sequencing technologies has spurred complementary strategies in microarrays, targeted sequencing and imputation to detect rare variants.

Microarrays

Microarrays require the knowledge of the genomic sequences for probe design. Hence, the initial microarrays for variant detection were more catered for common variants. With next-generation sequencing (NGS) efforts in obtaining rare variant sequences, many arrays have incorporated rare variant probes detected from NGS. This is also implemented in a more targeted, and context-specific manner, such as focusing on certain diseases, ethnicities or genomic regions. For example, bespoke customized genotyping arrays such as the Immunochip ([Cortes and Brown, 2011](#)), Illumina ExomeChip ([Illumina, URL available in references](#)), and the UK Biobank Axiom Array ([Affymetrix, URL available in references](#)) have probes that are specially designed for rare single nucleotide variants in different contexts. For larger SVs, the array comparative genomic hybridization technology ([Banerjee, 2013](#)) and the more recent SNP arrays from Illumina and Affymetrix have been used for detection of rare CNVs or SVs. These custom-built microarray chips allow for a cheaper and more focused option for the detection of rare variants. Moreover, rare variant detection in microarrays can conveniently capitalize on the bioinformatics methods and tools made for microarrays that have been well-characterized, and -utilized for almost two decades ([LaFramboise, 2009](#)).

In microarray data, preprocessing of the signal intensities from the probes consists of multiple steps, and needs to be executed before genotype calling. These include *in silico* background correction, resulting from technical artefacts such as non-specific probe hybridization ([Owzar et al., 2008](#)); normalization of the probes intensities, within and between chips, so that intensities can be compared across all chips ([Carvalho et al., 2007](#); [Hu and He, 2007](#); [Staaf et al., 2008](#)); removal of batch effects ([Miclous et al., 2010](#)); quality control and outlier removal ([Guo et al., 2014](#)). There are bioinformatics tools that implement statistical methods for these preprocessing steps. For example, Robust Multiarray Analysis (RMA) uses parametric background correction, quantile

normalization and robust fitting of a log-linear additive model (Irizarry *et al.*, 2003). There are also R packages conveniently available in Bioconductor (Huber *et al.*, 2015) streamlined for analyses in R (Morgan, 2016; Gentleman *et al.*, 2005). Following preprocessing, genotype calling tools can be implemented for clustering to call genotypes. These have included open source tools from academia (Teo, 2012; Wellcome Trust Case Control Consortium, 2007; Wang *et al.*, 2007), and proprietary software from companies such as BeadStudio, and Affymetrix clustering algorithms for different microarray platforms.

Exome/Whole Genome Sequencing

For variant detection in NGS technologies, the reads from sequencing experiments are mapped on the reference genome using short read mapping tools like BWA (Li *et al.*, 2008) and Bowtie (Langmead and Salzberg, 2012). In SNV detection, the SNVs are detected by finding the single nucleotide differences in the mapped reads compared to the reference genome (Fig. 1). Indels are detected by comparing the reference genome and the mapped reads to identify the insertions/deletions of nucleotides in the mapped reads (Fig. 1). The identification of SVs (and CNVs) requires detection of the breakpoints that correspond to DNA positions that formed the SVs. The mapped reads are analyzed to find breakpoints at the ends of SVs (Fig. 1). Currently, analysis of the paired end reads and fine mapping of breakpoints by splitting, and mapping the reads (split read mapping) are the most reliable ways to identify SVs. Finally, an SV is identified by careful analysis of mapping patterns of paired-end reads. These strategies are utilized by bioinformatics algorithms for detecting SVs, such as DELLY, PEMer, BreakDancer, and GenomeStrip (Rausch *et al.*, 2012; Korbel *et al.*, 2009; Alkan *et al.*, 2011; Handsaker *et al.*, 2011; Chen *et al.*, 2009). For the SVs that amplify and delete chunks of DNA, i.e., CNVs, analysis of the read depth has been successful, which is used by CNVnator to detect CNVs (Abyzov *et al.*, 2011).

The sensitivity of bioinformatics tools for variant detection is variable among different types of variants. Sensitivity of SNV detection using short read sequencing technologies is very high (1000 Genomes Project Consortium *et al.*, 2012, 2015). GATK (McKenna *et al.*, 2010) is currently the standard bioinformatics tool that is used for detecting SNVs. VarScan (Koboldt *et al.*, 2009) is another popular variant detection tool. The combination of the software packages GATK and dindel (Albers *et al.*, 2011) has been used in indel detection. The sensitivity of indel detection is high, albeit the false positive rates can be high as well (Hasan *et al.*, 2015). The accuracy of SV detection is hard to quantify as they differ depending on contexts, such as in diseased and normal individuals. This is mainly because there are currently no appropriate ground-truth references for SVs. Nonetheless, large-scale projects like the 1000 Genomes (1000 Genomes Project Consortium *et al.*, 2012) and The Cancer Genome Atlas (Collins, 2007) provide extremely useful SV sets for healthy and diseased genomes. In these datasets, it has been shown that the concordance of identified variants among different methods is generally modest (Tattini *et al.*, 2015). The most reliable strategy to identify SVs is to use multiple bioinformatics tools separately to detect the SVs. The SVs identified by each method are then merged. This method is effective in removing false positives at the expense of decreased sensitivity (Sudmant *et al.*, 2015).

The major hurdles for accurate variant calling can be divided into two categories. First, certain parts of human genome are hard to sequence experimentally. For example, DNA in heterochromatin regions cannot be easily fragmented while DNA library is being prepared (Shendure and Ji, 2008). A second hurdle is a bioinformatics problem, that is, a large fraction of the human genome is difficult to align to, because it contains repeated DNA sequences (de Koning *et al.*, 2011). This includes the transposable elements (such as Alu's), and segmental duplications of large chunks of human genome (Bailey and Eichler, 2006). The short reads that are sequenced from these regions cannot be uniquely mapped. Therefore, the variants in these regions are very hard, if not impossible, to detect. There are already bioinformatics methods developed to probabilistically assign reads in repeat regions. Even though these mitigate the effects of low mappability to a certain degree, short-read mapping to repeat regions remains an ongoing challenge.

The most promising (and logical) direction for the identification of variants in hard-to-map regions are long-read technologies, which include PacBio (Rhoads and Au, 2015), 10X (Zheng *et al.*, 2016), Illumina's Molecule (McCoy *et al.*, 2014), and Oxford Nanopore (Feng *et al.*, 2015). However, these long-read technologies often have different strengths and weaknesses. For example, PacBio generates very long reads, but with high error rates, while the 10X technology creates linked reads that have very low error rates, but are not exactly contiguous. There have been some bioinformatics tools that calibrate errors and call variants using data from long-read technologies (English *et al.*, 2014). Integrative bioinformatics methods are also in development, which utilize multiple long-read technologies in concert, to identify highly complex SVs and variants in the repeat regions and highly polymorphic regions, such as HLA locus (Nelson *et al.*, 2015). In addition, graph-based bioinformatics tools that use de-bruijn graphs have been used successfully in whole genome assembly and transcriptome assembly of long-read sequencing data (Lin *et al.*, 2016; Gordon *et al.*, 2016). New bioinformatics methods are utilized in cancer genomics to computationally de-convolve complex structural variation patterns in unstable cancer genomes (Zheng *et al.*, 2016) (please refer to the Section 'Detection of somatic variants in cancer genomes' for more discussion). Hence, long-read technologies, while still yielding notable issues, are nonetheless very powerful, especially when they are used in conjunction with traditional NGS (Mostovoy *et al.*, 2016).

Genotype Imputation

Genotype imputation is the process of predicting the genotypes of genetic variants that were not directly interrogated experimentally. It was conceptualized from the observation that some genomic variation is frequently observed to be associated with each other in the human population, i.e., they are in linkage disequilibrium (LD), and that these highly correlated variants can be

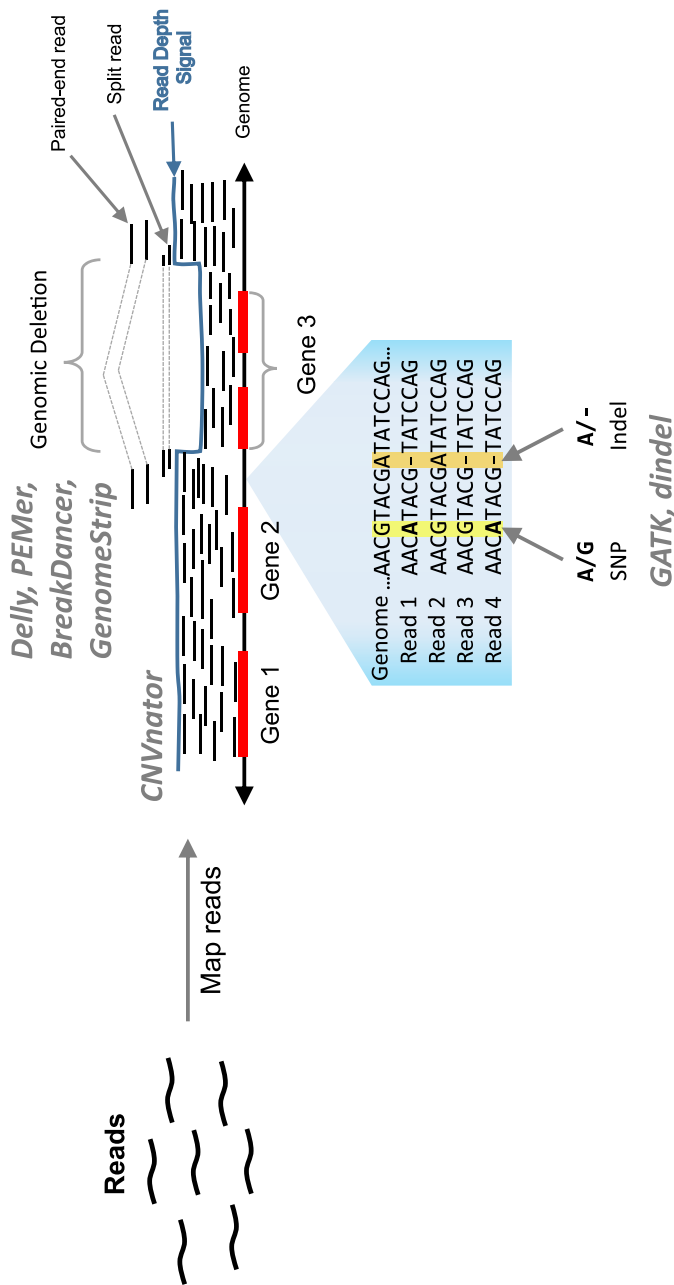


Fig. 1 Overview of a bioinformatic analysis of DNA sequencing reads in variant detection. First, the sequenced reads (wavy lines) are mapped to the genome (thick black line with arrowheads at termini) using read mapping tools. After mapping, the mapped reads (straight lines) at each location is counted to generate the read depth signal (blue line above the reads). A genomic deletion can be estimated when a drop in the read depth signal can be observed. Reads can be single-end (straight lines), or paired-end (reads connected by dashed lines with corners). When mapped to genes, reads can 'split' (reads connected by straight dashed lines), due to exon-intron junctions (exons are denoted by thick red lines on genome). Here, we focus on a specific genomic region (blue inset) to illustrate an example of read alignment and detection of an SNV (shaded yellow) and indel (shaded orange). For the SNV, a nucleotide change from 'G' to 'A' is supported by 2 reads, while an 'A' nucleotide deletion is supported by 3 reads. Some of the bioinformatics tools that use paired-end reads, split reads, and read depth are also shown (in grey below the blue inset).

inherited in haplotype blocks (Wall and Pritchard, 2003). The accuracy of imputation is contingent on both knowing the ‘phase’ (i.e., haplotypes) of the genetic markers, and the availability of accurate reference panels of LD patterns. Hence, there are two essential parts to *in silico* imputation: (1) the reference panel, and (2) the computational algorithm to estimate (or impute) the missing genotypes probabilistically, based on the haplotype information and LD patterns in the reference panel (Fig. 2).

Existing imputation reference panels are available, and already computationally constructed from large genotyping and sequencing studies (Fig. 2). The initial development of reference panels was first driven by the HapMap project for common variants (International HapMap Consortium *et al.*, 2007; The International HapMap 3 Consortium, 2010). Thus, imputation was skewed towards better estimation of genotypes from common variants. Further improvements in more recent years, particularly for the imputation of rare variants, came with the introduction of NGS and the 1000 Genomes Project (1000 Genomes Project Consortium *et al.*, 2012, 2015). Because imputation derives their genotype estimation from population allele frequencies, reference panels are typically built from larger consortia efforts with denser marker distribution and bigger and more diverse sample populations. They are constructed so that the allele frequencies of the genetic markers and LD patterns are better represented in accordance with ethnicities (Marchini and Howie, 2010). Reference panels can also be constructed by merging one’s own dataset with a reference panel, especially if the sample of interest is of an admixed origin or ethnic populations not found in the existing panels. Previous studies have already employed these existing and merged reference panels to successfully obtain rare variant imputation (Li *et al.*, 2011; Chen *et al.*, 2013). There are many bioinformatics tools and methodologies now that adopt several strategies with merging reference panels. For example, there has been early work where the merged panels are modeled jointly (Ewing *et al.*, 2012), or by first treating the panels as references for each other, and then imputing one’s data against the resultant panel (Howie *et al.*, 2009). There have also been strategies that used an ethnically diverse imputation panel, instead of a population-specific one, especially in situations where the dataset of interest is from an admixed population, or if the population of interest has a higher number of rare variants (Howie *et al.*, 2011).

There are many imputation tools currently available for use by the scientific community. They typically involve the following steps: phasing, which estimates the phase in the dataset of interest *in silico*, and then execute the genotype estimation process. There are many software for phasing, including MaCH (Li *et al.*, 2010), and SHAPEIT2 (Delaneau *et al.*, 2013). After phasing, the actual genotype imputation can be performed by several software, such as Minimac2 (Fuchsberger *et al.*, 2015), BEAGLE (Browning and Browning, 2011), and IMPUTE2 (Howie *et al.*, 2011). The performance and accuracy of these tools have been benchmarked and compared extensively in many publications over the years (Marchini and Howie, 2008; Roshyara *et al.*, 2016). Currently, there are also tools that use the biology of a specific genomic region to implement imputation, such as those that imputes genotypes in the complicated region of the major histocompatibility complex (MHC) (Karnes *et al.*, 2017).

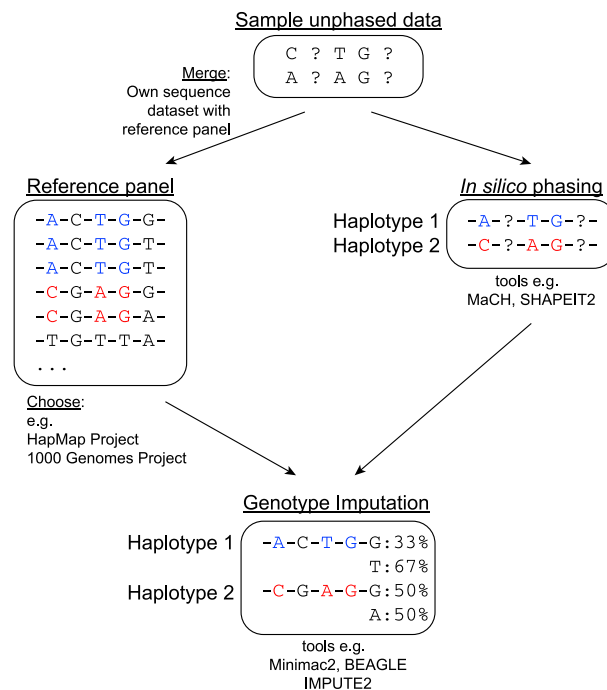


Fig. 2 General process of genotype imputation of rare variants. The process of imputation typically starts with a sample dataset that is usually unphased and contain missing genotypes (denoted by ‘?’). If the starting dataset is genotyped from a microarray, one can additionally sequence part of the dataset to merge with a selected reference panel, or choose from an existing reference panel, e.g., 1000 Genomes Project or HapMap Project. Concurrently, *in silico* phasing can be performed to estimate the haplotypes for each individual in the sample dataset using tools such as MaCH and SHAPEIT2. Finally, imputation software such as Minimac2, BEAGLE and IMPUTE2 can perform imputation to approximate the missing genotypes probabilistically, by comparing the user haplotypes to the reference haplotypes.

Detection of Somatic Variants in Cancer Genomes

So far, we have discussed mostly about the detection of rare germline variants. There is an additional set of non-germline mutations, i.e., somatic variants, which are heavily studied and researched particularly in cancer genomes. Cancer genomes typically harbor many types of somatic variants that run the gamut. Whole genome or exome sequencing are used to detect SNVs, CNVs, indels and SVs, whereas RNA-sequencing is ideal for detecting fusion transcript events.

Somatic variants can be detected in cancer, by comparing the genomic alterations in tumor samples with their matched normal samples. Somatic variant calling is a very challenging task, because of the complex nature of cancers. Tumor heterogeneity, normal cell contamination, sequencing and alignment errors might all limit the ability to detect somatic variants, especially since somatic variants can be present at very low allelic levels. Therefore, sensitivity in somatic variant calling is very important in cancer. There are various popular bioinformatics methods for calling somatic SNVs and indels in cancer. Both MuTect (Cibulskis *et al.*, 2013) and Strelka (Saunders *et al.*, 2012) use a Bayesian approach to detect somatic variants. MuTect is especially designed to detect variants with a very low allele fraction. SomaticSniper (Larson *et al.*, 2012) uses a Bayesian framework to combine the genotype likelihood of tumor samples and normal samples. VarScan2 (Koboldt *et al.*, 2009) has a heuristic approach based on thresholds for read depth, base quality, variant allele frequency and statistical significance. There are also different computational methods for identifying CNVs in cancer genomes. Some of them consider other factors in their detection algorithms, such as normal cell contamination, ploidy and clonality of the tumor cells. Some of the most popular CNV calling methods include ExomeCNV (Sathirapongsasuti *et al.*, 2011), CoNIFER (Krumm *et al.*, 2012), VarScan2 (Koboldt *et al.*, 2009) and CONTRA (Li *et al.*, 2012). For SV detection in cancer, BreakDancer (Chen *et al.*, 2009) and Delly (Rausch *et al.*, 2012), are widely used.

Another important computational challenge in cancer is distinguishing driver variants from passenger variants. Driver mutations provide growth advantage to a tumor cell, thus promoting cancer development. MUSIC (Dees *et al.*, 2012), MutSig (Lawrence *et al.*, 2013), CHASM (Carter *et al.*, 2009), SIFT (Ng and Henikoff, 2003), PolyPhen (Adzhubei *et al.*, 2013) and other tools aim to predict *in silico* whether a somatic mutation is a driver or passenger, and prioritize candidate variants for downstream experiments.

Annotation of Rare Variation

With large-scale variant detection from sequencing and array technologies, a looming challenge is the rapid and efficient functional annotation of these variants in a high throughput fashion. One of the goals of *in silico* functional annotation is to be able to rank and prioritize variants for downstream experimental validation. The functional impact of rare variants can be estimated computationally in several ways, including comparative sequence analysis, and guilt-by-association. A secondary objective of the annotation process is to assess possible clinical impact of the variants, particularly in the context of diseases. These are typically tested in rare variant association studies (RVAS).

Impact Estimation for Rare Variants

We can estimate the impact of rare variants bioinformatically in different genomic contexts, and approaches. Often, the functional consequences of variants can be assessed in the context of the functional elements that it resides, in a guilt-by-association fashion. This requires prior knowledge of functional genomic elements that are provided by bioinformatics databases such as RefSeq (O'Leary *et al.*, 2016), Ensembl (O'Leary *et al.*, 2016), GENCODE (Harrow *et al.*, 2012a,b), and ENCODE (Dunham *et al.*, 2012). For example, Annovar (Wang *et al.*, 2010) and Ensembl's Variant Effect Predictor (VEP) (McLaren *et al.*, 2016) map rare variants of interest onto basic genomic features. Given a list of rare variants with genomic locations on the human reference genome, VEP can assess the potential impact of the variants, depending on whether they reside in an exon, intron, splice site or untranslated region. If the rare variant is found in a coding region, VEP can output whether the variant will cause a synonymous or non-synonymous change in the protein sequence. In non-coding regions, variants can be associated with key elements such as enhancers, DNA-binding motifs, and promoters.

By comparing multiple sequences across species or within species, more quantitative tools were developed to examine the likelihood of a particular variant to be functionally impactful, based on the various biological properties and genomic contexts. For example, the GERP score measures the divergence or sequence conservation of every base in the human genome, based on sequence comparison across multiple primate genomes (Cooper *et al.*, 2005; Davydov *et al.*, 2010). SIFT uses the BLOSUM matrix, which is derived from comparing inter-species homologous sequences, to estimate the likelihood of non-synonymous variants being deleterious (Ng and Henikoff, 2001). PolyPhen2 (Adzhubei *et al.*, 2013) additionally uses information from the three-dimensional structure of a protein to further examine the impact of a coding variant. Other computational tools such as FATHMM (Shihab *et al.*, 2013), ENTPRISE (Zhou *et al.*, 2016), and MutationTaster (Schwarz *et al.*, 2014) are also more recent additions. Funseq2 (Fu *et al.*, 2014) provides a score for a non-coding variant based on the integration of multiple layers of information, such as the position of a variant in regulatory networks, and the disruptive effect of the variant on DNA-binding motifs. Combined Annotation Dependent Depletion (CADD) algorithm (Kircher *et al.*, 2014) also outputs a score based on a support vector machine classifier that integrates multiple annotations into its prediction. A non-exhaustive list of currently available bioinformatics tools for functional impact estimation is available in Table 1. There is also a recent strategy of aggregating and computing the enrichment of rare allele-specific variants across multiple individual genomes, in order to identify regions that exhibit allele-specific behavior (Chen *et al.*, 2016).

Table 1 Table of bioinformatics tools that can annotate and prioritize rare variants. This table provides a list of bioinformatics tools that can estimate the functional impact of variants in three broad categories, namely tools that can annotate variants in the coding and noncoding regions, and tools that gives a score for each nucleotide in the genome, allowing quantification and prioritization of variants

<i>Software tool</i>	<i>Year first introduced</i>	<i>PubMed ID and citation</i>
<i>Coding-variant-based</i>		
SIFT	2001	11337480 (Ng and Henikoff, 2001)
PolyPhen	2002	12202775 (Ramensky <i>et al.</i> , 2002)
MAPP	2005	15965030 (Stone and Sidow, 2005)
PMUT	2005	15879453 (Ferrer-Costa <i>et al.</i> , 2005)
PHD-SNP	2006	16895930 (Capriotti <i>et al.</i> , 2006)
SNAP	2007	17526529 (Bromberg and Rost, 2007)
MutPred	2009	19734154 (Li <i>et al.</i> , 2009)
SNPS&GO	2009	19514061 (Calabrese <i>et al.</i> , 2009)
MutationTaster	2010	19734154 (Schwarz <i>et al.</i> , 2010)
Condel	2011	21457909 (González-Pérez and López-Bigas, 2011)
MutationAssessor	2011	21727090 (Reva <i>et al.</i> , 2011)
CAROL	2012	22261837 (Lopes <i>et al.</i> , 2012)
FATHMM	2013	23033316 (Shihab <i>et al.</i> , 2013)
PredictSNP	2014	24453961 (Bendl <i>et al.</i> , 2014)
CADD	2014	24487276 (Kircher <i>et al.</i> , 2014)
<i>Noncoding-variant-based</i>		
FunSeq	2013	PMC3947637 (Khurana <i>et al.</i> , 2013)
GWAVA	2014	24487584 (Ritchie <i>et al.</i> , 2014)
FATHMM-XF	2017	28968714 (Rogers <i>et al.</i> , 2017)
<i>Nucleotide-based</i>		
GERP	2005	15965027 (Cooper <i>et al.</i> , 2005)
PhastCons	2005	16024819 (Siepel <i>et al.</i> , 2005)
PhyloP	2010	19858363 (Pollard <i>et al.</i> , 2010)

RVAS Studies: Predicting Functions of Rare Variants

The association studies are very useful for understanding the functional impact of variants in the context of traits and diseases. For example, GWASes give clues about the phenotypic effects of variants. In these studies, there is a sample with the disease or trait that is studied, termed the ‘case’ sample. There is also the ‘control’ sample, which includes individuals who do not have the manifestation of the disease or trait. The bioinformatics tools for GWAS look for genetic variants that are enriched in the case sample compared to the control sample. The enrichment of the variants indicate that the trait or disease associates with the variant (Bush and Moore, 2012). As mentioned, GWASes have identified thousands of loci that correlate with diverse set of traits and diseases, and can be found in the GWAS catalog (MacArthur *et al.*, 2017).

The GWASes generally address common variants that have small phenotypic effects (Fig. 3). These variants support the common variant common disease hypothesis (Pritchard and Cox, 2002), where many common variants with small effects culminate to a disease phenotype. However, based on heritability estimates in twin studies, the variants identified so far have been unable to account for all genetic contributions. Hence, for a while, researchers have thought that this ‘missing heritability’ can be explained at least in part by the effects of rare variants. This forms the basis of the rare-variant-common-disease hypothesis, which attributes rare variants to diseases (Cirulli and Goldstein, 2010).

In order to identify the effects of rare variants in traits and diseases, it is necessary to consider allelic heterogeneity, where different rare variants can affect a gene and give rise to the same phenotypic presentation. Moreover, there may be multiple causal genes. A bioinformatics method similar to GWAS can be deployed on rare variants to associate them with diseases and traits. These are termed generally as rare variant association studies (RVAS). In RVAS, it is necessary to use very large case and control sample sizes in order to obtain enough statistical power to detect the enrichment of rare variant alleles in the case sample (Lin, 2016). This may generally turn out to be impractical, especially if the prevalence of the rare allele is infinitesimally small. One approach is to group rare variants into biological elements, such as genes or signaling pathways. After grouping, a statistical test is performed to evaluate whether the element is under higher “burden” in the case sample compared to the control sample. This way, the association is computed for a region or element of multiple related rare variants, rather than a single variant. Because the frequency that an element is affected by the rare variants is increased, such burdening also increases the power of detection (Fig. 3). Most notably, RVAS studies have been useful in psychiatric diseases such as Alzheimer’s and Schizophrenia (Cruchaga *et al.*, 2014; Yang *et al.*, 2017; Singh *et al.*, 2017). SKAT (Wu *et al.*, 2011) is one of the most popular bioinformatics tools for performing various types of RVAS.

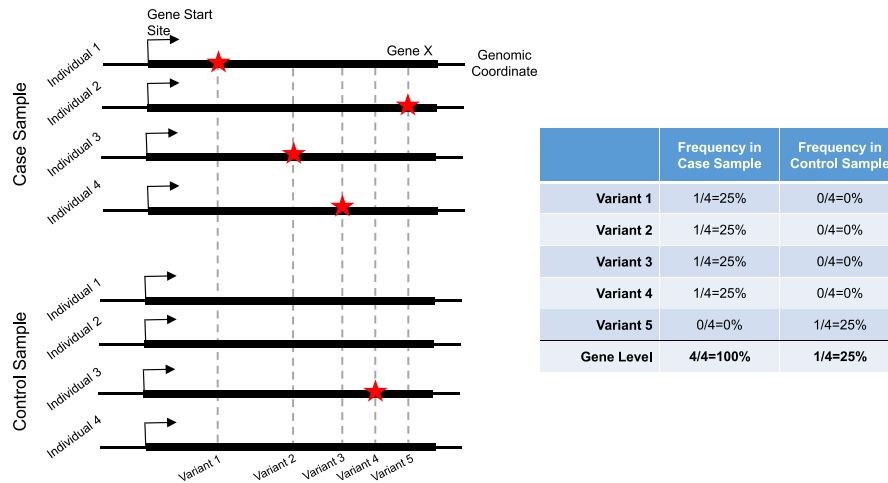


Fig. 3 Illustration of a burden test in a hypothetical RVAS study. This is a toy example showing the variants on a gene for eight individuals, four of whom are cases (with the trait or disease in question) and four are control samples. There are five mutations (red stars) in total that affect the gene X. The table summarizes the frequency of the variants in the case and control samples. Each mutation has a frequency of 25%, hence the individual mutations in the case sample do not show significant enrichment because each mutation is seen only once. However, when the mutations are counted on the gene as a whole, the case sample shows 4 times higher ‘burden’ compared to the control sample.

The accuracy of burden tests depends heavily on the precision and amount of element annotations. The annotations of protein coding genes on human genome can be assumed to be virtually complete (Harrow *et al.*, 2012a,b). It is, however, important to note that in the majority of GWAS and RVAS, variants are in non-coding regions of the genome (Zhang and Lupski, 2015). This implies that non-coding elements, like long non-coding RNAs and enhancers, must be studied in greater depth with respect to their functional roles. Moreover, the annotation of non-coding elements in the human genome is far from complete. Projects such as ENCODE (Bernstein *et al.*, 2012) and Roadmap Epigenome (Romanoski *et al.*, 2015) have been leading the way in the annotation of non-coding elements using functional genomics assays coupled with the corresponding bioinformatics pipelines, notably in ChIP-Seq (Harmanci *et al.*, 2014; Rozowsky *et al.*, 2009) and RNA-seq (Pepke *et al.*, 2009). Also, recent studies point to the importance of the three dimensional conformation of chromatin in genetic regulation (Belaghzal *et al.*, 2017). Taken altogether, understanding various non-coding elements can potentially enable a better delineation of the effects of rare variants when used in the RVAS studies.

One way to increase the power of RVAS studies is retrospectively combining results of previous RVAS studies. This bioinformatics analysis is referred to as meta-analysis of association studies (Begum *et al.*, 2012). Meta-analysis is very useful, because they make use of earlier association studies to increase the power of detection by increasing sample size. These methods combine the test statistics and p-values of individual studies to perform statistical test on the enrichment of rare variants in the pooled sample set. However, a major challenge is the computational adjustment for hidden covariates, such that the identified associations are based solely on genetic factors. Population stratification (Cardon and Palmer, 2003) and familial relationships (Manichaikul *et al.*, 2010) in the data are two major confounding factors that need to be adjusted for in association studies (Vilhjálmsdóttir and Nordborg, 2013).

Future Directions

As more research is being performed for rare variation, an important future direction is the development of novel and more effective statistical methods that are suitable for new sequencing technologies. This will be especially imperative for the technologies that generate long reads, which will enable genomics regions that were originally difficult to be sequenced by short reads to be made accessible. Using long-reads technologies, we will also be able to identify more accurately rare SVs, which will add more to our understanding of how these large variants affect diseases.

These developments will also bring new challenges. The large-scale sequencing efforts will generate a large number of variants, whose effects cannot be fully and quickly understood with the current knowledge and technologies. The new high throughput functional assays (Shalem *et al.*, 2015) have great potential in generating new data resources that can be combined with computational methods to better understand the effects of variants. These studies will also clarify how we should approach the classifications of rare and common variants with respect to their utility in public health and epidemiology (Visscher *et al.*, 2012). In addition to better uncovering their effects, sharing the data associated with variants of unknown significance (VUS) with patients will also create ethical challenges. We need to create standards and privacy-preserving frameworks for sharing these data. Understanding rare VUS will be best enabled using experimental protocols that follow up on

RVAS studies, by performing animal or cell culture experiments. These functional experiments are very important to functionally validate the identified GWAS and RVAS variants. Finally, the ethical considerations around sharing genomic data is still in its infancy in terms of implementation. Although there is a general consensus that genomic data may have adverse effects regarding the genomic privacy of the individual and his/her relatives, it is still an area of open debate as to how the balance between open data sharing incentives (Joly *et al.*, 2016) and privacy requirements can be maintained. In particular, the rare variants are one of the key aspects in genomic privacy, because they are private to a small group of individuals, so they can be sensitive enough for the identification of specific individuals (Homer *et al.*, 2008; Harmanci and Gerstein, 2016; Schadt *et al.*, 2012). New bioinformatic approaches will be necessary to share data related to rare variants in a privacy-preserving manner.

Closing Remarks

Our understanding of how rare variants impact human biology has come very far but is also still limited in many regards. In this article, we have reviewed some of the applications of bioinformatics in building computational approaches for detection and annotation of rare variants. We believe that increasing sample sizes in cohort studies, improving sequencing technologies, and new functional experiments will bring many exciting developments in the near future. Only through continued development and improvement of both experimental and computational methodologies, can we break new grounds.

See also: Natural Language Processing Approaches in Bioinformatics

References

- 1000 Genomes Project Consortium, *et al.*, 2015. A global reference for human genetic variation. *Nature* 526 (7571), 68–74. doi:10.1038/nature15393.
- Abecasis, G.R., *et al.*, 2012. An integrated map of genetic variation from 1,092 human genomes. *Nature* 491 (7422), 56–65. doi:10.1038/nature11632.
- Abyzov, A., *et al.*, 2011. CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Research* 21 (6), 974–984. doi:10.1101/gr.114876.110.
- Adzhubei, I., Jordan, D.M., Sunyaev, S.R., 2013. Predicting functional effect of human missense mutations using PolyPhen-2 (Chapter 7). In: Haines, J.L., *et al.*, *Current Protocols in Human Genetics*. doi:10.1002/0471142905.hg0720s76. p. Unit7.20
- Affymetrix (no date) *UK Biobank Axiom Array*. Available at: <http://www.ukbiobank.ac.uk/wp-content/uploads/2014/04/UK-Biobank-Axiom-Array-Datasheet-2014.pdf> (Accessed: 8 December 2017).
- Agarwala, V., *et al.*, 2013. Evaluating empirical bounds on complex disease genetic architecture. *Nature Genetics* 45 (12), 1418–1427. doi:10.1038/ng.2804.
- Albers, C.A., *et al.*, 2011. Dindel: Accurate indel calls from short-read data. *Genome Research* 21 (6), 961–973. doi:10.1101/gr.112326.110.
- Alkan, C., Coe, B.P., Eichler, E.E., 2011. Genome structural variation discovery and genotyping. *Nature Reviews Genetics*. 363–376. doi:10.1038/nrg2958.
- Bailey, J.A., Eichler, E.E., 2006. Primate segmental duplications: Crucibles of evolution, diversity and disease. *Nature Reviews Genetics*. 552–564. doi:10.1038/nrg1895.
- Banerjee, D., 2013. Array comparative genomic hybridization: An overview of protocols, applications, and technology trends. *Methods in Molecular Biology* (Clifton, N.J.) 973, 1–13. doi:10.1007/978-1-62703-281-0_1.
- Begum, F., *et al.*, 2012. Comprehensive literature review and statistical considerations for GWAS meta-analysis. *Nucleic Acids Research*. 3777–3784. doi:10.1093/nar/gkr1255.
- Belaghzal, H., Dekker, J., Gibcus, J.H., 2017. Hi-C 2.0: An optimized Hi-C procedure for high-resolution genome-wide mapping of chromosome conformation. *Methods* 123, 56–65. doi:10.1016/j.ymeth.2017.04.004.
- Belkadi, A., *et al.*, 2015. Whole-genome sequencing is more powerful than whole-exome sequencing for detecting exome variants. *Proceedings of the National Academy of Sciences of the United States of America* 112 (17), 5473–5478. doi:10.1073/pnas.1418631112.
- Bendl, J., *et al.*, 2014. PredictSNP: Robust and accurate consensus classifier for prediction of disease-related mutations. *PLoS Computational Biology* 10 (1), e1003440. doi:10.1371/journal.pcbi.1003440.
- Bernstein, B.E., *et al.*, 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74. doi:10.1038/nature11247.
- Bomba, L., Walter, K., Soranzo, N., 2017. The impact of rare and low-frequency genetic variants in common disease. *Genome Biology* 18 (1), 77. doi:10.1186/s13059-017-1212-4.
- Bromberg, Y., Rost, B., 2007. SNAP: Predict effect of non-synonymous polymorphisms on function. *Nucleic Acids Research* 35 (11), 3823–3835. doi:10.1093/nar/gkm238.
- Browning, B.L., Browning, S.R., 2011. A fast, powerful method for detecting identity by descent. *American Journal of Human Genetics* 88 (2), 173–182. doi:10.1016/j.ajhg.2011.01.010.
- Bush, W.S., Moore, J.H., 2012. Chapter 11: Genome-Wide Association Studies. *PLoS Computational Biology* 8 (12), doi:10.1371/journal.pcbi.1002822.
- Calabrese, R., *et al.*, 2009. Functional annotations improve the predictive score of human disease-related mutations in proteins. *Human Mutation* 30 (8), 1237–1244. doi:10.1002/humu.21047.
- Capriotti, E., Calabrese, R., Casadio, R., 2006. Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. *Bioinformatics (Oxford, England)* 22 (22), 2729–2734. doi:10.1093/bioinformatics/btl423.
- Cardon, L.R., Palmer, L.J., 2003. Population stratification and spurious allelic association. *Lancet*. 598–604. doi:10.1016/S0140-6736(03)12520-2.
- Carter, H., *et al.*, 2009. Cancer-specific high-throughput annotation of somatic mutations: Computational prediction of driver missense mutations. *Cancer Research* 69 (16), 6660–6667. doi:10.1158/0008-5472.CAN-09-1133.
- Carvalho, B., *et al.*, 2007. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. *Biostatistics (Oxford, England)* 8 (2), 485–499. doi:10.1093/biostatistics/kxl042.
- Castaldo, G., *et al.*, 1999. Detection of five rare cystic fibrosis mutations peculiar to Southern Italy: Implications in screening for the disease and phenotype characterization for patients with homozygote mutations. *Clinical Chemistry* 45 (7), 957–962. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10388469>
- Chen, J., *et al.*, 2016. A uniform survey of allele-specific binding and expression over 1000-Genomes-Project individuals. *Nature Communications* 7, 11101. doi:10.1038/ncomms11101.
- Chen, K., *et al.*, 2009. BreakDancer: An algorithm for high-resolution mapping of genomic structural variation. *Nature Methods* 6 (9), 677–681. doi:10.1038/nmeth.1363.

- Chen, Z., *et al.*, 2013. The G84E mutation of HOXB13 is associated with increased risk for prostate cancer: Results from the REDUCE trial. *Carcinogenesis* 34 (6), 1260–1264. doi:10.1093/carcin/bgt055.
- Chouraki, V., Seshadri, S., 2014. Genetics of Alzheimer's disease. *Advances in Genetics* 87, 245–294. doi:10.1016/B978-0-12-800149-3.00005-6.
- Cibulskis, K., *et al.*, 2013. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology* 31 (3), 213–219. doi:10.1038/nbt.2514.
- Cirulli, E.T., Goldstein, D.B., 2010. Uncovering the roles of rare variants in common disease through whole-genome sequencing. *Nature Reviews Genetics*. 415–425. doi:10.1038/nrg2779.
- Clark, M.J., *et al.*, 2011. Performance comparison of exome DNA sequencing technologies. *Nature Biotechnology* 29 (10), 908–916. doi:10.1038/nbt.1975.
- Collins, F.S., 2007. *The Cancer Genome Atlas (TCGA)*. Online. 1–17.
- Condit, C.M., *et al.*, 2002. The changing meanings of “mutation”: A contextualized study of public discourse. *Human Mutation* 19 (1), 69–75. doi:10.1002/humu.10023.
- Cooper, G.M., *et al.*, 2005. Distribution and intensity of constraint in mammalian genomic sequence. *Genome Research* 15 (7), 901–913. doi:10.1101/gr.3577405.
- Cortes, A., Brown, M.A., 2011. Promise and pitfalls of the ImmunoChip. *Arthritis Research & Therapy* 13 (1), 101. doi:10.1186/ar3204.
- Cruchaga, C., *et al.*, 2014. Rare coding variants in the phospholipase D3 gene confer risk for Alzheimer's disease. *Nature* 505 (7484), 550–554. doi:10.1038/nature12825.
- Davydov, E.V. *et al.*, 2010. Identifying a High Fraction of the Human Genome to be under Selective Constraint Using GERP++, *PLoS Computational Biology*. Edited by W. W. Wasserman, 6(12), e1001025. doi: 10.1371/journal.pcbi.1001025.
- Dees, N.D., *et al.*, 2012. MuSiC: Identifying mutational significance in cancer genomes. *Genome Research* 22 (8), 1589–1598. doi:10.1101/gr.134635.111.
- Delaneau, O., Zagury, J.-F., Marchini, J., 2013. Improved whole-chromosome phasing for disease and population genetic studies. *Nature Methods* 10 (1), 5–6. doi:10.1038/nmeth.2307.
- Dunham, I., *et al.*, 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature*. 57–74. doi:10.1038/nature11247.
- den Dunnen, J.T., *et al.*, 2016. HGVS Recommendations for the Description of Sequence Variants: 2016 Update. *Human Mutation* 37 (6), 564–569. doi:10.1002/humu.22981.
- English, A.C., Salerno, W.J., Reid, J.G., 2014. PBHoney: Identifying genomic variants via long-read discordance and interrupted mapping. *BMC Bioinformatics* 15 (1), 180. doi:10.1186/1471-2105-15-180.
- Ewing, C.M., *et al.*, 2012. Germline mutations in HOXB13 and prostate-cancer risk. *The New England Journal of Medicine* 366 (2), 141–149. doi:10.1056/NEJMoa1110000.
- Feng, Y., *et al.*, 2015. Nanopore-based fourth-generation DNA sequencing technology. *Genomics, Proteomics and Bioinformatics*. 4–16. doi:10.1016/j.gpb.2015.01.009.
- Ferrer-Costa, C., *et al.*, 2005. PMUT: A web-based tool for the annotation of pathological mutations on proteins. *Bioinformatics (Oxford, England)* 21 (14), 3176–3178. doi:10.1093/bioinformatics/bti486.
- Fu, Y., *et al.*, 2014. FunSeq2: A framework for prioritizing noncoding regulatory variants in cancer. doi: 10.1186/s13059-014-0480-5.
- Fuchsberger, C., Abecasis, G.R., Hinds, D.A., 2015. minimac2: Faster genotype imputation. *Bioinformatics (Oxford, England)* 31 (5), 782–784. doi:10.1093/bioinformatics/btu704.
- Gentleman, R., *et al.* (Eds.), 2005. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York, NY: Springer New York. (Statistics for Biology and Health). doi: 10.1007/0-387-29362-0.
- González-Pérez, A., López-Bigas, N., 2011. Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel. *American Journal of Human Genetics* 88 (4), 440–449. doi:10.1016/j.ajhg.2011.03.004.
- Gordon, D., *et al.*, 2016. Long-read sequence assembly of the gorilla genome. *Science* 352 (6281), aae0344. doi:10.1126/science.aae0344. aae0344.
- Griswold, A.J., *et al.*, 2015. Targeted massively parallel sequencing of autism spectrum disorder-associated genes in a case control cohort reveals rare loss-of-function risk variants. *Molecular Autism* 6, 43. doi:10.1186/s13229-015-0034-z.
- Guo, Y., *et al.*, 2014. Illumina human exome genotyping array clustering and quality control. *Nature Protocols* 9 (11), 2643–2662. doi:10.1038/nprot.2014.174.
- Handsaker, R.E., *et al.*, 2011. Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nature Genetics* 43 (3), 269–276. doi:10.1038/ng.768.
- Harmanci, A., Gerstein, M., 2016. Quantification of private information leakage from phenotype-genotype data: Linking attacks. *Nature Methods* 13 (3), 251–256. doi:10.1038/nmeth.3746.
- Harmanci, A., Rozowsky, J., Gerstein, M., 2014. MUSIC: Identification of enriched regions in ChIP-Seq experiments using a mappability-corrected multiscale signal processing framework. *Genome biology* 15 (10), 474. doi:10.1186/s13059-014-0474-3.
- Harrow, J., *et al.*, 2012a. GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research*. 1760–1774.
- Harrow, J., *et al.*, 2012b. GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research* 22 (9), 1760–1774. doi:10.1101/gr.135350.111.
- Hasan, M.S., Wu, X., Zhang, L., 2015. Performance evaluation of indel calling tools using real short-read data. *Human Genomics* 9 (1), 20. doi:10.1186/s40246-015-0042-2.
- Higgs, D.R., Wood, W.G., 2008. Genetic complexity in sickle cell disease. *Proceedings of the National Academy of Sciences of the United States of America* 105 (33), 11595–11596. doi:10.1073/pnas.0806633105.
- Homer, N., *et al.*, 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics* 4. doi:10.1371/journal.pgen.1000167.
- Howie, B., Marchini, J., Stephens, M., 2011. Genotype imputation with thousands of genomes. *G3 (Bethesda, Md.)* 1 (6), 457–470. doi:10.1534/g3.111.001198.
- Howie, B.N., Donnelly, P., Marchini, J., 2009. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genetics* 5 (6), e1000529. doi:10.1371/journal.pgen.1000529.
- Hu, J., He, X., 2007. Enhanced quantile normalization of microarray data to reduce loss of information in gene expression profiles. *Biometrics* 63 (1), 50–59. doi:10.1111/j.1541-0420.2006.00670.x.
- Huber, W., *et al.*, 2015. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods* 12 (2), 115–121. doi:10.1038/nmeth.3252.
- Illumina (no date) *Illumina Exome-2.4 BeadChip*. Available at: https://www.illumina.com/documents/products/datasheets/datasheet_humanexome_beadchips.pdf (Accessed: 8 December 2017).
- Irizarry, R.A., *et al.*, 2003. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics (Oxford, England)* 4 (2), 249–264. doi:10.1093/biostatistics/4.2.249.
- Joly, Y., *et al.*, 2016. Are Data Sharing and Privacy Protection Mutually Exclusive? *Cell*. 1150–1154. doi:10.1016/j.cell.2016.11.004.
- Karki, R., *et al.*, 2015. Defining “mutation” and “polymorphism” in the era of personal genomics. *BMC Medical Genomics* 8, 37. doi:10.1186/s12920-015-0115-z.
- Karnes, J.H., *et al.*, 2017. Comparison of HLA allelic imputation programs. *PLoS One* 12 (2), e0172444. doi:10.1371/journal.pone.0172444.
- Khurana, E., *et al.*, 2013. Integrative Annotation of Variants from 1092 Humans: Application to Cancer Genomics. *Science* 342 (6154), 1235587. doi:10.1126/science.1235587.
- Kircher, M., *et al.*, 2014. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics* 46 (3), 310–315. doi:10.1038/ng.2892.
- Koboldt, D.C., *et al.*, 2009. VarScan: Variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* 25 (17), 2283–2285. doi:10.1093/bioinformatics/btp373.
- de Koning, A.P.J., *et al.*, 2011. Repetitive elements may comprise over Two-Thirds of the human genome. *PLoS Genetics* 7 (12), doi:10.1371/journal.pgen.1002384.
- Korbel, J.O., *et al.*, 2009. PEMer: A computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology* 10 (2), R23. doi:10.1186/gb-2009-10-2-r23.
- Krumm, N., *et al.*, 2012. Copy number variation detection and genotyping from exome sequence data. *Genome Research* 22 (8), 1525–1532. doi:10.1101/gr.138115.112.
- LaFramboise, T., 2009. Single nucleotide polymorphism arrays: A decade of biological, computational and technological advances. *Nucleic Acids Research* 37 (13), 4181–4193. doi:10.1093/nar/gkp552.

- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nature Methods* 357–359. doi:10.1038/nmeth.1923.
- Larson, D.E., et al., 2012. Somaticsniper: Identification of somatic point mutations in whole genome sequencing data. *Bioinformatics* 28 (3), 311–317. doi:10.1093/bioinformatics/btr665.
- Lawrence, M.S., et al., 2013. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature* 499 (7457), 214–218. doi:10.1038/nature12213.
- Li, B., et al., 2009. Automated inference of molecular mechanisms of disease from amino acid substitutions. *Bioinformatics (Oxford, England)* 25 (21), 2744–2750. doi:10.1093/bioinformatics/btp528.
- Li, H., Ruan, J., Durbin, R., 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18 (11), 1851–1858. doi:10.1101/gr.078212.108.
- Li, J., et al., 2012. CONTRA: Copy number analysis for targeted resequencing. *Bioinformatics* 28 (10), 1307–1313. doi:10.1093/bioinformatics/bts146.
- Li, L., et al., 2011. Performance of genotype imputation for rare variants identified in exons and flanking regions of genes. *PLoS One* 6 (9), e24945. doi:10.1371/journal.pone.0024945.
- Li, M.M., et al., 2017. Standards and Guidelines for the Interpretation and Reporting of Sequence Variants in Cancer: A Joint Consensus Recommendation of the Association for Molecular Pathology, American Society of Clinical Oncology, and College of American Pathologists. *The Journal of Molecular Diagnostics : JMD* 19 (1), 4–23. doi:10.1016/j.jmoldx.2016.10.002.
- Li, Y., et al., 2010. MaCH: Using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic Epidemiology* 34 (8), 816–834. doi:10.1002/gepi.20533.
- Lin, W.Y., 2016. Beyond Rare-Variant Association Testing: Pinpointing Rare Causal Variants in Case-Control Sequencing Study. *Scientific Reports* 6. doi:10.1038/srep21824.
- Lin, Y., et al., 2016. Assembly of long error-prone reads using de Bruijn graphs. *Proceedings of the National Academy of Sciences* 113 (52), E8396–E8405. doi:10.1073/pnas.1604560113.
- Lopes, M.C., et al., 2012. A combined functional annotation score for non-synonymous variants. *Human Heredity* 73 (1), 47–51. doi:10.1159/000334984.
- MacArthur, J., et al., 2017. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). *Nucleic Acids Research* 45 (D1), D896–D901. doi:10.1093/nar/gkw1133.
- Manichaikul, A., et al., 2010. Robust relationship inference in genome-wide association studies. *Bioinformatics* 26 (22), 2867–2873. doi:10.1093/bioinformatics/btq559.
- Marchini, J., Howie, B., 2008. Comparing Algorithms for Genotype Imputation. *The American Journal of Human Genetics* 83 (4), 535–539. doi:10.1016/j.ajhg.2008.09.007.
- Marchini, J., Howie, B., 2010. Genotype imputation for genome-wide association studies. *Nature Reviews. Genetics* 11 (7), 499–511. doi:10.1038/nrg2796.
- McCoy, R.C., et al., 2014. Illumina TruSeq synthetic long-reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. *PLoS One* 9 (9), e106689. doi:10.1371/journal.pone.0106689.
- McKenna, A., et al., 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* 20 (9), 1297–1303. doi:10.1101/gr.107524.110.
- McLaren, W., et al., 2016. The Ensembl Variant Effect Predictor. *Genome Biology* 17 (1), 122. doi:10.1186/s13059-016-0974-4.
- Miclaus, K., et al., 2010. Batch effects in the BRLMM genotype calling algorithm influence GWAS results for the Affymetrix 500K array. *The Pharmacogenomics Journal* 10 (4), 336–346. doi:10.1038/tpj.2010.36.
- Morgan, A.P., 2016. argyle: An R Package for Analysis of Illumina Genotyping Arrays. *G3: GenesGenomesGenetics* 6 (2), 281–286. doi:10.1534/g3.115.023739.
- Mostovoy, Y., et al., 2016. A hybrid approach for de novo human genome sequence assembly and phasing. *Nature Methods* 13 (7), 587–590. doi:10.1038/nmeth.3865.
- Nelson, W.C., et al., 2015. An integrated genotyping approach for HLA and other complex genetic systems. *Human Immunology* 76 (12), 928–938. doi:10.1016/j.humimm.2015.05.001.
- Ng, P.C., Henikoff, S., 2001. Predicting deleterious amino acid substitutions. *Genome Research* 11 (5), 863–874. doi:10.1101/gr.176601.
- Ng, P.C., Henikoff, S., 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Research* 31 (13), 3812–3814. doi:10.1093/nar/gkg509.
- O'Leary, N.A., et al., 2016. Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research* 44 (D1), D733–D745. doi:10.1093/nar/gkv1189.
- Owzar, K., et al., 2008. Statistical challenges in preprocessing in microarray experiments in cancer. *Clinical Cancer Research : An Official Journal of the American Association for Cancer Research* 14 (19), 5959–5966. doi:10.1158/1078-0432.CCR-07-4532.
- Pepke, S., Wold, B., Mortazavi, A., 2009. Computation for ChIP-seq and RNA-seq studies. *Nature Methods* 6, S22–S32. doi:10.1038/nmeth.1371.
- Pollard, K.S., et al., 2010. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Research* 20 (1), 110–121. doi:10.1101/gr.097857.109.
- Pritchard, J.K., Cox, N.J., 2002. The allelic architecture of human disease genes: Common disease-common variant...or not? *Human Molecular Genetics* 11 (20), 2417–2423. doi:10.1093/hmg/11.20.2417.
- Ramensky, V., Bork, P., Sunyaev, S., 2002. Human non-synonymous SNPs: Server and survey. *Nucleic Acids Research* 30 (17), 3894–3900. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12202775>.
- Rausch, T., et al., 2012. DELLY: Structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* 28 (18), doi:10.1093/bioinformatics/bts378.
- Reva, B., Antipin, Y., Sander, C., 2011. Predicting the functional impact of protein mutations: Application to cancer genomics. *Nucleic Acids Research* 39 (17), e118. doi:10.1093/nar/gkr407.
- Rhoads, A., Au, K.F., 2015. PacBio Sequencing and Its Applications. *Genomics, Proteomics and Bioinformatics*. 278–289. doi:10.1016/j.gpb.2015.08.002.
- Richards, S., et al., 2015. Standards and guidelines for the interpretation of sequence variants: A joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genetics in Medicine : Official journal of the American College of Medical Genetics* 17 (5), 405–424. doi:10.1038/gim.2015.30.
- Ritchie, G.R.S., et al., 2014. Functional annotation of noncoding sequence variants. *Nature Methods* 11 (3), 294–296. doi:10.1038/nmeth.2832.
- Rogers, M.F., et al., 2017. FATHMM-XF: Accurate prediction of pathogenic point mutations via extended features. *Bioinformatics (Oxford, England)*. doi:10.1093/bioinformatics/btx536.
- Romanoski, C.E., et al., 2015. Epigenomics: Roadmap for regulation. *Nature* 518 (7539), 314–316. doi:10.1038/518314a.
- Roshyara, N.R., et al., 2016. Comparing performance of modern genotype imputation methods in different ethnicities. *Scientific Reports* 6 (1), 34386. doi:10.1038/srep34386.
- Rozowsky, J., et al., 2009. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature Biotechnology* 27, 66–75. doi:10.1038/nbt.1518.
- Sathirapongsasuti, J.F., et al., 2011. Exome sequencing-based copy-number variation and loss of heterozygosity detection: ExomeCNV. *Bioinformatics* 27 (19), 2648–2654. doi:10.1093/bioinformatics/btr462.
- Saunders, C.T., et al., 2012. Strelka: Accurate somatic small-variant calling from sequenced tumor-normal sample pairs. *Bioinformatics* 28 (14), 1811–1817. doi:10.1093/bioinformatics/bts271.
- Schadt, E.E., Woo, S., Hao, K., 2012. Bayesian method to predict individual SNP genotypes from gene expression data. *Nature Genetics*. 603–608. doi:10.1038/ng.2248.
- Schwarz, J.M., et al., 2010. MutationTaster evaluates disease-causing potential of sequence alterations. *Nature Methods* 7 (8), 575–576. doi:10.1038/nmeth0810-575.
- Schwarz, J.M., et al., 2014. MutationTaster2: Mutation prediction for the deep-sequencing age. *Nature Methods* 11 (4), 361–362. doi:10.1038/nmeth.2890.
- Shalem, O., Sanjana, N.E., Zhang, F., 2015. High-throughput functional genomics using CRISPR-Cas9. *Nature Reviews Genetics*. 299–311. doi:10.1038/nrg3899.
- Shendure, J., Ji, H., 2008. Next-generation DNA sequencing. *Nature Biotechnology* 26 (10), 1135–1145. doi:10.1038/nbt1486.
- Shihab, H.A., et al., 2013. Predicting the Functional, Molecular, and Phenotypic Consequences of Amino Acid Substitutions using Hidden Markov Models. *Human Mutation* 34 (1), 57–65. doi:10.1002/humu.22225.
- Siepel, A., et al., 2005. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research* 15 (8), 1034–1050. doi:10.1101/gr.3715005.

- Singh, T., *et al.*, 2017. The contribution of rare variants to risk of schizophrenia in individuals with and without intellectual disability. *Nature Genetics* 49 (8), 1167–1173. doi:10.1038/ng.3903.
- Soe, K., Gregoire-Bottex, M.M., 2017. A rare CFTR mutation associated with severe disease progression in a 10-year-old Hispanic patient. *Clinical Case Reports* 5 (2), 139–144. doi:10.1002/ccr3.764.
- Staaf, J., *et al.*, 2008. Normalization of Illumina Infinium whole-genome SNP data improves copy number estimates and allelic intensity ratios. *BMC Bioinformatics* 9, 409. doi:10.1186/1471-2105-9-409.
- Steinthorsdottir, V., *et al.*, 2014. Identification of low-frequency and rare sequence variants associated with elevated or reduced risk of type 2 diabetes. *Nature Genetics* 46 (3), 294–298. doi:10.1038/ng.2882.
- Stone, E.A., Sidow, A., 2005. Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity. *Genome Research* 15 (7), 978–986. doi:10.1101/gr.3804205.
- Sudmant, P.H., *et al.*, 2015. An integrated map of structural variation in 2,504 human genomes. *Nature* 526 (7571), 75–81. doi:10.1038/nature15394.
- Tattini, L., D'Aurizio, R., Magi, A., 2015. Detection of Genomic Structural Variants from Next-Generation Sequencing Data. *Frontiers in Bioengineering and Biotechnology*. 3. doi:10.3389/fbioe.2015.00092.
- Teo, Y.Y., 2012. Genotype calling for the Illumina platform. *Methods in Molecular Biology* (Clifton, N.J.) 850, 525–538. doi:10.1007/978-1-61779-555-8_29.
- The 1000 Genomes Project Consortium, 2012. An integrated map of genetic variation. *Nature* 135, 0–9. doi:10.1038/nature11632.
- The International HapMap 3 Consortium, 2010. Integrating common and rare genetic variation in diverse human populations. *Nature* 467 (7311), 52–58. doi:10.1038/nature09298.
- Vilhjálmsdóttir, B.J., Nordborg, M., 2013. The nature of confounding in genome-wide association studies. *Nature Reviews Genetics*. 1–2. doi:10.1038/nrg3382.
- Visscher, P.M., *et al.*, 2012. Five years of GWAS discovery. *American Journal of Human Genetics* 90 (1), 7–24. doi:10.1016/j.ajhg.2011.11.029.
- Wall, J.D., Pritchard, J.K., 2003. Haplotype blocks and linkage disequilibrium in the human genome. *Nature Reviews Genetics* 4 (8), 587–597. doi:10.1038/nrg1123.
- Wang, K., *et al.*, 2007. PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research* 17 (11), 1665–1674. doi:10.1101/gr.6861907.
- Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38 (16), e164. doi:10.1093/nar/gkq603.
- Wang, Q., *et al.*, 2017. Novel metrics to measure coverage in whole exome sequencing datasets reveal local and global non-uniformity. *Scientific Reports* 7 (1), doi:10.1038/s41598-017-01005-x.
- Wellcome Trust Case Control Consortium, 2007. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 447 (7145), 661–678. doi:10.1038/nature05911.
- Wineinger, N.E., Pajewski, N.M., Tiwar, H.K., 2011. A method to assess linkage disequilibrium between CNVs and SNPs inside copy number variable regions. *Frontiers in Genetics* 2 (APR), doi:10.3389/fgene.2011.00017.
- Wu, M.C., *et al.*, 2011. Rare variant association testing for sequencing data using the Sequence Kernel Association Test (SKAT). *American Journal of Human Genetics* 89, 82–93.
- Wu, J., *et al.*, 2012. Copy Number Variation detection from 1000 Genomes project exon capture sequencing data. *BMC Bioinformatics* 13 (1), 305. doi:10.1186/1471-2105-13-305.
- Yang, Z., *et al.*, 2017. Rare damaging variants in DNA repair and cell cycle pathways are associated with hippocampal and cognitive dysfunction: A combined genetic imaging study in first-episode treatment-naïve patients with schizophrenia. *Translational Psychiatry* 7 (2), e1028. doi:10.1038/tp.2016.291.
- Zhang, F., Lupski, J.R., 2015. Non-coding genetic variants in human disease. *Human Molecular Genetics*. R102–R110. doi:10.1093/hmg/ddv259.
- Zheng, G.X.Y., *et al.*, 2016. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nature Biotechnology* 34 (3), 303–311. doi:10.1038/nbt.3432.
- Zhou, H., Gao, M., Skolnick, J., 2016. ENTPRISE: An Algorithm for Predicting Human Disease-Associated Amino Acid Substitutions from Sequence Entropy and Predicted Protein Structures. *PLoS One* 11 (3), e0150965. doi:10.1371/journal.pone.0150965.
- Zuk, O., *et al.*, 2014. Searching for missing heritability: Designing rare variant association studies. *Proceedings of the National Academy of Sciences of the United States of America* 111 (4), E455–E464. doi:10.1073/pnas.1322563111.

Predicting Non-Synonymous Single Nucleotide Variants Pathogenic Effects in Human Diseases

Marwa S Hassan, National Research Centre, Cairo, Egypt

AA Shaalan, Zagazig University, Zagazig, Egypt

MI Dessouky, Menoufia University, Menouf, Egypt

Abdelaziz E Abdelnaiem, Zagazig University, Zagazig, Egypt

Dalia A Abdel-Haleem, National Research Centre, Cairo, Egypt

Mahmoud ElHefnawi, Nile University, Giza, Egypt and National Research Centre, Cairo, Egypt.

© 2019 Elsevier Inc. All rights reserved.

Introduction

The recent advances in bioinformatics technologies produce a vast amount of genetic data. Since gene or protein variants are random and most variations have no harmful consequences, recognition of genetic deviations is usually an essential (Zhou *et al.*, 2016). Variations in the protein have influence not only in the protein structure but also its stability and function. nsSNVs and mutations can affect protein function in an abundant way. Besides, early detection of nsSNVs and mutations can be helpful in diagnosing the illness at a first stage, prognosis, prevention, and treatment of illness (Kulshreshtha *et al.*, 2016).

Bioinformatics analyses are necessary to predict contributing Amino Acid Substitutions (AAS) to human diseases for each genome (Kulshreshtha *et al.*, 2016). Expecting the functional effect of AAS caused by nsSNVs is becoming increasingly significant as more and more novel variants are being discovered to discriminate disease-associated mutations with non-disease mutations (Kulshreshtha *et al.*, 2016). One of the main challenges in the human genome is – to recognize functional effects of Single-nucleotide Variants (SNVs). Although some of the Variants found in genes assessed in the laboratory, many others have not evaluated for their possible damaging effects on protein structure and function (Hepp *et al.*, 2015).

Several existing methods in this field established over the years for expecting the impact of SNVs on human health (Kulshreshtha *et al.*, 2016). Nevertheless, they still have a high false prediction rate, which is overcome by computational approaches depend on combined features (Capriotti and Fariselli, 2017). Recently, these computational tools that are available on the website as web servers work on algorithms (a set of rules) can satisfy better predictions of the effect of SNVs if used combined from rather than alone (Kulshreshtha *et al.*, 2016).

In this study, we present web servers that can use in predicting SNVs or mutated protein stability that causes diseases. Computational web servers based on machine learning techniques (MLTs) Support Vector Machine (SVM), Random Forest (RF), Hidden Markov Models (HMM) and Artificial Neural Network (ANN) and so on. The servers can categorize according to variant information into protein structure, evolutionary conservation, sequence parameters and Combined features (Kulshreshtha *et al.*, 2016; Dong *et al.*, 2015). Consensus tools (integration of various instruments) with specific to protein features may do better than others with unfocused component scores (Dong *et al.*, 2015). By detecting diseases causing mutations, servers help to detect and understand genotype-phenotype associations, predict disease, and choose suitable treatments for patients.

Computational Approaches

A significant challenge is how to assess the disease relevance of genomic variations on a large scale. Single Nucleotide Variants (SNVs) are the most plentiful type of mutations in the human genome (Wang and Wei, 2016). The ability of computational tools to produce accurate and meaningful predictions at the protein level is not yet sufficient (Gallion *et al.*, 2017). In the last years, many methods (Web servers) have been established to predict the effect of nsSNVs which considered as unsolved problems earlier. Among the most popular servers, both SIFT and PolyPhen were developed and updated in several studies (Ng, 2003; Xi *et al.*, 2004; Li *et al.*, 2009; Adzhubei *et al.*, 2010; 2013; Sim *et al.*, 2012; Gallion *et al.*, 2017). Computational tools based on the MLT (SVM, RF, ANN, HMM and so on). It divided into several Techniques related to variants information, sequence homology, protein structure, combined features (structure and sequence) and consensus tool (scores of individual devices) as indicated in (Fig. 1).

Sequence-Based Approaches

Nowadays, a huge amount of data generated from various genome sequencing projects which are utilized for making libraries or databases and further, applied for comparison of the query sequence to the target sequence (Kulshreshtha *et al.*, 2016). *PhD-SNPg* was developed and compared with CADD and FATHMM by Capriotti and Fariselli (2017). It is a binary approach built on a Gradient Boosting algorithm. It can expect the effect of single and multiple SNVs from an input file included four elements, which indicate the chromosome, the position, reference and alternative alleles. Its output is a probability score between 0 and 1. When the score is > 0.5, the SNVs predicted as Pathogenic otherwise Benign. *PhD-SNPg* web server is a user-friendly interface to expect the influence of SNVs in coding and non-coding regions (Capriotti and Fariselli, 2017).

A web-based tool *Cancer-Specific high-throughput Annotation of Somatic Mutations (CHASM)* is a computational tool to recognize and prioritize those missense mutations best likely to produce functional changes that enhance tumor cell creation. The CHASM

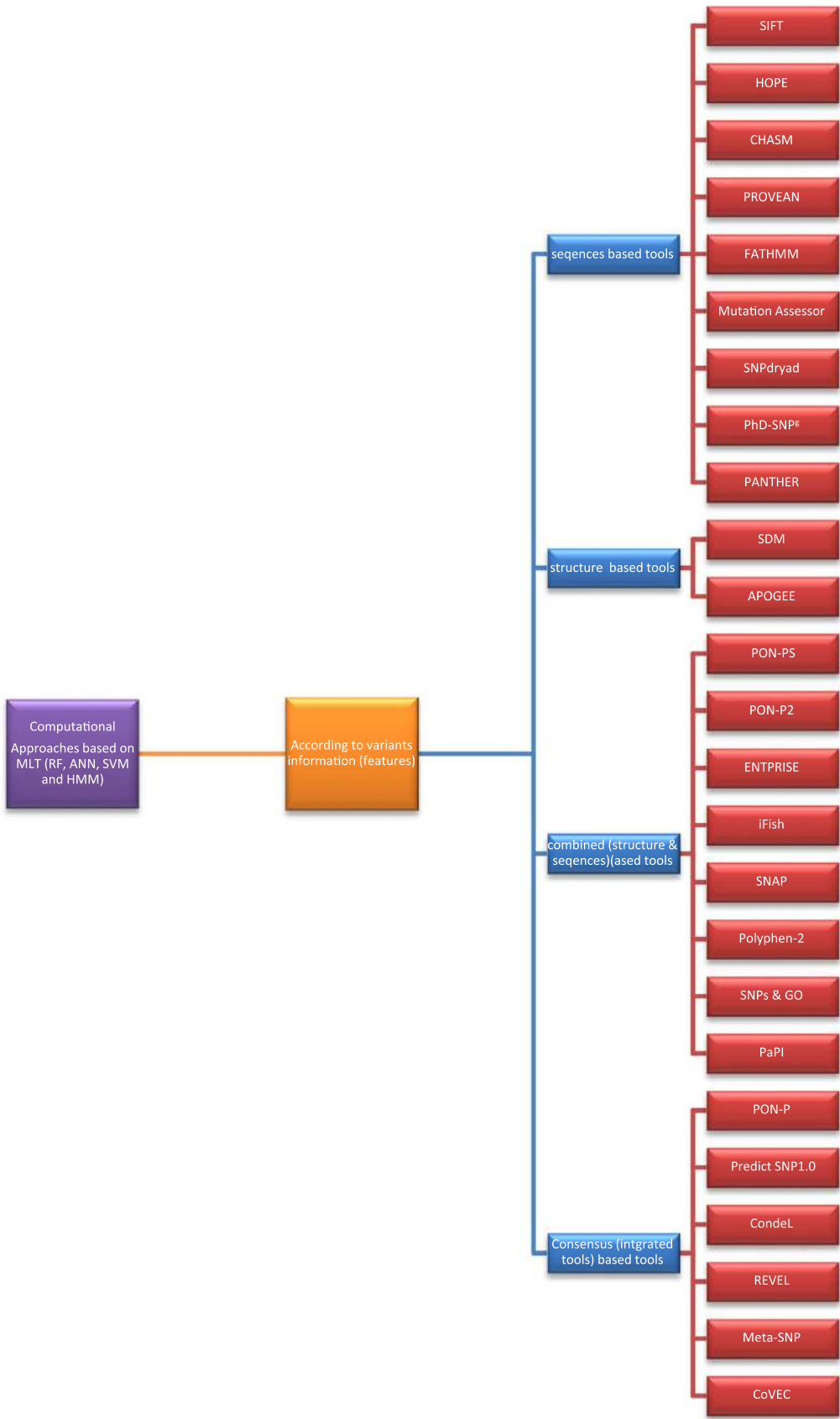


Fig. 1 Overall framework of the categories of computational approaches.

method built on a Random Forest classifier (RF). The method has high sensitivity and specificity when discriminating between known driver missense mutations and randomly created missense mutations (area under ROC curve > 0.91). It outperforms other expectation methods in discriminating known oncogenic mutations in TP53 (Carter *et al.*, 2009).

The most straight forward tool is *Homotopy optimization method (HOPE)* where any user of the online web server can submit a sequence and mutation. It can also collect information about the protein from various sources (3D protein structure, UniProt Database, and DAS servers). It is used for the prediction of standard phenotypic effects caused by single amino acid change. It discriminates between neutral and deleterious mutations. Further, it builds a model of mutated protein if the amount of identity between the query sequence and Protein Data Bank (PDB) file exceeds the threshold value (Kulshreshtha *et al.*, 2016).

Another sequence-based tool, *The Sorting Intolerant from Tolerant (SIFT)* uses a sequence homology based on the Numerous Sequence Alignment (MSA) conservation approach to classifying the nsSNVs as tolerated or damaged to the protein. The SIFT score is the normalized possibility that the amino acid change is tolerated. The score ranges from 0 to 1 with a cutoff score of 0.05. Amino acid substitutions with (cutoff < 0.05) are predicted to be deleterious, and those with (cutoff ≥ 0.05) are expected to be tolerated (Hepp *et al.*, 2015).

Protein Variation Effect Analyzer (PROVEAN) is another sequence based tool for the expectation of the damaging effect of variations in protein sequences. The prediction is based on the modification, caused by a nsSNP, in the similarity of the sequence of related protein sequences in an MSA. PROVEAN uses a delta alignment score built on the reference and variant types of the protein sequence concerning the alignment of homologous sequences. A score equal or below the threshold of -2.5 determines the classification as a deleterious nsSNP (Hepp *et al.*, 2015).

Another tool, *Protein Analysis Through Evolutionary Relationships (PANTHER)* evaluates the likelihood that a particular nsSNP will cause a functional alteration of the protein. It calculates the replacement Position-Specific Evolutionary Conservation (sub-PSEC) score built on a Hidden Markov Model (HMM) alignment of evolutionarily related proteins. Substitution with positive values (subPSEC > 0) is shown as functionally neutral; whereas negative values of (subPSEC < 0) expect deleterious substitutions. A subPSEC score (cutoff of -3) parallels to a 50% probability that a nsSNP is deleterious to the protein, with a possibility of causing a deleterious impact on the protein function (P deleterious) of 0.5 (Mi *et al.*, 2016).

Also, *Functional Analysis through Hidden Markov Models (FATHMM)* tool, software for the forecast of the functional impacts of protein missense variations, was described by Shihab *et al.* (2012). FATHMM Software and server achieved performance accuracies that outperformed traditional forecast strategies (SIFT, PolyPhen, and PANTHER) on two distinct benchmarks. Furthermore, in one benchmark, they achieved performance accuracies that outperform current state-of-the-art forecast strategies (SNPs & GO and MutPred). FATHMM can be professionally applied to large-scale human and nonhuman genome sequencing projects with the other benefit of phenotypic outcome associations. The VariBench database is available at see "Relevant Websites section" and the SwissVar at see "Relevant Websites section" (Shihab *et al.*, 2013).

The Mutation Assessor tool also calculates stability score that predicts the effect of the mutation on protein. It relies on the evolutionary conservation approach and uses MSA to generate a conservation score, which classifies the mutation into three categories; neutral, low, high or medium. All these mutations affect the functionality of protein and their stability. This tool also used for SNPs associated with diseases such as cancer. It has its database for the prediction of diseases; however, it can retrieve data from other databases such as COSMIC and UniProt (Kulshreshtha *et al.*, 2016).

SNPdryad is a novel computational technique that can forecast the pernicious influence of amino acid replacements happened in human proteins. SNP dryad consistently outperforms other leading algorithms in accurately predicting deleterious nsSNVs. It uses different conservation scoring and uses RF as a classifier. The SNP dryad's input is a non-synonymous human SNP and the sequence of the human protein; the production is the predicted deleterious score for the input nsSNP. With high score value, predicted nsSNP be more deleterious. SNP dryad on the complete human proteome produces a forecast scores for all the possible amino acid substitutions (Wong and Zhang, 2014).

Structure-Based Approaches

These tools predict the stability changes by detecting structural properties as secondary structure. *Site-Directed Mutate (SDM)* web server was generated by Worth *et al.* (2011). The SDM server provides users with a fast and perfect evaluating the impact that a mutation will have on protein structure and stability. It gives a 3D view of the wild and mutant- sort. It is a worthy tool for recognizing possible illness associations and used for forecasting deleterious nsSNVs at the genome scale. To run SDM, clients must upload a wild-type structure and the position and amino acid type of the mutation. The returned results contain information about the local structural environment of the wild and mutant-sort, a stability score and illness associated forecast (Worth *et al.*, 2011).

The pathogenicity Prediction through Logistic Model tree (APOGEE) server was developed in 2017 by Castellana *et al.* APOGEE is a Machine Learning technique that outperforms all current prediction techniques in assessing the harmfulness of non-Synonymous genome variations. It is based on the classification model of the Logistic Model Tree (LMT), which syndicates the logistic regression models with tree introduction resulting in a single tree (Castellana *et al.*, 2017). Structure-based tools cannot use if the protein cannot be crystallized; this limitation can resolve by using sequence-based prediction tools (Kulshreshtha *et al.*, 2016).

Combined Features Based Tools

Several preceding studies presented and assessed novel computational model that predict the impact of amino acid allelic variants on protein structure and function. Structural based approaches can utilize only when the structure is known. Otherwise, sequence-

based approaches suggested. However, predicted the accuracy of sequence-based approaches is lower than the structure-based approaches. None of them is confirmed to be accurate and provide complete mutant protein analysis. The prediction accuracy can increase by using the right combination of features. Therefore, combined and consensus approaches are developed with increased accuracy and efficiency and for use in all situations (Kulshreshtha *et al.*, 2016).

PON-PS is the first generic tool for classification of amino acid substitutions associated with benign, severe, and less severe phenotypes. It combines the PON-P2 pathogenicity predictor and an ML model that classifies variants into three categories showed an accuracy of 61% in the test dataset which is higher than for existing tolerance prediction methods. PON-PS also helps to detect and understand genotype-phenotype correlations (Niroula and Vihinen, 2017).

PON-P2 is a new computational technique for sorting of amino acid variations in human proteins. It is a Machine Learning based classifier that clusters the variants into pathogenic, neutral and unknown classes, by Random Forest likelihood score. It is trained using pathogenic and neutral variants obtained from the VariBench dataset. It uses information about evolutionary conservation of sequences, biochemical properties of amino acids, physical and if available, functional annotations of variation sites. It consistently showed more significant performance in comparison to existing tools. In 10-fold cross-validation test, its accuracy and MCC are 0.86 and 0.71, respectively. PON-P2 is a powerful tool for showing and ranking deleterious variants (Niroula *et al.*, 2015).

ENTPRISE webserver can predict disease-associated/cancer driver variations. It also performs excellently and better than some of the established methods such as Mutation Assessor and PolyPhen-2 for cancer driver prediction on the COSMIC data. The combination of the parameters (amino acid type, sequence entropy and contacting the residue composition) makes ENTPRISE do well in all over tests (Zhou *et al.*, 2016).

Integrated Functional inference of SNVs in human (iFish) is also a new online web server, which had flexible functionalities and user-friendly interfaces that permit users to analyze large sets of nsSNVs rapidly. iFish could predict the pathogenicity of human nsSNVs with increased accuracy. It outperformed other existing techniques on independent benchmarking datasets. It supports the collection of genetic evidence into the expectation of pathogenicity using a Bayesian model. Inputs specified by the chromosome, position, reference allele and an alternative allele of each nsSNV. It also enables registered users to create their gene-specific classifiers using their variations, which is not available with other existing tools. (iFish) accomplished high accuracies and outperformed other existing tools (Wang and Wei, 2016).

Screening for Non-Acceptable Polymorphisms (SNAP) is also a neural network-based method for the expectation of the functional effects of nsSNVs. SNAP uses features for the residue conservation within sequence families as protein structure and annotations, when available. It receipts protein sequences and lists of mutants and delivers a score for each substitution, which can then convert into binary predictions of a neutral or non-neutral effect (Hepp *et al.*, 2015).

Polymorphism Phenotyping v2 (Polyphen-2) is a structure and sequence-based method that determines the structural and functional consequences of nsSNVs. The PolyPhen-2 that capable of analyzing vast volumes of data created by Next-generation sequencing projects calculates the probability that a Bayesian classifier damages a nsSNP. It is an automatic tool for the expectation of the probable effect of an amino acid exchange on the structure and function of a human protein (Adzhubei *et al.*, 2013).

Another combined based tool *SNPs & GO* is a method based on SVM to predict disease-related mutations from the protein sequence, which uses information derived from evolutionary information. Protein sequence and function as encoded in the Gene Ontology (GO) terms annotation to expect if a given mutation can be secret as disease-related or neutral (Hepp *et al.*, 2015). It was implemented and updated by Capriotti *et al.* (2012). Its input comprises the sequence and its structure (when available), a set of target variants and its functional Gene Ontology (GO) terms, while the output of the server delivers, for each protein variation, is the probabilities to be related to human diseases. It tested on a massive set of marked variations extracted from the SwissVar database. The results of the server are (79% and 83%) overall accuracy for the sequence-based and structure-based inputs, respectively (Capriotti *et al.*, 2013b).

Pseudo Amino Acid Composition (PaPI) is a new machine-learning tool to sort and score human coding variations by estimating the probability to damage their protein-related function. The combination of pseudo amino acid composition, evolutionary conservation, and homologous proteins based methods outperform various prediction techniques, and it is also able to score multiple variants such as deletions, insertions, and indels (Mi *et al.*, 2016). The Position-Specific Independent Count (PSIC) of the results of the server ranges from 0 to 1. The categorization of the nsSNVs results is probably damaging and possibly damaging (PSIC > 0.5) or Benign (PSIC < 0.5) (Hepp *et al.*, 2015).

Consensus Tools With Combined Features

Consensus tools based on the integration of various tools. It provides information that needs to design a novel protein with desired characteristic and stability (Kulshreshtha *et al.*, 2016). Some classifying tools have been improved their accuracy by integrating them with other tools. Where each tool built on the specific dataset, the combination of these individual tools in a consensus classifier is enough to improve the classification capacity. Many studies cooperate in this kind of classifications.

PON-P is a Meta classifier that combines the outputs from the five methods (SIFT, PhD-SNP, PolyPhen-2, SNAP and I-Mutant 3.0) to predict the possibility that a nsSNP will affect protein function and may so be disease-related. It utilizes RF for predicting whether variants affect functioning and lead to diseases. The (PON-P) classifies the nsSNVs as neutral, unclassified or pathogenic with a corresponding possibility of pathogenicity, and provides the data existing in the UniProt database for each entry (Hepp *et al.*, 2015).

Predict SNP1.0 is an SNP classifier technique that merges six forecast techniques (MAPP, PhD-SNP, PolyPhen-1, PolyPhen-2, SIFT, and SNAP) to gain a consensus expectation of the result of the amino acid substitution. The six prediction tools run by a dataset of non-redundant mutations. The individual confidence scores are transformed to percentages to allow comparison and

the individual predictions combined in the consensus prediction. Most of these tools based on machine learning methods, primarily designed to classify neutral and deleterious mutations by sequence and structural parameters. It provides a consensus prediction with improved accuracy and efficiency over individual integrated tools (Bendl *et al.*, 2014).

Combined Annotation Dependent Depletion (Condel) is also a consensus tool which developed by integrating SIFT, PolyPhen-2, and Mutation Assessor. It collects, assembles and presents the results obtained by these tools. Condel can use as a web server or standalone tool (run on the computer after downloading).

Recently, *Rare Exome Variant Ensemble Learner (REVEL)* developed by Ioannidis *et al.* (2016) is a standard technique for expecting the pathogenicity of missense variants by individual tools. MutPred, FATHMM, VEST, Poly-Phen, SIFT, PROVEAN, Mutation Assessor, Mutation Taster, LRT, GERP, SiPhy, phyloP, and past Cons. REVEL had the best global performance as compared to any individual tool and seven ensemble methods: Meta SVM, MetaLR, KGGSeq, Condel, CADD, DANN, and Eigen. It is a standard technique that outperforms existing tools for distinguishing pathogenic variants from rare neutral variants (Ioannidis *et al.*, 2016).

Meta-SNP presented by Capriotti *et al.* (2013a,b) as a new approach for the discovery of illness-associated ns-SNVs that integrates four existing strategies: PANTHER, PhD-SNP, SIFT, and SNAP find that the Meta-SNP algorithm realizes better performance than the best single predictor. The results showed that the combination of forecasts from various resources is a good plan for the selection of high-reliability forecasts Capriotti *et al.* (2013a).

Consensus Variant Effect Classification (COVEC) is a Meta classifier that mixes the expected results of four methods SIFT, PolyPhen2, SNPs & GO and Mutation Assessor. The COVEC method outperforms unique methods and highlights the advantage of merging results from multiple tools. The SVM-based consensus classifier combines the raw output scores generated from these tools. The SVM model performs the classification using either a linear kernel or a radial based function (RBF) (Frousios *et al.*, 2013).

Consensus tools enhance accuracy and efficiency of the unique tools and predict the results better than any other tool. These tools provide a platform for comparing the results achieved by different tools in a familiar place and hence, reduce the efforts required to study the data by using different tools individually (Kulshreshtha *et al.*, 2016).

Web Servers as a Tool for Predicting the Snvs

Computational approaches, online web server, are tools for expecting the impact of SNVs on protein stability and function. After selecting the suitable database and determining the parameters affecting function and stability of the mutant protein, predicting the influence of SNVs on protein function and stability can do with the help of different computational approaches and tools. These tools are built on machine learning methods to detect and use non- redundant features that required for forecasting the effects of amino acid substitutes on protein function and classifying mutations as damaging or neutral (Kulshreshtha *et al.*, 2016).

Recent studies have indicated that the altering human behaviors have a significant influence on genetic variation (Bermejo *et al.*, 2014). Here, we launch a comprehensive review on an individual and ensemble servers for mutation analysis. We collect the most common unique tools in (Table 1) and ensemble tools in (Table 2) with their Web site, MLT that based on, Input parameters, Deleterious Threshold & Outputs, and references. The performance of the most widely used tools in predicting nsSNVs in (Table 3) through ACC (accuracy), AUC (Area under the curve), and MCC (Math well Correlation Coefficient). In additional to, Accuracy of the best singular and consensus tools that recently developed displayed in (Fig. 2).

Conclusion

In the last decade, many tools have been progressed to expect the effect of SNPs and mutation on genomic location, and on translated protein (synonymous and non-synonymous SNPs) that was considered as unsolved problems earlier (Kulshreshtha *et al.*, 2016). Recently, consensus tools have established by incorporating many tools together, which provided single window results for comparison purposes (Kaminker *et al.*, 2007). It also noted that the best ensemble scores not require a massive number of component scores to improve performance significantly, but need integrated component scores that are specific to protein features to be a marked improvement in performance. For example, KGGSeq, which integrated five module scores (SIFT, PolyPhen-2, LRT, Mutation Taster and PhyloP), performed healthy than (CADD), which integrated more than 40 component scores (Dong *et al.*, 2015).

In this review, we have performed a comprehensive study of several popular and recent prediction techniques to discriminate the pathogenic nsSNVs. It provides the fundamental knowledge based on sequence, structure and evolutionary relationship. These tools provide more reliable and accurate prediction due to compelling and comparing data from various tools. We found that some of them updated for already existing servers, others are the novel servers, while others are ensemble servers (integrated individual). Several previous studies have provided comparisons between many servers to highlight the most accurate and efficient servers and to combine them in an ensemble (Meta) server that enhances the expecting performance.

Finally, after doing a comprehensive review on all these categories of computational approaches, most studies conclude that Meta predictions are better than an individual tool. Many advanced computational approaches have been established over the years, to forecast the function and stability of a mutated protein. These methodologies based on structure, sequence features and combined features (both structure and sequence features) provide an accurate valuation of the influence of amino acid exchange on function and stability protein. By comparing the results, we realized from the range of servers performance (Table 3) and (Fig. 2) that REVEL, CADD, COVEC, Meta-SNP, PAPI, and Condel are better tools than other ensemble tools and PON-P2, SNP dryad, CHASM, PROVEAN, PhD-SNP[®], and Mutation Assessor are better tools than other individual tools. The consensus techniques have the highest overall performance.

Table 1 Shed lights on the most common individual tools with their main features as input parameters, MLT that based on, forecast scores and respective websites

ID	Server name	Website	MLT that based on	(Input features through the user)	Deleterious threshold & outputs	References
1	PON-PS	http://structure.bmc.lu.se/PON-PS	RF	Protein sequences and amino acid substitution.	Neutral, sever, non sever	Niroula and Vihinen (2017)
2	PhD-SNPg	http://snps.biofold.org/phdsnp	Gradient boosting algorithm	chromosome position, and Protein variation (position, and First amino acid, and second amino acid variant)	If the probability is > 0.5 then the SNV is predicted to be Pathogenic otherwise Benign	Capriotti and Fariselli (2017)
3	PON-P2	http://structure.bmc.lu.se/PON-P2/genomic_submission.html	RF	Protein sequences and amino acid substitution.	Pathogenic, neutral or unknown tolerance	Niroula et al. (2015)
4	SNP dryad	http://snps.ccr.utoronto.ca:8080/SNPdryad/DNAQuery.html	RF	Human genome version, Chromosome location and variant allele	Deleterious forecast score (DPS)	Wong and Zhang (2014)
5	Polyphen-2	http://genetics.bwh.harvard.edu/pph2/	Empirical rules	Protein, SNP identifier or Protein sequence in FASTA format and positions of the substitution.	Probably damaging or Benign, or Possibly damaging, Sensitivity, specificity, Multiple sequence alignment and 3D Visualization	Adzhubei et al. (2013)
6	SNAP2	https://www.roslab.org/services/SNAP/	ANN	Protein sequence	Non-neutral and neutral. Score and accuracy	Zhang et al. (2014)
7	Entprise	http://cssb.biology.gatech.edu/entprise	A boosted tree regression	Protein variation (First amino acid, position, and second amino acid variant) and protein RefSeq ID (not mRNA ID)	Pathogenicity score, deleterious (if score > 0.45) and Sequence entropy	Zhou et al. (2016)
8	Mutation assessor	http://mutationassessor.org/r3/	Conservation method	Genome build, chromosome position, reference allele and substituted allele or Protein ID and variant.	(VC) Variant conservation score and (VS) Variant specificity score	Kulshreshtha et al. (2016), Reva et al. (2011)
9	SIFT	http://sift.jcvi.org/www/SIFT_help.html#SIFT_OUTPUT	Alignment scores	The original amino acid, the position of the substitution and the new amino acid.	Score (0–1). The predicted is damaging if the score <= 0.05 and tolerated if the score > 0.05.	Hepp et al. (2015), Ng (2003), Kumar et al. (2009)
10	SNPs & GO	http://snps-and-go.biocomp.unibo.it/snps-and-go/	SVM	UniProt Accession Number, Mutation Position, Substituting residue and Wild-type residue.	Mutations are disease-related (If output = 0) and neutral polymorphism (If output = 1)	Calabrese et al. (2009), Capriotti et al. (2013b)
11	PROVEAN	http://provean.jcvi.org/protein_batch_submit.php?species=human	Delta alignment score	Query protein sequence in FASTA format, Position, reference amino acids and variant amino acids.	The forecast is deleterious if score <= -2.5 The forecast is neutral if score >= -2.5	Hepp et al. (2015)
12	FATHMM	http://fathmm.biocompute.org.uk/fathmmMKL.htm Or http://fathmm.biocompute.org.uk/	HMM	Chromosome, position, reference base and mutant base	P-values in the range [0, 1]: standards above 0.5 (deleterious), while those under 0.5 (neutral or benign)	Shihab et al. (2013)
13	Mutpred	http://mutpred.mutdb.org/	RF	Protein sequence, list of amino acid substitutions and an email address	Score probability to determine neutral or disease variants	Li et al. (2009)

(Continued)

Table 1 Continued

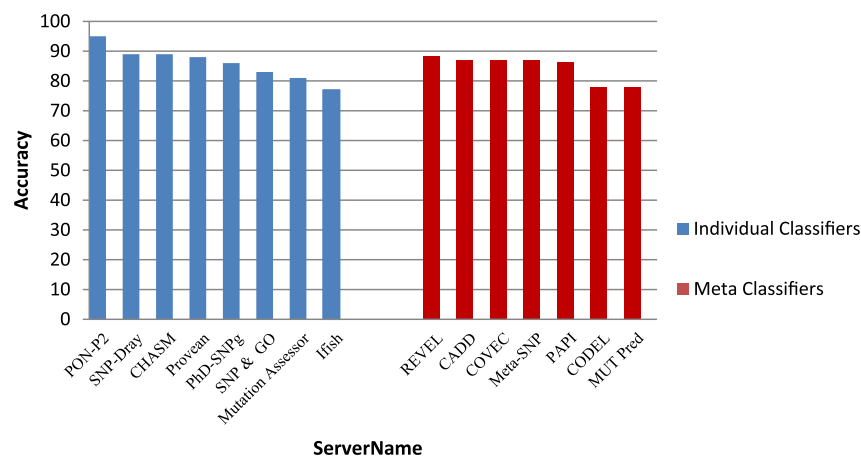
ID	Server name	Website	MLT that based on	(Input features through the user)	Deleterious threshold & outputs	References
14	PANTHER	http://www.pantherdb.org/about.jsp	Alignment Scores HMM	Protein sequence and substitution	All GO annotations & Phylogenetic annotation	Mi <i>et al.</i> (2016)
15	CHASM	http://www.cravat.us	RF	A protein accession identifier, codon number, reference and variant amino acid residues	Score for each variant of a mixture of the known driver and synthetic passenger mutations	Carter <i>et al.</i> (2009), Wong <i>et al.</i> (2011)
16	Hope	http://www.cmbi.ru.nl/hope/method Or http://www.cmbi.ru.nl/hope/	Wicket web framework	FASTA seq & list of amino acid substitutions	The wild-type residue conserved or not, and annotation about the protein	
17	HANSA	http://hansa.cdfd.org.in:8080/	SVM	Protein IDs or FASTA seq and List of amino acid substitutions	Neutral or disease	Acharya and Nagarajaram (2012)
18	PMut	http://mmb2.pcb.ub.es:8080/PMut/	ANN	FASTA seq and list of amino acid substitutions	A pathogenicity directory ranging from 0 to 1 (directories > 0.5 signal pathological mutations) and a confidence index (low or high)	Ferrer-Costa <i>et al.</i> (2005)
19	MuD (Mutation Detector)	http://mud.tau.ac.il/overview.php	RF	FASTA seq and list of variants (reference amino acids, Position, and amino acids)	Deleterious or neutral mutation	Wainreb <i>et al.</i> (2010)
20	VarMod (modeling the functional effects of non-synonymous variants)	http://www.wasslab.org/varmod/	SVM	FASTA sequence, Uniprot Id, and list of variants (reference amino acids, Position, and variant amino acids).	VarMod Probability	
21	EFIN (Evaluation of functional impact of non-synonymous SNPs)	http://147.8.193.83/EFIN/Query_DNA_loc.html	RF	Protein uniprot Accession id, reference amino acids, Position and variant amino acids	Damaging/neutral	
22	Ifish	http://fish.cbi.pku.edu.cn/submit.html	SVM	Chromosome, Position, Reference Base and Mutant Base	Neutral or Deleterious	Wang and Wei (2016)
23	Net Disease SNP (a sequence conservation-based predictor of the pathogenicity of mutations)	http://www.cbs.dtu.dk/services/NetDiseaseSNP/	ANNs	FASTA seq and [protein uniprot Accession id] [AAref] [Loc] [AAvar]	(Score > = 0.5: disease; score < 0.5: neutral)	

Table 2 Shed lights on the most common consensus tools with their main features as input parameters (output of individual tools), MLT that based on, forecast scores and respective websites

ID	Server Name	Website	MLT that based on	(Input parameters)	Deleterious threshold & outputs	References
1	CONDEL	http://bg.upf.edu/fannsnb/sgin?next=%2Fannsnb%2Fquery%2Fcondel Or http://bg.upf.edu/condel	A weighted average of the normalized notches of the individual methods (WAS)	A weighted average of the scores of Mutation Assessor and FATHMM.	(0.0 = Neutral, 1.0 = Deleterious)	
2	Meta-SNP	Snps.biofold.org/metasnps/pages/help.html	RF	A score of PANTHER, PhD-SNP, SIFT, and SNAP	Disease-related or Polymorphic non-synonymous SNVs.	Capriotti <i>et al.</i> (2013a)
3	PredictSNP	http://loschmidt.chemi.muni.cz/predictsnp	Weighted majority vote consensus	A score of PolyPhen-1, PolyPhen-2, SIFT, MAPP, PhD-SNP, SNPs&GO	Confidence scores and neutral or deleterious	Bendl <i>et al.</i> (2014)
4	CAROL	http://www.sanger.ac.uk/resources/software/carol/	A weighted Z method	A score of SIFT, PolyPhen2	The scaled scores (Pk) range between Neutral (0) and Deleterious (0)	Lopes <i>et al.</i> (2012)
5	Meta RL & Meta SVM	https://omictools.com/functional-forecasts-category	MMAF, linear kernel, radial kernel and polynomial kernel	A score of SIFT, PolyPhen-2, GERP ++, Mutation Taster, Mutation Assessor, FATHMM, LRT, SiPhy and PhyloP	D (Deleterious), N (Neutral) and U (Unknown)	Dong <i>et al.</i> (2015)
6	PON-P	http://bioinf.uta.fi/PON-P/	RF	A score of SIFT, PolyPhen2, SNAP, PhD-SNP, and I-Mutant	(0.5) uncertain class, (0) maximally certain neutral variant effect And (1) absolute certainty variant is deleterious (0.0 = Neutral, 1.0 = Deleterious)	Olatubosun <i>et al.</i> (2012)
7	Meta predict	http://www.folowsite.com/www.metapredict.eu	A weighted average of the normalized notches of the individual methods (WAS)	A score of SIFT, PolyPhen2, Condel, CHASM, mCluster, logRE, SNAP, and Mutation Assessor		
8	COVEC	http://www.dcs.kcl.ac.uk/pg/frousiok/variants/index.html	Weighted majority Vote Score (WMV) & SVM	A Score of SIFT, PolyPhen2, SNPs and GO and Mutation Assessor	Damaging (+2), neutral (-2), intermediate (0)	Frousios <i>et al.</i> (2013)
9	REVEL	https://sites.google.com/site/revelgenomics/	RF	A score of MutPred, FATHMM, VEST, Poly-Phen, SIFT, PROVEAN, Mutation Assessor, MutationTaster, LRT, GERP, SiPhy, phyloP, and phastCons.	Disease variants or rare neutral variants	Ioannidis <i>et al.</i> (2016)
10	CADD	http://cadd.gs.washington.edu/score	A linear kernel support vector machine (SVM)	SNVs	Functional, deleterious, and pathogenic variants	Kircher <i>et al.</i> (2014)
11	DANN	https://cbcl.ics.uci.edu/public_data/DANN/	A deep neural network (DNN)	SNVs	"Simulated" or "observed" variants	Quang <i>et al.</i> (2015)
12	rfPred	http://www.sbiim.fr/rfPred/	RF	A Score of SIFT, Polyphen2, LRT, PhyloP, and MutationTaster	Neutral or deleterious	Jabot-Hanin <i>et al.</i> (2016)
13	PAPI	http://papi.unipv.it/	RF	A score PolyPhen2, SIFT, and RF	Damaging or tolerated and confidence score	Mi <i>et al.</i> (2016)
14	Can predict (Cancer predict)	https://www.gene.com/research/genentech/canpredict	RF	A score of SIFT, LogR.	Activating or inactivating variants, evaluate score, and GO log-odds score for each variation.	Kaminker <i>et al.</i> (2007)

Table 3 Shed lights on the performance of the best individual and ensemble tools as ACC (accuracy), AUC (Area under the curve), and MCC (Math well Correlation Coefficient)

NO.	Server name	% ACC	% AUC	% MCC
1	PhD-SNPg	86	92	72
2	iFish (integrated Functional inference of SNVs in human)	76.99	0.73	0.54
3	SNP dryad	89	70–94	–
4	Prophen1 (Polymorphism Phenotyping v1)	65.8–70	65–69	31–39
5	Prophen2 (Polymorphism Phenotypingv2)	62–75	63–94	27–75
6	SNAP (screening for Non – Acceptable Polymorphisms)	63.1–68	66.7–79	26.3–37
7	Entprise	62.6–74.2	67–81.8	25.4–49.3
8	Mutation Assessor	57–81	74–89	40.2 62
9	Mutation Taster	63.8–71.78	66.2–86	30.5–49
10	SIFT (sorting intolerant from tolerant)	64–75	66–87	30–82
11	SNPs & GO	82–83	89–93	65–68
12	PROVEAN (Protein Variation Effect Analyzer)	77.4–88	78.3–84	46–74
13	FATHMM (Functional Analysis Through Hidden Markov Models)	59.2–73.07	63.7–93	18.2–45
14	Mutpred	50.4–78.8	83.5–50.3	1–54
15	PANTHER (Protein Analysis Through Evolutionary Relationships)	75	84	52
16	KGGseq	–	87–92	–
17	PON-P2	86–95	–	71–80
18	PON-PS	61–65	–	22–29
19	CONDEL (Consensus Deleteriousness score of missense SNVs)	64–78	85–90	33–58
20	Meta-SNP	56.2–87	73–75	20.5–35.1
21	Predict-SNP	66.2–70.8	72–78	33.2–43.3
22	CAROL (Combined Annotation Scoring Tool)	74.5–85.2	–42–	64
23	Meta LR (Logistic Regression)	59.8	63.5–95.16	–
24	Meta SVM (Support Vector Machine)	58.2	63.08–93.3	–
25	PON-P	75	86	40–73
26	COVEC (Consensus Variant Effect Classification)	87	–	74
27	PaPI (Pseudo amino acid composition to score human protein–coding variants)	86.4	92.6	0.74
28	CHASM (Cancer–specific High–throughput Annotation of Somatic Mutations)	50–89	79–99.6	8–79
29	REVEL (Rare Exome Variant Ensemble Learner)	89	90–95.7	–
30	CADD (Combined Annotation Dependent Depletion)	76.20–87	77–92	0.53–74
31	DANN (Deep Artificial Neural Network)	66.1	70–95.6	–

**Fig. 2** Accuracy of the best individual and ensemble classifiers.

We concluded that consensus classifiers are better than single ones. We recommend using multiple prediction algorithms as well as more available genetic level information may help to enhance the predictive power. This review is a useful user guide for future studies based on web servers for prediction the impact of genetic variants in human diseases.

See also: Detecting and Annotating Rare Variants. Disease Biomarker Discovery. Identification and Extraction of Biomarker Information. Natural Language Processing Approaches in Bioinformatics. Prediction of Protein-Binding Sites in DNA Sequences. Single Nucleotide Polymorphism Typing

References

- Acharya, V., Nagarajaram, H.A., 2012. Hansa: An automated method for discriminating disease and neutral human nsSNPs. *Hum. Mutat.* 33, 332–337.
- Adzhubei, I., Jordan, D.M., Sunyaev, S.R., 2013. Predicting functional effect of human missense mutations using PolyPhen-2. *Curr. Protoc. Hum. Genet.* (Chapter 7: Unit7 20).
- Adzhubei, I.A., Schmidt, S., Peshkin, L., *et al.*, 2010. A method and server for predicting damaging missense mutations. *Nat. Methods* 7, 248–249.
- Bendl, J., Stourac, J., Salanda, O., *et al.*, 2014. PredictSNP: Robust and accurate consensus classifier for prediction of disease-related mutations. *PLOS Comput. Biol.* 10, e1003440.
- Bermejo, C.N.H., Poch, O., Thompson, J., 2014. A comprehensive study of small non-frameshift insertions/deletions in proteins and prediction of their phenotypic effects by a machine learning method (KD4i). *BMC Bioinform.* 15, 111.
- Calabrese, R., Capriotti, E., Fariselli, P., Martelli, P.L., Casadio, R., 2009. Functional annotations improve the predictive score of human disease-related mutations in proteins. *Hum. Mutat.* 30, 1237–1244.
- Capriotti, E., Altman, R.B., Bromberg, Y., 2013a. Collective judgment predicts disease-associated single nucleotide variants. *BMC Genom.* 14 (Suppl 3), S2.
- Capriotti, E., Calabrese, R., Fariselli, P., *et al.*, 2013b. WS-SNPs&GO: A web server for predicting the deleterious effect of human protein variants using functional annotation. *BMC Genom.* 14 (Suppl 3), S6.
- Capriotti, E., Fariselli, P., 2017. PhD-SNPg: A webserver and lightweight tool for scoring single nucleotide variants. *Nucleic Acids Res.* doi:10.1093/nar/gkx369.
- Carter, H., Chen, S., Isik, L., *et al.*, 2009. Cancer-specific high-throughput annotation of somatic mutations: Computational prediction of driver missense mutations. *Cancer Res.* 69, 6660–6667.
- Castellana, S., Fusilli, C., Mazzoccoli, G., *et al.*, 2017. High-confidence assessment of functional impact of human mitochondrial non-synonymous genome variations by APOGEE. *PLOS Comput. Biol.* 13, e1005628.
- Dong, C., Wei, P., Jian, X., *et al.*, 2015. Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies. *Hum. Mol. Genet.* 24, 2125–2137.
- Ferrer-Costa, C., Gelpi, J.L., Zamakola, L., *et al.*, 2005. PMUT: A web-based tool for the annotation of pathological mutations on proteins. *Bioinformatics* 21, 3176–3178.
- Frousios, K., Iliopoulos, C.S., Schlitt, T., Simpson, M.A., 2013. Predicting the functional consequences of non-synonymous DNA sequence variants – Evaluation of bioinformatics tools and development of a consensus strategy. *Genomics* 102, 223–228.
- Gallion, J., Koiré, A., Katsonis, P., *et al.*, 2017. Predicting phenotype from genotype: Improving accuracy through more robust experimental and computational modeling. *Hum. Mutat.* 38, 569–580.
- Hepp, D., Gonçalves, G.L., de Freitas, T.R., 2015. Prediction of the damage-associated non-synonymous single nucleotide polymorphisms in the human MC1R gene. *PLOS ONE* 10, e0121812.
- Ioannidis, N.M., Rothstein, J.H., Pejaver, V., *et al.*, 2016. REVEL: An ensemble method for predicting the pathogenicity of rare missense variants. *Am. J. Hum. Genet.* 99, 877–885.
- Jabot-Hanin, F., Varet, H., Tores, F., Alcais, A., Jais, J.-P., 2016. doi:10.1101/037127.
- Kaminker, J.S., Zhang, Y., Waugh, A., *et al.*, 2007. Distinguishing cancer-associated missense mutations from common polymorphisms. *Cancer Res.* 67, 465–473.
- Kircher, M., Witten, D.M., Jain, P., *et al.*, 2014. A general framework for estimating the relative pathogenicity of human genetic variants. *Nat. Genet.* 46, 310–315.
- Kulshreshtha, S., Chaudhary, V., Goswami, G.K., Mathur, N., 2016. Computational approaches for predicting mutant protein stability. *J. Comput. Aided Mol. Des.* 30, 401–412.
- Kumar, P., Henikoff, S., Ng, P.C., 2009. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat. Protoc.* 4, 1073–1081.
- Li, B., Krishnan, V.G., Mort, M.E., *et al.*, 2009. Automated inference of molecular mechanisms of disease from amino acid substitutions. *Bioinformatics* 25, 2744–2750.
- Lopes, M.C., Joyce, C., Ritchie, G.R., *et al.*, 2012. A combined functional annotation score for non-synonymous variants. *Hum. Hered.* 73, 47–51.
- Mi, H., Poudel, S., Muruganujan, A., Casagrande, J.T., Thomas, P.D., 2016. PANTHER version 10: Expanded protein families and functions, and analysis tools. *Nucleic Acids Res.* 44, D336–D342.
- Ng, P.C., 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Res.* 31, 3812–3814.
- Niroula, A., Urolagin, S., Vihinen, M., 2015. PON-P2: Prediction method for fast and reliable identification of harmful variants. *PLOS ONE* 10, e0117380.
- Niroula, A., Vihinen, M., 2017. Predicting severity of disease-causing variants. *Hum. Mutat.* 38, 357–364.
- Olatubosun, A., Valiäho, J., Harkonen, J., Thusberg, J., Vihinen, M., 2012. PON-P: Integrated predictor for pathogenicity of missense variants. *Hum. Mutat.* 33, 1166–1174.
- Quang, D., Chen, Y., Xie, X., 2015. DANN: A deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics* 31, 761–763.
- Reva, B., Antipin, Y., Sander, C., 2011. Predicting the functional impact of protein mutations: Application to cancer genomics. *Nucleic Acids Res.* 39, e118.
- Shihab, H.A., Gough, J., Cooper, D.N., *et al.*, 2013. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Hum. Mutat.* 34, 57–65.
- Sim, N.L., Kumar, P., Hu, J., *et al.*, 2012. SIFT web server: Predicting effects of amino acid substitutions on proteins. *Nucleic Acids Res.* 40, W452–W457.
- Wainreb, G., Ashkenazy, H., Bromberg, Y., *et al.*, 2010. MuD: An interactive web server for the prediction of non-neutral substitutions using protein structural data. *Nucleic Acids Res.* 38, W523–W528.
- Wang, M., Wei, L., 2016. iFish: Predicting the pathogenicity of human nonsynonymous variants using gene-specific/family-specific attributes and classifiers. *Sci. Rep.* 6, 31321.
- Wong, W.C., Kim, D., Carter, H., *et al.*, 2011. CHASM and SNVBox: Toolkit for detecting biologically important single nucleotide mutations in cancer. *Bioinformatics* 27, 2147–2148.
- Wong, K.C., Zhang, Z., 2014. SNPdryad: Predicting deleterious non-synonymous human SNPs using only orthologous protein sequences. *Bioinformatics* 30, 1112–1119.
- Worth, C.L., Preissner, R., Blundell, T.L., 2011. SDM – A server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Res.* 39, W215–W222.
- Xi, T., Jones, I.M., Mohrenweiser, H.W., 2004. Many amino acid substitution variants identified in DNA repair genes during human population screenings are predicted to impact protein function. *Genomics* 83, 970–979.
- Zhang, J., Liu, J., Sun, J., *et al.*, 2014. Identifying driver mutations from sequencing data of heterogeneous tumors in the era of personalized genome sequencing. *Brief Bioinform.* 15, 244–255.
- Zhou, H., Gao, M., Skolnick, J., 2016. ENTPRISE. An algorithm for predicting human disease-associated amino acid substitutions from sequence entropy and predicted protein structures. *PLOS ONE* 11, e0150965.

Relevant Websites

<http://bioinf.uta.fi/VariBench>

A benchmark database for variations. Nair PS, Vihinen M. VariBench: A Benchmark Database for Variations. *Hum Mutat.* 2013, 34(1):42–9. PUBMED.

<http://swissvar.expasy.org>

ExPASy Bioinformatic Resource Portal.

Genome Analysis – Identification of Genes Involved in Host-Pathogen Protein-Protein Interaction Networks

Fransiskus Xavierius Ivan, Jie Zheng, and Chee-Keong Kwoh, Nanyang Technological University, Singapore
Vincent TK Chow, National University of Singapore, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

Among various types of molecular networks, host-pathogen protein-protein interaction (HP-PPI) networks have been considered as one of the most important types (Stebbins, 2005; Korkin *et al.*, 2011; Zoraghi and Reiner, 2013) in the study of infectious diseases. Advances in understanding the nature of HP-PPIs could enlighten ourselves as to the principles of mechanisms for the infection mechanisms by pathogens. A few principles have been proposed, including the specificity and multiplicity of pathogen proteins in targeting the host proteins, and the preference of pathogen proteins in targeting the hubs of the host protein-protein interaction networks (Arnold *et al.*, 2012). Recently, the centrality of a pathogen protein in the host-pathogen interactome, but not pathogen interactome, is shown to correlate with the fitness of the pathogen during *in vivo* infection (Asensio *et al.*, 2017). Knowledge of such network wiring would pave the way for implementing network medicine which may provide novel strategies for determining therapeutic targets and altering a disease state to a normal state (Barabási *et al.*, 2011).

To uncover HP-PPI networks, several high-throughput experimental techniques have been available and reviewed by Nicod *et al.* (2017). Of particular interest is the yeast two-hybrid (Y2H), an *in vivo* technique which detects an interaction based on the reconstitution of a functional transcription factor. This method has been used to uncover interactions between human proteins and proteins from varying viruses, including hepatitis C virus (De Chasseay *et al.*, 2008; Dolan *et al.*, 2013), herpesvirus (Lee *et al.*, 2011), influenza virus (Shapira *et al.*, 2009) and various flaviviruses (Le Breton *et al.*, 2011; Khadka *et al.*, 2011; Mairiang *et al.*, 2013) and oncoviruses (Calderwood *et al.*, 2007; Gulbahce *et al.*, 2012; Muller *et al.*, 2012; Rozenblatt-Rosen *et al.*, 2012). To some extent, it has also been limitedly used to uncover interactions between human proteins and effectors of varying bacteria and parasites, including *Bacillus anthracis* (Dyer *et al.*, 2010), *Burkholderia mallei* (Memisević *et al.*, 2013), *Escherichia coli* (Blasche *et al.*, 2014), *Francisella tularensis* (Dyer *et al.*, 2010; Wallqvist *et al.*, 2015), *Mycobacterium tuberculosis* (Mehra *et al.*, 2013), *Yersinia pestis* (Dyer *et al.*, 2010; Yang *et al.*, 2011), and *Toxoplasma gondii* (Liu *et al.*, 2017). However, note that although Y2H technique can test the interaction between any pair of proteins, it may not capture the interactions between proteins that require posttranslational modification in their original organisms and proteins that are not localized in the nucleus.

In addition to Y2H, high-throughput proteomic approaches have also gained popularity for uncovering HP-PPIs in the recent years. Among them, mass spectrometry (MS) based methods are the most common ones. Unlike Y2H, MS methods can provide information related to posttranslational modifications and capture interactions in physiologically relevant conditions. Additionally, MS methods also allow quantification of proteome to compare changes of proteins complexes between biological conditions. Unfortunately, most MS methods are not highly sensitive and sometimes require high amounts of purified protein complexes that are hard to be obtained (Nicod *et al.*, 2017). One of the most promising MS methods is coupled with affinity purification (AP), and it has been mainly used to study HP-PPIs during virus infections. These include infections by hepatitis C virus (Germain *et al.*, 2014; Ramage *et al.*, 2015), herpesvirus (Davis *et al.*, 2015), HIV (Jäger *et al.*, 2011; Kane *et al.*, 2015), influenza virus (Watanabe *et al.*, 2014), and various flaviviruses (Colpitts *et al.*, 2011) and oncoviruses (Rozenblatt-Rosen *et al.*, 2012; White *et al.*, 2012a). Further highlights to the application of MS methods on resolving HP-PPIs can be found in Gerold *et al.* (2016). MS methods have also been limitedly used to study HP-PPIs during infections by bacteria and parasites, including *Salmonella* (Auweter *et al.*, 2011), *Chlamydia trachomatis* (Mirrashidi *et al.*, 2015) and *Plasmodium falciparum* (Swearingen *et al.*, 2016).

Another emerging approach to identifying genes involved in HP-PPI networks is the dual RNA-seq technique which simultaneously captures transcripts in both the host and pathogen organisms (Westermann *et al.*, 2017). The method has been applied to studying interactions between human/animal proteins and proteins from various pathogens, including *Chlamydia trachomatis* (Humphrys *et al.*, 2013), *Lawsonia intracellularis* (Vannucci *et al.*, 2013), *Escherichia coli* (Mavromatis *et al.*, 2015), *Mycobacterium bovis* (Rienksma *et al.*, 2015), *Haemophilus influenzae* (Baddal *et al.*, 2015), *Salmonella typhi* (Avraham *et al.*, 2015; Westermann *et al.*, 2016) and *Streptococcus pneumoniae* (Aprianto *et al.*, 2016). For revealing the HP-PPI networks, a correlation between host's and pathogen's gene expression profiles can be reliably used; however, it must be noted that both direct and indirect interactions may be detected by the correlation methods (Reid and Berriman, 2013).

Despite their promising performance, experimental high-throughput approaches are costly and labor-intensive. Moreover, as discussed above, the resultant datasets may have inherent noises, biases and limited coverage (see Xing *et al.*, 2016 for detailed discussions on technological issues and flaws of various experimental methods). Thus, in light of limited reliability and voluminous nature of HP-PPI data, predictive computational methods that integrate genome-wide data sources would be a great complement to the experimental techniques in exploring the full range of possible genes involved in HP-PPIs.

Here, we review the latest achievements in the integrated genome-wide computational prediction of HP-PPI networks. First, we describe a workflow for predicting HP-PPI networks and highlight two different principles for detecting interacting pairs of proteins. Secondly, we illustrate several case studies in the area of human infectious diseases and categorize them according to the

types of pathogens. We highlight several aspects that need to be considered when uncovering related HP-PPI networks. Finally, we discuss issues on the methods, highlight efforts that have been made to address them, and outline future developmental works in the area.

Workflow for Predicting Host-Pathogen Protein-Protein Interactions

From a survey of the literature and previous literature reviews (Arnold *et al.*, 2012; Zhou *et al.*, 2013a; Nourani *et al.*, 2015; Sen *et al.*, 2016), we have recognized that a workflow for predicting genes involved in HP-PPI networks can be divided into four stages: (1) retrieval of host-pathogen genomes and PPI data, (2) detecting potential HP-PPIs, (3) filtering predicted HP-PPIs, and (4) validating predicted HP-PPIs. These stages are explained in detail in the following subsections.

Retrieval of Host-Pathogen Genomes/Proteomes and Protein-Protein Interaction Data

To conduct an investigation on HP-PPI networks, we must first collect genomes or proteomes of host and pathogen of interest. If available, such data can be retrieved from free genome or proteome repositories in NCBI (NCBI Resource Coordinators, 2017), ENA (Toribio *et al.*, 2017), ENSEMBL (Kersey *et al.*, 2016), or UniProt (The UniProt Consortium, 2017), as well as specialized ones such as CryptoDB (Heiges *et al.*, 2006), ToxoDB (Gajria *et al.*, 2008), PlasmoDB (Aurrecochea *et al.*, 2009), TubercuList (Lew *et al.*, 2011), GeneDB (Logan-Klumpner *et al.*, 2012), EuPathDB (Aurrecochea *et al.*, 2017), HIV Sequence Databases (Kuiken *et al.*, 2003), and many others.

Next, we require protein-protein interaction (PPI) datasets that can be considered as the ground truth for predicting new HP-PPIs. A number of specialized or generalized databases have been available, and they mainly include intra-species protein-protein interaction data. Some databases that maintain HP-PPI datasets have also been developed, including PHISTO (Durmuş *et al.*, 2013), VirHostNet (Guirimand *et al.*, 2015), HPIDB (Ammari *et al.*, 2016) and PHI-base (Urban *et al.*, 2017). In addition to those databases, PDB (Rose *et al.*, 2017) is also a valuable resource for collecting interaction data along with their protein structures; while 3did database (Mosca *et al.*, 2014) a secondary database that provides information about 3D interacting domains extracted from PDB. The list of PPI databases that have been used for investigations and are still actively maintained is given in Table 1. One important note is that the databases may contain PPIs that are uncovered from literature by an automated text mining approach. A recent review of the development of text mining approaches for HP-PPI recovery from literatures is given by Durmuş *et al.* (2015).

Table 1 Repositories for gold standard intra-species and inter-species (host-pathogen) protein-protein interactions

Organisms	Database	Link	References
Any	Database of Interacting Proteins (DIP)	http://dip.mbi.ucla.edu/dip/	Salwinski <i>et al.</i> (2004)
Any	PIBASE	https://salilab.org/pibase	Davis and Sali (2005)
Any	Biologic Interaction and Network Analysis (BIANA)	http://sbi.imim.es/web/BIANA.php	Garcia-Garcia <i>et al.</i> (2010)
Any	Molecular INTERaction (MINT)	http://mint.bio.uniroma2.it/	Licata <i>et al.</i> (2012)
Any	Three-dimensional interacting domains (3did)	https://3did.irbbarcelona.org	Mosca <i>et al.</i> (2014)
Any	IntAct	https://www.ebi.ac.uk/intact/	Orchard <i>et al.</i> (2014)
Any	Biological General Repository for Interaction Datasets (BioGRID)	https://thebiogrid.org/	Chatr-Aryamontri <i>et al.</i> (2017)
Any	Protein Data Bank (PDB)	https://www.rcsb.org/	Rose <i>et al.</i> (2017)
Any-any	Host-Pathogen Interaction Database (HPIDB)	http://agbase.msstate.edu/hpi/main.html	Ammari <i>et al.</i> (2016)
Any-prokaryotic or eukaryotic pathogen	Pathogen-Host Interaction database (PHI-base)	http://www.phi-base.org/	Urban <i>et al.</i> (2017)
Any-virus	VirHostNet	http://virhostnet.prabi.fr/	Guirimand <i>et al.</i> (2015)
Bacteria	PATRIC	https://www.patricbrc.org/	Wattam <i>et al.</i> (2017)
Human	Human Protein Reference Database (HPRD)	http://www.hprd.org/	Prasad <i>et al.</i> (2009)
Human	Human Proteinpedia Interaction	http://www.humanproteinpedia.org/	Muthusamy <i>et al.</i> (2013)
Human-any	PHISTO	http://www.phisto.org/	Durmuş <i>et al.</i> (2013)
Fly	Drosophila Interaction Database (DroID)	http://www.droidb.org/	Murali <i>et al.</i> (2011)
Mammals	MIPS mammalian protein-protein interaction database (MPPI)	http://mips.helmholtz-muenchen.de	Pagel <i>et al.</i> (2005)
<i>Plasmodium falciparum</i>	PlasmoDB	http://plasmodb.org	Aurrecochea <i>et al.</i> (2009)

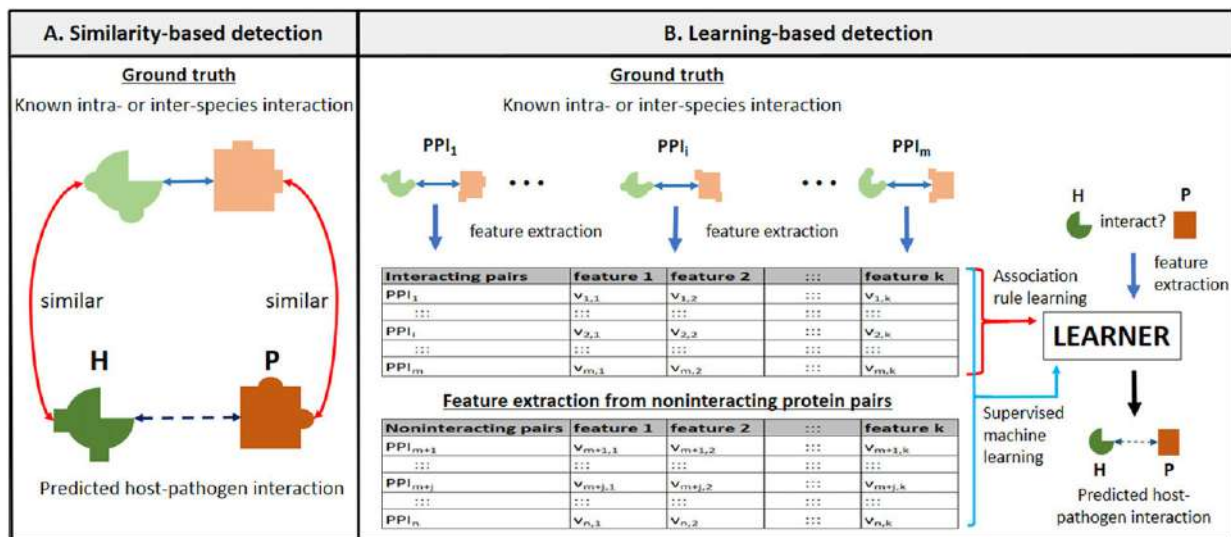


Fig. 1 Two principles for detecting potential HP-PPI between host protein H and pathogen protein P. (A) Similarity-based detection: H and P interact if each of them is similar to a different component of a known interacting protein pair. The similarity can be either based on sequences, domains, motifs, or structures. (B) Learning-based detection: The interaction between H and P is decided by a learner that is constructed using an association rule learning or a supervised machine learning approach. Association rule learning relies only on feature dataset extracted from known protein-protein interactions (PPIs), while supervised machine learning must be trained using feature datasets from both interacting and noninteracting protein pairs.

Detecting Potential Host-Pathogen Protein-Protein Interactions

At this stage, potential protein-protein interactions are detected by using known intra-species PPI or HP-PPI datasets as the ground truth. In general, there are two classes of methods for detecting potential interactions: similarity-based and learning-based methods. In principles, similarity-based methods compare the sequences, domains, motifs, or structures of candidate interacting protein pairs with those of known interacting protein pairs; while learning-based methods generate features from candidate and known interacting protein pairs for training a classifier or constructing rules for predictions. Illustration of these principles is presented in Fig. 1. Although both similarity- and learning-based approaches can be combined, here we explain each method separately.

Similarity-based detection methods

Similarity-based detection methods rely on the assumption that “If protein A and protein B are known to interact, and we have A' and B' are respectively similar to A and B in terms of sequence, domain, motif, or 3D information structures, then we may conclude that A' and B' are potential to interact”. Sequence similarity indicates homology or evolutionary relationship between sequences, and thus the method is often referred to as a homology-based method. Predicted interactions by homology-based methods are called interologs. Several databases have been developed to store information on homologous sequences, which include OrthoMCL-DB (Fischer *et al.*, 2011), InParanoid (Sonnhammer and Östlund, 2015), NCBI HomoloGene (NCBI Resource Coordinators, 2017) and OrthoDB (Zdobnov *et al.*, 2017); and thus, inference on interologs could be eased with the presence of such databases. In the absence of homologous information, homology between a host or a pathogen protein with a known interacting protein can be assessed by using BLASTP (Altschul *et al.*, 1990). For capturing distant similarity, PSI-BLAST (Altschul *et al.*, 1997; Schäffer *et al.*, 2001) should be considered. Tools for quick ortholog (homolog due to speciation) assignments between two genomes from different species, which include free tools that are used to create OrthoDB, OrthoMCL-DB, and InParanoid, may also be employed.

Similarly, classification of domain or motif of protein sequences can be checked in domain or motif databases (especially Pfam database (Finn *et al.*, 2016)) and used for predicting protein-protein interactions. Domain or motif of new protein sequences (proteins that are not in the databases) can be classified using various domain or motif search tools, including TMHMM (Krogh *et al.*, 2001), ScanProsite (De Castro *et al.*, 2006), HMMER (Mistry *et al.*, 2013), InterProScan (Jones *et al.*, 2014), RPS-BLAST (Marchler-Bauer *et al.*, 2015) and ELM (Dinkel *et al.*, 2016). For 3D structure, similar protein structures in PDB can be obtained in Dali Database (Holm and Rosenström, 2010) or SCOP2 (Andreeva *et al.*, 2014), while similarity search for query protein structures against protein structures in PDB can be done using DaliLite (Holm *et al.*, 2008). DaliLite, or alternatively TM-align (Zhang and Skolnick, 2005) may also be used for comparing two protein structures in a local database; alternatively. If only protein sequences are available, 3D structures of the proteins can be first predicted before performing similarity search. The

Table 2 Repositories for similar protein sequences, domains, motifs and structures

Information	Database	Link	References
Homologous sequences	OrthoMCL-DB2	http://orthomcl.org	Fischer <i>et al.</i> (2011)
	InParanoid	http://inparanoid.sbc.su.se/cgi-in/index.cgi	Sonnhammer and Östlund (2015)
	NCBI HomoloGene	https://www.ncbi.nlm.nih.gov/homologene	NCBI Resource Coordinators (2017)
	OrthoDB	http://www.orthodb.org/	Zdobnov <i>et al.</i> (2017)
Domain and motif classification	MuPSSM	http://mulpssm.mbu.iisc.ernet.in/	Gowri <i>et al.</i> (2006)
	PROSITE	http://prosite.expasy.org/prosite.html	Sigrist <i>et al.</i> (2013)
	InterPro	http://www.ebi.ac.uk/interpro	Mitchell <i>et al.</i> (2015)
	Pfam	http://pfam.xfam.org/	Finn <i>et al.</i> (2016)
Structural classification	Dali Database	http://ekhidna.biocenter.helsinki.fi/dali/start	Holm and Rosenström (2010)
	SCOP2	http://scop2.mrc-lmb.cam.ac.uk/	Andreeva <i>et al.</i> (2014)

Table 3 Sequence, domain, motif and structure similarity search tools

Type of search	Tool	Link	References
Sequence	BLASTP	https://blast.ncbi.nlm.nih.gov/Blast.cgi	Altschul <i>et al.</i> (1990)
	Position-Specific Iterated BLAST (PSI-BLAST)	https://blast.ncbi.nlm.nih.gov/Blast.cgi	Altschul <i>et al.</i> (1997), Schäffer <i>et al.</i> (2001)
Domain and motif	TMHMM	http://www.cbs.dtu.dk/services/TMHMM/	Krogh <i>et al.</i> (2001)
	ScanProsite	http://prosite.expasy.org/scanprosite/	De Castro <i>et al.</i> (2006)
	HMMER	http://hmmer.org/	Mistry <i>et al.</i> (2013)
	InterProScan	http://www.ebi.ac.uk/interpro/search/sequence-search	Jones <i>et al.</i> (2014)
	Reverse Position-Specific BLAST (RPS-BLAST)	https://www.ncbi.nlm.nih.gov/Structure/cdd/docs/cdd_search.html	Marchler-Bauer <i>et al.</i> (2015)
	Eukaryotic Linear Motif (ELM) prediction tool	http://elm.eu.org/	Dinkel <i>et al.</i> (2016)
Structure	TM-align	https://zhanglab.ccmb.med.umich.edu/TM-align/	Zhang and Skolnick (2005)
	DaliLite	http://ekhidna2.biocenter.helsinki.fi/dali/	Holm <i>et al.</i> (2008)

prediction can be done by using software such as I-TASSER (Yang *et al.*, 2015) and MODELLER (Webb and Sali, 2016) and utilizing PDB's structural information as a template. Respectively, **Tables 2** and **3** summarize repositories and search tools that are available publicly.

Learning-based detection methods

Learning-based detection methods have been widely applied for predicting HP-PPIs. Supervised machine learning methods, such as logistic regression, multilayer perceptron, random forest and support vector machine, have been applied in the presence of positive and negative datasets (i.e., interacting and noninteracting protein pairs). Regardless of the methods they use, and whether it is a single-task or multi-task learning, or involves transfer learning or not, generation of features for labeled data is crucial to apply supervised learning-based methods. The features have been extracted from various resources, including features from interacting proteins (sequence, domain, and motif features), known HP-PPI data (HP-PPIs with experimental evidence), network topology (mainly from host interactome), Gene Ontology (GO) annotations, and other varying biological contexts. A summary of features that have been used for prediction is shown in **Table 4**. Among them, *k*-mer composition of host and pathogen proteins has been the most popular feature being used.

Nonetheless, some authors (Mukhopadhyay *et al.*, 2010, 2012, 2014; Mondal *et al.*, 2012; Ray *et al.*, 2012; Nouretdinov *et al.*, 2012) have argued that negative dataset or a list of noninteracting protein pairs cannot be defined precisely. In this case, they proposed the use of association rule learning, a data mining approach for discovering interesting relations between items in a large dataset. Highlights about their methods are presented in the case studies.

Filtering Predicted Host-Pathogen Protein-Protein Interactions

Detection methods usually come along with a threshold parameter that needs to be set in order to obtain potential interactions with significant evidence. For example, when using BLASTP for detecting homology or sequence similarity, we must determine a cut-off for E-value for reasonable results. The choice of the parameter is made to maximize the proportion of true interactions without getting a significant proportion of negative interactions. In addition to filters associated with detection methods,

Table 4 Features exploited for learning-based detections of host-pathogen protein-protein interactions

<i>Source of information</i>	<i>Extracted features</i>	<i>References</i>
Sequences	Statistics based on the k-mer composition of host and pathogen proteins	Dyer <i>et al.</i> (2011), Wuchty (2011), Cui <i>et al.</i> (2012), Kshirsagar <i>et al.</i> (2012, 2013, 2017), Barman <i>et al.</i> (2014), Coelho <i>et al.</i> (2014), Dong <i>et al.</i> (2015), Abbasi and Minhas (2016), Eid <i>et al.</i> (2016), Jindalertudomdee <i>et al.</i> (2016), Nourani <i>et al.</i> (2016), Chen <i>et al.</i> (2017)
	Sequence feature engineering (e.g., ProFET features (Ofer and Linial, 2015))	Abbasi and Minhas (2016)
	E-value of local similarity of pathogen protein to host protein and host protein's binding partners	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Kshirsagar <i>et al.</i> (2012), Nourtdinov <i>et al.</i> (2012)
Domain and motif information	Statistics based on sequence homology (evolutionary features)	Kshirsagar <i>et al.</i> (2012), Coelho <i>et al.</i> (2014), Abbasi and Minhas (2016)
	Statistics of interacting domains	Dyer <i>et al.</i> (2011), Kshirsagar <i>et al.</i> (2012), Barman <i>et al.</i> (2014), Coelho <i>et al.</i> (2014), Nourani <i>et al.</i> (2016)
Known host-pathogen protein-protein interactions	The presence of motif-ligand interactions	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Nourtdinov <i>et al.</i> (2012)
	Matrix representation of host-pathogen interactions	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Mukhopadhyay <i>et al.</i> (2010, 2012, 2014), Mondal <i>et al.</i> (2012), Nourtdinov <i>et al.</i> (2012), Ray <i>et al.</i> (2012)
Network topology	Degree (centrality) of host protein in host PPI networks (or viral protein in known host-pathogen PPIs)	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Dyer <i>et al.</i> (2011), Kshirsagar <i>et al.</i> (2012, 2013), Nourtdinov <i>et al.</i> (2012), Barman <i>et al.</i> (2014), Dong <i>et al.</i> (2015), Jindalertudomdee <i>et al.</i> (2016), Nourani <i>et al.</i> (2016)
	Clustering coefficient of host protein in host PPI networks	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Kshirsagar <i>et al.</i> (2012, 2013), Nourtdinov <i>et al.</i> (2012), Jindalertudomdee <i>et al.</i> (2016)
	Betweenness centrality of host protein in host PPI networks	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Dyer <i>et al.</i> (2011), Kshirsagar <i>et al.</i> (2012, 2013), Nourtdinov <i>et al.</i> (2012), Dong <i>et al.</i> (2015), Jindalertudomdee <i>et al.</i> (2016)
Gene Ontology (GO) annotations	Graphlet degree vector of host protein in host PPI networks	Jindalertudomdee <i>et al.</i> (2016)
	The count of the interacting proteins that belong to each GO term	Mei (2013), Mei and Zhu (2014a,b)
	The count of the co-occurrence of GO terms for interacting proteins	Kshirsagar <i>et al.</i> (2013)
	Pairwise GO similarity for interacting proteins	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Kshirsagar <i>et al.</i> (2012, 2013), Nourtdinov <i>et al.</i> (2012), Coelho <i>et al.</i> (2014), Nourani <i>et al.</i> (2016)
Other resources	Neighbor GO similarity for interacting proteins	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Kshirsagar <i>et al.</i> (2012), Nourtdinov <i>et al.</i> (2012)
	Literature features – Concept profiles (co-occurrence statistics for interacting proteins) from abstracts of publications	Coelho <i>et al.</i> (2014)
	Gene expression features – Differential gene expression data under different control conditions	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Kshirsagar <i>et al.</i> (2012, 2013), Nourtdinov <i>et al.</i> (2012)
	RNAi expression features – Significance of pathways or protein complexes involving host protein in RNAi screenings of infected host	Kshirsagar <i>et al.</i> (2012)
	Posttranslational modification features – Binary codes for encoding the presence of shared posttranslational modification between pathogen protein and host protein's binding partner	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Nourtdinov <i>et al.</i> (2012)
	Conserved pathways – Total number of pathways shared by host-pathogen pairs and their interologs in different organisms	Kshirsagar <i>et al.</i> (2012)
	Pathway membership	Nourani <i>et al.</i> (2016)
	Percentage of disorder regions of host and pathogen proteins	Barman <i>et al.</i> (2014)
	Tissue feature – Susceptibility of a tissue as a site of infection	Tastan <i>et al.</i> (2009), Qi <i>et al.</i> (2010), Nourtdinov <i>et al.</i> (2012)
	Structural features – A variety of features derived from the structural component of host-pathogen protein pairs	Halder <i>et al.</i> (2017)

biological-context-based and structural-based filters have also been widely applied, especially when similarity-based methods are used to detect potential HP-PPIs. Biological-context-based filters utilize information related to molecular characteristics, cellular localization and functions, tissue specificity, pathogenicity, and gene expressions. Structural-based filters can be as simple as homomeric information and as complex as a score that measures interaction strength between residues in the interface of interacting host and pathogen protein pairs.

Validating Predicted Host-Pathogen Protein-Protein Interactions

Several methods have been used to validate the results of predicted host-pathogen interactions. Among them, one of the common methods is literature validation, where predicted HP-PPIs are matched with reported interactions that have experimental evidence. Another popular method is the gene set enrichment or over-representation analyses. This is commonly done by computing the significance of the overlap between the set of predicted host genes involved in HP-PPI networks and the set of proteins belong to the same GO category using a hypergeometric distribution. Specifically, if N is the total number of genes associated with a host genome, M is the number of genes belongs to a specific GO category, n is the number of predicted host genes involved in HP-PPI networks, and x is the number of predicted genes in the GO category, then the significance of observing x or more predicted genes in the GO category is given by the following formula:

$$p - value = \sum_{j=x}^n \frac{\binom{M}{j} \binom{N-M}{n-j}}{\binom{N}{n}}$$

As an alternative to GO categories, other lists of genes may also be used for enrichment analyses. These include a list of genes appeared in the same pathway or obtained from expression profiling and RNAi screening.

Network analyses have also been used to assess the prediction results of interacting proteins. Common network topology measures such as degree, betweenness centrality, shortest path length, clustering and density coefficients, number of connected components, and scale-freeness of the networks have been calculated for intra-host or host-pathogen PPIs (see [Costa et al., 2007](#) for a survey of various network measurements). Some other interesting measures that have been proposed are the distribution of distances between host protein pairs interacting with a particular pathogen protein (and vice versa; [Dyer et al., 2007](#)) and pathway participation coefficient that measures the diversity of the pathways for host protein's binding partners ([Wuchty, 2011](#)).

Some other methods for validating predicted interactions use aspects of data that are not exploited for predictions or filtering. For examples, [Tyagi et al. \(2009\)](#) used the superposition of predicted complex and template PDB structure to evaluate prediction results detected by homology- and domain-based methods, while [Dyer et al. \(2007\)](#) computed distribution of Spearman's correlation between the expression profiles of the protein pairs to evaluate predictions based on domain-based method. Additionally, methods for assessing the statistical significance of predicted interactions by using randomized interactions have also been proposed by [Doolittle and Gomez \(2010\)](#) and [Itzhaki \(2011\)](#).

Case Studies on Predicting Human-Pathogen Protein-Protein Interactions

To illustrate the application of the workflow for predicting HP-PPIs, we highlight their uses in uncovering protein-protein interactions that underlie human infectious diseases caused by viruses, bacteria, and parasites. [Table 5](#) lists studies that have been carried for each category of the pathogen. Depending on the type of the pathogen, a unique strategy might be applied to predict HP-PPIs.

Uncovering Human-Virus Protein-Protein Interactions

Human-virus interactions have been investigated for varying viruses, including dengue virus, Ebola virus, HIV, herpes simplex virus, influenza virus and varying oncoviruses (adenovirus, Epstein-Barr virus, hepatitis C virus, human papillomavirus, human T-lymphotropic virus, and Kaposi's sarcoma-associated herpesvirus). Among them, HIV has been widely studied and our discussion will focus on it. Different detection methods have been applied to identify potential interactions between human proteins and HIV proteins, including domain-/motif-based similarity ([Evans et al., 2009](#); [Segura-Cabrera et al., 2013](#); [Becerra et al., 2017](#)), structural-based similarity ([Doolittle and Gomez, 2010](#); [Cui et al., 2016](#)), supervised machine learning ([Tastan et al., 2009](#); [Qi et al., 2010](#); [Dyer et al., 2011](#); [Nouretdinov et al., 2012](#); [Mei, 2013](#); [Abbasi and Minhas, 2016](#)), and rule mining ([Mukhopadhyay et al., 2010, 2012, 2014](#); [Mondal et al., 2012](#); [Ray et al., 2012](#)).

Of those methods, the domain-/motif-based similarity and rules mining have been uniquely carried out for HIV. [Evans et al. \(2009\)](#) and [Becerra et al. \(2017\)](#) proposed a method that is based on direct interactions between host-mimicked short eukaryotic linear motifs (ELMs or SLiMs) and cluster domains (CDs) that are present in HIV-1 and human proteins, respectively. Different types of filters for ELMs – including conservation level, rarity, and association with disordered regions – have also been assessed by [Becerra et al. \(2017\)](#), demonstrating that the majority of conserved ELMs are associated with disordered regions. On the other

Table 5 Predictive host-pathogen protein-protein interaction studies related to human infectious diseases (categorized according to the types of pathogens)

<i>Pathogen</i>	<i>Detection methods^a</i>	<i>References</i>
Viruses		
Adenovirus	Learning with RF and SVM	Abbasi and Minhas (2016)
Dengue virus	Structural-based method	Doolittle and Gomez (2011)
	Domain-/motif-based method	Segura-Cabrera <i>et al.</i> (2013)
Epstein-Barr Virus (EBV)	Domain-/motif-based method	Itzhaki (2011)
Ebola	Learning with DT, KNN, NB and SVM	Halder <i>et al.</i> (2017)
	Multitask learning	Kshirsagar <i>et al.</i> (2017)
Hepatitis C Virus (HCV)	Learning with SVM	Cui <i>et al.</i> (2012)
	Domain-/motif-based method	Segura-Cabrera <i>et al.</i> (2013)
	Multitask learning	Kshirsagar <i>et al.</i> (2017)
Human Immunodeficiency Virus (HIV)	Learning with RF	Tastan <i>et al.</i> (2009)
	Domain-/motif-based method	Evans <i>et al.</i> (2009), Segura-Cabrera <i>et al.</i> (2013), Becerra <i>et al.</i> (2017)
	Structural-based method	Doolittle and Gomez (2010), Cui <i>et al.</i> (2016)
	Learning with MLP	Qi <i>et al.</i> (2010)
	Learning with ARs	Mukhopadhyay <i>et al.</i> , (2010, 2012, 2014), Mondal <i>et al.</i> (2012), Ray <i>et al.</i> (2012)
	Learning with SVM	Dyer <i>et al.</i> (2011)
	Learning with CP	Nouretdinov <i>et al.</i> (2012)
	Homology-based method and transfer learning	Mei (2013)
	Learning with RF and SVM	Abbasi and Minhas (2016)
Human Papillomavirus (HPV)	Learning with SVM	Cui <i>et al.</i> (2012), Dong <i>et al.</i> (2015)
	Domain-/motif-based method	Segura-Cabrera <i>et al.</i> (2013)
Herpes Simplex Virus (HSV)	Domain-/motif-based method	Itzhaki (2011)
Human T-cell Leukemia Virus (HTLV)	Homology-based method and transfer learning	Mei and Zhu (2014a)
Influenza	Structural-based method	De Chasseay <i>et al.</i> (2013)
	Domain-/motif-based method	Segura-Cabrera <i>et al.</i> (2013)
	Multitask learning	Kshirsagar <i>et al.</i> (2017)
Kaposi's Sarcoma-associated Human Virus (KSHV)	Domain-/motif-based method	Itzhaki (2011)
Multiple types of viruses	Homology-based method	Franzosa and Xia (2011)
	Domain-/motif-based method	Segura-Cabrera <i>et al.</i> (2013)
	Learning with NB, RF and SVM	Barman <i>et al.</i> (2014)
	Learning with SVM	Eid <i>et al.</i> (2016)
	Learning with AMPKE	Nourani <i>et al.</i> (2016)
Bacteria		
<i>Bacillus anthracis</i>	Multitask learning	Kshirsagar <i>et al.</i> (2013)
	Learning with SCW and SGD	Jindalertudomdee <i>et al.</i> (2016)
	Learning with ELM, SDA, and SVM	Chen <i>et al.</i> (2017)
<i>Clostridium difficile</i>	Learning with ELM, SDA, and SVM	Chen <i>et al.</i> (2017)
<i>Escherichia coli</i>	Homology- and domain-/motif-based methods	Krishnadev and Srinivasan (2011)
	Learning with ELM, SDA, and SVM	Chen <i>et al.</i> (2017)
<i>Francisella tularensis</i>	Multitask learning	Kshirsagar <i>et al.</i> (2013)
	Learning with SCW and SGD	Jindalertudomdee <i>et al.</i> (2016)
<i>Helicobacter pylori</i>	Homology- and domain-/motif-based methods	Tyagi <i>et al.</i> (2009)
<i>Microbes in oral cavity</i>	Ensemble learning	Coelho <i>et al.</i> (2014)
<i>Mycobacterium leprae</i>	Structural-based method	Davis <i>et al.</i> (2007)
<i>Mycobacterium tuberculosis</i>	Structural-based method	Davis <i>et al.</i> (2007)
	Domain-/motif-based method	Zhou <i>et al.</i> (2013b)
	Homology-based method	Zhou <i>et al.</i> (2014)
	Homology- and domain-/motif-based methods	Huo <i>et al.</i> (2015)
	Structural-based method	Cui <i>et al.</i> (2016)
	Domain-/motif-based method	Mahajan and Mande (2017)
<i>Salmonella</i>	Homology- and domain-/motif-based methods	Schleker <i>et al.</i> (2012)
<i>Salmonella typhi</i>	Homology- and domain-/motif-based methods	Krishnadev and Srinivasan (2011)
	Multitask learning	Kshirsagar <i>et al.</i> (2013)
	Homology-based method and transfer learning	Mei and Zhu (2014b)
	Learning with SCW and SGD	Jindalertudomdee <i>et al.</i> (2016)

Table 5 Continued

Pathogen	Detection methods ^a	References
<i>Yersinia pestis</i>	Homology- and domain-/motif-based methods Multitask learning Learning with SCW and SGD	Krishnadev and Srinivasan (2011) Kshirsagar <i>et al.</i> (2013) Jindalertudomdee <i>et al.</i> (2016)
Parasites		
<i>Aspergillus fumigatus</i>	Homology-based method	Remmele <i>et al.</i> (2015)
<i>Candida albicans</i>	Homology-based method	Remmele <i>et al.</i> (2015)
<i>Cryptosporidium spp.</i>	Structural-based method	Davis <i>et al.</i> (2007)
<i>Leishmania major</i>	Structural-based method	Davis <i>et al.</i> (2007)
<i>Plasmodium falciparum</i>	Domain-/motif-based method	Dyer <i>et al.</i> (2007), Liu <i>et al.</i> (2014)
	Homology- and domain-/motif-based methods	Krishnadev and Srinivasan (2008)
	Homology-based method	Lee <i>et al.</i> (2008)
	Homology-based method and learning with RF	Wuchty (2011)
<i>Plasmodium spp.</i>	Structural-based method	Davis <i>et al.</i> (2007)
<i>Trypanosoma spp.</i>	Structural-based method	Davis <i>et al.</i> (2007)

^aAcronyms for learning-based methods – AMPKE: adapted multiple kernel preserving embedding; ARs: association rules; CP: conformal predictor; DT: decision tree; ELM: extreme learning machine; KNN: k-nearest neighbor; MLP: multi-layer perceptron; NB: naïve Bayes; RF: random forest; SCW: soft confidence-weighted; SDA: stacked denoising autoencoder; SGD: stochastic gradient descent; SVM: support vector machine.

hand, Segura-Cabrera *et al.* (2013) utilized motif information from 3did datasets, in combination with a filter based on surface accessibility of motif residues.

The second method extracts association rules from only known interactions using a biclustering for predicting new interactions between HIV and human proteins (Mukhopadhyay *et al.*, 2010, 2012, 2014; Mondal *et al.*, 2012; Ray *et al.*, 2012). For this, the known interactions are presented as a matrix with rows and columns represent the viral and human proteins, respectively. Each entry of the matrix indicates a type of interaction. For extracting association rules as done for market basket or transaction data analysis, we consider each row as a transaction and each column as an item. If we have $[HP_1 HP_2 \dots HP_k] \rightarrow [HP_{k+1} HP_{k+2} \dots HP_m]$ as a highly confident rule, where each HP_i represents a host protein, then we basically infer that HP_{k+1} , HP_{k+2} , ..., and HP_m interact with each of viral protein that interacts with each of HP_1 , HP_2 , ..., and HP_k .

Of particular interest from those studies is small overlaps between the sets of interactions predicted using different approaches, which is a standing issue in HP-PPI prediction (Mukhopadhyay *et al.*, 2012, 2014; Segura-Cabrera *et al.*, 2013). Despite such inconsistency, HP-PPI predictions that utilize homology-based detection method and structural information has helped to provide insights about structural principles governing the interactions between human and viral proteins (Franzosa and Xia, 2011). These include the characteristics of HP-PPI interfaces that mimic host's PPI interfaces, but they are governed by their own structural, functional and evolutionary principles.

Uncovering Human-Bacterial Protein-Protein Interactions

Various detection methods have been applied independently or in different combinations to uncover varying human-bacteria protein-protein interactions. Of interest, Zhou *et al.* (2013b) proposed a stringent domain-based prediction method. The method first detects the PPIs using the domain-domain interaction approach that employs Bayesian formula (Dyer *et al.*, 2007; as explained in the next section). Then, a layer of assessment is applied; where for each predicted PPI, a score that indicates the strength of each interaction is calculated based on the interacting residues in their interaction interfaces extracted from the 3did database. Using this layer of assessment, the authors claim that the stringent method is better than the conventional one. A different layer of assessment has been proposed by Mahajan and Mande (2017). It employs the Smith-Waterman local alignment-based comparison of bacterial and human proteins with domain-domain interaction template sequences, and it is followed by the calculation of the geometric mean of the similarities for scoring the interaction.

In another human-*M. tuberculosis* PPIs study, Zhou *et al.* (2014) introduced a stringent homology-based method which predicts human-*M. tuberculosis* PPIs from known eukaryote-prokaryote inter-species PPIs. Such stringency takes into account the differences between (1) the structures of eukaryote and prokaryote genes, and (2) the interfaces of intra- and inter-species protein-protein interactions. The authors claimed that the stringent approach performs better than the conventional method that infers human-*M. tuberculosis* interologs from inter-species PPIs.

Uncovering Human-Parasites Protein-Protein Interactions

Studies in this category have frequently explored the interactions between human and *Plasmodium falciparum*, the deadliest *Plasmodium* that causes malaria disease. For detecting the interactions, the homology-, domain-, and structural-based methods have been employed independently by Davis *et al.* (2007), Dyer *et al.* (2007), Lee *et al.* (2008) and Liu *et al.* (2014). In addition, Krishnadev and Srinivasan (2008) have also combined homology- and domain-based methods for the detection, while Wuchty

(2011) applied a random forest classifier to a dataset containing the composition of amino acid triplet classes of interacting proteins in order to obtain more reliable interolog candidates.

Of interest, Dyer *et al.* (2007) applied the Bayesian formula to calculate the likelihood that the host and pathogen proteins interact given their domains. In particular, for host protein H and pathogen protein P that have domain h and p, respectively, the likelihood is computed as follows:

$$\Pr(I(P, H)|D(P, p) \cap D(H, h)) = \frac{\Pr(D(P, p) \cap D(H, h)|I(P, H))\Pr(I(P, H))}{\Pr(D(P, p) \cap D(H, h))}$$

where $I(P, H)$ denotes the event that protein P and H interact and $D(P, p) \cap D(H, h)$ denotes the event that protein P has domain p and protein H has domain h. Each term on the right-hand side can be calculated from intra-species PPIs and the domains present in each of the interacting proteins. If multiple pairs of domains (p_i, h_j) are contained in the pair of proteins, then the likelihood of the interaction is estimated by assuming the independence of domain pairs, which gives the following formula:

$$\Pr(I(P, H)|\cap_i \cap_j (D(P, p_i) \cap D(H, h_j))) = 1 - \prod_i \prod_j (1 - \Pr(I(P, H)|D(P, p_i) \cap D(H, h_j)))$$

The application of expectation-maximization algorithm to calculate this likelihood has also been proposed by Liu *et al.* (2014).

Another particular note from studies on human-*P. falciparum* PPIs is on how the candidate of interactions are further filtered with various biological contexts (Lee *et al.*, 2008; Krishnadev and Srinivasan, 2008; Wuchty, 2011; Liu *et al.*, 2014). These include filtering with: (1) various molecular characteristics of parasite proteins (e.g., the presence of transmembrane domains, signaling peptides and host cell-targeting signals); (2) cellular localization (proteins localized in nucleus or mitochondria are not expected to interact); (3) cellular functions (ubiquitin proteins, DNA polymerases, and splicing factors are not expected to interact); and (4) expression data in the corresponding human tissue and parasite's cell cycle stage (e.g., host proteins in liver tissue and parasite proteins in the sporozoite stage; host proteins in erythrocyte and pathogen proteins in merozoite stage).

Discussion

In the previous section, we illustrated the application of the workflow for HP-PPI prediction to investigating human-pathogen PPIs. One important observation is that a particular strategy may or may not be useful for prediction involving a particular type of pathogen, or a particular type of interaction. For example, the Bayesian method proposed by Dyer *et al.* (2007) may not be useful for investigating human-HIV interactions due to the absence of viral domains; instead, the utilization of ELMs or SLiMs would be more appropriate to capture host-viral interactions, such as works of Evans *et al.* (2009) and Becerra *et al.* (2017). For another example, we ought to utilize tools for predicting the presence of extracellular or trans-membrane domains if we focus on the interactions that drive invasion and intracellular signaling of bacteria or parasites. Hence, knowledge about biological and biochemical contexts would be very valuable when predicting HP-PPI networks (see Sen *et al.* (2016) for further insights).

Nonetheless, like other integrative genomic studies, computational HP-PPI studies are still in their infancy and many issues still need to be addressed before they become fruitful. First, each HP-PPI detection method comes with its own limitation. Sequence, domain, and motif similarity-based methods tend to produce many false positives; structure similarity-based methods are constrained by the available protein structures in the database; supervised learning methods lack definitive noninteracting datasets; lastly, association rule learning methods are limited by a small fraction of known interactions.

To cope with the high number of false positives produced by sequence-, domain- and motif-based similarity methods for detecting PPIs, the standard method being used is by filtering detected interactions. Various types of filters have been highlighted in the previous section, and filters based on biological context and structural information are the most popular ones. Another strategy to prevent the high number of false positives is to use a stringent criterion for PPI resources being used for comparison. For example, when predicting the interaction between human and bacterial proteins, we may follow the approach proposed by Zhou *et al.* (2014) which only uses known interactions between eukaryotic and bacterial proteins as gold standard PPIs.

Meanwhile, issues with scarcity and unavailability of gold standard data (in other words, missing data) for learning-based approaches have been resolved mainly by utilizing knowledge of homologous genes. Kshirsagar *et al.* (2012) have demonstrated that imputation of missing data for GO annotations and gene expression features with techniques that utilize homolog or cross-species information outperform other imputation techniques. In particular, they applied two phases of imputation: (1) cross-species information integration and (2) modeling-based imputation. The first phase is able to transfer a large number of cross-species attributes, but it still leaves missing values of gene expression for proteins whose homologs are unknown. The second phase estimates the rest of missing values by using a LASSO regularization technique. Mei (2013) also used homolog information features for missing data and demonstrated that prediction with a probability weighted ensemble learning that only utilizes homology features for training and testing has promising results, indicating the effectiveness of the approach. Mei and Zhu (2014a) further demonstrated the effectiveness of transferring homolog features as the second instance of proteins in predicting cross-species protein interactions using a multi-instance AdaBoost method.

As it is mentioned earlier, the lack of verifiable negative feature datasets for noninteracting protein pairs is also another problem in supervised machine learning methods for HP-PPI prediction. Assuming that the number of interacting host-pathogen protein pairs is far less than the number of noninteracting ones, such that the probability of randomly picking positive interaction is close to zero, it is very common to randomly choose the noninteracting pairs from the set of protein pairs with no interaction evidence. Nonetheless,

Yu *et al.* (2010) noted that the choice of noninteracting proteins in the training data may greatly influence the accuracy of the classifier or predictor due to the bias towards the interactions involving hub proteins in the positive dataset. To prevent the bias, Yu *et al.* (2010) performed a balanced random sampling that ensures each protein appears in the positive and negative datasets with the same frequency. In another study, Mei (2013) claimed that excluding the pairs from the same sub-cellular localization also allows the construction of a reliable and unbiased classifier. Furthermore, Eid *et al.* (2016) proposed the use of a dissimilarity threshold for alignment between pathogen protein pairs to pick unlikely interactions as negative examples. In particular, if pathogen protein y is known to interact with host protein h , and x and y are dissimilar, then we conclude that x is unlikely to interact with h .

In addition to missing data and the construction of negative dataset, practices of validating machine learning models have also been criticized. Note that the notion of validation here is different from the validation of predicted HP-PPIs in the presented workflow – validation in machine learning approach assesses how well a model has been trained, while validation in the workflow assesses the biological significance of predicted interactions. Abbasi and Minhas (2016) questioned the effectiveness of K-fold cross-validation for estimating the generalization ability of host-pathogen PPI prediction for proteins with no known interaction. In particular, they highlighted the possibility of K-fold cross-validation for not preventing both similarity and redundancy between training and testing examples within a fold, i.e., a protein in a test example in a fold can occur as part of another example in training. To resolve this, they proposed an alternative evaluation scheme called leave one pathogen protein out (LOPO) and demonstrated that it was more effective in modeling HP-PPI predictors. In relation to this proposed method, they have also proposed new performance metrics that include true hit rate (THR), false hit rate (FHR) and median rank of the first positive prediction (MRFP), which were claimed to be more intuitive for biologists.

Another issue related to learning-based methods is feature representation, which is still an ongoing research area for bioinformatics overall. As mentioned earlier, k-mer composition – especially the conjoint triad – has become one of the most popular features being used. Nonetheless, the study by Yu *et al.* (2010) has highlighted a pitfall in using such simple sequence features since some hub proteins that appear many more times in the positive set than the negative sets will lead to unrealistic accuracy. This is demonstrated by the performance of SVM learning method on balanced random sampled negative interaction datasets that is lower than the performance on purely random negative interaction sampled datasets. In another study related to human-virus interactions, Barman *et al.* (2014) demonstrated that domain-domain association appears to be the best descriptor among 44 descriptors that include amino acid composition and association with disordered regions. In a more recent study, Jindal *et al.* (2016) have explored the usefulness of graphlet degree vector (GDV), a network feature that represents the similarity of local topological structure between proteins in a host PPI. Such feature is claimed to increase the performance of stochastic gradient descent (SGD) and soft confidence-weighted (SCW) learning methods.

Finally, the major issue in applying integrative genomic approaches for predicting HP-PPIs is the fact that the overlap between predicted interactions from different approaches tend to be small, as exemplified by the human-HIV studies we discussed previously. Abstract models (e.g., the definition of the motif and the formulation of the learning method), filtering methods, and types of features extracted for interacting protein pairs and their interaction interfaces – all of them easily influence the outcome of the prediction. This issue is not yet resolved and should be paid attention to in the future.

Future Directions

Tremendous accumulation of biological data – including genomic data, 3D protein structures, and gold standard experimental HP-PPI data – could be expected to continue and their quality will be improved. On the other hand, the performance of integrative bioinformatics approach for predicting HP-PPIs would be enhanced and related algorithms are expected to be more accurate and effective. Thus, in the presence of cheaper genome data but the absence of costly gold standard protein-based interactome data, predictive methods provide an alternative to obtain valuable insights about the nature of specific HP-PPI networks, especially in the area of human infectious diseases. Hence, the development of bioinformatics tools and pipelines that allow a routine prediction and exploration of HP-PPI networks by life scientists would be expected in the next few years.

In addition to biological insights, the ultimate goal of uncovering HP-PPI networks is to drive the development of network medicine and personalized medicine for infectious diseases. Investigations of HP-PPI networks are expected to aid the discovery of new biomarkers and therapeutic targets based on gene essentiality and some other network principles. For example, it will assist the design of multiple drug combinations that target disease modules and re-route metabolic activity to compensate for the lost functions during a disease state (Barabási *et al.*, 2011). In relation to personalized medicine, a challenging problem in the prediction of HP-PPI networks is to take into account genetic make-up of both host and pathogen. A recent study by Recker *et al.* (2017) has clearly demonstrated the importance of both host and pathogen factors in determining the outcome of infections by *Staphylococcus aureus*. Hence, the ability to incorporate the host and pathogen genetic variations into HP-PPI network models should help us to understand further on why some pathogens become more virulent, or why some pathogens do not affect the host. Of interest, we like to uncover how small genetic changes could impact the activation or deactivation of disease modules. To some extent, this means that we attempt to measure the virulence of pathogens as defined by Casadevall and Pirofski (1999), i.e., the capacity of microbes to cause damage according to the interplay between unique host and pathogen properties. Such understanding will further drive the development of personalized medicine for infectious diseases, which has become more feasible these days.

Closing Remarks

We have presented recent progress in the development of computational prediction of HP-PPIs. Explicitly, we have differentiated two major principles in detection methods used in the prediction workflow for new HP-PPIs, i.e., similarity-based and learning-based methods. Case studies on human infectious diseases and some insights obtained from the studies are illustrated. Several issues on the approaches and some proposals for addressing the issues, mainly for learning based methods, are highlighted. One challenging problem to be resolved is the inconsistency of predicted HP-PPIs given by different methods. Nonetheless, while efforts for improvements are being made, existing workflows may be utilized to guide HP-PPI experimentation as well as gain further insights into principles underlying host-pathogen interactions. Finally, enhancements to the methods and overall workflows may allow us to advance the development of network and personalized medicine related to infectious diseases.

Acknowledgement

This project is supported by AcRF Tier 2 grant MOE2014-T2-2-023, Ministry of Education, Singapore.

See also: Host-Pathogen Interactions. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Sequence Analysis

References

- Abbasi, W.A., Minhas, F.U., 2016. Issues in performance evaluation for host-pathogen protein interaction prediction. *Journal of Bioinformatics and Computational Biology* 14 (3), 1650011.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 25 (17), 3389–3402.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3), 403–410.
- Ammari, M.G., Gresham, C.R., McCarthy, F.M., Nanduri, B., 2016. HPIDB 2.0: A curated database for host-pathogen interactions. *Database* (Oxford). (pii: baw103).
- Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., Murzin, A.G., 2014. SCOP2 prototype: A new approach to protein structure mining. *Nucleic Acids Research* 42 (Database issue), D310–D314.
- Aprianto, R., Slager, J., Holsappel, S., Veening, J.W., 2016. Time-resolved dual RNA-seq reveals extensive rewiring of lung epithelial and pneumococcal transcriptomes during early infection. *Genome Biology* 17 (1), 198.
- Arnold, R., Boonen, K., Sun, M.G., Kim, P.M., 2012. Computational analysis of interactomes: Current and future perspectives for bioinformatics approaches to model the host-pathogen interaction space. *Methods* 57 (4), 508–518.
- Asensio, N.C., Giner, E.M., de Groot, N.S., Burgas, M.T., 2017. Centrality in the host-pathogen interactome is associated with pathogen fitness during infection. *Nature Communications* 8, 14092.
- Aurrecochea, C., Brestelli, J., Brunk, B.P., *et al.*, 2009. PlasmoDB: A functional genomic database for malaria parasites. *Nucleic Acids Research* 37 (Database issue), D539–D543.
- Aurrecochea, C., Barreto, A., Basenko, E.Y., *et al.*, 2017. EuPathDB: The eukaryotic pathogen genomics database resource. *Nucleic Acids Research* 45 (D1), D581–D591.
- Auweter, S.D., Bhavsar, A.P., de Hoog, C.L., *et al.*, 2011. Quantitative mass spectrometry catalogues *Salmonella* pathogenicity island-2 effectors and identifies their cognate host binding partners. *The Journal of Biological Chemistry* 286 (27), 24023–24035.
- Avraham, R., Haseley, N., Brown, D., *et al.*, 2015. Pathogen cell-to-cell variability drives heterogeneity in host immune responses. *Cell* 162 (6), 1309–1321.
- Baddal, B., Muzzi, A., Censini, S., *et al.*, 2015. Dual RNA-seq of nontypeable *Haemophilus influenzae* and host cell transcriptomes reveals novel insights into host-pathogen cross talk. *mBio* 6 (6), e01765–15.
- Barabási, A.L., Gulbahce, N., Loscalzo, J., 2011. Network medicine: A network-based approach to human disease. *Nature Reviews Genetics* 12 (1), 56–68.
- NCBI Resource Coordinators, 2017. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research* 45 (D1), D12–D17.
- Barman, R.K., Saha, S., Das, S., 2014. Prediction of interactions between viral and host proteins using supervised machine learning methods. *PLOS ONE* 9 (11), e112034.
- Becerra, A., Bucheli, V.A., Moreno, P.A., 2017. Prediction of virus-host protein-protein interactions mediated by short linear motifs. *BMC Bioinformatics* 18 (1), 163.
- Blasche, S., Arens, S., Ceol, A., *et al.*, 2014. The EHEC-host interactome reveals novel targets for the translocated intimin receptor. *Scientific Reports* 4, 7531.
- Calderwood, M.A., Venkatesan, K., Xing, L., *et al.*, 2007. Epstein-Barr virus and virus human protein interaction maps. *Proceedings of the National Academy of Sciences of the United States of America* 104 (18), 7606–7611.
- Casadevall, A., Pirofski, L.A., 1999. Host-pathogen interactions: Redefining the basic concepts of virulence and pathogenicity. *Infection and Immunity* 67 (8), 3703–3713.
- Chatr-Aryamontri, A., Oughtred, R., Boucher, L., *et al.*, 2017. The BioGRID interaction database: 2017 update. *Nucleic Acids Research* 45 (D1), D369–D379.
- Chen H., Shen J., Wang L., Song J., 2017. Collaborative data analytics towards prediction on pathogen-host protein-protein interactions. In: *Proceedings of the International Conference on Computer Supported Cooperative Work in Design*, pp. 269–274. United States: IEEE.
- Coelho, E.D., Arrais, J.P., Matos, S., *et al.*, 2014. Computational prediction of the human-microbial oral interactome. *BMC Systems Biology* 8, 24.
- Colpitts, T.M., Cox, J., Nguyen, A., *et al.*, 2011. Use of a tandem affinity purification assay to detect interactions between West Nile and dengue viral proteins and proteins of the mosquito vector. *Virology* 417 (1), 179–187.
- Costa, L.D.F., Rodrigues, F.A., Travieso, G., Boas, P.R.V., 2007. Characterization of complex networks: A survey of measurements. *Advances in Physics* 56 (1), 167–242.
- Cui, G., Fang, C., Han, K., 2012. Prediction of protein-protein interactions between viruses and human by an SVM model. *BMC Bioinformatics* 13 (Suppl. 7), S5.
- Cui, T., Li, W., Liu, L., Huang, Q., He, Z.G., 2016. Uncovering new pathogen-host protein-protein interactions by pairwise structure similarity. *PLOS ONE* 11 (1), e0147612.
- Davis, F.P., Sali, A., 2005. PIBASE: A comprehensive database of structurally defined protein interfaces. *Bioinformatics* 21 (9), 1901–1907.
- Davis, F.P., Barkan, D.T., Eswar, N., McKerrow, J.H., Sali, A., 2007. Host pathogen protein interactions predicted by comparative modeling. *Protein Science* 16 (12), 2585–2596.
- Davis, Z.H., Verschueren, E., Jang, G.M., *et al.*, 2015. Global mapping of herpesvirus-host protein complexes reveals a transcription strategy for late genes. *Molecular Cell* 57 (2), 349–360.
- De Castro, E., Sigrist, C.J., Gattiker, A., *et al.*, 2006. ScanProsite: Detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. *Nucleic Acids Research* 34 (Web Server issue), W362–W365.

- De Chassey, B., Navratil, V., Tafforeau, L., *et al.*, 2008. Hepatitis C virus infection protein network. *Molecular Systems Biology* 4, 230.
- De Chassey, B., Meyniel-Schicklin, L., Aublin-Gex, A., *et al.*, 2013. Structure homology and interaction redundancy for discovering virus-host protein interactions. *EMBO Reports* 14 (10), 938–944.
- Dinkel, H., Van Roey, K., Michael, S., *et al.*, 2016. ELM 2016 – Data update and new functionality of the eukaryotic linear motif resource. *Nucleic Acids Research* 44 (D1), D294–D300.
- Dolan, P.T., Zhang, C., Khadka, S., *et al.*, 2013. Identification and comparative analysis of hepatitis C virus-host cell protein interactions. *Molecular BioSystems* 9 (12), 3199–3209.
- Dong, Y., Kuang, Q., Dai, X., *et al.*, 2015. Improving the understanding of pathogenesis of human papillomavirus 16 via mapping protein-protein interaction network. *BioMed Research International* 2015, 890381.
- Doolittle, J.M., Gomez, S.M., 2010. Structural similarity-based predictions of protein interactions between HIV-1 and Homo sapiens. *Virology Journal* 7, 82.
- Doolittle, J.M., Gomez, S.M., 2011. Mapping protein interactions between Dengue virus and its human and insect hosts. *PLOS Neglected Tropical Diseases* 5 (2), e954.
- Durmuş, S., Çakır, T., Özgür, A., Guthke, R., 2015. A review on computational systems biology of pathogen-host interactions. *Frontiers in Microbiology* 6, 235.
- Durmuş, T.S., Çakır, T., Ardic, E., *et al.*, 2013. PHISTO: Pathogen-host interaction search tool. *Bioinformatics* 29 (10), 1357–1358.
- Dyer, M.D., Murali, T.M., Sobral, B.W., 2007. Computational prediction of host-pathogen protein-protein interactions. *Bioinformatics* 23 (13), i159–i166.
- Dyer, M.D., Neff, C., Dufford, M., *et al.*, 2010. The human-bacterial pathogen protein interaction networks of *Bacillus anthracis*, *Francisella tularensis*, and *Yersinia pestis*. *PLOS ONE* 5 (8), e12089.
- Dyer, M.D., Murali, T.M., Sobral, B.W., 2011. Supervised learning and prediction of physical interactions between human and HIV proteins. *Infection, Genetics and Evolution* 11 (5), 917–923.
- Eid, F.E., ElHefnawi, M., Heath, L.S., 2016. DeNovo: Virus-host sequence-based protein-protein interaction prediction. *Bioinformatics* 32 (8), 1144–1150.
- Evans, P., Dampier, W., Ungar, L., Tozeren, A., 2009. Prediction of HIV-1 virus-host protein interactions using virus and host sequence motifs. *BMC Medical Genomics* 2, 27.
- The UniProt Consortium, 2017. UniProt: The Universal Protein knowledgebase. *Nucleic Acids Research* 45 (D1), D158–D169.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Research* 44 (D1), D279–D285.
- Fischer, S., Brunk, B.P., Chen, F., *et al.*, 2011. Using OrthoMCL to assign proteins to OrthoMCL-DB groups or to cluster proteomes into new ortholog groups. *Current Protocols in Bioinformatics*. (Chapter 6: Unit 6.12.1–19).
- Franzosa, E.A., Xia, Y., 2011. Structural principles within the human-virus protein-protein interaction network. *Proceedings of the National Academy of Sciences of the United States of America* 108 (26), 10538–10543.
- Gajria, B., Bahl, A., Brestelli, J., *et al.*, 2008. ToxoDB: An integrated *Toxoplasma gondii* database resource. *Nucleic Acids Research* 36 (Database issue), D553–D556.
- García-García, J., Guney, E., Aragües, R., Planas-Iglesias, J., Oliva, B., 2010. Biana: A software framework for compiling biological interactions and analyzing networks. *BMC Bioinformatics* 11, 56.
- Germain, M.A., Chatel-Chaix, L., Gagné, B., *et al.*, 2014. Elucidating novel hepatitis C virus-host interactions using combined mass spectrometry and functional genomics approaches. *Molecular and Cellular Proteomics* 13 (1), 184–203.
- Gerold, G., Bruening, J., Pietschmann, T., 2016. Decoding protein networks during virus entry by quantitative proteomics. *Virus Research* 218, 25–39.
- Gowri, V.S., Krishnadev, O., Swamy, C.S., Srinivasan, N., 2006. MulPSSM: A database of multiple position-specific scoring matrices of protein domain families. *Nucleic Acids Research* 34 (Database issue), D243–D246.
- Guirmand, T., Delmotte, S., Navratil, V., 2015. VirHostNet 2.0: Surfing on the web of virus/host molecular interactions data. *Nucleic Acids Research* 43 (Database issue), D583–D587.
- Gulbahce, N., Yan, H., Dricot, A., *et al.*, 2012. Viral perturbations of host networks reflect disease etiology. *PLOS Computational Biology* 8 (6), e1002531.
- Halder, A.K., Dutta, P., Kundu, M., Basu, S., Nasipuri, M., 2017. Review of computational methods for virus-host protein interaction prediction: A case study on novel Ebola-human interactions. *Briefings in Functional Genomics*.
- Heiges, M., Wang, H., Robinson, E., *et al.*, 2006. CryptoDB: A *Cryptosporidium* bioinformatics resource update. *Nucleic Acids Research* 34 (Database issue), D419–D422.
- Holm, L., Rosenström, P., 2010. Dali server: Conservation mapping in 3D. *Nucleic Acids Research* 38 (Web Server issue), W545–W549.
- Holm, L., Kääriäinen, S., Rosenström, P., Schenkel, A., 2008. Searching protein structure databases with DaliLite v.3. *Bioinformatics* 24 (23), 2780–2781.
- Humphrys, M.S., Creasy, T., Sun, Y., *et al.*, 2013. Simultaneous transcriptional profiling of bacteria and their host cells. *PLOS ONE* 8 (12), e80597.
- Huo, T., Liu, W., Guo, Y., *et al.*, 2015. Prediction of host-pathogen protein interactions between *Mycobacterium tuberculosis* and *Homo sapiens* using sequence motifs. *BMC Bioinformatics* 16, 100.
- Itzhaki, Z., 2011. Domain-domain interactions underlying herpesvirus-human protein-protein interaction networks. *PLOS ONE* 6 (7), e21724.
- Jäger, S., Cimermancic, P., Gulbahce, N., *et al.*, 2011. Global landscape of HIV-human protein complexes. *Nature* 481 (7381), 365–370.
- Jindalretudomdee, J., Hayashida, M., Song, J., Akutsu, T., 2016. Host-pathogen protein interaction prediction based on local topology structures of a protein interaction network. In: *Proceeding of International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 7–12. United States: IEEE.
- Jones, P., Binns, D., Chang, H.Y., *et al.*, 2014. InterProScan 5: Genome-scale protein function classification. *Bioinformatics* 30 (9), 1236–1240.
- Kane, J.R., Stanley, D.J., Hultquist, J.F., *et al.*, 2015. Lineage-specific viral hijacking of non-canonical E3 ubiquitin ligase cofactors in the evolution of Vif anti-APOBEC3 activity. *Cell Reports* 11 (8), 1236–1250.
- Kersey, P.J., Allen, J.E., Armean, I., *et al.*, 2016. Ensembl Genomes 2016: More genomes, more complexity. *Nucleic Acids Research* 44 (D1), D574–D580.
- Khadka, S., Vangeloff, A.D., Zhang, C., *et al.*, 2011. A physical interaction network of dengue virus and human proteins. *Molecular and Cellular Proteomics* 10 (12), M111.012187.
- Korkin, D., Thieu, T., Joshi, S., Warren, S., 2011. Mining hostpathogen interactions. In: Yang, N.-S. (Ed.), *Systems and Computational Biology – Molecular and Cellular Experimental Systems*. Rijeka: InTech, pp. 163–184.
- Krishnadev, O., Srinivasan, N., 2008. A data integration approach to predict host-pathogen protein-protein interactions: Application to recognize protein interactions between human and a malarial parasite. *In Silico Biology* 8 (3–4), 235–250.
- Krishnadev, O., Srinivasan, N., 2011. Prediction of protein-protein interactions between human host and a pathogen and its application to three pathogenic bacteria. *International Journal of Biological Macromolecules* 48 (4), 613–619.
- Krogh, A., Larsson, B., von Heijne, G., Sonnhammer, E.L., 2001. Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *Journal of Molecular Biology* 305 (3), 567–580.
- Kshirsagar, M., Carbonell, J., Klein-Seetharaman, J., 2012. Techniques to cope with missing data in host-pathogen protein interaction prediction. *Bioinformatics* 28 (18), i466–i472.
- Kshirsagar, M., Carbonell, J., Klein-Seetharaman, J., 2013. Multitask learning for host-pathogen protein interactions. *Bioinformatics* 29 (13), i217–i226.
- Kshirsagar, M., Murugesan, K., Carbonell, J.G., Klein-Seetharaman, J., 2017. Multitask matrix completion for learning protein interactions across diseases. *Journal of Computational Biology* 24 (6), 501–514.
- Kuiken, C., Korber, B., Shafer, R.W., 2003. HIV sequence databases. *AIDS Reviews* 5 (1), 52–61.
- Le Breton, M., Meyniel-Schicklin, L., Deloire, A., *et al.*, 2011. Flavivirus NS3 and NS5 proteins interaction network: A high-throughput yeast two-hybrid screen. *BMC Microbiology* 11, 234.
- Lee, S., Salwinski, L., Zhang, C., *et al.*, 2011. An integrated approach to elucidate the intra-viral and viral-cellular protein interaction networks of a gamma-herpesvirus. *PLOS Pathogens* 7 (10), e1002297.

- Lee, S.A., Chan, C.H., Tsai, C.H., *et al.*, 2008. Ortholog-based protein-protein interaction prediction and its application to inter-species interactions. *BMC Bioinformatics* 9 (Suppl. 12), S11.
- Lew, J.M., Kapopoulou, A., Jones, L.M., Cole, S.T., 2011. TubercuList – 10 years after. *Tuberculosis (Edinb)* 91 (1), 1–7.
- Licata, L., Briganti, L., Peluso, D., *et al.*, 2012. MINT, the molecular interaction database: 2012 update. *Nucleic Acids Research* 40 (Database issue), D857–D861.
- Liu, Q., Li, F.C., Elsheikha, H.M., Sun, M.M., Zhu, X.Q., 2017. Identification of host proteins interacting with *Toxoplasma gondii* GRA15 (TgGRA15) by yeast two-hybrid system. *Parasit and Vectors* 10 (1), 1.
- Liu, X., Huang, Y., Liang, J., *et al.*, 2014. Computational prediction of protein interactions related to the invasion of erythrocytes by malarial parasites. *BMC Bioinformatics* 15, 393.
- Logan-Klumpler, F.J., De Silva, N., Boehme, U., *et al.*, 2012. GeneDB – An annotation database for pathogens. *Nucleic Acids Research* 40 (Database issue), D98–D108.
- Mahajan, G., Mande, S.C., 2017. Using structural knowledge in the protein data bank to inform the search for potential host-microbe protein interactions in sequence space: Application to *Mycobacterium tuberculosis*. *BMC Bioinformatics* 18 (1), 201.
- Mairiang, D., Zhang, H., Sodja, A., *et al.*, 2013. Identification of new protein interactions between dengue fever virus and its hosts, human and mosquito. *PLOS ONE* 8 (1), e53535.
- Marchler-Bauer, A., Derbyshire, M.K., Gonzales, N.R., *et al.*, 2015. CDD: NCBI's conserved domain database. *Nucleic Acids Research* 43 (Database issue), D222–D226.
- Mavromatis, C.H., Bokil, N.J., Totsika, M., *et al.*, 2015. The co-transcriptome of uropathogenic *Escherichia coli*-infected mouse macrophages reveals new insights into host-pathogen interactions. *Cellular Microbiology* 17 (5), 730–746.
- Mehra, A., Zahra, A., Thompson, V., *et al.*, 2013. *Mycobacterium tuberculosis* type VII secreted effector EsxH targets host ESCRT to impair trafficking. *PLOS Pathogens* 9 (10), e1003734.
- Mei, S., 2013. Probability weighted ensemble transfer learning for predicting interactions between HIV-1 and human proteins. *PLOS ONE* 8 (11), e79606.
- Mei, S., Zhu, H., 2014a. Computational reconstruction of proteome-wide protein interaction networks between HTLV retroviruses and *Homo sapiens*. *BMC Bioinformatics* 15, 245.
- Mei, S., Zhu, H., 2014b. AdaBoost based multi-instance transfer learning for predicting proteome-wide interactions between *Salmonella* and human proteins. *PLOS ONE* 9 (10), e110488.
- Memisević, V., Zavaljevski, N., Pieper, R., *et al.*, 2013. Novel *Burkholderia mallei* virulence factors linked to specific host-pathogen protein interactions. *Molecular and Cellular Proteomics* 12 (11), 3036–3051.
- Mirrashidi, K.M., Elwell, C.A., Verschueren, E., *et al.*, 2015. Global mapping of the Inc-human interactome reveals that retromer restricts chlamydia infection. *Cell Host and Microbe* 18 (1), 109–121.
- Mistry, J., Finn, R.D., Eddy, S.R., Bateman, A., Punta, M., 2013. Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. *Nucleic Acids Research* 41 (12), e121.
- Mitchell, A., Chang, H.Y., Daugherty, L., *et al.*, 2015. The InterPro protein families database: The classification resource after 15 years. *Nucleic Acids Research* 43 (Database issue), D213–D221.
- Mondal, K.C., Pasquier, N., Mukhopadhyay, A., *et al.*, 2012. Prediction of protein interactions on HIV1 human PPI data using a novel closure-based integrated approach. In: *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms*, pp. 164–173. Vilamoura.
- Mosca, R., Céol, A., Stein, A., Olivella, R., Aloy, P., 2014. 3did: A catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Research* 42 (Database issue), D374–D379.
- Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., 2012. A novel biclustering approach to association rule mining for predicting HIV-1-human protein interactions. *PLOS ONE* 7 (4), e32289.
- Mukhopadhyay, A., Ray, S., Maulik, U., 2014. Incorporating the type and direction information in predicting novel regulatory interactions between HIV-1 and human proteins using a biclustering approach. *BMC Bioinformatics* 15, 26.
- Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Eils, R., 2010. Mining association rules from HIV-human protein interactions. In: *Proceeding of International Conference on Systems in Medicine and Biology*, pp. 344–348. Kharagpur.
- Muller, M., Jacob, Y., Jones, L., *et al.*, 2012. Large scale genotype comparison of human papillomavirus E2-host interaction networks provides new insights for e2 molecular functions. *PLOS Pathogens* 8 (6), e1002761.
- Murali, T., Pacifico, S., Yu, J., *et al.*, 2011. DrolD 2011: A comprehensive, integrated resource for protein, transcription factor, RNA and gene interactions for *Drosophila*. *Nucleic Acids Research* 39 (Database issue), D736–D743.
- Muthusamy, B., Thomas, J.K., Prasad, T.S., Pandey, A., 2013. Access guide to human proteinpedia. *Current Protocols in Bioinformatics*. (Chapter 1: Unit 1.21).
- Nicod, C., Banaei-Esfahani, A., Collins, B.C., 2017. Elucidation of host-pathogen protein-protein interactions to uncover mechanisms of host cell rewiring. *Current Opinion in Microbiology* 39, 7–15.
- Nourani, E., Khunjush, F., Durmuş, S., 2015. Computational approaches for prediction of pathogen-host protein-protein interactions. *Frontiers in Microbiology* 6, 94.
- Nourani, E., Khunjush, F., Durmuş, S., 2016. Computational prediction of virus-human protein-protein interactions using embedding kernelized heterogeneous data. *Molecular BioSystems* 12 (6), 1976–1986.
- Nouredinov, I., Gammerman, A., Qi, Y., Klein-Seetharaman, J., 2012. Determining confidence of predicted interactions between HIV-1 and human proteins using conformal method. *Pacific Symposium on Biocomputing*, 311–322.
- Ofer, D., Linial, M., 2015. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics* 31 (21), 3429–3436.
- Orchard, S., Ammari, M., Aranda, B., *et al.*, 2014. The MintAct project – IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research* 42 (Database issue), D358–D363.
- Pagel, P., Kovac, S., Oesterheld, M., *et al.*, 2009. The MIPS mammalian protein-protein interaction database. *Bioinformatics* 21 (6), 832–834.
- Prasad, T.S.K., Goel, R., Kandasamy, K., *et al.*, 2009. Human protein reference database – 2009 update. *Nucleic Acids Research* 37 (Database issue), D767–D772.
- Qi, Y., Tastan, O., Carbonell, J.G., Klein-Seetharaman, J., Weston, J., 2010. Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. *Bioinformatics* 26 (18), i645–i652.
- Ramage, H.R., Kumar, G.R., Verschueren, E., *et al.*, 2015. A combined proteomics/genomics approach links hepatitis C virus infection with nonsense-mediated mRNA decay. *Molecular Cell* 57 (2), 329–340.
- Ray, S., Mukhopadhyay, A., Maulik, U., 2012. Predicting annotated HIV1 – Human PPIs using a biclustering approach to association rule mining. In: *Proceedings of Third International Conference on Emerging Applications of Information Technology (EAIT)*, pp. 3–6. Kolkata.
- Recker, M., Laabei, M., Toleman, M.S., *et al.*, 2017. Clonal differences in *Staphylococcus aureus* bacteraemia-associated mortality. *Nature Microbiology* 2 (10), 1381–1388.
- Reid, A.J., Berriman, M., 2013. Genes involved in host-parasite interactions can be revealed by their correlated expression. *Nucleic Acids Research* 41 (3), 1508–1518.
- Remmele, C.W., Luther, C.H., Balkenhol, J., *et al.*, 2015. Integrated inference and evaluation of host-fungi interaction networks. *Frontiers in Microbiology* 6, 764.
- Rienksma, R.A., Suarez-Diez, M., Mollenkopf, H.J., *et al.*, 2015. Comprehensive insights into transcriptional adaptation of intracellular mycobacteria by microbe-enriched dual RNA sequencing. *BMC Genomics* 16, 34.
- Rose, P.W., Prlić, A., Altunkaya, A., *et al.*, 2017. The RCSB protein data bank: Integrative view of protein, gene and 3D structural information. *Nucleic Acids Research* 45 (D1), D271–D281.
- Rozenblatt-Rosen, O., Deo, R.C., Padi, M., *et al.*, 2012. Interpreting cancer genomes using systematic host network perturbations by tumour virus proteins. *Nature* 487 (7408), 491–495.
- Salwinski, L., Miller, C.S., Smith, A.J., *et al.*, 2004. The database of interacting proteins: 2004 update. *Nucleic Acids Research* 32 (Database issue), D449–D451.
- Schäffer, A.A., Aravind, L., Madden, T.L., *et al.*, 2001. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Research* 29 (14), 2994–3005.

- Schleker, S., Garcia-Garcia, J., Klein-Seetharaman, J., Oliva, B., 2012. Prediction and comparison of Salmonella-human and Salmonella-Arabidopsis interactomes. *Chemistry and Biodiversity* 9 (5), 991–1018.
- Segura-Cabrera, A., García-Pérez, C.A., Guo, X., Rodríguez-Pérez, M.A., 2013. A viral-human interactome based on structural motif-domain interactions captures the human infectome. *PLOS ONE* 8 (8), e71526.
- Sen, R., Nayak, L., De, R.K., 2016. A review on host-pathogen interactions: Classification and prediction. *European Journal of Clinical Microbiology & Infectious Diseases* 35 (10), 1581–1599.
- Shapira, S.D., Gat-Viks, I., Shum, B.O., *et al.*, 2009. A physical and regulatory map of host-influenza interactions reveals pathways in H1N1 infection. *Cell* 139 (7), 1255–1267.
- Sigrist, C.J., de Castro, E., Cerutti, L., *et al.*, 2013. New and continuing developments at PROSITE. *Nucleic Acids Research* 41 (Database issue), D344–D347.
- Sonnhammer, E.L., Östlund, G., 2015. InParanoid 8: Orthology analysis between 273 proteomes, mostly eukaryotic. *Nucleic Acids Research* 43 (Database issue), D234–D239.
- Stebbins, C.E., 2005. Structural microbiology at the pathogen-host interface. *Cellular Microbiology* 7 (9), 1227–1236.
- Swearingen, K.E., Lindner, S.E., Shi, L., *et al.*, 2016. Interrogating the plasmodium sporozoite surface: Identification of surface-exposed proteins and demonstration of glycosylation on CSP and TRAP by mass spectrometry-based proteomics. *PLOS Pathogens* 12 (4), e1005606.
- Tastan, O., Qi, Y., Carbonell, J.G., Klein-Seetharaman, J., 2009. Prediction of interactions between HIV-1 and human proteins by information integration. *Pacific Symposium on Biocomputing*. 516–527.
- Toribio, A.L., Alako, B., Amid, C., *et al.*, 2017. European nucleotide archive in 2016. *Nucleic Acids Research* 45 (D1), D32–D36.
- Tyagi, N., Krishnadev, O., Srinivasan, N., 2009. Prediction of protein-protein interactions between *Helicobacter pylori* and a human host. *Molecular BioSystems* 5 (12), 1630–1635.
- Urban, M., Cuzick, A., Rutherford, K., *et al.*, 2017. PHI-base: A new interface and further additions for the multi-species pathogen-host interactions database. *Nucleic Acids Research* 45 (D1), D604–D610.
- Vannucci, F.A., Foster, D.N., Gebhart, C.J., 2013. Laser microdissection coupled with RNA-seq analysis of porcine enterocytes infected with an obligate intracellular pathogen (*Lawsonia intracellularis*). *BMC Genomics* 14, 421.
- Wallqvist, A., Memišević, V., Zavaljevski, N., *et al.*, 2015. Using host-pathogen protein interactions to identify and characterize *Francisella tularensis* virulence factors. *BMC Genomics* 16, 1106.
- Watanabe, T., Kawakami, E., Shoemaker, J.E., *et al.*, 2014. Influenza virus-host interactome screen as a platform for antiviral drug development. *Cell Host and Microbe* 16 (6), 795–805.
- Wattam, A.R., Davis, J.J., Assaf, R., *et al.*, 2017. Improvements to PATRIC, the all-bacterial Bioinformatics Database and Analysis Resource Center. *Nucleic Acids Research* 45 (D1), D535–D542.
- Webb, B., Salí, A., 2016. Comparative protein structure modeling using MODELLER. *Current Protocols in Protein Science* 86, 2.9.1–2.9.37.
- Westermann, A.J., Förstner, K.U., Amman, F., *et al.*, 2016. Dual RNA-seq unveils noncoding RNA functions in host-pathogen interactions. *Nature* 529 (7587), 496–501.
- Westermann, A.J., Barquist, L., Vogel, J., 2017. Resolving host-pathogen interactions by dual RNA-seq. *PLOS Pathogens* 13 (2), e1006033.
- White, E.A., Kramer, R.E., Tan, M.J., *et al.*, 2012a. Comprehensive analysis of host cellular interactions with human papillomavirus E6 proteins identifies new E6 binding partners and reflects viral diversity. *Journal of Virology* 86 (24), 13174–13186.
- White, E.A., Sowa, M.E., Tan, M.J., *et al.*, 2012b. Systematic identification of interactions between host cell proteins and E7 oncoproteins from diverse human papillomaviruses. *Proceedings of the National Academy of Sciences of the United States of America* 109 (5), E260–E267.
- Wuchty, S., 2011. Computational prediction of host-parasite protein interactions between *P. falciparum* and *H. sapiens*. *PLOS ONE* 6 (11), e26960.
- Xing, S., Wallmeroth, N., Berendzen, K.W., Grefen, C., 2016. Techniques for the analysis of protein-protein interactions in vivo. *Plant Physiology* 171 (2), 727–758.
- Yang, H., Ke, Y., Wang, J., *et al.*, 2011. Insight into bacterial virulence mechanisms against host immune response via the *Yersinia pestis*-human protein-protein interaction network. *Infection and Immunity* 79 (11), 4413–4424.
- Yang, J., Yan, R., Roy, A., *et al.*, 2015. The I-TASSER Suite: Protein structure and function prediction. *Nature Methods* 12 (1), 7–8.
- Yu, J., Guo, M., Needham, C.J., *et al.*, 2010. Simple sequence-based kernels do not predict protein-protein interactions. *Bioinformatics* 26 (20), 2610–2614.
- Zdobnov, E.M., Tegenfeldt, F., Kuznetsov, D., *et al.*, 2017. OrthoDB v9.1: Cataloging evolutionary and functional annotations for animal, fungal, plant, archaeal, bacterial and viral orthologs. *Nucleic Acids Research* 45 (D1), D744–D749.
- Zhang, Y., Skolnick, J., 2005. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research* 33 (7), 2302–2309.
- Zhou, H., Jin, J., Wong, L., 2013a. Progress in computational studies of host-pathogen interactions. *Journal of Bioinformatics and Computational Biology* 11 (2), 1230001.
- Zhou, H., Rezaei, J., Hugo, W., *et al.*, 2013b. Stringent DDI-based prediction of *H. sapiens*-*M. tuberculosis* H37Rv protein-protein interactions. *BMC Systems Biology* 7 (Suppl. 6), S6.
- Zhou, H., Gao, S., Nguyen, N.N., *et al.*, 2014. Stringent homology-based prediction of *H. sapiens*-*M. tuberculosis* H37Rv protein-protein interactions. *Biology Direct* 9, 5.
- Zoraghi, R., Reiner, N.E., 2013. Protein interaction networks as starting points to identify novel antimicrobial drug targets. *Current Opinion in Microbiology* 16 (5), 566–572.

Biographical Sketch



Dr. Fransiskus Xavierius Ivan is a Research Fellow in the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. He received Ph.D. in Computations and Systems Biology, Singapore-MIT Alliance (SMA) Programme from NTU in 2014, Master of Computing and Information Technology from the University of New South Wales (UNSW) in 2008, and Bachelor of Science in Mathematics from Bogor Agricultural University in 1999. His research interest is interdisciplinary life science, which is not limited to Bioinformatics, Genomic Data Science, Computational Systems Biology, and Computational Host-Pathogen Interactions. He has published in experimental and computational life science journals and conferences, including *Genomics*, *Functional Integrative Genomics*, *IEEE Engineering in Medicine and Biology Conference (EMBC)*, and *ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB)*. Dr. Ivan has actively collaborated with scientists from various backgrounds, including Life Science and Biomedical Science.



Dr. Jie Zheng is a tenure-track assistant professor of School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. He received PhD in 2006 from the University of California, Riverside and his B. Eng (honors) in 2000 from Zhejiang University in China, both in Computer Science. Before joining NTU in 2011, he was a research scientist at the National Center for Biotechnology Information (NCBI), National Library of Medicine (NLM), National Institutes of Health (NIH), USA. His research interests are Bioinformatics, Computational Systems Biology and Genomics, aiming to develop novel algorithms and *in silico* models to help answer Biomedical questions (e.g. how are cell fates decided in cancer and stem cells). He has published in top-tier journals such as Genome Biology, Molecular Biology & Evolution, Nucleic Acids Research, Bioinformatics, PLoS Computational Biology, etc. While trained as a Computer Scientist, Dr. Zheng maintains active and long-standing collaborations with Life Scientists.



Dr. Vincent T.K. Chow is a medical virologist and molecular biologist who graduated with MD, PhD, MBBS, MSc and FRCPath qualifications. Currently, he serves as Associate Professor and Education Director of Microbiology, and Principal Investigator of the Host and Pathogen Interactivity Laboratory at the Yong Loo Lin School of Medicine, National University of Singapore (NUS). Since 1996, he established the Human Genome Laboratory that has isolated and characterized several novel human genes and proteins. Dr. Chow previously served as President of the Asia-Pacific Society for Medical Virology, as well as Chair of the Virology Section of the International Society of Chemotherapy. His laboratory has published over 250 articles in international refereed journals, and has presented over 260 conference papers. He has received several awards (including the Murex Virologist Award, the Special Commendation Award and Faculty Research Excellence Award from NUS, the Singapore Society of Pathology – Becton Dickinson Award, and the Chan Yow Cheong Oration at the 6th Asia-Pacific Congress of Medical Virology). His recent research interests focus on the molecular genetics and infectious of influenza pneumonia and of hand, foot and mouth disease, specifically on the cellular, molecular, and viral pathogenesis of severe influenza and enterovirus 71 infections.



Dr. Kwoh Chee Keong is in the School of Computer and Science Engineering, Nanyang Technology since 1993. He received his Bachelor degree in Electrical engineering (1st Class) and Master in Industrial System Engineering from the National University of Singapore in 1987 and 1991 respectively. He received his PhD from the Imperial College, University of London in 1995. His research interests include Data Analytics, Data Mining, and Graph-Based Inference; applications areas include Bioinformatics and Biomedical Engineering. He has done significant research work the research areas of Bioinformatics and Computational Biology. He has been active as organizing member, referee and reviewer for a number of premier conferences and journals. Dr. Kwoh is a member of The Institution of Engineers Singapore, Association for Medical and Bio-Informatics, Imperial College Alumni Association of Singapore (ICAAS). He has provided many service to professional bodies, and the Singapore and was conferred the Public Service Medal, the President of Singapore in 2008 and the National Day Award Ministry of Education (Long Service Medal) in 2016.

Comparative Genomics Analysis

Hui San Ong, Perdana University, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Deoxyribonucleic acid (DNA) is a nucleic acid that encodes the genetic information required to carry out activities in all living organisms. A DNA molecule comprises of two strands, each run in the direction of 5' to 3' and are antiparallel. Both strands pair based on Watson-Crick complementarity; adenine (A) binds to thymine (T) while guanine (G) binds to cytosine (C). A genome can be defined as a complete collection of an organism's DNA, inclusive of the genes (the coding regions) and noncoding DNA. Genomics, on the other hand is an interdisciplinary field concerned with the study of the genes and intergenic regions as well as their structures, function and evolution.

The availability of genomics information from living organisms has led to the empowerment of comparative genomics analysis. The goal is to use related genomes en masse to achieve a better understanding of every individual genome in a dataset. It involves the comparison of more than one genome to uncover the similarities and differences between them. Comparison can be conducted for the entire genomes or the syntenic regions of various species or for various strains of the same species (Wei *et al.*, 2002). However, the scope of comparisons varies, as it can be carried out between the same species or across different kingdoms of living organisms. For example, the study of HSP90 chaperone family was carried out in Archaea, Bacteria, Protista, Plantae, Fungi and Animalia (Chen *et al.*, 2006). There are three distinct levels where the genome comparison can be carried out; in order to gain different viewpoint on an organism, (i) genomes structures at DNA and gene levels, and overall nucleotide statistics, (ii) coding regions and (iii) noncoding regions, including the prediction of the regulatory elements (Wei *et al.*, 2002; Loots *et al.*, 2002). In contrast to the former two, studies on noncoding regions are rather at infancy and much work needs to be done. It is only until late 20th century that we started to understand that regulatory sequences in noncoding regions play essential roles in the post-transcriptional processes (Wedemeyer *et al.*, 2000; Mazumder *et al.*, 2003).

Comparative genomics can be dated back to 1984 through the efforts led by Professor Wimmer to compare different virus genomes. The comparison was made between several animal picornaviruses and cowpea mosaic virus, a plant virus that causes the "mosaic" pattern on plant leaves (Argos *et al.*, 1984). Two years later, a large-scale study was carried out between the varicella-zoster virus that causes chickenpox in humans and the Epstein-Barr virus that is associated with diseases such as gastric cancer, nasopharyngeal carcinoma and Hodgkin's lymphoma (McGeoch and Davison, 1986). Ever since then, this approach has been employed extensively and its application encompasses various fields such as agriculture (Sharma *et al.*, 2014; Sorrells, 2006; Wang *et al.*, 2017), evolutionary study and disease study (Alfoldi and Lindblad-Toh, 2013).

The main objective of this article is to provide the readers an overview of how bioinformatics is applied to carry out comparative genomics, with a focus on agriculture and biomedical research. It begins with a brief historical background and provides the current status on some notable genome sequencing projects. This is not meant to serve as an exhaustive dossier on comparative genomics, but rather to serve as a guide on the topic from the application perspective.

Background

The Past and Present: From Sequence to Genomes

The earliest bacterial genomes *Haemophilus influenza* and *Mycoplasma genitalium* (Fleischmann *et al.*, 1995; Fraser *et al.*, 1995) were fully sequenced in 1995 and 8 years later, the first human genome sequencing project was concluded (International Human Genome Sequencing, 2004). Since then, several teams have taken on the challenge of initiating ambitious genome sequencing projects, such as the 100,000 Genomes Project by Genomics England to sequence genomes of 100,000 patients with rare disease, registered with the National Health Service (NHS) (England, 2016); another is the GenomeAsia 100 K project to sequence 100,000 individuals in the Asian population (see Relevant Website section).

The rapid surge in the number of sequencing projects has led to the generation of massive amount of data, evidenced by the increment of sequencing data depositions in the Sequence Read Archive (SRA) (Leinonen *et al.*, 2011) at National Center for Biotechnology Information (Fig. 1). Many other public databases that cater for the specific projects based on the organism being sequenced have also been developed, for instance the Ensembl Genomes (Hubbard *et al.*, 2002), Mouse Genome Database (MGD) (Bult *et al.*, 2008), Maize Genetics Database (MaizeGDB) (Lawrence *et al.*, 2004), UCSC Genome Browser Database (Karolchik *et al.*, 2003) and other specific human disease-related databases, such as The Cancer Genome Atlas (Cancer Genome Atlas Research, 2008) and International Cancer Genome Consortium (Zhang *et al.*, 2011).

These efforts exemplify how the sequencing projects have progressed at a rapid speed, largely owing to the vast reduction in the sequencing cost, which stands at USD 0.012 per megabase as of 31st July 2017 (Wetterstrand, 2017). In the last five years (from 1st August 2013 to 1st August 2017), we have seen publications with the word "genome sequencing" (in the title/abstract) recorded in the PubMed database increase by two folds; with a latest a total of 13,314 articles published. During the

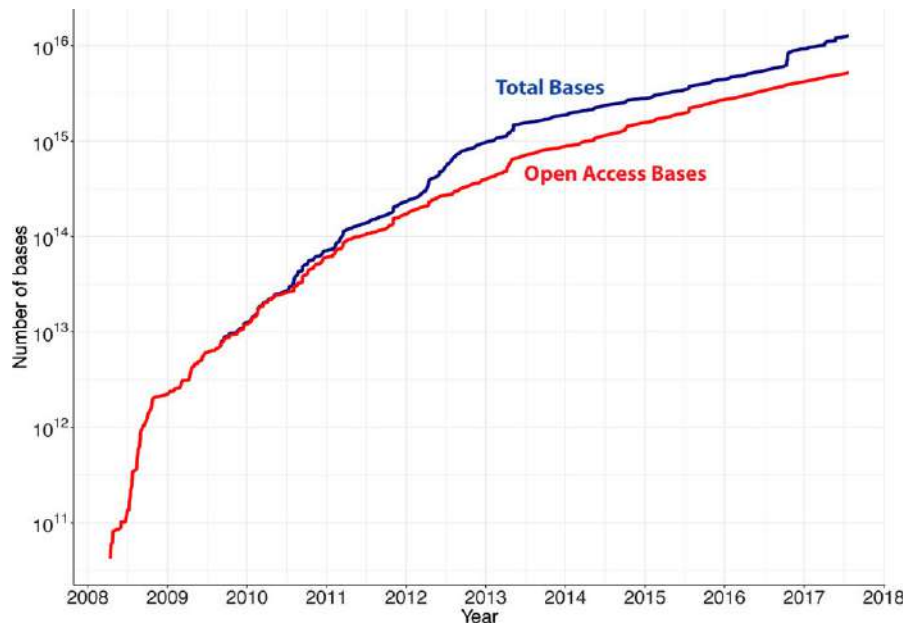


Fig. 1 The graph shows the growth in the Sequence Read Archive (SRA) database that stores raw sequencing data from high-throughput sequencing platforms from year 2008 to the present. The data used to plot this graph were extracted from Sequence Read Archive at <https://www.ncbi.nlm.nih.gov/Traces/sra>. The graph was built using R version 3.4.0. Reproduced from Leinonen, R., Sugawara, H., Shumwayand, M., International Nucleotide Sequence Database Collaboration, 2011. The sequence read archive. *Nucleic Acids Res* 39, D19–D21.

same time period, publications with the phrase “comparative genomics” in the title or abstract surged from 3546 articles to 5560 articles, an increment of approximately 57% from the year 2013. These figures indicate that one of the driving forces behind comparative genomics is the growth in the number of genome sequencing projects that provide the required genomics data.

Methods in Comparative Genomics

The process in comparative genomics frequently involves the alignment of DNA sequences; such as the use of pairwise alignment (e.g. Needle and Water [Needleman and Wunsch, 1970](#); [Smith and Waterman, 1981](#)) to detect closely related genes. An extension of this sequence alignment is multiple sequence alignment (MSA) such as Clustal Omega ([Sievers et al., 2011](#)) and T-coffee ([Notredame et al., 2000](#)) where it can be used to detect conserved regions through the alignment of three or more sequences. Housekeeping genes are examples of regions in a genome that tend to be highly conserved and evolve slower than other genes such as the tissue-specific genes ([Zhang and Li, 2004](#)), mainly due to their roles in the maintenance of basic cellular functions and are essential for the existence of a cell.

Homology is another crucial term that is frequently integrated into genomics study. The detection of homologous genes is of fundamental importance in many fields of biology, particularly in comparative genomics. Homology refers to a relationship between genes due to a common ancestry and it can be applied in two major contexts, depending on whether the genes were separated through a speciation (ortholog) or duplication (paralog) event. Orthologous genes are usually detected using sequence similarity, and a gene is considered as ortholog of each other if they have the highest percentage of sequence similarity compared to others genes. Reciprocal best blast hits ([Ward and Moreno-Hagelsieb, 2014](#); [Horiike et al., 2016](#)) and domain based detection ([Chen et al., 2010](#)) are among the standard protocols used to detect the orthologous genes. Unlike orthologous genes which preserve the same function throughout evolution, paralogous genes are the new genes that diverged after duplication events and later evolved into a new function. The procedures used in comparative genomics vary depending on the application, and the general workflow on the commonly used procedures is detailed in [Fig. 2](#).

Applications of Comparative Genomics

Prokaryotes

Mycobacterium ulcerans causes Buruli ulcer, an infectious tropical disease that causes skin ulcers. Although much effort has been invested, the slow growing nature of *M. ulcerans* in experimental conditions has hampered the progress in conventional drug discovery methods. To address this problem, the comparative genomics technique has been deployed by [Butt et al. \(2012a\)](#) to

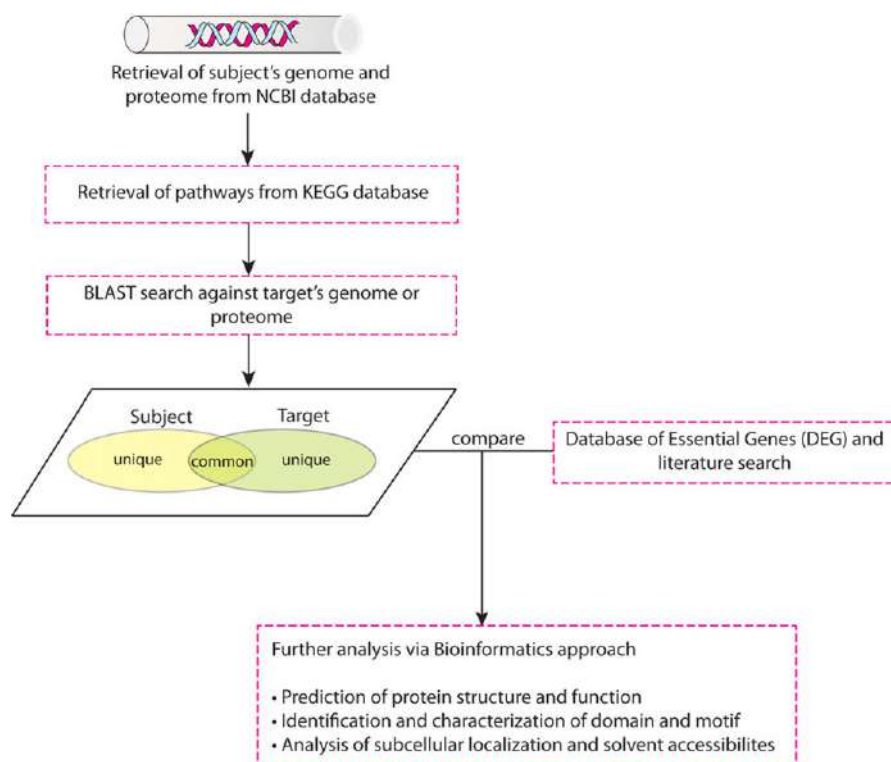


Fig. 2 General workflow of comparative genomics. It starts with the subject's genome and proteome as input and followed by the information retrieval from KEGG database. The BLAST search (e.g. BLASTP) against the target's proteome will reveal the existence of unique or common (shared) proteins between the subject and the target. Further bioinformatics analysis is often conducted on the proteins of interest.

identify novel therapeutic candidates against *M. ulcerans*, through homology searching of essential targets from twenty different bacteria that have no homology in human host. The group successfully discovered a total of 236 proteins based on these criteria as the potential drug and vaccine candidates against *M. ulcerans*. The comparison was done using a series of computational workflows which includes the homology blast search and analysis of host and pathogenic metabolic pathways (Butt *et al.*, 2012a).

The study carried out on *Salmonella Typhimurium* LT2 and *Mycoplasma genitalium* were based on the similar approach to *M. ulcerans*, that is focusing on the identification of essential targets in the bacterial pathogens but have no homology in the human host. These studies are coupled with metabolic pathway analysis in order to pinpoint the unique biological pathways that are present in *S. Typhimurium* LT2 (Samal *et al.*, 2015) and *M. genitalium* as compared to the *Homo sapiens* (Butt *et al.*, 2012b). The workflows, which include the drug targets prioritization and homology modelling have successfully identified 13 proteins as vaccine candidates in *S. Typhimurium* LT2 (Samal *et al.*, 2015). *M. genitalium* is another human parasitic pathogen that is affiliated with sexually transmitted diseases. Hence, *in silico* comparative genomics is being employed because conventional drug discovery process is costly and bear few drug targets. This comparison against human host has identified 67 non-homologous essential proteins of *M. genitalium*, which could serve as potential drug design and vaccine targets against *M. genitalium* (Butt *et al.*, 2012b).

Concretely, comparative methods are employed to pinpoint genes of essential function amongst the pathogenic bacteria, even more so when there is an increasing emergence of reported antibiotic-resistant pathogenic microorganisms (Choudhury *et al.*, 2012; Martinez, 2008). In view of this, Glass *et al.* (2002) compiled a list of essential attributes of a good drug targets in the bacteria genomes, which includes the target (i) being essential for the disease, (ii) unique to the bacteria and significantly different from its corresponding orthologous genes in human and (iii) present in key pathogenic bacteria (Glass *et al.*, 2002). As proposed by the former examples, target selection is mainly done using computational approaches, for example, comparison of pathways, sequence and structural homology as well as cluster and motif analysis (Gerstein, 2000; Rosamond and Allsop, 2000; Gerstein and Jansen, 2000).

Genome comparison can also be made with the aim to identify species-specific features or single-copy (unique) genes of an organisms. In the context of comparative genomics, some of the unique genes can be referred as the genes with no homolog, even with its closely related species (Llorente *et al.*, 2000; Ong *et al.*, 2012). These genes can be used to provide insights into the niche adaptation of the specific phenotypes (Kahlke *et al.*, 2012) and virulence factors. *Helicobacter pylori*, a gram negative bacteria that colonizes gastric epithelium and causes a persistent inflammatory response (Ruggiero, 2010) is an example of bacteria that possess such unique feature in their genetic makeup. Given its distinctive role in pathogenicity, genome comparison was carried out between *Helicobacter pylori* and its two pathogenic counterparts; *Haemophilus influenza* and *Escherichia coli*. The work has successfully contributed in the identification of species-specific features such as acid tolerance and outer membrane family proteins, which are

crucial for the bacteria's colonization in acidic environment (Huynen *et al.*, 1998; Matsuo *et al.*, 2017). These outer membrane proteins were used by *H. pylori* to attach to the host's gastric epithelial cells (Matsuo *et al.*, 2017).

In another similar approach, cross-species comparison was conducted by comparing *Vibrio harveyi* CAIM 1792 with its related genomes to resolve their phylogeny relation as well as to discover unique regions that served to differentiate these strains (Espinoza-Valles *et al.*, 2015). These unique traits include the regions that encode for transcriptional regulator, LysR, which was described to be associated with pathogenic activity, quorum sensing and motility (Maddocks and Oyston, 2008). *V. harveyi* CAIM 1792 is of interest because this marine bacterial strain causes mortality in shrimp (Soto-Rodriguez *et al.*, 2010; Moriarty, 1998), and therefore the discovery of virulence genes in this strain is crucial.

Agricultural Animals and Plants

Agriculture is one of the most important sectors that provides food supply, as well as industrial raw materials. Farm animals, for example, have been bred by targeting on their desirable traits, such as disease resistance and their rapid growth. The general location on a chromosome for a gene that expresses a particular trait is known as quantitative trait locus (QTL). Dairy cattle for instance, has at least 30 QTLs that were likely to be involved in the milk production traits (Chamberlain *et al.*, 2007). An animal or plant species with favourable QTLs will have strong advantage over the other species; and this serves as the basis for the urgency of comparative genomics study to be carried out. This is mainly to increase the production of the agricultural commodities. The aim is to discover genes that are related to disease resistance, breed-specific quantitative loci along with phenotypes of agricultural relevance, known as economic trait loci (ETL) (Womack and Kata, 1995; Womack, 1998). The identifications of ETLs in animal pedigrees are crucial given that they are important for the improvement and production of livestock.

Turkey and chicken are important poultry; they are raised mainly for their meat and eggs production and thus have significant contribution in agricultural economy. Due to their importance in daily food consumption, numerous scientific efforts (Reed *et al.*, 2005; Chaves *et al.*, 2006; Reed *et al.*, 2006; Wallis *et al.*, 2004) have been concerted on aspects that would be beneficial to the industry. Comparative works for instance have successfully established the cytogenetic map between the turkey and chicken that allowed for the transfer of genetic data from chicken to turkey and prediction of novel loci for target marker development (Griffin *et al.*, 2008). Besides chicken-turkey comparison, the comparison is also being extended to other genomes, including human genome (International Chicken Genome Sequencing, 2004).

Identification of orthologous genes is one of the common procedures in comparative analysis. Once the orthologous genes have been defined, the subsequent step is the identification of the conserved chromosomal regions that will lead to the generation of comparative map (Burt, 2002). If two genes are located on the same chromosome, the two are known as syntenic. The construction of cattle-human comparative map (Larkin *et al.*, 2003) assists in the adding of the cattle genome to the existing multispecies comparative genome studies, which is also inclusive of goat (Schibler *et al.*, 1998), pig (Pinton *et al.*, 2000; Ma *et al.*, 2011), and horse (Caetano *et al.*, 1999).

Besides agriculture livestock, identification of quantitative trait locus (QTLs) is also important in increasing crops yield, its tolerance against environmental stress and disease resistance, which would involve the construction of genetic maps of the crop species. To exemplify this, Hwang *et al.* (2006) successfully formulated a fine genetic map near Rsv4 gene in soybean by comparing against the genome of closely related model, legume *Lotus japonicas* (Hwang *et al.*, 2006). Rsv4 gene has profound economic value due its ability to confer resistance against all seven Soybean mosaic virus (SMV) strains (Ma *et al.*, 1995). Soybean [*Glycine max* L. (Merrill)] is one of the major sources of edible oil and proteins and it is under severe threat from SMV that could affect its production worldwide (Hill and Whitham, 2014). Following the formulation of this genetic map near Rsv4 gene, the researchers are able to derive a new biomarker, AW307114A from the soybean expressed sequence tags (ESTs). This newly discovered marker is important because it is more closely linked to Rsv4 resistance gene in the soybeans (Hwang *et al.*, 2006) than the previously discovered biomarkers.

Bacterial wilt (BW) is yet another plant disease caused by *Ralstonia solanacearum* that affects pepper, tomato and eggplant (Hayward, 1991). To overcome this, researchers resequenced the plants that harbour resistance traits towards BW, in this case the pepper plant YCM344 (Kang *et al.*, 2016). Using the comparative analysis against the pepper reference genome sequence (version 1.55), they have successfully identified single nucleotide polymorphisms (SNP) that have the potential to cause BW resistance in pepper plant. Not only that, they could also pinpoint genomic regions of these SNP to be homologous to the known resistance genes in the tomato genomes, exemplified yet again by the use of comparative analysis to identify conserved regions across different genomes.

Jute (*Corchorus* sp.) is one of the pivotal sources of natural fibre, with the advantages such as high tensile strength, moderate heat resistance and biodegradable. Given that jute production has profound economic value, genomics comparison was carried out on two species of jute (*Corchorus olitorius* and *Corchorus capsularis*) against 13 others genomes including *Arabidopsis thaliana* and *Glycine max* (Islam *et al.*, 2017). Subsequent identification of homologous genes has led the team to propose a model for bast fibre biogenesis in jute (Islam *et al.*, 2017). The understanding of molecular basis of the fibre biogenesis enabled improvement to the breeding strategies of jute to meet the increasing demand of natural fibres in industry, as the annual global production of jute is up to 2.821 million tonnes for year 2014/15 (Food and Agriculture Organization of The United Nations, 2016).

Rice (*Oryza sativa*) is the staple food in Asia and the productions of rice are among the highest in Asian populations. A number of comparative genomics studies have been conducted on rice. For instance, the comparison between rice and *Arabidopsis* genomes using the phylogenetic comparisons reveals that certain families that are involved in the auxin homeostasis and plant

development in Arabidopsis are missing from the rice genome (Nelson *et al.*, 2004). Another comparison of structural variation was conducted between rice and its closest relative in the genus *Oryza* (Hurwitz *et al.*, 2010). Together, this has resulted in the detailed catalog of structural variation across the *Oryza* family that can be used for forthcoming works in genome evolution, speciation, domestication and novel gene discovery (Hurwitz *et al.*, 2010).

Comparative genomics has showcased significant improvement in the plants yield production. This is of much dire urgency, as better yield of crops production would ensure better management of the famine outspread worldwide, with estimates about 795 million people from the 7.3 billion people in the world were affected from persistent undernourishment (Food and Agriculture Organization *et al.*, 2015)

Animal Models for Human Disease

From historical perspective, biologists remarkably benefited from the study of animals. Animal models (Perlman, 2016; Jiminez *et al.*, 2015; Schofield *et al.*, 2011; Iannaccone and Jacob, 2009; Shin and Fishman, 2002) are often used to get a better understanding on the diseases that are associated with human. For instance, nematode (*Caenorhabditis elegans*) is used as a model for studies such as in cancer research (Kyriakakis *et al.*, 2015) and in host-microbiome interactions at the apical surface of epithelial cells in the intestine (Pukkila-Worley and Ausubel, 2012). This interaction study is important because the immune defense system must be able to distinguish between pathogens and the harmless bacteria that are present in the mammalian intestine, in order to combat infection.

The use of human disease models in understanding the fundamental pathophysiology of a disease and the breakthrough of new therapeutic, signifies the need for comparative studies. For instance, the study of mouse genome involved the identification of QTLs for blood pressure that are known to contribute to hypertension (Wright *et al.*, 1999). This breakthrough has served as a precursor for the development of novel comparative genomics map for “candidate hypertension loci” in human based on the QTLs translation between rat model and human (Stoll *et al.*, 2000). It has then resulted in the prognostication of 26 chromosomal regions in human that are plausible to harbour hypertension genes (Stoll *et al.*, 2000), and thus can potentially be used for the development of new therapeutics.

Although rat and mouse remain the leading models to decipher the genetic basis of human diseases, zebrafish is yet another excellent model for human disease. This is mainly attributed to its optical clarity during embryogenesis, high productivity of large clutches of embryos and short generations time (Detrich *et al.*, 1999). On top of that, due to high genomic content similarity shared between zebrafish and human (approximately 70% the genes are orthologues) (Howe *et al.*, 2013), it offers huge opportunities for comparative analysis between these two genomes. Some of the applications of zebrafish as a model for better understanding of the human diseases would be the study of cardiovascular disease (Sehnert and Stainier, 2002; North and Zon, 2003), muscle disease (Guyon *et al.*, 2007; Kunkel *et al.*, 2006) and aging process (Keller and Murtha, 2004).

Animal models also served as valuable tools to study the biology and genetic aspects of human cancers, as well as for preclinical study of anti-cancer therapeutics (LeGendre-McGhee *et al.*, 2015; Steele *et al.*, 2005; Ruggeri *et al.*, 2014). Therefore, it is crucial to have animal data to be evaluated in the context of human cancers. Some initial cancer studies leverage on comparative genomics to identify orthologs of cancer genes (Pickeral *et al.*, 2000; Makalowski *et al.*, 1996), such as for the study of Wnt signalling pathway (Katoh and Katoh, 2005a,b). While other studies carry on with further analysis to define the syntenic region between the rat and human genomes and this helps to uncover the chromosomal regions for tumor and metastasis in the rat that are involved in human prostate cancer (Datta *et al.*, 2005).

In short, although human and animal models may have different physical appearance, but both share a certain degree of similarities in their genomes. It is these “similarities” that could give important clues on the development of the diseases in human and afford us an opportunity to treat these diseases more effectively. In fact, researchers are now able to find better treatment of human disease by recreating the human diseases in animal models, ensuring a better quality of life ever since for human.

Closing Remarks

The fundamental basis of comparative genomics is to identify the genes of interest that control traits variation and this is made possible through the ability to compare and contrast between one organism to another organism. Aided with the integration of genomics analysis, this translation is an exciting notion in gaining the momentum for the breakthrough in gene discovery for all domains of life, either for single-celled or multicellular organisms. The resulting genomics information is able to provide insights into the biology of an individual genome, benefitting the advances in human disease related study and greater yield improvement in agricultural animals, plant and crops.

See also: Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome Analysis – Identification of Genes Involved in Host-Pathogen Protein-Protein Interaction Networks. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Sequence Analysis. Whole Genome Sequencing Analysis

References

- Alfoldi, J., Lindblad-Toh, K., 2013. Comparative genomics as a tool to understand evolution and disease. *Genome Res* 23, 1063–1068.
- Argos, P., Kamer, G., Nicklin, M.J., Wimmer, E., 1984. Similarity in gene organization and homology between proteins of animal picornaviruses and a plant comovirus suggest common ancestry of these virus families. *Nucleic Acids Res* 12, 7251–7267.
- Bult, C.J., Eppig, J.T., Kadin, J.A., *et al.*, 2008. The Mouse Genome Database (MGD): Mouse biology and model systems. *Nucleic Acids Res* 36, D724–D728.
- Burt, D.W., 2002. Comparative mapping in farm animals. *Brief Funct Genomic Proteomic* 1, 159–168.
- Butt, A.M., Nasrullah, I., Tahir, S., Tong, Y., 2012a. Comparative genomics analysis of *Mycobacterium ulcerans* for the identification of putative essential genes and therapeutic candidates. *PLOS ONE* 7, e43080.
- Butt, A.M., Tahir, S., Nasrullah, I., *et al.*, 2012b. *Mycoplasma genitalium*: A comparative genomics study of metabolic pathways for the identification of drug and vaccine targets. *Infect Genet Evol* 12, 53–62.
- Caetano, A.R., Shiue, Y.L., Lyons, L.A., *et al.*, 1999. A comparative gene map of the horse (*Equus caballus*). *Genome Res* 9, 1239–1249.
- Chamberlain, A.J., McPartlan, H.C., Goddard, M.E., 2007. The number of loci that affect milk production traits in dairy cattle. *Genetics* 177, 1117–1123.
- Chaves, L.D., Knutson, T.P., Krueth, S.B., Reed, K.M., 2006. Using the chicken genome sequence in the development and mapping of genetic markers in the turkey (*Meleagris gallopavo*). *Anim Genet* 37, 130–138.
- Chen, B., Zhong, D., Monteiro, A., 2006. Comparative genomics and evolution of the HSP90 family of genes across all kingdoms of organisms. *BMC Genom* 7, 156.
- Chen, T.W., Wu, T.H., Ng, W.V., Lin, W.C., 2010. DODO: An efficient orthologous genes assignment tool based on domain architectures. Domain based ortholog detection. *BMC Bioinform* 11 (Suppl. 7), S6.
- Food and Agriculture Organization, International Fund for Agricultural Development, World Food Programme, International Fund for Agricultural Development & Program., W. F., 2015. The State of Food Insecurity in the World 2015. Strengthening the enabling environment for food security and nutrition. Rome: FAO.
- Cancer Genome Atlas Research Network, N., 2008. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* 455, 1061–1068.
- Choudhury, R., Panda, S., Singh, D.V., 2012. Emergence and dissemination of antibiotic resistance: A global problem. *Indian J Med Microbiol* 30, 384–390.
- Datta, M.W., Suckow, M.A., Twigger, S., *et al.*, 2005. Using comparative genomics to leverage animal models in the identification of cancer genes. Examples in prostate cancer. *Cancer Genom Proteom* 2, 137–144.
- International Chicken Genome Sequencing Consortium, C., 2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature* 432, 695–716.
- International Human Genome Sequencing Consortium, C., 2004. Finishing the euchromatic sequence of the human genome. *Nature* 431, 931–945.
- Detrich 3rd, H.W., Westerfield, M., Zon, L.I., 1999. Overview of the zebrafish system. *Methods Cell Biol* 59, 3–10.
- Food and Agriculture Organization of The United Nations, 2016. Jute, kenaf, sisal, abaca coir and allied fibres. Rome: FAO.
- Genomics England, 2016. The 100,000 Genomes Project.
- Espinoza-Valles, I., Vora, G.J., Lin, B., *et al.*, 2015. Unique and conserved genome regions in *Vibrio harveyi* and related species in comparison with the shrimp pathogen *Vibrio harveyi* CAIM 1792. *Microbiology* 161, 1762–1779.
- Fleischmann, R.D., Adams, M.D., White, O., *et al.*, 1995. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* 269, 496–512.
- Fraser, C.M., Gocayne, J.D., White, O., *et al.*, 1995. The minimal gene complement of *Mycoplasma genitalium*. *Science* 270, 397–403.
- Gerstein, M., 2000. Integrative database analysis in structural genomics. *Nat Struct Biol* 7 (Suppl), 960–963.
- Gerstein, M., Jansen, R., 2000. The current excitement in bioinformatics-analysis of whole-genome expression data: How does it relate to protein structure and function? *Curr Opin Struct Biol* 10, 574–584.
- Glass, J.I., Belanger, A.E., Robertson, G.T., 2002. *Streptococcus pneumoniae* as a genomics platform for broad-spectrum antibiotic discovery. *Curr Opin Microbiol* 5, 338–342.
- Griffin, D.K., Robertson, L.B., Tempest, H.G., *et al.*, 2008. Whole genome comparative studies between chicken and turkey and their implications for avian genome evolution. *BMC Genomics* 9, 168.
- Guyon, J.R., Steffen, L.S., Howell, M.H., *et al.*, 2007. Modeling human muscle disease in zebrafish. *Biochim Biophys Acta* 1772, 205–215.
- Hayward, A.C., 1991. Biology and epidemiology of bacterial wilt caused by *Pseudomonas solanacearum*. *Annu Rev Phytopathol* 29, 65–87.
- Hill, J.H., Whitham, S.A., 2014. Control of virus diseases in soybeans. *Adv Virus Res* 90, 355–390.
- Horiike, T., Minai, R., Miyata, D., Nakamura, Y., Tateno, Y., 2016. Ortholog-Finder: A tool for constructing an ortholog data set. *Genome Biol Evol* 8, 446–457.
- Howe, K., Clark, M.D., Torroja, C.F., *et al.*, 2013. The zebrafish reference genome sequence and its relationship to the human genome. *Nature* 496, 498–503.
- Hubbard, T., Barker, D., Birney, E., *et al.*, 2002. The Ensembl genome database project. *Nucleic Acids Res* 30, 38–41.
- Hurwitz, B.L., Kudrna, D., Yu, Y., *et al.*, 2010. Rice structural variation: A comparative analysis of structural variation between rice and three of its closest relatives in the genus *Oryza*. *Plant J* 63, 990–1003.
- Huynen, M., Dandekar, T., Bork, P., 1998. Differential genome analysis applied to the species-specific features of *Helicobacter pylori*. *FEBS Lett* 426, 1–5.
- Hwang, T.Y., Moon, J.K., Yu, S., *et al.*, 2006. Application of comparative genomics in developing molecular markers tightly linked to the virus resistance gene Rsv4 in soybean. *Genome* 49, 380–388.
- Iannaccone, P.M., Jacob, H.J., 2009. Rats! *Dis Model Mech* 2, 206–210.
- Islam, M.S., Saito, J.A., Emdad, E.M., *et al.*, 2017. Comparative genomics of two jute species and insight into fibre biogenesis. *Nat Plants* 3, 16223.
- Jimenez, J.A., Uwiera, T.C., Douglas Inglis, G., Uwiera, R.R., 2015. Animal models to study acute and chronic intestinal inflammation in mammals. *Gut Pathog* 7, 29.
- Kahlke, T., Goesmann, A., Hjerde, E., Willassen, N.P., Haugen, P., 2012. Unique core genomes of the bacterial family vibronaceae: Insights into niche adaptation and speciation. *BMC Genomics* 13, 179.
- Kang, Y.J., Ahn, Y.K., Kim, K.T., Jun, T.H., 2016. Resequencing of *Capsicum annuum* parental lines (YCM334 and Taeon) for the genetic analysis of bacterial wilt resistance. *BMC Plant Biol* 16, 235.
- Karolchik, D., Baertsch, R., Diekhans, M., *et al.*, 2003. The UCSC Genome Browser Database. *Nucleic Acids Res* 31, 51–54.
- Katoh, M., Katoh, M., 2005a. Comparative genomics on Wnt7a orthologs. *Oncol Rep* 13, 777–780.
- Katoh, M., Katoh, M., 2005b. Comparative genomics on Wnt8a and Wnt8b genes. *Int J Oncol* 26, 1129–1133.
- Keller, E.T., Murtha, J.M., 2004. The use of mature zebrafish (*Danio rerio*) as a model for human aging and disease. *Comp Biochem Physiol C Toxicol Pharmacol* 138, 335–341.
- Kunkel, L.M., Bachrach, E., Bennett, R.R., Guyon, J., Steffen, L., 2006. Diagnosis and cell-based therapy for Duchenne muscular dystrophy in humans, mice, and zebrafish. *J Hum Genet* 51, 397–406.
- Kyriakakis, E., Markaki, M., Tavemarakis, N., 2015. *Caenorhabditis elegans* as a model for cancer research. *Mol Cell Oncol* 2, e975027.
- Larkin, D.M., Everts-van der Wind, A., Rebeiz, M., *et al.*, 2003. A cattle-human comparative map built with cattle BAC-ends and human genome sequence. *Genome Res* 13, 1966–1972.
- Lawrence, C.J., Dong, Q., Polacco, M.L., Seigfried, T.E., Brendel, V., 2004. MaizeGDB, the community database for maize genetics and genomics. *Nucleic Acids Res* 32, D393–D397.
- LeGendre-McGhee, S., Rice, P.S., Wall, R.A., *et al.*, 2015. Time-serial assessment of drug combination interventions in a mouse model of colorectal carcinogenesis using optical coherence tomography. *Cancer Growth Metastasis* 8, 63–80.

- Leinonen, R., Sugawara, H., Shumway, M., International Nucleotide Sequence Database Collaboration, C., 2011. The sequence read archive. *Nucleic Acids Res* 39, D19–D21.
- Liorente, B., Durrens, P., Malpertuy, A., *et al.*, 2000. Genomic exploration of the hemiascomycetous yeasts: 20. Evolution of gene redundancy compared to *Saccharomyces cerevisiae*. *FEBS Lett* 487, 122–133.
- Loots, G.G., Ovcharenko, I., Pachter, L., Dubchak, I., Rubin, E.M., 2002. rVista for comparative sequence-based discovery of functional transcription factor binding sites. *Genome Res* 12, 832–839.
- Ma, G., Chen, P., Buss, G.R., Tolín, S.A., 1995. Genetic characteristics of two genes for resistance to soybean mosaic virus in PI486355 soybean. *Theor Appl Genet* 91, 907–914.
- Ma, J.G., Chang, T.C., Yasue, H., *et al.*, 2011. A high-resolution comparative map of porcine chromosome 4 (SSC4). *Anim Genet* 42, 440–444.
- Maddocks, S.E., Oyston, P.C., 2008. Structure and function of the LysR-type transcriptional regulator (LTTR) family proteins. *Microbiology* 154, 3609–3623.
- Makalowski, W., Zhang, J., Boguski, M.S., 1996. Comparative analysis of 1196 orthologous mouse and human full-length mRNA and protein sequences. *Genome Res* 6, 846–857.
- Martínez, J.L., 2008. Antibiotics and antibiotic resistance genes in natural environments. *Science* 321, 365–367.
- Matsuo, Y., Kido, Y., Yamaoka, Y., 2017. *Helicobacter pylori* outer membrane protein-related pathogenesis. *Toxins (Basel)* 9.
- Mazumder, B., Seshadri, V., Fox, P.L., 2003. Translational control by the 3'-UTR: The ends specify the means. *Trends Biochem Sci* 28, 91–98.
- McGeoch, D.J., Davison, A.J., 1986. DNA sequence of the herpes simplex virus type 1 gene encoding glycoprotein gH, and identification of homologues in the genomes of varicella-zoster virus and Epstein-Barr virus. *Nucleic Acids Res* 14, 4281–4292.
- Moriarty, D., 1998. Microbial ecology: Protecting the prawns in polluted ponds. *Microbiol. Aust* 19, 22–27.
- Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 48, 443–453.
- Nelson, D.R., Schuler, M.A., Paquette, S.M., Werck-Reichhart, D., Bak, S., 2004. Comparative genomics of rice and Arabidopsis. Analysis of 727 cytochrome P450 genes and pseudogenes from a monocot and a dicot. *Plant Physiol* 135, 756–772.
- North, T.E., Zon, L.I., 2003. Modeling human hematopoietic and cardiovascular diseases in zebrafish. *Dev Dyn* 228, 568–583.
- Notredame, C., Higgins, D.G., Heringa, J., 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol* 302, 205–217.
- Ong, H.S., Mohamed, R., Firdaus-Raii, M., 2012. Comparative genome sequence analysis reveals the extent of diversity and conservation for glycan-associated proteins in *Burkholderia* spp. *Comp Funct Genomics* 2012, 752867.
- Perlman, R.L., 2016. Mouse models of human disease: An evolutionary perspective. *Evol Med Public Health* 2016, 170–176.
- Pickeral, O.K., Li, J.Z., Barrow, I., *et al.*, 2000. Classical oncogenes and tumor suppressor genes: A comparative genomics perspective. *Neoplasia* 2, 280–286.
- Pinton, P., Schibler, L., Cribiu, E., Gellin, J., Yerle, M., 2000. Localization of 113 anchor loci in pigs: Improvement of the comparative map for humans, pigs, and goats. *Mamm Genome* 11, 306–315.
- Pukkila-Worley, R., Ausubel, F.M., 2012. Immune defense mechanisms in the *Caenorhabditis elegans* intestinal epithelium. *Curr Opin Immunol* 24, 3–9.
- Reed, K.M., Hall, M.K., Chaves, L.D., Knutson, T.P., 2006. Single nucleotide polymorphisms for integrative mapping in the Turkey (*Meleagris gallopavo*). *Anim Biotechnol* 17, 73–80.
- Reed, K.M., Chaves, L.D., Hall, M.K., Knutson, T.P., Harry, D.E., 2005. A comparative genetic map of the turkey genome. *Cytogenet Genome Res* 111, 118–127.
- Rosamond, J., Allsop, A., 2000. Harnessing the power of the genome in the search for new antibiotics. *Science* 287, 1973–1976.
- Ruggeri, B.A., Camp, F., Miknyoczki, S., 2014. Animal models of disease: Pre-clinical animal models of cancer and their applications and utility in drug discovery. *Biochem Pharmacol* 87, 150–161.
- Ruggiero, P., 2010. *Helicobacter pylori* and inflammation. *Curr Pharm Des* 16, 4225–4236.
- Samal, H.B., Prava, J., Suar, M., Mahapatra, R.K., 2015. Comparative genomics study of *Salmonella typhimurium* LT2 for the identification of putative therapeutic candidates. *J Theor Biol* 369, 67–79.
- Schibler, L., Vaiman, D., Oustry, A., Giraud-Delville, C., Cribiu, E.P., 1998. Comparative gene mapping: A fine-scale survey of chromosome rearrangements between ruminants and humans. *Genome Res* 8, 901–915.
- Schofield, P.N., Sundberg, J.P., Hoehndorf, R., Gkoutos, G.V., 2011. New approaches to the representation and analysis of phenotype knowledge in human diseases and their animal models. *Brief Funct Genomics* 10, 258–265.
- Sehnert, A.J., Stainier, D.Y., 2002. A window to the heart: Can zebrafish mutants help us understand heart disease in humans? *Trends Genet* 18, 491–494.
- Sharma, A., Li, X., Lim, Y.P., 2014. Comparative genomics of Brassicaceae crops. *Breed Sci* 64, 3–13.
- Shin, J.T., Fishman, M.C., 2002. From zebrafish to human: Modular medical models. *Annu Rev Genomics Hum Genet* 3, 311–340.
- Sievers, F., Wilm, A., Dineen, D., *et al.*, 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol Syst Biol* 7, 539.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *J Mol Biol* 147, 195–197.
- Sorrells, M.E., 2006. Applications of Comparative Genomics to Crop Improvement. Ames, IA: Blackwell Publishing.
- Soto-Rodríguez, S.A., Gómez-Gil, B., Lozano, R., 2010. 'Bright-red' syndrome in Pacific white shrimp *Litopenaeus vannamei* is caused by *Vibrio harveyi*. *Dis Aquat Organ* 92, 11–19.
- Steele, V.E., Lubet, R.A., Moon, R.C., 2005. Preclinical animal models for the development of cancer chemoprevention drugs. In: Kelloff, G.J., Hawk, E.T., Sigman, C.C. (Eds.), *Cancer Chemoprevention: Volume 2: Strategies for Cancer Chemoprevention*. Totowa, NJ: Humana Press.
- Stoll, M., Kwitek-Black, A.E., Cowley Jr., A.W., *et al.*, 2000. New target regions for human hypertension via comparative genomics. *Genome Res* 10, 473–482.
- Wallis, J.W., Aerts, J., Groenen, M.A., *et al.*, 2004. A physical map of the chicken genome. *Nature* 432, 761–764.
- Wang, W., Cao, X.H., Miclaus, M., Xu, J., Xiong, W., 2017. The promise of agriculture genomics. *Int J Genom* 2017, 9743749.
- Ward, N., Moreno-Hagelsieb, G., 2014. Quickly finding orthologs as reciprocal best hits with BLAT, LAST, and UBLAST: How much do we miss? *PLOS ONE* 9, e101850.
- Wedemeyer, N., Schmitt-John, T., Evers, D., *et al.*, 2000. Conservation of the 3'-untranslated region of the Rab1a gene in amniote vertebrates: Exceptional structure in marsupials and possible role for posttranscriptional regulation. *FEBS Lett* 477, 49–54.
- Wei, L., Liu, Y., Dubchak, I., Shon, J., Park, J., 2002. Comparative genomics approaches to study organism similarities and differences. *J Biomed Inform* 35, 142–150.
- Wetterstrand, K.A., 2017. DNA sequencing costs: Data from the NHGRI Genome Sequencing Program (GSP) 2017. Available at: www.genome.gov/sequencingcostsdata (accessed 05.12.17).
- Womack, J.E., 1998. The cattle gene map. *ILAR J* 39, 153–159.
- Womack, J.E., Kata, S.R., 1995. Bovine genome mapping: Evolutionary inference and the power of comparative genomics. *Curr Opin Genet Dev* 5, 725–733.
- Wright, F.A., O'Connor, D.T., Roberts, E., *et al.*, 1999. Genome scan for blood pressure loci in mice. *Hypertension* 34, 625–630.
- Zhang, J., Baran, J., Cros, A., *et al.*, 2011. International Cancer Genome Consortium Data Portal – A one-stop shop for cancer genomics data. *Database (Oxford)* 2011, bar026.
- Zhang, L., Li, W.H., 2004. Mammalian housekeeping genes evolve more slowly than tissue-specific genes. *Mol Biol Evol* 21, 236–239.

Relevant Website

<http://www.genomeasia100k.com/>
GenomeAsia 100k.

Single Nucleotide Polymorphism Typing

Srilakshmi Srinivasan, Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD, Australia

Jyotsna Batra, Institute of Health and Biomedical Innovation and School of Biomedical Sciences, Queensland University of Technology, Brisbane, QLD, Australia and Australian Prostate Cancer Research Centre – Queensland, Translational Research Institute, Woolloongabba, QLD, Australia

© 2019 Elsevier Inc. All rights reserved.

Abbreviations

ARMS	Amplification refractory mutation system	HRM	High-resolution melting curve
ASO	Allele-specific oligonucleotide	ISFET	Ion-sensitive field-effect transistor
AS-PCR	Allele-specific PCR	Mb	Mega bases
AS-SBE	Allele-specific single-base primer extension	NGS	Next generation sequencing
BAQ	Base alignment quality	OLA	Oligonucleotide ligation assay
bp	Base pair	ONT	Oxford Nanopore Technologies
CCD	Charged couple device	PCR	Polymerase chain reaction
CLR	Continuous long read	PGM	Personal genome machine
CMOS	Complementary metal-oxide-semiconductor	RFLP	Restricted-fragment length polymorphism
CRT	Cyclic reversible termination	SBL	Sequencing by ligation
dHPLC	Denaturing high-performance liquid chromatography	SBS	Sequencing-by-synthesis
dNTPs	Deoxy-nucleotide triphosphates	SMRT	Single-molecule real-time
dsDNA	Double-stranded DNA	SMS	Single molecule sequencing
FRET	Fluorescence resonance energy transfer	SNA	Single-nucleotide addition
GATK	Genome-analysis-toolkit	SNPs	Single nucleotide polymorphisms
Gb	Giga bases	SSCP	Single-strand conformation polymorphism
GWAS	Genome-wide association studies	ssDNA	Single-stranded
		ZMW	Zero-mode waveguides

Introduction

SNPs have emerged as important biomarkers to monitor disease prognosis and diagnosis. A SNP is a single base pair (bp) change either an insertion/deletion/substitution at a specific locus, with rare allele frequency usually $> 1\%$. These variations in DNA sequences may alter the genetic code and result in disease conditions or undesirable traits. Historically, a number of different SNP typing approaches have been used in both research and clinical laboratories (Srinivasan *et al.*, 2016). Presently, there is no single SNP genotyping method that is ideal for all research and/or clinical laboratory applications. Commonly used current chemistries include hybridization, primer extension, and cleavage methods coupled to various detection systems (Srinivasan and Batra, 2014). A multitude of innovations in these reagents and instrumentation have supported the initiation of the Human Genome Project and have led to more advanced technologies some of which are discussed in this review. However, the advanced technologies require a wide variety of algorithms for SNP detection from the raw data generated which is not crucial for the conventional methods. Thus we have categorized the SNP genotyping technologies into approaches that may or may not require computational algorithms for SNP detection as below.

Methods That do not Require Complex Computational Analysis

Amplification Refractory Mutation System (ARMS)

ARMS is an allele-specific polymerase chain reaction (PCR) for the detection of single base changes based on the use of sequence-specific PCR primers that allows amplification of template DNA only when the target allele is present. Thus, an ARMS primer can be designed specifically to amplify a specific member of a multi-allelic system. After PCR, the patterns of the PCR products (such as differences in length) allow differentiation of the alleles (Newton *et al.*, 1989).

High-Resolution Melting (HRM) Curve Analysis

HRM analysis is the technique introduced in 2003 to determine whether two PCR amplicons have identical sequences in the presence of a fluorescent reporter dye. Following amplification, the amplified product is exposed to an increasing gradient of temperature to reduce and denature the helicity of the double-stranded oligonucleotide, releasing the fluorescent dye (Montgomery *et al.*, 2007). Once released, the dye undergoes a conformational change that reduces the amount of fluorescence produced and a thermocycler will record the fluctuations in fluorescence and produce a melt-curve unique to the amplicon sequence analysed. HRM technique thus can compare

the similarity of two amplicons by the melting curve of each oligonucleotide, length, and primary structure (Hjelmso *et al.*, 2014). The weak A-T bond is disrupted at a lower temperature than the G-C bond, and therefore, A-T-rich regions of the amplicon denature at lower temperatures than the G-C-rich regions (Wittwer *et al.*, 2003). The distribution of these A-T- or G-C-rich regions of the oligonucleotide dictates the resulting melt curve and can be used to compare the similarity of amplicons from multiple samples. Although, a simple, cost effective technique, SNPs at GC rich regions are difficult to detect as they introduce additional melting domains into the melting curve and only a small number of SNPs can be multiplexed at one time (Venables *et al.*, 2014).

Restricted Fragment Length Polymorphism (RFLP)

RFLP is a common choice of genotyping in regular laboratories for genetic association studies. RFLP is based on the technique that differentiates SNPs by analysis of patterns derived from cleavage of amplified DNA fragments by restriction enzymes (Chuang *et al.*, 2008). The samples may yield different sized fragments at the SNP site due to either creation or loss of a restriction endonuclease site because of the change in the genomic DNA (Awasthi *et al.*, 2014). To discriminate SNP by this assay, the restriction enzymes must be unique and recognize only one of the SNP containing sequences. This technique is a simple cost-effective method but requires large amount of sample DNA and manual mining for restriction enzyme sites is challenging and cumbersome (Ding *et al.*, 2017).

Invader Assay

Invader assay developed by Third Wave Technologies, Inc. uses fluorescence resonance energy transfer (FRET) detection and does not involve PCR, restriction digestion, or gel electrophoresis. The technology relies on the specificity of Cleavase® enzymes, that recognize and cleave structures that form when the 3' end of an upstream "invading" oligonucleotide overlaps the hybridization site of the 5' end of a downstream oligonucleotide probe by at least 1 bp (Hayashi *et al.*, 2017). This activity enables detection of single-nucleotide mismatches immediately upstream of the cleavage site on the downstream DNA strand because mispairing results in the formation of a noninvasive structure that the enzyme does not recognize as a cleavable substrate. The generation of the proper enzyme substrate is dependent on base-pairing at a critical position between the two oligonucleotides and the target nucleic acid, which provides the ability to discriminate single-base changes.

Single-Strand Conformation Polymorphism (SSCP)

SSCP is a simple and sensitive assay for SNP detection and genotyping. The principle is based on the conformation of the single-stranded DNA (ssDNA). In the presence of a single base change, the conformation of the ssDNA is changed and causes shift in the migration of the DNA under non-denaturing electrophoresis conditions. This results in different band patterns for wild type and mutant DNA samples (Choi and Jung, 2017). A simple SSCP involves four steps: 1) a PCR amplification of target DNA sequence, 2) denaturation of the PCR products, 3) self-annealing of the denatured DNA, and 4) detection in mobility difference of the ssDNA molecules under non-denaturing conditions (Gupta *et al.*, 2014). The DNA can be visualized by different mobility shifts by incorporating radio-isotopes, fluorescent dyes, capillary-based or silver staining.

Heteroduplex Analysis

Hetero-duplex analysis is a low-medium high-throughput SNP detection technique. During heteroduplex analysis, the target gene is PCR amplified and the amplified products are then denatured, re-annealed slowly to form four different double-stranded DNA molecules from the two alleles of the gene (Paniego *et al.*, 2015). In the presence of a heterozygous mutation, two homoduplexes comprising of two perfectly complementary strands (mutant-mutant and wildtype-wildtype) and two heteroduplexes, that contain a sequence mismatch or denaturation bubble (mutant-wildtype and wildtype-mutant) are formed (Palais *et al.*, 2005). The re-annealed DNA molecules on re-annealing can be separated by denaturing high-performance liquid chromatography (dHPLC), enzymatic (RNase cleavage assay), electrophoretic methods and chemical cleavage assays. Heteroduplexes can then be analysed by their differential mobility during electrophoresis, which allows the detection of sequence changes in comparison to the wild type (Velasco *et al.*, 2007).

Allele-Specific Hybridization (Taqman Assay)

Allele-specific hybridization is a technique based on discriminating between alleles at SNP using allele-specific oligonucleotide (ASO) probes and exploits the 5' exonuclease activity of DNA polymerases (Gaudet *et al.*, 2009). Allele-specific PCR (AS-PCR) is achieved by PCR by an allele-specific primer rather than direct detection by a probe. Three primers are required, two to match the two alleles at the SNP site and the third to anneal to the opposite strand (Germer and Higuchi, 2003; Dhas *et al.*, 2015). Some of the tools used for designing AS primers are Tetra-primer, Visual-OMP and Primo SNP. Each of the two probes anneals to one allele to create a stable structure that would lead to its degradation by DNA polymerase, but when it anneals to the other allele, it forms a structure that is less stable so that the probe gets pushed off the template without being PCR amplified. A recent extension of the allele-specific expression is quantification of the allele-specific expression at mRNA level by fluorescence *in situ* analysis in a single-cell. Although simple and effective, the TaqMan assay requires substantial optimization and probe design remains largely empirical that makes it less preferable for large studies.

Allele-Specific Single-Base Primer Extension (AS-SBE)

AS-PCR makes use of the difference in extension between primers with matched and mismatched 3' bases. AS-PCR requires two primers that anneal to the target with their 3' bases matching the two alleles of an SNP. The allele-specific primers are labelled to enable their identification (Kisaki *et al.*, 2010; Balschun *et al.*, 2011). Allele-specific expression (ASE) can be conducted *in situ* on a chip when two allele-specific primers are anchored at different loci. SBE is considered to have a superior discrimination compared to ASE and is dependent on the sequence surrounding the polymorphism.

Oligonucleotide Ligation Assay (OLA)

OLA is a DNA dependent amplification that uses a set of four oligonucleotides, in the presence of a thermostable Taq DNA ligase, to detect SNP alleles. Any nucleotide variation at the ligation junction can be detected using a fluorescent or biotin labelling method (Dokuta *et al.*, 2013; Mutsavangwa *et al.*, 2014). The ligation occurs only when the 3' detector nucleotide is complementary to the SNP nucleotide. Software such as GeneMapper are used for analyzing the raw data and calling SNP genotypes. Two fluorescent peaks in an electropherogram represents two alleles for a specific SNP (Tobler *et al.*, 2005). A variation of this technique was recently reported and known as Heated OLA that uses a thermostable ligase and cycles of denaturation and hybridization for ligation and SNP detection (Beck *et al.*, 2014).

Mid and High-Throughput Technologies Requiring Computational Analysis

Mid-throughput technologies that identified SNPs utilizing SBE are also desirable as modest multiplexing assays with minimal assay setup costs such as Sequenom MassARRAY platform (Clendenen *et al.*, 2015; Miller *et al.*, 2014). After PCR of the SNP region, a third primer is added that binds one position upstream of the SNP site and the four nucleotide bases linked to terminators are added. The nucleotide at the SNP is extended by an enzyme. The samples genotype is determined by hybridizing extended primers to a universal DNA array and by determining the identity of the extended base by hybridization pattern analysis. This can be used as second-tier applications such as validating hits after genome-wide association studies (GWAS) (Batra *et al.*, 2011; Lose *et al.*, 2012; Lose *et al.*, 2013). SNP microarrays started more than two decades ago and have progressed from low-throughput to the current mid-throughput Affymetrix GeneChip SNP platforms, Illumina Golden Gate Assay or the Infinium Assay of Illumina to study thousands of SNPs simultaneously. In this approach the arrayed DNA is extended at the SNP site in a single extension reaction or the bar-coded oligos are hybridized to a universal array (Cutler *et al.*, 2001; Bumgarner, 2013).

Here for the high-throughput technologies reviewed, initially, we discuss the physical method followed by the computational analysis for the SNP detection.

Sequencing by Ligation (SBL)

SBL approaches comprise the hybridization and ligation of labelled probes of varying lengths and short, known anchor sequences to a DNA strand (Tomkinson *et al.*, 2006). The emission spectra by the fluorophore indicates its complementarity to the probe (Pu *et al.*, 2015). In SBL, the DNA is clonally amplified on a solid surface, the presence of thousands of copies reduces the background noise. The probes encode one or two known bases, enabling complementary binding of the probe to the template. The anchor fragment encodes a sequence complementary to an adapter sequence and initiates ligation. After ligation, the template is imaged to identify the known base or bases in the probe (Landegren *et al.*, 1988). A new cycle starts after removing the anchor-probe complex or removal of the fluorophore by a cleavage event and to restore the ligation site.

Next-generation sequencing (NGS) platforms based on SBL including, SOLiD and Complete Genomics systems utilizes deoxy-nucleotide triphosphates (dNTPs) that are probed multiple times (Valouev *et al.*, 2008). In Complete Genomic's DNA nanoballs' technique, the sequences are obtained by probe-ligation, but the clonal DNA amplification is performed by rolling circle amplification unlike the bead or emulsion amplification. The rolling circle amplification generates long DNA chains with repetitive elements of template bordered by adapters which then assemble into nanoballs that are fixed to a slide and sequenced. Both SOLiD and Complete Genomics systems are termed to have a very high accuracy of ~99.9% and generate both single-end and paired-end sequencing (in which the sequence at both ends of each DNA cluster is recorded) (Drmanac *et al.*, 2010; Liu *et al.*, 2012). Although highly accurate, they still are reported to miss true variants (Wall *et al.*, 2014) and false representation of AT-rich regions (Rieber *et al.*, 2013). For example, SOLiD system display substitution errors and GC-rich under-representation (Harismendy *et al.*, 2009). They also generate very short read lengths (~75 bp for SOLiD and 28–100 for Complete Genomics) limiting their wider application and longer run times.

Polony sequencing is a technique to amplify DNA *in situ* on a thin polyacrylamide film. The amplified DNA are localized in a gel and thus its mobility is restricted to the gel and form the so-called "polonies" for polymerase colonies (Shendure *et al.*, 2005). On a single microscope slide nearly 5 million polonies can be formed. The read length by this technique is up to 13 bases per colony (Zhang *et al.*, 2006) and multiplexing using this technique is also in use that combines polony amplification along with sequence-by-ligation method to sequence up to 14-base tags (Ruegger *et al.*, 2012). The limitation associated with this technique is low throughput and high costs.

Sequencing-by-Synthesis (SBS)

SBS can be classified into cyclic reversible termination (CRT) or as single-nucleotide addition (SNA)

Cyclic reversible termination (CRT)

CRT utilizes terminator molecules similar to Sanger sequencing by blocking the ribose 3'-OH group thus preventing elongation (Seo *et al.*, 2005). To begin amplification, the DNA template need to be primed by a complementary sequence to adapter. This will initiate the binding of DNA polymerase to double stranded DNA (dsDNA). At each cycle, uniquely labelled dNTPs specific to each base are added after which the surface is imaged to identify the type of dNTP incorporated at each cluster (Guo *et al.*, 2008). The blocking group is then removed and a new cycle begins.

One example for this technique is Solexa Genome Analyzer uses a flow cell consisting of an optical transparent slide with 8 lanes of bound oligonucleotide anchors (Voelkerding *et al.*, 2009). Template DNA is sheared into 100 bp lengths and end-paired to generate 5'-phosphorylated blunt ends. The DNA fragments are then passed over complementary oligonucleotides bound to a flowcell and a follow-up solid phase PCR generates clusters of clonal populations from each of the individual flow-cell binding DNA strands. Sequencing is by SBS using fluorescent 'reversible-terminator' dNTPs that allows the sequencing to occur in a synchronous manner (Turcatti *et al.*, 2008). The incorporated nucleotide at each cycle is monitored by a charged couple device (CCD) by exciting fluorophores with laser light, before the removal of the fluorescent moiety and continuation to next step. Earlier Genome Analyzers were only capable of short reads (~35 bases long) but can produce paired-end data. NextSeq and MiniSeq platforms utilize two-colour labelling system and reduce the costs by reducing scanning to two colour channels. However, this results in a higher error rate and underperformance for less diverse samples. MiSeq has a low-throughput but less expensive with faster turnaround and longer read lengths (Balasubramanian, 2011; Quail *et al.*, 2012).

GeneReader (Qiagen) is envisioned for clinical application on cancer gene panels and is well optimized (Darwanto *et al.*, 2017). It's an all-in-one platform from sample preparation to analysis. The NGS platform is coupled with QIAcube for sample preparation. With a long run-time GeneReader is likely to have the same merits and demerits as the MiSeq platform providing a lower cost effective option per gigabase (Gb) sequenced (Mardis, 2017). The standard Genome Analyzer was later taken over by the HiSeq that is capable of a greater read length and depth but needs rigorous optimization and therefore is limited for fewer applications such as whole genome sequencing. Cancer panels (Novogene, USA) for personalized medicine that specifically targets cancer specific utilizing HiSeq for sequencing are currently available. This cancer panel called NovoPM™ is available for both tissues, liquid biopsy, FFPE and circulating DNA and is considered to be cost-effective service for cancer studies.

Single-nucleotide addition (SNA)

SNA relies on a single signal unlike CRT to incorporate dNTPs into an elongating strand. The nucleotides are added to a sequencing reaction to ascertain that only one dNTP is responsible for the signal. This doesn't require blocking of the dNTP as in CRT to prevent elongation. For the homopolymer regions that involves identical dNTPs, a proportional increase in the signal will be observed as the nucleotides get incorporated.

Pyrosequencing is a sequence-by-synthesis method for *de novo* DNA sequencing. The technique is based on the incorporation of nucleotides base by base by converting the production of pyrophosphate into light (Ronaghi *et al.*, 1998). The amount of pyrophosphate is proportional to the light produced (Royo *et al.*, 2007; Royo and Galan, 2009). The read length for pyrosequencing has an upper limit of 400 bases and has a better detection and accuracy compared to the conventional Sanger sequencing. However, the major concern with this technique is determining the number of incorporated nucleotides in homopolymeric regions as the light response is not linear after incorporation of more than 5–6 identical nucleotides. Further, the primers need to be checked for primer-dimer, self-looping and cross-hybridizations (Heather and Chain, 2016). One example utilizing this principle is 454 Pyrosequencer developed by Life Science in 2000. The 454 Pyrosequencer is the first commercial high throughput-next generation sequencing (HT-NGS) platform. This SNA method distributes beads bound to the template into a TiterPlate along with enzyme cocktail (Batra *et al.*, 2014). As a dNTP gets incorporated into a strand, enzymatic cascade occurs liberating a bioluminescence signal. The light is then detected by a CCD camera that records the number of nucleotides incorporated at a particular bead.

The Ion Torrent is the first sequencer without optical sensing (Rothberg *et al.*, 2011). To generate a signal the Ion Torrent detects the H⁺ ions released after each dNTP incorporation. The resultant change in pH is recognised by an integrated ion-sensitive field-effect transistor (ISFET) and a complementary metal-oxide-semiconductor (CMOS). The change in the pH is proportional to the number of nucleotides identified. But this limits the accuracy in detection of the length of homopolymers. The 454 and Ion Torrent systems generate greater read lengths compared to other short-read sequencers with an average read length of 700 and 400 bp, respectively (Goodwin *et al.*, 2016). This offers advantage especially for sequencing repetitive DNA regions or complex DNA. However, insertion and deletion errors dominate for homopolymers although they are compared to be on par compared to other NGS platforms in sequencing non-homopolymers. The 454 sequencers could not compete with the upcoming new technologies in terms of yield or costs (Voelkerding *et al.*, 2010).

Ion Torrent platform offers a wide range of chips and instrument to consumer needs for chips ranging from ~50 megabase (Mb) to 15 Gb with a shorter runtime ~2–7 h, making it the faster than other current platforms (Stahl and Lundeberg, 2012). Some of them are currently used for clinical applications, transcriptome profiling and splice sites detection. The Ion Personal Genome Machine (PGM) and the Ion S5 series are some of the machines currently used for dedicated diagnostic applications

(Roy *et al.*, 2016). The Ion Proton and S5 devices cannot perform paired-end sequencing, thus limiting their application for elucidating long-range genomic or transcriptomic structure.

Long-Read Technologies

Genomic DNA harbors long repetitive elements, copy number alterations and structural variations that makes it highly complex. Short-read paired-end technologies are not sufficient to resolve these complex elements and require technologies that can deliver long-reads for resolution of the long structural features. Currently, there are mainly two types of long-read technologies: single-molecule real-time sequencing and synthetic long-reads

Single molecule real-time long reads

The first single molecule sequencing (SMS) was developed by Stephen Quake *et al.*, and later commercialized by Helicos Biosciences (Pushkarev *et al.*, 2009). In this technique, DNA template is attached to a planar surface, and fluorescent reversible terminator dNTPs are washed over one base at a time, imaged and cleaved before the next base is cycled. This is a slow and expensive technique but was the first sequencing technology of non-amplified DNA, thus avoiding errors and biases (Heather and Chain, 2016).

Currently the widely used long-read technology is the single-molecule real-time (SMRT) sequencing approach by Pacific Biosciences (PacBio). PacBio[®] RSII sequencer uses a flow cell with thousands of picolitre wells – zero-mode waveguides (ZMW). Short-read SBS techniques (Illumina sequencers) bind the DNA and allow polymerase to bind to the template, PacBio anchors the polymerase to the well and allows DNA to travel through the ZMW (Rhoads and Au, 2015; Larkin *et al.*, 2017). SMRT also uses a unique circular template for the polymerase to traverse repeatedly thus allowing the sequencing of DNA multiple times to generate continuous long read (CLR). The polymerase then cleaves the fluorophore bound to the dNTP incorporated. The average length of a CLR is 10–60 kb and depends on the polymerase half-life. The SMRT-sequencing reads are then mapped by tools such as BLASR that allows confident mapping of the reads to their reference sequence.

In 2014, a first consumer prototype of a nanopore sequencer to generate long-reads was released – the MinION, Oxford Nanopore Technologies (ONT). ONT directly sequences a native ssDNA by measuring the changes in the current as the bases are stringed through the nanopore by a molecular motor protein. ONT MinION uses a hairpin library structure, the DNA template and its complement are bound by a hairpin adapter similar to the PacBio circular DNA template (Oikonomopoulos *et al.*, 2016). The raw reads are then split into two “1D” reads (“template” and “complement”) after removing the adapter (Laver *et al.*, 2015). Due to similar data features with PacBio, many researchers have utilized or are testing ONT in applications where PacBio has been applied. Because of the unique nature of ONT technique, as there are 1000 distinct signals, it has a large error rate (~30%) particularly for indel errors. Homopolymer sequencing also remains a challenge for ONT MinION and PacBio. Further the cost per base is higher than Illumina platforms and the per-base error rates are high (5%–15%) (Cornelis *et al.*, 2017). Recently, improvements and algorithms to improve the accuracy are in progress.

Synthetic long-reads

Synthetic approaches generate long-reads by library preparation that barcodes associate fragments to allow assembly computationally of a larger fragment. Large DNA fragments are partitioned into microtiter wells or into an emulsion with few molecules in each partition. The template fragments within individual partition are sheared and barcoded. This allows sequencing on existing short-read instruments, after which the data are split by barcoding and reassembled. By segregating fragments, complicated regions can be isolated allowing for them to be assembled locally. This reduce the breaks (gaps) in the obtained sequence data (Schatz *et al.*, 2010; English *et al.*, 2015). Two popular systems using this technique are the Illumina synthetic long-read sequencing platform and 10X Genomics emulsion-based system.

The Illumina Truseq Long-Read DNA partitions DNA into a microtiter plate and do not require special instrumentation whereas the 10X Genomics emulsion-based system use emulsion to partition DNA and use a microfluidic instrument for pre-sequencing reactions (Mccoy *et al.*, 2014). The DNA required for 10X Genomics is as little as 1 ng (nanogram) and can arbitrarily partition large DNA fragments (~100 kb) into micelles known as ‘GEMs’. Each GEM would contain approximately 0.3x genome copies and a unique barcode. Within each GEM, smaller fragments of DNA are amplified. The reads are aligned and grouped together to form a series of anchored fragments across the original fragment. Grouped reads from each long fragment can be restacked, combining their individual coverage into an overall map. As the Illumina long-read approach relies on the existing instrument, the throughput and error profile are similar to those of the current short-read sequencers (Goodwin *et al.*, 2016). Depending on the method of DNA partition, higher coverage is required, thus is associated also with high costs using the Illumina long-read approach. The 10X Genomics uses less DNA but the partitioning is not very effective and can lead to surplus DNA within a droplet complicating deconvolution. This leads to ambiguity and makes analysis difficult.

Analysis of the SNP Data Generated by the Sequencers

The last few years have seen incredible progress in the development of algorithms and software to make sense of the short reads onto a reference sequence and to identify the differences between the individual and the reference. The complexity of genome

makes it impossible to generate genetic maps that are based on the segregation of genetic variations through pedigrees that provided information about the order of sequences at specific loci and at the chromosomes scale. Most popular SNP callers are focused primarily on the high recovery of SNPs while keeping low false discovery rates.

De novo assembly to the reference genome

The first step in data analysis is alignment of the short sequences obtained from the sequencing methods to the existing reference genome. Alignment starts with indexing, following alignment. The common aligners are Bowtie2 and SOAP3-dp that index reference genome rather than sequenced reads, which is less time consuming as indexing reference needs to be performed only once, while indexing sequence has to be performed for each sample individually (Luo *et al.*, 2013). Some of the indexing methods are suffix/prefix trie, enhanced suffix array, FM-index and hash table. The *suffix/prefix trie* is defined as a data structure representing the set of suffixes of a given string of characters, enabling fast matching of the string (a sequence read) to the reference genome. However for very large data, this method is not efficient and replaced by the enhanced suffix array approach that enables storing of large genome data. FM-index is considered to be better of the first three indexing methods and some popular software tools utilizing this algorithm are Bowtie, SOAP2 and SOAP3 (Chacon *et al.*, 2013). The hash table method is based on the seed-and-extend algorithm. Some of the algorithms in use using the hash table method are BFAST, MAQ, RMAP and SHRiMP (Wu, 2016).

Regardless of the indexing method, the alignment can be performed by many algorithms and the resulting alignment can be gapped or ungapped (Yorukoglu *et al.*, 2016). Alignment gaps are a result of genomic rearrangements, such as insertions/deletions. Allowing gaps in alignment is a preferred option by most alignment software tools. The first alignment tool, SHRiMP was able to perform alignment for single-end sequence and currently most of the algorithms support a paired-end alignment which considers both forward and reverse orders as a pair resulting in a longer piece of sequence and improving quality of alignment.

Post-alignment processing

The most commonly applied post-alignment methods include output file format converting, removing PCR artefacts or creating reports from the alignment process (Mielczarek and Szyda, 2016). These can be carried out using SAMtools package. SAMtools allows reducing the data set size and downstream compatibility with variant callers, such as Unified Genotyper (Genome Analysis Toolkit package) and other software tools (Cornish and Guda, 2015). A simple summary is generated to describe the alignment process (Bowtie2, SOAP2, and MOSAIK) and the description includes the total number of aligned reads, number of properly aligned reads, number of reads aligned only once. Summary statistics helps to assess the overall quality and accuracy of alignment. Removing PCR duplicates can be done by SAMtools rmdup tool (Ebbert *et al.*, 2016).

Pre-variant calling processing

To obtain reliable polymorphic variant calling, additional steps for SNP calling are recommended before the actual variant detection. Some alignment artefacts occur during alignment that may result in mismatching of different bases. These mismatches are most likely to be considered as SNPs (Nielsen *et al.*, 2011). To avoid this, local realignment tools are designed that allows realignment of reads at the site of the artefact to minimize the number of mismatching bases. A large amount of genomic DNA require realignment due to the presence of InDels with respect to the reference genome (Olson *et al.*, 2015). A sequencer may miscalculate the base quality expressed as the Phred score. Genome-analysis-toolkit (GATK) package tools recalibrate base quality scores to more accurate estimates to reduce the mismatches with the reference genome (Mckenna *et al.*, 2010; Maretty *et al.*, 2017). SAMtools provide a score for each base called a base alignment quality (BAQ) score, which is calculated as Phred-scaled probability for a base being misaligned. If a base is sub-optimally aligned to a different reference base, the BAQ score is low. Some programs do not allow pre-SNP calling step, two examples include SOAPsn and Atlas-SNP2 (Yu and Sun, 2013).

SNP calling

Conventionally SNPs were identified by techniques such as microarrays. Today microarrays have tremendously progressed to process about hundreds of thousands of SNPs. Some of the platforms such as Affymetrix, Illumina based on microarray for SNP detection utilize software such as Dynamic Model software (100 K SNPs) that do not require any normalization step (Di *et al.*, 2005), Bayesian Robust Linear Model with Mahalanobis Distance Classifier algorithm for 500 K SNP arrays (Hong *et al.*, 2008) and GenCall algorithm (Teo *et al.*, 2007; Steemers and Gunderson, 2007). Recent NGS techniques have provided new avenues to identify a higher number of variants including the rare SNPs that are important for complex disease phenotypes. However, for a reliable detection of these variants require a high depth of coverage so as to differentiate them from sequencing errors. Multiple algorithms and software tools are currently available for SNP calling. SNP calling may be defined as the process of identifying regions differing from a reference sequence, while genotype calling refers to the determination of genotypes (Nielsen *et al.*, 2011). SNP and genotype callers are based on either heuristic or probabilistic methods.

Heuristic methods call variants are less commonly used than probabilistic methods and are based on multiple information sources associated with the data structure and quality. Example of software using heuristic approach is VarScan2 which is also a statistical test based on the number of reads for each allele (Durtschi *et al.*, 2013). A heuristic approach determines genotype based on thresholds for base quality (minimum 20), coverage (minimum 33), and allele frequency (minimum 0.08). After determining genotype, Fisher's exact test of read counts for each allele is analysed in comparison to the expected distribution based on the sequencing error alone.

Probabilistic method measures statistical uncertainty for called genotypes allowing to monitor the accuracy of genotype calling. Additionally, information regarding allele frequencies, linkage disequilibrium can be included in the analysis. Genotype calling is based on likelihood calculations and adopts Bayes' theorem (Maruki and Lynch, 2014). After realignment and quality score recalibration, the next step is to calculate likelihood for each possible genotype at each base (a homozygote for the reference allele, a homozygote for the minor allele or a heterozygote). In Bayesian framework, the computed likelihood is combined with a previous genotype probability to derive a posterior probability of a genotype (Nielsen *et al.*, 2012; Fumagalli *et al.*, 2013). SAMtools is an example of software including probabilistic approach. SAMtools mpileup collects data stored in input BAM (binary version of the Sequence Alignment Map format) files and computes likelihood of data for each possible genotype. Recently, a study compared thirteen pipelines and popular read aligners against twelve sequence data sets from a single genome. For the SNP calls from Illumina data sets, the BWA-MEM and SAMtools exhibited best performance and Freebayes aligner showed high quality for SNP calls. For SNPs from Ion Proton data set, SAMtools performed best including TVC (Hwang *et al.*, 2015). Thus, researchers must exhibit extreme care when choosing the appropriate algorithms for variant detection and analysis after sequencing.

Conclusion and Prospects

It is difficult to directly compare and suggest the best sequencing system between different platforms. This is because the technique to choose is to be weighed between the costs, coverage and accuracy. Different platforms may require different levels of accuracy to achieve the same level of accuracy and comparisons have to be made either by considering coverage or higher costs to achieve better accuracy. With the increasing research to develop robust sequencing technologies, it is certainly possible in near term that a relatively inexpensive, highly accurate with better coverage sequencer will be achievable that can be used in clinics.

NGS techniques are now becoming a routine part of biological research. These techniques is boosting research into areas which once was thought to be unachievable. Some sequencing-based technologies target single cells that will address many new and longstanding questions. For example, single-cell genomics will help to discover cell lineage relationships and single-cell transcriptomics will succeed in discovering marker-based cell types. In a decade, these technologies may enable high-throughput, multi-dimensional analyses of single cells that will produce comprehensive information of the cell lineage of higher organisms. Such studies will have profound implications to allow revolutionary science including precision medicine based on personalized genome sequencing.

Declaration of Interest

J. Batra is a National Health and Medical Research Council Career Development Fellow. S. Srinivasan is supported by an Advance QLD ECR Research Fellowship.

See also: Detecting and Annotating Rare Variants. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Sequence Analysis. Whole Genome Sequencing Analysis

References

- Awasthi, S.P., Asakura, M., Neogi, S.B., *et al.*, 2014. Development of a PCR-restriction fragment length polymorphism assay for detection and subtyping of cholix toxin variant genes of *Vibrio cholerae*. *J. Med. Microbiol.* 63, 667–673.
- Balasubramanian, S., 2011. Sequencing nucleic acids: From chemistry to medicine. *Chem. Commun. (Camb.)* 47, 7281–7286.
- Balschun, K., Wenke, A.K., Rocken, C., Haag, J., 2011. Detection of KRAS and BRAF mutations in advanced colorectal cancer by allele-specific single-base primer extension. *Expert Rev. Mol. Diagn.* 11, 799–802.
- Batra, J., Nagle, C.M., O'mara, T., *et al.*, 2011. A Kallikrein 15 (KLK15) single nucleotide polymorphism located close to a novel exon shows evidence of association with poor ovarian cancer survival. *BMC Cancer* 11, 119.
- Batra, J., Srinivasan, S., Clements, J.A., 2014. Single nucleotide polymorphisms (SNPs). In: Jothy, G.M.Y.A.S. (Ed.), *Molecular Testing in Cancer*, 2014 ed. New York: Springer Science & Business.
- Beck, I.A., Deng, W., Payant, R., *et al.*, 2014. Validation of an oligonucleotide ligation assay for quantification of human immunodeficiency virus type 1 drug-resistant mutants by use of massively parallel sequencing. *J. Clin. Microbiol.* 52, 2320–2327.
- Bumgarner, R., 2013. Overview of DNA microarrays: Types, applications, and their future. *Curr. Protoc. Mol. Biol.* (Chapter 22, Unit 22 1).
- Chacon, A., Moure, J.C., Espinosa, A., Hernandez, P., 2013. n-step FM-Index for faster pattern matching. In: *Proceeding of 2013 International Conference on Computational Science*, 18, pp. 70–79.
- Choi, W., Jung, G.Y., 2017. Highly multiplex and sensitive SNP genotyping method using a three-color fluorescence-labeled ligase detection reaction coupled with conformation-sensitive CE. *Electrophoresis* 38, 513–520.
- Chuang, L.Y., Yang, C.H., Tsui, K.H., *et al.*, 2008. Restriction enzyme mining for SNPs in genomes. *Anticancer Res.* 28, 2001–2007.
- Clendenen, T.V., Rendleman, J., Ge, W., *et al.*, 2015. Genotyping of single nucleotide polymorphisms in DNA isolated from serum using sequenom MassARRAY technology. *PLOS ONE* 10, e0135943.
- Cornelis, S., Gansemans, Y., Deleye, L., Deforce, D., Van Nieuwerburgh, F., 2017. Forensic SNP Genotyping using Nanopore MinION Sequencing. *Sci. Rep.* 7, 41759.

- Cornish, A., Guda, C., 2015. A comparison of variant calling pipelines using genome in a bottle as a reference. *Biomed. Res. Int.*
- Cutler, D.J., Zwick, M.E., Carrasquillo, M.M., *et al.*, 2001. High-throughput variation detection and genotyping using microarrays. *Genome Res.* 11, 1913–1925.
- Darwanto, A., Hein, A.M., Strauss, S., *et al.*, 2017. Use of the QIAGEN GeneReader NGS system for detection of KRAS mutations, validated by the QIAGEN Therascreen PCR kit and alternative NGS platform. *BMC Cancer* 17.
- Dhas, D.B., Ashmi, A.H., Bhat, B.V., Parija, S.C., Banupriya, N., 2015. Modified low cost SNP genotyping technique using cycle threshold (Ct) & melting temperature (Tm) values in allele specific real-time PCR. *Indian J. Med. Res.* 142, 555–562.
- Di, X., Matsuzaki, H., Webster, T.A., *et al.*, 2005. Dynamic model based algorithms for screening and genotyping over 100 K SNPs on oligonucleotide microarrays. *Bioinformatics* 21, 1958–1963.
- Ding, M., Duan, X., Feng, X., Wang, P., Wang, W., 2017. Application of CRS-PCR-RFLP to identify CYP1A1 gene polymorphism. *J. Clin. Lab. Anal.*
- Dokuta, S., Utaipat, U., Praparattanapan, J., *et al.*, 2013. Improvement of the oligonucleotide ligation assay for detection of the M184V drug-resistant mutation in patients infected with human immunodeficiency virus type 1 subtype CRF01_AE. *J. Virol. Methods* 190, 20–28.
- Drmanac, R., Sparks, A.B., Callow, M.J., *et al.*, 2010. Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science* 327, 78–81.
- Durtschi, J., Margraf, R.L., Coonrod, E.M., Mallempati, K.C., Voelkerding, K.V., 2013. VarBin, a novel method for classifying true and false positive variants in NGS data. *BMC Bioinform.* 14.
- Ebbert, M.T., Wadsworth, M.E., Staley, L.A., *et al.*, 2016. Evaluating the necessity of PCR duplicate removal from next-generation sequencing data and a comparison of approaches. *BMC Bioinform.* 17, 239.
- English, A.C., Salerno, W.J., Hampton, O.A., *et al.*, 2015. Assessing structural variation in a personal genome-towards a human reference diploid genome. *Bmc Genom.* 16.
- Fumagalli, M., Vieira, F.G., Korneliussen, T.S., *et al.*, 2013. Quantifying population genetic differentiation from next-generation sequencing data. *Genetics* 195, 979.
- Gaudet, M., Fara, A.G., Beritognolo, I., Sabatti, M., 2009. Allele-specific PCR in SNP genotyping. *Methods Mol. Biol.* 578, 415–424.
- Germer, S., Higuchi, R., 2003. Homogeneous allele-specific PCR in SNP genotyping. *Methods Mol. Biol.* 212, 197–214.
- Goodwin, S., McPherson, J.D., McCombie, W.R., 2016. Coming of age: Ten years of next-generation sequencing technologies. *Nat. Rev. Genet.* 17, 333–351.
- Guo, J., Xu, N., Li, Z., *et al.*, 2008. Four-color DNA sequencing with 3'-O-modified nucleotide reversible terminators and chemically cleavable fluorescent dideoxynucleotides. *Proc. Natl. Acad. Sci. USA* 105, 9145–9150.
- Gupta, V., Arora, R., Gochhait, S., Bairwa, N.K., Bamezai, R.N., 2014. Gel-based nonradioactive single-strand conformational polymorphism and mutation detection: Limitations and solutions. *Methods Mol. Biol.* 1105, 365–380.
- Harismendy, O., Ng, P.C., Strausberg, R.L., *et al.*, 2009. Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol* 10, R32.
- Hayashi, K., Ishigami, M., Ishizu, Y., *et al.*, 2017. A comparison of direct sequencing and Invader assay for Y93H mutation and response to interferon-free therapy in hepatitis C virus genotype 1b. *J. Gastroenterol. Hepatol.*
- Heather, J.M., Chain, B., 2016. The sequence of sequencers: The history of sequencing DNA. *Genomics* 107, 1–8.
- Hjelmsø, M.H., Hansen, L.H., Baelum, J., *et al.*, 2014. High-resolution melt analysis for rapid comparison of bacterial community compositions. *Appl. Environ. Microbiol.* 80, 3568–3575.
- Hong, H., Su, Z., Ge, W., *et al.*, 2008. Assessing batch effects of genotype calling algorithm BRLMM for the Affymetrix GeneChip Human Mapping 500 K array set using 270 HapMap samples. *BMC Bioinform.* 9, S17.
- Hwang, S., Kim, E., Lee, I., Marcotte, E.M., 2015. Systematic comparison of variant calling pipelines using gold standard personal exome variants. *Sci. Rep.* 5, 17875.
- Kisaki, O., Kato, S., Shinohara, K., *et al.*, 2010. High-throughput single-base mismatch detection for genotyping of UDP-glucuronosyltransferase (UGT1A1) with probe capture assay coupled with modified allele-specific primer extension reaction (MASPER). *J. Clin. Lab. Anal.* 24, 85–91.
- Landegren, U., Kaiser, R., Sanders, J., Hood, L., 1988. A ligase-mediated gene detection technique. *Science* 241, 1077–1080.
- Larkin, J., Henley, R.Y., Jadhav, V., Korlach, J., Wanunu, M., 2017. Length-independent DNA packing into nanopore zero-mode waveguides for low-input DNA sequencing. *Nat. Nanotechnol.*
- Laver, T., Harrison, J., O'Neill, P.A., *et al.*, 2015. Assessing the performance of the oxford nanopore technologies minion. *Biomol. Detect. Quantif.* 3, 1–8.
- Liu, L., Li, Y., Li, S., *et al.*, 2012. Comparison of next-generation sequencing systems. *J. Biomed. Biotechnol.* 2012, 251364.
- Lose, F., Batra, J., O'mara, T., *et al.*, 2013. Common variation in Kallikrein genes KLK5, KLK6, KLK12, and KLK13 and risk of prostate cancer and tumor aggressiveness. *Urol. Oncol.* 31, 635–643.
- Lose, F., Lawrence, M.G., Srinivasan, S., *et al.*, 2012. The kallikrein 14 gene is down-regulated by androgen receptor signalling and harbours genetic variation that is associated with prostate tumour aggressiveness. *Biol. Chem.* 393, 403–412.
- Luo, R., Wong, T., Zhu, J., *et al.*, 2013. SOAP3-dp: Fast, accurate and sensitive GPU-based short read aligner. *PLOS ONE* 8, e65632.
- Mardis, E.R., 2017. DNA sequencing technologies: 2006–2016. *Nat. Protoc.* 12, 213–218.
- Marett, L., Jensen, J.M., Petersen, B., *et al.*, 2017. Sequencing and de novo assembly of 150 genomes from Denmark as a population reference. *Nature* 548, 87–91.
- Maruki, T., Lynch, M., 2014. Genome-wide estimation of linkage disequilibrium from population-level high-throughput sequencing data. *Genetics* 197, 1303. U421.
- Mccoy, R.C., Taylor, R.W., Blauwkamp, T.A., *et al.*, 2014. Illumina TruSeq synthetic long-reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. *PLOS ONE* 9, e106689.
- Mckenna, A., Hanna, M., Banks, E., *et al.*, 2010. The genome analysis toolkit: A mapreduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* 20, 1297–1303.
- Mielczarek, M., Szyda, J., 2016. Review of alignment and SNP calling algorithms for next-generation sequencing data. *J. Appl. Genet.* 57, 71–79.
- Miller, J.K., Buchner, N., Timms, L., *et al.*, 2014. Use of Sequenom sample ID Plus(R) SNP genotyping in identification of FFPE tumor samples. *PLOS ONE* 9, e88163.
- Montgomery, J., Wittwer, C.T., Palais, R., Zhou, L., 2007. Simultaneous mutation scanning and genotyping by high-resolution DNA melting analysis. *Nat. Protoc.* 2, 59–66.
- Mutsvangwa, J., Beck, I.A., Gwanzura, L., *et al.*, 2014. Optimization of the oligonucleotide ligation assay for the detection of nevirapine resistance mutations in Zambian Human Immunodeficiency Virus type-1 subtype C. *J. Virol. Methods* 210, 36–39.
- Newton, C.R., Graham, A., Heptinstall, L.E., *et al.*, 1989. Analysis of any point mutation in DNA. The amplification refractory mutation system (ARMS). *Nucleic Acids Res.* 17, 2503–2516.
- Nielsen, R., Korneliussen, T., Albrechtsen, A., Li, Y.R., Wang, J., 2012. SNP calling, genotype calling, and sample allele frequency estimation from new-generation sequencing data. *PLOS ONE* 7.
- Nielsen, R., Paul, J.S., Albrechtsen, A., Song, Y.S., 2011. Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.* 12, 443–451.
- Oikonomopoulos, S., Wang, Y.C., Djambazian, H., Badescu, D., Ragoussis, J., 2016. Benchmarking of the Oxford Nanopore MinION sequencing for quantitative and qualitative assessment of cDNA populations. *Sci. Rep.* 6, 31602.
- Olson, N.D., Lund, S.P., Colman, R.E., *et al.*, 2015. Best practices for evaluating single nucleotide variant calling methods for microbial genomics. *Front. Genet.* 6, 235.
- Palais, R.A., Liew, M.A., Wittwer, C.T., 2005. Quantitative heteroduplex analysis for single nucleotide polymorphism genotyping. *Anal. Biochem.* 346, 167–175.
- Panigero, N., Fusari, C., Lia, V., Puebla, A., 2015. SNP genotyping by heteroduplex analysis. *Methods Mol. Biol.* 1245, 141–150.
- Pu, D., Chen, J., Qian, X., Xiao, P., 2015. Probe optimization for sequencing by ligation. *J. Biochem.* 157, 357–364.
- Pushkarev, D., Neff, N.F., Quake, S.R., 2009. Single-molecule sequencing of an individual human genome. *Nat. Biotechnol.* 27, 847. U101.
- Quail, M.A., Smith, M., Coupland, P., *et al.*, 2012. A tale of three next generation sequencing platforms: Comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genom.* 13, 341.
- Rhoads, A., Au, K.F., 2015. PacBio sequencing and its applications. *Genom. Proteom. Bioinform.* 13, 278–289.

- Rieber, N., Zpatka, M., Lasitschka, B., *et al.*, 2013. Coverage bias and sensitivity of variant calling for four whole-genome sequencing technologies. *PLOS One* 8, e66621.
- Ronaghi, M., Uhlen, M., Nyren, P., 1998. A sequencing method based on real-time pyrophosphate. *Science* 281, 363.
- Rothberg, J.M., Hinz, W., Rearick, T.M., *et al.*, 2011. An integrated semiconductor device enabling non-optical genome sequencing. *Nature* 475, 348–352.
- Roy, S., Laframboise, W.A., Nikiforov, Y.E., *et al.*, 2016. Next-generation sequencing informatics: Challenges and strategies for implementation in a clinical environment. *Arch. Pathol. Lab. Med.* 140, 958–975.
- Royo, J.L., Galan, J.J., 2009. Pyrosequencing for SNP genotyping. *Methods Mol. Biol.* 578, 123–133.
- Royo, J.L., Hidalgo, M., Ruiz, A., 2007. Pyrosequencing protocol using a universal biotinylated primer for mutation detection and SNP genotyping. *Nat. Protoc.* 2, 1734–1739.
- Ruegger, P.M., Bent, E., Li, W., *et al.*, 2012. Improving oligonucleotide fingerprinting of rRNA genes by implementation of polony microarray technology. *J. Microbiol. Methods* 90, 235–240.
- Schatz, M.C., Delcher, A.L., Salzberg, S.L., 2010. Assembly of large genomes using second-generation sequencing. *Genome Res.* 20, 1165–1173.
- Seo, T.S., Bai, X.P., Kim, D.H., *et al.*, 2005. Four-color DNA sequencing by synthesis on a chip using photocleavable fluorescent nucleotides. *Proc. Natl. Acad. Sci. USA* 102, 5926–5931.
- Shendure, J., Porreca, G.J., Reppas, N.B., *et al.*, 2005. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* 309, 1728–1732.
- Srinivasan, S., Batra, J., 2014. Four generations of sequencing – Is it ready for the clinic yet? *J. Next Gener. Seq. Appl.*
- Srinivasan, S., Clements, J.A., Batra, J., 2016. Single nucleotide polymorphisms in clinics: Fantasy or reality for cancer? *Crit. Rev. Clin. Lab. Sci.* 53, 29–39.
- Stahl, P.L., Lundeberg, J., 2012. Toward the single-hour high-quality genome. *Annu. Rev. Biochem.* 81, 359–378.
- Steemers, F.J., Gunderson, K.L., 2007. Whole genome genotyping technologies on the BeadArray platform. *Biotechnol. J.* 2, 41–49.
- Teo, Y.Y., Inouye, M., Small, K.S., *et al.*, 2007. A genotype calling algorithm for the Illumina BeadArray platform. *Bioinformatics* 23, 2741–2746.
- Tobler, A.R., Short, S., Andersen, M.R., *et al.*, 2005. The SNPlex genotyping system: A flexible and scalable platform for SNP genotyping. *J. Biomol. Tech.* 16, 398–406.
- Tomkinson, A.E., Vijayakumar, S., Pascal, J.M., Ellenberger, T., 2006. DNA ligases: Structure, reaction mechanism, and function. *Chem. Rev.* 106, 687–699.
- Turcatti, G., Romieu, A., Fedurco, M., Tairi, A.P., 2008. A new class of cleavable fluorescent nucleotides: Synthesis and optimization as reversible terminators for DNA sequencing by synthesis. *Nucleic Acids Res.* 36, e25.
- Valouev, A., Ichikawa, J., Tonthat, T., *et al.*, 2008. A high-resolution, nucleosome position map of *C. elegans* reveals a lack of universal sequence-dictated positioning. *Genome Res.* 18, 1051–1063.
- Velasco, E., Infante, M., Duran, M., *et al.*, 2007. Heteroduplex analysis by capillary array electrophoresis for rapid mutation detection in large multiexon genes. *Nat. Protoc.* 2, 237–246.
- Venables, S.J., Mehta, B., Daniel, R., *et al.*, 2014. Assessment of high resolution melting analysis as a potential SNP genotyping technique in forensic casework. *Electrophoresis* 35, 3036–3043.
- Voelkerding, K.V., Dames, S., Durtschi, J.D., 2010. Next generation sequencing for clinical diagnostics-principles and application to targeted resequencing for hypertrophic cardiomyopathy: A paper from the 2009 William Beaumont Hospital Symposium on Molecular Pathology. *J. Mol. Diagn.* 12, 539–551.
- Voelkerding, K.V., Dames, S.A., Durtschi, J.D., 2009. Next-generation sequencing: From basic research to diagnostics. *Clin. Chem.* 55, 641–658.
- Wall, J.D., Tang, L.F., Zerbe, B., *et al.*, 2014. Estimating genotype error rates from high-coverage next-generation sequence data. *Genome Res.* 24, 1734–1739.
- Wittwer, C.T., Reed, G.H., Gundry, C.N., Vandersteen, J.G., Pryor, R.J., 2003. High-resolution genotyping by amplicon melting analysis using LCGreen. *Clin. Chem.* 49, 853–860.
- Wu, T.D., 2016. Bitpacking techniques for indexing genomes: I. Hash tables. *Algorithms Mol. Biol.* 11.
- Yorukoglu, D., Yu, Y.W., Peng, J., Berger, B., 2016. Compressive mapping for next-generation sequencing. *Nat. Biotechnol.* 34, 374–376.
- Yu, X.Q., Sun, S.Y., 2013. Comparing a few SNP calling algorithms using low-coverage sequencing data. *BMC Bioinform.* 14.
- Zhang, K., Zhu, J., Shendure, J., *et al.*, 2006. Long-range polony haplotyping of individual human chromosome molecules. *Nat. Genet.* 38, 382–387.

Genome-Wide Haplotype Association Study

Yongshuai Jiang and Jing Xu, Harbin Medical University, Harbin, China

© 2019 Elsevier Inc. All rights reserved.

Introduction

The single nucleotide polymorphism (SNP) is a concept in population genetics. It is a polymorphism of a single nucleotide at a specific chromosome position. A SNP locus is a specific chromosome location where the DNA base has two statuses (in most cases) in a population. With the development of high throughput SNP array technology (such as Affymetrix SNP array 6.0) (Nishida *et al.*, 2008) and whole-genome sequencing technology (such as second generation sequencing technology) (Kerstens *et al.*, 2009), the identification of millions (or tens of millions) of SNPs has facilitated the population genetics and medical genetics studies. With this background, the genome-wide association study (GWAS) has been developed and achieved great success (Welter *et al.*, 2014). Thousands of risk loci were successfully identified for complex diseases or phenotypes, such as breast cancer (Garcia-Closas *et al.*, 2013), type 2 diabetes (Cho *et al.*, 2011), and bone mineral density (Rivadeneira *et al.*, 2009). The GWAS focus on the association between single SNP locus and disease/phenotype. However, the single locus does not explain all the genetic risks. Therefore, the haplotype analysis is needed to identify the association between combination types of SNP alleles (multi-loci) and disease/phenotype, where a SNP allele is defined as the base status of one member of homologous chromosomes at a specific chromosome location. There are two alleles for a SNP locus in a population.

In the genome, a group of physically proximate SNPs often nonrandomly associate with each other to form a linkage disequilibrium (LD) block. For two SNP loci, the nonrandom association between SNP alleles at the two SNP loci is called LD. The human LD map for different human populations has been discussed in detail by the International HapMap Project (Altshuler *et al.*, 2010; Frazer *et al.*, 2007) and the 1000 Genomes Project (Abecasis *et al.*, 2010). The LD block may be the smallest heredity unit and is also known as a haplotype block, which is made up of a group of physically proximate SNPs. The SNPs in the LD block are in high LD with each other. A haplotype is defined as a collection of specific SNP alleles on a single chromosome. For k SNPs in a genomic region, if these SNPs are LE of each other, there will be 2^k haplotypes in the population. If the k SNPs are LD of each other, the types of haplotype will be less than 2^k . The genotype at a specific locus is defined as the combination of SNP alleles located on homologous chromosomes. We assume that there are two alleles at the locus: A and a. For an individual, there are three possible genotypes at the locus: AA (homozygote), Aa (heterozygote), and aa (homozygote). The relationships between SNP, allele, LD block, haplotype, and genotype can be found in Fig. 1. For k SNPs in a LD block region, the haplotypes will be less than 2^k . Currently, many genome-wide haplotype association studies have been successfully carried out for identifying the disease/phenotype related genes. For example, by using the genome-wide haplotype association studies strategy, Tregouet *et al.* identified the SLC22A3-LPAL2-LPA gene cluster (chromosome 6q26–q27) as a risk locus for coronary artery disease (Tregouet *et al.*, 2009), Lv *et al.* found that a haplotype TT (rs7090018 and rs2912759) at the gene FGFR2 locus had significant association with acute myeloid leukemia, and Sun *et al.* (2016) identified four oral squamous cell carcinoma related genes, that is, SERPINB9, SERPINE2, GAK, and HSP90B1. The genome-wide haplotype association study is an effective strategy for identifying susceptibility genes of complex diseases/phenotypes.

In general, there are six steps in genome-wide haplotype association study: Data preparation, quality control, LD block identification, haplotype frequency estimation, association test, and multiple testing corrections. Next, we will introduce the six parts in detail.

The Framework of Genome-Wide Haplotype Association Study

The framework of genome-wide haplotype association study includes the following steps (Fig. 2):

Data preparation

Phenotype data: The genome-wide haplotype association study supports both binomial phenotype (case-control) and continuous phenotype (quantitative).

Genotype data: The raw SNP genotypes can be obtained from the SNP array (such as Affymetrix SNP array 6.0) or whole-genome sequencing technology (such as second generation and third generation sequencing technology). SNP array can produce millions of SNPs, and sequencing can produce tens of millions of SNPs.

In general, the phenotype and genotype data can be shown as matrix in Fig. 3. We suppose that there are m samples and n SNPs. Then the genotype data consists of n rows (each row is a SNP) and $k + m$ columns (the first k columns are the basic information of SNPs, such as chr#, position and allele, and the last m columns are the genotypes of SNPs). The phenotype data consists of m rows (each row is a sample) and one column (phenotype).

Quality Control (QC)

In general, there were three commonly used criteria for filtering the raw SNP genotype data: (1) the percentage of individuals successfully genotyped > 75%, (2) Hardy – Weinberg equilibrium ($P > 0.001$), and (3) minor allele frequency (MAF, being the

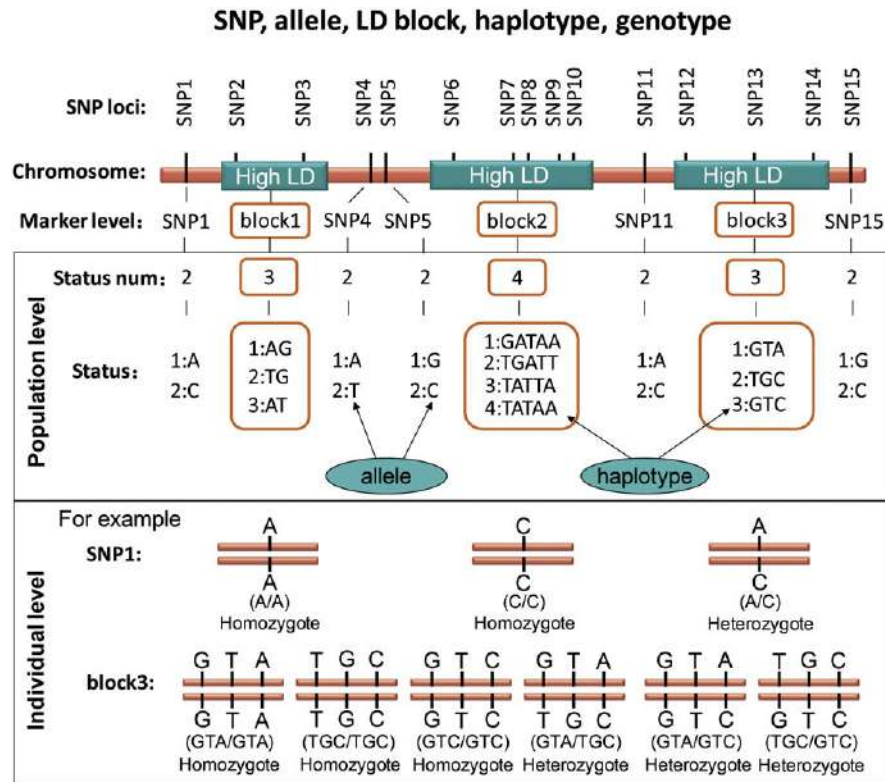


Fig. 1 The relationships between Single Nucleotide Polymorphism, allele, Linkage Disequilibrium block, haplotype, and genotype.

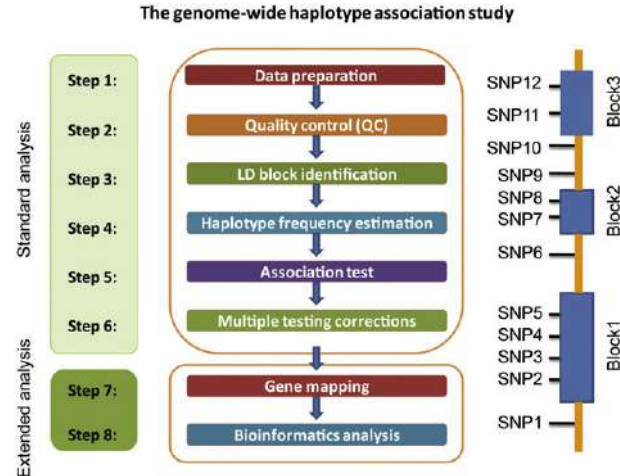


Fig. 2 The framework of genome-wide haplotype association study.

smaller of the two allele frequencies) $> 1\%$. These QC criteria ensure that the researchers can use high quality, stable, common SNPs (with the MAF > 0.01) to carry out the genome-wide haplotype association study.

Linkage disequilibrium block identification

The nonrandom association of alleles at different SNP loci is usually called LD (Slatkin, 2008). For two SNPs in LD, the frequency of association of their different alleles is higher or lower than random. A LD block (or haplotype block) is made up of a group of physically proximate SNPs that are highly LD to each other. In other words, the LDs between SNPs are higher within blocks and lower between blocks (Wang *et al.*, 2002).

The identification of the LD block is an important step in a genome-wide haplotype association study. There are many methods to identify LD blocks, such as Gabriel's confidence intervals method (Gabriel *et al.*, 2002), four-gamete test (FGT) (Wang *et al.*,

Genotype data

SNP	Chr#	position	Allele	Sample 1	Sample 2	Sample 3	...	Sample m
SNP 1	1	384531	A/T	AA	AA	AT	...	TT
SNP 2	1	396473	A/G	AG	AA	GG	...	AG
SNP 3	1	403251	C/G	CG	CC	CG	...	GG
...	...	...	...	...	...	...	...	...
SNP n-1	22	...	A/T	AT	TT	AA	...	TT
SNP n	22	...	A/C	AA	CC	AA	...	AC

Binomial phenotype (case-control)

sample	phenotype
Sample 1	Case
Sample 2	Case
Sample 3	Control
...	...
Sample m	Control

Continuous phenotype (quantitative)

sample	phenotype
Sample 1	1.78
Sample 2	1.92
Sample 3	1.69
...	...
Sample m	1.76

Fig. 3 The format of phenotype and genotype data.

2002), hidden Markov model (Daly *et al.*, 2001), and the dynamic programming algorithm (Zhang *et al.*, 2002). Although these algorithms can effectively identify haplotype blocks, they are still limited when millions or tens of millions of SNPs are encountered. This is because some blocks are often interrupted by limited computer memory or fixed size windows. To scan the whole-genome smoothly, Xu *et al.* (2016) developed an elastic sliding window method to avoid breaking the real block structure. Firstly, a fixed size sliding window was used to identify the LD blocks in the window. If the last locus in the window is not located in a block, the window was slid to the next locus. Otherwise, if the last locus in the window is located in an identified block, the block is likely to be broken by the border of the window. Then the window size was increased to twice the original fixed window size. The window size can be gradually increased until the last SNP is not in a block. By using this strategy, we can scan the entire genome and identify LD block using very little computer memory. Researchers can identify genome-wide LD blocks by using haplotype analysis software HAPLOTYPE 1.0 (see “Relevant Websites section”).

Haplotype Frequency Estimation

At present, neither SNP chip technology nor the second generation sequencing technology can directly detect haplotypes. This is because the gametic phase is ambiguous when individual genotypes are both heterozygous at two SNP loci. Therefore, some important algorithms, such as expectation-maximization algorithms (EM) (Barrett *et al.*, 2005), Bayesian methods (Scheet and Stephens, 2006), and Markov models (Delaneau *et al.*, 2013), have been developed to phase the haplotype and estimate the haplotype frequency. The phased haplotype can be obtained from the software SHAPEIT (Delaneau *et al.*, 2013) and fastPHASE (Scheet and Stephens, 2006), and the frequency of haplotype can be estimated from the software Haploview (Barrett *et al.*, 2005) and HAPLOTYPE 1.0.

Association Test

For the case-control study, the Pearson’s chi-square test can be used to identify the association between haplotype and phenotype. For a haplotype, we can estimate the haplotype frequency for case and control samples using methods mentioned above.

Then we can calculate the count of the haplotype in the case and control, and construct a fourfold table. The chi-square statistic measures for haplotype (this haplotype versus other haplotypes) and affection status (case versus control) can be found as following:

$$\chi^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

Researchers can also calculated the odd ratio and 95% (confidence interval).

For continuous or quantitative phenotype, researchers can identify the association between haplotype and phenotype using linear regression and variance analysis.

Multiple Testing Corrections

Genome-wide haplotype association study will scan all of the chromosomes, identify the LD blocks, and check the association between each haplotype and phenotype/disease. Therefore, the multiple testing is a challenging issue in genome-wide haplotype association study. Many techniques can be used to correct multiple hypotheses, such as the Bonferroni (Kowalski and Enck, 2010) and Sidak corrections (Guo and Romano, 2007), false discovery rate (Benjamini *et al.*, 2001), and permutation (Chen *et al.*, 2006). The simplest and strictest method of correction is the Bonferroni correction. For each individual hypothesis, researchers can obtain significant results by using a significance level of α/N , where α is the overall significant level and N is the number of hypotheses (the number of haplotypes). If the researchers think the threshold is too strict, they can choose a relatively loose threshold and use a bioinformatics strategy to filter the candidate genes (see Section “Applications”).

Applications

Identifying Complex Disease-Related Haplotype

One of the most important areas of application of the genome-wide haplotype association study is to identify the complex disease-related haplotypes. The disease-related haplotype can be classified into two categories: Risk haplotype and protective haplotype. A haplotype is labeled as risk (or protective) haplotype if the frequency of the haplotype in disease samples is higher (lower) than in control samples. The risk haplotype can increase the risk of disease, and the protective haplotype reduces disease risk.

Mapping Disease-Related Haplotype to Genes

Identifying the disease-related candidate gene is an important application for genome-wide haplotype association studies. The researchers can map the significant haplotypes to genes based on their chromosome location information. The information of start and end sites for all the human genes can be downloaded for the NCBI gene database (see “Relevant Websites section”). The SNP position can be downloaded from the NCBI SNP database (see “Relevant Websites section”). A notable problem is that the genome versions of gene and SNP must be consistent. A gene can be regarded as a candidate gene if the gene shares some chromosome fragment with a risk haplotype.

Analyzing the Gene Functions Using the Bioinformatics Strategy

After obtaining candidate genes, researchers can prioritize the candidate genes based on the correlation (such as sequence similarity and coexpression pattern) with known disease genes. The known disease genes can be obtained from some disease database, such as the Online Mendelian Inheritance (Amberger *et al.*, 2015). The software Endeavor can be used to calculate the similarity between the candidate genes and the known genes from several aspects, such as protein–protein interactions, gene expression, sequence information, text-mining, and gene annotation information (Tranchevent *et al.*, 2008).

Researchers can also analyze the gene functions using the gene annotation database, such as the GO (Gene Ontology) database (Blake and Harris, 2008) and KEGG pathway database (Kyoto Encyclopedia of Genes and Genomes) (Kanehisa and Goto, 2000). Some methods of gene enrichment analysis, such as hypergeometric test and enrichment disequilibrium (Jiang *et al.*, 2012), can be used to identify the disease-related pathways and gene functional categories.

The Haplotype Analysis Software

Here we briefly introduce the most commonly used haplotype analysis software. Using this software, researchers can successfully perform haplotype analysis.

HAPLOTYPE can carry out genome-wide haplotype association study using an elastic sliding window method. The software can be downloaded at the website provided in “Relevant Websites section”.

Haploview is one of the most commonly used haplotype analysis software programs. It supports single SNP and haplotype association tests for limited loci. The software can be downloaded at the website provided in “Relevant Websites section”.

Plink can scan the whole-genome and carry out the genome-wide haplotype association study. The software can be downloaded at the website provided in “Relevant Websites section”.

Shapit can quickly estimate the phase of haplotype. The software can be downloaded at the website provided in “Relevant Websites section”.

An Example

For better understanding the genome-wide haplotype association study, we describe an example. Previously, Wang *et al.* (2015) carried out a genome-wide haplotype association for 33 Chinese prostate cancer patients and 139 normal samples. All the samples

are unrelated individuals. A total of 632,532 SNPs on 22 autosomal chromosomes had passed QC criteria ($MAF > 1\%$, Hardy–Weinberg equilibrium $P > 0.001$, and call ratio $> 75\%$). Then they scanned the whole-genome using a FGT and identified 138,072 LD blocks. For each block, an EM algorithm was used to estimate haplotype frequencies for prostate cancer samples and normal samples. There were total 558,565 haplotypes in the 138,072 LD blocks. For each haplotype, a chi-square test was used to obtain the significant haplotypes. At last, 794 haplotypes were significantly associated with prostate cancer. These significant haplotypes were located in 694 LD blocks. To find the novel prostate cancer related gene, they mapped the significant haplotypes to genes based on physical position and obtained 454 candidate genes. Then Endeavor software was used to prioritize the candidate genes based on the similarity with 13 known prostate cancer genes. In the end, they identified seven novel susceptibility genes. Among these genes, the most significant gene is BLM in chr15. A haplotype GGTACCCCTC (rs2270131, rs2073919, rs11073953, rs12592875, rs16944863, rs2238337, rs414634, rs401549, rs17183344, rs16944884, and rs16944888) on the BLM gene region has significant association with prostate cancer ($P = 2.37E - 11$). The haplotype frequency is 0.16 in prostate cancer samples, whereas it is only 0.0004 in normal samples. The haplotype is a risk haplotype. For other genes, we will not describe them in detail. For more details, please see the research report (Wang *et al.*, 2015).

Discussion

The genome-wide haplotype association study is an important strategy for identifying diseases/traits related haplotypes (chromosome segments) and mapping candidate genes. However, there are still some technical limitations. The main reason is that current SNP detection techniques break the association between physically proximate SNPs. This directly leads to an unknown gametic phase when two SNPs are both heterozygous. Both SNP chip technology and second generation sequencing technology cannot solve this question. With the development of genome sequencing technology (perhaps the fourth or fifth generation sequencing technology), the haplotype will be sequenced directly rather than inferred, and haplotype analysis will become a hot issue.

In some cases, the researchers have only SNP genotype data for disease, and no control data. They can use some public genotype data in the same or similar population as control samples, such as the 1000 Genomes Project.

We will gradually share some learning materials and data on the HAPLOTYPE website provided in “Relevant Websites section”.

Conclusions and Future Directions

Genome-wide haplotype association studies are an important tool in medical genetics studies.

When the haplotype can be directly sequenced, genome-wide haplotype analysis will become a very important analytical tool. In addition, the genome-wide haplotype association study framework will possibly be extended to other omics, such as the epigenome. For DNA methylation, there was also nonrandom association between neighboring methylation loci (we called methylation disequilibrium, MD). A group of methylation status (M: Methylation or U: Unmethylation) on a chromosome can also be defined as a methylation haplotype (meplotype). The genome-wide haplotype association study could be extended to the epigenome-wide meplotype association study. Related concepts can be seen in the framework for population epigenetic study (Zhao *et al.*, 2018). The analysis program for meplotype and single methylation polymorphism (SMP) (Xu *et al.*, 2018) can be found at EWAS website provided in “Relevant Websites section”.

Acknowledgment

We thank Di Liu (Harbin Medical University, China) and Linna Zhao (Harbin Medical University, China) for their contributions to the figures and materials. This work was supported in part by the National Natural Science Foundation of China (Grant No. 91746113). The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

See also: Comparative Genomics Analysis. Detecting and Annotating Rare Variants. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Single Nucleotide Polymorphism Typing. Whole Genome Sequencing Analysis

References

- Abecasis, G.R., *et al.*, 2010. A map of human genome variation from population-scale sequencing. *Nature* 467, 1061–1073.
 Altshuler, D.M., *et al.*, 2010. Integrating common and rare genetic variation in diverse human populations. *Nature* 467, 52–58.

- Amberger, J.S., *et al.*, 2015. OMIM.org: Online Mendelian Inheritance in Man (OMIM(R)), an online catalog of human genes and genetic disorders. *Nucleic Acids Research* 43, D789–D798.
- Barrett, J.C., *et al.*, 2005. Haploview: Analysis and visualization of LD and haplotype maps. *Bioinformatics* 21, 263–265.
- Benjamini, Y., *et al.*, 2001. Controlling the false discovery rate in behavior genetics research. *Behavioural Brain Research* 125, 279–284.
- Blake, J.A., Harris, M.A., 2008. The Gene Ontology (GO) project: Structured vocabularies for molecular biology and their application to genome and expression analysis. *Current Protocols in Bioinformatics*. Chapter 7, Unit 7 2.
- Chen, B.E., *et al.*, 2006. Resampling-based multiple hypothesis testing procedures for genetic case-control association studies. *Genetic Epidemiology* 30, 495–507.
- Cho, Y.S., *et al.*, 2011. Meta-analysis of genome-wide association studies identifies eight new loci for type 2 diabetes in east Asians. *Nature Genetics* 44, 67–72.
- Daly, M.J., *et al.*, 2001. High-resolution haplotype structure in the human genome. *Nature Genetics* 29, 229–232.
- Delaneau, O., Zagury, J.F., Marchini, J., 2013. Improved whole-chromosome phasing for disease and population genetic studies. *Nature Methods* 10, 5–6.
- Frazer, K.A., *et al.*, 2007. A second generation human haplotype map of over 3.1 million SNPs. *Nature* 449, 851–861.
- Gabriel, S.B., *et al.*, 2002. The structure of haplotype blocks in the human genome. *Science* 296, 2225–2229.
- Garcia-Closas, M., *et al.*, 2013. Genome-wide association studies identify four ER negative-specific breast cancer risk loci. *Nature Genetics* 45, 392–398.
- Guo, W., Romano, J., 2007. A generalized Sidak-Holm procedure and control of generalized error rates under independence. *Statistical Applications in Genetics and Molecular Biology* 6, Article3.
- Jiang, Y., *et al.*, 2012. Enrichment Disequilibrium: A novel approach for measuring the degree of enrichment after gene enrichment test. *Biochemical and Biophysical Research Communications* 424, 563–567.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28, 27–30.
- Kerstens, H.H., *et al.*, 2009. Large scale single nucleotide polymorphism discovery in unsequenced genomes using second generation high throughput sequencing technology: Applied to Turkey. *BMC Genomics* 10, 479.
- Kowalski, A., Enck, P., 2010. Statistical methods: Multiple significance tests and the Bonferroni procedure. *Psychotherapie, Psychosomatik, Medizinische Psychologie* 60, 286–287.
- Nishida, N., *et al.*, 2008. Evaluating the performance of Affymetrix SNP Array 6.0 platform with 400 Japanese individuals. *BMC Genomics* 9, 431.
- Rivadeneira, F., *et al.*, 2009. Twenty bone-mineral-density loci identified by large-scale meta-analysis of genome-wide association studies. *Nature Genetics* 41, 1199–1206.
- Scheet, P., Stephens, M., 2006. A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *American Journal of Human Genetics* 78, 629–644.
- Slatkin, M., 2008. Linkage disequilibrium – Understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics* 9, 477–485.
- Sun, W., *et al.*, 2016. Genome-wide haplotype association analysis identifies SERPINB9, SERPINE2, GAK, and HSP90B1 as novel risk genes for oral squamous cell carcinoma. *Tumour Biology* 37, 1845–1851.
- Tranchevent, L.C., *et al.*, 2008. ENDEAVOUR update: A web resource for gene prioritization in multiple species. *Nucleic Acids Research* 36, W377–W384.
- Tregouet, D.A., *et al.*, 2009. Genome-wide haplotype association study identifies the SLC22A3-LPAL2-LPA gene cluster as a risk locus for coronary artery disease. *Nature Genetics* 41, 283–285.
- Wang, N., *et al.*, 2002. Distribution of recombination crossovers and the origin of haplotype blocks: The interplay of population history, recombination, and mutation. *American Journal of Human Genetics* 71, 1227–1234.
- Wang, Q., *et al.*, 2015. Genome-wide haplotype association study identifies BLM as a risk gene for prostate cancer in Chinese population. *Tumour Biology* 36, 2703–2707.
- Welter, D., *et al.*, 2014. The NHGRI GWAS catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research* 42, D1001–D1006.
- Xu, J., *et al.*, 2016. EWAS: Epigenome-wide association studies software 1.0 – Identifying the association between combinations of methylation levels and diseases. *Scientific Reports* 6, 37951.
- Xu, J., *et al.*, 2018. EWAS: Epigenome-wide association study software 2.0. *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/bty163>.
- Zhang, K., *et al.*, 2002. A dynamic programming algorithm for haplotype block partitioning. *Proceedings of the National Academy of Sciences of the United States of America* 99, 7335–7339.
- Zhao, L., *et al.*, 2018. The framework for population epigenetic study. *Briefings in Bioinformatics* 19 (1), 89–100.

Relevant Websites

<http://www.broadinstitute.org/haploview/>
BROAD INSTITUTE.

<http://www.ewas.org.cn>
EWAS.

<http://www.bioapp.org/ewas>
EWAS.

<ftp://ftp.ncbi.nlm.nih.gov/gene>
Gene - NCBI.

<http://www.haplotype.cn>
HAPLOTYPE1.0.

<http://www.cog-genomics.org/plink>
PLINK.

https://mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.html
SHAPEIT.

<ftp://ftp.ncbi.nlm.nih.gov/snp>
SNP - NCBI FTP Site.

Prediction of Protein-Binding Sites in DNA Sequences

Kenta Nakai, The University of Tokyo, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

In the nucleus of living cells, DNA molecules are specifically bound by a number of proteins to be functional. Of these, the most abundant ones are histones and their related proteins, which are important for organizing nuclear DNA into chromatin and for regulating their structure by epigenetic mechanisms. This topic is described in another article. Here we mainly deal with the binding of transcription factors (TFs) because it is critical in understanding how gene expression is regulated and has been extensively studied. The transcription-factor binding sites (TFBSs) are recognized as *cis*-elements in DNA sequences. Compared with a parallel problem of predicting DNA-binding regions in protein sequences or protein 3D structures, locating TFBSs in DNA sequences can be more difficult because the regions surrounding TFBSs are not necessarily similar between instances. In other words, (global) sequence alignment-based methods are not so effective, though so-called phylogenetic footprinting is effective when evolutionarily orthologous regions are compared (see below). Thus, a specialized field known as the motif-finding (or motif-discovery) problem has been formulated and studied well so far. In this problem, a set of DNA sequences, each of which is likely to contain at least one common TFBS, are given and these TFBSs which are recognized as an over-represented motif, i.e., a short sequence pattern with some statistical significance, are extracted. Typically, the input sequences are the potentially regulatory regions, such as the upstream 1000 bp-regions from the TSSs (transcriptional start sites), of co-expressed genes, i.e., genes which share a similar condition-specific expression and thus are likely to be regulated by a common mechanism, such as by a common transcription factor. As described below, many algorithms have been developed and their performances have been evaluated. More recently, the appearances of the NGS (next generation sequencing) technology, particularly the ChIP-seq (chromatin immunoprecipitation sequencing) technology, which enable to map the binding regions of a specified TF to the entire genome, have changed the situation of this field significantly. For one thing, the scale of the input sequences has been enlarged enormously. For another, a new need of algorithms for locating a TFBS in each ChIP-seq peak has emerged and algorithms specifically tailored for the ChIP-seq data analysis have been developed. In addition, demands for expanding the motif-finding problem to capture the more general architecture of gene regulatory regions have also emerged.

Many review articles have been written in this field. Of these, [Zambelli et al. \(2013\)](#) was written with a similar perspective with this article, though it is not so new. A more recent review by [Liu et al. \(2017\)](#) introduces recent advances, focusing on the algorithms for analysing ChIP-seq data.

How to Represent Motifs and the Problem of False Positives

The typical length of the binding sites of a monomeric TF is around 6 base pairs (bps) but the length of extracted *cis*-elements can become as long as 20 bps because TFs sometimes bind as a homodimer or a heterodimer. The homodimer binding motifs show the so-called palindromic feature and their central regions are sometimes close to random sequence, which can be regarded as a gap region. However, although the length of such 'gap's can even be varied, the length of sought TFBSs is practically assumed to be constant in motif-finding. In other words, variable gaps are not considered for aligning TFBSs in most cases.

Since the binding sequences of a TF are varied more or less, various ways have been used to represent its binding pattern. As described later, the way of how to represent the binding pattern (motif) is tightly related to the algorithms for finding it. Roughly speaking, there are two types of representing motifs: one is character-based and the other is numerical (profile-based).

The former character-based representation is related to the consensus-sequence representation, which shows the most frequent base at each position (in its simplest form). For example, the consensus sequence of the TATA-binding protein is known as 'TATAAT', though most of its known binding instances are not 'TATAAT'. It is well known that the frequency of base appearance at each position is not the same and thus sometimes bases that frequently appear are shown with capital letters while less frequent bases are shown in lower cases, such as 'TATaat'. When the majority of the preference of bases is shared with two bases, they are sometimes shown with parentheses, such as 'TATaa(t/a)'. The IUPAC code for representing degenerate bases is often used ([Table 1](#)). These character-based (or pattern-based) representation is a subset of regular expressions in computer science and is relatively easier to be treated in computers. In motif finding, the binding pattern of a TF is sometimes assumed to be represented as the so-called (l, d) -pattern, where l is the length of the pattern (e.g., 6) and d is the maximum number of allowed mismatches from the consensus (e.g., 3).

On the other hand, in the numeric or profile-based representation, the so-called positional weight matrix (PWM) is the standard (it is often referred as a profile). A PWM is a matrix, such as $\{w_{ij}\}$, $i=1,\dots,4$, $j=1,\dots,l$ for a motif with l -bases. The matrix element w_{ij} shows a score when base i appears at position j in the candidate instance. Typically, the score is proportional to the log-odds, which is the logarithm of the frequency of base i at position j in the set of known binding sequences for the TF divided by the background frequency of base i (for the base of the logarithm, 10 or 2 are often used). For each candidate binding site, the score at each position j is summed over the l positions, giving the score for the candidate. If this summed score exceeds a certain

Table 1 IUPAC codes for nucleotides

<i>Code</i>	<i>Base</i>
A	Adenine
C	Cytosine
G	Guanine
T (or U)	Thymine (or Uracil)
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
O	A or G or T
H	A or C or T
V	A or C or G
N	Any base

threshold parameter, the candidate is predicted to be one of the binding sites. Note that this method adds the logarithms of base frequency; this means that it essentially calculates the probability of observing motifs assuming that the base appearance at each position is independent each other. Practically, the observed frequency of base i can become 0 just because the size of the known set is not large. In such a case, candidates containing such a position will never be predicted to be positive. To avoid this, a small number of artificial occurrence is added to all the matrix elements. These values are called pseudo-counts. According to a study based on simulated data, a value 0.8 was recommended for practical uses (Nishida *et al.*, 2009).

Obviously, the PWM representation contains more information than the (l, d) -representation. In fact, the predicted sites with higher scores are not only more likely to be the true binding sites but also more likely that the TF binds more strongly. Nevertheless, both of these representations suffer from the problem of too many false positives. Namely, in both representations, the binding motif tends to match to too many positions in the genome, many of which are not the true binding sites. In this sense, the matrices stored in public transcription factor databases, such as JASPAR and TRANSFAC, are useful but should be used carefully. Indeed, our fundamental assumption that the short sequence motif is the determinant of specificity for any transcription factor is not accurate at least *in vivo*; exactly the same sequences can be bound at one site while unbound at another. Then, what kind of information should we add? We still do not have accurate answer; it is unlikely that the inherent assumption of positional independence in the PWM representation is the main reason because the incorporation of inter-position dependency does not seem to solve the problem (Zhang and Marr, 1993), though there are a few recent reports that the use of more flexible frameworks (e.g., HMM or deep learning) in motif representation improved the performance (Mathelier and Wasserman, 2013; Alipanahi *et al.*, 2015). Perhaps, the surrounding region with a relatively short range (say, $+/- 10$ bp) does not carry enough information. It is widely believed that the chromatin structure around the (relatively long-range) region is the key. Namely, if the surrounding chromatin is tightly packed and in an inactive state, the sites within it will not be bound but if it is in an open and active state, the sites within it will be more likely to be bound, though detailed confirmation has not been reported as far as the author notice. In this sense, phylogenetic conservation analyses of ChIP-seq data are useful to know what is happening. As an example, Schmidt *et al.* (2010) compared the binding profiles of CEBPA to the region around the PCK1 gene in the liver between five animals. There was a clear peak which is conserved between all five species at the 5' position of the gene but, interestingly, there was also a unique upstream peak which is human-specific, for example. It is not known whether this peak shows an evolutionary acquisition of a new functional element or just shows a randomly appeared, non-functional binding site. Thus, we should mind that even ChIP peaks may not show functional sites and that evolutionary conservation is an effective information to detect functionally important peaks (though not all functional sites are necessary to be conserved).

Algorithms of Motif Finding

In the motif-finding problem, the length of the motif sought is usually given as an input parameter. Each of the input sequence is often assumed to contain (at least) one target binding site (i.e., an instance) but practically, it would be all right, of course, if some of the input sequences do not contain target sites. Based on the representation methods for DNA motifs, the algorithms for motif finding are roughly-divided into two categories: the character-string (or word)-based methods and the numeric profile-based methods. Although there have been more preceding attempts based on word-based approach, a pioneering work which formalized the problem is done in Hertz *et al.* (1990), where they proposed a way to find a kind of PWM which is formed by the instances of input sequences and gives the highest sum of scores intuitively (the method thus belongs to the profile-based category and the improved algorithm is later released as CONSENSUS). Then, two classical algorithms, MEME (Bailey and Elkan, 1994) and Gibbs sampler (Lawrence *et al.*, 1993), which use the Expectation Maximization algorithm and the Gibbs

sampling algorithm, respectively, for optimizing the PWM, appeared. The basic idea of Gibbs sampler is first to randomly locate the instance position for each input sequence and make a PWM from them except one of them; then, the PWM is used to scan the excluded sequence; if another position gives higher score than the original instance. The instance of the sequence is updated with the probability determined by their score difference. Iterating this procedure many times, the PWM will happen to include target instances partially and will start to work efficiently to find the target instances in the remainder of sequences. Finally, the PWM will converge to the optimum one. A boom in this field had come after MEME and a number of algorithms have been developed. Practically, it is recommended to apply a motif finding algorithm with various parameter values and confirm these results by eyes. It would be even desirable to compare the results of multiple algorithms to confirm the stability of the extracted motifs (though there is no guarantee that frequently extracted motifs are more reliable). A tool has been released for helping such procedures (Okumura *et al.*, 2007).

As described above, in the character-string-based approach, the problem is formalized to efficiently enumerate motifs in the form of the (l, d) pattern (Pevzner and Sze, 2000). This formalism is appealing to researchers with the background of computer science and many techniques have been applied to reduce its computational cost. One successful example of this approach is Weeder (Pavesi *et al.*, 2001), which is proposed as a method to detect motifs with unknown length. In 2005, a relatively systematic evaluation of the performance of available algorithms was published, using various benchmark data (Tompkins *et al.*, 2005). Although there was not a single excellent algorithm for all purposes, the performance of Weeder was impressive.

However, with the appearance of the NGS (next generation sequencing) technology, including the ChIP-seq method, the situation has been changed. First, as noted above, more specialized software for the analysis of ChIP-seq data is required. Second and more importantly, the size of the input sequences has been enlarged enormously; for those large inputs, traditional software such as MEME and Weeder are not practical. Third, because the genome sequences of so many organisms have become available, comparison of orthologous regions between evolutionarily close organisms has become easier. Since the topic of phylogenetic footprinting is treated in another article, we do not describe about it further but just cite one pioneering work, which is even earlier than the establishment of the motif finding problem (Tagle *et al.*, 1988) as well as one recent progress (Liu *et al.*, 2016) in the field. Perhaps, the most popular tool for ChIP-seq motif finding is DREME by the author of MEME (Bailey, 2011). DREME belongs to the string (word)-based category and exhaustively finds top 100 words (of length l) which appear most significantly in the positive data compared with the negative data. Then, the top words are merged between similar ones to allow wild-card representation of a mismatch. DREME is also used in the MEME-ChIP web server, which integrates the two types of motif-finding algorithms, MEME and DREME (Machanick and Bailey, 2011), and it is included in a large collection of MEME-related tools, the MEME Suite (Bailey *et al.*, 2015; Fig. 1). The peak-motif tool in the RSAT (Regulatory Sequence Analysis Tools) web server, which is also well-known, belongs to the string-based class but the found motifs are compiled to the form of PWM (Thomas-Chollier *et al.*, 2012). As a pioneering example of numeric matrix-based approach for ChIP-seq peak detection, ChIPMunk, which is an iterative algorithm combining greedy optimization with bootstrapping, is famous (Kulakovskiy *et al.*, 2010). ChIPMunk has been used to construct a comprehensive collection of TF specificity of human and mouse TFs (HOCOMOCO; Kulakovskiy *et al.*, 2018). Another recent algorithm, DeepBind, employs a more general way of representing motifs than PWM and uses the deep learning algorithm, which is known to be quite powerful in various applications (Alipanahi *et al.*, 2015). So far, systematic benchmarking attempts for ChIP-seq motif finders have not been reported well (but see Wilbanks and Facciotti, 2010), probably because of the difficulty in setting correct binding motifs except trivial cases. However, it would be true that, generally speaking, more recently released algorithms are more likely to be better in their performance/speed and that the combined use of algorithms based on different principles are also recommended (for reviews, see Tran and Huang, 2014; Liu *et al.*, 2017, for example).

Variations of the Theme

One way to improve the sensitivity of motif detection is to improve the quality of background sequences, which are not random at all. For example, in the human genome, the frequency of the CpG dinucleotide is significantly fewer and many short repetitive sequence patterns exist. Markov-chain models have been used to capture these features. But there are also needs for more specific discriminative motif finding, where only motifs that significantly occur frequently in the positive dataset rather than the negative dataset are reported. For example, to extract motifs that are responsible for muscle-specific gene expression, ubiquitous signals, such as the TATA box, should not be extracted. Although the use of negative data has been attempted since very early years, the problem of discriminative motif finding can be a specific field to be explored (Redhead and Bailey, 2007; Huggins *et al.*, 2011). The need may have become even more important for the analysis of condition-specific changes of ChIP-seq peaks. And related to this, there is also a great need to extract a set of multiple non-redundant motifs that co-occur in a set of regulatory regions of co-expressed genes and this problem is regarded as the cofactor motif finding. Note that even ordinary motif finders can detect multiple motifs independently. For example, a rather simple algorithm, such as DREME, has been shown to be effective in finding cofactor motifs (Bailey, 2011). Of course, there have been algorithms which can detect co-occurring motifs simultaneously, such as SIOMICS (Ding *et al.*, 2015). This kind of approaches is also related to attempts to characterize the architecture (way of the placement of multiple TFBSs) in the upstream region of co-expressed genes. Some pioneering works were performed by Fickett and Wasserman (2000). Vandenberg and Nakai (2010) tried to systematically extract a set of rules on the combination and their relative placement of *de novo* motifs that can be used to distinguish a set of certain tissue-specific genes from the others. A similar approach was attempted to characterize the structure of *cis* regulatory modules that direct the

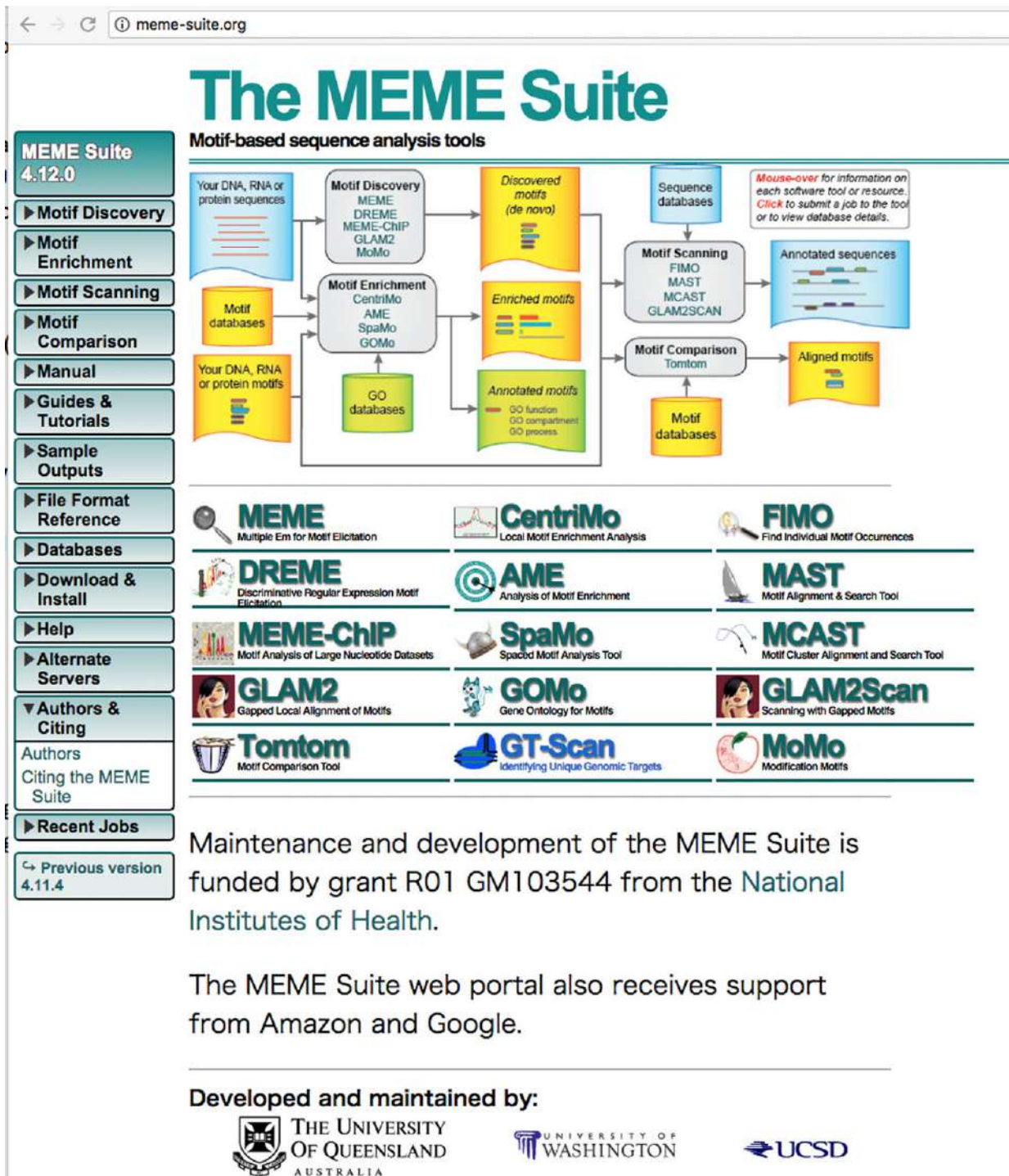


Fig. 1 The MEME Suite (<http://meme-suite.org/>).

developmental stage-specific expression of genes in *Drosophila* (Lopez *et al.*, 2017). With the advance of NGS technologies, such as the single-cell transcriptomics, this field should become more flourishing. An important remaining problem to be explored is to find motifs in enhancer regions. In spite of several large-scale experimental and computational studies, it is still a mystery on how transcription factors contribute to specify the enhancer activity (Kheradpour *et al.*, 2013). Systematic attempts to accumulate ChIP-seq and related data in several organisms, such as ENCODE, will offer a powerful foundation in the field (ENCODE Project Consortium, 2012). And future understanding on how the regulatory information is encoded in DNA would be undoubtedly useful in characterizing the relative importance of mutations/variations found in the non-coding regions of our genome, for example.

Concluding Remarks

Looking at the pace of publications in the field of motif-finding, this field has become to be regarded as a maturing field. The performance of the relevant software, however, is still not satisfactory at all. Especially, the problem of high false-positive rate must be overcome. As mentioned above, one important factor to be considered should be the chromatin states, which are closely related to the DNA looping structure, such as the TAD (topologically-associated domains; [Dixon *et al.*, 2016](#)). The chromatin states are predicted from a set of histone mark information, obtained by ChIP-seq. The DNA looping structure is also estimated with NGS-based technology, such as the Hi-C technology (reviewed in [Han *et al.* \(2018\)](#)). With the incorporation of such new advances, this field should experience another new boost in the near future.

Acknowledgement

This work was partly supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 17K00397.

See also: Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition

References

- Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J., 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 33, 831–838.
- Bailey, T.L., 2011. DREME: Motif discovery in transcription factor ChIP-seq data. *Bioinformatics* 27, 1653–1659.
- Bailey, T.L., Johnson, J., Grant, C.E. & Noble, W.S., 2015. The MEME Suite. *Nucleic Acids Res.* 43, W39–W49.
- Bailey, T.L., Elkan, C., 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 2, 28–36.
- Ding, J., Dhillon, V., Li, X., Hu, H., 2015. Systematic discovery of cofactor motifs from ChIP-seq data by SIOMICS. *Methods* 79–80, 47–51.
- Dixon, J.R., Gorkin, D.U., Ren, B., 2016. Chromatin domains: The unit of chromosome organization. *Mol. Cell* 62, 668–680.
- ENCODE Project Consortium, 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74.
- Fickett, J.W., Wasserman, W.W., 2000. Discovery and modeling of transcriptional regulatory regions. *Curr. Opin. Biotechnol.* 11, 19–24.
- Han, J., Zhang, Z., Wang, K., 2018. 3C and 3C-based techniques: The powerful tools for spatial genome organization deciphering. *Mol. Cytogenet.* 11, 21.
- Hertz, G.Z., Hartzell, G.W., Stormo, G.D., 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.* 6, 81–92.
- Huggins, P., Zhong, S., Shiff, I., *et al.*, 2011. DECOD: Fast and accurate discriminative DNA motif finding. *Bioinformatics* 27, 2361–2367.
- Kheradpour, P., Ernst, J., Melnikov, A., *et al.*, 2013. Systematic dissection of regulatory motifs in 2000 predicted human enhancers using a massively parallel reporter assay. *Genome Res.* 23, 800–811.
- Kulakovskiy, I.V., Boeva, V.A., Favorov, A.V., Makeev, V.J., 2010. Deep and wide digging for binding motifs in ChIP-Seq data. *Bioinformatics* 26, 2622–2623.
- Kulakovskiy, I.V., Vorontsov, I.E., Yevshin, I.S., *et al.*, 2018. HOCOMOCO: Towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Res.* 46, D252–D259.
- Lawrence, C.E., Altschul, S.F., Boguski, M.S., *et al.*, 1993. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* 262, 208–214.
- Liu, B., Yang, J., Li, Y., Mcdermaid, A., Ma, Q., 2017. An algorithmic perspective of de novo cis-regulatory motif finding based on ChIP-seq data. *Brief Bioinform.*
- Liu, B., Zhang, H., Zhou, C., *et al.*, 2016. An integrative and applicable phylogenetic footprinting framework for cis-regulatory motifs identification in prokaryotic genomes. *BMC Genomics* 17, 578.
- Lopez, Y., Vandenbon, A., Nose, A., Nakai, K., 2017. Modeling the cis-regulatory modules of genes expressed in developmental stages of *Drosophila melanogaster*. *PeerJ* 5, e3389.
- Machanick, P., Bailey, T.L., 2011. MEME-ChIP: Motif analysis of large DNA datasets. *Bioinformatics* 27, 1696–1697.
- Mathelier, A., Wasserman, W.W., 2013. The next generation of transcription factor binding site prediction. *PLoS Comput Biol.* 9, e1003214.
- Nishida, K., Friith, M.C., Nakai, K., 2009. Pseudocounts for transcription factor binding sites. *Nucleic Acids Res.* 37, 939–944.
- Okumura, T., Makiguchi, H., Makita, Y., Yamashita, R., Nakai, K., 2007. Melina II: A web tool for comparisons among several predictive algorithms to find potential motifs from promoter regions. *Nucleic Acids Res.* 35, W227–W231.
- Pavesi, G., Mauri, G., Pesole, G., 2001. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* 17 (Suppl. 1), S207–S214.
- Pevzner, P.A., Sze, S.H., 2000. Combinatorial approaches to finding subtle signals in DNA sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 8, 269–278.
- Redhead, E., Bailey, T.L., 2007. Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinform.* 8, 385.
- Schmidt, D., Wilson, M.D., Ballester, B., *et al.*, 2010. Five-vertebrate ChIP-seq reveals the evolutionary dynamics of transcription factor binding. *Science* 328, 1036–1040.
- Tagle, D.A., Koop, B.F., Goodman, M., *et al.*, 1988. Embryonic epsilon and gamma globin genes of a prosimian primate (*Galago crassicaudatus*). Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J. Mol. Biol.* 203, 439–455.
- Thomas-Chollier, M., Herrmann, C., Defrance, M., *et al.*, 2012. RSAT peak-motifs: Motif analysis in full-size ChIP-seq datasets. *Nucleic Acids Res.* 40, e31.
- Tomba, M., Li, N., Bailey, T.L., *et al.*, 2005. Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* 23, 137–144.
- Tran, N.T., Huang, C.H., 2014. A survey of motif finding Web tools for detecting binding site motifs in ChIP-Seq data. *Biol Direct*, 9, 4.
- Vandenbon, A., Nakai, K., 2010. Modeling tissue-specific structural patterns in human and mouse promoters. *Nucleic Acids Res.* 38, 17–25.
- Wilbanks, E.G., Facciotti, M.T., 2010. Evaluation of algorithm performance in ChIP-seq peak detection. *PLoS One*, 5, e11471.
- Zambelli, F., Pesole, G., Pavesi, G., 2013. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Brief. Bioinform.* 14, 225–237.
- Zhang, M.Q., Marr, T.G., 1993. A weight array method for splicing signal analysis. *Comput. Appl. Biosci.* 9, 499–509.

Genome-Wide Scanning of Gene Expression

Sung-J Park, The University of Tokyo, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Identifying genetic and epigenetic factors that impact molecular interplay and disease development is a high priority issue in the field of biological and medical sciences. As a solution to this issue, profiling of gene expression with microarray-based technologies has widely been adopted (Hoheisel, 2006). More recently, thanks to the improvement of next-generation sequencing (NGS) technologies, RNA sequencing (RNA-seq) has emerged as a new routine tool for the transcriptome profiling (Wang *et al.*, 2009). Beyond quantifying RNA population, RNA-seq facilitates to discover genome-wide alternative splicing isoforms, transcriptional start sites, epigenetic modifications, and allele specific transcripts. Currently, RNA-seq is gaining more popularity with growing demand of genome assays from communities.

It is often the case that billions of digital RNA fragments produced by NGS machines are computationally processed and statistically tested. Furthermore, to understand functional meaning of gene expression, incorporating multi-omics data that are rapidly increasing in quantity became inevitable. Thus, along with a well-designed experiment setting, successful transcriptome studies require to leverage bioinformatics and to carry out a pipeline that makes studies repeatable and transparent. In this regard, extensive community-wide efforts have been made to guide the effective usage of bioinformatics resources through developing online systems (Henry *et al.*, 2014; Cochrane *et al.*, 2016; Sloan *et al.*, 2016) and conducting benchmarks (Teng *et al.*, 2016; Williams *et al.*, 2016; Yan *et al.*, 2017). However, we frequently remain uncertain which background assumptions, statistical thresholds, or bioinformatics tools are appropriate to individual purposes (Fig. 1).

Herein, after succinctly briefing on the trends in NGS technology, we summarize recent progresses in scanning of genome-wide gene expression and in profiling of its functional importance. We focus on computational and statistical issues for filtering, mapping, quantifying, and annotating the RNA-seq outcomes. To help readers finding new opportunities and challenges, we provide a series of references that expedites understanding of each issue in detail. In addition, this article exemplifies a practical way of gene expression profiling with public data of spermatogenesis (Fig. 2(A)), which strives to avoid profound discussions on the germ cell development reviewed in excellent references (Bettegowda and Wilkinson, 2010; Saitou *et al.*, 2012; Bohacek and Mansuy, 2015; Ernst *et al.*, 2017).

Although we limit our scope of this review to transcriptome analysis, readers can also find attractive themes for processing of multi-omics data that gives a new multifaceted window to the identification of key factors. We expect that transcriptome atlases established by genome-wide scanning of gene expression provide fundamental resource to the extensive genomic studies that require the incorporation of large-scale heterogeneous datasets.

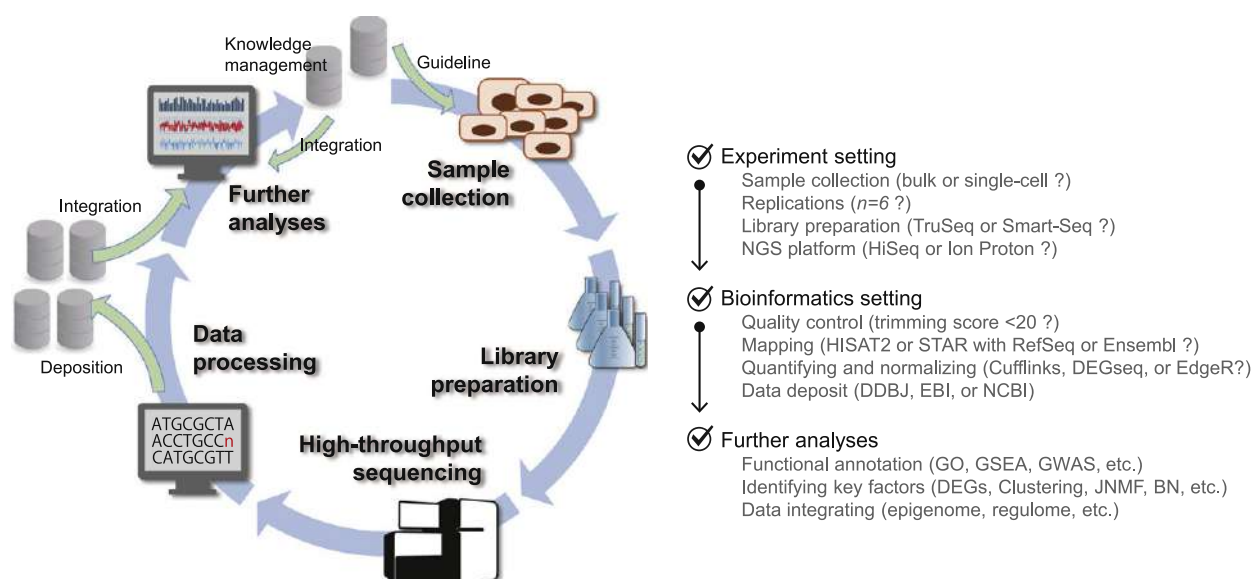


Fig. 1 Schematic representation of the routine scanning of genome-wide gene expression by NGS technology that requires appropriate setting of experiment and bioinformatics. Along with data sharing and integrating, an effective pipeline design is a critical facet in conducting NGS experiments. DEGs; Differentially Expressed Genes, JNMF; Joint Non-negative Matrix Factorization, BN; Bayesian Network.

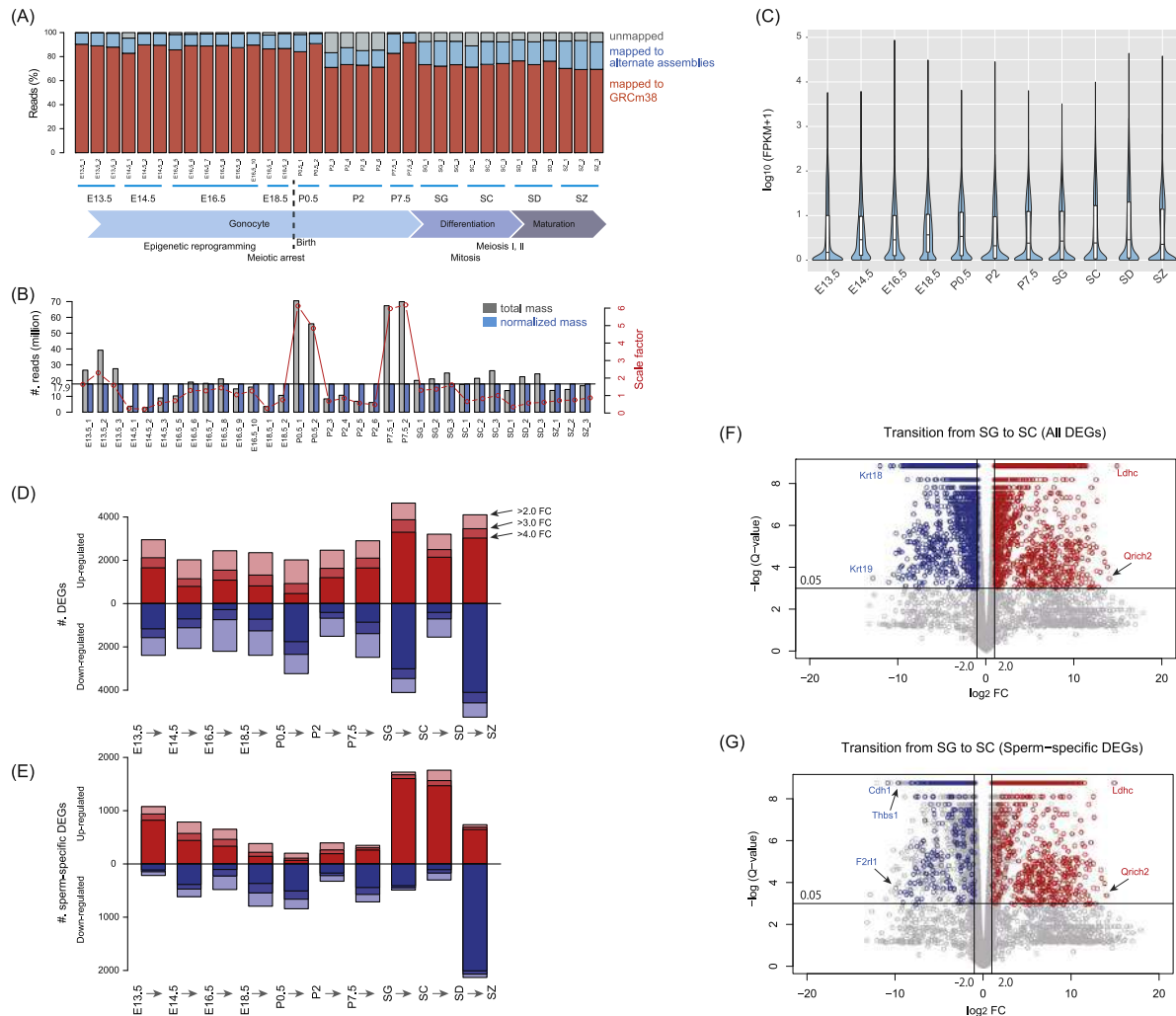


Fig. 2 An example of transcriptome analysis with HISAT2-Cufflinks2 pipeline. This example used public RNA-seq data of mouse spermatogenesis prepared at the time-points of embryonic day E13.5, E14.5, E16.5, E18.5, postnatal P0.5, P2, P7.5, Spermatogonia (SG), Spermatocytes (SC), Spermatids (SD), and Spermatozoa (SZ). (A) Schematic representation of male germ cell development, and read mappability of each replicate. (B) Normalized number of sequenced reads by scale factors. (C) Broad dynamic range of gene expression value FPKMs at each stage. (D) Differentially expressed genes (DEGs) in two consecutive stages (<0.05 FDR). (E) DEGs excluding genes expressed either MEF or Sertoli cells. (F) and (G) examples of volcano plot with Q-value and Fold-change (FC). FC is the FPKM ratio of SC/SG.

Genomic-Scale Transcriptome Profiling

Applications of High-Throughput Sequencing

The advent of massive parallel sequencing technology has led a new era of genome analysis, which can rapidly produce readouts of billions of DNA and RNA molecules. Since Sanger's success in sequencing a complete genome of virus (Sanger *et al.*, 1977), through a series of technical innovations (Mardis, 2013; Reuter *et al.*, 2015; Park *et al.*, 2016; Shendure *et al.*, 2017), the high-throughput sequencing also known as next-generation sequencing (NGS) is gaining more popularity with growing demand from academic and clinical communities (Koboldt *et al.*, 2013). Indeed, various consortium-based sequencing projects are ongoing towards specific goals (Table 1). They continuously manage the outcomes as public databases that provide new opportunities to unveil biological enigma.

The rapid progress in NGS technology has placed diverse features in commercialized machines (Loman *et al.*, 2012; Reuter *et al.*, 2015; Shendure *et al.*, 2017). For instance, Illumina platforms coupled with advanced experiment protocols, such as immunoprecipitation and enzyme reaction, produce relatively short genomic fragments (i.e., reads). On the other hand, synthetic long-read technologies, such as PacBio (Nakano *et al.*, 2017) and Oxford Nanopore Technologies (Loman and Watson, 2015), have emerged as a recent innovation of NGS technology. These NGS platforms often show trade-off features between pros and cons in throughput, cost per run, and signal-to-noise ratio (Mardis, 2013; Ross *et al.*, 2013; Reuter *et al.*, 2015). Thus, attention

Table 1 Example of consortium-based sequencing projects

<i>Consortium</i>	<i>Purpose</i>	<i>NGS type</i>	<i>Reference (PMID)</i>	<i>Website</i>
Cell Lines Project, Catalog of Somatic Mutations in Cancer (COSMIC)	Identifying genetic abnormalities from over 1000 human immortal cell lines	WES (Whole exome sequencing)	27727438	http://cancer.sanger.ac.uk/
Exome Aggregation (ExAC)	Cataloging human genetic diversity by high-quality exome sequencing data	WES	27535533	http://exac.broadinstitute.org/
Genetic European Variation in Health and Disease (GEUVADIS)	Understanding human transcriptome variation associated with disease-related genetic variation	RNA-seq	24037378	http://www.geuvadis.org/
Genotype-Tissue Expression (GTEx)	Characterizing human gene expression and regulation and its relationship to genetic variation	WGS (Whole genome sequencing), WES, RNA-seq	29022597	https://www.gtexportal.org/home/
International Wheat Genome Sequencing Consortium (IWGSC)	Comprehensive assembly and understanding of genetic features of bread wheat	WGS	25035484	https://www.wheatgenome.org/
Non-Human Primate Reference Transcriptome Resource (NHPRTR)	Cataloging transcriptome landscapes of over 20 tissues in non-human 14 species	RNA-seq	25392405	http://nhprtr.org/
The functional annotation of the mammalian genome 5 (FANTOM5)	Characterizing mammalian promoter features and regulatory elements	CASE-seq, RNA-seq	24670764	http://fantom.gsc.riken.jp/5/
4D Nucleosome Network	Characterizing four-dimensional nuclear architecture and its role in gene expression and cellular function	Hi-C, ATAC-seq, Repli-seq	28905911	https://data.4dnucleome.org/

needs to be paid to the choice of NGS platforms for maximizing the impact on genome sequencing projects, even though it is no doubt that their pervasive utilization will widely spread in communities.

The short-read technologies are frequently used to interrogate broad genome profiling; that is, polymorphism analysis by whole-genome and whole-exome sequencing (Lam *et al.*, 2012; Sudmant *et al.*, 2015), profiling of transcription factor (TF) occupancy (Park, 2009; Kidder *et al.*, 2011), profiling of genomic-scale transcriptome (Wang *et al.*, 2009). The NGS applications can also capture epigenetic variabilities, including DNA and RNA modifications (Laird, 2010; Frye *et al.*, 2016), chromatin accessibility (Buenrostro *et al.*, 2013), and chromosome 3D organization (Paulsen *et al.*, 2014; Ay and Noble, 2015). Although computationally intense, the long-read technologies demonstrate prominent performances in analyzing highly intricate genetic features (Sahraeian *et al.*, 2017); e.g., assembly of high-complex genome regions (McCoy *et al.*, 2014; Li *et al.*, 2015), detection of structural variations (Chaisson *et al.*, 2015), and identification of gene isoforms (Cho *et al.*, 2014; Xu *et al.*, 2015; Weirather *et al.*, 2017).

Profiling Transcriptome Landscape by RNA Sequencing

As a standard application of NGS, RNA sequencing (RNA-seq) enables the broad investigation of protein coding and non-coding transcripts on genomic scales. In this regard, microarray-based technology has been widely used (Hoheisel, 2006; Shi *et al.*, 2006), and is continuously improving its performance in (dis)concordant with RNA-seq outcomes (Marioni *et al.*, 2008; Wang *et al.*, 2014; Nazarov

et al., 2017). Although we expect synergistic and complementary effects of these two assays, RNA-seq offers comprehensive catalog of diverse RNA species, and depicts their transcriptomic and epitranscriptomic landscapes (Wang *et al.*, 2009; Ozsolak and Milos, 2011a; Helm and Motorin, 2017). In fact, recent large-scale RNA-seq projects have revealed remarkable biological characteristics, which includes pervasive transcriptomic feature (Djebali *et al.*, 2012), ingenious promoter usage (Yamashita *et al.*, 2011; Forrest *et al.*, 2014), disease-related transcripts (Weinstein *et al.*, 2013), and dynamic transcriptome waves (Park *et al.*, 2013).

In general, RNA-seq experiments first convert target RNAs into cDNAs that are further amplified. Then, they sequence the amplicons as reads that are computationally mapped on a reference genome. Consequently, the total number of reads mapped to single gene gives the degree of its expression. Of note, various alternative protocols have been developed, such as cDNA-conversion-free RNA-seq (Ozsolak and Milos, 2011b) and amplification-free RNA-seq (Mamanova *et al.*, 2010; Loman and Watson, 2015). Conventionally, RNAs are extracted from bulk cells by considering the cell homogeneity (Park *et al.*, 2013), giving averaged gene expression. Recently, RNA extraction from single cell has emerged (i.e., scRNA-seq) to address key biological issues (Tang *et al.*, 2009; Wang and Navin, 2015) including heterogeneous cell responses that are markedly observed in cancer diseases (Kim *et al.*, 2015), stem cells (Wen and Tang, 2016), and immune systems (Kadoki *et al.*, 2017; Papalexi and Satija, 2018).

Prior to the establishment of transcriptome profiles, researchers have to correct sequencing biases and errors that may occur due to the intrinsic feature of NGS applications (Li *et al.*, 2014; SeqC_MaQC-II Consortium, 2014). To do so, guidelines suggested by the community emphasize many technical issues, such as the importance of multiple biological and technical replicates (Schurch *et al.*, 2016; Merino *et al.*, 2017), the impact of library preparation methods (Parekh *et al.*, 2016), the influence of reference genomes and annotations on downstream interpretation (Hardwick *et al.*, 2017), and the prevention of microbial contaminations (Strong *et al.*, 2014; Olarerin-George and Hogenesch, 2015). Although some issues appear to be highly complex, bioinformatics tools will greatly help the error detection and correction through practical computation and mathematics (Greenfield *et al.*, 2014; Finotello and Di Camillo, 2015; Conesa *et al.*, 2016; Pimentel *et al.*, 2017). One can find a list of related bioinformatics software from the web site of OMIC Tools (Henry *et al.*, 2014).

Bioinformatics for Transcriptome Analysis

Once genomic fragments were determined as reads, bioinformatics further analyzes these digital outcomes for answering biological questions. Thus, along with a well-designed experiment setting, successful NGS studies require to leverage bioinformatics tools and to carry out a pipeline that makes researches repeatable and transparent (Conesa *et al.*, 2016; Kulkarni and Frommolt, 2017). In RNA-seq experiments, the components of a typical pipeline consist of read quality control, mapping to a reference genome, and expression quantification.

The quality control (QC) of reads trims and discards erroneous bases. Since ambiguous base calling leads to misalignments and therefore misinterpretation, the base-call error probability defined as Phred score (Ewing and Green, 1998) is indispensable information. Even if reads include high-quality bases, some of the bases are potentially adapter-origin. For dealing with this issue, many solutions have been proposed; in particular, FastQC (see “Relevant Websites section”), Prinseq (Schmieder and Edwards, 2011), Cutadapt (see “Relevant Websites section”), FASTX-Toolkit (see “Relevant Websites section”), and Trimmomatic (Bolger *et al.*, 2014). Importantly, it should be known that the QC process affects gene expression profile (Williams *et al.*, 2016), which indicates the necessity of scrupulous care.

Spliced mapping tools (Dobin *et al.*, 2013; Kim *et al.*, 2013; Pertea *et al.*, 2016) align RNA-seq reads to a reference genome that includes gene annotations (e.g., exon-exon junctions). This reference-guided approach is helpful to minimize algorithmic biases associated with particular tools. The tools are also applicable to reconstructing transcripts without any reference annotations. This *de novo* transcriptome assembly is a powerful approach for detecting unknown transcripts as well as for novel organisms (Grabherr *et al.*, 2011; Xie *et al.*, 2014). It should be noted that some portion of total sequenced reads are not mapped to the reference genome but mapped to its alternate assemblies and contigs that have been unplaced in building the reference genome. For instance, Fig. 2(A) exhibits 66.8%–91.6% mappability with the mouse reference genome GRCm38 (2647.54 Mb in size), and shows that 8.1%–24.1% in total reads are mappable to the alternate sequences of GRCm38 downloadable from public databases (8373.72 Mb in size). This underlines the possibility that the reference-guided approach misses one fourth of transcriptome in a worse case, which requires further investigation on its impact.

To quantify gene- or transcript-level expression, the mapped reads are rigorously addressed in terms of read depth and transcript-width coverage. Since the short reads are results from random cutting of RNA molecules, the transcript length and total number of sequenced reads affect the quantification (Conesa *et al.*, 2016). Therefore, normalization within a sample as well as among samples is an essential process that is still actively developed (Mortazavi *et al.*, 2008; Trapnell *et al.*, 2010; Risso *et al.*, 2014; Teng *et al.*, 2016; Jin *et al.*, 2017). The units frequently used in the normalization include FPKM (Fragments Per Kilobase of transcript per Million mapped reads), RPKM (Reads Per Kilobase of transcript per Million mapped reads), TPM (Transcripts Per Million), CPM (Counts-Per-Million), and PPM (Parts Per Million).

For example, HISAT2-Cufflinks2 pipeline (Trapnell *et al.*, 2012; Pertea *et al.*, 2016) first merges all the transcript assemblies coming from multiple replicates of each sample. Then, Cufflinks package counts mapped reads incorporated with the merged assembly, and scales the counts using the median of the geometric means of fragment counts across all samples (Anders and Huber, 2010). In the case of Fig. 2, the total read number of each replicate was adjusted to 1.78725×10^7 by a formula with each scale factor (Fig. 2(B)), which yields the broad dynamic range of gene expression value FPKMs that makes easy to detect differentially expressed genes (Fig. 2(C)).

Functional Profiling With Transcriptome Data

Identifying Differentially Expressed Genes

To scan biologically relevant genes, further investigation with a transcriptome profile may focus on categorizing gene expression changes. In this regard, the identification of differentially expressed genes (DEGs) between two samples is one of the most frequently used approaches, which employs a statistical model for the distributional approximation among biological replicates of the two samples. The statistical models, such as Poisson distribution, negative binomial distribution, Bayesian, and bootstrapping, have been embedded in a large number of tools (Marioni *et al.*, 2008; Hardcastle and Kelly, 2010; Robinson *et al.*, 2010; Wang *et al.*, 2010; Bray *et al.*, 2016; Dong *et al.*, 2016; Pimentel *et al.*, 2017). On the other hand, recent systematic benchmarks for these tools indicate that hidden factors underlying statistical assumption affect reproducibility and false discovery rate (FDR) in the DEG detection (Schurch *et al.*, 2016; Teng *et al.*, 2016; Jin *et al.*, 2017; Merino *et al.*, 2017; Pimentel *et al.*, 2017).

As an example, Fig. 2(D) shows the number of DEGs in consecutive stages, and Fig. 2(E) shows a subset of the DEGs excluding genes expressed (> 3.0 FPKM) either in MEF or Sertoli cells, which indicates sperm-specific transcriptomic bursts at the mitotic and meiotic stage SG, SC, and SD (Margolin *et al.*, 2014). Cuffdiff (Trapnell *et al.*, 2012) defined these DEGs using two empirical thresholds; FPKM fold-change (FC) and Q-value. The Q-value denotes FDR in which Benjamini-Hochberg correction adjusted P-value of two-group t-test (Storey and Tibshirani, 2003). Volcano plot visualizing the effect of two thresholds is typically used to identify the most meaningful DEGs (Fig. 2(F) and (G)).

Grouping Gene Expression Patterns

Given a transcriptome profile from multiple cells and/or multiple time points, genes and transcripts often exhibit dynamic expression waves that DEGs can hardly explain. To decompose the profile into expression waves, bioinformatics of so-called spatio-temporal gene expression analysis employs various computational algorithms including principal component analysis, matrix decomposition, t-distributed stochastic neighbor embedding (t-SNE) analysis, and network analysis (Domazet-Lošo and Tautz, 2010; Shapiro *et al.*, 2013; Hashimshony *et al.*, 2015; Chou *et al.*, 2016; van Dam *et al.*, 2017).

As a straightforward method, clustering analyses clearly show gene groups with similar expression patterns (Eisen *et al.*, 1998; Oyelade *et al.*, 2016). For instance, to prepare a clustering heatmap (Fig. 3(A)), FPKMs shown in Fig. 2(C) formatted a matrix Genes $\times$ Stages, and the matrix was converted into relative values by quantile normalization and matrix mean centering. Then, a clustering method grouped the matrix rows using Euclidian distance metric and hierarchical average linkage method (Fig. 3(A)). The method also grouped the matrix columns using Pearson's correlation coefficient, showing (dis)similarity of gene expression patterns among the developmental stages (Fig. 3(B)). As a result, one can trace gene clusters that may share biological functions (Fig. 3(C)). Of note, the clustering considered relative values without FPKM trimming and cutoff. Thus, absolute FPKMs of genes in a cluster might vary in degree even if their expression dynamics are highly similar. For example, Cluster 2 exhibits repressive FPKM distribution compared to Cluster 9 (Fig. 3(C)); 51.6% of genes in Cluster 2 (1503/2914) are < 3.0 FPKM at E14.5. Consequently, expression filtering with a threshold may affect further interpretation.

Annotating Functional Importance of Genes

In order to reveal functional meaning of key genes, bioinformatics tools search functional genome databases with a gene list, and subsequently perform statistical tests to infer whether any biological annotations are enriched in the genes compared to a background gene set. The approaches include Gene Set Enrichment Analysis (GSEA), Gene Ontology (GO) enrichment analysis, and pathway signature analysis (Subramanian *et al.*, 2005; Huang da *et al.*, 2009; Hung *et al.*, 2012; Yu *et al.*, 2016; Kanehisa *et al.*, 2017). Remarkably, all of the methods exploit knowledge repositories that have become indispensable for interpreting biological functions. Therefore, opening and sharing functional annotations in a standardized data format are increasingly important.

A difficult consideration in the enrichment analysis is to reduce FDR by controlling a balance between false positives (type I error, incorrectly inferring "enriched" from truly "not enriched" results) and false negatives (type II error, incorrectly inferring "not enriched" from truly "enriched" results) (Storey and Tibshirani, 2003; Curtis *et al.*, 2005). Also, since repeat enrichment tests for a gene set with different biological terms increase the change of observing false positives, researchers might adjust raw P-values by such as Benjamini-Hochberg procedure and Bonferroni correction (Storey and Tibshirani, 2003; Schwartzman and Lin, 2011). However, the multiplicity correction needs an attention due to leading to contradictory results with higher type II error rate, which is still under argument along with the issue of what P-value threshold is appropriate to enrichment decisions (Perneger, 1998; Sterne and Davey Smith, 2001).

In Fig. 3(A), among 16 clusters that the number of member genes is > 100 , ten clusters showed GO terms enriched by the tool of GOC (Gene Ontology Consortium, 2015), which indicates the involvement of common or unique biological processes (Fig. 3(D)). This enrichment analysis conducted Bonferroni correction, and picked up GO terms only if < 0.001 FDR. When other options for statistical test were used, the results presented broader GO term enrichment including the result in Fig. 3(D). Interestingly, genes in Cluster 8 transiently up-regulated at SG (Spermatogonia) are significantly involved in cell differentiation processes; at this stage, SG undergo mitotic proliferation and maturation (Ernst *et al.*, 2017). Also, Cluster 2 showing transient gene expression at E14.5 is

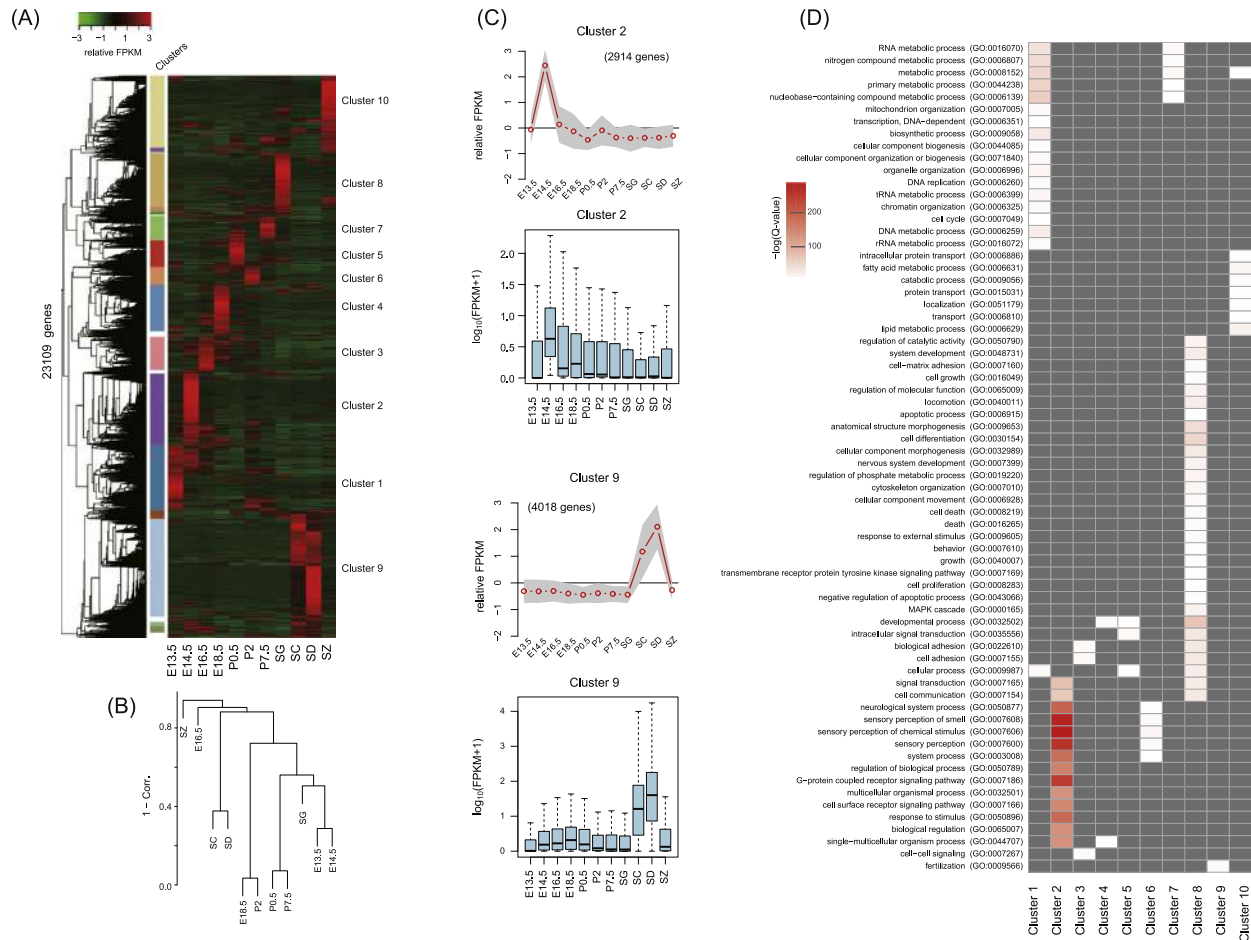


Fig. 3 An example of clustering analysis of transcriptome complexity. (A) Hierarchical clustering retrieving 16 clusters in which the number of genes in a cluster is over 100; among the clusters, 10 clusters showed significant GO term enrichment. (B) Dendrogram showing (dis)similarity of gene expression patterns among samples based on the distance of Pearson's correlation coefficient. (C) Examples of relative and absolute expression patterns; red line stands for the mean of relative FPKMs, and boxplot shows corresponding absolute FPKMs. (D) Result of GO term (biological process) enrichment analysis using Q-value of Bonferroni correction (< 0.001).

involved in olfactory sensory activity. Indeed, over 500 olfactory receptors (ORs) are located in this cluster, and the genes without weakly expressed (< 3.0 FPKM) still exhibited the enrichment of term “sensory perception of smell” (GO:0007608); FDR = 4.38×10^{-10} . Although ORs play importance roles in spermatogenesis (Fukuda and Touhara, 2006), unlike ORs highly expressed in Cluster 6, this activity might be related to epigenetic reprogramming; at E14.5 and E15.5, DNA methylation is globally erased and rapidly re-acquired (Bohacek and Mansuy, 2015).

Integrating Transcriptome With Multi-Omics Data

As ENCODE has illustrated (Djebali *et al.*, 2012), complex intra- and inter-molecular crosstalk is observed among regulatory elements, which indicates the necessity of projection of other omics data (e.g., epigenomics, proteomics, and metabolomics) onto transcriptome data. Extensive efforts have therefore been made for emerging multi-omics importance that can be manifested by various NGS assays, and have expedited the understanding of a wide range of biological phenomena (Hasin *et al.*, 2017; Li *et al.*, 2018; Lowe *et al.*, 2017). For example, ICGC and GTEx consortia have released multi-omics datasets in unprecedented scale, facilitating studies in the association of molecular signatures and phenotypes across large number of different cancer types and tissues (International Cancer Genome Consortium *et al.*, 2010; GTEx Consortium, 2017; Li *et al.*, 2017).

We expect that biases led from the heterogeneous datasets introduce highly confounding results with low reproducibility. Therefore, to integrate the multi-omics data by reducing unfavorable effects is a hot topic in bioinformatics, and a line of researches has proposed advanced computational algorithms (Gomez-Cabrero *et al.*, 2014; Huang *et al.*, 2017). These include matrix factorization (Zhang *et al.*, 2012), graph- or kernel-based methods (Yan *et al.*, 2017), and machine learning approaches (Li *et al.*, 2018; Chaudhary *et al.*, 2018). In regard to the association between transcriptome and a set of regulatory elements

(i.e., regulome), reverse engineering algorithms (Bussemaker, 2006) have been successfully applied in broad researches to infer key regulatory elements (Ouyang *et al.*, 2009; Park *et al.*, 2014; Sasamoto *et al.*, 2016). These algorithms predict a given gene expression profile with features from regulome data as a set of explanatory variables. If the removal test of a variable causes large variance in the predictive model, we should consider that the variable is indispensable factor for regulating the gene expression, which provides clues into further investigations.

Particular regulome data increasingly accumulated in the public domain include TF occupancy, histone modification, DNA methylation, and DNA accessibility. More recently, NGS data capturing higher-order chromatin organization became another omics data showing cooperativity between enhancers and promoters (Dixon *et al.*, 2015; Schwarzer *et al.*, 2017). These have been devoted to the development of databases freely accessible; for example, GTRD (see “Relevant Websites section”), ChIP-Atlas (see “Relevant Websites section”) and Cistrome DB (see “Relevant Websites section”). Meta-analysis of the databases further extends the empirical information of TF binding sites (TFBSs) managed in such as JASPAR (Vlieghe *et al.*, 2006), HOCOMOCO (Kulakovskiy *et al.*, 2013) and TRANSFAC (Wingender *et al.*, 2000). The qualified TFBSs can be extensive resources for online analytical tools (Ho Sui *et al.*, 2005; Okumura *et al.*, 2007; Contreras-Moreira and Sebastian, 2016).

Outlook and Conclusions

Comprehensive cataloging and quantifying RNA population with RNA-seq technique have convinced communities to conduct routine gene expression profiling, giving new opportunities to gain deeper insights into complex molecular interplay and disease development. It is likely that technical innovation in NGS will continue, and RNA-seq will gain more popularity to investigate transcriptome complexity. In parallel, the pervasive utilization of RNA-seq poses significant challenges to the communities, including the management of huge data and knowledge, quantitative and qualitative assessment of outcomes, and standardization of analytical protocols ensuring reproducibility and transparency. Bioinformatics has emerged to deal with these challenges, and recent RNA-seq experiment setting embeds them as an analytical pipeline.

We have described a range of issues in genome-scale scanning of gene expression associated with the intrinsic nature of experimental and computational methods. Although recent community-wide efforts have increasingly guided the effective usage of bioinformatics resources, some of issues still need intensive discussions. This includes statistical designs along with appropriate thresholds; the example of transcriptome analysis demonstrated the influence of FPKM cutoff, thresholds of FC and FDR, and multiplicity correction. Such empirical thresholds directly impact on filtering, mapping, quantifying, and annotating sequenced reads, which indicates the necessity of careful consideration.

We expect future bioinformatics for issues facing the multi-omics data integration to profile the biological functions of key genes that transcriptome analysis detected. Furthermore, continued data opening and sharing are expected for more comprehensive characterization of broad biological consequences; as a line of sequencing projects has instructed, the open innovation coupled with Big Data paradigm will provide valuable resources for systematic reciprocal linking of multiple regulatory layers, and also for linking genotypes to phenotypes. Therefore, effective bioinformatics methods for the integration approaches are becoming more indispensable. Finally, we believe that researchers will actively conduct open sharing of gene expression atlases established by tackling the issues mentioned here, and the atlases will propel genome analysis studies in biological science and clinical setting.

Materials and Methods

Data Preparing

Illumina SRA files for 34 RNA-seq samples were prepared from NCBI GEO; embryonic day E13.5 (GSM1064457), E14.5 (ERP015330), E16.5 (ERP015330), E18.5 (ERP015330), postnatal P0.5 (DRP002386), P2 (ERP015330), P7.5 (DRP002386), Spermatogonia; SG (GSM1069640), Spermatocytes; SC (GSM1069641), Spermatids; SD (GSM1069642), Spermatozoa; SZ (GSM1069643), Mouse embryonic fibroblast; MEF (GSM1643254), and Sertoli; SL (GSM1069639). SRA toolkit (ver. 2.8.2–1) converted the SRA files into FASTQ files by using options “fastq-dump -split-files -A accession”.

Rna-Seq Pipeline

Step1 (Quality control): Trimmomatic (ver. 0.36) assessed all the FASTQ files by using options “ILLUMINACLIP:adapter_file:2:30:10 LEADING:20 TRAILING:20 MINLEN:36”.

Step2 (Mapping): HISAT2 (ver.2.0.5) coupled with Bowtie2 (ver. 2.2.9) mapped the reads of each replicate to the mouse reference genome GRCm38 with options “-k 1 -dta -dta-cufflinks”, resulting 34 BAM files. After gathering unmapped reads from the BAMs by Samtools (ver. 1.5) with options “view -bf 4”, Blastn (ver. 2.2.31) aligned the unmapped reads by using options “-evalue 0.001 -perc_identity 80 -max_target_seqs 1” and a pre-built database with alternative assemblies and contigs downloaded from <ftp://ftp.ensembl.org/pub/> and <ftp://ftp.ncbi.nlm.nih.gov/blast/db/>.

Step 3 (Assembling and Quantifying): Cufflinks (ver. 2.2.1) with options “-total-hits-norm -b GRCm38.fa -G RefSeq(release 80)” assembled transcripts of each BAM, resulting 34 GTF (gene feature format) files. Cuffcompare (ver. 2.2.1) combined the 34

GTFs by using options “-C -T -R -r RefSeq(release 80) -s GRCm38.fa”. Cuffquant (ver. 2.2.1) with options “-b GRCm38.fa Combined_GTF” quantified each BAM and created 34 CBX files. Finally, Cuffdiff (ver. 2.2.1) profiled FPKMs and DEGs from 34 CBXs by using options “-FDR 0.05 -library-norm-method geometric -compatible-hits-norm Combined_GTF All_CBX_files”.

Bioinformatics Analysis

Clustering of gene expression performed R language function hclust; Euclidean distances among FPKMs of samples were adjusted by quantile normalization and mean centering, and the hierarchical average linkage method clustered genes. Sample clustering used Pearson's correlation coefficient as distances. The enrichment analysis of GO terms used the GOC web tool (see “Relevant Websites section”) with the dataset “PANTHER GO-Slim Biological Process”. Bonferroni correction that multiplies P-value of single Binomial test by the number of independent tests calculated FDR represented by Q-value.

Acknowledgement

The author thanks Prof. Kenta Nakai, the University of Tokyo, for helpful comments and discussions. Computational resources were provided by the supercomputer system SHIROKANE at Human Genome Center, the Institute of Medical Science, the University of Tokyo.

See also: Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Whole Genome Sequencing Analysis

References

- Anders, S., Huber, W., 2010. Differential expression analysis for sequence count data. *Genome Biol.* 11, R106.
- Ay, F., Noble, W.S., 2015. Analysis methods for studying the 3D architecture of the genome. *Genome Biol.* 16, 183.
- Bettegowda, A., Wilkinson, M.F., 2010. Transcription and post-transcriptional regulation of spermatogenesis. *Philos. Trans. R Soc. Lond. B Biol. Sci.* 365, 1637–1651.
- Bohacek, J., Mansuy, I.M., 2015. Molecular insights into transgenerational non-genetic inheritance of acquired behaviours. *Nat. Rev. Genet.* 16, 641–652.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30, 2114–2120.
- Bray, N.L., Pimentel, H., Melsted, P., Pachter, L., 2016. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* 34, 525–527.
- Buenrostro, J.D., Giresi, P.G., Zaba, L.C., Chang, H.Y., Greenleaf, W.J., 2013. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nat. Methods* 10, 1213–1218.
- Bussemaker, H.J., 2006. Modeling gene expression control using Omes Law. *Mol. Syst. Biol.* 2, 2006.0013.
- Chaisson, M.J., Huddleston, J., Dennis, M.Y., et al., 2015. Resolving the complexity of the human genome using single-molecule sequencing. *Nature* 517, 608–611.
- Chaudhary, K., Poirion, O.B., Lu, L., Garmire, L.X., 2018. Deep Learning based multi-omics integration robustly predicts survival in liver cancer. *Clin. Cancer Res.* 24 (6), 1248–1259.
- Cho, H., Davis, J., Li, X., et al., 2014. High-resolution transcriptome analysis with long-read RNA sequencing. *PLOS ONE* 9, e108095.
- Chou, S.J., Wang, C., Sintupisut, N., et al., 2016. Analysis of spatial-temporal gene expression patterns reveals dynamics and regionalization in developing mouse brain. *Sci. Rep.* 6, 19274.
- Cochrane, G., Karsch-Mizrachi, I., Takagi, T., International Nucleotide Sequence Database Collaboration, 2016. The International Nucleotide Sequence Database Collaboration. *Nucleic Acids Res.* 44, D48–D50.
- Conesa, A., Madrigal, P., Tarazona, S., et al., 2016. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17, 13.
- Contreras-Moreira, B., Sebastian, A., 2016. FootprintDB: Analysis of plant cis-regulatory elements, transcription factors, and binding interfaces. *Methods Mol. Biol.* 1482, 259–277.
- Curtis, R.K., Oresic, M., Vidal-Puig, A., 2005. Pathways to the analysis of microarray data. *Trends Biotechnol.* 23, 429–435.
- Dixon, J.R., Jung, I., Selvaraj, S., et al., 2015. Chromatin architecture reorganization during stem cell differentiation. *Nature* 518, 331–336.
- Djebali, S., Davis, C.A., Merkel, A., et al., 2012. Landscape of transcription in human cells. *Nature* 489, 101–108.
- Dobin, A., Davis, C.A., Schlesinger, F., et al., 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29, 15–21.
- Domazet-Lošo, T., Tautz, D., 2010. A phylogenetically based transcriptome age index mirrors ontogenetic divergence patterns. *Nature* 468, 815–818.
- Dong, K., Zhao, H., Tong, T., Wan, X., 2016. NBLDA: Negative binomial linear discriminant analysis for RNA-Seq data. *BMC Bioinform.* 17, 369.
- Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95, 14863–14868.
- Ernst, C., Odom, D.T., Kutter, C., 2017. The emergence of piRNAs against transposon invasion to preserve mammalian genome integrity. *Nat. Commun.* 8, 1411.
- Ewing, B., Green, P., 1998. Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res.* 8, 186–194.
- Finotello, F., Di Camillo, B., 2015. Measuring differential gene expression with RNA-seq: Challenges and strategies for data analysis. *Brief. Funct. Genomics* 14, 130–142.
- Forrest, A.R., Kawaji, H., Rehli, M., et al., 2014. The impact of rare variation on gene expression across tissues. *Nature* 507, 462–470.
- Frye, M., Jaffrey, S.R., Pan, T., Rechavi, G., Suzuki, T., 2016. RNA modifications: What have we learned and where are we headed? *Nat. Rev. Genet.* 17, 365–372.
- Fukuda, N., Touhara, K., 2006. Developmental expression patterns of testicular olfactory receptor genes during mouse spermatogenesis. *Genes Cells* 11, 71–81.
- Gene Ontology Consortium, 2015. Gene Ontology Consortium: Going forward. *Nucleic Acids Res.* 43, D1049–D1056.
- Gomez-Cabrero, D., Abugessaisa, I., Maier, D., et al., 2014. Data integration in the era of omics: Current and future challenges. *BMC Syst. Biol.* 8, 11.
- Grabherr, M.G., Haas, B.J., Yassour, M., et al., 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biotechnol.* 29, 644–652.
- Greenfield, P., Duesing, K., Papanicolaou, A., Bauer, D.C., 2014. Blue: Correcting sequencing errors using consensus and context. *Bioinformatics* 30, 2723–2732.
- GTEX Consortium, 2017. Genetic effects on gene expression across human tissues. *Nature* 550, 204–213.
- Hardcastle, T.J., Kelly, K.A., 2010. baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinform.* 11, 422.
- Hardwick, S.A., Deveson, I.W., Mercer, T.R., 2017. Reference standards for next-generation sequencing. *Nat. Rev. Genet.* 18, 473–484.
- Hashimshony, T., Feder, M., Levin, M., Hall, B.K., Yanai, I., 2015. Spatiotemporal transcriptomics reveals the evolutionary history of the endoderm germ layer. *Nature* 519, 219–222.
- Hasin, Y., Seldin, M., Lusis, A., 2017. Multi-omics approaches to disease. *Genome Biol.* 18, 83.
- Helm, M., Motorin, Y., 2017. Detecting RNA modifications in the epitranscriptome: Predict and validate. *Nat. Rev. Genet.* 18, 275–291.

- Henry, V.J., Bandrowski, A.E., Pepin, A.S., Gonzalez, B.J., Desieux, A., 2014. OMICtools: An informative directory for multi-omic data analysis. Database (Oxford) 2014, bau069.
- Hoheisel, J.D., 2006. Microarray technology: Beyond transcript profiling and genotype analysis. Nat. Rev. Genet. 7, 200–210.
- Ho Sui, S.J., Mortimer, J.R., Arenillas, D.J., *et al.*, 2005. oPOSSUM: Identification of over-represented transcription factor binding sites in co-expressed genes. Nucleic Acids Res. 33, 3154–3164.
- Huang da, W., Sherman, B.T., Lempicki, R.A., 2009. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat. Protoc. 4, 44–57.
- Huang, S., Chaudhary, K., Garmire, L.X., 2017. More is better: Recent progress in multi-omics data integration methods. Front. Genet. 8, 84.
- Hung, J.H., Yang, T.H., Hu, Z., Weng, Z., DeLisi, C., 2012. Gene set enrichment analysis: Performance evaluation and usage guidelines. Brief Bioinform. 13, 281–291.
- Hudson, T.J., Anderson, W., Artez, A., *et al.*, 2010. International Cancer Genome Consortium, International network of cancer genome projects. Nature 464, 993–998.
- Jin, H., Wan, Y.W., Liu, Z., 2017. Comprehensive evaluation of RNA-seq quantification methods for linearity. BMC Bioinform. 18, 117.
- Kadoki, M., Patil, A., Thais, C.C., *et al.*, 2017. Organism-level analysis of vaccination reveals networks of protection across tissues. Cell 171, e21.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K., 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 45, D353–D361.
- Kidder, B.L., Hu, G., Zhao, K., 2011. ChIP-Seq: Technical considerations for obtaining high-quality data. Nat. Immunol. 12, 918–922.
- Kim, D., Pertea, G., Trapnell, C., *et al.*, 2013. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol. 14, R36.
- Kim, K.T., Lee, H.W., Lee, H.O., *et al.*, 2015. Single-cell mRNA sequencing identifies subclonal heterogeneity in anti-cancer drug responses of lung adenocarcinoma cells. Genome Biol. 16, 127.
- Koboldt, D.C., Steinberg, K.M., Larson, D.E., Wilson, R.K., Mardis, E.R., 2013. The next-generation sequencing revolution and its impact on genomics. Cell 155, 27–38.
- Kulakovskiy, I.V., Medvedeva, Y.A., Schaefer, U., *et al.*, 2013. HOCOMOCO: A comprehensive collection of human transcription factor binding sites models. Nucleic Acids Res. 41, D195–D202.
- Kulkarni, P., Frommolt, P., 2017. Challenges in the setup of large-scale next-generation sequencing analysis workflows. Comput. Struct. Biotechnol. J. 15, 471–477.
- Laird, P.W., 2010. Principles and challenges of genomewide DNA methylation analysis. Nat. Rev. Genet. 11, 191–203.
- Lam, H.Y., Clark, M.J., Chen, R., *et al.*, 2012. Performance comparison of whole-genome sequencing platforms. Nat. Biotechnol. 30, 78–82.
- Li, H., Wang, X., Rukina, D., *et al.*, 2018. An integrated systems genetics and omics toolkit to probe gene function. Cell Syst. 6 (1), 90–102.
- Li, R., Hsieh, C.L., Young, A., *et al.*, 2015. Illumina synthetic long read sequencing allows recovery of missing sequences even in the “Finished” *C. elegans* genome. Sci. Rep. 5, 10814.
- Li, S., Labaj, P.P., Zumbo, P., *et al.*, 2014. Detecting and correcting systematic variation in large-scale RNA sequencing data. Nat. Biotechnol. 32, 888–895.
- Li, X., Kim, Y., Tsang, E.K., *et al.*, 2017. The impact of rare variation on gene expression across tissues. Nature 550, 239–243.
- Li, Y., Wu, F.X., Ngom, A., 2018. A review on machine learning principles for multi-view biological data integration. Brief. Bioinform. 1 (19), 325–340.
- Loman, N.J., Misra, R.V., Dallman, T.J., *et al.*, 2012. Performance comparison of benchtop high-throughput sequencing platforms. Nat. Biotechnol. 30, 434–439.
- Loman, N.J., Watson, M., 2015. Successful test launch for nanopore sequencing. Nat. Methods 12, 303–304.
- Lowe, E.K., Cuomo, C., Arnone, M.I., 2017. Omics approaches to study gene regulatory networks for development in echinoderms. Brief. Funct. Genom. 16, 299–308.
- Mamanova, L., Andrews, R.M., James, K.D., *et al.*, 2010. FRT-seq: Amplification-free, strand-specific transcriptome sequencing. Nat. Methods 7, 130–132.
- Mardis, E.R., 2013. Next-generation sequencing platforms. Annu. Rev. Anal. Chem. (Palo Alto Calif) 6, 287–303.
- Margolin, G., Khil, P.P., Kim, J., Bellani, M.A., Camerini-Otero, R.D., 2014. Integrated transcriptome analysis of mouse spermatogenesis. BMC Genom. 15, 39.
- Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., Gilad, Y., 2008. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. Genome Res. 18, 1509–1517.
- McCoy, R.C., Taylor, R.W., Blauwkamp, T.A., *et al.*, 2014. Illumina TruSeq synthetic long-reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. PLOS ONE 9, e106689.
- Merino, G.A., Conesa, A., Fernandez, E.A., 2017. A benchmarking of workflows for detecting differential splicing and differential expression at isoform level in human RNA-seq studies. Brief. Bioinform.
- Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B., 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat. Methods 5, 621–628.
- Nakano, K., Shiroma, A., Shimoji, M., *et al.*, 2017. Advantages of genome sequencing by long-read sequencer using SMRT technology in medical area. Hum. Cell 30, 149–161.
- Nazarov, P.V., Muller, A., Kaoma, T., *et al.*, 2017. RNA sequencing and transcriptome arrays analyses show opposing results for alternative splicing in patient derived samples. BMC Genom. 18, 443.
- Okumura, T., Makiguchi, H., Makita, Y., Yamashita, R., Nakai, K., 2007. Melina II: A web tool for comparisons among several predictive algorithms to find potential motifs from promoter regions. Nucleic Acids Res. 35, W227–W231.
- Olarerin-George, A.O., Hogenesch, J.B., 2015. Assessing the prevalence of mycoplasma contamination in cell culture via a survey of NCBI’s RNA-seq archive. Nucleic Acids Res. 43, 2535–2542.
- Ouyang, Z., Zhou, Q., Wong, W.H., 2009. ChIP-Seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. Proc. Natl. Acad. Sci. USA 106, 21521–21526.
- Oyelade, J., Isewon, I., Oladipupo, F., *et al.*, 2016. Clustering algorithms: Their application to gene expression data. Bioinform. Biol. Insights 10, 237–253.
- Ozsolak, F., Milos, P.M., 2011a. RNA sequencing: Advances, challenges and opportunities. Nat. Rev. Genet. 12, 87–98.
- Ozsolak, F., Milos, P.M., 2011b. Single-molecule direct RNA sequencing without cDNA synthesis. Wiley Interdiscip. Rev. RNA 2, 565–570.
- Papalexi, E., Satija, R., 2018. Single-cell RNA sequencing to explore immune cell heterogeneity. Nat. Rev. Immunol. 18 (1), 35–45.
- Parekh, S., Ziegenhain, C., Vieth, B., Enard, W., Hellmann, I., 2016. The impact of amplification on differential expression analyses by RNA-seq. Sci. Rep. 6, 25533.
- Park, P.J., 2009. ChIP-seq: Advantages and challenges of a maturing technology. Nat. Rev. Genet. 10, 669–680.
- Park, S.J., Komata, M., Inoue, F., *et al.*, 2013. Inferring the choreography of parental genomes during fertilization from ultralarge-scale whole-transcriptome analysis. Genes Dev. 27, 2736–2748.
- Park, S.J., Saito-Adachi, M., Komiyama, Y., Nakai, K., 2016. Advances, practice, and clinical perspectives in high-throughput sequencing. Oral Dis. 22, 353–364.
- Park, S.J., Umemoto, T., Saito-Adachi, M., *et al.*, 2014. Computational promoter modeling identifies the modes of transcriptional regulation in hematopoietic stem cells. PLOS ONE 9, e93853.
- Paulsen, J., Rodland, E.A., Holden, L., Holden, M., Hovig, E., 2014. A statistical model of ChIA-PET data for accurate detection of chromatin 3D interactions. Nucleic Acids Res. 42, e143.
- Perneger, T.V., 1998. What’s wrong with Bonferroni adjustments. BMJ 316, 1236–1238.
- Pertea, M., Kim, D., Pertea, G.M., Leek, J.T., Salzberg, S.L., 2016. Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown. Nat. Protoc. 11, 1650–1667.
- Pimentel, H., Bray, N.L., Puente, S., Melsted, P., Pachter, L., 2017. Differential analysis of RNA-seq incorporating quantification uncertainty. Nat. Methods 14, 687–690.
- Reuter, J.A., Spacek, D.V., Snyder, M.P., 2015. High-throughput sequencing technologies. Mol. Cell 58, 586–597.
- Risso, D., Ngai, J., Speed, T.P., Dudoit, S., 2014. Normalization of RNA-seq data using factor analysis of control genes or samples. Nat. Biotechnol. 32, 896–902.
- Robinson, M.D., McCarthy, D.J., Smyth, G.K., 2010. edgeR: A bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26, 139–140.
- Ross, M.G., Russ, C., Costello, M., *et al.*, 2013. Characterizing and measuring bias in sequence data. Genome Biol. 14, R51.
- Sahraeian, S.M.E., Mohiyuddin, M., Sebra, R., *et al.*, 2017. Gaining comprehensive biological insight into the transcriptome by performing a broad-spectrum RNA-seq analysis. Nat. Commun. 8, 59.
- Saitou, M., Kagiwada, S., Kurimoto, K., 2012. Epigenetic reprogramming in mouse pre-implantation development and primordial germ cells. Development 139, 15–31.
- Sanger, F., Air, G.M., Barrell, B.G., *et al.*, 1977. Nucleotide sequence of bacteriophage phi X174 DNA. Nature 265, 687–695.

- Sasamoto, Y., Hayashi, R., Park, S.J., *et al.*, 2016. PAX6 Isoforms, along with reprogramming factors, differentially regulate the induction of cornea-specific genes. *Sci. Rep.* 6, 20807.
- Schmieder, R., Edwards, R., 2011. Quality control and preprocessing of metagenomic datasets. *Bioinformatics* 27, 863–864.
- Schurch, N.J., Schofield, P., Gierlinski, M., *et al.*, 2016. How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA* 22, 839–851.
- Schwartzman, A., Lin, X., 2011. The effect of correlation in false discovery rate estimation. *Biometrika* 98, 199–214.
- Schwarzer, W., Abdennur, N., Goloborodko, A., *et al.*, 2017. Two independent modes of chromatin organization revealed by cohesin removal. *Nature* 551, 51–56.
- Seqc_Maqc-iii_Consortium, 2014. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nat. Biotechnol.* 32, 903–914.
- Shapiro, E., Biezuner, T., Linnarsson, S., 2013. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat. Rev. Genet.* 14, 618–630.
- Shendure, J., Balasubramanian, S., Church, G.M., *et al.*, 2017. DNA sequencing at 40: Past, present and future. *Nature* 550, 345–353.
- Shi, L., Reid, L.H., Jones, W.D., *et al.*, 2006. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat. Biotechnol.* 24, 1151–1161.
- Sloan, C.A., Chan, E.T., Davidson, J.M., *et al.*, 2016. ENCODE data at the ENCODE portal. *Nucleic Acids Res.* 44, D726–D732.
- Sterne, J.A., Davey Smith, G., 2001. Sifting the evidence-what's wrong with significance tests? *BMJ* 322, 226–231.
- Storey, J.D., Tibshirani, R., 2003. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA* 100, 9440–9445.
- Strong, M.J., Xu, G., Morici, L., *et al.*, 2014. Microbial contamination in next generation sequencing: Implications for sequence-based analysis of clinical samples. *PLOS Pathog.* 10, e1004437.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* 102, 15545–15550.
- Sudmant, P.H., Rausch, T., Gardner, E.J., *et al.*, 2015. An integrated map of structural variation in 2,504 human genome. *Nature* 526, 75–81.
- Tang, F., Barbacioru, C., Wang, Y., *et al.*, 2009. mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* 6, 377–382.
- Teng, M., Love, M.I., Davis, C.A., *et al.*, 2016. A benchmark for RNA-seq quantification pipelines. *Genome Biol.* 17, 74.
- Trapnell, C., Roberts, A., Goff, L., *et al.*, 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat. Protoc.* 7, 562–578.
- Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28, 511–515.
- van Dam, S., Vosa, U., van der Graaf, A., Franke, L., de Magalhaes, J.P., 2017. Gene co-expression analysis for functional classification and gene-disease predictions. *Brief. Bioinform.* 139.
- Vlieghe, D., Sandelin, A., De Bleser, P.J., *et al.*, 2006. A new generation of JASPAR, the open-access repository for transcription factor binding site profiles. *Nucleic Acids Res.* 34, D95–D97.
- Wang, C., Gong, B., Bushel, P.R., *et al.*, 2014. The concordance between RNA-seq and microarray data depends on chemical treatment and transcript abundance. *Nat. Biotechnol.* 32, 926–932.
- Wang, L., Feng, Z., Wang, X., Wang, X., Zhang, X., 2010. DEGseq: An R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics* 26, 136–138.
- Wang, Y., Navin, N.E., 2015. Advances and applications of single-cell sequencing technologies. *Mol. Cell* 58, 598–609.
- Wang, Z., Gerstein, M., Snyder, M., 2009. RNA-Seq: A revolutionary tool for transcriptomics. *Nat. Rev. Genet.* 10, 57–63.
- Weinstein, J.N., Collisson, E.A., Mills, G.B., *et al.*, 2013. The cancer genome Atlas pan-cancer analysis project. *Nat. Genet.* 45, 1113–1120.
- Weirather, J.L., de Cesare, M., Wang, Y., *et al.*, 2017. Comprehensive comparison of Pacific Biosciences and Oxford Nanopore Technologies and their applications to transcriptome analysis. *F1000Research* 6, 100.
- Wen, L., Tang, F., 2016. Single-cell sequencing in stem cell biology. *Genome Biol.* 17, 71.
- Williams, C.R., Baccarella, A., Parrish, J.Z., Kim, C.C., 2016. Trimming of sequence reads alters RNA-Seq gene expression estimates. *BMC Bioinform.* 17, 103.
- Wingender, E., Chen, X., Hehl, R., *et al.*, 2000. TRANSFAC: An integrated system for gene expression regulation. *Nucleic Acids Res.* 28, 316–319.
- Xie, Y., Wu, G., Tang, J., *et al.*, 2014. SOAPdenovo-Trans: De novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics* 30, 1660–1666.
- Xu, Z., Peters, R.J., Weirather, J., *et al.*, 2015. Full-length transcriptome sequences and splice variants obtained by a combination of sequencing platforms applied to different root tissues of *Salvia miltiorrhiza* and tanshinone biosynthesis. *Plant J.* 82, 951–961.
- Yamashita, R., Sathira, N.P., Kanai, A., *et al.*, 2011. Genome-wide characterization of transcriptional start sites in humans by integrative transcriptome analysis. *Genome Res.* 21, 775–789.
- Yan, K.K., Zhao, H., Pang, H., 2017. A comparison of graph- and kernel-based -omics data integration algorithms for classifying complex traits. *BMC Bioinform.* 18, 539.
- Yu, J., Gu, X., Yi, S., 2016. Ingenuity pathway analysis of gene expression profiles in distal nerve stump following nerve injury: Insights into Wallerian degeneration. *Front. Cell Neurosci.* 10, 274.
- Zhang, S., Liu, C.C., Li, W., *et al.*, 2012. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Res.* 40, 9379–9391.

Relevant Websites

<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
 Babraham Bioinformatics.
<http://chip-atlas.org/>
 ChIP-Atlas.
<http://dc2.cistrome.org/>
 Cistrome DB.
http://hannonlab.cshl.edu/fastx_toolkit/
 FASTX-Toolkit.
<http://www.geneontology.org/>
 Gene Ontology Consortium.

<https://github.com/marcelm/cutadapt>
GitHub - marcelm/cutadapt.
<http://gtrd.biouml.org/>
GTRD.

Biographical Sketch



Sung-Joon Park received the PhD degree in Engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2005. He has worked as a researcher in Kobe University and Kyoto University. He is currently a senior assistant professor (project) in Human Genome Center, the Institute of Medical Science, the University of Tokyo. He is working on the field of genome information science, and has explored genome features and developed several databases devoted to embryogenesis and stem cell biology. His main research interest is to develop computational ways for interpreting biological information, especially for the functional analysis of gene expression and regulation with high-throughput sequencing data. He is a member of Information Processing Society of Japan (IPSI), The Japan Society for Artificial Intelligence (JSAI), and Japanese Society for Bioinformatics (JSBi).

Metabolome Analysis

Camila Pereira Braga and Jiri Adamec, University of Nebraska-Lincoln, Lincoln, NE, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Omics strategies have been used in different biological systems to generate knowledge about new biomarkers associated with diagnosis, disease progression, response to treatment (Reichel, 2011), and generally to understand biological processes and their regulation. In the OMICS arena, the proteome represents all proteins expressed in a cell, tissue, or organism, while the genome is associated with the genetic information (Wilkins *et al.*, 1996), the transcriptome with the RNA transcripts, and the metabolome with the metabolites. In this context, the Omics areas are referred to as proteomics, genomics, transcriptomics, and metabolomics. Recently, new terms have been introduced referring either to new areas (interactomics and topomics) or targeting specific class of the molecules (lipidomics and glycomics as a part of metabolomics or degradomics as a part of proteomics) (Nicholson *et al.*, 2007).

The primary focus of this article is on metabolomics – metabolome analysis, the most recently introduced strategy among the Omics. In 1998, Oliver *et al.* (1998) introduced the term metabolome analysis for functional genomics using a yeast model, associating the results of gene deletion or overexpression with the change in the relative concentration of metabolites. That same year, Tweeddale *et al.* (1998) used the term introduced by Oliver *et al.* (1998) (metabolome analysis) in analyzing *Escherichia coli* cell metabolites for phenotypic profiling. One year later, Nicholson *et al.* (1999) defined the term metabonomics as the “quantitative measurement of the dynamic multiparametric metabolic response of living systems to pathophysiological stimuli or genetic modification”.

In 2001, such terms were clearly defined by Fiehn: metabolomics – the “metabolome of the biological system under study,” metabolic fingerprinting – “a rapid classification of samples according to their origin or their biological relevance,” metabolite profiling – the investigation of pre-defined metabolites (based in their association with a specific pathway or class of compounds), and metabolite target analysis – which aims “to study the primary effect of any alteration (e.g., a genetic mutation) directly” (Fiehn, 2001). Other terms have also appeared throughout the years, such as metabolic footprinting, which refers to exo metabolomes and secretomes (metabolites in extracellular fluids) (Kell *et al.*, 2005).

Metabolomics analysis can be categorized into two approaches, targeted and untargeted metabolomics. In targeted metabolomics, the metabolites selected for quantification are known and represent specific pathway(s) or class(es) of molecules. Untargeted metabolomics, on the other hand, is used to determine as many metabolites as possible and involves both metabolites quantification and their identification. (Cambiaghi *et al.*, 2017). Targeted metabolomics refers to absolute quantification (nM or mg mL⁻¹), specifically by using an internal standard and semiquantitative or quantitative analysis to detect known compounds related to specific pathways defined *a priori*, based on the hypotheses of the study (Roberts *et al.*, 2012; Cambiaghi *et al.*, 2017). Untargeted metabolomics refers to the measure of all possible metabolites in a sample using relative quantification (fold change), and comparison between samples (Roberts *et al.*, 2012; Cambiaghi *et al.*, 2017). The challenges of untargeted metabolomics are metabolite identification, characterization of unknown small molecules, and the time needed to process the generated data (Roberts *et al.*, 2012).

Metabolome Analysis

Metabolome analysis involves several steps, as indicated in Fig. 1. The study starts with an experimental design that consists of two main steps (Goodacre *et al.*, 2007). The first step is to define the biological problem, generating the hypotheses of the study. Based on that, the second step involves various metabolomics approaches used in the analysis. These steps typically include sample preparation, sample analysis and data acquisition, data preprocessing, and data statistical analysis and interpretation.

Sample Preparation

Sample preparation is an important step in metabolome analysis, consisting of sample collection and metabolite extraction. Metabolite extraction depends on the sample type (cell, biological fluid, or tissues), the metabolomics approach defined in the experimental design, and can involve steps such as quenching (removal of proteins and their enzymatic activities by denaturation and precipitation), use of internal standards, and optional derivatization of specific classes of metabolites (Klassen *et al.*, 2017).

In the quenching step, the inactivation of the cellular metabolic and enzymatic activities is achieved by sample exposure to an acidic or alkaline solution, or/and cold (< –40°C) or hot solutions (Villas-Bôas *et al.*, 2006). Some examples are the use of cold methanol, liquid nitrogen, and perchloric acid. The choice of the method(s) usually depends on the stability of the metabolites targeted for analysis. If, for example, metabolites of interests are not stable at higher temperature or low pH, hot solution or acids are avoided in quenching steps. Liquid nitrogen and cold methanol are usually used for untargeted metabolomics as they represent relatively mild conditions and have a minimal impact on the stability of samples. The quenching step is often combined with extraction. For example, performing an extraction of both biomass and media will allow a simultaneous intracellular and

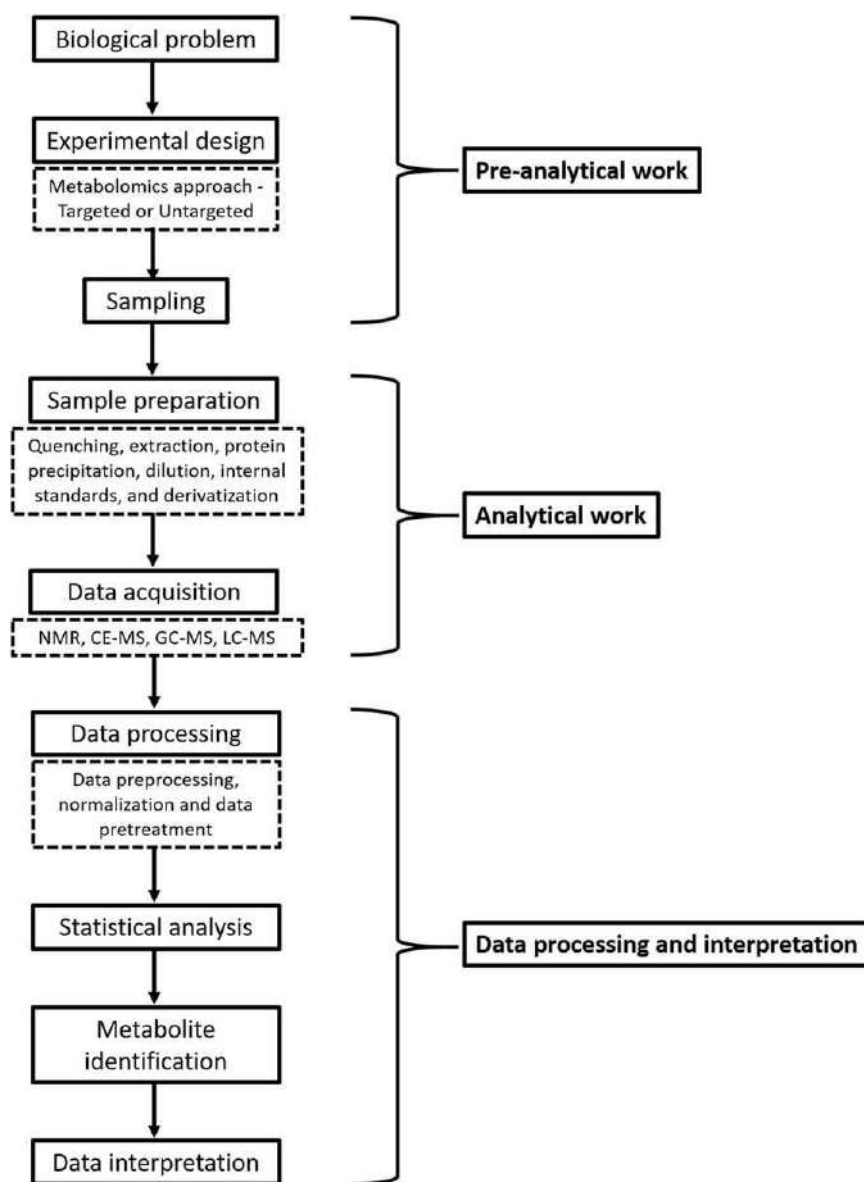


Fig. 1 Metabolome analysis workflow.

extracellular analysis. On the other hand, prior separation of the cellular components by, e.g., centrifugation will permit an intracellular analysis without the interference of extracellular compounds (Villas-Bôas *et al.*, 2005).

The most common solvents used to extract metabolites are organic solvents: polar solvents (methanol, ethanol, and mixtures of methanol-water) are used to extract polar metabolites, while non-polar solvents (chloroform, hexanes, and ethyl acetate) are employed to extract lipids (Villas-Bôas *et al.*, 2005, 2006). The selected extraction method should obtain the maximum quantity of intracellular metabolites to avoid loss of metabolites, and to be compatible with the analytical tools used for their quantification and identification (Villas-Bôas *et al.*, 2005, 2006). The concentration step (freeze-drying and reducing volume under vacuum using e.g., roto-vac) is recommended when an extraction method is used, as sample dilution can decrease metabolite concentrations under the detection limits, and affect the sensitivity and overall results. Furthermore, it is possible to analyze the metabolites of interest by using solid-phase extraction (SPE) and solid-phase micro-extraction (SPME) to concentrate the intracellular metabolites from a diluted sample.

Protein precipitation is an important step that removes proteins present in the sample, extracting the metabolites that can non-covalently bind to proteins by , and preserving the column when using liquid chromatography (LC) as an analytical method (Bruce *et al.*, 2008).

The derivatization step is particularly important in Gas Chromatography – Mass Spectrometry (GC–MS) based analytical techniques as the GC can separate only volatile compounds (Garcia and Barbas, 2011). In this case, silylation, alkylation, or other derivatization reactions can convert non-volatile compounds to volatile ones.

Microbial sample preparation

Winder *et al.* (2008) studied different methods of sample quenching by evaluating 60% aqueous methanol (-48°C), tricine-buffered (0.5 mM, pH 7.4) aqueous 60% methanol solution (-48°C), and boiling with ethanol (90°C); in combination with various extraction steps including 100% methanol (-48°C), methanol/chloroform (2:1), perchloric acid (0.25 M), and boiling with ethanol (90°C) and potassium hydroxide (0.25 M, 80°C). The study concluded that the procedure utilizing 60% aqueous methanol (-48°C) to quench the metabolism, and extraction with 100% methanol (-48°C) were the most appropriate methods to extract intracellular metabolites in *Escherichia coli* cultures (Winder *et al.*, 2008). Li *et al.* (2013) found that the most appropriate quenching method for *Bacillus licheniformis* consisted of perchloric acid rather than hydrochloric and phosphoric acids. Another study suggested a cold glycerol-saline solution as a quenching method for analyzing intracellular metabolites in microbial cell cultures (e.g., *Pseudomonas fluorescens* – a gram-negative bacterium, *Streptomyces coelicolor* – a gram-positive bacterium, and *Saccharomyces cerevisiae* – baker's yeast) (Villas-Bôas and Bruheim, 2007).

Plants tissue sample preparation

The most common method used as for quenching to extract metabolites in plants after harvesting is snap-freezing the plant tissue with liquid nitrogen, and subsequent storage at -80°C (Jorge *et al.*, 2016). In the extraction step, the most common solvents are chloroform, methanol, and water (2:1:1, v/v) to separate lipophilic (non-polar) and hydrophilic (polar) metabolites (Verpoorte *et al.*, 2008).

Animal tissue sample preparation

For animal and human tissues, the primary methods used to quench metabolism are snap-freezing in liquid nitrogen and cold methanol (Stentiford *et al.*, 2005; Wang *et al.*, 2011). For the extraction of polar and non-polar metabolites, chloroform and methanol (2:1, v/v) has been demonstrated to be the most efficient solvent (van Ginneken *et al.*, 2007; Atherton *et al.* 2008).

Sample Analysis and Data Acquisition

Common analytical platforms employed for data acquisition in metabolome analysis are nuclear magnetic resonance (NMR) and mass spectrometry (MS) analysis, typically coupled with separation techniques such as gas chromatography (GC), high or ultra-performance liquid chromatography (HPLC or UPLC), or capillary electrophoresis (CE).

NMR based metabolomics

NMR is a relatively robust technique which is typically applied to studies such as metabolite fingerprinting, profiling, and metabolic flux analysis, involving different samples such as cell extracts, cell cultures, tissues, and biofluids. NMR is based on spin behavior of atomic nuclei in magnetic field which is represented by the resonance frequency. NMR is nondestructive, noninvasive and nonequilibrium method requiring minimal or no sample preparation to analyze various metabolites in the micromolar range (Reo, 2002; Beckonert *et al.*, 2007). As NMR is a noninvasive method, it has been also used extensively in metabolomics analysis to diagnose diseases (Brindle *et al.*, 2002; Gowda *et al.* 2008; Capati *et al.*, 2017).

NMR in metabolomics analysis exhibits highly reproducible and quantitative characteristics; molecular properties such as hydrophobicity and acid dissociation constant (pKa) do not influence NMR sensitivity (Pan and Raftery, 2007). The limitations of NMR in metabolomics analysis are due to poor sensitivity and spectral resolution (minimum spectral distance between two distinguishable peaks) that can compromise metabolite identification in complex samples. The sample preparation is the same for the most types of samples. To generate good NMR data, it is necessary to add an internal chemical shift standard, a phosphate buffer solution, and deuterated water (D_2O , the latter of which is used as a lock frequency for long-term magnetic field drifts (Heude *et al.*, 2017).

The most common NMR spectroscopic methods used to quantify metabolites are one-dimensional proton NMR spectroscopy (1D NMR) and two-dimensional proton J-resolved NMR spectroscopy (2D Jres NMR). The advantages of 1D NMR are to provide a relatively good quantitative profile of the metabolites and rapid spectral acquisition. In addition, a single standard can be used for absolute quantification of the measured metabolites (Huang *et al.*, 2015). On the other hand, the only more abundant metabolites can be detected (concentrations $> 1\text{--}10\ \mu\text{M}$), and peak overlap hinders metabolite identification (Gebregiworgis and Powers, 2012). 2D-1H Jres can partially resolve the peak overlaps observed in 1D-1H NMR by determining the peak intensity and properties, using the separation of chemical shifts and *J* coupling in two spectral dimensions to improve metabolite identification (Huang *et al.*, 2014, 2015).

MS based metabolomics

In comparison to NMR, MS is more sensitive and specific. Furthermore, the increase in coverage of the metabolite analysis can potentially lead to the detection of thousands of metabolites. Every mass spectrometer consists of three integrated parts including 1/a ionization source that adds charge to molecules to be analyzed, 2/a mass analyzer that separates molecules based on their mass/charge ratio (*m/z*), and 3/a detector. Although direct injection is a rapid technique, often used in metabolomics analysis (in both targeted analysis and metabolite profiling – to quantify metabolites and elucidate metabolite structure), the major limitation is associated with low and inconsistent ionization efficiencies in complex biological samples (Zhang *et al.*, 2012).

Therefore, the best results are achieved by using separation techniques such as GC, HPLC/UPLC, and CE (Zhang *et al.*, 2012) prior to MS analysis. Although this can be done using off-line separation followed, by MS analysis of individually collected fractions, an in-line setup that couples the separation and MS steps into one system is preferred, and leads to better results.

GC-MS metabolomics

GC-MS in metabolomics analysis can separate and detect volatile metabolites that occur naturally (alcohols, aldehydes, esters, etc.), or metabolites that become volatile after derivatization (semi-volatile metabolites – amides, amines, amino acids, sugars, organic acids, peptides, and lipids; and nonvolatile metabolites that are chemically modified to increase their stability in the high temperatures used in GC-MS analysis) (Villas-Bôas *et al.*, 2005; Zhang *et al.*, 2012). Over the years, the GC-MS approach has demonstrated robustness, high reproducibility, and it is relatively inexpensive (Villas-Bôas *et al.*, 2005; Zhang *et al.*, 2012). The basic principle of GC separation is based on the partition of specific molecules between gas and liquid phases at a given temperature, and the effect of the portioning on their movement through the GC column. Molecules in the gas phase move through the GC capillary column and are subsequently detected by MS. Because the molecules have different volatility, and partition differently between the gas and liquid phases depending on the temperature, applying an increasing temperature gradient on the GC column allows for the separation of molecules in time. GC capillary columns are selected based on the metabolite characteristics, such as polarity or volatility (Villas-Bôas *et al.*, 2005). For semi-volatile and non-volatile metabolites, the derivatization step needs to be included in the sample preparation. The methods used for derivatization include usually silylation (especially used with sugars – amino sugars, sugar alcohols, and phosphorylated sugars), and alkylation or esterification (used with polyfunctional amines and organic acids) (Villas-Bôas *et al.*, 2005). For untargeted metabolite profiling, the GC-MS approach involves sample extraction, sample derivatization and separation by GC, and ionization of the separated molecules, followed by detection and data evaluation (Kopka *et al.*, 2004). For targeted analyses, additional steps such as extraction with an organic solvent to enrich specific classes of metabolites, or adjustment of separation conditions, such as the use of long capillary columns (30–60 m), or/and sample injection at various, sample specific, temperatures may be necessary (Villas-Bôas *et al.*, 2005).

Using GC-MS techniques, Lv and Yang (2012) found potential biomarkers of breast cancer in free fatty acids (palmitic acid, stearic acid, and linoleic acid) using serum samples analyzed by GC-MS. Changes in sn-glycerol-3-phosphate in lipid metabolism were found in metabolomics analysis by GC-MS when comparing breast cancer and normal tissues (Brockmöller *et al.*, 2012). In addition, Dória *et al.* (2012) found lysophospholipids and saturated fatty acids in phosphatidylinositols using an Electrospray Ionization Mass Spectrometry (ESI-MS) and cell culture model. In 2012, Nishiumi *et al.* (2012) used GC-MS to analyze a serum-targeted metabolome in Colorectal Cancer (CRC) patients, and found 2-hydroxybutyrate, aspartic acid, kynurenine, and cystamine as potential biomarkers for early detection. Years later, the use of NMR-based fecal metabolomic fingerprinting in CRC patients led to the observation of reduced levels of acetate, butyrate, propionate, glucose, glutamine, and elevated quantities of succinate, proline, alanine, dimethylglycine, valine, glutamate, leucine, isoleucine and lactate. The changes in metabolite levels between samples from CRC patients and the control group can be associated with malabsorption of nutrients, disturbance in the bacterial ecology, and increased glycolysis and glutaminolysis (Lin *et al.*, 2016).

LC-MS metabolomics

In LC-MS, liquid samples are directly injected into the LC column and metabolites are separated based on their interaction with a stationary phase prior MS detection. The stationary phase is usually represented by spherical beads covered on the surface with specific functional groups that define the nature of the separation. Whilst polar metabolites are typically separated by ion exchange or Hydrophilic Interaction Chromatography (HILIC) columns, non-polar metabolites are separated by their hydrophobicity on Reverse Phase (RPLC) columns (C4, C8, or C18). Non-hydrophobic molecules without charge can be separated by Normal Phase Liquid Chromatography (NPLC). In all cases, the metabolites retained on the column are eluted by a solvent that disrupts metabolite-stationary phase interactions, and leads to metabolite release. NPLC uses a solvent with a higher polarity than that of the stationary phase (nonpolar are eluted first), whereas RPLC uses a solvent with a lower polarity than the stationary phase (Lopes *et al.*, 2017). The ionization techniques applied in LC-MS are electrospray ionization (ESI), atmospheric pressure chemical ionization (APCI), and atmospheric pressure photoionization (APPI). These ionization processes do not modify or fragment biomolecules and therefore are considered to be soft ionization techniques. Although the most common technique is ESI (Villas-Bôas *et al.*, 2005; Lopes *et al.*, 2017), APPI and APCI are primary techniques that are used to analyze metabolites with limited ESI efficiency, such as Vitamin D and its metabolites (Adamec *et al.*, 2011).

The major difference between untargeted and targeted LC-MS approaches is the MS parameter setup. Untargeted metabolomics is designed to detect as many metabolites as possible, therefore the instrument is optimized for a broad range of molecular masses (m/z), typically set to 50–2200 m/z . Targeted metabolomics, on the other hand, is optimized for maximum sensitivity at the specific m/z of metabolite(s) of interest. Currently, two types of MS instruments, QQQ and qTRAP, are almost exclusively used for targeted analysis using techniques known as Selected Reaction Monitoring (SRM) and Multiple Reaction Monitoring (MRM). In these techniques, data is acquired from one or more specific product ions generated by fragmentation of selected precursor ion(s) (metabolite(s) of interest), with known m/z , using a multi-stage mass spectrometer (QQQ or qTRAP) (Hoffmann, 1996). In first stage, only ions with defined m/z are allowed to pass through to the second stage, in which the precursor ion is fragmented. In the third stage, only the intensities of fragments (product ions) corresponding to the original precursor ion are monitored and used for quantification.

LC-MS techniques are widely used in many biological studies. For example, using an untargeted metabolomic analysis of serum samples with 300 Type 2 Diabetes (T2D) patients and 300 healthy controls by UPLC-MS, the authors found lipids, hexose sugars, and

purine nucleotides as biomarkers of T2D (Drogan *et al.*, 2015). Metabolomics studies conducted in Alzheimer's patients plasma samples identified desmosterol as a potential biomarker (Sato *et al.*, 2012). Although, LC-MS is very powerful tool, to increase a number of identified metabolites, the approach is often combined with GC-MS. Using both, LC-MS and GC-MS, lysophosphocholine, tryptophan, phytosphingosine, dihydrosphingosine, hexadecosphinganine, arachidonic acid, *N,N*-dimethylglycine, thymine, glutamine, glutamic acid, and cytidine were also identified as potential biomarkers of Alzheimer disease (Li *et al.*, 2010; Wang *et al.*, 2014).

Capillary electrophoresis (CE)-MS metabolomics

CE-MS is a separation technique that uses capillary electrophoresis to achieve high efficiency and selectivity, peak capacity (by resolving more peaks), fast analysis, and small sample volume, to analyze polar and ionic compounds without a derivatization step (Ramautar *et al.*, 2009). This technique can be used to analyze targeted and untargeted metabolites, such as amino acids, carbohydrates, vitamins, thiols, peptides, and nucleotides (Villas-Bôas *et al.*, 2005).

In CE, capillary zone electrophoresis (CZE), in which separation is based in differences in electrophoretic mobilities (neutral compounds are not separated), is the most used mode. Other modes include micellar electrokinetic chromatography (MEKC, separation of neutral compounds is possible), capillary isoelectric focusing (CIEF, where a pH gradient is used to separate amphiprotics), capillary isotachopheresis (CITP, for separation of small molecules and ions), capillary gel electrophoresis (CGE; large molecules and polymers can be analyzed using gels), and affinity capillary electrophoresis (ACE, which is based on biospecific interactions) (Rodrigues *et al.*, 2017).

Data Processing and Analysis

A critical component of any metabolomics platform is the link between the generation of data and realization of the value it contains. This link is filled with various bioinformatics and statistical tools specifically designed for metabolomics data mining. The individual steps involved in this process are typically data preprocessing, normalization, data pretreatment, and statistical analysis with interpretation and visualization (Figs. 2 and 3) (Karaman, 2017).

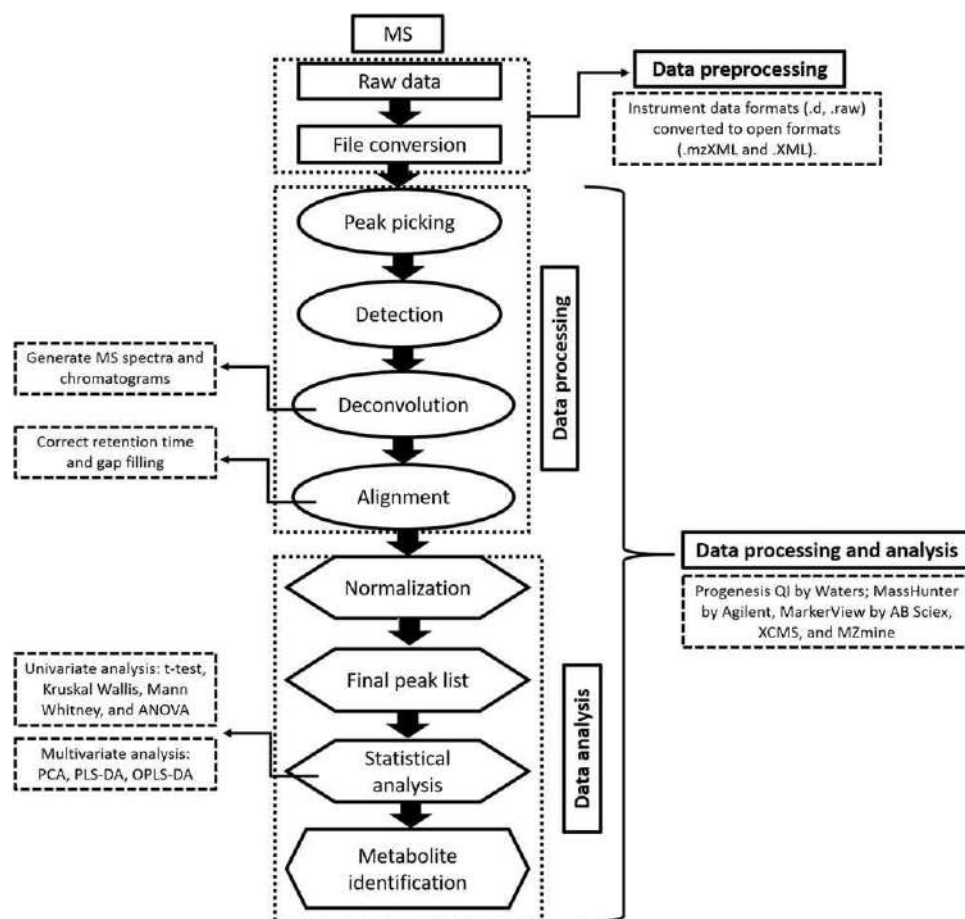


Fig. 2 MS data processing workflow.

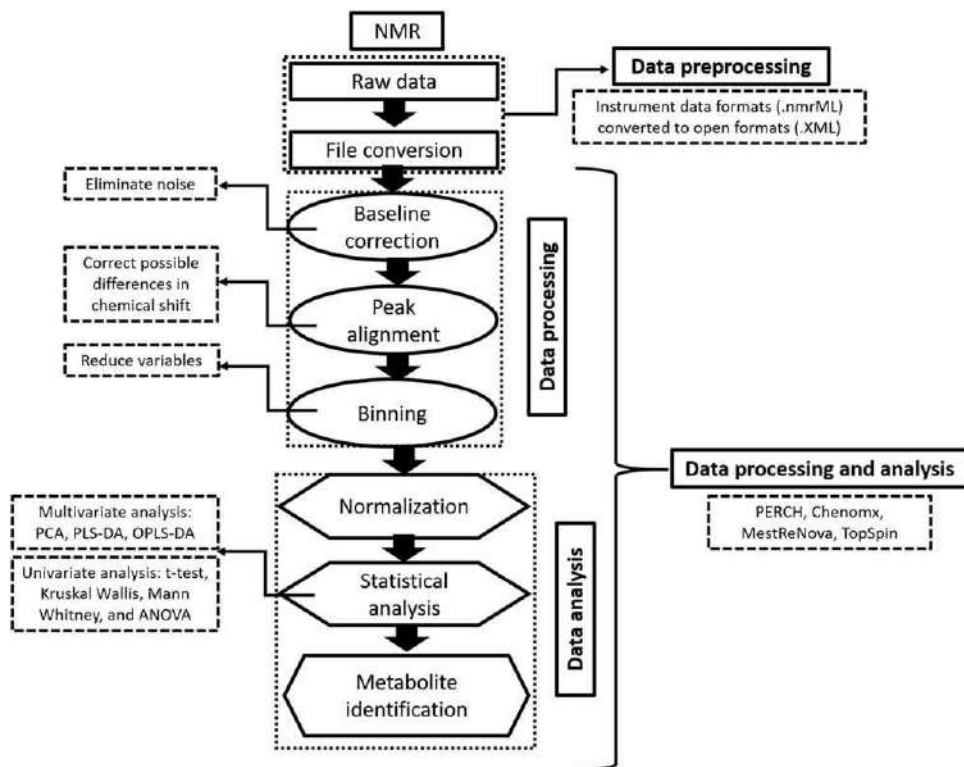


Fig. 3 NMR data processing workflow.

MS data preprocessing

Data generated by MS are collected in vendor proprietary data formats, and to be processed by vendor independent software packages, they need to be converted into standardized file formats such as mzData, mzXML or mzML (Karaman, 2017). Following the conversion, data is processed by various data preprocessing software packages (Table 1) to generate a single sample file in which characteristics such as m/z, retention time, and ion intensity can be accessed for each ion (Katajamaa and Orešič, 2007). The process consists of several steps including (but not limited to) generation of a peak list, optional peak list filtration, and removal of isotopes (as each compound (metabolite) is represented by several isotopic peaks). For processing of LC- and GC-MS data, chromatograms must be generated for individual peaks and results deconvoluted (separating signals corresponding to real metabolites from instrument interferences). These steps are used to generate extracted MS spectra and chromatograms that can be employed for metabolite identification, and to calculate relative concentrations, respectively (Katajamaa and Orešič, 2007). In addition, the generation of chromatograms and spectrum deconvolution help to differentiate signals that originate from the actual analytes, or from contaminants or instrumental noise, and reduce data dimensionality, to simplify statistical analysis and data interpretation.

Peak picking is the first step in data preprocessing. In this step, every peak (ion) detected in the sample is defined (assigned) by its m/z value and, in the case of LC- or GC-MS, by retention time (RT).

To compare multiple LC- or GC-MS analytical runs, a peak alignment step must be used. In a perfect case, the same metabolites from different samples, detected by the same analytical system, should have the same retention time and m/z (molecular mass). Due to the experimental variation and minor changes in analytical conditions, that is not usually the case. The objective of peak alignment is to distinguish the peaks of the same molecule, occurring in different samples, from the thousands of peaks detected during the course of an experiment. Alignment is therefore applied to correct retention times and to detected molecular mass variations between runs. Various strategies are used to carry out this step such as aligning detected peaks by the match score (join aligner), or by comparing individual peak lists with the master peak list, which is usually generated from the average of all aligned peak lists (Katajamaa and Orešič, 2007). Additional methods that may help to resolve and sort the data include summation or binning of chromatographic data, curve resolution, and gap filing (Nordström et al., 2006). The gap filling is particularly important step in analysis of larger number of very complex samples as peaks can be missed during peak picking step in some samples, primarily due to low intensity or/and irregular chromatographic peak distribution. When this occurs, preprocessing algorithms can be employed to search for the peak structures in the raw data and assign missing values (Karaman, 2017).

NMR data preprocessing

Currently, many software packages are available (**Table 1**) to transform NMR raw data to an appropriate form for the subsequent processing steps, including baseline correction, peak alignment, and binning.

Table 1 Software for metabolomics data processing

Name	Free or vendor	Description	Website
ACD	ACD Labs	Metabolomics with NMR data	http://www.acdlabs.com/resources/freeware/nmr_proc/
Analyst	SCIEX	Metabolomics with LC-MS data	https://sciex.com/products/software/analyst-software
Chenomx NMR MarkerView	Chenomx AB Sciex	Metabolomics with NMR data Metabolomics with LC-MS data	http://www.chenomx.com/ https://sciex.com/products/software/markerview-software
MassHunter	Agilent	Metabolomics with all MS data	http://www.agilent.com/en/products/software-informatics/masshunter-suite/masshunter#0
MATLAB	The MathWorks Inc.	Metabolomics with MS and NMR data	https://www.mathworks.com/help/bioinfo/mass-spectrometry-and-bioanalytics.html
MET-IDEA	Free	Metabolomics with CE-MS, GC-MS and LC-MS data	http://bioinfo.noble.org/download/
MetaboAnalyst	Free	Metabolomics with MS and NMR data	http://www.metaboanalyst.ca/
metAlign	Free	Metabolomics with GC-MS and LC-MS data	https://www.metalalign.nl
MestReNova	Mestrelab Research	Metabolomics with LC-MS, GC-MS and NMR data	http://mestrelab.com/
MNova	Mestrelab	Metabolomics with NMR data	http://mestrelab.com/software/mnova/nmr/
MS Resolver	Pattern Recognition Systems AS	Metabolomics with GC-MS and LC-MS data	http://www.prs.no/MS%20Resolver/MS%20Resolver.html
MSFACTs	Free	Metabolomics with GC-MS and LC-MS data	http://bioinfo.noble.org/download/
Mzmine	Free	Metabolomics with LC-MS data	http://mzmine.github.io/
PERCH	PERCH Solution Ltd	Metabolomics with NMR data	http://www.perchsolutions.com/
Progenesis QI	Waters	Metabolomics with LC-MS data	http://www.nonlinear.com/progenesis/qi/
TopSpin	Bruker. Free for academia and governmental institutions only	Metabolomics with NMR data	https://www.bruker.com/products/mr/nmr/nmr-software/software/topspin/overview.html
XCMS online	Free	Metabolomics with GC-MS and LC-MS data	https://xcmsonline.scripps.edu/

Baseline correction is applied to differentiate real peaks from instrumental noise. The most used baseline correction approaches is the frequency or time domain method, which can be easily applied to metabolomics spectra, subtracting a baseline estimation from the measured spectrum (Euceda *et al.*, 2015). Additional procedures that employ steps such as noise removal and/or smoothing include iterative polynomial fitting and asymmetric least squares smoothing (AsLS) (Euceda *et al.*, 2015).

During a relatively long time of NMR data acquisition process, resulting chemical shift can be affected by changes in pH, temperature and/or molecular interactions in the samples. This, in turn, creates a small variation in ppm spectra in different samples. Peak alignment methods are used to correct these differences. The alignment can be global (parametric time warping – PTW), in which the total spectrum is corrected, or the alignment can be carried out in each segment separately (correlation optimized warping – COW; icoshift) (Euceda *et al.*, 2015; Karaman, 2017).

The binning step is important in facilitating data analysis through the reduction in the number of variables. In this case, the spectra generated in NMR are segmented into various bins and signal in each bin is calculated by the integration. This may, however, lead to the loss of resolution or/and to the risk of splitting a peak (Vu and Laukens, 2013; Karaman, 2017). Therefore, depending on the data, binning methods can be of equal size (0.005 and 0.05 ppm), or bin size can be determined for each interval, and the bins can represent either individual peaks or a groups of peaks (Euceda *et al.*, 2015; Karaman, 2017).

Normalization

To minimize experimental variation in multi-sample quantitative analyses, it is necessary to normalize the data. Although several normalization methods including auto-scaling, use of a reference sample, log linear model, trimmed constant mean, and average intensity are available in software packages, the choice of the best method depends on the type of study and complexity of the samples. For example, by the addition of internal or external standards – samples can be normalized using the peak area/intensity of the known standard (Karaman, 2017). However, this method controls for variation due to sample preparation and analysis (technical variation), but not the sampling itself. Statistical models, on the other hand, allow one to normalize each sample with optimal scaling factors based on the complete database (Crawford and Morrison, 1968) in accordance with the unit vector norm generated by scaling each sample vector (variable – ion) (Scholz *et al.*, 2004) or median (Wang *et al.*, 2003) intensities, or by the maximum likelihood method (Orešič *et al.*, 2004), without any internal/external standards.

Data pretreatment

Another very useful tool, particularly for extraction of relevant biological information and for results interpretation, is data pretreatment (van den Berg *et al.*, 2006). Complex samples contain thousands of metabolites with concentrations covering enormous dynamic range (difference between low abundant and high abundant metabolites). Problem is that changes in cellular function (phenotype) is often not proportional to the fold change and overall abundance of the metabolites. For example, regulatory molecules usually exist in very low concentrations and small fold change have tremendous impact on the cell behavior (induced proliferation, cell survival etc.). Structural molecules, on the other hand, can be present in high concentrations or significant fold change may have only a minimal impact for overall cellular functions. To correct for these disproportion, data pretreatment methods can be used. These methods usually consist of centering, scaling, and data transformation. In centering, the mean that centers the differences between the samples (high and low abundant metabolites) is adjusted, leaving only the relevant variation (van den Berg *et al.*, 2006). In scaling, each variable is divided by the scaling factor, leading to an adjustment in the fold differences in the various metabolites (van den Berg *et al.*, 2006). Scaling can be based on data dispersion measured as a scaling factor. In autoscaling, metabolites are scaled based on standard deviation and all metabolites become equally important. Pareto scaling, on the other hand, reducing the relevant importance of large values. It is based on standard deviation, however, larger changes are scaled less than small changes. Similarly, Vast scaling focusing primarily on metabolites with small fluctuations. Level scaling uses the mean concentration as the scaling factor emphasizing relative response and therefore ideal for biomarker discovery (van Ginneken *et al.*, 2007; Karaman, 2017). Data transformation consists of eliminating heteroscedastic noise. (van Ginneken *et al.*, 2007). The most common transformations used are nonlinear transformations, such as log transformation, for which multiplicative noise is converted to additive noise, and power transformations, in which the original data is replaced by the calculated square root. (van Ginneken *et al.*, 2007; Karaman, 2017). It is important to note, that the selection of the data pretreatment methods can significantly affect the interpretation of results the metabolomics studies, and strongly depends on the biological question to be answered, the properties of the data, and the statistical analysis applied to the pretreated data. Inappropriate choice of these methods can lead to data misinterpretation.

Statistical Analysis

To find significant differences among the sample groups, basic statistical methods are usually a part of metabolomics data analysis software packages. Statistical significance tests identify data elements that make significant contributions to the protein and/or metabolite profile of a sample, or that distinguish a group of samples from others.

Phenotypic grouping is also an important tool for the investigation of the underlying interrelationships within a large data set. Statistical methodologies that identify these groups or clusters are known as clustering techniques, and originate from multivariate statistics. Hierarchical clustering can be used to sort data out into previously unknown clusters. Clustering objects on subsets of attributes (COSA) is an unsupervised method to cluster samples, that is an enhancement to distance based clustering methods (Friedman and Meulman, 2004). COSA detects groups of objects that have preferentially close values in different, possibly overlapping, subsets of attributes. Other clustering methods such as Principal Component Analysis (PCA), k-means clustering, and self-organizing maps (SOM) are also included in many software packages. Multivariate analyses such as Principal Component Analysis (PCA), Partial Least Squares-Discriminant Analysis (PLS-DA), and Orthogonal Projection to Latent Structures (OPLS-DA), can be employed to corroborate applied univariate analyses, such as t-tests, Kruskal Wallis tests, Mann Whitney tests, and ANOVA (Alonso *et al.*, 2015). Unsupervised methods are often used to explore, summarize and find hidden relation between the data (Liland, 2011). For example, to detect pattern correlated with experimental conditions or disease, or to assess data quality, PCA can be used. In this case, PCA is considered as the unsupervised methods pointing out various sample patterns (Liland, 2011).

PLS-DA and OPLS-DA, on the other hand, are supervised methods (OPLS-DA was developed to improve PLS-DA) used to discriminate known classes (e.g. control vs disease). Unlike unsupervised methods, these methods are applied to classify, predict, and discover potential biomarkers (Ren *et al.*, 2015).

Metabolite Identification and Pathway Visualization

Metabolite identification is necessary for untargeted metabolite analysis, since the metabolites of interest are unknown when the biological problem is defined. The list of possible metabolites can be generated using free databases and libraries, using either measured molecular mass (m/z) of the unknown metabolites and their retention time (for LC- or GC-MS data), or from chemical shifts and coupling constants (for NMR data) (Moco *et al.*, 2007). Common databases and libraries used in metabolomics include the Human Metabolome Database (HMDB), Spectral Database for Organic Compounds (SDBS), Comprehensive Species-Metabolite Relationship Database (KNapSack), Metlin, Golm Metabolome Database, MassBank, MassTRIX, PubChem, ChEMBL, Chemical Entities of Biological Interest (ChEBI), BioMagResBank, and Kyoto Encyclopedia of Genes and Genomes (KEGG) (Fig. 4) (Moco *et al.*, 2007; Klassen *et al.* 2017). For LC-MS experiments, the molecular mass of the metabolite is not sufficient for final identification as many compounds may have the same molecular mass, but differ in atomic composition and structure. Therefore, additional experiments using MS/MS techniques are necessary to analyze the fragmentation patterns of the metabolites of interest and their correctly identify them.

The final step in data analysis and interpretation is the visualization of the results, and identification of pathways associated with the biological response or regulation. The databases commonly used are the Kyoto Encyclopedia of Genes and Genomes (KEGG), MetaCyc, the Small Molecule Pathway Database (SMPDB), MetaboLights, and Reactome (Fig. 4).

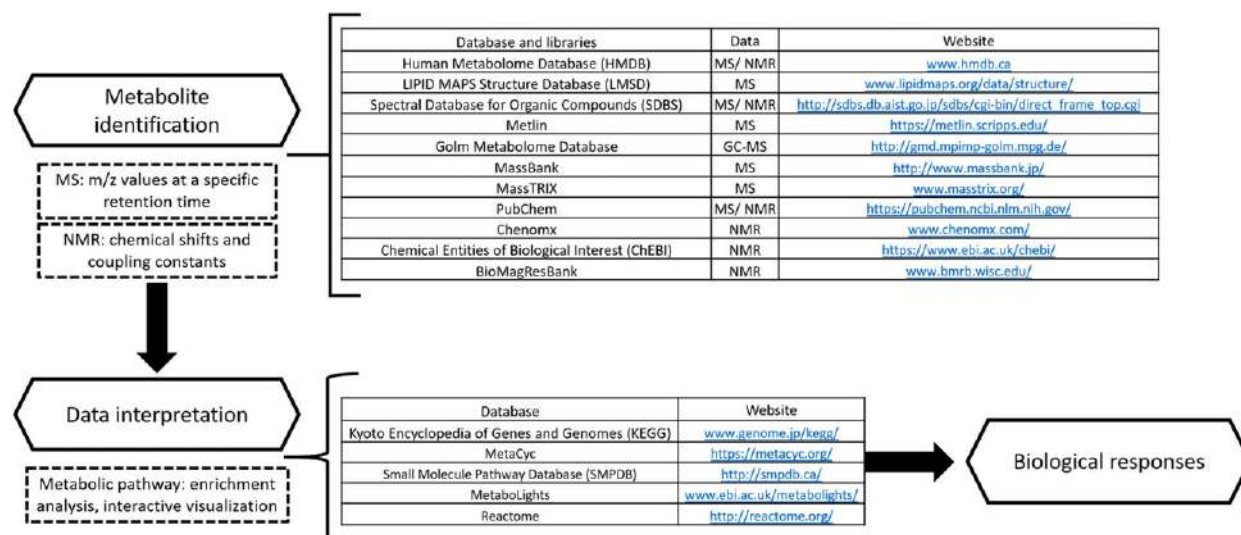


Fig. 4 Metabolite identification and data interpretation.

Applications

Metabolomics analyses have been applied in various research fields, such as medical research (diseases), plant biology, microbiology, toxicology, sustainable energy research etc. The focus primarily centers on the changes in the metabolite concentration, or their flux in specific pathways. Metabolomics data is important to the discovery of new biomarkers that can be associated with early diagnosis, progression, response to treatment, and to understand basic cellular processes and their regulation.

Metabolomics in Disease

In cancer, metabolomic analysis can be used to help to directly diagnose cancer stages from bodily fluids or histological samples. Colorectal cancer (CRC) is the third most commonly diagnosed cancer after lung and breast cancers; it also represents the fourth highest cancer-related cause of death (Siegel *et al.*, 2017).

Qiu *et al.* (2009) combined two analytical techniques, GC-MS and LC-MS, to analyze the serum metabolome of patients with CRC. Collected serum samples were spiked with internal standards L-2-chlorophenylalanine and heptadecanoic acid, and metabolites were extracted with methanol/chloroform (3:1 ratio (v/v)). First, the sample aliquots were derivatized using methoxyamine and BSTFA, followed by injection and metabolite separation using a DB-5 ms capillary column (30 m × 250 μm, 0.25 μm film thickness) in GC-MS system (Agilent 6890N gas chromatography instrument coupled with a Pegasus HT time-of-flight mass spectrometer from Leco Corporation). MATLAB 7.0 was used to convert the acquired data to netCDF format, and the ChromaTOF software was used to perform data processing. For LC-MS (Acquity ultra performance liquid chromatography MS system, Waters) analysis, the serum samples were spiked with the internal standard L-2-chlorophenylalanine and metabolites extracted using methanol/acetonitrile (5:3 ratio (v/v)). Samples were injected into a 1.7-μm BEH C18 column and run in both positive and negative mode. Data acquisition was performed using the MassLynx software. The SIMCA-P software was used for PCA and OPLS-DA statistical analysis, in which CRC and healthy controls patients were compared. NIST MS Search 2.0 was used for metabolite identification. Five metabolites, including pyruvate, lactate, tryptophan, tyrosine, and uridine, were identified by both analytical techniques as potential biomarkers, and indicated perturbation in glycolysis, arginine, and proline metabolism, as well as in the metabolism of fatty acids (Qiu *et al.*, 2009).

Urine is a biofluid commonly used in metabolomic analysis to identify biomarkers. Using urine samples and both GC-MS and LC-MS platforms, Cheng *et al.* (2012) identified citrate, hippurate, p-cresol, 2-aminobutyrate, myristate, putrescine, and kynurenate as potential biomarkers of CRC. Results suggest that the urinary metabolome of patients with CRC is potentially altered by metabolites derived from gut microbe-host co-metabolism (Cheng *et al.*, 2012).

In another study, using urine samples from 401 patients with clinically diagnosed Parkinson's disease, 106 patients with idiopathic Parkinson's disease, and 104 healthy control participants, metabolome analysis was carried out by LC-MS (Luan *et al.*, 2015). Urine samples were precipitated with methanol, and the supernatant injected in an LC-MS (Shimadzu Prominence LC system) coupled online to an LTQ Orbitrap Velos instrument (Thermo-Fisher Scientific), with which samples were analyzed in both positive and negative mode. Data analysis was conducted using XCMS and CAMERA software, with the R statistical package. The OPLS-DA and Human Metabolome Database were used to identify possible biomarkers, which included cortisol, 11-deoxycortisol, 21-deoxycortisol, histidine, urocanic acid, imidazoleacetic acid, and hydroxyphenylacetic acid (Luan *et al.*, 2015).

Rizza *et al.* (2014) used metabolic profiling to identify predictors of major cardiovascular events in elderly people. In this study, the samples were extracted with 75% methanol and 0.01% oxalic acid containing stable isotope internal standards. The samples were analyzed by direct infusion mass spectrometry (DIMS) using a Quattro Ultima Pt ESI tandem quadrupole mass spectrometer (Waters). Data analysis was performed with MassLynx v4.0 and NeoLynx software. The PCA statistical analysis revealed medium- and long-chain acylcarnitines, and alanine as potential biomarkers (Rizza *et al.*, 2014).

NMR was used to identify biomarkers of depressive disorder in urine samples (Zheng *et al.*, 2013). The study involved 82 first-episode drug-naïve patients and 82 healthy controls. Urine samples were first centrifuged, and the supernatant mixed with phosphate buffer in 90% D₂O. Data was collected on an Avance II 600 (Bruker) spectrometer using a frequency of 600.13 MHz ¹H and processed by the AMIX package (Bruker). Supervised OPLS-DA with a multivariate approach revealed formate, malonate, N-methylnicotinamide, m-hydroxyphenylacetate, and alanine as biomarkers of the disease (Zheng *et al.*, 2013).

Metabolomics in Plant Biology

Metabolomics studies in plants typically investigate the presence of biomarkers related to plant development, growth, and response to different environmental conditions and stressors.

Vaclavik *et al.* (2013) used two platforms, the Acquity UPLC system equipped with a reversed-phase analytical column for UHPLC–QqQ/LIT-MS and an AccuTOF–LP-TOF-MS for direct analysis in real time – mass spectrometry (DART–MS), to identify biomarkers of cold tolerance (Vaclavik *et al.*, 2013). In this study, metabolic fingerprints of three groups including a control group (plants growing at room temperature), cold acclimated group (plants exposed to –4°C for 2 weeks), and slow cooling group (plants exposed to –4°C for 8 h) of *Arabidopsis thaliana* were compared and evaluated. The metabolites were extracted using methanol (50 mg tissue/mL) and analyzed by UHPLC–QqQ/LIT MS. The acquired data was processed by the MarkerView software (AB SCIEX), using PCA and a *t*-test to compare the peak intensities among the groups. In addition, the DART–MS data was processed by the Mass Center software (version 1.3, JEOL), and STATISTICA software (version 8.0, StatSoft) was used for the PCA statistical analysis. To identify the structure of significantly affected metabolites, the authors used the MassLynx software (Waters), and isotope distribution, to estimate elemental composition (Vaclavik *et al.*, 2013). Whilst gluconapin was demonstrated to be biomarker of the cold sensitive phenotype, kaempferol-3,7-O-dirhamnoside and kaempferol-3-neohesperidoside-7-O-rhamnoside were identified as the biomarkers of high cold tolerance (Vaclavik *et al.*, 2013).

The plant metabolomics is associated with a high diversity of secondary metabolites also known as phytochemicals. Nakabayashi *et al.* (2009) studied *Arabidopsis thaliana* as a model for genome research, using non targeted metabolomics by 1D and 2D NMR, HRFABMS, FT-ESI-MS and GC–TOF-MS techniques. To maximize metabolome coverage, reduce the sample complexity, and enrich specific classes of metabolites, the authors used a series of extraction methods including methanol, n-hexane, ethyl acetate, chloroform containing 5% methanol and n-butanol. For example, the anthocyanin fraction was obtained by n-butanol extraction. A total of 37 compounds such as apocarotenoids, anthocyanins, carotenoids, chlorophyll derivatives, dicarboxylic acids, flavonols, galactolipids, glucosinolate, icosane, steroid, phenylpropanoids, and nucleosides were identified in this study (Nakabayashi *et al.*, 2009).

Metabolomics in Microbiology

Metabolomics in microbiology is a very important tool for understanding basic biological questions such as those for which microbes have been used as model organisms for many years. Recently, many studies also have focused on the improvement of microbes to produce alternative sources of energy, or to better understand microbial resistance to various antibiotics and identify new drug targets.

Ledala *et al.* (2014) used NMR as an analytical technique to investigate influence of iron and oxygen limitations on the metabolism of *Staphylococcus aureus*. Samples from two growth phases were harvested in this study, exponential (2 h) and post-exponential (6 h). Altogether, eight and four replicates, respectively, were used for one-dimensional ¹H NMR and two-dimensional ¹H-¹³C heteronuclear single quantum coherence (¹H-¹³C HSQC) analyses. The samples were quenched in liquid nitrogen, and bacteria were collected using ice-cold 20 mM phosphate buffer. TMSP-d₄ (3-(trimethylsilyl)propionic acid-2,2,3,3-d₄) was used as an internal standard for both ¹H NMR and ¹H-¹³C HSQC experiments. For ¹H NMR, the data was processed and analyzed by ACD/1D NMR manager version 12.0 (Advanced Chemistry Development), and ¹H-¹³C HSQC data were processed using the NMRPipe software package. To identify the significant metabolite changes, *t*-tests, PCA, OPLS-DA, and unique structure (SUS) calculations were performed using the SIMCA 12.0_statistical package (Umetrics). Metabolite identification was achieved by comparison of measured chemical shifts with the chemical shift references from Human Metabolomics Database, Platform for RIKEN Metabolomics (PRIME), and Biological Magnetic Resonance Data Bank (BMRB). The most significant changes in metabolite concentrations were found in the post-exponential phase, and were associated with the TCA cycle (Ledala *et al.*, 2014).

Stipetic *et al.* (2016) investigated differences in the metabolome of the planktonic and biofilm states of *Staphylococcus aureus*. The sample extraction was performed using chloroform/methanol/water (1:3:1 ratio (v/v/v)), followed by LC-MS analysis using a hydrophilic interaction liquid chromatography (HILIC). The MS data was analyzed using several packages including XCMS (peak picking), mzMatch (filtering and grouping), and R-scripts (filtering, post-processing and identification). In the statistical analysis step, PCA was implemented to compare the two states. Affected pathways were visualized in the KEGG, and suggested that the most prominent changes are in arginine biosynthesis (Stipetic *et al.*, 2016).

Closing Remarks

Metabolomics is the most recent OMICS strategy, and is being used to understand the molecular mechanisms of cellular processes in various biological systems. Whilst proteomics data determines the overall concentration of proteins in biological systems, it may not reflect their actual activities, as protein activities depend not only on their concentrations, but also on post-translational modification(s), protein-protein interactions, binding of activators, repressors, or co-factors, or changes in tertiary structure. Therefore metabolomics, which directly measures these activities, brings a new dimension to our knowledge, helps to explain many regulatory events, and can map the complex relationships of individual molecules and pathways to predict cellular behavior and responses to various stimuli. Like the other OMICS techniques, the results of metabolomics strongly depend on the selected methodology. GC-MS and LC-MS approaches detect different classes of metabolites, and should be used as complementary techniques to increase metabolome coverage. Extraction methods can be used to decrease the complexity of the samples and focus on very specific molecules. Therefore, the choice of individual methods is the most critical aspect of metabolomics experimental design, and must be driven by biological questions. In summary, metabolomics represents a very valuable tool, however, careful selection of individual analytical and statistical steps is essential to understand the results and avoid their misinterpretation.

See also: Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics

References

- Adamec, J., Jannasch, A., Huang, J., *et al.*, 2011. Development and optimization of an LC-MS/MS-based method for simultaneous quantification of vitamin D(2), vitamin D(3), 25-hydroxyvitamin D(2) and 25-hydroxyvitamin D(3). *Journal of Separation Science* 34, 11–20.
- Alonso, A., Marsal, S., Julià, A., 2015. Analytical methods in untargeted metabolomics: State of the art in 2015. *Frontiers in Bioengineering and Biotechnology* 3, 1–20.
- Atherton, H.J., Jones, O.A.H., Malik, S., Miska, E.A., Griffin, J.L., 2008. A comparative metabolomic study of NHR-49 in *Caenorhabditis elegans* and PPAR- α in the mouse. *FEBS Letters* 582, 1661–1666.
- Beckonert, O., Keun, H.C., Ebbels, T.M.D., *et al.*, 2007. Metabolic profiling, metabolomic and metabonomic procedures for NMR spectroscopy of urine, plasma, serum and tissue extracts. *Nature Protocols* 2, 2692–2703.
- Brindle, J.T., Antti, H., Holmes, E., *et al.*, 2002. Rapid and noninvasive diagnosis of the presence and severity of coronary heart disease using ^1H NMR-based metabolomics. *Nature Medicine* 8, 1439–1445.
- Brockmöller, S.F., Bucher, E., Müller, B.M., *et al.*, 2012. Integration of metabolomics and expression of glycerol-3-phosphate acyltransferase (GPAM) in breast cancer-link to patient survival, hormone receptor status, and metabolic profiling. *Journal of Proteome Research* 11, 850–860.
- Bruce, S.J., Jonsson, P., Antti, H., *et al.*, 2008. Evaluation of a protocol for metabolic profiling studies on human blood plasma by combined ultra-performance liquid chromatography/mass spectrometry: From extraction to data analysis. *Analytical Biochemistry* 372, 237–249.
- Cambiaghi, A., Ferrario, M., Masseroli, M., 2017. Analysis of metabolomic data: Tools, current strategies and future challenges for omics data integration. *Briefings in Bioinformatics* 18, 498–510.
- Capati, A., Ijare, O.B., Bezabeh, T., 2017. Diagnostic applications of nuclear magnetic resonance-based urinary metabolomics. *Magnetic Resonance Insights* 10, 1–12.
- Cheng, Y., Xie, G., Chen, T., *et al.*, 2012. Distinct urinary metabolic profile of human colorectal cancer. *Journal of Proteome Research* 11, 1354–1363.
- Crawford, L.R., Morrison, J.D., 1968. Computer methods in analytical mass spectrometry. Identification of an unknown compound in a catalog. *Analytical Chemistry* 40, 1464–1469.
- Dória, M.L., Cotrim, Z., Macedo, B., *et al.*, 2012. Lipidomic approach to identify patterns in phospholipid profiles and define class differences in mammary epithelial and breast cancer cells. *Breast Cancer Research and Treatment* 133, 635–648.
- Drogan, D., Dunn, W.B., Lin, W., *et al.*, 2015. Untargeted metabolic profiling identifies altered serum metabolites of type 2 diabetes mellitus in a prospective, nested case control study. *Clinical Chemistry* 61, 487–497.
- Euceda, L.R., Giskegærd, G.F., Bathen, T.F., 2015. Preprocessing of NMR metabolomics data. *Scandinavian Journal of Clinical and Laboratory Investigation* 75, 193–203.
- Fiehn, O., 2001. Combining genomics, metabolome analysis, and biochemical modelling to understand metabolic networks. *Comparative and Functional Genomics* 2, 155–168.
- Friedman, J.H., Meulman, J.J., 2004. Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 66, 815–839.
- Garcia, A., Barbas, C., 2011. Gas chromatography-mass spectrometry (GC-MS)-based metabolomics. In: Metz, T.O. (Ed.), *Metabolic Profiling: Methods and Protocols*. Totowa, NJ: Humana Press, pp. 191–204.
- Gebregiworgis, T., Powers, R., 2012. Application of NMR metabolomics to search for human disease biomarkers. *Combinatorial Chemistry & High Throughput Screening* 15, 595–610.
- Goodacre, R., Broadhurst, D., Smilde, A.K., *et al.*, 2007. Proposed minimum reporting standards for data analysis in metabolomics. *Metabolomics* 3, 231–241.
- Gowda, G.A.N., Zhang, S., Gu, H., *et al.*, 2008. Metabolomics-based methods for early disease diagnostics: A review. *Expert Review of Molecular Diagnostics* 8, 617–633.
- Heude, C., Nath, J., Carrigan, J.B., Ludwig, C., 2017. Nuclear magnetic resonance strategies for metabolic analysis. In: Sussulini, A. (Ed.), *Metabolomics: From Fundamentals to Clinical Applications*. Cham: Springer International Publishing, pp. 45–76.
- Hoffmann, E. de., 1996. Tandem mass spectrometry: A primer. *Journal of Mass Spectrometry* 31, 129–137.
- Huang, Y., Cai, S., Zhang, Z., Chen, Z., 2014. High-Resolution two-dimensional J-resolved NMR spectroscopy for biological systems. *Biophysical Journal* 106, 2061–2070.
- Huang, Y., Zhang, Z., Chen, H., *et al.*, 2015. A high-resolution 2D J-resolved NMR detection technique for metabolite analyses of biological samples. *Scientific Reports* 5, 8390.
- Jorge, T.F., Mata, A.T., António, C., 2016. Mass spectrometry as a quantitative tool in plant metabolomics. *Philosophical Transactions of the Royal Society A* 374, 20150370.
- Jorge, T.F., Rodrigues, J.A., Caldana, C., *et al.*, 2016. Mass spectrometry-based plant metabolomics: Metabolite responses to abiotic stress. *Mass Spectrometry Reviews* 35, 620–649.
- Karaman, I., 2017. Preprocessing and pretreatment of metabolomics data for statistical analysis. In: Sussulini, A. (Ed.), *Metabolomics: From Fundamentals to Clinical Applications*. Springer International Publishing, pp. 145–161.
- Katajamaa, M., Orešić, M., 2007. Data processing for mass spectrometry-based metabolomics. *Journal of Chromatography A* 1158, 318–328.
- Kell, D.B., Brown, M., Davey, H.M., *et al.*, 2005. Metabolic footprinting and systems biology: The medium is the message. *Nature Reviews Microbiology* 3, 557–565.

- Klassen, A., Faccio, A.T., Canuto, G.A.B., *et al.*, 2017. Metabolomics: Definitions and significance in systems biology. In: Sussulini, A. (Ed.), *Metabolomics: From Fundamentals to Clinical Applications*. Springer International Publishing, pp. 3–17.
- Kopka, J., Fernie, A., Weckwerth, W., Gibon, Y., Stitt, M., 2004. Metabolite profiling in plant biology: Platforms and destinations. *Genome Biology* 5, 109.
- Ledala, N., Zhang, B., Seravalli, J., Powers, R., Somerville, G.A., 2014. Influence of iron and aeration on *Staphylococcus aureus* growth, metabolism, and transcription. *Journal of Bacteriology* 196, 2178–2189.
- Li, N., Liu, W., Li, W., *et al.*, 2010. Plasma metabolic profiling of Alzheimer's disease by liquid chromatography/mass spectrometry. *Clinical Biochemistry* 43, 992–997.
- Li, X., Long, D., Ji, J., *et al.*, 2013. Sample preparation for the metabolomics investigation of poly- γ -glutamate-producing *Bacillus licheniformis* by GC–MS. *Journal of Microbiological Methods* 94, 61–67.
- Liland, K.H., 2011. Multivariate methods in metabolomics – From pre-processing to dimension reduction and statistical analysis. *TrAC Trends in Analytical Chemistry* 30, 827–841.
- Lin, Y., Ma, C., Liu, C., *et al.*, 2016. NMR-based fecal metabolomics fingerprinting as predictors of earlier diagnosis in patients with colorectal cancer. *Oncotarget* 7, 29454.
- Lopes, A.S., Cruz, E.C.S., Sussulini, A., Klassen, A., 2017. Metabolomic strategies involving mass spectrometry combined with liquid and gas chromatography. In: Sussulini, A. (Ed.), *Metabolomics: From Fundamentals to Clinical Applications*. Springer International Publishing, pp. 77–98.
- Luan, H., Liu, L.F., Meng, N., *et al.*, 2015. LC-MS-based urinary metabolite signatures in idiopathic Parkinson's disease. *Journal of Proteome Research* 14, 467–478.
- Lv, W., Yang, T., 2012. Identification of possible biomarkers for breast cancer from free fatty acid profiles determined by GC–MS and multivariate statistical analysis. *Clinical Biochemistry* 45, 127–133.
- Moco, S., Vervoort, J., Moco, S., *et al.*, 2007. Metabolomics technologies and metabolite identification. *TrAC – Trends in Analytical Chemistry* 26, 855–866.
- Nakabayashi, R., Kusano, M., Kobayashi, M., *et al.*, 2009. Metabolomics-oriented isolation and structure elucidation of 37 compounds including two anthocyanins from *Arabidopsis thaliana*. *Phytochemistry* 70, 1017–1029.
- Nicholson, J.K., Holmes, E., Lindon, J.C., 2007. Metabonomics and metabolomics techniques and their applications in mammalian systems. In: Nicholson, J.K., Holmes, E. (Eds.), *The Handbook of Metabonomics and Metabolomics*. Amsterdam: Elsevier Science B.V, pp. 1–33.
- Nicholson, J.K., Lindon, J.C., Holmes, E., 1999. Metabonomics: Understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological NMR spectroscopic data. *Xenobiotica* 29, 1181–1189.
- Nishiumi, S., Kobayashi, T., Ikeda, A., *et al.*, 2012. A novel serum metabolomics-based diagnostic approach for colorectal cancer. *PLOS ONE* 7, e40459.
- Nordström, A., O'Maille, G., Qin, C., Siuzdak, G., 2006. Non-linear data alignment for UPLC-MS and HPLC-MS based metabolomics: Application to endogenous and exogenous metabolites in human serum. *Analytical Chemistry* 78, 3289–3295.
- Oliver, S.G., Winson, M.K., Kell, D.B., Baganz, F., 1998. Systematic functional analysis of the yeast genome. *Trends in Biotechnology* 16, 373–378.
- Orešić, M., Clish, C.B., Davidov, E.J., *et al.*, 2004. Phenotype characterisation using integrated gene transcript, protein and metabolite profiling. *Applied Bioinformatics* 3, 205–217.
- Pan, Z., Raftery, D., 2007. Comparing and combining NMR spectroscopy and mass spectrometry in metabolomics. *Analytical and Bioanalytical Chemistry* 387, 525–527.
- Qiu, Y., Cai, G., Su, M., *et al.*, 2009. Serum metabolite profiling of human colorectal cancer using GC-TOFMS and UPLC-QTOFMS. *Journal of Proteome Research* 8, 4844–4850.
- Ramautar, R., Somsen, G.W., de Jong, G.J., 2009. CE-MS in metabolomics. *Electrophoresis* 30, 276–291.
- Reichel, C., 2011. OMICS-strategies and methods in the fight against doping. *Forensic Science International* 213, 20–34.
- Ren, S., Hinzman, A.A., Kang, E.L., Szczesniak, R.D., Lu, L.J., 2015. Computational and statistical analysis of metabolomics data. *Metabolomics* 11, 1492–1513.
- Reo, N.V., 2002. Nmr-based metabolomics. *Drug and Chemical Toxicology* 25, 375–382.
- Rizza, S., Copetti, M., Rossi, C., *et al.*, 2014. Metabolomics signature improves the prediction of cardiovascular events in elderly subjects. *Atherosclerosis* 232, 260–264.
- Roberts, L.D., Souza, A.L., Gerszten, R.E., Clish, C.B., 2012. Targeted metabolomics. *Current Protocols in Molecular Biology*. 1–24.
- Rodrigues, K.T., Cieslarová, Z., Tavares, M.F.M., Simionato, A.V.C., 2017. Strategies involving mass spectrometry combined with capillary electrophoresis in metabolomics. In: Sussulini, A. (Ed.), *Metabolomics: From Fundamentals to Clinical Applications*. Cham: Springer International Publishing, pp. 99–141.
- Sato, Y., Suzuki, I., Nakamura, T., *et al.*, 2012. Identification of a new plasma biomarker of Alzheimer's disease using metabolomics technology. *The Journal of Lipid Research* 53, 567–576.
- Scholz, M., Gatzek, S., Sterling, A., Fiehn, O., Selbig, J., 2004. Metabolite fingerprinting: Detecting biological features by independent component analysis. *Bioinformatics* 20, 2447–2454.
- Siegel, R.L., Miller, K.D., Fedewa, S.A., *et al.*, 2017. Colorectal cancer statistics, 2017. *CA: A Cancer Journal for Clinicians* 67, 177–193.
- Stentiford, G.D., Viant, M.R., Ward, D.G., *et al.*, 2005. Liver tumors in wild flatfish: A histopathological, proteomic, and metabolomic study. *Omics: A Journal of Integrative Biology* 9, 281–299.
- Stipetic, L.H., Dalby, M.J., Davies, R.L., *et al.*, 2016. A novel metabolomic approach used for the comparison of *Staphylococcus aureus* planktonic cells and biofilm samples. *Metabolomics* 12, 75.
- Tweeddale, H., Notley-McRobb, L., Ferenci, T., 1998. Effect of slow growth on metabolism of *Escherichia coli*, as revealed by global metabolite pool ('metabolome') analysis. *Journal of Bacteriology* 180, 5109–5116.
- Vaclavik, L., Mishra, A., Mishra, K.B., Hajslova, J., 2013. Mass spectrometry-based metabolomic fingerprinting for screening cold tolerance in *Arabidopsis thaliana* accessions. *Analytical and Bioanalytical Chemistry* 405, 2671–2683.
- van den Berg, R.A., Hoefsloot, H.C.J.H., Westerhuis, J.A., Smilde, A.K., van der Werf, M.J., 2006. Centering, scaling, and transformations: Improving the biological information content of metabolomics data. *BMC Genomics* 7, 142.
- van Ginneken, V., Verhey, E., Poelmann, R., *et al.*, 2007. Metabolomics (liver and blood profiling) in a mouse model in response to fasting: A study of hepatic steatosis. *Biochimica et Biophysica Acta (BBA) - Molecular and Cell Biology of Lipids* 1771, 1263–1270.
- Verpoorte, R., Choi, Y.H., Mustafa, N.R., Kim, H.K., 2008. Metabolomics: Back to basics. *Phytochemistry Reviews* 7, 525–537.
- Villas-Bôas, S.G., Bruheim, P., 2007. Cold glycerol-saline: The promising quenching solution for accurate intracellular metabolite analysis of microbial cells. *Analytical Biochemistry* 370, 87–97.
- Villas-Bôas, S.G., Mas, S., Akesson, M., Smedsgaard, J., Nielsen, J., 2005. Mass spectrometry in metabolome analysis. *Mass Spectrometry Reviews* 24, 613–646.
- Villas-Bôas, S.G., Roessner, U., Hansen, M.A., *et al.*, 2006. Sampling and sample preparation. In: Villas-Bôas, S.G., Nielsen, J., Smedsgaard, J., Hansen, M.A.E., Roessner-Tunali, U. (Eds.), *Metabolome Analysis: An Introduction*. Hoboken, NJ: John Wiley & Sons, Inc, pp. 39–82.
- Vu, T.N., Laukens, K., 2013. Getting your peaks in line: A review of alignment methods for NMR spectral data. *Metabolites* 3, 259–276.
- Wang, G., Zhou, Y., Huang, F.J., *et al.*, 2014. Plasma metabolite profiles of Alzheimer's disease and mild cognitive impairment. *Journal of Proteome Research* 13, 2649–2658.
- Wang, J., Zhang, S., Li, Z., *et al.*, 2011. 1H NMR-based metabolomics of tumor tissue for the metabolic characterization of rat hepatocellular carcinoma formation and metastasis. *Tumor Biology* 32, 223–231.
- Wang, W., Becker, C.H., Zhou, H., *et al.*, 2003. Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Analytical Chemistry* 75, 4818–4826.
- Wilkins, M.R., Sanchez, J.-C., Gooley, A.A., *et al.*, 1996. Progress with proteome projects: Why all proteins expressed by a genome should be identified and how to do it. In: *Proceedings of the Biotechnology and Genetic Engineering Reviews*, 13, pp. 19–50.
- Winder, C.L., Dunn, W.B., Schuler, S., *et al.*, 2008. Global Metabolic profiling of *Escherichia coli* cultures: An evaluation of methods for quenching and extraction of intracellular metabolites. *Analytical Chemistry* 80, 2939–2948.

- Zhang, A., Sun, H., Wang, P., Han, Y., Wang, X., 2012. Modern analytical techniques in metabolomics analysis. *The Analyst* 137, 293–300.
- Zheng, P., Wang, Y., Chen, L., *et al.*, 2013. Identification and validation of urinary metabolite biomarkers for major depressive disorder. *Molecular & Cellular Proteomics* 12, 207–214.

Further Reading

- Fan, T.W.M., Lane, A.N., Higashi, R.M., 2012. *The Handbook of Metabolomics*. Springer.
- Johnson, C.H., Ivanisevic, J., Siuzdak, G., 2016. Metabolomics: Beyond biomarkers and towards mechanisms. *Nature Reviews Molecular Cell Biology* 17, 451–459.
- Lindon, J.C., Nicholson, J.K., Holmes, E., 2011. *The handbook of Metabonomics and Metabolomics*. Elsevier.
- Smolinska, A., *et al.*, 2012. NMR and pattern recognition methods in metabolomics: From data acquisition to biomarker discovery: A review. *Analytica Chimica Acta* 750, 82–97.
- Zhou, B., Xiao, J.F., Tuli, L., Ransom, H.W., 2012. LC-MS-based metabolomics. *Molecular BioSystems* 8, 470–481.

Relevant Websites

- <http://www.cytoscape.org/>
Cytoscape.
- <http://www.hmdb.ca/>
Human Metabolome Database (HMDB).
- <http://www.genome.jp/kegg/>
Kyoto Encyclopedia of Genes and Genomes (KEGG).
- <http://www.metaboanalyst.ca/faces/home.xhtml>
MetaboAnalyst.
- <http://reactome.org/>
Reactome.

Disease Biomarker Discovery

Tiratha R Singh and Ankita Shukla, Jaypee University of Information Technology, Solan, India

Bensellak Taoufik and Ahmed Moussa, École Nationale Des Sciences Appliquées de Tanger, Tangier, Morocco

Brigitte Vannier, University of Poitiers, Poitiers France

© 2019 Elsevier Inc. All rights reserved.

Metabolic Networks: A Background for Biomarkers

Network biology is a branch of science that deals with the interactions among biomolecules that include genes, transcripts, proteins, metabolites, etc. With the advent of system biology, networks are being used widely across many branches of biology (proteomics, genomics, transcriptomics, and metabolomics) as a convenient representation of the interaction between specific biological elements. These graphical representations denote the molecular-level blueprint of interactions and mechanisms of regulation inside a cell. These biological networks include gene regulatory network, transcriptomic network, protein–protein interaction network, and metabolic network. The network biology approach helped to cover the overall aspects of the necessary facets that need to be considered while finding the probable therapeutic intervention for the particular disease type. The interaction data come through the high-throughput methods that are gathered from individual studies and large-scale screens that finally get assembled into a topological form (i.e., network format) that holds significant biological properties. In a recent scenario, more attention has been given to the gene and protein networks to study a complex form of diseases (Shukla and Singh, 2017). Although it has been found that the metabolic networks seem to play a significant role in the complex disease regulation like in case of cancer's Warburg effect (Vander Heiden *et al.*, 2009), which signifies uncontrolled cell division even in anaerobic conditions that involve numerous metabolites and the reaction mechanisms. The metabolic network comprises of metabolites and enzymes that take the role of nodes and the reactions describing their transformations and is represented as directed edges in Fig. 1 (Bourqui *et al.*, 2007).

Biochemical reactions happening inside a metabolic network allow an organism to grow, reproduce, and respond to the environment and maintain its structure (Xu *et al.*, 2016). In a biochemical pathway, the metabolic network centralizes its attention towards mass flow that generates essential components like amino acids, sugars, and lipids, and the energy required by the biochemical reactions (Zhu *et al.*, 2007). In a metabolic network, it's not only metabolites that perform the overall metabolism but there are genes and proteins too that commence their task in regulatory mechanisms; this is what makes the metabolic networks more efficient from the disease perspective and their immediate applications too for therapeutic interventions (Berkhout *et al.*, 2013). This shows that metabolic networks typically show the representation of not only metabolites but also for genes and proteins and therefore provide wide perspective in disease studies. In a cell, metabolism holds chemical processes by which cells break down food and nutrients into usable building blocks and then reassemble those building blocks to form the biological molecules known as metabolites (DeBerardinis and Thompson, 2012). The metabolites consumed are called the substrates of the reaction; however those produced are called the products. Most metabolic reactions do not occur spontaneously, or we can say that they occur at a very low rate; therefore enzymes are used to enhance the pace of the reaction to get it completed (Cooper, 2000). This breakdown and reassembly in a pathway entails a set of successive chemical reactions that convert initial inputs into useful end products via a series of steps and this complete set of reactions in the pathway forms the metabolic network (Sridharan *et al.*, 2015). To understand the interacting mechanism in a network it's necessary to understand the architecture of the network topology. In a metabolic network, nodes represent the chemicals produced and consumed by the reactions that include small molecules (i.e. carbohydrates, lipids, amino acids, and nucleotides), and the edges denote the metabolic flow or the regulatory effects of a specific reaction (Lee *et al.*, 2008). Understanding the complex network often requires a bottom-up approach that carries its path towards systems biology perspective (Shahzad and Loor, 2012). Thus there is need to examine a system, not only in terms of individual components but as a whole, which can be done by considering the elementary constituents individually as well as when they are connected. Numerous components of a system and their interactions are best characterized as networks and they are mainly represented as graphs where thousands of nodes are connected with thousands of vertices (Cho *et al.*, 2012).

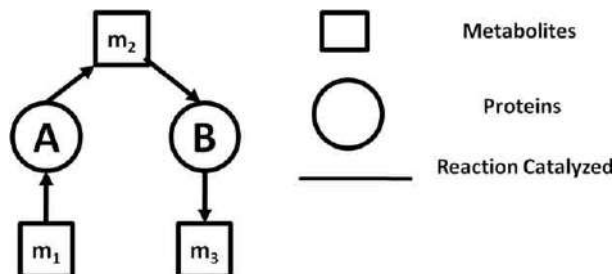


Fig. 1 Basic metabolic network.

Network analysis has suggested that biological networks have two imperative structural properties (Mahadevan and Palsson, 2005). First, it has been shown that several of these networks, including metabolic networks, are scale-free and possess a “small world” property (Barabasi and Oltvai, 2004). Second, scale-free networks are suggested to have high error tolerance and low attack tolerance (Crucitti *et al.*, 2004). In general, interactions in a network between biological entities (genes, transcripts, proteins, metabolites, etc.) can be classified on the basis of the nature of the interaction into two broad categories, i.e., influence networks and the flow networks. For influence networks, the nature of interactions are “influence-based” such as protein–protein interaction or signaling networks, i.e., connections mark the presence or absence of the reaction (Mahadevan and Palsson, 2005). This class might be extended with the case where the type of interaction is important in addition to presence or absence of the interaction, e.g., gene regulatory networks where transcription factors can either activate or repress gene expression. While in flow networks, where a specific variable like mass or energy flow may be conserved at each node, such as metabolic networks. However, it should be noted that the fundamental properties of biological networks in these two classes can be significantly different (Barabasi and Oltvai, 2004). It has been seen widely that along with the complete network graph, the subgraphs are also seen to play an essential functional role like network motifs (that represent most frequently occurring subgraph) and studies have shown the structural organization of the feedback loops (Fig. 2). Similarly, single-input and multiple-input motifs (Fig. 2) (Sehgal *et al.*, 2015) can influence the dynamics and the regulation of metabolic pathways (Beber *et al.*, 2012). For effective analysis it’s important to model the network graphs correctly for which there are a wide variety of approaches depending on the features of interest through which network dynamics can be modeled, like for small networks with explicit kinetics, which can be modeled with differential equations, while for larger networks dynamics can be accessed by flux balance analysis or stochastic kinetic modeling (Boccaletti *et al.*, 2006). Also, for the case where only stoichiometric information is available, more basic approaches like network expansion or Petri nets can be utilized (Peleg *et al.*, 2005).

Varieties of graph representations are available in network biology but studies have shown that bipartite graph (Fig. 3) (two nodes represent metabolites with edges joining each metabolite to the reaction) is the most correct representation of the metabolic network (Veeramani and Bader, 2010). The edges in the representative graphs are directed because some metabolites (the substrates) go into the reaction and some (the products) come out of it. Metabolic networks are represented through nodes as metabolites and the links as reactions that are catalyzed by specific gene products; this representation is different from protein–protein interaction networks, where the nodes are the gene products and the links correspond to interactions. The analysis of protein–protein interaction networks has suggested that the deletion of the most highly connected proteins correlates well with a lethal phenotype (Mahadevan and Palsson, 2005). In contrast, a node in metabolic networks cannot be deleted by genetic techniques, but links can (Jeong *et al.*, 2001).

Now the question comes as to how the resultant computational network are formed and how their global representation is possible. The answer lies in the computer readable file formats for the biological networks, i.e., Systems Biology Markup Language (SBML), a global format which could be utilized for the reusability of network models. It is a XML-like machine-readable language that is proficient to represent models to be analyzed by a computer. SBML can represent metabolic networks, cell signaling pathways, regulatory networks, and many other kinds of systems (Hucka *et al.*, 2003). An increasing number of diseases are now

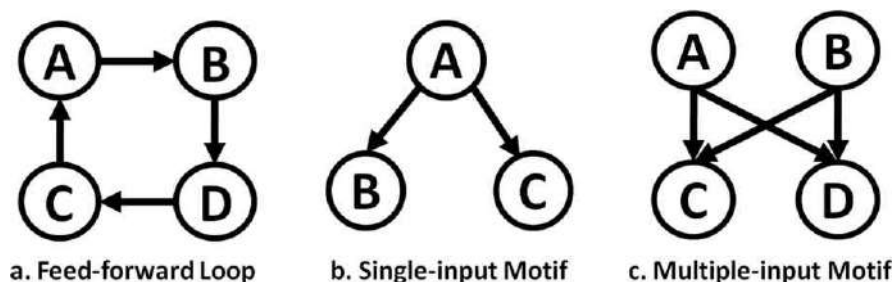


Fig. 2 Most frequent regulatory motif.

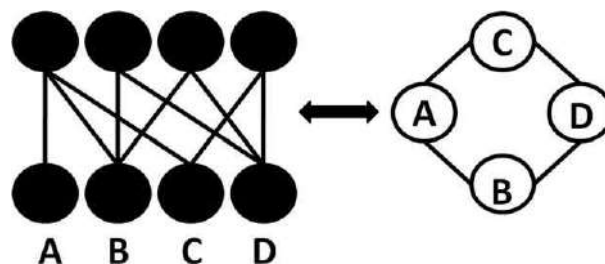


Fig. 3 Bipartite graph with corresponding simple representation.

seen to be a result of drastic perturbations of cellular functions that involve large sets of genes, where connections are complicated to understand. Diseases in particular like cancer, CVD, or diabetes causes huge perturbations in cell metabolism. Therefore, the study of metabolic networks primary perturbation in interactions and fluxes can aid better understanding of the physiopathology of such diseases. It notably permits an understanding of how and why alteration in activity or expression levels of a few enzymes can transmit substantial perturbations into entire cellular functions. Therefore these metabolites could have been proposed as putative biomarkers for the human diseases.

Metabolome Analysis: Computational Protocols and Methods

The metabolome comprises of the biochemical composition of the small-molecule metabolites present in a cell that are involved in the metabolic reaction mechanism and required for the maintenance, growth, and normal functioning of the cell (Mohny and Milburn, 2015). The metabolome was first described by Oliver and colleagues in 1998 (Oliver *et al.*, 1998), during their pioneering work on yeast metabolism that therefore confers the discipline of metabolomics; that follows the analysis of the metabolome. Metabolomics is sometimes also called metabonomics or metabolic profiling (Lindon *et al.*, 2011). Various experimental, statistical, and computational approaches have been discovered so far to perform the metabolome analysis. The most commonly used experimental techniques in metabolomics are mass spectrometry (MS), i.e., gas chromatography (GC-MS) or liquid chromatography (LC-MS), and nuclear magnetic resonance spectroscopy (NMR) spectra (Zhang *et al.*, 2012). Along with this statistical approaches have been devised to perform the analysis and it comprises of various standard analysis methods such as t-test and ANOVA, as well as more sophisticated methodologies such as univariate and multivariate analysis methods (Bartel *et al.*, 2013).

Also, bioinformatic approaches fall into the scenario due to the boom in high-throughput data and this includes databases for the data retrieval as well as web servers for carrying out the analysis (Maudsley *et al.*, 2011). Tremendous efforts have been implemented to archive all types of biological data, i.e., genes, proteins, gene products, metabolites, etc. With the advent of the genomic era, the amount of biochemical knowledge has exploded in the last two decades, which has necessitated its storage in large databases. Variety of top-down (gene to protein to metabolite) and bottom-up (chemical entity to biological function) approaches have been implied resulting in a rich expanse of metabolic knowledge from the biochemical network (Cakir and Khatibipour, 2014). Bioinformatics is a pioneer in preserving the data obtained through the experimental or natural resources and also provided the numerous tools for the analysis purpose. The databases provide the contextual biochemical basis for metabolomics data interpretation. By providing information regarding metabolites like defining which enzymatic reactions consume or produce them, and which pathways they're involved in, researchers can use this data to interpret their experiments towards a higher level of annotation.

METLIN, was the first metabolomics database, was established in 2004 (Smith *et al.*, 2005). Thereby in 2005, the Human Metabolome Project was launched to find and catalog all of the metabolites in human tissue and biofluids. This metabolite information is kept in the Human Metabolome Database, which produced its first draft in 2007 (Wishart *et al.*, 2007). There are many databases containing metabolomics data, and each has different information, ranging from NMR and MS spectra to metabolic pathways. Additionally, and more specifically to the development of metabolomics, mass spectral databases like the Golm Metabolome Database (Hummel *et al.*, 2007), which links mass spectrum and chromatographic retention time to specific compounds, have been developed. Some tools designed for higher level metabolomic analysis can take GC-MS spectra as input and use them for the identification stages of metabolomics. The purpose of metabolic databases is to organize the metabolites in a way that helps researchers in an easy identification and analysis. The information found in metabolite databases has continuously been updated to provide state-of-the-art data to the scientific community. Metabolomics is a new field and therefore new approaches are still being discovered and the existing ones are still improving. These databases are embraced with various types of information including concentration, anatomical location, and related disorders.

Other databases are MassBank (Horai *et al.*, 2008), lipid metabolites and pathways strategy (LIPID-MAPS) (LIPID), Madison metabolomics consortium database, and Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000). The major database till now is the KEGG, which is divided into several subdatabases with LIGAND, REACTION PAIR and PATHWAY being the most relevant to metabolomics (Booth *et al.*, 2013). In KEGG however there is a dauntingly large sum, i.e., 10,664 reactions and 18,107 metabolites (Kanehisa and Goto, 2000). These databases have been undergoing continuous updation and annotation for and so contain a great deal of valuable information. KEGG and MetaCyc (Karp *et al.*, 2002) are currently the largest (most number of organisms) and most in-depth comprehensive databases available. There are other databases too as Reactome (Joshi-Tope *et al.*, 2005), Model SEED (Devoid *et al.*, 2013), and BiGG (Schellenberger *et al.*, 2010), that can be more useful than the large databases if a specific organism is desired. The KEGG and MetaCyc databases each contain a generalized conserved set of pathways based on metabolic pathways. For KEGG, organism-specific annotations are available to query while for MetaCyc, individual "Cyc" databases have been generated for a number of organisms, some are just computationally derived while others are extensively manually curated such as AraCyc for *Arabidopsis* (Mueller *et al.*, 2003) and EcoCyc for *E.coli* strain K-12 MG1655 (Karp *et al.*, 2014). A more recent development is the cheminformatic databases like PubChem (Wang *et al.*, 2009) and ChEBI (Degtyarenko *et al.*, 2007), which provide a chemically ontological approach to catalog small molecules that are active in biological systems. These databases, therefore, can provide fruitful information regarding the metabolic datasets. Finally, it is important to note that these databases can be cross-referenced and linked to each other as well as against more widely known databases such as the well-known Chemical Abstract Service (CAS) among many others. Along with the above mentioned

Table 1 Selected databases and tools w.r.t metabolomics

<i>Databases</i>		
BioCyc	Collection of 10992 Pathway/Genome Databases (PGDBs)	http://biocyc.org
BiGG Models	Knowledgebase of genome-scale metabolic network reconstructions.	http://bigg.ucsd.edu/
BMRB	Repository for data from nuclear magnetic resonance spectroscopy (NMR) spectroscopy on proteins, peptides, nucleic acids, and other biomolecules	http://www.bmrw.wisc.edu/
BRENDA	Comprehensive enzyme repository, provide metabolic pathway details	http://www.brenda-enzymes.info/index.php
CHEBI	Freely available dictionary of molecular entities focused on “small” chemical compounds	https://www.ebi.ac.uk/chebi/
DIMEdB	Database of biologically relevant metabolite structures and annotations	http://dimedb.ifers.aber.ac.uk/
DRUGBANK	Unique bioinformatics and cheminformatics resource that combines detailed drug (i.e., chemical, pharmacological, and pharmaceutical) data with comprehensive drug target (i.e., sequence, structure, and pathway) information	https://www.drugbank.ca/
ECMDB	Contains extensive metabolomic data and metabolic pathway diagrams about <i>Escherichia coli</i> (strain K12, MG1655)	http://ecmdb.ca/
GMD	Facilitates the search for and dissemination of reference mass spectra from biologically active metabolites quantified using gas chromatography coupled to mass spectrometry (MS)	http://gmd.mpimp-golm.mpg.de/
HMDB	Archive information about small-molecule metabolites found in the human body	http://www.hmdb.ca/
KEGG	Resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information	http://www.genome.jp/kegg/pathway.html
MANET	Maps evolutionary relationships of molecule (metabolic, protein) architectures directly onto biological networks	http://manet.illinois.edu/
MassBank	Contains high resolution mass spectral data	http://massbank.eu/MassBank/
MetaboLights	Database for metabolomics experiments and derived information	http://www.ebi.ac.uk/metabolights/
MetaNetX	Automated model construction and genome annotation for large-scale metabolic networks	http://www.metanetx.org/
Reactome	Navigable map of human biological pathways, ranging from metabolic processes to hormonal signaling	http://www.reactome.org
SMPDB	Support pathway elucidation and pathway discovery in metabolomics, transcriptomics, proteomics and systems biology	http://smpdb.ca/
YMDB	Manually curated database of small-molecule metabolites found in or produced by <i>Saccharomyces cerevisiae</i>	http://www.ymdb.ca/
<i>Computational analysis tools</i>		
Arcadia	Visualization tool for metabolic pathways	http://arcadiapathways.sourceforge.net/
CellNetAnalyzer	MATLAB toolbox providing a various (partially unique) computational methods and algorithms for exploring structural and functional properties of metabolic, signaling, and regulatory networks	http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html
GLAMM	Unified web interface for visualizing metabolic networks, reconstructing metabolic networks from annotated genome data, visualizing experimental data in the context of metabolic networks, and investigating the construction of novel, transgenic pathways	http://glamm.lbl.gov/
IMPALA	Perform pathway overrepresentation and enrichment analysis with expression and/or metabolite data	http://impala.molgen.mpg.de
iPath	Web-based tool for the visualization, analysis and customization of the various pathways maps	http://pathways.embl.de
JDesigner	Graphical modeling environment for biochemical reaction networks	http://jdesigner.sourceforge.net/Site/JDesigner.html
KaPPA-View	Web-based analysis tool for integration of transcript and metabolite data on plant metabolic pathway	http://kpv.kazusa.or.jp/en/
LIPID MAPS	Online tools for lipid research	http://www.lipidmaps.org/tools/index.html
MapMan	A user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes	http://mapman.gabipd.org/web/guest/mapman
MetaMapp	Mapping and visualizing metabolomic data by integrating information from biochemical pathways and chemical and mass spectral similarity	http://uranus.fiehnlab.ucdavis.edu:8080/MetaMapp/homePage
MetPA	A web-based metabolomics tool for pathway analysis and visualization	http://metpa.metabolomics.ca
MassTRIX	Annotate metabolites in high precision MS data	http://masstrix3.helmholtz-muenchen.de/masstrix3/
MetaboAnalyst	Web server designed to permit comprehensive metabolomic data analysis, visualization and interpretation	http://www.metaboanalyst.ca/MetaboAnalyst/
MetaPath Online	For the analysis of metabolic networks	https://scopes.biologie.hu-berlin.de/

(Continued)

Table 1 Continued

<i>Databases</i>		
Meta P-server	A web based, easy-to-use analysis tool for the statistical analysis of metabolomics data	http://metabolomics.helmholtz-muenchen.de/metap2/
MetExplore	Find information about metabolite relationships in metabolic networks	http://metexplore.toulouse.inra.fr/metexplore/
Metscape	A plug-in for Cytoscape, used to visualize and interpret metabolomic data in the context of human metabolic networks	http://metscape.ncibi.org
MGV	A versatile generic graph viewer for multiomics data also it offers a comprehensive set of tools for analysis and visualization of graphs.	http://www.microarray-analysis.org/mayday
MPEA	Metabolite pathway enrichment analysis	http://ekhidna.biocenter.helsinki.fi/poxo/mpea/
MSEA	A web-based tool to identify biologically meaningful patterns in quantitative metabolomic data	http://www.msea.ca
Omix	Network drawing tool along with programmable visualization framework	http://www.omix-visualization.com/?from=http://www.13cflux.net/#sthash.vLyVKteK.dpbs
Paintomics	A web based tool for the joint visualization of transcriptomics and metabolomics data	http://www.paintomics.org
TICL	A web tool for network-based interpretation of compound lists inferred by high-throughput metabolomics	http://mips.helmholtz-muenchen.de/proj/cmp/home.html
Vanted	Network visualization and analysis tool for creating and editing the network and mapping experimental data onto networks	https://immersive-analytics.infotech.monash.edu/vanted/

Table 2 PAM50 genes list

<i>PAM50 genes</i>				
ACTR3B	CDCA1 (NUF2)	FOXA1	MDM2	PGR
ANLN	CDH3	FOXC1	MELK	PHGDH
BAG1	CENPF	GPR160	MIA	PTTG1
BCL2	CEP55	GRB7	MKI67	RRM2
BIRC5	CXXC5	KIF2C	MLPH	SFRP1
BLVRA	EGFR	KNTC2(NDC80)	MMP11	SLC39A6
CCNB1	ERBB2	KRT14	MYBL2	TMEM45B
CCNE1	ESR1	KRT17	MYC	TYMS
CDC20	EXO1	KRT5	NAT1	UBE2C
CDC6	FGFR4	MAPT	ORC6L(ORC6)	UBE2T

databases there are different types of databases built so far only the selected ones based upon accuracy and applications are mentioned in alphabetical order in [Table 1](#).

Similarly, there are many computational tools available for the metabolic data handling like for building, editing, enrichment, and interpreting metabolic network models, including Arcadia ([Villegier et al., 2010](#)), GLAMM ([Bates et al., 2011](#)), CellNetAnalyzer ([Klamt and von Kamp, 2011](#)), MetaMapp ([Barupal et al., 2012](#)), MetPA ([Xia and Wishart, 2010a](#)), Vanted ([Rohn et al., 2012](#)) for network visualization. Likewise there are tools for drawing a network, Omix ([Droste et al., 2013](#)) which have been widely used for creating a network model. TICL ([Antonov et al., 2009](#)) is used for the interpretation of compound lists that are inferred by high-throughput metabolomics. Enrichment analysis could be done with the help of MSEA ([Xia and Wishart, 2010b](#)), MPEA ([Kankainen et al., 2011](#)). MassTRIX ([Suhre and Schmitt-Kopplin, 2008](#)) performs the annotating metabolites in high precision MS data. There are other tools also, which have been mentioned in [Table 2](#) along with their descriptions.

Biomarker Discovery: A Challenge and Plausible Solutions Through Bioinformatics

Biomarkers are measured indicators of biological and pathogenic conditions or pharmacological responses to a therapeutic intervention ([Strimbu and Tavel, 2010](#)). Based on pathophysiological, epidemiological, therapeutic, or other scientific evidence they are intended to be useful in terms of clinical significance (i.e., to know whether they will benefit or harm) ([Baumgartner et al., 2011](#); [Downing, 2001](#)). Biomarkers have a generous impact on the care of patients; for those who are suspected to have the disease or those who have or have no visible disease symptoms ([Baumgartner et al., 2011](#)). For a long time biomarkers have served as a plausible diagnostic key to unraveling disease conditions, especially in case of cancers. Depending on the condition type they can be categorized as diagnostic, prognostic, and screening biomarkers ([Madu and Lu, 2010](#)). Currently, screening biomarkers are of high interest due to their ability to predict future events, but there are only a few accepted biomarkers for disease screening available today ([Melander et al., 2009](#)). Therefore it is necessary to have considerable search, verification, biological and

biochemical interpretation, and independent validation of disease biomarkers, which requires advancement in high-throughput technologies. To achieve this goal there is necessity to have interdisciplinary expertise, which requires the teamwork of clinicians, biologists, biochemists, and bioinformaticians to carry out biomarker cohort studies with professional planning, implementation, and control. Bioinformatics plays a key role in the biomarker discovery process via bridging the gap between initial discovery phases such as experimental design, clinical study execution, and bioanalytics, including sample preparation, separation, high-throughput profiling, and independent validation of identified candidate biomarkers (Baumgartner *et al.*, 2011) (Fig. 4).

It is well known that a disease or a phenotype is rarely a consequence of an abnormality of a single gene or its expression but instead reflects the interactions of various processes in a complex network; such network could combine multiple genes (proteins) and metabolites. For example, plants can produce numerous metabolites to handle different environmental conditions however the biosynthetic pathways for most of these compounds have not yet been revealed (Schlapfer *et al.*, 2017). From these facts, the need for a disease signature, a set of compounds generally presented as a network, becomes evident. Such disease-specific signature could be helpful in understanding all its mechanisms and evolution and in an earlier diagnosis. Multiple works and frameworks aimed to either use genes expression or metabolic data to extract a set of disease-relevant compounds; it's safe to say biomarkers (Strimbu and Tavel, 2010). Identifying such crucial biomarkers responsible for disease characteristics and revealing its mechanisms can be used to infer its evolution and development and offers better targets for drug development, treatment individualization, and dose regimen. These biomarkers are selected by analytic methods or pathway and network-centric methods (Wang *et al.*, 2015). A typical case would be gene expression data of two pairs of samples in both disease and normal states help in discovering genes and metabolites which can be potential biomarkers (Li *et al.*, 2013; Shlomi, 2010; Li *et al.*, 2012). Cancer is a heterogeneous disease, for instance, breast cancer. Biomarkers at the DNA, RNA, and protein levels were developed to better understand the biology of breast cancer, leading to the possibility to classify the disease into subtypes and subgroups, which may lead to new therapeutic opportunities (Le Du *et al.*, 2013). Many tests are available for the diagnosis and each one is based on a set of genes; we can list some of most known ones such as PAM50. PAM50 stands for Prediction Analysis of Microarray 50 (Sweeney *et al.*, 2014), fifty genes were probed like *ACTR3B*, *ANLN*, *BAG1*, *BCL2*, *BIRC5*, etc. Elaborating a set of biomarkers allowed the development of many tools online (cbioportal (Gao *et al.*, 2013)) and offline, in libraries and packages. These tools offer the possibility of analyzing targeted datasets and understanding the disease stage. Machine learning tools are also a very effective way of dealing with such complex diseases, predicting treatment effectiveness and new treated disease trajectory. Since these, new models have been well designed, which fully explains the relationship between each biomarker.

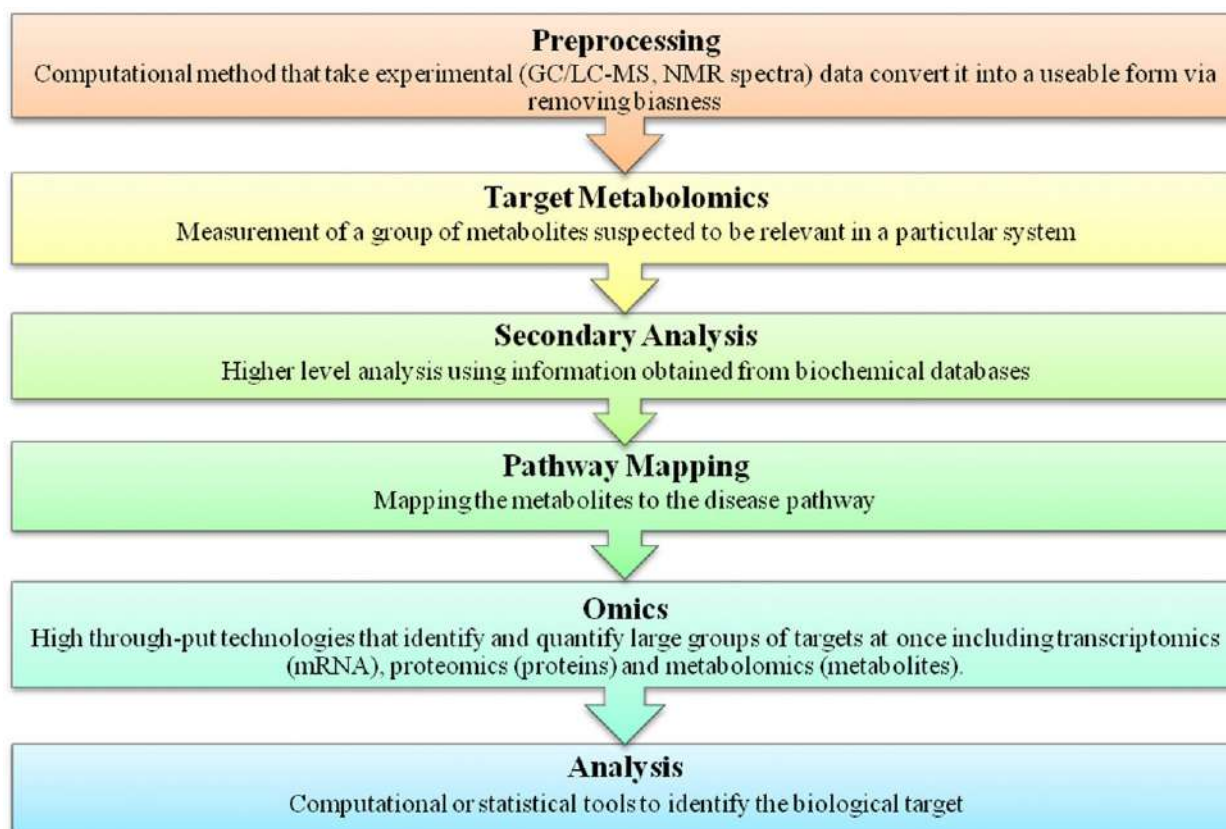


Fig. 4 Computational pipeline for metabolome analysis.

Gene Expression and Metabolic Network: Applications in Biomarker Discovery for Complex Diseases

DNA microarray technologies permit systematic approaches to the biological discovery that have a profound impact on biological research, pharmacology, and medicine (Yousef *et al.*, 2014). The ability to obtain quantitative information from the complete transcription profile of cells provides a powerful means to explore basic biology, disease diagnosis, drug development, mold therapeutics to specific pathologies, and generate databases (Young, 2000). Gene expression studies bridge the gap between DNA information and its trait information by dissecting the biochemical pathways into intermediate components, that is, between genotype and phenotype (Xiong *et al.*, 2001). The gene expression studies, therefore, open new avenues for identifying complex disease genes and biomarkers for disease diagnosis and also for assessing drug efficacy and toxicity. One particularly powerful application of gene expression analyses is in biomarker discovery, which can be used for disease risk assessment, early detection, prognosis, predicting response to therapy, and preventive measures.

For years, scientists studied one gene at a time and genes were indeed studied in isolation from the larger context of other involved genes. Nowadays, genomics via high-throughput techniques helps to study the genome of organisms as a whole thus allowing a wide picture of gene characteristics. One of the most popular high-throughput techniques are arrays, which are an orderly arrangement of a large number of samples allowing large-scale studies (Yousef *et al.*, 2014). This gave rise to the genomic era, which emerged from the sequencing of genomes from many organisms. The development of the first arrays started many years ago to study a large number of genes at a time (Hergenhahn *et al.*, 2003) and has widely expanded since then. Today the approach is also applicable to RNA probes, proteins, antibodies, and even biological samples allowing new types of research (Yousef *et al.*, 2014). Currently, other types of high-throughput techniques are also developing, for instance, to study the transcripts and metabolites.

Today, genomics has induced two new paradigms in biology; the first paradigm is a new approach that allows the study of the complex network through which genes and proteins communicate. It is attained via an amalgamation of the researcher having expertise in the field of biology, engineering, chemistry, and computer science; this multidisciplinary approach allows the development of systems biology. The second paradigm is a direct consequence of information derived from genomics studies where raw data needs to be analyzed and then to be used in the systemic approach. This led the development of bioinformatics,

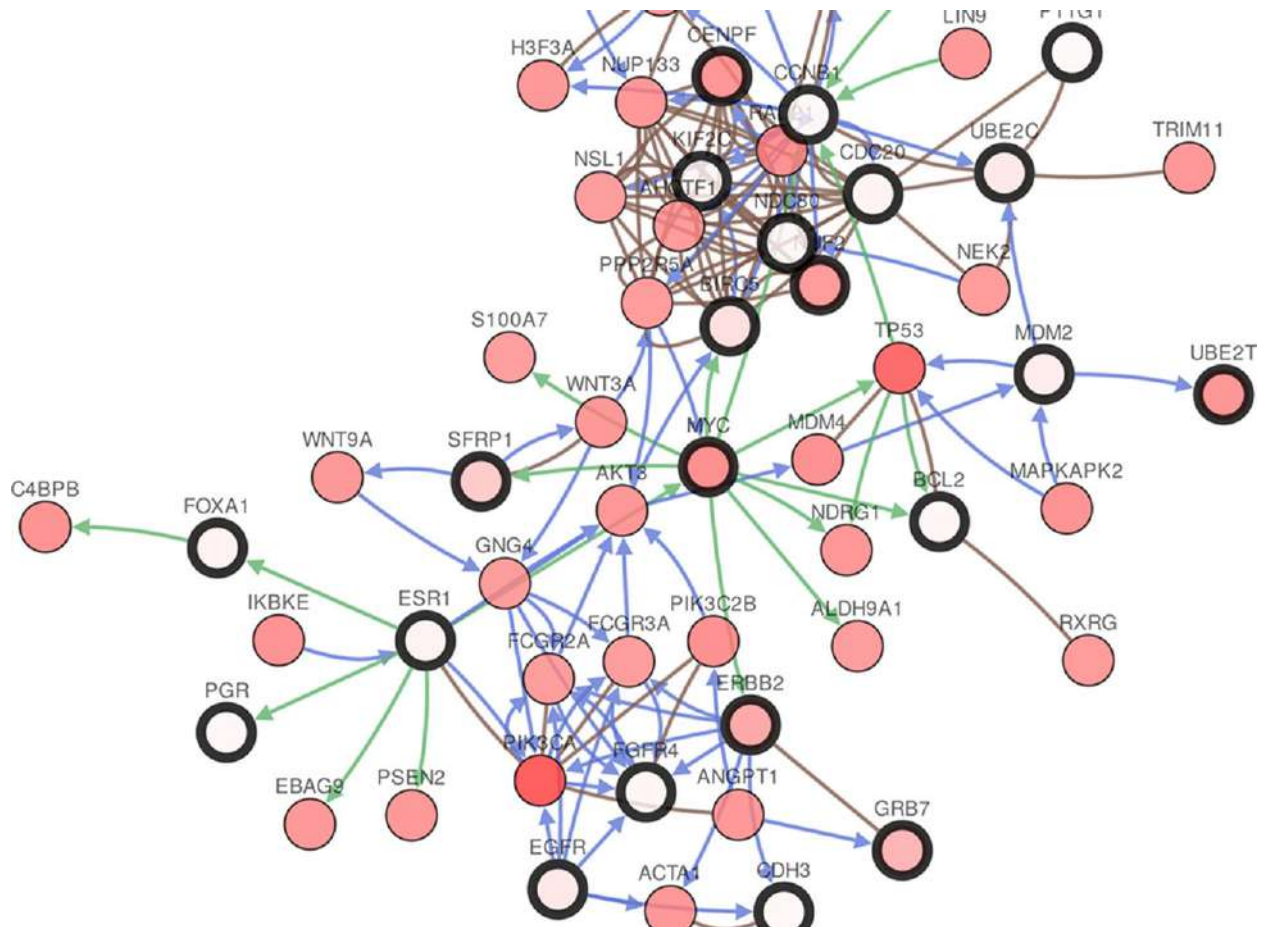


Fig. 5 Network interaction of the PAM50 genes set.

which requires the use of computers to manage biological information. The practical applications of gene expression analyses are numerous and only beginning to be realized. One particularly powerful application of gene expression analyses is in biomarker identification, which can be used for disease risk assessment, early detection, prognosis, prediction response to therapy, and preventative measures.

Therefore approaches to cancer biomarker discovery comprise genomics, epigenomics, transcriptomics, and proteomic analyses. Current efforts in the laboratory focus on the identification of biomarkers in chronic lymphocytic leukemia, lung cancer, and colon cancer (McDermott *et al.*, 2013). Along with the mRNA other small RNAs are also known to be a predictive indicator in disease studies; and it has been found that alterations in gene expression patterns due to dysregulation of miRNAs is a common cause in tumorigenesis (Chen *et al.*, 2012). High concentrations of cell-free miRNAs that originate from the primary tumor have been found in the plasma of cancer patients, and several lines of evidence indicate that plasma miRNAs are associated with specific vesicles called exosomes (Yang *et al.*, 2016). This led to the discovery of new biomarkers that comprises plasma miRNAs and seems to be promising in disease prognosis (Jeffrey, 2008). Recent discovery of quantifiable circulating cancer-associated miRNAs exposes the immense potential of their use as novel minimally invasive biomarkers for breast and other cancers (Heneghan *et al.*, 2009).

Discovery of Biomarkers Through Computational Pipeline: A Cancer Based Study

The classification of samples from gene expression datasets usually involves a small number of samples. The problem of selecting those biomarker genes that are vital for differentiating the different sample classes being compared poses a challenging problem in

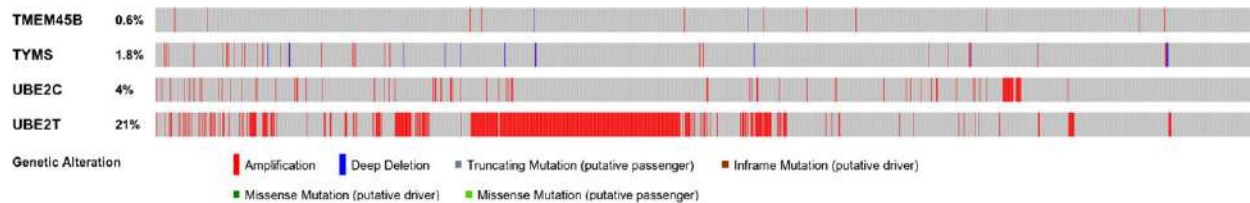


Fig. 6 Oncoprint for only four biomarkers.

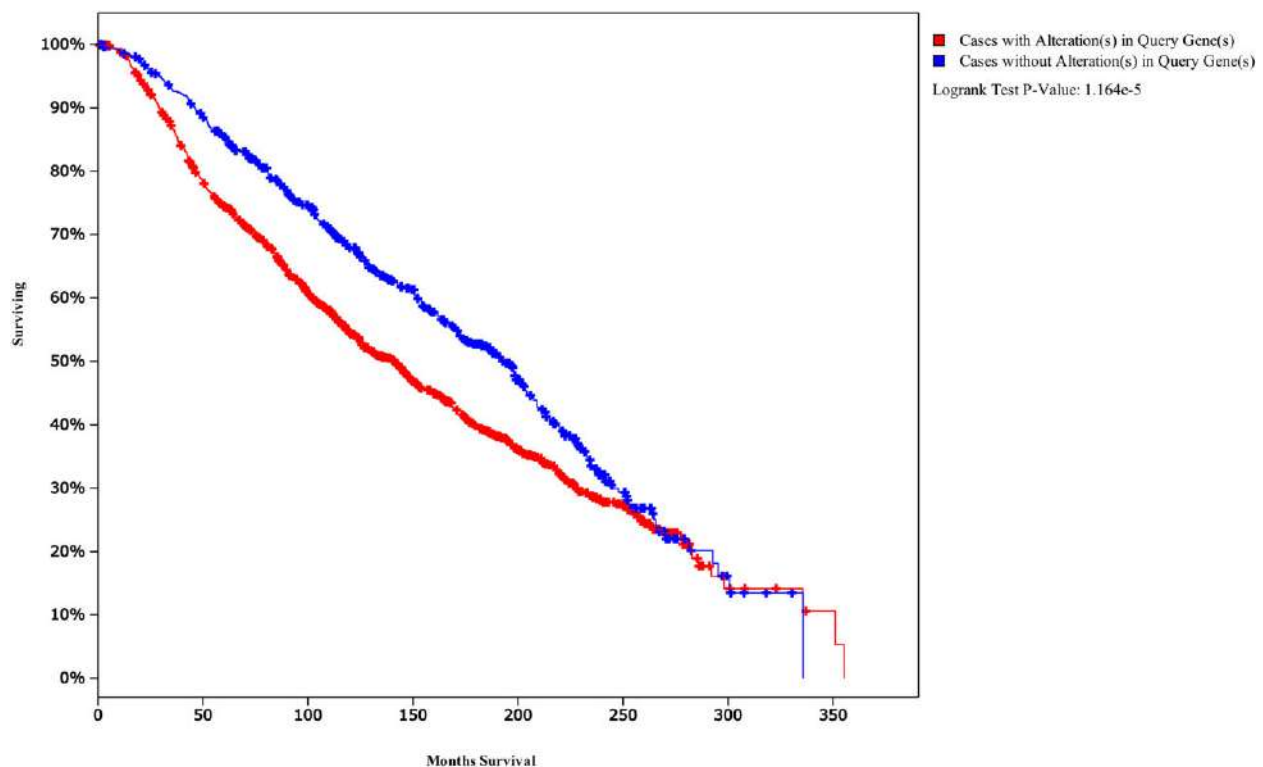


Fig. 7 Survival study.

case of the high dimensional data analysis. A variety of methods to address this problem have been implemented and these methods can be divided into two main categories: (1) Filtering based methods and (2) model-based or wrapper approaches (Wang *et al.*, 2005; Inza *et al.*, 2004). The filter method (Ben-Bassat, 1982) assesses the goodness of the proposed feature subset looking only at the intrinsic characteristics of the data, based on the relation of each single gene with the class label by the calculation of simple statistics computed from the empirical distribution. This approach is extensively used as a feature subset selection method in the microarray (Aris and Recce, 2002). While in the wrapper approach (Kohavi and John, 1997), which is a very powerful machine learning application, search is conducted in the space of genes, evaluating the goodness of each found gene subset by the estimation of the accuracy percentage of the specific classifier to be used.

One of the most important properties that should be considered for the biomarkers is the robustness. It is an important issue for the successful discovery of biomarkers, as it may greatly influence subsequent biological validations. In addition, a more robust set of markers may strengthen the confidence of an expert in the results of a selection method (Abeel *et al.*, 2010). Robustness, a property that allows a system to maintain its functions under certain external and internal perturbations, is a ubiquitously observed feature of biological systems (Kitano, 2004). Studying the relationship between the topology and robustness of

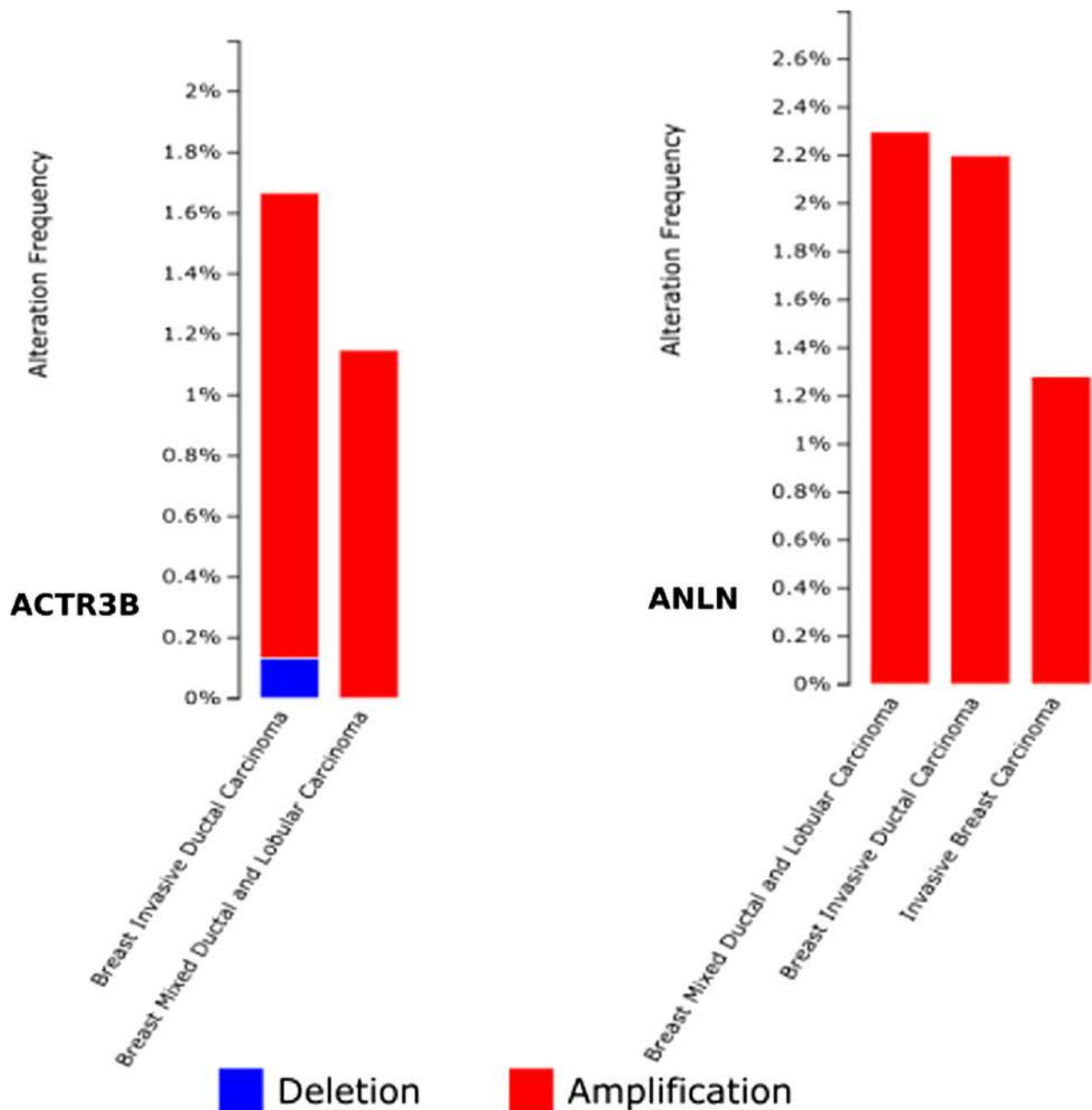


Fig. 8 Breast cancer histogram with subtypes.

metabolic networks may help to understand the functional organization principle of cells and could have important implications for disease studies (Schuster and Holzhütter, 1995) and drug target identifications (Bakker *et al.*, 2000).

The robustness of the metabolic network with respect to specific enzymes can be qualitatively estimated by the preservation or decay of the network after removal of the nodes or edges corresponding to these enzymes. The following topological features of metabolic networks may ensure the robustness of metabolism; these are (1) modularity, which contributes to the robustness of networks by decreasing the cross talk between different functional modules, detaining perturbations and damages to separable parts and preventing deleterious effects from spreading to the whole system (Stelling *et al.*, 2004); (2) bow-tie structure, which aids in forming a robust conserved core because it is the most tightly connected part of the network and there are multiple routes between any pair of nodes. Such connecting patterns provide an advantage in generating a coordinated response to various stimuli and increases the robustness of the whole system (Kitano, 2004); and (3) scale-free topology; the key feature of scale-free networks is the high-degree of error tolerance; that is, the ability of their nodes to communicate is unaffected by the failure of some randomly chosen nodes (Crucitti *et al.*, 2004). Thus the scale-free nature of metabolic networks indicates its high resistance towards random perturbations and thus could explain why some enzyme dysfunction at the metabolic level is without substantial phenotypic effect (Barabasi and Albert, 1999). The studies have shown that the scale-free networks are extremely vulnerable to attacks, i.e., the removal of a few hub nodes that play a crucial role in maintaining a network's connectivity will destroy the whole network (Crucitti *et al.*, 2004). Studies conducted by Mahadevan and Palsson showed that low-degree nodes are almost as likely to be critical to the overall network functions as high-degree nodes (Mahadevan and Palsson, 2005) by calculating the number of lethal reactions among all the reactions connected to every metabolite in the substrate graph of metabolic networks.

To show an example of analysis using biomarkers a dataset from (Pereira *et al.*, 2016) was used. It presents a somatic mutation profiling study of 2433 breast cancers, which approved the classification of the tumors into 10 integrative clusters (IntClusters).

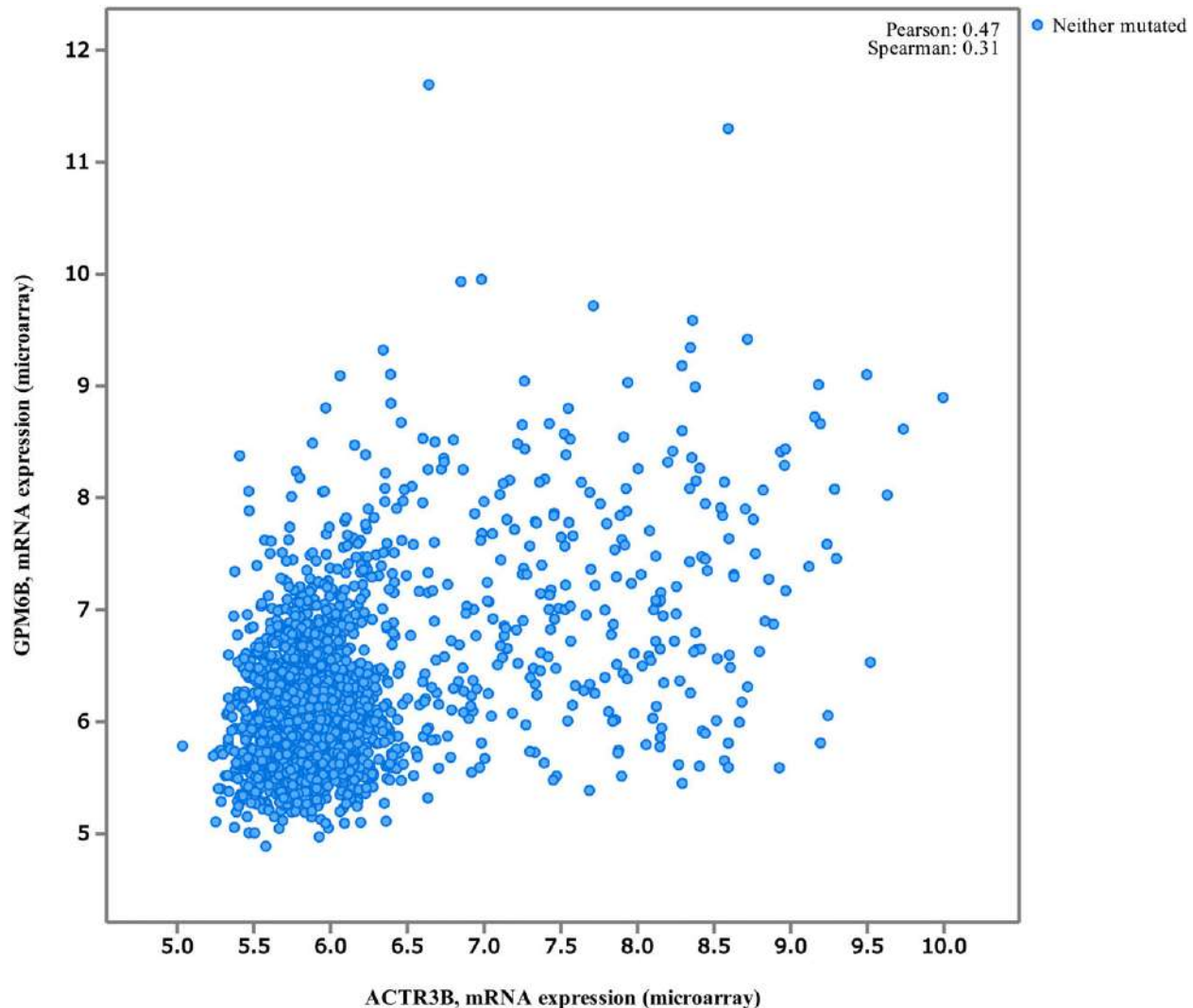


Fig. 9 Coexpression of two biomarkers, ACTR3B & GPM6B.

During this study (Pereira *et al.*, 2016) CNAs (copy number aberrations) were the main driver in an unsupervised clustering approach due to their influence on genes expression. The current analysis used the dataset above against the fifty genes found in PAM50 (Table 2). These genes served as a gene set for the cBioportal, so the mutations profiling could be performed. Figures below show different knowledges extracted, cancer types, and mutation happening along genes, genes with similar or exclusive expression, and finally survival estimation. Gene networks present an excellent input for models analysis using algorithmic methods such as Bayesian networks, leading to inferring and predicting the cancer evolution, trajectory, and progression (Caravagna *et al.*, 2016). Fig. 5 shows the interaction of the PAM50 gene set where nodes are representing genes and edges (arrows) denote the interactions between them.

Thereby an oncoprint of four biomarkers (TMEM45B, TYMS, UBE2C, UBE2T) has been captured (Fig. 6) that shows the type of mutation occurring to the gene set.

After this, a survival study has been conducted (Fig. 7), which is performed for the p-value of 1.164e-5 and red color denotes the alteration in query gene while blue color shows the query gene without alteration. It has been noticed that the cases with alterations have higher survival rate than the nonaltered ones.

Then alteration frequency of ACTR3B and ANLN were determined through the histogram (Fig. 8) that denotes the alteration and the deletion frequency. Finally Pearson and Spearmen correlation coefficient induced shows the coexpression of the two biomarkers, that is, ACTR3B & GPM6B (Fig. 9).

Concluding Remarks

The advent of high-throughput technologies, and as a result, the generation of various kinds of omics data, has challenged both the experimental or computational scientific communities. Therefore an “omics cascade” came into the scenario where the genotype to the phenotype can be connected through various intermediate steps for better understanding the disease biology and their overall impact on the phenotypic observations at the physical or organism level. Understanding the genotype to phenotype scenario and its consequences will provide an edge towards better implementation of computational methods for their experimental validations. It will be an added advantage for the scientific community to amalgamate the computational and experimental procedures for the betterment of mankind. It is anticipated that this comprehensive text will provide systematic and ordered information on the metabolic world with its exact connection to the biomarker identification procedure.

See also: Biological Database Searching. Identification and Extraction of Biomarker Information. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics

References

- Abeel, T., Helleputte, T., Van De Peer, Y., Dupont, P., Saeys, Y., 2010. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics* 26, 392–398.
- Antonov, A.V., Dietmann, S., Wong, P., Mewes, H.W., 2009. TICL – A web tool for network-based interpretation of compound lists inferred by high-throughput metabolomics. *The FEBS Journal* 276, 2084–2094.
- Aris, V., Recce, M., 2002. A method to improve detection of disease using selectively expressed genes in microarray data. In: *Proceedings of CAMDA'00*, pp. 69–81. (S.M. Lin and K.F. Johnson, editors).
- Bakker, B.M., Mensonides, F.I., Teusink, B., *et al.*, 2000. Compartmentation protects trypanosomes from the dangerous design of glycolysis. *Proceedings of the National Academy of Sciences* 97, 2087–2092.
- Barabasi, A.L., Albert, R., 1999. Emergence of scaling in random networks. *Science* 286, 509–512.
- Barabasi, A.L., Oltvai, Z.N., 2004. Network biology: Understanding the cell's functional organization. *Nature Review Genetics* 5, 101–113.
- Bartel, J., Krumsiek, J., Theis, F.J., 2013. Statistical methods for the analysis of high-throughput metabolomics data. *Computational and Structural Biotechnology Journal* 4, e201301009.
- Barupal, D.K., Haldiya, P.K., Wohlgemuth, G., *et al.*, 2012. MetaMapp: Mapping and visualizing metabolomic data by integrating information from biochemical pathways and chemical and mass spectral similarity. *BMC Bioinformatics* 13, 99.
- Bates, J.T., Chivian, D., Arkin, A.P., 2011. GLAMM: Genome-Linked Application for Metabolic Maps. *Nucleic Acids Research* 39, W400–W405.
- Baumgartner, C., Osl, M., Netzer, M., Baumgartner, D., 2011. Bioinformatic-driven search for metabolic biomarkers in disease. *Journal of Clinical Bioinformatics* 1, 2.
- Beber, M.E., Fretter, C., Jain, S., *et al.*, 2012. Artefacts in statistical analyses of network motifs: General framework and application to metabolic networks. *Journal of the Royal Society Interface* 9, 3426–3435.
- Ben-Bassat, M., 1982. Pattern recognition and reduction of dimensionality. *Handbook of Statistics* 2, 773–910.
- Berkhout, J., Teusink, B., Bruggeman, F.J., 2013. Gene network requirements for regulation of metabolic gene expression to a desired state. *Scientific Reports* 3, 1417.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U., 2006. Complex networks: Structure and dynamics. *Physics Reports* 424, 175–308.
- Booth, S.C., Weljie, A.M., Turner, R.J., 2013. Computational tools for the secondary analysis of metabolomics experiments. *Computational and Structural Biotechnology Journal* 4, e201301003.
- Bourqui, R., Cottret, L., Lacroix, V., *et al.*, 2007. Metabolic network visualization eliminating node redundancy and preserving metabolic pathways. *BMC Systems Biology* 1, 29.
- Cakir, T., Khatibipour, M.J., 2014. Metabolic network discovery by top-down and bottom-up approaches and paths for reconciliation. *Frontiers in Bioengineering and Biotechnology* 2, 62.
- Caravagna, G., Graudenzi, A., Ramazzotti, D., *et al.*, 2016. Algorithmic methods to infer the evolutionary trajectories in cancer progression. *Proceedings of the National Academy of Sciences of the United States of America* 113, E4025–E4034.
- Chen, P.S., Su, J.L., Hung, M.C., 2012. Dysregulation of microRNAs in cancer. *Journal of Biomedical Science* 19, 90.

- Cho, D.Y., Kim, Y.A., Przytycka, T.M., 2012. Chapter 5: Network biology approach to complex diseases. *PLOS Computational Biology* 8, e1002820.
- Cooper, G.M., 2000. The central role of enzymes as biological catalysts. Sinauer Associates.
- Crucitti, P., Latora, V., Marchiori, M., Rapisarda, A., 2004. Error and attack tolerance of complex networks. *Physica A: Statistical Mechanics and its Applications* 340, 388–394.
- DeBerardinis, R.J., Thompson, C.B., 2012. Cellular metabolism and disease: What do metabolic outliers teach us? *Cell* 148, 1132–1144.
- Deglyarenko, K., De Matos, P., Ennis, M., *et al.*, 2007. ChEBI: A database and ontology for chemical entities of biological interest. *Nucleic Acids Research* 36, D344–D350.
- Devoid, S., Overbeek, R., Dejongh, M., *et al.*, 2013. Automated genome annotation and metabolic model reconstruction in the SEED and Model SEED. *Systems Metabolic Engineering: Methods and Protocols*. 17–45.
- Downing, G., 2001. Biomarkers definitions working group. Biomarkers and surrogate endpoints. *Clinical Pharmacology & Therapeutics* 69, 89–95.
- Droste, P., Nöh, K., Wiechert, W., 2013. Omix – A visualization tool for metabolic networks with highest usability and customizability in focus. *Chemie Ingenieur Technik* 85, 849–862.
- Gao, J., Aksoy, B.A., Dogrusoz, U., *et al.*, 2013. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Science Signaling* 6, p11.
- Heneghan, H., Miller, N., Lowery, A., Sweeney, K., Kerin, M., 2009. MicroRNAs as novel biomarkers for breast cancer. *Journal of Oncology* 2009.
- Hergenrohn, M., Muhlemann, K., Hollstein, M., Kenzelmann, M., 2003. DNA microarrays: Perspectives for hypothesis-driven transcriptome research and for clinical applications. *Current Genomics* 4, 543–555.
- Horai, H., Aranita, M., Nishioka, T., 2008. MassBank: Mass spectral database for metabolome analysis. In: *Proceedings of the 56th ASMS Conference on Mass Spectrometry and Allied Topics*, Denver, CO.
- Hucka, M., Finney, A., Sauro, H.M., *et al.*, 2003. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.
- Hummel, J., Selbig, J., Walther, D., Kopka, J., 2007. The Golm Metabolome Database: A database for GC–MS based metabolite profiling. *Metabolomics*. 75–95.
- Inza, I., Larranaga, P., Blanco, R., Cerrolaza, A.J., 2004. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine* 31, 91–103.
- Jeffrey, S.S., 2008. Cancer biomarker profiling with microRNAs. *Nature Biotechnology* 26, 400–401.
- Jeong, H., Mason, S.P., Barabasi, A.L., Oltvai, Z.N., 2001. Lethality and centrality in protein networks. *Nature* 411, 41–42.
- Joshi-Tope, G., Gillespie, M., Vastrik, I., *et al.*, 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research* 33, D428–D432.
- Kanehisa, M., Goto, S., 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* 28, 27–30.
- Kankainen, M., Gopalacharyulu, P., Holm, L., Orešič, M., 2011. MPEA – Metabolite pathway enrichment analysis. *Bioinformatics* 27, 1878–1879.
- Karp, P.D., Riley, M., Paley, S.M., Pellegrini-Toole, A., 2002. The metacyc database. *Nucleic Acids Research* 30, 59–61.
- Karp, P.D., Weaver, D., Paley, S., *et al.*, 2014. The EcoCyc database. *EcoSal Plus* 6.
- Kitano, H., 2004. Biological robustness. *Nature Reviews Genetics* 5, 826–837.
- Klamt, S., von Kamp, A., 2011. An application programming interface for CellNetAnalyzer. *Biosystems* 105, 162–168.
- Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. *Artificial intelligence* 97, 273–324.
- Le Du, F., Ueno, N.T., Gonzalez-Angulo, A.M., 2013. Breast Cancer Biomarkers: Utility in clinical practice. *Current Breast Cancer Reports* 5.
- Lee, D.-S., Park, J., Kay, K., *et al.*, 2008. The implications of human metabolic network topology for disease comorbidity. *Proceedings of the National Academy of Sciences* 105, 9880–9885.
- Li, L., Jiang, H., Ching, W.-K., Vassiliadis, V.S., 2012. Metabolite biomarker discovery for metabolic diseases by flux analysis. In: *Proceedings of the 2012 IEEE 6th International Conference on Systems Biology (ISB)*, pp. 1–5. IEEE.
- Li, L., Jiang, H., Qiu, Y., Ching, W.K., Vassiliadis, V.S., 2013. Discovery of metabolite biomarkers: Flux analysis and reaction-reaction network approach. *BMC Systems Biology* 7 (Suppl. 2), S13.
- Lindon, J.C., Nicholson, J.K., Holmes, E., 2011. *The Handbook of Metabonomics and Metabolomics*. Elsevier.
- Lipid Maps, LIPID Metabolites And Pathways Strategy, Wellcome Trust.
- Madu, C.O., Lu, Y., 2010. Novel diagnostic biomarkers for prostate cancer. *Journal of Cancer* 1, 150–177.
- Mahadevan, R., Palsson, B.O., 2005. Properties of metabolic networks: Structure versus function. *Biophysical Journal* 88, L07–L09.
- Maudsley, S., Chadwick, W., Wang, L., *et al.*, 2011. Bioinformatic approaches to metabolic pathways analysis. *Methods in Molecular Biology* 756, 99–130.
- McDermott, J.E., Wang, J., Mitchell, H., *et al.*, 2013. Challenges in biomarker discovery: Combining expert insights with statistical analysis of complex omics data. *Expert Opinion on Medical Diagnostics* 7, 37–51.
- Melander, O., Newton-Cheh, C., Almgren, P., *et al.*, 2009. Novel and conventional biomarkers for prediction of incident cardiovascular events in the community. *JAMA* 302, 49–57.
- Mohney, R., Milburn, M., 2015. Providing insight into complex disease: Metabolomics links genetic loci to phenotype. *The FASEB Journal* 29, LB292.
- Mueller, L.A., Zhang, P., Rhee, S.Y., 2003. AraCyc: A biochemical pathway database for Arabidopsis. *Plant Physiology* 132, 453–460.
- Oliver, S.G., Winson, M.K., Kell, D.B., Baganz, F., 1998. Systematic functional analysis of the yeast genome. *Trends in Biotechnology* 16, 373–378.
- Peleg, M., Rubin, D., Altman, R.B., 2005. Using Petri net tools to study properties and dynamics of biological systems. *Journal of the American Medical Informatics Association* 12, 181–199.
- Pereira, B., Chin, S.-F., Rueda, O.M., *et al.*, 2016. The somatic mutation profiles of 2,433 breast cancers refines their genomic and transcriptomic landscapes. *Nature Communications* 7.
- Rohn, H., Junker, A., Hartmann, A., *et al.*, 2012. VANTED v2: A framework for systems biology applications. *BMC Systems Biology* 6, 139.
- Schellenberger, J., Park, J.O., Conrad, T.M., Palsson, B.O., 2010. BiGG: A biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics* 11, 213.
- Schlapfer, P., Zhang, P., Wang, C., *et al.*, 2017. Genome-wide prediction of metabolic enzymes, pathways, and gene clusters in plants. *Plant Physiology* 173, 2041–2059.
- Schuster, R., Holzhütter, H.G., 1995. Use of mathematical models for predicting the metabolic effect of large-scale enzyme activity alterations. *The FEBS Journal* 229, 403–418.
- Sehgal, M., Gupta, R., Moussa, A., Singh, T.R., 2015. An integrative approach for mapping differentially expressed genes and network components using novel parameters to elucidate key regulatory genes in colorectal cancer. *PLOS One* 10, e0133901.
- Shahzad, K., Loor, J.J., 2012. Application of top-down and bottom-up systems approaches in ruminant physiology and metabolism. *Current Genomics* 13, 379–394.
- Shlomi, T., 2010. Metabolic network-based interpretation of gene expression data elucidates human cellular metabolism. *Biotechnology & Genetic Engineering Reviews* 26, 281–296.
- Shukla, A., Singh, T.R., 2017. *Computational network approaches and their applications for complex diseases*. Translational Bioinformatics and its Application. Springer.
- Smith, C.A., O'maille, G., Want, E.J., *et al.*, 2005. METLIN: A metabolite mass spectral database. *Therapeutic drug monitoring* 27, 747–751.
- Sridharan, G.V., Ullah, E., Hassoun, S., Lee, K., 2015. Discovery of substrate cycles in large scale metabolic networks using hierarchical modularity. *BMC Systems Biology* 9, 5.
- Stelling, J., Sauer, U., Szallasi, Z., Doyle, F.J., Doyle, J., 2004. Robustness of cellular functions. *Cell* 118, 675–685.
- Strimbu, K., Tavel, J.A., 2010. What are biomarkers? *Current Opinion in HIV and AIDS* 5, 463.
- Suhre, K., Schmitt-Kopplin, P., 2008. MassTRIX: Mass translator into pathways. *Nucleic Acids Research* 36, W481–W484.
- Sweeney, C., Bernard, P.S., Factor, R.E., *et al.*, 2014. Intrinsic subtypes from PAM50 gene expression assay in a population-based breast cancer cohort: Differences by age, race, and tumor characteristics. *Cancer Epidemiology, Biomarkers and Prevention* 23, 714–724.

- Vander Heiden, M.G., Cantley, L.C., Thompson, C.B., 2009. Understanding the Warburg effect: The metabolic requirements of cell proliferation. *Science* 324, 1029–1033.
- Veeramani, B., Bader, J.S., 2010. Predicting functional associations from metabolism using bi-partite network algorithms. *BMC Systems Biology* 4, 95.
- Villegier, A.C., Pettifer, S.R., Kell, D.B., 2010. Arcadia: A visualization tool for metabolic pathways. *Bioinformatics*, 26. . pp. 1470–1471.
- Wang, J., Zuo, Y., Man, Y.G., *et al.*, 2015. Pathway and network approaches for identification of cancer signature markers from omics data. *Journal of Cancer* 6, 54–65.
- Wang, Y., Tetko, I.V., Hall, M.A., *et al.*, 2005. Gene selection from microarray data for cancer classification – A machine learning approach. *Computational Biology and Chemistry* 29, 37–46.
- Wang, Y., Xiao, J., Suzek, T.O., *et al.*, 2009. PubChem: A public information system for analyzing bioactivities of small molecules. *Nucleic Acids Research* 37, W623–W633.
- Wishart, D.S., Tzur, D., Knox, C., *et al.*, 2007. HMDB: The human metabolome database. *Nucleic Acids Research* 35, D521–D526.
- Xia, J., Wishart, D.S., 2010a. MetPA: A web-based metabolomics tool for pathway analysis and visualization. *Bioinformatics* 26, 2342–2344.
- Xia, J., Wishart, D.S., 2010b. MSEA: A web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic Acids Research* 38, W71–W77.
- Xiong, M., Fang, X., Zhao, J., 2001. Biomarker identification by feature wrappers. *Genome Research* 11, 1878–1887.
- Xu, C., Hu, S., Chen, X., 2016. Artificial cells: From basic science to applications. *Material Today (Kidlington)* 19, 516–532.
- Yang, Q., Diamond, M.P., Al-Hendy, A., 2016. The emerging role of extracellular vesicle-derived miRNAs: Implication in cancer progression and stem cell related diseases. *Journal of Clinical Epigenetics* 2.
- Young, R.A., 2000. Biomedical discovery with DNA arrays. *Cell* 102, 9–15.
- Yousef, M., Najami, N., Abedallah, L., Khalifa, W., 2014. Computational approaches for biomarker discovery. *Journal of Intelligent Learning Systems and Applications* 6, 153.
- Zhang, A., Sun, H., Wang, P., Han, Y., Wang, X., 2012. Modern analytical techniques in metabolomics analysis. *Analyst* 137, 293–300.
- Zhu, X., Gerstein, M., Snyder, M., 2007. Getting connected: Analysis and principles of biological networks. *Genes & Development* 21, 1010–1024.

Investigating Metabolic Pathways and Networks

Md Altaf-Ul-Amin and Shigehiko Kanaya, Nara Institute of Science and Technology, Ikoma, Japan
Zeti-Azura Mohamed-Hussein, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Living cells generate energy and produce building material for cell components and replenish enzymes by the process of metabolism. All organisms live and grow by receiving food and nutrients from the environment. The foods are processed through thousands of reactions. In cells chemical reactions take place round the clock constantly breaking and making chemical molecules and transferring ions and electrons and these reactions are called metabolic pathways. Therefore, the metabolites can be considered as the preliminary level molecules generated from food intakes which are gradually transformed into building blocks for producing proteins, RNAs and DNAs and other useful matter and energy for creating and maintaining cells and life.

Metabolic reactions follow the laws of physics and chemistry and thus modeling metabolic reactions require considering many physicochemical constraints (Palsson, 2006). Metabolic pathways are efficiently regulated to respond to external perturbations and internal needs and thus linked to signaling networks. Many severe human diseases are caused by metabolic imbalance e.g., diabetes, cancer, cardiovascular problems, obesity, gout, tyrosinemia are a few to name.

Metabolism is the general term for the following two kinds of reactions: (i) Catabolic reactions: Catabolism refers to chemical reactions that result in the breakdown of more complex organic molecules into simpler substances. Catabolic reactions usually release energy that is used to drive chemical reactions. In catabolism, large molecules such as polysaccharides, lipids, nucleic acids and proteins are broken down into smaller units such as monosaccharides, fatty acids, nucleotides, and amino acids. The energy of catabolic reactions is used to drive anabolic reactions. (ii) Anabolic reactions: Anabolism refers to chemical reactions in which simpler substances are combined to form more complex molecules. Anabolic reactions usually require energy. Anabolic reactions build new molecules and/or store energy. The energy for chemical reactions is stored in ATP.

The term metabolic network usually means a collection of metabolic reactions represented as networks where the metabolites are the nodes and two metabolites are connected if one of them is a substrate and the other is the product of a reaction. Genome scale reconstruction of a metabolic network involves thousands of metabolites and reactions. Metabolic reactions are catalyzed by enzymes. The enzymes are gene products or proteins in broader sense. Therefore, metabolic networks contain information about both metabolites and proteins where the metabolites are nodes and proteins/enzymes are edges. There are other ways of representing metabolic pathways such as Bipartite graphs and Petri nets (Koch *et al.*, 2004; Matsuno *et al.*, 2011). A Petri net is an attributed directed bipartite graph. A metabolic pathway can be represented as a bipartite graph by considering the metabolites as one set of nodes and the enzymes as another set of nodes and such representation can provide some overall preliminary information about the system. Other than the metabolic pathways, we can also construct different type of metabolic networks e.g., based on structural similarity or any other type of similarity between metabolites.

Some studies showed that the metabolic networks are scale free networks i.e., their degree distribution follows power law (Jeong *et al.*, 2000; Ma and Zeng, 2003). Above all metabolic networks can be regarded as small world networks for their power-law degree distribution, high clustering coefficient and low average path length and diameter (Strogatz, 2001; Fell and Wagner, 2000). High clustering coefficients imply the existence of modules in the networks. It has been proposed that the combined properties of power-law degree distribution and high clustering coefficient indicates that the modules in the networks are linked to one another in a hierarchical manner (Albert *et al.*, 2000).

Numerous and versatile researches have been conducted based on pathways and chemical structures of metabolites. In this book article, we focus somewhat detail on two research topics. In the first part of this article we discuss alkaloid synthesis pathways and in the second part of this article we discuss structural similarity based networks of metabolites and its usage in establishing structure activity relations.

Insight Into Alkaloid Synthesis Pathways

Metabolites can be broadly classified into two types, primary metabolites and secondary metabolites. Primary metabolites are involved in normal growth, development, and reproduction of an organism and are usually present in almost all organisms. On the other hand, secondary metabolites are not generally essential for the very survival of an organism but mainly have important ecological functions. Usually, secondary metabolites are present in taxonomically restricted groups of organisms such as plants, fungi, bacteria etc. The term 'secondary' was introduced in 1891 (Kossel, 1891) in the context of metabolites. The functions of secondary metabolites are acquired in evolution process involving pest and pathogen defense, UV-B-sunscreens (Dixon, 2001) etc. Secondary metabolites with known chemical structures are more than 3000 terpenoids, 9000 flavonoids, 1600 isoflavonoids and 12,000 alkaloids (Connolly and Hill, 1991; Ziegler and Facchini, 2008). Since 2004, we have been accumulating the relations between metabolites and producing species in KNApSACk Core Database (DB, see "Relevant Websites section"), presently which

are 102,005 species-metabolite relationships encompassing 20,741 species and 50,054 metabolites. It has been predicted based on current statistics of metabolites-species relations, that there are at least 1.06 million metabolites within all plants on this planet of which most are secondary metabolites (Afendi *et al.*, 2011).

The classification of secondary metabolites based on integrated information of chemical structure and metabolic pathways could provide important clues to activities of metabolites which lead to interpretation of function acquisition mechanisms of secondary metabolites in evolutionary process. In this part of the article, we examine whether or not classification of alkaloids by ring skeletons can be related to metabolic pathways. Alkaloids are a large group of nitrogen-containing secondary metabolites. Almost every variety of organisms such as bacteria, fungi, plants, and animals produce alkaloids. Diverged polycyclic compounds including unsaturated and saturated bonds are produced by various organisms (Fig. 1 shows some examples).

Systematic representation of alkaloid biosynthetic pathways has been established by Aniszewski based on ring skeletons. The skeleton nucleus of an alkaloid is the main criterion for alkaloid precursor determination in biosynthetic pathways. Skeletons for different alkaloid nuclei can be produced by L-lysine, for example, piperidine, indolizine, quinolizidine nuclei are respectively represented by skeletons C5N, C5NC3 and C5NC4 (Tadeusz, 2007). Here C5N means a skeleton with 6-membered ring including 1 nitrogen atom, C5NC3 means the heterocyclic ring skeleton composed by 6-membered ring including 1 nitrogen atom together with 5-membered ring including 1 nitrogen, and C5NC4 means the heterocyclic ring skeleton composed by 6-membered ring including 1 nitrogen atom together with 6-membered ring including 1 nitrogen. C5NC and C5NC4 contains common C-N bonds in two heterocyclic systems. As shown in Fig. 1, the synthetic pathways of quinolizidine alkaloids from L-Lysine (L-Lys) to Lupanine (C2) and Multiflorine (C3) and to Lycodine (C6), three compounds C1, C2, and C3 have identical ring skeleton S1, whereas, C4 has the skeleton S2, and C5 and C6 have the skeleton S3. Here ring skeleton is defined as ring structures without saturated or unsaturated chemical bonds. Thus, classification of alkaloids based on ring skeletons can provide important information for systematic understanding of chemical structures and for predicting biosynthetic pathways. Here, we discuss structure-based classification of secondary metabolites as a multidisciplinary field combining chemo- and bio-informatics.

Data Set

Information of 478 polycyclic compounds were accumulated whose metabolic pathways were available in references. We collected pathway information from scientific literatures and constructed 32 pathway maps (Eguchi *et al.*, 2017). The summary of these 32 pathway maps is depicted in Table 1. We made alkaloid pathway maps considering starting compounds as amino acids, compounds related with amino acid biosynthesis (anthranilate, formyl anthranilate, indole-3-glycerol phosphate, O-methyl tyrosine) terpenes, compounds related with TCA cycle and fatty acids and nucleic acids because alkaloids are often divided into the true alkaloids which originate from amino acids and pseudoalkaloids that do not originate from amino acids, for example, terpene-like, steroid-like and purine-like alkaloids.

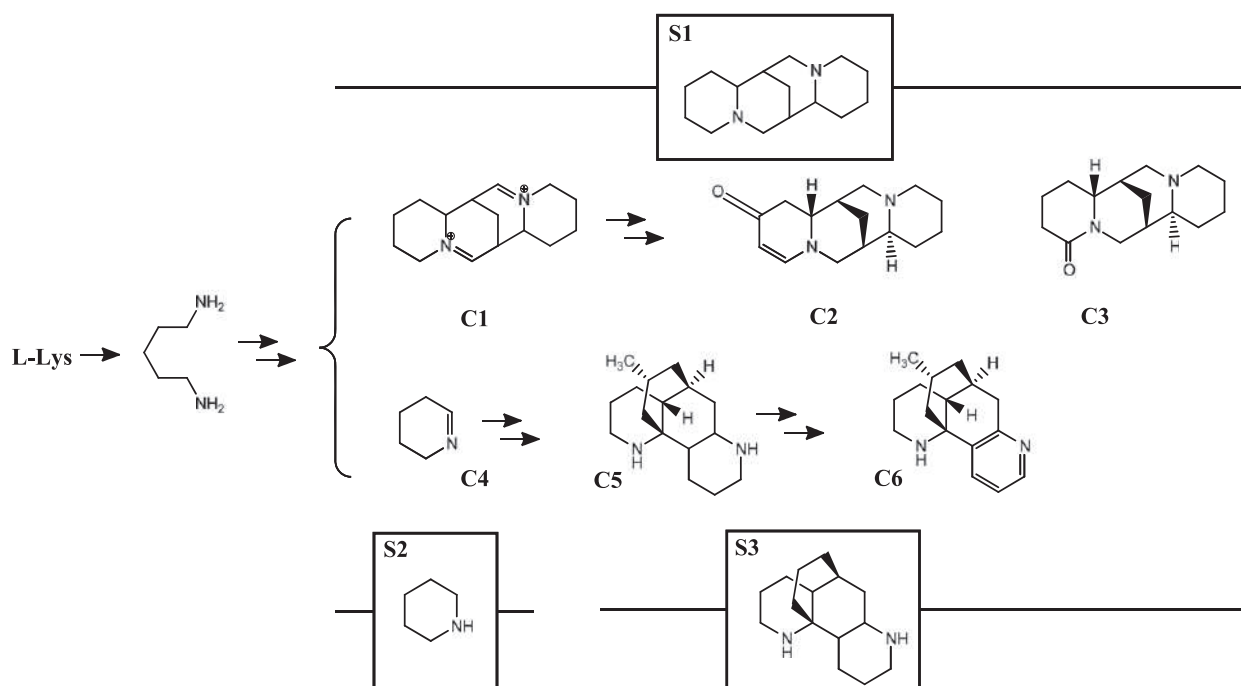


Fig. 1 Concept of subring skeleton structure.

Table 1 Summary of alkaloid metabolic pathways used in this study

Group of SS	3-phospho glycerate	Pyruvate	Phosphoenol pyruvate	Oxaloacetate	alpha-Ketoglutarate	Terpenes	TCA cycle	Fatty acid	Nucleic acids
Starting substance (SS)	Gly L-Ser D-Ser L-Cys L-Ala D-Ala L-Leu L-Val	L-Ala D-Ala L-Leu L-Val	L-Trp D-Trp L-Tyr L-Phe Anthranilate Formyl anthranilate Indole-3-glycerol phosphate O-Methyl tyrosine	L-Asp L-Thr L-Lys L-Arg L-His L-Pro	L-Glu	IPP DMAPP Secologanin GGPP Cholesterol	Acetyl CoA Oxaloacetate	Malonyl-CoA Acetoacetyl CoA	Adenine
Pathway Map ID	1								
	2								
	3								
	4								
	5								
	6								
	7								
	8								
	9								
	10								
	11								
	12								
	13								
	14								
	15								
	16								
	17								
	18								
	19								
	20								
	21								
	22								
	23								
	24								
	25								
	26								
	27								
	28								
	29								
	30								

Subring Skeleton Profiling

Fig. 2 shows the subring skeleton profiling with the example of benzoisoquinoline alkaloid biosynthetic pathway from L-Tyr to Sanguinarine (C13). Out of 7 compounds (C7–C13), C12 and C13 have the same ring skeleton S9 and the others have separate ring skeletons (S4–S8). Next, subring skeletons were produced for individual skeletons (S4–S9). For example, S4 corresponds to nine subring skeletons (SRSs) (A–I in Step 1 of **Fig. 2**). SRS matrix *S* was constructed by setting $s_{ij} = 1$ or 0 respectively depending on presence or absence of the *j*th SRS in the *i*th compound (Step 2). Here we used InChI (the IUPAC International Chemical Identifier) for isomorphism check for SRSs (Heller *et al.*, 2015; Spjuth *et al.*, 2013). In Step 3, hierarchical clustering of the compounds was performed using matrix *S*. There are several molecular fingerprint techniques such as PubChem (881 bits) (Bolton *et al.*, 2008), CDK (1024 bits) (Steinbeck *et al.*, 2003), Extended CDK (1024 bits) (Durant *et al.*, 2002), MACCS (166 bits) (Klekota and Roth, 2008), Klekota–Roth (4860 bits) (PubChem Substructure Fingerprint, see “Relevant Websites section”), Substructure (307 bits) (Person VUE see “Relevant Websites section”), Estate (79 bits) (Minato, 1993), and atom pairs (780 bits) (Iwashita *et al.*, 2012). Those molecular finger prints generally focuses on side-chain substructures of molecules. Alternatively, SRS profiling makes it possible to examine and compare compounds based on all possible SRS, so SRS profiling is useful for systematic understanding of the evolution process of ring systems of the compounds.

Subring Skeleton Profiling in Alkaloids

We extracted 2546 unique subring skeletons from 478 compounds. **Fig. 3** shows the distribution of the numbers of compounds along with the number of SRSs. It is interesting that there is no unique SRS in 478 compounds, that is, there is no SRS corresponding to only one compound.

Secondly, we represented each compound as a 2546 dimensional binary vector. Here, if *i*th subring skeleton is present in a target compound, the *i*th element was set to 1; otherwise, the *i*th element was set to 0. We applied ward clustering method to a matrix consisting of 478 compounds and 2546 unique SRSs and allocated 478 compounds into 29 clusters (Cluster ID = 1 to 29) as shown by the dendrogram in **Fig. 4**. All compounds were assigned to 187 ring skeletons which have been presented in Eguchi *et al.* (2017).

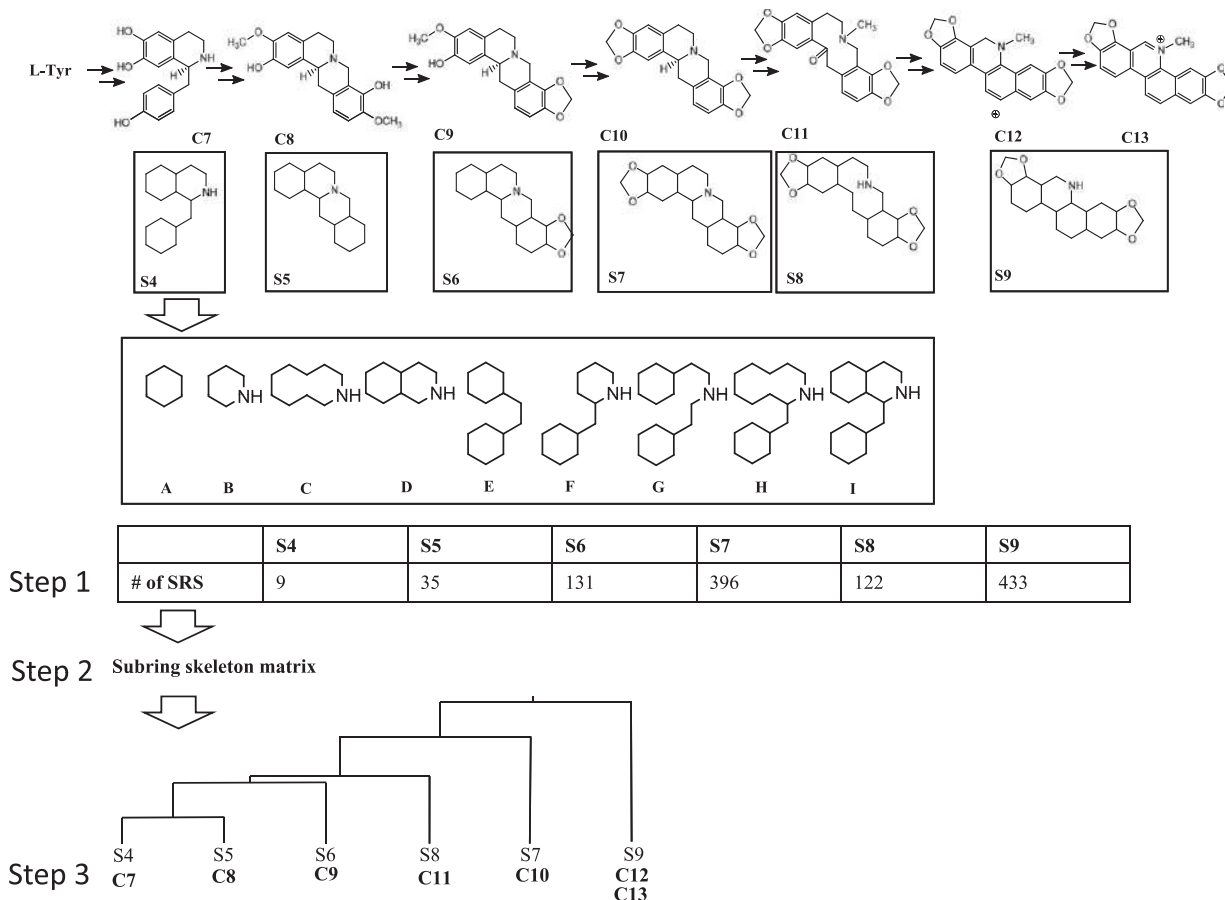


Fig. 2 Schematic diagram of subring skeleton (SRS) profiling.

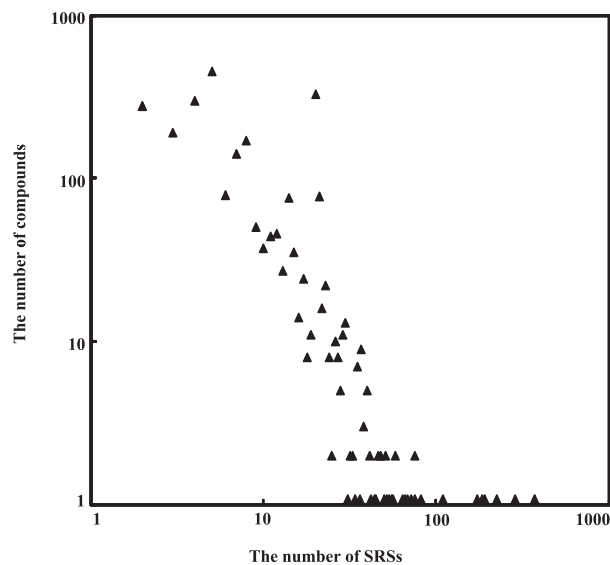


Fig. 3 Distribution of the number of SRSs along with number of compounds.

We divided the compounds into 6 major groups labelled as G1 (Clusters 1 and 2), G2 (Cluster 3), G3 (Cluster 4), G4 (Clusters 5–7), G5 (Clusters 8) and G6 (Cluster 9–29) as shown in [Fig. 4](#); G1 and G2 are related with indole-diterpenes involving paxilline, terpendoles and lolitrems (G1) and strictosides (G2). These consist of polycyclic compounds comprised of six to nine rings characterized by 5–6 member rings and only one nitrogen; Compounds in G3 (Cluster 4) are related with steroidal alkaloids with

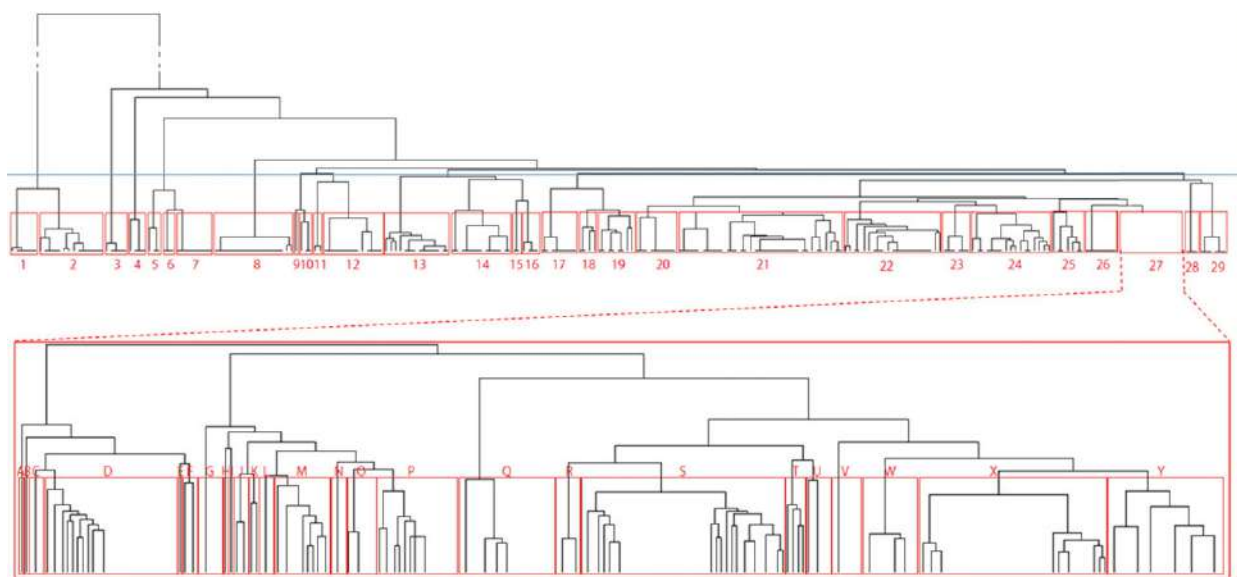


Fig. 4 Dendrogram of 478 compounds based on SRS profiling.

glucosides including alpha-solanine, alpha-chaconine, and alpha-tomatine and their skeletal chemical structures are assigned with steroidal alkaloids; compounds included in G4 (Clusters 5–7) are characterized by vindoline derivatives with greater than or equal to 2 nitrogen atoms; G5 corresponds to Ergot alkaloids. The other alkaloid compounds belong to G6 (Clusters 9–29) and also have very diverged ring skeletons. Clusters 9–20 can be characterized as follows; aglycones and glucosides of Tomatidines (Clusters 9 and 10, respectively), bis-isoquinolines (Cluster 11), morphinans (Cluster 12), quinazoline derivatives (Cluster 13), indolizidine derivatives (Cluster 14), polynuridines (Cluster 15), Ajmalicines (Cluster 16), Berberine and relatives (Clusters 17–19). Ipecac alkaloids (Cluster 20), isoquinolines including ring skeletons of glaudines (Cluster 21), beta-carbolines (Cluster 22), chaetoglobosin (Cluster 23), ring skeletons roquefortines and acetylazonaleins (Cluster 24), Emindoles (Cluster 25), quinolizidines (Cluster 26), iboganines (Cluster 28), and communesins (Cluster 29). Thus the 28 clusters except cluster 27 can be characterized by ring skeletons based on the sub-ring skeleton profiles proposed by [Eguchi et al. \(2017\)](#).

Almost half of the compounds (233 compounds) examined in this study belong to Cluster 27. So we further divide it into 25 sub-clusters to characterize ring skeletons as representatives of groups of compounds. The 25 sub-clusters in Cluster 27 are composed of relatively simple heterocyclic ring skeletons in comparison to other clusters. Five sub-clusters (A, B, C, D, E, F, and H) have indole type alkaloid structures based on ring skeletons, and especially characterized by brevianamides (A); cyclopiazonate derivatives (B) and chanoclavin derivatives, that is, the former corresponds to a five-cyclic ergot alkaloids and the latter corresponds to tri-cyclic ergot alkaloids; chimonanthines (E); and iboga alkaloids (H). Furthermore, specific ring skeletons can be observed in individual sub-clusters such as, ergolines (B and C), cinchona alkaloids (G), iboga alkaloids (H and I), lycopodium alkaloids (J and K), isoquinolines (L), lupine alkaloids (M), quinolones (N), quinolones (O), acridines and quinolones (P), purine alkaloids (Q), and benzyloquinoline alkaloids (T). In summary the conclusion is, alkaloid compounds can be classified and characterized according to generally defined ring systems based on subring skeleton profiles.

Relations Between Ring Structure and Pathways

We classified alkaloid compounds based on subring skeleton profiles and thus related general ring structures to groups of alkaloids. Next, we tried to find associations between modules in alkaloid metabolic pathways and ring structures. We mapped compounds onto 32 pathway maps and show in [Table 2](#) the summary of the relations between classification results and pathway maps. All but four clusters (Clusters 13, 22, 24, 25 and 27) correspond to single pathway maps (PMs). Thus, clusters expressed by the subring skeleton profiles are highly associated to alkaloid biosynthetic pathways. Such trends are also observed in case of the sub-clusters in Cluster 27 (10 out of 25 sub-clusters). Details of the relations between pathway maps and ring structures have been presented in [Eguchi et al. \(2017\)](#).

Structure Activity Relationship Based on Network of Metabolites

Organisms acquired selective functions through evolution to drive functional specialization together with retaining underlying enzymatic genes. Evolutionary pressures caused the appearance of different secondary metabolites and related pathways. These metabolites are important for healthy survival, e.g., pest and pathogen defense and UV-B sunscreens ([Dixon, 2001](#)). The aroma of black Périgord truffle, *T. melanosporum* is produced by secondary metabolism which is used to attract animals for sporulation ([Islam et al., 2013](#)). Thus, plant

Table 2 Classification of alkaloid compounds in clusters with pathway maps

Cluster	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1																														6		
2																														14		
3			5																													
4																															4	
5			3																													
6			3																													
7			3																													
8		17																														
9																																1
10																															3	
11												2																				
12												13																				
13																																
14				13																												
15			2																													
16			4																													
17												8																				
18												4																				
19												8																				
20													9																			
21													12																			
22																																
23																																
24																																
25																																
26																																
28																																
29																																
A																																
B																																
C																																
D																																
E																																
F																																
G																																
H																																
I																																
J																																
K																																
L																																
M																																
N																																
O																																
P																																
Q																																
R																																
S																																
T																																
U																																
V																																
W																																
X																																
Y																																

secondary metabolites are compounds of diverse types of structures and serve as defense weapons used against bacteria, fungi, amoebae, plants, insects, and herbivorous animals and as agents of symbiosis between microbes and plants, nematodes, insects, and higher animals and also as pheromones and pollinators. Secondary metabolites help an organism maintain various interactions with the eco-systems, often adapting to match the environmental needs. Secondary metabolites of plants that color the plant or plant parts e.g., flowers are a good example of this, as the coloring of a flower can attract pollinators and also certain coloring of leaves or stems can defend against attack by animals. Volatile organic compounds (VOCs) are special type of metabolites. VOCs emitted by bacteria and fungi might have the potential to be alternatives to chemical pesticides to protect plants from pests and pathogens. Microbial VOCs are seen as biocontrol agents to control various phytopathogens and as biofertilizers for plant growth promotion. Furthermore, metabolites are deeply involved in human health care. The use of VOCs as biomarkers to detect human diseases is rapidly increasing. Plant metabolites are a major sources of drugs and they are widely used in pharmaceutical industries.

In order to systematize the relations between secondary metabolites and biological activities and to facilitate a comprehensive understanding of the relationships between the chemical structure of metabolites and their biological activities, we developed a metabolite-activity relation database known as the KNApSACk Metabolite Activity DB (see “Relevant Websites section”) (Nakamura *et al.*, 2014). Here, we present a network-based approach to systematically understand the relationship between 3D-chemical structures of metabolites and their biological activities (Ohtana *et al.*, 2014). We initially constructed a 3D-chemical structure similarity based network of secondary metabolites and then applied DPCLUSO (Altaf-Ul-Amin *et al.*, 2006, 2012) algorithm to extract densely connected clusters (referred to as structure groups) of metabolites. We then statistically validated the relationship between the structure groups and biological activities and selected significant structure group-biological activity pairs using a threshold p -value. The p -value threshold was decided based on false discovery rate (FDR) to address the problems associated with multiple testing (Benjamini and Hochberg, 1995). Finally, we reconstructed a binary matrix consisting of selected structure groups and biological activities. Hierarchical clustering was then applied to the constructed matrix in order to detect global trends of relations between 3D structures and biological activities.

The Proposed Method to Assess Structure-Activity Relationships

The concept of the proposed method is depicted in Fig. 5 and the details of the individual steps are described below.

In Step 1, a network was constructed where a node represents a metabolite and an edge represents high 3D structural similarity between metabolites. Fast heuristic graph match algorithm COMPLIG (Saito *et al.*, 2012) was utilized to estimate structural similarity between metabolites at the 3D level. COMPLIG calculates structure similarity based on three factors namely, topology-distance, bond-order, and rotatable-bond and has been previously utilized for systematically classifying ligands in PDB (Saito *et al.*, 2012). We estimated similarity score $S(A, B)$ between two metabolites A and B using the following equations:

$$S(A, B) = M(A, B) / \max\{N(A), N(B)\}$$

Here, $M(A, B)$ is the number of atoms and bonds matched between metabolites A and B , and $N(A)$ and $N(B)$ are the number of atoms and bonds in metabolites A and B , respectively. We considered 0.80 as the threshold similarity which was recommended by Saito *et al.* (2012). Thus, structurally similar metabolites were connected with each other to construct a network of metabolites.

In Step 2, we used a network-clustering algorithm DPCLUSO to classify metabolites into highly structurally similar groups. DPCLUSO generates clusters characterized by high density and separated by periphery in the networks (Altaf-Ul-Amin *et al.*, 2006, 2012), and has been used in several big data analyses in molecular biology such as protein-protein interaction (Altaf-Ul-Amin *et al.*, 2006), metabolomics (Takahashi *et al.*, 2008) and prediction of functional relations between genes using gene expression (Altaf-Ul-Amin *et al.*, 2014).

In Step 3, we annotated structure groups by biological activities using chi-square statistic χ^2 and an over-representation test statistic OV (Kanaya and Kudo, 1994) defined as follows:

$$\chi^2 = \frac{(c(x, k) - f(k) \cdot T)^2}{f(k) \cdot T} + \frac{(c(x, \bar{k}) - f(\bar{k}) \cdot T)^2}{f(\bar{k}) \cdot T}$$

$$OV = \frac{c(x, k) - f(k) \cdot T}{\sqrt{f(k) \cdot T}}$$

Here, $c(x, k)$ and $c(x, \bar{k})$ represent the number of metabolites with and without the k th activity in structure group x , $f(k)$ and $f(\bar{k})$ represent relative numbers of metabolites with and without the k th activity in the database, T represents the total number of metabolites in the structure group x . It is evident that the degree of freedom is 1 for the χ^2 value represented above.

A structure group can have significant p -values in χ^2 -statistics either if it is overrepresented or underrepresented by metabolites of certain biological activity in the context of the ratio of the metabolites in the DB. However, our purpose is to relate a structure group to a biological activity if and only if the structure group is overrepresented by metabolites associated with that biological activity. Therefore, we first make sure that the OV statistic is positive for a structure group and then we evaluate its p -value.

In Step 4, we construct a data matrix based on the statistically significant relation between structure groups and biological activities, and carried out 2-dimensional hierarchical clustering for systematizing the relationship between metabolite structures and biological activities.

Selection of Statistically Significant Relationships Based on False Discovery Rate

We examined 3D-similarities among 2072 secondary metabolites by a fast heuristic graph-matching algorithm COMPLIG developed by Shirai and colleagues (Saito *et al.*, 2012) and then 50,228 pairs of secondary metabolites having similarity score more than the threshold were selected to construct a network. Then structure groups were generated using the graph clustering algorithm DPCLUSO (Altaf-Ul-Amin *et al.*, 2012) which was developed for overlapping clustering using the similar concepts of the DPCLUS algorithm (Altaf-Ul-Amin *et al.*, 2006). DPCLUSO generated 671 densely connected clusters of secondary metabolites, which are called structure groups (SGs) in this study.

We searched for statistically significant relationships between 671 SGs and 140 biological activities (Table 3). The biological activities are of mainly two types: (i) Chemical ecology related activities indicated as E01–E48 in Table 3 which are associated with metabolites involved in ecological interactions between species for survival and normal regulation of organisms, and (ii) human healthcare and medicine related activities indicated as M01–M92 which are associated to metabolites known as ingredients of healthy foods and medicines or biomarkers.

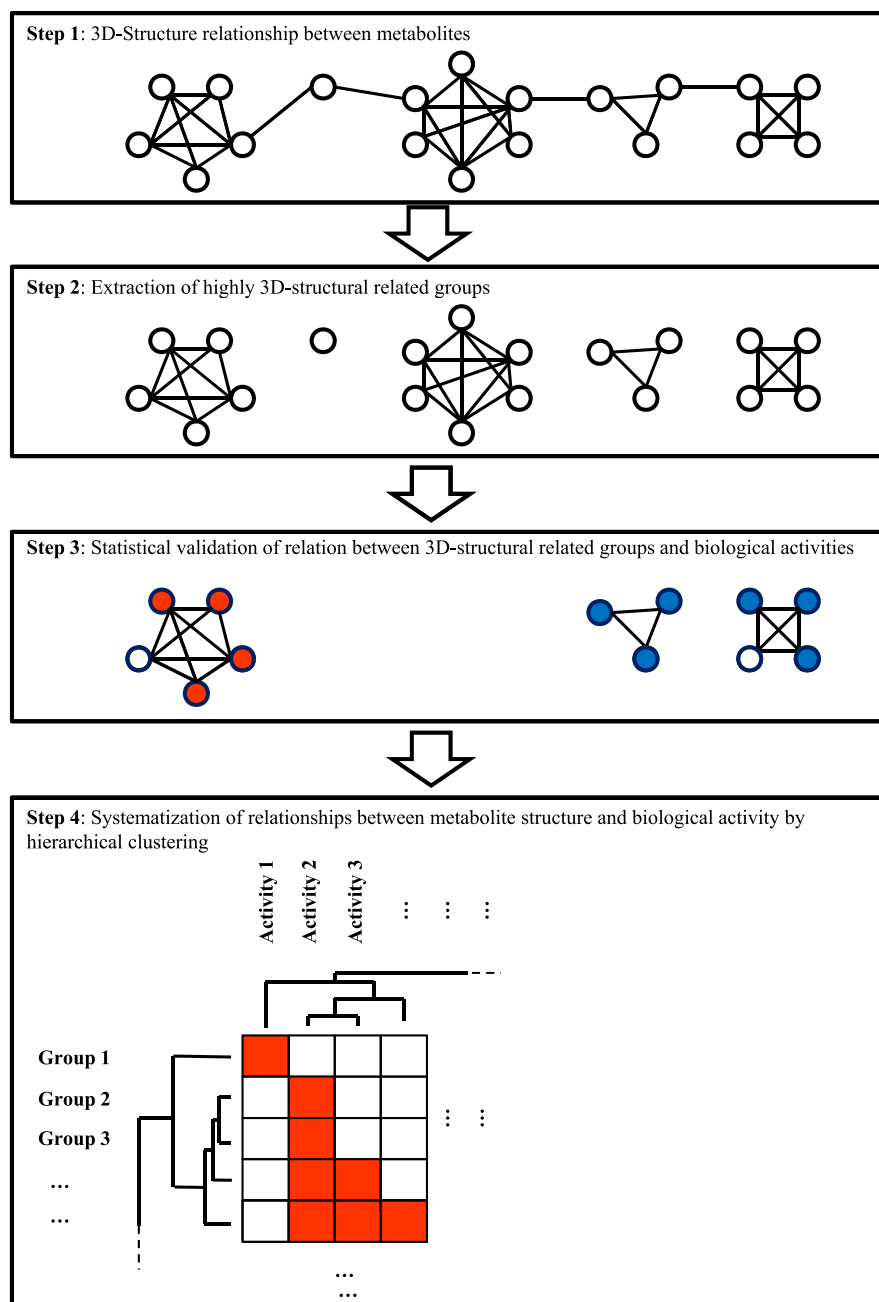


Fig. 5 Concept of the methodology: network construction based on 3D-structure similarity between metabolites. Nodes and edges represent metabolites and highly similar pairs of metabolites, respectively. Different colors in Step 3 represent different biological activities for individual metabolites. In Step 4, 2D-hierarchical clustering is carried out based on statistically significant relations between structure groups and biological activities. Red colors represent significant relation between corresponding structure group and biological activity.

We first selected 7650 structure group-biological activity pairs (SG-BA pairs) for which *OV* is positive to ensure over-representation of metabolites of the activity category in the associated cluster. We then examined FDR (cumulative *q*-values) using *p*-values calculated based on χ^2 -statistics, and finally selected 983 SG-BA pairs which correspond to $p < 0.001$ and $q < 7.78 \times 10^{-3}$.

Relation Between Structure and Biological Activities

Multiple activities are generally reported for a metabolite, for example, (+)-catechin has been reported to have several biological activities in chemical ecology (e.g., E18, E20 and E32 in [Table 3](#)) and in human healthcare and medicine (e.g. M29, M34, M39,

Table 3 Biological activities defined in the KNApSack Metabolite Activity DB

General description	Activity category
Plant growth regulator	Enhance germination (E01), Enhance stem growth (E02), Enhance root growth (E03), Enhance leaf growth (E04), Enhance flowering (E05), Enhance fruiting (E06), Enhance plant growth (E07), Inhibit seed germination (E08), Inhibit stem growth (E09), Inhibit root growth (E10), Inhibit leaf growth (E11), Inhibit flowering (E12), Inhibit fruiting (E13), Inhibit plant growth (E14), Allelopathic (E15), Phytoalexin (E16)
Attractant/repellent	Feeding attractant (E17), Feeding deterrent (E18), Pollinator attractant (E19), Oviposition attractant (E20), Oviposition deterrent (E21), Sex attractant (E22), Attractant (E23), Repellent (E24)
Selective toxicity	Phytotoxic (E25), Herbicidal (E26), Insecticidal (E27), Acaricidal (E28), Molluscicidal (E29), Piscicidal (E30), Nematocidal (E31)
Antimicrobial agent	Antibacterial (E32), Antituberculosis (E33), Antileprotic (E34), Antifungal (E35), Inhibit spore germination (E36), Antimicrobial (E37)
Antiviral agent	Antiviral (E38), Antihepatitic (E39), Anti-HIV (E40), Anti-HSV (E41)
Antiparasitic agent	Anthelmintic (E42), Antiprotozoal (E43), Antiamebic (E44), Antimalarial (E45), Antileishmanial (E46), Antitrypanosomal (E47), Pediculicide (E48)
Nervous system agent	Antipyretic (M01), Analgesic (M02), Antiarthritic (M03), Anesthetics (M04), Sedative (M05), Antispasmodic (M06), Anticonvulsant (M07), Antidementic (M08), Antidepressant (M09), CNS Stimulant (M10), Diaphoretic (M11), Emetic (M12), Antiemetic (M13), Antigout (M14), Antimigraine (M15), Antimyasthenic (M16), Antiparkinson (M17), Antipsychotic (M18), Muscle relaxant (M19)
Cardiovascular agent	Antidiabetic (M20), Hemostatic (M21), Antithrombotic (M22), Cardiotonic (M23), Antiarrhythmic (M24), Diuretic (M25), Antihypertensive (M26), Antihyperlipidemic (M27), Antianemic (M28), Other cardiovascular agent (M29)
Respiratory tract agent	Antitussive (M30), Expectorant (M31), Antiasthmatic (M32), Other respiratory tract agent (M33)
Digestive organ agent	Antidiarrheic (M34), Carminative (M35), Stomachic (M36), Laxative (M37), Choloretic (M38), Antihepatotoxic (M39), Other digestive organ agent (M40)
Genitourinary agent	Oxytocic (M41), Antifertility (M42), Abortifacient (M43), Other genitourinary agent (M44)
Anticancer agent	Antioxidant (M45), Anticancer (M46), Antitumor (M47), Antineoplastic (M48), Antimutagenic (M49)
Anti-inflammatory agent	Anti-inflammatory (M50), Antiallergic (M51), UV shield (M52), Antidermatitic (M53), Antiedemic (M54)
Immunological agent	Immunosuppressant (M55), Immunostimulant (M56), Immunomodulator (M57)
Nutrient	Nucleic acid (M58), Essential amino acid (M59), Nonessential amino acid (M60), Vitamin (M61), Nutrient (M62), Tonic (M63)
Nontherapeutic agent	Solvent (M64), Flavor (M65), Odor (M66), Pigment (M67), Emulsifying agent (M68), Antiseptic (M69)
Other health agent	Antilulcerogenic (M70), Depilatory (M71), Antidote (M72), Hormonal (M73), Dental (M74), Other health agent (M75)
Narcotic	Narcotic (M76)
Toxic	Phototoxic (M77), Neurotoxic (M78), Pneumotoxic (M79), Hepatotoxic (M80), Cytotoxic (M81), Toxic (M82)
Tumorigenic	Tumorigenic (M83), Mutagenic (M84), Genotoxic (M85), Teratogenic (M86)
Other disease-causing agent	Psychotomimetic (M87), Hemolytic (M88), Allergenic (M89), Irritant (M90), Dermatitic (M91), Edematous (M92)

M40, M45, M48, M70, M81, M82, M88 and M91 in [Table 3](#)) which are easily retrieved by entering “(+)-catechin” as Metabolite Name in the Metabolite Activity DB ([Nakamura et al., 2014](#)). To systematically understand the relations between biological activities and structure groups, we constructed a 2-dimensional dendrogram, using a matrix showing relations among 488 structure groups and 139 activities.

[Fig. 6](#) presents the relationship between SGs and biological activities in the perspective of hierarchical clustering. We tentatively distributed 139 biological activities into 13 clusters called BC1 to BC13. Relationships between major biological activities that are connected with 10 or more structure groups are listed in [Table 4](#). It should be noted that activity categories in three clusters (BC1, BC2 and BC3) described as antimicrobial agent and plant growth regulator are related to ecology and six clusters (BC4, BC5, BC7, BC10, BC11 and BC13) are associated with human healthcare. On the other hand, three clusters (BC6, BC9 and BC12) are related to both ecology and healthcare activities.

To examine the relationship between chemical structures and biological activities, we selected SGs that are related to single activity category ([Table 3](#) shows the list of activity categories) and assigned chemical structural properties into individual SGs (SG-ID in [Tables 4](#) and [5](#), SG-ID 389 is abbreviated as SG389 and so on in the following text). Three clusters of major metabolites namely isoflavonoids (BC1), flavones and isoflavones (BC2) and giberrellanes (BC3) were assigned as ecological activity agents. BC1 exhibits antifungal properties (E35) whereas BC2 contained phytoalexins (E16) and BC3 are characterized as plant growth regulators with properties such as stem growth enhancement (E02), plant growth enhancement (E07) and root growth inhibition (E10). BC6 mainly consists of phenolic substances with both ecological and healthcare activities, i.e. anthraquinones (SG113), coumarins (SG236 and SG237) flavones (SG147), phenylpropanoids (SG327 and SG404) as antibacterial agents (E32), and lignans (SG43, SG75, SG82, SG278 and SG376) as antitumor agents (M47), and stilbenoids (SG147) as antibacterial agents (E32). Flavonoids including isoflavonoids have diverse functions as both ecological and healthcare agents because most of the bioactivity groups (BC1, BC2, BC4, BC5, BC6, BC9 and BC12) include various structural derivatives and thus specific functions can be exerted by patterns of substitution groups in the chemical structures. Ecological activities of these metabolites are highly related with core chemical structures most likely because these metabolite groups characterize certain biosynthetic pathways. This leads to

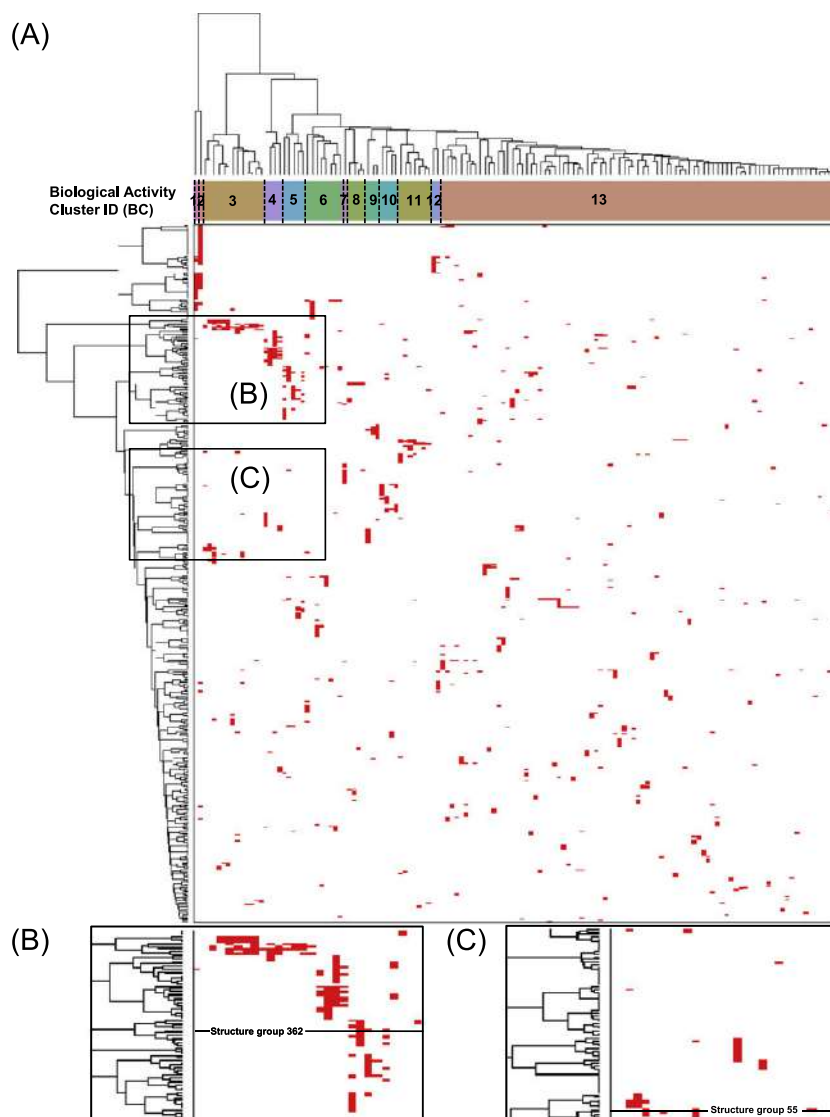


Fig. 6 2D-hierarchical clustering (Ward's method) between metabolite groups (vertical direction) and biological activities (horizontal direction). The whole map (A) and enlarged views (B) and (C). Index numbers in horizontal clustering represent cluster ID.

the explanation that the role of metabolites in plant physiological activities is highly related with evolution of biosynthetic pathways in plants.

SGs listed in the healthcare category of [Table 4](#) correspond to single activities but the extent of their effects in plants has not been fully understood. Efficacies in humans of many SGs consisting of alkaloids can be mentioned: BC5, isoquinoline alkaloids (SG217) as analgesic agents (M02), quinoline alkaloids (SG430) as sedatives (M05), terpene alkaloids (SG152 and SG220) as antihypertensivity agents (M26); BC6, indole alkaloids (SG651) with antitumor properties (M47); BC7, quinolizidine alkaloids (SG489) and terpene alkaloids (SG112 and SG267) with toxicity properties (M82); BC9, betalains (SG172, SG289, SG419, SG420, SG438, SG439 and SG497) as pigments (M67); BC11, steroid alkaloids (SG461) as laxatives (M37); BC13, pyrrolizidine alkaloids (SG320 and SG479) and quinolizidine alkaloids (SG429) as antidiabetic agents (M20), tropane alkaloids (SG578), and indole alkaloids (SG649) with hormonal properties (M73), quinoline alkaloids (SG261 and SG298) as phototoxic agents (M77), isoquinoline alkaloids (SG361, SG447 and SG644) and indole alkaloids (SG202 and SG317) with psychotomimetic properties (M87).

Alkaloids occur in 20% of all plants and the number of identified structures are more than 12,000 ([Connolly and Hill, 1991](#); [Ziegler and Facchini, 2008](#)). Initially in the 1950s, most of these alkaloids were believed to be metabolic waste products ([Wink et al., 1998](#)). Currently biological activities of alkaloids found in nature have been determined and the medical effects of alkaloids to humans are well known. Those include toxicity to animals, vertebrates and arthropods, effects of various metabolic systems in animals, inhibiting DNA synthesis and repair by intercalation with nucleic acids, and other reported effects on the nervous systems ([Mithöfer and Boland, 2012](#)). Specific healthcare activities can also be explained by class of metabolites,

Table 4 Related active categories in hierarchical clustering in Fig. 4

ID	Ecological activity			Healthcare		
	General description	Activity category	SG-ID	General description	Activity category	SG-ID
1	Antimicrobial agent	Antifungal (E35)	36, 68, 78, 250, 270, 389			
2	Plant growth regulator	Phytoalexin (E16)	7, 8, 11, 14, 16, 17, 22, 25, 27, 33, 49, 64, 70, 71, 85, 93, 115, 186, 269, 279			
3	Plant growth regulator	Enhance stem growth (E02) Enhance flowering (E05) Enhance fruiting (E06) Enhance plant growth (E07) Inhibit root growth (E10)	392, 594 511 510			
4				Nontherapeutic agent	Flavor (M65) Odor (M66)	319, 349, 624 409
				Other health agent	Antiseptic (M69)	97, 230
				Digestive organ agent	Carminative (M35)	469, 575
5				Cardiovascular agent	Antihypertensive (M26)	152, 220, 366
				Nervous system agent	Analgesic (M02) Sedative (M05)	217 430
				Narcotic	Narcotic (M76)	
6	Plant growth regulator	Allelopathic (E15)	331, 354, 460, 601	Anticancer agent	Antitumor (M47)	43, 75, 82, 278, 376, 651
	Selective toxicity	Antibacterial (E32)	113, 147, 236, 237, 327, 404			
7				Toxic	Toxic (M82)	112, 239, 267, 373, 411, 488, 489, 523
8			(no data)			(no data)
9	Attractant/repellent	Oviposition attractant (E20)	145, 167	Nontherapeutic agent	Pigment (M67)	40, 172, 289, 419, 420, 438, 439, 497, 574
10				Nutrient	Essential amino acid (M59)	231, 294, 314, 546, 639, 640
				Genitourinary agent	Other genitourinary agent (M44)	387, 412
11				Digestive organ agent	Laxative (M37)	241, 252, 449, 461, 655
12	Attractant/repellent	Feeding deterrent (E18)	104, 146, 183, 453	Anticancer agent	Antioxidant (M45)	2, 4, 9
13				Cardiovascular agent	Antidiabetic (M20)	320, 429, 479, 541, 578
				Other health agent	Hormonal (M73)	593, 622, 649
				Toxic	Phototoxic (M77)	98, 254, 261, 276, 298
					Neurotoxic (M78)	357, 483, 583, 654
				Other disease-causing agent	Psychotomimetic (M87)	202, 317, 361, 447, 644

especially class of alkaloids listed in Tables 4 and 5. Comparative activity relationships between ecological and healthcare agents lead to systematization of meta-metabolomics combined with human and ecological metabolic pathways (Raes and Bork, 2008; Ibrahim and Anishetty, 2012). As an example the structure group 55 (Fig. 7(A)) is significantly related with multiple biological activities, that is, plant growth regulators such as enhancing stem growth (E02), enhancing leaf growth (E04), and enhancing plant growth (E07), and consists of 21 metabolites which are highly related with gibberellin and its relatives. Another example, structure group 362 (Fig. 7(B)) is associated with the following healthcare activities: sedative (M05), analgesic (M02), anti-spasmodic (M06) under the general description nervous system agent, antitussive (M30) under respiratory tract agent; narcotic (M76) under narcotic; and antidiarrheic (M34) under digestive organ agent. The core structure in group 362 can be identified as morphinoids.

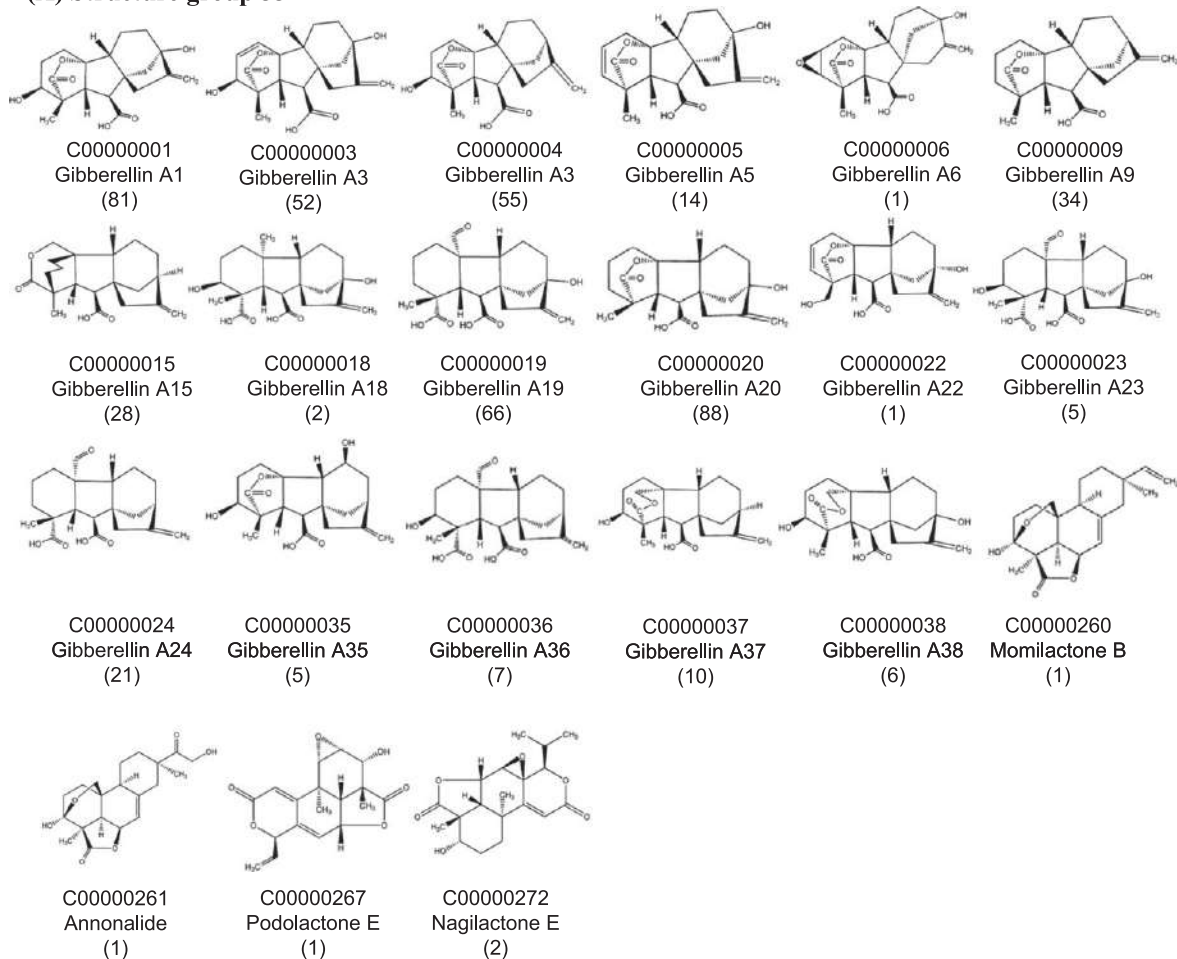
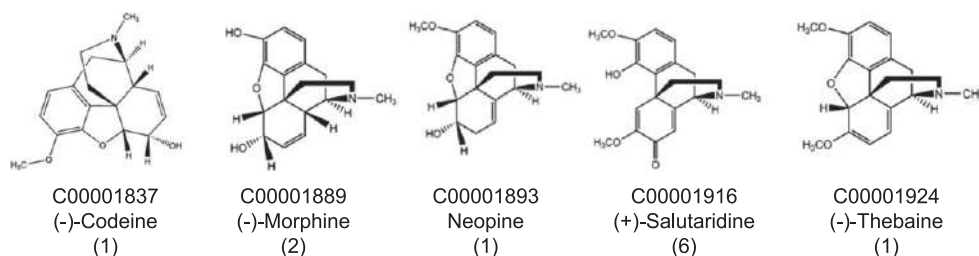
Table 5 Structure properties for SGs in Table 2. The number of metabolites in SG is represented in parentheses

BC	Structure properties	SG-ID
1	Coumestrols	389(5)
	Isoflavonoids	36(30), 68(18), 78(16), 250(7), 270(7)
2	Anthraquinones	8(55), 16(47)
	Flavones	7(56), 8(55), 11(51), 14(50), 16(47), 17(46), 22(40), 25(37), 27(37), 49(24), 70(18), 186(9), 269(7)
	Flavanols	27(37)
	Flavanones	8(55), 85(16), 279(7)
	Flavonoids	33(33), 64(19)
	Isoflavanones	71(18), 115(13)
	Isoflavones	11(51), 16(47), 27(37), 71(18), 115(13)
	Lignans	71(18), 115(13)
	Pterocarpanes	93(15)
3	Gibberellanes	392(5), 510(3), 594(3)
	Triholosides	511(3)
4	Biflavonoids	409(5)
	Cymenes	575(3)
	Furans	624(3)
	Glucosinolates	319(6)
	Nonalactones	349(5)
	Phenols	230(7)
	Phenylpropanoids	97(15), 469(4)
5	Coumarins	366(5)
	Isoquinoline alkaloids	217(8)
	Quinoline alkaloids	430(5)
	Terpene alkaloids	152(10), 220(8)
6	Anthraquinones	113(13)
	Coumarins	236(7), 237(7), 601(3)
	Sesquiterpenes	331(5), 354(5)
	Flavones	147(11)
	Indole alkaloids	651(5)
	Lignans	43(26), 75(17), 82(16), 278(7), 376(5)
	Phenylpropanoids	327(6), 404(5)
	Sesquiterpenes	460(4)
	Stilbenoids	147(11)
7	Cibarians	523(3)
	Cyanogenic glycosides	239(7), 373(5), 488(4)
	Heterodendrins	411(5)
	Quinolizidine alkaloids	489(4)
	Terpene alkaloids	112(13), 267(7)
9	Anthocyanins	574(3)
	Anthraquinones Betalains	40(27)
	Betalains	172(9), 289(6), 419(5), 420(5), 438(4), 439(4), 497(4)
	Flavanones	145(11), 167(10)
10	Amino acids	231(7), 294(6), 314(6), 546(3), 639(3), 640(3), 412(5)
	Lignans	387(5)
11	Anthraquinones	252(7)
	Carbazoles	449(4)
	Glycosides iridoids steroid Alkaloids	449(4)
	Iridoids	241(7), 655(3)
	Steroid alkaloids	461(4)
12	Flavanones	9(54), 104(14), 146(11), 183(9)
	Flavones	2(89), 4(75)
	Lycoricidines	453(4)
13	Amino acids	357(5)
	Benzofurans	254(7)
	Benzopyrans	276(7)
	Carboxylic acids	483(4)
	Indole alkaloids	202(8), 317(6), 649(3)
	Isoquinoline alkaloids	361(5), 447(4), 644(3)
	Monoterpenoids Phenylpropanoids	583(3)
	phenylpropanoids	98(15) 593(3), 622(3)

(Continued)

Table 5 Continued

BC	Structure properties	SG-ID
	Pyrrolizidine alkaloids	320(6), 479(4)
	Quinoline alkaloids	261(7), 298(6)
	Quinolizidine alkaloids	429(5)
	Skytanthines	541(3)
	Tropane alkaloids	578(3)
	Usambarensines	654(3)

(A) Structure group 55**(B) Structure group 362****Fig. 7** Chemical structures belonging to structure groups. (A) Structure groups 55 and (B) 362. ID in KNApSack Core DB, name of the metabolite and the number of species reported in parentheses are described in this order.

Conclusions

Bioinformatics and Cheminformatics play important roles in interpretation and understanding chemical and biological data. Two of the major targets of Big Data science are to advance human health care and to understand ecosystems based on huge distributed data. Secondary metabolites are used by humans for various purposes and they play important roles in ecological relationships between species. The evolutionary history and broad functional spectrum of secondary metabolites is still not fully understood. In the present book article, we discuss two research topics, first is on alkaloid synthesis pathways and the other is on structure activity relationships of metabolites. In the first study we utilized the subring skeleton (SRS) profiles to systematize alkaloids into groups and to relate individual compounds and groups to metabolic pathways. Applying clustering method to SRS profiles, the variety of ring skeletons across 29 clusters and 25 subclusters was clearly understood. Systematization of core cyclic structure, biological activity and biosynthetic pathway of metabolites can provide new insight into interpretation of evolution process for active compounds in nature. The second topic is on a methodology for the integrated interpretation of the relationships between chemical structures and biological activities based on network clustering algorithm and taking into consideration statistics controlled by false discovery rate. Given the importance of metabolites in agriculture, ecology and healthcare it would be of great help to various sectors of science and technology to predict the functions of the function-unknown metabolites by using some computational means based on their chemical structures. Natural product chemistry associated with metabolomics will shift towards the data-intensive sciences and needs to simultaneously integrate and examine different types of data such as chemical structure and biological activity for new discoveries and development of novel drugs interacted with proteins and the identification of viable drug resources and other useful compounds. Techniques and approaches discussed in this article provide clues to comprehensive understanding of evolution of metabolic pathways and structure-activity relationships in the context of metabolites.

See also: Biological Pathways. Integrative Analysis of Multi-Omics Data. Metabolome Analysis. Natural Language Processing Approaches in Bioinformatics. Networks in Biology

References

- Afendi, F.M., Okada, T., Yamazaki, M., *et al.*, 2011. KNApSack family databases: Integrated metabolite – Plant species databases for multifaceted plant research. *Plant and Cell Physiology* 53 (2), e1–e1.
- Albert, R., Jeong, H., Barabási, A.-L., 2000. Error and attack tolerance of complex networks. *Nature* 406, 378–382.
- Altat-UI-Amin, M., Katsuragi, T., Sato, T., Ono, N., Kanaya, S., 2014. An unsupervised approach to predict functional relations between genes based on expression data. *BioMed Research International* 2014.
- Altat-UI-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., Kanaya, S., 2006. Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics* 7 (1), 207.
- Altat-UI-Amin, M., Wada, M., Kanaya, S., 2012. Partitioning a PPI network into overlapping modules constrained by high-density and periphery tracking. *ISRN Biomathematics* 2012, 11.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 289–300.
- Bolton, E.E., Wang, Y., Thiessen, P.A., Bryant, S.H., 2008. PubChem: Integrated platform of small molecules and biological activities. *Annual Reports in Computational Chemistry* 4, 217–241.
- Connolly, J.D., Hill, R.A., 1991. *Dictionary of Terpenoids*. London: Chapman and Hall.
- Dixon, R.A., 2001. Natural products and plant disease resistance. *Nature* 411 (6839), 843.
- Durant, J.L., Leland, B.A., Henry, D.R., Nourse, J.G., 2002. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences* 42 (6), 1273–1280.
- Eguchi, R., Ono, N., Horai, H., *et al.*, 2017. Classification of alkaloid compounds based on Subring Skeleton (SRS) profiling: On finding relationship of compounds with metabolic pathways. *Journal of Computer Aided Chemistry* 18, 58–75.
- Fell, D.A., Wagner, A., 2000. The small world of metabolism. *Nature Biotechnology* 18 (11), 1121–1122.
- Heller, S.R., McNaught, A., Pletnev, I., Stein, S., Tchekhovskoi, D., 2015. InChI, the IUPAC international chemical identifier. *Journal of Cheminformatics* 7 (1), 23.
- Ibrahim, M., Anishetty, S., 2012. A meta-metabolome network of carbohydrate metabolism: Interactions between gut microbiota and host. *Biochemical and Biophysical Research Communications* 428 (2), 278–284.
- Islam, M.T., Mohamedali, A., Garg, G., *et al.*, 2013. Unlocking the puzzling biology of the black Périgord truffle *Tuber melanosporum*. *Journal of Proteome Research* 12 (12), 5349–5356.
- Iwashita, H., Kawahara, J., Minato, S.I., 2012. ZDD-based computation of the number of paths in a graph. *Hokkaido University, Division of Computer Science, TCS Technical Reports*, vol. TCS-TR-A-10-60.
- Jeong, H., Tombor, B., Albert, R., Olvai, Z.N., Barabási, A.L., 2000. The large-scale organization of metabolic networks. *Nature* 407 (6804), 651–654.
- Kanaya, S., Kudo, Y., 1994. Genome propensities of *Escherichia coli* and eleven coliphages for under- and over-abundances of sigma super (70)-consensus-like sequences. *International Journal of Genome Research* 1 (4), 261–277.
- Klekota, J., Roth, F.P., 2008. Chemical substructures that enrich for biological activity. *Bioinformatics* 24 (21), 2518–2525.
- Koch, I., Junker, B.H., Heiner, M., 2004. Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics* 21 (7), 1219–1226.
- Kossel, A., 1891. On the chemical composition of the cell. *Archives of Physiology* 181.
- Matsuno, H., Tanaka, Y., Aoshima, H., Matsui, M., Miyano, S., 2011. Biopathways representation and simulation on hybrid functional petri net. *Studies in Health Technology and Informatics* 162, 77–91.
- Ma, H., Zeng, A.P., 2003. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics* 19 (2), 270–277.

- Minato, S.I., 1993. Zero-suppressed BDDs for set manipulation in combinatorial problems. In: Proceedings of the 30th International Design Automation Conference, pp. 272–277. ACM.
- Mithöfer, A., Boland, W., 2012. Plant defense against herbivores: Chemical aspects. *Annual Review of Plant Biology* 63, 431–450.
- Nakamura, Y., Mochamad Afendi, F., Kawsar Parvin, A., *et al.*, 2014. KNApSAcK metabolite activity database for retrieving the relationships between metabolites and biological activities. *Plant and Cell Physiology* 55 (1), e7.
- Ohtana, Y., Abdullah, A.A., Altaf-Ul-Amin, M., *et al.*, 2014. Clustering of 3D-structure similarity based network of secondary metabolites reveals their relationships with biological activities. *Molecular Informatics* 33 (11–12), 790–801.
- Palsson, B.Ø., 2006. *Systems Biology*. Cambridge University Press.
- Raes, J., Bork, P., 2008. Molecular eco-systems biology: Towards an understanding of community function. *Nature Reviews Microbiology* 6 (9), 693.
- Saito, M., Takemura, N., Shirai, T., 2012. Classification of ligand molecules in PDB with fast heuristic graph match algorithm COMPLIG. *Journal of Molecular Biology* 424 (5), 379–390.
- Spjuth, O., Berg, A., Adams, S., Willighagen, E.L., 2013. Applications of the InChI in cheminformatics with the CDK and Bioclipse. *Journal of Cheminformatics* 5 (1), 14.
- Steinbeck, C., Han, Y., Kuhn, S., *et al.*, 2003. The Chemistry Development Kit (CDK): An open-source Java library for chemo-and bioinformatics. *Journal of Chemical Information and Computer Sciences* 43 (2), 493–500.
- Strogatz, S.H., 2001. Exploring complex networks. *Nature* 410 (6825), 268–276.
- Tadeusz, A., 2007. Alkaloids – Secrets of life alkaloid chemistry. In: *Biological Significance, Applications and Ecological Role*. Elsevier.
- Takahashi, H., Kai, K., Shinbo, Y., *et al.*, 2008. Metabolomics approach for determining growth-specific metabolites based on Fourier transform ion cyclotron resonance mass spectrometry. *Analytical and Bioanalytical Chemistry* 391 (8), 2769.
- Wink, M., Schmeller, T., Latz-Brüning, B., 1998. Modes of action of allelochemical alkaloids: Interaction with neuroreceptors, DNA, and other molecular targets. *Journal of Chemical Ecology* 24 (11), 1881–1937.
- Ziegler, J., Facchini, P.J., 2008. Alkaloid biosynthesis: Metabolism and trafficking. *Annual Review of Plant Biology* 59, 735–769.

Relevant Websites

- http://kanaya.naist.jp/knapsack_jsp/top.html
KNApSAcK Core System.
- <http://kanaya.naist.jp/MetaboliteActivity/top.jsp>
Metabolite Activity.
- <http://www.pearsonvue.com/fi/realestate/fingerprint/>
Pearson VUE.
- http://astro.temple.edu/~tua87106/list_fingerprints.pdf
PubChem Substructure Fingerprint.

Biomolecular Structures: Prediction, Identification and Analyses

Prasun Kumar, University of Bristol, Bristol, United Kingdom

Swagata Halder and Manju Bansal, Indian Institute of Science, Bangalore, India

© 2019 Elsevier Inc. All rights reserved.

Glossary

Active site It is an asymmetric pocket, present on or near the surface of a biomolecule, that promotes chemical catalysis upon binding of appropriate substrate.

Alignment Method for comparing two or more sequences/structures to assess the overall similarity.

Base stacking Aromatic rings of DNA or RNA bases lie on top of each other.

Coiled-coils These structures are formed when two or more α -helices wind around each other to give a supercoil. The constituent helices may either run in the same (parallel) or in the opposite (anti-parallel) directions.

Domain A compact unit of protein structure that can fold on its own and have independent function.

Electrostatic interactions The non-covalent interaction between oppositely charged atoms or groups of atoms.

Homology modelling A computational method for modelling the structure of a biomolecule based on its sequence similarity to one or more biomolecules of known structure.

Motif It can be either sequence or structure motif.

Sequence motifs have recognizable amino-acid sequences found in different proteins. Structural motifs are segments of protein 3D structure that is formed by the spatially close residues. These residues may or may not be adjacent in the sequence.

RNA-thermometer RNA-thermometer, also known as RNA-thermosensor, is a temperature sensitive non-coding RNA molecule that regulates gene expression.

Super-secondary structure Super-secondary structures are combination of secondary structures and can be considered as an intermediate between secondary and tertiary structures. α -hairpins, β -hairpins and β - α - β motifs are commonly found super-secondary structures.

Van der Waals interaction These are weak attractive forces between two atoms or groups of atoms that arise due to the fluctuations in electron distribution around the nuclei. Van der Waals forces are stronger between less electronegative atoms.

Central Dogma of Life

According to the classical view, also known as Crick's central dogma, "the coded genetic information hard-wired into DNA is transcribed into individual transportable cassettes, composed of messenger RNA (mRNA); each mRNA cassette contains the program for synthesis of a particular protein (or small number of proteins)" (Lodish *et al.*, 2000) (Fig. 1). Though all biological cells adhere to this rule, there are few notable exceptions as well (Birney *et al.*, 2007; Gerstein *et al.*, 2007). Over-simplification of this rule can allow us to make a statement: "DNA makes RNA and RNA makes proteins" and involves two distinct steps, namely transcription and translation. RNA polymerase along with transcription factors transfers the information coded in DNA to messenger RNA and the process is known as transcription. Translation process facilitates the sequence encoded in the RNA molecule to get decoded into an amino acid sequence which defines a protein. DNA must make identical copies of itself to pass on the genetic information from parent to offspring. DNA polymerase and its associated proteins make possible this phenomenon, known as replication. These events are highly dependent on the 3D structures of DNA/RNA as well as associated proteins. In this article, we will discuss various structural elements of DNA, RNA and proteins. We will also highlight different algorithms for their prediction, identification and analyses, with emphasis on more recent reports.

Base-Pairing and Double Helical Structure of DNA

The double helical structure of DNA, as proposed by Watson and Crick in 1953 (shown schematically in Fig. 2(A)), revolutionized our understanding of biology at the molecular level (Watson and Crick, 1953). The model had two strands entwining around a common helical axis and linked together by a specific Hydrogen-bond (H-bond) scheme (Fig. 2(B)) that plays a major role in stabilizing the nucleic acid structures. The canonical Watson-Crick DNA structure has right handed screw sense with the two chains in anti-parallel orientations and 10 nucleotides per turn. The separation among two consecutive bases in a chain is 3.4 Å along the helix axis (Fig. 2(A)). In general, there are two types of H-bonding in nucleic acid structures (1) Watson-Crick (WC) and (2) non-Watson-Crick (NWC). In the canonical Watson-Crick scheme two H-bonds occur between the Adenosine and the Thymine base-pairs (bps), while there are three such bonds between the Cytosine and the Guanine (Fig. 2(B)). The specificity of this base-pairing scheme is the key to conservative replication of DNA and the transmission of exact information from one generation to the next. However, the large number of nucleic acid structures determined by x-ray diffraction and available in various databases, such as Nucleic Acid Database (NDB; see Relevant Website section)

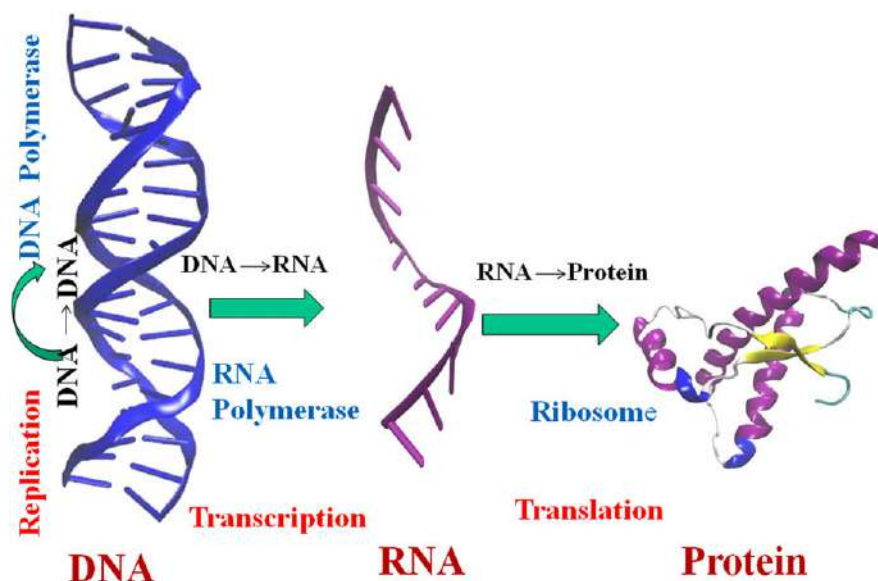


Fig. 1 Schematic representing the Central Dogma of Life.

(Coimbatore Narayanan *et al.*, 2014) and Protein Data Bank (PDB) (Berman *et al.*, 2000) has revealed a wide variety of structures and their conformational adaptability. In addition, a number of different types of non-WC base-pairing with two H-bonds have been observed (Fig. 3) (Lu and Olson, 2008).

DNA and RNA Can Adopt Different Structural Forms

Change in environment play a significant role on twisting, turning and stretching the structure of DNA leading to different forms of DNA (Bansal, 1999); described A-Z forms of DNA, with the exception of the letters F, Q, U, V and Y (Ghosh and Bansal, 2003). Among these helical forms, Watson-Crick model for B-DNA is the most commonly known. There are two other well-known type of helices: A-form and Z-form (Table 1). A-DNA was first observed by Franklin and Gosling in 1953 when they got the X-ray pattern recorded for fibres of the sodium salt of calf thymus DNA under low humidity (Franklin and Gosling, 1953). RNA double helices also have a structure similar to A-DNA (Rich and Davies, 1956). Z-DNA has two nucleotides in a repeating units and a left-handed screw sense. In general, it has alternating purine (Guanine) and pyrimidine (Cytosine) sequences with alternating sugar puckers as well as syn/anti conformations about the glycosyl bond (Wang *et al.*, 1979). Other forms of DNA are described elsewhere (Ghosh and Bansal, 2003).

Like DNA, each RNA strand contains nitrogenous bases covalently bound to a sugar-phosphate backbone. However, RNA is usually single-stranded and the sugar is ribose instead of deoxyribose as in DNA. It also contains Uracil in place of Thymine base. RNA molecules, usually, have a single-stranded structure that can fold over to form different secondary structures viz. hairpin loops, bulge loops, internal loops or multi loops. These loops are stabilized by intra-molecular H-bonds between complementary bases that are important for their functions as well as stability (Holley *et al.*, 1965). In general, RNA secondary structure can have four basic elements namely helices, loops, bulges, and junctions. Stem-loop or hairpin loop is the most common element of RNA secondary structure (Tinoco and Bustamante, 1999) and consists of a stem and a loop. Stem is formed when the RNA chain folds back to form a double helical tract and the unpaired nucleotides form a single stranded loop. Hairpins with loop length of four nucleotides are quite common and also known as tetraloops. Sequences of these tetraloops form three major clusters: UNCG, GNRA, and CUUG (N is one of the four nucleotides and R is a purine) with UNCG being the most stable (Hollyfield *et al.*, 1976). The unpaired nucleotides in one strand of a double helix produce bulge in that strand. However, if the bulge is present in both the strands, it is referred to as an internal loop (Fig. 4). Internal loops can be symmetric or asymmetric depending on the number of nucleotides in each bulge. Two hairpins or internal loops with single stranded RNA often interact with each other to produce pseudoknots, identified for the first time in the turnip yellow mosaic virus (Rietveld *et al.*, 1982). Pseudoknots are found to have catalytic activities (Staple and Butcher, 2005).

In the year 1960, the first experimental demonstration of DNA-RNA hybrid double helix paved the way for understanding the transfer of information from DNA to RNA (Rich, 1960). Currently, there are more than 100 DNA-RNA hybrid and chimera structures available in NDB.

Apart from double helices, DNA and RNA can form triplexes as well as quadruplexes. Triplexes are of two types namely (1) minor groove triplex and (2) major groove triplex. The minor groove triplex is found in almost all large RNAs (Devi *et al.*, 2015). The beet western yellow virus pseudoknot structure has a loop that forms a minor-groove triplex motif with a double

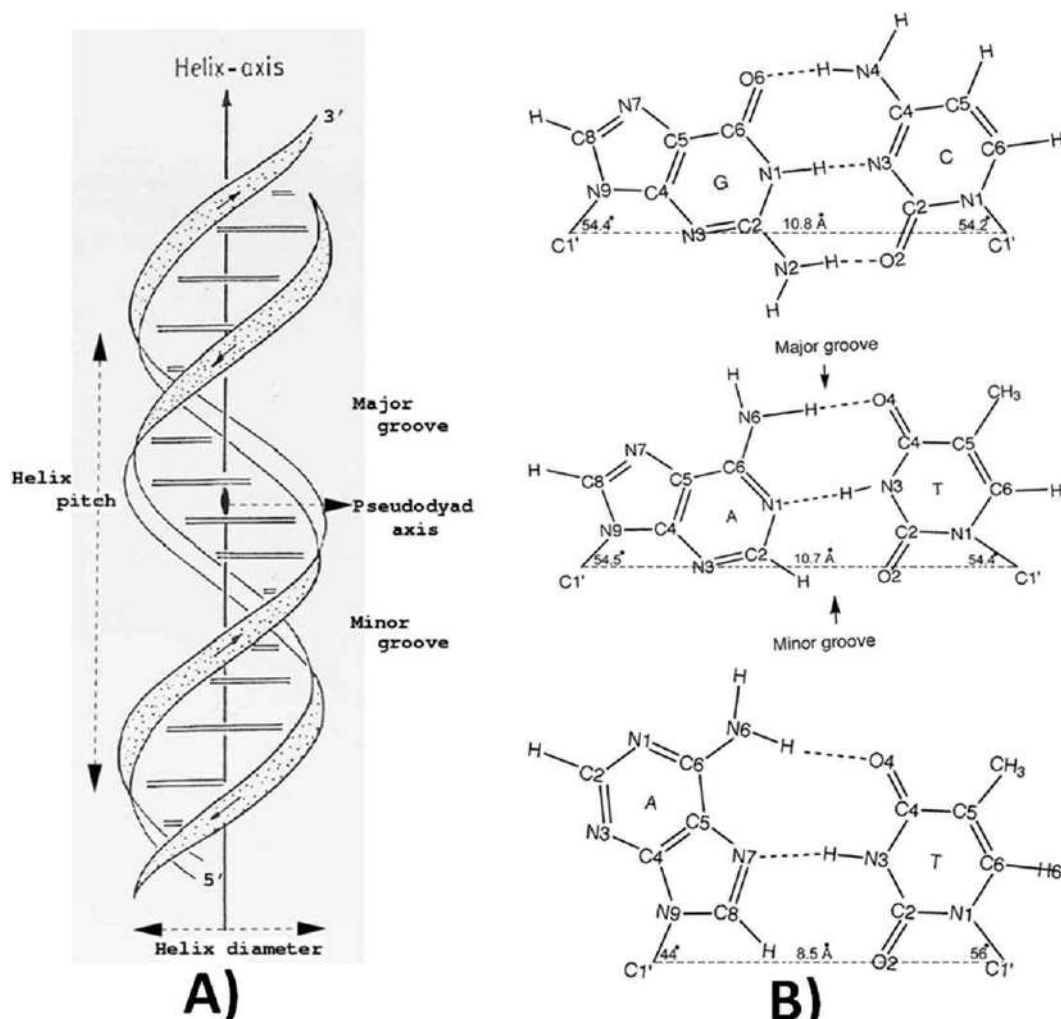


Fig. 2 (A) A schematic diagram of the Watson-Crick double helix. The base pairs are represented by horizontal bars and the sugar-phosphate backbones of the two chains, related by a twofold rotation axis perpendicular to the helix, are represented by ribbons running in opposite directions. The 5' and 3' ends are labelled for the ascending strand. The helix axis, its pitch and diameter and the major and the minor grooves have been identified. (B) Base pairs G · C and A · T with Watson-Crick and A · T with Hoogsteen hydrogen-bond schemes are shown as line diagrams. The base atoms are numbered according to the standard nomenclature and the hydrogen bonds between them are represented by dotted lines. The C1'–C1' distance and the C1'–C1'–N1/N9 angles are indicated in each case. The minor-groove and major-groove sides of the Watson-Crick base pair is indicated in the case of the A · T base pair. Reproduced with permission of the International Union of Crystallography. Available at: <https://journals.iucr.org/services/termsofuse.html>.

helical stem via 2'-OH and triple-base interactions (Su *et al.*, 1999). Minor groove triplexes allow a stable packing of loop and helix that make them critical in the structure of large ribonucleotides including group I intron (Szewczak *et al.*, 1998), group II intron (Boudvillain *et al.*, 2000) and the ribosome. In RNAs, major groove triplexes are not as common as their minor groove counterparts as the major groove in RNA double helices is narrow. However, triplex in B-DNA is possible via the formation of Hoogsteen or reversed Hoogsteen H-bonds in the major groove (Jain *et al.*, 2008).

Quadruplexes are higher order nucleic acid structures that are generally formed by G-rich sequences (Gellert *et al.*, 1962). Hoogsteen type H-bonds allow four Guanine bases to associate and form a square planar structure, a Guanine quartet, that stack on top of each other to give a G-quadruplex. The presence of Alkali ions in the central channel between each pair of tetrads adds to their stability (Largy *et al.*, 2016). Eukaryotic telomeres which are characterized by periodically repeating G-tracts (Sundquist and Klug, 1989; Wright *et al.*, 1997), are associated with quadruplex structures. However, they are also found in non-telomeric genomic DNA. Quadruplexes can be formed from one, two or four separate strands of nucleic acid and depending on the number of strands they can have a wide range of topologies (Burge *et al.*, 2006; Rhodes and Lipps, 2015) (Fig. 5). Depending on the topology and number of constituent strands, there is a variation in the consensus sequence as well (Burge *et al.*, 2006). Quadruplexes can be formed by triplet repeats like GAA, CCG, or CAG (Khateb *et al.*, 2004; Matsugami *et al.*, 2003). A statistical survey of short sequences that can form quadruplex in the human genome have identified few commonly occurring sequences like CCTGTCA, CCTGTT, CCTGTC and CCTGTTA (Todd *et al.*, 2005). It has also been shown

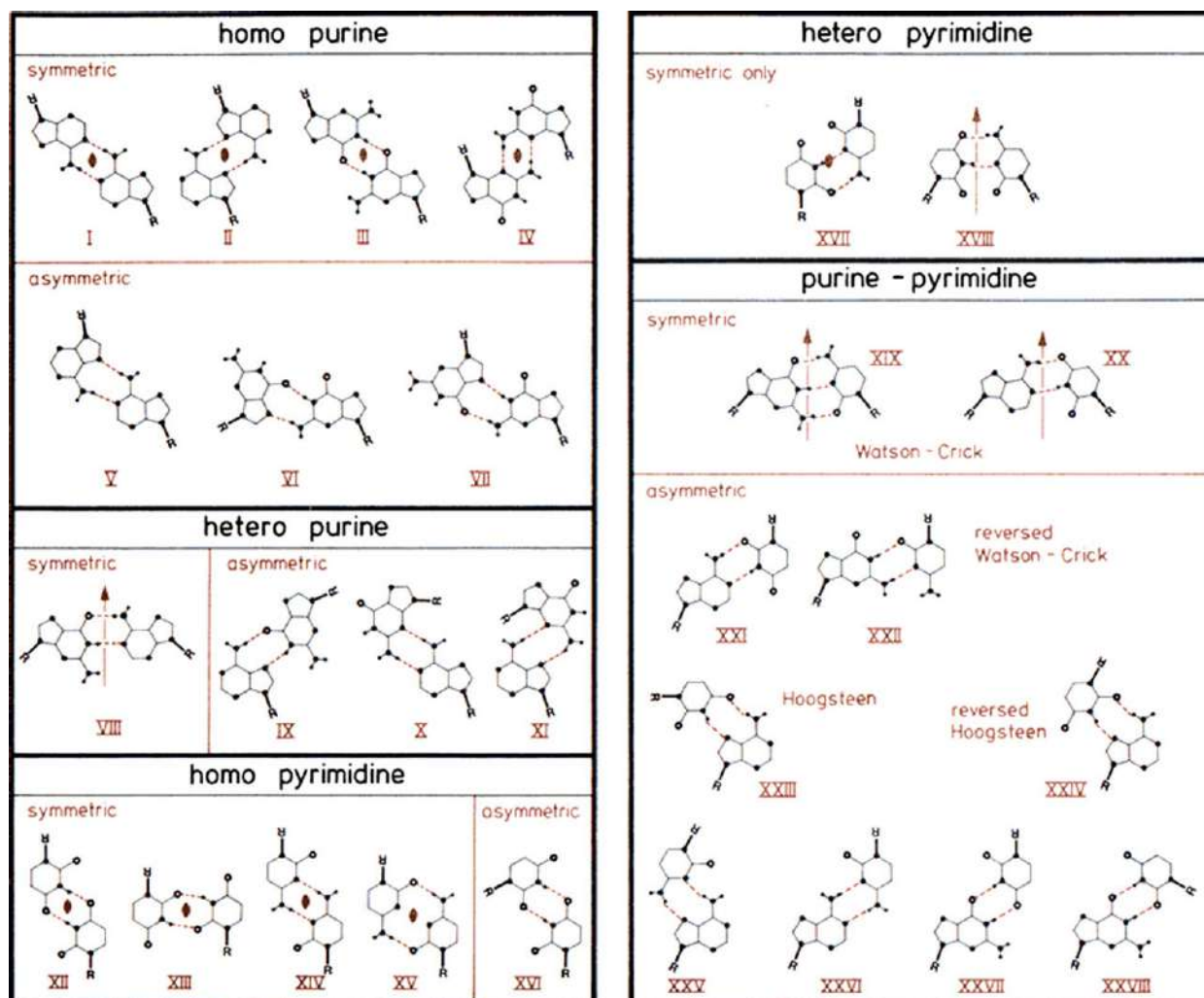


Fig. 3 Possible base pairing schemes between the bases. Adenine, Guanine, Uracil (Thymine), and Cytosine in their cononical (keto- and amino-) tautomeric forms and involving at least two H-bonds. A total of 28 base pairs including the two canonical WC pairs are possible. The reverse A · T/U and G · C WC pairs are asymmetric, and are numbered XXI and XXII respectively. Figure adopted from (with permission). Available at: <http://x3dna.org/highlights/reverse-watson-crick-base-pairs>.

Table 1 Comparison of various parameters of A-, B- and Z-DNA

Parameters	A-DNA	B-DNA	Z-DNA
Helix sense	Right-handed	Right-handed	Left-handed
Repeating unit	1 bp	1 bp	2 bp
Mean twist/bp (°)	+ 32.7	+ 36.0	− 60/2
Mean number of bp/turn	11	10	12
Mean Rise/bp along axis (Å)	2.6	3.4	3.7
Pitch of helix (Å)	28.6	34.0	44.4
Major Groove	Narrow and deep	Wide and deep	Flat
Minor groove	Wide and shallow	Narrow and deep	Narrow and deep
Glycosyl torsion angle	Anti	Anti	Pyrimidine: anti, Purine: syn
Nucleotide phosphate to phosphate distance (Å)	5.9	7.0	Pyrimidine: 5.7, Purine: 6.1
Sugar ring pucker	C3'-endo	C2'-endo	Pyrimidine: C2'-endo, Purine: C3'-endo
Helix Diameter (Å)	23	20	18

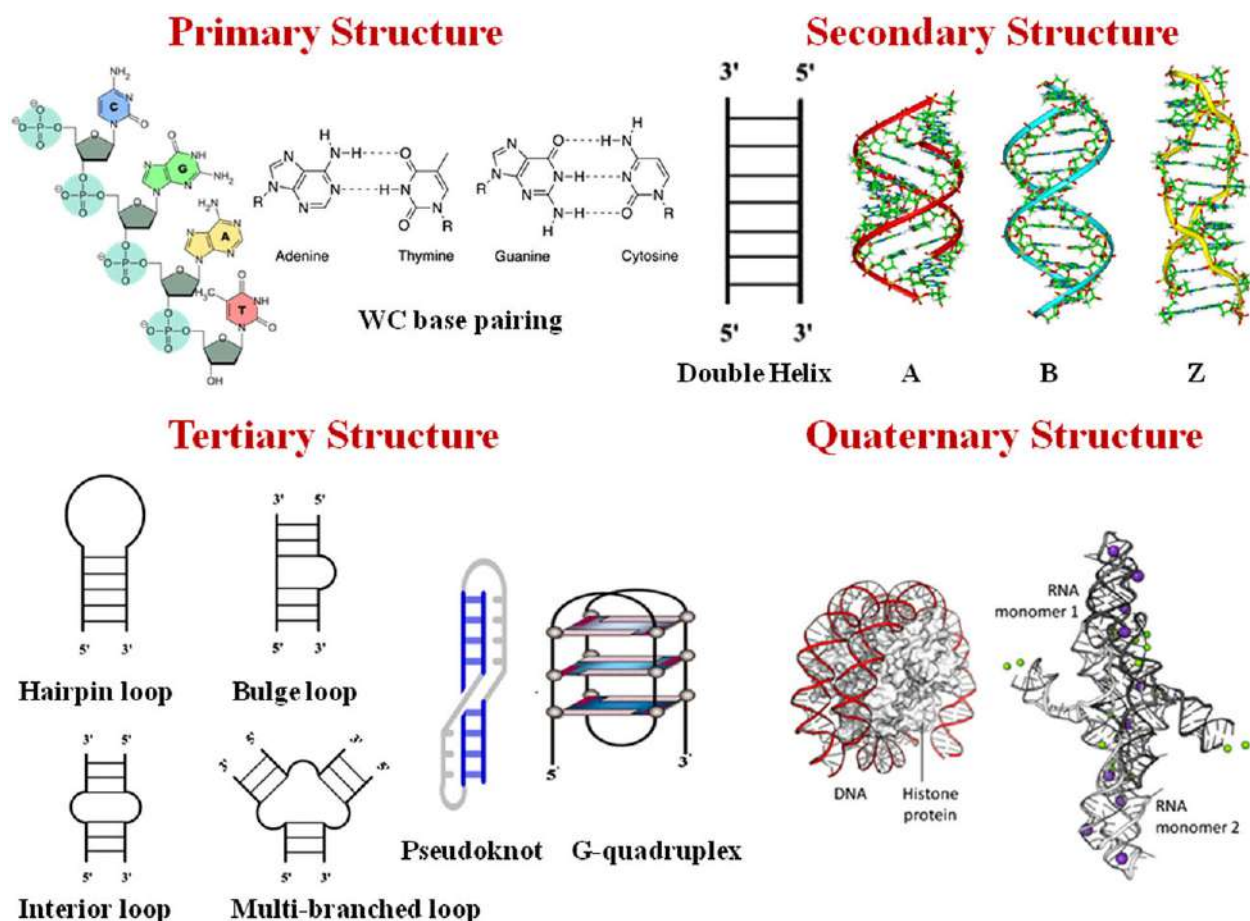


Fig. 4 Pictorial representation of primary, secondary, tertiary and quaternary structure of nucleic acid. PDB ID: 1EQZ and 4R4V are taken as representative examples of quaternary structures.

that the number of nucleotides in the loop, length of G-quartet and cations affect the formation and stability of duplex DNA vs quadruplexes (Cevcec and Plavec, 2005; Risitano and Fox, 2004). In early 2001, quadruplexes were described as “structures in search of a function” (Simonsson, 2001). However, several experiments have now shown their functional importance (Maizels and Gray, 2013; Rhodes and Lipps, 2015). G-quadruplex-forming DNA motif has been reported to be important in predicting the position of DNA replication origins in human cells (Besnard *et al.*, 2012). G-quadruplexes are known to protect the ends of telomere as well as regulate their length (Rice and Skordalakes, 2016). Biological insights along with the bioinformatics analyses of quadruplexes has paved the path for the discovery of drugs targeting them (Fernando *et al.*, 2009; Hänsel-Hertsch *et al.*, 2017; Xu *et al.*, 2017).

Intercalated motifs or i-Motifs are another type of quadruplexes that are formed from Cytosine rich sequences by intercalation and hemi-protonated cytosine–cytosine base-pairing (Gehring *et al.*, 1993). These sequences were found to occur in gene promoter regions and that may allow them to play a pivotal role in gene transcription (Guéron and Leroy, 2000). Hence, targeting them with ligand can provide novel methods for targeting genetic diseases (Day *et al.*, 2014; Sheng *et al.*, 2017). These motifs are not observed in RNA as there will be a repulsion between 2'-OH of ribose sugar (Lacroix *et al.*, 1996). The experimental validation of the formation of i-motifs can be done by experimental methods like NMR studies, X-ray crystallography, UV molecular absorption, molecular fluorescence based techniques, fluorescence resonance energy transfer, fluorescence correlation spectroscopy, time-resolved transient absorption and emission spectroscopy (Benabou *et al.*, 2014). The property of these motifs to reversibly fold or unfold with the change in pH has been exploited in designing nano-machines (Liu and Balasubramanian, 2003; Son *et al.*, 2014; Wang *et al.*, 2010).

Predicting Nucleic Acids Secondary Structures

With the wealth of sequence information available due to the advancements in sequencing technologies, there is a steep increase in secondary structure prediction algorithms (Schroeder, 2009; Shapiro *et al.*, 2007). Realizing the importance of RNA secondary

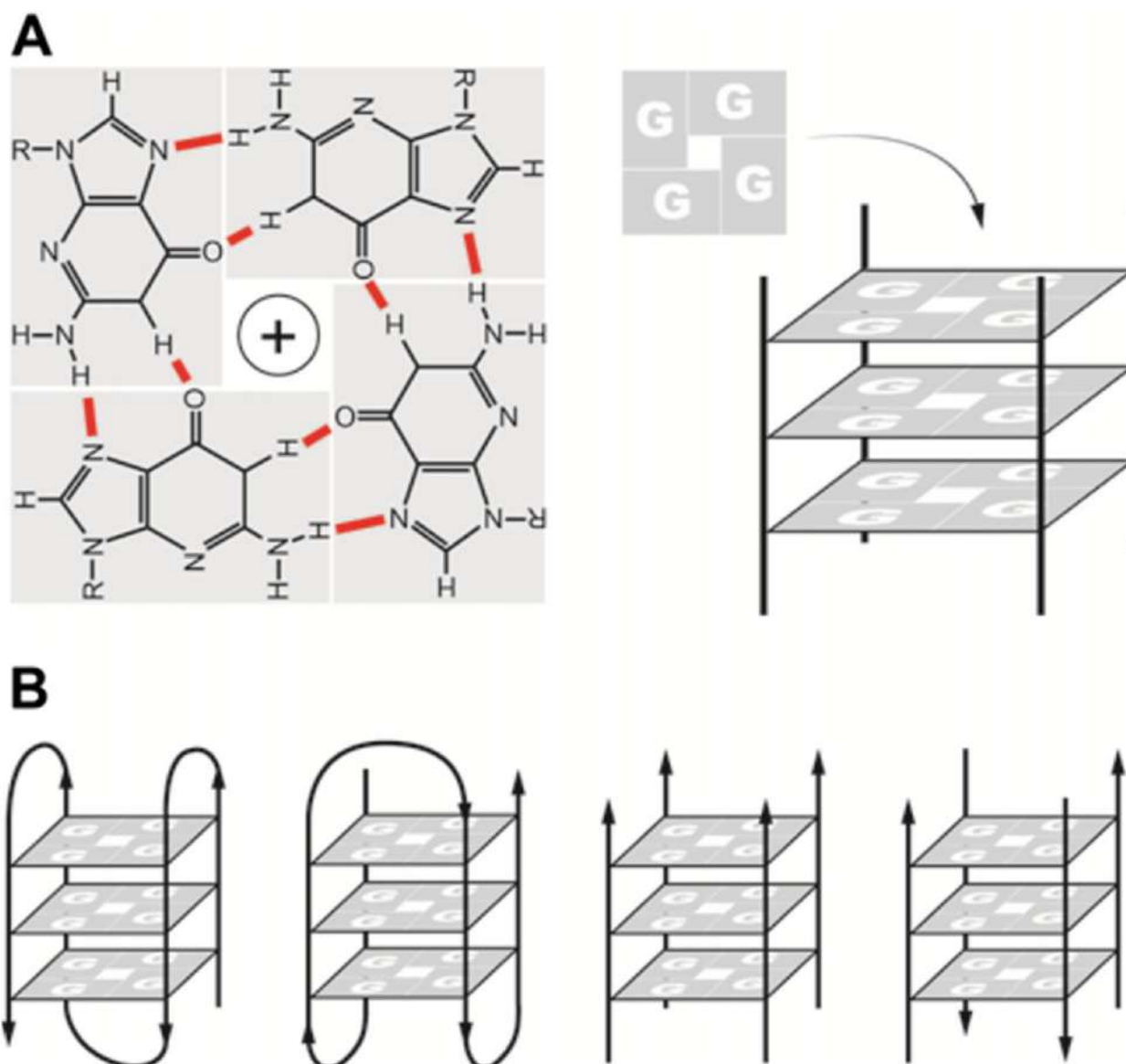


Fig. 5 Structure of G-quadruplexes. G-quadruplexes form in vitro in DNA or RNA sequences containing tracts of three to four guanine. (A) The building blocks of G-quadruplexes are G-quartets that arise from the association of four guanines into a cyclic arrangement stabilized by Hoogsteen hydrogen bonding (N1–N6 and N2–N7). The planar G-quartets stack on top of one another, forming four-stranded helical structures. G-quadruplex formation is driven by monovalent cations such as Na^+ and K^+ . (B) G-quadruplex structures are polymorphic and can be sub-grouped into different families, as for example parallel or antiparallel according to the orientation of the strands and can be inter- or intramolecular folded. Figure taken from Rhodes, D., Lipps, H.J., 2015. G-quadruplexes and their regulatory roles in biology. *Nucleic Acids Research* 43, 8627–8637.

structures, various algorithms and methods have been developed over the years (Schroeder, 2009). Accurate and reliable prediction of RNA secondary structure is a long but persisting challenge and predicting pseudoknots is even more challenging and hence some algorithms do not predict pseudoknots in the input RNA sequence. The various algorithms can be divided into two categories of software: (1) those that do not identify pseudoknots and (2) identify pseudoknots.

Programs for Predicting Secondary Structures Without the Pseudoknots

The *Unified Nucleic Acid Folding* or *UNAFold* (Markham and Zuker, 2008) constitutes a set of algorithms that simulate folding, hybridization and melting pathways for input nucleic acid sequence. *UNAFold* uses the Zuker algorithm (Zuker and Stiegler, 1981) for finding the minimum free-energy structure for a given sequence. Another software package *RNAstructure* can predict as well as analyze secondary structures in input DNA or RNA sequences (Bellaousov *et al.*, 2013; Reuter and Mathews, 2010) with various

base-pairing probabilities. The accuracy of prediction can be improved by incorporating experimental mapping data. The *Vienna RNA Websuite* (Gruber *et al.*, 2008) is a web-platform that provides users an access to various programs for designing and analysis of RNA sequences. It can predict folding of single and aligned sequences as well as RNA-RNA interactions. Using this websuite, it is also possible to design a sequence with a given structure.

Programs for Predicting Secondary Structures With the Pseudoknots

RNA-STAR predicts the secondary structure of linear RNA by simulating a hypothetical process of folding (Abrahams *et al.*, 1990) and uses the concept of local trigger regions for folding. RNA-STAR allows the RNA to fold into pseudoknotted structures during the simulation. Another program PKNOTS (Rivas and Eddy, 1999) generates the optimal minimum energy structure for an input RNA sequence. It uses standard RNA folding thermodynamic parameters along with parameters describing the thermodynamic stability of pseudoknots. However, due to the complexity of the algorithm, it can be computationally expensive for sequences with length > 100 nucleotides. NUPACK (Dirks and Pierce, 2003) is a dynamic programming algorithm that computes the partition function and hence finds the minimum energy structure. It also provides probabilities for various base pairs (bps) in a pseudoknot. The ILM (Ruan *et al.*, 2004) program is an iterative loop matching algorithm that allows the addition or subtraction of bps in different iterations to form pseudoknots and maximize bps. This method can be applied for individual and aligned sequences as it may use both thermodynamic or comparative information. HotKnots (Ren *et al.*, 2005) is a heuristic algorithm that predicts RNA secondary structures including pseudoknots. It tends to find a stable stem by an iterative process and uses a free energy minimization algorithm for pseudoknot free secondary structures that allows identification of potential candidate stems. A comparatively new and updated semi-automated algorithm, ConStruct (Wilm *et al.*, 2008) aligns multiple RNA sequences to find a consensus RNA structure and hence combines both thermodynamic stability and phylogeny information. The interactive graphical interface makes this program more user-friendly. Using a combination of various algorithms, it can predict tertiary interactions as well. A very recent algorithm TurboFold2 (Tan *et al.*, 2017) is an updated version of TurboFold (Harmanci *et al.*, 2011) and predicts secondary structures for multiple RNA homologs. TurboFold 2 has advantage over the TurboFold as it also provides a multiple sequence alignment, incorporating information from the conservation of secondary structures. ProbKnot (Bellaousov and Mathews, 2010) and TurboKnot (Seetin and Mathews, 2012) are other programs for predicting pseudoknots in input RNA sequences.

Prediction of 3D Structure

Prediction of tertiary structure of DNA/RNA is much more difficult with a wide gap between the sequence and structural space. Double helices have three rotational (tilt, roll and twist) and three translational (shift, slide and rise) base paired dinucleotide step parameters. These six parameters along with intra bp parameters (buckle, propeller, opening, shear, stretch and stagger) and local helical parameters (inclination, tip, displacement and rise) can be exploited to build a nucleic acid model. Programs like NUCGEN (Bansal *et al.*, 1995), 3DNA (Lu and Olson, 2008), Curves + (Blanchet *et al.*, 2011) and RNAHelix (Bhattacharyya *et al.*, 2017) use these parameters to build helical models with different sequences, but uniform or varying local step parameters. The 3D structures can be analyzed using various freely available algorithms like NUPARM (Bansal *et al.*, 1995), 3DNA (Lu and Olson, 2008), Curves + (Blanchet *et al.*, 2011) and DSSR (Lu *et al.*, 2015).

An early attempt at predicting RNA tertiary structure by Levitt in the 1969 has seen considerable progress with various algorithms being now available. ROSETTA (Das and Baker, 2008), MC-Sym (Parisien and Major, 2008), GARN2 (Boudard *et al.*, 2017) and F-RAG (Jain and Schlick, 2017) are few examples. ROSETTA uses the assembly of short fragments, taken from existing RNA crystal structures whose corresponding sequences match with a part of the target RNA. The generated models can be further refined to look more realistic. MC-Sym along with MC-Fold increases the accuracy of prediction of RNA 3D structure by using nucleotide cyclic motifs. GARN2 samples 3D RNA structure at a coarse-grained model and takes the corresponding secondary structure as input. Among all possible structures, it extracts two best possible structures that are close to the native one. F-RAG is a graph-based fragment assembly method that generates atomic coordinates from coarse-grained RNA models.

The second component of central dogma includes a process of decoding protein sequence from RNA and the process is called translation. Apart from DNA and RNA, proteins are the other major macromolecules that are important for the sustenance of life.

Proteins—Building Blocks of Life

The term 'Proteins' was first coined by Gerardus Johannes Mulder in his article titled 'On the composition of some animal substances' (Mulder, 1839), where he proposed that animals rely on plants for most of their protein. Since then, proteins have been the subject of extensive studies. Proteins are polypeptides comprising of 20 standard L-amino acids and their derivatives. These amino acids are covalently bonded through peptide bonds formed between the carboxyl and amide group of adjacent amino acid residues. Proteins are essential for survival and normal functioning of living cells and viruses. Almost all biological processes in a cell or group of cells involve proteins.

Proteins can be classified into different categories using numerous criteria such as biological roles, chemical composition and structure. Based on their biological roles, proteins are categorized into contractile, defense, enzymes, nutrients/storage, regulatory, structural, transport and other functional proteins. Using their compositions, proteins can be classified into two categories namely (1) simple and (2) complex. Simple proteins consist only of amino acids. Whereas, complex proteins contain prosthetic groups along with the amino acids in their structure. Based on their shape, proteins can be divided into two sets: fibrous and globular. Fibrous proteins are insoluble in water and have primarily mechanical and structural functions that provide support to the cells as well as the whole organism. The presence of hydrophobic amino acids on their surface facilitates their packaging into very complex supramolecular structures. Fibroin, collagen, α -keratins and elastin are few examples of this class. On the other hand, globular proteins are generally soluble in water or aqueous media containing acids, bases, salts or alcohol and more complex than the fibrous proteins. They have a compact and spherical/ovoid shape. The concept of tertiary and quaternary structures is usually associated with globular proteins only. The complex confirmation of globular proteins gives rise to a large variety of biological functions and makes them more dynamic rather than static in their activities. Defining the functional roles of a protein is essential considering the widespread influence every protein has towards the survival of an organism.

The first step towards deciphering the function of a protein involves understanding the physico-chemical properties of constituent amino acid residues containing an amide group, carboxylic group and a side chain 'R' (except glycine where hydrogen is present instead of R). There are 20 standard amino acids, which serve as building blocks of proteins. Amino acids can be categorized into 5 subgroups using the polarity and charge of side-chains (Nelson *et al.*, 2008): (1) Nonpolar (aliphatic R group: Ala, Gly, Ile, Leu, Met, Pro and Val), (2) Polar (uncharged R group: Asn, Cys, Gln, Ser, Thr and Tyr), (3) Aromatic (Phe, Trp and Tyr), (4) basic (positively charged R group: Arg, His and Lys) and (5) acidic (negatively charged R group: Asp and Glu). A covalent link ($-C(O)-NH$) between the carboxylic acid group of one and the α -amino group of the other amino acid is known as a peptide bond. It has a partial double bond nature thereby restricting rotation about the bond and conferring planarity (Pauling, 1960). However, considerable freedom for rotation prevails around the $N-C^\alpha$ and $C^\alpha-C$ bonds, giving rise to phi (ϕ) and psi (ψ) torsion angles (dihedral angles) respectively. In principle, (ϕ) and (ψ) can have any value between -180° and $+180^\circ$. However, certain values of (ϕ) and (ψ) are prohibited because they could result in steric clashes between atoms present in the polypeptide backbone as well as amino acid side chains. The polypeptide chain can take up the fully extended conformation with (ϕ, ψ) being $(-180^\circ, +180^\circ)$ resulting into the least steric hindrance. The peptide unit is most commonly characterized by *trans* conformation about the peptide bond in which the amide and carbonyl groups point in opposite directions. On the other hand, in *cis*-conformation, a less favorable conformation, both the amide and carbonyl groups in a peptide plane point in the same direction.

Using simple mechanized calculators, in 1963, G.N. Ramachandran and his coworkers predicted the possible allowed and disallowed combinations of the backbone dihedral angles (ϕ, ψ) by assessing the conformations of the two-linked peptide units for the presence of short contacts between non-bonded atoms in a polypeptide chain (Ramachandran *et al.*, 1963; Ramachandran and Sasisekharan, 1968; Ramakrishnan and Ramachandran, 1965; Sasisekharan, 1962). The map is popularly referred as Ramachandran map (Fig. 6). The predictions were found to be corroborating with the experimentally determined protein structure of myoglobin in 1958 (Kendrew *et al.*, 1958). The Ramachandran plot has repeatedly been reconsidered and redrawn during its first half century of life (Bansal and Srinivasan, 2013; Carugo and Djinovic-Carugo, 2013). However, several differences have been observed among various studies depending on: (1) the amount of data used, (2) data quality and (3) the criteria implemented. In fact, it is important to note that the original map used only theoretical computations when none of

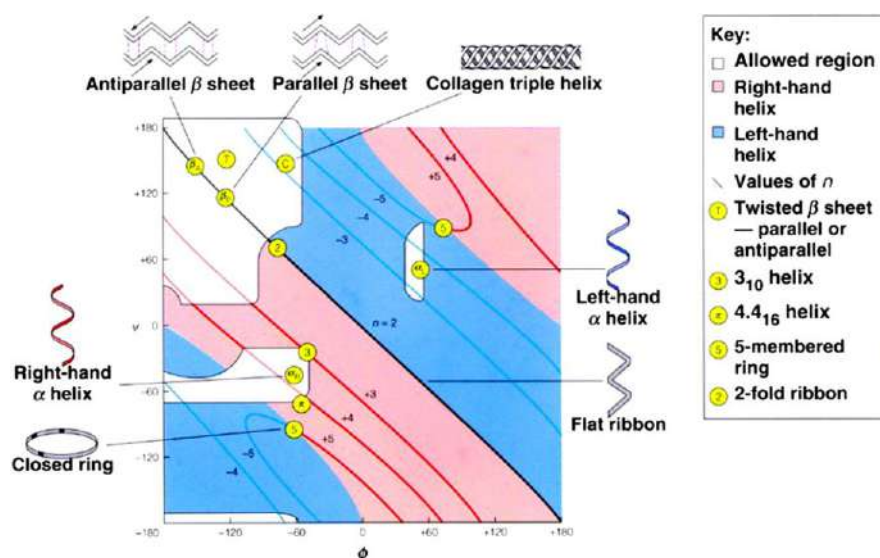


Fig. 6 Ramachandran map representing the sterically favorable regions in white. Various regular secondary structural elements are shown at location corresponding to their Φ and Ψ values. Figure is reproduced from text book Mathews, van Holde, and Ahren, Biochemistry, third edition, 2000.

the protein structures were available in public domain, whereas nowadays, authors make use of experimental observations also. Ramachandran map is used extensively to assess the stereochemical quality of 3D structures of proteins as a part of validation tools like PROCHECK (Laskowski *et al.*, 1993), Moleman2 (Kleywegt and Jones, 1996), and more recently MolProbity (Davis *et al.*, 2004; Williams *et al.*, 2017).

Hierarchical Organization and Classification of Proteins

Proteins are organized in a hierarchical manner and can be illustrated by four major levels defined as primary, secondary, tertiary and quaternary structures (Fig. 7). Such tiered organization of proteins facilitates better understanding and comprehension of their 3D structures. Primary structure can be defined as a linear sequence of constituent amino acids in a polypeptide chain without specifying their spatial arrangements. Secondary structures correspond to recurring arrangements of adjacent amino acids in a polypeptide chain. There are mainly three types of secondary structure elements (SSEs) namely (1) helix, (2) strand and (3) turns (connects two SSEs). Apart from these three regular SSEs, there also exist irregular secondary structures also known as 'random coils'. Higher than the level of secondary structure there exists 'super-secondary structure', a specific combination of secondary structures with explicit spatial arrangements such as α and β hairpins or β - α - β motif (Rao and Rossmann, 1973). SSEs organize themselves in 3D space to form the corresponding tertiary structure that is driven mainly by the non-specific hydrophobic interactions. The quaternary structure specifies the spatial relationships amongst the various polypeptides in a functional protein complex. The extensive growth of protein sequence and structure information has resulted in the creation of numerous classification resources for organizing proteins (Redfern *et al.*, 2005).

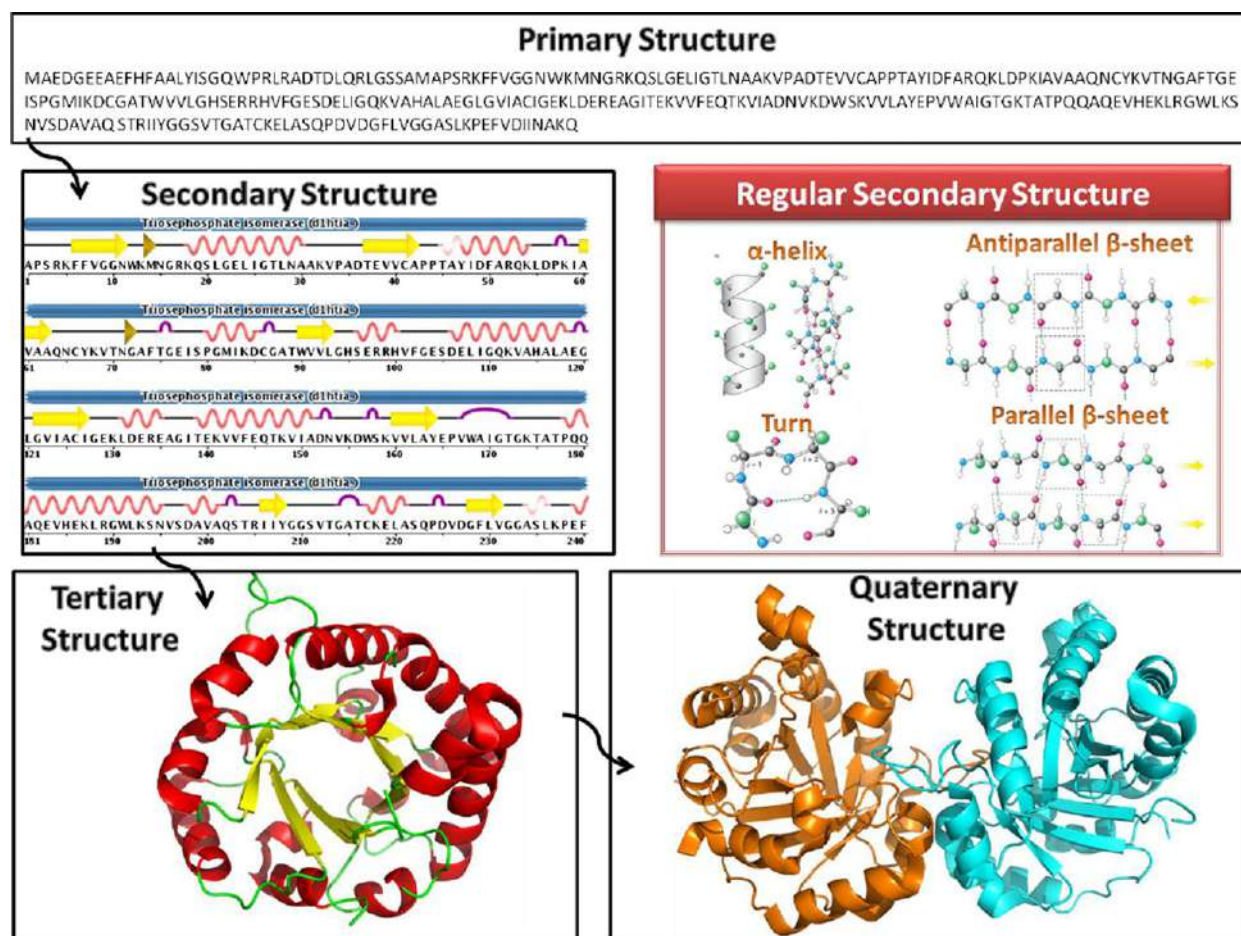


Fig. 7 The hierarchical organization in protein structure. Human triose phosphate isomerase (Uniprot code: P60174, Protein Data Bank (PDB) code: 1HTI) has been taken as an example. The amino acid sequence of a protein is considered as its primary structure. Stretches of amino acids adopt regular secondary structures (such as α -helix, β -sheets – antiparallel and parallel, turns), which are connected by irregular secondary structures, termed as loops. The local secondary structures fold to provide the native and usually functional tertiary structure of a protein. For many proteins, the tertiary structure represents the biologically functional form. For several proteins, assembly of tertiary structures gives rise to the quaternary structure, which is the biologically functional form of that protein.

Based on Structure

Two main structure-based classification databases, Structural Classification of Proteins (SCOP) (Andreeva *et al.*, 2014; Lo Conte *et al.*, 2000; Murzin *et al.*, 1995) and Class Architecture Topology and Homology (CATH) (Orengo *et al.*, 1997, 2003; Pearl *et al.*, 2003) combine sequence, structure and functional information to provide a hierarchical classification of known protein domains in the PDB. Fold classification based on Structure-Structure alignment of Proteins (FSSP) (Holm and Sander, 1996) classifies protein structures based on the pair-wise structural comparison using DALI program (Holm and Sander, 1996).

The focus of SCOP is to structurally characterize the proteins that are deposited in the PDB. SCOP considers evolutionary and structural domains of proteins and classifies them in a hierarchical manner. However, in the recently released version of SCOP, SCOP2 (Andreeva *et al.*, 2014), these relationships form a complex network of nodes representing a relationship of a particular type and is illustrated by a region of protein structure and sequence. Here we are describing the hierarchy of classification in SCOP database (Fig. 8).

- i. Class (secondary structure composition and sequence/arrangement): members of different folds are placed under same class based on the extent of secondary structural content and order of occurrence of SSEs. Domains in globular proteins are usually classified under the following categories of 'Class'.
 - a. All α class: members in this class are predominately composed of helices.
 - b. All β class: members in this class are predominantly composed of β -sheets.
 - c. α/β class: members comprise of interspersed helices and β strands in their structure.
 - d. $\alpha + \beta$ class: members comprise of segregated helices and β strands in their structure.
 - e. Multidomain class: a multidomain class comprises of members with domains of different folds.
 - f. Small proteins: this class includes members corresponding to several disulfide rich and metal binding proteins with few or almost no regular secondary structures.
 - g. Membrane proteins: this class includes membrane proteins.
- ii. Fold (gross structural similarity): members in different superfamilies are grouped into one fold if the arrangement of major SSEs along with their topological connections is the same. Structural similarity among members in the same fold group arises from physicochemical properties favoring certain packing arrangements and chain topologies.
- iii. Superfamily (probable evolutionary relationship): families showing overall structural similarity and in many cases gross functional similarity, thus indicating potential common evolutionary origin, are categorized into one Superfamily. Sometimes although the functions are not the same for the families within the superfamily, the nature of functions along with the topological and chemical equivalence of functional sites imply the potential evolutionary relationship between the families.
- iv. Family (clear evolutionary relationship): family can be defined as a collection of related protein regions, which share high sequence identity and usually good functional and structural similarity. Most of the members of a family show more than 30% sequence identity with each other. However, there exist few examples of families in SCOP containing members with low sequence similarity to the globin family where sequence identity between members could be as low as 15%. However, all

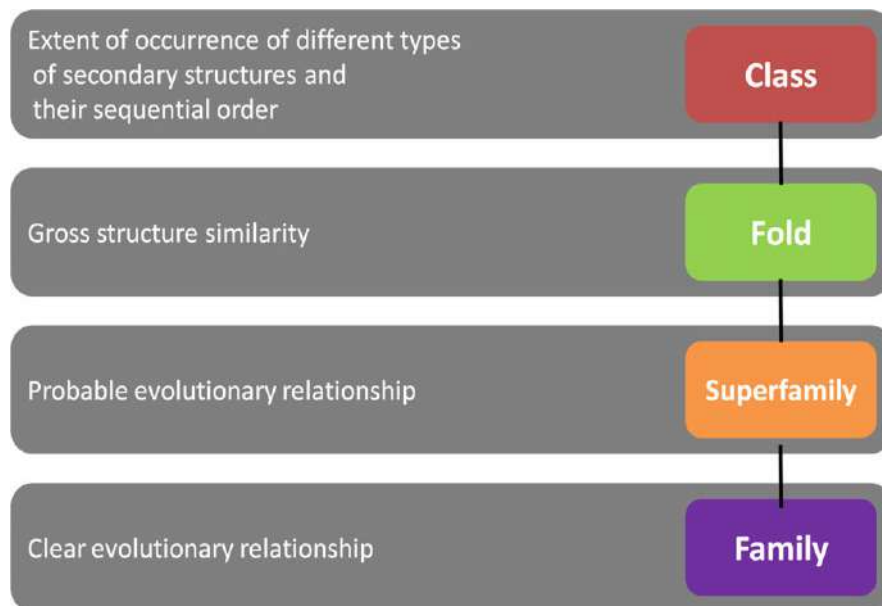


Fig. 8 Levels of classification in the database of Structural Classification of Proteins (SCOP).

members show the same overall structure and critical functional residues in topologically equivalent positions thus implying divergence from a common ancestor.

Based on Sequence Similarity

Protein sequences have been classified into different domain families based on sequence identity among sequences. Different databases which considers sequence identity as a major feature to classify proteins are Protein Family (Pfam) (Punta *et al.*, 2012), Simple Modular Architecture Research Tool (SMART) (Letunic *et al.*, 2015), InterPro (Mitchell *et al.*, 2015) and Protein Domain (ProDom) (Bru *et al.*, 2005). Homology searches in databases of sequence similarity based protein domain families are efficient and their coverage is higher than searches in structural databases as the number of sequences is higher than the number of structures.

Various Secondary Structural Elements in Proteins

The concept of repeating backbone torsion angles ((ϕ, ψ)) or any other geometrical parameters along with the pattern of main chain-main chain (MM) N-H...O H-bonds defining a regular SSE in proteins is well established. The α -helices and extended β -strands, which are major regular SSEs found in proteins, were first predicted from theoretical studies (Pauling and Corey, 1951; Pauling *et al.*, 1951) and subsequently confirmed by X-ray diffraction analysis (Blake *et al.*, 1965; Perutz, 1951). Other SSEs like 3_{10} -helix (Donohue, 1953), π -helix (Low and Grenville-Wells, 1953), PolyProlineII (PPII)-helix (Cowan and McGavin, 1955) and left-handed α , 3_{10} and π -helices were also proposed from model building studies and found to occur occasionally (Ramachandran and Sasisekharan, 1968). Table 2 enlists various parameters of these SSEs. They have found their importance in a number of structural biology applications, such as structure comparison (Gibrat *et al.*, 1996), visualization (Humphrey *et al.*, 1996; Sayle and Milner-White, 1995; Schrodinger, 2010) and classification (Murzin *et al.*, 1995; Orengo *et al.*, 1997). The formation of SSEs plays a major role in protein folding also (Murzin *et al.*, 1995; Orengo *et al.*, 1997).

α -Helices

Of all the postulated helices in proteins, α -helices are the most abundant and the first to be identified (Kendrew *et al.*, 1958; Pauling *et al.*, 1951). These helices have 3.6 residues per turn and 13 atoms in the ring formed by $\text{NH}_{(i+4)} \rightarrow \text{O}_i$ H-bonds (represented as 3.6_{13}) (Kendrew *et al.*, 1958; Pauling *et al.*, 1951). The model successfully explained the overall distribution of intensities in the fiber diffraction patterns for the fibrous proteins, recorded earlier by Astbury in early 1930s and for certain synthetic polypeptides. The helix was established as a familiar motif after the 3D structure of myoglobin using X-ray crystallography was solved (Kendrew *et al.*, 1958, 1960). The backbone dihedral angles ((ϕ, ψ)) of residues in ideal right-handed α -helices are $(-57^\circ, -47^\circ)$ that may vary for the constituent residues of the native α -helices. However, it appears that the sum of both dihedral angles ($(\phi)_i + \psi_{(i+1)}$) is approximately -105° . The amino acid side-chains protrude away from the helix axis and point roughly towards the N-terminus. This property has been used in preliminary, low resolution electron-density maps to determine the direction of the protein backbone (Terwilliger, 2010). The aggregate effect of the individual carbonyl groups (microdipoles) of the constituent residues, pointing along the helix axis, generates an overall dipole moment (Hol *et al.*, 1978). α -helix has attracted many world-renowned artists who have referred it explicitly in their work. The list of names includes Julie Newdoll in painting, Irving Geis in cartoon as well as Julian Voss-Andreae, Bathsheba Grossman, Byron Rubin and Mike Tyka in sculpture (Fig. 9(A–B)). Based on handedness of α -helix, it can be categorized into right-handed (clockwise) or left-handed (anticlockwise). Right-handed α -helices in protein structures outnumber their left-handed counterpart.

β -Strands

Besides α -helices, β -strands/sheets are another major SSE in globular proteins, which contain 20%–28% of all residues (Kabsch and Sander, 1983a,b). β -strands are the basic unit of a β -sheet, which can be considered as a helix with 2 residues/turn and

Table 2 Regular SSEs formed by polypeptide chains and their respective parameters. Plus and minus sign in 2nd column correspond to the right and left handed helices, respectively

Secondary structures	Residues per turn and chirality	Rise per residue ($\text{\AA}$)	Radius of helix ($\text{\AA}$)
α -helix	± 3.6	1.5	2.3
π -helix	$+ 4.3$	1.1	2.8
3_{10} -helix	$+ 3.0$	2.0	1.9
Planar parallel sheet	± 2.0	3.2	1.1
Planar anti-parallel sheet	± 2.0	3.4	0.9
Twisted parallel or anti-parallel sheet	$- 2.3$	3.3	1.0

sheet. Two interacting strands of a sheet can be arranged either in (1) anti-parallel or (2) parallel to each other (**Fig. 9(C)** and **(D)**). Antiparallel strands are commonly found in sheets. However, about 20% of the β -sheets are observed to have both parallel and antiparallel strands (**Richardson, 1977**). Although parallel β -strands usually form large, moderately twisted sheets, occasionally it forms a cylinder with helices being present at outside as in triosephosphate isomerase structure. On the other hand, large antiparallel sheets, usually roll up either partially (first domain of thermolysin or in ribonuclease) or entirely to join edges facilitating the formation of a cylinder or barrel (membrane-spanning proteins such as porins). The connection between the two strands of a sheet can be divided into two categories namely (1) hairpin connection and (2) crossover connection (**Richardson, 1981; Sternberg and Thornton, 1976, 1977a,b**). In hairpin connection, the backbone enters and exits from the same end of the sheet, while the backbone enters and exits from the opposite end in crossover connection. Though right-handed crossovers are the rule in globular proteins, quite a few examples of left-handed crossovers are also observed (e.g., subtilisin and glucose phosphate isomerase).

3₁₀-Helices

Besides the classical α -helices and β -sheets, the only other principal helical species, which occurs to any great extent in globular protein structure, is the 3₁₀-helix with a three-residue repeat and a H-bond between NH group of residues 'i + 3' instead of 'i + 4' and CO group of ith residue (**Fig. 9(B)**). This structure was first proposed by **Taylor (1941)** and later discussed in detail by **Huggins (1943)**, **Bragg et al. (1950)** and **Donohue (1953)**. Its backbone conformational angles are approximately $(\varphi) = -49^\circ$, $(\psi) = -26^\circ$ (**Ramachandran et al., 1966**), which lie within the same energy minimum as of the α -helix. However, for a long periodic structure, the 3₁₀-helix is considerably less favorable than the α -helix in terms of both local conformational energy and geometry of H-bonds. The same is evident by the less frequent occurrence of longer 3₁₀-helices (**Kumar and Bansal, 2015a,b**). Though long 3₁₀-helices are very rare, shorter 3₁₀-helices occur quite frequently (**Toniolo and Benedetti, 1991**). A survey of 57 crystal structures, showed that 3.4% of total residues are involved in 3₁₀-helix (**Barlow and Thornton, 1988**). A total of 132, 3₁₀-helices with six amino acids or more were observed by **Pal and Basu (1999)**. Another structural analysis of 1774 3₁₀-helices spanning five residues or more from highly resolved (resolution better than 1.6 Å) 689 protein chains with sequence identity being $\leq 20\%$ revealed that they occur quite frequently, but the longer helices are irregular (**Enkhbayar et al., 2006**). They are often found at the termini of α -helices. Structures of three potassium channels of the six transmembrane (TM) helix type and solved by x-ray crystallography (**Clayton et al., 2008; Long et al., 2005a,b**) revealed that the fourth TM segment of each subunit adopts a 3₁₀-helical conformation spanning 7–11 residues.

π -Helices

The discovery of α -helices triggered a race for finding various other possible helices. Donohue predicted the occurrence of 2.2₇, 3₁₀, 4.3₁₄ and 4.4₁₆ helices (**Donohue, 1953**), whereas Low and Baybutt independently suggested the possibility of the 4.4₁₆ or π -helix (**Low and Baybutt, 1952**). The π -helices are often characterized by the presence of a repeating pattern of H-bond between the backbone C=O of residue 'i' and the backbone HN of residue 'i + 5' (**Fig. 9(B)**). π -helices have, therefore, been described as α -aneurisms (**Keefe et al., 1993**), α -bulges (**Cartailler and Luecke, 2004**) or π -bulges (**Hardy et al., 2000**). The sum of dihedral angles $((\varphi)_i + \psi_{(i+1)})$ is approximately -125° , whereas the angle τ (N-CA-C') is larger (114.9°) than the standard tetrahedral angle of 109.5° . The side chains in π -helices are more staggered than the ideal 3₁₀-helix, but not as well in the α -helix. Like 3₁₀-helices, a majority of π -helices are found in conjunction with the α -helices. The large majority of naturally occurring π -helices consist of five residue segments with the minimal two π -type (NH_(i+5) → O_i) H-bonds (**Fodje and Al-Karadaghi, 2002**). The functional roles of π -helices have featured in various reports and the correlated presence of such helices with the active sites in proteins (**Medlock et al., 2007; Weaver, 2000**).

PolyProlineII-helices

The PPII-helix is an extended, flexible left-handed helix without intra-helical H-bonds (**Fig. 9(E)**). The backbone dihedral angles of constituent residues are restricted to around $((\varphi) = -75^\circ$, $(\psi) = 150^\circ$). They play an important role in structural proteins, unfolded states and as ligands for signaling proteins. The analysis of main chain conformations in proteins have shown that the left-handed PPII-helices are quite common in globular proteins (**Adzhubei and Sternberg, 1993; Kumar and Bansal, 2016**). Another survey of 274 non-homologous protein structures revealed that longer PPII-helices are rare, but at least one PPII-helix is present in the majority of proteins (**Stapley and Creamer, 1999**). Most PPII-helices are shorter than five residues. Though PPII-helices have a high preference for Pro residues, there is evidence for the presence of PPII conformations in helices containing non-Pro residues (**Makarov et al., 1975; Tiffany and Krimm, 1968; Woody, 1992**). A majority of the PPII-helices are solvent exposed and found to be frequently involved in protein-protein interactions (**Berisio and Vitagliano, 2012**). PPII-helix is observed to have a direct involvement in host-pathogen recognition for virus infections (**Vermeire et al., 2011**). These helices are also found to activate the phosphatidylinositol 3-kinase (PI3K)/Akt signaling pathway by the influenza-A virus (**Hale et al., 2010**). Recent advancements in identifying PPII-helices are well explained by **Narwani et al. (2017)**.

Collagen Triple Helix

Collagen is a fibrous protein which is present in abundance in the human body, being a major constituent of skin, bones as well as various connective tissues. It helps in forming a scaffold to provide strength and structure. Collagens have a unique tripeptide repeat sequence with Glycine at every third position and the iminoacids Proline and Hydroxyproline often being present at the other two positions (Bowes and Kenten, 1948; Eastoe, 1955). The first triple helical structure of collagen had two inter chain H-bonds for every three residues (Ramachandran and Kartha, 1954, 1955). This was slightly modified to a one hydrogen-bonded structure (Rich and Crick, 1955). However a second water mediated hydrogen bond is also found to stabilize the triple helix (Ramachandran and Sasisekharan, 1961). Each helix in collagen has left handed screw sense and the major helix formed by three such helices is right handed. Each helix has 10 residues in 3 turns resulting in a pitch of 85.8 Å. The twist between two neighboring helices was found to be -108.8° and rise of ~ 2.86 Å (Bhattacharjee and Bansal, 2005; Ramachandran and Kartha, 1955; Ramachandran and Sasisekharan, 1961). Molecular defects in biosynthesis of collagen can lead to many rare genetic diseases involving connective tissues like Ehlers-Danlos syndrome (De Paepe, 1998; Gaisl *et al.*, 2017; Miyake *et al.*, 2017; Prockop *et al.*, 1979). Collagen has been shown to have wide range of applications in the medical and cosmetic field as it is biodegradable, biocompatible, easily available and weak antigenic (Cheng *et al.*, 2017; Chvapil, 1977; Chvapil *et al.*, 1973; Lee *et al.*, 2001; Sheikh *et al.*, 2017). In the field of tissue engineering, collagen based biomaterials are used to improve tissue functions (Parenteau-Bareil *et al.*, 2010).

Turns

A turn is a region of the protein, which consists of four consecutive amino acids and allows the polypeptide chain in folding back on itself by nearly 180° (Fig. 9(F-G)). Venkatachalam (1968) explored all the plausible available conformations to a system of three linked peptide units consisting of four successive residues that can be stabilized by a backbone H-bond between the CO of residue 'i' and the NH of residue 'i + 3'. The resultant non-repetitive SSE, recognized from the theoretical conformational analysis, were termed as 'Turns'. He proposed three unique conformations based on $((\phi, \psi))$ values (Turn I, II and III) along with their mirror images having $((\phi, \psi))$ signs reversed (I', II' and III'). Lewis *et al.* (1973) analyzed the turns in a large dataset of 3D protein structures and proposed a more general definition using the distance (d) between the $C^\alpha(i)$ and the $C^\alpha(i + 3)$. The distance 'd' should be < 7 Å to be a part of turn. Kuntz (1972) and Chou and Fasman (1977) have found 45% and 32% of protein chain in turns respectively. At the same time, considering only the central dipeptide, Zimmerman and Scheraga found 24% of the non-helical residues in turns (Zimmerman and Scheraga, 1977). Turns are also known as reverse turns, β -turns, β -bends, hairpin bends, 3_{10} -bends, kinks or widgets, among others. Helices and strands are often connected to each other by the tight turns. It has also been suggested that turns can direct the process of protein folding to the native conformation as they are best suited to provide the decisive long-range interactions that form tertiary structure (Lewis *et al.*, 1971; Rose *et al.*, 1976). Few proteins have been reported to have structure, which is heavily dependent on turns. For example, high-potential iron protein consists of 17 turns in 85 residues (Carter *et al.*, 1974). Type-I and Type-II along with their mirror images I' and II' are most abundant types of turns.

Other SSEs

This class includes rarely occurring helices like 2.2_7 -helices or left handed α , 3_{10} or π -helices. The repetitive C7 structure, the 2.2_7 helix or 2.2_7 ribbon, was first proposed for polypeptides by Donohue (1953) and later it was considered theoretically with $((\phi) = -78.1^\circ, \psi = 59.2^\circ)$ (Ramachandran and Sasisekharan, 1968) for further studies. The 2.2_7 -helix is a tight helix of 2.2 amino acids per turn and a 7 atom loop, closed by the H-bond between $NH_{(i+2)}$ and O_i . So far, this structure has not been reported in protein crystal structure.

The analysis of 31 verified left-handed helices from a set of 7284 proteins suggests that despite being rarely found in proteins, when they occur, they have structural or functional significance (Novotny and Kleywegt, 2005). Backbone dihedral angles (ϕ) and ψ of residues in left-handed helices are found to range from 30° to 130° and -50° to 100° respectively, a mirror image of the residues in right-handed helices.

Computational Approaches for Identification of Secondary Structures in Proteins

SSEs define a protein motif and under physiological conditions almost all protein sequences have at least one 3D structure that determines their biological function. In this section, the computational methods developed for prediction/assignment of different SSEs from input sequence/protein structures are discussed.

Prediction of Secondary Structures Using Protein Sequences

One can find the root of protein secondary structure prediction in 1951, when the models for helix and sheet were proposed by Pauling and Corey (1951) and Pauling *et al.* (1951). Prediction of SSEs in bioinformatics aims to predict the local secondary structures of proteins based only on knowledge of their amino acid sequence. The prediction consists of classifying regions of the

amino acid sequence into helices, β -strands or turns. Interest in developing the methods for predicting the secondary structures in protein started as soon as the first crystal structure was solved. These methods (Guzzo, 1965; Kotelchuck and Scheraga, 1969; Lewis *et al.*, 1970; Prothero, 1966; Schiffer and Edmundson, 1967) focused mainly on identifying regions, which are most likely to take α -helix conformation. With the increase in the number of solved protein structures, significantly improved algorithms were developed in 1970s. However, these methods attained the accuracy of 60%–65% and often under-predicted the strands (Mount, 2004). More than 20 different SSEs prediction methods have been reported till date. Some of them are shown in Fig. 10(A).

The first major breakthrough came with the development of Chou-Fasman method (Chou and Fasman, 1974), which relies predominantly on the probability parameters determined from relative frequencies of appearance of different amino acids in each

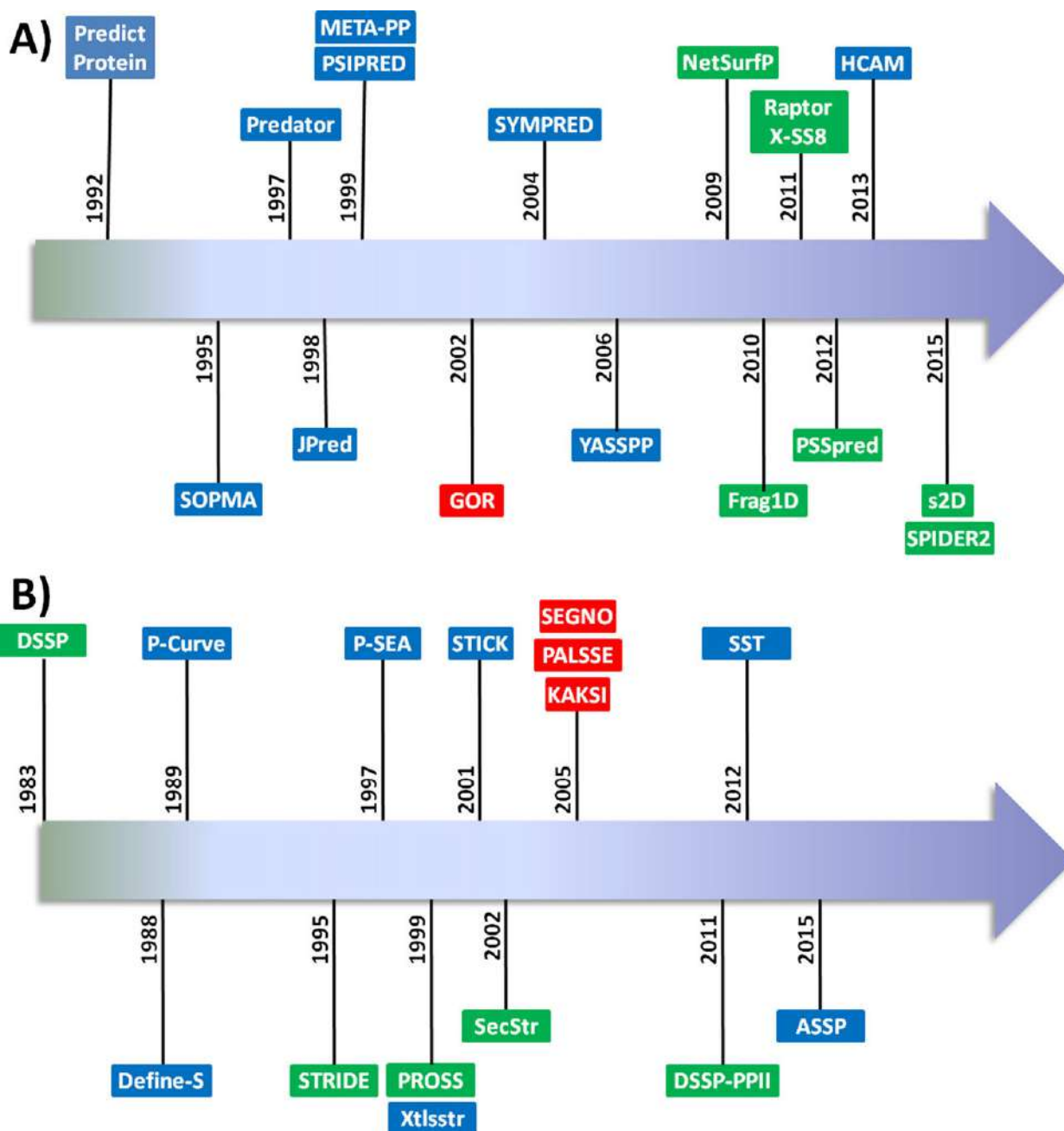


Fig. 10 (A) Timeline of protein secondary structure prediction algorithms. Boxes colored in blue indicate that the algorithms are available as web-server only, while those in green are available as standalone. GOR method, colored in red is available as a web-server as well as standalone program; (B) Timeline of protein secondary structure assignment algorithms. Boxes colored in blue suggest that the algorithm uses 3D geometry, while programs in green colored boxes use (ϕ, ψ) and/or H-bond patterns for SSE assignment. Algorithms in red colored box are hybrid methods.

type of major SSEs. This method is almost 50%–60% accurate in predicting SSEs (Kabsch and Sander, 1983a,b) that is significantly less than the modern machine learning-based techniques (Mount, 2004).

The GOR method (Garnier *et al.*, 1978), named after three scientists Garnier, Osguthorpe and Robson, is an information theory-based method and Bayesian in nature. The GOR method considers the probability of each amino acid having a secondary structure as well as the conditional probability of the amino acid in each structure given that its immediate neighbors have already formed that structure. The original GOR method predicted the secondary structures with roughly 65% accuracy. The original method is more successful in predicting α -helices than β -strands (Mount, 2004).

Methods like PSIPRED (Jones, 1999), SPINE (Dor and Zhou, 2007) and JPRED (Drozdetskiy *et al.*, 2015) are based on neural networks and predict the secondary structures with over 70% accuracy. SPINE-X (Faraggi *et al.*, 2009) algorithm can accurately predict the dihedral angles of the residues and hence improved the *ab-initio* structure prediction of proteins.

Apart from using only amino acid sequence, considering various other factors like effect of local environment (Zhong and Johnson, 1992), solvent accessibility of residues (Macdonald and Johnson, 2001) and protein structural class (Costantini *et al.*, 2006) can improve the SSE prediction as they also affect the SSEs in proteins (Adamczak *et al.*, 2005; Costantini *et al.*, 2007; Momen-Roknabadi *et al.*, 2008). Though the recent improvement produces a better prediction of SSEs and in particular β -strands, still the constraints to the prediction has not reached and continue to rise (Rost, 2001; Yang *et al.*, 2016).

Assigning SSEs to 3D Structures

Knowing the importance of secondary structures as well as the increase in the number of experimentally solved 3D structures, several methods have been proposed over the years to identify the SSEs from the given 3D structure of proteins. Secondary structures possess regularities in various geometric parameters like C^α distances, dihedral angles and specific patterns of H-bonds that can be utilized as criteria to define them. In general, most of the methods can correctly identify the location of the core of helices and strands in proteins. However, precise assignment of termini is still a problem as they are very often ill-defined and difficult to determine unambiguously. These methods can be broadly classified into three categories: (1) algorithms based on $((\varphi), \psi)$ and/ or H-bond patterns (2) algorithms based on 3D geometry and (3) hybrid methods, which use both (1) and (2). Programs like DSSP (Kabsch and Sander, 1983a,b; Touw *et al.*, 2015), STRIDE (Frishman and Argos, 1995) and PROSS (Srinivasan and Rose, 1999) fall into the first category, DEFINE (Richards and Kundrot, 1988), P-CURVE (Sklénar *et al.*, 1989), P-SEA (Labesse *et al.*, 1997), SST (Konagurthu *et al.*, 2012) and ASSP (Kumar and Bansal, 2015a,b) come under second category, whereas KAKSI (Martin *et al.*, 2005) and PALSSE (Majumdar *et al.*, 2005) fall under the third category. A few programs that specifically identify π (Fodje and Al-Karadaghi, 2002) and PPII (Cubellis *et al.*, 2005; King and Johnson, 1999; Mansiaux *et al.*, 2011; Srinivasan and Rose, 1999) -helices have also been developed.

The first ever automated method for SSE assignment was developed by Levitt and Greer (1977) who used distance and virtual torsion angle made by C^α atoms over a sliding window of four residues. Breakthrough in assigning SSEs came with the introduction of more comprehensive and widely used algorithm known as 'Dictionary of Secondary Structure of Proteins' (DSSP) (Kabsch and Sander, 1983a,b; Touw *et al.*, 2015) that is based on the detection of H-bond patterns defined by an electrostatic criterion. DSSP is considered as the gold standard for SSE assignment and used in number of software packages like Rasmol (Sayle and Milner-White, 1995) and GROMACS analysis tools (Berendsen *et al.*, 1995). DEFINE-S (Richards and Kundrot, 1988) uses only C^α coordinates, compares their distances with the distances in ideal SSEs and also provides information about the super-secondary structures. P-CURVE (Sklénar *et al.*, 1989) assigns SSEs based on the helicoidal parameters and global peptide axis for peptide units. Another widely used algorithm known as STRIDE (Frishman and Argos, 1995) uses $((\varphi), \psi)$ along with H-bond pattern. STRIDE has been implemented in a visualization tool VMD (Humphrey *et al.*, 1996) for assigning SSEs. P-SEA (Labesse *et al.*, 1997) uses a short C^α distance mask and two C^α dihedral angles to assigns SSEs, while PROSS (Srinivasan and Rose, 1999) is solely based on backbone dihedral angles. Another algorithm, Xtlstr (King and Johnson, 1999) calculates backbone dihedral angles as well as distances and assigns SSEs that would be consistent with interactions of amide-amide groups observed from circular dichroism of a protein in the ultraviolet range. SECSTR (Fodje and Al-Karadaghi, 2002) is more sensitive to the π -helices. Using SECSTR, for the first time, authors reported the biasness of DSSP and STRIDE towards the α -helices. However the latest version of DSSP (Touw *et al.*, 2015) has addressed this problem and tried resolving it (Kumar and Bansal, 2015a,b). The algorithm PALSSE (Majumdar *et al.*, 2005) mainly uses distance and torsion angle constraints to identify core elements and later extends them to longer segments. Authors claim to assign SSEs up to 80% of the protein structure. KAKSI (Martin *et al.*, 2005) uses C^α distances and backbone dihedral angles to show the concordance with the assignments found in the PDB (Berman *et al.*, 2000) files. Another algorithm SST (Konagurthu *et al.*, 2012) uses minimum message length inference for the assignment of SSEs in protein structures. A comparatively new method 'Assignment of Secondary Structure in Proteins' (ASSP) uses only the path traversed by the C^α atoms of the consecutive residues (Kumar and Bansal, 2015a,b) and is an extension of HELANAL-Plus (Bansal *et al.*, 2000; Kumar and Bansal, 2012), a program for analysis of geometry of helices in proteins. The algorithm is based on the premise that a protein structure can be divided into uniform stretches that can be defined in terms of helical parameters and depending on their values, the stretches can be further classified into different SSEs, viz. α , 3_{10} , π , extended β -strands, PPII and other left-handed helices. Another, recently reported algorithm (Cao *et al.*, 2015) identifies α -helices along with 3_{10} and π -helices by dividing it into a minimization problem and a restraint satisfaction problem. It follows rigorously the geometry of helices. Brief description about various algorithms is tabulated in Table 3 and Fig. 10(B).

Table 3 Brief description of different secondary structure assignment algorithms

Sl. No.	Algorithm	Description	Reference
Category (i)			
1	DSSP	Detects the H-bond patterns using bond energy criterion	Kabsch and Sander (1983a,b)
2	STRIDE	Uses $((\varphi), \psi)$ along with H-bond pattern	Frishman and Argos (1995)
3	PROSS*	Uses only on the backbone dihedral angles $((\varphi), \psi)$	Srinivasan and Rose (1999)
4	SECSTR	Uses DSSP like H-bond definition and was developed to identify and analyze π -helices	Fodje and Al-Karadaghi (2002)
5	DSSP-PPII*	Identifies the PPII-helices in the region not assigned as a major SSE by DSSP and gives the output in the DSSP format	Mansiaux <i>et al.</i> (2011)
Category (ii)			
6	Levitt <i>et al.</i>	Uses distance and virtual torsion angle made by the C α atoms over a sliding window of four residues	Levitt and Greer (1977)
7	DEFINE-S	Uses only C α coordinates and compares the distance between various C α 's with the distances in ideal SSEs	Richards and Kundrot (1988)
8	P-CURVE	To start with, it chooses the successive repeating unit and does the analysis of mathematical analysis of protein curvature	Sklenar <i>et al.</i> (1989)
9	P-SEA	Solely based on the C α atoms. Uses three distance, one angle and one dihedral angle	Labesse <i>et al.</i> (1997)
10	XTLSSTR*	Calculates two angles and three distances for assigning SSEs. The algorithm is driven by the concept of circular dichroism (CD) of a protein in the far ultraviolet range.	King and Johnson (1999)
11	STICK	Finds a set of best fit axes and later takes the average rise of the residues along each axis	Taylor (2001)
12	SST	Uses minimum message length inference for SSEs assignment	Konagurthu <i>et al.</i> (2012)
13	ASSP*	Use C α atoms to identify the continuous stretches and later divides them into different SSEs	Kumar and Bansal (2015a,b)
Category (iii)			
13	KAKSI	Uses C α distances and backbone dihedral angles to show the concordance with the assignments found in the Protein Data Bank	Martin <i>et al.</i> (2005)
14	PALSSE	Mainly uses distance and torsion angle constraints to identify core elements and later extends them to longer segments	Majumdar <i>et al.</i> (2005)
15	SEGNO*	The C α atoms along with the backbone dihedral angles $((\varphi), \psi)$ and the angle-distance H-bond	Cubellis <i>et al.</i> (2005)

Source: Sklenar, H., Etchebest, C., Lavery, R., 1989. Describing protein structure: A general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins: Structure, Function, and Bioinformatics* 6, 46–60.

The algorithms are divided according to the categories mentioned in the main text. Algorithms marked by "*" identify PPII-helices also along with other SSEs.

A comparative analysis of number of residues identified as being part of α -helices by different algorithms suggests that there is on an average $\sim 80\%$ agreement (Kumar and Bansal, 2015a,b). However, it has also been observed that many algorithms prefer α -helices over π - or 3_{10} -helices (Fodje and Al-Karadaghi, 2002; Kumar and Bansal, 2015a,b; Shelar *et al.*, 2013). For example, residues Thr87-Leu134 of Oxidoreductase protein (PDB ID: 1SY: A) have different residue-wise assignments by various algorithms. Surprisingly the program XTLSSTR assigned extended β -strand to the residues Glu120-Ala121, whereas the same segment remained unassigned by ASSP and DSSP (Fig. 11). Differences in the assignments by various algorithms suggest that one cannot have a single algorithm that works well for every protein structure and hence one should be careful in selecting an algorithm.

Experimental Approaches for Identification of Secondary Structures

Circular dichroism (CD): CD uses the differential absorption of left- and right-handed light by chromophores with intrinsic chirality or placed in a chiral environment. Proteins, in general, contain various chromophores that engender CD signals. CD spectroscopy can provide the information about the SSEs content, any conformational changes upon ligand binding or macromolecular interactions. The thermal stability of the biomolecule can also be studied (Kelly and Price, 2000). The CD spectrum obtained in the far UV region (260–190 nm) corresponding to the peptide bond absorption can delineate the content of SSEs like α -helix and β -sheet; whereas the spectrum in near UV region (320–260 nm) provides an information about the environment around the aromatic amino acids or chromophores and hence about the tertiary structure. Units used in CD are ellipticity (measured in millidegrees and represented as θ) or mean residue ellipticity (MRE and represented as $[\theta]$). Several web-servers use MRE values from the CD-spectrum to provide the percentage SSE content of the protein (Louis-Jeune *et al.*, 2012; Micsonai *et al.*, 2015; Perez-Iratxeta and Andrade-Navarro, 2008; Sreerama and Woody, 2000; Whitmore and Wallace, 2004; Wiedemann *et al.*, 2013). Web-servers are also available to predict the CD

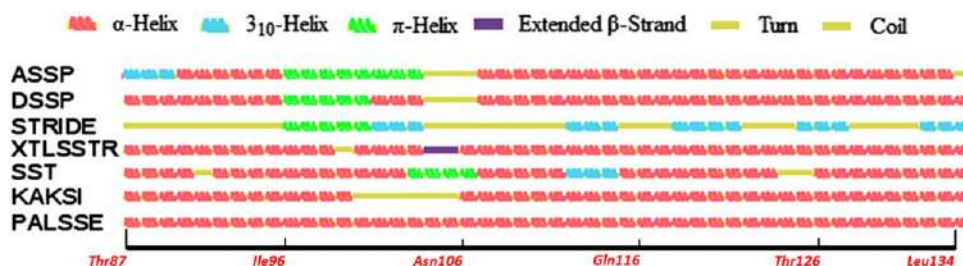


Fig. 11 Pictorial representation comparing the secondary structure assigned by different algorithms. Amino acid residues Thr87-Leu134 of Oxidoreductase protein (PDB ID: 1SYI; chain: A) is taken as an example.

spectrum of a modelled protein (Bulheller and Hirst, 2009; Mavridis and Janes, 2017). The updated version of CD is Synchrotron radiation circular dichroism (SRCD) spectroscopy that enhances the usage of conventional CD. The high light flux allows the collection of data at lower wavelengths, detection of spectra with higher signal-to-noise levels and measurements in the presence of strongly absorbing solvents also. Please refer to the reviews elsewhere (Kelly *et al.*, 2005; Kelly and Price, 2000; Miles and Wallace, 2006; Wallace, 2009) for better understanding and recent progresses in the field. CD and SRCD spectra as well as metadata can be deposited to open access data bank named PCDDDB (Whitmore *et al.*, 2017).

Knowledge of the 3D structures of biomolecules helps in understanding their function. At the same time, they can also provide an information about the SSEs that may be more accurate than CD. X-ray crystallography, NMR or cryo-electron microscopy are the most commonly used methods for solving 3D structures.

X-ray crystallography: this field uses the fundamentals of X-ray diffraction and mathematical approaches to symmetry to solve the structure of a biomolecule. Since 1957, the year in which the first crystal structure of myoglobin (Kendrew *et al.*, 1958) was solved, the field has helped researchers to understand the cellular processes in more detail and facilitated the advancement of modern medicines as well. The field has attracted more than 25 Nobel Prizes, starting in 1915 with the Bragg father-and-son team. Though the work on DNA double helix fetched noble prize to Watson and Crick in 1953, the structure of a 12-base-pair palindromic DNA was finally solved in 1980 by the X-ray crystallography method (Wing *et al.*, 1980). As of now, 30th September 2017, almost 90% of the total deposited structures in PDB are solved by this method. Few recent reviews elsewhere (Garman, 2014; Ilari and Savino, 2008; Powell, 2016; Shi, 2014; Smyth and Martin, 2000) can be an interesting read.

Nuclear Magnetic Resonance (NMR): two independent studies laid the foundation for a new technique, NMR (Bloch, 1946; Purcell *et al.*, 1946). For this pioneer work, Edward Mills Purcell and Felix Bloch received the 1952 Nobel Prize in Physics. NMR exploits the magnetic properties of certain atomic nuclei and can provide detailed information about the structure, dynamics, reaction state, and chemical environment of molecules. As of 30th September 2017, almost 12,000 three-dimension structures solved by NMR have been deposited in PDB. For better understanding of NMR spectroscopy, one can refer to available books (Abraham, 1978; Callaghan, 1993; Hore, 2015).

Cryo-electron microscopy (Cryo-EM): use of electrons in place of X-rays has a lot of advantages as they can be scattered more strongly and can be accelerated by electric fields. 3D structures of large biological assemblies in solution or cell can be well studied using 3D electron microscopy (3DEM) (Frank, 2011). Application of EM in biological specimens has seen a steep rise in popularity since the first electron microscope was developed and bacteriophages were imaged by the Ruska brothers (Ruska, 1941; Ruska *et al.*, 1939). Cryo-EM imaging technique requires much less sample compare to NMR or X-ray crystallography and have fewer restrictions on sample purity. There are mainly three categories of cryo-EM namely (1) Cryo-electron tomography, (2) single-particle cryo-EM and (3) electron crystallography. These sub-disciplines are either used in isolation or combined with other techniques to analyze biological structures in different contexts. The recent advancements in direct electron detectors have improved the resolution of the structures solved by cryo-EM such that in atomic resolution is now achievable (Grigorieff, 2013; Ruskin *et al.*, 2013). The gain in popularity of this method can be gauged by the fact that many reviews detailing the methodological advances are published every year (Milne *et al.*, 2013). Currently there are almost 1750 structures, solved by EM that have been deposited in PDB.

Distortions in Regular SSEs

The departure from the ideal values of the geometric parameters for a given SSE is often called distortions and most of the SSEs in protein structure possess it. Occurrence of $(i + 4 \rightarrow i)$ NH...O H-bonds between the amino acids in the main-chain, makes the α -helices quite uniform in terms of geometric parameters. However, most helices in proteins are curved (Barlow and Thornton, 1988; Kumar and Bansal, 1998) and deviate from the ideal values of defining parameters that can be because of the solvent induced distortions (Blundell *et al.*, 1983), peptide bond distortions (Barlow and Thornton, 1988; Love *et al.*, 1972) or presence of Pro (Chakrabarti *et al.*, 1986; MacArthur and Thornton, 1996). Blundell and coworkers (Blundell *et al.*, 1983) did the first survey on the curvature of helices to conclude that the majority of the helices are curved. Distortions caused by Pro residues (Chakrabarti *et al.*, 1986; Chakrabarti and Chakrabarti, 1998; MacArthur and Thornton, 1996; Sankaramakrishnan and Vishveshwara, 1990, 1992), Ser and Thr residues (Ballesteros *et al.*, 2000; Deupi *et al.*, 2004) residues have also been studied. An

interesting study of kinks caused by Pro residues in α -helices of transmembrane proteins (Yohannan *et al.*, 2004) reveals that the kinks are preserved even after the mutation of the Pro to other amino acids. These distortions in the helices do not allow the helix axis to follow a straight path. Recently, the analyses of simple insertions in α -helices of the coiled-coil structure has revealed that the insertion switches the coiled-coils into a novel fibrous protein fold (α/β coiled-coil) consisting of periodically interspersed short β -strands (Hartmann *et al.*, 2016). The interspersed π -helices are quite often considered as distortions in α -helices and often termed as α -aneurisms (Keefe *et al.*, 1993), α -bulges (Cartailler and Luecke, 2004) or π -bulges (Hardy *et al.*, 2000). The above mentioned belief was generally due to the biasness towards the α -helices (Fodje and Al-Karadaghi, 2002; Kumar and Bansal, 2015a,b). However, the presence of π or 3_{10} -helix distorts the entire helical segment by providing large bend (Kumar and Bansal, 2015a,b). Study has also shown that naturally occurring π -helices are evolutionarily related to α -helices (Cooley *et al.*, 2010).

Distortions in β -strands have also been reported (Blake and Oatley, 1977; Richardson *et al.*, 1978). One such distortion is a β -bulge that can be defined as a region between two consecutive β -type H-bonds that includes two residues on one strand and a single residue on the other strand (Richardson *et al.*, 1978). They occur very frequently in the coil regions, but are most easily visualized as local distortions in anti-parallel strands. Compared to parallel β -strands, it is generally believed that the anti-parallel strands are more stable as they can withstand greater twisting and other distortions like β -bulges and exposure to solvents. Only about 5% of the β -bulges were found between parallel strands (Richardson, 1981). Though β -bulges can be classified into several types, classic β -bulge occurs most frequently in the protein structures followed by 'G1', 'Wide' and 'parallel and GX' bulges. Classic β -bulge occurs between a narrow pair of H-bonds on anti-parallel strands and has the side chains of constituent residues in the same side of the β sheet, while G1 bulges can be defined as a classic bulge with Gly residue at the position 1. 'Wide type' β -bulges occur between a widely spaced pair of H-bonds on anti-parallel strands. GX bulges are the most unusual bulges and often contain Gly residue at position X. β -bulges affect the directionalities of the protein chain in the 3D structure, making them useful for determining the large features of β -sheet and/or extended hairpin loops.

A twist in the β -strands of a sheet can also be considered as a distortion as it introduces local conformational irregularities into the polypeptide backbone. Nevertheless, an extremely strong local twist helps to close the anti-parallel β -barrels containing five or six strands. Sheets with a right handed twist are more common and have a lower free energy than their straight or left-handed counterparts (Chothia, 1973).

Protein Structure and Sequence Repositories

PDB (Berman *et al.*, 2000) is the most comprehensive repository of structure data for biological macromolecules. The repository contains the primary structure and secondary structure information along with the atomic coordinates of a constituent atoms of biomolecule. It also contains corresponding experimental data. PDB101, an education portal of PDB provides detailed information about the PDB. As of 27th September 2017, PDB contains structure data for 133,920 Biological Macromolecular Structures (Fig. 12). On an average, the length of proteins ranges between 100 and 300 residues. However, there are big proteins containing 1000 or more residues as well as small proteins with at most 30 residues.

A dedicated data bank, EMDataBank (Lawson *et al.*, 2016), is also available online at EMDataBank (see Relevant Websites section) that is a unified global portal for deposition and retrieval of 3DEM density maps, atomic models, associated metadata and related stuffs. It has 5195 EMDB map entries, 1805 PDB coordinate entries as of 30th September 2017.

The 3D structures of experimentally-determined nucleic acids and complex assemblies can be deposited into a databank called nucleic acid database (NDB) (Coimbatore Narayanan *et al.*, 2014). The repository contains the sequences of the corresponding nucleic acid chain along with the derived geometric data, classifications of structures and motifs, standards for describing nucleic acid features. It is also linked to various software for the analyses. As of 27th September 2017, number of released structures in NDB is 9133.

Apart from PDB, NCBI (Geer *et al.*, 2010) and UniProt (Consortium, 2015) databases also provide comprehensive, high-quality and freely accessible resource of protein sequence and functional information. Another very important database, Pfam (Punta *et al.*, 2012), contains a large collection of protein families, each represented by multiple sequence alignments and Hidden Markov Models. The PDB ID of the members of each family is linked to the PDB database.

Several initiatives have been taken in the past to classify proteins into groups based on properties that are shared by all protein members of a group. Different properties such as structural similarity, sequence similarity and functional similarity have been employed to classify proteins. Classifying proteins in such a manner is especially useful in functional annotation of proteins which are newly discovered from genome sequence data. Also, it has been useful in predicting tentative structures for proteins with no 3D structural data.

Protein 3D Structure Prediction and Analyses

Protein structure prediction and engineering-design aim to fill the huge gap between the sequence and structure space. The protein structure prediction methods can be categorized into mainly three parts (1) *ab initio* methods (2) Threading (3) Homology modelling. The evolutionary information from the genomic sequences can be utilized efficiently to compute the protein 3D structures from their amino acid sequences (Marks *et al.*, 2012). The *ab initio* (*de-novo*) methods are based on the first

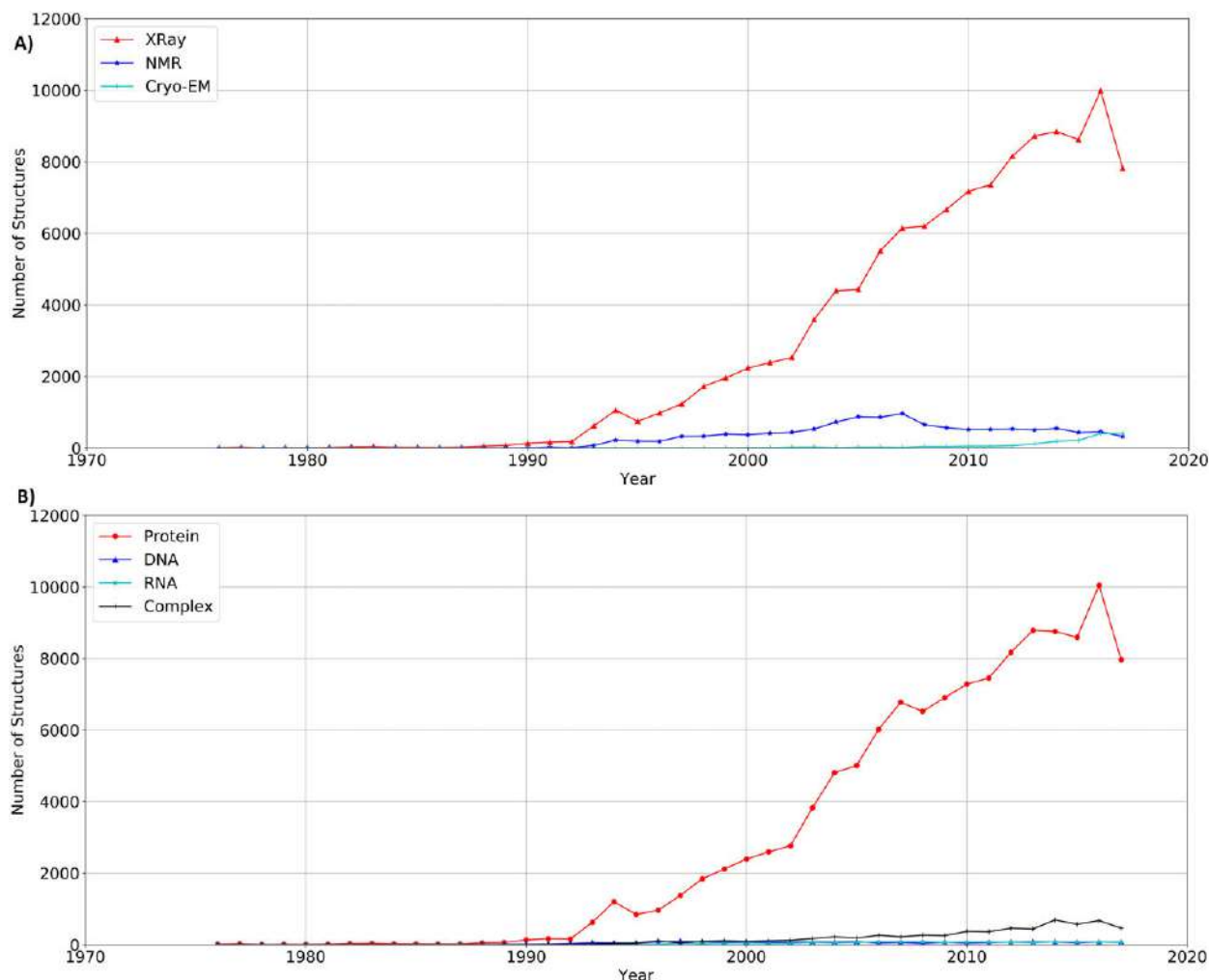


Fig. 12 Number of 3D structures deposited in the PDB on yearly basis as of 30th September 2017.

principle laws of physics as well as chemistry and based on the fact that the native structure of protein is always at energy minimum. Using this method, modelling a protein sequence of length > 150 residues can be a challenge. However, the *de-novo* design of peptides using the computer-aided parameterization offers one solution. Even, the *de-novo* protein design can be used to create small stable proteins that can bind to any specific therapeutic targets (Chevalier *et al.*, 2017). Another method, threading or fold recognition searches the protein structure template in a library with lowest possible energy for the query sequence. Homology modelling or comparative modelling uses the protein sequence identity between the input protein sequence and sequence of the template structure to construct the 3D structure. Hence better identity between query and target sequence will fetch a better modelled structure.

Plethora of review articles are available on strategies and challenges to protein structure prediction (Bonneau and Baker, 2001; Dorn *et al.*, 2014; Floudas *et al.*, 2006; Kc, 2016; Petrey and Honig, 2005).

Once the structure is modelled, it is always advised to check its quality using PROCHECK (Laskowski *et al.*, 1993), MOL-PROBITY (Chen *et al.*, 2010), VERIFY_3D (Bowie *et al.*, 1991; Luthy *et al.*, 1992) and WHAT_CHECK (Hoofst *et al.*, 1996). 3D structures of proteins open doors for various other structural analyses like functional annotation, interaction pathway analyses, target drug identification. Proteins in a cell cannot work alone and need interacting partners that can be a small molecule or biomolecule. Protein-protein interactions are mediated by hydrophobic interactions as well as H-bonds (De Las Rivas and Fontanillo, 2010). Understanding the importance of these interactions is vital in design and engineering of functional assemblies. The functional importance of the non-bonded interactions in assembly of biomolecular structures has been analyzed and firmly established through studies on DNA-protein (Mandel-Gutfreund *et al.*, 1998), RNA-protein (Wilson and Nierhaus, 2003), protein-protein (Singh *et al.*, 2003; Zondlo, 2010) and protein-ligand interactions (Panigrahi and Desiraju, 2007). Knowing the importance of these interactions, many algorithms have been developed over the years (Babu, 2003; Kumar *et al.*, 2014; Lindauer *et al.*, 1996; McDonald and Thornton, 1994; Tina *et al.*, 2007; Tiwari and Panigrahi, 2007).

Polysaccharides-Macromolecular Biopolymer

Classically, polysaccharides are defined as high molecular weight macromolecules composed of two or more monosaccharides that are connected by glycosidic bonds (Aspinall, 1985; Bochkov and Zaikov, 2016). Depending on the type of building units, polysaccharides may be classified as homosaccharides and heterosaccharides. Most polysaccharides are used as energy-storing units inside the living cells and tissues (Nonogaki *et al.*, 2000), while some of them are involved in cellular signaling (Tincer *et al.*, 2015) and in forming the protective outer membrane of living cells (Zhang *et al.*, 2016). It has some *in vitro* applications as well in pharmacy, textiles (Sanghi *et al.*, 2007) and nanomaterials (Lin *et al.*, 2012). Over the past two decades, numerous reviews and symposia have been published on the structure, metabolism and function of polysaccharides (Ramberg *et al.*, 2010; Srivastava and Kulshreshtha, 1989; Zong *et al.*, 2012). With the growing interest in this field, development of advanced tools and software for prediction of secondary structure of polysaccharides is becoming imperative (Pérez *et al.*, 2014). Most of these approaches are aimed at developing computational based techniques from NMR parameters of polysaccharides (Toukach and Ananikov, 2013).

Structural Features of Food Storage Polysaccharides

The three main food storage polysaccharides within the living cells are starch, cellulose and glycogen. A molecular genetics approach and 3D structure prediction methods have been chosen to determine the functional domain of the storage polysaccharides (Hostinová *et al.*, 2003). Sequence homology has also been applied for these purposes (Svensson *et al.*, 1989). Starch is formed when glucoses are linearly linked up by α 1–4 linkages (Fig. 13) (Brown and Kelly, 1993). Cellulose is mainly produced

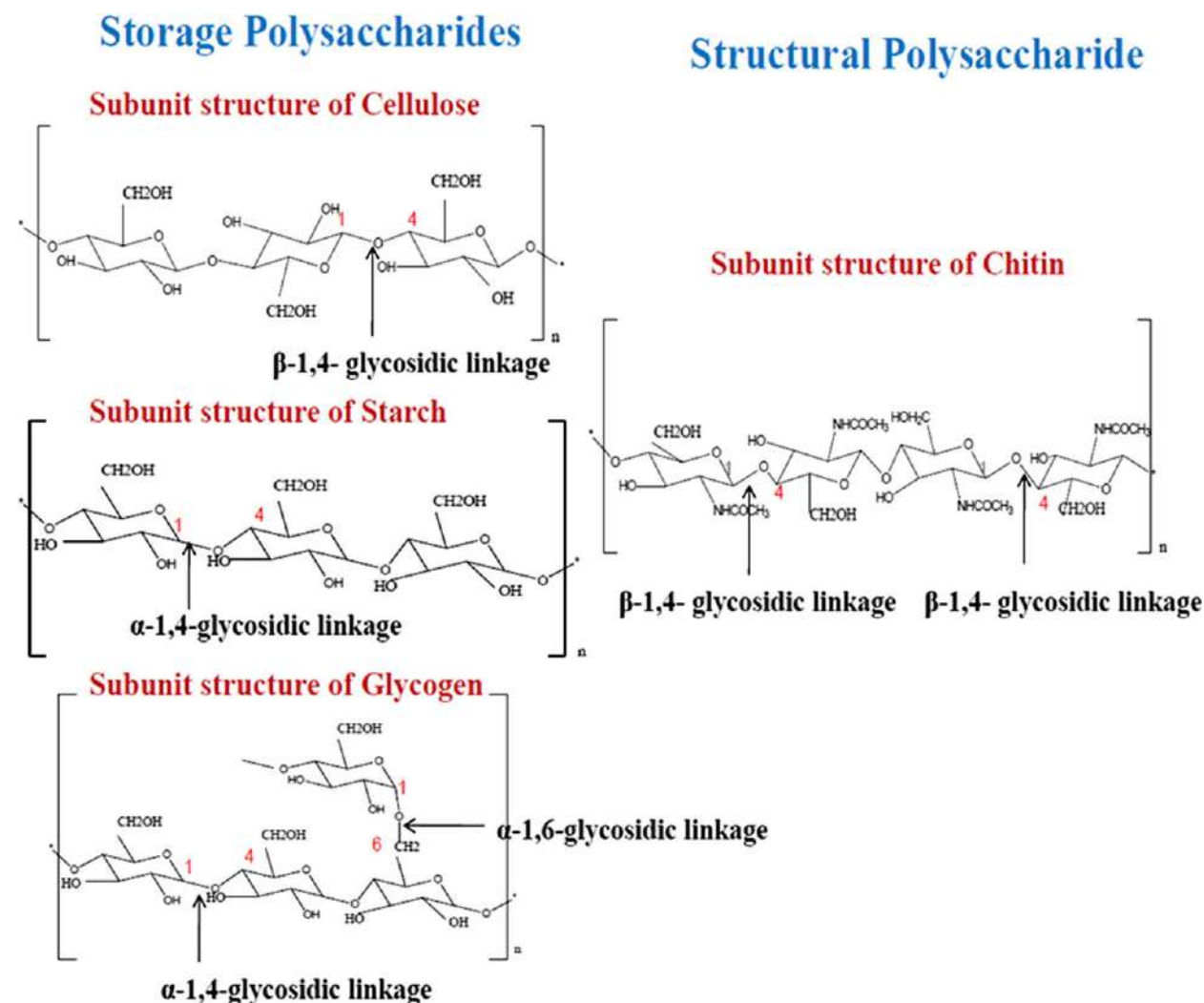


Fig. 13 Subunit structures of some common storage and structural polysaccharides. All structures are created by using ChemDraw Ultra software. Rhodes, D., Lipps, H.J., 2015. G-quadruplexes and their regulatory roles in biology. *Nucleic Acids Research* 43, 8627–8637.

from the hydrated form of some bacteria (e.g., *Acetobacter Xylinum*) (Ross *et al.*, 1987) containing thousands of beta-D-glucopyranose units.

Glycogen is the multi-branched polysaccharides of glucose. It serves as a form of energy storage in humans, animals, fungi, and bacteria (Arrese and Soulages, 2010). There are two types of linkages, mainly 1–4 linkages are found in the straight part where as 1–6 linkages are found in the branched part. The straight part is helically twisted in nature and containing six glucose units per turn.

Structural Features of Chitin Polysaccharides

Structural polysaccharides are mainly involved in forming cell walls of different plants and animals. Chitin and cellulose both are the examples of structural polysaccharides. Chitin is a naturally abundant mucopolysaccharide (Sekiguchi *et al.*, 1994). It consists of 2-acetamido-2-deoxy- β -D-glucose through a β (1 $\rightarrow$ 4) linkage (Muzzarelli, 1973).

Polysaccharide 3D Structure Prediction and Analyses

Polysaccharides and their glycol-conjugates play an important role in cell-cell adhesion (Gribova *et al.*, 2011), inflammation (Jiang *et al.*, 2010) and immune activity (Yang *et al.*, 2009; Yongxu and Jicheng, 2008). Therefore secondary structure predictions of polysaccharides become a crucial thing in understanding the biological phenomenon (Dwek, 1996). Secondary structures possess regularities in various geometric and conformational parameters like glycosidic angles and specific patterns of H-bonds. Several computational techniques have been developed for analyzing the conformations of polysaccharides *in vacuo* (Wormald *et al.*, 2002). Monte Carlo methods (Peters *et al.*, 1993) and genetic algorithms (Nahmany *et al.*, 2005) have also been used for these purpose. Molecular dynamics simulations were carried out using GROMACS software (Van Der Spoel *et al.*, 2005) with OPLS-AA force field (Jorgensen *et al.*, 1996). Carb-builder is one of the software that is used to build the molecular models of complex oligosaccharides or polysaccharides (Kuttel *et al.*, 2016). POLY 2.0 is another open source software (Engelsen *et al.*, 2014). It can generate complex branched carbohydrates and polysaccharides structures using data from their optimized building mono-saccharides and glycosidic linkages. SWEET also serves the same purpose. It is a web-based software that quickly forms reliable three dimensional structures of polysaccharides from their preliminary sequence information (Bohne *et al.*, 1999). Advanced version of the SWEET only generates one of the conformations out of the manifold. Generated structure shows all the glycosidic linkages with reported variation of phi, psi and omega values. Sugar folding, a computer based tool is also used to predict the conformation of complex oligosaccharides in solution (Xia *et al.*, 2007). There is an effective software CASPER, used to determine the structure of the branched oligosaccharides and polysaccharides (Stenutz *et al.*, 1998). The CASPER program was developed for the analysis of the primary structures of oligosaccharides and for polysaccharides with repeating units. The approach is based mainly on NMR spectroscopy. Once the structure is modelled, we can visualize the secondary structures by SweetUnityMol, which is one of the specific computer graphics based software and can depict the complex interactions within polysaccharides (Pérez *et al.*, 2014).

Recent Advancement

A lot of polysaccharide structures have been elucidated from the crystallography in the last few years. Though the actual structure-function relationship is not yet established for many polysaccharides due to their diverse and complex structure. For the last three decades, besides the computer based tools, advanced NMR spectroscopy has also played an important role in determination of the structure of polysaccharides. A lot of advanced NMR techniques have been used, such as COSY (Lack *et al.*, 2007; Sinha *et al.*, 2010), NOESY (Fedonenko *et al.*, 2002; Larocque *et al.*, 2003; Linker *et al.*, 2001), TOCSY (Johnson and Pinto, 2002; Kjellberg *et al.*, 1998; Maciel *et al.*, 2008), HETCOR (Cescutti *et al.*, 1993) and NOE (Martin-Pastor and Bush, 1999; Michael *et al.*, 2002; Yamasaki and Bacon, 1991). Advancements in NMR methods with potential applications towards polysaccharide studies open a new area of research. Coarse grained molecular simulations (Bathe *et al.*, 2005; Wohler and Berglund, 2011) can also provide some new insights in this field.

Closing Remarks: How Close Are We?

Lack of enough number of 3D structures and presence of more number of atoms that can interact to each other, make the SSE prediction and assignment difficult for nucleic acids. In contrast to di-peptide, di-nucleotide has got more flexibility as they have seven back bone torsion angles compared to only three in di-peptide. For example, there is an exponential increase in the number of possible structures with increase in the length of RNA (Zuker and Sankoff, 1984). A nucleic acid chain with 100 nucleotides has more than 10^{25} possible secondary structures (Mathews, 2006). Further, the discovery of RNA thermometers in 1989 (Altuvia *et al.*, 1989), made things more complicated as it was found that certain types of sequences can be very sensitive to change in temperature.

For a given protein structure, there is not a consensus in identifying SSEs. Especially, the termini of SSEs are not defined uniformly. In general, they have an agreement of about 80%. Many less frequently occurring SSEs like π -helices and PPII-helices are not defined in clear cut way. In fact, none of the available algorithms can identify 2.2₇-helices. Same holds true for the prediction of SSEs for any input sequence, but the accuracy of their prediction is improving with the increase in the number of 3D structures of proteins. Are we able to design a desired protein structure with natural amino acids? Answer is not a perfect yes. Are we able to incorporate the enzymatic activities in a designed peptides or proteins? Though the answer to this question can be yes, it will be with a caveat: the activity is far less than that in the native protein. A lot of work has been done, but there is still a long way to go. For example, information on which amino acids favour 3₁₀ or π -helical conformations over α -helices, is still not clear. However, we are slowly and steadily reaching towards the goal and increasing number of 3D structures, as well as computational studies are playing a major role in understanding these features.

See also: *Ab initio* Protein Structure Prediction. Algorithms for Structure Comparison and Analysis: Docking. Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins. Algorithms for Structure Comparison and Analysis: Prediction of Tertiary Structures of Proteins. Biological Database Searching. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition

References

- Abraham, R.J., 1978. Nuclear Magnetic Resonance. Royal Society of Chemistry.
- Abrahams, J.P., van den Berg, M., van Batenburg, E., Pleij, C., 1990. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Research* 18, 3035–3044.
- Adamczak, R., Porollo, A., Meller, J., 2005. Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins* 59, 467–475.
- Adzhubei, A.A., Sternberg, M.J., 1993. Left-handed polypyrrolone II helices commonly occur in globular proteins. *Journal of Molecular Biology* 229, 472–493.
- Altuvia, S., Kornitzer, D., Teff, D., Oppenheim, A.B., 1989. Alternative mRNA structures of the cIII gene of bacteriophage lambda determine the rate of its translation initiation. *Journal of Molecular Biology* 210, 265–280.
- Andreeva, A., Howorth, D., Chothia, C., *et al.*, 2014. SCOP2 prototype: A new approach to protein structure mining. *Nucleic Acids Research* 42, D310–D314.
- Arrese, E.L., Soulages, J.L., 2010. Insect fat body: Energy, metabolism, and regulation. *Annual Review of Entomology* 55, 207–225.
- Aspinall, G.O., 1985. The Polysaccharides, 3. Academic Press.
- Babu, M.M., 2003. NCI: A server to identify non-canonical interactions in protein structures. *Nucleic Acids Research* 31, 3345–3348.
- Ballesteros, J.A., Deupi, X., Olivella, M., *et al.*, 2000. Serine and threonine residues bend α -helices in the $\chi_1 = g^-$ conformation. *Biophysical Journal* 79, 2754–2760.
- Bansal, M., 1999. D.N.A. structure: Yet another avatar? *Current Science* 76, 1178–1181.
- Bansal, M., Bhattacharyya, D., Ravi, B., 1995. NUPARM and NUCGEN: Software for analysis and generation of sequence dependent nucleic acid structures. *Computer Applications in the Biosciences: CABIOS* 11, 281–287.
- Bansal, M., Kumar, S., Velavan, R., 2000. HELANAL: A program to characterize helix geometry in proteins. *Journal of Biomolecular Structure and Dynamics* 17, 811–819.
- Bansal, M., Srinivasan, N., 2013. Biomolecular Forms and Functions: A Celebration of 50 Years of the Ramachandran Map. World Scientific.
- Barlow, D.J., Thornton, J.M., 1988. Helix geometry in proteins. *Journal of Molecular Biology* 201, 601–619.
- Bathe, M., Rutledge, G.C., Grodzinsky, A.J., Tidor, B., 2005. A coarse-grained molecular model for glycosaminoglycans: Application to chondroitin, chondroitin sulfate, and hyaluronic acid. *Biophysical Journal* 88, 3870–3887.
- Bellaousov, S., Mathews, D.H., 2010. ProbKnot: Fast prediction of RNA secondary structure including pseudoknots. *RNA* 16, 1870–1880.
- Bellaousov, S., Reuter, J.S., Seetin, M.G., Mathews, D.H., 2013. RNAstructure: Web servers for RNA secondary structure prediction and analysis. *Nucleic Acids Research* 41, W471–W474.
- Benabou, S., Avino, A., Eritja, R., *et al.*, 2014. Fundamental aspects of the nucleic acid i-motif structures. *RSC Advances* 4, 26956–26980.
- Berendsen, H.J., van der Spoel, D., van Drunen, R., 1995. GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications* 91, 43–56.
- Berisio, R., Vitagliano, L., 2012. Polypyrrolone and triple helix motifs in host-pathogen recognition. *Current Protein & Peptide Science* 13, 855–865.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Research* 28, 235–242.
- Besnard, E., Babled, A., Lapasset, L., *et al.*, 2012. Unraveling cell type-specific and reprogrammable human replication origin signatures associated with G-quadruplex consensus motifs. *Nature Structural & Molecular Biology* 19, 837–844.
- Bhattacharjee, A., Bansal, M., 2005. Collagen structure: The Madras triple helix and the current scenario. *IUBMB Life* 57, 161–172.
- Bhattacharyya, D., Halder, S., Basu, S., *et al.*, 2017. RNAHelix: Computational modeling of nucleic acid structures with Watson–Crick and non-canonical base pairs. *Journal of Computer-Aided Molecular Design* 31, 219–235.
- Birney, E., Stamatoyannopoulos, J.A., Dutta, A., *et al.*, 2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 447, 799–816.
- Blake, C.C.F., Oatley, S.J., 1977. Protein-DNA and protein-hormone interactions in prealbumin: A model of the thyroid hormone nuclear receptor? *Nature* 268, 115–120.
- Blake, C.C., Koenig, D.F., Mair, G.A., *et al.*, 1965. Structure of hen egg-white lysozyme. A three-dimensional Fourier synthesis at 2 Å resolution. *Nature* 206, 757–761.
- Blanchet, C., Pasi, M., Zakrzewska, K., Lavery, R., 2011. CURVES+ web server for analyzing and visualizing the helical, backbone and groove parameters of nucleic acid structures. *Nucleic Acids Research* 39, W68–W73.
- Bloch, F., 1946. Nuclear induction. *Physical Review* 70, 460.
- Blundell, T., Barlow, D., Borkakoti, N., Thornton, J., 1983. Solvent-induced distortions and the curvature of alpha-helices. *Nature* 306, 281–283.
- Bochkov, A.F., Zaikov, G.E., 2016. Chemistry of the O-Glycosidic Bond: Formation and Cleavage. Elsevier.
- Bohne, A., Lang, E., von der Lieth, C.-W., 1999. SWEET-WWW-based rapid 3D construction of oligo- and polysaccharides. *Bioinformatics (Oxford, England)* 15, 767–768.
- Bonneau, R., Baker, D., 2001. *Ab initio* protein structure prediction: Progress and prospects. *Annual Review of Biophysics and Biomolecular Structure* 30, 173–189.
- Boudard, M., Barth, D., Bernauer, J., *et al.*, 2017. GARN2: Coarse-grained prediction of 3D structure of large RNA molecules by regret minimization. *Bioinformatics*. btx175.
- Boudvillain, M., de Lencastre, A., Pyle, A.M., 2000. A tertiary interaction that links active-site domains to the 5' splice site of a group II intron. *Nature* 406, 315–318.
- Bowes, J.H., Kenten, R., 1948. The amino-acid composition and titration curve of collagen. *Biochemical Journal* 43, 358.
- Bowie, J.U., Luthy, R., Eisenberg, D., 1991. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253, 164–170.

- Bragg, L., Kendrew, J.C., Perutz, M.F., 1950. Polypeptide chain configurations in crystalline proteins. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 203, 321–357.
- Brown, S.H., Kelly, R.M., 1993. Characterization of amylolytic enzymes, having both α -1, 4 and α -1, 6 hydrolytic activity, from the thermophilic archaea *Pyrococcus furiosus* and *Thermococcus litoralis*. *Applied and Environmental Microbiology* 59, 2614–2621.
- Bru, C., Courcelle, E., Carrere, S., *et al.*, 2005. The ProDom database of protein domain families: More emphasis on 3D. *Nucleic Acids Research* 33, D212–D215.
- Bulheller, B.M., Hirst, J.D., 2009. DichroCalc – Circular and linear dichroism online. *Bioinformatics* 25, 539–540.
- Burge, S., Parkinson, G.N., Hazel, P., *et al.*, 2006. Quadruplex DNA: Sequence, topology and structure. *Nucleic Acids Research* 34, 5402–5415.
- Callaghan, P.T., 1993. *Principles of Nuclear Magnetic Resonance Microscopy*. Oxford University Press on Demand.
- Cao, C., Xu, S., Wang, L., 2015. An algorithm for protein helix assignment using helix geometry. *PLOS ONE* 10, e0129674.
- Cartailler, J.P., Luecke, H., 2004. Structural and functional characterization of pi bulges and other short intrahelical deformations. *Structure* 12, 133–144.
- Carter Jr., C.W., Kraut, J., Freer, S.T., *et al.*, 1974. Two-Angstrom crystal structure of oxidized Chromatium high potential iron protein. *The Journal of Biological Chemistry* 249, 4212–4225.
- Carugo, O., Djinic-Carugo, K., 2013. Half a century of Ramachandran plots. *Acta Crystallographica Section D* 69, 1333–1341.
- Cesutti, P., Toffanin, R., Kvam, B.J., *et al.*, 1993. Structural determination of the capsular polysaccharide produced by *Klebsiella pneumoniae* serotype K40. *FEBS Journal* 213, 445–453.
- Cevce, M., Plavec, J., 2005. Role of loop residues and cations on the formation and stability of dimeric DNA G-quadruplexes. *Biochemistry* 44, 15238–15246.
- Chakrabarti, P., Bernard, M., Rees, D.C., 1986. Peptide-bond distortions and the curvature of alpha-helices. *Biopolymers* 25, 1087–1093.
- Chakrabarti, P., Chakrabarti, S., 1998. C–H...O hydrogen bond involving proline residues in alpha-helices. *Journal of Molecular Biology* 284, 867–873.
- Cheng, X., Shao, Z., Li, C., *et al.*, 2017. Isolation, characterization and evaluation of collagen from jellyfish *Rhopilema esculentum* Kishinouye for use in hemostatic applications. *PLOS ONE* 12, e0169731.
- Chen, V.B., Arendall III, W.B., Headd, J.J., *et al.*, 2010. MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallographica Section D* 66, 12–21.
- Chevalier, A., Silva, D.A., Rocklin, G.J., *et al.*, 2017. Massively parallel de novo protein design for targeted therapeutics. *Nature* 550 (7674), 74–79.
- Chothia, C., 1973. Conformation of twisted β -pleated sheets in proteins. *Journal of Molecular Biology* 75, 295–302.
- Chou, P.Y., Fasman, G.D., 1974. Prediction of protein conformation. *Biochemistry* 13, 222–245.
- Chou, P.Y., Fasman, G.D., 1977. Beta-turns in proteins. *Journal of Molecular Biology* 115, 135–175.
- Chvapil, M., 1977. Collagen sponge: Theory and practice of medical applications. *Journal of Biomedical Materials Research Part A* 11, 721–741.
- Chvapil, M., Kronenthal, R.L., van Winkle Jr, W., 1973. Medical and surgical applications of collagen. *International Review of Connective Tissue Research* 6.
- Clayton, G.M., Altieri, S., Heginbotham, L., *et al.*, 2008. Structure of the transmembrane regions of a bacterial cyclic nucleotide-regulated channel. *Proceedings of the National Academy of Sciences of the United States of America* 105, 1511–1515.
- Coimbatore Narayanan, B., Westbrook, J., Ghosh, S., *et al.*, 2014. The nucleic acid database: New features and capabilities. *Nucleic Acids Research* 42, D114–D122.
- Consortium, T.U., 2015. UniProt: A hub for protein information. *Nucleic Acids Research* 43, D204–D212.
- Cooley, R.B., Arp, D.J., Karplus, P.A., 2010. Evolutionary origin of a secondary structure: Pi-helices as cryptic but widespread insertional variations of alpha-helices that enhance protein functionality. *Journal of Molecular Biology* 404, 232–246.
- Costantini, S., Colonna, G., Facchiano, A.M., 2006. Amino acid propensities for secondary structures are influenced by the protein structural class. *Biochemical and Biophysical Research Communications* 342, 441–451.
- Costantini, S., Colonna, G., Facchiano, A.M., 2007. PreSSAPro: A software for the prediction of secondary structure by amino acid properties. *Computational Biology and Chemistry* 31, 389–392.
- Cowan, P.M., McGavin, S., 1955. Structure of poly-L-proline. *Nature* 176, 501–503.
- Cubellis, M.V., Cailliez, F., Lovell, S.C., 2005. Secondary structure assignment that accurately reflects physical and evolutionary characteristics. *BMC Bioinformatics* 6, S8.
- Das, R., Baker, D., 2008. Macromolecular modeling with rosetta. *Annual Review of Biochemistry* 77, 363–382.
- Davis, I.W., Murray, L.W., Richardson, J.S., Richardson, D.C., 2004. MOLPROBITY: Structure validation and all-atom contact analysis for nucleic acids and their complexes. *Nucleic Acids Research* 32, W615–W619.
- Day, H.A., Pavlou, P., Waller, Z.A., 2014. i-Motif DNA: Structure, stability and targeting with ligands. *Bioorganic & Medicinal Chemistry* 22, 4407–4418.
- De Las Rivas, J., Fontanillo, C., 2010. Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLOS Computational Biology* 6, e1000807.
- De Paepe, A., 1998. Heritable collagen disorders: From phenotype to genotype. *Verh K Acad Geneesk Belg* 60, 463–482.
- Deupi, X., Olivella, M., Govaerts, C., *et al.*, 2004. Ser and Thr residues modulate the conformation of pro-kinked transmembrane alpha-helices. *Biophysical Journal* 86, 105–115.
- Devi, G., Zhou, Y., Zhong, Z., *et al.*, 2015. RNA triplexes: From structural principles to biological and biotech applications. *Wiley Interdisciplinary Reviews: RNA* 6, 111–128.
- Dirks, R.M., Pierce, N.A., 2003. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry* 24, 1664–1677.
- Donohue, J., 1953. Hydrogen bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America* 39, 470–478.
- Dorn, M., S, M.B.E., Buriol, L.S., Lamb, L.C., 2014. Three-dimensional protein structure prediction: Methods and computational strategies. *Computational Biology and Chemistry* 53PB, 251–276.
- Dor, O., Zhou, Y., 2007. Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins* 66, 838–845.
- Drozdetskiy, A., Cole, C., Procter, J., Barton, G.J., 2015. JPred4: A protein secondary structure prediction server. *Nucleic Acids Research* 43, W389–W394.
- Dwek, R.A., 1996. Glycobiology: Toward understanding the function of sugars. *Chemical Reviews* 96, 683–720.
- Eastoe, J., 1955. The amino acid composition of mammalian collagen and gelatin. *Biochemical Journal* 61, 589.
- Engelsen, S.B., Hansen, P.I., Perez, S., 2014. POLYS 2.0: An open source software package for building three-dimensional structures of polysaccharides. *Biopolymers* 101, 733–743.
- Enkhbayar, P., Hikichi, K., Osaki, M., *et al.*, 2006. 3(10)-helices in proteins are parahelices. *Proteins* 64, 691–699.
- Faraggi, E., Yang, Y., Zhang, S., Zhou, Y., 2009. Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure* 17, 1515–1527.
- Fedonenko, Y.P., Zatonsky, G.V., Konnova, S.A., *et al.*, 2002. Structure of the O-specific polysaccharide of the lipopolysaccharide of *Azospirillum brasilense* Sp245. *Carbohydrate Research* 337, 869–872.
- Fernando, H., Sewitz, S., Darot, J., *et al.*, 2009. Genome-wide analysis of a G-quadruplex-specific single-chain antibody that regulates gene expression. *Nucleic Acids Research* 37, 6716–6722.
- Floudas, C., Fung, H., McAllister, S., *et al.*, 2006. Advances in protein structure prediction and de novo protein design: A review. *Chemical Engineering Science* 61, 966–988.
- Fodje, M.N., Al-Karadaghi, S., 2002. Occurrence, conformational features and amino acid propensities for the pi-helix. *Protein Engineering* 15, 353–358.
- Frank, J., 2011. *Molecular Machines in Biology: Workshop of the Cell*. Cambridge University Press.
- Franklin, R.E., Gosling, R.G., 1953. The structure of sodium thymonucleate fibres. I. The influence of water content. *Acta Crystallographica* 6, 673–677.
- Frishman, D., Argos, P., 1995. Knowledge-based protein secondary structure assignment. *Proteins* 23, 566–579.
- Gaisi, T., Giunta, C., Bratton, D.J., *et al.*, 2017. Obstructive sleep apnoea and quality of life in Ehlers-Danlos syndrome: A parallel cohort study. *Thorax*. [thoraxjnl-2016-209560].
- Garman, E.F., 2014. Developments in x-ray crystallographic structure determination of biological macromolecules. *Science* 343, 1102–1108.
- Garnier, J., Osguthorpe, D.J., Robson, B., 1978. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology* 120, 97–120.

- Geer, L.Y., Marchler-Bauer, A., Geer, R.C., *et al.*, 2010. The NCBI BioSystems database. *Nucleic Acids Research* 38, D492–D496.
- Gehring, K., Leroy, J.-L., Guéron, M., 1993. A tetrameric DNA structure with protonated cytosine-cytosine base pairs. *Nature* 363, 561–565.
- Gellert, M., Lipsett, M.N., Davies, D.R., 1962. Helix formation by guanylic acid. *Proceedings of the National Academy of Sciences* 48, 2013–2018.
- Gerstein, M.B., Bruce, C., Rozowsky, J.S., *et al.*, 2007. What is a gene, post-ENCODE? History and updated definition. *Genome Research* 17, 669–681.
- Ghosh, A., Bansal, M., 2003. A glossary of DNA structures from A to Z. *Acta Crystallographica D Biological Crystallography* 59, 620–626.
- Gibrat, J.F., Madej, T., Bryant, S.H., 1996. Surprising similarities in structure comparison. *Current Opinion in Structural Biology* 6, 377–385.
- Gribova, V., Auzely-Velty, R., Picart, C., 2011. Polyelectrolyte multilayer assemblies on materials surfaces: From cell adhesion to tissue engineering. *Chemistry of Materials* 24, 854–869.
- Grigorieff, N., 2013. Structural biology: Direct detection pays off for electron cryo-microscopy. *Elife* 2, e00573.
- Gruber, A.R., Lorenz, R., Bernhart, S.H., *et al.*, 2008. The Vienna RNA websuite. *Nucleic Acids Research* 36, W70–W74.
- Guéron, M., Leroy, J.-L., 2000. The i-motif in nucleic acids. *Current Opinion in Structural Biology* 10, 326–331.
- Guzzo, A.V., 1965. The influence of amino-acid sequence on protein structure. *Biophysical Journal* 5, 809–822.
- Hale, B.G., Kerry, P.S., Jackson, D., *et al.*, 2010. Structural insights into phosphoinositide 3-kinase activation by the influenza A virus NS1 protein. *Proceedings of the National Academy of Sciences of the United States of America* 107, 1954–1959.
- Hänsel-Hertsch, R., Di Antonio, M., Balasubramanian, S., 2017. DNA G-quadruplexes in the human genome: Detection, functions and therapeutic potential. *Nature Reviews Molecular Cell Biology* 18, 279–284.
- Hardy, J.A., Walsh, S.T., Nelson, H.C., 2000. Role of an alpha-helical bulge in the yeast heat shock transcription factor. *Journal of Molecular Biology* 295, 393–409.
- Harmanci, A.O., Sharma, G., Mathews, D.H., 2011. TurboFold: Iterative probabilistic estimation of secondary structures for multiple RNA sequences. *BMC Bioinformatics* 12, 108.
- Hartmann, M.D., Mender, C.T., Bassler, J., *et al.*, 2016. α/β coiled coils. *eLife* 5, e11861.
- Holley, R.W., Appar, J., Everett, G.A., *et al.*, 1965. Structure of a ribonucleic acid. *Science* 1462–1465.
- Hollyfield, J.G., Besharse, J.C., Rayborn, M.E., 1976. The effect of light on the quantity of phagosomes in the pigment epithelium. *Experimental Eye Research* 23, 623–635.
- Holm, L., Sander, C., 1996. Mapping the protein universe. *Science* 273, 595–603.
- Hol, W.G., van Duijnen, P.T., Berendsen, H.J., 1978. The alpha-helix dipole and the properties of proteins. *Nature* 273, 443–446.
- Hooft, R.W., Vriend, G., Sander, C., Abola, E.E., 1996. Errors in protein structures. *Nature* 381, 272.
- Hore, P.J., 2015. *Nuclear Magnetic Resonance*. USA: Oxford University Press.
- Hostinová, E., Solovíková, A., Dvorský, R., Gašperík, J., 2003. Molecular cloning and 3D structure prediction of the first raw-starch-degrading glucoamylase without a separate starch-binding domain. *Archives of Biochemistry and Biophysics* 411, 189–195.
- Huggins, M.L., 1943. The structure of fibrous proteins. *Chemical Reviews* 32, 195–218.
- Humphrey, W., Dalke, A., Schulten, K., 1996. VMD: Visual molecular dynamics. *Journal of Molecular Graphics* 14, 33–38.
- Ilari, A., Savino, C., 2008. Protein structure determination by x-ray crystallography. *Methods in Molecular Biology* 452, 63–87.
- Jiang, J.B., Qiu, J.D., Yang, L.H., *et al.*, 2010. Therapeutic effects of astragalus polysaccharides on inflammation and synovial apoptosis in rats with adjuvant-induced arthritis. *International Journal of Rheumatic Diseases* 13, 396–405.
- Jain, A., Wang, G., Vasquez, K.M., 2008. DNA triple helices: Biological consequences and therapeutic potential. *Biochimie* 90, 1117–1130.
- Jain, S., Schlick, T., 2017. F-RAG: Generating atomic coordinates from rna graphs by fragment assembly. *Journal of Molecular Biology* 429 (23), 3587–3605.
- Johnson, M.A., Pinto, B.M., 2002. Saturation transfer difference 1D-TOCSY experiments to map the topography of oligosaccharides recognized by a monoclonal antibody directed against the cell-wall polysaccharide of group A *Streptococcus*. *Journal of the American Chemical Society* 124, 15368–15374.
- Jones, D.T., 1999. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology* 292, 195–202.
- Jorgensen, W.L., Maxwell, D.S., Tirado-Rives, J., 1996. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society* 118, 11225–11236.
- Kabsch, W., Sander, C., 1983a. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.
- Kabsch, W., Sander, C., 1983b. How good are predictions of protein secondary structure? *FEBS Letters* 155, 179–182.
- Kc, D.B., 2016. Recent advances in sequence-based protein structure prediction. *Brief Bioinformatics* 18 (6), 1021–1032.
- Keefe, L.J., Sondek, J., Shortle, D., Lattman, E.E., 1993. The alpha aneurism: A structural motif revealed in an insertion mutant of staphylococcal nuclease. *Proceedings of the National Academy of Sciences of the United States of America* 90, 3275–3279.
- Kelly, S.M., Jess, T.J., Price, N.C., 2005. How to study proteins by circular dichroism. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* 1751, 119–139.
- Kelly, S.M., Price, N.C., 2000. The use of circular dichroism in the investigation of protein structure and function. *Current Protein & Peptide Science* 1, 349–384.
- Kendrew, J.C., Bodo, G., Dintzis, H.M., *et al.*, 1958. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature* 181, 662–666.
- Kendrew, J.C., Dickerson, R.E., Strandberg, B.E., *et al.*, 1960. Structure of myoglobin: A three-dimensional Fourier synthesis at 2 Å resolution. *Nature* 185, 422–427.
- Khateb, S., Weisman-Shomer, P., Herschko, I., *et al.*, 2004. Destabilization of tetraplex structures of the fragile X repeat sequence (CGG) n is mediated by homolog-conserved domains in three members of the hnRNP family. *Nucleic Acids Research* 32, 4145–4154.
- King, S.M., Johnson, W.C., 1999. Assigning secondary structure from protein coordinate data. *Proteins* 35, 313–320.
- Kjellberg, A., Nishida, T., Weintraub, A., Widmalm, G., 1998. NMR spectroscopy of ¹³C-enriched polysaccharides: Application of ¹³C–¹³C TOCSY to sugars of different configuration. *Magnetic Resonance in Chemistry* 36, 128–131.
- Kleywegt, G.J., Jones, T.A., 1996. Phi/psi-chology: Ramachandran revisited. *Structure* 4, 1395–1400.
- Konagurthu, A.S., Lesk, A.M., Allison, L., 2012. Minimum message length inference of secondary structure from protein coordinate data. *Bioinformatics* 28, i97–i105.
- Kotelchuck, D., Scheraga, H.A., 1969. The influence of short-range interactions on protein conformation. II. A model for predicting the alpha-helical regions of proteins. *Proceedings of the National Academy of Sciences of the United States of America* 62, 14–21.
- Kumar, P., Bansal, M., 2012. HELANAL-Plus: A web server for analysis of helix geometry in protein structures. *Journal of Biomolecular Structure and Dynamics* 30, 773–783.
- Kumar, P., Bansal, M., 2015a. Dissecting pi-helices: Sequence, structure and function. *FEBS Journal* 282, 4415–4432.
- Kumar, P., Bansal, M., 2015b. Identification of local variations within secondary structures of proteins. *Acta Crystallographica D Biological Crystallography* 71, 1077–1086.
- Kumar, P., Bansal, M., 2016. Structural and functional analyses of PolyProline-II helices in globular proteins. *Journal of Structural Biology* 196, 414–425.
- Kumar, P., Kailasam, S., Chakraborty, S., Bansal, M., 2014. MolBridge: A program for identifying nonbonded interactions in small molecules and biomolecular structures. *Journal of Applied Crystallography* 47, 1772–1776.
- Kumar, S., Bansal, M., 1998. Geometrical and sequence characteristics of alpha-helices in globular proteins. *Biophysical Journal* 75, 1935–1944.
- Kuntz, I.D., 1972. Protein folding. *Journal of the American Chemical Society* 94, 4009–4012.
- Kuttel, M.M., Stähle, J., Widmalm, G., 2016. CarbBuilder: Software for building molecular models of complex oligo- and polysaccharide structures. *Journal of Computational Chemistry* 37, 2098–2105.
- Labesse, G., Colloc'h, N., Pothier, J., Mornon, J.P., 1997. P-SEA: A new efficient assignment of secondary structure from C alpha trace of proteins. *Computer Applications in the Biosciences: CABIOS* 13, 291–295.
- Lack, S., Dulong, V., Pictou, L., *et al.*, 2007. High-resolution nuclear magnetic resonance spectroscopy studies of polysaccharides crosslinked by sodium trimetaphosphate: A proposal for the reaction mechanism. *Carbohydrate Research* 342, 943–953.
- Lacroix, L., Mergny, J.-L., Leroy, J.-L., Hélène, C., 1996. Inability of RNA to form the i-motif: Implications for triplex formation. *Biochemistry* 35, 8715–8722.

- Largy, E., Mergny, J.-L., Gabelica, V., 2016. Role of alkali metal ions in G-quadruplex nucleic acid structure and stability. *The Alkali Metal Ions: Their Role for Life*. pp. 203–258.
- Larocque, S., Brisson, J.-R., Thérissod, H., *et al.*, 2003. Structural characterization of the O-chain polysaccharide isolated from *Bordetella avium* ATCC 5086: Variation on a theme. *FEBS Letters* 535, 11–16.
- Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M., 1993. PROCHECK: A program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography* 26, 283–291.
- Lawson, C.L., Patwardhan, A., Baker, M.L., *et al.*, 2016. EMDatabank unified data resource for 3DEM. *Nucleic Acids Research* 44, D396–D403.
- Lee, C.H., Singla, A., Lee, Y., 2001. Biomedical applications of collagen. *International Journal of Pharmaceutics* 221, 1–22.
- Letunic, I., Doerks, T., Bork, P., 2015. SMART: Recent updates, new developments and status in 2015. *Nucleic Acids Research* 43, D257–D260.
- Levitt, M., Greer, J., 1977. Automatic identification of secondary structure in globular proteins. *Journal of Molecular Biology* 114, 181–239.
- Lewis, P.N., Go, N., Go, M., *et al.*, 1970. Helix probability profiles of denatured proteins and their correlation with native structures. *Proceedings of the National Academy of Sciences of the United States of America* 65, 810–815.
- Lewis, P.N., Momany, F.A., Scheraga, H.A., 1971. Folding of polypeptide chains in proteins: A proposed mechanism for folding. *Proceedings of the National Academy of Sciences of the United States of America* 68, 2293–2297.
- Lewis, P.N., Momany, F.A., Scheraga, H.A., 1973. Chain reversals in proteins. *Biochimica et Biophysica Acta* 303, 211–229.
- Lindauer, K., Bendic, C., Sühnel, J., 1996. HBExplore – A new tool for identifying and analysing hydrogen bonding patterns in biological macromolecules. *Computer Applications in the Biosciences: CABIOS* 12, 281–289.
- Linker, A., Evans, L.R., Impallomeni, G., 2001. The structure of a polysaccharide from infectious strains of *Burkholderia cepacia*. *Carbohydrate Research* 335, 45–54.
- Lin, N., Huang, J., Dufresne, A., 2012. Preparation, properties and applications of polysaccharide nanocrystals in advanced functional nanomaterials: A review. *Nanoscale* 4, 3274–3294.
- Liu, D., Balasubramanian, S., 2003. A Proton-fuelled DNA nanomachine. *Angewandte Chemie International Edition* 42, 5734–5736.
- Lo Conte, L., Ailey, B., Hubbard, T.J., *et al.*, 2000. SCOP: A structural classification of proteins database. *Nucleic Acids Research* 28, 257–259.
- Lodish, H., Berk, A., Zipursky, S.L., *et al.*, 2000. *Molecular Cell Biology*, fourth ed. National Center for Biotechnology Information's Bookshelf.
- Long, S.B., Campbell, E.B., Mackinnon, R., 2005a. Crystal structure of a mammalian voltage-dependent Shaker family K⁺ channel. *Science* 309, 897–903.
- Long, S.B., Campbell, E.B., Mackinnon, R., 2005b. Voltage sensor of Kv1.2: Structural basis of electromechanical coupling. *Science* 309, 903–908.
- Louis-Jeune, C., Andrade-Navarro, M.A., Perez-Iratxeta, C., 2012. Prediction of protein secondary structure from circular dichroism using theoretically derived spectra. *Proteins* 80, 374–381.
- Love, W.E., Klock, P.A., Lattman, E.E., *et al.*, 1972. The structures of lamprey and bloodworm hemoglobins in relation to their evolution and function. *Cold Spring Harbor Symposia on Quantitative Biology*. 349–357.
- Low, B., Baybutt, R., 1952. The pi-helix – A hydrogen bonded configuration of the polypeptide chain. *Journal of the American Chemical Society* 74, 5806–5807.
- Low, B.W., Grenville-Wells, H.J., 1953. Generalized mathematical relationships for polypeptide chain helices: The coordinates of the II helix. *Proceedings of the National Academy of Sciences of the United States of America* 39, 785–801.
- Luthy, R., Bowie, J.U., Eisenberg, D., 1992. Assessment of protein models with three-dimensional profiles. *Nature* 356, 83–85.
- Lu, X.-J., Bussemaker, H.J., Olson, W.K., 2015. DSSR: An integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Research* 43, e142.
- Lu, X.-J., Olson, W.K., 2008. 3DNA: A versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nature Protocols* 3, 1213–1227.
- MacArthur, M.W., Thornton, J.M., 1996. Deviations from planarity of the peptide bond in peptides and proteins. *Journal of Molecular Biology* 264, 1180–1195.
- Macdonald, J.R., Johnson Jr., W.C., 2001. Environmental features are important in determining protein secondary structure. *Protein Science* 10, 1172–1177.
- Macié, J.S., Chaves, L.S., Souza, B.W., *et al.*, 2008. Structural characterization of cold extracted fraction of soluble sulfated polysaccharide from red seaweed *Gracilaria birdiae*. *Carbohydrate Polymers* 71, 559–565.
- Maizels, N., Gray, L.T., 2013. The G4 genome. *PLOS Genetics* 9, e1003468.
- Majumdar, I., Krishna, S.S., Grishin, N.V., 2005. PALSSE: A program to delineate linear secondary structural elements from protein structures. *BMC Bioinformatics* 6, 202.
- Makarov, A.A., Esipova, N.G., Pankov, Y.A., *et al.*, 1975. A Conformational study of beta-melanocyte-stimulation hormone. *Biochemical and Biophysical Research Communications* 67, 1378–1383.
- Mandel-Gutfreund, Y., Margalit, H., Jernigan, R.L., Zhurkin, V.B., 1998. A role for CH... O interactions in protein-DNA recognition. *Journal of Molecular Biology* 277, 1129–1140.
- Mansiaux, Y., Joseph, A.P., Gelly, J.C., de Brevern, A.G., 2011. Assignment of PolyProline II conformation and analysis of sequence–structure relationship. *PLOS ONE* 6, e18401.
- Markham, N.R., Zuker, M., 2008. UNAFold: Software for nucleic acid folding and hybridization. *Methods in Molecular Biology* 453, 3–31.
- Marks, D.S., Hopf, T.A., Sander, C., 2012. Protein structure prediction from sequence variation. *Nature Biotechnology* 30, 1072–1080.
- Martin, J., Letellier, G., Marin, A., *et al.*, 2005. Protein secondary structure assignment revisited: A detailed analysis of different assignment methods. *BMC Structural Biology* 5, 17.
- Martin-Pastor, M., Bush, C.A., 1999. New strategy for the conformational analysis of carbohydrates based on NOE and ¹³C NMR coupling constants. Application to the flexible polysaccharide of *Streptococcus mitis* J22. *Biochemistry* 38, 8045–8055.
- Mathews, D.H., 2006. Revolutions in RNA secondary structure prediction. *Journal of Molecular Biology* 359, 526–532.
- Matsugami, A., Okuzumi, T., Uesugi, S., Katahira, M., 2003. Intramolecular higher order packing of parallel quadruplexes comprising a G: G: G tetrad and a G (: a): G (: a): G (: a): G heptad of GGA triplet repeat DNA. *Journal of Biological Chemistry* 278, 28147–28153.
- Mavridis, L., Janes, R.W., 2017. PDB2CD: A web-based application for the generation of circular dichroism spectra from protein atomic coordinates. *Bioinformatics* 33, 56–63.
- McDonald, I.K., Thornton, J.M., 1994. Satisfying hydrogen bonding potential in proteins. *Journal of Molecular Biology* 238, 777–793.
- Medlock, A.E., Dailey, T.A., Ross, T.A., *et al.*, 2007. A pi-helix switch selective for porphyrin deprotonation and product release in human ferrochelatase. *Journal of Molecular Biology* 373, 1006–1016.
- Michael, F.S., Szymanski, C.M., Li, J., *et al.*, 2002. The structures of the lipooligosaccharide and capsule polysaccharide of *Campylobacter jejuni* genome sequenced strain NCTC 11168. *FEBS Journal* 269, 5119–5136.
- Micsonai, A., Wien, F., Kernya, L., *et al.*, 2015. Accurate secondary structure prediction and fold recognition for circular dichroism spectroscopy. *Proceedings of the National Academy of Sciences of the United States of America* 112, E3095–E3103.
- Miles, A.J., Wallace, B.A., 2006. Synchrotron radiation circular dichroism spectroscopy of proteins and applications in structural and functional genomics. *Chemical Society Reviews* 35, 39–51.
- Milne, J.L., Borgnia, M.J., Bartschagi, A., *et al.*, 2013. Cryo-electron microscopy – A primer for the non-microscopist. *FEBS Journal* 280, 28–45.
- Mitchell, A., Chang, H.-Y., Daugherty, L., *et al.*, 2015. The InterPro protein families database: The classification resource after 15 years. *Nucleic Acids Research* 43, D213–D221.
- Miyake, M., Hori, S., Morizawa, Y., *et al.*, 2017. Collagen type IV alpha 1 (COL4A1) and collagen type XIII alpha 1 (COL13A1) produced in cancer cells promote tumor budding at the invasion front in human urothelial carcinoma of the bladder. *Oncotarget* 8, 36099–36114.
- Momen-Roknabadi, A., Sadeghi, M., Pezeshki, H., Marashi, S.A., 2008. Impact of residue accessible surface area on the prediction of protein secondary structures. *BMC Bioinformatics* 9, 357.
- Mount, D.W., 2004. *Bioinformatics: Sequence and Genome Analysis*. New York: Cold Spring Harbor Laboratory Press.
- Mulder, G.J., 1839. On the composition of some animal substances. *Journal für Praktische Chemie* 16, 15.

- Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology* 247, 536–540.
- Muzzarelli, R.A., 1973. Natural chelating polymers; alginic acid, chitin and chitosan. *Natural Chelating Polymers; Alginic Acid, Chitin and Chitosan*. Pergamon Press.
- Nahmany, A., Strino, F., Rosen, J., *et al.*, 2005. The use of a genetic algorithm search for molecular mechanics (MM3)-based conformational analysis of oligosaccharides. *Carbohydrate Research* 340, 1059–1064.
- Narwani, T.J., Santuz, H., Shinada, N., *et al.*, 2017. Recent advances on polyproline II. *Amino Acids* 49, 705–713.
- Nelson, D.L., Lehninger, A.L., Cox, M.M., 2008. *Lehninger Principles of Biochemistry*. Macmillan.
- Nonogaki, H., Gee, O.H., Bradford, K.J., 2000. A germination-specific endo- β -mannanase gene is expressed in the micropylar endosperm cap of tomato seeds. *Plant Physiology* 123, 1235–1246.
- Novotny, M., Kleywegt, G.J., 2005. A survey of left-handed helices in protein structures. *Journal of Molecular Biology* 347, 231–241.
- Orengo, C.A., Michie, A.D., Jones, S., *et al.*, 1997. CATH – A hierarchic classification of protein domain structures. *Structure* 5, 1093–1108.
- Orengo, C.A., Pearl, F.M., Thornton, J.M., 2003. The CATH domain structure database. *Methods of Biochemical Analysis* 44, 249–271.
- Pal, L., Basu, G., 1999. Novel protein structural motifs containing two-turn and longer 3(10)-helices. *Protein Engineering* 12, 811–814.
- Panigrahi, S.K., Desiraju, G.R., 2007. Strong and weak hydrogen bonds in the protein–ligand interface. *Proteins: Structure, Function, and Bioinformatics* 67, 128–141.
- Parenteau-Bareil, R., Gauvin, R., Berthod, F., 2010. Collagen-based biomaterials for tissue engineering applications. *Materials* 3, 1863–1887.
- Parisien, M., Major, F., 2008. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature* 452, 51–55.
- Pauling, L., 1960. *The Nature of the Chemical Bond*. Ithaca, NY: Cornell University Press.
- Pauling, L., Corey, R.B., 1951. The pleated sheet, a new layer configuration of polypeptide chains. *Proceedings of the National Academy of Sciences of the United States of America* 37, 251–256.
- Pauling, L., Corey, R.B., Branson, H.R., 1951. The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America* 37, 205–211.
- Pearl, F.M., Bennett, C.F., Bray, J.E., *et al.*, 2003. The CATH database: An extended protein family resource for structural and functional genomics. *Nucleic Acids Research* 31, 452–455.
- Perez-Iratxeta, C., Andrade-Navarro, M.A., 2008. K2D2: Estimation of protein secondary structure from circular dichroism spectra. *BMC Structural Biology* 8, 25.
- Pérez, S., Tubiana, T., Imbert, A., Baaden, M., 2014. Three-dimensional representations of complex carbohydrates and polysaccharides–SweetUnityMol: A video game-based computer graphic software. *Glycobiology* 25, 483–491.
- Perutz, M.F., 1951. New x-ray evidence on the configuration of polypeptide chains. *Nature* 167, 1053–1054.
- Peters, T., Meyer, B., Stuike-Prill, R., *et al.*, 1993. A Monte Carlo method for conformational analysis of saccharides. *Carbohydrate Research* 238, 49–73.
- Petrey, D., Honig, B., 2005. Protein structure prediction: Inroads to biology. *Molecular Cell* 20, 811–819.
- Powell, D.R., 2016. *Review of X-Ray Crystallography*. ACS Publications.
- Prockop, D.J., Kivirikko, K.I., Tuderman, L., Guzman, N.A., 1979. The biosynthesis of collagen and its disorders. *New England Journal of Medicine* 301, 77–85.
- Prothero, J.W., 1966. Correlation between the distribution of amino acids and alpha helices. *Biophysical Journal* 6, 367–370.
- Punta, M., Coghill, P.C., Eberhardt, R.Y., *et al.*, 2012. The Pfam protein families database. *Nucleic Acids Research* 40, D290–D301.
- Purcell, E.M., Torrey, H.C., Pound, R.V., 1946. Resonance absorption by nuclear magnetic moments in a solid. *Physical Review* 69, 37.
- Ramachandran, G.N., Kartha, G., 1954. Structure of collagen. *Nature* 174, 269–270.
- Ramachandran, G.N., Kartha, G., 1955. Structure of collagen. *Nature* 176, 593–595.
- Ramachandran, G.N., Ramakrishnan, C., Sasisekharan, V., 1963. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology* 7, 95–99.
- Ramachandran, G.N., Sasisekharan, V., 1961. Structure of collagen. *Nature* 190, 1004–1005.
- Ramachandran, G.N., Sasisekharan, V., 1968. Conformation of polypeptides and proteins. *Advances in Protein Chemistry* 23, 283–438.
- Ramachandran, G.N., Venkatachalam, C.M., Krimm, S., 1966. Stereochemical criteria for polypeptide and protein chain conformations. 3. Helical and hydrogen-bonded polypeptide chains. *Biophysical Journal* 6, 849–872.
- Ramakrishnan, C., Ramachandran, G.N., 1965. Stereochemical criteria for polypeptide and protein chain conformations. II. Allowed conformations for a pair of peptide units. *Biophysical Journal* 5, 909–933.
- Ramberg, J.E., Nelson, E.D., Sinnott, R.A., 2010. Immunomodulatory dietary polysaccharides: A systematic review of the literature. *Nutrition Journal* 9, 54.
- Rao, S.T., Rossmann, M.G., 1973. Comparison of super-secondary structures in proteins. *Journal of Molecular Biology* 76, 241–256.
- Redfern, O., Grant, A., Maibaum, M., Orengo, C., 2005. Survey of current protein family databases and their application in comparative, structural and functional genomics. *Journal of Chromatography. B, Analytical Technologies in the Biomedical and Life Sciences* 815, 97–107.
- Ren, J., Rastegari, B., Condon, A., Hoos, H.H., 2005. HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. *RNA* 11, 1494–1504.
- Reuter, J.S., Mathews, D.H., 2010. RNAstructure: Software for RNA secondary structure prediction and analysis. *BMC Bioinformatics* 11, 129.
- Rhodes, D., Lipps, H.J., 2015. G-quadruplexes and their regulatory roles in biology. *Nucleic Acids Research* 43, 8627–8637.
- Rice, C., Skordalakes, E., 2016. Structure and function of the telomeric CST complex. *Computational and Structural Biotechnology Journal* 14, 161–167.
- Rich, A., 1960. A hybrid helix containing both deoxyribose and ribose polynucleotides and its relation to the transfer of information between the nucleic acids. *Proceedings of the National Academy of Sciences* 46, 1044–1053.
- Rich, A., Crick, F.H., 1955. The structure of collagen. *Nature* 176, 915–916.
- Rich, A., Davies, D.R., 1956. A new two stranded helical structure: Polyadenylic acid and polyuridylic acid. *Journal of the American Chemical Society* 78, 3548–3549.
- Richards, F.M., Kundrot, C.E., 1988. Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. *Proteins* 3, 71–84.
- Richardson, J.S., 1977. beta-Sheet topology and the relatedness of proteins. *Nature* 268, 495–500.
- Richardson, J.S., 1981. The anatomy and taxonomy of protein structure. *Advances in Protein Chemistry* 34, 167–339.
- Richardson, J.S., Getzoff, E.D., Richardson, D.C., 1978. The beta bulge: A common small unit of nonrepetitive protein structure. *Proceedings of the National Academy of Sciences of the United States of America* 75, 2574–2578.
- Rietveld, K., Van Poelgeest, R., Pleij, C.W., *et al.*, 1982. The tRNA-Uke structure at the 3' terminus of turnip yellow mosaic virus RNA. Differences and similarities with canonical tRNA. *Nucleic Acids Research* 10, 1929–1946.
- Risitano, A., Fox, K.R., 2004. Influence of loop size on the stability of intramolecular DNA quadruplexes. *Nucleic Acids Research* 32, 2598–2606.
- Rivas, E., Eddy, S.R., 1999. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology* 285, 2053–2068.
- Rose, G.D., Winters, R.H., Wettlaufer, D.B., 1976. A testable model for protein folding. *FEBS Letters* 63, 10–16.
- Ross, P., Weinhouse, H., Aloni, Y., *et al.*, 1987. Regulation of cellulose synthesis in *Acetobacter xylinum* by cyclic diguanylic acid. *Nature* 325, 279–281.
- Rost, B., 2001. Review: Protein secondary structure prediction continues to rise. *Journal of Structural Biology* 134, 204–218.
- Ruan, J., Stormo, G.D., Zhang, W., 2004. An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics* 20, 58–66.
- Ruska, H., 1941. Über ein neues bei der bakterioophagen Lyse auftretendes Formelement. *Naturwissenschaften* 29, 367–368.
- Ruska, H., Borries, B.V., Ruska, E., 1939. Die Bedeutung der übermikroskopie für die Virusforschung. *Archives of Virology* 1, 155–169.
- Ruskin, R.S., Yu, Z., Grigorieff, N., 2013. Quantitative characterization of electron detectors for transmission electron microscopy. *Journal of Structural Biology* 184, 385–393.
- Sanghi, R., Bhattacharya, B., Singh, V., 2007. Seed gum polysaccharides and their grafted co-polymers for the effective coagulation of textile dye solutions. *Reactive and Functional Polymers* 67, 495–502.

- Sankaramakrishnan, R., Vishveshwara, S., 1990. Conformational studies on peptides with proline in the right-handed α -helical region. *Biopolymers* 30, 287–298.
- Sankaramakrishnan, R., Vishveshwara, S., 1992. Geometry of proline-containing α -helices in proteins. *International Journal of Peptide and Protein Research* 39, 356–363.
- Sasisekharan, V., 1962. Stereochemical criteria for polypeptide and protein structures. In: Ramanathan, N. (Ed.), *Collagen*. New York: Wiley & Sons, pp. 39–78.
- Sayle, R.A., Milner-White, E.J., 1995. RASMOL: Biomolecular graphics for all. *Trends in Biochemical Sciences* 20, 374.
- Schiffer, M., Edmundson, A.B., 1967. Use of helical wheels to represent the structures of proteins and to identify segments with helical potential. *Biophysical Journal* 7, 121–135.
- Schrodinger, L.L.C., 2010. The PyMOL Molecular Graphics System, Version 1.3r1.
- Schroeder, S.J., 2009. Advances in RNA structure prediction from sequence: New tools for generating hypotheses about viral RNA structure-function relationships. *Journal of Virology* 83, 6326–6334.
- Seetin, M.G., Mathews, D.H., 2012. TurboKnot: Rapid prediction of conserved RNA secondary structures including pseudoknots. *Bioinformatics* 28, 792–798.
- Sekiguchi, S., Miura, Y., Kaneko, H., *et al.*, 1994. Molecular weight dependency of antimicrobial activity by chitosan oligomers. *Food Hydrocolloids*, 71–76.
- Shapiro, B.A., Yingling, Y.G., Kasprzak, W., Bindewald, E., 2007. Bridging the gap in RNA structure prediction. *Current Opinion in Structural Biology* 17, 157–165.
- Sheikh, Z., Qureshi, J., Alshahrani, A.M., *et al.*, 2017. Collagen based barrier membranes for periodontal guided bone regeneration applications. *Odontology* 105, 1–12.
- Shelar, A., Kumar, P., Bansal, M., 2013. Defining α -helix geometry by C α atom trace vs $((\phi)-\psi)$ torsion angles: A comparative analyses. In: Bansal, M., Srinivasan, N. (Eds.), *Biomolecular Forms and Functions: A Celebration of 50 Years of the Ramachandran Map*. Bengaluru: World Scientific, pp. 116–127.
- Sheng, Q., Neaverson, J.C., Mahmoud, T., *et al.*, 2017. Identification of new DNA i-motif binding ligands through a fluorescent intercalator displacement assay. *Organic & Biomolecular Chemistry* 15 (27), 5669–5673.
- Shi, Y., 2014. A glimpse of structural biology through X-ray crystallography. *Cell* 159, 995–1014.
- Simonsson, T., 2001. G-quadruplex DNA structures—variations on a theme. *Biological Chemistry* 382, 621–628.
- Singh, S.K., Babu, M.M., Balam, P., 2003. Registering α -helices and β -strands using backbone C-H... O interactions. *Proteins: Structure, Function, and Bioinformatics* 51, 167–171.
- Sinha, S., Astani, A., Ghosh, T., *et al.*, 2010. Polysaccharides from *Sargassum tenerrimum*: Structural features, chemical modification and anti-viral activity. *Phytochemistry* 71, 235–242.
- Sklénar, H., Etchebest, C., Lavery, R., 1989. Describing protein structure: A general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins* 6, 46–60.
- Smyth, M.S., Martin, J.H., 2000. x ray crystallography. *Molecular Pathology* 53, 8–14.
- Son, S., Nam, J., Kim, J., *et al.*, 2014. i-motif-driven Au nanomachines in programmed siRNA delivery for gene-silencing and photothermal ablation. *ACS Nano* 8, 5574–5584.
- Sreerama, N., Woody, R.W., 2000. Estimation of protein secondary structure from circular dichroism spectra: Comparison of CONTIN, SELCON, and CDSSTR methods with an expanded reference set. *Analytical Biochemistry* 287, 252–260.
- Srinivasan, R., Rose, G.D., 1999. A physical basis for protein secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 96, 14258–14263.
- Srivastava, R., Kulshreshtha, D.K., 1989. Bioactive polysaccharides from plants. *Phytochemistry* 28, 2877–2883.
- Staple, D.W., Butcher, S.E., 2005. Pseudoknots: RNA structures with diverse functions. *PLOS Biology* 3, e213.
- Stapley, B.J., Creamer, T.P., 1999. A survey of left-handed polypyrrolone II helices. *Protein Science* 8, 587–595.
- Stenutz, R., Jansson, P.-E., Widmalm, G., 1998. Computer-assisted structural analysis of oligo- and polysaccharides: An extension of CASPER to multibranched structures. *Carbohydrate Research* 306, 11–17.
- Sternberg, M.J., Thornton, J.M., 1976. On the conformation of proteins: The handedness of the beta-strand- α -helix-beta-strand unit. *Journal of Molecular Biology* 105, 367–382.
- Sternberg, M.J., Thornton, J.M., 1977a. On the conformation of proteins: An analysis of beta-pleated sheets. *Journal of Molecular Biology* 110, 285–296.
- Sternberg, M.J., Thornton, J.M., 1977b. On the conformation of proteins: The handedness of the connection between parallel beta-strands. *Journal of Molecular Biology* 110, 269–283.
- Su, L., Chen, L., Egli, M., *et al.*, 1999. Minor groove RNA triplex in the crystal structure of a ribosomal frameshifting viral pseudoknot. *Nature Structural & Molecular Biology* 6, 285–292.
- Sundquist, W.I., Klug, A., 1989. Telomeric DNA dimerizes by formation of guanine tetrads between hairpin loops. *Nature* 342, 825–829.
- Svensson, B., Jespersen, H., Sierks, M., MacGregor, E., 1989. Sequence homology between putative raw-starch binding domains from different starch-degrading enzymes. *Biochemical Journal* 264, 309.
- Szewczak, A.A., Ortoleva-Donnelly, L., Ryder, S.P., *et al.*, 1998. A minor groove RNA triple helix within the catalytic core of a group I intron. *Nature Structural & Molecular Biology* 5, 1037–1042.
- Tan, Z., Fu, Y., Sharma, G., Mathews, D.H., 2017. TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Research* 45 (20), 11570–11581.
- Taylor, H.S., 1941. Large molecules through atomic spectacles. *Proceedings of the American Philosophical Society* 85, 1–12.
- Taylor, W.R., 2001. Defining linear segments in protein structure. *Journal of Molecular Biology* 310, 1135–1150.
- Terwilliger, T.C., 2010. Rapid model building of α -helices in electron-density maps. *Acta Crystallographica D Biological Crystallography* 66, 268–275.
- Tiffany, M.L., Krimm, S., 1968. New chain conformations of poly(glutamic acid) and polylysine. *Biopolymers* 6, 1379–1382.
- Tina, K., Bhadra, R., Srinivasan, N., 2007. PIC: Protein interactions calculator. *Nucleic Acids Research* 35, W473–W476.
- Tincer, G., Bayyurt, B., Arica, Y., Gürsel, İ., 2015. Chitosan polysaccharide suppress toll like receptor dependent immune response [Çitosan polisakaridi toll benzeri reseptöre bağımlı bağışıklık yanıtını baskılar]. *Turkish Journal of Immunology* 3, 15–20.
- Tinoco, I., Bustamante, C., 1999. How RNA folds. *Journal of Molecular Biology* 293, 271–281.
- Tiwari, A., Panigrahi, S.K., 2007. HBA: A complete package for analysing strong and weak hydrogen bonds in macromolecular crystal structures. *In Silico Biology* 9, 651–661.
- Todd, A.K., Johnston, M., Neidle, S., 2005. Highly prevalent putative quadruplex sequence motifs in human DNA. *Nucleic Acids Res* 33, 2901–2907.
- Toniolo, C., Benedetti, E., 1991. The polypeptide 310-helix. *Trends in Biochemical Sciences* 16, 350–353.
- Toukach, F.V., Ananikov, V.P., 2013. Recent advances in computational predictions of NMR parameters for the structure elucidation of carbohydrates: Methods and limitations. *Chemical Society Reviews* 42, 8376–8415.
- Touw, W.G., Baakman, C., Black, J., *et al.*, 2015. A series of PDB-related databanks for everyday needs. *Nucleic Acids Research* 43, D364–D368.
- Van Der Spoel, D., Lindahl, E., Hess, B., *et al.*, 2005. GROMACS: Fast, flexible, and free. *Journal of Computational Chemistry* 26, 1701–1718.
- Venkatachalam, C.M., 1968. Stereochemical criteria for polypeptides and proteins. V. Conformation of a system of three linked peptide units. *Biopolymers* 6, 1425–1436.
- Vermeire, J., Vanbillemont, G., Witkowski, W., Verhasselt, B., 2011. The Nef-infectivity enigma: Mechanisms of enhanced lentiviral infection. *Current HIV Research* 9, 474–489.
- Wallace, B.A., 2009. Protein characterisation by synchrotron radiation circular dichroism spectroscopy. *Quarterly Reviews of Biophysics* 42, 317–370.
- Wang, A., Quigley, G.J., Kolpak, F.J., *et al.*, 1979. Molecular structure of a left-handed double helical DNA fragment at atomic resolution. *Nature* 282, 680–686.
- Wang, Z.-G., Elbaz, J., Willner, I., 2010. DNA machines: Bipodal walker and stepper. *Nano Letters* 11, 304–309.
- Watson, J.D., Crick, F.H., 1953. Molecular structure of nucleic acids. *Nature* 171, 737–738.
- Weaver, T.M., 2000. The pi-helix translates structure into function. *Protein Science* 9, 201–206.
- Whitmore, L., Miles, A.J., Mavridis, L., *et al.*, 2017. PCDDb: New developments at the protein circular dichroism data bank. *Nucleic Acids Research* 45, D303–D307.

- Whitmore, L., Wallace, B.A., 2004. DICHROWEB, an online server for protein secondary structure analyses from circular dichroism spectroscopic data. *Nucleic Acids Research* 32, W668–W673.
- Wiedemann, C., Bellstedt, P., Grolach, M., 2013. CAPITO – A web server-based analysis and plotting tool for circular dichroism data. *Bioinformatics* 29, 1750–1757.
- Williams, C.J., Headd, J.J., Moriarty, N.W., *et al.*, 2017. MolProbity: More and better reference data for improved all-atom structure validation. *Protein Science* 27 (1), 293–315.
- Wilm, A., Linnenbrink, K., Steger, G., 2008. ConStruct: Improved construction of RNA consensus structures. *BMC Bioinformatics* 9, 219.
- Wilson, D.N., Nierhaus, K.H., 2003. The ribosome through the looking glass. *Angewandte Chemie International Edition* 42, 3464–3486.
- Wing, R., Drew, H., Takano, T., *et al.*, 1980. Crystal structure analysis of a complete turn of B-DNA. *Nature* 287, 755–758.
- Wohlert, J., Berglund, L.A., 2011. A coarse-grained model for molecular dynamics simulations of native cellulose. *Journal of Chemical Theory and Computation* 7, 753–760.
- Woody, R.W., 1992. Circular dichroism and conformation of unordered polypeptides. *Advanced Biophysical Chemistry* 2, 37–79.
- Wormald, M.R., Petrescu, A.J., Pao, Y.-L., *et al.*, 2002. Conformational studies of oligosaccharides and glycopeptides: Complementarity of NMR, X-ray crystallography, and molecular modelling. *Chemical Reviews* 102, 371–386.
- Wright, W.E., Tesmer, V.M., Huffman, K.E., *et al.*, 1997. Normal human chromosomes have long G-rich telomeric overhangs at one end. *Genes & Development* 11, 2801–2809.
- Xia, J., Daly, R.P., Chuang, F.-C., *et al.*, 2007. Sugar folding: A novel structural prediction tool for oligosaccharides and polysaccharides 2. *Journal of Chemical Theory and Computation* 3, 1629–1643.
- Xu, H., Di Antonio, M., McKinney, S., *et al.*, 2017. CX-5461 is a DNA G-quadruplex stabilizer with selective lethality in BRCA1/2 deficient tumours. *Nature Communications* 8, 14432.
- Yamasaki, R., Bacon, B., 1991. Three-dimensional structural analysis of the group B polysaccharide of *Neisseria meningitidis* 6275 by two-dimensional NMR: The polysaccharide is suggested to exist in helical conformations in solution. *Biochemistry* 30, 851–857.
- Yang, X.-M., Yu, W., Ou, Z.-P., *et al.*, 2009. Antioxidant and immunity activity of water extract and crude polysaccharide from *Ficus carica* L. fruit. *Plant Foods for Human Nutrition* 64, 167–173.
- Yang, Y., Gao, J., Wang, J., *et al.*, 2016. Sixty-five years of the long march in protein secondary structure prediction: The final stretch? *Brief Bioinformatics*. doi:10.1093/bib/bbw129.
- Yohannan, S., Faham, S., Yang, D., *et al.*, 2004. The evolution of transmembrane helix kinks and the structural diversity of G protein-coupled receptors. *Proceedings of the National Academy of Sciences of the United States of America* 101, 959–963.
- Yongxu, S., Jicheng, L., 2008. Structural characterization of a water-soluble polysaccharide from the roots of *Codonopsis pilosula* and its immunity activity. *International Journal of Biological Macromolecules* 43, 279–282.
- Zhang, T., Zheng, Y., Cosgrove, D.J., 2016. Spatial organization of cellulose microfibrils and matrix polysaccharides in primary plant cell walls as imaged by multichannel atomic force microscopy. *The Plant Journal* 85, 179–192.
- Zhong, L., Johnson Jr., W.C., 1992. Environment affects amino acid preference for secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 89, 4462–4465.
- Zimmerman, S.S., Scheraga, H.A., 1977. Local interactions in bends of proteins. *Proceedings of the National Academy of Sciences of the United States of America* 74, 4126–4129.
- Zondlo, N.J., 2010. Non-covalent interactions: Fold globally, bond locally. *Nature Chemical Biology* 6, 567–568.
- Zong, A., Cao, H., Wang, F., 2012. Anticancer polysaccharides from natural resources: A review of recent research. *Carbohydrate Polymers* 90, 1395–1410.
- Zuker, M., Stiegler, P., 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research* 9, 133–148.
- Zuker, M., Sankoff, D., 1984. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology* 46, 591–621.

Further Reading

- Almond, A., Sheehan, J.K., 2003. Predicting the molecular shape of polysaccharides from dynamic interactions with water. *Glycobiology* 13, 254–264.
- Caffall, K.H., Mohnen, D., 2009. The structure, function, and biosynthesis of plant cell wall pectic polysaccharides. *Carbohydrate Research* 344, 1879–1900.
- Cammas, A., Millevoi, S., 2017. RNA G-quadruplexes: Emerging mechanisms in disease. *Nucleic Acids Research* 45 (4), 1584–1595.
- Hänsel-Hertsch, R., Di Antonio, M., Balasubramanian, S., 2017. DNA G-quadruplexes in the human genome: Detection, functions and therapeutic potential. *Nature Reviews Molecular Cell Biology* 18 (5), 279–284.
- Jacobs, T.M., Williams, B., Williams, T., *et al.*, 2016. Design of structurally distinct proteins using strategies inspired by evolution. *Science* 352 (6286), 687–690.
- Jansson, P.E., Kenne, L., Widmalm, G., 1989. Computer-assisted structural analysis of polysaccharides with an extended version of CASPER using ¹H- and ¹³C NMR data. *Carbohydrate Research* 188, 169–191.
- Kelley, L.A., Sternberg, M.J., 2009. Protein structure prediction on the Web: A case study using the Phyre server. *Nature Protocols* 4 (3), 363–371.
- Laing, C., Schlick, T., 2011. Computational approaches to RNA structure prediction, analysis, and design. *Current Opinion in Structural Biology* 21 (3), 306–318.
- Lee, G., Nowak, G., Jaroniec, J., Zhang, Q., Marszalec, P.E., 2004. Molecular dynamics simulations of forced conformational transitions in 1,6-linked polysaccharides. *Biophysical Journal* 87, 1456–1465.
- Low, J.T., Weeks, K.M., 2010. SHAPE-directed RNA secondary structure prediction. *Methods* 52 (2), 150–158.
- Mathews, D.H., Disney, M.D., Childs, J.L., 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 101 (19), 7287–7292.
- Miao, Z., Adamiak, R.W., Antczak, M., *et al.*, 2017. RNA-Puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA* 23 (5), 655–672.
- Miao, Z., Westhof, E., 2017. RNA structure: Advances and assessment of 3D structure prediction. *Annual Review of Biophysics* 46, 483–503.
- Moult, J., 2005. A decade of CASP: Progress, bottlenecks and prognosis in protein structure prediction. *Current Opinion in Structural Biology* 15 (3), 285–289.
- Rohs, R., Jin, X., West, S.M., *et al.*, 2010. Origins of specificity in protein-DNA recognition. *Annual Review of Biochemistry* 79, 233–269.
- Sarkar, A., Perez, A., 2012. PolySac3BD: An annotated data base for 3 dimensional structures of polysaccharides. *BMC Bioinformatics* 13, 302.
- Woelfson, D.N., 2017. Coiled-coil design: Updated and upgraded. In: Parry, D.A.D., Squire, J.M. (Eds.), *Fibrous Proteins: Structures and Mechanisms*. Springer International Publishing, pp. 35–61.

Relevant Websites

- <http://people.cryst.bbk.ac.uk/~ubcg25a/>
Bonnie Ann Wallace.
- <http://pfam.xfam.org/>
EMBL-EBI.
- <http://www.emdatabank.org/>
EMDatabank.
- https://web.expasy.org/docs/swiss-prot_guideline.html
Expasy.

<http://www.glycam.com/>
 GLYCAM Web.
<https://rna.urmc.rochester.edu/index.html>
 Mathews Lab.
<http://nucleix.mbu.iisc.ernet.in/>
 MBU.
<http://ndbserver.rutgers.edu/>
 ndb Nucleic Acid Database.
<http://www.ebi.ac.uk/pdbe/pisa/>
 Protein Data Bank.
<http://kinemage.biochem.duke.edu/teaching/anatax/>
 The Anatomy and Taxonomy of Protein Structure.
<http://eddylab.org/>
 The Eddy/Rivas Laboratory.
<http://rna.tbi.univie.ac.at/>
 ViennaRNA Web Services.
<https://zhanglab.ccmb.med.umich.edu/>
 Zhang Lab.

Biographical Sketch



Prasun Kumar received a Bachelor of Technology degree in Bioinformatics from SASTRA University. For his Ph.D. in molecular biophysics, he worked with Prof. Manju Bansal at Molecular Biophysics Unit, Indian Institute of Science. The main thrust of his graduate studies was to develop algorithms for the identification and analyses of various secondary structure elements in protein structures. His work on comprehensive *in-silico* analyses of commonly occurring, as well as less frequently observed helices like π and PolyProline-II helices in globular proteins provided more insights into the sequence-structure relationships. He has also worked on structural analyses of nucleic acids and contributed to the modifications of in-house programs NUPARM as well as NUC-GEN. Prasun is now a postdoctoral research associate in Prof. Derek N. Woolfson's group at University of Bristol, United Kingdom, where he is working on the development of algorithms and *de novo* design of coiled-coils with a focus on probing non-covalent interactions within these.



Swagata Halder, received a Master of Science degree in Chemistry from Bengal Engineering and Science University, Shibpur. She has done her Ph.D in Molecular Biophysics under the supervision of Prof. Chaitali Mukhopadhyay, Department of Chemistry in University of Calcutta. Her research work was based on "Effect of Glycosylation on the Protein Structure and Dynamics: Insights from Molecular Modelling". She has modelled and simulated various glycosylated proteins like legume lectin Soybean agglutinin (SBA) and some antifreeze proteins like Ocean Pout during her graduate research. The main focus of the study was to predict the atomistic details of protein-carbohydrate interactions which play important roles in molecular mechanism of some biologically relevant phenomenon. Swagata is now a postdoctoral fellow in Prof. Manju Bansal's research group at Indian Institute of Science, India. Her current research is on identification and characterization of RNA thermometers in bacteria by molecular dynamic simulations. She is also working on aggregation property of some disease related proteins in presence of DNA by Replica Exchange MD techniques.



Professor Manju Bansal carried out her doctoral research under the guidance of Prof G N Ramachandran at the Indian Institute of Science, Bangalore and worked on the triple helical structure of fibrous protein collagen. She received her PhD degree in 1977 and joined the faculty of IISc in 1982, where she is currently an INSA Senior Scientist and J.C. Bose National Fellow. She has been an Alexander von Humboldt Fellow at EMBL Heidelberg and Frei Univ Berlin, and a Senior Fulbright Fellow at UCSF, San Francisco. She was the founder-Director of the Institute of Bioinformatics and Applied Biotechnology, Bangalore and represents India on the Advisory Board of wwPDB. She has published more than 120 papers in International journals and mentored about 60 doctoral and graduate students. Dr. Bansal's research interests have primarily focused on relating protein and DNA sequences to their structures and function, by developing new concepts, computational algorithms and tools, particularly in the area of DNA promoter architecture. Lab URL: <http://nucleix.iisc.ac.in/>.

Engineering of Supramolecular RNA Structures

Effirul I Ramlan, Malaysia Genome Institute (MGI-NIBM), Kajang, Malaysia and University of Malaya, Kuala Lumpur, Malaysia
Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Bangi, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Supramolecular RNA Structures

Proteins are responsible for carrying out vital cellular functions achieved through the combination of self-assembly and conformational dynamics. The structural variability of protein (complex three-dimensional folded structures) underlies its ability in mediating almost all functions in living cells. With the discovery of small regulatory RNA (sRNAs) (Zamore and Haley, 2005) and the discovery of the catalytic ability of ribozymes (Altman, 1990), RNA emerges as a more versatile molecule than previously perceived. Similar to protein, RNA structure determines its function, but unlike protein, the secondary structure of RNA molecule provides more information regarding its tertiary folding; thus, promoting rational design of programmable and functional RNA supramolecular structures to be constructed (i.e., nanostructures of the size of 10 nm in dimension (Goodsell, 2004)).

The discovery of short RNAs' (which is 21–25 nt in length) role in regulating gene expression (Couzin, 2002) has sparked a strong interest in RNA molecules. In RNA interference, a short interfering RNA molecule (siRNA) forms base pair with a target region in mRNA, allowing a protein complex called RNA-induced silencing complex or (RISC) to attach and cleaves the target region (Hannon, 2002; Petersen *et al.*, 2006). Another short RNAs called microRNA (miRNA), generated from an enzyme named Dicer that cleaved non-coding RNAs (i.e., RNA that do not code protein), binds imperfectly with the target region in mRNA (forming a bulge) to prevent the translation machinery from accessing this target region (Zamore and Haley, 2005). In addition to these short RNAs, another complex folded RNA domain called riboswitches also play a role in gene regulation. Riboswitches sense the presence of specific metabolites and harness their conformational switching to activate the gene-control mechanisms in preventing the production of protein (Mandal and Breaker, 2004; Nudler and Mironov, 2004).

RNA nanotechnology focuses on the engineering of functional and customizable supramolecular RNA structures as molecular devices utilizing the self-assembly principle of nucleic acids, reviewed in Jaeger and Chworos (2006); Guo (2010); Shu *et al.* (2014); Li *et al.* (2015); Jasinski *et al.* (2017). Advances in RNA nanotechnology have enabled the development of various RNA nanoparticles in nanomedicine, detailed explicitly in Jasinski *et al.* (2017). RNA nanoparticles have been developed as an alternate platform for targeted therapeutic delivery mechanism in the form of RNA aptamers (Esposito *et al.*, 2011; Shigdar *et al.*, 2011; Rockey *et al.*, 2011; Guo *et al.*, 2012; Thiel *et al.*, 2012; Kang and Lee, 2012; Zhou *et al.*, 2012; Sundaram *et al.*, 2013), as Ribozymes-based therapeutic strategy (Mulhbachter *et al.*, 2010; Shu *et al.*, 2011), as gene expression regulator through utilization of various Riboswitches (Suess and Weigand, 2008; Serganov, 2009; Berens *et al.*, 2015) and RNA interference therapy (Haque *et al.*, 2012; Ye *et al.*, 2012; Shu *et al.*, 2015; Lee *et al.*, 2015; Roh *et al.*, 2016).

Instead of mimicking the methodologies commonly associated with synthetic DNA nanostructures (commonly known as "DNA Origami"), RNA nanostructures are able to follow a more natural suprastructures formation. This can be achieved through the multi-self-assemblies of natural occurring RNA motifs (i.e., optimization of kinetic foldings and thermodynamic requirement based on combination of existing RNA motifs). The conceptualization of the desired nanostructures is rather straight-forward since the design will be based on a dictionary (or database) of well-characterized RNA motifs. This tutorial highlights the various methodologies for constructing supra-molecular RNAs and presents a generic computational protocols to support the in-silico construction.

Properties of Ribonucleic Acids

Nucleic acids are macromolecules that play an important role as information carriers in cells. Two types occur, ribonucleic acids (RNA) and de-oxyribonucleic acids (DNA), which are named after the structure of a sugar component in these molecules. RNA and DNA are typically long linear polymers that consist of a large number of monomers taken from a set of four different nucleotides (C, G, A, and T or U for RNA). The sequential order in which these bases are interlinked in the nucleic acid molecule represents information. RNA has the task of transmitting information within the cell, while DNA transmits information from generation to generation. The genetic information is encoded in a dimer of two complementary nucleotide chains ('single-stranded' DNA or ssDNA) which upon self-assembly assumes the well known double-helical structure ('double-stranded' DNA or dsDNA). DNA is well suited as carrier of genetic information because of its energy degeneracy with respect to the sequential order of the nucleotides. The properties of RNA molecules are more dependent on the sequence of nucleotides and as a consequence RNA takes on additional roles in the cell aside from representing information.

Within the RNA hierarchical framework, stacking and Watson-Crick base pairing between complementary nucleotides (canonical base pairing) trigger folding and assembly of RNA primary sequences into secondary structures (hairpin loops, bulges, internal loops and multiloops) as depicted in Fig. 1. The hydroxyl group (-OH) at the 2'-carbon in ribose is not present in deoxyribose (Berg *et al.*, 2003). The consequence of this structural difference is twofold. DNA is considerably more stable against hydrolysis and forms more compact double strands, while RNA has more conformational flexibility (Bloomfield *et al.*, 2000). The

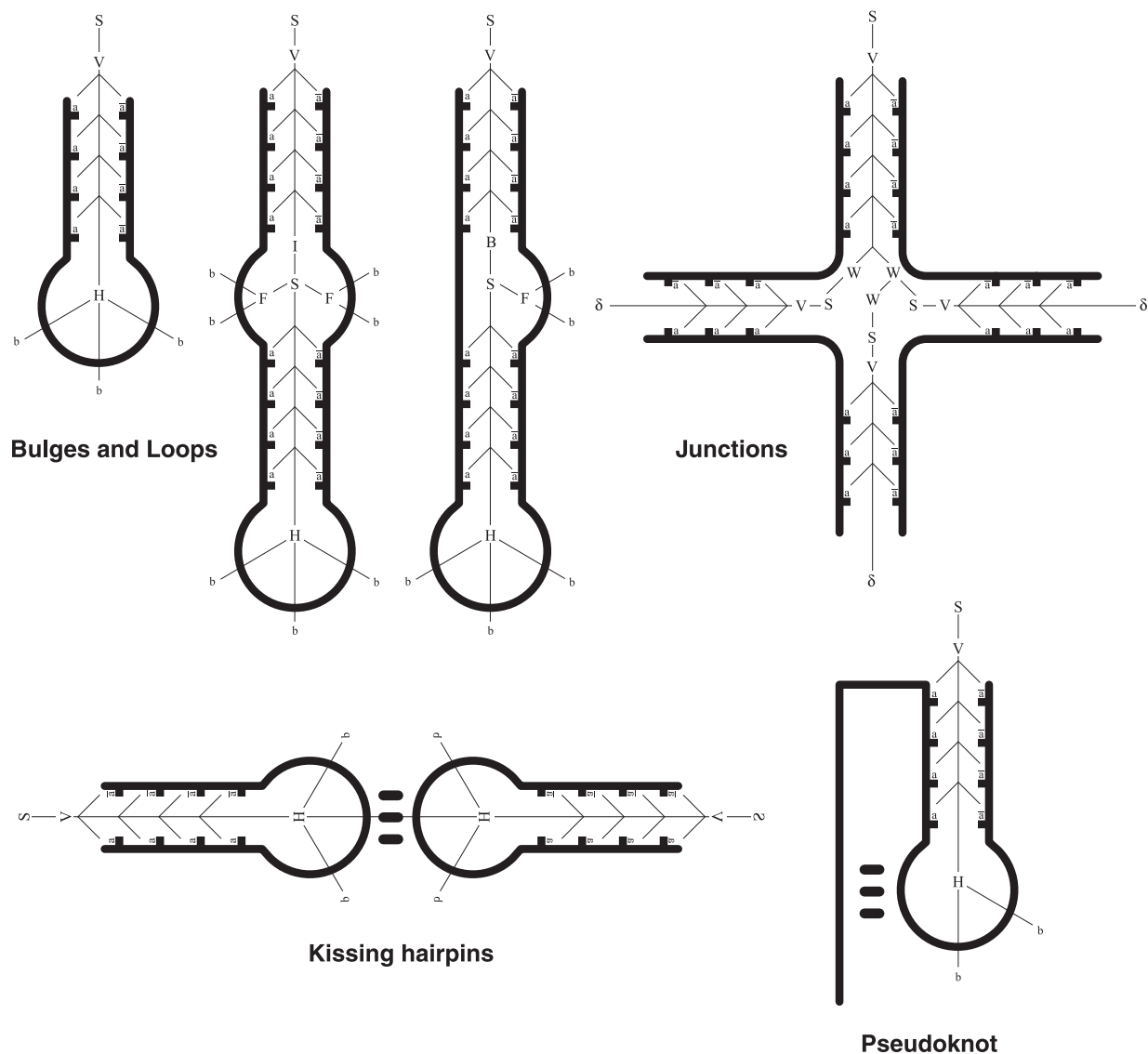


Fig. 1 RNA Structural Motifs. The intra-molecular binding creates RNA motifs such as bulges, internal, and hairpin loops. RNA molecule can forms inter-molecular binding, which produces junctions. From 2-way or kink-turn, the commonly used 3-way or 3WJ in nanoparticles fabrication, four-way (shown here), and available up to 9 way junctions), kissing hairpins and pseudoknot. There are also GNRA-Loop Receptor, Triple helical scaffold and G-Quadruplex motifs that are common in RNA suprastructure constructions. Reproduced from Jasinski, D., Haque, F., Binzel, D.W., Guo, P., 2017. Advancement of the emerging field of RNA nanotechnology. *ACS Nano* 11 (2), 1142–1164.

2'-hydroxyl group locks the ribose sugar into a 3'-endo chair conformation. It is structurally favorable for the RNA double helix to adopt the A-form which is 20% shorter and wider than B-form DNA helices. The A-form RNA helices are thermodynamically more stable than the B-form DNA helices (SantaLucia-Jr, 1998). However, the 2'-hydroxyl group permits RNA to be susceptible to nuclease digestion which limits RNA stability in-vivo.

For a given RNA sequence, there is often a diverse set of secondary structures it can fold into. Which structure is favored will depend on the physico-chemical environment of the molecule, for example, the presence or absence of substrates or ions. In addition, the presence of non-canonical base pairs can contribute significantly to RNA folding. Affecting the rigidity and thermodynamic stability of RNA secondary structures (Mathews and Turner, 2006). Conversely, a diverse set of sequence configurations can yield a particular secondary structure (Draper, 1996; Zuker, 1989) through energy minimization model. Subsequently, interactions among secondary structure motifs lead to the formation of a tertiary structure which in some cases entails functionality (Fig. 2).

Compared to the folding of protein, the amount of energy released during the formation of RNA secondary structure is much larger than the energy required during the formation of its tertiary structure. On its own, RNA secondary structure is quite informative. Long helices that are present in the secondary phase are most likely preserved in its tertiary structure. This, therefore provide a reasonable prediction of the basic form and relative positioning for some elements in the tertiary structure (Higgs, 1995). Determining the secondary structures is an integral element of predicting the tertiary motifs.

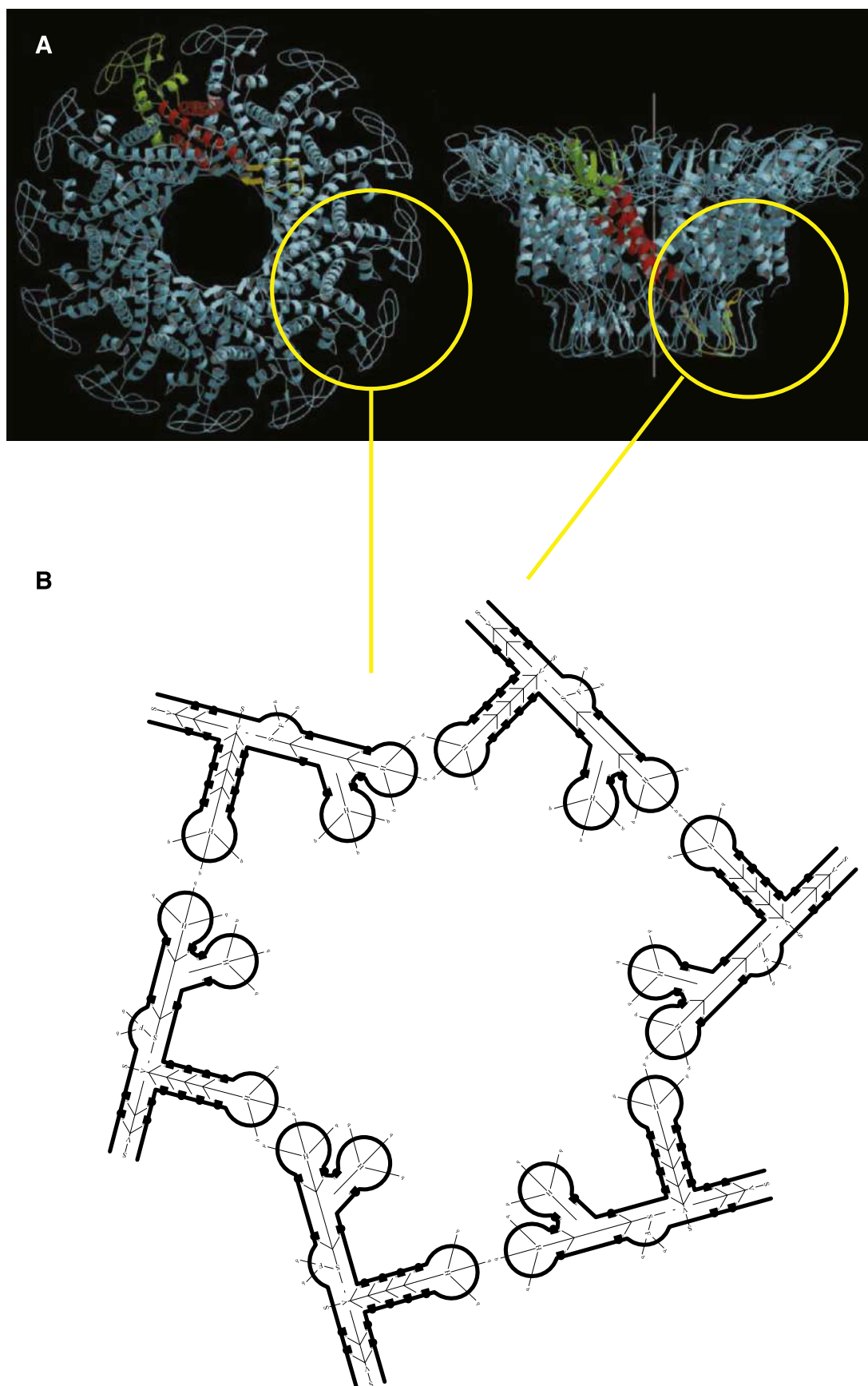
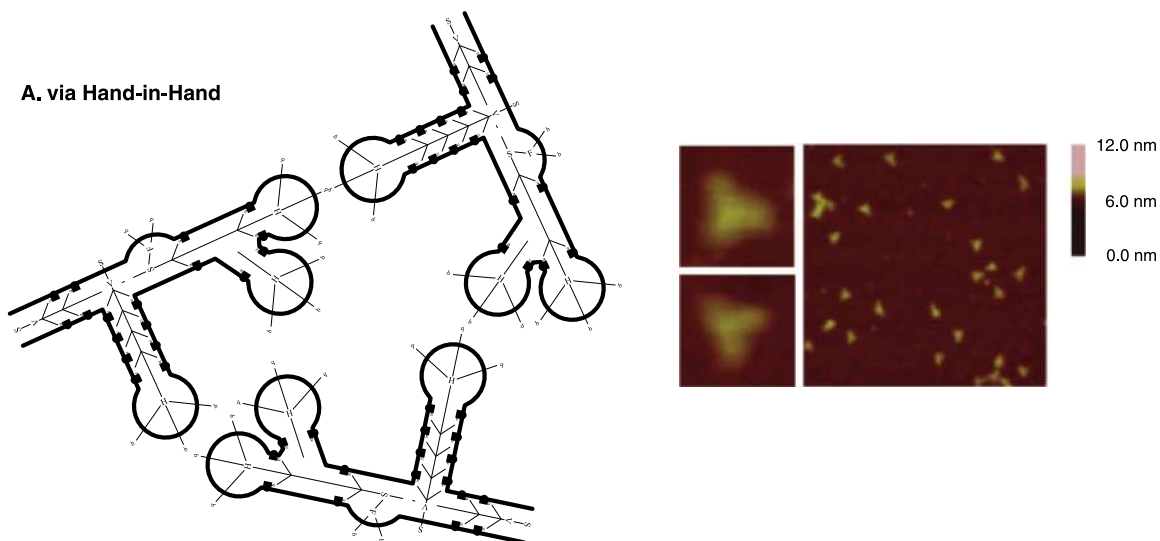
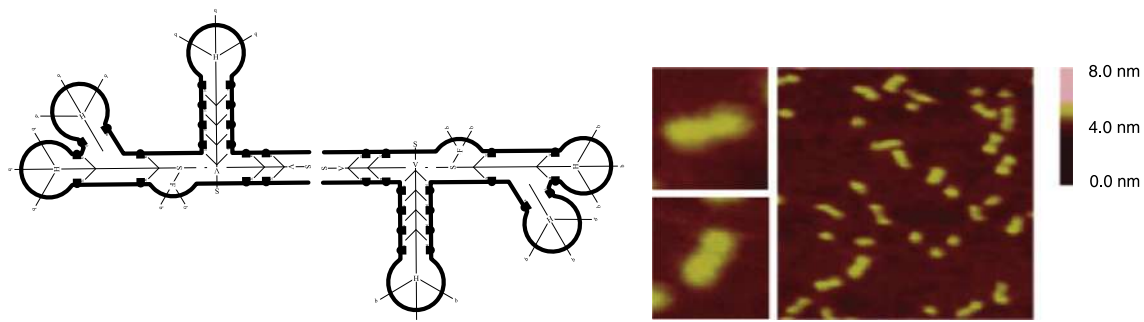
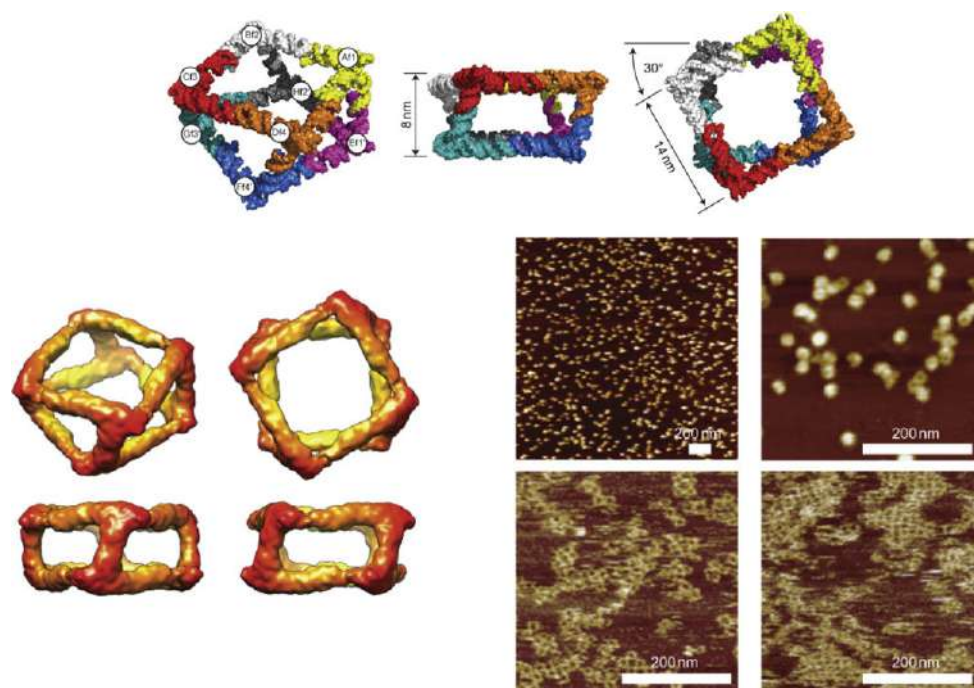


Fig. 2 The hierarchical folding of RNA molecule exemplified in the hexameric ring packaging RNA pRNA). A. The connector and structure ribbon diagrams of the Bacteriophage Φ 29 DNA packaging motor. Reproduced with permission from Simpson, A.A., Tao, Y., Leiman, P.G., *et al.*, 2000. Structure of the bacteriophage 29 DNA packaging motor. *Nature* 408, 745750. Copyright 2000 Nature Publishing Group. B. Conceptual representation of a pRNA hexamer assembled using the Hand-in-Hand strategy resembling the six packaging RNAs of Φ 29 DNA packaging motor.

A. via Hand-in-Hand**B. via Foot-to-Foot****C. via RNA Architectonics (tectoRNA)**

Natural occurring RNAs have been identified and characterized using NMR and crystallographic atomic structures (Leontis *et al.*, 1995; Moore, 1999). Natural occurring RNAs provide a definitive blueprint in the design of supramolecular structures through the representation of the sequence to structure relationship (i.e., natural occurring structures are encoded in naturally conserved sequences). These natural motifs provide geometry information of helical elements, and mediate stereochemically precise and readily reversible tertiary and quaternary interactions (Jaeger and Chworos, 2006). This information allows RNA supra-structures to be engineered with relatively accurate approximation supplemented by the secondary structure prediction. In terms of engineering supra-molecular RNA structures, this hierarchical folding state has to be observed closely to avoid kinetically trapped misfolded states, which disrupt the formation of the intended structures.

Techniques for Constructing RNA Nanostructures

A variety of supramolecular RNA structures (also known as RNA nanoparticles) have been successfully fabricated using various self-assembly approaches. In depth reviews of these methodologies are available in Li *et al.* (2015); Jasinski *et al.* (2017). This section provides a description of these construction methodologies, and we strongly recommend readers to refer to the references that are supplemented in each method for further understanding of the self-assembly mechanisms.

Via Hand-In-Hand Interactions

The strategy utilizes RNA loop elements as inter-RNA mounting dovetails (Fig. 3(A)). Incorporating these internal RNA loops eliminates the need for external dovetail elements. The design principles are exemplified by the structural features of the packaging RNA (pRNA) derived from bacteriophage *phi29* (Guo *et al.*, 1987). Through inter-locking loop interactions between different pRNAs, dimers, trimers, tetramers, pentamers, hexamers, and heptamers have been created (Guo *et al.*, 1998; Shu *et al.*, 2004, 2013a,b). Adaptation using non-covalent loop-loop contacts based on RNAI/III kissing-loop complex have been used to build RNA nanorings (Yingling *et al.*, 2007; Grabow *et al.*, 2011), which are thermostable, ribonuclease resistance and capable of delivering RNA interference modules.

Via Foot-to-Foot Interactions

Extending the 3' end of pRNA monomer with a single stranded palindrome sequence promotes self-assembly of two pRNA molecules. The complementarity of the palindrome sequence bridges the two foot orientations (i.e., foot-to-foot interaction) (as depicted Fig. 3(B)) (Shu *et al.*, 2004, 2013a,b). Construction of RNA hexamers, octamers, decamers, and dodecamers utilizing palindrome sequence to bridge RNA motifs, or scaffolds have been reported (Li *et al.*, 2015). Combination of both hand-in-hand and foot-to-foot interactions has been reported to construct pRNA array (Shu *et al.*, 2004). Fusions of sticky-end cohesion with foot-to-foot interactions were also reported to construct RNA nanoprisms in Hao *et al.* (2014) through the re-engineering of pRNA molecule.

Via Rational Design With Natural RNA Motifs

The discovery of natural occurring RNA molecules provides a definitive template in constructing RNA supramolecules, where precise structural information and structure to sequence relationship are defined through various RNA motifs. Rational design utilizing stable RNA motifs (e.g., kissing loops, dovetails, pseudoknots, kink turns, and multiway junctions) has produced exciting RNA supramolecule structures (Leontis *et al.*, 1995; Leontis and Westhof, 2003; Petrov *et al.*, 2013). Larger, more complex, and extraordinary stable natural motifs, such as *phi29* motor of pRNA (Guo *et al.*, 1998; Zhang *et al.*, 2013), tRNA (Severcan *et al.*, 2010), 5S RNA (Shu *et al.*, 2011), and others (Grabow *et al.*, 2011) have been used to form synthetic RNA supramolecules (Dibrov *et al.*, 2011; Shu *et al.*, 2013b; Jasinski *et al.*, 2014; Khisamutdinov *et al.*, 2014). A popular RNA motifs exploitation is the robust pRNA-3WJ motif. The extended work of hybrid strategy utilizing *phi29* pRNA-3WJ motifs for nanostructures fabrication with precise size, 3D architecture and container, RNA Dendrimer and co-transcription and inter-cellularly multifunctional nanostructures have been reported (Jasinski *et al.*, 2017).

Via RNA Architectonics (Tectona)

RNA tectonics focuses on the modularity of RNA molecules (Jaeger and Chworos, 2006). The strategy exploited the ability of RNA structures to decomposed and reassembled into a new modular RNA structure (Ref. Fig. 3(C)). Through the inversion of the

Fig. 3 The fabrication of RNA supramolecular structures using various assembly strategies. A. The conceptual representation of the trimers formed between three monomers and B. The formation of dimer using the foot-to-foot strategy. Reproduced from Shu, D., Moll, W.-D., Deng, Z., Mao, C., Guo, P., 2004. Bottom-up assembly of RNA arrays and superstructures as potential parts in nanotechnology. *Nano Letters* 4 (9), 1717–1723. Shu, Y., Haque, F., Shu, D., *et al.*, 2013a. Fabrication of 14 different RNA nanoparticles for specific tumor targeting without accumulation in normal organs. *RNA* 19, 767–777. The AFM images for both are reproduced with permission from Shu, Y., Shu, D., Haque, F., Guo, P., 2013b. Fabrication of pRNA nanoparticles to deliver therapeutic RNAs and bioactive compounds into tumor cells. *Nature Protocols* 8, 1635–1659. Copyright 2013 Nature Publishing Group. C. A polyhedron construction using tRNAs. 3D model, cryo-EM and AFM images are reproduced with permission from Severcan, I., Geary, C., Chworos, A., *et al.*, 2010. A polyhedron made of tRNAs. *Nature Chemistry* 2 (9), 772–779. Copyright 2010 Nature Publishing Group.

hierarchical folding (from tertiary to secondary structure to primary sequence), outline of the supramolecular RNA structures encompassing the tertiary motifs, secondary motifs, intra and inter-molecular interactions, and conserved sequences can be extracted. Thus, providing the geometry and self-assembly phases necessary to design the supramolecules. With the availability of Structural Classification of RNA (SCOR) (Klosterman *et al.*, 2004; Tamura *et al.*, 2004), nucleic acids databases (NDB) (Berman *et al.*, 1996), and the RNAjunction database (Bindewald *et al.*, 2008b), there exists a large number of candidate tectoRNA motifs. Readily configured as supramolecules through semi-rigid double helical bindings as exemplified in Westhof *et al.* (1996); Jaeger and Leontis (2000); Chworos *et al.* (2004); Shu *et al.* (2004); Nasalean *et al.* (2006); Dibrov *et al.* (2011); Ishikawa *et al.* (2013); Yu *et al.* (2015).

Via RNA Origami

The field of DNA nanotechnology leverages on the DNA complementarity self-assembly guided by the canonical Watson-Crick interactions (G-C and A-T base pairing), and structural rules (based on Holliday junction) to create DNA nanostructures (Lin *et al.*, 2006; Feldkamp and Niemeyer, 2006; Seeman, 2007, 2010; Lin *et al.*, 2009). These governing rules have resulted into various DNA nanoscaffolds (Brucalé *et al.*, 2006; Erben *et al.*, 2007; Andersen *et al.*, 2008; Aldaye *et al.*, 2008; Zimmermann *et al.*, 2008; Bhatia *et al.*, 2009; Licata and Tkachenko, 2009) with extensive functionalities formed through self-assembly of DNA subunits called “DNA tiles”. A different approach for DNA “origami” was presented in Rothmund (2006), where by a long stranded DNA is “stapled” into complex geometry using short complementary DNA strands. In this context, the less hierarchical and directed nature of DNA folding (i.e., double-stranded formation) are better suited for such approach, exemplified by the fabrication of various DNA supramolecules (Dietz *et al.*, 2009; Ke *et al.*, 2009; Douglas *et al.*, 2012; Zadegan *et al.*, 2012). The interpretation of RNA “origami” utilizing the short “stapled” approach is presented in Endo *et al.* (2014), where long ssRNAs are folded into rode-like structures using short RNA staples through in-vitro transcription of dsDNA scaffold and DNA staple templates. Further expansion of the “ssOrigami” method for single-stranded RNAs are discussed in Han *et al.* (2017).

Computational Methods of Constructing RNA Supramolecules

Despite the various strategies of constructing RNA supramolecules (as presented in Section “Techniques for Constructing RNA Nanostructures”), in actuality, the design and construction of these supramolecular structures relies heavily on manual assembly and fine tuning. Manual design has significant impact on the time and cost incurred to engineer these functional supramolecules, and taking into consideration the drive to produce more cost effective, and high demand for more complex structures, autonomous computational pipelines/protocols are a must. The general approach to RNA nanotechnology has now adapt to allow computational and folding prediction as one of the steps in the construction, immediately following the conceptualization as discussed in Guo (2010).

In contrast to protein, there exist well established computational tools that can aid in the secondary structure prediction and sequence design of RNA molecules. Tools such as *mfold* and *DINAMelt* (Zuker *et al.*, 1991; Zuker, 2003; Markham and Zuker, 2005), the *Vienna RNA Package* (Hofacker *et al.*, 1994), *RNAstructure* (Mathews *et al.*, 2004), *RNAsoft* (Andronescu *et al.*, 2003) and *RNAshapes* (Steffen *et al.*, 2006) are some of the most common and well-known in the prediction of RNA secondary structure. Secondary structure prediction is supplemented with sub-optimal prediction (Zuker, 1989; Wuchty *et al.*, 1999), which allows further determination of how well-defined the lowest free energy structure is, and at the same time highlight weak base-pairing positions that are present in a structure.

Secondary structure prediction has also been extended to includes interacting molecules (Andronescu, 2003; Dimitrov and Zuker, 2004; Muckstein *et al.*, 2006; Andronescu *et al.*, 2005; Bernhart *et al.*, 2006; Bindewald *et al.*, 2011). In the context of engineering RNA supramolecules, understanding the interactions between multiple RNA strands is essential in order to allow the evaluation of their binding compatibility to take place. The secondary structure prediction for interacting RNA molecules also quantifies the probability of homo-dimer formation (Dimitrov and Zuker, 2004). The relative probability of inter-molecular and intra-molecular binding is another critical aspect to consider. To arrive to the desired supramolecular structures, it is important that the base pairing within each molecule binds stronger than any possible intermolecular binding involving other molecules in the solution, or vice-versa depending on the intended function and structural constraints.

Although definitive sequence to structure mappings are available for natural occurring RNA motifs, the exploitation of multi-motifs self-assembly (following the previously discussed strategies) requires synthetic modification to the existing sequence. Therefore, consideration for the inverse folding prediction must be taken. The inverse folding problem involves designing RNA sequences that fold into a specified secondary structure (Schuster, 2006). By solving this problem, it allows for supramolecular structures to be artificially designed based on specific structural constraints and functionality. The three common inverse prediction tools are *RNAinverse* from the Vienna package (Hofacker *et al.*, 1994), *RNAdesigner* (Andronescu *et al.*, 2004) from the *RNAsoft* package (Andronescu *et al.*, 2003) and *INFO-RNA* (Busch and Backofen, 2006). In test cases of more than 150 nt artificially and naturally occurring RNA molecules, both these tools *RNAdesigner* and *INFO-RNA* produce encouraging results with both performs much better than *RNAinverse* overall (Andronescu *et al.*, 2004; Busch and Backofen, 2006; Zadeh *et al.*, 2011).

Since, the characterization and categorization of known RNA motifs plays a major role in constructing RNA supramolecular structures, the curation of RNA motifs into searchable databases is a powerful tool for the computer aided design pipeline (Jaeger and Chworos, 2006; Bindewald *et al.*, 2008a,b). To facilitate the construction, information related to the conserved sequences and

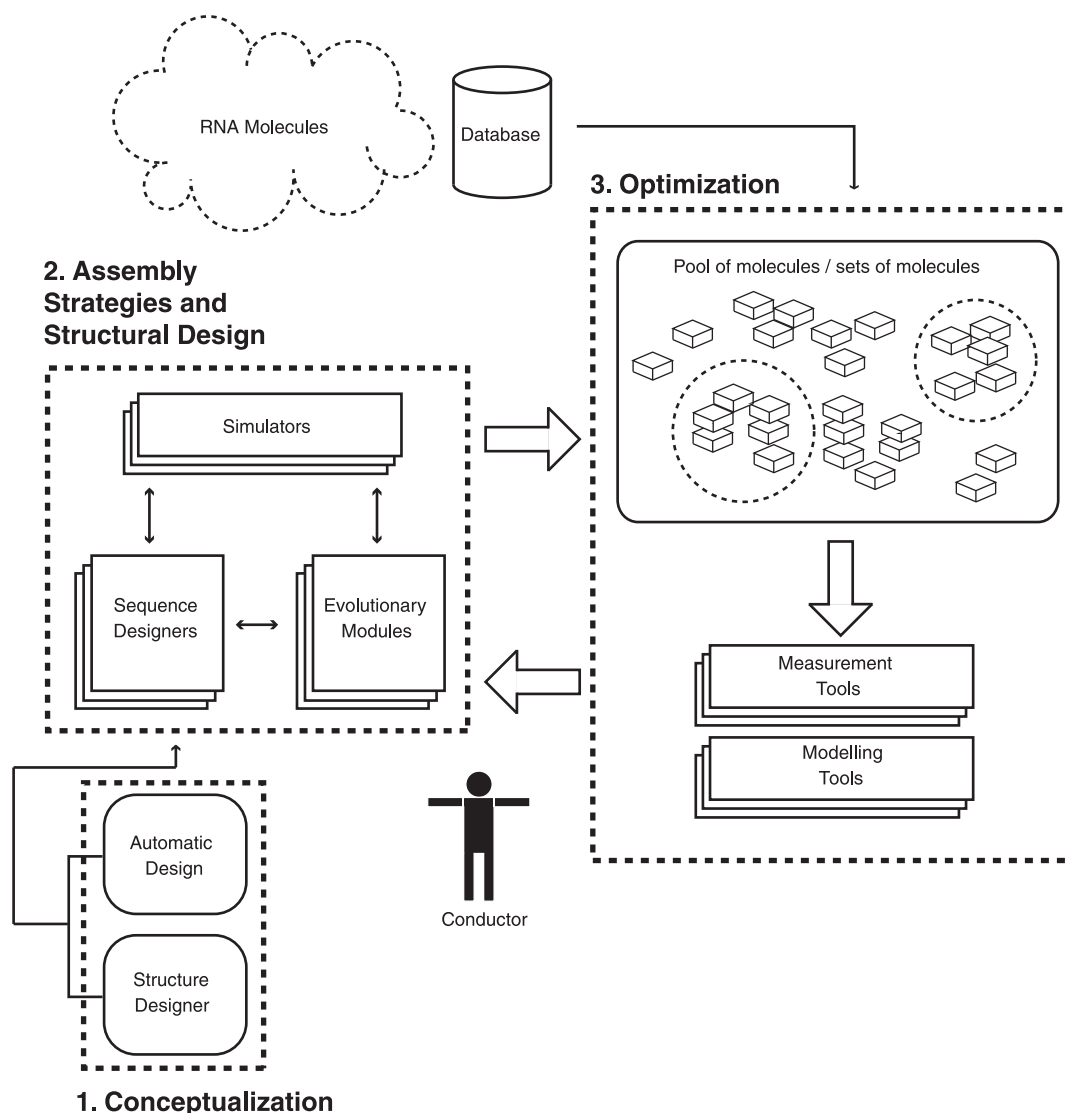


Fig. 4 The computational pipeline for engineering RNA supramolecular structures. The conceptualization produces hypothetical module of the structure that will undergo structural configuration and refinement before candidates are generated for lab verification.

structures mapping is key (i.e., to reduce the run time needed during inverse folding prediction and to ensure workability in the laboratory). This, determines the compatibility of multi-motifs self-assembly and other structural manipulations (Jaeger and Chworos, 2006). These databases classify RNA structures by function, tertiary interactions (geometry), as in the Structural Classification of RNA (SCOR) and nucleic acid database (NDB) (Berman *et al.*, 1996; Klosterman *et al.*, 2004; Tamura *et al.*, 2004). One of the most commonly associated databases for RNA supramolecular construction is RNAJunction (Bindewald *et al.*, 2008b). RNAJunction consists of approximately 13,000 junction motifs and kissing loops derived from the Protein Data Bank (PDB). These motifs are clustered into 2-way or 2WJ (bulge and internal loops) up to 9-way or 9WJ junctions. RNAJunction database also includes the basis of documented geometries and recurring base- and tertiary- interactions of these motifs.

In addition to the databases, there are various design tools that can aid the design of RNA supramolecular structures in three dimensional workspace (Bindewald *et al.*, 2008a; Martinez *et al.*, 2008; Jossinet *et al.*, 2010; Xia *et al.*, 2010). One such tool is NanoTiler (Bindewald *et al.*, 2008a). NanoTiler generates candidate RNA motifs from the RNAJunction database or similar databases. These candidate motifs can be assemble through a variable length pre-defined RNA helices. The tool also allows sequence conservations, to ensure conformity of the tertiary foldings. NanoTiler can function similar to a design studio, where custom candidate motifs can be generated with various structural modifications subjected to the plausible characteristic of natural RNA folding principles, and it is also capable of producing RNA primary sequences to match these custom designed motifs. These sequences have good agreement to be realized in the laboratory as exemplified in the construction of RNA supramolecules depicted in Bindewald *et al.* (2008a).

In addition to NanoTiler, we have RNA2D3D (Martinez *et al.*, 2008). RNA2D3D allows for rapid three dimensional modeling of the supramolecular structures using the primary sequence and secondary structure pairing information. It also provides adjustment tools for

structural modification; to avoid steric clashes and optimize the thermodynamics. RNA2D3D capacity to integrate multiple motifs into a draft model of supramolecule structures (through default kissing-loop interactions) is essential in providing immediate feedback to the user. Alternatively, there are various general purpose molecular modeling packages, such as the Chimera (Pettersen *et al.*, 2004), and pyMol (Grell *et al.*, 2006), which are capable in modeling RNA supramolecular structures. However, due to its generic form, these tools lack the specific knowledge-based of RNA foldings; but offer flexibility for integration with other customizable pipelines.

Using the strategies (discussed in section “Techniques for Constructing RNA Nanostructures”), a generic computational pipeline can be established as depicted in Fig. 4. The implementation of the pipeline is subjected to the level of depth in the conceptualization phase as suggested in Afonin *et al.* (2014). This level can either be; whether (i) candidate motifs are already identified (or partially identified) or, (ii) a discovery to identify suitable motifs is needed. Key in the pipeline is the conceptualization phase. During this phase, the identification of the desired structures and functions are established. This includes consideration on the physico-chemical environment, size, complexity, geometry, restriction on sequences and relevant factors that should be based on extensive RNA folding knowledge base. The conceptualization will either produced constraints for immediate structural assembly and modification – For (i) (also referred as “Draft Secondary Structure” in Afonin *et al.* (2014)), or a set of criterion that can be used as search parameters in motif discovery - for (ii) (also referred as “Connectivity Rules” in Afonin *et al.* (2014)). Either way, both of these findings are essential and will be adapted as specification sheet during the construction phase.

Next, the selection of strategy for multi-motifs self-assembly. Although categorically different, these strategies are in-fact overlap (Afonin *et al.*, 2014). A closed design principle (i.e., where the self-assembly motifs are manually pre-determined) promotes a top-to-bottom optimization, where hypothetical supramolecular structures are decomposed into motifs with inter-molecular reactions reassembly components based on the “best-fit” strategy conforming to the selected motifs. In contrast, an open system, where only connectivity rules are defined (case (ii)), the bottom-up optimization is preferred. Combinatory search of the databases to find candidate motifs is executed and inter-molecular assembly recommendations would be made based on the “best-fit” strategy depending on the motif compatibility scores. During these process of optimizing the formation of structures, RNA 3D modeling programs such as NanoTiler and RNA2D3D are important. These tools provide immediate feedback on the compatibility of self-assembly between the suggested motifs.

Structural optimization follows next. The summary of the protocols are presented in Shapiro *et al.* (2008). The NanoTiler optimization protocol is described in Algo. 1. Interactions with NanoTiler are permissible via a graphical user interface or scripting language. NanoTiler allows optimization to be performed on motif placements and helix distortion to improve on the “best-fit” score. The bottom-up protocol is described in Algo. 2, and requires both RNA2D3D and NanoTiler. RNA2D3D allows interaction rules between multiple building blocks to be included (i.e., “Connectivity Rules”) and rapidly generates the approximate supramolecular structure rendering. NanoTiler, in this instance acts as a refinement module to improve on the configuration of the selected motifs.

Algorithm 1 Protocol for RNA 3D Motifs using NanoTiler

- 1: 3D graph representation of structures $\leftarrow$ 3D coordinates
 - 2: Motifs selection $\leftarrow$ RNAJunction
 - 3: Measure fitting error (d)
 - 4: **while** d **do**
 - 5: Identify $J \leftarrow$ junction
 - 6: Generate dsHelices (dsRNA) interpolating between fragments
 - 7: fn-mutate (J , dsRNA) $\leftarrow$ random chosen vertices
 - 8: Measure d_1
 - 9: **if** $d_1 \leq d$ **then**
 - 10: replace J and dsRNA
 - 11: **end if**
 - 12: **end while**
 - 13: Optimize sequences for the model
 - 14: Apply MM and MD for structural refinement
-

Algorithm 2 Protocol for RNA 2D Motifs using NanoTiler and RNA2D3D

- 1: 3D graph representation of structures $\leftarrow$ 3D coordinates
- 2: Building Blocks $\leftarrow$ Determine Secondary structure templates
- 3: Optimize sequences
- 4: Extract 3D coordinates $\leftarrow$ Run RNA2D3D
- 5: Optimize sequences for the model
- 6: Apply MM and MD for structural refinement

As recommended by [Shapiro *et al.* \(2008\)](#), the designs are subjected to molecular mechanics and molecular dynamics inspection to ensure positive results in the laboratory. Output of the pipeline will then undergo the fabrication process. Two strategies for fabricating RNA supramolecular structure is discussed in [Jaeger and Chworos \(2006\)](#). The first approach is a single step assembly, in which all the sequences (i.e., primary sequence of the RNA motifs) are mixed together and assembled in a solution through a slow annealing process. The second strategy involves stepwise hierarchical self-assembly. Smaller sub-units are separately assembled before all these motifs are mixed together to form the supramolecular structures. Conceptually similar to the hierarchical folding property of RNA molecule.

Conclusion

RNA supramolecule constructions involves intra/inter-molecular assembly of various RNA motifs. Understanding the folding of RNA motifs is essential in ensuring the conformity of the structure and functionality. The flexibility of RNA folding, allows variations (or meta-stable states) to exist within a small free energy value difference. This variability is further compounded when multiple motifs assembly are involved. Despite the presence of prediction tools to aid in the construction of RNA supramolecular structures, accurate prediction of tertiary and quarternary motifs remains a challenge. Prediction tools specific for tertiary structures are yet to be explored. Within this context, inverse folding involving multiple strands is necessary.

On the computational aspects, more enhancements are needed. For instance, the construction of supramolecule must consider a combination of an increased number of RNA motifs to improve on the demand for more complex multi-functional RNA structures. Accuracy in size, structural and function of these supramolecules are aspect that should be addressed during the design phase. Perhaps, a better combinatorial search with sufficient parameterization can help in identification of suitable RNA motifs, and a deep learning methodology can be incorporated in understanding the geometry and self-assembling mechanisms of complex natural occurring RNA supramolecule, thus, providing valuable insights into a better strategy of constructing RNA supramolecule. This will greatly benefit the field of RNA nanotechnology especially towards the design of a more complex and advanced supramolecule applicable towards the realization of nanomedicine.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Natural Language Processing Approaches in Bioinformatics

References

- Afonin, K.A., Kasprzak, W.K., Bindewald, E., *et al.*, 2014. In silico design and enzymatic synthesis of functional RNA nanoparticles. *Accounts of Chemical Research* 47 (6), 1731–1741.
- Aldaye, F.A., Palmer, A.L., Sleiman, H.F., 2008. Assembling materials with dna as the guide. *Science* 321 (5897), 1795–1799.
- Altman, S., 1990. Enzymatic cleavage of RNA by RNA (Nobel lecture). *Ange-wandte Chemie International Edition* 29, 749–758.
- Andersen, F.F., Knudsen, B., Oliveira, C.L.P., *et al.*, 2008. Assembly and structural analysis of a covalently closed nano-scale dna cage. *Nucleic Acids Research* 36 (4), 1113–1119.
- Andronescu, M., 2003. Algorithms for predicting the secondary structure of pairs and combinatorial sets of nucleic acid strands. Master's thesis, University of British Columbia, Vancouver.
- Andronescu, M., Anguirre-Hernandez, R., Condon, A., Hoos, H.H., 2003. RNAsoft: A suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Research* 31 (13), 3416–3422.
- Andronescu, M., Fejes, A.P., Hutter, F., Condon, A., Hoos, H.H., 2004. A new algorithm for RNA secondary structure design. *Journal of Molecular Biology* 336 (3), 607–624.
- Andronescu, M., Zhang, Z.C., Condon, A., 2005. Secondary structure prediction of interacting RNA molecules. *Journal of Molecular Biology* 345, 987–1001.
- Shapiro, B.A., Bindewald, E., Kasprzak, W., Yingling, Y., 2008. Protocols for the in silico design of RNA nanostructures. In: Gazit E., N.R. (Ed.), *Nanostructure Design. Methods in Molecular Biology*, vol. 474. Humana Press, pp. 93–115.

- Berens, C., Groher, F., Suess, B., 2015. RNA aptamers as genetic control devices: The potential of riboswitches as synthetic elements for regulating gene expression. *Biotechnology Journal* 10 (2), 246–257.
- Berg, J.M., Tymoczko, J.L., Stryer, L., 2003. *Biochemistry*, fifth ed. New York: W. H. Freeman and Company.
- Berman, H.M., Gelbin, A., Westbrook, J., 1996. Nucleic acid crystallography: A view from the nucleic acid database. *Progress in Biophysics and Molecular Biology* 66 (3), 255–288.
- Bernhart, S.H., Tafer, H., Muckstein, U., *et al.*, 2006. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms for Molecular Biology* 1 (3), doi:10.1186/1748-7188 1–3.
- Bhatia, D., Mehtab, S., Krishnan, R., *et al.*, 2009. Icosahedral dna nanocapsules by modular assembly. *Angewandte Chemie International Edition* 48 (23), 4134–4137.
- Bindewald, E., Afonin, K., Jaeger, L., Shapiro, B.A., 2011. Multistrand rna secondary structure prediction and nanostructure design including pseudoknots. *ACS Nano* 5 (12), 9542–9551.
- Bindewald, E., Grunewald, C., Boyle, B., O'Connor, M., Shapiro, B.A., 2008a. Computational strategies for the automated design of RNA nanoscale structures from building blocks using nanotiler. *Journal of Molecular Graphics and Modelling* 27 (3), 299–308.
- Bindewald, E., Hayes, R., Yingling, Y.G., Kasprzak, W., Shapiro, B.A., 2008b. RNAJunction: A database of RNA junctions and kissing loops for three-dimensional structural analysis and nanodesign. *Nucleic Acids Research* 36, D392–D397.
- Bloomfield, V.A., Crothers, D.M., Tinoco Jr., I., 2000. *Nucleic Acids: Structures, Properties and Functions*, first ed. California: University Science Books.
- Brucale, M., Zuccheri, G., Rossi, L., *et al.*, 2006. Characterization and modulation of the hierarchical self-assembly of nanostructured dna tiles into supramolecular polymers. *Organic and Biomolecular Chemistry* 4, 3427–3434.
- Busch, A., Backofen, R., 2006. INFO-RNA – A fast approach to inverse RNA folding. *Bioinformatics* 22 (15), 1823–1831.
- Chworos, A., Severcan, I., Koyfman, A.Y., *et al.*, 2004. Building programmable jigsaw puzzles with RNA. *Science* 306 (5704), 2068–2072.
- Couzín, J., 2002. Small RNAs make big splash. *Science* 298, 2296–2297.
- Dibrov, S.M., McLean, J., Parsons, J., Hermann, T., 2011. Self-assembling RNA square. *Proceedings of the National Academy of Sciences* 108 (16), 6405–6408.
- Dietz, H., Douglas, S.M., Shih, W.M., 2009. Folding dna into twisted and curved nanoscale shapes. *Science* 325 (5941), 725–730.
- Dimitrov, R.A., Zuker, M., 2004. Prediction of hybridization and melting for double-stranded nucleic acids. *Biophysical Journal* 87, 215–226.
- Douglas, S.M., Bachelet, I., Church, G.M., 2012. A logic-gated nanorobot for targeted transport of molecular payloads. *Science* 335 (6070), 831–834.
- Draper, D.E., 1996. Strategies for RNA folding. *Trends in biochemistry. Sciences* 21, 145–149.
- Endo, M., Takeuchi, Y., Emura, T., Hidaka, K., Sugiyama, H., 2014. Preparation of chemically modified RNA origami nanostructures. *Chemistry A European Journal* 20 (47), 15330–15333.
- Erben, C.M., Goodman, R.P., Turberfield, A.J., 2007. A self-assembled dna bipyramid. *Journal of the American Chemical Society* 129 (22), 6992–6993.
- Esposito, C.L., Passaro, D., Longobardo, I., *et al.*, 2011. A neutralizing RNA aptamer against EGFR causes selective apoptotic cell death. *PLOS ONE*. e24071.
- Feldkamp, U., Niemeyer, C.M., 2006. Rational design of dna nanoarchitectures. *Angewandte Chemie International Edition* 45 (12), 1856–1876.
- Goodsell, D.S., 2004. *Bionanotechnology: Lessons from Nature*, first ed. Hoboken, New Jersey: Wiley-Liss.
- Grabow, W.W., Zakrevsky, P., Afonin, K.A., *et al.*, 2011. Self-assembling RNA nanorings based on RNA/II inverse kissing complexes. *Nano Letters* 11 (2), 878–887.
- Grell, L., Parkin, C., Slate, L., Craig, P.A., 2006. Ez-viz, a tool for simplifying molecular viewing in pymol. *Biochemistry and Molecular Biology Education* 34 (6), 402–407.
- Guo, P., 2010. The emerging field of RNA nanotechnology. *Nature Nanotechnology* 12 (5), 833–842.
- Guo, P., Erickson, S., Anderson, D., 1987. A small viral RNA is required for in vitro packaging of bacteriophage phi29 DNA. *Science* 236 (4802), 690–694.
- Guo, P., Haque, F., Hallahan, B., Reif, R., Li, H., 2012. Uniqueness, advantages, challenges, solutions, and perspectives in therapeutics applying RNA nanotechnology. *Nucleic Acid Therapeutics* 22 (4), 226–245.
- Guo, P., Zhang, C., Chen, C., Garver, K., Trottier, M., 1998. Inter-RNA interaction of phage 29 pRNA to form a hexameric complex for viral DNA transportation. *Molecular Cell* 2 (1), 149–155.
- Han, D., Qi, X., Myhrvold, C., *et al.*, 2017. Single-stranded dna and rna origami. *Science* 358 (6369), Available at: <http://science.sciencemag.org/content/358/6369/eaao2648>.
- Hannon, G.J., 2002. RNA interference. *Nature* 418, 244–251.
- Hao, C., Li, X., Tian, C., *et al.*, 2014. Construction of RNA nanocages by re-engineering the packaging RNA of Phi29 bacteriophage. *Nature Communications* 5, 3890.
- Haque, F., Shu, D., Shu, Y., *et al.*, 2012. Ultrastable synergistic tetraivalent {RNA} nanopar-ticles for targeting to cancers. *Nano Today* 7 (4), 245–257.
- Higgs, P.G., 1995. Thermodynamic properties of transfer RNA: A computational study. *Journal of the Chemical Society, Faraday Transactions* 9 (16), 2531–2540.
- Hofacker, I.L., Fontana, W., Stadler, P.F., *et al.*, 1994. Fast folding and comparison of RNA secondary structures. *Chemical Monthly* 125 (2), 167–188.
- Ishikawa, J., Furuta, H., Ikawa, Y., 2013. RNA tectonics (tectoRNA) for RNA nanostructure design and its application in synthetic biology. *Wiley Interdisciplinary Reviews: RNA* 4 (6), 651–664.
- Jaeger, L., Chworos, A., 2006. The architectonics of programmable RNA and DNA nanostructures. *Current Opinion in Structural Biology* 16 (4), 531–543.
- Jaeger, L., Leontis, N.B., 2000. Tecto-RNA: One-dimensional self-assembly through tertiary interactions. *Angewandte Chemie International Edition* 39 (14), 2521–2524.
- Jasinski, D., Haque, F., Binzel, D.W., Guo, P., 2017. Advancement of the emerging field of RNA nanotechnology. *ACS Nano* 11 (2), 1142–1164.
- Jasinski, D.L., Khisamutdinov, E.F., Lyubchenko, Y.L., Guo, P., 2014. Physicochemically tunable polyfunctionalized RNA square architecture with fluorogenic and ribozymatic properties. *ACS Nano* 8 (8), 7620–7629.
- Jossinet, F., Ludwig, T.E., Westhof, E., 2010. Assemble: An interactive graphical tool to analyze and build RNA architectures at the 2d and 3d levels. *Bioinformatics* 26 (16), 2057–2059.
- Kang, K.-N., Lee, Y.-S., 2012. RNA aptamers: A review of recent trends and applications. In: Z.J., J. (Ed.), *Future Trends in Biotechnology. Advances in Biochemical Engineering/Biotechnology*. Berlin, Heidelberg: Springer, pp. 153–169.
- Ke, Y., Sharma, J., Liu, M., *et al.*, 2009. Scaffolded dna origami of a dna tetrahedron molecular container. *Nano Letters* 9 (6), 2445–2447.
- Khisamutdinov, E.F., Jasinski, D.L., Guo, P., 2014. RNA as a boiling-resistant anionic polymer material to build robust structures with defined shape and stoichiometry. *ACS Nano* 8 (5), 4771–4781.
- Klosterman, P.S., Hendrix, D.K., Tamura, M., Holbrook, S.R., Brenner, S.E., 2004. Three dimensional motifs from the scor, structural classification of RNA database: Extruded strands, base triples, tetraloops and uturns. *Nucleic Acids Research* 32 (8), 2342–2352.
- Lee, T.J., Haque, F., Shu, D., *et al.*, 2015. RNA nanoparticle as a vector for targeted siRNA delivery into glioblastoma mouse model. *Oncotarget* 6, 14766–14776.
- Leontis, N.B., Lescoute, A., Westhof, E., 1995. The building blocks and motifs of RNA architecture. *Current Opinion Structural Biology* 16, 279287.
- Leontis, N.B., Westhof, E., 2003. Analysis of RNA motifs. *Current Opinion in Structural Biology* 13 (3), 300–308.
- Li, H., Lee, T., Dziubla, T., *et al.*, 2015. RNA as a stable polymer to build controllable and defined nanostructures for material and biomedical application. *Nano Today* 10 (5), 631–655.
- Licata, N.A., Tkachenko, A.V., 2009. Self-assembling dna-caged particles: Nanoblocks for hierarchical self-assembly. *Physical Review E* 79, 011404.
- Lin, C., Liu, Y., Rinker, S., Yan, H., 2006. Dna tile based self-assembly: Building complex nanoarchitectures. *ChemPhysChem* 7 (8), 1641–1647.
- Lin, C., Liu, Y., Yan, H., 2009. Designer dna nanoarchitectures. *Biochemistry* 48 (8), 1663–1674.
- Mandal, M., Breaker, R.R., 2004. Gene regulation by riboswitches. *Molecular cell biology* 5, 451–463.
- Markham, N.R., Zuker, M., 2005. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acid Research* 33, W577–W581. doi:10.1093/nar/gki591.
- Martinez, H.M., Jr, J.V.M., Shapiro, B.A., 2008. RNA2D3D: A program for generating, viewing, and comparing 3-dimensional models of RNA. *Journal of Biomolecular Structure and Dynamics* 25 (6), 669–683.

- Mathews, D.H., Disney, M.D., Childs, J.L., *et al.*, 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. In: *Proceedings of the National Academy of Sciences* vol. 101, pp. 7287–7292. USA.
- Mathews, D.H., Turner, D.H., 2006. Prediction of RNA secondary structure by free energy minimization. *Current Opinion in Structural Biology* 16 (3), 270–278.
- Moore, P.B., 1999. Structural motifs in RNA. *Annual Review of Biochemistry* 68, 287–300.
- Muckstein, U., Tafer, H., Hackermuller, J., *et al.*, 2006. Thermodynamics of RNA-RNA binding. *Bioinformatics* 22 (10), 1177–1182.
- Mulhbachter, J., St-Pierre, P., Lafontaine, D.A., 2010. Therapeutic applications of ribozymes and riboswitches. *Current Opinion in Pharmacology* 10 (5), 551–556.
- Nasalean, L., Baudrey, S., Leontis, N.B., Jaeger, L., 2006. Controlling RNA self-assembly to form filaments. *Nucleic Acids Research* 34 (5), 1381–1392.
- Nudler, E., Mironov, A.S., 2004. The riboswitch control of bacterial metabolism. *Trends in biochemical sciences* 29 (1), 11–17.
- Petersen, C.P., Doench, J.G., Grishok, A., Sharp, P.A., 2006. The biology of short RNAs. In: Gesteland, R.F., Cech, T.R., Atkins, J.F. (Eds.), *RNA World*, third ed. New York: Cold Spring Harbor Laboratory Press, pp. 535–565.
- Petrov, A.I., Zirbel, C.L., Leontis, N.B., 2013. Automated classification of RNA 3D motifs and the RNA 3D motif atlas. *RNA* 19 (10), 1327–1340.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. Ucsf chimera visualization system for exploratory research and analysis. *Journal of Computational Chemistry* 25 (13), 1605–1612.
- Rockey, W.M., Hernandez, F.J., Huang, S.-Y., *et al.*, 2011. Rational truncation of an RNA aptamer to prostate-specific membrane antigen using computational structural modeling. *Nucleic Acid Therapeutics* 21 (5), 299–314.
- Roh, Y.H., Deng, J.Z., Dreaden, E.C., *et al.*, 2016. A Multi-RNAi microsphere platform for simultaneous controlled delivery of multiple small interfering RNAs. *Ange-wandte Chemie* 128 (10), 3408–3412.
- Rothmund, P.W.K., 2006. Folding dna to create nanoscale shapes and patterns. *Nature* 440, 297–302.
- SantaLucia-Jr., J., 1998. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. In: *Proceedings of the National Academy of Sciences* vol. 95, pp. 1460–1465. USA.
- Schuster, P., 2006. Prediction of RNA secondary structures: From theory to models and real molecules. *Reports on Progress in Physics* 69, 1419–1477.
- Seeman, N.C., 2007. An overview of structural dna nanotechnology. *Molecular Biotechnology* 37 (3), 246.
- Seeman, N.C., 2010. Nanomaterials based on DNA. *Annual Review of Biochemistry* 79 (1), 65–87.
- Serganov, A., 2009. The long and the short of riboswitches. *Current Opinion in Structural Biology* 19 (3), 251–259.
- Severcan, I., Geary, C., Chworos, A., *et al.*, 2010. A polyhedron made of tRNAs. *Nature Chemistry* 2 (9), 772–779.
- Shigdar, S., Lin, J., Yu, Y., *et al.*, 2011. RNA aptamer against a cancer stem cell marker epithelial cell adhesion molecule. *Cancer Science* 102 (5), 991–998.
- Shu, D., Li, H., Shu, Y., *et al.*, 2015. Systemic delivery of Anti-miRNA for suppression of triple negative breast cancer utilizing RNA nanotechnology. *ACS Nano* 9 (10), 9731–9740.
- Shu, D., Moll, W.-D., Deng, Z., Mao, C., Guo, P., 2004. Bottom-up assembly of RNA arrays and superstructures as potential parts in nanotechnology. *Nano Letters* 4 (9), 1717–1723.
- Shu, D., Shu, Y., Haque, F., Abdelmawla, S., Guo, P., 2011. Thermodynamically stable RNA three-way junction for constructing multifunctional nanoparticles for delivery of therapeutics. *Nature Nanotechnology* 6, 658667.
- Shu, Y., Haque, F., Shu, D., *et al.*, 2013a. Fabrication of 14 different RNA nanoparticles for specific tumor targeting without accumulation in normal organs. *RNA* 19, 767–777.
- Shu, Y., Pi, F., Sharma, A., *et al.*, 2014. Stable RNA nanoparticles as potential new generation drugs for cancer therapy. *Advanced Drug Delivery Reviews* 66 (Suppl. C), 74–89.
- Shu, Y., Shu, D., Haque, F., Guo, P., 2013b. Fabrication of pRNA nanoparticles to deliver therapeutic RNAs and bioactive compounds into tumor cells. *Nature Protocols* 8, 1635–1659.
- Steffen, P., Voss, B., Rehmsmeier, M., Reeder, J., Giegerich, R., 2006. RNASHapes: An integrated RNA analysis package based on abstract shapes. *Bioinformatics* 22 (4), 500–503.
- Suess, B., Weigand, J.E., 2008. Engineered riboswitches: Overview, problems and trends. *RNA Biology* 5 (1), 24–29.
- Sundaram, P., Kurniawan, H., Byrne, M.E., Wower, J., 2013. Therapeutic RNA aptamers in clinical trials. *European Journal of Pharmaceutical Sciences* 48 (1), 259–271.
- Tamura, M., Hendrix, D.K., Klosterman, P.S., *et al.*, 2004. Scór: Structural classification of rna, version 2.0. *Nucleic Acids Research* 32, D182–D184.
- Thiel, K.W., Hernandez, L.I., Dassie, J.P., *et al.*, 2012. Delivery of chemo-sensitizing siRNAs to HER2 + -breast cancer cells using RNA aptamers. *Nucleic Acids Research* 40 (13), 6319–6337.
- Westhof, E., Masquida, B., Jaeger, L., 1996. RNA tectonics: Towards RNA design. *Folding and Design* 1 (4), R78–R88.
- Wuchty, S., Fontana, W., Hofacker, I.L., Schuster, P., 1999. Complete suboptimal folding of RNA and the stability of secondary structure. *Biopolymers* 49, 145–165.
- Xia, Z., Gardner, D.P., Gutell, R.R., Ren, P., 2010. Coarse-grained model for simulation of RNA three-dimensional structures. *The Journal of Physical Chemistry B* 114 (42), 13497–13506.
- Ye, X., Hemida, M., Zhang, H.M., *et al.*, 2012. Current advances in Phi29 pRNA biology and its application in drug delivery. *Wiley Interdisciplinary Reviews: RNA* 3 (4), 469–481.
- Yingling, G., Shapiro, Y., B., A., 2007. Computational design of an RNA hexagonal nanoring and an RNA nanotube. *Nano Letters* 7 (8), 2328–2334.
- Yu, J., Liu, Z., Jiang, W., Wang, G., Mao, C., 2015. De novo design of an RNA tile that self-assembles into a homo-octameric nanoprism. *Nature Communications* 6, 5724.
- Zadegan, R.M., Jepsen, M.D.E., Thomsen, K.E., *et al.*, 2012. Construction of a 4 zeptoliters switchable 3D DNA box origami. *ACS Nano* 6 (11), 10050–10053.
- Zadeh, J.N., Steenberg, C.D., Bois, J.S., *et al.*, 2011. Nupack: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry* 32 (1), 170–173.
- Zamore, P.D., Haley, B., 2005. Ribo-gnome: The big world of small RNAs. *Science* 309, 1519–1524.
- Zhang, H., Endrizzi, J.A., Shu, Y., *et al.*, 2013. Crystal structure of 3wj core revealing divalent ion-promoted thermostability and assembly of the phi29 hexameric motor pRNA. *RNA* 19, 1226–1237.
- Zhou, J., Bobbin, M., Burnett, J., Rossi, J., 2012. Current progress of RNA aptamer-based therapeutics. *Frontiers in Genetics* 3, 234.
- Zimmermann, J., Cebulla, M., Mnninghoff, S., vonKiedrowski, G., 2008. Self-assembly of a dna dodecahedron from 20 trisiliconucleotides with c3h linkers. *Angewandte Chemie International Edition* 47 (19), 3626–3630.
- Zuker, M., 1989. On finding all suboptimal foldings of an RNA molecule. *Science* 244, 48–52.
- Zuker, M., 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research* 31 (13), 3406–3415.
- Zuker, M., Jaeger, J.A., Turner, D.H., 1991. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Research* 19 (10), 2707–2714.

Predicting RNA-RNA Interactions in Three-Dimensional Structures

Hazrina Y Hamdani, Universiti Sains Malaysia, Kepala Batas, Malaysia

Zatil H Yahaya and Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Bangi, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Tertiary interactions, such as those mediated by hydrogen bonds, serve as stabilization elements of RNA 3D structure (Krasilnikov and Mondragón, 2003; Nissen *et al.*, 2001). To date, numerous examples of tertiary interactions have been reported and these can easily be browsed via resources such as the RNA 3D Hub (Leontis and Zirbel, 2012), InterRNA (Appasamy *et al.*, 2015), NCIR (Nagaswamy *et al.*, 2002), SCOR (Tamura *et al.*, 2004), RNAJunction (Bindewald *et al.*, 2007), RNA 3D Motif Atlas (Parlea *et al.*, 2016) and RNA Bricks (Chojnowski *et al.*, 2014). Tertiary interactions that are highly conserved can be considered as 3D or tertiary motifs.

The capability to predict RNA-RNA interactions is of even more importance given recent depositions of large assemblies of the ribosomal subunits with reasonably high resolution (Prokhorova *et al.*, 2017; Almutairi *et al.*, 2017; Tereshchenkov *et al.*, 2018). With high-resolution structures, the intricate atomic level interactions within the RNA's 3D structure can be adequate for structural analysis that can provide details of functional mechanisms (Koizumi *et al.*, 2016). Tertiary interactions within the RNA structure are biologically significant from various aspects. In a recent study on viroid RNA, the 3D base motifs are important factors for function, viroid infection and viroid genome evolution (Wang *et al.*, 2018). In addition, 3D structure of RNA has also shown to be conserved despite extensive sequence divergence (Capriotti and Marti-Renom, 2010). RNA tertiary motifs also provide specific transcriptional regulation together and may have therapeutic purposes in diseases (Huang *et al.*, 2015). The 3D structure of messenger RNA (mRNA) is also crucial in regulating the production of encoded proteins in addition to controlling and modulating synthesis and function of proteins (Mauger *et al.*, 2013).

There are variations as to what is defined as an RNA 3D motif. Here, we specifically discuss RNA base motifs that we define as a pattern of base arrangements that are annotated repeatedly in different locations either in the same RNA structure, or in different RNA molecules. Although there is a requirement for similarity in the 3D arrangement of the bases to be considered a motif, an alignment at the 1D sequence level is not required. In this article, we focus on computational methods that are able to identify base motifs and base-base interactions in the 3D structures of RNA.

RNA 3D motifs can be classified based on properties arising from the conformation of the RNA backbone and base pair interactions. Examples of conformational motifs include kink-turns, tetraloops, hook-turns, π turns, Ω turns and α loops (Klein *et al.*, 2001; Woese *et al.*, 1990; Szép *et al.*, 2003; Wadley and Pyle, 2004). However, for base motifs or base-base interactions, the arrangement similarities of the bases in 3D space are used for classification.

Background

Hydrogen bonds play crucial roles in the structures of biological macromolecules. The role of the hydrogen bond in protein structures is well recognized – an excellent example of their contribution towards orderly maintenance of 3-D structure is their core structure stabilization contributions in protein secondary structures. For example, the amino group nitrogen in an alpha helix backbone is the donor atom for a carbonyl group. In DNA, the two hydrogen bond interactions for A-T and the three hydrogen bonds between G-C that hold together the two strands of DNA in a double helix conformation as first described by Watson and Crick (1953) have become canon. In RNA structures, the base components of the nucleotides are also able to hydrogen bond in the canonical Watson-Crick arrangement. Additionally, the more diverse conformations that an RNA structure is able to fold into allow for the bases to interact in a plethora of arrangements that in turn result in a diverse array of base pairing arrangements as reported by Tinoco *et al.* (1973) as RNA are generally single-stranded; hence, less restricted to double helical conformation compared to DNA.

The work of Tinoco *et al.* (1973) expanded the repertoire of possible base pairings by considering other base interaction interfaces additional to the Watson-Crick interface (such as the sugar and Hoogsteen edges) and different geometries (such as the reverse Watson-Crick base pair). These additional base pair arrangements were; thus, termed non-canonical interactions that did not conform to the AT GC hydrogen bonding rule described by Watson and Crick (1953) in addition to possibly deviating from the ratio of Chargaff's rules (Chargaff, 1951) which is applicable to DNA content.

It became clear that these non-canonical interactions were not mere rare exceptions to the rule through work, such as the NCIR database (Nagaswamy *et al.*, 2002) by Fox and co-workers that catalogued a large number of non-canonical hydrogen bonded base interactions found in RNA structures reported in literature. Among the interactions in NCIR were arrangements that extended beyond simple base pairings to interactions consisting of three to five bases (see "Relevant Website section"). An extensive, manually curated literature sourced catalogue such as NCIR (Nagaswamy *et al.*, 2002); although useful, would not be able to provide an overview or proper accounting of the breadth of interactions that can probably be found in the RNA structures currently available in the PDB (Berman, 2008).

The increasing number of RNA structures in the PDB signalled the requirement for accurate computational means of identifying base interactions and the motifs they partake in. The ability to identify such base interactions and their locations in the 3D structure can be an important contributor in the function prediction or determination process. The increase in the number of RNA structures in the PDB such as the more recently available entries for ribosomal subunits (e.g., PDB ID: 6CFK), riboswitches (e.g., PDB ID: 6BFB), ribozymes (e.g., PDB ID: 5V3I), non-coding transcripts (e.g., PDB IDs: 5LYU, 5LSN) that are associated to human diseases ranging from cancer to infections are ripe targets for analysis of their base interactions in order to elucidate atomic level mechanistic details that can lead to potential therapeutic applications (Dasgupta *et al.*, 2017; Martinez-Zapien *et al.*, 2017; Podbevšek *et al.*, 2018; Rizvi *et al.*, 2018; Tereshchenkov *et al.*, 2018).

The computational analysis of RNA structures, specifically the interactions between and within RNA structures is not a new endeavor. However, due to the more recent availability of an increasing number of high resolution RNA structures, including very large complex assemblies, the computational analysis and comparisons of these structures in bulk is a nascent field that have the potential to be developed into pipelines to mine and develop RNA molecules as targets for therapeutic applications.

At this point, it must be made clear that an RNA base motif can consist of an arrangement of bases that are independent of their manner of hydrogen bonding. Nevertheless, using hydrogen bonds to define a motif is advantageous as there are clear definable parameters that can be calculated and these can also be used to differentiate different motifs that are otherwise visually similar.

The availability of high resolution RNA 3D structures enable the study of the basic interactions that stabilize the folded RNA molecule. In the early yeast tRNA structure, a network of hydrogen bond interactions was observed to run through most of the molecule with the exception of the exposed amino-acid and anticodon stems (Ladner *et al.*, 1975). Interaction networks in structured RNAs are therefore expected to feature prominently as a collective factor in stabilizing RNA structures. Among the interactions that can be found in such networks as listed by Lescoute and Westhof (2006) are: (i) phosphate-phosphate contacts mediated by water molecules or positively charged cations; (ii) phosphate-sugar hydrogen bonding usually mediated by the 2'-hydroxyl group; (iii) sugar-sugar hydrogen bonding interactions; (iv) base to phosphate or base to sugar hydrogen bonding; (v) base to phosphate or base to sugar stacking interactions; (vi) base to base hydrogen bonding interactions; (vii) base to base stacking interactions (Lescoute and Westhof, 2006).

Descriptions and Nomenclature of Base-Base Interactions

The base component of a nucleic acid can be divided into three edges: (i) the Watson-Crick face which is involved in the canonical Watson-Crick interactions, (ii) the sugar edge on the side of the glycosidic bond and (iii) the Hoogsteen edge (for purines) and CH edge (for pyrimidines) which are on the opposite side of the sugar face. Non-canonical base-base interactions therefore involve hydrogen bonding interfaced by one non-Watson-Crick base edge. Another type of non-canonical interaction may also occur which however involves the canonical Watson-Crick edges and was proposed by Francis Crick as the wobble-hypothesis (Crick, 1966). The orientation of bases to each other in RNA structures is not as easily described as can be done in DNA. The conformational diversity of RNA results in numerous other opportunities for bases to interact via non-canonical interactions as previously stated.

One method that can be used is to name the interactions based on the hydrogen bonding present. However, this is not as systematic nor does it provide an easy and immediate mental picture of the interaction. A nomenclature and classification system was proposed by Leontis and Westhof (2001) that enabled the diversity of possible base-base interactions to be better described. The basis of this nomenclature involved describing the base edges involved in the interaction in addition to the orientation of the glycosidic bond of one base to the other. We therefore propose that when descriptions of the arrangements are required, the Leontis and Westhof be used for uniformity and consistency.

Systems and/or Applications

Eleven servers that were found in the literature were compared for their utility in annotating base motifs or specific base-base interactions in RNA structures (Tables 1 and 2). The capacity of these different servers to find different types of base motifs and base-base interactions result in there being no single one stop solution for RNA structure annotation needs. These resources however complement each other in two ways – by using different methods for the searching and annotation and by covering different types of motifs that can be searched for and annotated. In general, all servers are able to accept as input either an existing PDB structure as a PDBID input or via an upload of the structure to be queried. Another feature shared by all the servers is the capability of providing visualization of the search output and making these output files available for downloads.

Analysis Approach and Utility of Base Arrangement Annotation

There are also a very limited set of tools that specifically annotates base arrangements or base-base interactions. In this analysis approach section, we compare and integrate these tools to those that take into account the folding of the RNA molecule into its substructure components, thus the services listed in Tables 1 and 2 do not necessarily carry out base motif and base interaction

Table 1 A listing of the eleven web servers and their URLs that can be used for annotating 3D motifs in RNA structures

Web server	URL
NASSAM	http://mfrlab.org/grafss/nassam
WebFR3D	http://www.bgsu.edu/research/rna/web-applications/webfr3d.html
ARTS	http://bioinfo3d.cs.tau.ac.il/ARTS/
RCLICK	http://mspc.bii.a-star.edu.sg/minhn/rclick.html
R3D-BLAST	http://genome.cs.nthu.edu.tw/R3D-BLAST/
RAG-3D	http://www.biomath.nyu.edu/?q=RAG3D
COGNAC	http://mfrlab.org/grafss/cognac
FASTR3D	http://genome.cs.nthu.edu.tw/FASTR3D/
iPARTS2	http://genome.cs.nthu.edu.tw/iPARTS2/
SARA	http://structure.biofold.org/sara/
SETTER	http://setter.projekty.ms.mff.cuni.cz

annotation specifically. This enables expanding the discussion into how such tools can be used together to extract the most amount of information from an available RNA 3D structure.

In perhaps the majority of cases needing annotation of base arrangements in an RNA structure, the investigator starts with a newly solved structure, usually determined by X-ray diffraction. Tools that enable such structures to be annotated can thus provide a reference point for the investigator during analysis and visual inspection, especially for RNA structures that may involve complex interactions such as the ribosomal subunits. Unlike the redundancy often seen for sequence analysis programs, those that analyse at the structural level implement different approaches and were designed to find different targets thus resulting in the various servers being complementary to each other.

As an example, submitting a structure to the NASSAM server can provide a reference map for the investigator to identify known motifs. However, submitting the same structure to RAG-3D will retrieve a different set of results related to conserved topology that were not covered by NASSAM. The majority of these services annotate the RNA structure using a reference database thus limiting searches to known motifs and interactions.

In order to enable novel motif discovery, algorithms that are not dependent on a reference dataset need to be used. One method of discovering novel motifs would be to compare structures in order to look for unreported similar base arrangements or interactions present in multiple structures that may signal novelty. Such structure to structure annotations are possible by services such as ARTS and COGNAC, while some others actually require two or more structures as the input, such as RCLICK (see [Table 3](#)).

The use of hydrogen bonding patterns can also help differentiate between patterns. Subtle changes in the hydrogen bonding networks in the same structures that have undergone mutations or have been solved (crystallized) under different conditions can be used to identify structural shifts resulting from changes in the interactions between the bases. Furthermore, a highly hydrogen bonded grouping of bases that are repeated in different structures can imply that it may have functional or structural stabilization roles. Annotating such base interactions can thus be another means of identifying novel motifs.

Since most of these programs search and annotate an actual occurrence as opposed to being a prediction, what remains for the user to determine is how much the annotation conforms to the intended query. It is still possible to identify novel 3D motifs from such searches because the retrieved hits may deviate from the original query and may represent a yet unreported base arrangement. This was demonstrated by [Firdaus-Raih *et al.* \(2011\)](#) using the NASSAM program.

RNA structure annotation programs can also be used as a means of high-throughput structure annotation. The InterRNA database is an example of how the NASSAM and COGNAC algorithms were used to annotate high resolution RNA structures in the PDB ([Appasamy *et al.*, 2015](#)). The RNA 3D Motif Atlas is the result of annotating a representative set of RNA 3D structures using FR3D ([Sarver *et al.*, 2008](#)), which is available as the WebFR3D service. The DARTS database ([Abraham *et al.*, 2008](#)) is derived from structural comparisons using the ARTS program.

Although structural genomics level approaches for RNA chain containing structures have never been attempted, these tools will allow for rapid and systematic annotation for the output of such projects. Due to the different but complementary nature of the available services, selection of the correct analysis program will depend on the objectives of the investigator.

Illustrative Example(s) or Case Studies

In order to demonstrate the RNA structure annotation process, specifically that of base motifs and base interactions, we have used the structure of a T box riboswitch (PDB ID 4MGN) ([Grigg and Ke, 2013](#)) as a query that was submitted to the services listed in [Table 1](#). For services where a minimum of two structures needed to be submitted, the structure of another T box riboswitch (PDB ID 4LCK) ([Zhang and Ferré-D'amaré, 2013](#)) was also provided. The differences in the types of output and what each service is capable of presenting is provided in [Table 3](#).

It is useful to note that novel motifs can also arise from a known or previously characterised arrangement in a different structural context. One such example is where a UAU Hoogsteen, Watson-Crick base triple, which is a known arrangement, was

Table 2 Comparison of web servers for RNA 3D motifs and substructures searching. A comparison of the eleven web servers for 3D motif searching in terms of their input, algorithm and output

[illegible]

Table 3 Comparison of the various results from the different tools for a T box riboswitch (PDB ID 4MGN) used as an input query. For services that require at least two input structure, the PDB entry for another T box riboswitch (PDB ID 4LCK) was also provided

Search	Server	
Motifs/substructures/ patterns	NASSAM	Returned one annotation of an A-minor motif and four annotations of base triple.
	WebFR3D	The server was unavailable for a period of time during this comparison due to repairs and upgrades.
	R3D- BLAST	The webserver searched one chain at a time; thus needing the search to be executed twice - once for each chain Chain A: Returned 12 hits of similar structures compared to 4MGN_A. Chain B: Returned 511 hits of similar structures compared to 4MGN_B.
	RAG-3D	Similar to the R3D-BLAST option, the RAG-3D searched only one chain at a time. Therefore, the search is done in two times by using Chain A and Chain B. Chain A (glyQS T box riboswitch): Returned 10 hits of similar structures compared to 4MGN_A and 90 hits of similar substructures compared to substructures of 4MGN_A. The similar substructures resulted from the nine subgraphs of the query topology (4MGN_A) that represented the input structure. Chain B (tRNA-glycine): Resulted 10 hits of similar structures compared to 4MGN_B and no substructure reported. The 10 hits from the 4MGN_B searched are different from the search of 4MGN_A.
	COGNAC	The webserver provides options to annotate the clusters of bases that are connected by hydrogen bonds in a structure or to annotate and compare two input structures. In this case study we annotate and compare two input structures which are 4MGN and 4LCK. COGNAC returned 15 annotations of base triples, six annotations of quadruple base Type 1, an annotation of quadruple base Type 2, two annotations of quintuple base Type 1, an annotation of quintuple base Type 2 and an annotation of sextuple base Type 1 for 4MGN. Compared to 4LCK, COGNAC returned 10 annotations of base triples and two annotations of quadruple base Type1. An annotation of base triples of 4MGN and 4LCK annotated at similar location in tRNA glycine's o-arm and variable loop.
	FASTR3D	No results returned – the 4MGN structure was not in the database used and no option to upload structures is available.
Alignment of RNA tertiary structures	ARTS	The webserver is temporary unstable.
	RCLICK	The server needs two input structures to execute the alignment. 4MGN is compared to 4LCK. The results of two structures are viewed using JSMol viewer. Rclick shows the alignment of RNA interactions between both input structures. Based on the results of the output, the value of the RMSD given is 2.07 Å. The hits of match residues for 4mgn when compared to 4LCK is 105 atoms. The server also returns with structure overlap of 33.55% and highlighted the aligned residues of the input structures using JSMol viewer.
	iPARTS2	The webserver requires the structures to be searched only one chain at the time and need two structures to be executed. Thus, the global alignment search is done between 4MGN and 4LCK with different chains. Chain IDs with T box riboswitch structure: Chain C of 4MGN is superimposed with chain C of 4LCK and returned with 84 hits of aligned residue pairs. The RMSD obtained from the alignment is 4.490 Å. Chain IDs with tRNA-glycine: Chain B of 4MGN is superimposed with chain B of 4LCK and returned with 68 hits of aligned residue pairs. The RMSD value resulted with 2.961 Å.
	SARA	Provides two options of search that are structure-based function assignment and pair-wise structure alignment. The case study search is done using the structure-based function assignment because it required an input of file in PDB format to be searched throughout the SCOR database with the parameter to also search against RNA structures that are larger than 1000 nucleotides. Returned 192 structures sorted based on the mean of the negative logarithm of the three <i>P</i> -values (MEANLN). The three <i>P</i> -values are negative logarithm of the <i>P</i> -value of the sequence alignment score (LNPID), negative logarithm of the <i>P</i> -value of the secondary structure alignment score (LNPSS) and negative logarithm of the <i>P</i> -value of the structure alignment score (LNPSI). (LNPSI).
	SETTER	The webserver provides options for the structures to be aligned as a whole structure or based on the selected chain. In this case study we aligned 4MGN and 4LCK for the whole structure and based on the selected chain (chain C) for both structures. The chain C represents the T-box riboswitch. The SETTER provides the alignment quality measures, which are RMSD, the percentage of structural identity (PSI), the percentage of sequence identity (PID), the number of aligned nucleotides and the number of exact base matches. The alignment quality for 4MGN and 4LCK are 2.209 for the RMSD, 0.060 for the PSI, 0.019 for the PID, 19 nucleotides were aligned and 6 bases matches. The alignment quality for the chain C for 4MGN and 4LCK are 1.794 for the RMSD, 0.233 for the PSI, 0.070 for the PID, 20 nucleotides were aligned and 6 bases matches.

found by the computer program NASSAM to be in a stacked arrangement on top of each other at a junction holding three subdomains of domain V in the prokaryotic 23S subunit and was found to be conserved in all available 23S subunit structures at the time it was reported (Firdaus-Raih *et al.*, 2011). In this particular case, annotation of the base triples led to the discovery that there were two of the same triples stacked together – such an arrangement may not be obvious even on close visual inspection due to the dense packing of nucleotides and the difficulty in discerning 3D motifs when visualizing these structures.

We further used the COGNAC service to compare and identify atomic level differences of base arrangements connected by hydrogen bonding interactions. This was demonstrated by using the structures of two mutant cyclic dinucleotide bis-(3'-5')-cyclic dimeric guanosine monophosphate (c-di-GMP-I) riboswitch that were bound to GpA (PDB ID 3UD4) and pGpA (PDB ID 3UD3) (Smith *et al.*, 2012). Due to space limitations, we selected only the base triple interactions to illustrate the differences between the structures. In 3UD4, six base triples interactions were annotated: U53.C55.G85, C17.C93.G14, G97.C13.G94, G94.C93.G14, G94.C93.G17 and C93.G94.C13; while in 3UD3, three base triple interactions were annotated: U53.C55.G85, C17.C93.G14, G97.C13.G94 (Fig. 1). Three of the base triples interactions in 3UD3 can be found in 3UD4, however a slight shift in the base arrangement between C93 and C94 in 3UD3 resulted in the loss of the hydrogen bonds between atom N4 of C93 and atom O6 of G94 (Fig. 1).

The differences in base interactions detected occurred on the opposite end to where the ribonucleoprotein interactions were occurring and closer to the sites of where pGpA and GpA binding occurred. This can imply those interactions differences observed may be due to the differences brought about by interactions with these two different molecules. However, as always, care must be taken when interpreting such data because they can also be experimental artifacts (i.e., crystallographic packing, X-ray resolution). Nevertheless, the differences can provide clues that can assist in forming hypotheses and thus lead towards experiments that can be more conclusive.

Discussion and Future Directions

There is a tendency to concentrate on descriptions and cataloguing with little reference to the relevance interactions in a functional context when computational means are deployed to annotate RNA structures. Perhaps this is not unexpected due to the need to first develop such capacity. However, the future direction of developments in this field should take into consideration more intensive integration of the mechanistic insights that can be gained from the annotation of structural motifs, particularly for base motifs and base interactions.

As larger and more complex structures, such as the ribosomal assemblies, are solved via crystallographic and Cryo-EM approaches, there is a clear need for computational approaches that can accurately annotate these structures and provide a means of comparison between them. In the case of NMR structures, the atomic level differences between the different models can be

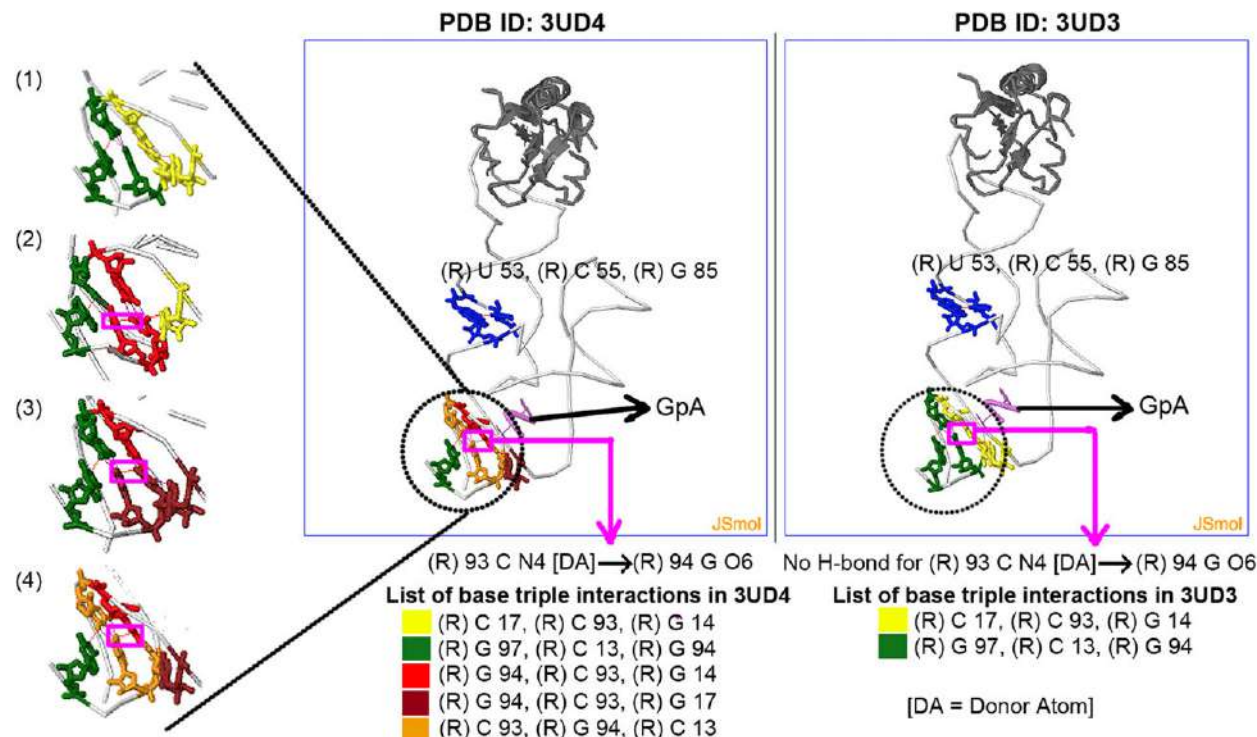


Fig. 1 Comparison and identification base arrangements arising from differences in the base-base hydrogen bonding interactions. The comparison and identification of the atomic level differences of base arrangements connected by hydrogen bonding interactions using the COGNAC service. This was demonstrated by using the structures of two mutant cyclic dinucleotide bis-(3'-5')-cyclic dimeric guanosine monophosphate (c-di-GMP-I) riboswitch that were bound to GpA (PDB ID 3UD4) and pGpA (PDB ID 3UD3).

quickly annotated and discerned. Such annotations can provide a useful mapping of the differences in base interactions and arrangements in the different models and correlated to biological or experimental context. Additionally, the ability to track the changes in the annotation, especially for NMR models, can be especially useful for tracking the dynamics of the RNA conformation with regard to the formation and/or deformation of base interactions and the effects these changes have on biological function.

Acknowledgement

MFR is funded by the Ministry of Science, Technology and Innovation grant 02-01-02-SF1278 and Universiti Kebangsaan Malaysia ICONIC-2013-007 and GUP-2014-008 grants.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Characterizing and Functional Assignment of Noncoding RNAs. Engineering of Supramolecular RNA Structures. Natural Language Processing Approaches in Bioinformatics

References

- Abraham, M., Dror, O., Nussinov, R., Wolfson, H.J., 2008. Analysis and classification of RNA tertiary structures. *RNA* 14 (11), 2274–2289.
- Almutairi, M.M., Svetlov, M.S., Hansen, D.A., *et al.*, 2017. Co-produced natural ketolides methymycin and pikromycin inhibit bacterial growth by preventing synthesis of a limited number of proteins. *Nucleic Acids Research* 45, 9573–9582.
- Appasamy, S.D., Hamdani, H.Y., Ramlan, E.I., Firdaus-Raih, M., 2015. InterRNA: A database of base interactions in RNA structures. *Nucleic Acids Research* 44, 266–271.
- Berman, H.M., 2008. The Protein Data Bank: A historical perspective. *Acta Crystallographica Section A: Foundations of Crystallography* 64, 88–95.
- Bindewald, E., Hayes, R., Yingling, Y.G., Kasprzak, W., Shapiro, B.A., 2007. RNAJunction: A database of RNA junctions and kissing loops for three-dimensional structural analysis and nanodesign. *Nucleic Acids Research* 36, 392–397.
- Capriotti, E., Marti-Renom, M.A., 2010. Quantifying the relationship between sequence and three-dimensional structure conservation in RNA. *BMC Bioinformatics* 11, 322.
- Chargaff, E., 1951. Structure and function of nucleic acids as cell constituents. *Federation Proceedings*, 654–659.
- Chojnowski, G., Walen, T., Bujnicki, J.M., 2014. RNA Bricks – a database of RNA 3D motifs and their interactions. *Nucleic Acids Research* 42, 123–131.
- Crick, F.H., 1966. Codon – anticodon pairing: The wobble hypothesis. *Journal of Molecular Biology* 19, 548–555.
- Dasgupta, S., Suslov, N.B., Piccirilli, J.A., 2017. Structural basis for substrate helix remodeling and cleavage loop activation in the Varkud satellite ribozyme. *Journal of the American Chemical Society* 139, 9591–9597.
- Firdaus-Raih, M., Harrison, A.-M., Willett, P., Artymiuk, P.J., 2011. Novel base triples in RNA structures revealed by graph theoretical searching methods. *BMC Bioinformatics* 12 (Suppl. 13), S2.
- Grigg, J.C., Ke, A., 2013. Structural determinants for geometry and Information Decoding of tRNA by T Box Leader RNA. *Structure* 21, 2025–2032.
- Huang, W., Thomas, B., Flynn, R.A., *et al.*, 2015. DDX5 and its associated lncRNA Rmrp modulate Th17 cell effector functions. *Nature* 528, 517–522.
- Klein, D.J., Schmeing, T.M., Moore, P.B., Steitz, T.A., 2001. The kink-turn: A new RNA secondary structure motif. *The EMBO Journal* 20, 4214–4221.
- Koizumi, H., Suzuki, R., Tachibana, M., *et al.*, 2016. Importance of determination of Crystal Quality in Protein Crystals when Performing High-Resolution Structural Analysis. *Crystal Growth & Design* 16, 4905–4909.
- Krasilnikov, A.S., Mondragón, A., 2003. On the occurrence of the T-loop RNA folding motif in large RNA molecules. *RNA* 9, 640–643.
- Ladner, J.E., Jack, A., Robertus, J.D., *et al.*, 1975. Structure of yeast phenylalanine transfer-RNA at 2.5 Å resolution. *Proceedings of the National Academy of Sciences of the United States of America* 72 (11), 4414–4418.
- Leontis, N.B., Westhof, E., 2001. Geometric nomenclature and classification of RNA base pairs. *RNA* 7, 499–512.
- Leontis, N.B., Zirbel, C.L., 2012. Nonredundant 3D Structure Datasets for RNA Knowledge extraction and benchmarking. In: Leontis, N., Westhof, E. (Eds.), *RNA 3D Structure Analysis and Prediction*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Lescaute, A., Westhof, E., 2006. The interaction networks of structured RNAs. *Nucleic Acids Research* 34, 6587–6604.
- Martinez-Zapien, D., Legrand, P., McEwen, A.G., *et al.*, 2017. The crystal structure of the 5p' functional domain of the transcription riboregulator 7SK. *Nucleic Acids Research* 45, 3568–3579.
- Mauger, D.M., Siegfried, N.A., Weeks, K.M., 2013. The genetic code as expressed through relationships between mRNA structure and protein function. *FEBS Letters* 587 (8), 1180–1188.
- Nagaswamy, U., Larios-Sanz, M., Hury, J., *et al.*, 2002. NCIR: A database of non-canonical interactions in known RNA structures. *Nucleic Acids Research* 30 (1), 395–397.
- Nissen, P., Ippolito, J.A., Ban, N., Moore, P.B., Steitz, T.A., 2001. RNA tertiary interactions in the large ribosomal subunit: The A-minor motif. *Proceedings of the National Academy of Sciences* 98 (9), 4899–4903.
- Parlea, L.G., Sweeney, B.A., Hosseini-Asanjan, M., Zirbel, C.L., Leontis, N.B., 2016. The RNA 3D Motif Atlas: Computational methods for extraction, organization and evaluation of RNA motifs. *Methods* 103, 99–119.
- Podbevšek, P., Fasolo, F., Bon, C., *et al.*, 2018. Structural determinants of the SINE B2 element embedded in the long non-coding RNA activator of translation AS Uchl1. *Scientific Reports* 8, 3189.
- Prokhorova, I., Altman, R.B., Djumagulov, M., *et al.*, 2017. Aminoglycoside interactions and impacts on the eukaryotic ribosome. *Proceedings of the National Academy of Sciences of the United States of America* 114, E10899–E10908.
- Rizvi, N.F., Howe, J.A., Nahvi, A., *et al.*, 2018. Discovery of selective RNA-Binding small molecules by affinity-selection mass spectrometry. *ACS Chemical Biology* 13 (3), 820–831.
- Sarver, M., Zirbel, C.L., Stombaugh, J., Mokdad, A., Leontis, N.B., 2008. FR3D: Finding local and composite recurrent structural motifs in RNA 3D structures. *Journal of Mathematical Biology* 56 (1–2), 215–252.
- Smith, K.D., Lipchick, S.V., Strobel, S.A., 2012. Structural and biochemical characterization of linear dinucleotide analogs bound to the c-di-GMP-I aptamer. *Biochemistry* 51 (1), 425–432.
- Szép, S., Wang, J., Moore, P.B., 2003. The crystal structure of a 26-nucleotide RNA containing a hook-turn. *RNA* 9, 44–51.
- Tamura, M., Hendrix, D.K., Klosterman, P.S., *et al.*, 2004. SCOR: Structural Classification of RNA, version 2.0. *Nucleic Acids Research* 32, 182–184.
- Tereshchenkov, A.G., Dobosz-Bartoszek, M., Osterman, I.A., *et al.*, 2018. Binding and action of amino acid analogs of chloramphenicol upon the bacterial ribosome. *Journal of Molecular Biology* 430 (6), 842–852.
- Tinoco, I., Borer, P.N., Dengler, B., *et al.*, 1973. Improved estimation of secondary structure in ribonucleic acids. *Nature* 246, 40–41.

- Wadley, L.M., Pyle, A.M., 2004. The identification of novel RNA structural motifs using COMPADRES: An automated approach to structural discovery. *Nucleic Acids Research* 32, 6650–6659.
- Wang, Y., Zirbel, C.L., Leontis, N.B., Ding, B., 2018. RNA 3-dimensional structural motifs as a critical constraint of viroid RNA evolution. *PLOS Pathogens* 14, e1006801.
- Watson, J.D., Crick, F.H.C., 1953. The structure of DNA. *Cold Spring Harbor Symposia on Quantitative Biology* 18, 123–131.
- Woese, C.R., Winker, S., Gutell, R.R., 1990. Architecture of ribosomal RNA: Constraints on the sequence "tetra-loops". *Proceedings of the National Academy of Sciences of the United States of America* 87 (21), 8467–8471.
- Zhang, J., Ferré-D'amaré, A.R., 2013. Cocystal structure of a T-box riboswitch Stem I domain in complex with its cognate tRNA. *Nature* 500 (7462), 363–366.

Relevant Website

http://prion.bchs.uh.edu/bp_type/bp_structure.html

Department of Biology and Biochemistry, University of Houston, Texas, USA.

Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights

Chyan Leong Ng and Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

RNA

The Central Dogma of Molecular Biology (Crick, 1958) has long recognized messenger RNA (mRNA) as an intermediary molecule for decoding genetic information to functional molecules. Along with transfer RNA (tRNA) and ribosomal RNA (rRNA), the roles of RNA molecules were primarily linked to protein synthesis. However, the current knowledge of RNA species has extended beyond mRNA, tRNA, and rRNA. Other RNA transcripts that are not translated to proteins (non-protein coding or ncRNA) such as microRNA (miRNA), small interfering RNA (siRNA), piwi-interacting RNA (piRNA) and long non-coding RNA (lncRNA). They partake in numerous other important cellular roles that include posttranscriptional modification and gene expression regulation. RNA molecules can be self-folded to form complex three-dimensional (3D) shapes that in turn allow them to perform functions similar to proteins such as catalysis. Examples of such RNA enzymes, termed ribozymes, were discovered in the 1980s (Guerrier-Takada *et al.*, 1983; Kruger *et al.*, 1982). Such independent catalytic capacity lends support to the RNA world hypothesis (Robertson and Joyce, 2012; Cech, 2012; Gilbert, 1986).

All RNA is transcribed through the complex machinery of RNA polymerase (RNAP). Bacteria uses a single type of RNAP for all RNA transcription while eukaryotic cells generally contain RNA polymerases I, II, and III. RNA polymerase I is responsible for the synthesis of ribosomal RNA (rRNAs) 28S, 18S, and 5.8S; RNA polymerase II transcribes mRNA, snRNA, and microRNA; RNA polymerase III is responsible for transcribing tRNA, 5S rRNA, and other small RNAs.

RNA Sequence and Structure Analysis

There are three levels of RNA structure – Primary, secondary, and tertiary. The primary structure can be obtained from the DNA sequence. However, the actual RNA polynucleotide may contain posttranscriptional modifications of the sugar moieties and nucleotide bases, as well as the results of splicing such as in the case of pre-mRNA (Batey *et al.*, 1999) that are not present in the DNA sequence. A single-stranded RNA is able to form different types of secondary structure elements including stems, bulges, pseudoknots, hairpin loops, internal loops, and multiloops. These elements will then fold to form the 3D structures that are able to execute their intended biological functions. Similar to proteins, the sequence composition and resulting 3D structure of the folded RNA molecule dictate their biological functions. For example, although mRNA is generally regarded as a single-stranded sequence, its local sequence patterns are able to form secondary structures that have been shown to have important roles in regulating protein expression (Kudla *et al.*, 2009; Hall *et al.*, 1982).

Studies into the relationships of the sequences to the secondary and tertiary structure of tRNAs in the 1960s were important milestones that enabled RNA structure prediction. The tRNA molecules, which generally consist of ~76–90 nucleotides, were shown to form a conserved cloverleaf secondary structure with four helical segments that accommodate anticodon, acceptor, D, and T arms despite sharing low sequence identities (Rajbhandary *et al.*, 1967; Madison *et al.*, 1966; Holley *et al.*, 1965). These secondary structure predictions were confirmed with the determination of the three-dimensional (3D) structure of RNA for yeast alanine transfer RNA (Kim *et al.*, 1974; Robertus *et al.*, 1974; Kim *et al.*, 1973; Holley *et al.*, 1965).

The knowledge that RNA secondary structures can be evolutionarily conserved despite low sequence identity spurred the development of RNA secondary structure prediction methods. The advent of these methods led to the prediction of the secondary structures of the ribosomal RNAs - the 5S, 16S, and 23S subunits (Noller and Woese, 1981; Noller *et al.*, 1981; Woese *et al.*, 1980; Fox and Woese, 1975). The RNA secondary structure prediction algorithms were further improved by using a combination of sequence comparison analysis, thermodynamic parameters that quantified the minimum free energy of structure folds, chemical modification experiments, and pseudoknot predictions (Reuter and Mathews, 2010; Mathews *et al.*, 2004, 1999, 1997; Lück *et al.*, 1996; Zuker and Stiegler, 1981). Other parameters, including the implementation of the centroids Boltzmann-weighted ensemble model (Ding *et al.*, 2008) and maximization of the expected base-pair accuracy (Lu *et al.*, 2009), were able to further improve the secondary structure predictions. These algorithms have now been implemented as web accessible tools allowing for users to generate RNA secondary structures from sequence data – Examples of such tools are the Vienna RNA secondary structure server (see “Relevant Websites section”) (Hofacker, 2003) and RNAstructure (see “Relevant Websites section”) (Bellaousov *et al.*, 2013).

Although the predicted secondary or tertiary structures of RNA can be confidently used to provide biological function information, experimentally determined RNA structures remain crucial in providing atomic level insights on the mechanisms of specific RNA molecules. The first crystal structure of tRNA that was determined in 1970s provided the evidence that allowed for the validation of the interactions predicted in the cloverleaf secondary structure model. The tRNA molecule was folded into an L-shaped molecule with the nucleotide components stabilized via hydrogen bonding and base stacking, thus further lending

credence to the hypothesis that all tRNA may share a similar structure due to the specific position requirement of the anticodon loop and acceptor stem (Kim *et al.*, 1974; Robertus *et al.*, 1974; Kim *et al.*, 1973).

Since then, many 3D structures of RNA molecules such as the hammerhead ribozyme, group I ribozyme domain, hepatitis delta virus ribozyme, ribosomal RNAs, and small nuclear RNA (Nguyen *et al.*, 2016; Yan *et al.*, 2015; Ogle *et al.*, 2001; Ban *et al.*, 2000; Schluenzen *et al.*, 2000; Wimberly *et al.*, 2000; Ferré-D'Amaré *et al.*, 1998; Cate *et al.*, 1996; Scott *et al.*, 1995) have been determined. The availability of a wider set of experimentally determined 3D structures had also allowed for progress to be made in the development of methods for RNA fold and motif analysis (Butcher and Pyle, 2011; Klosterman *et al.*, 2002; Batey *et al.*, 1999; Ferré-D'Amaré and Doudna, 1999).

Due to the difficulty of experimentally acquiring high-resolution three-dimensional structures of RNA molecules using either nuclear magnetic resonance spectrometry (NMR), X-ray crystallography, or electron microscopy (Cryo-EM), computational methods that can predict RNA structural information and allow for their comparisons, are useful tools that can take advantage of the availability of the numerous genome sequences.

These computational approaches do share some similarity to 3D structure prediction of proteins in using approaches that include fragment assembly (including Monte-Carlo algorithms) such as for the programs FARNA, DMD, FARFAR (Das *et al.*, 2010; Ding *et al.*, 2008; Das and Baker, 2007), molecular dynamics simulations used in iFoldRNA (Sharma *et al.*, 2008); and the nucleotide cyclic motif employed by MC-Fold and MC-Sym (Parisien and Major, 2008). Many other 3D RNA structure prediction tools can be found in the “Relevant Websites section”.

The key function of the ribosome to synthesize proteins is conserved in all domains of life including organelles. Due to this, the structure of the ribosome is also generally conserved despite variations in the nucleotide sequence (Table 1). This allows for the study of how these sequences evolved while maintaining the ultimate requirement of needing to preserve the end structure that can be formed. In this article, we explore the bioinformatics tools of varying utility that can be used to compare RNA sequences and structures to provide functional and evolutionary insights.

Background

Ribosomal rRNA

More than half of the cellular RNA content is rRNAs (Russell and Zomerdijs, 2006). The rRNAs form the main structural components of the ribosome's protein synthesis machinery. Generally, two-thirds of the ribosome consists of rRNA while the remainder is made up of ribosomal proteins; however, an opposite ratio exists for mammalian mitochondrial ribosomes (Sharma *et al.*, 2003). Ribosomes are assembled from two ribosomal subunits, the small ribosomal subunit (SSU), specifically the 30S for prokaryotic cells and 40S for eukaryotic cells, and the large ribosomal subunit (LSU), specifically 50S for prokaryotic cells and 60S for eukaryotic cells. The two ribosomal subunits form the 70S and 80S ribosomes in prokaryotes and eukaryotes, respectively, while mitochondrial organelles contain 55S ribosomes. Each subunit contains a number of ribosomal proteins and rRNAs as shown in Table 1. The rRNA molecules have extensive secondary structure that together with the ribosomal proteins assemble into the large RNA–protein complex that is the ribosome.

Table 1 General components of ribosome

	Bacteria ^a	Archaea ^a	Chloroplast ^{b, c}	Eukaryote ^a	Mitochondria	
					Mammalian ^a	Fungi ^d
Ribosome		70S		80S	55S	67S–74S
Molecular weight (MDa)		2.3–2.6		3.3–4.5	~2.7	3–3.3
Small subunit						
Sedimentation coefficient	30S		30–35S	40S	28S	37S
Molecular weight (MDa)	0.8		1–2	1.4	1.2	1.1
SSU rRNAs		16S		18S	12S	15S
Number of ribosomal protein		20–33		32	33	34
Large subunit						
Sedimentation coefficient (S)	50S			60S	39S	57S
Molecular weight (MDa)	1.6		2–3	2.6	2.4	1.9
LSU rRNAs	23S, 5S		23S, 5S, 4.5S	26S, 5.8S, 5S	16S	21S
Number of ribosomal protein		34–40		46	52	39

^aPoehlsgaard and Douthwaite (2005).

^bHooper (2012).

^cOlinares *et al.* (2010).

^dAmunts *et al.* (2014).

Bioinformatics Tools in Ribosomal SSU rRNA Sequence and Structure Studies

The SSU rRNA is involved in decoding of mRNA (Carter *et al.*, 2000; Clemons *et al.*, 1999) while the large ribosomal subunit rRNA (LSU rRNA) is the site of the peptidyl transferase center (PTC) with ribozyme activity that catalyzes the formation of the peptide bond during protein synthesis (Ban *et al.*, 2000; Noller *et al.*, 1992).

SSU rRNA Sequence Studies

Sequence and phylogenetic studies of the SSU rRNAs have been intensively carried out for the past few decades (Cannone *et al.*, 2002; Gutell *et al.*, 1994; Woese, 1987; Gutell *et al.*, 1985) with the first 16S rRNA completely sequenced for *E. coli* in 1978 (Brosius *et al.*, 1978). The SSU rRNA sequence is particularly useful for phylogenetic studies because it is generally accepted that the SSU rRNA genetic material does not undergo horizontal gene transfer. Furthermore, each domain of life contains universally conserved sequences within a close phylogenetic distance while at the same time containing variable regions as a result of evolution that are significant enough to serve as molecular markers to distinguish eukarya, bacteria, and archaea or between species in the same domain of life (Woese *et al.*, 1990). There are nine commonly known conserved regions (C1–C9) and nine variable regions (V1–V9) in SSU rRNA, although the variability for some of the variable regions differ between eukaryotes and prokaryotes (Hadziavdic *et al.*, 2014; Neefs *et al.*, 1993).

Considering that structural elements may play a role in evolution, analysis of evolutionary rates across the entire rRNA has shown that structural elements in the rRNA do evolve with different rates, and this rate also varies between the different domains of life. For instance, nucleotides at the conserved regions were found mainly unpaired while sequences that form stems in bacteria or loops in eukaryotes tend to evolve faster. At the 3D structure level, the regions that are important for protein synthesis including the decoding center of SSU rRNA and the PTC of LSU rRNA were consistently more conserved than the residues on the surface of the ribosome thus suggesting that the structurally dependent functional roles of these nucleotides are the reason for their conservation (Smit *et al.*, 2007; Wuyts *et al.*, 2001; Woese *et al.*, 1980).

There are approximately 6 million SSU rRNA (16S/18S) sequences available in the high quality ribosomal RNA databases as of December 2017 in the SILVA Database (Quast *et al.*, 2012) (see “Relevant Websites section”). The availability of such a vast number of rRNA sequences has permitted faster and more accurate analysis of rRNA using sequence alignment tools. For instance, one can apply BLASTN (Basic Local Alignment Search Tool) (Altschul *et al.*, 1990) to find the species or closest homolog of a newly acquired rDNA sequence. If the analysis entails the comparison of several selected rRNA sequences from different species, multiple sequence alignment (MSA) programs such as Clustal W (Larkin *et al.*, 2007; Thompson *et al.*, 1994), T-Coffee (Tree-based Consistency Objective Function for alignment Evaluation) (Notredame *et al.*, 2000), and MAFFT (Katoh and Standley, 2013; Katoh *et al.*, 2002) can be used. A phylogenetic tree can then be constructed from the output of the MSA results. Nonetheless, these sequence alignment programs depend only on the primary sequence similarity of rRNA without considering any features of the evolutionarily conserved secondary structures.

Alignment programs that integrate covariation analysis and specifically intended for use with rRNA sequences that integrate covariation analysis have also been developed. For example, the Infernal (INFERence of RNA Alignment) program has been designed to align rRNA more accurately by applying a model-based approach using covariance models that can recognize covariations or patterns of conserved base pairs in the rRNA secondary structure (Nawrocki and Eddy, 2013; Eddy and Durbin, 1994). The concepts and methods associated with RNA comparative sequence analysis and structure prediction can also be obtained at <http://www.ma.cbb.utexas.edu/CAR/1D/> (Cannone *et al.*, 2002). To deal with the high volume of rRNA sequences from the different species available, programs such as SINA (SILVA Incremental Aligner) (Pruesse *et al.*, 2012) have been developed.

The available SSU rRNA sequences are an important resource for studying evolution that are housed in several high quality updated SSU rRNA databases that include SILVA (Quast *et al.*, 2012), RDP (Cole *et al.*, 2013), Greengenes (McDonald *et al.*, 2012), and National Center for Biotechnology Information (NCBI) (Federhen, 2011). To ease the phylogenetic study of SSU rRNA sequences, including those that are newly sequenced from previously unstudied organisms, one can apply comprehensive software packages like SSU-ALIGN (Nawrocki, 2009) that contain tools to identify, analyze, perform secondary structure alignment, and visualize SSU rRNA. A rRNA sequence identified from a newly sequenced genome also can be annotated using programs such as RNAmmer (Lagesen *et al.*, 2007) and HMMER 3.0 (Huang *et al.*, 2009) or web-based tools such as EzCloud (see “Relevant Websites section”) (Yoon *et al.*, 2017).

SSU rRNA Secondary Structure Predictions

Pairwise and MSAs of SSU rRNA are not able to provide a structural level insight for the sequences being compared. However, secondary structure prediction is crucial for providing a higher level of understanding with regard to evolutionary and functional conservation of SSU rRNA. Unlike RNA folding algorithms that predict secondary and tertiary structures based on their global minimum energy or by using free energy minimization that involves thermodynamic parameters (Mathews *et al.*, 1999; Zuker, 1989; Zuker and Stiegler, 1981), comparative analysis of rRNA secondary structures is based on the principle that RNA with different sequences are able to form similar structures (secondary and tertiary) and the unique structures with functional features will be evolutionarily conserved (Gutell *et al.*, 2002; Woese *et al.*, 1983, 1980; Noller *et al.*, 1981). The free energy minimization

approach was reported to be less accurate for generally long RNA secondary structure prediction including SSU rRNA (Doshi *et al.*, 2004). The accuracy of covariation based rRNA comparative analysis was further improved with the availability of a high number of rRNA sequences (Gutell *et al.*, 2002, 1986; Gutell, 1996).

In the 1980s, the secondary structures of 16S rRNA were predicted and described using comparative sequence analysis and combined to data of chemical modifications and enzymatic assays (Gutell *et al.*, 1985; Woese *et al.*, 1983, 1980). In general, a prokaryote 16S secondary structure that consists of about 1550 nucleotides can be arranged into several domains – including 5' domain, central domain, 3' major domain, and 3' minor domain – which assemble into 50 helices for SSU rRNA. Comparisons of SSU rRNA secondary structures for prokaryotes, eukaryotes, and organelles have revealed core structural features that are conserved in all SSU rRNA, and in addition, unique structural features for each domain of life or organelles were also identified. For example, 18S rRNA is more complex with less regular helices and bigger loops compared to bacterial 16S rRNA (Woese *et al.*, 1983). Integrating sequence alignments to secondary and tertiary structure analysis of SSU RNAs further revealed unique features of SSU rRNA conserved and variable regions that can be applied to distinguish a species at the molecular level and for categorization as eukarya, bacteria, or archaea (Woese *et al.*, 1990; Woese, 1987).

For instance, all bacterial SSU rRNAs were found to have a side bulge with six nucleotides at the hairpin loop (between nucleotide 500 and 545, *E. coli* numbering) that protrudes from the stalk of the rRNA structure. However, this same side bulge in archaea and eukarya were formed with seven nucleotides. Several of these distinct features were further identified to distinguish the rRNA of compared organisms, which can help in building a universal phylogenetic tree (Woese *et al.*, 1990, 1983; Gutell *et al.*, 1985). Furthermore, analysis of SSU rRNAs from organelles such as the 16S rRNA of chloroplast and the 12S rRNA of mitochondria also revealed substantial similarities of the primary sequences and at secondary structure level despite being of very different lengths thus suggesting that the core sequence and structure of rRNA is also conserved in the ribosomes of organelles (Zwieb *et al.*, 1981). In knowing that the secondary structures of SSU rRNA are more conserved than the sequence, differences in the secondary structures between SSU rRNAs have been extracted for taxonomic and evolutionary studies (Du *et al.*, 2017). Comparison of all SSU rRNA secondary structures when mapped across the three domains of life obtained from Comparative RNA Web (CRW) (Mears *et al.*, 2002) provides a valuable picture of phylogenetic conservation by allowing for the identification of nucleotides or structural regions that are either highly conserved or variable.

Secondary structure studies of SSU rRNAs before the atomic structure of the ribosome were determined to have contributed to the early functional understanding of the ribosome. For example, the highly conserved sequence and secondary structure of 530 loop region of the 16S rRNA, which consists of nucleotide G530, was predicted to form a pseudoknot that was of structural and functional importance in protein translation (Powers and Noller, 1991). When the atomic resolution structure of the 30S ribosome was solved, it was indeed revealed that the 530 loop is interacting with the anticodon stem-loop of tRNA that recognizes the mRNA codon at the ribosomal A site. The nucleotide G530 in the 530 loop together with bases A1492 and A1493 on helix 44 of SSU rRNA are now known to be crucial for correct codon–anticodon interaction during the decoding process (Ogle *et al.*, 2001).

Computer programs such as SSU-DRAW, which is part of the SSU-ALIGN package (Nawrocki, 2009), can generate secondary structures of newly identified SSU rRNAs with a high degree of confidence by using a template-based comparative analysis method. However, comparative analysis with covariation approach is still not able to reveal noncanonical base pairings that are often found in complex RNA structures such as rRNA. More recently available structure prediction approaches can integrate information on specific geometric and molecular interactions obtained from experimentally determined rRNA 3D structures that may thus improve secondary structure prediction (Petrov *et al.*, 2014).

SSU rRNA 3D Structure Studies

While sequences of SSU rRNA are now fairly easy to obtain via DNA sequencing and from there the secondary structure can also be predicted with high confidence, the accurate computational prediction of its 3D structure is still not possible. This is due to the generally long sequences that are in excess of 1000 nucleotides that also have interactions with ribosomal proteins in order to form the mega Dalton RNA–protein complex. Nonetheless, much effort is ongoing towards developing accurate tools for RNA 3D structure prediction. One initiative in this direction is RNA-Puzzles, a collective blind experiment to evaluate RNA prediction techniques (Cruz *et al.*, 2012).

At present, the 3D structure of SSU rRNA can only be acquired by determining the structure of the ribosome. X-ray crystallography and cryoelectron microscopy (cryo-EM) remain the only routes to experimentally obtain the ribosome's structure. The 3D coordinates of the rRNA chains can be extracted from the atomic resolution structures of the ribosome that have been deposited in the Protein Data Bank (PDB). Ribosomal RNA forms a stable ribosomal subunit when in complex with ribosomal proteins; this may explain why no independent structures of rRNA are available.

The 3D structure of rRNA can be analyzed and displayed with available molecular visualization software already being used for protein structures such as COOT (Emsley and Cowtan, 2004), Chimera (Pettersen *et al.*, 2004), PyMOL (Delano, 2002), and several others. Other programs such as Ribovision (Bernier *et al.*, 2014) were exclusively designed to visualize and analyze the primary sequences as well as secondary and three-dimensional structures of rRNA in a single platform. Selected bioinformatics tools that are available for sequence and structure analysis of SSU rRNA are listed in Table 2.

In 2000–2001, the high-resolution crystal structures were obtained for the 30S ribosomal subunits from thermophilic bacteria, *Thermus thermophilus* (Ogle *et al.*, 2001; Wimberly *et al.*, 2000; Schlueder *et al.*, 2000) and the 50S subunit from the halophilic

Table 2 The list of programs and their applications in rRNA sequence and structure analysis

Function	Program/Server (URL)	Method/Content	URL
Sequence alignment	BLASTN	rDNA sequence alignment to rRNA sequence obtained from PDB database	https://blast.ncbi.nlm.nih.gov/Blast.cgi
	Clustal W	Multiple sequence alignment	https://www.ebi.ac.uk/Tools/msa/clustalw2/
	T-Coffee	Iterative alignment of DNA, RNA and protein sequences to get conserved residue/motif	http://tcoffee.crg.cat/apps/tcoffee/do:regular
	SINA	Quality checked and aligned ribosomal RNA sequence data	https://www.arb-silva.de/
	SSU-ALIGN	Structure based multiple sequences alignment of SSU ribosomal RNA sequences (16S & 18S) using covariance models	http://eddylib.org/software/ssu-align/
	Infernal ("INFERence of RNA Alignment")	rRNA alignment with model-based approach using covariance models (CM)	http://eddylib.org/infernal/
	CRWAlign	Template-based alignment	http://www.rna.icmb.utexas.edu/SAE/2F/CRWAlign/index.php
Ribosomal RNA database	MAFFT	Multiple ncRNA alignments	https://mafft.cbrc.jp/alignment/software/
	SILVA	A comprehensive online database for all ribosomal RNA including 16S/18S/23S/28S	https://www.arb-silva.de/
	RDP	Ribosomal Database Project	https://rdp.cme.msu.edu/
Ribosomal RNA prediction	Greengenes	16S rRNA gene database	http://greengenes.lbl.gov
	rmm_rRNA	Identification of rRNA sequences	http://weizhong-lab.ucsd.edu/metagenomic-analysis/server/hmm_rRNA/
Protein structure database	RNAmer	Predict rRNAs from genome sequences	http://www.cbs.dtu.dk/services/RNAmer/
	SSU-ALIGN	Identify, align, mask, and visualize SSU rRNA	http://eddylib.org/software/ssu-align/
	Protein Data Bank (PDB)	A database for the three-dimensional structural data of macromolecules including large complexes like ribosome	https://www.rcsb.org/
3D Structural display and analysis	COOT	Macromolecular model building, model completion and validation	https://www2.mrc-lmb.cam.ac.uk/personal/pemsley/coot/
	PyMOL	Molecular visualization system	https://pymol.org/2/
	Chimera	Structure visualization and analysis	https://www.cgl.ucsf.edu/chimera/
Ribosome information viewer	Ribovision	Open-source ribosome-information viewer website	http://apollo.chemistry.gatech.edu/RiboVision/Documentation/index.html

archaeon, *Haloarcula marismortui* (Ban *et al.*, 2000), which revealed in great depth the details of how the ribosome decodes mRNA and synthesizes the corresponding polypeptide en route to becoming a functional protein molecule. For the first time, the previously predicted secondary and tertiary structure of rRNAs could be compared and validated against experimentally determined examples. As a testament to the accuracy of the prediction tools, about 97%–98% of previously predicted rRNAs' base pairings were indeed found in the ribosomal crystal structures (Gutell *et al.*, 2002).

Advancements in cryo-EM and X-ray crystallography have resulted in several high-resolution ribosomal structures being determined for the ribosomes of eukaryotic species – human, protozoa, and yeast – (Zhang *et al.*, 2016; Khatter *et al.*, 2015; Wong *et al.*, 2014; Ben-Shem *et al.*, 2011; Rabl *et al.*, 2011) as well as the ribosome of organelles – mitochondria (human, pig, and yeast) and chloroplast (spinach) (Desai *et al.*, 2017; Amunts *et al.*, 2015; Greber *et al.*, 2015). Comparisons of SSU rRNA crystal structures for bacteria and eukaryotes have revealed that both domains of life share a similar structure at several of the hypervariable regions and help to provide information on secondary structure mappings that were not previously identified in 18S rRNA (Lee and Gutell, 2012).

Case Studies: Sequence Alignment Guided Structure Analysis of SSU rRNA From Prokaryotes, Eukaryotes, and Organelles

High-Resolution Ribosome Structures

Fourteen high-resolution ($<4\text{\AA}$) structures of whole ribosomes available in the PDB were selected for this case study to demonstrate MSA guided structure analysis of SSU rRNA for two domains of life (prokaryote and eukaryote) and organelles (chloroplast and mitochondrial). The domain archaea was not included due to the lack of atomic structures for their 70S and 30S ribosomes. The selected target ribosomes include **prokaryotic 70S ribosomes**: Gram-negative bacteria *Escherichia coli* (PDB ID: 4YBB), and *Thermus thermophilus* (PDB ID: 4V51), Gram-positive bacteria *Bacillus subtilis* (PDB ID: 3J9W) and *Staphylococcus aureus* (PDB ID: 5LI0); and *Mycobacterium tuberculosis* (PDB ID: 5V93) and *Mycobacterium smegmatis* (PDB ID: 5O61); **Eukaryotic 80S ribosomes**: *Saccharomyces cerevisiae* (PDB ID: 4V88), *Homo sapiens* (human) (PDB ID: 4UG0), *Leishmania donovani* (PDB ID: 5T2A) and *Plasmodium falciparum* (PDB ID: 3J7A); **Mitochondrial ribosomes**: *Saccharomyces cerevisiae* mitoribosome (PDB ID:

5MRC), *Sus scrofa* (Porcine) mitoribosome (PDB ID: 5AJ3) and *Homo sapiens* 55S mitoribosome (PDB ID: 3J9M); **Chloroplastid 70S ribosome:** *Spinacia oleracea* (spinach) chloroplast ribosome (PDB ID: 5MMM).

Multiple Sequence Alignment and Phylogenetic Tree Construction

The SSU rRNA sequences corresponding to the species of selected ribosomes including the SSU rRNA of *E. coli* (16S_E_coli), *T. thermophilus* (16S_T_thermophilus), *B. Subtilis* (16S_B_subtilis), *S. aureus* (16S_S_aureus), *M. smegmatis* (16S_M_smegmatis), *M. tuberculosis* (16S_M_tuberculosis), chloroplast of *S. oleracea* (16S_S_oleracea_Chloroplast), *S. cerevisiae* (18S_S_cerevisiae), *L. donovani*

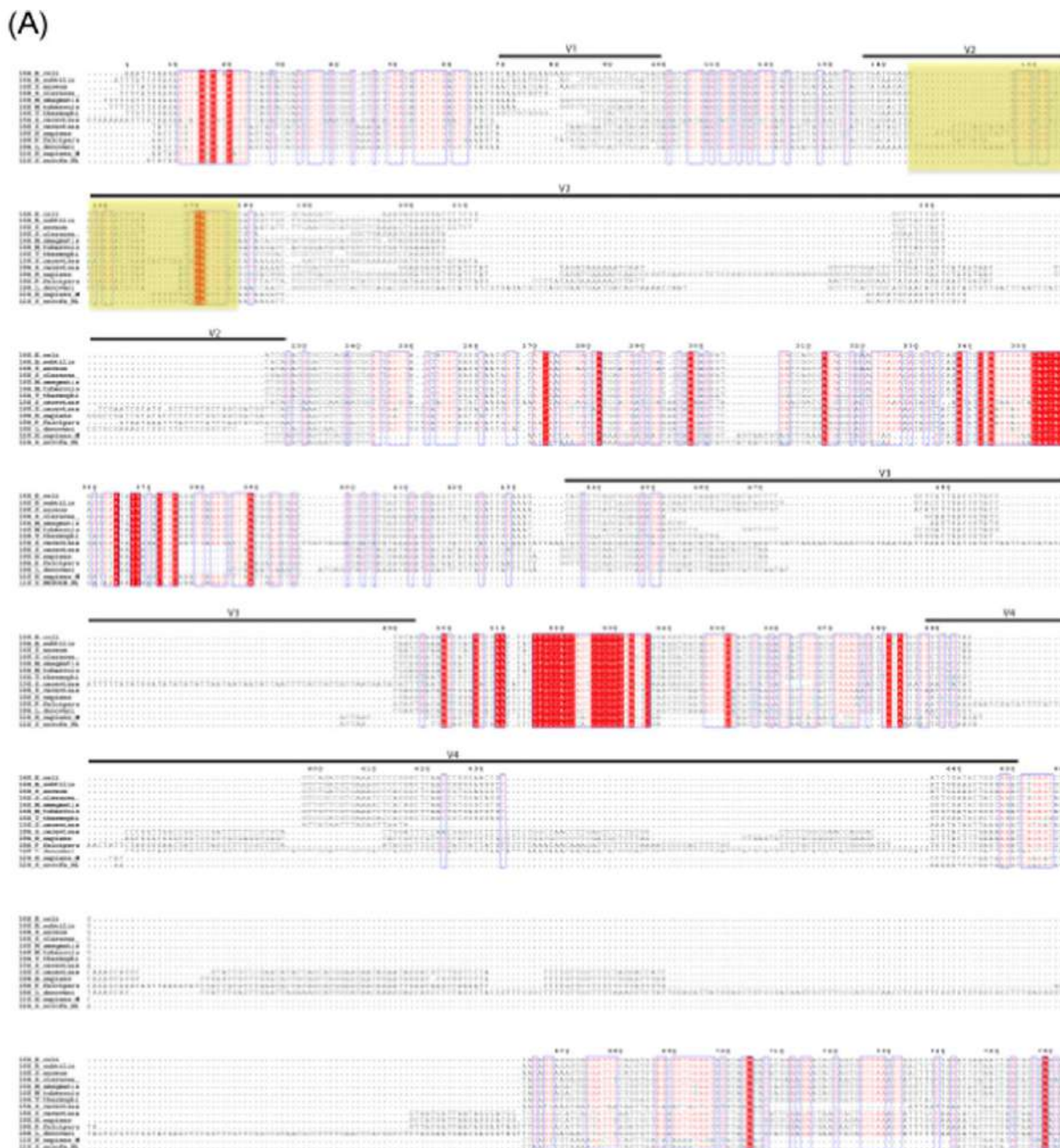


Fig. 1 Multiple sequence alignments for the SSU rRNA of 14 selected species with high-resolution ($< 4 \text{ \AA}$) ribosomal structures that were analyzed using fast Fourier transform and with the results subjected to phylogenetic tree construction using the neighbor-joining method with MAFFT (<https://mafft.cbrc.jp/alignment/software/>). (A) The MSA result in Clustal format as visualized using Esprout program with the highly conserved regions highlighted in red (A). (B) The phylogenetic tree as visualized using Phylo.io. The variable regions (V1–V9) of *Escherichia coli* 16S SSU rRNA are indicated with horizontal black bars. The highlighted variable region of V2 (nucleotides 144–178) indicate the nucleotides that form helix h8 in bacterial 16S SSU rRNA, but form h3 and h4 of 12S mammalian mito-SSU rRNA.

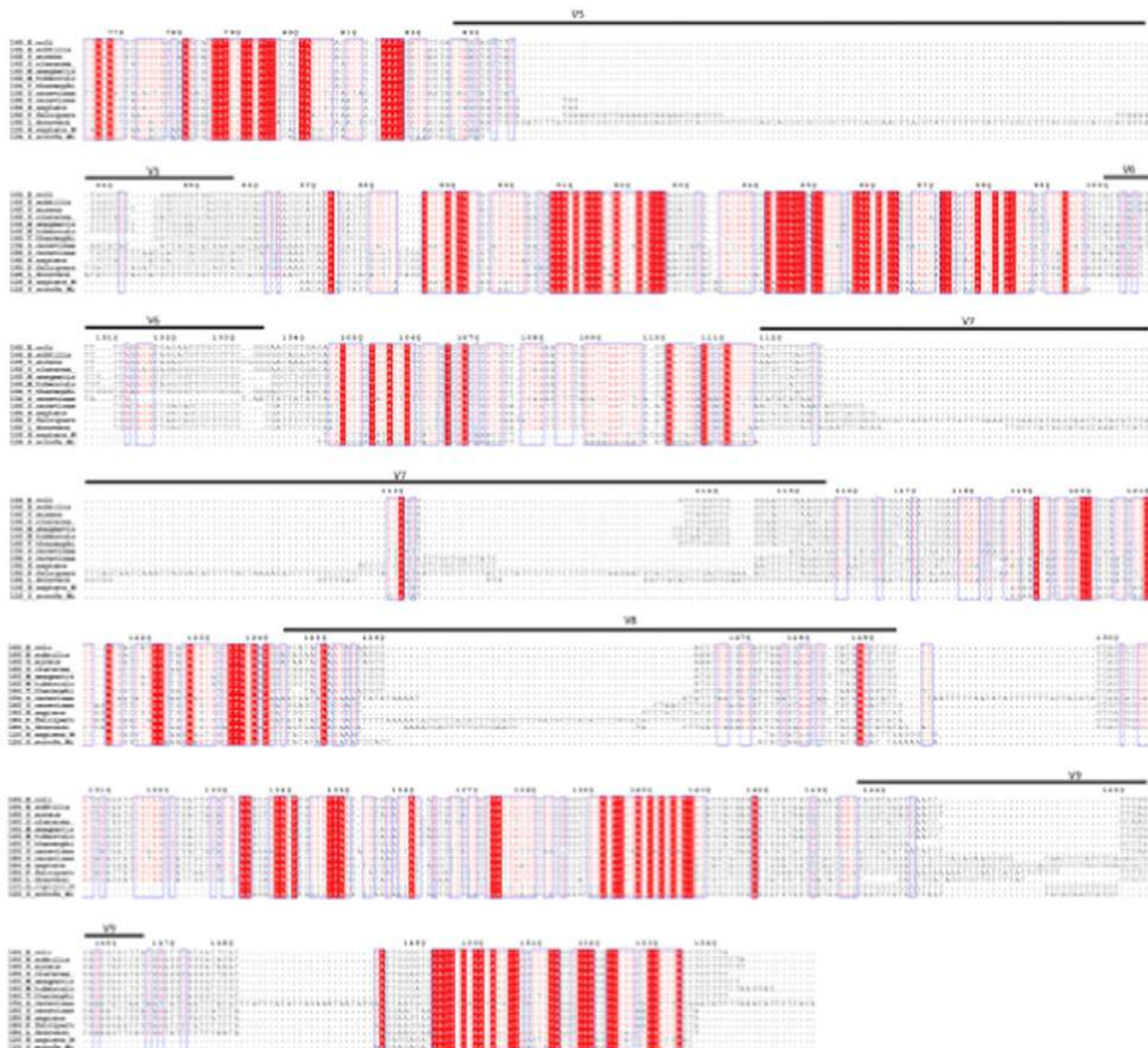


Fig. 1 Continued

(18S_L_donovani), *P. falciparum* (18S_P_falciparum), *H. sapiens* (18S_H_sapiens), mitochondrial ribosome of *S. cerevisiae* (15S_S_cerevisiae_Mitoribosome) and mitochondrial ribosome of *H. sapiens* (12S_H_sapiens_Mitoribosome) were obtained from NCBI. The FASTA formatted sequences were subjected to MSA using MAFFT, which employs fast Fourier transform in sequence alignments with an Auto setting, and then subjected to phylogenetic analysis using the neighbor-joining method (see “Relevant Websites section”). The phylogenetic trees were visualized with Phylo.io (Robinson *et al.*, 2016; Katoh and Standley, 2013; Katoh *et al.*, 2002). The MSA results in Clustal format were visualized with the ESPrnt program (Gouet *et al.*, 1999) to highlight the conserved regions.

Secondary Structure Diagram of 16S SSU RNA

The secondary structure diagrams of SSU rRNAs of human 18S, *E. coli* 16S, human mito ribosome 12S, and *S. cerevisiae* mito ribosome 15S were obtained from the Comparative RNA Web server (see “Relevant Websites section”). Additionally, an overview of SSU rRNA conserved secondary structures across the three domains of life was also obtained from CRW (see “Relevant Websites section”) (Mears *et al.*, 2002).

Comparison of 3D Structure of SSU rRNAs

The selected SSU rRNAs structure coordinates were obtained from the structures deposited in the PDB as listed in Section “SSU rRNA Sequence Studies.” The 16S rRNA of *E. coli* (PDB ID: 4YBB) was used as a reference structure for 3D structure comparisons of human 18S rRNA (PDB ID: 4UG0), 15S rRNA of *S. cerevisiae* mito ribosome (PDB ID: 4V88), and 12S rRNA of



Fig. 1 Continued

human mito ribosome (PDB ID: 3J9M). The structure superposition was carried out using least-squares fitting in the COOT program (Emsley and Cowtan, 2004). The program PyMOL (DeLano, 2002) was used to visualize all the 3D structures that were analyzed in this case study.

Results and Discussion

The sequences of the 14 selected SSU rRNAs from prokaryote, eukaryote, and organelles with nucleotides lengths of 962–2205 were aligned. Despite the significant difference in length, the MSA showed that all the SSU rRNAs were generally well aligned, particularly at the previously identified conserved regions (Baker *et al.*, 2003; Van de Peer *et al.*, 1996) including helices h18, h27, h28, h30, h31, h44, and h45. In comparison to the mammalian cell's 12S mito ribosome, all the aligned 16S and 18S SSU rRNAs were found to have insertions mainly at the variable regions V1–V9 (Fig. 1(A)). Phylogenetic analysis based on the MSA results showed that bacteria, eukaryote, and mito ribosome SSU rRNAs were shown as different branches in the unrooted tree (Fig. 1(B)). The 16S rRNA of Gram-positive, Gram-negative bacteria, and mycobacterium are closely related to each other in one branch (Fig. 1(B)). In addition, chloroplast rRNA from spinach also clustered to prokaryote SSU rRNA despite having evolved to be an organelle that is important for photosynthetic plants and algae. The eukaryotic SSU rRNA for mammals, parasites, and yeasts was also shown to be more closely related to each other. Nonetheless, unlike the nuclear ribosome, the SSU rRNA of the mitochondrial ribosome in yeast is significantly deviated compared to the mammalian mito ribosome. The overall MSA and phylogenetic analysis can clearly distinguish sequences of SSU rRNA to the domain of life and organelle that it belongs to.

While the alignments are able to show that the nucleotide sequences are universally conserved across the SSU rRNA, especially at the conserved regions, such as the helices of h1 and h2 (nucleotide 11–23) and loop 530 region of helix h18 (nucleotide 515–536, all the nucleotide numberings used refer to those in *E. coli* SSU rRNA), there are limitations for the MSA results when it comes to demonstrating the conservation for the secondary structure. For example, the nucleotides 144–178 (V2 region) are known to form helix h8 in 16S rRNA, however similar residues of the mito ribosome were found to form helices h3 and h4 of 12S rRNA (Fig. 1(A) and Fig. 2(A)). The use of sequence comparisons alone is unable to provide a structural context to the RNA molecule. Comparisons of secondary structure are therefore necessary to further identify conservation in the RNA molecules from a structural perspective.

Based on the predicted secondary structures of the 12S, 15S, 16S, and 18S SSU rRNAs that were obtained from the Comparative RNA Web server (CRW) (see "Relevant Websites section") (Cannone *et al.*, 2002) as shown in Fig. 2, all SSU rRNA secondary structure diagrams showed an overall similarity in containing the 3' minor and major domains, 5' major domains, and central domains (Fig. 2). The secondary structure regions that are formed with universally conserved nucleotides identified in the MSA (Fig. 1(A) and Fig. 2(A)) were found to be mostly conserved. Nonetheless, one can clearly identify the structural differences between

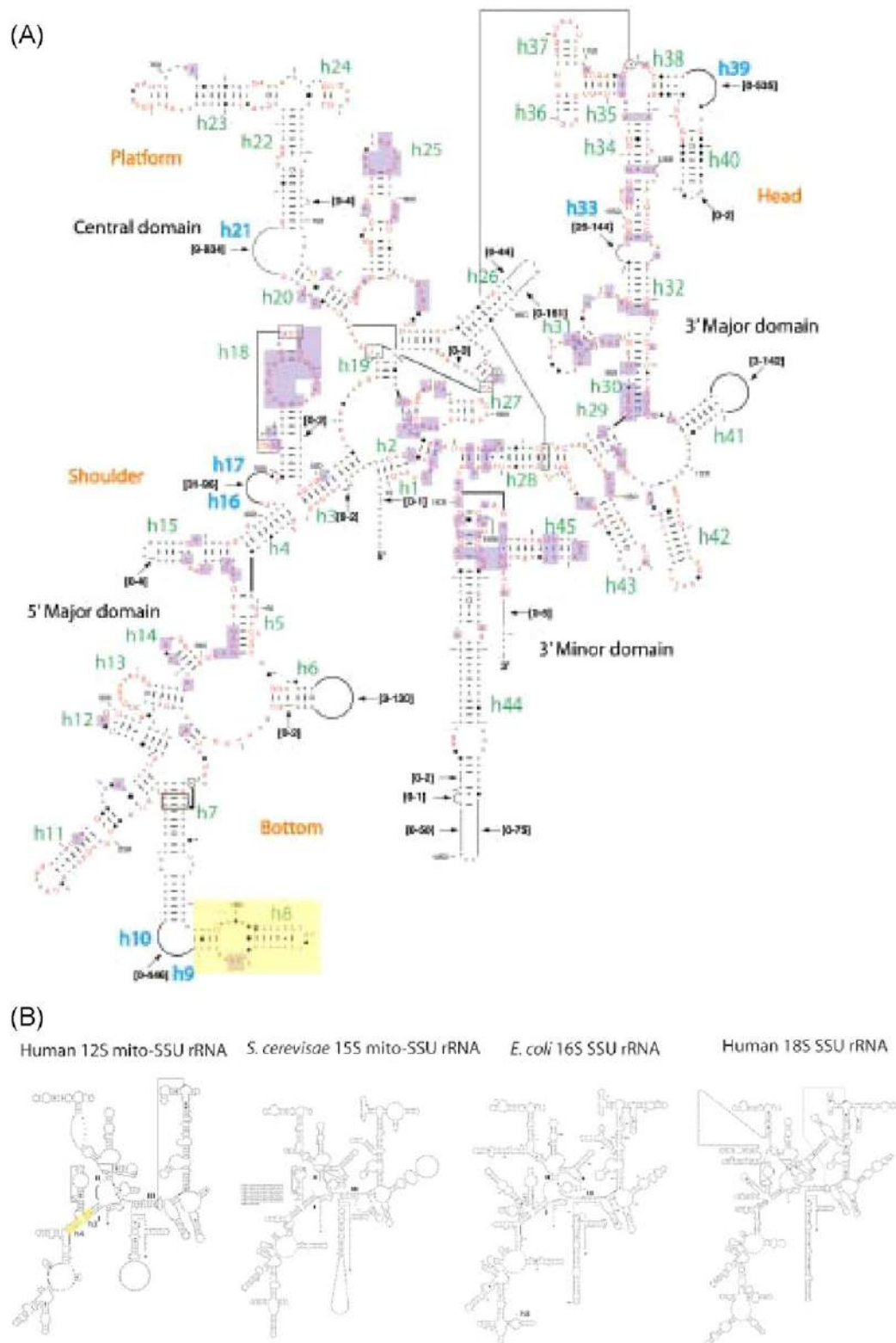


Fig. 2 The secondary structure of SSU rRNA. (A) The secondary structure of *Escherichia coli* 16S SSU rRNA with phylogenetic conservation superimposed across the three domains of life. This diagram was obtained from Comparative RNA Web (CRW) (http://www.rna.icmb.utexas.edu/SIM/4A/Minimal_Ribosome/) with slight modifications. Conserved nucleotide sequences are shown as ACGU: 98 + % conserve, acgu: 90%–98% conserved, •: 80%–90% conserved, and •: less than 80% conserved. Arcs represent the variable regions with the nucleotide number (lower and upper) known to exist. The nucleotides that were 100% conserved in the MSA shown in Fig. 1 are highlighted in purple. The helices that are conserved in the three domains of life were labeled in green, while the helices in the variable regions are in blue. Helix h8 is highlighted in yellow. (B) The secondary structures of *Escherichia coli* 16S SSU rRNA, human 18S SSU rRNA, 15S yeast mito-SSU rRNA, and 12S human mito-SSU rRNA were obtained from CRW (http://www.rna.icmb.utexas.edu/SIM/4A/Minimal_Ribosome/) for comparison. Helices h3 and h4 of 12S human mito-SSU rRNA are highlighted in yellow.

the compared SSU rRNAs. For example, while the head region is highly similar, the bottom region was significantly deviated (Fig. 2(B)). Such structural deviations will provide a better understanding of the evolutionary events that specific regions of the RNA molecule had undergone. Although the secondary structure is informative in revealing the general structure conservation of the RNA molecule that is missing in sequence alignments, it remains difficult to predict noncanonical base pairings and highly idiosyncratic RNA sequences – For example, a GC pairing can be assumed to conform to the Watson–Crick geometry but in reality, that GC pairing may actually be of the noncanonical reverse Watson–Crick arrangement. Recently, high-resolution structures have been used to aid in constructing a more complete and accurate secondary structure map of SSU rRNA (Petrov *et al.*, 2014).

The overall shape of SSU rRNAs in this case study, with the head, neck, beak, platform, body, shoulder, and spur features can be clearly discerned using molecular visualization (Fig. 3). The 12S mammalian mt-SSU rRNA was shown to have a contracted form in accordance to its shortest sequence length, while 15S, 16S, and 18S SSU rRNA have similar overall dimension (Fig. 3(A)). The contracted mammalian mt-SSU rRNA has been known to recruit more proteins in order to perform the specific synthesis of hydrophobic integral membrane proteins (Amunts *et al.*, 2015; O'Brien, 2003). The yeast 15S mt-SSU rRNA had not undergone similar evolutionary events compared to its mammalian counterpart and thus maintains a size similar to bacterial 16S rRNA. The expansion segments of 18S human SSU rRNA compared to 16S SSU rRNA can also be identified (Fig. 3(B)). For detailed in depth discussion of the features for each SSU rRNA structure, please refer to the respective original publications. Overall, the 3D structures provide details that reveal universally conserved regions as well as structural features that show how rRNA have evolved specifically for a particular domain of life and organelles.

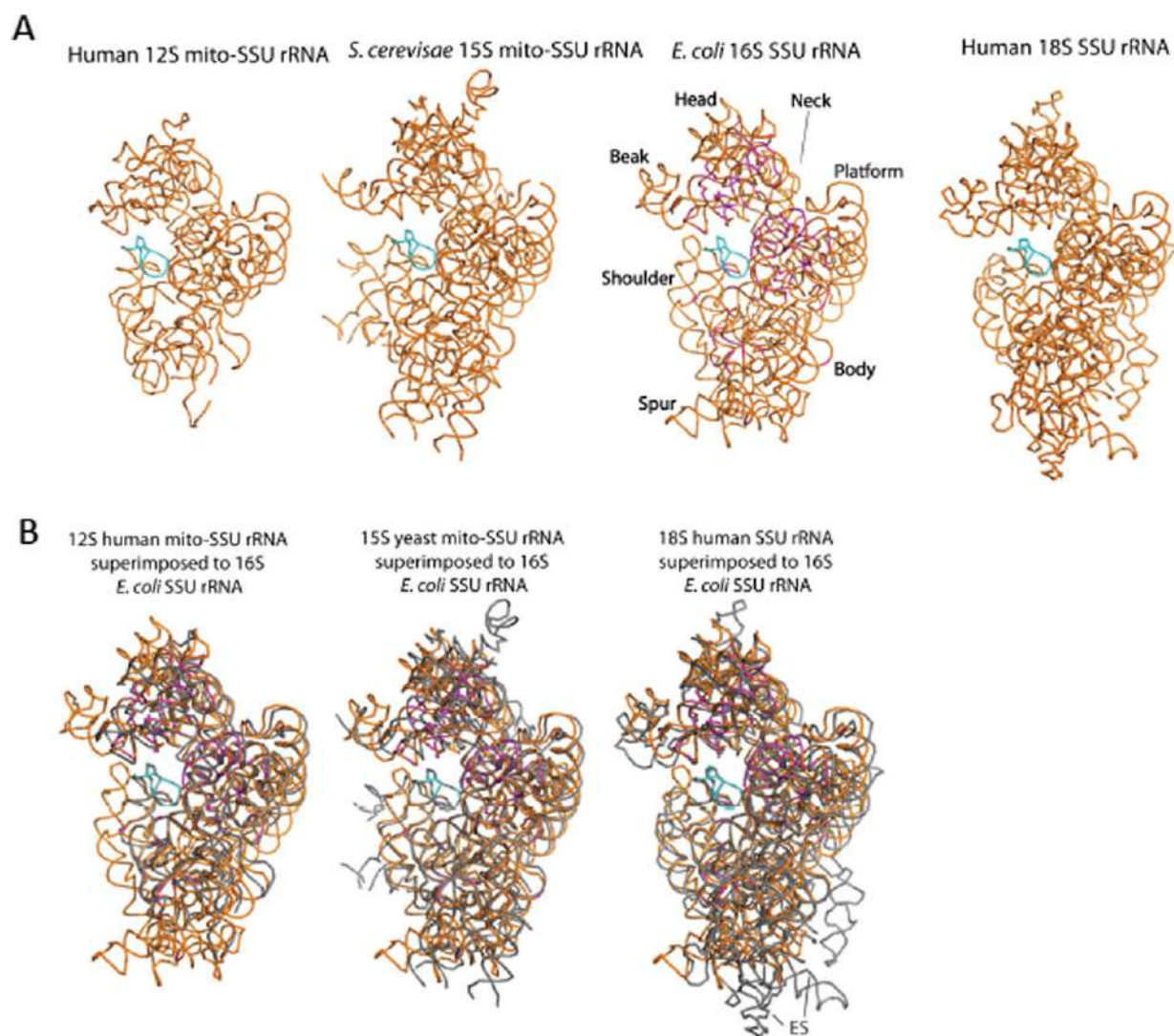


Fig. 3 The 3D structure of SSU rRNAs. (A) The 3D structure of SSU rRNAs were shown in same dimension by superimposed to the loop 530 nucleotides (cyan) of *Escherichia coli* 16S SSU rRNA. The nucleotides found to be 100% conserved in multiple sequence alignment of Fig. 1(A) are shown as pink in *E. coli* 16S SSU rRNA. The figure is generated by using the program PyMOL. (B) Comparison of SSU rRNA by superimposing of 12S human mito-SSU rRNA, 15S yeast mito-SSU rRNA, and 18S human SSU rRNA (gray) to 16S *E. coli* SSU rRNA (orange). ES indicates part of the expansion segments of 18S human SSU rRNA.

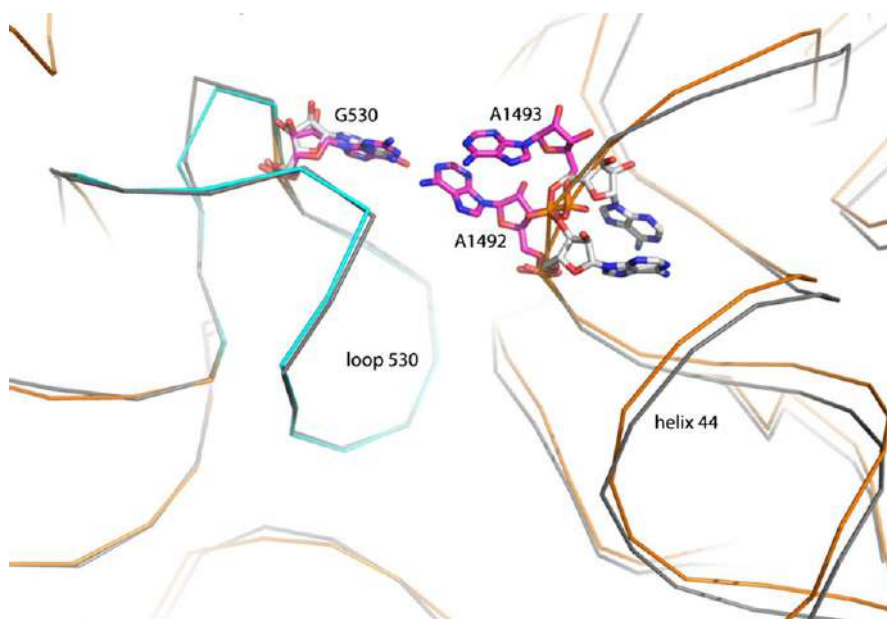


Fig. 4 The induced-fit conformational change that was employed for the universally conserved nucleotides upon cognate tRNA binding demonstrated that the RNA molecule is dynamic in carrying out its biological function. Both A1492 and A1493 were shown to flip out from helix 44 for *Thermus thermophilus* 16S subunit (pink). The same nucleotides in the yeast mito-SSU rRNA (gray) remained in the helix.

In **Fig. 4**, nucleotides A1492 and A1493 are shown to flip out from helix 44 of *T. thermophilus* 16S (PDB ID: 4V51), while the same nucleotides for the yeast mito-SSU rRNA remained in the helix. These nucleotides are known to be universally conserved and important for cognate tRNA recognition at the A site (Selmer *et al.*, 2006; Ogle *et al.*, 2001; Wimberly *et al.*, 2000). Here, the nucleotides in *T. thermophilus* 16S were shown at the stage of where the ribosome is bound with cognate tRNA at the A-site of 16S SSU rRNA, while 15S mito-SSU rRNA of yeast is in the empty form (without cognate tRNA at the A-site). Despite the universally conserved sequences, as well as overall agreement of the secondary and overall tertiary structures for loop 530 and helix 44, the use of an induced-fit conformational change on the localized nucleotides upon the cognate tRNA binding (Ogle *et al.*, 2001) demonstrating that the RNA molecule is dynamic in anticipating its biological functions, hence the need to increase the complexity of the challenge to computationally predict accurate tertiary structures.

Future Directions and Closing Remarks

Currently available computational tools can be traced back to work that was first reported in the 1960s. Over the years, these programs have expanded their utility and accuracy. They are now well established in their capacity to identify well-conserved nucleotides, even in long RNA sequences with large amounts of variation. The limitation of primary structure analysis in being unable to identify structurally conserved regions with more diverse sequence variations can be complemented by secondary structure prediction methods that integrate covariation and free energy minimization approaches.

Despite improvements in secondary structure prediction methods, they are still not able to definitively discern the specific molecular arrangements for base pairings and are thus not able to differentiate canonical versus noncanonical pairs. Nevertheless, such tools remain highly functional and can provide a useful map and guidance when navigating the sequences or structures of newly acquired data for long and complex RNA molecules such as the ribosomal subunits.

RNA 3D structure prediction remains a challenge and is generally limited to chains with less than 500 nucleotides. Accurate 3D structure prediction of RNA that can correctly differentiate canonical and noncanonical pairs, as well as identify interaction sites to metal ions, organic molecules, and other macromolecules will have great utility in using the vast amount of sequence data available. Such programs can in the future trace minor evolutionary changes that are relevant from not only a sequence perspective but also with relevant structural context. They can also perhaps be applicable for the identification of RNA molecules as targets for drug development work. Initial computational screening of such targets can direct investigators towards specific RNA structures that need to be acquired for in depth experimental structural analysis.

Acknowledgment

MFR is funded by the Ministry of Science, Technology and Innovation Malaysia grant 02-01-02-SF1278 and Universiti Kebangsaan Malaysia grant GP-K011849.

CLN is funded by the Universiti Kebangsaan Malaysia grant GP-K019584 and GUP-2017-070.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Characterizing and Functional Assignment of Noncoding RNAs. Engineering of Supramolecular RNA Structures. Natural Language Processing Approaches in Bioinformatics. Predicting RNA-RNA Interactions in Three-Dimensional Structures. Prediction of Coding and Non-Coding RNA

References

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215, 403–410.
- Amunts, A., Brown, A., Bai, X.C., *et al.*, 2014. Structure of the yeast mitochondrial large ribosomal subunit. *Science* 343 (6178), 1485–1489.
- Amunts, A., Brown, A., Toots, J., Scheres, S.H., Ramakrishnan, V., 2015. The structure of the human mitochondrial ribosome. *Science* 348 (6230), 95–98.
- Baker, G.C., Smith, J.J., Cowan, D.A., 2003. Review and re-analysis of domain-specific 16S primers. *Journal of Microbiological Methods* 55 (3), 541–555.
- Ban, N., Nissen, P., Hansen, J., Moore, P.B., Steitz, T.A., 2000. The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science* 289 (5481), 905–920.
- Batey, R.T., Rambo, R.P., Doudna, J.A., 1999. Tertiary motifs in RNA structure and folding. *Angewandte Chemie International Edition* 38 (16), 2326–2343.
- Bellaousov, S., Reuter, J.S., Seetin, M.G., Mathews, D.H., 2013. RNAstructure: Web servers for RNA secondary structure prediction and analysis. *Nucleic Acids Research* 41 (W1), W471–W474.
- Ben-Shem, A., de Loubresse, N.G., Melnikov, S., *et al.*, 2011. The structure of the eukaryotic ribosome at 3.0 Å resolution. *Science* 334 (6062), 1524–1529.
- Bernier, C.R., Petrov, A.S., Waterbury, C.C., *et al.*, 2014. RiboVision suite for visualization and analysis of ribosomes. *Faraday Discussions* 169, 195–207.
- Brosius, J., Palmer, M.L., Kennedy, P.J., Noller, H.F., 1978. Complete nucleotide sequence of a 16S ribosomal RNA gene from *Escherichia coli*. *Proceedings of the National Academy of Sciences* 75 (10), 4801–4805.
- Butcher, S.E., Pyle, A.M., 2011. The molecular interactions that stabilize RNA tertiary structure: RNA motifs, patterns, and networks. *Accounts of Chemical Research* 44 (12), 1302–1311.
- Cannone, J.J., Subramanian, S., Schnare, M.N., *et al.*, 2002. The comparative RNA web (CRW) site: An online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics* 3 (1), 2.
- Carter, A.P., Clemons, W.M., Brodersen, D.E., *et al.*, 2000. Functional insights from the structure of the 30S ribosomal subunit and its interactions with antibiotics. *Nature* 407 (6802), 340–348.
- Cate, J.H., Gooding, A.R., Podell, E., *et al.*, 1996. Crystal structure of a group I ribozyme domain: Principles of RNA packing. *Science* 273 (5282), 1678–1685.
- Cech, T.R., 2012. The RNA worlds in context. *Cold Spring Harbor Perspectives in Biology* 4 (7), a006742.
- Clemons, W.M., May, J.L., Wimberly, B.T., *et al.*, 1999. Structure of a bacterial 30S ribosomal subunit at 5.5 Å resolution. *Nature* 400 (6747), 833–840.
- Cole, J.R., Wang, Q., Fish, J.A., *et al.*, 2013. Ribosomal Database Project: Data and tools for high throughput rRNA analysis. *Nucleic Acids Research* 42 (D1), D633–D642.
- Crick, F.H.C., 1958. On protein synthesis. *Symposia of the Society for Experimental Biology* 12, 138–163.
- Cruz, J.A., Blanchet, M.F., Boniecki, M., *et al.*, 2012. RNA-Puzzles: A CASP-like evaluation of RNA three-dimensional structure prediction. *RNA* 18 (4), 610–625.
- Das, R., Baker, D., 2007. Automated de novo prediction of native-like RNA tertiary structures. *Proceedings of the National Academy of Sciences of the United States of America* 104 (37), 14664–14669.
- Das, R., Karanicolas, J., Baker, D., 2010. Atomic accuracy in predicting and designing noncanonical RNA structure. *Nature Methods* 7 (4), 291–294.
- DeLano, W.L., 2002. The PyMOL molecular graphics system. Available at: <http://pymol.org>.
- Desai, N., Brown, A., Amunts, A., Ramakrishnan, V., 2017. The structure of the yeast mitochondrial ribosome. *Science* 355 (6324), 528–531.
- Ding, F., Sharma, S., Chalasani, P., *et al.*, 2008. Ab initio RNA folding by discrete molecular dynamics: From structure prediction to folding mechanisms. *RNA* 14 (6), 1164–1173.
- Doshi, K.J., Cannone, J.J., Cobaugh, C.W., Gutell, R.R., 2004. Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics* 5 (1), 105.
- Du, Y.H., Zhao, Y.J., Tang, F.H., 2017. A new molecular approach based on the secondary structure of ribosomal RNA for phylogenetic analysis of mobilid ciliates. *Current Microbiology*. 1–9.
- Eddy, S.R., Durbin, R., 1994. RNA sequence analysis using covariance models. *Nucleic Acids Research* 22 (11), 2079–2088.
- Emsley, P., Cowtan, K., 2004. Coot: Model-building tools for molecular graphics. *Acta Crystallographica Section D: Biological Crystallography* 60 (12), 2126–2132.
- Federhen, S., 2011. The NCBI taxonomy database. *Nucleic Acids Research* 40 (D1), D136–D143.
- Ferré-D'Amaré, A.R., Doudna, J.A., 1999. RNA folds: Insights from recent crystal structures. *Annual Review of Biophysics and Biomolecular Structure* 28 (1), 57–73.
- Ferré-D'Amaré, A.R., Zhou, K., Doudna, J.A., 1998. Crystal structure of a hepatitis delta virus ribozyme. *Nature* 395 (6702), 567–574.
- Fox, G.E., Woese, C.R., 1975. 5S RNA secondary structure. *Nature* 256 (5517), 505–507.
- Gilbert, W., 1986. Origin of life: The RNA world. *Nature* 319 (6055), 30–33.
- Gouet, P., Courcelle, E., Stuart, D.I., Métot, F., 1999. ESPript: Analysis of multiple sequence alignments in PostScript. *Bioinformatics (Oxford)* 15 (4), 305–308.
- Greber, B.J., Bieri, P., Leibundgut, M., *et al.*, 2015. The complete structure of the 55S mammalian mitochondrial ribosome. *Science* 348 (6232), 303–308.
- Guerrier-Takada, C., Gardiner, K., Marsh, T., Pace, N., Altman, S., 1983. The RNA moiety of ribonuclease P is the catalytic subunit of the enzyme. *Cell* 35 (3), 849–857.
- Gutell, R.R., 1996. Comparative sequence analysis and the structure of 16S and 23S rRNA. *Ribosomal RNA: Structure, Evolution, Processing, and Function in Protein Biosynthesis*. 111–128.
- Gutell, R.R., Larsen, N., Woese, C.R., 1994. Lessons from an evolving rRNA: 16S and 23S rRNA structures from a comparative perspective. *Microbiological Reviews* 58 (1), 10–26.
- Gutell, R.R., Lee, J.C., Cannone, J.J., 2002. The accuracy of ribosomal RNA comparative structure models. *Current Opinion in Structural Biology* 12 (3), 301–310.
- Gutell, R.R., Noller, H.F., Woese, C.R., 1986. Higher order structure in ribosomal RNA. *The EMBO Journal* 5 (5), 1111.
- Gutell, R.R., Weiser, B., Woese, C.R., Noller, H.F., 1985. Comparative anatomy of 16S-like ribosomal RNA. *Progress in Nucleic Acid Research and Molecular Biology* 32, 155–216.
- Hadziavdic, K., Lekang, K., Lanzen, A., *et al.*, 2014. Characterization of the 18S rRNA gene for designing universal eukaryote specific primers. *PLOS ONE* 9 (2), e87624.
- Hall, M.N., Gabay, J., Débarbouillé, M., Schwartz, M., 1982. A role for mRNA secondary structure in the control of translation initiation. *Nature* 295 (5850), 616–618.
- Hofacker, I.L., 2003. Vienna RNA secondary structure server. *Nucleic Acids Research* 31 (13), 3429–3431.
- Holley, R.W., Apgar, J., Everett, G.A., *et al.*, 1965. Structure of a ribonucleic acid. *Science*. 1462–1465.
- Hooper, J.K., 2012. Chloroplasts. *Springer Science & Business Media*. pp. 164–177.
- Huang, Y., Gilna, P., Li, W., 2009. Identification of ribosomal RNA genes in metagenomic fragments. *Bioinformatics* 25 (10), 1338–1340.
- Katoh, K., Standley, D.M., 2013. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution* 30 (4), 772–780.
- Katoh, K., Misawa, K., Kuma, K.I., Miyata, T., 2002. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research* 30 (14), 3059–3066.

- Khatte, H., Myasnikov, A.G., Natchiar, S.K., Klaholz, B.P., 2015. Structure of the human 80S ribosome. *Nature* 520 (7549), 640–645.
- Kim, S.H., Quigley, G.J., Suddath, F.L., *et al.*, 1973. Three-dimensional structure of yeast phenylalanine transfer RNA: Folding of the polynucleotide chain. *Science* 179 (4070), 285–288.
- Kim, S.H., Suddath, F.L., Quigley, G.J., *et al.*, 1974. Three-dimensional tertiary structure of yeast phenylalanine transfer RNA. *Science* 185 (4149), 435–440.
- Klosterman, P.S., Tamura, M., Holbrook, S.R., Brenner, S.E., 2002. SCOR: A structural classification of RNA database. *Nucleic Acids Research* 30 (1), 392–394.
- Kruger, K., Grabowski, P.J., Zaug, A.J., *et al.*, 1982. Self-splicing RNA: Autoexcision and autocyclization of the ribosomal RNA intervening sequence of *Tetrahymena*. *Cell* 31 (1), 147–157.
- Kudla, G., Murray, A.W., Tollervey, D., Plotkin, J.B., 2009. Coding-sequence determinants of gene expression in *Escherichia coli*. *Science* 324 (5924), 255–258.
- Lagesen, K., Hallin, P., Rødland, E.A., *et al.*, 2007. RNAmmer: Consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Research* 35 (9), 3100–3108.
- Larkin, M.A., Blackshields, G., Brown, N.P., *et al.*, 2007. Clustal W and Clustal X version 2.0. *Bioinformatics* 23 (21), 2947–2948.
- Lee, J.C., Gutell, R.R., 2012. A comparison of the crystal structures of eukaryotic and bacterial SSU ribosomal RNAs reveals common structural features in the hypervariable regions. *PLOS ONE* 7 (5), e38203.
- Lu, Z.J., Gloor, J.W., Mathews, D.H., 2009. Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA* 15 (10), 1805–1813.
- Lück, R., Steger, G., Riesner, D., 1996. Thermodynamic prediction of conserved secondary structure: Application to the RRE element of HIV, the tRNA-like element of CMV and the mRNA of prion protein. *Journal of Molecular Biology* 258 (5), 813–826.
- Madison, J.T., Everett, G.A., Kung, H., 1966. Nucleotide sequence of a yeast tyrosine transfer RNA. *Science* 153 (3735), 531–534.
- Mathews, D.H., Banerjee, A.R., Luan, D.D., Eickbush, T.H., Turner, D.H., 1997. Secondary structure model of the RNA recognized by the reverse transcriptase from the R2 retrotransposable element. *RNA* 3 (1), 1–16.
- Mathews, D.H., Disney, M.D., Childs, J.L., *et al.*, 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 101 (19), 7287–7292.
- Mathews, D.H., Sabina, J., Zuker, M., Turner, D.H., 1999. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology* 288 (5), 911–940.
- McDonald, D., Price, M.N., Goodrich, J., *et al.*, 2012. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *The ISME Journal* 6 (3), 610–618.
- Mears, J.A., Cannone, J.J., Stagg, S.M., *et al.*, 2002. Modeling a minimal ribosome based on comparative sequence analysis. *Journal of Molecular Biology* 321 (2), 215–234.
- Nawrocki, E.P., 2009. Structural RNA Homology Search and Alignment Using Covariance Models. Washington University in St. Louis.
- Nawrocki, E.P., Eddy, S.R., 2013. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* 29 (22), 2933–2935.
- Neefs, J.M., Van de Peer, Y., De Rijk, P., Chapelle, S., De Wachter, R., 1993. Compilation of small ribosomal subunit RNA structures. *Nucleic Acids Research* 21 (13), 3025–3049.
- Nguyen, T.H.D., Gajek, W.P., Bai, X.C., *et al.*, 2016. Cryo-EM structure of the yeast U4/U6. U5 tri-snRNP at 3.7 Å resolution. *Nature* 530 (7590), 298–302.
- Noller, H.F., Hoffarth, V., Zimmak, L., 1992. Unusual resistance of peptidyl transferase to protein extraction. *Science* 265, 3587–3590.
- Noller, H.F., Kop, J., Wheaton, V., *et al.*, 1981. Secondary structure model for 23S ribosomal RNA. *Nucleic Acids Research* 9 (22), 6167–6189.
- Noller, H.F., Woese, C.R., 1981. Secondary structure of 16S ribosomal RNA. *Science* 212 (4493), 403–411.
- Notredame, C., Higgins, D.G., Heringa, J., 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302 (1), 205–217.
- O'Brien, T.W., 2003. Properties of human mitochondrial ribosomes. *IUBMB Life* 55 (9), 505–513.
- Ogle, J.M., Brodersen, D.E., Clemons, W.M., *et al.*, 2001. Recognition of cognate transfer RNA by the 30S ribosomal subunit. *Science* 292 (5518), 897–902.
- Olinares, P.D.B., Ponnala, L., van Wijk, K.J., 2010. Megadalton complexes in the chloroplast stroma of *Arabidopsis thaliana* characterized by size exclusion chromatography, mass spectrometry, and hierarchical clustering. *Molecular & Cellular Proteomics* 9 (7), 1594–1615.
- Parisien, M., Major, F., 2008. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature* 452 (7183), 51–55.
- Petrov, A.S., Bernier, C.R., Gulen, B., *et al.*, 2014. Secondary structures of rRNAs from all three domains of life. *PLOS ONE* 9 (2), e88222.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., *et al.*, 2004. UCSF Chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry* 25 (13), 1605–1612.
- Poehlsgaard, J., Douthwaite, S., 2005. The bacterial ribosome as a target for antibiotics. *Nature Reviews Microbiology* 3 (11), 870–881.
- Powers, T., Noller, H.F., 1991. A functional pseudoknot in 16S ribosomal RNA. *The EMBO Journal* 10 (8), 2203.
- Pruess, E., Peplies, J., Glöckner, F.O., 2012. SINA: Accurate high-throughput multiple sequence alignment of ribosomal RNA genes. *Bioinformatics* 28 (14), 1823–1829.
- Quast, C., Pruesse, E., Yilmaz, P., *et al.*, 2012. The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Research* 41 (D1), D590–D596.
- Rabl, J., Leibundgut, M., Ataide, S.F., Haag, A., Ban, N., 2011. Crystal structure of the eukaryotic 40S ribosomal subunit in complex with initiation factor 1. *Science* 331 (6018), 730–736.
- RajBhandary, U.L., Chang, S.H., Stuart, A., *et al.*, 1967. Studies on polynucleotides, lxviii* the primary structure of yeast phenylalanine transfer RNA. *Proceedings of the National Academy of Sciences of the United States of America* 57 (3), 751–758.
- Reuter, J.S., Mathews, D.H., 2010. RNAstructure: Software for RNA secondary structure prediction and analysis. *BMC Bioinformatics* 11 (1), 129.
- Robertson, M.P., Joyce, G.F., 2012. The origins of the RNA world. *Cold Spring Harbor Perspectives in Biology* 4 (5), a003608.
- Robertus, J.D., Ladner, J.E., Finch, J.T., *et al.*, 1974. Structure of yeast phenylalanine tRNA at 3 Å resolution. *Nature* 250, 546–551.
- Robinson, O., Dylus, D., Dessimoz, C., 2016. Phylo. io: Interactive viewing and comparison of large phylogenetic trees on the web. *Molecular Biology and Evolution* 33 (8), 2163–2166.
- Russell, J., Zomerdijs, J.C., 2006. The RNA polymerase I transcription machinery. *Biochemical Society Symposia* 73, 203–216.
- Schluenzen, F., Tocilj, A., Zarivach, R., *et al.*, 2000. Structure of functionally activated small ribosomal subunit at 3.3 Å resolution. *Cell* 102 (5), 615–623.
- Scott, W.G., Finch, J.T., Klug, A., 1995. The crystal structure of an AII-RNA hammerhead ribozyme: A proposed mechanism for RNA catalytic cleavage. *Cell* 81 (7), 991–1002.
- Selmer, M., Dunham, C.M., Murphy, F.V., *et al.*, 2006. Structure of the 70S ribosome complexed with mRNA and tRNA. *Science* 313 (5795), 1935–1942.
- Sharma, M.R., Koc, E.C., Datta, P.P., *et al.*, 2003. Structure of the mammalian mitochondrial ribosome reveals an expanded functional role for its component proteins. *Cell* 115 (1), 97–108.
- Sharma, S., Ding, F., Dokholyan, N.V., 2008. iFoldRNA: Three-dimensional RNA structure prediction and folding. *Bioinformatics* 24 (17), 1951–1952.
- Smit, S., Widmann, J., Knight, R., 2007. Evolutionary rates vary among rRNA structural elements. *Nucleic Acids Research* 35 (10), 3339–3354.
- Thompson, J.D., Higgins, D.G., Gibson, T.J., 1994. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* 22 (22), 4673–4680.
- Van de Peer, Y., Chapelle, S., De Wachter, R., 1996. A quantitative map of nucleotide substitution rates in bacterial rRNA. *Nucleic Acids Research* 24 (17), 3381–3391.
- Wimberly, B.T., Brodersen, D.E., Clemons, W.M., *et al.*, 2000. Structure of the 30S ribosomal subunit. *Nature* 407 (6802), 327–339.
- Woese, C.R., 1987. Bacterial evolution. *Microbiological Reviews* 51 (2), 221.
- Woese, C.R., Gutell, R., Gupta, R., Noller, H.F., 1983. Detailed analysis of the higher-order structure of 16S-like ribosomal ribonucleic acids. *Microbiological Reviews* 47 (4), 621–629.
- Woese, C.R., Kandler, O., Wheelis, M.L., 1990. Towards a natural system of organisms: Proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences of the United States of America* 87 (12), 4576–4579.

- Woese, C.R., Magrum, L.J., Gupta, R., *et al.*, 1980. Secondary structure model for bacterial 16S ribosomal RNA: Phylogenetic, enzymatic and chemical evidence. *Nucleic Acids Research* 8 (10), 2275–2294.
- Wong, W., Bai, X.C., Brown, A., *et al.*, 2014. Cryo-EM structure of the *Plasmodium falciparum* 80S ribosome bound to the anti-protozoan drug emetine. *eLife* 3, e03080.
- Wuyts, J., Van de Peer, Y., De Wachter, R., 2001. Distribution of substitution rates and location of insertion sites in the tertiary structure of ribosomal RNA. *Nucleic Acids Research* 29 (24), 5017–5028.
- Yan, C., Hang, J., Wan, R., *et al.*, 2015. Structure of a yeast spliceosome at 3.6-angstrom resolution. *Science* 349 (6253), 1182–1191.
- Yoon, S.H., Ha, S.M., Kwon, S., *et al.*, 2017. Introducing EzBioCloud: A taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies. *International Journal of Systematic and Evolutionary Microbiology* 67 (5), 1613–1617.
- Zhang, X., Lai, M., Chang, W., *et al.*, 2016. Structures and stabilization of kinetoplastid-specific split rRNAs revealed by comparing leishmanial and human ribosomes. *Nature Communications* 7, 13223.
- Zuker, M., 1989. On finding all suboptimal foldings of an RNA molecule. *Science* 244, 48–52.
- Zuker, M., Stiegler, P., 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research* 9 (1), 133–148.
- Zwieb, C., Glotz, C., Brimacombe, R., 1981. Secondary structure comparisons between small subunit ribosomal RNA molecules from six different species. *Nucleic Acids Research* 9 (15), 3621–3640.

Relevant Websites

- <http://www.rna.cccb.utexas.edu/CAR/1D/>
1D. Methods: Structure Prediction with Comparative Sequence Analysis.
- <https://www.arb-silva.de>
Arb SILVA.
- <https://www.ezbiocloud.net>
EzBioCloud.net.
- <https://mafft.cbrc.jp/alignment/software/>
MAFFT – A multiple sequence alignment program – CBRC.
- http://www.rna.icmb.utexas.edu/SIM/4A/Minimal_Ribosome/
Modeling a Minimal Ribosome.
- <https://rna.urmc.rochester.edu/RNAstructureWeb/>
RNAstructure Webserver.
- <https://omictools.com/rna-structures-category>
RNA analysis – OMICtools.
- <http://www.rna.icmb.utexas.edu/DAT/3C/Structure/index.php>
Secondary Structure Diagram Retrieval – The Comparative RNA Web.
- <http://rna.tbi.univie.ac.at/>
ViennaRNA Web Services.

Computational Design and Experimental Implementation of Synthetic Riboswitches and Riboregulators

Munyati Othman and Siuk M Ng, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Bangi, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Riboregulators consist of two basic components; an aptamer domain and an actuator (effector) domain. The aptamer domains are short single stranded DNA or RNA molecules made up of 20–100 nucleotides. These oligonucleotides adopt a variety of unique 3-dimensional (3D) structures that enable binding to specific ligands with high selectivity and specificity (Dunn *et al.*, 2017). The actuator domains are located downstream to the aptamer domain will undergo a conformational change upon ligand binding and is the effector that controls regulation of gene expression (Chang *et al.*, 2012). The modularity of the actuator domain allows it to effect control of gene expression through a variety of mechanisms including transcription, splicing, stability, RNA interference, and translation (Chang *et al.*, 2012; Liang *et al.*, 2011). The mixing and matching of RNA aptamers and actuator domains leads towards a functional synthetic riboregulator construct. Due to the sequence diversity and their high affinity for specific targets, aptamers are often used in riboregulator design.

Several strategies have been used to generate aptamers as candidates for riboregulators. One obvious strategy is to select from naturally occurring aptamers that are found in riboswitches. Naturally occurring aptamers such as the thiamine pyrophosphate (TPP) aptamer domain, have previously been reengineered to control the expression of green fluorescent protein (GFP) (Nomura and Yokobayashi, 2007). Aptamers can also be designed via computational approaches; for example, an aptamer that is able to interact with cytochrome P450 had been designed using molecular modelling methods (Shcherbinin *et al.*, 2015). Additionally, aptamers can also be derived from *de novo* selection of RNA binding elements via SELEX (Systematic Evolution of Ligands by Exponential enrichment), a method that has been used to identify synthetic aptamers that exhibit high affinity to specific ligands of interest (Tuerk and Gold, 1990). The first experiment deploying this method was for the detection of RNAs that are able to interact with T4 DNA polymerase (Tuerk and Gold, 1990). Since then, the targets for these SELEX sourced aptamers have greatly expanded to encompass a variety of molecules ranging from inorganic molecules to small organic molecules that include complex structures (Stoltenburg *et al.*, 2007), amino acids (Lozupone *et al.*, 2003), metal ions (Kawakami *et al.*, 2000), small organic compounds (Mann *et al.*, 2005), biological factors (Lauhon and Szostak, 1995), metabolites (Long *et al.*, 2016), proteins (Hassan *et al.*, 2017) and antibiotic (Berens *et al.*, 2001; Wallace and Schroeder, 1998).

A SELEX process starts with chemically synthesizing random oligonucleotide libraries consisting of 10^{13} to 10^{15} different sequence motifs that are then incubated directly with the target. The oligo-target complexes formed are subsequently partitioned from unbound and weakly bound oligonucleotides. The oligo-target complexes are eluted and amplified by PCR for DNA SELEX or subject to reverse transcription (RT)-PCR for RNA SELEX. The resulting bound oligo needs to be transformed into a new oligonucleotide pool. The new and enriched pool of selected oligonucleotides is used for the binding reaction with the target in the next SELEX round. Through iterative cycles of selection and amplification, the initial random oligonucleotide pool is reduced to relatively few sequence motifs with the highest affinity and specificity for the target.

High throughput sequencing (HTS) has revolutionized the SELEX technique (Nguyen Quang *et al.*, 2016) because after a series (> 15) of selection cycles, millions of aptamer sequences can be determined from HTS datasets. Programs such as Aptamotif, MPBind, AptaCluster, APTANI and AptaPLEX were developed to analyse the fast emerging data (Hoinka *et al.*, 2012, 2014; Jiang *et al.*, 2014; Caroli *et al.*, 2016; Hoinka and Przytycka, 2016). In this paper, the tools used to analyse SELEX data and for predicting the secondary structures of riboregulators are discussed.

Background

Riboregulator

RNAs are known to mediate gene expression in both prokaryotes and eukaryotes. In prokaryotes, riboregulators such as riboswitches and ribozymes control gene expression via *cis* or *trans* interactions with their targets (Winkler *et al.*, 2004; Mandal *et al.*, 2003). In eukaryotes, small RNAs act as microRNAs in order to control gene expression (Gottesman, 2002). The regulatory capacity of such RNAs have led to *in-vitro* and *in-silico* riboregulator design that impact transcription and translation in response to specific input.

Early studies of riboregulators were conducted by Isaac and co-workers (Isaacs *et al.*, 2004). The first type of engineered riboregulator consisted of complementary *cis* RNA sequence located upstream of the ribosome binding site (RBS) in a target gene. Upon transcription, the RNA transcript forms a stem-loop that interferes with ribosome binding. However, the subsequent binding of a cognate *trans*-acting RNA acts to promote and allow translation.

Another type of riboregulator, the toehold switch, is based on RNA-RNA interactions between the switch and the trigger RNAs (Green *et al.*, 2014). The switch RNA, also known as the transducer strand, contains the initial binding site for the trigger or *trans*-acting RNA strand, ribosome binding site (RBS), start codon, a 21nts linker sequence, followed by the coding sequence of the gene. The upstream of the coding sequence forms a hairpin that blocks translation. The complementary trigger RNA strand that binds to the initial binding site will expose the RBS and the start codon in order to initiate translation of the target gene (Green *et al.*, 2014).

Both riboregulators were rationally designed and comprised of a combination of aptamer and actuator domains. At present, secondary structure prediction is commonly used to assess the quality of the riboregulators (Wachsmuth *et al.*, 2013; Topp *et al.*, 2010).

Synthetic Riboswitches

Riboswitches have become an important tool in synthetic biology due to their ability to regulate gene expression and to integrate into a more complex genetic circuit (Topp and Gallivan, 2010). Synthetic riboswitches are widely used as riboregulators; for example, those that detect theophylline and antibiotics such as tetracycline and streptomycin, have been used to activate transcription, termination and protein translation by designing different actuator domains (Wachsmuth *et al.*, 2015; Desai and Gallivan, 2004; Domin *et al.*, 2017).

Systems and/or Applications

Computer programs can be used to filter SELEX or HT-SELEX data to discover new aptamers. The assembly of aptamer and actuator domains can then be analyzed using available webserver programs.

Input Formats

Most software accept FASTQ and FASTA formatted files as input. Secondary structure prediction via RNAfold, RNAstructure and Mfold accept a single RNA or DNA sequence in plain text or FASTA format as input.

Output and Average Time for Computations

Seven aptamer analysis software are discussed in this chapter. On average, these programs take up to several hours to run, depending on the size of the input data and the technical specifications of the computer being used. For secondary structure prediction, most tools take from only a few seconds to a few minutes to provide the predicted minimal free energy (MFE) secondary structure.

Analysis and Assessment

Aptamer Analysis

Seven programs were chosen to analyse aptamer sequences from available SELEX/HT-SELEX data. The programs selected were written in Python, Perl and C + +. The general approach taken by all tools is summarized in Table 1.

Aptamotif, APTANI and AptATRACE deploy principal sequence-structure analysis to filter for aptamers (Hoinka *et al.*, 2012; Caroli *et al.*, 2016; Dao *et al.*, 2016). APTANI builds on the Aptamotif algorithm and extends its usage to analyse both SELEX and

Table 1 List of tools for the analysis of SELEX/HT-SELEX. A list of aptamer analysis programs available and the approach used by each program in analyzing aptamers

Name (Language)	Input	Search approach	Reference
Aptamotif (Python)	FASTQ	Identify sequence-structure motifs	Hoinka <i>et al.</i> (2012)
MPBind (Python/R)	FASTA/FASTQ	Predict the binding potential combination of binding of all <i>n</i> -mers within a sequence	Jiang <i>et al.</i> (2014)
AptaCluster (C + +)	FASTA/FASTQ	Cluster aptamers sequence based on k-mer counting	Hoinka <i>et al.</i> (2014)
FASTAptamer (Perl)	FASTQ	Compare population for sequence distribution, generates sequence cluster, calculator fold enrichment value and search for multiple sequence motifs	Alam <i>et al.</i> (2015)
APTANI (Python 3.3)	FASTQ	Identify aptamer motif from HT-SELEX and secondary structure information	Caroli <i>et al.</i> (2016)
AptaPLEX (C + +)	FASTQ	Standalone demultiplexing software for HT-SELEX	Hoinka and Przytycka (2016)
AptaTRACE (C + +, Java)	FASTQ	Identifying sequence structure motifs that show a signature of selection	Dao <i>et al.</i> (2016)

HT-SELEX data. Both tools use RNAsubopt to predict secondary structures in a specific energy range (Wuchty *et al.*, 1999; Backofen and Will, 2004). AptaTRACE requires Sfold or CapR for secondary structure prediction. The Sfold program needs to be installed, while CapR has been integrated into the program (Ding *et al.*, 2004; Fukunaga *et al.*, 2014). The AptaTrace process is divided into three parts: data processing, secondary structure profiling and motif extraction. The data processing involves selecting sequences as input based on user defined frequency thresholds. Then, AptaTRACER will select sequence motif(s) with the tendency of residing in a hairpin, bulge loop, inner loop, multiple loop, dangling end or being paired convergently into a specific structural context. The distribution of the k-mer's secondary structure contacts (K-context distribution) was estimated using the relative entropy (KL-divergence). In Aptamotif and APTANI, four different types of secondary structure motifs are inspected, i.e. hairpin loop left, hairpin loop right, bulge loop and intra-strand loop. The AptaTRACE program identifies hairpins, bulge loops, inner loops, multibranch loops and dangling ends.

In contrast, MPBind predicts aptamers based on a statistical test (Jiang *et al.*, 2014). The aptamer sequences are assumed to consist of combination of n -mer. The combination of n -mer is given a score based on relative frequency change and relative abundance in the database. Then the potential binding aptamer is inferred from the combination of all n -mer within the sequence.

FASTAptamer is a multi-instruction tool starts by determining sequence frequency, multiplicity or copy number via FASTAptamer-Count. The output of the analysis can then be applied as input into downstream FASTAptamer scripts such as FASTAptamer-compared, FASTAptamer-cluster, FASTAptamer-enrich, FASTAptamer-search, other FASTAptamer scripts or any program that accepts FASTAQ formatted files as input (Fig. 1). Using available scripts, the inputs are compared to the population for each sequence distribution, clusters are then generated for the sequences, the fold-enrichment of the sequences throughout the course of a selection are calculated and then the search for degenerate sequence motifs in the SELEX/HT-SELEX data is carried out (Alam *et al.*, 2015).

AptaCluster clusters SELEX/HT-SELEX data based on a calculated k-mer similarity measure (Hoinka *et al.*, 2014) that is executed in three steps. First, the data is compressed and filtered using Locality Sensitive Hashing (LSH). After filtration, the sequences with the same hash values will be in the same pool (Andoni *et al.*, 2013). Next, the clusters are extracted based on the high frequency of the sequence in the pool. Then the k-mer based on distance function is used to compute the distance between sequences and cluster them.

AptaPLEX is a standalone and independent demultiplexer specifically designed for HT-SELEX (Hoinka and Przytycka, 2016). The program can be used to analyse single or multiple HT-SELEX experiments at once. Aptaplex automatically recognizes and handles gzip compressed data and makes use of all available processing resources via its multi-threaded design while minimizing memory usage.

Secondary Structure Prediction

The stability and biochemical properties of a RNA molecule is largely determined by the base pairing interactions. After the aptamer couples with the actuator domain, secondary structure prediction is carried out. This step is important to visualise the folding of the aptamer domain, the interaction between the aptamer domain and actuator domain, or to determine the stability of the folding via its minimum free energy. Several webservers are available for secondary structure prediction - RNAstructure, RNAfold and Mfold are three of the software discussed here (Reuter and Mathews, 2010; Gruber *et al.*, 2008; Zuker, 2003).

RNAstructure uses thermodynamics and utilizes parameters from the Turner group to predict secondary structure (Reuter and Mathews, 2010). This webserver can also predict base pair probabilities using a partition function, in addition to bimolecular secondary structure and common secondary structure prediction for two or more sequences. The bimolecular secondary structure prediction folds two sequences into their lowest hybrid free energy conformation. The server combines the capabilities of bimolecular folding and duplex folding to create two distinct sets of possible bimolecular structures. To predict the common secondary structure for two or more sequences, the server combines the capabilities of Multilign and TurboFold to create distinct sets of possible structures for multiple sequences.

RNAfold is a secondary structure prediction program that is built based on algorithms proposed by Zuker and Stiegler, and John McCaskill (Gruber *et al.*, 2008). Zuker and Stiegler proposed a dynamic programming algorithm to predict the minimum free energy (MFE) of a sequence, while John McCaskill calculated the equilibrium base pairing probability of secondary structures

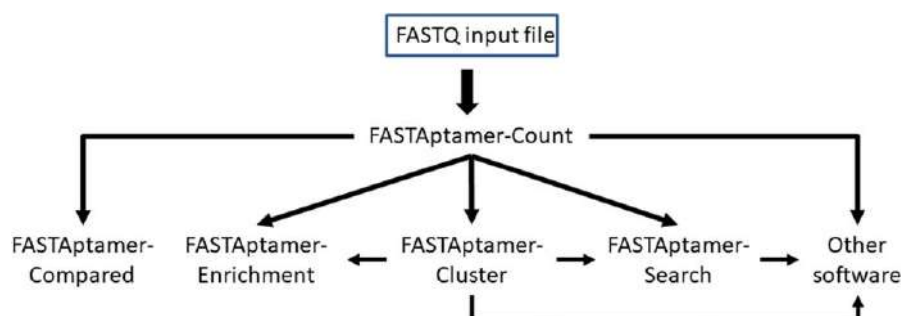


Fig. 1 Flow chart of FASTAptamer tools. FASTAptamer consist of multiple tools to analyse data from SELEX or HT-SELEX.

through a partition function (PF) algorithm (Zuker and Stiegler, 1981; McCaskill, 1990). However, it is also suggested that the predicted structures can be further augmented with reliability information to get more accurate structures.

Mfold is used to predict the secondary structures of RNA or DNA sequences (Zuker, 2003). Depending on the input sequence, mfold predicts a minimum free energy, ΔG , as well as the minimum free energies for all potential foldings. Over the years, new functions have been implemented such as the prediction of two state melting temperature and hybridization of two RNA or DNA strands.

Illustrative Examples and Case Studies

Previous experiments have demonstrated that the combination of SELEX, high throughput sequencing and bioinformatics can successfully identify aptamers for specific ligands such as LAG3 and TIM3 (Soldevilla *et al.*, 2017; Hervás-Stubbs *et al.*, 2016; Pei *et al.*, 2017; Eaton *et al.*, 2015). However, these identified aptamers were not reported to be riboregulators. The reason why the identified aptamers were not used as riboregulators because they are cytotoxic, insoluble, highly reactive and impermeable to cell membranes.

There are three aptamers identified via SELEX that have been reported as riboregulators namely the theophylline, tetracycline and streptomycin aptamers. The theophylline aptamer, can bind to theophylline with high affinity compared to other ligands and is the most prominent aptamer that can be manipulated as a riboregulator (Berens *et al.*, 2001; Wallace and Schroeder, 1998; Jenison *et al.*, 1994). The theophylline aptamer has been exploited to trigger expression via translation and transcription depending on the actuator domain (Wachsmuth *et al.*, 2013, 2015; Desai and Gallivan, 2004; Suess *et al.*, 2004). In 2004, a translational control element was carried out by cloning a theophylline aptamer upstream of the ribosome binding site of the reporter gene (Desai and Gallivan, 2004; Suess *et al.*, 2004). In 2013, the theophylline aptamer was fused to an intrinsic terminator thus creating a transcriptional riboregulator (Wachsmuth *et al.*, 2013). The secondary structure combination of the aptamer and actuator domains were analysed using the RNAfold program of the Vienna RNA package. The RNAfold program was used to calculate the free energy value and predict the secondary structure. The conformation of this aptamer has been studied extensively and shows a similar structure to the minimum free energy (MFE) predicted by RNAfold. The construct was able to terminate the expression of a reporter gene in the absence of theophylline. The same approach was used to design riboregulators using tetracycline and streptomycin as ligands (Domin *et al.*, 2017). The tetracycline and streptomycin aptamers were cloned into the 5' untranslated region of the *bgab* reporter. The aptamer responded to tetracycline and showed β -galactosidase activity.

Discussions and Future Directions

In the early 2000s, aptamers present in riboregulators were isolated *in vitro* and none of them were identified using SELEX/HT-SELEX analysis programs (Berens *et al.*, 2001; Wallace and Schroeder, 1998; Jenison *et al.*, 1994). The development of SELEX/HT-SELEX analysis programs had only started in early 2010 although the search for aptamers started 20 years ago. The programs can

Table 2 Characteristic of tools for the analysis of SELEX and HT-SELEX data. Characteristic differences of tools for the analysis of SELEX and HT-SELEX data

	<i>Aptamotif</i>	<i>APTANI</i>	<i>Aptatrace</i>	<i>MPBind</i>	<i>AptaCluster</i>	<i>FASTAptamer</i>	<i>AptaPLEX</i>
SELEX	✓	✓	✓	✓	✓	✓	
HT-SELEX		✓	✓		✓	✓	✓
Program language	Phython	Python 3.3	C + +	Python 2.7	C + +	Perl	C + +
Strategies							
Motif structure prediction	✓ Mfold	✓ RNAsubopt	✓				
MFE	✓	✓					
Secondary structure Motif	✓	✓		✓			
Loop structure investigation	✓	✓	✓				
Cluster investigation		✓ Cluster Omega			✓ Hamming edit distance	✓ Levenshtein edit distance	
Statistical test				✓			
Secondary structure	Mfold			RNAfold			
Minimum free energy	✓			✓			
Other dependencies		Cluster Omega, RNAsubopt	Sfold, CapR, CMAKE	None	Boost libraries, MySQL database	None	Boost libraries
License		None				GNU general public License V3	GLP v.2
Output can be used as input for other analysis						✓	✓
Equilibrium base pairing probability				✓			

be group into four strategies; i. sequence-structure motifs ii. statistical analysis iii. clustering and iv. multiplexer. The differences and similarities of each SELEX/ HT-SELEX analysis program are summarized in [Table 2](#).

In order to use these programs, they need to be installed together with their software dependencies. Fastaaptamer and MpBind are the only programs that do not require any additional software to be installed. Among these programs, FASTAaptamer is the most user friendly program. It requires only a basic knowledge of command line operations and the user's manual include the command line options. Advanced users can opt to integrate the analysis into a customized workflow or an existing sequence analysis pipeline ([Alam et al., 2015](#)).

However, to our knowledge, there are no known assessments that compare the efficiency, accuracy and speed of each program in analyzing SELEX or HT-SELEX data. It is necessary to examine more sensitive aptamers to a simple ligand so that they can be explored for eventual application riboregulator. The information obtained can be analyzed using the existing program. However, existing programs can be improved to make it easier to use and analyse data more accurately for shorter time.

Riboregulators generated from the mixing and matching of aptamer and actuator domains can work well with their specific ligands. To avoid false positives, a rationally designed riboregulator is evaluated through iterative design-build-test cycles. This is followed by global modelling of the RNA structure. As RNA sequence folds into secondary structure during transcription and this process is difficult to mimic *in vitro*, it is important to carry out the secondary structure prediction prior to actual synthesis of the candidates.

In the case of riboregulator design, it is suggested to use stochastic kinetic folding simulation approach ([Endoh and Sugimoto, 2015](#); [Thimmaiah et al., 2015](#); [Carothers et al., 2011](#)). In addition, an *in-silico* prediction method that can perform direct comparisons to *in-vivo* gene regulation activities through parallel assays should be developed ([Geertz et al., 2012](#)).

Closing Remarks

The construction of riboregulators via rational design can be a future conventional approach to analyse RNA sequences engaged in gene regulation. As more RNA data are made available, computational design of synthetic riboregulators provides a fast and cost-effective way to discover their potential applications in the organism of interest. However, the computational approaches should ideally be combined with laboratory evidence in order to best evaluate the versatility and functionality of the design.

Acknowledgement

MFR is funded by the Ministry of Science, Technology and Innovation grant 02-01-02-SF1278 and the Universiti Kebangsaan Malaysia grant DIP-2017-013.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Biomolecular Structures: Prediction, Identification and Analyses. Natural Language Processing Approaches in Bioinformatics. Predicting RNA-RNA Interactions in Three-Dimensional Structures. Sequence Analysis. Sequence Composition

References

- Alam, K.K., Chang, J.L., Burke, D.H., 2015. FASTAptamer: A bioinformatic toolkit for high-throughput sequence analysis of combinatorial selections. *Molecular Therapy- Nucleic Acids* 4, e230.
- Andoni, A., Indyk, P., 2013. In: *Conference Proceedings, 2006 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 459–468. Berkeley, California: IEEE.
- Backofen, R., Will, S., 2004. Local sequence-structure motifs in RNA. *Journal of Bioinformatics and Computational Biology* 2, 681–698.
- Berens, C., Thain, A., Schroeder, R., 2001. A tetracycline-binding RNA aptamer. *Bioorganic & Medicinal Chemistry* 9, 2549–2556.
- Caroli, J., Taccioli, C., De La Fuente, A., et al., 2016. APTANI: A computational tool to select aptamers through sequence-structure motif analysis of HT-SELEX data. *Bioinformatics* 32, 161–164.
- Carothers, J.M., Goler, J.A., Juminaga, D., Keasling, J.D., 2011. Model-driven engineering of RNA devices to quantitatively program gene expression. *Science* 334, 1716–1719.
- Chang, A.L., Wolf, J.J., Smolke, C.D., 2012. Synthetic RNA switches as a tool for temporal and spatial control over gene expression. *Current Opinion in Biotechnology* 23, 679–688.
- Dao, P., Hoinka, J., Takahashi, M., et al., 2016. AptATRACE elucidates RNA sequence-structure motifs from selection trends in HT-SELEX experiments. *Cell Systems* 3, 62–70.
- Desai, S.K., Gallivan, J.P., 2004. Genetic screens and selections for small molecules based on a synthetic riboswitch that activates protein translation. *Journal of the American Chemical Society* 126, 13247–13254.
- Ding, Y., Chan, C.Y., Lawrence, C.E., 2004. Stold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Research* 32, W135–W141.
- Domin, G., Findeiß, S., Wachsmuth, M., et al., 2017. Applicability of a computational design approach for synthetic riboswitches. *Nucleic Acids Research* 45, 4108–4119.
- Dunn, M.R., Jimenez, R.M., Chaput, J.C., 2017. Analysis of aptamer discovery and technology. *Nature Reviews Chemistry* 1, 0076.
- Eaton, R.M., Shallick, J.A., Mael, L.E., et al., 2015. Selection of DNA aptamers for ovarian cancer biomarker HE4 using CE-SELEX and high-throughput sequencing. *Analytical and Bioanalytical Chemistry* 407, 6965–6973.
- Endoh, T., Sugimoto, N., 2015. Rational design and tuning of functional RNA switch to control an allosteric intermolecular interaction. *Analytical Chemistry* 87, 7628–7635.
- Fukunaga, T., Ozaki, H., Terai, G., et al., 2014. CapR: Revealing structural specificities of RNA-binding protein target recognition using CLIP-seq data. *Genome Biology* 15, R16.

- Geertz, M., Shore, D., Maerkl, S.J., 2012. Massively parallel measurements of molecular interaction kinetics on a microfluidic platform. *Proceedings of the National Academy of Sciences* 109, 16540–16545.
- Gottesman, S., 2002. Stealth regulation: Biological circuits with small RNA switches. *Genes & Development* 16, 2829–2842.
- Green, A.A., Silver, P.A., Collins, J.J., Yin, P., 2014. Toehold switches: *De-novo* designed regulators of gene expression. *Cell* 159, 925–939.
- Gruber, A.R., Lorenz, R., Bernhart, S.H., *et al.*, 2008. The Vienna RNA websuite. *Nucleic Acids Research* 36, W70–W74.
- Hassan, E.M., Willmore, W.G., McKay, B.C., DeRosa, M.C., 2017. In vitro selections of mammaglobin A and mammaglobin B aptamers for the recognition of circulating breast tumor cells. *Scientific Reports* 7, 14487.
- Hervas-Stubbs, S., Soldevilla, M.M., Villanueva, H., *et al.*, 2016. Identification of TIM3 2'-fluoro oligonucleotide aptamer by HT-SELEX for cancer immunotherapy. *Oncotarget* 7, 4522–4530.
- Hoinka, J., Berezynoy, A., Sauna, Z.E., *et al.*, 2014. AptCluster – A method to cluster HT-SELEX aptamer pools and lessons from its application. *Research in Computational Molecular Biology* 8394, 115–128.
- Hoinka, J., Przytycka, T., 2016. AptAPLEX – A dedicated, multithreaded demultiplexer for HT-SELEX data. *Methods* 106, 82–85.
- Hoinka, J., Zotenko, E., Friedman, A., *et al.*, 2012. Identification of sequence–structure RNA binding motifs for SELEX-derived aptamers. *Bioinformatics* 28, i215–i223.
- Isaacs, F.J., Dwyer, D.J., Ding, C., *et al.*, 2004. Engineered riboregulators enable post-transcriptional control of gene expression. *Nature Biotechnology* 22, 841–847.
- Jenison, R.D., Gill, S.C., Pardi, A., Polisky, B., 1994. High-resolution molecular discrimination by RNA. *Science* 263, 1425.
- Jiang, P., Meyer, S., Hou, Z., *et al.*, 2014. MPBind: A meta-motif-based statistical framework and pipeline to predict binding potential of SELEX-derived aptamers. *Bioinformatics* 30, 2665–2667.
- Kawakami, J., Imanaka, H., Yokota, Y., Sugimoto, N., 2000. In vitro selection of aptamers that act with Zn²⁺. *Journal of Inorganic Biochemistry* 82, 197–206.
- Lauhon, C.T., Szostak, J.W., 1995. RNA aptamers that bind flavin and nicotinamide redox cofactors. *Journal of the American Chemical Society* 117, 1246–1257.
- Liang, J.C., Bloom, R.J., Smolke, C.D., 2011. Engineering biological systems with synthetic RNA molecules. *Molecular Cell* 43, 915–926.
- Long, Y., Pfeiffer, F., Mayer, G., *et al.*, 2016. Selection of aptamers for metabolite sensing and construction of optical nanosensors. In: Mayer, G. (Ed.), *Nucleic Acid Aptamers: Selection, Characterization, and Application*. New York, NY: Springer New York, pp. 3–19.
- Lozupone, C., Changayil, S., Majerfeld, I., Yarus, M., 2003. Selection of the simplest RNA that binds isoleucine. *RNA* 9, 1315–1322.
- Mandal, M., Boese, B., Barrick, J.E., *et al.*, 2003. Riboswitches control fundamental biochemical pathways in *Bacillus subtilis* and other bacteria. *Cell* 113, 577–586.
- Mann, D., Reinemann, C., Stollenburg, R., Strehlitz, B., 2005. In vitro selection of DNA aptamers binding ethanolamine. *Biochemical and Biophysical Research Communications* 338, 1928–1934.
- McCaskill, J.S., 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* 29, 1105–1119.
- Nguyen Quang, N., Perret, G., Ducongé, F., 2016. Applications of high-throughput sequencing for *in vitro* selection and characterization of aptamers. *Pharmaceuticals* 9, 76.
- Nomura, Y., Yokobayashi, Y., 2007. Reengineering a natural riboswitch by dual genetic selection. *Journal of the American Chemical Society* 129, 13814–13815.
- Pei, S., Slinger, B.L., Meyer, M.M., 2017. Recognizing RNA structural motifs in HT-SELEX data for ribosomal protein S15. *BMC Bioinformatics* 18, 298.
- Reuter, J.S., Mathews, D.H., 2010. RNAstructure: Software for RNA secondary structure prediction and analysis. *BMC Bioinformatics* 11, 129.
- Shcherbinin, D.S., Gnedenko, O.V., Khmeleva, S.A., *et al.*, 2015. Computer-aided design of aptamers for cytochrome p450. *Journal of Structural Biology* 191, 112–119.
- Soldevilla, M.M., Hervas, S., Villanueva, H., *et al.*, 2017. Identification of LAG3 high affinity aptamers by HT-SELEX and Conserved Motif Accumulation (CMA). *PLOS ONE* 12, e0185169.
- Stollenburg, R., Reinemann, C., Strehlitz, B., 2007. SELEX- A (r)evolutionary method to generate high-affinity nucleic acid ligands. *Biomolecular Engineering* 24, 381–403.
- Suess, B., Fink, B., Berens, C., *et al.*, 2004. A theophylline responsive riboswitch based on helix slipping controls gene expression in vivo. *Nucleic Acids Research* 32, 1610–1614.
- Thimmaiah, T., Voje, W.E., Carothers, J.M., 2015. Computational design of RNA parts, devices, and transcripts with kinetic folding algorithms implemented on multiprocessor clusters. In: Marchisio, M.A. (Ed.), *Computational Methods in Synthetic Biology*. New York, NY: Springer New York, pp. 45–61.
- Topp, S., Gallivan, J.P., 2010. Emerging applications of riboswitches in chemical biology. *ACS Chemical Biology* 5, 139–148.
- Topp, S., Reynoso, C.M.K., Seeliger, J.C., *et al.*, 2010. Synthetic riboswitches that induce gene expression in diverse bacterial species. *Applied and Environmental Microbiology* 76, 7881–7884.
- Tuerk, C., Gold, L., 1990. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science* 249, 505–510.
- Wachsmuth, M., Domin, G., Lorenz, R., *et al.*, 2015. Design criteria for synthetic riboswitches acting on transcription. *RNA Biology* 12, 221–231.
- Wachsmuth, M., Findeiß, S., Weissheimer, N., *et al.*, 2013. *De novo* design of a synthetic riboswitch that regulates transcription termination. *Nucleic Acids Research* 41, 2541–2551.
- Wallace, S.T., Schroeder, R., 1998. In vitro selection and characterization of streptomycin-binding RNAs: Recognition discrimination between antibiotics. *RNA* 4, 112–123.
- Winkler, W.C., Nahvi, A., Roth, A., *et al.*, 2004. Control of gene expression by a natural metabolite-responsive ribozyme. *Nature* 428, 281.
- Wuchty, S., Fontana, W., Hofacker, I.L., Schuster, P., 1999. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers* 49, 145–165.
- Zuker, M., 2003. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research* 31, 3406–3415.
- Zuker, M., Stiegler, P., 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research* 9, 133–148.

Genome-Wide Probing of RNA Structure

Xiaojing Huo, Jeremy Ng, Mingchen Tan, and Greg Tucker-Kellogg, National University of Singapore, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction

RNA is among the most versatile molecules in biology. While the information-encoding role of RNA has been known since the discovery of messenger RNA, research over the last three decades has uncovered functional roles for RNA in a vast range of processes (Cech and Steitz, 2014), including regulation of transcription, RNA splicing and editing, catalysis of protein synthesis (Nissen *et al.*, 2000), and control of chromatin structure (Khalil *et al.*, 2009; Názzer and Lei, 2014). To carry out its functions RNA must form secondary structures and fold into tertiary structures to bind small molecules and proteins, as well as to respond to environmental cues (Cruz and Westhof, 2009; Tinoco and Bustamante, 1999). Understanding these structures and how they form is essential for understanding how RNA performs its functions.

While the proper folding of RNA is crucial for its normal functions, RNA misfolding often leads to dysregulation and contributes to disease. For example, a large number of pathogenic RNA structures have been found in neurological diseases, including APP for Alzheimer's disease and -synuclein for Parkinson's disease with the formation of internal ribosome entry sites in mRNAs (Bernat and Disney, 2015). The motor neuron disease-associated RNA-binding proteins TDP-43 and FUS have been reported to function as RNA chaperones to ensure proper folding of repeat-associated RNAs for translation (Ishiguro *et al.*, 2017). Outside of neuroscience, a semiglobular ribozyme RNA can misfold into a dimer and activate the innate immune response protein, Protein Kinase R (PKR), providing a potential link between RNA misfolding and disease via innate immunity (Heinicke and Bevilacqua, 2012).

Here we consider RNA secondary structure to comprise regions of RNA base pairing, loops, and hairpins that form a hierarchical architecture (Cruz and Westhof, 2009), and tertiary structure to comprise the (often divalent cation-dependent) interactions between secondary structural units. Some recurrent structural features such as pseudoknots may be considered secondary structures despite not being strictly hierarchical, while loop-loop interactions involving Watson-Crick base pairing (Lee and Crothers, 1998) are usually considered tertiary structures.

X-ray crystallography and NMR spectroscopy have been used to determine numerous high resolution three-dimensional structures for both RNA and ribonucleoprotein complexes (Feigon, 2015; Westhof, 2015). Many now classic secondary structural patterns in RNA, such as the cloverleaf base pairing structures of tRNA (Kim *et al.*, 1974) and ultra-stable RNA stem loops (Cheong *et al.*, 1990) have been defined in three dimensions, while important new tertiary structural motifs have been identified and characterized (Devi *et al.*, 2015; Nissen *et al.*, 2001). Structure determination studies have also been used to dissect the detailed structural features of catalytic RNA function, such as the basis of polypeptide bond formation in protein synthesis (Nissen *et al.*, 2000) and the nuclease function of RNase P (Reiter *et al.*, 2010).

The size and flexibility of many RNA molecules, however, make determination of their full three dimensional structures impossible. RNA molecules undergo dynamic changes in structure over an enormous range of timescales, and large RNA molecules are often best described as an ensemble of structures (Mustoe *et al.*, 2014). Refolding of RNA molecules *in vitro* is highly dependent on experimental conditions and at best can represent only a subset of the dynamic conformations assumed by RNA in the cell. In addition, RNA functions are both so widespread and so dynamic that they need to be understood in concert across the genome in addition to being determined one structure at a time.

Secondary structure prediction applications are discussed elsewhere in this volume need cross-reference and have been greatly aided by improved thermodynamic rules from experimental studies as well as algorithms to enumerate multiple structures from a thermodynamic ensemble (Mathews and Turner, 2006). Free energy minimization of secondary structure from sequence, however, is far from perfect, and is greatly improved by structural constraints provided by experimental data (Hajdin *et al.*, 2013; Mathews *et al.*, 2004).

A widely used and powerful approach for interrogating RNA structure is to challenge RNA molecules in solution with structure-dependent chemical or enzymatic probes. Structural probing experiments can identify regions of sequence that are accessible to cleavage or modification, and thus infer which nucleotides are more likely to be double stranded, single stranded, flexible, constrained, or involved in tertiary interactions. The traditional approach is a form of footprinting experiment: treating an RNA with a reagent to induce modification or cleavage, and then detecting termination of reverse transcription (RT) products on a polyacrylamide gel. A simple interpretation is for the footprints (indicating termination of RT) to be evenly distributed and more prevalent along the unfolded stretches of sequence. The termination positions are determined by polyacrylamide gel electrophoresis and quantitative analysis from the changes of footprint on, say, unfolding. Thus, it is possible to detect location of the secondary and tertiary structure of RNA (Peattie and Gilbert, 1980).

While powerful, conventional RNA structure probing approaches have also suffered from severe limitations. Enzymatic probes and many chemical probes cannot be employed for *in vivo* studies, and traditional chemical probes are dependent on the RNA sequence as well as structure. RNA molecules are interrogated one species at a time. Using conventional polyacrylamide gels to analyze RT termination severely restricts the size of RNA fragments that can be analyzed, and interrogates only the first modified position of any RNA molecule. Finally, traditional probing methods examine structure features at individual sequence positions, but are unable to directly interrogate long range interactions, which form the vast majority of RNA secondary and tertiary structure.

Advances over the last decade are removing all of these limitations. Selective 2'-hydroxyl acylation and primer extension (SHAPE) chemistry makes it possible to probe RNA flexibility at single nucleotide resolution independent of nucleotide identity (Merino *et al.*, 2005). Capillary electrophoresis increases the size and throughput of structure probing experiments over conventional gels (Mitra *et al.*, 2008; Wilkinson *et al.*, 2008). Conditions that convert adducts to mutations during reverse transcription allow measurement of correlated modifications at multiple positions (Siegfried *et al.*, 2014; Zubrady *et al.*, 2016). High throughput sequencing technology makes it possible to simultaneously interrogate large numbers of RNAs, even whole transcriptomes, in a single experiment (Kwok, 2016). Finally, direct chemical probes of base pairing make it possible to interrogate long range interactions *in vitro* and *in vivo* (Aw *et al.*, 2016; Lu *et al.*, 2016; Sharma *et al.*, 2016).

Computational innovations have accompanied each new experimental advance. New computational methods and bioinformatics resources not only help researchers interpret high throughput data from new experimental technologies, but also help to improve predictions of RNA folding using both hard and soft constraints supplied from experiments. One happy result is that databases have arisen to store and organize the increasing output of RNA structure probing studies, and now make structure probing results openly available to researchers. The vast increase in structural data is creating new opportunities to understand the rules of RNA folding.

Background

Chemical and Enzymatic Structure Probing of RNA

Both chemical reagents and nucleases can be used as structure-dependent probes by cleaving RNA or creating covalent modifications that can be detected by subsequent cleavage and direct labeling or primer extension (Ziehler and Engelke, 2001). Chemical probes of RNA structure have been used for RNA footprinting studies since 1980 (Peattie and Gilbert, 1980). The widely used chemical probe dimethyl sulfate (DMS), for example, methylates the N1 of adenine, the N7 of guanine, and the N3 of cytidine. When a reverse transcriptase is used to generate a cDNA from a DMS-modified RNA, the primer extension halts opposite modified A and C nucleotides; the first modified position is identified via polyacrylamide gel electrophoresis. Since the positions methylated by DMS are directly involved in double stranded base pairing, accessibility to DMS modification is a surrogate measure of single-stranded A and C positions (Tijerina *et al.*, 2007).

A number of other base-specific probes are used besides DMS to interrogate RNA secondary structure. For example, kethoxal modifies guanines and CMCT can modify U or G; these modifications only occur at single-stranded regions and so can be used as probes of secondary structure (Brow and Noller, 1983; Ho and Gilham, 1971). When multiple chemical probes with different specificities are used in a combination of experiments, base-specific chemical probes can interrogate every position of an RNA molecule, but individual probes that modify base moieties are inherently sequence-dependent.

For many years the only sequence-independent chemical probe in widespread use was hydroxyl radical (Tullius and Greenbaum, 2005), which leads to RNA cleavage at positions where the RNA backbone is accessible to solvent. In 2005, Kevin Weeks' team introduced a new structure probing chemistry called SHAPE that modifies the ribose moiety of an RNA molecule (Merino *et al.*, 2005). A SHAPE reagent probes nucleotide flexibility by esterifying the 2' position of a nucleotide; the reaction is inhibited by a conformationally constrained 3' phosphodiester, and so SHAPE serves as a conformational probe of RNA. Much like DMS modifications at A and C nucleotides, SHAPE modifications halt reverse transcription so that termination products can be detected on polyacrylamide gels, but unlike DMS, SHAPE can interrogate every position of an RNA sequence. DMS and SHAPE reagents are the two most widely used chemical probes of RNA structure.

While the examples above are chemical, nucleases can also function as probes of single-stranded or double-stranded RNA. Many RNases used for such experiments are base-specific (Knapp, 1989), but some more recently used endonucleases show little base specificity *in vitro* (Daou-Chabo and Condon, 2009). It should be noted that depending on enzymatic reactions for RNA structure may introduce additional dependencies such as steric accessibility of enzyme-substrate interactions.

High-Throughput Structure Probing and Automation

The throughput of RNA footprinting was dramatically improved by combining DMS and SHAPE probes with capillary electrophoresis (CE) (Mitra *et al.*, 2008; Wilkinson *et al.*, 2008), and developing automated algorithms to process and quantify chemical reactivities at single-nucleotide resolution (Mitra *et al.*, 2008; Vasa *et al.*, 2008). Capillary electrophoresis also greatly increased the size of individual RNA molecules that could be analyzed by structure probing experiments, limited by the capillary electrophoresis itself and the software's ability to align the electrophoretic signal to sequence, correct for drop-off over the length of sequence, and convert to per-nucleotide reactivities. The underlying methods are not only applicable to RNA structure studies, but for footprinting protein binding sites to RNA.

The increased throughput of capillary electrophoresis and automation applied to RNA structure probing provided a major step towards genome-scale probing of RNA structure. In addition to the technology development and invention of CE-SHAPE, Wilkinson *et al.* (2008) identified structural properties of the HIV genome that would later be recapitulated and extended in later

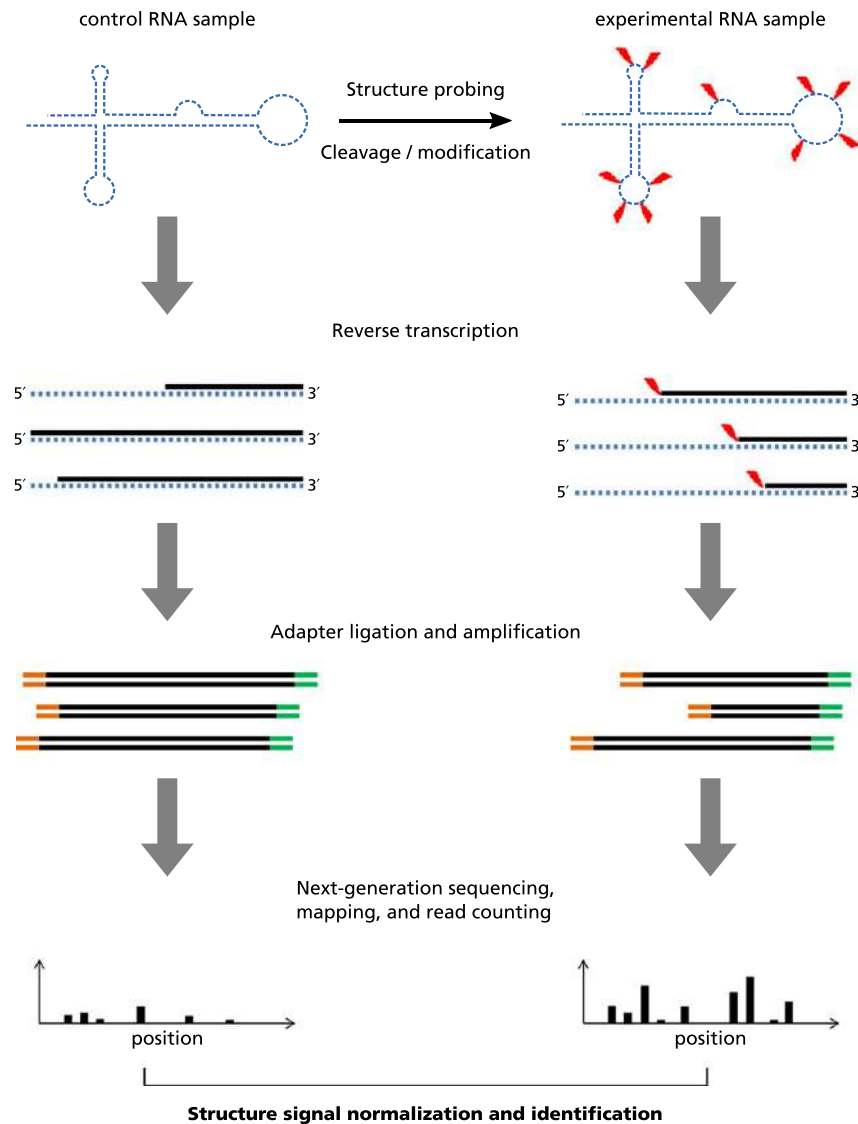


Fig. 1 .Schematic diagram of NGS-based genome-wide probing of RNA structure. The sample of an RNA population with secondary structure is split evenly into control and experiment. In experimental RNA sample, the particular structured nucleotides (double stranded or single stranded) are probed through enzymatic cleavage or chemical modification. These probing nucleotides terminate RT or introduce point mutations at the probing sites. Both control and experimental samples are processed for cDNA library preparation and NGS. The RT stops or mutations are counted on each nucleotide in experimental sample and normalized against the control. By identifying the probing signals (RT stops or mutations), the structure of each nucleotide is determined.

genome wide studies, such as the conservation of structural domains across different biological states, quantitative differences in structure between coding and non-coding regions, high levels of structure in regulatory RNA sequences, and identification of RNA recognition motifs for protein binding in intact biological systems. CE-SHAPE was also essential for the first whole-genome RNA structure studies of retroviruses such as the HIV genome (Watts *et al.*, 2009), which identified not only the secondary structure of HIV but important links between RNA structure and protein coding.

RNA Structure Probing With Deep Sequencing

Both conventional and capillary electrophoresis-based structure probing extend primers from defined sites, often combined with a structure “cassette” to ensure processivity of the reverse transcriptase. Next-generation sequencing (NGS) technology dramatically increased the potential for whole-genome studies (Aviran *et al.*, 2011; Kertesz *et al.*, 2010; Lucks *et al.*, 2011; Underwood *et al.*, 2010) of RNA structure by allowing simultaneous measurement of hundreds or thousands of RNA molecules. Rather than using labeled primers as in capillary electrophoresis, double stranded cDNA libraries are prepared after enzymatic or chemical

Table 1 Methodological advances towards genome-wide probing of RNA structure

Method	Probe	Structural specificity	Sequence specificity	Detection of probing site	RNA status	Milestone	Reference
PARS	Nuclease S1	ssRNA	None	RT stop	<i>In vitro</i> ; native deproteinized	Detection of both ss and ds RNA; genome-wide	Kertesz <i>et al.</i> (2010)
FragSeq dsRNA-seq	RNAse V1	dsRNA				Structure probing with enzymatic probe	Wan <i>et al.</i> (2012), Wan <i>et al.</i> (2013, 2014)
	Nuclease P1 RNAse I	ssRNA ssRNA	None None	RT stop RT stop	<i>In vitro</i> <i>In vitro</i> ; native deproteinized	Genome-wide structure probing with enzymatic probe Genome-wide structure probing with enzymatic probe	Underwood <i>et al.</i> (2010) Zheng <i>et al.</i> (2010), Li <i>et al.</i> (2012)
ssRNA-Seq SHAPE-seq; Shape-seq 2.0	RNAase V1	dsRNA			<i>In vitro</i>	NGS combined with SHAPE probing; rigorous model for intensity drop-off	Lucks <i>et al.</i> (2011), Loughrey <i>et al.</i> (2014)
	NMIA; 1M7	ssRNA	None	RT stop	<i>In vitro</i>	Genome-wide <i>in vitro</i> structure probing; Circularization libraries	Rouskin <i>et al.</i> (2014)
DMS-seq	DMS	ssRNA	A, C	RT stop	<i>In vitro</i> ; denatured	Genome-wide <i>in vitro</i> structure probing	Ding <i>et al.</i> (2015)
	DMS	ssRNA	A, C	RT stop	<i>In vitro</i> ; <i>in vitro</i>	Genome-wide <i>in vitro</i> structure probing	
Structure-seq	DMS	ssRNA	A, C	mutation	<i>In vitro</i>	Genome-wide RNA structure mutational profiling <i>in vivo</i> ; circularization libraries	Zubradt <i>et al.</i> (2016)
DMS-MaPseq	DMS	ssRNA	A, C	mutation	<i>In vitro</i>	Detection of long-range interactions and correlations reflecting distinct conformations	Homan <i>et al.</i> (2014)
RING-MAP	DMS	ssRNA	A, C	RT stop	<i>In vitro</i>	High-throughput sequencing for structure probing with long RNAs	Talkish <i>et al.</i> (2014)
Mod-seq	DMS	ssRNA	A, C	RT stop	<i>In vitro</i>	two probes cover all four nucleotides	Incarnato <i>et al.</i> (2014, 2015)
CIRS-seq	DMS; CMCT	ssRNA	A, C; G, U	RT stop	Native deproteinized	Enable the detection of low-abundance RNAs	Siegrfried <i>et al.</i> (2014)
SHAPE-MaP	NMIA; 1M7	ssRNA	None	mutation	<i>In vitro</i>	Modified SHAPE reagent to penetrate cell; Use biotin conjugation on probes to enrich probed RNAs	Spitale <i>et al.</i> (2015)
icSHAPE	NAI-N3	ssRNA	None	RT stop	<i>In vitro</i> ; <i>in vitro</i>	Use psoralen derivative to penetrate cell and cross-link dsRNA	Lu <i>et al.</i> (2016)
PARIS	AMT (psoralen)	dsRNA	None	Cross-links	<i>In vitro</i>	Ligation after cross-link to study RNA-RNA interactions	Sharma <i>et al.</i> (2016)
LIGR-seq	AMT (psoralen)	dsRNA	None	Cross-links	<i>In vitro</i>	Enrichment on streptavidin beads	Aw <i>et al.</i> (2016)
SPLASH	biopsoralen crosslinking	RNA-RNA interactions	None	Cross-links	<i>In vitro</i>		

modification with adapter sequences added at both ends for paired end deep sequencing (Fig. 1). The additional steps, such as conversion of the modified RNA to cDNA and preparation of the library with adapters for sequencing, add additional sources of bias and variation to the resulting signal that need to be addressed by some combination of experiment and analysis. A summary of advances is shown in Table 1.

Genome-Wide Enzymatic Probing Methods

Enzymatic probes were first used for genome-wide next-generation RNA structure probing in 2010. The methods differ in a variety of ways, as well as in their applications, but all share a common outline of *in vitro* RNA isolation, cleavage with structure-specific enzymes, adapter ligation and library preparation, and paired end sequencing.

In the development of FragSeq, Underwood *et al.* (2010) used endonuclease P1 from *Penicillium citrinum* to cut RNA preferentially at single stranded regions. RNAase P1 cleaves RNA to leave a 5' phosphate and 3' hydroxyl, features which were exploited in the adapter ligation steps of the method. True RNAase P1 digestion products were distinguished from both endogenous 5' phosphate-containing RNAs and endogenous breaks by comparing experimental libraries to control libraries prepared from undigested RNA or RNA treated with polynucleotide kinase + ATP, respectively. The authors developed a "cutting score" to interpret FragSeq data, and while they initially validated and calibrated the method on *in vitro* RNA molecules with known secondary structure, they also applied it to the entire mouse nuclear transcriptome.

In dsRNA-seq, Zheng *et al.* (2010) focused on double stranded regions of RNA from *Arabidopsis* in order to identify novel substrates of the RNA-dependent RNA polymerase 6 (RDR6), which synthesizes dsRNA during small RNA biogenesis. Unlike the other methods described here, the dsRNA-seq approach does not interrogate individual nucleotides for their structure-specific cleavage, but isolates and sequences fragmented double-stranded RNA after enzymatic depletion single stranded RNA regions.

In PARS, Kertesz *et al.* (2010) used two different enzymes with different structural specificities: RNAse VI cleaves 3' of double stranded nucleotides and endonuclease S1 specific for single stranded regions. Like FragSeq, the cleavage products contain 5' phosphates at the cleavage site and so adapter ligation is specific to this expectation. A "PARS score" is set as the log ratio of sequence reads for each nucleotide after V1 and S1 cleavage, and thus a high PARS score reflects a higher likelihood of being double stranded. The original PARS study was applied to the entire yeast transcriptome, validating known structures and characterizing previously unknown structures. Interestingly, coding regions of mRNA were more structured on average than flanking UTR sequences.

Rigorous Computational Models of High-Throughput Structure Probing Data

SHAPE provides nucleotide flexibility information at single-nucleotide resolution. The heuristic models of reverse transcriptase drop-off used for SHAPE-CE are insufficient for comprehensive analysis of next generation sequencing platform data, so for SHAPE-seq (Lucks *et al.*, 2011). Aviran *et al.* (2011) developed a rigorous maximum likelihood model for the experimental process of chemical probing followed by paired end deep sequencing. The SHAPE-seq model includes parameter estimates for the natural drop-off propensity at each nucleotide in a transcript, which is itself a property of local structure and so varies along the sequence. The chemical modification model represents the number of exposures of an RNA to an electrophile as a Poisson process. The conditions of RNA structure probing experiments are generally designed to favor single-hit kinetics, corresponding to a Poisson rate process $c \approx 1$, but under such conditions many molecules are not exposed to electrophiles at all and a significant proportion will experience multiple modifications. By incorporating the ML estimate of the natural drop-off and developing an efficient optimization procedure for the model parameters, the authors were able to obtain per-nucleotide reactivity estimates that both supported the use of deep sequencing technology for highly parallel structure probing and provided a theoretical framework to improve existing methods.

In Vivo Chemical Probing of RNA

Enzymatic probing approaches are largely limited to *in vitro* studies of RNA structure, which does not directly interrogate the state of RNA in the cell. *In vitro* probing studies also require additional preparation steps that can alter RNA structure or intentionally re-fold RNA molecules prior to interrogation. Chemical probes have the potential to penetrate the cell for *in vivo* measurements. DMS has been used to probe RNA in cells since at least 1998 (Luo *et al.*, 1998). The original SHAPE reagents react with the hydroxyl group of water, which obviates the need to quench the reaction, but relatively long reaction times prevents the SHAPE reagent NMIA from being used in intact cells. Two SHAPE reagents designed for use *in vivo* studies (NAI and FAI) were developed (Spitale *et al.*, 2013) with high solubility and reactivity; along with some use of 1M7, these reagents make *in vivo* SHAPE experiments more accessible.

Whole transcriptome *in vivo* chemical probing was further advanced with the development of icSHAPE (Flynn *et al.*, 2016), in which the NAI reagent designed for *in vivo* SHAPE was further modified by addition of an azide (NAI-N₃). Any genome wide *in vivo* profiling of RNA structure must contend with the overwhelming proportion of unmodified RNA sequence; the azido moiety on NAI-N₃ is used in a 'click' reaction to attach biotin at the modified nucleotide, and thus enrich the RNA for modifications by isolation on streptavidin beads.

Beyond One Modification Per RNA Molecule

Mutational profiling (MaP)

Identifying modification sites by truncation of reverse transcription identifies the first modified nucleotide encountered by reverse transcriptase, and necessitates conditions of one-hit kinetics. This limitation prevents the detection of correlated modifications in single RNA molecules which would give insights into alternative structures and long range interactions. To overcome this limitation, conditions of reverse transcription have been developed that allow SHAPE modifications to be read through and converted into mutations (Smola *et al.*, 2015). The result is a mutational profile, or MaP, that encodes chemical modification in sequence, thereby simplifying the library preparation steps for highly parallel sequencing, gaining accuracy for low abundance RNAs, and allowing in principle the measurement of multiple modifications in single molecules.

Multiple types of mutations are created in SHAPE-MaP experiments, so the interpretations of data is handled differently from traditional CE-SHAPE or SHAPE-seq. Software for interpreting SHAPE-MaP data (Busan and Weeks, 2018) is designed to account for the different types of mutations, including possibly ambiguous reads, in order to estimate reactivities for each nucleotide with more precision than conventional SHAPE.

Mutational profiling can be performed with DMS as well as with SHAPE reagents (Homan *et al.*, 2014; Zubradt *et al.*, 2016). Homan *et al.* (2014) used a DMS-MaP approach with conditions optimised to produce multiple modifications per RNA molecule and identify RNA interactions groups, or RINGs. The interacting groups are observed as correlated mutations from either dynamic “breathing” of RNA structures (thus identifying through-space interactions) or multiple distinct structures taken by a molecule. To identify the underlying structures, the correlated reactivities can be subjected to spectral decomposition into distinct clusters; each cluster can be modeled as arising from a distinct structural form. Interestingly, the RING-MaP approach makes it possible to characterize ligand-dependent conformational changes in riboswitches, as well as underlying structures in ensembles more generally.

Combining systematic mutation with chemical probing

An alternative approach to long range interactions is to combine structural probing with systematic mutation of a target RNA molecule. Since mutations at base paired nucleotides are likely to increase the flexibility of its normal pairing partner, a two-dimensional “mutate-and-map” strategy can identify base pairs with high precision (Kladwang and Das, 2010; Kladwang *et al.*, 2011). The systematic mutation approach of mutate-and-map adds precision to DMS and SHAPE profiling experiments, and can be used to infer long range interactions (Tian *et al.*, 2014).

Direct determination of long-range interactions

RING-MaP and mutate-and-map data provide important indirect inference of long range interactions from structure probing experiments. In 2016, three groups simultaneously developed methods for *in vivo* profiling of long range RNA interactions using variations of psoralen crosslinking (Aw *et al.*, 2016; Lu *et al.*, 2016; Sharma *et al.*, 2016). Derivatives of the small molecule psoralen have long been used to cross-link RNA. For *in vivo* cross-linking, psoralen has favorable properties: psoralen intercalates at base-paired nucleotides, some derivatives readily enter cells, and psoralen forms reversible cross-links across double-stranded RNA after illumination at 365 nm.

In LIGR-seq, Sharma *et al.* (2016) used AMT, psoralen variant that readily penetrates cells, to cross-link RNA followed by endonuclease digestion and proximity ligation to identify interactions by paired end sequencing. They applied LIGR-seq to HEK-293 cells and were able to identify both known and novel interactions involving non-coding RNAs, and highlighted a surprising number of interactions for the orphan snoRNA SNORD83B. In PARIS, Lu *et al.* (2016) also used AMT for cross-linking, but enriched the cross-linked regions with 2D gel electrophoresis. The results are discussed below as one of the case studies of genome-wide RNA structure probing. In SPLASH, Aw *et al.* (2016) used a biotinylated psoralen for cross-linking. The use of biotinylated psoralen required a mild detergent to increase cellular uptake in mammalian cells and in yeast, but enabled enrichment of crosslinked regions using streptavidin beads. The SPLASH study identified a number of clear structural properties of classes of RNA, such as the long range intramolecular interactions within UTRs of mRNAs, and newly identified interactions between snoRNAs and rRNA.

Direct detection of RNA interaction in helices is a major step forward for genome-wide RNA structure studies. Each of the methods described above uses proximity ligation and high-throughput paired end sequencing to determine gapped reads that represent long range interactions through base pairing. While most of the mapped reads are ungapped, possibly due to limitations of the proximity ligation, the sensitivity and power of direct interaction methods dramatically increase the capabilities of RNA structure probing methods.

Example Applications

Genome wide profiling RNA structure profiling applications are expanding rapidly. Here we briefly review two studies that provide some indication of how much insight these experiments can provide.

Genome Wide Measurement of RNA Unfolding

The melting temperature of a double stranded RNA molecule has long been used as a measure of folding stability. RNA unfolding is conventionally measured by change in UV absorbance. [Wilkinson *et al.* \(2005\)](#) first combined RNA structure probing with temperature-driven unfolding to determine the temperature dependence of structural interactions in the unfolding of a tRNA^{Asp} transcript. [Wan *et al.* \(2012\)](#) used the PARS approach with RNase V1 to determine the melting temperature per base across the yeast genome by a method they termed Parallel Analysis of RNA structures with Temperature Elevation (PARTE). PARS had already been used for genome-wide analysis of the yeast transcriptome, and RNase V1 was shown to retain its specificity for double stranded RNA up to 75°C, so it was possible to develop conditions for single-hit kinetics with PARS over a range of temperatures after *in vitro* RNA refolding.

A computational method for estimating T_m values for individual nucleotides was developed along with PARTE and used to estimate the T_m values for over 350,000 bases in the yeast transcriptome. When these T_m values were averaged per gene to generate an estimated melting temperature (eT_m) per gene, the eT_m values were moderately concordant with UV-measured T_m values of randomly chosen short transcripts, and more highly concordant for those transcripts with the most read coverage. The eT_m values were also correlated with T_m values predicted using RNAfold, but showed marked deviations for some molecules, suggesting that experimental temperature-dependent structure probing data could be used to improve RNA unfolding predictions much like ordinary structure probing data are used to constrain predicted RNA secondary structures.

The PARTE analysis made it possible to associate RNA folding energy with RNA feature annotations. Noncoding RNAs, for example, showed both more highly stable bases (T_m > 75°C) and higher eT_m values than mRNAs. Among mRNA transcripts, the coding sequence (CDS) appears to be more structured than the 5'-UTR and 3'-UTR regions, as had been shown earlier ([Kertesz *et al.*, 2010](#)). However, the PARTE analysis also identified a polarity of structural features (local eT_m) within mRNAs that demarcated the CDS and could not be explained by GC content alone: the most meltable region of mRNA is -3 to +3 nucleotides around the start codon and the most stable paired region is 20 nucleotides immediately upstream and 10 nucleotides immediately downstream of the stop codon. Interestingly, structural stability at the start of the CDS showed a weak inverse correlation with translation rate.

The PARTE analysis identified known and putative RNA functional elements by virtue of their structural stability. Very stable structures in 3'-UTRs, for example, included known structures directly involved in heat shock response, while 5'-UTRs were enriched in structures with lower eT_m (30–37°C) and highly enriched for genes whose expression decreased in heat shock transcriptional profiling experiments, suggesting that these low eT_m structures may function as RNA thermometers. Indeed, when RNAs were binned by eT_m values, the least stable RNAs (with eT_m 26–40°C) were predictive of the pattern of heat shock-induced *in vivo* RNA decay. The authors propose, and show data to support, that these less structured RNA regions may function as RNA thermometers of RNA decay by blocking exosome processing at low temperature. Increasingly stable RNA structures progressively blocked exosome processing, indicating the exosome recognizes unpaired bases to degrade RNAs that unfold during heat shock.

Phylogenetic Conservation of Directly Observed Duplexes

DMS-seq, structure-seq and icSHAPE have all been used to identify single-stranded regions of RNA *in vivo* by chemical probing of flexible or unpaired bases ([Ding *et al.*, 2014](#); [Rouskin *et al.*, 2014](#); [Spitale *et al.*, 2015](#)). However, while these methods provide constraints on the structure at individual nucleotide positions, they do not provide any direct information about long range interactions. [Lu *et al.* \(2016\)](#) used psoralen probing of RNA duplex in living cells, termed PARIS (psoralen analysis of RNA interactions and structures) to identify long range interactions directly via paired end sequencing (see Section “Direct Determination of Long-Range Interactions” for a discussion of the method itself).

Conservation and covariation of duplex structures

[Lu *et al.* \(2016\)](#) performed PARIS in two human cell lines (HeLa and HEK293T) and in mouse embryonic stem (mES) cells, and identified duplex groups (DGs) from gapped reads. A substantial proportion DGs spanned much larger regions of sequence (> 1000 bp) than normally identifiable or predictable, which offers new opportunities to investigate structure conservation and covariation ([Smith *et al.*, 2013](#)). By directly pairing the helices observed in DGs for human and mouse data, [Lu *et al.* \(2016\)](#) were able to perform covariation analysis on structures conserved between human and mouse, and use the PARIS-identified helices along with icSHAPE data to guide a much broader phylogenetic analysis of conserved structures.

Long range interactions are typically ignored in such structure covariation analysis because of the computational complexity of an unconstrained search; the PARIS constraints make it possible to focus directly on covariation at observed structural interactions, regardless of the distance between the paired sequence arms. In addition, PARIS data make it possible to avoid incorrectly assigned covariation searches that naturally become the focus of an unconstrained analysis.

Distinguishing between alternative RNA structures

As discussed in Section “Beyond One Modification Per RNA Molecule”, one-dimensional structure probing data using single-hit kinetics infers only an average reactivity, even if multiple alternative RNA structures exist. Mutational profiling, correlation methods, and mutate-and-map strategies provide information about dynamics and alternative structures important for RNA structure-function relationships. PARIS also provides precise information about alternative structures, which can be directly

detected as duplexes that are mutually incompatible. The authors analyzed the 50 mRNAs with the most detected helices as candidates for alternative structures: about 20%–50% of DGs from these mRNAs were found involved in at least one pair of alternative structures, suggesting that alternative structures are pervasive. The result also suggests that direct duplex detection can be used to identify functional RNA conformational switches such as riboswitches and RNA thermometers.

Higher order structure of XIST lncRNA in epigenetic silencing

PARIS, icSHAPE and phylogenetic conservation were combined to determine the higher order structure of XIST, a 19kb lncRNA essential for X chromosome inactivation, but whose structure has been contested. Global PARIS data revealed both local duplexes and multiple long-range interactions in the XIST lncRNA in what appear to be four structural domains. Phylogenetic analysis showed conservation of a subset of PARIS-determined helical structures, consistent with a functional role for three of the four domains.

An intriguing feature of XIST is the A-repeat, a spaced multicopy conserved sequence at the 5' end of the XIST RNA. Previous data had shown that the A-repeat is not required for XIST to coat the X chromosome in mouse, but is required for epigenetic silencing. The A-repeat had also been shown to be required for XIST RNA binding to the Spen protein, which provides an appealing link to epigenetic silencing. The A-repeat structure was difficult to determine, however, in part because of its repetition, and in part because of the limitations of conventional structure probing data and associated computational predictions.

The PARIS data indicate that the A-repeat region forms an isolated domain with extensive inter-repeat interactions, between the first halves of repeats, and mostly between neighboring repeats. The structure of inter-repeat units from PARIS is consistent with the icSHAPE data, and suggest that the A-repeats in XIST form a higher order structure. Since previous data had shown that Spen did not show any preference for single A-repeat units (Monfort *et al.*, 2015), it was possible that higher order structure was required for Spen RNA binding. Lu *et al.* (2016) performed iCLIP experiments to test this and found that Spen interacted virtually exclusively with the single stranded spacer regions between A-repeat units, suggesting that the A-repeat domain RNA architecture facilitates Spen binding.

Challenges and Future Directions

In this article, we have discussed the experimental and computational approaches for genome wide, next-generation sequencing-based RNA structure probing. The expanding repertoire of experimental RNA probing studies, now able to interrogate long range interactions as well as individual nucleotide positions, provide data for constraint based RNA folding algorithms that greatly exceed the performance of unconstrained folding predictions. Scientists in the field are taking advantage of these emerging technologies to provide tools and resources for the broader scientific community.

Many of the individual labs cited in this article have developed software for processing chemical probing data and producing reactivities or predictions as described in their papers. The full list is too numerous to describe here, but the citations above and the summary of Table 1 are meant to provide a brief guide.

Constraints and Databases

RNA folding and prediction models that include constraint from experimental probing data are rapidly improving. The power of such constraints has been recognized for some time (Deigan *et al.*, 2008; Low and Weeks, 2010; Weeks, 2010), but techniques such as mutate-and-map (Kladwang *et al.*, 2011) SHAPE-MaP (Smola *et al.*, 2015), and direct duplex probing methods add a great deal of power to the available constraints. Computational methods for incorporating those constraints into folding prediction continue to advance (Lorenz *et al.*, 2016; Rice *et al.*, 2014), including recent extension to homologous sequences (Tan *et al.*, 2017).

Making use of such data requires it to be organized and accessible. The FoldAtlas database (Norris *et al.*, 2017) and Structure Surfer (Berkowitz *et al.*, 2016) are intended as a comprehensive archive of RNA structure probing data. FoldAtlas includes a focus on mutate-and-map studies.

The combination of high-throughput sequencing, innovative library preparation, and rigorous computational models is rapidly creating a new understanding of RNA structure *in vitro* and *in vivo*. The development of *in vivo* probing methods is exciting for many reasons, not least because it has suggested that RNA may be less structured in the cell than often observed in the tube. It is tempting to think that whole genome RNA probing technology may obviate the need for more focused structure probing studies, but this is unlikely to be the case: focused methods such as mutate-and-map interrogate target RNAs in depth and provide unique insight into sequence-structure relationships.

Emergence of Single-Molecule Sequencing Technologies

Virtually all of the recent sequencing studies for RNA structure probing have utilized “next-generation sequencing” that provide tens of millions of relatively short paired end reads. Using this high-throughput sequencing technology for RNA structure studies

has required development of innovative library generation protocols; the protocols can be complex, but are a basic requirement of the sequencing technology. Furthermore, termination-based structure profiling is limited to one modification per molecule. While mutational profiling (MaP) approaches can read multiple modifications of a single molecule they are still limited by the underlying sequencing technology and library preparation requirements.

Recently developed single molecule sequencing technologies have been reported to read extremely long DNA and RNA sequences (Garalde *et al.*, 2018; Jain *et al.*, 2018), and can in principle directly detect covalent modifications of individual nucleotides (Vilfan *et al.*, 2013). This exciting capability has the potential to eliminate the library preparation requirements of current structure probing methods. Indeed, it seems inevitable that the next era of RNA structure probing studies will use single molecule direct sequencing of chemically modified RNAs to combine the best features of chemical probing and mutational profiling, detecting numerous modifications of single RNA molecules read across their entire lengths.

Acknowledgements

This work is supported by Singapore MOE AcRF T2 grant R154-000-A24-112 to Greg Tucker-Kellogg.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Biomolecular Structures: Prediction, Identification and Analyses. Characterizing and Functional Assignment of Noncoding RNAs. Computational Design and Experimental Implementation of Synthetic Riboswitches and Riboregulators. Engineering of Supramolecular RNA Structures. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Predicting RNA-RNA Interactions in Three-Dimensional Structures. Prediction of Coding and Non-Coding RNA. Whole Genome Sequencing Analysis

References

- Aviran, S., Trapnell, C., Lucks, J.B., *et al.*, 2011. Modeling and automation of sequencing-based characterization of RNA structure. *Proceedings of the National Academy of Sciences of the United States of America* 108, 11069–11074. Available at: <https://doi.org/10.1073/pnas.1106541108>.
- Aw, J.G., Shen, Y., Wilm, A., *et al.*, 2016. In vivo mapping of eukaryotic RNA interactomes reveals principles of higher-order organization and regulation. *Molecular Cell* 62, 603–617. Available at: <https://doi.org/10.1016/j.molcel.2016.04.028>.
- Berkowitz, N.D., Silverman, I.M., Childress, D.M., *et al.*, 2016. A comprehensive database of high-throughput sequencing-based RNA secondary structure probing data (Structure Surfer). *BMC Bioinformatics* 17, 1–9. Available at: <https://doi.org/10.1186/s12859-016-1071-0>.
- Bernat, V., Disney, M.D., 2015. RNA structures as mediators of neurological diseases and as drug targets. *Neuron* 87, 28–46. Available at: <https://doi.org/10.1016/j.neuron.2015.06.012>.
- Brow, D.A., Noller, H.F., 1983. Protection of ribosomal RNA from kethoxal in polyribosomes. Implication of specific sites in ribosome function. *Journal of Molecular Biology* 163, 27–46. Available at: [https://doi.org/10.1016/0022-2836\(83\)90028-1](https://doi.org/10.1016/0022-2836(83)90028-1).
- Busan, S., Weeks, K.M., 2018. Accurate detection of chemical modifications in RNA by mutational profiling (MaP) with ShapeMapper 2. *RNA* 24, 143–148. Available at: <https://doi.org/10.1261/rna.061945.117>.
- Cech, T.R., Steitz, J.A., 2014. The noncoding RNA revolution – Trashing old rules to forge new ones. *Cell* 157, 77–94. Available at: <https://doi.org/10.1016/j.cell.2014.03.008>.
- Cheong, C., Varani, G., Tinoco, I., 1990. Solution structure of an unusually stable RNA hairpin, 5'GGAC(UUCG)GUCC. *Nature* 346, 680–682. Available at: <https://doi.org/10.1038/346680a0>.
- Cruz, J.A., Westhof, E., 2009. The dynamic landscapes of RNA architecture. *Cell* 136, 604–609. Available at: <https://doi.org/10.1016/j.cell.2009.02.003>.
- Daou-Chabo, R., Condon, C., 2009. RNase J1 endonuclease activity as a probe of RNA secondary structure. *RNA* 15, 1417–1425. Available at: <https://doi.org/10.1261/rna.1574309>.
- Deigan, K.E., Li, T.W., Mathews, D.H., Weeks, K.M., 2008. Accurate SHAPE-directed RNA structure determination. *Proceedings of the National Academy of Sciences of the United States of America* 2008, 97–102. Available at: <https://doi.org/10.1073/pnas.0806929106>.
- Devi, G., Zhou, Y., Zhong, Z., Toh, D.F.K., Chen, G., 2015. RNA triplexes: From structural principles to biological and biotech applications. *Wiley Interdisciplinary Reviews: RNA* 6, 111–128. Available at: <https://doi.org/10.1002/wrna.1261>.
- Ding, Y., Kwok, C.K., Tang, Y., Bevilacqua, P.C., Assmann, S.M., 2015. Genome-wide profiling of in vivo RNA structure at single-nucleotide resolution using structure-seq. *Nature Protocols* 10, 1050–1066 <https://doi.org/10.1038/nprot.2015.064>.
- Ding, Y., Tang, Y., Kwok, C.K., Zhang, Y., Bevilacqua, P.C., Assmann, S.M., 2014. In vivo genome-wide profiling of RNA secondary structure reveals novel regulatory features. *Nature* 505, 696–700 <https://doi.org/10.1038/nature12756>.
- Feigon, J., 2015. Back to the future of RNA structure. *RNA* 21, 611–612. Available at: <https://doi.org/10.1261/rna.050716.115>.
- Flynn, R.A., Zhang, Q.C., Spitale, R.C., *et al.*, 2016. Transcriptome-wide interrogation of RNA secondary structure in living cells with icSHAPE. *Nature Protocols* 11, 273–290. Available at: <https://doi.org/10.1038/nprot.2016.011>.
- Garalde, D.R., Snell, E.A., Jachimowicz, D., *et al.*, 2018. Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods* 15, 201–206. Available at: <https://doi.org/10.1038/nmeth.4577>.
- Hajdin, C.E., Bellaousov, S., Huggins, W., *et al.*, 2013. Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots. *Proceedings of the National Academy of Sciences of the United States of America* 110, 5498–5503. Available at: <https://doi.org/10.1073/pnas.1219988110>.
- Heinicke, L.A., Bevilacqua, P.C., 2012. Activation of PKR by RNA misfolding: HDV ribozyme dimers activate PKR. *RNA* 18, 2157–2165. Available at: <https://doi.org/10.1261/rna.034744.112>.
- Ho, N.W., Gilham, P.T., 1971. Reaction of pseudouridine and inosine with N-cyclohexyl-N'-beta-(4-methylmorpholinium)ethylcarbodiimide. *Biochemistry* 10, 3651–3657.
- Homan, P.J., Favorov, O.V., Lavender, C.A., *et al.*, 2014. Single-molecule correlated chemical probing of RNA. *Proceedings of the National Academy of Sciences of the United States of America* 111, 13858–13863. Available at: <https://doi.org/10.1073/pnas.1407306111>.
- Incarinato, D., Neri, F., Anselmi, F., Oliviero, S., 2014. Genome-wide profiling of mouse RNA secondary structures reveals key features of the mammalian transcriptome. *Genome Biology* 15, 491 <https://doi.org/10.1186/s13059-014-0491-2>.

- Incarnato, D., Neri, F., Anselmi, F., Oliviero, S., 2015. RNA structure framework: Automated transcriptome-wide reconstruction of RNA secondary structures from high-throughput structure probing data. *Bioinformatics* 32, 459–461. <https://doi.org/10.1093/bioinformatics/btv571>
- Ishiguro, T., Sato, N., Ueyama, M., *et al.*, 2017. Regulatory role of RNA chaperone TDP-43 for RNA misfolding and repeat-associated translation in SCA31. *Neuron* 94, 108–124.e7. Available at: <https://doi.org/10.1016/j.neuron.2017.02.046>.
- Jain, M., Olsen, H.E., Turner, D.J., *et al.*, 2018. Linear assembly of a human centromere on the Y chromosome. *Nature Biotechnology* 36, 321–323. Available at: <https://doi.org/10.1038/nbt.4109>.
- Kertesz, M., Wan, Y., Mazor, E., *et al.*, 2010. Genome-wide measurement of RNA secondary structure in yeast. *Nature* 467, 103–107. Available at: <https://doi.org/10.1038/nature09322>.
- Khalil, A.M., Guttman, M., Huarte, M., *et al.*, 2009. Many human large intergenic noncoding RNAs associate with chromatin-modifying complexes and affect gene expression. *Proceedings of the National Academy of Sciences of the United States of America* 106, 11667–11672. Available at: <https://doi.org/10.1073/pnas.0904715106>.
- Kim, S.H., Suddath, F.L., Quigley, G.J., *et al.*, 1974. Three-dimensional tertiary structure of yeast phenylalanine transfer RNA. *Science* 185, 435–440. Available at: <https://doi.org/10.1126/science.185.4149.435>.
- Kladwang, W., Das, R., 2010. A mutate-and-map strategy for inferring base pairs in structured nucleic acids: Proof of concept on a DNA/RNA helix. *Biochemistry* 49, 7414–7416. Available at: <https://doi.org/10.1021/bi101123g>.
- Kladwang, W., VanLang, C.C., Cordero, P., Das, R., 2011. A two-dimensional mutate-and-map strategy for non-coding RNA structure. *Nature Chemistry* 3, 954–962. Available at: <https://doi.org/10.1038/nchem.1176>.
- Knapp, G., 1989. Enzymatic approaches to probing of RNA secondary and tertiary structure. *Methods in Enzymology* 180, 192–212. Available at: [https://doi.org/10.1016/0076-6879\(89\)80102-8](https://doi.org/10.1016/0076-6879(89)80102-8).
- Kwok, C.K., 2016. Dawn of the *in vivo* RNA structurome and interactome. *Biochemical Society Transactions* 44, 1395–1410. Available at: <https://doi.org/10.1042/BST20160075>.
- Lee, A.J., Crothers, D.M., 1998. The solution structure of an RNA loop-loop complex: The ColE1 inverted loop sequence. *Structure* 6, 993–1005. Available at: [https://doi.org/10.1016/S0969-2126\(98\)00101-4](https://doi.org/10.1016/S0969-2126(98)00101-4).
- Li, F., Zheng, Q., Ryvkin, P., Dragomir, I., Desai, Y., Aiyer, S., Valladares, O., Yang, J., Bambina, S., Sabin, L.R., Murray, J.I., Lamitina, T., Raj, A., Cherry, S., Wang, L.S., Gregory, B.D., 2012. Global Analysis of RNA Secondary Structure in Two Metazoans. *Cell Reports* 1, 69–82. <https://doi.org/10.1016/j.celrep.2011.10.002>
- Lorenz, R., Wolfinger, M.T., Tanzer, A., Hofacker, I.L., 2016. Predicting RNA secondary structures from sequence and probing data. *Methods* 103, 86–98. Available at: <https://doi.org/10.1016/j.ymeth.2016.04.004>.
- Loughrey, D., Watters, K.E., Settle, A.H., Lucks, J.B., 2014. SHAPE-Seq 2.0: Systematic optimization and extension of high-throughput chemical probing of RNA secondary structure with next generation sequencing. *Nucleic Acids Research* 42, e165–e165. <https://doi.org/10.1093/nar/gku909>
- Low, J.T., Weeks, K.M., 2010. SHAPE-directed RNA secondary structure prediction. *Methods* 52, 150–158. Available at: <https://doi.org/10.1016/j.ymeth.2010.06.007>.
- Lu, Z., Zhang, Q.C., Lee, B., *et al.*, 2016. RNA duplex map in living cells reveals higher-order transcriptome structure. *Cell* 165, 1267–1279. Available at: <https://doi.org/10.1016/j.cell.2016.04.028>.
- Lucks, J.B., Mortimer, S.A., Trapnell, C., *et al.*, 2011. Multiplexed RNA structure characterization with selective 2'-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). *Proceedings of the National Academy of Sciences* 108, 11063–11068. Available at: <https://doi.org/10.1073/pnas.1106501108>.
- Luo, D., Condon, C., Grunberg-Manago, M., Putzer, H., 1998. *In vitro* and *in vivo* secondary structure probing of the thrS leader in *Bacillus subtilis*. *Nucleic Acids Research* 26, 5379–5387.
- Mathews, D.H., Disney, M.D., Childs, J.L., *et al.*, 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 101, 7287–7292. Available at: <https://doi.org/10.1073/pnas.0401799101>.
- Mathews, D.H., Turner, D.H., 2006. Prediction of RNA secondary structure by free energy minimization. *Current Opinion in Structural Biology* 16, 270–278. Available at: <https://doi.org/10.1016/j.sbi.2006.05.010>.
- Merino, E.J., Wilkinson, K.A., Coughlan, J.L., Weeks, K.M., 2005. RNA structure analysis at single nucleotide resolution by selective 2'-hydroxyl acylation and primer extension (SHAPE). *Journal of the American Chemical Society* 127, 4223–4231. Available at: <https://doi.org/10.1021/ja043822v>.
- Mitra, S., Shcherbakova, I.V., Altman, R.B., Brenowitz, M., Laederach, A., 2008. High-throughput single-nucleotide structural mapping by capillary automated footprinting analysis. *Nucleic Acids Research* 36, 1–10. Available at: <https://doi.org/10.1093/nar/gkn267>.
- Monfort, A., Di Minin, G., Postlmayr, A., *et al.*, 2015. Identification of Spen as a crucial factor for Xist function through forward genetic screening in haploid embryonic stem cells. *Cell Reports* 12, 554–561. Available at: <https://doi.org/10.1016/j.celrep.2015.06.067>.
- Mustoe, A.M., Brooks, C.L., Al-Hashimi, H.M., 2014. Hierarchy of RNA functional dynamics. *Annual Review of Biochemistry* 83, 441–466. Available at: <https://doi.org/10.1146/annurev-biochem-060713-035524>.
- Názzer, E., Lei, E.P., 2014. Modulation of chromatin modifying complexes by noncoding RNAs in trans. *Current Opinion in Genetics & Development* 25, 68–73. Available at: <https://doi.org/10.1016/j.gde.2013.11.019>.
- Nissen, P., Hansen, J., Ban, N., Moore, P.B., Steitz, T.A., 2000. The structural basis of ribosome activity in peptide bond synthesis. *Science* 289, 920–930.
- Nissen, P., Ippolito, J.A., Ban, N., Moore, P.B., Steitz, T.A., 2001. RNA tertiary interactions in the large ribosomal subunit: The A-minor motif. *Proceedings of the National Academy of Sciences of the United States of America* 98, 4899–4903. Available at: <https://doi.org/10.1073/pnas.081082398>.
- Norris, M., Kwok, C.K., Cheema, J., *et al.*, 2017. FoldAtlas: A repository for genome-wide RNA structure probing data. *Bioinformatics* 33, 306–308. Available at: <https://doi.org/10.1093/bioinformatics/btw611>.
- Peattie, D.A., Gilbert, W., 1980. Chemical probes for higher-order structure in RNA. *Proceedings of the National Academy of Sciences of the United States of America* 77, 4679–4682. Available at: <https://doi.org/10.1073/pnas.77.8.4679>.
- Reiter, N.J., Osterman, A., Torres-Larios, A., *et al.*, 2010. Structure of a bacterial ribonuclease P holoenzyme in complex with tRNA. *Nature* 468, 784–789. Available at: <https://doi.org/10.1038/nature09516>.
- Rice, G.M., Leonard, C.W., Weeks, K.M., 2014. RNA secondary structure modeling at consistent high accuracy using differential SHAPE. *RNA* 20, 846–854. Available at: <https://doi.org/10.1261/rna.043323.113>.
- Rouskin, S., Zubradt, M., Washietl, S., Kellis, M., Weissman, J.S., 2014. Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. *Nature* 505, 701–705. Available at: <https://doi.org/10.1038/nature12894>.
- Sharma, E., Sterne-Weiler, T., O'Hanlon, D., Blencowe, B.J., 2016. Global mapping of human RNA-RNA interactions. *Molecular Cell* 62, 618–626. Available at: <https://doi.org/10.1016/j.molcel.2016.04.030>.
- Siegrfried, N.A., Busan, S., Rice, G.M., Nelson, J.A.E., Weeks, K.M., 2014. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nature Methods* 11, 959–965.
- Smith, M.A., Gesell, T., Stadler, P.F., Mattick, J.S., 2013. Widespread purifying selection on RNA structure in mammals. *Nucleic Acids Research* 41, 8220–8236. Available at: <https://doi.org/10.1093/nar/gkt596>.
- Smola, M.J., Rice, G.M., Busan, S., Siegrfried, N.A., Weeks, K.M., 2015. Selective 2'-hydroxyl acylation analyzed by primer extension and mutational profiling (SHAPE-MaP) for direct, versatile and accurate RNA structure analysis. *Nature Protocols* 10, 1643–1669. Available at: <https://doi.org/10.1038/nprot.2015.103>.
- Spitale, R.C., Crisalli, P., Flynn, R.A., *et al.*, 2013. RNA SHAPE analysis in living cells. *Nature Chemical Biology* 9, 18–20. Available at: <https://doi.org/10.1038/nchembio.1131>.
- Spitale, R.C., Flynn, R.A., Zhang, Q.C., *et al.*, 2015. Structural imprints in vivo decode RNA regulatory mechanisms. *Nature* 519, 486–490. Available at: <https://doi.org/10.1038/nature14263>.

- Talkish, J., May, G., Lin, Y., Woolford, J.L., McManus, C.J., 2014. Mod-seq: High-throughput sequencing for chemical probing of RNA structure. *Rna* 20, 713–720 <https://doi.org/10.1261/rna.042218.113>
- Tan, Z., Sharma, G., Mathews, D.H., 2017. Modeling RNA secondary structure with sequence comparison and experimental mapping data. *Biophysical Journal* 113, 330–338. Available at: <https://doi.org/10.1016/j.bpj.2017.06.039>
- Tian, S., Cordero, P., Kladwang, W., Das, R., 2014. High-throughput mutate-map-rescue evaluates SHAPE-directed RNA structure and uncovers excited states. *RNA* 20, 1815–1826. Available at: <https://doi.org/10.1261/rna.044321.114>
- Tijerina, P., Mohr, S., Russell, R., 2007. DMS footprinting of structured rnas and rna-protein complexes. *Nature Protocols* 2, 2608–2623. Available at: <https://doi.org/10.1038/nprot.2007.380>
- Tinoco, I., Bustamante, C., 1999. How RNA folds. *Journal of Molecular Biology* 293, 271–281. Available at: <https://doi.org/10.1006/jmbi.1999.3001>
- Tullius, T.D., Greenbaum, J.A., 2005. Mapping nucleic acid structure by hydroxyl radical cleavage. *Current Opinion in Chemical Biology* 9, 127–134. Available at: <https://doi.org/10.1016/j.cbpa.2005.02.009>
- Underwood, J.G., Uzielov, A.V., Katzman, S., et al., 2010. FragSeq: Transcriptome-wide RNA structure probing using high-throughput sequencing. *Nature Methods* 7, 995–1001. Available at: <https://doi.org/10.1038/nmeth.1529>
- Vasa, S.M., Guex, N., Wilkinson, K.A., Weeks, K.M., Giddings, M.C., 2008. ShapeFinder: A software system for high-throughput quantitative analysis of nucleic acid reactivity information resolved by capillary electrophoresis. *RNA* 14, 1979–1990. Available at: <https://doi.org/10.1261/rna.1166808>
- Vilfan, I.D., Tsai, Y.C., Clark, T.A., et al., 2013. Analysis of RNA base modification and structural rearrangement by single-molecule real-time detection of reverse transcription. *Journal of Nanobiotechnology* 11, 1–11. Available at: <https://doi.org/10.1186/1477-3155-11-8>
- Wan, Y., Qu, K., Ouyang, Z., Chang, H.Y., 2013. Genome-wide mapping of RNA structure using nuclease digestion and high-throughput sequencing. *Nature Protocols* 8, 849–869 <https://doi.org/10.1038/nprot.2013.045>
- Wan, Y., Qu, K., Ouyang, Z., et al., 2012. Genome-wide measurement of RNA folding energies. *Molecular Cell* 48, 169–181. Available at: <https://doi.org/10.1016/j.molcel.2012.08.008>
- Wan, Y., Qu, K., Zhang, Q.C., Flynn, R.A., Manor, O., Ouyang, Z., Zhang, J., Spitale, R.C., Snyder, M.P., Segal, E., Chang, H.Y., 2014. Landscape and variation of RNA secondary structure across the human transcriptome. *Nature* 505, 706–709 <https://doi.org/10.1038/nature12946>
- Watts, J.M., Dang, K.K., Gorelick, R.J., et al., 2009. Architecture and secondary structure of an entire HIV-1 RNA genome. *Nature* 460, 711–716. Available at: <https://doi.org/10.1038/nature08237>
- Weeks, K.M., 2010. Advances in RNA structure analysis by chemical probing. *Current Opinion in Structural Biology* 20, 295–304. Available at: <https://doi.org/10.1016/j.sbi.2010.04.001>
- Westhof, E., 2015. Twenty years of RNA crystallography. *RNA* 21, 486–487. Available at: <https://doi.org/10.1261/rna.049726.115>
- Wilkinson, K.A., Gorelick, R.J., Vasa, S.M., et al., 2008. High-throughput SHAPE analysis reveals structures in HIV-1 genomic RNA strongly conserved across distinct biological states. *PLOS Biology* 6, e96. Available at: <https://doi.org/10.1371/journal.pbio.0060096>
- Wilkinson, K.A., Merino, E.J., Weeks, K.M., 2005. RNA SHAPE chemistry reveals nonhierarchical interactions dominate equilibrium structural transitions in tRNA^{Asp} transcripts. *Journal of the American Chemical Society* 127, 4659–4667. Available at: <https://doi.org/10.1021/ja0436749>
- Zheng, Q., Ryvkin, P., Li, F., et al., 2010. Genome-wide double-stranded RNA sequencing reveals the functional significance of base-paired RNAs in arabidopsis. *PLOS Genetics* 6, e1001141. Available at: <https://doi.org/10.1371/journal.pgen.1001141>
- Ziehler, W.A., Engelke, D.R., 2001. Probing RNA structure with chemical reagents and enzymes. In: *Current Protocols in Nucleic Acid Chemistry*. Hoboken, NJ: John Wiley & Sons, Inc., pp. 6.1.1–6.1.21. Available at: <https://doi.org/10.1002/0471142700.nc0601s00>
- Zubradt, M., Gupta, P., Persad, S., et al., 2016. DMS-MaPseq for genome-wide or targeted RNA structure probing in vivo. *Nature Methods* 14, 75–82. Available at: <https://doi.org/10.1038/nmeth.4057>

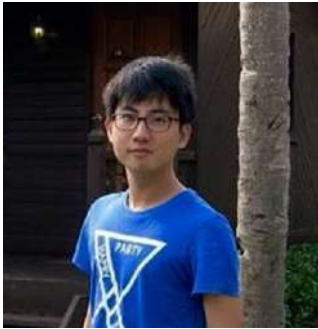
Biographical Sketch



Xiaojing Huo is a postdoctoral research fellow in Prof. Greg Tucker-Kellogg's at the National University of Singapore. She did her PhD training in functional genomics and RNA-seq studies of tumorigenesis models in zebrafish. Her current research interest is on dissection of TGF β signaling in models of human cancers.



Jeremy Ng is a graduate student in computational and molecular biology at the National University of Singapore. His research interest is the regulation of TGF β signaling axis decisions following ligand exposure.



Tan Mingchen is a PhD candidate in computational biology at National University of Singapore. His research a Prize Collecting Steiner system to dissect regulatory networks from heterogeneous genomic data.



Greg Tucker-Kellogg is Professor in Practice in the Department of Biological Sciences and Director of the Faculty of Science Computational Biology Program at the National University of Singapore. He received his PhD in from Yale University under Peter Moore, and was a Jane Coffin Childs Postdoctoral Fellow at Harvard Medical School with Christopher T. Walsh. Prior to joining the NUS faculty, he spent nearly 15 years in the biotech and pharmaceutical industries. His laboratory is developing tools for both optogenetic regulation of cancer signaling pathways and for measurement of single-cell chromatin dynamics.

Assessment of Structure Quality (RNA and Protein)

Nicolas Palopoli, Universidad Nacional de Quilmes & CONICET, Bernal, Buenos Aires, Argentina

© 2019 Elsevier Inc. All rights reserved.

Introduction

Knowledge of the three-dimensional structure of biological molecules has been a permanent goal and often a breakthrough achievement in modern science. Structural (molecular) biology has developed since the late 1950s and now constitutes one of the major forces of scientific progress; for an exciting account on the history of the field by some of its key players, see [Nature Structural and Molecular Biology \(2011\)](#). Among these were [Kendrew *et al.* \(1958\)](#), who first published the structure of myoglobin by X-ray analysis. The application of this technique to determine the tertiary structure of proteins as rigid solids has brought about a revolution in structural biology. Ever since, more than 130,000 protein structures have been resolved at atomic resolution by different techniques and deposited in the publicly available Protein Data Bank (PDB) ([Berman *et al.*, 2000](#)). Nuclear magnetic resonance (NMR) spectroscopy has allowed the determination of dynamic protein structures in solution or in membranes since the late 1970s ([Wüthrich, 2001](#)). Small angle scattering of neutrons (SANS) and X-rays (SAXS) provide means to study the size and shape of proteins ([Mertens and Svergun, 2010](#)). These techniques can be applied jointly with X-ray crystallography ([Tsutakawa *et al.*, 2007](#); [Wang and Wang, 2017](#)) or NMR ([Hennig and Sattler, 2014](#)) to analyze tertiary structures, including those of flexible proteins or large complexes. More recently, cryo-electron microscopy (cryo-EM or simply EM) has been successfully applied to reveal the structures of proteins that could not be determined by other means, although generally at a lower resolution ([Bai *et al.*, 2015](#)). Despite the valuable advances in the field, protein structure determination remains a challenging endeavour, due to high associated costs, experimental constraints like the need for overexpression and purification of the sample and certain properties of some proteins such as highly flexible disordered regions.

The methodologies for high-quality, atomic-resolution RNA structure determination are similar to those applied for proteins ([Felden, 2007](#)). X-ray crystallography and NMR spectroscopy can provide detailed information about RNA in solid phase or in solution, respectively, with cryo-EM still offering lower resolution structures. However, general interest in RNA biology developed later than for proteins and our knowledge of the field is comparatively restricted. To date, the Protein Data Bank ([Berman *et al.*, 2000](#)) lists 1297 RNA-only structures and 2067 RNA-protein complexes, most of them also linked in the Nucleic Acid Database ([Coimbatore Narayanan *et al.*, 2014](#)). These represent barely 2.5% of the total PDB entries as of September 2017, with a steady pace of 30–81 new structures deposited per year in the last two decades. This scarcity must also be related with intrinsic experimental difficulties such as dealing with a vast RNA backbone conformational space ([Murray *et al.*, 2003](#)). Besides the possibility of determining the position of its atoms, low-resolution RNA structure analysis can be performed by other experimental methods such as different *in vivo* and *in vitro* structure-specific chemical and enzymatic probes of base pairing and interactions ([Kubota *et al.*, 2015](#)), or the selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE) ([Weeks and Mauger, 2011](#)) for detecting unconstrained, flexible positions.

Experimental procedures for the determination of protein or RNA three-dimensional structure include a final step of quality control and model refinement. The worldwide Protein Data Bank (wwPDB) has provided in-house tools for deposition and annotation of protein and nucleic acid structures ([Dutta *et al.*, 2009](#)), including the PDB Validation Suite ([Westbrook *et al.*, 2003](#)) for curation and validation of submissions, that recently converged in the unified OneDep system ([Young *et al.*, 2017](#)). The agreement between diffraction data and the atomic coordinates of the final model is typically measured by the R-value criterion of accuracy and by the less biased and more reliable R_{free} cross-validation statistical ([Brünger, 1992](#)). Model-to-data quality of fit, plus the geometric and stereochemical properties of solved structures, can also be evaluated with external, standalone tools such as the PHENIX crystallographic software ([Adams *et al.*, 2010](#)) and PROSESS (Protein Structure Evaluation Suite & Server) ([Berjanskii *et al.*, 2010](#)). Moreover, PROCHECK ([Laskowski *et al.*, 1993](#)), WHAT_CHECK ([Hooft *et al.*, 1996](#)) and MolProbity ([Chen *et al.*, 2010](#)), among others, seek to assess how far the properties of experimental or computational structures are from those observed in known structures. Even after quality checks and improvements, many solved structures are neither complete nor free from errors, and although X-ray crystallography models are generally acceptable, there is still inconsistency in terms of resolution and quality measures in the PDB ([Domagalski *et al.*, 2014](#)).

Regardless of the methodology and despite the many advances in the field, experimental structure determination remains a challenging task for many reasons, including high associated costs in materials and manpower, experimental constraints like sample size needed or the difficulty to obtain good crystals, and intrinsic properties of biomolecules such as large molecular weights and high flexibility. In this context, computational prediction of three-dimensional structures has lately become a readily applicable approach to generate structural data when no further evidence is available. Enough computational power is now accessible and affordable, and modern methods, although still far from perfect, can provide sufficiently reliable and useful structure models.

Several methods exist that can predict one or more likely conformations of the target molecule that might resemble its native conformers. Historically, the most successful methods have been those that try to derive three-dimensional information from close homologues of known structure. This comparative modeling approach is better suited for globular proteins of identifiable folds, where a suitable template should be easy to retrieve. When no homologous structure is available, or when trying to reconstruct

hard targets such as membrane proteins or intrinsically disordered regions, template-free methods can derive hundreds or thousands of structures from first principles, aiming to include a sample of several native-like conformations among them. Similarly, RNA structure prediction can be performed *ab initio* or based on secondary structure elements and long-range tertiary contacts (Laing and Schlick, 2011). RNA molecules can be represented as highly simplified, coarse-grained models that work better for long RNAs, or detailed all-atom simulations most suitable for small RNAs (Laing and Schlick, 2010, 2011). Overall, nevertheless, modeling of RNA tertiary structure still suffers from our relatively limited knowledge of RNA native structures.

The struggle for better structural modeling techniques led to the establishment of several efforts for the large-scale evaluation of protein structure prediction methods. Although now deprecated, LiveBench (Bujnicki *et al.*, 2001) and EVA (Eyrich *et al.*, 2001; Koh *et al.*, 2003) were the first to offer an automated benchmark of several public methods. The Critical Assessment of protein Structure Prediction (CASP) experiments (Moult *et al.*, 2017) are the longest-running and quite possibly most productive of these initiatives. Every two years since 1994 (Moult *et al.*, 1995), the organizers of CASP propose several target proteins which structures remain unknown to the participants but have already been solved experimentally by external collaborators. Developers of manual and automatic programs for protein structure modeling submit their predictions on these targets, for the assessment of the structural similarity between target and model structures and the overall quality of the prediction methods employed. The last published results continue to show a steady progress in modeling performance, with yet much room for improvement (Moult *et al.*, 2016). Although template-free modeling techniques (for the rare cases where no structure with detectable similarity to the target is available) has proven to successfully predict large proteins (Ovchinnikov *et al.*, 2016), this area remains particularly challenging (Kinch *et al.*, 2016). Thus, CASP ROLL (Moult *et al.*, 2014) has been introduced to provide a continuous and equally blind assessment of *de novo* structure prediction. Similarly, and recognizing the constant need for better quality assessment methods, the Continuous Automated Model EvaluatiOn (CAMEO) (Haas *et al.*, 2013) provides an ongoing evaluation of protein structure prediction servers by comparison to novel experimental structures released by the PDB (Berman *et al.*, 2000). A similar effort has been set up more recently by the RNA structure modeling community. Named RNA-Puzzles (Cruz *et al.*, 2012), it provides a blind evaluation of prediction methods on a wide diversity of functional RNA molecules, including large RNA structures (Miao *et al.*, 2015), riboswitches and more (Miao *et al.*, 2017).

This article discusses the Model Quality Assessment Programs, or MQAPs for short, which aim to predict the quality of three-dimensional structure models of biomolecules by applying computational methods that produce global and/or local scores. The field of Model Quality Assessment (MQA) was introduced as a CASP category in CASP7 (2007) (Cozzetto *et al.*, 2007), when participants were assessed on their ability to determine a priori the quality of models built by prediction servers for CASP targets. Since 'quality' is hard to define, and thus cannot be assessed confidently, Estimation of Model Accuracy (EMA) was recently proposed as a more descriptive definition of the task (Kryshtafovych *et al.*, 2016). Although this is a fair concern, the terms MQA and MQAP have been extensively used in the community and are preferred for this work.

Many programs allow scoring and ranking of structural models, or decoys, based on their intrinsic properties and regardless of the availability of an observed structure. Several other protocols can be used for benchmarking models by comparing them to known experimental structures. Some programs propose a mixed approach that relies on the comparison of the model under evaluation with a set of alternative decoys, built for the occasion, to identify conserved structural features. Although these approaches can be collectively referred to as MQAPs, they serve different goals and thus will be treated as separate categories. This article tries to avoid outdated and conflicting resources. For example, different energy functions have been used as single scores for the determination of the top-ranked models and may qualify (perhaps incorrectly) as MQAPs, but they are not covered extensively here. This article is also not intended as a comprehensive catalogue of all available methods for structure quality assessment. Instead, it recalls those methods which have had an impact on the field and highlights other established alternatives that are part of the current state-of-the-art. Also, it is not focused on the algorithms and parameters of the methods, but rather presents their foundations and comments on their applicability.

Background/Fundamentals

The biomolecular modeling community has reached a point of maturity in which the generation of structural models appears straightforward. However, their quality would be largely dependent on the properties of the target molecule, the capabilities of the program employed and the knowledge and experience of the researcher for selecting the right tools and options. Given this complexity, it is hard to ensure in advance the adequacy and utility of models that are generally derived from approximate methods. Most programs rely on finding the most efficient paths towards native-like models, even if these are not optimal. To cope with this limitation, a usual choice is to output a varying number of possible models, expecting that they will provide enough good options. Therefore, regardless of the modeling procedure, a crucial step is the ability to score, rank and select among the many models proposed (Kihara *et al.*, 2009). This is generally attempted by a large-scale computational assessment of structure quality, possibly followed by human evaluation of outstanding candidates. Both protein and RNA structures can be assessed in relationship to a reference structure or by analysing their characteristic structural features. Proteins are organized hierarchically, from local geometric arrangements and bonding patterns that define secondary structures, drive the adoption of conserved tertiary folds and favour interchain contacts to form functional complexes. RNAs are defined by specific base pairing and stacking patterns which result in local structure motifs like bulges, stems or hairpin loops, linked by long-range interactions that determine the tertiary structure. These features, like many others, can be extracted from high-confidence existing structures and processed in different ways to generate a knowledge-based predictor of model quality.

Energy functions have been used for a relative estimation of model accuracy since the first attempts in structural modeling. Early studies on incorrectly folded protein models demonstrated that wrong models can have comparable energy values to acceptable structures, even if their interactions, side-chain packing, or solvent accessibility are distorted (Novotný *et al.*, 1984, 1988). Further developments led to more comprehensive and specific formulations to estimate the total free energy of structural models. For example, MODELLER (Sali and Blundell, 1993) applies internal scoring functions to assess its protein homology models, including the empirical and distant-dependent Discrete Optimized Protein Energy statistical potential (DOPE) (Shen and Sali, 2006). The ROSETTA suite for ab initio structure modeling (Leaver-Fay *et al.*, 2011) combines a statistical potential with a physical energy function for the scoring of ab initio protein and RNA models. Unfortunately, energy-based scores such as these are limited in their approach. Given the enormous diversity and changing nature of biomolecules, it is difficult to determine empirically and model reliably the expected energy of each native structure. An evaluation based in energy properties may oversee other conflicting characteristics of the models; plus, it would not be possible to draw a line between good and bad predictions based on their energies alone. With programs applying different definitions and optimizations of the energy calculation, their scores may not be directly comparable among structures. Therefore, they could only be used for relative rankings of models obtained by individual programs, which may generally not converge to the same answer. To deal with these issues, a whole set of criteria for testing the quality of structure models have matured over the last decades, implemented through many programs specifically designed for the task.

A common approach to determine the quality of structure models has been the evaluation of their stereochemical properties. Both PROCHECK (Laskowski *et al.*, 1993) and WHATCHECK (Hoofit *et al.*, 1996) quantify the deviation in several structural measures between the models and known protein structures. These methods oversee that both native and non-native models can be stereochemically correct. To differentiate among these, several programs incorporate energy functions or statistical potentials derived from the analysis of known structures. Two highly popular approaches have been PROSA II (Sippl, 1993), with a knowledge-based energy function built from inter-residue distance probabilities, and VERIFY3D (Eisenberg *et al.*, 1997), which evaluates the compatibility of the primary and tertiary structures of a model according to a 3D profile of the structural environments observed in real structures. All the previous methods have enjoyed the greatest relevance in the first years of model quality assessment, and although other approaches have proved to be more effective (many of them covered in Section “Model Quality Assessment of Protein Structures”), usage of these methods can still be found in new publications.

Model quality assessment programs can be organized into different categories depending on their general approach to model evaluation. Single-model programs estimate global accuracy by assigning an absolute score to each model based on their geometrical or energetic properties alone. Their performance would depend on a good correlation of the scoring function with actual model quality. Global consensus-based methods (also called clustering methods), while may still calculate absolute scores, would also aim at providing a reliable ranking of a set of models by performing all-against-all comparisons, recognizing the best one(s) and ideally drawing a line between good and bad structures. A mixed type are the quasi-single-model MQAPs. These methods take as input a single model and expands it into an extensive set of decoys, which could be built *ad hoc* with local or external methods. They assess the quality of every model by pairwise comparisons against a subset of high-quality models, taken as reference from the dataset via single-model methods. A special category could be the meta-methods, which combine predictions from different MQAPs into a single quality score. It can be argued that these methods are single-model or consensus-based approaches depending on the nature of the scores they combine; thus, for simplicity, meta-methods are not treated separately in this manuscript. The methods described above may evaluate model accuracy at different levels. Local predictors would seek a per-residue evaluation, while others would provide a unique descriptor of the global quality of the model, either by combining local scores or by analysing the structure as a whole. Also, they can be implemented as automatic, semi-automatic or manual methods, requiring varying levels of human intervention to run the programs and translate raw evaluation results into meaningful information.

Applications

The following subsections present a selection of current relevant options for model quality assessment. The information is organized for protein and RNA structures separately, although some methods can be applied to both types of macromolecules. MQAPs are also classified based on their general approach to assessment, with methods measuring similarity to a reference structure, predicting the quality of single models alone, or by a consensus analysis of multiple decoys. A summary of the classification, characteristics and online availability of these methods is provided in Table 1.

Model Quality Assessment of Protein Structures

Validation against reference structure

Programs to calculate root mean square deviation (RMSD)

RMSD (Nishikawa and Ooi, 1972; Levitt, 1976) is a measure of the average Euclidean distance between equivalent atoms of two biomolecular structures, computed as

$$RMSD = \sqrt{\frac{1}{N} \sum_{i=1}^N d_i^2}$$

Table 1 Classification and availability of model quality assessment programs (MQAPs) described in Section “Applications”

Program	Target	Category	Type	Description	Webserver	Download
MAMMOTH	Proteins	Similarity	Global	Pairwise, rigid-body structure alignment. Similarity assessment by RMSD, p-value. Multiple alignment version available (MAMMOTH-mult).	https://ub.cbm.uam.es/software/online/mammoth.php	https://ub.cbm.uam.es/software/mammoth.php
SuperPose	Proteins	Similarity	Local/global	Pairwise and multiple structure alignment. Similarity assessment by RMSD at residue resolution.	http://wishart.biology.ualberta.ca/SuperPose/	No
ProFit	Proteins	Similarity	Local/global	Pairwise structure alignment. Similarity assessment by RMSD at residue resolution.	http://www.bioinf.org.uk/profit/	http://www.bioinf.org.uk/software/swapreg.html
TopMatch	Proteins	Similarity	Local/global	Pairwise fuzzy/precise structure alignment. Similarity assessment by RMSD and Gaussian function.	https://topmatch.services.came.sbg.ac.at	https://www.came.sbg.ac.at/downloads.php
CE	Proteins	Similarity	Global	Pairwise, rigid-body structure alignment. Similarity assessment by RMSD.	http://source.rcsb.org/	http://source.rcsb.org/download.jsp
FatCat	Proteins	Similarity	Global	Pairwise, flexible and rigid-body structure alignment. Similarity assessment by RMSD.	http://source.rcsb.org/	http://source.rcsb.org/download.jsp
MaxSub	Proteins	Similarity	Local/global	Pairwise, rigid-body structure alignment. Similarity assessment by RMSD and normalized S-score.	https://www.cs.bgu.ac.il/~dfischer/MaxSub/query.html	http://fisherlab.cse.buffalo.edu/maxsub/
TM-score	Proteins, RNA	Similarity	Local/global	Pairwise, rigid-body structure alignment. Similarity assessment by RMSD and TM-score.	https://zhanglab.cmb.med.umich.edu/TM-score/	https://zhanglab.cmb.med.umich.edu/TM-score/
LGA	Proteins	Similarity	Local/global	Pairwise, rigid-body structure alignment. Similarity assessment by RMSD and a combination of Longest Continuous Segments (LCS) and Global Distance Test (GDT) scores.	http://proteinmodel.org/AS2TS/LGA/lga.html	http://proteinmodel.org/AS2TS/download_area/
Dali	Proteins	Similarity	Global	Pairwise, rigid-body structure alignment by search of largest common substructure. Similarity assessment by Z-score.	http://ekhidna2.biocenter.helsinki.fi/dali/	No
local Distance Difference Test (IDDT)	Proteins	Similarity	Local/global	Pairwise and multiple superposition-free structure alignment. Similarity assessment on different distance cutoffs.	https://swissmodel.expasy.org/lddt/	https://swissmodel.expasy.org/lddt/downloads/
CAD-score	Proteins, RNA	Similarity	Local/global	Pairwise structure alignment based on Voronoi tessellations. Similarity assessment on all-atom, main-chain only and side-chain only contacts.	http://bioinformatics.ibt.it/cad-score/	https://bitbucket.org/kilment/cadscore/downloads/
MolProbity	Proteins, RNA	Single-model	Local/global	Allows calculation of steric clashes, incorrect bonds and wrong bond lengths and angles. Other protein-specific validations are available.	http://molprobity.biochem.duke.edu/	https://github.com/rlabduke/MolProbity
ProQ	Proteins	Single-model	Local/global	Neural network-based assessment of structural similarity to hypothetical reference via LG-score and MaxSub scores. Local prediction version developed as ProQres.	https://proq.bioinfo.se/cgi-bin/ProQ/ProQ.cgi	http://proq.bioinfo.se/ProQ/ProQ.html

(Continued)

Table 1 Continued

Program	Target	Category	Type	Description	Webserver	Download
ProQ2	Proteins	Single-model	Local/global	Similar to ProQ, with individual residue weighting based on evolutionary profiles.	http://duffman.it.liu.se/ProQ2/	https://github.com/bjornwallner/ProQ_scripts
ProQ3	Proteins	Single-model	Local/global	Similar to ProQ but using Support Vector Machines.	http://proq3.bioinfo.se/	https://bitbucket.org/ElofssonLab/proq3/downloads/
ProQ3D	Proteins	Single-model	Local/global	Similar to ProQ but using deep neural networks.	http://proq3.bioinfo.se/	https://bitbucket.org/ElofssonLab/proq3/downloads/
MESHI-Score	Proteins	Single-model	Global	Machine learning-based assessment of structural quality aiming to predict real GDT_TS. MESHI-Score is a combination of scores from multiple predictors of structure and energy features.	No	http://meshi1.cs.bgu.ac.il/meshi_score/
SVMQA	Proteins	Single-model	Global	Assessment of structural quality based on Support Vector Machines to predict TM-score and GDT_TS. Uses statistical potential and structural features.	No	http://ee.kias.re.kr/~protein/wiki/doku.php
ModFOLD	Proteins	Single-model (original), quasi-single-model (from v3)	Local/global	Assessment of structural quality based on artificial neural networks. Combines outputs from different single- and quasi-single-model methods.	http://www.reading.ac.uk/bioinf/ModFOLD/	http://www.reading.ac.uk/bioinf/downloads/
QMEAN	Proteins	Single-model	Local/global	Assessment based on combination of statistical potentials with structure descriptors, given as Z-scores. Consensus version QMEANclust is available.	http://swissmodel.expasy.org/qmean	No
Qprob/MULTICOM-NOVEL	Proteins	Single-model	Global	Assessment based on probability density functions to infer errors in predicted structural and physico-chemical properties.	http://calla.rnet.missouri.edu/qprob/	http://calla.rnet.missouri.edu/qprob/
MetaMQAP	Proteins	Single-model	Local/global	Machine learning-based approach to predict deviation of structural features against expected real structure.	https://genesilico.pl/toolkit/unimod?method=MetaMQAPII	No
VoroMQA	Proteins	Single-model	Local/global	Assessment based on statistical potentials considering intra-protein and protein-solvent interatomic areas.	http://bioinformatics.ibt.lt/vtsam/voromqa	https://bitbucket.org/kliment/voromqa/downloads/
Pcons	Proteins	Consensus	Local/global	Neural network approach to predict and rank ProQ score by comparison with decoys obtained from public fold recognition servers.	http://pcons.net	http://pcons.net/index.php?about=download
ModFOLDclust	Proteins	Consensus	Local/global	Consensus-based assessment using ModFOLD scores.	http://www.reading.ac.uk/bioinf/ModFOLD/	http://www.reading.ac.uk/bioinf/downloads/
Pcomb	Proteins	Consensus	Local/global	Score based on linear combination of ProQ2 or ProQ3 and Pcons.	No	https://github.com/ElofssonLab/Pcomb3_CASP12
iFoldRNA	RNA	Similarity	Global	Pairwise, global structure alignment. Similarity assessment by RMSD, p-value.	http://redshift.med.unc.edu/ifoldrna/	http://redshift.med.unc.edu/ifoldrna/

RNAlyzer	RNA	Similarity	Local/global	Pairwise, global/local structure alignment. Similarity assessment by RMSD on different distance cutoffs.	No	http://rnanalyzer.cs.put.poznan.pl/
RNAAssess	RNA	Similarity	Local/global	Based on RNAlyzer but similarity also measured by Deformation Index and Deformation Profile.	http://rnaassess.cs.put.poznan.pl/	No
MCQ4Structures	RNA	Similarity	Local/global	Pairwise and multiple superposition-free structure comparison in torsional angle space.	No	http://www.cs.put.poznan.pl/tzok/wiki/index.php?n=Projects.MCQ4Structures
Rclick	RNA	Similarity	Global	Pairwise structure alignment based on graph theory and least-squares fitting. Similarity assessment by global RMSD, Fragment Score and Topology Score.	http://mspc.bii.a-star.edu.sg/minhn/rclick.html	No
FARFAR	RNA	Consensus	Global	Comparison against dataset of <i>de novo</i> generated decoy sets with heavy-atom RMSD calculation. Standalone version requires Rosetta.	http://rosie.rosettacommons.org/rna_denovo/submit	https://www.rosettacommons.org/docs/latest/application_documentation/rna/rna-denovo
WebRASP	RNA	Consensus	Local/global	Assessment based on energy scores calculated with distance-dependent empirical contact potential. Requires manual comparison of decoy scores.	http://melolab.org/webrasp/home.php	http://melolab.org/webrasp/download.php
3dRNAcore	RNA	Consensus	Local/global	Assessment based on energy scores from contact potential including base-stacking and torsion angle contribution.	http://biophy.hust.edu.cn/3dRNAcore	http://biophy.hust.edu.cn/3dRNAcore.html

where d_i is the Euclidean distance between the i -th pair of atoms, one from each structure, and N is the total number of pairs being considered. RMSD is expressed in units of length, with Ångström (Å, equal to 10^{-10} m) being the most common. Calculation of RMSD requires a good (and fast) or optimal superposition of the structures to minimize the distances (Kabsch, 1976; Coutsias *et al.*, 2004).

RMSD is a simple and yet useful measure that has been so widely adopted for the last three decades that it is pointless trying to enumerate all implementations available. It is currently possible to calculate RMSD using standalone tools (such as MAMMOTH (Ortiz *et al.*, 2002)), web servers (e.g., SuperPose (Maiti *et al.*, 2004)) and bioinformatics software libraries (including BioPython (Cock *et al.*, 2009), BioJava (Holland *et al.*, 2008) and other extensions to popular programming languages). Although RMSD has been widely applied as a measure of global similarity, programs such as ProFit, which implements a special least squares fitting algorithm (McLachlan, 1982), provide local RMSD values to compare specific regions or even individual residues. Some programs, like TopMatch (Sippl and Wiederstein, 2012), would not limit the comparison to RMSD values but extend the output with additional descriptors of structural similarity. Each program may introduce small variations to the calculation to suit different goals. For example, RMSD is often computed from distances between C α atoms only, but C β and other sidechain carbons can also be incorporated. RMSD relies on a superposition of the structures that could treat them as rigid bodies, as in the Combinatorial Extension (CE) algorithm (Shindyalov and Bourne, 1998) or allow some flexibility for more accurate alignments (e.g., in FatCat (Ye and Godzik, 2004)).

Levitt-Gerstein score (LG-score)

LG-score (Levitt and Gerstein, 1998) was designed to tolerate local fluctuations in order to gain sensitivity for the similarity in global topology. It was first developed as a standalone application but, since the early 2000s, it has been adopted and modified for the development of several other similarity measures, as exposed in the following sections.

Superposition of two structures is performed by first calculating all pairwise distances between every atom in each structure, which are then used for an iterative process of least-squares fitting of one structure onto the other. Prioritization of global similarities is achieved by setting an empirically derived distance cutoff (typically 5 Å) to downweigh equivalent residues far from each other, such as those that can be found in flexible local regions. This is directly incorporated into the original LG-score formulation:

$$LG = M \left(\sum \frac{1}{1 + (d_{ij}/d_0)^2} - \frac{N_{gap}}{2} \right)$$

Here $M=20$, d_{ij} is the distance between equivalent residues i and j (the summation is performed over all pairs), d_0 is the distance cutoff (5 Å) and N_{gap} is the total number of internal gaps in the alignment. Thus, the LG-score represents the score of the best alignment found by the algorithm. The LG-score can be transformed into a statistical significance score or P-value, independent of protein size, by comparison to a distribution of scores from random alignments of unrelated proteins.

MaxSub

MaxSub (Siew *et al.*, 2000) identifies the largest subset of C α atoms of a protein structure model that can be superposed correctly on the equivalent atoms of a reference structure. Given the complexity of the process, the maximal superposition needs to be found by an heuristical algorithm. MaxSub uses a sliding window of a certain size (usually 4 residues) to compare equivalent residues in an initial residue-based alignment. In each comparison step, the algorithm adds the residues that are within a distance threshold of each other (3.5 Å by default) to the largest subset found so far, recalculating the superposition of this set and removing those residues that exceed the threshold afterwards. When the algorithm converges to a global alignment, a single, normalized score S is calculated as:

$$S = \left(\sum_i^N \frac{1}{1 + \left(\frac{d_i}{d}\right)^2} \right) / q$$

Here d_i is the distance between the i -th equivalent C α atoms and d is the distance threshold for the comparison; the sum is performed over all residues i in the largest aligned subset and normalized by q , the number of C α in the reference structure. The MaxSub S score ranges from 0 when no superimposable atoms are found to 1 when the structures are identical.

Template Modeling score (TM-score)

TM-score (Zhang and Skolnick, 2004) was designed as a size-independent, normalized and alignment-based variation of the LG-score. After optimal superposition of the model and reference structures, the TM-score is calculated almost identically to the MaxSub S score, but with the threshold d replaced by a d_0 scale factor. Therefore, the distances between all residues aligned to the reference structure are considered (instead of only those with $d_i < d$ as in MaxSub), giving stronger weights to close contacts. TM-scores are within the range (0,1] with higher scores indicating more similarity. It was found empirically that TM-scores < 0.20 correspond to random alignments, while TM-scores > 0.50 indicate a conserved fold.

Local-Global alignment (LGA)

LGA (Zemla, 2003) was designed to provide the best global superposition between two structures while simultaneously finding their regions of local similarity, either with or without the use of an alignment to define equivalent residues. LGA applies

two complementary algorithms, LCS and GDT (see below), to identify continuous segments of superimposable residues and to provide the largest superimposable sets of residues, respectively. Their similarity scores are linearly combined in the scoring function LGA_S, which allows selection and ranking of similar regions and helps in evaluating the total level of structural similarity.

Longest continuous segments (LCS)

LCS Identifies the longest fragment of connected residues within the predicted model for which the RMSD against the target is below certain thresholds (1 Å, 2 Å and 5 Å are generally selected). The use of several RMSD cutoffs allows detailed inspection of local similarities that may go unnoticed under a unique value.

Global distance test (GDT)

Given two structures, GDT finds the largest set of residues in common that fit under a predefined distance threshold. Since the residues do not need to be connected to each other, this is a measure of whole structure similarity. The algorithm starts by obtaining the RMSD and a superposition of all continuous segments identified by LCS, plus all possible segments of three, five or seven residues. Then, it iteratively extends the list of equivalent residues by adding all those that fit under the chosen thresholds. Typical values are in the range of 0.5–10.0 Å, with a step of 0.5 Å. Many variants of the GDT score can be applied. To provide a single-value description of structural similarity, the Global Distance Test Total Score (GDT_TS) (Cristobal *et al.*, 2001) was defined as the average of GDT calculations at 1 Å, 2 Å, 4 Å and 8 Å cutoffs:

$$GDT_{TS} = (GDT_{1\text{\AA}} + GDT_{2\text{\AA}} + GDT_{4\text{\AA}} + GDT_{8\text{\AA}}) / 4\text{\AA}$$

Therefore, GDT_TS represents the average coverage of the reference structure by all common substructures. It has become the *de facto* version of GDT, although a stricter High Accuracy variant (GDT_HA), where the cutoffs are reduced in half, is also frequently observed. It also provides a standard of quality in CASP, where the correlation to the GDT_TS has been used to evaluate model quality assessment methods.

Dali

Dali (Holm and Laakso, 2016) has been a reference method for protein structure comparison for the last two decades (Holm and Sander, 1993). The program and its web server version search for the largest common substructure between two structures (e.g., a model and a reference conformation) to calculate a suboptimal alignment. This is built by comparing the distance matrices of all-versus-all C α contacts in each structure, building up from elementary patterns such as hexapeptides. The significance of the structural similarity between two structures *A* and *B* is measured by a unique quantity, called Z-score, defined as

$$Z = \frac{S(A,B) - \mu(N)}{\sigma(N)}$$

Here, $S(A,B)$ is the sum over all pairs of residues in the common substructure of *A* and *B*, of a custom pairwise similarity score that accounts for the distance between residues (see Supplementary material in Holm and Rosenström (2010) for more details). $\mu(N)$ and $\sigma(N)$ represent the mean and standard deviation of the *S* scores, respectively, averaged for different protein lengths. A structural alignment is considered significantly different from a random alignment when $Z > 2$. Therefore, the Z-score represents a relative measure of structural similarity that, while not directly useful to compare unrelated structures, can be taken as a quantitative indicator of the ‘native-likeness’ of a protein model with respect to a native structure.

local distance difference test (lDDT)

lDDT (Mariani *et al.*, 2013) provides a score of local distance differences between a model and one (or many) reference structures, without the need of superposing the structures in advance. To achieve this, lDDT calculates all pairwise interatomic distances between residues that are found below a certain threshold in each structure. The final lDDT score is the average fraction of conserved interatomic distances at thresholds 0.5 Å, 1 Å, 2 Å and 4 Å. Since the distances are evaluated per atom, both local (residue-based) and global scores can be determined.

CAD-score

CAD-score (Olechnovič *et al.*, 2013) evaluates atomic-level contact area differences (CAD) in physical contacts between a model and a reference structure of a macromolecule, including proteins and nucleic acids. Each of the contact areas, modelled from a Voronoi tessellation of the structure, contain a C α and all points of the protein surface closer to this than to any other C α . The analysis can comprise the whole structure or just an interface. Both local and global scores are provided and allow all pairwise comparisons between all-atom contacts, contacts of atoms in side chains or contacts involving main chain atoms only.

Single-model and quasi-single-model MQAP

MolProbity

MolProbity (Davis *et al.*, 2004) is a web-server and standalone suite of programs for local and global validation of both protein and nucleic acid structures. Its signature feature is the calculation of the number of steric clashes, defined by overlaps ≥ 0.4 Å, and its proportion per thousand residues (the “clash score”) (Word *et al.*, 1999). It can be used to identify weak or missing hydrogen

bonds where expected, unusual van der Waals contacts leading to under- or over-packing and deviations of covalent bond lengths and angles in any 3D structure. It provides additional protein-specific validations like Ramachandran analysis, assessment of peptide bond angles and cis-peptides and identification of unexpected beta-carbon geometries and side-chain rotamers. Some of these scores have been incorporated into the Structure Validation Reports, available for the structures deposited at the worldwide Protein Data Bank websites (Read *et al.*, 2011).

ProQ

ProQ (Wallner and Elofsson, 2003) and several of its variants have been among the top performers in many of the CASP competitions. The original method introduced the ability to predict the quality of a protein structure model, aiming to separate correct models among many incorrect alternatives. ProQ trains a neural network to estimate the values of typical structural similarity measures that would be achieved by a protein model if it were compared with a reference structure. In particular, ProQ informs the expected values of the LG-score and the MaxSub score. Training is based on several structural features that can be calculated from any single protein structure, both at local level (interatomic and residue contacts) and more globally (solvent accessibility, secondary structure, overall shape). Although ProQ provides a global quality score, it was later extended for local prediction with the ProQres method (Wallner and Elofsson, 2006).

ProQ2 (Ray *et al.*, 2012) is based on the same ideas of ProQ, but successfully improved to be the top performing method in different benchmarks (Uziela and Wallner, 2016). In ProQ2, an update and extension of sequence and structure features was combined with individual weighting of all residues based on evolutionary profiles obtained from sequence alignments. This enables a local prediction of quality represented by the S-score, originally by Levitt and Gerstein, but calculated here for each residue i with a distance cutoff $d_i=3 \text{ \AA}$, as follows:

$$S_i = \frac{1}{1 + (d_i/d_0)^2}$$

The local scores for all residues are averaged to yield a global quality measure of correctness in the range [0,1].

The latest versions of ProQ achieve a better assessment of model quality by incorporating methods of machine learning based on deep learning from input parameters. Different approaches have been developed, with ProQ3 (Uziela *et al.*, 2016) using Support Vector Machines (SVMs) and ProQ3D (Uziela *et al.*, 2017) applying a deep neural network. Both ProQ3 and ProQ3D combine the input features used in ProQ2 with two energy potentials taken from the Rosetta ab initio modeling suite (Leaver-Fay *et al.*, 2011): one potential is based on all atoms for greater accuracy, while the other only considers simplified side-chain representations to account for incomplete models.

MESHI-Score

MESHI-Score (Mirzaei *et al.*, 2016) is a machine learning method that aims to predict the real GDT_TS score of a protein model, based on the combination of estimates of numerous structural features made by an extensive number of predictors. ~80,000 decoys from the latest CASP experiments were used to train more than a thousand prediction functions. Each predictor was trained on a non-linear combination of 15 features selected at random from pairwise energy potentials, bond lengths and angles, solvation, radius of gyration, protein frustration, compatibility of predicted and observed secondary structure and solvent accessibility, etc. To predict the quality of a decoy, it is first regularized to correct structure or energy inconsistencies, and their structural features are then fed to the predictors for determination of the individual scores. The weighted median of these scores constitute the MESHI-Score global assessment of quality.

SVMQA

SVMQA (Manavalan and Lee, 2017) is a single-model global quality assessment program based on support vector machines. These have been trained on target domain from previous CASP experiments to predict TM-score and GDT_TS. The prediction is performed through a combination of 8 statistical potential energy terms and an estimation of the compatibility between 11 structural features predicted from sequence and observed in the model. If a set of alternative models is provided, SVMQA can both rank the models and select the best from a given subset.

ModFOLD

ModFOLD (McGuffin, 2007) was originally developed as a single-model quality assessment method. The evaluation was performed with an artificial neural network trained to predict TM-score as a measure of quality, using a combination of outputs from ProQ, the ModCHECK method for assessing sequence-structure compatibility using threading potentials (Pettitt *et al.*, 2005) and the ModSSEA method based on the alignment of secondary structure elements that were predicted in the target and observed in the model (McGuffin, 2007). After further improvements (McGuffin, 2009), ModFOLD3 (Roche *et al.*, 2014) pioneered the development of quasi-single MQAPs for local and global validation, using ModFOLDclust2 to compare the structure of interest with alternative models generated ad hoc. The ModFOLD3 web server has been consistently ranked among the top methods for quality assessment in recent CASP editions (Kryshtafovych *et al.*, 2014; Kryshtafovych *et al.*, 2016).

ModFOLD6 (Maghrabi and McGuffin, 2017) is the latest version of the method. Although it now works under a hybrid approach, it is still among the best performers in CASP. The neural network is now trained and applied with scores from three single-model methods (ProQ2 and the newly developed Secondary Structure Agreement, SSA, and Contact Distance Agreement,

CDA) plus three quasi-single model scoring methods (Disorder B-factor Agreement, DBA, and residue-specific versions of the ModFOLD5 and ModFOLDclust algorithms). For each model, the output is a set of improved per-residue quality scores, together with their mean value as an indicator of global quality. Two variants of ModFOLD6 were presented that differ in the criterium used for global model ranking. ModFOLD6_rank (the default option in the ModFOLD web server) optimizes the selection of models with best quality, even if their predicted scores do not resemble the observed. Instead, ModFOLD6_cor optimizes the correlation between predicted and observed global scores.

Qualitative model energy analysis (QMEAN)

The QMEAN scoring function (Benkert *et al.*, 2008) is used to derive local and global quality assessments for single structure models. It is formulated as a linear combination of four statistical potentials covering the local geometry and interactions within the model, with two terms describing the agreement of predicted and observed secondary structure and solvent accessibility. The global scores are expressed in the range [0,1] (with higher scores for better models). However, the QMEAN web server transforms them into statistically valid Z-scores by comparison with the expectations from a dataset of high-resolution X-ray crystallography structures. Local scores are color-coded within the same range and mapped onto each position of the sequence.

The hybrid version QMEANclust (Benkert *et al.*, 2009) allowed the evaluation of an ensemble of alternative models via an all-against-all comparison of single-model QMEAN scores. QMEANclust calculates QMEAN scores of all models to identify the subset of highest-scoring candidates. Their pairwise distances are calculated as GDT_TS scores and then converted to global QMEANclust scores (the median of a model's GDT_TS to all others). The best models are taken again for the calculation of local consensus scores, but as the selection is now based on their QMEANclust scores, this increases the probability of analysing locally correct conformations.

Qprob/MULTICOM-NOVEL

Qprob (Cao *et al.*, 2016a) is a novel method for quality assessment based on a statistical analysis of estimated errors in the prediction of model structure features. Besides being a standalone tool, Qprob is part of the MULTICOM suite, thus sometimes referred to as MULTICOM-NOVEL (Cao and Cheng, 2016). For each model, Qprob calculates the scores for several descriptors of secondary structure, compactness and solvent accessibility and four different energy scores, all normalized in the range [0,1] when needed. Each of these feature scores is compared with a probability density function (p.d.f.) to predict a feature-exclusive global quality score. This p.d.f. needs to be calculated from a reference set of structures. Here, the same features were extracted from many CASP9 decoy sets and contrasted with the real GDT_TS score against the target structure. Assuming a normal distribution of the differences between predicted and observed scores, their mean and standard deviation are obtained and used for the calculation of the p.d.f. Finally, a weighted combination of the predicted scores for all 11 features provides a single score taken as global estimator of model quality.

A mixed approach (Cao *et al.*, 2014a) was also designed to improve the selection of good models by identifying them with a comprehensive selection of quality assessment methods, combined with a clustering procedure to ensure diversity among the selected models. Decoys were scored by combining 14 single-model quality assessment methods such as ProQ2 and Qprob/MULTICOM-NOVEL, plus clustering methods like Pcons and ModFOLDclust. The average scores from all MQAPs and from six selected methods are used together with a clustering procedure to select the top five and sufficiently diverse models.

MetaMQAP

MetaMQAP (Pawlowski *et al.*, 2008) is a meta-server that predicts the local and global structural deviation of a model against the real (and possibly unknown) structure. It uses a machine learning approach, based on a multivariate linear regression model originally trained on CASP5 and CASP6 datasets, to combine the output of several MQAPs and local structure descriptors. The regression model is applied on selected residues to avoid prediction biases from trivial features like accessibility, hydrophobicity, secondary structure, etc. MetaMQAPII, the latest version of this program, reports a score that predicts the absolute deviation of each C α atom against the equivalent atom in the putative reference structure.

VoroMQA

VoroMQA (Olechnovič and Venclovas, 2017) is an all-atom statistical potential-based MQAP built on the same principles than the CAD-score, but without the need for a reference structure. Observed interactions within the protein and inferred contacts with solvent atoms are described by contact areas instead of the commonly used atom distances. VoroMQA provides a normalized score at the atomic, residue and global levels.

Consensus MQAP

Pcons

Pcons (Lundström *et al.*, 2001; Wallner and Elofsson, 2007) was the first automated consensus method in CASP. Given a structure of interest, a set of alternative models is obtained from several public servers for fold recognition. Models from different servers are collectively compared to identify common structural patterns which would be more likely to be correct. In this regard, it is similar to the once popular but now discontinued 3D-Jury method (Ginalski *et al.*, 2003), but the structural superposition is performed in Pcons using the Levitt-Gerstén algorithm. The models are ranked using a neural network approach that predicts the residue-based S-score of model quality and its global average, as in Section ProQ.

ModFOLDclust

When many models of a target protein are available, ideally built with several alternative modeling methods, ModFOLDclust (McGuffin and Roche, 2010) would probably offer the most accurate local and global quality assessment results among all the ModFOLD-derived methods. The original ModFOLDclust algorithm adapted the 3D-Jury method (Ginalski *et al.*, 2003), just like Pcons, and later incorporated per-residue quality scores. ModFOLDclust2 is the faster and still accurate current version of the method. It combines (by averaging) the scores from ModFOLDclust with the scores provided by ModFOLDclustQ, a variant that uses the Q estimate of structural similarity based on internal residue distances (McGuffin and Roche, 2010). The high speed of the latter algorithm allows clustering of many thousands of alternative models without compromising accuracy. It is now developed as part of ModFOLD.

Pcomb

Pcomb (Larsson *et al.*, 2009) (called Wallner in CASP11) is a consensus approach for local and global quality assessment. The Pcomb score is derived from a linear combination of the scores from a single-model ProQ method (originally ProQ2, now using ProQ3) and the consensus-based predictor (ProQ and Pcons, respectively), optimized with a four-fold higher weight to the former. Pcomb-domain (Elofsson *et al.*, 2017), a slightly modified version that uses an initial definition of domains, was also developed to identify good models of individual domains that are not among the best for the whole structure.

Model Quality Assessment of RNA Structures***Validation against reference structure****Programs to calculate RMSD and other traditional measures in proteins*

Like in protein quality assessment, RNA structure models can be evaluated by comparison to a reference structure taken as gold standard of native-likeness. Structural distance is usually determined by typical scores for protein comparisons, most commonly Root Mean Square Deviation (RMSD), TM-score, GDT_TS and CAD-score. However, as presented in the following subsections, new programs and scoring functions were developed that not only calculate these measures, but also account for RNA-specific features like base pairing constraints. Many model-reference comparisons are alignment-based, and some programs, besides identifying equivalent residues, provide different similarity measures that help to assess the quality of the predicted models (see for example Rclick).

iFoldRNA

Besides its main goal of providing RNA folding simulations, the iFoldRNA web server (Hajdin *et al.*, 2010) can be used to estimate the statistical significance of an RMSD value. This is achieved by computing the RMSD *P*-value against a pre-calculated distribution of RMSDs obtained from conformational space sampling of RNA molecules using replica-exchange discrete molecular dynamics. These RMSD are normally distributed, with a mean dependent on the length of the molecule and whether base pairing constraints were considered or not. *P*-values > 0.01 can be taken as successful predictions in which RNA models are better than those expected by chance.

RNAlyzer

RNAlyzer (Lukasiak *et al.*, 2013) is a standalone application that facilitates a quantitative and visual evaluation of RNA structure models by contrast to a reference structure. For every nucleotide of the reference structure, the program defines the set of atoms in the whole structure that are located inside a sphere of a certain radius and centered on a user-selected atom (either P, C1', O5' or O3'). These are superposed with their equivalent atoms from the model and the RMSD between them is determined. The comparison can be based on all atoms or restricted to a selected type.

RNAlyzer uses increasing sphere radii to simultaneously plot the structural distance between multiple models and the reference structure at different accuracy levels, thus describing the change in prediction quality from local environments to the whole structure. Multiple-model plots allow visual ranking of the predicted models, while detailed individual representations are available for nucleotide-specific inspection of prediction quality. RNAlyzer remains one of the few methods for local evaluation of RNA structure models.

RNAssess

RNAssess (Lukasiak *et al.*, 2015) is a web server built on the RNAlyzer framework by the same authors, adding the convenience of an online resource. Like RNAlyzer, it provides a continuous comparison of predicted and known structures from local to global environments, with multi-model and single-model plots available for a detailed quality assessment. A defining feature of RNAssess is that several metrics can be applied for a tailored comparison against the reference structure. In particular, it provides direct access to Deformation Index and Deformation Profile, two useful similarity metrics designed specifically for RNA structures that have been adopted for the official ranking of models submitted to RNA-Puzzles (Miao *et al.*, 2017).

Deformation index (DI)

The Deformation Index (Parisien *et al.*, 2009) is a normalization of the RMSD by the base Interaction Network Fidelity (INF), which accounts for the accuracy in predicted base-pair and base-stacking interactions:

$$DI = \frac{RMSD}{INF}$$

The complete sets of interactions between nucleotides in each of the structures are compared by set theory operations to build a confusion matrix of True and False predictions. This is used to compute INF as the Matthews Correlation Coefficient (MCC), a measure of quality for binary classifications that equals 1 when the model reproduces the exact base interactions found in the reference (and consequently, $DI = \text{RMSD}$) and approaches 0 (with DI tending to infinite) as the number of shared interactions diminishes.

Deformation profile (DP)

The Deformation Profile (Parisien *et al.*, 2009) consists of a distance matrix of size $N \times M$, where N and M are the lengths of the model and reference structures, respectively. Taken in turns, each pair of equivalent nucleotides is optimally superposed, affecting the relative positioning of all other nucleotides. In each step, the average distances between all corresponding nucleotides are calculated. The resulting RMSD values are stored as one row of the matrix; its average value would be indicative of local similarity around the nucleotide. Similarly, average values of each column would suggest if the distance between the pair of nucleotides is highly dependent on the prediction quality of all other nucleotides. The main diagonal stores the average atomic distance of each nucleotide. Defined this way, DP values can simultaneously suggest similarity at different scales, with the global DP score simply computed as the average distance between a model and the reference structure.

Mean of circular quantities (MCQ4Structures)

MCQ4Structures (Zok *et al.*, 2014) implements the Mean of Circular Quantities (MCQ) algorithm. It was designed to assess similarity based on a reduced trigonometric depiction of RNA structure where every nucleotide is only represented by eight torsion angles. MCQ shows the dissimilarity between two RNA structures, computed from the distances between each pair of equivalent torsion angles (see extended definition and formula in Zok *et al.* (2014)). Thus, MCQ is a distance measure, with greater MCQ values showing more structural differences.

Longest Continuous Segments in Torsion Angle space (LCS-TA) (Wiedemann *et al.*, 2017) is an extension of the MCQ algorithm that allows identification of the longest contiguous segment with significant local similarity between two structures. For any pair of segments compared, a certain MCQ maximum threshold should be defined by the user to address if their sets of torsion angles are similar. LCS-TA informs the MCQ score of the longest segment under this threshold, its location within both structures and its length (called LCS) serving as a measure of local similarity. LCS-TA has been implemented as part of the MCQ4 Structures application by the same group.

Rclick

Rclick (Nguyen and Verma, 2015; Nguyen *et al.*, 2017) extends the CLICK algorithm (Nguyen *et al.*, 2011; Nguyen and Madhusudhan, 2011), designed for optimal protein structure superposition, to the alignment of pairs of RNA structures. It can be used to provide a set of alignment-based statistics representing the RNA model-reference structural similarity. Rclick reduces the structures to sets of three to seven nucleotide residues, each represented by one or more points in space, and in which all possible interconnections are considered. These sets, or cliques (a term borrowed from graph theory), are subjected to a least squares fit to identify and superpose equivalent residues between the structures. Since the connectivity between residues is not relevant for clique matching, Rclick works as a topology-independent alignment program. The output of the Rclick server informs the RMSD and structure overlap (number of equivalent positions within 4 Å) between the optimally aligned structures. These are common structural similarity measures that are also presented by other RNA structure alignment tools such as SETTER (Cech *et al.*, 2012). More importantly, Rclick calculates a Fragment Score, which becomes maximum when all equivalent fragments are contiguous in both sequences, plus a Topology Score that is maximized when the matched fragments have the same directionality.

Single-model and Quasi-single-model MQAP

MolProbity

MolProbity (Davis *et al.*, 2007) provides several automatic analyses of local and global quality for different types of biological structures. It was already covered before for the assessment of protein structure models. Besides general assessments for both proteins and nucleic acids, evaluation of backbone conformations and incorrectly modelled ribose sugar puckers is also available for single RNA structures.

Consensus MQAP

Fragment assembly of RNA with full-atom refinement (FARFAR)

FARFAR (Das *et al.*, 2010) is an extension of the original FARNAL protocol (Das and Baker, 2007) based on the Rosetta methodology for ab initio modeling (Leaver-Fay *et al.*, 2011). It includes a novel and more descriptive force field which allows prediction and assessment of RNA 3D models of multiple sizes, from small motifs to complex folds. FARFAR, like many others, generates sets of decoys that can be clustered according to fold similarity to select groups of native-like models. A clustering procedure was developed to select the largest cluster of lowest-energy structures with less structural divergence among them (Cheng *et al.*, 2015). By setting an appropriate RMSD threshold within each cluster it is possible to control the allowed variability of the modeling procedure and thus estimate its predictive accuracy.

WebRASP

WebRASP (Norambuena *et al.*, 2013) is an online implementation of the standalone Ribonucleic Acids Statistical Potential (RASP), a knowledge-based, distance-dependent pairwise contact potential for RNA structures (Capriotti *et al.*, 2011). The

WebRASP server calculates the total energy score, the normalized energy per contact and the energy profile per site for a single RNA structure. Its output allows visual mapping of energy scores, coded by color gradients, onto the RNA structure. Since the energy scores do not have a direct correlation with prediction quality, it works as a consensus-based MQAP in which all models under scrutiny should be run and later ranked manually for a comparative determination of the best predictions.

3dRNAScore

3dRNAScore (Wang *et al.*, 2015) is an empirical potential for RNA structure evaluation. Its contact potential adds the contribution of RNA base-stacking between adjacent nucleotide residues to total energy calculations. Moreover, it also contemplates the stability effect of backbone torsion dihedral angles, which are directly related with RNA structure flexibility.

The latest version of the 3dRNA web server (Wang *et al.*, 2017) implements 3dRNAScore for a consensus assessment and ranking of its own predicted RNA models. The models are built from their secondary structure elements, selecting from a template library of helices and loop conformations which are later assembled into a complete tertiary structure. Further refinement by simulated annealing is performed, including the use of co-evolutionary information by Direct Coupling Analysis (DCA) (Morcos *et al.*, 2011; Weinreb *et al.*, 2016) to contemplate direct interactions between nucleotides. The final models are clustered by the density-based DBSCAN method (Ester *et al.*, 1996) that groups close structures together. 3dRNAScan is applied to all models in the five most populated clusters to select the best model from each of them.

Analysis and Assessment

Validation against Reference Structure

RMSD has been commonly used to quantify dissimilarity between two or more biomolecular structures, and therefore, it also became a method of choice for validation of protein and RNA models when the reference structure is known. Despite its simplicity and convenience, it is hardly the best option, with many known flaws such as the dependence on protein length and structure resolution (Carugo, 2007). Recognizing these limitations, the community has developed alternative and improved measures (which, unfortunately, still lack the wide adoption of the traditional RMSD, especially outside the community of experts in structural bioinformatics.) Among these is the RMSD₁₀₀, a normalization of the RMSD to reflect the expected value if the structures under comparison were 100 residues (Carugo and Pongor, 2001). In the last years, the more accurate GDT_TS score is becoming commonplace, being used as the default measure to quantify the performance in prediction of CASP participants since its third edition. Like many of the novel approaches, it was designed to find the most conserved substructure. LG-score was probably the first measure of structural similarity specifically proposed to overcome known drawbacks of traditional distances measures like the RMSD. It has had a great impact on the field, being incorporated with modifications into several other popular MQAPs like MaxSub and LGA. TM-score was first proposed as a method to evaluate structural templates for comparative modeling, under the premise that template correctness would be correlated with the measured quality of final full-length models. This role has been surpassed and TM-score is now a standard method to quantify global pairwise structural similarity.

These better methods are not without known flaws. Despite being the official measure in CASP for the last two decades, the uncertainty of GDT_TS calculations has only been estimated very recently (Li *et al.*, 2016). Also, although they are less biased than the RMSD, both the similarity scores GDT and MaxSub still show some dependency on protein length (Zhang and Skolnick, 2004). In fact, MaxSub is more likely to give a good score when the reference structure is short, while the LG-score would tend to be good for comparisons against long structures (Wallner and Elofsson, 2003). (These contrasting behaviours were cleverly combined in the development of the single-model method ProQ, with its different versions being among the best performers in many CASP editions.) A common limitation of all these methods is the need for a subjective distance threshold for selecting pairs of residues to compare, which are hard to justify and customize sensibly.

Many methods for comparison against a reference structure aim to identify the number of equivalent residues and the deviation between the superposed structures. These two scores are hard to optimize simultaneously. The assessment has often been done on C α differences alone, to account for conserved folds, but leaving non-C α out of the comparison overlooks differences in almost 90% of the atoms that constitute a protein model (Keedy *et al.*, 2009). It is expected that in good models, the overall fold similarity would compensate for highly flexible regions. Superposition-based scores like RMSD and GDT may not provide adequate measurements of quality for good models that are only partly correct. A unique global score would tend to average the comparison and miss atomic-level discrepancies, but at the same time, local structural differences would inflate the global dissimilarity score. Given the case of multi-domain proteins, treating the structures as rigid bodies does not allow consideration of changes in the relative orientation of domains, affecting the reliability of global scores. Flexible superpositions can overcome this limitation, but would only be able to provide good local or domain-based scores of long superposed substructures. Applying non-rigid-body scoring schemes would be a sensible option in those cases, and for that reason IDDT, CAD and SphereGrinder (Antczak *et al.*, 2015) have been incorporated as novel reference measures for evaluation of MQAPs from CASP11 (Kryshtafovich *et al.*, 2016). IDDT has also been the default measure in the CAMEO project (Haas *et al.*, 2017), since an automated and continuous quality assessment protocol should be robust under different scenarios. Besides, DALI and CE can provide an alternative to superposition by comparing intra-structure residue-residue distances.

The previous methods are based on comparisons of general properties against a reference structure. They can be tailored to evaluate similarity between pairs of structures of most biomolecules, including RNA models that are usually assessed by RMSD, GDT or CAD-score. RNA-specific methods are designed to contemplate special features of RNA structure that allow a more

meaningful discrimination, even among models showing similar RMSD to a reference structure. For example, the Deformation Index incorporates the evaluation of nucleotide residue interactions commonly observed in RNA structures. Organizers of the RNA-Puzzles experiments recognized the outstanding need to develop new criteria for the comparative assessment of RNA structure models at different levels, from local to global, while considering the structural topology, torsion angles, and bond angles and lengths (Miao *et al.*, 2017). There are methods such as RNAnalyzer that use RMSD to evaluate the global quality of structure models but also offers details on local environments. These could be of great importance for functional RNAs like riboswitches or ribozymes where an accurate modeling of binding or active sites is particularly wanted (Lukasiak *et al.*, 2013).

Single-model and quasi-single-model MQAP

In comparison with reference-based measures, there have been less alternatives for evaluating the quality of a model based on intrinsic properties of the model alone. This is still critical in the assessment of RNA structure models, where MolProbity is one of the few, if not the only available pure method for the analysis of individual structures. Although its clash score evaluation (Chen *et al.*, 2010) is widely used, it is limited to the assessment of atomic distances only. Protein-based single-model methods were also uncommon, with only 5 participants in this category in CASP10 (Kryshtafovych *et al.*, 2014). Perhaps due to this limited development, single-model MQAPs were a step behind consensus MQAPs in ranking a set of alternative models for CASP targets. This changed lately, as 22 single-model methods took part in the latest edition of CASP and brought an evident increase in performance (Kryshtafovych *et al.*, 2017). Single-model methods are now competitive for many tasks, thanks to the use of improved energy functions and recent developments in machine learning approaches (Moult *et al.*, 2017).

ProQ2 was among the top-ranked methods in CASP11 for the identification of the best model in a dataset, and it was also a good choice for identifying correct local features (Kryshtafovych *et al.*, 2016). The performance of ProQ2 was improved by its newest version ProQ3 in CASP12, which saw single-model methods as the top alternatives for many analyses. The best among these surpassed top CASP11 predictors in selecting the best models out of each decoy dataset (Elofsson *et al.*, 2017). ProQ3, together with single-model methods SVMQA and MESHI-score, have been the top performers overall in CASP12 for this task (Kryshtafovych *et al.*, 2017). In fact, GDT-based measures recognized SVMQA as a top method for identifying good-quality models, regardless of the general approach (Manavalan and Lee, 2017). While distinguishing good from bad models is still better with consensus or quasi-single methods, the best among these use single-model approaches to derive their scores (Elofsson *et al.*, 2017). The benefits of single-model methods are not only recognized by the systematic assessment of CASP. For example, MolProbity, which can evaluate the stereochemical properties of both protein and nucleic acids structures, is a convenient single-model method that has been recommended for the evaluation of side-chain rotamers in X-ray structures by the PDB Validation Task Force team (Read *et al.*, 2011).

Single-model approaches are still behind quasi-single-model methods that, if dataset requirements are met, can often match the performance of consensus-based approaches. When many models are available and show a wide range of accuracy values, and they are prone to be partitioned into good and bad models, quasi-single methods can compete with consensus approaches (Kryshtafovych *et al.*, 2014). In CASP12, estimation of the absolute global correctness of a model (measured as the difference against four reference evaluation scores such as GDT_TS) showed the quasi-single-model approach of ModFOLD6 as the best, followed closely by ProQ3 and a single-model version of MULTICOM (Kryshtafovych *et al.*, 2017). In general, if a reliable set of multiple models is not available for consensus approaches, quasi-single model methods like ModFOLD6 could be a good choice (Kryshtafovych *et al.*, 2017).

It should be noted that the choice of accuracy measure has a crucial impact on the comparative evaluation of the different approaches. It was seen in CASP11 that single-model methods could rank even higher if assessed by measures that consider local similarity, probably because many are built on local descriptions of geometries and energies (Kryshtafovych *et al.*, 2016). Indeed, while GDT_TS shows a clear advantage of consensus and quasi-single based methods in model ranking, other measures like LDDT (based on local distance similarity) and CAD (which compares contact areas) highlight that model accuracy correlates best with the scores from ProQ3 and other single-based methods (Elofsson *et al.*, 2017).

Consensus MQAP

Consensus-based MQAPs usually perform better than single-model methods and up until CASP11, the organizers reported best performances from consensus-based approaches in all categories. ModFOLDclust outperformed all other methods in predicting both global and residue-based quality in CASP8 and was also among the best in CASP9 and CASP10 (Kryshtafovych *et al.*, 2014). QMEANclust was best ranked overall in CASP9 (Kryshtafovych *et al.*, 2011), and other consensus-based methods like MULTICOM, QMEANclust and MetaMQAPclust achieved the top eight spots, with statistically equal performances on a per-target analysis. In CASP11, the top performance of the improved single-model method ProQ2 in selecting the best model available was matched by the clustering-based programs Pcomb and Pcons (Kryshtafovych *et al.*, 2016). Consensus-based methods were still better than other approaches in separating correct and incorrect models in CASP12 (Elofsson *et al.*, 2017). Pcomb showed a statistically better performance than most other methods, followed closely by other consensus-based alternatives. In CASP11 and CASP12, local accuracy was generally better predicted by consensus-based approaches, such as Pcons and ModFOLDclust (Kryshtafovych *et al.*, 2016; Elofsson *et al.*, 2017), which were also significantly better than all other methods in assigning per-residue error estimates. Together with Pcomb, they were the top ranked methods for separating good from bad regions of the models (Kryshtafovych *et al.*, 2017).

The scarcity of non-comparative methods for RNA model quality assessment affects the progress of the field and our understanding of its strengths and limitations. Although no comprehensive analysis has been performed, organizers of the RNA-Puzzles experiments highlighted the advantages of applying a set of metrics that evaluate different structural characteristics for a broader assessment of RNA models (Miao *et al.*, 2017). Following this path, it is expected that integrative consensus-based approaches such as the implemented in the 3DRNA web server could provide the best assessments in the near future.

It is clear from the previous paragraphs that consensus-based methods are generally a good choice for model assessment, but they should not be applied lightly. There are certain caveats associated with their design or performance that may lead to the adoption of other methods better suited for the task at hand. Most consensus-based methods are not intended to assess multi-domain protein models, as they would systematically give higher scores to suboptimal models of the complete structure than to better structures of individual domains. This could become a major issue as modeling methods are now more capable of predicting structures in complex, which are particularly difficult for experimental determination.

An independent assessment by MULTICOM authors showed consensus-based approaches could fail when low-quality models dominate the dataset, especially if many of them are similar to each other (Cao *et al.*, 2014a). In such a scenario it is expected to find a high proportion of bad models, generally within a broad distribution of quality scores, and clustering methods would perform badly since most structures would resemble a low-quality consensus (Moult *et al.*, 2017). The performance of consensus methods (and quasi-single methods too) seems to depend on the existence of best models which scores are much better than the rest (Kryshtafovych *et al.*, 2016). This may all explain why single-model method like SVMQA gave a top performance in CASP12, in which many of the protein targets were difficult to predict (Manavalan and Lee, 2017). Limiting the analysis to hard targets from CASP7 and CASP8 also showed their behaviour was relatively on par with that of single-model methods (Cozzetto *et al.*, 2009).

On the practical side, consensus-based methods would become much slower than single-model methods as the number of models in the dataset grows (Cao *et al.*, 2016a). Although just 20 models were found sufficient for the top performance of consensus-based methods in CASP8 (Cozzetto *et al.*, 2009), the end user may need to generate many more models to achieve a wide spread of quality values (Kryshtafovych *et al.*, 2014). Therefore, consensus-based methods may not be the best available option for researchers that need to assess the global or local quality of a single model to allow its functional analysis.

Discussion

The field of quality assessment of protein and RNA structure models has been making steady progress for more than three decades. Some well-performing MQAPs are now unavailable, like the promising methods MQAPsingle (quasi-single) (Pawlowski *et al.*, 2016) and MetaMQAPclust (consensus-based) (Pawlowski and Bujnicki, 2012), and the clustering methods MUFOLD-QA and MUFOLD-WQA (Zhang *et al.*, 2011) that were among the best in CASP9 (Kryshtafovych *et al.*, 2011). Fortunately, and although with varying levels of success, current MQAPs still comprise several useful methodologies and approaches.

Model assessment is usually performed by comparison against a reference. It is important to recall that experimentally solved structures are also scientific models, based on the interpretation of empirical data, and thus they are not necessarily a faithful representation of reality (Mackay *et al.*, 2017). Quality assessment protocols that rely on comparisons with a reference structure should be taken cautiously. As most similarity measures (like GDT_TS or TM-score) may not have a direct relationship with native-likeness, the best models ranked by such measures might not necessarily resemble real structures.

Typical tools for the analysis of individual structures like Verify3D and Prosa, which have long been used for the evaluation of structures obtained from experimental techniques, have now been outperformed by some of the best modern MQAPs like ModFOLD and ProQ, according to certain analyses (Haas *et al.*, 2017). Both single- or multi-model methods could provide the best assessment of structural models depending on the task at hand, although no method is ideal. Many MQAPs have been developed to distinguish between native and non-native structures, but as mentioned above, structural modeling could be such a hard task that even the best models may fail to describe the native state. Other MQAP have been designed to find the best available models even in these complicated scenarios. The CASP experiments demonstrated that regardless of the approach, not even the best MQAPs could consistently perform well for every protein (Kryshtafovych and Fidelis, 2009). While some methods could succeed in picking the best available structures, they may perform poorly in ranking all models in a decoy set, or vice versa (Kryshtafovych *et al.*, 2014).

As a cautionary note, it should be highlighted that scores derived from MQAPs must behave as a true metric to allow a fair and predictable comparison of alternative models. This has often not been respected, as illustrated by a well-established method such as the Z-score provided by Dali (Adams and Naylor, 2003). Its formulation is such that two independent pairs of identical structures could receive different Z-scores, violating a basic premise of a similarity metric.

Future Directions

The field of model quality assessment is heading to improved and more capable methods. Several metrics based on GDT_TS were already proposed to capture differences beyond C α atoms (Keedy *et al.*, 2009) and we could expect others to integrate independent assessments of both local and global similarity. Some of the new MQAP may revisit typical assets in structural analysis like evolutionary information. It has only been incorporated into MQAPs as site-specific residue conservation data extracted from sequence databases. ConQuass (Kalman and Ben-Tal, 2010), now deprecated, tried to determine the correlation between the

observed amino acids at each site with the model structure. We developed the BeEP server (Palopoli *et al.*, 2013), an MQAP that estimates the correspondence between the substitution pattern simulated under the structural constraints imposed by a protein model and its observed evolutionary history. We expect the explicit use of evolutionary information, as implemented in BeEP, as a clear path towards the advancement of existing programs.

Novel methods should be able to handle special cases of biomolecules like membrane proteins or small RNAs. MQAPs are crucial here since computational models of these structures are usually difficult to obtain with confidence. Unfortunately, these models are still hard to assess by common methods which are not parameterized for their unique characteristics. Advances are being made, as in ProQM-resample (Wallner, 2014) and QMEANBrane (Studer *et al.*, 2014) which allow for the specific assessment of transmembrane protein models.

The last years have seen the development of methods that depart from traditional approaches. The current advancement of deep learning techniques promises a substantial improvement in computational methods for biological research. Model quality assessment programs are not exempt from this trend, with new methods like SMOQ (Cao *et al.*, 2014b), DeepQA (Cao *et al.*, 2016b) and QACon (Cao *et al.*, 2017) already matching state-of-the-art performances. Other modern approaches are also performing well and are likely to be further explored soon. Among these are the graph-based methods implemented in GMQ (Shin *et al.*, 2017), and the addition of information from homologous structures to existing quality scores by machine learning, used by the yet unpublished QMEANDisCo method that performed among the best in the latest CAMEO report (Haas *et al.*, 2017).

As the native state is better described by an ensemble of conformers, an important upcoming challenge is to account for the identification of models integrating this ensemble. They could involve from large conformational changes such as the relative rigid-body movements of large domains (Gerstein and Echols, 2004) to tiny rearrangements as derived from rotations of residues to open, close or enlarge tunnels and cavities (Gora *et al.*, 2013). Discrimination of models with such structural differences should require the development of more sophisticated methods in order to provide functional and reliable 3D models (Palopoli *et al.*, 2016; Monzon *et al.*, 2017). Although some efforts on this direction have been taken, such as with the IDDT score (Mariani *et al.*, 2013) that can take an ensemble of equivalent conformers as reference, the community still needs to widely adopt conformational diversity as a central element in quality evaluation.

Closing Remarks

Current state-of-the-art model quality assessment programs can be useful to assess structural models in real-life study cases. Although consensus-based and quasi-single methods used to perform best than pure single-model MQAPs, and they may still be preferred for local and absolute quality estimations, progress in the latter means they could now be equally or even more effective in ranking models and selecting the best ones. There is still no optimal and universal solution, however, and there's a need for novel protocols for model quality assessment that can achieve a high level of discrimination in complicated scenarios. Within this active field, we expect the strengths and weaknesses shared by successful current programs will lead to improved performances of the upcoming methods.

Acknowledgements

The author would like to thank Dr. Gustavo Parisi for his insights on this manuscript, and both him and Dr. Cristina Marino-Buslje for allowing the time to work in it. The author is Researcher of the National Scientific and Technical Research Council (CONICET) in Argentina.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Biomolecular Structures: Prediction, Identification and Analyses. Engineering of Supramolecular RNA Structures. Genome-Wide Probing of RNA Structure. Natural Language Processing Approaches in Bioinformatics. Predicting RNA-RNA Interactions in Three-Dimensional Structures

References

- Adams, D.C., Naylor, G.J.P., 2003. A comparison of methods for assessing the structural similarity of proteins. In: Guerra, C., Istrail, S. (Eds.), *Mathematical Methods for Protein Structure Analysis and Design*, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 109–115.
- Adams, P.D., Afonine, P.V., Bunkóczi, G., *et al.*, 2010. PHENIX: A comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.* 66, 213–221.
- Antczak, P.L.M., Ratajczak, T., Lukasiak, P., Blazewicz, J., 2015. SphereGrinder – Reference structure-based tool for quality assessment of protein structural models. In: *Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, pp. 665–668.
- Bai, X., McMullan, G., Scheres, S.H.W., 2015. How cryo-EM is revolutionizing structural biology. *Trends Biochem. Sci.* 40, 49–57.
- Benkert, P., Schwede, T., Tosatto, S.C., 2009. QMEANclust: Estimation of protein model quality by combining a composite scoring function with structural density information. *BMC Struct. Biol.* 9, 35.
- Benkert, P., Tosatto, S.C.E., Schomburg, D., 2008. QMEAN: A comprehensive scoring function for model quality assessment. *Proteins* 71, 261–277.

- Berjanskii, M., Liang, Y., Zhou, J., *et al.*, 2010. PROSESS: A protein structure evaluation suite and server. *Nucleic Acids Res.* 38, W633–W640.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The protein data bank. *Nucleic Acids Res.* 28, 235–242.
- Brünger, A.T., 1992. Free R value: A novel statistical quantity for assessing the accuracy of crystal structures. *Nature* 355, 472–475.
- Bujnicki, J.M., Elofsson, A., Fischer, D., Rychlewski, L., 2001. LiveBench-1: Continuous benchmarking of protein structure prediction servers. *Protein Sci.* 10, 352–361.
- Cao, R., Adhikari, B., Bhattacharya, D., *et al.*, 2017. QAcon: Single model quality assessment using protein structural and contact information with machine learning techniques. *Bioinformatics* 33, 586–588.
- Cao, R., Bhattacharya, D., Adhikari, B., Li, J., Cheng, J., 2016a. Massive integration of diverse protein quality assessment methods to improve template based modeling in CASP11. *Proteins* 84 (Suppl. 1), S247–S259.
- Cao, R., Bhattacharya, D., Hou, J., Cheng, J., 2016b. DeepQA: Improving the estimation of single protein model quality with deep belief networks. *BMC Bioinform.* 17, 495.
- Cao, R., Cheng, J., 2016. Protein single-model quality assessment by feature-based probability density functions. *Sci. Rep.* 6, 23990.
- Cao, R., Wang, Z., Cheng, J., 2014a. Designing and evaluating the MULTICOM protein local and global model quality prediction methods in the CASP10 experiment. *BMC Struct. Biol.* 14, 13.
- Cao, R., Wang, Z., Wang, Y., Cheng, J., 2014b. SMOQ: A tool for predicting the absolute residue-specific quality of a single protein model with support vector machines. *BMC Bioinform.* 15, 120.
- Capriotti, E., Norambuena, T., Marti-Renom, M.A., Melo, F., 2011. All-atom knowledge-based potential for RNA structure prediction and assessment. *Bioinformatics* 27, 1086–1093.
- Carugo, O., 2007. Statistical validation of the root-mean-square-distance, a measure of protein structural proximity. *Protein Eng. Des. Sel.* 20, 33–37.
- Carugo, O., Pongor, S., 2001. A normalized root-mean-square distance for comparing protein three-dimensional structures. *Protein Sci.* 10, 1470–1473.
- Cech, P., Svozil, D., Hoksza, D., 2012. SETTER: Web server for RNA structure comparison. *Nucleic Acids Res.* 40, W42–W48.
- Chen, V.B., Arendall, W.B., Headd, J.J., *et al.*, 2010. MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D Biol. Crystallogr.* 66, 12–21.
- Cheng, C.Y., Chou, F.-C., Das, R., 2015. Modeling complex RNA tertiary folds with Rosetta. *Methods Enzymol.* 553, 35–64.
- Cock, P.J.A., Antao, T., Chang, J.T., *et al.*, 2009. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25, 1422–1423.
- Coimbatore Narayanan, B., Westbrook, J., Ghosh, S., *et al.*, 2014. The nucleic acid database: New features and capabilities. *Nucleic Acids Res.* 42, D114–D122.
- Coutsias, E.A., Seok, C., Dill, K.A., 2004. Using quaternions to calculate RMSD. *J. Comput. Chem.* 25, 1849–1857.
- Cozzetto, D., Kryshchuk, A., Ceriani, M., Tramontano, A., 2007. Assessment of predictions in the model quality assessment category. *Proteins* 69 (Suppl. 8), S175–S183.
- Cozzetto, D., Kryshchuk, A., Tramontano, A., 2009. Evaluation of CASP8 model quality predictions. *Proteins* 77 (Suppl. 9), S157–S166.
- Cristobal, S., Zemla, A., Fischer, D., Rychlewski, L., Elofsson, A., 2001. A study of quality measures for protein threading models. *BMC Bioinform.* 2, 5.
- Cruz, J.A., Blanchet, M.-F., Boniecki, M., *et al.*, 2012. RNA-Puzzles: A CASP-like evaluation of RNA three-dimensional structure prediction. *RNA* 18, 610–625.
- Das, R., Baker, D., 2007. Automated de novo prediction of native-like RNA tertiary structures. *Proc. Natl. Acad. Sci. USA* 104, 14664–14669.
- Das, R., Karanikolas, J., Baker, D., 2010. Atomic accuracy in predicting and designing noncanonical RNA structure. *Nat. Methods* 7, 291–294.
- Davis, I.W., Leaver-Fay, A., Chen, V.B., *et al.*, 2007. MolProbity: All-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Res.* 35, W375–W383.
- Davis, I.W., Murray, L.W., Richardson, J.S., Richardson, D.C., 2004. MOLPROBITY: Structure validation and all-atom contact analysis for nucleic acids and their complexes. *Nucleic Acids Res.* 32, W615–W619.
- Domagalski, M.J., Zheng, H., Zimmerman, M.D., *et al.*, 2014. The quality and validation of structures from structural genomics. *Methods Mol. Biol.* 1091, 297–314.
- Dutta, S., Burkhardt, K., Young, J., *et al.*, 2009. Data deposition and annotation at the worldwide protein data bank. *Mol. Biotechnol.* 42, 1–13.
- Eisenberg, D., Lüthy, R., Bowie, J.U., 1997. VERIFY3D: Assessment of protein models with three-dimensional profiles. In: Carter Jr., C., Sweet, R. (Eds.), *Macromolecular Crystallography Part B, Methods in Enzymology*. Elsevier, pp. 396–404.
- Elofsson, A., Joo, K., Keasar, C., *et al.*, 2017. Methods for estimation of model accuracy in CASP12. *Proteins* 86, 361–373.
- Ester, M., Kriegel, H.-P., Sander, J., *et al.*, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 226–231.
- Eyrich, V.A., Marti-Renom, M.A., Przybylski, D., *et al.*, 2001. EVA: Continuous automatic evaluation of protein structure prediction servers. *Bioinformatics* 17, 1242–1243.
- Felden, B., 2007. RNA structure: Experimental analysis. *Curr. Opin. Microbiol.* 10, 286–291.
- Gerstein, M., Echols, N., 2004. Exploring the range of protein flexibility, from a structural proteomics perspective. *Curr. Opin. Chem. Biol.* 8, 14–19.
- Ginalski, K., Elofsson, A., Fischer, D., Rychlewski, L., 2003. 3D-Jury: A simple approach to improve protein structure predictions. *Bioinformatics* 19, 1015–1018.
- Gora, A., Brezovsky, J., Damborsky, J., 2013. Gates of enzymes. *Chem. Rev.* 113, 5871–5923.
- Haas, J., Barbato, A., Behringer, D., *et al.*, 2017. Continuous automated model evaluation (CAMEO) complementing the critical assessment of structure prediction in CASP12. *Proteins* 86, 387–398.
- Haas, J., Roth, S., Arnold, K., *et al.*, 2013. The protein model portal – A comprehensive resource for protein structure and model information. *Database* 2013, bat031.
- Hajdin, C.E., Ding, F., Dokholyan, N.V., Weeks, K.M., 2010. On the significance of an RNA tertiary structure prediction. *RNA* 16, 1340–1349.
- Hennig, J., Sattler, M., 2014. The dynamic duo: Combining NMR and small angle scattering in structural biology. *Protein Sci.* 23, 669–682.
- Holland, R.C.G., Down, T.A., Pocock, M., *et al.*, 2008. BioJava: An open-source framework for bioinformatics. *Bioinformatics* 24, 2096–2097.
- Holm, L., Laakso, L.M., 2016. Dali server update. *Nucleic Acids Res.* 44, W351–W355.
- Holm, L., Rosenström, P., 2010. Dali server: Conservation mapping in 3D. *Nucleic Acids Res.* 38, W545–W549.
- Holm, L., Sander, C., 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* 233, 123–138.
- Hoof, R.W., Vriend, G., Sander, C., Abola, E.E., 1996. Errors in protein structures. *Nature* 381, 272.
- Kabsch, W., 1976. A solution for the best rotation to relate two sets of vectors. *Acta Cryst. A* 32, 922–923.
- Kalman, M., Ben-Tal, N., 2010. Quality assessment of protein model-structures using evolutionary conservation. *Bioinformatics* 26, 1299–1307.
- Keedy, D.A., Williams, C.J., Headd, J.J., *et al.*, 2009. The other 90% of the protein: Assessment beyond the Calphas for CASP8 template-based and high-accuracy models. *Proteins* 77 (Suppl. 9), S29–S49.
- Kendrew, J., Bodo, G., Dintzis, H., *et al.*, 1958. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature* 181, 662–666.
- Kihara, D., Chen, H., Yang, Y.D., 2009. Quality assessment of protein structure models. *Curr. Protein Pept. Sci.* 10, 216–228.
- Kinch, L.N., Li, W., Monastyrsky, B., Kryshchuk, A., Grishin, N.V., 2016. Evaluation of free modeling targets in CASP11 and ROLL. *Proteins* 84 (Suppl. 1), S51–S66.
- Koh, I.Y., Eyrich, V.A., Marti-Renom, M.A., *et al.*, 2003. EVA: Evaluation of protein structure prediction servers. *Nucleic Acids Res.* 31, 3311–3315.
- Kryshchuk, A., Barbato, A., Fidelis, K., *et al.*, 2014. Assessment of the assessment: Evaluation of the model quality estimates in CASP10. *Proteins* 82 (Suppl. 2), S112–S126.
- Kryshchuk, A., Barbato, A., Monastyrsky, B., *et al.*, 2016. Methods of model accuracy estimation can help selecting the best models from decoy sets: Assessment of model accuracy estimations in CASP11. *Proteins* 84 (Suppl. 1), S349–S369.
- Kryshchuk, A., Fidelis, K., 2009. Protein structure prediction and model quality assessment. *Drug Discov. Today* 14, 386–393.
- Kryshchuk, A., Fidelis, K., Tramontano, A., 2011. Evaluation of model quality predictions in CASP9. *Proteins* 79 (Suppl. 10), S91–S106.
- Kryshchuk, A., Monastyrsky, B., Fidelis, K., Schwede, T., Tramontano, A., 2017. Assessment of model accuracy estimations in CASP12. *Proteins* 86, 345–360.

- Kubota, M., Tran, C., Spitale, R.C., 2015. Progress and challenges for chemical probing of RNA structure inside living cells. *Nat. Chem. Biol.* 11, 933–941.
- Laing, C., Schlick, T., 2010. Computational approaches to 3D modeling of RNA. *J. Phys. Condens. Matter* 22, 283101.
- Laing, C., Schlick, T., 2011. Computational approaches to RNA structure prediction, analysis, and design. *Curr. Opin. Struct. Biol.* 21, 306–318.
- Larsson, P., Skwark, M.J., Wallner, B., Elofsson, A., 2009. Assessment of global and local model quality in CASP8 using Pcons and ProQ. *Proteins* 77 (Suppl. 9), S167–S172.
- Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M., 1993. PROCHECK: A program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.* 26, 283–291.
- Leaver-Fay, A., Tyka, M., Lewis, S.M., *et al.*, 2011. ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* 487, 545–574.
- Levitt, M., 1976. A simplified representation of protein conformations for rapid simulation of protein folding. *J. Mol. Biol.* 104, 59–107.
- Levitt, M., Gerstein, M., 1998. A unified statistical framework for sequence comparison and structure comparison. *Proc. Natl. Acad. Sci. USA* 95, 5913–5920.
- Li, W., Schaeffer, R.D., Otwinowski, Z., Grishin, N.V., 2016. Estimation of uncertainties in the global distance test (GDT_TS) for CASP models. *PLOS ONE* 11, e0154786.
- Lukasiak, P., Antczak, M., Ratajczak, T., *et al.*, 2013. RNalyzer – Novel approach for quality analysis of RNA structural models. *Nucleic Acids Res.* 41, 5978–5990.
- Lukasiak, P., Antczak, M., Ratajczak, T., *et al.*, 2015. RNAssess – A web server for quality assessment of RNA 3D structures. *Nucleic Acids Res.* 43, W502–W506.
- Lundström, J., Rychlewski, L., Bujnicki, J., Elofsson, A., 2001. Pcons: A neural-network-based consensus predictor that improves fold recognition. *Protein Sci.* 10, 2354–2362.
- Mackay, J.P., Landsberg, M.J., Whitten, A.E., Bond, C.S., 2017. Whaddaya know: A guide to uncertainty and subjectivity in structural biology. *Trends Biochem. Sci.* 42, 155–167.
- Maghrabi, A.H.A., McGuffin, L.J., 2017. ModFOLD6: An accurate web server for the global and local quality estimation of 3D protein models. *Nucleic Acids Res.* 45.
- Maiti, R., Van Domselaar, G.H., Zhang, H., Wishart, D.S., 2004. SuperPose: A simple server for sophisticated structural superposition. *Nucleic Acids Res.* 32, W590–W594.
- Manavalan, B., Lee, J., 2017. SVMQA: Support-vector-machine-based protein single-model quality assessment. *Bioinformatics* 33, 2496–2503.
- Mariani, V., Biasini, M., Barbato, A., Schwede, T., 2013. IDDT: A local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* 29, 2722–2728.
- McGuffin, L.J., 2007. Benchmarking consensus model quality assessment for protein fold recognition. *BMC Bioinform.* 8, 345.
- McGuffin, L.J., 2009. Prediction of global and local model quality in CASP8 using the ModFOLD server. *Proteins* 77 (Suppl. 9), S185–S190.
- McGuffin, L.J., Roche, D.B., 2010. Rapid model quality assessment for protein structure predictions using the comparison of multiple models without structural alignments. *Bioinformatics* 26, 182–188.
- McLachlan, A.D., 1982. Rapid comparison of protein structures. *Acta Cryst. A* 38, 871–873.
- Mertens, H.D.T., Svergun, D.I., 2010. Structural characterization of proteins and complexes using small-angle X-ray solution scattering. *J. Struct. Biol.* 172, 128–141.
- Miao, Z., Adamiak, R.W., Antczak, M., *et al.*, 2017. RNA-puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA* 23, 655–672.
- Miao, Z., Adamiak, R.W., Blanchet, M.-F., *et al.*, 2015. RNA-puzzles Round II: Assessment of RNA structure prediction programs applied to three large RNA structures. *RNA* 21, 1066–1084.
- Mirzaei, S., Sidi, T., Keasar, C., Crivelli, S., 2016. Purely structural protein scoring functions using support vector machine and ensemble learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.*
- Monzon, A.M., Zea, D.J., Marino-Buslje, C., Parisi, G., 2017. Homology modeling in a dynamical world. *Protein Sci.* 26, 2195–2206.
- Morcos, F., Pagnani, A., Lunt, B., *et al.*, 2011. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc. Natl. Acad. Sci. USA* 108, E1293–E1301.
- Moult, J., Fidelis, K., Krysztofowicz, A., Schwede, T., Tramontano, A., 2014. Critical assessment of methods of protein structure prediction (CASP) – Round X. *Proteins* 82 (Suppl. 2), S1–S6.
- Moult, J., Fidelis, K., Krysztofowicz, A., Schwede, T., Tramontano, A., 2016. Critical assessment of methods of protein structure prediction: Progress and new directions in Round XI. *Proteins* 84 (Suppl. 1), S4–S14.
- Moult, J., Fidelis, K., Krysztofowicz, A., Schwede, T., Tramontano, A., 2017. Critical assessment of methods of protein structure prediction (CASP)–Round XII. *Proteins* 86, 7–15.
- Moult, J., Pedersen, J.T., Judson, R., Fidelis, K., 1995. A large-scale experiment to assess protein structure prediction methods. *Proteins* 23, ii–v.
- Murray, L.J.W., Arendall, W.B., Richardson, D.C., Richardson, J.S., 2003. RNA backbone is rotameric. *Proc. Natl. Acad. Sci. USA* 100, 13904–13909.
2011. Celebrating structural biology. *Nat. Struct. Mol. Biol.* 18, 1304–1316.
- Nguyen, M.N., Madhusudhan, M.S., 2011. Biological insights from topology independent comparison of protein 3D structures. *Nucleic Acids Res.* 39, e94.
- Nguyen, M.N., Sim, A.Y.L., Wan, Y., Madhusudhan, M.S., Verma, C., 2017. Topology independent comparison of RNA 3D structures using the CLICK algorithm. *Nucleic Acids Res.* 45, e5.
- Nguyen, M.N., Tan, K.P., Madhusudhan, M.S., 2011. CLICK – Topology-independent comparison of biomolecular 3D structures. *Nucleic Acids Res.* 39, W24–W28.
- Nguyen, M.N., Verma, C., 2015. Rclick: A web server for comparison of RNA 3D structures. *Bioinformatics* 31, 966–968.
- Nishikawa, K., Ooi, T., 1972. Tertiary structure of proteins. II. Freedom of dihedral angles and energy calculation. *J. Phys. Soc. Jpn.* 32, 1338–1347.
- Norabuena, T., Cares, J.F., Capriotti, E., Melo, F., 2013. WebRASP: A server for computing energy scores to assess the accuracy and stability of RNA 3D structures. *Bioinformatics* 29, 2649–2650.
- Novotný, J., Brucoleri, R., Karplus, M., 1984. An analysis of incorrectly folded protein models. *J. Mol. Biol.* 177, 787–818.
- Novotný, J., Rashin, A.A., Brucoleri, R.E., 1988. Criteria that discriminate between native proteins and incorrectly folded models. *Proteins* 4, 19–30.
- Olechnovič, K., Kulberkytė, E., Venclovas, C., 2013. CAD-score: A new contact area difference-based function for evaluation of protein structural models. *Proteins* 81, 149–162.
- Olechnovič, K., Venclovas, C., 2017. VoroMQA: Assessment of protein structure quality using interatomic contact areas. *Proteins* 85, 1131–1145.
- Ortiz, A.R., Strauss, C.E.M., Olmea, O., 2002. MAMMOTH (matching molecular models obtained from theory): An automated method for model comparison. *Protein Sci.* 11, 2606–2621.
- Ovchinnikov, S., Kim, D.E., Wang, R.Y.-R., *et al.*, 2016. Improved de novo structure prediction in CASP11 by incorporating coevolution information into Rosetta. *Proteins* 84 (Suppl. 1), S67–S75.
- Palopoli, N., Lanzarotti, E., Parisi, G., 2013. BeEP Server: Using evolutionary information for quality assessment of protein structure models. *Nucleic Acids Res.* 41, W398–W405.
- Palopoli, N., Monzon, A.M., Parisi, G., Fornasari, M.S., 2016. Addressing the role of conformational diversity in protein structure prediction. *PLOS ONE* 11, e0154923.
- Parisien, M., Cruz, J.A., Westhof, E., Major, F., 2009. New metrics for comparing and assessing discrepancies between RNA 3D structures and models. *RNA* 15, 1875–1885.
- Pawlowski, M., Bujnicki, J.M., 2012. The utility of comparative models and the local model quality for protein crystal structure determination by molecular replacement. *BMC Bioinform.* 13, 289.
- Pawlowski, M., Gajda, M.J., Matlak, R., Bujnicki, J.M., 2008. MetaMQAP: A meta-server for the quality assessment of protein models. *BMC Bioinform.* 9, 403.

- Pawlowski, M., Kozłowski, L., Kloczkowski, A., 2016. MQAPsingle: A quasi single-model approach for estimation of the quality of individual protein structure models. *Proteins* 84, 1021–1028.
- Pettitt, C.S., McGuffin, L.J., Jones, D.T., 2005. Improving sequence-based fold recognition by using 3D model quality assessment. *Bioinformatics* 21, 3509–3515.
- Ray, A., Lindahl, E., Wallner, B., 2012. Improved model quality assessment using ProQ2. *BMC Bioinform.* 13, 224.
- Read, R.J., Adams, P.D., Arendall, W.B., *et al.*, 2011. A new generation of crystallographic validation tools for the protein data bank. *Structure* 19, 1395–1412.
- Roche, D.B., Buenavista, M.T., McGuffin, L.J., 2014. Assessing the quality of modelled 3D protein structures using the ModFOLD server. *Methods Mol. Biol.* 1137, 83–103.
- Sali, A., Blundell, T.L., 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* 234, 779–815.
- Shen, M.-Y., Sali, A., 2006. Statistical potential for assessment and prediction of protein structures. *Protein Sci.* 15, 2507–2524.
- Shin, W.-H., Kang, X., Zhang, J., Kihara, D., 2017. Prediction of local quality of protein structure models considering spatial neighbors in graphical models. *Sci. Rep.* 7, 40629.
- Shindyalov, I.N., Bourne, P.E., 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng. Des. Sel.* 11, 739–747.
- Siew, N., Elofsson, A., Rychlewski, L., Fischer, D., 2000. MaxSub: An automated measure for the assessment of protein structure prediction quality. *Bioinformatics* 16, 776–785.
- Sippl, M.J., 1993. Recognition of errors in three-dimensional structures of proteins. *Proteins* 17, 355–362.
- Sippl, M.J., Wiederstein, M., 2012. Detection of spatial correlations in protein structures and molecular complexes. *Structure* 20, 718–728.
- Studer, G., Biasini, M., Schwede, T., 2014. Assessing the local structural quality of transmembrane protein models using statistical potentials (QMEANBrane). *Bioinformatics* 30, i505–i511.
- Tsutakawa, S.E., Hura, G.L., Frankel, K.A., Cooper, P.K., Tainer, J.A., 2007. Structural analysis of flexible proteins in solution by small angle X-ray scattering combined with crystallography. *J. Struct. Biol.* 158, 214–223.
- Uziela, K., Menéndez Hurtado, D., Shu, N., Wallner, B., Elofsson, A., 2017. ProQ3D: Improved model quality assessments using deep learning. *Bioinformatics* 33, 1578–1580.
- Uziela, K., Shu, N., Wallner, B., Elofsson, A., 2016. ProQ3: Improved model quality assessments using Rosetta energy terms. *Sci. Rep.* 6, 33509.
- Uziela, K., Wallner, B., 2016. ProQ2: Estimation of model accuracy implemented in Rosetta. *Bioinformatics* 32, 1411–1413.
- Wallner, B., 2014. ProQM-resample: Improved model quality assessment for membrane proteins by limited conformational sampling. *Bioinformatics* 30, 2221–2223.
- Wallner, B., Elofsson, A., 2003. Can correct protein models be identified? *Protein Sci.* 12, 1073–1086.
- Wallner, B., Elofsson, A., 2006. Identification of correct regions in protein models using structural, alignment, and consensus information. *Protein Sci.* 15, 900–913.
- Wallner, B., Elofsson, A., 2007. Prediction of global and local model quality in CASP7 using Pcons and ProQ. *Proteins* 69 (Suppl. 8), S184–S193.
- Wang, H.-W., Wang, J.-W., 2017. How cryo-electron microscopy and X-ray crystallography complement each other. *Protein Sci.* 26, 32–39.
- Wang, J., Mao, K., Zhao, Y., *et al.*, 2017. Optimization of RNA 3D structure prediction using evolutionary restraints of nucleotide-nucleotide interactions from direct coupling analysis. *Nucleic Acids Res.* 45, 6299–6309.
- Wang, J., Zhao, Y., Zhu, C., Xiao, Y., 2015. 3dRNAscore: A distance and torsion angle dependent evaluation function of 3D RNA structures. *Nucleic Acids Res.* 43, e63.
- Weeks, K.M., Mauger, D.M., 2011. Exploring RNA structural codes with SHAPE chemistry. *Acc. Chem. Res.* 44, 1280–1291.
- Weinreb, C., Riesselman, A.J., Ingraham, J.B., *et al.*, 2016. 3D RNA and functional interactions from evolutionary couplings. *Cell* 165, 963–975.
- Westbrook, J., Feng, Z., Burkhardt, K., Berman, H.M., 2003. Validation of protein structures for protein data bank. In: Carter, C.W., Sweet, R.M. (Eds.), *Macromolecular Crystallography, Part D. Methods in Enzymology*. Elsevier, pp. 370–385.
- Wiedemann, J., Zok, T., Milostan, M., Szachniuk, M., 2017. LCS-TA to identify similar fragments in RNA 3D structures. *BMC Bioinform.* 18, 456.
- Word, J.M., Lovell, S.C., LeBeau, T.H., *et al.*, 1999. Visualizing and quantifying molecular goodness-of-fit: Small-probe contact dots with explicit hydrogen atoms. *J. Mol. Biol.* 285, 1711–1733.
- Wüthrich, K., 2001. The way to NMR structures of proteins. *Nat. Struct. Biol.* 8, 923–925.
- Ye, Y., Godzik, A., 2004. FATCAT: A web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Res.* 32, W582–W585.
- Young, J.Y., Westbrook, J.D., Feng, Z., *et al.*, 2017. OneDep: Unified wwPDB system for deposition, biocuration, and validation of macromolecular structures in the PDB archive. *Structure* 25, 536–545.
- Zemla, A., 2003. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Res.* 31, 3370–3374.
- Zhang, J., Wang, Q., Vantasini, K., *et al.*, 2011. A multilayer evaluation approach for protein structure prediction and model quality assessment. *Proteins* 79 (Suppl. 10), S172–S184.
- Zhang, Y., Skolnick, J., 2004. Scoring function for automated assessment of protein structure template quality. *Proteins* 57, 702–710.
- Zok, T., Popena, M., Szachniuk, M., 2014. MCQ4Structures to compute similarity of molecule structures. *Cent. Eur. J. Oper. Res.* 22, 457–473.

Further Reading

- Jain, S., Richardson, D.C., Richardson, J.S., 2015. Computational methods for RNA structure validation and improvement. *Methods Enzymol.* 558, 181–212. doi:10.1016/bs.mie.2015.01.007.
- Kufareva, I., Abagyan, R., 2012. Methods of protein structure comparison. *Methods Mol. Biol.* 857, 231–257. doi:10.1007/978-1-61779-588-6_10.
- McGuffin, L.J., 2010. Model quality prediction (Chapter 15). In: Rangwala, H., Karypis, G. (Eds.), *Introduction to Protein Structure Prediction*. Hoboken, New Jersey, USA: Published by John Wiley & Sons, Inc.
- Miao, Z., Westhof, E., 2017. RNA structure: Advances and assessment of 3D structure prediction. *Annu. Rev. Biophys.* 46, 483–503. doi:10.1146/annurev-biophys-070816-034125.
- Sierk, M.L., Kleywegt, G.J., 2004. Déjà Vu all over again: Finding and analyzing protein structure similarities. *Structure* 12, 2103–2111. doi:10.1016/j.str.2004.09.016.
- Wallner, B., Elofsson, A., 2008. Quality assessment of protein models. In: Bujnicki, J.M. (Ed.), *Prediction of Protein Structures, Functions, and Interactions*. Chichester, UK: John Wiley & Sons, Ltd. Available at: <http://dx.doi.org/10.1002/9780470741894.ch6>.

Relevant Websites

- <http://predictioncenter.org/>
CASP Protein Structure Prediction Center.
- http://psvs-1_5-dev.nesg.org/
Protein Structure Validation Software suite.
- <http://services.mbi.ucla.edu/SAVES/>
The Structure Analysis and Verification Server.
- <http://www.wwpdb.org/validation/validation-reports>
wwPDB Validation Reports: User Guide.

Biographical Sketch



Nicolás Palopoli has been computational biologist since 2005. He studied Biotechnology at *Universidad Nacional de Quilmes* (UNQ) in Buenos Aires, Argentina, where he also got his PhD title in Basic and Applied Sciences. He was a postdoctoral research fellow at *Universidad de Buenos Aires*, Argentina and University of Southampton, UK. He recently returned to the Structural Bioinformatics Group (UNQ) as Assistant Researcher from the National Scientific and Technical Research Council of Argentina (CONICET). He is always interested in understanding the structural properties, biological function and evolutionary relationships of proteins. His current research projects involve protein-protein interactions mediated by Short Linear Motifs (SLiMs) and the special features associated with protein intrinsic disorder. He is also a patient lecturer, respectful colleague and proud husband and dad.

Study of the Variability of the Native Protein Structure

Xusi Han, Woong-Hee Shin, Charles W Christoffer, Genki Terashi, Lyman Monroe, and Daisuke Kihara, Purdue University, West Lafayette, IN, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteins are flexible molecules. After being translated from a messenger RNA by a ribosome, a protein folds into its native structure (the structure of lowest free energy), which is suitable for carrying out its biological function. Although the native structure of a protein is stabilized by physical interactions of atoms including hydrogen bonds, disulfide bonds between cysteine residues, van der Waals interactions, electrostatic interactions, and solvation (interactions with solvent), the structure still admits flexible motions. Motions include those of side-chains, and some parts of main-chains, especially regions that do not form the secondary structures, which are often called loop regions. In many cases, the flexibility of proteins plays an important or essential role in the biological functions of the proteins. For example, for some enzymes, such as triosephosphate isomerase (Derreumaux and Schlick, 1998), loop regions that exist in vicinity of active (i.e., enzymatic reaction) sites, takes part in binding and holding a ligand molecule. Transporters, such as maltose transporter (Chen, 2013), are known to make large open-close motions to transfer ligand molecules across the cellular membrane. For many motor proteins, such as myosin V that “walk” along actin filament as observed in muscle contraction (Kodera and Ando, 2014), flexibility is the central for their functions.

Reflecting such intrinsic flexibility of protein structures, differences are observed in protein tertiary structures determined by experimental methods such as X-ray crystallography, nuclear magnetic resonance (NMR) when they are solved under different conditions. In this article, we start by introducing two studies that surveyed such structural differences of the same proteins found in the public repository of protein structures, Protein Data Bank (PDB) (Berman *et al.*, 2000), which contains over 136,000 entries at the time of writing this article (December 2017). Thus, structural variability of a protein can be observed by comparing static structures determined under different conditions. Experimentally, conformational variability can be measured by NMR and other spectroscopic techniques (Greenleaf *et al.*, 2007; Parak, 2003; Hinterdorfer and Dufrene, 2006). Alternatively, structural changes, i.e., flexibility, can be observed for a single protein structure by performing computational simulations or predictions of dynamics of the protein structure. Computational methods for elucidating protein structure flexibility are also useful for estimating the free energy of molecular interactions as well as structure refinement needed when determining protein tertiary structures. In this article we overview such computational tools with some examples of application.

Discussion in this article is focused on protein structures that have overall stable fixed structures (with some flexibility in side-chains and parts of the main-chain, e.g., loops). However, note that there is a different class of proteins that do not form stable structures at all under physiological conditions. Such proteins are called intrinsically disordered proteins (IDPs) (Dunker *et al.*, 2008). IDPs were not paid much attention for a long time since their structures cannot be determined by regular experimental structural biology methods, such as X-ray crystallography or NMR, due to their intrinsic flexibility. But from around 2000, IDPs attracted large attention as long-neglected protein structures, and have been studied extensively since then. IDPs have characteristic amino acid sequence patterns, from which IDPs can be predicted from their amino acid sequences (Ferron *et al.*, 2006). It is estimated that about 5%–30% of amino acid sequences in an organism's proteome are intrinsically disordered (Oates *et al.*, 2013). It was found that disordered regions are often responsible for establishing protein-protein interactions, especially for proteins that interact with multiple proteins. The D2P2 database (Oates *et al.*, 2013) provides predicted IDPs in over 1700 genomes by many existing IDP prediction methods.

Fundamentals

Conformational Transitions Observed in Experimentally Determined Structures

Structural variability of proteins can be observed by comparing structures that are experimentally determined in different conditions. Kosloff and Kolodny (2008) performed a systematic study on protein structure variability relative to the sequence identity including cases that two structures are 100% identical in their sequences. The dataset was comprised of 19,295 protein chains taken from the April 2005 PDB, which are longer than 35 residues and were determined at a resolution of 2.5 Å or higher using X-ray crystallography. To eliminate redundant protein chains in the dataset, no pairs have 100% sequence identity and less than 1 Å root-mean square deviation (RMSD) between each other. Out of the 1941 chain pairs with 100% sequence identity in this dataset, there were 444 (22.9%) pairs with an RMSD of 3 Å or larger and 158 (8.1%) pairs with an RMSD over 6 Å. Such cases included chain pairs that have very different structures of over a 10 Å RMSD. They classified the sources of structural dissimilarity, which included different quaternary protein-protein interactions, protein-ligand and protein-DNA/RNA interactions, different crystallization conditions such as pH or salt conditions, and alternative crystallographic conformations of the same proteins. Thus, as is also discussed in other studies (Goh *et al.*, 2004; Boehr *et al.*, 2009), one of the main reasons for the existence of alternative conformations is due to physical interactions with other molecules. Most large conformational changes observed in protein structures are inherent to their biological functions, as Gan *et al.* (2002) pointed out in their work.

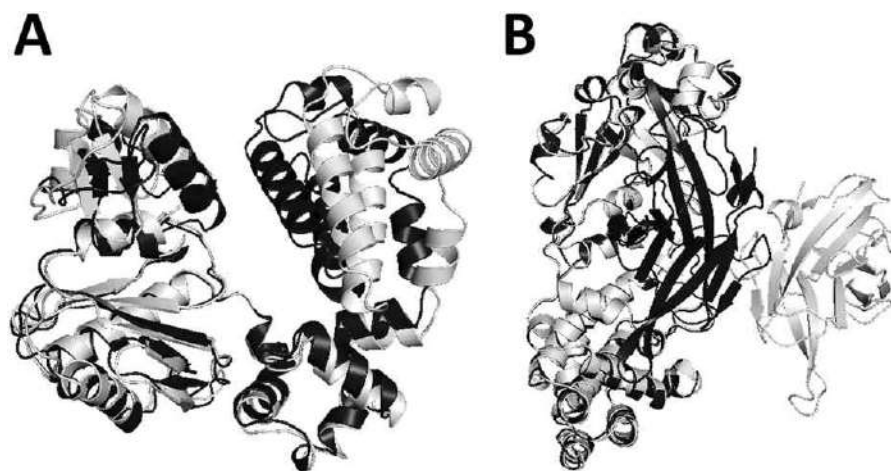


Fig. 1 Superimpositions of protein structures with a large conformational change. (A) Structure of DNA polymerase β in open (PDB ID: 9ICX, chain A, colored in light grey) and closed (PDB: 1BPY, chain A, colored in black) conformations. DNA in the crystal structures are omitted in the figure. (B) Structure of diphtheria toxin in open (PDB: 1DDT, chain A, colored in light grey) and closed (PDB: 1MDT, chain A, colored in black) conformations.

We show several examples of large conformational change of proteins. The first example is human DNA polymerase β in open and closed conformations (**Fig. 1(A)**) (Pelletier *et al.*, 1996; Sawaya *et al.*, 1997). The RMSD between the two structures is 5.3 Å. This conformational change is needed for its biological function, catalytic polymerization of nucleotides as well as binding and releasing DNA. The second example is diphtheria toxin in open and closed conformations (**Fig. 1(B)**). The RMSD is 11.0 Å (Bennett and Eisenberg, 1994). This protein has three domains, the catalytic domain, the transmembrane domain, and the C-terminus receptor binding domain. In the two structures superimposed in **Fig. 1(B)**, the arrangement of these three domains is very different. Particularly, the receptor domain shown on the right side of the figure is distant from the rest of the structure in the open conformation. This drastic conformational change is speculated to be essential for penetrating the host cell membrane, a critical step for intoxication.

Fig. 2 shows three more examples where binding with other molecule is involved in the conformational change. **Fig. 2(A)** shows structure of export chaperone FliS in *Aquifex aeolicus* (Evdokimov *et al.*, 2003). FliS acts as a flagellar export chaperone, which binds specifically to FliC to regulate its export and assembly process. The structure of unbound FliS is an antiparallel four-helix bundle (**Fig. 2(A)**, Left). The N-terminal cap blocks the hydrophobic binding site in FliS when FliC is absent. Upon complex formation with FliC, there is a substantial conformation change in FliS (**Fig. 2(A)**, Right), where the N-terminus in FliS displaces to form a short helix on one side of the bundle (on the right side of the panel), interacting with one helical segment in FliC. The RMSD between the two conformation is 6.4 Å. The next one is a textbook example, calmodulin (**Fig. 2(B)**). Calmodulin undergoes a large conformational change between the apo (i.e., non-ligand binding) state and the calcium-binding state (Yamniuk and Vogel, 2004). In calcium-free calmodulin, each globular domain is made up of four helices running in parallel/antiparallel orientations to each other (**Fig. 2(B)**, Left). In the calcium-bound form, the interdomain region forms a long α -helix (the right panel) instead of two short α -helices in the apo form. This helical rearrangement exposes hydrophobic binding patches on the surface of each domain, which binds to peptide sequences in target enzymes. The RMSD between the two structures is 12.9 Å. The last example in **Fig. 2** is RfaH, a member of a universally conserved family of transcription factors (**Fig. 2(C)**). In its closed form, RfaH C-terminal domain (CTD) forms an α hairpin that masks an RNA Polymerase binding site in RfaH N-terminal domain (NTD) (Belogurov *et al.*, 2007) (**Fig. 2(C)**, Left, the light grey region). But upon release from RfaH-NTD, RfaH-CTD refolds into a β barrel that binds to the ribosome to activate translation (the right panel) (Burmam *et al.*, 2012). The RMSD between the two structures of the RfaH-CTD is 14.0 Å. The dramatic switch from α helix to β barrel transforms the function of RfaH from transcription factor to translation factor.

Molecular Dynamics

From this subsection, we introduce several computational methods that can simulate or model structural variability of proteins.

Molecular dynamics (MD) simulation is a computer simulation technique to investigate the movement of atoms and molecules based on the principles of physics. Since its first application for biomolecules published 40 years ago (Mccammon *et al.*, 1977), it has become a popular and standard tool for studying biomolecular motion. The basic concepts of MD are (1) to divide time into discrete time steps (1 or 2 fs, generally) and (2) to solve Newton's equations of motion (Eq. (1)) at every time step:

$$F(\mathbf{x}) = -\nabla U(\mathbf{x}) = m \frac{d^2 \mathbf{x}}{dt^2} \quad (1)$$

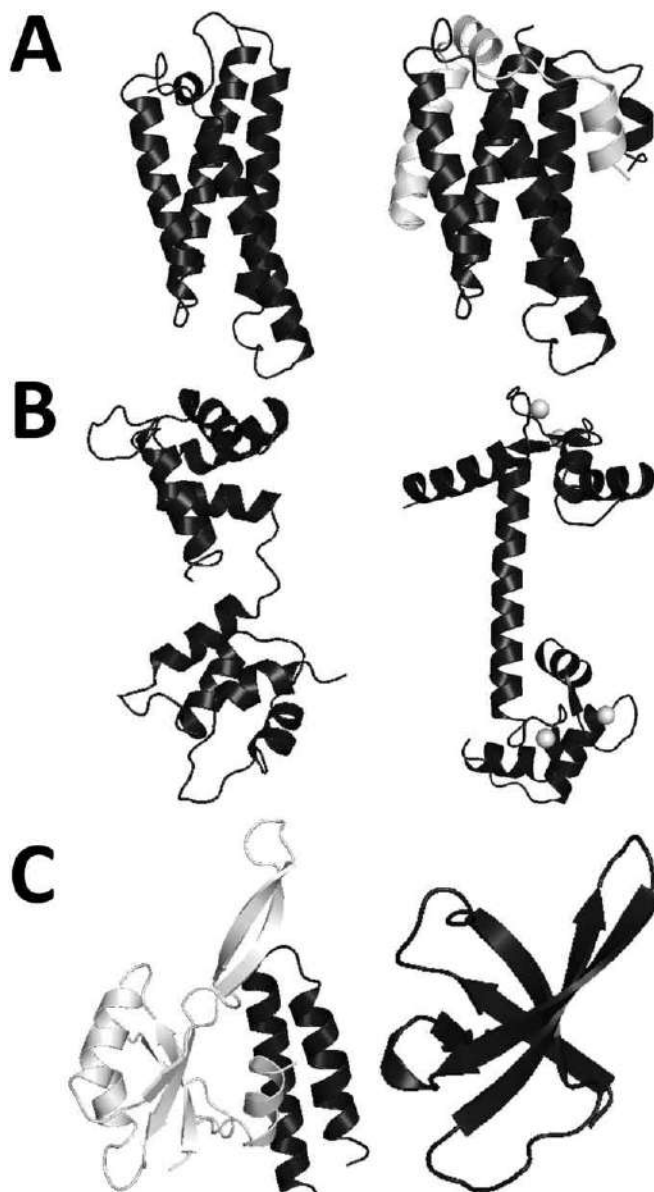


Fig. 2 Conformational transitions upon protein-ligand interactions. (A) Structure of FliS in unbound state (PDB: 1ORJ, chain A, left panel) and bound state (PDB: 1ORY, right panel). FliC is colored in light grey in the FliS-FliC complex. (B) Structure of calmodulin in the unbound state (PDB: 1DMO, chain A) and the calcium-binding state (PDB: 1CLL, chain A). Calcium ions in the crystal structure are shown as spheres colored in light grey on the right panel. (C) Structure of full-length RfaH in closed form (PDB: 2OUG, chain A) and RfaH C terminal domain in open form (PDB: 2LCL, chain A). RfaH N terminal domain is colored in light grey in the closed state (left).

$F(\mathbf{x})$, $U(\mathbf{x})$, $\mathbf{x}$, and m are a force, a potential energy, a coordinate of an atom, and a mass of an atom, respectively. Since the system is composed of a number of atoms, Eq. (1) cannot be solved analytically. Therefore, the trajectory of an atom can be obtained by integrating the potential energy at every time step numerically. The most famous algorithm for the numerical integration is Verlet algorithm (Verlet, 1967). A potential energy acting on an atom, $U(\mathbf{x})$, is calculated by a force field, which is composed of a functional form representing each force terms. Basically, the force field is composed of bonded and nonbonded terms (Eq. (2)).

$$U_{total} = U_{Bonded} + U_{Nonbonded} \quad (2)$$

A bonded term is a pairwise interaction energy of atoms that are connected by covalent bonds. It is composed of four terms; bond, angle, torsion, and improper torsion (Fig. 3, Eq. (3)). Different from the previous three terms that work on all atoms, improper torsion term only acts on special sets of atoms such as a peptide bond and benzene and prevents a deviation from planarity of the atom sets.

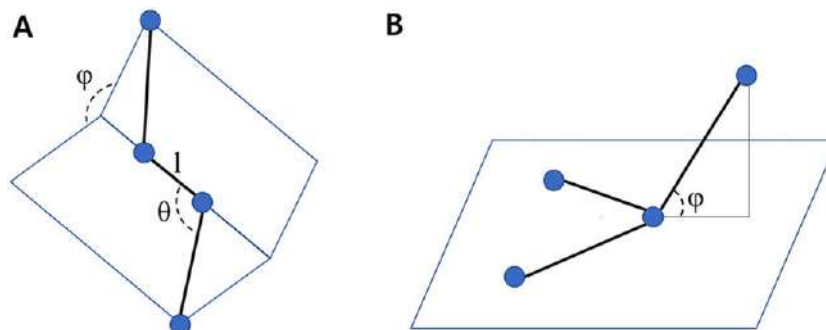


Fig. 3 Definitions of angles in the force field. (A) The bond length l , bond angle θ and the torsion angle, (φ) . (B) Improper torsion.

$$\begin{aligned}
 U_{\text{Bonded}} &= U_{\text{Bond}} + U_{\text{angle}} + U_{\text{Torsion}} + U_{\text{Improper}} \\
 &= \sum_{\text{Bonds}} \frac{k_{bi}}{2} (l - l_{i,eq})^2 + \sum_{\text{Angles}} \frac{k_{\theta i}}{2} (\theta_i - \theta_{i,eq})^2 + \sum_{\text{Torsions}} \sum_n \frac{V_{n,i}}{2} \left[1 + \cos(n(\varphi)_i - (\varphi)_{i,0}) \right] + \sum_{\text{Improper}} \frac{k_{(\varphi)i}}{2} ((\varphi)_i - (\varphi)_{i,eq})^2 \quad (3)
 \end{aligned}$$

The bond and the angle terms reflect vibration of bond lengths and angles along with covalent bonds. l and θ are the bond length and bond angle at the current state (Fig. 3(A)). The parameters, k_b , l_{eq} , k_θ , and θ_{eq} , are force constants and values of a bond in the equilibrium state. They are derived from gas phase spectroscopy and/or crystal structures of small molecules. The functional forms of the terms are approximated as harmonic oscillators. The torsion term calculates bond energies associated with proper torsion angles, (φ) in Fig. 3(A). Since it is periodic, it has a cosine functional form, and the parameters, V_n and $(\varphi)_0$ are obtained from quantum mechanics calculation. The improper torsion has a functional form of a harmonic oscillator (Fig. 3(B)).

A nonbonded term describes the long-range interaction of atoms, which consists of van der Waals and Coulomb potentials:

$$U_{\text{Nonbonded}} = U_{\text{van der Waals}} + U_{\text{Coulomb}} = \sum_{i=1}^N \sum_{j=i+1}^N \left\{ 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\epsilon r_{ij}} \right\} \quad (4)$$

In the van der Waals term, σ_{ij} and r_{ij} are the radius and well depth, respectively. They are calculated by taking the arithmetic mean ($\sigma_{ij} = (\sigma_{ii} + \sigma_{jj})/2$) and the geometric mean ($\epsilon_{ij} = \sqrt{\epsilon_{ii} \epsilon_{jj}}$) of parameters of the di-atomic system. For the Coulomb term, q_i is an atomic charge of atom i , and ϵ is a dielectric constant that reflects a solvent polarity (e.g., $\epsilon = 1$ for vacuum, $\epsilon = 4$ for protein, and $\epsilon = 80$ for water in implicit solvent model).

Most biological reactions take place in a cell, surrounded by water molecules; thus it is important to consider the effects of water. In MD and other simulation methods, water molecules can be treated in two ways; an implicit solvation and an explicit solvation model. An implicit solvation model (Fig. 4(A)), which is also called continuum solvent, considers solvent as a continuous isotropic medium. It approximates solvent molecules as a homogeneously polarizable, averaged medium (Skyner *et al.*, 2015). The main parameter for the continuum model is a dielectric constant ϵ , and supplementary parameters are used to describe solvent properties such as surface tension. Atoms in solution are represented as spherical cavities with atomic charges embedded in homogeneously polarizable solvent (Fig. 4(A)). An explicit solvation model (Fig. 4(B)) treats water molecules explicitly. This is a more realistic model than the implicit solvation model, since physical interaction between a solute and solvent molecules can be directly calculated from the same functional form of the force field. However, the number of atoms in the system is increased drastically in an explicit solvation model; therefore, it takes much more time to calculate potential energy of the solute atoms and trajectory of them. The most famous model of explicit water is TIP3P (Jorgensen *et al.*, 1983).

Coarse-Grained Protein Structure Models

If the size of a protein to simulate is very large, a MD simulation for a biologically relevant time span with an atomistic representation may be computationally infeasible. One way to solve this issue is to represent a protein structure with a simplified (coarse-grained, CG) model. In a coarse-grained model, the complexity of the system is reduced by grouping atoms together into pseudo particles (Barnoud and Monticelli, 2015). Reducing the number of particles decreases the number of interactions to calculate and makes the energy landscape smoother, both of which contribute to shortening the computational time for simulation compared to an atomistic model. The functional form of a CG force field is similar to that for an atomistic force field. The parameters of a CG force field are determined to be able to produce similar trajectories as atomistic MD simulation.

One of the well-known CG force fields is MARTINI (Marrink *et al.*, 2007). Generally, the MARTINI force field maps a group of four heavy atoms to one interaction site (bead). The beads of amino acids are mainly classified into four; polar, nonpolar, apolar, and charged. These are further categorized into subclasses based on hydrogen-bonding capabilities and polarity, resulting in 18 types of beads. Comparing to an atomistic simulation, the MARTINI force field yields a speed-up of 2–3 orders of magnitude. Other CG protein models include UNRES (United RESidues) (Liwo *et al.*, 2004, 2005) and CABS (C-Alpha, c-Beta, Side-chain)

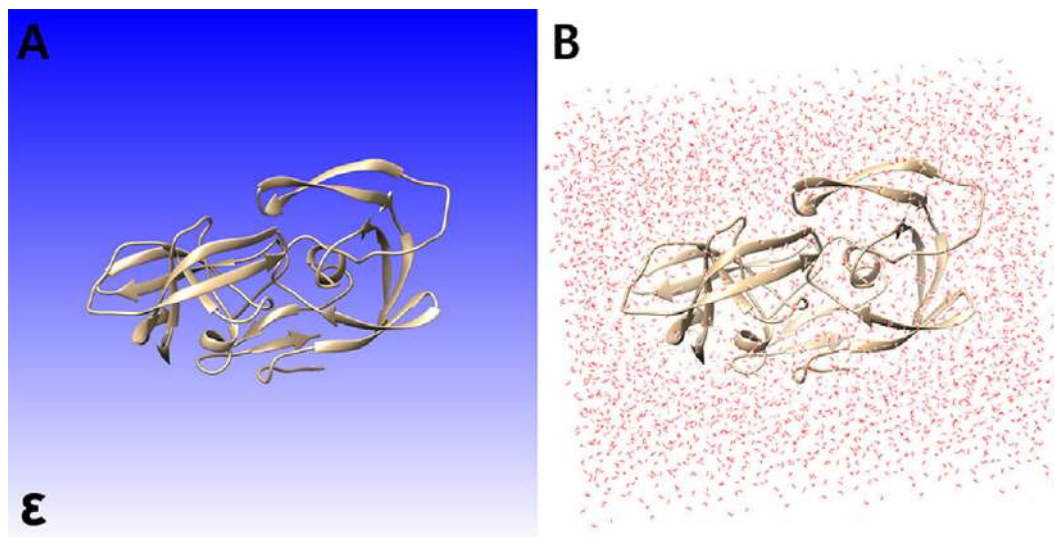


Fig. 4 Two solvent models. (A) The implicit solvent model, where solvent is represented as a dielectric constant of ϵ . (B) The explicit solvent model. A water molecule is represented as a bent line with oxygen and hydrogen atoms colored in red and white, respectively. Interactions between atoms in water molecules and proteins are computed in the same way using the equations of the force field used.

(Kolinski, 2004). In the UNRES model, each amino acid residue is represented by only two interaction sites, an ellipsoid at the side-chain centroid and another one at the center of $C\alpha-C\alpha$ bond. The force field of UNRES is derived as a restricted free energy function, which considers averaging the all-atom energy over the degrees of freedom that are neglected in the UNRES model. Thus, UNRES is a physics-based united residue force field. In the CABS model, each amino acid residue is represented by four interaction centers; the $C\alpha$ atom, the center of $C\alpha-C\alpha$ bond, the $C\beta$ atom, and the center of mass of the side-group. To facilitate faster simulations, $C\alpha$ atoms in the CABS model are projected onto a cubic lattice with 0.61 Å spacing. The force field of CABS is based on statistical potentials that mimic averaged interactions observed in globular protein structures in PDB.

Elastic Network Models

As discussed in Section “Coarse-Grained Protein Structure Models”, coarse-grained models of protein dynamics are used to reduce the computational cost of analysis. Some of the simplest such models are elastic network models (ENMs). In an ENM, pairwise interactions in a structure are modelled as Hookean springs at equilibrium, i.e., a model where beads representing interaction centers (e.g., amino acid residues) are connected by springs. In other words, the potential energy function has the form $\sum \frac{1}{2}\gamma_{ij}(r_{ij} - r_{ij}^0)^2$, where γ_{ij} are positive constants characterizing the stiffness of the interaction and r_{ij}^0 is the initial distance between atoms i and j ; in contrast to the general form of MD potentials given in Section “Coarse-Grained Protein Structure Models”, ENMs effectively model all interactions as bonds. Tirion considered an all-atom elastic network model with all γ_{ij} identical and atom pairs interacting if their distance was less than the sum of their van der Waals radii and an arbitrary cutoff (Tirion, 1996). That study established that normal mode analysis (NMA), which decomposes the general motion of a system into linearly independent modes of harmonic motion, of such potentials could reproduce the density of slow normal modes of more complicated and costly potentials. Hinsen showed that the slow modes of ENMs can in fact reproduce large-scale deformation, even when only α carbons were considered (Hinsen, 1998). Such coarse ENMs with a single point mass per residue have become popular; in the following sections, we discuss two such models.

Anisotropic network model

Although not chronologically the first residue-level ENM, the anisotropic network model (ANM) is the more general of the two we discuss here. Introduced by Doruker *et al.* (2000) and expounded by Atilgan *et al.* (2001) and Eyal *et al.* (2006), the ANM potential has all γ_{ij} identical and all interaction cutoffs identical (e.g., 7 Å). Here, the NMA is performed with respect to the Cartesian coordinates of each α carbon; i.e., the Hessian of the potential has the block matrix form

$$H = \begin{bmatrix} H_{ii} & \cdots & H_{ij} \\ \vdots & \ddots & \vdots \\ H_{ji} & \cdots & H_{jj} \end{bmatrix} \quad (5)$$

where each block expands to

$$H_{ij} = \begin{bmatrix} \frac{\partial^2 U_{ij}}{\partial x_i \partial x_j} & \frac{\partial^2 U_{ij}}{\partial x_i \partial y_j} & \frac{\partial^2 U_{ij}}{\partial x_i \partial z_j} \\ \frac{\partial^2 U_{ij}}{\partial y_i \partial x_j} & \frac{\partial^2 U_{ij}}{\partial y_i \partial y_j} & \frac{\partial^2 U_{ij}}{\partial y_i \partial z_j} \\ \frac{\partial^2 U_{ij}}{\partial z_i \partial x_j} & \frac{\partial^2 U_{ij}}{\partial z_i \partial y_j} & \frac{\partial^2 U_{ij}}{\partial z_i \partial z_j} \end{bmatrix} \quad (6)$$

Computing the eigenpairs of H yields $3N - 6$ eigenpairs corresponding to internal modes. The eigenvalues λ_k and frequencies ω_k of the modes are related by

$$\omega_k^2 = \gamma \lambda_k \quad (7)$$

Although H is not invertible, we can construct a pseudo-inverse

$$H^{-1} = \sum_{k=7}^{3N} \frac{u_k u_k^T}{\lambda_k} \quad (8)$$

where u_k are eigenvectors and eigenpairs are in increasing order by eigenvalue magnitude. Cross-correlations between the equilibrium fluctuations of residues i and j are then given by

$$C_{ij} = \frac{\text{tr}(H_{ij}^{-1})}{\sqrt{\text{tr}(H_{ii}^{-1}) \text{tr}(H_{jj}^{-1})}} \quad (9)$$

where H_{ij}^{-1} is the (i,j) th 3×3 block of H^{-1} . The B-factor of atom i , defined as $8\pi^2$ times the mean squared displacement of atom i ,

$$B_i = \frac{8\pi^2 k_B T}{3\gamma} \text{tr}(H_{ii}^{-1}) \quad (10)$$

where k_B is the Boltzmann constant and T is the temperature.

ANM has been used to study the dynamics of biomolecules that are related to their biological functions (Navizet *et al.*, 2004; Keskin *et al.*, 2002).

Gaussian network model

Compared to ANM, which can evaluate directional preferences of vibrational dynamics of proteins, Gaussian Network Model (GNM) only provides displacements and correlations between displacements of amino acid residues (Bahar *et al.*, 1997). Mathematically, any information obtained from GNM can also be obtained from ANM; however, if only isotropic information is needed, GNM is computationally cheaper. To calculate the isotropic cross-correlations and B-factors under a GNM, we start from the graph Laplacian Γ of the graph where vertices correspond to atoms and edges correspond to interactions. Then, the cross-correlations are given by

$$C_{ij} = \frac{\Gamma_{ij}^{-1}}{\sqrt{\Gamma_{ii}^{-1} \Gamma_{jj}^{-1}}} \quad (10)$$

where Γ^{-1} is a pseudoinverse as in Section “Anisotropic network model”, and the B-factors are given by

$$B_i = \frac{8\pi^2 k_B T}{3\gamma} \Gamma_{ii}^{-1} \quad (11)$$

Residue fluctuations computed from GNM have been shown to agree with the X-ray crystallographic B-factor, which reflects fluctuation of each atom at its position in the crystal structure (Bahar *et al.*, 1998). GNM has also been used for characterizing functional motions of proteins (Yang and Bahar, 2005) and macromolecules, such as ribosome (Wang *et al.*, 2004).

Flexpred

Variability of positions of residues in a protein structure can be also predicted from structural features of amino acids. Jamroz *et al.* (2012) developed FlexPred, which uses support vector regression (SVR) to predict residue fluctuations observed in 10 ns MD simulations, with a mean error in tests of 1.1 Å. Fluctuation is defined as

$$\sqrt{\langle (\Delta R_i)^2 \rangle^{\text{ref}}} = \sqrt{\frac{1}{T} \sum_{t_j=1}^T (x_i(t_j) - x_i^{\text{ref}})^2} \quad (12)$$

where T is the number of frames in the MD trajectory, $x_i(t_j)$ is the position of the α carbon of residue i at time t_j , and x_i^{ref} is the position of the α carbon of residue i in the protein structure. Structural features for each α carbon/residue that were found to be useful as input to SVR include distance to the structure's center of mass, the number of surrounding residues that are in contact, the

number of hydrophobic/hydrophilic contacts, solvent-accessible surface area, residue depth (the distance of the residue from protein surface), and secondary structure type.

Application and Case Studies

In this section we provide available software for analysing protein structure variability with some example applications.

Examples of Structural Variability in Protein Families

Structural variation of a protein can be quantified by superimposing the structures. Structure superimposition can help us reveal conserved regions among the structures which may be of functional importance as well as structure divergence and flexibility.

There have been a variety of multiple structure alignment methods developed over the years, which employ different scoring functions and different heuristics. Two examples are MAMMOTH-mult (Lupyan *et al.*, 2005) (see “Relevant Websites section”) and POSA (Ye and Godzik, 2005) (see “Relevant Websites section”). Both methods first conduct all pairwise structure alignments and constructs a dendrogram from pairwise similarity scores. Then, following the dendrogram, pairwise structure alignment is performed from leaf nodes, and finally all the structures are superimposed. This superimposition then undergoes an iterative refinement process (progressive alignment).

Multiple structure alignment is useful for understanding structural variability of single protein as well as structures of a protein family. Fig. 5 shows such an example, an alignment of three lectins, which have the legume lectin-like fold. Lectins are a group of carbohydrate binding proteins with large variation in size and structure. Characteristics of the lectin fold is the presence of a two

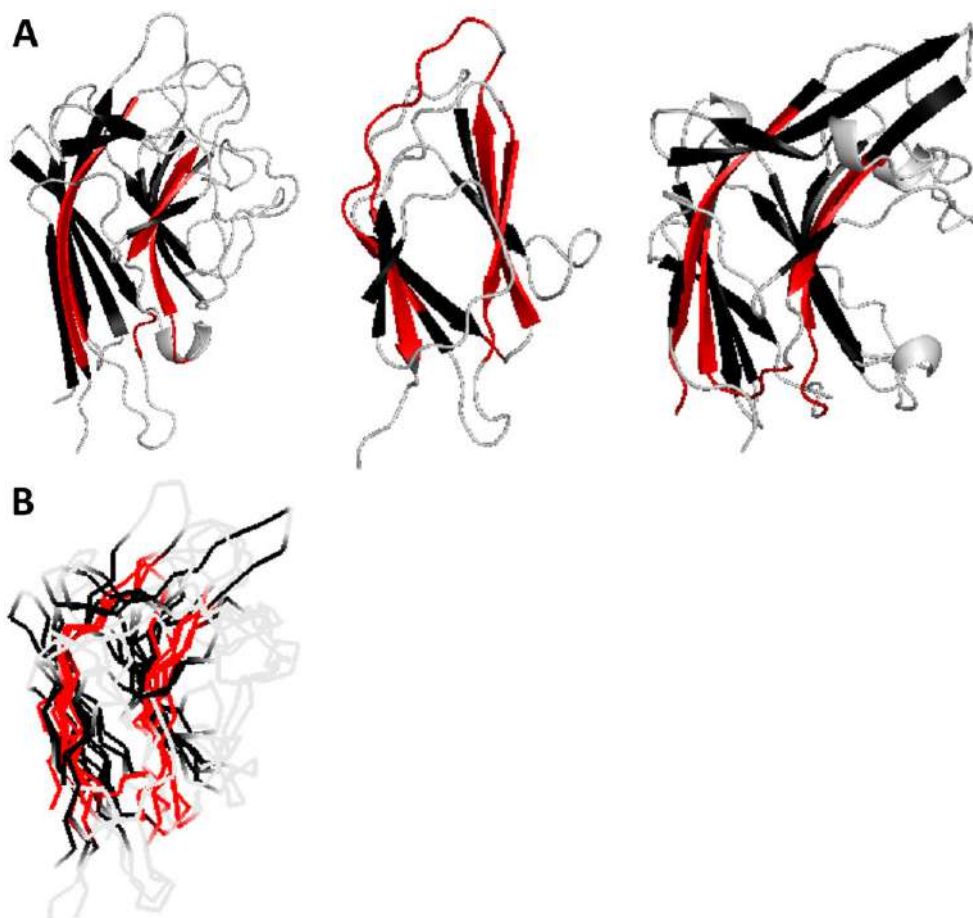


Fig. 5 Structure variation of proteins containing lectin fold. (A) Ribbon diagrams of three structures containing legume lectin-like fold. From left to right: Peanut lectin (PDB: 2PEL, chain A), Spermadhesin (PDB: 1SPP, chain A), Glucanase (PDB: 2AYH, chain A). All structures all shown in the same orientation. Common core identified in the POSA alignment are colored in red. Residues in common core are: residues 44–48, 51–53, 64–69, 200–209, 217–228 in 2pel-A; residues 28–35, 44–49, 85, 87–95, 97–108 in 1spp-A; residues 62–66, 69–77, 175–184, 203–214 in 2ayh-A. Beta sheets are colored in black. Helices and loops are colored in grey. (B) Superimposition of all structures. Structure alignments were produced using POSA.

Table 1 Popular molecular dynamics software

Name	Website
Amber	ambermd.org
CHARMM	www.charmm.org
GROMACS	gromacs.org
NAMD	www.ks.uiuc.edu/Research/namd
Desmond	www.deshawresearch.com/resources_desmond.html

β -sheets positioned almost in parallel (Chandra *et al.*, 2001). Although all structures with lectin fold possess the β -sheets with the particular geometry, curvature of a sheet and the presence of binding site loops and hydrophobic cores vary among those structures (Fig. 5(A)). The multiple structure alignment by POSA overlays β -sheets of the three proteins and particularly reveals that all three proteins possess a common core of 36 amino acids, three β strands shown in red in Fig. 5(B). The average RMSD between three structure pairs is 3.0 Å. Compared with peanut lectin (Left in Fig. 5(A)), spermadhesin (Middle) and glucanase (Right) have an RMSD of 2.88 Å (for aligned 80 residues) and an RMSD of 3.02 Å (for aligned 172 residues), respectively.

Software for Molecular Dynamics Simulations

MD can be performed using several program packages to simulate protein flexibility. Table 1 lists popular MD simulation programs. All of the programs are free for academic users. The first MD package, Amber (Assisted Model Building with Energy Refinement) (Case *et al.*, 2005), was originally developed for refining NMR structures. The name 'Amber' refers to both force fields for the simulation of biomolecules and the MD program package. The most widely used force field versions are Amber94, Amber99SB, and Amber03. The next one, CHARMM (Chemistry at HARvard Macromolecular Mechanics) (Brooks *et al.*, 2009), also refers to both force fields and the program package. It was originally developed by Martin Karplus of Harvard University, USA. It is the oldest biomolecular MD package listed in Table 1. CHARMM-GUI (see "Relevant Website section") provides a web-based graphical tools for setting up input files for the simulation. GROMACS (GRONingen MACHine for Chemical Simulations) (Pronk *et al.*, 2013) typically runs 3–10 times faster than other MD programs. Unlike Amber and CHARMM, GROMACS does not have its own force field. Instead, it can import Amber, CHARMM, GROMOS, and the OPLS force field to run MD simulation. The unique feature of the program is that it is an open-source software released under the GPL license. NAMD (NANoscale Molecular Dynamics) is developed by the Theoretical and Computational Biophysics Group at the University of Illinois, Urbana-Champaign, USA (Phillips *et al.*, 2005). It is designed to efficiently run on parallel machines for simulating large molecules. The program has high compatibility with other MD programs such as CHARMM, since NAMD has same input, output, and force field formats as CHARMM.

Desmond is developed by D.E. Shaw Research (Bowers *et al.*, 2006). The program uses its novel parallel algorithms and numerical methods to achieve high computing performance. Desmond can import AMBER, CHARMM, and the OPLS force field to run MD simulation.

MD-Based Protein Structure Model Refinement

From Sections "MD-Based Protein Structure Model Refinement", "Refinement of Structures from Electron Microscopy Data", "Protein Folding" and "Free Energy Calculation" we will discuss notable applications of MD simulation for addressing protein structure variability. The first one is structure refinement for computationally modelled protein structure models. Protein structures can be computationally modelled with a reasonable accuracy if a structure of a related protein is already solved and available in PDB. The protein structure modelling technique that uses known structures as a template is called homology modelling or comparative modelling (Fiser, 2010). However, structure models usually still have deviation from the correct (native) structure of the protein, which may be as subtle as side-chain orientations to as large as a domain or loop motion relative to the native structure. The basic assumption of using MD for refining a structure model is that the conformational ensemble generated by MD simulation from the model contains a better structure that is closer to the native structure. The approach can also be applied to structure models built from low-resolution experimental data such as low-resolution X-ray crystallography (McGreevy *et al.*, 2014) and cryo-electron microscopy (cryo-EM) density maps (Singharoy *et al.*, 2016).

In the field of protein structure prediction, It has been recognized that running MD simulations for a structure model does not improve a model consistently, but rather drifts the structure away from the native structure because the conformational space is very large. However, the situation has changed in 2012, when MD-based methods were developed that refined models consistently in a protein structure prediction contest, the Critical Assessment of Techniques for Protein Structure Prediction (CASP) (Nugent *et al.*, 2014). Since then several MD-based structure refinement methods have been developed. These methods typically run MD with constraints so that the overall structure does not change drastically. Also the methods combine a structural averaging step and cross-check structures with additional structure evaluation scores (Mirjalili and Feig, 2013; Lee *et al.*, 2018; Terashi and Kihara, 2018). A drawback of the current methods is that the structure improvement they can achieve is small due to the constraints applied in the simulation, which do not allow large conformational changes in the model.

Refinement of Structures From Electron Microscopy Data

MD is also used for refining protein structures derived from low-resolution experimental data. The experimental data referred to here is the three-dimensional reconstruction of electron microscopy (EM) data, commonly referred to as EM maps. The fundamental methodology for combining MD approaches and experimental data is to perform the MD with an additional metric meant to measure the quality of fit between the atomic model and the EM map.

One method for measuring quality of fit is to convert the EM map data into a potential energy landscape. In this methodology, high density regions of the EM map will have low energy penalties for atoms to occupy that space, while low density regions will have higher energy penalties for atoms to occupy that space. The energy term is then added to the rest of the terms in an MD simulation (Eq. (3)). The most popular method which applies a quality-of-fit metric in this way is Molecular Dynamics Flexible Fitting (MDFF) (Singharoy *et al.*, 2016). An alternative metric for fit quality is cross correlation. The cross correlation between an atomic model and an EM map is determined by generating a simulated density map from the atomic model and determining the similarity of the simulated and experimental maps. A popular method which implements this kind of quality of fit metric is ROSETTA (Dimaio *et al.*, 2015). It is important when implementing these kinds of refinement techniques to remember that the refinement can only be as good as the experimental data allows. As the resolution of EM maps decreases, the extent to which multiple refinement techniques agree with each other will tend to decrease (Monroe *et al.*, 2017).

Protein Folding

Many proteins fold over a time scale on the order of millisecond or longer. Some proteins in the low millisecond range are Trp cage and Villin headpiece with sizes of 20 and 35 residues, respectively. These two α -helical proteins have successfully been simulated from unfolded states to folded states in simulations lasting 10 ns to 1 ms (Simmerling *et al.*, 2002; Duan and Kollman, 1998), and achieving C α RMSDs of 0.97 and 3.0 Å, respectively. On an even larger scale, the folding and unfolding of ubiquitin, a 76 residue $\alpha\beta$ protein, has been studied through high temperature, 1 ms simulations to an RMSD of 0.5 Å to the crystal structure of the protein (Piana *et al.*, 2013).

Free Energy Calculation

One of the most interesting applications of MD simulation is free energy calculation of a system. From a statistical thermodynamics point of view, free energy calculation is based on the ergodicity hypothesis, which claims that all possible microstates in the phase space can be covered by a trajectory of a system in a long time. With this hypothesis, we can assume that the average of an observable, such as enthalpy, over time is the same as its statistical ensemble. One of the classical example of this application is a protein folding energy landscape (Lei *et al.*, 2007).

Another useful application of free energy calculation is protein-ligand binding free energy (ΔG_{bind}) using The Molecular Mechanics energies combined with the Poisson–Boltzmann and Surface Area continuum solvation (MM/PBSA) method (Genheden and Ryde, 2015). MM/PBSA starts by generating MD trajectories of a protein-ligand complex, a protein, and a ligand. From the trajectories, snapshots of the system are picked up with a certain time interval. The Gibbs free energy of a molecule at a snapshot is calculated as:

$$G = E_{\text{Bond}} + E_{\text{vdW}} + E_{\text{Coul}} + G_{\text{pol}} + G_{\text{np}} - TS \quad (13)$$

E_{Bond} , E_{vdW} , and E_{Coul} are bonded term, van der Waals, and Coulomb force of a molecule, respectively. The three terms are calculated by the molecular mechanics force field (MM). G_{pol} and G_{np} are solvation energies of polar and nonpolar atoms of a molecule, obtained by solving the Poisson-Boltzmann equation for polar atoms and is proportional to surface area for nonpolar atoms. Although the MD trajectory is obtained from explicit solvent condition, the solvation energy is calculated with implicit solvation (PBSA). The last term, TS in Eq. (13) is entropy, calculated from normal mode analysis, multiplied by absolute temperature.

By taking average of Gibbs free energy across snapshots, binding free energy is calculated as follows:

$$\Delta G_{\text{bind}} = \langle G_{\text{Protein-Ligand}} \rangle - \langle G_{\text{Protein}} \rangle - \langle G_{\text{Ligand}} \rangle \quad (14)$$

$\langle G_i \rangle$ is an average of Gibbs free energy of a molecule i . The MM/PBSA has been successfully applied to estimating binding free energies of drugs (Zoete *et al.*, 2003; Pearlman, 2005).

Protein Flexibility Prediction With Flexpred

As discussed in Section “FlexPred”, flexibility of a protein structure can be predicted from structural features of amino acids in the structure. The method, FlexPred is available as a web server (see “Relevant Websites section”) and downloadable software (Peterson *et al.*, 2017). Fig. 6 shows an example of prediction for bovine angiogenin (PDB: 1AGI, 125 residues long). The plot shows flexibility, how far in Euclidean distance (Å) each amino acid moves on average (in 10 ns MD simulation) along the protein chain (Eq. (12)). At the bottom of the page, there are links to download the fluctuation predictions in CSV form and in the B-factor field of a PDB file. In this example, the predicted fluctuations and crystallographic B-factors correlate with correlation coefficient 0.82. Fig. 6(B) shows a side-by-side comparison of the predicted fluctuations (Left) and the B-factors (Right).

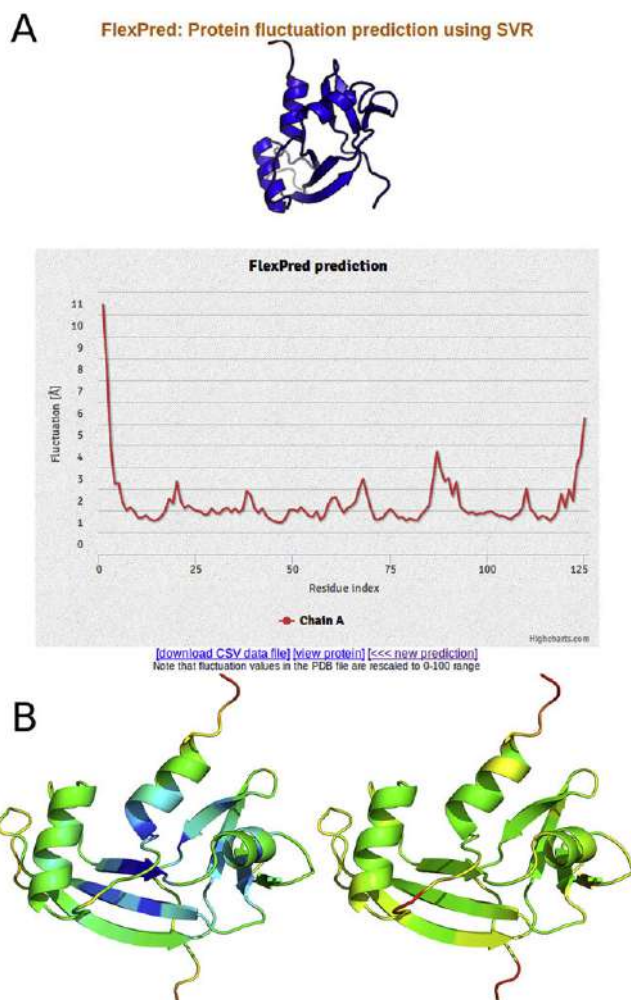


Fig. 6 Example of protein structure flexibility prediction. FlexPred was used. (A) Predicted fluctuation of residues of bovine angiogenin (PDB ID: 1AGI). The y-axis shows the fluctuation (Eq. (12)) of each residue. (B) Predicted fluctuation mapped on residues in the PDB file. Values are stored in the B-factor field of the PDB file. Left, prediction; Right, structure colored by crystallographic B-factor of 1AGI.

Conclusion

Proteins are flexible molecules. Some motions are important for biological function of proteins. How the structure of a protein varies can be obtained if the protein structure is solved under different conditions by experimental methods. Variability of a structure can be also experimentally measured by using NMR or spectroscopic techniques. Computationally, structure variability can be simulated by atomistic MD or coarse-grained models, and can also be predicted from the structure. Applications of MDs, multiple structure alignment, and flexibility prediction are shown.

Acknowledgement

This work was partly supported by the National Institutes of Health (R01GM123055) and the National Science Foundation (IIS1319551, IOS1127027, DMS1614777).

See also: Biological Database Searching. Biomolecular Structures: Prediction, Identification and Analyses. Natural Language Processing Approaches in Bioinformatics

References

- Atilgan, A.R., Durell, S.R., Jernigan, R.L., *et al.*, 2001. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.* 80, 505–515.
- Barnoud, J., Monticelli, L., 2015. Coarse-grained force fields for molecular simulations. *Methods Mol. Biol.* 1215, 125–149.
- Bahar, I., Atilgan, A.R., Erman, B., 1997. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Fold. Des.* 2, 173–181.
- Bahar, I., Wallqvist, A., Covell, D.G., Jernigan, R.L., 1998. Correlation between native-state hydrogen exchange and cooperative residue fluctuations from a simple model. *Biochemistry* 37, 1067–1075.
- Belogurov, G.A., Vassilyeva, M.N., Svetlov, V., *et al.*, 2007. Structural basis for converting a general transcription factor into an operon-specific virulence regulator. *Mol. Cell* 26, 117–129.
- Bennett, M.J., Eisenberg, D., 1994. Refined structure of monomeric diphtheria toxin at 2.3 Å resolution. *Protein Sci.* 3, 1464–1475.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The Protein Data Bank. *Nucleic Acids Res.* 28, 235–242.
- Boehr, D.D., Nussinov, R., Wright, P.E., 2009. The role of dynamic conformational ensembles in biomolecular recognition. *Nat. Chem. Biol.* 5, 789–796.
- Bowers, K.J., Chow, E., Xu, H., *et al.*, 2006. Scalable algorithms for molecular dynamics simulations on commodity clusters. In: *Proceedings of the ACM/IEEE Conference on Supercomputing (SC06)*. Tampa, Florida.
- Brooks, B.R., Brooks, C.L., Mackerell 3rd, A.D., *et al.*, 2009. CHARMM: The biomolecular simulation program. *J. Comput. Chem.* 30, 1545–1614.
- Burmann, B.M., Knauer, S.H., Sevostyanova, A., *et al.*, 2012. An alpha helix to beta barrel domain switch transforms the transcription factor RfaH into a translation factor. *Cell* 150, 291–303.
- Case, D.A., Cheatham, T.E., Darden 3rd, T., *et al.*, 2005. The Amber biomolecular simulation programs. *J. Comput. Chem.* 26, 1668–1688.
- Chandra, N.R., Prabu, M.M., Suguna, K., Vijayan, M., 2001. Structural similarity and functional diversity in proteins containing the legume lectin fold. *Protein Eng.* 14, 857–866.
- Chen, J., 2013. Molecular mechanism of the Escherichia coli maltose transporter. *Curr. Opin. Struct. Biol.* 23, 492–498.
- Derreumaux, P., Schlick, T., 1998. The loop opening/closing motion of the enzyme triosephosphate isomerase. *Biophys. J.* 74, 72–81.
- Dimajo, F., Song, Y., Li, X., *et al.*, 2015. Atomic-accuracy models from 4.5-Å cryo-electron microscopy data with density-guided iterative local refinement. *Nat. Methods* 12, 361–365.
- Doruker, P., Atilgan, A.R., Bahar, I., 2000. Dynamics of proteins predicted by molecular dynamics simulations and analytical approaches: Application to alpha-amylase inhibitor. *Proteins Struct. Funct. Genet.* 40, 512–524.
- Duan, Y., Kollman, P.A., 1998. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science* 282, 740–744.
- Dunker, A.K., Silman, I., Uversky, V.N., Sussman, J.L., 2008. Function and structure of inherently disordered proteins. *Curr. Opin. Struct. Biol.* 18, 756–764.
- Evdokimov, A.G., Phan, J., Tropea, J.E., *et al.*, 2003. Similar modes of polypeptide recognition by export chaperones in flagellar biosynthesis and type III secretion. *Nat. Struct. Biol.* 10, 789–793.
- Eyal, E., Yang, L.W., Bahar, I., 2006. Anisotropic network model: Systematic evaluation and a new web interface. *Bioinformatics* 22, 2619–2627.
- Ferron, F., Longhi, S., Canard, B., Karlin, D., 2006. A practical overview of protein disorder prediction methods. *Proteins* 65, 1–14.
- Fiser, A., 2010. Template-based protein structure modeling. *Methods Mol. Biol.* 673, 73–94.
- Gan, H.H., Perlow, R.A., Roy, S., *et al.*, 2002. Analysis of protein sequence/structure similarity relationships. *Biophys. J.* 83, 2781–2791.
- Genheden, S., Ryde, U., 2015. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert Opin. Drug Discov.* 10, 449–461.
- Goh, C.S., Milburn, D., Gerstein, M., 2004. Conformational changes associated with protein-protein interactions. *Curr. Opin. Struct. Biol.* 14, 104–109.
- Greenleaf, W.J., Woodside, M.T., Block, S.M., 2007. High-resolution, single-molecule measurements of biomolecular motion. *Annu. Rev. Biophys. Biomol. Struct.* 36, 171–190.
- Hinsen, K., 1998. Analysis of domain motions by approximate normal mode calculations. *Proteins Struct. Funct. Genet.* 33, 417–429.
- Hinterdorfer, P., Dufrene, Y.F., 2006. Detection and localization of single molecular recognition events using atomic force microscopy. *Nat. Methods* 3, 347–355.
- Jamroz, M., Kolinski, A., Kihara, D., 2012. Structural features that predict real-value fluctuations of globular proteins. *Proteins* 80, 1425–1435.
- Jorgensen, W.L., Chandrasekhar, J., Madura, J.D., Impey, R.W., Klein, M.L., 1983. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* 79, 926–935.
- Keskin, O., Durell, S.R., Bahar, I., Jernigan, R.L., Covell, D.G., 2002. Relating molecular flexibility to function: A case study of tubulin. *Biophys. J.* 83, 663–680.
- Kodera, N., Ando, T., 2014. The path to visualization of walking myosin V by high-speed atomic force microscopy. *Biophys. Rev.* 6, 237–260.
- Kolinski, A., 2004. Protein modeling and structure prediction with a reduced representation. *Acta Biochim. Pol.* 51, 349–371.
- Kosloff, M., Kolodny, R., 2008. Sequence-similar, structure-dissimilar protein pairs in the PDB. *Proteins* 71, 891–902.
- Lee, G.R., Heo, L., Seok, C., 2017. Simultaneous refinement of inaccurate local regions and overall structure in the CASP12 protein model refinement experiment. *Proteins* 86 (Suppl. 1), S168–S176.
- Lei, H., Wu, C., Liu, H., Duan, Y., 2007. Folding free-energy landscape of villin headpiece subdomain from molecular dynamics simulations. *Proc. Natl. Acad. Sci. USA* 104, 4925–4930.
- Liwo, A., Khalili, M., Scheraga, H.A., 2005. Ab initio simulations of protein-folding pathways by molecular dynamics with the united-residue model of polypeptide chains. *Proc. Natl. Acad. Sci. USA* 102, 2362–2367.
- Liwo, A., Oldziej, S., Czaplewski, C., Kozłowska, U., Scheraga, H.A., 2004. Parameterization of backbone-electrostatic and multibody contributions to the UNRES force field for protein-structure prediction from ab initio energy surfaces of model systems. *J. Phys. Chem. B* 108, 9421–9438.
- Lupyan, D., Leo-Macias, A., Ortiz, A.R., 2005. A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics* 21, 3255–3263.
- Marrink, S.J., Risselada, H.J., Yefimov, S., Tieleman, D.P., De Vries, A.H., 2007. The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B* 111, 7812–7824.
- Mccammon, J.A., Gelin, B.R., Karplus, M., 1977. Dynamics of folded proteins. *Nature* 267, 585–590.
- McGreevy, R., Singharoy, A., Li, Q., *et al.*, 2014. xMDFF: Molecular dynamics flexible fitting of low-resolution X-ray structures. *Acta Crystallogr. D Biol. Crystallogr.* 70, 2344–2355.
- Mirjalili, V., Feig, M., 2013. Protein structure refinement through structure selection and averaging from molecular dynamics ensembles. *J. Chem. Theory Comput.* 9, 1294–1303.
- Monroe, L., Terashi, G., Kihara, D., 2017. Variability of protein structure models from electron microscopy. *Structure* 25, 592–602. e2.
- Navizet, I., Lavery, R., Jernigan, R.L., 2004. Myosin flexibility: Structural domains and collective vibrations. *Proteins* 54, 384–393.
- Nugent, T., Cozzetto, D., Jones, D.T., 2014. Evaluation of predictions in the CASP10 model refinement category. *Proteins* 82 (Suppl. 2), S98–S111.
- Oates, M.E., Romero, P., Ishida, T., *et al.*, 2013. D(2)P(2): Database of disordered protein predictions. *Nucleic Acids Res.* 41, D508–D516.
- Parak, F.G., 2003. Proteins in action: The physics of structural fluctuations and conformational changes. *Curr. Opin. Struct. Biol.* 13, 552–557.
- Pearlman, D.A., 2005. Evaluating the molecular mechanics poisson-boltzmann surface area free energy method using a congeneric series of ligands to p38 MAP kinase. *J. Med. Chem.* 48, 7796–7807.
- Pelletier, H., Sawaya, M.R., Wolffe, W., Wilson, S.H., Kraut, J., 1996. Crystal structures of human DNA polymerase beta complexed with DNA: Implications for catalytic mechanism, processivity, and fidelity. *Biochemistry* 35, 12742–12761.
- Peterson, L., Jamroz, M., Kolinski, A., Kihara, D., 2017. Predicting real-valued protein residue fluctuation using FlexPred. *Methods Mol. Biol.* 1484, 175–186.
- Phillips, J.C., Braun, R., Wang, W., *et al.*, 2005. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* 26, 1781–1802.
- Piana, S., Lindorff-Larsen, K., Shaw, D.E., 2013. Atomic-level description of ubiquitin folding. *Proc. Natl. Acad. Sci. USA* 110, 5915–5920.
- Pronk, S., Pall, S., Schulz, R., *et al.*, 2013. GROMACS 4.5: A high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* 29, 845–854.
- Sawaya, M.R., Prasad, R., Wilson, S.H., Kraut, J., Pelletier, H., 1997. Crystal structures of human DNA polymerase beta complexed with gapped and nicked DNA: Evidence for an induced fit mechanism. *Biochemistry* 36, 11205–11215.

- Simmerling, C., Strockbine, B., Roitberg, A.E., 2002. All-atom structure prediction and folding simulations of a stable protein. *J. Am. Chem. Soc.* 124, 11258–11259.
- Singharoy, A., Teo, I., McGreevy, R., *et al.*, 2016. Molecular dynamics-based refinement and validation for sub-5 Å cryo-electron microscopy maps. *eLife*. 5.
- Skyner, R.E., McDonagh, J.L., Groom, C.R., Van Mourik, T., Mitchell, J.B.O., 2015. A review of methods for the calculation of solution free energies and the modelling of systems in solution. *Phys. Chem. Chem. Phys.* 17, 6174–6191.
- Terashi, G., Kihara, D., 2018. Protein structure model refinement in CASP12 using short and long molecular dynamics simulations in implicit solvent. *Proteins* 86 (Suppl. 1), S189–S201.
- Tirion, M.M., 1996. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys. Rev. Lett.* 77, 1905–1908.
- Verlet, L., 1967. Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.* 159, 98.
- Wang, Y., Rader, A.J., Bahar, I., Jernigan, R.L., 2004. Global ribosome motions revealed with elastic network model. *J. Struct. Biol.* 147, 302–314.
- Yamniuk, A.P., Vogel, H.J., 2004. Calmodulin's flexibility allows for promiscuity in its interactions with target proteins and peptides. *Mol. Biotechnol.* 27, 33–57.
- Yang, L.W., Bahar, I., 2005. Coupling between catalytic site and collective dynamics: A requirement for mechanochemical activity of enzymes. *Structure* 13, 893–904.
- Ye, Y., Godzik, A., 2005. Multiple flexible structure alignment using partial order graphs. *Bioinformatics* 21, 2362–2369.
- Zoete, V., Michielin, O., Karplus, M., 2003. Protein-ligand binding free energy estimation using molecular mechanics and continuum electrostatics. Application to HIV-1 protease inhibitors. *J. Comput. Aided Mol. Des.* 17, 861–880.

Relevant Websites

<http://www.charmm-gui.org>
 Charmm-Gui.

<http://kiharalab.org/flexPred/>
 FlexPred Server - Kihara Lab.

<https://ub.cbm.uam.es/software/online/mamothmult.php>
 MAMMOTH-Mult - Unidad de Bioinformática CBMSO.

<http://posa.sanfordburnham.org/>
 POSA Home - Sanford Burnham Prebys Medical Discovery Institute.

Biographical Sketch



Xusi Han is currently a PhD candidate in Department of Biological Sciences at Purdue University, West Lafayette, Indiana, USA. She is also pursuing toward the M.S. degree in Applied Statistics at Purdue University. She received the B.S. degree from Minzu University of China, Beijing, China, in 2009. Her current research interests include protein global structure comparison, protein structure classification, and graph analysis for electron microscopy density maps.



Woong-Hee Shin is currently a postdoctoral researcher of the Kihara Lab in the Department of Biological Sciences at Purdue University. He received the B.Sc. in chemistry from Seoul National University, South Korea in 2008. He also received the PhD degree in chemistry in 2014 from Seoul National University. During his PhD research, he developed a flexible-receptor ligand docking program, GalaxyDock, and performed virtual ligand library screening to find active molecules of nuclear receptors. After receiving his PhD, he joined the Kihara lab in August 2014. His current research interests are development and application of computational drug discovery programs.



Charles Christoffer is currently a graduate member of the Kihara Lab at Purdue University, West Lafayette, Indiana, USA. He received the B.S. degree in Computer Science and Applied Mathematics from Purdue University in 2015. He is currently pursuing the PhD degree with the Department of Computer Science at Purdue University. His research interests include geometric data structures, computer vision, and image processing, especially their application to structural bioinformatics. Within structural bioinformatics, he is particularly interested in protein docking and structure database search.



Genki Terashi is currently a postdoctoral researcher of the Kihara Lab in the Department of Biological Sciences at Purdue University, West Lafayette, Indiana, USA. He received the B.S., M.S., and PhD degrees in Pharmaceutical Sciences from Kitasato University, Tokyo, Japan, in 2002, 2004 and 2008, respectively. His current research interests include protein structure model refinement, protein structure modelling and comparison, and protein-protein docking. Application of Machine Learning method to the structural biology is also his research interest.



Lyman Monroe is a PhD candidate in the Department of Biological Sciences at Purdue University, West Lafayette, Indiana, USA. He received the B.S. degree in Physics from Purdue University in 2012, as well as a second B.S. degree in Chemistry from Purdue University in 2013. His projects include protein structure modelling and refinement using molecular dynamics simulation.



Daisuke Kihara is a full professor in the Department of Biological Sciences and the Department of Computer Science at Purdue University, West Lafayette, Indiana, USA. He received the B.S. degree in Biochemistry from the University of Tokyo, Japan in 1994, and the M.S. and PhD degrees from Kyoto University, Japan in 1996 and 1999, respectively. He was a postdoctoral researcher with Prof. Jeffrey Skolnick at the Danforth Plant Science Center, St. Louis and at SUNY Buffalo from 1999 to 2003. He joined Purdue University as an assistant professor in 2003 and was promoted to associate professor in 2009 and to full professor in 2014. His research field is bioinformatics. His research projects include protein docking, protein tertiary structure prediction, structure- and sequence-based protein function prediction, and computational drug design. He has published over 140 research papers and book chapters. His research projects have been supported by funding from the National Institutes of Health, the National Science Foundation, the Office of the Director of National Intelligence, and industry. In 2013, he was named a University Faculty Scholar by Purdue University.

Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures

Mohd Firdaus-Raih and Nur Syatila Abdul Ghani, Universiti Kebangsaan Malaysia, Bangi, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The primary computational approach used for assigning biological function to protein sequences is dependent on the extrapolation of function based on sequence level similarities. Although the increase in the number of sequences with annotated functions has made sequence alignment based tools highly accurate and adequate, there remains a subset of proteins that have yet to be characterized for function. In the numerous genome sequences being generated at present, many of these sequences are annotated as hypothetical proteins based on the prediction that their open reading frames can hypothetically code for a protein. Many such hypothetical proteins have been expressed, purified and had their structures determined by crystallographic methods via structural genomics approaches. This is evident in the number of proteins with uncharacterized functions that are being deposited in the Protein Data Bank (PDB) (Rose *et al.*, 2017). It is known that at levels where sequence similarity is no longer detectable, fold similarity may still exist and hence still result in similar functions. Therefore, considerable efforts have been made to determine the functions of such hypothetical proteins by characterizing their three-dimensional (3D) structures.

Although the molecular structure of a protein can be expected to reveal potential fold similarities to structures with characterized function, there are still protein structures in the PDB that do not have any fold similarities to any other characterized examples (Nadzirin and Firdaus-Raih, 2012). As a result, these proteins are still classified as having uncharacterized function despite having lost the hypothetical protein label. One approach towards identifying the potential function of such proteins would be to investigate the potential for specific arrangements of amino acids to carry out a particular chemical reaction or biological function. A common example can be seen where convergent evolution leads toward the formation of catalytic triads that share no common evolutionary history and in such cases the resultant 3D arrangement to form the triad is a requirement for catalysis to occur. In order to execute the analysis of 3D structures for such arrangements, the tools employed will have to extend beyond sequence and fold level comparisons to the level of specific 3D formations of the amino acid residues.

The availability of information regarding specific interacting residues for different modes of protein functions and interactions such as catalytic sites (Porter *et al.*, 2004), protein-DNA (Luscombe *et al.*, 2001; Selvaraj *et al.*, 2002; Angarica *et al.*, 2008; Contreras-Moreira, 2010), protein-RNA (Jones *et al.*, 2001), protein-protein (Raih *et al.*, 2005; Kinjo and Nakamura, 2010), protein-carbohydrate (Malik *et al.*, 2010) as well as other protein-ligand interactions and sites that are potential interaction surfaces such as clefts (Laskowski *et al.*, 1996), can be utilized as a database of known motifs to search against. The SIFTS database is an excellent example of a resource containing cross-referenced datasets that include residue level mappings which can in turn be deployed for use as a motifs database (Velankar *et al.*, 2013).

By integrating computational tools that are able to search at the sub-structure level or for specific residue arrangements with the prior knowledge on the function of similarly arranged residues, the user of such tools may be able to pin-point a possible line of enquiry for further investigation that may lead towards successful functional characterization of a protein of unknown function. In this article, computational methods of searching or comparing 3D level arrangements of amino acids are compared, and the approaches as well as general strategies for their use are discussed.

Background

Fold Comparisons Versus Amino Acid 3D Motif Searching

It is well known that in homologous proteins, divergence of sequence identity to even less than 30% can still result in the maintenance of a similar general fold (Chothia and Lesk, 1986). This has led to the development of numerous methods and databases specifically aimed at classifying and comparing the similarities in folding for protein 3D structures. Methods that perform structural alignments such as DALI (Holm *et al.*, 2006), FATCAT (Ye and Godzik, 2004) and VAST (Gibrat *et al.*, 1996) have long been frontline tools for the annotation of protein structures. Such fold comparison methods have proven useful even in the absence of any significant sequence similarity as demonstrated by Artymiuk *et al.* (1997) who reported that the domain folding of a then recently solved structure of adenylyl cyclase resembled the 'palm' domain of the polymerase I family of prokaryotic DNA polymerases.

However, in cases of convergent evolution, the absence of not only detectable sequence similarity, but also fold similarity, can be observed in proteins that execute generally similar mechanisms in order to carry out their functions. For proteins that have sites sharing a similar mechanistic function with others, but are unrelated in other parts of the structure, there may exist clusters of amino acids that are similarly arranged despite undetectable sequence similarity and different folding. Such similarity in 3D arrangements of amino acids in evolutionarily unrelated proteins can result in similar chemical activity or molecular function such as catalytic sites, ligand binding sites and various nucleic acid binding sites. In order to enable a structure to be annotated or

compared for such similar 3D amino acid constellations, the methods need to be able to accept a query that can be compared against 3D structure data such as those in the PDB. By developing methods that can search for such similarities, we would then be able to associate hypothetical proteins to characterized proteins as well as generate an inventory of conserved side chain arrangements and the functions that they serve.

In general, there are two approaches that can be taken to carry out searches for such tertiary motifs. One approach would be to enable the search engine or comparison program to accept an input file containing 3D structural information such as a PDB coordinate file and using that input as a query to search against a database of 3D motifs or a database of 3D structures. Another approach would be to allow a user to define or design the query that is representative of a 3D motif and using that input as a query to search against a database of 3D structures. There are now a number of programs that are able to carry out such searches and although they subscribe to the two general approaches mentioned above, the implemented algorithms between these available programs differ. In many cases, the different programs available complement each other by filling in the gaps in terms of search methodology, coverage and computational capacity. In this paper, only computer programs that carry out these types of 3D motif searches that are also available as web services and using PDB associated data are discussed (Table 1).

Fold and Sequence Independent Motifs in Protein Structures

Advances in macromolecular crystallography and high throughput molecular biology have led to numerous structural genomics projects. In Chothia (1992) reported that the number of different families was finite and may be as limited as one thousand distinct protein shapes. Therefore, as the PDB expands, it is most likely that the number of novel folds reported will not increase dramatically despite the contributions by the depositions from the structural genomics initiatives. This slow-down in the rate of growth for novel protein structural data in the PDB had been reported by Levitt (2007). As a result of the structural genomics contributions and the decrease in novel folds being discovered, concerted efforts are now being directed by many structural biology groups to target the solution of structures for proteins with unknown functions, specifically proteins that were annotated

Table 1 List of computational programs for 3D motif search

<i>Input format</i>	<i>Query type</i>	<i>Program/service</i>	<i>Search approach</i>
Input file format: PDB	Complete protein structure	COFACTOR	Identify proteins of similar fold and functional sites through global and local structural similarities (Roy <i>et al.</i> , 2012).
		eF-Seek	Search for similar binding sites in PDB based on molecular surfaces and calculation of electrostatic potentials (Kinoshita <i>et al.</i> , 2007).
		GIRAF	Identification of ligand binding sites in the PDB that are structurally similar to substructures of the query (Kinjo and Nakamura, 2007).
		MetaPocket 2.0	Identification of binding cavities using multiple computational platforms (Zhang <i>et al.</i> , 2011).
		MultiBind	Recognize common spatial patterns through multiple structural alignment of binding sites from a set of bound structures (Shulman-Peleg <i>et al.</i> , 2008).
		ProFunc	Identification of probable active sites and possible homologues in the PDB (Laskowski <i>et al.</i> , 2005).
		SPRITE	Convert a protein structure into representative pseudo-atom vectors for searching against a database of pseudo-atom vectors representing known 3D motifs and sites (Nadzirin <i>et al.</i> , 2012).
		ProBis	Compare protein surfaces and recognizes geometrically similar regions (Konc and Janezic, 2010).
		SA-Mot	Identification of structural motif from protein loop structures using structural alphabet HMM-SA (Regad <i>et al.</i> , 2010).
	Complete protein structure/ 3D motif	SuMo	Find similarity in 3D structure or substructures of proteins against ligand binding site in PDB through heuristic search of chemical group triplets (Jambon <i>et al.</i> , 2005).
		PINTS	Find recurring three-dimensional patterns of amino acids common to two sets structures by measuring the inter-residue distances (Stark <i>et al.</i> , 2003).
	3D motif	ASSAM	Convert a PDB coordinate file representing a 3D motif or arrangement into representative pseudo-atom vectors for searching against a database of pseudo-atom vectors representing structures in the PDB (Nadzirin <i>et al.</i> , 2012).
		RASMOT-3D PRO	Search a PDB file for secondary structure independent patterns or larger super secondary structure residue arrangements (Debret <i>et al.</i> , 2009).
Other	Other	IMAAAGINE	Search a user provided residue arrangement based on the distances between residues or residue types (Nadzirin <i>et al.</i> , 2013).
		PDBeMotif	Search possible binding sites and structural motifs of a protein structure, or all instances of residue pattern in the PDB (Golovin and Henrick, 2008).

Note: A listing of the fifteen web servers discussed that can search for specific 3D arrangements of amino acid residues.

as hypothetically encoding a protein that bears no sequence similarity to any proteins of known or characterized function (Eisenstein *et al.*, 2000; Cruz-Migoni *et al.*, 2011).

Once the structures of such proteins have been solved, they are typically subjected to fold similarity searching in order to determine whether the fold is already existent in the PDB. In cases where fold similarity can be attributed, there may likely be significant insights gleaned from the structure to influence the direction of further study. Nevertheless, there are numerous other protein structures that remain annotated as ‘uncharacterized’ in the PDB. A survey of the proteins annotated as ‘uncharacterized’ in the PDB revealed that there were 1084 proteins that had no similarity to any known sequence or folds at the time of the analysis as of May 2012 (Nadzirin and Firdaus-Raih, 2012). When both sequence and fold similarity options have been ruled out for inferring homologous function for a newly solved structure, an alternative option would be to investigate whether conserved 3D motifs that constitute catalytic and binding sites could be identified despite the absence of detectable sequence and fold similarity. The programs ProFunc (Laskowski *et al.*, 2005), GIRAF (Kinjo and Nakamura, 2009), PINTS (Stark *et al.*, 2003), SA-Mot (Regad *et al.*, 2010) and SPRITE (Nadzirin *et al.*, 2012) (Table 1) are capable of identifying such 3D motifs. These programs approach a search using different methodologies and at present there is no single all capable program. Instead, they can be used collectively to complement each other where a gap in the capacity of one program is taken up by other programs. Another factor for consideration is the fact that these programs search different databases. As a result, a user may need to run multiple searches to attain coverage of all the relevant databases used and thus increasing the possibility of retrieving a significant hit.

Systems and/or Applications

In this article, we assess and compare the utility of fifteen different computer programs for structural analysis (Table 1), particularly those that can identify specific 3D arrangements of the amino acid residues and are also accessible to users as a web service. We also evaluate their accuracy in different scenarios and present a strategy for the users on how to potentially analyse structures with uncharacterized functions via these tools. Undoubtedly, some programs may even perform better than the ones we have evaluated here, however, if they are not freely available and accessible via a web interface, they were not incorporated into this article.

Input Formats

With the exception of the IMAAAGINE server, all the other services accept a PDB formatted coordinate file as input. Both the IMAAAGINE and PDBeMotif servers allow for a user-defined pattern of amino acids provided as input via the web interface. The IMAAAGINE server requires that the user conceptualize an amino acid arrangement that is defined by distances between the residues in the pattern. The PDBeMotif server additionally provides comprehensive analyses of sequence, 3D motifs, and the ligand environment that can be provided as input via the web interface.

Output and Average Time for Computations

The fifteen servers assessed typically take from approximately a minute up to several hours for a calculation involving a protein structure that we had used as a test input. However, this rather large variation was expected due to the calculation times depending on factors that include, but are not limited to input size, the number of jobs being computed or queued on the servers and the different algorithms employed by the services. Most servers returned a tabulated summary of hits ranked by local structural similarity scores and downloadable results, while some servers also facilitated or allowed for the visualization of matched 3D arrangements (Table 2).

Analysis and Assessment

Searching Motifs/Sites Databases With a Structure Query

In the case of many protein structures that have no detectable sequence similarity, a fold similarity search using DALI may be sufficient for identifying structural homology. However, even in such cases, comparing specific clusters of amino acids within the structure can yield further insights regarding the specific activity or interaction partners of the protein. A simple step by step strategy that can be adopted after an initial fold level similarity search would be to submit the structure of interest to either selected or all of the services listed in Table 1 that are able to process a complete PDB structure as a search query. In selecting specific services, one should consider the capacities and database scope for the programs used. For example, the database attached to SPRITE is drawn from motifs in the catalytic site atlas (CSA) (Porter *et al.*, 2004), the ProCarb dataset (Malik *et al.*, 2010), the 3D-Footprint dataset and those manually extracted from literature. The ProBis service works by comparing protein surfaces and identifying geometrically similar regions thus making it less dependent on requiring prior knowledge regarding a motif such as SPRITE (Konc and Janezic, 2010) (Table 3).

When using predictive and/or computational methods, it is generally prudent to use several programs that employ different approaches by selecting representatives or highly used programs from each methodology class. The results from these various programs can then be compared and in cases where there is a clear agreement, those particular results can be further investigated. Nevertheless, the problem

Table 2 Comparison of different computational programs for detection of 3D motif

Program/service name															
COFACTOR	Et-Seek	GIRAF	MetaPocket	MultiBind	ProFunc	SPRITE	ProBis	SA-Mot	SuMo	PINTS	ASSAM	RASMOT-3D Pro	IMAAAGINE	PDBeMotif	
Input query															
✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓			✓	
PDB ID or coordinate file of a complete protein structure in PDB format															
Coordinate file of 3D pattern consists of several amino acids in PDB format															
A set of PDB IDs or coordinate files of bound protein structures in PDB format															
User-defined template of residue arrangement or combination of multiple queries															
Approaches															
✓					✓									✓	
Implements sequence and fold-dependent methods															
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Implements structure-based methods															
✓					✓										
Searches for global structural similarity															
✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Searches for local structural similarity															
Searches for pairwise structural similarity															
Searches/predicts for:															
✓					✓										
Homologous structures															
✓						✓									
Catalytic sites															
✓	✓	✓	✓	✓	✓	✓	✓		✓	✓				✓	
Ligand binding sites															
DNA/RNA-binding sites															
✓		✓				✓	✓	✓							
Protein-protein interfaces															
Pocket sites on protein surfaces															
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
3D motifs															
Output type/format															
✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
List of predicted 3D motifs ranked by structural similarity scores															
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Direct molecular visualization of predicted 3D motifs enabled															
✓	✓	✓	✓	✓		✓	✓		✓		✓	✓			
Downloadable output files															

Note: A comparison of the fifteen web servers for 3D motif searching in terms of their input, algorithm and output.

Table 3 Relevant hits for query of BPSL1549 protein

Prediction	Server	Results
Homologs	COFACTOR	Returned 10 chain homologs from fold comparison, which includes CNF1-C, and several uncharacterized proteins. Gene Ontology (GO) term prediction for Biological Process (BP) resulted with 'arginine biosynthetic process'.
	ProFunc	Returned 10 PDB chains with similar fold, including chemotaxis deamidase and several uncharacterized proteins. The server predicts possible function of BPSL1549 in the regulation of helicase activity and molecular function, based on GO prediction for Biological Process (BP).
3D motif	MetaPocket 2.0	Resulted with top 3 pocket sites clustered from sites generated by 8 prediction tools, where each consists of large number of residues.
	ProBis	Returned 100 similar proteins with aligned residues from structural superposition, which include a putative ABC transporter (PDB chain: 4pe6A) with 11 aligned residues. The server predicted putative ligands of BPSL1549 protein for small molecules, proteins and ions. The server also predicted a binding site for Bromide ion, similarly reported by PDBeMotif.
	COFACTOR	Predicted PDB chains with similar binding sites for adenosine monophosphate, cerulenin, magnesium and sodium ions.
	GIRAF	Returned 100 structures with similar 3D templates, ranked by normalized GIRAF score, where top five hits consist of more than 10 aligned residues from structural superposition.
	PDBeMotif	The server reported matched small 3D motifs and a binding site for bromide ions within 5.0Å to the ligand.
	ProFunc	The server returned several hits from reverse template search that mapped to auto-generated templates from BPSL1549 protein itself, which include templates from several keap proteins.
	Ef-seek	Returned predicted ligand binding sites from matched PDB structures ranked by Z-score, which include binding sites for PDB ligand code of PEF, BCR, STE, and LIL.
	SA-mot	Predicted a functional candidate motif consisting of a fragment of seven residues that are continuous in sequence.
	SPRITE	Returned proteins with similar 3D template as BPSL1549 ranked by RMSD value. The hits include a chemotaxis methylation protein (2f9zD) where a residue arrangement of Thr88, Cys94, and His106 from BPSL1549 protein matches Thr21, Cys27, and His44 residues from the chemotaxis methylation protein.
	SuMo	Returned matching ligand binding sites sorted by site volume, which include binding sites for PDB ligand codes of ACT, PO4, MLI, SF4, and GLC.

Note: A listing of the results returned by the different services when the BPSL1549/3tu8 protein structure was used as a query to identify 3D motifs present in the structure.

should not be approached using a one size fits all solution and as such, other results may also warrant further investigation. For example, if two seemingly non-homologous structures that share no detectable sequence identity were to have 3D motifs that could be superimposed after using a 3D site search program such as SPRITE, PROBis and PINTS, their sequences can be manually aligned using the superposed arrangement as a reference point. The decision to proceed further with a particular motif can take into account a number of factors that are determined by the user depending on their own interests and motivation. The residues that match to sites from a database can be further extracted and submitted as a query for searching against a database of PDB structures. Computer programs such as RASMOT-3D Pro (Debret *et al.*, 2009) and ASSAM (Nadzirin *et al.*, 2012) are able to execute such queries.

Searching a Structure Database Using a 3D Motif Query

The obvious limitation of using a complete structure to query a motif database is the requirement of prior knowledge for the motifs while the clear advantage would be that such motifs would probably be well characterized with perhaps attached experimental annotations. Although it would be useful to be able to annotate a newly solved protein structure for known sites, there are situations where no prior knowledge exists of a motif that can be found for that query. In such a scenario, a visual examination or surface analysis of the structure may provide incipient information on potentially functional residues. Once the residues of interest have been identified, they can then be provided as queries to search for structures where similar arrangements occur. This approach does not require these sites to have been previously identified or annotated. If similar arrangements are found, contextual references and comparisons can then be made for the structures where the arrangements are present. In comparison to methods that use a whole structure input, there are fewer programs that are able to receive a pattern input for searching against a database of 3D structures. Two such services that are available on the web are ASSAM (Nadzirin *et al.*, 2012) and RASMOT-3D Pro (Debret *et al.*, 2009).

Although the RASMOT-3D Pro (Debret *et al.*, 2009) and ASSAM (Nadzirin *et al.*, 2012) programs essentially carry out the same type of search, where a motif in PDB format can be submitted as a query, the representation system used for the amino acids in 3D space are different. In the RASMOT-3D Pro program, the C α and C β positions are used to represent a residue (Debret *et al.*, 2009); while in ASSAM, the side chain positions are represented using pseudo-atom vectors (Nadzirin *et al.*, 2012). This has been demonstrated to result in a different retrieval list because an overlapping side chain position may have extended from C α atoms

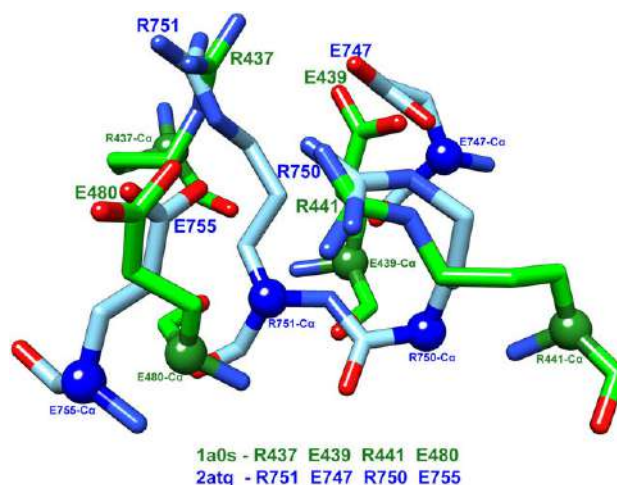


Fig. 1 Superposition of Arg-Glu-Arg-Glu patterns found in two different protein structures. The superposition of an Arg-Glu-Arg-Glu pattern consisting of two arginine and two glutamic acid from a sucrose-specific porin (PDBID: 1a0s) (green) and a DNA polymerase (PDBID: 2atq) (blue) where the side chains overlap although the C α atoms do not. The atoms are shown in ball-and-stick representation, C α atoms as spheres, and all other atoms in stick representation.

that do not overlap (Nadzirin *et al.*, 2012) (Fig. 1). Therefore, a side chain based representation can be useful in cases of convergent evolution where the positions of the side chains remain conserved in 3D space in order to effect a specific chemical outcome such as catalysis. RASMOT-3D Pro offers the user an option of searching for the query motif in a set of up to 10 user provided structures; ASSAM only allows for pre-set database options.

Discovery of Novel Motifs Present Only in a Structural Context

With the exception of PDBeMotif (Golovin and Henrick, 2008) and IMAAAGINE (Nadzirin *et al.*, 2013), other programs listed in Table 1 have a limitation where they require prior knowledge of the 3D motifs available for searching against, or the coordinates of a specific 3D arrangement to be provided as a search query. PDBeMotif and IMAAAGINE can carry out PDB searches using a query where the user can determine the type of amino acid and specify the distances between them without knowing how they are exactly arranged in 3D space. This overcomes the need for the user to define a specific pattern such as for several of the other previously listed programs in Table 1. Due to the use of pre-computed distance information within each single protein chain in a PDB file for its database, PDBeMotif only finds sub-structures if all the amino acids are in the same chain.

The IMAAAGINE program was engineered to carry out searches for a query that was designed by the user although it does not provide extensive key word querying capabilities such as in PDBeMotif. The IMAAAGINE queries can be very generic in nature such as: “are there arrangements where the side chain of an aspartic acid is surrounded by four hydrophobic residues?” (Fig. 2). The advantages of such a search capability are: (i) the ability to search for conserved 3D arrangements that are novel; and (ii) the ability to identify motifs present only in a structural context that are not detectable at the sequence level. This differs from the capability provided by other programs, such as RASMOT-3D Pro, ASSAM and SPRITE, that search a database of PDB structures for motifs. In these programs, prior knowledge of the motif is required to be provided as a PDB formatted input. However, in the case of IMAAAGINE, the user can approach a query based on a biochemical understanding of how a cluster of amino acids can be theoretically arranged in order to achieve a certain function. Since no prior knowledge is required regarding an exact arrangement of the amino acids to be used as a query, this allows for a broader search that may in turn retrieve hits that were not originally intended by the user.

Due to its flexible query input, the IMAAAGINE program has the capacity to retrieve highly conserved arrangements that may have been below the radar of standard structural analysis and annotation approaches. Furthermore, an IMAAAGINE search is not restricted by the location of residues to a particular chain or monomer in a multimeric assembly since it considers only the relative positions of a side chain representation to others based on calculated distances. The results of an IMAAAGINE search can therefore uncover arrangements, that if found to be highly conserved, can then be considered a novel 3D motif. Such motifs may prove to be functionally relevant although its conservation is identifiable only from a structural context and can therefore be not obvious from a purely sequence analysis approach. In the next section, case studies are presented that compare and illustrate the differences of the webserver discussed.

Illustrative Examples and Case Studies

Detection of 3D Motif in Protein Structure – Possible Functional Annotation of a Hypothetical Protein

The protein sequence encoded by the ORF *bps1549*, from the genome of the pathogen *Burkholderia pseudomallei*, was originally annotated as a hypothetical protein (Holden *et al.*, 2004) with no detectable sequence similarity to any sequences other than

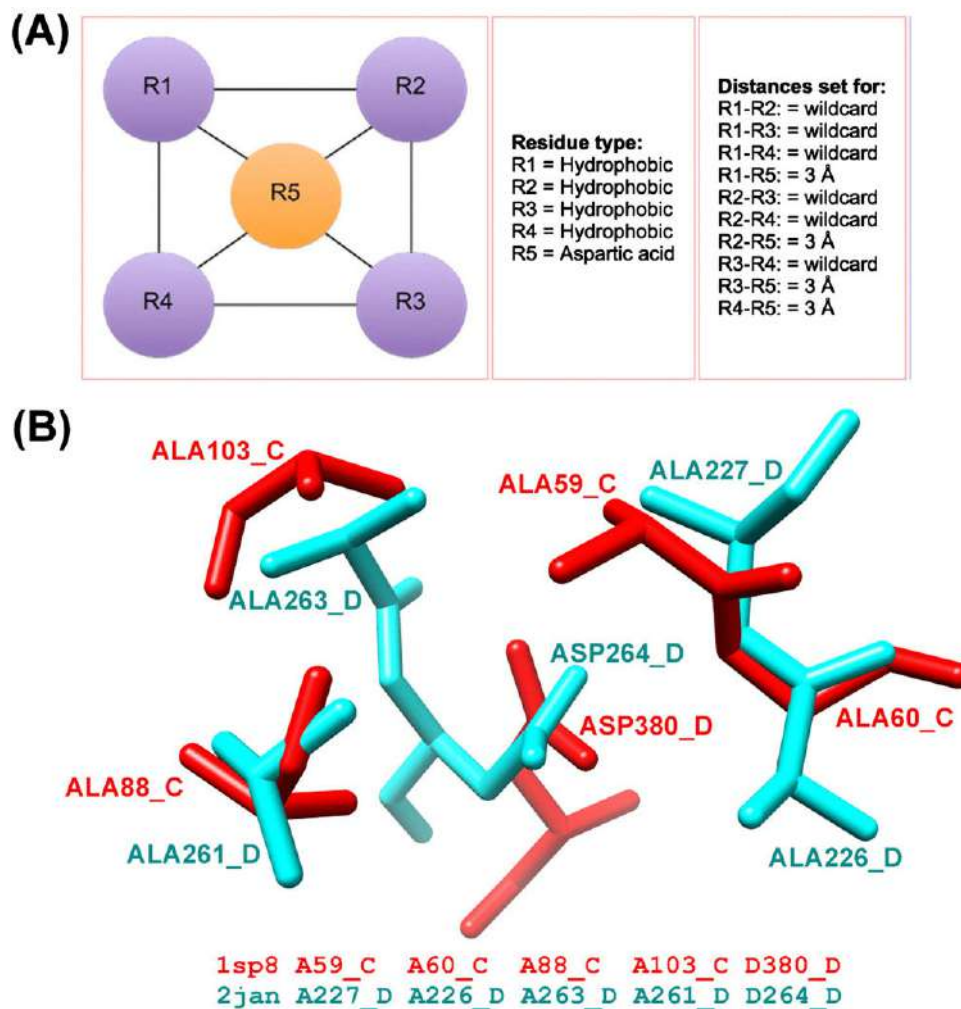


Fig. 2 An example of an IMAAAGINE query for a user-defined arrangement of amino acids. (A) This IMAAAGINE input example shows a search for an aspartic acid (R5) surrounded by four hydrophobic residues within the distance of 3.0 Å (R1-R4). (B) The retrieved overlapping side chains for the search is shown from the superposition of 3D patterns consisting of four alanine surrounding an aspartic acid from a 4-hydroxyphenylpyruvate dioxygenase (PDBID: 1sp8) and a tyrosyl-tRNA synthetase (PDBID: 2jan).

obvious homologs of itself within the Burkholderia genus. The crystallographic structure for BPSL1549 was then acquired (Cruz-Migoni *et al.*, 2011) (Fig. 3(A)) and fold comparisons using DALI revealed fold similarity to the catalytic domain of cytotoxic necrotizing factor 1 (CNF1-C) from *Escherichia coli* (PDB ID: 1hq0), with a Z-score of 9.5 and a chemotaxis deamidase from *Thermotoga maritima* (PDB ID: 2f9z) with a Z-score of 5.9 (Fig. 3(B)).

The structure of BPSL1549 (PDB ID: 3tu8) consists of a beta-sandwich fold surrounded by α -helices and loops (Fig. 3(A)) as similarly observed in the catalytic domain of CNF1-C (Cruz-Migoni *et al.*, 2011), an enzyme catalyzing glutamine deamidation that results in the alteration of the host cell actin cytoskeleton. This provided a clue that BPSL1549 may be involved in a similar activity and was compounded by the fact that a SPRITE search identified similarly arranged residues consisting of Thr88, Cys94 and His106 in BPSL1549 that matched the CNF1-C catalytic triad consisting of Val833, Cys866 and His881, where the cysteine and histidine are conserved among such toxins and required for deamidation of glutamine (Buetow *et al.*, 2001; Cruz-Migoni *et al.*, 2011) (Fig. 3(C)). The SPRITE search also returned a similar arrangement of Thr-Cys-His shared with a chemotaxis deamidase from *T. maritima* (PDB ID: 2f9z) at an RMSD value of 0.49, thus signifying the possible role of BPSL1549 and the importance of such arrangement in glutamine deamidation. A search on SuMo web server on the other hand, returned a similar arrangement of Thr-His-Val shared between BLF1 and D-xylose isomerase from *Streptomyces rubiginosus* (PDB IDs: 3kbn and 1xic) (Kovalevsky *et al.*, 2011; Carrell *et al.*, 1994).

COFACTOR and ProFunc were used to search for possible homologs of BPSL1549 and both servers returned a different set of similar structures and GO terms prediction. The COFACTOR server proposed a possible role for BPSL1549 in glutamate N-acetyltransferase activity for Molecular Function (MF) and arginine biosynthesis for Biological Process (BP) prediction, while ProFunc predicted the function of BLF1 to be involved in regulation of helicase activity and molecular function for Biological Process (BP) prediction. The information from different servers can prove useful collectively (Fig. 3(D)) as it was ultimately proven that BPSL1549 is a deamidase that converts Gln to Glu on its target molecule, the translation initiation factor eIF4A which has a

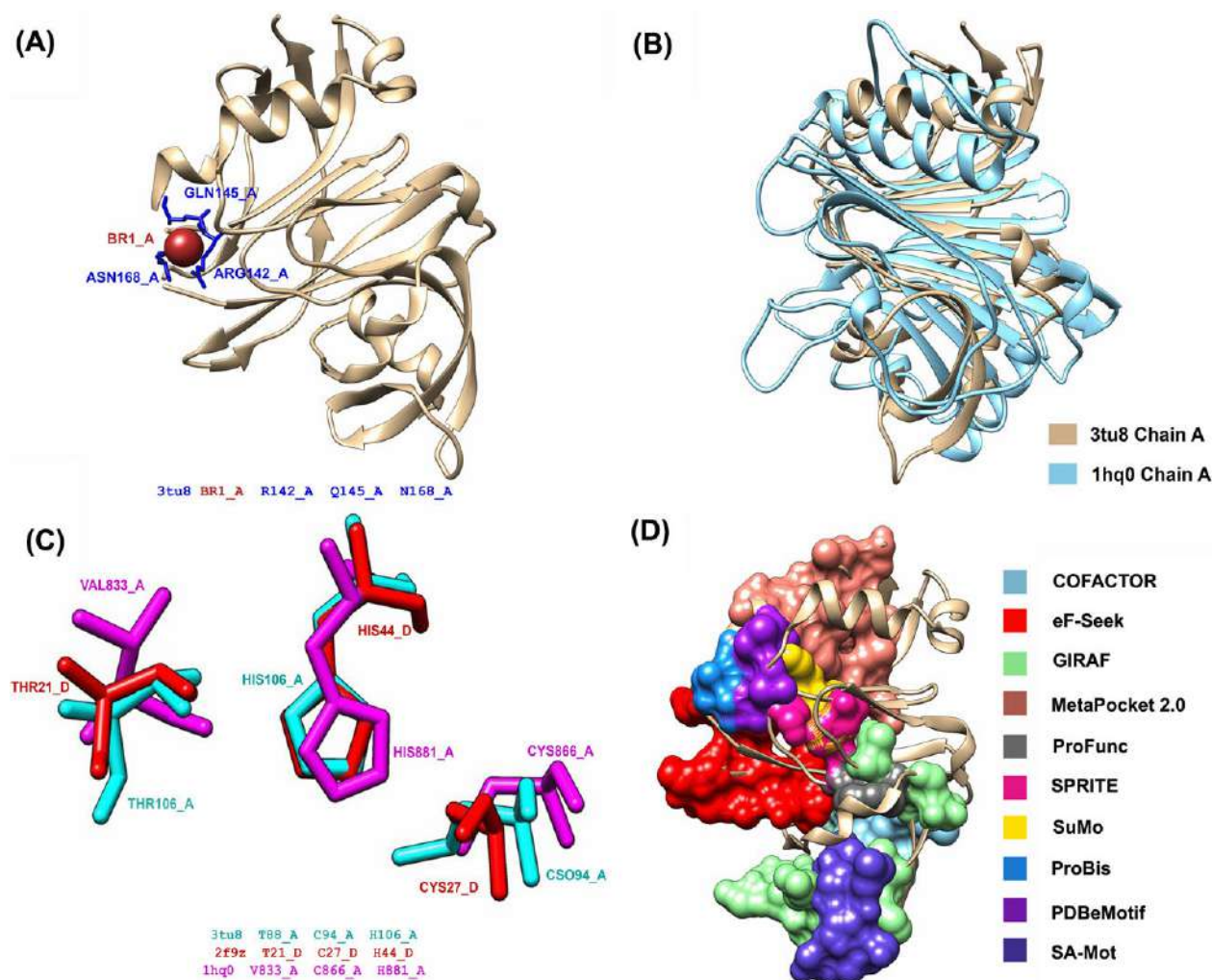


Fig. 3 Detection of 3D motif using various 3D motif searching methods. (A) Ribbon representation of a crystal structure of BPSL1549 from *Burkholderia pseudomallei* (PDBID: 3tu8). The residues binding the bromide ion (in red) are displayed as blue sticks. (B) The BPSL1549 (PDBID: 3tu8) in tan color and the CNF1-C (PDBID: 1hq0) in light blue color are shown in ribbon representation, after the superposition. (C) Superposition of catalytic triad residues from chemotaxis deamidase (PDBID: 2f9z) (red) and CNF1-C (PDBID: 1hq0) (magenta) to BPSL1549 (PDBID: 3tu8) (cyan). (D) A graphical representation displaying the BPSL1549/3tu8 structure with the output from various web servers (in different colors) mapped onto it in molecular surface representation.

helicase function (Cruz-Migoni *et al.*, 2011). This resulted in the renaming of BPSL1549 to *Burkholderia* lethal factor 1 (BLF1) due to it being the first confirmed toxin for *B. pseudomallei*. Similar residues for the bromine ligand binding site was detected by ProBis and PDBeMotif – Gln145, Arg142, and Asn168 with no other protein sharing a similar residue arrangement. Table 2 summarizes the findings when BLF1 (PDB ID: 3tu8) was used as the search query.

Detection of Arg-Glu-Arg-Glu Query Motif in Database of PDB Structures

It is also a common approach to visualize a structure using the various molecular graphics software available in order to glean some insights into the function of a newly solved structure. Through such visual inspections, specific residue arrangements of interest can be identified – for example arrangements that may have potential as a catalytic site or ligand binding site. It then becomes useful to identify whether such similar residue arrangements exist in other structures and whether there are known functions associated with them. To demonstrate this, a four residue arrangement consisting of Arg, Glu, Arg, Glu (RERE – Fig. 4(A)) from a *Salmonella typhimurium* sucrose-specific porin (PDB ID: 1a0s) was extracted from the main PDB file and submitted as an ASSAM search. The 1a0s structure was classified as a LamB porin domain (PFAM accession: PF02264) in PFAM and contains 16 antiparallel strands with several binding sites for glycosyl groups along the channel involved in sugar transport (Forst *et al.*, 1998) (Fig. 4(A)).

The ASSAM search retrieved a RERE arrangement in a fusion protein consisting of a DNA polymerase with a single-stranded DNA binding protein (PDB ID: 2atq; Fig. 4(B)). The R437, E439, R441, E480 side-chains from 1a0s are similarly arranged spatially with R751, E747, R750 E755 from 2atq (Fig. 4). The 2atq structure consists of an N-terminal domain that allows binding of T4/

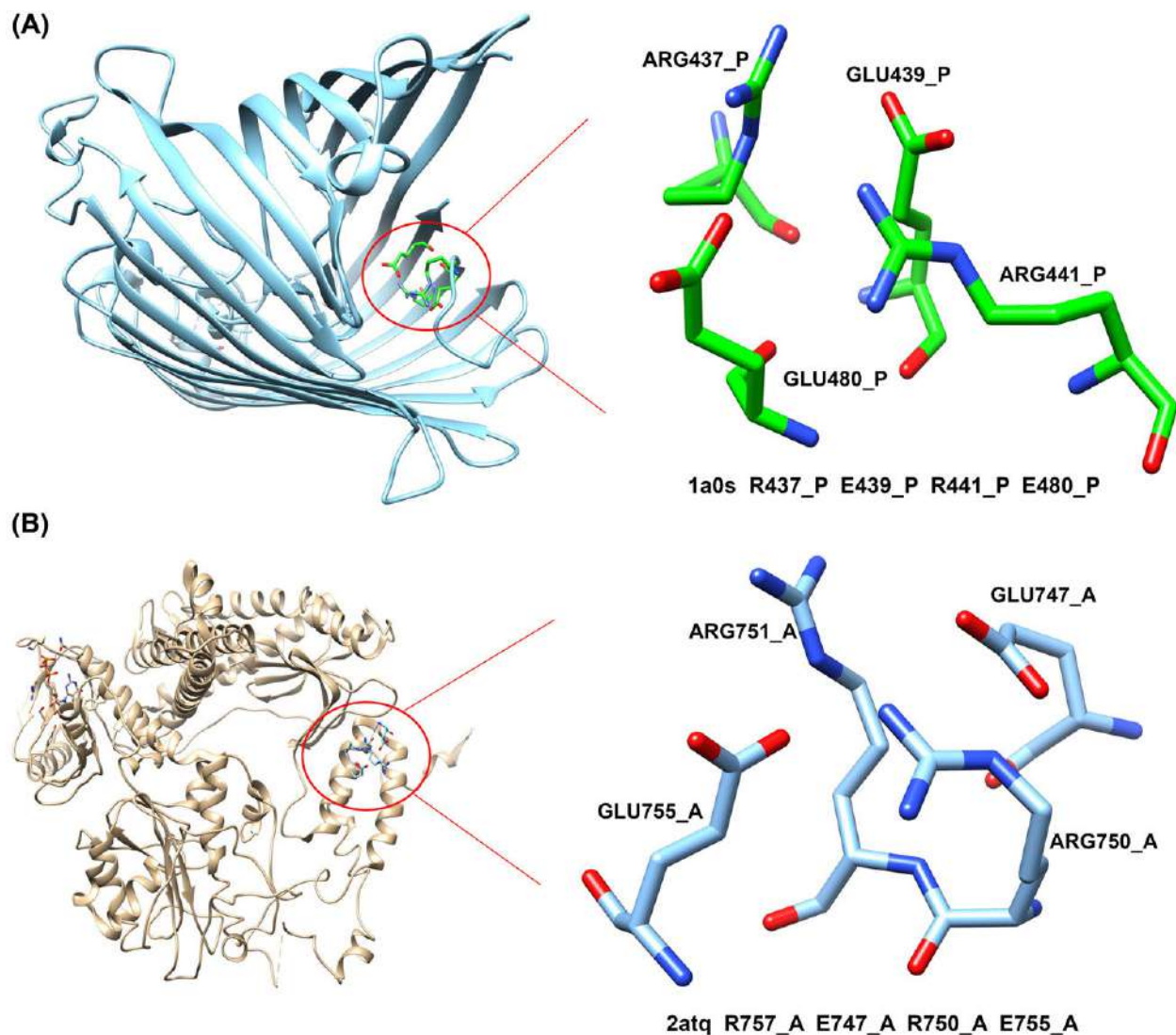


Fig. 4 Detection of Arg-Glu-Arg-Glu patterns in distinct protein structures. (A) The pattern consisting of two arginine and two glutamic acid (RERE) in a sucrose-specific porin (PDBID: 1a0s) is depicted as sticks (green) and magnified on the right panel. (B) The same RERE arrangement in a DNA-polymerase (PDBID: 2atq) depicted as sticks (light blue) and magnified in the right panel.

RB69 single-stranded DNA binding protein to single-stranded DNA, a C-terminal domain that mediates the interaction with accessory proteins and a DNA-binding domain that binds to single-stranded DNA (Sun *et al.*, 2006) (Fig. 4(B)). It is clear that the two structures do not share any similar sequence or fold despite the fact that these four side-chains form a restricted spatial arrangement present in both structures that can be 3-dimensionally superimposed (Fig. 1). From the superpositions, it is evident that the overlap occurs at the side-chain positions and not at the alpha-carbon positions thus making them potentially significant in terms of the functional chemistry for both molecules. However, in cases when conservation is at the core back-bone level with variations in the spatial arrangements of the side-chain leading to slightly different chemistry – such as different target ligands for the site, then the RASMOT-3D Pro server would be a more suitable option to search with.

Discussions and Future Directions

The capacity to infer function by associating the chemical capacity or functional potential of 3D motifs is an important step towards the computational annotation and ultimate functional characterization of the many structures of proteins with uncharacterized functions. Nevertheless, most computational function annotation is still carried out at the primary structure or amino acid sequence level. While many of the programs discussed in this article may be able to make associations between conserved 3D amino acid arrangements in proteins of unrelated sequences and folds, there is still a need for novel 3D motifs to be identified using mainstream sequence analysis, especially 3D motifs that are functionally relevant only in a structural context.

As an example, a 3D interface between three chains of the same sequence that consists of a single residue, each at the interface, such as histidine, can be considered a three histidine tertiary motif. However, such a 3D motif will only be represented by a single residue and position in the sequence. This therefore necessitates the development of programs that are able to bridge the gap and identify such single residue motifs perhaps via machine learning approaches using the physico-chemical profiles of the surrounding residues once they are identified as structurally contextual motifs. The information from such investigations can therefore be incorporated back into sequence level analysis thus allowing for accurate identification of motifs that are relevant only in a structural context.

Closing Remarks

The ability to search for similar 3D arrangements of amino acids is of great utility in the post-genomics era, especially in light of the many structural genomics initiatives. However, there is no clear one-stop service or one size fits all solution at present. Despite this, expert use of the available tools and services have been proven as useful means of providing insights towards a line of enquiry that can ultimately enable the functions of uncharacterized proteins to be successfully determined.

Acknowledgement

MFR is funded by the Ministry of Science, Technology and Innovation grant 02-01-02-SF1278.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Natural Language Processing Approaches in Bioinformatics. Study of The Variability of The Native Protein Structure

References

- Angarica, V., Perez, A., Vasconcelos, A., Collado-Vides, J., Contreras-Moreira, B., 2008. Prediction of TF target sites based on atomistic models of protein-DNA complexes. *BMC Bioinform.* 9, 436.
- Artymiuk, P.J., Poirrette, A.R., Rice, D.W., Willett, P., 1997. A polymerase I palm in adenyl cyclase? *Nature* 388, 33–34.
- Buetow, L., Flatau, G., Chiu, K., Boquet, P., Ghosh, P., 2001. Structure of the Rho-activating domain of Escherichia coli cytotoxic necrotizing factor 1. *Nat. Struct. Mol. Biol.* 8, 584–588.
- Carrell, H.L., Hoier, H., Glusker, J.P., 1994. Modes of binding substrates and their analogues to the enzyme D-xylose isomerase. *Acta Crystallogr. D Biol. Crystallogr.* 50, 113–123.
- Chothia, C., 1992. Proteins. One thousand families for the molecular biologist. *Nature* 357, 543–544.
- Chothia, C., Lesk, A.M., 1986. The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5, 823–826.
- Contreras-Moreira, B., 2010. 3D-footprint: A database for the structural analysis of protein – DNA complexes. *Nucleic Acids Res.* 38, D91–D97.
- Cruz-Migoni, A., Hautbergue, G.M., Artymiuk, P.J., *et al.*, 2011. A Burkholderia pseudomallei toxin inhibits helicase activity of translation factor eIF4A. *Science* 334, 821–824.
- Debret, G., Martel, A., Cuniasse, P., 2009. RASMOT-3D PRO: A 3D motif search webserver. *Nucleic Acids Res.* 37, W459–W464.
- Eisenstein, E., Gilliland, G.L., Herzberg, O., *et al.*, 2000. Biological function made crystal clear – Annotation of hypothetical proteins via structural genomics. *Curr. Opin. Biotechnol.* 11, 25–30.
- Forst, D., Welte, W., Wacker, T., Diederichs, K., 1998. Structure of the sucrose-specific porin ScrY from Salmonella typhimurium and its complex with sucrose. *Nat. Struct. Mol. Biol.* 5, 37–46.
- Gibrat, J.-F., Madej, T., Bryant, S.H., 1996. Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.* 6, 377–385.
- Golovin, A., Henrick, K., 2008. MSDmotif: Exploring protein sites and motifs. *BMC Bioinform.* 9, 312.
- Holden, M.T., Titball, R.W., Peacock, S.J., *et al.*, 2004. Genomic plasticity of the causative agent of melioidosis, Burkholderia pseudomallei. *Proc. Natl. Acad. Sci. USA* 101, 14240–14245.
- Holm, L., Kaariainen, S., Wilton, C., Plewczynski, D., 2006. Using Dali for structural comparison of proteins. *Curr. Protoc. Bioinform.* (Chapter 55.5.1-5.5.24).
- Jambon, M., Andrieu, O., Combet, C., *et al.*, 2005. The SuMo server: 3D search for protein functional sites. *Bioinformatics* 21, 3929–3930.
- Jones, S., Daley, D.T.A., Luscombe, N.M., Berman, H.M., Thornton, J.M., 2001. Protein – RNA interactions: A structural analysis. *Nucleic Acids Res.* 29, 943–954.
- Kinjo, A.R., Nakamura, H., 2007. Similarity search for local protein structures at atomic resolution by exploiting a database management system. *Biophysics (Nagoya-shi)* 3, 75–84.
- Kinjo, A.R., Nakamura, H., 2009. Comprehensive structural classification of ligand-binding motifs in proteins. *Structure* 17, 234–246.
- Kinjo, A.R., Nakamura, H., 2010. Geometric similarities of protein–protein interfaces at atomic resolution are only observed within homologous families: An exhaustive structural classification study. *J. Mol. Biol.* 399, 526–540.
- Kinoshita, K., Murakami, Y., Nakamura, H., 2007. eF-seek: Prediction of the functional sites of proteins by searching for similar electrostatic potential and molecular surface shape. *Nucleic Acids Res.* 35, W398–W402.
- Konc, J., Janezic, D., 2010. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics* 26, 1160–1168.
- Kovalevsky, A.Y., Hanson, L., Fisher, S.Z., *et al.*, 2011. Metal ion roles and the movement of hydrogen during the reaction catalyzed by D-xylose isomerase: A joint X-ray and neutron diffraction study. *Structure* 18, 688–699.
- Laskowski, R.A., Luscombe, N.M., Swindells, M.B., Thornton, J.M., 1996. Protein clefts in molecular recognition and function. *Protein Sci.* 5, 2438–2452.
- Laskowski, R.A., Watson, J.D., Thornton, J.M., 2005. ProFunc: A server for predicting protein function from 3D structure. *Nucleic Acids Res.* 33, W89–W93.
- Levitt, M., 2007. Growth of novel protein structural data. *Proc. Natl. Acad. Sci. USA* 104, 3183–3188.
- Luscombe, N.M., Laskowski, R.A., Thornton, J.M., 2001. Amino acid–base interactions: A three-dimensional analysis of protein–DNA interactions at an atomic level. *Nucleic Acids Res.* 29, 2860–2874.

- Malik, A., Firoz, A., Jha, V., Ahmad, S., 2010. PROCARB: A database of known and modelled carbohydrate-binding protein structures with sequence-based prediction tools. *Adv. Bioinform.* 436036.
- Nadzirin, N., Firdaus-Raih, M., 2012. Proteins of unknown function in the Protein Data Bank (PDB): An inventory of true uncharacterized proteins and computational tools for their analysis. *Int. J. Mol. Sci.* 13, 12761–12772.
- Nadzirin, N., Gardiner, E.J., Willett, P., Artymiuk, P.J., Firdaus-Raih, M., 2012. SPRITE and ASSAM: Web servers for side chain 3D-motif searching in protein structures. *Nucleic Acids Res.* 40, W380–W386.
- Nadzirin, N., Willett, P., Artymiuk, P.J., Firdaus-Raih, M., 2013. IMAAAGINE: A webserver for searching hypothetical 3D amino acid side chain arrangements in the protein data bank. *Nucleic Acids Res.* 41, W432–W440.
- Porter, C.T., Bartlett, G.J., Thornton, J.M., 2004. The Catalytic Site Atlas: A resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res.* 32, D129–D133.
- Raih, M.F., Ahmad, S., Zheng, R., Mohamed, R., 2005. Solvent accessibility in native and isolated domain environments: General features and implications to interface predictability. *Biophys. Chem.* 114, 63–69.
- Regad, L., Martin, J., Nuel, G., Camproux, A.-C., 2010. Mining protein loops using a structural alphabet and statistical exceptionality. *BMC Bioinform.* 11, 75.
- Rose, P.W., Prlic, A., Altunkaya, A., *et al.*, 2017. The RCSB protein data bank: Integrative view of protein, gene and 3D structural information. *Nucleic Acids Res.* 45, D271–D281.
- Roy, A., Yang, J., Zhang, Y., 2012. COFACTOR: An accurate comparative algorithm for structure-based protein function annotation. *Nucleic Acids Res.* 40, W471–W477.
- Selvaraj, S., Kono, H., Sarai, A., 2002. Specificity of Protein – DNA Recognition Revealed by Structure-based Potentials: Symmetric/asymmetric and Cognate/Non-cognate Binding. *J. Mol. Biol.* 322, 907–915.
- Shulman-Peleg, A., Shatsky, M., Nussinov, R., Wolfson, H.J., 2008. MultiBind and MAPPIS: Webserver for multiple alignment of protein 3D-binding sites and their interactions. *Nucleic Acids Res.* 36, W260–W264.
- Stark, A., Sunyaev, S., Russell, R.B., 2003. A model for statistical significance of local similarities in structure. *J. Mol. Biol.* 326, 1307–1316.
- Sun, S., Geng, L., Shamoo, Y., 2006. Structure and enzymatic properties of a chimeric bacteriophage RB69 DNA polymerase and single-stranded DNA binding protein with increased processivity. *Proteins* 65, 231–238.
- Velankar, S., Dana, J.M., Jacobsen, J., *et al.*, 2013. SIFTS: Structure integration with function, taxonomy and sequences resource. *Nucleic Acids Res.* 41, D483–D489.
- Ye, Y., Godzik, A., 2004. FATCAT: A web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Res.* 32, W582–W585.
- Zhang, Z., Li, Y., Lin, B., Schroeder, M., Huang, B., 2011. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics.* 27, 2083–2088.

Computational Protein Engineering Approaches for Effective Design of New Molecules

Jaspreet Kaur Dhanjal, Vidhi Malik, Navaneethan Radhakrishnan, Moolchand Sagar, Anjani Kumari, and Durai Sundar, DBT-AIST International Laboratory for Advanced Biomedicine (DAILAB), Indian Institute of Technology Delhi, New Delhi, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteins are one of the most important building blocks of the body, and are comprised of 20 different amino acids. They display a wide variety of functions including structural, catalytic, sensory, motility, immune response, transport, and many more. Owing to such diverse functions, proteins have attracted a lot of attention in the fields of agriculture, therapeutics, and industry. Protein engineering, in general, involves either synthesis of new proteins or alteration of existing protein structure and sequence in order to achieve desired functions. It aims at improving catalytic properties, stability (thermal, pH, solvent), specificity, and yield of proteins, and finds wide applications in the fields of medicine, nanotechnology, biopolymer production, detergent, and food industry.

In the early 1980s, the advancement in the field of recombinant DNA technology enabled the engineering of proteins that was mediated via changes in the proteins' amino acid sequence (Ulmer, 1983). Subsequently several other methods were developed for engineering proteins including random mutagenesis, site directed mutagenesis, evolutionary methods, domain shuffling, and others. Subtilisins in detergent industry; tissue plasminogen activator (tPA) in heart attack and stroke treatment; Humalog, Novolog, and Apidra (engineered insulin) for diabetes treatment; and engineered amylases and lipases used in food industry are some of the examples of commercially available engineered proteins (Walsh, 2007; Turanli-Yildiz *et al.*, 2012). Despite being so promising, widespread application of protein engineering techniques was limited owing to labor-intensive and time-consuming protocols, and use of large amount of resources.

In the past few decades, a tremendous revolution in computational protein engineering has helped protein manipulation in overcoming the above mentioned barriers. Novel protein design algorithms, advancement in structural bioinformatics, molecular force fields, and availability of immense information regarding 3D protein structure in databases like Protein Data Bank have revolutionized the field of computational protein engineering. Computational protein designing broadly focuses on (1) generation of *de novo* amino acid sequences that can adopt stable 3D structure, or modification of the existing scaffold to improve the stability of the protein in non-native environment, and (2) evolving newer functions in proteins by harnessing information from known crystal structures (Alvizo and Mayo, 2008). A schematic representation of common computational protein engineering approaches is illustrated in Fig. 1. These include (1) *de novo* protein modeling, (2) rational designing of proteins based on existing structural templates, and (3) statistical modeling. *De novo* protein structure prediction methods make use of general protein folding principles and energetics (deduced from native structures) to assign tertiary structures to the sequence of amino acids. It is generally used for the sequences with no close homologs with experimentally solved structures. *De novo* methods involve vast computations and hence are so far limited to small proteins only. In rational designing approach, rather than attempting to start from scratch, the experimentally known protein structures can be utilized as templates to build on. This approach at least ensures the *in vivo* stability of the engineered proteins. Here, the rationale is to look for the part of protein that is less important for proper folding, and remodel it to introduce new functionality. Statistical modeling or the data-driven approaches are used to infer relationships between protein sequence, structure, and function from experimental data. These learning algorithms are based on the fact that the amino acids comprising the proteins encode their chemical properties and govern their biological functions. These statistical models are capable of capturing the global behavior of proteins, are not biased, and therefore can be used to make predictions for novel sequences.

Some of the commonly used computational algorithms and approaches for aiding in protein engineering include dead-end elimination (DEE), molecular dynamics simulations, Monte Carlo simulated annealing, and genetic algorithms (Park *et al.*, 2004; Floudas *et al.*, 2006). Many tools are available to computationally evaluate the effect of changes made in the protein structure. For example, SCHEMA is a tool that calculates DNA recombination mediated disruption in terms of the *E* value (where *E* is the average disruption of residue contact). The engineered protein with low *E* value is taken further to achieve functionally active modified protein that is less identical to the parent protein (Meyer *et al.*, 2003). Protein Sequence Activity Relationships (ProSAR) is another tool that can sort diverse mutations by predicting their effect on desired protein activity and thus providing mutants that are highly improved in comparison to the wild-type protein (Fox *et al.*, 2007). Rosetta, the most commonly used computational method, aims at creating new biocatalysts from existing template structure, taking into account an active site's quantum mechanical parameters and carrying out design around the reaction's transition state (Rothlisberger *et al.*, 2008; Der *et al.*, 2013). A few other computational protein design tools include FoldX, FlexPreD, and Iterative Protein Redesign and Optimization (IPRO) (Li *et al.*, 2012; Verma *et al.*, 2012; Saraf *et al.*, 2006). Table 1 lists some of these commonly used tools with their brief description.

A large number of successful stories of protein engineering are reported in the literature. The structures predicted by computational methods have been shown to have atomic level accuracy with respect to NMR structure of protein (Murphy *et al.*, 2015). Recently, a peptide design was reported that mimicked the N-trimer region of the GP2 fusion protein of the Ebola virus, which is a potential target for drugs (Clinton *et al.*, 2015).

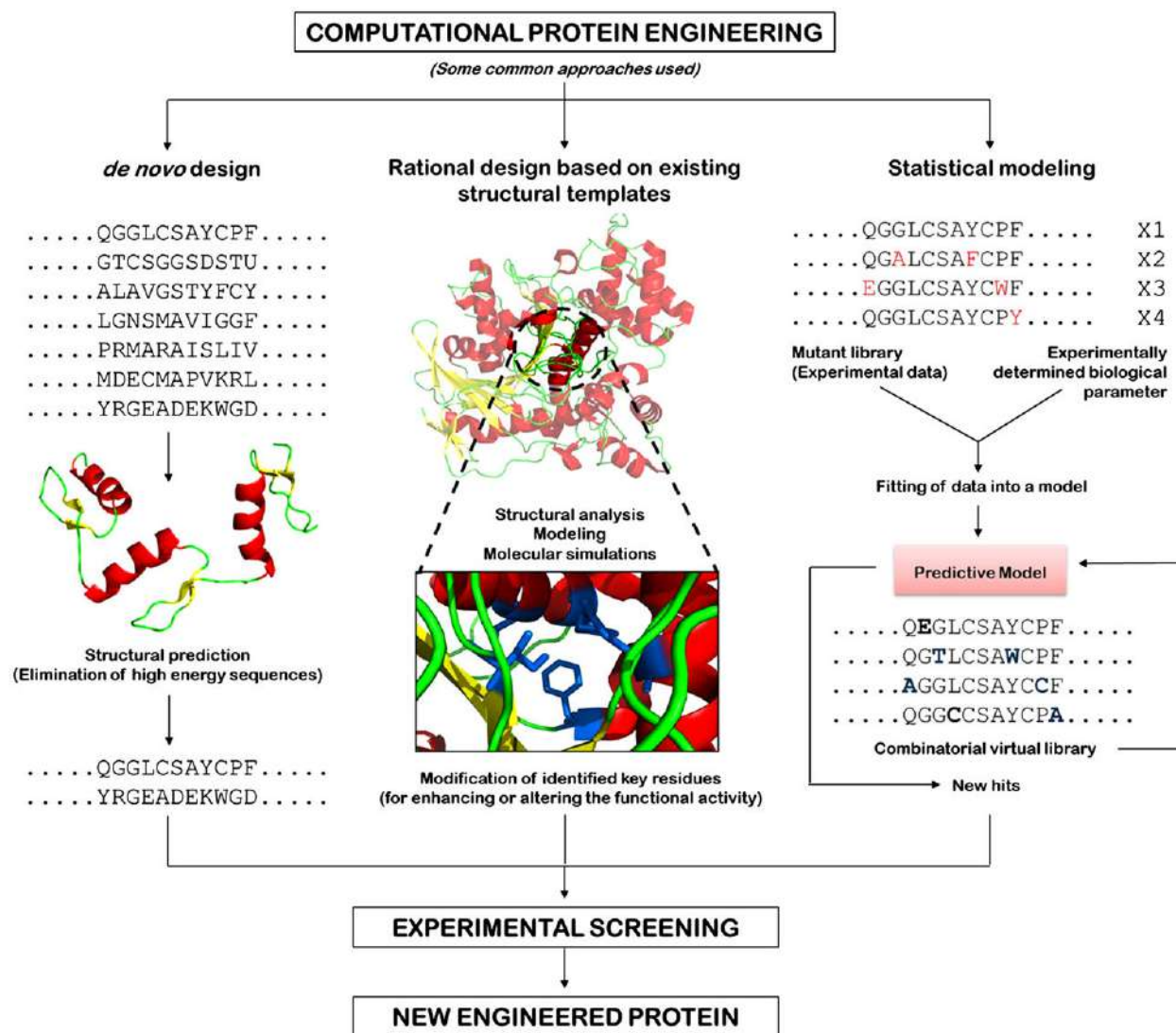


Fig. 1 A schematic representation of the common computational protein engineering approaches: *de novo* designing, protein engineering based on the existing templates, and statistical modeling.

The available computational methods are robust and help in decoding the information hidden in the sequences for protein engineering. Most of the biochemical processes depend upon the specificity of interacting proteins in a crowded cellular environment. Also the stability of the structured proteins influences the formation of functional complexes to initiate the biological response. Thus, the efficiency of any such interaction is primarily governed by two factors: stability and specificity. So, rational re-engineering of proteins must involve improvement in stability, and reduction in affinity for competing binders. Maintaining a balance between the stability and specificity has always remained a challenging task and holds potential implications for the development of newer tools based on existing biological systems for varied applications. Here, successful stories of proteins that have been engineered using various computational approaches have been discussed, with a focus on three important aspects: protein specificity, stability, and novel functionality.

Applications of Computational Protein Engineering

Alteration in the Specificity of Macromolecules by Structural Modifications

The function of nearly all proteins depends on their ability to bind other molecules, or ligands, with a high degree of specificity. High specificity in proteins makes them important players for many industrial and therapeutic applications. Specificity, in simple words, can be described as the ability of a protein to bind one molecule in preference to other molecules. In this section, we have

Table 1 Computational tools for engineering of proteins

Protein engineering approach	Resource	Description
De novo designing	PyRosetta Toolkit (Adolf-Bryfogle and Dunbrack, 2013)	Graphical user interface for <i>de novo</i> designing of proteins using Rosetta toolkit
	EvoDesign (Mitra <i>et al.</i> , 2013)	Structure and evolutionary profile based search to design proteins
	ORBIT (Dahiyat and Mayo, 1996)	Uses rotamer description of side chains to predict the most optimal geometry for any amino acid sequence from vast number of possible solutions
	POCKETOPTIMIZER (Stiel <i>et al.</i> , 2016)	Optimizes protein-binding site for new ligand
	SABER (Nosrati and Houk, 2012)	Selection of active/binding sites for enzyme redesign (SABER). Designs new active/binding site by identifying protein active sites with similar arrangement of catalytic residues from PDB
	DEEPer (Hallen <i>et al.</i> , 2013)	Algorithm for protein engineering through generation of ensemble of protein backbone and side chain flexibility
Structure or template based designing	DEZYMER (Hellings and Richards, 1991)	Utilizes backbone information to identify positions of residues to be modified for incorporation of new binding site
	CAVER (Chovancova <i>et al.</i> , 2012)	Uses molecular dynamics simulation trajectories to identify tunnel and channel characteristics from large ensemble of protein conformations. This information can be utilized to engineer phenomena of molecular transport, enzymatic catalysis, and molecular recognition
	Metal Search (Clarke and Yuan, 1995)	Uses backbone coordinates to identify cluster of residues that can form tetrahedral metal binding site upon replacement with Cys or His
	FlexPred (Kuznetsov and McDuffie, 2008)	Web server to predict protein flexibility by recognizing residues involved in conformational switches
Statistical modeling	FoldX (Schymkowitz <i>et al.</i> , 2005)	An empirical force-field based method designed to check the effect of mutations on stability and folding dynamics of proteins
	AUTO-MUTE (Masso and Vaisman, 2010)	Web server based on machine learning algorithms designed for predicting effect of mutations on stability of proteins
	iPTREE-STAB (Huang <i>et al.</i> , 2007)	Utilizes decision tree algorithm to discriminate between stabilizing and destabilizing mutations; employs regression and classification tree algorithm for prediction of effect of mutations on protein stability
	MuD (Wainreb <i>et al.</i> , 2010)	Mutation Detector (MuD): utilizes Random Forest based algorithm to discriminate between functionally neutral and non-neutral mutations. It utilizes structure and sequence information to predict the effect of mutations on protein function
	PEAT-SA (Johnston <i>et al.</i> , 2011)	Empirical based method to predict the effect of mutation on ligand affinity, <i>pKa</i> values and stability of protein

discussed some examples where computational approaches have been used to enhance the specificity of the proteins for their successful diverse applications.

Computational design of zinc finger proteins for specific binding to DNA sequences

Zinc finger nucleases (ZFNs) are promising genome engineering tools having the potential to edit DNA at the molecular level, thereby paving the way for many biological applications. These chimeric proteins are made up of two domains, one that recognizes specific DNA sequence within the complete genome and the other is a non-specific endonuclease domain that helps in introducing a double stranded break in the target sequence. The versatility of the ZFNs in the field of targeted genome editing is attributed to its DNA binding domain, which is made up of three zinc finger (ZF) motifs linked using short spacer sequences (Urnov *et al.*, 2005).

These ZF proteins comprise the largest family of transcription factors in eukaryotes. Each ZF motif contains amino acids folded into a $\beta\beta\alpha$ -structure. Extensive structural studies revealed that the modular structure remains highly conserved and the specificity of these factors toward their target DNA relies only on four amino acids, positioned at -1 , 2 , 3 , and 6 residues with respect to the start of alpha helix (Elrod-Erickson *et al.*, 1998; Wolfe *et al.*, 2000; Pabo *et al.*, 2001). Soon, it was realized that these positions can be altered artificially for changing the binding specificity of these motifs toward some other sequences. Knowledge of ZF motifs with the ability to recognize all possible DNA codons thus became a prime objective of the scientific community to gain the power for constructing artificial ZFPs for the purpose of genome manipulation for therapeutic advantages.

Experimental techniques for engineering ZF proteins by rational designing or selection from large libraries, despite being useful, were time consuming, expensive, laborious, and not accessible in academic settings (Beerli and Barbas, 2002; Greisman and Pabo, 1997). Also, randomization of all 20 amino acids at essential DNA contact positions even in three finger ZF proteins can result in an enormously huge number of possibilities that become impossible to handle experimentally. Therefore, only partial libraries could be constructed by restricting the substitution at variable sites with some key amino acid residues. Thus, one possible solution was to computationally design ZF proteins using screening or prediction algorithms.

Construction of the first non-natural ZF motif using computational approach was first reported in 1997 (Dahiyat and Mayo, 1997). A virtual combinatorial library consisting of 1.9×10^{27} amino acid sequences was constructed. This library size was reported to be 15 times larger than the random libraries obtained by experimental approaches. The size of possibilities further increased to 1.1×10^{62} by explicitly considering the rotamers of each possible amino acid at each possible position. DEE algorithm was used to find globally optimal sequences in their optimal conformations. The library was screened using an algorithm designed on the basis of physiochemical potential functions and stereochemical constraints. The structure of top ranking sequence hit was solved using nuclear magnetic resonance spectroscopy and was found to be well-ordered and in agreement with the design target structure, Zif-268. This study demonstrated the potential of computational techniques for searching huge combinatorial libraries, essential for designing novel proteins with high specificity (Dahiyat and Mayo, 1997).

Consequently, many prediction tools were developed based on the knowledge gained from the growing structural information on ZF proteins. These tools can basically be divided into two categories: (1) tools developed using experimental sequence- or structure-based data, and (2) tools simulating the biochemical interactions between ZF protein and its target DNA and estimating the specificity of binding (Grover *et al.*, 2010). Here, we briefly discuss a few tools to give an insight to the approach these tools use for computational design of ZF proteins. *Zinc Finger Tools* is a web-based utility that makes use of 49 different helices that can recognize 16 GNN, 15 ANN, 15 CNN, and 3 TNN DNA triplets. The search tool looks for all possible target sites (adjacent or interspaced) formed by any of these DNA triplets on either strand. It also reports the specificity of the ZF protein predicted using these domains from the results of a multi-target ELISA specificity assay carried out for individual ZF. Apart from predicting ZF proteins for a given sequence, the tool can also predict the probable binding site for a given ZF protein (Mandell and Barbas, 2006). Further, along with the information of the binders (i.e., known examples of high affinity ZFs and their corresponding targets), weak and non-binders were used in the development of another tool. This predicts the most optimal ZF proteins for a given DNA sequence using support vector machines (Persikov *et al.*, 2009). *Zif-Predict* is a sequence-based tool employing artificial neural networks for prediction of ZF proteins for the input nucleotide sequence. An advantage of this tool is that it takes into account the binding preference of one amino acid in the presence of another in its neighborhood, called the synergistic mode of binding (Molparia *et al.*, 2010). Another addition to this league is *Zif-Predict Interfacial Hydrogen Bond Energy (IHBE)* that uses hydrogen bond energies between amino acid residues and nucleotides to predict the highest affinity ZF proteins for the user-input DNA. This is carried out by using energies calculated for all possible combinations of triplets and ZFs with different amino acids at 4 crucial positions ($-1, +2, +3, +6$), thus covering the complete interaction sample set (Dutta *et al.*, 2016b). *ZifNN* is a tool different from the ones mentioned above. It does not depend upon the experimental data, rather uses an *in silico* derived data set of 50 three-dimensional ZF protein-DNA complexes and the estimation of hydrogen bond energies. The predictions are made using neural network models (Dutta *et al.*, 2016a). In addition to this, attempts were also made to improve the *in silico* modeling of ZF protein-DNA docked complexes using HADDOCK (Chou *et al.*, 2010). The number of sequences that can be targeted using engineered ZF proteins can be tremendously increased by designing ZF motifs for nearby but not contiguous regions in DNA. For this, a linker with appropriate length that can help in skipping the nucleotides not to be targeted while helping the next motif to correctly align with the DNA triplet is designed. A computational model has been developed for structure-based design of protein linkers for ZFNs that can skip up to 10 bp between adjacent ZF, recognized DNA triplet in the target genomic sequence (Anand *et al.*, 2013). All these efforts have taken the computational design of ZF proteins to the next level.

Structure-based computational engineering of a therapeutic antibody for improving its binding affinity

High specificity and affinity of antibodies are routinely being exploited for therapeutic and industrial purposes. These molecules are an excellent choice for fabricating diagnostic kits and biosensors.

Antibody-antigen complexes display an intense network of interactions involving multiple non-covalent bonds, making it difficult to quantify these interactions at the molecular and atomic levels. However, advancement in computational power and understanding of biological systems has led to the development of force-field parameters that can help in simulating these systems *in silico* (Barderas *et al.*, 2008; Park and Jeon, 2011). Here, we have discussed a recent report where structure-based computational techniques have been used to improve the biotherapeutic potential of 11K2 antibody (Kiyoshi *et al.*, 2014). The main objective of the study was to increase the binding affinity of 11K2 for its target antigen, monocyte chemotactic protein-1 (MCP-1). MCP-1 is a key player in many inflammatory diseases like arteriosclerosis, allergy, and rheumatoid arthritis. Knowledge gained from the crystal structure of K2-MCP-1 complex (PDB ID: 2BDN) suggested that the binding affinity of the antibody can further be increased by optimizing the complementarity determining region (CDR) of the variable domain of the light chain (VL). The interaction surface between the CDR of the VL and MCP-1 was found to be much smaller than that of the CDR of the variable domain of the heavy chain (VH) (Kiyoshi *et al.*, 2014).

Firstly, the 62 amino acids forming the CDR of 11K2 were mutated *in silico* with another 19 possible amino acids to generate a virtual library of 1178 mutants of 11K2. For each mutation, 100 randomized models were generated using MODELLER, Discovery

Studio Suite. Each model was further optimized by using a combination of simulated annealing and molecular mechanics minimization. Similarly, 1000 structures for the wild-type Ab were also generated to be considered as reference. The mutants with more favorable energy of interaction than the wild-type 11K2 were then selected for *in vitro* examination. It was observed that the selected candidates were the ones with substitution by a charged residue (Arg, Lys, Glu, or Asp). The technique of surface plasmon resonance spectroscopy was then used to calculate the kinetic rate constants of the binding of wild-type 11K2 and engineered single-mutants to immobilized MCP-1. The enhanced affinity of the new models was attributed to slower dissociation rate constants rather than faster association rate constants (Kiyoshi *et al.*, 2014). This structure based computational engineering of therapeutic antibody led to approximately 5-fold increase in affinity of Ab for its target antigen and hence strongly reflects upon the potential of computational protein engineering in the coming era (Kiyoshi *et al.*, 2014).

Computational approaches for studying the dynamics of CRISPR-Cas9 system and predicting their binding specificity

CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) system has become a sensational genome editing technology in recent days. This system consists of two components: a guide RNA and a Cas9 (CRISPR associated protein 9) nuclease. The 5' end of guide RNA contains the 20-nucleotide stretch complementary to the region of interest in genome followed by a protospacer adjacent motif. This CRISPR RNA drives the Cas9 nuclease to this complementary site in DNA and guides it to introduce a double strand break (DSB) in that region. Hence, Cas9 can be directed to produce DSB in any region in the genomic DNA that is of the form N(20)-NGG by designing guide RNA with the complementary sequence of target (Hsu *et al.*, 2014; Doudna and Charpentier, 2014; Ran *et al.*, 2013b). Designing the experiment using CRISPR is simple and promising; however the major drawback that is limiting its widespread use for industrial and therapeutic purposes is its off-target activity (Tsai *et al.*, 2015). The guide RNA-Cas9 complex can bind and cleave other unintended regions in the genome having sequence similarity with the target region. A major part of published studies in CRISPR is thus aimed at understanding the factors governing and improving the specificity of the system (Doudna and Charpentier, 2014; Hsu *et al.*, 2014; Chéron *et al.*, 2017; Ran *et al.*, 2013b).

Modifications to the actual system were designed to tackle this problem of off-target activity. The strategy of paired nickases was developed, which reduces the off-target activity by 50–1500 without affecting the on-target activity (Shen *et al.*, 2014; Ran *et al.*, 2013a). fCas9 is another strategy similar to paired nickases produced by fusing inactive dCas9 (Cas9 that lacks endonuclease activity) with one of the two monomers of *FokI* endonuclease. Hence, to make a DSB, fusion of two fCas9 monomers is required. Each monomer is complexed with a guide RNA that binds to targets 15–25 base pairs apart in the region of interest to make a DSB (Guilinger *et al.*, 2014). It has also been shown that usage of truncated guide RNAs (17 nt) led to decreased off-target activity (Fu *et al.*, 2014). Despite these efforts, off-target activity could not be avoided. Checking for potential sites similar to the target DNA locus therefore becomes a prerequisite to prevent wastage of time and resources.

A number of computational tools have been developed and are available online for selecting a potential target within the region of interest in the genome with least or no off-target activity. *CHOPCHOP*, *CCTop*, *E-CRISP*, *Cas-OFFinder*, *GT-Scan*, *sgRNAcas9*, and *MIT CRISPR Design* are some of the popular tools among them (Xie *et al.*, 2014; O'Brien and Bailey, 2014; Heigwer *et al.*, 2014; Montague *et al.*, 2014; Labun *et al.*, 2016; Stemmer *et al.*, 2015; Bae *et al.*, 2014; Ran *et al.*, 2013b). These tools generally use similarity search methods to find off-target sites for each target and rank them. However, theoretically there is no single tool that accurately predicts all possible off-target mutations for a target site.

Another approach to improve specificity of Cas9 was engineering it through structure-guided protein engineering techniques. This method was based on studying the available crystal structure of Cas9 complexed with guide RNA and DNA and hypothesizing mutations that improve specificity. The detailed investigation of the system can be done using computational approaches. It was found from the structure that non-target strand of DNA was stabilized by a charged groove in Cas9 located in between HNH, RuvC, and PAM recognition domain. It was hypothesized that if the positively charged residues in this non-target charged groove is neutralized, it might encourage rehybridization of the DNA thereby reducing the mismatch tolerance in RNA–DNA binding. Hence mutants with individual substitutions in the 31 residues in the non-target groove were generated and their specificity was studied. Five of the 31 mutants showed decreased off-target activity by a factor of 10 compared to the wild type. After performing combinatorial mutagenesis studies using top performing mutants from previous screening, three were identified to be efficient. They were (1) SpCas9 (K855A), (2) mSpCas9 (K810A/K1003A/R1060A) or eSpCas9(1.0), and (3) SpCas9 (K848A/K1003A/R1060A) or eSpCas9(1.1). All three were assessed for specificity and their off-target activity was found to be significantly lower than that of wild type (Slaymaker *et al.*, 2016; Doudna and Charpentier, 2014; Ran *et al.*, 2013b).

Recently, researchers have started using molecular dynamics simulations to study the activity of Cas9-guide RNA complex. Conformational dynamics studies during cleavage have shown the plasticity of HNH domain and role played by non-target DNA in activation of HNH nuclease domain (Palermo *et al.*, 2016). Simulation revealed that Cas9 cleavage results in one base pair staggered end with overhangs and not blunt end in DNA (Zuo and Liu, 2016). Further studies on guide RNA-Cas9 mediated DSB mechanism may aid in designing more specific complexes without any off-target activity.

Re-engineering of Proteins to Make Their Structure More Stable

The feasibility of biological processes is initially dependent on the stability of the proteins involved. One of the main challenges faced while enhancing the stability is to not affect the activity of the target protein. Protein stability in general implies minimizing the total potential energy stored within a protein. Efficient functioning of any molecule requires it to be stable, which is generally

maintained by various interactions. In this section, we discuss various computational approaches available to analyze and predict structural changes that can be incorporated into the protein to enhance its stability.

In silico approach to enhance the stability of TEV protease

The tobacco etch virus (TEV) protein is a nuclear inclusion protease weighing 27 kDa (Allison *et al.*, 1986). The protease is involved during TEV biogenesis for the formation of single proteins after proteolytic degradation of the viral polyprotein (Adams *et al.*, 2005). Its stringent specificity and robustness make it ideal for *in vitro* protein purification and *in vivo* testing of protein–protein interactions (Wehr *et al.*, 2006). The canonical recognition sequence targeted by the protease is ENLYFQ-G/S (Adams *et al.*, 2005).

Here, *in silico* engineering of protease was carried out to enhance its stability and solubility (Cabrita *et al.*, 2007). Being a popular protein for removal of tags from recombinant proteins, its production and storage presented major challenges. The drawbacks were overcome by performing a rational search of point mutations that could be introduced in the protein for improvement of its stability and solubility than its native-state. With the aim of selecting the variants with better results, potential single-site mutations were introduced using PoPMuSiC algorithm (Kwasigroch *et al.*, 2002). The program PoPMuSiC predicts changes in the free energy of protein folding. The predicted changes were based on database-derived potentials (Gilis and Rooman, 1996). The features of TEV were improved in such a manner that (a) the structural integrity as well as the activity is sustained, and (b) the solubility remained even at higher concentrations.

The analysis of calculated potentials predicted five variants to be more stable. The calculations of potentials were obtained by linear combination of the interactions involved and the propensity of the interacting residues. Experimental characterization of the five variants was done by using assays like solubility assay, enzyme kinetics, and equilibrium unfolding analysis. Thus, the goal of designing a completely active TEV variant having better stability and more solubility was successful. Similar rational approach combining different techniques can be applied to any protein structure. Such an accomplishment becomes a powerful tool in the field of biotechnology.

Computational designing of β -sheet surfaces to boost protein stability

β -strands constitute β -sandwich or β -barrel in approximately one quarter of all known protein structures (Gerstein, 1998). In such conformations, one face of the sheet points toward the hydrophobic core of the protein, while the other face aims toward the solvent. The residues facing toward the core are vital in determining the stability of the protein by forming strong bonding interactions. However, the stability can also be detected through the solvent-facing residues. Hence, a significant amount of effort has been made to understand the structural features contributing to β -sheet stability (Lassila *et al.*, 2002; Der *et al.*, 2013; Lawrence *et al.*, 2007).

A study was conducted to redesign the β -sheet surfaces of the fibronectin type III domain of the protein tenascin (TNfn3) using the molecular modeling program Rosetta (Gerstein, 1998). TNfn3 has been a model system to extensively study protein folding and stability. Mutational changes to improve these features have already been demonstrated by many reports. The mutations stabilizing the protein structure have mostly been located in the protein core (Gilbreth *et al.*, 2014; Jacobs *et al.*, 2012).

In silico experiments conducted for this study used the all-atom energy function in Rosetta to achieve the objective. This module was parameterized with a diverse set of sequence design and structure prediction tests. In addition, an attempt was made to validate an empirical approach to achieve the protein stability. The number of salt bridges (glutamate or aspartate paired with lysine or arginine) was increased on the surface of the β -sheets. A significant challenge faced while designing β -sheet proteins was to specify the pairing. The problem was created during tertiary folding of proteins because strands that are distant in primary sequence tend to be paired in the final structure.

Interestingly, the charged residues on TNfn3 could be placed on the strands based on opposite charging through mutational studies. Also, it was ensured that β -strands that are close in primary sequence are not paired in the final protein structure. The rationale for this arrangement was that the charges would favor the folding and stability of the protein due to favorable electrostatic interactions in the folded structural state. Simultaneously, disfavoring of kinetically accessible misfolded states would occur.

Thus, dramatic increase in protein stability was achieved by optimizing interactions on the surfaces of small β -sheet proteins. Two variants of the β -sandwich protein from tenascin were obtained after designing. They were carrying seven and 14 mutations respectively on their β -sheet surfaces. Due to these changes, the thermal midpoint for unfolding was increased. Additionally, the approach based on increasing the number of potential salt bridges on the surfaces of the β -sheets was not found to be a robust strategy.

Computational Protein Engineering for Addition of Novel Function to Existing Proteins

The previous sections detailed the potential of computational protein engineering techniques in improving the specificity and stability of the target protein. Similar approaches can also be utilized to add a completely new function to a protein, such as addition of substrate binding site, metal binding site, and catalytic site. The following section describes some successful studies.

Study of adaptability of periplasmic binding proteins: Computational modification of ligand-binding site for different substrates and their application as biosensors

Biosensors are based on the general principle of detection of protein–ligand interaction and transmission of this interaction into the signal detection mechanism. Application of protein as a biosensor requires a well-defined change in conformation of its structure upon binding with ligand molecule. This change in conformation can be coupled with fluorescent detection by incorporating

fluorophores at the allosteric site of protein that can detect change induced by ligand binding (Hellinga and Marvin, 1998; Dwyer and Hellinga, 2004). Emerging protein engineering techniques allow the integration of signal transducing functional groups into the protein molecule itself that can bring electrostatic or optical changes upon interaction with ligands (Hellinga and Marvin, 1998). Moreover, specificity of protein for ligands can be altered using these techniques (Hellinga and Marvin, 1998; Reimer *et al.*, 2014).

Periplasmic binding proteins (PBPs) from *E. coli* are a widely studied superfamily of proteins for changing their ligand specificities. PBPs are periplasmic protein receptors involved in chemotaxis and nutrient uptake with wide variety of substrate specificities, including amino acids, carbohydrates, oligopeptides, metal ions, and anions (Dwyer and Hellinga, 2004). The PBPs superfamily is characterized by high sequence diversity with conserved structural domain (Hellinga and Marvin, 1998). It consists of two globular domains with ligand-binding site at the interface, connected by two- or three-stranded hinge region (Quijcho and Ledvina, 1996; Hellinga and Marvin, 1998; Dwyer and Hellinga, 2004). Ligand binding at the interface changes the conformation of protein from open ligand-free form to closed ligand-bound form by induction of large bending at the hinge region. This ligand-based conformational change of PBPs provides a great advantage in coupling it to physical detectable signal. Allosteric coupling of reporter group with ligand-induced conformational change of PBPs was successfully implemented to create allosterically regulated biosensors for trinitrotoluene (TNT), L-lactate, serotonin, and Zn(II), (Marvin and Hellinga, 2001; Looger *et al.*, 2003; Dwyer and Hellinga, 2004). Here, we have discussed a few case studies as an example for application of protein engineering in creating biosensors for TNT and metal ion Zn(II) by incorporation of metal binding site.

Modification of PBPs ligand-binding site by adding sensory function for TNT: TNT is not a natural molecule, has carcinogenic properties, and is a potent contaminant of soil and water. Development of strategies to design cost-effective sensors for detection of level of TNT in soil and water was required. Manipulation of PBPs for binding with chemical threats and pollutants like TNT might provide the advantage of designing specific cell-based detectors for these pollutants (Looger *et al.*, 2003). The ligand-binding site of three members of the PBPs superfamily of *E. coli* was engineered for binding of TNT in place of their wild-type ligand (Looger *et al.*, 2003). Three PBPs under study were arabinose-binding protein (ABP), ribose-binding protein (RBP), and histidine-binding protein (HBP). High-resolution three-dimensional structures of proteins in association with their respective ligands were obtained from databases. First, interaction studies of protein with wild-type ligands were performed to identify the amino acid residues involved in protein–ligand interactions. This step was followed by repeated combination of protein–target docking and mutation of identified amino acids residues; that resulted into 10^{45} – 10^{68} mutant structures for 12–18 amino acid residues. Resulting combinatorial mutant structures were resolved based on DEE theorem. Identification of global minimum of a semiempirical potential function was determined for system's molecular interactions. This yielded a complementary surface design for TNT in ABP, RBP, and HBP by exploring essential parameters for recognition of molecule, comprising of molecular shape, size, functional groups, chirality, charge, water solubility, internal flexibility, and molecular surface. The designed surface for TNT was electrically neutral, satisfied hydrogen bonding potential of TNT's functional groups, and hydrophobic residues of TNT were interacting with aliphatic amino acid's side chain. Interactions with aromatic side chain were also noticed in one of the design for ABP and HBP, TNT.A1, and TNT.H1, respectively.

The structures of six receptors were selected with binding specificity for TNT (with mutations ranging from 5 to 17 amino acid residues) for experimental validation. Mutant proteins were designed, overexpressed, purified, and altered by incorporation of fluorescent dye, that is thiol-reactive styryl dye conjugated to cys residue, at the site that responds to the conformation change at the hinge region upon binding of TNT. Ligand dependent changes in fluorescence were monitored for wild-type ligand, TNT, and closely related decoy structures of TNT by titration. All six receptors showed detectable binding affinity for TNT and no binding affinity with wild-type ligand was observed. Also when their binding affinity with related decoy structures was analyzed, specifically 2,4- and 2,6-dinitrotoluene (2,4-DNT and 2,6-DNT) and trinitrobenzene (TNB), it was observed that all six receptors can distinguish between TNT and their related decoy compounds with exception of ABP design, which could not recognize the absence of single methyl group in TNB. Out of the three TNT designs of RBP, affinities of two of them were comparable to the affinity of wild-type protein, that is, 0.1–1.5 mM. Affinity of TNT to nanomolar level was observed for one of the TNT designs of RBP receptor, TNT.R3, which was used for the first time to develop a biosensor for detection of underwater level of TNT into the sea using robotic vehicles for tracing underwater plumes (Mead, 2002).

Modification of Periplasmic binding proteins ligand-binding site by adding sensory function for Zn(II): Modification of maltose-binding protein (MBP) to add zinc binding site near or within maltose-binding interface will be discussed in this section (Marvin and Hellinga, 2001). DEZYMER was used to generate 20 initial designs for zinc binding site consisting of three histidine residues and one water molecule. Four of these were structurally constructed, two of them were designed to replace three maltose-binding residues (that is A1: A63_H, R66_H, W340_H; A2: A63_H, R66_H, Y155_H), and the others were located near the maltose-binding site; additional mutations were also introduced to prohibit steric clashes (this site was named as B site, B1: K42_H, E44_H, Y34_H, E45_H, R344_A; B2: P48_H, G69_H, S337_H, R66_S, Y70_A). Both A and B sites showed the same degree of fluorescence change in presence of the ligand and were able to bind the zinc molecule, while only A site could form zinc mediated formation of closed state. This suggested the need of iterative design strategy for increasing zinc binding affinity, optimization of target site, and elimination of vestigial interactions. Three strategies were used for an iterative progressive design cycle to improve binding affinity for zinc:

1. *Combination of zinc binding sites:* Out of all possible combinations of A1, A2 and B1, B2, only A2B1 combination was sterically feasible. A2B1 showed sigmoidal zinc binding; hill coefficient of 2.0 indicated cooperative interaction between two sites and upon zinc binding fluorescence level was increased by 17.7 fold. This showed that A2B1 mutant resulted in improved zinc bound closed state of protein with increased fluorescence detection signal.

2. *Primary coordination sphere optimization*: Primary coordination sphere was optimized instead of His₃Zn sites as designed in previous steps. Alanine mutagenesis screening of "A" sites was performed by mutating its each residue to alanine. H63_{II} site was common to both A1 and A2 site and its mutation to H63_{II}A partly destroyed zinc binding. Other non-overlapping residue mutations were also vital but the mutation in nearby residues, D65_{II}, demonstrates its importance in coordinating zinc ion. Molecular modeling studies were carried out that suggested the placement of forth residue at Y155_I in A1 and W340_I in A2 for optimization of coordination sphere. Mutation studies for these positions led to selection of Y155_IE and W340_IE mutants that showed higher binding affinity for zinc than their original counterparts A1 and A2, respectively. New site was designed by fusion of vertices, from His₂Glu₂ vertices of A1 and A2, to form A* site (H63_{II}, H66_{II}, E155_I, E340_I) with zinc binding affinity increased to K_d value of 5.1 μM.
3. *Exploitation of conformational equilibrium*: Overall zinc binding affinity is the sum of intrinsic affinity for closed state for zinc and intrinsic equilibrium of open and closed state in absence of zinc. Therefore, zinc binding affinity can be improved by manipulation of these factors. Two strategies were employed for manipulation of intrinsic equilibrium between open and closed state:
 - a. Vestigial maltose binding residues, E111_I and K15_{II}, single and double mutants were created by replacing them with Met and Ala, respectively in A* design. These mutants resulted in enhanced binding efficiency of zinc.
 - b. Mutations were made in protoallosteric site in hinge region that is conformationally different in two states. I329 was tightly packed in open state but binding of ligand changed its compact packing in closed state. Its mutation to I329F was studied to destabilize its open state, which led to increased zinc binding affinity for A* design, with K_d value of 350 nM.

It can be concluded from the above study that this kind of iterative progressive design approach can be utilized to convert proteins like MBP into biosensors for metal through addition of metal binding sites within the binding site of its substrate.

Computational modification of periplasmic binding proteins by addition of catalytic function for design of biologically active enzyme

PBPs structures were also tested for addition of new catalytic function, by creating catalytic site for triose phosphate isomerase (TIM) in RBP and the mutated RBP was found to be a biologically active enzyme that can support the growth of *E. coli* in gluconeogenic conditions (Dwyer *et al.*, 2004). TIM is a glycolytic pathway enzyme that interconverts dihydroxyacetone phosphate (DHAP) and glyceraldehyde 3-phosphate (GAP). This isomerization reaction requires successive transfer of two protons with enediol(ate) as intermediate and requires three catalytic residues, i.e., glutamate, histidine, and lysine. Introduction of TIM catalytic site into RBP was accomplished by mutating ligand-binding interface residues in a step-by step manner. Firstly, RBP was modified to add substrate binding site for DHAP and GAP irrespective of catalytic activity. This was accomplished using similar methods as mentioned in the case study describing addition of binding site of TNT in PBPs. Secondly, the catalytic site residues were introduced into the modified substrate binding site by specifying geometrical constraints of the active site that should be compatible with bond formation with substrates, enediol(ate) intermediate and transition state. The whole process was executed in three steps:

1. First geometrical constraints critical for interactions required for catalysis were designed.
2. Combinatorial search algorithm was employed to generate geometric constraints that satisfy constraints for both substrate specificity and catalysis simultaneously.
3. Receptor design algorithms were used to place generated complementary surface around bound substrate.

The catalytic design site obtained in this step was distinct from the designs generated for substrate binding in the first step. Fourteen (14) designs were selected for testing, out of which one design, NovoTim1.0, showed increased activity and was competitively inhibited by a known inhibitor of TIM, i.e., phosphoglycolate. NovoTim1.0 was found to be less thermostable in comparison to TIM; therefore further modification of residues surrounding binding surfaces of substrates was carried out. These modifications provided variants of NovoTim1.0, namely NovoTim1.2, which showed increased stability of protein by 15°C and increased k_{cat} (rate of reaction) and K_M (Michaelis constant) by twofold. NovoTim1.2 was formed by mutation and redesign of nine interfacial and nine binding residues of NovoTim1.0, its parent molecule. NovoTims1.0 and 1.2 were further tested by their expression in TIM deficient strains of *E. coli* for growth of bacteria on gluconeogenic substrates, glycerol and lactate, in presence and absence of inducer IPTG (isopropyl β-D-1-thiogalactopyranoside). Both versions supported growth of bacteria on lactate medium but could not utilize glycerol as a substrate for growth. Therefore, further mutants were created to design a mutant that, when transformed into bacteria, can aid bacteria to utilize glycerol as a medium for growth. NovoTims1.2.1 to 1.2.4 versions were obtained with mutations of protein surface residues and increased k_{cat} and k_{cat}/K_M values by two-fold and three-fold, respectively.

In this study, the authors clearly demonstrated how combination of computational and experimental procedures can convert non-catalytic protein into a biologically active enzyme that can aid bacteria to utilize gluconeogenic substrate for growth. This approach can be generalized to create various types of enzymes by either utilizing PBPs or any other protein.

Future Challenges of Computational Protein Engineering

With rapidly advancing computational power, *in silico* protein engineering methods have become more common and many successful applications have been reported. But some challenges still remain to be addressed in the field of computational protein engineering to have reliable and successful *in silico* protein designs. Some of the challenges are discussed here.

Initially, to lower the complexity and to decrease the computational time, the protein backbone is kept fixed and its degrees of freedom are eliminated in the protein engineering methods. Hence, residual combinations that require slight changes in backbone become unattainable. Also, these backbone methods cannot be applied to design *de novo* backbone structures. Owing to the drawbacks of fixed backbone methods, flexible backbone protein design methods like parameterization of structures, ensemble approach, and simultaneous optimization have been developed that have their own challenges. Parameterization of structures is a method that allows flexibility while modeling backbone, but decreases the degrees of freedom. Yet there are some successful studies reported using this method (Harbury *et al.*, 1995, 1998). The drawback in this method is that it does not allow explicit flexibility of backbone. In the ensemble approach, an ensemble of protein backbone structures is used to achieve flexibility and then the fixed backbone method is applied to each structure in the ensemble. The limitation in this method is that only a limited number of backbone structures can be studied. In the simultaneous optimization method, flexibility is allowed to backbone and side chains simultaneously. Although different flexibility backbone methods are established, there is an absence of good correlation between stabilities of a protein determined experimentally and computationally (Choi *et al.*, 2009). Allowing backbone flexibility thus still remains a challenge in protein engineering.

Negative design is a method to ensure specificity. Positive design involves engineering of protein to have a desired function, whereas negative design is for preventing undesired functions. In negative design, a sequence is interrogated against multiple structures so that it does not form alternate conformations. It helps in selecting sequences that form a unique structure. Prior knowledge about target structure and undesired complexes is needed in order to avoid sequences that can form undesired structures. Different negative design approaches have been reported in literature. Of these, multi-state design is a potential approach for designing proteins with specificity. Combinatorial engineering is another useful method to attain high specificity. Combining the power of computational negative design tools along with high throughput assays poses a challenge. Protein folding is a longstanding issue in protein science. Incorporating negative design to destabilize alternate structures is a challenging task (Choi *et al.*, 2009).

Another daunting task in the protein engineering pipeline is high throughput sampling of predicted structures. After computational design, all the successful designs have to be cloned in a vector, expressed in a cell, and purified for characterization. Repeating this for all successful designs one after another will be expensive, time consuming, and laborious. Hence, in future, any high throughput protein fabrication method to be developed should have automation in all the aforementioned tasks. Another approach can be constructing a gene library that represents the ensemble of proteins and expressing them simultaneously. Screening can be applied to select the protein with the desired property. So far, the selection methods applied on these libraries have been of low resolution without atomistic level sequence interrogations (Choi *et al.*, 2009).

We have discussed here some successful computational approaches that have been used to engineer proteins with better specificity, stability, and newer functionality. These examples reflect the advantages of using computational tools to mine the information embedded within the sequence and structure of native proteins, and redefine it for our interests. Though computational protein engineering methods are evolving to be imperative tools in the tedious and complex process of engineering proteins, some associated challenges still leave the scope for further innovation and improvement.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition. Study of The Variability of The Native Protein Structure

References

- Adams, M.J., Antoniw, J.F., Beaudoin, F., 2005. Overview and analysis of the polyprotein cleavage sites in the family Potyviridae. *Molecular Plant Pathology* 6, 471–487.
- Adolf-Bryfogle, J., Dunbrack Jr, R.L., 2013. The PyRosetta Toolkit: A graphical user interface for the Rosetta software suite. *PLOS ONE* 8, e66856.
- Allison, R., Johnston, R.E., Dougherty, W.G., 1986. The nucleotide sequence of the coding region of tobacco etch virus genomic RNA: Evidence for the synthesis of a single polyprotein. *Virology* 154, 9–20.
- Alvizo, O., Mayo, S.L., 2008. Evaluating and optimizing computational protein design force fields using fixed composition-based negative design. *Proceedings of the National Academy of Science of the United States of America* 105, 12242–12247.
- Anand, P., Schug, A., Wenzel, W., 2013. Structure based design of protein linkers for zinc finger nuclease. *FEBS Letters* 587, 3231–3235.
- Bae, S., Park, J., Kim, J.-S., 2014. Cas-OFFinder: A fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases. *Bioinformatics*. btu048.
- Barderas, R., Desmet, J., Timmerman, P., Meloen, R., Casal, J.I., 2008. Affinity maturation of antibodies assisted by in silico modeling. *Proceedings of the National Academy of Sciences* 105, 9029–9034.
- Beerli, R.R., Barbas, C.F., 2002. Engineering polydactyl zinc-finger transcription factors. *Nature Biotechnology* 20, 135–141.
- Cabrita, L.D., Gillis, D., Robertson, A.L., *et al.*, 2007. Enhancing the stability and solubility of TEV protease using in silico design. *Protein Science* 16, 2360–2367.
- Chéron, J.-B., Casciuc, I., Golebiowski, J., Antonczak, S., Fiorucci, S., 2017. Sweetness prediction of natural compounds. *Food Chemistry* 221, 1421–1425.
- Choi, E.J., Guntas, G., Kuhlman, B., 2009. 18 future challenges of computational protein design. *Protein Engineering and Design* 75, 367.
- Chou, C.-C., Rajasekaran, M., Chen, C., 2010. An effective approach for generating a three-Cys 2 His 2 zinc-finger-DNA complex model by docking. *BMC Bioinformatics* 11, 334.
- Chovancova, E., Pavelka, A., Benes, P., *et al.*, 2012. CAVER 3.0: A tool for the analysis of transport pathways in dynamic protein structures. *PLOS Computational Biology* 8, e1002708.
- Clarke, N.D., Yuan, S.M., 1995. Metal search: A computer program that helps design tetrahedral metal-binding sites. *Proteins: Structure, Function, and Bioinformatics* 23, 256–263.
- Clinton, T.R., Weinstock, M.T., Jacobsen, M.T., *et al.*, 2015. Design and characterization of ebolavirus GP prehairpin intermediate mimics as drug targets. *Protein Science* 24, 446–463.
- Dahiyat, B.I., Mayo, S.L., 1996. Protein design automation. *Protein Science* 5, 895–903.
- Dahiyat, B.I., Mayo, S.L., 1997. De novo protein design: Fully automated sequence selection. *Science* 278, 82–87.

- Der, B.S., Kluwe, C., Miklos, A.E., *et al.*, 2013. Alternative computational protocols for supercharging protein surfaces for reversible unfolding and retention of stability. *PLOS ONE* 8, e64363.
- Doudna, J.A., Charpentier, E., 2014. The new frontier of genome engineering with CRISPR-Cas9. *Science* 346, 1258096.
- Dutta, S., Madan, S., Parikh, H., Sundar, D., 2016a. An ensemble micro neural network approach for elucidating interactions between zinc finger proteins and their target DNA. *BMC Genomics* 17, 97.
- Dutta, S., Madan, S., Sundar, D., 2016b. Exploiting the recognition code for elucidating the mechanism of zinc finger protein-DNA interactions. *BMC Genomics* 17, 109.
- Dwyer, M.A., Hellinga, H.W., 2004. Periplasmic binding proteins: A versatile superfamily for protein engineering. *Current Opinion in Structural Biology* 14, 495–504.
- Dwyer, M.A., Looger, L.L., Hellinga, H.W., 2004. Computational design of a biologically active enzyme. *Science* 304, 1967–1971.
- Elrod-Erickson, M., Benson, T.E., Pabo, C.O., 1998. High-resolution structures of variant Zif268–DNA complexes: Implications for understanding zinc finger–DNA recognition. *Structure* 6, 451–464.
- Floudas, C.A., Fung, H.K., Mcallister, S.R., Mönnigmann, M., Rajgaria, R., 2006. Advances in protein structure prediction and de novo protein design: A review. *Chemical Engineering Science* 61, 966–988.
- Fox, R.J., Davis, S.C., Mundorff, E.C., *et al.*, 2007. Improving catalytic function by ProSAR-driven enzyme evolution. *Nature Biotechnology* 25, 338–344.
- Fu, Y., Sander, J.D., Reyon, D., Cascio, V.M., Joung, J.K., 2014. Improving CRISPR-Cas nuclease specificity using truncated guide RNAs. *Nature Biotechnology* 32, 279–284.
- Gerstein, M., 1998. How representative are the known structures of the proteins in a complete genome? A comprehensive structural census. *Folding and Design* 3, 497–512.
- Gilbreth, R., Chacko, B., Grinberg, L., Swers, J., Baca, M., 2014. Stabilization of the third fibronectin type III domain of human tenascin-C through minimal mutation and rational design. *Protein Engineering Design and Selection* 27, 411–418.
- Gillis, D., Rooman, M., 1996. Stability changes upon mutation of solvent-accessible residues in proteins evaluated by database-derived potentials. *Journal of Molecular Biology* 257, 1112–1126.
- Greisman, H.A., Pabo, C.O., 1997. A general strategy for selecting high-affinity zinc finger proteins for diverse DNA target sites. *Science* 275, 657–661.
- Grover, A., Pande, A., Choudhary, K., Gupta, K., Sundar, D., 2010. Re-programming DNA-binding specificity in zinc finger proteins for targeting unique address in a genome. *Systems and Synthetic Biology* 4, 323–329.
- Guillinger, J.P., Thompson, D.B., Liu, D.R., 2014. Fusion of catalytically inactive Cas9 to FokI nuclease improves the specificity of genome modification. *Nature Biotechnology* 32, 577–582.
- Hallen, M.A., Keedy, D.A., Donald, B.R., 2013. Dead-end elimination with perturbations (DEEPer): A provable protein design algorithm with continuous sidechain and backbone flexibility. *Proteins: Structure, Function, and Bioinformatics* 81, 18–39.
- Harbury, P.B., Plecs, J.J., Tidor, B., Alber, T., Kim, P.S., 1998. High-resolution protein design with backbone freedom. *Science* 282, 1462–1467.
- Harbury, P.B., Tidor, B., Kim, P.S., 1995. Repacking protein cores with backbone freedom: Structure prediction for coiled coils. *Proceedings of the National Academy of Sciences* 92, 8408–8412.
- Heigwer, F., Kerr, G., Boutros, M., 2014. E-CRISP: Fast CRISPR target site identification. *Nature Methods* 11, 122–123.
- Hellings, H.W., Marvin, J.S., 1998. Protein engineering and the development of generic biosensors. *Trends in Biotechnology* 16, 183–189.
- Hellings, H.W., Richards, F.M., 1991. Construction of new ligand binding sites in proteins of known structure: I. Computer-aided modeling of sites with pre-defined geometry. *Journal of Molecular Biology* 222, 763–785.
- Hsu, P.D., Lander, E.S., Zhang, F., 2014. Development and applications of CRISPR-Cas9 for genome engineering. *Cell* 157, 1262–1278.
- Huang, L.-T., Gromiha, M.M., Ho, S.-Y., 2007. iPTREE-STAB: Interpretable decision tree based method for predicting protein stability changes upon mutations. *Bioinformatics* 23, 1292–1293.
- Jacobs, S.A., Diem, M.D., Luo, J., *et al.*, 2012. Design of novel FN3 domains with high stability by a consensus sequence approach. *Protein Engineering Design and Selection* 25, 107–117.
- Johnston, M.A., Søndergaard, C.R., Nielsen, J.E., 2011. Integrated prediction of the effect of mutations on multiple protein characteristics. *Proteins: Structure, Function, and Bioinformatics* 79, 165–178.
- Kiyoshi, M., Caaveiro, J.M., Miura, E., *et al.*, 2014. Affinity improvement of a therapeutic antibody by structure-based computational design: Generation of electrostatic interactions in the transition state stabilizes the antibody-antigen complex. *PLOS ONE* 9, e87099.
- Kuznetsov, I.B., McDuffie, M., 2008. FlexPred: A web-server for predicting residue positions involved in conformational switches in proteins. *Bioinformatics* 3, 134.
- Kwasigroch, J.M., Gillis, D., Dehouck, Y., Rooman, M., 2002. PoPMuSiC, rationally designing point mutations in protein structures. *Bioinformatics* 18, 1701–1702.
- Labun, K., Montague, T.G., Gagnon, J.A., Thyme, S.B., Valen, E., 2016. CHOPCHOP v2: A web tool for the next generation of CRISPR genome engineering. *Nucleic Acids Research*. gkw398.
- Lassila, K.S., Datta, D., Mayo, S.L., 2002. Evaluation of the energetic contribution of an ionic network to beta-sheet stability. *Protein Science* 11, 688–690.
- Lawrence, M.S., Phillips, K.J., Liu, D.R., 2007. Supercharging proteins can impart unusual resilience. *Journal of the American Chemical Society* 129, 10110–10112.
- Li, X., Zhang, Z., Song, J., 2012. Computational enzyme design approaches with significant biological outcomes: Progress and challenges. *Computational and Structural Biotechnology Journal* 2, e201209007.
- Looger, L.L., Dwyer, M.A., Smith, J.J., Hellings, H.W., 2003. Computational design of receptor and sensor proteins with novel functions. *Nature* 423, 185–190.
- Mandell, J.G., Barbas, C.F., 2006. Zinc Finger Tools: Custom DNA-binding domains for transcription factors and nucleases. *Nucleic Acids Research* 34, W516–W523.
- Marvin, J.S., Hellings, H.W., 2001. Conversion of a maltose receptor into a zinc biosensor by computational design. *Proceedings of the National Academy of Sciences* 98, 4955–4960.
- Masso, M., Vaisman, I.I., 2010. AUTO-MUTE: Web-based tools for predicting stability changes in proteins due to single amino acid replacements. *Protein Engineering, Design & Selection* 23, 683–687.
- Mead, K.S., 2002. Using lobster noses to inspire robot sensor design. *Trends in Biotechnology* 20, 276–277.
- Meyer, M.M., Silberg, J.J., Voigt, C.A., *et al.*, 2003. Library analysis of SCHEMA-guided protein recombination. *Protein Science* 12, 1686–1693.
- Mitra, P., Shultis, D., Zhang, Y., 2013. EvoDesign: De novo protein design based on structural and evolutionary profiles. *Nucleic Acids Research* 41, W273–W280.
- Molparia, B., Goyal, K., Sarkar, A., Kumar, S., Sundar, D., 2010. ZIF-Predict: A web tool for predicting DNA-binding specificity in C2H2 zinc finger proteins. *Genomics, Proteomics & Bioinformatics* 8, 122–126.
- Montague, T.G., Cruz, J.M., Gagnon, J.A., Church, G.M., Valen, E., 2014. CHOPCHOP: A CRISPR/Cas9 and TALEN web tool for genome editing. *Nucleic Acids Research* 42, W401–W407.
- Murphy, G.S., Sathyamoorthy, B., Der, B.S., *et al.*, 2015. Computational de novo design of a four-helix bundle protein–DND_4HB. *Protein Science* 24, 434–445.
- Nosrati, G.R., Houk, K., 2012. SABER: A computational method for identifying active sites for new reactions. *Protein Science* 21, 697–706.
- O'brien, A., Bailey, T.L., 2014. GT-Scan: Identifying unique genomic targets. *Bioinformatics*. btu354.
- Pabo, C.O., Peisach, E., Grant, R.A., 2001. Design and selection of novel Cys2His2 zinc finger proteins. *Annual Review of Biochemistry* 70, 313–340.
- Palermo, G., Miao, Y., Walker, R.C., Jinek, M., Mccammon, J.A., 2016. Striking plasticity of CRISPR-Cas9 and key role of non-target DNA, as revealed by molecular simulations. *ACS Central Science* 2, 756–763.
- Park, H., Jeon, Y.H., 2011. Free energy perturbation approach for the rational engineering of the antibody for human hepatitis B virus. *Journal of Molecular Graphics and Modelling* 29, 643–649.
- Park, S., Yang, X., Saven, J.G., 2004. Advances in computational protein design. *Current Opinion in Structural Biology* 14, 487–494.
- Persikov, A.V., Osada, R., Singh, M., 2009. Predicting DNA recognition by Cys2His2 zinc finger proteins. *Bioinformatics* 25, 22–29.

- Quioco, F.A., Ledvina, P.S., 1996. Atomic structure and specificity of bacterial periplasmic receptors for active transport and chemotaxis: Variation of common themes. *Molecular Microbiology* 20, 17–25.
- Ran, F.A., Hsu, P.D., Lin, C.-Y., *et al.*, 2013a. Double nicking by RNA-guided CRISPR Cas9 for enhanced genome editing specificity. *Cell* 154, 1380–1389.
- Ran, F.A., Hsu, P.D., Wright, J., *et al.*, 2013b. Genome engineering using the CRISPR-Cas9 system. *Nature Protocols* 8, 2281–2308.
- Reimer, A., Yagur-Kroll, S., Belkin, S., Roy, S., Van Der Meer, J.R., 2014. *Escherichia coli* ribose binding protein based bioreporters revisited. *Scientific Reports* 4, 5626.
- Rothlisberger, D., Khersonsky, O., Wollacott, A.M., *et al.*, 2008. Kemp elimination catalysts by computational enzyme design. *Nature* 453, 190–195.
- Saraf, M.C., Moore, G.L., Goodey, N.M., *et al.*, 2006. IPRO: An iterative computational protein library redesign and optimization procedure. *Biophysics Journal* 90, 4167–4180.
- Schymkowitz, J., Borg, J., Stricher, F., *et al.*, 2005. The FoldX web server: An online force field. *Nucleic Acids Research* 33, W382–W388.
- Shen, B., Zhang, W., Zhang, J., *et al.*, 2014. Efficient genome modification by CRISPR-Cas9 nickase with minimal off-target effects. *Nature Methods* 11, 399–402.
- Slaymaker, I.M., Gao, L., Zetsche, B., *et al.*, 2016. Rationally engineered Cas9 nucleases with improved specificity. *Science* 351, 84–88.
- Stemmer, M., Thumberger, T., Del Sol Keyer, M., Wittbrodt, J., Mateo, J.L., 2015. CCTop: An intuitive, flexible and reliable CRISPR/Cas9 target prediction tool. *PLOS ONE* 10, e0124633.
- Stiel, A.C., Nellen, M., Höcker, B., 2016. Pocket optimizer and the design of ligand binding sites. *Computational Design of Ligand Binding Proteins*. 63–75.
- Tsai, S.Q., Zheng, Z., Nguyen, N.T., *et al.*, 2015. GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. *Nature Biotechnology* 33, 187–197.
- Turanli-Yildiz, B., Alkim, C., Cakar, Z.P., 2012. *Protein Engineering Methods and Applications*. Rijeka, Croatia: INTECH Open Access Publisher.
- Ulmer, K.M., 1983. Protein engineering. *Science* 219, 666–671.
- Urnov, F.D., Miller, J.C., Lee, Y.-L., *et al.*, 2005. Highly efficient endogenous human gene correction using designed zinc-finger nucleases. *Nature* 435, 646–651.
- Verma, R., Schwaneberg, U., Roccatano, D., 2012. Computer-aided protein directed evolution: A review of web servers, databases and other computational tools for protein engineering. *Computational and Structural Biotechnology Journal* 2, e201209008.
- Wainreb, G., Ashkenazy, H., Bromberg, Y., *et al.*, 2010. MuD: An interactive web server for the prediction of non-neutral substitutions using protein structural data. *Nucleic Acids Research* 38, W523–W528.
- Walsh, G., 2007. Protein engineering: Case studies of commercialized engineered products. *Biochemistry and Molecular Biology Education* 35, 2–8.
- Wehr, M.C., Laage, R., Bolz, U., *et al.*, 2006. Monitoring regulated protein-protein interactions using split TEV. *Nature Methods* 3, 985–993.
- Wolfe, S.A., Nekudova, L., Pabo, C.O., 2000. DNA recognition by Cys2His2 zinc finger proteins. *Annual Review of Biophysics and Biomolecular Structure* 29, 183–212.
- Xie, S., Shen, B., Zhang, C., Huang, X., Zhang, Y., 2014. sgRNAs9: A software package for designing CRISPR sgRNA and evaluating potential off-target cleavage sites. *PLOS ONE* 9, e100448.
- Zuo, Z., Liu, J., 2016. Cas9-catalyzed DNA cleavage generates staggered ends: Evidence from molecular dynamics simulations. *Scientific Reports* 5.

Biographical Sketch



Jaspreet Kaur received her M.Tech. in Bioinformatics from Delhi Technological University, India in 2013, and B.Tech. in Biotechnology from Amity University, India (2011). She joined the Indian Institute of Technology, Delhi, India in 2014 for her PhD program. She works in the field of computer-aided drug design and is devising computational approaches for aiding in targeted genome editing.



Vidhi Malik received her M.Tech. in Bioinformatics from Delhi Technological University, India in 2013 and B.Tech. in Biotechnology from Sardar Vallabhbhai Patel University of Agriculture and Technology, India in 2011. She joined the Indian Institute of Technology, Delhi, India in 2015 for her PhD program. Her research interest lies in next generation sequence (NGS) data analysis and computer-aided drug design.



Navaneethan Radhakrishnan received his B.Tech. in Bioinformatics from Tamil Nadu Agricultural University, India in 2016. He joined the Indian Institute of Technology Delhi, India in 2016 as Junior Research Fellow in the Department of Biochemical Engineering and Biotechnology. His research interest lies in understanding the biological systems using computational approaches.



Moolchand Sigar received his M.Sc. in Biochemistry from University of Hyderabad, India in 2011, and B.Sc. in Biotechnology from University of Bikaner, India (2008). He joined Indian Institute of Technology, Delhi, India in 2012 for his PhD program. His research interest lies in engineering and production of therapeutic proteins.



Anjani Kumari received her B.Tech. in Biotechnology from Amity University, India in 2015. She completed her M.S. (Research) program in Biochemical Engineering and Biotechnology at IIT Delhi in 2017. Her research interest lies in computer-aided drug discovery.



Durai Sundar is a DuPont Young Professor in the Department of Biochemical Engineering and Biotechnology at Indian Institute of Technology, Delhi. He obtained his education from Pondicherry University and Johns Hopkins University, Baltimore, USA. He is a specialist in molecular and computational biology and his current research interests are in rational design of genome editing tools and in the biological activity of natural drugs.

Protein Design

Ragothaman M Yennamalli, Jaypee University of Information Technology, Wanknaghat, Himachal Pradesh, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

A cursory search of the Protein Data Bank (PDB) with the keyword “*de novo*” returns 962 entries, while for the keyword “designed”, 6225 entries were returned [as of August 2017]. Although these numbers (*de novo* or designed structures) are 1000 times lesser in magnitude (compared to the total number of structures the PDB holds, which is close to 135,000 structures), these ~7000 structures indicate how far the field of protein design has advanced, since 1950s. Protein designing projects are ambitious in their goal due to the simple yet complex problem of protein folding. Much has been learned about how a protein folds, and these fundamental knowledge have helped further the area of protein design to conceive proteins with imaginative structures (Richardson and Richardson, 1989; Bowie *et al.*, 1991).

The impetus for protein design is two-fold: (i) assumption that we can design a complex natural system from first principles, and (ii) the “made to order” macromolecules that can solve important biochemical hurdles. The basic or fundamental problem with protein design in achieving a three dimensional, stable, and functional macromolecule is to cross the conformational entropy from the primary to the tertiary structure (Bowie *et al.*, 1991; Dahiyat and Mayo, 1997). There are many methods that can be used to reduce the conformational entropy (Baxa *et al.*, 2014). These include covalent cross-links and other artificial constraints that limit the conformational possibilities of the designed molecule (Leitner *et al.*, 2010; Sinz *et al.*, 2015).

There have been two basic design principles employed in designing proteins *de novo*: positive design and negative design (DeGrado *et al.*, 1989). Positive design of protein structures is the idea to design a protein with the desired structure as the goal, and rationally add/remove residues to achieve that structure. In contrast, negative design involves designing a structure along with ways to reduce formation of or competition from alternative conformations that may arise. While both methods involve rational design, there are advantages of using one over the other. Nevertheless, *de novo* design of protein structures using both methods has been successful to a varying degree. Fig. 1 shows a representative set of structures that have been successfully designed and have advanced the field of protein design.

Since 1950s, designing alpha helices took precedence than the beta sheets, due to:

- The stabilizing hydrogen bond network (Fig. 2).
- The observation of isolated helices being stable in solution (Brown and Klee, 1971; Kim and Baldwin, 1984; Marqusee and Baldwin, 1987; Shoemaker *et al.*, 1987; Marqusee *et al.*, 1989).
- Its oligomerization property observed by Crick, Pauling and Corey in supercoiled helices that when two helices twist around each other there are 3.5 residues per turn, which is less than the ideal 3.6 residues per turn rule, thereby leading to a repetition of the entire structure at every seven residues (Crick, 1953; Pauling and Corey, 1953).
- The possibility of designing the minimal sequence, which can be repeated to construct a four- or six-bundle helices that can either self assemble or assemble in the presence of an external assembly inducer (Schafmeister *et al.*, 1997).

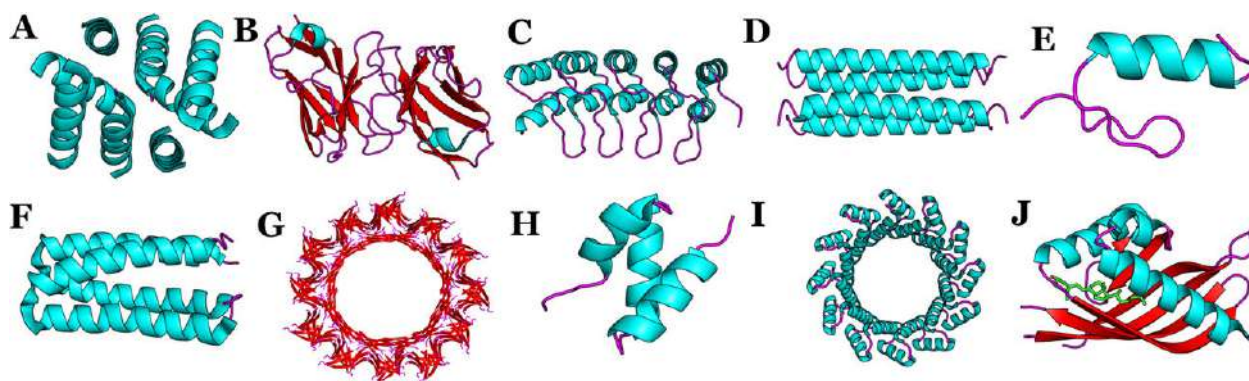


Fig. 1 A representative set of structures that have been designed *de novo*, where they are shown in cartoon representation with helices colored cyan, β -strands colored red, and loops colored magenta. (A) Octameric *de novo* Designed Peptide (pdb id:1l4x), (B) *de novo* Design of an Antibody Combining Site (pdb id:1ivl), (C) Self-Assembling Cyclic Protein Homo-Oligomer (pdb id:4hb5), (D) Right-Handed Coiled Coil Tetramer (pdb id:1rh4), (E) Beta Beta Alpha Protein Motif (pdb id:1fsv), (F) *de novo* Design of a Hyperstable Non-Natural Protein-Ligand Complex (pdb id:5tgw), (G) Giant Double-Walled Peptide Nanotube (pdb id:5vf1), (H) *de novo* Designed Mini Protein Hhh_Rd1_0142 (pdb id:5uoi), (I) Computationally Designed Left-Handed Alpha/Alpha Toroid With 12 Repeats (pdb id:5byo), and (J) Computationally Designed Vitamin-D3 Binder (pdb id:5iep).

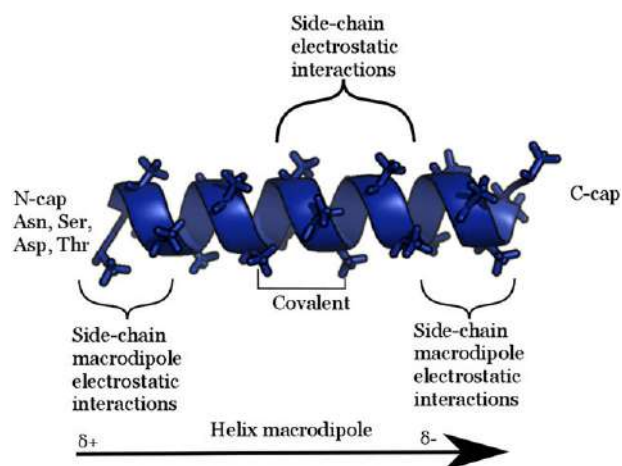


Fig. 2 Helix stabilizing properties. Schematic representation of an ideal helix indicating the various properties that stabilize the helix. Adapted from Bryson, J.W., *et al.*, 1995. Protein design: A hierarchic approach. Science (New York, NY) 270 (5238), 935–941.

Fundamentals

Before taking a sequence and designing it into a novel structure (α -helical or β -sheets or mixed), few pointers need to be kept in mind:

1. Similar stretch of residues can form different secondary structure.
It has been observed that the same residue stretch of up to five amino acids can form different conformations, such as α -helix, β -strand, and loop. This indicates that a sequence forming a structure is dependent on its local environment.
2. There are only so many folds present.
The relatively limited number of folds in the protein universe can be considered as a double edged sword. The 1000 odd folds we know today indicate that proteins tend to fold in one of the restricted ways. It also suggests the possibility of creating new folds using computational tools. The former has led to the reverse folding problem, called threading, where a sequence is checked for its compatibility to fold into one of the known folds.
3. Reducing the entropy of protein by introducing disulfide bonds (Wetzel, 1987).
4. Glycine introduces relatively more conformational freedom than the other 19 amino acids.
5. In contrast, Proline introduces less conformational freedom due to its covalent bond with the main chain.
6. It is known that the residues at the end of a helix are positively (at the C-terminus) and negatively (at the N-terminus) charged, which help in stabilizing the helix by balancing the helix dipole charge. Thus, changing the helix stabilizing residues influences the stability of a protein (Nicholson *et al.*, 1988; Richardson and Richardson, 1988).
7. If a core of a protein has a cavity, filling the cavity makes the protein destabilize it.

Designing an α -Helical Structure

Historically, alpha helical peptides and all- α proteins were relatively easier to design. In order to design alpha helical proteins, the sequence under consideration should be scrutinized with the following guidelines.

1. The sequence should be able to form amphiphilic secondary structures, where there is a periodicity of polar and nonpolar residues. For example, every third or fourth position being nonpolar.
2. A general rule of thumb is that the sequence should have residues that are highly likely to be part of a helix (Ala, Glu, Leu, and Met) (Richardson, 1981). Specifically, multiple alanine residues stabilize the helix. However, if the nonpolar periodicity of 3 or 4 is not maintained, then there is a lower preference to form a helical structure (Xiong *et al.*, 1995).
3. Introduction of salt-bridges and hydrogen bonds between side chains of residues that are one helical turn away (Marqusee *et al.*, 1989; Lyu *et al.*, 1990; Huyghues-Despointes *et al.*, 1993; Park *et al.*, 1993; Scholtz *et al.*, 1993).
4. A charged residue can be introduced at the N or C terminus of the α -helix, creating a macrodipole (Fig. 2).
5. Capping the helix ends with Asn, Ser, Asp, or Thr to satisfy the hydrogen-bond donors and acceptors.
6. Adding hydrogen bond(s) between side chains of residues that are one helical turn away.
7. In the case of Ser, Thr, and other amino acids that can be phosphorylated, their location in the interior of the helix may lead to destabilization. However, the N-terminus capping of a phosphorylated residue leads to stabilization of the helix. This is due to the electronic interaction of the phosphorylated residue with the peptide backbone.

Certain tools that can be used specifically to design helices are helical wheel and predicting helicity.

Helical wheel is a simple tool to identify if the distribution of charged residues and hydrophobic residues would lead to aggregation. When the helix has majorly hydrophobic or nonpolar residues on one side, there is a high chance of protein aggregation. This highly efficient tool has been used for designing proteins that are primarily coiled coil, and for helix bundles comprised of 3, 4, 5, and 6 helices. Specifically, the tool becomes effective when these helices are linked by loops or turns and to check if the “ridges in groove” or “knobs in holes” packing needs to be checked.

Some of the tools that plot a helical wheel for a given sequence are: DrawCoil 1.0 (see Relevant Websites Section), Pepwheel (see Relevant Websites Section), and Helical Wheel Projection (see Relevant Websites Section).

Also, the designed sequence can be checked for its propensity to form helices by submitting the sequence to AGADIR (see Relevant Websites Section) that calculates helicity (Munoz and Serrano, 1994, 1995a,b, 1997; Lacroix *et al.*, 1998).

Designing a β -Sheet Structure

β -sheets can be parallel, antiparallel, and mixed in nature. Due to absence of planarity, parallel β -sheets are less stable than antiparallel sheets. This is due to the relative absence of inter-strand hydrogen bonds in parallel β -sheets, which provide for stronger interaction. Fig. 1(B), (G), and (J) highlight the successful design of a β -strand rich proteins.

In 1996, the design of a decapeptide adopting a β -hairpin structure involving a β -bulge and three other dodecapeptides adopting a type I' β -turn made the protein design possible for β -rich proteins (de Alba *et al.*, 1996; Ramirez-Alvarado *et al.*, 1996; Stanger and Gellman, 1998). These reports fueled enthusiasm in four different groups to design a three stranded antiparallel β -sheet proteins (each differing in residue length) (Kortemme *et al.*, 1998; Schenck and Gellman, 1998; Sharman and Searle, 1998; de Alba *et al.*, 1999). Irrespective of the method they employed, some of the criteria that were included in all the above mentioned β -sheet proteins are:

1. Residues that have higher β -strand propensities were included in the sequence.
2. Inter-strand pairs were selected that have higher preference to make a stable bond.
3. Involving positively charged residues (2 to 5 residues, at least) so that aggregation of proteins is eliminated, and increases solubility.

Positive charge distribution is essential in designing β -sheet proteins, because when they are distributed on both sides of the sheet, aggregation of proteins is reduced to a large extent. Another method to design a β -sheet protein with a stable hydrophobic core involves adding a type I' β -turn to the designed β -hairpin protein by using residues, such as Phe, Trp, Asn, and Gly, so that the side chains face the other strand's hydrophobic residues' side chains (Tyr and Val) (Griffiths-Jones and Searle, 2000).

Similar to α -helices, some of the guidelines to be used while designing β -sheet proteins are as follows:

1. The role of a β -turn is crucial, as it dictates the β -strand. This is a necessary but not a sufficient condition while designing β -sheet proteins. For example, the D form of Pro in the turn stabilizes the β -hairpin structure, compared to the L form.
2. Residues that have high propensity to form β -sheet should be used. For example, Val, Ile, Phe, Trp, Tyr, Leu, and Thr.
3. Gly should be avoided, as its incorporation leads to destabilization.
4. Including salt-bridge forming residues in inter-strands stabilize the β -sheet protein. For example, a salt-bridge formed between Glu and Lys residues. On the same note, salt-bridges at the ends of β -hairpin are more stabilizing.
5. Presence of interactions between hydrophobic residues, such as Ile-Trp, Ser-Thr, and Trp/Val-Tyr/Phe stabilize the protein as their contributions are larger.
6. Side chain-side chain interactions from diagonal directions between two strands also contribute towards stability. For example, Tyr-Lys, Phe/Trp-Lys/Arg interactions.
7. Presence of a right-handed β -sheet twist.
8. The hydrophobic cluster between the strands contributed by Trp, Val, Phe, and Tyr residues vastly stabilize the β -sheet proteins.
9. As a general rule of thumb, incorporating disulfide bonds stabilize the protein with β -hairpin structures.

Relatively speaking, there are fewer examples of β -sheet proteins designed *de novo*. However, α/β mixed structures have been designed with great success (Struthers *et al.*, 1996). For example, Top7 by David Baker group and a $\beta\beta\alpha$ motif structure that is similar to a zinc-finger, which has a β -hairpin structure (Kuhlman *et al.*, 2003).

Tools Currently Available for Protein *de novo* Design

Irrespective of topology of the intended protein to be designed, some tools are listed below that are routinely used for *de novo* protein design, and also keeping in mind the ease of use from user's perspective.

Rosetta and Rosetta Design

Among the tools that are currently available, Rosetta ([Simons *et al.*, 1999](#)), developed by David Baker group at University of Washington, has been the most popular and widely used ([Jiang *et al.*, 2008](#); [Siegel *et al.*, 2010](#); [Damborsky and Brezovsky, 2014](#)). The suite of software (see Relevant Websites Section) has sped the design of protein structures. Details of Rosetta's design algorithm and specifics are discussed in [Simons *et al.* \(1997, 1999\)](#), [Raveh *et al.* \(2010\)](#) and [DiMaio *et al.*, \(2011\)](#) and they have been reviewed elsewhere ([Mandell and Kortemme, 2009](#); [Der and Kuhlman, 2011](#)). Specifically, Rosetta Design (see Relevant Websites Section) can be accessed via the command line interface of Rosetta or via webserver ([Lyskov *et al.*, 2013](#)).

Evodesign

Evodesign (see Relevant Websites Section) developed by Zhang group at University of Michigan is a web based server to design protein sequences using a scaffold as an input. The scaffold is searched against the known protein families and the resulting conformation takes into consideration local environmental factors, such as solvent accessibility, packing, and secondary structure ([Mitra *et al.*, 2013](#)).

Protein WISDOM

Protein Wisdom (see Relevant Websites Section) is a web server tool for designing proteins from sequence information. The design and validation involves two steps: using either a rigid or flexible scaffold (or template), a sequence is selected from a pool of candidate sequences; and validation is done by fitting the sequence into known folds to calculate the "fold specificity" ([Smadbeck *et al.*, 2013](#)).

OSPREDY

Open Source Protein REdesign for You (OSPREDY) (see Relevant Websites Section) like Rosetta is a suite of programs developed by Donald group at Duke University. While Rosetta is licensed free for academics, OSPREDY is open-source and freely available to download and use. OSPREDY uses protein flexibility to create low-energy corpus of structures to identify the globally optimal structure. From the user's perspective, OSPREDY runs as a standalone tool and is not available as a web-server ([Gainza *et al.*, 2013](#)).

ISAMBARD

Intelligent System for Analysis, Model Building And Rational Design (ISAMBARD) (see Relevant Websites Section) is another open-source suite of software for designing proteins developed by Woflsoon group at University of Bristol. Keeping with the popularity of Python, ISAMBARD uses predefined python object based method to design protein structures. Those with a basic python skill will be able to use this modular and scalable software to design protein structures ([Wood *et al.*, 2017](#)).

FireProt

Rather than a general protein design tool, FireProt (see Relevant Websites Section) is a specific protein design tool for designing multiple-point mutant proteins that are likely to be thermostable ([Musil *et al.*, 2017](#)).

iRDP

Similar to FireProt, in-silico Rational Design of Proteins (iRDP) (see Relevant Websites Section) is a webserver that uses a four-step approach to rationally design proteins. Specifically, from the input it compares existing protein structures for structural stability factors, followed by mutational analysis, and their impact to local environmental changes, and identifying the optimal structure that would have a higher thermostability than the previous structure ([Panigrahi *et al.*, 2015](#)).

IPRO

Iterative Protein Redesign and Optimization procedure (IPRO) (see Relevant Websites Section) from the Maranas group at Pennsylvania State University, uses a combinatorial approach to redesign a protein library using energy based scoring functions. As claimed by the developers, it uses an iterative process to make additive mutations that improve the designed protein's substrate specificity ([Saraf *et al.*, 2006](#); [Fazelinia *et al.*, 2007](#)).

Scaffold Selection

Keeping in view that new protein can be designed from pre-existing structures; ScaffoldSelection (see Relevant Websites Section) is a tool that scans large sets of structures for a particular reaction scaffold (Malisi *et al.*, 2009). For the user, the tool can be downloaded as binaries for Linux, Mac, and Windows operating systems and run as a standalone software/tool.

Pocketoptimizer

Pocketoptimizer (see Relevant Websites Section) is an allied tool that can be used to design active site region, either to improve or modify ligand/substrate binding (Stiel *et al.*, 2016).

Conclusions

Protein design is an active, exciting area of research that has wide applications in drug design, medicine, and advancing the study of protein folding. When designing protein structures, one has to ask few questions and these questions drive or direct the use of tools to answer those questions.

1. Is the aim to engineer a protein's function/activity or designing a structure *de novo*?
2. Is the design fitting with the already known do's and don't's?
3. Is there another variable that is specific for the protein and its intended use?

There can be more additional questions added and the framework/checklist can act as a guide to the specific task in hand.

While, there are many success stories in *de novo* protein design, there are some challenges for the road ahead. Some of the challenges that have been discussed (Kuhlman *et al.*, 2009) are:

1. Sampling the conformational space of the backbone to move towards a completely flexible backbone based design. Currently, majority of the *de novo* designed proteins are on the basis of having a rigid scaffold or frame throughout the design pipeline.
2. To reduce the presence of alternate conformations from the desired/designed conformation.
3. Designing specific protein-protein interactions using a much easier and faster method.

Recently, Rocklin *et al.* from David baker's lab have used minimal proteins (proteins having a residue length below 50 amino acids) to understand the factors that determine protein folding. Specifically, they used computationally driven approach, where four topologies ($\alpha\alpha\alpha$, $\beta\alpha\beta\beta$, $\alpha\beta\beta\alpha$, and $\beta\beta\alpha\beta\beta$) were designed using 5000–40,000 sequences for each topology. Using yeast based proteolysis assay, a stability score was given to each designed protein, which enabled them to identify 2788 stable proteins. Such "massively-parallel" design has indeed pushed the limits of high-throughput design and the method can be applied to proteins of more than 50 amino acids in length (Rocklin *et al.*, 2017).

Acknowledgement

RMY acknowledges the infrastructural support given by Jaypee University of Information Technology, Waknaghat for completing this manuscript. Authorship was decided using a tic-tac-toe game with self. He also acknowledges Pulkit Anupam Srivastava for critical reading of the manuscript.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Computational Protein Engineering Approaches for Effective Design of New Molecules. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition. Study of The Variability of The Native Protein Structure

References

- Baxa, M., Haddadian, E., Jumper, J., Freed, K., Sosnick, T., 2014. Loss of conformational entropy in protein folding calculated using realistic ensembles and its implications for NMR-based calculations. *Proceedings of the National Academy of Sciences* 111 (43), 15396–15401.
- Bowie, J.U., Luthey, R., Eisenberg, D., 1991. A method to identify protein sequences that fold into a known three-dimensional structure. *Science (New York, NY)* 253 (5016), 164–170.
- Brown, J.E., Klee, W.A., 1971. Helix-coil transition of the isolated amino terminus of ribonuclease. *Biochemistry* 10 (3), 470–476.
- Crick, F.H.C., 1953. The packing of $\{\alpha\}$ -helices: Simple coiled-coils. *Acta Crystallographica* 6 (8–9), 689–697. doi:10.1107/S0365110X53001964.
- Dahiyat, B.I., Mayo, S.L., 1997. *De novo* protein design: Fully automated sequence selection. *Science (New York, NY)* 278 (5335), 82–87.
- Damborsky, J., Brezovsky, J., 2014. Computational tools for designing and engineering enzymes. *Current Opinion in Chemical Biology* 19, 8–16. doi:10.1016/j.cbpa.2013.12.003.
- de Alba, E., *et al.*, 1996. Conformational investigation of designed short linear peptides able to fold into beta-hairpin structures in aqueous solution. *Folding & Design* 1 (2), 133–144.
- de Alba, E., *et al.*, 1999. *De novo* design of a monomeric three-stranded antiparallel beta-sheet. *Protein Science : A Publication of the Protein Society* 8 (4), 854–865. doi:10.1110/ps.8.4.854.

- DeGrado, W.F., Wasserman, Z.R., Lear, J.D., 1989. Protein design, a minimalist approach. *Science* (New York, NY) 243 (4891), 622–628.
- Der, B.S., Kuhlman, B., 2011. Biochemistry. From computational design to a protein that binds. *Science* (New York, NY) 332 (6031), 801–802. doi:10.1126/science.1207082.
- DiMaio, F., et al., 2011. Modeling symmetric macromolecular structures in Rosetta3. *PLOS ONE* 6 (6), e20450. doi:10.1371/journal.pone.0020450.
- Fazelinia, H., Cirino, P.C., Maranas, C.D., 2007. Extending Iterative Protein Redesign and Optimization (IPRO) in protein library design for ligand specificity. *Biophysical Journal* 92 (6), 2120–2130. doi:10.1529/biophysj.106.096016.
- Gainza, P., et al., 2013. OSPREY: Protein design with ensembles, flexibility, and provable algorithms. *Methods in Enzymology* 523, 87–107. doi:10.1016/B978-0-12-394292-0.00005-9.
- Griffiths-Jones, S.R., Searle, M.S., 2000. Structure, folding, and energetics of cooperative interactions between the β -Strands of a de novo designed three-stranded antiparallel β -sheet peptide. *Journal of the American Chemical Society* 122 (35), 8350–8356. doi:10.1021/ja000787t.
- Huyghues-Despointes, B.M., Scholtz, J.M., Baldwin, R.L., 1993. Effect of a single aspartate on helix stability at different positions in a neutral alanine-based peptide. *Protein Science: A Publication of the Protein Society* 2 (10), 1604–1611. doi:10.1002/pro.5560021006.
- Jiang, L., et al., 2008. De novo computational design of retro-aldol enzymes. *Science* (New York, NY) 319 (5868), 1387–1391. doi:10.1126/science.1152692.
- Kim, P.S., Baldwin, R.L., 1984. A helix stop signal in the isolated S-peptide of ribonuclease A. *Nature* 307 (5949), 329–334.
- Kortemme, T., Ramirez-Alvarado, M., Serrano, L., 1998. Design of a 20-amino acid, three-stranded beta-sheet protein. *Science* (New York, NY) 281 (5374), 253–256.
- Kuhlman, B., et al., 2003. Design of a novel globular protein fold with atomic-level accuracy. *Science* (New York, NY) 302 (5649), 1364–1368. doi:10.1126/science.10894270.
- Kuhlman, B., Jung Choi, E., Guntas, G., 2009. Future challenges of computational protein design. In: *Protein Engineering and Design*, CRC Press. Available at: <http://dx.doi.org/10.1201/9781420076592.ch18>.
- Lacroix, E., Viguera, A.R., Serrano, L., 1998. Elucidating the folding problem of alpha-helices: Local motifs, long-range electrostatics, ionic-strength dependence and prediction of NMR parameters. *Journal of Molecular Biology* 284 (1), 173–191. doi:10.1006/jmbi.1998.2145.
- Leitner, A., Walzthoeni, T., Kahraman, A., et al., 2010. Probing native protein structures by chemical cross-linking, mass spectrometry, and bioinformatics. *Molecular & Cellular Proteomics* 9 (8), 1634–1649.
- Lyskov, S., et al., 2013. Serverification of molecular modeling applications: The rosetta online server that includes everyone (ROSIE). *PLOS ONE* 8 (5), e63906. doi:10.1371/journal.pone.0063906.
- Lyu, P.C., et al., 1990. Side chain contributions to the stability of alpha-helical structure in peptides. *Science* (New York, NY) 250 (4981), 669–673.
- Malisi, C., Kohlbacher, O., Hocker, B., 2009. Automated scaffold selection for enzyme design. *Proteins* 77 (1), 74–83. doi:10.1002/prot.22418.
- Mandell, D.J., Kortemme, T., 2009. Backbone flexibility in computational protein design. *Current Opinion in Biotechnology* 20 (4), 420–428. doi:10.1016/j.copbio.2009.07.006.
- Marqusee, S., Baldwin, R.L., 1987. Helix stabilization by Glu-Lys+ salt bridges in short peptides of de novo design. *Proceedings of the National Academy of Sciences of the United States of America* 84, 8988–8992.
- Marqusee, S., Robbins, V.H., Baldwin, R.L., 1989. Unusually stable helix formation in short alanine-based peptides. *Proceedings of the National Academy of Sciences of the United States of America* 86 (14), 5286–5290.
- Mitra, P., Shultis, D., Zhang, Y., 2013. EvoDesign: de novo protein design based on structural and evolutionary profiles. *Nucleic Acids Research* 41, W273–W280. doi:10.1093/nar/gkt384.
- Munoz, V., Serrano, L., 1994. Elucidating the folding problem of helical peptides using empirical parameters. *Nature Structural Biology* 1 (6), 399–409.
- Munoz, V., Serrano, L., 1995a. Elucidating the folding problem of helical peptides using empirical parameters. II. Helix macrodipole effects and rational modification of the helical content of natural peptides. *Journal of Molecular Biology* 245 (3), 275–296.
- Munoz, V., Serrano, L., 1995b. Elucidating the folding problem of helical peptides using empirical parameters. III. Temperature and pH dependence. *Journal of Molecular Biology* 245 (3), 297–308. doi:10.1006/jmbi.1994.0024.
- Munoz, V., Serrano, L., 1997. Development of the multiple sequence approximation within the AGADIR model of alpha-helix formation: Comparison with Zimm-Bragg and Lifson-Roig formalisms. *Biopolymers* 41 (5), 495–509. doi:10.1002/(SICI)1097-0282(19970415)41:5 < 495::AID-BIP2 > 3.0.CO;2-H.
- Musil, M., et al., 2017. FireProt: Web server for automated design of thermostable proteins. *Nucleic Acids Research*. doi:10.1093/nar/gkx285.
- Nicholson, H., Becktel, W.J., Matthews, B.W., 1988. Enhanced protein thermostability from designed mutations that interact with alpha-helix dipoles. *Nature* 336 (6200), 651–656. doi:10.1038/336651a0.
- Panigrahi, P., et al., 2015. Engineering proteins for thermostability with iRDP web server. *PLOS ONE* 10 (10), e0139486. doi:10.1371/journal.pone.0139486.
- Park, S.H., Shalongo, W., Stellwagen, E., 1993. Residue helix parameters obtained from dichroic analysis of peptides of defined sequence. *Biochemistry* 32 (27), 7048–7053.
- Pauling, L., Corey, R.B., 1953. Compound helical configurations of polypeptide chains: Structure of proteins of the alpha-keratin type. *Nature* 171 (4341), 59–61.
- Ramirez-Alvarado, M., Blanco, F.J., Serrano, L., 1996. De novo design and structural analysis of a model beta-hairpin peptide system. *Nature Structural Biology* 3 (7), 604–612.
- Raveh, B., London, N., Schueler-Furman, O., 2010. Sub-angstrom modeling of complexes between flexible peptides and globular proteins. *Proteins* 78 (9), 2029–2040. doi:10.1002/prot.22716.
- Richardson, J.S., 1981. The anatomy and taxonomy of protein structure. *Advances in Protein Chemistry* 34, 167–339.
- Richardson, J.S., Richardson, D.C., 1988. Amino acid preferences for specific locations at the ends of alpha helices. *Science* (New York, NY) 240 (4859), 1648–1652.
- Richardson, J.S., Richardson, D.C., 1989. The de novo design of protein structures. *Trends in Biochemical Sciences* 14 (7), 304–309.
- Rocklin, G.J., et al., 2017. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* (New York, NY) 357 (6347), 168–175. doi:10.1126/science.aan0693.
- Saraf, M.C., et al., 2006. IPRO: An iterative computational protein library redesign and optimization procedure. *Biophysical Journal* 90 (11), 4167–4180. doi:10.1529/biophysj.105.079277.
- Schafmeister, C.E., et al., 1997. A designed four helix bundle protein with native-like structure. *Nature Structural Biology* 4 (12), 1039–1046.
- Schenck, H.L., Gellman, S.H., 1998. Use of a designed triple-stranded antiparallel beta-Sheet to probe beta-sheet cooperativity in aqueous solution. *Journal of the American Chemical Society* 120 (11), 4869–4870. doi:10.1021/ja973984+.
- Scholtz, J.M., et al., 1993. The energetics of ion-pair and hydrogen-bonding interactions in a helical peptide. *Biochemistry* 32 (37), 9668–9676.
- Sharman, G.J., Searle, M.S., 1998. Cooperative interaction between the three strands of a designed antiparallel β -sheet. *Journal of the American Chemical Society* 120 (21), 5291–5300. doi:10.1021/ja9705405.
- Shoemaker, K.R., et al., 1987. Tests of the helix dipole model for stabilization of alpha-helices. *Nature* 326 (6113), 563–567. doi:10.1038/326563a0.
- Siegel, J.B., et al., 2010. Computational design of an enzyme catalyst for a stereoselective bimolecular Diels-Alder reaction. *Science* (New York, NY) 329 (5989), 309–313. doi:10.1126/science.1190239.
- Simons, K.T., et al., 1997. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology* 268 (1), 209–225. doi:10.1006/jmbi.1997.0959.
- Simons, K.T., et al., 1999. Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins* 34 (1), 82–95.
- Sinz, A., Arit, C., Chorev, D., Sharon, M., 2015. Chemical cross-linking and native mass spectrometry: A fruitful combination for structural biology. *Protein Science* 24 (8), 1193–1209.
- Smadbeck, J., et al., 2013. Protein WISDOM: A workbench for in silico de novo design of biomolecules. *Journal of Visualized Experiments: JoVE* 77. doi:10.3791/50476.
- Stanger, H.E., Gellman, S.H., 1998. Rules for antiparallel β -sheet design: D-pro-gly is superior to L-Asn-Gly for β -hairpin nucleation. *Journal of the American Chemical Society* 120 (17), 4236–4237. doi:10.1021/JA973704Q.
- Stiel, A.C., Nellen, M., Hocker, B., 2016. PocketOptimizer and the design of ligand binding sites. *Methods in Molecular Biology Clifton NJ* 1414, 63–75. doi:10.1007/978-1-4939-3569-7_5.
- Struthers, M.D., Cheng, R.P., Imperiali, B., 1996. Design of a monomeric 23-residue polypeptide with defined tertiary structure. *Science* (New York, NY) 271 (5247), 342–345.

- Wetzel, R., 1987. Harnessing disulfide bonds using protein engineering. *Trends in Biochemical Sciences* 12 (Suppl.), 478–482. Available at: [https://doi.org/10.1016/0968-0004\(87\)90234-9](https://doi.org/10.1016/0968-0004(87)90234-9).
- Wood, C.W., *et al.*, 2017. ISAMBARD: An open-source computational environment for biomolecular analysis, modelling and design. *Bioinformatics* (Oxford). doi:10.1093/bioinformatics/btx352.
- Xiong, H., *et al.*, 1995. Periodicity of polar and nonpolar amino acids is the major determinant of secondary structure in self-assembling oligomeric peptides. *Proceedings of the National Academy of Sciences of the United States of America* 92 (14), 6349–6353.

Further Reading

- Baldwin, R.L., 1995. Alpha-helix formation by peptides of defined sequence. *Biophysical Chemistry* 55 (1–2), 127–135.
- Barlow, D.J., Thornton, J.M., 1988. Helix geometry in proteins. *Journal of Molecular Biology* 201 (3), 601–619.
- Blanco, F., Ramirez-Alvarado, M., Serrano, L., 1998. Formation and stability of beta-hairpin structures in polypeptides. *Current Opinion in Structural Biology* 8 (1), 107–111.
- Carey, P., 2008. *Protein Engineering and Design*. San Diego, CA: Academic Press.
- Chakrabarty, A., Baldwin, R.L., 1995. Stability of alpha-helices. *Advances in Protein Chemistry* 46, 141–176.
- Fisk, J.D., Gellman, S.H., 2001. A parallel beta-sheet model system that folds in water. *Journal of the American Chemical Society*. 343–344.
- Gueriois, R., De la Paz, M., 2006. *Protein Design Methods in Molecular Biology*, 340. Springer.
- Jensen, K., 2010. *Peptide and Protein Design for Biopharmaceutical Applications*. Chichester: Wiley.
- Lacroix, E., *et al.*, 1999. The design of linear peptides that fold as monomeric beta-sheet structures. *Current Opinion in Structural Biology* 9 (4), 487–493.
- Nowick, J.S., Cary, J.M., Tsai, J.H., 2001. A triply templated artificial beta-sheet. *Journal of the American Chemical Society* 123 (22), 5176–5180.
- Park, S., Cochran, J., 2010. *Protein Engineering and Design*. Boca Raton: CRC Press.
- Presta, L.G., Rose, G.D., 1988. Helix signals in proteins. *Science* (New York, NY) 240 (4859), 1632–1641.
- Richardson, J.S., Richardson, D.C., 1988. Amino acid preferences for specific locations at the ends of alpha helices. *Science* New York, NY) 240 (4859), 1648–1652.
- Rohl, C.A., Baldwin, R.L., 1998. Deciphering rules of helix stability in peptides. *Methods in Enzymology* 295, 1–26.
- Scholtz, J.M., Baldwin, R.L., 1992. The mechanism of alpha-helix formation by peptides. *Annual Review of Biophysics and Biomolecular Structure* 21, 95–118. doi:10.1146/annurev.bb.21.060192.000523.
- Searle, M.S., Ciani, B., 2004. Design of beta-sheet systems for understanding the thermodynamics and kinetics of protein folding. *Current Opinion in Structural Biology* 14 (4), 458–464. doi:10.1016/j.sbi.2004.06.001.
- Serrano, L., 2000. The relationship between sequence and structure in elementary folding units. *Advances in Protein Chemistry* 53, 49–85.
- Venktraman, J., Shankaramma, S.C., Balaram, P., 2001. Design of folded peptides. *Chemical Reviews* 101 (10), 3131–3152.

Relevant Websites

- <http://agadir.crg.es/>
Adagir.
- <http://www.cs.duke.edu/donaldlab/osprey.php>
Donald lab.
- <http://www.grigoryanlab.org/drawcoil>
Drawcoil 1.0.
- <http://r2lab.ucr.edu/scripts/wheel/wheel.cgi>
Helical Wheel Projections - RZLab.
- <http://irfp.ncl.res.in>
iRDP.
- <https://loschmidt.chemi.muni.cz/fireprot>
LOSCHMIDT laboratories.
- <http://www.eb.tuebingen.mpg.de/research/research-groups/birte-hoecker/algorithms-and-software/pocketoptimizer.html>
Max Planck Institute for Developmental Biology.
- <http://www.eb.tuebingen.mpg.de/research/research-groups/birte-hoecker/algorithms-and-software/scaffoldselection.html>
Max Planck Institute for Developmental Biology.
- http://maranas.che.psu.edu/submission/IPRO_2.htm
PENNSTATE.
- <http://embooss.bioinformatics.nl/cgi-bin/embooss/pepwheel>
Pepwheel.
- <http://atlas.princeton.edu/proteinwisdom>
Protein WISDOM.
- <https://pypi.python.org/pypi/isambard>
Python.
- <https://www.rosettacommons.org/>
Rosetta commons.
- <http://rosettadesign.med.unc.edu>
Rosetta commons.
- <http://rosie.rosettacommons.org/>
Rosetta commons.
- <https://zhanglab.cmb.med.umich.edu/EvoDesign>
Zhang lab.

Biographical Sketch



Ragothaman Yennamalli is an Assistant Professor at Jaypee University of Information Technology, Wagnaghat, Himachal Pradesh, India. He completed his PhD in Computational Biology and Bioinformatics from Jawaharlal Nehru University, New Delhi. He conducted postdoctoral research at Jawaharlal Nehru University in India and at Iowa State University (2009–2011), University of Wisconsin-Madison (2011–2012), and Rice University (2012–2014) in USA. He is a structural and computational biologist with more than a decade of experience in predictive modelling and biomolecular simulation projects. Dr. Yennamalli's skills involve molecular docking, molecular dynamics simulation, coarse grained modelling, machine learning, data mining, data analytics, and molecular modelling. Website: http://bit.ly/raghu_juit.

Molecular Dynamics Simulations in Drug Discovery

Sy-Bing Choi, Beow Keat Yap, Yee Siew Choong, and Habibah Wahab, Universiti Sains Malaysia, Pulau Pinang, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Background

The need for reduction of cost and time had led to the emergence of computer-aided drug design technology in drug discovery pipeline. While molecular docking and other high throughput virtual screening have played significant roles in screening large libraries of compounds, atomistic Molecular Dynamics (MD) simulation, is increasingly gaining trust in guiding the prediction of comprehensive drug-target binding interactions with the cognate protein receptors, which are important in successful hit discovery and optimisation. The classical 'lock-and-key' theory of ligand binding where the lock is a rigid protein receptor complemented by small molecules which act as keys, is no longer able to account for the drug binding event, as conformational changes upon ligand binding to receptor proteins are well known. These conformational changes range from modest loop motion to hinge bending (Jorgensen, 1991). Most docking programmes can only give a static picture ('lock-and-key') of what is happening during drug binding. Thus, MD simulations are the method of choice, to study drug-target binding allowing conformational changes to be investigated further (De Vivo *et al.*, 2016).

In a typical all-atom MD simulation, the movements of all the atoms of a drug and its target protein (or other macromolecule) in physiological solvent environment are simulated over time (lasting for hundreds of nanoseconds or longer) in a series of discrete short time steps (e.g., 2 fs). At each time step, the forces on each atom are computed (based on force field method) from the atomic position and velocity according to Newton's Laws of Motions. The trajectories obtained will be used to study the dynamics and behaviours of the system as well as making prediction on the binding affinity of the ligands to the receptor. The use of molecular dynamics (MD) simulation along with the current advancement in computational technology allows such valuable prediction to be obtained with reasonable cost and time. The applications of MD simulation in drug discovery is wide ranging, from the investigation of protein-ligand binding and unbinding to mutation effects for this binding and computing the free energy of binding. The application of molecular dynamics in relation to drug discovery in general have been reviewed previously (De Vivo *et al.*, 2016; Durrant and McCammon, 2011; Borhani and Shaw, 2012). In this review, we focus specifically on the contribution of MD simulations in the attempt to battle tuberculosis (TB), one of the ancient deadly diseases that still persist today.

Mycobacterium tuberculosis (MTB), the causative agent of human tuberculosis (TB), is a bacterial pathogen, listed as one of the top ten causes of death worldwide. Currently, it is the leading cause of death from a single infectious agent surpassing even human immunodeficiency virus (HIV) with 1.4 million people killed per year and an estimated 2 billion people are latently infected worldwide. In 2016, an estimated 10.4 million people developed active TB (WHO, 2017). MTB (Fig. 1) has been known to infect human from thousands of years. In general, there are two phases in MTB life cycle, i.e., the replication phase (active TB) and the dormant phase (inactive TB) (Fig. 2). Both phases involve different metabolic pathways, with carbohydrate as the main source in active TB phase, and lipid as sole carbon source in dormant phase. To date, most drugs available for MTB infection are only targeting the replication phase of the MTB infection. However, there is an imminent need for finding effective drugs that can target MTB in the dormant phase, as 95% of current MTB infection is becoming latent.

MTB has a remarkable ability to avoid the attacks of host's immune system and has developed sophisticated strategies for successful infection that make it challenging to treat the disease. The current MTB therapy is a two-month intensive phase of a four-drug combinations of isoniazid (INH), rifampicin (RIF), pyrazinamide (PZA) and ethambutol (EMB) followed by a longer continuation phase of INH and RIF to eradicate the remaining bacilli that have entered a dormant, slowly-replicating latent phase.

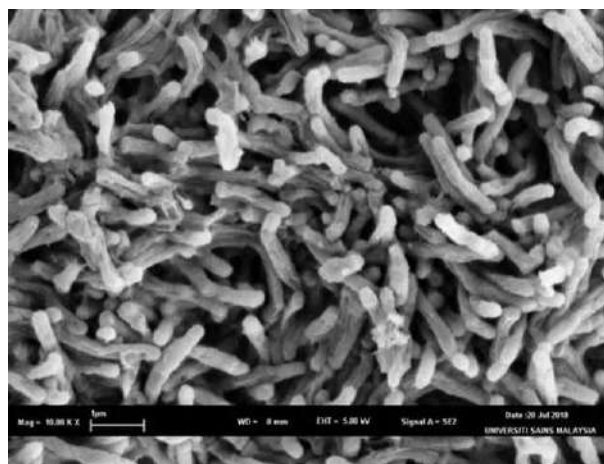


Fig. 1 Scanning electron microscope (SEM) image of H37Rv strain (ATCC 25618) of *Mycobacterium tuberculosis*.

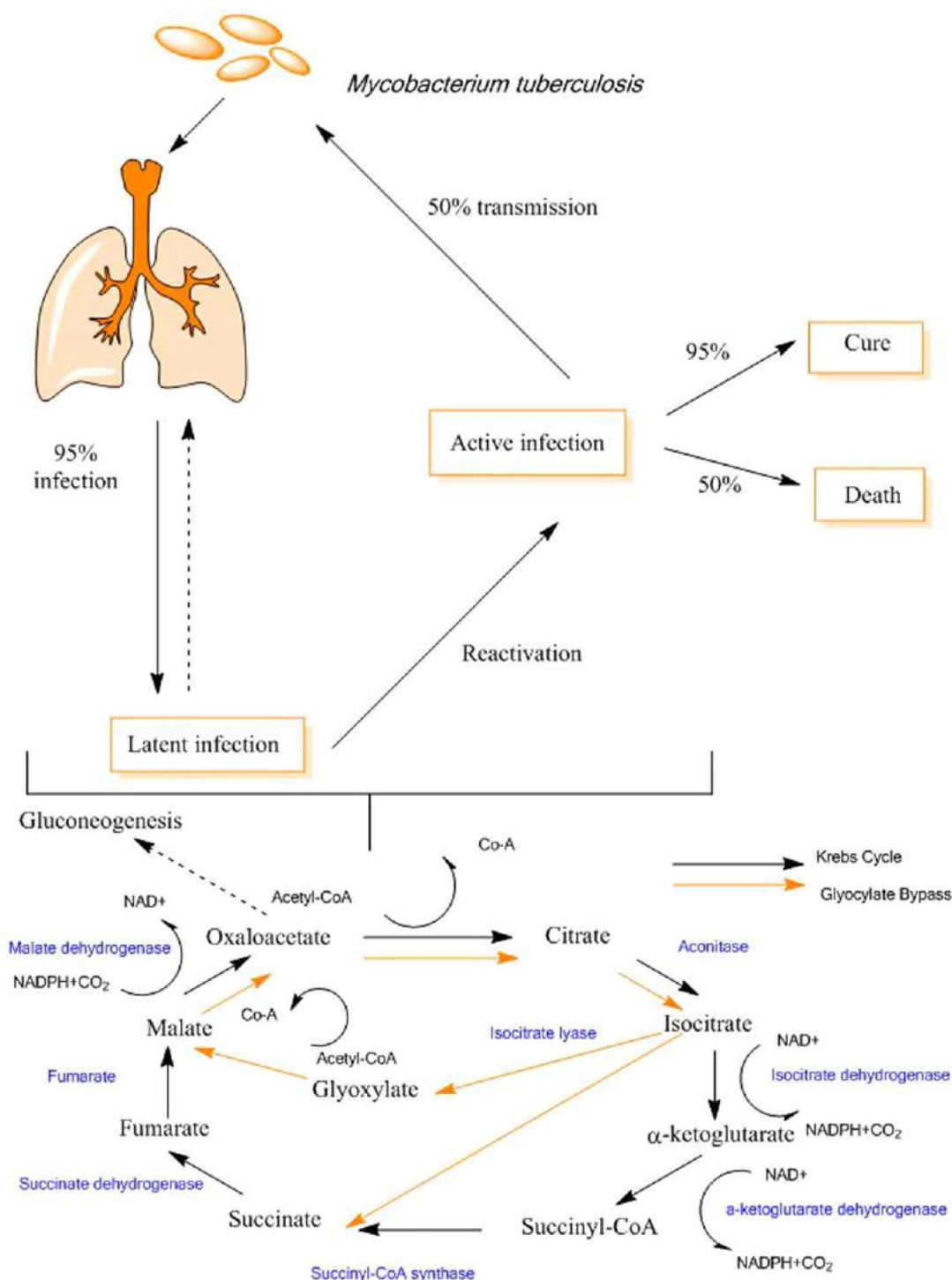


Fig. 2 Stages of transmission and progression of MTB infections. Two different metabolic pathways used by MTB in replication and dormant phase of MTB infection are shown.

Fig. 3 shows the targets of these drugs in MTB. The emergence of multidrug-resistant MDR-TB strains (resistance to RIF and INH) complicates the therapy as the treatment can extend up to 2 years and requires more toxic, less efficacious second- or third-line agents such as fluoroquinolone and kanamycin. The appearance of the extensively drug-resistant tuberculosis (XDR-TB) and totally drug-resistant tuberculosis (TDR-TB) have made the treatment even more difficult (Hoagland *et al.*, 2016) and expensive. The average cost of treating a TB patient increases with resistance; from \$18,000 to treat drug-susceptible TB to \$513,000 to treat the most drug-resistant form of the disease (XDR TB). The therapy takes a long time to complete, disrupts lives, and has potentially serious and side effects such as depression or psychosis, hearing loss, hepatitis, and kidney impairment (CDC, 2017).

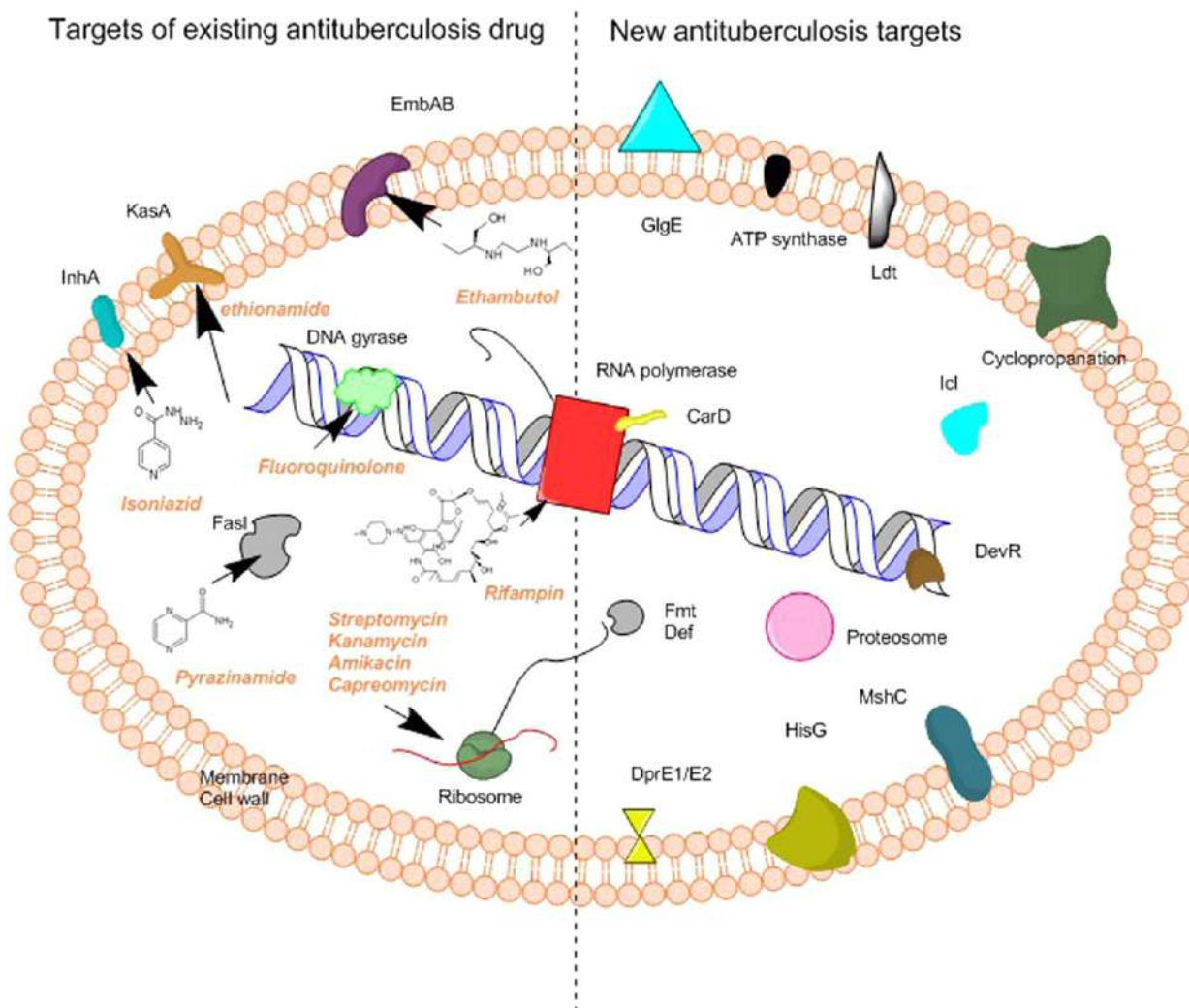


Fig. 3 The targets of the existing antitubercular drugs and the newly discovered targets of MTB.

Understanding the basic biology of the pathogen that underlie the development and spread of MDR TB is as equally important as developing novel effective antitubercular for drug-susceptible and drug-resistant TB in the battle with this disease. In the former case, one really needs to look at the resistance mechanism at the molecular level as it not only improves our understanding of the pathogenesis of MTB but hopefully to prevent the development of drug resistance in the future. Similarly, in order to develop novel antitubercular, one also requires deep understanding of drug-target interactions such that the agent will be more active and selective.

The main challenges faced towards developing anti-TB drug include the organism's slow replication, the ability of *M. tuberculosis* to survive in a dormant state and its capability to develop resistance against drugs, especially towards isoniazid (INH), the first line drug for the treatment of TB. INH is a prodrug, catalysed by a catalase peroxidase enzyme (KatG), that produces the ultimate inhibitor for enoyl-acyl carrier protein reductase (InhA). InhA is involved in the synthesis of mycolic acid components of the outer mycobacterial cell (Telenti *et al.*, 1997; Wang *et al.*, 1998; Fang *et al.*, 1999; Vilcheze *et al.*, 2000). Mutation(s) in *katG* (Banerjee *et al.*, 1994) has been associated with high level of INH resistance. Besides *katG*, mutations in other MTB genes, such as *inhA*, *ahpC*, *kasA* and *ndh* are also involved in the INH resistance (Vilcheze *et al.*, 2000; Piatek *et al.*, 2000; Slayden and Barry, 2000; Torres *et al.*, 2000) with the mutations within *inhA* have been reported up to 32% in INH-resistant isolates (Heym *et al.*, 1995; Morris *et al.*, 1995; Lee *et al.*, 1999; Kiepiela *et al.*, 2000). Beside INH, MTB has been shown to develop resistance towards other TB drugs. Table 1 lists some examples of current tuberculosis drugs and their targets as well as genes associated with their resistance (Telenti, 1998; Zhang *et al.*, 2006; Hards *et al.*, 2015; Lewis and Sloan, 2015).

To date, in the quest for developing new active antitubercular, there have been many new drug targets identified alongside the well established targets for MTB and these numbers are increasing as new technologies emerged. Recent reviews (Wellington and Hung, 2018; Ferraris *et al.*, 2018; Hoagland *et al.*, 2016) have provided interesting snapshots of some of these molecular targets which are involved in vital cellular biological processes such as cell wall synthesis, energy metabolism, protein synthesis, phosphate transport and the metabolism of key molecules and cofactors. Here, we aim to provide dynamic snapshots of these molecular targets using MD simulations to further improve our understanding of the action and resistance mechanisms of various

Table 1 Current tuberculosis drugs, their molecular targets and gene(s) associated with their resistance

Drug	Mechanism of action	Target	Gene(s) involved in resistance
Isoniazid	Inhibition of cell wall mycolic acid synthesis	Enoyl acyl carrier protein reductase (InhA)	katG, inhA,
Rifampicin	Inhibition of RNA synthesis	RNA polymerase, subunit B (rpoB)	rpoB
Pyrazinamide	Depletion of membrane energy	Membrane energy metabolism, precise target still debatable	pncA, panD, rpsA
Ethambutol	Inhibition of cell wall synthesis	Arabisoyl transferase	embCAB
Streptomycin	Inhibition of protein synthesis	Ribosomal S12 protein and 16SrRNA	rpsL, rrs
Kanamycin	Inhibition of protein synthesis	16SrRNA	Rrs
Capreomycin	Inhibition of protein synthesis	16SrRNA, 50Sribosome, rRNA methyltransferase (TlyA)	Rrs, tlyA
Fluoroquinolones	Inhibition of DNA synthesis	DNA gyrase	gyrA, gyrB
Ethionamide	Inhibition of mycolic acid synthesis	Acyl carrier protein reductase (InhA)	inhA, etaA/ethA
Para-amino salicylic acid	Inhibition of folate pathway	Dihydrofolate reductase, dhfrA	thyA, dhfrA
Bedaquiline	Inhibition of ATP synthase	The proton pump of mycobacterial ATP (adenosine 5'-triphosphate) synthase	atpE
Delamanid	Inhibition of mycobacterial cell wall components, methoxy mycolic acid and ketomycolic acid synthesis	Deazaflavin dependent nitroreductase	fgd, Rv3547, fbiA, fbiB, and fbiC

Note: Reproduced from Telenti, A., 1998. Genetics of drug resistant tuberculosis. *Thorax* 53, 793–797. Zhang, Y., Post-Martens, K., Denkin, S., 2006. New drug candidates and therapeutic targets for tuberculosis therapy. *Drug Discov. Today* 11, 21–27. Hards, K., Robson, J.R., Berney, M., *et al.*, 2015. Bactericidal mode of action of bedaquiline. *J. Antimicrob. Chemother.* 70, 2028–2037. Lewis, J.M., Sloan, D.J., 2015. The role of delamanid in the treatment of drug-resistant tuberculosis. *Ther. Clin. Risk Manag.* 11, 779–791.

drugs towards their corresponding TB targets and how the results have been instrumental in the optimisation stage of drug discovery against existing and newly discovered targets.

Applications and Case Studies

MD simulations have been applied to investigate the structural dynamics of various drug targets and to unravel mechanistic information. Such information includes the effects of mutations, role of waters on the ligand binding as well as the prediction of potentially new ligand binding site. The methodology has also been used in *in silico* drug discovery campaign, from preparation of protein structure for structure-based drug design and discovery studies, to evaluation of the binding interactions, stability and free binding energies of protein-ligand complexes (Fig. 4). These applications in battling tuberculosis are discussed in greater detail below.

Elucidating the Structural Dynamics and Mechanisms of MTB Targets

The static models provided by docking simulations give valuable insights into macromolecular structure, but drug binding is a dynamic event. At physiological condition, macromolecules (targets) are in constant motion surrounded by solvents. Drug binding to the target typically results in the target's conformational shifts that is usually different in the absence of drug binding. This binding event, is governed by enthalpic contribution which stabilizes the interactions through the formation of hydrogen bonds, salt bridges and/or van der Waals interactions; as well as the entropic penalties from the loss of conformational freedom of the target and drug. Amino acid mutations occurring in many MTB drug targets, which have contributed to many clinically drug-resistant strains, may disrupt shape complementarity as well as physicochemical compatibility that affect the binding of drug to the target (Lahti *et al.*, 2012). All these dynamic events, can easily be studied using MD simulations.

The understanding of conformational changes of selected MTB targets is crucial in designing effective molecules for drug discovery. For this reason, *in silico* site-directed mutagenesis has been performed to understand how the functionally important binding site residues in targeted MTB enzymes affect the conformational changes as well as the binding of potential inhibitors. To illustrate, studies on an enzyme involved in glyoxylate pathway is chosen as an example. As dormant phase MTB is in non-replicating mode, MTB shifts its metabolism pathway to utilise fatty acids or lipids as the carbon source to survive in the macrophage. Therefore, the energy of MTB is produced via glyoxylate pathway to bypass the oxygen-dependent tricarboxylic acid cycle (also known as Krebs cycle). Shukla and co-workers revealed that H46A, L418A and F345A mutations in isocitrate lyase, an enzyme in the glyoxylate shunt, resulted in the loss of activity due to geometry alteration in the binding site. The altered active site geometry disturbed the enzyme transition state, rendering the enzyme inactive (Shukla *et al.*, 2018, 2017a,b; Fig. 5). A similar approach has also been used against different MTB targets such as pantothenate synthase (Pandey *et al.*, 2018), catalase peroxidase (Singh *et al.*, 2018; Srivastava *et al.*, 2017; Pimentel *et al.*, 2017; Unissa *et al.*, 2017), pyrazinamidase (Aggarwal *et al.*, 2018), RecA Mtu Intein (Zwarycz *et al.*, 2017), RNA polymerase (Singh *et al.*, 2017), RpoB protein (Nusrath Unissa *et al.*, 2016), where conformational stability as well as shrinking or enlarging the binding site cavity caused by mutations were studied with respect to their effect on individual functionality of MTB drug targets.

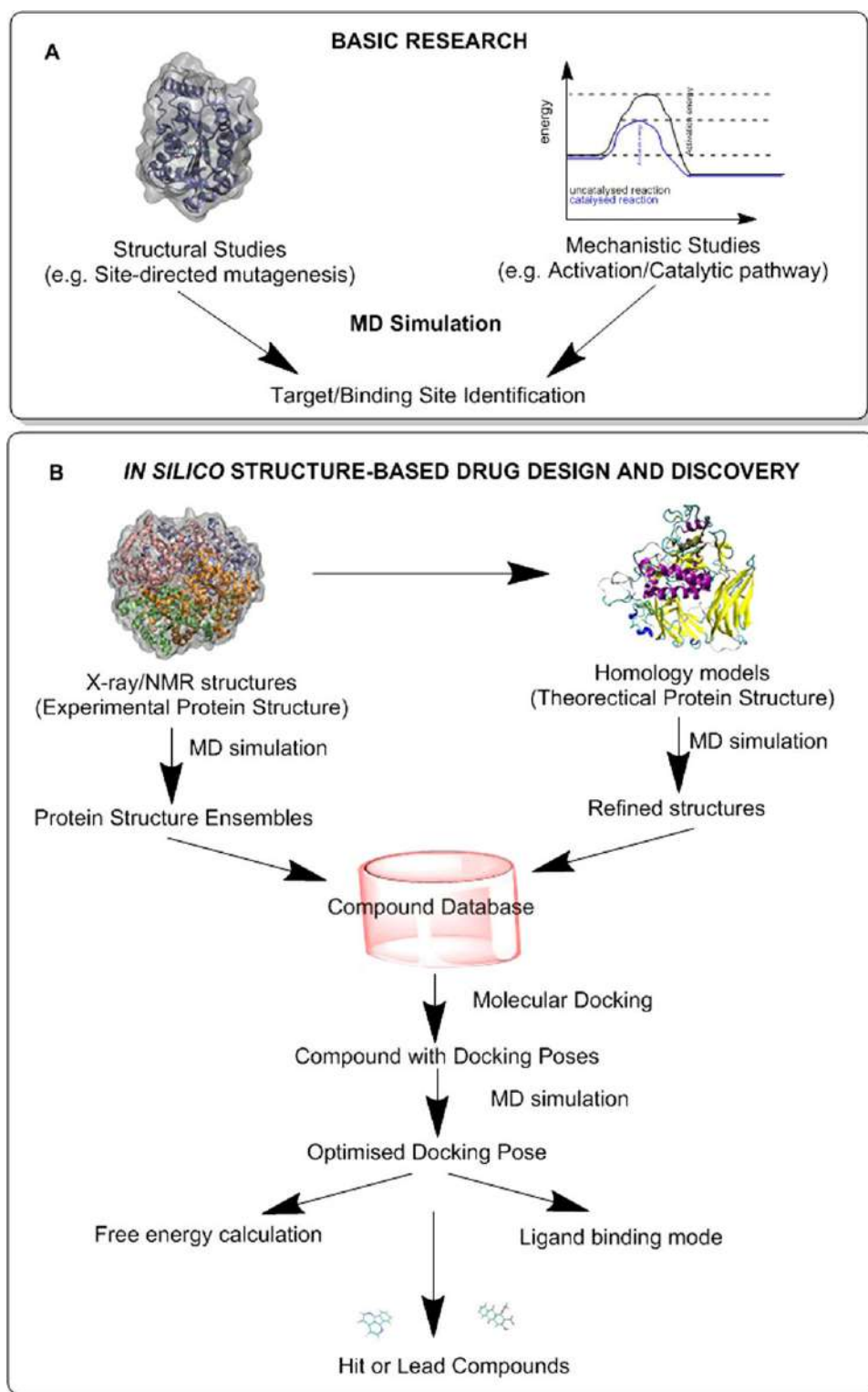


Fig. 4 General applications of MD simulations in (A) basic research and (B) *in silico* structure-based drug design and discovery.

Apart from studying the conformational changes upon mutation, MD has also been used to understand the functional behaviour of drug target in its different subunit form. To illustrate, Parker and co-workers revealed that disruption of the tetrameric conformation of 3-deoxy-D-arabino-heptulosonate 7-phosphate synthase (DAH7PS) by substitution of G232P to dimer resulted in changes in conformation as well as the polar and solvation properties by MD analysis. This ultimately resulted in the reduction of catalytic efficiency and the loss of allosteric response, suggesting the importance of maintaining

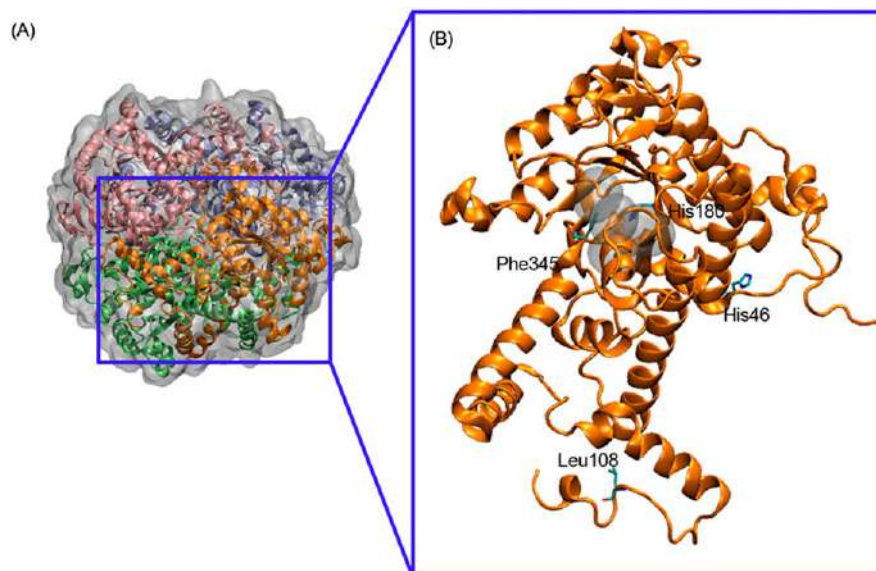


Fig. 5 (A) ICL In its tetrameric form (PDB ID: 1F8I). (B) Point mutations H46A, H180A, F345A and L108A constructed by Shukla *et al.* (2015, 2017a,b, 2018). L108 and H46 are distant from the ICL binding site (in surface representation).

the quaternary structure (i.e., the tetrameric form) for its function. This was further validated experimentally with enzyme kinetics study (Jiao *et al.*, 2017).

2-*trans*-enoyl-ACP (CoA) reductase (InhA) is involved in the biosynthesis of mycolic acid, an essential unique component of the MTB cell wall. This enzyme is the primary target of isoniazid (the first-line antituberculosis drug). Therefore, inhibition of InhA by isoniazid will hinder the synthesis of the MTB cell wall. MD simulations have also been used by Basso's group to investigate the crowding effect on 2-*trans*-enoyl-ACP (CoA) reductase (InhA) enzyme from MTB (Rotta *et al.*, 2017). As crowding effect is impossible to mimic in *in vitro* experiment, investigation of the crowding effect on the respective protein was performed *in silico*, by mimicking the cellular crowd (Ferreira *et al.*, 2016). Briefly, the behaviour of InhA in different solutions such as sucrose, glucose and PEG at different concentration is compared. Their study suggested that the flexibility of InhA and the variation in the volume of the active site is strongly related to the concentrations of the sucrose and glucose solutions used, with high rigidity and compactness of InhA shown on the B-factor interpretation in highly concentrated sucrose and glucose solutions mimicking the cellular environment. Based on these observations, Basso and co-workers suggested that the active site packing of InhA could interfere with processes such as substrate entrance and/or product release. The findings from this study also serve as a surrogate of *in vivo* intracellular medium for anti-tuberculosis drug screening campaigns (Rotta *et al.*, 2017).

In combination with QM/MM (hybrid quantum mechanics/molecular mechanics), MD simulation is also able to provide in-depth details of a chemical reaction. Yao *et al.* (2017) utilised QM/MM approach to study the catalytic mechanism and the nature of the rate-limiting step involving the shikimate kinase enzyme in the shikimate pathway which has recently been recognised as an antimicrobial drug target for MTB. The shikimate kinase pathway is used by plants and bacteria to synthesize shikimate, which is a common precursor in the synthesis of aromatic amino acids and secondary metabolites. Their results were found to correlate well with the experimental data where the phosphorylation transfer process, which is the rate-limiting step of SK-catalysed phosphorylation of shikimic acid (SKM), is a concerted one step reaction proceeding through a loose transition state.

Prediction of a New Ligand-Binding Site in MTB Targets

Predicting a new ligand binding sites on receptor structures, which are obtained either from experimental or computational methods, is a useful first step in structure-based drug design (Jian *et al.*, 2016). MD simulations have also been used to predict the binding site of isoniazid (INH) in MTB's KatG protein, in which its S315T mutation has been associated with INH resistance, presumably due to its reduced affinity (Yu *et al.*, 2003). INH is a prodrug which needs activation by the KatG protein to exert its action. Although the exact binding site of INH to KatG is unknown, it has been previously proposed that there are three potential binding sites of INH on KatG, close to the heme site, referred to as Site-1, Site-2 and Site-3 (Kamachi *et al.*, 2015a,b; Vidossich *et al.*, 2014). To pinpoint the exact binding site, a 100 ns MD simulations for each of the docked complex of INH at the three potential binding sites of the wild-type and S315T mutant KatG were carried out (Srivastava *et al.*, 2017). Interestingly, only Site-1 bound INH resulted in conformational change to the important heme group of KatG, suggesting that Site-1 is the main INH binding site. Moreover, further MD trajectories analysis revealed no significant change to RMSD, RMSF, Rg, hydrogen bond occupancy and solvent-accessible surface area (SASA) for both wild-type and mutant systems of all but Site-1 bound INH systems. In addition, subsequent estimation of INH binding energy with MM-PBSA from the stable MD trajectories of the complex revealed a strong binding energy for Site-1 bound INH (wild-type) with a value of

– 44.201 kJ/mol versus – 0.090 kJ/mol and – 3.805 kJ/mol in Site-2 and Site-3, respectively. A comparatively weaker binding affinity of INH in Site-1 of the mutant S315T (– 35.307 kJ/mol), with reduced hydrogen bond occupancy involving key residues for INH activation, i.e., Tyr229, His108 and Arg104, further establishing Site-1 as the main INH binding site.

MD simulations have also been used to assist prediction of potential allosteric sites of various MTB targets, such as the UDP-Galactopyranose Mutase (UGM) (Shi *et al.*, 2016) and the shikimate kinase (SK) enzymes (Mehra *et al.*, 2016a). UGM plays an important role in the biosynthesis of mycobacterial cell wall, as it catalyses the reversible conversion of UDP-galactopyranose (UDP-Galp) to UDP-galactofuranose (UDP-Galf), which is required for the building of arabinogalactan chain in the mycobacterial cell wall (Besra *et al.*, 1995; Nassau *et al.*, 1996). UGM gene knockout was found to inhibit the growth of MTB, indicating that UGM is essential for the growth of mycobacteria (Pan *et al.*, 2001). Moreover, both UGM and Galf are not found in human, suggesting that UGM is a viable anti-TB drug target. Recently, Pinto and colleagues (Shi *et al.*, 2016) have successfully discovered a second druggable binding site in UGM based on the analysis of 80 ns MD simulations (RMSF, number of contacts and interaction energies) of the complex of a novel non-substrate-like UGM inhibitor (MS-208, identified from saturation transfer difference (STD) NMR experiments and kinetic assays), docked at the allosteric site (A-site, predicted from blind docking) or the active site (S-site) of the UGM enzyme. As the docked complex (at the A-site) was found to have much smaller fluctuations, larger number of contacts, and larger interaction energy than the complex of inhibitor docked at the S-site, the authors concluded that MS-208 is likely to bind to the A-site. Further point mutation of the two interacting residues on the two flanking loops of A-site (Y253A and D322A) resulted in a significant loss in MS-208 inhibitory activities, thus further supported this conclusion.

In a separate study by Nargotra and co-workers (Mehra *et al.*, 2016a), MD simulation was used to elucidate the probability of inhibitor binding at the allosteric site of the SK enzyme. SK, which catalyses the phosphorylation of 3-OH-shikimic acid (in the presence of ATP as co-substrate), is one of the key enzymes in the shikimate pathway responsible for the biosynthesis of chorismate, an important precursor for the subsequent biosynthesis of common aromatic amino acid residues (Phe, Trp, Tyr) and secondary metabolites essential for the MTB survival (Herrmann and Weaver, 1999; Parish and Stoker, 2002). As the best SK inhibitor, 5489375 showed uncompetitive inhibition for SK and noncompetitive inhibition for ATP in *in vitro* experiments. Thus, it is suggested that the inhibitor may have potentially bound at another binding site. To investigate this, the inhibitor was docked to the common allosteric site of three different protein complexes, i.e., the Michaelis complex, complex A and the product complex, as predicted by SiteMap, and each complexes was subjected to a 10 ns MD simulation. Interestingly, the inhibitor was found to be displaced out of the initial binding cavity of the Michaelis complex and complex A to the surface of the protein after about 4 ns and 1 ns of simulations, respectively. Whilst the inhibitor remained in the initial binding cavity of the product complex throughout the simulation, with low ligand RMSD (~ 5 Å) and RMSF values (< 3.5 Å). This suggests that the inhibitor is likely to bind to the allosteric site only after the reaction products are formed. Additionally, further MD analysis demonstrated that hydrogen bond and hydrophobic contacts formed between residues Arg43, Ile45 and Phe57 of the enzyme and the inhibitor for at least 23% of the simulation time, suggesting that the binding of the inhibitor at the allosteric site of the SK enzyme is stabilised by these residues.

Preparation of Target Structure for Structure-Based Drug Design and Discovery Studies

A protein undergoes continual conformational changes under physiological conditions while maintaining its overall fold, i.e., 3D structure. A snapshot of a protein typically provided by the X-ray crystal structure would not be able to account for the continual conformational changes although the overall topology of the folded state remains conserved. During docking using a protein's crystal structure, a slightly different orientation of even a single side chain in the binding site of the protein can significantly influence docking results (Zhao and Caflisch, 2015). To address this issue, MD simulations have been used to generate multiple conformers of target proteins, where ensembles of snapshots from MD simulations of target proteins such as InhA enzymes from wild-type and 116T and 121V mutants, were selected for docking with inhibitors (Cohen *et al.*, 2011; Khan *et al.*, 2017). The idea is to model the explicit flexibility of the target proteins, which is postulated to be more accurate and realistic than rigid crystal structure for docking. Not surprisingly, the mode of ligand binding, the number of interacting residues and the ligand binding affinity in this models varied significantly from the docked ligand on the crystal structure, with more interacting residues observed for the ensembles. For example, Ser94 and Gly96 residues are also found to be crucial for strong drug-InhA interactions, in addition to the Lys165 and Ile194 residues, but these interactions (involving Ser94 and Gly96) were not observed in the crystal structure (PDB: 2NSD) (Khan *et al.*, 2017; He *et al.*, 2007).

Generation of ensembles of target proteins with MD simulations for docking, however, is not possible when no 3D structure is available. This is especially true for novel targets where the experimental 3D structures are usually yet to be determined. In TB, this include enzymes involved in the biochemical pathways for the biosynthesis of peptidoglycan, a key component of the MTB cell wall, such as the enzymes for the formation of UDP-GlcNAc from D-fructose-6-phosphate (GlmS, GlmM, GlmU), UDP-MurNAc from UDP-GlcNAc (MurA, MurB), UDP-MurNAc-tetrapeptide from UDP-MurNAc, and UDP-MurNAc-pentapeptide from UDP-MurNGlyc (MurC-MurF) (Alderwick *et al.*, 2015). Although these enzymes have long been characterised biochemically, not much work has been done on them until recently when the incidence of multi-drug resistant TB is getting more rampant. As such, homology models of the target enzymes such as MurA (Babajan *et al.*, 2011), MurC (Anuradha *et al.*, 2010), MurD (Arvind *et al.*, 2012), MurG (Fakhar *et al.*, 2016), Muri, MraY, DapE, DapA, Alr and Ddl (Fakhar *et al.*, 2016) were built based on the 3D structure of the template proteins in the PDB database with high sequence similarities to the target. Homology models, however, generally require further refinement, and this is usually performed by MD simulations on a 10–100 ns timescale on explicit environment (Fan and Mark, 2004). In these models, 5–20 ns of MD simulations were performed for refinement, and root mean square

deviation (RMSD) and radius of gyration (Rg) were calculated and compared to their corresponding templates or previous states to confirm the stability of these models. Subsequent use of these models for docking highlighted certain important residues for ligand and substrate binding and selectivity (e.g., Gly125, Lys126, Arg331 and Arg332 in MurC; His192, Arg382, Ser463 and Tyr470 in MurD) (Anuradha *et al.*, 2010; Arvind *et al.*, 2012). In addition, 10 potential inhibitors for MurG have been identified from the MD-refined model (Saxena *et al.*, 2017), although none of these observations have been validated by *in vitro* experiments.

Evaluation of the Binding Interactions, Stability and Free Binding Energies of Protein-Ligand Complexes

Apart from preparing protein structures for structure-based studies, MD simulations are also often used in tandem with docking studies to further optimise the docking poses of anti-TB drug candidate as well as identified hits from virtual screening or *in vitro* assays. One example is a study by de Jonge and colleagues, who have used energy minimisation (EM) – MD simulation cycles (60 cycles of alternate 1500 EM iterations and 1500 MD iterations of 0.2 fs) to optimise docking pose for each stereoisomer of the lead inhibitor of MTB ATPase, R207910 (de Jonge *et al.*, 2007). Computed interaction energies of the optimised docking poses showed that the RS stereoisomer bound more strongly to MTB ATPase than the SR, RR and SS stereoisomers, which are in agreement with the observed antimicrobial activity.

MD simulations are also commonly used to explore stability of ligand binding modes to various TB protein targets such as GyrB (Maharaj and Soliman, 2013; Islam and Pillay, 2017), GlgE (Sengupta *et al.*, 2015), MurG (Saxena *et al.*, 2017), GlmU (Mehra *et al.*, 2016b; Soni *et al.*, 2015), LipU (Kaur *et al.*, 2018), MtbA (Maganti *et al.*, 2015), MbtI (Maganti *et al.*, 2016), sHSP16.3 (Jee *et al.*, 2017), ASADH (Kumar *et al.*, 2015), AAC (Prabu *et al.*, 2015), PknG (Singh *et al.*, 2015), Polyphosphoglutamate synthetase (Hung *et al.*, 2014) and α subunit Trp synthase (Naz *et al.*, 2018). Generally, as low as 2 ns (Maharaj and Soliman, 2013) and as high as 60 ns (Jee *et al.*, 2017) of MD simulations have been performed for this purpose, although 10 ns simulations are also seen being carried out for most protein-ligand complexes. To evaluate the stability of the protein-ligand complexes, convergence of RMSD and root mean square fluctuation (RMSF) plots of the protein backbone (e.g., C α atoms) and ligands' non-hydrogen atom are monitored. In some studies, radius of gyration was also calculated to determine the changes in protein compactness and conformation upon ligand binding (Islam and Pillay, 2017; Kaur *et al.*, 2018). Less commonly, variability in potential energies and important ligand/residue distances were measured in order to evaluate the stability of the docked complex (Maharaj and Soliman, 2013; Kumar *et al.*, 2015).

A stable ligand-protein complex observed from MD simulations is important as it allows a better prediction of a ligand binding mode on the target protein which are dynamic in nature. More importantly, this information allows the identification of moieties critical for ligand-protein interaction. Generally, there are two common approaches that have been used to extract such information from stable trajectories of MD simulations involving TB protein target-inhibitor complexes. In the first approach, an average structure from the stable trajectories is calculated followed by visualisation of the protein-ligand interactions via a molecular visualisation software. This approach is illustrated in a study by Saxena and colleagues who compared the protein-ligand interactions of the average structure from stable MD simulations (20–25 ns) to the initial docked pose, and identified two of the seven residues (i.e., Leu290 and Met310) at the binding site of MurG that retained close interactions with the lead molecules as key residues important for MurG inhibition (Saxena *et al.*, 2017).

The second approach is by calculating the hydrogen bond occupancy involving the active site residues of the protein target and the ligand throughout the MD simulations. For examples, Kumar and colleagues discovered that the hydrogen bond occupancy for β -AFP ligand and inhibitor, NSC4862 on MTB-ASADH during whole MD simulation are 81.4% and 54.3% with Arg99 and 43.8%; and 30.3% with Arg249, respectively; suggesting that these highly conserved active site residues are critical for ASADH inhibition (Kumar *et al.*, 2015). Similarly, Mehra and colleagues revealed that Ser320, Thr321 and Asp424 of MTB Proteasome involved in hydrogen bond interactions with G7650246 in more than 25% of the simulation times, while residues Ala434, Arg439, Ala451 and Lys464 of GlmU have at least 20% hydrogen bond occupancy with inhibitor 5810599, suggesting that these residues are important for Proteasome and GlmU inhibitions, respectively (Mehra *et al.*, 2015, 2016b).

In addition to obtaining protein-ligand stability and interaction information, MD simulations are also being used for free energy calculation and free energy decomposition analysis of various anti-TB inhibitors. These are performed on stable MD trajectories of ligand-receptor complex, using Molecular Mechanics Poisson-Boltzmann Surface Area (MM-PBSA), Molecular Mechanics Generalised Born Surface Area (MM-GBSA) or Linear Interaction Energy (LIE) approximation approaches. The use of MM-PBSA and MM-GBSA to estimate ligand-binding affinities are not new (Genheden and Ryde, 2015). For instance, Kamsri and colleagues calculated the binding free energies of diphenyl ether inhibitors (compounds 17, 18, 19 and 29) in InhA using MM-PBSA method based on 125 snapshots from last 1 ns of MD trajectory; this calculated energies were found to agree well with the experimentally determined ΔG (Kamsri *et al.*, 2014b). Similarly, Kumar and colleagues used MM-GBSA to calculate binding free energy from 200 frames of the last 20 ns of 100 ns production run and successfully identified two InhA inhibitors (TB2, TB10) with better energies than the control ligand, (5-hexyl-2-(2-methylphenoxy)) phenol (IC_{50} = 5 nM). These compounds were found to give >90% inhibition of H37Rv strain of MTB in BACTEC assay (Kumar *et al.*, 2017). On the other hand, Timmers and colleagues demonstrated that calculation of free energy binding from MD trajectories with LIE method (which is a semi-empirical approach that combines advantages of free energy perturbation and thermodynamics integration) is as accurate as MM-PBSA or MM-GBSA methods, with only 0.7 kcal/mol difference in the calculated binding energy of an inhibitor (compound 9) on cytidine deaminase (MtCDA) compared to the experimental value (IC_{50} = 151.74 μ M, ΔG_{exp} = – 5.2 kcal/mol, ΔG_{LIE} = – 5.9 kcal/mol) (Timmers *et al.*, 2012).

MD simulations have also been used to build quantitative models to predict binding affinities of e.g., aryl acid-AMP bisubstrate inhibitors of MbtA (Labello *et al.*, 2008). The model was constructed based on correlation between the LIE-estimated free energy binding from MD trajectories and the experimental indicator of binding affinity. Interestingly, this model was able to predict the binding affinity of an unknown related compound, N-6-cyclopropyl-2-phenyl-SaI-AMS, with predicted binding affinity = 1.6 nM versus experimental K_i of 0.7 nM. In a separate study, snapshots from MD trajectories have been used to generate structure-based e-pharmacophore model of CmaA1 based on the interactions of the active site residues with the natural ligands SAM and SAHC (Choudhury *et al.*, 2015). These models were found to be able to correctly identify up to 17 out of 23 reference compounds screened compared to only one for model generated from crystal structure.

Example of Successful MD Study

MD simulation has been used to understand the mechanism of resistance of INH towards its target, i.e., InhA. Wahab and co-workers had investigated the mechanism of mycobacterial resistance against isoniazid, focusing on the active site of MTB InhA (Wahab *et al.*, 2009) (Fig. 6(A)). This work was initiated with molecular docking simulation of isonicotinic acyl-NADH (INADH; the active form of isoniazid) in both the wild-type and resistant-type (S94A) of InhA (Fig. 5(B)), followed by MD simulation to further investigate the dynamics behaviour of INADH in InhA. Interestingly, higher flexibility at the binding site was observed for resistant-type InhA compared to the wild-type, as reflected by the disruption of hydrogen bond network, and lesser hydrophobic, van der Waals and electrostatics interactions in the former. The smaller free energy binding, ΔG in resistant-type also correlates well with the interactions analysis where higher fluctuation caused by INADH was observed. In addition, the existence of water-mediated hydrogen bond network with INADH in the wild-type further supports the hypothesis that the conversion of prodrug isoniazid into its active form, INADH, is mediated by KatG on InhA. Further study (Choong and Wahab, 2011) on the dynamics of INADH in other reported single point mutations of InhA (I16T, I21T, I21V, I47T, V78 and I95P) (Fig. 6(B); Rouse *et al.*, 1995; Milano *et al.*, 1996; Miesel *et al.*, 1998), showed that the reduced side chain volume in all InhA mutants has provided more rooms for structural fluctuations in both INADH and mutated InhA residues, thus decreasing the INADH binding affinity, in agreement with the experimental findings on low-level isoniazid resistance for InhA mutants (Fig. 6). Although the resistance to isoniazid has been known for 70 years, there are still rooms to improve our understanding to this phenomena by using MD simulations in combination of other physical methods such as ultrafast two-dimensional infrared spectroscopy as presented recently by Hunt and co-workers (Shaw *et al.*, 2017) and the knowledge from the simulations could be exploited further in searching for potential inhibitors as alternative to isoniazid.

Using MD to understand the underlying INH mechanism of action, has allowed Venugopala and co-workers to venture into discovering InhA inhibitors by combining this technique with docking calculation (Khedr *et al.*, 2017). *In silico* docking trisubstituted indolizine analogues along with resazurin microplate assay screening were conducted to screen on the

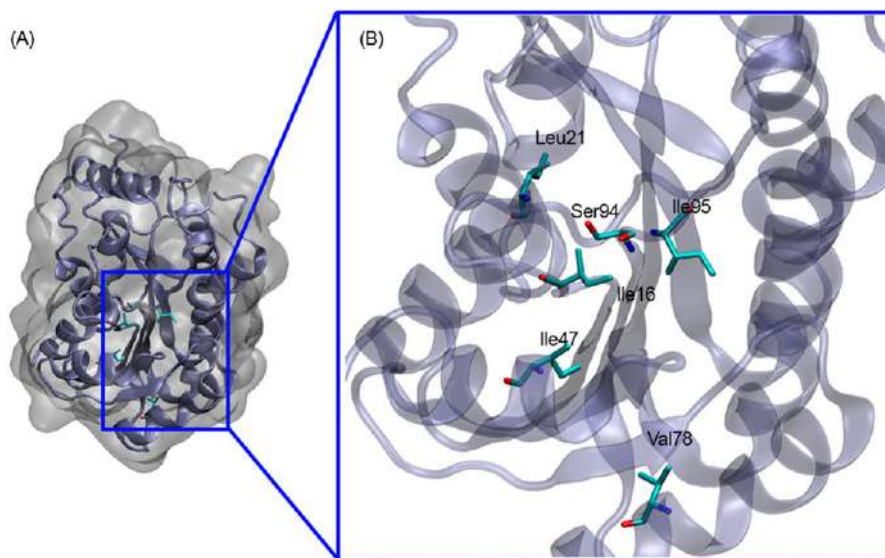


Fig. 6 (A) Crystal structure of MTB enoyl-acyl carrier protein reductase 9PDB ID: 1ZID. Reproduced from Rozwarski, D.A., Grant, G.A., Barton, D. H., Jacobs, W.R., Jr., Sacchettini, J.C., 1998. Modification of the NADH of the isoniazid target (InhA) from *Mycobacterium tuberculosis*. *Science* 279, 98–102. (B) Point mutations S94A, I16T, I21T, I21V, I47T, V78 and I95P constructed by Choong and Habibah. Reproduced from Choong, Y. S., Wahab, H., 2011. Effects of enoyl-acyl protein carrier reductase mutations on physiochemical interactions with isoniazid: Molecular dynamics simulation. Wahab, H.A., Choong, Y.S., Ibrahim, P., Sadikun, A., Scior, T., 2009. Elucidating isoniazid resistance using molecular modeling. *J. Chem. Inf. Model.* 49, 97–107.

indolizine analogues. 50 ns MD simulation was conducted on the top selected hit from docking and assay results. Their assay results had shown that by substitution of ethyl group at second position of indolizine nucleus, it exhibited activity against susceptible and multidrug-resistant strains of MTB at concentration of 5.5 and 11.3 $\mu\text{g/mL}$, respectively. Correlation was found for compound with ethyl substitution at second position of indolizine nucleus on the docking data with highest free binding energy of $\Delta G - 24.11$ (kcal/mol). With MD simulation, MM-PB/GBSA results had also indicated well correlation with docking as well as the experimental data. Besides, Pongpo and co-workers had also adopted MD simulation (Kamsri *et al.*, 2014a) that further extended the work with 3D-QSAR to understand the structural basis of diphenyl ether derivatives on InhA (Kamsri *et al.*, 2014b). The volume of the binding cavity and the surface area of the compounds were correlated with the inhibitory activity. Their results indicated that smaller binding site cavity with higher surface area of the compound served to increase the inhibitory activity. Larger binding site cavity may imply less fitting of the compound to the binding site and thus leading to the loss of activity. Besides, energy decomposition of important residues such as Phe149, Tyr158, Met161, Met199, Val203 were analysed using MM-PBSA method. In their work later Kamsri *et al.* (2014b), 3D-QSAR approach was used to identify the key features of diphenyl ether derivatives for InhA inhibition were and again MD was used to study the binding energy using MM-PBSA calculation.

As a prodrug, isoniazid needs to be activated by KatG enzyme of which the mutations, has also been correlated with the INH resistance. To date, there is no crystal structure of isoniazid bound to KatG MTB. Using molecular docking of INH on KatG followed by 100ns MD simulation has allowed the identification of INH binding site on MtKatG whereby only one site (Site-1) resulted in conformational change when INH bound to it suggesting it as the main INH binding site (Srivastava *et al.*, 2017). This information is crucial such that future INH targeting inhibitors should avoid being metabolised by KatG enzyme. Theoretically, MD can be used to assess whether or not a new potential inhibitor of InhA can be metabolised by KatG from the prediction of the binding interaction of the molecule with the enzyme. Avoiding being metabolised by KatG enzyme will prevent drug resistance due to mutation or deletion of this enzyme in MTB.

The ability of MTB to go into a dormant state has caused a treatment nightmare as such targeting an enzyme important to the survival of MTB in the dormant state is a good alternative in finding potential TB drugs. Isocitrate lyase (ICL), is one of the key (Fig. 4(A)) enzymes involved in energy generation in MTB via glyoxylate shunt. It is one of the important targets in dormant phase MTB. Research on ICL especially on structure based drug design is still popular and ongoing. However, due to its small polar active site, only moderate ICL inhibitors have been reported (Mdluli *et al.*, 2014), thus opportunities to understand the structure, dynamics and stability of this enzyme for drug design cannot be understated.

To date, both *apo*- and *holo*- forms of ICL have been reported (PDB ID: 1F61; 1F8I; 1F8M; 5DQL). To understand the functional behaviour of ICL in its different subunit forms, MD simulations of ICL have been carried out (Lee *et al.*, 2017). Briefly, ICL in its dimer form was studied in three different conditions, i.e., one apo and two complex forms in order to investigate the open and close conformation of both bound and unbound states with its substrate. Analysis from the MD trajectories suggested that the open-close behaviour of the entrance of ICL active site is highly dependent on the type of ligands as free energy calculations showed a stronger binding affinity for isocitrate (gate-opening) than glyoxylate or succinate (gate-closing) (Fig. 7(A-D)). A total of four residues were suggested to be the possible points for cleavage of isocitrate at the C2-C3 bond by nucleophilic attack.

Investigation on ICL is also the centre of attention for Shukla and co-workers who were interested to understand the effects of mutation of this enzyme (Shukla *et al.*, 2015) using MD approach. The effect of flexibility of ICL on the functionality of this enzyme was first investigated by a single point mutation of His180 (Fig. 4(B)). Observations on the dynamic changes of ICL revealed a disrupted interaction between His180 and Tyr89, which affects the tetrameric assembly important for the enzymatic activities. Subsequent site-directed mutagenesis work on a different binding site residue, Phe345 (Shukla *et al.*, 2017a; Fig. 4(B)), discovered that the flexibility of the enzyme in the binding site was increased for the mutant type, resulting in the decreased stability of the whole enzyme.

Point mutation on H46A (Shukla *et al.*, 2018) using MD were also conducted. Interestingly, although the point mutation site (H46A) is far away from the binding site of ICL, they found that the conformation at the binding site and the volume of binding site cavity were altered, and the loss of enzyme activity was observed. They predicted that this could be due to the loss in structural plasticity and collective motion. This postulation is reflected by the time-dependent secondary structure analysis. In their extensive MD simulation on both wild-type and mutant-type (H46A), significant secondary structure changes were observed where the catalytic cleft of the wild-type was found to adopt a higher numbers of turn and bend with the formation of 3-helices. The number of these loops, turns and 3-helices structures were significantly lower in the mutant-type, indicating a more rigid structure in the mutant. In addition, with H46A mutation, the residue interaction network at the active site was also found to lead to the changes in the active site environment. These observations suggest that apart from the active site residues, His46, which is distance away from the active site cleft, can also affects the ICL functionality, and thus is also a promising site to be exploited (other than the active site) in discovery of novel ICL inhibitors.

Again, the information generated by MD simulations have been used in *in silico* identification of potential inhibitor of ICL from the ZINC database (Shukla *et al.*, 2017c). To compare the dynamics of the protein upon ligand binding, MD simulations were performed, and principle component analysis (PCA) was calculated for the stable trajectories (last 20 ns) to predict large concerted motions during ligand binding to ICL. Values for the first few eigenvectors of apo-ICL were found to be higher than the ICL-ligand complexes, suggesting that the apo-ICL has more correlated motions, and thus less stable than the ICL-ligand complexes. In other words, ligand binding stabilised the complex. Additionally, 2D projection of the trajectories for PC1 and PC2 for ICL-

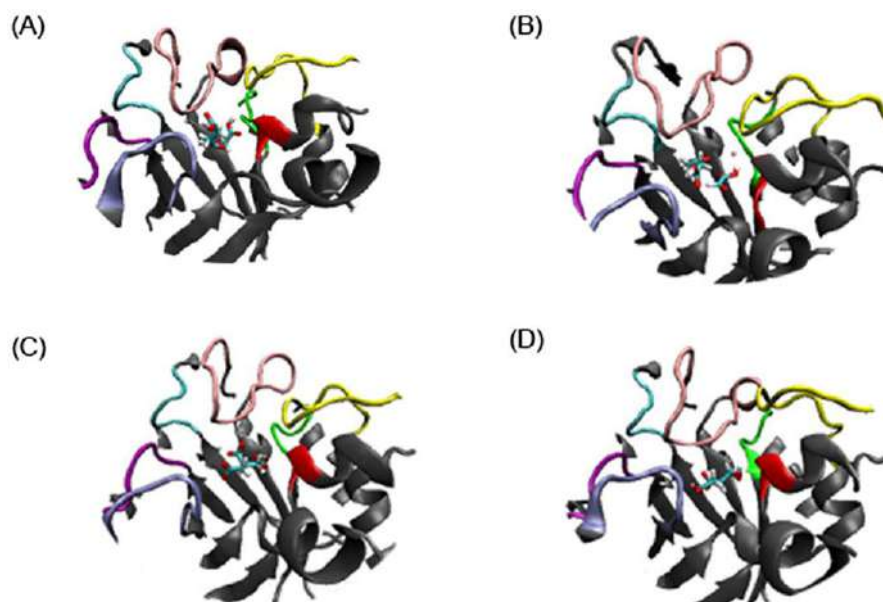


Fig. 7 Enzyme active site views of holo ICL dimer at 30 ns MD simulation. (A) Complex_ICL (with the presence of glycoxylate and succinate) active site 1, (B) Complex_ICL (with the presence of glycoxylate and succinate) active site 2, (C) Complex_ICL (with the presence of isocitrate only) active site 1 and (D) Complex_ICL (with the presence of isocitrate only) active site 2. Loops were highlighted in colours with respective active site 1/active site 2 residue number. Red: residues 91–93/518–520; Yellow: residues 104–114/531–541; Green: residues 154–159/581–585; Pink: residues 185–196/612–623; Cyan: residues 230–234/657–661; Purple: residues 285–289/712–716; Ice-blue: residues 314–319/741–746.

ZINC2111081 complex revealed that it occupied less space in the phase space than other compounds, with well-defined cluster, suggesting that ZINC2111081 formed more stable complex with ICL.

The above scenarios although limited to two enzymes can easily be extended to other MTB drug targets of which some of them have been covered in the previous section. Clearly, the application of MD in drug design is of immense importance, however, it should be noted that the potency of inhibitors is only the first step in drug discovery. Other aspects of drug development such as toxicity, pharmacokinetics and pharmacodynamics are also important determinants for the inhibitors to succeed in clinical trials. Limitations of MD must also be taken into consideration. Improvement of force fields, length of the simulations, enhanced sampling, inclusion of entropy effects, allosteric modulation and solvation effects should also be given special emphasis in the MD study relating to drug discovery.

Concluding Remarks

From the evaluation of the fundamental understanding of conformation alteration, functionally significant interaction as well as mechanism of action of lead compound on potential drug targets in MTB, MD simulation no doubt is one of the most time and cost-effective tools to be used. Moreover, it is also able to enhance the drug-receptor binding mode, as well as predicting drug binding affinity at better accuracy, which is important in understanding the structure-activity relationship in drug discovery. Additionally, combining molecular insights from MD simulation with other *in silico* approaches such as high-throughput screening of potential hit, in-depth atomic interaction studies using QM/MM as well as further validation by conventional experiments would certainly help to further enhance the efficiency in discovery of new anti-TB drugs for the benefits of the mankind.

Acknowledgement

We thank Dr Thaigarajan Parumasivam for his generosity in providing the SEM image of the H37Rv strain (ATCC 25618) of *Mycobacterium tuberculosis*. Special thanks to Dr Lee Yie Vern for her permission to reuse and readapt the ICL MD trajectories. Choong YS would like to acknowledge Universiti Sains Malaysia Bridging Grant (304/CIPPM/6316018).

See also: Biological Database Searching. Biomolecular Structures: Prediction, Identification and Analyses. Cloud-Based Molecular Modeling Systems. Comparative Epigenomics. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. Natural Language Processing Approaches in Bioinformatics. Structure-Based Drug Design Workflow

References

- Aggarwal, M., Singh, A., Grover, S., *et al.*, 2018. Role of *pncA* gene mutations W68R and W68G in pyrazinamide resistance. *J. Cell .Biochem.* 119, 2567–2578.
- Alderwick, L.J., Harrison, J., Lloyd, G.S., Birch, H.L., 2015. The mycobacterial cell wall–peptidoglycan and arabinogalactan. *Cold Spring Harb.Perspect. Med.* 5, a021113.
- Anuradha, C.M., Mulakayala, C., Babajan, B., *et al.*, 2010. Probing ligand binding modes of *Mycobacterium tuberculosis* MurC ligase by molecular modeling, dynamics simulation and docking. *J. Mol. Model.* 16 (1), 77–85.
- Arvind, A., Kumar, V., Saravanan, P., Mohan, C.G., 2012. Homology modeling, molecular dynamics and inhibitor binding study on MurD ligase of *Mycobacterium tuberculosis*. *Interdiscip. Sci.* 4, 223–238.
- Babajan, B., Chaitanya, M., Rajsekhar, C., *et al.*, 2011. Comprehensive structural and functional characterization of *Mycobacterium tuberculosis* UDP-NAG enolpyruvyl transferase (Mtb-MurA) and prediction of its accurate binding affinities with inhibitors. *Interdiscip. Sci.* 3, 204–216.
- Banerjee, A., Dubnau, E., Quemard, A., *et al.*, 1994. *inhA*, a gene encoding a target for isoniazid and ethionamide in *Mycobacterium tuberculosis*. *Science* 263, 227–230.
- Besra, G.S., Khoo, K.H., Mcneil, M.R., *et al.*, 1995. A new interpretation of the structure of the mycolyl-arabinogalactan complex of *Mycobacterium tuberculosis* as revealed through characterization of oligoglycosylalditol fragments by fast-atom bombardment mass spectrometry and ¹H nuclear magnetic resonance spectroscopy. *Biochemistry* 34, 4257–4266.
- Borhani, D.W., Shaw, D.E., 2012. The future of molecular dynamics simulations in drug discovery. *J. Comput. Aided Mol. Des.* 26, 15–26.
- CDC, 2017. Economic toll of drug resistant TB [Online]. Available at: <https://www.cdc.gov/tb/topic/drtb/default.htm> (accessed 31.03.18).
- Choong, Y.S., Wahab, H., 2011. Effects of Enoyl-Acyl protein carrier reductase mutations on physicochemical interactions with isoniazid: Molecular dynamics simulation.
- Choudhury, C., Priyakumar, U.D., Sastry, G.N., 2015. Dynamics based pharmacophore models for screening potential inhibitors of mycobacterial cyclopropane synthase. *J. Chem. Inf. Model* 55, 848–860.
- Cohen, E.M., Machado, K.S., Cohen, M., D.E. Souza, O.N., 2011. Effect of the explicit flexibility of the *InhA* enzyme from *Mycobacterium tuberculosis* in molecular docking simulations. *BMC Genom.* 12 (Suppl. 4), S7.
- Durrant, J.D., McCammon, J.A., 2011. Molecular dynamics simulations and drug discovery. *BMC Biol.* 9, 71.
- Fakhar, Z., Naiker, S., Alves, C.N., *et al.*, 2016. A comparative modeling and molecular docking study on *Mycobacterium tuberculosis* targets involved in peptidoglycan biosynthesis. *J. Biomol. Struct. Dyn.* 34, 2399–2417.
- Fang, Z., Doig, C., Rayner, A., *et al.*, 1999. Molecular evidence for heterogeneity of the multiple-drug-resistant *Mycobacterium tuberculosis* population in Scotland (1990 to 1997). *J. Clin. Microbiol.* 37, 998–1003.
- Fan, H., Mark, A.E., 2004. Refinement of homology-based protein structures by molecular dynamics simulation techniques. *Protein Sci.* 13, 211–220.
- Ferraris, D., Miggiano, R., Rossi, F., Rizzi, M., 2018. *Mycobacterium tuberculosis* molecular determinants of infection, survival strategies, and vulnerable targets. *Pathogens* 7, 17.
- Ferreira, L.A., Madeira, P.P., Breydo, L., *et al.*, 2016. Role of solvent properties of aqueous media in macromolecular crowding effects. *J. Biomol. Struct. Dyn.* 34, 92–103.
- Genheden, S., Ryde, U., 2015. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert Opin. Drug Discov.* 10, 449–461.
- Hards, K., Robson, J.R., Berney, M., *et al.*, 2015. Bactericidal mode of action of bedaquiline. *J. Antimicrob. Chemother.* 70, 2028–2037.
- Herrmann, K.M., Weaver, L.M., 1999. The shikimate pathway. *Annu. Rev. Plant Physiol. Plant Mol. Biol.* 50, 473–503.
- Heym, B., Alzari, P.M., Honore, N., Cole, S.T., 1995. Missense mutations in the catalase-peroxidase gene, *katG*, are associated with isoniazid resistance in *Mycobacterium tuberculosis*. *Mol. Microbiol.* 15, 235–245.
- He, X., Alian, A., Ortiz De Montellano, P.R., 2007. Inhibition of the *Mycobacterium tuberculosis* enoyl acyl carrier protein reductase *InhA* by arylamides. *Bioorg. Med. Chem.* 15, 6649–6658.
- Hoagland, D., Liu, J., Lee, R.B., Lee, R.E., 2016. New agents for the treatment of drug-resistant *Mycobacterium tuberculosis*. *Advanced Drug Deliv. Rev.* 102, 55–72.
- Hung, T.C., Chen, K.B., Lee, W.Y., Chen, C.Y., 2014. The inhibition of folypolyglutamate synthetase (*folC*) in the prevention of drug resistance in *Mycobacterium tuberculosis* by traditional Chinese medicine. *Biomed. Res. Int.* 2014, 635152.
- Islam, M.A., Pillay, T.S., 2017. Identification of promising DNA GyrB inhibitors for Tuberculosis using pharmacophore-based virtual screening, molecular docking and molecular dynamics studies. *Chem. Biol. Drug. Des.* 90, 282–296.
- Jee, B., Kumar, S., Yadav, R., *et al.*, 2017. Ursolic acid and carvacrol may be potential inhibitors of dormancy protein small heat shock protein16.3 of *Mycobacterium tuberculosis*. *J. Biomol. Struct. Dyn.* 1–10.
- Jian, J.-W., Elumalai, P., Pitti, T., *et al.*, 2016. Predicting ligand binding sites on protein surfaces by 3-dimensional probability density distributions of interacting atoms. *PLOS ONE* 11, e0160315.
- Jiao, W., Blackmore, N.J., Nazmi, A.R., Parker, E.J., 2017. Quaternary structure is an essential component that contributes to the sophisticated allosteric regulation mechanism in a key enzyme from *Mycobacterium tuberculosis*. *PLOS ONE* 12, e0180052.
- de Jonge, M.R., Koymans, L.H., Guillemon, J.E., Koul, A., Andries, K., 2007. A computational model of the inhibition of *Mycobacterium tuberculosis* ATPase by a new drug candidate R207910. *Proteins* 67, 971–980.
- Jorgensen, W.L., 1991. Rusting of the lock and key model for protein-ligand binding. *Science* 254, 954.
- Kamsri, P., Koohatammakun, N., Srisupan, A., *et al.*, 2014a. Rational design of *InhA* inhibitors in the class of diphenyl ether derivatives as potential anti-tubercular agents using molecular dynamics simulations. *SAR QSAR Environ. Res.* 25, 473–488.
- Kamsri, P., Punkvang, A., Saparpakorn, P., *et al.*, 2014b. Elucidating the structural basis of diphenyl ether derivatives as highly potent enoyl-ACP reductase inhibitors through molecular dynamics simulations and 3D-QSAR study. *J. Mol. Model.* 20, 2319.
- Kamachi, S., Hirabayashi, K., Tamoi, M., *et al.*, 2015a. The crystal structure of isoniazid-bound *KatG* catalase-peroxidase from *Synechococcus elongatus* PCC7942. *FEBS J.* 282, 54–64.
- Kamachi, S., Hirabayashi, K., Tamoi, M., *et al.*, 2015b. Crystal structure of the catalase-peroxidase *KatG* W78F mutant from *Synechococcus elongatus* PCC7942 in complex with the antitubercular pro-drug isoniazid. *FEBS Lett* 589, 131–137.
- Kaur, G., Pandey, B., Kumar, A., *et al.*, 2018. Drug targeted virtual screening and molecular dynamics of LipU protein of *Mycobacterium tuberculosis* and *Mycobacterium leprae*. *J. Biomol. Struct. Dyn.* 1–38.
- Khan, A.M., Shawon, J., Halim, M.A., 2017. Multiple receptor conformers based molecular docking study of fluorine enhanced ethionamide with mycobacterium enoyl ACP reductase (*InhA*). *J. Mol. Graph. Model.* 77, 386–398.
- Khedr, M.A., Pillay, M., Chandrashekarappa, S., *et al.*, 2017. Molecular modeling studies and anti-TB activity of trisubstituted indolizine analogues; molecular docking and dynamic inputs. *J. Biomol. Struct. Dyn.* 1–16.
- Kiepiela, P., Bishop, K.S., Smith, A.N., Roux, L., York, D.F., 2000. Genomic mutations in the *katG*, *inhA* and *aphC* genes are useful for the prediction of isoniazid resistance in *Mycobacterium tuberculosis* isolates from KwaZulu Natal, South Africa. *Tuber. Lung Dis.* 80, 47–56.
- Kumar, R., Garg, P., Bharatam, P.V., 2015. Shape-based virtual screening, docking, and molecular dynamics simulations to identify Mtb-ASADH inhibitors. *J. Biomol. Struct. Dyn.* 33, 1082–1093.
- Kumar, V., Jhamb, S.S., Sobhia, M.E., 2017. Cell wall permeability assisted virtual screening to identify potential direct *InhA* inhibitors of *Mycobacterium tuberculosis* and their biological evaluation. *J. Biomol. Struct. Dyn.* 1–17.

- Labbello, N.P., Bennett, E.M., Ferguson, D.M., Aldrich, C.C., 2008. Quantitative three dimensional structure linear interaction energy model of 5'-O-[N-(salicyl)sulfamoyl]adenosine and the aryl acid adenylating enzyme MbtA. *J. Med. Chem.* 51, 7154–7160.
- Lahti, J.L., Tang, G.W., Capriotti, E., Liu, T., Altman, R.B., 2012. Bioinformatics and variability in drug response: A protein structural perspective. *J. R. Soc. Interface* 9, 1409–1437.
- Lee, Y.V., Choi, S.B., Wahab, H.A., Choong, Y.S., 2017. Active site flexibility of Mycobacterium tuberculosis isocitrate lyase in dimer form. *J. Chem. Inf. Model.* 57, 2351–2357.
- Lee, A.S., Lim, I.H., Tang, L.L., Telenti, A., Wong, S.Y., 1999. Contribution of kasA analysis to detection of isoniazid-resistant Mycobacterium tuberculosis in Singapore. *Antimicrob. Agents Chemother.* 43, 2087–2089.
- Lewis, J.M., Sloan, D.J., 2015. The role of delamanid in the treatment of drug-resistant tuberculosis. *Ther. Clin. Risk Manag.* 11, 779–791.
- Maganti, L., Consortium, O., Ghoshal, N., 2015. 3D-QSAR studies and shape based virtual screening for identification of novel hits to inhibit MbtA in Mycobacterium tuberculosis. *J. Biomol. Struct. Dyn.* 33, 344–364.
- Maganti, L., Grandhi, P., Ghoshal, N., 2016. Integration of ligand and structure based approaches for identification of novel MbtI inhibitors in Mycobacterium tuberculosis and molecular dynamics simulation studies. *J. Mol. Graph. Model.* 70, 14–22.
- Maharaj, Y., Soliman, M.E., 2013. Identification of novel gyrase B inhibitors as potential anti-TB drugs: Homology modelling, hybrid virtual screening and molecular dynamics simulations. *Chem. Biol. Drug. Des.* 82, 205–215.
- Mdluli, K., Kaneko, T., Upton, A., 2014. Tuberculosis drug discovery and emerging targets. *Ann. N. Y. Acad. Sci.* 1323, 56–75.
- Mehra, R., Chib, R., Munaga, G., *et al.*, 2015. Discovery of new Mycobacterium tuberculosis proteasome inhibitors using a knowledge-based computational screening approach. *Mol. Divers* 19, 1003–1019.
- Mehra, R., Rajput, V.S., Gupta, M., *et al.*, 2016a. Benzothiazole derivative as a novel Mycobacterium tuberculosis shikimate kinase inhibitor: Identification and elucidation of its allosteric mode of inhibition. *J. Chem. Inf. Model.* 56, 930–940.
- Mehra, R., Rani, C., Mahajan, P., *et al.*, 2016b. Computationally guided identification of novel Mycobacterium tuberculosis GlmU inhibitory leads, their optimization, and in vitro validation. *ACS Comb. Sci.* 18, 100–116.
- Miesel, L., Weisbrod, T.R., Marcinkeviciene, J.A., Bittman, R., Jacobs Jr., W.R., 1998. NADH dehydrogenase defects confer isoniazid resistance and conditional lethality in Mycobacterium smegmatis. *J. Bacteriol.* 180, 2459–2467.
- Milano, A., de Rossi, E., Gusberti, L., *et al.*, 1996. The katE gene, which encodes the catalase HPII of Mycobacterium avium. *Mol. Microbiol.* 19, 113–123.
- Morris, S., Bai, G.H., Suffys, P., *et al.*, 1995. Molecular mechanisms of multiple drug resistance in clinical isolates of Mycobacterium tuberculosis. *J. Infect. Dis.* 171, 954–960.
- Nassau, P.M., Martin, S.L., Brown, R.E., *et al.*, 1996. Galactofuranose biosynthesis in Escherichia coli K-12: Identification and cloning of UDP-galactopyranose mutase. *J. Bacteriol.* 178, 1047–1052.
- Naz, S., Farooq, U., Ali, S., *et al.*, 2018. Identification of new benzamide inhibitor against alpha-subunit of tryptophan synthase from Mycobacterium tuberculosis through structure-based virtual screening, anti-tuberculosis activity and molecular dynamics simulations. *J. Biomol. Struct. Dyn.* 1–11.
- Nusrath Unissa, A., Hassan, S., Indira Kumari, V., Revathy, R., Hanna, L.E., 2016. Insights into RpoB clinical mutants in mediating rifampicin resistance in Mycobacterium tuberculosis. *J. Mol. Graph. Model.* 67, 20–32.
- Pandey, B., Grover, S., Goyal, S., *et al.*, 2018. Alanine mutation of the catalytic sites of Pantothenate Synthetase causes distinct conformational changes in the ATP binding region. *Sci. Rep.* 8, 903.
- Pan, F., Jackson, M., Ma, Y., Mcneil, M., 2001. Cell wall core galactofuran synthesis is essential for growth of mycobacteria. *J. Bacteriol.* 183, 3991–3998.
- Parish, T., Stoker, N.G., 2002. The common aromatic amino acid biosynthesis pathway is essential in Mycobacterium tuberculosis. *Microbiology* 148, 3069–3077.
- Piatek, A.S., Telenti, A., Murray, M.R., *et al.*, 2000. Genotypic analysis of Mycobacterium tuberculosis in two distinct populations using molecular beacons: Implications for rapid susceptibility testing. *Antimicrob. Agents Chemother.* 44, 103–110.
- Pimentel, A.L., de Lima Scodro, R.B., Caleffi-Ferracioli, K.R., *et al.*, 2017. Mutations in catalase-peroxidase KatG from isoniazid resistant Mycobacterium tuberculosis clinical isolates: Insights from molecular dynamics simulations. *J. Mol. Model.* 23, 121.
- Prabu, A., Hassan, S., Prabuseenivasan, Shainaba, A.S., Hanna, L.E., Kumar, V., 2015. Andrographolide: A potent antituberculosis compound that targets aminoglycoside 2'-N-acetyltransferase in Mycobacterium tuberculosis. *J. Mol. Graph. Model.* 61, 133–140.
- Rotta, M., Timmers, L., Sequeiros-Borja, C., *et al.*, 2017. Observed crowding effects on Mycobacterium tuberculosis 2-trans-enoyl-ACP (CoA) reductase enzyme activity are not due to excluded volume only. *Sci. Rep.* 7, 6826.
- Rouse, D.A., Li, Z., Bai, G.H., Morris, S.L., 1995. Characterization of the katG and inhA genes of isoniazid-resistant clinical isolates of Mycobacterium tuberculosis. *Antimicrob. Agents Chemother.* 39, 2472–2477.
- Saxena, S., Abdullah, M., Sriram, D., Guruprasad, L., 2017. Discovery of novel inhibitors of Mycobacterium tuberculosis MurG: Homology modelling, structure based pharmacophore, molecular docking, and molecular dynamics simulations. *J. Biomol. Struct. Dyn.* 1–15.
- Sengupta, S., Roy, D., Bandyopadhyay, S., 2015. Structural insight into Mycobacterium tuberculosis maltosyl transferase inhibitors: Pharmacophore-based virtual screening, docking, and molecular dynamics simulations. *J. Biomol. Struct. Dyn.* 33, 2655–2666.
- Shaw, D.J., Hill, R.E., Simpson, N., *et al.*, 2017. Examining the role of protein structural dynamics in drug resistance in Mycobacterium tuberculosis. *Chem. Sci.* 8, 8384–8399.
- Shi, Y., Colombo, C., Kuttijayveetil, J.R., *et al.*, 2016. A Second, druggable binding site in UDP-Galactopyranose mutase from Mycobacterium tuberculosis? *Chembiochem* 17, 2264–2273.
- Shukla, H., Kumar, V., Singh, A.K., *et al.*, 2015. Insight into the structural flexibility and function of Mycobacterium tuberculosis isocitrate lyase. *Biochimie* 110, 73–80.
- Shukla, H., Shukla, R., Sonkar, A., Pandey, T., Tripathi, T., 2017a. Distant Phe345 mutation compromises the stability and activity of Mycobacterium tuberculosis isocitrate lyase by modulating its structural flexibility. *Sci. Rep.* 7, 1058.
- Shukla, R., Shukla, H., Sonkar, A., Pandey, T., Tripathi, T., 2017c. Structure-based screening and molecular dynamics simulations offer novel natural compounds as potential inhibitors of Mycobacterium tuberculosis isocitrate lyase. *J. Biomol. Struct. Dyn.* 1–13.
- Shukla, H., Shukla, R., Sonkar, A., Tripathi, T., 2017b. Alterations in conformational topology and interaction dynamics caused by L418A mutation leads to activity loss of Mycobacterium tuberculosis isocitrate lyase. *Biochem. Biophys. Res. Commun.* 490, 276–282.
- Shukla, R., Shukla, H., Tripathi, T., 2018. Activity loss by H46A mutation in Mycobacterium tuberculosis isocitrate lyase is due to decrease in structural plasticity and collective motions of the active site. *Tuberculosis* 108, 143–150.
- Shi, Y., Colombo, C., Kuttijayveetil, J.R., *et al.*, 2016. A Second, Druggable Binding Site in UDP-Galactopyranose Mutase from Mycobacterium tuberculosis? *Chembiochem* 17, 2264–2273.
- Singh, A., Grover, S., Pandey, B., Kumari, A., Grover, A., 2018. Wild-type catalase peroxidase vs G279D mutant type: Molecular basis of Isoniazid drug resistance in Mycobacterium tuberculosis. *Gene* 641, 226–234.
- Singh, A., Grover, S., Sinha, S., *et al.*, 2017. Mechanistic principles behind molecular mechanism of rifampicin resistance in mutant rna polymerase beta subunit of mycobacterium tuberculosis. *J. Cell Biochem.* 118, 4594–4606.
- Singh, N., Tiwari, S., Srivastava, K.K., Siddiqi, M.I., 2015. Identification of novel inhibitors of mycobacterium tuberculosis pngk using pharmacophore based virtual screening, docking, molecular dynamics simulation, and their biological evaluation. *J. Chem. Inf. Model.* 55, 1120–1129.
- Slayden, R.A., Barry 3rd, C.E., 2000. The genetics and biochemistry of isoniazid resistance in Mycobacterium tuberculosis. *Microbes. Infect.* 2, 659–669.

- Soni, V., Suryadevara, P., Sriram, D., *et al.*, 2015. Structure-based design of diverse inhibitors of Mycobacterium tuberculosis N-acetylglucosamine-1-phosphate uridylyltransferase: Combined molecular docking, dynamic simulation, and biological activity. *J. Mol. Model.* 21, 174.
- Srivastava, G., Tripathi, S., Kumar, A., Sharma, A., 2017. Molecular investigation of active binding site of isoniazid (INH) and insight into resistance mechanism of S315T-MtKatG in Mycobacterium tuberculosis. *Tuberculosis* 105, 18–27.
- Telenti, A., 1998. Genetics of drug resistant tuberculosis. *Thorax* 53, 793–797.
- Telenti, A., Honore, N., Bernasconi, C., *et al.*, 1997. Genotypic assessment of isoniazid and rifampin resistance in Mycobacterium tuberculosis: A blind study at reference laboratory level. *J. Clin. Microbiol.* 35, 719–723.
- Timmers, L.F., Ducati, R.G., Sanchez-Quitian, Z.A., *et al.*, 2012. Combining molecular dynamics and docking simulations of the cytidine deaminase from Mycobacterium tuberculosis H37Rv. *J. Mol. Model.* 18, 467–479.
- Torres, M.J., Criado, A., Palomares, J.C., Aznar, J., 2000. Use of real-time PCR and fluorimetry for rapid detection of rifampin and isoniazid resistance-associated mutations in Mycobacterium tuberculosis. *J. Clin. Microbiol.* 38, 3194–3199.
- Unissa, A.N., Doss, C.G., Kumar, T., *et al.*, 2017. Analysis of interactions of clinical mutants of catalase-peroxidase (KatG) responsible for isoniazid resistance in Mycobacterium tuberculosis with derivatives of isoniazid. *J. Glob. Antimicrob. Resist.* 11, 57–67.
- Vilcheze, C., Morbidoni, H.R., Weisbrod, T.R., *et al.*, 2000. Inactivation of the inhA-encoded fatty acid synthase II (FASII) enoyl-acyl carrier protein reductase induces accumulation of the FASII end products and cell lysis of Mycobacterium smegmatis. *J. Bacteriol.* 182, 4059–4067.
- Vidossich, P., Loewen, P.C., Carpena, X., *et al.*, 2014. Binding of the antitubercular pro-drug isoniazid in the heme access channel of catalase-peroxidase (KatG). A combined structural and metadynamics investigation. *J Phys Chem B* 118, 2924–2931.
- De Vivo, M., Masetti, M., Bottegoni, G., Cavalli, A., 2016. Role of molecular dynamics and related methods in drug discovery. *J. Med. Chem.* 59, 4035–4061.
- Wahab, H.A., Choong, Y.S., Ibrahim, P., Sadikun, A., Scior, T., 2009. Elucidating isoniazid resistance using molecular modeling. *J. Chem. Inf. Model.* 49, 97–107.
- Wang, J.Y., Burger, R.M., Drlica, K., 1998. Role of superoxide in catalase-peroxidase-mediated isoniazid action against mycobacteria. *Antimicrob. Agents Chemother.* 42, 709–711.
- Wellington, S., Hung, D.T., 2018. The expanding diversity of Mycobacterium tuberculosis drug targets. *ACS Infect. Dis.*
- WHO, 2017. Global tuberculosis report 2017.
- Yao, J., Wang, X., Luo, H., Gu, P., 2017. Understanding the catalytic mechanism and the nature of the transition state of an attractive drug-target enzyme (Shikimate Kinase) by quantum mechanical/molecular mechanical (QM/MM) studies. *Chemistry* 23, 16380–16387.
- Yu, S., Giroto, S., Lee, C., Magliozzo, R.S., 2003. Reduced affinity for Isoniazid in the S315T mutant of Mycobacterium tuberculosis KatG is a key factor in antibiotic resistance. *J. Biol. Chem.* 278, 14769–14775.
- Zhang, Y., Post-Martens, K., Denkin, S., 2006. New drug candidates and therapeutic targets for tuberculosis therapy. *Drug Discov. Today* 11, 21–27.
- Zhao, H., Caffisch, A., 2015. Molecular dynamics in drug design. *Eur. J. Med. Chem.* 91, 4–14.
- Zwarycz, A.S., Fossat, M., Akanyeti, O., *et al.*, 2017. V67L mutation fills an internal cavity to stabilize reca mtu intein. *Biochemistry* 56, 2715–2722.

Protein-Carbohydrate Interactions

Adeel Malik, Chungnam National University, Daejeon, South Korea

Mohammad H Baig, Yeungnam University Gyeongsan, Gyeongbuk, South Korea

Balachandran Manavalan, Ajou University School of Medicine, Suwon, Republic of Korea

© 2019 Elsevier Inc. All rights reserved.

Introduction

Carbohydrates (also known as sugars or glycans) provide the primary source of energy for living organisms, are used as structural elements, and are the essential components of cofactors (such as ATP, NADPH, etc), glycoproteins and polynucleotides (Quiocho, 1989). These carbohydrates interact with diverse types of proteins and are the basis of many biological processes, both normal and pathological ones including fertilization, immune response, cancer, etc. These interactions also play key roles in several cell adhesion phenomena, among them the attachment of parasites, fungi, bacteria, and viruses to host cells; an indispensable step for the infection (Karlsson, 1991; Ofek *et al.*, 2003). In numerous cases, the protein-carbohydrate interaction may not be a straightforward event but only an initial step that may trigger complex signaling cascades (De Schutter and Van Damme, 2015). Carbohydrates can play so many different roles in molecular recognition due to their ability to generate a broad range of structurally diverse moieties from few monosaccharides that are connected by various linkage types (Audette *et al.*, 2003). Additionally, these carbohydrates can be extremely branched, therefore allowing oligosaccharides to offer nearly a limitless diversity of structural variations.

The significance of protein-carbohydrate interactions in regulating several physiological processes has been acknowledged since decades now. However, the molecular basis underlying such interactions has emerged in recent times. Application of X-ray crystallography to study protein-carbohydrate interactions is the main technique involved, but carbohydrates are quite challenging to crystallize because their inherent flexibility may not be compliant to structural analysis. They may even avert crystallographic studies if this flexible nature on the protein surface hampers the arrangement of crystal contacts, which normally prompts their enzymatic elimination as part of the sample preparation for crystallization (Agirre *et al.*, 2017). Therefore, to get the conformational and dynamical information; the application of NMR spectroscopy was utilized. Because of the typical features of carbohydrates, it is established that relaxation NMR parameters should be integrated with computational methods in order to define the structural characteristics of carbohydrates in an explicit manner. Therefore, to better understand the structure/activity relationships of protein-carbohydrate interactions, interdisciplinary approaches are probably the best options that can be applied to study these complexes (Jiménez-Barbero *et al.*, 1999).

Other traditional approaches, including the hemagglutination inhibition assay (Lis and Sharon, 1972), enzyme-linked lectin assay (McCoy *et al.*, 1983), surface plasmon resonance (Duverger *et al.*, 2003) and isothermal titration calorimetry (Dam and Brewer, 2002), have been used to investigate carbohydrate recognition by proteins. Regardless of their successful application to elucidate the details of protein-carbohydrate interactions, such techniques are labor-intensive and may have certain restraints in terms of requirements of large amounts of pure carbohydrate samples, protein size, solubility, or ease of crystallization, besides the cost (Zhang *et al.*, 2016). Because of these shortcomings the conventional methods seem inapplicable as high-throughput analytic methods (Park *et al.*, 2007). With the introduction of carbohydrate microarrays in 2002 (Wang *et al.*, 2002; Fukui *et al.*, 2002) the limitations of the requirements for low sample quantities and attaining high-throughput parallel screening were overcome (Puvirajesinghe and Turnbull, 2016). However, limited number of efficient analysis tools to extract relevant information limits the use of glycan microarray data (Malik *et al.*, 2014). Rapid advances in the method, in addition to its application in investigation of protein-carbohydrate interactions, it is now possible to successfully discriminate the selective interactions of carbohydrates with bacteria, viruses and eukaryotic as well as live cell responses to immobilized glycans (Puvirajesinghe and Turnbull, 2016). Recently, hydrogen/deuterium exchange mass spectrometry (HDX-MS) was applied to investigate the sites for ligand binding within carbohydrate-binding proteins (Zhang *et al.*, 2016). No doubt these experimental methods are indispensable to fully understand protein-carbohydrate interactions; however, they have certain challenges associated with them such as weak binding affinity and synthetic complexity of specific carbohydrates (Ng *et al.*, 2015). Additionally, other limiting factors that hinder these studies include cost, time and labor. To overcome these challenges, several computational methods to study these types of interactions have been proposed and applied to identify carbohydrate binding sites in proteins.

These computational methods can be divided into three categories: (i) knowledge- or empirical-based methods; (ii) machine-learning based methods; and (iii) molecular dynamics based methods. Here, we discuss these methods that have been applied to study protein carbohydrate interactions.

Knowledge- or Empirical-Based Methods

A three-dimensional (3D) structure of a protein-carbohydrate complex obtained *via* X-ray crystallographic or NMR based methods is desirable to understand the interactions involved. Because of the limitations of these experimental methods to determine such complexes, knowledge-, empirical-, and structure-based methods have been developed for the prediction of glycan interacting

residues from the 3D structures. (Taroni *et al.*, 2000) calculated six parameters [relative accessible surface area (RSA), residue propensity, solvation potential, planarity, hydrophobicity, and protrusion] of glycan binding sites on a set of 19 non-homologous carbohydrate binding proteins using PATCH program (Taroni *et al.*, 2000). These parameters were then utilized to resolve if the surface patch under scrutiny is a carbohydrate-binding site. This analysis revealed that RSA, protrusion index and residue propensity were the three best parameters that differentiate the carbohydrate binding sites from other surfaces. Evaluation of this method on two independent datasets (Test set1, which does not show any homology to the main dataset, and, Test set2 which consists of structures homologous to either the main data set or the Test set1) showed the overall prediction accuracy of 65%, which was a higher success rate for enzymes as compared to lectins.

In another work, empirical rules of the spatial distribution of protein atoms around carbohydrate-binding sites were used for prediction (Shionyu-Mitsuyama *et al.*, 2003). This method also requires a known 3D structure of a protein-carbohydrate complex to construct the empirical rules, and subsequently, these rules were utilized to search the carbohydrate binding sites on the target protein. The empirical rules were constructed using a non-redundant data set of 80 protein-carbohydrate complexes and the prediction system was tested on an independent dataset of 50 experimentally derived complexes. This method worked better for Galactose, Glucose and *N*-Acetylglucosamine binding sites as compared to Mannose binding sites. The attributes or properties used between the above-mentioned two methods is different in a sense that the former method (Taroni *et al.*, 2000) uses amino acid propensity whereas the latter system employs the spatial distribution of amino acid residues at the glycan-binding sites (Shionyu-Mitsuyama *et al.*, 2003).

Sujatha and Balaji (2004) developed a program called COTRAN to search for the potential galactose-interacting sites in proteins. They utilized 18 protein-galactose complexes belonging to 7 non-homologous families and investigated the galactose-binding sites. This method utilizes a combination of the inferred geometrical characteristics in addition to structural features [for example, accessible surface area (ASA) and secondary structure] to detect a possible galactose-binding site signature with very high accuracy.

Recently, Zhao *et al.* (2014) developed a structure-based function-prediction method called SPOT-Struc that recognizes carbohydrate-binding proteins (Zhao *et al.*, 2014). This method predicts interacting residues by structural alignment program SPAlign (Yang *et al.*, 2012) and knowledge-based statistical potential [distance-scaled finite-ideal gas reference state (DFIRE)] is used for predicting the binding affinity scoring. In this method, a template library of 523 (T523) carbohydrate binding proteins (CBPs) was derived from PROCARB (Malik *et al.*, 2010) database. Additionally, a positive binding domain dataset consisting of 113 CBPs (BD113) and a negative dataset of 3442 non-CBPs (NB3442) were constructed at 30% sequence identity cutoff. The prediction system works by first aligning the target structure against the templates (T523 dataset) by structural alignment tool SPAlign, where the structural similarity score is calculated by a scoring function called SP-score (Yang *et al.*, 2012). In case the SP-score is larger than a given threshold, the model for the complex structure between the query protein and template carbohydrate is generated by substituting template protein structure with the query structure in the template complex structure. The model complex structure is further used to calculate the binding affinity by the DFIRE (Zhou and Zhou, 2002; Yang and Zhou, 2008). The query is predicted as a CBP if the binding affinity is above a certain threshold; else non-CBP. Finally, the predicted structures from SPOT-Struc are utilized for the prediction of interacting residues. If any heavy atom of an amino acid is within a contact distance of < 4.5 Å from any heavy atom of the carbohydrate was defined as a binding site. A leave-one-out cross-validation (LOOCV) method on BD113 and NB3442 datasets achieved the MCC of 0.63 and 0.58 for the prediction of CBPs and carbohydrate-interacting residues, respectively at 52% sensitivity and 79% positive predictive value (PPV) of CBP prediction. A comparable level of accuracy was observed for two additional datasets of bound (HOLO) and unbound (APO) structures of CBP.

An approach based on the energy function for identifying the carbohydrate binding sites has also been proposed (Gromiha *et al.*, 2014) that uses AMBER force field with GLYCAM06 (Kirschner *et al.*, 2008) parameters for calculating the interaction energy between the atoms in a protein and the bound carbohydrate. The residues that are involved in interactions with carbohydrates have been analyzed in terms of binding segments which is based on the number of sequential binding residues in protein sequences. For instance, a 4-residue binding segment has a region of four back-to-back binding residues. This study generated several statistics, including incidence of amino acid residues at different interaction energies, binding propensity scores of amino acid residues in protein-carbohydrate complexes, contribution of various atom types in these types of interactions, inclination of dipeptides and tripeptides around the binding sites, highlighting the significance of stronger atomic level contributions from carbohydrates as compared to proteins.

Machine Learning (ML)-Based Methods

ML-based methods have been successfully applied in both sequence- and structure-based methods. Although, ML-based approach is the same but the input features or attributes are different between sequence- and structure-based methods (Fig. 1). In general, different features are extracted from the protein sequences or structures (including knowledge-based information) and are used as inputs for ML methods, and the binding site is obtained from them. Extensive use of ML has been applied to address various biological questions. The biggest advantages of ML is that it can deal with multiple features or attributes at the same time and capture the hidden relationships among them, which are otherwise challenging to derive with knowledge- or empirical-based methods (Manavalan *et al.*, 2014; Manavalan and Lee 2014a; Manavalan *et al.*, 2017b,c; Alpaydin, 2014; Bastanlar and Ozuysal, 2014; Nilsson, 1996). In the following subsections, we discuss sequence- and structure-based methods in which ML has been applied to study protein-carbohydrate interactions.

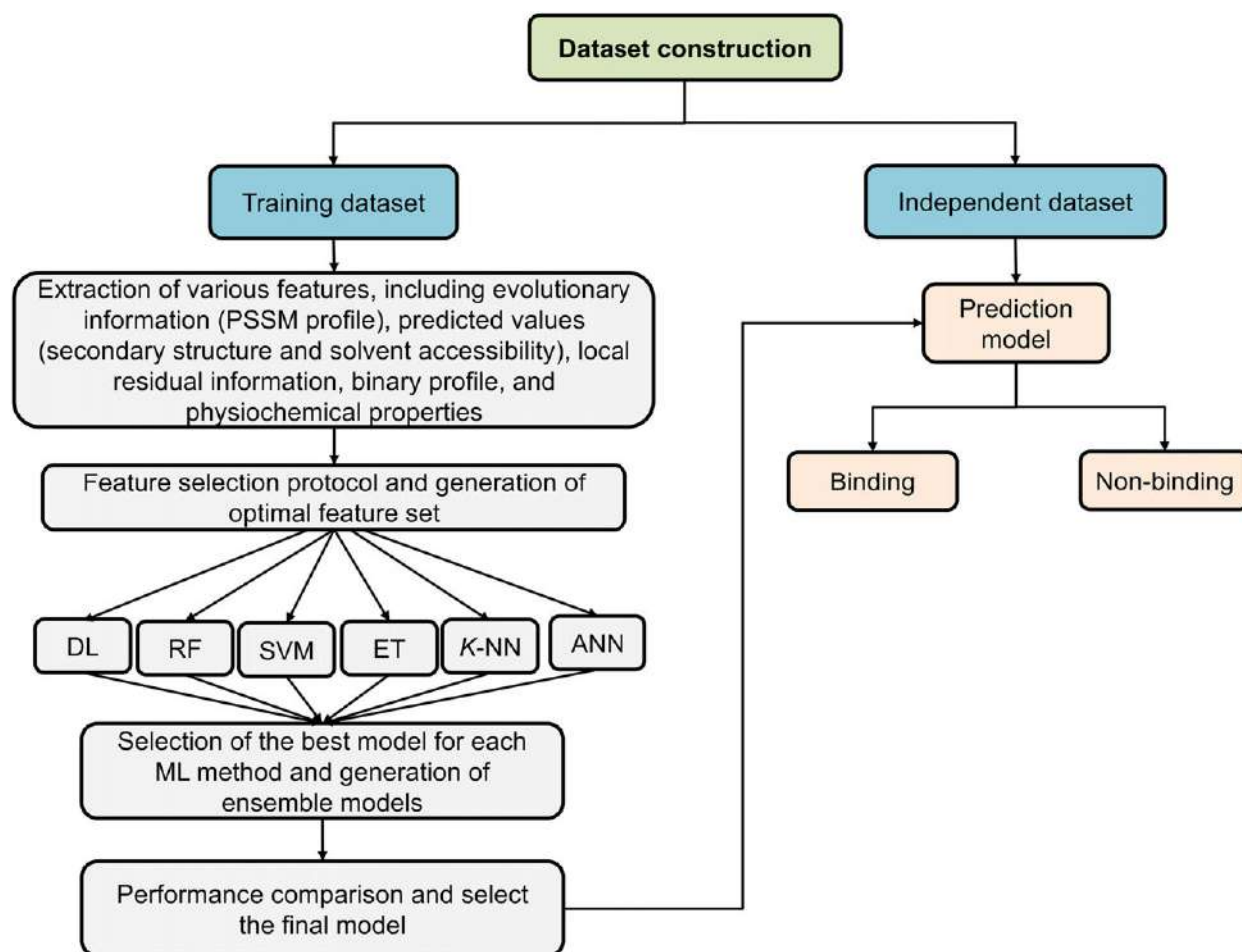


Fig. 1 An overview of ML-based prediction framework. It involves four different stages: (i) construction of the datasets; (ii) extraction of features from the sequence or structures and the selection of important features; (iii) development of the prediction models using various ML methods; (iv) final model selection based on their performances in training dataset and independent dataset.

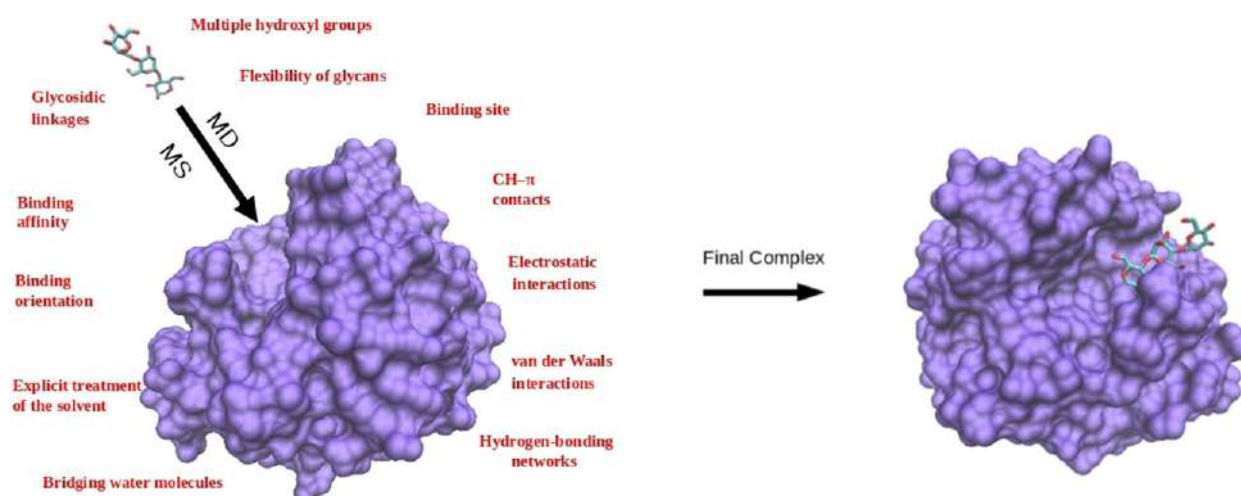


Fig. 2 A summary of important Molecular simulations (MS) and Molecular docking (MD) parameters that need to be considered while studying protein-carbohydrate interactions by using molecular dynamics methods. The coordinates of PDB ID 3MAN was used to generate the figure by using VMD (Humphrey *et al.*, 1996).

Structure-Based Prediction Methods

The first structure-based method using ML was proposed in 2009, where they applied support vector machine (SVM) to predict the glucose binding sites (Nassif *et al.*, 2009), whose optimal features were identified using random forest (RF) algorithm. This study revealed that in addition to carboxylate residues, the significance of ordered water molecules and ions might participate in deciphering glucose-binding specificity. Additionally, the relevance of high concentration of negatively charged atoms that make direct contacts with the glucose was highlighted. Using LOOCV, this method achieved 8.11% error, 89.66% sensitivity and 93.33% specificity with a dataset of 43 glucose- and 68 non-glucose-binding sites. Similarly, mannose binding sites were also predicted by using RF by extracting at least 15 features (e.g., charge, hydrophobicity, hydrogen bonding, etc.) both at the atomic as well as residue level (Khare *et al.*, 2012). This method achieved an accuracy of 96% and 94% with respect to individual feature and combined features, respectively. However, the applicability of this prediction model remains to be determined because a very small dataset (only 11 protein-mannose complexes) was used for the prediction model development. The common feature of two structure-based methods and the knowledge-based method by Sujatha and Balaji is their specificity for a particular type of sugar (e.g., glucose, mannose or galactose).

ML algorithms have also been applied to predict non-specific carbohydrate binding sites in proteins. A method called InCa-SiteFinder, which utilizes the amino acid propensities and van der Waals energy of a protein-probe interaction for searching or predicting the non-covalent inositol and non-specific glycan binding sites on the protein structures (Kulharia *et al.*, 2009). This method used a dataset of 375 protein-carbohydrate complexes (which is larger compared to the earlier studies) to derive amino acid propensities. InCa-SiteFinder uses a two-step procedure to predict the carbohydrate binding sites. The first step employs an energetic grid-based approach (Laurie and Jackson, 2005) to detect potential sites on the protein with a higher possibility of being a binding site, whereas in the next step it uses potential sites information along with the amino acid propensities to predict the region of carbohydrate binding sites. The performance of InCa-SiteFinder was assessed by a single parameter, tau (τ) which in turn is a product of two individual parameters, precision (P) and coverage (C) (*i.e.*, $\tau = P \times C$). This method performed reasonably well when tested on 80 protein-ligand complexes. Furthermore, this method achieved 98% specificity and 73% sensitivity with an overall error rate of 12%, when tested on dataset of 40 known drug-like compound binders as well as 40 known carbohydrate binders to predict the carbohydrate binding site locations and to further distinguish both the types from one another.

Another method was developed which was different as compared to earlier described methods in a sense that it utilizes a unique encoding scheme of the 3D probability density maps representing the distributions of 36 non-covalent interacting atom types around protein surfaces (Tsai *et al.*, 2012). The 36 atoms represent 30 atom types from proteins, 5 from carbohydrate ligands, and 1 from the water. A ML model was trained for all the 30 types of protein atoms. In ML models, each attribute is representing the normalized distance-weighted sum of the 3D probability density for the one interacting atom type (out of 36) on the protein surface. Three types of ML algorithms were used to predict the carbohydrate binding sites: (a) Artificial neural network (ANN), (b) SVM, and (c) ANN with bootstrap aggregation (ANN_BAGGING). For training as well as testing, only atoms on protein surfaces that have solvent accessible area of larger than 0 were included. These ML models were trained to learn the patterns of the attributes to discriminate between binding- and non-binding site atoms. Furthermore, the probability of each amino acid atom assumed to be within a glycan binding site was scaled into a prediction confidence level between 0 and 1. Surface atoms of protein with greater certainty levels from the predictors were grouped to output provisional glycan-binding patches. The predictors were validated on a non-redundant dataset of 497 proteins with known binding sites for carbohydrate by a 10-fold cross-validation, and additionally tested on an independent dataset of 108 proteins. ANN_BAGGING method produced the best result with an average Matthews correlation coefficient (MCC), sensitivity, and specificity of 0.45, 0.49, and 0.97 respectively, when compared to other two methods.

In general, the above methods are dependent on the protein's local structural descriptors and hence are of little significance in case of the unavailability of their 3D structures.

Sequence-Based Prediction Methods

It is well known that the frequency of mutations at functional sites during evolution is much lesser as compared to the other parts of a protein (Livingstone and Barton, 1993; Zvelebil *et al.*, 1987). The difference in amino acids at these functional sites can be represented in the form of sequence conservation patterns by using matrices of position-specific variations, known as profiles (Gribskov *et al.*, 1990), or by statistical methods, such as Hidden Markov Models (HMMs) (Krogh *et al.*, 1994; Eddy, 1998). Sequence homologies (local as well as global) can be detected by these methods or by more sensitive approaches such as Position Specific Iterative (PSI) blast program (Altschul *et al.*, 1997) search, thereby implying putative functions and functional sites, by comparison. Sequence-based methods exploiting ML have become indispensable to the functional annotation of proteins since decades (Kneller *et al.*, 1990; Ahmad and Sarai, 2005; Liu and Rost, 2004; Firoz *et al.*, 2011). In addition to many other disciplines of bioinformatics, ML methods have been extensively employed in the prediction of interaction sites between two biomacromolecules. In general, ML refers to the modifications in systems that carry out tasks such as recognition, diagnosis, planning, robot control, prediction, etc., which are linked with artificial intelligence (AI) (Nilsson, 1996). The main objective of ML is to program computers to use model data or previous experience to resolve a problem at hand (Alpaydin, 2014). The rapid upsurge in biological data as well as the processing potential and storage capacity of computers has led to enhanced developments in the field of ML (Bastanlar and Ozuysal, 2014). Given a protein sequence which is expected to interact with another molecule (DNA, RNA, carbohydrates, ligands, etc.) the challenge of ML is to guess which amino acid residues might be involved in the interactions (Wang and Brown, 2006).

In sequence-based methods, there are two types of approaches that can be used to encode input protein sequences (Hwang *et al.*, 2007): (a) Single sequence approach – Each amino acid residue is converted into a 20 bit vector and propagated through a ML model along with its neighboring residues (Gallet *et al.*, 2000; Ofra and Rost, 2007) as well as combining additional structural information (Ahmad and Sarai, 2005). (b) Evolutionary information (EI) – This approach utilizes residue conservation and EI (position specific scoring matrices (PSSMs)) by using PSI-BLAST (Altschul *et al.*, 1997). In this method, the target sequence is searched for homology against a protein database and generates a profile for the input sequence in the form of scoring matrices, which is then supplied to a ML algorithm such as neural networks or SVM for prediction. Sequence profiles of neighboring residues and structural information have increased the prediction accuracy (Hwang *et al.*, 2007; Wang and Brown, 2006; Zhou and Shan, 2001; Kuznetsov *et al.*, 2006).

Several developments have been made for the prediction of protein-protein interaction sites (Zhou and Shan, 2001; Ofra and Rost, 2007; Li *et al.*, 2012; Fariselli *et al.*, 2002), prediction of DNA (Ahmad and Sarai, 2005; Carson *et al.*, 2010; Ofra *et al.*, 2007) and ligand binding sites in proteins (Passerini *et al.*, 2006; Shu *et al.*, 2008; Chen *et al.*, 2014; Suresh *et al.*, 2015). Recognizing the significance of protein-carbohydrate interactions, Malik and Ahmad (2007) developed the first sequence-based method using ANNs to predict carbohydrate binding sites in proteins (Malik and Ahmad, 2007). They trained two separate neural network architectures using evolutionary profiles of protein sequences and information from single sequences to predict carbohydrate binding sites. The network using EI achieved 87% sensitivity of prediction at 23% specificity whereas with the single sequences, 68% sensitivity and 55% specificity for all carbohydrate-binding sites was obtained. The authors proposed that the prediction accuracy can be improved further when their method trained with the larger size of the dataset in the future. Although, this method was developed using the dataset of 40 protein-carbohydrate complexes, it showed an improved performance when tested on a dataset generated by Sujatha and Balaji (2004). The method was later implemented in the form of a web server, CBS-Pred (Malik *et al.*, 2010) and can be accessed at see “Relevant Websites section”. The work triggered interest within the scientific community which resulted in the development of new sequence-based methods to predict carbohydrate binding sites by employing ML. Most of these methods focused on a specific type of sugar. For example, SVM based classifiers were developed to predict mannose interacting residues (MIRs) from the primary sequences of mannose binding proteins (MBPs) (Agarwal *et al.*, 2011). The method employs several modules such as similarity-based module, SVM based module which uses binary profile of patterns, SVM module using EI, and a module on the basis of composition profile of patterns (CPP) or local composition for predicting mannose binding residues in proteins. Among these different modules, the CPP based SVM method outperformed other modules by achieving a maximum MCC of 0.61. A method called MOWGLI was also developed to predict MIRs that uses an ensemble of base classifiers using PSSM (Pai and Mondal, 2016). In this method, the base classification algorithm was developed by exploring RF and SVM trained by various subsets of training data set of 128 mannose binding proteins using 10-fold cross-validation. The performance of the method was measured on a separate test dataset of 29 mannose binding proteins and achieved an accuracy of 92%. A standalone version of this method is available at see “Relevant Websites section”. With the exception of CBS-Pred (Malik *et al.*, 2010), the type of interacting carbohydrate is known in all of the above-mentioned methods. However, it becomes more difficult when such information is unknown well in advance. Another sequence-based method that integrates sequence and predicted structural features for the prediction of nonspecific carbohydrate binding sites was developed (Taherzadeh *et al.*, 2016). The method in addition to EI, uses sequence derived structural information (such as solvent accessibility and secondary structure), seven physico-chemical features (volume, steric parameter, isoelectric point, hydrophobicity, polarizability, alpha-helix and beta-sheet probability) of amino acids, information about the disordered residues, and the protein length as the input features. The most effective features were selected to build a classifier based on SVMs and the method achieved an area under receiver operating characteristic curve (AUC) of 0.78 and 0.77 for 10-fold cross-validation and independent test.

Recently, a method was developed which classifies sugars into acidic and non-acidic sugars and discriminates between their amino acid composition at the binding sites (Banno *et al.*, 2017). These characteristics were then used to design two predictors viz, an acid sugar binding predictor and a non-acidic sugar binding predictor. Additionally, a third predictor was developed that combined the output of the two above mentioned predictors. The dataset used to develop this method consists of 369 sugar-binding proteins, 136 acidic sugar-, and 270 non-acidic-sugar-binding proteins. A PSSM profile of sugar-interacting sites and their neighboring residues were employed to predict the sugar binding sites by using SVM. The method achieved an AUC of 0.787 and 0.752 for the acidic and non-acidic sugar binding predictors, respectively, on the basis of a 5-fold cross-validation. In case, the characteristic nature of sugar was unknown or is known to be non-acidic, the combined predictor performed best with AUC of 0.760. The method is freely available for download as a standalone program (see “Relevant Websites section”).

To date, there are three structure-based methods and five sequence-based methods publicly available for the prediction of carbohydrate binding sites. The web server link and other details are provided in Table 1.

Advantages of sequence-based methods over structure-based methods

Due to the advancement in sequencing technology, the number of sequences deposited in the protein sequence database is growing exponentially, which has outpaced the experimental determination of protein structures. This has created a big void between the number of protein sequence entries and their experimentally determined 3D structures. This sequence-structure gap is expected to increase further in the future. As a result, sequence-based bioinformatics tools have become indispensable for researchers as they are not dependent on a costly and time-consuming procedure of experimental determination of protein structures (Hwang *et al.*, 2007). Moreover, sequence-based tools have attractive leverage in managing high-throughput data generated in the fast-growing proteomics field. Although, sequence-based methods have several advantages, the prediction accuracy is slightly lower than the structure-based methods. Their accuracy may become better once the key question to identify the best features that may help in improving the prediction of these binding sites is addressed.

Table 1 List of available methods to predict carbohydrate-binding sites in proteins. The methods highlighted in bold font and normal fonts respectively denote the structure- and sequence-based methods

Method	Link	Carbohydrate specificity	Status (As of 12th Oct 2017)	Reference
InCa-SiteFinder	http://www.modeling.leeds.ac.uk/InCaSiteFinder/	Non-specific	In active	Kulharia <i>et al.</i> (2009)
ISMBLab_PCA	http://ismlab.genomics.sinica.edu.tw/	Non-specific	Active	Tsai <i>et al.</i> (2012)
SPOT-CBP	http://sparks-lab.org/yueyang/server/SPOT-CBP/	Non-specific	Active	Zhao <i>et al.</i> (2014)
CBS-Pred	http://www.procarb.org/procarb/cbs-pred.html	Non-specific	Active	Malik <i>et al.</i> (2010)
PreMieR	http://crdd.osdd.net/raghava/premier/	Mannose	Active	Agarwal <i>et al.</i> (2011)
MOWGLI	https://sites.google.com/site/sukantamondal/software	Mannose	Active	Pai and Mondal (2016)
SPRINT-CBH	http://sparks-lab.org/server/SPRINT-CBH/	Non-specific	Active	Taherzadeh <i>et al.</i> (2016)
SBRP	https://zenodo.org/record/61513#.WcB-4hdLeis	Acidic and non-acidic sugars	Active	Banno <i>et al.</i> (2017)

Table 2 Summary of the parameterization protocols for some of the well known force-fields dedicated to carbohydrates and an example application of these force fields in simulating the protein-carbohydrate complexes

Force field	Parameterization protocol used for the development					Example application
	Vdw terms	Torsion terms	1,4-Scaling (Elec/vdW)	Unique atom types for α - and β -anomers	Unique charge sets for α - and β -anomers	
GLYCAM	AMBER PARM94	Fit to QM rotational energy curves	X	x	✓	Mishra <i>et al.</i> (2014)
CHARMM	CHARMM22		X	✓	x	Mallajosyula <i>et al.</i> (2015)
OLPS-AA-SEI	OPLS-AA		✓	✓	✓	Kony <i>et al.</i> (2002)
GROMOS	GROMOS-45A4		X	✓	✓	Vital De Oliveira (2014)

Note: For detailed review of these force fields, interested readers are referred to Fadda, E., Woods, R.J., 2010. Molecular simulations of carbohydrates and protein-carbohydrate interactions: Motivation, issues and prospects. *Drug Discov. Today* 15, 596–609 and Xiong, X., Chen, Z., Cossins, B.P., *et al.*, 2015. Force fields and scoring functions for carbohydrate simulation. *Carbohydr. Res.* 401, 73–81 of the reference section.

In general, structure-based methods always perform better than the sequence-based methods. This is mainly due to the following reasons: (i) structure-based methods contain features derived from both the sequences as well as from the structures. Basically, it includes a sequence-based method and improving the corresponding algorithm by adding the structural information (ii) the information/features derived from the structures is more accurate when compared to the sequences. However, this method requires a 3D structure as an input, but again solving a 3D structure is not simple, and requires lot of time and effort. As a result, sequence-based methods are preferred.

Molecular Dynamics Based Methods

Force Fields

Molecular dynamic simulation studies are performed to gain a possible observation of the progression of motions in the molecular systems at atomic level (Hansson *et al.*, 2002). These are based on the deterministic propagation of various forces interim on a molecular system. These computer simulation processes have provided us with powerful tools to deeply investigate the detailed molecular interactions. For a successful simulation, we need an accurate force field which could provide a realistic portrayal of the systems. The development of accurate force-fields for protein-carbohydrate interactions has been an area of active research. It is a very laborious task to develop an accurate force field compatible to deal with the carbohydrates (Xiong *et al.*, 2015). Over the last couple of decades a large number of force fields capable of dealing with carbohydrates were developed. These force fields were developed by outspreading the classic force fields making them to deal with different carbohydrates.

Here we are listing some of the major force fields specifically developed to handle carbohydrates. These force fields are either dedicated ones for carbohydrates or they are an extension of the previously used force fields with changes implemented. GLYCAM06, CHARMM36, 53A6GLYC, GROMOS and OPLS-AA-SEI (Scaling Electrostatic Interaction), are some of the major carbohydrate force fields (Table 2). These force fields were developed for better consistency for simulating the larger systems including nucleic acids, proteins, and lipids.

GLYCAM06

This force field is compatible for a large group of carbohydrates, in addition to acetylated amino sugars, neuraminic acid, and iduronic acid (Kirschner *et al.*, 2008). This force field is widely used for modeling the protein-carbohydrate complexes in addition

to the modeling of carbohydrates, glycolipids and glycoproteins (Tessier *et al.*, 2008; DeMarco and Woods, 2008). GLYCAM06 is compatible with the AMBER force field (Fadda and Woods, 2010) and comprises of parameters appropriate for the simulation of carbohydrates (Kirschner *et al.*, 2008). These parameters of GLYCAM06 can be effectively used for simulating carbohydrates of different conformations and ring sizes (both oligosaccharides and monosaccharides). Using appropriate file-conversion tools, GLYCAM06 is fit for use in other simulation packages (Dejoux *et al.*, 2001). Additionally, GLYCAM06 is loaded with parameters developed by considering a test set of 100 molecules from different chemical families, like carboxylates, alcohols, amides, esters, hydrocarbons, ethers and other molecules of diverse functional groups which have been fitted to quantum-mechanical data (Perez and Tvaroska, 2014). Several stereoelectronic effects influencing the bond and angle variations at the anomeric carbon atom have been parametrized in this force field (Biarnès *et al.*, 2010). These features allow it for mimicking the ring inversion observed in glycosidic monomers during several catalytic events. The comparison of the GLYCAM06 derived results with experimental findings suggested that this force field correctly reproduces the rotational energies and carbohydrate features (Guvench *et al.*, 2009).

CHARMM36

The force field for handling carbohydrates is one of the modifications added to CHARMM all-atom biomolecular force fields (Hatcher *et al.*, 2009; Guvench *et al.*, 2008; Guvench *et al.*, 2009; MacKerell *et al.*, 1998, 2004). This updated force field is embedded with parameters allowing the simulation of unsubstituted oligosaccharides as well as monosaccharides. A large number of revisions for handling carbohydrates have been implemented in the upgraded version of this force field. These updates extend the capability of this force field for handling the five-membered sugar rings and oligosaccharides. The same hierarchical parameterization process and handling of 1,4 nonbonded interactions are exploited to guarantee complete compatibility with other CHARMM biomolecular force fields (MacKerell *et al.*, 1998). The current parameterization of CHARMM makes it compatible for simulating the monosaccharides (i.e., glucopyranoses and diastereoisomers (Guvench *et al.*, 2008) and aldo- and fructofuranoses (Hatcher *et al.*, 2009) and for unsubstituted oligosaccharides (Guvench *et al.*, 2008).

OPLS-AA-SEI

OPLS-AA Scaling of Electrostatic Interactions also known as OPLS-AA-SEI, an upgraded version of OPLS-AA biomolecular force field with parameters for carbohydrates for the improved prediction of its conformational changes (mainly Φ and Ψ angles) (Kony *et al.*, 2002). In comparison to OPLS-AA, OPLS-AA-SEI has been reported to be more accurate in terms of reproducing quantum mechanics (QM) relative energies.

GROMOS

Two different versions for dealing with carbohydrates have been developed. The first one is the classic GROMOS 45A4 and the advanced one is GROMOS 53A6GLYC. The GROMOS 45A4 is compatible for dealing with mono-, di-, oligo-, or polysaccharides (Lins and Hunenberger, 2005) and comprises of advance and better parameters designed specifically for carbohydrates. It also consists of parameters for handling mono- and oligosaccharides on the basis of pyranosides. GROMOS 53A6GLYC is another improved version with more advanced features set for simulating the carbohydrates in explicit-solvent. This version allowed appropriate depiction of the best possible conformation out of all 16 aldohexopyranose-based monosaccharides (Pol-Fachin, 2017).

Molecular Docking

As compared to protein-ligands, the interaction of a protein and a carbohydrate is more dynamic (Fadda and Woods, 2010). The affinities of these interactions are more likely dependent on several relatively weak interactions. In protein-carbohydrate-interactions, CH- π interactions and hydrogen bonds are the main types of interactions (Spiwok *et al.*, 2005). Molecular docking of proteins and their interacting carbohydrates is still a challenge, due to the inaccuracy of scoring functions for accurate prediction of their interaction. At present, very limited number of carbohydrate specific scoring functions have been developed and reported. It is a great need to develop a scoring function capable of accurately predicting the correct binding pose, along with binding free energy of the complex.

Some major docking tools like AutoDock (Morris *et al.*, 2008), Glide (Friesner *et al.*, 2004), GOLD (Verdonk *et al.*, 2003), and FlexX (Cross, 2005) are in use for studying protein-carbohydrate interactions. In a recent study the ability of these docking tools were compared and assessed. The top ranked poses generated by each of these tools was compared to its already known crystal counterpart. This comparison highlighted the weakness and strength of each tool. The findings of this study revealed the best results were obtained by GLIDE (Agostino *et al.*, 2009). Though the finding of this study shows the RMSD of the predicted results were in range with the experimentally determined structures, however, the best predicted results do not correspond to the experimental data.

The findings of several studies conducted on protein-carbohydrate interactions gives an impression that it is still extremely challenging to develop accurate and efficient methods for calculating the binding affinity of such complexes (DeMarco and Woods, 2008). In the last decade, a significant increase in the application of molecular docking methods for studying protein-carbohydrate interactions was observed (Reina *et al.*, 2008; Takaoka *et al.*, 2007; De Geus *et al.*, 2009; Chen and Shoichet, 2009; Gomez *et al.*, 2016; Bakkers *et al.*, 2016; Mompean *et al.*, 2017; Walker *et al.*, 2017). For a successful docking, an efficient searching algorithm along with a robust scoring function is critically required (Hill and Reilly, 2008). In case of protein-carbohydrate

interactions the bridging water molecules and CH- π interactions are also key role players (Raju *et al.*, 2009; Spiwok *et al.*, 2005; Vandenbussche *et al.*, 2008). The more commonly used molecular docking programs had certain limitations in dealing with these important factors involved in the interaction of carbohydrates to proteins.

Limitations of traditional molecular docking tools

It is very challenging for the regularly used docking programs to identify the shallow carbohydrate binding sites. In addition, compared to other organic molecules, carbohydrates are significantly more flexible (Kerzmann *et al.*, 2008; Imberty, 1997). These widely used molecular docking programs are mainly developed for analyzing the binding of small molecules. The use of these traditional docking methods is not capable to accurately predict the orientation of carbohydrate within the binding site and the relative binding affinity of this conformational pose (interaction energy). There are several factors which are responsible for their inability to deal with the carbohydrates. Unlike small ligands, carbohydrates are extremely flexible, and consist of several hydroxyl groups (DeMarco and Woods, 2008). The presence of wide hydrogen-bonding networks along with the formation of CH- π contacts are the major reasons which restrict these traditional docking tools from accurately predicting carbohydrate-protein complexes (Spiwok, 2017; Grant and Woods, 2014).

Moreover, the explicit treatment of the solvent is required for capturing the hydrogen bonds between carbohydrates with water (Nivedha *et al.*, 2016, 2014; Kirschner and Woods, 2001). Some of the essential parameters required for the efficient docking of protein-carbohydrate interactions are summarized in Fig. 2. Due to these major challenges, it is difficult to correctly predict the carbohydrate binding sites and their affinities using traditional docking methods.

Advancement towards designing computational methods for protein-carbohydrate docking

Several approaches were made towards preparing a better method for accurate prediction of protein-carbohydrate interactions. Studies performed on carbohydrate docking used the scoring functions which were customized either by recalibrating the existing terms (Laederach and Reilly, 2003) or by including few more additional functions capable of modeling some of the specific features of protein-carbohydrate interactions (Kerzmann *et al.*, 2006). Kerzmann *et al.* (2006) specifically designed a scoring and an energy function termed as Sugar-Lectin Interactions and DoCKing (SLICK), for predicting the binding poses and free energy calculations for carbohydrates to proteins. This scoring function accounts for CH- π interactions, solvation effects, hydrogen bonds, van der Waals interactions and electrostatics. The binding free energies predicted using SLICK showed high accuracy when compared to the experimental binding energies. This is due to the calibration of the parameters for the empirical energy function. This function is available in the molecular modeling package of BALLDock (see "Relevant Websites section") (Gauto *et al.*, 2013). The glycosidic linkages within oligosaccharides are the critical source of flexibility. In previous studies, scoring functions were customized for carbohydrate docking. This customization was done by either recalibration of existing terms or the annexation of supplementary functions capable of modeling the protein-carbohydrate interactions. The SLICK scoring function comprises of an energy term for CH- π stacking interactions and was calibrated by using a set of lectin-carbohydrate complexes (Kerzmann *et al.*, 2006). Later it was found that incorporating the explicit water molecules during the docking of carbohydrates (Samsonov *et al.*, 2011) as well as inclusion of the placement of hydroxyl groups of carbohydrates in water site positions (Gauto *et al.*, 2013) leads to improvements in docking accuracy. Inclusion of the conformational preferences of oligosaccharides within a scoring function additionally enhances the accuracy and efficiency of carbohydrate docking. Recently, Carbohydrate Intrinsic (CHI) energy functions assign relative energies to the torsion angles of the glycosidic linkages. The distribution of glycosidic torsion angles in protein-carbohydrate complexes obtained by CHI-energy profiles resembled with those obtained from the Protein Data Bank (PDB) (Nivedha *et al.*, 2014).

Reports also suggest the inclusion of Carbohydrate Intrinsic (CHI) energy functions in carbohydrate docking algorithms would be beneficial (Nivedha *et al.*, 2014). The CHI-energy function was incorporated into the AutoDock Vina (ADV), and was termed Vina-Carb (VC) (Trott and Olson, 2010; Nivedha *et al.*, 2016). For the development of Vina-Carb, a dataset consisting of 29 apoprotein structures and 101 protein-carbohydrate complexes were used. Findings show that the addition of CHI-energy function within the Vina-Carb increases its success rate from 55% to 74% (Nivedha *et al.*, 2016). However, while dealing with the ligands that partially extend into solution, Vina-Carb was unable to reproduce the experimental findings (Nivedha *et al.*, 2016). This is an issue which needs to be addressed for a successful development of scoring function for accurately describing the protein-carbohydrate interactions. Recently, RosettaCarbohydrate Framework was developed for the modeling and docking of oligomeric and polymeric carbohydrate ligands and glycoconjugates (Labonte *et al.*, 2017). The framework can handle the sampling of glycosidic bonds, side-chain conformations, and ring forms, and it can also use a glycan-specific term within its scoring function.

Applications

Among the currently available methods for the prediction of carbohydrate binding sites, two each from the sequence- and structure-based methods have been presented here (Table 3). Briefly, CBS-Pred has been used in three such studies and predicted three aromatic residues involved in polysaccharide binding (Park *et al.*, 2014), lipopolysaccharide binding amino acids in several peptides (Brzozowska *et al.*, 2015), and mannose interacting residues in CrataBL glycoprotein (Pol-Fachin, 2017). SPRINT-CBH was also used to predict the mannose interacting residues of CrataBL, and a carbohydrate binding tyrosine residue of an unknown gene PA5359 from *Pseudomonas aeruginosa* (Gunasekera *et al.*, 2017). In case of structure-based methods, ISMBLab_PCA has been used most commonly. For example, it was used for the prediction of carbohydrate binding sites in a toxin, pneumolysin

Table 3 Application of different carbohydrate binding site prediction methods. The methods highlighted in bold are structure-based and require a 3D structure of a protein for prediction

<i>Method</i>	<i>Types of carbohydrates under investigation</i>	<i>Reference</i>
CBS-Pred	Polysaccharide Monosaccharide Lipopolysaccharide	Park <i>et al.</i> (2014) Pol-Fachin (2017) Brzozowska <i>et al.</i> (2015)
SPRINT-CBH	Carbohydrate binding Monosaccharide	Gunasekera <i>et al.</i> (2017) Pol-Fachin (2017)
InCa-SiteFinder	Druggable pocket	Dharra <i>et al.</i> (2017)
ISMBLab_PCA	Monosaccharide Sialyl Lewis X, Lewis B and mannose LewisX and Sialyl LewisX	Pol-Fachin (2017) Lawrence <i>et al.</i> (2015) Shewell <i>et al.</i> (2014)

(Shewell *et al.*, 2014; Lawrence *et al.*, 2015). In addition to the sequence-based prediction, the mannose binding sites of CrataBL (Pol-Fachin, 2017) were also predicted by using ISMBLab_PCA.

In the case of docking, over the past two decades various attempts have been made to dock carbohydrates to their protein partners (Imberty, 1997; Bitomsky and Wade, 1999; Mulakala and Reilly, 2005; Gohier *et al.*, 1996). The docking ability of several softwares was also compared to calculate the binding affinity of *Ralstonia solanacearum* lectin (RSL) (Mishra *et al.*, 2012). More recent docking methods have also been applied to investigate the interactions of carbohydrates and their interacting proteins. Being published in 2016, a limited number of studies highlighting the application of Vina-Carb are available. For example, a recent study by Gómez *et al.* (2016) demonstrated the use of Vina-Carb for docking of hexasaccharide into the endo- β -1,4-xylanase (Xyl2) active site. Since the methods for specifically docking the carbohydrates are just beginning to appear, it will be a while before their accuracy could be tested for a wide range of carbohydrates.

Future Directions

The information about the factors, which direct or inhibit the binding of a carbohydrate to an amino acid, is believed to exist in the evolutionary profile of the sequence as well as the identity and structure of amino acid residues in the neighboring environments of potential carbohydrate binding sites (Malik and Ahmad, 2007). Although several computational methods have been developed for the protein-carbohydrate interactions, yet none of these methods can offer satisfactory predictions given the diversity of carbohydrates. Methods dealing with specific types of carbohydrates might be promising but they have a limited practical application, whereas the predictors for non-specific carbohydrates may have a broader application, however, at the cost of accuracy. Another issue with the existing methods is that there is no consensus definition for a binding site. The binding and non-binding residues are defined based on a certain distance cut-off. Therefore, it is necessary to optimize the distance cut-off threshold because the introduction of higher cut-off may introduce many false positive and false negative in the dataset (Ding *et al.*, 2010). Currently, the existing sequence-based methods use similar features, small datasets for their prediction model development, no systematic feature selection protocol was employed to quantify the important features, and none of the methods explored different ML methods using the same dataset to identify the most appropriate one for this problem. The key question, however remains to derive or develop a novel feature beyond the PSSM profile, predicted secondary structure and solvent accessibility, along with the exploration of ML-based methods and systematic feature selection protocol, which will be helpful to improve the prediction accuracy and stimulate further development in this field. Additionally, the carbohydrate binding residues predicted from ML-based methods could be utilized to design molecular dynamics based studies. Similar approaches were suggested for studying protein–DNA interactions (Kaufman and Karypis, 2012). Such approaches may help to increase the overall carbohydrate-binding sites prediction performance.

Conclusion

Experimental identification of protein-carbohydrate complex is very difficult, expensive and often time-consuming, and, therefore, the development of computational methods is necessary to predict the carbohydrate-binding sites from the given uncharacterized protein sequence. The expeditious increase of the structure databases offers an opportunity to apply knowledge-based or molecular dynamics based approaches for prediction, but these methods offer limited utility and cannot be used if the 3D structures of proteins are unavailable. Therefore, the development of more accurate sequence-based predictors is the need of the hour.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Natural Language Processing Approaches in Bioinformatics. Protein–Protein Interaction Databases

References

- Agarwal, S., Mishra, N.K., Singh, H., Raghava, G.P., 2011. Identification of mannose interacting residues using local composition. *PLOS ONE* 6, e24039.
- Agirre, J., Davies, G.J., Wilson, K.S., Cowtan, K.D., 2017. Carbohydrate structure: The rocky road to automation. *Curr. Opin. Struct. Biol.* 44, 39–47.
- Agostino, M., Jene, C., Boyle, T., Ramsland, P.A., Yuriev, E., 2009. Molecular docking of carbohydrate ligands to antibodies: Structural validation against crystal structures. *J. Chem. Inf. Model.* 49, 2749–2760.
- Ahmad, S., Sarai, A., 2005. PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinform.* 6, 33.
- Alpaydin, E., 2014. *Introduction to Machine Learning*. MIT press.
- Altschul, S.F., Madden, T.L., Schaeffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.
- Audette, G.F., Delbaere, L.T., Xiang, J., 2003. Mapping protein: Carbohydrate interactions. *Curr. Protein Pept. Sci.* 4, 11–20.
- Bakkers, M.J., Zeng, Q., Feitsma, L.J., *et al.*, 2016. Coronavirus receptor switch explained from the stereochemistry of protein-carbohydrate interactions and a single mutation. *Proc. Natl. Acad. Sci. USA* 113, E3111–E3119.
- Banno, M., Komiya, Y., Cao, W., *et al.*, 2017. Development of a sugar-binding residue prediction system from protein sequences using support vector machine. *Comput. Biol. Chem.* 66, 36–43.
- Bastanlar, Y., Ozuysal, M., 2014. Introduction to machine learning. *Methods Mol. Biol.* 1107, 105–128.
- Biarnés, X., Ardèvol, A., Planas, A., Rovira, C., 2010. Substrate conformational changes in glycoside hydrolase catalysis. A first-principles molecular dynamics study. *Biocatal. Biotransform.* 28, 33–40.
- Bitomsky, W., Wade, R.C., 1999. Docking of glycosaminoglycans to heparin-binding proteins: Validation for aFGF, bFGF, and antithrombin and application to IL-8. *J. Am. Chem. Soc.* 121, 3004–3013.
- Brzozowska, E., Pyra, A., Miskow, M., Gorska, S., Gamian, A., 2015. C-terminal sequence determinants of T4 bacteriophage tail fiber adhesin for specific lipopolysaccharide recognition.
- Carson, M.B., Langlois, R., Lu, H., 2010. NAPS: A residue-level nucleic acid-binding prediction server. *Nucleic Acids Res.* 38, W431–W435.
- Chen, P., Huang, J.Z., Gao, X., 2014. LigandRFs: Random forest ensemble to identify ligand-binding residues from sequence information alone. *BMC Bioinform.* 15 (Suppl. 15), S4.
- Chen, Y., Shoichet, B.K., 2009. Molecular docking and ligand specificity in fragment-based inhibitor discovery. *Nat. Chem. Biol.* 5, 358–364.
- Cross, S.S., 2005. Improved FlexX docking using FlexS-determined base fragment placement. *J. Chem. Inf. Model.* 45, 993–1001.
- Dam, T.K., Brewer, C.F., 2002. Thermodynamic studies of lectin-carbohydrate interactions by isothermal titration calorimetry. *Chem. Rev.* 102, 387–430.
- De Geus, D.C., Van Roon, A.M., Thomassen, E.A., *et al.*, 2009. Characterization of a diagnostic Fab fragment binding trimeric Lewis X. *Proteins* 76, 439–447.
- De Schutter, K., Van Damme, E.J., 2015. Protein-carbohydrate interactions, and beyond. Multidisciplinary Digital Publishing Institute.
- Dejoux, A., Cieplak, P., Hannick, N., Moyna, G., Dupradeau, F.-Y., 2001. AmberFFC, a flexible program to convert AMBER and GLYCAM force fields for use with commercial molecular modeling packages. *J. Mol. Model.* 7, 422–432.
- DeMarco, M.L., Woods, R.J., 2008. Structural glycobiology: A game of snakes and ladders. *Glycobiology* 18, 426–440.
- Dharra, R., Talwar, S., Singh, Y., Gupta, R., *et al.*, 2017. Rational design of drug-like compounds targeting Mycobacterium marinum Melf protein. *PLoS ONE* 12, e0183060.
- Ding, X.M., Pan, X.Y., Xu, C., Shen, H.B., 2010. Computational prediction of DNA-protein interactions: A review. *Curr. Comput. Aided Drug Des.* 6, 197–206.
- Duverger, E., Frison, N., Roche, A.-C., Monsigny, M., 2003. Carbohydrate-lectin interactions assessed by surface plasmon resonance. *Biochimie* 85, 167–179.
- Eddy, S.R., 1998. Profile hidden Markov models. *Bioinformatics (Oxford)* 14, 755–763.
- Fadda, E., Woods, R.J., 2010. Molecular simulations of carbohydrates and protein-carbohydrate interactions: Motivation, issues and prospects. *Drug Discov. Today* 15, 596–609.
- Fariselli, P., Pazos, F., Valencia, A., Casadio, R., 2002. Prediction of protein-protein interaction sites in heterocomplexes with neural networks. *Eur. J. Biochem.* 269, 1356–1361.
- Firoz, A., Malik, A., Joplin, K.H., *et al.*, 2011. Residue propensities, discrimination and binding site prediction of adenine and guanine phosphates. *BMC Biochem.* 13 (12), 20.
- Friesner, R.A., Banks, J.L., Murphy, R.B., *et al.*, 2004. Glide: A new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.* 47, 1739–1749.
- Fukui, S., Feizi, T., Galustian, C., Lawson, A.M., Chai, W., 2002. Oligosaccharide microarrays for high-throughput detection and specificity assignments of carbohydrate-protein interactions. *Nature Biotechnol.* 20, 1011–1017.
- Gallet, X., Charlotiaux, B., Thomas, A., Brasseur, R., 2000. A fast method to predict protein interaction sites from sequences. *J. Mol. Biol.* 302, 917–926.
- Gauto, D.F., Petruk, A.A., Modenutti, C.P., *et al.*, 2013. Solvent structure improves docking prediction in lectin-carbohydrate complexes. *Glycobiology* 23, 241–258.
- Gohier, A., Espinosa, J.F., Jimenez-Barbero, J., *et al.*, 1996. Knowledge-based modeling of a legume lectin and docking of the carbohydrate ligand: The Ulex europaeus lectin I and its interaction with fucose. *J. Mol. Graph.* 14, 363–364.
- Gomez, S., Payne, A.M., Savko, M., *et al.*, 2016. Structural and functional characterization of a highly stable endo-beta-1,4-xylanase from Fusarium oxysporum and its development as an efficient immobilized biocatalyst. *Biotechnol. Biofuels* 9, 191.
- Grant, O.C., Woods, R.J., 2014. Recent advances in employing molecular modelling to determine the specificity of glycan-binding proteins. *Curr. Opin. Struct. Biol.* 28, 47–55.
- Gribskov, M., Lofth, R., Eisenberg, D., 1990. Profile analysis. *Methods Enzymol.* 183, 146–159.
- Gromiha, M.M., Veluraja, K., Fukui, K., 2014. Identification and analysis of binding site residues in protein-carbohydrate complexes using energy based approach. *Protein Pept. Lett.* 21, 799–807.
- Gunasekera, T.S., Bowen, L.L., Zhou, C.E., *et al.*, 2017. Transcriptomic analyses elucidate adaptive differences of closely related strains of pseudomonas aeruginosa in fuel. *Appl. Environ. Microbiol.* 83.
- Guvench, O., Greene, S.N., Kamath, G., *et al.*, 2008. Additive empirical force field for hexopyranose monosaccharides. *J. Comput. Chem.* 29, 2543–2564.
- Guvench, O., Hatcher, E.R., Venable, R.M., Pastor, R.W., Mackerell, A.D., 2009. CHARMM additive all-atom force field for glycosidic linkages between hexopyranoses. *J. Chem. Theory Comput.* 5, 2353–2370.
- Hansson, T., Oostenbrink, C., Van Gunsteren, W., 2002. Molecular dynamics simulations. *Curr. Opin. Struct. Biol.* 12, 190–196.
- Hatcher, E., Guvench, O., Mackerell, A.D., 2009. CHARMM additive all-atom force field for aldopentofuranoses, methyl-aldopentofuranosides, and fructofuranose. *J. Phys. Chem. B* 113, 12466–12476.
- Hill, A.D., Reilly, P.J., 2008. A Gibbs free energy correlation for automated docking of carbohydrates. *J. Comput. Chem.* 29, 1131–1141.
- Hwang, S., Gou, Z., Kuznetsov, I.B., 2007. DP-Bind: A web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics* 23, 634–636.
- Humphrey, W., Dalke, A., Schulten, K., 1996. VMD: Visual molecular dynamics. *J. Mol. Graph.* 14 (33–38), 27–28.
- Imberty, A., 1997. Oligosaccharide structures: Theory versus experiment. *Curr. Opin. Struct. Biol.* 7, 617–623.
- Jiménez-Barbero, J., Asensio, J.L., Cañada, F.J., Poveda, A., 1999. Free and protein-bound carbohydrate structures. *Curr. Opin. Struct. Biol.* 9, 549–555.
- Karlsson, K.-A., 1991. Glycobiology: A growing field for drug design. *Trends Pharmacol. Sci.* 12, 265–272.
- Kaufman, C., Karypis, G., 2012. Computational tools for protein-DNA interactions. *WIREs Data Mining Knowl. Discov.* 2, 14–28.

- Kerzmann, A., Fuhrmann, J., Kohlbacher, O., Neumann, D., 2008. BALLDock/SLICK: A new method for protein-carbohydrate docking. *J. Chem. Inf. Model.* 48, 1616–1625.
- Kerzmann, A., Neumann, D., Kohlbacher, O., 2006. SLICK – Scoring and energy functions for protein-carbohydrate interactions. *J. Chem. Inf. Model.* 46, 1635–1642.
- Khare, H., Ratnaparkhi, V., Chavan, S., Jayaraman, V., 2012. Prediction of protein-mannose binding sites using random forest. *Bioinformatics* 8, 1202–1205.
- Kirschner, K.N., Woods, R.J., 2001. Solvent interactions determine carbohydrate conformation. *Proc. Natl. Acad. Sci. USA* 98, 10541–10545.
- Kirschner, K.N., Yongye, A.B., Tschampel, S.M., *et al.*, 2008. GLYCAM06: A generalizable biomolecular force field. *Carbohydrates. J. Comput. Chem.* 29, 622–655.
- Kneller, D.G., Cohen, F.E., Langridge, R., 1990. Improvements in protein secondary structure prediction by an enhanced neural network. *J. Mol. Biol.* 214, 171–182.
- Kony, D., Damm, W., Stoll, S., Van Gunsteren, W.F., 2002. An improved OPLS-AA force field for carbohydrates. *J. Comput. Chem.* 23, 1416–1429.
- Krogh, A., Mian, I.S., Haussler, D., 1994. A hidden Markov model that finds genes in *E. coli* DNA. *Nucleic Acids Res.* 22, 4768–4778.
- Kulharia, M., Bridgett, S.J., Goody, R.S., Jackson, R.M., 2009. InCa-SiteFinder: A method for structure-based prediction of inositol and carbohydrate binding sites on proteins. *J. Mol. Graph. Model.* 28, 297–303.
- Kuznetsov, I.B., Gou, Z., Li, R., Hwang, S., 2006. Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins* 64, 19–27.
- Labonte, J., Adolf-Bryfogle, J., Schief, W.R., Gray, J.J., 2017. Residue-centric modeling and design of saccharide and glycoconjugate structures. *J. Comput. Chem.* 38, 276–287.
- Laederach, A., Reilly, P.J., 2003. Specific empirical free energy function for automated docking of carbohydrates to proteins. *J. Comput. Chem.* 24, 1748–1757.
- Laurie, A.T., Jackson, R.M., 2005. Q-SiteFinder: An energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics* 21, 1908–1916.
- Lawrence, S.L., Feil, S.C., Morton, C.J., *et al.*, 2015. Crystal structure of *Streptococcus pneumoniae* pneumolysin provides key insights into early steps of pore formation. *Sci. Rep.* 5, 14352.
- Li, B.Q., Feng, K.Y., Chen, L., Huang, T., Cai, Y.D., 2012. Prediction of protein-protein interaction sites by random forest algorithm with mRMR and IFS. *PLOS ONE* 7, e43927.
- Lins, R.D., Hunenberger, P.H., 2005. A new GROMOS force field for hexopyranose-based carbohydrates. *J. Comput. Chem.* 26, 1400–1412.
- Lis, H., Sharon, N., 1972. Soy bean (Glycine max) agglutinin. *Methods Enzymol.* 28, 360–365.
- Liu, J., Rost, B., 2004. Sequence-based prediction of protein domains. *Nucleic Acids Res.* 32, 3522–3530.
- Livingstone, C.D., Barton, G.J., 1993. Protein sequence alignments: A strategy for the hierarchical analysis of residue conservation. *Bioinformatics* 9, 745–756.
- MackKerell, A.D., Bashford, D., Bellott, M., *et al.*, 1998. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. B* 102, 3586–3616.
- MackKerell Jr., A.D., Feig, M., Brooks 3rd, C.L., 2004. Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. *J. Comput. Chem.* 25, 1400–1415.
- Malik, A., Ahmad, S., 2007. Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network. *BMC Struct. Biol.* 7, 1.
- Malik, A., Firoz, A., Jha, V., Ahmad, S., 2010. PROCARB: A database of known and modelled carbohydrate-binding protein structures with sequence-based prediction tools. *Adv. Bioinform.* 436036.
- Malik, A., Lee, J., Lee, J., 2014. Community-based network study of protein-carbohydrate interactions in plant lectins using glycan array data. *PLOS ONE* 9, e95480.
- Mallajosyula, S.S., Jo, S., Im, W., MackKerell Jr., A.D., 2015. Molecular dynamics simulations of glycoproteins using CHARMM. *Methods Mol. Biol.* 1273, 407–429.
- Manavalan, B., Basith, S., Shin, T.H., *et al.*, 2017b. MLACP: Machine-learning-based prediction of anticancer peptides. *Oncotarget* 8 (44), 77121–77136.
- Manavalan, B., Lee, J., 2017a. SVMQA: Support-vector-machine-based protein single-model quality assessment. *Bioinformatics* 33, 2496–2503.
- Manavalan, B., Lee, J., Lee, J., 2014. Random forest-based protein model quality assessment (RFMQA) using structural features and potential energy terms. *PLOS ONE* 9, e106542.
- Manavalan, B., Shin, T.H., Lee, G., 2017c. DHSpred: Support-vector-machine-based human DNase I hypersensitive sites prediction using the optimal features selected by random forest. *Oncotarget*.
- McCoy, J.P., Varani, J., Goldstein, I.J., 1983. Enzyme-linked lectin assay (ELLA): Use of alkaline phosphatase-conjugated Griffonia simplicifolia B4 isolectin for the detection of α -D-galactopyranosyl end groups. *Anal. Biochem.* 130, 437–444.
- Mishra, S.K., Adam, J., Wimmerova, M., Koca, J., 2012. In silico mutagenesis and docking study of *Ralstonia solanacearum* RSL lectin: Performance of docking software to predict saccharide binding. *J. Chem. Inf. Model.* 52, 1250–1261.
- Mishra, S.K., Kara, M., Zacharias, M., Koca, J., 2014. Enhanced conformational sampling of carbohydrates by Hamiltonian replica-exchange simulation. *Glycobiology* 24, 70–84.
- Mompean, M., Villalba, M., Bruix, M., Zamora-Carreras, H., 2017. Insights into protein-carbohydrate recognition: A novel binding mechanism for CBM family 43. *J. Mol. Graph. Model.* 73, 152–156.
- Morris, G.M., Huey, R., Olson, A.J., 2008. Using AutoDock for ligand-receptor docking. *Curr. Protoc. Bioinform.* (Chapter 8, Unit 8 14).
- Mulakala, C., Reilly, P.J., 2005. Force calculations in automated docking: Enzyme-substrate interactions in *Fusarium oxysporum* Cel7B. *Proteins* 61, 590–596.
- Nassif, H., Al-Ali, H., Khuri, S., Keirouz, W., 2009. Prediction of protein-glucose binding sites using support vector machines. *Proteins: Struct. Funct. Bioinform.* 77, 121–132.
- Ng, S., Lin, E., Kitov, P.I., *et al.*, 2015. Genetically encoded fragment-based discovery of glycopeptide ligands for carbohydrate-binding proteins. *J. Am. Chem. Soc.* 137, 5248–5251.
- Nilsson, N.J., 1996. Introduction to Machine Learning. An Early Draft of a Proposed Textbook.
- Nivedha, A.K., Makeneni, S., Foley, B.L., Tessier, M.B., Woods, R.J., 2014. Importance of ligand conformational energies in carbohydrate docking: Sorting the wheat from the chaff. *J. Comput. Chem.* 35, 526–539.
- Nivedha, A.K., Thieker, D.F., Makeneni, S., Hu, H., Woods, R.J., 2016. Vina-Carb: Improving glycosidic angles during carbohydrate docking. *J. Chem. Theory Comput.* 12, 892–901.
- Ofek, I., Hasty, D.L., Sharon, N., 2003. Anti-adhesion therapy of bacterial diseases: Prospects and problems. *FEMS Immunol. Med. Microbiol.* 38, 181–191.
- Ofran, Y., Mysore, V., Rost, B., 2007. Prediction of DNA-binding residues from sequence. *Bioinformatics* 23, i347–i353.
- Ofran, Y., Rost, B., 2007. Protein-protein interaction hotspots carved into sequences. *PLOS Comput. Biol.* 3, e119.
- Pai, P.P., Mondal, S., 2016. MOWGLI: Prediction of protein-Mannose interacting residues with ensemble classifiers using evolutionary information. *J. Biomol. Struct. Dyn.* 34, 2069–2083.
- Park, S., Lee, M.R., Shin, I., 2007. Fabrication of carbohydrate chips and their use to probe protein-carbohydrate interactions. *Nat. Protoc.* 2, 2747–2758.
- Park, Y.D., Shin, S., Panepinto, J., *et al.*, 2014. A role for LHC1 in higher order structure and complement binding of the *Cryptococcus neoformans* capsule. *PLOS Pathog.* 10, e1004037.
- Passerini, A., Punta, M., Ceroni, A., Rost, B., Frasconi, P., 2006. Identifying cysteines and histidines in transition-metal-binding sites using support vector machines and neural networks. *Proteins* 65, 305–316.
- Perez, S., Tvaroska, I., 2014. Carbohydrate-protein interactions: Molecular modeling insights. *Adv. Carbohydr. Chem. Biochem.* 71, 9–136.
- Pol-Fachin, L., 2017. Insights into the effects of glycosylation and the monosaccharide-binding activity of the plant lectin CrataBL. *Glycoconj. J.* 34, 515–522.
- Puvirajesinghe, T.M., Turnbull, J.E., 2016. Glycoarray technologies: Deciphering interactions from proteins to live cell responses. *Microarrays (Basel)* 5.
- Quiocho, F.A., 1989. Protein-carbohydrate interactions: Basic molecular features. *Pure Appl. Chem.* 61, 1293–1306.
- Raju, R.K., Ramraj, A., Hillier, I.H., Vincent, M.A., Burton, N.A., 2009. Carbohydrate-aromatic pi interactions: A test of density functionals and the DFT-D method. *Phys. Chem. Chem. Phys.* 11, 3411–3416.
- Reina, J.J., Diaz, I., Nieto, P.M., *et al.*, 2008. Docking, synthesis, and NMR studies of mannosyl trisaccharide ligands for DC-SIGN lectin. *Org. Biomol. Chem.* 6, 2743–2754.

- Samsonov, S.A., Teyra, J., Pisabarro, M.T., 2011. Docking glycosaminoglycans to proteins: Analysis of solvent inclusion. *J. Comput. Aided Mol. Des.* 25, 477–489.
- Shewell, L.K., Harvey, R.M., Higgins, M.A., *et al.*, 2014. The cholesterol-dependent cytolysins pneumolysin and streptolysin O require binding to red blood cell glycans for hemolytic activity. *Proc. Natl. Acad. Sci. USA* 111, E5312–E5320.
- Shionyu-Mitsuyama, C., Shirai, T., Ishida, H., Yamane, T., 2003. An empirical approach for structure-based prediction of carbohydrate-binding sites on proteins. *Protein Eng.* 16, 467–478.
- Shu, N., Zhou, T., Hovmöller, S., 2008. Prediction of zinc-binding sites in proteins from sequence. *Bioinformatics* 24, 775–782.
- Spiwok, V., Lipovova, P., Skalova, T., *et al.*, 2005. Modelling of carbohydrate-aromatic interactions: Ab initio energetics and force field performance. *J. Comput. Aided Mol. Des.* 19, 887–901.
- Spiwok, V., 2017. CH/π interactions in carbohydrate recognition. *Molecules* 22.
- Sujatha, M.S., Balaji, P.V., 2004. Identification of common structural features of binding sites in galactose-specific proteins. *Proteins* 55, 44–65.
- Suresh, M.X., Gromiha, M.M., Suwa, M., 2015. Development of a machine learning method to predict membrane protein-ligand binding residues using basic sequence information. *Adv. Bioinform.* 2015, 843030.
- Taherzadeh, G., Zhou, Y., Liew, A.W., Yang, Y., 2016. Sequence-based prediction of protein-carbohydrate binding sites using support vector machines. *J. Chem. Inf. Model.* 56, 2115–2122.
- Takaoka, T., Mori, K., Okimoto, N., Neya, S., Hoshino, T., 2007. Prediction of the structure of complexes comprised of proteins and glycosaminoglycans using docking simulation and cluster analysis. *J. Chem. Theory Comput.* 3, 2347–2356.
- Taroni, C., Jones, S., Thornton, J.M., 2000. Analysis and prediction of carbohydrate binding sites. *Protein Eng.* 13, 89–98.
- Tessier, M.B., Demarco, M.L., Yongye, A.B., Woods, R.J., 2008. Extension of the GLYCAM06 biomolecular force field to lipids, lipid bilayers and glycolipids. *Mol. Simul.* 34, 349–363.
- Trott, O., Olson, A.J., 2010. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* 31, 455–461.
- Tsai, K.C., Jian, J.W., Yang, E.W., *et al.*, 2012. Prediction of carbohydrate binding sites on protein surfaces with 3-dimensional probability density distributions of interacting atoms. *PLOS ONE* 7, e40846.
- Vandenbussche, S., Diaz, D., Fernandez-Alonso, M.C., *et al.*, 2008. Aromatic-carbohydrate interactions: An NMR and computational study of model systems. *Chemistry* 14, 7570–7578.
- Verdonk, M.L., Cole, J.C., Hartshorn, M.J., Murray, C.W., Taylor, R.D., 2003. Improved protein-ligand docking using GOLD. *Proteins* 52, 609–623.
- Vital De Oliveira, O., 2014. Molecular dynamics and metadynamics simulations of the cellulase Cel48F. *Enzyme Res.* 2014, 692738.
- Walker, A.R., Bonomi, R., Popov, V., Gelovani, J.G., Andres Cisneros, G., 2017. Investigating carbohydrate based ligands for galectin-3 with docking and molecular dynamics studies. *J. Mol. Graph. Model.* 71, 211–217.
- Wang, D., Liu, S., Trummer, B.J., Deng, C., Wang, A., 2002. Carbohydrate microarrays for the recognition of cross-reactive molecular markers of microbes and host cells. *Nat. Biotechnol.* 20, 275–281.
- Wang, L., Brown, S.J., 2006. BindN: A web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences. *Nucleic Acids Res.* 34, W243–W248.
- Xiong, X., Chen, Z., Cossins, B.P., *et al.*, 2015. Force fields and scoring functions for carbohydrate simulation. *Carbohydr. Res.* 401, 73–81.
- Yang, Y., Zhan, J., Zhao, H., Zhou, Y., 2012. A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. *Proteins* 80, 2080–2088.
- Yang, Y., Zhou, Y., 2008. Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions. *Protein Sci.* 17, 1212–1219.
- Zhang, J., Kitova, E.N., Li, J., *et al.*, 2016. Localizing carbohydrate binding sites in proteins using hydrogen/deuterium exchange mass spectrometry. *J. Am. Soc. Mass Spectrom.* 27, 83–90.
- Zhao, H., Yang, Y., Von Itzstein, M., Zhou, Y., 2014. Carbohydrate-binding protein identification by coupling structural similarity searching with binding affinity prediction. *J. Comput. Chem.* 35, 2177–2183.
- Zhou, H., Zhou, Y., 2002. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.* 11, 2714–2726.
- Zhou, H.X., Shan, Y., 2001. Prediction of protein interaction sites from sequence profile and residue neighbor list. *Proteins* 44, 336–343.
- Zvelebil, M.J., Barton, G.J., Taylor, W.R., Sternberg, M.J., 1987. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J. Mol. Biol.* 195, 957–961.

Relevant Websites

www.ball-project.org
 BALL Project.
<http://www.procarb.org/procarbdb/>
 PROCARB HOME.
<https://zenodo.org/record/61513#.WcB-4hdLeis>
 SBRP: sbrp | Zenodo.
<https://sites.google.com/site/sukantamondal/software>
 Software – Sukanta MONDAL – Google Sites.

Computational Prediction of Nucleic Acid Binding Residues From Sequence

Nur SA Ghani and Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Bangi, Malaysia
Shandar Ahmad, Jawaharlal Nehru University, New Delhi, India

© 2019 Elsevier Inc. All rights reserved.

Introduction and Background

Protein–nucleic acid interactions play key roles in many biological processes. In a large group of such events, the binding of the proteins to their respective nucleic acid counterpart affects some regulatory outcome such as transcriptional and translational regulation and protein synthesis. Activation of an immune system in response to pathogen invasion is another example in which specific nucleic acid compositions of pathogens are recognized as the first step in a complex biological pathway. Approximately 21% of proteins are denoted as nucleic acid binding, based on UniprotKB/SwissProt database (Boutet *et al.*, 2007). Nucleic acid binding proteins are mostly composed of at least one DNA/RNA-binding domain where the interfacing with amino acids take place in specific or nonspecific manner.

DNA-binding proteins (DBPs), such as transcription factors, DNA polymerases, DNA ligases, some Toll like receptors, and methyltransferases can be represented by various DNA-binding domains carrying different molecular functions. Common DNA-binding domains include the zinc finger, helix-turn-helix, leucine zipper, and helix-loop-helix domains. On the other hand, many RNA-binding proteins are important for posttranscriptional regulation (Glisovic *et al.*, 2008) and the regulation of gene expression (Ray *et al.*, 2013). One or more RNA-binding domains facilitate the formation of protein–RNA interfaces. This includes RNA recognition motifs such as the K Homology motif, RGG box motif, and DEAD/DEAH box domain (Lunde *et al.*, 2007).

While proteins adopt different mechanisms for both DNA- and RNA-binding activity, general methods of detecting binding residues, before the crystal structure of a complex is solved, are useful to comprehend the molecular recognition capacity and interface mechanics and hence designing molecular interventions. Experimental methods like X-ray crystallography and nuclear magnetic resonance have been used to determine the structures of numerous protein–nucleic acid complexes and these structures have greatly contributed to the understanding of what constitutes as protein–nucleic acid interfaces and favorable modes of interactions in the structural context.

Such experimental methods lead to the steady growth in the numbers of protein–nucleic acid complexes deposited into the Protein Data Bank (Rose *et al.*, 2017). As of September 2017, the database contains entries for 4317 and 2290 protein–DNA and protein–RNA complexes, respectively. Experimental methods of determining protein–nucleic acid interactions are time-consuming and expensive. Furthermore, experimental approaches will not be able to cope with the ever-increasing number of protein sequences requiring annotation for their potential nucleic acid binding ability. It is also not possible to solve the crystal structure of every homolog for every NA-binding protein, even though small sequence-level changes may significantly alter the interaction dynamics. Hence, computational platforms for prediction of nucleic acid binding sites from sequence can provide an alternative that can be just as reliable to analyze protein–nucleic acid interactions.

Over the years, a number of computational tools have been developed for the prediction of DNA/RNA-binding residues in sequence data. Here, we survey a selection of such tools that are available as web services. Our focus in this article is to compare web servers in terms of: (i) the data they use as the input during training and subsequent use by researchers; (ii) the engine that drives their predictions; (iii) the kind of output they generate; and (iv) the performance levels claimed during their publications, some of which we have benchmarked for a quick estimate of their relative advantage. Such reviews are also available in the literature as part of other studies but with a growing number of new tools and perspectives that can be developed, we feel that an overview in a nutshell for reference purposes is needed. We have structured our discussion into four parts. First, we take a look at the user interaction of the tools for binding site predictions. Particularly, we discuss the user inputs and output formats in each of the available tools. Subsequently, we look at the techniques used in developing these prediction methods, including the datasets and the features for data driven models. Then, we describe the performance scores and evaluation strategies of binding site prediction tools and discuss the challenges therein. Finally, we present some case studies and discuss how users can get the best out of multiple prediction tools available over the web.

Computational Tools for Nucleic Acid Binding Prediction: The User Perspective

Detection of amino acids that interact with nucleic acids from protein sequence can be most useful as well as challenging when no structural information is available. A number of computational platforms have been developed for predicting DNA-binding residues from sequence, such as DP-Bind (Hwang *et al.*, 2007), DQPred-DBR (Chai *et al.*, 2016), TargetDNA (Hu *et al.*, 2016), and DRNAPred (Yan and Kurgan, 2017). Some of the methods to predict RNA-binding site/residues from protein sequence include DRNAPred (Yan and Kurgan, 2017), AaRNA (Li *et al.*, 2014), PRIdictor (Tuvshinjargal *et al.*, 2016), RNABindRPlus (Walia *et al.*, 2014), RNABindR (Walia *et al.*, 2012), and SRCPred (Fernandez *et al.*, 2011). AaRNA and SRCPred are also able to predict dinucleotide contacts. These methods allow for the rapid detection of amino acids that are thought to be interacting with nucleic acids. In some cases, users are interested in specific DNA or RNA-binding predictions and the use of different tools may provide overlapping predictions. Some tools are developed with this deconvolutionary objective in mind. Table 1 summarizes the list of selected tools for the prediction together with their descriptions while Table 2 provides a comparison of the features and capability of each service.

Server Inputs and Formats

Most prediction methods require a protein sequence in the FASTA format or single code amino acid sequence as the query by either pasting them or uploading the file(s). The ability of several programs – DRNAPred, RNABindRPlus and RNABindR to rapidly predict DNA-binding residues for a batch of protein sequences provide them an advantage in contrast with other prediction methods. Amino acid sequences of desired proteins can be culled from protein chains in the Protein Data Bank (Rose *et al.*, 2017) or protein sequence databases like NCBI RefSeq (Pruitt *et al.*, 2007) and SwissProt (Bairoch and Apweiler, 2000). In an unlikely scenario, when users only have a PDB-formatted protein structure, various programming scripts and PDB-related web servers can be used to extract amino acid sequence from SEQRES records found in the PDB file, such as ccPDB (Singh *et al.*, 2012), PDBFINDER (Touw *et al.*, 2015) and PISCES (Wang and Dunbrack, 2003) web servers.

Output and Average Time for Computations

The time for the computation to be completed is often influenced by the list of queries submitted to the server at a given time. A prediction conducted for a single protein sequence could take a few minutes on average to be completed, where the output is either submitted to an email address or displayed on the web server directly upon job completion. The predictions conducted by these web servers mostly return binding propensities, with the predicted probability values and binary label for each amino acid in the sequence, commonly termed as “0” for nonbinding or “1” for binding.

Methodologies Behind Binding Site Prediction Tools

Training Datasets

Protein–nucleic acid contact data

Protein structure datasets play a crucial role in the development of prediction methods for protein–nucleic acid interfaces. Methods that predict nucleic acid binding residues are commonly trained and tested upon a set of nucleic acid binding residues with a distinct definition of contacts. The contact data are derived from proteins known to bind to nucleic acids, either from protein–nucleic acid complexes or annotated DNA/RNA-binding proteins. Protein structures are retrieved from the Protein Data Bank (Rose *et al.*, 2017) and certain filters are applied before training them. In most cases, refinements are carried out based on (1) structure quality, that is, X-ray crystallography determined structures at high resolution; and (2) sequence identity with distinct cut-off values to generate nonredundant structures/folds. The datasets generated by each prediction tool might also differ from one another according to different exclusion criteria such as exclusion of proteins with inadequate residue lengths (Li *et al.*, 2014; Walia *et al.*, 2012), and the removal of residues with missing coordinates (Hwang *et al.*, 2007; Yan and Kurgan, 2017).

Different annotations of protein–nucleic acid interactions at the residue level have been used to generate a sample pool containing binding residues for the training of prediction methods. The binding interfaces are defined by the maximal distance between amino acids to any atom from the nucleic acid. One often-used definition for a binding residue is that any amino acid

Table 1 Computational programs for sequence-based DNA/RNA-binding residue prediction

Prediction	Service/ program	Description
DBR	DP-Bind	A sequence-based prediction method using three different machine learning methods to generate a consensus prediction of DNA-binding residues (Hwang <i>et al.</i> , 2007); (http://lcf.rit.albany.edu/dp-bind/)
	DQPred-DBR	An evolution-based prediction method using dynamic model to detect DNA-binding residues (Chai <i>et al.</i> , 2016) (http://www.inforstation.com/webserver/DQPred-DBR/predict.html)
	TargetDNA	A SVM-based method using weighted evolutionary profiles to target protein–DNA binding residues (Hu <i>et al.</i> , 2016) (http://csbio.njust.edu.cn:8080/TargetDNA/)
DBR/RBR	DRNAPred	A combined prediction method for DNA- and RNA-binding residues using sequence and structural features based on protein sequence (Yan and Kurgan, 2017) (http://biomine.cs.vcu.edu/servers/DRNAPred/)
RBR	AaRNA	A sequence and structure-based prediction method for RNA-binding propensity using homology modeling and neural networks (Li <i>et al.</i> , 2004) (https://sysimm.ifrec.osaka-u.ac.jp/aarna/)
	PRIdictor	A sequence-based method to predict protein–RNA interfaces from RNA and protein sequences according to global and local features (Tuvshinjargal <i>et al.</i> , 2016) (http://bclab.inha.ac.kr/pridictor/pridictor.html)
	RNABindRPlus	A combined homology and SVM-based prediction method using evolutionary profile to locate RNA-binding residues (Walia <i>et al.</i> , 2014) (http://ailab1.ist.psu.edu/RNABindRPlus/)
	RNABindR	A protein–RNA interface prediction tool that combine naïve Bayes and SVM methods using both sequence and structural features from protein sequence (Walia <i>et al.</i> , 2012) (http://ailab1.ist.psu.edu/RNABindR/)
	SRCPred	A PSSM-based prediction tool for RNA dinucleotide contacts in protein sequence using neural networks (Fernandez <i>et al.</i> , 2011) (http://ccbb.jnu.ac.in/shandar/servers/srcpred/)

Source: A listing of nine web servers aimed for the prediction of nucleic acid binding residue using sequence-based methods.

Table 2 Comparison of different computational programs for prediction of nucleic acid binding residues

	Program/service name								SRCpred
	DP-Bind	DQPred-DBR	TargetDNA	DRNAPred	AaRNA	PRIdictor	RNABindPlus	RNABindR	
Input query									
FASTA-formatted/one letter code protein sequence	✓	✓	✓	✓	✓	✓	✓	✓	✓
Nucleic acid sequence						✓			
Allows multiple sequences				✓			✓	✓	
PDB structure					✓				
User-defined method/threshold value	✓	✓	✓				✓		
Prediction approach									
Dynamic model		✓							
Support vector machine (SVM)	✓	✓	✓			✓	✓	✓	
Logistic regression				✓			✓		
Naïve Bayes								✓	
Neural networks					✓				✓
Hidden Markov model				✓	✓				
Homology-based method		✓					✓		
Training datasets									
Protein Data Bank (PDB)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Protein sequences	✓			✓					
Type of sequence-based prediction									
Predict DBS/DBR	✓	✓	✓	✓					
Predict RBS/RBR				✓	✓	✓	✓	✓	✓
Dinucleotide contacts					✓				✓
Use sequence features	✓	✓		✓	✓	✓	✓	✓	✓
Use structural features	✓			✓		✓		✓	✓
Sequence-derived features									
<i>Sequence features</i>									
PSSM-based evolutionary conservation score	✓	✓	✓				✓	✓	✓
Non-PSSM-based evolutionary conservation score	✓			✓	✓		✓	✓	
Sequence neighborhood				✓	✓			✓	✓
Physicochemical and biochemical properties				✓	✓	✓			✓
<i>Structural features</i>									
Spatial neighborhood	✓							✓	
Interaction propensity						✓			
Solvent accessibility	✓		✓	✓		✓			✓
Predicted secondary structure	✓			✓					
Predicted intrinsic disorder				✓					
Performance measures									
Accuracy	✓	✓	✓			✓			
Area under the ROC curve (AUC)		✓		✓	✓				✓
F-measure					✓		✓	✓	
Matthews correlation coefficient (MCC)		✓	✓	✓	✓	✓	✓	✓	
Precision			✓		✓				
Sensitivity	✓	✓	✓	✓	✓	✓	✓	✓	✓
Specificity	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Output</i>									
Prediction score	✓	✓	✓	✓	✓	✓	✓	✓	✓
Binary labeled amino acid	✓	✓	✓	✓	✓	✓	✓	✓	✓
A set of homologous proteins						✓	✓		
Downloadable output file	✓	✓	✓	✓	✓	✓	✓	✓	✓

Source: A comparison of nine web servers for nucleic acid binding residue prediction in terms of their input, prediction approach, data set, and output.

residue with a distance of less than 3.5 Å to any nucleic acid atom is treated to be in the interface. This definition has been used by DBSPred, SDCPred, DBS-PSSM, DRNAPred, AaRNA, and SRCPred web servers (Ahmad *et al.*, 2004; Andrabi *et al.*, 2009; Ahmad and Sarai, 2005; Yan and Kurgan, 2017; Li *et al.*, 2014; Fernandez *et al.*, 2011) with Fernandez *et al.* (2011) additionally integrating the annotation of RNA-binding motifs retrieved from the SCOR database (Hubbard and Thornton, 1993). Tuvshinjargal *et al.* (2016) (PRIdictor), on the other hand, defined protein–nucleic acid interaction in a more specific way that involved three different types of intermolecular forces including hydrogen bonds, hydrophobic forces and water-mediated contacts – As retrieved from Nucleic acid–Protein Interaction Database (Kirsanov *et al.*, 2013).

Protein sequences

Most prediction methods rely on experimentally determined crystal structures of protein–nucleic acid complexes. Such high resolution datasets provide binding or nonbinding information at a single residue level. On the other hand long stretches of amino acids are annotated as NA-binding in sequence datasets, which is useful in discriminating binding domains. Some NA-binding site prediction methods have also used such annotations to evaluate the prediction at the proteome scale. Examples of this are the use of nonredundant protein sequences from NCBI in the work of Hwang *et al.* (2007) and human protein sequences from UniProt database and several curated databases as adopted by Yan and Kurgan (2017). Programs that implement a homology-based method like DQPred-DBR and RNABindPlus generate their training sets of known binding residues obtained from sequence homologs of the query proteins searched against the Protein–RNA Interface Database (PRIDB) (Lewis *et al.*, 2011) and SwissProt (Bairoch and Apweiler, 2000), respectively.

Prediction Algorithms

Machine learning methods

Machine learning methods are commonly employed on other problems of sequence-based structure and function predictions in proteins (Al-Shahib *et al.*, 2007), for example, secondary structure prediction (Wang *et al.*, 2016) and prediction of interfaces for protein–protein complexes (Sun *et al.*, 2017). These methods have also been adapted to predict protein–nucleic acid interactions (Hwang *et al.*, 2007; Chai *et al.*, 2016; Hu *et al.*, 2016). Prediction models based on these machine learning methods are trained and tested with benchmark datasets of known class labels (binding or nonbinding) and a variety of prediction features usually derived from sliding windows over the whole sequence. These methods include neural networks (Li *et al.*, 2014; Ahmad and Sarai, 2005; Ahmad *et al.*, 2004; Fernandez *et al.*, 2011), support vector machines (SVMs) (Hwang *et al.*, 2007; Chai *et al.*, 2016; Hu *et al.*, 2016; Tuvshinjargal *et al.*, 2016; Walia *et al.*, 2014, 2012), naïve Bayes (Walia *et al.*, 2012); random forest (Ma *et al.*, 2011), logistic regression models (Yan and Kurgan, 2017; Walia *et al.*, 2014), and hidden Markov models (Yan and Kurgan, 2017; Li *et al.*, 2014).

Artificial neural network (ANN) models typically contain three *layers*- the input, hidden, and output layers - where each layer is made up of a designed number of *nodes* according to the features used for prediction (Lancashire *et al.*, 2009). SVM is another popular technique adopted to predict nucleic acid binding residues, where it was shown to outperform other machine learning methods apparently due to its ability to develop more robust models. An SVM-based prediction model is based on trained parameters of a kernel, which assigns class labels to new data. In the case of binding residue prediction, the prediction model is trained and tested with sets of annotated binding residues (positive class samples) and/or nonbinding residues (negative class samples), where they learn to classify objects from known data and compare with expected output in a supervised manner.

Most prediction methods on the web receive a set of amino acids in the form of a vector as the input, yield a binding propensity for individual residue in the sequence that indicates probability of target residue to be in a particular class, and at given thresholds generate a binary label for each residue.

Alignment-based methods

Alignment-based methods can be useful to predict protein–nucleic acid interactions when similar NA-binding sequences are available for the query sequence, since binding residues are often conserved among similar proteins. As an example of such approaches, Walia *et al.* (2014) combined HomPRIP, a homology-based method and SVM to predict RNA-binding residues in protein sequences. They defined the conservation of residues as an interface conservation (IC) score, that is, the degree of conservation of residues in the query protein as compared to its sequence homologs. These scores were derived from pairwise sequence alignments. The program first searches for sequence homologs using BLAST against PRIDB, calculates an IC score for each sequence homolog, and later assigns putative binding residues based on the known binding residues annotated in homologs. Another prediction program, DQPred-DBR, generates a dynamic sample to train the SVM-based prediction model, which includes a set of binding residues derived from sequence homologs of the query protein, searched against the SwissProt database. Alignment-based methods are intuitive and powerful but are greatly dependent on similarity thresholds. Lower similarity thresholds result in a lesser number of initial samples for prediction. When no homologs can be detected for the query protein, there is no choice but to use a method developed using a sliding window approach, employed in almost all the other methods described in this article.

Discriminative Features in Data Driven Models

A combination of features obtained from protein sequences are used to train and evaluate the performance of prediction models. These features include evolutionary information, sequence features within a window, structural features, and physical or biochemical features. Commonly used features for protein–nucleic acid interactions that are extracted from sequence only information include evolutionary information from PSSM, neighboring residues, accessible surface area (ASA), and secondary structure prediction.

Sequence features

Evolutionary conservation

Evolutionary conservation in the form of position specific scoring matrices (PSSMs) essentially indicates the probability of each residue to be conserved in a specific position of the sequence (Ahmad and Sarai, 2005). However, it goes beyond simple conservation as it also accounts for specific substitution patterns, which provides it with clues to characterize each position. These features have been used with the assumption that most binding sites are conserved to some extent. The substitution profiles of specific NA-binding residues should reflect a particular trend so as not to lose the binding capacity and this pattern can be captured by a machine learning model such as a neural network. Thus, PSSM is perhaps the most popular representation of sequence information for predicting binding sites. A large number of prediction programs use evolutionary information in the form of PSSM, including DP-Bind, DQPred-DBR, TargetDNA, RNABindPlus, RNABindR, and SRCPred.

PSSM profiles are generally obtained from PSI-BLAST searches of the query protein against a sequence database and a Nx20 matrix is obtained representing the substitution probability of individual residues to all other amino acid types in a given length of the sequence. Ahmad and Sarai (2005) proved that evolutionary information from PSSM is a powerful feature to discriminate binding and nonbinding residues and this was similarly suggested by Hwang *et al.* (2007) where the accuracy of prediction increased when using PSSM profiles alone or when combined with structural features.

Sequence neighborhood

Residue nearest neighbors can influence prediction, with the assumption that a network of residues is required for the protein to bind to nucleic acid residues (Yan and Kurgan, 2017; Miao and Westhof, 2015). Thus, residues with high propensity sequence neighbors would be more likely to bind to nucleic acids. Some prediction methods include sequential neighbors of target residues as one of the features for prediction using sliding windows of size $2N + 1$ where N represents nearest neighbors on either side (Walia *et al.*, 2012). Yan and Kurgan (2017) defined nearest neighbors using a sliding window of size 3 (one neighbor on each side); Li *et al.* (2014) used a sliding window of size 5 (2 contiguous residues on each side); Walia *et al.* (2012) used a sliding window of 25 for 12 sequential residues on each side; and Fernandez *et al.* (2011) used varying sizes of sliding windows for 1–8 sequence neighbors.

Structural features

The use of structural features such as solvent accessibility and secondary structure can also assist in the prediction of protein–nucleic acid interactions. Such low resolution structure information can either be derived from a solved protein structure or taken from another sequence-based prediction program, as discussed below.

(1) Solvent accessibility

A protein's binding activities typically occur via their exposed surfaces. Binding residues are more likely to reside in the exposed area, rather than in buried regions. Some methods include predicted solvent accessibility as one of the features used in the prediction (Yan and Kurgan, 2017; Hwang *et al.*, 2007; Tuvshinjargal *et al.*, 2016; Hu *et al.*, 2016). TargetDNA uses SANN (Joo *et al.*, 2012) to predict solvent accessibility using sequence neighborhood and classify each residue into three classes – Buried, intermediate, and exposed residue; DPbind uses information on accessible surface area derived from a DSSP profile (Kabsch and Sander, 1983) in which higher ASA values represent a higher degree of exposure for target residues, and residues below a defined threshold value are annotated as “surface accessible residue”; SRCPred uses ASA information from NACCESS (Hubbard and Thornton, 1993); DRNAPred combines several prediction tools to predict solvent accessibility from sequence using PROFphd (Rost and Sander, 1994), NETASA (Ahmad and Gromiha, 2002), and RVP-net (Ahmad *et al.*, 2003).

(2) Secondary structure

Secondary structure preferences for DNA/RNA-binding sites may also influence the prediction. The DSSP program is able to calculate hydrogen bonds and label predicted secondary structure elements for each residue, including beta sheets, alpha helices, and turns. A PSI-blast based secondary structure prediction method, PSIPRED (McGuffin *et al.*, 2000), is used by DRNAPred to assist the prediction. The program adopts neural networks to predict secondary structure from protein sequence based on the evolutionary conservation profile.

Evaluating Confidence Levels of Predictions

Accuracy, Sensitivity, and Specificity

A number of measures, such as accuracy, specificity, F-measure, MCC, receiving operating characteristics (ROC) curve, and the area under the ROC curve (AUROC) have been used to evaluate the performance of prediction models for making correct classification

of unseen data during testing (Table 3). The raw data generated from predictions are presented in the confusion matrix (Sokolova *et al.*, 2006), which also encode values for true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Values in the confusion matrix are used to calculate the performance measures like accuracy, sensitivity, and specificity, which are some good measures used to evaluate the correctness of the prediction, whether binding residue is correctly identified as such. Accuracy measures the proportion of correctly classified samples against all samples. This is generally not a good measure if the class labels are not equally distributed as in the NA-binding site data. A high accuracy may simply indicate that more residues are labeled in the more populated class (nonbinding). Sensitivity ($TP/(TP + FN)$) measures the true positive rate by calculating true positive samples against the total number of false negative and true positive samples, that is, the proportion of binding residues that are correctly identified. A highly sensitive model for NA-binding sites leads to many residues wrongly labeled to be binding. Specificity, on the other hand, measures the true negative rate by calculating true negative against the total number of true negative and false positive samples that is, the proportion of nonbinding residues that are predicted as such. The F-measure is a harmonic mean of sensitivity and specificity, and MCC can provide a more balanced performance estimate of a prediction model. A comparison of the prediction capability of each tool in Table 1 is provided in Table 3.

Area Under the ROC Curve

Sensitivity and specificity of prediction models can typically be adjusted by a threshold as the predictions are usually on a continuous scale between 0 and 1. To gain a model performance estimate over its full range of predicted scores, the ROC is used. ROC curves can be built between various combinations such as precision recall (P-R curve) or as a plot of true positive rate (sensitivity) represented on the y-axis and true negative rate (1-specificity) on the x-axis, indicating sensitivity/specificity pairs at different threshold values (Sokolova *et al.*, 2006; Fawcett, 2006). AUROC curve can be used to measure accuracy based on a ROC curve. AUROC quantifies to what extent a classifier can make accurate estimates of data labels at various thresholds (Fawcett, 2006).

It may be noted that the prediction scores from different methods may not be directly comparable to each other due to distinct datasets and cross-validation strategies used by individual methods. Therefore, when choosing a method for a new dataset, careful understanding of reported performance levels may be needed.

Illustrative Example(s) or Case Studies

To demonstrate the utility of web servers in identifying DNA- and RNA-binding residues, we have tested a selection of them using individual query sequences derived from protein–nucleic acid complexes – A human transcription factor in complex with double-stranded DNA (PDBID: 5fd3, Chain B) (Marceau *et al.*, 2016) and a human E3 ubiquitin ligase bound to an RNA molecule (PDBID: 5www, Chain A) (Yang *et al.*, 2017). It is clear that most prediction servers can estimate the interface residues in the test protein sequences with high accuracy with small differences in performance.

Prediction of DNA-Binding Residues

The crystal structure of DNA-binding protein Lin54 bound to its DNA promoter (PDBID: 5fd3) was solved with the residues involved in DNA recognition for the cell cycle genes homology region sequence clearly characterized (Marceau *et al.*, 2016). A

Table 3 Evaluation of the prediction of binding residues using different prediction methods

Prediction	TP	FP	TN	FN	SN	SP	ACC	F-m	MCC
DNA-binding residue prediction									
DP-Bind	12	23	115	4	0.75	0.83	0.83	0.47	0.42
TargetDNA	11	8	111	5	0.69	0.93	0.90	0.63	0.58
DRNAPred	14	36	83	2	0.88	0.70	0.72	0.42	0.38
DQPred-DBR	14	32	87	2	0.88	0.73	0.75	0.45	0.41
RNA-binding residue prediction									
AaRNA	5	1	73	15	0.25	0.99	0.83	0.38	0.40
DRNAPred	3	15	59	17	0.15	0.80	0.66	0.16	– 0.05
PRIdictor	2	2	72	18	0.10	0.97	0.79	0.17	0.15
RNABindR	8	36	38	12	0.40	0.51	0.49	0.25	– 0.07
RNABindRPlus	3	3	71	17	0.15	0.96	0.79	0.23	0.18
SRCPred	7	21	53	13	0.35	0.72	0.64	0.29	0.06

Source: Prediction of DNA- and RNA-binding residue using different prediction methods – DP-Bind (annotation propensities of ≥ 0.5 for major consensus prediction); TargetDNA (FPR $\approx 5\%$, annotation propensities of ≥ 0.05); DRNAPred (propensities > 0.4727 and > 0.1493 for DNA- and RNA-binding residue annotation); DQPred-DBR (similarity threshold of 0.7 and prediction threshold of 0.5, annotation propensities of ≥ 0.5); AaRNA (annotation propensities of ≥ 0.5); PRIdictor (annotation propensities of ≥ 0.5); RNABindR (annotation propensities of ≥ 0.5); RNABindRPlus (annotation propensities of ≥ 0.5); and SRCPred (annotation propensities of ≥ 0.05 for at least one dinucleotide contact). TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative, SN = Sensitivity, SP = Specificity, ACC = Accuracy, F-m = F-measure, MCC = Matthews Correlation Coefficient. Highest value for each column is highlighted in bold.

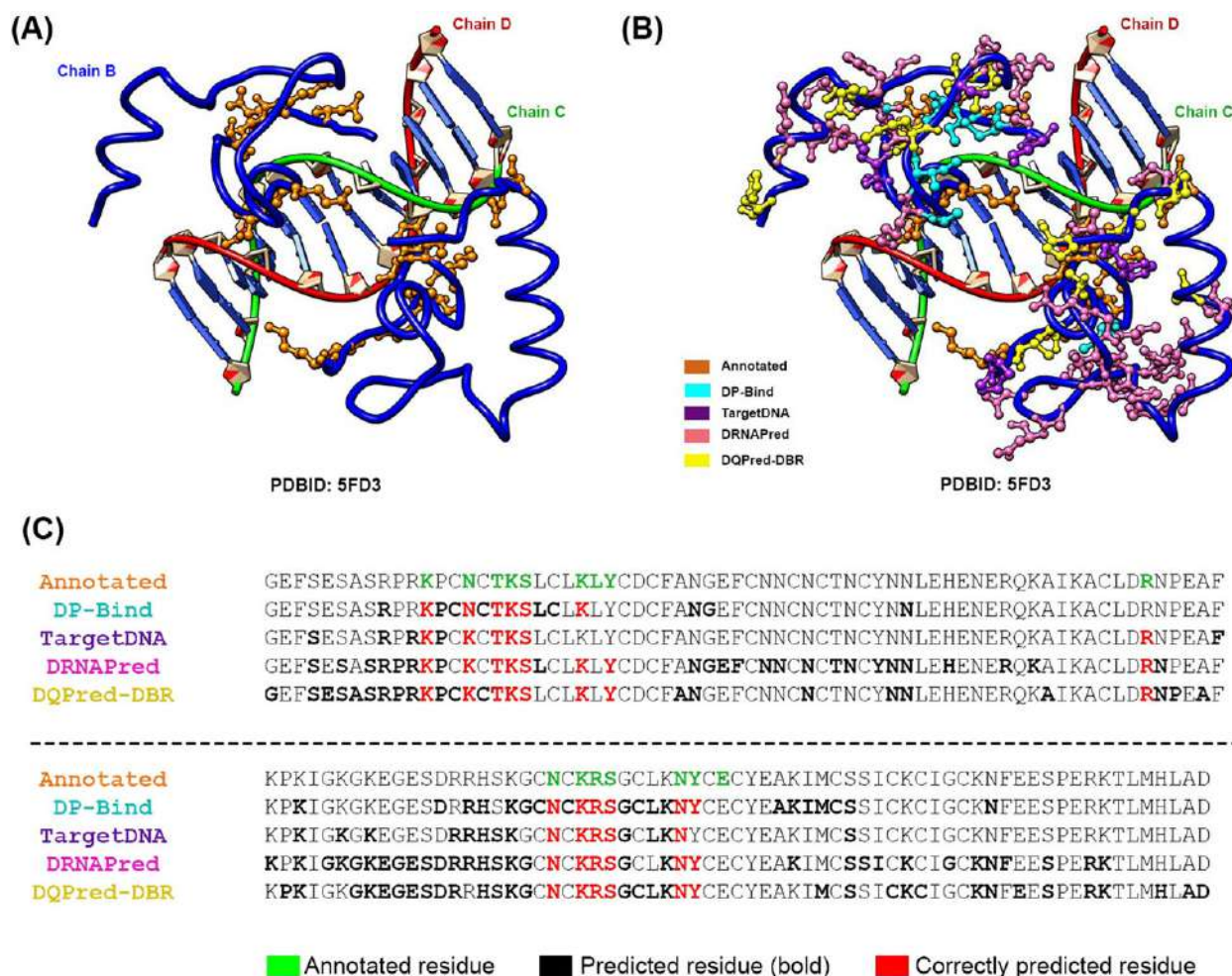


Fig. 1 Prediction of DNA-binding residues for a human transcription factor sequence. (A) A crystal structure of Lin54 protein containing tesmin domain (Chain B) bound to double-stranded DNA (Chain C and D) (PDBID: 5FD3), with annotated binding residues highlighted in orange color. (B) Predicted binding residues from different prediction tools depicted in the structure. Binding residues are shown in ball and stick representation. (C) Prediction of binding residues from the query sequence containing 134 amino acids. Annotated binding residues is colored in green, predicted binding residue in bold letter, and residue that is correctly predicted as binding is colored in red.

BLAST search against the PDB yielded no other structure with a similar sequence, while searching against the UniProtKB/SwissProt database retrieved similarities to tesmin and cysteine-rich (CXC) domain-containing proteins. Sixteen of the 135 residues in Lin54 are involved in protein–DNA contacts through hydrogen bonds and van der Waals interactions. All tested prediction tools correctly predicted at least 10 residues out of the 16 annotated binding residues (Fig. 1).

Prediction of RNA-Binding Residues

KH domains are known to bind RNA and are important for posttranscriptional activities. The KH-RNA binding interfaces could be observed in the structure of a KH1 domain of human RNA-binding E3 ubiquitin-protein ligase MEX-3C complex with RNA (PDB ID: 5www). Fig. 2 illustrates the annotations derived from the experimental data (Fig. 2(B)) and the comparison between predicted binding residues from different prediction tools (Fig. 2(C–H)). Results indicate lower performance of most prediction methods for RNA-binding residue prediction compared to methods for DNA-binding residue prediction. All methods correctly predicted at least two binding residues from 20 annotated binding residues.

NA-binding residues are usually conserved among homologous proteins. Most prediction methods include annotations from representative protein–nucleic acid complexes in the training sets. Querying for a protein sequence with adequate sequence similarity (sequence identity of more than 30%) to any of the proteins in training datasets would more likely infer binding residues similarly as that of its homologs. Protein sequences used for both case studies however lack such sequence similarities to protein–nucleic acid complexes in the PDB. In both cases, all prediction methods successfully discriminate binding and non-binding residues, regardless of their extent of correctly predicting real binding residues. In such a scenario, the use of PSSM profiles

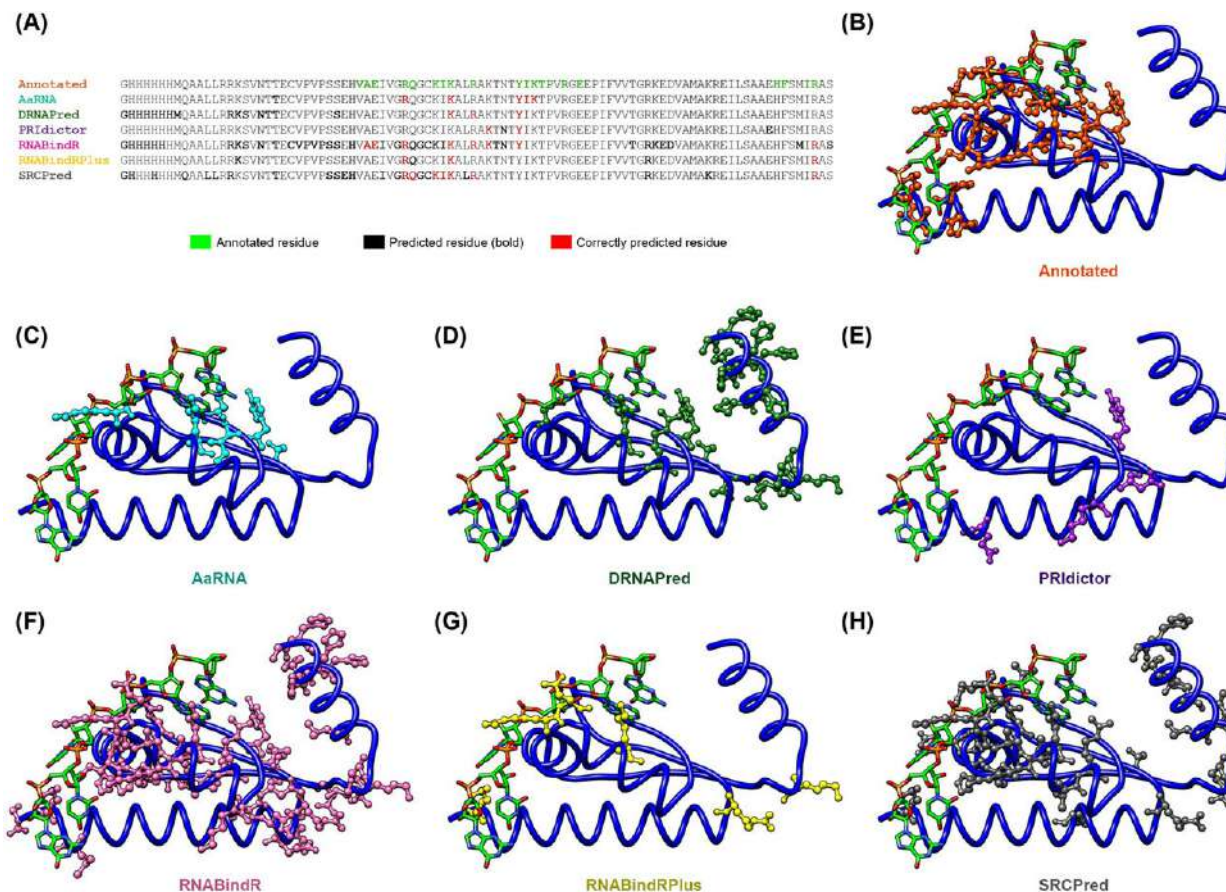


Fig. 2 Protein-RNA interactions in a KH domain protein predicted by different prediction tools. (A) A comparison of binding residues predicted from the query sequence containing 94 amino acids. Annotated binding residues is colored in green, predicted binding residue in bold letter, and residue that is correctly predicted as binding is colored in red. (B) A crystal structure of KH domain from human E3 ubiquitin ligase, with annotated RNA-binding residues colored in orange. (C–H) Predicted binding residues for RNA sequence annotated from different prediction web servers are shown in the structural context, in order, AaRNA, DRNAPred, PRIdictor, RNABindR, RNABindRPlus, and SRCPred. Binding residues are indicated in ball and stick atom representation.

could improve predictions, as similarly suggested by [Walia et al. \(2012\)](#), since evolutionary information for the query protein is retrieved from a larger number of protein sequences compared to the amount of annotated protein structures. Other than that, the definition of contacts used by individual methods could contribute to the performance variations observed. For example, DP-Bind uses a larger cut-off value of 4.5 Å compared to typical distance cut-off value of 3.5 Å adopted by several methods like AaRNA, DQPred, and DRNAPred, thus creating a larger pool of positive samples.

Discussion and Future Directions

Most of the prediction methods highlighted in this article were intended to determine putative residues important for protein–nucleic acid interaction from protein sequence in the absence of prior annotation for nucleic acid binding or sequence homology. Most prediction methods have been designed with a broad selection of features that include evolutionary conservation, amino acid composition, sequence neighborhood, solvent accessibility, and secondary structure, that have been incorporated into well-established prediction models like neural networks and SVMs. Despite the dependence of these methods on the smaller pool of experimental protein structures, they perform reliable discrimination of binding and nonbinding residues with the assumption that nucleic acid binding interfaces can be similar among proteins of the same group. These methods may potentially be useful to determine novel nucleic acid binding proteins on a proteome level, especially in the case of proteins of unknown function, or proteins for which their ability to bind to nucleic acids has yet to be characterized. Another potential application would be to identify residues for docking simulations to visualize the interactions between proposed binding residues for predicted protein structures or structures believed to be DNA/RNA-binding but solved without the nucleic acid partner present. One perspective that could be of great interest for research in this area is the context dependent functional annotations. In this direction, a breakthrough of sorts has recently appeared in the works of [Ahmad et al. \(2017\)](#) in which they have argued that some DBPs might acquire their

function only in specific cellular environments represented by the gene expression levels of their source gene and also others that are coexpressed. We believe this opens a new direction for functional annotations not only for nucleic acid binding but also for functional annotations in general.

Closing Remarks

The abundance of protein sequences in the sequence repositories, including many genomes that consist of a high number of hypothetical proteins, makes computational sequence level tools useful for assigning functional annotations. Specific tools that are able to predict nucleic acid binding residues will thus be of high utility in cases where nucleic acid binding capacity is hypothesized and/or to identify hypothetical proteins that may have nucleic acid binding capability from the numerous genome sequence datasets now available.

Acknowledgments

MFR was funded by the Universiti Kebangsaan Malaysia grant DIP-2017-013 and Ministry of Science, Technology and Innovation, Malaysia grant 02-01-02-SF1278; NSAG was funded by the Ministry of Higher Education, Malaysia grant LEP2.0/14/UKM/BT/02/2.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Biomolecular Structures: Prediction, Identification and Analyses. Computational Design and Experimental Implementation of Synthetic Riboswitches and Riboregulators. Natural Language Processing Approaches in Bioinformatics. Predicting RNA-RNA Interactions in Three-Dimensional Structures. Protein-Protein Interaction Databases. Sequence Analysis

References

- Ahmad, S., Gromiha, M.M., 2002. NETASA: Neural network based prediction of solvent accessibility. *Bioinformatics* 18, 819–824.
- Ahmad, S., Gromiha, M.M., Sarai, A., 2003. RVP-net: Online prediction of real valued accessible surface area of proteins from single sequences. *Bioinformatics* 19, 1849–1851.
- Ahmad, S., Gromiha, M.M., Sarai, A., 2004. Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics* 20, 477–486.
- Ahmad, S., Prathipati, P., Tripathi, L.P., *et al.*, 2017. Integrating sequence and gene expression information predicts genome-wide DNA-binding proteins and suggests a cooperative mechanism. *Nucleic Acids Res.* 1–17. [GX1166](#).
- Ahmad, S., Sarai, A., 2005. PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinform.* 6, 33.
- Al-Shahib, A., Breitling, R., Gilbert, D.R., 2007. Predicting protein function by machine learning on amino acid sequences – A critical evaluation. *BMC Genom.* 8, 78.
- Andrabi, M., Mizuguchi, K., Sarai, A., Ahmad, S., 2009. Prediction of mono- and di-nucleotide-specific DNA-binding sites in proteins using neural networks. *BMC Struct. Biol.* 9, 30.
- Bairoch, A., Apweiler, R., 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.* 28, 45–48.
- Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., A, B., 2007. UniProtKB/Swiss-Prot. *Methods Mol. Biol.* 406, 89–112.
- Chai, H., Zhang, J., Yang, G., Ma, Z., 2016. An evolution-based DNA-binding residue predictor using a dynamic query-driven learning scheme. *Mol. Biosyst.* 12, 3643–3650.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognit. Lett.* 27, 861–874.
- Fernandez, M., Kumagai, Y., Standley, D.M., *et al.*, 2011. Prediction of dinucleotide-specific RNA-binding sites in proteins. *BMC Bioinform.* 12 (Suppl. 13), S5.
- Glisovic, T., Bachorik, J.L., Yong, J., Dreyfuss, G., 2008. RNA-binding proteins and post-transcriptional gene regulation. *FEBS Lett.* 582, 1977–1986.
- Hubbard, S.J., Thornton, J.M., 1993. NACCESS. Department of Biochemistry and Molecular Biology. University College London.
- Hu, J., Li, Y., Zhang, M., *et al.*, 2016. Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 14 (6), 1389–1398.
- Hwang, S., Gou, Z., Kuznetsov, I.B., 2007. DP-Bind: A web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics* 23, 634–636.
- Joo, K., Lee, S.J., Lee, J., 2012. Sann: Solvent accessibility prediction of proteins by nearest neighbor method. *Proteins*, 80. pp. 1791–1797.
- Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.
- Kirsanov, D.D., Zanegina, O.N., Aksianov, E.A., *et al.*, 2013. NPIDB: Nucleic acid-Protein Interaction DataBase. *Nucleic Acids Res.* 41, D517–D523.
- Lancashire, L.J., Lemetre, C., Ball, G.R., 2009. An introduction to artificial neural networks in bioinformatics—application to complex microarray and mass spectrometry datasets in cancer studies. *Briefings in bioinformatics* 10, 315–329.
- Lewis, B.A., Walia, R.R., Terribilini, M., *et al.*, 2011. PRIDB: A protein–RNA interface database. *Nucleic Acids Res.* 29, D277–D282.
- Li, S., Yamashita, K., Amada, K.M., Standley, D.M., 2014. Quantifying sequence and structural features of protein–RNA interactions. *Nucleic Acids Res.* 42, 10086–10098.
- Lunde, B.M., Moore, C., Varani, G., 2007. RNA-binding proteins: Modular design for efficient function. *Nat. Rev. Mol. Cell Biol.* 8, 479–490.
- Marceau, A.H., Felthousen, J.G., Goetsch, P.D., *et al.*, 2016. Structural basis for LIN54 recognition of CHR elements in cell cycle-regulated promoters. *Nat. Commun.* 7, 12301.
- Ma, X., Guo, J., Wu, J., *et al.*, 2011. Prediction of RNA-binding residues in proteins from primary sequence using an enriched random forest model with a novel hybrid feature. *Proteins* 79, 1230–1239.
- McGuffin, L.J., Bryson, K., Jones, D.T., 2000. The PSIPRED protein structure prediction server. *Bioinformatics* 16, 404–405.
- Miao, Z., Westhof, E., 2015. Prediction of nucleic acid binding probability in proteins: A neighboring residue network based score. *Nucleic Acids Res.* 43, 5340–5351.
- Pruitt, K.D., Tatusova, T., Maglott, D.R., 2007. NCBI reference sequences (RefSeq): A curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.* 35, D61–D65.
- Ray, D., Kazan, H., Cook, K.B., *et al.*, 2013. A compendium of RNA-binding motifs for decoding gene regulation. *Nature* 499, 172–177.
- Rose, P.W., Prlic, A., Altunkaya, A., *et al.*, 2017. The RCSB protein data bank: Integrative view of protein, gene and 3D structural information. *Nucleic Acids Res.* 45, D271–D281.

- Rost, B., Sander, C., 1994. Conservation and prediction of solvent accessibility in protein families. *Proteins Struct. Funct. Bioinform.* 20, 216–226.
- Singh, H., Chauhan, J.S., Gromiha, M.M., Raghava, G.P., 2012.ccPDB: Compilation and creation of data sets from Protein Data Bank. *Nucleic Acids Res.* 40, D486–D489.
- Sokolova, M., Japkowicz, N., Szpakowicz, N., 2006. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In: *Proceedings of the Australasian Joint Conference on Artificial Intelligence*, vol. 4304, pp. 1015–1021.
- Sun, T., Zhou, B., Lai, L., Pei, J., 2017. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinform.* 18, 277.
- Touw, W.G., Baakman, C., Black, J., *et al.*, 2015. A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.* 43, D364–D368.
- Tuvshinjargal, N., Lee, W., Park, B., Han, K., 2016. PRIdictor: Protein–RNA Interaction predictor. *Biosystems* 139, 17–22.
- Walia, R.R., Caragea, C., Lewis, B.A., *et al.*, 2012. Protein–RNA interface residue prediction using machine learning: An assessment of the state of the art. *BMC Bioinform.* 13, 89.
- Walia, R.R., Xue, L.C., Wilkins, K., *et al.*, 2014. RNABindRPlus: A predictor that combines machine learning and sequence homology-based methods to improve the reliability of predicted RNA-binding residues in proteins. *PLOS ONE* 9, e97725.
- Wang, G., Dunbrack, R.L., 2003. PISCES: A protein sequence culling server. *Bioinformatics* 19, 1589–1591.
- Wang, S., Peng, J., Ma, J., Xu, J., 2016. Protein secondary structure prediction using deep convolutional neural fields. *Sci. Rep.* 6, 18962.
- Yang, L., Wang, C., Li, F., *et al.*, 2017. The human RNA-binding protein and E3 ligase MEX-3C binds the MEX-3-recognition element (MRE) motif with high affinity. *J. Biol. Chem.* 292, 16221–16234.
- Yan, J., Kurgan, L., 2017. DRNAPred, fast sequence-based method that accurately predicts and discriminates DNA- and RNA-binding residues. *Nucleic Acids Res.* 45, e84.

Protein-Peptide Interactions in Regulatory Events

Upadhyayula S Raghavender, Birla Institute of Scientific Research (BISR), Jaipur, India

Ravindranath S Rathore, Central University of South Bihar, Patna, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Protein-Protein Interactions (PPI)

Eukaryotic cell is a complex collection of physical entities, i.e., biological molecules, which are in a constant flux of association and dissociation reactions. The milieu of cell provides the necessary and sufficient environment for cooperative interactions taking place between biological macromolecules (proteins, RNA, DNA, peptides). Majority of the physical interactions between the cellular components are context-dependent and are highly *specific*. In the case of proteins, the signals for binding events are encoded in their sequences. These signals, or interaction modules, diversify the functions of the proteins to performing activities like catalytic, regulatory, and binding a wide array of ligands. The multifarious functions of proteins in a cell is the reason for the diversity of interactions in a cell (Van Roey *et al.*, 2014).

The eukaryotic cell is composed of many compartments within which a host of reactions and interactions are happening at exceptionally *fast rate* and with high *accuracy* and *fidelity*. The classical approach to understanding interactions is that of a structured protein interacting with a partner. Traditionally, protein-protein interactions (PPIs) are viewed as archetypal interactions between molecules with high affinity and stable domain structures. The assertion that stable, ordered and structured domains are necessary for describing the “interactome” has given way to recent observations that unstructured regions in a protein play a critical role in explaining the cellular events and dynamics of interactions. The regulatory events in a cell tend to opt, perhaps a large fraction, regions of protein without a well-defined stable conformation for the binding sites. These seminal observations have paved the way for a new field of interactome studies known as *Intrinsically Disordered Regions* (IDRs) in proteins (Tompa, 2012). A short stretch of amino acids in protein sequences, in particular IDRs, comprises of binding residues (*functional modules*). These linear interaction sites, or short peptides, have been designated as *short linear motifs* (SLiMs). Hence, the entire array of functions which can be performed by a protein are concentrated in SLiMs. With the availability of a suitable partner which can recognise these sites and bind to them, a sizeable fraction of the interactome can be understood quantitatively (Van Roey *et al.*, 2014; Gouw *et al.*, 2017).

Protein-protein interactions form the basis for many cellular processes. The protein interaction refers to specific physical contact(s) between two or more proteins, forming biologically active protein complex. There are two categories of contacts – *direct* PPI and *indirect* PPI. Indirect PPI are those wherein two interacting proteins may interact forming a complex via intervening partners. Disruption or deregulation of these complex interactions is the main cause of a significant number of diseases. Consequently, there is intense research interest in designing inhibitors that target specific protein-protein interactions. This places intricate protein-protein interactions at the heart of the development for novel protein-based drug leads often referred to as *biologics* (Caputo *et al.*, 2008). The emergence of ‘omic’ technologies, namely genomics, transcriptomics and proteomics, has greatly accelerated our understanding of the protein-protein interaction networks leading to the discovery of a number of proteins and their interactions (Archakov *et al.*, 2003).

Structural Intricacies of PPIs

The number of complex structures has exponentially grown in the Protein Data Bank in recent decades, which partly owes to various Structural Genomics initiatives. With the availability of a large number of structures of protein-protein complexes, the focus in the area has been shifted to understanding the molecular determinants of binding affinity and specificity of complex. The description of the interface in a protein-protein complex serves as a starting point for understanding the chemistry and physics at the interface. When complemented with the physico-chemical description of residues lining the interface, the study becomes quantitative providing deep insight into the interplay of strong and weak interactions during complex formation (Jones and Thornton, 1996; Bravo and Aloy, 2006).

PPI surfaces are relatively shallow and lack features as compared to protein-ligand or drug interactions. These interactions can be viewed as arising out of many cooperative weak interatomic interactions, in contrast to a few strong interactions dictating protein-ligand complex formation (Jones *et al.*, 2000). PPI's are known to be highly regulated and depend on the environment in which they occur. In many cases, the complex formation occurs by post translational modification (like phosphorylation) which promotes recognition by conformational changes in the binding partners. SLiMs are inherently labile and serve as *regulatory motifs* in promoting a large set of protein interactions in surprisingly complex ways (Davey *et al.*, 2015).

Computational Approaches

The existing high throughput experimental approaches such as two-hybrid, chip-based or TAP-MS to map protein interactions suffers from several limitations – Limited data set with high degree of false positives and negatives. X-ray crystallography,

employed to elucidate atomic details of protein interactions, is often not feasible due to crystallization difficulties and complexities of interactions involved in case of membrane domains and multiple interacting partners. Many of the proteins are estimated to possess several interacting partners (Gogl *et al.*, 2013; Meller and Porollo, 2012). The computational methods have the potential to fill-in the gap in this scenario. Computation can be used to predict possible interactions and binding modes, to validate the experimental high-throughput screening data and to analyze the PPI networks from the data repositories (Zahiri *et al.*, 2013). Over 100 such PPI related repositories are available online (Orchard *et al.*, 2012), the prominent among them are – Biological General Repository for Interaction Datasets (BioGRID), Database of Interacting Proteins (DIP), Biomolecular INteraction Network Database (BIND), Molecular Interaction Database (MINT), Human Protein Reference Database (HPRD), Search Tool for the Retrieval of Interacting Genes/Protein (STRING) and IntAct (Miryala *et al.*, 2017).

Protein-Peptide Interactions

Protein-peptide interactions play an important role in many regulatory processes of cell from co-activators to inhibitors and account for almost 40% of the protein-protein interactions. Peptides characteristically interact with specific protein domains such as SH2, SH3, MFC and PDZ domains. Thus development of peptide or peptidomimetic leads is an emerging area in biological therapeutics. Peptides 'as against small molecules' possess specific motifs hence they can mimic protein-binding domains and at the same time they are large enough to serve as competitive inhibitors for disrupting protein-protein interactions. With the objective of developing PPI inhibitors based on peptides or designed peptide derivatives to mimic the binding motif of one of the partners, one usually starts with a sequence similar to or harbouring the *recognition motif* (Stanfield and Wilson, 1995). In solution, the free form of peptides tend to possess a large array of conformations. But, in the presence of an interaction partner there is a drastic reduction in the conformational space sampled by the peptide and the complexation is promoted by the recognition motif with a concomitant lowering of the total entropy of the system (Jackrel *et al.*, 2009; Speltz *et al.*, 2015). It has been commented that PPIs are stabilised in the milieu of strong and weak interactions; but in the case of protein-peptide complexes the hydrogen bonds (particularly backbone-mediated) are pivotal and are key to complex formation (Zvelebil and Thornton, 1993). An attempt to correlate the geometrical parameters of the individual residues in designed peptides, in both free (deposited in Cambridge Structural Database) and bound (as in PDB), has revealed similarities in backbone conformations in the case of aliphatic residues from both the databases. This led to the generation of residue level backbone conformational libraries of peptide residues (canonical) in PDB (Raghavender, 2017).

Peptide-protein interactions have biological significance. They are important signaling components, acting as hormones and neurotransmitters (Brooks *et al.*, 2005; Petsalaki and Russell, 2008). The most famous and important example of a disease involving peptides derived from natural proteins is amyloid beta ($A\beta$), resulting from the hydrolysis of APP protein. Antimicrobial peptides are implied in innate immune system (Ganz, 2003). Further fusogenic peptides have been used as cargo to deliver drugs to target cells (Trabulo *et al.*, 2010). It has been claimed that this subclass of interactions are largely underrepresented in high-throughput experimental techniques used for studying protein-protein interactions (Lensink *et al.*, 2017). The transient nature of the interaction, dictated by the low affinity binding interfaces compounds the existing problems. Computational predictions of peptide binding to interaction partners have been assessed recently (Lensink *et al.*, 2017).

Theoretical Background

Polypeptide Conformation

G. N. Ramachandran is credited with the seminal contribution of describing the conformation of a polypeptide chain in 1960s (Ramachandran and Sasiskharan, 1968; Ramakrishnan, 2001). This resulted in what is now popularly known as **Ramachandran** or (ϕ, ψ) map. The basic idea is to get a 2-dimensional representation of the protein conformation, using backbone dihedral angles (ϕ, ψ). Surprisingly, the map was able to clearly separate regions of regular secondary structural features (α -helices and β -sheets) (Ramachandran *et al.*, 1963). Reverse turns such as α and β -turns, involved in the reversal of the polypeptide chain, could also be completely described using the backbone torsions (Venkatachalam, 1968; Nataraj *et al.*, 1995). The field of designed peptides gained momentum, with the application of conformationally constrained residues which were amenable to chemical synthesis (Balaram, 1992; Toniolo *et al.*, 2001). Aminoisobutyric acid (Aib) is a prototypical achiral amino acid which has been widely examined for the design of helices specifically 3_{10} helices. Proline, an imino acid, is known to occur frequently at the β -turn positions facilitating chain reversals. Hence, both L- and D-enantiomers of proline have been used to design synthetic β -turns as models of β -sheets (Mahalakshmi and Balaram, 2006).

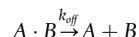
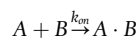
The side chain conformations of residues (rotamers) were also observed to adopt specific conformations. This led to the development of rotamer libraries (Ponder and Richards, 1987) for modelling residue side chain conformations. It was also profitably used in sampling probable conformations in molecular modelling and simulations of proteins and in accurately modelling side chains into electron density maps of structures with not so high resolution (Dunbrack and Karplus, 1993; Dunbrack, 2002; Beglov *et al.*, 2012; Lovell *et al.*, 2000).

Many regulatory events in the eukaryotic cell are transient in nature, meaning they are short-lived as compared to rigid and stable protein complexes. In some such instances, either a short segment of a polypeptide chain or a fragment is involved in interacting with a protein. Initial attempts to elucidate the structural features of protein-peptide interactions revealed that extended

chains (β -strands), β -turns and then by α -helix are the ways in which peptides fold and interact with their partners (Zvelebil and Thornton, 1993). The motifs in these segments were called 'recognition motifs' in the earlier days and are now appropriately called 'SLiMs'. In proteases, like HIV-1 protease, MHC-1, MHC-II, chaperones like PapD and Src homology domains (SH2 & SH3) the recognition motif is usually in an extended chain. Peptide and anti-peptide antibody complexes are frequently seen to bind a peptide with β -turn. Ca^{2+} -dependent calmodulin domain is seen to bind helical peptides. It has become clear that sequence conservation is not necessary for binding to a peptide-binding site, still structures of the bound peptides would have local conserved features (Stein and Aloy, 2008).

Statistical Mechanical Formulation of Molecular Association

A short review of the physio-chemical principles of binding are presented in this section. This should help reader get familiar with the methods used for binding energy estimation and computational docking currently in vogue. We begin with a simplistic formulation of molecular association and disassociation, borrowing from Kilburg and Gallicchio, (2017). Two physical entities A and B can associate by molecular interactions, in a reversible way and at equilibrium can be described by the equations:



The rate of forward and backward reactions are equal at equilibrium, hence *equilibrium constant* for binding K_b can be derived as:

$$K_b = \frac{1}{C^0} \frac{k_{on}}{k_{off}}$$

where C^0 is the standard state concentration set to 1 M. We can now get the standard *binding free energy* ΔG_b^0 , which in principle is a thermodynamic measure of the *binding affinity*, as:

$$\Delta G_b^0 = -k_B T \ln K_b$$

where k_B is Boltzmann's constant. Such a simple formulation will be difficult to reconcile in a biological context, when the interacting species are in a constant flux of interconverting conformations. Hence, the need to resort to statistical mechanical approaches for modelling an "ensemble" of conformations. We skip the derivation and introduce the result of *configurational partition function* of the complex as:

$$Z_{N, A \cdot B} = \int_{\text{bound}} e^{-\beta U(x_A, x_B, \zeta_B, r_s)} dx_A dx_B d\zeta_B dr_s$$

where N is the number of molecules, r_s represents the degrees of freedom of solvent, $U(x)$ represents potential energy of solvent + solute system and ζ_B is any suitable structural parameter (e.g., distance between the center of mass of A and B). The above formulation is the backbone for present-day theoretical models on understanding protein-ligand/protein interactions.

The free energy calculation methods implemented in various programs fall into two distinct approaches. The first approach is based on pathways such as Free Energy Perturbation (FEP), Thermodynamic Integration (TI) and Slow Growth (SG). Such methods are very accurate but computationally intensive. The second set of approaches are less rigorous end-point (two-points) methods such as Linear interaction energy (LIE), Molecular Mechanics-Poisson Boltzmann Surface/Generalized Born Area (MM-PBSA/GBSA) and lambda (λ)-dynamics. They try to balance speed and accuracy to increase throughput to calculate free energy differences (Rathore et al., 2013; Reddy et al., 2014). An end-point method is effective as it ignores the intervening pathway intermediates when evaluating binding free energy, with the description being complete with just free protein, free peptide and protein-peptide states. The binding free energy is estimated in MM-GBSA formulation as:

$$\Delta G_{\text{bind}} = G_{A \cdot B} - G_A - G_B$$

with, $G = \langle E_{\text{bond}} \rangle + \langle E_{\text{ele}} \rangle + \langle E_{\text{vdW}} \rangle + \langle E_{\text{pol}} \rangle + \langle E_{\text{np}} \rangle - TS$

where the contributions come from interactions through covalent bonds (E_{bond}), electrostatic interactions (E_{ele}), van der Waals (E_{vdW}), polar (E_{pol}) and non-polar (E_{np}) forces to solvation free energy. These are typically modelled under implicit solvent representations.

Prediction & Modelling of Protein-Peptides Interactions

Protein-Peptide Docking

The interest towards the development of novel peptide or peptidomimetic has grown substantially in recent times. Peptides are highly flexible and they often interact weakly with their substrate, underlining their importance in signal transduction or regulation which often relies on transient processes. These obstacles make experimental structure determination often non-trivial and calls for complementary computational approaches like docking. Molecular docking is a preliminary step in lead identification and design. Although many well-established methods exist for protein-ligand and protein-protein rigid-body docking, they often fail to yield accurate results in cases that involve receptor flexibility and docking of large, flexible ligands like peptides (Audie and Swanson, 2012; Trellet et al., 2013). Peptides in contrast to proteins are observed to adopt variable conformations. The excursions into

conformational space can be curtailed by designing sequences which have conformationally constrained residues like Pro or non-standard aminoisobutyric acid (Aib) (Mahalakshmi and Balaram, 2006; Raghavender *et al.*, 2010). In a computational investigation involving peptide-protein docking, the peptide conformations are sampled either by running an explicit solvent molecular dynamics (MD) simulation or by a procedure involving implicit solvent simulations using Monte Carlo approaches (Raghavender and Sowdhamini, 2015). The latter is faster and yields results quickly, hence recommended for studies in general. The trajectories in a MD are clustered and low-energy representatives from top clusters are used for docking studies. Due to the lack of prior information on allowed conformations of peptide, it is suggested that all possible conformations be explored for studies.

Docking a peptide to a protein is relatively challenging as it involves considerably more number of degrees of freedom, as opposed to protein-ligand or protein-protein docking. The structure of the receptor protein must be known or one should have structural information of a closely related homolog (Kilburg and Gallicchio, 2016). Present protein-peptide docking procedures fall into two classes, namely:

- (a) *Local Docking* – When there is information available of the binding site on the protein, predicting interactions is more accurately possible (Lavi *et al.*, 2013). In this case, a library of peptides are evaluated for best complex structure. HADDOCK program is suited for this type of investigation. It even allows the user to guide the docking by incorporating the biochemical and biophysical data into the docking protocol (Spiliotopoulos *et al.*, 2016). GalaxyPepDock is web based service which employs similarity-based docking. It builds model peptide conformations by identifying templates from the database of experimentally determined structures followed by energy-based optimization that allows for structural flexibility (Lee *et al.*, 2015). MedusaDock is another program which can effectively be used for induced-fit docking procedures (Ding and Dokholyan, 2013). The best pose selection in any of the above programs follows the authors own approach, hence there is a need to look at each pose carefully before proceeding further with analysis. Glide peptide docking module in Bioluminate (Schrödinger, 2017) systematically improve pose prediction accuracy of flexible peptides by enhancing characteristic Glide sampling using a series of hierarchical filters to search for accurate binding mode. The scoring of the poses is improved by post-processing with implicit solvation based MM-GBSA calculations (Tubert-Brohman *et al.*, 2013). Glide peptide docking method is accurate and several times faster than other programs. Rosetta FlexPepDock is a robust protocol for Protein-peptide docking (Raveh *et al.*, 2010) which tries to optimize the peptide backbone and rigid body orientation, using the Monte-Carlo with Minimization approach. Dyna method is new approach for docking peptides into flexible receptors. It follows a two step procedure: the Protein-Peptide conformational space is scanned first to approximately identify ligand poses followed by the pose-refinement by a new molecular dynamics-based method, optimized potential molecular dynamics (OPMD) using soft-core potentials for the protein-peptide interactions (Antes, 2010).
- (b) *Global docking* – When there is little or no information available on the binding site, it needs to be searched for on protein surface. Such methods have limited accuracy for predicting the complex structures. However, there are recent examples of successful predictions of such methods (Yan *et al.*, 2016; Petsalaki *et al.*, 2009). In this scenario, one performs a docking with an ensemble of peptide conformations onto the surface of the target protein. Typically, one can complement it with some biochemical information on the plausibility of the binding site to be correlated with some interaction. Generally, one proceeds in a two-tiered way involving coarse-grained modelling of the receptor for filtering binding sites followed by atomic resolution docking and refinement. Programs like CABS-dock and PepATTRACT have been used recently in such studies (Blaszczuk *et al.*, 2016; De Vries *et al.*, 2017).

Predicting Hot-Spot Residues: Residue Scanning

The prediction of energetically important amino acid residues at the interface of protein-protein complexes is a step forward in biologics design. Such *hot-spot* residues (i.e., residues that significantly destabilize the PPI when mutated to Ala or specific residues possessing different chemical and structural character) have been investigated experimentally as well as computationally. Experimental site-directed mutagenesis is a powerful method for delineating important residues and key interactions at the protein-protein interfaces. A number of reports describe computational mutagenesis involving calculation of the effect of mutations on the binding free energy of protein-protein complexes using empirical energy models such as Robetta & FoldX (Kortemme *et al.*, 2004; Schymkowitz *et al.*, 2005). Other sophisticated methods of binding free energy calculation upon alanine mutation using implicit solvation models such as MM-PBSA/GBSA and Thermodynamic Integration have been proposed (Martins *et al.*, 2013; Beard *et al.*, 2013) even though they are computationally much more expensive. The MM-GBSA approach implemented in Bioluminate Residue scanning module (Schrödinger, 2017) employs the OPLS force field and the VSGB2.0 solvent model to calculate binding free energy differences between wild type and mutants. The individual or multiple residues are mutated followed by rotamer search algorithms implemented in Prime (Schrödinger, 2017). In an experimental data set comprising of 418 single residue mutations in 21 targets, the MM-GBSA based method has been demonstrated to perform well at picking hot-spots, and mutations within an accuracy of 1 kcal/mol (Beard *et al.*, 2013).

Case Studies

We present a few of the research investigations here which cover in part of the above concepts.

- (i) Nuclear receptor (NR) box II peptide, with LxxLL motif (x is any amino acid), binds to oestrogen receptor α . One can see that the peptide forms an amphipathic α -helix and is lodged in a complementary groove on the receptor surface (Shiau *et al.*, 1998).

- (ii) PDZ domains which are protein interaction modules involved in signaling networks, bind the C-terminal regions of partners by adding a β -strand. This is known as β -augmentation (Remaut and Waksman, 2006).
- (iii) SH3, WW and EVH1 domains, bind to linear motifs rich in proline residues and adopting polyproline II helix structures (Zarrinpar et al., 2003).
- (iv) SH2, PTB and 14-3-3 domains are known to bind sites involved in post-translational modifications (Rittinger et al., 1999).
- (v) MDM2-p53 interaction is known to be lethal. Mouse Double minute 2 (MDM2) binds to a short helix on p53. This interaction was mimicked by *cis*-imidazoline analogs, called Nutlins, which could completely inhibit the complex formation (Vassilev et al., 2004).

Discussion

We elaborate on few of the important investigations discussed above in the following section.

- (i) *NR box II peptide + oestrogen receptor α (ER α)*. The transcription factor ER α is involved in the regulation of differentiation and maintenance of neural, skeletal, cardiovascular and reproductive tissues. Treatment of breast cancer, osteoporosis and cardiovascular disease employs compounds which modulate ER α activity. ER α binds a variety of ligands, and all the ligands bind at the C-terminus. In the structure of NR box II peptide bound to ER α , it can be seen that the peptide is helical, with approximately 1000 Å² hydrophobic surface area buried at the interface. The ligand binding domain (LBD) interacts primarily with the side chains of the peptide residues Ile-689, and Leu-690, Leu-693 and Leu-694. This interactions was found to be very potent and involving mostly non-polar interactions (van der Waals). Terminal capping interactions stabilize the lodged peptide, the γ -carboxylate of Glu-542 hydrogen bonds to the N-terminus of the peptide and Lys-362 (ϵ -amino group) forms hydrogen bonds with the C-terminus of the peptide helix (PDB code: 3ERD).

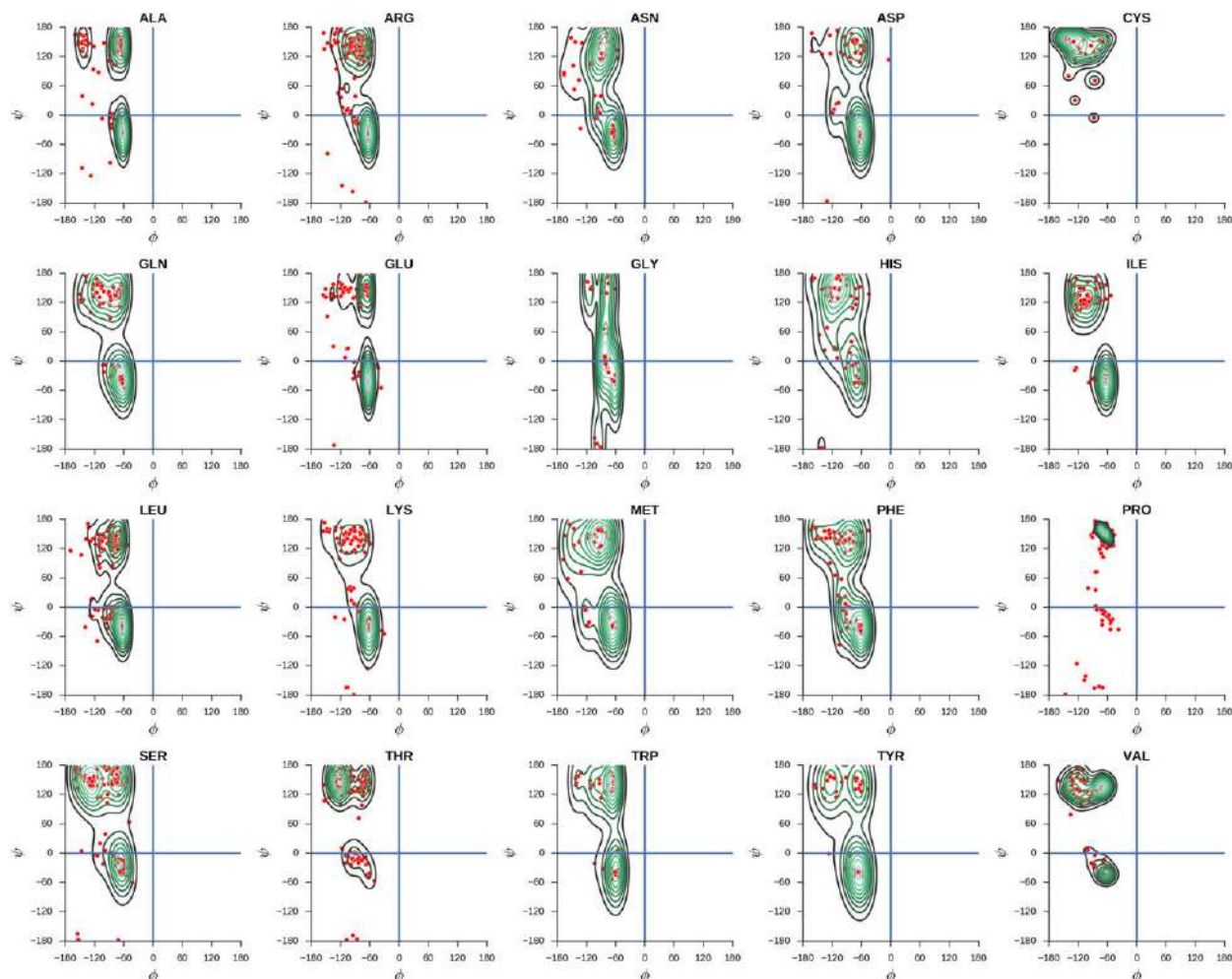


Fig. 1 Bivariate kernel density estimates (kde) of backbone torsion angles (ϕ , ψ) of peptide residues bound to protein partners in PDB. High resolution crystal structures, better than 1.5 Å, were used for the analysis. Only canonical residues have been investigated, with a view of utility in chemical synthesis of analogs. Cyclic peptides were excluded from the study.

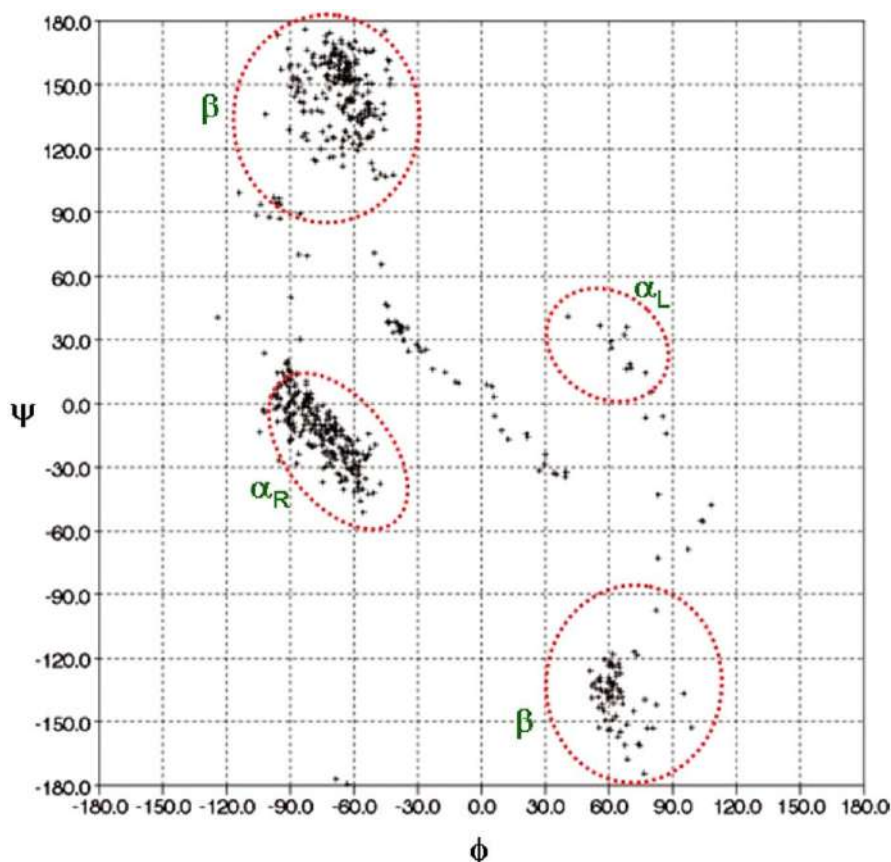


Fig. 2 Ramachandran Plot of the proline residues in CSD. The regions corresponding to helical (α_R -right-handed α_L -left-handed) and extended strands are marked in red circles. Cyclic peptides are also included in this plot for completeness, and they tend to be clustered at (0, 0). Reproduced from the Doctoral Thesis of Dr. U.S. Raghavender.

- (ii) *C-terminus pentapeptide from post synaptic protein CRIPT + Synaptic protein PSD-95*. PDZ domains are known to be involved in important metabolic pathways. In the structure of synaptic protein PSD-95 in complex with a peptide derived from C-terminus of CRIPT, one observes that the peptide adopts a β -strand conformation and binds to the edge of the β -sheet from PSD-95. The protein-peptide complex formation in this case is augmented by the main chain interactions. The affinity of the interaction is directed by peptide backbone interactions leaving a wide scope for specificity, promoting the binding of shorter motifs at the binding site (PDB code: 1BE9).
- (iii) *MDM2-Nutlin-2 complex*. MDM2 is a ubiquitin protein ligase and is known to bind to a short amphipathic α -helix on the p53 tumor suppressor and modulate its transcriptional activity and stability. The functions of p53 are impaired upon overexpression of MDM2 in human tumours. Hence, inhibition of MDM2-p53 interaction is a way forward for cancer therapy. The small molecule inhibitor, Nutlin-2, mimics the interactions of the p53 peptide to a high degree, with one bromophenyl moiety sitting deeply in the Trp pocket, the other bromophenyl group occupying the Leu pocket, and the ethyl ether side chain directed toward the Phe pocket. The imidazoline scaffold serves as a functional replacement of helical backbone of the peptide and is able to direct the projection of three groups into the pockets normally occupied by Phe. This is one of the very first studies in identifying small molecule compounds which involved disrupting a pathological protein-peptide interaction (PDB code: 1RV1).

Future Directions

First principle studies on protein-peptide interaction are presently out of reach from the perspective of both methods and the software programs implementing the theoretical approaches. A great step forward would be to comprehensively catalog, characterize, and associate Protein-peptide interactions with disease and other phenotypes. There is a good scope for development in all the areas of mechanistic description of protein-peptide interactions. In particular, sound theoretical approaches to accurately model the interaction incorporating both the chemical and physical information can be explored. Peptide design, protein-peptide docking and refinement of docked complexes have already been addressed computationally using many different approaches; still the search is ON for the best method which would be able to predict the correct protein-peptide interaction accounting for all the

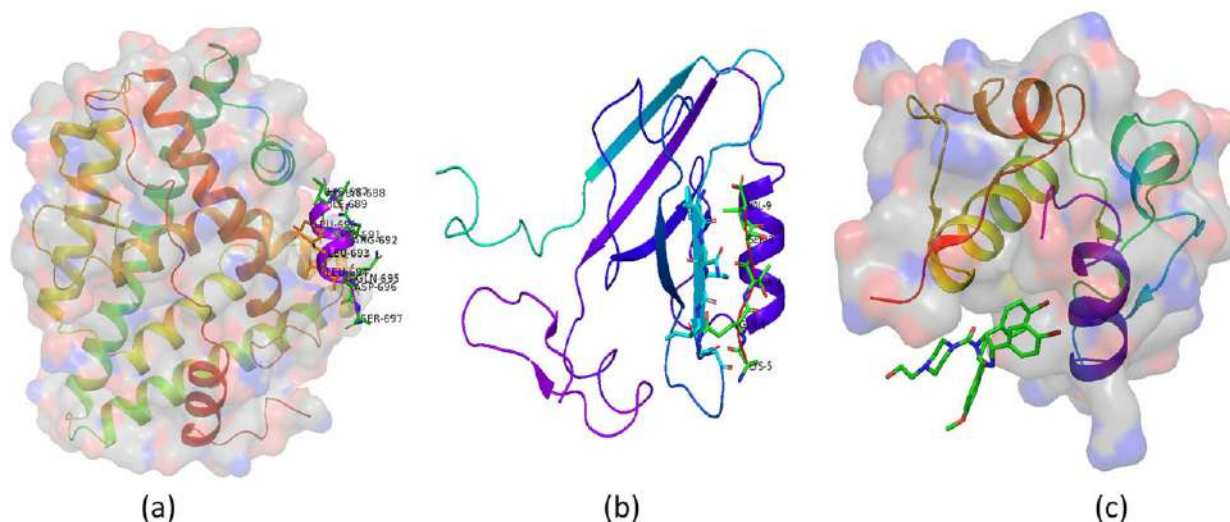


Fig. 3 (a) NR box II peptide + oestrogen receptor α (ER α). (b) C-terminus pentapeptide from post synaptic protein CRIPT + Synaptic protein PSD-95. (c) MDM2-Nutlin-2 complex. The molecular surface is shown imposed on the cartoon representation of receptors. The peptide residues are labelled. Nutlin-2 is shown as sticks in green in (c).

variabilities in peptide conformations. Computers with ever increasing and improving processing speeds and hardwares, are making these studies feasible in desktop workstations. With the surge of GPUs, there is a good scientific reason to traverse to the new avenues exploiting the new computer architectures for these studies.

Closing Remarks

The prevalence of peptide-protein interactions in a cell has been established by many complementary physical, chemical and biochemical experiments. It is very clear that these are very abundant in a cell and are primarily involved in cellular regulation, thus offering enormous scope for protein therapeutics. The existing experimental approaches to characterize PPI have high rates of false positives and computational methods serve an important role to predict and derive a mechanistic picture of these interactions. The limited accuracy of predictive methods calls for improving the existing computational methodologies. Additional and complementary physical and computational approaches can help remove ambiguity in predicting a true positive interaction, which can subsequently be validated using experimental techniques. The description of the entire interactome warrants accurate characterization and prediction of this important subclass of PPIs (Figs. 1, 2, 3).

Acknowledgement

Raghavender acknowledges SERB-DST for funding (File No: YSS/2015/1077/LS). Infrastructure support of BISIR is acknowledged.

See also: Natural Language Processing Approaches in Bioinformatics

References

- Antes, I., 2010. DynaDock: A new molecular dynamics-based algorithm for protein– Peptide docking including receptor flexibility. *Proteins* 78, 1084–1104.
- Archakov, A.I., Govorun, V.M., Dubanov, A.V., *et al.*, 2003. Protein-protein interactions as a target for drugs in proteomics. *Proteomics* 3, 380–391.
- Audie, J., Swanson, J., 2012. Recent work in the development and application of protein– Peptide docking. *Future Med. Chem.* 4, 1619–1644.
- Balaran, P., 1992. Protein non-standard amino acids in peptide design and protein engineering. *Curr. Opin. Struct. Biol.* 2, 845–851.
- Beard, H., Cholleti, A., Pearlman, D., Sherman, W., Loving, K.A., 2013. Applying physics-based scoring to calculate free energies of binding for single amino acid mutations in protein-protein complexes. *PLOS ONE* 8, e82849.
- Beglov, D., Hall, D.R., Brenke, R., *et al.*, 2012. Minimal ensembles of side chain conformers for modeling protein-protein interactions. *Proteins* 80, 591–601.
- Błaszczak, M., Kurcinski, M., Kouza, M., *et al.*, 2016. Modeling of protein-peptide interactions using the CABS-dock web server for binding site search and flexible docking. *Methods* 93, 72–83.
- Bravo, J., Aloy, P., 2006. Target selection for complex structural genomics. *Curr. Opin. Struct. Biol.* 16, 385–392.
- Brooks, H., Lebleu, B., Vives, E., 2005. Tat peptide-mediated cellular delivery: Back to basics. *Adv. Drug Deliv. Rev.* 57, 559–577.
- Caputo, G.A., Litvinov, R.I., Li, W., *et al.*, 2008. Computationally designed peptide inhibitors of protein-protein interactions in membranes. *Biochemistry* 47, 8600–8606.

- Davey, N.E., Cyert, M.S., Moses, A.M., 2015. Short linear motifs – *Ex nihilo* evolution of protein regulation. *Cell Commun. Signal.* 13, 43.
- De Vries, S.J., Rey, J., Schindler, C.E.M., Zacharias, M., Tuffery, P., 2017. The pepATTRACT web server for blind, large-scale peptide-protein docking. *Nucleic Acids Res.* 45, W361–W364.
- Ding, F., Dokholyan, N.V., 2013. Incorporating backbone flexibility in MedusaDock improves ligand-binding pose prediction in the CSAR2011 docking benchmark. *J. Chem. Inf. Model.* 53, 1871–1879.
- Dunbrack Jr., R.L., 2002. Rotamer libraries in the 21st century. *Curr. Opin. Struct. Biol.* 12, 431–440.
- Dunbrack Jr., R.L., Karplus, M., 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. *J. Mol. Biol.* 230, 543–574.
- Ganz, T., 2003. Defensins: Antimicrobial peptides of innate immunity. *Nat. Rev. Immunol.* 3, 710–720.
- Gogl, G., Toro, I., Remenyi, A., 2013. Protein-peptide complex crystallization: A case study on the ERK2 mitogen-activated protein kinase. *Acta Crystallogr. D Biol. Crystallogr.* 69, 486–489.
- Gouw, M., Samano-Sanchez, H., Van Roey, K., *et al.*, 2017. Exploring Short Linear Motifs Using the ELM Database and Tools. *Curr. Protoc. Bioinform.* 58, 8.22.1–8.22.35.
- Jackrel, M.E., Valverde, R., Regan, L., 2009. Redesign of a protein-peptide interaction: Characterization and applications. *Protein Sci.* 18, 762–774.
- Jones, S., Marin, A., Thornton, J.M., 2000. Protein domain interfaces: Characterization and comparison with oligomeric protein interfaces. *Protein Eng.* 13, 77–82.
- Jones, S., Thornton, J.M., 1996. Principles of protein-protein interactions. *Proc. Natl. Acad. Sci. USA* 93, 13–20.
- Kilburg, D., Gallicchio, E., 2016. Recent advances in computational models for the study of protein-peptide interactions. *Adv. Protein Chem. Struct. Biol.* 105, 27–57.
- Kortemme, T., Kim, D.E., Baker, D., 2004. Computational alanine scanning of protein-protein interfaces. *Sci. STKE* 2004, pl2.
- Lavi, A., Ngan, C.H., Movshovitz-Attias, D., *et al.*, 2013. Detection of peptide-binding sites on protein surfaces: The first step toward the modeling and targeting of peptide-mediated interactions. *Proteins* 81, 2096–2105.
- Lee, H., Heo, L., Lee, M.S., Seok, C., 2015. GalaxyPepDock: A protein-peptide docking tool based on interaction similarity and energy optimization. *Nucleic Acids Res.* 43, W431–W435.
- Lensink, M.F., Velankar, S., Wodak, S.J., 2017. Modeling protein-protein and protein-peptide complexes: CAPRI 6th edition. *Proteins* 85, 359–377.
- Lovell, S.C., Word, J.M., Richardson, J.S., Richardson, D.C., 2000. The penultimate rotamer library. *Proteins: Struct. Funct. Genet.* 40, 389–408.
- Mahalakshmi, R., Balam, P., 2006. Non-protein amino acids in the design of secondary structure scaffolds. *Methods. Mol. Biol.* 340, 71–94.
- Martins, Silvia A., Perez, Marta A.S., Moreira, Irina S., *et al.*, 2013. Computational alanine scanning mutagenesis: MM-PBSA vs TI. *J. Chem. Theory Comput.* 9, 1311–1319.
- Meller, J., Porollo, A., 2012. Computational methods for prediction of protein-protein interaction sites. In: *Proceedings of the Protein-Protein Interactions – Computational and Experimental Tools* edited by Weibo Cai and Hao Hong, ISBN 978-953-51-0397-4.
- Miryal, K., AnandAnbarasu, A., Ramaiah, S., 2017. Gene, in press. Discerning molecular interactions: A comprehensive review on biomolecular interaction databases and network analysis tools. doi: 10.1016/j.gene.2017.11.028.
- Nataraj, D.V., Srinivasan, N., Sowdhamini, R., Ramakrishnan, C., 1995. Alpha turns in protein structure. *Curr. Sci.* 69, 435–447.
- Orchard, S., Kerrien, S., Abbani, S., *et al.*, 2012. Protein interaction data curation: The International Molecular Exchange (IMEx) consortium. *Nat. Methods* 9, 345–350.
- Petsalaki, E., Russell, R.B., 2008. Peptide-mediated interactions in biological systems: New discoveries and applications. *Curr. Opin. Biotechnol.* 19, 344–350.
- Petsalaki, E., Stark, A., Garcia-Urdiales, E., Russell, R.B., 2009. Accurate prediction of peptide binding sites on protein surfaces. *PLOS Comput. Biol.* 5, e1000335.
- Ponder, J.W., Richards, F.M., 1987. Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.* 193, 775–791.
- Raghavender, U.S., 2017. Analysis of residue conformations in peptides in Cambridge structural database and protein-peptide structural complexes. *Chem. Biol. Drug Des.* 89, 428–442.
- Raghavender, U.S., Aravinda, S., Rai, R., Shamala, N., Balam, P., 2010. Peptide hairpin nucleation with the obligatory Type I' beta-turn Aib-DPro segment. *Org. Biomol. Chem.* 8, 3133–3135.
- Raghavender, U.S., Sowdhamini, R., 2015. Mechanistic basis of peptide-protein interaction in AtPep1-PEPR1 complex in *Arabidopsis thaliana*. *Protein Pept. Lett.* 22, 618–627.
- Raveh, B., London, N., Schueler-Furman, O., 2010. Sub-angstrom modeling of complexes between flexible peptides and globular proteins. *Proteins: Struct. Funct. Bioinform.* 78, 2029–2040.
- Ramachandran, G.N., Ramakrishnan, C., Sasisekharan, V., 1963. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* 7, 95–99.
- Ramachandran, G.N., Sasisekharan, V., 1968. Conformation of polypeptides and proteins. *Adv. Protein Chem.* 23, 283–437.
- Ramakrishnan, C., 2001. Resonance, pp. 48–56.
- Remaut, H., Waksman, G., 2006. Protein-protein interaction through beta-strand addition. *Trends Biochem. Sci.* 31, 436–444.
- Rathore, R.S., Sumakanth, M., Reddy, M.S., *et al.*, 2013. Advances in binding free energies calculations: QM/MM-based free energy perturbation method for drug design. *Curr. Pharm. Des.* 19, 4674–4686.
- Reddy, M.R., Reddy, C.R., Rathore, R.S., *et al.*, 2014. Free energy calculations to estimate ligand-binding affinities in structure-based drug design. *Curr. Pharm. Des.* 20, 3323–3337.
- Rittinger, K., Budman, J., Xu, J., *et al.*, 1999. Structural analysis of 14-3-3 phosphopeptide complexes identifies a dual role for the nuclear export signal of 14-3-3 in ligand binding. *Mol. Cell* 4, 153–166.
- Schrödinger, L.L.C., 2017. Small molecule drug discovery & biologics suite. New York, NY.
- Schymkowitz, J., Borg, J., Stricher, F., *et al.*, 2005. The FoldX web server: An online force field. *Nucleic Acids Res.* 33 (Web Server issue), W382–W388.
- Shiau, A.K., Barstad, D., Loria, P.M., *et al.*, 1998. The structural basis of estrogen receptor/coactivator recognition and the antagonism of this interaction by tamoxifen. *Cell* 95, 927–937.
- Speltz, E.B., Nathan, A., Regan, L., 2015. Design of protein-peptide interaction modules for assembling supramolecular structures in vivo and in vitro. *ACS Chem. Biol.* 10, 2108–2115.
- Spiliotopoulos, D., Kastiris, P.L., Melquiond, A.S., *et al.*, 2016. dMM-PBSA: A new HADDOCK scoring function for protein-peptide docking. *Front. Mol. Biosci.* 3, 46.
- Stanfield, R.L., Wilson, I.A., 1995. Protein-peptide interactions. *Curr. Opin. Struct. Biol.* 5, 103–113.
- Stein, A., Aloy, P., 2008. Contextual specificity in peptide-mediated protein interactions. *PLOS ONE* 3, e2524.
- Tompa, P., 2012. Intrinsically disordered proteins: A 10-year recap. *Trends Biochem. Sci.* 37, 509–516.
- Toniolo, C., Crisma, M., Formaggio, F., Peggion, C., 2001. Control of peptide conformation by the Thorpe-Ingold effect (C alpha-tetrasubstitution). *Biopolymers* 60, 396–419.
- Trabulo, S., Cardoso, A.L., Mano, M., Pedrosa de Lima, M.C., 2010. Cell-penetrating peptides – Mechanisms of cellular uptake and generation of delivery systems. *Pharmaceuticals (Basel)* 3, 961–993.
- Trellet, M., Melquiond, S.J.A., Bonvin, A.M.J.J., 2013. A unified conformational selection and induced fit approach to protein-peptide docking. *PLOS ONE* 8, e58769.
- Tubert-Brohman, I., Sherman, W., Repasky, M., Beuming, T., 2013. Improved docking of polypeptides with glide. *J. Chem. Inform. Model.* 53, 1689–1699.
- Van Roey, K., Uyar, B., Weatheritt, R.J., *et al.*, 2014. Short linear motifs: Ubiquitous and functionally diverse protein interaction modules directing cell regulation. *Chem. Rev.* 114, 6733–6778.
- Vassilev, L.T., Vu, B.T., Graves, B., *et al.*, 2004. In vivo activation of the p53 pathway by small-molecule antagonists of MDM2. *Science* 303, 844–848.
- Venkatachalam, C.M., 1968. Stereochemical criteria for polypeptides and proteins. V. Conformation of a system of three linked peptide units. *Biopolymers* 6, 1425–1436.
- Yan, C., Xu, X., Zou, X., 2016. Fully blind docking at the atomic level for protein-peptide complex structure prediction. *Structure* 24, 1842–1853.
- Zahiri, J., Bozorgmehr, J.H., Masoudi-Nejad, A., 2013. Computational prediction of protein – Protein interaction networks: Algorithms and resources. *Curr. Genom.* 14, 397–414.
- Zarrinpar, A., Bhattacharyya, R.P., Lim, W.A., 2003. The structure and function of proline recognition domains. *Sci. STKE* RE8.
- Zvelebil, M.J., Thornton, J.M., 1993. Peptide-protein interactions: An overview. *Q. Rev. Biophys.* 26, 333–363.

Relevant Websites

<http://foldxsuite.org.eu/>

FoldX server.

<http://galaxy.seoklab.org/cgi-bin/submit.cgi?type=PEPDOCK>

GalaxyPepDock.

<http://bioserv.rpbs.univ-paris-diderot.fr/services/pepATTRACT/>

pepATTRACT.

<https://www.ccdc.cam.ac.uk/>

The Cambridge Crystallographic Data Centre (CCDC).

<http://www.rcsb.org/>

The Protein Data Bank.

Biographical Sketch



Dr. U.S. Raghavender did his Masters in Physics (specialization in Condensed Matter Physics) from the University of Hyderabad and obtained his PhD in Biocrystallography of Designed Peptides from the Indian Institute of Science (IISc, Bangalore) in 2010. He did his post-doctoral studies in Stanford Research International – Center for Advanced Drug Research (USA), National Center for Biological Sciences (NCBS-TIFR, Bangalore, India) and The Hebrew University of Jerusalem (Israel) in the area of computational studies involving protein-peptide interactions. He has expertise in peptide design, conformational analysis and computational studies of peptide mediated interactions. His research interests include applying computational approaches to studying transient protein-protein and protein-peptide interactions. His interests also span in studying genomes and transcriptomes of medicinally important plants. He is a recipient of award of Young Scientist Scheme of Science and Engineering Research Board (Department of Science & Technology, Government of India).



Dr. R.S. Rathore obtained M. Phil. In 1991 from Devi Ahilya University, Indore and PhD in 1999 from Indian Institute of Science, Bangalore, India, in Structural Biology in the area of X-ray crystallography and peptide *de novo* design. He further pursued postdoctoral work on protein crystallography and peptide design at the University of Texas Medical Branch at Galveston, USA. He is currently working as a Professor in the School of Earth, Biological and Environmental Sciences, Central University of South Bihar, Gaya, India. Rathore earlier worked in University of Hyderabad, India and Schrödinger Inc. (USA). His main interests are protein modelling and design and Free Energy Perturbation method for lead optimizations. He has published 95 articles in peer reviewed journals.

Homologous Protein Detection

Xuefeng Cui and Yaosen Min, Tsinghua University, Beijing, China

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteins carry a wide variety of functions within organisms. For example, antibody proteins can act as virus fighters to neutralize the target function causing a disease; enzyme proteins can act as catalysis to accelerate a chemical reaction by lowering its activation energy; hemoglobin proteins can transport oxygen via arterial blood; fibrous proteins can make up structural components, such as skin, hair, and nails. This is why it is very important to study how proteins function in organisms.

If we could understand how proteins function, we should be able to modify existing proteins or even to design new proteins. The newly designed proteins could help us to fight against cold virus or even Ebola virus, to further accelerate or even to slow down a chemical reaction within organisms. The ultimate goal of such protein designs is to benefit human lives, such as no one would get cold any more, or everyone could drink sea water directly. Unfortunately, we still understand very little about how proteins function, and biologists are still trying hard to study how proteins function in organisms.

It is widely accepted that protein functions are determined by protein structures (Crick, 1958). If we could find proteins with similar structures and similar functions, one could focus on the similarities as hints. Such hints suggest that the similar part of the complete protein structure tends to be the functioning part of the protein. This approach should significantly reduce the search space and hence speed up the study on protein functions. Initially, this work was done manually by expert. However, since biologists have experimentally determined more than 100,000 protein structures (Berman *et al.*, 2000), it is extremely labor extensive to exam the similarities between each pair of proteins manually.

This motivates the computational problem of homologous protein detection that automatically find proteins sharing a common ancestor during the course of evolution. Such homologous proteins tend to share conserved protein sequences, structures and functions. Thus, these proteins should be ideal cases for biologists to study how proteins function. Moreover, computers are much cheaper and faster comparing to human labors. Therefore, the homologous protein detection becomes an irreplaceable step towards understanding how proteins function.

Background

Formally, a protein complex consists of one or more chains of amino acids. Each chain is also called a polypeptide because the amino acids are connected by peptide bonds. For sake of simplicity, a protein in this article is referred to a single polypeptide. Then, a protein can be represented by a sequence of amino acids connected by peptide bonds, or a number of atoms in three-dimensional space connected by peptide or covalent bonds. Detailed explanations of related terminologies are provided as followings.

Amino acid: Amino acids are small organic molecules containing amine ($-NH_2$) and carboxyl ($-COOH$) functional groups, along with a side chain (i.e., the R group), which is used to identify the amino acid as shown in Fig. 1. Moreover, the carbon atom connected to the side chain is referred as the C_α atom, which is marked as a star in the figure. In total, there are 20 kinds of common amino acids identified by 20 kinds of side chains, and each kind of amino acids can be represented by a single character.

Polypeptide: Two or more amino acids form a polypeptide via dehydration reactions, and the polypeptide could be identified by its peptide bonds, which are amide groups as shown in Fig. 2. Each polypeptide has an N-terminus (i.e., the amine end) and a C-terminus (i.e., the carboxyl end). Usually, the amino acids that have been incorporated into polypeptides are also called “residues”, and the residues are counted from the N-terminus to the C-terminus.

Primary structure: A protein is a polypeptide containing more than 50 residues. Recall that a polypeptide is a chain of residues, and each residue can be represented by a single character. Then, the primary structure or simply the sequence of a protein can be represented by a sequence of characters, where each character represents the type of the corresponding residue.

Secondary structure: The secondary structure of a protein indicates local structure patterns formed by hydrogen bonds between residues. The two main types of protein secondary structures are α -helices and β -sheets. The cartoon in Fig. 3 (drawn using PyMOL, Delano, 2002) has been widely used to show the trace of residues forming an α -helix on the left and a β -sheet on the right. It can be seen that the α -helix has a right-handed helix pattern, and the β -sheet has a sheet pattern formed by parallel or anti-parallel β -strands.

Tertiary structure: The tertiary structure or simply the structure of a protein refers to the three-dimensional structure of the polypeptide. Fig. 4 shows an example protein containing mainly α -helices that are connected by loop regions. A simplified representation of a protein structure consists of the three-dimensional coordinates of representative atoms. Typically, the C_α atom can be used as the representative atom for each residue because it approximately defines the three-dimensional location of the residue. Hence, the protein structure can be represented by a sequence of C_α coordinates in three-dimensional space.

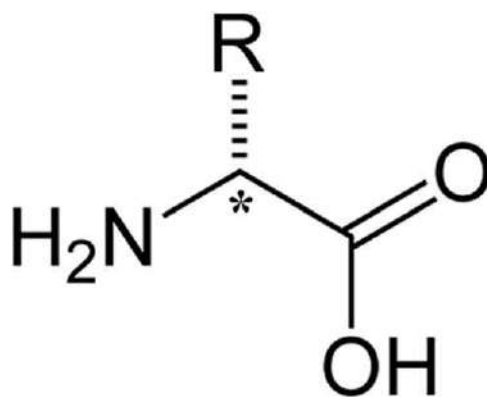


Fig. 1 An amino acid containing a C_α atom (star), an amine group ($-NH_2$), a carboxyl group ($-COOH$) and a side chain (R).

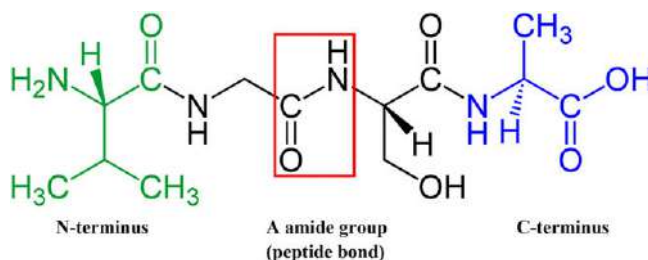


Fig. 2 A polypeptide with five residues.



Fig. 3 Secondary structures of proteins: An α -helix (left) and a β -sheet (right).

Quaternary structure: The quaternary structure of a protein complex refers to the overall three-dimensional structure of a protein complex consisting of two or more polypeptide molecules and possibly other molecules, such as ligand or RNA molecules. Since all the molecules are interacting with each other, they are called a protein complex. [Fig. 5](#) shows one example quaternary structure, but quaternary structures are beyond the scope of this article.

Homologous Protein Detection

In biology, homologous proteins are the ones sharing a common ancestor. Since we do not know the exact evolution histories of proteins, homologous proteins are computationally defined as proteins sharing significant similarities. For example, assuming that each residue is randomly selected from the 20 kinds of residues, the probability to see two protein sequences sharing N common residues is 20^{-N} . If N is a big number, such as 100, the probability becomes very small (i.e., $20^{-100} = 7.9 \times 10^{-131}$). Thus, such similarities are highly unlikely to happen by chance, and such observations should be strong evidences for homologous proteins. Therefore, such proteins are presumed to be homologous proteins in computational biology.

There are mainly two kinds of evidences commonly used to find homologous proteins: protein sequences and protein structures. The computational problem to find sequence similarities between a pair of sequences is called the sequence alignment problem, which is actually a string matching problem. Given two strings, the goal is to find character matchings between the two strings that maximizes a scoring function. Typically, a match could have a score of $+1$, a mismatch could have a score of -1 , and an in-del (i.e., an insertion or a deletion) could also have a score of -1 . Then, it is possible to use a dynamic programming algorithm ([Needleman and Wunsch, 1970](#); [Smith and Waterman, 1981](#)) to find the optimal solution that maximizes the scoring function.

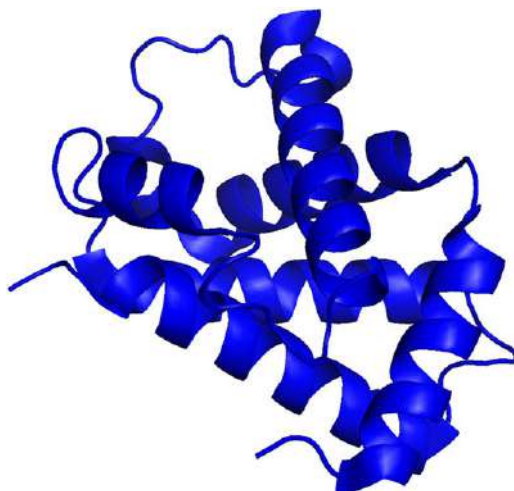


Fig. 4 The tertiary structure of a protein.

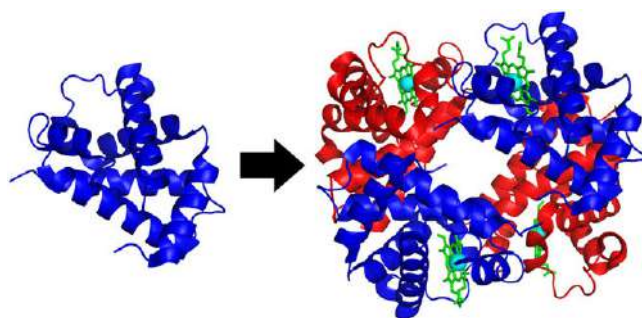


Fig. 5 The quaternary structure of a protein complex: A single chain (left) making up a complex (right).

The problem described above is a simple version of the protein sequence alignment problem (Henikoff and Henikoff, 1992; Altschul *et al.*, 1990; Ma *et al.*, 2002; Remmert *et al.*, 2012). In practice, a biologist might be interested in different things, and hence there are different algorithms designed for different requirements: finding global similarities or finding local similarities, targeting high accuracy or targeting high speed, and so on. Moreover, different scoring functions have also been designed: assuming all aligned residue pairs are independent, assuming the aligned residues pairs are position-specific, assuming the aligned residues pairs are pairwise dependent, and so on.

Finding homologous proteins with structure similarities is much more complicated but more meaningful. Recall that protein functions are determined by protein structures. Sequence similarities work well in practice because proteins with highly similar sequences tend to have highly similar structures (e.g., Fig. 6). Moreover, it is computationally much faster to find sequence similarities than structure similarities. However, there are many homologous proteins sharing little sequence similarities but still sharing high structure similarities (e.g., Fig. 7). Such homologous proteins are called remote homologous proteins because they are conceptually remote as suggested by sequence similarities.

The computational problem to find structure similarities between a pair of protein structures is called the structure alignment problem (Kolodny *et al.*, 2005; Hasegawa and Holm, 2009). It can be divided into two sub-problems: the residue matching problem and the superposition problem. The residue matching problem is similar to the sequence alignment problem. The only difference is on the scoring function, which is now based on the distances between the aligned residue pairs in the three-dimensional space. The superposition problem is to find a rotation and a translation to rigidly transform one protein structure to the other one so that the aligned residue pairs are located closely each other.

The protein structure alignment problem is computationally expensive because its two sub-problems have a chicken-egg relationship: solving the residue matching problem requires the structures to be rigidly transformed first, while solving the superposition problem requires the residues to be aligned first. If we know the answer of one problem, it is computationally efficient (i.e., there exists algorithms running in polynomial time) to solve the other problem. However, if we would like to solve both problems at the same time, the problem becomes NP-hard (Goldman *et al.*, 1999; Li and Ng, 2011). Therefore, many heuristic algorithms are designed based on the idea of sampling local similarities first and then producing global similarities by

Seq 1	XGDVEKGKKIFVQKCAQCHTVEKGGKHKHTGPNLHGLFGRKTGQAPGFTYTDANKNKGITWK	61
Match	GDVEKGKKIF KC QCHTVEKGGKHKHTGPNLHGLFGRKTGQAPG YT ANKNKI W	
Seq 2	-GDVEKGKKIFIMKCSQCHTVEKGGKHKHTGPNLHGLFGRKTGQAPGYSYTAANKNKGIIWG	60

Seq 1	EETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKATNE	105
Match	E TLMEYLENPKKYIPGTMIF GIKKK ER DLIAYLKATNE	
Seq 2	EDTLMEYLENPKKYIPGTMIFVGIKKKEERADLIAYLKATNE	104

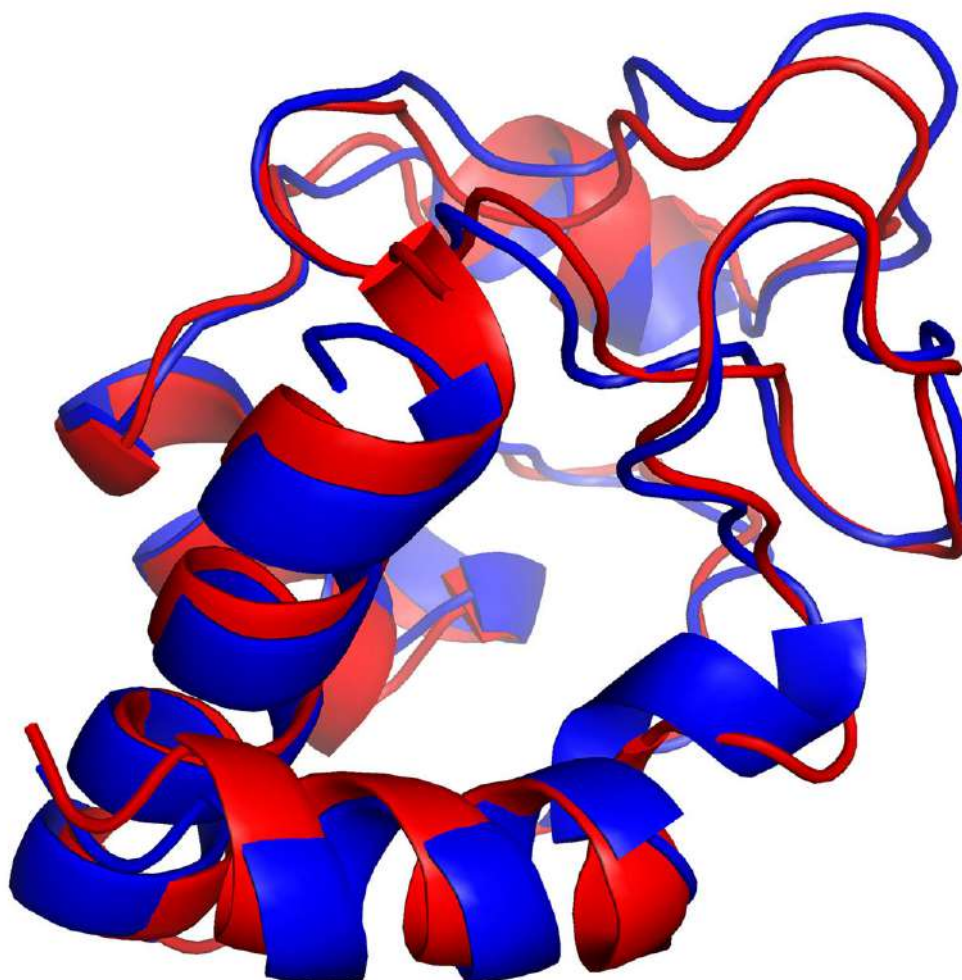


Fig. 6 Two close homologous proteins with similar sequences (top) and similar structures (bottom).

enlarging or combining local similarities (Shindyalov and Bourne, 1998; Krissinel and Henrick, 2004; Zhang and Skolnick, 2005; Cui *et al.*, 2013). Unfortunately, such heuristic algorithms can only find local optimal solutions.

Like the sequence alignment problem, different structure alignment algorithms have been designed to satisfy different interests of biologists: using global structures or focusing on interaction interfaces, targeting high accuracy or targeting high speed, using only structure information or fusing other available information (e.g., secondary structure information), and so on. Some algorithms also try to avoid solving the superposition problem, and to solve the residue matching problem directly (Holm and Sander, 1993; Holm and Rosenström, 2010). The main idea here is to use pairwise distances instead of three-dimensional coordinates to represent protein structures. The trick of such algorithms is that the pairwise distances do not change after any rigid transformations. However, the residue matching problem for the new presentation is still NP-hard, and simulated annealing algorithms have been used to solve the problem with local optimal solutions.

There is a highly related problem called the protein threading problem (Roy *et al.*, 2010; Källberg *et al.*, 2012; Söding *et al.*, 2005), which tries to find similarities between a protein without structure information (i.e., with primarily sequence information) and a protein with structure information. The motivation is the fact that there are significantly more proteins with known sequences than proteins with known structures. The main challenge here is how to compare the cross-modal information. The classical approach is transferring the sequence and the structure space into other spaces, such as the secondary structure space, the

Seq 1	VL	SAADKSNVKACWGKIGSHAGEYGAEALERTFC	SFPTTKTYFPHFDLSHGSAQVKA----	57
Match	VLS	V W K G R S P T F A KA		
Seq 2	VL	SEGEWQLVLHVWAKVEADVAGHGQDIHIRLYKSHPETLEKHDRFKHLKTEAEMKASED		60
Seq 1	---	HGQKVADALTQAVAHMDDLPTAMSALSDLHAYKLRVDPVNFKFLSHCLLVTLACHHP		114
Match	HG V AL	L HA K F S L HP		
Seq 2	LKKHGVTVL	TALGAILKKKGHHEAELKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHP		120
Seq 1	AEFTPAVHASLDKFFSAVSTVLT	SKYR	141	
Match	F K KY			
Seq 2	GDFGADAQGAMNKALELFRKDIAAKYK		147	

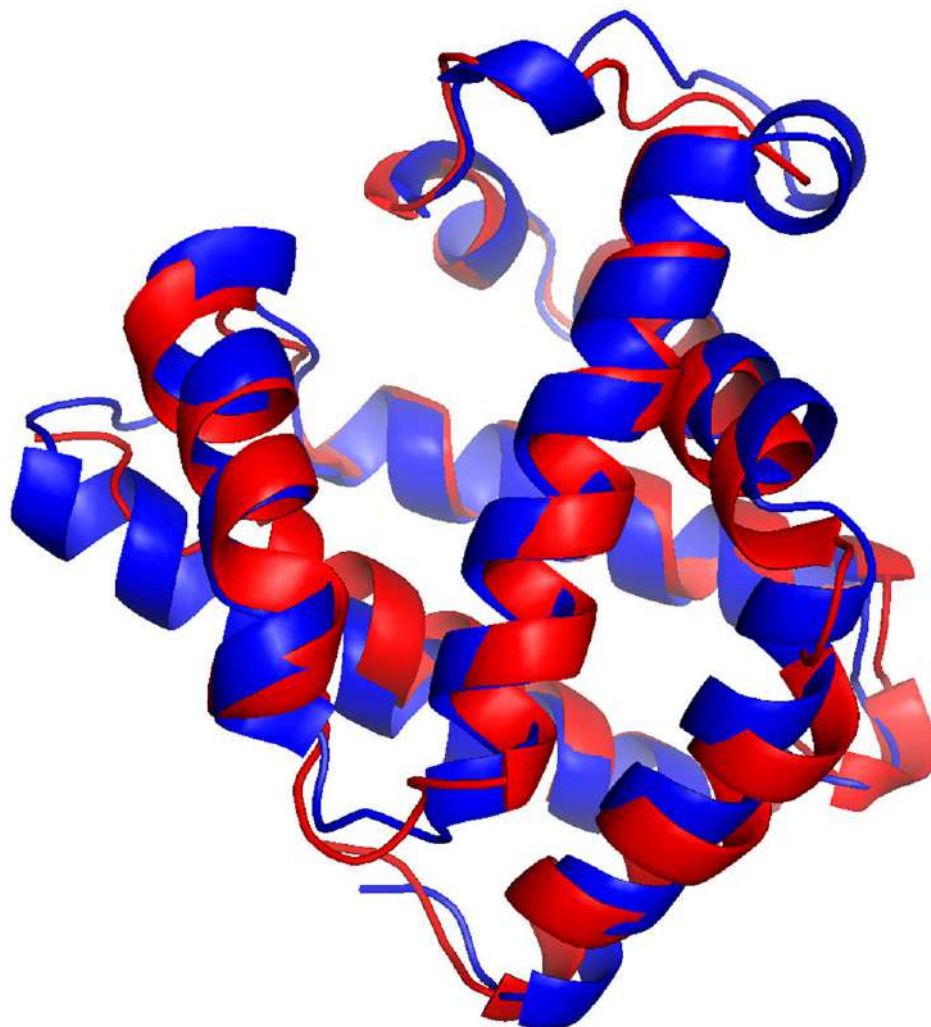


Fig. 7 Two remote homologous proteins with dissimilar sequences (top) and similar structures (bottom).

solvent accessibility space, and so on. Then, similarities can be found in the sequence and the transformed spaces. Recently, cross-modal learning algorithms (Lhota *et al.*, 2015; Cui *et al.*, 2016) have also been used to avoid such space transformations, and to solve the problem in two or more modals directly with improved accuracy.

Other than protein sequences and protein structures, many evidences have also been used to find homologous proteins. For example, if two proteins interact with the same group of proteins, they could also be presumed to be homologous proteins. However, such evidences carry less information comparing to the structure evidences because finding correlations between structures and functions is the key to understand how proteins function. Based on such protein-structure correlations, biologists can first hypothesize how proteins function, and then design biology experiments or molecular dynamic simulations to validate the hypothesis (Wüthrich and Wagner, 1978; Bennett *et al.*, 1984). Although protein-protein interaction evidences cannot directly help to understand how proteins function, they could perform better than sequence or structure information for applications, such as the protein function prediction (Radivojac *et al.*, 2013).

Application: Drug Repositioning

Drug design is both a time consuming and expensive task. The complete drug design process contains several stages: the early drug discovery stage, the pre-clinical research stage, the three clinical trial stages, the application stage and the post-approval study stage. Each stage could take between two and eight years, and the total time required for the complete process is typically between 15 and 25 years. Unfortunately, many patients cannot live that long waiting for the drug release. On the other hand, many patients cannot afford new drugs because of the high drug price due to the high cost of drug design. Taking America in 2013 for example, the total amount of money spent on drug design is approximately 50 billion US dollars, while there are only approximately 25 drugs released. On average, the cost to design a new drug is approximately two billion US dollars. Therefore, it is highly desirable to save both time and money to design new drugs.

It is difficult to save time and the money to design new drugs for several reasons. One reason for the long time is that finding a new drug candidate in the early drug discovery stage is labor intensive. This is actually where computers could help; thus, many algorithms have been designed for this purpose. Even with the help of modern computers and classical algorithms, finding a new drug candidate is still very difficult because of the infinite number of possible candidates. One reason for the high cost is that more than 90% of the drug candidates will fail the clinical stages. For example, it is very difficult to predict if a drug candidate has toxicities or other unacceptable side effects. Such issues can only be discovered by biology experiments or even clinical trials. Since most drug candidates failed the clinical trials, all the money spend in earlier stages will not produce any outcome.

Recently, a key observation opens a new door to solve the drug design problem. Traditionally, each drug was proposed to treat only one disease. However, recent experiments show that each drug could interact with six different proteins, on average, that exists in the human organism. Other than the target protein that is responsible for the disease, most interacting proteins are responsible for side effects or even toxicities. However, there are happy accidents such that the released drug could also interact with a protein that is responsible for another disease. That means, there exists drugs proposing to treat one disease could also treat another disease.

Finding such unknown relationships between drugs and diseases could significantly save time for drug design. Recall that finding a new drug candidate is difficult because of the infinite search space. Scanning the complete drug database for drug candidates reduces the infinite search space to a much smaller and finite search space. As a result, it could take only a couple of days for modern computers to finish the task. Once succeeded, it could save more than eight years of time by skipping the early drug discovery stage, the pre-clinical research stage, and the first clinical stage of drug design. Consequently, this could reduce the drug design time by a factor of approximately two.

Finding such unknown relationships between drugs and diseases could also save a lot of money for drug design. Recall that the high cost of drug design is mainly due to the high failure rate. However, new drug candidates are selected from the released drugs that has previously passed all clinical trials. Then, such drug candidates are highly likely to pass the clinical trials again. Previously, the main reason of the failure rate is because of the unacceptable side effects and toxicities. Now, the main reason becomes the advances in the clinical trials. This should instantly change the high failure rate to a high success rate. As a result, this could reduce the money required to design a new drug by a factor of at least ten.

In computational biology, the problem of finding such unknown relationships between drugs and diseases is called the drug repositioning problem (Ashburn and Thor, 2004; Chong and Sullivan, 2007). One approach to solve the problem is to apply structure alignment algorithms on the interaction interface structures (Gao and Skolnick, 2010; Cui *et al.*, 2015a,b; Naveed *et al.*, 2015). The idea is that if two interaction interfaces have highly similar structures, they tend to interact with the same group of drugs. For example, two novel target proteins, the peroxisome proliferator-activated receptor gamma and the oncogene B-cell lymphoma 2, have been discovered using this method. Both target proteins were first computationally predicted by structure alignment methods, and then experimentally verified by biological methods.

The drug repositioning approach is becoming more and more important for pharmaceutical companies as more and more drugs have been proposed using the approach. For example, Requip was initially proposed as a treatment of Parkinsonian, and then repositioned as a treatment of Restless Legs Syndrome; gabapentin was initially designed as an anti-epileptics drug, and then repositioned as a neuropathic pain reliever drug; and so on. Therefore, the success of drug repositioning has saved not only decades of time, but also tens of billions of US dollars for pharmaceutical companies. In the coming years, more computational drug repositioning methods will be proposed by computational biologists, and more repositioned drugs will be released by pharmaceutical companies. Consequently, more patients would be benefited from these repositioned drugs.

Application: Structure Determination and Prediction

Since protein structures are irreplaceable for understanding protein functions, several experimental methods have been designed to determine protein structures. Due to the importance of the problem, three Nobel Prizes have been awarded to protein structure determination methods: X-ray crystallography (Garman and Schneider, 1997), nuclear magnetic resonance (NMR) spectroscopy (Wüthrich, 1990) and cryo-electron microscopy (cryo-EM, Kühlbrandt, 2014; Callaway, 2015) in 1915, 1991 and 2017, respectively.

Current protein structure determination methods share a common high-level procedure. First, protein samples are purified and pre-processed. Then, experimental data are measured using the protein structure determination device. For example, an X-ray crystallography device produces scatter plots of the differentiation effects caused by the protein crystal; an NMR spectroscopy

device produces atom-atom distances between close atoms of the protein structure; and a cryo-EM device produces two-dimensional projections of the three-dimensional protein structures. Finally, these experimental data are analyzed to build protein structure models (i.e., the three-dimensional coordinates of all atoms). The last step is done computationally as a data science problem to find protein structure models that are consistent with the observed experimental data.

Ideally, the experimental data contain small errors and the true protein structure model could be easily unveiled. For example, the scatter plots of X-ray crystallography could be used to reconstruct the electron density map containing all atoms of the protein structure. Such an electron density map could be imagined as a three-dimensional photo of the protein structure, where each atom is represented by a ball and each bond is represented by two overlapping balls. Thus, the protein structure model can be computed from the electron density map. Unfortunately, only a very small number of X-ray crystallography experiments have produced such ideal-quality protein structure models.

Generally, experimental data are not sufficient to build protein structure models directly; thus, additional knowledge-bases have to be used. For example, the bond lengths and the bond angles (between two connected bonds sharing a common atom) should have very small variances (Evans, 2007; Jaskolski *et al.*, 2007). Moreover, the bond torsion angles (between two bond surfaces defined by three connected bonds) should obey a specific distribution (Laskowski *et al.*, 1993; Davis *et al.*, 2007). Such constraint and restraint knowledge can be used to reduce the model search space to protein-like structures. The goal is to build a protein structure model that is consistent with both the knowledge-bases and the experimental data. Approximately, 30% of the X-ray crystallography experiments have produced such high-quality protein structure models (Berman *et al.*, 2000).

For the remaining 70% of the X-ray crystallography experiments, homology modelling methods are used to produce the protein structure model. The main problem here is the quality of the experimental data. Actually, homology modelling methods are even more popular when processing cryo-EM data because of their low data quality (Lawson *et al.*, 2010). The idea of homology modelling methods is first identifying a homologous protein with a similar structure, and then refining the homologous structure to better fit the experimental data. Here, the key problem is to find a good homologous protein as the template structure. This can be done by employing sequence alignment algorithms, threading algorithms and recently cross-modal learning algorithms. These methods are highly related to the protein structure prediction methods described in the following two paragraphs.

Currently, multiple protein structure determination methods are frequently used because each method has its own advantages and limitations. Specifically, an X-ray crystallography device is able to produce protein structures with the highest qualities but it requires the proteins to be crystalized first, which is difficult or even impossible for large proteins; an NMR spectroscopy device is able to produce dynamic protein structures but the produced structure might contain a lot of errors; a cryo-EM device is able to produce large protein complex structures but the qualities are limited. Moreover, neither method is capable of experimentally determining all protein structures.

This motivates the protein structure prediction problem, and homology modelling methods again play key roles to solve the problem (Moult *et al.*, 2017; Haas *et al.*, 2017). Specifically, given the sequence of a target protein and the knowledge-base of all known protein structures, the first step is to identify a homologous protein structure from the knowledge-base. Then, the homologous protein structure is used as a template to initialize a decoy structure. The final predicted structure is produced by refining the decoy structure to optimize a potential energy function. A typical potential energy function involves the bond forces, the electrostatic forces and the van der Waals forces, and it is usually time consuming to evaluate (Brooks *et al.*, 1983; Webb and Sali, 2014). Since homology modelling methods significantly reduce the model search space, the number of times to evaluate the potential energy function is significantly reduced (Roy *et al.*, 2010; Källberg *et al.*, 2012; Söding *et al.*, 2005; Cui *et al.*, 2016). Therefore, homology modelling methods usually perform much faster than Ab Initio methods.

Protein structure determination and prediction problems are among the most wanted solutions in biology, and breakthrough technologies have been developed recently to advance the existing solutions. The central dogma of molecular biology tells us that the RNA polymerase is responsible for the transcription function, the spliceosome is responsible for the splicing function, and the ribosome is responsible for the translation function. Two Nobel Prizes have been awarded to solving the RNA polymerase structure and the ribosome structure in 2006 and 2009, respectively. Since 2012, cryo-EM opens a new gate to determine protein structures, and it has been used to solve many key protein structures, including the polymerase structure in 2015, playing key functions in organisms. In the future, computational methods will play more and more important roles in structure determination technologies and structure biology.

Discussions

In this article, we have covered the fundamentals of finding homologous proteins and its applications. It has been shown that it is important to find homologous proteins because they carry key information on studying how proteins function. Indeed, homologous protein detection is highly likely the first step in almost all protein function researches. Therefore, finding homologous proteins is a fundamental problem in protein bioinformatics.

Although many researchers have been working on this topic, there are still a lot of open problems waiting to be solved. For example, existing methods work well for homologous proteins within the same family or the same super-family. The main reason is that such homologous proteins tend to have global structure similarities, and this reduces the difficulty of the problem. However, for homologous proteins within the same fold, conserved high-level structure patterns can still be observed by human experts. Due to the lack of global structure similarities, it is more challenging for computational methods to distinguish true

homologous proteins from the false ones. Thus, different computational methods tend to have different results, and the results might disagree with each other. Actually, there is no theoretical guarantee on finding such homologous proteins for any computational methods; thus, homologous protein detection in the fold level remains open.

This article also shows that computational methods are playing critical roles in protein structural biology. Actually, this is not only true in protein structural biology, but also true in life science. Indeed, recent advances in life science is driven by mathematical, statistical and computational methods. It is not difficult to see that the situation will remain the same in the future, and more interdisciplinary talents are needed to accomplish our goals.

See also: Identification of Homologs. Natural Language Processing Approaches in Bioinformatics

References

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215 (3), 403–410.
- Ashburn, T.T., Thor, K.B., 2004. Drug repositioning: Identifying and developing new uses for existing drugs. *Nature Reviews Drug Discovery* 3 (8), 673–683.
- Bennett, W.S., Huber, R., Engel, J., 1984. Structural and functional aspects of domain motions in protein. *Critical Reviews in Biochemistry and Molecular Biology* 15 (4), 291–384.
- Berman, H.M., Westbrook, J., Feng, Z., *et al.*, 2000. The Protein Data Bank. *Nucleic Acids Research* 28, 235–242.
- Brooks, B.R., Brucoleri, R.E., Olafson, B.D., *et al.*, 1983. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry* 4 (2), 187–217.
- Callaway, E., 2015. The revolution will not be crystallized. *Nature* 525 (7568), 172.
- Chong, C.R., Sullivan, D.J., 2007. New uses for old drugs. *Nature* 448 (7154), 645–646.
- Crick, F.H.C., 1958. On protein synthesis. *The Symposia of the Society for Experimental Biology* 12, 138–163.
- Cui, X., Kuwahara, H., Li, S.C., Gao, X., 2015b. Compare local pocket and global protein structure models by small structure patterns. In: *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, ACM*, pp. 355–365.
- Cui, X., Li, S.C., Bu, D., Li, M., 2013. Towards reliable automatic protein structure alignment. In: *Proceedings of the International Workshop on Algorithms in Bioinformatics, WABI*, pp. 18–32.
- Cui, X., Lu, Z., Wang, S., Jing-Yan Wang, J., Gao, X., 2016. CMsearch: Simultaneous exploration of protein sequence space and structure space improves not only protein homology detection but also protein structure prediction. *Bioinformatics* 32 (12), i332–i340.
- Cui, X., Naveed, H., Gao, X., 2015a. Finding optimal interaction interface alignments between biological complexes. *Bioinformatics* 31 (12), i133–i141.
- Davis, I.W., Leaver-Fay, A., Chen, V.B., *et al.*, 2007. MolProbity: All-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Research* 35 (Web Server issue), W375–W383.
- Delano, W.L., 2002. The PyMOL molecular graphics system. *Proteins Structure Function and Bioinformatics* 30, 442–454.
- Evans, P.R., 2007. An introduction to stereochemical restraints. *Acta Crystallographica Section D* 63 (1), 58–61.
- Gao, M., Skolnick, J., 2010. iAlign: A method for the structural comparison of protein-protein interfaces. *Bioinformatics* 26 (18), 2259–2265.
- Garman, E.F., Schneider, T.R., 1997. Macromolecular cryocrystallography. *Journal of Applied Crystallography* 30 (3), 211–237.
- Goldman, D., Istrail, S., Papadimitriou, C.H., 1999. Algorithmic aspects of protein structure similarity. In: *Proceedings of the IEEE 40th Annual Symposium on Foundations of Computer Science*, pp. 512–521.
- Haas, J., Barbato, A., Behringer, D., *et al.*, 2017. Continuous automated model evaluation (cameo) complementing the critical assessment of structure prediction in casp12. *Proteins: Structure, Function, and Bioinformatics* 86, 387–398.
- Hasegawa, H., Holm, L., 2009. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology* 19 (3), 341–348.
- Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America* 89 (22), 10915–10919.
- Holm, L., Rosenström, P., 2010. Dali server: Conservation mapping in 3D. *Nucleic Acids Research* 38 (Suppl_2), W545–W549.
- Holm, L., Sander, C., 1993. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology* 233 (1), 123–138.
- Jaskolski, M., Gilski, M., Dauter, Z., Wlodawer, A., 2007. Stereochemical restraints revisited: How accurate are refinement targets and how much should protein structures be allowed to deviate from them? *Acta Crystallographica Section D* 63 (5), 611–620.
- Källberg, M., Wang, H., Wang, S., *et al.*, 2012. Template-based protein structure modeling using the raptorx web server. *Nature Protocols* 7 (8), 1511–1522.
- Kolodny, R., Koehl, P., Levitt, M., 2005. Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures. *Journal of Molecular Biology* 346 (4), 1173–1188.
- Krissinel, E., Henrick, K., 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography* 60 (12), 2256–2268.
- Kühlbrandt, W., 2014. Microscopy: Cryo-EM enters a new era. *eLife* 3, e03678.
- Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M., 1993. PROCHECK: A program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography* 26 (2), 283–291.
- Lawson, C.L., Baker, M.L., Best, C., *et al.*, 2010. EMDatabank.org: Unified data resource for CryoEM. *Nucleic Acids Research* 39 (Suppl. 1), D456–D464.
- Lhota, J., Hauptman, R., Hart, T., Ng, C., Xie, L., 2015. A new method to improve network topological similarity search: Applied to fold recognition. *Bioinformatics* 31 (13), 2106–2114.
- Li, S.C., Ng, Y.K., 2011. On protein structure alignment under distance constraint. *Theoretical Computer Science* 412 (32), 4187–4199.
- Ma, B., Tromp, J., Li, M., 2002. PatternHunter: Faster and more sensitive homology search. *Bioinformatics* 18 (3), 440–445.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2017. Critical assessment of methods of protein structure prediction (CASP) round XII. *Proteins: Structure, Function, and Bioinformatics* 86, 7–15.
- Naveed, H., Hameed, U.S., Harrus, D., *et al.*, 2015. An integrated structure-and system-based framework to identify new targets of metabolites and known drugs. *Bioinformatics* 31 (24), 3922–3929.
- Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48 (3), 443–453.
- Radivojac, P., Clark, W.T., Oron, T.R., *et al.*, 2013. A large-scale evaluation of computational protein function prediction. *Nature Methods* 10 (3), 221–227.
- Remmert, M., Biegert, A., Hauser, A., Söding, J., 2012. HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods* 9 (2), 173–175.

- Roy, A., Kucukural, A., Zhang, Y., 2010. I-TASSER: A unified platform for automated protein structure and function prediction. *Nature Protocols* 5 (4), 725–738.
- Shindyalov, I.N., Bourne, P.E., 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering* 11 (9), 739–747.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* 147 (1), 195–197.
- Söding, J., Biegert, A., Lupas, A.N., 2005. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Research* 33 (suppl_2), W244–W248.
- Webb, B., Sali, A., 2014. Protein structure modeling with modeller. *Protein Structure Prediction*. 1–15.
- Wüthrich, K., 1990. Protein structure determination in solution by nmr spectroscopy. *Journal of Biological Chemistry* 265 (36), 22059–22062.
- Wüthrich, K., Wagner, G., 1978. Internal motion in globular proteins. *Trends in Biochemical Sciences* 3 (4), 227–230.
- Zhang, Y., Skolnick, J., 2005. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research* 33 (7), 2302–2309.

Mutation Effects on 3D-Structural Reorganization Using HIV-1 Protease as a Case Study

Biswa R Meher, Berhampur University, Berhampur, India

Megha Vaishnavi, Central University of Jharkhand, Ranchi, India

Venkata SK Mattaparthi, Tezpur University, Tezpur, Assam, India

Seema Patel, San Diego State University, San Diego, CA, United States

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

AIDS and HIV

Acquired immunodeficiency syndrome (AIDS) spread by the human immunodeficiency virus (HIV) has become an epidemic worldwide (Sanou *et al.*, 2012). In 2016, it is assessed that about 36.7 million people were living with HIV, 19.5 million people were living with HIV on antiretroviral therapy, and a huge 1.8 million people were newly infected with HIV, and the infection is spreading at a shocking proportion. UNAIDS projections indicate that an additional 50 million people will be freshly infected in the coming decade, if the world doesn't get through to develop a potent therapy/medication (drugs or vaccine). However, remarkable progress against AIDS over the past 15 years has stimulated a global commitment to end the epidemic by 2030 (see "Relevant Websites section").

The lethal virus attacks the human immune system targeting the helper T-cells (specifically CD4⁺ T cells), macrophages, and dendritic cells (Cunningham *et al.*, 2010) reducing the human immunity (Hatzioannou and Evans, 2012). Regardless of vigorous public health efforts and laborious research efforts, AIDS remains a fatal syndrome. Nevertheless, antiretroviral therapy has given a chance to tackle AIDS in part, but due to the clever HIV the goal for complete destruction of the epidemic remains distant. Mutation-induced drug resistance has abolished the clinical effectiveness of most of the FDA-approved drugs administered. So, there is a great demand for developing and designing a potent and less vulnerable drug for antiretroviral therapy. As shown in Fig. 1, HIV is a globular enveloped retrovirus, enclosing dual copies of single-stranded, positive-sense RNA (Ganser-Pomillos *et al.*, 2012). HIV infection starts with the attachment of the matured virus to the host cells containing CD4⁺ receptors and coreceptors CCR5 or CXCR4 with their envelope glycoproteins gp41 and gp120 (Tran *et al.*, 2012). Upon attachment, the host cell membrane and the viral envelope dissolves and the inner genomic content (RNAs) of the virus enters to the host cell. The RNA is then reversibly transcribes to viral DNA by the reverse transcriptase (Le Grice, 2012), which is followed by transcription, translation to form large polypeptide chain (Gag and Gag-pol). The large polypeptide chain is eventually cleaved by the proteolytic events by HIV-pr to form small structural and functional proteins of the virus. Subsequently, with the process of budding and assembly (Bukrinskaya, 2007), the synthesized proteins and part of host cellular membrane form the new virions for next phase of infections to the new CD4⁺ cells.

HIV-1-Protease (HIV-Pr)

HIV-pr is an essential enzyme of HIV replication, and is a vital target for drug design strategies to fight AIDS. It cleaves the Gag and Gag-Pol polypeptides to generate the mature infectious virions capable of CD4⁺ cells infections (Fun *et al.*, 2012). Deactivating the enzyme's function creates immature virions without having the infectious supremacy. Keeping that in mind, several drugs have

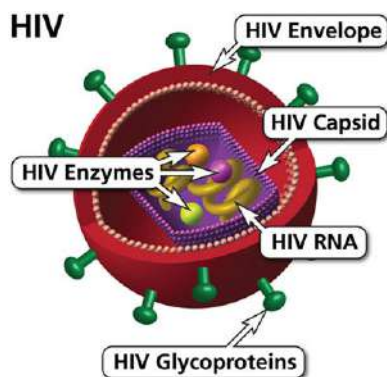


Fig. 1 A schematic structure of a Human Immunodeficiency Virus type 1(HIV-1). Credit: National Institute of Health (NIH).

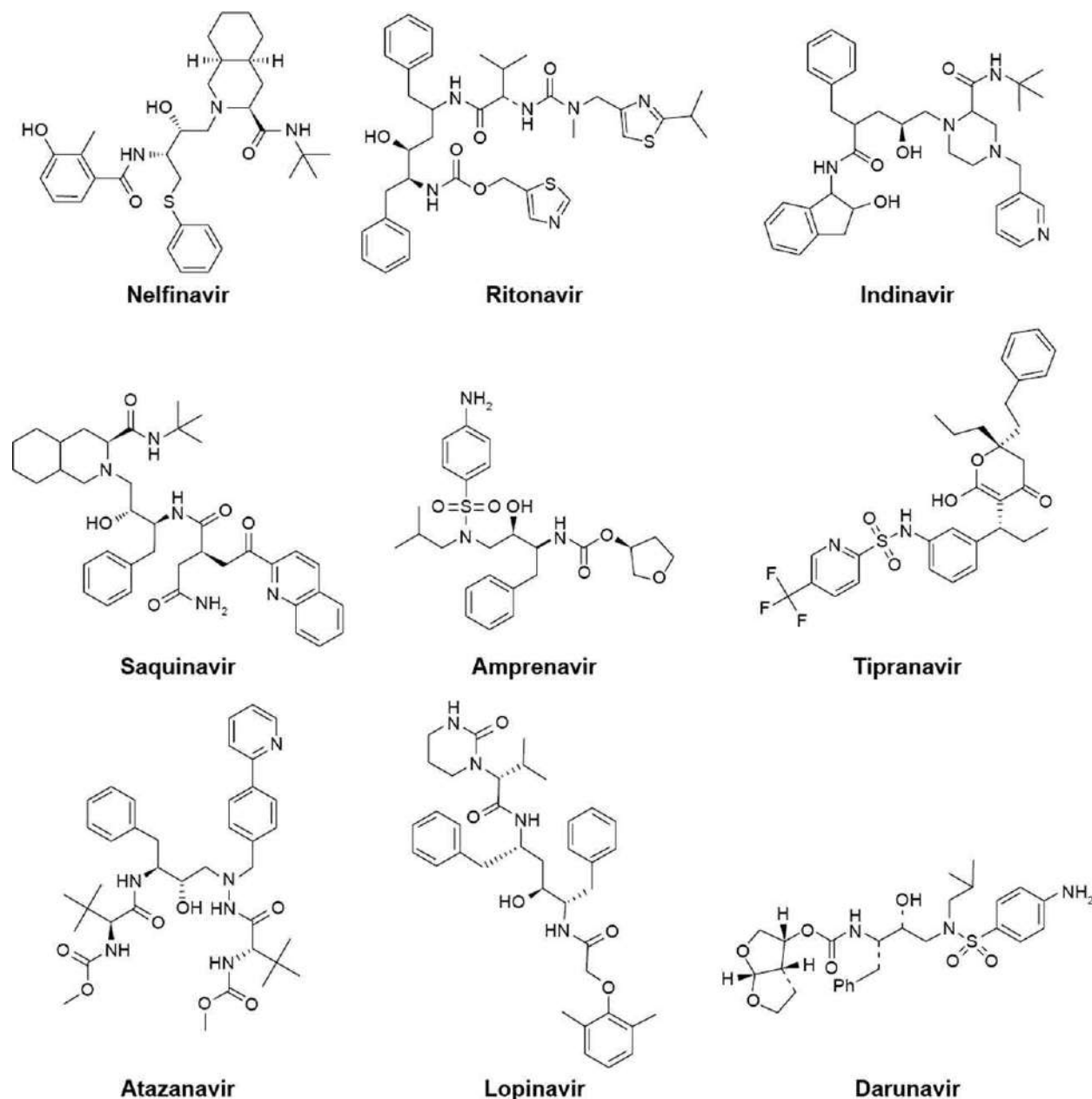


Fig. 2 2D structures of HIV-1 protease inhibitors that are approved by FDA to treat AIDS. Adapted from a research article Arodola, O.A., Soliman, M.E.S., 2015. "Could the FDA-approved anti-HIV PR inhibitors be promising anticancer agents? An answer from enhanced docking approach and molecular dynamics analyses." *Drug Design, Development and Therapy* 9, 6055–6065.

been designed, developed, and finally approved by FDA against AIDS. To date, at least nine FDA-approved protease inhibitors have been rolled out (**Fig. 2**), but none of them is effective after prolonged treatment time due to mutation-assisted drug resistance.

HIV-pr is a homodimeric aspartyl protease with C2 symmetric in the free form (Brik and Wong, 2003), containing 99 amino acids in both of its chain-A and B. HIV-pr residues are numbered as 1–99 for chain A and 1'–99' for chain B. Flap (residues 43–58 and 43'–58'), flap elbow (residues 35–42 and 35'–42'), fulcrum (residues 11–22 and 11'–22'), cantilever (residues 59–75 and 59'–75'), and the active ligand binding site organize different regions of the enzyme (**Fig. 3**). The active site of the protein is formed by dimerization of the two monomers and is crowned by two identical flexible glycine rich flaps. The volume of the active site and size is controlled by dynamics of the flaps (Piana et al., 2002a,b). As a member of the aspartic protease family, the protease contains a catalytic triad (Asp25-Thr26-Gly27) in both the chains keeping functional aspartate residues at the dimer interface. The Asp residues are essential both catalytically and structurally while the Thr and Gly residues functions are still unknown at this time and are buried in the active site (Mager, 2001).

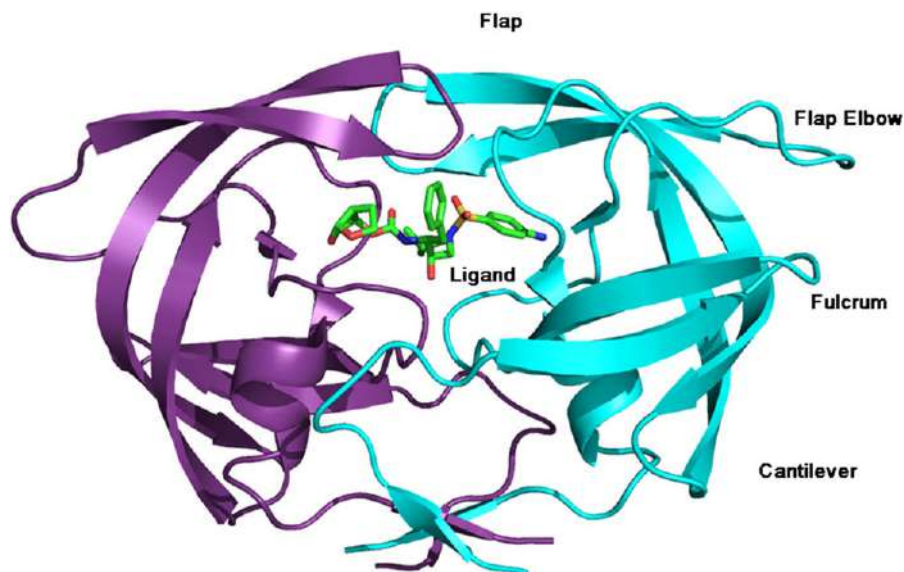


Fig. 3 Schematic representation of the structure of HIV-1 protease (HIV-pr) bound to the ligand in the active site. Important regions of the protein are labeled. The homodimer protein is shown in violet and cyan ribbon structures.

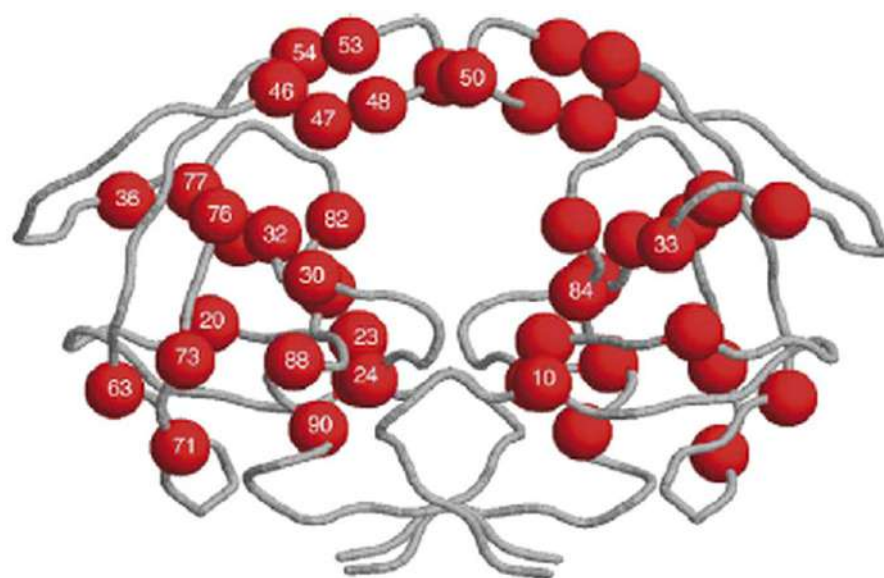


Fig. 4 The structural distribution of the most common mutations associated with drug resistance in the HIV-1 protease. Mutations can occur anywhere in the protease structure. Mutations within the binding cavity are very conservative and operate by distorting the shape of the cavity. Conformationally constrained inhibitors have difficulties in adapting to the altered geometry and lose significant binding affinity. Adapted from Overcoming HIV-1 resistance to protease inhibitors. Drug Discovery Today. Disease Mechanisms | Infectious diseases, vol. 3, No. 2, 2006.

Mutations and Drug Resistance of HIV-Protease Inhibitors

Nevertheless, the HIV-pr-based therapeutic tactics have accomplished reasonable victory, but there are still some hurdles of serious side effects due to mutation-induced drug resistance. Currently, at least more than 50 mutations at near about 30 different codon positions of HIV-pr have been acknowledged. Lists of mutations that occur on the HIV-pr backbone are shown in [Fig. 4](#). A molecular level understanding of drug resistance requires the knowledge of both direct and indirect effects of mutation ([Johnson et al., 2010](#)). The mutant strains are increased in numbers by the drugs' abuse. Taking of numerous drugs has put selective pressure on the HIV leading to mutations and subsequent evolution of resistant variants ([Chen and Lee, 2006](#)). The mutations in HIV-pr are classified as those present near the active site (primary) and those appearing away from the active site (secondary). Both the

mutations affect the ligand/drug binding directly and indirectly by direct or indirect effects (Meher and Wang, 2012; Bandyopadhyay and Meher, 2006). There have been several studies (both experimental and computer simulation) indicating the importance of different mutations for the drug resistance in HIV-pr. Molecular dynamics (MD) simulation based approaches has been utilized by researchers worldwide to understand the HIV-pr 3D-structure dynamics and the drug resistance behavior (Piana *et al.*, 2002a, 2002b; Meher and Wang, 2012; Bandyopadhyay and Meher, 2006; Collins *et al.*, 1995; Hornak *et al.*, 2006; Meagher and Carlson, 2005; Ode *et al.*, 2006; Perryman *et al.*, 2004; Scott and Schiffer, 2000; Toth and Borics, 2006).

HIV-pr 3D-Structure and its Dynamics

HIV-pr 3D-structural dynamics is mainly from the contribution of its flap and flap elbow dynamics, which have the maximum movements for accession of ligands or substrates in its active site region. To date, researchers have analyzed the dynamics of both unliganded and liganded forms of the HIV-pr. In all of the liganded forms, the flaps are pulled in toward the bottom of the active site, leading to a flap curling-in event. In the unliganded enzyme, flaps are shifted away from the active site (Hornak *et al.*, 2006) making the flaps curl out. The contribution of residues in the active site region and flaps to the stability is more distinct in the liganded form than in the unliganded form (Kurt *et al.*, 2003). However, in HIV-pr the anticorrelation movements of the flap-active site distances and fulcrum-flap elbow distances is more notable (Perryman *et al.*, 2004).

Flap Movement and Dynamics

The flexibility of the flap tips (Gly48-Gly52 and Gly48'-Gly52') is known as flap dynamics. It opens and closes the flaps determining the cavity size. The conformational change in the flaps is correlated with structural reorganization of residues in the active site (Torbeev *et al.*, 2011). The mutations in flap region result in adjustment of the nonbonding interactions (van der Waals and electrostatic interactions) between the drugs and protein, subsequently helping drug resistance and rendering the drugs ineffective (Cai *et al.*, 2012). Thus numerous computational studies have made an effort to understand flap dynamics behavior. The effects of mutations on the 3D-conformational dynamics and reorganization of HIV-pr have been considered using MD simulations.

Mutation Effects on HIV-pr 3D-Conformation

JE-2147 (Yoshimura *et al.*, 1999) is an experimental peptidomimetic HIV-pr inhibitor developed by Pfizer (Fig. 5). It was measured to be more effective than other prevailing HIV-pr inhibitors, which is potent against a wide range of HIV-1, HIV-2 strains. JE-2147 retains exclusive resistance profile (Kar and Knecht, 2012a) with two major mutations I84V and I47V. Nevertheless, I84V is common for other related ligands, I47V appears to be very particular for JE-2147 (Bandyopadhyay and Meher, 2006). The influence of I47V mutation is explored with MD simulation. Simulation outcomes showed greater flexibility of the side-chain of mutant Val47 than that of WT Ile47 in chain B of HIV-pr (Bandyopadhyay and Meher, 2006). Structural investigation exposed that the existence of a flexible P2' moiety is significant for the effectiveness of JE-2147 concerning wild-type (WT) and mutant viruses. These data propose that the use of flexible mechanisms may open a new opportunity for designing protease inhibitors with greater efficacy.

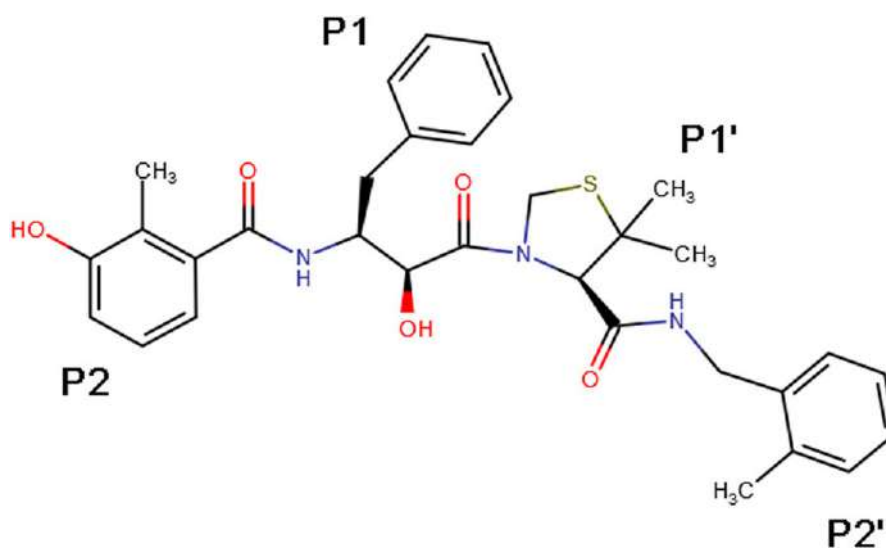


Fig. 5 Molecular structure of the experimental inhibitor JE-2147. Four sites of interactions (P1, P2, P1', and P2') to the protein are labeled. Atoms are shown bonded to each other and are shown in solid lines. Atoms are shown in color as Carbon: Black, Oxygen: Red, Nitrogen: Blue and Sulfur: Gray.

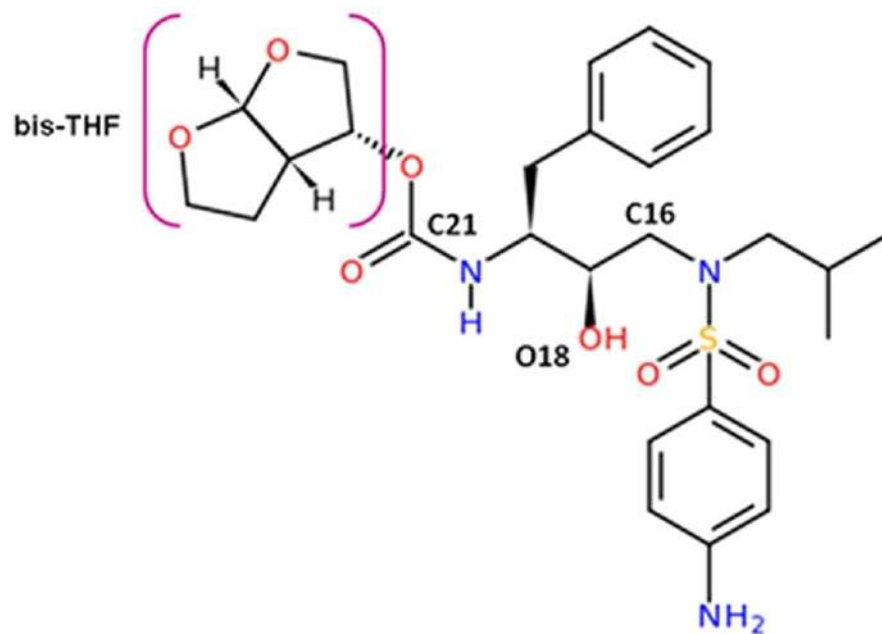


Fig. 6 Configuration of the inhibitor TMC114, with the moiety bis-THF enclosed in square bracket. Atoms playing critical role in the interactions between the inhibitor and the enzyme HIV-pr have been denoted in bold letters. Reprinted from Meher, B.R., Wang, Y., 2012. Interaction of I50V mutant and I50L/A71V double mutant HIV-protease with inhibitor TMC114 (darunavir): Molecular dynamics simulation and binding free energy studies. *Journal of Physical Chemistry B* 116, 1884–1900. With permission from J. Phys. Chem. B and publisher American Chemical Society (ACS).

TMC114, a nonpeptidic compound ended by the bis-tetrahydrofuran (bis-THF) moiety shown in **Fig. 6**, is an enormously effective protease inhibitor (PI) to deal with the drug resistant HIV strains. With the presence of the terminal bis-THF moiety, it slightly differs from its chemical analog, amprenavir. Several studies have shed light on the drug resistance behavior of HIV-pr mutants towards TMC114 (Meher and Wang, 2012, 2015; Kar and Knecht, 2012a, 2012b; Kovalevsky *et al.*, 2006b; Tie *et al.*, 2007; Chen *et al.*, 2010; Vaishnavi *et al.*, 2017).

I47V mutation effect on JE-2147 and TMC114 binding

Vaishnavi *et al.* (2017) have investigated the binding of inhibitor TMC114 and JE-2147 to WT, and I47V mutant HIV-pr with all-atom MD simulations as well as MM-PBSA (molecular mechanics with Poisson–Boltzmann and surface area solvation) calculation. In I47V mutant apo HIV-pr, flap–flap distance was larger than WT or TMC114 and JE2 complexed mutant form (**Fig. 7**). The I47V-mutant complex HIV-pr has less curled flap tips and flexibility compared to WT and the apo mutant I47V. The mutant I47V decreases the binding affinity of I47V-HIV-pr to both the inhibitors (TMC114 and JE2), resulting in a drug resistance; due to an increased volume of the active site. (**Fig. 9**) However, the drug resistance of TMC114 to I47V mutant is heavier than JE-2147. The decrease of the binding affinity for the TMC114 complexed mutant I47V-HIV-pr is resultant of the the decreased electrostatic energy as well as van der Waals energy.

Comparing the apo form of protein WT vs. Mutant

The difference in RMSF (root mean square fluctuations) between the mutant and WT HIV-pr for each residue shows that the maximum changes in RMSF occurs between WT and mutant HIV-pr for the residues in the flap elbows of the two chains (35–42, 40'–42'), the dimerization region (Trp6), part of fulcrum (Gln18), and part of the cantilever region (67–69). MD simulation data from Vaishnavi *et al.* shows that the distance between the flap tip–active site and between flap tips has higher fluctuations for WT-APO and I47V-APO as expected. (**Fig. 8**).

Comparing the complexed form of protein WT vs. Mutant

In the complexed form HIV-pr, the difference in RMSF between WT and I47V-mutant is reduced for most of the residues. It was observed that regions around residues like Pro39-Trp42 (for TMC complex), and Trp6'–Arg8' (for JE2 complex) shows remarkable fluctuations compared to WT with more than 0.75 Å. The relatively larger RMSF of the mutant I47V-complex to its apo-form counterpart is likely to be arising due to larger conformational fluctuations and weaker binding. The distance between Ile50–Ile50' was determined to measure the relative motion of the flap tips. The distance variation between the complexed WT, I47V-TMC and I47V-JE2 HIV-pr was found to be fewer and tighter than apo HIV-pr (**Fig. 6**). Flap dynamics analysis also suggests larger active site volume in case of I47V-TMC and I47V-JE2. The results of these studies indicate that although the Ile50–Ile50' distance was similar in

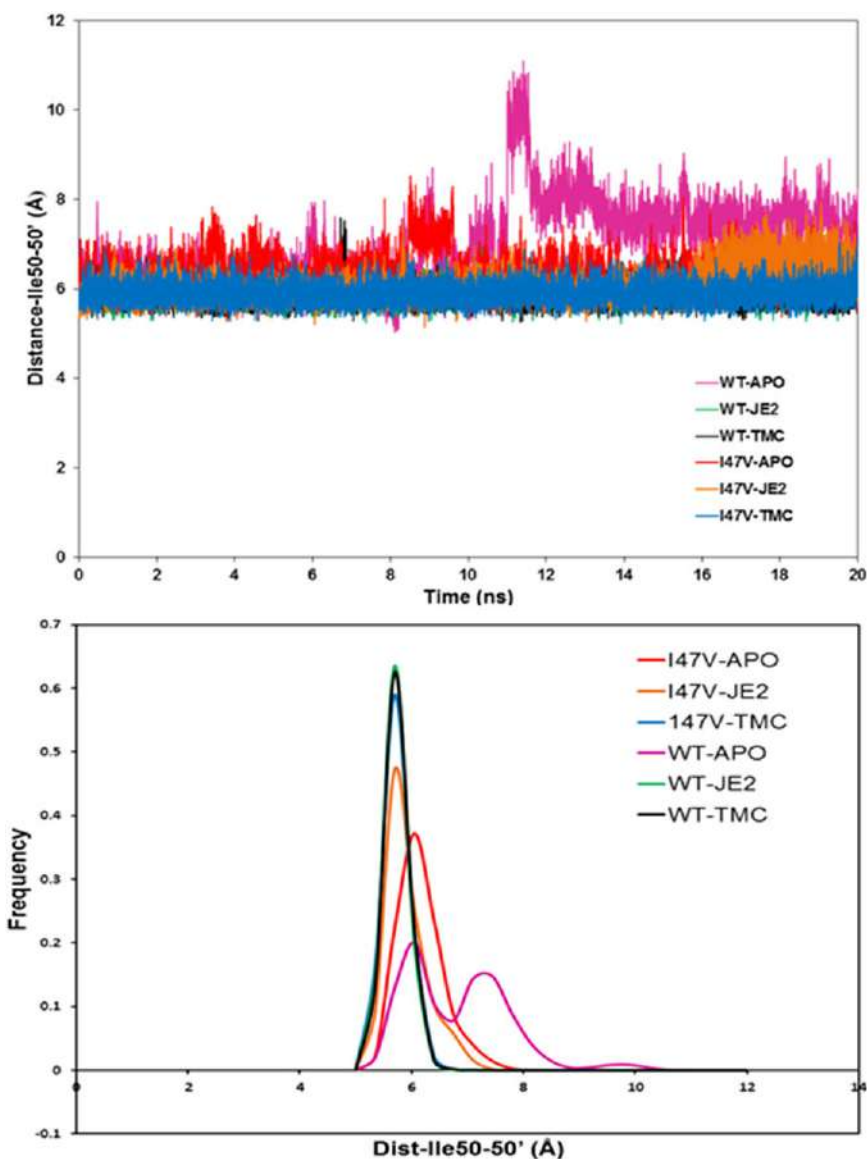


Fig. 7 Time-series (above) and frequency distribution (below) plots for the flap tip-flap tip distances for the HIV-pr WT vs I47V mutant complexes and apo-type.

the complexed HIV-pr there still exist differences in the Asp25'-Ile50' distance, indicating the unique behavior of the two chains of a homodimeric enzyme like HIV-pr.

Molecular mechanism of drug resistance

In the I47V-mutant HIV-pr, the substitution of isoleucine with valine leads to the removal of a methyl group, which is likely to be decreasing the interaction with the central phenyl of TMC114 through C-H... π . Also, it is likely to be shortening the hydrophobic side chain and increasing the size of the active site, resulting in a reduced binding affinity to TMC114 and JE-2147. This change results in a decrease of van der Waals energy between Val47 and TMC114 comparative to the WT. However, for Val47 (47') the change shows a significant decrease in van der Waals energy, which could possibly be due to the lessening of C-H...O interactions between the Val47 side chains and the P2' position of JE-2147 and Val47 side chains and bis-THF (bis-tetrahydrofuran) moiety. The calculated binding free energies of complexes WT-JE2, I47V-JE2, WT-TMC, and I47V-TMC are -31.03 , -29.76 , -34.43 , and -30.73 kcal/mol, respectively, indicating that the binding free energy of WT is higher than the mutant I47V. The binding affinities (ΔG) of I47V-JE2 and I47V-TMC complexes decrease by 1.27 and 3.70 kcal/mol from their WT counterpart, suggesting drug resistance for both the mutant. Residues like Val47' in I47V-JE2 complex directly lowers the ΔG along with other residues like Gly49 and Val82' (Fig. 9).

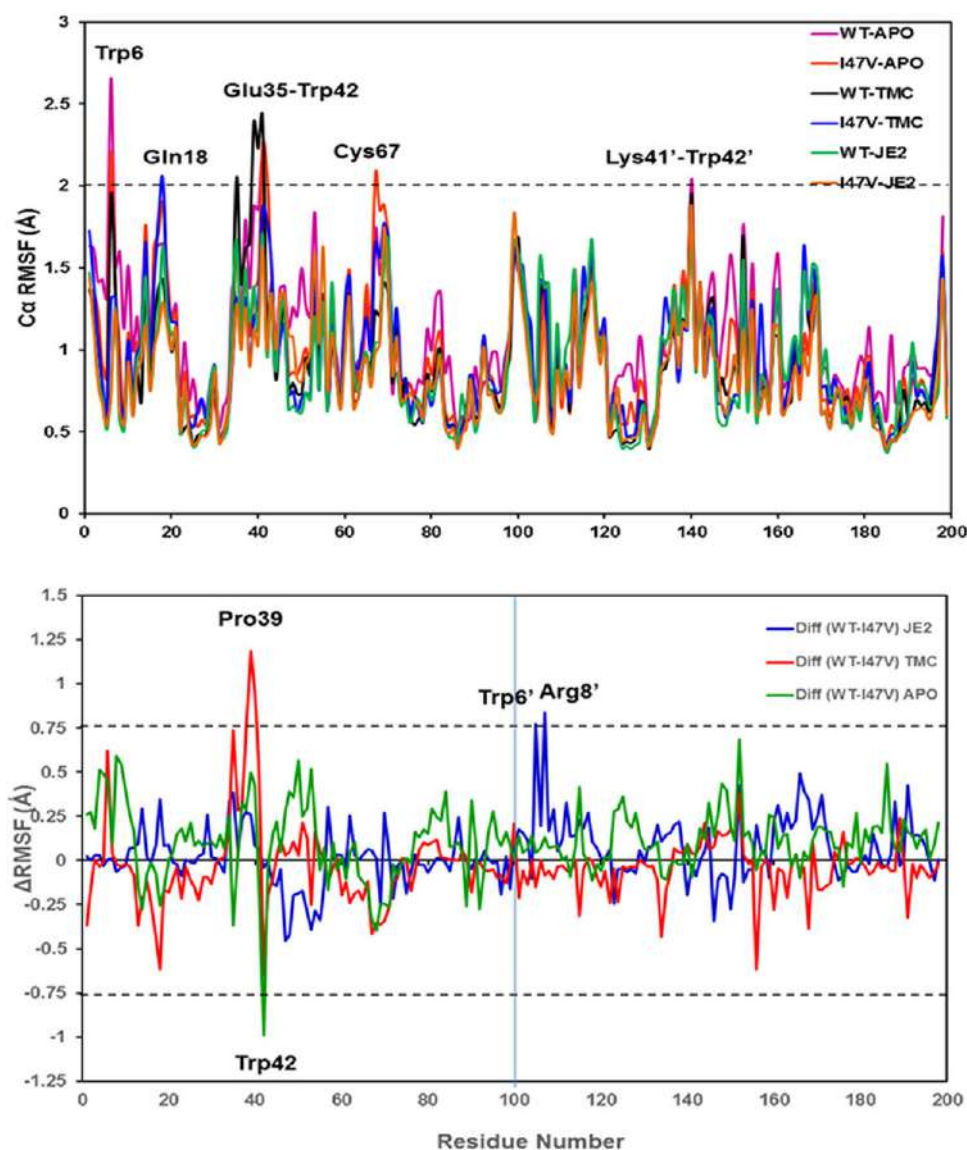


Fig. 8 C α RMSF plot for the HIV-pr WT vs. I47V (above) and the difference is shown (below).

I50V and I50L/A71V mutations effect on TMC114 binding

The single mutation I50V and the double mutation (I50L/A71V) are recognized as two important residue point mutations in HIV-pr which affect protease inhibitors efficacy. Though both the mutations are located at critical region of HIV-pr structure, and their effect on other protease inhibitors have been studied previously, the effect of I50L/A71V mutation on TMC114 binding and the mechanism for drug resistance is still elusive. [Meher and Wang \(2012\)](#) explored the binding of TMC114 ([Fig. 6](#)) to WT, single (I50V) along with the double mutant HIV-pr with all-atom MD simulations and MM-PBSA calculation. The analysis of the apo and complexed HIV-pr indicates that the flap curling and opening events in double mutant I50L/A71V are more stable than WT or I50V. Further, the flap-flap and flap-active site distances also appears to be smaller in I50L/A71V when compared to that of WT or I50V ([Fig. 10](#)), resulting in a compact active site with smaller volume. I50V mutant reduces the binding affinity to inhibitor TMC114, causing drug resistance; while the I50L/A71V double mutant escalates the binding affinity ([Fig. 11](#)). It is remarkable to observe that the I50L/A71V escalates the binding affinity possibly by the stronger binding and better adaptability of the inhibitor TMC114 in its active site.

Comparing the apo form of protein WT vs. Mutant

Difference in B-factors or isotropic temperature factors could offer direct perceptions into the structural variations of HIV-pr in its WT and mutant forms. Each residue difference in B-factors amongst the mutant and WT HIV-pr is highest in the dimer interface region (6, 8 and 6', 8'), flap elbow-A (35, 37, 39–41), and flap-A (49–52). Analysis of the WT and mutant simulations

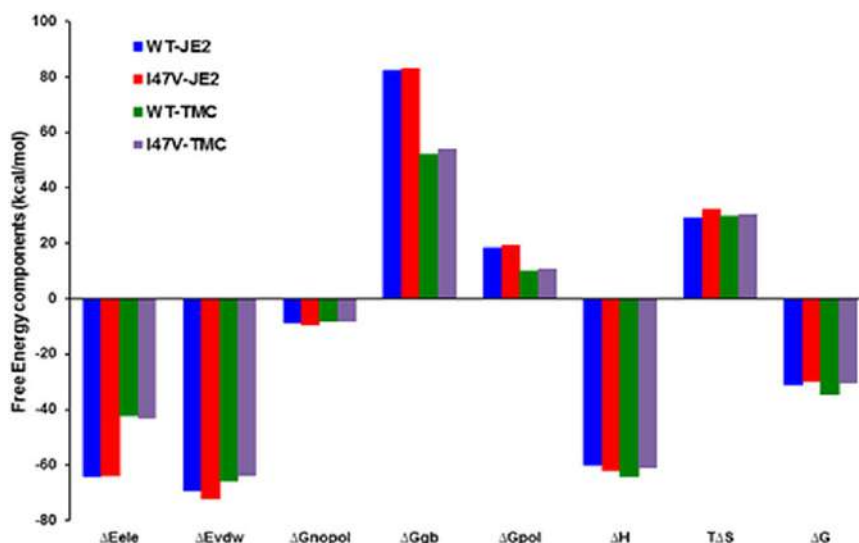


Fig. 9 Energy components (kcal/mol) for the binding of TMC114 and JE-2147 to the WT and I47V mutant: ΔE_{ele} : Electrostatic energy in the gas phase; ΔE_{vdw} : Van der Waals energy; ΔG_{nopol} : Nonpolar solvation energy; ΔG_{gb} : Polar solvation energy; ΔG_{pol} : $\Delta E_{ele} + \Delta G_{gb}$; $T\Delta S$: Total entropy contribution; $\Delta G_{total} = \Delta E_{ele} + \Delta E_{vdw} + \Delta E_{int} + \Delta G_{pb}$; $\Delta G = \Delta G_{total} - T\Delta S$.

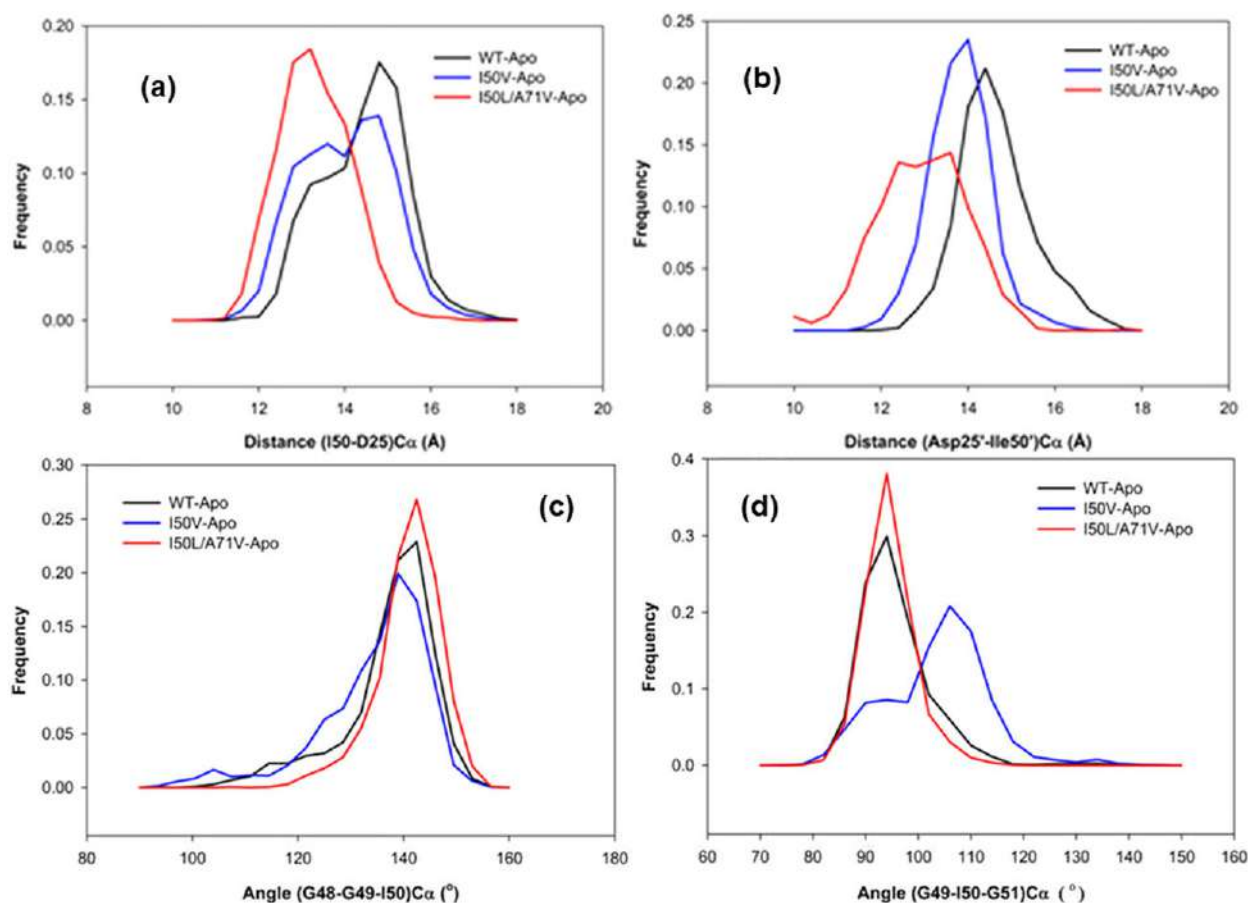


Fig. 10 Variability of histograms for the (a) Ile50–Asp25 distance; (b) Ile50'–Asp25' distance; (c) Gly48–Gly49–Ile50 TriCa angle; and (d) Gly49–Ile50–Gly51 TriCa angle for WT, I50V and I50L/A71V mutants' HIV-pr simulation of the apo-type. Reprinted from Meher, B.R., Wang, Y., 2012. Interaction of I50V mutant and I50L/A71V double mutant HIV-protease with inhibitor TMC114 (darunavir): Molecular dynamics simulation and binding free energy studies. *Journal of Physical Chemistry B* 116, 1884–1900. With permission from J. Phys. Chem. B and publisher American Chemical Society (ACS).

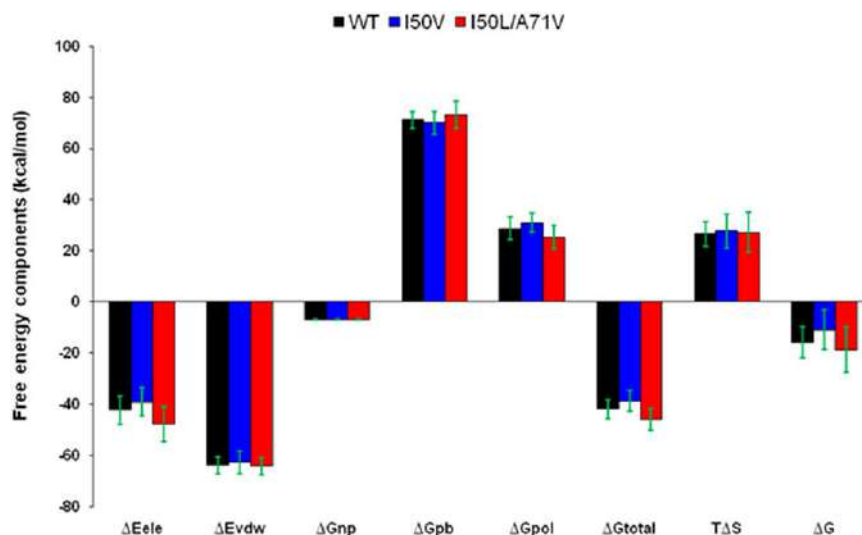


Fig. 11 Energy components (kcal/mol) for the binding of TMC114 to the WT, I50V and I50L/A71V: ΔE_{ele} : Electrostatic energy in the gas phase; ΔE_{vdw} : Van der Waals energy; ΔG_{np} : Nonpolar solvation energy; ΔG_{pb} : Polar solvation energy; ΔG_{pol} : $\Delta E_{ele} + \Delta G_{pb}$; $T\Delta S$: Total entropy contribution; $\Delta G_{total} = \Delta E_{ele} + \Delta E_{vdw} + \Delta E_{int} + \Delta G_{pb}$; $\Delta G = \Delta G_{total} - T\Delta S$. Error bars in green solid line indicates the difference. Reprinted from Meher, B.R., Wang, Y., 2012. Interaction of I50V mutant and I50L/A71V double mutant HIV-protease with inhibitor TMC114 (darunavir): Molecular dynamics simulation and binding free energy studies. *Journal of Physical Chemistry B* 116, 1884–1900. With permission from J. Phys. Chem. B and publisher American Chemical Society (ACS).

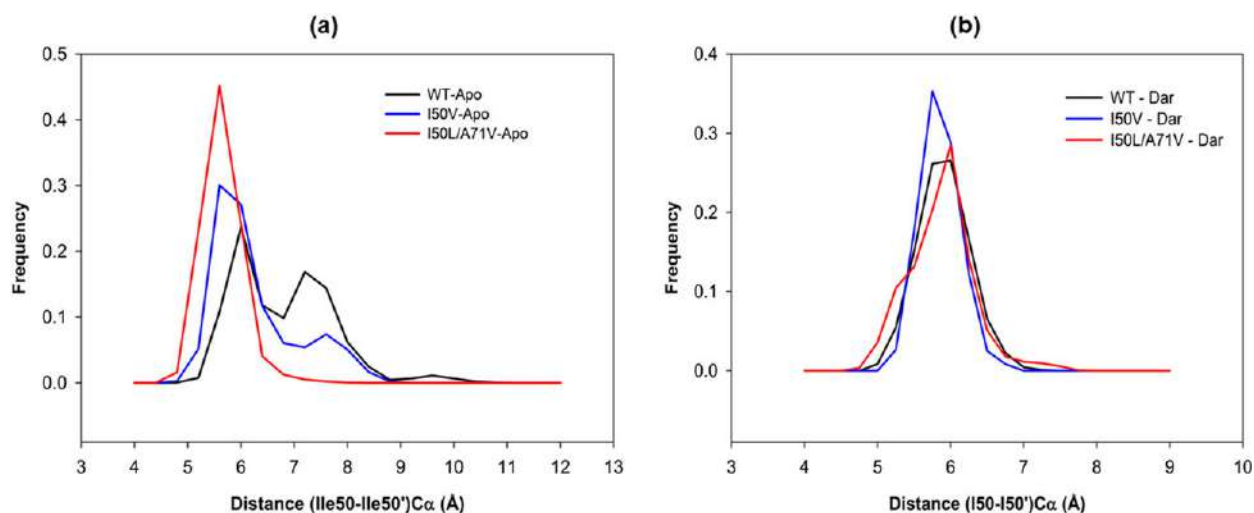


Fig. 12 Histogram distribution of distance between the flap tips in the (a) apo form and (b) TMC114 complexed form for WT, I50V, and I50L/A71V HIV-pr. Reprinted from Meher, B.R., Wang, Y., 2012. Interaction of I50V mutant and I50L/A71V double mutant HIV-protease with inhibitor TMC114 (darunavir): Molecular dynamics simulation and binding free energy studies. *Journal of Physical Chemistry B* 116, 1884–1900. With permission from J. Phys. Chem. B and publisher American Chemical Society (ACS).

demonstrates that the flap–flap distance changes more in the WT and I50V than I50L/A71V (**Fig. 12(a)**), and flap–active site distance is measured to be smaller in I50L/A71V than in WT and I50V (**Fig. 10**). Therefore, both the flap–flap and flap–active site distance results recommend a neighboring movement of flaps in I50L/A71V comparing WT and I50V and possibly modifying the active site size reduced, which could help in improved binding of the TMC114 to the active site region. Improved binding of the TMC114 might be due to the increase in van der Waals (vdw) contacts between the TMC114 and the HIV-pr residues.

Comparing the complexed form of protein WT vs. Mutant

All the complexes (WT vs mutants) show similar fashion of dynamic features from the B-factor perspectives albeit a few exceptions. B-factor difference between the TMC114 complexed and apo HIV-pr for WT and mutant shows that the B-factor is reduced for most of the residues, specifically noticeable in the flap tip and flap elbow regions. It was observed that four regions around

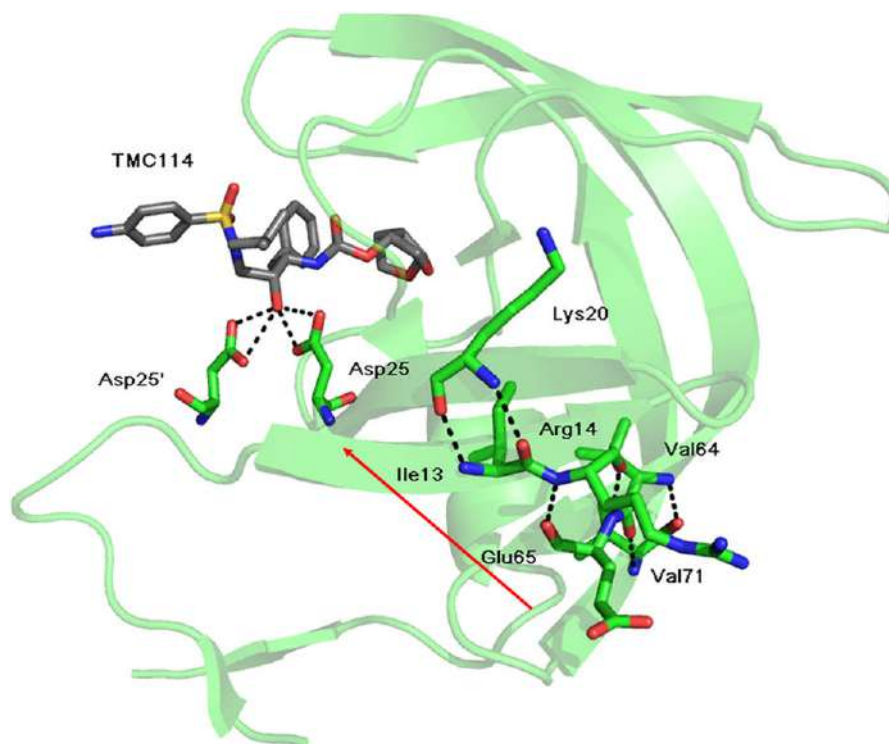


Fig. 13 Graphical representation of H-bond propagation in the double mutant I50L/A71V and Asp25(25')-TMC114 flip-flop interaction. Red color arrow shows the path of propagation.

17 (17'), 41 (41'), 53 (53'), and 70 (70') exert the highest dynamic fluctuations. The slightly reduced B-factor of the I50L/A71V-complex may be described by the comparatively less conformational variations and stronger binding. The reduced flexibility in the inhibitor-binding site directs to the reduction in K_m , resulting in an increase in the affinity of the enzymes for the inhibitor and stronger binding (Zoldak *et al.*, 2004). In order to understand the flap dynamics behavior, the Ile50-Ile50' distance was studied. The difference among the complexed WT, I50V, and I50L/A71V HIV-pr was observed to be fewer and thinner than that of the apo HIV-pr (Fig. 12(b)). The utmost distinct motion for the HIV-pr complex was found to be the side-chain mobility of catalytic Asp25 about the inhibitor TMC114. It shows a flip-flop interaction of the Asp25 OD1/OD2 atoms with the O18 of TMC114 may be originated by the change in H-bonding pattern of the double mutant induced from A71V mutation (Fig. 13).

Molecular mechanism of drug resistance

In the I50L/A71V double mutant, the substitution of isoleucine to leucine, places the methyl group in an altered position, though structure of the side chain is not altered substantially. However, the replacement from alanine to valine adds two methyl groups to the backbone carbon in place of single methyl, which renders side chain bulkier. It is noted that, the change from Ala71 to Val71 in the cantilever region of the HIV-pr has not affected H-bond pattern; however, the H-bond between the residues Arg14' and Glu65' has been reduced (Meher and Wang, 2012). So, it is confirmed that the mutation A71V have no direct influence on the active site conformation and binding affinity. The mutation has allosteric effect on the binding affinity with change in the mobility of active site residues (Fig. 13). The resultant alteration in the conformation of the enzyme may affect its binding affinity to the inhibitor TMC114 to the protease.

In the I50V mutant HIV-pr, the replacement of isoleucine with valine leads to the loss of a methyl group. It lowers the contact with the central phenyl of TMC114, which leads to the decrement in the van der Waals energy between Val50 and TMC114. On the contrary, for Val50' the change displays a substantial decline in van der Waals energy (by 0.54 kcal/mol), which may be due to the falling of C-H...O interactions between the Val50' side chains and the O22 of TMC114. Fig. 14 confirms that the distance of C...O22 (3.4 Å) for I50V-HIV-pr is longer than that for WT and I50L/A71V-HIV-pr (4.1 and 3.6 Å, respectively), which is likely to be the reason for the less binding affinity and drug resistance.

V32I and M46L mutations effect on TMC114 binding

Meher and Wang (2015) investigated the binding of inhibitor TMC114 (Fig. 5) to WT, V32I mutant, and M46L mutant HIV-pr with all-atom MD simulations as well as MM-PBSA calculation. The analysis describes the resistance profile of both the mutants (V32I and M46L) with their 1T (single TMC114 bound alone to the active site region) and 2T (bound to the flap and active site region simultaneously) forms. The average flap-flap distance and flap tip-active site distances are longer for the M46L-2T HIV-pr complex as compare to the WT-1T and V32I-2T complexes suggesting an increased flexibility in M46L-2T, feasibly making the

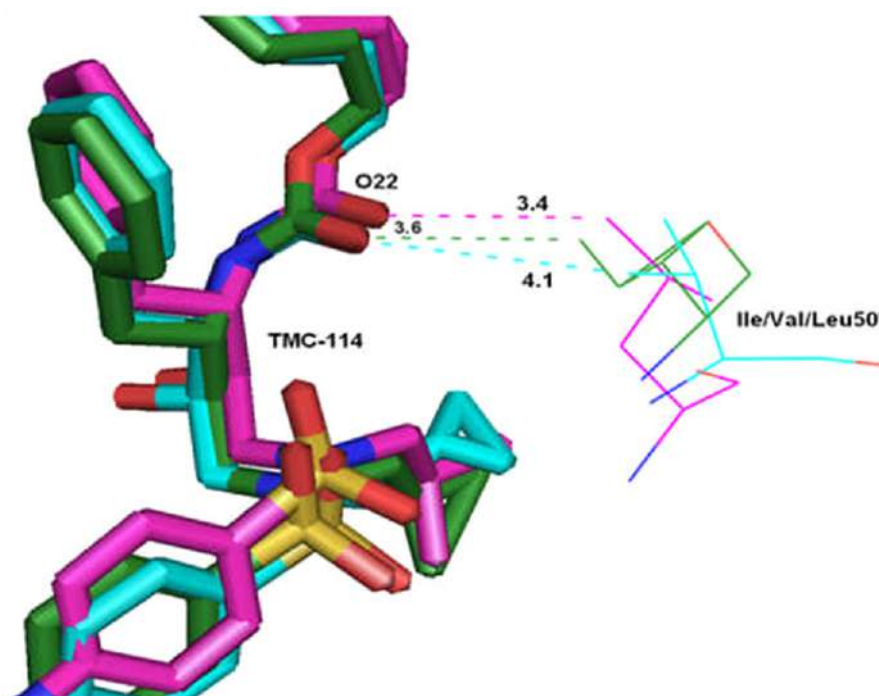


Fig. 14 C-H...O interactions between the inhibitor TMC114 and the flap region amino acids (Gly49, Gly40', Ile/Val/Leu50, and Ile/Val/Leu50'). TMC114 in stick form is colored by the atom type, and residues are denoted as lines (green-WT; cyan-I50V; purple-I50L/A71V). Reprinted from Meher, B.R., Wang, Y., 2012. Interaction of I50V mutant and I50L/A71V double mutant HIV-protease with inhibitor TMC114 (darunavir): Molecular dynamics simulation and binding free energy studies. *Journal of Physical Chemistry B* 116, 1884–1900. With permission from J. Phys. Chem. B and publisher American Chemical Society (ACS).

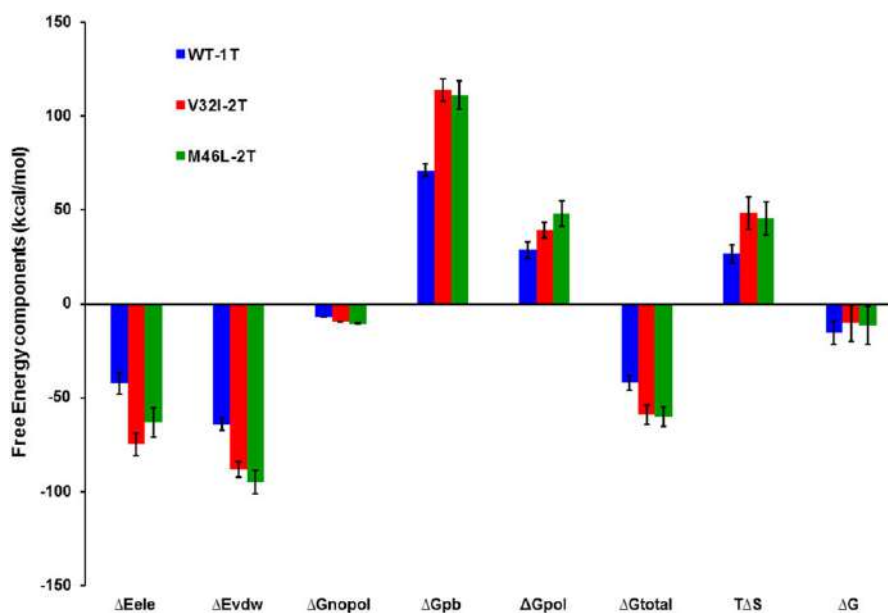


Fig. 15 Energy components (kcal/mol) for the binding of TMC114 to the WT-1T, V32I-2T, and M46L-2T: Eele, electrostatic energy in the gas phase; Evdw, van der Waals energy; Gnp, nonpolar solvation energy; Gpb, polar solvation energy; Gpol = Gele + Gpb; TS, total entropy contribution; H = Eele + Evdw + Gnp + Gpb; $\Delta G = \Delta H - T\Delta S$. The error bars refer to standard deviations (Std.). Reprinted from Meher, B.R., Wang, Y., 2015. Exploring the drug resistance of V32I and M46L mutant HIV-1 protease to inhibitor TMC114: Flap dynamics and binding mechanism. *Journal of Molecular Graphics and Modelling* 56, 60–73. With permission from J. Mol. Graph. Mod. and publisher Elsevier.

active site capacity larger. From the binding free energies of all the complexes, it was gathered that both the 1T and 2T forms of the protease HIV-pr mutants show resistance to the inhibitor TMC114. Also, it came forth that the binding of the TMC114 on the flap region has inconspicuous impact on the binding affinity (Fig. 15).

Comparing the complexed form of protein WT vs. Mutant

The difference of RMSF (root mean square fluctuations) for the whole protein in its complexed form shows that the two mutations (V32I and M46L) cause more conformational changes of the HIV-pr near the flap elbows, fulcrum (Trp6, Ile15-Gly16 and Glu35-Lys41 & Trp6', Glu35' and Arg57'), and cantilever regions (Pro63-His69 and Ala67'), than the WT counterpart. To explore the relative motion of the flap tips, flap tip-flap tip distance was examined, where, the variation between the double bound and single bound V32I-2T and M46L-2T/HIV-pr was found to be different, the former being broader (Fig. 16). Further studies on flap dynamics studies revealed that there is a floppy flaps movement in M46L-2T in comparison to WT-1T and V32I-2T/HIV-pr structures. Flap RMSD analysis indicates that the binding of TMC114 on the flap -B of the mutant structures has only subtle effect on the flap dynamics.

Molecular mechanism of drug resistance

V32I mutation can lead to drug resistance by affecting the interactions between the amino acid side chains and the inhibitor. In the V32I-mutant HIV-pr, the substitution of valine with isoleucine leads to a gain of methyl group. It increases the interaction with the central phenyl of TMC114, and possibly increases the steric hindrance (unfavorable interactions), resulting in a reduced binding affinity to TMC114. This change results in an increase in total entropy contribution ($T\Delta S$) by about 3.42 kcal/mol for V32I-1T and 20.89 kcal/mol for V32I-1T as compared to the WT-1T/TMC114 HIV-pr complex (Meher and Wang, 2015). M46L mutation does not impact the inhibitor binding the active site region, but the atoms of Met46 residue may be forming H-bonds

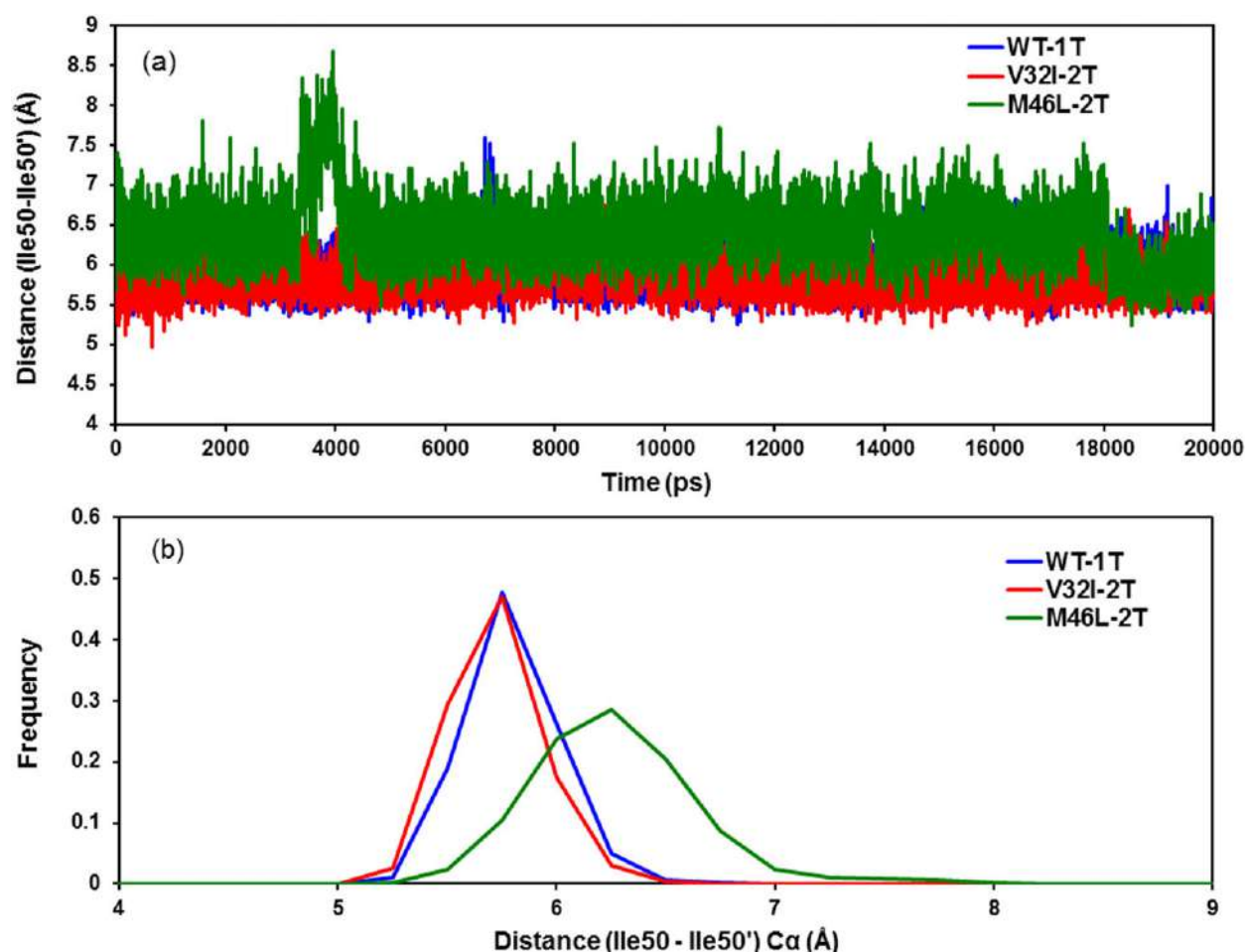


Fig. 16 (a) Time-series plot and (b) frequency distribution plot for the distance between the flap tip (Ile50-Ile50') C atoms for the double bound TMC114 to HIV-1-pr mutants and single bound to WT. Reprinted from Meher, B.R., Wang, Y., 2015. Exploring the drug resistance of V32I and M46L mutant HIV-1 protease to inhibitor TMC114: Flap dynamics and binding mechanism. *Journal of Molecular Graphics and Modelling* 56, 60–73. With permission from J. Mol. Graph. Mod. and publisher Elsevier.

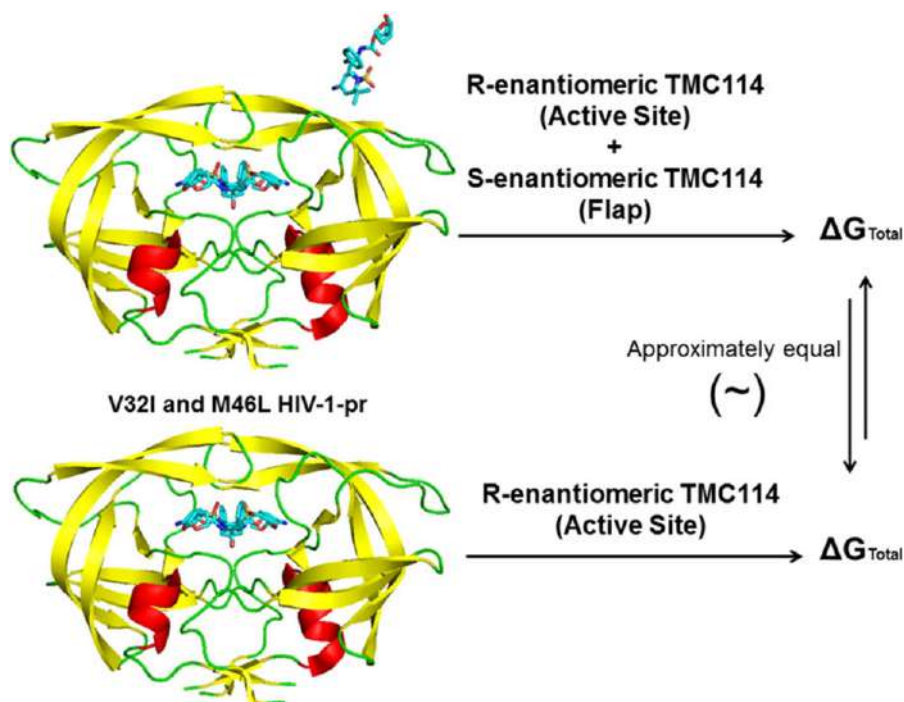


Fig. 17 Comparative schematic view showing the binding of TMC114 in the allosteric site (flap region) for mutants (V32I-2T and M46L-2T) does not influence the binding affinity of the system, which can be compared with the mutants (V32I-1T and M46L-1T). Reprinted from Meher, B.R., Wang, Y., 2015. Exploring the drug resistance of V32I and M46L mutant HIV-1 protease to inhibitor TMC114: Flap dynamics and binding mechanism. *Journal of Molecular Graphics and Modelling* 56, 60–73. With permission from J. Mol. Graph. Mod. and publisher Elsevier.

with substrate analogs (Tie *et al.*, 2005). Hence, M46L mutation can have effect on binding affinity indirectly via the weakened hydrophobic interactions (Kovalevsky *et al.*, 2006a). This alteration results in an increase in total entropy contribution (TAS) by about 2.64 kcal/mol for M46L-1T and 16.83 kcal/mol for V32I-1T as compared to the WT-1T/TMC114 HIV-pr complex (Meher and Wang, 2015). Therefore, the entropy penalty ultimately compresses the binding affinities and elicit the drug resistance for the V32I and M46L mutations. However, binding of TMC114 in the allosteric site (flap region) does not contribute much in the total gain in binding affinity of the system, due to significant entropy loss leading to the lower binding free energies (Fig. 17).

Future Directions

The synergy between mutation and conformational dynamics has exposed the mechanisms of drug resistance. There are potential directions that can be discovered on the basis of the current work.

- 1) Information regarding the placing of a larger group at the P2' position of JE-2147 and also replacement of a group at the position of O18 in the TMC114 structure will be helpful in drug designing. One can proceed for the structural improvement of TMC114 and JE-2147 in vitro and in silico as well based on the combinatorial chemistry.
- 2) Binding of the inhibitor (TMC114) in the flap region does not improve the binding affinity of the system. This understanding will promote the development of drugs/inhibitors targeting mostly to the active site region and not to the flap region. However, targeting to other important regions like flap elbow and fulcrum may help in development of allosteric site inhibitors, which may promote the total gain in binding free energies of the system.

Concluding Remarks

MD simulation has been a tool of tremendous importance and offers generous understandings into the mechanistic features of macromolecular 3D-conformation and dynamics at atomic level. It illuminates the mechanisms of drug binding and resistance profile towards HIV-pr. The study steered to the outcome that 3D-conformational dynamics of HIV-pr is affected by the change in 3D-data coordinates in the protein crystal structure due to mutations that have an important role on HIV-pr conformation and dynamics. Conclusively, the understanding of the molecular basis of drug resistance is a difficult job. However, the result of this study as well as other prior revisions on HIV-pr may offer insightful knowledge with which to design favorable and potent drugs/inhibitors.

Acknowledgement

The authors enthusiastically acknowledge Albany State University, Georgia, USA, SERC at Indian Institute of Science (IISc), Bangalore, India, and Pittsburgh Supercomputing Center (PSC), Pittsburgh, USA for provision of computing facility to accomplish this study. The authors also thankfully acknowledge IISc and Central University of Jharkhand (CUJ) for provision of Library facilities. BRM thanks DBT-BUILDER programme (BT/PR-9028/INF/22/193/2013) at CUJ for financial assistance to carry out some of the work depicted here.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Natural Language Processing Approaches in Bioinformatics

References

- Bandyopadhyay, P., Meher, B.R., 2006. Drug resistance of HIV-1 protease against JE-2147: I47V mutation investigated by molecular dynamics simulation. *Chemical Biology & Drug Design* 67, 155–161.
- Brik, A., Wong, C.-H., 2003. HIV-1 protease: Mechanism and drug discovery. *Organic and Biomolecular Chemistry* 1, 5–14.
- Bukrinskaya, A., 2007. HIV-1 matrix protein: A mysterious regulator of the viral life cycle. *Virus Research* 124, 1–11.
- Cai, Y., Yilmaz, N.K., Myint, W., Ishima, R., Schiffer, C.A., 2012. Differential flap dynamics in Wild-type and a drug resistant variant of HIV-1 protease revealed by molecular dynamics and NMR relaxation. *Journal of Chemical Theory and Computation* 8, 3452–3462.
- Chen, L., Lee, C., 2006. Distinguishing HIV-1 drug resistance, accessory, and viral fitness mutations using conditional selection pressure analysis of treated versus untreated patient samples. *Biology Direct* 1, 14.
- Chen, J., Zhang, S., Liu, X., Zhang, Q., 2010. Insights into drug resistance of mutations D30N and I50V to HIV-1 protease inhibitor TMC-114: Free energy calculation and molecular dynamic simulation. *Journal of Molecular Modeling* 16, 459–468.
- Collins, J.R., Burt, S.K., Erickson, J.W., 1995. Flap opening in HIV-1 protease simulated by ‘activated’ molecular dynamics. *Nature Structural Biology* 2, 334–338.
- Cunningham, A.L., Donaghy, H., Harman, A.N., Kim, M., Turville, S.G., 2010. Manipulation of dendritic cell function by viruses. *Current Opinion in Microbiology* 13, 524–529.
- Fun, A., Wensing, A.M., Verheyen, J., Nijhuis, M., 2012. Human Immunodeficiency Virus Gag and protease: Partners in resistance. *Retrovirology* 9, 63.
- Ganser-Pornillos, B.K., Yeager, M., Pornillos, O., 2012. Assembly and architecture of HIV. *Advances in Experimental Medicine and Biology* 726, 441–465.
- Le Grice, S.F., 2012. Human immunodeficiency virus reverse transcriptase: 25 years of research, drug discovery, and promise. *Journal of Biological Chemistry* 287, 40850–40857.
- Hatzioannou, T., Evans, D.T., 2012. Animal models for HIV/AIDS research. *Nature Reviews Microbiology* 10, 852–867.
- Hornak, V., Okur, A., Rizzo, R.C., Simmerling, C., 2006. HIV-1 protease flaps spontaneously open and reclose in molecular dynamics simulations. *Proceedings of the National Academy of Sciences of the United States of America* 103, 915–920.
- Johnson, V.A., Brun-Vezina, F., Clotet, B., *et al.*, 2010. Update of the drug resistance mutations in HIV-1: December 2010. *Topics in HIV Medicine* 18, 156–163.
- Kar, P., Knecht, V., 2012a. Origin of decrease in potency of darunavir and two related antiviral inhibitors against HIV-2 compared to HIV-1 protease. *Journal of Physical Chemistry B* 116, 2605–2614.
- Kar, P., Knecht, V., 2012b. Energetic basis for drug resistance of HIV-1 protease mutants against amprenavir. *Journal of Computer-Aided Molecular Design* 26, 215–232.
- Kovalevsky, A.Y., Liu, F., Leshchenko, S., *et al.*, 2006a. Ultra-high resolution crystal structure of HIV-1 protease mutant reveals two binding sites for clinical inhibitor TMC114. *Journal of Molecular Biology* 363, 161–173.
- Kovalevsky, A.Y., Tie, Y., Liu, F., *et al.*, 2006b. Effectiveness of non-peptide clinical inhibitor TMC-114 on HIV-1 protease with highly drug resistant mutations D30N, I50V, and L90M. *Journal of Medicinal Chemistry* 49, 1379–1387.
- Kurt, N., Scott, W.R., Schiffer, C.A., Haliloglu, T., 2003. Cooperative fluctuations of unliganded and substrate-bound HIV-1 protease: A structure-based analysis on a variety of conformations from crystallography and molecular dynamics simulations. *Proteins* 51, 409–422.
- Mager, P.P., 2001. The active site of HIV-1 protease. *Medicinal Research Reviews* 21, 348–353.
- Meagher, K.L., Carlson, H.A., 2005. Solvation influences flap collapse in HIV-1 protease. *Proteins: Structure, Function, and Bioinformatics* 58, 119–125.
- Meher, B.R., Wang, Y., 2012. Interaction of I50V mutant and I50L/A71V double mutant HIV-protease with inhibitor TMC114 (darunavir): Molecular dynamics simulation and binding free energy studies. *Journal of Physical Chemistry B* 116, 1884–1900.
- Meher, B.R., Wang, Y., 2015. Exploring the drug resistance of V32I and M46L mutant HIV-1 protease to inhibitor TMC114: Flap dynamics and binding mechanism. *Journal of Molecular Graphics and Modelling* 56, 60–73.
- Ode, H., Neya, S., Hata, M., Sugiura, W., Hoshino, T., 2006. Computational simulations of HIV-1 proteases multi-drug resistance due to non-active site mutation L90M. *Journal of the American Chemical Society* 128, 7887–7895.
- Perryman, A.L., Lin, J.-H., McCammon, J.A., 2004. HIV-1 protease molecular dynamics of a wild-type and of the V82F/I84V mutant: Possible contributions to drug resistance and a potential new target site for drugs. *Protein Science* 13, 1108–1123.
- Piana, S., Carloni, P., Parinello, M., 2002a. Role of conformational fluctuations in the enzymatic reaction of HIV-1 protease. *Journal of Molecular Biology* 319, 567–583.
- Piana, S., Carloni, P., Rothlisberger, U., 2002b. Drug resistance in HIV-1 protease: Flexibility assisted mechanism of compensatory mutations. *Protein Science* 11, 2393–2402.
- Sanou, M.P., De Groot, A.S., Murphey-Corb, M., Levy, J.A., Yamamoto, J.K., 2012. HIV-1 Vaccine Trials: Evolving concepts and designs. *Open AIDS Journal* 6, 274–288.
- Scott, W.R., Schiffer, C.A., 2000. Curling of flap tips in HIV-1 protease as a mechanism for substrate entry and tolerance of drug resistance. *Structure* 8, 1259–1265.
- Tie, Y., Boross, P.I., Wang, Y.F., *et al.*, 2005. Molecular basis for substrate recognition and drug resistance from 1.1 to 1.6 angstroms resolution crystal structures of HIV-1 protease mutants with substrate analogs. *The FEBS Journal* 272, 5265–5277.
- Tie, Y., Kovalevsky, A.Y., Boross, P., *et al.*, 2007. Atomic resolution crystal structures of HIV-1 protease and mutants V82A and I84V with saquinavir. *Proteins* 67, 232–242.
- Torbееv, V.Y., Raghuraman, H., Hamelberg, D., *et al.*, 2011. Protein conformational dynamics in the mechanism of HIV-1 protease catalysis. *Proceedings of the National Academy of Sciences USA* 108, 20982–20987.
- Toth, G., Borics, A., 2006. Flap opening mechanism of HIV-1 protease. *Journal of Molecular Graphics & Modelling* 24, 465–474.
- Tran, E.E., Borgnia, M.J., Kuybeda, O., *et al.*, 2012. Structural mechanism of trimeric HIV-1 envelope glycoprotein activation. *PLOS Pathogens* 8, 1002797.
- Vaishnavi, M., Kumari, S., Misra, A.N., Meher, B.R., 2017. Nature of I47V mutation to inhibitors (JE-2147 and TMC114) binding on HIV-1 protease: Flap dynamics and binding strategies. In: *Proceedings of the International Conference on Frontiers in Chemical Sciences (ICFCS-2017)*, 16–18th March 2017 at Central University of Jharkhand, Ranchi, India.
- Yoshimura, K., Kato, R., Yusa, K., *et al.*, 1999. JE-2147: A dipeptide protease inhibitor (PI) that potently inhibits multi-PI-resistant HIV-1. *Proceedings of the National Academy of Sciences USA* 96, 8675–8680.

Zoldak, G., Sprinzl, M., Sedla, E., 2004. Modulation of activity of NADH oxidase from *Thermus thermophilus* through change in flexibility in the enzyme active site induced by Hofmeister series anions. *European Journal of Biochemistry* 271, 48–57.

Relevant Website

www.unaids.org
UNAIDS.

Biographical Sketch



Dr. Biswa Ranjan Meher is an Assistant Professor at the Computational Biology and Bioinformatics Laboratory, Department of Botany, Berhampur University, Berhampur, Odisha, India. He received his PhD in Biotechnology from Indian Institute of Technology, Guwahati, India. He worked as a Post-Doctoral Researcher at different US universities like the University of Kansas (KU), Kansas; Albany State University (ASU), Georgia; and University of Richmond (UR), Virginia. He also worked as a DS Kothari Post-Doctoral Fellow (DSKPDF) in the Department of Biochemistry at the Indian Institute of Science, Bangalore, India from 2014 to 2016. Before joining his current position, he also served as an Assistant Professor in the DBT-BUILDER Programme at the Centre for Life Sciences, Central University of Jharkhand, Ranchi, India from 2016 to 2017. He has published his research work in reputed international journals like *Journal of Physical Chemistry B*, *Journal of Biomolecular Structure and Dynamics*, *PLoS One*, *Chemical Biology and Drug Design*, *Journal of Molecular Graphics and Modeling*, and others. His current research interests include nanomedicine development and applications, molecular modeling and simulations of biomolecules, understanding the protein conformations through structure network analysis, and application of computational tools to solve biological problems.



Dr. Venkata Satish Kumar Mattaparthi received his PhD in Biotechnology from Indian Institute of Technology Guwahati, Assam, India in 2010. From 2010 to 2012, he was with the Centre for Condensed Matter Theory (CCMT), Department of Physics, Indian Institute of Science Bangalore, India. He is currently working as Assistant Professor in the Department of Molecular Biology and Biotechnology, School of Sciences, Tezpur University, Assam, India. His research interests include understanding the functioning of intrinsically disordered proteins (IDPs), pathways of protein aggregation mechanism in neurodegenerative diseases like Alzheimer, Parkinson, etc., and also the features of protein–RNA interactions using computational techniques. He has published his research in reputed international journals like *Journal of Biomolecular Structure and Dynamics*, *PLoS One*, *Journal of Physical Chemistry B*, *Biophysical Journal*, *Journal of Bioinformatics and Computational Biology*, and others.



Dr. Seema Patel, MSc, MS, PhD, is a graduate of Indian Institute of Technology Guwahati, India and San Diego State University, USA. She has worked in industrial microbiology and then on in silico clinical microbiology. She has served as Assistant Professor in Lovely Professional University, India and as Research Assistant in San Diego State University. She has been in the biomedical research field since 2007, and has written or edited eight books, and published more than 100 papers on microbiology, food science, bioinformatics, enzymology, immunology, endocrine disruption, and correlated topics.

Megha Vaishnavi, has worked as a Junior Research Fellow at the Computational Biology and Bioinformatics Laboratory at the Centre for Life Sciences, Central University of Jharkhand with the DBT-BUILDER Programme under the guidance of Dr. Biswa Ranjan Meher. She is a Post-Graduate in Microbiology from Central University of Rajasthan and graduate in Biotechnology from IPS Academy Indore, Devi Ahilya Vishva Vidyalaya (DAVV). Her research interest includes molecular modelling and simulations of therapeutically significant biomolecules and structure-based drug design.



Dr. Sandeep Kaushik is a passionate computational biologist with a wide exposure of computational approaches. Presently, he is carrying out his research as an Assistant Researcher (equivalent to Assistant Professor) at 3B's Research Group, University of Minho, Portugal. He has a PhD in Bioinformatics from National Institute of Immunology, New Delhi, India. He has a wide exposure on analysis of scientific data ranging from transcriptomic data on mycobacterial and human samples to genomic data from wheat. His research experience and expertise entails molecular dynamics simulations, RNA-sequencing data analysis using Bioconductor (R language based package), de novo genome assembly using various software like AbySS, automated protein prediction and annotation, database mining, agent-based modeling, and simulations. He has a cumulative experience of more than 10 years of programming using PERL, R language, and NetLogo. He has published his research in reputed international journals like *Molecular Cell*, *Biomaterials*, *Biophysical Journal*, and others. Currently, his research interest involves in silico modeling of disease conditions like breast cancer, using an agent-based modeling and simulations approach.

Structural Genomics

Shuhaila Mat-Sharani, Doris HX Quay, Chyan L Ng, and Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Selangor, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Computational means of assigning function have been crucial for the annotation of the numerous genomes that are being sequenced. Without a computational function annotation system, much of the genome sequences generated would remain isolated voluminous non-sensical digital outputs. The primary means of assigning a function to a new open reading frame (ORF) or coding sequence (CDS) has been sequence alignment.

The function of a known protein sequence can be transferred to its homolog once the sequence similarity is considered adequate to infer shared function. The basis of such a function inference is the fact that the function of proteins are dependent on their three-dimensional (3D) structures. The work of Anfinsen (1972) has demonstrated that the information needed for a protein to fold into its functional 3D structure is contained within the sequence of amino acids that make up the polypeptide. The entropic (disordered) state of the different amino acids in a newly synthesized polypeptide chain drives it to fold into the 3D structure. Chothia and Lesk (1986) had determined that sequence similarity at 30% is enough for a protein to share similar folding and retain a similar function. This has thus been used as a cutoff point for many sequence alignments when assigning function from similarity.

Nevertheless, there remain vast numbers of sequences that have uncharacterized function. For many genome annotations, predicted ORFs with no characterized homologs have been assigned the label 'hypothetical protein'. Solving the structure of a hypothetical protein can be enough to identify any folding similarities to structures already available in the Protein Data Bank (PDB) (Dey *et al.*, 2018; Laskowski *et al.*, 2018) and thus point to potentially similar functions. The PDB is the central repository for structural coordinates of biological macromolecules that have been made publicly accessible. Visual examination and computational analysis of a protein's structure may yield clues regarding its function. If function characterization could be potentially achieved by acquiring the structures of the many proteins with yet to be determined functions, then a systematic high throughput means of acquiring such structures would ultimately accelerate such efforts.

Developments and progress in macromolecular structure determination methods, especially in X-ray crystallography, and lately in cryo-electron microscopy, have resulted in more than 130,000 structures being deposited in the PDB. The bulk of this number (approximately 90% in November 2017) were determined using X-ray crystallography. The availability of numerous genome sequences and the progress made in structural biology have therefore led to large scale projects with the specific objectives of determining the 3D structures for proteins encoded in these genomes, such an approach has been termed as structural genomics. For many of these projects, emphasis was placed on the hypothetical proteins that had no known sequence homologs with characterized functions. One of the earliest test case for structural genomics was the MJ0577 hypothetical protein structure of the hyperthermophile *Methanococcus jannaschii* that was solved to a high resolution (Zarembinski *et al.*, 1998). The structure of MJ0577 was found to contain an ATP molecule, which directed the function search of the protein that was later confirmed as an ATPase (Zarembinski *et al.*, 1998).

In this article, we review and discuss computational strategies currently in use for structural genomics, specifically for structures solved by X-ray diffraction of protein crystals. While the protein production and crystallography remain as molecular biology and chemistry problems, the selection of targets to be fed into a structural genomics pipeline and the analyses of the resulting structures are computational biology problems.

Background

Defining Structural Genomics

There are currently different definitions available for the term structural genomics. Several often used definitions are: (i) an effort to describe 3D structure of every protein encoded in a genome via experimental approaches or computational modelling or a combination of both; and (ii) knowledge on the structural organization of the genome thus allowing for all the genes in a genome to be located and characterized. This article will touch on the former, but will only proceed in depth for a subset, which is the solution of structures via experimental means. The second will not be discussed under this topic. In a way, the definitions have overlaps and all are arguably correct; however, these differences and similarities will not be discussed in depth here.

The Extent and Realities of Structural Genomics in Practice

Methods used for determining the 3D structures of proteins have improved tremendously over the past decade. The bulk of the PDB is composed of structures solved through X-ray crystallography with the remainder consisting of coordinates acquired through nuclear magnetic resonance (NMR) spectroscopy and cryo-electron microscopy (Cryo-EM). The X-ray crystallography technique is the most common method used in structural genomics because it can be applied to a wide range of protein sizes and types, including membrane proteins.

An increase in the computational capacity to process the theoretical folding of protein sequences into their 3D structures means that the generation of 3D structures for every protein in a genome is moving closer towards reality. Nevertheless, such capability is not expected in the foreseeable future. However, even for a wholly computational structural genomics approach, it may not be necessary to compute all the CDS in a target genome because for some proteins, an experimentally solved structure may already exist (Baugh *et al.*, 2013).

Similarly, it may not be necessary to experimentally determine the structure for every CDS in a genome because they are already available in the PDB, either for the same organism from another work, or as a homolog from another source. The constraints for an experimental structural genomics approach is grounded in the reality of costs and other limitations that include but are not limited to: the ability to clone, express, purify and crystallize the protein; followed by a requirement for the crystal to diffract and the data to be processable. This process of molecular biology by attrition can result from an initial starting point being thousands of CDS ending with perhaps solved structures that number in the hundreds or even less (Franklin *et al.*, 2015).

The structural genomics process gradually reduces the initial number of target genes acquired from the genome sequence via a systematic process up to the point of coordinate data acquisition for the encoded proteins. As such, this process can be made into a pipeline (Fig. 1) that begins by taking in all the predicted CDS from the genome sequence and proceeding with steps that can then be defined specifically to fit a project's requirements and objectives. For example, for many groups, a logical first step is to identify CDS that code for possibly insoluble proteins; in practice, this is done by identifying membrane proteins (Fig. 1). The next step may be to filter the CDS for those that have very similar examples already available in the PDB. The cut-off for similarity can vary depending on the interests and requirements of the project.

The filtering process can also be intervened from the start to begin with a smaller subset; for example, only CDS that have been annotated as hypothetical proteins can be selected for input into the structural genomics pipeline. Other examples would be to target for a selection that may be of existing focus in a research group – for example, potential drug targets (Franklin *et al.*, 2015; Fedorov *et al.*, 2007). Nevertheless, it must be noted that even for sequences that were initially targets in a structural genomics pipeline, but do not end up as crystallographic structures, may actually have varying degrees of useful functional characterization as a result of having passed through the pipeline (Ahmad *et al.*, 2015; Yusof *et al.*, 2016; Hashim *et al.*, 2013; Ramli *et al.*, 2013).

Although we do not discuss structures solved by NMR and cryo-EM, switching the experimental method that the pipeline feeds from protein crystallography to either NMR, cryo-EM or a combination of all three methods can also be carried out. Thus structural genomics in the present context can be an integration of all available methods to acquire 3D structure data. Similar to the case of filtering the ORFs for the purpose of crystallography, the pipeline can be redirected to identify targets for these other methods – such as identification of smaller molecular weight proteins that would be amenable to structure solution by NMR.

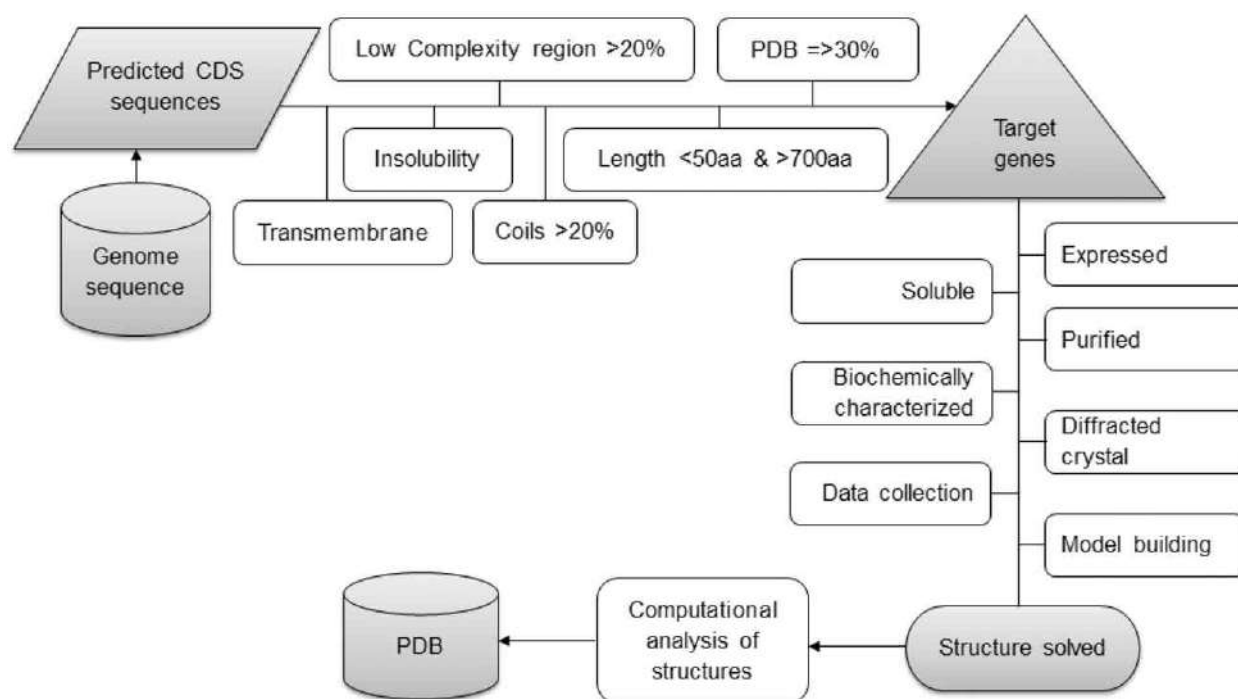


Fig. 1 The overall structural genomics process begins with computational filtration to get target genes and proceeds to solved structures via experimental means. The general filtration begins by removing genes that have transmembrane regions, are insoluble, have more than 20% of low complexity regions and coils. This is followed by removing predicted CDS with lengths less than 50 or more than 700 amino acids and have more than 30% sequence identity with entries in the PDB database.

Systems and/or Applications

A pipeline to process a genome sequence for the purpose of structural genomics can be assembled using proven tools (Table 1). Due to the potential volume of sequences that will be analysed, a local or fit for purpose installation of the programs required would be ideal. This would thus prevent the pipelines engineered for such projects from overloading publicly available servers at the expense of other users. Nevertheless, when such a practice is not possible, care and due consideration of other users of the service should be exercised when scheduling input submissions. As discussed previously (Section “The Extent and Realities of Structural Genomics in Practice”), the starting point of the pipeline would vary depending on the requirements and capacity of the project.

A filter for insoluble proteins, which would be a logical feature and starting point for many projects, would obviously be unsuitable if the project were to actually include the solution of membrane proteins among its objectives. In order to perhaps maximize the return of the project in terms of finding novel folds or characterizing undescribed functions for hypothetical proteins, some projects would opt to further restrict the targets to sequences that are unrepresented in the PDB. This is usually done by employing a 30% sequence identity cutoff or dipping as low as 25% identity (Nair *et al.*, 2009).

For perhaps many projects, these two initial steps of screening for membrane and membrane associated proteins would be adequate to proceed on to the process of elimination from the pipeline by experimental attrition. However, it is also possible to specifically filter for a subset of sequences that contain either N-terminal or C-terminal associated membrane insertions. This is done by identifying whether the first and last thirty residues could potentially be membrane embedded by plotting hydrophobicity plots or predicting transmembrane helices for these sections of the sequence for every CDS in the genome (Marsden *et al.*, 2007). Once identified, these sequences are then cloned and expressed without the predicted transmembrane anchor. Although this does not guarantee solubility, this does increase the initial set of targets while at the same time allowing for the structures of membrane anchored proteins to be solved.

At this stage, the project can then proceed to the non-computational steps highlighted as in Fig. 1. A computational component can then be integrated again at the point where the structure data is available in PDB format. This second phase of the computational analysis will include structural similarity comparisons and in depth structural analysis. It is common practice to identify fold similarities for a solved structure once the final structure file has been acquired. Here we will discuss aspects of computational analysis on the structure but will exclude discussion of in depth visual scrutiny of the structure using molecular graphics.

In most cases, a fold similarity search, such as by the DALI program (Holm and Laakso, 2016) is sufficient to identify other structural homologs. There are however hypothetical proteins that have no similarity to known folds or are matched to structures of proteins with uncharacterized functions (Nadzirin and Firdaus-Raih, 2012). In such cases, the lack of fold similarity will not allow for association with any known structural features than can in turn provide clues as to possible function to be made. In such a situation, computer programs that can carry out structural analysis independent of any homology requirements can be used.

There are several computer programs that work by identifying similarities of 3D substructures or motifs that can bypass the necessity to have fold similarity in order to make a functional association. Many of these methods are able to search and/or

Table 1 Selected programs and their applications that can be integrated into a structural genomics pipeline. A listing of the seven programs or web servers for sequence alignment, transmembrane prediction, protein solubility prediction, coils region prediction, low complexity region prediction and a protein structure database

Function	Program/server	Method/content	URL
Sequence alignment	BLAST	Protein sequence alignment to protein sequence in PDB database.	https://blast.ncbi.nlm.nih.gov/Blast.cgi
	MUSCLE	Iterative alignment of protein sequences to get conserved residue/motif.	https://www.ebi.ac.uk/Tools/msa/muscle/
Transmembrane prediction	TMHMM	Predicts TMs using HMMs (edit this) – just providing an example.	http://www.cbs.dtu.dk/services/TMHMM/
Protein solubility prediction	PROSOII	Predict solubility based on a classifier by two-layered structure in which the output of a primary Parzen window model for sequence similarity and a logistic regression classifier of amino acid k-mer composition serve as input for a second level logistic regression classifier to exploiting subtle differences between soluble proteins from TargetDB and the PDB and notoriously insoluble proteins from TargetDB.	http://mbilij45.bio.med.uni-muenchen.de:8888/prosolII/prosolII.seam
Coils regions prediction	COILS/Coiled-coil prediction	Predict coils regions using MTK and MTIDK matrices both assign high probabilities to known coiled coils segment.	https://embnet.vital-it.ch/software/COILS_form.html https://npsa-prabi.ibcp.fr/cgi-bin/primaln_lupas.pl
Low complexity region prediction	SEG	Predict low complexity region by the SEG algorithm represent compositionally biased regions based only on residue composition and taking into account the improbability of appearance of such sequences.	http://mendel.imp.ac.at/METHODS/seg.server.html
Protein structure database	Protein Data Bank (PDB)	A crystallographic database for the three-dimensional structural data of large biological molecules, such as proteins and nucleic acids.	https://www.rcsb.org/pdb/

compare directly a query arrangement (Debret *et al.*, 2009; Nadzirin *et al.*, 2012, 2013) against a database or take a query structure and compare it against a database of motifs (Nadzirin *et al.*, 2012). There are also services that allow for annotation of specific sites such as binding sites (Konc and Janezic, 2017).

Analysis and Assessment

The need to experimentally determine a structure can be made based on several factors. In situations where detailed information of the differences for atomic level interactions are required, such as for drug development applications, an experimentally solved structure is necessary. It is therefore not uncommon for structural genomics projects to target not only CDS that have less than 30% sequence identity to examples in the PDB but also proteins that are deemed as important where mechanistic details at the atomic level can still be different even though the proteins are homologous. However, for applications where the investigator would like to select residues for site directed mutagenesis as part of a larger characterization effort, then high quality predicted models may suffice (Bhattacharya *et al.*, 2007).

We provide an example where the sequence for an arginase acquired from the genome of *G. antarctica* (Firdaus-Raih *et al.*, 2018), a psychrophilic yeast, was predicted and compared to the structure that was solved by molecular replacement of the X-ray diffraction data (PDBID: 5ynl). The sequence identity between the *G. antarctica* arginase and the reference used for molecular replacement, a human arginase (PDBID: 4hze), is 45%. In this particular case, the decision was made to solve the structure of the arginase despite the sequence identity being higher than the 30% cutoff point used for most of the other targets in the project. It was deemed necessary to acquire a structure that could provide atomic level details of the function as they may be specific to yeasts, psychrophiles or simply unique to *G. antarctica*.

To make a comparison of the differences that could arise from a computationally predicted model, we compared the solved structure to one that was predicted using SWISS-MODEL (Biasini *et al.*, 2014). As expected, the prediction was able to generate the correct fold with discernible disagreements between the two structures only in the loop regions with the compared structures having an RMSD value of 1.06 Å. When the computed model was compared to the template, the differences were much lower at an RMSD of 0.31 Å. It is clear that the model is biased towards the template than the experimentally determined structure (Fig. 2).

We then examined how much an effect the modelling had on the side chain arrangements when the model was compared to the template and the experimental structure. For this comparison, eleven residues that have been reported to be involved in metal coordination and arginine binding were selected. As observed for the folding, it is clear that the predicted model is more biased to the reference used (Fig. 3). Nevertheless, with the exception of D194 and E288 – two of the putative arginine binding residues – the arrangement of side chains for the other nine residues were generally similar and could be seen as having acceptable overlaps. In E288, the side chains ended up in opposite directions despite the C-alpha position generally overlapping. For D194, the general position of the residues in the experimental structure and model were the same, but neither the side chain or the C-alpha position overlapped.

These limitations of protein structure prediction are well understood. Here, we demonstrate the fact that even when a homologous PDB structure is available at 45% sequence identity, an experimentally determined structure will still have valuable atomic level details that cannot be adequately transferred via comparative modelling. It is clear that despite the fold conservation, it is the intricate details that could be the factor in understanding mechanistic differences between homologous proteins. This in a way affects the target selection criteria for the structural genomics pipeline. As a result, manual intervention of the pipeline may allow for a more refined target list as opposed to a target list generated wholly based on pre-determined values and computed without expert curation.

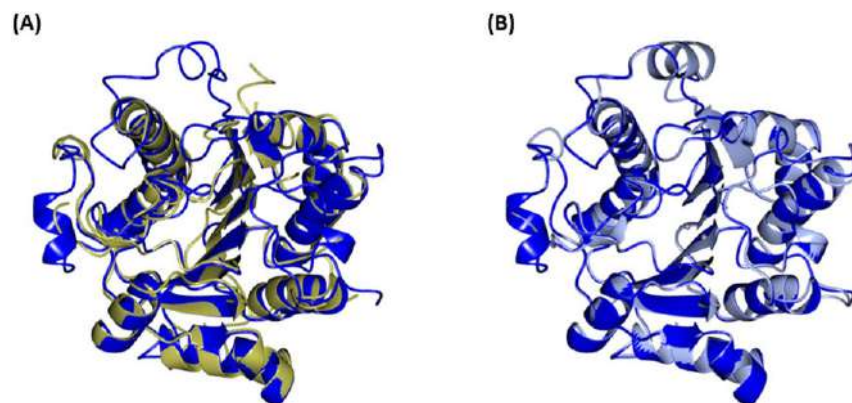


Fig. 2 Comparison of *G. antarctica* arginase structure generated from protein crystallography against a predicted model generated by SWISS-MODEL. The superposition of *G. antarctica* arginase structures between (A) predicted 3D model (dark blue) with X-ray crystal structure (gold) (RMSD: 1.06 Å/243 C α) and (B) predicted 3D model (dark blue) with the template used for prediction (PDBID:4ity) (cyan) (RMSD: 0.31 Å/300 C α). Superposition was done using CCP4MG.

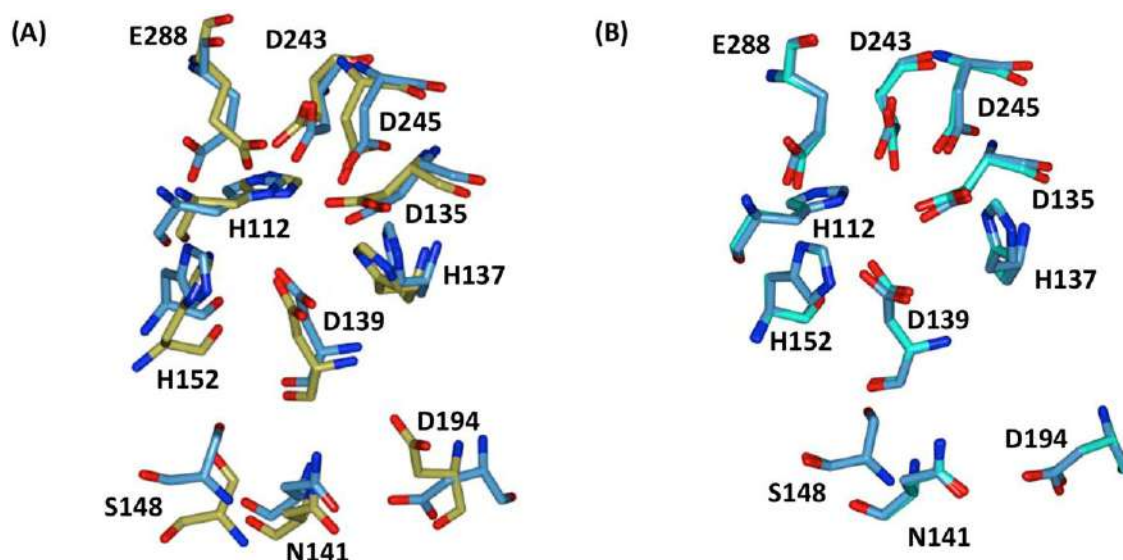


Fig. 3 Superposition of active site residues between (a) *G. antarctica* arginase structure solved by X-ray crystallography (gold) (PDB template used: 4hze) against the SWISS-MODEL predicted arginase (light blue) and (b) The SWISS-MODEL predicted arginase (light blue) against the template used for prediction (PDBID: 4ity). Active site residues are labeled based on the numbering of *G. antarctica* arginase crystal structure. The strictly conserved putative metal coordinating residues in *G. antarctica* arginase are H112, H137, D135, D139, D243 and D245 and several conserved putative arginine binding residues are D139, N141, S148, H152, D194 and E288.

Illustrative Examples and Case Studies

Burkholderia pseudomallei

Burkholderia pseudomallei is a pathogenic soil bacteria that causes the disease melioidosis. At the initial time of writing (circa November 2017), there were 237 PDB entries for *B. pseudomallei* in the PDB. Filtering those for structures that have less than 95% sequence identity retrieved 99 PDB entries. Due to the use of mutant structures as a means to characterize functional mechanisms, this reduced number is most likely the actual count for unique *B. pseudomallei* structures in the PDB. Of the 237 structures, 51% (121 entries) were deposited by the Seattle Structural Genomics Center for Infectious Disease (SSGCID) (Myler *et al.*, 2009). The number submitted by the SSGCID could be reduced to 63 entries when a sequence identity filter of 95% was used.

The SSGCID is a National Institute of Allergy and Infectious Diseases (NIAID) consortium applying structural genomics approaches for identifying potential drug targets (Stacy *et al.*, 2011). In the search for drugs against *B. pseudomallei*, the SSGCID embarked on a combined functional and structural genomics approach to identify essential proteins and solve their structures (Baugh *et al.*, 2013). This effort involved experimental identification of essential genes using transposon mutagenesis in *Burkholderia thailandensis*, a phylogenetically similar member of the *Burkholderia* genus. With 406 putative essential genes identified, an ‘ortholog rescue’ strategy was employed for difficult targets by integrating seven other *Burkholderia* species into the pipeline as sources for orthologous proteins.

These efforts resulted in 31 structures of which 25 proteins have properties of a potential antimicrobial drug target. Although the identification of essential genes was experimentally carried out using transposon mutagenesis, the genomes of the *Burkholderia* species were compared to examples of known essential genes from UniProtKB (see “Relevant Websites section”) and the Database of Essential Genes – DEG (see “Relevant Websites section”) (Zhang *et al.*, 2009). The identified targets were also used to identify the existence of human homologs using BLASTP – the results of this was used to further filter the target list to only proteins that did not have any human homologs.

The work that reported the discovery of *Burkholderia* lethal factor 1 (BLF-1) (Cruz-Migoni *et al.*, 2011) started on a similar footing. The project was aimed at identifying *B. pseudomallei* essential genes, as well as novel pathogenicity factors. Despite some similarities to the SSGCID initiative, the drafting of the target list was carried out differently. Instead of using transposon mutagenesis, potential essential genes were computationally identified based on their homologs in the DEG resource. An additional dataset of genes that were annotated to encode hypothetical proteins was also generated. The essential genes homologs and the hypothetical proteins were then cross-referenced against the proteomic profiles of *B. pseudomallei* and *B. thailandensis* reported by Wongtrakongate *et al.* (2007). *B. thailandensis* is non-pathogenic, therefore proteins present in the *B. pseudomallei* proteome but absent in the *B. thailandensis* proteome were assumed to have potential roles in pathogenesis – these were then selected as targets for the pipeline. The BPSL1549 protein was found to be present in the 2D gel electrophoresis data of *B. pseudomallei* but absent in that of *B. thailandensis* thus leading to its selection for structure determination.

For decades, conclusive evidence for the pathogenicity and virulence factors of *B. pseudomallei* had eluded investigators (Stone, 2007). The BPSL1549 structure, which was renamed to BLF-1, had fold similarities to the catalytic domain of *Escherichia coli*

cytotoxic necrotizing factor 1 (CNF-1) (Cruz-Migoni *et al.*, 2011). The solution of the BLF-1 structure led to the identification of its substrate, the eukaryotic translation eIF4A, hence characterizing the role of BLF-1 in the disruption of protein synthesis in its host.

Glaciozyma antarctica

Glaciozyma antarctica is an obligate psychrophilic yeast that is able to survive at temperatures below 20°C. From its genome annotation, a total of 7857 genes were predicted (Firdaus-Raih *et al.*, 2018). Target genes for structural genomics were then chosen by filtering out 1610 genes encoding transmembrane regions, 4874 proteins predicted to be insoluble, 198 proteins computed to have low complexity regions and 23 proteins where coils make up more than 20% of the total sequence. These filtering steps also selected genes that have lengths of between 50 and 700 amino acids and no similar examples in the PDB at a sequence identity of 30% or lower which produced 453 target genes with 53% of this number having been annotated as hypothetical proteins.

Several target genes from *G. antarctica* were further crystallized and six of the structures had been successfully solved at the time of writing with three having been deposited in the PDB (PDBIDs: 4xxf, 5yhp, 5ynl). A resource that tracks the progress of target CDS for structure solution is available as part of the GlacIER (*Glaciozyma antarctica* Integrated Exploration Resource) database (see “Relevant Websites section”). This is perhaps another aspect that a computational and informatics based approach can be integrated into structural genomics. In this case, a simple MySQL database was made accessible via a web interface that allowed members of the project as well as the public varying levels of access to the data. Users of the database can thus track the progress of the targets as they proceed through the pipeline. Undoubtedly, more complex enterprise level management of such pipelines are in use by larger structural genomics consortia that work on multiple target organisms, however, the GlacIER database demonstrates that a small group in extended collaboration with several other labs can also execute a structural genomics project.

Discussion and Future Directions

Although the bulk of the work done in structural genomics consists of experimental work to produce proteins and generate 3D structure coordinate data, it is clear that extensive sequence database searching needs to be carried out during the identification of targets for the pipeline. Computational approaches are once again called upon for analysis of the 3D structures once they become available. While most structural genomics pipelines that can be found in the literature adopt a similar structure to that presented in Figs. 1 and 3, this is expected to evolve as the resolutions of structures acquired using cryo-EM get higher.

It is not expected for cryo-EM to supplant protein crystallography in providing the bulk of the 3D structures in the PDB; however, future structural genomics projects may employ all three approaches in order to provide a wider coverage of the structures that can be acquired from an available genome sequence. These would thus require the target selection process to take into account the specific requirement for each method and no longer be highly dependent solely on protein solubility. Informatic approaches would be required to provide decision making tools to identify targets and estimate the requirement and/or necessity of solving specific homologous structures. The pipeline may also evolve further to include computational methods of structure determination.

Closing Remarks

The structural genomics pipeline and specific filters enable what can be an expensive endeavour to have a reduced workload and cost. This allows for smaller and less well funded groups outside of large structural genomics consortia to also practice structural genomics as discussed in the case studies examples.

Acknowledgement

MFR is funded by the Ministry of Science, Technology and Innovation grant 02-01-02-SF1278 and the Universiti Kebangsaan Malaysia grant GP-K011849.

See also: Biological Database Searching. Biomolecular Structures: Prediction, Identification and Analyses. Comparative Genomics Analysis. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Sequence Analysis. Whole Genome Sequencing Analysis

References

Ahmad, L., Hung, T.L., Mat Akhir, N.A., *et al.*, 2015. Characterization of Burkholderia pseudomallei protein BPSL1375 validates the Putative hemolytic activity of the COG3176 N-Acyltransferase family. BMC Microbiol. 15, 270.

- Anfinsen, C.B., 1972. The formation and stabilization of protein structure. *Biochem. J.* 128, 737–749.
- Baugh, L., Gallagher, L.A., Patrapuvich, R., *et al.*, 2013. Combining functional and structural genomics to sample the essential Burkholderia structome. *PLOS ONE* 8, e53851.
- Bhattacharya, A., Tejero, R., Montelione, G.T., 2007. Evaluating protein structures determined by structural genomics consortia. *Proteins* 66, 778–795.
- Biasini, M., Bienert, S., Waterhouse, A., *et al.*, 2014. SWISS-MODEL: Modelling protein tertiary and quaternary structure using evolutionary information. *Nucleic Acids Res.* 42, W252–W258.
- Chothia, C., Lesk, A.M., 1986. The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5, 823–826.
- Cruz-Migoni, A., Ruzhenikov, S.N., Sedelnikova, S.E., *et al.*, 2011. Cloning, purification and crystallographic analysis of a hypothetical protein, BPSL1549, from Burkholderia pseudomallei. *Acta Crystallogr. Sect. F Struct. Biol. Cryst. Commun.* 67, 1623–1626.
- Debret, G., Martel, A., Cuniasse, P., 2009. RASMOT-3D PRO: A 3D motif search webserver. *Nucleic Acids Res* 37, W459–W464.
- Dey, S., Ritchie, D.W., Levy, E.D., 2018. PDB-wide identification of biological assemblies from conserved quaternary structure geometry. *Nat. Methods* 15, 67–72.
- Fedorov, O., Sundstrom, M., Marsden, B., Knapp, S., 2007. Insights for the development of specific kinase inhibitors by targeted structural genomics. *Drug Discov. Today* 12, 365–372.
- Firdaus-Raih, M., Hashim, N.H.F., Bharudin, I., *et al.*, 2018. The Glaciozyma antarctica genome reveals an array of systems that provide sustained responses towards temperature variations in a persistently cold habitat. *PLOS ONE* 13, e0189947.
- Franklin, M.C., Cheung, J., Rudolph, M.J., *et al.*, 2015. Structural genomics for drug design against the pathogen Coxiella burnetii. *Proteins* 83, 2124–2136.
- Hashim, N.H., Bharudin, I., Nguong, D.L., *et al.*, 2013. Characterization of Atp1, an antifreeze protein from the psychrophilic yeast Glaciozyma antarctica PI12. *Extremophiles* 17, 63–73.
- Holm, L., Laakso, L.M., 2016. Dali server update. *Nucleic Acids Res.* 44, W351–W355.
- Konc, J., Janecz, D., 2017. ProBIS tools (algorithm, database, and web servers) for predicting and modeling of biologically interesting proteins. *Prog. Biophys. Mol. Biol.* 128, 24–32.
- Laskowski, R.A., Jablonska, J., Pravda, L., Varekova, R.S., Thornton, J.M., 2018. PDBsum: Structural summaries of PDB entries. *Protein Sci.* 27, 129–134.
- Marsden, R.L., Lewis, T.A., Orengo, C.A., 2007. Towards a comprehensive structural coverage of completed genomes: A structural genomics viewpoint. *BMC Bioinform.* 8, 86.
- Myler, P.J., Stacy, R., Stewart, L., *et al.*, 2009. The seattle structural genomics center for infectious disease (SSGCID). *Infect. Disord. Drug Targets* 9, 493–506.
- Nadzirin, N., Firdaus-Raih, M., 2012. Proteins of unknown function in the protein data bank (pdb): An inventory of true uncharacterized proteins and computational tools for their analysis. *Int. J. Mol. Sci.* 13, 12761–12772.
- Nadzirin, N., Gardiner, E.J., Willett, P., Artymiuk, P.J., Firdaus-Raih, M., 2012. SPRITE and ASSAM: Web servers for side chain 3D-motif searching in protein structures. *Nucleic Acids Res.* 40, W380–W386.
- Nadzirin, N., Willett, P., Artymiuk, P.J., Firdaus-Raih, M., 2013. IMAAAGINE: A webserver for searching hypothetical 3D amino acid side chain arrangements in the Protein Data Bank. *Nucleic Acids Res.* 41, W432–W440.
- Nair, R., Liu, J., Soong, T.-T., *et al.*, 2009. Structural genomics is the largest contributor of novel structural leverage. *J. Struct. Funct. Genom.* 10, 181–191.
- Ramli, A.N., Azhar, M.A., Shamsir, M.S., *et al.*, 2013. Sequence and structural investigation of a novel psychrophilic alpha-amylase from Glaciozyma antarctica PI12 for cold-adaptation analysis. *J. Mol. Model.* 19, 3369–3383.
- Stacy, R., Begley, D.W., Phan, I., *et al.*, 2011. Structural genomics of infectious disease drug targets: The SSGCID. *Acta. Crystallogr. Sect. F Struct. Biol. Cryst. Commun.* 67, 979–984.
- Stone, R., 2007. Infectious disease. Racing to defuse a bacterial time bomb. *Science* 317, 1022–1024.
- Wongtrakongate, P., Mongkoldhumrongkul, N., Chaijan, S., Kamchonwongpaisan, S., Tungpradabkul, S., 2007. Comparative proteomic profiles and the potential markers between Burkholderia pseudomallei and Burkholderia thailandensis. *Mol. Cell. Probes* 21, 81–91.
- Yusof, N.A., Hashim, N.H., Beddoe, T., *et al.*, 2016. Thermotolerance and molecular chaperone function of an SGT1-like protein from the psychrophilic yeast, Glaciozyma antarctica. *Cell Stress Chaperones* 21, 707–715.
- Zarembinski, T.I., Hung, L.-W., Mueller-Dieckmann, H.-J., *et al.*, 1998. Structure-based assignment of the biochemical function of a hypothetical protein: A test case of structural genomics. *Proc. Natl. Acad. Sci. USA* 95, 15189–15193.
- Zhang, Y., Lin, Z.A., Pan, J.J., *et al.*, 2009. Concurrent control study of different radiotherapy for primary nasopharyngeal carcinoma: Intensity-modulated radiotherapy versus conventional radiotherapy. *Ai Zheng* 28, 1143–1148.

Relevant Websites

- <http://mfrlab.org/glacier/>
Glacier: Glaciozyma antarctica Integrated Exploration Resource.
- <http://www.uniprot.org>
The UniProt knowledgebase.
- <http://tubic.tju.edu.cn/deg/>
Tianjin University Bioinformatics Centre.

Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4)

Teo Chian Ying, International Medical University, Bukit Jalil, Kuala Lumpur, Malaysia

Zalikhah Ibrahim, International Islamic University Malaysia, Kuantan, Malaysia

Mohd Basyaruddin Abd Rahman, Universiti Putra Malaysia, Serdang, Selangor, Malaysia

Bimo A Tejo, UCSI University, Cheras, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Rheumatoid arthritis (RA) is a chronic inflammatory disease that is characterized by symmetrical pain in one or more joints (Landre-Beauvais, 2001). The symptoms usually start with stiffness in the morning, followed by joint tenderness and swelling. The number of affected joints varies, but most of the disease development involves five or more joints, including both small and big joints. Since its identification, RA has been the cause of joint inflammations in 0.5%–1% of the world's population (Gabriel, 2001).

The number of people diagnosed with RA in year 2008 has been estimated to be nearly 1.3 million in the United States (Helmick *et al.*, 2008). The affected patients have a low health-related quality of life due to physical damages induced by the disease (Salaffi *et al.*, 2009). This caused difficulties for the patients to perform their routine activities, and for some, it resulted in losing their jobs. With the assumption that human life span increases every year, it is expected that the frequency of correctly diagnosed patients with RA will increase in the future.

RA is regarded as an autoimmune disease (Davidson and Diamond, 2001). The word 'autoimmune' is a combination of two separate words that is auto and immune. Auto refers to self, while immune means the immune system. Therefore, autoimmune disease can be easily understood as a disease that is triggered by the abnormal response of the immune system against one's substance or tissue that is naturally presented in the body. In the case of RA, the target is the joint and the tissues that surround it. The onset of RA is caused by the stimulation of the immune system that is triggered by foreign substrates that mimic the joint cells (Davidson and Diamond, 2001). As a defensive mechanism, the immune components start to attack the invaders and coincidentally, the joint cells too. This in turn, causes the swelling of the synovial membrane, leading to joint inflammation.

Over the past 30 years, several classes of drugs have been introduced to treat RA. Routinely, these drugs are prescribed after the first tissue is damaged. The main aim is to minimize RA joint inflammations, but the drugs do not treating the underlying causes. At present, three classes of drugs are commercially available, namely corticosteroids, non-steroidal anti-inflammatory drugs (NSAID) and disease-modifying anti-rheumatic drugs (DMARD). Depending on the disease severity, an appropriate drug is prescribed and the patient status is reviewed from time to time. Among all three drugs classes, DMARD is the most commonly prescribed medication for RA patients. An example of a drug from this class is methotrexate (MTX) (Aletaha *et al.*, 2010). Although it is effective in relieving RA symptoms, the mechanism of action of MTX in treating RA remains unclear. A major drawback of MTX is that it can cause the development of multiple side effects on the central nervous, hepatic, pulmonary, hematologic and gastrointestinal systems especially in long-term usage (Smolen *et al.*, 2005). This drug also has a toxicity issue, which is why MTX was discontinued in certain, more susceptible RA patients (Smolen *et al.*, 2010).

RA is not be a fatal disease, however, severe and unmanageable conditions may arise that gradually disrupts overall human function, and thus, the quality of life (Salaffi *et al.*, 2009). Long-standing RA patients have an increased risk of suffering multiple health problems. Therefore, it is crucial to diagnose RA at an earlier stage of the disease; so that treatment can be administered sooner and the major damage of joint tissues can be prevented. Recent interest has been focused on citrullinated peptides, which is a true RA antigen (Schellekens *et al.*, 1998, 2000). Detection of autoantibodies against this type of antigen is more specific and sensitive than the current RA serological marker, Rheumatoid Factor (RF). This makes the antibodies correspond to the citrullinated peptides, name Anti-Citrullinated Peptide Antibody (ACPA) a suitable clinical marker to be developed (Wegner *et al.*, 2009).

Citrulline can be produced from an arginine residue in a peptide/protein (van Venrooij and Pruijn, 2000). Since the overall conversion involves the replacement of an imine group to a carbonyl group, the chemical reaction is often referred to as deimination (Fig. 1). The reaction is catalyzed by an enzyme called Protein Arginine Deiminase (PAD). In normal physiology, citrullinated peptides are present in proteins such as myelin sheath and histone (Jones *et al.*, 2009). In the case of RA, irregular expressions of citrullinated peptides (and hence, ACPA) have been reported (Hoet and van Venrooij, 1992; Hoet *et al.*, 1991; Vincent *et al.*, 1999). Traces of the peptides and the enzyme catalyzing the reaction, PAD type 4 (PAD4) are found in the sera and joints of the patients (Kinloch *et al.*, 2008). This triggers the autoimmune response, resulting in joint inflammation. Therefore, suppression of the citrullination pathway has been hypothesized to be a strategic yet specific way to slow down the disease. An ideal approach is to modulate the protein that catalyzes the reaction, namely PAD4.

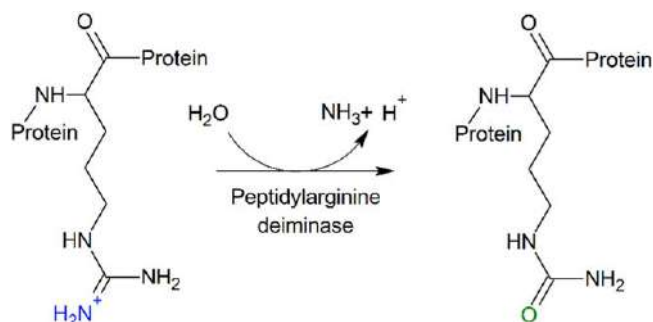


Fig. 1 Enzymatic conversion of peptidylarginine into peptidyl citrulline.

Related Autoantibodies in RA Diagnosis

RA is indeed a tricky disease to diagnose. This is because the clinical symptoms are visible only after the first tissue damage. Luckily, during the disease development, the immune cells of the RA patient excrete various types of autoantibodies that can be used as indicators. There are a numbers of excellent reviews regarding RA-related autoantibodies and their biochemistry (Boekel *et al.*, 2002; Vossenaar, 2004).

Rheumatoid Factor (RF) is the first autoantibody found in RA patients. It was described in the late 1930s and is still used as one of the criteria for diagnosing RA (Aletaha *et al.*, 2010; Arnett *et al.*, 1988). It is reactive towards the tail (or known as Fc portion) of immunoglobulin G (IgG), which is an antibody that protects the human body from infections. This factor however, has also been found in many other diseases involving immune reactions and infections (Boekel *et al.*, 2002). To a certain extent, this antibody has been found in 5% of healthy population, in which the percentage increases up to 30% in healthy elderly. It is interesting to note that not all RA patients would exhibit this formula. This results in the classification of 'seropositive' or 'seronegative' RA groups. The former are the ones who demonstrated physical symptoms with RF antibodies, the latter are those who manifested the symptoms with no evidence of elevated RF values (Arnett *et al.*, 1988). Altogether, this indicates that the RF had low specificity and sensitivity; hence it can not be used as a single indicator in RA diagnosis.

In recent years, the emergence of autoantibodies towards citrullinated peptides has created a new path of research in rheumatology (van Venrooij and Pruijn, 2000). Historically, two autoantibodies namely the anti-perinuclear factor (APF) and the anti-keratin antibodies (AKA) are described to be specific for RA (Hoet and van Venrooij, 1992). Although the latter has lower specificity than the former, investigation on both antibodies shows that they are interrelated (Hoet *et al.*, 1991; Vincent *et al.*, 1999). They share the same antigen namely filaggrin in its matured form, which is citrulline-specific. Several years later, researchers find traces of other antibodies that also respond to citrulline-containing peptides, for example, anti-filaggrin in RA patients (Schellekens *et al.*, 1998). This becomes a foundation for further studies on the use of citrulline-containing peptides for RA diagnosis. Reactivity investigation of diverse citrullinated peptides shows a remarkable pattern of APF and AKA responses. Of all tested peptides, antibody responses towards cyclic citrullinated peptides (CCP) are remarkable in RA patients with 98% specificity. Nowadays, enzymatic assay against CCP is used as a common procedure in RA diagnosis (Aletaha *et al.*, 2010). The fact that the anti-citrullinated peptides antibodies (ACPA) can be detected before the first tissue damage makes them a reliable serological marker for early RA detection (Jansen *et al.*, 2002).

Citrullination in the RA Cycle

While the anti-citrullinated peptide antibodies (ACPA) acts as a potential serological marker for RA, it has been hypothesized that the citrullination pathway may participate in the RA cycle (van Venrooij and Pruijn, 2008). The citrullination process converts the peptidyl-bound arginine residue into a peptidyl-bound citrulline residue, as a result of posttranslational modification. The upregulation of citrullinated peptides is said to break the immune tolerance, triggering the release of the CCP-related antibodies and consequently the onset of RA. This creates a new level of understanding and opened opportunities for studies of the disease process.

In a review by Klareskog and colleagues, they summarize several factors including environmental, citrullination and genetic variants that contributed primarily towards the rise of ACPA (Klareskog *et al.*, 2011). The factors are interrelated, in which the disease could be triggered when multiple factors are present. Based on their early work, the research group discovered that the relation between smoking and RA could be seen via the presence of HLA-DR shared epitope (SE) genes in a particular person (Klareskog *et al.*, 2006). The probability of stimulating the disease is 21-fold higher in the smokers with the genes, when compared to those with the absence of either factor. As the enzyme that catalyzes citrullination, PAD, is also involved in apoptosis, it becomes activated. This indirectly increases the ACPA.

On a different note, there is a review on oral citrullination by a Gram-negative bacterium called *Porphyromonas gingivalis* (Mangat *et al.*, 2010). It is a major pathogen in periodontitis, an advanced gum disease that involves the inflammatory immune

response. Given the fact that *P. gingivalis* is the only bacteria to date to express PAD in its cells, it is speculated that the PAD expressed by the bacteria may have been able to catalyze the citrullination of human peptidylarginine too. Once the tolerance is broken, the immune responses are invoked. The bacteria association with RA is also supported by the level of antibodies to *P. gingivalis* in RA patient when compared to a healthy sample, indicating that the bacteria may have indeed plays a role in the disease development (Mikuls *et al.*, 2009).

Protein Arginine Deiminase IV (PAD4)

PAD4 belongs to the family of guanidinium modifying superfamily (GMSF). As the name suggests, these are proteins that are responsible in converting the guanidinium side chain into other functional groups. Together with PAD4, there are also PAD1, PAD2, PAD3 and PAD6 in the PAD family. As a matter of fact, it was thought that human PAD5 was a novel PAD. However, detailed investigations on the sequence and expression shows that the human PAD5 is analogous to the mouse PAD4. Therefore, it is renamed as PAD4, leaving a vacancy on the fifth type (Vossenaar *et al.*, 2003). PAD4 has several specific roles in normal body. It is involved in the formation of neutrophil extracellular traps (NET). The foundations of this NET are neutrophils, which are among the first responders during the occurrence of a bacterial infection. By the stimulation from PAD4, the neutrophils are combined to form a more complex NET system. This is proven by the disruption of NET when PAD4 is inhibited (Lewis *et al.*, 2015). Apart from that, PAD4 is also a corepressor of the p53 protein. This p53 is a tumor suppressor, which can initiate apoptosis of damaged deoxyribonucleic acid (DNA). By corepressing p53, PAD4 might induce the growth of the tumor, evidenced by the trace of PAD4 in patients with malignant tumors (Chang *et al.*, 2009). Other reviews on PAD4 and its roles can be found below (Fuhrmann *et al.*, 2015; Jones *et al.*, 2009).

PAD4 Structure

PAD4 contains 663 amino acids, with a molecular weight of 74 kDa. The first 294 amino acids are arranged in two immunoglobulin-like (I g) structures, making the N-terminal domain (Arita *et al.*, 2004). These subdomains are proposed to assist substrate selection and protein-protein interaction. Inside the subdomain 1, there is a nuclear localization signal (NLS) (56-PPAKKKST-63) that is arranged in a loop. This loop region is highly flexible as it exposed to solvent, which is explained by the high temperature factor in the crystallographic data. Based on the records, PAD4 is the only PAD with this NLS region. The peptide sequence acts as a tag that will interact with the nuclear transport receptors embedded in the nucleus envelope, for import into the cell nucleus.

The remaining sequence, is folded into a α/β propeller and forms the C-terminal domain. This domain ports all the active residues that are needed for the citrullination catalysis. The key residues are D350, H471, D473 and C645. PAD4 is only active in the presence of calcium and naturally exists as a homodimeric protein in solution. The dimer is reported to be in a head-to-tail fashion; that is between the N-terminal subdomain 1 of one monomer and catalytic domain of the other monomer. Fig. 2 illustrates the dimer structure of PAD4.

Calcium Dependency

It is common for proteins to have metal binding sites. A protein may contain one or more metal ions, which are unique in terms of their role and coordination. The basic functions of metals in proteins can be classified into five categories (Holm *et al.*, 1996): (1) For structural stability of the secondary or ternary protein structure, (2) for storage, (3) for electron transfer, (4) oxygen binding and lastly (5) for catalytic reaction. Data from crystallographic studies reveals that PAD4 had five unique calcium binding sites; three (Ca3–5) are in the N-terminal domain and another two (Ca1 and Ca2) are in the catalytic domain (Arita *et al.*, 2004). The recent 1.8 Å resolution structures of calcium-free and calcium-bound PAD4 (PDB ID: 1WD8 and 1WD9) reveals an interesting role of Ca1 and Ca2 towards the enzyme. The residues on the surface of PAD4 catalytic pocket is exposed to solvent in the calcium free PAD4. Upon the binding of the calcium, the residues are more structured. Further analysis of the electrostatic surface potential also indicates changes from the highly acidic to the more basic surface, where it is more favourable for peptidyl-arginine to bind. In addition, the geometry of the active site in the substrate-free, calcium bound PAD4 is found to be similar to that of the substrate bound PAD4, signifying that the formation of the active site is not dependent on substrate binding (Fig. 3). Overall, it is deduced that the catalytic pocket shape is highly dependent on these catalytic ions.

Liu and colleagues (Liu *et al.*, 2013) recently reported the role of calcium ions in the PAD4 N-terminal domain. Based on the kinetic studies, it is observed that the binding network of Ca3 and Ca4 is vital for full enzyme activation. Mutation of the residues that are coordinated with Ca3 and Ca4 namely D155, D157, and D179 diminishes the enzyme catalytic ability significantly. In contrast, the mutants with respect to the Ca5 binding show an indistinct effect on the catalytic efficiency, indicating that the enzyme has less dependency on the Ca5 binding.

Metal dependency of PAD4 has also been tested in a work reported by Kearney and co-workers (Kearney *et al.*, 2005). They concluded that only calcium can efficiently activate the PAD4 catalytic activity. Additionally, some of the metals even have the

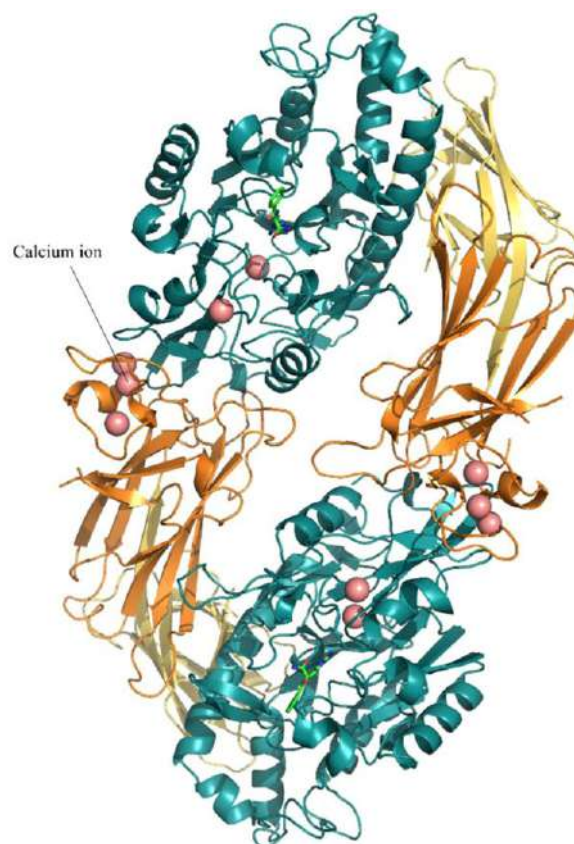


Fig. 2 PAD4 dimer structure in head-to-tail fashion. Subdomain 1 is colored in yellow, subdomain 2 is in orange, and the catalytic domain is in deep teal.

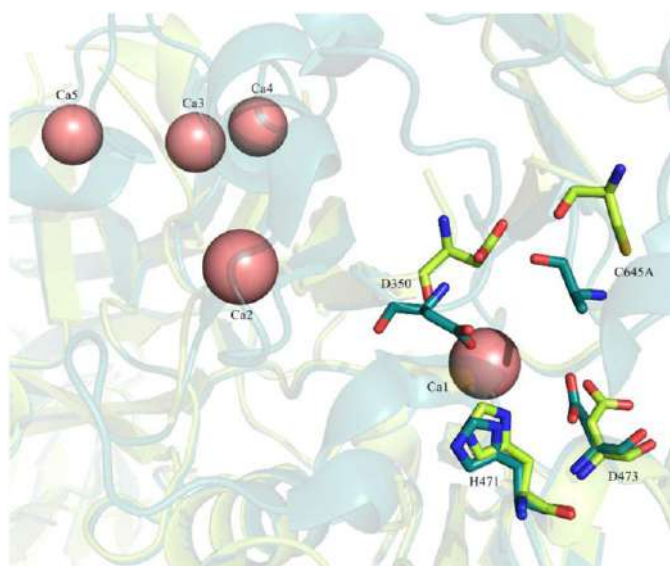


Fig. 3 Superimposition of deep-teal colored calcium-unbound PAD4 (PDB ID: 1WD8) onto lemon color calcium-bound PAD4 (1WD9). Some parts of the proteins were set in higher transparency for visualization purpose.

ability to inhibit the protein and could possibly be a PAD4 deactivator. Metals that display strong PAD4 inhibition properties are manganese, samarium and zinc. These findings prove the importance of calcium in PAD4 activation. Interestingly, although the calcium ions are needed for citrullination catalysis, they do not affect the formation of the PAD4 dimer ([Arita et al., 2004](#); [Liu et al., 2011](#)).

Substrate Recognition

The PAD4 active site has a U-tunnel shape, with two door entrances (Fig. 4). The “front door” is the entrance where the actual substrate conversion takes place (Arita *et al.*, 2004, 2006). The “back door” on the other hand, is thought to provide access for solvent to enter and subsequently complete the hydrolysis (Linsky and Fast, 2010). The shape of tunnel is narrow, allowing only small molecules to enter and interact with the active residues. Similar two-door pockets have also been found in other guanidinium-modifying hydrolases, such as arginine deiminase (ADI) and dimethylarginine dimethylaminohydrolase (DDAH) (Linsky and Fast, 2010). An inspection on the structure of PAD4 complexed with benzoyl-arginine amide (BAA) substrate (Fig. 5) reveals several key residues that are important for substrate recognition. BAA is small substrate that mimics peptidylarginine and has been used widely as a positive control in any PAD4 related experiments. The guanidinium nitrogen atom of BAA is found to form double hydrogen bonds with the carbonyl side chain of D473 and D350. This secures the position of the guanidinium group for nucleophilic attack by C645. The opposite part of BAA is fastened by double and single hydrogen bonds from the R374 side chain and R639 backbone respectively. Another important residue is W347, which holds the part in between the substrate’s ends. The effect is best described as providing large hydrophobic effects to the substrates. Consistent with this interaction analysis, mutations of the above residues distract protein-ligand interactions significantly.

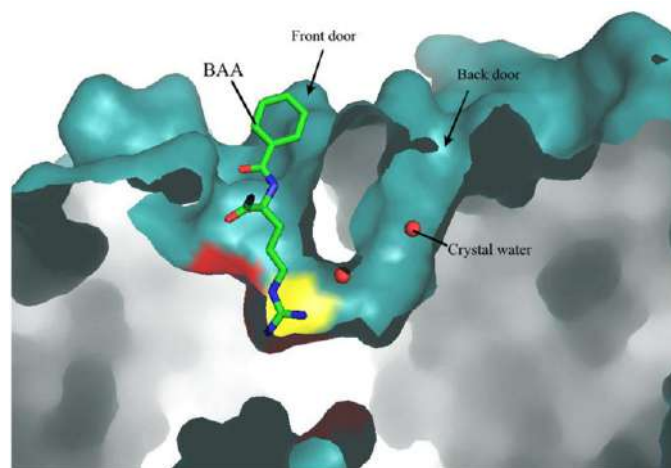


Fig. 4 Two-door pocket of PAD4 rendered using the crystal structure of PAD4 in bound with BAA (PDB ID: 1WDA). Some parts of protein are removed to ease visualization.

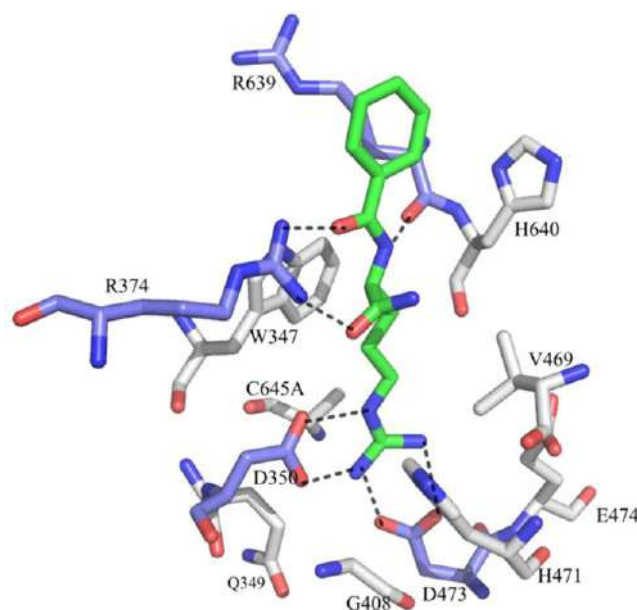


Fig. 5 Key residues in the PAD4-BAA interaction. Hydrogen bonds within 3.35 Å are presented in purple, when non-bonded contacts within the range of 2.90–3.90 are rendered in white.

Table 1 Sequences of designed peptides. The residues are presented in 1 letter amino acid code. C-terminal of each peptide was amidated

Peptide	Sequence
Pep1	MSPLRPQNY-NH ₂
Pep2	QLSLRTVSL-NH ₂
Pep3	PVVLRLKCG-NH ₂
Pep4	ISGKRSPAG-NH ₂
Pep5	KKSIRDTPA-NH ₂
Pep6	SSTPRSKGQ-NH ₂
Pep7	KNCFRMTDQ-NH ₂
Pep8	LWQWRKSL-NH ₂

The ability of PAD4 to convert free arginine remains a point of argument. PAD4 is indeed highly selective towards its substrates (Arita *et al.*, 2004). The enzyme recognizes oxygen atoms of BAA via R374 and R639. However, by using free arginine, the oxygen atom recognition by R374 cannot be established. This interrupts the interactions at R639, making free arginine a poor substrate.

Design of Peptide-Based PAD4 Inhibitors

There are several natural substrate of PAD4 such as histone, α -enolase, vimentin and nucleophosmin (Arita *et al.*, 2004; Hagiwara *et al.*, 2002; Suzuki *et al.*, 2007). To determine the peptide sequence that can bind strongly to PAD4, our group designed several peptides based on the sequence of nucleophosmin (Teo *et al.*, 2017). There are eight target residues i.e., arginine in the sequence of nucleophosmin so eight peptides were designed and named as Pep1 to Pep8. Pep1 to Pep7 consist of nine residues with four flanking residues before and after the target residue, arginine (region -4 to +4). While Pep8 consists of eight residues with only three flanking residues after arginine because it is near to the C-terminal of nucleophosmin. The sequence of each peptide is listed in Table 1.

The target amino acid of PAD4, arginine was positioned in the middle of the sequence and based on the sequence of nucleophosmin, four or three amino acids before and after arginine were chosen as the flanking residues that would aid in binding to residues around the active site of PAD4. The binding of each peptide to PAD4 was studied by comparing the rate of citrullination of each peptide to PAD4 small substrate, N- α -benzoyl arginine ethyl ester (BAEE). Peptides that bind well to the active site of PAD4 would be favourable in citrullination process and therefore more citrulline would be produced. To increase the affinity of the peptides in binding with PAD4, the best peptide was chosen for modification where the sequence was shortened from nine residues to 7, 5, and 3 residues.

The binding of the peptides after shortening were again compared to BAEE in order to identify the best sequence for PAD4 binding. The peptide having the greatest affinity to bind with PAD4 (with highest % relative activity) was selected for peptide-based inhibitor design. The target amino acid, arginine in the middle of the sequence was substituted with two amino acids: an amino acid with free amino side chain and L-2-furylalanine (Fig. 6).

Activity of Peptide-Based PAD4 Inhibitors

The relative activity of PAD4 with the designed peptides as substrate is shown in Fig. 7. Each peptide shows different substrate efficiency for citrullination by PAD4 demonstrating the influence of primary sequence of a peptide flanking an arginine. The influence of flanking residues in citrullination has also been reported in previous studies (Arita *et al.*, 2006; Hagiwara *et al.*, 2005; Nakayama-Hamada *et al.*, 2005; Takahara *et al.*, 1985; Tarcza *et al.*, 1996). However the amino acids could not be classified as favourable or unfavourable residues simply based on their physicochemical features. The influence of a residue on the citrullination by PAD4 is dependent on the overall sequence in which it is embedded (Stensland *et al.*, 2009).

Pep5 shows the highest activity to PAD4 with relative activity of 38%. It contains an aspartic acid at +1 position of arginine residue which results in a higher degree of citrullination (Stensland *et al.*, 2009). To improve the substrate efficiency for citrullination by PAD4, Pep5 was shortened from 9 residues to 7, 5 and 3 residues with the position of arginine remained at the middle of the peptides. The new peptides are named as Pep5_7, Pep5_5 and Pep5_3, respectively.

Fig. 8 shows the relative activity of PAD4 when the modified peptide served as substrate. When two amino acids are removed from each terminal in the sequence (Pep5_7), the relative activity of PAD4 increases drastically from less than 40% to nearly 100%. But when the number of residues is further reduced to five and three residues, the relative activity of PAD4 decreases. Pep5_5 and Pep5_3 may be too short in exhibiting sufficient interaction with PAD4 thus affected the substrate efficiency of the peptides on citrullination by the enzyme. This is in agreement with previous study where when long peptide was truncated, the peptides would be more unfavourable for citrullination (Stensland *et al.*, 2009).

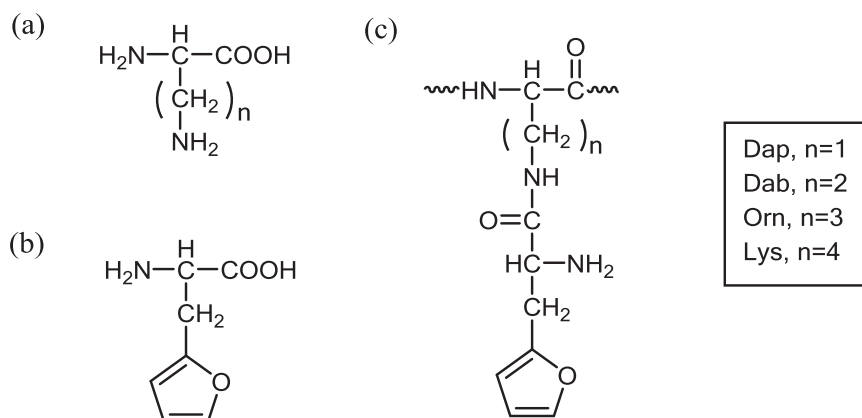


Fig. 6 Chemical structures of amino acids substituted arginine in peptide-based inhibitors. (a) Amino acids with different side chain length ended with a free amino group, L-2,3-diaminopropionic acid (Dap), L-2,4-diaminobutyric acid (Dab), L-ornithine (Orn), and L-lysine (Lys). (b) L-2-furylalanine (Fal). (c) Branched peptide was formed by coupling free amino group of (a) with carboxylic group of (b).

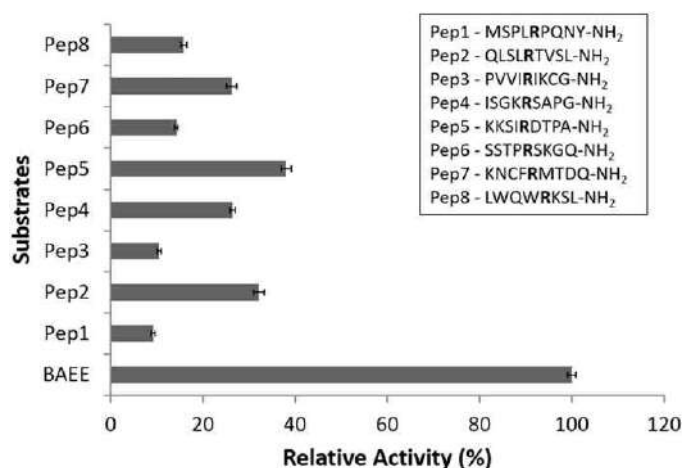


Fig. 7 Screening of peptide substrates of PAD4. Eight designed peptides are utilized as substrates of PAD4 and the relative activity of PAD4 is calculated based on BAEE. Sequences of the peptides are shown in 1 letter amino acid code. The target arginine residue of PAD4 is shown in bold. C-termini of each peptide is amidated.

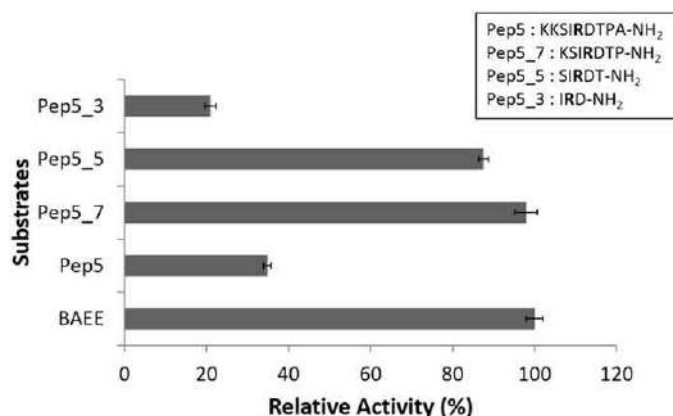


Fig. 8 Screening of shortened Pep5 as substrate of PAD4. Pep5 was shortened from 9 residues to 7, 5 and 3 residues and the modified peptides were utilized as substrate for PAD4. Sequences of the peptides are shown in 1 letter amino acid code. The target arginine residue of PAD4 is shown in bold. C-terminal of each peptide was amidated.

Among the designed peptides, Pep5_7 shows the highest relative activity on citrullination by PAD4 indicating that it has the most appropriate length and sequence to interact with PAD4. The peptide-based inhibitors are then designed according to the sequence of Pep5_7 (see Fig. 8). The arginine residue in the middle of the sequence was substituted with non-standard amino acid containing furan ring which was the common chemotype among the active compounds discovered by our previous works utilising structure-based and ligand-based drug design approaches (Teo *et al.*, 2012, 2013). Based on the findings from both approaches, peptide-based inhibitors were designed as branched peptides with the incorporation of furan ring in the structure. Branched peptide and peptidomimetic yielded from introduction of non-standard amino acid in peptide structure are better than an unmodified peptide in terms of biological activity, stability and specificity (Vlieghe *et al.*, 2010). They overcome the limitation of a peptide in acting as a drug with having better bioavailability. The general structure of the designed peptide-based inhibitors is shown in Fig. 9.

As the furan ring of Compound 5 discovered by ligand-based approach (Teo *et al.*, 2013) is able to enter the binding pocket of PAD4 and interacted with C645 i.e., an essential catalytic residue of PAD4 (Knuckley *et al.*, 2007) via hydrophobic interaction, our hypothesis is that the branch of the peptide would enter the binding pocket of PAD4 and interacts with the amino acids that are essential for the enzymatic activity. Inhibitors with different length of “neck” are designed by using four amino acids with different side chain length i.e., L-2, 3-diaminopropionic acid (Dap), L-2, 4-diaminobutyric acid (Dab), L-orthinine (Orn), and L-lysine (Lys) and named as Inh Dap, Inh Dab, Inh Orn and Inh Lys, respectively.

Most of the synthetic substrates and reported inhibitors of PAD4 (Jones *et al.*, 2012; Luo *et al.*, 2006a,b; Causey *et al.*, 2011) also having the “backbone”, “neck” and “warhead” parts in the structures (Fig. 10). Such design is inspired by the structure of a small synthetic substrate of PAD4, benzoyl-L-arginine amide (BAA). The fluoroacetamidine acts as the warhead to the active site of PAD4 in F-amidine. Luo *et al.* (2006a) then modified the warhead by changing fluoroacetamidine to chloroacetamidine. They discovered that chloroacetamidine is more potent than fluoroacetamidine in inhibiting PAD4. The second portion, which is the neck, consists of an alkyl skeleton and plays a role in strengthening the binding by interacting with residues lining the pocket. The length of side chain i.e., the “neck” was shown to be important in inactivation positioning (Luo *et al.*, 2006a). For their backbones, they are basically consist of a benzene ring and peptide bonds to mimic the structure of natural peptides. Modification of the backbone results in inhibitors with improved potency (Causey *et al.*, 2011).

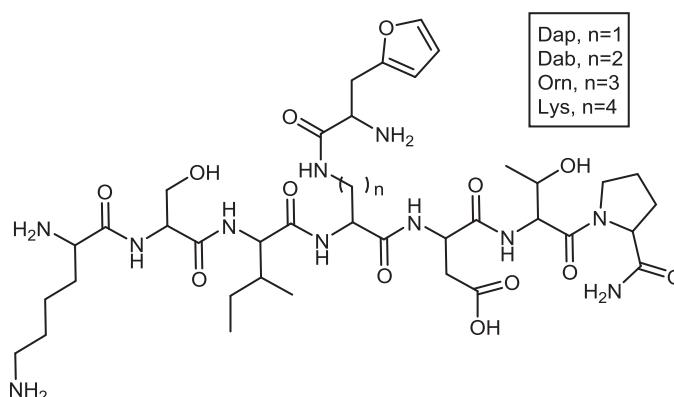


Fig. 9 General structure of designed peptide-based inhibitors. The target residue, arginine was substituted with non-standard amino acid containing furan ring with different side chain length. C-termini of peptide-based inhibitors were amidated.

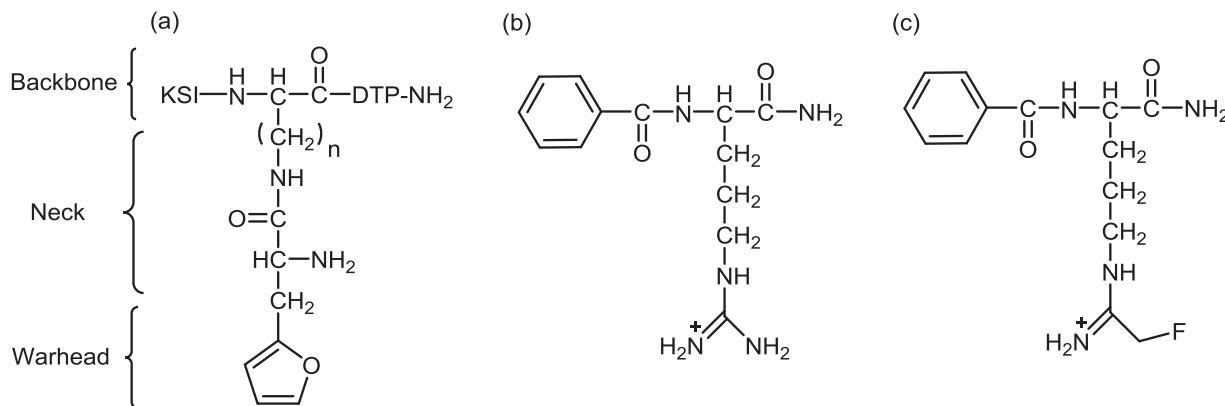


Fig. 10 The chemical structures of (a) designed peptide-based inhibitors, (b) BAA, and (c) F-amidine. All the structures are generally consisted of backbone, neck and warhead. Standard residues are shown in 1 letter amino acid code.

Table 2 IC_{50} values of the designed peptide-based inhibitors. IC_{50} values were calculated using GraphPad Prism 5.1

Inhibitors	$IC_{50} \pm SD$ (μM)
Inh Dap	243.2 ± 2.4
Inh Dab	315.5 ± 6.2
Inh Orn	> 2 mM
Inh Lys	277.3 ± 2.7

Inhibitory Activity of Peptide-Based Inhibitors

The inhibitory activity of the designed peptide-based inhibitors is investigated by calculating the IC_{50} value. Most of the inhibitors inhibit PAD4 significantly by reducing PAD4 activity to less than 50% at 2 mM except Inh Orn. The IC_{50} values of the peptide-based inhibitors are shown in **Table 2**. Among the inhibitors, Inh Dap is the most active inhibitor followed by Inh Lys and Inh Dab.

There is no direct relationship between the “neck” lengths of the inhibitor with the inhibitor potency. Luo *et al.* (2006a) have also studied the relationship between the “neck” lengths with the inhibitory ability of the amidine-based inhibitors. The length of the “neck” of the amidine inhibitors is varied by manipulating the methylene group from two to four units. Similar to peptide-based inhibitors, the inhibition potency is not directly related to the “neck” length of the compound where they found that amidines with three methylene units are more potent compared to that of two or four units. However this indicates that suitable “neck” length is crucial for proper position of the warhead in interacting with the active site of PAD4.

Inh Dap inhibits PAD4 the most among the inhibitors could be due to the suitable “neck” length. The relatively low IC_{50} values of peptide-based inhibitors compared to compounds discovered by structure-based and ligand-based virtual screening (Teo *et al.*, 2012, 2013) indicates that the incorporation of peptide with furan ring could be a better PAD4 inhibitor against PAD4 compare to conventional small molecule inhibitor.

Molecular Docking Analysis of Inh Dap

Molecular docking of Inh Dap against PAD4 is performed to investigate the interactions between the peptide with PAD4. The conformation of Inh Dap with the lowest energy calculated using NMR restrained MD simulations is selected for docking purpose. The binding affinity of Inh Dap to PAD4 is -5.4 kcal/mol, suggesting that the peptide has favourable interaction with PAD4. There are three hydrogen bonds formed between the inhibitor and the residues on the binding site of PAD4. The first one is established between the backbone carbonyl group of Thr6 and the side chain hydroxyl group of Ser468. The second is formed between the side chain carboxylic acid of Asp5 and the side chain amine of R374, while the third one is observed between the side chain amine of Dfa4 and the side chain carboxylic acid of D350. The binding of Inh Dap is mostly favoured by hydrophobic interactions between Dfa4 side chain that contains the furan ring and residues such as R639, W347, C645, H471, H640, E474, and D473. Other parts of Inh Dap such as Asp5 and Thr6 also make hydrophobic contacts with E575, R441, G403, V469, and G641 (**Fig. 11**).

It is interesting to highlight that the furan ring enters the deep binding pocket of PAD4 and made contacts with the residues that involved in the catalytic mechanism of PAD4, which are H471, D473, D350 and C645 (Kearney *et al.*, 2005). Similar interaction is also observed in the docked complex of PAD4-ligand discovered via ligand-based virtual screening (Teo *et al.*, 2013). Moreover, it is also worth to note that the presence of a hydrogen bond between Asp5 of Inh Dap with R374 of PAD4. As mentioned before, R374 is an important amino acid in the substrate recognition for PAD4 (Arita *et al.*, 2006). We anticipate that the contacts made between Inh Dap with C645 and R374 give the major contribution in inhibiting the PAD4 activity.

CD spectroscopy and NMR studies showing that Inh Dap is in unordered structure. A bent conformation is induced on Inh Dap upon binding to PAD4 (see **Fig. 12**). Previous studies reported that it is important for the peptide around the target residue to have a highly disordered conformation as PAD4 recognizes an unstructured peptide at the molecular surface near the binding site cleft and induces a bent conformation on the peptide (Arita *et al.*, 2006; Takahara *et al.*, 1985; Tarcsa *et al.*, 1996). The induction of the bent conformation is due to the hydrogen bonding between Inh Dap and Arg374 (Arita *et al.*, 2006). The bent shape allows the furan ring to enter the deep binding pocket of PAD4. The binding of furan ring in the active site cleft of PAD4 again confirms the importance of this structural fragment in the structure of PAD4 inhibitors.

Concluding Remarks

We show a systematic approach in designing a peptide inhibitor for PAD4. The unique architecture of PAD4 binding pocket requires the inhibitor to be able to penetrate deeply into the binding pocket while anchoring itself on the outer surface of the

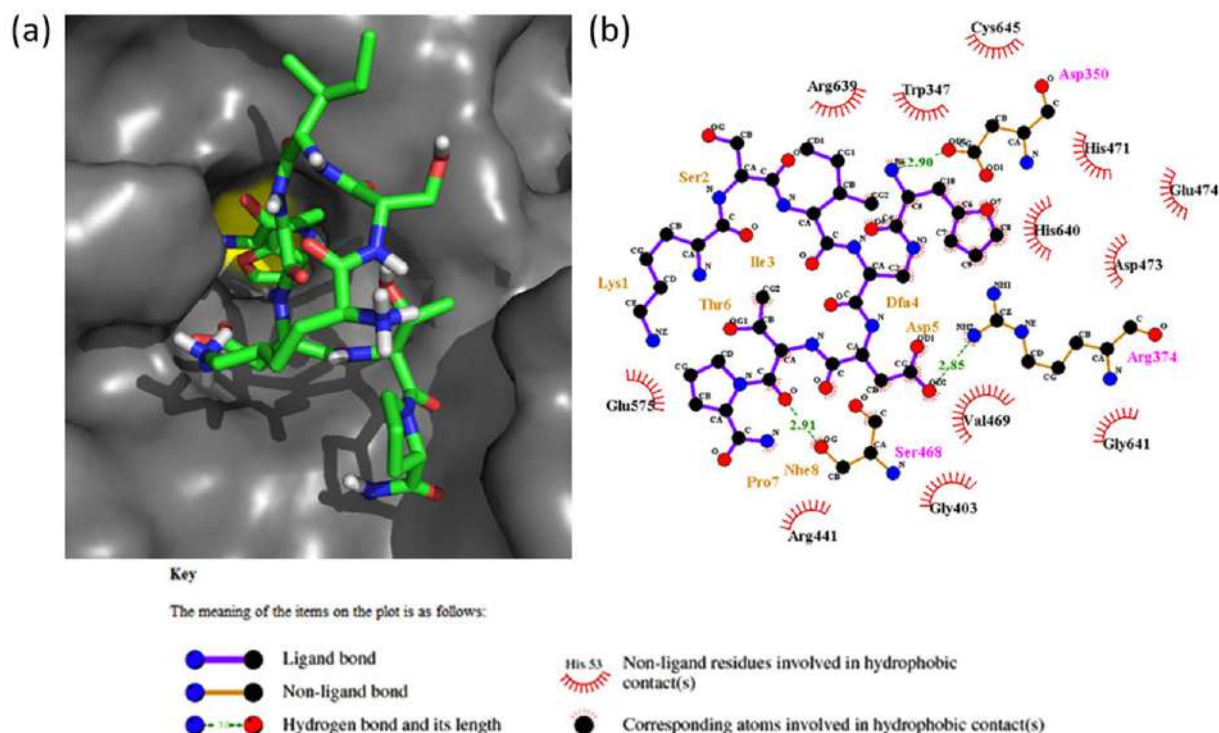


Fig. 11 Binding conformation of *Inh Dap* inside the PAD4 binding pocket. (a) *Inh Dap* is seen occupying the binding pocket with the furan ring is located in proximity with C645 (yellow). Some of the protein parts were removed to ease visualization (Schrodinger, 2002). (b) Hydrogen bonds and hydrophobic interactions of *Inh Dap* with residues in the binding pocket of PAD4. The 2-dimensional binding diagram was created using Ligplot + version 1.4.5 (Laskowski and Swindells, 2011).

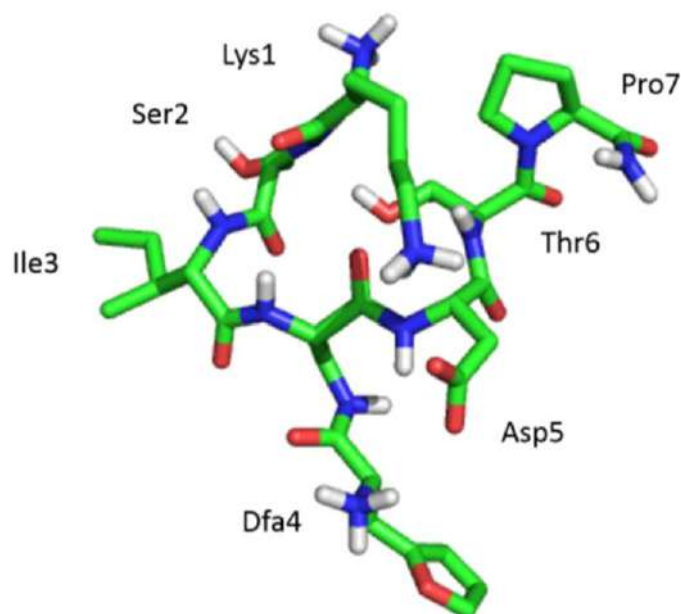


Fig. 12 Conformation of *Inh Dap* after binding to PAD4. A bent conformation was induced on *Inh Dap* upon its binding at the binding site of PAD4.

pocket. The screening of small molecules is carried out to choose the best “warhead” to attack the catalytic residues inside the binding pocket. The “warhead” is then attached to the carefully designed peptide with optimum neck and anchor lengths to ensure the peptide strongly binds to the neighboring residues while the “warhead” is attacking the catalytic residues. This approach in designing peptide inhibitor can be applied to similar cases where the catalytic residues are located far inside the binding pocket.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Comparative Epigenomics. Computational Protein Engineering Approaches for Effective Design of New Molecules. Identifying Functional Relationships Via the Annotation and Comparison of Three-Dimensional Amino Acid Arrangements in Protein Structures. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Protein Design. Structure-Based Drug Design Workflow

References

- Aletaha, D., Neogi, T., Silman, A.J., *et al.*, 2010. 2010 Rheumatoid arthritis classification criteria: An American College of Rheumatology/European league against rheumatism collaborative initiative. *Arthritis Rheum.* 62 (9), 2569–2581.
- Arita, K., Hashimoto, H., Shimizu, T., *et al.*, 2004. Structural basis for Ca^{2+} -induced activation of human PAD4. *Nat. Struct. Mol. Biol.* 11 (8), 777–783.
- Arita, K., Shimizu, T., Hashimoto, H., *et al.*, 2006. Structural basis for histone N-terminal recognition by human peptidylarginine deiminase 4. *Proc. Natl. Acad. Sci. USA* 103 (14), 5291–5296.
- Arnett, F.C., Edworthy, S.M., Bloch, D.A., *et al.*, 1988. The American Rheumatism Association 1987 revised criteria for the classification of rheumatoid arthritis. *Arthritis Rheum.* 31 (3), 315–324.
- van Boekel, M.A., Vossenaar, E.R., van den Hoogen, F.H., van Venrooij, W.J., 2002. Autoantibody systems in rheumatoid arthritis: Specificity, sensitivity and diagnostic value. *Arthritis Res.* 4 (2), 87–93.
- Causey, C.P., Jones, J.E., Slack, J.L., *et al.*, 2011. The development of N- α -(2-carboxyl)benzoyl-N(5)-(2-fluoro-1-iminoethyl)-l-ornithine amide (o-F-amidine) and N- α -(2-carboxyl)benzoyl-N(5)-(2-chloro-1-iminoethyl)-l-ornithine amide (o-Cl-amidine) as second generation protein arginine deiminase (PAD) inhibit. *J. Med. Chem.* 54 (19), 6919–6935.
- Chang, X., Han, J., Pang, L., *et al.*, 2009. Increased PAD4 expression in blood and tissues of patients with malignant tumors. *BMC Cancer* 9 (1), 40.
- Davidson, B., Diamond, B., 2001. Autoimmune diseases. *N. Engl. J. Hum. Genet.* 345 (5), 340–350.
- Fuhrmann, J., Clancy, K.W., Thompson, P.R., 2015. Chemical biology of protein arginine modifications in epigenetic regulation. *Chem. Rev.* 115 (11), 5413–5461.
- Gabriel, S., 2001. The epidemiology of rheumatoid arthritis. *Rheum. Dis. Clin. N. Am.* 27 (2), 269–281.
- Hagiwara, T., Hidaka, Y., Yamada, M., 2005. Deimination of histone H2A and H4 at arginine 3 in HL-60 granulocytes. *Biochemistry* 44 (15), 5827–5834.
- Hagiwara, T., Nakashima, K., Hirano, H., Senshu, T., Yamada, M., 2002. Deimination of arginine residues in nucleophosmin/B23 and histones in HL-60 granulocytes. *Biochem. Biophys. Res. Commun.* 290 (3), 979–983.
- Helmick, C.G., Felson, D.T., Lawrence, R.C., *et al.*, 2008. Estimates of the prevalence of arthritis and other rheumatic conditions in the United States. Part I. *Arthritis Rheum.* 58 (1), 15–25.
- Hoet, R.M., Boerbooms, A.M., Arends, M., Ruiter, D.J., van Venrooij, W.J., 1991. Antiperinuclear factor, a marker autoantibody for rheumatoid arthritis: Colocalisation of the perinuclear factor and profilaggrin. *Ann. Rheum. Dis.* 50 (9), 611–618.
- Hoet, R.M., van Venrooij, W.J., 1992. The Antiperinuclear Factor (APF) and Antikeratin Antibodies (AKA) in rheumatoid arthritis. In: Smolen, J.S., Kalden, J.R., Maini, R.N. (Eds.), *Rheumatoid Arthritis*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 299–318.
- Holm, R.H., Kennepohl, P., Solomon, E.I., 1996. Structural and functional aspects of metal sites in biology. *Chem. Rev.* 96 (7), 2239–2314.
- Jansen, A.L.M.A., van der Horst-Bruinsma, I., van Schaardenburg, D., *et al.*, 2002. Rheumatoid factor and antibodies to cyclic citrullinated Peptide differentiate rheumatoid arthritis from undifferentiated polyarthritis in patients with early arthritis. *J. Rheumatol.* 29 (10), 2074–2076.
- Jones, J.E., Causey, C.P., Knuckley, B., Slack-Noyes, J.L., Thompson, P.R., 2009. Protein arginine deiminase 4 (PAD4): Current understanding and future therapeutic potential. *Curr. Opin. Drug Discov. Dev.* 12 (5), 616–627.
- Jones, J.E., Slack, J.L., Fang, P., *et al.*, 2012. Synthesis and screening of a haloacetamide containing library to identify PAD4 selective inhibitors. *ACS Chem. Biol.* 7 (1), 160–165.
- Kearney, P.L., Bhatia, M., Jones, N.G., *et al.*, 2005. Kinetic characterization of protein arginine deiminase 4: A transcriptional corepressor implicated in the onset and progression of rheumatoid arthritis. *Biochemistry* 44 (31), 10570–10582.
- Kinloch, A., Lundberg, K., Wait, R., *et al.*, 2008. Synovial fluid is a site of citrullination of autoantigens in inflammatory arthritis. *Arthritis Rheum.* 58 (8), 2287–2295.
- Klareskog, L., Malmström, V., Lundberg, K., Padyukov, L., Alfredsson, L., 2011. Smoking, citrullination and genetic variability in the immunopathogenesis of rheumatoid arthritis. *Semin. Immunol.* 23 (2), 92–98.
- Klareskog, L., Stolt, P., Lundberg, K., *et al.*, 2006. A new model for an etiology of rheumatoid arthritis: Smoking may trigger HLA-DR (shared epitope)-restricted immune reactions to autoantigens modified by citrullination. *Arthritis Rheum.* 54 (1), 38–46.
- Knuckley, B., Bhatia, M., Thompson, P.R., 2007. Protein arginine deiminase 4: Evidence for a reverse protonation mechanism. *Biochemistry* 46 (22), 6578–6587.
- Landre-Beauvais, A.J., 2001. The first description of rheumatoid arthritis. Unabridged text of the doctoral dissertation presented in 1800. *Joint Bone Spine: Revue Du Rhumatisme* 68 (2), 130–143.
- Laskowski, R.A., Swindells, M.B., 2011. LigPlot + : Multiple ligand–protein interaction diagrams for drug discovery. *J. Chem. Inform. Model.* 51 (10), 2778–2786.
- Lewis, H.D., Liddle, J., Coote, J.E., *et al.*, 2015. Inhibition of PAD4 activity is sufficient to disrupt mouse and human NET formation. *Nat. Chem Biol.* 11 (3), 189–191.
- Linsky, T., Fast, W., 2010. Mechanistic similarity and diversity among the guanidine-modifying members of the pteinte superfamily. *Biochim. Biophys. Acta* 1804 (10), 1943–1953.
- Liu, Y.L., Chiang, Y.H., Liu, G.Y., Hung, H.C., 2011. Functional role of dimerization of human peptidylarginine deiminase 4 (PAD4). *PLOS ONE* 6 (6), e21314.
- Liu, Y.L., Tsai, I.C., Chang, C.W., *et al.*, 2013. Functional roles of the non-catalytic calcium-binding sites in the N-terminal domain of human peptidylarginine deiminase 4. *PLOS ONE* 8 (1), e51660.
- Luo, Y., Arita, K., Bhatia, M., *et al.*, 2006a. Inhibitors and inactivators of protein arginine deiminase 4: Functional and structural characterization. *Biochemistry* 45 (39), 11727–11736.
- Luo, Y., Knuckley, B., Lee, Y.H., Stallcup, M.R., Thompson, P.R., 2006b. A fluoroacetamide-based inactivator of protein arginine deiminase 4: Design, synthesis, and in vitro and in vivo evaluation. *J. Am. Chem. Soc.* 128 (4), 1092–1093.
- Mangat, P., Wegner, N., Venables, P.J., Potempa, J., 2010. Bacterial and human peptidylarginine deiminases: Targets for inhibiting the autoimmune response in rheumatoid arthritis? *Arthritis Res. Ther.* 12 (3), 201–209.
- Mikuls, T.R., Payne, J.B., Reinhardt, R.A., *et al.*, 2009. Antibody responses to *Porphyromonas gingivalis* (*P. gingivalis*) in subjects with rheumatoid arthritis and periodontitis. *Int. Immunopharmacol.* 9 (1), 38–42.
- Nakayama-Hamada, M., Suzuki, A., Kubota, K., *et al.*, 2005. Comparison of enzymatic properties between hPAD2 and hPAD4. *Biochem. Biophys. Res. Commun.* 327 (1), 192–200.
- Salaffi, F., Carotti, M., Gasparini, S., Intorcchia, M., Grassi, W., 2009. The health-related quality of life in rheumatoid arthritis, ankylosing spondylitis, and psoriatic arthritis: A comparison with a selected sample of healthy people. *Health Qual. Life Outcomes* 7 (1), 25.
- Schellekens, G.A., de Jong, B.A., van den Hoogen, F.H., van de Putte, L.B., van Venrooij, W.J., 1998. Citrulline is an essential constituent of antigenic determinants recognized by rheumatoid arthritis-specific autoantibodies. *J. Clin. Invest.* 101 (1), 273–281.

- Schellekens, G.A., Visser, H., de Jong, B.A., *et al.*, 2000. The diagnostic properties of rheumatoid arthritis antibodies recognizing a cyclic citrullinated peptide. *Arthritis Rheum.* 43 (1), 155–163.
- Schrodinger, L., 2002. The PyMOL Molecular Graphics System, Version 1.4.1.
- Smolen, J.S., Aletaha, D., Machold, K.P., *et al.*, 2005. Therapeutic strategies in early rheumatoid arthritis. *Best Pract. Res. Clin. Rheum.* 19 (1), 163–177.
- Smolen, J.S., Landewé, R., Breedveld, F.C., *et al.*, 2010. EULAR recommendations for the management of rheumatoid arthritis with synthetic and biological disease-modifying antirheumatic drugs. *Ann. Rheum. Dis.* 69 (6), 964–975.
- Stensland, M.E., Pollmann, S., Molberg, O., Sollid, L.M., Fleckenstein, B., 2009. Primary sequence, together with other factors, influence peptide deimination by peptidylarginine deiminase-4. *Biol. Chem.* 390 (2), 99–107.
- Suzuki, A., Yamada, R., Yamamoto, K., 2007. Citrullination by peptidylarginine deiminase in rheumatoid arthritis. *Ann. N.Y. Acad. Sci.* 1108, 323–339.
- Takahara, H., Okamoto, H., Sugawara, K., 1985. Specific modification of the functional arginine residue in soybean trypsin inhibitor (Kunitz) by peptidylarginine deiminase. *J. Biol. Chem.* 260 (14), 8378–8383.
- Tarcsa, E., Marekov, L.N., Mei, G., *et al.*, 1996. Protein unfolding by peptidylarginine deiminase. Substrate specificity and structural relationships of the natural substrates trichohyalin and filaggrin. *J. Biol. Chem.* 271 (48), 30709–30716.
- Teo, C.Y., Abdul Rahman, M.B., Chor, A.L.T., *et al.*, 2013. Ligand-Based virtual screening for the discovery of inhibitors for Protein Arginine Deiminase Type 4 (PAD4). *J. Postgenom. Drug Biomarker Dev.* 3 (1), 1–5.
- Teo, C.Y., Shave, S., Chor, A.L.T., *et al.*, 2012. Discovery of a new class of inhibitors for the protein arginine deiminase type 4 (PAD4) by structure-based virtual screening. *BMC Bioinform.* 13 (Suppl. 17), S4.
- Teo, C.Y., Tejo, B.A., Leow, A.T.C., Salleh, A.B., Abdul Rahman, M.B., 2017. Novel furan-containing peptide-based inhibitors of protein arginine deiminase type IV (PAD4). *Chem. Biol. Drug Des.* 90 (6), 1134–1146.
- Wegner, N., Lundberg, K., Kinloch, A., *et al.*, 2009. Autoimmunity to specific citrullinated proteins gives the first clues to the etiology of rheumatoid arthritis. *Immunol. Rev.* 233, 1–21.
- van Venrooij, W.J., Pruijn, G.J.M., 2000. Citrullination : A small change for a protein with great consequences for rheumatoid arthritis. *Arthritis Res.* 2, 249–251.
- van Venrooij, W.J., Pruijn, G.J.M., 2008. An important step towards completing the rheumatoid arthritis cycle. *Arthritis Res. Ther.* 10 (5), 117.
- Vincent, C., de Keyser, F., Masson-Bessière, C., *et al.*, 1999. Anti-perinuclear factor compared with the so called “antikeratin” antibodies and antibodies to human epidermis filaggrin, in the diagnosis of arthritides. *Ann. Rheum. Dis.* 58 (1), 42–48.
- Vlieghe, P., Lisowski, V., Martinez, J., Khrestchatisky, M., 2010. Synthetic therapeutic peptides: Science and market. *Drug Discov. Today* 15 (1–2), 40–56.
- Vossenaar, E.R., Zendman, A.J.W., Van, Venrooij, *et al.*, 2003. PAD, a growing family of citrullinating enzymes: Genes, features and involvement in disease. *BioEssays* 25 (11), 1106–1118.
- Vossenaar, E., 2004. Anti-CCP antibodies, a highly specific marker for (early) rheumatoid arthritis. *Clin. Appl. Immunol.* 4 (4), 239–262.

Small Molecule Drug Design

Vartika Tomar, University of Delhi, Delhi, India

Mohit Mazumder, Jawaharlal Nehru University, Delhi, India and University of Saskatchewan, Saskatoon, SK, Canada

Ramesh Chandra, University of Delhi, Delhi, India

Jian Yang and Meena K Sakharkar, University of Saskatchewan, Saskatoon, SK, Canada

© 2019 Elsevier Inc. All rights reserved.

Introduction

For the balanced and proper functioning of all the life sustaining processes, nature has provided our body with all the necessary chemical components or precursors, enzymes and neurotransmitters. Despite this, due to several factors, some exogenous and some endogenous, some machineries or bioprocesses fail to function. The exogenous factors responsible for disrupting the normal bodily function may vary from parasitic invasion to some chemical entities. The endogenous factors include over or under-production of few chemicals, faulty functioning of organs or any genetic or congenital factor leading to disorders like neuro-degenerative disorders like Alzheimer's or Parkinson's disease resulting from the imbalance of acetylcholine and dopamine in the central nervous system (Moore *et al.*, 2005). Hence, to restore the normal functioning, external aids called 'Drugs' or 'Medicines' are required and the process of designing/discovery of drugs is called drug discovery. During this process, combinations of computational, translational, experimental and clinical models are employed to identify new potential therapeutic entities. In early days, most of the drugs were discovered either by the identification of the active ingredients from traditional remedies or by serendipitous discovery. Despite having knowledge of biological systems and advanced biotechnology, drug discovery and its development is still an expensive, time consuming, laborious, complicated and inefficient process with a high attrition rate of new therapeutic discovery. Hence, designing a drug or a molecule with desired properties and function is an important industrial challenge.

What is a Drug?

Drugs are biological or chemical entities of synthetic or natural origin, which modulate the functions of the body without causing any new action on the body. They can be a single compound or a mixture of different compounds. Drugs function by interacting and modifying specific 'targets' in our bodies to create a molecular interaction signature that can be exploited for rapid therapeutic repurposing and discovery. They interact and bind with the targets that are complementary to them in shape and charge and work either by stimulating or blocking activity of their targets. For example, the analgesic effect of drug Aspirin is due to inhibition of prostaglandin biosynthesis by acetylation of cyclo-oxygenase.

What is a Small-Molecule Drug?

According to National Cancer Institute, a small molecule drug is any organic compound or substance capable of entering cells with an ease due to its low molecular weight (below 900 Daltons). Once it enters the cells, it can affect other molecules, such as proteins, and may cause death of cancer cells. This is different from drugs, such as monoclonal antibodies, which are not able to enter the cells easily because of their large molecular weight. Therefore, most of the targeted therapies are small-molecule drugs or small molecule inhibitors.

Advantages of a Small-Molecule Drug

1. Small-molecule drugs are mostly administered orally which gives them an edge over larger drugs that require invasive procedures like injections.

The simple structures and smaller sizes of the small-molecule drugs facilitate more generic competition in comparison to complex biologic drugs. Owing to small molecular weight (i.e., less than 900 Daltons), they have easy penetration into cell membranes or target organs, thereby giving an intra-cellular targeting advantage over extracellularly oriented, specific biological proteins. Since small molecule drugs can be processed into easily ingestible tablets or capsule, they can be orally administrated. This increases the treatment regime, upregulates patient satisfaction and adherence and improves efficacy. Furthermore, these small molecule drugs have short half-life but longer shelf lives, which is of substantial therapeutic benefit especially in cases of rapid metabolism. Also, they require easy and less vigorous manufacturing processes as compared to complex biological drugs.

Characteristics of a Drug

A drug becomes active when it binds to its biological target, usually receptors. Receptors are protein having active sites for ligand binding. Hence, a good ligand can be designed, if the structure of such receptors and their active sites can be identified accurately. Before designing a drug, it is important to know what features we are looking for, in an "ideal drug candidate. The drug

- (i) Must be safe and effective.
- (ii) Should be well absorbed orally and have good bioavailability.
- (iii) Should be metabolically stable and have a long half-life.
- (iv) Nontoxic with minimal or no side effects.
- (v) Should have selective distribution to target tissues.

Drug Discovery

Drug discovery plays an important role to the society and for any pharmaceutical industry in 'improving the therapeutic value and safety of agents' by launching newer and safe drugs in the market. Unfortunately, the process of discovery and development of drug is a long and complicated process. It involves the identification, synthesis, characterization, screening and assays of potential candidates for therapeutic efficacy. Drug discovery and development process includes preclinical studies on cell-based and animal models followed by clinical trials on humans, and finally moving towards the step of obtaining regulatory approval to market the drug. Usually it takes at least 14–16 years of research with a cost of 800 million US dollars for a compound to get developed into a drug. After establishment of the pre-clinical data and confirmation of its action and toxicity, the compound is approved for clinical studies which takes 1–2 years before it is released into the market. After release, several post-market surveillance and pharmacovigilance practices are maintained to identify adverse reactions or incompatibilities when used in combination therapies (Congreve *et al.*, 2005). **Fig. 1** depicts entire drug discovery process with their tentative timelines.

Approaches for Drug Discovery/Designing

The traditional approach involves blind screening of chemical molecules obtained either from nature or synthesized in laboratories causing long design cycles and higher production cost. In modern drug discovery the identification of hits by screening, medicinal chemistry and optimization of these hits for enhances selectivity, metabolic stability and efficacy/potency, affinity and oral bioavailability. Enhanced electivity contributes towards reduction of potential side effects and metabolic stability increases half-life making the drug more stable. Drug discovery process can be made cost-effective and faster by using computer-aided drug design discovery process which involves structure-based drug design using in silico approach which plays a significant role in all stages of drug development. Since only a potential molecule is selected, this prevents late stage clinical failures thereby reducing the cost significantly. **Table 1** lists some inhibitors developed with computational chemistry and rational drug design strategies. Better understanding of the quantitative relationship of structure and biological activity reduces the development time of the drug to 6–8 years from 10 to 16 years. This supports the reasons to develop computer-aided molecular design (CAMD), towards automation of molecular design (Blaney, 1990; Bugg *et al.*, 1993). **Fig. 2** outlines the steps involved in the drug development process.

Various Approaches Involved in Drug Designing

Virtual screening

Virtual screening is a computational method used for identifying lead compounds using a vast and diverse chemical compound library as a reference. This computational method is an important tool for discovering lead compounds as it is faster, more cost-efficient, and less resource intensive in comparison to experimental methods such as high-throughput screening (Walters *et al.*, 1998; Shoichet, 2004; Kitchen *et al.*, 2004; Schneider and Bohm, 2002). Virtual screening process consists of two steps: docking and scoring. AutoDock (Morris *et al.*, 1998; Huey *et al.*, 2007) performs ligand conformational searches for identifying potential bound conformations, and X-Score (Wang *et al.*, 2002) is used in re-evaluating the binding affinity of the predicted structures. Novel lead compounds in many investigations have been successful identified using the freely available Autodock and X-score.

Docking and scoring

Docking is a computational way of predicting the most preferred orientation of one small molecule bound to a target, resulting into a stable complex. It consists of several steps. The first step is the application of docking algorithms that pose small molecules within the active site of the target. Through docking we aim to predict the stable drug interactions by inspecting and modelling drug molecular interactions between drug- and target receptor molecules. Various molecular docking tools are available, including AutoDock, FRED, eHITS, and FTDock, etc. (Maithri *et al.*, 2016). Scoring function is used for computing non-bonded interaction terms between the receptor and ligand atoms. Scoring functions approximate the receptor–ligand interaction energy using multivariate regression of multiple parameters such as the number of hydrogen bonds, lipophilicity, ionic interactions, entropy penalties, etc. Scoring functions are specifically designed to complement these docking algorithms as they evaluate the interactions between compounds and potential targets, thereby predicting their biological activity.

High-throughput screening (HTS)

In this technique, a large number of biological modulators and effectors are screened and assayed against selected and specific targets. The principles and methods of HTS can be exploited for screening of proteins, peptides, genomics and combinatorial

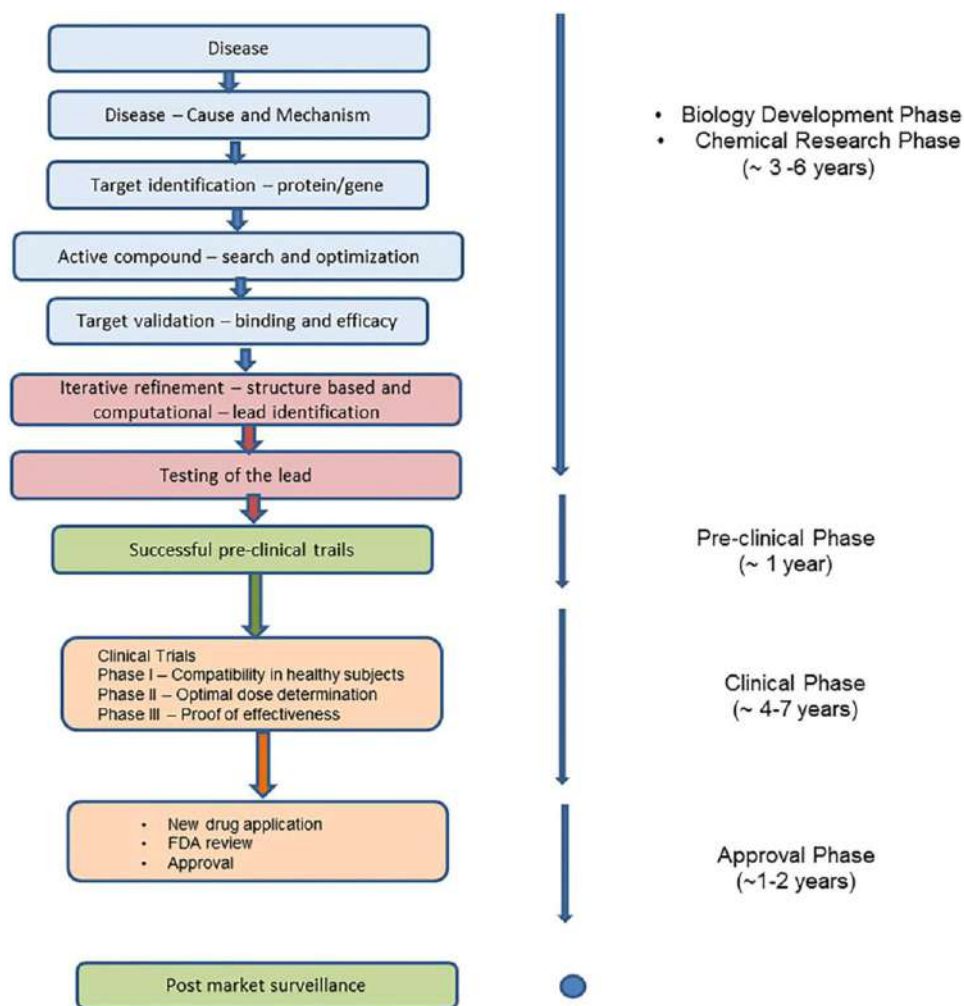


Fig. 1 Different phases of drug discovery process. Adapted and modified from Lombardino, J.G., Lowe, J.A., 2004. The role of the medicinal chemist in drug discovery—then and now. *Nature Reviews Drug Discovery* 3, 853–862.

chemistry libraries. HTS is a high-tech way to hasten the drug discovery process, allowing quick and efficient screening of large compound libraries at a rate of a few thousand compounds per day or per week. (Martis *et al.*, 2011).

Homology modelling

Computational methods can be used to predict the 3D structure of target proteins when experimental structures are not present. Since proteins with similar sequence have similar structures, comparative modelling is used to predict target structures based on a template with similar sequences. Homology modelling is a specific kind of comparative modelling in which the same evolutionary origin is shared by template and target proteins. Some computer programmes and web servers which are commonly used for automated homology modelling process are PSI-PRED and MODELER (Buchan *et al.*, 2010; Martí-Renom *et al.*, 2000).

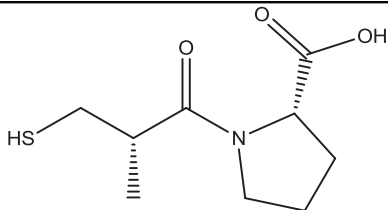
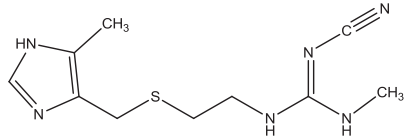
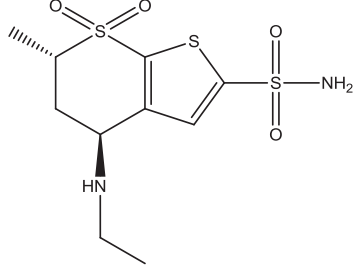
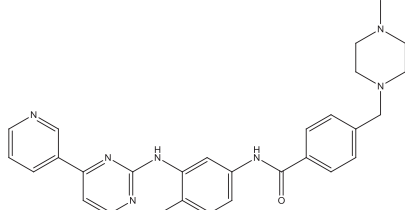
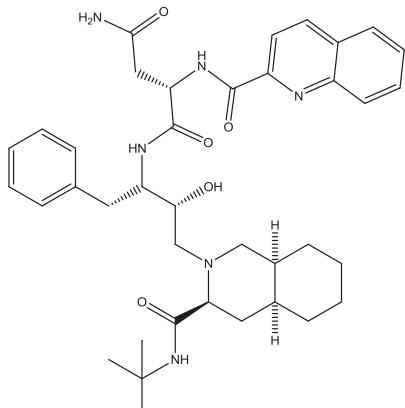
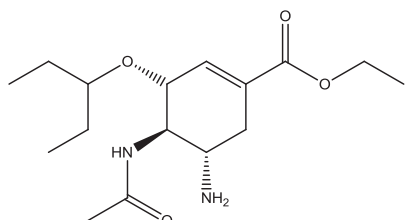
Molecular dynamic simulations

Owing to the dynamic nature of biomolecules, a single static structure cannot be applied to predict putative binding sites. An assembly of target conformations starting from a single structure are obtained by applying classic molecular dynamic (MD) simulations. In MD methods principles of Newtonian mechanics are employed to calculate trajectory of conformations of a protein as a function of time. The problem with classic MD method is that it gets trapped in local energy minima. To overcome this problem, various advanced MD algorithms such as conformational folding simulations (Grubmüller, 1995), targeted-MD (Schlitter *et al.*, 1994), replica exchange MD (Sugita and Okamoto, 1999) and temperature accelerated MD simulations (Abrams and Vanden-Eijnden, 2010) have been employed for traversing multiple minima energy surface of proteins.

Monte Carlo simulation

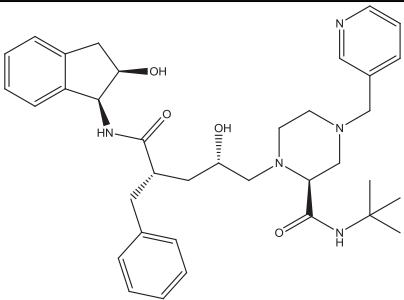
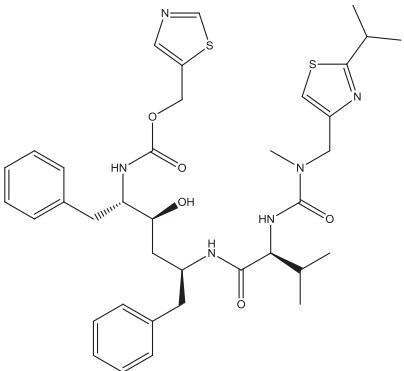
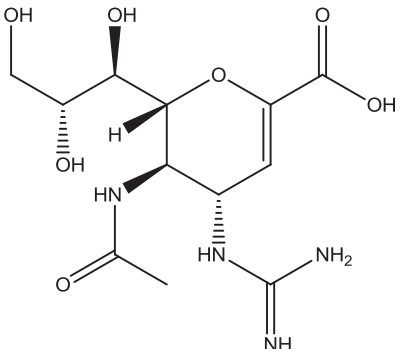
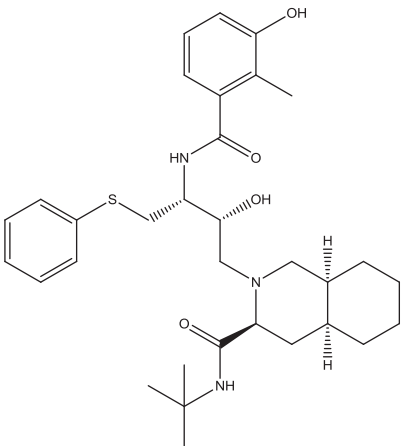
The Monte Carlo simulation is based on the concept of statistical mechanics. Monte Carlo introduces a randomness in the system that allows hopping over the energy barriers thereby preventing the system from getting stuck in the local energy minima

Table 1 Selected inhibitors developed with computational chemistry and rational drug design strategies

Compound name	Structure of the compound	Therapeutic area	Function	Approval
Captopril		Hypertension congestive heart failure myocardial infarction diabetic nephropathy	ACE inhibitor	1975
Cimetidine		Treatment of heartburn and peptic ulcers	H2-receptor antagonist	1978
Dorzolamide		Antiglaucoma agent	Antiglaucoma agent carbonic anhydrase inhibitor	1989
Imatinib		Chronic myeloid leukemia	Tyrosine kinase inhibitor	1990
Saquinavir		Antiretroviral drug used to treat or prevent HIV/AIDS (1st generation)	HIV-1 protease inhibitor	1995
Oseltamivir		Antiviral (to treat influenza A and influenza B)	Influenza neuraminidase inhibitor	1996

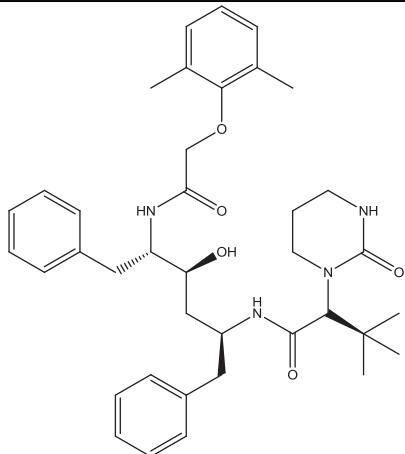
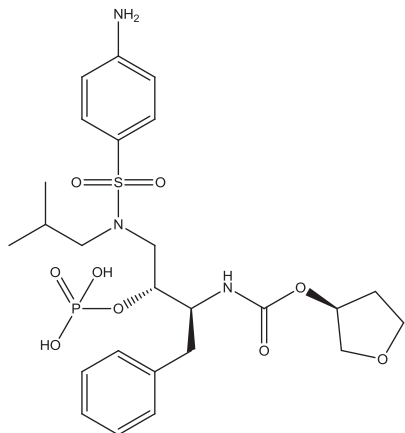
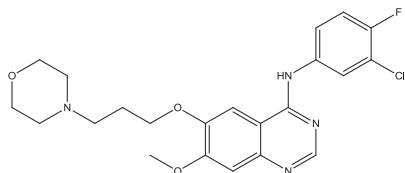
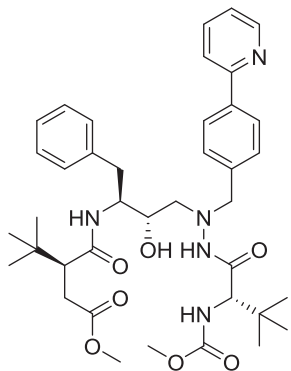
(Continued)

Table 1 Continued

Compound name	Structure of the compound	Therapeutic area	Function	Approval
Indinavir		Antiretroviral drug used to treat HIV/AIDS (1st generation)	HIV protease inhibitor	1996
Ritonavir		Antiretroviral drug used to treat HIV/AIDS (1st generation)	HIV protease inhibitor	1996
Zanamivir		Antiviral (to treat influenza A and influenza B)	Neuraminidase inhibitor	1999
Nelfinavir		Antiretroviral drug used to treat HIV/AIDS (1st generation)	HIV protease inhibitor	1999

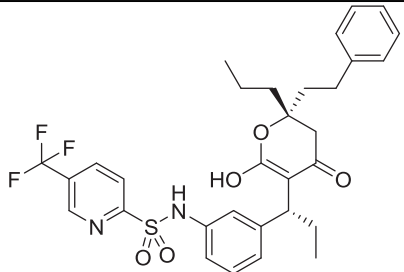
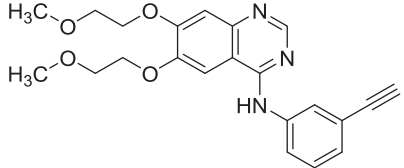
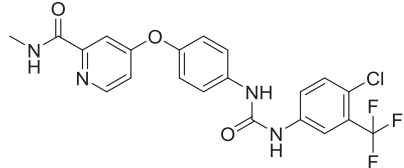
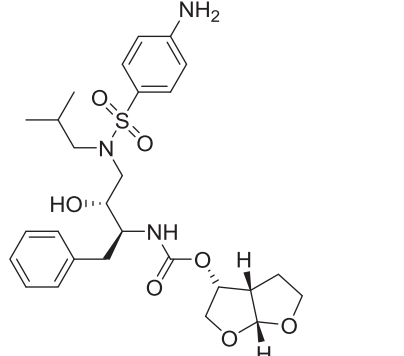
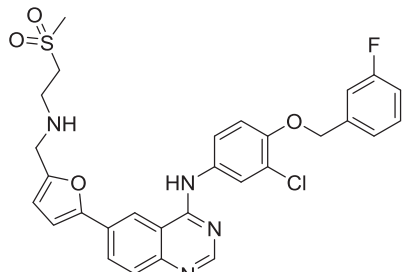
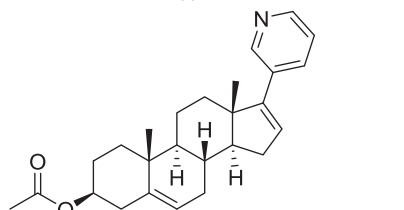
(Continued)

Table 1 Continued

Compound name	Structure of the compound	Therapeutic area	Function	Approval
Lopinavir		Antiretroviral drug used to treat HIV/AIDS against strains that are resistant to other protease inhibitors (1st generation)	Peptidomimetic HIV protease inhibitor	2000
Fosamprenavir		Antiretroviral prodrug used to treat HIV/AIDS (phosphorester that is rapidly and extensively metabolized to amprenavir) (1st generation)	HIV protease inhibitor	2003
Gefitinib		NSCLC	EGFR kinase inhibitor	2003
Atazanavir		Antiretroviral drug used to treat HIV/AIDS (2nd generation)	HIV protease inhibitor	2004

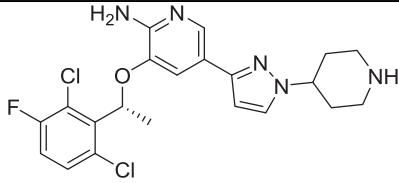
(Continued)

Table 1 Continued

Compound name	Structure of the compound	Therapeutic area	Function	Approval
Tipranavir		Antiretroviral drug used to treat HIV/AIDS (tipranavir is active against strains that are resistant to other protease inhibitors) (2nd generation)	Nonpeptidic HIV-1 protease inhibitor	2005
Erlotinib		NSCLC pancreatic cancer	EGFR kinase inhibitor	2005
Sorafenib		Renal cancer Liver cancer thyroid cancer	VEGFR kinase inhibitor	2005
Darunavir		Antiretroviral drug used to treat HIV/AIDS (2nd generation)	Nonpeptidic HIV-1 protease inhibitor	2006
Lapatinib		ERBB2-positive breast cancer	EGFR/ERBB2 inhibitor	2007
Abiraterone acetate		Metastatic castration-resistant prostate cancer or hormone-refractory prostate cancer	Androgen synthesis inhibitor	2011

(Continued)

Table 1 Continued

Compound name	Structure of the compound	Therapeutic area	Function	Approval
Crizotinib		NSCLC	ALK inhibitor	2011

100ACE, angiotensin-converting enzyme; HIV, human immunodeficiency virus; AIDS, acquired immunodeficiency syndrome; EGFR, epidermal growth factor receptor; NSCLC, non-small cell lung cancer; VEGFR, vascular epidermal growth factor receptor; ERBB2, erb-b2 receptor tyrosine kinase 2 (also known as NEU, NGL, HER2, TKR1, CD340, HER-2, MLN 19, HER-2/neu); ALK, anaplastic lymphoma kinase.

Adapted and modified from Prada-Gracia, D., Huerta-Yépez, S., Moreno-Vargas, L.M., 2016. Application of computational methods for anticancer drug discovery, design, and optimization. *Boletín Médico Del Hospital Infantil De México* 73 (6), 411–423.

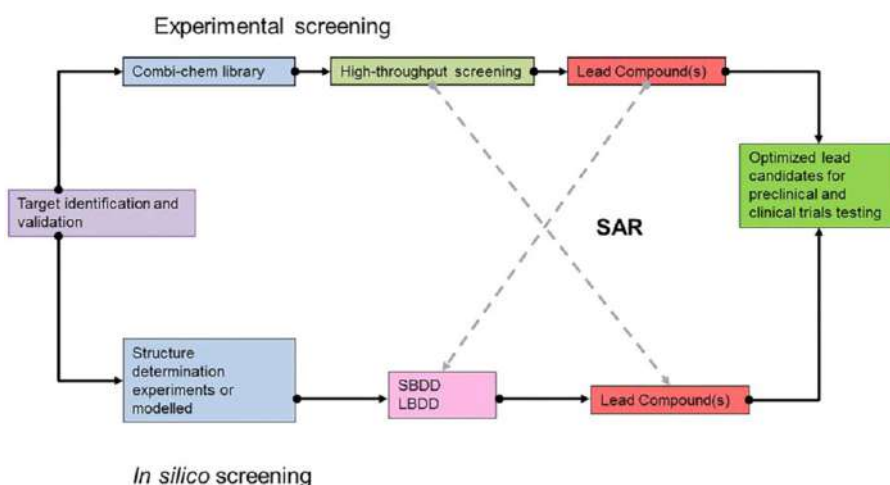


Fig. 2 Outline of the steps involved in drug development process. Adapted and modified from Kuhn, P., Wilson, K., Patch, M.G., Stevens, R.C., 2002. The genesis of high-throughput structure-based drug discovery using protein crystallography. *Current Opinion in Chemical Biology* 6 (5), 704–710.

(Allen and Tildesley, 1989). In protein dynamics, Monte Carlo simulation run is a sequence of random steps in conformation space inside a box where the step is decided on the basis of changes in the values of energy function during a simulation.

Metropolis criterion (MCM) simulations

Since MCM requires only energy function for evaluation and not the derivate of energy function unlike traditional MD which drives a system towards local energy minimum, conformational space is sampled faster with MCM than molecular dynamics. For flexible docking applications such as MCDOCK, MCM simulations have been adopted (Liu and Wang, 1999).

Rational drug design

Rational drug designing is based on the principle of logical reasoning before designing any therapeutic agents. For example, to prepare a competitive inhibitor relative to any specific target, the predicted structure should ensure that the designed molecule exhibits endogenous properties. The active site should be closely examined to get idea regarding the interacting amino acid residues, so that the nature and type of substituents and the favourable position in the molecule can be predicted, resulting in better binding.

Genetic algorithms

Genetic algorithms recombine parent conformations to child conformations which provides for molecular flexibility. The best scoring or “fittest” combinations are kept for another round of recombination during this evolutionary simulation process. Best possible set of solutions are evolved in the process retain favourable features from one generation to the next. Rational drug design does not involve exhaustive searches that involve impractical large combinatorial problems. The investigation of such problems is greatly aided by genetic algorithms; a class of algorithms mimicking some of the major characteristics of Darwinian evolution. In order to design small organic molecule with satisfying quantitative structure-activity relationship based rules (fitness), a specific

algorithm called an LEA (Ligand by Evolutionary Algorithm) has been conceived. The fitness consists of a sum of constraints that act as range properties. The LEA takes an initial set of fragments and repeatedly improves them by means of mutation and crossover operators which are related to those involved in the Darwinian model of evolution (Dougueta et al., 2000).

Molecular fingerprint and similarity searches

Molecular fingerprinting is a way of representing molecules which can be exploited to identify and compare structurally similar molecules or to cluster them on the basis of structural similarity. This method is based on fewer hypotheses and is less computationally taxing than QSAR or pharmacophore mapping models and is more quantitative in nature since it relies entirely on chemical structure and omits compounds with known biological activity. Moreover, fingerprint-based methods equally consider all parts of the molecule instead of focusing on only those parts of the molecule which are thought to be most important for activity. Hence, this method is less prone to error overfitting and requires smaller datasets to begin with. Fingerprint methods can also be used for searching databases for compounds that are similar in structure to a lead query, thereby providing an extended collection of compounds that can be tested for improved activity over the lead.

De Novo ligand designing

Constructing novel molecules from scratch is called de novo drug design. This approach is particularly challenging because the compound space to be searched can often become enormous. Defining a binding site for molecules in a target and determining what atoms or functional groups should be placed at certain loci at the binding site is one of the most critical and difficult steps of de novo drug design. (Klebe, 2000; Bohm and Stahl, 2000).

Case Studies

Discovery of tyrosine kinase inhibitors by docking into a molecular dynamics generated inactive kinase conformation

Type II inhibitors, which are the inactive DFG-out conformation of tyrosine kinases show good binding and selective profiles with several small molecules over other kinase targets. An explicit solvent molecular dynamics (MD) simulation of the complex of the catalytic domain of a tyrosine kinase receptor, ephrin type-A receptor 3 (EphA3), and manual docking of type II inhibitors was performed to obtain a set of DFG-out structures by Zhao et al. (2012). A single snapshot from the MD trajectory (for virtual screening) was selected using the automatic docking of four previously reported type II inhibitors (Fig. 3). Since a glycine-rich loop (Gloop), can adopt various conformations, it tightly encompasses the small type I head group of compound 1 and collapses into the ATP binding site. As a result, the bigger type I head groups of compounds 2–4 clashes with the G-loop in the ATP binding site. In addition, the entry of the piperazine group of compound 4 is blocked by the side chain of Tyr742 (Fig. 4). High-throughput docking of a pharmacophore-tailored library of 175,000 molecules results in about 4 million poses, which can be further filtered and sorted by van der Waals efficiency and force-field-based energy function. Remarkably, about 20% of the compounds with predicted binding energy smaller than $-10 \text{ kcal mol}^{-1}$ are known to be type II inhibitors and besides, a series of 5-(piperazine-1-yl)isoquinoline derivatives was identified as a novel class of low-micromolar inhibitors of both EphA3 and dephosphorylated Abelson tyrosine kinase (Abl1). A similar affinity to the gatekeeper mutant T315I of Abl1 is suggested by the in silico binding mode of the new inhibitors and competition binding assay studies. Additional evidence for the type II binding mode was obtained by two 300 ns MD simulations of the complex between N-(3-chloro-4-(difluoromethoxy)-phenyl)-2-(4-(8-nitroisoquinolin-5-yl)piperazin-1-yl)acetamide and EphA3 provided additional evidence of the type II binding mode.

Most type II inhibitors can be mapped well into three major hydrophobic interactions: ATP front site, ATP back site, and the allosteric site (Fig. 5).

Computer-Aided Drug Design (CADD)

CADD helps scientists in minimizing the synthetic and biological testing efforts by focussing only on the most promising compounds. Besides explaining the molecular basis of therapeutic activity, it also predicts possible derivatives that would improve activity. CADD entails (Kapetanovic, 2008):

- (1) Drug discovery and development processes being streamlined by the use of computing power.
- (2) Identification and optimization of new drugs using leverage of chemical and biological information about targets and/or ligands.
- (3) In silico designing of filters for the elimination of undesirable compounds with properties like poor activity and/or poor absorption, distribution, metabolism, excretion and toxicity, ADMET which facilitate selection of the most promising candidates.

Advantages of CADD

The main advantages of drug discovery through CADD are:

- (i) For experimental testing, smaller set of compounds are selected from large compound libraries.

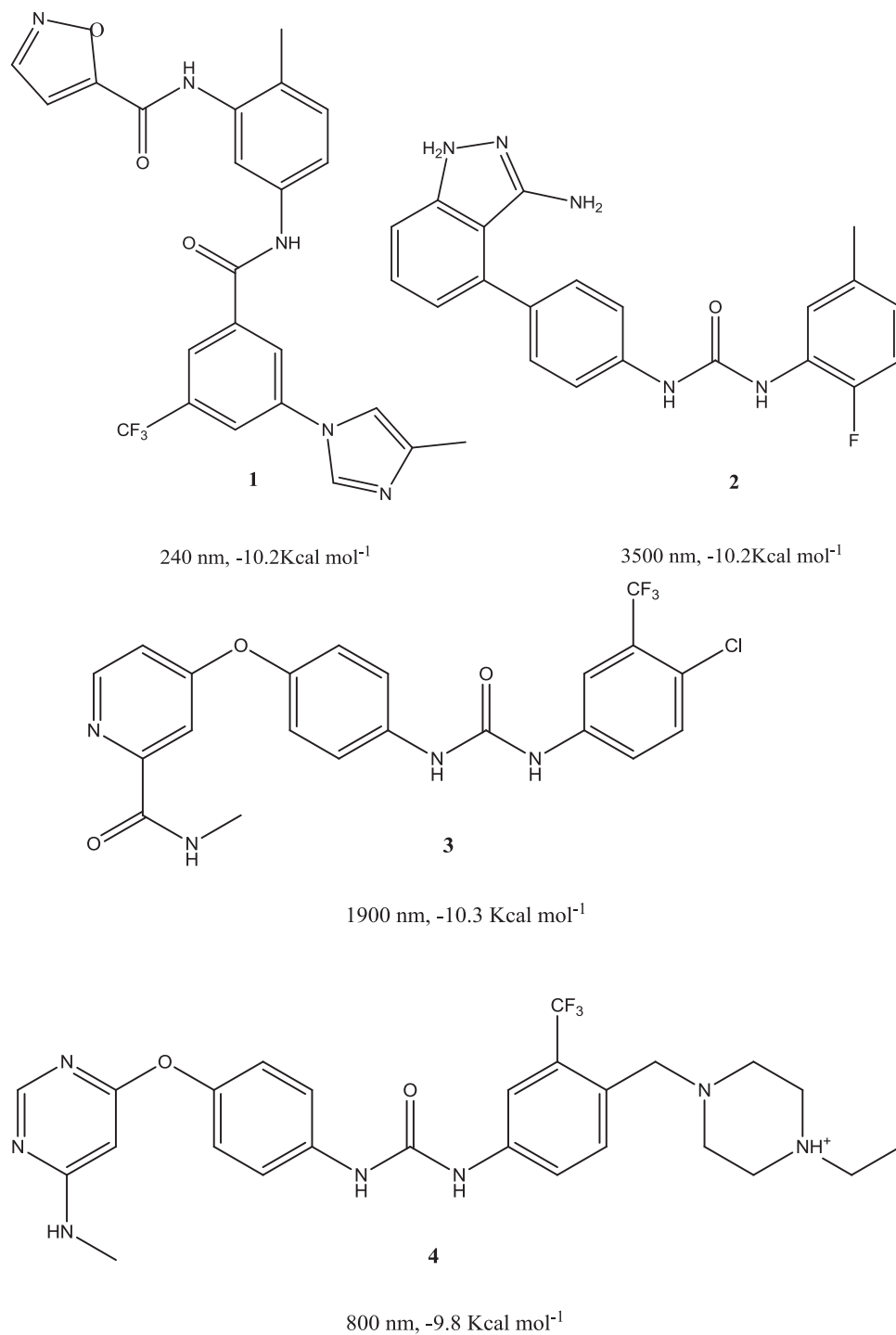


Fig. 3 Known type II inhibitors of EphA3. The values next to the compound number are the experimentally measured dissociation constant against phosphorylated EphA3 and the predicted binding free energy. These values were calculated by using the MD-IF structure and a scoring function with continuum solvation and hydrogen bonding penalty.

- (ii) Drug metabolism and pharmacokinetics (DMPK) properties like absorption, distribution, metabolism, excretion and the potential for toxicity (ADMET) are increased by optimization of lead compounds.
- (iii) Designing of novel compounds can be achieved either by “growing” starting molecules one functional group at a time or by piecing together fragments into novel chemotypes (Veselovsky and Ivanov, 2003).
- (iv) Traditional experimentation which requires animal and human models can be replaced by CADD, saving both time and cost (Mallipeddi *et al.*, 2014).

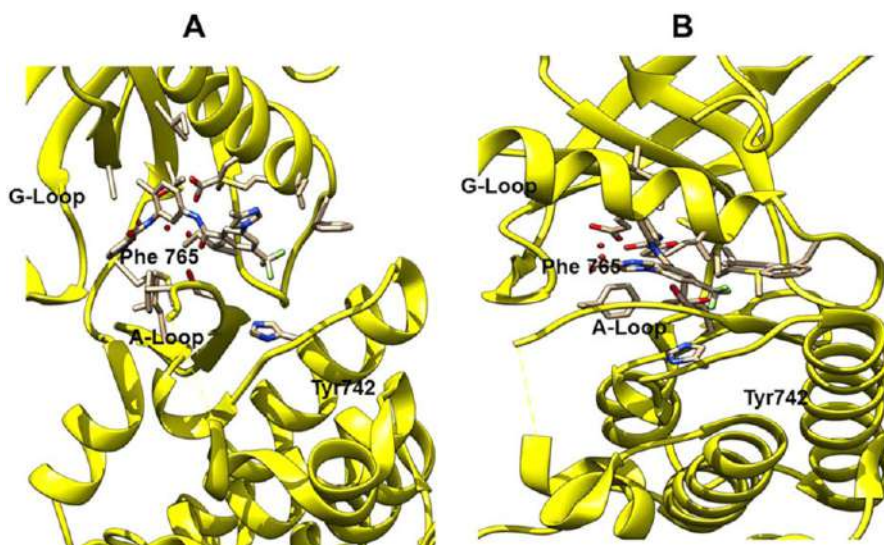


Fig. 4 Comparison of A) the crystal structure (PDB: 3DZQ) of the complex of EphA3 with inhibitor 1 and B) the binding mode obtained by docking compound 4 into the MD-IF structure. For the differences in orientations of Tyr 742 and Phe 765, and in the G-loop. Adapted and modified from Zhao, H., Huang, D., Caffisch, A., 2012. Discovery of tyrosine kinase inhibitors by docking into an inactive kinase conformation generated by molecular dynamics. *ChemMedChem* 7 (11), 1983–1990.

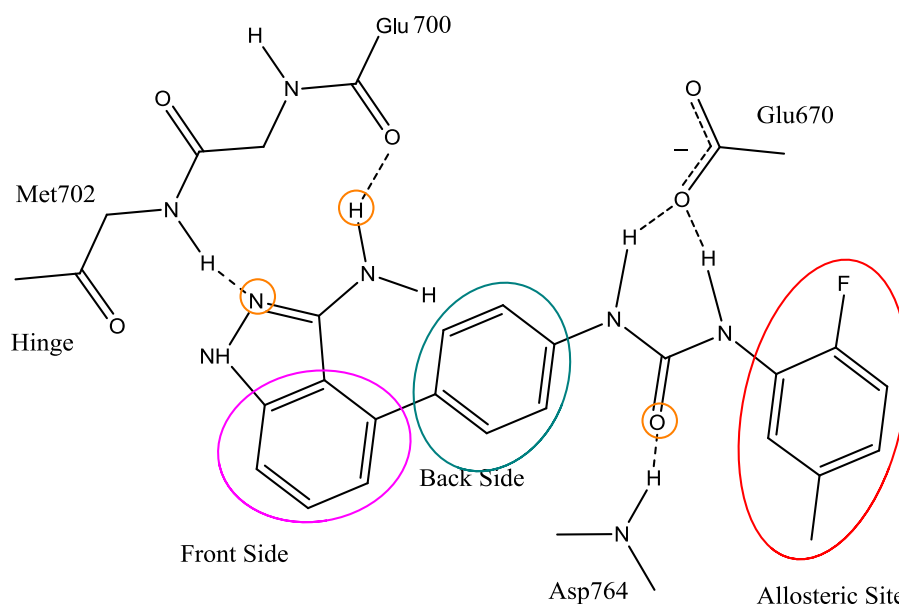


Fig. 5 Key interactions pharmacophore mapping of type II kinase inhibitors illustrated by compound 2 and EphA3. Dashed lines are hydrogen bonds. Pharmacophore elements used to filter the ZINC library: two acceptors (dashed circles), one donor (solid circle), and three hydrophobic rings (ovals). Because of the flexibility and solvent exposure of the Glu 670 side chain, the hydrogen bond to Glu 670 was not used as a pharmacophore.

- (v) Reduces the chances of drug resistance and thus would lead to production of lead compounds which would target the causative factor.
- (vi) CADD also leads to the construction of high quality datasets and libraries that can be optimized for high molecular diversity or similarity (Ou-Yang *et al.*, 2012).

Types of CADD

The choice of CADD approaches to be employed is determined by the availability of the experimentally determined 3D structures of target proteins. Structure-based CADD uses our knowledge of the target protein structure to calculate interaction energies,

whereas in ligand-based CADD, chemical similarity searches or construction of predictive, quantitative structure-activity relationship (QSAR) models exploits our knowledge of known active and inactive molecules. (Kalyanamoorthy and Chen, 2011). Structure based CADD combines information from several fields, for example, X-ray crystallography and/or NMR, synthetic organic chemistry, molecular modelling, QSAR, and biological evaluation (Marrone *et al.*, 1997). Through structure based CADD, we aim to design compounds with strong binding affinity with the target, thereby exhibiting properties like reduced free energy, improved DMPK/ADMET properties and target specification i.e., reduced off-target (Jorgensen *et al.*, 2010). Virtual high-throughput screening (vHTS) also known as screening of virtual compound libraries is one of the most common applications of CADD (Kalyanamoorthy and Chen, 2011). Fig. 6 represents an overview of CADD drug designing/design pipeline.

Structure-based drug discovery

This method exploits knowledge of the three-dimensional structure of a receptor complexed with a lead molecule for optimization of the bound ligand or a series of congeneric molecules. It requires the understanding of receptor-ligand interactions. The structural information can be obtained either from X-ray crystallography, NMR, or from homology modelling. A medicinal chemist can use a model with a given structure for computing the activity of a molecule (Lewis, 2005). Some of these approaches provide accurate binding modes, while cater to fast searching of large databases. Some approaches of structure-based drug designing are explained below.

Structure-based virtual high-throughput screening

Structure-based virtual high-throughput screening (SB-vHTS) is an *in-silico* method which helps identify putative hits out of hundreds of thousands of compounds to the targets of known structure. It is usually based on molecular docking. In molecular docking, a small molecule is fitted into the active site of protein model and here, comparison of the 3D structure of small molecule with the putative binding pocket is carried out. In the traditional HTS, the general ability of a ligand to bind, inhibit or allosterically alter the proteins function is asserted experimentally, whereas in SB-vHTS selects the ligands that are predicted to bind to a specific binding site. To ensure the feasibility of screening of large compound libraries within a finite time, limited conformational sampling of proteins and ligands is used by SB-vHTS along with a simplified approximation of binding energy that can be computed rapidly (Becker *et al.*, 2006).

Structure-based virtual screening

This is a computational approach for identifying potential drug candidates (hits) that are capable of binding to a drug target (protein receptors, enzymes). This method involves quick searching of large libraries of chemical followed by docking of the hit into a protein target and finally application of a scoring function for estimating the probability of binding affinity of drug candidate with the protein target (Cheng *et al.*, 2012). The most important advantage of this screening is that it enhances the hit rate by considerably decreasing the number of compounds that are estimated experimentally for their activity and hence improves

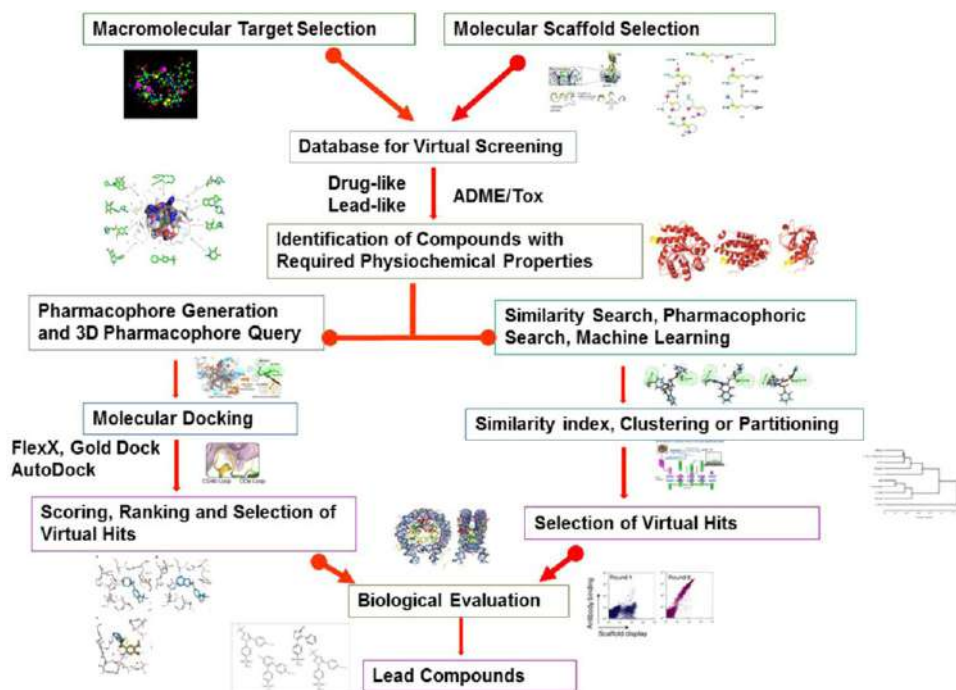


Fig. 6 An overview of CADD drug designing/design pipeline. Adapted and modified from Guido, R.V., Glaucius Oliva, G., Andricopulo, A.D., 2008. Virtual screening and its integration with modern drug design technologies. Current Medicinal Chemistry 15, 37–46.

the success rate of the *in vitro* experiments. This method has been applied extensively in pharmaceutical companies and academic groups for early-stage drug discovery.

Fragment-based lead discovery

This approach is based upon structure-activity relationships (SAR), obtained from NMR for identifying and optimizing the lead (Bienstock, 2011). High purity, weak potency but effective binding, good aqueous solubility, (molecular weight < 300, ClogP < 3, number of rotatable bonds, number of hydrogen bond donors and acceptors each should be < 3) are the criteria for selecting the chemical fragments (Congreve *et al.*, 2003). Later, these fragments are either expanded or combined for producing a lead with a higher affinity.

In silico structure-based lead optimization

After the desired hits are identified through virtual screening, this method speeds up the search for optimized lead by delineating the prediction about its pharmacological properties, thereby reducing the *in vitro* and *in vivo* experimental time.

ADMET modelling

This method, a common name for which is physiologically-based pharmacokinetic modelling is used in drug design and development, and in assessing of toxicity threat evaluation and specifically predicts absorption, distribution, metabolism, excretion and toxicology (ADMET) of drugs/compounds in humans. The ADMET parameters are based on the kinetics of the drug exposure to tissues and how the body will react to them, influencing the performance and pharmacological activity of the compound. Therefore, this method provides a key insight into the behaviour of a pharmaceutical compound within an organism. This approach aids in the selection of compounds during the very early phases of drug thereby playing a crucial role in drug discovery and development. This technique is cost- and time effective owing to a reduction in attrition of drugs during the pre-clinical / clinical phase trials at a later stage.

Ligand-based drug designing

The existing knowledge of active compounds against the target is used to predict new chemical entities that present similar behaviour in Ligand-based methods (Martin *et al.*, 2002). Given a single known active molecule, a pharmacophore model can be derived from a library of molecules to define the minimum necessary structural characteristics a molecule must possess in order to bind to the target of interest. A fingerprint-based similarity search is usually used to compare the active molecule to the library as here, the molecules are represented as bit strings which represent the presence or absence of predefined structural descriptors (Mishra and Siva-Prasad, 2011). In comparison, targeting structural information to determine whether a new compound is likely to bind and interact with a receptor is the method that structure-based methods rely on. No prior knowledge of active ligands is required in this method, which is a significant advantage (Kolb *et al.*, 2009). It is possible to design new ligands that can elicit a therapeutic effect from 3D structures. Therefore, the development of new drugs through the discovery and optimization of the initial lead compound are greatly impacted by structure-based approaches.

Ligand-based virtual screening (LBVS)

Ligand-based virtual screening is based on the "similarity principle" according to which similar molecules tend to exhibit similar biological properties. Scaffold hopping i.e., identification of iso-functional molecular structures with significantly different molecular backbones is the usual objective when using LBVS. "Scaffold hopping" is also known as "leapfrogging", "scaffold searching" and "leap hopping" (Kalliokoki, 2010). These methods are usually helpful in drug repurposing, wherein new targets and diseases are pursued for existing drug molecules.

Molecular descriptors

This is one of the simplest approaches in which the reference molecule/set of molecules are compared with a large library of compounds at a very low cost on the basis of physicochemical properties descriptors, such as molecular weight, volume, geometry, surface areas, atom types, dipole moment, polarizability, molar refractivity, octanol-water partition coefficient (log P), planar structures, electronegativity, or solvation properties that are obtained from experimental measurements or theoretical models. Molecules are represented by symbols for effective execution of the task (Prada-Graciaa *et al.*, 2016).

Quantitative structure-activity relationship models (QSAR)

The mathematical relation between structural attributes and target response for a set of chemicals are explained by Quantitative Structure-Activity Relationship models. Structural and/or property descriptors of compounds can also be correlated with their biological activities using QSAR (Bernard *et al.*, 2005). Through QSAR models we can correlate various features like rate constants, binding sites affinities of ligands, inhibition constants and other biological activities, either with certain structural features (Free Wilson analysis) or with atomic, molecular or group properties, such as lipophilicity, electronic, steric and polarizability, among congeneric series of compounds. (Kubinyi, 1995). Hence, the success of QSAR is dependent on the choice of descriptors and the ability to generate the appropriate mathematical relationship besides the quality of initial set of active/inactive compounds.

Pharmacophore modelling

More significant information can be drawn by employing various conformations of a range of ligands than just a single ligand structure. A pharmacophore model of the receptor site can be generated with a sufficiently broad range of ligands. Pharmacophore

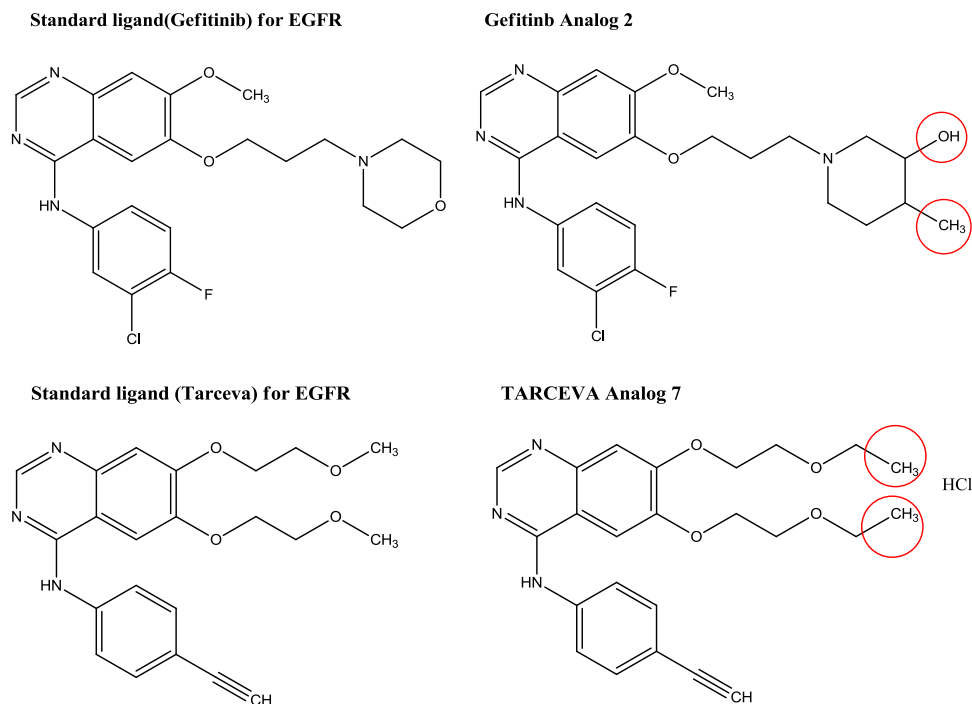


Fig. 7 Structures of Gefitinib and Tarceva and their analogs.

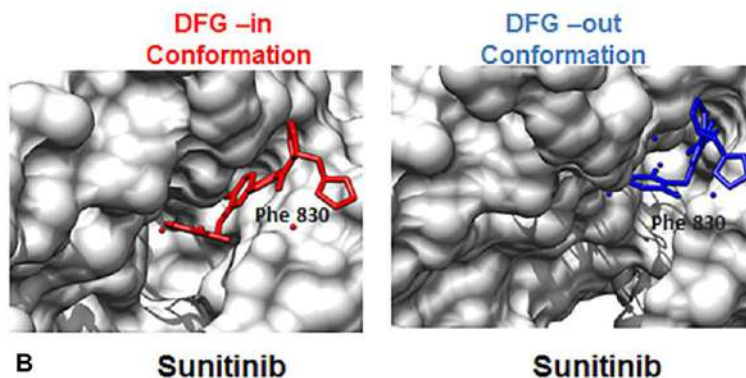
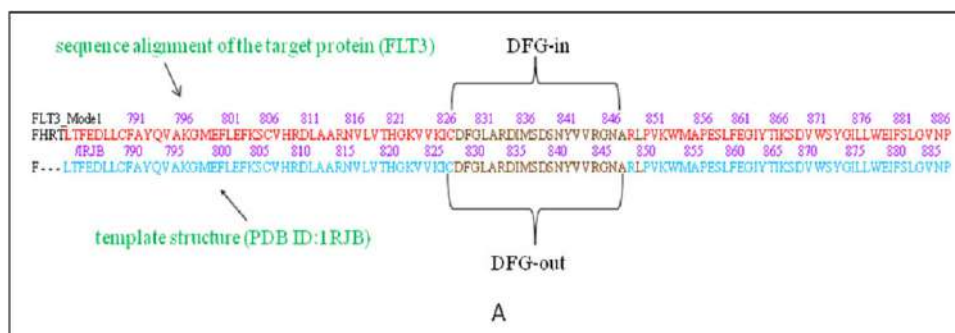


Fig. 8 (A) The sequence alignment of the target protein (FLT3) with the template structure in the DFG loop region. (B) Comparison of the DFG-in and DFG-out FLT3 structures. Adapted and modified from Ke, Y.Y., Singh, V.K., Coumar, M.S., *et al.*, 2015. Homology modeling of DFG-in FMS-like tyrosine kinase 3 (FLT3) and structure-based virtual screening for inhibitor identification. Scientific Reports 5, 11702. Doi: 10.1038/srep11702.

modelling of smaller, non-peptide molecules that might have improved stability and bioavailability over their peptide counterparts has resulted in successful outcomes so far (Nielsen *et al.*, 1999).

Case Studies

Computer aided drug designing (CADD) for EGFR protein controlling lung cancer

Epidermal growth factor receptor (EGFR), a receptor tyrosine kinase plays an important role in tumour cell survival. Phosphorylated EGFR upon activation causes phosphorylation of downstream proteins that lead to changes in cell proliferation, invasion, metastasis, and inhibition of apoptosis. Human EGFR was taken as a protein by Baskaran *et al.* (2012), and commercially available drugs (such as Gemzar, Gefitinib, Tarceva) were taken as ligands. The drugs were docked to this receptor and Gefitinib and Tarceva were chosen based on the energy values. To improve the binding efficiency and steric compatibility of these two drugs, modifications were made to the probable functional groups which were interacting with the receptor molecule. Analogs were prepared using ACD Chem Sketch in MOL format which were then converted to 3D structure using Weblab Viewer Lite, a 3D modelling program. It was then docked using Vega ZZ docking software. Gefitinib Analog 2(281) and Tarceva Analog 7 had significantly lower energy values and were found to be better than the conventional drugs available (Fig. 7).

Structure-based virtual screening (SBVS), for the identification of new chemotypes for FLT3 inhibition

FMS-like tyrosine kinase 3 (FLT3) is a type III tyrosine kinase receptor and is highly expressed in hematopoietic stem and progenitor cells. FLT3 binds to extracellular domain which activates cytoplasmic tyrosine kinase activity leading to activation of

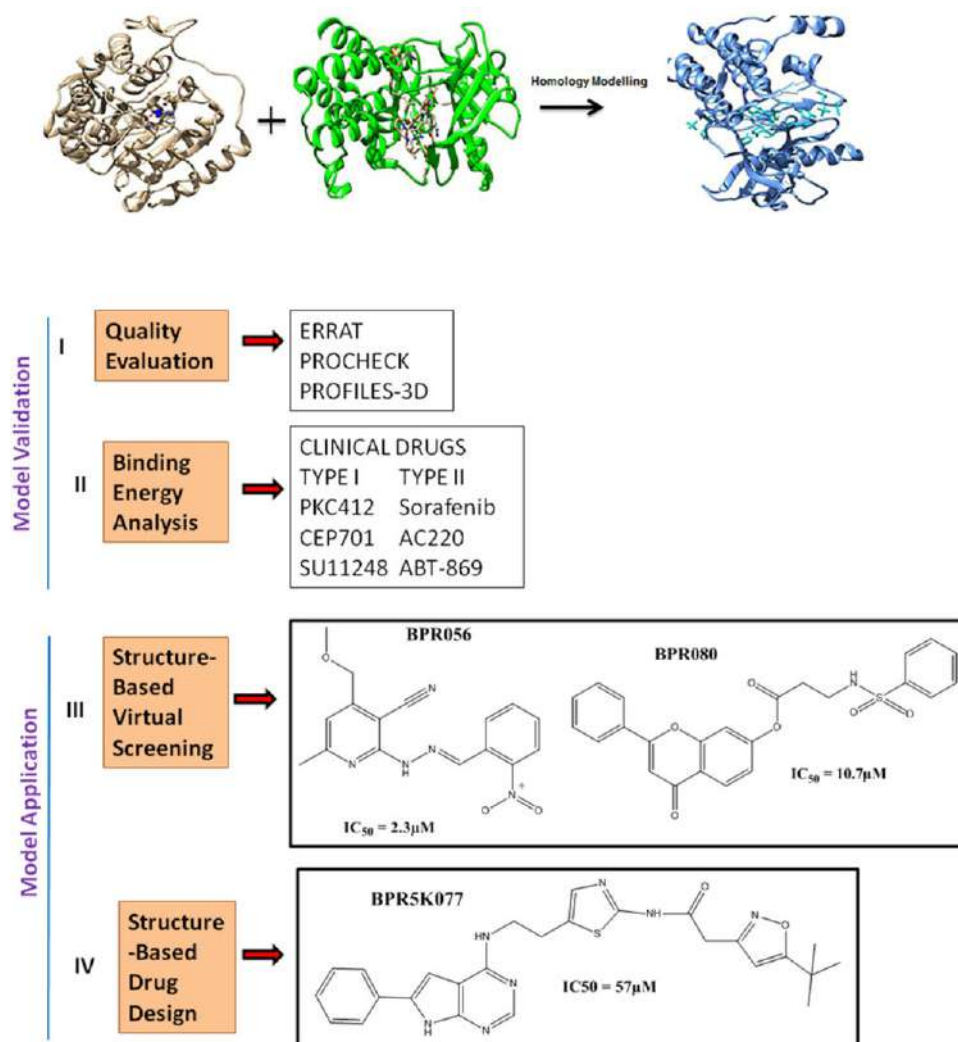


Fig. 9 Computer-aided drug design (CADD) strategy for FLT3 inhibitor identification. Adapted and modified from Ke, Y.Y., Singh, V.K., Coumar, M.S., *et al.*, 2015. Homology modeling of DFG-in FMS-like tyrosine kinase 3 (FLT3) and structure-based virtual screening for inhibitor identification. Scientific Reports 5, 11702. Doi: 10.1038/srep11702.

downstream cellular signalling which is essential for proliferation. Based on the position of the Phe residue of the DFG motif, the inhibitors are named Type I or Type II inhibitors. In a DFG-in conformation, the Phe residue is oriented outside the ATP binding site while in a DFG-out conformation; the Phe residue is oriented inside the ATP binding site. Inhibitors that bind to the DFG-in conformation are termed type-I inhibitors, and those that bind to the DFG-out conformation are referred to as type-II inhibitors. In Fig. 8, Ke *et al.* (2015), superimposed the FLT3-modeled (DFG-in) structure on the 1RJB (DFG-out) template structure to depict the structural differences between the DFG-in and DFG-out conformations. Since type-I inhibitors bind strongly to FLT3-ITD-mutated kinase, they are proven to be more effective against treatment of acute myeloid leukemia.

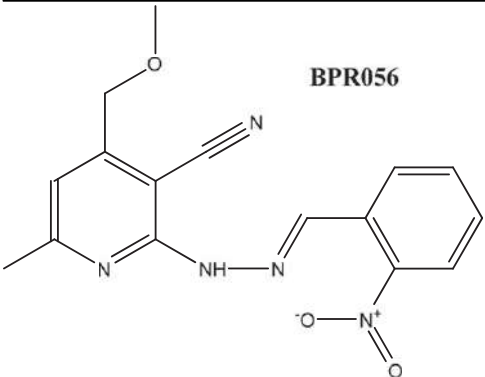
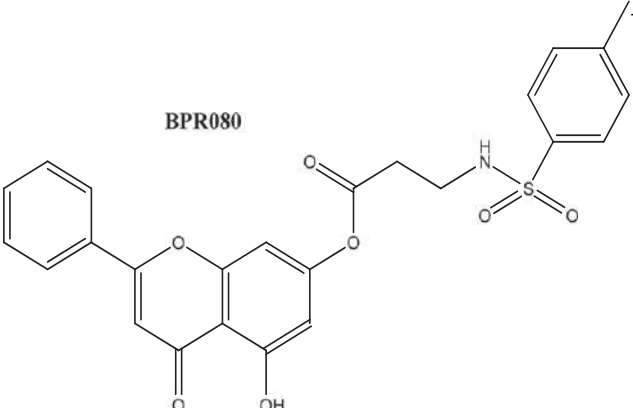
Homology modelling (HM) of the DFG-in FLT3 structure identifies known DFG-in (SU11248, CEP-701, and PKC-412) and DFG-out (Sorafenib, ABT-869 and AC220) FLT3 inhibitors using docking studies (citation of the study). SBVS of an HTS library of 125,000 compounds was done with the modeled structure. Out of the 97 top scoring compounds, two hits (BPR056, $IC_{50}=2.3$ and BPR080, $IC_{50}=10.7 \mu M$) were identified. Based on molecular dynamics simulation and density functional theory calculations BPR056 (MW: 325.32; cLogP: 2.48) was identified to interact with FLT3 in a stable manner and was studied to be chemically optimizable to realize a drug-like lead in the future. The overall CADD strategy used in this study is shown in Fig. 9.

This superimposition reveals that the activation loop in the DFG-in structure flipped away from the active site which leads to the placement of the Phe830 group away from this site. In the DFG-out structure, the activation loop overturned towards the active site, resulting in the placement of Phe830 in the active site. The different arrangements of the activation loop, particularly in regards to the placement of the Phe830 group, have profound implications for the binding of different types of inhibitors to FLT3 kinase.

More than 40% inhibition at a $10 \mu M$ concentration was observed for BPR056 and BPR080, out of 97 screened compounds (Table 2).

In order to design better FLT3 inhibitors it is important to understand the binding modes of these two high scoring poses in the modeled structure. Both the molecules were observed to bind at the ATP binding site by forming Hydrogen bonds with the

Table 2 FLT3 kinase inhibition profiles, docking score and binding energies of the hits identified from the in-house HTS database. PKC412 and sorafenib are used as reference compounds for the comparison

Compound	FLT3 IC_{50} (nM)	Docking score Predicted binding energy (kcal/mol)			
		DFG-in	DFG-out	DFG-in	DFG-out
 BPR056	2300	-355	-276	-23.04	16.41
 BPR080	10,700	-350	-314	-22.92	19.82
Midostaurin (PKC412)	37	-269	-165	-34.21	19.13
Sorafenib (BAY43-9006)	102	-183	-300	-7.06	-12.32

Adapted and modified from Ke, Y.Y., Singh, V.K., Coumar, M.S., *et al.*, 2015. Homology modeling of DFG-in FMS-like tyrosine kinase 3 (FLT3) and structure-based virtual screening for inhibitor identification. Scientific Reports 5, 11702. Doi: 10.1038/srep11702.

hinge-region Cys694 residue. A hinge-region H-bond interaction between an inhibitor and kinase is an essential component in inhibitory activity and are common in several inhibitor–kinase complex structures.

DFG and FLT3 inhibitor complex was subjected to 20 ns using the Gromacs MD suite. The root mean square deviation showed that the both cases found to be at equilibrium from initial position and remained stable after 10 ns. In addition RMSF values of complex were also determined and it showed that the major differences occurs between the both complexes occurred in the loop region and more fluctuations in the FLT3-BPR080 interacting complex (Ke *et al.*, 2015). Detailed docking, DFT calculations and 20-ns MD simulations of the new hits in the DFG-in FLT3-modeled structure suggest that the interaction between BPR056 (MW: 325.32; cLogP: 2.48) and FLT3 is stable and could, in the future, be chemically optimized to realize a drug-like lead.

Fragment-based drug design to target the EphA4 kinase domain

The first approved drug that was developed via fragment-based approach is Vemurafenib (4, Fig. 10), a kinase inhibitor of the B-Raf V600E oncogenic mutant. High content screening (HCS), a type of phenotypical screening, followed by X-ray crystallography yielded an unselective 7-Azaindole fragment (Figure 12) as hit fragment, which was optimized via 2 into the selective B-Raf inhibitor PLX472077 (3) to end up with Vemurafenib (PLX4032, 4) (Bollag *et al.*, 2010).

Limitations of CADD

Despite the emergence of the numerous above-mentioned pioneering approaches, drug design and development is still an inherently risky business where the input costs are high and the success rate is low. Generally, only one in 1000 lead compounds reaches phase 1 clinical trials and only one in five drugs make it from phase 1 trials into the marketplace (Wishart, 2006). Some procedures concerning Computer Aided Drug Designing are time consuming, especially while looking for a proper lead component (Jorgensen, 2004). The current molecular docking algorithms do not estimate the absolute energy associated with the intermolecular interaction with satisfactory accuracy as most of the available scoring functions are classical approximations of events ruled by quantum mechanics. Further, there is lack of accurate experimental data that restricts further advancement of CADD (Bharath *et al.*, 2011). Hence, new experimental or computational tools and scientific approaches to identify correlations between the nature and structure-based properties of the drug and of its safety and efficacy in the human body (pharmacovigilance-based) are in vital need of improvement so

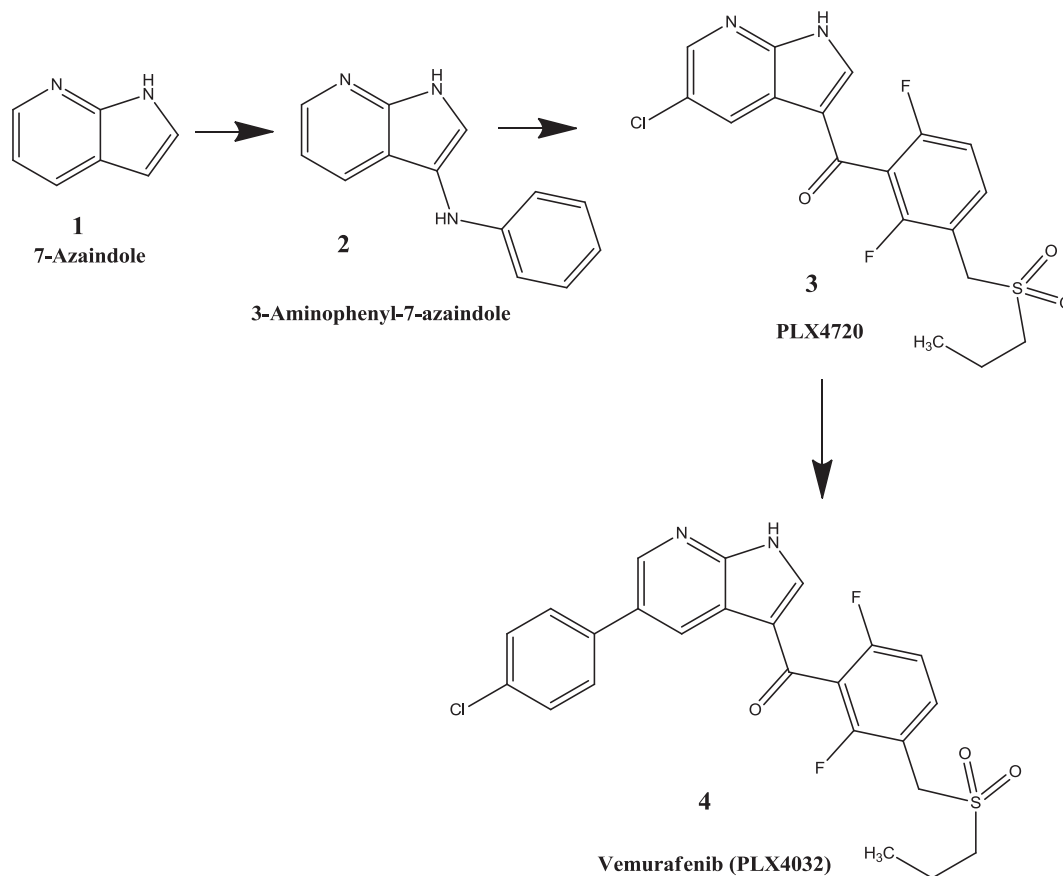


Fig. 10 Optimization of the non-selective 7-Azaindole fragment (24) into Vemurafenib (PLX4032, 27).

potentially problematic drug leads could be identified at the early stages in their development. This will improve the public health and provide a safe, effective and rational use of medicines and their development.

Conclusion & Future Perspectives

CADD has now become an indispensable tool in the long process of drug discovery and development. It also provides options for understanding chemical systems in different ways, yielding information that is not easy to obtain in laboratory analysis, with considerably less cost and effort than experiments. Initially, CADD had a rocky perception in the field of drug development, and perhaps some over-hyping of its promises, as is present in the initial stages of almost any new technology or development. Today, one can say that the discipline of computational medicinal chemistry has begun to mature and is a routinely used component of modern drug discovery process. Mastering different kinds of CADD approaches and their software and utilizing all computational resources that are valuable for drug design are certainly essential for becoming a successful computational medicinal chemist in today's world. In addition, having skills in one or more programming languages, such as Python or JAVA will help smooth routine drug-design work. SBVs and LBVs are also very likely to become routine in drug-discovery projects if they are not considered to have already done so. The use of more accurate methods like MD and QM, continue to grow. In conclusion, CADD is beneficial for pharmaceutical development in the areas of prediction of 3D structures, design of compounds, prediction of druggability, in silico ADMET prediction however, it must be realised that computational predictions need to be integrated with experimental approaches for successful drug discovery and development.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Comparative Epigenomics. Computational Protein Engineering Approaches for Effective Design of New Molecules. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow. Study of The Variability of The Native Protein Structure

References

- Abrams, C.F., Vanden-Eijnden, E., 2010. Large-scale conformational sampling of proteins using temperature-accelerated molecular dynamics. *Proceedings of National Academy of Sciences of the United States of America* 107, 4961–4966.
- Allen, M.P., Tildesley, D.J., 1989. *Computer Simulation of Liquids*. Oxford, UK: Oxford Science Publications, p. 385.
- Baskaran, C., Bai, V.R., Kumaran, K., Kumar, K.M., 2012. Computer aided drug designing (CADD) for EGFR protein Controlling lung cancer. *Annals of Biological Research* 3 (4), 1815–1820.
- Becker, O.M., Dhanoa, D.S., Marantz, Y., *et al.*, 2006. An integrated in silico 3D model-driven discovery of a novel, potent, and selective amidosulfonamide 5-HT_{1A} agonist (PRX-00023) for the treatment of anxiety and depression. *Journal of Medicinal Chemistry* 49, 3116–3135.
- Bernard, D., Coop, A., MacKerell Jr., A.D., 2005. Computer-aided drug design: Structure-activity relationships of delta opioid ligands. *Drug Design Review* 2, 277291.
- Bharath, E.N., Manjula, S.N., Vijaychand, A., 2011. In silico drug design tool for overcoming the innovation deficit in the drug discovery process. *International Journal of Pharmacy and Pharmaceutical Sciences* 3 (2), 1–5.
- Bienstock, J.R., 2011. Overview: Fragment-based drug design. *ACS Symposium Series* 1076, 1–26.
- Blaney, F., 1990. Molecular modelling in the pharmaceutical industry. *Chemistry and Industry (London)* 23, 791–794.
- Bohm, H.J., Stahl, M., 2000. Structure-based library design: Molecular modeling merges with combinatorial chemistry. *Current Opinion in Chemical Biology* 4, 283–286.
- Bollag, G., Hirth, P., Tsai, J., *et al.*, 2010. Clinical efficacy of a RAF inhibitor needs broad target blockade in BRAF-mutant melanoma. *Nature* 467, 596–599.
- Buchan, D.W., Ward, S.M., Loble, A.E., *et al.*, 2010. Protein annotation and modelling servers at University College London. *Nucleic Acids Research* 38, 563–568.
- Bugg, C.E., Carson, W.M., Montgomery, J.A., 1993. Drugs by design. *Scientific American Magazine*, 60–66.
- Cheng, T., Li, Q., Zhou, Z., Wang, Y., Bryant, S.H., 2012. Structure-based virtual screening for drug discovery: A problem-centric review. *American Association of Pharmaceutical Scientists (AAPS) Journal* 14, 133–141.
- Congreve, M., Carr, R., Murray, C., Jhoti, H., 2003. A 'rule of three' for fragment based lead discovery? *Drug Discovery Today* 8, 876–877.
- Congreve, M., Murray, C.W., Blundell, T.L., 2005. Structural biology and drug discovery. *Drug Discovery Today* 10 (13), 895–907.
- Dougueta, D., Thoreau, E., Grassy, G., 2000. A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *Journal of Computer-Aided Molecular Design* 14, 449–466.
- Grubmüller, H., 1995. Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Physical Review E. Statistical, Physics, Plasmas, Fluids and Related Interdisciplinary Topics* 52, 2893–2906.
- Guido, R.V., Glaucius Oliva, G., Andricopulo, A.D., 2008. Virtual screening and its integration with modern drug design technologies. *Current Medicinal Chemistry* 15, 37–46.
- Huey, R., Morris, G.M., Olson, A.J., Goodsell, D.S., 2007. A semiempirical free energy force field with charge-based desolvation. *Journal of Computational Chemistry* 28, 1145–1152.
- Jorgensen, W.L., 2010. Drug discovery: Pulled from a protein's embrace. *Nature* 466, 42–43.
- Jorgensen, W.L., 2004. The many roles of computation in drug discovery. *Science* 303, 1813–1817.
- Kalliokoki, T., 2010. Accelerating three dimensional virtual screening. *Dissertations In Health Sciences*.
- Kalyanamoorthy, S., Chen, Y.P., 2011. Structure-based drug design to augment hit discovery. *Drug Discovery Today* 16, 831–839.
- Kapetanovic, I.M., 2008. Computer-aided drug discovery and development (CADD) – In silico-chemical biological approach. *Chemico-Biological Interactions* 171, 165–176.
- Ke, Y.Y., Singh, V.K., Coumar, M.S., *et al.*, 2015. Homology modeling of DFG-in FMS-like tyrosine kinase 3 (FLT3) and structure-based virtual screening for inhibitor identification. *Scientific Reports* 5, 11702. doi:10.1038/srep11702.
- Kitchen, D.B., Decornez, H., Furr, J.R., Bajorath, J., 2004. Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nature Reviews Drug Discovery* 3, 935–949.
- Klebe, G., 2000. Recent developments in structure-based drug design. *Journal of Molecular Medicine* 78, 269–281.
- Kolb, P., Ferreira, R.S., Irwin, J.J., Shoichet, B.K., 2009. Docking and chemoinformatic screens for new ligands and targets. *Current Opinion in Biotechnology* 20, 429–436.

- Kubinyi, H., 1995. In Burger's Medicinal Chemistry. Wolff, M.E. (Ed.), vol. 1. John Wiley & Sons, pp. 497–571.
- Kuhn, P., Wilson, K., Patch, M.G., Stevens, R.C., 2002. The genesis of high-throughput structure-based drug discovery using protein crystallography. *Current Opinion in Chemical Biology* 6 (5), 704–710.
- Lewis, R.A., 2005. A general method for exploiting QSAR models in lead optimization. *Journal of Medicinal Chemistry* 48 (5), 1638–1648.
- Liu, M., Wang, S.M., 1999. MCDock: A Monte Carlo simulation approach to the molecular docking problem. *Journal of Computer-Aided Molecular Design* 13, 435–451.
- Lombardino, J.G., Lowe, J.A., 2004. The role of the medicinal chemist in drug discovery — then and now. *Nature Reviews Drug Discovery* 3, 853–862.
- Maithri, G., Manasa, B., Vani, S.S., Narendra, A., Harshita, T., 2016. Computational drug design and molecular dynamic studies — A review. *International Journal of Biomedical Data Mining*. 1–7.
- Mallipeddi, P.L., Kumar, G., White, S.W., Webb, T.R., 2014. Recent advances in computer-aided drug design as applied to anti-influenza drug discovery. *Current Topics in Medicinal Chemistry* 14, 1875–1889.
- Marrone, T.J., Briggs, J.M., McCammon, J.A., 1997. Structure-based drug design: Computational advances. *Annual Review of Pharmacology and Toxicology* 37, 71–90.
- Martin, Y.C., Kofron, J.L., Linda, M., Traphagen, L.M., 2002. Do Structurally Similar Molecules Have Similar Biological Activity? *Journal of Medicinal Chemistry* 45, 4350–4358.
- Martí-Renom, M.A., Stuart, A.C., Fiser, A., *et al.*, 2000. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure* 29, 291–325.
- Martis, E.A., Radhakrishnan, R., Badve, R.R., 2011. High-throughput screening: The hits and leads of drug discovery — An overview. *Journal of Applied Pharmaceutical Science* 1 (1), 2–10.
- Mishra, V., Siva-Prasad, C.V., 2011. Ligand based virtual screening to find novel inhibitors against plant toxin Ricin by using the ZINC database. *Bioinformation* 7, 46–51.
- Moore, D.J., West, A.B., Dawson, V.L., Dawson, T.M., 2005. Molecular pathophysiology of Parkinson's disease. *Annual Review of Neuroscience* 28, 57–87.
- Morris, G.M., Goodsell, D.S., Halliday, R.S., *et al.*, 1998. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry* 19, 1639–1662.
- Nielsen, K.J., Adarns, D., Thomas, L., *et al.*, 1999. Structure-activity relationships of omega-conotoxins MVIIA, MVIIIC and 14 loop splice hybrids at N and P/Q-type calcium channels. *Journal of Molecular Biology* 289 (5), 1405–1421.
- Ou-Yang, S.S., Lu, J.Y., Kong, X.Q., *et al.*, 2012. Computational drug discovery. *Acta Pharmacologica Sinica* 33, 1131–1140.
- Prada-Gracia, D., Huerta-Yépez, S., Moreno-Vargas, L.M., 2016. Application of computational methods for anticancer drug discovery, design, and optimization. *Boletín Médico Del Hospital Infantil De Mexico* 73 (6), 411–423.
- Schlitter, J., Engels, M., Krüger, P., 1994. Targeted molecular dynamics: A new approach for searching pathways of conformational transitions. *Journal of Molecular Graphics* 12, 84–89.
- Schneider, G., Bohm, H.J., 2002. Virtual screening and fast automated docking methods. *Drug Discovery Today* 7, 64–70.
- Shoichet, B.K., 2004. Virtual screening of chemical libraries. *Nature* 432, 862–865.
- Sugita, Y., Okamoto, Y., 1999. Replica-exchange molecular dynamics method for protein folding. *Chemistry Physics Letters* 314, 141–151.
- Veslovsky, A.V., Ivanov, A.S., 2003. Strategy of computer-aided drug design. *Current Drug Targets-Infectious Disorders* 3, 33–40.
- Walters, W.P., Stahl, M.T., Murcko, M.A., 1998. Virtual screening — An overview. *Drug Discovery Today* 3, 160–178.
- Wang, R.X., Lai, L.H., Wang, S.M., 2002. Further development and validation of empirical scoring functions for structure based binding affinity prediction. *Journal of Computer-Aided Molecular Design* 16, 11–26.
- Wishart, D.S., 2006. Metabolomics for drug discovery, development and monitoring. *Business Briefing: Future Drug Discovery*. 1–3.
- Zhao, H., Huang, D., Caffisch, A., 2012. Discovery of tyrosine kinase inhibitors by docking into an inactive kinase conformation generated by molecular dynamics. *ChemMedChem* 7 (11), 1983–1990.

Further Reading

- Arumugasamy, K., Tripathi, S.K., Singh, P., Singh, S.K., 2016. Protein-protein interaction for the de novo design of cyclin-dependent kinase peptide inhibitors. In: Orzáez, M., Sancho Medina, M., Pérez-Payá, E. (Eds.), *Cyclin-Dependent Kinase (CDK) Inhibitors. Methods in Molecular Biology*, vol. 1336. New York, NY: Humana Press. (ISBN: 978-1-4939-2926-9).
- Baig, H.M., Ahmad, K., Roy, S., *et al.*, 2016. Computer aided drug design: Success and limitations. *Current Pharmaceutical Design* 22 (5), 572–581.
- Barril, X., 2017. Computer-aided drug design: Time to play with novel chemical matter. *Expert Opinion on Drug Discovery* 12 (10), 977–980.
- Broijmans, N., Kuntz, I.D., 2003. Molecular recognition and docking algorithms. *Annual Review of Biophysics and Biomolecular Structure* 32, 335–373.
- Bursulaya, B.D., Totrov, M., Abagyan, R., Brooks 3rd, C.L., 2003. Comparative study of several algorithms for flexible ligand docking. *Journal of Computer-Aided Molecular Design* 17 (11), 755–763.
- Friesner, R.A., Banks, J.L., Murphy, R.B., *et al.*, 2004. Glide: A new approach for rapid, accurate docking and scoring. 1 Method and assessment of docking accuracy. *Journal of Medicinal Chemistry* 47 (7), 1739–1749.
- Godwin, R.C., Melvin, R., Salsbury, F.R., 2015. Molecular dynamics simulations and computer-aided drug discovery. In: Zhang, W. (Ed.), *Computer-Aided Drug Discovery. Methods in Pharmacology and Toxicology*. New York, NY: Humana Press. ISBN: 978-1-4939-3521-5.
- Liao, C., Sitzmann, M., Pugliese, A., Nicklaus, M.C., 2011. Software and resources for computational medicinal chemistry. *Future Medicinal Chemistry* 3 (8), 1057–1085.
- Muegge, I., Bergner, A., Kriegl, J.M., 2017. Computer-aided drug design at Boehringer Ingelheim. *Journal of Computer-Aided Molecular Design* 31 (3), 275–285.
- Talele, T.T., Khedkar, S.A., Rigby, A.C., 2010. Successful applications of computer aided drug discovery: Moving drugs from concept to the clinic. *Current Topics in Medicinal Chemistry* 10 (1), 127–141.
- Wathieu, H., Issa, N.T., Stephen, W.B., Dakshanamurthy, S., 2016. Harnessing polypharmacology with computer-aided drug design and systems biology. *Current Pharmaceutical Design* 22 (21), 3097–3108.
- Wong, Y.H., Chiu, C.C., Lin, C.L., *et al.*, 2016. A new era for cancer target therapies: Applying systems biology and computer-aided drug design to cancer therapies. *Current Pharmaceutical Biotechnology* 17 (14), 1246–1267.
- Yoshitomi, F., Tadaaki, M., Kiyotaka, M., *et al.*, 2016. Miscellaneous topics in computer-aided drug design: Synthetic accessibility and GPU computing, and other topics. *Current Pharmaceutical Design* 22 (23), 3555–3568.

Relevant Websites

<http://chem.sis.nlm.nih.gov/chemidplus>
ChemIDplus.

www.ebi.ac.uk/Tools/sss/psiblast

European Bioinformatics Institute. PSI-BLAST.

www.moldiscovery.com

Molecular Discovery.

www.ncbi.nlm.nih.gov

National Center for Biotechnology Information.

<http://dtp.nci.nih.gov/webdata.html>

NCI discovery services.

<http://cactus.nci.nih.gov/chemical/structure>

NCI/CADD Chemical Identifier Resolver.

<http://195.178.207.233/PASS2008/en/index.html>

Prediction of activity spectra for substances.

www.pdb.org

Protein Data Bank.

<http://predictioncenter.org>

Protein Structure Prediction Center.

www.chemspider.com

Royal Society of Chemistry. ChemSpider.

www.schrodinger.com

Schrödinger Inc.

www.simulations-plus.com

Simulations Plus, Inc.

http://thomsonreuters.com/products_services/science/science_products/a-z/world_drug_index/

Thomson Reuters World Drug Index.

http://en.wikipedia.org/wiki/Molecular_dynamics#Major_software_for_MD_simulations

Wikipedia. MD simulation program list.

In Silico Identification of Novel Inhibitors

Chong-Yew Lee, Ezatul E Kamarulzaman, Beow Keat Yap, Sy Bing Choi, and Habibah A Wahab, Universiti Sains Malaysia, Minden, Malaysia

Maywan Hariono, Sanata Dharma University, Sleman, Indonesia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Computational docking simulation is no longer an unfamiliar step in drug discovery since the last few decades. As time and financial costs are crucial factors in drug discovery and development, it is important to constantly reform the drug discovery pipeline with novel technologies such as virtual screening or docking that can narrow down the candidates to the most promising lead compound for further testing. This *in silico* approach has been widely applied in the preliminary stage as well as to probe the insight into the molecular mechanism of the protein-ligand interaction in drug discovery life cycle. The past decade has witnessed tremendous growth in computational capabilities that enable computational docking to expedite the discovery of many potential inhibitors. Here, we review its potential application in the discovery of potential inhibitors to one of the deadly infectious diseases, namely the Dengue infection.

Background

Dengue is a growing global health concern with an estimated 390 million infections occurring each year (Bhatt *et al.*, 2013). It is caused by the Dengue virus (DENV), a flavivirus (*Flaviviridae*) that is transmitted by *Aedes aegypti* and *Aedes albopictus* mosquitoes. The infections are endemic in tropical and subtropical regions but are spreading to higher latitudes in recent times. To date, there are still no approved antivirals to stop dengue and currently, patients are only treated with supportive care to relieve fever, pain and dehydration. On the other hand, there have been a few vaccine candidates in advanced clinical pipeline. However, the development of an efficacious and safe vaccine for dengue is not without challenges as the virus has at least four serotypes. A major hurdle is the requirement for a working vaccine to be “tetraivalent” that is immunogenically effective against all the four serotypes. If protection against one or more serotypes is insufficient, the “vaccinated” person may develop a more severe disease due to the mechanism called antibody-dependent enhancement (Halstead, 2014). Recent discovery of the fifth serotype of the DENV (Vasilakis *et al.*, 2013) further complicates the vaccine development effort.

Alternatively, the development of dengue antiviral small molecule seems viable as the structures of the proteins, the parts which make up the virus particle are largely well characterised (Kuhn *et al.*, 2002; Smit *et al.*, 2011). These DENV components are divided into “structural” proteins (capsid, envelope protein and M protein) that construct the virus and seven “non-structural” proteins (NS1, NS2A, NS2B, NS3, NS4A, NS4B and NS5) that play functional and enzymatic roles (Fig. 1). Knowledge and understanding of their structural features and roles in virus infection has grown tremendously. “Structural” proteins despite their names, have been shown to play active roles in the life cycle and propagation of the DENV. For example, the DENV envelope (E) protein, has been shown to play crucial roles in the attachment and entry of the virus into the host cells, the key steps in the infection (Rey, 2003; Modis *et al.*, 2004).

It is not surprising that anti-flaviviral discovery efforts in the recent decade (2006–2016) have focused on these viral proteins as targets for new small molecule inhibitors. In these target-oriented efforts, computational or *in silico* methods have played increasingly important roles in the discovery process. The availability of high resolution DENV proteins’ crystal structures in the Protein Database (PDB) (Berman *et al.*, 2003) partly encourages this trend where docking technique, in particular, has been employed as a tool to virtually screen large libraries of compounds against the protein crystal structures. To a lesser extent, docking procedures have also been used to model the inhibitory activity of a promising lead and help rationalise its mechanism of action towards improving the ligand’s potency.

This review aims to highlight recent (2006–2016) efforts in which *in silico* docking played important roles in the discovery of new DENV inhibitors. Whenever possible, we attempt to keep the review to projects where the docking results are followed through or validated with experimental data. We discuss the outcomes of these projects and evaluate their prospects, and finally identify least explored areas or targets which may hold promise for future *in silico* discovery of potential DENV inhibitors.

Case Studies

Elucidation of Potential Inhibitors on DENV Protein Targets

DENV structural proteins

The envelope (E) protein

The E protein of DENV (DENV E) is responsible for receptor recognition, attachment of the virus on the host cell and endocytosis into the cell. It also participates in the fusion of the viral membrane with the host membrane (Harrison, 2008). This protein comprises three domains (Domains I, II and III) (Fig. 2) that have been pursued as drug targets. However, the hydrophobic pocket

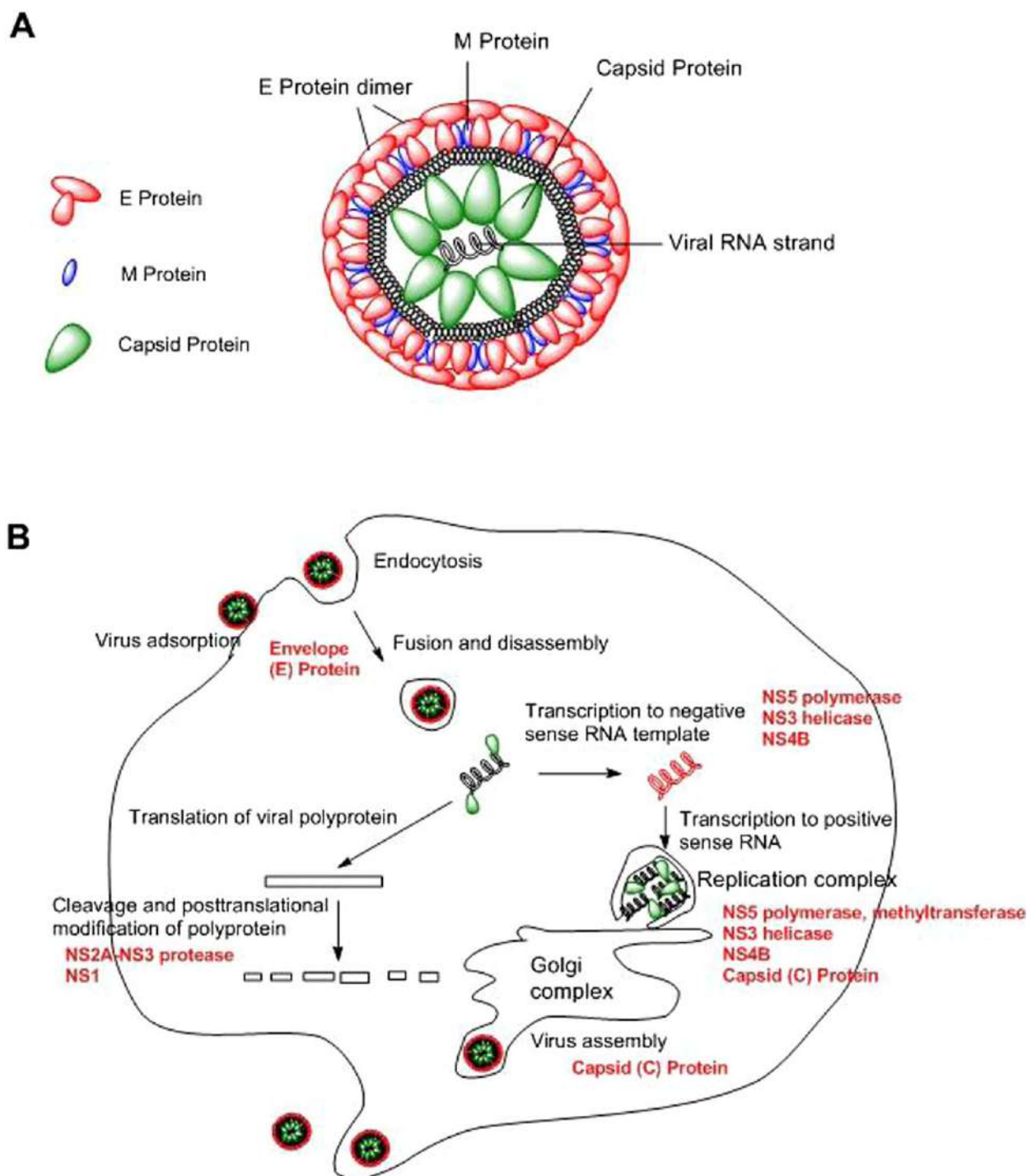


Fig. 1 Targeted and targetable dengue proteins. (A) Structure of a dengue virus particle. (B) Steps in the infection process. Reproduced from Behnam, M.A.M., Nitsche, C., Boldescu, V., Klein, C.D., 2016. The medicinal chemistry of dengue virus. *J. Med. Chem.* 59, 5622–5649.

between Domains I and II (Modis *et al.*, 2003) is the most amenable and widely used for virtual screening or docking. Also known as the n-octyl- β -D-glucoside (β -OG) pocket, this domain acts as a hinge point between the two domains and plays a role in the conformational rearrangement necessary for the fusion process.

Wang and co-workers was among the first groups to utilize the β -OG as a target for virtual screens (Wang *et al.*, 2009). Noting the conformational flexibility of the pocket from the two crystal structures published by Harrison and co-workers (Modis *et al.*, 2003, 2004), they incorporated soft van der Waals contact parameters into the docking parameters as a

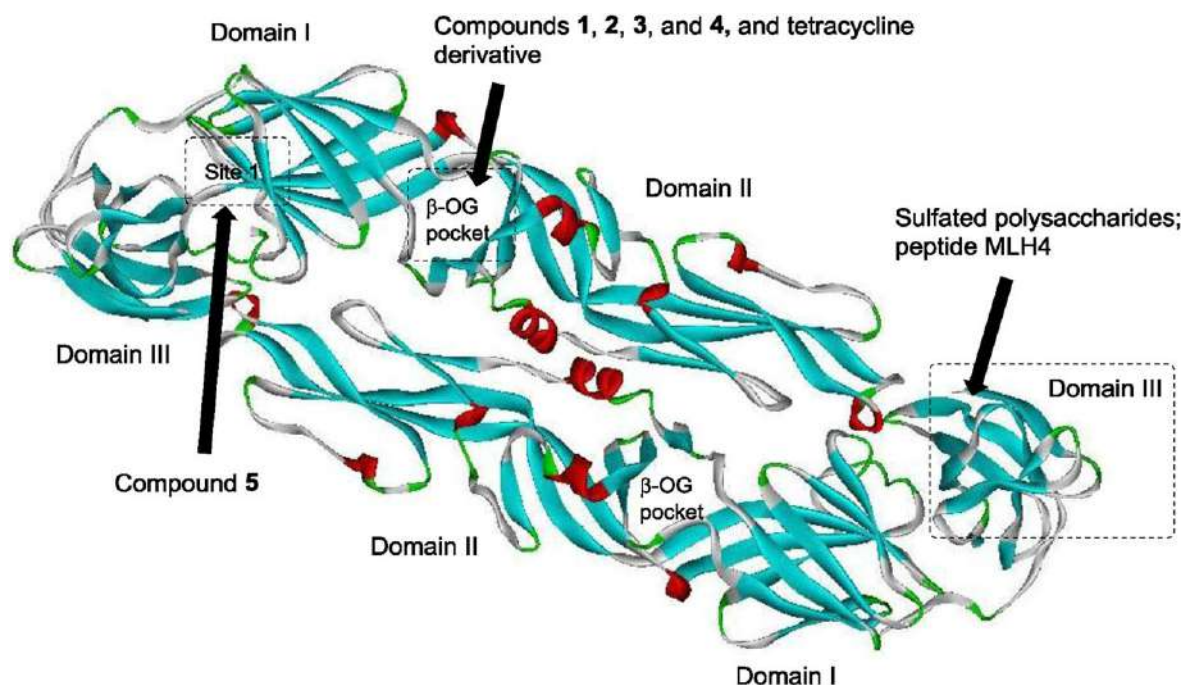


Fig. 2 Binding sites for identified DENV entry inhibitors on the crystal structure of DENV2 E protein dimer (PDB: 1OKE), shown in cartoon representation, coloured by secondary structure (helices in red and sheets in cyan).

compromise for the rigid protein requirement for docking. Compound (1) (**Fig. 3**) was the most potent compound as a result of this screen and verified by plaque reduction assay (EC_{50} = 0.068 to 0.496 μ M in the four DENV serotypes). Redocking of (1) into the β -OG pocket using GOLD software, revealed a similar pose to that of the amphiphilic β -OG, with its hydrophobic portion (chloro-phenyl-thiophene) buried within the space occupied by the octyl tail of β -OG. The polar pyridinyl-methyl-quinazolinyl-amine part of the molecule bound near the entrance of the pocket and was exposed to solvent but the phenyl group of the quinazoline ring had a hydrophobic interaction with Leu198 and possibly Pro53 or Ala205 while the amine hydrogen interacted with Glu49 or Thr48. In the same year, a multi-tiered docking screen of the Maybridge chemical database on the β -OG pocket resulted in the identification of five hit compounds which were then subjected to biological evaluation (Kampmann *et al.*, 2009). The most potent compound (2) (IC_{50} = 1.2 μ M), interestingly showed similar binding pose as that reported by Wang *et al.* suggesting a common pharmacophore for β -OG pocket binders across diverse chemical structures. Indeed, on hindsight, the discovery of two tetracycline derivatives as verified hits (IC_{50} = 55–67 μ M in a plaque reduction assay) against the β -OG pocket in an earlier *in silico* screen (Yang *et al.*, 2007) suggested that this binding site was particularly distinct. Of the ten hits initially identified, these tetracycline derivatives were found to dock at the entrance of the pocket and extended into the pocket, particularly between the D and I segments along the stretch of residues 48–52. In contrast, other inactive compounds were docked deeper inside the β -OG pocket.

A recent *in silico* screen against the β -OG pocket (Leal *et al.*, 2017) identified a quinazoline derivative (3) bearing some similarities to (1) as one of the hits. However, (3) showed a lower potency than (1) in a luciferase reporter assay of DENV (EC_{50} = 3.1 μ M). Despite adopting a novel “hybrid” design using motifs from two previous *in silico* screen hits (Li *et al.*, 2008; Zhou *et al.*, 2008), Jayaprakash and co-workers discovered (4) (EC_{50} = 1.39 μ M), the best hit which has several moieties of (2) (Jadav *et al.*, 2015). Results as these, where similarly structured hits were discovered, may be due to overlaps in the ligands available in the databases used in these virtual screening campaigns as much as the fact that the docking models were fundamentally based on the binding mode of β -OG.

Taking advantage of the shifting features of the DENV E domains during the fusion process, Yennamalli and co-workers attempted to search for alternative sites similar to the β -OG pocket on the E protein in both pre-fusion (PDB ID: 1OKE) and post-fusion (PDB: 1OK8) forms of the protein using a cavity-detection algorithm termed “putative active sites with spheres” (Yennamalli *et al.*, 2009). Two sites (named as Sites 1 and 2) were identified and were used in a large *in silico* docking screen of thirteen publicly and commercially available databases using a combination of GOLD, FlexX, and DOCK docking programs. After winnowing the hits against several criteria including drug-like features and chemical stability, seven compounds were biologically tested. The best compound (5), was able to reduce the plaque formation in infected Vero cells with an EC_{50} of 4 μ M thus validating these novel sites of DENV E for virtual screening of new and potential DENV entry inhibitors.

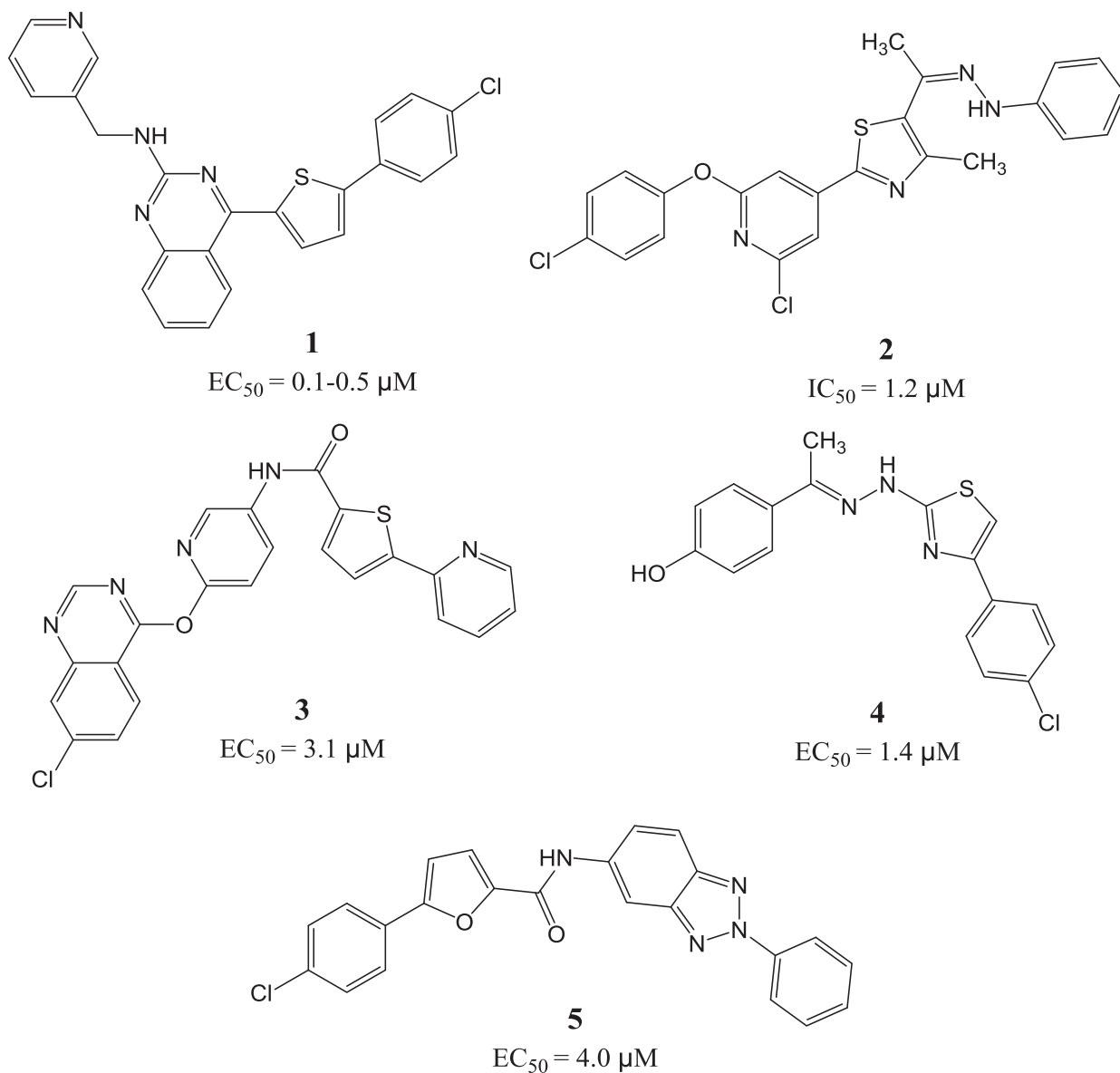


Fig. 3 DENV entry inhibitors identified from docking screens and verified experimentally.

The capsid (C) protein

The DENV Capsid (DENV C) is a 100 amino acid-length homodimeric protein with four α -helical regions and an intrinsically disordered NH terminal domain. A number of DENV C form a complex with the single-stranded viral RNA (nucleocapsid). They play a role in viral RNA recruitment during virus assembly and release of the genome during infection. Notably these RNA replication processes involve their interactions with the host intracellular lipid droplets and very low density lipoproteins (VLDL) (Byk and Gamarnik, 2016).

Research on the DENV C has just begun to shed light on its interaction mechanism with the host lipid entities (Martins *et al.*, 2012; Faustino *et al.*, 2015). Recent identification of key target ligands responsible for DENV C-lipid interactions such as apolipoprotein E and perilipin 3 (Faustino *et al.*, 2014; Carvalho *et al.*, 2012) has paved the way for the possibility of a structure-based discovery of DENV C binding inhibitors. However, there is presently no deposition of crystal structures of DENV C in the PDB except for a nuclear magnetic resonance (NMR) solution structure (PDB: 1R6R) (Ma *et al.*, 2004; Fig. 4). As such, very limited docking studies against this protein have been reported to date. The best lead for initiating such endeavours so far lies in the only small molecule DENV C ligand ST-148 which was fortuitously discovered in a high-throughput screening of a library of over 200,000 compounds (Byrd *et al.*, 2013). Docking analysis of ST-148 on the DENV C (PDB: 1R6R) suggested that the compound interacts with the protein thus enhanced its self-dimerization and this may have led to interference in the assembly and disassembly of the nucleocapsid necessary for RNA replication (Scaturro *et al.*, 2014).

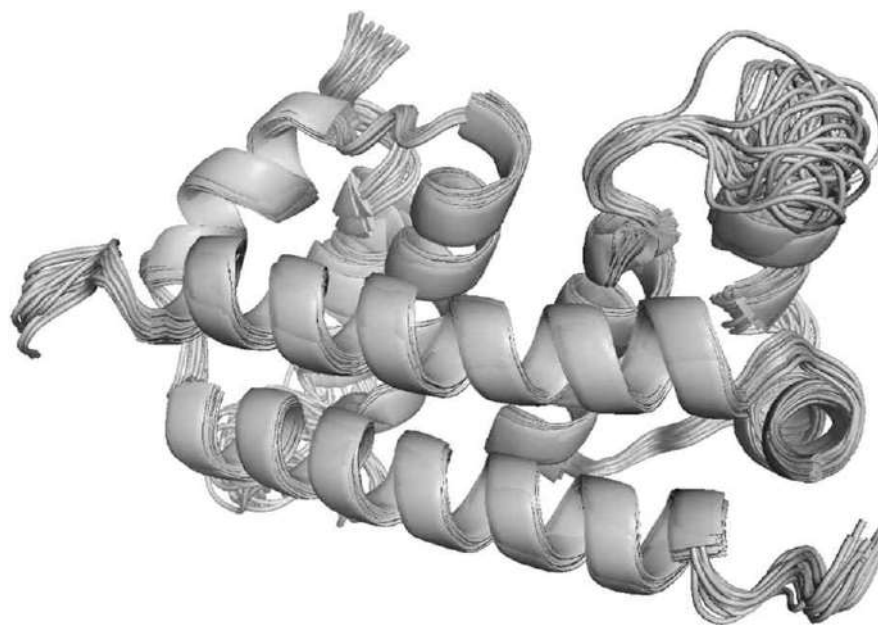


Fig. 4 NMR solution structure of DENV C Protein (PDB: 1R6R). Reproduced from Ma, L., Jones, C.T., Groesch, T.D., Kuhn, R.J., Post, C.B., 2004. Solution structure of dengue virus capsid protein reveals another fold. *Proc. Natl. Acad. Sci. USA* 101, 3414–3439. In cartoon representation.

DENV Non-Structural Proteins

NS1

NS1 is the first of seven non-structural proteins encoded for by the viral genome. It is a highly conserved 46 kDa non-structural glycoprotein of the DENV flavivirus, exists intracellularly, on cell surface and in secreted forms. NS1 plays an important role in viral particle production and viral RNA replication, presumably through post-translational N-glycosylation. NS1 contains two key N-glycosylation sites, located at Asn130 and Asn207, respectively, with Asn130 glycosylation site is more biologically important as Asn130 mutation has been shown to result in attenuation of dengue virus infection in both mammalian and mosquito cells (Tajima *et al.*, 2008).

To date, there is limited literature can be found with regards to *in silico* studies to discover novel NS1 inhibitors despite the fact that the crystal structure of DENV NS1 dimer has been solved (Fig. 5, PDB: 4O6B) (Akey *et al.*, 2014). The structure contains three domains: A small β -roll domain formed by two intertwined β -hairpins; a *Wing* domain, comprises an α/β subdomain and a discontinuous connector that sits against the β -roll and a β -ladder domain, formed by 18 antiparallel β -strands assembled in a continuous β -sheet that runs along the whole length of the dimer. Using this crystal structure, Qamar *et al.* (2014) attempted to identify potential inhibitors of Asn130 glycosylation site of NS1 by docking 2200 flavonoids from 405 medicinal plants using MOE docking program. Out of the top 100 flavonoids screened, six flavonoids showed significant interactions with Asn130, suggesting that flavonoids have potential to be DENV inhibitors, although further experimental evidence is required to support this observation.

Apart from the post-translational N-glycosylation, recent work found that NS1 also acts as an important modulator of cellular energy metabolism (Allonso *et al.*, 2015). Results from co-immunoprecipitation, cross-linking and ELISA assays have proposed that NS1 interacts with glyceraldehyde 3-phosphate dehydrogenase (GADPH), which results in the increase of glycolytic flux and ultimately the energy production, which is required for proper viral replication. *In silico* docking studies between the homology model of GADPH (PDB template: 1ZNQ) and the crystal structure of NS1 (PDB: 4O6B) using the web-based ClusPro server (Version 2.0; Kozakov *et al.*, 2017) revealed that the hydrophobic protrusion of NS1, located at the β -roll domain (Cys4, Val5, Leu13, Lys14, Cys15 of both monomers), interacts with the hydrophobic residues of GADPH located opposite its catalytic site. Interestingly, the NS1-GADPH binding interface is at a distant location from the N-glycosylation sites (Fig. 5), suggesting its potential as a novel target for the design and discovery of new drug candidate against the DENV viral replication.

NS2B–NS3 protease

NS3 is one of the most widely studied DENV proteins. It is a 69 kDa multifunctional protein which consists of 618 amino acids that carry two functional domains, the N-terminal trypsin-like serine protease domain and the C-terminus domain that contains the activities of a RNA helicase and a RNA-stimulated NTPase. The NS3 serine protease domain comprises 180 amino acid residues at its N-terminal with a catalytic triad formed by residues His51, Asp75 and Ser135 (Brinkworth *et al.*, 1999; Lescar *et al.*, 2008). The proteolytic activity of DENV NS3 protease (NS3pro) requires the NS2B protein, which is located in the polypeptide precursor

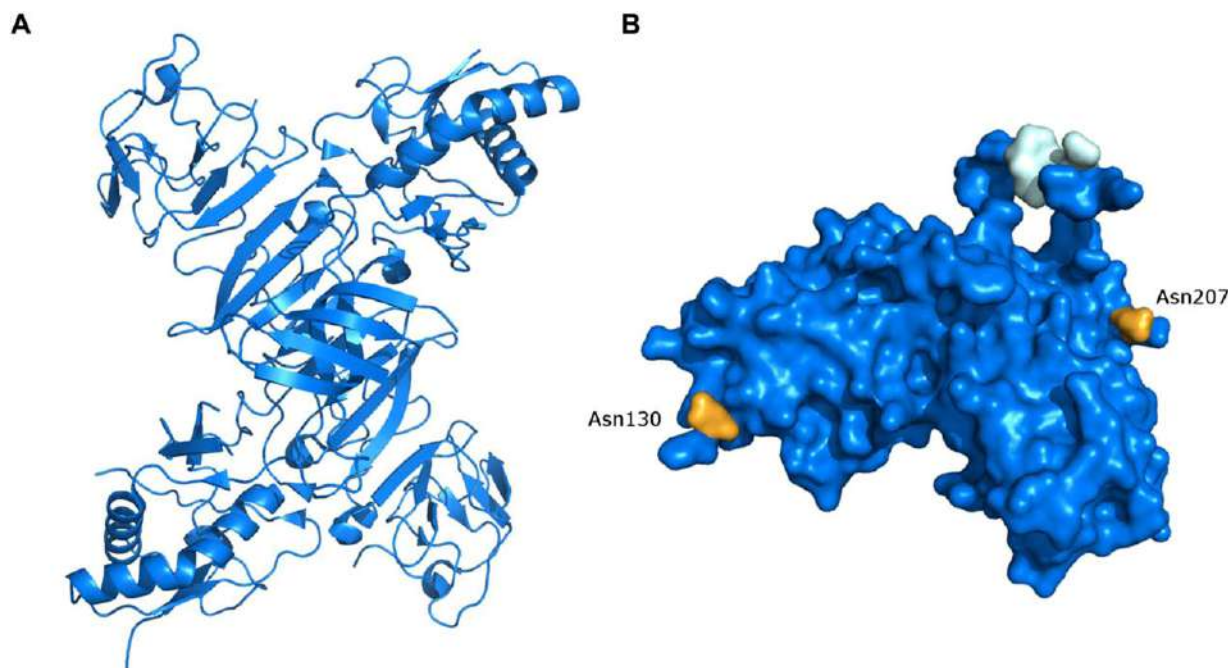


Fig. 5 (A) The crystal structure of DENV NS1 protein (PDB: 406B), in cartoon representation. (B) N-glycosylation sites (gold) and the GADPH binding site (cyan) of the DENV NS1 protein surface (blue). The separation between the N-glycosylation and GADPH binding sites suggest two possible targets for NS1 inhibitors as potential antiviral candidates.

upstream of the NS3pro domain, as a cofactor (Aleshin *et al.*, 2007). This structural activation probably resulted in conformational rearrangements of the catalytic triad and the subsite (S^5 - $S^{5'}$) residues, which facilitates substrate binding and/or optimizes proton exchange during catalysis (Barbato *et al.*, 1999; Kim *et al.*, 1996). NS2B-NS3pro is responsible for the cleavage of DENV C and for cleavage at the NS2A/NS2B, NS2B/NS3, NS3/NS4A, NS4A/NS4B and NS4B/NS5 boundaries (Chambers *et al.*, 1993; Arias *et al.*, 1993; Yotmanee *et al.*, 2015). Therefore, blocking or inhibiting the activities of DENV NS2B-NS3pro has become an attractive target in anti-dengue drug discovery.

To date, there are thirteen 3D structures of NS3pro from the four dengue serotypes deposited in the Protein Data Bank (PDB). In general, the crystal structures of NS3pro in complex with the cofactor NS2B for the four serotypes demonstrated that the NS3pro domain adopts chymotrypsin-like structure consisting two β -barrels of which each barrel contains six β -strands. The β -barrel conformation typically is compact with short or absent loop structures. The active site is located in between the cleft of the two β -barrels. It is relatively shallow, contourless and composed of a catalytic triad containing His51, Asp75 and Ser135 (Yildiz *et al.*, 2013; Noble *et al.*, 2012; Luo *et al.*, 2008, 2010; Erbel *et al.*, 2006; Chandramouli *et al.*, 2010).

The initial structure-activity relationship of NS2B-NS3pro can be inferred from well-established cleavage sites of DENV polyprotein by NS2B-NS3pro. The cleavage sites preferable by this enzyme usually occur after two basic amino acid residues (Lys-Arg, Arg-Arg or Arg-Lys) or occasionally after Gln-Arg at the P1 and P2 positions (Fig. 6) and are usually followed by a short side-chain amino acids, most commonly Gly, Ser or Ala at P1' (Chambers *et al.*, 1990). A functional substrate profiling of the P1-P5 and P1'-P5' regions of the proteases from the four DENV serotypes showed that the enzyme shared very similar substrate specificities across the serotypes (Khumthong *et al.*, 2002; Chanprapaph *et al.*, 2005) with recent finding shows that the interaction between these substrates with the NS2B/NS3 protease is in the order of DENV C > intNS3 > NS2A/NS2B > NS4B/NS5 > NS3/NS4A > NS2B/NS3 (Yotmanee *et al.*, 2015) which is in agreement with experimental values.

Two tetrapeptides, Bz-Nle-Lys-Arg-Arg and Bz-Nle-Lys-Thr-Arg have been shown to have high affinities for NS2B-NS3pro (K_i =12.42 and 33.9 μ M, respectively) (Yin *et al.*, 2006). A stepwise removal of the octapeptide down to tetrapeptide i.e., WYCW-NH₂ resulted in protease inhibition in low micromolar activities (K_i =4.2, 4.8, 24.4 and 11.2 μ M for DENV1-4, respectively) (Prusis *et al.*, 2013). The docked pose of the tetrapeptide on DENV3 is shown in Fig. 7. Tripeptide with a varied N-terminal cap groups and P2 Lys' side chain modification have also been shown to have micromolar activities against DENV2 NS2B-NS3pro. It was suggested that P2Arg play the specificity shift towards the protease while an extended P2 Lys failed to block the protease activity, suggesting a size-constrained S2 pocket (Schüller *et al.*, 2011).

Over the past ten years, there have been many efforts to search for small molecules anti-dengue through the mean of *in silico* approaches that target NS2B/NS3-protease. Many compounds either small molecules from natural products or synthetic chemicals have been discovered as potential inhibitors for the enzyme. Two cyclohexyl chalcone derivatives, 4-hydroxypanduratin A (Fig. 8: 6) and panduratin A (Fig. 8: 7) isolated from *Boesenbergia rotunda* (L.) Mansf. Kulturpfl. have shown good competitive inhibitory activity with K_i values of 21, and 25 μ M, respectively (Tan *et al.*, 2006). The discovery has prompted the docking studies

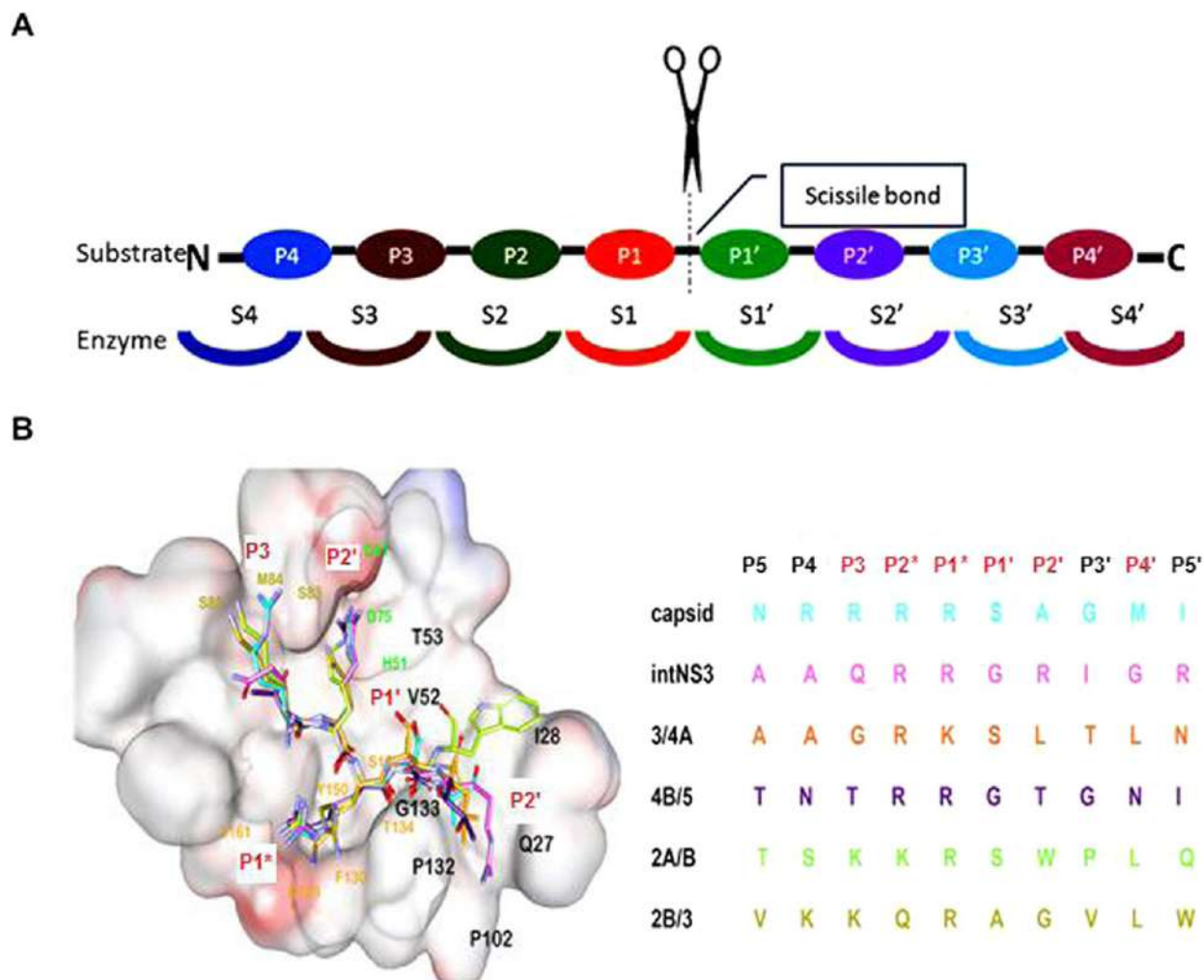


Fig. 6 (A) Schechter and Berger's Nomenclature of the peptide substrate. The substrate is cleaved between positions P1-P1'. Reproduced from Schechter, I., Berger, A., 1967. On the size of the active site in proteases. I. Papain. *Biochem. Biophys. Res. Commun.* 27, 157–162. (B) Models of DENV polypeptide substrates, Capsid, intNS3, NS3/NS4A, NS4B/NS5, NS2A/NS2B, NS2B/NS3 (stick model) superposed on the surface of DENV2 NS2B/NS3pro and their sequences from P5 – P5'. Reproduced from Yotmanee, P., Rungrotmongkol, T., Wichapong, K., *et al.*, 2015. Binding specificity of polypeptide substrates in NS2B/NS3pro serine protease of dengue virus type 2: A molecular dynamics Study. *J. Mol. Gr. Modell.* 60, 24–33.

of these compounds which led to the synthesis of ethyl 3-(4-(hydroxymethyl)-2-methoxy-5-nitrophenoxy)propanoate (Fig. 8: 8) which also have strong inhibition ($K_i=59\ \mu\text{M}$) (Kee *et al.*, 2007). Docking noncompetitive inhibitors (Fig. 8: 9–14), which were previously isolated (Tan *et al.*, 2006) using AutoDock3 (Morris *et al.*, 1998) showed that these noncompetitive ligands bound to the allosteric sites, particularly at the backbone carbonyl of Lys74 (which is bonded to Asp75, one of the catalytic triad residues) of DENV2 NS2B-NS3pro, the results of which, are in accordance with the experimental study (Chandramouli *et al.*, 2010). Based on these results, three series of panduratin A derivatives (i.e., of 246DA (15), 2446DA (16) and 20H46DA (17)) were designed by adding various substituents at the positions 1–5 of the benzyl ring A for both 4-hydroxypanduratin A and panduratin A. A docking calculation of these substituted benzyl compounds with NS2B/NS3pro showed favourable binding energies which are close to the computed energy of the parent compounds (Frimayanti *et al.*, 2011).

Noncompetitive inhibition of NS2B-NS3pro especially that involved binding at the allosteric sites continues to generate interests in many researches. Previously, the 3F10 antibody was found to disrupt the interaction of NS2B with NS3 *in vitro*, which suggest the possibility of NS2B-NS3pro allosteric inhibition (Phong *et al.*, 2011). Docking analysis of the noncompetitive inhibitors screened from 13,341 small compounds, with the backbone structures of chalcone, flavanone, and flavone, available in the ZINC database, showed four compounds (Fig. 9: 18–21) (the most potent, compound (Fig. 9: 18) with $K_i=69.9\ \mu\text{M}$) formed hydrogen bonding interactions with the amino acid residues Glu88, Gly124, and Asn167, as well as π -cation interactions with Lys74. This site is located behind the active site (Ser135, Asp175, and His51) (Heh *et al.*, 2013). Allosteric region near Ala125 has also been proposed as a target for the binding of noncompetitive NS2B-NS3pro inhibitors. Ala125 sits between the 120s and 150s loops, which have been shown to be very flexible (Yildiz *et al.*, 2013). Other allosteric sites including that formed by the amino

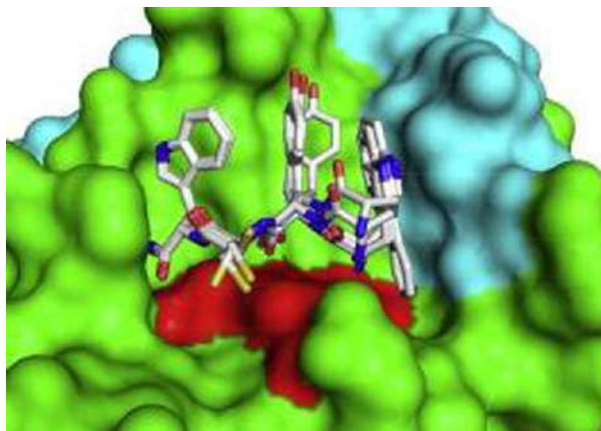


Fig. 7 The docked poses of the tetrapeptide WYCW-NH₂ at the DENV3 NS2B-NS3pro binding site are presented in grey stick form. The protein is presented in surface form with NS2B coloured by blue, NS3 in green and the catalytic triad are spotted in red surface form. taken with permission from Prusis, P., Junaid, M., Petrovska, R., *et al.*, 2013. Design and evaluation of substrate-based octapeptide and non substrate-based tetrapeptide inhibitors of dengue virus NS2B-NS3 proteases. *Biochem. Biophys. Res. Commun.* 434, 767–772.

acids Trp89, Thr120, Gly121, Glu122, Ile123, Gly124, Gly164, Ile165, Ala166, and Gln167 of NS3 and Lys73, Lys74, Asn152, Val78, Gly82, and Met84 of NS2B has been identified and used in the design of eight potent NS2B-NS3pro inhibitors (22–29) (IC_{50} = 1–98 μ M, **Table 1**) (Wu *et al.*, 2015).

Virtual screening of large chemical databases such as Maybridge, ChemBridge and ZINC databases have aided the discovery of many potent compounds as well as scaffolds for further optimisation. 300,000 compounds were docked using AutoDock 3 (Morris *et al.*, 1998) on the GVSS platform and led to the identification of seven novel inhibitors (IC_{50} ~ 4.0–90 μ M) with three of them as competitive inhibitors (K_i ~ 3.0–5.0 μ M). The inhibitors were stabilised by forming H-bonds with the catalytic residues (His51 and Asp75) and through hydrophobic interactions with various amino acids in the NS3pro active site pocket (Nguyet *et al.*, 2013). In another virtual screening campaign, compound (30) was found to be active against NS2B-NS3pro (IC_{50} = 13.12 ± 1.03 μ M) and was used to design fourteen derivatives which in turn lead to the discovery of four new inhibitors with biological activity. Based on the results, small molecule-based scaffold hopping was performed to design new compounds which led to the discovery of 17 novel NS2B-NS3pro inhibitors with IC_{50} values of 7.46 ± 1.15 to 48.59 ± 3.46 μ M (Deng *et al.*, 2012).

Fragment-based virtual screening approach has also been applied to discover small molecule inhibitors of DENV NS2B-NS3pro (Knehans *et al.*, 2011). Since suitable co-crystal structure of DENV-2 protease with bound inhibitor was unavailable at that time, the homology model of DENV-2 NS2B-NS3pro (derived from various flavivirus NS2B-NS3pro structures) was used to screen 150,000 molecular fragments collected from ZINC database using AutoDock Vina (Knehans *et al.*, 2011; Trott and Olson, 2010) focusing on the interactions with the S1 and S2 pockets of NS2B-NS3pro. High scored fragments were then linked to generate new bioactive molecule where docking analysis showed that this compound interacted with S1 pocket (Asp129) *via* hydrogen bonding and electrostatic interaction. 23 compounds were tested experimentally that led to the identification of two inhibitors, compounds (31) and (32) (IC_{50} = 7.7 μ M and 37.9 μ M, respectively).

There are many other docking studies as well as virtual screenings using docking which have been performed and able to give insight into the binding of potential NS2B-NS3pro. However, they are not validated experimentally, making it difficult to assess the successful of their *in silico* prediction. Such examples include Qamar *et al.* (2014) who had carried out a docking analysis with DENV NS2B-NS3 protease using a library of 940 phytochemicals. In their study, they found *Garcinia* phytochemicals to be their best hits, including gossypol, mangostenone C, garcidepsidone A, and dimethyl- calabaxanthone bound deeply inside the active site of DENV NS2B/NS3 protease among all tested phytochemicals and had interactions with catalytic triad (His51, Asp75, Ser135). In another study, a molecular docking analysis (using Molegro Molecular Docker (Thomsen and Christensen, 2006)) of 2194 plant-derived secondary metabolites with dengue virus protein targets carried out by Powers and Setzer (2016), showed that 24 compounds docked strongly to dengue virus NS2B-NS3 protease ($E_{\text{dock}} < -130$ kJ/mol) (Powers and Setzer, 2016). The lowest-energy docking poses of the chalcone ligands balsacone A, balsacone B, and balsacone C, bound into both the binding site of Lys-Arg as well as the catalytic site occupied by the co-crystallized inhibitor M9P. Although all these findings have not been validated experimentally, they did provide some insights into the development of DENV NS2B-NS3pro inhibitors.

NS3 helicase

Besides having the protease activity, the C-terminal fragment of NS3 behaves as an ATP-dependent RNA helicase and unwinds the double-stranded RNA genome during virus replication (Utama *et al.*, 2000). The helicase activity (Gorbalenya *et al.*, 1989; Li *et al.*, 1999) is vital in virus replication as it separates dsRNA during viral replication. The crystal structure of DENV-2 helicase catalytic domain revealed a three-lobed flattened topology with a large number of loops. It has a distinct structural feature with a long tunnel runs across the centre of one face of the protein (Xu *et al.*, 2005; Behnam *et al.*, 2016; Swarbrick *et al.*, 2017).

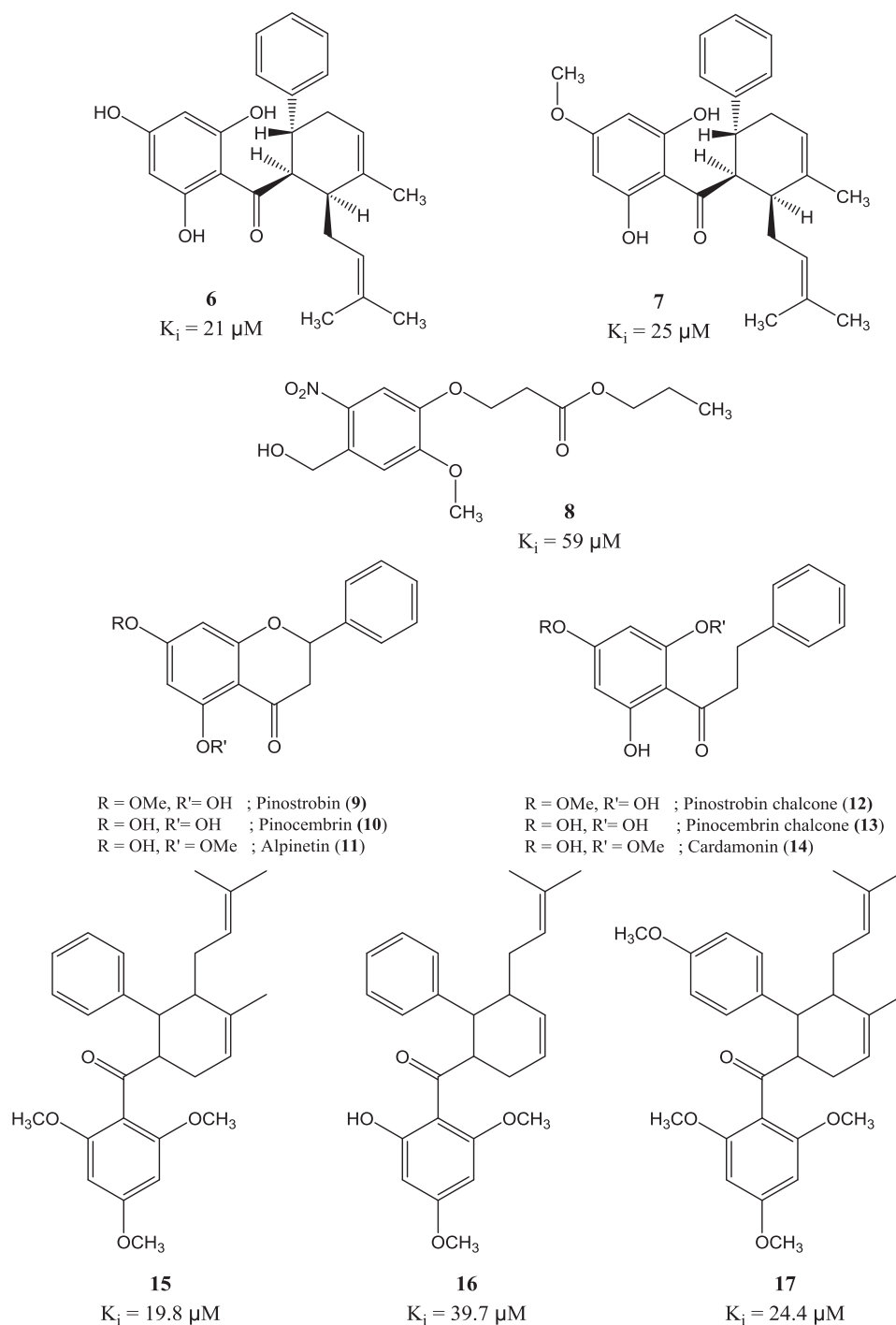


Fig. 8 Structures of the competitive (4-hydroxypanduratin A (**6**), panduratin A (**7**)) and noncompetitive inhibitors (**9–14**) isolated from *Boesenbergia rotunda* (L.) Mansf. Kulturpfl. (Ethyl 3-(4-(hydroxymethyl)-2-methoxy-5-nitrophenoxy) propanoate (**8**), and panduratin A derivatives (**15–17**) are synthetic competitive inhibitors).

Therefore, due to its role in virus replication, NS3 helicase has also been considered as a target with high potential to develop dengue antiviral agents. There is limited information available on molecular docking involving this target. Recently, Halim and co-worker highlighted both structure and ligand-based approach in search of potential NS3 inhibitor from ZINC database (Halim *et al.*, 2017). A pharmacophore model generated using 16 compounds from literatures (Basavannacharya and Vasudevan, 2014; Sweeney *et al.*, 2015; Mastrangelo *et al.*, 2012) was used to filter 1,201,474 compounds from the ZINC database. Molecular docking using FRED docking program was then conducted on those compounds which fit the pharmacophore model. 25

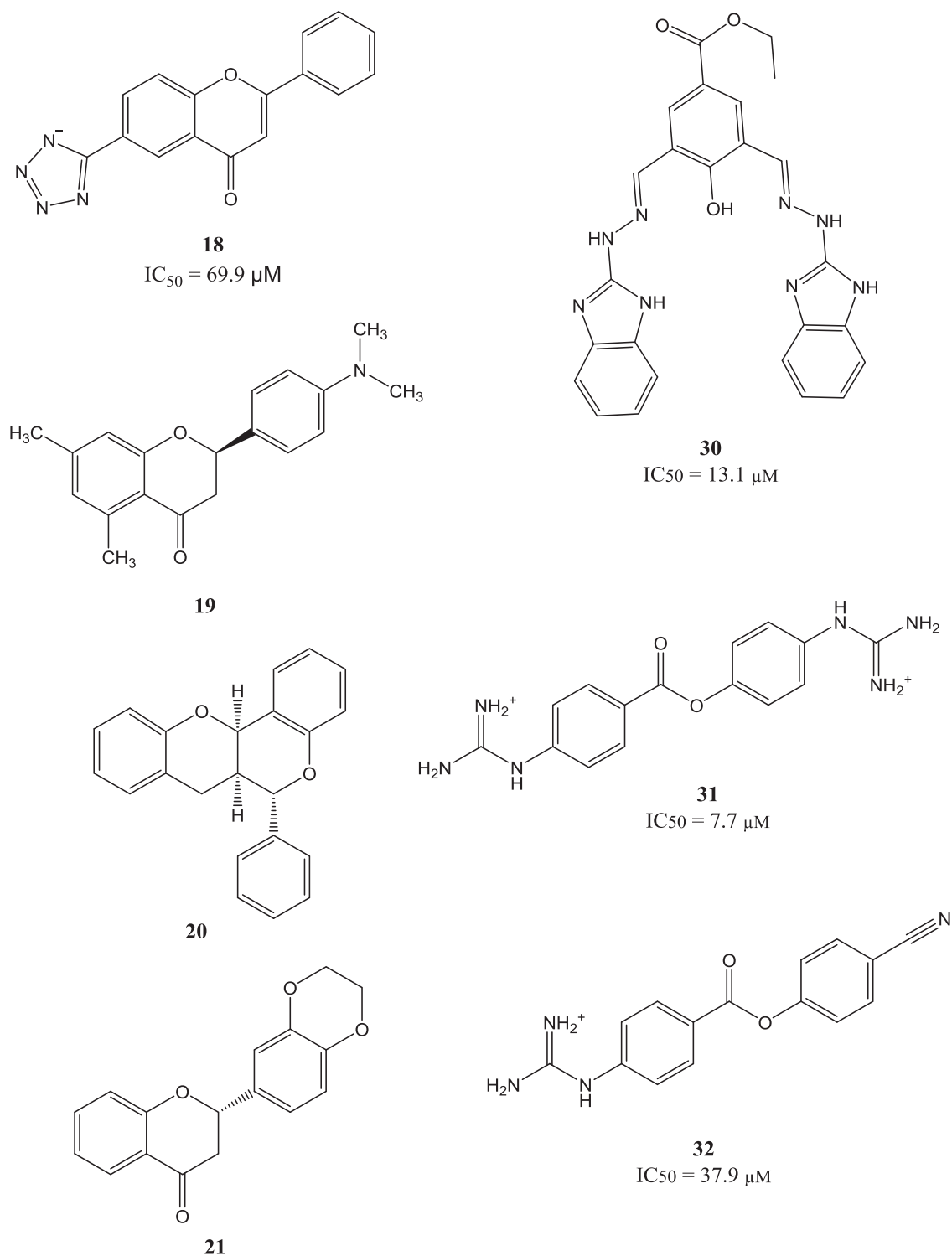
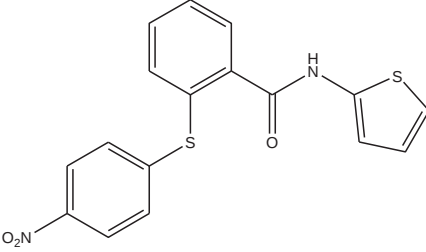
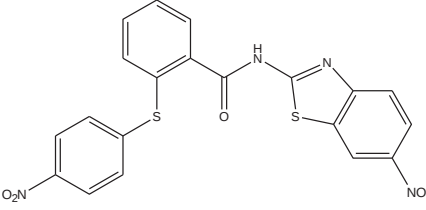
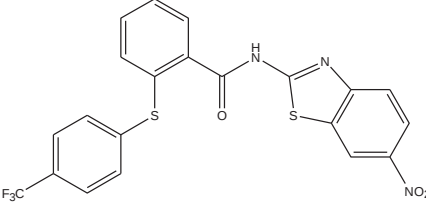
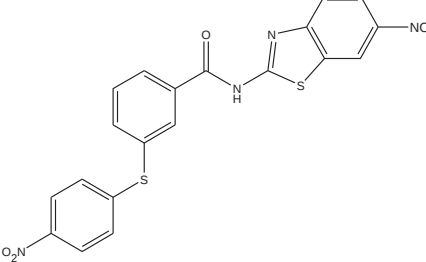
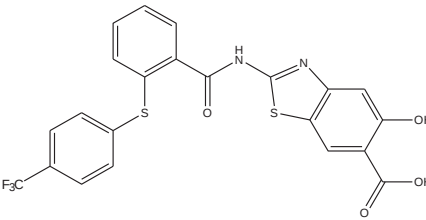
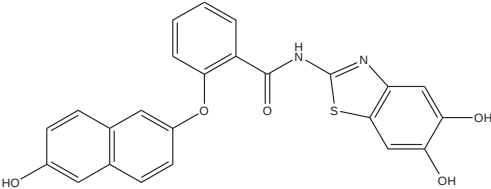


Fig. 9 Inhibitors of NS2B-NS3pro identified from small molecule-based virtual screening (18–21: noncompetitive inhibitors; (30): competitive inhibitor) and fragment-based virtual screening (31–32).

compounds with drug-like properties were selected and postulated as potent inhibitors for NS3 helicase. This postulation however, is not validated experimentally.

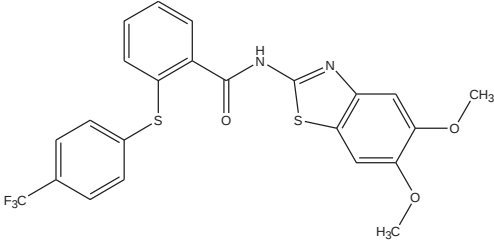
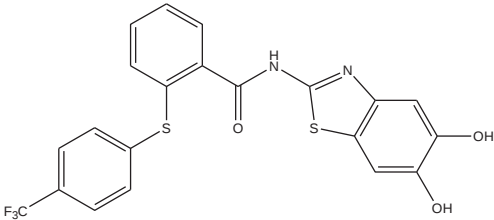
Since the crystal structure of this enzyme is only recently available, Milani and co-worker had initiated an investigation of DENV NS3 helicase using a built 3D model based on other flavivirus (Milani *et al.*, 2009). *In silico* screening was conducted against virtual Library of Pharmacologically Active Compounds (LOPAC) which included 1280 commercially available compounds from

Table 1 Structures and activities of potent NS2B/NS3pro inhibitors

Compound	Chemical structure	Result for compound ^a :				
		DENV-2 (IC ₅₀ [μM]) ^b	DENV-3 (IC ₅₀ [μM]) ^b	Toxicity (IC ₅₀ [μM]) ^c	EC ₅₀ [μM] ^d	IC ₅₀ [μM] ^e
22		98 ± 4	31.8 ± 4.5	30	3.5 ± 0.3	15.6 ± 3.4
23		34 ± 5	31.8 ± 4.5	< 1	ND ^f	ND
24		22 ± 1	21 ± 4	1	0.1 ± 0.0	0.2 ± 0.0
25		26 ± 1	ND	3	0.3 ± 0.1	0.7 ± 0.1
26		66 ± 3	12.3 ± 2.2	10	0.9 ± 0.1	2.3 ± 0.7
27		4.2 ± 0.44	0.99 ± 0.1	10	0.8 ± 0.2	3.2 ± 1.2

(Continued)

Table 1 Continued

Compound	Chemical structure	Result for compound ^a :				
		DENV-2 (IC ₅₀ [μM]) ^b	DENV-3 (IC ₅₀ [μM]) ^b	Toxicity (IC ₅₀ [μM]) ^c	EC ₅₀ [μM] ^d	IC ₅₀ [μM] ^e
28		10% inhibition at 50 μM	NI ^g	30	2.5 ± 0.1	9.3 ± 2.5
29		3.6 ± 0.11	9.1 ± 1.02	3	>3	>3

^aValues are indicated as means ± standard deviations from 3 independent experiments performed in triplicate.

^bInhibition of isolated proteases determined by fluorometric enzyme assays with an AMC-derived substrate.

^cConcentration at which no influence on cellular metabolism was observed.

^dAntiviral activity, inhibition of viral replication.

^eBiochemical inhibition of PR in cell culture.

^fND, not done.

^gNI, no inhibition at 50 μM.

Note: Adapted from Wu, H., Bock, S., Snitko, M., *et al.*, 2015. Novel dengue virus NS2B/NS3 protease inhibitors. *Antimicrob. Agents Chemother.* 59, 1100–1109.

Sigma Aldrich (see “Relevant Website section”) using Autodock 4. Three best compounds were selected namely paromomycin sulphate, ouabain and ivermectin displaying ΔG values between −11.5 and −9.5 kcal/mol. These compounds were found to interact with Arg409 and Tyr410. These compounds were then further tested experimentally using numerous *in vitro* approach from enzymatic to cell-based assay to rule out all non-specific inhibition of NS3 helicase. They proved that ivermectin behaves like uncompetitive helicase inhibitor which only binds to protein/RNA and blocks the enzymatic activity. However, the results from cell-based assay is not able to exclude the possibility that ivermectin exerts its antiviral activity on DENV *via* other mechanism (Mastrangelo *et al.*, 2012).

NS4

NS4A and NS4B are integral membrane proteins which play multiple roles in viral replication and virus-host interaction (Zou *et al.*, 2015). To date, only one *in silico*-based drug discovery campaign on NS4B was performed. Although Paul and co-workers proposed that three of the seven docked bioactive phytochemical compounds on the NS4B protein models of DENV1-4, i.e., (-)-catechin, epicatechin and DL-catechin, have the potential to be developed into effective anti-dengue drugs on DENV-1,2 and 4, however, it is unclear how the authors came to this conclusion as the predicted binding energies of these compounds were modest (< −6 kcal/mol) and they were not validated experimentally (Paul *et al.*, 2016).

NS5

NS5 is the largest and most conserved protein in the dengue genome. It contains two functional domains, i.e., the RNA methyltransferase (MTase) domain at its N-terminus and the RNA-dependent RNA-polymerase (RdRp) domain at its C-terminus. MTase is highly important for mRNA stability, protein synthesis and viral replication as it is involved in the last two steps of mRNA capping process, i.e., transferring a methyl group from S-adenosyl-L-methionine (SAM) onto the N7 position of guanine-N7 methyltransferase (cap guanine) and to the 2′O position of the first nucleotide following the cap guanine (nucleoside-2′-methyltransferase), generating S-adenosyl-L-homocysteine (SAH) as the by-product. On the other hand, RdRp domain is involved in the generation of both minus and plus RNA strands, thus its role in viral RNA replication. Recently, Ahmad Jamil and co-workers have proposed based on their protein-protein docking results with HADDOCK that NS5 may also interact with the host protein SIAH2 which would then binds to STAT2 protein and results in the ubiquitination of the host proteins involved in IFN-mediated antiviral effect (Aslam *et al.*, 2015). This, however, remains to be confirmed by wet-lab experiments.

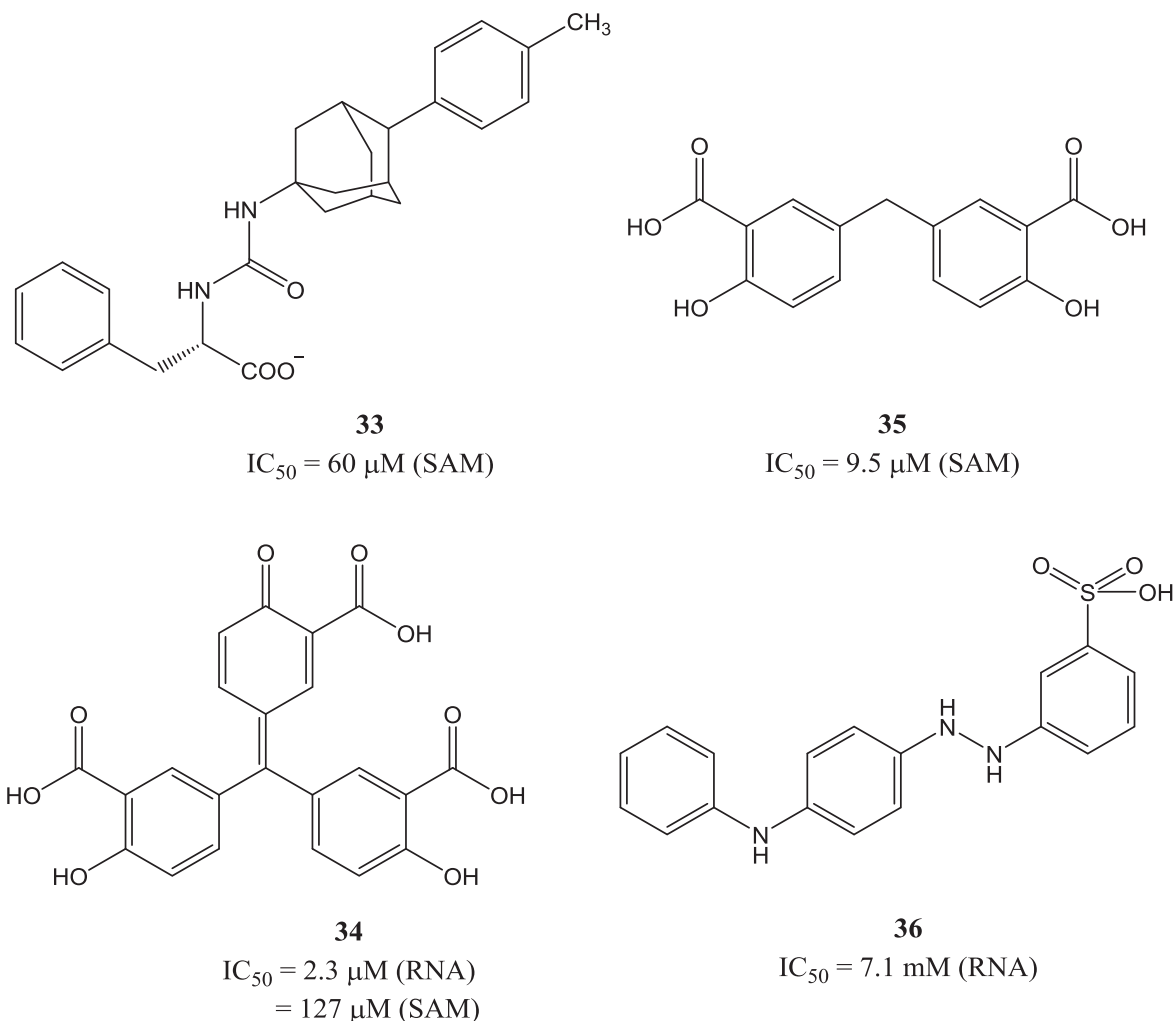


Fig. 10 Chemical structures of the inhibitors of NS5 (MTase domain) with their corresponding IC_{50} values from *in vitro* bioassays.

In the past decade, various research groups have attempted to find inhibitors of MTase, targeting either the S-adenosylmethionine (AdoMet/SAM) or the RNA cap site (GTP site), using a combination of computational and experimental techniques. [Luzhkov *et al.* \(2007\)](#) performed a high-throughput based virtual screening from 2.1 million commercially available compounds using 2D similarity searching, pharmacophore filtering based on the structure of SAH, and docked onto the SAM site of the NS5MTaseDV (dengue virus mRNA cap (nucleoside-2'O)-methyltransferase) with GOLD docking software ([Luzhkov *et al.*, 2007](#)). The 15 top-ranked compounds were then tested with bioassay test on recombinant MTase in competition with the RNA substrate. A novel inhibitor scaffold, (**33**) with IC_{50} value of $60 \mu M$ was identified from this approach ([Fig. 10](#)). Interestingly, investigation into the docked complex between (**33**) and the SAM site of MTase revealed that (**33**) did not form important polar interactions with Asp79, Trp87, Gly106, Glu111 and Asp131 at the binding site as SAH. Rather, the efficient binding of (**33**) is due to its good complementary fit to the target site. Similarly, [Milani *et al.*](#) successfully identified a non-specific inhibitor, ATA, (**34**) with $IC_{50} = 2.3 \mu M$ (RNA cap site); $127 \mu M$ (SAM site) from *in vitro* assay of the top three ranked *in silico* docked compounds from 1280 commercially available compounds in the LOPAC library with AutoDock4. Another successful example is by [Podvinec *et al.* \(2009\)](#) who performed a multistage docking of more than 5 million commercially available compounds against the two binding sites of MTase with Glide, followed by experimental verification of 263 best compounds with MTase enzyme inhibition assay ([Podvinec *et al.*, 2010](#)). Four inhibitors with IC_{50} of $< 10 \mu M$ in *in vitro* assay were successfully identified from this work, although only 2 of the four compounds were found later to be genuine, non-aggregating, low micromolar inhibitors (compounds (**35**) and (**36**), [Fig. 10](#)).

Recently, [Benmansour *et al.* \(2016\)](#) attempted to use *in silico* docking to guide fragment linking of 3 of the 7 fragment hits bound close to SAM binding site (IC_{50} values range between $180 \mu M - 9 mM$) identified from a 500 fragments library *via* thermal shift assay, fragment-based x-ray crystallography screening, and enzymatic assays ([Benmansour *et al.*, 2017](#)). Analogues with different linkers at different positions of the two fragment hits were designed using the Link Multiple Fragment Tools of MOE and docked with and without 3D-pharmacophore constraint using MOE Docking tool and Glide, respectively. As only amide and urea-

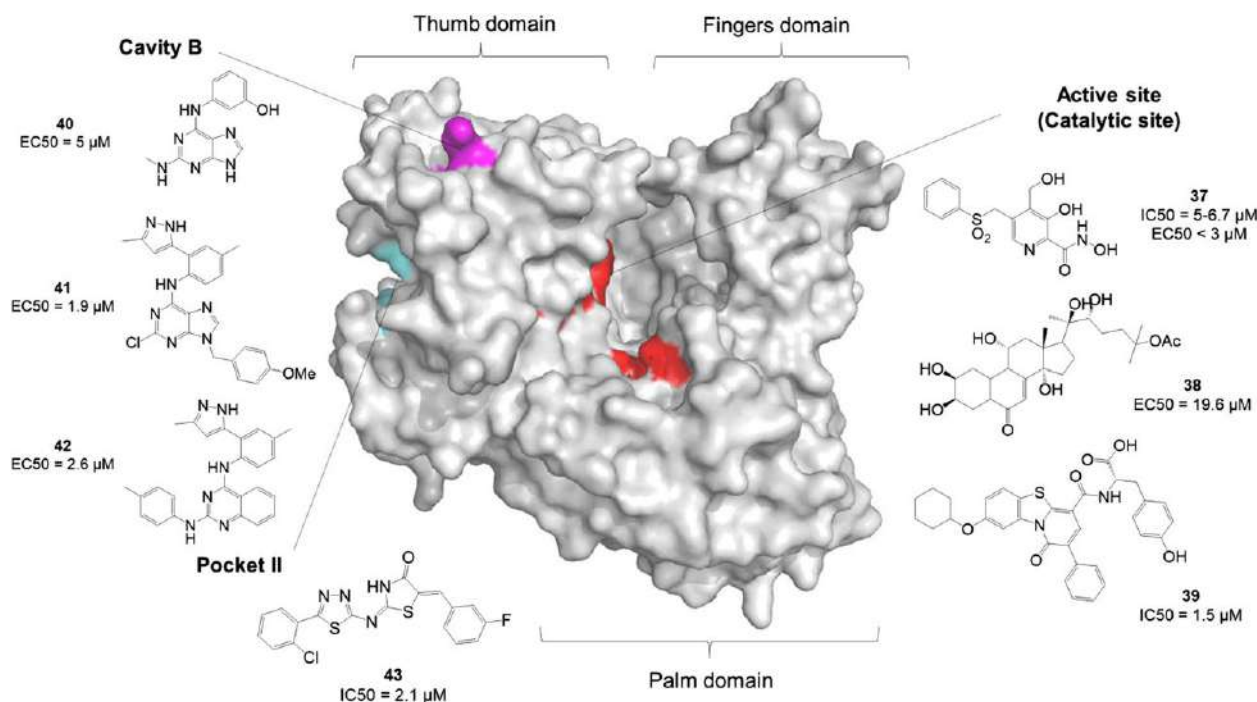


Fig. 11 Binding sites on RdRp domain of NS5 (PDB: 2J7U). Reproduced from Yap, T. L., Xu, T., Chen, Y. L., *et al.*, 2007. Crystal structure of the dengue virus RNA-dependent RNA polymerase catalytic domain at 1.85-angstrom resolution. *J. Virol.* 81, 4753–4765. Surface as potential drug targets with the active/catalytic site (red) and allosteric site (cavity B, magenta; pocket II, cyan). Chemical structures of the inhibitors at each site are shown with their corresponding IC_{50} values from *in vitro* bioassays.

based linker were able to maintain the correct orientation of the initial fragments, various analogues with these linkers were synthesized and tested. Ultimately, this led to a novel series of non-nucleoside inhibitors (N-phenyl-[(phenylcarbamoyl) amino] benzene-1-sulfonamide and phenyl [(phenylcarbamoyl) amino]benzene-1-sulfonate derivatives) with a 10–100-fold stronger inhibition of 2'-OMTase activity compared to the initial fragments (IC_{50} =91–452 μ M). Their activities in the virus inhibition assay, however, remain uninspiring, presumably due to intrinsic low activity *ex vivo* or their poor ability to cross membrane. Other potential inhibitors of MTase that have been reported include cyclic peptides CTWYC and [Tyr123] human Prepro Endothelin (110 – 130) (SAM site), cyclic peptides CYEFC and human urotensin II (RNA-cap site) (Tambunan *et al.*, 2014; Idrus *et al.*, 2012), phytochemical compounds (isoborreverine, diplacone, styracifolin B, neosilyhermin A, and β -[4-hydroxy-5-(5-hydroxytigloyloxy)tigloyl]-santamarin), SAH analogues (e.g., SAH-M331, SAH-M2696, SAH-M1356) (Tambunan *et al.*, 2017; Singh *et al.*, 2016) and ribavirin analogues (Sivakumar and Sivaraman, 2011). These ligands, however, were identified solely by *in silico* docking and molecular dynamics simulations, and thus further validation by *in vitro* assays is warranted.

While MTase domain remains the mainstream target for the inhibition of NS5 protein for the past decade, RdRp domain is becoming more popular in recent years. According to Yap *et al.* (2007), RdRp domain can be divided into three key subdomains, i.e., fingers, thumb and palm domains, stabilised by the NLS region, with loops L1, L2 and L3, linker α 21- α 22 and the priming loop encircling the catalytic active site (GTP site) and shaping the template tunnel (Fig. 11).

To date, most inhibitors of the RdRp domain of NS5 focus on either the active catalytic site or the allosteric sites in the thumbs domain. To illustrate, a highly specific, pyridoxine-derived small molecule inhibitor, (37) (DMB220) (IC_{50} =5–6.7 μ M in enzymatic assay, EC_{50} < 3 μ M in antiviral cell-based assay) was designed as an active-site metal ion chelator to chelate divalent metal ions in the catalytic site of DENV RdRp. Docking studies with AutoDock Vina suggest that it interacts with active site residue Asp533, Asp663 and Asp664, with its sulfoxide oxygen and hydroxyl groups on the pyrimidine ring coordinates with both Mg ions at the active site (Xu *et al.*, 2015). Similarly, Cheng *et al.* (2016) reported a new and potent, highly selective antiviral compound, (38) (zoanthone A, EC_{50} =19.6 μ M, CC_{50}/EC_{50} =36.7), isolated from ethanolic extract of plant *Zoanthus spp* (Cheng *et al.*, 2016). (38) formed three hydrogen bonds between C-6=O and charged NH residue of Arg729, OH-2 and backbone NH of Trp795, as well as OH-20 and NH at imidazole side chain of His798 of the NS5 RdRp tunnel, suggesting that the carbonyl group at C-6 and hydroxyl group at C-2 in ecdysone skeleton is essential for their bioactivities, in agreement with the SAR results of the ecdysone analogues performed. On the other hand, Milani and co-workers found that the top two (out of the 203) docked compounds on the protein active site of DENV-3 RdRp (Tarantino *et al.*, 2016) (PDB 2J7U) (Yap *et al.*, 2007) with AutoDock 4.2, the pyridobenzothiazole compound, HeE1-2Tyr, (39), was found to be able to inhibit Dengue RdRps' activity (IC_{50} DENV3 RdRp= 1.5 μ M) and showed antiviral activity in cell culture against all four DENV serotypes (IC_{50} 6.8–15 μ M). Other potential inhibitors of RdRp domain targeting the active site include plant-derived secondary metabolites (i.e., dimethylisoborreverine, drummondin

D, flinderole B, and pungioid A), quercetin derivatives (i.e., quercetin 3-(6''-(E)-p-coumaroylsophoroside)-7-rhamnoside) and fragment-derived inhibitor (i.e., [2-(4-carbamoylpiperidin-1-yl)-2-oxoethyl]8-(1,3-benzothiazol-2-yl)naphthalene-1-carboxylate) (Yap *et al.*, 2007; Powers and Setzer, 2016; Anusuya *et al.*, 2016; Anusuya and Gromiha, 2016).

Apart from targeting the active site, various groups have also come up with inhibitors targeting the allosteric sites of NS5 RdRp. For example, Vincetti *et al.* (2015) performed virtual screening on the cavity B of DENV-2 NS5 RdRp (Leu328, Lys330, Trp859, Ile863) from a library of about 3000 known tyrosine protein kinase, Src active scaffolds (ChEMBL, BindingDB databases) and 10,000 virtual synthetically accessible compounds designed in house using the SmiLib v2.0 with both Glide and AutoDock Vina and found that purine derivatives were able to inhibit DEN-2 replication in virus cell-based assay (EC₅₀: 5–20 μ M) while (40) also inhibit NS3-NS5 interaction (33% inhibition at 50 μ M concentration), presumably due to the hydrophobic contacts and cation- π interactions between purine and the side chain of Lys330. Similar observation was made by Venkatesham *et al.* (2017) for their purine lead compound, (41) and quinazoline lead compound, (42) with EC₅₀ of 1.9 μ M and 2.6 μ M, respectively, although the reported binding modes varied slightly from Vincetti's, presumably due to difference in type of docking programme used. In a separate study, the most potent 4-thiazolidinone analogue, (43) was found to bind to pocket-II of the thumb domain *via* hydrophobic interactions with Ala776, Met809, Cys780, Ser785, Trp833, Thr880 and Tyr882, hydrogen bond interaction with backbone of Pro884, and electrostatic interaction with amide side chain of Asn777 of DENV-2 NS5 model, as predicted by molecular docking and confirmed by the fluorescence quenching assay (Manvar *et al.*, 2016). On the other hand, 39 non-nucleotide compounds, consisting of natural compounds and NITD1/2/29 analogues, were predicted *in silico* to bind strongly to a binding site at the RNA template tunnel (e.g., Arg737, Thr413, Met343 in DENV-3) (Galiano *et al.*, 2016).

Discussion and Future Direction

Progress in structural biology has successfully elucidated the structures of various proteins of dengue which certainly has contributed to vast efforts in computer-aided drug discovery of many potential anti-viral dengue inhibitors in the last ten years. Many studies have focused on the DENV E protein due to the fact that multiple of this protein cover most of the surface area of the viral particle. Furthermore, the knowledge of its conformational changes as well as well-defined crystal structures in the fusion event have created several binding site targets for potential DENV entry inhibitors such as β -OG pocket (the hydrophobic pocket), stem region and the domain III (Modis *et al.*, 2003, 2004).

DENV E protein has a key role in viral entry to the cells. It provides attachment through receptor interaction and subsequent endosomal fusion. Inhibiting the entry of the dengue virus to the cell thereby avoiding the infection is an attractive way to develop potent and specific dengue antivirals. These molecules would exert their effects without having to enter the cells, thus not subjected to strict structural and chemical constraints and may potentially limits the immune system hyperactivation that could lead to severe dengue (De La Guardia and Leonart, 2014).

Perhaps a different approach to target the DENV E is through preventing it from establishing contacts with other proteins of the virus or the host cell for viral adsorption (Perera-Lecoin *et al.*, 2013). The discovery of some large molecular weight polysaccharides such as curdlan sulphate (Ichiyama *et al.*, 2013) and 3-O-sulfated GlcA (Hidari *et al.*, 2012) that were able to inhibit virus entry pointed to this possibility. Docking simulations with the active compounds predicted their binding to sites in which DENV E interacts with the host receptor such as the glycosaminoglycan receptors. A similar approach using a peptide that mimics the ectodomain of the DENV M protein (Panya *et al.*, 2014) also suggested the ability of the peptide inhibitor MLH40 to disrupt protein-protein interaction as predicted by their docking experiment. To date, no small molecules (m.w. < 500), were reported as hit compounds targeting these protein-protein "contact" sites of the E protein and thus, there is an immense opportunity to venture on finding small molecules targeting these receptor binding sites on DENV E.

Most of the entry inhibitors are peptidic compounds. Similarly, this class of compounds has also been considered as DENV protease inhibitors. Although the peptidic or peptidomimetic compounds are able prevent fusion to take place or inhibit DENV protease in low micromolar concentration, they do not possess drug-like structure. Thus, they will be likely to encounter physicochemical stability as well as pharmacokinetic problems during further stage of drug development. It is thus not surprising that, currently, no peptide inhibitor has been clinically used as a dengue antiviral agent. However, the knowledge of how these peptide inhibitors bind to DENV drug targets can be applied to derive a model with certain pharmacophoric features which in turn can be used in designing small molecule inhibitors (Gao *et al.*, 2010; Steuer *et al.*, 2011).

Other proteins such as C, NS2B-NS3 protease, NS3helicase, NS4 and NS5 are also feasible targets to look into in the screening for antiviral inhibitors. However, lack of crystal structure has hampered the discovery of potential binders to the capsid (C). High resolution crystal structures with well-defined binding sites are very much preferable in conducting molecular docking studies, the fact of which is true to the case of DENV NS2B-NS3 protease which has no doubt been the most favourite target for discovery of DENV antiviral agent as reviewed above.

NS2B-NS3pro is a chymotrypsin-like protease that has a catalytic triad formed by His51-Asp75-Ser135, which is essential for its substrate cleavage activity. In addition to the catalytic triad, Gly151 and Gly153 are also important residues as they participate in the formation of the oxyanion hole (along with Ser135) to stabilize the tetrahedral intermediate. This active site of the protease is flat and it would require a substantial conformational change in NS2B to enable the inhibitor to bind, thus, represent a major obstacle for the inhibitor to bind (Aguilera-Pesantes *et al.*, 2017). However, this limitation has not stalled the progress in searching for potential inhibitors as many researchers have also considered the inhibitors that target allosteric binding sites. In fact, a review

by Leonel *et al.* (2018) showed that about 3,384,268 molecules including nitro, catechol, halogenated and quarternary compounds have been evaluated for the NS2B-NS3pro inhibition activity. However, none of them has entered clinical studies presumably that their toxicity, pharmacokinetic and pharmacodynamic (*absorption, distribution, metabolism, and excretion* and toxicity: ADMET) properties have yet to be characterised.

Most of the earlier researchers attempted to discover peptidic inhibitors targeting the active site, based on their understanding of cleavage activities of DENV polyprotein by NS2B-NS3pro. The cleavage sites preferable by this enzyme usually occur after two basic amino acid residues (Lys-Arg, Arg-Arg or Arg-Lys) or occasionally after Gln-Arg at the P1 and P2 positions. However, the host serine protease such as furin also recognises the two basic amino acid residues at P1 and P2 positions, thus, the selectivity and toxicity of the potential inhibitors might represent significant issues during the development stage. This limitation thus must be taken into account during the development of DENV NS2-NS3pro inhibitors. Considering other targets for DENV virus, which do not have any resemblance with the host proteins such as that displayed by DENV NS5 RNA-dependent RNA polymerase (RdRp) would be a good alternative as inferred from previous knowledge of antiviral agents that are now in clinical trials. This enzyme plays a critical role in viral replication by forming part of a multimeric complex. As there is no equivalent to flavivirus RdRp in the human host cells, DENV NS5 inhibitors are most likely to have little or no toxicity issues.

However, this fact is not true to the DENV NS5 inhibitors under current development. Although a lot of researches are ongoing in the search for inhibitors of NS5, in particular the RdRp protein, none of these potential drug leads has actually reaches clinical trial. To illustrate, the nucleoside analogue, NITD008, which showed significant reduction in DENV 1–4 replication in *in vitro* studies, appeared to be highly toxic in mice (Yin *et al.*, 2009). Similarly, various non-nucleoside inhibitors were also found to be either toxic in animal model or having poor inhibition potency (El Sahili and Lescar, 2017). The present drug for Hepatitis C virus (HCV), e.g., balapiravir, which are supposed to bind to RdRp protein of dengue (as both dengue and HCV shares similar architecture of RdRp), is found to be ineffective against the dengue RdRp protein (Nguyen *et al.*, 2013). These suggest that the search of novel inhibitors of RdRp that are potent and less toxic is highly essential. As this protein is highly druggable, with many potential druggable sites, including the active site and various allosteric sites, and with multiple crystal structures available, RdRp NS5 still definitely is an attractive target for *in silico*-guided design and discovery of novel dengue antiviral agents.

The review has highlighted many case studies which yield many potent compounds that could be developed further as dengue antiviral agents. However, it is worth noted that docking alone might not be sufficient to create molecules with high potential as dengue antiviral agent. One should elucidate further the pharmacophoric features of the active compounds or thorough mechanism of action by molecular dynamics simulation or a hybrid quantum/molecular mechanics (QM/MM) method before proceeding to lead optimisation. Multi-step *in silico* approach is valuable for optimisation of hit compounds discovered by virtual screening or molecular docking. Pharmacophore modelling, 3D quantitative structure-activity relationship (QSAR), fragment and scaffold hopping could be used further to optimise the hits reported in the literature for inhibitors with better activity. As demonstrated by Deng *et al.* (2012), small molecule-based scaffold hopping was able to optimise compound 30 (Fig. 9) identified from virtual screening using docking (verified active against NS2B-NS3pro with $IC_{50} = 13.12 \pm 1.03 \mu M$) towards to the discovery of 17 novel NS2B-NS3pro inhibitors with IC_{50} values of 7.46 ± 1.15 to $48.59 \pm 3.46 \mu M$. In addition, the other future direction that should be considered is the evaluation of ADMET of these already discovered potential inhibitors. *In silico* approaches for ADMET have reached maturity and can be utilised, albeit, must be eventually validated by *in vitro* or *in vivo* experiments.

Taking novel compounds either new peptidic or small molecules from research to clinic is a daunting task and take a long time and resources. Drug repurposing has been considered by many researchers as it can significantly reduce the cost and development time as these off-patent, already marketed drugs, have demonstrated *safety* in humans and thus negates the need for Phase I clinical trials (Oprea *et al.*, 2011). Many researchers have conducted virtual screening as the starting point for drug repurposing for dengue inhibitors. Using *in silico* docking to explore the single-strand RNA access site within WNV helicase, ivermectin, a broad spectrum antiparasitic drug, was identified as potential inhibitor of NS3 helicase (Milani *et al.*, 2009). Ivermectin displayed uncompetitive inhibition of flaviviral helicase unwinding activity in biochemical assays in the upper nanomolar range for WNV and DENV but did not affect the ATPase activity of the NS₃ helicase domain. In cell culture, ivermectin caused virus titer reduction with an EC_{50} of $0.7 \mu M$ for DENV and $4 \mu M$ for WNV (Mastrangelo *et al.*, 2012). Clinical trial involving ivermectin sponsored by Mahidol University and The Thailand government (clinical trial identifier: NCT02045069) is underway although the results on the clinical safety and efficacy have yet to be published as far as we are aware. Ivermectin is a natural product isolated from bacteria. The structure is rather big (m.w. = 875.10 g/mol), therefore might pose problem during absorption. Structure optimisation using either structure-based or ligand-based design could be further carried out to yield analogues with more desirable properties. Moreover, ivermectin is commercially used as a topical preparation for parasite, and thus its administration by oral route deserves to be studied more carefully. Ivermectin is just one example of potential drug for anti-dengue discovered *via* drug repurposing. Perhaps, more rigorous *in silico* drug repurposing would be able to discover other already marketed or off-patent drugs as clinical dengue antiviral agents.

Concluding Remark

Over the last decade, significant number of researches had been reported using computational docking approach in dengue drug discovery. As discussed earlier, either at the preliminary stage *via* high-throughput screening to confine the search from huge number of compound databases or to further predict the potential interaction from the selected hit, molecular docking no doubt

aids in identifying potential hits. Nevertheless, it is also important to note that the integration of other computational approaches such dynamics study, QM/MM and QSAR will serve to enhance the chances of the elucidation of credible potential hits.

Acknowledgement

Authors acknowledge grant no 1001/PKIMIA/855006 that fund the work related to dengue in Universiti Sains Malaysia.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Comparative Epigenomics. Computational Protein Engineering Approaches for Effective Design of New Molecules. Natural Language Processing Approaches in Bioinformatics. Structure-Based Design of Peptide Inhibitors for Protein Arginine Deiminase Type IV (PAD4). Structure-Based Drug Design Workflow. Study of The Variability of The Native Protein Structure

References

- Aguilera-Pesantes, D., Robayo, L.E., Mendez, P.E., *et al.*, 2017. Discovering key residues of dengue virus NS2b-NS3-protease: New binding sites for antiviral inhibitors design. *Biochem. Biophys. Res. Commun.* 492, 631–642.
- Akey, D.L., Brown, W.C., Dutta, S., *et al.*, 2014. Flavivirus NS1 structures reveal surfaces for associations with membranes and the immune system. *Science* 343, 881–885.
- Aleshin, A., Shiryayev, S., Strongin, A., Liddington, R., 2007. Structural evidence for regulation and specificity of flaviviral proteases and evolution of the Flaviviridae fold. *Protein Sci.* 16, 795–806.
- Allonso, D., Andrade, I.S., Conde, J.N., *et al.*, 2015. Dengue virus NS1 protein modulates cellular energy metabolism by increasing glyceraldehyde-3-phosphate dehydrogenase activity. *J. Virol.* 89, 11871–11883.
- Anusuya, S., Gromiha, M.M., 2016. Quercetin derivatives as non-nucleoside inhibitors for dengue polymerase: Molecular docking, molecular dynamics simulation, and binding free energy calculation. *J. Biomol. Struct. Dyn.* 1–15.
- Anusuya, S., Velmurugan, D., Gromiha, M.M., 2016. Identification of dengue viral RNA-dependent RNA polymerase inhibitor using computational fragment-based approaches and molecular dynamics study. *J. Biomol. Struct. Dyn.* 34, 1512–1532.
- Arias, C., Preugschat, F., Strauss, J., 1993. Dengue 2 virus NS2B and NS3 form a stable complex that can cleave NS3 within the helicase domain. *Virology* 193, 888–899.
- Aslam, B., Ahmad, J., Ali, A., *et al.*, 2015. Structural modeling and analysis of dengue-mediated inhibition of interferon signaling pathway. *Genet. Mol. Res.* 14, 4215–4237.
- Barbato, G., Cicero, D.O., Nardi, M.C., *et al.*, 1999. The solution structure of the N-terminal proteinase domain of the hepatitis C virus (HCV) NS3 protein provides new insights into its activation and catalytic mechanism. *J. Mol. Biol.* 289, 371–384.
- Basavannacharya, C., Vasudevan, S.G., 2014. Suramin inhibits helicase activity of NS3 protein of dengue virus in a fluorescence-based high throughput assay format. *Biochem. Biophys. Res. Commun.* 453, 539–544.
- Behnam, M.A.M., Nitsche, C., Boldescu, V., Klein, C.D., 2016. The medicinal chemistry of dengue virus. *J. Med. Chem.* 59, 5622–5649.
- Benmansour, F., Trist, I., Coutard, B., *et al.*, 2017. Discovery of novel dengue virus NS5 methyltransferase non-nucleoside inhibitors by fragment-based drug design. *Eur. J. Med. Chem.* 125, 865–880.
- Berman, H., Henrick, K., Nakamura, H., 2003. Announcing the worldwide Protein Data Bank. *Nat. Struct. Biol.* 10, 980.
- Bhatt, S., Gething, P.W., Brady, O.J., *et al.*, 2013. The global distribution and burden of dengue. *Nature* 496, 504–507.
- Brinkworth, R.I., Fairlie, D.P., Leung, D., Young, P.R., 1999. Homology model of the dengue 2 virus NS3 protease: Putative interactions with both substrate and NS2B cofactor. *J. Gen. Virol.* 80 (Pt 5), 1167–1177.
- Byk, L.A., Gamarnik, A.V., 2016. Properties and functions of the dengue virus capsid protein. *Annu. Rev. Virol.* 3, 263–281.
- Byrd, C.M., Dai, D., Grosenbach, D.W., *et al.*, 2013. A novel inhibitor of dengue virus replication that targets the capsid protein. *Antimicrob. Agents Chemother.* 57, 15–25.
- Carvalho, F.A., Carneiro, F.A., Martins, I.C., *et al.*, 2012. Dengue virus capsid protein binding to hepatic lipid droplets (LD) is potassium ion dependent and is mediated by LD surface proteins. *J. Virol.* 86, 2096–2108.
- Chambers, T., Nestorowicz, A., Amberg, S., Rice, C., 1993. Mutagenesis of the yellow fever virus NS2B protein: Effects on proteolytic processing, NS2B-NS3 complex formation, and viral replication. *J. Virol.* 67, 6797–6807.
- Chambers, T.J., Weir, R.C., Grakoui, A., *et al.*, 1990. Evidence that the N-terminal domain of nonstructural protein NS3 from yellow fever virus is a serine protease responsible for site-specific cleavages in the viral polyprotein. *Proc. Natl. Acad. Sci. USA* 87, 8898–8902.
- Chandramouli, S., Joseph, J.S., Daudenarde, S., *et al.*, 2010. Serotype-specific structural differences in the protease-cofactor complexes of the dengue virus family. *J. Virol.* 84, 3059–3067.
- Chanprapaph, S., Saparakorn, P., Sangma, C., *et al.*, 2005. Competitive inhibition of the dengue virus NS3 serine protease by synthetic peptides representing polyprotein cleavage sites. *Biochem. Biophys. Res. Commun.* 330, 1237–1246.
- Cheng, Y.B., Lee, J.C., Lo, I.W., *et al.*, 2016. Ecdysones from *Zoanthus* spp. with inhibitory activity against dengue virus 2. *Bioorg. Med. Chem. Lett.* 26, 2344–2348.
- Deng, J., Li, N., Liu, H., *et al.*, 2012. Discovery of novel small molecule inhibitors of dengue viral NS2B-NS3 protease using virtual screening and scaffold hopping. *J. Med. Chem.* 55, 6278–6293.
- Erbel, P., Schiering, N., D'Arcy, A., *et al.*, 2006. Structural basis for the activation of flaviviral NS3 proteases from dengue and West Nile virus. *Nat. Struct. Mol. Biol.* 13, 372–373.
- Faustino, A.F., Carvalho, F.A., Martins, I.C., *et al.*, 2014. Dengue virus capsid protein interacts specifically with very low-density lipoproteins. *Nanomedicine* 10, 247–255.
- Faustino, A.F., Guerra, G.M., Huber, R.G., *et al.*, 2015. Understanding dengue virus capsid protein disordered N-Terminus and pep14-23-based inhibition. *ACS Chem. Biol.* 10, 517–526.
- Frimayanti, N., Chee, C.F., Zain, S.M., Rahman, N.A., 2011. Design of new competitive dengue NS2B/NS3 protease inhibitors—a computational approach. *Int. J. Mol. Sci.* 12, 1089–1100.
- Galiano, V., Garcia-Valtanen, P., Micol, V., Encinar, J.A., 2016. Looking for inhibitors of the dengue virus NS5 RNA-dependent RNA-polymerase using a molecular docking approach. *Drug Des. Dev. Ther.* 10, 3163–3181.
- Gao, Y., Cui, T., Lam, Y., 2010. Synthesis and disulfide bond connectivity – Activity studies of a kalata B1-inspired cyclopeptide against dengue NS2B–NS3 protease. *Bioorg. Med. Chem.* 18, 1331–1336.
- Gorbalenya, A.E., Donchenko, A.P., Koonin, E.V., Blinov, V.M., 1989. N-terminal domains of putative helicases of flavi- and pestiviruses may be serine proteases. *Nucleic Acids Res.* 17, 3889–3897.

- De La Guardia, C., Leonart, R., 2014. Progress in the identification of dengue virus entry/fusion inhibitors. *BioMed. Res. Int.* 2014, 13.
- Halim, S.A., Khan, S., Khan, A., *et al.*, 2017. Targeting dengue virus NS-3 helicase by ligand based pharmacophore modeling and structure based virtual screening. *Front. Chem.* 5, 88.
- Halstead, S.B., 2014. Dengue antibody-dependent enhancement: Knowns and unknowns. *Microbiol. Spectr.* 2.
- Harrison, S.C., 2008. Viral membrane fusion. *Nat. Struct. Mol. Biol.* 15, 690–698.
- Heh, C.H., Othman, R., Buckle, M.J., *et al.*, 2013. Rational discovery of dengue type 2 non-competitive inhibitors. *Chem. Biol. Drug Des.* 82, 1–11.
- Hidari, K.I., Ikeda, K., Watanabe, I., *et al.*, 2012. 3-O-sulfated glucuronide derivative as a potential anti-dengue virus agent. *Biochem Biophys Res Commun* 424, 573–578.
- Ichihama, K., Gopala Reddy, S.B., Zhang, L.F., *et al.*, 2013. Sulfated polysaccharide, curdlan sulfate, efficiently prevents entry/fusion and restricts antibody-dependent enhancement of dengue virus infection in vitro: A possible candidate for clinical application. *PLOS Negl. Trop. Dis.* 7, e2188.
- Idrus, S., Tambunan, U.S., Zubaidi, A.A., 2012. Designing cyclopentapeptide inhibitor as potential antiviral drug for dengue virus ns5 methyltransferase. *Bioinformation* 8, 348–352.
- Jadav, S.S., Kaptein, S., Timiri, A., *et al.*, 2015. Design, synthesis, optimization and antiviral activity of a class of hybrid dengue virus E protein inhibitors. *Bioorg. Med. Chem. Lett.* 25, 1747–1752.
- Kampmann, T., Yennamalli, R., Campbell, P., *et al.*, 2009. In silico screening of small molecule libraries using the dengue virus envelope E protein has identified compounds with antiviral activity against multiple flaviviruses. *Antivir. Res.* 84, 234–241.
- Kee, L.Y., Kiat, T.S., Wahab, H.A., Yusof, R., Rahman, N.A., 2007. Nonsubstrate based inhibitors of Dengue virus serine protease: A molecular docking approach to study binding interactions between protease and inhibitors. *Asia Pac. J. Mol. Biol. Biotechnol.* 15, 53–59.
- Khumthong, R., Angsuthanasombat, C., Panyim, S., Katzenmeier, G., 2002. In vitro determination of dengue virus type 2 NS2B-NS3 protease activity with fluorescent peptide substrates. *J. Biochem. Mol. Biol.* 35, 206–212.
- Kim, J.L., Morgenstern, K.A., Lin, C., *et al.*, 1996. Crystal structure of the hepatitis C virus NS3 protease domain complexed with a synthetic NS4A cofactor peptide. *Cell* 87, 343–355.
- Knehans, T., Schuller, A., Doan, D.N., *et al.*, 2011. Structure-guided fragment-based in silico drug design of dengue protease inhibitors. *J. Comput. Aided Mol. Des.* 25, 263–274.
- Kozakov, D., Hall, D.R., Xia, B., *et al.*, 2017. The ClusPro web server for protein-protein docking. *Nat. Protoc.* 12 (2), 255–278.
- Kuhn, R.J., Zhang, W., Rossmann, M.G., *et al.*, 2002. Structure of dengue virus: Implications for flavivirus organization, maturation, and fusion. *Cell* 108, 717–725.
- Leal, E.S., Aucar, M.G., Gebhard, L.G., *et al.*, 2017. Discovery of novel dengue virus entry inhibitors via a structure-based approach. *Bioorg. Med. Chem. Lett.* 27, 3851–3855.
- Leonel, C.A., Lima, W.G., Dos Santos, M., *et al.*, 2018. Pharmacophoric characteristics of dengue virus NS2B/NS3pro inhibitors: A systematic review of the most promising compounds. *Arch. Virol.* 163, 575–586.
- Lescar, J., Luo, D., Xu, T., *et al.*, 2008. Towards the design of antiviral inhibitors against flaviviruses: The case for the multifunctional NS3 protein from Dengue virus as a target. *Antivir. Res.* 80, 94–101.
- Li, H., Clum, S., You, S., Ebner, K.E., Padmanabhan, R., 1999. The serine protease and RNA-stimulated nucleoside triphosphatase and RNA helicase functional domains of dengue virus type 2 NS3 converge within a region of 20 amino acids. *J. Virol.* 73, 3108–3116.
- Li, Z., Khaliq, M., Zhou, Z., *et al.*, 2008. Design, synthesis, and biological evaluation of antiviral agents targeting flavivirus envelope proteins. *J. Med. Chem.* 51, 4660–4671.
- Luo, D., Wei, N., Doan, D.N., *et al.*, 2010. Flexibility between the protease and helicase domains of the dengue virus NS3 protein conferred by the linker region and its functional implications. *J. Biol. Chem.* 285, 18817–18827.
- Luo, D., Xu, T., Hunke, C., *et al.*, 2008. Crystal structure of the NS3 protease-helicase from dengue virus. *J. Virol.* 82, 173–183.
- Luzhkov, V.B., Selisko, B., Nordqvist, A., *et al.*, 2007. Virtual screening and bioassay study of novel inhibitors for dengue virus mRNA cap (nucleoside-2′O)-methyltransferase. *Bioorg. Med. Chem.* 15, 7795–7802.
- Manvar, D., Kucukguzel, I., Erensoy, G., *et al.*, 2016. Discovery of conjugated thiazolidinone-thiadiazole scaffold as anti-dengue virus polymerase inhibitors. *Biochem. Biophys. Res. Commun.* 469, 743–747.
- Martins, I.C., Gomes-Neto, F., Faustino, A.F., *et al.*, 2012. The disordered N-terminal region of dengue virus capsid protein contains a lipid-droplet-binding motif. *Biochem. J.* 444, 405–415.
- Mastrangelo, E., Pezzullo, M., De Burghgraeve, T., *et al.*, 2012. Ivermectin is a potent inhibitor of flavivirus replication specifically targeting NS3 helicase activity: New prospects for an old drug. *J. Antimicrob. Chemother.* 67, 1884–1894.
- Ma, L., Jones, C.T., Groesch, T.D., Kuhn, R.J., Post, C.B., 2004. Solution structure of dengue virus capsid protein reveals another fold. *Proc. Natl. Acad. Sci. USA* 101, 3414–3419.
- Milani, M., Mastrangelo, E., Bollati, M., *et al.*, 2009. Flaviviral methyltransferase/RNA interaction: Structural basis for enzyme inhibition. *Antivir. Res.* 83, 28–34.
- Modis, Y., Ogata, S., Clements, D., Harrison, S.C., 2003. A ligand-binding pocket in the dengue virus envelope glycoprotein. *Proc. Natl. Acad. Sci. USA* 100, 6986–6991.
- Modis, Y., Ogata, S., Clements, D., Harrison, S.C., 2004. Structure of the dengue virus envelope protein after membrane fusion. *Nature* 427, 313–319.
- Morris, G.M., Goodsell, D.S., Halliday, R.S., *et al.*, 1998. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.* 19, 1639–1662.
- Nguyen, N.M., Tran, C.N., Phung, L.K., *et al.*, 2013. A randomized, double-blind placebo controlled trial of balapiravir, a polymerase inhibitor, in adult dengue patients. *J. Infect. Dis.* 207, 1442–1450.
- Nguyet, M.N., Duong, T.H., Trung, V.T., *et al.*, 2013. Host and viral features of human dengue cases shape the population of infected and infectious *Aedes aegypti* mosquitoes. *Proc. Natl. Acad. Sci. USA* 110, 9072–9077.
- Noble, C.G., Seh, C.C., Chao, A.T., Shi, P.Y., 2012. Ligand-bound structures of the dengue virus protease reveal the active conformation. *J. Virol.* 86, 438–446.
- Oprea, T.I., Bauman, J.E., Bologa, C.G., *et al.*, 2011. Drug repurposing from an academic perspective. *Drug Discov. Today Ther. Strat.* 8, 61–69.
- Panya, A., Bangphoomi, K., Choowongkamon, K., Yenchitsomanus, P.T., 2014. Peptide inhibitors against dengue virus infection. *Chem. Biol. Drug Des.* 84, 148–157.
- Paul, A., Vibhuti, A., Raj, S., 2016. Molecular docking NS4B of DENV 1–4 with known bioactive phyto-chemicals. *Bioinformation* 12, 140–148.
- Perera-Lecoin, M., Meertens, L., Carnec, X., Amara, A., 2013. Flavivirus entry receptors: An update. *Viruses* 6, 69–88.
- Phong, W.Y., Moreland, N.J., Lim, S.P., *et al.*, 2011. Dengue protease activity: The structural integrity and interaction of NS2B with NS3 protease and its potential as a drug target. *Biosci. Rep.* 31, 399–409.
- Podvinec, M., Lim, S.P., Schmidt, T., *et al.*, 2010. Novel inhibitors of dengue virus methyltransferase: Discovery by in vitro-driven virtual screening on a desktop computer grid. *J. Med. Chem.* 53, 1483–1495.
- Powers, C.N., Setzer, W.N., 2016. An in-silico investigation of phytochemicals as antiviral agents against dengue fever. *Comb. Chem. High Throughput Screen.* 19, 516–536.
- Prusis, P., Junaid, M., Petrovska, R., *et al.*, 2013. Design and evaluation of substrate-based octapeptide and non substrate-based tetrapeptide inhibitors of dengue virus NS2B–NS3 proteases. *Biochem. Biophys. Res. Commun.* 434, 767–772.
- Qamar, M.T., Mumtaz, A., Naseem, R., *et al.*, 2014. Molecular docking based screening of plant flavonoids as Dengue NS1 inhibitors. *Bioinformation* 10, 460–465.
- Rey, F.A., 2003. Dengue virus envelope glycoprotein structure: New insight into its interactions during viral entry. *Proc. Natl. Acad. Sci. USA* 100, 6899–6901.
- El Sahili, A., Lescar, J., 2017. Dengue virus non-structural protein 5. *Viruses* 9.
- Scaturro, P., Trist, I.M., Paul, D., *et al.*, 2014. Characterization of the mode of action of a potent dengue virus capsid inhibitor. *J. Virol.* 88, 11540–11555.
- Schüller, A., Yin, Z., Chia, C.B., *et al.*, 2011. Tripeptide inhibitors of dengue and West Nile virus NS2B–NS3 protease. *Antivir. Res.* 92, 96–101.

- Singh, J., Kumar, M., Mansuri, R., Sahoo, G.C., Deep, A., 2016. Inhibitor designing, virtual screening, and docking studies for methyltransferase: A potential target against dengue virus. *J. Pharm. Bioallied Sci.* 8, 188–194.
- Sivakumar, D., Sivaraman, T., 2011. In silico designing and screening of lead compounds to NS5-methyltransferase of dengue viruses. *Med Chem* 7, 655–662.
- Smit, J.M., Moesker, B., Rodenhuis-Zybert, I., Wilschut, J., 2011. Flavivirus cell entry and membrane fusion. *Viruses* 3, 160–171.
- Steuer, C., Gege, C., Fischl, W., *et al.*, 2011. Synthesis and biological evaluation of α -ketoamides as inhibitors of the Dengue virus protease with antiviral activity in cell-culture. *Bioorg. Med. Chem.* 19, 4067–4074.
- Swarbrick, C.M.D., Basavannacharya, C., Chan, K.W.K., *et al.*, 2017. NS3 helicase from dengue virus specifically recognizes viral RNA sequence to ensure optimal replication. *Nucleic Acids Res.* 45, 12904–12920.
- Sweeney, N.L., Hanson, A.M., Mukherjee, S., *et al.*, 2015. Benzothiazole and pyrrolone flavivirus inhibitors targeting the viral helicase. *ACS Infect. Dis.* 1, 140–148.
- Tajima, S., Takasaki, T., Kurane, I., 2008. Characterization of Asn130-to-Ala mutant of dengue type 1 virus NS1 protein. *Virus Genes* 36, 323–329.
- Tambunan, U.S.F., Nasution, M.A.F., Azhima, F., *et al.*, 2017. Modification of S-Adenosyl-L-Homocysteine as inhibitor of nonstructural protein 5 methyltransferase dengue virus through molecular docking and molecular dynamics simulation. *Drug Target Insights* 11.1177392817701726.
- Tambunan, U.S., Zahroh, H., Utomo, B.B., Parikesit, A.A., 2014. Screening of commercial cyclic peptide as inhibitor NS5 methyltransferase of dengue virus through molecular docking and molecular dynamics simulation. *Bioinformation* 10, 23–27.
- Tan, S.K., Pippen, R., Yusof, R., *et al.*, 2006. Inhibitory activity of cyclohexenyl chalcone derivatives and flavonoids of fingerroot, *Boesenbergia rotunda* (L.), towards dengue-2 virus NS3 protease. *Bioorg. Med. Chem. Lett.* 16, 3337–3340.
- Tarantino, D., Cannalire, R., Mastrangelo, E., *et al.*, 2016. Targeting flavivirus RNA dependent RNA polymerase through a pyridobenzothiazole inhibitor. *Antivir. Res.* 134, 226–235.
- Thomsen, R., Christensen, M.H., 2006. MolDock: A new technique for high-accuracy molecular docking. *J. Med. Chem.* 49, 3315–3321.
- Trott, O., Olson, A.J., 2010. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* 31, 455–461.
- Utama, A., Shimizu, H., Morikawa, S., *et al.*, 2000. Identification and characterization of the RNA helicase activity of Japanese encephalitis virus NS3 protein. *FEBS Lett.* 465, 74–78.
- Vasilakis, N., Ooi, M., Rabaa, M., *et al.*, 2013. The daemon in the forest-emergence of a new dengue serotype in SouthEast Asia. In: *Proceedings of the International Conference on Dengue and Dengue Haemorrhagic Fever*, Bangkok, Thailand.
- Venkatesham, A., Saudi, M., Kaptein, S., *et al.*, 2017. Aminopurine and aminoquinazoline scaffolds for development of potential dengue virus inhibitors. *Eur. J. Med. Chem.* 126, 101–109.
- Vincetti, P., Caporuscio, F., Kaptein, S., *et al.*, 2015. Discovery of multitarget antivirals acting on both the dengue Virus NS5-NS3 interaction and the host Src/Fyn kinases. *J. Med. Chem.* 58, 4964–4975.
- Wang, Q.Y., Patel, S.J., Vangrevelinghe, E., *et al.*, 2009. A small-molecule dengue virus entry inhibitor. *Antimicrob. Agents Chemother.* 53, 1823–1831.
- Wu, H., Bock, S., Snitko, M., *et al.*, 2015. Novel dengue virus NS2B/NS3 protease inhibitors. *Antimicrob. Agents Chemother.* 59, 1100–1109.
- Xu, H.T., Colby-Germinario, S.P., Hassounah, S., *et al.*, 2015. Identification of a pyridoxine-derived small-molecule inhibitor targeting dengue virus RNA-dependent RNA polymerase. *Antimicrob. Agents Chemother.* 60, 600–608.
- Xu, T., Sampath, A., Chao, A., *et al.*, 2005. Structure of the dengue virus helicase/nucleoside triphosphatase Catalytic domain at a resolution of 2.4 Å. *J. Virol.* 79, 10278–10288.
- Yang, J.-M., Chen, Y.-F., Tu, Y.-Y., Yen, K.-R., Yang, Y.-L., 2007. Combinatorial computational approaches to identify tetracycline derivatives as flavivirus inhibitors. *PLOS ONE* 2, e428.
- Yap, T.L., Xu, T., Chen, Y.L., *et al.*, 2007. Crystal structure of the dengue virus RNA-dependent RNA polymerase catalytic domain at 1.85-angstrom resolution. *J. Virol.* 81, 4753–4765.
- Yennamalli, R., Subbarao, N., Kampmann, T., *et al.*, 2009. Identification of novel target sites and an inhibitor of the dengue virus E protein. *J. Comput. Aided Mol. Des.* 23, 333–341.
- Yildiz, M., Ghosh, S., Bell, J.A., Sherman, W., Hardy, J.A., 2013. Allosteric inhibition of the NS2B-NS3 protease from dengue virus. *ACS Chem. Biol.* 8, 2744–2752.
- Yin, Z., Chen, Y.L., Schul, W., *et al.*, 2009. An adenosine nucleoside inhibitor of dengue virus. *Proc. Natl. Acad. Sci. USA* 106, 20435–20439.
- Yin, Z., Patel, S.J., Wang, W.L., *et al.*, 2006. Peptide inhibitors of dengue virus NS3 protease. Part 2: SAR study of tetrapeptide aldehyde inhibitors. *Bioorg. Med. Chem. Lett.* 16, 40–43.
- Yotmanee, P., Rungrotmongkol, T., Wichapong, K., *et al.*, 2015. Binding specificity of polypeptide substrates in NS2B/NS3pro serine protease of dengue virus type 2: A molecular dynamics study. *J. Mol. Gr. Model.* 60, 24–33.
- Zhou, Z., Khaliq, M., Suk, J.E., *et al.*, 2008. Antiviral compounds discovered by virtual screening of small-molecule libraries against dengue virus E protein. *ACS Chem. Biol.* 3, 765–775.
- Zou, J., Xie, X., Wang, Q.Y., *et al.*, 2015. Characterization of dengue virus NS4A and NS4B protein interaction. *J. Virol.* 89, 3455–3470.

Relevant Website

www.sigmaaldrich.com
Sigma-Aldrich.

Drug Repurposing and Multi-Target Therapies

Ammu P Kumar and Suryani Lukman, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Minh N Nguyen, Agency for Science, Technology and Research, Singapore

© 2019 Elsevier Inc. All rights reserved.

Drug Repurposing: Polypharmacology, Benefits and Drawbacks

Drug repurposing is based on polypharmacology, a field of study related to design, discovery, synthesis, and use of pharmaceutical agents (drugs, small molecules, antibodies, stabilized peptides, natural products, etc.) that act on multiple targets (genes or gene products) or disease pathways. Complex diseases (such as neurodegenerative diseases and cancers) involve networks of multiple genes, hence necessitate multi-pharmacophores as a part of their therapeutic and system biology-based approaches. By specifically regulating multiple targets, more effective drugs can be developed through polypharmacology (Anighoro *et al.*, 2014). In need of treatment for critical illnesses and neglected diseases, efforts to match old drugs, that are already known to be safe and/or approved by US Food and Drug Administration (FDA) for diseases are in ongoing demand.

Since pharmaceutical agents can interact with multiple targets, unintended interactions between drugs and off-targets could result in toxicities or side effects. Side effects arising from a particular drug have been serendipitously used to usher a new use, such as treating other conditions and diseases. For example, sildenafil, commercially known as Viagra, is a blockbuster drug used to treat male erectile dysfunction; Viagra's clinical effects on treating erectile dysfunction were observed when it was studied for treating hypertension (Reaume, 2011).

Drug repurposing can reduce healthcare costs, given how laborious, time-consuming, and costly *de novo* drug discovery projects have been. Drug-candidate molecules that have been shown as being safe, yet insufficiently effective for the intended target (Oprea *et al.*, 2011), can be considered for other targets, hence possibly reducing the costs of clinical trial. The increasing amount and availability of public and open source databases, knowledge, algorithms, and servers, have encouraged more participants of drug repurposing projects.

Drug discovery efforts have resulted in discovery of new modes of actions for approved drugs. For example, Fasudil is a Rho-kinase inhibitor and vasodilator, which was repurposed to improve memory and treat several neurodegenerative disorders (Iorio *et al.*, 2010), such as Alzheimer's disease. In addition, orphan, rare, and neglected diseases with limited fundings have benefited from drug discovery efforts. For example, closantel, a veterinary anthelmintic, was repurposed to inhibit the chitin metabolism of the filarial nematode *Onchocerca volvulus*, that causes a neglected tropical disease Onchocerciasis (river blindness) (Gloeckner *et al.*, 2010). Drug repurposing has also been achieved through combining approved drugs (multidrug cocktails) in novel ways to synergistically treat diseases, including HIV infection, cancer, and *Mycobacterium tuberculosis* infection (Borisy *et al.*, 2003).

While drug repurposing brings numerous benefits, there are associated drawbacks of such an effort. Our understandings of the pathways and mechanisms of many complex diseases remain incomplete (Reddy and Zhang, 2013), yet efforts for drug repurposing often require analyzing complete data. Public databases have increasingly provided enormous information that can facilitate drug repurposing, yet they are non-synchronized (Wiegiers *et al.*, 2009). According to Oprea *et al.* (2011), there is no systematic mechanism to obtain fundings to support drug repurposing projects hitherto, in comparison to *de novo* drug discovery projects. To address this challenge, drug discovery project leaders have sought limited fundings from private and/or non-for-profit organizations, such as the Bill & Melinda Gates Foundation, Howard Hughes Medical Institute, and Simons Foundation.

Databases and Computational Resources for Drug Repurposing

The process of a new drug development is costly and time-consuming, and therefore drug repurposing has emerged as a time-efficient and cost-effective strategy to discover new indications for already approved drugs (Fu *et al.*, 2013). A number of available and searchable drug databases has been designed and developed for supporting drug repurposing studies such as PROMISCUOUS (von Eichborn *et al.*, 2011), DRAR-CPI (Luo *et al.*, 2011), e-Drug3D (Pihan *et al.*, 2012), PharmDB (Lee *et al.*, 2012), PharmDB-K (Lee *et al.*, 2015), DrugPredict (Simon *et al.*, 2012), DrugMap Central (Fu *et al.*, 2013), DrugBank (Law *et al.*, 2014), RE:fine drugs (Moosavinasab *et al.*, 2016), Mantra 2.0 (Carrella *et al.*, 2014), repoDB (Brown and Patel, 2017), RepurposeDB (Shameer *et al.*, 2017), SIDER (Kuhn *et al.*, 2016), SWEETLEAD (Novick *et al.*, 2013), DeSigN (Lee *et al.*, 2017), and Connectivity Map (Cmap) (Lamb, 2006).

PROMISCUOUS database (see "Relevant Websites section") contains comprehensive data of drug-target interaction, drug side effect, and protein-protein interaction data. PROMISCUOUS has been proven to be useful for drug repurposing by connecting structural similarity of drugs and their side effects to protein – protein interactions. In addition, the integrated network visualization tools of PROMISCUOUS allow researchers to explore and understand the analysis of the interplay between drugs and targets, as well as identify candidates for drug repurposing.

DRAR-CPI server (see "Relevant Websites section") uses the database of 254 active form from 166 drugs with known adverse drug reaction (ADR) and 385 pockets from 353 human protein targets for predicting drug repurposing potential via chemical-protein interactome (CPI). When users submit a drug molecule, the DRAR-CPI server computes the binding energies of this drug with all 385

protein pockets in the database using the DOCK program (Ewing *et al.*, 2001), and then generates its CPI profile. Based on this CPI profile, the server identifies the association scores between the drug and all 166 drugs in the database, and predicts off-target proteins that possibly interact with it. Recently, the DRAR-CPI has been upgraded to DPDR-CPI server (see “Relevant Websites section”) with the larger database of 2515 drugs and 611 human protein targets (Luo *et al.*, 2016). The AutoDock Vina program (Trott and Olson, 2009) is used in the new server for docking of drug molecules with human protein targets to generate CPI profiles.

The e-Drug3D (see “Relevant Websites section”) provides the annotated database of 3D structures from FDA approved drugs as well as commercial sub-structures (fragments) of drugs. This database of 3D structures of drug can be used for virtual screening applications, such as fragment-based drug design and drug repurposing. DrugPredict collects and identifies effect profiles and 3D structures of small molecule drugs (Simon *et al.*, 2012). Using docking calculations, DrugPredict determines interactions of each drug and non-target protein binding sites, which are used for new drug effect predictions. DrugMap Central collects multi-level drug data from various established databases, including drug targets, chemical structures, target-related signaling pathways, FDA and clinical trial information (Fu *et al.*, 2013). DrugMap Central provides online query tool and visualization-based search to support researchers for drug repurposing studies.

PharmDB is an integrated database of drug development, disease indications, associated proteins, and their known interactions (Lee *et al.*, 2012). The Shared Neighborhood Scoring algorithm is developed for PharmDB to identify new indications of known drugs, which can be used for predicting drug repurposing potential. Another database, PharmDB-K offers comprehensive information relating to Traditional Korean Medicine (TKM), associated drugs (compound), disease indication, and protein relationships (Lee *et al.*, 2015). In PharmDB-K, information is organized as a network with five kinds of nodes: TKMs, drugs, diseases, proteins, and side effects. The website also integrates diverse tools to explore TKM-disease, TKM-drug, drug-disease, drug-drug, drug-protein, drug-side effect, disease-protein, and protein-protein relationships by analyzing the network.

RE:fine drugs contains information on 916 drugs, 567 genes and 1770 diseases (Moosavinasab *et al.*, 2016). The database is constructed based on the Transitive Property of Equality between drug-gene and gene-disease pairs of information extracted from previously published datasets (Moosavinasab *et al.*, 2016). Users can explore the database using drug/gene/disease information and identify drug-disease candidates for repurposing. The results can also be sorted based on P-value, odds ratio, literature support, clinical trials, disease name and/or drug name.

Mantra 2.0 maintains information on drugs as a network, with drugs as nodes and edges highlighting communities of drugs sharing similar mode of action (Carrella *et al.*, 2014; Iorio *et al.*, 2010). The network is constructed by exploiting similarities in gene expression profiles following drug treatment (Carrella *et al.*, 2014). Mantra allows users to provide gene expression profiles before and after drug treatment in one or multiple cell types as input. These gene expression profiles are transformed into a drug node and integrated in the drug network, which allows users to inspect its mode of action and repurposing opportunities.

The databases such as repoDB, RepurposeDB, SIDER, SWEETLEAD and DrugBank also have various information on repurposed/approved drugs (see Table 1) that can be used as references for drug repurposing experiments. The repoDB also has information on failed drugs (Brown and Patel, 2017). Information on failed drugs can be useful as there are many drugs that are abandoned after clinical trials due the lack of efficacy for their primary indications, and which possess the same benefits of approved drugs (Novick *et al.*, 2013). SWEETLEAD (Novick *et al.*, 2013) is a highly curated database of chemical structures for the globally approved drugs. It is developed with the objective of providing well curated, high quality information of the known approved drugs for drug repurposing experiments. DrugBank is also known for its expertly curated data on drugs and is the referential drug data source for many well-known databases (Law *et al.*, 2014), such as PDB (Rose *et al.*, 2013), PubChem (NCBI Resource Coordinators, 2013), UniProt (UniProt Consortium, 2013), among others. DrugBank has detailed information on drugs (i.e., chemical, pharmacological and pharmaceutical) and drug targets (i.e., sequence, structure, and pathway) (Law *et al.*, 2014). The latest version of DrugBank (version 5.0.9) has information on 10,505 drugs.

Resources such as DeSigN and Cmap have large collections of drug-induced signatures and they integrate computational tools that facilitate drug repositioning based on genomic screening (see Virtual Screening) (Lee *et al.*, 2017; Lamb, 2006). The details of the databases discussed here are detailed in Table 1.

Computational Tools for Drug Repurposing

Currently, there are two major structure-based categories for drug repurposing by using (i) ligands/compounds similarity and (ii) binding pocket similarity. Computational methods using 1D or 2D representations for compound structures are commonly used in ligand-based virtual screening because chemical molecules are typically represented by 1D fingerprints or 2D molecular formula (Schwartz *et al.*, 2013; Helguera *et al.*, 2008; Hong *et al.*, 2008). However, the 3D structural information of compounds and target proteins as well as their atomic interactions in the 3D space are not captured by 1D and 2D methods. Furthermore, although the 1D and 2D representations of chemical molecules are not similar, these molecules could share biological activities if their 3D shape are similar. Binding affinity between compounds and target proteins is governed by atomic interactions in the 3D space. Therefore, 3D methods have potential to overcome the limitations of 1D and 2D methods and enhance performance for ligand-based virtual screening. In the past, there have been efforts with some degree of success in developing computational methods to identify similarities between the 3D structures of compounds (Schuffenhauer *et al.*, 2000; Kombo *et al.*, 2013; Shin *et al.*, 2015). Recently, we have applied our CLICK algorithm (Nguyen and Madhusudhan, 2011), whose main strength emerges from its unique ability to carry out topology independent comparisons, for comparing 3D structures of compounds.

Table 1 Databases for drug repurposing^a

Databases	Key features	Website
PROMISCOUS von Eichborn <i>et al.</i> (2011)	- Contains information on 25,000 drugs (including withdrawn and experimental drugs), their side effects and targets. - Users can explore the database based on drugs, their targets, side effects or, metabolic and signaling pathways.	http://bioinformatics.charite.de/promiscuous
DPDR-CPI Luo <i>et al.</i> (2016)	- Contains information on 2515 drugs. - Predicts off-targets and potential indications for the small molecule input from the user.	https://cpi.bio-x.cn/dpdr/
e-Drug3D Pihan <i>et al.</i> (2012)	- Contains ready to screen SD files of drugs and commercial drug fragments. - Contains information on 1852 molecular structures.	http://chemoinfo.ipmc.cnrs.fr/MOLDB/
DrugPredict Simon <i>et al.</i> (2012)	- Contains information on 100,000 small molecule compounds including 1200 approved drugs. - Users can explore the database based on drugs/any desired effects.	http://www.drugpredict.com/
PharmDB Lee <i>et al.</i> (2012)	Contains information on 11,792 drugs, 38,057 proteins and 6607 diseases and their relationship.	http://www.i-pharm.org/
PharmDB-K Lee <i>et al.</i> (2015)	- Provides information on Traditional Korean Medicine (TKM). - Contains information on 262 TKMs, 7815 drugs, 3721 diseases, 32,373 proteins, and 1887 side effects.	http://pharmdb-k.org/
RE:fine drugs Moosavinasab <i>et al.</i> (2016)	- Contains information on 916 drugs, their target genes and associated diseases. - Users can explore the database based on drug, indication or gene symbol.	http://drug-repurposing.nationwidechildrens.org
Mantra 2.0 Carrella <i>et al.</i> (2014)	- Provides the users insights on mode of action of drugs by comparing drug-induced signatures. - Contains information on 1309 small molecules.	http://mantra.tigem.it/
repoDB Brown and Patel (2017)	- Contains information on 1571 drugs (including approved and failed drugs), their indications and clinical trial status. - Users can explore the database based on drug/disease information.	http://apps.chiragjpgroup.org/repoDB/
RepurposeDB Shameer <i>et al.</i> (2017)	- Contains information on 253 repurposed drugs and their indications. - Users can explore the database based on drug/disease information, side effects, drug targets or pathways. - Facilitates chemical similarity search for new compounds and sequence similarity search for new protein-drugs against RepurposeDB. - Allows drug repurposing investigators to report their results to the community.	http://repurposedb.dudleylab.org/
SIDER Kuhn <i>et al.</i> (2016)	- Contains information on 1430 drugs and their side effects. - Users can explore the database based on drugs/side effect information.	http://sideeffects.embl.de/
SWEETLEAD Novick <i>et al.</i> (2013)	- Contains accurate chemical structures of 4442 compounds (including approved drugs and non-toxic chemicals). - The database can be downloaded and used for virtual screening/drug repurposing campaigns.	https://simtk.org/home/sweetlead
DrugBank 5.0.9 Law <i>et al.</i> (2014)	- Contains information on 10,505 drugs (including 1733 approved small molecule drugs, 870 approved biotech drugs, 105 nutraceuticals and over 5025 experimental drugs). - Provides detailed information on drugs (i.e., chemical, pharmacological and pharmaceutical) and drug targets (i.e., sequence, structure, and pathway) information.	https://www.drugbank.ca/
DeSigN Lee <i>et al.</i> (2017)	- Contains information on cancer-associated gene expression profiles from 140 drugs. - Allows drug repositioning based on genomic screening.	http://design.cancerresearch.my/
Cmap Lamb (2006)	- Contains gene expression profiles from 5000 small molecule compounds. - Allows drug repositioning based on genomic screening.	https://www.broadinstitute.org/connectivity-map-cmap

^aSome relevant databases for drug repurposing, their key features and websites are listed as of 9th November 2017. The table lists only databases having active websites.

It is well understood that drugs bind to their protein targets because of complementarity in shapes of ligands fitting into their binding pocket. However, the shape of a particular binding pocket on a target is possibly found in some other proteins, given the diversity of protein shapes in a cell. This will inevitably lead to the binding of the drug to other proteins (off-target proteins) possessing structurally similar binding pockets (John Fox *et al.*, 2016). Hence, the discovery of the off-target proteins of a drug can lead to further use of this drug for additional targets in disease treatment. Clearly, the discovery of the off-target proteins will rely on an accurate determination of similarity of binding pocket. A major effort towards developing tools to search for such

similarities has been focused on local structural alignment approaches (Shulman-Peleg *et al.*, 2007; Konc and Janežič, 2010). One such effort is the CLICK method (Nguyen *et al.*, 2011) that performs 3D structural superposition on pairs of structures based on similarity of local structural packing, and thus is capable of aligning structures with dissimilar conformations or even molecular types (Nguyen and Verma, 2015; Nguyen *et al.*, 2017b). These unique properties make CLICK particularly ideal for comparing the similarity of binding pocket (Lukman *et al.*, 2017; Nguyen *et al.*, 2017a).

In addition, investigation of complex interactions between drugs and diseases is crucial for new drug-disease association discovery and drug repurposing (Yang *et al.*, 2014). However, it is difficult to identify the comprehensive interactions of drugs and diseases because of the polypharmacological profiles of drugs (Reddy and Zhang, 2013) and complex diseases induced by genes' collective abnormalities (Wu *et al.*, 2013). Disease-based computational methods have been developed to identify the interactions between drugs and diseases by using shared molecular pathology, associative indication transfer, and side effect similarity (Dudley *et al.*, 2011a,b). Recently, computational methods using pathway-based Bayesian inference (Pratanwanich and Lió, 2014), network propagation (Huang *et al.*, 2013), and network-based inference (Cheng *et al.*, 2012a,b) have been applied for investigating drug-disease interactions. In these methods, the network of drug – Target – Pathway – Gene – Disease is constructed from multi-level interactions of drugs and diseases integrated from known databases. Next step, the interaction scores of drugs and diseases are computed by evaluating effects of drugs on multiple targets and pathways (Yang *et al.*, 2014). The new potential repurposing of the existing drug is then determined from the network of drug targets and disease-related genes (Pratanwanich and Lió, 2014).

Virtual Screenings: Genomic Screenings, High-Throughput Screenings

Virtual screening can be defined as a set of computer methods that analyse compounds in large databases in order to identify a smaller number of them for biological testing (Sottriffer, 2011). Virtual high-throughput screening (vHTS) and genomic screening are two virtual screening techniques that can be employed for drug repurposing. vHTS is the computational analogue of *in vitro* high-throughput screening (HTS). vHTS involves screening of large compound libraries to identify compounds/ligands that can bind to the biological target of interest with high affinity (Shoichet, 2004). Using vHTS, large collections of known compounds can be screened to identify their applicability to treat a new disease.

There are many public databases that can be employed for vHTS (Cheng *et al.*, 2012a,b). For example, ZINC (zinc.docking.org) is a public database that contains over twenty million commercially available compounds in biologically relevant representations, that can be downloaded in ready-to-dock formats (Irwin *et al.*, 2012). It is also possible to download the database in fractions based on physical properties, purchasability, and vendors, among other attributes. In a previous study, we have employed vHTS of 'Drugs Now' subset (a subset of 10,639,555 compounds having drug-like properties such as adequate chemical stability, metabolic stability, minimal toxic effects and oral bioavailability Lipinski, 2000) from the ZINC database, for the identification of inhibitors of aberrant proteins, such as (1) Ras proteins that are implicated in diverse cancers and developmental diseases (Grant *et al.*, 2011), (2) protein tyrosine phosphatase 1B that is associated with diabetes, obesity, cancers, and neurodegenerative disorders (Kumar *et al.*, 2018). In these examples, the inhibitors act allosterically. Indeed, drug repurposing efforts will generally be more beneficial if the drug binding sites are allosteric rather than orthosteric (Nussinov and Tsai, 2012).

vHTS can be classified as structure-based and ligand-based approaches (Fig. 1). Structure-based screening is employed when the three-dimensional structure of a disease-associated target is known. When the target structure is unknown and its prediction is challenging, ligand-based screening is employed. The objective of both these approaches is the same, i.e., to select a smaller subset of compounds from larger libraries and rank them using some scoring criteria, to be further used for experimental testing of their biological activity.

Structure-based screening involves molecular docking of candidate ligands from the library into a protein (target) followed by applying a scoring function (Liang *et al.*, 2009; Deng *et al.*, 2004; Amari *et al.*, 2006) to estimate the likelihood that the ligand will bind to the protein (Kroemer, 2007; Wang *et al.*, 2000). Since the computational cost of the docking is directly proportional to the number of compounds, pre-filtering of large libraries is highly desirable (Radusky *et al.*, 2017). Candidate compounds from the libraries can be filtered based on their drug-like properties (Lipinski, 2000; Lipinski *et al.*, 2001). Target-specific filters can also be used for creating focused libraries (Gozalbes *et al.*, 2008; Sage *et al.*, 2011). For example, more recently Radusky *et al.* developed 'LigQ' server (see "Relevant Websites section") that can assist in various steps of virtual screening (Radusky *et al.*, 2017). The webserver can be employed in initial stages of virtual screening to identify potential binders for targets from large databases. Enriching compound libraries in these ways can greatly reduce the computational cost of subsequent docking. The target and ligand structures also need to be prepared before docking by assigning proper tautomeric, stereoisomeric, and protonation states, among others (Rapp *et al.*, 2009; ten Brink and Exner, 2010). After pre-processing of target and compound library, docking is performed.

Many docking programs are available to computationally model the ligand – Target interaction. Some examples of docking programs in vHTS are listed in Table 2. These docking programs generally involve a search algorithm that searches the conformational space to find docking poses and a scoring function to predict the affinity of the ligand with the target in that pose (Sliwoski *et al.*, 2014). The candidate ligands that are selected through docking are then post-processed by examining their binding scores, interactions, and binding pose, among others, based on which a smaller number of compounds are selected for experimental testing (Lionta *et al.*, 2014). Numerous success stories have been reported in drug repurposing through the use of structure-based screening (Palos *et al.*, 2017; Lukman *et al.*, 2017; Bi *et al.*, 2017; Sahoo *et al.*, 2016). For example, Chan *et al.* (2011) identified an FDA approved drug methylene blue from a database of over 3000 compounds using structure-based screening to

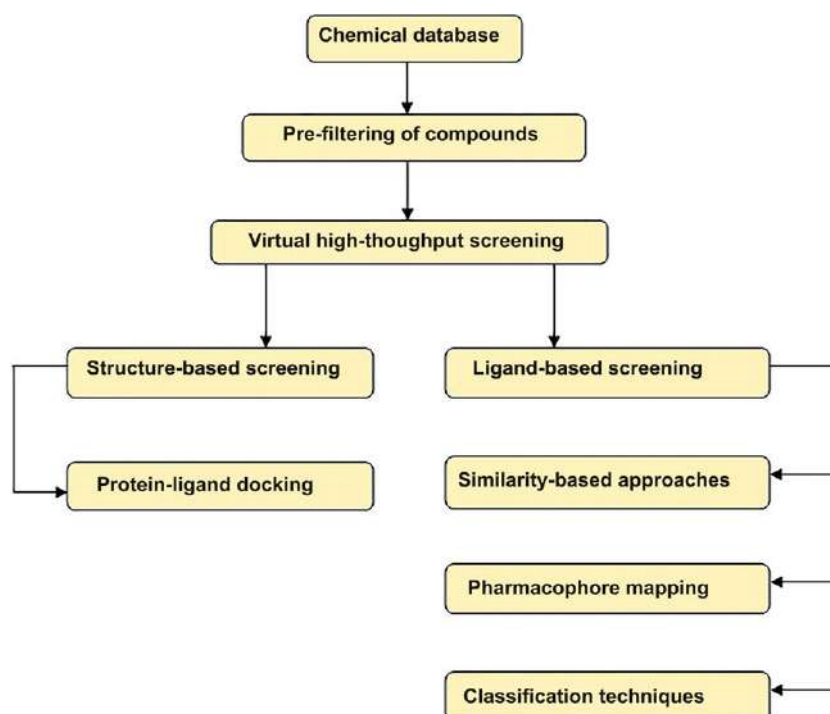


Fig. 1 Virtual high-throughput screening (vHTS) and its classifications. In vHTS, large compound databases are screened to identify candidate ligands for experimental testing. Pre-filtering of compounds in the databases is usually done before screening. vHTS can be classified as structure-based screening and ligand-based screening. Structure-based screening involves protein-ligand docking. Ligand-based screening can be classified as similarity-based approaches, pharmacophore mapping and classification techniques.

Table 2 Some docking programs employed in virtual high-throughput screening^a

Program	Free for academic license ?	Website
AutoDock Forli <i>et al.</i> (2016), Morris <i>et al.</i> (2008)	Yes	http://autodock.scripps.edu/
FlexX Rarey <i>et al.</i> (1996)	No	https://omictools.com/flexx-tool
Surflex Jain (2003)	No	http://www.jainlab.org/downloads.html
GOLD Jones <i>et al.</i> (1997)	No	https://www.ccdc.cam.ac.uk/solutions/csd-discovery/components/gold/
ICM Abagyan <i>et al.</i> (1994), Neves <i>et al.</i> (2012)	No	http://www.molsoft.com/docking.html
Glide Friesner <i>et al.</i> (2004)	No	https://www.schrodinger.com/
Lead Finder Stroganov <i>et al.</i> (2008)	No	http://moltech.ru/
MOE-Dock Sun (2016), Corbeil <i>et al.</i> (2012)	No	https://www.chemcomp.com/MOE-Structure_Based_Design.htm
iGEMDOCK	Yes	http://gemdock.life.nctu.edu.tw/dock/igemdock.php
AutoDock Vina Trott and Olson (2009)	Yes	http://vina.scripps.edu/
rDock Ruiz-Carmona <i>et al.</i> (2014)	Yes	http://rdock.sourceforge.net/
UCSF Dock Allen <i>et al.</i> (2015)	Yes	http://dock.compbio.ucsf.edu/

^aSome examples of docking programs and their websites are listed.

Note: Modified from Cheng, T., Li, Q., Zhou, Z., Wang, Y., Bryant, S.H., 2012. Structure-based virtual screening for drug discovery: A problem-centric review. *AAPS J.* 14, 133–141. Available from: <https://doi.org/10.1208/s12248-012-9322-0>.

stabilize the c-myc Pu27 G-quadruplex DNA that are putatively present in the promoter regions of oncogenes from the human genome Chan *et al.* (2011). Li *et al.* repositioned two organohalogen drugs to inhibit B-Raf protein kinase which is mutated in a broad range of human cancers by developing a docking program that takes into account halogen bond interactions Li *et al.* (2016).

Ligand-based screening makes use of the knowledge of ligands that binds to the target of interest for screening, based on the notion that ligands interacting with the same target show same structural or physicochemical properties. Ligand-based screening can be classified as molecular similarity-based approaches, pharmacophore mapping and compound classification techniques (Acharya *et al.*, 2011). Molecular similarity-based approaches usually employ 2D/3D chemical similarity analysis of the active ligand structure against a library of molecules (Bajorath, 2011; Bologna *et al.*, 2006; Lindert *et al.*, 2013) to identify chemically similar compounds. Molecular descriptors such as fingerprints derived from molecular graphs (2D) or conformers (3D)

(Cereto-Massagué *et al.*, 2015), topological indices (Fukunishi and Nakamura, 2009), molecular fields (Cheeseright *et al.*, 2006), and others, can be used for the similarity search of ligands. As a measure of comparison, similarity between molecular descriptors of two structures is quantified using a similarity coefficient. Most frequently, Tanimoto coefficient is used, which is defined as $N_{ab}/N_a + N_b - N_{ab}$, where N_a and N_b are the number of features in the fingerprint of compounds “a” and “b”, respectively and N_{ab} is the number of features in fingerprints of both the compounds.

Pharmacophore mapping relies on the knowledge of molecular features (such as hydrogen bonds, charged interactions, hydrophobic and aromatic contacts) that are necessary for molecular recognition of a ligand by a target. By using the knowledge about the structurally diverse ligands that bind to a target, a model of target/pharmacophore can be built and can be employed for rapid screening of large databases (Sukumar and Das, 2011; Sun, 2008; Lin, 2000; Patel *et al.*, 2002; Pan *et al.*, 2013; Goodford, 1985). The quality of fitting the ligand into the pharmacophore query is more commonly expressed by the root mean square deviation (RMSD) between the features of the query and atoms of the molecule (Langer and Hoffmann, 2006).

Compound classification techniques include clustering techniques (such as PCA), partitioning techniques (such as recursive partitioning), and machine learning approaches (such as support vector machines, decision trees, k-nearest neighbours, naïve Bayesian methods, and artificial neural networks). The goal of this approach is to predict the activity of the compound based on models derived from the training set as well as the ranking of the database compounds based on the probability of their activity (Melville *et al.*, 2009; Bajorath, 2001).

As in structure-based screening, pre-filtering of compound libraries in accordance with the goals and constraints of the study is also beneficial in ligand-based screening (Yan *et al.*, 2016; Glaab, 2016). Many recent works have reported the usage of ligand-based screening for drug repurposing (Paz *et al.*, 2017; Keiser *et al.*, 2009; Vasudevan *et al.*, 2012; Crisan *et al.*, 2017). For example, Crisan *et al.* (2017) employed pharmacophore model for virtual screening and identification of known drugs that could inhibit glycogen synthase kinase-3 (GSK-3) which is involved in diseases such as psychiatric and neurological diseases, inflammatory diseases, and cancer. Anighoro *et al.* (2015) employed 2D ligand-based similarity analysis of ChEMBL database (see “Relevant Websites section”) combined with support vector machine models and analysis of 3D structural information of ligand-target complexes, and identified a promising set of target combinations and associated ligands within the Hsp90 interactome, which contains several targets of key importance in cancer Anighoro *et al.* (2015).

A limitation of vHTS is that, it cannot identify the full spectrum of targets that a small molecule may be hitting, as a result of which drugs can elicit undesirable off-target effects. To overcome this drawback, vHTS can be integrated with deep molecular characterization and network analysis (Leung *et al.*, 2013). Network models allow integration of data from various information sources, capturing both quantitative and qualitative relationships between entities and the presence or absence of an interaction (Paolini *et al.*, 2006). For example connecting drugs by side effect similarity can provide insights into the drug’s side effects and may also allow prediction of novel off-targets (Campillos *et al.*, 2008).

Often, structure-based screening requires thorough knowledge of the 3D structures of targets and their mechanisms at atomistic levels. Methods to resolve the 3D structures, such as X-ray crystallography and nuclear magnetic resonance method, are time- and cost-consuming. Atomistic-level of target mechanisms, observed through molecular dynamics simulations (Lukman *et al.*, 2012), can be very critical for drug repurposing. When we do not know the underlying mechanism associated with a disease, genomic screening methods are particularly useful. In such cases candidate ligands can be predicted using the molecular signature (i.e., significantly up and down regulated genes) of the disease that is generated through the analysis of differences in genomic patterns from disease affected and unaffected individuals. The corresponding signature of differential gene expression can be compared with differential gene expression signature induced by drugs, (i.e., changes brought about in the genomic patterns after exposure to a drug). If both the disease-associated and drug-induced signatures are sufficiently negatively correlated (i.e., the effects induced by drugs are opposite to the effects associated with the disease), then the drug may be able to revert the disease-associated signature and hence it can be considered as candidate for repurposing (Sirota *et al.*, 2011). Another approach employed in genomic screening is to look for drugs that can induce similar effects in gene expressions, by comparing the drug-induced signatures. It can be hypothesized that two drugs that can induce the same changes share similar mode of action/therapeutic application (Iorio *et al.*, 2010). Candidate ligands are thus identified in genomic screening by their direct or inverse correlation to the query signature (Fig. 2).

Publicly available databases such as Cmap (see “Relevant Websites section”) (Lamb, 2006, 2007) and LINCS (see “Relevant Websites section”) (Duan *et al.*, 2014) have thousands of gene signatures representing a diverse range of FDA approved drugs. The data from these databases can be integrated with functional genomics databases like NCBI-GEO (see “Relevant Websites section”) (Edgar *et al.*, 2002) and ArrayExpress (see “Relevant Websites section”) (Brazma *et al.*, 2003) for drug repositioning studies using genomic screening. Many previous studies have shown the benefits of employing Cmap data in drug repositioning (Iorio *et al.*, 2009, 2010; Napolitano *et al.*, 2013; Silberberg *et al.*, 2012; Parkkinen and Kaski, 2014; Yu *et al.*, 2015; Jadamba and Shin, 2016). For example, Dudley *et al.* employed Cmap data for repurposing topiramate, an anticonvulsant drug for treating inflammatory bowel disease (IBD) (Dudley *et al.*, 2011a,b). They employed gene expression data from NCBI-GEO and created a gene expression signature for IBD using the Significance Analysis of Microarrays (SAM) software (see “Relevant Websites section”) (Tusher *et al.*, 2001). Then they systematically compared the signature to drug-induced signatures obtained from the Cmap. Drugs that are anti-correlated to the disease were selected based on their therapeutic scores that were computed based on a randomization algorithm.

Although genomic screening methods are advantageous in studying genome-wide patterns of how drugs are changing biological systems, there are still some limitations. Some portion of the genes that show significant expression differences in the drug-induced signatures, may be produced by drug side effects (Jadamba and Shin, 2016). Furthermore, the differential gene expression induced by drugs represents only a small subset of biological pathway (Jadamba and Shin, 2016). So as suggested in vHTS, the integration of

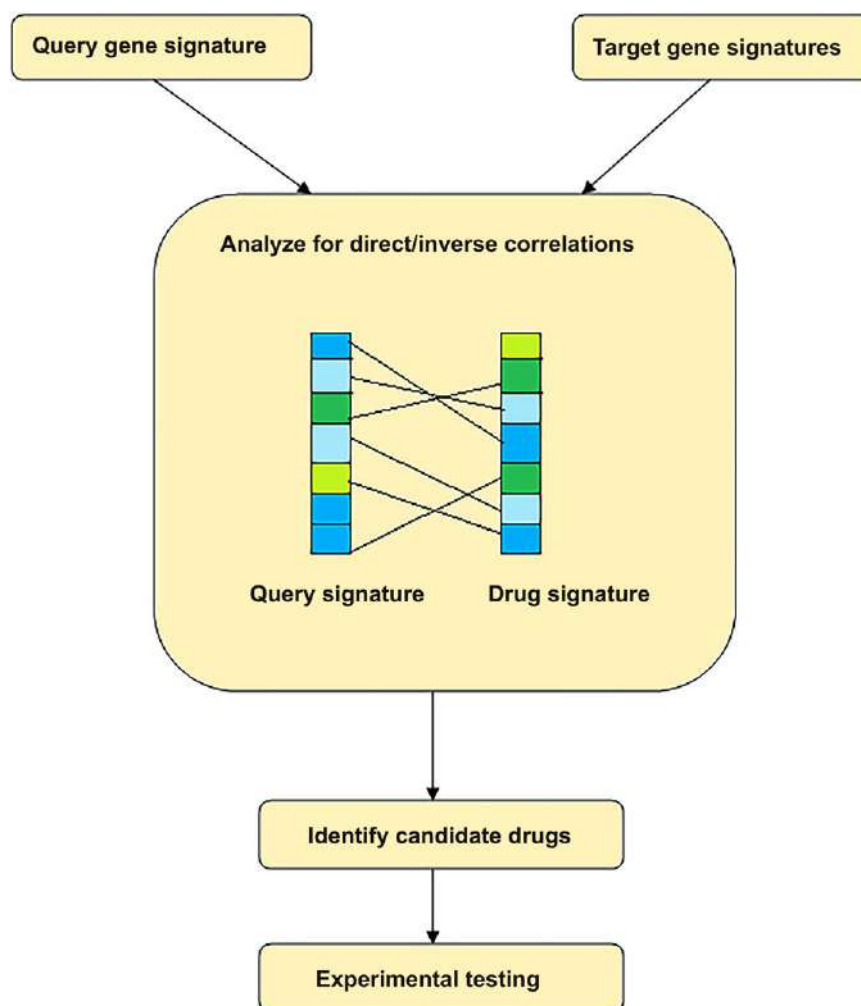


Fig. 2 Genomic screening for drug repurposing. In genomic screening approach, the query signature (either disease-associated or drug-induced signature) is compared with target signatures (drug-induced signatures) and their inverse or direct correlations are analyzed based on which the candidate ligands are chosen for experimental testing.

network models with genomic screening methods can provide better understanding of the drug-disease relationship. For example, more recently, [Mulas *et al.* \(2017\)](#) employed a network-based differential analysis of gene expression profiles to identify modes of actions of drugs. The proposed method is able to identify pathways significantly associated with the drug-induced signatures.

There are also many reports in literature of the combined use of the above-mentioned methods for drug repurposing studies. For example, [Han *et al.* \(2017\)](#) have recently identified a known drug that could inhibit Human enterovirus 71 (EV71) that can cause hand, foot and mouth disease. They have constructed EV71 protein interaction network and identified drugs that can target the interacting proteins. They used Cmap for the evaluation of drug influences on the genomic signatures of targets. Similarly, genomic screening methods can also be employed following vHTS for evaluating the candidate drugs. [Phatak *et al.*](#) have employed a multi-modal approach to repurpose existing drugs against ACK1, a cancer target significantly over expressed in breast and prostate cancers during their progression ([Phatak and Zhang, 2013](#)). They used structure-based screening followed by a pharmacophore-guided method to select a set of drugs as potential ACK1 inhibitors. To compensate for the limitations of the docking methods employed in their study, they employed different metrics based on chemical similarity, genomic similarity and drug-target bipartite graph to evaluate if complementary results can be obtained.

Since each screening method described above has its own field of applicability, drawbacks and limitations, integration of multiple methods described above may be beneficial for drug repurposing.

Challenges and Possible Solutions

Despite the potential benefits of drug repurposing, the pharmaceutical industry is less interested to find new uses of old drugs. One of the major barriers that hinders the development of repurposed drugs is lack of data exclusivity ([Kato *et al.*, 2015](#)). The period of

data exclusivity for repurposed drugs is often insufficient for the pharmaceutical industries for making a reasonable profit (Shineman *et al.*, 2014; Gaudry, 2011). Moreover, commercial returns for a newly formulated product depends on the coverage and reimbursement provided for it by a payer such as a third party private or government insurer (Shineman *et al.*, 2014). When a generic drug is repurposed for a new indication through a new formulation, the payer is unlikely to provide the coverage unless they accept its clinical benefit over the generic drug (Shineman *et al.*, 2014). All these reasons withhold the pharmaceutical industry from investing in drug repurposing.

To tackle these problems, non-profit organizations and governments can collaborate and take the initiatives to accelerate drug repurposing. The government can take actions to extend the period of data exclusivity for repurposed drugs (Shineman *et al.*, 2014). Additionally, the government can offer meaningful incentives to the pharmaceutical companies for drug repurposing (Kato *et al.*, 2015). Governments and non-profit organizations should work together to create awareness at the organizational and individual level to establish the notion that patient-relevant outcomes should ultimately drive the definition of value for drugs (Shineman *et al.*, 2014). Moreover, research done at the academic institutions can also be motivational for the pharmaceutical industries to repurpose old drugs. Such research can be funded and an infrastructure can be created to co-ordinate the research interests of academic institutions and the pharmaceutical companies (Beachy *et al.*, 2014).

Additionally, the interactions between drugs and their respective target(s) are influenced by the target's structure which is not static, but dynamic (Lukman *et al.*, 2014). Considering the intrinsic dynamics of the target structure, i.e., the structure of proteins where drugs can bind to and regulate them, efforts to identify existing and novel binding sites from multiple structures of the same and/or related protein structures, for example through homology modeling, molecular dynamics simulations, and clustering approaches (Sim *et al.*, 2013), have been undertaken for several targets. Examples of the targets are Bromo and Extra terminal (BET) proteins (Lukman *et al.*, 2015a), insulin receptor (Lukman *et al.*, 2015b), and insulin-degrading enzyme (Lukman, 2016). Identification of binding sites are critical for docking and ensemble docking (Lionta *et al.*, 2014) of known drugs that was previously unknown for affecting a particular target of interest, yet have minimal toxicity.

Acknowledgements

We thank Al Jalila Foundation (AJF201507) for support. M.N. Nguyen would like to thank A*STAR Joint Council Office (JCO) Career Development Award [15302FG145] for support.

See also: Biomolecular Structures: Prediction, Identification and Analyses. Comparative Epigenomics. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Population Analysis of Pharmacogenetic Polymorphisms. Structure-Based Drug Design Workflow. Study of The Variability of The Native Protein Structure

References

- Abagyan, R., Totrov, M., Kuznetsov, D., 1994. ICM? A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. *J. Comput. Chem.* 15, 488–506. <https://doi.org/10.1002/jcc.540150503>.
- Acharya, C., Coop, A., Polli, J.E., MacKerell, A.D., 2011. Recent advances in ligand-based drug design: Relevance and utility of the conformationally sampled pharmacophore approach. *Curr. Comput. Aided-Drug Des.* 7, 10–22. Available at: <https://doi.org/10.2174/157340911793743547>.
- Allen, W.J., Balius, T.E., Mukherjee, S., *et al.*, 2015. DOCK 6: Impact of new features and current docking performance. *J. Comput. Chem.* 36, 1132–1156. <https://doi.org/10.1002/jcc.23905>.
- Amari, S., Aizawa, M., Zhang, J., *et al.*, 2006. VISCANA: Visualized cluster analysis of protein — Ligand interaction based on the ab initio fragment molecular orbital method for virtual ligand screening. *J. Chem. Inf. Model* 46, 221–230. Available at: <https://doi.org/10.1021/ci050262q>.
- Anighoro, A., Bajorath, J., Rastelli, G., 2014. Polypharmacology: Challenges and opportunities in drug discovery: Miniperspective. *J. Med. Chem.* 57, 7874–7887. Available at: <https://doi.org/10.1021/jm5006463>.
- Anighoro, A., Stumpfe, D., Heikamp, K., *et al.*, 2015. Computational polypharmacology analysis of the heat shock protein 90 interactome. *J. Chem. Inf. Model* 55, 676–686. Available at: [10.1021/ci5006959](https://doi.org/10.1021/ci5006959).
- Bajorath, J., 2001. Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput. Sci.* 41, 233–245. Available at: <https://doi.org/10.1021/jm5006463>.
- Bajorath, J., 2011. *Chemoinformatics and Computational Chemical Biology*. New York: Humana Press.
- Beachy, S.H., Johnson, S.G., Olson, S., Berger, A.C., Institute of Medicine(U.S.), 2014. *Drug Repurposing and Repositioning: Workshop Summary*. Washington, D.C.: The National Academies Press.
- Bi, Y., Might, M., Vankayalapati, H., Kuberan, B., 2017. Repurposing of proton pump inhibitors as first identified small molecule inhibitors of endo - β - N -acetylglucosaminidase (ENGase) for the treatment of NGLY1 deficiency, a rare genetic disease. *Bioorg. Med. Chem. Lett.* 27, 2962–2966. Available at: <https://doi.org/10.1016/j.bmcl.2017.05.010>.
- Bologa, C.G., Revankar, C.M., Young, S.M., *et al.*, 2006. Virtual and biomolecular screening converge on a selective agonist for GPR30. *Nat. Chem. Biol.* 2, 207–212. Available at: <https://doi.org/10.1038/nchembio775>.
- Borisy, A.A., Elliott, P.J., Hurst, N.W., *et al.*, 2003. Systematic discovery of multicomponent therapeutics. *Proc. Natl. Acad. Sci. USA* 100, 7977–7982. Available at: <https://doi.org/10.1073/pnas.1337088100>.
- Brazma, A., Parkinson, H., Sarkans, U., *et al.*, 2003. ArrayExpress — A public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* 31, 68–71.
- Brown, A.S., Patel, C.J., 2017. A standard database for drug repositioning. *Sci. Data* 4, 170029. Available at: <https://doi.org/10.1038/sdata.2017.29>.

- Campillos, M., Kuhn, M., Gavin, A.-C., Jensen, L.J., Bork, P., 2008. Drug target identification using side-effect similarity. *Science* 321, 263–266. Available at: <https://doi.org/10.1126/science.1158140>.
- Carrella, D., Napolitano, F., Rispoli, R., *et al.*, 2014. Mantra 2.0: An online collaborative resource for drug mode of action and repurposing by network analysis. *Bioinform. Oxf. Engl.* 30, 1787–1788. Available at: <https://doi.org/10.1093/bioinformatics/btu058>.
- Cereto-Massagué, A., Ojeda, M.J., Valls, C., *et al.*, 2015. Molecular fingerprint similarity search in virtual screening. *Methods* 71, 58–63. Available at: <https://doi.org/10.1016/j.ymeth.2014.08.005>.
- Chan, D.S.-H., Yang, H., Kwan, M.H.-T., *et al.*, 2011. Structure-based optimization of FDA-approved drug methylene blue as a c-myc G-quadruplex DNA stabilizer. *Biochimie* 93, 1055–1064. Available at: <https://doi.org/10.1016/j.biochi.2011.02.013>.
- Cheeseright, T., Mackey, M., Rose, S., Vinter, A., 2006. Molecular field extrema as descriptors of biological activity: Definition and validation. *J. Chem. Inf. Model.* 46, 665–676. Available at: <https://doi.org/10.1021/ci050357s>.
- Cheng, F., Liu, C., Jiang, J., *et al.*, 2012a. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLOS Comput. Biol.* 8, e1002503. Available at: <https://doi.org/10.1371/journal.pcbi.1002503>.
- Cheng, T., Li, Q., Zhou, Z., Wang, Y., Bryant, S.H., 2012b. Structure-based virtual screening for drug discovery: a problem-centric review. *AAPS J.* 14, 133–141. Available at: <https://doi.org/10.1224/s12248-012-9322-0>.
- Corbeil, C.R., Williams, C.I., Labute, P., 2012. Variability in docking success rates due to dataset preparation. *J. Comput. Aided Mol. Des.* 26, 775–786. <https://doi.org/10.1007/s10822-012-9570-1>.
- Crisan, L., Avram, S., Pacureanu, L., 2017. Pharmacophore-based screening and drug repurposing exemplified on glycogen synthase kinase-3 inhibitors. *Mol. Divers.* 21, 385–405. Available at: <https://doi.org/10.1007/s11030-016-9724-5>.
- Deng, Z., Chuaqui, C., Singh, J., 2004. Structural Interaction Fingerprint (SIF): A novel method for analyzing three-dimensional protein – Ligand binding interactions. *J. Med. Chem.* 47, 337–344. Available at: <https://doi.org/10.1021/jm030331x>.
- Duan, Q., Flynn, C., Niepel, M., *et al.*, 2014. LINCS canvas browser: Interactive web app to query, browse and interrogate LINCS L1000 gene expression signatures. *Nucleic Acids Res.* 42, W449–W460. Available at: <https://doi.org/10.1093/nar/gku476>.
- Dudley, J.T., Deshpande, T., Butte, A.J., 2011a. Exploiting drug-disease relationships for computational drug repositioning. *Brief. Bioinform.* 12, 303–311. Available at: <https://doi.org/10.1093/bib/bbr013>.
- Dudley, J.T., Sirota, M., Shenoy, M., *et al.*, 2011b. Computational repositioning of the anticonvulsant topiramate for inflammatory bowel disease. *Sci. Transl. Med.* 3, 96ra76. Available at: <https://doi.org/10.1126/scitranslmed.3002648>.
- Edgar, R., Domrachev, M., Lash, A.E., 2002. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* 30, 207–210.
- Ewing, T.J.A., Makino, S., Skillman, A.G., Kuntz, I.D., 2001. DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases. *J. Comput. Aided Mol. Des.* 15, 411–428. Available at: <https://doi.org/10.1023/A:1011115820450>.
- Forli, S., Huey, R., Pique, M.E., *et al.*, 2016. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nat. Protoc.* 11, 905–919. <https://doi.org/10.1038/nprot.2016.051>.
- Friesner, R.A., Banks, J.L., Murphy, R.B., *et al.*, 2004. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.*, 47, pp. 1739–1749. <https://doi.org/10.1021/jm0306430>.
- Fu, C., Jin, G., Gao, J., *et al.*, 2013. DrugMap central: An on-line query and visualization tool to facilitate drug repositioning studies. *Bioinformatics* 29, 1834–1836. Available at: <https://doi.org/10.1093/bioinformatics/btt279>.
- Fukunishi, Y., Nakamura, H., 2009. A similarity search using molecular topological graphs. *J. Biomed. Biotechnol.* 2009, 1–8. Available at: <https://doi.org/10.1155/2009/231780>.
- Gaudry, K.S., 2011. Evergreening: A common practice to protect new drugs. *Nat. Biotechnol.* 29, 876–878. Available at: <https://doi.org/10.1038/nbt.1993>.
- Glaab, E., 2016. Building a virtual ligand screening pipeline using free software: A survey. *Brief. Bioinform.* 17, 352–366. Available at: <https://doi.org/10.1093/bib/bbv037>.
- Gloeckner, C., Garner, A.L., Mersha, F., *et al.*, 2010. Repositioning of an existing drug for the neglected tropical disease Onchocerciasis. *Proc. Natl. Acad. Sci.* 107, 3424–3429. Available at: <https://doi.org/10.1073/pnas.0915125107>.
- Goodford, P.J., 1985. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *J. Med. Chem.* 28, 849–857.
- Gozalbes, R., Simon, L., Froloff, N., *et al.*, 2008. Development and experimental validation of a docking strategy for the generation of kinase-targeted libraries. *J. Med. Chem.* 51, 3124–3132. Available at: <https://doi.org/10.1021/jm701367r>.
- Grant, B.J., Lukman, S., Hocker, H.J., *et al.*, 2011. Novel allosteric sites on Ras for lead generation. *PLOS ONE* 6, e25711. Available at: <https://doi.org/10.1371/journal.pone.0025711>.
- Han, L., Li, K., Jin, C., *et al.*, 2017. Human enterovirus 71 protein interaction network prompts antiviral drug repositioning. *Sci. Rep.* 7, 43143. Available at: <https://doi.org/10.1038/srep43143>.
- Helguera, A., Combes, R., Gonzalez, M., Cordeiro, M.N., 2008. Applications of 2D descriptors in drug design: A DRAGON Tale. *Curr. Top. Med. Chem.* 8, 1628–1655. Available at: <https://doi.org/10.2174/156802608786786598>.
- Hong, H., Xie, Q., Ge, W., *et al.*, 2008. Mold², molecular descriptors from 2D structures for chemoinformatics and toxicoinformatics. *J. Chem. Inf. Model* 48, 1337–1344. Available at: <https://doi.org/10.1021/ci800038f>.
- Huang, Y.-F., Yeh, H.-Y., Soo, V.-W., 2013. Inferring drug-disease associations from integration of chemical, genomic and phenotype data using network propagation. *BMC Med. Genom.* 6, S4. Available at: <https://doi.org/10.1186/1755-8794-6-S3-S4>.
- Iorio, F., Bosotti, R., Scacheri, E., *et al.*, 2010. Discovery of drug mode of action and drug repositioning from transcriptional responses. *Proc. Natl. Acad. Sci.* 107, 14621–14626. Available at: <https://doi.org/10.1073/pnas.1000138107>.
- Iorio, F., Tagliaferri, R., di Bernardo, D., 2009. Identifying network of drug mode of action by gene expression profiling. *J. Comput. Biol.* 16, 241–251. Available at: <https://doi.org/10.1089/cmb.2008.10TT>.
- Irwin, J.J., Sterling, T., Mysinger, M.M., Bolstad, E.S., Coleman, R.G., 2012. ZINC: A free tool to discover chemistry for biology. *J. Chem. Inf. Model* 52, 1757–1768. Available at: <https://doi.org/10.1021/ci3001277>.
- Jadamba, E., Shin, M., 2016. A systematic framework for drug repositioning from integrated omics and drug phenotype profiles using pathway-drug network. *BioMed Res. Int.* 2016. Available at: <https://doi.org/10.1155/2016/7147039>.
- Jain, A.N., 2003. Surflex: Fully Automatic Flexible Molecular Docking Using a Molecular Similarity-Based Search Engine. *J. Med. Chem.* 46, 499–511. <https://doi.org/10.1021/jm020406h>.
- John Fox, S., Li, J., Sing Tan, Y., *et al.*, 2016. The multifaceted roles of molecular dynamics simulations in drug discovery. *Curr. Pharm. Des.* 22, 3585–3600. Available at: <https://doi.org/10.2174/1381612822666160425120507>.
- Jones, G., Willett, P., Glen, R.C., Leach, A.R., Taylor, R., 1997. Development and validation of a genetic algorithm for flexible docking 1 Edited by F. E. Cohen. *J. Mol. Biol.* 267, 727–748. <https://doi.org/10.1006/jmbi.1996.0897>.
- Kato, S., Moulder, S.L., Ueno, N.T., *et al.*, 2015. Challenges and perspective of drug repurposing strategies in early phase clinical trials. *Oncoscience* 2, 576–580.
- Keiser, M.J., Setola, V., Irwin, J.J., *et al.*, 2009. Predicting new molecular targets for known drugs. *Nature* 462, 175–181. Available at: <https://doi.org/10.1038/nature08506>.
- Kombo, D.C., Tallapragada, K., Jain, R., *et al.*, 2013. 3D molecular descriptors important for clinical success. *J. Chem. Inf. Model* 53, 327–342. Available at: <https://doi.org/10.1021/ci300445e>.
- Konc, J., Janežič, D., 2010. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics* 26, 1160–1168. Available at: <https://doi.org/10.1093/bioinformatics/btq100>.
- Kroemer, R.T., 2007. Structure-based drug design: Docking and scoring. *Curr. Protein Pept. Sci.* 8, 312–328.

- Kuhn, M., Letunic, I., Jensen, L.J., Bork, P., 2016. The SIDER database of drugs and side effects. *Nucleic Acids Res.* 44, D1075–D1079. Available at: <https://doi.org/10.1093/nar/gkv1075>.
- Kumar, A.P., Nguyen, M.N., Verma, C., Lukman, S., 2018. Structural analysis of protein tyrosine phosphatase 1B reveals potentially druggable allosteric binding sites. *Proteins Struct. Funct. Bioinform.* 86, 301–321. Available at: <https://doi.org/10.1002/prot.25440>.
- Lamb, J., 2006. The Connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313, 1929–1935. Available at: <https://doi.org/10.1126/science.1132939>.
- Lamb, J., 2007. The Connectivity map: A new tool for biomedical research. *Nat. Rev. Cancer* 7, 54–60. Available at: <https://doi.org/10.1038/nrc2044>.
- Langer, T., Hoffmann, R.D., 2006. *Pharmacophores and Pharmacophore Searches*. Weinheim; Chichester: Wiley-VCH; John Wiley, distributor.
- Law, V., Knox, C., Djombou, Y., et al., 2014. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Res.* 42, D1091–D1097. Available at: <https://doi.org/10.1093/nar/gkt1068>.
- Lee, B.K.B., Tiong, K.H., Chang, J.K., et al., 2017. DeSigN: Connecting gene expression with therapeutics for drug repurposing and development. *BMC Genom.* 18. Available at: <https://doi.org/10.1186/s12864-016-3260-7>.
- Lee, H., Bae, T., Lee, J.-H., et al., 2012. Rational drug repositioning guided by an integrated pharmacological network of protein, disease and drug. *BMC Syst. Biol.* 6, 80. Available at: <https://doi.org/10.1186/1752-0509-6-80>.
- Lee, J.-H., Park, K.M., Han, D.-J., et al., 2015. PharmDB-K: Integrated bio-pharmacological network database for traditional Korean medicine. *PLOS ONE* 10, e0142624. Available at: <https://doi.org/10.1371/journal.pone.0142624>.
- Leung, E.L., Cao, Z.-W., Jiang, Z.-H., Zhou, H., Liu, L., 2013. Network-based drug discovery by integrating systems biology and computational technologies. *Brief. Bioinform.* 14, 491–505. Available at: <https://doi.org/10.1093/bib/bbs043>.
- Liang, S., Meroueh, S.O., Wang, G., Qiu, C., Zhou, Y., 2009. Consensus scoring for enriching near-native structures from protein-protein docking decoys. *Proteins Struct. Funct. Bioinform.* 75, 397–403. Available at: <https://doi.org/10.1002/prot.22252>.
- Lindert, S., Zhu, W., Liu, Y.-L., et al., 2013. Farnesyl Diphosphate Synthase inhibitors from in silico screening. *Chem. Biol. Drug Des.* 81, 742–748. Available at: <https://doi.org/10.1111/cbdd.12121>.
- Lin, S.-K., 2000. Pharmacophore perception, development and use in drug design. In: Osman, F. (Ed.), *Güner Molecules* 5, pp. 987–989. Available at: <https://doi.org/10.3390/50700987>.
- Lionta, E., Spyrou, G., Vassiliatis, D.K., Cournia, Z., 2014. Structure-based virtual screening for drug discovery: Principles, applications and recent advances. *Curr. Top. Med. Chem.* 14, 1923–1938. Available at: <https://doi.org/10.2174/1568026614666140929124445>.
- Lipinski, C.A., 2000. Drug-like properties and the causes of poor solubility and poor permeability. *J. Pharmacol. Toxicol. Methods* 44, 235–249.
- Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney, P.J., 2001. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* 46, 3–26.
- Li, Y., Guo, B., Xu, Z., et al., 2016. Repositioning organohalogen drugs: A case study for identification of potent B-Raf V600E inhibitors via docking and bioassay. *Sci. Rep.* 6. Available at: <https://doi.org/10.1038/srep31074>.
- Lukman, S., 2016. Novel druggable sites of insulin-degrading enzyme identified through applied structural bioinformatics analysis. In: *Proceedings of the International Conference on Computational Science, Procedia Computer Science, ICCS 2016*, 80, pp. 2292–2296. San Diego, California, USA. Available at: <https://doi.org/10.1016/j.procs.2016.05.419>.
- Lukman, S., Aung, Z., Sim, K., 2015a. Multiple structural clustering of the Bromo and Extra Terminal (BET) proteins highlights subtle differences in their structural dynamics and acetylated leucine binding pocket. *Procedia Comput. Sci.* 51, 735–744. Available at: <https://doi.org/10.1016/j.procs.2015.05.192>.
- Lukman, S., Nguyen, M.N., Sim, K., Teo, J.C.M., 2017. Discovery of Rab1 binding sites using an ensemble of clustering methods: Clustering for finding Rab1 binding sites. *Proteins Struct. Funct. Bioinform.* 85, 859–871. Available at: <https://doi.org/10.1002/prot.25254>.
- Lukman, S., Robinson, R.C., Wales, D., Verma, C.S., 2012. Conformational dynamics of capping protein and interaction partners: Simulation studies. *Proteins* 80, 1066–1077. Available at: <https://doi.org/10.1002/prot.24008>.
- Lukman, S., Safar, H.A., Lee, S.M., Sim, K., 2015b. Harnessing structural data of insulin and insulin receptor for therapeutic designs. *J. Endocrinol. Metab.* 5, 273–283. Available at: <https://doi.org/10.14740/jem302w>.
- Lukman, S., Verma, C.S., Fuentes, G., 2014. Exploiting protein intrinsic flexibility in drug design. In: Han, K., Zhang, X., Yang, M. (Eds.), *Protein Conformational Dynamics*. Springer International Publishing, Cham, pp. 245–269. Available at: https://doi.org/10.1007/978-3-319-02970-2_11.
- Luo, H., Chen, J., Shi, L., et al., 2011. DRAR-CPI: A server for identifying drug repositioning potential and adverse drug reactions via the chemical – Protein interactome. *Nucleic Acids Res.* 39, W492–W498. Available at: <https://doi.org/10.1093/nar/gkr299>.
- Luo, H., Zhang, P., Cao, X.H., et al., 2016. DPDR-CPI, a server that predicts drug positioning and drug repositioning via chemical-protein interactome. *Sci. Rep.* 6. Available at: <https://doi.org/10.1038/srep35996>.
- Melville, J., Burke, E., Hirst, J., 2009. Machine learning in virtual screening. *Comb. Chem. High Throughput Screen.* 12, 332–343. Available at: <https://doi.org/10.2174/138620709788167980>.
- Moosavinasab, S., Patterson, J., Strouse, R., et al., 2016. 'RE: fine drugs': An interactive dashboard to access drug repurposing opportunities. *Database* 2016, baw083. Available at: <https://doi.org/10.1093/database/baw083>.
- Morris, G.M., Huey, R., Olson, A.J., 2008. Using AutoDock for ligand-receptor docking. *Curr. Protoc. Bioinform.* Chapter 8, Unit 8.14 <https://doi.org/10.1002/0471250953.bi0814s24>.
- Mulas, F., Li, A., Sherr, D.H., Monti, S., 2017. Network-based analysis of transcriptional profiles from chemical perturbations experiments. *BMC Bioinform.* 18. Available at: <https://doi.org/10.1186/s12859-017-1536-9>.
- Napolitano, F., Zhao, Y., Moreira, V.M., et al., 2013. Drug repositioning: A machine-learning approach through data integration. *J. Cheminform.* 5, 30. Available at: <https://doi.org/10.1186/1758-2946-5-30>.
- NCBI Resource Coordinators, 2013. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 41, D8–D20. Available at: <https://doi.org/10.1093/nar/gks1189>.
- Neves, M.A.C., Totrov, M., Abagyan, R., 2012. Docking and scoring with ICM: the benchmarking results and strategies for improvement. *J. Comput. Aided Mol. Des.* 26, 675–686. <https://doi.org/10.1007/s10822-012-9547-0>.
- Nguyen, M.N., Madhusudhan, M.S., 2011. Biological insights from topology independent comparison of protein 3D structures. *Nucleic Acids Res.* 39, e94. Available at: <https://doi.org/10.1093/nar/gkr348>.
- Nguyen, M.N., Pradhan, M.R., Verma, C., Zhong, P., 2017a. The interfacial character of antibody paratopes: Analysis of antibody – Antigen structures. *Bioinformatics* 33, 2971–2976. Available at: <https://doi.org/10.1093/bioinformatics/btx389>.
- Nguyen, M.N., Sim, A.Y.L., Wan, Y., Madhusudhan, M.S., Verma, C., 2017b. Topology independent comparison of RNA 3D structures using the CLICK algorithm. *Nucleic Acids Res.* 45, e5. Available at: <https://doi.org/10.1093/nar/gkw819>.
- Nguyen, M.N., Tan, K.P., Madhusudhan, M.S., 2011. CLICK – Topology-independent comparison of biomolecular 3D structures. *Nucleic Acids Res.* 39, W24–W28. Available at: <https://doi.org/10.1093/nar/gkr393>.
- Nguyen, M.N., Verma, C., 2015. Rclick: A web server for comparison of RNA 3D structures. *Bioinformatics* 31, 966–968. Available at: <https://doi.org/10.1093/bioinformatics/btu752>.
- Novick, P.A., Ortiz, O.F., Poelman, J., Abdulhay, A.Y., Pande, V.S., 2013. SWEETLEAD: An in silico database of approved drugs, regulated chemicals, and herbal isolates for computer-aided drug discovery. *PLOS ONE* 8, e79568. Available at: <https://doi.org/10.1371/journal.pone.0079568>.

- Nussinov, R., Tsai, C.-J., 2012. The different ways through which specificity works in orthosteric and allosteric drugs. *Curr. Drug Metab.* 18, 1311–1316. Available at: <https://doi.org/10.2174/138920012799362855>.
- Oprea, T.I., Bauman, J.E., Bologa, C.G., *et al.*, 2011. Drug repurposing from an academic perspective. *Drug Discov. Today Ther. Strateg.* 8, 61–69. <https://doi.org/10.1016/j.ddstr.2011.10.002>.
- Palos, I., Lara-Ramirez, E.E., Lopez-Cedillo, J.C., *et al.*, 2017. Repositioning FDA Drugs as Potential Cruzain Inhibitors from *Trypanosoma cruzi*: Virtual Screening, In Vitro and In Vivo Studies. *Molecules* 22, 1015. <https://doi.org/10.3390/molecules22061015>.
- Pan, Y., Wang, Y., Bryant, S.H., 2013. Pharmacophore and 3D-QSAR characterization of 6-Arylquinazolin-4-amines as Cdc2-like Kinase 4 (Cdk4) and dual specificity tyrosine-phosphorylation-regulated kinase 1A (Dyrk1A) inhibitors. *J. Chem. Inf. Model* 53, 938–947. Available at: <https://doi.org/10.1021/ci300625c>.
- Paolini, G.V., Shapland, R.H.B., van Hoorn, W.P., Mason, J.S., Hopkins, A.L., 2006. Global mapping of pharmacological space. *Nat. Biotechnol.* 24, 805–815. Available at: <https://doi.org/10.1038/nbt1228>.
- Parkkinen, J.A., Kaski, S., 2014. Probabilistic drug connectivity mapping. *BMC Bioinform.* 15, 113. Available at: <https://doi.org/10.1186/1471-2105-15-113>.
- Patel, Y., Gillet, V.J., Bravi, G., Leach, A.R., 2002. A comparison of the pharmacophore identification programs: Catalyst, DISCO and GASP. *J. Comput. Aided Mol. Des.* 16, 653–681. Available at: <https://doi.org/10.1023/A:1021954728347>.
- Paz, O.S., de Jesus Pinheiro, M., do Espírito Santo, R.F., Villarreal, C.F., Castilho, M.S., 2017. Nanomolar anti-sickling compounds identified by ligand-based pharmacophore approach. *Eur. J. Med. Chem.* 136, 487–496. Available at: <https://doi.org/10.1016/j.ejmech.2017.05.035>.
- Phatak, S.S., Zhang, S., 2013. A novel multi-modal drug repurposing approach for identification of potent ack1 inhibitors. *Pac. Symp. Biocomput.* 29–40.
- Pihan, E., Colliandre, L., Guichou, J.-F., Douguet, D., 2012. e-Drug3D: 3D structure collections dedicated to drug repurposing and fragment-based drug design. *Bioinformatics* 28, 1540–1541. Available at: <https://doi.org/10.1093/bioinformatics/bts186>.
- Pratanwanich, N., Lió, P., 2014. Pathway-based Bayesian inference of drug – Disease interactions. *Mol. BioSyst.* 10, 1538–1548. Available at: <https://doi.org/10.1039/C4MB00014E>.
- Radusky, L., Ruiz-Carmona, S., Modenutti, C., *et al.*, 2017. LigQ: A webserver to select and prepare ligands for virtual screening. *J. Chem. Inf. Model.* 57, 1741–1746. Available at: <https://doi.org/10.1021/acs.jcim.7b00241>.
- Rapp, C.S., Schonbrun, C., Jacobson, M.P., Kalyanaraman, C., Huang, N., 2009. Automated site preparation in physics-based rescoring of receptor ligand complexes. *Proteins Struct. Funct. Bioinform.* 77, 52–61. Available at: <https://doi.org/10.1002/prot.22415>.
- Rarey, M., Kramer, B., Lengauer, T., Klebe, G., 1996. A Fast Flexible Docking Method using an Incremental Construction Algorithm. *J. Mol. Biol.* 261, 470–489. <https://doi.org/10.1006/jmbi.1996.0477>.
- Reaume, A.G., 2011. Drug repurposing through nonhypothesis driven phenotypic screening. *Drug Discov. Today Ther. Strateg.* 8, 85–88. Available at: <https://doi.org/10.1016/j.ddstr.2011.09.007>.
- Reddy, A.S., Zhang, S., 2013. Polypharmacology: Drug discovery for the future. *Expert Rev. Clin. Pharmacol.* 6, 41–47. Available at: <https://doi.org/10.1586/ecp.12.74>.
- Rose, P.W., Bi, C., Bluhm, W.F., *et al.*, 2013. The RCSB protein data bank: New resources for research and education. *Nucleic Acids Res.* 41, D475–D482. Available at: <https://doi.org/10.1093/nar/gks1200>.
- Ruiz-Carmona, S., Alvarez-Garcia, D., Foloppe, N., *et al.*, 2014. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLOS Comput. Biol.* 10, e1003571. <https://doi.org/10.1371/journal.pcbi.1003571>.
- Sage, C., Wang, R., Jones, G., 2011. G-protein coupled receptors virtual screening using genetic algorithm focused chemical space. *J. Chem. Inf. Model* 51, 1754–1761. Available at: <https://doi.org/10.1021/ci200043z>.
- Sahoo, M., Jena, L., Daf, S., Kumar, S., 2016. Virtual screening for potential inhibitors of NS3 protein of zika virus. *Genom. Inform.* 14, 104. Available at: <https://doi.org/10.5808/GI.2016.14.3.104>.
- Schuffenhauer, A., Gillet, V.J., Willett, P., 2000. Similarity searching in files of three-dimensional chemical structures: Analysis of the BIOSTER database using two-dimensional fingerprints and molecular field descriptors. *J. Chem. Inf. Comput. Sci.* 40, 295–307. Available at: <https://doi.org/10.1021/ci990263g>.
- Schwartz, J., Awale, M., Raymond, J.-L., 2013. SMItip (SMILES fingerprint) chemical space for virtual screening and visualization of large databases of organic molecules. *J. Chem. Inf. Model* 53, 1979–1989. Available at: <https://doi.org/10.1021/ci400206h>.
- Shameer, K., Glucksberg, B.S., Hodos, R., *et al.*, 2017. Systematic analyses of drugs and disease indications in RepurposeDB reveal pharmacological, biological and epidemiological factors influencing drug repositioning. *Brief. Bioinform.* Available at: <https://doi.org/10.1093/bib/bbw136>.
- Shineman, D.W., Alam, J., Anderson, M., *et al.*, 2014. Overcoming obstacles to repurposing for neurodegenerative disease. *Ann. Clin. Transl. Neurol.* 1, 512–518. Available at: <https://doi.org/10.1002/acn3.76>.
- Shin, W.-H., Zhu, X., Bures, M., Kihara, D., 2015. Three-dimensional compound comparison methods and their application in drug discovery. *Molecules* 20, 12841–12862. Available at: <https://doi.org/10.3390/molecules200712841>.
- Shoichet, B.K., 2004. Virtual screening of chemical libraries. *Nature* 432, 862–865. Available at: <https://doi.org/10.1038/nature03197>.
- Shulman-Peleg, A., Shatsky, M., Nussinov, R., Wolfson, H.J., 2007. Spatial chemical conservation of hot spot interactions in protein-protein complexes. *BMC Biol.* 5, 43. Available at: <https://doi.org/10.1186/1741-7007-5-43>.
- Silberberg, Y., Gottlieb, A., Kupiec, M., Rupp, E., Sharan, R., 2012. Large-scale elucidation of drug response pathways in humans. *J. Comput. Biol.* 19, 163–174. Available at: <https://doi.org/10.1089/cmb.2011.0264>.
- Sim, K., Yap, G.-E., Hardoon, D.R., *et al.*, 2013. Centroid-based actionable 3D subspace clustering. *IEEE Trans. Knowl. Data Eng.* 25, 1213–1226.
- Simon, Z., Peragovics, A., Vigh-Smeller, M., *et al.*, 2012. Drug effect prediction by polypharmacology-based interaction profiling. *J. Chem. Inf. Model* 52, 134–145. Available at: <https://doi.org/10.1021/ci2002022>.
- Sirota, M., Dudley, J.T., Kim, J., *et al.*, 2011. Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Sci. Transl. Med.* 3, 96ra77. Available at: <https://doi.org/10.1126/scitranslmed.3001318>.
- Sliwoski, G., Kothiwale, S., Meiler, J., Lowe, E.W., 2014. Computational methods in drug discovery. *Pharmacol. Rev.* 66, 334–395. Available at: <https://doi.org/10.1124/pr.112.007336>.
- Sottriffer, C., 2011. *Virtual Screening: Principles, Challenges, and Practical Guidelines, Methods and Principles in Medicinal Chemistry*. Weinheim, Germany: Wiley-VCH.
- Stroganov, O.V., Novikov, F.N., Stroylov, V.S., Kulkov, V., Chilov, G.G., 2008. Lead finder: An approach to improve accuracy of protein-ligand docking, binding energy estimation, and virtual screening. *J. Chem. Inf. Model.* 48, 2371–2385. <https://doi.org/10.1021/ci800166p>.
- Sukumar, N., Das, S., 2011. Current trends in virtual high throughput screening using ligand-based and structure-based methods. *Comb. Chem. High Throughput Screen.* 14, 872–888.
- Sun, H., 2008. Pharmacophore-based virtual screening. *Curr. Med. Chem.* 15, 1018–1024.
- Sun, H., 2016. *A practical guide to rational drug design*.
- ten Brink, T., Exner, T.E., 2010. pK(a) based protonation states and microspecies for protein-ligand docking. *J. Comput. Aided Mol. Des.* 24, 935–942. Available at: <https://doi.org/10.1007/s10822-010-9385-x>.
- Trott, O., Olson, A.J., 2009. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* 31 (2), 455–461. Available at: <https://doi.org/10.1002/jcc.21334>.
- Tusher, V.G., Tibshirani, R., Chu, G., 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA* 98, 5116–5121. Available at: <https://doi.org/10.1073/pnas.091062498>.
- UniProt Consortium, 2013. Update on activities at the Universal Protein Resource (UniProt) in 2013. *Nucleic Acids Res.* 41, D43–D47. Available at: <https://doi.org/10.1093/nar/gks1068>.

- Vasudevan, S.R., Moore, J.B., Schymura, Y., Churchill, G.C., 2012. Shape-based reprofiling of FDA-approved drugs for the H₁ histamine receptor. *J. Med. Chem.* 55, 7054–7060. Available at: <https://doi.org/10.1021/jm300671m>.
- von Eichborn, J., Murgueitio, M.S., Dunkel, M., *et al.*, 2011. PROMISCUOUS: A database for network-based drug-repositioning. *Nucleic Acids Res.* 39, D1060–D1066. Available at: <https://doi.org/10.1093/nar/gkq1037>.
- Wang, R., Gao, Y., Lai, L., 2000. LigBuilder: A multi-purpose program for structure-based drug design. *J. Mol. Model.* 6, 498–516. Available at: <https://doi.org/10.1007/s0089400060498>.
- Wieggers, T.C., Davis, A., Cohen, K.B., Hirschman, L., Mattingly, C.J., 2009. Text mining and manual curation of chemical-gene-disease networks for the Comparative Toxicogenomics Database (CTD). *BMC Bioinform.* 10, 326. Available at: <https://doi.org/10.1186/1471-2105-10-326>.
- Wu, Z., Wang, Y., Chen, L., 2013. Network-based drug repositioning. *Mol. Biosyst.* 9, 1268. Available at: <https://doi.org/10.1039/c3mb25382a>.
- Yang, J., Li, Z., Fan, X., Cheng, Y., 2014. Drug – Disease association and drug-repositioning predictions in complex diseases using causal inference – Probabilistic matrix factorization. *J. Chem. Inf. Model.* 54, 2562–2569. Available at: <https://doi.org/10.1021/ci500340n>.
- Yan, X., Liao, C., Liu, Z., *et al.*, 2016. Chemical structure similarity search for ligand-based virtual screening: Methods and computational resources. *Curr. Drug Targets* 17, 1580–1585. Available at: <https://doi.org/10.2174/1389450116666151102095555>.
- Yu, J., Putcha, P., Silva, J.M., 2015. Recovering drug-induced apoptosis subnetwork from connectivity Map data. *BioMed Res. Int.* 2015, 708563. Available at: <https://doi.org/10.1155/2015/708563>.

Revelant Websites

- <https://www.ebi.ac.uk/arrayexpress/>
ArrayExpress.
- <https://www.ebi.ac.uk/chembl/>
ChEMBL.
- <https://www.broadinstitute.org/cmap/>
Connectivity Map.
- <http://chemoinfo.ipmc.cnrs.fr/e-drug3d.html>
e-LEA3D: ChemInformatic Tools and Databases.
- <http://www.ncbi.nlm.nih.gov/geo/>
Gene Expression Omnibus.
- <http://www.lincsccloud.org/>
lincsccloud.
- <http://ligq.qb.fcen.uba.ar/>
LigQ.
- <https://cpi.bio-x.cn/dpdr/>
Predict drug positioning and drug repositioning - DPDR-CPI.
- <https://cpi.bio-x.cn/drdr/>
Predict drug repositioning and adverse drug reaction - DRAR-CPI.
- <http://bioinformatics.charite.de/promiscuous>
PROMISCUOUS.
- <http://www-stat.stanford.edu/~tibs/SAM>
SAM: Significance Analysis of Microarrays.

Transcriptome Analysis

Anuj Srivastava, Joshy George, and Radha KM Karuturi, The Jackson Laboratory, CT, United States

© 2019 Elsevier Inc. All rights reserved.

Introduction

Transcriptome is a collection of all RNA molecules in one cell or a population of cells. RNA has long been at the center of molecular biology and its importance had been established in a series of paradigm-shifting discoveries over the last 60–70 years. Prior to 1940s, it was considered that proteins carried out both genetic information and enzymatic catalytic function of cells. In the 1950s, Crick published the central dogma that showed RNA served as an intermediary molecule during the transfer of genetic information of proteins (Crick, 1958). For many years, after the establishment of the central dogma, it was believed that RNA molecules come in 3 flavors (rRNA, tRNA, and mRNA) and their primary function involved protein syntheses. Later in the 1980s, several other types of RNAs including uridine (U)-rich U RNAs were discovered; in the same time period, Cech and Altman, studying the splicing in *Tetrahymena thermophile* and bacterial RNase P complex, respectively, concluded that RNA had catalytic functions as ribozymes (Cech *et al.*, 1983). In the 1990s, Ambros and colleagues (Lee *et al.*, 1993) showed the first evidence of RNA mediated regulation (by microRNA or miRNA) in *Caenorhabditis elegans* development and thus establishing three primary roles of RNAs: (i) Being the genetic material (ii) act as ribozymes (iii) and regulate other macromolecular processes. Hence, both mRNAs or messenger RNAs (protein-coding RNAs) and non-coding RNAs or ncRNAs (do not code proteins) are of interest in a variety of studies. The ncRNAs of particular interest are miRNAs which are short ncRNAs that regulate the lifetime of mRNAs; and, lncRNAs or long non-coding RNAs which are ≥ 200 nucleotides long, regulate the gene expression and also take part in a broad range of cellular & developmental processes.

The type and quantity of genes transcribed is dependent on the cell-type and its environment, and is tightly regulated. Disruptions to this regulatory process is often the cause of several diseases. Being intermediary between DNA and proteome as well as being molecules of regulatory role, understanding the variations in the transcriptome of a cell in response to the changes in the function and environment of the cell is critical to understanding the regulatory mechanisms pertinent to the mechanistic understanding of the development and disease. Thus, the composition of transcriptome is often used as a proxy for cellular functional status.

Our understanding of the transcriptome has been largely enabled by the advancement in technology. In the late 90's, microarrays could be used to profile the transcriptome of a population of cells. This technology essentially was used to measure the average expression level for each gene across a large population of input cells. However, microarrays detect only known RNAs, so they can't be used for discovery. In addition, background hybridization interferes with detection of low-level expression and probe saturation causes variation in highly expressed genes undetected. Recent advances in sequencing technology have made it possible to quantify the transcriptome of a population of cells by sequencing (RNA-seq or bulk RNA-seq). The number of reads mapping to a gene is a measure of the transcriptional level of that gene. RNA-seq, while being more precise than array based methods, it also offers simultaneous transcript discovery and quantification. Bulk RNA-sequencing enabled us to compare the transcriptome across different samples and had significant input into our understanding of several disease processes including cancer. In 2009 (Tang *et al.*, 2009), a new method was proposed to profile transcriptome from an individual cell. This technology is now referred to as single-cell RNA-seq (scRNA-seq) and has improved our understanding of the heterogeneity present in the otherwise homogeneous tissues. scRNA-seq allows to study new biological questions in which cell-specific changes in transcriptome are important and these include, cell type identification, heterogeneity of cell responses, stochasticity of gene expression and inference of gene regulatory networks across the cells.

Thus, in this article, we describe the guidelines for experimental design, data quality control and computational methods that can be used to derive biological insights from bulk RNA-seq data for mRNA, miRNA, and lncRNA; and scRNA-seq data. The procedures described in this article have been applied in multitude of studies such as Chow *et al.* (2017); George *et al.* (2016); Ucar *et al.* (2017).

Experimental Design, Data Processing & Quality Assessment

Experimental design is very important, though often neglected, step of the project. A well-designed experiment will have sufficient power to address the biological question of interest. A good experimental design in RNA-seq experiment considers library types (single end/paired end), number of reads or sequencing depth and number of replicates per condition. If the goal of the experiment does not involve the novel isoform discovery then single end data should serve the purpose. If the project requires only quantification of gene expression, as suggested by the ENCODE (encyclopedia of DNA elements) study, ~ 36 million mapped reads would be sufficient to accurately quantify the abundance of 80% of genes (Sims *et al.*, 2014). It's also been noted that to detect the novel and rare isoforms, ~ 200 million paired-end reads will be required in human cells.

Adding control sequences: It's important to consider including exogenous reference transcripts ('spike-ins') (Chen *et al.*, 2015) that are useful both for quality control and for library-size normalization. The *External RNA control consortium (ERCC) spike-in*

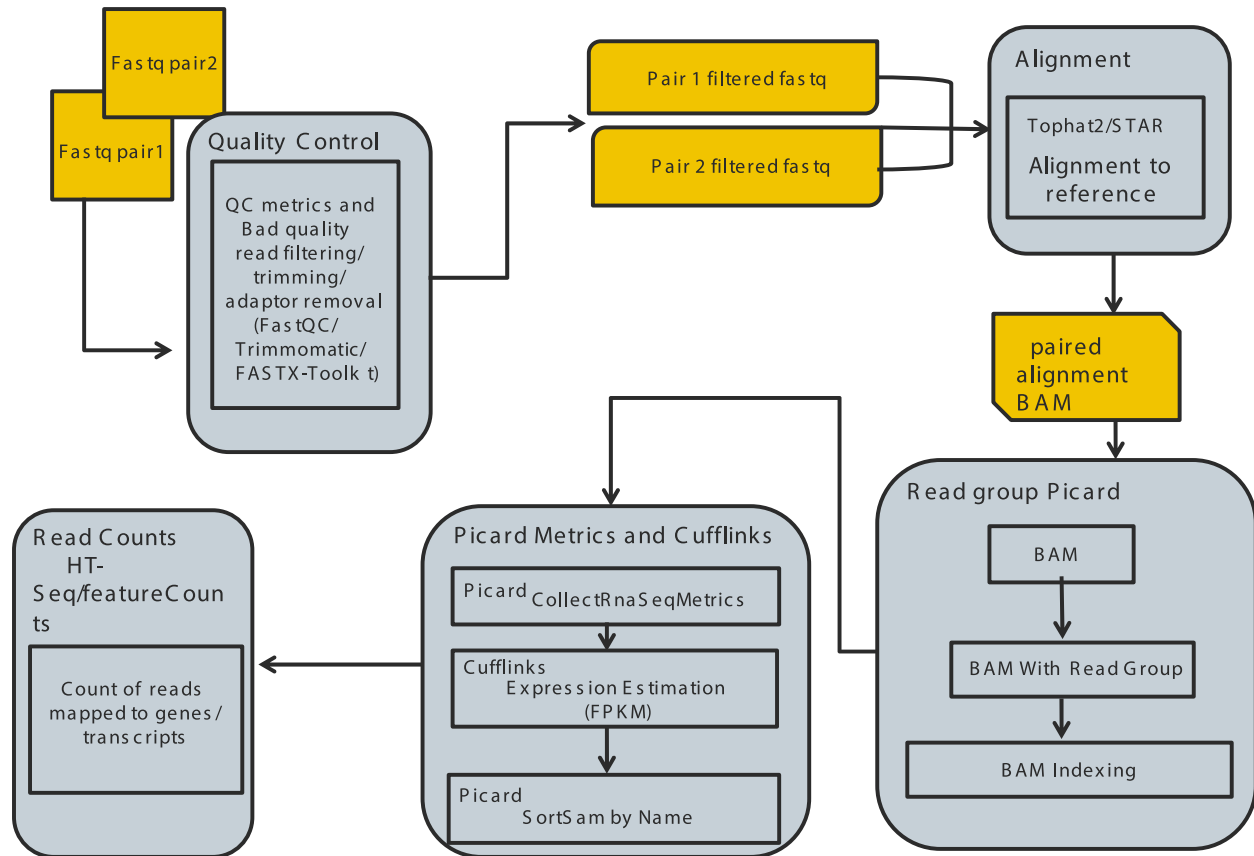


Fig. 1 Schematic for mRNA-seq quantification pipeline.

control mix consists of pre-formulated set of 96 polyadenylated transcripts with varying GC content and length can be used for this purpose.

Sample size estimation: Besides library preparation, estimation of the number of samples per condition is also important for a reliable identification of transcripts relevant to the study. Statistical power to detect differentially expressed genes (DEGs) varies with effect size, sequencing depth and the number of samples per condition. Analysis of data sets from several studies indicates the need for negative binomial distribution to account for the over-dispersion mainly caused by natural biological variation present in RNA-seq studies (Anders and Huber, 2010; Robinson and Oshlack, 2010). The complexity of this distribution makes it difficult to arrive at an analytical solution for power analysis and sample size estimation (Pham and Jimenez, 2012). Other issues that complicate the power estimation include multiple hypotheses correction, techniques used to estimate dispersion, and normalization methods used. Therefore, numerical methods such as Monte Carlo simulations have often been employed to analyze and estimate the sample size required for RNA-seq studies (Ching et al., 2014). However, with fixed cost and limited resources for the experiment, reducing the number of reads per sample while increasing the number of samples will yield better results for DEG analysis. As a general rule, at least three samples per condition is recommended.

Batch Effect is defined as a non-biological source of the signal that is introduced into the data and that can confound with the biological signal. The batch effect is typically consistent across transcripts, exons or genes among samples in a batch and so may lead to gross errors in the calculation of statistical significance, estimation of effect sizes and other statistical measures. Several studies have noticed batch effects due to reagent batch and sequencing dates. Therefore, it is important to design RNA-seq experiments to minimize the effect of technical noise due to batch effects. However, batch effects cannot be completely eliminated due to the constraints of sample availability or other logistical limitations. In such situations, randomization of samples across library preparation batches and lanes is highly recommended to avoid technical factors completely confounding with the experimental conditions. For example, an experiment with all treatment samples are sequenced in one batch while the control samples are sequenced in another batch is difficult to analyze. A few examples of desirable and undesirable experimental designs are depicted in Fig. 2:

The processing of transcriptomic data (schematics are provided in Figs. 1, 10 and 11), RNA-seq data precisely, can be divided into 3 major steps.

- (1) **Quality control (QC):** Quality control (QC) is a critical step in both bulk and single-cell RNA-seq data analysis. For bulk RNA-seq, the QC steps include assessment of RNA quality and samples with inadequate quality will not be sequenced. Quality of raw sequencing data will be assessed based on the number of reads and the read quality scores. Examples for good and poor-

Batch	1	1	1	1	1	1	1
Experiment	T	T	T	T	C	C	C

(a) Most desirable design.
All samples are
processed in one batch
i.e. no batch effects.

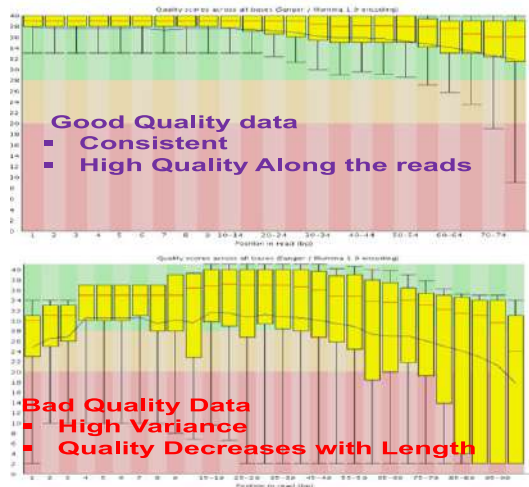
Batch	1	1	2	2	1	2	2
Experiment	T	T	T	T	C	C	C

(b) Desirable design if batches are
inevitable. Experimental
conditions are not completely
confounded with batches.

Batch	1	1	1	2	2	2	2
Experiment	T	T	T	C	C	C	C

(c) Undesirable design,
batches confounded
completely with
experimental conditions.

Fig. 2 Desirable and undesirable experimental designs for treatment (T) and control (C) sample profiling.



(a) Per base Quality (Phred) Score
Distribution along the base position of reads



(b) Per Sequence Quality (Phred score)
Distribution

Fig. 3 Quality assessment of RNA-seq libraries by checking.

quality data are shown in **Fig. 3**. Finally, the mapping quality score which is the probability that the read is misplaced after the alignment of the read, provides a measure of the overall quality of the data.

The raw read quality estimation involves determining per base quality and per sequence quality, identification of duplicates, GC content, and presence of adaptors/over-represented sequences (see **Fig. 4** for an example). FastQC (for Illumina: see “Relevant Websites section”) and NGSQC (platform independent) (Dai *et al.*, 2010) are two popular tools, that provides the quick diagnostics of an NGS library. Removal of poor quality reads & bases, trimming of adaptors can be performed via FASTX-Toolkit (see “Relevant Websites section”) and Trimmomatic (Bolger *et al.*, 2014).

In case of scRNA-seq data, distinguishing biological heterogeneity from technical artifacts makes it much more difficult to assess the quality. The limited amount of RNA available from a single cell makes it difficult to assess the quality of RNA prior to sequencing. Previous studies have used the expression levels of housekeeping genes. But this approach is not used anymore after the demonstration that they are highly variable and associated with cell-types (Oyulu *et al.*, 2012). Often the number of genes detected and the mapping rates are used as QC measures to remove low-quality cells prior to downstream analysis. The proportion of reads mapping to the mitochondrial genes is an indication of cells that are undergoing apoptosis and can be used to remove low-quality cells (Jiang *et al.*, 2016).

- (2) **Mapping:** Reads from an RNA-seq experiment can be mapped to the genome or a known transcriptome. Mapping to the genome is performed if the goal of the experiment includes identification of novel isoforms. For a typical high-quality human sample library, >70% of mapping rate to human genome is expected. TopHat2 (Kim *et al.*, 2013) and STAR (Dobin *et al.*, 2013) are popular algorithms (spliced aware mapper i.e., accounting for splice junctions during mapping) for genome mapping. Whereas, the transcriptome based mapping can be performed by bowtie2/bwa (Langmead and Salzberg, 2012; Li and Durbin, 2010). Transcriptome based mapping yields lower percentage of reads mapped due to lack of sequences of unannotated transcripts. When a reference genome is not available (for non-model organisms) or is incomplete, then *de novo* transcriptome assembly should be performed. Trinity (Grabherr *et al.*, 2011), Oases (Schulz *et al.*, 2012) and SOAP *denovo*-Trans (Xie *et al.*, 2014) are some of the popular algorithms for transcriptome assembly. Longer reads or paired-end reads will aid in the *de novo* assembly. It is recommended to combine all the reads from multiple samples together and then do the combined assembly to generate the

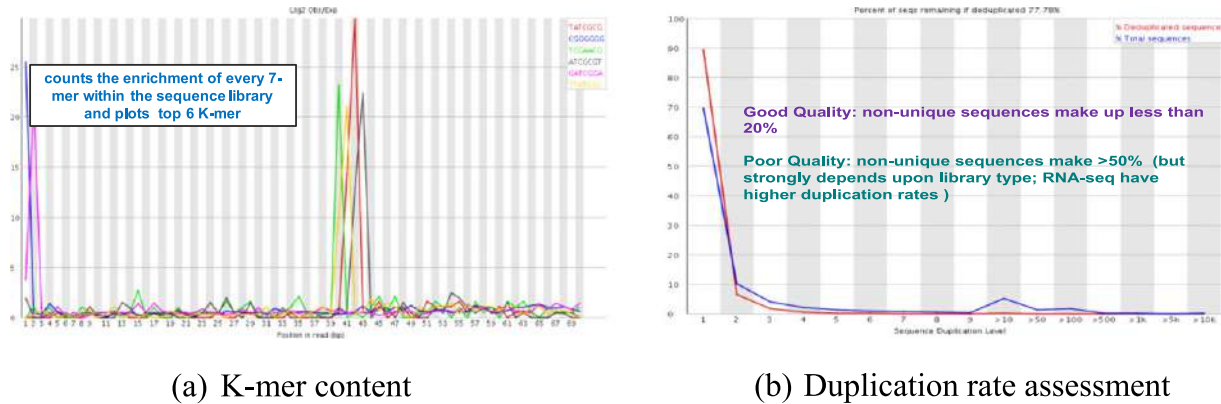


Fig. 4 K-mer and Duplication rate assessment.

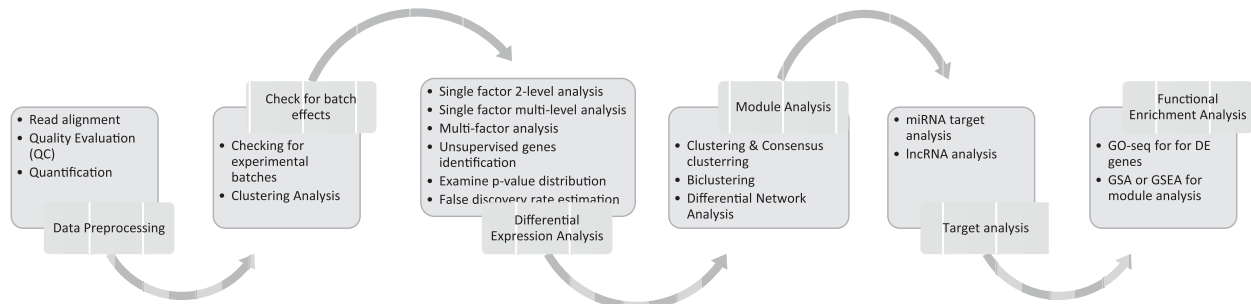


Fig. 5 Schematic of bulk sample analysis.

reference set of contigs. These reference contigs can be used for mapping, expression quantification, and comparison across different groups.

- (3) **Quantification:** The important step in RNA-seq data analysis is to estimate gene and transcript abundance. *HTSeq* (Anders *et al.*, 2015) or *featureCounts* (Liao *et al.*, 2014) quantify the abundance by aggregating the number of mapped reads in each exon (determined by Gene Transfer Format [GTF] file), as desired. In addition to aggregating the raw counts, various per sample normalization methods have been developed that accounts for feature-length and library sizes. RPKM (reads per kilobase of exon model per million reads), FPKM (Fragment per kilobase of exon model per million reads) and TPM (Transcript Per Million) are common measures of expression provided by various analysis tools: cufflinks (Trapnell *et al.*, 2010), RSEM (Li and Dewey, 2011), eXpress (Roberts and Pachter, 2013) and Kallisto (Bray *et al.*, 2016). FPKM is used for paired-end read sequencing and RPKM for single-end reads. Whereas TPM, unlike RPKM, skips normalization for gene length after sequencing depth normalization. It makes the sum of all TPMs in all samples same and aid in the comparison of expression profiles among different samples. Transcript abundance is computed as average of normalized abundance of exons in the transcript.

Bulk Transcriptomic Analysis

Bulk sample analysis is a common feature of all experimental designs involving RNA-seq. Its primary aim is to identify genes and gene modules associated with biologically relevant structure in samples e.g., treated vs control samples, tumors of varying clinical attributes, samples of different cellular states, etc. The schematic in Fig. 5 captures the core steps in bulk transcriptome analysis. Each step after the data preprocessing step is described below.

Differential Expression Analysis

A typical RNA-seq study may involve analysis of the impact of one or more factors relevant to the study on the transcriptome, the identification of DEGs. For example, several clinical conditions can affect genes' expression in a patient sample, multiple biological conditions such as cell cycle phase and gene knock-outs can affect the expression of genes in the associated pathways. A factor, representing a biological or a clinical condition, may be categorical with 2 or more levels, or real-valued.

From statistics point of view, the study designs can be classified into three categories: (1) Single factor, 2-level design, (2) single factor, multi-level design, and (3) multi-factor design. Though the analytical approaches used for multi-factor designs are applicable

for the single factor designs, well-tailored methods that may yield better results in sensitivity and specificity are available for the single factor designs. Hence, we will discuss the methods for all three cases separately. The popularly used statistical packages LIMMA (Ritchie *et al.*, 2015), edgeR (Robinson *et al.*, 2010) and DESeq (Anders and Huber, 2010) offer options for all these cases.

Single factor differential expression analysis

In single factor differential expression analysis, the factor takes two or more discrete levels or continuous valued. For example, the factors could be oestrogen receptor (ER) status (ER+ vs ER−, a 2-level factor), p53 mutation status (p53+ vs. p53−, a 2-level factor), histological grade of the tumor (Grade1, Grade2 and Grade 3; a multi-level factor), treatment response (continuous valued in terms of dosage) or any other factor of interest. If the expression of a gene is significantly different for any level of the factor relative to the other levels, the gene will be identified to be DEG.

2-Level single factor DEG analysis

A 2-level differential expression can be analyzed using t-test and Mann–Whitney–Wilcoxon test (Kanji, 1993). Each level of the factor is treated as a group. However, they are not suitable for designs of small sample sizes of less than 10 per group. In such cases, empirical Bayes methods such as LIMMA, edgeR, or DESeq are to be used.

Multi-level single factor DEG analysis

While being applicable, it is not optimal to use 2-level DEG analysis methods to multi-level DEG analysis, owing to $K(K-1)/2$ tests needed for each gene for K-levels. Therefore, multi-level DEG analysis is performed by ANOVA and Kruskal–Wallis tests (Kanji, 1993) for large sample size designs such as large disease cohorts. However, the multi-group option available in the empirical Bayes methods (LIMMA, edgeR and DESeq) needs to be used for typical designs of sample sizes < 10 per group. All these methods test whether there are any statistically significant differences between the means of any one of the group compared to the remaining groups.

Multi-factor differential expression analysis

To decipher the effects of multiple factors, a multivariate or multi-factor analysis will be performed using linear models and generalized linear models (Dobson, 1990). One may use generalized linear models if the errors are not normally distributed but belong to one of the members of the exponential family of distributions. In this case, all samples are treated to belong to one group and the model accounts for the sample characteristics by their attributes. For example, Miller *et al.* (2005) used a linear model to identify p53 mutation specific changes among expression of nearly 30,000 genes using a cohort of > 250 tumors. Expression of a gene is modeled as a combination of *tumor grade*, *ER status*, and *p53 mutation status*. The statistical significance of the estimated parameters of the linear model was evaluated using ANOVA procedure. The genes were ranked by the coefficient of the factor *p53 status* and the top genes were selected to have been affected by p53 mutation specifically.

Deconvolving batch effects

It is important to remove batch effects prior to downstream analysis of data using batch effect removal methods (Leek, 2014). The known batch effects can also be dealt by COMBAT (Johnson *et al.*, 2007) analysis or by incorporating the known batch factors as independent variables in multi-factor differential expression analysis. The unknown batch effects can be identified by whole transcriptome clustering of samples or Surrogate Variable Analysis (SVA) (Leek *et al.*, 2012). However, the observations have to be carefully interpreted for their biological relevance in the data before correcting for them. The more complicated situation is when the batch is completely confounded with that of the biological factor of interest (Fig. 2(c)). Though it is most undesirable, DEGs can be identified if the biological effect is significantly stronger than the batch effect on the gene expression measurements. In such a case, one can use iPLR (Li *et al.*, 2012), as was applied to identify DEGs in batch confounded analysis in this publication, for a better estimate of false discovery rate (FDR) of differential expression followed by effect size filtering.

False discovery rate (FDR) assessment

A critical step towards choosing the right set of differentially expressed genes is the assessment of false discovery rate. The popular differential expression analysis packages such as LIMMA and DESeq provide estimates of local and global FDR. However, the distribution of p-values needs to be examined, see Fig. 6(A) for ideal or theoretical distribution. However, it is common to see the distributions shown in Fig. 6(B–D). FDR estimates can be improved by using *fdrtool* (Strimmer, 2008) or *ConReg-R* (Li *et al.*, 2011). Careful consideration of all aspects of FDR estimation lead to discovering the impact of mutation in PRPF8 on mRNA splicing in retinal tissue (Korir *et al.*, 2014).

Analysis for Module Identification

A Gene's function is dependent on how the other genes in the cell function. For example, transcription factors' expression affects the expression of their downstream genes, coordinated expression or repression of multiple genes is important to carry out functions such as DNA replication, proliferation and apoptosis. Such functional dependencies among genes need to be elicited in the biological conditions under investigation. Though biological systems are complex, they are amenable to systematic analysis to

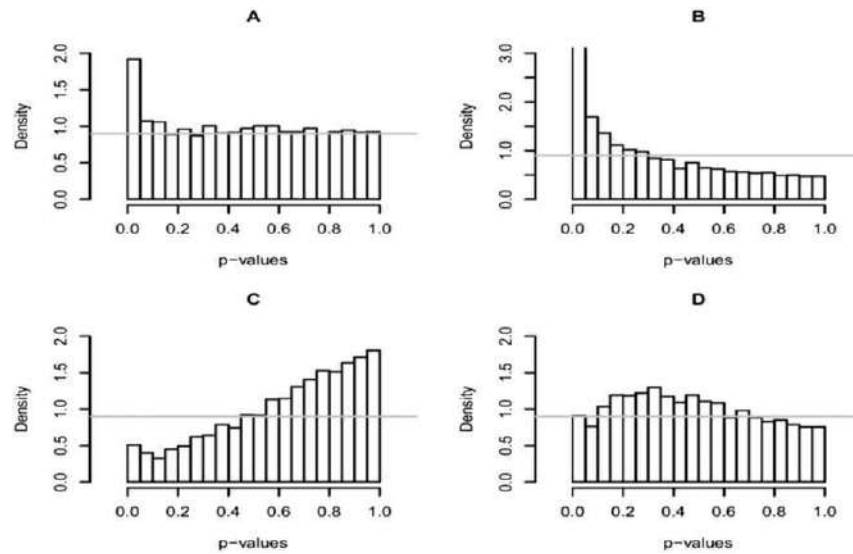


Fig. 6 (A) Desired or theoretical distribution of p-value, and (B–D) are typical distributions encountered in practice which need to be appropriately analyzed to get improved estimates of false discovery rates. The grey horizontal line shows the proportion of null hypotheses in the data, π_0 . Reproduced from Li *et al.* (2011).

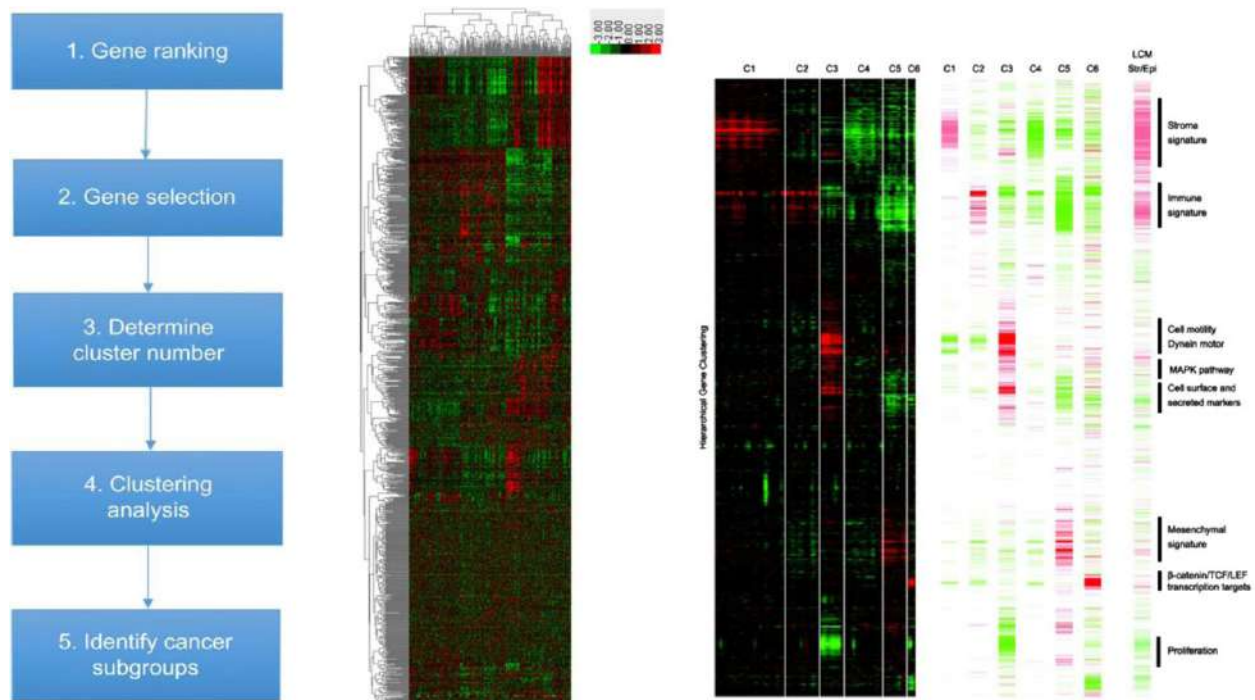


Fig. 7 Transcriptomic clustering procedure and an example of hierarchical clustering of expression data. Reproduced from Lawlor *et al.* (2012). And k-means based consensus clustering of ovarian cancer data for subtyping. Reproduced from Tothill *et al.* (2008).

elicit these functional dependencies by identifying gene sets or modules associated with biological conditions using ‘module analysis’ approaches: (1) Clustering, (2) biclustering, and (3) differential network analysis.

Clustering is an unsupervised analysis technique to discover hidden structure in the data such as, gene modules and heterogeneity among samples. Of the multitude of clustering techniques, hierarchical and K-means clustering are used extensively. The clustering analysis procedure for gene expression data and examples of clustering are shown in Fig. 7. The *multiClust* (Lawlor *et al.*, 2016) is an easy-to-use R-package to deal with all steps associated with clustering, including visualization. However, ‘cluster’ (Eisen *et al.*, 1998) software offers an easy to use graphical interface for clustering, though it doesn’t offer important options for a variety of steps in clustering such as feature selection and distance metrics. The package “Treeview” (Eisen *et al.*, 1998) offers an easy to use graphical interface for interactive visualization of clustering results.

To eliminate spurious and unstable clustering of genes and samples, consensus clustering (CC) (Korir *et al.*, 2014) is employed. The central idea of consensus clustering is that the genes/samples that cluster together consistently for multiple bootstrap runs are the robust clusters and the remaining are spurious.

Clustering and CC analysis of expression data was successfully applied to identify subtypes in cancer. Cancers are typically classified depending on their tissue of origin. However, several genomic studies are providing more detailed molecular characterizations of tumors, and thus bring about the possibility of a more accurate classification based on their molecular profiling (Perou *et al.*, 2000; Tothill *et al.*, 2008). The molecular profiles, in *The Cancer Genome Atlas (TCGA)* project, revealed that cancer from the same tissue can be classified into molecular subtypes with distinct biology and clinical outcome. In many cases, molecular subtype information was shown to provide avenues for better therapeutic intervention.

Biclustering is based on mutually reinforcing modules of genes and samples, rather than all sample approach used in the clustering described above. The difference between biclustering and clustering (Chia and Karuturi, 2010) is depicted in Fig. 8. Biclustering was used to identify co-modules of genes and samples to elicit underlying biology in large cohorts of samples. *biclust* package (Kaiser and Leisch, 2008) allows researchers to explore a variety of biclustering algorithms which were shown to identify a different types of biclusters (Chia and Karuturi, 2010). The *ChiaKaruturi()* function in 'biclust' package can be used to unify the ranking of the biclusters output by multiple biclustering algorithms. Xie *et al.* (2018) provides overview of how biclustering was used in biological discovery.

Differential Network (co-expression) Analysis, in contrast to clustering and biclustering, is used to identify co-expressed gene modules in a supervised setting. In this case, we are interested in identifying gene modules that exhibit significant changes in their co-expression patterns, see Fig. 9 for illustration. It has been shown to be an effective technique to identify gene modules and hub genes in numerous studies (Bostrom *et al.*, 2011; Gillis and Pavlidis, 2009; Ray and Zhang, 2010). A variety of algorithms are available to conduct univariate (2-level or multi-level) and multivariate differential co-expression analysis (Kostka and Spang, 2004; Li and Karuturi, 2010; Karuturi *et al.*, 2006; Liany *et al.* 2017).

Variant Calling From RNA-seq Data

Identification of genomic variants (SNPs, Indels, SVs) is important to uncover the genotype and phenotype relationship. Whole exome sequencing (WES) and whole genome sequencing (WGS) are most commonly used to identify these genomic variants.

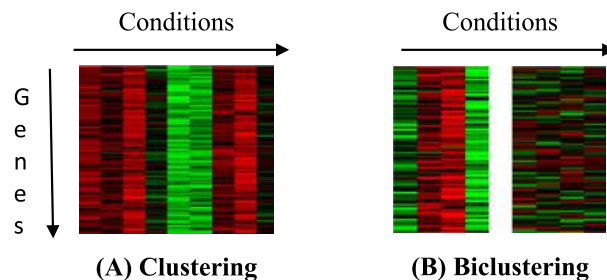


Fig. 8 Heatmaps (red for induction and green for repression) illustrating difference between clustering and biclustering (A) a cluster of genes, genes are co-expressed across all conditions; (B) a bicluster of genes, genes are co-expressed only among a subset of conditions (heatmap on the left) and the heatmap on the right shows no co-expression on the remaining conditions. Reproduced from Chia and Karuturi (2010).

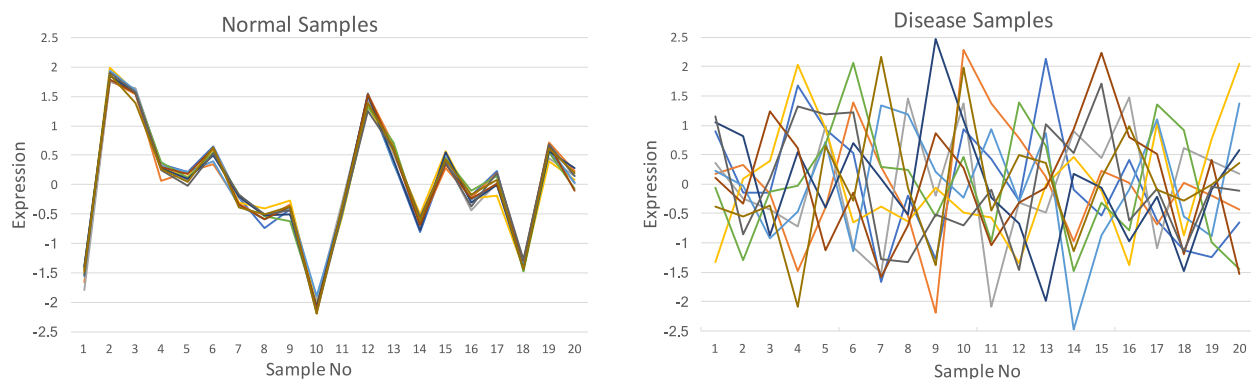


Fig. 9 An illustration of differential co-expression, a gene module is co-expressed in normal samples but not in disease samples.

Recently, calling genomic variants from RNA-seq data has become popular as it offers the variant information for samples without WES/WGS data, and with corresponding WES/WGS data, it offers secondary validation for the variant calls from within expressed transcripts. Piskol *et al.* (2013) have shown that ~40%–50% of the coding variants identified in WGS data and 70% of the variants in expressed genes can be identified from RNA-seq.

The primary challenge in RNA-seq based variant calling is the accurate placement of reads in the genome. Additional complexity in the mapping arises due to RNA splicing. Average human exon length is 150 bp and common sequence reads are (2×100) which leads to the generation of reads crossing the splice junctions. This can create a significant problem in downstream variant calling due to misalignments and thus pipeline used for variant calling needs to have high-quality splice junction databases along with additional filters to remove the false calls. A procedure for variant calling, including the recommended filters, is given in the schematic in Fig. 10 (based on GATK best practices for RNA-seq variant calling and filter suggested in the Piskol *et al.* (2013)).

Analysis of Non-Coding RNAs (ncRNAs)

ncRNAs, such as miRNAs, exert their function by targeting mRNAs. Hence, it is important to understand the analysis of ncRNAs. We focus on the analysis of miRNAs and lncRNAs in this section. These procedures are used only when respective library preparation methods are used for sequencing.

miRNA-seq analysis

miRNAs belong to the family of small (~22 nucleotide long) ncRNAs and function as post-transcriptional regulators in plants and animals. miRNAs commonly function by binding to 3'UTR of target mRNAs and causing its cleavage or translation inhibition (Ambros, 2004; Lu *et al.*, 2008). Though, there is evidence that miRNAs may act as positive regulators in some cases (Vasudevan *et al.*, 2007). High conservation of miRNAs among evolutionarily divergent species suggests that these molecules are involved in essential biological processes (Pasquinelli *et al.*, 2000) such as cell growth, cell proliferations, embryonic development, apoptosis and tissue differentiation (Esquela-Kerscher and Slack, 2006).

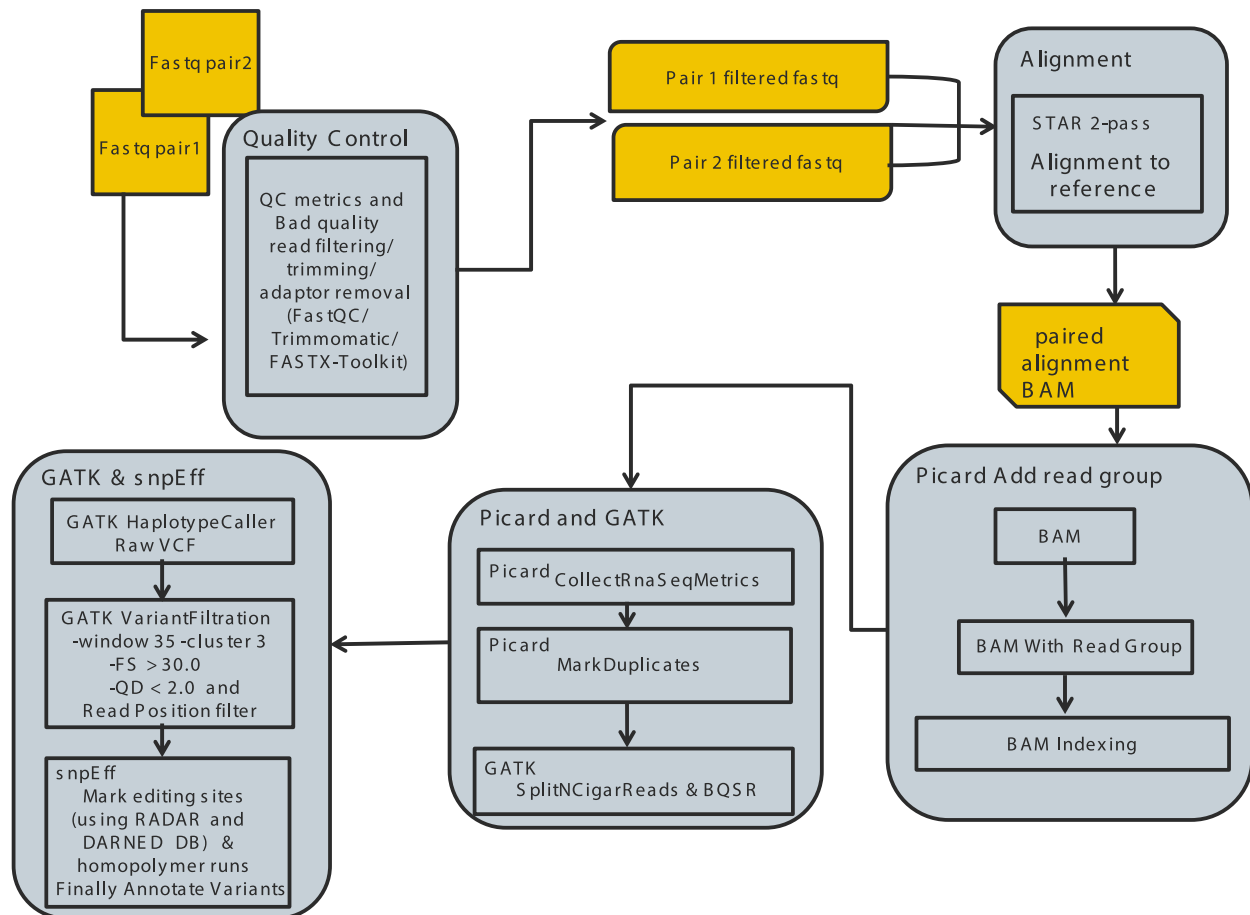


Fig. 10 Schematic of analysis of RNA-seq data for variant calling.

miRNA-seq data analysis scheme is shown in **Fig. 11(a)**. The first step in the analysis of miRNA-seq data is trimming of adaptor sequence and then examining the distribution of sequencing length to determine the quality of the library, peak at ~22 bp usually ensures high-quality library (**Fig. 11(b)**). Afterwards, depending on the goal of the experiment, sequences can be mapped to miRBase (a high-quality database of known miRNA species) or reference genome. Mapping to miRBase ([Kozomara and Griffiths-Jones, 2014](#)) will provide the quick estimation of the known miRNA species but mapping to the genome can detect novel miRNAs. Popular tools for miRNA analysis include miRdeep2 ([Friedlander et al., 2012](#)), miRanalyzer ([Hackenberg et al., 2011](#)) and miRspring ([Humphreys and Suter, 2013](#)). In [Pullagura et al. \(2018\)](#) we used these tools to investigate the requirement of the DICER cofactors TARBP2 and PRKRA in miRNA biogenesis.

lncRNA-seq analysis

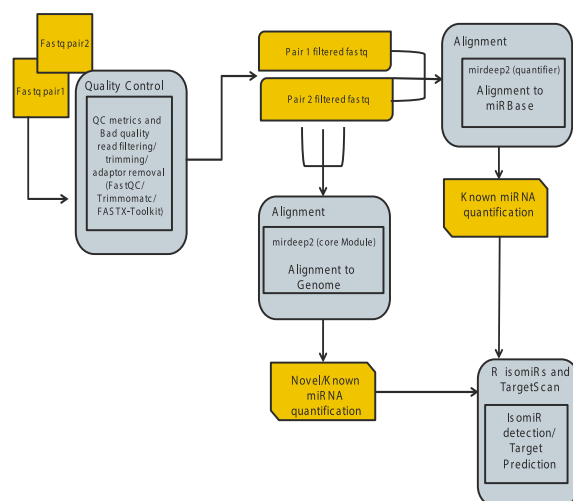
Another class of ncRNAs is long ncRNAs (lncRNAs) with a length of more than 200 nucleotides ([Kung et al., 2013](#)). Numerous lncRNAs are present throughout the genome. One-way to classify them is according to their genomic location ([Kung et al., 2013](#)). According to the genomic location, lncRNAs can be classified as stand-alone lncRNAs (distinct transcription units located in sequence space not overlapping with protein-coding genes), Natural antisense transcripts, Pseudogenes ("remnant" of genes that have lost their coding potential due to mutations), Long intronic ncRNAs (Found within introns of annotated genes) and Divergent transcripts, promoter-associated transcripts, and enhancer RNAs (produced from the vicinity of transcription start sites or produced from the enhancer region). lncRNAs do perform various biological functions in a variety of developmental process and disease like cancer ([Huarte, 2015](#)). Typical bioinformatics workflow for the screening of lncRNAs has three steps: (i) Alignment to the genome of interest [Tophat2 and STAR]; (ii) reference guided (Cufflinks) or *ab initio* transcriptome assembly ([Guttman et al., 2010](#)); and, (iii) coding potential estimation of assembled transcripts ([Hu et al., 2017](#)). A comprehensive pipeline for lncRNA analysis is available online, see "Relevant Websites section".

Functional Analysis

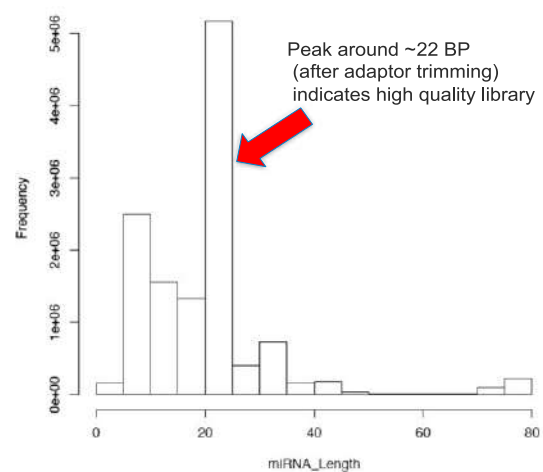
The functional relevance analysis of the DEGs and gene modules is a critical step towards interpreting their function. It is typically carried out by conducting enrichment analysis of the pathways (e.g., KEGG ([Kanehisa et al., 2017](#))), Gene Ontology (GO) terms ([Ashburner et al., 2000](#)), and pre-identified publicly available gene signatures such as MSigDB ([Liberzon, 2014](#)), GeneSigDB ([Culhane et al., 2012](#)) and DSigDB ([Yoo et al., 2015](#)). The central idea is if a module represents a certain function, the genes in the relevant pathway or the signature will be enriched in the module. It is typically tested using Fisher's exact test ([Fisher, 1945](#)) or KS test as in GSA ([Subramanian et al., 2005](#)) and GSEA ([Subramanian et al., 2005](#)). However, GO-seq ([Young et al., 2010](#)) is recommended for the functional analysis of DEGs from RNA-seq data, GO-seq accounts for the gene length dependent bias in identifying DEGs (power to identify differential expression of longer genes is more in RNA-seq studies).

Diagnosis & prognosis

The National Institutes of Health Biomarkers Definitions Working Group defined a biomarker as a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a



(a) Schematic of miRNA-analysis



(b) Length based filtering of miRNAs from an RNA library

Fig. 11 Schematic of miRNA-seq data analysis procedure.

therapeutic intervention (Biomarkers Definitions Working, G, 2001). Gene expression signatures based biomarkers are used to diagnose disease (diagnostic biomarker) as well as to predict patient's prognosis (prognostic marker) or response to therapeutics (predictive marker) (Italiano, 2011).

In general, biomarkers are designed using systematic principles of classifier design. Based on the empirical study (Dudoit *et al.*, 2002) of gene expression based classifier design, simple classification algorithms like k-nearest neighbor (KNN) and diagonal linear discriminant analysis (DLDA) are recommended over more complicated classification algorithms like support vector machines (SVMs) for biomarker studies.

Single Cell Transcriptomic Analysis

Single cell transcriptome profiling is used to identify cell types, interactions among them and modeling stochasticity of gene expression in a sample, primarily using mRNA profiling. DEG analysis is fundamental to these objectives.

Differential Expression Analysis

The methods developed for identifying DEGs in bulk RNA-seq are suboptimal for scRNA-seq data due to higher noise contributed by both technical and biological factors specific to scRNA-seq data. The primary technical factors are low amount of transcriptome in a cell and hi-level of amplification required for sequencing which lead to amplification biases or dropout events i.e., the amplification or the capture is not successful for many mRNAs (Vallejos *et al.*, 2017). Biological variability is also higher in scRNA-seq data due to the stochastic nature of transcription (Raj *et al.*, 2006) and multimodality in gene expression (Shalek *et al.*, 2013) originating from the presence of multiple possible cell states within a cell population. All these together pose serious challenges for detection of DEGs.

Recently, several methods that explicitly model the technical and biological variability in scRNA-seq data have been developed to identify the DEGs in scRNA-seq data (Dal Molin *et al.*, 2017). Model-based Analysis of Single-cell Transcriptomics, MAST (Finak *et al.*, 2015), explicitly considers the dropouts using a bimodal distribution with expression strongly different from zero and proposes a generalized linear model to fit the data. Single-Cell Differential Expression (Kharchenko *et al.*, 2014), models the counts of each cell as a mixture of a zero-inflated Negative Binomial distribution and a dropout component and uses a Bayesian model to estimate the posterior probability that a gene is differentially expressed in one group with respect to another. Single-cell Differential Distributions, scDD (Korthauer *et al.*, 2016), is based on a multimodal Bayesian modeling framework for explicitly modeling the multimodal distributions of single cells and testing for differentially distributed genes associated with this multimodality.

Cell Type Identification

In a typical human body, there exists an estimated 40 trillion cells and is classified into approximately 200 cell-types based on morphology (Bianconi *et al.*, 2013). Advances in immunofluorescence and flow cytometry have enabled more refined classification based on the presence or absence of various surface markers (Pruszek *et al.*, 2007). However, the number of cell surface markers are limited to discern the plethora of cell types and these techniques are limited to easily dissociable tissues like blood (Roussel *et al.*, 2010). The development of scRNA-seq has changed the status quo and enabled the identification of novel cell-types based on the entire transcriptome of individual cells. scRNA-seq has been used to study different tissues and organs, both during development and at a fixed point in time (Rizvi *et al.*, 2017). Almost all the methods involve feature selection steps to remove noisy and irrelevant features, dimensionality reduction steps to avoid the "curse of dimensionality" and finally clustering steps to identify cell-types. The process of cell type identification and the recommended tools are outlined in Fig. 12. The most popular options currently available for each step are shown. These are not exhaustive lists and the most appropriate method in each step will depend on the data type and the technical platform used for generating the scRNA-seq data. SCEED package (Abrams *et al.*, 2018) helps in the experimental design and identifying optimal analysis procedure.

Cell:Cell Communication Analysis

Cell:Cell communication analysis is used to elicit cell communication networks facilitated by ligand-receptor interaction. *Single Cell-2-Cell Communicator* (SC2CC) [manuscript in preparation] helps identify cell:cell communication network using scRNA-seq data, in contrast to CCCExplorer (Choi *et al.*, 2015) which is used for sorted bulk RNA-seq data analysis. A preliminary version of SC2CC was used in Lawlor *et al.* (2017); Fig. 13. The primary features of SC2CC are: (1) SC2CC integrates cell type identification methodology described above into cell:cell communication analysis. (2) Selection of expressed ligands is based on both expression level and differential expression analysis. As scRNA-seq data enables identification of multiple cell types, SC2CC will identify differentially expressed ligands using multi-group analysis in *edgeR* and annotate a cell type with a ligand if the ligand's average expression in the cell type is more than the average expression in all cell types and it is significant by *edgeR* multi-group analysis. (3) SC2CC analyzes for both autocrine (cell communicates with itself) and paracrine (one cell communicates with another cell) signaling among all cell types. (4) SC2CC has augmented ligand-receptor pairs (1578, with 469 ligands and 342 receptors) database for the analysis. SC2CC identifies the receptor activation and assigns

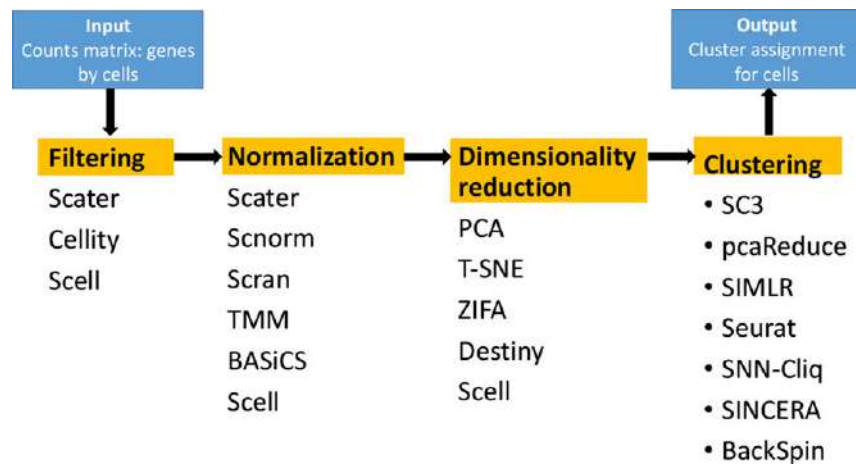


Fig. 12 Steps involved in cell-type identification from single-cell RNA-seq data.

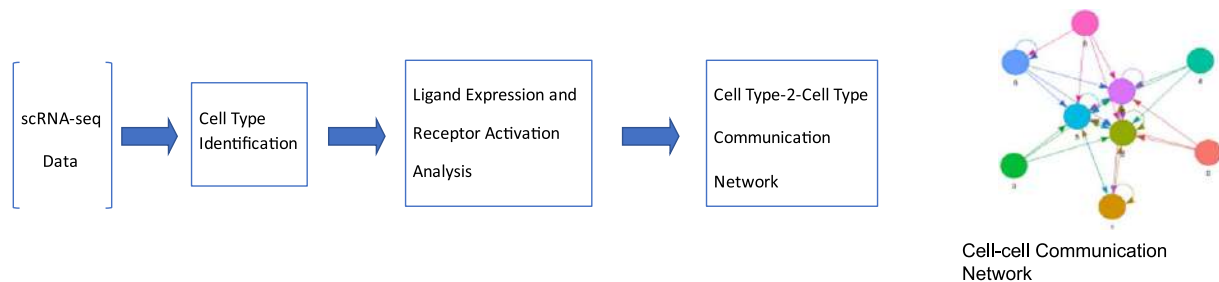


Fig. 13 Schematic of SC2CC.

autocrine/paracrine signaling using PAGODA (Fan *et al.*, 2016). The analysis of scRNA-seq data for cell:cell communication with SC2CC leads to a network of communication among different cell types, both autocrine and paracrine signaling.

Discussion

Transcriptomics data is a closer representation of abundance of regulatory molecules in the cell. In addition, the availability of easier-to-work and cost-effective technologies to generate transcriptomic data made it an integral part of most genomic studies, including disease genomics. Though mRNA, miRNA, and lncRNA are most commonly explored transcriptomic data, several other types of ncRNAs are being studied recently: circRNAs, piRNAs, and nasRNAs (Dong *et al.*, 2017; Kim *et al.*, 2009; Korneev and O'Shea, 2005; Weick and Miska, 2014). RNA specific library preparation methods have been employed to generate data for the type of RNA in consideration. Hence, several data processing steps need to be customized for the type of RNA analyzed. Multiple studies such as spermatogenesis require complex data processing of deeply sequenced data to effectively identify one type of ncRNA from the other type of ncRNAs (e.g., piRNAs from miRNAs) as they are of similar length and abundance. In addition to ncRNAs, fusion genes are another class of RNAs resulting from structural variations in the genome, especially in cancer transcriptomes. Several algorithms are available in the literature to identify fusion genes from RNA-seq data (Jia *et al.*, 2013; Kim and Salzberg, 2011). These methods analyze the junction reads (reads mapping to exons of different genes) and paired-end read mapping information (reads of the same paired-end read mapping to different genes). In addition, isoforms play a critical role in several biological systems, especially in the immune system. Though shotgun sequencing data (e.g., Illumina) has been used to identify isoforms, for better accuracy, it is being used in conjunction with long read sequencing technologies (PacBio and Oxford-nanopore) recently. These require specialized data processing steps (Weirather *et al.*, 2017).

Besides processing and analysis of primary RNA-seq data, a typical genomic study may require probing of transcriptomic data in different systems: model organisms, diseases, and populations. A variety of databases host plethora of transcriptomic data from a vast range of systems for researchers to query and examine the observations of the study in the context. Generic databases such as GEO and SRA, as well as specialized databases such as Oncomine (Rhodes *et al.*, 2004), CBio Portal (Cerami *et al.*, 2012; Gao *et al.*, 2013), and ImmGen (Heng *et al.*, 2008) facilitate such a contextual analysis and validation of the observations of the study. One of the major impediment to such a contextual analysis is the integration of data from multiple platforms as the transcriptomics data has been historically generated using a variety of platforms (e.g., microarrays: Affymetrix, Illumina, and Agilent; sequencing:

Illumina and PacBio, etc). Such an integrative analysis has to be carried out by considering platform specific normalization, and simple procedures such as z-scoring and housekeeping genes' based normalization have been proven to be useful.

In addition, most genomic studies include more than one type of transcriptomic data and genomic data (e.g., genomic variants data, epigenetic data, methylation data and TF-DNA interaction data). Typically, we are interested in studying the relationship and impact of a variety of genomic variants on the transcriptome. Each such analysis requires specialized integration strategies. For example, considerations of mapping single nucleotide or epigenetic variants in promoters to genes vary from that of enhancers or insulators due to their distance from genes; short-range structural variants may need to be interpreted differently from that of long-range structural variants. The strategies include mapping to the closest genes and conducting functional analysis of the mapped genes while accounting for non-uniform distribution of genes in the genome (McLean *et al.*, 2010; Zhou and Murthy Karuturi, 2012)

Taken together, transcriptomic data generation has been extremely impactful and the analysis is reasonably matured: from data processing, quality control, down-stream analysis and integrative analysis. Suitable experimental design, normalization and careful use of the plethora of databases will help elicit important biology of development and disease.

Author Contributions

George and Karuturi conceptualized and lead the writing of the chapter. All authors contributed equally to the chapter.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Biological Database Searching. Characterizing and Functional Assignment of Noncoding RNAs. Codon Usage. Computational Pipelines and Workflows in Bioinformatics. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Prediction of Coding and Non-Coding RNA. Sequence Analysis. Whole Genome Sequencing Analysis

References

- Abrams, D., *et al.*, 2018. A computational method to aid the design and analysis of single cell RNA-seq experiments for cell type identification. Available at: <https://doi.org/10.1101/247114>.
- Ambros, V., 2004. The functions of animal microRNAs. *Nature* 431 (7006), 350–355.
- Anders, S., Huber, W., 2010. Differential expression analysis for sequence count data. *Genome Biol.* 11 (10), R106.
- Anders, S., Pyl, P.T., Huber, W., 2015. HTSeq – A python framework to work with high-throughput sequencing data. *Bioinformatics* 31 (2), 166–169.
- Ashburner, M., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25 (1), 25–29.
- Bianconi, E., *et al.*, 2013. An estimation of the number of cells in the human body. *Ann. Hum. Biol.* 40 (6), 463–471.
- Biomarkers Definitions Working Group, G., 2001. Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework. *Clin. Pharmacol. Ther.* 69 (3), 89–95.
- Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* 30 (15), 2114–2120.
- Bostrom, P., *et al.*, 2011. MMP-1 expression has an independent prognostic value in breast cancer. *BMC Cancer* 11, 348.
- Bray, N.L., *et al.*, 2016. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* 34 (5), 525–527.
- Cech, T.R., *et al.*, 1983. Secondary structure of the Tetrahymena ribosomal RNA intervening sequence: Structural homology with fungal mitochondrial intervening sequences. *Proc. Natl. Acad. Sci. USA* 80 (13), 3903–3907.
- Cerami, E., *et al.*, 2012. The cBio cancer genomics portal: An open platform for exploring multidimensional cancer genomics data. *Cancer Discov.* 2 (5), 401–404.
- Chen, K., *et al.*, 2015. The overlooked fact: Fundamental need for spike-in control for virtually all genome-wide analyses. *Mol. Cell. Biol.* 36 (5), 662–667.
- Chia, B.K., Karuturi, R.K.M., 2010. Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms. *Algorithms Mol. Biol.* 5, 23.
- Ching, T., Huang, S., Garmire, L.X., 2014. Power analysis and sample size estimation for RNA-seq differential expression. *RNA* 20 (11), 1684–1696.
- Choi, H., *et al.*, 2015. Transcriptome analysis of individual stromal cell populations identifies stroma-tumor crosstalk in mouse lung cancer model. *Cell Rep.* 10 (7), 1187–1201.
- Chow, K.H., *et al.*, 2017. S100A4 Is a biomarker and regulator of glioma stem cells that is critical for mesenchymal transition in glioblastoma. *Cancer Res.* 77 (19), 5360–5373.
- Crick, F.H., 1958. On protein synthesis. *Symp. Soc. Exp. Biol.* 12, 138–163.
- Culhane, A.C., *et al.*, 2012. GeneSigDB: A manually curated database and resource for analysis of gene expression signatures. *Nucleic Acids Res.* 40 (Database issue), D1060–D1066.
- Dai, M., *et al.*, 2010. NGSQC: Cross-platform quality analysis pipeline for deep sequencing data. *BMC Genom.* 11 (Suppl 4), S7.
- Dal Molin, A., Baruzzo, G., Di Camillo, B., 2017. Single-cell RNA-sequencing: Assessment of differential expression analysis methods. *Front. Genet.* 8, 62.
- Dobin, A., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* 29 (1), 15–21.
- Dobson, A.J., 1990. *An Introduction to Generalized Linear Models*. London: Chapman and Hall.
- Dong, Y., *et al.*, 2017. Circular RNAs in cancer: An emerging key player. *J. Hematol. Oncol.* 10 (1), 2.
- Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.* 97 (457), 77–87.
- Eisen, M.B., *et al.*, 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95 (25), 14863–14868.
- Esquela-Kerscher, A., Slack, F.J., 2006. OncomiRs – microRNAs with a role in cancer. *Nat. Rev. Cancer* 6 (4), 259–269.
- Fan, J., *et al.*, 2016. Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nat. Methods* 13 (3), 241–244.

- Finak, G., *et al.*, 2015. MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* 16, 278.
- Fisher, R.A., 1945. A new test for 2×2 tables. *Nature* 156, 388.
- Friedlander, M.R., *et al.*, 2012. miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. *Nucleic Acids Res.* 40 (1), 37–52.
- Gao, J., *et al.*, 2013. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Sci. Signal.* 6 (269), p11.
- George, J., *et al.*, 2016. Leukaemia cell of origin identified by chromatin landscape of bulk tumour cells. *Nat. Commun.* 7, 12166.
- Gillis, J., Pavlidis, P., 2009. A methodology for the analysis of differential coexpression across the human lifespan. *BMC Bioinform.* 10, 306.
- Grabherr, M.G., *et al.*, 2011. Full-length transcriptome assembly from RNA-seq data without a reference genome. *Nat. Biotechnol.* 29 (7), 644–652.
- Guttman, M., *et al.*, 2010. Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nat. Biotechnol.* 28 (5), 503–510.
- Hackenberg, M., Rodriguez-Ezpeleta, N., Aransay, A.M., 2011. miRanalyzer: An update on the detection and analysis of microRNAs in high-throughput sequencing experiments. *Nucleic Acids Res.* 39 (Web Server issue), W132–W138.
- Heng, T.S., Painter, M.W., 2008. The Immunological Genome Project: Networks of gene expression in immune cells. *Nat. Immunol.* 9 (10), 1091–1094.
- Huarte, M., 2015. The emerging role of lncRNAs in cancer. *Nat. Med.* 21 (11), 1253–1261.
- Humphreys, D.T., Suter, C.M., 2013. miRspring: A compact standalone research tool for analyzing miRNA-seq data. *Nucleic Acids Res.* 41 (15), e147.
- Hu, L., *et al.*, 2017. COME: A robust coding potential calculation tool for lncRNA identification and characterization based on multiple features. *Nucleic Acids Res.* 45 (1), e2.
- Italiano, A., 2011. Prognostic or predictive? It's time to get back to definitions!. *J. Clin. Oncol.* 29 (35), 4718. (author reply 4718–9).
- Jiang, P., Thomson, J.A., Stewart, R., 2016. Quality control of single-cell RNA-seq by SinQC. *Bioinformatics* 32 (16), 2514–2516.
- Jia, W., *et al.*, 2013. SOAPfuse: An algorithm for identifying fusion transcripts from paired-end RNA-seq data. *Genome Biol.* 14 (2), R12.
- Johnson, W.E., Li, C., Rabinovic, A., 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 8 (1), 118–127.
- Kaiser, S., Leisch, F., 2008. Biclust – A toolbox for bicluster analysis in R. In: *Proceedings of the Computational Statistics*.
- Kanehisa, M., *et al.*, 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* 45 (D1), D353–D361.
- Kanji, G.K., 1993. 100 Statistical Tests. London: Sage Publications.
- Karuturi, R.K.M., *et al.*, 2006. Differential friendly neighbors algorithm for differential relationships based gene selection and classification using microarray data. In: *Proceedings of the International Conference on Data Mining (DMIN)*.
- Kharchenko, P.V., Silberman, L., Scadden, D.T., 2014. Bayesian approach to single-cell differential expression analysis. *Nat. Methods* 11 (7), 740–742.
- Kim, D., *et al.*, 2013. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.* 14 (4), R36.
- Kim, V.N., Han, J., Siomi, M.C., 2009. Biogenesis of small RNAs in animals. *Nat. Rev. Mol. Cell Biol.* 10 (2), 126–139.
- Kim, D., Salzberg, S.L., 2011. TopHat-fusion: An algorithm for discovery of novel fusion transcripts. *Genome Biol.* 12 (8), R72.
- Korir, P.K., *et al.*, 2014. A mutation in a splicing factor that causes retinitis pigmentosa has a transcriptome-wide effect on mRNA splicing. *BMC Res. Notes* 7, 401.
- Korneev, S., O'Shea, M., 2005. Natural antisense RNAs in the nervous system. *Rev. Neurosci.* 16 (3), 213–222.
- Korthauer, K.D., *et al.*, 2016. A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol.* 17 (1), 222.
- Kostka, D., Spang, R., 2004. Finding disease specific alterations in the co-expression of genes. *Bioinformatics* 20 (Suppl. 1), i194–i199.
- Kozomara, A., Griffiths-Jones, S., 2014. miRBase: Annotating high confidence microRNAs using deep sequencing data. *Nucleic Acids Res.* 42 (Database issue), D68–D73.
- Kung, J.T., Colognori, D., Lee, J.T., 2013. Long noncoding RNAs: Past, present, and future. *Genetics* 193 (3), 651–669.
- Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* 9 (4), 357–359.
- Lawlor, N., *et al.*, 2012. *multiClust*: An R-package for identifying biologically relevant clusters in Cancer Transcriptome profiles. *Cancer Inform., Libertas Academica*, 15, 103–114.
- Lawlor, N., *et al.*, 2017. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. *Genome Res.* 27 (2), 208–222.
- Leek, J.T., 2014. svaseq: Removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res.* 42 (21), 740–742.
- Leek, J.T., *et al.*, 2012. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics* 28 (6), 882–883.
- Lee, R.C., Feinbaum, R.L., Ambros, V., 1993. The *C. elegans* heterochronic gene *lin-4* encodes small RNAs with antisense complementarity to *lin-14*. *Cell* 75 (5), 843–854.
- Liany, H., Rajapakse, J.C., Karuturi, R.K.M., 2017. MultiDCoX: Multi-factor analysis of differential co-expression. *BMC Bioinform.* 18 (Suppl. 16), 576. (bioRxiv).
- Liao, Y., Smyth, G.K., Shi, W., 2014. featureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* 30 (7), 923–930.
- Liberzon, A., 2014. A description of the Molecular Signatures Database (MSigDB) Web site. *Methods Mol. Biol.* 1150, 153–160.
- Li, J., Choi, K.P., Karuturi, R.K., 2012. Iterative piecewise linear regression to accurately assess statistical significance in batch confounded differential expression analysis. In: *Proceedings of the International Symposium Bioinformatics Research and Applications (ISBRA)*, vol. 7292, pp. 153–164. Springer.
- Li, B., Dewey, C.N., 2011. RSEM: Accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinform.* 12, 323.
- Li, H., Durbin, R., 2010. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* 26 (5), 589–595.
- Li, J., *et al.*, 2011. ConReg-R: Extrapolative recalibration of the empirical distribution of p-values to improve false discovery rate estimates. *Biol. Direct* 6, 27.
- Li, H., Karuturi, R.K.M., 2010. Significance analysis and improved discovery of disease-specific differentially co-expressed gene sets in microarray data. *Int. J. Data Min. Bioinform.* 4 (6), 617–638.
- Lu, M., *et al.*, 2008. An analysis of human microRNA and disease associations. *PLOS ONE* 3 (10), e3420.
- McLean, C.Y., *et al.*, 2010. GREAT improves functional interpretation of cis-regulatory regions. *Nat. Biotechnol.* 28 (5), 495–501.
- Miller, L.D., *et al.*, 2005. An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. *Proc. Natl. Acad. Sci. USA* 102 (38), 13550–13555.
- Oyolu, C., Zakharia, F., Baker, J., 2012. Distinguishing human cell types based on housekeeping gene signatures. *Stem Cells* 30 (3), 580–584.
- Pasquinelli, A.E., *et al.*, 2000. Conservation of the sequence and temporal expression of *let-7* heterochronic regulatory RNA. *Nature* 408 (6808), 86–89.
- Perou, C.M., *et al.*, 2000. Molecular portraits of human breast tumours. *Nature* 406 (6797), 747–752.
- Pham, T.V., Jimenez, C.R., 2012. An accurate paired sample test for count data. *Bioinformatics* 28 (18), i596–i602.
- Piskol, R., Ramaswami, G., Li, J.B., 2013. Reliable identification of genomic variants from RNA-seq data. *Am. J. Hum. Genet.* 93 (4), 641–651.
- Pruszk, J., *et al.*, 2007. Markers and methods for cell sorting of human embryonic stem cell-derived neural cell populations. *Stem Cells* 25 (9), 2257–2268.
- Pullagura, S.R.N., *et al.*, 2018. Functional redundancy of DICER cofactors TARBP2 and PRKRA during murine embryogenesis does not involve miRNA biogenesis. *Genetics* 208 (4), 1513–1522.
- Raj, A., *et al.*, 2006. Stochastic mRNA synthesis in mammalian cells. *PLOS Biol.* 4 (10), e309.
- Ray, M., Zhang, W., 2010. Analysis of Alzheimer's disease severity across brain regions by topological analysis of gene co-expression networks. *BMC Syst. Biol.* 4, 136.
- Rhodes, D.R., *et al.*, 2004. ONCOMINE: A cancer microarray database and integrated data-mining platform. *Neoplasia* 6 (1), 1–6.
- Ritchie, M.E., *et al.*, 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* 43 (7), e47.
- Rizvi, A.H., *et al.*, 2017. Single-cell topological RNA-seq analysis reveals insights into cellular differentiation and development. *Nat. Biotechnol.* 35 (6), 551–560.
- Roberts, A., Pachter, L., 2013. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nat. Methods* 10 (1), 71–73.
- Robinson, M.D., McCarthy, D.J., Smyth, G.K., 2010. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26 (1), 139–140.

- Robinson, M.D., Oshlack, A., 2010. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* 11 (3), R25.
- Roussel, M., *et al.*, 2010. Refining the white blood cell differential: The first flow cytometry routine application. *Cytometry A* 77 (6), 552–563.
- Schulz, M.H., *et al.*, 2012. Oases: Robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics* 28 (8), 1086–1092.
- Shalek, A.K., *et al.*, 2013. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature* 498 (7453), 236–240.
- Sims, D., *et al.*, 2014. Sequencing depth and coverage: Key considerations in genomic analyses. *Nat. Rev. Genet.* 15 (2), 121–132.
- Strimmer, K., 2008. fdrtool: A versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics* 24 (12), 1461–1462.
- Subramanian, A., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* 102 (43), 15545–15550.
- Tang, F., *et al.*, 2009. mRNA-seq whole-transcriptome analysis of a single cell. *Nat. Methods* 6 (5), 377–382.
- Tothill, R.W., *et al.*, 2008. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clin. Cancer Res.* 14 (16), 5198–5208.
- Trapnell, C., *et al.*, 2010. Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28 (5), 511–515.
- Ucar, D., *et al.*, 2017. The chromatin accessibility signature of human immune aging stems from CD8(+) T cells. *J. Exp. Med.* 214 (10), 3123–3144.
- Vallejos, C.A., *et al.*, 2017. Normalizing single-cell RNA sequencing data: Challenges and opportunities. *Nat. Methods* 14 (6), 565–571.
- Vasudevan, S., Tong, Y., Steitz, J.A., 2007. Switching from repression to activation: MicroRNAs can up-regulate translation. *Science* 318 (5858), 1931–1934.
- Weick, E.M., Miska, E.A., 2014. piRNAs: From biogenesis to function. *Development* 141 (18), 3458–3471.
- Weirather, J.L., *et al.*, 2017. Comprehensive comparison of pacific biosciences and oxford nanopore technologies and their applications to transcriptome analysis. *F1000Research* 6, 100.
- Xie, Y., *et al.*, 2014. SOAPdenovo-Trans: De novo transcriptome assembly with short RNA-seq reads. *Bioinformatics* 30 (12), 1660–1666.
- Xie, J., *et al.*, 2018. It is time to apply biclustering: A comprehensive review of biclustering applications in biological and biomedical data. *Brief. Bioinform.*
- Yoo, M., *et al.*, 2015. DSigDB: Drug signatures database for gene set analysis. *Bioinformatics* 31 (18), 3069–3071.
- Young, M.D., *et al.*, 2010. Gene ontology analysis for RNA-seq: Accounting for selection bias. *Genome Biol.* 11 (2), R14.
- Zhou, J., Li, H.R., Karuturi, R.K.M., 2012. RK Bias in genome scale functional analysis of transcription factors using binding site data. *J. Phys. Chem. Biophys.* S4 (S4:002),

Relevant Websites

<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
 Babraham Bioinformatics.
http://hannonlab.cshl.edu/fastx_toolkit/
 FASTX-Toolkit.
<https://github.com/NYU-BFX/lncRNA-screen>
 GitHub.

Regulation of Gene Expression

Y-h Taguchi, Chuo University, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

The Importance of Regulation of Gene Expression

The regulation of gene expression (Fig. 1) is critical process because of the following reasons. All multicellular organisms start their development from the duplication of a single cell: a fertilized egg. Then cascade of duplication results in the formation of multicellular organism. Since no cells' fate (i.e., into which tissues or organs each cell will differentiate) can be predicted, each cell must keep all set of genes everytime it is duplicated. Consequently, all of differentiated cells must keep full set of genes. Thus, it is critically important for each cell to control which genes should be expressed. Loss of proper controls might result in diseases, not functional organs, and even death.

Because of its importance, bioinformatics is also aiming to target regulation of gene expression. Although gene expression is composed of two steps: transcription and translation, both of which are samely important, because of imbalanced development of measurement technology, regulation of transcription is more extensively investigated than that of translation. Although translation is also an important process, because of inferior measurement technology, bioinformatics of translation process is investigated much less. Therefore this article also mainly describes regulation of transcription processes.

Pre-Transcriptional Regulation

Regulation of transcription process is divided to two parts, pre- and post-transcription regulation, respectively. Among these two, the pre-transcription regulation is more popular topics.

Transcription Factor Binding Site

Especially, inference of transcription factor binding site (TFBS) is ever attracting researchers' interest. TF is the protein that binds to promoter region and initiates transcription. Because of its sequence specific nature of binding, TFBS is a easy target to attack using bioinformatics that has great benefits of sequence analysis. Suppose that you have a set of DNA sequences identified by using some technology to which a TF often binds (e.g., ChIP-Seq). Then, all you need to do is to identify enriched sequence (so called motif) among those collection of DNA fragments. If you are very successful, you can predict where each TF binds along DNA sequence by searching where the identified motifs are located.

Promoter Methylation

Although TFBS has ever been studied since TF is protein which can be studied even in pre-genomic era, recent popular topics are not always directly related to protein; it is called epigenetics which means gene expression alteration without DNA sequence

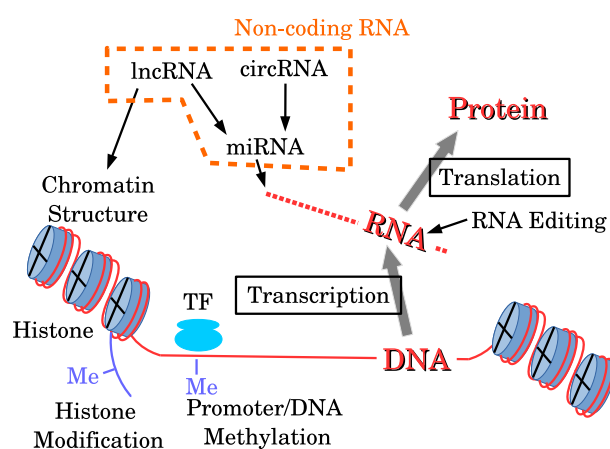


Fig. 1 Schematic that outlines regulation of gene expression. TF bindings, promoter/DNA methylation, histone modification, and chromatin structure affect regulation of gene expression during pre-transcriptional phase. On the other hand, various non-coding RNAs (miRNA, lncRNA and circRNA) and RNA editing affect it during post-transcription phases. Although there are numerous additional protein machineries that mediate these processes, all of them are omitted in order for simplicity. Although non-coding RNAs also must be transcribed from DNA, this process was also excluded from this illustration in order to avoid complicatedness.

modification. Epigenetics is composed of multiple functional elements. Most extensively studied epigenetic effect is promoter methylation. If promoter is methylated, since the binding of TF to the promoter is disturbed, transcription of corresponding mRNA is also disturbed. Since exhausted measurements of promoter methylation was ever established and its functional importance is cleared out, promoter methylation was the most frequently measured epigenetics.

From the technical point of views, methylation is measured by bisulfite treatment that transforms unmethylated cytosine to uracil while methylated cytosine is not. Thus, comparison between original DNA sequence and that treated by bisulfite can give us the information about which cytosine is methylated.

Histone Modification

The next popular epigenetic measurements is histone modification (Senda and Adachi, 2013). Histone is a protein around which DNA winds. Correct and tight winding between DNA and histone is critically important to store DNA which is extremely long polynucleotide into small space that cell nuclei occupies. From the point of views of regulation of gene expression, histone modification can contribute to store DNA in small space since histone modification can alter how tightly DNA winds around histone. When the density of histone along DNA is low, DNA sequence is bared and is easy to be transcribed. On the other hand, if DNA tightly winds around histone, it is not easy for DNA to be transcribed. Histone modification that means binding of small molecules to histone tails can alter affinity between DNA and histone, thus can affect the easiness of DNA to be transcribed. Fortunately, histone is protein, thus we can make use of the technology invented for the identification of TFBS. We can measure which histone modification is enriched in which place along DNA. In contrast to the methylation, histone modification has more than twenty variations dependent upon which small molecule binds and where in histone tail it binds to, some of which contribute to expression of genes while others of which interrupt the transcription. Thus, complicated competition between multiple histone modifications can contribute to regulation of gene expression in the multiple ways.

Chromatin Structure

The third popularly measured epigenetics is chromatin structure itself. Chromatin is a complex composed of DNA and histone. As mentioned in the above, although histone modification can alter chromatin structure, histone modification is not an only factor affecting chromatin structure. Thus, chromatin structure can be an independent factor that can affect regulation of gene expression. In contrast to the promoter methylation and histone modification, there are no de facto standard measurement methodology of chromatin structure. It is measured by wide range of technologies.

Post-Transcriptional Regulation

Although all of these above are pre-transcriptional regulation of gene expression, alternatively, some post-transcriptional regulation of gene expression is also often considered.

miRNA Regulation of Gene Expression

miRNA regulation of gene expression is ever the most popular mechanism of post-transcriptional regulation of gene expression. miRNA is non-coding RNA whose length is as long as about 22 nucleotide. Their seed region composed of eight nucleotide binds to complementary polynucleotide region within 3 prime untranslated region of mRNA with the help of protein machinery. miRNAs that bind to target mRNA either degradate mRNAs or interrupt translation of mRNAs. The number of miRNAs is as many as 2000 and their regulation of gene expression is highly context dependent. Thus, in spite of importance within the field of disease and developmental researches, their functionality is not yet fully understood. No effective measurements of miRNA regulation of gene expression are yet proposed. The inference of miRNA-mRNA interaction from bioinformatics is still under the development stage.

Others

Epitranscriptomics is another field of post-transcriptional regulation of gene expression and come to be collecting researchers' interest recently. Epitranscriptomics is mainly studying RNA editing that affects mRNA functions. Although it is obvious that epitranscriptomics plays critical roles in regulation of gene expression, the comprehensive investigations of these functions has just started (Xiong *et al.*, 2017). Epitranscriptomics is composed of many kinds of modification of RNA; some examples are N¹-methyladenosine (m¹A), pseudouridine (Ψ), 5-methylcytidine (m⁵C), and N⁶-methyladenosine (m⁶A). They affect RNA in the various ways, e.g., affecting RNA structure, reducing mRNA stability, enhancing mRNA translation, and so on. At the moment, it is difficult to infer using bioinformatics the effect of epitranscriptomics toward regulation of gene expression.

Long non-coding RNA (lncRNA) is also an important factor that can regulate gene expression. lncRNA is defined as non-coding RNA longer than 200 nucleotide. It is known to contribute to modification of chromatin structure (Bhmdorfer and Wierzbicki, 2015) or to function as miRNA sponge (Wang *et al.*, 2015a). At the moment, it is difficult to relate lncRNA to regulation of gene expression directly. Thus, investigation in bioinformatics along this line is also under the development stage.

Circular RNA (circRNA) (Szabo and Salzman, 2016) can also work as miRNA sponge (Wang *et al.*, 2015a). CircRNA is a different class of non-coding RNA which is expected to play critical roles in regulation of gene expression. Nevertheless, its functions remains not well known. As shown by name, circRNA forms circular structure that is distinct from other non-coding RNA. Although there are some proposed data bases (Chen *et al.*, 2016; Glažar *et al.*, 2014) for circRNA, main study of circRNA is at the moment still their identification. Although the investigation of their functions is waited, it is not yet fully understood.

RNA structure was also started to be considered as contributors to regulation of gene expression (Jacobs *et al.*, 2012). This means, even secondary and/or tertiary structures of mRNAs can contribute to gene expression in post-transcriptional manner. In the study of RNA structure, although theoretical inference of secondary structures of RNA has ever developed, the understanding of its relationship to biological function remains not well studied. The possible way by which RNA structure can contribute to post-transcriptional process is to affect splicing. Alternate splicing is important mechanism that generates multiple proteins (isoforms) based on a single DNA sequence. Since splicing is mediated by the interaction of proteins and mature mRNA, structure of mRNA can alter the binding affinity between proteins and mRNA and as a result affect the splicing. RNA structure can also affect translation since translation is also mediated by the binding between mRNA and proteins that perform translation.

As a consequence, we can have distribution along DNA of various factors that might affect regulation of gene expression in the phase of pre-transcription as well as various factors that might affect regulation of gene expression in the phase of post-transcription. The next task, which is also a main topics from the bioinformatics point of views, is how to relate these factors to regulation of gene expression.

Identification of Relationship Between Various Factors and Regulation of Gene Expression

In order to interpret the various factors that might affect regulation of gene expression, we need to relate various factors to gene body. The most frequently employed, but not biologically validated, way is to use transcription start site (TSS). If some factors enriched around TSS, these factors are assumed to regulate expression of the corresponding genes. Typically, 500 bp upstream and 2000 bp downstream around TSS is used, but this range can be altered. Then, some factors enriched around TSS are assumed to regulate differentially gene expression.

Although it is one to one correspondence, more sophisticated ways are popular in now a day. For example, inference of gene regulatory network (Macneil and Walhout, 2011) (GRN) is a useful strategy. GRN is network representation of regulation of gene expression. Multiple factors are connected with multiple genes. This structure itself expresses the complex regulatory relation between gene expression and multiple factors. There are multiple ways to construct GRN from gene expression profiles, e.g., correlation based and Bayesian based. As for correlation based GRN, correlation between elements (e.g., mRNA and miRNA expression) were employed to evaluate if elements are related. Pairs of elements associated with the correlation coefficients (or their absolute values) larger than the threshold values (or associated with significantly small *P*-values) are connected in GRN and others are not connected. In Bayesian approach, Bayesian network having the maximum posterior probability (or likelihood) was sought. In addition to this, additional external information (e.g., protein-protein interaction or miRNA-mRNA target information) are often considered. After that hub elements that have many connections with other elements are selected as critical elements that can play critical roles in regulation of gene expression.

Measurement Technologies

Microarray

Microarray is an old technology to measure mRNA expression. It is still used for the measurements of various factors that contribute to regulation of gene expression. Methylation arrays is also used for the measurement of DNA methylation. ChIP technology was originally invented for the usage together with microarray. It is yet used for histone modification and TFBS. miRNA and lncRNA expression is even measured using microarray specifically designed for the measurements of miRNA or lncRNA. The advantages of microarray technology is its easiness to use once they are suitably designed. On the other hand, the weakness of this technology is that it is not usable for organisms other than model organisms, e.g. mouse and fly, that are often used for genomic study.

Short Read

High-throughput sequencing (HTS) is a newly invented technology alternative to microarray. Although it is still relatively more expensive than microarray, it has several advantages even for the measurements of factors that affect regulation of gene expression. For example, HTS can be applied to non-model organism while microarray is restricted to model organism to which microarray has already been designed as mentioned above. Although RNA must be converted to DNA before HTS, since HTS can directly count the number of DNA/RNA fragments, HTS is believed to be also more quantitative than microarray. If HTS is binded to ChIP technology invented for ChIP-chip technology for microarray, ChIP-seq technology can be used for the measure histone modification and TFBS. If HTS is combined with bisulfite treatments, HTS can also be used for identification DNA methylation. The

disadvantage of short read technology is that short read must be mapped to genome that is not always available. If genome sequence is missing, genome must be assembled independently prior to mapping short reads.

Long Read

Long read, which was invented by companies like nanopore (see Relevant Websites Section) or PacBio (see Relevant Websites Section), is the next generation HTS. In contrast to short read technology whose length is limited to up to a few hundreds nucleotide, long read technology can measure read as long as a few thousand nucleotides. At the moment, although long read technology is rarely used for studying regulation of gene expression, it has potential to open a new era for the research of regulation of gene expression. For example, long read has a potential to identify multiple factors binding to DNA illustrated in [Fig. 1](#) simultaneously. This might enable us to study interplay between these factors more accurately. Alternatively, long read technology can detect isoforms ([Byrne et al., 2017](#)) that short read technology can hardly identify since short fragments generated from distinct isoforms are hardly distinguished by short read technology. Since there is a possibility that distinct isoforms are differently regulated by various factors, employment of long read technology might be critical in order to understand regulation of gene expression more precisely.

Applications

There are some applications aiming these above purposes. Especially, Bioconductor ([Huber et al., 2015](#)) platform provides many ready made and useful packages.

GRN

For example, GeneNetworkBuilder ([Ou et al., 2017](#)) package generates GRN from ChIP-chip/ChIP-seq and expression data. If your data set is time course, GRENITS ([Morrissey, 2012](#)) can be used. CoRegNet ([Nicolle et al., 2015](#)) is another package that infers GRN from gene expression, TFBS and ChIP data set.

TFBS/Histone Modification

TFBSTools ([Tan and Lenhard, 2016](#)) is well made package that can analyze data for the identification of TFBS. Ringo ([Toedling et al., 2007](#)) aims investigation of ChIP-chip microarray. BAC ([Gottardo, 2007](#)) uses a Bayesian hierarchical model to detect enriched regions from ChIP-chip experiments. Starr ([Zacher et al., 2009](#)) aims simple tiling array analysis of Affymetrix ChIP-chip data. Rcade ([Cairns, 2017](#)) aims analysis of ChIP-seq and differential expression. geneXtender ([Khomtchouk et al., 2017](#)) is designed to process histone Modification ChIP-seq data. rMAT ([Cheung et al., 2017](#)) is implementation from MAT program to normalize and analyze tiling arrays and ChIP-chip data. iChip ([Mo, 2012](#)) is Bayesian modeling of ChIP-chip data through hidden Ising models. Using these packages, researchers easily relate gene expression and TF binding and/or histone modifications

Chromatin State

There is a STAN ([Zacher et al., 2014](#)) that infers chromatin state from histone modification as well as DNA methylation. DChIPRep ([Chabbert et al., 2016](#)) also infers chromatin structure from ChIP-seq data set.

DNA Methylation

As for DNA methylation, ChAMP ([Tian et al., 2017](#)) provides us the method of processing methylation array. BPRMeth ([Kapourani, 2017](#)) is another package that can extract methylation profiles from HTS gene expression and row methylation profiles.

RNA Structure

RNAprobR ([Kielpinski et al., 2015](#)) is a package that processes HTS data for the identification of RNA structure.

Gene Expression

In order to investigate regulation of gene expression, it is of course necessary to predict gene expression itself. However, it should have been discussed in other topics. Thus, this is not discussed here. Access other topics.

miRNA and lncRNA

Expression profiles of miRNA and lncRNA are easily obtained by HTS and microarray. Although treatment of miRNA/lncRNA array is similar to that of mRNA arrays, miRNA-seq does differ from RNA-seq aiming to measure mRNA expression (lncRNA-seq can be

treated similarly to RNA-seq). The researcher must use tools designed specifically for miRNA profile. As for miRNA-seq, miRDeep2 (Friedlander *et al.*, 2011) or miRDeep* (An *et al.*, 2012) are popular tools. Unfortunately, there are no de facto standard methods that relate miRNA expression to mRNA expression, although some tools are available (Wang *et al.*, 2015b) in Bioconductor. First of all, we need to know which mRNAs individual miRNAs target. Nevertheless, mRNA-miRNA interaction is highly context dependent. Even the most trustable experimentally validated miRNA-mRNA interaction database, miRTarBase (Chou *et al.*, 2015), is only valid for the diseases form which experimental results were collected. Thus, at the moment, although miRNA is believed to play critical roles in regulation of gene expression, it is practically difficult to relate miRNA expression to mRNA expression.

Illustrative Examples

In order to make easier to understand how one can process data sets, some illustrative examples are presented in the below. Since most of packages included in Bioconductor are accompanied with more examples, see documents attached with these packages for more examples.

Methylation Analysis Using ChAMP

Suppose R (R Core Team, 2017) is installed. In order to run with sample data sets, try Fig. 2 in R. Then the list of genes associated with differential methylation regions should be in my DMR. In order to list genes, try Fig. 3. You can get a list of chromosomal regions, associated *P*-values, and genes having overlapping promoters. The sample data set is 450 k lung tumor data set and contains only 8 samples, 4 lung tumor samples and 4 control samples.

State Identification Using STAN

STAN enables us to identify chromosomal state with considering histone modification as well as methylation profiles. In order to run with sample data sets, try Fig. 4 in R. In order to see results try Fig. 5. Now, chromosomal regions are classified into ten classes with considering histone modification as well as methylation profiles. The sample data set contains ChIP-Seq experiments of seven histone modifications (H3K4me1, H3K4me3, H3K36me3, H3K27me3, H3K27ac and H3K9ac), as well as DNase-Seq and genomic input.

```
source("https://bioconductor.org/biocLite.R")
biocLite("ChAMP")
testDir=system.file("extdata",package="ChAMPdata")
myLoad <- champ.load(testDir,arraytype="450K")
champ.process(directory = testDir)
```

Fig. 2 Installation and execution of ChAMP.

```
head(myDMR$DMRcateDMR)
##      seqnames      start      end width strand no.cpgs      minfdr
## DMR_1      chr2 177021702 177030228 8527      *      48 3.243739e-69
## DMR_2      chr2 177012117 177017797 5681      *      33 1.137990e-59
## DMR_3     chr12 115134148 115136308 2161      *      51 3.571093e-108
## DMR_4     chr11 31820441 31828715 8275      *      40 3.087712e-43
## DMR_5     chr17 46655289 46658170 2882      *      27 1.565729e-85
## DMR_6      chr6 32115964 32118811 2848      *      50 5.055382e-123
##      Stouffer maxbetafc meanbetafc
## DMR_1 4.011352e-24 0.6962753 0.4383726
## DMR_2 6.555658e-20 0.6138975 0.4676655
## DMR_3 5.711951e-19 0.5372083 0.3601741
## DMR_4 1.206741e-18 0.5673115 0.4112588
## DMR_5 2.087453e-17 0.6859260 0.4812521
## DMR_6 7.216567e-16 0.5772168 0.2971849
##      overlapping.promoters
## DMR_1 HOXD3-001, HOXD3-004, RP11-387A1.5-001
## DMR_2 HOXD4-001, MIR10B-201, HOXD3-005
## DMR_3
## DMR_4 PAX6-016, PAX6-006, PAX6-014, PAX6-015, PAX6-007, PAX6-028, PAX6-025, PAX6-027, PAX6-029, PAX6-024, PAX6-026
## DMR_5 HOXB4-001, MIR10A-201, HOXB3-009, HOXB3-005, MIR10A-001
## DMR_6 PRRT1-002, PRRT1-201, PRRT1-007, PRRT1-006, PRRT1-003
```

Fig. 3 The results of ChAMP.

```

source("https://bioconductor.org/biocLite.R")
biocLite("STAN")
## Loading library and data
library(STAN)
data(trainRegions)
data(pilot.hg19)
## Model initialization
hmm_nb = initHMM(trainRegions[1:3], nStates=10, "NegativeBinomial")
## Model fitting
hmm_fitted_nb = fitHMM(trainRegions[1:3], hmm_nb, maxIters=10)
## Calculate state path
viterbi_nb = getViterbi(hmm_fitted_nb, trainRegions[1:3])
## Convert state path to GRanges object
viterbi_nb_gm12878 = viterbi2GRanges(viterbi_nb, pilot.hg19, 200)

```

Fig. 4 Installation and execution of STAN using sample data.

```

viterbi_poilog_gm12878
## GRanges object with 381 ranges and 1 metadata column:
## seqnames ranges strand | name
## <Rle> <IRanges> <Rle> | <character>
## [1] chr1 [151158001, 151159201] * | 6
## [2] chr1 [151159201, 151160801] * | 1
## [3] chr1 [151160801, 151161001] * | 2
## [4] chr1 [151161001, 151161601] * | 3
## [5] chr1 [151161601, 151162801] * | 4
## ... ..
## [377] chr11 [2324601, 2326601] * | 10
## [378] chr11 [2326601, 2327601] * | 1
## [379] chr11 [2327601, 2328001] * | 2
## [380] chr11 [2328001, 2328201] * | 1
## [381] chr11 [2328201, 2349401] * | 10
## -----
## seqinfo: 3 sequences from an unspecified genome; no seqlengths

```

Fig. 5 The results of STAN.

Discussion

In the above, multiple factors that contribute to regulation of gene expression, technologies that measure these factors and bioinformatics tools to analyze these factors were discussed. Investigation of these factors are still on going and additional new factors that contribute to regulation of gene expression might appear. As more data accumulate and more technology develop, the manner of relating these factors to regulation of gene expression will become more complicated. Thus, the researchers aiming analysis of regulation of gene expression need to try to continuously collect updated knowledge. Even when simply more data become available, the conventional views listed in the above might be altered. The study of regulation of gene expression is on going and just started field. If there would be the revised version of this topics in the future, contents will be altered drastically.

Future Direction

Future direction of this topics is definitely the integration of multiple omics data set, i.e., trans omics analysis (Yugi *et al.*, 2016). As can be seen in the above multiple factors contribute to regulation of gene expression with interacting with one another. In spite of that, most of researches currently performed is mainly on regulation of gene expression with considering individual factors.

In spite of serious need along this direction, no established methods on trans omics analysis exist. Some promising direction includes network based (Yugi *et al.*, 2016), gene set enrichment analysis based (Zhang *et al.*, 2015), and tensor based (Taguchi, 2017) ones. Nevertheless, the best-fitted method for trans omics analysis is still continuously sought.

Concluding Remarks

In this article, various factors that might affect regulation of genes were discussed together with measurements technology and software that can process these measurements. Although it was not yet employed for the study of regulation of gene expression, single cell sequencing technology must be employed since regulation of gene expression differs from cell to cell. After the success of this direction, we might understand cellular function that manages regulation of gene expression more precisely.

See also: Codon Usage. Epigenetics: Analysis of Cytosine Modifications at Single Base Resolution. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Transcriptome Analysis. Whole Genome Sequencing Analysis

References

- An, J., Lai, J., Lehman, M.L., Nelson, C.C., 2012. miRDeep*: An integrated application tool for miRNA identification from RNA sequencing data. *Nucleic Acids Research* 41, 727–737. doi:10.1093/nar/gks1187. Available at: <https://doi.org/10.1093%2Fnar%2Fgks1187>.
- Bhmdorfer, G., Wierzbicki, A.T., 2015. Control of chromatin structure by long noncoding RNA. *Trends in Cell Biology* 25, 623–632. doi:10.1016/j.tcb.2015.07.002. Available at: <https://doi.org/10.1016%2Fj.tcb.2015.07.002>.
- Byrne, A., Beaudin, A.E., Olsen, H.E., et al., 2017. Nanopore long-read RNAseq reveals widespread transcriptional variation among the surface receptors of individual b cells. Available at: <https://doi.org/10.1101%2F126847>.
- Cairns, J., 2017. Rcade: A tool for integrating a count-based chip-seq analysis with differential expression summary data. Available at: <https://www.bioconductor.org/packages/release/bioc/html/Rcade.html>. R package version 1.18.0.
- Chabbert, C.D., Steinmetz, L.M., Klaus, B., 2016. DChIPRep, an R/bioconductor package for differential enrichment analysis in chromatin studies. *PeerJ* 4, e1981. doi:10.7717/peerj.1981. Available at: <https://www.bioconductor.org/packages/release/bioc/html/DChIPRep.html>.
- Chen, X., Han, P., Zhou, T., et al., 2016. circRNADb: A comprehensive database for human circular RNAs with protein-coding annotations. *Scientific Reports* 6. doi:10.1038/srep34985. Available at: <https://doi.org/10.1038%2Fsrep34985>.
- Cheung, C., Droit, A., Gottardo, R., 2017. rMAT: R implementation from MAT program to normalize and analyze tiling arrays and ChIP-chip data. Available at: <https://www.bioconductor.org/packages/release/bioc/html/rMAT.html>. R package version 3.26.0. Available at: <http://www.rglab.org>.
- Chou, C.H., Chang, N.W., Shrestha, S., et al., 2015. miRTarBase 2016: Updates to the experimentally validated miRNA-target interactions database. *Nucleic Acids Research* 44, D239–D247. doi:10.1093/nar/gkv1258. Available at: <https://doi.org/10.1093%2Fnar%2Fgkv1258>.
- Friedlander, M.R., Mackowiak, S.D., Li, N., Chen, W., Rajewsky, N., 2011. miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. *Nucleic Acids Research* 40, 37–52. doi:10.1093/nar/gkr688. Available at: <https://doi.org/10.1093%2Fnar%2Fgkr688>.
- Glažar, P., Papavasiliou, P., Rajewsky, N., 2014. circBase: A database for circular RNAs. *RNA* 20, 1666–1670. doi:10.1261/rna.043687.113. Available at: <https://doi.org/10.1261%2Frna.043687.113>.
- Gottardo, R., 2007. BAC: Bayesian Analysis of Chip-Chip Experiment. Available at: <https://www.bioconductor.org/packages/release/bioc/html/BAC.html>. R package version 1.36.0.
- Huber, W., Carey, V.J., Gentleman, R., et al., 2015. Orchestrating high-throughput genomic analysis with bioconductor. *Nature Methods* 12, 115–121. doi:10.1038/nmeth.3252. Available at: <https://www.bioconductor.org/>.
- Jacobs, E., Mills, J.D., Janitz, M., 2012. The role of RNA structure in posttranscriptional regulation of gene expression. *Journal of Genetics and Genomics* 39, 535–543. doi:10.1016/j.jgg.2012.08.002. Available at: <https://doi.org/10.1016%2Fj.jgg.2012.08.002>.
- Kapourani, C., 2017. BPRMeth: Model higher-order methylation profiles. Available at: <https://www.bioconductor.org/packages/release/bioc/html/BPRMeth.html>. R package version 1.2.0.
- Khomtchouk, B.B., Van Booven, D., Wahlestedt, C., 2017. Genextender: Optimized functional^o annotation of chip-seq data. *bioRxiv*. doi:10.1101/082347. Available at: <https://www.bioconductor.org/packages/release/bioc/html/geneXtender.html>.
- Kiepiński, L.J., Sidiropoulos, N., Vinther, J., 2015. Reproducible analysis of sequencing-based RNA structure probing data with user-friendly tools. *Methods in Enzymology*. Elsevier. pp. 153–180. doi:10.1016/bs.mie.2015.01.014. Available at: <https://www.bioconductor.org/packages/release/bioc/html/RNAProb.html>.
- Macneil, L.T., Walhout, A.J., 2011. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome Res.* 21, 645–657.
- Mo, Q., 2012. iChip: Bayesian Modeling of ChIP-chip Data Through Hidden Ising Models. Available at: <https://www.bioconductor.org/packages/release/bioc/html/iChip.html>. R package version 1.30.0.
- Morrissey, E., 2012. GRENITS: Gene Regulatory Network Inference Using Time Series. Available at: <https://www.bioconductor.org/packages/release/bioc/html/GRENITS.html>. R package version 1.28.1.
- Nicolle, R., Radvanyi, F., Elati, M., 2015. CoReg-Net: Reconstruction and integrated analysis of co-regulatory networks. *Bioinformatics* 31, 3066–3068. doi:10.1093/bioinformatics/btv305. Available at: <https://www.bioconductor.org/packages/release/bioc/html/CoRegNet.html>.
- Ou, J., Liu, H., Tissenbaum, H.A., Zhu, L.J., 2017. GeneNetworkBuilder: Build Regulatory Network From ChIP-chip/ChIP-seq and Expression Data. Available at: <https://www.bioconductor.org/packages/release/bioc/html/GeneNetworkBuilder.html>. R package version 1.18.1.
- R Core Team, 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. Available at: <https://www.R-project.org/>.
- Senda, T., Adachi, N., 2013. Post-translational modifications, histone. *Encyclopedia of Systems Biology*. New York: Springer, p. 896. doi:10.1007/978-1-4419-9863-7_100627. Available at: https://doi.org/10.1007%2F978-1-4419-9863-7_100627.
- Szabo, L., Salzman, J., 2016. Detecting circular RNAs: Bioinformatic and experimental challenges. *Nature Reviews Genetics* 17, 679–692. doi:10.1038/nrg.2016.114. Available at: <https://doi.org/10.1038%2Fnrg.2016.114>.
- Taguchi, Y.H., 2017. Tensor decomposition-based unsupervised feature extraction applied to matrix products for multi-view data processing. *PLOS ONE* 12, e0183933. doi:10.1371/journal.pone.0183933. Available at: <https://doi.org/10.1371%2Fjournal.pone.0183933>.
- Tan, G., Lenhard, B., 2016. TFBSTools: An R/bioconductor package for transcription factor binding site analysis. *Bioinformatics* 32, 1555–1556. doi:10.1093/bioinformatics/btw024. Available at: <https://www.bioconductor.org/packages/release/bioc/html/TFBSTools.html>.
- Tian, Y., Morris, T.J., Webster, A.P., et al., 2017. ChAMP: Updated methylation analysis pipeline for illumina BeadChips. *Bioinformatics*. doi:10.1093/bioinformatics/btx513. Available at: <https://www.bioconductor.org/packages/release/bioc/html/ChAMP.html>.
- Toedling, J., Skylar, O., Krueger, T., et al., 2007. Ringo an R/Bioconductor package for analyzing ChIP-chip readouts. *BMC Bioinformatics* 8, 443. doi:10.1186/1471-2105-8-443. Available at: <https://www.bioconductor.org/packages/release/bioc/html/Ringo.html>.
- Wang, P., Zhi, H., Zhang, Y., et al., 2015a. miRSponge: A manually curated database for experimentally supported miRNA sponges and ceRNAs. *Databases* 67e5 2015, bav098. doi:10.1093/database/bav098. Available at: <https://doi.org/10.1093%2Fdatabase%2Fbav098>.

- Wang, T., Lu, T., Lee, C., *et al.*, 2015b. anamiRan integrated analysis package of miRNA and mRNA expression. Available at: <https://www.bioconductor.org/packages/release/bioc/html/anamiR.html>. R package version 1.4.2.
- Xiong, X., Yi, C., Peng, J., 2017. Epitranscriptomics: Toward a better understanding of RNA modifications. *Genomics, Proteomics & Bioinformatics* 15, 147–153. doi:10.1016/j.gpb.2017.03.003. Available at: <https://doi.org/10.1016%2Fj.gpb.2017.03.003>.
- Yugi, K., Kubota, H., Hatano, A., Kuroda, S., 2016. Trans-omics: How to reconstruct biochemical networks across multiple 'omic' layers. *Trends in Biotechnology* 34, 276–290. doi:10.1016/j.tibtech.2015.12.013. Available at: <https://doi.org/10.1016%2Fj.tibtech.2015.12.013>.
- Zacher, B., Lidschreiber, M., Cramer, P., Gag-neur, J., Tresch, A., 2014. Annotation of genomics data using bidirectional hidden markov models unveils variations in pol II transcription cycle. *Molecular Systems Biology* 10, 768. doi:10.15252/msb.20145654. Available at: <https://www.bioconductor.org/packages/release/bioc/html/STAN.html>.
- Zacher, B., Soeding, J., Kuan, P., Siebert, M., Tresch, A., 2009. Starr: Simple tiling array analysis of Affymetrix ChIP-chip data. Available at: <https://www.bioconductor.org/packages/release/bioc/html/Starr.html>. R package version 1.32.0.
- Zhang, F., Xiao, X., Hao, J., *et al.*, 2015. CPAS: A trans-omics pathway analysis tool for jointly analyzing DNA copy number variations and mRNA expression profiles data. *Journal of Biomedical Informatics* 53, 363–366. doi:10.1016/j.jbi.2014.12.012. Available at: <https://doi.org/10.1016%2Fj.jbi.2014.12.012>.

Further Reading

- Bioinformatics, *Methods in Molecular Biology* 1526/1527. In: Keith, J.M. (Ed.), Springer Protocols, Volume I: Data, Sequence Analysis, and Evolution, Volume II: Structure, Function, and Applications, second ed. New York: Humana Press/Springer. ISBN: 978-1-4939-6620-2/978-1-4939-6611-0.
- Bioinformatics in MicroRNA Research. In: Huang, J., Borchert, G.M., Dou, D., *et al.* (Eds.), *Methods in Molecular Biology* 1617. New York: Springer Protocols, Humana Press/Springer. ISBN: 978-1-4939-7044-5.
- Cho, W.C.S. (Ed.), 2011. *MicroRNAs in Cancer Translational Research*. New York: Springer. ISBN: 978-94-007-0297-4.
- Computational Methods for Next Generation Sequencing Data Analysis. In: Mandouli, I.I., Zelikovsky, A. (Eds.), *Wiley Series on Bioinformatics: Computational Techniques and Engineering*. Hoboken: Wiley. ISBN: 978-1-118-16948-3.
- Epigenetic Contributions in Autoimmune Disease. In: Ballestar, E. (Ed.), *Advances in Experimental Medicine and Biology*. New York: Springer. ISBN: 978-1-4419-8215-5.
- Epigenetic Alterations in Oncogenesis. In: Karpf, A.R. (Ed.), *Advances in Experimental Medicine and Biology* 754. New York: Springer. ISBN 978-1-4419-9966-5.
- Khalil, A.M., Collier, J. (Eds.), 2013. *Molecular Biology of Long Non-Coding RNAs*. New York: Springer. ISBN: 978-1-4614-8620-6.
- Mallick, B., Ghosh, Z. (Eds.), 2012. *Regulatory RNAs, Basics, Methods and Applications*. New York: Springer. ISBN: 978-3-642-22516-1.
- RNA Bioinformatics, *Methods in Molecular Biology*. In: Picardi, E. (Ed.), Springer Protocols 1269. New York: Springer. ISBN: 978-1-4939-2290-1.
- Statistical Analysis of Next Generation Sequencing Data. In: Datta, S., Nettleton, D. (Eds.), *Frontiers in Probability and the Statistical Sciences*. New York: Springer. ISBN 978-3-319-07211-1.
- Wu, W. (Ed.), *MicroRNA and Cancer*, *MicroRNA and Cancer*, *Methods Humana Press/Springer*, New York, 2011, ISBN 978-1-60761-862-1.
- Yousef, M., Allmer, J. (Eds.) *miRNomics*, *Methods in Molecular Biology* 1107, Springer Protocols, Humana Press/Springer, New York, 2014, ISBN 978-1-62703-747-1.

Relevant Websites

- <http://www.genecards.org>
GeneCards.
- <http://www.mirbase.org>
miRBase.
- <https://nanoporetech.com/>
Nanopore.
- <https://www.ncbi.nlm.nih.gov/geo/>
NCBI.
- <https://www.ncbi.nlm.nih.gov/pubmed>
NCBI.
- <https://www.ncbi.nlm.nih.gov/sra>
NCBI.
- <http://www.pacb.com/>
Pacbio.

Biographical Sketch

Prof. Y-h. Taguchi is a physics professor at Department of Physics, Chuo University, Tokyo, Japan. He has obtained Dr. Sci. at 1988 for Statistical Physics, from Tokyo Institute of Technology, Japan. He has been an assistant Professor at Department of Physics, Tokyo Institute of Technology, 1988. Then, he has move to the present position as an associate professor at 1997. He has been a full professor since 2006. He has published more than 90 peer reviewed and conference proceedings papers including 28 single authored papers. His main present interests are feature extraction and feature selection using principal component analysis and tensor decomposition. His methods were applied to transcriptome and epigenetic profiles, which were further applied to in silico drug discovery as well as biomarker identification.

Comparative Transcriptomics Analysis

Y-h Taguchi, Chuo University, Tokyo, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Comparative Transcriptomics Analysis (CTA) is, as the name implies, the comparisons of the amount of transcripts between two distinct conditions. Typical distinct conditions are healthy controls vs disease patients, a pair of distinct tissues, distinct time points and even distinct species. The reason for this frequent execution of CTA is because almost all cells keep the full set of genes, although not all of the genes are expressed. Because of this sharing of the full set of genes between cells, CTA is a primary method to investigate the factors that make cells distinct from one another.

In contrast to the clearer purpose of CTA, performing CTA is much harder. This is primarily because the amount of transcripts is naturally a real number. Since real numbers cannot be exactly equal to one another by chance, we need some criterion when the amounts of two transcripts can be regarded to differ from each other. The most simplest solution of this problem is the introduction of some statistical models. Based upon the statistical models under the null hypothesis that the amounts of two transcripts are equal, we can identify transcripts associated with distinct values between two conditions by rejecting the null hypothesis. Thus, the point is how to assume the null hypothesis with which statistical models are assumed.

Dependent upon assumed statistical models, various strategies of CTA are possible. They are primarily divided to two categories dependent upon what kind of technology is used to measure the amount of transcript. The major two such technologies are microarray and high throughput sequencing (HTS), respectively. Although these two technologies measure biologically the same variables, i.e., the amount of transcripts, the outcomes differ from each other. In microarray, the amounts of transcripts measured are inevitably real numbers, since microarray measures the amount of transcripts by the amount of light emission from nucleotide fragments that bind to complementary probes. On the other hands, the amount of transcript measured by HTS must be integer, since HTS counts the number of fragments cut out from mRNA. Since suitable statistical models for real numbers and integer differ, in the following, I separately discuss the basics and the applications of the two.

Statistical Models

Microarray

As described in the above, statistical models applied to the amount of transcripts measured by microarray must be that of real numbers. Most of popular statistical models assume that the amount of the transcript obeys t distribution, $t(n)$, where n is the number of samples in each class. When aiming to identify genes that have distinct mean μ between two classes, the null hypothesis that μ is equivalent between two classes is rejected with some P -value that is the threshold value, e.g., 0.05.

t -test is only applicable to two classes problem. Although t -test can be formally applied to multiple classes by dividing multiple classes to a set of pairwise comparison, it is erroneous because of the following reason. Suppose we have K multiple classes. Then, the number of pairwise comparisons is as many as $K(K-1)/2$. Thus, $P=2/K(K-1)$ can be achieved by chance. If K is as large as ten, since $P=2/K(K-1) \simeq 0.02$, usual threshold value $P=0.05$ is clearly meaningless. If P -values are corrected, e.g., $P<0.05 \times 2/K(K-1)$, the ability that t -test can detect genes associated with distinct amount of transcript between any pairs of multiple classes will drastically decrease.

In order to avoid these problems, for the CTAs associated with multiple classes, other strategies are employed. These other strategies include categorical regression (in other words, analysis of variance (ANOVA) (Upton and Cook, 2008)), χ^2 test, or any other extensions of them. Details of implementations differ from applications to applications, it will be discussed in the below in the applications section.

HTS

HTS is an alternate technology that can outperform microarray that has many limitations. Microarray must be designed prior to measurements. This means, non-model organisms are hard to be tested. In HTS, we do not need anything prior to the measurements, although genome sequence must be decided if it is not available prior to counting of the number of reads attributed to each gene. Genome sequence also can be decided by HTS.

HTS is roughly divided to two classes: genome sequencing and RNA-seq. In genome sequencing, fragmented genomic DNA are sequenced and whole genome is assembled from the reads sequence. On the other hand, RNA-seq tries to sequence reads taken from RNAs. For both cases, read can be single end or paired ends. For the latter, reads are generated from both ends of longer fragmented DNA or RNA. This strategy can increase the accuracy than the single end, because paired ends require additional constraint that paired end must be within the length of mRNA and fragmented DNA.

Assembling fragmented DNA to get whole genome sequence and transforming RNA-seq reads into transcript with consideration of splicing are the issues of statistical modeling. In other words, they are the tasks that decide the most probable genome sequence or mapping of RNA-seq reads to genome, based on the obtained read sequences. Since the measurement of the amount of transcript itself is out of scope of this article, refer to other articles for more details.

In contrast to the conventional HTS, where length of read sequences is limited to up to a few hundreds nucleotide, an alternative technology, long read sequencing (e.g., nanopore and PacBio, see Relevant Websites section), is recently gaining traction. In long read technology, each read can be as long as a few thousand nucleotides, which is long enough to measure whole genome of prokaryotes and individual whole transcript of eukaryotes. This has several advantages compared with short read technologies. For prokaryotes genome, it is obvious that long read technology allows to omit assembly in order to obtain whole genome sequences. For eukaryote transcripts, complicated splice junction identification process can be omitted. Thus, short read technology soon will be completely replaced with long read technologies.

Normalization

Prior to CTA, the amount of transcript must be normalized, since independent of the measurement technologies, microarray or HTS, the total amount of transcript is impossible to control during measurement. This process can be done either outside of CTA or as a part of CTA. In the former case, researchers can select their favorable methods while the latter employs the pre-defined strategy for normalization. In any case, dependent on the normalization strategy, the outcome of CTA varies and there are no de facto standard techniques for the normalization prior to CTA. This is an additional reason why CTA is difficult to perform, since incorrect normalization might result in the miss-identification of genes expressed distinctly between two conditions. Various applications aiming normalization prior to CTA will be introduced in applications section below.

Fold Change

Even if some genes can be identified as expressed differently between two conditions based on the criterion given by a statistical test, some genes might likely not biologically play critical roles. This is because the standard statistical tests often ignore the amount of transcripts itself. For example, in *t*-test, very small difference of μ s between two classes can be judged as significant when estimated standard deviation in each class is much smaller than the estimated difference of μ s between two classes. In extreme case, when the amounts of transcript in each class are completely same within each class, i.e., standard deviation is equal to zero, any tiny difference of μ s between two classes can be regarded as significant, since $P=0$. Nevertheless, such a judgement is clearly meaningless from the biological point of views (Conversely, genes not associated with significant difference may biologically function. This often simply means inability of tests because of, e.g., not enough number of available samples).

In order to avoid these somewhat meaningless identifications of genes associated with distinct amount of transcript between distinct classes, fold change is often considered together with statistical tests. Fold change is often useful to screen genes associated with differential expression when statistical test cannot list any genes with significantly smaller *P*-values. Fold change is, as its name says, the ratio between the amount of transcripts in two classes. Then, genes associated with both significantly small *P*-values and large enough fold change (e.g., larger than two or less than a half) are regarded to be those associated with significant difference of amount of transcript between two classes. Although this strategy is known to work well empirically, there are no systematic ways that can decide how large or small fold change (larger than twice, larger than three times, smaller than one half, or smaller than one third) should be employed in order to get biologically reasonable answers.

Multiple Comparisons

Multiple comparison is another issue of CTA. Any statistical test can attribute to each gene, *P*-values that reject null hypothesis that two classes are equivalent in some sense. Nevertheless, since the number of genes is huge (c.a. 10^4), these *P*-values are misleading. If the number of genes is *N*, $P=1/N$ can happen by chance. This suggests that the frequently used criterion that $P<0.05$ is significant is useless.

At the moment, although there are no definite ways to address this problem, numerous trials were proposed. The simplest one is Bonferroni (Armstrong, 2014) where *P* is transformed to be *NP*. This apparently simple transformation has drawback that often misses the significant difference because of its too strict criterion. More robust way is to assume the uniform distribution for *P*-values. If null hypothesis is true, *P*-values must obey uniform distribution. Then, genes whose attributed *P*-values deviate from uniform distribution is regarded to be significant. The most frequently used criterion along this line is Benjamin-Hochberg (Benjamini and Hochberg, 1995). Empirically, the latter strategy was more often employed, since it can give us more biologically reasonable (interpretable) results empirically.

Batch Effect

Although it is somewhat related to normalization issue, batch effect can often heavily affect the outcomes of CTA. Batch effects generally means the uncontrollable external factors that can affect measured amount of transcript. Numerous factors can cause batch effects, e.g.,

days, time, institutes, persons, and so on. If batch effect is not removed effectively, CTA results in not the comparison between treated and control samples, but that among batches. Although there are many packages proposed for removing batch effects (Reese *et al.*, 2013; Leek, 2014; Chen *et al.*, 2011), there are no de facto standard methods that can remove batch effect well independent of the experimental situations. The best strategy that researchers can employ is to evaluate outcomes with considering biological significance of outcomes carefully. In principal, it is dangerous to mix gene expression profiles taken from different batches.

Enrichment Analysis

Although it is not directly related to CTA, biological interpretation of selected genes is often required and important. One of typical analysis for this purpose is enrichment analysis. In enrichment analysis, a set of genes is given, and various biological terms associated with the gene set with significantly small *P*-values computed by some statistical test, e.g., Fisher's exact test, are identified.

Here I list some of famous servers to be used for this purpose: DAVID (Huang *et al.*, 2008), g:profiler (Reimand *et al.*, 2016), TargetMine (Chen *et al.*, 2016), Enricher (Kuleshov *et al.*, 2016), IPA (QIAGEN, 2017) and MSigDB (Subramanian *et al.*, 2005), although they have their own pros and cons.

qPCR

Other than two major technologies, microarray and HTS, quantitative polymerase chain reaction (qPCR) is also sometimes used to measure amount of transcript. In spite of the quantitative accuracy of qPCR compared with other two technologies, qPCR is used only less frequently. This is because qPCR has no ability to measure numerous transcripts simultaneously. For each transcript, experiments must be repeated independently. This is far from cost effective. In addition to this, qPCR often requires reference transcript that is not altered between treated and control samples. Since either identification of not altered transcript or artificial spike in of reference genes are additional time and cost consuming process, qPCR is mainly used for validating the limited number of transcripts among all transcripts measured by either microarray or HTS.

Single Cell Analysis

Although single cell analysis (SCA) (Yuan *et al.*, 2017), which measures transcripts cell by cell, is a rising field, it is not developed enough to be included in this encyclopedia as an established field. First of all, because of technology currently developing, some transcripts are often missing. At the moment, there are no ways to judge if missing transcript is really missing (biologically) or not (technologically missing, i.e., failure of measurements). Thus, the purpose of SCA is often presently not identifying genes whose transcripts are expressed distinctly between treated and control samples, but clustering (grouping) cells based upon the measured transcripts. Since clustering cells is somewhat outside of CTA, SCA is not discussed here in details. Nonetheless, SCA is surely replacing conventional CTA in the future when the SCA technology is established.

Applications

There are numerous applications for CTA. In this article, those in Bioconductor (Huber *et al.*, 2015) will be specifically introduced.

Normalization for Microarray

There are numerous methods to achieve normalization prior to CTA for microarray. There are generally two branches along this direction. The first one is the normalization based upon single microarray. This means that the amount of transcript is normalized with considering single measurement. One of the most frequent methodologies of single array based normalization is mas5, which is implemented as mas5 function, included in affy (Gautier *et al.*, 2004) package. In single microarray based normalization, total amount of transcripts is assumed to be constant, independent of measurements. Another frequent strategy is normalization based upon multiple array. The most popular one along this direction is rma, which is implemented as rma function, also included in affy (Gautier *et al.*, 2004) package. In multiple array based strategy, the amount of transcripts that share ranking among multiple arrays are assumed to take the same values. In actual, there are no ways by which we can judge the better one between these two. Generally speaking, multiple array based strategies are more popular since they are less affected by individual measurements. In principle, the choice of better strategy is highly context dependent. It must be evaluated based upon the biological outcomes.

Normalization for HTS

Although the amount of transcripts obtained by HTS technology is an integer, it has different difficulties than microarray normalization. Since the number of reads are that of RNA fragments, the number of reads mapped to individual genes is not proportional to the amount of transcripts. It is obvious that longer RNA has more fragments. This is a sharp contrast to microarray

where the amount of transcripts measured is per gene base. Thus, there are two kinds of normalization strategy. One is to transform the number of reads mapped to each gene to that of per gene base. The most frequent definition of this line is RPKM (Reads per million mapped reads), which is defined as

$$\text{raw counts} \times \frac{10^6}{\text{all reads}} \times \frac{10^3}{\text{gene length}}$$

where raw counts is the number of reads mapped to each gene, all reads are total number of reads in each measurement. Although it looks reasonable, there is one drawback. When single gene expression drastically increases, because of all reads in denominator, all of other transcripts are regarded as being decreased, although it is clearly not reasonable.

Another strategy is raw reads, which is without any normalization. It might look strange, but raw reads as it is can be treated if suitable statistical models are proposed (see below).

CTA for Real Numbers

Since RPKM is a real number analogous to the amount of transcripts measured by microarray, we discuss these with that of microarray. As for CTA of real numbers, there are huge number of applications. Here I introduce two of them as the most frequently used ones. The first one is SAM (Significance Analysis of Microarrays) (Tusher *et al.*, 2001), which is implemented as sam function in siggenes package (Schwender, 2012). It can deal with both two classes and multiple classes, by employing modified *t*-test and χ^2 test accordingly. Another one is limma (Ritchie *et al.*, 2015), which is based upon linear model assuming Bayesian work frame. limma can also deal with both two classes and multiple classes. In actual, limma can be adapted to almost all situations, since it employs design matrix strategy by which user can designate any kinds of possible comparisons. sam and limma also can give users the adjusted *P*-values which considered multiple comparison criterion. Thus, researchers do not have to consider correction assuming multiple comparisons.

CTA for Integer Numbers

Since reads count by HTS is positive integers, we need null hypothesis fitted to this situation. Although Poisson distribution has long been employed, it has one “drawback”; since Poisson distribution has only one parameter, it cannot be fitted to mean and variance simultaneously. In order to overcome this problem, negative binomial distribution is more often used. DESeq2 (Love *et al.*, 2014) is the most frequently used packages for this purpose. It also accepts raw reads as input and normalization is included in the data processing. It also gives us adjusted *P*-values so as not to consider multiple comparisons separately. It is also fitted to both two classes and multiple classes, since it employ design matrix strategy that limma employs.

In spite of frequent and successful usage of DESeq2 in CTA of HTS, the appearance of negative binomial distribution is not always guaranteed, since it lacks convergence theorem that normal distribution has. Because of this drawback, a non-parametric strategy is sometimes employed. NOISeq (Tarazona *et al.*, 2015) is one of the most frequently used non-parametric packages. NOISeq also implements data normalization, multiple comparison corrections, adapted to both two classes and multiple classes and so on.

CTA often results in distinct outcomes between DESeq2 and NOISeq. As in the microarray, there are not definite criteria that decide the best applications.

Conclusions

In summary, CTA is rather art than science. At the moment, there are no definite ways guaranteed to always work well regardless the situations considered. CTA must be done with much care in order to avoid getting results without any biological meanings. It is not an easy way, but must be tried.

See also: Characterizing and Functional Assignment of Noncoding RNAs. Codon Usage. Comparative Genomics Analysis. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Regulation of Gene Expression. Sequence Analysis. Transcriptome Analysis. Whole Genome Sequencing Analysis

References

- Armstrong, R.A., 2014. When to use the bonferroni correction. *Ophthalmic and Physiological Optics* 34, 502–508. doi:10.1111/opo.12131. Available at: <https://doi.org/10.1111%2Fopo.12131>.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)* 57, 289–300. doi:10.2307/2346101. Available at: <https://doi.org/10.2307/2346101>.

- Chen, C., Grennan, K., Badner, J., *et al.*, 2011. Removing batch effects in analysis of expression microarray data: An evaluation of six batch adjustment methods. PLOS ONE 6, e17238. doi:10.1371/journal.pone.0017238. Available at: <https://doi.org/10.1371%2Fjournal.pone.0017238>.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2016. An integrative data analysis platform for gene set analysis and knowledge discovery in a data warehouse framework. In: Database 2016, baw009. doi:10.1093/database/baw009. Available at: <https://doi.org/10.1093%2Fdatabase%2Fbaw009>.
- Gautier, L., Cope, L., Bolstad, B.M., Irizarry, R.A., 2004. affy-analysis of affymetrix GeneChip data at the probe level. Bioinformatics 20, 307–315. doi:10.1093/bioinformatics/btg405. Available at: <https://doi.org/10.1093%2Fbioinformatics%2Fbtg405>.
- Huang, D.W., Sherman, B.T., Lempicki, R.A., 2008. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nature Protocols 4, 44–57. doi:10.1038/nprot.2008.211. Available at: <https://doi.org/10.1038%2Fnprot.2008.211>.
- Huber, W., Carey, V.J., Gentleman, R., *et al.*, 2015. Orchestrating high-throughput genomic analysis with bioconductor. Nature Methods 12, 115–121. doi:10.1038/nmeth.3252. Available at: <https://www.bioconductor.org/>.
- Kuleshov, M.V., Jones, M.R., Rouillard, A.D., *et al.*, 2016. Enrichr: A comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Research 44, W90–W97. doi:10.1093/nar/gkw377. Available at: <https://doi.org/10.1093%2Fnar%2Fgkw377>.
- Leek, J.T., 2014. svaseq: Removing batch effects and other unwanted noise from sequencing data. Nucleic Acids Research 42. doi:10.1093/nar/gku864. Available at: <https://doi.org/10.1093%2Fnar%2Fgku864>.
- Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology 15. doi:10.1186/s13059-014-0550-8. Available at: <https://doi.org/10.1186%2Fs13059-014-0550-8>.
- QIAGEN, 2017. Ingenuity pathway analysis. Available at: <https://www.qiagenbioinformatics.com/products/ingenuity-pathway-analysis/>.
- Reese, S.E., Archer, K.J., Therneau, T.M., *et al.*, 2013. A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. Bioinformatics 29, 2877–2883. doi:10.1093/bioinformatics/btt480. Available at: <https://doi.org/10.1093%2Fbioinformatics%2Fbtt480>.
- Reimand, J., Arak, T., Adler, P., *et al.*, 2016. g:profiler—A web server for functional interpretation of gene lists (2016 update). Nucleic Acids Research 44, W83–W89. doi:10.1093/nar/gkw199. Available at: <https://doi.org/10.1093%2Fnar%2Fgkw199>.
- Ritchie, M.E., Phipson, B., Wu, D., *et al.*, 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research 43. doi:10.1093/nar/gkv007. Available at: <https://doi.org/10.1093%2Fnar%2Fgkv007>.
- Schwender, H., 2012. siggenes: Multiple testing using SAM and Efron's empirical Bayes approaches. R package version 1.50.0. Available at: <https://bioconductor.org/packages/release/bioc/html/siggenes.html>.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences 102, 15545–15550. doi:10.1073/pnas.0506580102. Available at: <https://doi.org/10.1073%2Fpnas.0506580102>.
- Tarazona, S., Furió-Tarí, P., Turrà, D., *et al.*, 2015. Data quality aware analysis of differential expression in RNA-seq with NOISeq r/bioc package. Nucleic Acids Research. gkv711. doi:10.1093/nar/gkv711. Available at: <https://doi.org/10.1093%2Fnar%2Fgkv711>.
- Tusher, V.G., Tibshirani, R., Chu, G., 2001. Significance analysis of microar-rays applied to the ionizing radiation response. Proceedings of the National Academy of Sciences 98, 5116–5121. doi:10.1073/pnas.091062498. Available at: <https://doi.org/10.1073%2Fpnas.091062498>.
- Upton, G., Cook, I., 2008. A Dictionary of Statistics. Oxford University Press. Available at: <https://doi.org/10.1093%2Ffacref%2F9780199541454.001.0001>.
- Yuan, G.C., Cai, L., Elowitz, M., *et al.*, 2017. Challenges and emerging directions in single-cell analysis. Genome Biology 18. doi:10.1186/s13059-017-1218-y. Available at: <https://doi.org/10.1186%2Fs13059-017-1218-y>.

Relevant Websites

<https://nanoporetech.com/>
 NANOPORE.
<http://www.pacb.com/>
 PACBIO.

Analyzing Transcriptome-Phenotype Correlations

Bryan T Li and Jin X Lim, Temasek Polytechnic, Singapore

Maurice HT Ling, Colossus Technologies LLP, Singapore

© 2019 Elsevier Inc. All rights reserved.

Introduction: Network Effects Between Transcriptome and Phenotype

Consider an electronic musician facing potentially thousands of knobs, switches, and myriad of various controls on his digital instruments and software. Although each dial, such as volume control, can be adjusted by the composer independently but it may be connected to one or more dials. For example, an incremental unit on Dial A may result in 0.1 unit increment to the setting on Dial B and 0.15 unit decrement to the setting on Dial C. Dial B and C may in turn affect other dials, which may then indirectly affect Dial A. The task of the musician is to manipulate the various controls to compose a techno piece.

In this analogy, each control represents a gene and the setting on that control represents the transcription level of that gene. The diversity of controls represents the thousands of genes in the genome. Given that the genome of each cell is identical in an organism (which several exceptions, such as immunoglobulin-producing cells), the musician represents the external stimuli to the cell and the relationships between the controls represent the complex network of signalling cross-talk where the expression of one gene may up-regulate or down-regulate other genes to varying degrees. Hence, how can the same set of controls (genome) result in different musical pieces (phenotype)?

It has been known that changes in the transcriptome may result in phenotypic changes (Sul *et al.*, 2009), some of which may result in diseased conditions; such as oncogenesis (Kang *et al.*, 2009), metabolic effects (Kyung *et al.*, 2017), or other diseases (Casamassimi *et al.*, 2017). However, the determination of phenotype from transcriptome may not be direct and can be affected by multiple levels of modulations (Fig. 1). For example, the effective concentration of the mRNA transcript may be affected by its antisense transcript expression (Zhao *et al.*, 2014b) and half-life (Belgrader *et al.*, 1994), various sequence features on the mRNA transcript may affect transcription efficiency leading to varying protein levels given the same amount of mRNA transcript (Evfratov *et al.*, 2017; Gamble *et al.*, 2016; Hockenberry *et al.*, 2017; Kumar *et al.*, 2014; Lahtvee *et al.*, 2017), and the effective abundance of each protein may be affected by varying half-life (Fishbain *et al.*, 2015).

Although phenotype prediction may be best carried out from proteome, the experimental techniques for large-scale proteomics studies are less developed compared to transcriptomics or genomics studies. A reason is that most transcriptome and genome techniques are based on polymerase chain reaction, which only exist for nucleic acids, and corresponding technique is absent in proteomics studies. Hence, a large volume of transcriptomics studies and data is present. Thus, there are substantial benefits to be able to predict phenotype from its transcriptome.

Determining the Transcriptome

The transcriptome is defined as the complete set of transcripts in a cell, and their quantity, for a specific developmental stage or physiological condition (Wang *et al.*, 2009). Transcripts are RNAs, the single-stranded nucleic acids, which possesses structural and

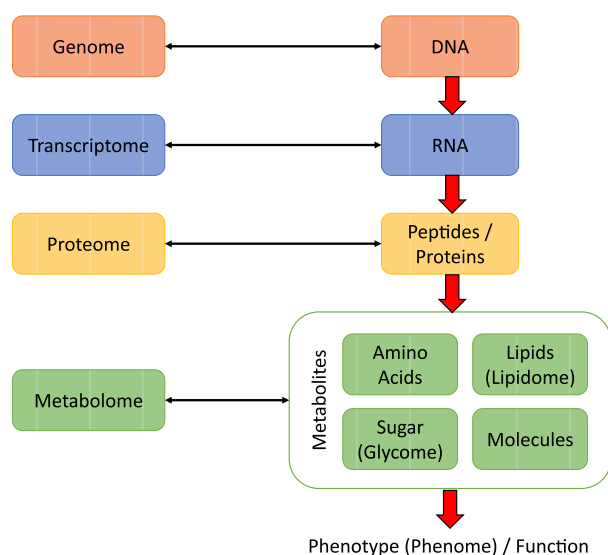


Fig. 1 Levels of omics and molecules between transcriptome and phenotype.

regulatory roles in many cellular processes. It is classified as protein-coding RNAs and nonprotein-coding RNAs (Eddy, 2001). Methods to comprehensively and systematically interrogate the expression of virtually all these RNA species have been developed over the past 20 years, from the initial expression sequencing tag (EST) based method (Boguski *et al.*, 1994), to serial analysis of gene expression, SAGE (Velculescu *et al.*, 1995) and cap analysis gene expression, CAGE (Shiraki *et al.*, 2003), to the hybridization based gene microarray (Lashkari *et al.*, 1997; Schena *et al.*, 1995), and now the RNA-seq (Wang *et al.*, 2009). The use of next generation sequencing (NGS) technology allows for a higher throughput and resolution level of transcriptome studies (Mardis, 2008).

EST Based Method. An expression sequencing tag (EST) is a short sub-sequence of a cDNA sequence. ESTs can be rapidly generated from either the 5' or 3' end of a cDNA clone in a high-throughput manner. In 1992, scientists at NCBI developed a new database, called dbEST, designed to serve as a collection point for ESTs. Data are submitted directly by laboratories and are not curated. Private EST databases such as SANBI (South Africa), MIPS (Munich), TIGEM (Italy) and DOTS (Pennsylvania) exist. Since ESTs represent portions of expressed genes, Sanger sequencing of EST is used in the early days of genome research (Boguski *et al.*, 1994). ESTs can be used to design probes for DNA microarrays as well. However, it is impossible to achieve transcriptomic quantitative analysis using this method because of the limitation on throughput and high cost.

SAGE and CAGE. SAGE was invented by Dr. Victor Velculescu in 1995 to allow quantitative and simultaneous analysis of a large number of transcripts (Velculescu *et al.*, 1995). The mRNA of an input sample is isolated, and cDNA is synthesized by means of biotinylated oligonucleotide primer. The cDNA is immobilized to streptavidin beads and type II restriction enzyme such as *NlaIII* is used to cleave cDNA to an average length of 256 bp with sticky ends. Captured cDNAs are ligated to linkers and then cleaved using tagging enzyme such as *BsmF1*. The ends were repaired using DNA polymerase to produce blunt end cDNA fragments. T4 DNA ligase is used to link two tags to form a 'ditag' with linker on either end. These linker-ditag-linker constructs are amplified by PCR, digested by the original anchoring enzyme, and then link together with other ditags to form cDNA concatemers. These concatemers are inserted into bacteria to create more copies, sequenced using high-throughput DNA sequencers and analyzed with computer programs. Over the years, variants of SAGE; such as, LongSAGE (Saha *et al.*, 2002), RL-SAGE (Gowda *et al.*, 2004) and SuperSAGE (Matsumura *et al.*, 2005) have been developed.

While in SAGE, multiple 3' cDNA ends were concatenated to be one close, CAGE concatenated the 5' cDNA ends. The original CAGE method (Shiraki *et al.*, 2003) was using biotinylated cap-trapper for capturing the 5' ends, oligo-dT₁₂₋₁₈ primers with superscript II RT in the presence of trehalose and sorbitol for synthesizing the cDNAs, the class II restriction enzyme Mmel for cleaving the tags, and the Sanger method (Sanger and Coulson, 1975; Sanger *et al.*, 1977) for sequencing them. To better detect the non-polyadenylated RNAs, random reverse-transcription primers were introduced (Kodzius *et al.*, 2006).

The output of SAGE and CAGE are a set of short nucleotide sequences known as tags with their observed counts. Using a reference genome, the mRNA from which the tag was extracted and its abundance could be obtained. Although protocols like *nAnTi-CAGE* (Murata *et al.*, 2014) and updated *nanoCAGE* (Poulain *et al.*, 2017) are published recently, due to the high cost of Sanger method for sequencing and the difficulty to map tags to the genome, CAGE and SAGE were most replaced by DNA microarray.

Microarray. The Microarray is based on nucleic acid hybridization. An array is constructed by immobilizing oligonucleotide probes on a solid support such as a silicon chip in an orderly manner. In the late 90's, DNA array technology progressed rapidly and three basic types of arrays production methods emerged during this frame: spotted arrays on glass (DeRisi *et al.*, 1996), *in-situ* synthesized arrays (Fodor *et al.*, 1991) and self-assembled arrays (Ferguson *et al.*, 2000; Michael *et al.*, 1998; Steemers *et al.*, 2000; Walt, 2000). Fluorescence-labeled cDNA is incubated with the chip. Due to complementary binding of the cDNA to the probe, the abundance of cDNA, hence the RNA can be determined by measuring the fluorescence density. Several different types of microarrays; such as, splicing-specific microarray (Clark *et al.*, 2002) and genome tiling array (Bertone *et al.*, 2004); had been developed to cater for different needs.

The microarray is based on hybridization and can only detect sequences that the array was designed to detect. Genes that have not yet been annotated in a genome or non-coding RNAs that are not yet recognized as expressed cannot be detected and RNA molecules sharing high sequence similarity cannot be distinguished from one another using this method. The advent of next generation sequencing technologies combined with the rapid decrease in the cost of sequencing makes RNA-seq (Wang *et al.*, 2009) the method of choice for measuring transcriptomes (Ledford, 2008; Su *et al.*, 2014; Zhao *et al.*, 2014a).

RNA-seq. RNA-seq refers to the combination of high-throughput next generation sequencing (NGS) with computational method to capture and quantify transcripts present in an RNA extract (Ozsolak and Milos, 2011). The RNA transcripts are isolated and reversely transcribed into cDNA. Massive parallel signature sequencing (Brenner *et al.*, 2000), which consists of four rounds of restriction enzyme digestion and ligation reaction, is used to generate sequence tags of 25–500 bp. By analyzing these tags, the transcriptome can be studied qualitatively and quantitatively (Nagalakshmi *et al.*, 2008).

The NGS technique generates a large volume of raw sequence data, which need to be processed to give meaning information. Numerous bioinformatics tools have developed to support the different steps of the process, and they can be classified under the following categories: quality control, alignment, quantification, and differential expression (Wang *et al.*, 2009; Zhao *et al.*, 2014a). Although it is the dominant transcriptomic technology, RNA-seq has its own limitations such as introducing bias during the cDNA library construction and RNA amplification stage, and the high cost for low input RNA-seq.

Making Sense of Transcriptome

After obtaining the transcriptome, we need to go from transcriptome to phenotype. This will be equivalent to making sense of the transcriptome data. For example, how changes in gene expressions lead to metabolic phenotypes? There are 2 main ways of

looking at it: to map the data onto an existing database for visualization and functional analysis, or to compare the data with a baseline transcriptome data obtained earlier, such as comparing the genome-wide transcriptome profiling of cancerous tissue with the normal tissue to find out which subset of the transcriptome is up or down regulated. This can then be mapped onto metabolic pathways as differential gene expression is likely to result in differences in enzyme concentrations or to be analyzed by Gene Ontology, to gain insights into the molecular mechanism of cancer initiation and progression.

Transcriptome to Proteomics and Metabolomics. To integrate the transcriptomic data with proteomics and metabolomics is controversial due to the low squared Pearson's correlation coefficient (R^2) between mRNA and protein concentrations (de Sousa Abreu *et al.*, 2009). In bacteria, the correlation coefficient ranges from 0.20 to 0.47, in yeasts from 0.34 to 0.87, and in multi-cellular organisms from 0.09 to 0.46. The poor correlation suggests a significant role for post-transcriptional, translational and degradation regulation in the determination of protein concentrations (Vogel and Marcotte, 2012). Nevertheless, one may perform integrative analysis to understand genetic control of molecular dynamics and phenotypic diversity in a system-wide manner, such as or to probe protein splice variants in different tissues (Ning and Nesvizhskii, 2010), organisms (Bai *et al.*, 2014) and cells (Sheynkman *et al.*, 2013). A study (Low *et al.*, 2013) examining the liver transcriptome (using RNA sequencing) and proteome (using protein sequencing) from two rat strains (SHR and BN-Lx) found large number of RNA editing and splice variant, which demonstrates the power of integrating and interrogating various sources of omics data. Post-translational regulation can also be identified if certain peptides are present in the mass spectrometry analysis but are absent from the expressed genes of the RNA-seq dataset (Conesa *et al.*, 2016).

Integration of transcriptomics with metabolomics data has been used to identify pathways that are regulated at both the gene expression and the metabolite level, and many tools are capable of visualizing time-course data to see which pathways are active. This can be used to infer the changes of metabolic reaction rates (fluxes) across time or between different samples. This is commonly known as fluxomics (Niedenführ *et al.*, 2015; Winter and Krömer, 2013), which aims to study the differential movement of metabolites across metabolic network, leading to phenotypic observations. Large number of metabolites from prokaryotic and eukaryotic organism can be easily and precisely detected using mass spectrometry. MassTRIX uses a hypothesis-driven approach to annotate MS data within the genomic context of the organism under study, allowing user to interpret the metabolic state of the organism in the context of its potential and real enzymatic capacities (Suhre and Schmitt-Kopplin, 2008). Paintomics, on the other hand, takes complete transcriptomics and metabolomics datasets that are measured on different conditions for the same samples, together with lists of significant gene or metabolite changes, and paints this information on KEGG pathway maps (García-Alcalde *et al.*, 2011). To supports scientists during data analysis and interpretation phase, VANTED integrate experimental data into biological networks and providing a rich variety of simulation, analysis and visualization functionalities. It aims to achieve seven main tasks: network reconstruction, data visualization, integration of various data types, network simulation, data exploration, a manifold support of systems biology standards for visualization and data exchange (Rohn *et al.*, 2012). Sometimes more than one type of data is collected during the experiment and the Omics Integrator, which is improved from the initial SteinerNet, perform advanced integration of multiple types of high-throughput data as well as create condition-specific subnetworks of protein interactions that best connect the observed changes in various datasets. Unannotated molecular pathways might be detected as well (Tuncbag *et al.*, 2016).

Phenotypic Inference using GSEA. Gene Ontology (Ashburner *et al.*, 2000; Gene Ontology Consortium, 2001) is a set of defined hierarchical vocabulary aiming to describe genes and proteins, based on their molecular functions, biological processes, and cellular components. Molecular function is the biochemical activity (including specific binding to ligands or structures) of a gene product. Biological process refers to a biological objective to which the gene or gene product contributes. Cellular component refers to the cellular location where a gene product is active. Several genomes; such as, *Escherichia coli*, *Saccharomyces cerevisiae*, *Drosophila melanogaster*, *Mus musculus*, and *Homo sapiens*; had been annotated with Gene Ontology (GO). This allows us to use GO as a tool to analyse differential transcriptomic data (Fruzangohar *et al.*, 2017; Ho and Fraser, 2016; Kim *et al.*, 2016). Differential transcriptomes are result of statistical comparison between two transcriptomes (Trapnell *et al.*, 2013); for example, treated versus control, treatment A versus treatment B, and time point A versus time point B; and usually results in a list of differentially up-regulated genes and down-regulated genes. Recent examples of such studies are the study of gene expressions of seabuckthorn carpenter moth between room temperature and cold (Cui *et al.*, 2017), differential gene expression between the shoots and rhizomes of wild rice (*O. longistaminata*) subjected to cold stress (Zhang *et al.*, 2017), and differential gene expression between *Rhodnius montenegrensis* and *Rhodnius robustus* (de Carvalho *et al.*, 2017), potential vectors of Chagas disease. In all three studies, GO was used.

It is common to have lists of differentially expressed genes (both up-regulated and down-regulated) running into hundreds of genes. For example, a study of transcriptomic differences between thin and normal human endometrial tissue during the mid-luteal phase of the menstrual cycle (Maekawa *et al.*, 2017) showed 318 up-regulated genes and 322 down-regulated genes in the thin endometrium, compared to the control endometrium, as thin endometrium results in lower pregnancy rates due to implantation failures. The next issue facing the biologist is to make sense of these genes – which biological processes are up-regulated or down-regulated in thin endometrium compared to control?

This can be answered by Gene Set Enrichment Analysis (Subramanian *et al.*, 2005), or GSEA. The premise of GSEA is that; given that each gene within the genome is tagged with one or more GO terms; for each GO term, the number of genes in the up-regulated gene list can be analyzed to see whether the GO term occurs statistically more than random. The null hypothesis (random occurrence) is based on the proportion of genes in the genome or in a platform (such as, a microarray platform) that is tagged with a specific GO term. For example, if 10% of all human genes are tagged with “mitotic cell cycle” (GO:0000278), we will expect that 10% of the 318 up-regulated genes (or 32 genes) will tagged with “mitotic cell cycle” if null hypothesis is true. This implies that we expect 90% of the 318 up-regulated genes (or 286 genes) that are not tagged with “mitotic cell cycle”. However, if there are 45 genes out of 318 up-regulated genes are tagged with “mitotic cell cycle” (278 genes not tagged with “mitotic cell cycle”), then is the function of mitosis occurring higher than random?

Using χ^2 test, we get a p-value of 0.02, suggesting that mitotic function occurs higher than random and implying that mitotic function is up-regulated in thin endometrium compared to normal endometrium. This can be done repeatedly with each of the more than 40 thousand GO terms and each GO term that occurs more than random is known as enriched term. The result of this series of analyses is to obtain a set of enriched GO terms, which represents the phenotypic differences between the 2 samples – in this case, phenotypes that are more prevalent in thin endometrium compared to normal endometrium. GSEA has been used in many studies to deduce the phenotype changes from one sample to the next (Elgendy *et al.*, 2016; He *et al.*, 2013; Heng *et al.*, 2011; Kong *et al.*, 2014; Li *et al.*, 2017; Ling, 2011; Ling and Poh, 2014; Mühleisen *et al.*, 2017; Too and Ling, 2012; Weidner *et al.*, 2017).

Gene Regulatory Network Inference. Scientist may want to elucidate gene regulatory network (GRN) from large scale experimental data. To select the network model and fit the available data into the network's structure parameter is a crucial step for constructing the network from expression data. These network models can be classified into two major categories: those use continuous variables and those use discrete variables in the modelling process.

For models with discrete variables, GenYsis (Garg *et al.*, 2008) utilises Boolean network model which assumes the gene is in either on (active or expressed) or off (inactive or unexpressed). Although it is easy to simulate, Boolean network model is not able to capture certain system behaviours that can be captured by continuous models (de Jong, 2002). To overcome this problem, the probabilistic Boolean network (Shmulevich *et al.*, 2002) was proposed, which introduced uncertainty at each network node in the form of regulatory functions with a predefined probability. Another popular model for discrete variables is the Bayesian network model, which specifies a set of conditional independence assumptions in addition to a set of conditional probabilities, to obtain a directed acyclic graph that explicitly establishes probabilistic relationships between network nodes. Although it is classified as a model for discrete variable, Bayesian network can also be used for continuous variables (Lee and Tzou, 2009). For continuous variable, one may also develop a model based on differential equation, such as the one uses stochastic modelling for parameter identification of genetic regulatory networks in prokaryotes (Cinquemani *et al.*, 2008).

The primary concern of biologists is how to translate the inferred network into hypotheses that can be tested with real-life experiments. It is important to utilize information from multiple sources to narrow down the search space after a GRN is constructed (Mobini *et al.*, 2009). Pathway databases; such as, KEGG (Okuda *et al.*, 2008), and DAVID (Dennis *et al.*, 2003); are used to match the inferred GRN to a module or subnetwork. The module or subnetwork is then used to search for genes with similar gene expression profiles.

The advances of the high-throughput next generation sequencing technology (NGS) and the development of various bioinformatics tools allowed researchers to fit theoretical models to experimental data on gene expression profile. A study (Gluck *et al.*, 2016) mapping mouse salivary gland transcriptome data onto gene regulatory networks not only confirms known modulators involved in salivary gland morphogenesis but also reveal novel transcription factors and signalling pathways unique to mouse salivary glands.

Concluding Remarks

One of the major reasons for using transcriptomic data as the source data to elucidate phenotype is the relative advancement in experimental techniques in the area of transcriptomics compared to proteomics and metabolomics. Hence, transcriptomic data acts as a convenient proxy to proteomic and metabolomics data. Earlier studies using only transcriptomic data had been promising (Curtis *et al.*, 2012). For example, a study (Urzúa *et al.*, 2010) found 60% of the biological variability observed in spontaneous ovarian tumor rates and reproductive parameters across four mouse strains (BalbC, C57BL6, FVB and SWR) can be attributed to transcriptomic differences measured using microarrays.

However, in recent years, there is an increase in the number of studies attempting to integrate various sources of omics data, including transcriptomics, to elucidate the molecular basis of phenotypes. For example, epigenomic, transcriptomic, and proteomic data had been integrated to explain dysfunction in human hepatocyte caused by valproic acid (Wolters *et al.*, 2018). Multi-omics approach had also been shown to improve survival prediction in patients with neuroblastoma (Francescato *et al.*, 2018) and liver cancer (Chaudhary *et al.*, 2018). This has in turn triggered the development of algorithms to use multi-omics data (Mandal and Maji, 2018). Hence, it is foreseeable that with advances in high-throughput experimental techniques and improved computational algorithms, transcriptomics will be one of the data sources in a multi-omics pool to understand the why and how of phenotypes.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Characterizing and Functional Assignment of Noncoding RNAs. Codon Usage. Comparative Transcriptomics Analysis. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Regulation of Gene Expression. Transcriptome Analysis. Whole Genome Sequencing Analysis

References

- Ashburner, M., Ball, C.A., Blake, J.A., *et al.*, 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25, 25–29.
- Bai, Y., Hassler, J., Ziyar, A., *et al.*, 2014. Novel bioinformatics method for identification of genome-wide non-canonical spliced regions using RNA-Seq data. *PLOS ONE* 9, e100864.

- Belgrader, P., Cheng, J., Zhou, X., Stephenson, L.S., Maquat, L.E., 1994. Mammalian nonsense codons can be cis effectors of nuclear mRNA half-life. *Mol. Cell. Biol.* 14, 8219–8228.
- Bertone, P., Stolic, V., Royce, T.E., *et al.*, 2004. Global identification of human transcribed sequences with genome tiling arrays. *Science* 306, 2242–2246.
- Boguski, M.S., Tolstoshev, C.M., Bassett, D.E., 1994. Gene discovery in dbEST. *Science* 265, 1993–1994.
- Brenner, S., Johnson, M., Bridgham, J., *et al.*, 2000. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nat. Biotechnol.* 18, 630–634.
- de Carvalho, D.B., Congrains, C., Chahad-Ehlers, S., *et al.*, 2017. Differential transcriptome analysis supports *Rhodnius montenegrensis* and *Rhodnius robustus* (Hemiptera, Reduviidae, Triatominae) as distinct species. *PLOS ONE* 12, e0174997.
- Casamassimi, A., Federico, A., Rienzo, M., Esposito, S., Ciccociola, A., 2017. Transcriptome profiling in human diseases: New advances and perspectives. *Int. J. Mol. Sci.* 18, 1652.
- Chaudhary, K., Poirion, O.B., Lu, L., Garmire, L.X., 2018. Deep learning-based multi-omics integration robustly predicts survival in liver cancer. *Clin. Cancer Res.* 24, 1248–1259.
- Cinquemani, E., Miliadis-Argeitis, A., Summers, S., Lygeros, J., 2008. Stochastic dynamics of genetic networks: Modelling and parameter identification. *Bioinformatics* 24, 2748–2754.
- Clark, T.A., Sugnet, C.W., Ares, M., 2002. Genomewide analysis of mRNA processing in yeast using splicing-specific microarrays. *Science* 296, 907–910.
- Conesa, A., Madrigal, P., Tarazona, S., *et al.*, 2016. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17, 13.
- Cui, M., Hu, P., Wang, T., Tao, J., Zong, S., 2017. Differential transcriptome analysis reveals genes related to cold tolerance in seabuckthorn carpenter moth, *Eogystia hippophaecolus*. *PLOS ONE* 12, e0187105.
- Curtis, R.E., Yin, J., Kinnaird, P., Xing, E.P., 2012. Finding genome-transcriptome-phenome association with structured association mapping and visualization in GenAMap. *Pac. Symp. Biocomput.* 327–338.
- Dennis Jr., G., Sherman, B.T., Hosack, D.A., *et al.*, 2003. DAVID: Database for annotation, visualization, and integrated discovery. *Genome Biol.* 4, P3.
- DeRisi, J., Penland, L., Brown, P.O., *et al.*, 1996. Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nat. Genet.* 14, 457–460.
- Eddy, S.R., 2001. Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.* 2, 919–929.
- Elgendy, R., Giantin, M., Castellani, F., *et al.*, 2016. Transcriptomic signature of high dietary organic selenium supplementation in sheep: A nutrigenomic insight using a custom microarray platform and gene set enrichment analysis. *J. Anim. Sci.* 94, 3169–3184.
- Evratov, S.A., Osterman, I.A., Komarova, E.S., *et al.*, 2017. Application of sorting and next generation sequencing to study 5'-UTR influence on translation efficiency in *Escherichia coli*. *Nucleic Acids Res.* 45, 3487–3502.
- Ferguson, J.A., Steemers, F.J., Walt, D.R., 2000. High-density fiber-optic DNA random microsphere array. *Anal. Chem.* 72, 5618–5624.
- Fishbain, S., Inobe, T., Israeli, E., *et al.*, 2015. Sequence composition of disordered regions fine-tunes protein half-life. *Nat. Struct. Mol. Biol.* 22, 214–221.
- Fodor, S.P., Read, J.L., Pirrung, M.C., *et al.*, 1991. Light-directed, spatially addressable parallel chemical synthesis. *Science* 251, 767–773.
- Francescato, M., Chierici, M., Rezvan Dezfouli, S., *et al.*, 2018. Multi-omics integration for neuroblastoma clinical endpoint prediction. *Biol. Direct* 13, 5.
- Fruzangohar, M., Ebrahimie, E., Adelson, D.L., 2017. A novel hypothesis-unbiased method for Gene Ontology enrichment based on transcriptome data. *PLOS ONE* 12, e0170486.
- Gamble, C.E., Brule, C.E., Dean, K.M., Fields, S., Grayhack, E.J., 2016. Adjacent codons act in concert to modulate translation efficiency in yeast. *Cell* 166, 679–690.
- García-Alcalde, F., García-López, F., Dopazo, J., Conesa, A., 2011. Paintomics: A web based tool for the joint visualization of transcriptomics and metabolomics data. *Bioinformatics* 27, 137–139.
- Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., De Micheli, G., 2008. Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 24, 1917–1925.
- Gene Ontology Consortium, 2001. Creating the gene ontology resource: Design and implementation. *Genome Res.* 11, 1425–1433.
- Gluck, C., Min, S., Oyelakin, A., *et al.*, 2016. RNA-seq based transcriptomic map reveals new insights into mouse salivary gland development and maturation. *BMC Genom.* 17, 923.
- Gowda, M., Jantasuriyarat, C., Dean, R.A., Wang, G.-L., 2004. Robust-LongSAGE (RL-SAGE): A substantially improved LongSAGE method for gene discovery and transcriptome analysis. *Plant Physiol.* 134, 890–897.
- He, W., Qi, B., Zhou, Q., *et al.*, 2013. Key genes and pathways in thyroid cancer based on gene set enrichment analysis. *Oncol. Rep.* 30, 1391–1397.
- Heng, S.S.J., Chan, O.Y.W., Keng, B.M.H., Ling, M.H.T., 2011. Glucan Biosynthesis Protein G (mdoG) is a suitable reference gene in *Escherichia coli* K-12. *ISRN Microbiol.* 2011, Article ID 469053.
- Ho, M.-M., Fraser, D.A., 2016. Transcriptome data and gene ontology analysis in human macrophages ingesting modified lipoproteins in the presence or absence of complement protein C1q. *Data Brief* 9, 362–367.
- Hockenberry, A.J., Pah, A.R., Jewett, M.C., Amaral, L.A.N., 2017. Leveraging genome-wide datasets to quantify the functional role of the anti-Shine-Dalgarno sequence in regulating translation efficiency. *Open Biol.* 7.
- de Jong, H., 2002. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.* 9, 67–103.
- Kang, C.-J., Chen, Y.-J., Liao, C.-T., *et al.*, 2009. Transcriptome profiling and network pathway analysis of genes associated with invasive phenotype in oral cancer. *Cancer Lett.* 284, 131–140.
- Kim, H.-I., Kim, J.-H., Park, Y.-J., 2016. Transcriptome and Gene Ontology (GO) enrichment analysis reveals genes involved in biotin metabolism that affect L-lysine production in *Corynebacterium glutamicum*. *Int. J. Mol. Sci.* 17, 353.
- Kodzius, R., Kojima, M., Nishiyori, H., *et al.*, 2006. CAGE: Cap analysis of gene expression. *Nat. Methods* 3, 211.
- Kong, B., Yang, T., Chen, L., *et al.*, 2014. Protein-protein interaction network analysis and gene set enrichment analysis in epilepsy patients with brain cancer. *J. Clin. Neurosci.* 21, 316–319.
- Kumar, S., van Raam, B.J., Salvesen, G.S., Cieplak, P., 2014. Caspase cleavage sites in the human proteome: CaspDB, a database of predicted substrates. *PLOS ONE* 9, e110539.
- Kyung, D.S., Sung, H.R., Kim, Y.J., *et al.*, 2017. Global transcriptome analysis identifies weight regain-induced activation of adaptive immune responses in white adipose tissue of mice. *Int. J. Obes.*
- Lahtvee, P.-J., Sánchez, B.J., Smialowska, A., *et al.*, 2017. Absolute quantification of protein and mRNA abundances demonstrate variability in gene-specific translation efficiency in yeast. *Cell Syst.* 4, 495–504. e5.
- Lashkari, D.A., DeRisi, J.L., McCusker, J.H., *et al.*, 1997. Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proc. Natl. Acad. Sci. USA* 94, 13057–13062.
- Ledford, H., 2008. The death of microarrays? *Nature* 455, 847.
- Lee, W.-P., Tzou, W.-S., 2009. Computational methods for discovering gene networks from expression data. *Brief. Bioinform.* 10, 408–423.
- Li, W.-X., He, K., Tang, L., *et al.*, 2017. Comprehensive tissue-specific gene set enrichment analysis and transcription factor analysis of breast cancer by integrating 14 gene expression datasets. *Oncotarget* 8, 6775–6786.
- Ling, M.H., 2011. Bactome II: Analyzing gene list for gene ontology over-representation. *The Python Papers Source Codes* 3, 3.
- Ling, M.H., Poh, C.L., 2014. A predictor for predicting *Escherichia coli* transcriptome and the effects of gene perturbations. *BMC Bioinform.* 15, 140.
- Low, T.Y., van Heesch, S., van den Toorn, H., *et al.*, 2013. Quantitative and qualitative proteome characteristics extracted from in-depth integrated genomics and proteomics analysis. *Cell Rep.* 5, 1469–1478.

- Maekawa, R., Taketani, T., Mihara, Y., *et al.*, 2017. Thin endometrium transcriptome analysis reveals a potential mechanism of implantation failure. *Reprod. Med. Biol.* 16, 206–227.
- Mandal, A., Maji, P., 2018. FaRoC: Fast and robust supervised canonical correlation analysis for multimodal omics data. *IEEE Trans. Cybern.* 48, 1229–1241.
- Mardis, E.R., 2008. Next-generation DNA sequencing methods. *Annu. Rev. Genom. Hum. Genet.* 9, 387–402.
- Matsumura, H., Ito, A., Saitoh, H., *et al.*, 2005. SuperSAGE. *Cell. Microbiol.* 7, 11–18.
- Michael, K.L., Taylor, L.C., Schultz, S.L., Walt, D.R., 1998. Randomly ordered addressable high-density optical sensor arrays. *Anal. Chem.* 70, 1242–1248.
- Mobini, R., Andersson, B.A., Erjefält, J., *et al.*, 2009. A module-based analytical strategy to identify novel disease-associated genes shows an inhibitory role for interleukin 7 Receptor in allergic inflammation. *BMC Syst. Biol.* 3, 19.
- Mühleisen, T.W., Reinbold, C.S., Forstner, A.J., *et al.*, 2017. Gene set enrichment analysis and expression pattern exploration implicate an involvement of neurodevelopmental processes in bipolar disorder. *J. Affect. Disord.* 228, 20–25.
- Murata, M., Nishiyori-Sueki, H., Kojima-Ishiyama, M., *et al.*, 2014. Detecting expressed genes using CAGE. *Methods Mol. Biol.* 1164, 67–85.
- Nagalakshmi, U., Wang, Z., Waern, K., *et al.*, 2008. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320, 1344–1349.
- Niedenführ, S., Wiechert, W., Nöh, K., 2015. How to measure metabolic fluxes: A taxonomic guide for (13)C fluxomics. *Curr. Opin. Biotechnol.* 34, 82–90.
- Ning, K., Nesvizhskii, A.I., 2010. The utility of mass spectrometry-based proteomic data for validation of novel alternative splice forms reconstructed from RNA-Seq data: A preliminary assessment. *BMC Bioinform.* 11 (Suppl. 1), S14.
- Okuda, S., Yamada, T., Hamajima, M., *et al.*, 2008. KEGG Atlas mapping for global analysis of metabolic pathways. *Nucleic Acids Res.* 36, W423–W426.
- Ozsolak, F., Milos, P.M., 2011. RNA sequencing: Advances, challenges and opportunities. *Nat. Rev. Genet.* 12, 87–98.
- Poulain, S., Kato, S., Arnaud, O., *et al.*, 2017. NanoCAGE: A method for the analysis of coding and noncoding 5'-capped transcriptomes. *Methods Mol. Biol.* 1543, 57–109.
- Rohn, H., Junker, A., Hartmann, A., *et al.*, 2012. VANTED v2: A framework for systems biology applications. *BMC Syst. Biol.* 6, 139.
- Saha, S., Sparks, A.B., Rago, C., *et al.*, 2002. Using the transcriptome to annotate the genome. *Nat. Biotechnol.* 20, 508–512.
- Sanger, F., Coulson, A.R., 1975. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J. Mol. Biol.* 94, 441–448.
- Sanger, F., Nicklen, S., Coulson, A.R., 1977. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA* 74, 5463–5467.
- Schena, M., Shalon, D., Davis, R.W., Brown, P.O., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467–470.
- Sheynkman, G.M., Shortreed, M.R., Frey, B.L., Smith, L.M., 2013. Discovery and mass spectrometric analysis of novel splice-junction peptides using RNA-Seq. *Mol. Cell Proteom.* 12, 2341–2353.
- Shiraki, T., Kondo, S., Katayama, S., *et al.*, 2003. Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proc. Natl. Acad. Sci. USA* 100, 15776–15781.
- Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W., 2002. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18, 261–274.
- de Sousa Abreu, R., Penalva, L.O., Marcotte, E.M., Vogel, C., 2009. Global signatures of protein and mRNA expression levels. *Mol. Biosyst.* 5, 1512–1526.
- Steemers, F.J., Ferguson, J.A., Walt, D.R., 2000. Screening unlabeled DNA targets with randomly ordered fiber-optic gene arrays. *Nat. Biotechnol.* 18, 91.
- Su, Z., Fang, H., Hong, H., *et al.*, 2014. An investigation of biomarkers derived from legacy microarray data for their utility in the RNA-seq era. *Genome Biol.* 15, 523.
- Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* 102, 15545–15550.
- Suhre, K., Schmitt-Kopplin, P., 2008. MassTRIX: Mass translator into pathways. *Nucleic Acids Res.* 36, W481–W484.
- Sul, J.-Y., Wu, C.K., Zeng, F., *et al.*, 2009. Transcriptome transfer produces a predictable cellular phenotype. *Proc. Natl. Acad. Sci. USA* 106, 7624–7629.
- Too, I.H.K., Ling, M.H.T., 2012. Signal peptidase complex subunit 1 and Hydroxyacyl-CoA Dehydrogenase Beta Subunit are suitable reference genes in human lungs. *ISRN Bioinform.* 2012, Article ID 790452.
- Trapnell, C., Hendrickson, D.G., Sauvageau, M., *et al.*, 2013. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nat. Biotechnol.* 31.10.1038/nbt.2450.
- Tuncbag, N., Gosline, S.J.C., Kedaigle, A., *et al.*, 2016. Network-based interpretation of diverse high-throughput datasets through the omics integrator software package. *PLOS Comput. Biol.* 12, e1004879.
- Urzúa, U., Owens, G.A., Zhang, G.-M., *et al.*, 2010. Tumor and reproductive traits are linked by RNA metabolism genes in the mouse ovary: A transcriptome-phenotype association analysis. *BMC Genom.* 11 (Suppl. 5), S1.
- Velculescu, V.E., Zhang, L., Vogelstein, B., Kinzler, K.W., 1995. Serial analysis of gene expression. *Science* 270, 484–487.
- Vogel, C., Marcotte, E.M., 2012. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nat. Rev. Genet.* 13, 227–232.
- Walt, D.R., 2000. Techview: Molecular biology. Bead-based fiber-optic arrays. *Science* 287, 451–452.
- Wang, Z., Gerstein, M., Snyder, M., 2009. RNA-Seq: A revolutionary tool for transcriptomics. *Nat. Rev. Genet.* 10, 57–63.
- Weidner, C., Steinfath, M., Wistorf, E., *et al.*, 2017. A protocol for using gene set enrichment analysis to identify the appropriate animal model for translational research. *J. Vis. Exp.*
- Winter, G., Krömer, J.O., 2013. Fluxomics – Connecting 'omics analysis and phenotypes. *Environ. Microbiol.* 15, 1901–1916.
- Wolters, J.E.J., van Breda, S.G.J., Grossmann, J., *et al.*, 2018. Integrated 'omics analysis reveals new drug-induced mitochondrial perturbations in human hepatocytes. *Toxicol. Lett.* 289, 1–13.
- Zhang, T., Huang, L., Wang, Y., *et al.*, 2017. Differential transcriptome profiling of chilling stress response between shoots and rhizomes of *Oryza longistaminata* using RNA sequencing. *PLOS ONE* 12, e0188625.
- Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., Liu, X., 2014a. Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLOS ONE* 9, e78644.
- Zhao, T., Wu, Z., Wang, S., Chen, L., 2014b. Expression and function of natural antisense transcripts in mouse embryonic stem cells. *Sci. China Life Sci.* 57, 1183–1190.

Transcriptome During Cell Differentiation

Dwi A Pujianto, Universitas Indonesia, Jakarta, Indonesia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The term development in animal is a very complex process in which a fertilized egg, the zygote, undergoes a series of gene-regulated changes to form a complete and adult organism. These changes involve not only cell proliferation but also specialization of the cells to become a certain cell type with a specific function. As the process begins, the single cell-zygote undergoes a series of cleavage to produce a morula (16–32 cells), blastula and gastrula in which three germ layers are formed. The three germ layers will differentiate into all organ systems in the mature organism. Development of an individual combines several different molecular mechanisms that can affect cell's number, morphology, behavior and function (Twyman, 2001). During development process there are changes as follow:

1. Cell proliferation: The number of cell increases as result of repeated cell division.
2. Cell growth: Increase in the production of molecules such as nucleotides, proteins, carbohydrates and lipids that cause an increase in cell's total volume.
3. Cell differentiation: Alteration of the cells from the stage of multipotent that can transform to any kind of cell to become more specific type with a specialized function.

All of these mechanisms are regulated by many genes and involves various gene regulatory proteins which determine time and location of the expression producing a specific protein in the embryo that directs cell behavior (Zasso *et al.*, 2018). Since multicellular organisms develop from a single cell (zigot), thus all cells contain the same genes. The question arises on how cells that contain the same genes behave differently? Therefore, for cell to behave differently during embryonic development, different set of genes are expressed in a certain type of cell. Moreover, especially in the eukaryotic cells, gene expression is much regulated by many transcription factors (Alberts *et al.*, 2015). Hence, development of an individual from zygote to adult is also governed by specific transcription factors (Kuo *et al.*, 1992). Simultaneous gene transcription analyses during development will reveal the key players in maintaining the multipotency and also key players responsible for cell differentiation to become specialized cell type with a specific function. In this article we will focus on gene expression profile during cell differentiation.

Cell Specialization During Development

After fertilization the zygote undergoes early cleavage producing blastomeres which are unspecialized. These blastomeres have characteristic to be totipotent, because they have capability to transform into all kinds of cells in the body, this includes amnion and chorion which are considered to be extra-embryonic membranes (Fig. 1). When the development occurs, cells undergo not only multiplication but also differentiation producing cells with a specialized function. After early cleavage of the zygote producing 2–4 blastomeres, the number of cell increases to form morula-like structure called morula containing 16–32 blastomeres (Moore and Persaud, 2008). The blastomeres then differentiate into blastocyst with two different groups of cells. The first is the inner cell mass which will develop into embryo proper and amnion and the second population is the trophoblast which will develop into the chorion and part of the embryo called the placenta. The inner cell mass cells have characteristic of being pluripotent which mean these cells have potency to differentiate into any cell type in the developing embryo but they no longer have capacity to develop into extra-embryonic structure derived from the trophoblast.

Cells that have differentiated into inner cell mass subsequently develop into one of the three germ layers which is the characteristic of the gastrula. The three germ layers will differentiate into specific cell type that forms organ systems (Fig. 1). The internal layer endoderm is the future of internal organs such as respiratory and digestive organs. The middle layer mesoderm will differentiate into muscles, urogenital organs and connective tissues including blood cells and vessel. The external layer ectoderm gives rise to organs in the neural system, sense organs and outer epithelium of the body (Moore and Persaud, 2008). Cells in the three germ layers have characteristic of being multipotent. These cells have capacity to differentiate into many cell types but not all cell types in the embryo. Cells in the immune system and blood for instance, can differentiate into a progenitor cell with capacity to give rise only to few cells in the lineage such as lymphocytes which subsequently differentiate into lymphocyte B and T. This kind of cell is characterized to be oligopotent. Finally, some cells differentiate into precursor cells with capacity to transform only to one cell type. These cells are categorized as unipotent. The precursor cells such as hepatocytes, osteoblast and chondroblast usually have capacity to self-renewal or act as stem cell for terminally differentiated cells (Strachan and Read, 2004).

Gene Transcription During Cell Differentiation

Back to the previous question, how cells with the same genome behave differently and give rise to various cells with specific function? Obviously there are mechanisms to regulate gene expression during changes from cells with totipotent to multipotent

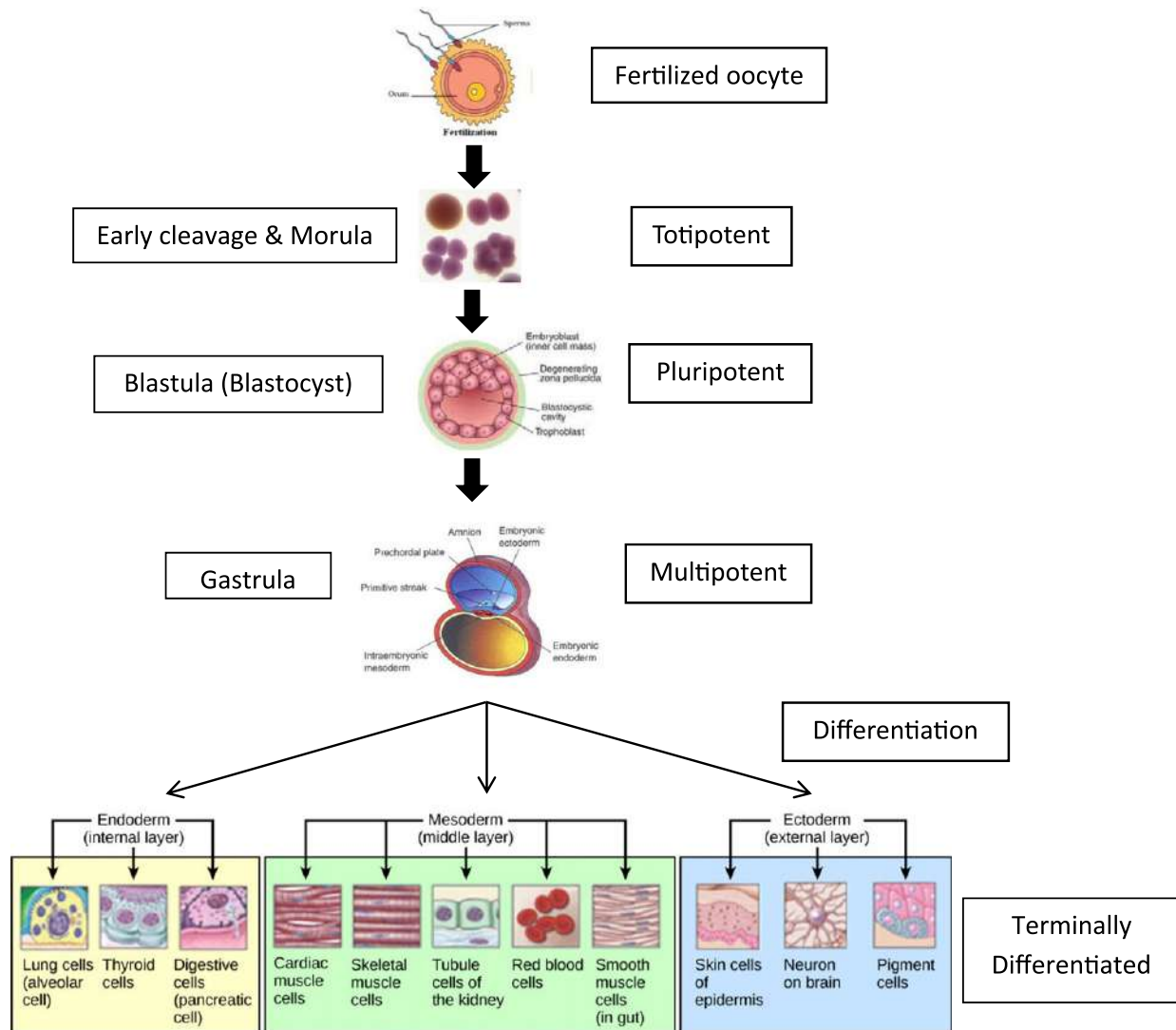


Fig. 1 Development of zygote (fertilized oocyte) to become an individual start from early cleavage, morula, blastula, and gastrula in which three germ layers are formed. The three germ layers undergo differentiation into various tissues and organs. Each stage of the development has its own potency whether totipotent, pluripotent or multipotent.

capacities to become specialized cell. Since development and differentiation involve cell proliferation, movement, interaction with other cells and specialization, the first important genes are the gene that encode for transmembrane molecules which is important for cell-cell adhesion and cell signaling (Alberts *et al.*, 2015; Sanchez-Arrones *et al.*, 2012). The second gene is the genes that encode for transcription factors which is important for regulating gene expression (Alberts *et al.*, 2015; Cave and Sockanathan, 2018). Regulation of gene expression during cell differentiation is also performed by epigenetic mechanisms involving chromatin modification and microRNA (Alvarez-Errico *et al.*, 2015; Huang *et al.*, 2014; Youngblood *et al.*, 2013). Differential gene expression among the cells undergoing differentiation is also dictated by signal molecules present in the environment. Signal molecules affect gene regulatory protein (transcription factor) binding to the gene promoter, thus some transcriptions are "ON" some others are "OFF" during differentiation. The presence of signal molecules also determine whether cell commit asymmetric or symmetric division to produce specialized cell (Alberts *et al.*, 2015). Some well characterized signal proteins among other things are TGF β superfamily, Wnt, Hedgehog and Notch. Gene expression profiling using cDNA microarray to analyze gene transcription (Transcriptome) during cell differentiation is a very challenging task to identify key players in the process.

Transcriptome of Cell Differentiation

The molecular mechanisms that can explain pluripotency and then differentiation from the inner cell mass of the blastocyst (embryonic stem cell, ESCs) are largely unknown. Embryonic stem cells are capable of self-renew because they have stem cell

specific factors (Hochedlinger *et al.*, 2005; Hough *et al.*, 2006) and to initiate differentiation into any cell type of the three germ layers by activation of specific sets of genes by transcription factors that are required for each specific lineage (Szutorisz and Dillon, 2005). In addition to activation by specific transcription factors cell pluripotency and differentiation is also regulated by epigenetic mechanisms (Azuara *et al.*, 2006). Chromatin structure in the ESCs is characterized by specific features which is different with that of differentiated cells (Niwa, 2007). Chromatin in the ESCs is morphologically distinct from that of differentiated cells in which in the ESCs chromatin is looser so that more gene expressed in the ESCs compare to the differentiated cells (Aoto *et al.*, 2006). In the undifferentiated cells like ESCs chromatin indicate transcriptionally active state and express large regions of the genome, probably not in specific manner and at low levels (Meshorer and Misteli, 2006). Undifferentiated ESCs also express repetitive sequences, mobile elements, as well as lineage and tissue specific genes at low levels.

According to Efroni *et al.*, the properties of ESC chromatin such as global decondensation, looser binding of chromatin proteins and enrichment of active histone modification indicate transcriptionally active chromatin and it is hypothesized that undifferentiated cells (ESCs) are globally transcriptionally more active than differentiated cells. Efroni test this hypothesis, global transcription was measured by (3H) uridine incorporation and compared between undifferentiated ESCs and 7 day neuronal progenitor cells (NPCs) derived from ESCs by invitro differentiation. The results showed that total RNA and mRNA levels were almost two-fold higher in the ESCs compared to the differentiated NPCs (Efroni *et al.*, 2008). The increase in the transcriptional activity in the in the ESCs is caused by activity of the specific set of genes that reflect global activation of the genome in the ESCs. The phenomenon that higher transcription is found in the ESCs compare to the differentiated cells is evidenced in the experiment performed by Efroni which compared transcription of 12 lineage specific genes (Fig. 2).

Identification of Genes Involved in Cell Differentiation

At the blastocyst stage, the inner cell mass (ICM) will develop into three germ layers namely the ectoderm, mesoderm and endoderm. The ICM constitutes the embryonic stem cells that are capable of differentiating into any type of cells. This formation three germ layers is the characteristic of gastrulation and marks the beginning of differentiation into organs, for instance nervous system is derived from the ectoderm, skeletal muscle from the mesoderm and digestive system from the endoderm (Moore and Persaud, 2008). One of the clear examples of cell differentiation is the development of brain from the ectoderm layer. Brain development is one of the most complicated processes during embryogenesis. In the neuroectoderm differentiation, expression of pluripotent genes such as POU5F1, SOX2, NANOG decreases and the expression of neuroectoderm genes such as POU3F1, ZNF521, and also neural epithelial markers such as PAX6 and SOX1 increase and reach the peak at day 12 post fertilization (Li *et al.*, 2017) (Fig. 3).

Analyses gene expression profiling on the potency of human embryonic stem cell (hESC) to develop into neural system based on developmental period suggesting that stage 3, which is at day 8–10 of the invitro differentiation, is the critical windows for the transition from pluripotency to the neural epithelium. Moreover, several transcription factors are identified to be involved in this stage such as PAX6, SIX3, HESX1 and ID3. Deletion of either SIX3 or HESX1 causes abnormality in the development of hESC to become neural cells (Li *et al.*, 2017). Correlation analysis of the downstream target of these transcription factor found that SIX3 negatively correlated with pluripotent genes such as NANOG. This suggests that SIX3 gene promotes neural differentiation by regulating its downstream transcription factors.

	Undifferentiated		Differentiated		
	ESC	NPC	PMN	MEF	C2C12
Gene name					
Actin, alpha 1 (Acta1)					
Receptor activator of NF kappa B ligand (RANKL)					
Prostate androgen induced 1 (PMEPA1)					
Small proline rich protein 2A (SPRR2A)					
Albumin					
CD3					
CD8					
Glial fibrillary acidic protein (GFAP)					
Surfactant protein B (SP-B)					
Uromodulin					
Synaptotagmin-11					
Myogenin					

Fig. 2 Reduced global transcription in differentiated cells compared to undifferentiated cells (ESC). The differentiated cells are represented by ESC-derived neuronal progenitor cells (NPC), ESC-derived postmitotic neurons (PMN), mouse embryonic fibroblast (MEF), and immortalized mouse myoblast cell line (C2C12). Black filled boxes represent expressed genes whereas white boxes represent undetected gene expression. Modified from Efroni, S., Duttagupta, R., Cheng, J., *et al.*, 2008. Global transcription in pluripotent embryonic stem cells. *Cell Stem Cell* 2, 437–447.

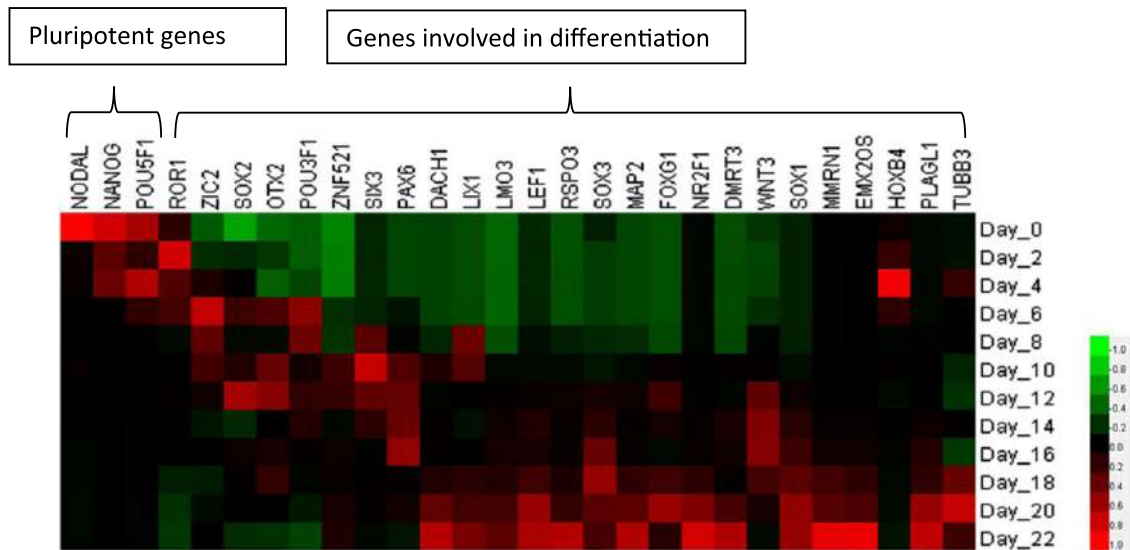


Fig. 3 Expression profile of pluripotent genes (NODAL, NANOG and POU5F1) and differentiation markers such as SOX2, OTX2, and POU3F1. At the beginning of embryogenesis at day 2–4 pluripotent genes are expressed (Indicates by red color), whereas during differentiation at day 6–22 expression of pluripotent genes decrease and differentiation genes are up-regulated. Red color indicates increase in the level of expression, green indicates expression decrease. Modified from Li, Y., Wang, R., Qiao, N., *et al.*, 2017. Transcriptome analysis reveals determinant stages controlling human embryonic stem cell commitment to neuronal cells. *J. Biol. Chem.* 292, 19590–19604.

Conclusion

The development of multicellular organism starts with cell proliferation, growth and differentiation that produces specialized cells. At the pluripotent stage (stem cells) the chromatin packaging is looser so that there are more genes expressed at this stage although at low levels. When cells undergo differentiation, expression of pluripotent genes such as POU5F1, SOX2, NANOG decreases, whereas genes involved in differentiation such as POU3F1 and ZNF521 increase. At the critical windows for the transition from pluripotency to differentiated cells, there is a negative correlation between transcription factors and pluripotent genes such as NODAL, NANOG and POU5F1, but positive correlation with genes involved in differentiation such as SOX2, OTX2, and POU3F1.

See also: Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Characterizing and Functional Assignment of Noncoding RNAs. Codon Usage. Comparative Transcriptomics Analysis. Coupling Cell Division to Metabolic Pathways Through Transcription. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Regulation of Gene Expression. Transcriptome Analysis. Whole Genome Sequencing Analysis

References

- Alberts, B., Johnson, A., Lewis, J., *et al.*, 2015. *Molecular Biology of the Cell*, sixth ed. New York: Garland Science.
- Alvarez-Errico, D., Vento-Tormo, R., Sieweke, M., Ballestar, E., 2015. Epigenetic control of myeloid cell differentiation, identity and function. *Nat. Rev. Immunol.* 15, 7–17.
- Aoto, T., Saitoh, N., Ichimura, T., Niwa, H., Nakao, M., 2006. Nuclear and chromatin reorganization in the MHC-Oct3/4 locus at developmental phases of embryonic stem cell differentiation. *Dev. Biol.* 298, 354–367.
- Azura, V., Perry, P., Sauer, S., *et al.*, 2006. Chromatin signatures of pluripotent cell lines. *Nat. Cell Biol.* 8, 532–538.
- Cave, C., Sockanathan, S., 2018. Transcription factor mechanisms guiding motor neuron differentiation and diversification. *Curr. Opin. Neurobiol.* 53, 1–7.
- Efroni, S., Duttagupta, R., Cheng, J., *et al.*, 2008. Global transcription in pluripotent embryonic stem cells. *Cell Stem Cell* 2, 437–447.
- Hochedlinger, K., Yamada, Y., Beard, C., Jaenisch, R., 2005. Ectopic expression of Oct-4 blocks progenitor-cell differentiation and causes dysplasia in epithelial tissues. *Cell* 121, 465–477.
- Hough, S.R., Clements, I., Welch, P.J., Wiederholt, K.A., 2006. Differentiation of mouse embryonic stem cells after RNA interference-mediated silencing of OCT4 and Nanog. *Stem Cells* 24, 1467–1475.
- Huang, B., Jiang, C., Zhang, R., 2014. Epigenetics: The language of the cell? *Epigenomics* 6, 73–88.
- Kuo, C.J., Conley, P.B., Chen, L., *et al.*, 1992. A transcriptional hierarchy involved in mammalian cell-type specification. *Nature* 355, 457–461.
- Li, Y., Wang, R., Qiao, N., *et al.*, 2017. Transcriptome analysis reveals determinant stages controlling human embryonic stem cell commitment to neuronal cells. *J. Biol. Chem.* 292, 19590–19604.
- Meshorer, E., Misteli, T., 2006. Chromatin in pluripotent embryonic stem cells and differentiation. *Nat. Rev. Mol. Cell Biol.* 7, 540–546.
- Moore, K., Persaud, T., 2008. *The Developing Human: Clinically Oriented Embryology*. Philadelphia: Saunders.
- Niwa, H., 2007. Open conformation chromatin and pluripotency. *Genes Dev.* 21, 2671–2676.

- Sanchez-Arrones, L., Cardozo, M., Nieto-Lopez, F., Bovolenta, P., 2012. Cdon and Boc: Two transmembrane proteins implicated in cell-cell communication. *Int. J. Biochem. Cell Biol.* 44, 698–702.
- Strachan, T., Read, A., 2004. *Human Molecular Genetic*, third ed. London and New York: Garland Science.
- Szutorisz, H., Dillon, N., 2005. The epigenetic basis for embryonic stem cell pluripotency. *Bioessays* 27, 1286–1293.
- Twyman, R., 2001. Signal transduction in development. In: *Instant Notes in Developmental Biology*. BIOS Scientific Publisher, Oxford.
- Youngblood, B., Hale, J.S., Ahmed, R., 2013. T-cell memory differentiation: Insights from transcriptional signatures and epigenetics. *Immunology* 139, 277–284.
- Zasso, J., Mastad, A., Cutarelli, A., Conti, L., 2018. Inducible alpha-synuclein expression affects human Neural Stem Cell behavior. *Stem Cells Dev.*

Characterization of Transcriptional Activities

Adison Wong, Singapore Institute of Technology, Singapore
Maurice HT Ling, Colossus Technologies LLP, Singapore, Singapore

© 2019 Elsevier Inc. All rights reserved.

Recent progress in DNA microarray and RNA sequencing have enabled large scale and high quality whole cell transcriptome analysis to be performed. When complemented with Gene Ontology, both technologies could rapidly isolate important genetic determinants for further validation by qPCR. Besides the measurement of transcription at the mRNA level, gene expressions can be quantified using reporter proteins; such as green fluorescent protein. The results of these studies are usually integrated into computational models to enable the prediction of gene expression at larger scale. The steps for doing this will be discussed in this review.

Experimental Techniques

In this section, three major experimental techniques for analyzing transcriptomic activities will be discussed, followed by a discussion on reporter proteins, which are instrumental for experimental validation of transcriptomic activity.

Reverse Transcription and qPCR. qPCR exploits the sensitivity and sequence specificity of PCR for gene expression analysis (Fig. 1) where transcriptional activities can be characterized as absolute transcripts abundance or relative quantitation between biological samples. qPCR had been used to validate results from high-throughput DNA microarray and RNA sequencing. An RNA stabilizing agent, RNAlater[®], is often added prior to cell disruption to protect RNA integrity during mRNA extraction process. TRIzol[®], a monophasic solution of phenol and guanidine isothiocyanate, is routinely used to extract total RNA. When chloroform is added to the samples homogenized in TRIzol[®], RNA is preferentially extracted into the top aqueous phase. Isopropanol is added to precipitate RNA from aqueous phase. Purified RNA in sterile, diethylpyrocarbonate (DEPC)-treated water, comprising mainly of rRNAs and tRNAs and

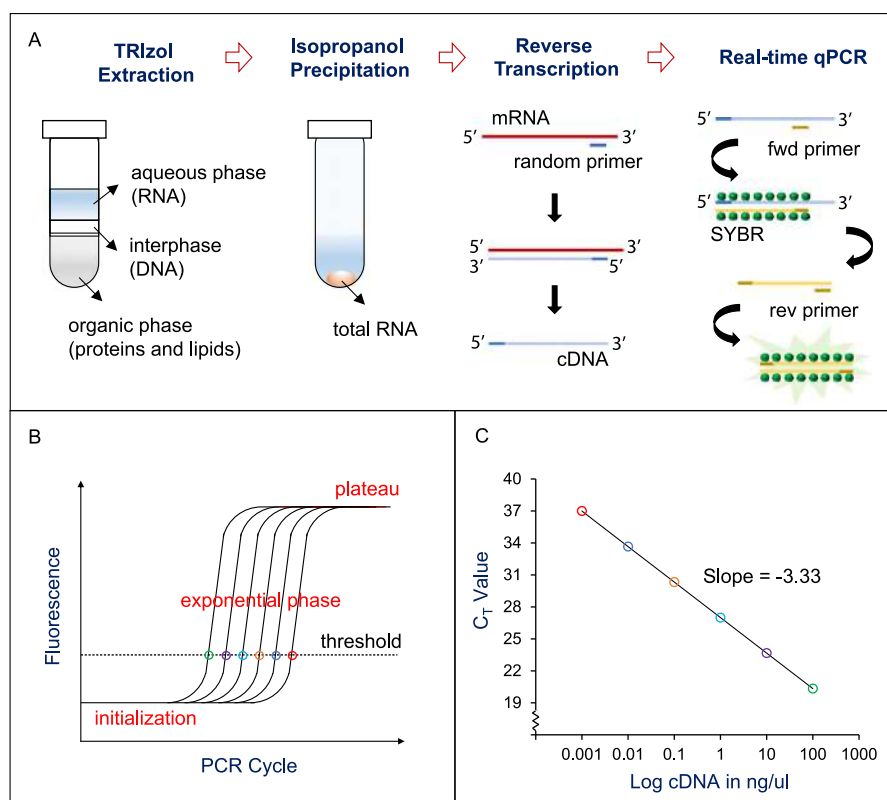


Fig. 1 qPCR analysis of targeted transcripts. (A) Sample preparation for qPCR. Total RNA is chemically extracted from biological sample and reverse transcribed into single-stranded cDNA molecules. Thereafter, the cDNA is used as templates for qPCR in real time thermocyclers. SYBR green bind to double-stranded cDNA during the extension process and fluorescence readings are recorded. (B) Sigmodal amplification of DNA during qPCR. A threshold is set above noise level in the exponential phase and used to determine the threshold cycle, Ct. (C) Relationship between cDNA concentration and Ct values under ideal condition, i.e., DNA exactly doubles with each PCR cycle. Hypothetical Ct values from (B) is reflected in (C) by colored circles.

about 1%–5% mRNA, is assayed for RNA integrity and quantity by spectrophotometer. Heuristically, the 260:280 nm ratio of the RNA preparation should be between 1.9 and 2.1 and that of 260:230 nm should be 2.0–2.2 as lower values may indicate the presence of contaminants. Distinct rRNA bands on agarose gel electrophoresis is indicative of good quality RNA. To generate cDNA templates for qPCR, the total RNA mRNA containing mRNA is reversed transcribed into single-stranded cDNA using random primers, dNTPs, buffer and reverse transcriptase. Depending on the expression level of the target gene, 1 ng–100 ng of total RNA may be required. Random 6–9mer primers are able to bind to any region in the RNA pool of most organisms. Other primers used for cDNA synthesis are oligo (dT)s which anneal to 3′-poly(A) tails, and sequence-specific primers to amplify targeted mRNA regions. Unlike random primers, oligo (dT)s are seldom used in prokaryotic cDNA synthesis as the poly(A) tails of prokaryotic mRNA are generally shorter (Sarkar, 1997). Although nonspecific primers may generate truncated cDNAs from internal mRNA sequence, gene expression results would not be much affected so long as qPCR primers are designed targeting the 5′-end of the mRNA. Sequence-specific primers for cDNA synthesis are commonly used for synthetic RNA standards or where analysis only involves a few gene targets.

In qPCR reaction, cDNA template, sequence-specific primers and nuclease-free water are added to qPCR master mix, comprising of DNA-binding dye (SYBR), dNTPs, magnesium and Taq polymerase, before importing into the qPCR thermocycler for gene expression measurement. qPCR could be performed in PCR tubes, 96-well or 384-well PCR plates. qPCR primers, typically 20–30 bp are designed to amplify approximately 60–150 bp of the sense strand of the target cDNA at the 5′-end. Amplicons above 150 bp usually suffers from poor PCR efficiency, resulting in gross underestimation on the number of transcripts. qPCR thermocycles are optically enabled to detect fluorescent signals. As qPCR cycle proceeds, double-stranded amplicons are generated, resulting in more PCR products for SYBR to bind which increases green fluorescence emission. Fluorescence intensity correlates directly with the amount of DNA amplicons, which is a proxy to the amount of cDNA template (mRNA transcripts). As a standard, gene expressions are calculated based on the threshold cycle (Ct) – the number of PCR cycles required for fluorescence to be at a critical value above background, which also indicates the start of PCR exponential phase. Ct values are inversely proportional to initial template amount. Variations in sampling procedures, RNA extraction, reverse transcription and qPCR reaction may significantly lead to bias and compounding error (Sanders *et al.*, 2014). To mitigate such bias, gene expression results are normalized to internal reference genes. One common practice is using constitutive housekeeping genes as the normalization reference, which are assumed to have constant expression (mRNA levels), despite several studies showing otherwise (Bustin, 2000; Dundas and Ling, 2012; Thellin *et al.*, 1999). To improve accuracy, target gene expression could be normalized to the geometric average internal control genes (Vandesompele *et al.*, 2002), which should be validated with statistical software such as geNorm, NormFinder and BestKeeper (Andersen *et al.*, 2004; Pfaffl *et al.*, 2004; Vandesompele *et al.*, 2002). The relative expression ratio of two samples could be determined using the generalized Pfaffl method as illustrated in Example 1 (Hellemans *et al.*, 2007; Pfaffl, 2001). To determine the absolute amount of mRNA transcript, a standard curve of threshold cycle number plotted against varying amount of cDNA standard, Ct versus log [cDNA], could be developed by performing qPCR on synthesized cDNA standards with six or more serial dilutions. Importantly, the synthesized standard and target cDNA should be similar in PCR efficiency, amplicon length and primer binding efficiency, ensuring that the standard curve is applicable for the gene of interest. Known amounts of synthetic RNA standards could also be spiked into the extracted RNA to account for reverse transcription bias. Recommendations on experimental design, results reporting and selection of reference genes are described in MIQE guidelines (Bustin *et al.*, 2009, 2010).

Example 1: Fold-expression Calculation. qPCR was performed to analyze how the gene expression of an efflux pump differed when cells were exposed to antibiotic. Results of the qPCR run is shown in Table 1. Here, three different housekeeping genes (Ref Gene A, B and C) were used as reference genes. “Treated” and “Control” represent the mRNA from cells with and without antibiotics treatment, respectively. For each gene, threshold cycles are the arithmetic mean of technical replicates.

The general formula for normalized relative quantities (NRQ) with multiple reference genes is,

$$NRQ = \frac{E_{GOI}^{\Delta Ct_{GOI}(\text{control}-\text{treated})}}{\sqrt[n]{\prod_0 E_{ref0}^{\Delta Ct_{ref0}(\text{control}-\text{treated})}}}$$

We assumed 100% efficient amplification and amplicon exactly doubles with every exponential phase cycle. This gives an efficiency value of 2 for both the GOI and reference genes,

$$NRQ = \frac{2^{(34.00-28.50)}}{\sqrt[3]{0.59 \times 0.84 \times 0.5}} = \frac{45.25}{0.63} = 71.83$$

The $2^{\Delta Ct}$ value of GOI is 45.25. Similarly, the $2^{\Delta Ct}$ values of reference genes A, B and C are 0.59, 0.84 and 0.5, respectively; giving the geometric mean of 0.63. Hence, the efflux pump gene increased by ~72-folds upon antibiotic exposure, implying gene up-regulation.

Table 1 qPCR analysis of gene encoding for efflux pump after exposure to antibiotics

	GOI	Ref Gene A	Ref Gene B	Ref Gene C
Efficiency	2	2	2	2
Treated	28.50	21.25	19.00	23.00
Control	34.00	20.50	18.75	22.00

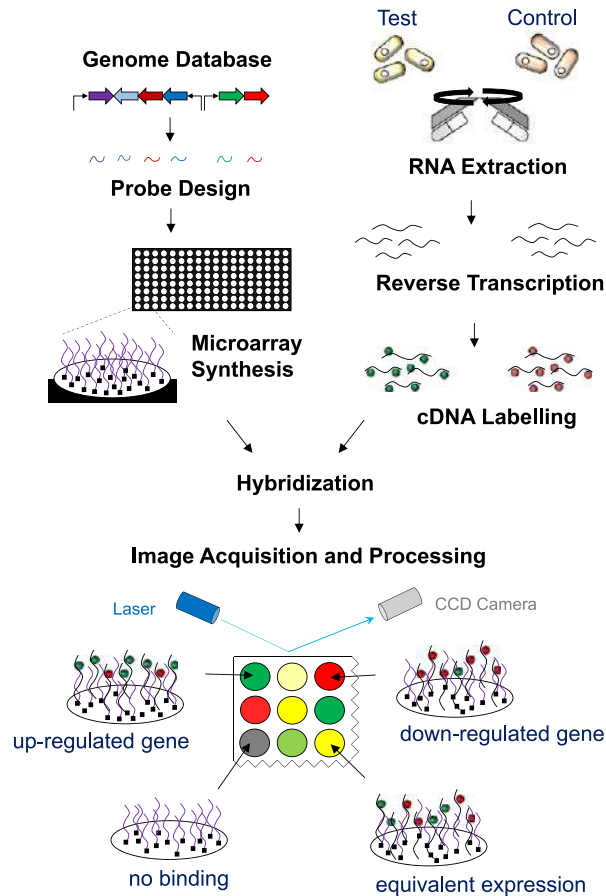


Fig. 2 Schematic of a DNA microarray workflow.

DNA Microarray. The DNA microarray is a structured 2D-array of short (25–80 bp), single-stranded picomoles of DNA molecules discretely spotted or synthesized on solid surfaces (Fig. 2). These DNA molecules, also known as oligo probes, are complementary to target gene sequences for hybridization between a fluorescent labeled target DNA and the immobilized probe upon contact. Experimentally, total RNA is isolated from both test and control biological samples, and assayed for microarray suitability using capillary electrophoresis and spectrophotometry. RNA quality can be assayed by Agilent 2100 Bioanalyzer and a RNA integrity number (RIN) value of 7 or higher is considered suitable for microarray experiment. Depending on the microarray technology, cell type and nature of the study, 200 ng–20 µg of total RNA (~1 µg/µl) may be required (Kang and Chang, 2012; Ling *et al.*, 2013; Trevino *et al.*, 2007). In Affymetrix arrays, mRNA are used to generate labeled cDNAs in a one pot reverse transcription reaction. Comparatively, in Agilent arrays, mRNAs are retro-transcribed into T7 RNA promoter tailed cDNAs before converted into labeled cRNAs by the Eberwine reaction (Van Gelder *et al.*, 1990). In either workflows, the cDNA and cRNA molecules are labeled using fluorescent labeled nucleotides. Common fluorescent dyes used are the red Cy5 and the green Cy3. Following the generation of single-stranded cDNA (cRNA) molecules, equal amounts of test and control cDNAs (cRNAs) are pooled together and hybridized on the microarray slides where competitive hybridization occurs between the test and control cDNAs (cRNAs), each tagged to different fluorescent dye. The ratio of red and green intensities of each probe spot corresponds to the relative abundance of the target mRNA transcripts. These can be measured by fluorescent scanner equipped with excitation lasers and digital camera for imaging. Images are overlapped and processed to convert spot intensities into numerical values. Yellow spots are observed when transcripts from both samples are of comparable abundance. In general, genes are considered differentially expressed when test and control hybridizations exhibit at least two-fold differences. Microarray results are usually reported using MIAME guidelines by the Functional Genomics Data Society (Brazma *et al.*, 2001).

Example 2: Statistical Analysis of Microarray Results. Two-sample microarrays, also known as two-channel microarrays are used to determine differential gene expressions between two samples; such as, healthy tissues versus diseased tissues (as shown in Fig. 2). It is common for a gene to be represented more than once within a microarray for statistical calculations. In this example, we assume a hypothetical microarray of only five genes and each gene is represented by five probes. The summarized results are given in Table 2. The t-statistic of each gene and its p-value can be calculated, where n is the number of probes per gene.

$$t - \text{statistic} = \frac{|\overline{\text{Healthy}}_{\text{Gene}(i)} - \overline{\text{Diseased}}_{\text{Gene}(i)}|}{s_{\text{Healthy}} / \sqrt{n}}$$

Table 2 Microarray summarized values and statistics

Gene name	Healthy tissue signal		Diseased tissue signal		<i>t</i> -statistic	<i>p</i> -value	Significance
	Mean intensity	Standard deviation	Mean intensity	Standard deviation			
Gene A	14.09	1.703	14.82	0.023	1.242	0.2821	Not significant
Gene B	13.64	0.642	15.43	0.375	4.983	0.0076	Significant
Gene C	14.14	0.053	15.99	0.796	18.002	5.6E-05	Significant
Gene D	8.18	0.836	9.73	1.487	3.785	0.0194	Not significant
Gene E	12.72	1.102	13.39	1.227	1.413	0.2303	Not significant

The threshold (α) for determining statistical significance is taken as 0.05. From this point of view, Gene D is significantly upregulated in diseased tissue compared to healthy tissue (p -value=0.0194). However, as there are multiple *t*-tests to be carried out (one test for each gene), the total threshold will increase – when 2 and 3 *t*-tests are carried out for an experiment, the threshold increases from 0.05 to 0.0975 ($1-0.95^2$) and 0.143 ($1-0.95^3$) respectively. Hence, there is a need to maintain the overall threshold at 0.05. Bonferroni correction is a simple method, which reduces the threshold to the quotient of 0.05 and the number of statistical tests required. Here, the Bonferroni corrected threshold is 0.01 (0.05/5) as there are 5 *t*-tests required. As a result, Gene D is not significantly upregulated in diseased tissue compared to healthy tissue.

RNA-Seq. In 1977, a method to decode extracted DNA sequences at single nucleotide level was developed by Nobel Prize chemist (Sanger *et al.*, 1977). This method, based on the selective incorporation of chain-terminating fluorescent dideoxynucleotide, was the core technology in the Human Genome Project (Schmutz *et al.*, 2004). Since then, a remarkable transformation is witnessed in the field of DNA sequencing with newly-developed chemistries and chip-based analytical platforms, giving rise to second and third generation sequencing technologies (Chen *et al.*, 2013; Heather and Chain, 2016; Schadt *et al.*, 2010; Voelkerding *et al.*, 2009). These led to higher throughput, sequencing depth and reliability, enabling applications beyond functional genomics, including the transcriptome analysis of prokaryotic and higher-order eukaryotic organisms, also known as RNA sequencing (RNA-Seq) (Croucher and Thomson, 2010; Mäder *et al.*, 2011; McGettigan, 2013). RNA-Seq measures transcript abundance by repeated, direct sequencing of reverse-transcribed cDNAs. Sequence data are mapped onto a reference genome and the number of mapped reads are normalized to quantify output expression as reads per kilobase of transcript per million mapped reads (RPKM) or transcripts per million (TPM) (Mortazavi *et al.*, 2008; Wagner *et al.*, 2012). Example 3 shows the different procedures to normalize raw transcripts output into RPKM or TPM. The normalization procedures are required because sequencing depth, i.e., the number of times a sample is read, and transcript length result in higher raw output readings on the detection platform. TPM provides a more straightforward breakdown of gene expression profile, although both normalization approaches are equally accepted by the scientific community (Wagner *et al.*, 2012).

Over the years, RNA-Seq has gradually displaced microarrays for transcriptomic analysis. Without the need to hybridize to transcript complementary probes, RNA-Seq provides significant advantages over microarrays. Firstly, microarray experimental design relied on the availability of known genome sequence for chip synthesis. When genome sequences are not available, the only way forward is by using chips originally intended for other closely related specie to capture cDNA. This approach may miss out on capturing the full transcriptome profile of the target due to sequence variability. In contrast, RNA-Seq could be performed for unknown samples in parallel with whole genome sequencing to access both functional genomics and transcriptome data. This greatly reduces experiment down time while also enables the identification of novel transcript isoforms (Trapnell *et al.*, 2010). A second advantage is the reliability of transcriptome data. RNA-Seq is not affected by the non-specific hybridization of cDNA to probes. Instead, multiple reads are performed on the same sample to ensure sequence reliability. This procedure allows transcription to be accurately mapped down to single nucleotide resolution, and could be particularly useful when characterizing transcriptomes with frequent repeats. As transcript abundance is measured by direct sequencing reads, RNA-Seq exhibits a larger dynamic range of detection, with better sensitivity at both low and high gene expression levels, whereas the upper detection limit of microarray is bounded by the optical sensitivity of microarray scanner under saturating condition (blinding effect). Despite the apparent advantages, there are a few considerations on the use of RNA-Seq, namely data management and storage, speed, and cost. RNA-Seq data are typically in the terabytes range while microarray data are in megabyte range. Hence, sequencing data is inherently more complicated to work with and would require a reliable suite of bioinformatics tools for data normalization and analysis (Trapnell *et al.*, 2012). However, a typical sequencing run takes between 2 and 14 days to complete while microarray experiment could be done within a day. Microarray consumables are currently substantially cheaper than RNA-Seq consumables. In the longer run, these concerns would likely be reconciled as new sequencing platforms with higher throughput, sequence accuracy and longer read lengths are developed. Ongoing efforts to correct experimental bias and consolidate best practices in RNA-Seq would eventually pave the way for dominance of this technology in high throughput transcriptomics study (Conesa *et al.*, 2016).

In the RNA-Seq workflow, total RNA is extracted and assayed for amount and quality on spectrophotometer and Agilent 2100 Bioanalyzer. RNA requirements for sequencing are more stringent than that for microarray; at least a RIN >8 for eukaryotic cells and >9.5 for prokaryotic cells should be satisfied. Illumina also recommends intact RNA in terms of the percentage of RNA fragments >200 nucleotides (DV₂₀₀) on Agilent 2100 Bioanalyzer. A few hundred nanograms total RNA with DV₂₀₀>30% would be appropriate for downstream application. Before RNA-Seq library preparation, it is prudent to

remove unwanted rRNA transcripts which competes with mRNA for sequencing capacity. For this purpose, terminator exonuclease (TEX), which degrades only rRNA substrates with 5'-monophosphate ends, could be added to the total RNA pool (Croucher *et al.*, 2009; Sharma *et al.*, 2010). Alternatively, commercial subtractive-hybridization kits could be used to deplete rRNAs. Oligonucleotide probes on magnetic beads support are used to capture rRNAs by hybridization, followed by subsequent ethanol extraction of mRNA from supernatant after beads pull down (Croucher *et al.*, 2009; O'Neil *et al.*, 2013; Petrova *et al.*, 2017; Stewart *et al.*, 2010; Westermann *et al.*, 2016). Depending on the sequencing equipment and the purpose of the study, different methods have been developed to prepare RNA-Seq cDNA libraries, which in turn determines if the directional information of transcriptional activities could be effectively captured (Croucher and Thomson, 2010). A common workflow involves fragmenting rRNA-depleted samples by ultra-sound sonication or nebulization into 200–400 bp fragments, followed by dephosphorylation on the 5'-end and ligation with sequencing adaptor on the 3'-end. Transcript 5'-ends before re-phosphorylated with T4 polynucleotide kinase and ligated with the second sequencing adaptor. Processed transcripts are reverse transcribed using 3'-adaptor as primer to generate first strand cDNA (Westermann *et al.*, 2016). Alternatively, rRNA-depleted samples could be 3'-polyadenylated with poly(A) polymerase and then treated with tobacco acid pyrophosphatase. This effectively converts 5'-triphosphate into monophosphate for 5'-adaptor ligation. Processed transcripts are reverse transcribed using oligo(dT)-adaptor primers to generate first strand cDNA (Westermann *et al.*, 2016). Adaptors are attached stepwise in specific orientation to preserve directional information. Besides mechanical fragmentation, transpososome could be used to fragment and incorporate modified adaptors onto doublestranded cDNA in a one pot reaction known as tagmentation (Adey *et al.*, 2010; Kia *et al.*, 2017). In this process, samples are reverse transcribed and amplified to generated double-stranded cDNA, using uracil instead of thymine for second strand synthesis. The resultant cDNA molecules are transposed with sequencing adaptors and treated with USER enzyme mix to deplete the second cDNA strand, retaining only tagged single stranded cDNA. This cDNA library is selectively amplified by Phusion polymerase (Phusion polymerase do not extent well on DNA templates with uracil residues) and purified (Gertz *et al.*, 2012).

The next step in RNA-Seq involves hybridizing cDNA libraries to solid surface, typically insoluble beads or glass slide, which contains oligonucleotides complementary to the adaptors. With sufficient dilution, each bead probabilistically capture only one cDNA molecule or sparsely distributed on the glass slide. Ion Torrent and SOLiD sequencing uses bead capture while Illumina uses glass slide in the form of a flow cell. Then, captured transcripts are clonally amplified by emulsion PCR into bead-bound libraries, or by isothermal bridge amplification into clusters of DNA clones. Finally,

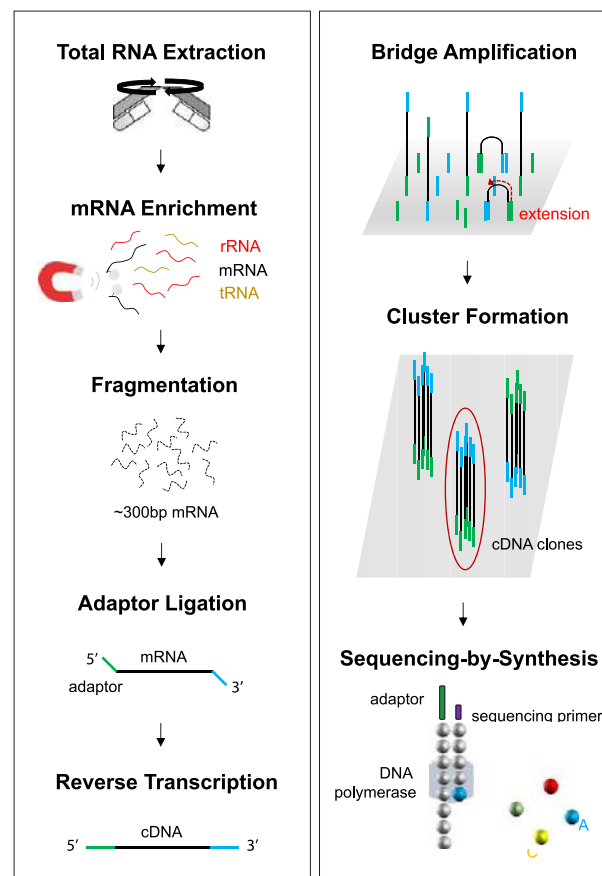


Fig. 3 Schematic of Illumina RNA-Seq workflow.

Table 3 Transcript abundance data from RNA-Seq

	Size (kb)	Mapped reads ($\times 10^6$)		
		Replicate I	Replicate II	Replicate III
Gene A	0.5	1.2	1.5	1.6
Gene B	1	3.0	3.2	2.9
Gene C	2	4.0	4.2	4.2
Gene D	3	2.0	1.8	2.2
Gene E	4	0.5	0.5	0.6

sequencing primers are added to initiate the actual sequencing process. Raw sequence reads are mapped to specific transcripts of the reference genome and normalized to provide accurate expression output data in RKPM or TPM. Currently, the Illumina sequencing-by-synthesis platform (SBS) is considered the most dominant next generation sequencing (NGS) technology (Fig. 3) (Greenleaf and Sidow, 2014). Other NGS platforms include (1) Ion Torrent that also works on the principle of SBS but detects H^+ proton instead of fluorescence during nucleotide addition, (2) ABI SOLiD that relies on ligation chemistry (synthesis-by-ligation), and (3) single molecule sequencing technologies from PacBio and Oxford Nanopore. For more information on the various NGS platforms, the reader is referred to comprehensive reviews published elsewhere (Heather and Chain, 2016; Mardis, 2017; Schadt *et al.*, 2010).

Example 3: RKPM and TPM Calculation. An RNA-Seq experiment was performed to determine the transcript abundance in a cell culture with three technical replicates. For simplicity, it is assumed that the cell has expressed 5 different genes. Results of the sequencing run are presented in Table 3 as raw transcript counts.

TPM is obtained after two normalization steps, first with gene size and then by sequencing depth. Sample calculation for Replicate I, Gene A with size of 0.5 kb.

$$\text{normalized seq counts} = \frac{\text{seq counts}}{\text{gene size}} = \frac{1.2}{0.5} = 2.40$$

			Mapped reads ($\times 10^6$)		
			Replicate I	Replicate II	Replicate III
1st normalization by gene size	Gene A	0.5	2.40	3.00	3.20
	Gene B	1	3.00	3.20	3.30
	Gene C	2	2.00	2.10	2.10
	Gene D	3	0.67	0.60	0.73
	Gene E	4	0.13	0.13	0.15
	Total count		8.19	9.03	9.48

Total sequencing depth of Replicate I after 1st normalization is 8.19 million reads.

$$\text{TPM} = \frac{\text{normalized seq counts}}{\text{total counts}} = \frac{2.40}{8.19} = 0.29$$

			Mapped reads in TPM		
			Replicate I	Replicate II	Replicate III
2nd normalization by sequencing depth	Gene A	0.5	0.29	0.33	0.34
	Gene B	1	0.37	0.35	0.35
	Gene C	2	0.24	0.23	0.22
	Gene D	3	0.08	0.07	0.08
	Gene E	4	0.02	0.01	0.02
	Total count		1.00	1.00	1.00

RKPM is obtained after two normalization steps in the reverse order of TPM, first with sequencing depth and then by gene size. Total sequencing depth of Replicate I before normalization is 10.7 million reads ($\text{Total seq counts} = 1.2 + 3.0 + 4.0 + 2.0 + 0.5 = 10.7$).

Sample calculation for Replicate I, Gene A (1st normalization by sequencing depth).

$$\frac{\text{seq counts}}{\text{total counts}} = \frac{1.2}{10.7} = 0.11$$

			Mapped reads ($\times 10^6$)		
			Replicate I	Replicate II	Replicate III
1st normalization by sequencing depth	Gene A	0.5	0.11	0.13	0.13
	Gene B	1	0.28	0.29	0.28
	Gene C	2	0.37	0.38	0.35
	Gene D	3	0.19	0.16	0.18
	Gene E	4	0.05	0.04	0.05
	Total count		1.00	1.00	1.00

2nd normalization by gene size for the same gene, $RKPM = \frac{\text{normalized seq. counts}}{\text{gene size}} = \frac{0.11}{0.5} = 0.22$

			Mapped reads in RKPM		
			Replicate I	Replicate II	Replicate III
2nd normalization by gene size	Gene A	0.5	0.22	0.27	0.27
	Gene B	1	0.28	0.29	0.28
	Gene C	2	0.19	0.19	0.18
	Gene D	3	0.06	0.05	0.06
	Gene E	4	0.01	0.01	0.01
	Total count		0.75	0.81	0.80

Note that RKPM can be converted to TPM by dividing by the total RKPM values. Sample calculation for Replicate I, Gene A with RKPM value of 0.22, $TPM = \frac{RKPM}{\sum RKPM} = \frac{0.22}{0.75} = 0.29$.

Reporter Proteins. Reporter proteins are fluorescent proteins, or enzymes that can convert specific substrates into colorimetric or luminescent products. To quantify gene expression, genes that encode for reporter proteins are assembled downstream of DNA promoters and transformed into the biological host of choice. This approach has been adopted to characterize promoter strengths in diverse genetic context, including bacterial, yeast, plant, mammalian cells, and cell-free extracts (Auslander *et al.*, 2012; Canton *et al.*, 2008; Chappell *et al.*, 2013; Huang *et al.*, 2010; Redden and Alper, 2015; Reeve *et al.*, 2016; Schaumberg *et al.*, 2016; Wong *et al.*, 2015). The amount of reporter proteins generated during gene expression is assumed to be proportional to background-subtracted readings measured on optical instruments; such as, flow cytometry and fluorescent plate readers. Reporter proteins could also be purified and quantitated to obtain calibration plots. Direct quantification of gene expression by reporter proteins circumvent the need to isolate mRNA from cell cultures, a procedure that is often time consuming, labor intensive, costly and prone to human error. For this reason, the use of reporter proteins is preferred over qPCR for quantifying gene expression, especially for the large-scale validation of DNA promoters. Importantly, results from this approach reflects the combined effect of transcription, translation and post-translational activities, rather than mRNA transcription alone. Incorporating the results of reporter protein outputs into computational models could give estimate of the amount of transcript produced. This will require key modeling parameters, including the rates of mRNA degradation, protein maturation, protein degradation, and cell growth (Canton *et al.*, 2008; Milo *et al.*, 2010). Reporter proteins could also be used for functional analysis. For example, promoter sequences could be mutated and then assayed for changes in gene expression. When complemented with DNA sequencing, this procedure could rapidly identify core promoter regulatory regions and develop synthetic promoter libraries of varying strengths (Alper *et al.*, 2005; Hartner *et al.*, 2008; Rud *et al.*, 2006; Siegl *et al.*, 2013).

Due to the ease of use, fluorescent proteins are among the most commonly used reporter proteins in gene expression studies. The superfolder green fluorescent protein (sfGFP) in particular, is a variant of wild type GFP with unmatched performance in terms of folding kinetics, solubility and tolerance to high temperature and chemicals (Pédélec *et al.*, 2006). A major limitation of GFP and its variants is the need for molecular oxygen as cofactor during fluorophore formation; thus, restricting its application to aerobic organisms. Flavin mononucleotide-based fluorescent proteins (FbFP) was developed for use in both aerobic and anaerobic condition but the low output fluorescence remains a technical hurdle to widespread adoption (Drepper *et al.*, 2007). For anaerobic organisms, non-fluorescent reporters such as light emitting luciferases and absorbance changing enzymes are still commonplace. Popular examples of enzyme-based reporter proteins include β -galactosidase, β -glucosidase and β -glucuronidase (Partow *et al.*, 2010; Siegl *et al.*, 2013). The recent development of Spinach2, an RNA aptamer that binds to 3,5-difluoro-4-hydroxybenzylidene-imidazolinone (fluorophore mimic of GFP), provides an exciting platform to study gene expression at the mRNA level *in vivo* (Strack *et al.*, 2013).

Bioinformatics Analysis/Applications

Bioinformatics can be applied to the study of transcriptional activities in two major ways – (1) the prediction of gene expression using sequence features (such as, response elements in promoters) or from the expression of other genes, and (2) constructing mathematical models for further analyses.

Predicting Gene Expressions. In prokaryotes, functionally related genes are often regulated by a common promoter and a set of enhancers, forming an operon. Hence, expression of genes within the same operon tends to be correlated (Sabatti *et al.*, 2002). This is supported by a study (De Hoon *et al.*, 2004) showing that genes within operons show higher correlation than genes across operons. Using expressional correlation, De Hoon *et al.* (2014) can predict whether the genes are from the same operon, with 79.9% accuracy. This concept of expressional correlation from genes within the same operon has been used to improve microarray analysis by noise reduction (Xiao *et al.*, 2006). However, the relative position of the genes within the operon may affect expression. A study on *Streptomyces coelicolor* operons found that the expression of genes decreases along the relative position on the operon by normalizing expression of all genes in the operon to that of the first gene in the operon (Laing *et al.*, 2006) but this expression decline is not statistically significant up to the 5th gene on the operon but this is not observed in *Escherichia coli* where the expression of all genes in the operon is constant regardless of position. However, a study using synthetic operon found that gene expression increases linearly with the distance from the start of a gene to the end of the operon (Lim *et al.*, 2011). Nevertheless, these studies suggest that expression of genes within the same operon can be predicted by knowing the expression of one of the genes within the operon, subjected to species-specific variability as in the case of *S. coelicolor*.

In both prokaryotes and eukaryotes, several genes may be activated by the same transcription factor. Like the reason why genes within the same operon is likely to be expressional correlated, a set of genes activated by the same set of transcription factors are also likely to be expressional correlated. Binding of transcription factors to the promoter may affect chromatin accessibility (Lamparter *et al.*, 2017), leading to correlated expression patterns from genes affected by the same transcription factors (Mahdevar *et al.*, 2013). A study (Zhang and Li, 2017) examining more than 1000 human transcription factors found that transcription factor usage can be used to predict gene expression level (r^2 of up to 0.617). Genes with similar sequence features in the promoter sequence may also exhibit correlated expression profiles. For example, it has been shown that genes with dioxin-response element in their promoter demonstrate correlated expression (Kim *et al.*, 2006). Similar cases have also been discovered in other response elements; such as, immune response (Care *et al.*, 2015). It is plausible to conceive that several genes must be expressed in response to a given stimulus (Kim *et al.*, 2003), and the regulatory signaling of such stimulus may involve the same set of transcription factors or response elements.

This is supported by a previous study examining the functional aspects of genes with correlated expressions and found that genes with correlated expression demonstrate functional correspondence (Reverter *et al.*, 2005). Moreover, a study on transcriptomics (Ling *et al.*, 2008) also found large number of genes that are correlated. Hence, it may be possible to predict entire transcriptome from known expression of a handful of genes by constructing gene co-expression network. This concept is demonstrated by a study (Ling and Poh, 2014), which uses the expression values of 59 genes to predict the expression of the entire *E. coli* transcriptome. The correlation between predicted and actual expression value is 0.467, which is similar to the microarray intra-array variation. This suggests that intra-array variation accounts for a substantial portion of the transcriptome prediction error and further strengthen the potential of this approach with more reliable experimental data. Ling and Poh (2014) demonstrated the application of their predictive model using a case study – hydrogenase 2 maturation endopeptidase (hybD) can affect the efficiency of hydrogenase 2, a critical enzyme in hydrogen production during glucose (Maeda *et al.*, 2007) or glycerol fermentation (Trchounian *et al.*, 2013). The model predicted 87 genes significantly affected by the 56% knockdown of hybD, and was subjected to Gene Ontology Enrichment Analysis (Zheng and Wang, 2008). All 5 significant molecular functions enriched were of carbon/sugar transferase-typed activity, which corresponds to the expected activity as previously described (Maeda *et al.*, 2007; Trchounian *et al.*, 2013). Knowing that 56% knockdown of hybD significantly affects the expression of 87 genes, it is then possible to ask for the range of hybD expression variation that will not significantly affect any other genes; that is, the expression buffer of hybD. Using 10% stepwise changes of *hybD* from 100% knockout to 2x over-expression the number of affected genes is symmetrical and fits a quadratic model and solving the roots of the quadratic model, the expressional buffer of hybD in *E. coli* MG1655 is estimated to be 73.88% and 124.52% from its average expression (Fig. 4).

Two sequence features of the gene, GC content and codon usage, have been found to be useful in predicting gene expressions. In a study of more than four thousand human promoters across eight different cell lines (Landolin *et al.*, 2010), it was found that the GC content of promoters can be predictive of gene expression as promoters high in GC content (more than 50% GC) demonstrating constitutive expression and promoters with low GC content (less than 50% GC) more inclined towards cell-specific expression. This resulted in significant correlation ($r=0.43$) between promoter activity and reporter gene expression. Of the 789 genes expressed in all eight cell lines, 719 (91%) were genes with high GC promoters and only 70 (9%) were from genes with low GC promoters. Conversely, 378 of the 483 genes (78.3%) that were expressed in only one of the eight cell lines were from low GC promoters. This suggests that GC content of human promoters can be indicative of cell specificity.

However, the relative impact of GC content and codon usage on gene expression and transcript stability is a subject of debate. A study (Barahimipour *et al.*, 2015) attempted to resolve this by designing 4 yellow fluorescent protein (YFP) genes encoding the same amino acid sequence for expression in *Chlamydomonas reinhardtii* using differing GC content and codon usages. Using this, Barahimipour *et al.* (2015) found that codon usage plays a more important role in expression efficiency and transcript stability compared to GC content in *C. reinhardtii*. Expression efficiency is contributed by translational efficiency as the relative abundance between RNA and protein are highly correlated using RNA blot analysis, suggesting that expression efficiency is a result of transcriptional efficiency and the codon usage plays an important role in transcriptional efficiency. Moreover, this study also adds strength to the usefulness of transcriptomics analysis as a proxy for proteomics analysis as it depends on high correlation between transcript abundance and peptide abundance, given that it is experimentally easier to conduct transcriptomics studies (using experimental tools such as microarrays and RNA-seq) than proteomics studies (using experimental tools such as mass spectrometry).

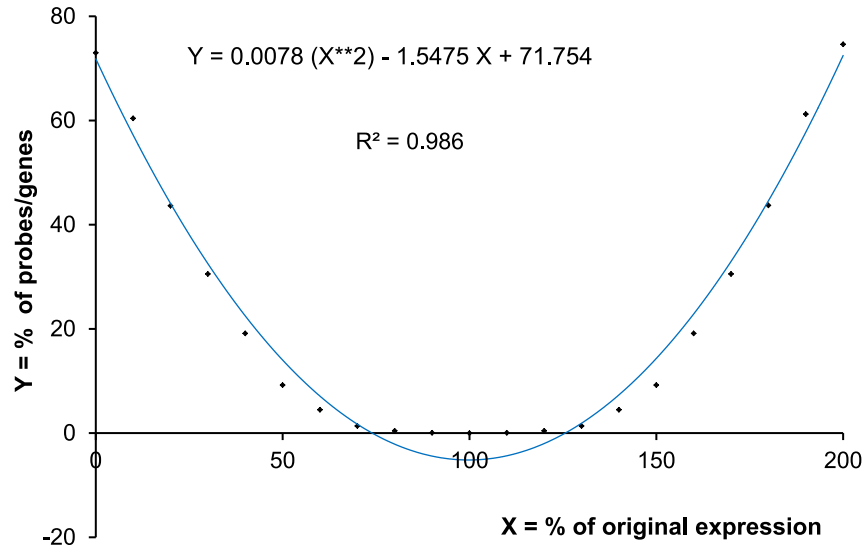


Fig. 4 Percentage of reachable genes affected by varying levels of hydrogenase 2 maturation endopeptidase (hybD) expressions.

There are several metrics to calculate codon usages bias. These include codon adaptation index (Sharp and Li, 1987), relative codon bias strength (Das et al., 2012), relative codon adaptation (Fox and Erill, 2010), and modified relative codon bias strength (Sahoo and Das, 2014). Fox and Erill (2010) used relative codon usage bias to predict the expression levels of *E. coli* genes of more than 1000 bp, achieving a correlation of 0.489 between predicted and actual expression. Similarly, a study (Das et al., 2017) found correlation between these metrics but modified relative codon bias strength was found to have the highest correlation with transcriptomics ($r = -0.31$) and proteomics ($r = -0.44$). Although the correlations are not high, these studies do suggest the potential of using codon usage bias to predict transcriptional activities.

Modeling and Simulation. The main reason for studying transcriptional activities of a cell is to use it as a proxy for peptide/protein abundance. This in turn allows for a door towards elucidating the genetic and biochemical basis of various conditions. For example, it is the fundamental step for analyzing differential gene expression between various conditions (such as, disease versus healthy, or different treatment conditions) or across various time points (Creecy and Conway, 2015; Du et al., 2017; Herrera-Marcos et al., 2017; Kato et al., 2005; Lin et al., 2016; Liu et al., 2015), or to construct models to predict the phenotype given a transcriptomic change (transcriptome-phenotype correlation).

A biochemical reaction is commonly written as (where A, B, and C are the substrates; P, Q, and where A, B, and C are the substrates; P, Q, and R are the products are the products), with the following rate laws,

$$A \xrightarrow{\text{Enzyme}} P \text{ rate law 1} = \frac{k_{cat}[\text{Enzyme}][A]}{K_m + [\text{Enzyme}][A]}$$

$$B + C \xrightarrow{\text{Enzyme}} Q + R \text{ rate law 2} = \frac{k_{cat}[\text{Enzyme}][B][C]}{K_m + [\text{Enzyme}][B][C]}$$

Where k_{cat} and K_m are the turnover number (per unit time) and Michaelis-Menten constant (concentration) of the enzyme respectively, and $k_{cat}[\text{Enzyme}]$ corresponds to the V_{max} of the reaction (per unit time). When there is more than one substrate or product, the mechanism can take more complicated forms; such as ternary-complex mechanisms (Yang et al., 2011) or ping-pong mechanisms (Nakamura et al., 1994). However, these more complicated mechanisms will be unwieldy to be extended to enzymes using more than two substrates. In addition, these mechanisms will require a larger set of enzyme kinetics which is not easily obtainable. Hence, we propose to use an approximation where we consider the enzyme can only work when all the required substrates are present at the same location and with a random probability. Hence, the approximated rate law in generalized form can be written as

$$\frac{k_{cat}[\text{enzyme}](\prod_{i=1}^N [\text{substrate}_i])}{K_m + (\prod_{i=1}^N [\text{substrate}_i])}$$

The concentration of enzymes, which can be directly represented by the transcriptional activity of the gene encoding the enzyme, play a crucial role in the rate laws. The concentrations of both substrate(s) and product(s) over time can be defined as ordinary differential equations (ODEs) as

$$\frac{d[A]}{dt} = -\text{rate law 1} \quad \frac{d[P]}{dt} = \text{rate law 1}$$

$$\frac{d[B]}{dt} = \frac{d[C]}{dt} = -\text{rate law 2} \quad \frac{d[P]}{dt} = \frac{d[Q]}{dt} = \text{rate law 2}$$

As enzymes are the result of gene expression, the concentration of an enzyme can be modelled using ODEs developed in previous studies (Jayaraman *et al.*, 2016; Saeidi *et al.*, 2011). Briefly, expressed protein concentration from an inducible promoter can be modelled as

$$\text{For constitutive expression: } \frac{d[RNA]}{dt} = v_0 - \gamma_{RNA}[RNA]$$

$$\text{For inducible expression: } \frac{d[RNA]}{dt} = v_0 + \frac{v_{max} + [inducer]^n}{K_m + [inducer]^n} - \gamma_{RNA}[RNA]$$

$$\text{For repressible expression: } \frac{d[RNA]}{dt} = v_0 + \frac{v_{max}}{K_m + [repressor]^n} - \gamma_{RNA}[RNA]$$

$$\frac{d[Enzyme]}{dt} = \frac{d[protein]}{dt} = \beta[RNA] - \gamma_{protein}[protein]$$

Where v_0 and v_{max} are the baseline and maximum expression (concentration per second) of the promoter, K_m is the Michaelis-Menten constant (concentration), n is the Hill coefficient, β is the relative ribosome binding site (RBS) strength (Wang *et al.*, 2017), and γ_{RNA} and $\gamma_{protein}$ are the degradation rates (per unit time) of RNA and protein respectively.

Once created, models can be used as both a repository of knowledge and an *in silico* platform for hypothesis testing and analysis (Ling, 2016) for the purpose of advising the experimentalist on the parameters to optimize for the optimal yield (MacDonald *et al.*, 2011). This approach had been used in several studies. For example, an enzyme pathway kinetic model of mevalonate pathway had been constructed to study the crucial steps for the production of amorphaadiene from mevalonate in *E. coli* (Weaver *et al.*, 2015). After model construction, global sensitivity analysis was carried out to determine which of the six enzymes are important and found that amorphaadiene synthase expression and activity are most critical. Another study attempted to increase carotenoid production in maize also constructed an enzyme kinetic model (Comas *et al.*, 2016). While amorphaadiene from mevalonate is a linear pathway (Weaver *et al.*, 2015), many carotenoids can be synthesized from phytoene in a branched pathway with many enzymes (Comas *et al.*, 2016). Therefore, it is not obvious which enzyme expression to target to increase the yield of specific carotenoids. By analyzing the model, four independent maize lines were engineered using insights from the model analysis and validated experimentally.

These studies suggest that modeling and simulation can provide important computational tools to advise experimentalists but the success of these tools depends on the quality of characterization data from previous experiments. The quality of experimental data can be improved using bioinformatics approaches to reduce noise (Xiao *et al.*, 2006). Hence, there is a reflective process between laboratory and experimental experimentation. It can then be expected that increased experimental data will improve current computational methods, and improved computational methods will better inform laboratory experiments. This trend is likely to stay in the foreseeable future.

See also: Analyzing Transcriptome-Phenotype Correlations. Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Characterizing and Functional Assignment of Noncoding RNAs. Codon Usage. Comparative Transcriptomics Analysis. Detecting and Annotating Rare Variants. Epigenetics: Analysis of Cytosine Modifications at Single Base Resolution. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Pipeline of High Throughput Sequencing. Prediction of Coding and Non-Coding RNA. Regulation of Gene Expression. Transcriptome Analysis. Transcriptome During Cell Differentiation. Whole Genome Sequencing Analysis

References

- Adey, A., Morrison, H.G., Asan, *et al.*, 2010. Rapid, low-input, low-bias construction of shotgun fragment libraries by high-density *in vitro* transposition. *Genome Biol.* 11, R119.
- Alper, H., Fischer, C., Nevoigt, E., Stephanopoulos, G., 2005. Tuning genetic control through promoter engineering. *Proc. Natl. Acad. Sci. USA* 102, 12678–12683.
- Andersen, C.L., Jensen, J.L., Orntoft, T.F., 2004. Normalization of real-time quantitative reverse transcription-PCR data: A model-based variance estimation approach to identify genes suited for normalization, applied to bladder and colon cancer data sets. *Cancer Res.* 64, 5245–5250.
- Auslander, S., Auslander, D., Muller, M., Wieland, M., Fussenegger, M., 2012. Programmable single-cell mammalian biocomputers. *Nature* 487.
- Barahimipour, R., Strenkert, D., Neupert, J., *et al.*, 2015. Dissecting the contributions of GC content and codon usage to gene expression in the model alga *Chlamydomonas reinhardtii*. *Plant J.* 84, 704–717.
- Brazma, A., Hingamp, P., Quackenbush, J., *et al.*, 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat. Genet.* 29, 365–371.
- Bustin, S.A., 2000. Absolute quantification of mRNA using real-time reverse transcription polymerase chain reaction assays. *J. Mol. Endocrinol.* 25, 169–193.
- Bustin, S.A., Beaulieu, J.-F., Huggett, J., *et al.*, 2010. MIQE précis: Practical implementation of minimum standard guidelines for fluorescence-based quantitative real-time PCR experiments. *BMC Mol. Biol.* 11, 74.

- Bustin, S.A., Benes, V., Garson, J.A., *et al.*, 2009. The MIQE guidelines: Minimum information for publication of quantitative real-time PCR experiments. *Clin. Chem.* 55, 611–622.
- Canton, B., Labno, A., Endy, D., 2008. Refinement and standardization of synthetic biological parts and devices. *Nat. Biotechnol.* 26.
- Care, M.A., Westhead, D.R., Tooze, R.M., 2015. Gene expression meta-analysis reveals immune response convergence on the IFN γ -STAT1-IRF1 axis and adaptive immune resistance mechanisms in lymphoma. *Genome Med.* 7, 96.
- Chappell, J., Jensen, K., Freemont, P.S., 2013. Validation of an entirely in vitro approach for rapid prototyping of DNA regulatory elements for synthetic biology. *Nucleic Acids Res.* 41, 3471–3481.
- Chen, F., Dong, M., Ge, M., *et al.*, 2013. The history and advances of reversible terminators used in new generations of sequencing technology. *Genom. Proteom. Bioinform.* 11, 34–40.
- Comas, J., Benfeitas, R., Vilaprinyo, E., *et al.*, 2016. Identification of line-specific strategies for improving carotenoid production in synthetic maize through data-driven mathematical modeling. *Plant J.* 87, 455–471.
- Conesa, A., Madrigal, P., Tarazona, S., *et al.*, 2016. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17, 13.
- Creedy, J.P., Conway, T., 2015. Quantitative bacterial transcriptomics with RNA-seq. *Curr. Opin. Microbiol.* 23, 133–140.
- Croucher, N.J., Fookes, M.C., Perkins, T.T., *et al.*, 2009. A simple method for directional transcriptome sequencing using Illumina technology. *Nucleic Acids Res.* 37, e148.
- Croucher, N.J., Thomson, N.R., 2010. Studying bacterial transcriptomes using RNA-seq. *Curr. Opin. Microbiol.* 13, 619–624.
- Das, S., Chottopadhyay, B., Sahoo, S., 2017. Comparative analysis of predicted gene expression among Crenarchaeal genomes. *Genom. Inform.* 15, 38–47.
- Das, S., Roymondal, U., Chottopadhyay, B., Sahoo, S., 2012. Gene expression profile of the cyanobacterium *Synechocystis* genome. *Gene* 497, 344–352.
- De Hoon, M.J.L., Imoto, S., Kobayashi, K., Ogasawara, N., Miyano, S., 2004. Predicting the operon structure of *Bacillus subtilis* using operon length, intergene distance, and gene expression information. *Pac. Symp. Biocomput.* 276–287.
- Drepper, T., Eggert, T., Circolone, F., *et al.*, 2007. Reporter proteins for in vivo fluorescence without oxygen. *Nat. Biotechnol.* 25, 443–445.
- Dundas, J., Ling, M.H., 2012. Reference genes for measuring mRNA expression. *Theory Biosci.* 131, 1–9.
- Du, Y., Zhao, B., Liu, Z., *et al.*, 2017. Molecular subtyping of pancreatic cancer: Translating genomics and transcriptomics into the clinic. *J. Cancer* 8, 513–522.
- Fox, J.M., Erill, I., 2010. Relative codon adaptation: A generic codon bias index for prediction of gene expression. *DNA Res.* 17, 185–196.
- Gertz, J., Varley, K.E., Davis, N.S., *et al.*, 2012. Transposase mediated construction of RNA-seq libraries. *Genome Res.* 22, 134–141.
- Greenleaf, W.J., Sidow, A., 2014. The future of sequencing: Convergence of intelligent design and market Darwinism. *Genome Biol.* 15, 303.
- Hartner, F.S., Ruth, C., Langenegger, D., *et al.*, 2008. Promoter library designed for fine-tuned gene expression in *Pichia pastoris*. *Nucleic Acids Res.* 36, e76.
- Heather, J.M., Chain, B., 2016. The sequence of sequencers: The history of sequencing DNA. *Genomics* 107, 1–8.
- Helleman, J., Mortier, G., De Paep, A., Speleman, F., Vandesompele, J., 2007. qBase relative quantification framework and software for management and automated analysis of real-time quantitative PCR data. *Genome Biol.* 8, R19.
- Herrera-Marcos, L.V., Lou-Bonafonte, J.M., Arnal, C., Navarro, M.A., Osada, J., 2017. Transcriptomics and the Mediterranean diet: A systematic review. *Nutrients* 9.
- Huang, H.-H., Camsund, D., Lindblad, P., Heidorn, T., 2010. Design and characterization of molecular tools for a Synthetic Biology approach towards developing cyanobacterial biotechnology. *Nucleic Acids Res.* 38, 2577–2593.
- Jayaraman, P., Devarajan, K., Chua, T.K., *et al.*, 2016. Blue light-mediated transcriptional activation and repression of gene expression in bacteria. *Nucleic Acids Res.* 44, 6994–7005.
- Kang, A., Chang, M.W., 2012. Identification and reconstitution of genetic regulatory networks for improved microbial tolerance to isooctane. *Mol. Biosyst.* 8, 1350–1358.
- Kato, H., Saito, K., Kimura, T., 2005. A perspective on DNA microarray technology in food and nutritional science. *Curr. Opin. Clin. Nutr. Metab. Care* 8, 516–522.
- Kia, A., Gloeckner, C., Onothpraporn, T., *et al.*, 2017. Improved genome sequencing using an engineered transposase. *BMC Biotechnol.* 17, 6.
- Kim, B.-R., Hu, R., Keum, Y.-S., *et al.*, 2003. Effects of glutathione on antioxidant response element-mediated gene expression and apoptosis elicited by sulforaphane. *Cancer Res.* 63, 7520.
- Kim, W.K., In, Y.-J., Kim, J.-H., *et al.*, 2006. Quantitative relationship of dioxin-responsive gene expression to dioxin response element in Hep3B and HepG2 human hepatocarcinoma cell lines. *Toxicol. Lett.* 165, 174–181.
- Laing, E., Mersini, V., Smith, C.P., Hubbard, S.J., 2006. Analysis of gene expression in operons of *Streptomyces coelicolor*. *Genome Biol.* 7, R46.
- Lamparter, D., Marbach, D., Rueedi, R., Bergmann, S., Kutalik, Z., 2017. Genome-wide association between transcription factor expression and chromatin accessibility reveals regulators of chromatin accessibility. *PLOS Comput. Biol.* 13, e1005311.
- Landolin, J.M., Johnson, D.S., Trinklein, N.D., *et al.*, 2010. Sequence features that drive human promoter function and tissue specificity. *Genome Res.* 20, 890–898.
- Lim, H.N., Lee, Y., Hussein, R., 2011. Fundamental relationship between operon organization and gene expression. *Proc. Natl. Acad. Sci. USA* 108, 10626–10631.
- Ling, M., 2016. Of (biological) models and simulations. *MOJ Proteom. Bioinform.* 3, 00093.
- Ling, H., Chen, B., Kang, A., Lee, J.-M., Chang, M.W., 2013. Transcriptome response to alkane biofuels in *Saccharomyces cerevisiae*: Identification of efflux pumps involved in alkane tolerance. *Biotechnol. Biofuels* 6, 95.
- Ling, M.H., Lefevre, C., Nicholas, K.R., 2008. Filtering microarray correlations by statistical literature analysis yields potential hypotheses for lactation research. *Python Pap.* 3, 4.
- Ling, M.H., Poh, C.L., 2014. A predictor for predicting *Escherichia coli* transcriptome and the effects of gene perturbations. *BMC Bioinform.* 15, 140.
- Lin, M., Lachman, H.M., Zheng, D., 2016. Transcriptomics analysis of iPSC-derived neurons and modeling of neuropsychiatric disorders. *Mol. Cell. Neurosci.* 73, 32–42.
- Liu, Y., Ai, N., Liao, J., Fan, X., 2015. Transcriptomics: A sword to cut the Gordian knot of traditional Chinese medicine. *Biomark. Med.* 9, 1201–1213.
- MacDonald, J.T., Barnes, C., Kitney, R.I., Freemont, P.S., Stan, G.B., 2011. Computational design approaches and tools for synthetic biology. *Integr. Biol. (Camb.)* 3, 97–108.
- Mäder, U., Nicolas, P., Richard, H., Bessi eres, P., Aymerich, S., 2011. Comprehensive identification and quantification of microbial transcriptomes by genome-wide unbiased methods. *Curr. Opin. Biotechnol.* 22, 32–41.
- Maeda, T., Sanchez-Torres, V., Wood, T.K., 2007. Enhanced hydrogen production from glucose by metabolically engineered *Escherichia coli*. *Appl. Microbiol. Biotechnol.* 77, 879–890.
- Mahdevar, G., Nowzari-Dalini, A., Sadeghi, M., 2013. Inferring gene correlation networks from transcription factor binding sites. *Genes Genet. Syst.* 88, 301–309.
- Mardis, E.R., 2017. DNA sequencing technologies: 2006–2016. *Nat. Protoc.* 12, 213–218.
- McGettigan, P.A., 2013. Transcriptomics in the RNA-seq era. *Curr. Opin. Chem. Biol.* 17, 4–11.
- Milo, R., Jorgensen, P., Moran, U., Weber, G., Springer, M., 2010. BioNumbers – The database of key numbers in molecular and cell biology. *Nucleic Acids Res.* 38, D750–D753.
- Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B., 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods* 5, 621–628.
- Nakamura, A., Haga, K., Yamane, K., 1994. The transglycosylation reaction of cyclodextrin glucanotransferase is operated by a Ping-Pong mechanism. *FEBS Lett.* 337, 66–70.
- O’Neil, D., Glowatz, H., Schlumpberger, M., 2013. Ribosomal RNA depletion for efficient use of RNA-seq capacity. *Curr. Protoc. Mol. Biol.* (Chapter 4, Unit 4.19).
- Partow, S., Siewers, V., Bj rn, S., Nielsen, J., Maury, J., 2010. Characterization of different promoters for designing a new expression vector in *Saccharomyces cerevisiae*. *Yeast* 27, 955–964.
- P delacq, J.-D., Cabantous, S., Tran, T., Terwilliger, T.C., Waldo, G.S., 2006. Engineering and characterization of a superfolder green fluorescent protein. *Nat. Biotechnol.* 24, 79–88.
- Petrova, O.E., Garcia-Alcalde, F., Zampaloni, C., Sauer, K., 2017. Comparative evaluation of rRNA depletion procedures for the improved analysis of bacterial biofilm and mixed pathogen culture transcriptomes. *Sci. Rep.* 7, 41114.
- Pfaffl, M.W., 2001. A new mathematical model for relative quantification in real-time RT-PCR. *Nucleic Acids Res.* 29, e45.

- Pfaffl, M.W., Tichopad, A., Prgomet, C., Neuvians, T.P., 2004. Determination of stable housekeeping genes, differentially regulated target genes and sample integrity: BestKeeper—Excel-based tool using pair-wise correlations. *Biotechnol. Lett.* 26, 509–515.
- Redden, H., Alper, H.S., 2015. The development and characterization of synthetic minimal yeast promoters. *Nat. Commun.* 6, 7810.
- Reeve, B., Martinez-Klimova, E., de Jonghe, J., Leak, D.J., Ellis, T., 2016. The *Geobacillus* plasmid set: A modular toolkit for thermophile engineering. *ACS Synth. Biol.* 5, 1342–1347.
- Reverter, A., Barris, W., Moreno-Sanchez, N., *et al.*, 2005. Construction of gene interaction and regulatory networks in bovine skeletal muscle from expression data. *Aust. J. Exp. Agric.* 45, 821–829.
- Rud, I., Jensen, P.R., Naterstad, K., Axelsson, L., 2006. A synthetic promoter library for constitutive gene expression in *Lactobacillus plantarum*. *Microbiology (Reading)* 152, 1011–1019.
- Sabatti, C., Rohlin, L., Oh, M.-K., Liao, J.C., 2002. Co-expression pattern from DNA microarray experiments as a tool for operon prediction. *Nucleic Acids Res.* 30, 2886–2893.
- Saeidi, N., Wong, C.K., Lo, T.-M., *et al.*, 2011. Engineering microbes to sense and eradicate *Pseudomonas aeruginosa*, a human pathogen. *Mol. Syst. Biol.* 7, 521.
- Sahoo, S., Das, S., 2014. Analyzing gene expression and codon usage bias in diverse genomes using a variety of models. *Curr. Bioinform.* 9, 102–112.
- Sanders, R., Mason, D.J., Foy, C.A., Huggett, J.F., 2014. Considerations for accurate gene expression measurement by reverse transcription quantitative PCR when analysing clinical samples. *Anal. Bioanal. Chem.* 406, 6471–6483.
- Sanger, F., Nicklen, S., Coulson, A.R., 1977. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA* 74, 5463–5467.
- Sarkar, N., 1997. Polyadenylation of mRNA in prokaryotes. *Annu. Rev. Biochem.* 66, 173–197.
- Schadt, E.E., Turner, S., Kasarskis, A., 2010. A window into third-generation sequencing. *Hum. Mol. Genet.* 19, R227–R240.
- Schauberg, K.A., Antunes, M.S., Kassaw, T.K., *et al.*, 2016. Quantitative characterization of genetic parts and circuits for plant synthetic biology. *Nat. Methods* 13, 94–100.
- Schmutz, J., Wheeler, J., Grimwood, J., *et al.*, 2004. Quality assessment of the human genome sequence. *Nature* 429, 365–368.
- Sharma, C.M., Hoffmann, S., Darfeuille, F., *et al.*, 2010. The primary transcriptome of the major human pathogen *Helicobacter pylori*. *Nature* 464, 250–255.
- Sharp, P.M., Li, W.H., 1987. The codon Adaptation Index – A measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res.* 15, 1281–1295.
- Siegl, T., Tokovenko, B., Myronovskiy, M., Luzhetskyy, A., 2013. Design, construction and characterisation of a synthetic promoter library for fine-tuned gene expression in actinomycetes. *Metab. Eng.* 19, 98–106.
- Vandesompele, J., De Preter, K., Pattyn, F., *et al.*, 2002. Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. *Genome Biol.* 3.RESEARCH0034.
- Stewart, F.J., Ottlesen, E.A., DeLong, E.F., 2010. Development and quantitative analyses of a universal rRNA-subtraction protocol for microbial metatranscriptomics. *ISME J.* 4, 896–907.
- Strack, R.L., Disney, M.D., Jaffrey, S.R., 2013. A superfolder Spinach2 reveals the dynamic nature of trinucleotide repeat-containing RNA. *Nat. Methods* 10, 1219–1224.
- Thellin, O., Zorzi, W., Lakaye, B., *et al.*, 1999. Housekeeping genes as internal standards: Use and limits. *J. Biotechnol.* 75, 291–295.
- Trapnell, C., Roberts, A., Goff, L., *et al.*, 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and cufflinks. *Nat. Protoc.* 7, 562–578.
- Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28, 511–515.
- Trchounian, K., Soboh, B., Sawers, R.G., Trchounian, A., 2013. Contribution of hydrogenase 2 to stationary phase H₂ production by *Escherichia coli* during fermentation of glycerol. *Cell Biochem. Biophys.* 66, 103–108.
- Trevino, V., Falciani, F., Barrera-Saldaña, H.A., 2007. DNA microarrays: A powerful genomic tool for biomedical and clinical research. *Mol. Med.* 13, 527–541.
- Van Gelder, R.N., von Zastrow, M.E., Yool, A., *et al.*, 1990. Amplified RNA synthesized from limited quantities of heterogeneous cDNA. *Proc. Natl. Acad. Sci. USA* 87, 1663–1667.
- Voelkerding, K.V., Dames, S.A., Durtschi, J.D., 2009. Next-generation sequencing: From basic research to diagnostics. *Clin. Chem.* 55, 641–658.
- Wagner, G.P., Kin, K., Lynch, V.J., 2012. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory Biosci.* 131, 281–285.
- Wang, H., Ling, M.H., Chua, T.K., Poh, C.L., 2017. Two cellular resource-based models linking growth and parts characteristics aids the study and optimisation of synthetic gene circuits. *Eng. Biol.* 1, 30–39.
- Weaver, L.J., Sousa, M.M.L., Wang, G., *et al.*, 2015. A kinetic-based approach to understanding heterologous mevalonate pathway function in *E. coli*. *Biotechnol. Bioeng.* 112, 111–119.
- Westermann, A.J., Förstner, K.U., Amman, F., *et al.*, 2016. Dual RNA-seq unveils noncoding RNA functions in host-pathogen interactions. *Nature* 529, 496–501.
- Wong, A., Wang, H., Poh, C.L., Kitney, R.I., 2015. Layering genetic circuits to build a single cell, bacterial half adder. *BMC Biol.* 13, 40.
- Xiao, G., Martinez-Vaz, B., Pan, W., Khodursky, A.B., 2006. Operon information improves gene expression estimation for cDNA microarrays. *BMC Genom.* 7, 87.
- Yang, Y., Yamashita, T., Nakamaru-Ogiso, E., *et al.*, 2011. Reaction mechanism of single subunit NADH-ubiquinone oxidoreductase (Ndi1) from *Saccharomyces cerevisiae*: Evidence for a ternary complex mechanism. *J. Biol. Chem.* 286, 9287–9297.
- Zhang, L.-Q., Li, Q.-Z., 2017. Estimating the effects of transcription factors binding and histone modifications on gene expression levels in human cells. *Oncotarget* 8, 40090–40103.
- Zheng, Q., Wang, X.-J., 2008. GOEAST: A web-based software toolkit for Gene Ontology enrichment analysis. *Nucleic Acids Res.* 36, W358–W363.

Survey of Antisense Transcription

Maurice HT Ling, Colossus Technologies, LLP, Singapore and University of Melbourne, Melbourne, VIC, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Antisense transcript refers to RNA transcripts that are complementary to the RNA transcript from a known gene. It has been shown that antisense transcription is widespread throughout eukaryotic genome with occurrence as high as 43% (Györfy *et al.*, 2006). Terminologically speaking, the RNA transcript transcribed from a gene is then known as the sense transcript, which will be used as template for translation. In terms of nomenclature, a common way to name antisense transcript is to prefix or suffix “AS” to its sense transcript; for example, the antisense transcript of interleukin-1- α (IL-1 α) is known as AS-IL-1 α (Chan *et al.*, 2015) whereas the antisense transcript of topomyosin I (TPM1) is known as TPM1-AS (Huang *et al.*, 2017). The first of natural antisense transcripts discovered was found in the b2 gene region of coliphage lambda (Bovre and Szybalski, 1969). An early example of antisense transcript found (Mizuno *et al.*, 1984) is complementary to the 5'-end of Outer Membrane Protein F (OmpF) RNA transcript. Upon hybridization, two consequences may occur. Firstly, the translation of sense transcript may be restricted as the abundance of single-stranded sense transcript decreases. In other cases, the splicing process may be interfered. For example, Zeb2 sense transcript hybridizes with its antisense transcript and reduces the efficiency of ribosome-binding. This results in translational inefficiency, leading to effective down-regulation of the peptide (Beltran *et al.*, 2008). Secondly, half-life of the sense transcript is likely to decrease as double-stranded RNA can be degraded by DICER (Fruscoloni *et al.*, 2003), or short antisense transcript may be incorporated into RNA-induced silencing complex (RISC) to target the complementary sense transcript for degradation (Ibrahim *et al.*, 2006).

In both cases, the abundance of resulting peptide/protein product can be modulated in the presence of antisense transcript. Moreover, it has also been suggested that sense and antisense transcripts may be independently regulated (Goyal *et al.*, 2017); hence, adding another layer of translational regulation. It has also been shown that antisense transcripts may play a significant role in human oncogenesis (Balbin *et al.*, 2015); thus, emphasizing antisense transcription as an important field of study.

However, antisense transcripts can be broadly divided into cis-antisense transcripts, where the antisense transcript originates from the same genomic locus as its sense counterpart, and trans-antisense transcript, where the antisense transcript originates from a different genomic locus as its sense counterpart. In this article, we will examine both cis- and trans-natural antisense transcription before exploring several known cases of natural antisense transcription and their effects.

Cis- and Trans-Antisense Transcription

The mRNA, which is used for translation, is transcribed from the antisense strand of the coding sequence of the DNA; hence, has the same sequence as the sense transcript. Cis-antisense transcript is then transcribed from the sense strand of the coding sequence; as a result, is complementary to the mRNA. Trans-antisense transcripts, on the other hand, refers to antisense transcripts that originate from a different gene locus but happen to share sequence similarity for RNA-RNA duplex formation. Compared to an estimated 20% occurrence in cis-antisense transcription (Ling *et al.*, 2013), trans-antisense transcription occurs about 4% (Li *et al.*, 2008).

There can be four possible orientations (Fig. 1) that can result in RNA-RNA hybrids between cis-antisense/sense transcript pairs (Wang *et al.*, 2005). Of these four different orientations, head-to-head and tail-to-tail, also known as divergent and convergent respectively (Villegas and Zaphiropoulos, 2015), are common orientations (Lavorgna *et al.*, 2004; Osato *et al.*, 2007). Head-to-head orientation (Fig. 1(A)) is likely to result in occlusion of the ribosome-binding site (Osato *et al.*, 2007) and this results in reduced efficiency of ribosomes to bind to the mRNA transcript to start translation. This in turn leads to reduced rate of peptide formation. However, it is also possible that head-to-head sense/antisense can lead to increased peptide levels in some cases. For example, antisense transcript of mouse ubiquitin carboxy-terminal hydrolase L1 (*Uchl1*) gene can activate polysomes for increased translation of *Uchl1* under specific stress conditions (Carrieri *et al.*, 2012), resulting in unilateral increase in peptide level without the corresponding increase in mRNA level. Tail-to-tail orientation (Fig. 1(B)) is commonly found in sense/antisense pairs with either similar polyadenylation tails (Gu *et al.*, 2009; Zubko *et al.*, 2011) or 3' untranslated regions (Swaminathan and Beilharz, 2016). When tail-to-tail sense and antisense RNA are transcribed at the same time, this can result in collision and stalling of RNA polymerase at the 3' end, which allows the free 5' end of the antisense transcript to bind to nascent sense transcripts in the nucleus (Dang *et al.*, 2016). In the cytoplasm, tail-to-tail sense/antisense duplex can also result in Dicer-2 processing, leading to silencing of both sense and antisense transcripts (Russo *et al.*, 2016). Full overlap (Fig. 1(C)), known as embedded, occurs when the antisense transcript overlaps with the full length of the sense transcript, resulting in complete masking and protection of the sense transcript by RNase activity (Rosikiewicz and Makołowska, 2016). RNase masking occurs whenever RNA-RNA or RNA-DNA duplexes are formed; hence, can occur in all four orientations. For example, interferon- α 1 mRNA increases its stability after being partially masked by its natural antisense transcript (Kimura *et al.*, 2013). Hence, it is plausible to consider that full overlapping antisense transcript may reduce RNase degradation to a greater level than partial overlaps (head-to-head, tail-to-tail, and internal).

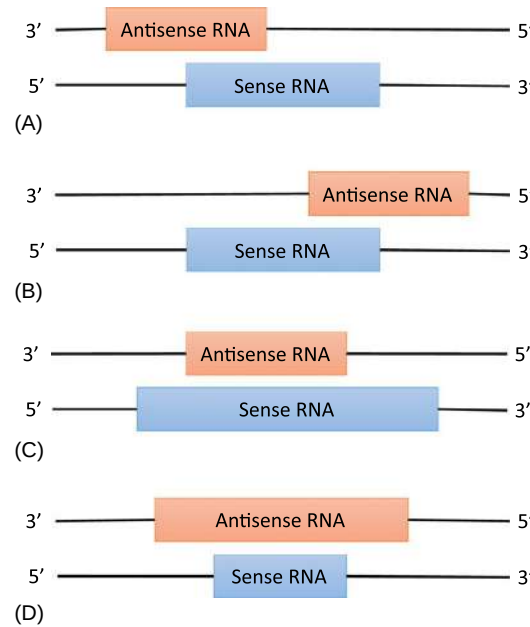


Fig. 1 Four possible orientations of cis-antisense/sense pairs. (A) 5' to 5' Overlap (Head-to-head), (B) 3' to 3' Overlap (Tail-to-tail), (C) Full Overlap, (D) Internal Overlap.

Detection of antisense transcripts. There are three main methods for detecting cis-antisense transcripts; namely, hybridization method (Collani and Barcaccia, 2012), antisense microarray method (Ge *et al.*, 2008), and RNA-seq identification method (Li *et al.*, 2013). Using the principle that double stranded DNA is protected against exonuclease degradation the same way that double stranded RNA is protected against RNase degradation, Collani and Barcaccia (2012) propose to reverse transcribe total RNA into cDNA for hybridization, before degradation of single strand cDNA by exonucleases. Undegraded hybridized cDNA molecules represents sense/antisense pairs, which can be identified by sequencing. Ge *et al.* (2008) removed a step from standard microarray protocol. This step elimination ensures that cDNA from antisense transcripts, instead of cDNA from sense transcripts, is complementary to the microarray probes; hence, able to use commercial microarrays to detect for antisense transcription. However, commercial microarrays is generally dependent on the sequencing status and quality of expressed sequence tags (ESTs) for a particular species. Hence, the availability of microarrays is limited to commonly studies species, which limits its use in antisense detection in new species. Next generation sequencing (NGS) is able to overcome this limitation by *ab initio* sequencing of entire RNA pool (which is commonly known as RNA-sequencing or RNA-seq), including antisense transcripts. This is the basis of NASTIseq (Li *et al.*, 2013), which uses strand-specific RNA sequencing (ssRNA-seq) of Arabidopsis and statistical modelling to identify thousands of potential cis-antisense transcripts. Furthermore, it has been well-acknowledge (Gluck *et al.*, 2016; Green *et al.*, 2017; Kukurba and Montgomery, 2015; Merrick *et al.*, 2013; Miller *et al.*, 2014; Zhai *et al.*, 2014) that RNA-seq provides higher sensitivity and resolution, and reduced biases, compared to microarray technology. Thus, RNA-seq is currently the preferred method for antisense detection (Oono *et al.*, 2017; Shao *et al.*, 2017) due to its increasing employment and gradual cost reduction.

Detection of cis-antisense transcripts differs from that of trans-antisense transcripts. For trans-antisense transcripts, an identification pipeline has been proposed (Li *et al.*, 2008). Briefly, trans-antisense/sense pairs can be detected by pair-wise sequence comparisons using either databases of known antisense transcripts; such as, NATsDB (Zhang *et al.*, 2007); or RNA-seq data sets; to identify high-quality potential sense/antisense pairs before *in silico* assessment of hybridization potential using DINAMelt (Markham and Zuker, 2005). Although NATsDB focuses on cis-antisense, it is also possible that these antisense transcripts may exhibit sufficient sequence similarity to act as antisense for sense transcripts of other gene loci; thus, making them trans-antisense transcripts as well. This is used in the detection of trans-antisense transcripts of soybeans (Zheng *et al.*, 2013). NATpipe (Yu *et al.*, 2016) is a computational tool to discover NATs from RNA-seq data by identifying potential pairs of complementary transcripts; thus, making it useful for identifying both cis- and trans-antisense transcripts. As an example, NATpipe was used to analyse RNA-seq data of *Dendrobium officinale*, a non-model plant species, and identified more than 600 thousand potential antisense transcript pairs; of which, 436 were identified as potential cis-antisense transcript pairs.

Due to the large numbers of computationally identified antisense transcripts over the years, several databases catering to antisense transcripts had been built. For example, antiCODE (see "Relevant Websites section") collated published antisense listings of more than 10 species, including mouse, human, and rice. PlantNATsDB (Chen *et al.*, 2012) (see "Relevant Websites section") had collated more than a million antisense transcripts from 70 plant species. It can be expected that existing antisense transcript databases will grow and antisense transcripts will be a feature in organism-specific databases.

Antisense is not transcriptional noise. Despite the prevalence of antisense transcription (Györfy *et al.*, 2006) and proposed biological functions (Werner, 2005; Werner and Sayer, 2009; Werner and Swan, 2010; Werner *et al.*, 2009), there were early concerns whether

antisense transcription is a form of transcriptional noise (Ling *et al.*, 2013; Werner and Berdal, 2005) due to leaky transcription. It is known that evolutionary pressure favours sequence conservation (Hutter *et al.*, 2010; Keng *et al.*, 2014; Paulsen, 2011; Siafakas *et al.*, 2005) and several studies examining the expression of sense transcripts (Bergmiller *et al.*, 2012; Jordan *et al.*, 2005) found that expression conservation is a feature between orthologous genes. Hence, the presence of expression conservation in orthologous antisense transcripts suggests evolutionary pressure and precludes the concept that antisense transcription is solely transcriptional noise. Ling *et al.* (2013) studied orthologous antisense transcripts and found evidence of expression conservation among orthologous antisense transcripts despite lower expression conservation compared to sense transcripts. This suggests the presence of evolutionary pressure in antisense expression, implying that antisense transcription is not entirely transcriptional noise. This is corroborated by a study (Nielsen *et al.*, 2014) demonstrating conservation of antisense expression in humans and how antisense transcription may be regulated (Agarwal *et al.*, 2014; Uprety *et al.*, 2016). Hence, antisense transcription is recognized as a regulatory mechanism despite weaker regulation compared to sense transcripts (Kapusta and Feschotte, 2014), which leads to further studies in various medical fields; such as, skeletal (Huynh *et al.*, 2017) and cardiovascular (Rühle and Stoll, 2016) research.

Differing Effects of Antisense Transcription

Although it is generally considered that antisense transcripts result in down-regulation of effective level of the corresponding sense transcripts, leading to reduced peptide level; this is a heuristic with many exceptions. In addition, although the expression level of antisense transcript is generally around 10-folds lower than its sense counterpart (Villegas and Zaphiropoulos, 2015) and there is positive correlation between sense expression and antisense expression, Pearson's correlation averages at 0.44 (Ling *et al.*, 2013). This is supported by another study (Balbin *et al.*, 2015) showing overall Spearman's correlation of 0.28 and Pearson's correlation of 0.41 between head-to-head sense/antisense pairs in human. On the other hand, cases of negative correlation between sense and antisense expression has been found (Xu *et al.*, 2011). In this section, four cases are reviewed to exemplify varying and non-down-regulation impacts of antisense – (1) as-ssm1 increases the transcript level of ssm1 but has no impact on the protein level of ssm1 (Ostrowski and Saville, 2017), (2) TALAM1 (the antisense of MALAT1) increases the level of MALAT1 protein by increasing the maturation rate of MALAT1 transcript (Zong *et al.*, 2016), (3) Sox4 antisense transcripts have no effect on Sox4 mRNA or peptide level (Ling *et al.*, 2016), and (4) AS1DHRS1 silences DHRS4 gene cluster by chromatin modification (Li *et al.*, 2012). Taken together, this suggests that down-regulation of peptide levels is not the sole impact of antisense transcript. Rather, the effects of antisense transcript on the eventual peptide/protein level can be complex and should be considered on a case-by-case basis.

Ustilago maydis is a fungus that infects the corn, *Zea mays*, resulting in a fungal disease called smut. *U. maydis* mitochondrial seryl-tRNA synthetase (ssm1) is a crucial enzyme for mitochondrial function and is encoded by 2448 bases. The antisense of ssm1, as-ssm1, is a tail-to-tail antisense transcript of 2057 bases (Ostrowski and Saville, 2017). The 5' end of as-ssm1 extends 20 bases into the 3'-UTR of ssm1 transcript. Hence, as-ssm1 overlaps the entire 3' end of ssm1 transcript except 411 bases of its 5' end (83.2% overlap). Using quantitative PCR, it was found that the formation of ssm1/as-ssm1 double stranded RNA duplex protects ssm1 transcript from S1 nuclease degradation but this protection does not extend to the non-duplexed 411 bases at the 5' end of ssm1. Hence, the steady-state concentration of ssm1 is significantly increased as a result of as-ssm1 protection, leading to significant differences in growth rate and virulence of *U. maydis*. However, the protein abundance of ssm1 is not significantly different in the presence of as-ssm1 using standardized Bradford Assay. This suggests that the impact of as-ssm1 is not entirely on the level of ssm1 protein but may act as a preservation role to extend the half-life of ssm1 RNA transcript, leading to rapid translation of ssm1 transcript without undergoing transcription.

Maturation of metastasis-associated lung adenocarcinoma transcript 1 (MALAT1) RNA requires 3' cleavage, resulting in a non-polyadenylated transcript. The antisense transcript of MALAT1 is known as TALAM1 (Zong *et al.*, 2016), which hybridize to MALAT1 at the 3' end. In human genome, hMALAT1 is 8708 bases and hTALAM1 overlaps from 1551th base to the 3' end of hMALAT1 with 963 bases of the 5' end of hTALAM1 extending from the 3' end of hMALAT1. Although the expression of MALAT1 is correlated to the expression of TALAM1 across different cell types, quantitative PCR shows that the TALAM1 transcripts in HeLa cells is about 290-times less abundant than MALAT1 transcripts. Using actinomycin D chase experiment, TALAM1 half-life drops from about 8 hours to about 3 hours in MALAT1-depleted cells (depleted using phosphorothioate internucleosidic linkage-modified DNA antisense oligonucleotides) compared to control cells. This suggests that MALAT1 confers protection to its antisense transcript despite a protection ratio of only 0.34% due to the relative abundance between MALAT1 and TALAM1. The reverse is also true – TALAM1-depleted cells shows significant reduction in the abundance of mature 3' non-polyadenylated MALAT1 RNA transcript while the abundance of nascent 3' polyadenylated MALAT1 RNA transcript is significantly increased. This suggests that MALAT1/TALAM1 sense/antisense pairs are co-reliant to maintain half-life and TALAM1 has an important role in processing and maturation of MALAT1 transcripts.

Sry-related high mobility group box 4 (Sox4) is a 47 kDa transcription factor, which binds to WWCAAW sequence, and is involved in DNA binding/bending, protein interactions and nuclear import/export. More than 20 antisense transcripts of varying lengths and from different transcription start sites have been found for Sox4 in E15.5 mouse cerebral cortex using 5' RACE-Southern hybridization (Ling *et al.*, 2016). All of them are cytoplasmically co-localized using fluorescent *in situ* hybridization (FISH), suggesting that these antisense transcripts are exported out of the nucleus; hence, potentially active. However, over-expression of Sox4 antisense transcripts did not result in significant differences in the mRNA and peptide level of Sox4, which contradicted FISH results. It is known that RNA-RNA duplex, which can result from sense/antisense hybridization, may serve as

template for processing into small RNA molecules and function via RNA interference (RNAi) machinery. A small RNA molecule, Sox4_sir3, has been found to originate from sense/antisense duplex and found to play a role in embryonic brain development. This suggests that the effects of antisense transcripts may go beyond duplex formation.

Short-chain dehydrogenase/reductase family member 4 (DHRS4) on human chromosome 14q11.2 encodes the NADPH dependent retinol dehydrogenase/reductase (NRDR), which is important in metabolism and detoxification of carbonyl compounds. The antisense transcript of DHRS4, AS1DHRS4, is a head-to-head spliced antisense transcript overlapping with exon 1 of DHRS4 (Li *et al.*, 2012). DHRS4 mRNA level and NRDR protein level increase after using antisense oligonucleotides to AS1DHRS4 to reduce effective transcript abundance of AS1DHRS4, suggesting that AS1DHRS4 abundance is inversely proportional to the abundance DHRS4 mRNA, which implies that AS1DHRS4 transcript reduces the effective level of DHRS4 transcript. In addition, AS1DHRS4 duplexing with DHRS4 transcript results in histone modification and DNA methylation as revealed by ChIP assays. These led to the reduction of transcriptional activity of DHRS4 gene.

The above cases demonstrate the variety of means in which antisense transcript can affect the abundance and half-life of the sense transcript which may or may not lead to significant changes in the abundance of resulting peptide/protein. The reverse can also be true – antisense transcript may affect the abundance of resulting peptide/protein without significant changes to the abundance of the sense transcript. Hence, suggesting a separation between transcription and translation. More importantly, these cases illustrate that antisense transcript can function at various levels. At the first order level, antisense transcript forms RNA-RNA duplex with its sense transcript but this does not lead to a uniform reduction of effective sense transcript. At higher order, antisense expression may lead to a range of other effects; such as, RNA interference and changes at the epigenetic level; which must be considered on a case-by-case basis.

Collectively, this illustrates the myriad of roles played by antisense transcripts. It is clear the roles played by antisense goes beyond initial expectations – reducing the abundance of sense transcripts. Even in this role, the methods used by each sense-antisense system is likely to differ significantly. Therefore, presenting antisense as an important area for on-going study. It is highly plausible that antisense transcription may be a field of study in its own rights as antisense is currently being explored for therapeutic purposes (Chery, 2016; Zarouchlioti *et al.*, 2018).

See also: Analyzing Transcriptome-Phenotype Correlations. Applications of Ribosomal RNA Sequence and Structure Analysis for Extracting Evolutionary and Functional Insights. Characterization of Transcriptional Activities. Characterizing and Functional Assignment of Noncoding RNAs. Comparative Transcriptomics Analysis. Exome Sequencing Data Analysis. Functional Enrichment Analysis. Genome-Wide Scanning of Gene Expression. Integrative Analysis of Multi-Omics Data. Natural Language Processing Approaches in Bioinformatics. Next Generation Sequencing Data Analysis. Regulation of Gene Expression. Transcriptome Analysis. Transcriptome During Cell Differentiation. Whole Genome Sequencing Analysis

References

- Agarwal, T., Roy, S., Kumar, S., Chakraborty, T.K., Maiti, S., 2014. In the sense of transcription regulation by G-quadruplexes: Asymmetric effects in sense and antisense strands. *Biochemistry* 53, 3711–3718.
- Balbin, O.A., Malik, R., Dhanasekaran, S.M., *et al.*, 2015. The landscape of antisense gene expression in human cancers. *Genome Res.* 25, 1068–1079.
- Beltran, M., Puig, I., Peña, C., *et al.*, 2008. A natural antisense transcript regulates Zeb2/Sip1 gene expression during Snail1-induced epithelial-mesenchymal transition. *Genes Dev.* 22, 756–769.
- Bergmiller, T., Ackermann, M., Silander, O.K., 2012. Patterns of evolutionary conservation of essential genes correlate with their compensability. *PLOS Genet.* 8, e1002803.
- Bovre, K., Szybalski, W., 1969. Patterns of convergent and overlapping transcription within the b2 region of coliphage lambda. *Virology* 38, 614–626.
- Carrieri, C., Cimatti, L., Biagioli, M., *et al.*, 2012. Long non-coding antisense RNA controls Uchl1 translation through an embedded SINEB2 repeat. *Nature* 491, 454–457.
- Chan, J., Atianand, M., Jiang, Z., *et al.*, 2015. Cutting edge: A natural antisense transcript, AS-IL1 α , controls inducible transcription of the proinflammatory cytokine IL-1 α . *J. Immunol.* 195, 1359–1363.
- Chen, D., Yuan, C., Zhang, J., *et al.*, 2012. PlantNATsDB: A comprehensive database of plant natural antisense transcripts. *Nucleic Acids Res.* 40, D1187–D1193.
- Chery, J., 2016. RNA therapeutics: RNAi and antisense mechanisms and clinical applications. *Postdoc. J.* 4, 35–50.
- Collani, S., Barcaccia, G., 2012. Development of a rapid and inexpensive method to reveal natural antisense transcripts. *Plant Methods* 8, 37.
- Dang, Y., Cheng, J., Sun, X., Zhou, Z., Liu, Y., 2016. Antisense transcription licenses nascent transcripts to mediate transcriptional gene silencing. *Genes Dev.* 30, 2417–2432.
- Fruscoloni, P., Zamboni, M., Baldi, M.I., Tocchini-Valentini, G.P., 2003. Exonucleolytic degradation of double-stranded RNA by an activity in *Xenopus laevis* germinal vesicles. *Proc. Natl. Acad. Sci. USA* 100, 1639–1644.
- Ge, X., Rubinstein, W.S., Jung, Y.C., Wu, Q., 2008. Genome-wide analysis of antisense transcription with Affymetrix exon array. *BMC Genom.* 9, 27.
- Gluck, C., Min, S., Oyelakin, A., *et al.*, 2016. RNA-seq based transcriptomic map reveals new insights into mouse salivary gland development and maturation. *BMC Genom.* 17, 923.
- Goyal, A., Fiškin, E., Gutschner, T., *et al.*, 2017. A cautionary tale of sense-antisense gene pairs: Independent regulation despite inverse correlation of expression. *Nucleic Acids Res.*
- Green, R., Wilkins, C., Ferris, M.T., Gale, M., 2017. RNA-seq in the collaborative cross. *Methods Mol. Biol.* 1488, 251–263.
- Gu, R., Zhang, Z., DeCervo, J.N., Carmichael, G.G., 2009. Gene regulation by sense-antisense overlap of polyadenylation signals. *RNA* 15, 1154–1163.
- Györfy, A., Tulassay, Z., Surowiak, P., Györfy, B., 2006. Computational analysis reveals 43% antisense transcription in 1182 transcripts in mouse muscle. *DNA Seq.* 17, 422–430.
- Huang, G.-W., Zhang, Y.-L., Liao, L.-D., Li, E.-M., Xu, L.-Y., 2017. Natural antisense transcript TPM1-AS regulates the alternative splicing of tropomyosin I through an interaction with RNA-binding motif protein 4. *Int. J. Biochem. Cell Biol.* 90, 59–67.
- Hutter, B., Bieg, M., Helms, V., Paulsen, M., 2010. Imprinted genes show unique patterns of sequence conservation. *BMC Genom.* 11, 649.
- Huynh, N.P.T., Anderson, B.A., Guilak, F., McAlinden, A., 2017. Emerging roles for long noncoding RNAs in skeletal biology and disease. *Connect. Tissue Res.* 58, 116–141.

- Ibrahim, F., Rohr, J., Jeong, W.-J., Hesson, J., Cerutti, H., 2006. Untemplated oligoadenylation promotes degradation of RISC-cleaved transcripts. *Science* 314, 1893.
- Jordan, I.K., Marino-Ramirez, L., Koonin, E.V., 2005. Evolutionary significance of gene expression divergence. *Gene* 345, 119–126.
- Kapusta, A., Feschotte, C., 2014. Volatile evolution of long noncoding RNA repertoires: Mechanisms and biological implications. *Trends Genet.* 30, 439–452.
- Keng, B.M., Chan, O.Y., Ling, M.H., 2014. Codon usage bias is evolutionarily conserved. *Asia Pac. J. Life Sci.* 7, 233–242.
- Kimura, T., Jiang, S., Nishizawa, M., *et al.*, 2013. Stabilization of human interferon- α 1 mRNA by its antisense RNA. *Cell. Mol. Life Sci.* 70, 1451–1467.
- Kukurba, K.R., Montgomery, S.B., 2015. RNA Sequencing and Analysis. *Cold Spring Harb. Protoc.* 2015, 951–969.
- Lavorgna, G., Dahary, D., Lehner, B., *et al.*, 2004. In search of antisense. *Trends Biochem. Sci.* 29, 88–94.
- Li, J.T., Zhang, Y., Kong, L., Liu, Q.R., Wei, L., 2008. Trans-natural antisense transcripts including noncoding RNAs in 10 species: Implications for expression regulation. *Nucleic Acids Res.* 36, 4833–4844.
- Li, Q., Su, Z., Xu, X., *et al.*, 2012. AS1DHRS4, a head-to-head natural antisense transcript, silences the DHRS4 gene cluster in cis and trans. *Proc. Natl. Acad. Sci. USA* 109, 14110–14115.
- Li, S., Liberman, L.M., Mukherjee, N., Benfey, P.N., Ohler, U., 2013. Integrated detection of natural antisense transcripts using strand-specific RNA sequencing data. *Genome Res.* 23, 1730–1739.
- Ling, K.-H., Brautigan, P.J., Moore, S., *et al.*, 2016. Derivation of an endogenous small RNA from double-stranded Sox4 sense and natural antisense transcripts in the mouse brain. *Genomics* 107, 88–99.
- Ling, M.H.T., Ban, Y., Wen, H., Wang, S.M., Ge, S.X., 2013. Conserved expression of natural antisense transcripts in mammals. *BMC Genom.* 14, 243.
- Markham, N.R., Zuker, M., 2005. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res.* 33, W577–W581.
- Merrick, B.A., Phadke, D.P., Auerbach, S.S., *et al.*, 2013. RNA-Seq profiling reveals novel hepatic gene expression pattern in aflatoxin B1 treated rats. *PLOS ONE* 8, e61768.
- Miller, J.A., Menon, V., Goldy, J., *et al.*, 2014. Improving reliability and absolute quantification of human brain microarray data by filtering and scaling probes using RNA-Seq. *BMC Genom.* 15, 154.
- Mizuno, T., Chou, M.Y., Inouye, M., 1984. A unique mechanism regulating gene expression: Translational inhibition by a complementary RNA transcript (micRNA). *Proc. Natl. Acad. Sci. USA* 81, 1966–1970.
- Nielsen, M.M., Tehler, D., Vang, S., *et al.*, 2014. Identification of expressed and conserved human noncoding RNAs. *RNA* 20, 236–251.
- Oono, Y., Yazawa, T., Kanamori, H., *et al.*, 2017. Genome-wide analysis of rice cis-natural antisense transcription under cadmium exposure using strand-specific RNA-Seq. *BMC Genom.* 18, 761.
- Osato, N., Suzuki, Y., Ikeo, K., Gojobori, T., 2007. Transcriptional interferences in cis natural antisense transcripts of humans and mice. *Genetics* 176, 1299–1306.
- Ostrowski, L.A., Saville, B.J., 2017. Natural antisense transcripts are linked to the modulation of mitochondrial function and teliospore dormancy in *Ustilago maydis*. *Mol. Microbiol.* 103, 745–763.
- Paulsen, M., 2011. Unique patterns of evolutionary conservation of imprinted genes. *Clin. Epigenet.* 2, 405–410.
- Rosikiewicz, W., Makatowska, I., 2016. Biological functions of natural antisense transcripts. *Acta Biochim. Pol.* 63, 665–673.
- Rühle, F., Stoll, M., 2016. Long non-coding RNA databases in cardiovascular research. *Genom. Proteom. Bioinform.* 14, 191–199.
- Russo, J., Harrington, A.W., Steiniger, M., 2016. Antisense transcription of retrotransposons in *Drosophila*: An origin of endogenous small interfering RNA precursors. *Genetics* 202, 107–121.
- Shao, J., Chen, H., Yang, D., *et al.*, 2017. Genome-wide identification and characterization of natural antisense transcripts by strand-specific RNA sequencing in *Ganoderma lucidum*. *Sci. Rep.* 7, 5711.
- Siatakas, N., Papaventsis, D., Levidiotou-Stefanou, S., Vamvakopoulos, N.C., Markoulatos, P., 2005. Classification and structure of echovirus 5'-UTR sequences. *Virus Genes* 31, 293–306.
- Swaminathan, A., Beilharz, T.H., 2016. Epitope-tagged yeast strains reveal promoter driven changes to 3'-end formation and convergent antisense-transcription from common 3' UTRs. *Nucleic Acids Res.* 44, 377–386.
- Upreti, B., Kaja, A., Ferdoush, J., Sen, R., Bhaumik, S.R., 2016. Regulation of antisense transcription by NuA4 histone acetyltransferase and other chromatin regulatory factors. *Mol. Cell. Biol.* 36, 992–1006.
- Villegas, V.E., Zaphiropoulos, P.G., 2015. Neighboring gene regulation by antisense long non-coding RNAs. *Int. J. Mol. Sci.* 16, 3251–3266.
- Wang, X.-J., Gaasterland, T., Chua, N.-H., 2005. Genome-wide prediction and identification of cis-natural antisense transcripts in *Arabidopsis thaliana*. *Genome Biol.* 6, R30.
- Werner, A., 2005. Natural antisense transcripts. *RNA Biol.* 2, 53–62.
- Werner, A., Berdal, A., 2005. Natural antisense transcripts: Sound or silence? *Physiol. Genom.* 23, 125–131.
- Werner, A., Sayer, J.A., 2009. Naturally occurring antisense RNA: Function and mechanisms of action. *Curr. Opin. Nephrol. Hypertension* 18, 343–349.
- Werner, A., Swan, D., 2010. What are natural antisense transcripts good for? *Biochem. Soc. Trans.* 38, 1144–1149.
- Werner, A., Carlile, M., Swan, D., 2009. What do natural antisense transcripts regulate? *RNA Biol.* 6, 43–48.
- Xu, Z., Wei, W., Gagneur, J., *et al.*, 2011. Antisense expression increases gene expression variability and locus interdependency. *Mol. Syst. Biol.* 7, 468. n/a.
- Yu, D., Meng, Y., Zuo, Z., Xue, J., Wang, H., 2016. NATpipe: An integrative pipeline for systematical discovery of natural antisense transcripts (NATs) and phase-distributed nat-siRNAs from de novo assembled transcriptomes. *Sci. Rep.* 6, 21666.
- Zarouchioli, C., Sanchez-Pintado, B., Hafford Tear, N.J., *et al.*, 2018. Antisense therapy for a common corneal dystrophy ameliorates TCF4 repeat expansion-mediated toxicity. *Am. J. Hum. Genet.* 102, 528–539.
- Zhai, W., Yao, X.-., Xu, Y.-., *et al.*, 2014. Transcriptome profiling of prostate tumor and matched normal samples by RNA-Seq. *Eur. Rev. Med. Pharmacol. Sci.* 18, 1354–1360.
- Zhang, Y., Li, J., Kong, L., *et al.*, 2007. NATsDB: Natural antisense transcripts database. *Nucleic Acids Res.* 35, D156–D161.
- Zheng, H., Qiyan, J., Zhiyong, N., Hui, Z., 2013. Prediction and identification of natural antisense transcripts and their small RNAs in soybean (*Glycine max*). *BMC Genom.* 14, 280.
- Zong, X., Nakagawa, S., Freier, S.M., *et al.*, 2016. Natural antisense RNA promotes 3' end processing and maturation of MALAT1 lncRNA. *Nucleic Acids Res.* 44, 2898–2908.
- Zubko, E., Kunova, A., Meyer, P., 2011. Sense and antisense transcripts of convergent gene pairs in *Arabidopsis thaliana* can share a common polyadenylation region. *PLOS ONE* 6, e16769.

Relevant Websites

<http://www.bioinfo.org.cn/anticode/index.htm>
antiCODE.
<http://bis.zju.edu.cn/pnatdb/>
Plant Natural Antisense Transcripts DataBase.

Statistical and Probabilistic Approaches to Predict Protein Abundance

Rubbiya A Ali and Ahmed M Mehdi, University of Queensland, Brisbane, QLD, Australia

Ali Naqi, Queensland University of Technology, Brisbane, QLD, Australia

Sarah Ali, Hamdard Institute of Engineering and Technology, Islamabad, Pakistan

Mashal Fatima, Bahauddin Zakariya University, Multan, Pakistan

Musarat Ishaq, St. Vincent's Institute of Medical Research, Melbourne, VIC, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Proteins are complex molecules with many vital roles in a cell such as signal reception, amplification and transduction. They also help to control gene expression, produce and transport molecules (Ford *et al.*, 2001; Yilmaz and Grotewold, 2010). The proper functioning of these processes is critically dependent on the protein abundance that is highly regulated in prokaryotes and eukaryotes. The proteins are involved in regulating the cellular homeostasis and other metabolic mechanisms. Therefore, to study an organism, it's crucial to understand the homeostasis and the balance between DNA, RNA and proteins. Any mutation in DNA/RNA or protein results in the disruption of the total cellular homeostasis and could lead to a disease (Leu *et al.*, 2011). Measuring the protein abundance to understand the dynamics between mRNA and protein levels has been investigated by several researchers. Here, we review experimental and predictive approaches to understand these dynamics.

Experimental Technologies for Measuring Protein Abundance

The expressions of proteins assist in regulating all the biochemical mechanism including signalling, transportation, degradation, genomic replication and translation of genomic data. The protein abundance and depletion is dealt with delicacy as it could disrupt the whole cellular biochemical dogma within cells (Steen and Pandey, 2002). Interestingly in yeast, 27% of total protein is expressed in the nucleus (Kumar *et al.*, 2002), indicating the importance of nuclear proteins abundance in replication, maintenance and repair of the genome. Furthermore, several diseases have been associated with proteins that are translocated in the nucleus. Therefore, it is crucial to correctly quantify protein to determine a disease or a type of disease for therapeutic purposes (Powers and Palecek, 2012). Several protein quantification techniques (such as immunohistochemistry, flow cytometry and ELISA) are in current use for the diagnosis of several diseases (Ferrier *et al.*, 1999; Yaziji *et al.*, 2008; Fromm *et al.*, 2009). Furthermore, techniques such as mass spectrometry (MS) (Vlahou *et al.*, 2003), proximity assays (Blokzijl *et al.*, 2010) and protein microarray (Ingvarsson *et al.*, 2008) are the emerging techniques for the protein quantification with more thorough output and more accurate quantification of multiple proteins at an instance. By so far, these protein quantification techniques have helped to recognize multiple markers for several diseases. However experimentally determining protein abundance is not as cost-effective as measuring mRNA concentration.

Researchers have measured protein abundance data using different technologies. For example, Chaemmaghami *et al.*, (2003) created a *S. cerevisiae* library by tagging each ORF with high-affinity epitope and by using immunodetection to calculate the absolute measurements of protein abundance. Newman *et al.* (2006) identified GF-tagged strains at single-cell resolution using high throughput flow cytometric experiment. Newman and colleagues measured protein abundance levels both in rich and minimal medium conditions to investigate the impact of medium in measuring protein abundance. Lee and colleagues studied the association between mRNA expression and protein abundance. The authors measured protein abundance by combining mass spectrometry with isobaric tagging (Lee *et al.*, 2011).

Determinants of Protein Abundance

Several scientists have described using mRNA concentration to predict protein abundance and activity, assuming that mRNA expression values are the main basis of protein abundances predictions (Tuller *et al.*, 2007; Zur and Tuller, 2012; Mehdi *et al.*, 2014). The mRNA expression data can be used to develop accurate models of protein abundance (Greenbaum *et al.*, 2003). Experimentally measuring protein abundance is highly complex and expensive. The mRNA expression is usually studied utilising several techniques among them is Microarray and RNAseq techniques (Zhu *et al.*, 2008). The RNA expression is intern designed by the RNA polymerases through DNA. The ribosomal proteins then translate this mRNA and translated into protein (Roeder, 1996). The new studies have shown that the RNA expression is sometimes unable to represent the actual protein expression levels within cells (Koussounadis *et al.*, 2015). Therefore, it is always complicated to identify protein expression through RNA expression. Furthermore, other factors also contribute to protein expression such as the half-life of mRNA and protein molecules, protein specificity towards cell cycle (Belle *et al.*, 2006). Other important determinants of protein abundance are relative tRNA abundance (Shah and Gilchrist, 2010), evolutionary rates (Tuller *et al.*, 2007; Zur and Tuller, 2012), folding energy and RNA binding proteins (Mehdi *et al.*, 2014).

Linear Regression Methods to Predict Protein Abundance

A number of different researchers use univariate approach to measure the correlation between predicted and actual protein abundance values (Yu *et al.*, 2007; Lahtvee *et al.*, 2017; Zur and Tuller, 2012). Tuller *et al.* (2007) developed two linear models that predict protein abundance and translational efficiency. These models are based on the mRNA expression values, relative tRNA abundance and evolutionary rate. For any transcript/mRNA/protein/ORF represented as 'x', the two predicted protein abundance models are given below;

$$\text{Tuller} - 1 : \log\text{PA}(x) = 3.97 + 0.4 * \log\text{mRNA}(x) + 10.34 * \text{tAI}(x) - 3.35 * \text{ER}(x) \quad (1)$$

$$\text{Tuller} - 2 : \log\text{PA}(x) = 3.47 + 0.63 * \log\text{mRNA}(x) + 10.89 * \text{tAI}(x) - 2.923 * \text{ER}(x) \quad (2)$$

Tuller and colleagues have shown that by comparing actual protein abundance with predicted protein abundance the above models achieve spearman rank correlation coefficients of 0.63 and 0.76 respectively.

Probabilistic Approaches to Predict Protein Abundance

Very few scientists have used non-linear approaches to predict protein abundance. For example, Tuller *et al.* (2007) used support vector machines to develop protein abundance predictor, however the accuracy of their model was lower than linear models. The non-linear regression performed by Lahtvee *et al.* (2017) explained proteins abundance with maximum of 61% of its variance.

Recently a Bayesian networks approach was shown to surpass accuracy of all available predictors of protein abundance (Mehdi *et al.*, 2014). Bayesian networks are belief networks that contain set of random variables. These variables have casual relationships with other variables that are represented as directed acyclic graphs. Mehdi *et al.* (2014) developed a probabilistic Bayesian networks models that integrate random variables from four different sources. The four data sources include mRNA values, protein abundance, mRNA sequence and mRNA/protein interaction data. Thus, their model combines different random variables measured with different technologies. The model is shown in Fig. 1 (reproduced from the work of Mehdi and colleagues) where PA(x) is a random variable that predicts the abundance of proteins (see "Relevant Websites section" to access the model). The variable 'x' represents ORF/gene/protein/molecule.

Zur and Tuller (2012) have previously shown that mRNA expressions, folding energy and tRNA adaptation index (tAI) correlates with protein abundance. In the Bayesian networks model, the authors first integrate mRNA expression variable with protein abundance variable and mRNA folding energy variable by assuming that increase in folding of a mRNA molecule will increase its expression and/or translation (Zur and Tuller, 2012). The model links RBP(x) with mRNA(x) representing the interaction of RNA binding proteins with mRNA molecule that modulates protein abundance. Here RBP(x) is a discrete valued node representing no interaction with mRNA (RBP(x)=1), maximum of two interactions (RBP(x)=2), maximum of four interactions and minimum of three interactions (RBP(x)=3) and above four interactions between mRNA and RBP (RBP(x)=4). Fig. 2(A) represents the probability of interaction of mRNA with RBP learned by the model. For example, 5.4% of mRNA molecules do not interact with any RBPs. 35.1% of mRNA interacts with maximum of two RBPs, 42.4% of mRNA interacts with maximum of 3 or 4 RBPs and 17.1% of mRNA molecules interact with at least 5 RBPs. We further note (Fig. 2(A)) that mRNAs that do not interact with RBPs possess low expression ($\mu = -0.15$) as compared to those who interact with RBPs ($\mu = -0.08, -0.04$ and -0.07).

To further represent the effect of mRNA and protein degradation on protein translation, the model links mHL(x) and pHL(x) indirectly with protein abundance node PA(x). Fig. 2(B) represent the parameters learnt by nodes mRNA and protein half-life; mHL(x) and pHL(x) respectively. The model learns parameters with separate Gaussian distributions for each random variable. The smaller half-life (node L3=False) node gets smaller standard deviation. By looking into the regression weights (β) in Table 1, we note that shorter half-life of proteins result in high mRNA folding energy and codon usage and low protein abundance (see row 2 of Table 1).

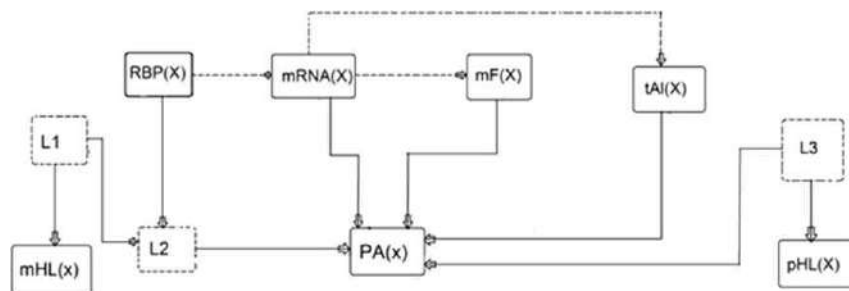


Fig. 1 Bayesian networks model predicts the protein abundance. Reproduced from Mehdi, A.M., Patrick, R., Bailey, T.L., Boden, M., 2014. Predicting the dynamics of protein abundance. *Molecular and Cellular Proteomics* 13 (5), 1330–1340. Each feature in the model is represented as a node. Latent nodes are represented with dotted lines. The casual relationships between parents and child nodes are shown with arrows.

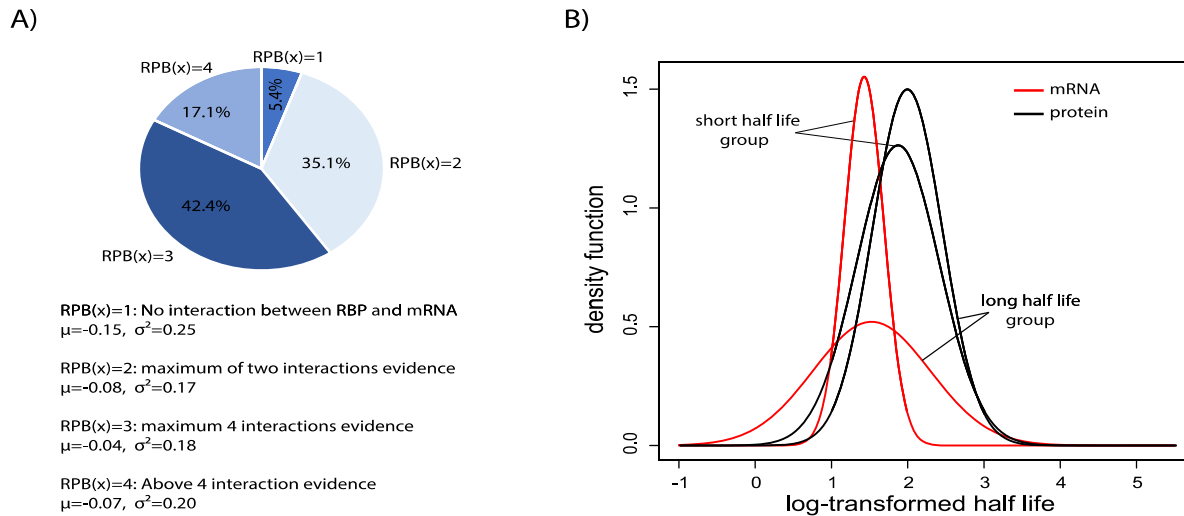


Fig. 2 Parameters learnt in Bayesian networks model. (A) Pie chart represents the probability distribution (in percentage) for RBPs to interact with mRNA. (B) Parameters of log transformed mRNA (red) and protein (black) half-life values are shown in terms of Gaussian distributions.

Table 1 Parameters learnt for PA(x) node in Bayesian networks model

Latent 2	Latent 3	μ_{PA}, σ_{PA}^2	β_{mRNA}	β_{mF}	β_{tAI}
F	F	−5.5, 0.63	−0.19	0.08	14.5
T	F	−27.5, 85.9	−1.3	0.29	62.3
F	T	−0.78, 0.04	0.12	0.02	2.6
T	T	−2.5, 0.07	0.32	0.03	7.7

Predicting Protein Abundance of Nuclear Proteins: A Case Study

This review now presents a case study by using four previously developed models to derive a correlation between RNA expression and protein abundance within the cell. Specifically, we present how to predict the abundance for the proteins typical involve in cell differentiation. Because multiple diseases including various cancers are associated with cell cycle related proteins (Otto and Sicinski, 2017), therefore, our case study will explore to what extent the nuclear proteins can be quantified in each phase of cell cycle. We also investigate how best a yeast-trained model can predict protein abundance in a human cancer cell line.

Data Resources to Develop Models

Experimental protein abundance data: In this case study, we used four experimental protein abundance datasets labelled as S1, S2, S3 and S4 (see “Relevant Websites section” to data access). The data source S1 is taken from the studies of Ghaemmaghami et al. (2003). The data sources S2 and S3 are taken from the study of Newman et al. (2006). The data source S4 is taken from the study of Lee et al. (2011).

Nuclear protein localization: Huh et al. (2003), classified proteins into 22 distinct localization in *S. cerevisiae*. 1434 of these were found to be either localized into nucleus or nucleolus. We downloaded the list of these open reading frames (ORFs) from Yeast GFP fusion localization database (see “Relevant Websites section”) to define nuclear proteins.

mRNA expression data: The mRNA expression values of two large-scale datasets from the study of Ingolia and colleagues and Wang and colleagues have been averaged as described by Mehdi and colleagues are used in the case study (Wang et al., 2002; Ingolia et al., 2009). The final mRNA expression data was log-transformed prior to be used in the prediction model.

tRNA adaptation index: The tRNA adaptation index values that has been previously constructed by Mehdi and colleagues has been used in the case study (Mehdi et al., 2014).

Half-life values of transcript and protein: The half-life values (for both transcript and protein) were taken from the study of Shalem and colleagues and Belle and colleagues are used (Belle et al., 2006; Shalem et al., 2008). These half-life values were log-transformed before applying to the available models (where required).

mRNA folding energy and interaction data: The mRNA folding energy values were constructed by using the averaged PARS (parallel analysis of RNA structure) values available through the study of Kertesz et al. (2010). The mRNA-RNA binding proteins interaction data was taken from the research of Hogan et al. (2008).

Cell cycle data and analyses: The cell cycle data has been taken from Cyclebase, referred to Pramila-alpha38 set (Gauthier *et al.*, 2008; Santos *et al.*, 2015). The data has been log2 transformed and centred at zero values by Cyclebase team. To determine the location of protein in cell-cycle phases (G1, S, G2 and M, G2 and M), we first determined the predicted protein abundance in each phase separately. We then performed one-tailed t-test between the abundance of proteins in each phase with all other phases. Specifically, the t-test was performed between G1 versus all phases, S versus all phases, G2 versus all phases and M versus all phases. The proteins with p-values < 0.05 were considered as significant and labelled as maximally expressed proteins in the corresponding phase.

Human RNAseq and Proteomics data: Melanoma cell (Mel007) were locally cultured in 6-well plates at 5×10^5 cells per well for overnight. We then treated the culture with atmospheric gas plasma (AGP) for 3 minutes as described previously (Ishaq *et al.*, 2014). An 8 h after later following the above approach, total cell RNA was extracted using Trizole reagent (Invitrogen, C#15596-026) and cells were lysed in RIPA lysis buffer (Thermoscientific, C#89901) as detailed by Ishaq *et al.* (2014). Total RNA was used for next generation RNA sequencing using Illumina HiSeqTM2000 at Beijing Genomic Institute (BGI) facility, Beijing, China. The RNAseq data relative to non-treated control was fed into mRNA(x) node of the model.

Comparing the Accuracy of Available Prediction Models

Mehdi *et al.* (2014) developed two Bayesian networks models, trained on log10 (model-1) and log2 (model-2) transformed mRNA and half-life data respectively. We ran model-1 and model-2 by providing input features (mRNA expression, mRNA-RBP interactions, mRNA and protein half-life, tRNA adaptation index and folding energy). One of the advantages of using Bayesian networks model is its capability to deal with missing values. Therefore, there were many instances where the proteins have more than one missing values while predicting their abundance, therefore the nodes corresponding to missing values were unspecified. To find the correlation between predicted and actual protein abundance of nuclear proteins, we determined the Pearson and Spearman rank correlation coefficients. These correlation coefficients have been used to study protein abundance in several previous studies (Tuller *et al.*, 2007; Zur and Tuller, 2012). In practice, the Pearson correlation assumes a linear relationship between predicted and actual protein abundance. Spearman rank correlation is the non-parametric way of finding the Pearson correlation and is more useful where the strength of direction of two variables (predicted and actual protein abundance) is required. Spearman rank correlation determines if there is a monotonic relationship between predicted and actual protein abundance.

We used both models (model-1 and model-2) to determine the abundance of nuclear proteins. To determine the accuracy of the model, we used four protein abundance sources S1-S4, as described in Data Resources section to correlate with the predicted protein abundance. The correlation values (Spearman rank, Pearson) along with p-values are shown in Table 2.

The scatter plots in Fig. 3(A-D) shows the predicted protein abundance values (using model-1) and four experimental protein abundance datasets (S1-S4) along with Pearson correlation values. The scatter plot in Fig. 3(E-H) shows the correlation between predicted protein abundance (model-2) and actual protein abundance remain similar to that of model-1. Correlation using both methods are highly significant ($p\text{-val} < 0.0001$) and comparable. Thus, we displayed Pearson correlation in the Fig. 3. Table 2 shows that highest correlation was found for protein abundance sources S2 and S3 (Pearson correlation, $r > 0.7$). Sources S2 and S3 are generated under rich and minimal medium growth conditions respectively, thus representing that nuclear protein abundances are not affected by change in the medium. Sources S1 and S4 also showed high correlations ($r = 0.62$ and 0.64 for data source S1 and 0.58 and 0.59 for data source S4) however lower than S2 and S3.

We also compared our predictions with the two models (Tuller-1 and Tuller-2) of Tuller *et al.* (2007) and the results are shown in Table 2. We found that the correlation values of protein abundance models of Tuller and colleagues are slightly higher than the

Table 2 Pearson and Spearman rank correlation values of predicted and actual protein abundance

Data source	Model	Pearson correlation	p-values	Spearman rank correlation	p-values
S1	Model-1	0.62	2.2×10^{-135}	0.62	2.2×10^{-134}
S2		0.71	5.8×10^{-124}	0.68	4.5×10^{-109}
S3		0.72	5.1×10^{-120}	0.66	3.7×10^{-95}
S4		0.59	2.5×10^{-69}	0.70	1.1×10^{-105}
S1	Model-2	0.64	6.9×10^{-146}	0.64	6.2×10^{-145}
S2		0.72	6.2×10^{-129}	0.69	2.7×10^{-113}
S3		0.72	3.3×10^{-123}	0.67	4.9×10^{-98}
S4		0.59	5.4×10^{-70}	0.69	2.3×10^{-104}
S1	Tuller-1	0.68	1.3×10^{-103}	0.68	2.7×10^{-102}
S2		0.74	2.2×10^{-83}	0.72	6.0×10^{-77}
S3		0.73	4.4×10^{-76}	0.70	1.6×10^{-65}
S4		0.64	1.7×10^{-50}	0.68	0.0×10^{00}
S1	Tuller-2	0.68	8.7×10^{-105}	0.68	1.4×10^{-102}
S2		0.75	6.6×10^{-86}	0.73	1.5×10^{-78}
S3		0.73	1.0×10^{-75}	0.69	6.8×10^{-65}
S4		0.64	1.3×10^{-50}	0.68	0.0×10^{00}

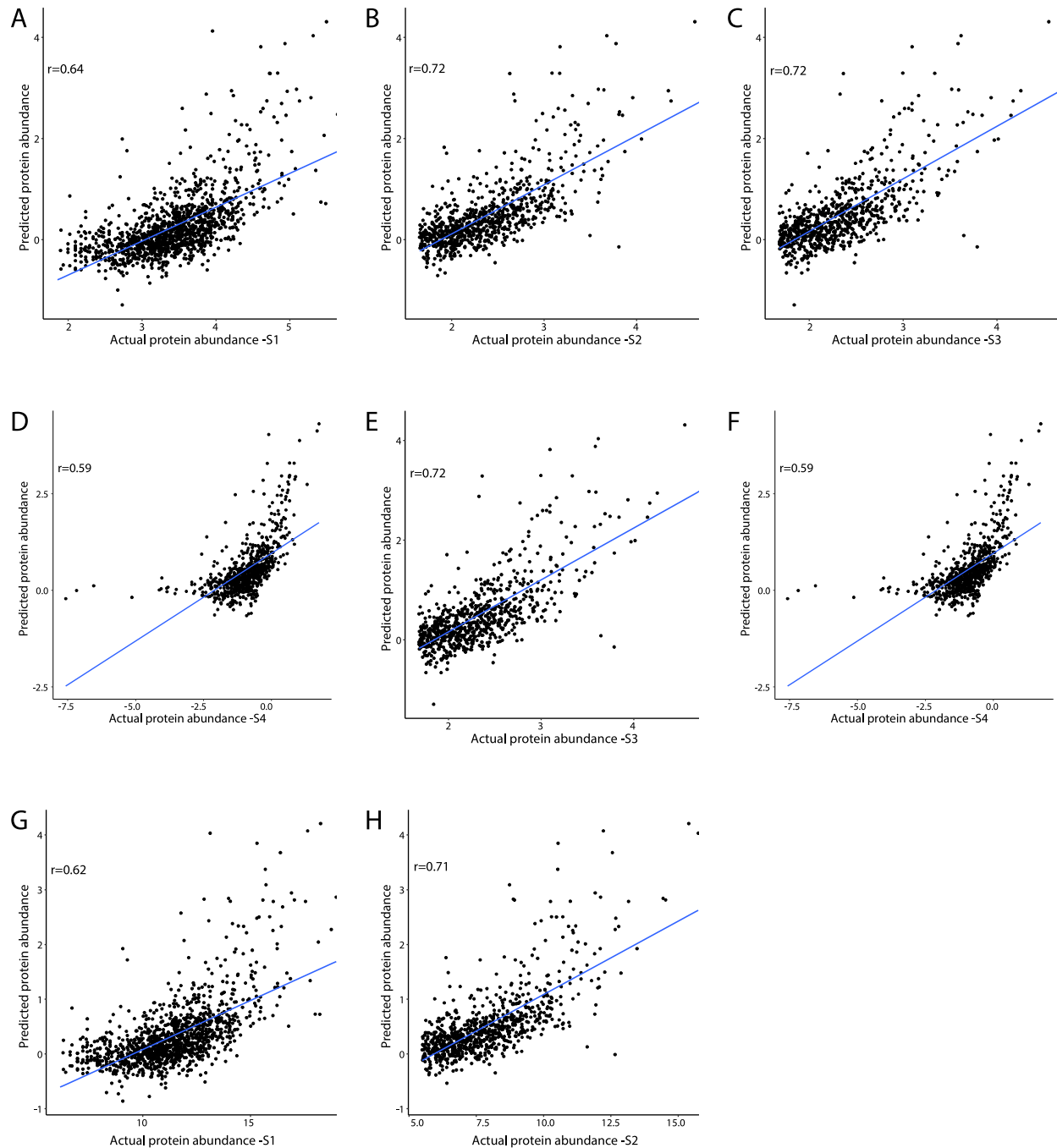


Fig. 3 Correlation of actual and predicted protein abundance using model-1 (A–D) and model-2 (E–H). Pearson correlation coefficient and linear fit is also shown.

models of Mehdi and colleagues. Highest Pearson correlation, $r=0.74$ (Tuller-1) and $r=0.75$ (Tuller-2) were found by correlating their protein abundance with S2 data source. Because the models of Tuller and colleagues are not capable to run on data with missing values, their model operated only on 703 (out of 1434) ORFs to determine the protein abundance thus we believe that the models of Mehdi *et al.*, are quite useful to predict protein abundance in practice and accuracy of prediction is not significantly compromised.

Predicting Abundance of Nuclear Proteins in Cell-Cycle Phases

The models of Mehdi and colleagues are capable of operating under condition variant stages such as cell cycle process. Perturbations in cell cycle can result in cellular death leading to various diseases including cancer. It has been

previously shown that the predicted protein abundance during cell cycle phases better explains the processes involved in cell cycle as compared to mRNA values alone (Mehdi *et al.*, 2014). Therefore, predicting cell-cycle protein abundance may help to precisely characterize the role of proteins, uncovering regulation pathways. Here we evaluated the Bayesian networks model to predict protein abundance during cell-cycle process by investigating PA(x) node and providing cell-cycle mRNA expression data generated by Permella and colleagues available at Cyclebase (Gauthier *et al.*, 2008). We were specifically interested to quantify the number of proteins in each cell cycle phase. To perform quantification, we performed one-tailed t-tests of predicted protein abundance data in G1 versus all phases, G2 versus all phases, S versus all phases and M versus all phases to count number of proteins that are maximally abundant in each phase. As shown in Fig. 4, we found that 41.8% of the proteins were not found to be significantly abundant in any of the phases and their location inside cell-cycle phases could not be uncovered. Additionally, we found 16.6% of the proteins are associated with G1 phase, 15.4% S-phase, 15.5% G2 phase and 10.6% proteins were quantified into M phase. When we specifically looked the quantification of nuclear proteins in each phase, 13.4% of those were found to be highly expressed in G1 phase, 17.9% in S phase, 14.7% in G2 phase and 9.5% in M phase. Similar to previous analysis, we were not able to quantify 44.5% of the nuclear proteins in any of the phases of cell cycle.

Predicting mRNA Folding Energy Using Protein Abundance

One of the advantages of using probabilistic Bayesian networks is their ability to designate any node as input and any node as an output node. Current methods that predict folding energy of transcripts are not based on experimental data (except the Bayesian networks model of Mehdi and colleagues), therefore could lead in making inaccurate conclusions regarding mRNA folding. It has been previously shown that the prediction accuracy of mRNA folding energy of available methods is worse than random (Zur and Tuller, 2012; Mehdi *et al.*, 2014). We therefore inferred mF(x) node of Bayesian networks model to predict folding energy of nuclear proteins of *S. cerevisiae* data. For model-1, the Pearson correlation values were found; $r=0.62$, $r=0.57$, $r=0.58$ and $r=0.55$ when providing protein abundance data from data sources S1, S2, S3 and S4 respectively. We also provided data values to all other nodes of model where available. Similarly, for model-2, the respective Pearson correlation values were; $r=0.67$ for S1, S2 and S3 and $r=0.65$ for S4. Data values to all other nodes were also provided where available.

Predicting Protein Abundance in Human Melanoma Cell Line

Atmospheric Gas plasma (AGP), a source of nitric oxide (NO), reactive oxygen/nitrogen species (RONS) has been under investigation for its properties to specifically kill variety of tumor cells. Melanoma (Mel007) is the drug-resistant tumor type which lack genome sequence data study. The Bayesian networks model developed by Mehdi and colleagues has been trained on yeast data set. We aimed to test the yeast trained model to predict abundance of proteins in human melanoma cell line (Mel007). Specifically, we investigated the anticancer effects of AGP on Mel007 cells in in-vitro cell models. Using next generation sequencing (NGS) technology and Proteomics, we first experimentally detected expression of genes. We then provided expression values of genes to the model and inferred PA(x) node to predict protein abundance, all other nodes (mHL(x), pHL(x), tAl(x) and RBP(x)) were unspecified. We then correlated the predicted protein abundance with the expression actual expression values. The positive Pearson correlation was found to be 0.11 ($p\text{-value} < 0.01$). The results show that the yeast trained model is able to measure approximately 11% of the human protein abundance correctly. To predict abundance of human proteins, a new model that is trained on human data needs to be developed.

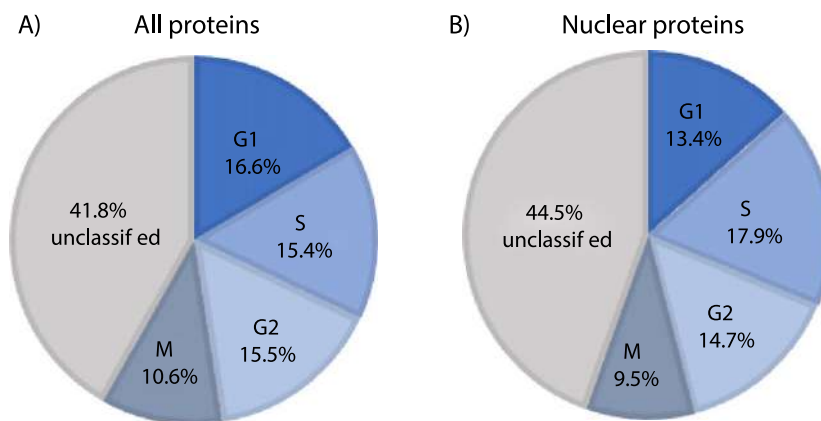


Fig. 4 Quantification of proteins in all phase of cell cycle. (A) Percentage of proteins in each phase is shown for all data. (B) Quantification for nuclear proteins in each cell-cycle shown.

Conclusions

While a number of different features have been shown to be associated with protein abundance, the available methods use very few of them. The models of Tuller and colleagues use mRNA expression, relative tRNA abundance and evolutionary rates. The models of Mehdi and colleagues use mRNA expression, relative tRNA abundance, mRNA folding energy, RBP and mRNA interactions, half-life values of transcripts and proteins. The reason for using fewer variables amongst the all determinants of protein abundance could be because of (i) availability of an insufficient data to use them in developing models, (ii) the association between these variables and protein abundance further require experimental validations.

In this review, we presented a case study by using probabilistic Bayesian networks models developed by Mehdi and colleagues and linear regression models developed by Tuller and colleagues. The models of Mehdi and colleagues link transcriptional information with protein translation and predicts protein abundance. Their models help to understand how the transcript expressions are linked with its folding and interaction with RBPs, codon usage, half-life of ORFs and protein abundance. We have particularly shown how best their models predict abundance of nuclear proteins in *S. cerevisiae* and human melanoma cell line (Mel007).

We note that the models of Tuller and colleagues use additional information of evolutionary rate. Additionally, the inability to operate on missing value data makes them less important. The quantification of proteins in each phase of cell cycle can be performed using predicted protein abundance. We have also reviewed how the available prediction models can be used to predict mRNA folding energy. For example, when correlating predicted and experimental mRNA folding energy by using models of Mehdi and colleagues, we found a maximum Pearson correlation of only 0.67, challenging further research in this area.

See also: Analyzing Transcriptome-Phenotype Correlations. Characterization of Transcriptional Activities. Comparative Transcriptomics Analysis. Genome-Wide Scanning of Gene Expression. Introduction to Biostatistics. Natural Language Processing Approaches in Bioinformatics. Regulation of Gene Expression. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation

References

- Belle, A., Tanay, A., Bitincka, L., Shamir, R., O'Shea, E.K., 2006. Quantification of protein half-lives in the budding yeast proteome. *Proc. Natl. Acad. Sci. USA* 103 (35), 13004–13009.
- Blotkijl, A., Friedman, M., Ponten, F., Landegren, U., 2010. Profiling protein expression and interactions: Proximity ligation as a tool for personalized medicine. *J. Intern. Med.* 268 (3), 232–245.
- Ferrier, C.M., de Witte, H.H., Straatman, H., *et al.*, 1999. Comparison of immunohistochemistry with immunoassay (ELISA) for the detection of components of the plasminogen activation system in human tumour tissue. *Br. J. Cancer* 79 (9–10), 1534–1541.
- Ford, K.G., Souberbielle, B.E., Darling, D., Farzaneh, F., 2001. Protein transduction: An alternative to genetic intervention? *Gene Ther.* 8 (1), 1–4.
- Fromm, J.R., Thomas, A., Wood, B.L., 2009. Flow cytometry can diagnose classical hodgkin lymphoma in lymph nodes with high sensitivity and specificity. *Am. J. Clin. Pathol.* 131 (3), 322–332.
- Gauthier, N.P., Larsen, M.E., Wernersson, R., *et al.*, 2008. Cyclebase.org – A comprehensive multi-organism online database of cell-cycle experiments. *Nucleic Acids Res.* 36 (Database issue), D854–D859.
- Ghaemmaghami, S., Huh, W.K., Bower, K., *et al.*, 2003. Global analysis of protein expression in yeast. *Nature* 425 (6959), 737–741.
- Greenbaum, D., Colangelo, C., Williams, K., Gerstein, M., 2003. Comparing protein abundance and mRNA expression levels on a genomic scale. *Genome Biol.* 4 (9), 117.
- Hogan, D.J., Riordan, D.P., Gerber, A.P., Herschlag, D., Brown, P.O., 2008. Diverse RNA-binding proteins interact with functionally related sets of RNAs, suggesting an extensive regulatory system. *PLOS Biol.* 6 (10), e255.
- Huh, W.K., Falvo, J.V., Gerke, L.C., *et al.*, 2003. Global analysis of protein localization in budding yeast. *Nature* 425 (6959), 686–691.
- Ingolia, N.T., Ghaemmaghami, S., Newman, J.R., Weissman, J.S., 2009. Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *Science* 324 (5924), 218–223.
- Ingvarsson, J., Wingren, C., Carlsson, A., *et al.*, 2008. Detection of pancreatic cancer using antibody microarray-based serum protein profiling. *Proteomics* 8 (11), 2211–2219.
- Ishaq, M., Evans, M.D., Ostrikov, K.K., 2014. Atmospheric pressure gas plasma-induced colorectal cancer cell death is mediated by Nox2-ASK1 apoptosis pathways and oxidative stress is mitigated by Srx-Nrf2 anti-oxidant system. *Biochim. Biophys. Acta* 1843 (12), 2827–2837.
- Ishaq, M., Kumar, S., Varinli, H., 2014. Atmospheric gas plasma-induced ROS production activates TNF-ASK1 pathway for the induction of melanoma cancer cell apoptosis. *Mol. Biol. Cell* 25 (9), 1523–1531.
- Kertesz, M., Wan, Y., Mazor, E., *et al.*, 2010. Genome-wide measurement of RNA secondary structure in yeast. *Nature* 467 (7311), 103–107.
- Koussounadis, A., Langdon, S.P., Um, I.H., Harrison, D.J., Smith, V.A., 2015. Relationship between differentially expressed mRNA and mRNA-protein correlations in a xenograft model system. *Sci. Rep.* 5, 10775.
- Kumar, A., Agarwal, S., Heyman, J.A., *et al.*, 2002. Subcellular localization of the yeast proteome. *Genes Dev.* 16 (6), 707–719.
- Lahtvee, P.J., Sanchez, B.J., Smialowska, A., *et al.*, 2017. Absolute quantification of protein and mRNA abundances demonstrate variability in gene-specific translation efficiency in yeast. *Cell Syst.* 4 (5), 495–504. e495.
- Lee, M.V., Topper, S.E., Hubler, S.L., *et al.*, 2011. A dynamic model of proteome changes reveals new roles for transcript alteration in yeast. *Mol. Syst. Biol.* 7, 514.
- Leu, K., Obermayer, B., Rajamani, S., Gerland, U., Chen, I.A., 2011. The prebiotic evolutionary advantage of transferring genetic information from RNA to DNA. *Nucleic Acids Res.* 39 (18), 8135–8147.
- Mehdi, A.M., Patrick, R., Bailey, T.L., Boden, M., 2014. Predicting the dynamics of protein abundance. *Mol. Cell Proteom.* 13 (5), 1330–1340.
- Newman, J.R., Ghaemmaghami, S., Ihmels, J., *et al.*, 2006. Single-cell proteomic analysis of *S. cerevisiae* reveals the architecture of biological noise. *Nature* 441 (7095), 840–846.
- Otto, T., Sicinski, P., 2017. Cell cycle proteins as promising targets in cancer therapy. *Nat. Rev. Cancer* 17 (2), 93–115.
- Powers, A.D., Palecek, S.P., 2012. Protein analytical assays for diagnosing, monitoring, and choosing treatment for cancer patients. *J. Healthc. Eng.* 3 (4), 503–534.
- Roeder, R.G., 1996. Nuclear RNA polymerases: Role of general initiation factors and cofactors in eukaryotic transcription. *Methods Enzymol.* 273, 165–171.

- Santos, A., Wernersson, R., Jensen, L.J., 2015. Cyclebase 3.0: A multi-organism database on cell-cycle regulation and phenotypes. *Nucleic Acids Res.* 43 (Database issue), D1140–D1144.
- Shah, P., Gilchrist, M.A., 2010. Effect of correlated tRNA abundances on translation errors and evolution of codon usage bias. *PLOS Genet.* 6 (9), e1001128.
- Shalem, O., Dahan, O., Levo, M., *et al.*, 2008. Transient transcriptional responses to stress are generated by opposing effects of mRNA production and degradation. *Mol. Syst. Biol.* 4, 223.
- Steen, H., Pandey, A., 2002. Proteomics goes quantitative: Measuring protein abundance. *Trends Biotechnol.* 20 (9), 361–364.
- Tuller, T., Kupiec, M., Rupp, E., 2007. Determinants of protein abundance and translation efficiency in *S. cerevisiae*. *PLOS Comput. Biol.* 3 (12), e248.
- Vlahou, A., Laronga, C., Wilson, L., *et al.*, 2003. A novel approach toward development of a rapid blood test for breast cancer. *Clin. Breast Cancer* 4 (3), 203–209.
- Wang, Y., Liu, C.L., Storey, J.D., *et al.*, 2002. Precision and functional specificity in mRNA decay. *Proc. Natl. Acad. Sci. USA* 99 (9), 5860–5865.
- Yaziji, H., Taylor, C.R., Goldstein, N.S., *et al.*, 2008. Consensus recommendations on estrogen receptor testing in breast cancer by immunohistochemistry. *Appl. Immunohistochem. Mol. Morphol.* 16 (6), 513–520.
- Yilmaz, A., Grotewold, E., 2010. Components and mechanisms of regulation of gene expression. *Methods Mol. Biol.* 674, 23–32.
- Yu, E.Z., Burba, A.E., Gerstein, M., 2007. PARE: A tool for comparing protein abundance and mRNA expression data. *BMC Bioinform.* 8, 309.
- Zhu, J., Zhang, B., Smith, E.N., *et al.*, 2008. Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nat. Genet.* 40 (7), 854–861.
- Zur, H., Tuller, T., 2012. Strong association between mRNA folding strength and protein abundance in *S. cerevisiae*. *EMBO Rep.* 13 (3), 272–277.

Relevant Websites

<https://cloudstor.aarnet.edu.au/plus/s/2R07zzlguoR8lrn>

Data Resources and Bayesian Network models used in case study.

<https://yeastgfp.yeastgenome.org>

Yeast GFP Fusion Localization Database.

Identification of Proteins From Proteomic Analysis

Zainab Noor, Abidali Mohamedali, and Shoba Ranganathan, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Introduction

The identification of proteins in a biological system is one of the cornerstones of the study of biological systems. Proteins are biopolymers (polypeptides) transcribed from DNA, made up of a set of 20 amino acids and are the functional elements in a biological system. The combinations of these amino acids in a sequence gives rise to each proteins' unique shape, structure and hence function. In addition, in higher pro- and eu-karyotes, these polypeptides can be modified post-translationally (PTMs) to form even more complex structures and functions (Ngounou Wetie *et al.*, 2014). Indeed, most PTM's have given rise to their own fields of study such as glycomics (sugar polymers on polypeptides), phosphoproteomics (phosphate molecules attached to amino acids), among others. Proteins can be identified in one of two ways: i) through their structure and shape – usually performed by tagged antibodies (Kingsmore, 2006) or ii) by sequencing the protein- usually done by mass spectrometry (MS). Often, proteins identified by MS are validated using an antibody approach to ensure the accuracy of identification especially in critical applications such as biomarker discovery (Lindskog, 2015; Marx, 2013). After the discovery of human genome profile, both, antibody and MS-based techniques have been utilized to create the map of the human proteome, e.g. Human Protein Atlas (Uhlen *et al.*, 2015) and Human Proteome Map (Kim *et al.*, 2014), respectively. This article will focus on protein identification by mass spectrometry.

Mass Spectrometry Strategies

There are two different strategies for the identification of proteins using MS, 'Bottom-up' proteomics or 'Top-down' proteomics. In the bottom-up method, proteins are digested into smaller sized peptides prior to ionization and mass analysis and consequently reassembled *in silico* to infer identity. This strategy is highly advantageous in case of complex samples of abundant proteins that require efficient and sensitive separation. In the top-down method, proteins are subjected to ionization and fragmentation in the intact form without digestion and passed through mass analysers often in tandem. This method determines the complete characterization of protein isoforms and post-translational modifications for small to medium-sized proteins. Both of these strategies can be employed for protein identification depending upon the size and nature of target protein.

Mass Spectrometry-Based Protein Identification

Protein characterization and identification by MS can be performed using various approaches (see Fig. 1). More commonly, data is acquired in 'Data Dependent Acquisition (DDA)' mode, which is also known as shotgun or discovery proteomics, to identify which proteins/peptides are present in the sample. In this approach, after passing through the first mass analyzer (MS1), only the fixed number of most abundant ions are selected and get fragmented into product ions and analysed in the second analyzer (MS2) (see Fig. 2). A further two approaches, mainly used for quantification and identification of less abundant proteins, are 'Targeted Proteomics (MRM- Multiple Reaction Monitoring or SRM- Selected Reaction Monitoring and PRM- Parallel Reaction Monitoring)' and 'Data Independent Acquisition (DIA)' to perform targeted examination where proteins of interest, identified in discovery mode, are analysed for their abundance (relative or absolute) in a particular sample (see Fig. 2). The basic working mechanism of these techniques is illustrated in Fig. 2.

The identification of proteins in the last two decades has taken a significant leap and use of MS is exponentially increasing the understanding of things as diverse as truffles (Islam *et al.*, 2013) to the discovery of biomarkers in disease (Domon and Aebersold, 2006). This article focuses on the brief understanding of MS data generation, and detailed overview of MS data analysis procedures and, the tools and methods that can be applied at each step.

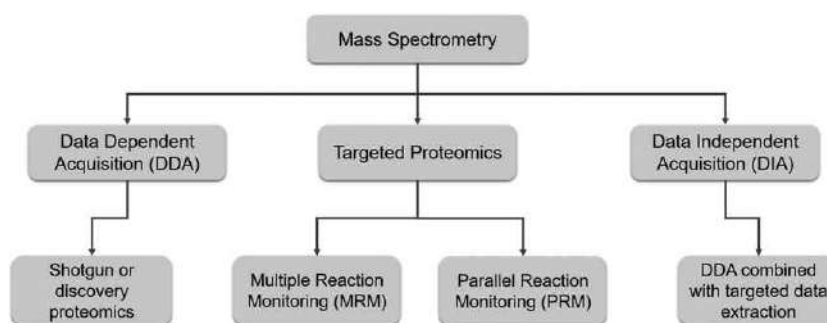


Fig. 1 Commonly used approaches in mass spectrometry (MS).

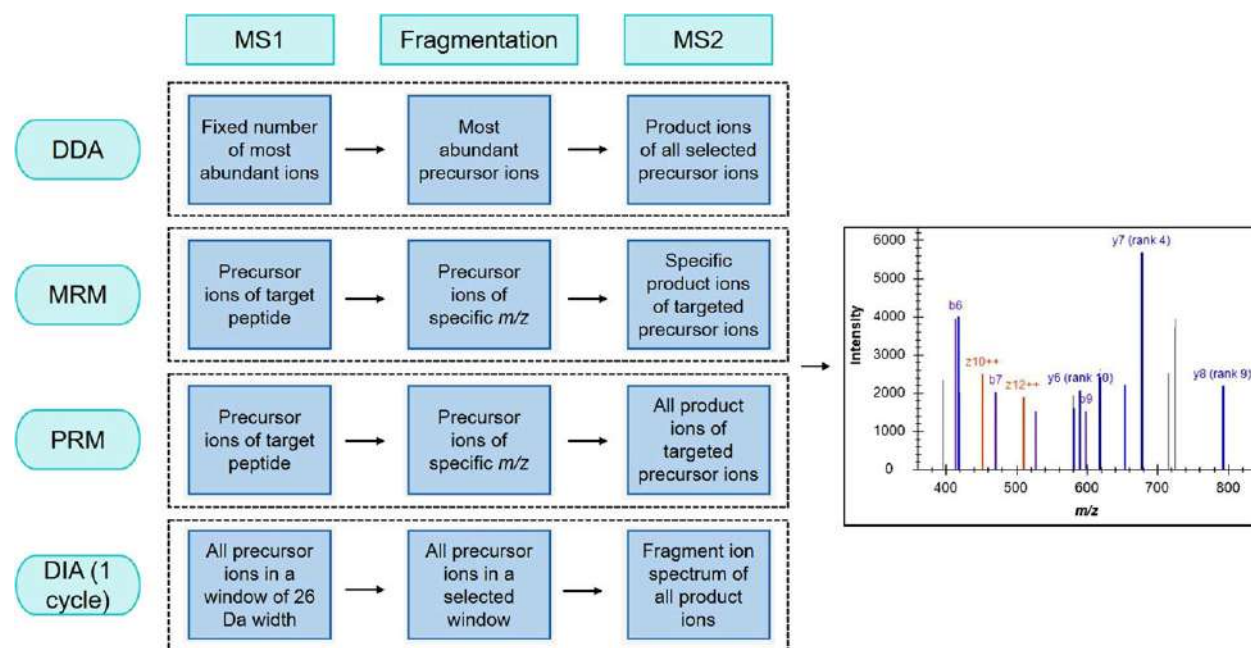


Fig. 2 Working mechanisms of MS-based approaches for protein identification and quantification.

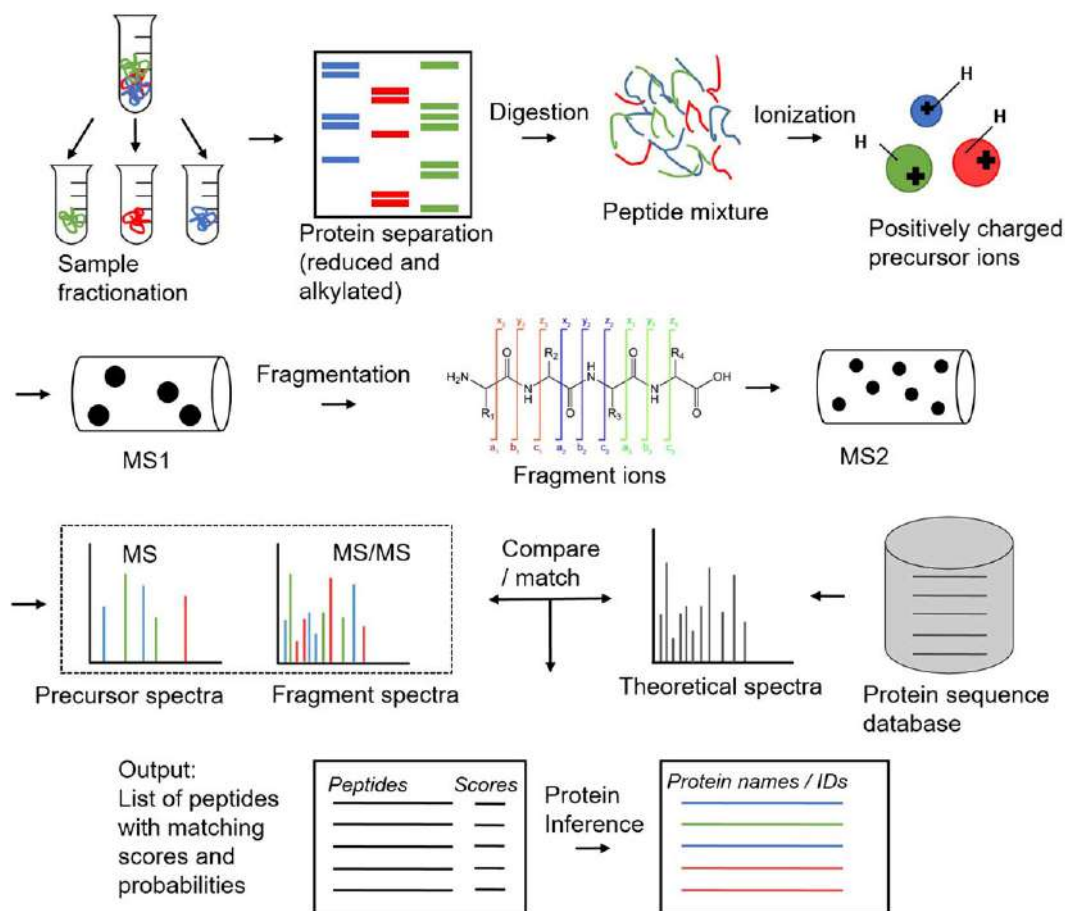


Fig. 3 Workflow of MS-based protein identification.

Sample Preparation

The MS procedure starts with the sample collection and purification of proteins. Complex mixtures are often fractionated or separated using some form of chromatography (Gel electrophoresis, ion exchange etc.). The cysteine residues are then alkylated and reduced (resulting in the oxidation of methionine residues as well as carbamylation of lysine residues) (Herbert *et al.*, 2003), which results in the formation linear polypeptides.

The complex mixture of polypeptides undergoes the process of digestion more often using trypsin that cleaves the proteins at lysine or arginine residues (Huynh *et al.*, 2009) (see Fig. 3). Digestion using trypsin alone may have certain limitations and has been found to be less efficient for membrane proteins and post-translational modification (PTM) sites. To overcome this problem, proteases, such as chymotrypsin, LysC, LysN, AspN, GluC and ArgC (Giansanti *et al.*, 2016), have been analysed in parallel with trypsin and are shown to be effective in the MS analysis of large-size proteins (Low *et al.*, 2013; Benevento *et al.*, 2014; Cauci *et al.*, 2009). Similarly, *in silico* proteolytic digestion is also performed using computational tools with the same enzyme as used in the sample preparation to generate the theoretical spectral profile of the protein. Software programs such as PeptideMass (Wilkins *et al.*, 1999), PeptideCutter (Wilkins *et al.*, 1999) and Proteogest (Cagney *et al.*, 2003) perform *in silico* cleavage of polypeptides using a range of enzymes or chemical reagents to produce search or match databases.

Subsequently, the digested peptide mixture undergoes separation using chromatography and enters into the ionization source of the mass spectrometer (MS) where they are charged. The charged peptide ions traverse into the mass analyzer for separation, according to their mass to charge (m/z) ratios, in a vacuum environment, under the influence of electric/magnetic field (see Fig. 3). Not all digested peptides are able to 'fly', if they are too big, for instance.

Fragmentation

In a single-analyzer system (MS1), the separated ions generate the spectra of precursor ions when they hit a detector. In multi-analyzer systems (tandem or MS/MS), precursor ions undergo fragmentation (usually using some form of inert gas (collision induced ionization, CID)) and are converted into smaller charged fragments or product ions. The precursor ions fragment in various ways (see Fig. 3) to form unique daughter ions, dependent on the fragmentation method, e.g. b-y ions by Collision Induced Dissociation (CID) and c-z ions by Electron Capture Dissociation (ECD) and Electron Transfer Dissociation (ETD). The daughter or product ions are then analysed again in the mass analyzer to generate spectra (MS2) (see Fig. 3). For sequencing and

ISB/SPC Trans Proteomic Pipeline :: Home

Home Files TPP Tools External Tools Account Jobs Pipeline: **Tandem**

Comet
Mascot
Sequest
SpectraST
Tandem
Kojak

Welcome

Welcome to the Trans-Proteomic Pipeline (TPP) web interface. These tools and interfaces were developed and are being... Please visit www.proteomecenter.org and tools.proteomecenter.org for more information.

ANALYSIS PIPELINE

Follow these steps to convert, search, and analyze your **Tandem** data:

- 1. RAW to mzML Conversion**
Convert original .RAW files to the standard mzML input format used by the tools
- 2. Edit Tandem Default Parameters, if necessary**
It is recommended that you copy the default input file to your local directory, and edit it as necessary
- 3. Peptide Database Search and Identification**
Perform X-Tandem search.
- 4. Conversion to pepXML**
Convert original search results to the pepXML input format used by xinteract
- 5. Data Curation and (optional) Peptide Validation and Quantification**
Use Xinteract and PepXMLViewer to filter, sort, group, and highlight data based on various criteria.
You can also validate peptide identifications using PeptideProphet, and further refine peptide-level analysis and combine results from iProphet.
Calculate the relative abundances of peptides and proteins, and the corresponding confidence intervals, from isotopically-labeled XPRESS. Or use Libra to do this for isobarically-labeled data (e.g. iTRAQ).
- 6. Protein Assignment and Validation**
ProteinProphet provides a statistical model for validation of peptide identifications at the protein level.

Fig. 4 Trans-proteomic pipeline software interface for proteomic identification and analysis.

identification, the precursor ions can be subjected to peptide mass finger printing (PMF) (Thiede *et al.*, 2005) or if MS2 data is also used (which is more accurate), the sequence of the precursor ions can be determined and searched against a database (described below) (see Fig. 3). It is also called as fragment ion spectra, produced in the raw form (details of this process can be found in a review article (Aebersold and Mann, 2003)).

Data Analysis

MS data acquisition and data analysis are performed utilising several computational tools, designed explicitly for MS-based protein identification. In this article, a sample data has been acquired through public repository to demonstrate different phases of protein identification using Trans-Proteomic pipeline (TPP) by Institute of Systems Biology (ISB) (Deutsch *et al.*, 2015). This will provide an overview of data analysis, required parameters at every stage and output of results. Additionally, The OpenMS Proteomics Pipeline (TOPP) by OpenMS (Kohlbacher *et al.*, 2007) and Central Proteomics Facilities Pipeline (CPFP) (Trudgian and Mirzaei, 2012) also maintains a compilation of bioinformatics analysis tools for MS-based proteomics. With the same parameters and guidelines, data can be analysed through standalone versions of all identification programs as well.

Sample Data Set

The raw sample data was obtained from 'fission yeast' (*Schizosaccharomyces pombe*) proteome study where ~3500 proteins (71%) of the predicted genes of yeast have been analysed using MS (Gunaratne *et al.*, 2013). The experiment was performed on nano-HPLC instrument coupled to the high-resolution Orbitrap mass spectrometer and is considered to be the largest protein dataset presented for fission yeast (Gunaratne *et al.*, 2013). The dataset is available at Peptide Atlas with the accession ID 'PAe003810'.

ISB/SPC Trans Proteomic Pipeline :: mzML/mzXML

Home Files **TPP Tools** External Tools Account Jobs Pipeline: Tandem

INPUT FILE FORMAT ☒ show

Select instrument file type you want to convert: Thermo RAW

Please note that, in certain instances, this converter will only work on machines that contain the appropriate vendor libraries.

CHOOSE FILE(S) TO CONVERT TO MZML ☒ show

c:/TPP/data/demo/tandem/yeast.RAW

Add Files Or choose from list: c:/TPP/data/demo/tandem/ [raw]

CONVERSION OPTIONS ☒ show

☒ Centroid all scans (MS1 and MS2) -- meaningful only if data was acquired in profile mode

☐ Compress peak lists for smaller output file

☐ Write the output as a gzipped file (other TPP tools can read gzipped files directly)

Enter additional options to pass directly to the command-line (expert use only!)

☐ Convert to mzXML instead of mzML

CONVERT!

Convert to mzML

Fig. 5 Trans-proteomic pipeline data conversion module (Msconvert) and its parameters.

Computation Tools

TPP has multiple pipelines for, (i) identification through protein databases, such as Comet (Eng *et al.*, 2013), Sequest (Eng *et al.*, 2008), Tandem (Craig and Beavis, 2003) pipelines, and (ii) by means of spectral libraries, such as SpectraST (Lam *et al.*, 2007) pipeline (see Fig. 4). Searching through spectral libraries can also be performed through Biblisspec (Frewen and Maccoss, 2007) for comparison purposes. The sample input data (raw files), parameter files and output files can be accessed through PeptideAtlas (see Relevant Websites section) and TPP web interface can be accessed through the provided link (see Relevant Websites section).

Step 1: Data Conversion

The data which we acquire from mass spectrometry machines is in the raw form. Generally, the formats include *.wiff, *.dat, *.yep/ *.baf, and *.raw from AB/Agilent, Agilent, Bruker and Thermo/Waters vendors, respectively. These data files are converted into standard data openXML formats such as mzXML, mzDATA and mzML. This can be performed through software, Mascot Distiller by ProteoWizard (Koenig *et al.*, 2008), Msconvert by ProteoWizard (Kessner *et al.*, 2008) and ReADW by Xcalibur (see Relevant Websites section). Conversion through any of these programs sets up the parameters for centroiding the MS scans, which are acquired in profile mode, also known as peak picking. Centroiding the data reduces the data points and lowers the sizes of files. In this example, data was available in raw format (*yeast.raw*) and was simply converted into mzML (*yeast.mzML*) format through Msconvert, under Tandem module of TPP, with centroiding (see Fig. 5).

A.

UniProtKB results

Filter by: Reviewed (5,145) Swiss-Prot

Popular organisms: SCHPO (5,141)

Other organisms: [input field] Go

Download menu:

- Download selected (0)
- Download all (5145)
- Format: FASTA (canonical)
- Compressed (selected) / Uncompressed
- Preview first 10

Entry	Gene names	Organism	Length
P04551	cdk1, cdk2, p1002, BC11B10.09	Schizosaccharomyces pombe (strain 972 / ATCC 24843) (Fission yeast)	297
Q09892	hog1, hog1, spc1, SPAC24B11.06c	Schizosaccharomyces pombe (strain 972 / ATCC 24843) (Fission yeast)	349
P50528	Serine/threonine-protein kinase plo1	Schizosaccharomyces pombe (strain 972 / SPAC23C11.16)	683

B.

PomBase

The scientific resource for fission yeast

Home Find Tools Submit Downloads

Downloads

- Downloads
 - Datasets
 - Genome Datasets
 - CDS Coordinates
 - Protein Datasets
 - GO Associations
 - Intron Data
 - Data Mappings
 - UTRs
 - Documents
 - Chado Database Dumps

Datasets

Links in the table go to web pages, except where "ftp" indicates that they go directly to ftp site directories. Note that the ftp subdirectories are on a separate page. If you have trouble finding anything, please ask the helpdesk.

Dataset	Description
ftp site	S. pombe ftp site root directory
DNA sequence and feature datasets	GFF files of sequence feature coordinates; files with genome sequence (FASTA), features (GFF), or both (EMBL)
CDS coordinates	Chromosomal coordinates of protein coding sequences - CDS only, or exons
Protein datasets	Protein sequence FASTA database, peptide features, properties, etc.
GO annotations	Gene ontology annotation files
Macromolecular complexes	Subunits of protein and ribonucleoprotein complexes (GO cellular component terms and annotated genes; ftp)

Fig. 6 Database FASTA file generation from (A) UniProt and (B) PomBase.

Step 2: Spectrum Identification

Spectrum identifications can be performed using one of three approaches, namely a database search, a spectral library search or a de-novo or hybrid search, each requiring their own properties and parameters. Identification through sequence databases entails searching against a broad list of potential proteins, which can be created or gathered from UniProt/SwissProt databases. Whereas a spectral library contains precursor mass/charge values, often obtained in one lab but is a prerequisite for identification through spectral matching.

Database search

For the fission yeast sample data, a database search is executed by Tandem, Sequest (the most common and freely available search engines) or Comet pipelines and respective search modules in TPP. The input files for the programs are:

- Mass spectrometry data files in mzML format, *yeast.mzML* (created in the previous step).
- Tandem parameters file, *tandem.params* (available online and may be modified).
- Sequence database file, *yeast.fasta* (including decoys).

Tandem parameter file (*tandem.params*) contains information related to:

- Search scoring scheme (*k-score*).
- File locations (path to the input/output files).

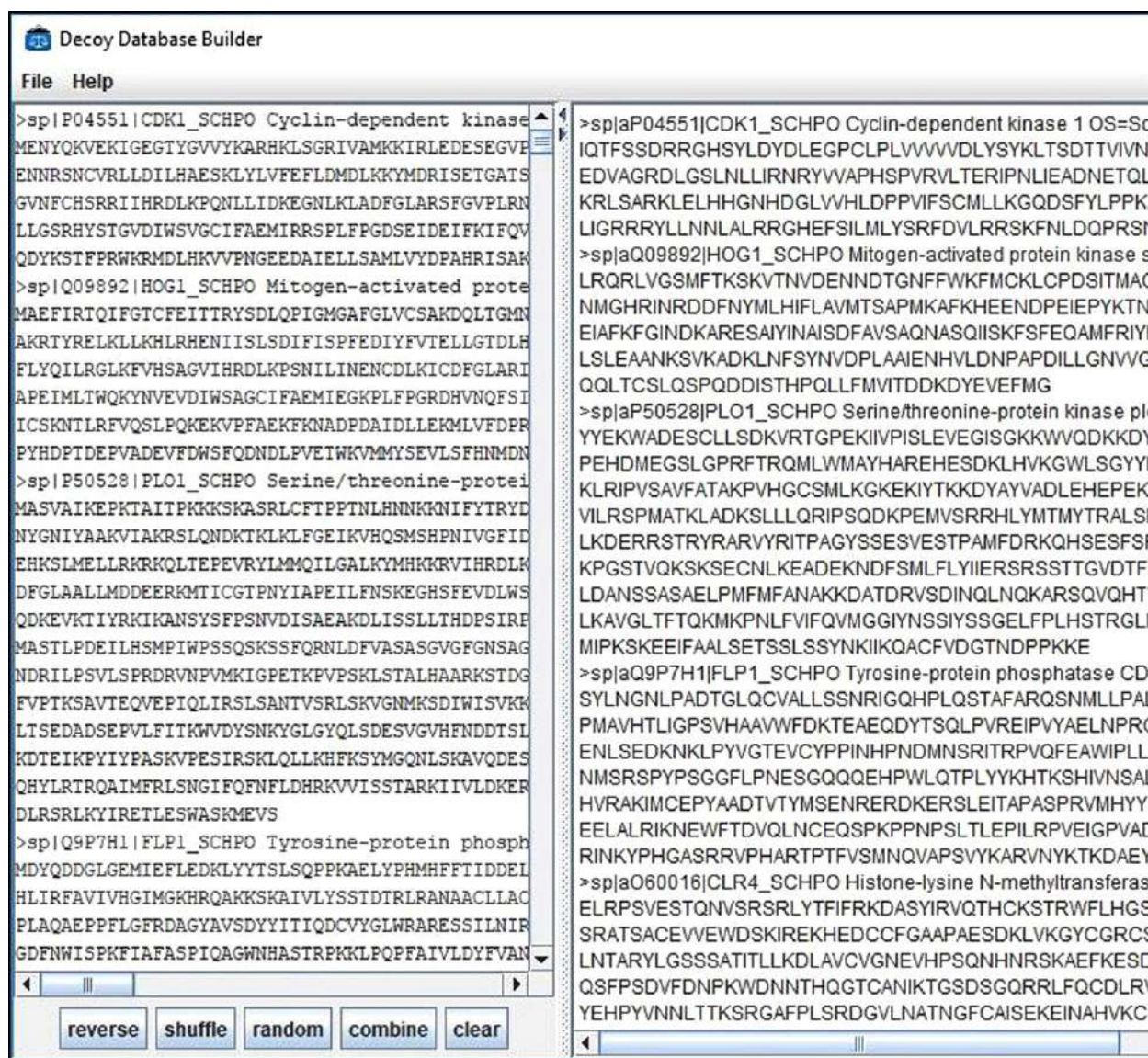


Fig. 7 Decoy sequence generation using "Decoy Database Builder."

3. Sequence database or taxonomy (name of the database file).
4. Precursor mass tolerance dependent on instrument used (default = - 2.0 to 4.0 Da).
5. Structural and isotopic modifications (according to sample run on the MS e.g. carbamidomethyl in Cysteine and/or oxidation in Methionine).
6. Peptide digestion through trypsin (fully-tryptic or semi-tryptic peptides).
7. Missed cleavages (default = 2).

Parameter files for other database search software (*sequest.params* and *comet.params*) contain similar information with very few modifications and are available with the TPP, TOPP or CPFP identification pipeline packages.

Sequence database files, usually in FASTA format (*yeast.fasta*), can be created by downloading the list of protein sequences (usually related to the target species) from UniProt (see Relevant Websites section). In this example, a single FASTA file comprising of more than 10,000 protein sequences has been formed with all the 'reviewed' proteins found against organism 'fission yeast' in UniProt and PomBase (protein sequence resource of fission yeast) (see Relevant Websites section) repositories (see Fig. 6). The FASTA file should also contain decoy protein sequences in order to perform statistical evaluation of peptide-spectrum match (PSM) (at a later stage) using target-decoy approach. To generate decoy sequences, several methods are available such as shifting the precursor m/z ratio values of *in silico* generated spectra and, randomly shuffling and reversing the database sequences (Elias and Gygi, 2010) (see Fig. 7). Different freely accessible programs and scripts have been developed to accomplish this task and are listed below:

1. DecoyPYrat by Sanger Institute generates a set of decoy sequences through reversing the database sequences (Wright and Choudhary, 2016). It is a python-based script 'decoyPYrate.py', can be downloaded (see Relevant Websites section) and run on Python ver. 3 or above (see Relevant Websites section).

The screenshot displays the 'X!Tandem Search' interface within the 'ISB/SPC Trans Proteomic Pipeline'. The top navigation bar includes 'Home', 'Files', 'TPP Tools', 'External Tools', 'Account', and 'Jobs'. The main content area is divided into several sections:

- CHOOSE MZ[X]ML INPUT FILES**: Shows a text input field with the path 'c:/TPP/data/demo/tandem/yeast.mzML' and an 'Add Files' button.
- CHOOSE TANDEM PARAMETERS FILE**: Shows a text input field with the path 'c:/TPP/data/demo/tandem/tandem.params' and an 'Add Files' button.
- CHOOSE A SEQUENCE DATABASE**: Shows a text input field with the path 'c:/TPP/data/dbase/yeast.fasta' and an 'Add Files' button. Below this is a dropdown menu labeled 'Or choose from list:' with the selected path 'c:/TPP/data/dbase/ [*]'.
- OPTIONS**: Contains a checked checkbox labeled 'Convert output files to pepXML'.
- SEARCH!**: Features a prominent green 'Run Tandem Search' button.

Fig. 8 Trans-proteomic pipeline database search module and parameters (X! Tandem).

CHEMDATA.NIST.GOV
Mass Spectrometry Data Center

Trace: - yeastpombe

peptidew:lib:yeastpombe

Mass Spectrometry Data center

- Libraries/Tools/Service
- Publications
- Thermochemical data tables
- Download

Peptide Libraries

- Software
- Download Libraries

Team

Yeast Pombe Ion Trap Library

May 18, 2013

This library was assembled from data contributed by many laboratories. Each library entry contains annotation specifying its origin. Additionally, a summary table of the many contributors to this library, including references, can be downloaded [here](#) (Excel file).

File - Description	Size	Link
2013_05_18_yeast_pombe_consensus_final_true_lib.tar.gz - Consensus library in plain ASCII text (msp format). This format can be used as an exchange format. It is also directly readable by some software applications, including Skyline, after unzipping.	81MB	Download
2013_05_18_yeast_pombe_nist.tar.gz - Consensus library in NIST binary format. The folder generated by extracting this package can be used directly by MS Search or MSPepSearch and by the MSPepSearch node within Proteome Discoverer.	162MB	Download
2013_05_18_yeast_pombe_spectrast.tar.gz - Consensus library in SpectraST format. These library files can be directly searched by	195MB	Download

Fig. 9 Yeast spectral library from the NIST database.

- Decoy Database Builder (DecoyDBB), a module of PeakQuant Suite, build the decoy databases by reversing, reshuffling and random generation of target protein sequences (Reidegeld *et al.*, 2008). A stand-alone version of this java-based program is available at the link (see Relevant Websites section).

X! Tandem, Sequest and Comet perform *in silico* digestion of proteins in FASTA file according to the provided parameters, generate the theoretical spectra (with precursor m/z values), followed by matching it to the observed spectra and calculates the search scores (see Fig. 8).

Spectral libraries search

Investigation of MS spectra by inquiring previously identified spectral libraries is performed by employing SpectraST search module with the following input files:

- Mass spectrometry data files in mzML format, *yeast.mzML* (created in the previous step)
- Spectral libraries, *yeast.splib* (downloaded from NIST)
- SpectraST parameters file, *spectrast.params* (available online and can be modified)
- Sequence database file, *yeast.fasta* (including decoys)

Spectral libraries can be downloaded from different public repositories. These repositories contained highly-organized data for a number of species, for example, PeptideAtlas (Desiere *et al.*, 2006), SRMATlas (Kusebauch *et al.*, 2016), ProteomeXchange (Vizcaino *et al.*, 2014), and National Institute of Standards and Technology (NIST) (see Relevant Websites section), among others. The spectral library against *S. pombe*, used in this review, is the NIST 'Yeast Pombe Ion Trap Library'. This consensus library is created using spectra generated in different scientific laboratories (Pointner *et al.*, 2012; Gunaratne *et al.*, 2013), and is available at the link (see Relevant Websites section) (see Fig. 9). The different data formats of spectral libraries are *.blib, *.clib, *.msp, *.sptxt, and *.splib. SpectraST usually works with libraries in the *.splib format.

SpectraST parameter file contains information related to:

- File locations (path to input/output files)
- Precursor m/z tolerance (default=3.0)
- Structural and isotopic modifications
- Accept/ignore the abnormal spectra
- Peak selection options

SpectraST (see Fig. 10) matches the peaks in experimental and library spectra and calculates the dot product or *dotp* score (similarity measure). Values of *dotp* range from 0 to 1, where 0 means poor similarity and 1 means a perfect match.

Fig. 10 Trans-proteomic pipeline spectraST search module and parameters.

Search output

The output result of both the database and spectral library search includes a *yeast.log* and a *yeast.pepXML* files, which contains the following information (see Fig. 11):

1. Query spectra
2. Retention time values
3. Identified peptides and their ranks
4. Search scores
5. Dot product scores (similarity significance)
6. Sensitivity and error scores

De Novo and hybrid search approaches

Besides database and spectral matching, *de novo* sequencing, i.e. explicit translation of spectral peaks into the sequence, and hybrid approach, i.e. a combination of database search and *de novo*, are also available but are not commonly used, only utilized in specific cases. PepNovo by Centre for Computational Mass Spectrometry (CCMS) (Frank *et al.*, 2007) and Sequit by Proteome Factory (Demine and Walden, 2004) are highly efficient and compatible tools for *de novo* sequencing. These approaches have been discussed elsewhere (Islam *et al.*, 2017b). Guten Tag by Yates Laboratory (Tabb *et al.*, 2003) and Spectral Networks by CCMS (Bandeira, 2011) are competent tools for identification of peptides through a hybrid approach.

Step 3: Peptide-Spectrum Match Statistical Validation

Peptide-spectrum match (PSM) validation is performed through PeptideProphet (Ma *et al.*, 2012), iProphet (Shteynberg *et al.*, 2011) and PTMProphet (Shteynberg *et al.*, 2012) modules of TPP which estimates the false discovery rates (FDRs) for PSMs.

A.

```
<spectrum_query spectrum="yeast.00031.00031.3" start_scan="31" end_scan="31" precursor_neutral
<search_result>
  <search_hit hit_rank="1" peptide="RFAFRHLYNTEK" peptide_prev_aa="R" peptide_next_aa="F" pro
    <search_score name="hyperscore" value="366"/>
    <search_score name="nextscore" value="355"/>
    <search_score name="bscore" value="0"/>
    <search_score name="yscore" value="1"/>
    <search_score name="cscore" value="0"/>
    <search_score name="zscore" value="0"/>
    <search_score name="ascore" value="0"/>
    <search_score name="xscore" value="0"/>
    <search_score name="expect" value="3.7"/>
  </search_hit>
</search_result>
</spectrum_query>
```

B.

```
<spectrum_query spectrum="yeast.02301.02301.2" start_scan="02301" end_scan="02301" precursor_neutral_mass="1146
<search_result>
<search_hit hit_rank="1" peptide="LVNHFIQEF" peptide_prev_aa="R" peptide_next_aa="K" protein="UniRef100_059855"
<search_score name="delta" value="0.578"/>
<search_score name="dot_bias" value="0.000"/>
<search_score name="precursor_mz_diff" value="0.706"/>
<search_score name="hits_num" value="661"/>
<search_score name="hits_mean" value="0.096"/>
<search_score name="hits_stdev" value="0.062"/>
<search_score name="fval" value="0.790"/>
<search_score name="p_value" value="-1.000e+000"/>
<search_score name="KS_score" value="0.000"/>
<search_score name="first_non_homolog" value="2"/>
<search_score name="open_mod_mass" value="0.000"/>
<search_score name="open_mod_locations" value=""/>
<search_score name="charge" value="2"/>
<search_score name="lib_file_offset" value="168221077"/>
<search_score name="lib_probability" value="0.9910"/>
<search_score name="lib_status" value="Normal"/>
<search_score name="lib_num_replicates" value="3"/>
<search_score name="lib_remark" value="_NONE_"/>
</search_hit>
</search_result>
</spectrum_query>
```

Fig. 11 (A) Database search output from X! Tandem and (B) spectral library search output from SpectraST.

Additionally, standalone tools such as Scaffold by Proteome Software (Searle, 2010) and MAYU (Reiter *et al.*, 2009) can also be used for the statistical assessment. PeptideProphet is executed by assigning probabilities to PSMs and takes *yeast.pepXML* search output files as input. It takes into account the entire search score profile from pepXML files against correct and incorrect peptides and generates a 'correct match' probability for each result. On the TPP web interface, the program can be found in the 'Analyze Peptides' section of TPP tools (see Fig. 12) and run either in default mode or with other advanced options available for PeptideProphet such as considering pI information, phosphorylation or other PTMs information, and spectra RT values, among others. To filter out the correct identified peptides, significance level or 'PeptideProphet probability' is set to 0.05 (i.e. the probability of a peptide to be selected as correct by chance is less than 0.05) and minimum length of peptide is set to seven (7).

The output files generated by running PeptideProphet, on both Tandem and SpectraST files, are usually saved with the name *interact.pepXML*. These files comprise of statistical evaluation values for each of the peptides along with the peptide sequences and protein names from the protein sequence database file (*yeast.fasta*). The list of all the validated peptides can be viewed in TPP interface in tabular form where it can be sorted in a number of ways such as by probability, dot product values, protein names and *m/z* values (see Fig. 13). The peptides with the probability score close to 1 are most likely to be correct.

The program iProphet or InterProphet performs the statistical refinement of the PeptideProphet results. It takes the *interact.pepXML* files as input and generates *interact.ipro.pepXML* files with the refined probability score (see Fig. 13), also called as 'iprob' and can be analysed in a similar way as PeptideProphet results. The files, *interact.pepXML* and *interact.ipro.pepXML*, can also be viewed in other text format programs for a detailed overview.

Step 4: Protein Inference and Validation

In order to correctly infer the protein identity using the identified peptides, which were actually present in the sample, is a highly extensive task. The presence of a single peptide sequence in multiple proteins complicates the process and needs a comprehensive statistical assessment. Most commonly programs used to accomplish this task are DTASelect (Tabb *et al.*, 2002), DBParser by Mascot Distiller (Yang *et al.*, 2004), Scaffold by Proteome Software (Searle, 2010) and ProteinProphet (Nesvizhskii *et al.*, 2003) by TPP. In

ISB/SPC Trans Proteomic Pipeline :: Analyze Peptides

Home Files **TPP Tools** External Tools Account Jobs

SELECT FILE(S) TO ANALYZE ☒ show

c:/TPP/data/demo/tandem/yeast.tandem.pep.xml

Add Files Or choose from list: c:/TPP/data/demo/tandem/ [xml]

OUTPUT FILE AND FILTER OPTIONS

File path (folder): c:/TPP/data/demo/tandem

Write output to file: interact.pep.xml

Filter out results below this PeptideProphet probability: 0.05

Minimum peptide length considered in the analysis: 7

PEPTIDEPROPHET OPTIONS ☒ show

☒ **RUN PeptideProphet**

☐ Use accurate mass binning, using: PPM

☐ Use pI information

☐ Use Phospho information

RUN ANALYSIS!

Run XInteract

Fig. 12 Statistical validation of identified peptides using trans-proteomic pipeline "analyze peptides" module.

Summary Display Options Pick Columns Filtering Options Other Actions [Hide options]							
Sorting: iprobability desc asc PeptideProphet: min 1.0000 max iProphet: min 1.0000 max Update Page							
pepXML file: c:/TPP/data/interact-ipro.pep.xml trypsin digest, (iProphet analysis, combined results) search engine, quantitation: [none] displaying 7198 of 332919 total spectra, page 2 of 144 704 unique peptides, 420 unique stripped peptides, 285 unique proteins, 33 single hits PepXML Viewer, 2006-2016 SPC/ISB							
Page 2 of 144							
2 FIRST PREV 1 2 3 4 5 6 7 12 52 102 NEXT LAST							
IPOB	PROB	SPECTRUM	IONS	PEPTIDE	PROTEIN	CALC	MASS
1	1.0000	ysticat2 1012 1.02952.02952.3	[spectrum]	K.DVTFLNDC553.76VGPEVEAAVK.A	YCR012W	2356.7600	
1	1.0000	ysticat2 1012 1.01428.01428.2	[spectrum]	R.SGQGAFGNMC545.71R.G	YBR031W +1	1569.8308	
1	1.0000	ysticat2 1012 1.02934.02934.3	[spectrum]	K.DVTFLNDC553.76VGPEVEAAVK.A	YCR012W	2356.7600	
1	1.0000	ysticat2 1012 1.02082.02082.3	[spectrum]	R.AEOLYEGPADDANC545.71IAIK.N	YDR385W +1	2363.6817	
1	1.0000	ysticat2 1012 1.01762.01762.2	[spectrum]	K.YGAYSFC545.71TPK.E	YMR307W	1578.8603	
1	1.0000	ysticat2 1012 1.01412.01412.2	[spectrum]	R.SGQGAFGNMC553.76R.G	YBR031W +1	1577.8801	
1	1.0000	ysticat2 1012 1.01742.01742.2	[spectrum]	K.YGAYSFC545.71TPK.E	YMR307W	1578.8603	
1	1.0000	ysticat2 1012 1.02232.02232.2	[spectrum]	K.ELIFYC553.76ASGK.R	YOR285W	1580.9461	
1	1.0000	ysticat2 1012 1.02066.02066.3	[spectrum]	R.AEOLYEGPADDANC553.76IAIK.N	YDR385W +1	2371.7310	

Fig. 13 Visualization of pepXML files in trans-proteomic pipeline's web interface.


```

<protein_group group_number="369" probability="0.9954">
  <protein protein_name="YML123C" n_indistinguishable_proteins="1" probability="0.9954">
    <parameter name="prot_length" value="588"/>
    <annotation protein_description="PHO84 SGDID:S000004592, Chr XIII from 25801-24038, ...>
    <peptide peptide_sequence="VGAIIAQTALGTLIDHNCAR" initial_probability="0.9990" nsp_a
      <indistinguishable_peptide peptide_sequence="3_VGAIIAQTALGTLIDHNC[330]AR" charg
      <modification_info modified_peptide="VGAIIAQTALGTLIDHNC[330]AR"/>
    </indistinguishable_peptide>
      <indistinguishable_peptide peptide_sequence="3_VGAIIAQTALGTLIDHNC[546]AR" charg
      <modification_info modified_peptide="VGAIIAQTALGTLIDHNC[546]AR"/>
    </indistinguishable_peptide>
      <indistinguishable_peptide peptide_sequence="3_VGAIIAQTALGTLIDHNC[554]AR" charg
      <modification_info modified_peptide="VGAIIAQTALGTLIDHNC[554]AR"/>
    </indistinguishable_peptide>
    </peptide>
  </protein>
</protein_group>

```

Fig. 14 Statistical validation at the protein level using ProteinProphet.

the current example, ProteinProphet was used which utilizes the statistical values and peptide/protein related data from *interact.pepXML* and *interact.ipro.pepXML* files and generates the following information in a file called as *interact.ipro.protXML* (see Fig. 14):

1. List of identified proteins
2. List of identified peptides in each protein
3. Probability score against each identification
4. Calculated mass values for peptides
5. Number and nature of ions/transitions against each peptide entry (b/y ions)
6. Number of unique peptides, proteins and single hits

Step 5: Validation through HUPO Metrics

The Human Proteome Organisation (HUPO), recognising the need to standardise the parameters and conditions to allow for accurate validation of protein identifications, proposed a set of matrices that would increase confidence in MS results, primarily for 'missing proteins' (Baker *et al.*, 2017). A systematic approach to evaluating the quality of proteomics data can be obtained from Islam *et al.* (2017a) which encompasses the most recent communally accepted metrics (Omenn *et al.*, 2016) (see Relevant Websites section). Briefly, HUPO recommends applying a threshold of $\leq 1\%$ global FDR at the protein level, having two or more non-nested peptides that are ≥ 9 amino acids long in addition to other checks, such as calibration of instruments and reporting all FDR's, scores, databases, parameters (instrument and search) used and finally depositing data in a public repository. Although this is only applicable for missing proteins, applying such metrics in all protein identifications can greatly improve the confidence in proteins identified.

Step 6: Data Storage in Public Repositories

In order to eliminate the computational expense and data redundancy, all the mass spectrometer raw data and annotated files related to the specific sample, are encouraged to be submitted to publicly available data repositories. Some of the most common sources of MS-based proteomics identification data (other than above-mentioned library resources) are PRoteomics IDentification (PRIDE) by EBI (Vizcaino *et al.*, 2016), Computational Proteomics Analysis System (CPAS) (Rauch *et al.*, 2006), Open Proteomics Database (OPD) by Marcotte Lab (Prince *et al.*, 2004) and PeptideAtlas (Desiere *et al.*, 2006). The general conditions and requirements for data submission are as follows:

1. Targeted species and sample proteome.
2. Identify the type of MS data produced (MS or MS/MS).
3. Experiment protocols and instrumentation (growth, extraction, separation, digestion and informatics).
4. Usually, all the data files such as raw, mzML, pepXML, protXML etc.
5. Publication manuscript.

Discussion and Conclusions

The process of protein identification from mass spectrometry as is observed above is an iterative process that relies heavily on computation and statistical models and prediction. This introduces a significant number of variables and thresholds and

depending on the parameters and computational algorithm or search database used (or chosen), can result in variable outcomes (Ramos-Fernandez *et al.*, 2008). In addition, the starting material, the proteins themselves, are often highly heterogeneous both in structure and sequence with the same protein having multiple forms in form of splice variants, PTM's, single amino acid variations (SAAV's), and mutations, among others. Other compounding factors such as the resolution of different MS instruments, resolution and nature of sample preparation techniques and separation methodologies, the abundance of proteins, the ability of peptides to 'fly', and others, all play a role in the successful identification of a protein (Baldwin, 2004).

Despite the above, the probability-based statistical approach used in the identification of proteins in mass spectrometry over the last two decades has made significant improvements with more robust, reproducible and reliable computational methodologies allowing for greater confidence in data. Furthermore, improvements in quality and resolution of MS instrumentation also has had a significant impact in the identification of a large number of proteins accurately. Lastly, a communal recognition of the variability of data has led to initiatives such as that of HUPO to attempt to standardise and streamline the reporting of protein identifications and develop communally acceptable metrics for positive identification, though to date, no identification standards exist for PE1 (proteins with protein-level evidence). It is for this reason, that aside from orthogonal validation, the validation of MS data requires comprehensive and transparent reporting of all parameters and methodologies used (Martens and Hermjakob, 2007).

Further Analysis after Protein Identification

Upon accurate identification and validation of MS data, the list of identified proteins can be further processed through a series of post-data analysis computational and informatics pipelines to allow annotation (Islam *et al.*, 2014), quantification, expression or network analysis, gene ontology analysis, among others. One of the primary ways to look at detailed annotation is to copy the protein ID from the output table (step 4) and search against UniProtKB database – one of the biggest protein databases. UniProtKB has accumulated, for every protein, the information related to its function, taxonomy, modifications, interactions, families and domains, among others. It also contains cross-references to additional databases such as GO for gene annotation, IntAct, STRING and MINT for molecular interactions, RCSB PDB for 3D structures against target protein sequence, SMART, Pfam and PROSITE for family and domains, Ensembl database for phylogenetics and KEGG for network analysis. The detailed post-data analysis methodologies and tools are beyond the scope of this article but have been discussed elsewhere (Bessarabova *et al.*, 2012; Schmidt *et al.*, 2014). Often the information from a post-MS data analysis is used to inform and select candidates for further orthogonal validation of the identification.

Orthogonal Validation of Identity

The most efficient and commonly acceptable way to confirm the identity of a protein is to use orthogonal techniques, especially in sensitive areas such as biomarker discovery (Rifai *et al.*, 2006). The most common form of orthogonal validation is that of affinity-based identification such as western blotting, ELISA or protein/peptide arrays whereas more robust or stringent applications may indeed use functional (biochemical or biological) assays (Rabilloud and Lescuyer, 2014) to validate identifications, functional annotations and identities. As a discovery and hypothesis generating tool, MS-based identification in proteomics of multiple proteins remains the most efficient and accurate as long as accepted and standardised informatics methodologies are applied.

See also: Clinical Proteomics. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis. Sequence Composition

References

- Aebersold, R., Mann, M., 2003. Mass spectrometry-based proteomics. *Nature* 422, 198–207.
- Baker, M.S., Ahn, S.B., Mohamedali, A., *et al.*, 2017. Accelerating the search for the missing proteins in the human proteome. *Nat Commun* 8, 14271.
- Baldwin, M.A., 2004. Protein identification by mass spectrometry: Issues to be considered. *Mol. Cell Proteomics* 3, 1–9.
- Bandeira, N., 2011. Protein identification by spectral networks analysis. *Methods Mol. Biol.* 694, 151–168.
- Benevento, M., Di Palma, S., Snijder, J., *et al.*, 2014. Adenovirus composition, proteolysis, and disassembly studied by in-depth qualitative and quantitative proteomics. *J. Biol. Chem.* 289, 11421–11430.
- Bessarabova, M., Ishkin, A., Jebailey, L., Nikolskaya, T., Nikolsky, Y., 2012. Knowledge-based analysis of proteomics data. *BMC Bioinformatics* 13 (Suppl 16), S13.
- Cagney, G., Amiri, S., Premawardena, T., Lindo, M., Emili, A., 2003. *In silico* proteome analysis to facilitate proteomics experiments using mass spectrometry. *Proteome Sci.* 1, 5.
- Craig, R., Beavis, R.C., 2003. A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Commun. Mass Spectrom.* 17, 2310–2316.
- Demine, R., Walden, P., 2004. Sequit: Software for de novo peptide sequencing by matrix-assisted laser desorption/ionization post-source decay mass spectrometry. *Rapid Commun. Mass Spectrom.* 18, 907–913.
- Desiere, F., Deutsch, E.W., King, N.L., *et al.*, 2006. The PeptideAtlas project. *Nucleic Acids Res.* 34, D655–D658.
- Deutsch, E.W., Mendoza, L., Shteynberg, D., *et al.*, 2015. Trans-Proteomic Pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics. *Proteomics Clin. Appl.* 9, 745–754.
- Domon, B., Aebersold, R., 2006. Mass spectrometry and protein analysis. *Science* 312, 212–217.
- Elias, J.E., Gygi, S.R., 2010. Target-decoy search strategy for mass spectrometry-based proteomics. *Proteome Bioinformatics* 604, 55–71.

- Eng, J.K., Fischer, B., Grossmann, J., Maccoss, M.J., 2008. A fast SEQUEST cross correlation algorithm. *J. Proteome Res.* 7, 4598–4602.
- Eng, J.K., Jahan, T.A., Hoopmann, M.R., 2013. Comet: An open-source MS/MS sequence database search tool. *Proteomics* 13, 22–24.
- Frank, A.M., Savitski, M.M., Nielsen, M.L., Zubarev, R.A., Pevzner, P.A., 2007. De novo peptide sequencing and identification with precision mass spectrometry. *J. Proteome Res.* 6, 114–123.
- Frewen, B., Maccoss, M.J., 2007. Using BiblioSpec for creating and searching tandem MS peptide libraries. *Curr. Protocols Bioinformatics*. 13.7. 1–13.7. 12.
- Gauci, S., Helbig, A.O., Slijper, M., *et al.*, 2009. Lys-N and trypsin cover complementary parts of the phosphoproteome in a refined SCX-based approach. *Anal. Chem.* 81, 4493–4501.
- Giansanti, P., Tsatsiani, L., Low, T.Y., Heck, A.J., 2016. Six alternative proteases for mass spectrometry-based proteomics beyond trypsin. *Nat. Protoc.* 11, 993–1006.
- Gunaratne, J., Schmidt, A., Quandt, A., *et al.*, 2013. Extensive mass spectrometry-based analysis of the fission yeast proteome: The *Schizosaccharomyces pombe* PeptideAtlas. *Mol. Cell. Proteomics* 12, 1741–1751.
- Herbert, B., Hopwood, F., Oxley, D., *et al.*, 2003. beta-elimination: An unexpected artefact in proteome analysis. *Proteomics* 3, 826–831.
- Huynh, M.L., Russell, P., Walsh, B., 2009. Tryptic digestion of in-gel proteins for mass spectrometry analysis. *Methods Mol Biol* 519, 507–513.
- Islam, M.T., Garg, G., Hancock, W.S., *et al.*, 2014. Protannotator: A semiautomated pipeline for chromosome-wise functional annotation of the “missing” human proteome. *J. Proteome Res.* 13, 76–83.
- Islam, M.T., Mohamedali, A., Ahn, S.B., *et al.*, 2017a. A Systematic Bioinformatics Approach to Identify High Quality Mass Spectrometry Data and Functionally Annotate Proteins and Proteomes. *Methods Mol. Biol.* 1549, 163–176.
- Islam, M.T., Mohamedali, A., Fernandes, C.S., Baker, M.S., Ranganathan, S., 2017b. De Novo peptide sequencing: Deep mining of high-resolution mass spectrometry data. *Methods Mol. Biol.* 1549, 119–134.
- Islam, M.T., Mohamedali, A., Garg, G., *et al.*, 2013. Unlocking the puzzling biology of the black perigord truffle tuber *melanosporum*. *J. Proteome Res.* 12, 5349–5356.
- Kessner, D., Chambers, M., Burke, R., Agus, D., Mallick, P., 2008. ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* 24, 2534–2536.
- Kim, M.S., Pinto, S.M., Getnet, D., *et al.*, 2014. A draft map of the human proteome. *Nature* 509, 575.
- Kingsmore, S.F., 2006. Multiplexed protein measurement: Technologies and applications of protein and antibody arrays. *Nat. Rev. Drug Discovery* 5, 310–320.
- Koenig, T., Menze, B.H., Kirchner, M., *et al.*, 2008. Robust prediction of the MASCOT score for an improved quality assessment in mass spectrometric proteomics. *J. Proteome Res.* 7, 3708–3717.
- Kohlbacher, O., Reinert, K., Gropi, C., *et al.*, 2007. TOPP – the OpenMS proteomics pipeline. *Bioinformatics* 23, E191–E197.
- Kusebauch, U., Campbell, D.S., Deutsch, E.W., *et al.*, 2016. Human SRMAtlas: A resource of targeted assays to quantify the complete human proteome. *Cell* 166, 766–778.
- Lam, H., Deutsch, E.W., Edes, J.S., *et al.*, 2007. Development and validation of a spectral library searching method for peptide identification from MS/MS. *Proteomics* 7, 655–667.
- Lindskog, C., 2015. The potential clinical impact of the tissue-based map of the human proteome. *Expert Rev. Proteomics* 12, 213–215.
- Low, T.Y., Van Heesch, S., Van Den Toorn, H., *et al.*, 2013. Quantitative and qualitative proteome characteristics extracted from in-depth integrated genomics and proteomics analysis. *Cell Rep.* 5, 1469–1478.
- Ma, K., Vitek, O., Nesvizhskii, A.I., 2012. A statistical model-building perspective to identification of MS/MS spectra with PeptideProphet. *BMC Bioinformatics* 13 (Suppl 16), S1.
- Martens, L., Hermjakob, H., 2007. Proteomics data validation: Why all must provide data. *Mol. Biosyst.* 3, 518–522.
- Marx, V., 2013. Finding the right antibody for the job. *Nat. Methods* 10, 703–707.
- Nesvizhskii, A.I., Keller, A., Kolker, E., Aebersold, R., 2003. A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.* 75, 4646–4658.
- Ngounou Wetie, A.G., Woods, A.G., Darie, C.C., 2014. Mass spectrometric analysis of post-translational modifications (PTMs) and protein-protein interactions (PPIs). *Adv. Exp. Med. Biol.* 806, 205–235.
- Omenn, G.S., Lane, L., Lundberg, E.K., *et al.*, 2016. Metrics for the Human Proteome Project 2016: Progress on identifying and characterizing the human proteome, including post-translational modifications. *J. Proteome Res.* 15, 3951–3960.
- Pointner, J., Persson, J., Prasad, P., *et al.*, 2012. CHD1 remodelers regulate nucleosome spacing in vitro and align nucleosomal arrays over gene coding regions in *S. pombe*. *EMBO J.* 31, 4388–4403.
- Prince, J.T., Carlson, M.W., Wang, R., Lu, P., Marcotte, E.M., 2004. The need for a public proteomics repository. *Nat. Biotechnol.* 22, 471–472.
- Rabilloud, T., Lescuyer, P., 2014. The proteomic to biology inference, a frequently overlooked concern in the interpretation of proteomic data: A plea for functional validation. *Proteomics* 14, 157–161.
- Ramos-Fernandez, A., Parada, A., Navajas, R., Albar, J.P., 2008. Generalized method for probability-based peptide and protein identification from tandem mass spectrometry data and sequence database searching. *Mol. Cell Proteomics* 7, 1748–1754.
- Rauch, A., Bellew, M., Eng, J., *et al.*, 2006. Computational proteomics analysis system (CPAS): An extensible, open-source analytic system for evaluating and publishing proteomic data and high throughput biological experiments. *J. Proteome Res.* 5, 112–121.
- Reidegeld, K.A., Eisenacher, M., Kohl, M., *et al.*, 2008. An easy-to-use Decoy Database Builder software tool, implementing different decoy strategies for false discovery rate calculation in automated MS/MS protein identifications. *Proteomics* 8, 1129–1137.
- Reiter, L., Claassen, M., Schrimpf, S.P., *et al.*, 2009. Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry. *Mol. Cellular Proteomics* 8, 2405–2417.
- Rifai, N., Gillette, M.A., Carr, S.A., 2006. Protein biomarker discovery and validation: The long and uncertain path to clinical utility. *Nat. Biotechnol.* 24, 971–983.
- Schmidt, A., Forne, I., Imhof, A., 2014. Bioinformatic analysis of proteomics data. *BMC Syst. Biol.* 8 (Suppl 2), S3.
- Searle, B.C., 2010. Scaffold: A bioinformatic tool for validating MS/MS-based proteomic studies. *Proteomics* 10, 1265–1269.
- Shteynberg, D., Deutsch, E., Mendoza, L., *et al.*, 2012. PTMPephet: TPP software for validation of modified site locations on post-translationally modified peptides. In: *Proceedings of the 60th ASMS Conference on Mass Spectrometry*, Vancouver, BC, Canada, pp. 20–24.
- Shteynberg, D., Deutsch, E.W., Lam, H., *et al.*, 2011. iProphet: Multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates. *Mol Cell Proteomics* 10.[M1111 007690].
- Tabb, D.L., McDonald, W.H., Yates, J.R., 2002. DTASelect and Contrast: Tools for assembling and comparing protein identifications from shotgun proteomics. *J. Proteome Res.* 1, 21–26.
- Tabb, D.L., Saraf, A., Yates 3RD, J.R., 2003. GutenTag: High-throughput sequence tagging via an empirically derived fragmentation model. *Anal Chem* 75, 6415–6421.
- Thiede, B., Hohenwarter, W., Krah, A., *et al.*, 2005. Peptide mass fingerprinting. *Methods* 35, 237–247.
- Trudgian, D.C., Mirzaei, H., 2012. Cloud CFP: A shotgun proteomics data analysis pipeline using cloud and high performance computing. *Journal of Proteome Research* 11, 6282–6290.
- Uhlen, M., Fagerberg, L., Hallstrom, B.M., *et al.*, 2015. Tissue-based map of the human proteome. *Science*. 347.
- Vizcaino, J.A., Csordas, A., Del-Toro, N., *et al.*, 2016. 2016 update of the PRIDE database and its related tools. *Nucleic Acids Res.* 44, D447–D456.
- Vizcaino, J.A., Deutsch, E.W., Wang, R., *et al.*, 2014. ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotechnol.* 32, 223–226.
- Wilkins, M.R., Gasteiger, E., Bairoch, A., *et al.*, 1999. Protein identification and analysis tools in the ExPASy server. *Methods Mol. Biol.* 112, 531–552.

Wright, J.C., Choudhary, J.S., 2016. DecoyPyrat: Fast non-redundant hybrid decoy sequence generation for large scale proteomics. *J. Proteomics Bioinform.* 9, 176–180.

Yang, X., Dondeti, V., Dezube, R., *et al.*, 2004. DBParser: Web-based software for shotgun proteomic data analyses. *J. Proteome Res.* 3, 1002–1008.

Further Reading

Kumar, C., Mann, M., 2009. Bioinformatics analysis of mass spectrometry-based proteomics data sets. *FEBS Lett.* 583 (11), 1703–1712.

Martins-De-Souza, D., 2014. Shotgun Proteomics. *Methods Mol. Biol.* 1156.

Matthiesen, R. (Ed.), 2007. *Mass Spectrometry Data Analysis in Proteomics 1*. Totowa, NJ: Humana Press.

Schmidt, A., Forne, I., Imhof, A., 2014. Bioinformatic analysis of proteomics data. *BMC Syst. Biol.* 8 (2), S3.

Smith, R., Mathis, A.D., Ventura, D., Prince, J.T., 2014. Proteomics, lipidomics, metabolomics: A mass spectrometry tutorial from a computer scientist's point of view. *BMC Bioinformatics* 15 (7), S9.

Veenstra, T.D., Yates, J.R., 2006. *Proteomics for Biological Discovery*. John Wiley & Sons.

Wu, C.H., Chen, C. (Eds.), 2011. *Bioinformatics for Comparative Proteomics*. Humana Press.

Relevant Websites

<http://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:lib:yeastpombe>
CHEMDATA.NIST.GOV.

<http://ftp.sanger.ac.uk/pub/resources/software/decoypyrat/>
DecoyPYrate.py.

<https://hupo.org/Guidelines>
HUPO.

<http://www.nist.gov>
National Institute of Standards and Technology.

<http://ftp.peptideatlas.org/pub/PeptideAtlas/Repository/PAe003810>
PeptideAtlas.

<https://www.pombase.org/>
PomBase.

<https://www.python.org/>
Python.

<https://www.ruhr-uni-bochum.de/mpc/software/DecoyBuilder/index.html.en>
RUB.

<https://sourceforge.net/projects/sashimi/files/latest/download?source=files>
SOURCEFORGE.

<https://www.thermofisher.com/au/en/home.html>
ThermoFisher.

<http://www.uniprot.org/>
UniProt.

Biographical Sketch



Zainab is a PhD student in the department of Chemistry and Biomolecular sciences (Bioinformatics group) at Macquarie University. She has been working in the area of protein structure and function analysis during her bachelors and masters' studies. Previously, she worked on identifying novel potent drug-like compounds against histone deacetylases proteins by employing *in silico* drug designing techniques. Currently, she is involved in programming-based mass spectrometry data analysis of data generated through data-dependent acquisition (DDA) and data-independent acquisition (DIA) approaches related to colorectal cancer. More specifically, she is dealing with the customization of DDA based spectral libraries to be used in DIA data analysis for biomarker discovery.



Abidali is a Lecturer at Macquarie University in the Faculty of Science and Engineering. He completed his PhD in 2010 studying, using proteomics, the effects of single mutations in mouse model on brain development in the autism spectrum disorder, Rett syndrome. He then returned to Macquarie University undertaking a post-doc to study the biology of metastasis in colorectal cancer using proteomics and to determine novel therapeutic targets and develop novel technologies to examine the plasma proteome.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia, as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books, as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology, as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.

Quantification of Proteins From Proteomic Analysis

Zainab Noor, Subash Adhikari, Shoba Ranganathan, and Abidali Mohamedali, Macquarie University, Sydney, NSW, Australia

© 2019 Elsevier Inc. All rights reserved.

Protein Quantification – Introduction

All living biological entities rely on an intricate balance of metabolic activity, bio-molecules and non-organic molecules to maintain life. Inherent to this balance is the ability to respond rapidly to changes from within or without. Although the breadth of this response can be vast from activating enzymes to opening channels to allow ions to leave or enter, its temporal nature can be varied leading to permanent changes due to chronic signals or transient changes due to acute signals and every variation in between. One of the most fundamental changes that occur in cells is the variation in the abundance of proteins (Vogel and Marcotte, 2012) in response to external or internal stimuli. Indeed, it is well understood that although transient and rapid alterations in cells is often due to modification of the activity of proteins, sustained stimuli lead to sizeable changes in the proteome (the protein complement) of an organism due to changes in transcriptional and translational activity; therefore, protein abundance. This fact has been the source of understanding how biological systems respond to change which has led to the development of markers of change that have revolutionized medicine. An illustration is the discovery of Early pregnancy factor (EPF) as an indicator of early pregnancy (Fan and Zheng, 1997) has revolutionized early maternity practice.

Although it has been known that multiple proteins change expression in relation to stimuli (or treatment), most studies were able to measure and quantitate only a small number of proteins at a time using immunological methodologies (such as, ELISA and Western blotting). Upon the advent of mass spectrometry, researchers began to investigate how this new method could not only identify but also quantitate multiple proteins. The rapid expansion of the technologies and the development of novel label-based quantification and methodologies of label-free quantification using spectral counting led to significant advances in the field. By early 2007, it was possible to quantitate thousands of proteins in a single experiment. These methodologies had significant drawbacks especially around experiment design, technical variability, reproducibility and reliability (Bantscheff *et al.*, 2007). A more recent and powerful technique, loosely based on targeted quantification (multiple reaction monitoring), using a Data Independent Acquisition (DIA) mode on a mass spectrometer has made quantification far more reliable and robust. Fig. 1 summarises the most common methodologies used to quantitate proteins. This book article will discuss in brief detail some of the most common proteomic methodologies of protein quantification, with a particular emphasis on DIA quantification. Using an example, the article will guide a reader through setting up their own data analysis workflow and to apply statistical and experimental validation techniques to accurately quantify thousands of proteins from a biological sample.

Label-Based Quantification

The basic principle of label-based protein quantification technique is labelling or tagging of peptides/proteins with stable heavy isotopes (^{13}C , ^{15}N , ^{18}O and ^2H) and comparing the unlabelled 'light' form of the sample with labelled 'heavy' variant in mass spectrometry. In a given mass spectrum, a mass shift of 3–4 Da occurs in a heavy isoform where the ratio of peak intensities of both isoforms depict the ratio of abundances (Lindemann *et al.*, 2017). This method has been employed in studying modified proteins, membrane proteins, quantifying disturbances or perturbations in a system due to changes in proteins and effects of drugs or inhibitory compounds on protein expression. Label-based quantification is limited by the number of labels that can be applied at the same time (multiplexing) such that up to 12 samples can currently be studied simultaneously (King *et al.*, 2017).

Label-Free Quantification

Label free quantitative proteomics is an extensively used semi-quantitative approach capable of multiplexing multiple MS runs into a single experiment. This approach has wide application in high throughput applications with large individual variation and large sample size, such as in clinical proteomics. Label free quantification in such case can yield more analytical depth and higher dynamic range for quantification (Distler *et al.*, 2016).

Label free quantification is based on comparison of precursor ion intensities between MS runs, measured in terms of peak area or peak intensity after feature alignment according to retention time, m/z , charge states etc. A reproducible LC-MS system is an absolute requirement for efficient feature alignment (Norbeck *et al.*, 2005). Label free quantification requires resource intensive computational capability for feature detection and alignment (Mueller *et al.*, 2008), but unlike label-based quantification, the only limit to number of samples that can be analyzed is computational.

Regardless of platform and method adopted, label free quantification relies on extracted ion chromatogram (XIC) intensity-based quantitation or spectral counting-based quantitation. It has been observed that ion concentration correlates with ESI signal intensity; that is, higher peptide concentration in sample yields higher area under integrated chromatogram of a MS spectra. Relative area under curve/chromatogram (peak area) or peak intensity between samples can be used for relative quantitation of peptides/proteins across samples (Neilson *et al.*, 2011).

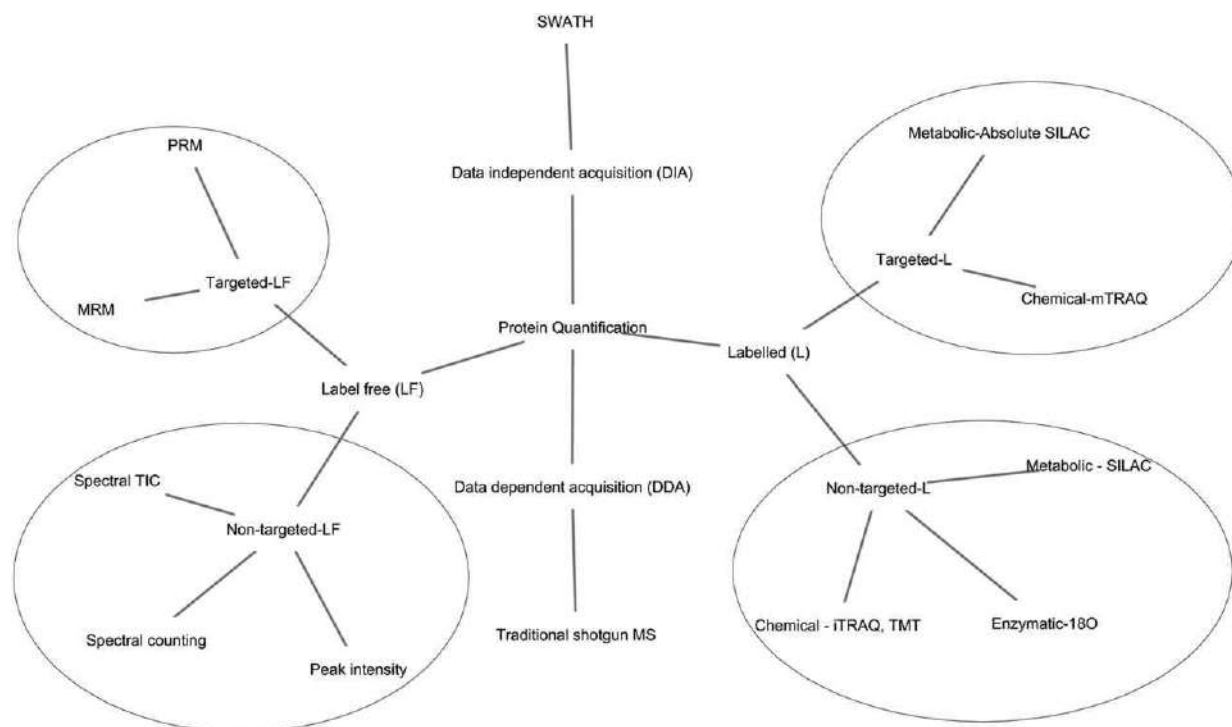


Fig. 1 Multiple approaches exist for MS based protein quantification. Labelling quantifies protein on basis of difference in signal intensities between heavy and light labelled peptides from different samples combined into a single run. Label free quantification is performed by measuring relative signal intensities between multiple sample runs. Quantifications are accomplished on DDA and DIA mode, depending on targeted and comprehensive data analysis.

Spectral counting-based quantification relies on assumption that number of tandem mass spectra are proportional to abundance of a protein, that more spectra are observed with increasing peptide concentration in the sample (Washburn *et al.*, 2001). Normalization factors relating to length and physiological properties of a peptide and its ability to be fragmented are taken into consideration (Ahme *et al.*, 2013). One challenge with combining XIC-based quantitation with DDA-based identification is finding a balance between MS (quantitation) and MS² (identification) cycling. Devoting more scanning time to peptide identification decreases the number of data points available for MS quantitation, which limits the resolution, and hence accuracy of XIC-based quantitation.

Relative Quantification

Relative quantification is the measure of change in protein expression patterns as a function of biological function, expressed in terms of ratios between different biological states. Relative quantification is measured by either of labelled-based or label-free quantification, depending upon scope of the experiment. A number of labelled-based relative quantification strategies have been developed which are characterized by their mode of action. They are broadly classified into three categories (i) metabolic labelling, biological incorporation of isotopes or labels in a cell culture; (ii) chemical labelling, integration of heavy isotopes into the proteins using isotopic reagents or isobaric tags and (iii) enzymatic labelling, labelling with heavy isotope of oxygen during hydrolysis of protein through trypsin or other proteases. Similarly, various label-free approach could be adopted for relative quantification.

Absolute Quantification

Absolute quantification is the process of measuring the exact amount of a protein in a proteome. Proteins of interest (sample proteins) are spiked with known concentrations of labelled standard peptides bearing identical sequence and physicochemical properties as sample peptides. Quantification is performed by comparing the intensities of the sample and standard peptides/proteins, where unknown quantities of sample proteins can be deduced by means of known concentrations of standard proteins. Several peptide or protein-based approaches have been designed for absolute quantification including absolute SILAC, protein standard absolute quantification (PSAQ), full-length expressed stable isotope-labelled quantification (FLEXIQuant) and quantification Concatemers (QconCAT). Similarly, in the label-free context, MRM assays use the same approach described to accurately quantify selected peptide transitions and hence infer absolute protein abundance.

Quantification Through Data Independent Acquisition (DIA)

During the last decade, most of the MS-based proteomics studies were based on a discovery-based shotgun proteomics method, run in data-dependent acquisition (DDA) mode. Although, this helps to maximize amount of information obtained from an experiment, it has certain inherent limitations such as scan speed (Wenner and Lynn, 2004), selection of ion for further fragmentation, poor reproducibility, narrow dynamic range etc. (Law and Lim, 2013). Due to biases towards certain most intense fragments, DDA method has limitation of not being able to detect low abundant ion fragments. Targeted MS-based proteomics was developed to overcome these limitations of DDA mode. Targeted MS approach on Data independent mode (DIA) is dependent on a known reference spectrum, where identification could be performed from selected few fragment ion, rather than all possible fragment in DDA mode (Gillette and Carr, 2013). Success of data independent quantification relies on proper selection of distinct precursor ion-transitions (Tang et al., 2014). Hence, DDA mode is employed to filter top performing precursor-ions as inclusion list for DIA mode, so the MS instrument could select those specific fragments to aid in identification of low abundant ions/spectra.

Sequential Window Acquisition of all THEoretical fragment ion spectra coupled to Tandem Mass Spectrometry (SWATH-MS) is a novel identification/quantification methodology that operates in DIA mode in a label-free context. SWATH-MS, currently takes into account the peptides of mass/charge (m/z) range from 400 to 1200 and generates fragment ion spectra of all the precursor peptides falling within the window width of 25 Da in each cycle. A prerequisite assay library of peptides/proteins of interest, created in shotgun or discovery mode is required to identify and translate the detailed fragment ion spectral data which is produced in DIA mode and extracted through targeted data extraction approach (Vidova and Spacil, 2017). While performing the data analysis, a number of parameters are usually taken into consideration to perform efficient and high-confidence data analysis. These parameters are, precursor and product ion mass/charge values, retention times, generated spectra and relative intensities. Scores are calculated against peak groups and statistical validation is performed (see Fig. 2).

In this article, label-free relative quantification analysis using computational tools have been discussed and demonstrated on DIA data. Quantification data setup and pre-requisite steps to extract the useful information and measurements from experimental data have been illustrated in detail. Moreover, statistical analysis has been carried out on quantified experimental data to analyze proteome expression.

Data Analysis Software

The data independent acquisition strategy has led to the development of a number of specialized software and packages which implement high-speed accurate DIA analysis, from chromatogram extraction to quantification. Currently, in addition to the generalized data analysis packages, mass spectrometry vendors also encompass built-in software responsible for not only

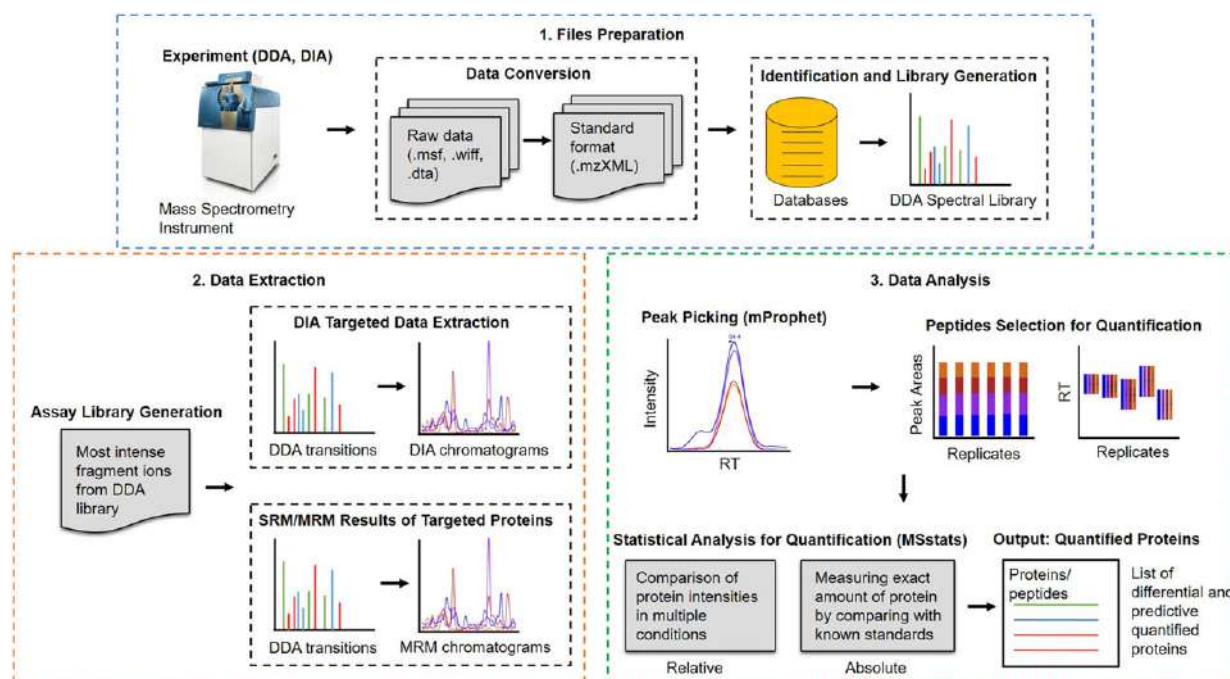


Fig. 2 Workflow of mass spectrometry based protein quantification. Data generated either in DIA or DDA (for the library) mode from a TOF or Orbitrap instrument are first converted to suitable standards followed by library generation from DDA data. Peaks are then extracted and picked for quantification. Finally, statistical analysis is performed to identify a list of differentially quantified proteins.

Table 1 Commonly used DIA (Data Independent Acquisition) data analysis software packages

Software	Enterprise	Specifications	Versions	Download link
PeakView	AB Sciex	● Commercial ● Windows-based ● Bio Tool Kit	V 1.1 (2010–2011) V 2.2 (2014)	https://sciex.com/products/software/peakview-software
OpenSWATH (Rost <i>et al.</i> , 2014)	OpenMS	● Free – Open Source ● Windows, Linux, Mac-based ● Compatible with AB Sciex, Thermo and Water DIA data	V 2.2.0 (2017) V 2.1.0 (2016)	https://github.com/OpenMS/OpenMS/releases
Skyline (Pino <i>et al.</i> , 2017)	MacCoss lab	● Free ● Windows-based ● Panorama repository	12 versions V 3.3 (2017)	https://skyline.ms/project/home/software/Skyline/begin.view
Spectronaut (Bruderer <i>et al.</i> , 2017)	Biognosys	● Commercial ● Windows-based ● Quantification of 1000 proteins from a single run	5 versions V 10.0 – Orion (latest)	https://biognosys.com/shop/spectronaut
DIA-Umpire (Tsou <i>et al.</i> , 2015)	Alexey Nesvizhskii lab	● Free – Open Source ● Untargeted, library-free peptide identification from DIA data	V 1.4 (2015) V 2.0 (2016) V 2.1 (2016)	http://diaumpire.sourceforge.net/

monitoring the MS experiment but also performing all the downstream data analysis. Some of the widely used software for DIA data analysis are presented in [Table 1](#).

Sample Dataset

The sample data was collected for the detection and quantification studies of yeast (*Saccharomyces cerevisiae*) proteome (Selevsek *et al.*, 2015). In this study, changes in yeast proteome expression have been analyzed in the presence of osmotic stress, progressively at different times. As stated above, DIA data analysis requires initial DDA runs to generate the library of assays. It is preferred to run the shotgun injections on the same instrument type as the final DIA data, however, the library assays can be transferred to different instrument platforms if similar fragmentation and chromatography techniques are adopted. In the current example, shotgun and DIA runs were performed on a 5600 TripleTof mass spectrometer (ABSciex) with NanoLC-2Dplus HPLC system. The instrument setup, parameters, collision energy and window settings are available in the study (Selevsek *et al.*, 2015). Additionally, all the data files were submitted to the ProteomeXchange repository (PRIDE) (ID: PXD001010) (see “Relevant Websites section”). These data files include DDA and DIA .wiff files which can be accessed and utilized for performing the quantitative analysis.

Computational Tools

For this article, to perform the identification and quantification of yeast proteome using DIA strategy, a number of tools were employed. For instance, DDA data was identified using TPP (Deutsch *et al.*, 2015). The identified spectra were converted into spectral libraries using SpectraST (Lam *et al.*, 2007) and Skyline. Targeted data extraction was performed in Skyline (Pino *et al.*, 2017). Peak scoring and statistical analysis were executed in Skyline using mProphet algorithm (Reiter *et al.*, 2011). Lastly, protein level relative quantification and significance analysis were carried out using MSstats functionalities (Choi *et al.*, 2014), supported in Skyline. Skyline software (see [Fig. 3](#)) can be freely downloaded, with mProphet and MSstats modules incorporated in Skyline.

Step1: Spectral Identification and Library Generation

The DDA-based library was generated by running shotgun experiments, comprising of 46 MS injections. These include samples from BY4741 strain and at different time points in response to osmotic shock. Details of these 46 injections are provided in the study (Selevsek *et al.*, 2015). The 46 data files are available online in wiff format with the following names:

1. ‘nselevse_L120203_006.wiff’ to ‘nselevse_L120203_029.wiff’ (24 files);
2. ‘nselevse_L120327_001.wiff’ to ‘nselevse_L120327_018.wiff’ (18 files);
3. ‘igillet_L120122_001.wiff’ to ‘igillet_L120122_003.wiff’ (3 files), and
4. ‘igillet_L120124_003.wiff’ (1 file).

The generated spectra were identified using TPP tools against *Saccharomyces* Genome Database (SGD) (Cherry *et al.*, 2012). The search results were statistically sorted according to probability at 1% FDR. The identified spectra from all the samples were then compiled to generate ‘iProphet_Combined.pepXML’ file and used to build a consensus spectral library.

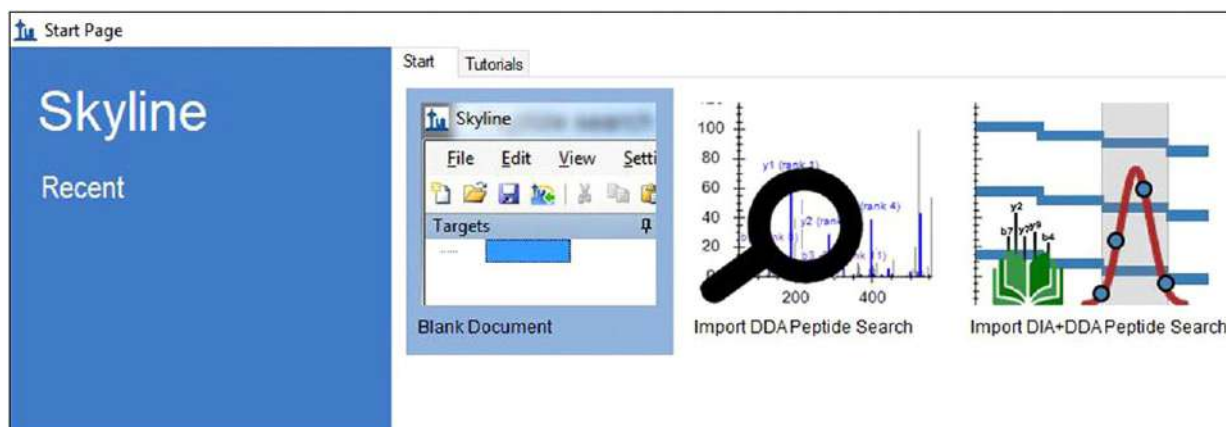


Fig. 3 Skyline software start-up page.

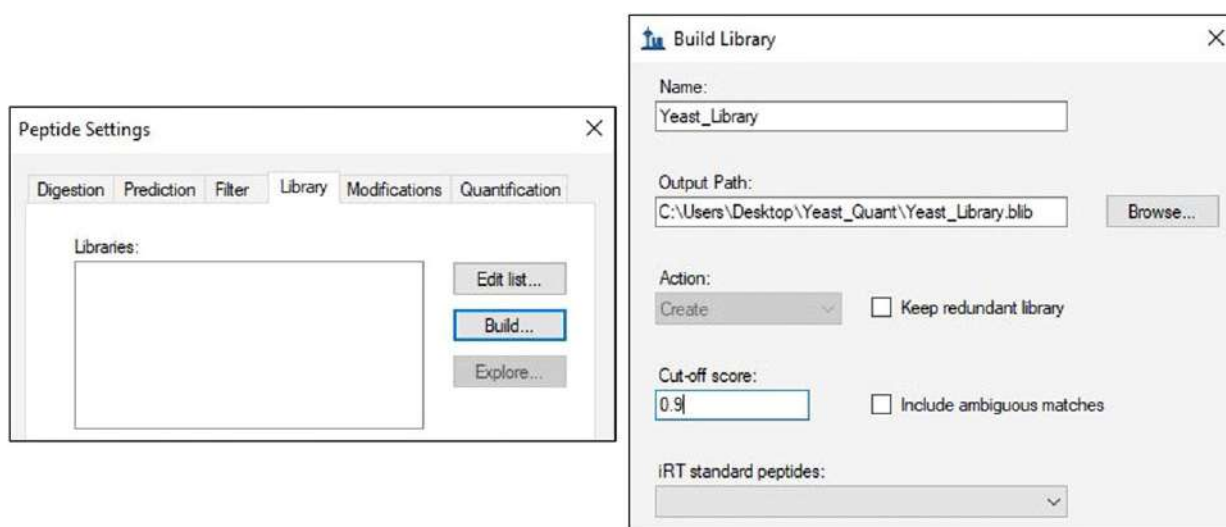


Fig. 4 Library settings in Skyline for generation of spectral library.

Building a spectral library requires the following files:

- pepXML file (search engine result)
- mzXML files (spectral data)

The pepXML file is available at ProteomeXchange repository (ID: PXD001010) whereas mzXML spectra files can be acquired by converting 46 DDA wiff files to XML format using Msconvert tool (Kessner *et al.*, 2008). These converted mzXML data files are necessary to extract the chromatograms from the spectral library during DIA targeted data extraction.

Spectral library using Skyline

Skyline has a 'Build Library' module in 'Peptide Settings' to generate the spectral library (see Fig. 4). It requires library name as 'Yeast_Library.blib', output path and PeptideProphet cut-off score as '0.9' (false discovery rate of below 1%). Retention times, mass to charge values and relative intensity values for each spectrum are also stored in the spectral library, which can be visualized in the Skyline (see Fig. 5).

Step 2: Fragment Ion Library or Assay Library

Fragment ion library or assay library is a subset of the spectral library. It contains the most intense particular precursor and product ions (MS/MS transitions) from the spectral library or a targeted proteins list. These assays contribute in DIA targeted data extraction of peptides and proteins of interest. These characteristics have to be determined prior to importing the DIA data into the Skyline. For this purpose, Skyline provides various peptide and transition settings with 'Filter', 'Library' and 'Modifications' modules. 'Modification' allows one to include and exclude different types of structural and isotopic modifications. In addition to the default options, new

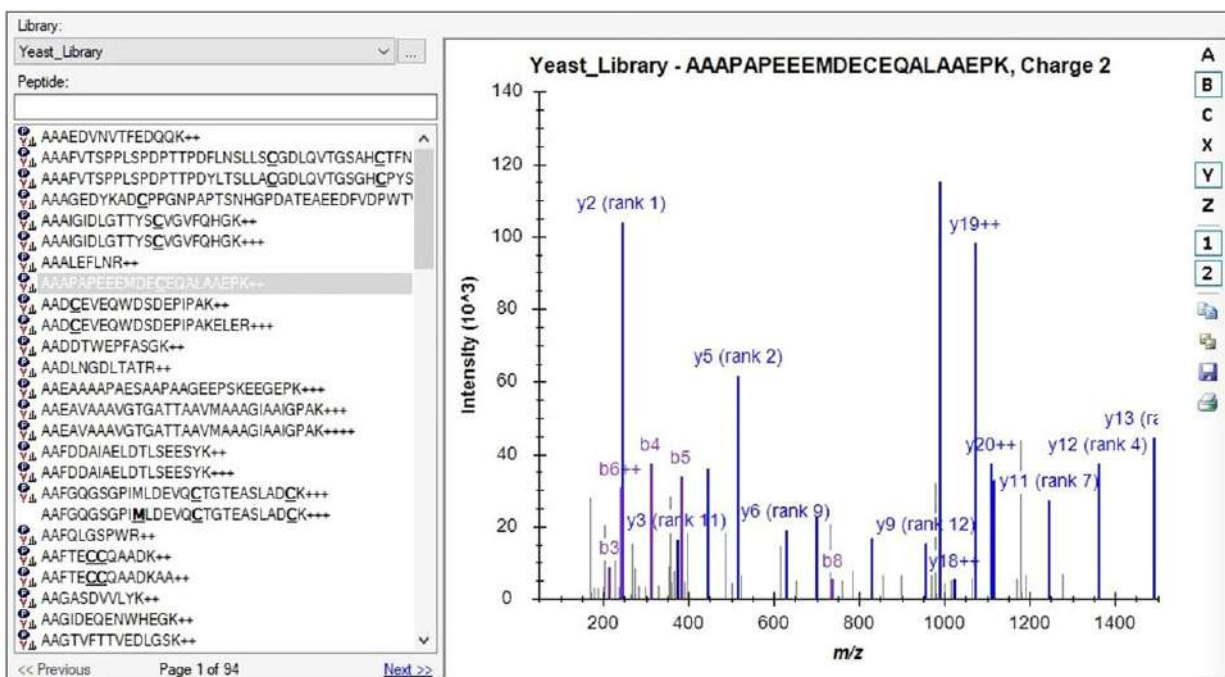


Fig. 5 Spectral library visualization in Skyline.

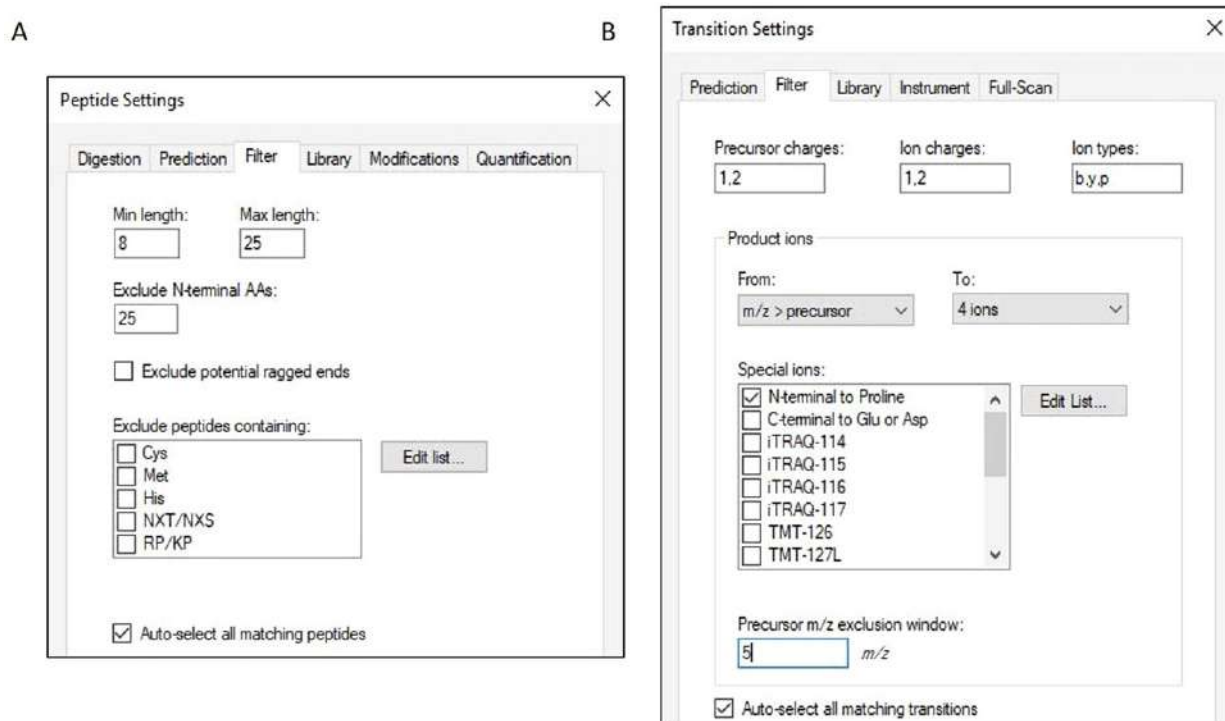


Fig. 6 Filter settings for transition selection in Skyline. (A) Peptide settings. (B) Transition settings.

modifications can also be added to the Skyline. In 'Filter' module, peptides length, precursor/product ion charges and types, and a number of ions can be determined (see Fig. 6). Subsequently, through 'Library' settings, ion match tolerance values (match between DIA and DDA data) and library to be added in the analysis can be set. It also provides options for selecting most intense transitions from the library, which also fulfils the 'Filter' settings (see Fig. 7). For the current data set, following criteria were set (see Figs. 6 and 7):

- Peptide length = min 8 and max 25
- Structural modifications = no oxidation on methionine

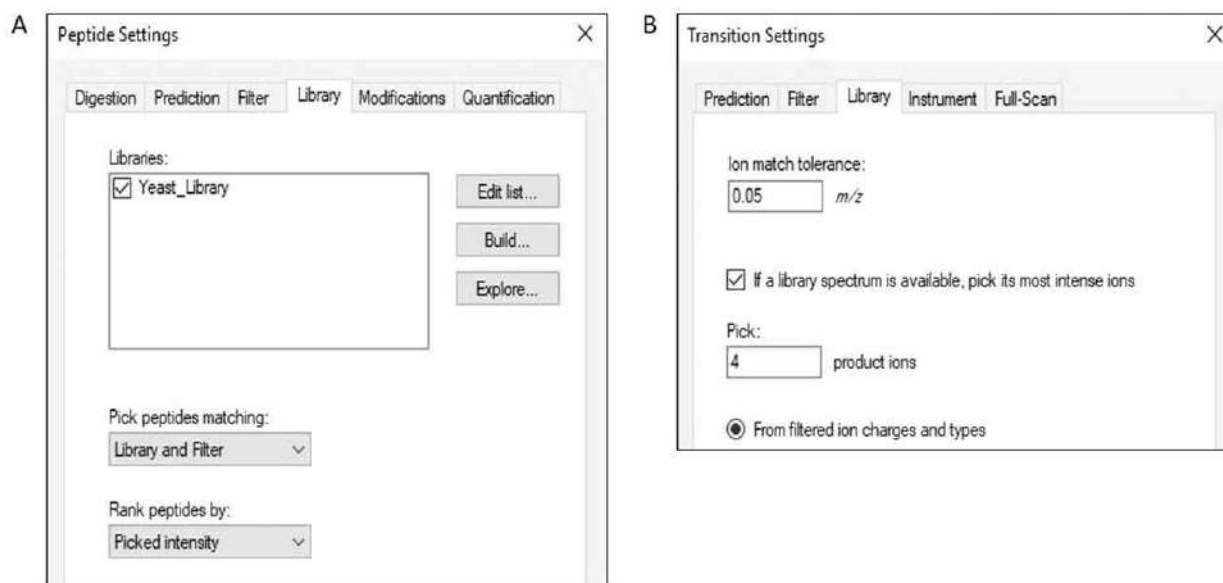


Fig. 7 Library settings for transition selection in Skyline. (A) Peptide settings. (B) Transition settings.

- Ion Intensities=highly intense in spectral library
- Ion Charges (precursor and product ions)=single (1) and double (2)
- Ion types=b, y and p (precursor) ions
- Precursor m/z exclusion window=5 m/z
- Match tolerance=0.05 m/z
- Number of selected transitions=top four

Using 'precursor m/z exclusion window' option, spectra in a mass window around precursor m/z value can be excluded to minimize the noise and extract the refined transitions. In the 'Library' and 'Filter' settings, further transition options are also given such as, 'auto-select all matching transitions', 'pick most intense transitions from library spectrum' and 'pick peptides matching both the library and filter settings'. After setting up the environment for transitions selection, a list of targeted proteins is loaded into the Skyline to generate the fragment or assay library.

Step 3: Targeted Proteins

In contrast to MRM data analysis, where proteins of interest are already determined prior to performing the experiment, in DIA, the list of targeted proteins or peptides are loaded after the experiment has been performed. This allows the extraction and analysis of DIA data that is only related to proteins of interest from the whole data independent run, which usually analyses all the peptides within the range of 400–1200 m/z (depending upon the instrument settings). The targeted proteins/peptides list can be provided in different ways in Skyline, such as, selected externally and imported into the skyline using 'Import Fasta' module, and/or adding all those proteins/peptides which are available in the spectral library. It can also be added by simply inserting the proteins and peptide sequences in 'Targets' module. In this study, peptides which were identified and stored in the spectral library, and external protein data i.e. 'Yeast.fasta' were added to the target list. The proteins present in the fasta file also undergo *in silico* tryptic digestion in Skyline. While the protein data is provided, all those transitions which matched the transition settings (in step 2) were added to the target list along with their abundances and retention time values, as shown in Fig. 8.

Step 4: DIA Data Extraction Environment

Before importing and extracting the DIA data from experimental runs, Skyline needs to set up the DIA environment. In this, all the instrument and isolation settings for MS1 and MS/MS filtering are configured in 'Transition Settings' under 'Full-Scan' module. The full-scan features include:

- Precursor mass analyzer
- Product mass analyzer
- Resolving powers of analyzers

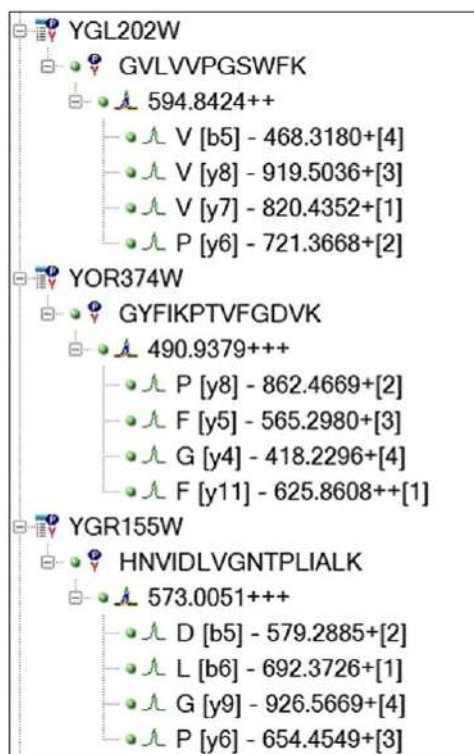


Fig. 8 Visualization of targeted transitions list in Skyline.

- Acquisition method
- Isolation Scheme (pattern of precursor isolation windows in DIA acquisition)
- Retention time filtering

A DIA isolation scheme defines the pattern for each acquisition window (m/z range) through the entire mass range by the specific instrument. Skyline provides default isolation schemes with a variety of patterns, for example, SWATH (15 m/z), SWATH (25 m/z) etc. A customized pattern can also be designed and added according to the particular instrument used for data acquisition experiment. To set up a user-specific isolation scheme, following parameters are required:

- Mass range=400 to 1200 m/z
- Window (isolation) width=25 m/z
- Number of windows=32
- Total cycle time=3.3 s
- Accumulation time (time spent on single transition)=10 ms

For the current data set, these values are filled up according to the 5600 TripleTof mass spectrometer used for yeast proteome quantification, provided in the study (Selevsek *et al.*, 2015). After specifying the settings, Skyline displays the isolation windows in tabular and graphical form.

Retention time filtering defines the time range over which each chromatogram is extracted from DIA data against a single targeted transition. The chromatogram can be extracted over the entire gradient, however, it may result in the noisy and distorted peaks with a minimum understanding of correct chromatogram peak. For precise data extraction and identification, retention time value for each peptide in the experiment is required. Skyline provides couple of options limiting the RT range, such as, (i) using RT values of peptides found in the spectral library, (through DDA runs), (ii) predicting these values using SSRCalc hydrophobicity algorithm (Spicer *et al.*, 2007) and (iii) calculating RT values using standard RT peptides through empirical measurements (also called as normalized RTs or iRTs). Details of the SSRCalc and normalized RTs is available in the Skyline tutorial.

Subsequently, assigning the values to the other parameters in 'Full-Scan' module, set up the DIA and DDA environment in Skyline for data extraction and analysis (see Fig. 9).

- Precursor mass analyzer=TOF
- Product mass analyzer=TOF
- Resolving powers of analyzers=30,000
- Acquisition method=DIA

The screenshot shows the 'Transition Settings' dialog box in Skyline. It has five tabs: Prediction, Filter, Library, Instrument, and Full-Scan. The 'Filter' tab is selected. Inside the 'Filter' tab, there are two main sections: 'MS1 filtering' and 'MS/MS filtering'.
 In the 'MS1 filtering' section:
 - 'Isotope peaks included:' is set to 'Count'.
 - 'Precursor mass analyzer:' is set to 'TOF'.
 - 'Peaks:' is set to '3'.
 - 'Resolving power:' is set to '30,000'.
 - 'Isotope labeling enrichment:' is set to 'Default'.
 In the 'MS/MS filtering' section:
 - 'Acquisition method:' is set to 'DIA'.
 - 'Product mass analyzer:' is set to 'TOF'.
 - 'Isolation scheme:' is set to 'Yeast_Scheme'.
 - 'Resolving power:' is set to '30,000'.
 Below these sections, there is a checkbox 'Use high-selectivity extraction' which is unchecked.
 At the bottom, there is a 'Retention time filtering' section with three radio button options:
 - 'Use only scans within 5 minutes of MS/MS IDs' (selected).
 - 'Use only scans within 5 minutes of predicted RT'.
 - 'Include all matching scans'.
 At the very bottom are 'OK' and 'Cancel' buttons.

Fig. 9 MS1 and MS/MS filtering settings for DIA data extraction in Skyline.

- Isolation Scheme (pattern of precursor isolation windows in DIA acquisition)=Yeast_Scheme (user-defined)
- Retention time filtering=scans within 5 min of MS/MS IDs

This retention time filtering value will extract the chromatograms from DIA spectra within 5 min range of the peptide-spectrum match for each targeted peptide from the DDA run in a spectral library.

Step 5: DIA Targeted Data Extraction

Following setting up the environment in Skyline, data files from SWATH-MS runs for yeast biological replicates can be imported. In this study, changes in the *S. cerevisiae* proteome were quantified over 6-time points i.e. 0 min (T0), 15 min (T1), 30 min (T2), 60 min (T3), 90 min (T4) and 120 min (T5), as a result of osmolarity stress. Samples were collected, digested and analyzed using SWATH-MS in triplicates at each time point. The 18 data files are available on ProteomeXchange database (see “Relevant Websites section”) in wiff format with following names:

1. 'L120412_001_SW.wiff' to 'L120412_003_SW.wiff' (T0) (3 files);
2. 'L120412_004_SW.wiff' to 'L120412_006_SW.wiff' (T1) (3 files);
3. 'L120412_007_SW.wiff' to 'L120412_009_SW.wiff' (T2) (3 files);
4. 'L120412_010_SW.wiff' to 'L120412_012_SW.wiff' (T3) (3 files);
5. 'L120412_013_SW.wiff' to 'L120412_015_SW.wiff' (T4) (3 files); and
6. 'L120412_016_SW.wiff' to 'L120412_018_SW.wiff' (T5) (3 files).

These files can be imported directly into Skyline (compatible with different file formats) using 'Import Results' option in 'File' menu. 'Import Results' provides multiple preferences including (i) add single-injection replicates in files, (ii) add multi-injection replicates in directories, (iii) add one new replicate for different types of results (see Fig. 10). As the SWATH runs were added to the Skyline for analysis, it extracted the chromatograms against peptides available in the 'Targets' list while taking into account the retention time filtering parameter. This is also called as 'Targeted Data Extraction'.

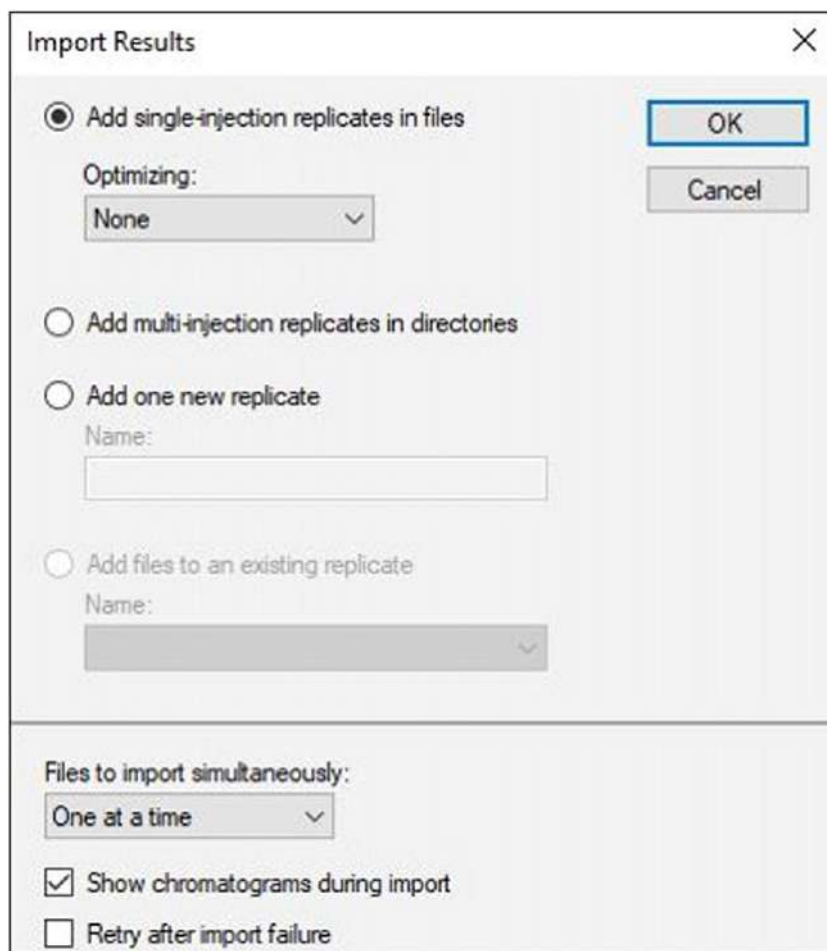


Fig. 10 Import results module for importing mass spectrometry results in Skyline.

Peak picking and scoring

The extracted chromatograms then undergo the process of identifying the correct peak for each targeted peptide, which correlates best with the spectrum in the spectral library. Although this is done automatically, peaks can also be picked manually. For each match, matching scores are allocated, also called as 'dot products'. In order to select the best peak, in addition to its conventional heuristic method, Skyline has also incorporated mProphet algorithm (Reiter *et al.*, 2011) scoring strategy for measuring the similarity between chromatogram and library peaks to choose the best match. The mProphet algorithm takes into account various parameters for scoring purposes, such as:

- Co-elution
- Mass accuracy
- Peak shape
- Ion intensity
- Relative product ion abundance correlation
- Predicted retention time

Additionally, Skyline is equipped with the facility of training and customizing the mProphet peak picking and scoring model for peak determination. This can be executed by generating decoy peptides and calculating FDRs for peaks selection. Details of manual peak-spectrum matching are available in further readings.

Step 6: DIA Data Exploration and Analysis

After importing the DIA data and performing peak selection, chromatograms for 18 biological replicates can be visualized, simultaneously. This can be done by arranging result files systematically into three groups in tabular form using 'Arrange Graphs' module of the Skyline in 'View' menu. Each group indicates the data for all six time points in each replicate. Chromatograms for the first replicate at time T0 to T5 are displayed in Fig. 11. In each graph, best peaks are displayed within the black dotted lines

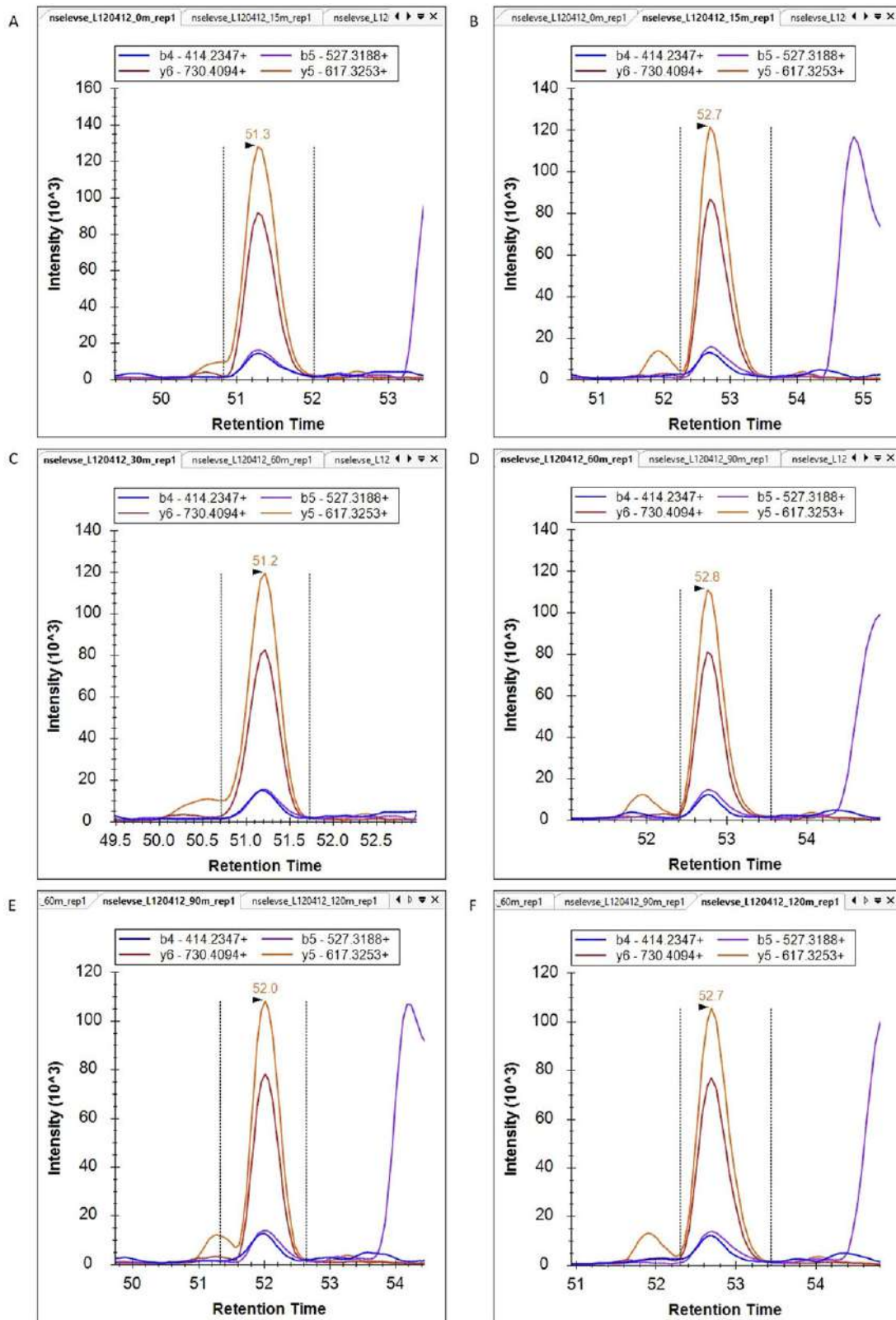


Fig. 11 Chromatograms of yeast peptide from DIA data in replicate 1 at six time points. (A) 0 min (T0). (B) 15 min (T1). (C) 30 min (T2). (D) 60 min (T3). (E) 90 min (T4). (F) 120 min (T5).

which indicates the peak boundaries. Visualization and graphical attributes of chromatograms can be customized using a variety of options in 'View' menu, such as:

- Retention times (all or best peak)=Best peak
- Peptide ID times (none or matching with library)=Matching
- Transitions (all, precursor or products)=All
- Transform (interpolated or Savitsky-Golay smoothing)=Savitzky-Golay Smoothing
- Auto-zoom X-axis=Best peak
- Auto-scale Y-axis

Peptides selection for quantification

From the given list of targeted proteins, a number of peptides or peaks are matched and identified (from DIA and DDA data). For the downstream quantification measurements, those peptides should be incorporated in a way that shows consistency in their peak areas and retention times in multiple replicates. So, in order to perform this refinement, peak intensities and RTs of all the peptides can be compared and inspected using Skyline. In 'View' menu, 'Replicate Comparison' option for 'Retention Times' and 'Peak Areas' can be used to analyze the intensity and time comparisons in all 18 DIA replicate runs for yeast proteome at six time points.

In retention time replicate view (see Fig. 12(A)), following graph parameters were set:

- Value (all or retention time)=All
- Transitions (all, precursor or product ions)=All
- Order (acquired time or document)=Document

The RT of each fragment ion is represented in separate color and collectively each set of bars indicates the RT pattern of the whole peptide, over 18 runs. The start and end retention time points depict the elution times of the peptide in triplicates. The same height of all the groups shows that the peptide elutes in approximately equal time at all six time points.

In peak intensity replicate view (see Fig. 12(B)), following graph parameters were set:

- Normalized to (total, maximize or none)=None
- Transitions (all, precursor or product ions)=All
- Order (acquired time or document)=Document

The intensity of each fragment ion is illustrated in different colors and collectively each bar indicates the peak intensity of the whole peptide, over 18 runs. From the graph, it can be demonstrated that intensity values get lower after 15 min of exposure to osmotic stress in replicate 1 and 3, whereas, a slight increase in intensity is observed at 90 min and 120 min of exposure. However, in replicate 2, the intensity values during first 60 min are approximately the same and get decreased at 90 and 120 min. Moreover, relative transition intensity of peptides in all 18 runs can be visualized by normalizing the values to 'Total' in 'Peak Area' option in the replicate graph.

Consequently, similar peak intensity and retention time pattern among all the replicates validate the reproducibility of the MS instrument and experiment. It also depicts that the Skyline has identified and picked the correct peak for that particular peptide. Similarly, all the peptides can be inspected through these graphs individually from the target list, to select for quantification purposes.

If the intensity and retention time data for a peptide is not consistent among replicates (see Fig. 13), it can be improved by manual selection of correct peaks. It can be observed in Fig. 14 that for this particular peptide, unlike replicate 3, the correct peak is not selected in replicate 1 and replicate 2. This can be resolved by moving the peak boundaries via dragging the black dotted lines

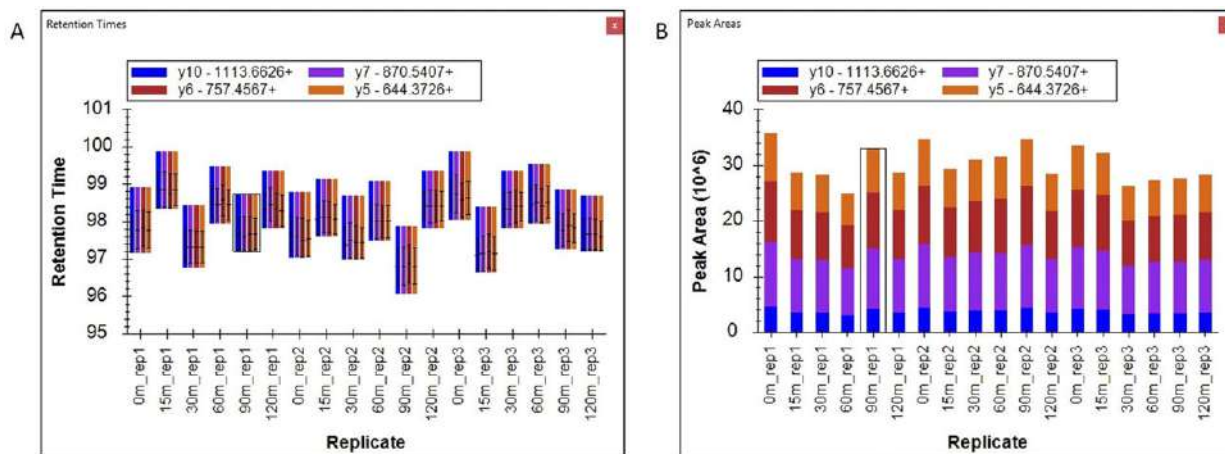


Fig. 12 Consistent replicate comparison graphs in Skyline. (A) Retention times (B) Peak areas.

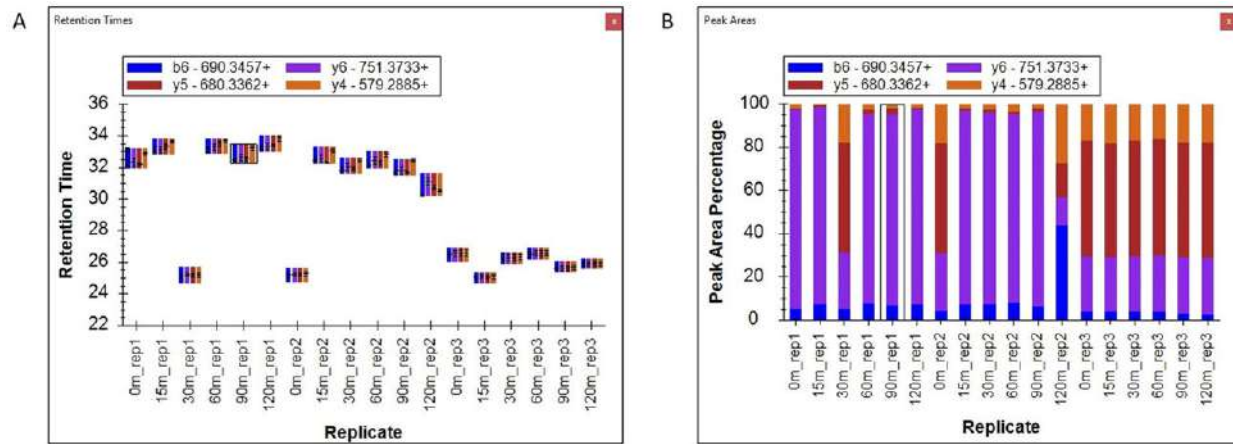


Fig. 13 Inconsistent replicate comparison graphs in Skyline. (A) Retention times (B) Peak areas.

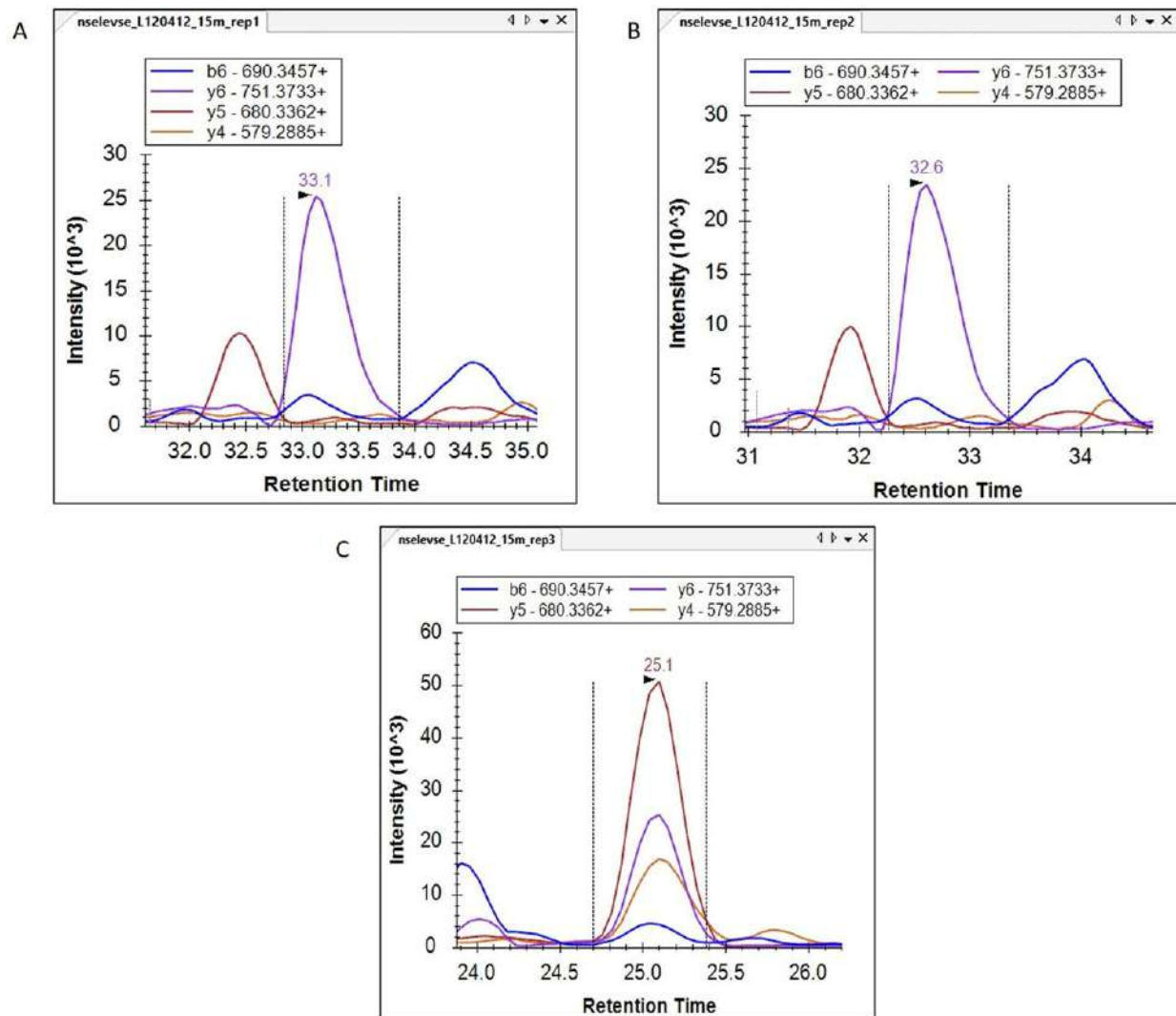


Fig. 14 Chromatograms of selected peaks at 15 min in Skyline. Incorrect peaks are selected in (A) Replicate 1 and (B) Replicate 2. (C) Peak in Replicate 3 is correctly identified.

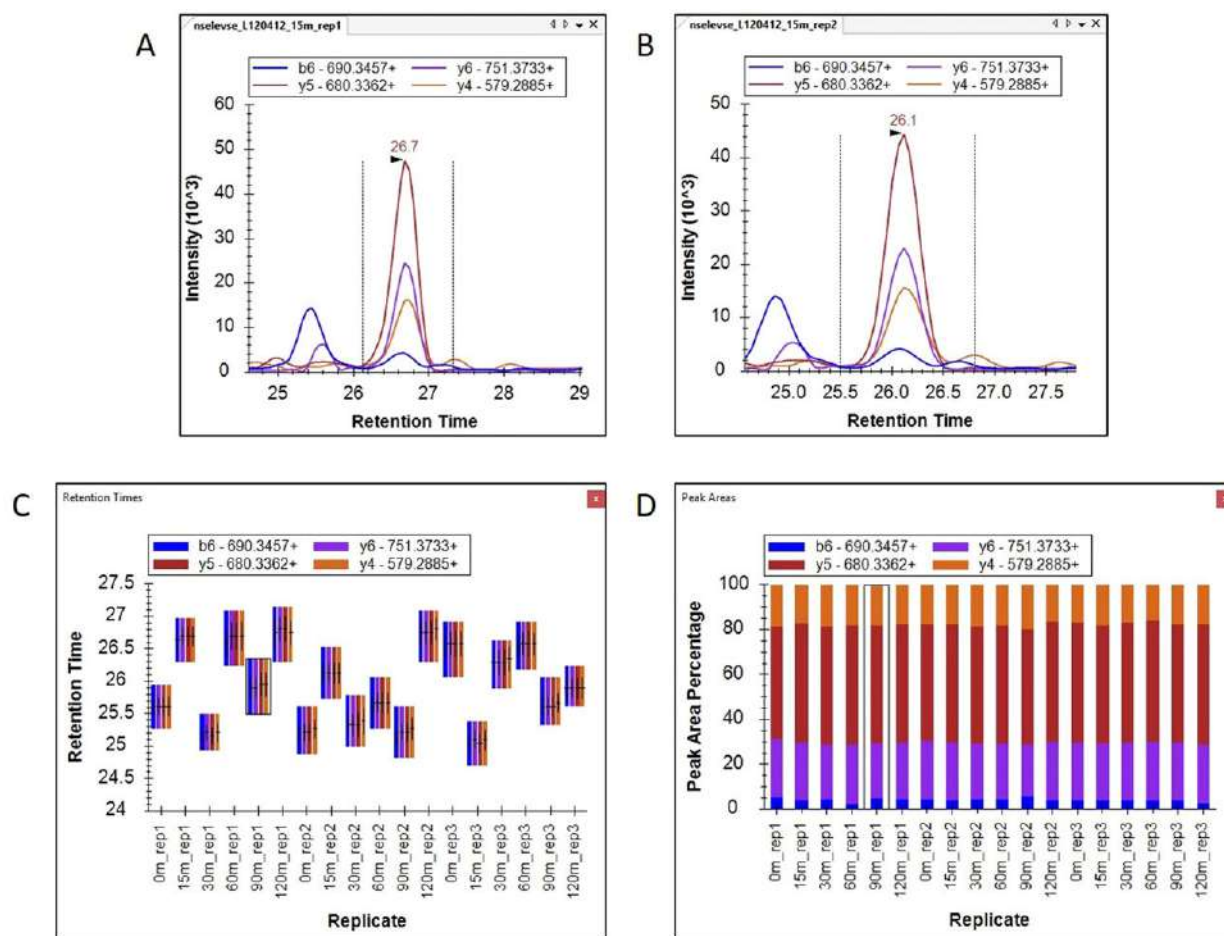


Fig. 15 Chromatograms of correctly selected peaks for specific peptide at 15 min. (A) Replicate 1. (B) Replicate 2. (C) Retention times consistency. (D) Peak areas consistency.

around the peaks. These corrections can now be observed in peak area and retention time graphs as well (see Fig. 15). If the data does not show consistency even after adjustments, the insignificant peptide can be deleted from the study.

Step 7: Statistical Analysis for Quantification

The aim of this study was to quantify the abundance of yeast proteome in response to the osmotic stress condition and identify those proteins which show a significant change over the course of the time period of two hours. Skyline provides the facility of performing this statistical analysis by annotating the replicate data with some additional classes or conditions. In this case study, conditions are the different time points, along which we have to observe the changes. Data annotation can be performed using 'Define Annotation' module in the 'Document Settings' in Skyline. Condition values are named as 'a0m', 'b15m', 'c30m', 'd60m', 'e90m' and 'f120m' and applied onto 'Replicates' (see Fig. 16). Select 'Condition' and 'BioReplicate' in 'Document Settings' to incorporate these annotations into the current Skyline document to analyze yeast data.

Intensity and time variation among technical replicates

Using replicate comparison function of Skyline, variation among the intensities and retention times of the quantotypic peptides in different conditions can be analyzed. In peak areas and retention time views, following graph parameters were set:

- Transitions (all, precursor, products or total)=Total
- Group by (replicate or condition)=Condition
- Normalized to (maximum, total or none)=Maximize
- Select CV values

Graphs shown in Fig. 17 illustrate the variation among different conditions or time points for this particular peptide belonging to protein 'YGR240C', in all replicates. All the peptides can be analyzed by iterating over the entire set of targeted proteins. It can be deduced from the graph (see Fig. 17) that intensity of this specific peptide greatly varies among all replicates at 90 min after the exposure to osmotic stress environment and the least variation is observed at 0 min and 120 min.

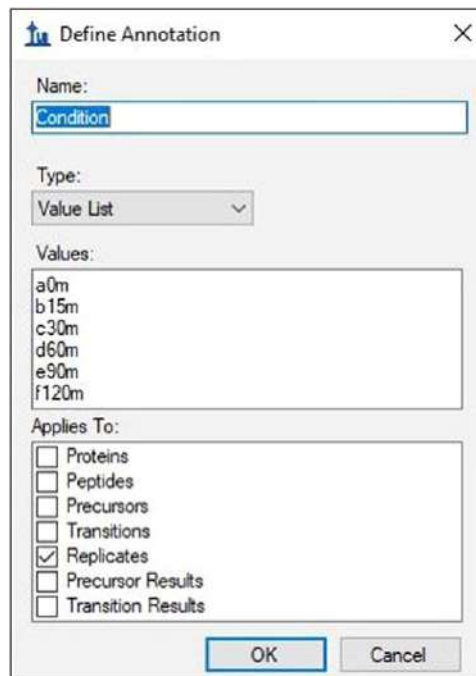


Fig. 16 Annotation module for defining new annotations for replicate data in Skyline.

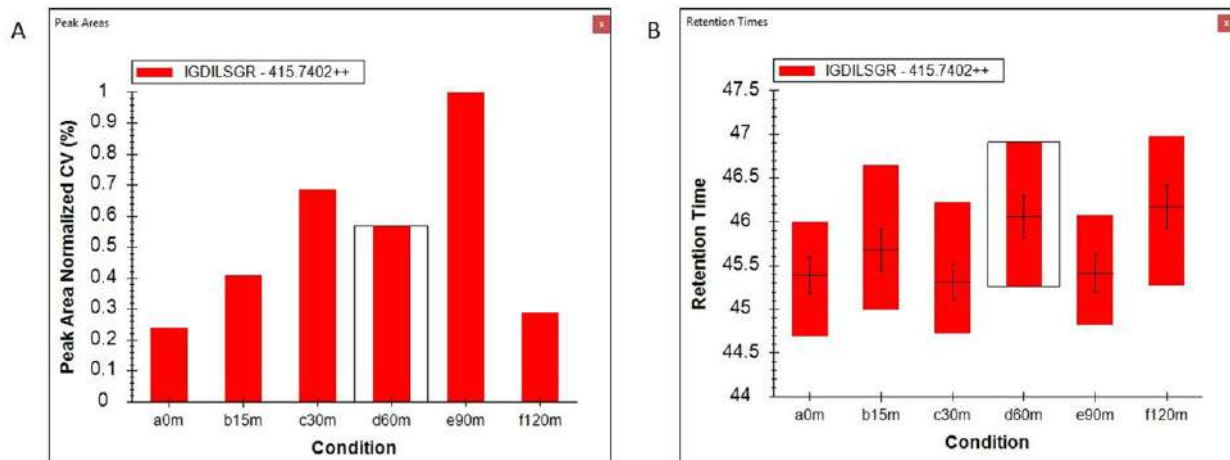


Fig. 17 Variation among replicates at six time points. (A) Intensity. (B) Time.

Average peptide abundance among conditions

To study the average peptide abundance or difference in peptide expression among different conditions, following graph parameters were set in peak areas replicate comparison view:

- Transitions (all, precursor, products or total)=Total
- Group by (replicate or condition)=Condition
- Normalized to (maximum, total or none)=Total
- Deselect CV values

The graph in [Fig. 18](#) presents the average mean of peptide abundances in triplicates among all conditions. The red bars in the graph represent the mean average values and black lines on the bars represent the standard deviation, which measures the amount of dispersion in the intensity values. A high standard deviation at 90 min depicts the high variation in data values at this time in all replicates.

Overlapping mean values for this specific peptide at six time points, shown in the [Figs. 17](#) and [18](#), do not contribute greatly to the differential abundance analysis, and therefore, demonstrate that protein 'YGR240C' exhibits no significant response to osmotic stress. Contrary to this, peptide response from protein 'YMR169C', shown in [Fig. 19](#), is highly predictive and presents a supposition

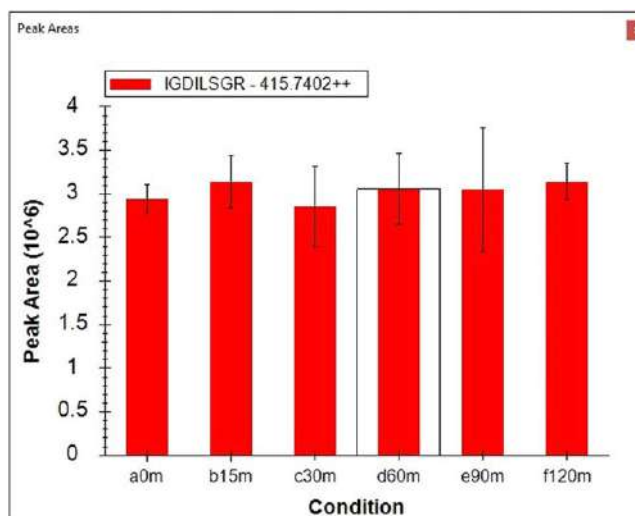


Fig. 18 Average peptide abundances among replicates at six time points.

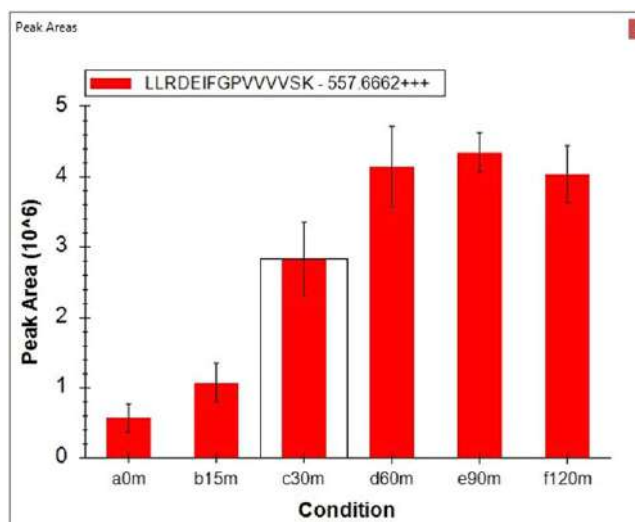


Fig. 19 Significant change in abundances during six time points after osmotic shock.

that this protein may have been upregulated under the influence of osmotic stress. However, this hypothesis requires the similar behaviour from more than one peptide to validate the up or down-regulation of the corresponding protein.

Group comparison among conditions

Differential abundance analysis between time points can also be performed through pairwise group comparison investigation i.e. measure the change in abundance between time points:

- 0 min (T0) to 15 min (T1)
- 0 min (T0) to 30 min (T2)
- 0 min (T0) to 60 min (T3)
- 0 min (T0) to 90 min (T4)
- 0 min (T0) to 120 min (T5)

In Skyline, 'Group Comparison' module in 'Document Settings' provides the facility of adding as many comparisons as required. In 'Edit Group Comparison' form, for the first comparison, following parameters were set:

- Name=T0 v. T1
- Control group annotation=Condition
- Control group value=a0m

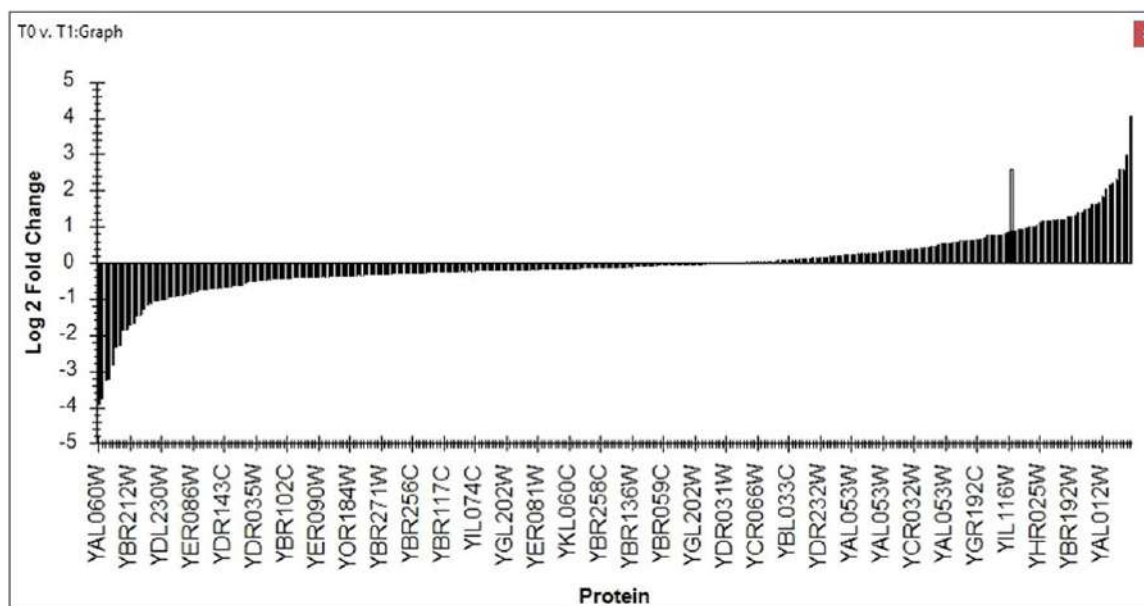


Fig. 20 Log2 fold change in abundances of proteins during the time period between 0 min (T0) and 15 min (T1).

- Value to compare against=b15m
- Identity annotation=BioReplicate
- Normalization method=Equalize medians
- Confidence level=95%
- Scope=Peptide or Protein for peptide or protein level analysis, respectively

To inspect the abundances of proteins within the time frame of first 15 min, the graph can be visualized in 'View' and 'Group Comparison' menu of Skyline. The graph shown in [Fig. 20](#) illustrates the overall fold change, either increase or decrease or no effect, in yeast proteins. By employing the same method on all other comparison groups, protein expressions during different time periods can be analyzed.

Data Visualization

Different quantification strategies provide a measure of change in proteome between different biological conditions/samples. Elucidation of biological relevancy of these quantifications requires additional analysis. Perseus is widely adopted for analysis post-database search, for visualization and statistical analysis. There are various tools available for pathway enrichment and analysis, Gene ontology (GO) enrichment could be performed by Panther ([Mi et al., 2016](#)), protein-protein interaction information could be obtained from STRING ([Szklarczyk et al., 2015](#)), IntAct ([Orchard et al., 2014](#)), Human Protein Reference Database (HPRD) ([Keshava Prasad et al., 2009](#)) and Biological General Repository for Interaction Datasets (BioGRID) ([Stark et al., 2011](#)) amongst others.

Perseus ([Tyanova et al., 2016](#)) is a freely available software that performs shotgun proteomics analysis of biological relevance from processed raw files. In addition, can perform various visualization, clustering, principal component analysis (PCA) and statistical tests on quantitative and time series data, details of which are available on Perseus documentation.

Conclusion

Mass spectrometry based quantitative proteomics has rapidly changed the landscape of proteomics research spurred on by rapid technological advances and more importantly huge strides ahead in computational analysis and informatics. It has to be noted though, that the major limitation of quantitative studies is the thresholds and stringencies set by the users. Although there is yet to be standards developed for quantitative proteomics ([Mohamedali et al., 2017](#)), it behoves researchers and users to be wary of and thoroughly statistically analyze all mass spectrometry data keeping strict stringency to eliminate false positives. The example used in this article illustrates that the procedure for the analysis for protein quantification by proteomics is relatively straightforward with freely available tools. It has to be emphasised though, that users have a good understanding of the theoretical underpinnings of Mass Spectrometry and informatics to ensure that the results obtained from their studies are robust and reproducible.

Furthermore, upon the identification of differentially expressed proteins between samples after a rigorous experimental and analytical approach, orthogonal techniques to validate potential candidates using targeted techniques such as MRM's and ELISA are almost a compulsory part of the reliability of MS quantification ([Vidova and Spacil, 2017](#)). The use therefore, of large scale quantitative proteomics using a label-free DIA approach has been a mainstay of the discovery part of proteomic studies and indeed

has proven to be a formidable, robust and reliable method of quantifying large numbers of proteins in complex biological samples. Certain limitations such as the size and nature of libraries, quality of sample preparation, reproducibility on varied instruments, quality of data, statistical threshold guidelines etc. all still mean that quantitative proteomics has a way to go to achieve the robustness and reliability of quantitative transcriptomics.

See also: Clinical Proteomics. Genome-Wide Scanning of Gene Expression. Identification of Proteins from Proteomic Analysis. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis

References

- Ahrne, E., Molzahn, L., Glatter, T., Schmidt, A., 2013. Critical assessment of proteome-wide label-free absolute abundance estimation strategies. *Proteomics* 13 (17), 2567–2578.
- Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., Kuster, B., 2007. Quantitative mass spectrometry in proteomics: A critical review. *Anal. Bioanal. Chem.* 389 (4), 1017–1031.
- Bruderer, R., Sondermann, J., Tsou, C.C., *et al.*, 2017. New targeted approaches for the quantification of data-independent acquisition mass spectrometry. *Proteomics* 17 (9), 1797–1808.
- Cherry, J.M., Hong, E.L., Amundsen, C., *et al.*, 2012. Saccharomyces genome database: The genomics resource of budding yeast. *Nucleic Acids Res.* 40 (D1), D700–D705.
- Choi, M., Chang, C.Y., Clough, T., *et al.*, 2014. MSstats: An R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments. *Bioinformatics* 30 (17), 2524–2526.
- Deutsch, E.W., Mendoza, L., Shteynberg, D., *et al.*, 2015. Trans-Proteomic Pipeline, a standardized data processing pipeline for large-scale reproducible proteomics informatics. *Proteomics Clin. Appl.* 9 (7–8), 745–754.
- Distler, U., Kuharev, J., Navarro, P., Tenzer, S., 2016. Label-free quantification in ion mobility-enhanced data-independent acquisition proteomics. *Nat. Protoc.* 11 (4), 795–812.
- Fan, X.G., Zheng, Z.Q., 1997. A study of early pregnancy factor activity in preimplantation. *Am. J. Reprod. Immunol.* 37 (5), 359–364.
- Gillette, M.A., Carr, S.A., 2013. Quantitative analysis of peptides and proteins in biomedicine by targeted mass spectrometry. *Nat. Methods* 10 (1), 28–34.
- Keshava Prasad, T.S., Goel, R., Kandasamy, K., *et al.*, 2009. Human protein reference database – 2009 update. *Nucleic Acids Res.* 37 (Database issue), D767–D772.
- Kessner, D., Chambers, M., Burke, R., Agusand, D., Mallick, P., 2008. ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* 24 (21), 2534–2536.
- King, C.D., Dudenhoefter, J.D., Gu, L., Evans, A.R., Robinson, R.A.S., 2017. Enhanced sample multiplexing of tissues using combined precursor isotopic labeling and isobaric tagging (cPILOT). *J. Vis. Exp.* 123.
- Lam, H., Deutsch, E.W., Eddes, J.S., *et al.*, 2007. Development and validation of a spectral library searching method for peptide identification from MS/MS. *Proteomics* 7 (5), 655–667.
- Law, K.P., Lim, Y.P., 2013. Recent advances in mass spectrometry: Data independent analysis and hyper reaction monitoring. *Expert Rev. Proteomics* 10 (6), 551–566.
- Lindemann, C., Thomanek, N., Hundt, F., *et al.*, 2017. Strategies in relative and absolute quantitative mass spectrometry based proteomics. *Biol. Chem.* 398 (5–6), 687–699.
- Mi, H., Poudel, S., Muruganujan, A., Casagrande, J.T., Thomas, P.D., 2016. PANTHER version 10: Expanded protein families and functions, and analysis tools. *Nucleic Acids Res.* 44 (D1), D336–D342.
- Mohamedali, A., Ahn, S.B., Sreenivasan, V.K.A., Ranganathan, S., Baker, M.S., 2017. Human Prestin: A candidate PE1 protein lacking stringent mass spectrometric evidence? *J. Proteome Res.* 16 (12), 4531–4535.
- Mueller, L.N., Brusniak, M.Y., Mani, D.R., Aebersold, R., 2008. An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data. *J. Proteome Res.* 7 (1), 51–61.
- Neilson, K.A., Ali, N.A., Muralidharan, S., *et al.*, 2011. Less label, more free: Approaches in label-free quantitative mass spectrometry. *Proteomics* 11 (4), 535–553.
- Norbeck, A.D., Monroe, M.E., Adkins, J.N., *et al.*, 2005. The utility of accurate mass and LC elution time information in the analysis of complex proteomes. *J. Am. Soc. Mass Spectrom.* 16 (8), 1239–1249.
- Orchard, S., Ammari, M., Aranda, B., *et al.*, 2014. The MINTAct project – IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* 42 (Database issue), D358–D363.
- Pino, L.K., Searle, B.C., Bollinger, J.G., *et al.*, 2017. The Skyline ecosystem: Informatics for quantitative mass spectrometry proteomics. *Mass Spectrom. Rev.*
- Reiter, L., Rinner, O., Picotti, P., *et al.*, 2011. mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nat. Methods* 8 (5), 430–435.
- Rost, H.L., Rosenberger, G., Navarro, P., *et al.*, 2014. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nat. Biotechnol.* 32 (3), 219–223.
- Selevsek, N., Chang, C.Y., Gillet, L.C., *et al.*, 2015. Reproducible and consistent quantification of the Saccharomyces cerevisiae proteome by SWATH-mass spectrometry. *Mol. Cell. Proteomics* 14 (3), 739–749.
- Spicer, V., Yamchuk, A., Cortes, J., *et al.*, 2007. Sequence-specific retention calculator. A family of peptide retention time prediction algorithms in reversed-phase HPLC: Applicability to various chromatographic conditions and columns. *Anal. Chem.* 79 (22), 8762–8768.
- Stark, C., Breitkreutz, B.J., Chatr-Aryamontri, A., *et al.*, 2011. The BioGRID interaction database: 2011 update. *Nucleic Acids Res.* 39 (Database issue), D698–D704.
- Szklarczyk, D., Franceschini, A., Wyder, S., *et al.*, 2015. STRING v10: Protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* 43 (Database issue), D447–D452.
- Tang, H., Fang, H., Yin, E., *et al.*, 2014. Multiplexed parallel reaction monitoring targeting histone modifications on the QExactive mass spectrometer. *Anal. Chem.* 86 (11), 5526–5534.
- Tsou, C.C., Avtonomov, D., Larsen, B., *et al.*, 2015. DIA-Umpire: Comprehensive computational framework for data-independent acquisition proteomics. *Nat. Methods* 12 (3), 258–264.
- Tyanova, S., Temu, T., Sinitcyn, P., *et al.*, 2016. The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nat. Methods* 13 (9), 731–740.
- Vidova, V., Spacil, Z., 2017. A review on mass spectrometry-based quantitative proteomics: Targeted and data independent acquisition. *Anal. Chim. Acta* 964, 7–23.
- Vogel, C., Marcotte, E.M., 2012. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nat. Rev. Genet.* 13 (4), 227–232.
- Washburn, M.P., Wolters, D., Yates 3rd, J.R., 2001. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nat. Biotechnol.* 19 (3), 242–247.
- Wenner, B.R., Lynn, B.C., 2004. Factors that affect ion trap data-dependent MS/MS in proteomics. *J. Am. Soc. Mass Spectrom.* 15 (2), 150–157.

Further Readings

- Anand, S., Samuel, M., Ang, C.S., Keerthikumar, S., Mathivanan, S., 2017. Label-based and label-free strategies for protein quantitation. *Proteome Bioinform.* 31–43.
- Anderle, M., Roy, S., Lin, H., Becker, C., Joho, K., 2004. Quantifying reproducibility for differential proteomics: Noise analysis for protein liquid chromatography-mass spectrometry of human serum. *Bioinformatics* 20 (18), 3575–3582.

- Bantscheff, M., Lemeer, S., Savitski, M.M., Kuster, B., 2012. Quantitative mass spectrometry in proteomics: Critical review update from 2007 to the present. *Anal. Bioanal. Chem.* 404 (4), 939–965.
- Chen, Y., Wang, F., Xu, F., Yang, T., 2016. Mass spectrometry-based protein quantification. In: Mirzaei, H., Carrasco, M. (Eds.), *Modern Proteomics—Sample Preparation, Analysis and Practical Applications*. Springer International Publishing, pp. 255–279.
- Clough, T., Thaminy, S., Ragg, S., Aebersold, R., Vitek, O., 2012. Statistical protein quantification and significance analysis in label-free LC-MS experiments with complex designs. *BMC Bioinform.* 13 (16), S6.
- Karpievitch, Y., Stanley, J., Taverner, T., *et al.*, 2009. A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics* 25 (16), 2028–2034.
- Lill, J., 2003. Proteomic tools for quantitation by mass spectrometry. *Mass Spectrom. Rev.* 22 (3), 182–194.
- Pan, S., Aebersold, R., Chen, R., *et al.*, 2008. Mass spectrometry based targeted protein quantification: Methods and applications. *J. Proteome Res.* 8 (2), 787–797.
- Wang, M., You, J., Bemis, K.G., Tegeler, T.J., Brown, D.P., 2008. Label-free mass spectrometry-based protein quantification technologies in proteomic analysis. *Brief. Funct. Genom. Proteom.* 7 (5), 329–339.
- Wilm, M., 2009. Quantitative proteomics in biological research. *Proteomics* 9 (20), 4590–4605.

Relevant Websites

<http://www.coxdocs.org/doku.php?id=maxquant:start>
MaxQuant.

<http://www.openms.de/tutorials/>
OpenSWATH.

<http://www.coxdocs.org/doku.php?id=perseus:start>
Perseus.

<http://www.ebi.ac.uk/pride/archive/projects/PXD001010>
PRIDE Archive - PXD001010.

<https://skyline.ms/wiki/home/software/Skyline/page.view?name=tutorials>
Skyline.

Biographical Sketch



Zainab is a PhD student in the department of Chemistry and Biomolecular sciences (Bioinformatics group) at Macquarie University. She has been working in the area of protein structure and function analysis during her bachelors and masters' studies. Previously, she worked on identifying novel potent drug-like compounds against histone deacetylases proteins by employing *in silico* drug designing techniques. Currently, she is involved in programming-based mass spectrometry data analysis of data generated through data-dependent acquisition (DDA) and data-independent acquisition (DIA) approaches related to colorectal cancer. More specifically, she is dealing with the customization of DDA based spectral libraries to be used in DIA data analysis for biomarker discovery.



Subash is a PhD student in Cancer proteomics. His main interest is on mass spectrometry based quantitative proteomics. He is currently working on peptide antagonist against specific membrane protein-protein interaction that are known to drive epithelial cancer metastasis.



Shoba Ranganathan holds a Chair in Bioinformatics at Macquarie University since 2004. She has held research and academic positions in India, USA, Singapore and Australia as well as a consultancy in industry. Shoba's research addresses several key areas of bioinformatics to understand biological systems using computational approaches. Her group has achieved both experience and expertise in different aspects of computational biology, ranging from metabolites and small molecules to biochemical networks, pathway analysis and computational systems biology. She has authored as well as edited several books as well as contributed several articles to Springer's Encyclopedia of Systems Biology. She is currently the Editor-in-Chief of Elsevier's Encyclopedia of Bioinformatics and Computational Biology as well as the Bioinformatics Section Editor for Elsevier's Reference Module in Life Sciences.



Abidali is a Lecturer at Macquarie University in the Faculty of Science and Engineering. He completed his PhD in 2010 studying, using proteomics, the effects of single mutations in mouse model on brain development in the autism spectrum disorder, Rett syndrome. He then returned to Macquarie University undertaking a post-doc to study the biology of metastasis in colorectal cancer using quantitative proteomics and to determine novel therapeutic targets and develop novel technologies to examine the plasma proteome.

Utilising IPG-IEF to Identify Differentially-Expressed Proteins

David I Cantor, Macquarie University, Sydney, NSW, Australia

Harish R Cheruku, Preston, VIC, Australia

© 2019 Elsevier Inc. All rights reserved.

Abbreviations

2D-DIGE	Two-dimensional differential in-gel electrophoresis	MRM	Multiple reaction monitoring
CRC	Colorectal cancer	MS	Mass spectrometry
HiRIEF	High-resolution isoelectric focusing	pI	Isoelectric point
HPLC	High performance liquid chromatography	SDS-PAGE	Sodium dodecyl sulfate polyacrylamide gel electrophoresis
IPG-IEF	Immobilized pH gradient isoelectric focussing	SILAC	Stable isotope labelling with amino acids in cell culture
iTRAQ	Isobaric tag for relative and absolute quantitation	SRM	Single reaction monitoring
MALDI	Matrix assisted laser desorption/ionisation	SWATH	Sequential window acquisition of all theoretical mass spectra
MARS	Multiple affinity removal system	TMT	Tandem mass tags
		TOF	Time of Flight

Introduction

Living cells and tissues exist in a state of constant flux, responding to, and acting upon, a complex matrix of biochemical signals and genetic programming in order to maintain and perpetuate life. As such, cells and tissues can be considered fluidic in terms of stimulus and response rather than remaining static. Though DNA encodes the foundational 'blueprint' for life, it is the shifting creation or destruction of its downstream effectors (RNAs, proteins and their modifications) that carry out the functionality of the ~20,300 protein-coding genes. The study of these shifts in the protein population as a result of disease, cancer, lifestyle, aging or response to treatment is known as proteomic analysis (or proteomics); which refers to the systematic identification and quantitation of the complete complement of proteins (the proteome) within a biological system at a specific point in time (see Relevant Websites section). Therefore, by establishing specific time points or suitable models of comparison, proteomic analysis can be applied to investigations of biological fluids, cells, tissues, organs or organisms as a means to assess the shifts in protein expression that accompany disease states such as cancer or as a means to identify specific markers of disease (biomarkers).

Thus, proteomics is a method for identifying the change in protein populations that correlate with the presence or progression of a disease. Though it can be applied to the tissue and whole-organism level for microbiota, its strength lies in its capacity to identify cellular behaviour, making it a well-adopted method to explore the cellular biology of cancer. One of the most studied yet still poorly understood diseases, cancer originates from a single malignantly-transformed cell in a multi-staged process that confers a growth advantage compared to the surrounding healthy cells (see Relevant Websites section) (Uhlen *et al.*, 2015). The transformation of a 'healthy' somatic cell into a cancerous lesion is not the result of a single mutation but rather the interplay between large numbers of genes with diverse normal functions. More than 500 genes have been implicated in the transformational process. Whilst these genes and their protein products are necessary for normal growth, survival and function, dysregulation of their expression is what drives or enables a cell to escape healthy cell programming and become cancerous. Dysregulation of normal expression can occur as either a result of overexpression, reduced expression (downregulation) or the expression of a defect protein, each of which can contribute to unregulated tumour growth. To grasp the scale of global morbidity, cancer killed 8.2 million people in 2012 as a result of unregulated cell growth that led to eventual metastasis, multiple organ failure and death (Ferlay *et al.*, 2015). A given cancer can be classified as one of several diseases based upon the organ of origin (i.e., where the primary cancer developed), tumour morphology or molecular features. Along with stage at the time of diagnosis, each of these factors form important determinants for the clinical outcome and choice of treatment for the patient.

Proteomic profiling is one of the several means to characterise a cancerous cell or tissue, providing information on the protein population within that sample and revealing a valuable insight into the biology of a given tumour. Proteomic profiling also facilitates detailed comparison between tumours and their corresponding tissue of origin, providing researchers and clinicians with the differential expression of biologically-relevant proteins between individual tumours or between different types of cancer. Proteomic profiling can also be employed to examine differential protein expression between regions of a given tumour, or of tumour cells at different cancer stages. The key task of these investigations is to identify proteins that are functionally relevant to the progression of the disease, such as markers of cancer stage, aggression and indicators of response to therapy. However, one of the primary difficulties faced by these studies is that these biologically-relevant proteins are found in very low abundance (or copy-number) relative to the vast numbers of somatic 'house-keeping' proteins. More specifically, the dynamic range of protein

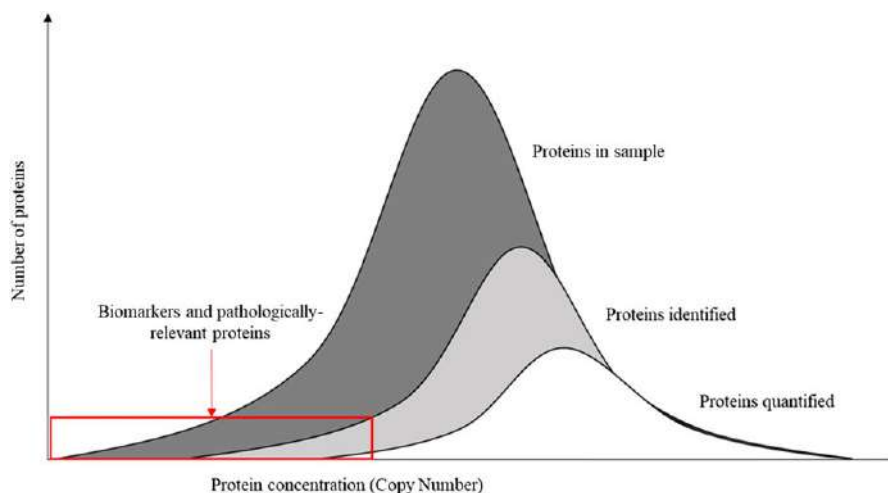


Fig. 1 Schematic representation of proteins identified and quantified within the observable proteome. Cellular proteins span a wide range of copy numbers, with only a fraction of the total being able to be identified, of which only a small percentage can be quantified using currently available molecular biology and mass spectrometry techniques. Biomarkers and pathologically important proteins are often found in very low abundance which makes it challenging to identify and quantify them. Figure modified from Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., Kuster, B., 2007. Quantitative mass spectrometry in proteomics: A critical review. *Anal Bioanal Chem* 389, 1017–1031.

populations in a sample can span several orders of magnitude. The issue of dynamic range between proteins of interest and the somatic population is a constant challenge to the analysis of a range of specimen types such as cell lysates, tissue lysates or plasma. For example, clinical tissue sections contain highly abundant blood-derived proteins that originate from the tissue-embedded network of blood/lymph capillaries and interstitial fluid (Prieto *et al.*, 2014), whilst in biofluids such as plasma, the 22 most-abundant proteins constitute 99% of the total plasma protein mass (Fig. 1) (Prieto *et al.*, 2014; Bantscheff *et al.*, 2007).

As a result, investigations require the use of effective separation techniques to delve deeper into the proteome to identify proteins and peptides that are relevant to a given disease state. The field of proteomics has developed an extensive toolbox of methods that can be applied to such a problem, employing mass spectrometry (MS) as the principal technique for the identification of proteins (and their associated pathways) that are differentially-expressed as a result of carcinogenesis, progression or metastasis. MS enables researchers and clinicians to identify the specific mass and charge state of amino acid sequences which can be matched *in silico* to a peptide sequence and thus identify the parent protein from which the peptide was derived. This process is known as peptide-to-spectra matching. However, the complexity of biological samples can result in the masking of proteins of interest amongst the much more abundant house-keeping proteins. Therefore, reducing sample complexity or enriching for a protein subpopulation of interest is a critical component of the proteomic workflow, which can be achieved by methods such as mass-, charge- or hydrophobicity-based fractionation, subcellular or affinity-based fractionation (Table 1). These methods can be employed either individually or in combination at both the peptide and protein levels.

Once sample complexity has been reduced, multiple proteomic strategies may be used to identify the complement of proteins within. A hierarchical structure of the some commonly used workflows by biomedical researchers is illustrated in Fig. 2. These techniques and workflows are chosen on the basis of the experimental design and include methods such as: label-free and labelled proteomics, gel-based and gel-free proteomics. Each method possesses respective strengths and weaknesses which must be factored into a proposed experimental design.

One method that enables researchers to reduce sample complexity without relying upon mass-based separation is immobilised pH gradient isoelectric focussing (IPG-IEF), which enables separation based on isoelectric point (pI). This method addresses a problem with direct injection of a sample onto an LC-MS/MS system, which is that a single-dimension separation by reversed-phase LC and does not afford a sufficient peak capacity to successfully process all of the tens of thousands of peptide species that are present within an unfractionated tryptic digest of a complex biological sample. More specifically, IPG-IEF employs an immobilised pH gradient within an acrylamide matrix by integrating acrylamide buffer molecules during gel casting. As a result, when placed under a current, the peptides or proteins that are loaded onto the gel strip will migrate along the one-dimensional gel until they reach their respective isoelectric points, where they will cease to migrate. Once the run is complete, the gel strip can be fractionated by excising sections of the gel, eluting the peptides into solution followed by further separation by the front-end liquid chromatography prior to in-line MS (Pergande and Cologna, 2017; Liu *et al.*, 2016).

As mentioned previously, MS is most often applied for analysis of proteins in a single cell or an organism. Once sample complexity has been reduced using the protein/peptide separation techniques outlined in Table 1, low abundance proteins peptides/proteins can be identified by label-free or labelled proteomic approaches. The respective advantages and disadvantages of these methods are outlined in Table 2.

These respective strengths and weaknesses must be considered as current proteomic methods still only identify a small percentage of the total proteome (as shown in Fig. 1) (Bantscheff *et al.*, 2007). Furthermore, proteomic investigations must also

Table 1 A comparison of protein/peptide fractionation or enrichment methods that are commonly employed in proteomic workflows in order to reduce sample complexity

Fractionation/ enrichment procedure	Principle	Strengths	Weaknesses	Comments	Ref (s)
<i>Gel-based fractionation methods</i>					
1D SDS-PAGE	Separation based on protein charge and mass (i.e. protein size)	1. Applicable to a wide variety of samples 2. Simple to run 3. Fast turnaround	1. Requires fairly large amount of sample 2. High inter-gel variability 3. Multiplexing is not possible 4. Requires multiple runs for quantitation 5. Difficult to identify low-abundance and small proteins unless the sample has been enriched for those proteins 6. Proteins with similar masses cannot be resolved due to masking 7. Incomplete digestion may introduce error	This techniques is very widespread and used for many applications such as comparing the protein composition of different samples, analysis of the number and size of polypeptide subunits, western blotting coupled to immunodetection	Tiwari and Tiwari (2014), Abdallah <i>et al.</i> (2012)
IPG-IEF	Separation based on isoelectric point (pI) of peptides and/or proteins	1. High-resolution and high-capacity method for peptide separation 2. Full control of a well-defined pH range 3. Flexibility of choosing the desired pH gradient 4. Buffering capacity 5. Reproducible separation for technical replicates	1. Oil from IPG-IEF fractionation must be removed prior to MS 2. Additional sample-handling steps 3. Cathodic drift of proteins with alkaline pls 4. Poor resolution under non-denaturing conditions	Separation by net charge at different pH values is not affected by protein molecular weight	Moreda-Pineiro <i>et al.</i> (2014) Abdallah <i>et al.</i> (2012); Manadas <i>et al.</i> (2010), Essader <i>et al.</i> (2005)
2D SDS-PAGE or 2D-DIGE	Separation based on pI (first dimension separation) and then by protein size (second dimension separation)	1. Simplistic to run 2. Robust 3. Can be multiplexed using Cy-dyes	1. Requires large sample amounts 2. Low throughput 3. High inter-gel variability 4. Co-migration of multiple proteins in a single spot renders comparative quantification rather inaccurate 5. Poor representation of low abundant proteins 6. Has issues separating highly acidic/basic proteins, or proteins with extreme size or hydrophobicity		Tiwari and Tiwari (2014), Abdallah <i>et al.</i> (2012)

(Continued)

Table 1 Continued

Fractionation/ enrichment procedure	Principle	Strengths	Weaknesses	Comments	Ref (s)
<i>Salts/detergent based enrichment methods</i>					
Sodium Carbonate	High alkalinity of sodium carbonate solution (pH 11) solubilises and strips loosely associated proteins from cell membranes	1. Simple procedure 2. Inexpensive reagents 3. Avoid labelling 4. Extended proteomic coverage 5. Highly abundant cytosolic proteins (e.g., ribosomal proteins, elongation factors) are easily separated	1. Require large amount of samples 2. Only enriches loosely bound membrane proteins which sometimes is not ideal 3. Involves centrifugation at $120,000 \times g$, which many not be suitable for all membrane types		Cantor <i>et al.</i> (2013), Molloy (2008)
Triton X-114 Phase partitioning	Triton X-114 is used to solubilise the membrane proteins which are then separated (at 20°C) from hydrophilic proteins	1. Simple workflow 2. Hydrophobic and hydrophilic proteins are enriched 3. Ideal for analysis of membrane proteins 4. Inexpensive	1. More difficult sample preparation procedure 2. Require large amount of samples 3. Sample loss at each step 4. Requires detergent removal which would otherwise interfere with MS analysis	This method was first introduced by Bordier in 1981. It has now established itself as one of the most powerful tools for preparing membrane proteins for analysis	Mathias <i>et al.</i> (2011), Pryde (1998); Bordier (1981)
<i>Chromatographic fractionation methods</i>					
Ion Exchange chromatography	AX: Positive groups have affinity for negatively charged peptides at basic pH	1. Robust 2. Less labour intensive	1. Limited resolution	Strong cation exchange (SCX) is the most commonly used fractionation technique for proteomics. It has also extensively used to study post-translational modifications	Chan and Issaq (2013), Mohammed and Heck (2011) Manadas <i>et al.</i> (2010), Essader <i>et al.</i> (2005)
Anion exchange chromatography (AX)	CX: Negative groups attract positively charged peptides at acidic pH				
Cation exchange chromatography (CX)	Separation is based on the analyte partition coefficient between the polar mobile phase and the hydrophobic (nonpolar) stationary phase	1. Can provide extremely high resolution separations 2. Reverse phase columns are relatively cheap and commercially available 3. Easy to use	1. Very sensitive to polarity, changes in temperature and pH 2. These can have profound effects on the results		Abdallah <i>et al.</i> (2012)
Reverse phase chromatography					

OFFGEL electrophoresis	Separation is based on pI of peptides/proteins	1. Improved pI resolutions 2. Sample is recovered in liquid phase 3. Low influence of offgel reagents on further separation techniques 4. High reproducibility 5. Multiple samples can be fractionated in parallel	1. Requires high amount of sample 2. Possible loss of sample 3. Long separation-times (few hours to 2–4 days) 4. Moderate number of protein identifications 5. Cathodic drift of proteins with alkaline pls 6. Poor resolution under non-denaturing conditions	Moreda-Pineiro <i>et al.</i> (2014), Manadas <i>et al.</i> (2010) http://www.agilent.com
<i>Depletion columns for enrichment</i>				
Multiple Affinity Removal System (MARS)-14	Antibody based protein capture	1. Removes 14 high-abundance proteins from human plasma samples 2. Enhances the ability to identify low abundance proteins	1. Variable depletion efficiency between proteins 2. Concomitant loss of non-targeted proteins 3. High cost 4. Can only be used with serum/plasma	Ahn and Khan (2014), Tu <i>et al.</i> (2010) Pernemalm <i>et al.</i> (2008) http://www.agilent.com
ProteoPrep20 Plasma Immunodepletion Kit	Antibody based protein capture	1. Removes the 20 most-abundant proteins which allows for 50-fold increased protein loads 2. Increase the ability to visualize low abundance proteins and biomarkers 3. Convenient spin column format	1. Can only be used with biological fluids such as serum/plasma 2. Variable depletion efficiency 3. Concomitant loss of non-targeted proteins 4. High cost	Liu <i>et al.</i> (2011), Cellar <i>et al.</i> (2009), Sigdel and Sarwal (2008) http://www.sigmaldrich.com
HiTrap Albumin and IgG Depletion columns	Media containing recombinant Protein G fragments and recombinant protein binding human serum albumin (HAS)	1. Columns can be prepacked for either serum or plasma 2. Rapid and easy process 3. Large sample volumes (~150 µl) 4. Can be used online with an LC system or be used manually with a syringe 5. High depletion capacity, > 95% HSA and 90% IgG	14 proteins: Albumin, IgG, antitrypsin, IgA, transferrin, haptoglobin, fibrinogen, alpha2-macroglobulin, alpha1-acidglycoprotein, IgM, apolipoprotein AI, apolipoprotein AII, complement C3, transthyretin. 20 proteins: Albumin, IgG, transferrin, fibrinogen, IgA, alpha2-macroglobulin, IgM, alpha1-antitrypsin, complement C3, haptoglobin, apolipoprotein A1, A3 and B; alpha1-acid glycoprotein, ceruloplasmin, complement C4, C1q; IgD, prealbumin, and plasminogen Smaller columns can be purchased as Spin Trap from GE Healthcare	de Moraes-Zani <i>et al.</i> (2011) http://www.gelifesciences.com

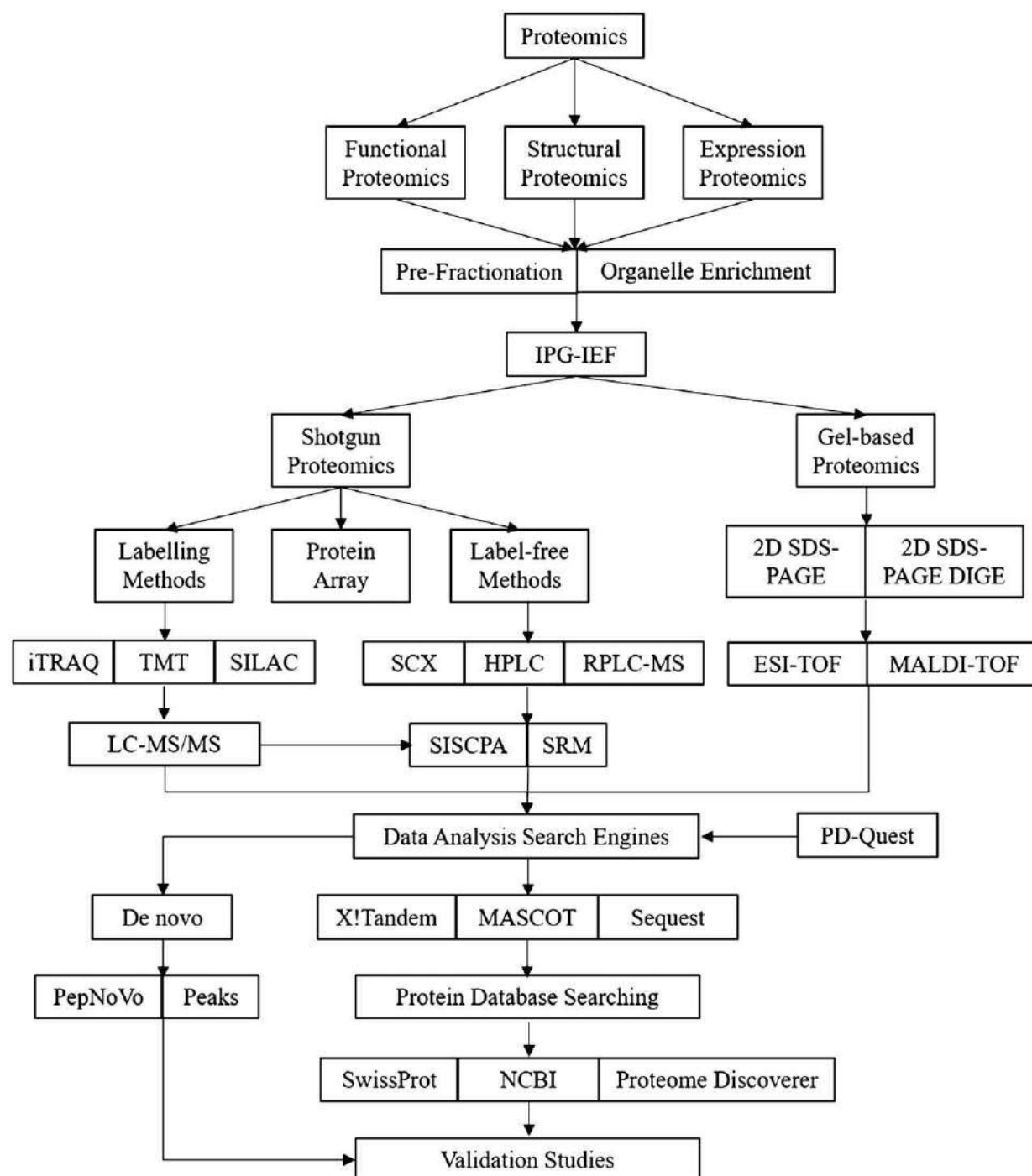


Fig. 2 An overview of mass spectrometry-based proteomic workflows that are commonly employed by biomedical researchers. Figure adapted from Chandramouli, K., Qian, P.Y., 2009. Proteomics: Challenges, techniques and possibilities to overcome biological sample complexity. *Hum Genomics Proteomics* 2009. doi:10.4061/2009/239204.

account for the required use of combination/s of separation techniques in order to reduce sample complexity, especially during technical stages which may introduce experimental variation and downstream quantitative error (Fig. 3).

Stand-alone MS is not inherently quantitative due to ion loss and the sources of variation outlined in Fig. 3. As a result, labelling methods such as TMT, iTRAQ or SILAC, or multiple reaction monitoring (MRM) are often employed to enable researchers to accurately quantitate protein expression between samples or in reference to standards (Chan *et al.*, 2016). The key challenge encountered in MS-based biomarker discovery studies is the need for the greatest proteomic coverage that can be reliably and reproducibly achieved in order to identify and quantify the low abundance proteins whose altered expression may indicate or mediate disease (Chan *et al.*, 2016).

Table 2 Relative strengths and weaknesses of labelled and label-free proteomic strategies

	Advantages	Disadvantages
Label-free proteomic approach	● Requires less amounts of sample ● Higher proteomic coverage ● Avoid labelling ● Reagents are inexpensive ● Applicable to wide variety of samples ● Applicable to a wider variety of sample preparation workflows	● Multiplexing is not possible in one experiment ● Semi-quantitative, requiring multiple runs for quantitation ● Not suitable for low abundance proteins unless the sample has been enriched for those proteins ● Incomplete digestion may introduce error ● Technical variation must be tightly controlled
Labelled proteomic approach	● Faster MS analysis, requiring fewer runs. ● Applicable to wide variety of samples ● Allows multiplexing (3–10 samples) in a single in MS run, reducing technical variation between MS gradients. ● Allows better quantitation through the use of standards ● Minimises background by identifying tagged peptides only ● Covalent labelling can be achieved using iTRAQ and TMT ● Metabolic labelling can be performed using SILAC	● Requires greater amounts of sample ● Incomplete/partial labelling can introduce quantitative error ● Loss of the label causes peptide loss ● Reagents are expensive, therefore limiting the number of experiments that can be performed using a single sample

Sources: Reproduced from Tiwari, V., Tiwari, M., 2014. Quantitative proteomics to study carbapenem resistance in *Acinetobacter baumannii*. *Front Microbiol* 5, 512; Abdallah, C., Dumas-Gaudot, E., Renaut, J., Sergeant, K., 2012. Gel-based and gel-free quantitative proteomics approaches at a glance. *Int J Plant Genomics* 2012, 494572.

As a result, additional dimensions of sample separation are beneficial to biomarker discovery studies as they increase resolution and sensitivity of the shotgun proteomic workflow, whilst facilitating the application of a pI filter to reduce false positive or false negative PSMs (Cargile *et al.*, 2005). This article will explore the use of the gel-based and gel-free methods to identify differentially-expressed proteins, outlining their respective strengths and utility for vastly different proteomic applications, from the detection of performance enhancing drugs to elucidating the molecular biology of leading cancer subtypes.

Application of IPG-IEF

IPG-IEF is a robust protein/peptide fractionation technique that provides reproducible high resolution separation, (Moreda-Pineiro *et al.*, 2014; Abdallah *et al.*, 2012) and is therefore an attractive method to employ in biomarker studies where it is crucial to overcome the issue of sample heterogeneity. As a method of separation, IPG-IEF and its related methodologies have been a widely employed approach to fractionate protein mixtures prior to MS (Pergande and Cologna, 2017; Cargile *et al.*, 2005). One of the earliest incarnations of this approach was two-dimensional electrophoresis (2DE), in which IPG-IEF was applied as the first dimension of separation. The strength of 2DE over one-dimensional gels was the ability to separate and identify different proteoforms present within the sample. This additional dimension enabled early efforts to investigate, characterise and identify post-translational modifications (PTMs), the strengths of which have been reviewed by Westermeyer (2014).

However, early 2DE efforts were hamstrung by the inherent variations in gel composition and electrophoretic conditions, resulting in potentially significant gel-to-gel variation. Despite this issue, when employed in isolation, one of the greatest powers of 2DE and related IEF techniques was their complementarity with mass spectrometry. When used together, two-dimensional separation was able to reduce proteome complexity with sufficient resolution to facilitate the analysis of single proteoforms in greater depth compared to a standard shotgun approach (Pergande and Cologna, 2017). Furthermore, it is now common practice to perform 2DE with multiple Cy-dye labelled samples, thereby separating multiplexed protein samples under identical electrophoretic conditions and enabling researchers to visualise differential protein expression based on the size and location of protein spots. These gel spots are then easily excised and processed for MS analysis by either LC-MS or MALDI-TOF for *m/z*-based identification (Pergande and Cologna, 2017).

Additionally, the recent development of high-resolution isoelectric focusing (HiRIEF) by Branca *et al.* (2014) has allowed far deeper proteome coverage and the discovery of new protein-coding loci (Branca *et al.*, 2014). Using this specialised method, Branca *et al.*, were able to identify 77,985 human peptides, 39,941 of which had not been present in the human subset of the 2012 release of Peptide Atlas, corresponding to an increase of ~52% (Branca *et al.*, 2014). HiRIEF employs a series of ultra-narrow IPG gradients within the acidic pH 3.7–5.0 range (i.e., 3.70–4.05, 4.00–4.25, 4.20–4.45 and 4.39–4.99) which were then fractionated into 72 fractions for analysis by LC-MS/MS (Branca *et al.*, 2014). This method concentrated on the separation of acidic peptides as it enabled the identification of one third of the human tryptic peptidome. Furthermore, fractionation can be performed with minimal redundancy whilst remaining relatively robust in terms of resolution, the reduction in sample complexity and overall

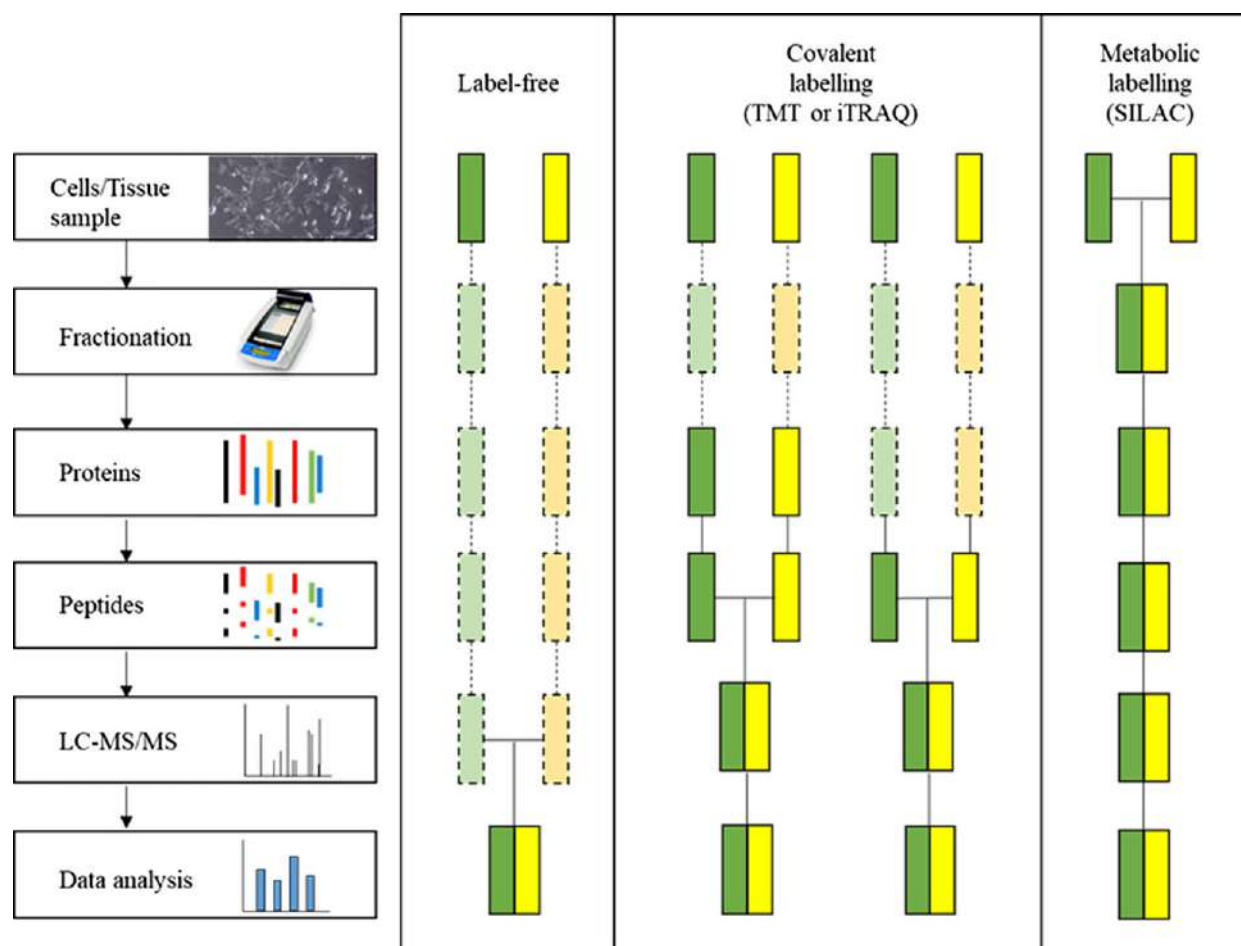


Fig. 3 An overview of potential sources of variation and quantitative error within various proteomic workflows. Boxes displayed in green or yellow represent separate experimental conditions along a vertical workflow, with horizontal lines indicating stages where samples/data are combined. Paler boxes with dashed lines indicate technical stages that are susceptible to experimental variation. Covalent chemical labelling can be performed at either the protein or peptide level depending upon experimental workflow. Figure modified and adapted from Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., Kuster, B., 2007. Quantitative mass spectrometry in proteomics: A critical review. *Anal Bioanal Chem* 389, 1017–1031.

reproducibility (Branca *et al.*, 2014). HiRIEF has more recently been applied to analyse sequential phosphorylation by protein kinases through the combined use of fluorescent peptide substrates with microfluidic isoelectric focusing and monitoring of protein folding dynamics (Pergande and Cologna, 2017).

IPG-IEF has been applied to investigate a great number of biological questions, some of which have been summarised in Table 3. These studies employed IPG-IEF and/or related methodologies in order to investigate an array of disease models, or to identify and/or elucidate the role of specific proteins as potential markers of disease. Here, we provide two case studies to highlight the application of IPG-IEF in biomedical research. In the first case study, IPG-IEF was used in combination with MARS depletion and iTRAQ to identify markers of the use of Human Growth Hormone in athletes as a performance enhancing drug; whilst in the second case study, IPG-IEF was paired with sample enrichment techniques to facilitate the identification of low-abundance and technically challenging proteins by LC-MS.

Case Studies

Case Study I

The first Case Study that we wish to explore is the recently published work by Tan *et al.* (2017). This study demonstrates the application of proteomic technologies to identify markers of substance abuse, specifically the administration of human growth hormone (GH) as a performance enhancing drug in competitive sports (Tan *et al.*, 2017). The World Anti-Doping Agency currently adopts two approaches for the detection of GH abuse in elite sports; the GH isoform test and GH biomarker test (Tan *et al.*, 2017). The GH biomarker tests measure the concentration of two human GH-sensitive markers which are present in serum; insulin-like

Table 3 A selection of IPG-IEF-based investigations that have been undertaken in the past 5 years

Author	Year	Disease/cancer type/ application	IPG-IEF utilisation	Key findings/notes	References
Tan SH <i>et al.</i>	2017	Testing of human growth hormone intake	MARS-7 depleted plasma samples were subjected to 2D-DIGE and selected spots were identified using MALDI-TOF MS	52 Spots from two phases were excised for MS analysis 20 Proteins were identified from both phases - Five proteins AHSG, APOL1, KNG1, VDBP and VN were differentially expressed - Further cross validation using iTRAQ and Western blotting showed AHSG and APOL1 could be potential growth hormone responsive markers in athletes	Tan <i>et al.</i> (2017)
Panizza E <i>et al.</i>	2017	Identification of low abundance phospho-proteins using HeLa cells	Phospho-peptide enriched samples labelled with TMT tags were subjected to high-resolution isoelectric focusing (HRIEF) and examined by LC-MS/MS analysis	- This is a novel workflow for identification of low abundance phospho-proteins - Identified a total of 22,712 phosphorylation sites 19,075 (14,965 Serine, 2916 threonine and 1203 tyrosine) phosphorylation sites were localized with high confidence 1264 Novel sites (previously unreported) were identified	Panizza <i>et al.</i> (2017)
Wang DL <i>et al.</i>	2017	Breast cancer	Serum samples enriched for low abundant proteins were subjected to 2D-DIGE ^a and the separated proteins were transferred to the protein elution plate (PEP) and hexokinase (HK) activity was measured. Followed by measurement of protease activity using FITC-labelled casein as the protease substrate	- Several potential functional protein biomarkers were identified from breast cancer patient sera - The workflow used showcases functional proteomics technology	Wang <i>et al.</i> (2017)
Sun Z <i>et al.</i>	2016	Lung cancer	2D-DIGE was employed following which the samples were transferred to the PEP and HK activity was measured. Fractions with high HK activity were analysed by LC-MS/MS	- Many time-dependent hexokinase activity fractions were detected from both the normal and lung cancer sera - Multiple fractions showed a 10-fold difference between normal serum and lung cancer sera - Hexokinase regulation can provide a potential panel of biomarkers for lung cancer	Sun <i>et al.</i> (2016)

(Continued)

Table 3 Continued

Author	Year	Disease/cancer type/ application	IPG-IEF utilisation	Key findings/notes	References
Liu S <i>et al.</i>	2016	Cervical cancer	iTRAQ-labelled peptides were fractionated by IPG-IEF followed by MS analysis	1. MS analysis identified 3300 proteins of which 137 and 193 were found to be up-regulated down-regulated respectively. 2. LYN was significantly overexpressed in higher in cervical cancer tissues 3. LYN knockdown and overexpression studies showed that it could inhibit or promote migration, invasion and cell proliferation in vitro and in vivo	Liu <i>et al.</i> (2016)
Hsu TY <i>et al.</i>	2016	Edwards Syndrome (Trisomy 18) Pregnancies	Samples labelled with Cy Dye were subjected to 2D-DIGE. Spots with a fold change of more than ± 1.5 were selected for MS/MS analysis by MALDI-TOF/TOF	1. 12 Spots were chosen for MS/MS analysis 2. 5 Spots were successfully identified, of which APOA1 was up-regulated and four proteins, IGFBP-1, VDBP, TTR, and A1AT were down-regulated 3. Further validation of these results is required using a larger sample size	Hsu <i>et al.</i> (2016)
Nakamura S <i>et al.</i>	2015	Tumour samples from gastric, pancreatic, colon, ovarian, renal, breast, lung, hepatocellular, oesophageal, prostate and thyroid cancers	Samples were subjected to 2D-DIGE, following which MS/MS analysis of selected spots was carried out by using MALDI-TOF/TOF Proteomic studies were performed using a FLAG-LIX1L-expressing HEK-293 (HEK-293FLG-LIX1L) cell line	1. Oncogenic activity of LIX1L and associated proteins was carried out using proteomic analysis 2. LIX1L was found to be associated with C-p89 and C-p80 kDa protein complexes when the cytoplasmic fraction was not treated with RNAase 3. Among the complexes, high immunoreactivity was observed for RIOK1, NCL and PABPC4 4. MS analysis of the nuclear fraction identified DHX9, NCL and HNRNPL to be differentially-expressed 5. Along with several other experiments LIX1L was suggested to be a putative RNA-binding proteins with implications in for therapeutic approaches for targeting LIX1L in LIX1L-expressing cancer cells	Nakamura <i>et al.</i> (2015)

Wu JY <i>et al.</i>	2014	Gastric cancer	Cy Dye labelled samples were subjected to 2D-DIGE followed by MALDI-TOF MS analysis	15 proteins spots were examined. 10 proteins were excluded under the suspicion of PTMs as they were found in different locations - Five proteins, GRP78, GSTP1, APOA1, A1AT and GKN1 were confirmed and further validated by Western blotting and IHC as putative markers of gastric cancer 1371 proteins were identified in the screened pleural effusions from patients with malignant mesothelioma ($n=6$), lung adenocarcinoma ($n=6$), or benign mesotheliosis ($n=7$) - Seven candidates from the proteomic data (AKR1B10, APOC1, LGALS1, MYO7B, SOD2, TNC, and THBS1) were validated by ELISA in a larger group of patients with mesothelioma ($n=37$) or metastatic carcinomas ($n=25$) - Galectin 1, aldo-keto reductase 1B10, and apolipoprotein C-I were all identified as potential prognostic biomarkers for malignant mesothelioma	Wu <i>et al.</i> (2014)
Mundt F <i>et al.</i>	2014	Malignant mesothelioma	iTRAQ-labelled samples were subjected to HIRIEF. 72 fractions were collected and subjected to LC-MS/MS analysis	1371 proteins were identified in the screened pleural effusions from patients with malignant mesothelioma ($n=6$), lung adenocarcinoma ($n=6$), or benign mesotheliosis ($n=7$) - Seven candidates from the proteomic data (AKR1B10, APOC1, LGALS1, MYO7B, SOD2, TNC, and THBS1) were validated by ELISA in a larger group of patients with mesothelioma ($n=37$) or metastatic carcinomas ($n=25$) - Galectin 1, aldo-keto reductase 1B10, and apolipoprotein C-I were all identified as potential prognostic biomarkers for malignant mesothelioma	Mundt <i>et al.</i> (2014)
Cantor D <i>et al.</i>	2013	Colorectal cancers	A membrane-enriched protein sample from SW480 cells overexpressing integrin $\beta 6$ was separated by IPG-IEF and the resulting fractions were examined by LC-MS/MS	708 proteins were found to have significant change in expression 54 proteins previously proposed as cancer biomarkers were identified $\beta 6$ expression enhanced cell invasion and proteins involved in cancer-related pathways such as ILK and RAN	Cantor <i>et al.</i> (2013)

^a2D-DIGE: First-dimension separation is done using isoelectric focussing (IEF) which is facilitated by immobilized pH gradient (IPG) strips. The second dimension separation performed using a gel.

Note: Observe the variety of applications and ability to be incorporated to different biological investigations, from identifying expression changes in various disease models to phospho-peptide enrichment and identification. Please note that Table 3 only identifies a brief selection of studies in order to demonstrate the utility of workflows incorporating IPG-IEF.

growth factor-I (IGF-I) and N-terminal pro-peptide of type III collagen (P-III-NP) (Tan *et al.*, 2017). However, these tests are insufficiently specific to indict or exonerate an athlete as having been administered GH as they do not ameliorate the variability imparted by factors such as age, gender, sport, ethnicity and dosage concentration. Furthermore, the GH isoform test remains unable to detect the administration of purified, pituitary-derived GH, which is commercially-available. As a result, there was an unmet need to accurately identify the exogenous administration of GH to athletes (Tan *et al.*, 2017). In their study, Tan *et al.* (2017) aimed to identify novel markers of GH administration in athlete plasma samples by an unbiased proteomic screening approach. This approach employed gel-based and gel-free proteomic methods which were undertaken in parallel. A simplified outline of this workflow is shown in Fig. 4.

Tan *et al.* employed a simple antibody-based sample enrichment step by utilising a MARS Human 7 (MARS-7) depletion column to remove seven high-abundance plasma proteins (albumin, IgG, IgA, transferrin, haptoglobin, antitrypsin, and fibrinogen) which respectively equated to approximately 40–50% of total plasma protein (by weight) (Tan *et al.*, 2017). Following the removal of these abundant constituents, the flowthrough from MARS-7 immunodepletion was then analysed in an unbiased fashion by employing a parallel analysis of each sample by 2D-DIGE and iTRAQ-based LC-MS analysis (Tan *et al.*, 2017).

In order to minimise technical variability between the experiments, Tan *et al.* carried out the study in two independent phases, wherein the 112 individual plasma samples were separated into two equal groups of 56 (Tan *et al.*, 2017). In an effort to further

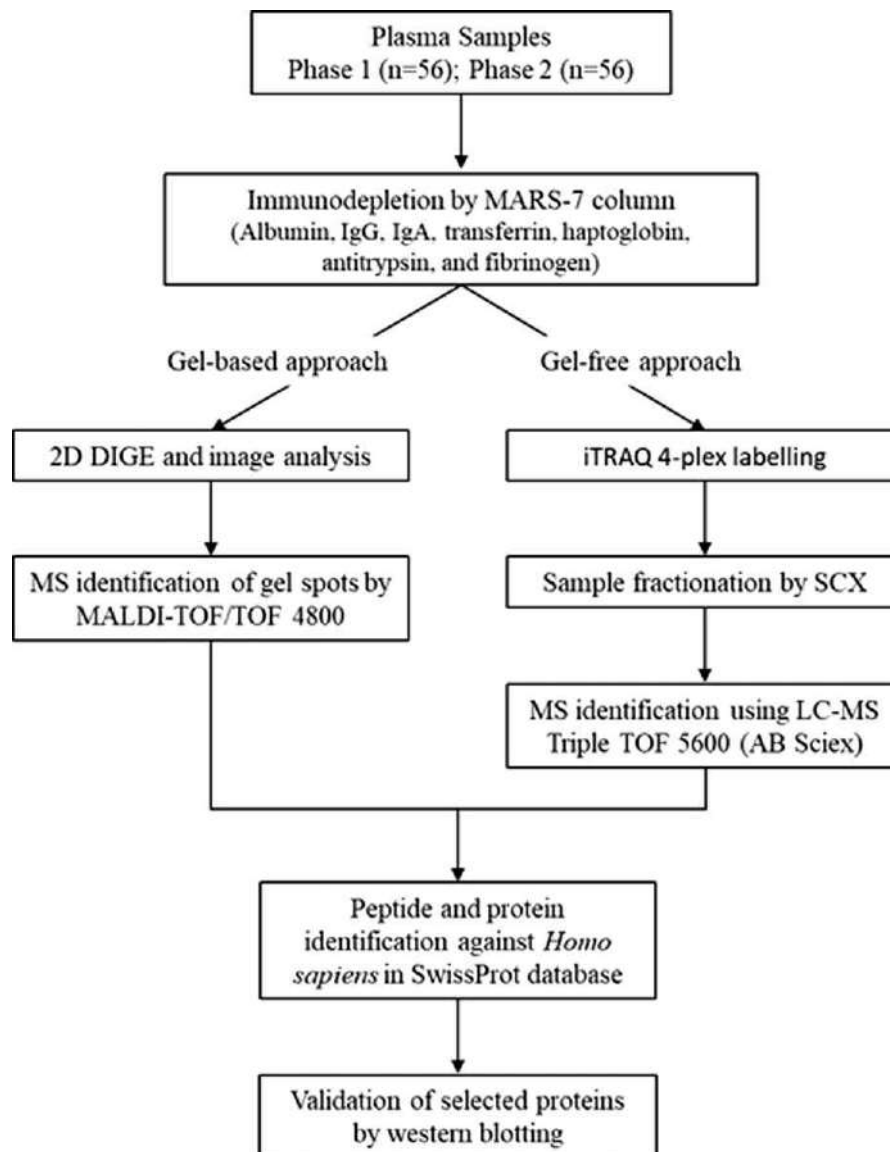


Fig. 4 Schematic overview of the proteomic workflow employed by Tan *et al.* The plasma samples were immunodepleted using a MARS-7 column with the flowthrough analysed in parallel by 2D-DIGE and iTRAQ LC-MS/MS. This was followed by peptide and protein identification, with subsequent validation performed upon of selected proteins. Reproduced from Tan, S.H., Lee, A., Pascovici, D., *et al.*, 2017. Plasma biomarker proteins for detection of human growth hormone administration in athletes. *Sci Rep* 7 (1), 10039.

reduce variation, Tan *et al.* created a pooled reference sample which was then incorporated into each DIGE gel to facilitate gel alignment and matching. By adopting these two simple approaches, Tan *et al.* estimated the coefficient of variation (CV) between 28 gels in this first phase of the study to be only 8% (Tan *et al.*, 2017), which was considered acceptable given the extent of sample handling involved in performing a DIGE experiment.

Following image analysis of DIGE gels, 195 proteins were found to be differentially expressed. Stringent selection criteria (Tan *et al.*, 2017) were then imposed upon on the candidate spots, leading to the selection of 54 spots for further analysis. Of these, only 52 spots were excised for identification by MALDI-MS/MS analysis. It is important to note that the remaining two spots were not visible when the gels were re-stained, resulting in their omission from MALDI-MS/MS analysis. This highlights one of the disadvantages of employing a gel-based approach, as some proteins may not bind to the dye used to visualise and excise protein spots for MS analysis. Of the 52 spots analysed by MS, 39 differentially-expressed proteins were identified, of which 20 were differentially expressed from both Phases. Among the identified proteins, APOL1, AHSG, KGN1, VDBP, and VN were also shown to be differentially expressed in the recombinant GH (rGH)-treated group from both analysis Phases. Subsequent analysis later showed that KNG1 and VN exhibited variable expression patterns between Phases 1 and 2, resulting in their exclusion from further evaluation (Tan *et al.*, 2017). Tan *et al.* reported that these varying expression patterns for KNG1 and VN could be a result of individual variability.

In this study, Tan *et al.* employed 2D-DIGE, an IPG-IEF methodology, to investigate the effects of rGH administration to athletes. 2D-DIGE is a robust and fairly simplistic method to evaluate protein samples, providing an in-depth view of the proteome whilst identifying differentially-expressed proteins (Tiwari and Tiwari, 2014; Abdallah *et al.*, 2012). To negate the issue of dynamic range, Tan *et al.* performed 2D-DIGE on plasma samples whose complexity had been reduced by pre-treatment on a MARS-7 immunodepletion column (Tan *et al.*, 2017). However, while immunodepletion columns are excellent in reducing sample complexity, they possess significant disadvantages such as those outlined in Table 1, which must be carefully considered. Encouragingly, Tan *et al.* demonstrated excellent reproducibility from the immunodepletion process, estimating that ~40%–50% of the total plasma proteins (by weight) had been successfully removed with a coefficient of variation of 3.5% (Tan *et al.*, 2017). This low coefficient of variation supports the experimental system and that conditions were tightly controlled during preliminary depletion of the plasma samples.

In addition to immunodepletion, the application of 2D-DIGE in this study further reduced sample complexity by providing two additional layers of separation (i.e., pI and mass) prior to identification of selected gel spots by MALDI-MS/MS. These separations reduced sample complexity by multiple folds, however it did not negate the co-migration of multiple proteins in a single spot (Tiwari and Tiwari, 2014; Abdallah *et al.*, 2012) or the inherent weaknesses of 2D-DIGE that were observed in this study. For example, AHSG and KNG1 proteins were concurrently identified from spot 1921 (Phase 1), and AHSG and VN proteins were identified from spot 1021 (Phase 2). Despite this observation, the results from the DIGE study identified several differentially-expressed proteins which outweighed the observation of co-migration in few gel-spots (Tan *et al.*, 2017).

In addition to 2D DIGE, Tan *et al.* employed iTRAQ LC-MS/MS, an independent approach, using pooled samples. Unlike 2D-DIGE which considers proteoforms as independent gel spots, iTRAQ protein quantitation is inferred from the detected tryptic peptides (Tan *et al.*, 2017). Using this approach, Tan *et al.* identified 385 proteins (Tan *et al.*, 2017). Upon imposing filtering restrictions of $a \geq \pm 1.2$ fold change ($p \leq 0.05$), ≥ 2 peptides match and 95% confidence, 7 unique proteins were found to be differentially expressed between the rGH-treated and placebo-treated (Group A) samples, whilst 10 unique proteins were found to be differentially-expressed between within the rGH-treated baseline against other time-points (Group B). Six proteins (APOL1, IGFBP-ALS, AFM, IGFBP 3, LUM, and ECM1) were found to be consistently different between Group A and Group B. Surprisingly, only APOL1 was similarly observed in the results of the 2D-DIGE analysis.

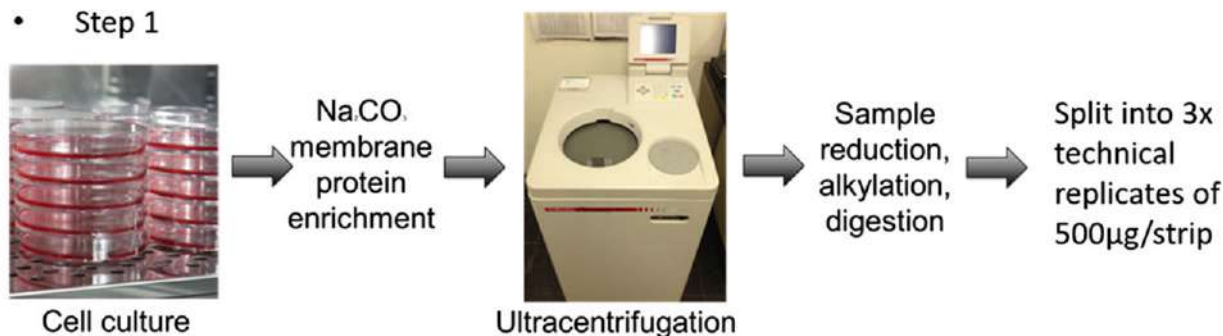
These 6 proteins that were identified by iTRAQ and the 3 proteins (APOL1, AHSG and VDBP) that were identified by 2-D DIGE were then proposed as rGH-dependent biomarkers of GH administration. Further statistical analysis showed that two proteins, APOL1 and AHSG, were found to be consistently GH-responsive within the 2D-DIGE studies and that this trend was also observed for APOL1 from the iTRAQ study (Tan *et al.*, 2017). However, AHSG did not show significant difference between rGH-treated and placebo-treated samples in iTRAQ analysis (Tan *et al.*, 2017). To confirm these expression changes, APOL1 and AHSG were evaluated by Western blotting which supported the proteomic results observed (Tan *et al.*, 2017). Unfortunately, AFM, IGFBP3, IGFBP-ALS, LUM and ECM1 proteins, which showed significant differential expression, were not evaluated by Western blotting (Tan *et al.*, 2017). Validation of these observations would be of value for clinical practices and should be evaluated by independent studies.

In summary, Tan *et al.* employed an unbiased proteomic approach to identify eight rGH-dependent plasma proteins. Two of these differentially expressed proteins (APOL1 and AHSG) had not been previously demonstrated to be GH-responsive and indicated potential clinical utility. These putative biomarkers were then validated by Western blotting to confirm both their identities and their expression patterns in the rGH- and placebo-treated subject cohorts (Tan *et al.*, 2017). We believe that independent studies on the remaining putative GH-responsive biomarkers would be of great value for clinical practices by providing an alternate method for detecting the administration of GH to athletes as a performance-enhancing drug.

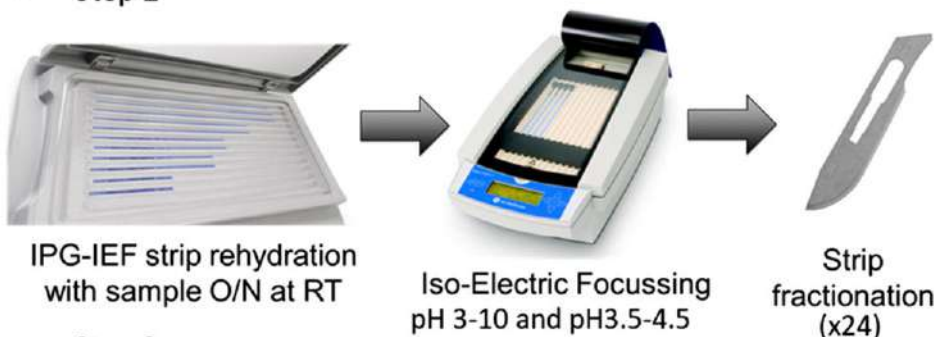
Case Study II

In the second Case Study, we wish to demonstrate how IPG-IEF can be utilised to profile the technically-challenging membrane fraction of whole cell lysate, enriching low-abundance hydrophobic proteins for detection by LC-MS/MS. In our previously

- Step 1



- Step 2



- Step 3

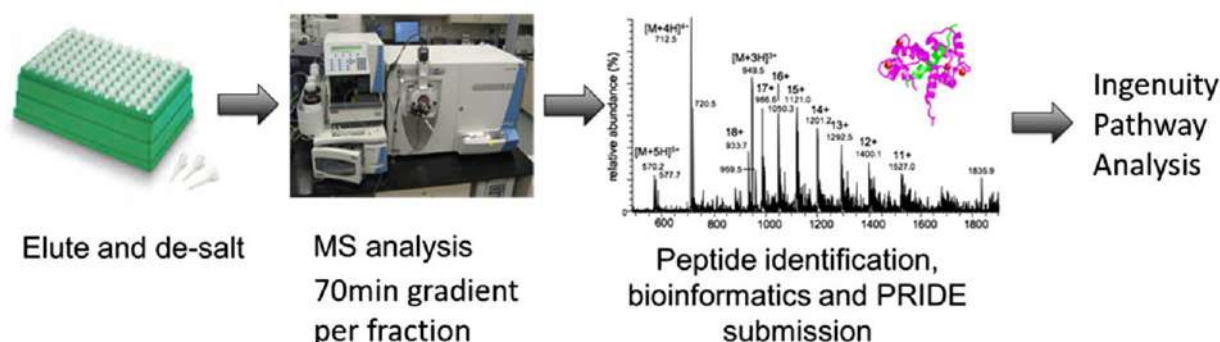


Fig. 5 Schematic overview of the proteomic workflow employed in the study (Cantor *et al.*, 2013). There are three main steps to the workflow presented. The first involved the culturing of at least 10^8 cells per cell line and strip range, followed by cell lysis, membrane protein enrichment, ultracentrifugation to precipitate the membrane-enriched fraction, sample reduction, alkylation and finally methanol-assisted tryptic digestion. The second step consisted of rehydrating the IPG-IEF strips with the peptide sample, followed by iso-electric focussing, before strip fractionation and elution. The third step of the workflow involved the elution of digested peptides out of the gel strip followed by sample clean-up, MS analysis, peptide identification and bioinformatics analysis.

published work, Cantor *et al.* (2013) generated a proteomic profile of CRC cell lines in order to determine the effect/s of induced expression of a suspected driver of CRC aggression (Cantor *et al.*, 2013). This was achieved by enriching CRC cell lysates for membrane-associated proteins by performing a 0.1 M sodium carbonate wash followed by ultracentrifugation, and has been displayed in a workflow schematic in Fig. 5 (Cantor *et al.*, 2013).

This simplified enrichment method was employed as integral- or peripheral membrane proteins are often under represented within proteomic studies due to a combination of low relative abundance and reduced aqueous solubility; yet it is the cell membrane which contains the greatest potential for drug targeting or biomarker identification as the plasma membrane contains the interface between the cell and the extracellular space (Yin and Flynn, 2016). Furthermore, integral membrane proteins have been suggested to constitute greater than 60% of current therapeutic targets (Yin and Flynn, 2016). Following reduction, alkylation and methanol-assisted tryptic digestion, the membrane-enriched lysate samples were focussed on linear IPG-IEF strips with pH ranges of 3–10 and 3.5–4.5, respectively, fractionated and desalted to remove MS-incompatible salts (Cantor *et al.*, 2013). Membrane-enriched peptides were then analysed by LC-MS/MS followed by the calculation of normalised spectral abundance factors (NSAF) to test for statistical significance.

Comparing the two cell lines found that 492 proteins were significantly up- or downregulated ($p < 0.05$) in the pH 3–10 dataset, whilst 263 proteins were significantly different in the pH 3.5–4.5 dataset (Cantor *et al.*, 2013). Interestingly, of these 263 additional proteins, 216 were unique to the pH3.5–4.5 strip range, illustrating that delving deeper into the acidic peptide population was able to bolster the number of proteins identified by 43%, identifying proteins that may have been masked in the broader pH3–10 range (Cantor *et al.*, 2013). In total, 708 proteins were significantly up- or downregulated between the two cell lines, potentially indicating a proteomic signature of inducing the expression of a key cancer-associated protein, the $\alpha v\beta 6$ integrin (Cantor *et al.*, 2013). Pairing sodium carbonate-based membrane protein enrichment with IPG-IEF resulted in the identification of 115 membrane proteins whose expression was significantly different in the pH3–10 dataset, potentially representing actionable biomarkers (Cantor *et al.*, 2013). Once again, the use of the narrower pH range of 3.5–4.5 afforded a deeper insight into the membrane proteome, identifying 186 membrane-associated proteins whose expression was significantly different between the two cell lines (Cantor *et al.*, 2013). This is likely due to fact that while membrane proteins are hydrophobic in nature, they invariably contain hydrophilic peptide sequences that interact with the aqueous environment.

IPG-IEF was a valuable component of this study, providing an inexpensive yet effective insight into the membrane proteome of CRC cell lines in order to identify significant shifts in protein expression that may promote CRC aggression and metastasis. Membrane proteomics has historically been regarded as being inherently difficult due to the nature of the highly hydrophobic sub-population and their relatively low abundance within the cell (Helbig *et al.*, 2010), however, IPG-IEF helped overcome some of these intrinsic challenges. Firstly, adopting IPG-IEF helped reduce sample complexity by providing an additional dimension of sample preparation by separating peptides based on pI prior to separation by reversed phase high performance liquid chromatography prior to in-line MS. By fractionating the sample into 24 pI-based fractions, each 70 min LC-MS/MS gradient was able to observe and fractionate a greater number of peptides than would have been possible for the unfractionated digest. Though this did not remove high-abundance proteins from the sample, low-abundance peptides from proteins of interest (i.e. the urokinase-type plasminogen activator receptor and the αv and $\beta 6$ subunits of the $\alpha v\beta 6$ integrin) were able to be identified by MS for the first time within our research group. Though protein depletion methods such as MARS-14 or immunoaffinity columns are also able to reduce sample complexity, there remains a risk of variable depletion efficiency between proteins (Tu *et al.*, 2010; Pernemalm *et al.*, 2008) and the concomitant loss of non-targeted proteins (Bellei *et al.*, 2011; Zolotarjova *et al.*, 2008). Both of these consequences must be tightly controlled in biomarker discovery studies and require the purchase of expensive consumables to be performed.

The workflow employed in this study was also able to overcome the challenges of membrane protein solubility without resorting to the use of detergents or other MS-incompatible solvents, which would require additional sample clean-up procedures. This in particular was a result of employing methanol-assisted tryptic digestion, wherein proteins were digested in a 60:40 mixture of LC-MS-grade methanol to reduced and alkylated protein preparation (Chick *et al.*, 2008). The use of methanol increases the mobility of integral membrane proteins, increasing tryptic access to cleavage sites that may be shielded by associated proteins or by their proximity to the plasma membrane itself. Methanol is easily removed during routine vacuum drying and provides an inexpensive alternative to detergents such as CHAPS or Triton X-114 which must be carefully removed prior to LC-MS/MS lest they damage the column prior to in-line MS and impede successful liquid chromatography. Though it can be performed without much technical difficulty, it must be mentioned that this label-free workflow was labour intensive, requiring a great deal of sample handling and the generation of considerable amounts of starting material (10–20 mg of lysate protein per cell line and strip range) in order to account for sample loss at each stage of the process. In summary, IPG-IEF proved to be a successful method from the extensive toolbox of proteomic methods and techniques, effectively delving into this particular niche of the CRC proteome.

Discussion

Identifying differentially-expressed proteins is a key requirement to elucidate the molecular processes of cells, tissues or organs, whether they are healthy, responding to hormone treatment or afflicted with diseases such as cancer. The importance of identifying differentially expressed proteins is that it can reveal the associated protein networks that are disturbed or significantly altered between biological conditions. This is particularly important when disturbances can be observed in protein networks responsible for cellular signalling activity or the governance of cellular behaviours such as growth, migration and death. Understanding these disturbed protein networks in cancer specimens can even yield clinically-relevant information such as the potential loss of avenues for therapeutic intervention.

For example, the expression or absence of the estrogen, progesterone and Her-2 receptors are used to classify breast cancers and guide treatment options. The absence of these three oncoproteins in a breast cancer specimen denotes a particularly heterogenous form of breast cancer ('triple-negative') that is characterised by poor clinical outcomes for patients and a lack of targeted treatment options for clinicians (Lawrence *et al.*, 2015). Therefore, there is little purpose in employing treatments that target the Her-2 receptor in order to antagonise triple-negative breast cancers which already lack the expression of the oncoprotein. However, proteomic analysis of the cancer tissue may reveal the molecular features of a tumour specimen and potentially indicate the presence of compensatory signalling pathway activity, such as that observed by Lawrence *et al.* in the Akt1/2 pathway, which may then serve as a personalised option for treatment (Lawrence *et al.*, 2015). Strategies such as those employed by Lawrence *et al.* (2015), have successfully sought to quantify signalling proteins, biological pathways and biomarkers by mass spectrometry. However, despite their extensive study, Lawrence *et al.* employed a very simple sample preparation workflow involving lysis, brief centrifugation, reduction and alkylation followed by digestion with Lys-C or trypsin, desalting and stepped high-pH RP-HPLC prior to LC-MS/MS. Though understandable to adopt a

simple shotgun proteomic workflow given the extensive amount of work performed in their study, Lawrence *et al.* were still only able to identify 12,775 distinct proteins encoded by 11,466 out of the estimated 20,537 genes (~56%) with a protein false discovery rate of <1%. Thus, despite significant advances from identifying ~3000 proteins per sample from our study in 2013 to ~9000 for each of the twenty samples in 2015 (Lawrence *et al.*, 2015), there remains a great need to increase proteomic coverage in order to identify the technically-challenging or low-abundance biomarkers that remain undetected.

Therefore, linking sequential protein separation methodologies such as IPG-IEF with RP-HPLC fractionation or Hi-Res IPG-IEF stands to greatly increase proteomic coverage (Griffel, 1970). However, this consequently increases sample processing time and may introduce additional layers of handling-based technical variation; both of which are not ideal for biomarker studies. In spite of this, there is no current workflow which can sufficiently overcome all the various challenges of the human proteome. Despite the need to manage the strengths and weaknesses of respective workflows, MS-based methods stand to become increasingly quantitative, meeting the sensitivity and speed to measure proteomes at depths that are comparable to gene expression studies (Lawrence *et al.*, 2015; Kim *et al.*, 2014; Wilhelm *et al.*, 2014). Paired with specific sample preparation and fractionation methods, MS-based investigations are providing an increasingly favoured platform for biomarker discovery compared to the more traditional antibody-based methods. One of the key strengths of MS-based studies is the specificity of the analyte being detected; whilst antibodies recognise a three dimensional epitope that is susceptible to denaturation or proteolysis, MS provides a much more robust *m/z* value, corresponding to the specific mass and charge state for each ionised peptide detected. Though a protein may be degraded by uncontrolled proteolysis, the *m/z* value of a digested peptide will remain unique to that amino acid sequence, whilst remaining unaffected by denaturation of the secondary/tertiary protein structure; a factor that could prevent the formation of the antibody/antigen complex. Though MS-based methods excel at identifying the amino acid sequence of a peptide, they are less capable of identifying the extensive array of post-translational modifications (PTMs) that can occur during protein maturation. These variable structures and less predictable fragmentation patterns are much more difficult to identify from spectra. On the other hand, antibodies can be raised against phosphorylated or glycosylated forms of proteins and detected much more easily and efficiently. MS-based methods are also highly dependent upon the properties of the instrumentation. For example, quadrupole mass spectrometers exhibit greater mass ranges and resolution than ion trap instruments. Time-of-flight-based instruments demonstrate greater mass ranges again and are much faster; however, this comes at the cost of lower resolution and difficulty in adapting to electrospray ionisation. As a result, MS platforms must be carefully chosen in order to ensure that they possess the required capabilities for analysis.

Future Directions

Antibody-based methods have been routinely used in the previous decades of biomolecular research as they offered the ability to be signal-amplifying, detecting small quantities amongst a complex and heterogeneous background which early MS-based approaches had difficulty penetrating. It has only been in recent years with the adoption of efficient sample fractionation methodologies and the continual development of increasingly sensitive and selective MS instruments that MS-based methods have begun to surpass those requiring the use of antibodies.

In this article, we have explored IPG-IEF and its related techniques as a means to identify differentially-expressed proteins that would enable researchers to peer deeper into the observable proteome, shedding further insight into the molecular biology of cells and tissues. Once a distinct set of proteins have been observed to consistently reflect a pathological condition such as disease or cancer, it then will become necessary to validate these potential biomarkers by highly-selective MS methods such as SRM, MRM or sequential window acquisition of all theoretical mass spectra (SWATH) to ensure that the relative shift in protein expression can reproducibly indicate the presence of disease. As data-independent acquisition (DIA) methods such as SWATH begin to supersede most traditional proteomic technologies in the study of whole proteomes, it is important to recognise that it is a costly exercise to generate spectral libraries in order to perform full DIA-based analysis on a patient by patient basis, both in terms of financial cost and the time required.

Despite their weaknesses, MS-based methods are becoming increasingly economical once the capital equipment (i.e., mass spectrometer) has been purchased, with day to day running costs being relatively inexpensive as opposed to the necessity to purchase antibodies and specialised reagents for methods such as ELISA. Furthermore the use of MS-based proteomic workflows counters a key weakness of antibody-based methods which depend upon the availability of specific antibodies (either as a single or pair) which possess sufficient specificity for the analyte. High specificity is a crucial requirement as poor specificity can result in reduced sensitivity, leading to false positive/negative results. This issue also extends to the fact that immunoassays are multi-staged processes that are susceptible to batch-to-batch variation of the antibody itself, with a greater potential for challenges to both intra- and inter-lab reproducibility compared to MS-based workflows. MS-based investigations also stand to facilitate the analysis of smaller sample volumes than methods such as ELISA, requiring smaller sample amounts whilst also enabling further characterisation of the antigen beyond mere abundance. This is afforded by providing sequence identification and potential analysis of protein isoforms and post-translational modifications, which cannot be supported by immunoassays. Whilst both platforms offer the opportunity to multiplex assays, immunoassays offer the simultaneous detection of 92 human proteins by utilising the Olink Proseek proximity extension assays whilst MS is capable of detecting 12,775 distinct protein species (Lawrence *et al.*, 2015; Mahboob *et al.*, 2015).

The continual development and refinement of tools and techniques are helping to reshape the field of proteomics from a role of investigative discovery in biomedical research to a first-response clinical tool (Cantor *et al.*, 2015). These developments support the use of MS as a primary technology in biomarker discovery as protein identifications must be robust, capable of identifying a specific peptide or single reaction monitoring (SRM) transition which may be inherently variable as a result of post-translational

modifications or glycosylation. MS stands as an effective method to identify differentially-expressed proteins in a manner that is reproducible between patients and institutions, in order to establish proteins of interest as clinically-relevant biomarkers.

Closing Remarks

All diseases involve proteins and networks of proteins, be it directly or indirectly. As a result, the relative flux of protein populations can be employed as biological insights into disease. However, the key challenge that faces biomedical researchers is the difficulty of finding that one protein of interest amongst the billions of somatic house-keeping proteins that constitute >99% of the cellular proteome. By continuing to drill deeper, it is hoped that the field of proteomics can generate more extensively characterised interpretations of what is occurring both within and between cells. It is the hope that these shifts in expression may ultimately be utilised as a clinical tool, either as a routine diagnostic or surveillance method, indicating the development of a disease before the onset of symptoms and thereby affording clinicians the best chance for a favourable patient outcome.

Acknowledgement

The authors would like to greatly thank Dr. Abidali Mohamedali (Macquarie University, Australia) for proof-reading the manuscript prior to submission. This manuscript has been prepared by the authors in their own time, with each contributing to the text. The authors declare that they received neither financial nor material assistance for the preparation of this work.

Appendix 1: List of gene abbreviations used in the article

<i>Abbreviation</i>	<i>Gene Name</i>	<i>Protein name</i>	<i>UniProtKB accession number</i>
AFM	AFM	Afamin	P43652
AHSG	AHSG	Alpha-2-HS-glycoprotein	P02765
AKR1B10	AKR1B10	Aldo-keto reductase family 1 member B10	O60218
APOA1	APOA1	Apolipoprotein A1	P02647
APOC1	APOC1	Apolipoprotein C-I	P02654
APOL1	APOL1	Apolipoprotein L1	O14791
N-p150	DHX9	ATP-dependent RNA helicase A	Q08211
ECM1	ECM1	Extracellular matrix protein 1	Q16610
VDBP	GC	Vitamin D-binding protein	P02774
GKN1	GKN1	Gastrophilin-1	Q9NS71
GSTP1	GSTP1	Glutathione S-transferase P	P09211
N-p68	HNRNPL	Heterogeneous nuclear ribonucleoprotein L	P14866
GRP78	HSPA5	78 kDa glucose-regulated protein	P11021
IGFBP-1	IGFBP1	Insulin-like growth factor-binding protein 1	P08833
IGFBP3	IGFBP3	Insulin-like growth factor-binding protein 3	P17936
IGFBP-ALS	IGFBPALS	Insulin-like growth factor-binding protein complex acid labile subunit	P35858
ILK	ILK	Integrin-linked protein kinase	Q13418
KGN1	KNG1	Kininogen-1	P01042
LGALS1	LGALS1	Galectin-1	P09382
LIX1L	LIX1L	LIX1-like protein	Q8IVB5
LUM	LUM	Lumican	P51884
LYN	LYN	Tyrosine-protein kinase Lyn	P07948
MYO7B	MYO7B	Myosin-VIIb	Q6PIF6
C-p90	NCL	Nucleolin	P19338
PABPC4	PABPC4	Polyadenylate-binding protein 4	Q13310
RAN	RAN	GTP-binding nuclear protein Ran	P62826
C-p89	RIOK1	Serine/threonine-protein kinase RIO1	Q9BRS2
A1AT	SERPINA1	Alpha-1-antitrypsin	P01009
SOD2	SOD2	Superoxide dismutase, mitochondrial	P04179
THBS1	THBS1	Thrombospondin-1	P07996
TNC	TNC	Tenascin-C	P24821
TTR	TTR	Transthyretin	P02766
VN	VTN	Vitronectin	P04004

See also: Analyzing Transcriptome-Phenotype Correlations. Characterization of Transcriptional Activities. Comparative Transcriptomics Analysis. Genome-Wide Scanning of Gene Expression. Natural Language Processing Approaches in Bioinformatics. Regulation of Gene Expression. Statistical and Probabilistic Approaches to Predict Protein Abundance. Survey of Antisense Transcription. Transcriptome Analysis. Transcriptome During Cell Differentiation

References

- Abdallah, C., Dumas-Gaudot, E., Renaut, J., Sergeant, K., 2012. Gel-based and gel-free quantitative proteomics approaches at a glance. *Int J Plant Genomics* 2012, 494572.
- Ahn, S.B., Khan, A., 2014. Detection and quantitation of twenty-seven cytokines, chemokines and growth factors pre- and post-high abundance protein depletion in human plasma. *EuPA Open Proteomics* 3, 78–84.
- Bantscheff, M., Schirle, M., Sweetman, G., Rick, J., Kuster, B., 2007. Quantitative mass spectrometry in proteomics: A critical review. *Anal Bioanal Chem* 389, 1017–1031.
- Bellei, E., Bergamini, S., Monari, E., *et al.*, 2011. High-abundance proteins depletion for serum proteomic analysis: Concomitant removal of non-targeted proteins. *Amino Acids* 40, 145–156.
- Bordier, C., 1981. Phase separation of integral membrane proteins in Triton X-114 solution. *J Biol Chem* 256, 1604–1607.
- Branca, R.M., Orre, L.M., Johansson, H.J., *et al.*, 2014. HiRIEF LC-MS enables deep proteome coverage and unbiased proteogenomics. *Nat Methods* 11, 59–62.
- Cantor, D.I., Nice, E.C., Baker, M.S., 2015. Recent findings from the Human Proteome Project: Opening the mass spectrometry toolbox to advance cancer diagnosis, surveillance and treatment. *Expert Rev Proteomics* 12, 279–293.
- Cantor, D., Slapetova, I., Kan, A., McQuade, L.R., Baker, M.S., 2013. Overexpression of alphavbeta6 integrin alters the colorectal cancer cell proteome in favor of elevated proliferation and a switching in cellular adhesion that increases invasion. *J Proteome Res* 12, 2477–2490.
- Cargile, B.J., Sevinsky, J.R., Essader, A.S., Stephenson Jr., J.L., Bundy, J.L., 2005. Immobilized pH gradient isoelectric focusing as a first-dimension separation in shotgun proteomics. *J Biomol Tech* 16, 181–189.
- Cellar, N.A., Karnoup, A.S., Albers, D.R., Langhorst, M.L., Young, S.A., 2009. Immunodepletion of high abundance proteins coupled on-line with reversed-phase liquid chromatography: A two-dimensional LC sample enrichment and fractionation technique for mammalian proteomics. *J Chromatogr B Anal Technol Biomed Life Sci* 877, 79–85.
- Chan, K.C., Issaq, H.J., 2013. Fractionation of peptides by strong cation-exchange liquid chromatography. *Methods Mol Biol* 1002, 311–315.
- Chan, P.P., Wasinger, V.C., Leong, R.W., 2016. Current application of proteomics in biomarker discovery for inflammatory bowel disease. *World J Gastrointest Pathophysiol* 7, 27–37.
- Chick, J.M., Haynes, P.A., Bjellqvist, B., Baker, M.S., 2008. A combination of immobilised pH gradients improves membrane proteomics. *J Proteome Res* 7, 4974–4981.
- Essader, A.S., Cargile, B.J., Bundy, J.L., Stephenson Jr., J.L., 2005. A comparison of immobilized pH gradient isoelectric focusing and strong-cation-exchange chromatography as a first dimension in shotgun proteomics. *Proteomics* 5, 24–34.
- Ferlay, J., Soerjomataram, I., Dikshit, R., *et al.*, 2015. Cancer incidence and mortality worldwide: Sources, methods and major patterns in GLOBOCAN 2012. *Int J Cancer* 136, E359–E386.
- Griffel, B., 1970. Kidney in collagen diseases. *Harefuah* 78, 455–457.
- Helbig, A.O., Heck, A.J., Slijper, M., 2010. Exploring the membrane proteome – Challenges and analytical strategies. *J Proteomics* 73, 868–878.
- Hsu, T.Y., Lin, H., Hung, H.N., *et al.*, 2016. Two-dimensional differential gel electrophoresis to identify protein biomarkers in amniotic fluid of Edwards syndrome (Trisomy 18) pregnancies. *PLOS ONE* 11, e0145908.
- Kim, M.S., Pinto, S.M., Getnet, D., *et al.*, 2014. A draft map of the human proteome. *Nature* 509, 575–581.
- Lawrence, R.T., Perez, E.M., Hernandez, D., *et al.*, 2015. The proteomic landscape of triple-negative breast cancer. *Cell Rep* 11, 630–644.
- Liu, B., Qiu, F.H., Voss, C., *et al.*, 2011. Evaluation of three high abundance protein depletion kits for umbilical cord serum proteomics. *Proteome Sci* 9, 24.
- Liu, S., Hao, X., Ouyang, X., *et al.*, 2016. Tyrosine kinase LYN is an oncotarget in human cervical cancer: A quantitative proteomic based study. *Oncotarget* 7, 75468–75481.
- Mahboob, S., Ahn, S.B., Cheruku, H.R., *et al.*, 2015. A novel multiplexed immunoassay identifies CEA, IL-8 and prolactin as prospective markers for Dukes' stages A-D colorectal cancers. *Clin Proteomics* 12, 10.
- Manadas, B., Mendes, V.M., English, J., Dunn, M.J., 2010. Peptide fractionation in proteomics approaches. *Expert Rev Proteomics* 7, 655–663.
- Mathias, R.A., Chen, Y.S., Kapp, E.A., *et al.*, 2011. Triton X-114 phase separation in the isolation and purification of mouse liver microsomal membrane proteins. *Methods* 54, 396–406.
- Mohammed, S., Heck Jr., A., 2011. Strong cation exchange (SCX) based analytical methods for the targeted analysis of protein post-translational modifications. *Curr Opin Biotechnol* 22, 9–16.
- Molloy, M.P., 2008. Isolation of bacterial cell membranes proteins using carbonate extraction. *Methods Mol Biol* 424, 397–401.
- Morais-Zani, D.E., Grego, K., Tanaka, A. S. K.F., Tanaka-Azevedo, A.M., 2011. Depletion of plasma albumin for proteomic analysis of *Bothrops jararaca* snake plasma. *J Biomol Tech* 22, 67–73.
- Moreda-Pineiro, A., Garcia-Otero, N., Bermejo-Barrera, P., 2014. A review on preparative and semi-preparative offgel electrophoresis for multidimensional protein/peptide assessment. *Anal Chim Acta* 836, 1–17.
- Mundt, F., Johansson, H.J., Forshed, J., *et al.*, 2014. Proteome screening of pleural effusions identifies galectin 1 as a diagnostic biomarker and highlights several prognostic biomarkers for malignant mesothelioma. *Mol Cell Proteomics* 13, 701–715.
- Nakamura, S., Kahyo, T., Tao, H., *et al.*, 2015. Novel roles for LIX1L in promoting cancer cell proliferation through ROS1-mediated LIX1L phosphorylation. *Sci Rep* 5, 13474.
- Panizza, E., Branca, R.M.M., Olviusson, P., Orre, L.M., Lehtio, J., 2017. Isoelectric point-based fractionation by HiRIEF coupled to LC-MS allows for in-depth quantitative analysis of the phosphoproteome. *Sci Rep* 7, 4513.
- Pergande, M.R., Cologna, S.M., 2017. Isoelectric point separations of peptides and proteins. *Proteomes* 5.
- Pernemalm, M., Orre, L.M., Lenggqvist, J., *et al.*, 2008. Evaluation of three principally different intact protein prefractionation methods for plasma biomarker discovery. *J Proteome Res* 7, 2712–2722.
- Prieto, D.A., Johann Jr., D.J., Wei, B.R., *et al.*, 2014. Mass spectrometry in cancer biomarker research: A case for immunodepletion of abundant blood-derived proteins from clinical tissue specimens. *Biomark Med* 8, 269–286.
- Pryde, J.G., 1998. Partitioning of proteins in Triton X-114. *Methods Mol Biol* 88, 23–33.
- Sigdel, T.K., Sarwal, M.M., 2008. The proteogenomic path towards biomarker discovery. *Pediatr Transplant* 12, 737–747.
- Sun, Z., Chen, X., Wang, G., *et al.*, 2016. Identification of functional metabolic biomarkers from lung cancer patient serum using PEP technology. *Biomark Res* 4, 11.
- Tan, S.H., Lee, A., Pascovici, D., *et al.*, 2017. Plasma biomarker proteins for detection of human growth hormone administration in athletes. *Sci Rep* 7, 10039.
- Tiwari, V., Tiwari, M., 2014. Quantitative proteomics to study carbapenem resistance in *Acinetobacter baumannii*. *Front Microbiol* 5, 512.
- Tu, C., Rudnick, P.A., Martinez, M.Y., *et al.*, 2010. Depletion of abundant plasma proteins and limitations of plasma proteomics. *J Proteome Res* 9, 4982–4991.

- Uhlen, M., Fagerberg, L., Hallstrom, B.M., *et al.*, 2015. Proteomics. Tissue-based map of the human proteome. *Science* 347, 1260419.
- Wang, D.L., Xiao, C., Fu, G., Wang, X., Li, L., 2017. Identification of potential serum biomarkers for breast cancer using a functional proteomics technology. *Biomark Res* 5, 11.
- Westermeier, R., 2014. Looking at proteins from two dimensions: A review on five decades of 2D electrophoresis. *Arch Physiol Biochem* 120, 168–172.
- Wilhelm, M., Schlegel, J., Hahne, H., *et al.*, 2014. Mass-spectrometry-based draft of the human proteome. *Nature* 509, 582–587.
- Wu, J.Y., Cheng, C.C., Wang, J.Y., *et al.*, 2014. Discovery of tumor markers for gastric cancer by proteomics. *PLOS ONE* 9, e84158.
- Yin, H., Flynn, A.D., 2016. Drugging membrane protein interactions. *Annu Rev Biomed Eng* 18, 51–76.
- Zolotarjova, N., Mrozinski, P., Chen, H., Martosella, J., 2008. Combination of affinity depletion of abundant proteins and reversed-phase fractionation in proteomic analysis of human plasma/serum. *J Chromatogr A* 1189, 332–338.

Further Reading

- Aebersold, R., Bader, G.D., Edwards, A.M., *et al.*, 2013. The biology/disease-driven human proteome project (B/D-HPP): Enabling protein research for the life sciences community. *J Proteome Res* 12, 23–27. doi:10.1021/pr301151m.
- Amaya, M., Baer, A., Voss, K., *et al.*, 2014. Proteomic strategies for the discovery of novel diagnostic and therapeutic targets for infectious diseases. *Pathog Dis* 71, 177–189. doi:10.1111/2049-632X.12150.
- Anjo, S.I., Santa, C., Manadas, B., 2015. Short GeLC-SWATH: A fast and reliable quantitative approach for proteomic screenings. *Proteomics* 15, 757–762. doi:10.1002/pmic.201400221.
- Baker, M.S., Ahn, S.B., Mohamedali, A., *et al.*, 2017. Accelerating the search for the missing proteins in the human proteome. *Nat Commun* 8, 14271. doi:10.1038/ncomms14271.
- Cao, J., Seegmiller, J., Hanson, N.Q., Zaun, C., Li, D., 2015. A microfluidic multiplex proteomic immunoassay device for translational research. *Clin Proteomics* 12, 28. doi:10.1186/s12014-015-9101-x.
- Chick, J.M., Haynes, P.A., Molloy, M.P., *et al.*, 2008. Characterization of the rat liver membrane proteome using peptide immobilized pH gradient isoelectric focusing. *J Proteome Res* 7, 1036–1045. doi:10.1021/pr700611w.
- Dayon, L., Sanchez, J.C., 2012. Relative protein quantification by MS/MS using the tandem mass tag technology. *Methods Mol Biol* 893, 115–127. doi:10.1007/978-1-61779-885-6_9.
- Eravci, M., Sommer, C., Selbach, M., 2014. IPG strip-based peptide fractionation for shotgun proteomics. *Methods Mol Biol* 1156, 67–77. doi:10.1007/978-1-4939-0685-7_5.
- Gao, Y., Wang, X., Sang, Z., *et al.*, 2017. Quantitative proteomics by SWATH-MS reveals sophisticated metabolic reprogramming in hepatocellular carcinoma tissues. *Sci Rep* 7, 45913. doi:10.1038/srep45913.
- Lee, M.S., Ji, Q.C. (Eds.), 2017. Protein Analysis using Mass Spectrometry: Accelerating Protein Biotherapeutics from Lab to Patient. John Wiley & Sons, Inc. Available at: <http://dx.doi.org/10.1002/9781119371779>
- Ponten, F., Schwenk, J.M., Asplund, A., Edqvist, P.H., 2011. The Human Protein Atlas as a proteomic resource for biomarker discovery. *J Intern Med* 270, 428–446. doi:10.1111/j.1365-2796.2011.02427.x.

Relevant Websites

- <https://www.proteinatlas.org>
Human Protein Atlas.
- www.nature.com/subjects/proteomic-analysis
Nature.com.
- <https://www.nextprot.org/>
NeXtProt.
- <http://www.srmatlas.org/>
SRM Atlas.
- <https://www.hupo.org>
The Human Proteome Organisation (HUPO).
- <http://www.missingproteins.org/protein/web/>
The Missing ProteinPedia.

Biographical Sketch



David Cantor is a Scientific Officer with extensive experience in proteomic research. After being awarded a Bachelor of Medical Science (Hons) from Macquarie University in 2011, he undertook a PhD in Biomedical Sciences, investigating the role of the $\alpha v \beta 6$ in integrin in colorectal cancer. During this project, he performed an array of proteomic and phenotypic studies in order to elucidate the molecular biology of this putative oncogene. After being awarded his PhD in 2017, he has been responsible for the preparation and analysis of various samples/specimens within the Australian Proteome Analysis Facility (APAF). He has trained multiple students in the performance of IPG-IEF and other proteomic workflows and has authored or co-authored several scientific publications in the field.



Harish Cheruku is a PhD graduate with extensive experience in proteomic research. After being awarded a Master of Biotechnology from Macquarie University in 2010, he worked as a Research Assistant and continued on to a PhD in Biomedical Sciences, investigating the biology of transforming growth factor-beta (TGF β) in colorectal cancer. He utilised an extensive set of proteomic tools to elucidate the effects TGF β in colorectal cancer during his PhD which was awarded in 2017. He has extensive experience in label-free and label-based proteomic workflows in addition to various cell and molecular biology techniques. He has authored or co-authored several scientific publications in the field.

Clinical Proteomics

Marwenie F Petalcorin, Queen Mary University of London, London, United Kingdom

Naeem Shafqat and Zen H Lu, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

Mark IR Petalcorin, PAPRSB Institute of Health Sciences, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei Darussalam

© 2019 Elsevier Inc. All rights reserved.

Introduction

Clinical proteomics is the study of the entire protein content measured in patient's samples at a specific state or condition. Its primary goal is to analyze global protein profile of a biological system (cells, tissues, fluids) at a specific time point and state to confirm diagnosis, monitor treatment, screen for diseases, and take prognostic medical decisions (He and Chiu, 2003; Kulasingham and Diamandis, 2008a,b). The term "clinical" comes from the Greek word "klinike," which means "bedside," and the term "proteomics" is derived from both "proteome" and "omics" referring to the large-scale study of proteins including modified structures, variants, and mutations collectively called "proteoforms" (Schmit *et al.*, 2017; Toby *et al.*, 2016; Patrie, 2016; Wasinger *et al.*, 1995; Wilkins *et al.*, 1996a,b; Trenchevska *et al.*, 2016; Thygesen *et al.*, 2018). There is currently a tremendous amount of discoveries and information on proteins derived by mass spectrometry (MS) that can be applied in the clinic and can eventually revolutionize medical sciences and treatment. Although the amount of clinical proteomic profiles generated by MS measurements is remarkably increasing during the past two decades as indicated by the increase in rate of publications (Fig. 1), there is still a current gap between the patient's proteome profile and the real time conversion of these protein data into a more meaningful format that can reliably be applied in the clinics (Geyer *et al.*, 2017; Carvalho *et al.*, 2015; Lisitsa *et al.*, 2014; Kelleher *et al.*, 2014).

Multiparametric Panel of Proteome Profiles

Traditional clinical biochemistry relies on laboratory investigation results based on only a few single individual protein biomarkers, such as alanine aminotransferase (in liver function test), albumin, and creatinine (in kidney function test) among many others, to diagnose, treat, and monitor diseases while clinical proteomics considers profiling, which is the detection of panels of multiple biomarkers that provide more sensitivities and specificities (Penn *et al.*, 2018; Geyer *et al.*, 2017). In an era of information deluge with increasing data storage capacity and faster computational capability, the current challenge is how to increase further the technological advancement in proteomics instrumentation and data processing. The multiparametric and global nature of proteomic profiles derived from a patient's sample requires translation into a more useful form that is clinically relevant and more suitable in making diagnostic and prognostic decisions over the health status of patients (Matthiesen and Carvalho, 2013; Matthiesen, 2013a,b). This means increasing the sensitivity and robustness of MS-based instruments coupled with good experimental design standardized sampling techniques and more powerful bioinformatics analysis. This is necessary to generate accurate multiple biomarker profile panels that truly reflect the entire complement of proteins at a given particular state and condition (Matthiesen and Carvalho, 2013; Matthiesen and Bunkenborg, 2013; Penn *et al.*, 2018).

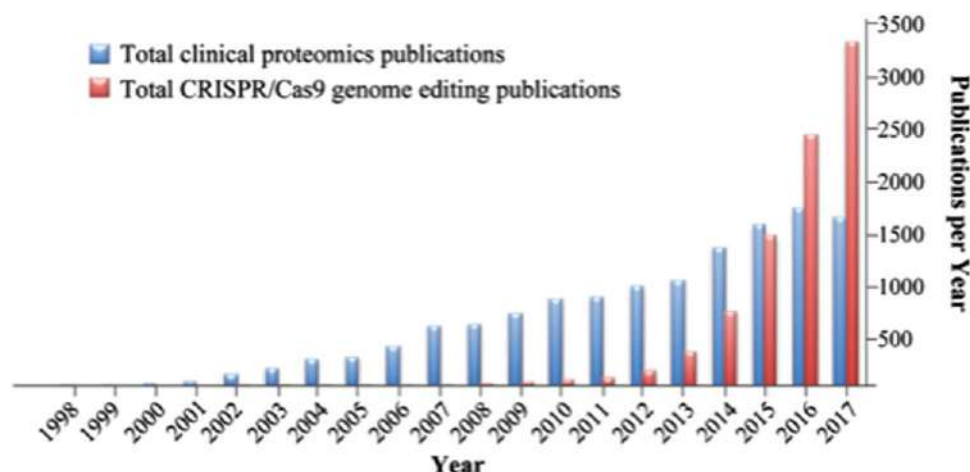


Fig. 1 Clinical proteomics and CRISPR/Cas9 literature review comparison. Total number of publications using MS-based clinical proteomics (blue) in comparison with the total number of publications in CRISPR/Cas9 (red), which is a technology that revolutionized genome editing recently and driving the medical field towards personalized treatments.

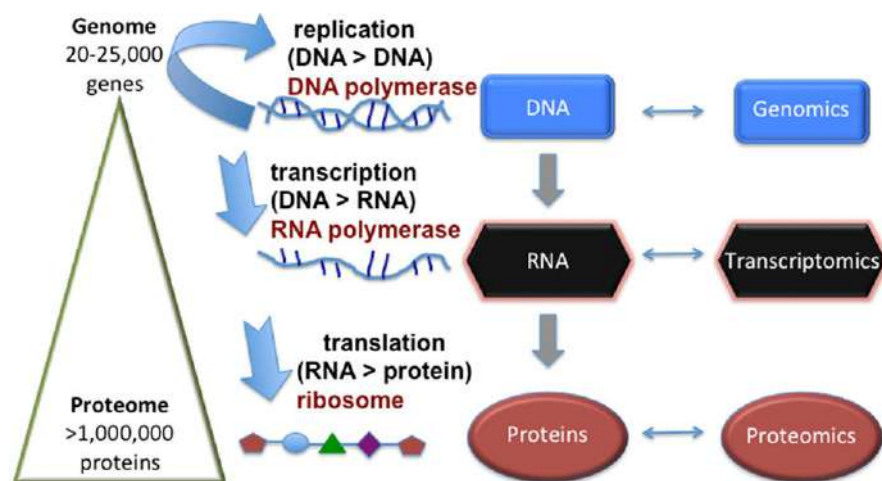


Fig. 2 Pathway from genome to proteome. Proteome is the entire complement of proteins including modifications produced by an organism or system. This set of proteins varies with time, physiological condition, and exposure to environment including stress that the cell or organism experience. Comparing between genome and proteome, genomics focuses on products of replicating the genetic code from DNA to another DNA mediated by DNA polymerase. Transcriptomics studies the products after transcribing DNA to RNA by the action of RNA polymerase. Proteomics, on the other hand, investigates the products of protein synthesis that uses the genetic code in RNA to produce proteins through ribosomal activity. The human genome is static over time with approximately 20,000 genes, which can be amplified by polymerase chain reaction (PCR), while the human proteome is dynamic that changes over time with more than 1,000,000 different proteins (according to the partial proteome map published in 2014) that determine functionalities of cells, tissues, and the human body.

Post-Translational Modifications in Proteoforms

Following the identification of about 20,500 genes after the completion of the human genome sequence in the year 2000, the “omics” era including genomics, transcriptomics, metabolomics, and proteomics began to receive a lot more attention (Van Karnebeek *et al.*, 2018; Piening *et al.*, 2018; Lacoste, 2018). The focus of scientific efforts has shifted from the human genome towards identifying the entire human proteome and measuring all the coded proteins and proteoforms (Fig. 2). Each gene gives rise to many proteins of various derivatives and alterations that cater to a particular specific functional role for homeostasis maintenance of the body’s biological processes. These functional alterations are achieved by differential RNA splicing and post-translational modifications (PTM), such as phosphorylation, glycosylation, acetylation, proteolytic processing, and numerous other types of PTMs that can diversify protein structure, conformation, and functionalities to produce over a million types of proteins comprising the entire human proteome (Nickchi *et al.*, 2015; Sergeant *et al.*, 2017). Simultaneously analyzing multiple changes in PTMs by quantitative PTMomics is useful for biomarker discovery and disease mechanisms elucidation using low amounts of tissue or body fluids (Thygesen *et al.*, 2018). Diseases occur with deviation from the normal homeostatic metabolic processes resulting in specific and detectable changes in protein profiles. Thus, direct measurements of the global protein changes can be performed by MS to determine both the diseased and normal states of the human body. The protein patterns in a patient sample correlate with disease characteristics that can predict clinical outcomes. However, the changes in protein levels are more complex and dynamic than that of genomics as they are easily influenced by several factors such as disease occurrence, environmental changes, diet or biological perturbations, and the combinatorial interactions among these factors. It is for this reason that to gain clinical acceptance, the use of protein profiles requires improvement in its disease-specificity and sensitivity by reducing non-disease-related variables. Under personalized medicine, each individual serves as their own control across various time points of protein measurements that look for any quantitative difference in abundance from the baseline or before and after clinically relevant perturbations (Keshishian *et al.*, 2015).

Single Reactor in Sample Preparation

The most desirable techniques that address the limited availability of clinical samples are those with high sensitivity to detect protein profiles using the smallest amount of sample. The desirability of microanalytical procedure with miniaturization is defined by sample volume reduction that saves precious reagents and limits clinical sampling resulting in faster analysis time, and increased high throughput potential with possible automation of single-use devices that suppress contamination (Lion *et al.*, 2004). The use of a single-reactor in a pipette tip that is fitted with a variety of filters for protein digestion, separation, and purification (Kulak *et al.*, 2014) is less invasive showing improvement in the efficiency of sample preparation with minimal sample loss (Hughes *et al.*, 2014). To avoid protein alteration or contamination and to optimize protein digestion, a known concentration of samples is separated from reaction salts that impede readout and is stored at a specified temperature. Samples are labeled with

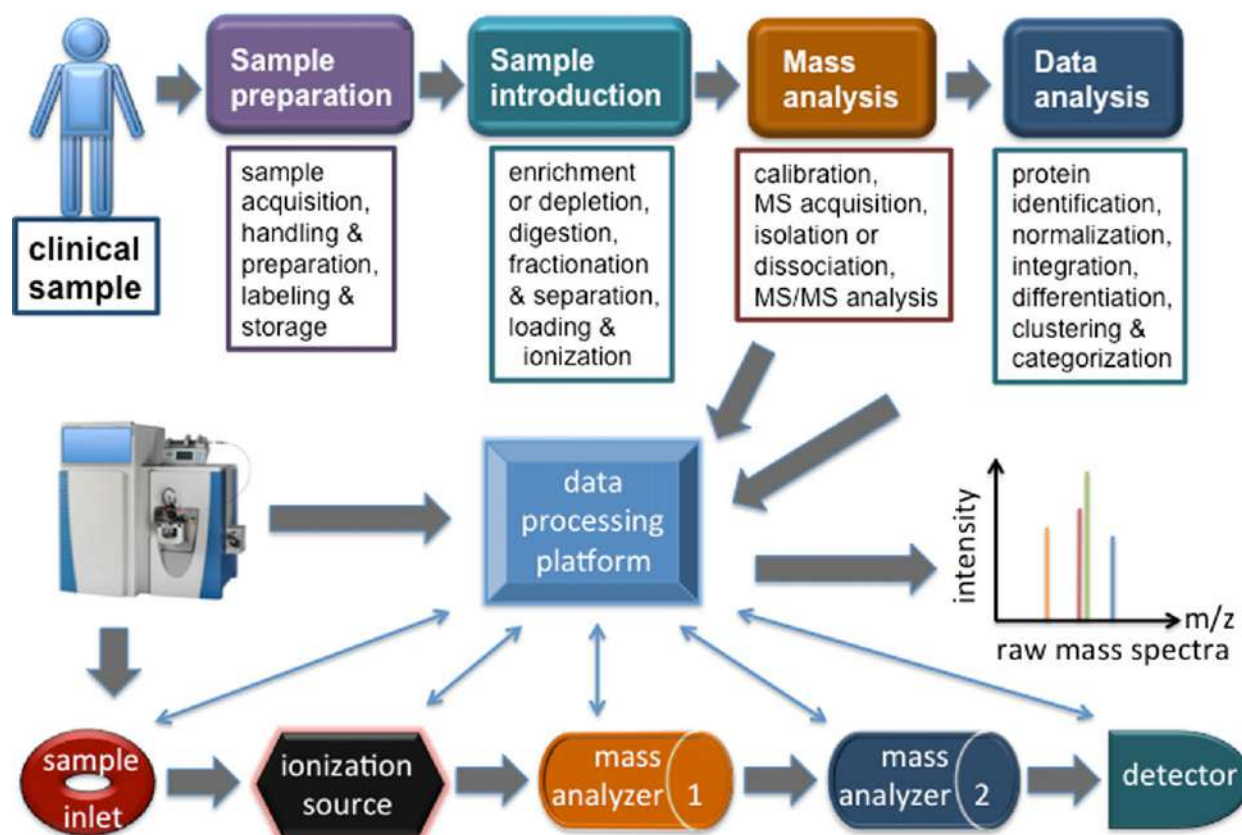


Fig. 3 Mass spectrometry-based clinical proteomics workflow from clinical sample acquisition and introduction to mass and data analysis. All mass analyzers use electromagnetic fields to manipulate gas-phase ions. Results are plotted as *spectra*, with mass-over-charge (m/z) on the X-axis and ion intensity on the Y-axis. The latter can be *absolute* (counts) or *relative*. The *ionization source* ensures that (a part) of the *sample* molecules are ionized and brought into the gas phase. The *detector* is an electron multiplier responsible for recording the presence of ions as signals that represent the abundance of the analyte. *Digitizers* (analog to digital converters, ADC) transform the continuous analog detector signal into a digital, discretized spectrum by data processing. The two *ion sources* are: (1) *MALDI* (matrix assisted laser desorption and ionization), which relies on matrix molecules alpha-cyano-hydroxycinnamic acid (CHCA), sinapinic acid (SA), 2,5-dihydroxybenzoic acid (DHB), and a pulsed nitrogen UV laser ($\nu=337$ nm) and produce singly charged peptide ions; and (2) *ESI* (Electrospray ionization), which heats the needle to 40–100°C to facilitate nebulization and evaporation, and produces multiple charged peptide ions (2^+ , 3^+ , 4^+). Mass analyzers include: (a) Time-of-flight (TOF) relates m/z to the velocity of the ion, and the speed to the travel time the field-free tube length; (b) ion trap (IT) operates by effectively trapping the ions in an oscillating electrical field achieving mass separation by ejecting only ions of a specific mass; (c) quadrupole (Q) uses a combined RF AC and DC current creating a high-pass mass filter between the first two rods and a low-pass mass filter between the other two rods permitting only a specific m/z interval; (d) *orbitrap* consists of an outer and inner coaxial electrode generating an electrostatic field in which the ions form an orbitally harmonic oscillation along the axis of the field where the frequency of the oscillation is inversely proportional to the m/z , and can again be calculated by Fourier transform. *Tandem mass spectrometry* (*MS/MS*, *MS²*) is accomplished by using two mass analyzers in series (tandem). The first mass analyzer performs the function of ion selector, by selectively allowing only ions of a given m/z to pass through while the second mass analyzer is situated after triggering fragmentation and is used in its normal capacity as a mass analyzer for the fragments.

stable isotopes for mass identification and data quantification with reference to the protein sequence databases and the peptide quantitative information (Bantscheff and Kuster, 2012; Bantscheff et al., 2012). Various methods have emerged in handling different types of samples. The range of human protein abundance is from 0.01 to 10,000 ppm (Wang et al., 2012) with high abundance proteins such as extracellular matrix (ECM) in tissues masking the low abundance cellular proteins resulting in a narrow proteome coverage (Mann et al., 2013). For instance, differences of protein levels in normal and tumor tissues are measured and used to select against cancer cells that proliferate disproportionately and to identify drug targets or weak points in a cancer cell (Soldes et al., 1999a,b; Seow et al., 2001; Celis et al., 1999a,b,c, 2002; Celis and Gromov, 1999; Meehan et al., 2002; Franzen et al., 1997; Bini et al., 1997).

In sample handling (Fig. 3), storage conditions affect the serum peptidome MS profiles suggesting the importance of standardized blood sampling and storage for biobanks and interlaboratory multistudies using serum samples. Variations in proteome profiles can be avoided by standardizing sample collection considering the time between venipuncture and serum preparation that affects peptidome patterns (Tsuchida et al., 2018). Targeted in situ peptide fractionation in shotgun proteomics aims to reduce the complexity of peptide mixtures. This is achieved by selective enrichment of those digested peptides that possess specific features enabling them to bind into functionalized filter material in pipette tips. The nonbinding nontargeted peptides flow through as

unbound fraction after digestion. A good example for this is the CS-trap sample preparation method (Zougman and Banks, 2015), which uses the quartz depth filter surface functionalized with the pyridyl-dithiol group thereby allowing reversible in situ capture of the peptides with cysteines in the suspension trap-based digest (Zougman and Banks, 2015).

Human Tissue Samples

Protein analysis is carried out in two types of human tissue samples, frozen and formalin-fixed paraffin-embedded tissues (FFPE) (Wisniewski, 2013; Wisniewski *et al.*, 2013). Cancer diagnosis is generally dependent upon the FFPE with properly preserved tissue morphology and good staining. FFPE works with majority of the tissues for easy maintenance and long-term storage (Fox *et al.*, 1985). However protein profiles vary in samples that are processed differently at the preanalytical phase such as the time interval between tissue removal from the body and stabilization affecting data quality. Moreover, tissues collected from different patients exhibit variation in protein levels (Becker, 2015). Differences in protein profiles due to a disease and its progression are influenced by the procedural variables. Hence, the preanalytical phase involving sample collection, handling (transport, stabilization), storage, and analyte extraction need standardization to increase the sample quality and consistency (Fig. 3). Performing histological examination on samples prior to protein extraction to verify the cell structure and adopting an internationally standard step-by-step process of preanalytical assay are important for a systematized workflow (Becker, 2015) (Fig. 3).

Assessing tissues for cancer studies is challenging because they are comprised of cell populations that are a mixture of neoplastic and nonneoplastic cells. To overcome the difficulty of detecting indistinct signals due to the noise from surrounding cells, the laser capture tissue microdissection and cell sorting are now extensively used (Cecconi *et al.*, 2011a,b). Microanalysis handling of minute quantities of samples improves analytical performance with reduced sampling volume and faster time to obtain results (Lion *et al.*, 2004). In handling formalin-fixed paraffin-embedded (FFPE) tissues, the quality of protein profiles measured by label-free MS is less affected by the length of archival storage time for tissues, thus supporting their use in biomarker discovery and validation studies (Craven *et al.*, 2013). This will make the millions of FFPE tissues stored in biobanks a practical source of proteomic samples.

Enrichment of Low Abundance Proteins

Biological fluids contain a wide range of protein levels having the high-abundant proteins obscuring the detection of low-abundant proteins. The approach to mitigate this challenge is to remove the overabundant proteins thereby reducing the dynamic range of protein levels in the sample. Antibody-based affinity is used for selective removal of unwanted highly abundance specific proteins by binding the unwanted proteins to recover the flow-through consisting of low abundance proteins. Multiple affinity removal system is used in biological fluid proteomics to deplete abundant proteins by bundling to the carrier proteins thereby enriching detection of low abundance deep proteome (Krief *et al.*, 2012; Darde *et al.*, 2007). To address this, the affinity bound fraction is analyzed doubling the number of samples entering the downstream process (Giorgianni and Beranova-Giorgianni, 2016). Another method of reducing the level of high-abundant proteins is through the use of dynamic range compression that uses combinatorial ligand libraries in capturing and preferentially enriching low-abundant proteins (Righetti and Boschetti, 2008). This approach allows equalization of all proteins where high-abundant proteins are decreased and low-abundant proteins are boosted (Giorgianni and Beranova-Giorgianni, 2016). Another alternative technique is by using biochemical fractionation where peptides or proteins are divided into several fractions thereby reducing complexity in each fraction (Keshishian *et al.*, 2015). Sampling of body fluids for discovery and development of new biomarkers is in a form of serum/plasma, urine, cerebrospinal fluid, saliva, or bronchoalveolar lavage fluid. Prior to processing, the nonprotein elements are removed including the impurities such as cell debris, salts, lipids, and other minute molecular components. Fluids containing very low protein level undergo initial centrifugation to remove particles and contaminants followed by concentration of protein analytes and cleaning by either ultrafiltration with membrane filters or by protein precipitation with acetone or trichloroacetic acid.

Clinical Sample Processing: Electrospray Ionization and Matrix-Assisted Laser Desorption/Ionization

Proteomics by MS has two major approaches, comprehensive and targeted proteomics, which generally follow a workflow that includes sample preparation, protein extraction, protein digestion, peptide separation, mass analysis, and protein identification, validation, and biochemical characterization (Fig. 3). Comprehensive proteomics aims to identify all proteins in a sample using search algorithms like Mascot, Andromeda, SEQUEST, and so on (see Table 1). Targeted proteomics is used to program mass spectrometers into analyzing the preselected set of peptides produced by cleaving target proteins with Trypsin, LysC, or other proteases (Tsiatsiani and Heck, 2015).

Various samples, such as, cells, tissues, or body fluids, have been used in proteomics to characterize the entire protein content (Aebersold and Mann, 2016; Aebersold *et al.*, 2016). These samples are analyzed to diagnose particular diseases using an array of potential protein biomarkers measured that can indicate the incidence of outcome or disease (Ruhaak *et al.*, 2016). A promising biomarker molecule must exhibit precision, sensitivity, and reproducibility (Chahrour *et al.*, 2015) as accurately quantified by

Table 1 List of bioinformatics tools useful for clinical proteomics

Bioinformatics tools	Website/Source	Description
A-Score	https://www.ncbi.nlm.nih.gov/pubmed/16964243	A probability-based score that measures the probability of correct phosphorylation site localization based on the presence and intensity of site-determining ions in MS/MS spectra (Beausoleil <i>et al.</i> , 2006).
Andromeda	https://omictools.com/andromeda-tool	A peptide search engine based on probabilistic scoring with performance similar to Mascot that handles data with arbitrarily high fragment mass accuracy, assigns and scores complex patterns of PTM, and accommodates extremely large databases (Cox <i>et al.</i> , 2011).
APOSTL	https://github.com/bornea/APOSTL	Automated Processing of SAINT Templated Layouts (APOSTL) is a freely available Galaxy-integrated software suite and analysis pipeline for reproducible, interactive analysis of affinity proteomics (AP-MS) data using Galaxy workflows, which are intuitive for the user to move from raw data to publication-quality figures within a single interface.
Bio-TDS	http://biotds.org/	Bio-TDS (Bioscience Query Tool Discovery Systems) assists researchers in retrieving the most applicable analytic tools by allowing them to formulate their questions as free text using a flexible retrieval system that affords users from multiple bioscience domains (e.g., genomic, proteomic, bioimaging) the ability to query over 12,000 analytic tool descriptions integrated from well-established, community repositories.
CPAS	https://www.ncbi.nlm.nih.gov/pubmed/16396501	The open-source Computational Proteomics Analysis System contains an entire data analysis and management pipeline for liquid chromatography tandem mass spectrometry (LC-MS/MS) proteomics, including experiment annotation, protein database searching and sequence management, and mining LC-MS/MS peptide and protein identifications featuring general experiment annotation component, installation software, and data security management useful for collaborative projects across geographical locations and for proteomics laboratories without substantial computational support (Rauch <i>et al.</i> , 2006).
DBToolKit	http://genesis.Ugent.be/dbtoolkit/	DBToolKit allows the processing of protein sequence databases to peptide-centric sequence databases to enhance optimized search spaces for efficient identification of peptide fragmentation spectra obtained by mass spectrometry and for solving a range of other typical tasks in processing sequence databases (Martens <i>et al.</i> , 2005).
ExPASy	https://www.expasy.org/	SIB Bioinformatics Resource Portal that provides access to scientific databases and software tools in proteomics, genomics, phylogeny, systems biology, population genetics, transcriptomics (expasy.org).
gpmdb	http://gpmdb.thegpm.org	The Global Proteome Machine Database provides the information obtained by GPM servers to aid in validating peptide MS/MS spectra and protein coverage patterns, integrated into the GPM server pages allowing users to quickly compare their experimental results with the best results previously observed by other users of the machine (thegpm.org).
IDPicker	http://proteowizard.sourceforge.net/idpicker/	IDPicker views and filters peptide-spectrum-matches by reading and combining results from many different proteomic identification tools, including database search engines and spectral library matches.
InsPecT	https://bioinfo.ird.fr/index.php/resources/49-uncategorised/111-inspect	INSPECT (INtracellular ParasitE CounTer) is a free and open source software dedicated to automate quantification of <i>Leishmania</i> intracellular parasites based on fluorescent DNA images.
IntAct	https://www.ebi.ac.uk/intact/	IntAct provides a freely available, open source database system and analysis tools for molecular interaction data derived from literature curation or direct user submissions (ebi.ac.uk).
InterPro	https://www.ebi.ac.uk/interpro/	InterPro provides functional analysis of proteins by families, predicted domains and important sites with searchable resource for protein signatures from powerful integrated database and diagnostic tool (ebi.ac.uk).
itracker	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1276793/	i-Tracker software extracts reporter ion peak ratios from noncentroided tandem MS peak lists in a format easily linked to the results of protein identification tools such as Mascot and Sequest.
jPOSTTrepo	https://repository.jpostdb.org/	An international standard data repository for proteomes with a high-speed file uploading, flexible file management, and easy-to-use interfaces containing various proteomic datasets and is available for researchers worldwide.
MASCOT	http://www.matrixscience.com	A powerful search engine that integrates all of the proven methods of searching using the mass spectrometry data to identify proteins from primary sequence databases and serves as a benchmark for identification, characterization and quantitation of proteins. (matrixscience.com)
MaxQuant	http://www.biochem.mpg.de/5111795/maxquant	A quantitative proteomics software package that analyzes large-scale mass-spectrometric data sets using a set of algorithms that includes peak detection and scoring of peptides for mass calibration, database searching, protein identification, quantitation of identified proteins, and provision of summary statistics.
ModifiComb	https://www.ncbi.nlm.nih.gov/pubmed/16439352	Maps hundred types of PTM at a time including novel and unexpected PTMs.

(Continued)

Table 1 Continued

Bioinformatics tools	Website/Source	Description
MSAmanda	http://ms.imp.ac.at/?goto=msamanda	A scoring algorithm for high mass accuracy generally applicable to HCD, ETD, and CID fragmentation type data that provides free of charge both as stand alone version for integration into custom workflows and as a plugin for the Proteome Discoverer platform (Dorfer <i>et al.</i> , 2014).
msCompare	https://trac.nbic.nl/mscompare/ ,	A modular framework that allows the arbitrary combination of different feature detection/quantification and alignment/matching algorithms with a novel scoring method to evaluate their overall performance and to assess the performance of workflows built from modules of publicly available data processing packages such as SuperHirn, OpenMS, and MZmine.
mzIdentML	http://www.psdev.info/mzidentml .	Proteomics Standards Initiative mzIdentML open data standard captures the outputs of peptide and protein identification software supporting scores associated with localization of modifications on peptides, statistics performed at the level of peptides, identification of cross-linked peptides, and support for proteogenomics approaches.
neXtProt	https://www.nextprot.org)	The neXtProt human protein knowledgebase focuses on proteomics and genetic variation data for over 85% of the human proteins, as well as new tools tailored to the proteomics community including over 8000 phenotypic observations for over 4000 variations in a number of genes involved in hereditary cancers and channelopathies.
OMSSA	http://proteomicsresource.washington.edu/protocols06/omssa.php	A free search engine for analyzing and identifying peptides from tandem mass spectrometry (ms/ms) peptide spectra using the OMSSA algorithm that scores peptide hits using a probability-based method comparing experimental fragments with those calculated from libraries of known protein sequences (proteomicsresource.washington.edu).
Panther Classification System	http://pantherdb.org	The PANTHER (Protein ANALYSIS THrough Evolutionary Relationships) Classification System classifies proteins (and their genes) to facilitate high throughput analysis. The PANTHER Classifications are the result of human curation as well as sophisticated bioinformatics algorithms (pantherdb.org).
PaxDb	https://pax-db.org/about	A meta-resource of whole-organism and tissue-resolved data covering 85,000 proteins in 12 model organisms that integrates information on absolute protein abundance levels with emphasis on deep coverage, consistent postprocessing, and comparability across different organisms.
PDBe	http://www.ebi.ac.uk/pdbe/	(Protein Database Europe) European resource that provides the collection, organization and dissemination of data on macromolecular structures (ebi.ac.uk).
PEAKS	https://academic.oup.com/bioinformatics/article/23/2/243/204806	Analyzes any group of sequences that share a known reference element, such as the transcription start site, the initiation codon, a known TFBS or any other predefined site to detect any other motifs that show a significant clustering at a particular distance from the reference element, or sequence positions that show matches to motifs from a user-selected library.
PepNovo	http://proteomics.ucsd.edu/software-tools/531-2/	A high throughput de novo peptide sequencing analysis for tandem mass spectrometry (MS/MS) data that produces models of peptide fragmentation events by using a probabilistic network and a likelihood ratio hypothesis test to detect if peaks are produced under the fragmentation model or under a probabilistic model.
Peptide fragment fingerprinting	http://prowl.rockefeller.edu/prowl/pepfrag.html	PepFrag identifies proteins from a collection of sequences that matches a single tandem mass spectrum (expasy.org).
Peptide sequence database	http://pepbank.mgh.harvard.edu	Database of peptides based on sequence text mining and public peptide data sources that stores only peptides with 20 amino acids or shorter and with available sequences (pepbank.mgh.harvard.edu).
PeptideAtlas	http://www.peptideatlas.org	It is a multiorganism, publicly accessible compendium of peptides identified in a large set of tandem mass spectrometry experiments that provides output files collected from human, mouse, yeast, and several other organisms, searchable using the latest search engines and protein sequences containing the raw data, search results, and full builds (peptideatlas.org).
Peptidome (NCBI Peptide Data Resources)	http://www.ncbi.nlm.nih.gov/peptidome .	A public resource that archives and freely distributes tandem MS peptide and protein identification data generated by the scientific community, promotes sharing and dissemination of experimental data at a level of detail that is useful to both the specialized community and to the general biological community (ncbi.nlm.nih.gov).
Perseus	http://www.biochem.mpg.de/5111810/perseus	A software package for shotgun proteomics data analyses that extracts biologically meaningful information from processed raw files and performs bioinformatic analyses of the output of MaxQuant thereby completing the proteomics analysis pipeline (coxdocx.org).
Phenyx	http://www.ionsource.com/functional_reviews/Phenyx/phenyx-web.htm	Phenyx is a new sequence search engine based on the true probabilistic and flexible scoring system OLAV that uses mass spectrometry data to identify proteins and provide an automated second round search function.
PRIDE	https://www.ebi.ac.uk/pride/archive/	PRIDE database is a centralized, standards compliant, public data repository for proteomics data including protein and peptide identifications, PTM, supporting spectral evidence that provides a single point to submit MS-based proteomics data to public domain repositories (ebi.ac.uk).

Table 1 Continued

<i>Bioinformatics tools</i>	<i>Website/Source</i>	<i>Description</i>
ProDom	http://prodom.prabi.fr/prodom/current/html/home.php	ProDom is a comprehensive set of protein domain families automatically generated from the UniProt Knowledge Databases, constructed automatically by clustering homologous segments consisting of domain family entries that provides a multiple sequence alignment of homologous domains and a family consensus sequence (prodom.prabi.fr).
ProSite	https://prosite.expasy.org	Provides documentation entries describing protein domains, families and functional sites as well as associated patterns and profiles to identify them (prosite.expasy.org).
ProteinProphet	http://proteinprophet.sourceforge.net	ProteinProphet™ automatically validates protein identification made on the basis of peptides assigned to MS/MS spectra by database search programs such as SEQUEST.
ProteomicsDB	https://www.ProteomicsDB.org	A protein-centric in-memory database for the exploration of large collections of quantitative mass spectrometry-based proteomics data to enable the interactive exploration of the first draft of the human proteome.
Proteowizard	http://proteowizard.sourceforge.net	The ProteoWizard Library and Tools are a set of modular and extensible open source, cross-platform tools and software libraries that facilitate proteomics data analysis and enable rapid tool creation by providing a robust, pluggable development framework, which simplifies and unifies data file access, and performs standard chemistry and LCMS dataset computations.
PTM Score	http://www.bioinform.com/peaks-ptm/	Identifies and characterizes PTMs of proteins such as phosphorylation, ubiquitination, acetylation, and methylation that play critical roles in diverse biological processes such as signaling and regulatory processes, protein activity and degradation, regulation of gene expression essential for a comprehensive understanding of cellular biology and human diseases with a wide range of applications.
R	https://www.r-project.org	R is a free, software environment for statistical analysis, with many useful features that promote and facilitate reproducible research (r-project.org).
ROTS	https://www.bioconductor.org/packages/ROTS	The reproducibility-optimized test statistic adjusts a modified t-statistic according to the inherent properties of the data and provides a ranking of the features based on their statistical evidence for differential expression between two groups applicable in transcriptomics and proteomics.
SEQUEST	https://omictools.com/sequest-tool	SEQUEST provides fragmentation patterns in tandem mass spectra to detect AAS from protein and nucleotide database, correlates the spectrum and allows searches with the experimental data via the prediction of fragment ions of an AAS and offers to users the ability to correlate exactly uninterpreted tandem mass spectra to sequences in the database (omictools.com).
Skyline	http://proteome.gs.washington.edu/software/skyline	An open source and freely available windows client application for targeted proteomics method creation and quantitative data analysis that simplifies the development of mass spectrometer methods and the analysis of data from targeted proteomics experiments performed using SRM.
SMART	http://smart.embl-heidelberg.de	SMART (Simple Modular Architecture Research Tool) allows the identification and annotation of genetically mobile domains and the analysis of domain architectures (embl-heidelberg.de).
Spectrum Mill	https://www.agilent.com/en/products/software-informatics/masshunter-suite/masshunter-for-life-science-research/spectrum-mill	Spectrum Mill for MassHunter Workstation quickly identifies proteins and peptides through fast database searches with automatic or manual match validation and unique algorithms that minimize false positives, offers de novo spectral interpretation for proteins not found in databases, identifies relative abundance differences of two-fold or greater without complicated isotope labeling, and summarizes results in formats that provide maximum insight and convenience.
SWISS-PROT	https://web.expasy.org/docs/swiss-prot_guideline.html	Curated protein sequence database that provides a high level of annotation, minimal level of redundancy and high level of integration with other databases (Bairoch <i>et al.</i> , 2004).
TOPP	http://ftp.mi.fu-berlin.de/pub/OpenMS/release-documentation/html/index.html	TOPP (The OpenMS Proteomics Pipeline) provides a set of computational tools combined into easy analysis pipelines for nonexperts of proteomics workflows with useful utilities file format conversion, peak picking, and supports known applications (e.g., Mascot) to generate new algorithmic techniques for data reduction and data analysis.
TPP	http://tools.proteomecenter.org/wiki/index.php?title=Software:TPP	The Trans-Proteomic Pipeline is a collection of integrated tools for MS/MS proteomics developed at the SPC.
UniProt (universal Protein Resource)	http://www.uniprot.org	A comprehensive resource for protein sequence and annotation data that provides the scientific community with a comprehensive, high quality and freely accessible resource of protein sequence and functional information (Bairoch <i>et al.</i> , 2005) (uniprot.org).
X!Tandem	http://www.thegpm.org/TANDEM	X! Tandem open source is software that can match tandem mass spectra with peptide sequences in protein identification.
ZBIT Bioinformatics Toolbox	https://webservices.cs.uni-tuebingen.de/	The Center for Bioinformatics Tuebingen (ZBIT) Bioinformatics Toolbox provides web-based access to a collection of bioinformatics tools developed for systems biology, protein sequence annotation, and expression data analysis (Romer <i>et al.</i> , 2016).

proteomic technologies. Electrospray ionization (ESI) and matrix-assisted laser desorption/ionization (MALDI) are the two main technologies for sample processing in MS that focus on quantitative proteomics analysis. ESI processes neutral compounds to generate ions that are detected as mass/charge (m/z) ratio of each peptide, which is recorded by a computer and shown as a mass spectrum (Ho *et al.*, 2003). MALDI on the other hand uses a laser beam to light up a solid sample into an ionized form that travels through the mass spectrometer, detected and translated into the mass/charge (m/z) measurements (Andersson *et al.*, 2010; Duncan *et al.*, 2016). Variability of outcome in protein measurements depends on sample handling whereby protein denaturation, incomplete digestions, desalting, and depletion failures, which cause loss of pertinent data, need to be avoided (Buchler *et al.*, 2016). Both the molecular labeling method used to tag samples for quantification and the resolution capacity of MS also influence results, and thus need consideration early on in the experimental design prior to measurements (Tsuchida *et al.*, 2018). Quantification of targeted proteomics depends on selected or multiple reaction monitoring, which requires complex statistical analysis, and validation of the result is laborious. Several groups endeavor to improve the validation by developing the Sequential Windowed Acquisition of All Theoretical Fragment Ion Mass Spectra in combination with a nontargeted acquisition or combining different ion sources without any interference in a single mass spectrometer (Kostyukevich and Nikolaev, 2018).

Bottom-Up and Top-Down Proteomics

Proteomics is divided into two types of analysis based on whether the measurements are made on broken-to-pieces proteins (bottom-up proteomics) and intact proteins (top-down proteomics). Although top-down proteomics is advantageous for examining protein modifications, it is not as well-established as the bottom-up approach, facing more technical challenges as intact proteins are more complicated to handle with their poor solubility than peptides. Bottom-up or shotgun proteomics is easier to fractionate in liquid chromatography and easier to analyze in tandem mass spectrometer (LC-MS/MS) due to their smaller size and easier transfer to MS through ESI. Using proteases, proteins are first converted to small pieces of peptides, into which measurements are performed. Proteins are digested with trypsin that cleaves after every lysine and arginine residues resulting in truncated peptides of around 6–25 amino acids, although other proteases are also used. MS is then applied to measure the mass-to-charge ratio of each peptide in mixtures of thousands of peptides to determine the amino acid sequence. There are four stages of the process, consisting of sample preparation, separation of peptides, MS, and bioinformatics analysis of the data (Fig. 3). The critical factor in this approach is the sample preparation that requires protein fractionation or depletion methodologies to isolate low-abundant proteins. Peptide matches also need examination and manual validation for a series of consecutive sequence-specific b-type and y-type ions. Moreover, the correct peptide sequence and accurate quantification of proteins are achieved by using a newer generation of mass spectrometers with higher resolution. In targeted LC-MS/MS, a set of phenotypic/signature peptides containing the target protein sequences are selected using bioinformatics tools such as UniProt Database, ExPASy, PeptideAtlas, and so on (Table 1). The selected peptides are then synthesized as labeled internal standards (absolute quantification AQUA (Gerber *et al.*, 2007, 2003), QconCAT (Beynon *et al.*, 2014), PSAQ (Dupuis *et al.*, 2008)), chromatographed, eluted by ESI to MS, detected, and measured in available platform of LC-MS/MS (Fig. 3). Targeted MS/MS instruments include quadrupole-Orbitraps and quadrupole time-of-flight with high resolution of 140–240,000 and 100,000, and with mass accuracy of 2 ppm and 2–5 ppm, respectively.

Top-down proteomics identifies intact proteins containing PTMs including the truncated proteoforms translated from reference ORFs or alternative open reading frames (altORFs) increasing the size of the actual proteome. Top-down proteomics can provide the whole sequence information of the proteoforms useful in analyzing various structural modifications due to variations in genes, RNA, and proteins (Fig. 2). Top-down mass spectral database searches that align millions of spectra against thousands of protein sequences in high throughput proteome level analyses have been made available recently (Leduc *et al.*, 2018; Kou *et al.*, 2018). In MALDI-imaging or MALDI-IMS, proteins including their molecular arrangement in samples are analyzed. It can scan a broad range of protein from a tissue section at defined geometrical coordinates producing an ion-density map. A fresh-frozen tissue section is used resembling the real tissue where the relative abundance of the selected ion is displayed on a color-intensity scale at each coordinate location (pixel) (Seeley *et al.*, 2008). MALDI-IMS has limited mass range that detects only small proteins thus needing further enhancement. MALDI-TOF-MS provides a rapid technique to identify microorganisms useful in pathogenic bacteria taxonomy, clinical microbiology, and food safety (Fagerquist, 2017).

An important step in proteomics is validation that is needed in translating research into the clinic. Confirmation by Western blotting excludes false-positive results. The functional significance of the molecules involved is verified by *in vitro* or *in vivo* assays, which demonstrate inhibition or activation of the pathway followed by analysis of its biological effects. *In situ* validation uses antibodies to detect isoforms such as phosphorylated proteins at specific locations. Several antibodies are used to validate the various components of a pathway (Cecconi *et al.*, 2011b).

Personalized Proteomics and Precision Medicine

Currently drug prescription for patients is based on general information about what generally works. If the medication is not working then the patient is switched to another medicine based on trial and error, which sometimes results in adverse drug responses making patients suffer from side effects of unnecessary or inappropriate use of medications. This approach to treatment

is now changing with the progress made in personalized medicine. P4 medicine (preventive, predictive, personalized, and participatory) tailors medical treatment to the individual characteristics, needs, and preferences of a patient during all stages of care, including prevention, diagnosis, treatment, and follow-up (Tian *et al.*, 2012; Van Karnebeek *et al.*, 2018). The patient's treatment is catered upon their individual genetic makeup, entire genetic profile, and personalized proteomics data (Duarte and Spencer, 2016). Proteomics studies open the door for the clinicians to understand diseases at protein levels. A collection of protein data in combination with the DNA information provides insights into the development of therapies. Proteomics connects the gap between diagnostics and therapeutics by facilitating detection of protein biomarkers (Jain, 2016). Pharmaceutical industries use clinical proteomics to determine relevant drug targets, which are typically proteins that play many important roles in the metabolic pathways (Snyder *et al.*, 2014). The FDA and other agencies also request protein profiles in new drug development, attracting high interest for both for the healthcare sector, as well as the pharmaceutical and biotech industry (Szasz *et al.*, 2017; Marko-Varga and Labaer, 2017).

To illustrate the potential of proteomic analysis, a study was conducted in bladder cancer using different proteomic platforms. In bladder cancer cases, the degree of tumor spread into the bladder wall is classified into three different categories, namely non-muscle invasive, muscle invasive, and metastatic disease (Babjuk *et al.*, 2017). The monitoring device is invasive cystoscopy, which uses a thin camera to look inside the bladder of patients. However, having proteomics profiling of urine and tissues decreases the dependency on invasive cystoscopy improving bladder cancer patient management. In this example, the negative result avoids the unnecessary cystoscopy invasion while the positive result points to a more thorough examination. Tissue molecular profiling further revealed that bladder cancer is a highly heterogeneous disease with genetic alterations found in the adjoining tissues in bladder cancer patients (Thomsen *et al.*, 2017). Detailed assessment of molecular alterations indicates the direct site of disease initiation and progression thus is beneficial in identifying potential therapeutic targets and prediction of treatment response. Urine proteome profiling enables diagnosis, monitoring, and stratification by looking at the gradual change of the amount of urinary peptides as the cancer advances (Latosinska *et al.*, 2017a,b).

Several studies have utilized proteomics to identify biomarkers. For example, endogenous peptide of pleiotrophin was identified as a new biomarker of Alzheimer's disease (Blennow and Zetterberg, 2015). A group of Japanese researchers have also identified leucine-rich alpha-2 glycoprotein as a potential biomarker to diagnose Kawasaki disease based on proteomics analysis (Kimura *et al.*, 2017). High throughput verification of hundreds of biomarker candidate proteins to detect early stage colorectal cancer was done using targeted proteomics resulting in identification of the annexin family of proteins as promising colorectal cancer biomarker candidates (Shiromizu *et al.*, 2017). With the improvement and enhancements made in the proteomic technologies, workflows, computational tools, instruments, and availability of data obtained from other "omics" analysis, personalized medicine is now becoming possible although there are still limitations that need to be mitigated. Proteomics data are useful in designing a treatment plan distinct to the condition the individual experiences.

Proteform Bioinformatics

The Human Proteome Project (HPP) (see "Relevant Websites section") was created through the coordinated efforts of many research laboratories around the world with the goal of mapping the entire human proteome (Omenn *et al.*, 2017). It has two main goals: The comprehensive and systematic analysis of the human proteome map and the creation of analytical tools to measure proteins relevant in health and disease (Segura *et al.*, 2017). In 2014, two independent groups published the first draft of human proteome using many different tissues by MS, thereby, putting proteomics into the spotlight and laying a foundation for development of diagnostic, prognostic, therapeutic, and preventive medical applications. The advances in biology using technologies that produce high throughput data have fueled the development of systems biology. ProteomicsDB (see "Relevant Websites section") enables interactive use of the human proteome and its large quantitative MS database (Schmidt *et al.*, 2018). Powerful bioinformatics tools are much needed to enable new applications of proteomics data in precision medicine and in the clinics. JS-MS is a stand-alone cross-platform that enables 3-D visualization for MS data processing and evaluation (Rosen *et al.*, 2017).

The SWISS-PROT is an annotated protein sequence database created by A. Bairoch on 21 July 1986 that integrates with other databases at that time including EMBL Nucleotide Sequence Database, PDB (Protein Data Bank), PIR (Protein Identification Resource), HIV (the human retroviruses and AIDS Database), OMIM (the online version of the book *Mendelian Inheritance in Man*), PROSITE (the Dictionary of Protein Sites and Patterns), and REBASE (the database of type 2 restriction enzymes) (Bairoch and Boeckmann, 1991). The first software developed for 2DE image analysis in 1979 was ELSIE, which was followed by a vast array of current proteomics tools such as ExPASy, IntAct, InterPro, Matrix Science, OMSSA, Panther Classification System, PDBe (Protein Database Europe), PeptideAtlas, Peptidome (NCBI Peptide Data Resources), PRIDE, ProDom, ProSite, thegpmdb, SEQUEST, UniProt (universal Protein Resource), R, and SMART among many others (see Table 1). The neXtProt is a protein knowledgebase (see "Relevant Websites section") that focuses on proteomics and genetic variation data for over 85% of the human proteins, includes new proteomics tools and over 8000 phenotypic observations for over 4000 variations in genes involved in hereditary cancers and channelopathies, and provides an API access and a SPARQL endpoint for technical applications (Gaudet *et al.*, 2017).

Clinical proteomics involves a complex process that starts with sample extraction and processing, followed by protein and peptide fractionation, MS measurement, and finally protein identification and quantification (Fig. 3) (Mallick and Kuster, 2010).

The synergy of the high throughput MS-based proteomics data along with the network-based computational analysis has facilitated the investigation of the cellular networks associated with certain disease important to basic and clinical research (Schmidt *et al.*, 2014). Bioinformatics has provided the tools and methods necessary for processing genetic information from patients integrating the clinical, environmental and genetic data. It has played a vital role in generating, storing, and curating protein sequence databases, developing relevant software for the proteomics data analysis, integrating protein interaction networks, and consolidating data from other “omics” to produce a meaningful biological information (Felgueiras *et al.*, 2018; Uppal *et al.*, 2018).

Several steps are involved in bioinformatics, namely: (1) Collecting statistics from biological data, (2) building a computational model, (3) solving a computational modeling problem, and (4) testing and evaluating computational algorithm (Can, 2014). A successful analysis outcome is obtained by integrating proteomics data with other “omics” data using data-driven integration and network analysis tool such as xMWAS, for example (Uppal *et al.*, 2018). Various repositories have also been created such as PeptideAtlas (Desiere *et al.*, 2006; Deutsch *et al.*, 2005; Schwenk *et al.*, 2017), GPMDB (Craig *et al.*, 2006), PRoteomics IDentifications (PRIDE) (see “Relevant Websites section”) (Martens *et al.*, 2005; Vizcaino *et al.*, 2016a,b), UniProt (The Uniprot Consortium, 2018), and neXtProt (Gaudet *et al.*, 2017) among many others that store proteomics experimental information. These repositories were designed to be made publicly available and can be directly used or reused in producing novel findings, which would generate new knowledge (Jarnuczak and Vizcaino, 2017; Vizcaino *et al.*, 2017; Okuda *et al.*, 2017; Chen *et al.*, 2017; Keerthikumar and Mathivanan, 2017). These data can be used for investigation and mining to extract relevant, usable, and quality information for clinical application (Islam *et al.*, 2017). To assess the key steps of MS data analysis process, database search engines, and methods in quality control (QC), protein assembly and data integration, the entrapment sequence method has been used as the standard (Feng *et al.*, 2017).

The HPP provides MS and antibody-based data on all human proteins from various physiological and pathological conditions (Omenn *et al.*, 2017). For 15 years, the Proteomics Standards Initiative of the Human Proteome Organization (HUPO) has provided guidelines for sharing data sets to the public establishing open community standards and software tools for proteomics such as PSI-MI, XML, MITAB, mzML, mzIdentML, mzQuantML, mzTab, and the MIAPE (Minimum Information About a Proteomics Experiment) (Deutsch *et al.*, 2017). One of its key aspects is the use of standardized and well-defined terminology to complete the values in the data files (Mayer *et al.*, 2014). Moreover, two novel standard data formats (proBed and proBAM) for proteogenomics were recently developed with functionalities in file generation, file conversion, and data analysis (Menschaert *et al.*, 2018) improving the integration of genomics and proteomics information and the evaluation of peptide spectrum matches (PSM). An easy-to-use tool proBAMconvert (see “Relevant Websites section”) allows conversion of mzIdentML, mzTab, and pepXML file formats to proBAM/probed or output at PSM and peptide levels using a command line and graphical user interface (Olexiouk and Menschaert, 2017). Furthermore, several ongoing analytical designs are nearly completed including formats for a spectral library standard, a universal spectrum identifier, the qcML quality control, and the Protein Expression Interface web services Application Programming Interface in synergy with ProteomeXchange Consortium, HPP, and other “omics” community (Deutsch *et al.*, 2017). Additional initiatives such as multiorganism proteome (iMOP) (Tholey *et al.*, 2017), the Human Cancer Proteome Project (Cancer-HPP) (Jimenez *et al.*, 2018), the Human Eye Proteome Project (EyeOme) (Ahmad *et al.*, 2018), Chromosome-Centric Human Proteome Project (C-HPP) (Meyfour *et al.*, 2018), Human Immuno-Peptidome Project (Caron *et al.*, 2017), fecal proteome (Jin *et al.*, 2017) among many others are examples that support HPP in proteome research aimed at improving the human health using multiparameter diagnostics applicable for personalized medicine.

The Human Protein Atlas provides protein localization and expression data in human tissues and cells (Thul and Lindskog, 2018). It has a collection of antibodies to map the entire human proteome by immunohistochemistry and immunocytochemistry generating more than 10 million images of protein expression patterns at a single-cell level (see “Relevant Websites section”) (Thul and Lindskog, 2018). These resources can be freely accessed allowing scientists and clinicians to explore the human proteome. Public data sharing used to be problematic but is now becoming feasible encouraged by the scientific journals and funding agencies as part of good scientific practice (Spidlen and Brinkman, 2018) and enhanced by the availability of user-friendly online resources and tools (Perez-Riverol *et al.*, 2016a,b). To bypass difficulty of data sharing due to variable types of data formats, the PSI of the HUPO has lead the development of standard file formats and controlled vocabularies (Vizcaino *et al.*, 2017; Bittremieux *et al.*, 2017). A popular format is mzML, which stores raw MS data and processed peak list spectra (Martens *et al.*, 2011). The freely available pymzML (see “Relevant Websites section”) has optimized source code and data retrieval algorithms, provided faster random access to compressed files and quicker numerical calculations, suitable for large file sizes having a versatile scripting language (Kosters *et al.*, 2018). Another format is mzIdentML for peptide and protein identifications obtained from MS data that allows various software tools to communicate with each other (Vizcaino *et al.*, 2017). Moreover, the mzQuantML data standard format records detailed quantitative information (Walzer *et al.*, 2013) from a variety of popular discovery based workflows such as SILAC or dimethyl or MS2 tag based TMT or iTRAQ. This format is now able to store results from targeted techniques such as selected/parallel reaction monitoring (SRM). The ProteoWizard’s msConvert and msConvertGUI softwares are useful in converting data from vendor-specific propriety binary files generated by diverse mass spectrometers to open-format files (Adusumilli and Mallick, 2017). Recently a simpler text-based format called msTab was produced that stores both identification and quantification information (Griss *et al.*, 2014). It follows a tabular design making it simpler to load into a spreadsheet. MzTab is now supported as part of popular analysis tools such as PRIDE, MassIVE, and jPOST. The PRIDE Inspector Toolsuite is a freely available modular set of open-source cross-platform libraries for handling and visualizing different experimental output files including spectra (mzML, mzXML), peptide and protein identification data (mzIdentML, PRIDE XML, mzTab), and quantification data (mzTab, PRIDE XML) (see “Relevant Websites section”) (Perez-Riverol *et al.*, 2016b).

With open-access data, QC of proteomics has been given more importance and attention recently. Relevant software tools for performing QC are now available, which assess the consistency of the data submission in a form of correspondence between the mass spectra and identification results, detecting clear annotation errors, ensuring acceptable level of technical and biological metadata. Users of Pride Inspector, a freely available tool to browse MS proteomic data are also encouraged to flag any potential errors detected.

Bioinformatics plays a key role in various aspects of proteomics analysis including peptide-spectra matching (PSM) based on the new data-independent acquisition paradigm, resolving missing proteins, dealing with biological and technical heterogeneity in data and statistical feature selection. Further work is still required to enhance the proteomic analysis, performance, and reproducibility so that it can be applied in clinical medicine. More powerful bioinformatics tools with greater accuracy and resolution, faster protein identification algorithms, reduced false positives, and automated data validation capabilities that allow the consolidation of results from multiple studies will enable deeper understanding of the full proteome map of tissues or cells. Linking various omics is important to achieve proper interpretation of complex biological systems.

Conclusion

Examining the proteome is likened to getting a “snapshot” of the total protein environment at any given time point to have various insight into what is happening in the cell and how the cells are working. Interaction among proteins is a dynamic process that changes quickly over time but can be captured in proteome analysis to predict the underlying causes of diseases. The capability of time course proteomics data as a monitoring process is confirmed in several experiments such as the analysis of dynamic responses to ER stress in HeLa cell proteome (Cheng *et al.*, 2016), the discovery of different features of human metabolism in long-term space travel simulation experiments (Binder *et al.*, 2014), and the measurements of proteome dynamics in pathological cardiac hypertrophy model (Lau *et al.*, 2018) among many others. The proteome map illustrates beneficial information about the patients’ well being and the effectiveness of any drug administration for treatment. Through proteomics, molecules that affect drug target identification such as proteoforms and all modified proteins are characterized to develop more effective novel drugs with fewer side effects (Kuhlmann *et al.*, 2018; Daher and De Groot, 2018). Quantitative analysis of these proteoforms has been anticipated to deliver the next generation of multiparametric biomarkers applicable in personalized medicine (Mueller *et al.*, 2018). But approval of these biomarkers in the clinic requires a tremendous amount of work involving standardized collection of samples from patients towards simplifying the grand scale of the collected protein information into something that can easily be interpreted as useful by health care professionals (Schmit *et al.*, 2017; Nedelkov, 2017). Although faced with a lot of challenges, it is just a matter of time before the application of proteomics in personalized medicine by proteome profiling of an individual’s condition is achieved in a clinical setting. The use of bioinformatics in clinical proteomic research has not been exploited extensively as most have favored the wet lab based activities (Akhter and Shehu, 2018). However, big proteomics data generated would certainly bring benefits to the clinical environment particularly in the understanding of the molecular origins of pathologies and drug mechanisms (Trindade *et al.*, 2018).

See also: Disease Biomarker Discovery. Drug Repurposing and Multi-Target Therapies. Experimental Platforms for Extracting Biological Data: Mass Spectrometry, Microarray, Next Generation Sequencing. Identification and Extraction of Biomarker Information. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis

References

- Adusumilli, R., Mallick, P., 2017. Data conversion with ProteoWizard msConvert. *Methods in Molecular Biology* 1550, 339–368.
- Aebersold, R., Bensimon, A., Collins, B.C., Ludwig, C., Sabido, E., 2016. Applications and developments in targeted proteomics: From SRM to DIA/SWATH. *Proteomics* 16, 2065–2067.
- Aebersold, R., Mann, M., 2016. Mass-spectrometric exploration of proteome structure and function. *Nature* 537, 347–355.
- Ahmad, M.T., Zhang, P., Dufresne, C., Ferrucci, L., Semba, R.D., 2018. The human eye proteome project: Updates on an emerging proteome. *Proteomics* 18 (5–6), e1700394.
- Akhter, N., Shehu, A., 2018. From extraction of local structures of protein energy landscapes to improved decoy selection in template-free protein structure prediction. *Molecules* 23.
- Andersson, M., Andren, P., Caprioli, R.M., 2010. MALDI imaging and profiling mass spectrometry in neuroproteomics. In: Alzate, O. (Ed.), *Neuroproteomics. FSL*. Boca Raton.
- Babjuk, M., Böhle, A., Burger, M., *et al.*, 2017. EAU guidelines on non-muscle-invasive urothelial carcinoma of the bladder: Update 2016. *European Urology* 71, 447–461.
- Bairoch, A., Apweiler, R., Wu, C.H., *et al.*, 2005. The Universal Protein Resource (UniProt). *Nucleic Acids Research* 33, D154–D159.
- Bairoch, A., Boeckmann, B., Ferro, S., Gasteiger, E., 2004. Swiss-Prot: Juggling between evolution and stability. *Briefing in Bioinformatics* 5, 39–55.
- Bairoch, A., Boeckmann, B., 1991. The SWISS-PROT protein sequence data bank. *Nucleic Acids Research* 19 (Suppl.), 2247–2249.
- Bantscheff, M., Kuster, B., 2012. Quantitative mass spectrometry in proteomics. *Analytical and Bioanalytical Chemistry* 404, 937–938.
- Bantscheff, M., Lemeier, S., Savitski, M.M., Kuster, B., 2012. Quantitative mass spectrometry in proteomics: Critical review update from 2007 to the present. *Analytical and Bioanalytical Chemistry* 404, 939–965.
- Beausoleil, S.A., Villen, J., Gerber, S.A., Rush, J., Gygi, S.P., 2006. A probability-based approach for high-throughput protein phosphorylation analysis and site localization. *Nature Biotechnology* 24, 1285–1292.
- Becker, K.F., 2015. Using tissue samples for proteomic studies-critical considerations. *Proteomics Clinical Applied* 9, 257–267.
- Beynon, R.J., Armstrong, S.D., Gomez-Baena, G., *et al.*, 2014. The complexity of protein semiochemistry in mammals. *Biochemical Society Transactions* 42, 837–845.

- Binder, H., Wirth, H., Arakelyan, A., *et al.*, 2014. Time-course human urine proteomics in space-flight simulation experiments. *BMC Genomics* 15 (Suppl. 12), S2.
- Bini, L., Magi, B., Marzocchi, B., *et al.*, 1997. Protein expression profiles in human breast ductal carcinoma and histologically normal tissue. *Electrophoresis* 18, 2832–2841.
- Bittremieux, W., Walzer, M., Tenzer, S., *et al.*, 2017. The human proteome organization-proteomics standards initiative quality control working group: Making quality control more accessible for biological mass spectrometry. *Analytical Chemistry* 89, 4474–4479.
- Blennow, K., Zetterberg, H., 2015. Understanding biomarkers of neurodegeneration: Ultrasensitive detection techniques pave the way for mechanistic understanding. *Nature Medicine* 21, 217–219.
- Buchler, R., Wendler, S., Muckova, P., Grosskreutz, J., Rhode, H., 2016. The intricacy of biomarker complexity-the identification of a genuine proteomic biomarker is more complicated than believed. *Proteomics Clinical Applications* 10, 1073–1076.
- Can, T., 2014. Introduction to bioinformatics. *Methods in Molecular Biology* 1107, 51–71.
- Caron, E., Aebersold, R., Banaei-Esfahani, A., Chong, C., Bassani-Sternberg, M., 2017. A case for a Human Immuno-Peptidome Project Consortium. *Immunity* 47, 203–208.
- Carvalho, A.S., Penque, D., Matthiesen, R., 2015. Bottom up proteomics data analysis strategies to explore protein modifications and genomic variants. *Proteomics* 15, 1789–1792.
- Cecconi, D., Lonardoni, F., Favretto, D., *et al.*, 2011a. Changes in amniotic fluid and umbilical cord serum proteomic profiles of fetuses with intrauterine growth retardation. *Electrophoresis* 32, 3630–3637.
- Cecconi, D., Palmieri, M., Donadelli, M., 2011b. Proteomics in pancreatic cancer research. *Proteomics* 11, 816–828.
- Celis, J.E., Celis, P., Ostergaard, M., *et al.*, 1999a. Proteomics and immunohistochemistry define some of the steps involved in the squamous differentiation of the bladder transitional epithelium: A novel strategy for identifying metaplastic lesions. *Cancer Research* 59, 3003–3009.
- Celis, J.E., Celis, P., Palsdottir, H., *et al.*, 2002. Proteomic strategies to reveal tumor heterogeneity among urothelial papillomas. *Molecular and Cellular Proteomics* 1, 269–279.
- Celis, J.E., Gromov, P., 1999. 2D protein electrophoresis: Can it be perfected? *Current Opinion in Biotechnology* 10, 16–21.
- Celis, J.E., Ostergaard, M., Rasmussen, H.H., *et al.*, 1999b. A comprehensive protein resource for the study of bladder cancer: <http://biobase.dk/cgi-bin/celis>. *Electrophoresis* 20, 300–309.
- Celis, J.E., Wolf, H., Ostergaard, M., 1999c. Proteomic strategies in bladder cancer. *IUBMB Life* 48, 19–23.
- Chahrouh, O., Cobice, D., Malone, J., 2015. Stable isotope labelling methods in mass spectrometry-based quantitative proteomics. *Journal of Pharmaceutical and Biomedical Analysis* 113, 2–20.
- Chen, C., Huang, H., Wu, C.H., 2017. Protein bioinformatics databases and resources. *Methods in Molecular Biology* 1558, 3–39.
- Cheng, Z., Rendleman, J., Vogel, C., 2016. Time-course proteomics dataset monitoring HeLa cells subjected to DTT induced endoplasmic reticulum stress. *Data in Brief* 8, 1168–1172.
- Cox, J., Neuhauser, N., Michalski, A., *et al.*, 2011. Andromeda: A peptide search engine integrated into the MaxQuant environment. *Journal of Proteome Research* 10, 1794–1805.
- Craig, R., Cortens, J.C., Fenyo, D., Beavis, R.C., 2006. Using annotated peptide mass spectrum libraries for protein identification. *Journal of Proteome Research* 5, 1843–1849.
- Craven, R.A., Cairns, D.A., Zougman, A., *et al.*, 2013. Proteomic analysis of formalin-fixed paraffin-embedded renal tissue samples by label-free MS: Assessment of overall technical variability and the impact of block age. *Proteomics Clinical Applications* 7, 273–282.
- Daher, A., De Groot, J., 2018. Rapid identification and validation of novel targeted approaches for Glioblastoma: A combined ex vivo-in vivo pharmaco-omic model. *Experimental Neurology* 299, 281–288.
- Darde, V.M., Barderas, M.G., Vivanco, F., 2007. Depletion of high-abundance proteins in plasma by immunoaffinity subtraction for two-dimensional difference gel electrophoresis analysis. *Methods in Molecular Biology* 357, 351–364.
- Desiere, F., Deutsch, E.W., King, N.L., *et al.*, 2006. The PeptideAtlas project. *Nucleic Acids Research* 34, D655–D658.
- Deutsch, E.W., Eng, J.K., Zhang, H., *et al.*, 2005. Human Plasma PeptideAtlas. *Proteomics* 5, 3497–3500.
- Deutsch, E.W., Orchard, S., Binz, P.A., *et al.*, 2017. Proteomics standards initiative: Fifteen years of progress and future work. *Journal of Proteome Research* 16, 4288–4298.
- Dorfer, V., Pichler, P., Stranzl, T., *et al.*, 2014. MS Amanda, a universal identification algorithm optimized for high accuracy tandem mass spectra. *Journal of Proteome Research* 13, 3679–3684.
- Duarte, T.T., Spencer, C.T., 2016. Personalized proteomics: The future of precision medicine. *Proteomes* 4.
- Duncan, M.W., Nedelkov, D., Walsh, R., Hattan, S.J., 2016. Applications of MALDI mass spectrometry in clinical chemistry. *Clinical Chemistry* 62, 134–143.
- Dupuis, A., Hennekinne, J.A., Garin, J., Brun, V., 2008. Protein Standard Absolute Quantification (PSAQ) for improved investigation of staphylococcal food poisoning outbreaks. *Proteomics* 8, 4633–4636.
- Fagerquist, C.K., 2017. Unlocking the proteomic information encoded in MALDI-TOF-MS data used for microbial identification and characterization. *Expert Review of Proteomics* 14, 97–107.
- Felgueiras, J., Silva, J.V., Fardilha, M., 2018. Adding biological meaning to human protein-protein interactions identified by yeast two-hybrid screenings: A guide through bioinformatics tools. *Journal of Proteomics* 171, 127–140.
- Feng, X.D., Li, L.W., Zhang, J.H., *et al.*, 2017. Using the entrapment sequence method as a standard to evaluate key steps of proteomics data analysis process. *BMC Genomics* 18, 143.
- Fox, C.H., Johnson, F.B., Whiting, J., Roller, P.P., 1985. Formaldehyde fixation. *Journal of Histochemistry and Cytochemistry* 33, 845–853.
- Franzen, B., Linder, S., Alaiya, A.A., *et al.*, 1997. Analysis of polypeptide expression in benign and malignant human breast lesions. *Electrophoresis* 18, 582–587.
- Gaudet, P., Michel, P.A., Zahn-Zabal, M., *et al.*, 2017. The neXtProt knowledgebase on human proteins: 2017 update. *Nucleic Acids Research* 45, D177–D182.
- Gerber, S.A., Kettenbach, A.N., Rush, J., Gygi, S.P., 2007. The absolute quantification strategy: Application to phosphorylation profiling of human separase serine 1126. *Methods in Molecular Biology* 359, 71–86.
- Gerber, S.A., Rush, J., Stemman, O., Kirschner, M.W., Gygi, S.P., 2003. Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. *Proceedings of the National Academy of Sciences of the United States of America* 100, 6940–6945.
- Geyer, P.E., Holdt, L.M., Teupser, D., Mann, M., 2017. Revisiting biomarker discovery by plasma proteomics. *Molecular Systems Biology* 13, 942.
- Giorgianni, F., Beranova-Giorgianni, S., 2016. Phosphoproteome discovery in human biological fluids. *Proteomes* 4.
- Griss, J., Jones, A.R., Sachsenberg, T., *et al.*, 2014. The mzTab data exchange format: Communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience. *Molecular and Cellular Proteomics* 13, 2765–2775.
- He, Q.Y., Chiu, J.F., 2003. Proteomics in biomarker discovery and drug development. *Journal of Cellular Biochemistry* 89, 868–886.
- Ho, C.S., Lam, C.W., Chan, M.H., *et al.*, 2003. Electrospray ionisation mass spectrometry: Principles and clinical applications. *Clinical Biochemist Reviews* 24, 3–12.
- Hughes, C.S., Foehr, S., Garfield, D.A., *et al.*, 2014. Ultrasensitive proteome analysis using paramagnetic bead technology. *Molecular Systems Biology* 10, 757.
- Islam, M.T., Mohamedali, A., Ahn, S.B., *et al.*, 2017. A systematic bioinformatics approach to identify high quality mass spectrometry data and functionally annotate proteins and proteomes. *Methods in Molecular Biology* 1549, 163–176.
- Jain, K.K., 2016. Role of proteomics in the development of personalized medicine. In: *Advances in Protein Chemistry and Structural Biology* 102, pp. 41–52.
- Jarnuczak, A.F., Vizcaino, J.A., 2017. Using the PRIDE database and ProteomeXchange for submitting and accessing public proteomics datasets. *Current Protocols in Bioinformatics* 59, 13.31.11–13.31.12.
- Jimenez, C.R., Zhang, H., Kinsinger, C.R., Nice, E.C., 2018. The cancer proteomic landscape and the HUPO Cancer Proteome Project. *Clinical Proteomics* 15, 4.
- Jin, P., Wang, K., Huang, C., Nice, E.C., 2017. Mining the fecal proteome: From biomarkers to personalised medicine. *Expert Review of Proteomics* 14, 445–459.

- Keerthikumar, S., Mathivanan, S., 2017. Proteomic data storage and sharing. *Methods in Molecular Biology* 1549, 5–15.
- Kelleher, N.L., Thomas, P.M., Ntai, I., Compton, P.D., Leduc, R.D., 2014. Deep and quantitative top-down proteomics in clinical and translational research. *Expert Review of Proteomics* 11, 649–651.
- Keshishian, H., Burgess, M.W., Gillette, M.A., *et al.*, 2015. Multiplexed, quantitative workflow for sensitive biomarker discovery in plasma yields novel candidates for early Myocardial Injury. *Molecular and Cellular Proteomics* 14, 2375–2393.
- Kimura, Y., Yanagimachi, M., Ino, Y., *et al.*, 2017. Identification of candidate diagnostic serum biomarkers for Kawasaki disease using proteomic analysis. *Scientific Reports* 7, 43732.
- Kosters, M., Leufken, J., Schulze, S., *et al.*, 2018. pymzML v2.0: Introducing a highly compressed and seekable gzip format. *Bioinformatics*.
- Kostyukevich, Y.I., Nikolaev, E.N., 2018. Ion source multiplexing on a single mass spectrometer. *Analytical Chemistry* 90 (5), 3576–3583.
- Kou, Q., Wu, S., Liu, X., 2018. Systematic evaluation of protein sequence filtering algorithms for proteoform identification using top-down mass spectrometry. *Proteomics* 18 (3–4).
- Krief, G., Deutsch, O., Zaks, B., *et al.*, 2012. Comparison of diverse affinity based high-abundance protein depletion strategies for improved bio-marker discovery in oral fluids. *Journal of Proteomics* 75, 4165–4175.
- Kuhlmann, L., Cummins, E., Samudio, I., Kislinger, T., 2018. Cell-surface proteomics for the identification of novel therapeutic targets in cancer. *Expert Review of Proteomics* 1–17.
- Kulak, N.A., Pichler, G., Paron, I., Nagaraj, N., Mann, M., 2014. Minimal, encapsulated proteomic-sample processing applied to copy-number estimation in eukaryotic cells. *Nature Methods* 11, 319–324.
- Kulasingam, V., Diamandis, E.P., 2008a. Strategies for discovering novel cancer biomarkers through utilization of emerging technologies. *Nature Clinical Practice Oncology* 5, 588–599.
- Kulasingam, V., Diamandis, E.P., 2008b. Tissue culture-based breast cancer biomarker discovery platform. *International Journal of Cancer* 123, 2007–2012.
- Lacoste, J., 2018. Research in rare disease: From genomics to proteomics. *Assay and Drug Development Technologies* 16, 12–14.
- Latosinska, A., Frantzi, M., Vlahou, A., Merseburger, A.S., Mischak, H., 2017b. Clinical proteomics for precision medicine: The bladder cancer case. *Proteomics Clinical Applications* 12 (2).
- Latosinska, A., Mokou, M., Makridakis, M., *et al.*, 2017a. Proteomics analysis of bladder cancer invasion: Targeting EIF3D for therapeutic intervention. *Oncotarget* 8, 69435–69455.
- Lau, E., Cao, Q., Lam, M.P.Y., *et al.*, 2018. Integrated omics dissection of proteome dynamics during cardiac remodeling. *Nature Communication* 9, 120.
- Leduc, R.D., Schwammle, V., Shortreed, M.R., *et al.*, 2018. ProForma: A standard proteoform notation. *Journal of Proteome Research* 17 (3), 1321–1325.
- Lion, N., Reymond, F., Girault, H.H., Rossier, J.S., 2004. Why the move to microfluidics for protein analysis? *Current Opinion in Biotechnology* 15, 31–37.
- Lisitsa, A., Moshkovskii, S., Chernobrovkin, A., Ponomarenko, E., Archakov, A., 2014. Profiling proteoforms: Promising follow-up of proteomics for biomarker discovery. *Expert Review of Proteomics* 11, 121–129.
- Mallick, P., Kuster, B., 2010. Proteomics: A pragmatic perspective. *Nature Biotechnology* 28, 695–709.
- Mann, M., Kulak, N.A., Nagaraj, N., Cox, J., 2013. The coming age of complete, accurate, and ubiquitous proteomes. *Molecular Cell* 49, 583–590.
- Marko-Varga, G., Labaer, J., 2017. The Immune system and the proteome. *Journal of Proteome Research* 16, 1.
- Martens, L., Chambers, M., Sturm, M., *et al.*, 2011. mzML – A community standard for mass spectrometry data. *Molecular and Cellular Proteomics* 10.R110 000133.
- Martens, L., Hermjakob, H., Jones, P., *et al.*, 2005. PRIDE: The proteomics identifications database. *Proteomics* 5, 3537–3545.
- Martens, L., Vandekerckhove, J., Gevaert, K., 2005. DBToolKit: Processing protein databases for peptide-centric proteomics. *Bioinformatics* 21, 3584–3585.
- Matthiesen, R., 2013a. Algorithms for database-dependent search of MS/MS data. *Methods in Molecular Biology* 1007, 119–138.
- Matthiesen, R., 2013b. LC-MS spectra processing. *Methods in Molecular Biology* 1007, 47–63.
- Matthiesen, R., Bunkenborg, J., 2013. Introduction to mass spectrometry-based proteomics. *Methods in Molecular Biology* 1007, 1–45.
- Matthiesen, R., Carvalho, A.S., 2013. Methods and algorithms for quantitative proteomics by mass spectrometry. *Methods in Molecular Biology* 1007, 183–217.
- Mayer, G., Jones, A.R., Binz, P.A., *et al.*, 2014. Controlled vocabularies and ontologies in proteomics: Overview, principles and practice. *Biochimica et Biophysica Acta* 1844, 98–107.
- Meehan, K.L., Holland, J.W., Dawkins, H.J., 2002. Proteomic analysis of normal and malignant prostate tissue to identify novel proteins lost in cancer. *Prostate* 50, 54–63.
- Menschaert, G., Wang, X., Jones, A.R., *et al.*, 2018. The proBAM and proBed standard formats: Enabling a seamless integration of genomics and proteomics data. *Genome Biology* 19, 12.
- Meyfour, A., Pahlavan, S., Sobhanian, H., Salekdeh, G.H., 2018. 17(th) Chromosome-Centric Human Proteome Project (C-HPP) symposium in Tehran. *Proteomics* 18 (7), e1800012.
- Mueller, C., Haymond, A., Davis, J.B., Williams, A., Espina, V., 2018. Protein biomarkers for subtyping breast cancer and implications for future research. *Expert Review of Proteomics* 15, 131–152.
- Nedelkov, D., 2017. Human proteoforms as new targets for clinical mass spectrometry protein tests. *Expert Review of Proteomics* 14, 691–699.
- Nickchi, P., Jafari, M., Kalantari, S., 2015. PEIMAN 1.0: Post-translational modification Enrichment, Integration and Matching ANalysis. *Database (Oxford)* 2015, bav037.
- Okuda, S., Watanabe, Y., Moriya, Y., *et al.*, 2017. jPOSTrepo: An international standard data repository for proteomes. *Nucleic Acids Research* 45, D1107–D1111.
- Olexiouk, V., Menschaert, G., 2017. proBAMconvert: A conversion tool for proBAM/proBed. *Journal of Proteome Research* 16, 2639–2644.
- Omenn, G.S., Lane, L., Lundberg, E.K., Overall, C.M., Deutsch, E.W., 2017. Progress on the HUPO draft human proteome: 2017 metrics of the human proteome project. *Journal of Proteome Research* 16, 4281–4287.
- Patrie, S.M., 2016. Top-down mass spectrometry: Proteomics to proteoforms. *Advances in Experimental Medicine and Biology* 919, 171–200.
- Penn, A.M., Bibok, M.B., Saly, V.K., *et al.*, 2018. Verification of a proteomic biomarker panel to diagnose minor stroke and transient ischaemic attack: Phase 1 of SpectRA, a large scale translational study. *Biomarkers*. 1–48.
- Perez-Riverol, Y., Gatto, L., Wang, R., *et al.*, 2016a. Ten simple rules for taking advantage of Git and GitHub. *PLOS Computational Biology* 12, e1004947.
- Perez-Riverol, Y., Xu, Q.W., Wang, R., *et al.*, 2016b. PRIDE inspector tool suite: Moving toward a universal visualization tool for proteomics data standard formats and quality assessment of proteomeXchange datasets. *Molecular and Cellular Proteomics* 15, 305–317.
- Piening, B.D., Zhou, W., Contrepas, K., *et al.*, 2018. Integrative personal omics profiles during periods of weight gain and loss. *Cell Systems* 6 (2), 157–170.
- Rauch, A., Bellew, M., Eng, J., *et al.*, 2006. Computational Proteomics Analysis System (CPAS): An extensible, open-source analytic system for evaluating and publishing proteomic data and high throughput biological experiments. *Journal of Proteome Research* 5, 112–121.
- Righetti, P.G., Boschetti, E., 2008. The ProteoMiner and the FortyNiners: Searching for gold nuggets in the proteomic arena. *Mass Spectrometry Reviews* 27, 596–608.
- Romer, M., Eichner, J., Drager, A., *et al.*, 2016. ZBIT Bioinformatics Toolbox: A web-platform for systems biology and expression data analysis. *PLOS ONE* 11, e0149263.
- Rosen, J., Handy, K., Gillan, A., Smith, R., 2017. JS-MS: A cross-platform, modular javascript viewer for mass spectrometry signals. *BMC Bioinformatics* 18, 469.
- Ruhaak, L.R., Van Der Burg, Y.E., Cobbaert, C.M., 2016. Prospective applications of ultrahigh resolution proteomics in clinical mass spectrometry. *Expert Review of Proteomics* 13, 1063–1071.
- Schmidt, M.S., King, C.L., Thomsen, E.K., *et al.*, 2014. Liquid chromatography-mass spectrometry analysis of diethylcarbamazine in human plasma for clinical pharmacokinetic studies. *Journal of Pharmaceutical and Biomedical Analysis* 98, 307–310.
- Schmidt, T., Samaras, P., Frejno, M., *et al.*, 2018. ProteomicsDB. *Nucleic Acids Research* 46, D1271–D1281.

- Schmit, P.O., Vialaret, J., Wessels, H., *et al.*, 2018. Towards a routine application of Top-Down approaches for label-free discovery workflows. *Journal of Proteomics* 175, 12–26.
- Schwenk, J.M., Omenn, G.S., Sun, Z., *et al.*, 2017. The Human Plasma Proteome Draft of 2017: Building on the Human Plasma PeptideAtlas from mass spectrometry and complementary assays. *Journal of Proteome Research* 16, 4299–4310.
- Seeley, E.H., Oppenheimer, S.R., Mi, D., Chaurand, P., Caprioli, R.M., 2008. Enhancement of protein sensitivity for MALDI imaging mass spectrometry after chemical treatment of tissue sections. *Journal of the American Society for Mass Spectrometry* 19, 1069–1077.
- Segura, V., Garin-Muga, A., Guruceaga, E., Corrales, F.J., 2017. Progress and pitfalls in finding the 'missing proteins' from the human proteome map. *Expert Review of Proteomics* 14, 9–14.
- Seow, T.K., Korke, R., Liang, R.C., *et al.*, 2001. Proteomic investigation of metabolic shift in mammalian cell culture. *Biotechnology Progress* 17, 1137–1144.
- Sergeant, K., Printz, B., Gutsch, A., *et al.*, 2017. Didehydrophenylalanine, an abundant modification in the beta subunit of plant polygalacturonases. *PLOS ONE* 12, e0171990.
- Shiromizu, T., Kume, H., Ishida, M., *et al.*, 2017. Quantitation of putative colorectal cancer biomarker candidates in serum extracellular vesicles by targeted proteomics. *Scientific Reports* 7, 12782.
- Snyder, M., Mias, G., Stanberry, L., Kolker, E., 2014. Metadata checklist for the integrated personal OMICS study: Proteomics and metabolomics experiments. *OMICS* 18, 81–85.
- Soldes, O.S., Kuick, R.D., Thompson 2nd, I.A., *et al.*, 1999b. Differential expression of Hsp27 in normal oesophagus, Barrett's metaplasia and oesophageal adenocarcinomas. *British Journal of Cancer* 79, 595–603.
- Soldes, O.S., Younger, J.G., Hirsch, R.B., 1999a. Predictors of malignancy in childhood peripheral lymphadenopathy. *Journal of Pediatric Surgery* 34, 1447–1452.
- Spidlen, J., Brinkman, R.R., 2018. Use FlowRepository to share your clinical data upon study publication. *Cytometry Part B: Clinical Cytometry* 94, 196–198.
- Szasz, A.M., Gyorffy, B., Marko-Varga, G., 2017. Cancer heterogeneity determined by functional proteomics. *Seminars in Cell and Developmental Biology* 64, 132–142.
- 2018.UniProt: The universal protein knowledgebase. *Nucleic Acids Research*.
- Tholey, A., Taylor, N.L., Heazlewood, J.L., Bendixen, E., 2017. We are not alone: The iMOP initiative and its roles in a biology- and disease-driven human proteome project. *Journal of Proteome Research* 16, 4273–4280.
- Thomsen, M.B.H., Nordentoft, I., Lamy, P., *et al.*, 2017. Comprehensive multiregional analysis of molecular heterogeneity in bladder cancer. *Scientific Reports* 7, 11702.
- Thul, P.J., Lindskog, C., 2018. The human protein atlas: A spatial map of the human proteome. *Protein Science* 27, 233–244.
- Thygesen, C., Boll, I., Finsen, B., Modzel, M., Larsen, M.R., 2018. Characterizing disease-associated changes in post-translational modifications by mass spectrometry. *Expert Review of Proteomics* 1–14.
- Tian, Q., Price, N.D., Hood, L., 2012. Systems cancer medicine: Towards realization of predictive, preventive, personalized and participatory (P4) medicine. *Journal of Internal Medicine* 271, 111–121.
- Toby, T.K., Fornelli, L., Kelleher, N.L., 2016. Progress in Top-Down Proteomics and the Analysis of Proteoforms. *Annual Review of Analytical Chemistry (Palo Alto Calif)* 9, 499–519.
- Trenchevska, O., Nelson, R.W., Nedelkov, D., 2016. Mass Spectrometric Immunoassays in Characterization of Clinically Significant Proteoforms. *Proteomes* 4.
- Trindade, F., Ferreira, R., Magalhaes, B., *et al.*, 2018. How to use and integrate bioinformatics tools to compare proteomic data from distinct conditions? A tutorial using the pathological similarities between Aortic Valve Stenosis and Coronary Artery Disease as a case-study. *Journal of Proteomics* 171, 37–52.
- Tsiatsiani, L., Heck, A.J., 2015. Proteomics beyond trypsin. *The FEBS Journal* 282, 2612–2626.
- Tsushima, S., Satoh, M., Umemura, H., *et al.*, 2018. Assessment by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry of the effects of preanalytical variables on serum peptidome profiles following long-term sample storage. *Proteomics Clinical Applications* 12 (13), e1700047.
- Uppal, K., Ma, C., Go, Y.M., Jones, D.P., Wren, J., 2018. xMWAS: A data-driven integration and differential network analysis tool. *Bioinformatics* 34, 701–702.
- Van Karnebeek, C.D.M., Wortmann, S.B., Tarailo-Graovac, M., *et al.*, 2018. The role of the clinician in the multi-omics era: Are you ready? *Journal of Inherited Metabolic Disease*.
- Vizcaino, J.A., Csordas, A., Del-Toro, N., *et al.*, 2016a. 2016 update of the PRIDE database and its related tools. *Nucleic Acids Research* 44, 11033.
- Vizcaino, J.A., Csordas, A., Del-Toro, N., *et al.*, 2016b. 2016 update of the PRIDE database and its related tools. *Nucleic Acids Research* 44, D447–D456.
- Vizcaino, J.A., Mayer, G., Perkins, S., *et al.*, 2017. The mzIdentML Data Standard Version 1.2, Supporting Advances in Proteome Informatics. *Molecular and Cellular Proteomics* 16, 1275–1285.
- Walzer, M., Qi, D., Mayer, G., *et al.*, 2013. The mzQuantML data standard for mass spectrometry-based quantitative studies in proteomics. *Molecular and Cellular Proteomics* 12, 2332–2340.
- Wang, M., Weiss, M., Simonovic, M., *et al.*, 2012. PaxDb, a database of protein abundance averages across all three domains of life. *Molecular and Cellular Proteomics* 11, 492–500.
- Wasinger, V.C., Cordwell, S.J., Cerpa-Poljak, A., *et al.*, 1995. Progress with gene-product mapping of the Mollicutes: *Mycoplasma genitalium*. *Electrophoresis* 16, 1090–1094.
- Wilkins, M.R., Sanchez, J.C., Gooley, A.A., *et al.*, 1996b. Progress with proteome projects: Why all proteins expressed by a genome should be identified and how to do it. *Biotechnology & Genetic Engineering Reviews* 13, 19–50.
- Wilkins, M.R., Sanchez, J.C., Williams, K.L., Hochstrasser, D.F., 1996a. Current challenges and future applications for protein maps and post-translational vector maps in proteome projects. *Electrophoresis* 17, 830–838.
- Wisniewski, J.R., 2013. Proteomic sample preparation from formalin fixed and paraffin embedded tissue. *Journal of Visualized Experiments* 79.
- Wisniewski, J.R., Dus, K., Mann, M., 2013. Proteomic workflow for analysis of archival formalin-fixed and paraffin-embedded clinical samples to a depth of 10,000 proteins. *Proteomics Clinical Applications* 7, 225–233.
- Zougman, A., Banks, R.E., 2015. C-Strap sample preparation method – In-situ cysteinyl peptide capture for bottom-up proteomics analysis in the STrap format. *PLOS ONE* 10, e0138775.

Relevant Websites

<https://hupo.org/human-proteome-project>
 HUPO.
<https://www.nextprot.org>
 neXtProt.
<http://www.ebi.ac.uk/pride/archive/>
 PRIDE.
<http://github.com/PRIDE-Toolsuite/>
 PRIDE-Toolsuite.
<http://probam.biobix.be>
 proBAM.

<https://www.ProteomicsDB.org>
ProteomicsDB.
<https://pymzml.github.io>
pymzML.
www.proteinatlas.org
THE HUMAN PROTEIN ATLAS.

Molecular Mechanisms Responsible for Drug Resistance

Mun Fai Loke and Aimi Hanafi, University of Malaya, Kuala Lumpur, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

In general, drug resistance is the reduction in effectiveness of a medication or drug to cure a disease or condition. More commonly, antimicrobial resistance (also known as drug resistance) is the ability of microbes, such as bacteria, viruses, parasites, or fungi, to survive in the presence of a chemical drug that was previously effective in killing it or inhibiting its growth. The concept of drug resistance was first considered when bacteria became resistant to certain antibiotics, but since then similar mechanisms have been found to occur in other diseases; among many are cancer cells developing resistance to chemotherapy drugs and human immunodeficiency virus (HIV) becoming resistance to antiretroviral drugs.

The first true antibiotic, penicillin, was discovered by Alexander Fleming, Professor of Bacteriology at the St. Mary's Hospital in London in 1928. However, as early as 1908, Paul Ehrlich, the father of modern chemotherapy, had already observed the existence of stably inherited drug resistance in *Trypanosoma* (Ehrlich, 1909). Thus, antimicrobial resistance is not a new concept. Antimicrobial resistance was recognized even before the dawn of the antibiotic era. By the 1950s, the medical community was already aware of the problem of antimicrobial resistance. Notably, methicillin was first clinically used in 1959 and only 2 years later, scientists had already identified the first case of methicillin-resistant *Staphylococcus aureus* (Jevons, 1961). This demonstrated the short evolution cycle in the development of antimicrobial resistance in bacteria and other microorganisms.

Less than a century since the discovery of the first antibiotic, antimicrobial resistance has become one of the biggest threats to global health and human development. Many common pathogens are becoming resistant to the antibiotics resulting in longer illnesses and higher mortality. Concurrently, not enough new antimicrobial drugs are being developed to replace increasingly ineffective ones. In September 2016, a historical meeting of world leaders at the United Nations (UN) General Assembly was held to address the seriousness and scope of the situation and commit to fighting antimicrobial resistance together. Notably, this was only the fourth time in UN history that a health topic is discussed at the General Assembly (HIV, noncommunicable diseases, and Ebola were the others).

Definitions

Following the definition by Brauner *et al.* (2016), “resistance” describes the inheritable ability of microorganisms to grow at high concentrations of an antimicrobial agent by acquiring resistance mutations and is quantified by the minimum inhibitory concentration (MIC) of the particular agent. The genes that are involved in these mechanisms are collectively termed the resistome. However, microorganisms can also survive extensive antimicrobial treatments without acquiring resistance mutations. The terms “tolerance” and “persistence” distinguish these modes of survival from “resistance,” but the definitions of these different terms, and their distinction from one another, have remained somewhat ambiguous. “Tolerance” is more generally used to describe the ability of microorganisms to survive transient exposure to high concentrations of an antimicrobial agent without a change in the MIC, which is often achieved by slowing down an essential microbial process, such as growth or metabolism. Tolerance may be acquired through a genetic mutation or conferred by environmental conditions. Dormancy may be viewed as an extreme case of slow growth and low metabolism, and dormancy that leads to tolerance may also be termed “drug indifference.” For example, *Bacillus anthracis* and *Clostridium difficile* can form highly resistant endospores under unfavorable conditions. In contrast to resistance and tolerance, which are attributes of the whole bacterial populations, “persistence” is the ability of a subpopulation of a clonal bacterial population to survive exposure to high concentrations of an antimicrobial agent. Persistence is typically observed when the majority of the bacterial population is rapidly killed while a subpopulation persists for a much longer period of time, despite the population being clonal.

The joint initiative by the European Centre for Disease Prevention and Control and the Centers for Disease Control and Prevention defined multiple drug resistance (MDR) as antimicrobial resistance shown by a microorganism to at least one antimicrobial agent in three or more drug categories. Recognizing different degrees of MDR, the terms “extensively drug resistant (XDR)” and “pandrug-resistant (PDR)” have also been introduced (Magiorakos *et al.*, 2012). XDR is defined as nonsusceptibility to at least one antimicrobial agent in all but two or fewer drug categories (i.e., bacterial isolates remain susceptible to only one or two categories) and PDR refers to nonsusceptibility to all antimicrobial agents in all drug categories.

Molecular Mechanisms of Antimicrobial Resistance

The genetic characterization of antibiotic-resistant bacterial strains has uncovered many molecular mechanisms of resistance. The five fundamental mechanisms of antibiotic-resistance are (1) enzymatic activity that directly inactivate or degrade the antibiotic, (2) alteration of bacterial proteins that are targets of antibiotics, (3) reduction in membrane permeability to antibiotics, (4)

activation of efflux pumps that pump out antibiotics, and (5) activation of resistant bacterial metabolic pathways. Key candidate genes involved in the mechanisms of antibiotic-resistance are described in [Table 1](#).

Bacteria can destroy or modify antibiotics, thus resisting their action. Inactivation of antibiotics by hydrolysis is a major mechanism of antibiotic-resistance. Antibiotics can also be inactivated by transfer of chemical group. The addition of chemical groups to vulnerable sites on the antibiotic molecule by bacterial enzymes prevents the antibiotic from binding to its target protein as a result of steric hindrance. Various different chemical groups can be transferred, including acyl, phosphate, nucleotidyl, and ribitoyl groups, and the enzymes that are responsible form a large and diverse family of antibiotic-resistance enzymes.

Modification of the bacterial protein target is also an effective means of antibiotic resistance. Most antibiotics bind to the specific protein targets with high affinity, thus preventing the normal activity of the proteins. Changes to the protein structure that prevent efficient antibiotic binding, but still enable the protein to perform its normal function, can confer resistance to antibiotics.

Most antibiotics are targeted at intracellular processes, and must be able to penetrate the bacterial cell envelope to be effective. Compared to Gram-positive bacteria, Gram-negative bacteria are intrinsically less permeable to many antibiotics because of their outer membrane, which forms a permeability barrier. There are essentially two pathways by which antibiotics can penetrate the outer membrane: A lipid-mediated pathway for hydrophobic antibiotics, and general diffusion porins for hydrophilic antibiotics. The lipid and protein compositions of the outer membrane have a strong impact on the sensitivity of bacteria to many types of antibiotics, and drug resistance involving modifications of these macromolecules is common. The two major porin-based mechanisms for antibiotic-resistance are (1) reduction of porins or replacement of one or two major porins by another, and (2) altered function due to specific mutations reducing permeability.

Bacterial efflux pumps have physiological functions and their expression is tightly regulated in response to environmental and physiological triggers. Genes encoding for efflux pump proteins can be found on chromosomes or plasmids. Bacterial efflux pumps can be classified into five families based on their composition, number of transmembrane spanning regions, energy sources, and substrates. The five families are (1) resistance-nodulation-division (RND) family, (2) major facilitator superfamily (MFS), (3) ATP (adenosine triphosphate)-binding cassette (ABC) superfamily, (4) drug/metabolite transporter superfamily, and (5) multidrug and toxic compound extrusion family. The RND superfamily is only found in Gram-negative bacteria. On the other hand, efflux systems of the other four families are widely found in both Gram-positive and -negative bacteria. Bacterial efflux pumps actively transport many antibiotics out of the bacterial cell and are major contributors to the intrinsic resistance of bacteria. RND and MFS family pumps are associated extensively with clinically significant antibiotic resistance. When overexpressed, efflux pumps can also confer high levels of drug resistance to many clinically useful antibiotics. Some efflux pumps have narrow substrate specificity but many transport a wide range of structurally dissimilar substrates and are known as multidrug resistance efflux pumps. Our knowledge of MDR determinants are continuously propelled by the growing availability of the X-ray crystal structures of efflux pumps. These guide the development of novel efflux pump inhibitors that can potentially help in our war

Table 1 Key candidate genes involved in the mechanisms of antibiotic resistance

Mode of action	Class of antibiotics	Candidate genes	Mechanisms of resistance
Inhibition of protein synthesis	Macrolides	Erythromycin ribosome methylation genes (<i>erm</i>)	Posttranscriptional modifications of the <i>23SrRNA</i>
		Macrolide efflux genes (<i>mef</i>) <i>macA-macB</i> genes	Active drug efflux
		<i>23SrRNA</i> gene	Mutations of the 50S ribosomal subunit target
	Chloramphenicol	Chloramphenicol acetyltransferase gene (<i>cat</i>)	Elaboration of chloramphenicol acetyltransferase that inactivates chloramphenicol
		<i>23SrRNA</i> gene	Mutations of the 50S ribosomal subunit target
Interference with cell wall production	Aminoglycosides	Acriflavine resistance genes (<i>acr</i>)	Probable aminoglycoside efflux pump
		Aminoglycoside N(6')-acetyltransferase genes (<i>aacA</i>)	Aminoglycoside-modifying enzyme
	Tetracyclines	<i>16SrRNA</i> gene	Mutations of the 30S ribosomal subunit target
		Tetracycline resistance gene (<i>Tet</i>)	Active drug efflux
		β -lactamase structural genes (<i>bla</i>)	Inactivate the antibiotics by hydrolyzing the β -lactam ring
Interrupting nucleic acid synthesis	Fluoroquinolones	<i>mecA</i> gene	Alternative low-affinity penicillin-binding protein PBP2a
		DNA gyrase (<i>gyr</i>), DNA topoisomerase IV (<i>par</i>)	Mutations in target enzyme
	Rifampicin	DNA-dependent RNA polymerase B (<i>rpoB</i>)	Mutations in target enzyme
Blocking of DNA replication	Trimethoprim	<i>dfp</i> gene	Mutations of the target dihydrofolate reductase (DHFR)
		<i>folA</i> gene	Mutations of the structural gene for DHFR
	Sulfonamide (structural analogs of p-aminobenzoic acid)	Dihydropteroate synthase gene (<i>dhps</i>)	Mutations of the target dihydropteroate synthase
		Sulfonamide resistance genes (<i>sul</i>) <i>folP</i> gene	Resistant sulfonamide-resistant dihydropteroate synthase Mutations of the structural gene for DHPS

against antimicrobial resistance. In addition, preventing the overexpression of efflux genes by targeting to their transcription regulators is also an alternative emerging strategy to combat antimicrobial resistance.

Bacteria acquire resistant gene mutation and/or horizontal gene transfer via transformation, transduction, or conjugation. Antibiotic-resistance can be either plasmid-mediated or maintained on the bacterial chromosome. Many times, microorganisms employ several mechanisms in order to attain drug resistance.

Drug resistance among pathogenic fungal species, including *Candida*, *Aspergillus*, and *Cryptococcus*, generally involve similar mechanisms as antibiotic-resistance in bacteria. Reduced drug accumulation in fungal cells involve overexpression of efflux pumps as well as pH- and ATP-independent facilitated diffusion mechanisms. Efflux pumps in fungi mainly belong to the ABC superfamily and the MFS family. In addition, mutations in drug target genes may result in structural change of the fungal protein targets, thereby reducing the affinity of the inhibitors to their specific target. For example, mutations in *ERG11* (in *Candida albicans* and *Cryptococcus neoformans*) or *cyp51* (in *Aspergillus fumigatus*) that code for the ergosterol biosynthetic enzyme, lanosterol demethylase, change the structure of the enzyme, and decrease its affinity to specific antifungal drugs (Gonçalves *et al.*, 2016). Inhibition of ergosterol biosynthesis prevents the biosynthesis of ergosterol, which is a unique fungal cell membrane component, therefore the evoking the resistance mechanism. Another mechanism employed in resistant fungal strains is the overexpression of the drug target. Antifungal resistance can also be activated through stress response pathway regulation. For example, Hsp90 is a molecular chaperone that stabilizes the regulators in cellular stress responses. In response to drug inhibition, Hsp90 and other components of the pathway, including calcineurin and the Pkc1 signaling, are overexpressed (Robbins *et al.*, 2017).

Besides antimicrobial resistance, drug resistance also emerges in treatments of noninfectious human diseases. Drug resistance in cancer cells can also be observed. Besides playing a role in antimicrobial resistance, efflux pumps in cancer cells also act to reduce the accumulation of chemotherapeutic drugs in cancer cells. This often involves efflux pump proteins belonging to the ABC superfamily, which includes P-glycoprotein (P-gp), multidrug resistance protein 1, breast cancer resistance protein, and lung resistance-related protein. Anticancer drugs can also be inactivated by the enzymatic activity of glutathione S-transferase, which catalyzes the conjugation of glutathione to a xenobiotic compound by forming a thioether bond. Alterations in drug targets can also occur in drug-resistant cancer cells. Unlike normal cells, resistant tumor cells in breast cancer that lack estrogen receptors no longer depend on estrogen for growth. This leads to resistance to endocrine therapy. Decreased level of membrane permeability to chemotherapeutic drugs is demonstrated in the presence of lower levels of cholesterol and differences in lipid composition in cancer cells compared to normal cells.

In addition to the general mechanisms mentioned, cancer cells have also developed specific drug resistance mechanisms. Changes in the cell cycle may affect the balance between cell cycle arrest and apoptosis and disturbance to cell cycle checkpoints could lead to drug resistance. The activation of DNA damage checkpoint components allows the survival of cells after a drug treatment. Drug-resistant cancer cells may also evade apoptosis by enabling cell proliferation. Tumor suppressors in normal cells, such as p53, and phosphatase and tensin homolog deleted on chromosome 10, are defective in cancer cells, thereby disabling DNA damage repairs and cell cycle arrests. In addition to the defect in tumor suppressors, the overexpression of oncogenes, such as *Bcl-2* genes, may also inhibit apoptosis (Kartal-Yandim *et al.*, 2016).

The molecular mechanisms in antiviral resistance differ at different stages of viral infections. Mutations in genes coding for replication of viral DNA can decrease the affinity and binding of DNA polymerase to inhibitors, decrease the viral replication capacity, and affect hydrophobic interactions between nucleotides. Furthermore, antiviral drug resistance may also be caused by mutations in viral thymidine kinase encoding genes and lead to the decrease in phosphorylation of nucleoside analogs. It is because the phosphotransferase is used in antiviral therapies to produce competitive inhibitors for viral DNA polymerase. Mutations can also occur around active sites to prevent integrase and protease inhibitors from binding. Virus can also use different host cell coreceptors or the same coreceptors with changes in tropism, which may be the result of multiple mutations, to bypass the effects of inhibitors (Shafer *et al.*, 2011).

Bioinformatics in Combating Antimicrobial Resistance

Mutations introduce diversity into genomes, leading to selective changes and driving evolution. In particular, nonsynonymous single-nucleotide polymorphisms within the protein coding regions of the genome have been strongly associated with occurrence of drug resistance. Next-generation sequencing technologies have made the sequencing of whole-microbial genomes and metagenomes more affordable and have increased the feasibility of applying rapid whole-microbial genome sequencing in clinical microbiology laboratories. Furthermore, the number of microbial genomes and metagenomes deposited in the NCBI GenBank database has increased exponentially as more and more genomes are sequenced in a short time span. However, the barrier to the routine implementation of whole-genome sequencing is the lack of automated, user-friendly bioinformatics tools that interpret the sequence data and provide clinically meaningful information by scientists and clinicians. Several general databases are available to aid in rapidly identifying existing, putative, new, or emerging antibiotics resistance genes and mutations in chromosomal target genes that are associated with resistance. These include Antibiotic Resistance Genes Online (see "Relevant Website section"), the microbial database of protein toxins, virulence factors, and antibiotic-resistance genes for biodefense applications (see "Relevant Website section"), the Antibiotic Resistance Genes Database (see "Relevant Website section"), Resfinder database (see "Relevant Website section"), the Comprehensive Antibiotic Resistance Database (see "Relevant Website section"), the Antibiotic Resistance Gene-ANNOTation database (see "Relevant Website section"), the Resfams (see "Relevant Website section") and the MEGARes database (see "Relevant Website section"). In response to increasing concern over antimicrobial resistance, the team at Pathosystems Resource

Integration Center (PATRIC) has also developed new tools to facilitate researchers' understanding of the genetic basis of drug resistance (Antonopoulos *et al.*, 2017). The team used antimicrobial resistance phenotype data of more than 15,000 genomes in the PATRIC database to build machine learning-based classifiers that can predict the antimicrobial resistance phenotypes and the genomic regions associated with drug resistance that can be used for comparative analysis. Drug resistance databases for specific organisms, such as HIV and *Mycobacterium tuberculosis*, have also been developed. However, most research on bacterial antibiotic-resistance did not take into consideration gene–gene interactions and multiple genetic mutations in the emergence of transmissible drug resistance. Recently, a group of researchers from Huazhong Agricultural University (Wuhan, China) identified gene pairs associated with *Mycobacterium tuberculosis* drug resistance using GBOOST, a software package for identifying gene–gene interactions in genome-wide association studies (Cui *et al.*, 2016). GBOOST was used to detect SNP–SNP interactions on risk of drug resistance and a chi-square test method to examine the interaction effect between two SNPs and phenotypes. This method is also useful for providing a deeper insight into the mechanisms underlying drug resistance in other bacteria.

Cancer therapies are limited by the development of drug resistance, and mutations in drug targets are one of the main driving factors for the developing resistance to chemotherapeutic drugs. CancerDR is a cancer drug resistance database that will be useful for identification of genetic alterations in genes encoding drug targets, and in turn the residues responsible for drug resistance.

Systematic experimental evaluation of all genomic mutations and characterizing mutation effects of all mutations in a system of interest is impractically time-consuming and not cost-effective, even more so considering the range of different mechanisms in which mutations can affect protein function and interactions. Thus, this has created interest in the development of computational tools to understand the molecular consequences of mutations to aid and guide rational experimentation. These tools can also be employed to prioritize mutations for further experimental investigation, identification, and anticipation of resistant variants and resistance hotspots. Such knowledge can then be applied to the design of drugs less prone to resistance, as well as to drive the development of public health policies and aid in establishing more appropriate and personalized treatments. Computational methods, particularly homology modeling of the 3-dimensional structure of proteins, have been developed to understand the effects of coding mutations to protein function and interactions.

Currently, high-throughput screening (HTS) is widely used to screen large natural and artificial compound libraries against specific protein targets in drug discovery. However, this method can be costly and time-consuming even with the use of robotics. Computer-assisted or virtual screening (VS) can help to overcome some of the disadvantages of a full HTS when used to complement the HTS process in drug discovery. In the latter, VS is used to shortlist potential active compounds from large libraries, thus reducing the size of the library prior to the more costly and time-consuming HTS experiments. Unlike HTS experiments, VS simulations do not require the physical synthesis of compounds. However, VS still requires experimental information, such as protein structure for structure-based approach or binding properties of known active compounds for ligand-based approach. Molecular docking methodology is widely used in VS simulations in large part due to the efficiency of these calculations. In contrast to laboratory testing, molecular docking typically requires only a few minutes of computing time on a single core per ligand; thus, 10,000 to 100,000 compounds can be virtually screened on a small- to medium-sized cluster in a day. While molecular docking forms the basis for structure-based VS, the ligand-based approach is built upon the similarity principle in which similar compounds are assumed to cause similar biological effects. With this approach large ligand libraries can then be screened for compounds with similar chemical properties to the known actives, resulting in the identification of novel potentially active compounds. However, computer-assisted drug screening should still be a verified experimentally and VS is perhaps best used as an aid to HTS for prefiltering larger libraries to select subsets of compounds (focused library) to increase productivity in drug discovery.

Conclusion

The application of bioinformatics tools make it possible to practice precision medicine – the right medicine at the right dose for the right patient at the right time – in combating the drug resistance crisis by changing the way therapeutic drugs are developed and used to treat both infectious and noninfectious diseases, such as cancers.

See also: Comparative Epigenomics. Molecular Dynamics Simulations in Drug Discovery. Natural Language Processing Approaches in Bioinformatics. Structure-Based Drug Design Workflow

References

- Antonopoulos, D.A., Assaf, R., Aziz, R.K., *et al.*, 2017. PATRIC as a unique resource for studying antimicrobial resistance. *Briefings in Bioinformatics* 2017, 1–9.
- Brauner, A., Fridman, O., Gefen, O., Balaban, N.Q., 2016. Distinguishing between resistance, tolerance and persistence to antibiotic treatment. *Nature Reviews Microbiology* 14 (5), 320–330.
- Cui, Z.J., Yang, Q.Y., Zhang, H.Y., Zhu, Q., Zhang, Q.Y., 2016. Bioinformatics identification of drug resistance-associated gene pairs in *Mycobacterium tuberculosis*. *International Journal of Molecular Sciences* 27 (9), E1417.
- Ehrlich, P., 1909. *Ueber Moderne Chemotherapie*. Leipzig: Akademische Verlagsgesellschaft m.b.H., pp. 167–202.
- Gonçalves, S.S., Souza, A.C.R., Chowdhary, A., Meis, J.F., Colombo, A.L., 2016. Epidemiology and molecular mechanisms of antifungal resistance in *Candida* and *Aspergillus*. *Mycoses* 59 (4), 198–219.
- Jevons, M.P., 1961. "Celbenin" – Resistant Staphylococci. *British Medical Journal* 1, 124.
- Kartal-Yandim, M., Adan-Gokbulut, A., Baran, Y., 2016. Molecular mechanisms of drug resistance and its reversal in cancer. *Critical Reviews in Biotechnology* 36 (4), 716–726.

- Magiorakos, A.P., Srinivasan, A., Carey, R.B., *et al.*, 2012. Multidrug-resistant, extensively drug-resistant and pandrug-resistant bacteria: An international expert proposal for interim standard definitions for acquired resistance. *Clinical Microbiology and Infection* 18 (3), 268–281.
- Robbins, N., Caplan, T., Cowen, L.E., 2017. Molecular evolution of antifungal drug resistance. *Annual Review of Microbiology* 71 (1).
- Shafer, R.W., Najera, I., Chou, S., 2011. Mechanisms of resistance to antiviral agents. In: Versalovic, J., Carroll, K.C., Funke, G., *et al.* (Eds.), *Manual of Clinical Microbiology*, tenth ed. American Society of Microbiology, pp. 1710–1728.

Further Reading

- Alliance for the Prudent Use of Antibiotics (APUA) Fact Sheets on Antibiotic Resistance. Available at: http://emerald.tufts.edu/med/apua/consumers/fact_sheets.shtml.
- Centers for Disease Control and Prevention. Available at: <https://www.cdc.gov/drugresistance/index.html>.
- "The Race Against Drug Resistance": A Report of the Center for Global Development's Drug Resistance Working Group (CGD). Available at: <https://www.cgdev.org/publication/race-against-drug-resistance>.
- World Health Organization Fact Sheets on Antibiotic resistance and Antimicrobial resistance. Available at: <http://www.who.int/mediacentre/factsheets/en/>.

Relevant Websites

- <http://www.mediterranee-infection.com/article.php?laref=282&titer=arg-annot>
Antibiotic Resistance Gene-ANNOTation (ARG-ANNOT) database.
- <https://ardb.cbcb.umd.edu/>
Antibiotic Resistance Genes Database (ARDB).
- <http://www.argodb.org/>
Antibiotic Resistance Genes Online (ARGO).
- <http://crdd.osdd.net/raghava/cancerdr/>
CancerDR.
- <https://card.mcmaster.ca/>
Comprehensive Antibiotic Resistance Database (CARD).
- <https://megares.meglab.org>
MEGARes.
- <http://mvirdb.llnl.gov/>
Microbial database of protein toxins, virulence factors, and antibiotic resistance genes for bio-defence applications (MvirDB).
- <http://umr5558-bibiserv.univ-lyon1.fr/mubii/mubii-select.cgi>
MUBII-TB-DB database.
- <https://patricbrc.org/view/DataType/AntibioticResistance>
Pathosystems Resource Integration Center Antimicrobial Resistance (AMR).
- <http://www.dantaslab.org/resfams/>
Resfams.
- <https://cge.cbs.dtu.dk/services/ResFinder/>
Resfinder database.
- <https://hivdb.stanford.edu/>
Stanford H.I.V. Drug Resistance Database (HIVDB).
- <https://tbdreamdb.ki.se/Info/Default.aspx>
Tuberculosis Drug Resistance Mutation Database (TBDReamDB).
- <https://extranet.who.int/hivdrug-surveillance/>
W.H.O. HIV drug resistance database (Database access is restricted to designated ministry of health/ART programme users only).

Biographical Sketch



Mun Fai Loke obtained his PhD from the Department of Microbiology, Yong Loo Lin School of Medicine, National University of Singapore and his MBA from the University of Southern Queensland. He was a Senior Lecturer at the Department of Medical Microbiology, Faculty of Medicine, University of Malaya and a researcher with KK Women's and Children's Hospital. His main research interests are *Helicobacter pylori*, human gastrointestinal diseases, and rare genetic disorders.



Aimi Hanafi graduated with a master's degree from the Faculty of Medicine and Health Science, Universiti Putra Malaysia. Currently, she is a PhD student at the Department of Medical Microbiology, Faculty of Medicine, University of Malaya. Her PhD work focuses on the studying of antibiotic resistance mechanism and compensation in the human gastric pathogen, *Helicobacter pylori*.

Network-Based Analysis of Host-Pathogen Interactions

Lokesh P Tripathi, Yi-An Chen, and Kenji Mizuguchi, National Institutes of Biomedical Innovation, Health and Nutrition, Osaka, Japan
Eiji Morita, Hirosaki University, Hirosaki, Japan

© 2019 Elsevier Inc. All rights reserved.

Introduction

Infectious diseases are defined as disorders caused by pathogenic microorganisms, such as bacteria, viruses, parasites or fungi that can be spread directly or indirectly (vector-borne) from one individual to another. Despite the rapid advances in treating infectious diseases over the last century, infectious diseases - such as pneumonia, flu, tuberculosis, HIV/AIDS, malaria etc. - continue to result in millions of deaths worldwide. As per the World Health Organization (WHO) and Center for Infectious Disease Research (CIDR) estimates, the mortality rates of many infectious diseases may have actually worsened over the past few decades; the infectious diseases therefore, remain a pressing concern for the public healthcare systems. The societal burden of such diseases have been further exacerbated by the emergence of new and more virulent strains of drug resistant pathogens; these events have, therefore, led to more and more research efforts to understand disease pathogenesis and to identify new candidate targets for more effective therapeutic intervention against infectious diseases. However, the strategies by which pathogens evade the host immune system and manipulate key host cellular processes for their survival remain to be fully elucidated and are, therefore, an impediment to the development of more effective therapeutics. Understanding the behaviour of the pathogen as it colonises its host and the emergent host behaviour when it comes into contact with the pathogen are, therefore, of prime interest for the researchers investigating the pathogenesis of infectious diseases.

Host-pathogen protein-protein interactions (PHIs) perform vital roles in the pathogenesis of infectious diseases. In the PHIs, the proteins encoded by the pathogenic genome invade the host cellular networks to subvert the host immune responses and to co-opt the host cellular machinery to colonise the host and spread the infection (Bhavsar *et al.*, 2007; Fig. 1). In response, host cells have evolved mechanisms to sense and clear out the pathogenic infection. This host-pathogen interplay can be highly competitive and often antagonistic in nature as the pathogens competitively disrupt the native PPIs and establish new ones with the host cellular factors. Therefore, the underlying organisational and functional structures of the PHI networks can be markedly different from those of the native host interactome (Franzosa and Xia, 2011). To understand the biology of pathogen infection, it is necessary to obtain a detailed understanding of how pathogenic proteins function inside the host cell, their interactions with the host cellular machinery and how these interactions impact cellular signalling pathways. Therefore, there is a pressing need for an integrative analysis of PHIs (Fels *et al.*, 2017; Ma-Lauer *et al.*, 2012).

The availability of complete genome sequences of many pathogens and their hosts, together with the rapid proliferation of omics technologies and the development of new experimental technologies that were more amenable to PPI mapping on a larger

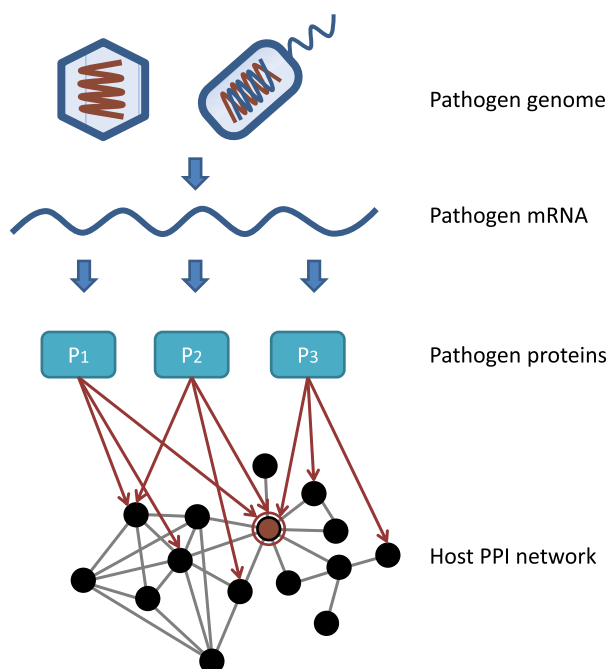


Fig. 1 A schematic of host-pathogen interactions (PHIs).

scale, have made it easier to generate large scale PHI data for many different host-pathogen systems at different organisational levels. Systems-based approaches seek to integrate experimental data with bioinformatics tools for data analysis and modelling to generate new hypotheses and prioritise further experimental investigation; such approaches have emerged as ideal approaches to investigate the biology of PHIs, generate new hypotheses and prioritise targets for clinical therapies (Durmus *et al.*, 2015).

Significant amount of progress has been made in genome-wide mapping of PHIs for different pathogens (de Chassey *et al.*, 2008; Tripathi *et al.*, 2010; Uetz *et al.*, 2006; Brito and Pinney, 2017; Calderwood *et al.*, 2007; Friedel and Haas, 2011; de Chassey *et al.*, 2013a) and also there have been attempts to integrate interaction networks and 3D structural data to extract the general principles of PHIs (Franzosa and Xia, 2011). These studies have provided valuable insights into mechanisms of pathogenicity of different pathogens. Among the various pathogens, the maximum research effort has gone into systematic mapping of virus-human host interactions on a genome wide scale (Lum and Cristea, 2016); the low complexity of the viral genomes makes it relatively easier to map PHIs involving viral pathogens compared with PHIs involving pathogenic bacteria, fungi or eukaryotes.

In this review, we will discuss PHIs largely from the perspective of the viral pathogens of human diseases. We first discuss how experimentally defined PHI data have been utilised for understanding the infection and survival mechanisms of pathogens in their hosts. Next, we briefly outline the computational methods for large-scale PHI prediction and how they can help in highlighting useful PHIs and in knowledge discovery. Finally, we discuss the applications and likely future directions of PHI data analysis.

Experimental Methods to Determine PHIs on a Large Scale

Different experimental methods have been applied to identify cellular proteins interacting with pathogen proteins. These methods differ widely in their approach; they have their advantages and shortcomings and often provide coverage of the different aspects of PHIs. Yeast two-hybrid (Y2H) system and co-affinity purification are among the most frequently used and traditional technologies to experimentally characterise PHIs; these methodologies can be easily scaled to survey PHIs on a proteome-wide scale and it is, therefore, not surprising that they together have contributed over 90% of the publicly available PHI data (Gurimand *et al.*, 2015). In addition, protein arrays and protein-complementation assays are also being increasingly employed.

Y2H is an *in vivo* system based on the fact that a certain transcription factor (TF) can be divided into two separate domains, an activating domain (AD) and the DNA binding domain (DBD), and these two domains can still function together to drive transcription when they co-locate in close proximity. In Y2H, both the DBD and the AD are fused to two separate proteins, called the *bait* and *prey*. If the bait and prey proteins interact with each other, the DBD recognises a specific DNA sequence in a promoter region and the AD is recruited within the proximity; the reconstituted TF stimulates transcription by enabling the RNA polymerase II to transcribe the downstream reporter gene (Fields and Song, 1989). Hence, the bait and prey proteins form a bridge between the DBD and AD. The yeast strain is usually deficient in the components of the pathways for histidine biosynthesis. Therefore, by using HIS3 gene, which encodes for the protein Imidazoleglycerol-phosphate dehydratase that catalyses the sixth step in histidine biosynthesis, as a reporter for the transcriptional activation, *bait* and *prey* protein interactions are easily detected by monitoring yeast growth on culture dishes with a solid medium lacking histidine. Researchers can also construct a cDNA library instead of the prey-fused protein coding regions, and perform the screening of the binding partners from the cDNA library. Sometimes, whole cDNA library screening process is automated, thereby, permitting high-throughput analysis (Buckholz *et al.*, 1999).

Y2H system has many advantages. It does not require any specialised equipment, and it is cost-effective. PPIs can be easily detected by monitoring the transformed yeast growth. Moreover, Y2H screening is highly sensitive. Therefore, this system is suitable for detecting weak and transient interactions, which are often the most interesting in PHI-induced signalling cascades. Third, the interaction reflects *in vivo* status. Yeast is a eukaryote; therefore, it supports the correct folding of eukaryotic proteins, including post-translational modifications (PTMs). These advantages therefore, easily facilitate the characterisation of physiologically relevant PPIs; thus, Y2H has become the method of choice to characterise PHIs in many host-pathogen systems (Durmus *et al.*, 2015; de Chassey *et al.*, 2008; Uetz *et al.*, 2006; Tripathi *et al.*, 2013; Nicod *et al.*, 2017; Dolan *et al.*, 2013; Shapira *et al.*, 2009).

However, Y2H suffers from notable deficiencies. Since the assay is highly sensitive, the self-activation of transcription and reporter gene expression by *bait* protein in the absence of *prey* protein may commonly occur; thus, leading to the identification of false positive PPIs. Y2H may also fail to detect many physiologically relevant PPIs since the yeast cell environment is different from that of a mammalian cell. Therefore, some important post-translational aspects such as protein sub-cellular localisation and/or post-translational modifications (PTMs), which are often required for certain PPIs, may not be reflected within yeast. Y2H is also at a slight disadvantage when characterising PPIs involving extracellular proteins or membrane proteins, since the two-hybrid system requires the fusion proteins to be translocated to the yeast nucleus. Moreover, Y2H is optimised for binary protein interactions and some PPIs require correct folding as subunits of a large protein complex. To overcome such limitations, different variations of Y2H, such as the mammalian cell based two-hybrid assay (Luo *et al.*, 1997), the membrane anchored two-hybrid assay (Snider *et al.*, 2010), and the three-hybrid assay (Maruta *et al.*, 2016) have been developed. For example, the Y2H membrane protein system, which identifies PPIs involving integral membrane proteins and membrane-associated proteins in an *in vivo* setting at the cellular membrane, was used to identify novel host protein interactions for hepatitis C virus (HCV) Core and NS4B proteins (Tripathi *et al.*, 2010), nearly all of which were not characterised by previously by the high throughput screens for HCV-host interactions using the conventional Y2H approach (de Chassey *et al.*, 2008).

Affinity purification–mass spectrometry (AP-MS) is another method to identify PPIs on a large scale; the method involves biochemical purification of protein complexes from cells, followed by the mass spectrometric identification of the components of the purified protein complexes (Dunham *et al.*, 2012). Immuno-precipitation using specific antibodies is a classical method to purify target proteins from the cells under native conditions. Affinity purification system is also available to purify ectopically expressed proteins. After the purification of an endogenous protein or an ectopically expressed affinity-tagged target protein, the co-purified proteins that are associated with the target proteins are identified by mass-spectrometry analysis, resulting in the mapping of the PPIs of the target protein. This method is widely used to characterise protein complexes of target proteins due to the simplicity and sensitivity of the mass-spectrometry analysis. For instance, flaviviruses, a group of positive-strand RNA viruses generally induce intracellular membrane rearrangement and form an organelle like structure, called the “replication organelle” that facilitates efficient genomic replication. Replication organelle appears in close proximity to the endoplasmic reticulum and also serves as a shell that protects the viruses against various cellular stress responses and therefore, it allows for a persistent viral replication in the cytoplasm. In a specific study, the replication organelle was biochemically purified from JEV (Japanese encephalitis virus, a member of the flaviviridae) infected cells and subject to extensive mass-spectrometry analyses that identified many host factors, including ESCRT subunits, and ER membrane shaping proteins, innate immunity factors as the constituents of the replication organelle. Subsequent analysis by Y2H revealed binary interactions between some of these factors and viral non-structural proteins and the follow up siRNA depletion experiments further confirmed their essential roles in viral propagation (Tabata *et al.*, 2016). In another study, Jäger *et al.* (2011) employed AP-MS coupled with a quantitative scoring system to identify 497 high-confidence HIV-human PPIs.

To overcome non-specific detection of co-purified proteins, several two-step tandem affinity protein purification systems have been developed (Burckstummer *et al.*, 2006). This approach allows the preparation of a substantially pure target protein complex and reduces the background signals. The quantitative mass-spectrometry analysis also has been used to identify the contaminants (Trinkle-Mulcahy *et al.*, 2008). However, the method does not always capture the direct interactions between a target protein and its binding partners and reflects only steady-state interactions, and therefore, possibly missing weak and transient interactions.

Eventually, a combination of different methods will provide a more comprehensive map of protein interaction dynamics, since each method will likely lead to the exploration of a different aspect of PHI interactome.

PPI Network-Based Analysis of PHIs

Construction and analysis of PHI networks (PHINs) spanning extensive protein-protein interaction (PPI) maps between host and pathogen interactomes is a promising avenue to obtain a better understanding of the biological context of PHIs and to understand the pathogenesis of infectious diseases.

Network topology that is the connectivity of the proteins in a PPI network (PPIN) can often provide relevant insights into the biological functions of the proteins and their relative importance in a PPIN (Raman, 2010). Analyses of PHIN across different pathogen-host systems have revealed systematic trends in host-pathogen interaction networks. In specific studies that examined host-pathogen PPIs in the context of pathogenesis of infectious diseases, it was noted that viral and bacterial pathogens preferentially interacted with network ‘hubs’ and ‘bottlenecks’ in the human protein interactome (de Chassey *et al.*, 2008; Tripathi *et al.*, 2013; Dyer *et al.*, 2008). Network topological attributes can indeed guide the prioritisation and selection of candidates for experimental characterisation of pathogen-induced disorders (Tripathi *et al.*, 2012). For instance, Diamond *et al.* (2012) combined topological properties derived from a protein association network based on clinical data to evaluate the onset and acuteness of HCV-induced liver fibrosis and they identified a novel role of protein kinase A RII- α , which was judged to be a ‘bottleneck’ in the protein association network, to play a key role in HCV-induced liver injury (Diamond *et al.*, 2012).

Halehalli and Nagarajaram (2015) further established that in addition to targeting ‘hubs’ and ‘bottlenecks’, pathogenic proteins disproportionately form complexes with intrinsically disordered proteins and they proposed disorder associated conformational flexibility as one of the typical characteristics of VHIs (Halehalli and Nagarajaram, 2015).

PHIs Across Pathogenic Species

PHINs are also very useful in understanding the differences in strategies adopted by phylogenetically different pathogenic species to circumvent the host defence systems and modulate the host cellular machinery to survive in the intracellular environment and for their propagation during the infectious process. For instance, a comparison of PHI networks involving DNA and RNA viruses highlighted that while the DNA viruses simultaneously perturbed both the cellular and metabolic processes, RNA viruses preferentially targeted host factors associated with intracellular transport, translational machinery and sub-cellular localisation (Durmus and Ulgen, 2017). Among other examples, pathogenic bacteria also interact with host interactomes via different mechanisms and some of these mechanisms are remarkably different from those employed by pathogenic viruses (Nicod *et al.*, 2017). A key difference is that bacteria do not explicitly depend on the host cellular machinery for their own replication, since unlike viruses, bacteria have a living and a fully functional cellular machinery of their own. Therefore, a deeper understanding of the specifics of PHI networks involved in the infection by different classes of pathogenic organisms will facilitate a development of more specific and probably more effective therapeutics against infectious diseases.

Despite the rapid strides in the construction and analysis of PHINs, a more detailed and broad-based analysis is hampered by the limited amount of large-scale PHI data and the host PPI interactome data, especially in the case of animal model systems

(Murakami *et al.*, 2017) that are often used to investigate the pathogenesis of infectious human diseases. Therefore, it is necessary to deploy computational tools and approaches for literature mining and PHI prediction to maximise the availability of PHI and host PPI data and ensure the robustness of biological analyses leveraging PHINs (Table 1).

An example of such an approach would to leverage the principle of orthology in model organism research; this is based on the belief that the genetic makeup of the different cellular processes (such as PPIs) is inherently conserved across different organisms. Tripathi *et al.* (2012), for instance, analysed the biological significance of differential protein levels in transgenic mouse models of HCV infection. The differentially expressed mouse proteins were assimilated into an orthologous human protein interactome with the help of TargetMine data analysis platform (Chen *et al.*, 2011, 2016), to highlight the cellular pathways significantly impacted by HCV infection and to prioritise novel anti-HCV host factors. Follow-up cellular assays validated VTI1A, a vesicular transport associated factor, as a novel regulator of HCV propagation.

Extracting PHIs From Literature and Public Databases

PHIs are scattered across several published studies in the biomedical literature and many publicly available databases that catalogue different types of PHIs. For instance, virus–host interactions (VHIs) for a large number of viruses can be easily sourced from organism-specific resources such as Human Immunodeficiency Virus (HIV)-1 Human-Interaction Database (Fu *et al.*, 2009) or databases that are entirely dedicated to VHIs such as VirHostNet (Guirimand *et al.*, 2015), VirusMentha (Calderone *et al.*, 2015) and ViRBase (Li *et al.*, 2015). Likewise, PATRIC (Wattam *et al.*, 2014) is a resource that exclusively catalogues bacterial–host interactions. Additionally, there are comprehensive publicly available databases such as HPIDB (Ammari *et al.*, 2016), PHI-base (Urban *et al.*, 2017), PHIDIAS (Xiang *et al.*, 2007) and PHISTO (Durmus *et al.*, 2013) that offer a more generic repository of PHIs and occasionally tools to visualise and analyse PHI data. However, due to an ever increasing volume of the biomedical literature describing the pathogenesis of infectious diseases, the identification of specific PHIs and their roles in pathogenicity is a non-trivial task and therefore, it can be a while before many discoveries are reflected in the PHI databases. A vast amount of PHI data, thus, remains tucked away uncured in a large number of published studies; therefore, the recent years have witnessed a rapid development of computational methods for biomedical literature mining, especially in the context of PHIs (see below).

Some studies have used publicly available computational tools that facilitate the retrieval and extraction of relevant information from the biomedical literature such as PubMed to gather abstracts containing the relevant host and pathogen keywords as well as interaction verbs (including “interact”, “bind”, “attach”, “associate”). These steps were followed by a careful manual curation to extract pairwise PHIs to complement the experimentally determined PHIs (de Chassey *et al.*, 2008; Tripathi *et al.*, 2013). However, such methods are laborious, highly context-specific and difficult to employ on a larger scale and for extracting PHIs in general. To address the challenges of extracting PHIs from the literature, different methods using text mining have been employed to extract PHIs from the biomedical literature (Durmus *et al.*, 2015). These have included methods that focus on gathering and selecting literature abstracts that contain PHI-related information; Yin *et al.* (2010) were the first to report a machine learning-based automated text-mining system to identify research articles describing PHIs. Subsequently, Thieu *et al.* (2012) developed a hybrid procedure encompassing machine learning and language-based approaches to perform a more rigorous

Table 1 A selection of computational approaches for prediction and analysis of PHIs

S. No.	Method	Type of method	Pathogenic species	Reference
1.	Yin <i>et al.</i> – Document classification	Text-mining	All	Yin <i>et al.</i> (2010)
2.	Thieu <i>et al.</i> – Semantic analysis with machine learning and language processing	Text-mining	All	Thieu <i>et al.</i> (2012)
3.	Cui <i>et al.</i> – Protein sequence information	Machine learning using SVM classifier	Viruses	Cui <i>et al.</i> (2012)
4.	Dyer <i>et al.</i> – Protein domain profiles	Bayesian statistics	<i>Plasmodium falciparum</i>	Dyer <i>et al.</i> (2007)
5.	de Chassey <i>et al.</i> – Protein structure and interactome information	Structural homology and interaction redundancy	Viruses – Influenza virus N1 protein	de Chassey <i>et al.</i> (2013b)
6.	Coelho <i>et al.</i> – Ensemble approach using literature mining, sequence, functional overlap, orthology and domain interactions	Machine learning using naïve Bayes classifier	Oral microbiome	Coelho <i>et al.</i> (2014)
7.	Kshirsagar <i>et al.</i> – Transfer learning based on homology	Combinatorial supervised machine learning	Bacteria – <i>Salmonella</i> , <i>Yersinia</i>	Kshirsagar <i>et al.</i> (2012)
8.	Kshirsagar <i>et al.</i> – Combinatorial feature selection for model building	Multitask learning	Bacteria – <i>Y. pestis</i> , <i>F. tularensis</i> , <i>Salmonella</i> , <i>B. anthracis</i>	Kshirsagar <i>et al.</i> (2013)
9.	Zhou <i>et al.</i> – Protein sequence information and amino acid properties	Homology based PPI prediction	<i>M. tuberculosis</i>	Zhou <i>et al.</i> (2014)
10.	Han <i>et al.</i> – Interlog and protein domain-domain interaction	PPI network-based prediction	<i>Bacillus licheniformis</i>	Han <i>et al.</i> (2016)

semantic analysis of the biomedical literature to extract PHIs. The development of newer and more sophisticated methods that take into account the unique aspects of different PHIs (Karadeniz *et al.*, 2015) for PHI extraction from the biomedical literature will not only improve and the visibility and coverage of PHIs with the researchers and existing PHI repositories, but also complement PHIN-centric analyses to better understand the biology of pathogenesis.

In Silico Prediction of PHIs

Despite the increasing availability of experimentally defined PHIs, much of the available data provides only a partial coverage of all the probable PHIs and that too mostly for well-studied pathogens. The limited coverage of experimentally-determined PHIs is chiefly because these methods are resource-intensive and time consuming and thus, the construction of PHI interactomes using experimental methods is technically challenging. Therefore, it is of significant importance to develop new computational methods that can build inter-species PPI prediction models to prioritise the experimental scope and accelerate the discovery of PHIs (Nourani *et al.*, 2015; Halder *et al.*, 2017).

Many different approaches for *in silico* prediction of PHIs have been developed; they variously utilise protein sequence information (Cui *et al.*, 2012), protein domain profiles (Dyer *et al.*, 2007), protein structural properties and structural similarity (Franzosa and Xia, 2011; de Chassey *et al.*, 2013b) and principles of homology and “interologues” based on known PHIs (Coelho *et al.*, 2014) among others, using classical machine learning approaches. These “classical” methods work well enough where sufficient training data are available, but as often is the case with PHIs, there is a distinct scarcity of high-quality PHI data, especially for many less studied pathogens, and this problem contributes to a lack of sufficient PHI features for supervised machine learning (Kshirsagar *et al.*, 2012). More recently, methods that employ integrative genomics data and more sophisticated machine-learning frameworks, such as multitask learning that aims to simultaneously examine different related tasks (in this instance sufficiently related pathogenic diseases) and to extract common features and combine them into a predictive model, are becoming increasingly popular for predicting PHIs (Kshirsagar *et al.*, 2013).

However, the bulk of PHI prediction methods remain focused on species-specific PHIs. To name a few, Zhou *et al.* (2014) employed a homology-based approach to predict host-pathogen PPIs between human and a tuberculosis-causing bacterial pathogen *M. tuberculosis* and identified key properties of the host and pathogen proteins involved in these PPIs. Han *et al.* (2016) constructed a computationally predicted PPIN of *Bacillus licheniformis* and identified novel protein complexes and assigned functional annotations for previously uncharacterised proteins.

Conclusions

Studies on PHIs have observed common themes underlying the onset of various human diseases associated with pathogenic infection in humans, a better understanding of which may be helpful in optimising broad spectrum approaches to counteracting a wide range of pathogenic infections.

A deeper analysis and understanding of PHIs will considerably aid in the identification of new clinically relevant targets for optimising the therapeutic strategies to manipulate PHIs and thus, more effectively combat infectious diseases. Moreover, the genetic variability of pathogenic, especially viral genomes has facilitated the emergence of drug resistance against drugs that target pathogenic components. Therefore, antibacterials and antivirals that target less mutable host proteins that are critical to pathogenesis, preferably with minimal adverse side effects, may provide attractive alternatives to existing therapies. Additional studies on the evolution of PHIs along with the evolution of the pathogens and hosts themselves, should also contribute to developing better strategies for inhibiting PHIs for newer and better therapies.

See also: Algorithms for Graph and Network Analysis: Clustering and Search of Motifs in Graphs. Algorithms for Graph and Network Analysis: Graph Alignment. Algorithms for Graph and Network Analysis: Graph Indexes/Descriptors. Algorithms for Graph and Network Analysis: Traversing/Searching/Sampling Graphs. Genome Analysis – Identification of Genes Involved in Host-Pathogen Protein-Protein Interaction Networks. Host-Pathogen Interactions. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Networks in Biology. Protein-Protein Interaction Databases. Vaccine Target Discovery

References

- Amari, M.G., *et al.*, 2016. HPIDB 2.0: A curated database for host-pathogen interactions. Database 2016, baw103.
 Bhavsar, A.P., Guttman, J.A., Finlay, B.B., 2007. Manipulation of host-cell pathways by bacterial pathogens. Nature 449 (7164), 827–834.
 Brito, A.F., Pinney, J.W., 2017. Protein-protein interactions in virus-host systems. Front. Microbiol. 8, 1557.
 Buckholz, R.G., *et al.*, 1999. Automation of yeast two-hybrid screening. J. Mol. Microbiol. Biotechnol. 1 (1), 135–140.
 Burckstummer, T., *et al.*, 2006. An efficient tandem affinity purification procedure for interaction proteomics in mammalian cells. Nat. Methods 3 (12), 1013–1019.

- Calderone, A., Licata, L., Cesareni, G., 2015. VirusMentha: A new resource for virus-host protein interactions. *Nucleic Acids Res.* 43 (Database issue), D588–D592.
- Calderwood, M.A., *et al.*, 2007. Epstein-Barr virus and virus human protein interaction maps. *Proc. Natl. Acad. Sci. USA* 104 (18), 7606–7611.
- de Chassey, B., *et al.*, 2008. Hepatitis C virus infection protein network. *Mol. Syst. Biol.* 4, 230.
- de Chassey, B., *et al.*, 2013a. The interactomes of influenza virus NS1 and NS2 proteins identify new host factors and provide insights for ADAR1 playing a supportive role in virus replication. *PLOS Pathog.* 9 (7), e1003440.
- de Chassey, B., *et al.*, 2013b. Structure homology and interaction redundancy for discovering virus-host protein interactions. *EMBO Rep.* 14 (10), 938–944.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2011. TargetMine, an integrated data warehouse for candidate gene prioritisation and target discovery. *PLOS ONE* 6 (3), e17844.
- Chen, Y.A., Tripathi, L.P., Mizuguchi, K., 2016. An integrative data analysis platform for gene set analysis and knowledge discovery in a data warehouse framework. *Database* 2016.
- Coelho, E.D., *et al.*, 2014. Computational prediction of the human-microbial oral interactome. *BMC Syst. Biol.* 8, 24.
- Cui, G., Fang, C., Han, K., 2012. Prediction of protein-protein interactions between viruses and human by an SVM model. *BMC Bioinform.* 13 (Suppl 7), S5.
- Diamond, D.L., *et al.*, 2012. Proteome and computational analyses reveal new insights into the mechanisms of hepatitis C virus-mediated liver disease posttransplantation. *Hepatology* 56 (1), 28–38.
- Dolan, P.T., *et al.*, 2013. Identification and comparative analysis of hepatitis C virus-host cell protein interactions. *Mol. Biosyst.* 9 (12), 3199–3209.
- Dunham, W.H., Mullin, M., Gingras, A.C., 2012. Affinity-purification coupled to mass spectrometry: Basic principles and strategies. *Proteomics* 12 (10), 1576–1590.
- Durmus, S., *et al.*, 2015. A review on computational systems biology of pathogen-host interactions. *Front. Microbiol.* 6, 235.
- Durmus, T.S., *et al.*, 2013. PHISTO: Pathogen-host interaction search tool. *Bioinformatics* 29 (10), 1357–1358.
- Durmus, S., Ulgen, K.O., 2017. Comparative interactomics for virus-human protein-protein interactions: DNA viruses versus RNA viruses. *FEBS Open Bio* 7 (1), 96–107.
- Dyer, M.D., Murali, T.M., Sobral, B.W., 2007. Computational prediction of host-pathogen protein-protein interactions. *Bioinformatics* 23 (13), i159–i166.
- Friedel, C.C., Haas, J., 2011. Virus-host interactomes and global models of virus-infected cells. *Trends Microbiol.* 19 (10), 501–508.
- Fels, U., Gevaert, K., Van Damme, P., 2017. Proteogenomics in aid of host-pathogen interaction studies: A bacterial perspective. *Proteomes* 5 (4), 26.
- Fields, S., Song, O., 1989. A novel genetic system to detect protein-protein interactions. *Nature* 340 (6230), 245–246.
- Franzosa, E.A., Xia, Y., 2011. Structural principles within the human-virus protein-protein interaction network. *Proc. Natl. Acad. Sci. USA* 108 (26), 10538–10543.
- Friedel, C.C., 2011. Virus-host interactomes and global models of virus-infected cells. *Trends Microbiol.* 19 (10), 501–508.
- Fu, W., *et al.*, 2009. Human immunodeficiency virus type 1, human protein interaction database at NCBI. *Nucleic Acids Res.* 37 (Database issue), D417–D422.
- Guirimand, T., Delmotte, S., Navratil, V., 2015. VirHostNet 2.0: Surfing on the web of virus/host molecular interactions data. *Nucleic Acids Res.* 43 (Database issue), D583–D587.
- Halder, A.K., *et al.*, 2017. Review of computational methods for virus-host protein interaction prediction: A case study on novel Ebola-human interactions. *Brief. Funct. Genomics*.
- Halehalli, R.R., Nagarajaram, H.A., 2015. Molecular principles of human virus protein-protein interactions. *Bioinformatics* 31 (7), 1025–1033.
- Han, Y.C., *et al.*, 2016. Prediction and characterization of protein-protein interaction network in *Bacillus licheniformis* WX-02. *Sci. Rep.* 6, 19486.
- Jager, S., *et al.*, 2011. Global landscape of HIV-human protein complexes. *Nature* 481 (7381), 365–370.
- Karadeniz, I., *et al.*, 2015. Literature mining and ontology based analysis of Host-Brucella Gene-Gene Interaction Network. *Front. Microbiol.* 6, 1386.
- Kshirsagar, M., Carbonell, J., Klein-Seetharaman, J., 2012. Techniques to cope with missing data in host-pathogen protein interaction prediction. *Bioinformatics* 28 (18), i466–i472.
- Kshirsagar, M., Carbonell, J., Klein-Seetharaman, J., 2013. Multitask learning for host-pathogen protein interactions. *Bioinformatics* 29 (13), i217–i226.
- Li, Y., *et al.*, 2015. VirBase: A resource for virus-host ncRNA-associated interactions. *Nucleic Acids Res.* 43 (Database issue), D578–D582.
- Lum, K.K., Cristea, I.M., 2016. Proteomic approaches to uncovering virus-host protein interactions during the progression of viral infection. *Expert Rev. Proteomics* 13 (3), 325–340.
- Luo, Y., *et al.*, 1997. Mammalian two-hybrid system: A complementary approach to the yeast two-hybrid system. *Biotechniques* 22 (2), 350–352.
- Ma-Lauer, Y., *et al.*, 2012. Virus-host interactomes – Antiviral drug discovery. *Curr. Opin. Virol.* 2 (5), 614–621.
- Maruta, N., Trusov, Y., Botella, J.R., 2016. Yeast three-hybrid system for the detection of protein-protein interactions. *Methods Mol. Biol.* 1363, 145–154.
- Murakami, Y., *et al.*, 2017. Network analysis and in silico prediction of protein-protein interactions with applications in drug discovery. *Curr. Opin. Struct. Biol.* 44, 134–142.
- Nicod, C., Banaei-Esfahani, A., Collins, B.C., 2017. Elucidation of host-pathogen protein-protein interactions to uncover mechanisms of host cell rewiring. *Curr. Opin. Microbiol.* 39, 7–15.
- Nourani, E., Khunjush, F., Durmus, S., 2015. Computational approaches for prediction of pathogen-host protein-protein interactions. *Front. Microbiol.* 6, 94.
- Raman, K., 2010. Construction and analysis of protein-protein interaction networks. *Autom. Exp. 2* (1), 2.
- Shapira, S.D., *et al.*, 2009. A physical and regulatory map of host-influenza interactions reveals pathways in H1N1 infection. *Cell* 139 (7), 1255–1267.
- Snider, J., *et al.*, 2010. Detecting interactions with membrane proteins using a membrane two-hybrid assay in yeast. *Nat. Protoc.* 5 (7), 1281–1293.
- Tabata, K., *et al.*, 2016. Unique requirement for ESCRT factors in flavivirus particle formation on the endoplasmic reticulum. *Cell Rep.* 16 (9), 2339–2347.
- Thieu, T., *et al.*, 2012. Literature mining of host-pathogen interactions: Comparing feature-based supervised learning and language-based approaches. *Bioinformatics* 28 (6), 867–875.
- Trinkle-Mulcahy, L., *et al.*, 2008. Identifying specific protein interaction partners using quantitative mass spectrometry and bead proteomes. *J. Cell Biol.* 183 (2), 223–239.
- Tripathi, L.P., *et al.*, 2010. Network based analysis of hepatitis C virus Core and NS4B protein interactions. *Mol. Biosyst.* 6 (12), 2539–2553.
- Tripathi, L.P., *et al.*, 2012. Proteomic analysis of hepatitis C virus (HCV) core protein transfection and host regulator PA28gamma knockout in HCV pathogenesis: A network-based study. *J. Proteome Res.* 11 (7), 3664–3679.
- Tripathi, L.P., *et al.*, 2013. Understanding the biological context of NS5A-host interactions in HCV infection: A network-based approach. *J. Proteome Res.* 12 (6), 2537–2551.
- Uetz, P., *et al.*, 2006. Herpesviral protein networks and their interaction with the human proteome. *Science* 311 (5758), 239–242.
- Urban, M., *et al.*, 2017. PHI-base: A new interface and further additions for the multi-species pathogen-host interactions database. *Nucleic Acids Res.* 45 (D1), D604–D610.
- Wattam, A.R., *et al.*, 2014. PATRIC, the bacterial bioinformatics database and analysis resource. *Nucleic Acids Res.* 42 (Database issue), D581–D591.
- Xiang, Z., Tian, Y., He, Y., 2007. PHIDIAS: A pathogen-host interaction data integration and analysis system. *Genome Biol.* 8 (7), R150.
- Yin, L., *et al.*, 2010. Document classification for mining host pathogen protein-protein interactions. *Artif. Intell. Med.* 49 (3), 155–160.
- Zhou, H., *et al.*, 2014. Stringent homology-based prediction of *H. sapiens*-*M. tuberculosis* H37Rv protein-protein interactions. *Biol. Direct* 9, 5.

Phylogenetic Analysis: Early Evolution of Life

Himakshi Sarma, Sushmita Pradhan, and Venkata SK Mattaparthi, Tezpur University, Tezpur, India

Sandeep Kaushik, European Institute of Excellence on Tissue Engineering and Regenerative Medicine, Guimaraes, Portugal

© 2019 Elsevier Inc. All rights reserved.

Introduction

The earth formed about 4.5 billion years ago had no signs of life, until 1 billion years later, when Earth saw the emergence of life, the evidence of which can be found in fossils dating back to 3.5 billion years ago (Brasier *et al.*, 2006; Dalrymple, 2001; Dodd *et al.*, 2017; Mojzsis *et al.*, 1996; Schopf and Packer, 1987). Since then, life on earth has seen three major phases of evolution. The first was the appearance of prokaryotic organisms (single celled organism having undefined nucleus), then came eukaryotes (highly organized single cell with defined nucleus and chromosomes with an arranged DNA), and finally the evolution of multicellular organisms (combination of eukaryotic cells), which occurred mainly in the Precambrian period (Knoll, 2003, 2015; Schopf, 2001). Thus, evolution can be defined as a transition or adaptation of an organism, in response to the environmental conditions over the course of time. Evolution can be classified into three types: Chemical evolution (Oró, 1983), molecular evolution, and Darwinian evolution. Chemical evolution majorly denotes the prebiotic (before biology) formation of organic molecules in the period between the formation of the earth and the first appearance of a living cell. Molecular evolution refers to the changes in living systems that have occurred thereafter, eventually leading to the formation of unicellular and multicellular organisms. Darwinian evolution refers to the events after molecular evolution, which led to the speciation as seen today on Earth. Fig. 1 provides a schematic representation of the evolutionary changes that took place (Pattabhi and Gautham, 2002).

Stages of Evolution

Chemical Evolution (Prebiotic Earth)

The current understanding of the conditions on prebiotic Earth and the idea of gradual chemical evolution of life, otherwise known as abiogenesis (spontaneous generation), were put forth independently by Oparin and Haldane (see references in Fox, 1974).

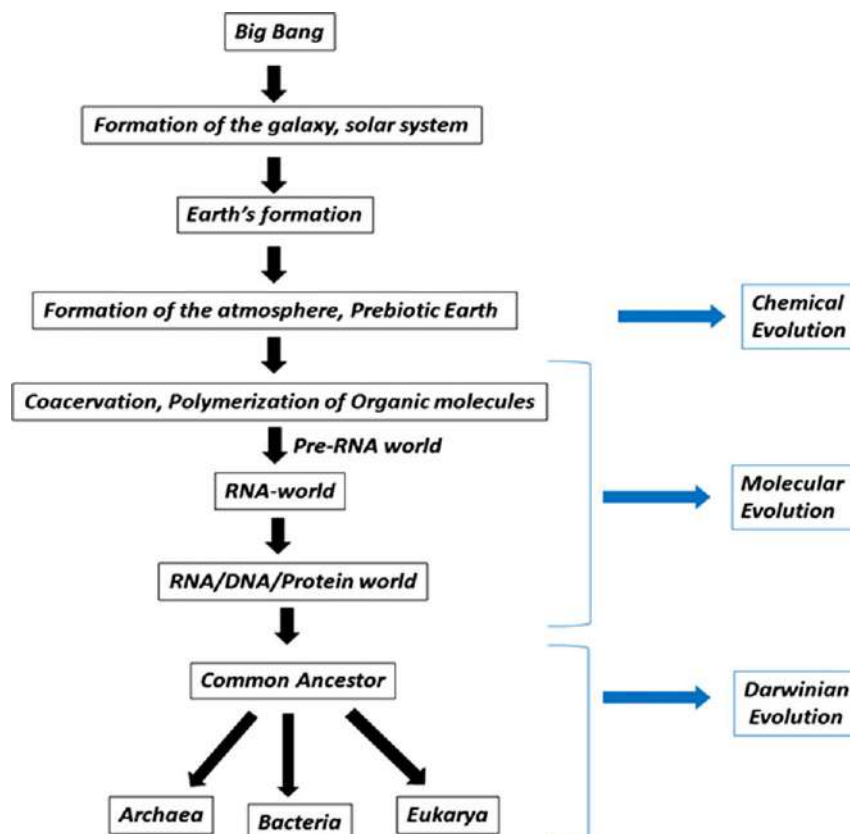


Fig. 1 Beginning of life on Earth.

Spontaneous generation

Oparin and Haldane were the first ones to revive the idea of abiogenesis. Haldane, in particular, was fascinated with the experiments where organic molecules were formed from the mixture of H_2O , CO_2 , and NH_3 in the presence of ultraviolet light. Haldane further suggested that the strong reducing conditions may have been present at that time, which paved the way for the atmosphere of the early earth. Haldane's article in 1929 recommended that the earth's prebiotic soup would have formed first, after which organic mixes of simple sugars and amino acids came into being (Haldane, 1929). Oparin, on the other hand, stated that the primordial soup consisting of organic molecules could be created in an anaerobic atmosphere in the presence of sunlight, wherein the coacervate droplets would be formed as a result of the complex reaction undergone by the organic molecules (Oparin, 1924). These coacervate droplets would then merge with the neighbor droplets, and reproduce by the fission process (Fox, 1974). Inspired by the strong arguments of Oparin and Haldane, Stanley Miller, along with his graduate advisor Harold Urey, tested Oparin–Haldane's early earth hypothesis in 1953 (Miller, 1953; Miller and Urey, 1959).

Miller–Urey experiment

The Miller–Urey experiment established the capability of the earth's primitive atmosphere in building life from inorganic materials. It proved that the conditions existing in the earth's primitive atmosphere were sufficient enough to form organic molecules like amino acids. Their experimental steps consisted of (1) assemblage of a reducing atmosphere rich in hydrogen, and devoid of gaseous oxygen; (2) reducing atmospheres of H_2 over water at ocean's brim; (3) maintenance of H_2 and H_2O mixture below 100°C ; and (4) trigger lightning in the form of sparks.

In order to prove the hypothesis, they placed the molecules that may have been present at the early stages of the earth's time, like methane (CH_4), hydrogen (H_2), and ammonia (NH_3) gases into a moist locale above a water-containing flask (mock ocean of water). To mimic the primordial lightning, they used electrical current for the setup. A few days later, they could observe organic compounds in the flask, some of which were amino acids, the precursor of proteins. Using chromatographic techniques, they were able to confirm the presence of amino acids (glycine, alanine, aspartic acid, glutamic acid, β -alanine), purine and pyrimidines, and other organic compounds such as urea, lactic acid, glycolic acid, succinic acid, and propionic acid. The Miller–Urey experiment was able to successfully validate the emergence of biological molecules from the nonbiological substances (Miller, 1953; Miller and Urey, 1959) (Fig. 2).

Molecular Evolution (Complex Biological Molecules and Protocells)

Though Miller–Urey demonstrated the formation of simple biomolecules, the actual chemical evolution would depend upon the polymerization or condensation of these monomers into polymers (macromolecules like polypeptides, polysaccharides). For example, nucleic acid bases can be synthesized under prebiotic conditions (condensation of HCN and H_2O catalyzed by NH_3), while the sugars can be synthesized by the polymerization of formaldehyde (CH_2O) catalyzed by the divalent cations, alumina, or clay (Miller, 1953; Miller and Urey, 1959). But how these molecules organized themselves to form a cell was the next big question.

Sydney Fox and his team in the 1950s synthesized some peptidelike products by heating a dry mixture of amino acids at the temperature between 150 and 180°C , which they termed as proteinoids, meaning a thermal protein. Unlike the default linear

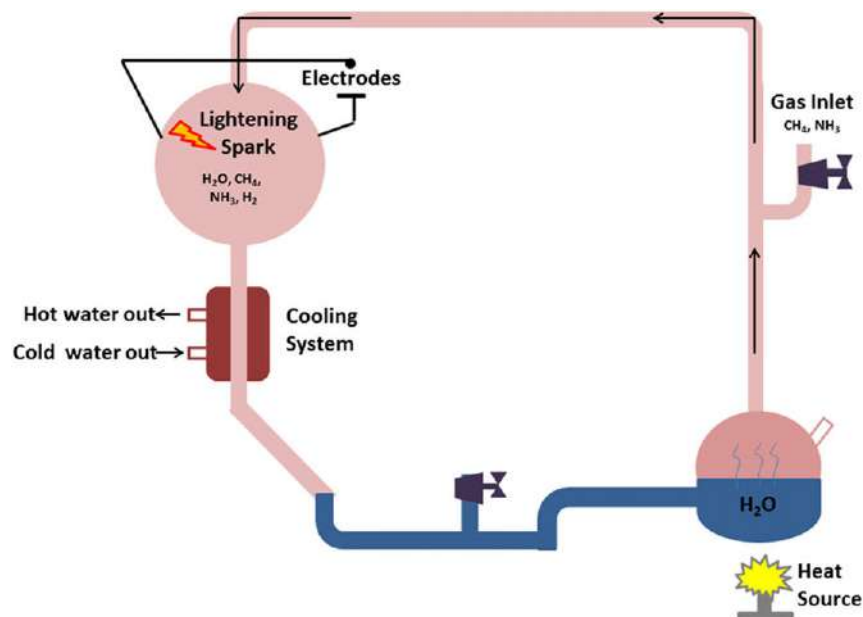


Fig. 2 Schematic representation of Miller–Urey's experiment.

proteins that are produced by the cell, proteinoids are highly branched polypeptides, exhibiting catalytic activity, and susceptibility to digestion by proteases, which are the properties shown only by the biological proteins (Fox *et al.*, 1959).

Origin of cells and organized structure

Macromolecules, due to their intermolecular forces, may have aggregated to form the membrane-enclosed droplets known as coacervates and microspheres, which exhibits features similar to that of a living system (selective permeability, and energy use). Coacervates (meaning “to assemble together or cluster”) are an agglomeration of colloidal particles in a liquid phase that stays in the form of minuscule droplets (Fig. 3). Coacervates display a number of compelling features:

- exchange substance with the environment,
- increases in size,
- selective concentration of the compounds within them (De Jong and Kruyt, 1929).

Oparin had earlier proposed that coacervates may have been the intermediate stage between loose molecules and living systems. He observed that droplets survived longer when they carried out polymerization reactions. Microscopic membrane-bound spheres (termed microspheres), however, are formed when proteinoids are boiled in water and cooled off. These microspheres are uniform in size, stable, and double membrane-bound, resembling a typical cell, and can multiply by fission and budding. They also present the properties of osmosis, growth in size, and selective absorption of chemicals, which are usually associated with normal cells. This spontaneous self-congregation of macromolecules into coacervates and microspheres is indicative of the fact that the contingency of similar structures under those primitive atmospheres may have probably led to the development of more organized and defined membrane-bound structure containing molecules, called protocells. Later, protocells may have been subjected to natural selection, which acted as a driving force in the origination of much more dominant and favorable molecules, and structures (Oparin, 2003).

Membrane defined the first cell

This idea was first proposed by Haldane (1929), saying that “the cell consists of numerous half-living chemical molecules suspended in water and enclosed in an oily film. When the whole sea was a vast chemical laboratory the conditions for the formation of such films must have been relatively favourable....”. Goldacre (1958) went on to suggest that the first cell membranes may be formed as a result of the wave-action behavior of the lipid-like surfactants. Bangham *et al.* (1965) were the first group to prove that phospholipids readily formed lipid-bilayer vesicles, called liposomes. The development of an outer membrane must have been the path-defining events leading to the formation of the first cell. The need for containment is generally attained by the amphipathic property of any molecule, where the molecule is semihydrophobic and semihydrophilic. When such a molecule is placed in water, they assemble, arranging their hydrophobic halves as close to one another away from water, while keeping their hydrophilic halves exposed to water. These amphipathic molecules readily gather to form bilayers, creating closed vesicles. Several studies have shown the production of the phospholipids using simulated prebiotic conditions from mixtures of fatty acids, glycerol, and phosphate (Hargreaves *et al.*, 1977; Rao *et al.*, 1982).

Majorly all the present-day cells are surrounded by plasma consisting of amphipathic molecules mainly the phospholipids. Hence, it can be assumed that the first membrane-bound cells were formed by the spontaneous assembly of phospholipids molecules from the prebiotic soup, enfolding a self-replicating mixture of RNA and other molecules within (Fig. 4). However, it is

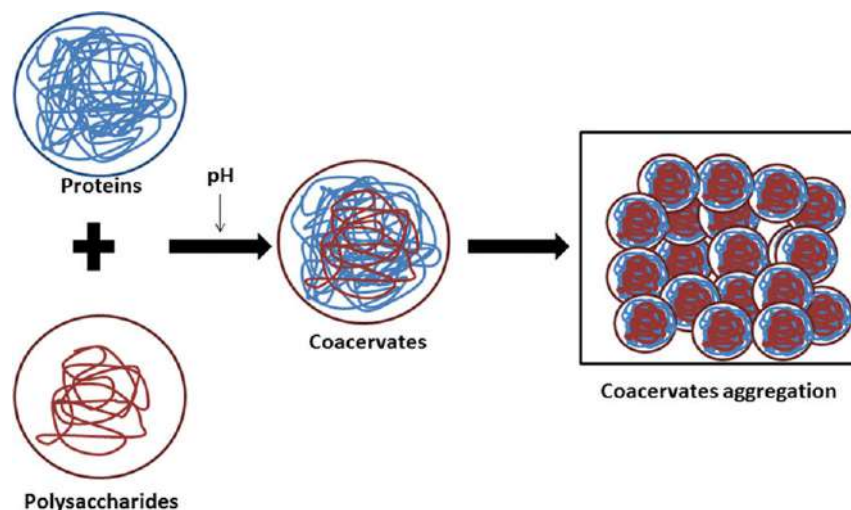


Fig. 3 Coacervation process.

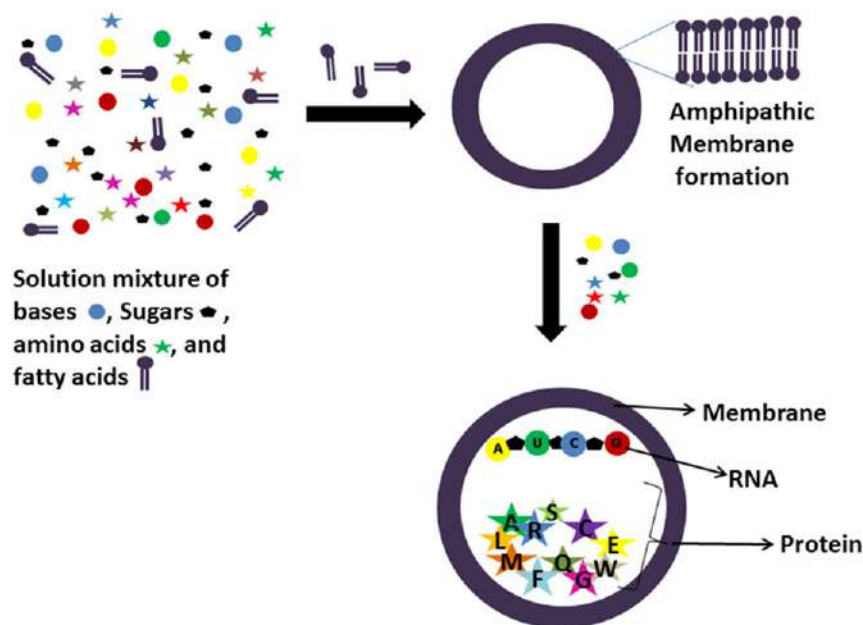


Fig. 4 Formation of the membrane enclosed first cell.

not clear as to when this happened. But, once the RNA molecules were enclosed inside a closed membrane, they may have begun to evolve into carriers of genetic instructions.

Current model for origin of life hypothesis

Many scientists have debated as to what came first – Was it RNA, proteins, or lipids? However, the hypotheses that have been proposed so far depend on the number of discoveries about the origin of molecular and cellular components of life; these are listed in the order of postulated emergence:

Clay hypothesis

The complexity of RNA makes it doubtful as to whether it can be produced nonbiologically or not. So scientists have suggested that some clays have properties to act as enhancers for the creation of the RNA world. This proposal saw very less support among the scientific community (Ferris, 1999). Researchers have demonstrated that clay (montmorillonite) has the ability to enhance the conversion of fatty acids to “bubbles,” which can encapsulate RNA. These bubbles then merge with nearby lipids, and then multiply. This proves that primitive cells may have been formed via similar process (Hanczyc *et al.*, 2003).

Lipid world

Many evolutionists believe that double-walled lipids may have set the initial basis of life on earth as these can readily form liposome structures, and reproduce (Garwood, 2012; Trevors and Psenner, 2001). Though they are devoid of any genetic information, but after passing through selection pressure points, RNA may have formed inside the liposome, thus giving way to the origin of life (Segré *et al.*, 2001).

Iron–sulfur world

A series of experiments successfully demonstrated that using sulfides (iron sulfide and nickel sulfide) as catalysts, one can produce proteins, amino acids from inorganic materials (CO, H₂S). The steps required were maintained at 100°C, with moderate pressures. This suggested that some of the self-sustaining protein may have been produced in the vicinity of hydrothermal vents (Wächtershäuser, 2000).

RNA world

This is the most widely accepted model to explain the initiation of life on Earth. The discovery of ribozymes puts forth that earlier life forms may have been entirely based on RNA. They held the view that RNA may have acted as both template and enzyme (ribozyme), and thus replicated on its own, thereby setting the foundations of life on Earth. This theory was further supported when cofactors, which often carry RNA nucleotides with them, were placed in the argued scenario (Orgel and Crick, 1993; Shapiro, 2007).

Though ribozymes like enzymes were produced in *in vitro* conditions, some were still doubtful of their reality in the early Earth time frame. Many came up with suggestions that there may have been a pre-RNA world before the RNA world, where the earliest ribozymes may have been formed of PNA (peptide nucleic acid), a simpler form of RNA (Böhler *et al.*, 1995; Nelson *et al.*, 2000;

Orgel, 2000). PNA was believed to have been composed of bases bound to a peptide-like backbone, with spontaneous polymerization capabilities. Later, it may have been replaced by RNA that has a sugar-phosphate backbone. They would also have had no speciation, but depended on selection points, and over the years, RNA was replaced by DNA (Hoenigsberg, 2003; Joyce, 2002; Trevors and Abel, 2004).

Another experimental proof to support this RNA theory was provided by Martin and Russell (2003). They proposed that precipitates of some porous metal sulfide can aid in the synthesis of RNA at oceanic pressures and the temperature of about 100°C near hydrothermal vents. This would mean that lipids may have been the last to appear among cell components.

Panspermia

Panspermia theory was first described by the chemist Arrhenius (1903). According to this theory, life arose somewhere in the universe, and these seeds of life (microbial spores) were brought to Earth probably by a meteor or comets. The astronomer duo, Fred Hoyle and Chandra Wickramasinghe, have actively propagated this theory after finding traces of biopolymer in the space dust. They stated that comets containing life forms inside entered the earth about 4 billion years ago, after which life began on earth 4.5 billion years ago. The dust from space has been reported to have all the features that are typical for a microbial occurrence (Hoyle and Wickramasinghe, 1977, 2000). They further believe that life forms have been continuously invading earth from outside, and are causing disease outbreaks and microevolution (Hoyle and Wickramasinghe, 1979). With the proof that the microbes can survive the ultraviolet exposure in space (Clancy *et al.*, 2005; Horneck *et al.*, 2010) and in meteorites (Kvenvolden *et al.*, 1970; McKay *et al.*, 1996), the presence of glycine in Sagittarius B2 (Kuan *et al.*, 2003) and its precursor, “methanimine” in Arp 220 (Salter *et al.*, 2008), the isolation of microbes *Bacillus simplex*, *Staphylococcus pasteurii*, and a fungus *Engyodontium album* via air sampling of the tropopause (Narlikar *et al.*, 2003; Wainwright *et al.*, 2002), and synthesis of DNA and RNA using the compounds found inside a meteorite (Callahan *et al.*, 2011), the panspermia theory has gained more acceptance among scientists. Most recent studies have detected anthracene, naphthalene 700 light-years from the sun, towards the star Cernis 52 (Iglesias-Groth *et al.*, 2008, 2010).

Darwinian Evolution

Darwin was an English naturalist who provided the world with the evidence for the scientific theory of evolution. He defined evolution as “descent with modification,” which states that all species have evolved over the time from a “common ancestor” through the process of natural selection. This theory of natural selection acting as a base point for evolution has been widely accepted by the scientific community, and now has become the basis of modern evolutionary theory.

Darwin’s trip on the *Beagle* and his study on finches led him to theorize the three main principles of evolution in his book *The Origin of Species* (Fig. 5): (1) variation in characters exists among individuals in a given population, (2) these variations can be inherited by the next generation, (3) some forms of inherited traits prove to be advantageous for some individual in terms of survival and reproduction rates than the others (Darwin, 1859, 1871; Darwin and Wallace, 1858). Although Darwin developed his theory of evolution without basics on the molecular basis of life, but even at the molecular level, Darwin’s principles can be observed. For example, when one molecule becomes divergent, giving many variations, such variations at the molecular level can

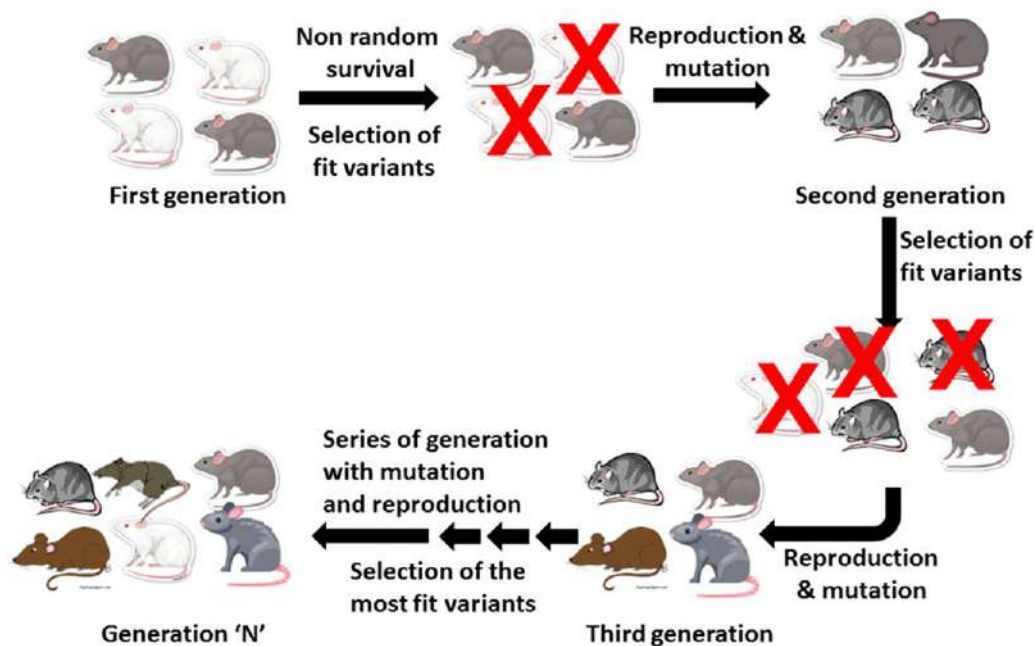


Fig. 5 Process of natural selection.

be induced by various kinds of mutations, genetic drift, etc. at the nucleotide level, which in turn can hamper the protein structure and its function (Hartwell *et al.*, 2014; Patthy, 2009). However, the diversification of the common ancestors to their new progenies are dependent not only upon the natural selection, but also on the mutational and genetic drift effects.

Homology and Development

Richard Owen introduced the term *homolog* in 1843, defining it as “the same organ in different animals under every variety of form and function.” He next defined analogs as a “part or organ in one animal which has the same function as another part or organ in a different animal” (Owen, 1848). Though Darwin never used the term homology, Huxley’s review of Darwin’s book *The Origin of Species* invoked homology as evidence of evolution (Huxley, 1860).

Evolution is directly associated with homology, which is the eventual result of synapomorphies. *Homologs* can be either paralogs or orthologs (Fig. 6), where *paralogs* are homologous sequences separated by a gene duplication event, and *orthologs* are homologous sequences separated by a speciation event (when one species diverges into two) (Fitch, 1970; Panchen, 1994).

Speciation

Speciation is an evolutionary process that gives rise to two or more new and distinct biological species as a result of lineage-splitting from ancestral population. This manifests upon the insufficiency of gene flow between sister populations that directs each population to become irreversibly committed to entirely different evolutionary descent. The primal causes of speciation: Natural selection, genetic drift, and mutation.

Natural selection

Darwin popularized the idea of natural selection stating that speciation was the direct impact of the prolonged action of natural selection upon them. It is a process of diversification in survival and reproduction rate of individuals due to variation in their phenotypes. Here, the organisms in a given population that best adapt to the environment increase in frequency compared to less adapted ones over a number of generations (Darwin, 1859, 1871; Darwin and Wallace, 1858).

Genetic Drift

Genetic drift, often called the Sewall Wright effect, after the individual who set forth the idea of hereditary float, is the adjustment in the recurrence of a previous gene variant (allele) in a populace because of impact of random sampling (Barton and Charlesworth, 1984; Masel, 2011). Drift can cause allele fixation in a population.

Once an allele is fixed, genetic drift ceases; the allele frequency cannot be changed unless another allele is brought into the populace by means of mutation or gene flow. Although both genetic drift and natural selection affect evolution, genetic drift operates randomly, while natural selection functions in a direction that ensures survival and reproductive fitness of that particular organism (Kimura, 1968). One of the best examples of genetic drift is a genetic bottleneck (Charlesworth, 2009).

Genetic bottleneck

This is an evolutionary event where a population reduces drastically to a smaller size as a result of environmental factors, like earthquakes, floods, fires, disease, or droughts, or human activities, like genocide. This causes reduction in the genetic diversity of

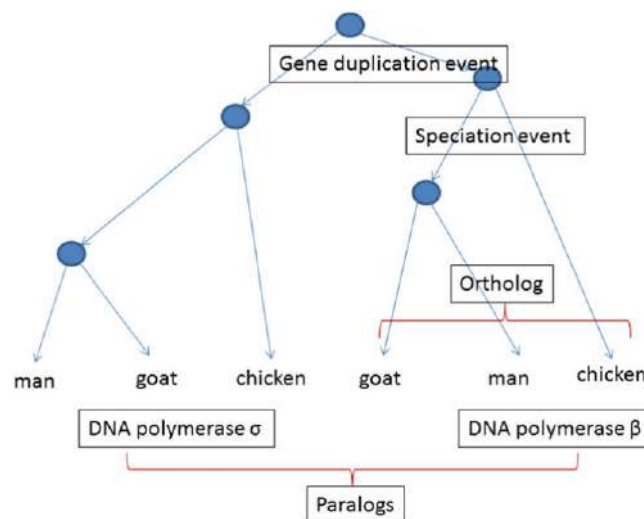


Fig. 6 Orthologs and paralogs.

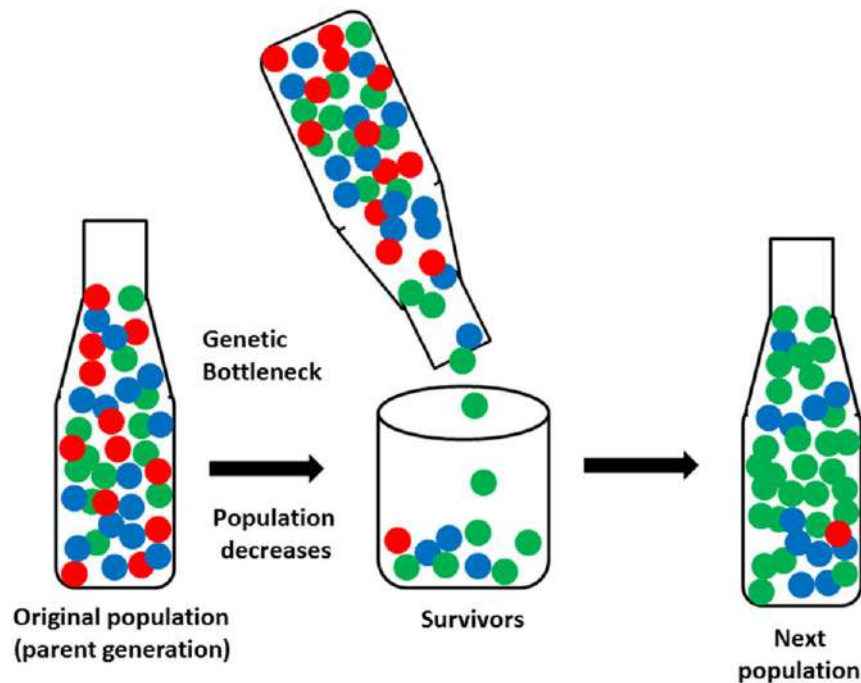


Fig. 7 Bottleneck effect.

the small population, compared to its parent population (Fig. 7). This leads to certain alleles getting overexpressed in the survival population, while other alleles may get underexpressed, or are absent altogether. As a direct aftermath of genetic drift, inbreeding and homogeneity increase in these populations, giving rise to the occurrence of deleterious mutations. For example, antigenic shift of the influenza virus (Webster, 1999), diversification of hyperthermophilic *Pyrococcus* due to genetic drift and natural selection (Escobar-Páramo *et al.*, 2005), etc.

Mutation

Neither selective pressure nor genetic drift can operate in the absence of mutation, which is responsible for the genetic variation (Fig. 8). The randomness of mutation is the prerequisite for Darwinian evolution where changes in the heritable feature in an organism occur randomly, without the interference from the organism's environment. Cairns observed that *E. coli* genes mutated in a such a directed way that it enabled the bacterial population to survive in the different environmental situations (Cairns *et al.*, 1988). Mutations of the ancestral genes can lead to specific adaptations in an individual, which may or may not be beneficial. For example, alcohol dehydrogenases in yeasts that help them to consume accumulated alcohol (Thomson *et al.*, 2005), and gain of pathogenicity in some bacteria (Groisman and Ochman, 1997).

Common Ancestor

Darwin was the first to propose that all living beings on earth share a common descent, saying "Therefore, I should infer from analogy that probably all the organic beings which have ever lived on this earth have descended from some one primordial form, into which life was first breathed" (Darwin, 1859). The common ancestor gave rise to the "last universal common ancestor" (LUCA), which is used to denote the most current common ancestor of all life forms on earth that share a common descent. The last known LUCA was morphologically and metabolically diverse, temperature adaptive, contained an RNA genome, and is said to have lived during the Paleoproterozoic era (Glansdorff *et al.*, 2008), signifying that it was probably similar to extremophiles. Scientists have recently identified 355 genes from the LUCA, by comparing the genomes from archaea, bacteria, and eukaryotes, which constitute the three domain classifications of life (Weiss, 2001). This would mean that even before speciation into three domains of life, LUCA already consisted of the means of conducting complex processes such as transcription and translation, and may have later acquired important genes via horizontal transfer (Gogarten and Deamer, 2016).

Diving deep into the evidences of early forms

It has been reported that the prokaryotes were the earliest forms of life on earth after LUCA. They possessed a single nucleus, DNA, RNA, some proteins and enzymes, had membrane-bound cell walls, but no cell organelles, except for the presence of ribosomes. They were believed to have lived in aquatic environments around 3 billion to 1.5 billion years ago (Penny and Poole, 1999). Below, we discuss the proof of their existence in early earth durations.

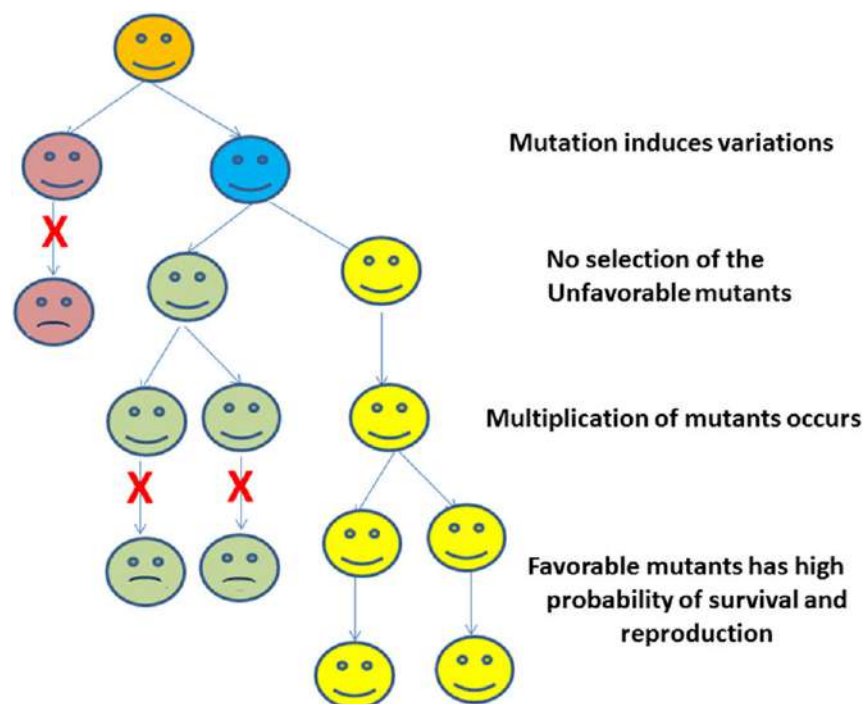


Fig. 8 Effect of mutation in evolution.

Extremophiles

Extremophiles are unicellular microbes (bacteria and archaea) that thrive in extreme environments. Many of these belong to archaea, which are considered to be an ancient bridge between bacteria and eukarya. They can be thermophilic, hyperthermophilic, acidophilic, and alkalophilic, etc. They have been considered to be the closest link to the organism that evolved on the earth (Rampelotto 2010; Rothschild and Mancinelli, 2001).

Thermophiles are heat-loving microbes that grow above 45°C, while hyperthermophiles live beyond 110°C, preferably in between 80 and 121°C. They have been found in hot springs, hydrothermal vents, and the deep subsurface, etc. (Rampelotto 2010; Rothschild and Mancinelli, 2001). Many scientists believe that life on earth started from the hydrothermal vents in the form of thermophiles and hyperthermophiles, due to their ability to resist a wide range of temperature and anaerobic conditions (Li and Kusky, 2007). Wächtershäuser, the iron-sulfur theorist, had identified LUCA inside a black smoker (hydrothermal vent) in the Atlantic Ocean (Huber and Wächtershäuser, 1998). *Pyrococcus fumaris*, one of the highly heat resistant bacteria, which is able to reproduce at 113°C, was also found near the walls of black smokers (Postgate, 1994). Some of the hyperthermophiles studied by Lin *et al.* (2006) were found to have lived in hydrothermal vents in South Africa for about million of years without photosynthesis. *Arqueobacteria*, found near the ancient black smoker chimneys in China are known to live in temperature of about 400°C, depending mainly upon chemosynthesis to produce H₂S (Martin and Russell, 2003). A thermoalkalophilic bacteria, *Thermus brockianus* isolated by Thompson *et al.* (2003) shows catalase activity over wide pH, and temperature range of 6–10 and 30–94°C, respectively. All these instances thus provide strong reinforcement that life may have been indeed originated from the deep ocean beds.

Acidophiles thrive at a low pH, while alkalophiles live in higher pH environments. Though most of the organisms are mesophilic in nature within the pH range of 5–9, acidophiles and alkalophiles have specific adaptations that help them survive an extreme pH range. Acidic environments can be observed in places having geochemical activities, which include the generation of H₂S near in hydrothermal vents, hot springs, coal mine debris, etc. (Rampelotto 2010; Rothschild and Mancinelli, 2001). Alkalophiles can be found in places rich in carbonates, like soda lakes in Egypt, the Rift Valley of Africa, and the western US. Halophiles prefer living in regions that have a high level of salinity, like the Great Salt Lake (US). Hoover *et al.* (2003) identified a novel species of obligate anaerobic, haloalkalophilic spirochete, *Spirocheta Americana*, that prefers living in salty alkaline sediments of California's Mono Lake; it can survive within a temperature range of 10–44°C, salt concentration of 2%–12% (w/v), and pH range 8–10.5.

Another type of extremophile is psychrophiles, which live in extreme cold environments, like the Arctic and Antarctica regions, due to the adaptation of their protein structures to the cold climate. The high amount of glycine in their structure help retain flexibility, while the low amount of charged and hydrophobic amino acids help reduce their inter/intra molecular interactions and allows the protein core to be smaller, in order to avoid freezing (Rampelotto 2010; Rothschild and Mancinelli, 2001). Christner *et al.* (2014) have found some archaean species in West Antarctic Ice Sheet that depended solely on NH₄⁺ and CH₄ for survival, and have been doing so for millions of years in total absence of sunlight or wind. Cryptoendoliths, also known as endoliths, in

particular, survive in the microscopic spaces deep within the rocks, aquifers, and fissures, deriving nutrients via chemosynthesis (Ehrenfreund *et al.*, 2001; Wynn-Williams *et al.*, 2001). These arouse high interest among astrobiologists who theorize that the endolithic regions of Mars and other planets may harbor these organisms. Endoliths have been reported to exist in the permafrost regions of Antarctica (José *et al.*, 2003), the Alps (Horath and Bachofen, 2009), the basalt from the Atlantic, Indian, and Pacific oceans (Lysnes *et al.*, 2004), and the Rocky Mountains (Walker *et al.*, 2005). Radioresistant bacteria are able to resist high levels of radiation (ionizing and nuclear). *Deinococcus radiodurans* can withstand radiation up to 5000 Gy (Gy), with no viability loss. Their radioresistance evolved due to the prolonged cellular desiccation (Mattimore and Battista, 1996). They can survive extreme cold, dehydration, vacuum, and acid, thus are known as a polyextremophile (Anderson, 1956). *D. geothermalis* can resist higher concentrations of gamma radiation, and thrives well in nuclear plants (Ferreira *et al.*, 1997).

Historical Proofs and Scientific Evidences of Early Evolution of Life

Evolution study requires a huge dedication in estimating the most likely number of possible mutations, and the probable elements that factor them. Since the evolution has already occurred, and there is no other means of witnessing these events, so the only way to understand them is to study historical evidence that has been the result of those journeys.

Evolution of Glucocorticoid Receptor

The glucocorticoid receptor (GR) is a known cellular receptor for glucocorticoids (GCs) (cortisol and corticosterone), which are a stress hormone. GCs are used in the treatment of various immune diseases, and this efficacy of the treatment is wholly dependent upon one's sensitivity towards GCs due to the variations in the GR protein that are governed by the GR gene. This particular protein, GR, would not have reached its present-day target function, had it not gone through the two most unlikely permissive mutations in the first place. Though these mutations didn't affect the functional aspect of the protein, they made sure that the protein gained tolerance towards later mutations, so that its sensitivity towards cortisol could be achieved (Carroll *et al.*, 2011; Harms and Thornton, 2014; Orlund *et al.*, 2007).

Harms and Thornton (2010) were able to resurrect the ancestral gene of GR (precursor of modern-day GR and mineralocorticoid (MR) receptors), which existed 450 million years ago, before it evolved its capacity to specifically recognize cortisol. They were able to identify two historical mutations that led to the evolution of GR's specificity. These mutations altered few crucial residues to create novel intermolecular (protein-ligand) and intramolecular contacts. The permissive effect of the mutation helped stabilize the specific section of the receptor, without obstructing protein structure, which in turn allowed it to endure the later function-switching mutations, thus defining the GR's sensitivity towards GC's reaction. Scientists have explored many other alternate theories, but none of them could explain the evolution of GR's sensitivity towards cortisol, except this one.

Evolution of Biological Carbon Fixation

The early ancestors of cyanobacteria are believed to be the ones to start the carbon fixation. When they first evolved their process of photosynthesis, water (H₂O) was used as the electron donor source in the etc pathway for the generation of ATP molecules, as a result of which the autotrophy came into being. However, this ability of cyanobacteria to use water as an electron source created the radical oxygenation of the atmosphere/environment, thereby leading to the large amount of CO₂ consumption (Brasier *et al.*, 2006; Kopp *et al.*, 2005; Tomitani *et al.*, 2006).

There are six known CO₂ fixation pathways that have functional overlaps, but their evolutionary journey and diversification from the parent pathway was not understood well before. It was Braakman and Smith (2012) whose study shed light into this complex diversification. They mapped the evolutionary journey of biological carbon fixation using the metabolic and parsimony-built phylogenetic data analysis, which traced all the modern pathways to a single parent pathway. It was remarked that this diversification was one of the earliest divergences in the tree of life that occurred solely due to the physicochemical changes (energy optimization and oxygen toxicity) in that particular environment. The root of the parsimony-generated phylogenetic tree was observed to be the combination two subnetwork pathway (reductive citric acid cycle and the Wood-Ljungdahl pathway) coalescing into a single network. These trees combine into a single connected network, one being the reductive citric acid cycle and the other being the Wood-Ljungdahl pathway, which is a linear folate-based pathway of the CO₂ reduction pathway. Though these selection pressures cannot be observed in the modern pathways, these were responsible for the primitive diversification of the ancestral carbon fixation.

The Origins of Fermentation, Evolution of Alcohol Dehydrogenases

Alcohol dehydrogenases (Adh) are crucial to the fermentation process, which converts pyruvate to ethanol forming acetaldehyde in between. Yeast was the first organism to have evolved this enzyme, wherein they convert sugars into alcohol using Adh1, and then later consume the accumulated alcohol using Adh2 enzyme, a homolog of Adh1 differing by 24 amino acids. However, research shows that the primitive form of Adh had the ability to only produce alcohol, and not consume them

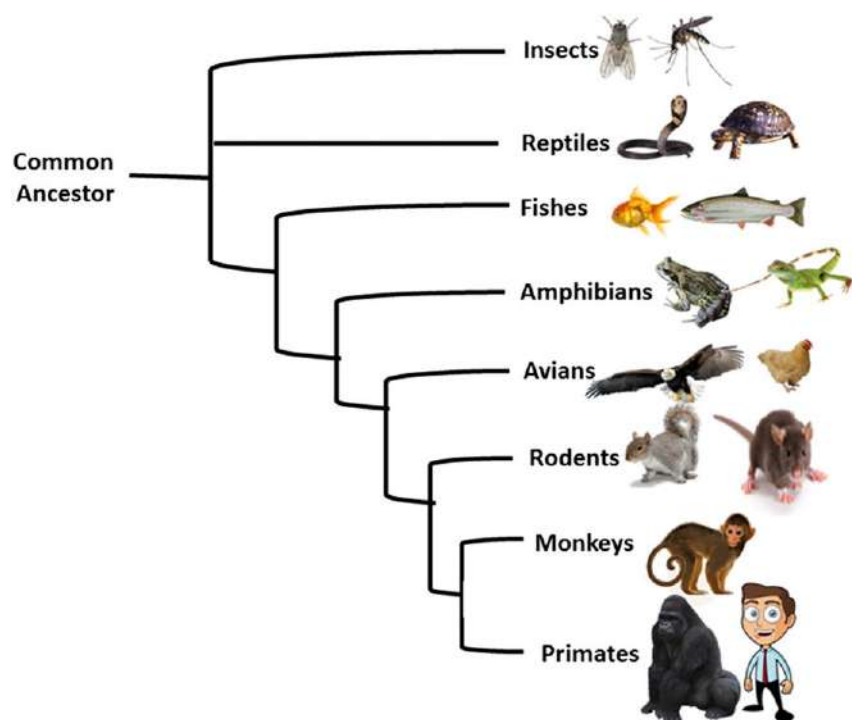


Fig. 9 Phylogenetic tree of contemporary organisms.

(Danielsson and Jönvall, 1992; Gutheil *et al.*, 1992; Persson *et al.*, 2008). Thomson *et al.* (2005) used paleobiochemistry techniques to isolate AdhA, the last common ancestor of both Adh (Adh1 and Adh2) to identify the reason behind this variation. They found that the Adh_A was optimized to only produce alcohol from sugars and recycle the NADH generated from glycolysis. Excess production of alcohol helped yeasts eliminate their competitors, but as the earth evolved, the Cretaceous age brought upon the existence of fleshy fruits. Since fruits contain high amount of alcohols, especially rotting fruits, silent nucleotide dating proposed that yeasts needed a way to metabolize these exogenous alcohols, hence duplicated of Adh into Adh1 and Adh2.

Adaptations of Aromatases in Pigs

Aromatases are cytochrome P450-dependent enzymes that catalyze the conversion of an androgenic steroid (such as testosterone) to an estrogenic steroid (such as estradiol). The protein existed much before the emergence of fish and tetrapods (Callard *et al.*, 1984). Researchers have reported many aromatase genes to date. Teleost fish have two aromatase genes (Callard and Tchoudakova, 1997; Chang *et al.*, 1997), while cattle possess both a functional gene and a pseudogene (Fürbaß and Vanselow, 1995; Hinchelwood *et al.*, 1993). The majority of mammals have only one aromatase gene that codes for the aromatase enzyme through alternative splicing (Boerboom *et al.*, 1997; Delarue *et al.*, 1996, 1998; Harada, 1988; Hickey *et al.*, 1990; Terashima *et al.*, 1991; Simpson *et al.*, 1997).

However, pigs (*Sus scrofa*) have been reported to have three different mRNAs for aromatases (Callard and Tchoudakova, 1997; Choi *et al.*, 1996, 1997a,b; Corbin *et al.*, 1995; Conley *et al.*, 1997). Evidence suggests that these three aromatase gene variants were the result of gene duplication events, which occurred recently in the geologic time scale (Corbin *et al.*, 2004; Graddy *et al.*, 2000). The speculations identifying the functional aspects of these paralogs depend upon the order in which the gene duplication took place. For example, the recent events may have helped domestication of the pigs.

Adaptations of the Elongation Factors

The physical environment to which an ancestral organism may have been exposed can be speculated by resurrecting the protein of that particular organism in *in vitro* conditions, and then study its properties (Adey *et al.*, 1994; Chandrasekharan *et al.*, 1996; Chang *et al.*, 2002; Golding and Dean, 1998; Jermann *et al.*, 1995; Malcolm *et al.*, 1990; Miyazaki *et al.*, 2001). Researchers have been able to revive the protein sequences for Tu family (EF-Tu), the elongation factors that existed in ancient bacteria, and estimated their properties with respect to temperature. The ancient EF-Tu proteins were observed to be functional at the optimal temperature of 55–65°C, which indicates that they were thermophilic in nature, and not hyperthermophilic or mesophilic. This result can be thoroughly corroborated from the dispersion of thermophiles in the bacterial ancestries (Hugenholtz *et al.*, 1998), the G + C content of ancient rRNAs (Galtier *et al.*, 1999), and the geographical record of the molecular clocks (Cavalier-Smith, 2002).

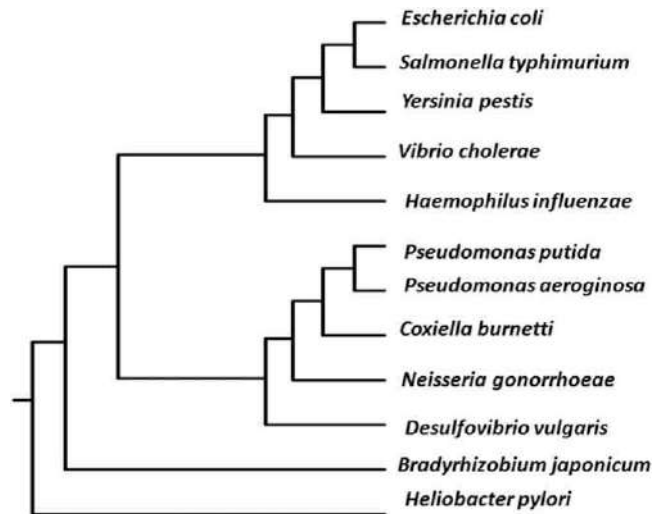


Fig. 10 Phylogenetic tree of bacteria.

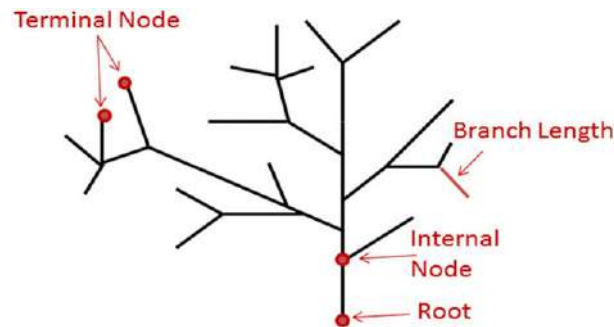


Fig. 11 Elements of a typical phylogram.

Researchers have made use of paleobiochemistry techniques and phylogenetic analysis between many such microorganisms to investigate more about their ancestors.

Tracing the Evolutionary Steps Using Phylogeny

Phylogeny can be described as the relationship between all the organisms on Earth that have descended from a common ancestor, whether they are extinct or extant. Phylogenetics is the science of studying the evolutionary relatedness among biological groups and a phylogenetic tree is used to graphically represent this evolutionary relation related to the species of interest (Figs. 9–11).

More specialized phylogenetic methods have been developed now to meet specific needs, such as species and molecular phylogenetic tests, biogeographic hypotheses, testing, for evaluating amino acids of extinct or extant proteins, establish disease epidemiology and evolution, and even in forensic studies (Linder and Warnow, 2005).

Conclusion

The evolution traces the journey of life on Earth from water to land and air, from an ill-defined membrane-bound cell to modern day multicellular, complex organism over the vast layers of time and generations. With each step of evolution, distinct characters emerged following speciation; some retained their previous features, some developed new sets of traits making contemporary humans curious to understand these changes.

In this encyclopedic article, we have studied the different stages, theories, and evidences of evolution. We shed light on the forces responsible for the genetic variations and speciation. We have also discussed some of the specific adaptations undergone by some of the genes/characters and how these adaptations proved to be beneficial in shaping their survival. Last, but not the least, we have also vaguely touched upon the topic of phylogeny using which the phylogenetic relatedness of any organism can be traced.

Acknowledgement

Firstly, we would like to extend our sincere gratification to Central Library, Tezpur University for lending us their vast resource of study material in order to complete this article.

See also: Genetics and Population Analysis. Natural Language Processing Approaches in Bioinformatics

Reference

- Adey, N.B., Tollefsbol, T.O., Sparks, A.B., Edgell, M.H., Hutchison, C.A., 1994. Molecular resurrection of an extinct ancestral promoter for mouse L1. *Proceedings of the National Academy of Sciences of the United States of America* 91 (4), 1569–1573.
- Anderson, A.W., 1956. Studies on a radioresistant micrococcus. 1. Isolation, morphology, cultural characteristics, and resistance to γ radiation. *Food Technology* 10, 575–578.
- Arrhenius, S., 1903. Die verbreitung des lebens im weltenraum. *Die Umschau* 7, 481–485.
- Bangham, A.D., Standish, M.M., Watkins, J.C., 1965. Diffusion of univalent ions across the lamellae of swollen phospholipids. *Journal of molecular biology* 13 (1), 238–252.
- Barton, N.H., Charlesworth, B., 1984. Genetic revolutions, founder effects, and speciation. *Annual Review of Ecology and Systematics* 15 (1), 133–164.
- Boerboom, D., Kerban, A., Sirois, J., 1997. Molecular characterization of the equine cytochrome P450 aromatase cDNA and its regulation in preovulatory follicles. In: *Biology of Reproduction*, vol. 56. Madison, WI: Soc Study Reproduction, p. 479.
- Böhler, C., Nielsen, P.E., Orgel, L.E., 1995. Template switching between PNA and RNA oligonucleotides. *Nature* 376 (6541), 578–581.
- Braakman, R., Smith, E., 2012. The emergence and early evolution of biological carbon-fixation. *PLOS Computational Biology* 8 (4), e1002455.
- Brasier, M., McLoughlin, N., Green, O., Wacey, D., 2006. A fresh look at the fossil evidence for early Archaean cellular life. *Philosophical Transactions of the Royal Society B: Biological Sciences* 361 (1470), 887–902.
- Cairns, J., Overbaugh, J., Miller, S., 1988. The origin of mutants. *Nature* 335 (6186), 142–145.
- Callahan, M.P., Smith, K.E., Cleaves, H.J., *et al.*, 2011. Carbonaceous meteorites contain a wide range of extraterrestrial nucleobases. *Proceedings of the National Academy of Sciences of the United States of America* 108 (34), 13995–13998.
- Callard, G.V., Pudney, J.A., Kendall, S.L., Reinboth, R., 1984. In vitro conversion of androgen to estrogen in amphioxus gonadal tissues. *General and Comparative Endocrinology* 56 (1), 53–58.
- Callard, G.V., Tchoudakova, A., 1997. Evolutionary and functional significance of two CYP19 genes differentially expressed in brain and ovary of goldfish. *The Journal of Steroid Biochemistry and Molecular Biology* 61 (3–6), 387–392.
- Carroll, S.M., Ortlund, E.A., Thornton, J.W., 2011. Mechanisms for the evolution of a derived function in the ancestral glucocorticoid receptor. *PLOS Genetics* 7 (6), e1002117.
- Cavalier-Smith, T., 2002. The neomuran origin of archaebacteria, the negibacterial root of the universal tree and bacterial megaclassification. *International Journal of Systematic and Evolutionary Microbiology* 52 (1), 7–76.
- Chandrasekharan, U.M., Sanker, S., Glynnias, M.J., Karnik, S.S., Husain, A., 1996. Angiotensin II-forming activity in a reconstructed ancestral chymase. *Science* 271 (5248), 502–505.
- Chang, B.S., Jönsson, K., Kazni, M.A., Donoghue, M.J., Sakmar, T.P., 2002. Recreating a functional ancestral archosaur visual pigment. *Molecular Biology and Evolution* 19 (9), 1483–1489.
- Chang, X.T., Kobayashi, T., Kajiuira, E.H., Nakamura, M., Nagahama, Y., 1997. Isolation and characterization of the cDNA encoding the tilapia (*Oreochromis niloticus*) cytochrome P450 aromatase (P450arom): Changes in P450arom mRNA, protein and enzyme activity in ovarian follicles during oogenesis. *Journal of Molecular Endocrinology* 18 (1), 57–66.
- Charlesworth, B., 2009. Effective population size and patterns of molecular evolution and variation. *Nature Reviews Genetics* 10 (3), 195–205.
- Choi, I., Collante, W.R., Simmen, R.C., Simmen, F.A., 1997a. A developmental switch in expression from blastocyst to endometrial/placental-type cytochrome P450 aromatase genes in the pig and horse. *Biology of Reproduction* 56 (3), 688–696.
- Choi, I.N.H.O., Simmen, R.C., Simmen, F.A., 1996. Molecular cloning of cytochrome P450 aromatase complementary deoxyribonucleic acid from periimplantation porcine and equine blastocysts identifies multiple novel 5'-untranslated exons expressed in embryos, endometrium, and placenta. *Endocrinology* 137 (4), 1457–1467.
- Choi, I., Troyer, D.L., Cornwell, D.L., *et al.*, 1997b. Closely related genes encode developmental and tissue isoforms of porcine cytochrome P450 aromatase. *DNA and Cell Biology* 16 (6), 769–777.
- Christner, B.C., Prisco, J.C., Achberger, A.M., *et al.*, 2014. A microbial ecosystem beneath the West Antarctic ice sheet. *Nature* 512 (7514), 310–313.
- Clancy, P., Brack, A., Horneck, G., 2005. Looking for Life, Searching the Solar System. Cambridge University Press.
- Conley, A., Corbin, J., Smith, T., *et al.*, 1997. Porcine aromatases: Studies on tissue-specific, functionally distinct isozymes from a single gene? *The Journal of Steroid Biochemistry and Molecular Biology* 61 (3–6), 407–413.
- Corbin, C.J., Khalil, M.W., Conley, A.J., 1995. Functional ovarian and placental isoforms of porcine aromatase. *Molecular and Cellular Endocrinology* 113 (1), 29–37.
- Corbin, C.J., Mapes, S.M., Marcos, J., *et al.*, 2004. Paralogues of porcine aromatase cytochrome P450: A novel hydroxylase activity is associated with the survival of a duplicated gene. *Endocrinology* 145 (5), 2157–2164.
- Dalrymple, G.B., 2001. The age of the Earth in the twentieth century: A problem (mostly) solved. *Geological Society, London, Special Publications* 190 (1), 205–221.
- Danielsson, O., Jörnvall, H., 1992. "Enzymogenesis": Classical liver alcohol dehydrogenase origin from the glutathione-dependent formaldehyde dehydrogenase line. *Proceedings of the National Academy of Sciences of the United States of America* 89 (19), 9247–9251.
- Darwin, C., Wallace, A., 1858. On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. *Zoological Journal of the Linnean Society* 3 (9), 45–62.
- Darwin, C., 1871. *The Descent of Man and Selection in Relation to Sex*. London: John Murray.
- Darwin, C., 1859. *The Origin of Species by Means of Natural Selection*. Modern Library.
- Delarue, B., Breard, E., Mitre, H., Leymarie, P., 1998. Expression of two aromatase cDNAs in various rabbit tissues. *The Journal of Steroid Biochemistry and Molecular Biology* 64 (1–2), 113–119.
- Delarue, B., Mitre, H., Féral, C., Benhaim, A., Leymarie, P., 1996. Rapid sequencing of rabbit aromatase cDNA using RACE PCR. *Comptes rendus de l'Academie des sciences. Serie III, Sciences De La Vie* 319 (8), 663–670.
- Dodd, M.S., Papineau, D., Grenne, T., *et al.*, 2017. Evidence for early life in Earth's oldest hydrothermal vent precipitates. *Nature* 543 (7643), 60–64.
- Ehrenfreund, P., Bernstein, M.P., Diworkin, J.P., Sandford, S.A., Allamandola, L.J., 2001. The photostability of amino acids in space. *The Astrophysical Journal Letters* 550 (1), L95–L99.
- Escobar-Páramo, P., Ghosh, S., DiRuggiero, J., 2005. Evidence for genetic drift in the diversification of a geographically isolated population of the hyperthermophilic archaeon *Pyrococcus*. *Molecular Biology and Evolution* 22 (11), 2297–2303.

- Ferreira, A.C., Nobre, M.F., Rainey, F.A., *et al.*, 1997. *Deinococcus geothermalis* sp. nov. and *Deinococcus murrayi* sp. nov., two extremely radiation-resistant and slightly thermophilic species from hot springs. *International Journal of Systematic and Evolutionary Microbiology* 47 (4), 939–947.
- Ferris, J.P., 1999. Prebiotic synthesis on minerals: Bridging the prebiotic and RNA worlds. *The Biological Bulletin* 196 (3), 311–314.
- Fitch, W.M., 1970. Distinguishing homologous from analogous proteins. *Systematic Zoology* 19 (2), 99–113.
- Fox, S.W., 1974. Coacervate droplets, proteinoid microspheres, and the genetic apparatus. In: *The Origin of Life and Evolutionary Biochemistry*. Boston, MA: Springer, pp. 119–132.
- Fox, S.W., Harada, K., Kendrick, J., 1959. Production of spherules from synthetic proteinoid and hot water. *Science* 129 (3357), 1221–1223.
- Fürbaß, R., Vanselow, J., 1995. An aromatase pseudogene is transcribed in the bovine placenta. *Gene* 154 (2), 287–291.
- Galtier, N., Tournasse, N., Gouy, M., 1999. A nonhyperthermophilic common ancestor to extant life forms. *Science* 283 (5399), 220–221.
- Garwood, R., 2012. Patterns in Palaeontology: The first 3 billion years of evolution. *Palaeontology* [Online] 2.
- Glansdorff, N., Xu, Y., Labedan, B., 2008. The last universal common ancestor: Emergence, constitution and genetic legacy of an elusive forerunner. *Biology Direct* 3 (1), 29.
- Gogarten, J.P., Deamer, D., 2016. Is LUCA a thermophilic progenote? *Nature Microbiology* 1 (12), 16229.
- Goldacre, R.J., 1958. Surface films, their collapse on compression, the shapes and sizes of cells, and the origin of life. *Surface Phenomena in Chemistry and Biology* 10.
- Golding, G.B., Dean, A.M., 1998. The structural basis of molecular adaptation. *Molecular Biology and Evolution* 15 (4), 355–369.
- Graddy, L.G., Kowalski, A.A., Simmen, 2000. Multiple isoforms of porcine aromatase are encoded by three distinct genes. *The Journal of steroid biochemistry and molecular biology* 73 (1–2), 49–57.
- Groisman, E.A., Ochman, H., 1997. How *Salmonella* became a pathogen. *Trends in Microbiology* 5 (9), 343–349.
- Gutheil, W.G., Holmquist, B., Vallee, B.L., 1992. Purification, characterization, and partial sequence of the glutathione-dependent formaldehyde dehydrogenase from *Escherichia coli*: A class III alcohol dehydrogenase. *Biochemistry* 31 (2), 475–481.
- Haldane, J.B.S., 1929. The origin of life, *rationalist annual* (reprinted in Haldane, J.B.S., *science and life*, with an introduction by Maynard Smith, J (1968).
- Hanczyc, M.M., Fujikawa, S.M., Szostak, J.W., 2003. Experimental models of primitive cellular compartments: Encapsulation, growth, and division. *Science* 302 (5645), 618–622.
- Harada, N., 1988. Cloning of a complete cDNA encoding human aromatase: Immunochemical identification and sequence analysis. *Biochemical and Biophysical Research Communications* 156 (2), 725–732.
- Hargreaves, W.R., Mulvihill, S.J., Deamer, D.W., 1977. Synthesis of phospholipids and membranes in prebiotic conditions. *Nature* 266 (5597), 78–80.
- Harms, M.J., Thornton, J.W., 2010. Analyzing protein structure and function using ancestral gene reconstruction. *Current Opinion in Structural Biology* 20 (3), 360–366.
- Harms, M.J., Thornton, J.W., 2014. Historical contingency and its biophysical basis in glucocorticoid receptor evolution. *Nature* 512 (7513), 203–207.
- Hartwell, L., Goldberg, M.L., Fischer, J.A., Hood, L., Aquadro, C.F., 2014. *Genetics: From Genes to Genomes*. McGraw-Hill Science/Engineering/Math.
- Hickey, G.J., Krasnow, J.S., Beattie, W.G., Richards, J.S., 1990. Aromatase cytochrome P450 in rat ovarian granulosa cells before and after luteinization: Adenosine 3', 5'-monophosphate-dependent and independent regulation. Cloning and sequencing of rat aromatase cDNA and 5' genomic DNA. *Molecular Endocrinology* 4 (1), 3–12.
- Hinshelwood, M.M., Corbin, C.J., Tsang, P.C., Simpson, E.R., 1993. Isolation and characterization of a complementary deoxyribonucleic acid insert encoding bovine aromatase cytochrome P450. *Endocrinology* 133 (5), 1971–1977.
- Hoenigsberg, H., 2003. Evolution without speciation but with selection: LUCA, the Last Universal Common Ancestor in Gilbert's RNA world. *Genetics and Molecular Research Journal* 2 (4), 366–375.
- Hoover, R.B., Pikuta, E.V., Bej, A.K., *et al.*, 2003. *Spirochaeta americana* sp. nov., a new haloalkaliphilic, obligately anaerobic spirochaete isolated from soda Mono Lake in California. *International Journal of Systematic and Evolutionary Microbiology* 53 (3), 815–821.
- Horath, T., Bachofen, R., 2009. Molecular characterization of an endolithic microbial community in dolomite rock in the central Alps (Switzerland). *Microbial Ecology* 58 (2), 290–306.
- Horneck, G., Klaus, D.M., Mancinelli, R.L., 2010. Space microbiology. *Microbiology and Molecular Biology Reviews* 74 (1), 121–156.
- Hoyle, F., Wickramasinghe, C., 1979. *Diseases from Space*. London: JM Dent, p. 196.
- Hoyle, F., Wickramasinghe, N.C., 1977. Polysaccharides and infrared spectra of galactic sources. *Nature* 268 (5621), 610–612.
- Hoyle, F., Wickramasinghe, N.C., 2000. On the nature of interstellar grains. In: *Astronomical Origins of Life*. Dordrecht: Springer, pp. 249–262.
- Huber, C., Wächtershäuser, G., 1998. Peptides by activation of amino acids with CO on (Ni, Fe) S surfaces: Implications for the origin of life. *Science* 281 (5377), 670–672.
- Hugenholtz, P., Goebel, B.M., Pace, N.R., 1998. Impact of culture-independent studies on the emerging phylogenetic view of bacterial diversity. *Journal of Bacteriology* 180 (18), 4765–4774.
- Huxley, T.H.H., 1860. The origin of species. *Westminster Rev.* 17, 541–570.
- Iglesias-Groth, S., Manchado, A., García-Hernández, D.A., Hernández, J.G., Lambert, D.L., 2008. Evidence for the naphthalene cation in a region of the interstellar medium with anomalous microwave emission. *The Astrophysical Journal Letters* 685 (1), L55–L58.
- Iglesias-Groth, S., Manchado, A., Rebolo, R., *et al.*, 2010. A search for interstellar anthracene towards the Perseus anomalous microwave emission region. *Monthly Notices of the Royal Astronomical Society* 407 (4), 2157–2165.
- Jermann, T.M., Opitz, J.G., Stackhouse, J., Benner, S.A., 1995. Reconstructing the evolutionary history of the aridodactyl ribonuclease superfamily. *Nature* 374 (6517), 57–59.
- De Jong, H.B., Kruyt, H.R., 1929. Coacervation. In: *Proceedings of the Royal Academy of Sciences at Amsterdam*, 32, pp. 849–856.
- José, R., Goebel, B.M., Friedmann, E.I., Pace, N.R., 2003. Microbial diversity of cryptoendolithic communities from the McMurdo Dry Valleys, Antarctica. *Applied and Environmental Microbiology* 69 (7), 3858–3867.
- Joyce, G.F., 2002. The antiquity of RNA-based evolution. *Nature* 418 (6894), 214–221.
- Kimura, M., 1968. Evolutionary rate at the molecular level. *Nature* 217 (5129), 624–626.
- Knoll, A.H., 2003. *Life on a Young Planet: The First Three Billion Years of Evolution on Earth*. Princeton University Press.
- Knoll, A.H., 2015. *Life on a Young Planet: The First Three Billion Years of Evolution on Earth*. Princeton University Press.
- Kopp, R.E., Kirschvink, J.L., Hilburn, I.A., Nash, C.Z., 2005. The Paleoproterozoic snowball Earth: A climate disaster triggered by the evolution of oxygenic photosynthesis. *Proceedings of the National Academy of Sciences of the United States of America* 102 (32), 11131–11136.
- Kuan, Y.J., Charney, S.B., Huang, H.C., Tseng, W.L., Kiesel, Z., 2003. Interstellar glycine. *The Astrophysical Journal* 593 (2), 848–867.
- Kvenvolden, K., Lawless, J., Pering, K., *et al.*, 1970. Evidence for extraterrestrial amino-acids and hydrocarbons in the Murchison meteorite. *Nature* 228 (5275), 923–926.
- Linder, C.R., Warnow, T., 2005. Chapter 19: An overview of phylogeny reconstruction. In: *Aluru, S. (Ed.), Handbook of Computational Molecular Biology*. CRC Press.
- Lin, L.H., Wang, P.L., Rumble, D., *et al.*, 2006. Long-term sustainability of a high-energy, low-diversity crustal biome. *Science* 314 (5798), 479–482.
- Li, J., Kusky, T.M., 2007. World's largest known Precambrian fossil black smoker chimneys and associated microbial vent communities, North China: Implications for early life. *Gondwana Research* 12 (1–2), 84–100.
- Lysnes, K., Torsvik, T., Thorseth, I.H., Pedersen, R.B., 2004. Microbial populations in ocean floor basalt: Results from ODP Leg 187. In: *Pedersen, R.B., Christie, D.M., Miller, D.J. (Eds.), Proceedings of the Ocean Drilling Program, Scientific Results, vol. 187*. College Station, TX: Ocean Drilling Program, pp. 1–27.
- Malcolm, B.A., Wilson, K.P., Matthews, B.W., Kirsch, J.F., Wilson, A.C., 1990. Ancestral lysozymes reconstructed, neutrality tested, and thermostability linked to hydrocarbon packing. *Nature* 345 (6270), 86–89.
- Martin, W., Russell, M.J., 2003. On the origins of cells: A hypothesis for the evolutionary transitions from abiotic geochemistry to chemoautotrophic prokaryotes, and from prokaryotes to nucleated cells. *Philosophical Transactions of the Royal Society B: Biological Sciences* 358 (1429), 59–85.
- Masel, J., 2011. Genetic drift. *Current Biology* 21 (20), R837–R838.

- Mattimore, V., Battista, J.R., 1996. Radioresistance of *Deinococcus radiodurans*: Functions necessary to survive ionizing radiation are also necessary to survive prolonged desiccation. *Journal of Bacteriology* 178 (3), 633–637.
- McKay, D.S., Gibson, E.K., Thomas-Keptra, K.L., *et al.*, 1996. Search for past life on Mars: Possible relic biogenic activity in Martian meteorite ALH84001. *Science* 273 (5277), 924–930.
- Miller, S.L., Urey, H.C., 1959. Organic compound synthesis on the primitive earth. *Science* 130 (3370), 245–251.
- Miller, S.L., 1953. A production of amino acids under possible primitive earth conditions. *Science* 117 (3046), 528–529.
- Miyazaki, J., Nakaya, S., Suzuki, T., *et al.*, 2001. Ancestral residues stabilizing 3-isopropylmalate dehydrogenase of an extreme thermophile: Experimental evidence supporting the thermophilic common ancestor hypothesis. *The Journal of Biochemistry* 129 (5), 777–782.
- Mojzsis, S.J., Arrhenius, G., McKeegan, K.D., *et al.*, 1996. Evidence for life on Earth before 3800 million years ago. *Nature* 384 (6604), 55–59.
- Narlikar, J.V., Lloyd, D., Wickramasinghe, N.C., *et al.*, 2003. A balloon experiment to detect microorganisms in the outer space. *Astrophysics and Space Science* 285 (2), 555–562.
- Nelson, K.E., Levy, M., Miller, S.L., 2000. Peptide nucleic acids rather than RNA may have been the first genetic molecule. *Proceedings of the National Academy of Sciences of the United States of America* 97 (8), 3868–3871.
- Oparin, A., 1924. *Proiskhozhdenie Zhizni* [The Origin of Life]. Moscow, The Origin of Life (1938). New York: The MacMillan Company.
- Oparin, A.I., 2003. *The origin of life*. Courier Corporation.
- Orgel, L., 2000. A simpler nucleic acid. *Science* 290 (5495), 1306–1307.
- Orgel, L.E., Crick, F.H., 1993. Anticipating an RNA world. Some past speculations on the origin of life: Where are they today? *The FASEB Journal* 7 (1), 238–239.
- Oró, J., 1983. Chemical evolution and the origin of life. *Advances in Space Research* 3 (9), 77–94.
- Ortlund, E.A., Bridgman, J.T., Redinbo, M.R., Thornton, J.W., 2007. Crystal structure of an ancient protein: Evolution by conformational epistasis. *Science* 317 (5844), 1544–1548.
- Owen, R., 1848. *On the Archetype and Homologies of the Vertebrate Skeleton*. Van Voorst.
- Panchen, A.L., 1994. Richard Owen and The Concept of Homology. *Homology: The Hierarchical Basis of Comparative Biology*. 1994. San Diego, CA: Academic Press, pp. 21–62.
- Pattabhi, V., Gautham, N., 2002. Origin and evolution of life. *Biophysics*. 232–241.
- Patthy, L., 2009. *Protein Evolution*. John Wiley & Sons.
- Penny, D., Poole, A., 1999. The nature of the last universal common ancestor. *Current Opinion in Genetics & Development* 9 (6), 672–677.
- Persson, B., Hedlund, J., Jörmvall, H., 2008. Medium- and short-chain dehydrogenase/reductase gene and protein families. *Cellular and Molecular Life Sciences* 65 (24), 3879–3894.
- Postgate, J.R., 1994. *The Outer Reaches of Life*. Cambridge University Press.
- Rampelotto, P.H., 2010. Resistance of microorganisms to extreme environmental conditions and its contribution to astrobiology. *Sustainability* 2 (6), 1602–1623.
- Rao, M., Eichberg, J., Oró, J., 1982. Synthesis of phosphatidylcholine under possible primitive Earth conditions. *Journal of Molecular Evolution* 18 (3), 196–202.
- Rothschild, L.J., Mancinelli, R.L., 2001. Life in extreme environments. *Nature* 409 (6823), 1092–1101.
- Salter, C.J., Ghosh, T., Catinella, B., *et al.*, 2008. The arecibo ARP 220 spectral census. I. Discovery of the pre-biotic molecule methanimine and new cm-wavelength transitions of other molecules. *The Astronomical Journal* 136 (1), 299–389.
- Schopf, J.W., Packer, B.M., 1987. Early Archean (3.3-billion to 3.5-billion-year-old) microfossils from Warrawoona Group, Australia. *Science* 237 (4810), 70–73.
- Schopf, J.W., 2001. *Cradle of Life: The Discovery of Earth's Earliest Fossils*. Princeton University Press.
- Segré, D., Ben-Eli, D., Deamer, D.W., Lancet, D., 2001. The lipid world. *Origins of Life and Evolution of the Biosphere* 31 (1–2), 119–145.
- Shapiro, R., 2007. A simpler origin for life. *Scientific American* 296 (6), 46–53.
- Simpson, E.R., Michael, M.D., Agarwal, V.R., *et al.*, 1997. Cytochromes P450 11: Expression of the CYP19 (aromatase) gene: An unusual case of alternative promoter usage. *The FASEB Journal* 11 (1), 29–36.
- Terashima, M., Toda, K., Kawamoto, T., *et al.*, 1991. Isolation of a full-length cDNA encoding mouse aromatase P450. *Archives of Biochemistry and Biophysics* 285 (2), 231–237.
- Thompson, V.S., Schaller, K.D., Apel, W.A., 2003. Purification and characterization of a novel thermo-alkali-stable catalase from *Thermus brockianus*. *Biotechnology Progress* 19 (4), 1292–1299.
- Thomson, J.M., Gaucher, E.A., Burgan, M.F., *et al.*, 2005. Resurrecting ancestral alcohol dehydrogenases from yeast. *Nature Genetics* 37 (6), 630–635.
- Tomitani, A., Knoll, A.H., Cavanaugh, C.M., Ohno, T., 2006. The evolutionary diversification of cyanobacteria: Molecular-phylogenetic and paleontological perspectives. *Proceedings of the National Academy of Sciences of the United States of America* 103 (14), 5442–5447.
- Trevors, J.T., Abel, D.L., 2004. Chance and necessity do not explain the origin of life. *Cell Biology International* 28 (11), 729–739.
- Trevors, J.T., Psenner, R., 2001. From self-assembly of life to present-day bacteria: A possible role for nanocells. *FEMS Microbiology Reviews* 25 (5), 573–582.
- Wächtershäuser, G., 2000. Life as we don't know it. *Science* 289 (5483), 1307–1308.
- Wainwright, M., Wickramasinghe, N.C., Narlikar, J.V., Rajaratnam, P., 2002. Microorganisms cultured from stratospheric air samples obtained at 41 km. *FEMS Microbiology Letters* 218 (1), 161–165.
- Walker, J.J., Spear, J.R., Pace, N.R., 2005. Geobiology of a microbial endolithic community in the Yellowstone geothermal environment. *Nature* 434 (7036), 1011–1014.
- Webster, R.G., 1999. Antigenic variation in influenza viruses. In: Domingo, E., Webster, R., Holland, J. (Eds.), *Origin and Evolution of Viruses*. Elsevier, pp. 377–390.
- Weiss, R.A., 2001. Polio vaccines exonerated. *Nature* 410 (6832), 1035–1036.
- Wynn-Williams, D.A., Newton, E.M., Edwards, H.G.M., 2001. The role of habitat structure for biomolecule integrity and microbial survival under extreme environmental stress in Antarctica (and Mars?): Ecology and technology. *Exo-/Astro-Biology* 496, 225–237.

Biographical Sketch



Himakshi Sarma is a PhD scholar at Molecular Modelling and Simulation Laboratory, Department of Molecular Biology and Biotechnology, Tezpur University, Assam, India. She has done her Post Graduation in Biotechnology from NEHU, Shillong, Meghalaya, in the year 2014. She is expertise on molecular dynamics (MD) simulation of biomolecules, molecular docking, virtual screening, drug designing and in silico alanine scanning mutagenesis (ASM). She is presently working on in silico modeling of Lemur tyrosine kinase (LMTK3), which is a protein kinase implicated in breast cancer progression and metastasis, using modeling and simulation approach.



Sushmita Pradhan is a PhD research scholar at the Molecular Modelling and Simulation Laboratory, Department of Molecular Biology and Biotechnology, Tezpur University, Tezpur, Assam since 2016. She completed her Masters in Microbiology from Bangalore University in the year 2015. She has strong hands on analyzing the biomolecules using on molecular dynamics (MD) simulation, molecular docking, free energy analysis, mutational approaches, etc. She is currently working on Xeroderma pigmentosum A (XPA) protein, functions as the scaffold protein in nucleotide excision repair (NER).



Sandeep Kaushik is a passionate computational biologist with a wide exposure of computational approaches. Presently, he is carrying out his research as an Assistant Researcher (equivalent to Assistant Professor) at 3B's Research Group, University of Minho, Portugal. He has a PhD in Bioinformatics from National Institute of Immunology, New Delhi, India. He has a wide exposure on analysis of scientific data ranging from transcriptomic data on mycobacterial and human samples to genomic data from wheat. His research experience and expertise entails molecular dynamics simulations, RNA-sequencing data analysis using Bioconductor (R language based package), de novo genome assembly using various software like AbySS, automated protein prediction and annotation, database mining, agent-based modeling, and simulations. He has a cumulative experience of more than 10 years of programming using PERL, R language, and NetLogo. He has published his research in reputed international journals like *Molecular Cell*, *Biomaterials*, *Biophysical Journal*, and others. Currently, his research interest involves in silico modeling of disease conditions like breast cancer, using agent-based modeling and simulations approach.



M. V. Satish Kumar received a PhD in Biotechnology from Indian Institute of Technology Guwahati, Assam, India in 2010. From 2010 to 2012, he was with the Centre for Condensed Matter Theory (CCMT), Department of Physics, Indian Institute of Science Bangalore, India. He is currently working as Assistant Professor in the Department of Molecular Biology and Biotechnology, School of Sciences, Tezpur University, Assam, India. His research interests include understanding the functioning of intrinsically disordered proteins (IDPs), pathways of protein aggregation mechanism in neurodegenerative diseases like Alzheimer, Parkinson, etc., and also the features of protein–RNA interactions using computational techniques.

Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material

Suhaila Sulaiman, Nur S Yusoff, Ng S Mun, Haslina Makmur, and Mohd Firdaus-Raih, Universiti Kebangsaan Malaysia, Bangi, Malaysia

© 2019 Elsevier Inc. All rights reserved.

Introduction

Horizontal gene transfer (HGT) is defined as the movement of genetic materials between distantly related organisms (Crisp *et al.*, 2015). HGT is also referred to as lateral gene transfer. It was first reported in 1928 by Frederick Griffith, who demonstrated the uptake of foreign DNA through transformation in bacteria (Griffith, 1928). HGT has been shown to take place in both prokaryotes and eukaryotes although its mechanisms are more extensively studied for prokaryotes (Huang, 2013; Zhaxybayeva and Doolittle, 2011). Popa and Dagan (2011) reported that almost 75% of total genes in a microbial genome are the results of HGT events. In microbes, HGT may occur via three classical mechanisms: (i) Transformation (the uptake of nucleic acids), (ii) transduction (virus-mediated gene transfer) or (iii) conjugation (plasmid-mediated gene transfer).

Detection of HGT requires the separation of vertical and horizontal arrangements in a genome of interest. Vertical transfer involves a time-consuming transfer of genetic material from parents to progenies. On the other hand, horizontal transfer usually happens between individuals that are not in parent-progenies relationships; thus, it influences the genome diversity and evolution of a species. A process named amelioration usually occurs that ensures the horizontally acquired gene is beneficial to the new host. Thus, the acquired regions will be less apparent from their original hosts and amended according to new host genome (Lawrence and Ochman, 1997). Normally the acquired gene will ameliorate due to spontaneous mutations, ecological adaptation and interactions with environment (Gyles and Boerlin, 2014). This process makes the detection between closest relatives laborious, due to the similarities in genetic content and phylogeny (Adato *et al.*, 2015).

HGT often contributes to new functions and phenotypic variations such as adaptations to changing environments (Gyles and Boerlin, 2014; Yue *et al.*, 2012) or variations in pathogenicity and antibiotic resistance (Juhas, 2015; McElroy *et al.*, 2014). Cases of HGT in eukaryotes are reportedly associated with many modifying traits (Keeling and Palmer, 2008). Besides parasitic plants, HGT encompasses a huge range of autotrophic lineages (Keeling and Palmer, 2008; Sanchez-Puerta, 2014), bryophytes (Yue *et al.*, 2012), ferns (Davis *et al.*, 2005), basal angiosperms (Berghorsson *et al.*, 2004) and grasses (Christin *et al.*, 2012). It is hypothesized that the number of horizontally transferred genes correlates with increasing heterotrophic dependence (Yang *et al.*, 2016).

Three approaches are widely implemented in HGT detection based on best pair-wise alignment (BLAST) matches, sequence composition, or phylogenetic analysis (Zhaxybayeva, 2009). The composition of the sequence is one of the earliest ways to detect the HGT event and is done by assessing the bias of synonymous codons in the recipient genome (Lawrence and Ochman, 1997). Besides assessing the discrepancy of genome composition, analyzing phylogenetic relationships is also a commonly used practice to identify horizontally transferred genes; where gene phylogenies that diverge from the organism's own known lineage is an indication of the presence of HGT events (Keeling and Palmer, 2008).

In this article, we review and compare several approaches for HGT detection in bacteria and parasitic plant systems. Due to the diverse complexity of genome composition, determining a pipeline for HGT detection remains challenging.

Background

HGT Events

HGT is the transfer of genetic information from one organism to another where the transferred material can be between related organisms within the same kingdom or even between different kingdoms.

Intra-kingdom (prokaryote to prokaryote; eukaryote to eukaryote)

HGT is more common in prokaryotes due to its asexual reproduction and simpler cell organization. Other than gaining drug resistance, HGT rescues prokaryotes from irreversible deleterious mutations (Muller's ratchet) to maintain the genomic information (Takeuchi *et al.*, 2014). Mechanisms on how HGT occur in prokaryotes are clearer compared to in eukaryotes because the passing of genetic information in eukaryotes is mainly via vertical gene transfer. HGT events that occur between eukaryotes are more complex and can be of different combinations: between unicellular eukaryotes, between multicellular eukaryotes, between unicellular and multicellular eukaryotes, and within the eukaryote that is between the organellar genome and the nuclear genome (Blanchard and Lynch, 2000; Keeling and Palmer, 2008; Richardson and Palmer, 2007).

Inter-kingdom (prokaryote to eukaryote; eukaryote to prokaryote)

Inter-kingdom HGT is highly interesting because it occurs between distant organisms. The most common direction of inter-kingdom HGT is the transfer of genetic material from prokaryotes to eukaryotes. The fact that there are larger populations of

prokaryotes compared to eukaryotes in nature tend to make eukaryotes the gene recipients instead of the donor (Keeling and Palmer, 2008). Normally, eukaryotes gain prokaryotic genes mainly for adaptive evolution (Husnik and McCutcheon, 2017; Lacroix and Citovsky, 2016). Eukaryote-to-prokaryote transfer events are rare; however, the receipt of eukaryotic genes by prokaryotes has been reported as a means environmental adaptation (Sieber *et al.*, 2017) and for the general improvement of functions in prokaryotic systems (Koonin *et al.*, 2001).

HGT Detection

The surrogate and comparative methods are commonly used to study HGT (Ravenhall *et al.*, 2015) and is also applicable in eukaryotes (Jaron *et al.*, 2014). The surrogate or parametric method investigates the nucleotide base composition of the DNA sequence of interest (Jaron *et al.*, 2014). This method is extremely useful before the advent of next-generation sequencing technologies, and is able to depend only on the genome of interest to infer HGT events (Ravenhall *et al.*, 2015). In contrast, the comparative approach requires prior knowledge of a genome, which utilises phylogenetic identification for input genes or local alignment comparisons against a reference sequence database (Jaron *et al.*, 2014). All-against-all BLASTP searches are carried out to identify sequences with high similarities. Genes with BLAST hits to organisms distant from its close relatives and having relevance to the host are considered as horizontally derived ones. Over the years, many methods have been developed to detect HGT at the genome level. It is believed that by combining predictions from parametric and comparative methods, a more complete set of putative horizontally acquired genes can be retrieved (Ravenhall *et al.*, 2015).

Computational Analysis of HGT Genes

Traditionally, attention was given to a single genome of interest to study HGT events. As a result, only a handful of HGT events were identified due to the limitation of available data. With genome sequencing becoming more common and less cost prohibitive, the acquisition of genome sequences need no longer be limited to a single representative strain or individual as a species representative. This allows for the full complement of genes in a taxonomy to be represented by a pan-genome (Tettelin *et al.*, 2005).

However, in some cases where there is no genome sequence available, transcriptome data exploration is an alternative approach to discover the HGTs. This is applicable especially for larger organisms such as eukaryotes (plants and fungi) that are more costly to sequence the genomes compared to smaller genomes like bacteria and viruses. Here, we discuss the identification of HGT genes in bacterial genomes via pan-genome analysis (Fig. 1) and in plants via transcriptome profiling. A list of the computational tools that are discussed in this article is presented in Table 1.

Pan-Genome Informatics

Annotated genome sequences for organisms of interest are required to study HGT events via pan-genome analysis. Many approaches are taken to first annotate the genome sequences, and these may depend on several factors including computational capacities, genome size, type of organism and project strategic planning. For instance, The National Center for Biotechnology Information (NCBI) uses the Prokaryotic Genome Annotation Pipeline (PGAP) and the Eukaryotic Genome Annotation Pipeline (EGAP) for annotating prokaryote and eukaryote genomes, respectively. They are supported by the high performance computing cluster that enable the implementation of both pipelines containing many integrated analysis tools. On the other hand, other researchers with less capacity of computational resources show different implementation of genome annotation based on their customised approaches (Firdaus-Raih *et al.*, 2018; Noor *et al.*, 2014).

Although various genome annotation methods are in use, the mutual goal is still to derive as much information from the genome sequence as possible. The availability of annotated complete genomes enable the pan-genome analysis for an organism; this allows for the sets of core and variable genes from the organism's gene pool to be determined. The core genes refers to genes that are present in all organisms in the taxonomy and normally consist of housekeeping genes that are responsible for main phenotypic traits (Tettelin *et al.*, 2005). In contrast, the variable genes, also known as dispensable genes, are those that are present in some of the organisms (two or more strains) and typically involve functions such as virulence mechanisms, ecological adaptation and survival in different environments (Jeong *et al.*, 2017; Medini *et al.*, 2005).

Compiling the pan-genome can be done using several tools including EDGAR (Blom *et al.*, 2009), PGAT (Brittnacher *et al.*, 2011), PanGP (Zhao *et al.*, 2012) and Panseq (Xiao *et al.*, 2015) (Table 1). The resulting orthologous clusters can then be used to identify genes that are possibly involved in HGT events. On top of the catalogue of entire genes in the organism, more analysis can be performed to enrich the information, thus consolidation of these data will converge into a holistic understanding of the entire organism.

Atypical sequence signatures

Deviation of a gene's sequence composition signature, such as GC content, from the mean can be indicative of its origins from a foreign genome (Garcia-Vallvé *et al.*, 2000). Thus, the prediction of HGT gene candidates can be inferred with lower or higher GC content than the mean of the genome's GC content (Liu *et al.*, 2009). Many tools were developed to analyze the GC content of genes and genomes, such as EMBOSS geecee which calculates the percentage of GC (Rice *et al.*, 2000), GC-Profile which analyses

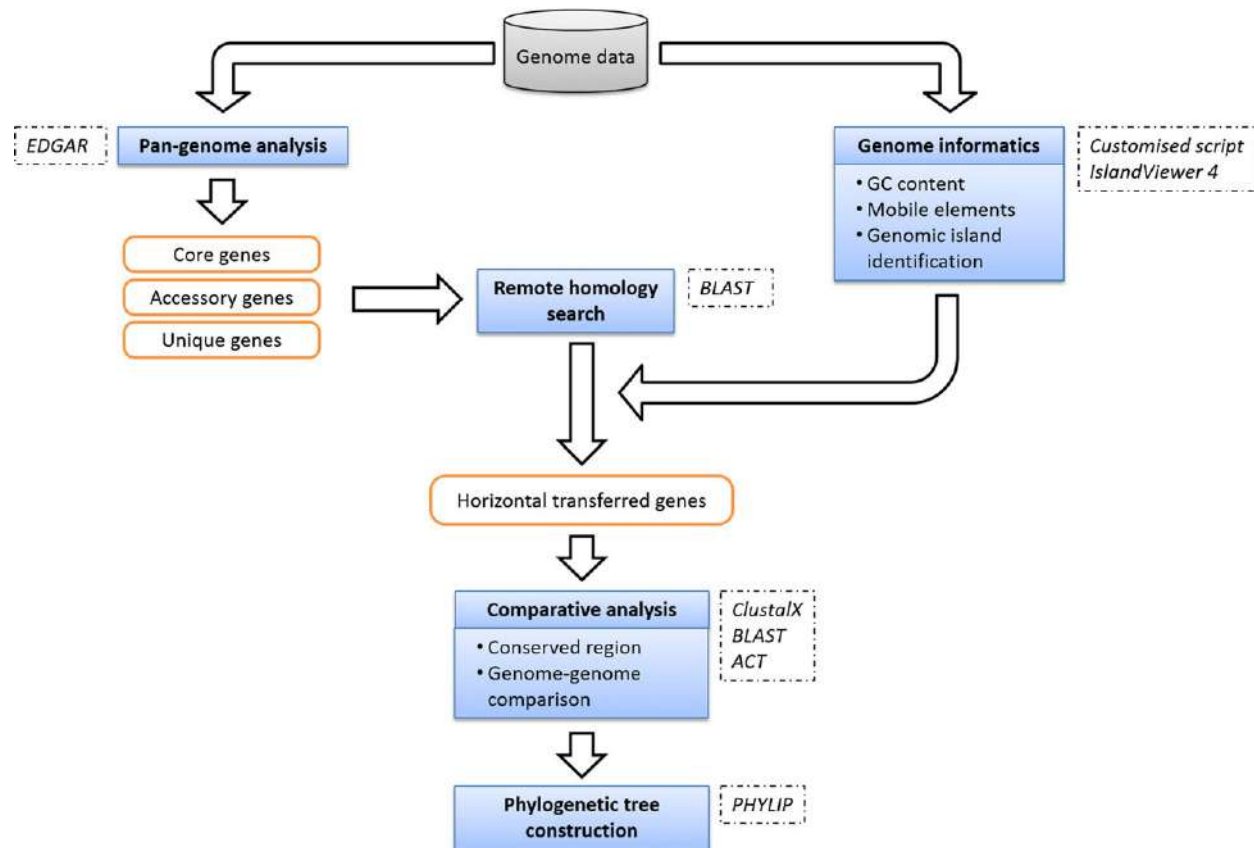


Fig. 1 Approaches in analysis of HGT using pan-genome data. The availability of genome data allows for a series of investigations via pan-genome analysis that can lead to the identification of horizontally transferred genes within the organism.

GC content by segmenting the genome (Gao and Zhang, 2006), and Artemis which allows for visualization of both the GC percentage and its deviation (Carver *et al.*, 2008).

Identification of genomic islands (GIs)

Genomic islands (GIs) refer to discrete DNA segments that establish horizontally transferred genes in a population (Jaron *et al.*, 2014). This region may integrate into the chromosome of the host; excised and transferred to a new host by transformation, conjugation or transduction. Proteins encoded within this region possibly present new functions in the recipient organism (Casacuberta and González, 2013). The prediction of GIs in prokaryotes can be better carried out than in eukaryotes due to the complexity of eukaryotic genomes. Many tools were developed to predict the occurrence of GIs in prokaryotic genomes such as GIPSy (Soares *et al.*, 2015), Zisland Explorer (Wei *et al.*, 2017) and IslandViewer 4 (Bertelli *et al.*, 2017). A good practice in predicting GIs is by consolidating the prediction outcomes from different tools by taking into account the different approaches that each tool implements to solve the problem.

In the most widely used software, IslandViewer 4, the prediction of GIs can be carried out using pre-computed genomes (complete bacterial and archaeal genomes from NCBI) or by uploading a genome sequence in EMBL or GENBANK format. This software incorporates four high precision prediction methods: IslandPick (Langille *et al.*, 2008), IslandPath-DIMOB (Hsiao *et al.*, 2003, 2005; Langille *et al.*, 2008), SIGI-HMM (Waack *et al.*, 2006), and Islander (Hudson *et al.*, 2015). The implementation of multiple prediction tools is important to ensure precise results and to reduce false positives in the predictions.

Predicting accurate GI boundaries is essential to obtain precise GI regions in the genome. This can be achieved by using IslandPick via comparative genomics approach if a reference genome is available (Langille *et al.*, 2008). Otherwise, Islander is recommended as it is capable of precisely discovering direct repeats that flank the acquired region (Hudson *et al.*, 2015). On the other hand, IslandPath-DIMOB and SIGI-HMM seem to have lower precision when predicting GI boundaries and tend to return fragmented horizontally transferred regions. Nevertheless, in the case of finding GIs via gene-based nucleotide bias approaches, both tools are favoured to generally anticipate more dispersed segments gained by HGT (Hsiao *et al.*, 2003, 2005; Waack *et al.*, 2006).

Identification of mobile elements

Mobile elements are DNA segments that can move within the genome or out to other organisms and while moving, it will bring along the donor genes to pass on to the recipient genome (Rankin *et al.*, 2011). Such an ability make these mobile elements a key

Table 1 List of tools applicable for the analysis of horizontal gene transfer events. Different computational tools for different platforms (web-based and operating systems) to analyze HGT genes

Category	Programs	Approach	URL	Platform	Reference
(a) Pan-genome analysis	EDGAR	A software framework for the comparative analysis of prokaryotic genomes.	http://edgar.computational.bio/	Web server	Blom <i>et al.</i> (2009)
	PGAT	A web-based database that compared multistrain microbial organisms' gene content and SNPs and impact of SNPs to encoded proteins.	http://nwrce.org/pgat	Web server	Brittnacher <i>et al.</i> (2011)
	PanGP	Integrates two sampling algorithms TR (totally random) and DG (distance guide) to perform pan-genome analysis in large scale bacterial genomes.	http://pangp.big.ac.cn/	Windows, Linux	Zhao <i>et al.</i> (2014)
	Panseq	Determine core and accessory region in genomes, SNPs within shared core regions and calculate loci in sets of accessory loci or core SNPs.	https://fz.corefacility.ca/panseq/	Web server, Windows, Linux	Xiao <i>et al.</i> (2015)
(b) Identification of atypical sequence signatures	geecee	A tool from the EMBOSS package to calculate fraction of GC per 100 bp within a gene or genome.	http://www.bioinformatics.nl/cgi-bin/emboss/geecee	Web server, Linux	Rice <i>et al.</i> (2000)
	GC-Profile	GC profile analysis by segmenting the genome.	http://tubic.tju.edu.cn/GC-Profile/		
	Web server Artemis	GC content (%) and its 2.5 SD cutoff from graph menu.	http://www.sanger.ac.uk/science/tools/artemis	Windows, Linux, Mac OS	Carver <i>et al.</i> (2008)
	IslandViewer	Integrate four GIs prediction tools: IslandPick, IslandPath-DIMOD, SIGI-HMM and Islander.	http://www.pathogenomics.sfu.ca/Webserver/islandviewer/	Web server	Bertelli <i>et al.</i> (2017)
(c) Identification of GIs	IslandPick ⁴	Identify GIs by using distance function, unique region only to query genome and conserved regions across genomes.	http://www.pathogenomics.sfu.ca/Webserver/islandviewer2/run_islandpick.php	Web server	Langille <i>et al.</i> (2008)
	IslandPath-DIMOB	Identify GIs by require dinucleotide bias and presence of mobility genes.	http://www.brinkman.mbb.sfu.ca/Linux/~mlangill/islandpath_dimob/	Linux	Hsiao <i>et al.</i> (2003, 2005), Langille <i>et al.</i> (2008)
	SIGI-HMM	It predicts GIs and putative donor of foreign genes based on codon usage analysis of gene or genome and HMM approach.	http://www.brinkman.mbb.sfu.ca/~mlangill/sigi-hmm/	Web server	Waack <i>et al.</i> (2006)
	Islander	A database that mapped GIs that were integrated into tDNAs, flanked by intact tDNAs and displaced tDNAs fragments, and contain integrase gene of tyrosine recombinase family.	http://bioinformatics.sandia.gov/islander/	Web server	Hudson <i>et al.</i> (2015)
(d) Identification of ME	Zisland explorer	Predict GIs based on segmental cumulative GC profile (from GC-profile tool).	http://cefg.uestc.edu.cn/Zisland_Explorer/	Web server, Windows, Linux, Mac OS	Wei <i>et al.</i> (2017)
	GIPro	Predict types of GIs: PAIs, MIs, RIs and Sis. It predicts by using genomic signatures deviation, tRNA genes, presence of MEs, and conserved regions.	https://www.bioinformatics.org/groups/?group_id=1180	Windows, Linux, Mac OS	Soares <i>et al.</i> (2015)
	ISQuest	Identify and annotate mobile genetic elements in raw read, partially assembled genome or fully assembled genome. It uses Blast to identify the MGEs.	https://sourceforge.net/projects/isquest/	Windows, Linux	Biswas <i>et al.</i> (2015)
	ISFinder PHASTER	A bacterial insertion sequences database. Identify and annotate phage sequences within bacterial genomes or plasmids by performing database comparison.	https://www-is.biotoul.fr/http://phaster.ca	Web server, Web server	Siguiet <i>et al.</i> (2006) Arndt <i>et al.</i> (2016)
(e) HGT detection	Alienness HGtector	Parse BLAST results against public libraries to identify candidate HGT in a genome of interest. Analyze BLAST hit distribution patterns based on sequence homology search hit distribution statistics.	http://alienness.sophia.inra.fr/https://github.com/DittmarLab/HGTector	Web server, Windows, Linux, Mac OS	Rancurel <i>et al.</i> (2017) Zhu <i>et al.</i> (2014)
	T-REX	All-against-all BLASTP search Reconstruct phylogenetic trees, reticulate networks and to infer the horizontal gene transfer events.	http://www.trex.uqam.ca/	Web server	Boc <i>et al.</i> (2012)

factor in evolution. Mobile elements include plasmid, bacteria-infecting viruses (bacteriophages), transposons, and insertion sequences. Examples of prediction tools for mobile elements are PHASTER for phages (Arndt *et al.*, 2016), ISFinder for identifying bacterial insertion sequences (Siguier *et al.*, 2006) and ISQuest for finding insertion sequences and transposons (Biswas *et al.*, 2015).

Identification of HGT Events Using Transcriptome Data

Since *de novo* whole genome assembly of a plant genome is complicated by the presence of repeat sequences, assessing HGT at the transcriptome level provides a direct approach to reveal the expression status of the genes. In host-parasite systems, a gene is considered horizontally acquired if it is phylogenetically placed nearer to its host rather than to its closest relatives. HGT detection using software is made possible for plants with the development of various tools, such as T-rex (Boc *et al.*, 2012), Alieness (Rancurel *et al.*, 2017) and HGTector (Zhu *et al.*, 2014). Nevertheless, some of these tools require specific files and preparation steps to conduct the HGT search. For instance, two newick trees generated for species and gene of interest are required as the input files. As with Alieness, a BLAST file of a whole proteome of interest is required to blast against any NCBI protein library. This tool calculates an Alien Index (AI) for early query protein and an AI > 0 indicates a possible HGT. HGTector takes one or more protein sets as input. In principle, this tool formulates a grouping scenario, calculates the three weights of each gene, defines certain cutoffs and then identifies the genes that exhibit atypical distribution.

In a case study that involves a parasitic plant (Section HGT in Plants), two web servers, OrthoVenn (see “Relevant Websites section”) (Wang *et al.*, 2015) and eggNOG-mapper of the EggNOG v 4.5 (see “Relevant Websites section”) (Huerta-Cepas *et al.*, 2016), were used to cluster orthologous genes across *Rafflesia* (parasite) and *Tetrastigma* (host) predicted proteomes. Homologous sequences were retrieved for each candidate HGT genes and were aligned using the multiple sequence alignment tool MUSCLE (Edgar, 2004) embedded in MEGA7 (Kumar *et al.*, 2016), which was later used to perform phylogenetic analysis through maximum likelihood searching.

Illustrative Case Studies

HGT in Bacteria

Here, we describe some cases of genes that were possibly transferred by HGT in a Gram-negative bacteria, *Burkholderia pseudomallei*, that in addition to being a versatile pathogen is notable for its highly plastic genome (Tumapa *et al.*, 2008). This flexibility is thought to be partly due to the HGT effect which turns this pathogenic soil bacteria into a highly adaptable organism that can live in a wide range of hosts (Currie, 2003) and has even been shown to actively respond and survive in very low nutrient environments (Mohd-Padil *et al.*, 2017). The pan-genome data of 48 complete *B. pseudomallei* genomes resulted in a total of 10,698 orthologous groups (OG) that encompassed 39% (4220 OGs) core genes and 61% (6478 OGs) accessory genes. Two observations of HGT are highlighted here; conserved hypothetical genes of *B. pseudomallei* and unique genes that were found in a specific *B. pseudomallei* strain.

Core hypothetical proteins in *B. pseudomallei*

In *B. pseudomallei* pan-genome data, 171 core genes encoding for hypothetical proteins (HPs) that were conserved in 48 complete *B. pseudomallei* genomes obtained from NCBI (Table 2). Further remote homology searching using BLAST (Altschul *et al.*, 1990) of these core HPs against a non redundant database revealed a range of conservation level in the sequences, from phylum down to species (Fig. 2).

Case 1: Hypothetical genes conserved in Burkholderiaceae family and two non-proteobacteria species

Up to December 2017, four hypothetical proteins – BPSL3171, BPSL3235, BPSS0337 and BPSS1590 (using the nomenclature for K96243) – are conserved in the Burkholderiaceae family and remote organisms from non proteobacteria phylum namely *Mumia flava* and *Streptomyces pluripotens*. Both bacteria were isolated from mangrove soil in Kuantan and Tanjung Lumpur of Pahang state, Malaysia, respectively (Lee *et al.*, 2014a,b). They are members of actinobacteria phylum that are highly adapted to various ecological environments and might be inhabitants of soil or aquatic environments, plant symbionts, plant pathogens or gastrointestinal commensal (Barka *et al.*, 2016). Mangrove soil is a highly dynamic ecosystem that is believed to force the soil bacterial populations to undergo rapid niche-specific adaptation. One mechanism for such adaptation is by acquiring beneficial genes from other bacteria in the same niche. Limited information is known about the *M. flava* genome as it is a novel actinobacterial strain (Lee *et al.*, 2014a), while *S. pluripotens* showed the presence of a wide-range of bacteriocins against pathogens such as *Staphylococcus aureus*, *Salmonella typhi* and *Aeromonas hydrophila* (Lee *et al.*, 2014b).

In terms of GC content, all four genes showed deviation from its chromosome GC average. Both BPSL3171 and BPSL3235 have lower GC content than K96243's large chromosome (67.7%), denoted by 51.46% and 55.84%, respectively. On another note, in the smaller second chromosome with GC percentage of 68.5%, the GC content in BPSS0337 is lower (62.59%), while it is higher in BPSS1590 (75.32%). This deviation is an informative indicator of horizontal gene transfer events that might have occurred at those positions.

Table 2 List of 48 *B. pseudomallei* strains used in the pan-genome study. A list of complete *B. pseudomallei* genomes that were sequenced up to October 2017

Country	Strain	Assembly ID	Size (Mb)	GC%
Australia	668	GCA_000015905.1	7.04	68.27
Australia	MSHR146	GCA_000521645.1	7.313	68.02
Australia	MSHR1655	GCA_000756165.1	7.028	67.99
Australia	MSHR2543	GCA_000959225.1	7.447	67.9
Australia	MSHR305	GCA_000439695.1	7.428	67.89
Australia	MSHR491	GCA_000959205.1	7.356	67.98
Australia	MSHR511	GCA_000520895.1	7.316	68.02
Australia	MSHR520	GCA_000583835.1	7.451	67.89
Australia	MSHR5848	GCA_000755965.1	7.29	68.08
Australia	MSHR5855	GCA_000756065.1	7.298	68.02
Australia	MSHR5858	GCA_000755945.1	7.072	68.28
Australia	MSHR62	GCA_000770395.1	7.225	68.17
Australia	MSHR668	GCA_000959305.1	7.043	68.27
Australia	MSHR840	GCA_000959185.1	7.13	68.09
Australia	NAU20B	GCA_000511915.1	7.314	68.02
Australia	NCTC_13179	GCA_000494855.1	7.337	67.99
Australia	NAU35A-3	GCA_000764575.1	7.204	68.08
Australia	NCTC 13,178	GCA_000511895.1	7.391	67.94
Australia	TSV 48	GCA_000770495.1	7.338	68.02
China	BPC006	GCA_000294635.1	7.155	68.22
Ecuador	7894	GCA_000959265.1	7.382	67.92
Malaysia	982	GCA_001277975.1	7.185	68.21
Malaysia	D286	In-house	7.102	68.25
Malaysia	H10	In-house	7.287	68.15
Malaysia	M1	GCA_001695715.1	7.318	67.98
Malaysia	MS	GCA_001695735.1	7.314	67.98
Malaysia	PMC2000	In-house	7.187	68.2
Malaysia	R15	In-house	7.179	68.25
Papua New Guinea	A79A	GCA_000770535.1	7.272	68.04
Papua New Guinea	B03	GCA_000770455.1	7.265	68.04
Papua New Guinea	K42	GCA_000770515.1	7.297	67.98
Taiwan	vgh07	GCA_000954175.1	7.046	68.2
Taiwan	vgh16R	GCA_001277875.1	7.267	68.08
Taiwan	vgh16W	GCA_001277895.1	7.267	68.08
Thailand	576	GCA_000756185.1	7.267	68.03
Thailand	3921	GCA_000953095.1	7.162	68.15
Thailand	1026b	GCA_000260515.1	7.231	68.16
Thailand	1106a	GCA_000756145.1	7.086	68.26
Thailand	1710b	GCA_000012785.1	7.308	67.99
Thailand	406e	GCA_000959145.1	7.272	68.07
Thailand	HBPUB10134a	GCA_000755925.1	7.218	68.12
Thailand	HBPUB10303a	GCA_000755905.1	7.178	68.18
Thailand	K96243	GCA_000011545.1	7.25	68.05
Thailand	Mahidol-1106a	GCA_000756125.1	7.085	68.26
Thailand	PHLS 112	GCA_000757015.2	7.202	68.18
USA	Bp1651	GCA_001318245.1	7.26	68.1
USA	PB08298010	GCA_000959345.1	7.376	67.95
Vietnam	Pasteur 52,237	GCA_000757035.2	7.325	68

Two domains of unknown function (DUFs), DUF1835 and DUF 3658, that have orthologs in both *M. flava* and *S. pluripotens* could also be found in the sequence encoding for BPSL3235. The latter domain can be found in bacteria and retains two highly conserved residues (Aspartic Acid and Arginine) that may play important functional roles. The gene is located in the same operon with that containing *bpsl3236*, a tetracycline (TetR) family transcriptional regulator that controls the level of susceptibility towards tetracycline antibiotics, a commonly used antibiotics against bacteria (Aleksandrov *et al.*, 2008; Grkovic *et al.*, 2002). There are also inverted repeats (palindrome) motifs found within 500 bp upstream and downstream of the *bpsl3235* gene together with additional motif of Rho-independent transcription terminator TERM 1195 located on the upstream region. According to Ooi *et al.* (2013), *bpsl3235* was found to be expressed in several conditions including during antibiotic treatment, osmotic stress, temperature stress, UV irradiation, oxidative stress and aerobic state.

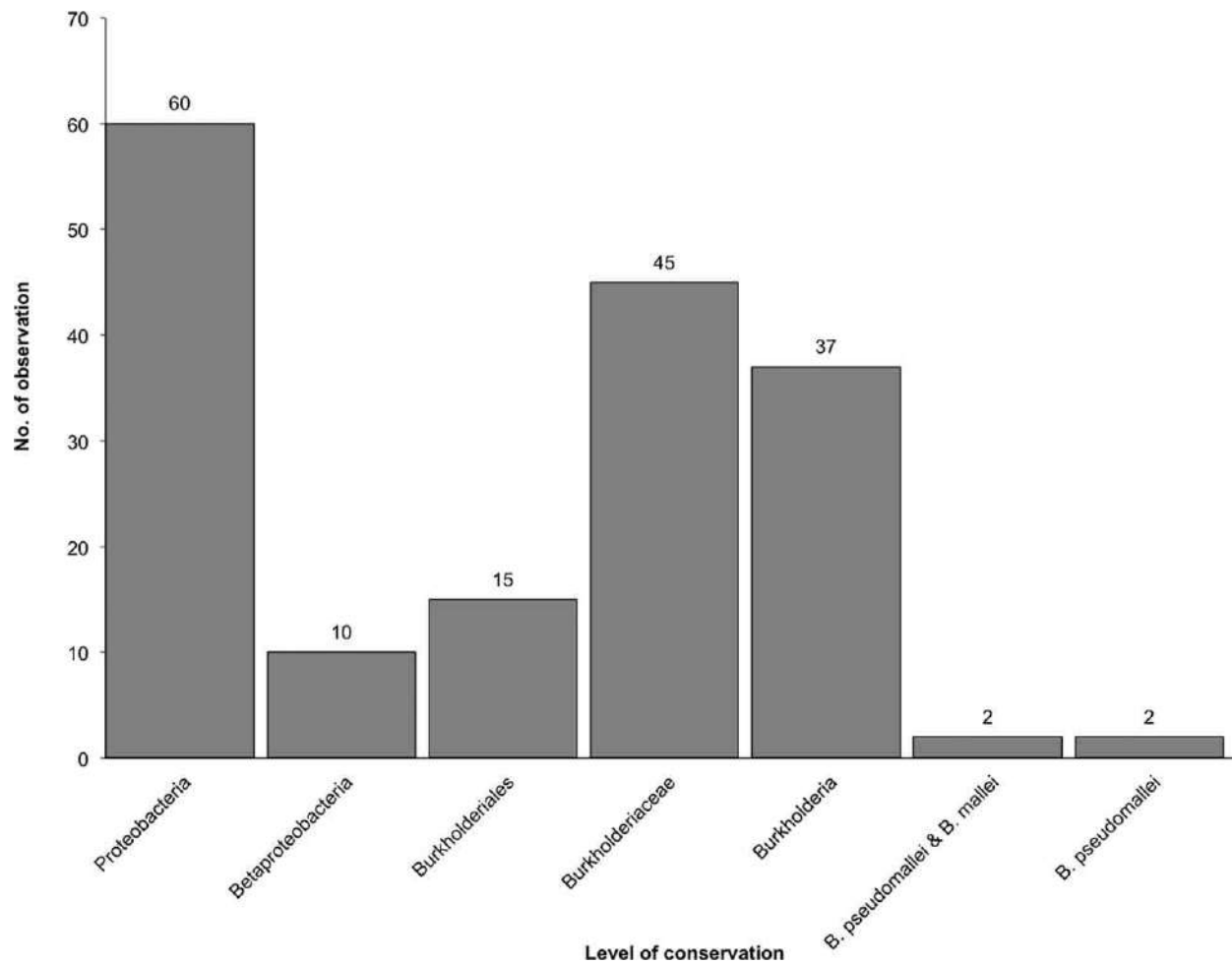


Fig. 2 Number of cases for different conservation levels in *B. pseudomallei* core hypothetical proteins. Various distribution of conservation from phylum to species can be seen in core HPs of *B. pseudomallei*.

The other three hypothetical proteins, BPSL3171, BPSS0337 and BPSS1590 have similarities with *M. flava* only. Inverted repeats (palindrome) motifs are found at 500 bp downstream of each gene while the *bpsl3171* gene is located in the operon together with sequences encoding two other hypothetical proteins (BPSL3172 and BPSL3173). Both *bpsl3171* and *bpss1590* are flanked respectively by secretion system protein genes and type III secretion system gene together with type IV pilus protein. Whereas *bpss0337* is flanked by an insertion element gene and an *araC* family transcriptional regulator. Interestingly, *bpsl3171* is only expressed during anaerobic condition and *in vivo* infection (Ooi *et al.*, 2013) thus indicating that the gene may have important roles for *B. pseudomallei* survival and pathogenicity. In contrast, expression of BPSS0337 and BPSS1590 were detected in most conditions except *in vivo* infection conditions.

However, none of the genes discussed are predicted to be part of genomic islands. Since not all horizontally transferred genes must be located in a genomic island, this observation alludes to the possibility that these genes may have been transferred and have undergone amelioration, hence providing a possible explanation as to why some tools can no longer distinguish these regions as being foreign. In context, it can be assumed that these genes may have been horizontally transferred between *Burkholderia* and two non proteobacteria due to their presence in the same niche although the origin species of these genes remains unknown or have yet to be isolated and sequenced.

Unique genes in *B. pseudomallei*

In this discussion, a unique gene or singleton refers to a strain-specific gene that is found in the *B. pseudomallei* pan-genome (Tettelin *et al.*, 2005). In total, 13% (1352 OGs) of the accessory genes in *B. pseudomallei* are categorized as singletons. Such genes could also be of uncharacterized function, however due consideration should be given when annotating such singletons in a pan-genome dataset because homologs may actually exist in strains that do not yet have available genome sequences. It is recommended that these annotations are cross-checked against the vast non-redundant database in GenBank and also periodically revisited as the number of genomes in the pan-genome dataset increases.

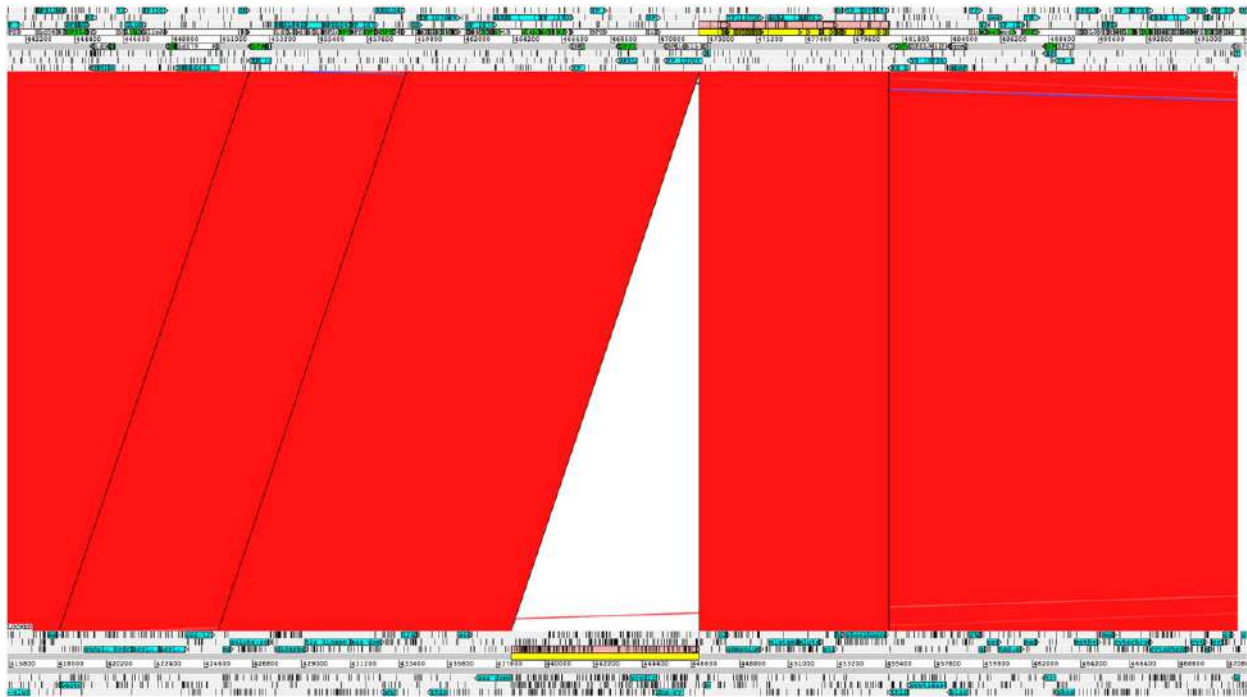


Fig. 3 An ACT view showing an insertion of a region (8514 bp) in the first *B. pseudomallei* chromosome that absent in reference genome of K96243 strain. The highlighted region in yellow consists of D286.1_0401 and D286.1_0402 genes that may have arisen due to HGT events in the D286 strain. The region is not observed in the reference genome (upper section).

Case 1: Unique genes in *B. pseudomallei* that are also found in other soil bacteria

In *B. pseudomallei* strain D286 (Lee *et al.*, 2007), two adjacent unique genes encoding 393 residues (D286.1_0401) of a sequence with no other available homologs and 408 residues (D286.1_0402) of DNA-cytosine methyltransferase, are strain-specific. Extension of the genomic regions containing both genes suggests that a sequence with 8514 bp in the first chromosome of *B. pseudomallei* is observed in D286 strain while it is absent in reference genome of strain K96243 (Fig. 3). The low GC percentage of this region (53.01%), a relatively large deviation from the mean GC of its genome (67.87%) disturbs the genomic stability. The deviation gives an insight into gene acquisition of the strain although D286.1_0401 is predicted to be part of a genomic island albeit with low confidence and the extended discussed region has no predicted mobile elements.

The homology search using D286.1_0401 as a query yielded detectable similarities to five distant organisms from different phylums including proteobacteria (*Novosphingobium barchaimii* and *Pelobacter seleniigenes*) (Nasaringarao and Häggblom, 2007; Niharika *et al.*, 2013), acidobacteria (*Granulicella pectinivorans* and *Acidobacteria bacterium*) (Pankratov and Dedysh, 2010) and Terrabacteria group (*Brevibacterium* sp. 239c). Interestingly, all of them share the mutual characteristics of being physiologically diverse and capable of living in a wide range of environments, especially soil, although this is common of many bacteria. On the other hand, its adjacent gene of D286.1_0402, has a vast similarity to many other organisms including *G. pectinivorans*, *A. bacterium* and *Eubacterium plexicaudatum* (Pankratov and Dedysh, 2010; Wilkins *et al.*, 1974).

Case 2: Unique genes in *B. pseudomallei* detected as part of a genomic island

Based on available pan-genome data, four unique genes in the D286 strain were identified to reside within a 53,131 bp length GI region predicted by IslandViewer 4. The four genes are annotated as hypothetical gene (D286.1_3048), HNH endonuclease family protein, (D286.1_3063), ATPase (D286.1_3064) and hypothetical protein encoding gene, (D286.1_3071). The HNH endonuclease family protein, (D286.1_3063) and ATPase (D286.1_3064) share similarity with diverse remote genomes but no similarities to any from *Burkholderia* sp. while D286.1_3048 is similar but also appear to have orthologs in three *Burkholderia* sp.. The sequence encoded by D286.1_3071 shows similarity with only one hypothetical protein from *B. vietnamiensis*. Compared to the mean GC of its genome (67.87%), this region carries a lower GC percentage of 59.66%, indicating that foreign gene influx may have occurred at that position. This is supported by the finding of the presence of tRNA genes that is well known to be associated to GI that were obtained from foreign organisms (Tuller, 2011).

Around 20% of genes in the GI encode for hypothetical proteins. Our results show that this island also consists of a phage-related integrase gene next to the tRNA gene, proposing that this GI may possibly have been generated by a phage or plasmid with integrative functions (Odenbreit and Haas, 2002; Semsey *et al.*, 2002). In addition to the phage integrase family protein gene, another group of genes that were also present were those for components of the Type VI secretion system (T6SS) that is well known as a virulence factor for pathogenic bacteria and as a mechanism for protein transport across the cell envelope in proteobacteria

(Pukatzki *et al.*, 2006). Furthermore, there are also phage-related genes present, such as those encoding phage virion morphogenesis protein, phage tail protein and phage capsid protein. A part of the GI seems to be similar with other *B. pseudomallei* strains, however part of the region is missing in the reference genome of the K96243 strain. Thus, from the pan-genome data, we can obtain some indicators that these unique genes might be an effect of a HGT event.

HGT in Plants

Lateral gene transfer in parasitic flowering plants is known to occur via genomic DNA (Xi *et al.*, 2012), or via mRNA intermediates (Kim *et al.*, 2014). Therefore, HGT involving plants is studied using both genome (Yue *et al.*, 2012) and transcriptome (Kim *et al.*, 2014; Xi *et al.*, 2012; Zhang *et al.*, 2014, 2013) data whenever possible. Nuclear HGTs are less common among multicellular eukaryotes such as animals, fungi and plants (Richardson and Palmer, 2007) compared to organellar genes. However, a total of 47 horizontally acquired nuclear genes were identified from *R. cantleyi* using a combination of BLAST and phylogenomic approaches (Xi *et al.*, 2012). Using the same approach, an approximation of 24%–41% of the mitochondrial gene sequences were identified to be horizontally acquired from their host (Xi *et al.*, 2013).

Clustering of orthologous groups across two or more distant species results in a list of horizontally acquired candidate genes. Through reciprocal proteome clustering between *Rafflesia* and *Tetrastigma*, 59 orthologous clusters and 67 candidate orthologs were identified from OrthoVenn and eggNOG-mapper respectively to be *Tetrastigma*-related and these genes were subjected to phylogenetic analysis. Thirteen genes were identified to be transferred through HGT. Interestingly, only rhamnose synthase was similarly identified by the two approaches used in this case study and the analysis conducted by Xi *et al.* (2012). The remaining 12 genes were found to span across cell wall formation, degradation, defense mechanism, stress response, amino acid biosynthesis and glycolysis. Additionally, 24 *Rafflesia* genes were not grouped together, either with its closest relatives nor with *Tetrastigma*. Of these genes, six were reported to be horizontally transferred by Xi *et al.* (2012). The discrepancy could be attributed to the polyphyletic status or unresolved ambiguity exerted by the evolving elements.

Discussion and Future Directions

Several methods have been developed to improve the accuracy in addition to reducing the computational time to search for HGTs. Compared to the different approaches employed by Xi *et al.* (2012), the methods described here identifies much lower numbers of HGTs, possibly due to stringent parameters applied for orthologous clustering during the early search. This may rule out the set of genes that have lost the signatures of their donor as a result of evolution. In contrast, the classification of homologous sequences into the respective species presents a higher number of candidates for HGT identification in an organism of interest.

Choosing an appropriate method for HGT detection is not straightforward. It depends on the nature of the study, species of interest, computational resources and time. With the development of bioinformatic tools and pipelines, HGT detection can now be performed on large-scale data sets, covering both prokaryotes and eukaryotes. Care should be taken in order to avoid false positives that could lead to a pool of unresolved artifacts. For instance, despite the application of BLAST-based detection in some studies (Yoshida *et al.*, 2010; Zhang *et al.*, 2014, 2017), comprehensive follow up work is required to eliminate high false positives, such as misidentification of contaminant sequences, deficient taxon sampling, intriguing gene birth and death processes, improper rooting and frame-shift errors that must also be accounted for (Yang *et al.*, 2016). Similarly, incomplete lineage sorting during phylogeny construction can lead to inaccurate HGT identification (Than *et al.*, 2007).

To date, there is no true one-stop solution for the identification of genes that were potentially horizontally transferred. However, the results of tools used for sequence analysis and phylogenetics can be used and the relevant context provided to allow for inference of HGT events. As more genomes become available, it would perhaps be timely for a fit for purpose tool or pipeline to be developed for inferring HGT events despite such tools requiring significant capacity and resource allocations.

Acknowledgement

MFR is funded by the Universiti Kebangsaan Malaysia grant DIP-2017-013.

See also: Genetics and Population Analysis. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Sequence Analysis

References

- Adato, O., Ninyo, N., Gophna, U., Snir, S., 2015. Detecting horizontal gene transfer between closely related taxa. *PLOS Comput. Biol.* 11, e1004408.
- Aleksandrov, A., Schuld, L., Hinrichs, W., Simonson, T., 2008. Tet repressor induction by tetracycline: A molecular dynamics, continuum electrostatics, and crystallographic study. *J. Mol. Biol.* 378, 896–910. Available at: <https://doi.org/10.1016/j.jmb.2008.03.022>.

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410. Available at: [https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2).
- Arndt, D., Grant, J.R., Marcu, A., *et al.*, 2016. PHASTER: A better, faster version of the PHAST phage search tool. *Nucleic Acids Res.* 44 (W1), W16–W21. Available at: <https://doi.org/10.1093/nar/gkw387>.
- Barka, E.A., Vatsa, P., Sanchez, L., *et al.*, 2016. Taxonomy, physiology, and natural products of actinobacteria. *Microbiol. Mol. Biol. Rev.* 80, 1–43. Available at: <https://doi.org/10.1128/MMBR.00019-15>.
- Bergthorsson, U., Richardson, A.O., Young, G.J., Goertzen, L.R., Palmer, J.D., 2004. Massive horizontal transfer of mitochondrial genes from diverse land plant donors to the basal angiosperm *Amborella*. *Proc. Natl. Acad. Sci. USA* 101, 17747 LP–17717752.
- Bertelli, C., Laird, M.R., Williams, K.P., *et al.*, 2017. IslandViewer 4: Expanded prediction of genomic islands for larger-scale datasets. *Nucleic Acids Res.* 45, W30–W35.
- Biswas, A., Gauthier, D.T., Ranjan, D., Zubair, M., 2015. ISQuest: Finding insertion sequences in prokaryotic sequence fragment data. *Bioinformatics* 31, 3406–3412. Available at: <https://doi.org/10.1093/bioinformatics/btv388>.
- Blanchard, J.L., Lynch, M., 2000. Organellar genes: Why do they end up in the nucleus? *Trends Genet.* 16, 315–320.
- Blom, J., Albaum, S.P., Doppmeier, D., *et al.*, 2009. EDGAR: A software framework for the comparative analysis of prokaryotic genomes. *BMC Bioinform.* 10, 154. Available at: <https://doi.org/10.1186/1471-2105-10-154>.
- Boc, A., Diallo, A.B., Makarenkov, V., 2012. T-REX: A web server for inferring, validating and visualizing phylogenetic trees and networks. *Nucleic Acids Res.* 40, W573–W579. Available at: <https://doi.org/10.1093/nar/gks485>.
- Brittnacher, M.J., Fong, C., Hayden, H.S., *et al.*, 2011. PGAT: A multistain analysis resource for microbial genomes. *Bioinformatics* 27, 2429–2430. Available at: <https://doi.org/10.1093/bioinformatics/btr418>.
- Carver, T., Berriman, M., Tivey, A., *et al.*, 2008. Artemis and ACT: Viewing, annotating and comparing sequences stored in a relational database. *Bioinformatics* 24, 2672–2676. Available at: <https://doi.org/10.1093/bioinformatics/btn529>.
- Casacuberta, E., González, J., 2013. The impact of transposable elements in environmental adaptation. *Mol. Ecol.* 22, 1503–1517. Available at: <https://doi.org/10.1111/mec.12170>.
- Christin, P.-A., Wallace, M.J., Clayton, H., *et al.*, 2012. Multiple photosynthetic transitions, polyploidy, and lateral gene transfer in the grass subtribe Neurachninae. *J. Exp. Bot.* 63, 6297–6308. Available at: <https://doi.org/10.1093/jxb/ers282>.
- Crisp, A., Boschetti, C., Perry, M., Tunnaciffe, A., Micklem, G., 2015. Expression of multiple horizontally acquired genes is a hallmark of both vertebrate and invertebrate genomes. *Genome Biol.* 16, 50. Available at: <https://doi.org/10.1186/s13059-015-0607-3>.
- Currie, B.J., 2003. Melioidosis: An important cause of pneumonia in residents of and travellers returned from endemic regions. *Eur. Respir. J.* 22, 542 LP–542550.
- Davis, C.C., Anderson, W.R., Wurdack, K.J., 2005. Gene transfer from a parasitic flowering plant to a fern. *Proc. R. Soc. B Biol. Sci.* 272, 2237–2242. Available at: <https://doi.org/10.1098/rspb.2005.3226>.
- Edgar, R.C., 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* 32, 1792–1797. Available at: <https://doi.org/10.1093/nar/gkh340>.
- Firdaus-Raih, M., Hashim, N.H.F., Bharudin, I., *et al.*, 2018. The *Glaciozyma antarctica* genome reveals an array of systems that provide sustained responses towards temperature variations in a persistently cold habitat. *PLOS ONE* 13, e0189947.
- Gao, F., Zhang, C.-T., 2006. GC-Profile: A web-based tool for visualizing and analyzing the variation of GC content in genomic sequences. *Nucleic Acids Res.* 34 (Web Server Issue), W686–W691. Available at: <https://doi.org/10.1093/nar/gkl040>.
- García-Vallvé, S., Romeu, A., Palau, J., 2000. Horizontal gene transfer in bacterial and archaeal complete genomes. *Genome Res.* 10 (11), 1719–1725. Available at: <https://doi.org/10.1101/gr.130000>.
- Griffith, F., 1928. The significance of pneumococcal types. *J. Hyg. (Lond.)* 27, 113–159.
- Grkovic, S., Brown, M.H., Skurray, R.A., 2002. Regulation of bacterial drug export systems. *Microbiol. Mol. Biol. Rev.* 66, 671–701. Available at: <https://doi.org/10.1128/MMBR.66.4.671>.
- Gyles, C., Boerlin, P., 2014. Horizontally transferred genetic elements and their role in pathogenesis of bacterial disease. *Vet. Pathol.* 51, 328–340. Available at: <https://doi.org/10.1177/030098513511131>.
- Hsiao, W.W.L., Ung, K., Aeschliman, D., *et al.*, 2005. Evidence of a large novel gene pool associated with prokaryotic genomic islands. *PLOS Genet.* 1, e62. Available at: <https://doi.org/10.1371/journal.pgen.0010062>.
- Hsiao, W., Wan, I., Jones, S.J., Brinkman, F.S.L., 2003. IslandPath: Aiding detection of genomic islands in prokaryotes. *Bioinformatics* 19, 418–420.
- Huang, J., 2013. Horizontal gene transfer in eukaryotes: The weak-link model. *Bioessays* 35, 868–875. Available at: <https://doi.org/10.1002/bies.201300007>.
- Hudson, C.M., Lau, B.Y., Williams, K.P., 2015. Islander: A database of precisely mapped genomic islands in tRNA and tmRNA genes. *Nucleic Acids Res.* 43, D48–D53.
- Huerta-Cepas, J., Szklarczyk, D., Forslund, K., *et al.*, 2016. eggNOG 4.5: A hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic Acids Res.* 44, D286–D293. Available at: <https://doi.org/10.1093/nar/gkv1248>.
- Husnik, F., McCutcheon, J.P., 2017. Functional horizontal gene transfer from bacteria to eukaryotes. *Nat. Rev. Microbiol.* 16, 67.
- Jaron, K.S., Moravec, J.C., Martinková, N., 2014. SigHunt: Horizontal gene transfer finder optimized for eukaryotic genomes. *Bioinformatics* 30, 1081–1086.
- Jeong, D.-W., Heo, S., Ryu, S., Blom, J., Lee, J.-H., 2017. Genomic insights into the virulence and salt tolerance of *Staphylococcus equorum*. *Sci. Rep.* 7, 5383. Available at: <https://doi.org/10.1038/s41598-017-05918-5>.
- Juhas, M., 2015. Type IV secretion systems and genomic islands-mediated horizontal gene transfer in *Pseudomonas* and *Haemophilus*. *Microbiol. Res.* 170, 10–17. Available at: <https://doi.org/10.1016/j.micres.2014.06.007>.
- Keeling, P.J., Palmer, J.D., 2008. Horizontal gene transfer in eukaryotic evolution. *Nat. Rev. Genet.* 9, 605–618. Available at: <https://doi.org/10.1038/nrg2386>.
- Kim, G., LeBlanc, M.L., Wafula, E.K., dePamphilis, C.W., Westwood, J.H., 2014. Plant science. Genomic-scale exchange of mRNA between a parasitic plant and its hosts. *Science* 345, 808–811. Available at: <https://doi.org/10.1126/science.1253122>.
- Koonin, Eugene V., Makarova, Kira S., Aravind, L., 2001. Horizontal gene transfer in prokaryotes: Quantification and classification. *Annu. Rev. Microbiol.* 55, 709–742. Available at: <https://doi.org/10.1146/annurev.micro.55.1.709>.
- Kumar, S., Stecher, G., Tamura, K., 2016. MEGA7: Molecular Evolutionary Genetics Analysis version 7.0 for bigger datasets. *Mol. Biol. Evol.* 33, 1870–1874. Available at: <https://doi.org/10.1093/molbev/msw054>.
- Lacroix, B., Citovsky, V., 2016. Transfer of DNA from bacteria to eukaryotes. *MBio* 7 (4), Available at: <https://doi.org/10.1128/mBio.00863-16>.
- Langille, M.G.I., Hsiao, W.W.L., Brinkman, F.S.L., 2008. Evaluation of genomic island predictors using a comparative genomics approach. *BMC Bioinform.* 9, 329. Available at: <https://doi.org/10.1186/1471-2105-9-329>.
- Lawrence, J.G., Ochman, H., 1997. Amelioration of bacterial genomes: Rates of change and exchange. *J. Mol. Evol.* 44. Available at: <https://doi.org/10.1007/PL00006158>.
- Lee, S.-H., Chong, C.-E., Lim, B.-S., *et al.*, 2007. *Burkholderia pseudomallei* animal and human isolates from Malaysia exhibit different phenotypic characteristics. *Diagn. Microbiol. Infect. Dis.* 58, 263–270. Available at: <https://doi.org/10.1016/j.diagmicrobio.2007.01.002>.
- Lee, L.H., Zainal, N., Azman, A.S., *et al.*, 2014a. *Mumia flava* gen. nov., sp. nov., an actinobacterium of the family Nocardioidaceae. *Int. J. Syst. Evol. Microbiol.* 64, 1461–1467. Available at: <https://doi.org/10.1099/ijs.0.058701-0>.
- Lee, L.H., Zainal, N., Azman, A.S., *et al.*, 2014b. *Streptomyces pluripotens* sp. nov., a bacteriocin-producing streptomycete that inhibits meticillin-resistant *Staphylococcus aureus*. *Int. J. Syst. Evol. Microbiol.* 64, 3297–3306. Available at: <https://doi.org/10.1099/ijs.0.065045-0>.

- Liu, M., Siezen, R.J., Nauta, A., 2009. *In silico* prediction of horizontal gene transfer events in *Lactobacillus bulgaricus* and *Streptococcus thermophilus* reveals proto-cooperation in yogurt manufacturing. *Appl. Environ. Microbiol.* 75, 4120–4129. Available at: <https://doi.org/10.1128/AEM.02898-08>.
- McElroy, K., Thomas, T., Luciani, F., 2014. Deep sequencing of evolving pathogen populations: Applications, errors, and bioinformatic solutions. *Microb. Inform. Exp.* 4, 1. Available at: <https://doi.org/10.1186/2042-5783-4-1>.
- Medini, D., Donati, C., Tettelin, H., Massignani, V., Rappuoli, R., 2005. The microbial pan-genome. *Curr. Opin. Genet. Dev.* 15, 589–594. Available at: <https://doi.org/10.1016/j.gde.2005.09.006>.
- Mohd-Padil, H., Damiri, N., Sulaiman, S., *et al.*, 2017. Identification of sRNA mediated responses to nutrient depletion in *Burkholderia pseudomallei*. *Sci. Rep.* 7, 17173. Available at: <https://doi.org/10.1038/s41598-017-17356-4>.
- Nasaringarao, P., Häggblom, M.M., 2007. *Pelobacter seleniigenes* sp. nov., a selenaterespiring bacterium. *Int. J. Syst. Evol. Microbiol.* 57, 1937–1942. Available at: <https://doi.org/10.1099/ijs.0.64980-0>.
- Niharika, N., Moskalikova, H., Kaur, J., *et al.*, 2013. *Novosphingobium barchaimii* sp. nov., isolated from hexachlorocyclohexane-contaminated soil. *Int. J. Syst. Evol. Microbiol.* 63, 667–672. Available at: <https://doi.org/10.1099/ijs.0.039826-0>.
- Noor, Y.M., Samsulrizal, N.H., Jema'on, N.A., *et al.*, 2014. A comparative genomic analysis of the alkalitolerant soil bacterium *Bacillus lehensis* G1. *Gene* 545, 253–261. Available at: <https://doi.org/10.1016/j.gene.2014.05.012>.
- Odenbreit, S., Haas, R., 2002. *Helicobacter pylori*: impact of gene transfer and the role of the *cag* pathogenicity island for host adaptation and virulence. *Current topics in microbiology and immunology* 264, 1–22.
- Ooi, W.F., Ong, C., Nandi, T., *et al.*, 2013. The condition-dependent transcriptional landscape of *Burkholderia pseudomallei*. *PLOS Genet.* 9. Available at: <https://doi.org/10.1371/journal.pgen.1003795>.
- Pankratov, T.A., Dedysh, S.N., 2010. *Granulicella paludicola* gen. nov., sp. nov., *Granulicella pectinivorans* sp. nov., *Granulicella aggregans* sp. nov. and *Granulicella rosea* sp. nov., acidophilic, polymer-degrading acidobacteria from Sphagnum peat bogs. *Int. J. Syst. Evol. Microbiol.* 60, 2951–2959. Available at: <https://doi.org/10.1099/ijs.0.021824-0>.
- Popa, O., Dagan, T., 2011. Trends and barriers to lateral gene transfer in prokaryotes. *Curr. Opin. Microbiol.* 14 (5), 615–623. Available at: <https://doi.org/10.1016/j.mib.2011.07.027>.
- Pukatzki, S., Ma, A.T., Sturtevant, D., *et al.*, 2006. Identification of a conserved bacterial protein secretion system in *Vibrio cholerae* using the *Dictyostelium* host model system. *Proc. Natl. Acad. Sci. USA* 103, 1528–1533. Available at: <https://doi.org/10.1073/pnas.0510322103>.
- Rancurel, C., Legrand, L., Danchin, E.G.J., 2017. Alienness: Rapid detection of candidate horizontal gene transfers across the tree of life. *Genes (Basel)* 8, E248.
- Rankin, D.J., Rocha, E.P.C., Brown, S.P., 2011. What traits are carried on mobile genetic elements, and why? *Heredity (Edinb)* 106 (1), 1–10. Available at: <https://doi.org/10.1038/hdy.2010.24>.
- Ravenhall, M., Skunca, N., Lassalle, F., Dessimoz, C., 2015. Inferring horizontal gene transfer. *PLOS Comput. Biol.* 11, e1004095. Available at: <https://doi.org/10.1371/journal.pcbi.1004095>.
- Rice, P., Longden, I., Bleasby, A., 2000. EMBOSS: The European Molecular Biology Open Software Suite. *Trends Genet.* 16, 276–277.
- Richardson, A.O., Palmer, J.D., 2007. Horizontal gene transfer in plants. *J. Exp. Bot.* 58, 1–9. Available at: <https://doi.org/10.1093/jxb/eri148>.
- Sanchez-Puerta, M.V., 2014. Involvement of plastid, mitochondrial and nuclear genomes in plant-to-plant horizontal gene transfer. *Acta Soc. Bot. Pol.* 83, 317–323. Available at: <https://doi.org/10.5586/asbp.2014.041>.
- Semsey, S., Blaha, B., Köles, K., Orosz, L., Papp, P.P., 2002. Site-specific integrative elements of rhizobiophage 16-3 can integrate into proline tRNA (CGG) genes in different bacterial genera. *J. Bacteriol.* 184, 177–182. Available at: <https://doi.org/10.1128/JB.184.1.177-182.2002>.
- Sieber, K.B., Bromley, R.E., Hotopp, J.C.D., 2017. Lateral gene transfer between prokaryotes and eukaryotes. *Exp. Cell Res.* 358, 421–426. Available at: <https://doi.org/10.1016/j.yexcr.2017.02.009>.
- Siguier, P., Perochon, J., Lestrade, L., Mahillon, J., Chandler, M., 2006. ISfinder: The reference centre for bacterial insertion sequences. *Nucleic Acids Res.* 34 (Database Issue), D32–D36. Available at: <https://doi.org/10.1093/nar/gkj014>.
- Soares, S.C., Geyik, H., Ramos, R.T.J., *et al.*, 2016. GIPSY: Genomic island prediction software. *J. Biotechnol.* 232, 2–11. Available at: <https://doi.org/10.1016/j.jbiotec.2015.09.008>.
- Takeuchi, N., Kaneko, K., Koonin, E.V., 2014. Horizontal gene transfer can rescue prokaryotes from Muller's Ratchet: Benefit of DNA from dead cells and population subdivision. *G3 Genes/Genomes/Genetics* 4 (2), 325–339. Available at: <https://doi.org/10.1534/g3.113.009845>.
- Tettelin, H., Massignani, V., Cieslewicz, M.J., *et al.*, 2005. Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: Implications for the microbial "pan-genome". *Proc. Natl. Acad. Sci. USA* 102, 13950–13955. Available at: <https://doi.org/10.1073/pnas.0506758102>.
- Than, C., Ruths, D., Innan, H., Nakhleh, L., 2007. Confounding factors in HGT detection: Statistical error, coalescent effects, and multiple solutions. *J. Comput. Biol.* 14, 517–535. Available at: <https://doi.org/10.1089/cmb.2007.A010>.
- Tuller, T., 2011. Codon bias, tRNA pools and horizontal gene transfer. *Mob. Genet. Elements* 1, 75–77. Available at: <https://doi.org/10.4161/mge.1.1.15400>.
- Tumapa, S., Holden, M.T.G., Vesaratchavest, M., *et al.*, 2008. *Burkholderia pseudomallei* genome plasticity associated with genomic island variation. *BMC Genom.* 9, 190. Available at: <https://doi.org/10.1186/1471-2164-9-190>.
- Waack, S., Keller, O., Asper, R., *et al.*, 2006. Score-based prediction of genomic islands in prokaryotic genomes using hidden Markov models. *BMC Bioinform.* 7, 142. Available at: <https://doi.org/10.1186/1471-2105-7-142>.
- Wang, Y., Coleman-Derr, D., Chen, G., Gu, Y.Q., 2015. OrthoVenn: A web server for genome wide comparison and annotation of orthologous clusters across multiple species. *Nucleic Acids Res.* 43, W78–W84. Available at: <https://doi.org/10.1093/nar/gkv487>.
- Wei, W., Gao, F., Du, M.-Z., *et al.*, 2017. Zisland Explorer: Detect genomic islands by combining homogeneity and heterogeneity properties. *Brief. Bioinform.* 18, 357–366. Available at: <https://doi.org/10.1093/bib/bbw019>.
- Wilkins, T.D., Fulghum, R.S., Wilkins, J.H., 1974. *Eubacterium plexicaudatum* sp. nov., an anaerobic bacterium with a subpolar tuft of flagella, isolated from a mouse cecum. *Int. J. Syst. Bacteriol.* 24, 408–411. Available at: <https://doi.org/10.1099/00207713-24-4-408>.
- Xi, Z., Bradley, R.K., Wurdack, K.J., *et al.*, 2012. Horizontal transfer of expressed genes in a parasitic flowering plant. *BMC Genom.* 13, 227. Available at: <https://doi.org/10.1186/1471-2164-13-227>.
- Xi, Z., Wang, Y., Bradley, R.K., *et al.*, 2013. Massive mitochondrial gene transfer in a parasitic flowering plant clade. *PLOS Genet.* 9, e1003265. Available at: <https://doi.org/10.1371/journal.pgen.1003265>.
- Xiao, J., Zhang, Z., Wu, J., Yu, J., 2015. A brief review of software tools for pangenomics. *Genom. Proteom. Bioinform.* 13, 73–76. Available at: <https://doi.org/https://doi.org/10.1016/j.gpb.2015.01.007>.
- Yang, Z., Zhang, Y., Wafula, E.K., *et al.*, 2016. Horizontal gene transfer is more frequent with increased heterotrophy and contributes to parasite adaptation. *Proc. Natl. Acad. Sci. USA* 113, E7010–E7019. Available at: <https://doi.org/10.1073/pnas.1608765113>.
- Yoshida, S., Maruyama, S., Nozaki, H., Shirasu, K., 2010. Horizontal gene transfer by the parasitic plant *Striga hermonthica*. *Science* 328, 1128 LP–1121128. (80-).
- Yue, J., Hu, X., Sun, H., Yang, Y., Huang, J., 2012. Widespread impact of horizontal gene transfer on plant colonization of land. *Nat. Commun.* 3, 1152.
- Zhang, Y., Fernandez-Aparicio, M., Wafula, E.K., *et al.*, 2013. Evolution of a horizontally acquired legume gene, albumin 1, in the parasitic plant *Phellipanche aegyptiaca* and related species. *BMC Evol. Biol.* 13, 48. Available at: <https://doi.org/10.1186/1471-2148-13-48>.
- Zhang, X., Liu, X., Liang, Y., *et al.*, 2017. Adaptive evolution of extreme acidophile *Sulfobacillus thermosulfidooxidans* potentially driven by horizontal gene transfer and gene loss. *Appl. Environ. Microbiol.* 83. Available at: <https://doi.org/10.1128/AEM.03098-16>.

- Zhang, D., Qi, J., Yue, J., *et al.*, 2014. Root parasitic plant *Orobancha aegyptiaca* and shoot parasitic plant *Cuscuta australis* obtained Brassicaceae-specific strictosidine synthase-like genes by horizontal gene transfer. BMC Plant Biol. 14, 19. Available at: <https://doi.org/10.1186/1471-2229-14-19>.
- Zhao, Y., Wu, J., Yang, J., *et al.*, 2012. PGAP: Pan-genomes analysis pipeline. Bioinformatics 28, 416–418. Available at: <https://doi.org/10.1093/bioinformatics/btr655>.
- Zhao, Y., Jia, X., Yang, J., *et al.*, 2014. PanGP: A tool for quickly analyzing bacterial pan-genome profile. Bioinformatics 30, 1297–1299. Available at: <https://doi.org/10.1093/bioinformatics/btu017>.
- Zhaxybayeva, O., 2009. Detection and quantitative assessment of horizontal gene transfer. In: Gogarten, M.B., Gogarten, J.P., Olendzenski, L.C. (Eds.), Horizontal Gene Transfer: Genomes in Flux. Totowa, NJ: Humana Press, pp. 195–213. Available at: https://doi.org/10.1007/978-1-60327-853-9_11.
- Zhaxybayeva, O., Doolittle, W.F., 2011. Lateral gene transfer. Curr. Biol. 21, R242–R246. Available at: <https://doi.org/https://doi.org/10.1016/j.cub.2011.01.045>.
- Zhu, Q., Kosoy, M., Dittmar, K., 2014. HGTector: An automated method facilitating genome-wide discovery of putative horizontal gene transfers. BMC Genom. 15, 717. Available at: <https://doi.org/10.1186/1471-2164-15-717>.

Relevant Websites

<http://eggnog-mapper.embl.de>
EggNOG.
<http://www.bioinfogenome.net/OrthoVenn/>
OrthoVenn.

Gene Duplication and Speciation

Tiratha R Singh and Ankush Bansal, Jaypee University of Information Technology, Solan, India

© 2019 Elsevier Inc. All rights reserved.

Background

Significant progress has been made in the past few decades in understanding Darwin's theory of the origin of species. Different genomic methods have helped to understand the genetic variations and gene flow that makes new species (Magadum *et al.*, 2013; Shapiro *et al.*, 2016). Although new methods have not changed the previous speculations about how species formed, they have quickened the pace of information gathering (Reams and Roth, 2015). Compiling studies on hereditary investigations would be useful to answer queries of the upcoming generations on the relative occurrence and significance of various procedures that occur during speciation (Liu *et al.*, 2016).

Many 20th century biologists viewed genes as traits of species, exquisitely tuned to current utility. This resulted in the assumption that each species should possess different genes. Gene duplication was recognized, but was implicitly assumed to have occurred recently (Rose and Oakley, 2007). Many biologists now assume that most genes have their origins in gene duplication events, which happen throughout evolutionary history. As a result, many genes form families that have persisted for hundreds of millions of years.

Gene duplication events and results of such events play a crucial role in determination of the function of novel genes. There have been various models and theories that have emerged to support the concept of gene copies (Liu *et al.*, 2016; Singh *et al.*, 2009). However, a clear picture of gene duplication events is still not clear and needs more information to come to any conclusion. Different prediction software and tools based models such as hidden Markov models give insight to evolutionary functional properties and dynamics (Singh and Pardasani, 2009). Hence, understanding the gene duplication events and speciation is an essential step towards understanding and identifying the major mechanisms that are involved in the evolution (Seehausen *et al.*, 2014).

Gene Duplication

The evolutionary understanding of gene duplication events was first performed by Haldane and John (1932), who suggested that a redundant duplicate(s) of a gene may acquire divergent mutations and eventually emerge as a new gene. A gene duplication event was first noted by Bridges (1936) in the Bar locus in *Drosophila*. A substantial increase in the number of copies of a DNA segment can be brought by various types of gene duplication (White, 1977; Raj Singh, 2008). There are various studies where gene duplication and deletion events are being systematically observed (Ma *et al.*, 2014; Schacherer *et al.*, 2004; Simillion *et al.*, 2002; Stephens, 1951). Many types of duplications are recognized: First, partial gene duplication; second, complete gene duplication; third, partial chromosomal duplication; fourth, complete chromosomal duplication; fifth, genome duplication (Innan and Kondrashov, 2010; Conant and Wolfe, 2008; Panchy *et al.*, 2016). The first four are treated as regional duplicates as they do not alter the haploid set of chromosomes. The main reason for gene duplication includes uneven crossing over (Iñiguez and Hernández, 2017; Levasseur and Pontarotti, 2011; Qian and Zhang, 2014). Uneven crossing over in two nonaligned sequences gives a duplicated region on one chromosome along with deletion on second on the basis of the size of the nonaligned region. DNA sequence duplication in tandem results in a progressive increase in uneven crossing over, which ultimately increases the number of duplicate copies (Mendivil Ramos and Ferrier, 2012).

Partial Gene Duplication

Tandem duplication events in DNA sequences may provide information about genetic evolution events in terms of the complete gene while tandem duplication in a small region, or maybe in part of gene, ultimately results in mutations and is the cause of various diseases (Hu and Worton, 1992; Hu *et al.*, 1988). The duplication arrangement can be understood by taking the inference from molecular information lying in the sequence (Toll-Riera *et al.*, 2011). For instance, structure level changes induced by changes in nucleotide and amino acid sequence level influences the protein evolution. Toll-Riera *et al.* (2011) have demonstrated this by considering a large dataset of human and mouse orthologs protein and later mapping with PDB structures. Evidence from literature suggests that duplication may arise from either homologous (Alu-Alu) recombination or nonhomologous recombination, the latter possibly mediated by topoisomerases. For the dystrophin gene, in which most duplications have been identified, these recombination events are intrachromosomal, which suggests that unequal sister chromatid exchange is the major mechanism (Toll-Riera *et al.*, 2011).

Domain Duplication and Gene Elongation

A domain is a well-defined region within a protein that either performs a specific function within a protein, such as substrate binding, or constitutes a stable, independently folding, compact structural unit within the protein that can be distinguished from

all the other parts" (Li and Makova, 2001). Theoretically, several possible relationships may be envisioned between the structural domains and the arrangements of the exons in the gene, e.g., in many globular proteins, a more or less exact correspondence exists between exons of gene and the structural domains of the protein product. Alternate splicing is one of the main reasons for exon shuffling in a duplication event as there are always chances for repetition of a similar exon set again. In a considerable number of cases, several adjacent models were found to be encoded by the same exon (Li and Makova, 2001).

The vertebrate hemoglobin α and β chains, consist of four domains, whereas their genes consist of only three exons, the second of which encodes two adjacent domains. In *Caenorhabditis elegans*, a globin-encoding gene, during the evolution of a globin gene family from a four exon ancestral gene, several lineages lost some or all of their three introns, thereby, generating panoply of exon–intron permutations (Vogel *et al.*, 2005). In the majority of cases, a domain duplication at the protein level indicates that an exon duplication has occurred at the DNA level. Moreover, many proteins of present day organisms show internal repeats often correspond to functional or structural domains within the proteins. A survey of modern genes in eukaryotes shows that the internal duplications have occurred frequently in evolution. This gene duplication is one of the most important steps in the evolution of complex genes from the simple ones (Vogel *et al.*, 2005).

Theoretically, elongation of genes can also occur by other means; for example, a mutation change converting a stop codon into a sense codon can also elongate the gene, which could be a part of recoding event (Singh and Pardasani, 2009). Similarly, either insertion of a foreign DNA segment into an exon or the occurrence of a mutation obliterating a splicing site will achieve same result. These types of molecular changes most probably disrupt the function of the elongated gene. In the vast majority of cases, such molecular changes have been found to be associated with pathological manifestations. By contrast, duplication of a structural domain is less likely to be problematic. Indeed, such a duplication can sometimes even enhance the function of the protein produced for example by increasing the number of active sites (a quantitative change), thus enabling the gene to perform its function more rapidly and efficiently or by having a synergistic effect yielding a new function (a qualitative change) (Nacher *et al.*, 2010).

Emergence of novel function can be derived from partial gene as divergence in the sequence may lead to different functions. Complete gene duplication produces two identical paralogous copies (Nacher *et al.*, 2010). Duplicated genes can be divided into two types – Variant and invariant repeats. Invariant repeats are identical or nearly identical in sequence to one another. Variant repeats are copies of a gene that, although similar to each other, differ in their sequences to a lesser or greater extent. All the genes that belong to a certain group of repeated sequences in a genome are referred to as a gene or a multigene family. Functional and nonfunctional members of a gene family may reside in close proximity to one another on the same chromosome or they may be located on different chromosomes. A member of a gene family that is located alone at a different genomic location than the other members of the family is called an orphan. The term *superfamily* was coined by Dayhoff in order to distinguish closely related proteins from distantly related ones (Dayhoff and Schwartz, 1978). Proteins that exhibit at least 50% similarity to each other at the amino acid level are considered as members of a superfamily, for example, α and β globins are classified into two separate families and together with myoglobin they form the globin superfamily (Li and Makova, 2001). An important feature associated with gene duplication is that as long as two or more copies of a gene exist in proximity to each other, the process of gene duplication can be greatly accelerated in this region, and numerous copies may be produced. There may be two reasons for the general positive correlation between genome size and number of copies of RNA specifying genes. Either a large genome requires large quantities of RNA, or the number of RNA-specifying genes is simply a passive consequence of genome enlargement by duplication. Highly repetitive genes, like rRNA genes, are generally very similar to each other (Li and Makova, 2001).

One factor responsible for homogeneity may be purifying selection. In addition to invariant repeats, the genomes of higher organisms contain numerous multigene families whose members have diverged to various extents such as genes coding for isozymes, such as lactate dehydrogenase, aldolase, creatine kinase, carbonic anhydrase, and pyruvate kinase. Isozymes are enzymes that catalyze the same biochemical reaction but may differ from one another in tissue specificity, developmental regulation, electrophoretic mobility, or biochemical properties. Isozymes are encoded by different loci, usually duplicated genes, as opposed to allozymes, which are distinct forms of the same enzyme encoded by different alleles at a single locus (Li and Makova, 2001; Vogel *et al.*, 2005).

Exon Shuffling

There are three types of exon shuffling: (1) Exon duplication, (2) exon insertion, and (3) exon deletion. Exon duplication refers to the duplication of one or more exons in a gene and so is a type of internal duplication (Pathy, 1999). Exon insertion is the process by which structural or functional domains are exchanged between proteins or inserted into a protein. Exon deletion results in the removal of a segment of amino acids from the protein. All types of shuffling have occurred in the evolutionary process of creating new genes (Kolkman and Stemmer, 2001; Pathy, 1999). Exon shuffling could be represented through various phases as shown in Fig. 1.

Mosaic or Chimeric Proteins

A mosaic or chimeric protein is a protein encoded by a gene that contains regions that are also found in other genes (Nicolson, 2015; Singer and Nicolson, 1972). The existence of such proteins indicates that exon shuffling has occurred during the evolutionary history of their genes. The first described mosaic protein was tissue plasminogen activator (Iñiguez and Hernández, 2017). For an exon to be inserted, deleted, or duplicated without causing a frameshift in the reading frame, certain phase limitations of the exonic structure of the gene must be respected. Mosaic proteins can be made when two adjacent genes are transcribed together and are therefore made into the same protein.

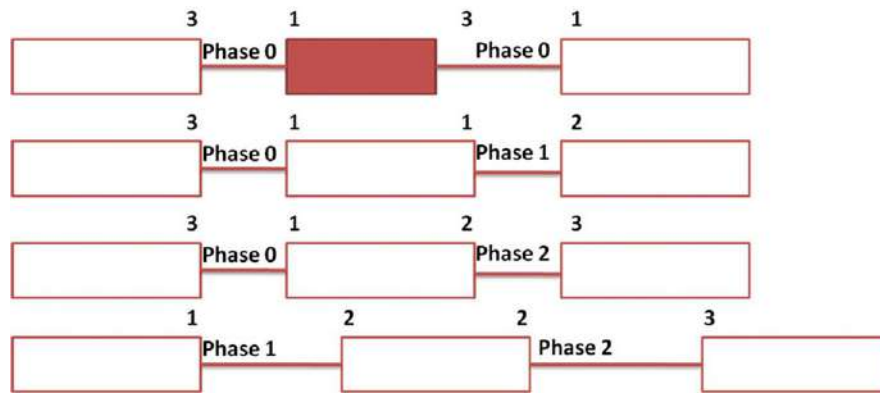


Fig. 1 Schematic representation of exon shuffling. Numbers above the box represent the position of nucleotide.

To understand this, we need to consider introns in terms of their possible positions relative to the coding regions (Nicolson, 2014). Introns residing between coding regions are classified into three types according to the way in which the coding region is interrupted (Jeon, 2004). An intron is of phase 0 if it lies between two codons, of phase 1 if it lies between the first and second nucleotide of a codon, and of phase 2 if it lies between the second and third nucleotides of a codon. Exons are grouped into classes according to the phases of their flanking introns. Here are four middle exons, said to be (0–0), (0–1), (0–2), and (1–2) exons. An exon that is flanked by introns of the same phase at both ends is called a symmetrical exon, otherwise it is asymmetrical. The first exon (middle) is a symmetrical exon, represented by black box. The length of a symmetrical exon is always a multiple of three nucleotides. Only symmetrical exons can be duplicated in tandem or deleted without affecting the reading frame.

Duplication or deletion of asymmetrical exons would disrupt the reading frame downstream. Similarly, only symmetrical exons can be inserted into introns, but with the restriction of, a 0–0 exon can only be inserted in phase-0-introns, a 1–1 exon is inserted into phase 1 introns, and 2–2 exons into phase-2 introns for avoiding frameshifts. All the exons coding for the modules of mosaic proteins are symmetrical. Since nonrandom intron phase usage is a necessary consequence of exon duplication or insertion, this property may be used as a diagnostic feature of gene assembly through exon shuffling. In terms of splicing, introns are classified into two categories, self-splicing and spliceosomal (Wang *et al.*, 2013). The vast majority of introns in eukaryotic nuclear genes are spliceosomal. Self-splicing introns play a vital role in their own removal, some regions of the introns are involved in self-complementary interactions important for forming the 3-D structure possessing splicing activity. Exon shuffling probably did not play a role in the formation of genes in the early stages of evolution. Exon shuffling came to full bloom with the evolution of spliceosomal introns, which do not play a role in their self excision. These introns contain mainly nonessential parts and therefore could accommodate quantities of “foreign” DNA (Wang *et al.*, 2013).

Exonization and Pseudoexonization

Exonization is the process through which an intronic sequence becomes an exon. An exon created by exonization must abide by the same rules of exon insertion (Schmitz and Brosius, 2011). The opposite process is called pseudoexonization. It occurs when nonfunctionalization affects a single exon rather than the entire gene. The result is the creation of a pseudoexon, and the most obvious consequence of such a process is gene abridgement (as opposed to gene elongation). Pseudoexons are often created by the nonfunctionalization of internal gene duplications, for example, the aggrecan gene in rat contains 18 repeated exons and one pseudoexon. Some complex biological functions that require several enzymes may be specified by genes encoding different combinations of protein modules. In some species we may find single-module proteins, while in others we may find different combinations of multimodular proteins, for example, genes in a multistep process in the synthesis of fatty acids from Acetyl Co-A require seven enzymatic activities and an acyl-carrier protein. In fungi, these activities are distributed between two nonidentical polypeptides encoded by two unlinked intronless genes, FAS1 and FAS2. FAS1 encodes two and FAS2 encodes five enzymatic activities. In most bacteria, however, these functions are carried on by discrete monofunctional proteins (Schweizer and Hofmann, 2004).

In animals, key functions associated with fatty acid metabolism are controlled with a single polypeptide chain called fatty acid synthase. The fatty acid synthase gene in fungi and mammals are most probably mosaic proteins that have assembled from single-domain proteins like the ones found in bacteria (Stower, 2013). The fact that arrangement of domains is different in fungi from that in mammals indicates not only that the two lineages evolved multimodularity independently, but also that different strategies may be employed in the assembly of genes encoding multimodular proteins.

Nested and Overlapping Genes and Their Association With Speciation

In addition to gene duplication and exon shuffling, many other mechanisms for producing new genes or polypeptides are available. Few such entities are overlapping genes, pseudogenes, and nested genes.

Overlapping Genes

A DNA fragment (segment) can code for more than one gene product by using different reading frames or different initiation codons. This phenomenon of overlapping genes is widespread in DNA and RNA viruses, as well as in organelles, and bacteria, also known in nuclear eukaryotic genomes (Makalowska *et al.*, 2005). Overlapping genes can also arise by the use of the complementary strand of a gene; for example, genes specifying tRNA^{ILE} and tRNA^{GLN} in the human mitochondrial genome are located on different strands and there is a three-nucleotide overlap between these that reads 5'-CTA-3' in the former and 5'-TAG-3' in the latter (Makalowska *et al.*, 2005). The rate of evolution is expected to be slower in stretches of DNA encoding overlapping genes than in similar DNA sequences that only use one reading frame. The reason is that the proportion of nondegenerate sites is higher in overlapping genes than in nonoverlapping genes, thus vastly reducing the proportion of synonymous mutations out of total number of mutations (Johnson and Chisholm, 2004; Normark *et al.*, 1983). Since gene duplication is a widespread phenomenon for the maintenance of overlapping genes, it would require quite strong selective pressure (against increasing genome size). Studies on aminoacyl tRNA synthetases indicate that overlapping genes may have played a momentous role in the evolution of life (Johnson and Chisholm, 2004; Normark *et al.*, 1983).

Alternate Splicing

Alternative splicing of a primary RNA transcript results in the production of different mRNAs from the same DNA segment, which in turn may be translated into different polypeptides (Baralle and Giudice, 2017). There are two types of exons: Constitutive, that is, exons that are included within all the mRNAs transcribed from a gene, and facultative, that is, exons that are sometimes spliced in and sometimes spliced out (Kornblihtt *et al.*, 2013). There are different types of alternative splicing; the most trivial form is the intron retention (Lee and Rio, 2015). However, more commonly, intron retention results in the premature termination of translation due to frameshifts. Sometimes, alternative splicing involves the use of alternative internal donor or acceptor sites, that is, excisions of introns of different lengths with complementary variation in the size of neighboring exons. Such use of competing splice sites was found in several transcription units of adenoviruses, as well as in eukaryotic cells such as the transformer gene in *Drosophila melanogaster* (Lee and Rio, 2015). Some cases of alternative splicing involved the use of mutually exclusive exons, that is, two exons are never spliced out together, nor are both retained in the same mRNA for example, M1 and M2 from a single gene by mutually exclusive use of exons 9 and 10. A special case of mutual exclusivity is the cassette exon (Roy *et al.*, 2013). A cassette is either spliced in or spliced out in the alternative mRNA molecules. Alternative splicing has often been used as a means of developmental regulation (Wang *et al.*, 2015). A very intriguing situation is seen in several genes involved in the process of sex determination in *D. melanogaster*. At least three genes, doublesex (*dsx*), Sexlethal (*sxl*), and transformer (*tra*), are spliced differently in males and females (Wang *et al.*, 2015). There is rich literature available on alternate splicing and its distribution in almost all available lineages.

Intron-Encoded Proteins and Nested Genes

An intron may sometimes contain an ORF that encodes a protein or part of a protein that is completely different in function from the one encoded by the flanking exons (Kumar, 2009). In many cases, intron-encoded protein genes are located within type-I self-splicing introns. From a mechanistic point of view, an intron-encoded protein gene that is transcribed from the same strand as the neighboring exons may be regarded as special instance of alternative splicing (Lee and Chang, 2013; Yu *et al.*, 2005). When an intron-encoded protein gene is transcribed from the opposite strand of the other gene, it is referred to as a nested gene. A case of nested genes was found in *Drosophila*, where a pupal cuticle protein gene is encoded on the positive strand of an intron within the gene encoding the purine pathway enzyme glycineamide ribotide-transformylase (Kaer *et al.*, 2011).

Functional Convergence

Function of a protein is frequently determined by only a few of its amino acids; a protein performing one function may sometimes arise from a gene encoding a protein performing a markedly different function. If the new function is performed in other species by proteins of unrelated structure and descent, functional convergence may occur. The myoglobin of abalone *Sulculus* consists of 377 amino acids, which are 2.5 times larger than myoglobins belonging to the globin superfamily (Suzuki *et al.*, 1996). Functional convergence provides a robust parameter associated with the existence of a functional protein for a family or superfamily. This convergence is reflected in various levels of organisms at the family and superfamily level based upon rate of selection pressure.

RNA Editing

RNA editing is a molecular process by which protein-coding gene change its message. One of the most common types of RNA editing is C-to-U conversion. This conversion may occur partially or completely in some tissues but not in others, leading to differential gene expression. Occasionally, it can produce a new protein with a different function from the unedited transcript, for example, apolipoprotein B gene, one of the lipid carriers in the blood (Cooper, 1999). There are two types – Apo B-100 and apo B-48. Despite differences in length, amino acid sequences of the gigantic protein apo B-100 (4536 amino acid) with that of

apo B-48 (2152 amino acid), the result of alignment for the alignable part is 100% identity. It was found that apo B-48 is translated from a very long mRNA that is identical to that of apo B-100 with the execution of an in-frame stop codon resulting from the RNA editing of codon 2153 from CAA (Gln) to UAA (stop). Thus, by using RNA editing, two quite different proteins are produced from the same gene (Cooper, 1999). (Figs. 2,3,4,5).

Gene Sharing

Gene sharing means that a gene acquires and maintains a second function without divergent duplication and without loss of the primary function. Gene sharing may, however, require a change in the regulation system of tissue specificity or developmental timing (Cvekl and Zheng, 2009; Patthy, 2007). In literature, the term “multifunctional protein” is frequently used instead of “gene sharing.” Gene sharing was first discovered in crystallins, which are the major water-soluble proteins in the

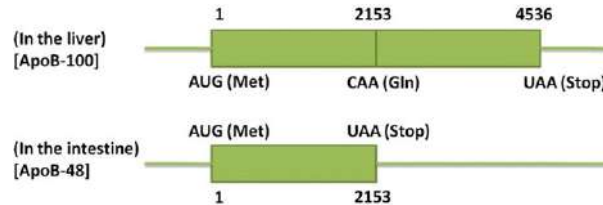


Fig. 2 Mechanism of RNA editing may leads to truncation.

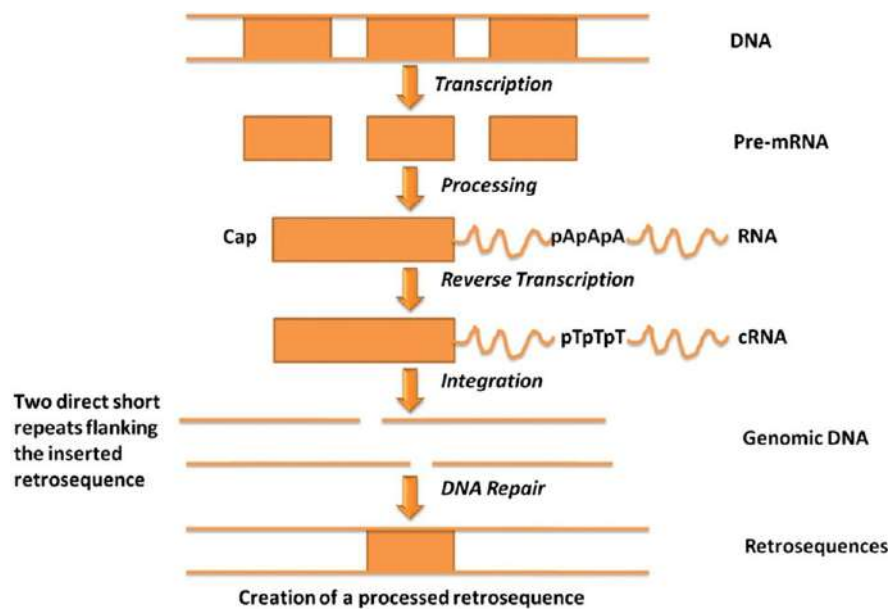


Fig. 3 Molecular mechanism explaining molecular repair.

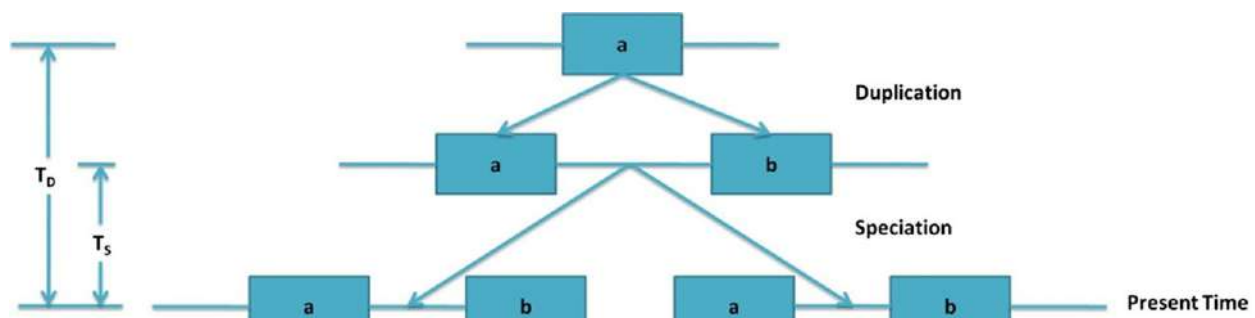


Fig. 4 Duplication and Speciation Events.

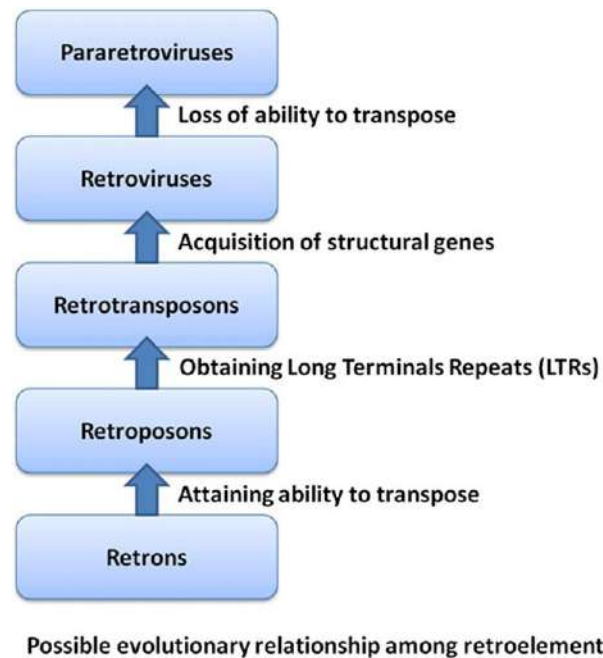


Fig. 5 Evolutionary relationship among retro elements.

eye lens, and whose function is to maintain lens transparency and proper light refraction (Cvekl and Zheng, 2009). Gene sharing might be a fairly common phenomenon, also suspected for several proteins in the cornea and other tissues. Gene sharing clearly adds to the compactness of the genome, even though compactness does not seem to have a high priority in eukaryotes (McKay, 2008).

Molecular Repair

The more we learn about the evolution of genes, the more we recognize that true innovations are only rarely produced during evolution. Many proteins that were originally considered to be relatively recent evolutionary additions turned out to be derived from ancient proteins (McKay, 2008). Besides, all the discussed mechanisms that facilitate tinkering at the molecular level are gene conversion and transposition. We may deduce that molecular tinkering is most probably the paradigm of molecular evolution, and it is reasonable to assume that tinkering also characterizes the evolution of morphological, anatomical, and physiological traits as well.

Gene Gain and Loss and Speciation Events

Gene Loss

More than 7000 diseases and disorders were documented in research papers and literature sources, which state that mutations have a crucial role in destroying and gaining function at the gene level (McKusick, 2007). A number of such mutations either get deleted at a very fast rate from population or sustained at very low frequency due to genetic drift. If there are many copies of genes present and functions normally then deleterious mutations which occur more often collate together compared to significant ones. Repeated duplicate genes usually become nonfunctional instead of being a new gene (Stone *et al.*, 1998).

Lateral Gene Transfer

Lateral gene transfer (LGT) is a process that makes complicated distribution of genes and dissimilar phylogenies with an rRNA tree. Still there is a debate on robust organismal phylogeny over extensive LGT (Kettler *et al.*, 2007). In case there is a presence of a core set of genes that are resistant to LGT, then there should be a reflection of vertical descent and ascent along with cell division. Moreover, various next generation sequencing techniques like metagenomics, genomics, and proteomics pave a path in understanding the core mechanism of LGT. In particular, it will be informative to know the complete genome diversity (Kettler *et al.*, 2007).

Dating Gene Duplication and other Computations

Dating Gene Duplications

Two genes are said to be paralogous if they are derived from a duplication event, but orthologous if they are derived from a speciation event.

Here, genes a and b were derived from the duplication of an ancestral gene and are paralogous, while gene a from species 1 and gene a from species 2 are orthologous, as are genes b from species 1 and gene b from species 2 (Chen *et al.*, 2000). We can estimate the date of duplication, T_D , from sequence data if we know the rate of substitution in genes a and b. The rate of substitution can be estimated from the number of substitutions between the orthologous genes in conjunction with knowledge of the time of divergences T_S between species 1 and 2. For gene a, let K_a be the number of substitutions per site between the two species. Then the rate of substitution in gene a, γ_a is estimated by:

$$\gamma_a = \frac{K_a}{2T_S} \quad (1)$$

The rate of substitution in gene b, γ_b can also be obtained in a similar way. The average substitution rate for the two genes:

$$\gamma = \frac{\gamma_a + \gamma_b}{2} \quad (2)$$

To estimate T_D , we need to know the number of substitutions per site between gene a and b (K_{ab}). This number can be estimated from four pairwise comparisons: (1) Gene a from species 1 and gene b from species 2, (2) (Robinson-Rechavi *et al.*, 2004). Gene a from species 2 and gene b from species 1. (3) Gene a and gene b from species 1. (4) Gene a and b from species 2. From these four estimates, we can compute the average value for K_{ab} from which we can estimate T_D as:

$$TD = \frac{K_{ab}}{2\gamma} \quad (3)$$

In case of protein coding genes, by using the number of synonymous and nonsynonymous substitutions separately, we can obtain two independently estimates of T_D . The average of these two may be used as the final estimate of T_D . Sometimes problems are due to concerted T_D estimation. Another method for dating gene duplication events is to consider the phylogenetic distribution of genes in conjunction with paleontological data pertinent to the divergence date of the species in question (Zhou *et al.*, 2010).

Unprocessed Pseudogenes

The silencing of a gene due to deletion in a nucleotide and causing deleterious mutation ultimately results in production of pseudogene. Such pseudogenes do not undergo RNA processing. Unprocessed pseudogenes may be the result of derivation from nonfunctionality of a duplicate functional gene. There is much less chance that functional genes come into existence without duplication. Unprocessed pseudogenes may result in various diseases and disorders like frameshift mutation, nonmature stop codon, and misorientation of sliced transcripts or regulatory elements; therefore, it is easy to identify a change in sequence in terms of mutations resulting directly in gene silencing (Tutar, 2012). It is possible in some cases to identify the mutation responsible for nonfunctionalization of a gene through a phylogenetic analysis, for example, human pseudogene $\psi\eta$ in the b-globin family contains numerous defects, each of which could have been sufficient to silence it (Pink *et al.*, 2011). The b-globin clusters in chimpanzee and gorilla were found to contain the same number of genes and pseudogenes as in humans, indicating that the pseudogenes were created and silenced before these three species diverged from one another.

Interestingly, mutations that cause nonfunctionalization are only rarely missense mutations, most probably because such mutations result in the production of defective proteins that may be incorporated into final biological products and thus may have deleterious effects. Because unprocessed pseudogenes are usually created by duplication, they usually found in the neighborhood of the homologous functional genes from which they have been derived.

Unitary Pseudogenes

The pseudogene has no functional correlation with the human genome, called unitary pseudogene. Guinea pigs and humans suffer from scurvy unless they consume L-ascorbic acid in their diet, because they lack a protein called L-gulonono-y-lactone oxidase, an enzyme that catalyzes the terminal step in L-ascorbic acid synthesis (Pink *et al.*, 2011). In humans, L-gulonono-y-lactone oxidase is a pseudogene that contains molecular defects as the deletion of at least two exons (out of 12), deletions and insertion of nucleotides in the reading frame, and obliterations of intron-exon boundaries (Pink *et al.*, 2011). It has been assumed that the guinea pig and human ancestors managed to survive on a naturally ascorbic acid-rich diet; hence, the loss of this enzyme did not reflect a disadvantage.

Case Studies

There are several case studies where bioinformatics has been implemented successfully to connect molecular evolution events with their functional consequences. Genomics studies paved the path to understand variation at the genomic level through phylogenetic approaches (Pradhan *et al.*, 2017; Wang *et al.*, 2018). Further network reconstruction approaches using co-expression network modules also helps to trace the gene duplication events (Feng *et al.*, 2016; Malviya *et al.*, 2016). For instance, phylogenetic investigation of human FGFR-bearing paralogs favors piecemeal duplication theory of vertebrate genome evolution (Ajmal *et al.*, 2014). All given studies indicate that duplication and speciation events are very much diverse in nature and take many years to evolve and confirm to contribute to significant changes.

Tools and Methods

S.No.	Tool or method	Description	References
1.	Control-FREEC CNV detection: HTS analysis	A tool for detection of copy-number changes and allelic imbalances (including LOH) using deep-sequencing data. Control-FREEC automatically computes, normalizes, segments copy number and beta allele frequency profiles, then calls copy number alterations and LOH. The control (matched normal) sample is optional. The program can also use mappability data (files created by GEM).	Boeva <i>et al.</i> (2012, 2011)
2.	mrCaNaVaR micro-read Copy number variant regions	A copy number caller that analyzes the whole-genome next-generation sequence mapping read depth to discover large segmental duplications and deletions. mrCaNaVaR also has the capability of predicting absolute copy numbers of genomic intervals.	Alkan <i>et al.</i> (2009)
3.	forestSV Structural variant detection: HTS analysis	Integrates prior knowledge about the characteristics of SVs. forestSV is a statistical learning approach, based on Random Forests, that leads to improved discovery in high throughput sequencing (HTS) data. This application offers high sensitivity and specificity coupled with the flexibility of a data-driven approach. It is particularly well suited to the detection of rare variants because it is not reliant on finding variant support in multiple individuals.	Michaelson and Sebat (2012)
4.	CTDGFinder Duplication detection: HTS analysis	Formalizes and automates the identification of clusters of tandemly duplicated genes (CTDGs) by examining the physical distribution of individual members of families of duplicated genes across chromosomes. Application of CTDGFinder accurately identified CTDGs for many well-known gene clusters (e.g., Hox and beta-globin gene clusters) in the human, mouse, and 20 other mammalian genomes. Examination of human genes showing tissue-specific enhancement of their expression by CTDGFinder identified members of several well-known gene clusters (e.g., cytochrome P450s and olfactory receptors) and revealed that they were unequally distributed across tissues. By formalizing and automating CTDG identification, CTDGFinder will facilitate understanding of CTDG evolutionary dynamics, their functional implications, and how they are associated with phenotypic diversity.	Ortiz and Rokas (2017)
5.	cn.MOPS Copy number estimation by a Mixture Of PoissonS	A data processing pipeline for copy number variations and aberrations (CNVs and CNAs) from next generation sequencing (NGS) data. The package supplies functions to convert BAM files into read count matrices or genomic ranges objects, which are the input objects for cn.MOPS. It models the depths of coverage across samples at each genomic position. Therefore, it does not suffer from read count biases along chromosomes. Using a Bayesian approach, cn.MOPS decomposes read variations across samples into integer copy numbers and noise by its mixture components and Poisson distributions, respectively.	Klambauer <i>et al.</i> (2012)
6.	RDXplorer CNV detection: HTS analysis	A computational tool for copy number variants (CNV) detection in whole human genome sequence data using read depth (RD) coverage. CNV detection is based on the event-wise testing (EWT) algorithm. The read depth coverage is estimated in nonoverlapping intervals (100 bp Windows) across an individual genome based on the pileup generated by SAMTools.	Yoon <i>et al.</i> (2009)
7.	cnvHiTSeq CNV detection: HTS analysis	A set of Java-based command-line tools for detecting copy number variants (CNVs) using next-generation sequencing data.	Bellos <i>et al.</i> (2012)

Source: *This data is referred from OMICSTOOLS resource. There are various other tools which may be referred at <https://omicstools.com/duplication-detection-category>.

Conclusion

Gene duplication and speciation are two prominent mechanisms for finding clues for evolution. Phylogenetic analysis helps researchers to understand the ancestral association of species or sequence of their interest. Gene duplication events, exon shuffling,

and speciation have a potent role in the process of evolution and to study convergence and divergence from ancestral data. This article provides descriptive information about basic concepts of phylogeny that may help students and researchers to become aware of terminologies used in gene duplication and speciation analysis. It is presented in a way where basic to advanced level information is being compiled on the diverse topics and it is estimated that this information will serve as comprehensive information to students, faculty members, and researchers working in this area. It reflects how bioinformatics can shape a computational pipeline for the analysis of biological data in an evolutionary scenario and will help the evolutionary biologists also to implement bioinformatics at some level for further exploration of basic principles.

See also: Genetics and Population Analysis. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Sequence Analysis. Sequence Composition

References

- Ajmal, W., Khan, H., Abbasi, A.A., 2014. Phylogenetic investigation of human FGFR-bearing paralogs favors piecemeal duplication theory of vertebrate genome evolution. *Molecular Phylogenetics and Evolution* 81, 49–60. Available at: <https://doi.org/10.1016/j.ympev.2014.09.009>.
- Alkan, C., Kidd, J.M., Marques-Bonet, T., et al., 2009. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature Genetics* 41, 1061–1067. Available at: <https://doi.org/10.1038/ng.437>.
- Baralle, F.E., Giudice, J., 2017. Alternative splicing as a regulator of development and tissue identity. *Nature Reviews Molecular Cell Biology* 18, 437–451. Available at: <https://doi.org/10.1038/nrm.2017.27>.
- Bellos, E., Johnson, M.R., M. Coin, L.J., 2012. cnvHiTSeq: Integrative models for high-resolution copy number variation detection and genotyping using population sequencing data. *Genome Biology* 13, R120. Available at: <https://doi.org/10.1186/gb-2012-13-12-r120>.
- Boeva, V., Popova, T., Bleakley, K., et al., 2012. Control-FREEC: A tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics* (Oxford, England) 28, 423–425. Available at: <https://doi.org/10.1093/bioinformatics/btr670>.
- Boeva, V., Zinovyev, A., Bleakley, K., et al., 2011. Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization. *Bioinformatics* (Oxford, England) 27, 268–269. Available at: <https://doi.org/10.1093/bioinformatics/btq635>.
- Bridges, C.B., 1936. The bar "gene" a duplication. *Science* 83, 210–211. Available at: <https://doi.org/10.1126/science.83.2148.210>.
- Chen, K., Durand, D., Farach-Colton, M., 2000. NOTUNG: A program for dating gene duplications and optimizing gene family trees. *Journal of Computational Biology* 7, 429–447. <https://doi.org/10.1089/106652700750050871>.
- Conant, G.C., Wolfe, K.H., 2008. Turning a hobby into a job: How duplicated genes find new functions. *Nature Reviews Genetics* 9, nrg2482. Available at: <https://doi.org/10.1038/nrg2482>.
- Cooper, D.N., 1999. *Human Gene Evolution*. Elsevier.
- Cvekl, A., Zheng, D., 2009. Gene sharing and evolution. *Human Genomics* 4, 66–67. Available at: <https://doi.org/10.1186/1479-7364-4-1-66>.
- Dayhoff, M.O., Schwartz, R.M., 1978. Chapter 22: A model of evolutionary change in proteins, in: *In Atlas of Protein Sequence and Structure*.
- Feng, K., Liu, F., Zou, J., et al., 2016. Genome-wide identification, evolution, and co-expression network analysis of mitogen-activated protein kinase kinases in *Brachypodium distachyon*. *Frontiers in Plant Science* 7, 1400. Available at: <https://doi.org/10.3389/fpls.2016.01400>.
- Haldane, J.B.S., John, B.S., 1932. *The Causes of Evolution*. London: Longmans, Green.
- Hu, X.Y., Burghes, A.H., Ray, P.N., et al., 1988. Partial gene duplication in Duchenne and Becker muscular dystrophies. *Journal of Medical Genetics* 25, 369–376.
- Hu, X., Worton, R.G., 1992. Partial gene duplication as a cause of human disease. *Human Mutation* 1, 3–12. Available at: <https://doi.org/10.1002/humu.1380010103>.
- Iñiguez, L.P., Hernández, G., 2017. The evolutionary relationship between alternative splicing and gene duplication. *Frontiers in Genetics* 8. Available at: <https://doi.org/10.3389/fgene.2017.00014>.
- Innan, H., Kondrashov, F., 2010. The evolution of gene duplications: Classifying and distinguishing between models. *Nature Reviews Genetics* 11, nrg2689. Available at: <https://doi.org/10.1038/nrg2689>.
- Jeon, K.W., 2004. *International Review of Cytology: A Survey of Cell Biology*. Academic Press.
- Johnson, Z.I., Chisholm, S.W., 2004. Properties of overlapping genes are conserved across microbial genomes. *Genome Research* 14, 2268–2272. Available at: <https://doi.org/10.1101/gr.2433104>.
- Kaer, K., Branovets, J., Hallikma, A., Nigumann, P., Speek, M., 2011. Intronic L1 retrotransposons and nested genes cause transcriptional interference by inducing intron retention, exonization and cryptic polyadenylation. *PLOS ONE* 6, e26099. Available at: <https://doi.org/10.1371/journal.pone.0026099>.
- Kettler, G.C., Martiny, A.C., Huang, K., et al., 2007. Patterns and Implications of gene gain and loss in the evolution of *prochlorococcus*. *PLOS Genetics* 3, e231. Available at: <https://doi.org/10.1371/journal.pgen.0030231>.
- Klambauer, G., Schwarzbauer, K., Mayr, A., et al., 2012. cnMOPS: Mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic Acids Research* 40, e69. Available at: <https://doi.org/10.1093/nar/gks003>.
- Kolkman, J.A., Stemmer, W.P.C., 2001. Directed evolution of proteins by exon shuffling. *Nature Biotechnology* 19. Available at: <https://doi.org/10.1038/88084>.
- Kornblihtt, A.R., Schor, I.E., Alló, M., et al., 2013. Alternative splicing: A pivotal step between eukaryotic transcription and translation. *Nature Reviews Molecular Cell Biology* 14, 153–165. Available at: <https://doi.org/10.1038/nrm3525>.
- Kumar, A., 2009. An overview of nested genes in Eukaryotic genomes. *Eukaryotic Cell* 8, 1321–1329. Available at: <https://doi.org/10.1128/EC.00143-09>.
- Lee, Y.C.G., Chang, H.-H., 2013. The evolution and functional significance of nested gene structures in *Drosophila melanogaster*. *Genome Biology and Evolution* 5, 1978–1985. Available at: <https://doi.org/10.1093/gbe/evt149>.
- Lee, Y., Rio, D.C., 2015. Mechanisms and regulation of alternative pre-mRNA splicing. *Annual Review of Biochemistry* 84, 291–323. Available at: <https://doi.org/10.1146/annurev-biochem-060614-034316>.
- Levasseur, A., Pontarotti, P., 2011. The role of duplications in the evolution of genomes highlights the need for evolutionary-based approaches in comparative genomics. *Biology Direct* 6, 11. Available at: <https://doi.org/10.1186/1745-6150-6-11>.
- Liu, Z., Tavares, R., Forsythe, E.S., et al., 2016. Evolutionary interplay between sister cytochrome P450 genes shapes plasticity in plant metabolism. *Nature Communications* 7. Available at: <https://doi.org/10.1038/ncomms13026>.
- Li, W.-H., Makova, K.D., 2001. *Domain Duplication and Gene Elongation*. ELS. John Wiley & Sons, Ltd. Available at: <https://doi.org/10.1038/npg.els.0005097>.
- Malviya, N., Jaiswal, P., Yadav, D., 2016. Genome-wide characterization of Nuclear Factor Y (NF-Y) gene family of sorghum [*Sorghum bicolor* (L.) Moench]: a bioinformatics approach. *Physiol. Mol. Biol. Plants* 22, 33–49. Available at: <https://doi.org/10.1007/s12298-016-0349-z>.

- Magadum, S., Banerjee, U., Murugan, P., Gangapur, D., Ravikesavan, R., 2013. Gene duplication as a major force in evolution. *Journal of Genetics* 92, 155–161.
- Makalowska, I., Lin, C.-F., Makalowski, W., 2005. Overlapping genes in vertebrate genomes. *Computational Biology and Chemistry* 29, 1–12. Available at: <https://doi.org/10.1016/j.compbiolchem.2004.12.006>.
- Ma, Q., Reeves, J.H., Liberles, D.A., *et al.*, 2014. A phylogenetic model for understanding the effect of gene duplication on cancer progression. *Nucleic Acids Research* 42, 2870–2878. Available at: <https://doi.org/10.1093/nar/gkt1320>.
- McKay, S.A.B., 2008. Gene sharing and evolution: The diversity of protein functions. By Joram Piatigorsky. *The Quarterly Review of Biology* 83, 99. Available at: <https://doi.org/10.1086/586921>.
- McKusick, V.A., 2007. Mendelian inheritance in man and its online version, OMIM. *American Journal of Human Genetics* 80, 588–604.
- Mendivil Ramos, O., Ferrier, D.E.K., 2012. Mechanisms of gene duplication and translocation and progress towards understanding their relative contributions to animal genome evolution [WWW Document]. *International Journal of Evolutionary Biology*. Available at: <https://doi.org/10.1155/2012/846421>.
- Michaelson, J.J., Sebat, J., 2012. forestSV: Structural variant discovery through statistical learning. *Nature Methods* 9, 819–821. Available at: <https://doi.org/10.1038/nmeth.2085>.
- Nacher, J.C., Hayashida, M., Akutsu, T., 2010. The role of internal duplication in the evolution of multi-domain proteins. *Biosystems* 101, 127–135. Available at: <https://doi.org/10.1016/j.biosystems.2010.05.005>.
- Nicolson, G.L., 2015. Cell membrane fluid-mosaic structure and cancer metastasis. *Cancer Research* 75, 1169–1176. Available at: <https://doi.org/10.1158/0008-5472.CAN-14-3216>.
- Nicolson, G.L., 2014. The Fluid – Mosaic Model of Membrane Structure: Still relevant to understanding the structure, function and dynamics of biological membranes after more than 40 years. *Biochimica et Biophysica Acta (BBA) – Biomembranes* 1838, 1451–1466. Available at: <https://doi.org/10.1016/j.bbamem.2013.10.019>.
- Normark, S., Bergstrom, S., Edlund, T., *et al.*, 1983. Overlapping Genes. *Annual Review of Genetics* 17, 499–525. Available at: <https://doi.org/10.1146/annurev.ge.17.120183.002435>.
- Ortiz, J.F., Rokas, A., 2017. CTDFinder: A novel homology-based algorithm for identifying closely spaced clusters of tandemly duplicated genes. *Molecular Biology and Evolution* 34, 215–229. Available at: <https://doi.org/10.1093/molbev/msw227>.
- Panchy, N., Lehti-Shiu, M., Shiu, S.-H., 2016. Evolution of gene duplication in plants[OPEN]. *Plant Physiology* 171, 2294–2316. Available at: <https://doi.org/10.1104/pp.16.00523>.
- Patthy, L., 2007. A general theory of gene sharing. *Nature Genetics* 39, ng0607–ng0701. Available at: <https://doi.org/10.1038/ng0607-701>.
- Patthy, L., 1999. Genome evolution and the evolution of exon-shuffling – A review. *Gene* 238, 103–114.
- Pink, R.C., Wicks, K., Caley, D.P., *et al.*, 2011. Pseudogenes: Pseudo-functional or key regulators in health and disease? *RNA* 17, 792–798. Available at: <https://doi.org/10.1261/rna.2658311>.
- Pradhan, S., Kant, C., Verma, S., Bhatia, S., 2017. Genome-wide analysis of the CCCH zinc finger family identifies tissue specific and stress responsive candidates in chickpea (*Cicer arietinum* L.). *PLOS ONE* 12. Available at: <https://doi.org/10.1371/journal.pone.0180469>.
- Qian, W., Zhang, J., 2014. Genomic evidence for adaptation by gene duplication. *Genome Research* 24, 1356–1362. Available at: <https://doi.org/10.1101/gr.172098.114>.
- Raj Singh, T., 2008. Mitochondrial gene rearrangements: New paradigm in the evolutionary biology and systematics. *Bioinformation* 3, 95–97.
- Reams, A.B., Roth, J.R., 2015. Mechanisms of gene duplication and amplification. *Cold Spring Harbor Perspectives in Biology* 7. Available at: <https://doi.org/10.1101/cshperspect.a016592>.
- Robinson-Rechavi, M., Boussau, B., Laudet, V., 2004. Phylogenetic dating and characterization of gene duplications in vertebrates: The cartilaginous fish reference. *Molecular Biology and Evolution* 21, 580–586. Available at: <https://doi.org/10.1093/molbev/msh046>.
- Rose, M.R., Oakley, T.H., 2007. The new biology: Beyond the modern synthesis. *Biology Direct* 2, 30. Available at: <https://doi.org/10.1186/1745-6150-2-30>.
- Roy, B., Haupt, L.M., Griffiths, L.R., 2013. Review: Alternative Splicing (AS) of genes as an approach for generating protein complexity. *Current Genomics* 14, 182–194. Available at: <https://doi.org/10.2174/1389202911314030004>.
- Schacherer, J., Tourrette, Y., Souciet, J.-L., Potier, S., de Montigny, J., 2004. Recovery of a function involving gene duplication by retroposition in *Saccharomyces cerevisiae*. *Genome Research* 14, 1291–1297. Available at: <https://doi.org/10.1101/gr.2363004>.
- Schmitz, J., Brosius, J., 2011. Exonization of transposed elements: A challenge and opportunity for evolution. *Biochimie* 93, 1928–1934. Available at: <https://doi.org/10.1016/j.biochi.2011.07.014>.
- Schweizer, E., Hofmann, J., 2004. Microbial Type I Fatty Acid Synthases (FAS): Major players in a network of cellular FAS systems. *Microbiology and Molecular Biology Reviews*, 68, pp. 501–517. Available at: <https://doi.org/10.1128/MMBR.68.3.501-517.2004>.
- Seehausen, O., Butlin, R.K., Keller, I., *et al.*, 2014. Genomics and the origin of species. *Nature Reviews Genetics* 15, nrg3644. Available at: <https://doi.org/10.1038/nrg3644>.
- Shapiro, B.J., Leducq, J.-B., Mallet, J., 2016. What is speciation? *PLOS Genetics* 12, e1005860. Available at: <https://doi.org/10.1371/journal.pgen.1005860>.
- Simillion, C., Vandeputte, K., Montagu, M.C.E.V., Zabeau, M., Peer, Y.V. de, 2002. The hidden duplication past of *Arabidopsis thaliana*. *Proceedings of the National Academy of Sciences* 99, 13627–13632. Available at: <https://doi.org/10.1073/pnas.212522399>.
- Singer, S.J., Nicolson, G.L., 1972. The fluid mosaic model of the structure of cell membranes. *Science* 175, 720–731.
- Singh, T.R., Pardasani, K.R., 2009. Ambush hypothesis revisited: Evidences for phylogenetic trends. *Computational Biology and Chemistry* 33, 239–244. Available at: <https://doi.org/10.1016/j.compbiolchem.2009.04.002>.
- Singh, T.R., Tsagkogeorga, G., Delsuc, F., *et al.*, 2009. Tunicate mitogenomics and phylogenetics: Peculiarities of the *Herdmania momus* mitochondrial genome and support for the new chordate phylogeny. *BMC Genomics* 10, 534. Available at: <https://doi.org/10.1186/1471-2164-10-534>.
- Stephens, S.G., 1951. Possible significance of duplication in evolution. In: Demerec, M. (Ed.), *Advances in Genetics*. Academic Press, pp. 247–265. Available at: [https://doi.org/10.1016/S0065-2660\(08\)60237-0](https://doi.org/10.1016/S0065-2660(08)60237-0).
- Stone, D.L., Agarwala, R., Schäffer, A.A., *et al.*, 1998. Genetic and physical mapping of the McKusick-Kaufman syndrome. *Human Molecular Genetics* 7, 475–481. Available at: <https://doi.org/10.1093/hmg/7.3.475>.
- Stower, H., 2013. Alternative splicing: Regulating *Alu* element “exonization”. *Nature Reviews Genetics* 14, nrg3428. Available at: <https://doi.org/10.1038/nrg3428>.
- Suzuki, T., Yuasa, H., Imai, K., 1996. Convergent evolution. The gene structure of Sulculus 41 kDa myoglobin is homologous with that of human indoleamine dioxygenase. *Biochimica et Biophysica Acta* 1308, 41–48.
- Toll-Riera, M., Laurie, S., Radó-Trilla, N., Alba, M.M., 2011. Partial Gene Duplication and the Formation of Novel Genes. *IntechOpen*. Available at: <https://doi.org/10.5772/21846>.
- Tutar, Y., 2012. Pseudogenes [WWW Document]. *International Journal of Genomics* 2012, 4. Available at: <https://doi.org/10.1155/2012/424526>.
- Vogel, C., Teichmann, S.A., Pereira-Leal, J., 2005. The relationship between domain duplication and recombination. *Journal of Molecular Biology* 346, 355–365. Available at: <https://doi.org/10.1016/j.jmb.2004.11.050>.
- Wang, Y., Liu, J., Huang, B., *et al.*, 2015. Mechanism of alternative splicing and its regulation (Review). *Biomedical Reports* 3, 152–158.
- Wang, T.-T., Si, F.-L., He, Z.-B., Chen, B., 2018. Genome-wide identification, characterization and classification of ionotropic glutamate receptor genes (iGluRs) in the malaria vector *Anopheles sinensis* (Diptera: Culicidae). *Parasites & Vectors* 11 (1), 34. Available at: <https://doi.org/10.1186/s13071-017-2610-x>.
- Wang, X., Wheeler, D., Avery, A., *et al.*, 2013. Function and evolution of DNA methylation in *Nasonia vitripennis*. *PLOS Genetics* 9, e1003872. Available at: <https://doi.org/10.1371/journal.pgen.1003872>.
- White, M.J.D., 1977. *Animal Cytology and Evolution*. CUP Archive.
- Yoon, S., Xuan, Z., Makarov, V., Ye, K., Sebat, J., 2009. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Research* 19, 1586–1592. Available at: <https://doi.org/10.1101/gr.092981.109>.
- Yu, P., Ma, D., Xu, M., 2005. Nested genes in the human genome. *Genomics* 86, 414–422. Available at: <https://doi.org/10.1016/j.ygeno.2005.06.008>.
- Zhou, X., Lin, Z., Ma, H., 2010. Phylogenetic detection of numerous gene duplications shared by animals, fungi and plants. *Genome Biology* 11, R38. Available at: <https://doi.org/10.1186/gb-2010-11-4-r38>.

Epidemiology: A Review

Vijayaraghava S Sundararajan, Ministry of Environmental Agency, Singapore; Bioclues.org, Hyderabad, India and Bioclues.org, Hyderabad, India

© 2019 Elsevier Inc. All rights reserved.

Introduction

Those living in the tropics and subtropical regions are at higher risk of dengue fever and dengue hemorrhagic fever and epidemics occur unexpectedly and can cause a great concern to the healthcare authorities. Climate, demography and ecology, urbanization are vital reasons in dengue transmission and epidemiology.

Dengue virus, an arbovirus that is transmitted by *Aedes* mosquitos in the tropics and subtropics of the world, causes an estimated 390 million infections per year, resulting in 96 million clinically symptomatic cases (Bhatt *et al.*, 2013a). DENV has four serotypes (DENV-1, DENV-2, DENV-3, and DENV-4) that each circulate worldwide. Spatial propagation of long-distance spread over large geographical distances is difficult to measure directly, but epidemiological coupling of locations revealed by synchrony in population-level disease patterns has been used successfully to infer mechanisms of spread (Grenfell *et al.*, 2001; Viboud *et al.*, 2006; Rohani *et al.*, 1999).

Dengue virus is transmitted mainly by *Aedes aegypti* and *Aedes albopictus* mosquitoes (Rosen *et al.*, 1983). World Health Organization has estimated that there are 50–100 million dengue infections per year (Rigau-Pérez *et al.*, 1998) and a recent study estimates that this figure to 390 million, of which ~96 million are symptomatic (Bhatt *et al.*, 2013b). Dengue infection in humans is mostly self-limiting although antiviral drugs are under development (Lim *et al.*, 2013; Rathore *et al.*, 2011). Some dengue cases may require hospital admission, and the more severe manifestations of dengue may lead to death (Murphy and Whitehead, 2011) and case fatality rates of dengue fever and severe dengue vary from 0%–5% to 3%–5% (Halstead, 1999).

DENV is a positive sense single stranded RNA virus of the genus *Flavivirus*, genome of approximately 11.8 kb encodes for a single polypeptide arranged in three structural (Capsid-pre-Membrane-Envelope) and seven non-structural (NS) proteins (NS1-NS2A-NS2B-NS3-NS4A-NS4B-NS5). The exposure to a particular serotype of DENV elicits type-specific life-long immunity (Simmons *et al.*, 2012) and display high mutation rates as a result of the error-prone RNA polymerase activity during genome replication (Steinhauer *et al.*, 1992). Genetic recombination further facilitates their opportunity for genetic novelty (Behura and Severson, 2013) and results in genetically diverse virus populations consisting of multiple genotypes of monophyletic nature. RNA virus populations often appear to exist as ensembles of mutant spectra that collectively behave as “quasi-species” (Domingo *et al.*, 2012) and serotype is composed of highly heterogeneous genotypes that demonstrates geographical-wide pattern spread (Holmes and Twiddy, 2003).

Dengue fever replaced malaria as the most important mosquito-borne disease in Singapore in the mid-1960s and attention is on vector source reduction through surveillance, enforcement, community engagement, careful urban planning and operational research. *Aedes* house index was successfully reduced to 50% in 1960s and further to a less than 5% by the late 1970s and dengue incidence reduced (Chan *et al.*, 1977; Goh, 1997, 1995; Chan *et al.*, 1971). However, dengue fever started to resurge in the country in 1980s, in a typical 5–6 year epidemic cycle and epidemics escalated within each cycle during the last decade even though house index was below 1% (Koh *et al.*, 2008; Lee *et al.*, 2010; Ler *et al.*, 2011). The geo-expansion of *A. aegypti* and resurgence is due to an increase in human population density, low herd immunity resulting from long periods of low transmission, local transportation that facilitates virus dissemination, virus importations and the (Ang *et al.*, 2015; Yew *et al.*, 2009; Goh and Yamazaki, 1987; Low *et al.*, 2015).

Research Studies

The entomological surveillance show caused an increase in both larval and adult *Aedes aegypti* (main source of vector transmission) population during the epidemic season. The ground conditions are conducive to sustain virus transmission on a longer period causing uncontrollable epidemic of dengue.

Region-Wide Synchrony and Traveling Waves of Dengue Across Eight Countries in Southeast Asia (Panhuis *et al.*, 2015)

The spatial and temporal dynamics of dengue transmission are poorly understood, limiting disease control efforts. A study with a large-scale dataset and analysed continental-scale patterns of dengue in Southeast Asia, travel pattern of dengue was studied and reported.

Dengue Transmission: Strong synchrony over the region for dengue cycles is reported. Statistically significant multiannual cycles changed over time, the highest in 1993–2004. In comparison, the average power of annual cycles was more constant over time but reduced in 1997–2001. The regional median for annual cycles was consistent over time, ranging between 0.50 and 0.64. The average synchrony decreased as the distance between province pairs increased up to ~1000 km. Synchrony of multiannual cycles peaked at 0.55 [95% confidence interval (95% CI)=0.45–0.67] compared with annual cycles.

Climate: The dengue cycles in 1997–2001 coincided with high temperatures across most latitude; but not with an anomaly in precipitation. High temperatures in these years were related to the strongest El Niño episode of the past century (Slingo and Annamalai, 2000; Webster and Palmer, 1997). We measured a strong wavelet coherency (>0.8) between multiannual dengue cycles and the Oceanic Niño Index (ONI) during 1993–2002 and during 2009–2010 for almost all provinces.

Travel Waves: Epidemic timings between provinces were measured and found, positive θ indicated that a province was timed earlier (leading ahead) vs. the other, and a negative θ indicated that a province was timed later (lagging behind). Provinces with outgoing traveling waves were located in west Thailand and the Bangkok area, central Laos (Savannakhét and Khammouan), and southern Philippines (Bohol).

Data and methods: Monthly dengue surveillance data and corresponding population (U.S. Census Bureau, 2014) and climate data (Fan and van den Dool, 2008; Becker *et al.*, 2013; Center for International Earth Science Information Network CIESIN, Columbia University, Centro Internacional de Agricultura Tropical (CIAT), 2005) at the provincial level (GADM, 2013) were available for 273 provinces in Thailand, Cambodia, Laos, Vietnam, Malaysia, Singapore, the Philippines, and Taiwan. Province names were standardized based on the administrative levels. Wavelet analysis which is well suited to characterize epidemiological time series is used isolate cycles with multiannual and annual periodicities. Synchrony between provinces was derived as the pairwise Pearson correlation coefficient with timing and amplitude of signals. Wavelet Coherency uses wave transforms of two time series to indicate their localized phase relationship in a time–frequency spectrum. Phase angles for annual and multiannual cycles as indicators of epidemic timing were computed. Traveling waves for a province is computed as a statistically significant positive linear association between θ and geographical distances between that province and the other provinces. Sensitivity analysis is conducted on the effect of the value for the wavelet parameters ω_0 and δj on synchrony.

The multi-country collaborative study improved insight that may lead to improved prediction of dengue transmission patterns and more effective disease surveillance and control efforts.

Three-Month Real-Time Dengue Forecast Models: An Early Warning System for Outbreak Alerts and Policy Decision Support in Singapore (Shi *et al.*, 2016)

Statistical models are built through machine learning methods in deriving potential recurrent infectious dengue outbreaks. In this study models are built with data that are available at the time of forecast, to forecast weeks or months, to get predictive performance, and be able to process new data rapidly. Weekly covariates were used to match the frequency of reported dengue data (Ministry of Health), time horizon used for all variables was January 2001 to December 2012.

Case data is downloaded as the weekly number of cases (natural log–transformed, + 1) from Ministry of Health, Singapore. *Population data* (mid-year) is retrieved from Singapore Department of Statistics for residents and foreign non-residents (natural log–transformed). Weekly mean temperature (T) in degrees Celsius, maximum hourly temperature, number of hours of high temperature ($>27.8^\circ\text{C}$) each week, and weekly relative humidity (RH) were obtained from Meteorological Services, Singapore. *Vector surveillance* weekly breeding percentage (BP) is an in-house index developed by NEA that provides an estimate of the proportion of *Ae. aegypti*, the primary vector of dengue in Singapore is calculated for the study. *Trend and seasonality data* with climatic factors (dengue is affected by other factors such as changes to vector control and circulating serotypes) and non-climatic factors (disease dynamics, dengue incidence into terms for trend and for annual seasonality) are derived.

The Least Absolute Shrinkage and Selection Operator (LASSO) is a technique that has inspired much interest in the statistical methodology community on “small n large p ” problems (Tibshirani, 1996). This framework use logistic regression selecting parameters to be included in the model rather than optimizing the (log) likelihood $L(y|\beta, x)$ for dependent variable y , independent variables x and coefficients β , as in standard regression. Covariates were considered at lags of up to 20 weeks based on the findings (Hii *et al.*, 2012a,b), but in contrast to their approach, we allowed the effect of a single factor (such as temperature) to have multiple lags in influencing future dengue cases.

Result: 12-week forecasts including 95% prediction intervals for various time points over the period 2001–2002 are reported. Also the forecasts at various time points in Singapore’s record-breaking 2013 epidemic, in which 22,170 cases were reported, are presented.

Epidemiology studies are challenging as local weather conditions within a region differs significantly. Integrating weather data into predictive models may not be always feasible as long term data not available.

Intra-Epidemic Evolutionary Dynamics of a Dengue Virus Type 1 Population Reveal Mutant Spectra That Correlate With Disease Transmission (Hapuarachchi *et al.*, 2016a)

Environmental Health Institute (EHI) received blood samples obtained from dengue suspected patients through an island wide network of hospitals and general practitioners during November 2012 to May 2014 for the DENV infection confirmed by SD Bioline Dengue Duo kit. All NS1 positive samples were subjected to serotyping. All consented DENV positive samples were used for genome sequencing as part of the existing virus surveillance programme.

Data: Virus strains were isolated from acute-phase sera (Koo *et al.*, 2013) and a maximum of three passages was done for each sample based on the original virus load. Viral RNA was extracted from sera using the QIAGEN viral RNA mini kit. The complementary DNA (cDNA) was synthesized using extracted RNA (Koo *et al.*, 2013). DENV serotypes were determined by using a

real-time PCR assay (Koo *et al.*, 2013; Lai *et al.*, 2007). Whole genome and envelope (E) gene sequences were generated directly from patient sera, whereas whole genome sequencing was performed on strains isolated from respective patient sera. Raw nucleotide sequences were assembled using the Laser gene package. Contiguous sequences were aligned in Bio-Edit (Hall, 1999).

Methods: The whole-polyprotein based maximum clade credibility tree was constructed using the Bayesian Markov Chain Monte Carlo (MCMC) method implemented in the BEAST (Drummond *et al.*, 2012). Based on jModeltest (Darriba *et al.*, 2012), the general time reversible model with gamma distribution and invariant sites (GTR+G+I) were used together with an uncorrelated log-normal relaxed molecular clock model. The dataset included 38 whole polyprotein sequences generated in this study and 44 DENV-1 sequences retrieved from GenBank database. Of 83 whole genomes generated, only 38 sequences representative of each variant were included in the phylogenetic analyses to minimize overcrowding.

Protein Data Bank (PDB) structures 3J2P (Zhang *et al.*, 2013) which illustrates the M and E-protein heterotetramer, 2JLV (Luo *et al.*, 2008) which represents the NS3 protein (serine protease as well as RNA helicase and RTPase/NTPase) and 4C11 showing the NS5 protein were downloaded from the Research Collaborator for Structural Bioinformatics (RCSB) protein data bank and visualized using YASARA (Krieger *et al.*, 2002). Changes to protein stability were estimated using Fold-X (Schymkowitz *et al.*, 2005).

Statistical analysis: The correlation between the diversity of epidemic lineage and transmission intensity, are compared with the number of DENV-1 genotype III variants against the total number of DENV-1 cases and total number of reported cases on a monthly basis (November 2012 to May 2014). The strength of correlation was quantified using Pearson's correlation coefficient in 'R' package.

Results: DENV-1 genotype III strains of the epidemic lineage were first detected locally in November 2012. The genetic distinction of the epidemic lineage from the locally circulating DENV-1 strains indicated that its emergence in Singapore was likely to be due to an introduction. The resulting mutant diversity intensified during periods of high transmission, implying a relationship between the force of *in-situ* evolution and transmission intensity during the epidemic. Tracking genetic diversity through time is a useful tool to infer DENV transmission dynamics and to complement surveillance data in a risk assessment and cross-sectional surveys and longitudinal virus surveillance efforts with limited sample sizes tend to underestimate the true diversity of viral populations.

Epidemic Resurgence of Dengue Fever in Singapore in 2013–2014: A Virological and Entomological Perspective (Hapuarachchi *et al.*, 2016b)

Case burden of the dengue epidemic in 2013 and 2014 was unprecedented, recording 1.7 times more cases than those reported during the last DENV-1 epidemic during 2004–05. The epidemic is due to demographic, social, virological, entomological, immunological, climatic and ecological factors that contribute to DENV transmission.

Epidemic: The control of vector-borne dengue in Singapore is carried out by National Environmental Agency which manages the day-to-day vector control and Clinical management and surveillance is overseen by Ministry of Health (Ng and Vythilingum, 2011). NEA and MOH jointly perform the below activities to manage the epidemic crisis at Singapore:

1. Enhanced case surveillance measures to increase the diagnostic coverage at the very early phase of the epidemic. MOH initiates high clinical suspicion of dengue among febrile cases and existing network of general practitioners was encouraged to utilize a subsidized laboratory testing service provided by EHI, one of the public health laboratories under NEA, Singapore.
2. Projection of case numbers to facilitate stockpiling of diagnostic reagents in advance to tackle a sudden endemic of dengue. Statistical models are used on weekly case numbers (Shi *et al.*, 2016) arrangements were made to extend EHI diagnostic services during weekends and to stockpile diagnostic reagents several months ahead of the peak of the epidemic. The close communication between MOH and NEA on the weekly case projection also allowed hospitals to plan for increased demand on the healthcare system resulting from extra consultations and admissions.
3. Expansion of virus surveillance was extended to include samples from polyclinics, hospitals and private laboratories through a joint initiative between NEA and MOH. Even during non-epidemic periods, joint efforts are thrown to screen a small proportion of reported cases to have regular surveillances. Serotype analysis was achieved in 33% of reported cases and genotype coverage was doubled during 2013–14. Genetic fingerprinting of virus strains allowed monitoring of their spread between locations and facilitated targeted vector control in newly-introduced sites.
4. Early launch of the dengue campaign was a key success to promote community awareness on dengue. NEA launched "Do the Mozzie Wipe-out" drive at clustered housing estates b street banners and social media. A tri-colour coded banner (Green: no dengue; Yellow: cluster with less than 10 cases; Red: cluster with more than 10 cases) was introduced to make public to have an immediate knowledge on their sector. Dengue volunteers are trained to support community outreach through seminars, talks, roadshows and media.
5. Enhanced source reduction for mosquito breeding is achieved by increased premises inspections, entomological surveillance, enforcement activities, and daily checks on common ground areas, public areas and congregation areas for potential mosquito breeding spots. Chemical larviciding activities were carried out in housing estates, industrial premises and public places on a regular basis and whenever *Aedes* breeding was detected through inspections. The public was encouraged to use pellets containing *Bacillus thuringiensis* in places such as roof gutters. Fogging activities were conducted in major clusters during the peak periods of transmission to reduce adult *Aedes* population density especially when the intensity of transmission sustained

despite other source reduction efforts. Data on *Aedes* immatures collected through field inspections was used to gauge *Aedes* population density in a particular area.

6. Launch of gravidtrap is a greater milestone in detecting and monitoring adult *Aedes* population and viral carriers in any area. The breeding data collected on a regular basis from sentinel locations and gravidtrap and are matched with cluster locations.
7. Spatial risk map is drawn based on data pertaining to *Aedes* breeding, past dengue exposure, population density, and circulation virus strains and is used as a guide for resource allocation during epidemic.
8. Integrated vector management activities in Singapore are closely aligned with the whole-of-government's effort in establishing people, public and private (3P) partnership to develop innovative and sustainable initiatives to promote environmental ownership in the community. NEA leads an Inter-Agency Dengue Task Force (Koh *et al.*, 2008) comprising 27 stakeholders from the 3P sectors to coordinate nationwide dengue control efforts. The Inter-Agency Dengue Task Force assisted the epidemic control by ensuring the planning and implementation of operational activities of each agency to align with source reduction and vector control efforts carried out by NEA.

The integrated surveillance framework is supported by Improved operational response; Early warning of outbreaks based on virus surveillance; Understanding the distribution of vectors and their density fluctuations through entomological surveillance; Understanding the relationship between environmental parameters and outbreak risk.

Discussion

The spread and prevalence of dengue and dengue fever are better understood through region-wide synchrony, travelling waves, dengue forecast models, evolutionary studies, and epidemic resurgence.

In this short review, we have briefed studies that were conducted in and around Singapore. These countries (Thailand, Singapore, Malaysia) notice a huge human traffic cross bordering each and every day. The patterns from these studies caution authorities to monitor the spread across these countries in this region. An epidemiological study to understand the virus resurgence can't be successfully performed without inherent data from nearby countries and disease population data. A multi-level approach involving epidemiological, virological and entomological is required in a combined population across nations.

Acknowledgments

We thank Elsevier for giving an opportunity to write this article. We sincerely thank Dr Asif Khan for the invitation to write a book article.

See also: Ecosystem Monitoring Through Predictive Modeling. Genetics and Population Analysis. Large Scale Ecological Modeling With Viruses: A Review. Mapping the Environmental Microbiome. Natural Language Processing Approaches in Bioinformatics. Sequence Analysis

References

- Ang, L.W., Cutter, J., James, L., Goh, K.T., 2015. Sero-epidemiology of dengue virus infection in the adult population in tropical Singapore. *Epidemiol. Infect.* 143, 1585–1593.
- Becker, A., *et al.*, 2013. A description of the global land-surface precipitation data products of the Global Precipitation Climatology Centre with sample applications including centennial (trend) analysis from 1901–present. *Earth Syst. Sci. Data* 5 (1), 71–99.
- Behura, S.K., Severson, D.W., 2013. Nucleotide substitutions in dengue virus serotypes from Asian and American countries: Insights into intra codon recombination and purifying selection. *BMC Microbiol.* 13, 37.
- Bhatt, S., *et al.*, 2013a. The global distribution and burden of dengue. *Nature* 496 (7446), 504–507.
- Bhatt, S., Gething, P.W., Brady, O.J., *et al.*, 2013b. The global distribution and burden of dengue. *Nature* 496, 504–507.
- Center for International Earth Science Information Network (CIESIN), Columbia University; Centro Internacional de Agricultura Tropical (CIAT), 2005. Gridded Population of the World Version 3 (GPWv3), NASA Socioeconomic Data and Applications Center (SEDAC), Palisades, NY.
- Chan, K.L., Ng, S.K., Chew, L.M., 1977. The 1973 dengue haemorrhagic fever outbreak in Singapore and its control. *Singap. Med. J.* 18, 81–93.
- Chan, Y.C., Ho, B.C., Chan, K.L., 1971. *Aedes aegypti* (L.) and *Aedes albopictus* (Skuse) in Singapore City. 5. Observations in relation to dengue haemorrhagic fever. *Bull. World Health Organ.* 44, 651–657.
- Darriba, D., Taboada, G.L., Doallo, R., Posada, D., 2012. jModelTest 2: More models, new heuristics and parallel computing. *Nat. Methods* 9, 772.
- Domingo, E., Sheldon, J., Perales, C., 2012. Viral quasi-species evolution. *Microbiol. Mol. Biol. Rev.* 76, 159–216.
- Drummond, A.J., Suchard, M.A., Xie, D., Rambaut, A., 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Mol. Biol. Evol.* 29, 1969–1973.
- Fan, Y., van den Dool, H., 2008. A global monthly land surface air temperature analysis for 1948–present. *J. Geophys. Res.* 113 (D1), D01103.
- GADM, 2013. Database of global administrative areas. gadm.org/home (accessed 21.07.15).
- Goh, K.T., 1995. Changing epidemiology of dengue in Singapore. *Lancet* 346, 1098.
- Goh, K.T., 1997. Dengue – A re-emerging infectious disease in Singapore. *Ann. Acad. Med. Singap.* 26, 664–670.
- Goh, K.T., Yamazaki, S., 1987. Serological survey on dengue virus infection in Singapore. *Trans. R. Soc. Trop. Med. Hyg.* 81, 687–689.
- Grenfell, B.T., Bjørnstad, O.N., Kappey, J., 2001. Travelling waves and spatial hierarchies in measles epidemics. *Nature* 414 (6865), 716–723.
- Hall, T.A., 1999. BioEdit: A user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. *Nucleic Acid Symp. Ser.* 41, 95–98.
- Halstead, S.B., 1999. Is there an apparent dengue explosion? *Lancet* 353, 1100–1101.

- Hapuarachchi, H.C., Koo, C., Kek, R., *et al.*, 2016a. Intra-epidemic evolutionary dynamics of a Dengue virus type 1 population reveal mutant spectra that correlate with disease transmission. *Sci. Rep.* 6, 22592.
- Hapuarachchi, H.C., Koo, C., Rajarethinam, J., *et al.*, 2016b. Epidemic resurgence of dengue fever in Singapore in 2013–2014: A virological and entomological perspective. *BMC Infect. Dis.* 16, 300.
- Hii, Y.L., Rocklöv, J., Wall, S., *et al.*, 2012a. Optimal lead time for dengue forecast. *PLOS Negl. Trop. Dis.* 6, e1848.
- Hii, Y.L., Zhu, H., Ng, N., Ng, L.C., Rocklöv, J., 2012b. Forecast of dengue incidence using temperature and rainfall. *PLOS Negl. Trop. Dis.* 6, e1908.
- Holmes, E.C., Twiddy, S.S., 2003. The origin, emergence and evolutionary genetics of dengue virus. *Infect. Genet. Evol.* 3, 19–28.
- Koh, B.K., Ng, L.C., Kita, Y., *et al.*, 2008. The 2005 dengue epidemic in Singapore: Epidemiology, prevention and control. *Ann. Acad. Med. Singap.* 37, 538–545.
- Koo, C., *et al.*, 2013. Evolution and heterogeneity of multiple serotypes of Dengue virus in Pakistan, 2006–2011. *Viol. J.* 10, 275.
- Krieger, E., Koraimann, G., Vriend, G., 2002. Increasing the precision of comparative models with YASARA NOVA – A self-parameterizing force field. *Proteins* 47, 393–402.
- Lai, Y.L., *et al.*, 2007. Cost-effective real-time reverse transcriptase PCR (RT-PCR) to screen for Dengue virus followed by rapid single-tube multiplex RT-PCR for serotyping of the virus. *J. Clin. Microbiol.* 45, 935–941.
- Lee, K.S., Lai, Y.L., Lo, S., *et al.*, 2010. Dengue virus surveillance for early warning, Singapore. *Emerg. Infect. Dis.* 16 (5), 847–849.
- Ler, T.S., Ang, L.W., Yap, G.S., *et al.*, 2011. Epidemiological characteristics of the 2005 and 2007 dengue epidemics in Singapore – Similarities and distinctions. *Western Pac. Surveill. Response J.* 2, 24–29.
- Lim, S.P., Wang, Q.Y., Noble, C.G., *et al.*, 2013. Ten years of dengue drug discovery: Progress and prospects. *Antiviral Res.* 100, 500–519.
- Low, S.L., Lam, S., Wong, W.Y., *et al.*, 2015. Dengue seroprevalence of healthy adults in Singapore: Serosurvey among blood donors, 2009. *Am. J. Trop. Med. Hyg.* 93, 40–45.
- Luo, D., *et al.*, 2008. Insights into RNA unwinding and ATP hydrolysis by the flavivirus NS3 protein. *EMBO J.* 27, 3209–3219.
- Murphy, B.R., Whitehead, S.S., 2011. Immune response to dengue virus and prospects for a vaccine. *Annu. Rev. Immunol.* 29, 587–619.
- Ng, L.C., Vythilingum, I. (Eds.), 2011. *Vectors of Flaviviruses and Strategies for Control*, in *Molecular Virology and Control of Flaviviruses*. Poole, United Kingdom: Caister Academic Press.
- Panhuys, W.G., Choisy, M., Xiong, X., *et al.*, 2015. Region-wide synchrony and traveling waves of dengue across eight countries in Southeast Asia. *Proc. Natl. Acad. Sci. USA* 112 (42), 13069–13074.
- Rathore, A.P.S., Paradkar, P.N., Watanabe, S., *et al.*, 2011. Celgosivir treatment misfolds dengue virus NS1 protein, induces cellular pro-survival genes and protects against lethal challenge mouse model. *Antiviral Res.* 92, 453–460.
- Rigau-Pérez, J.G., Clark, G.G., Gubler, D.J., *et al.*, 1998. Dengue and dengue haemorrhagic fever. *Lancet* 352, 971–977.
- Rohani, P., Earn, D.J., Grenfell, B.T., 1999. Opposite patterns of synchrony in sympatric disease metapopulations. *Science* 286 (5441), 968–971.
- Rosen, L., Shroyer, D.A., Tesh, R.B., Freier, J.E., Lien, J.C., 1983. Transovarial transmission of dengue viruses by mosquitoes: *Aedes albopictus* and *Aedes aegypti*. *Am. J. Trop. Med. Hyg.* 32, 1108–1119.
- Schymkowitz, J.W., *et al.*, 2005. Prediction of water and metal binding sites and their affinities by using the Fold-X force field. *Proc. Natl. Acad. Sci. USA* 102, 10147–10152.
- Shi, Y., Liu, X., Kok, S.-Y., *et al.*, 2016. Three-month real-time dengue forecast models: An early warning system for outbreak alerts and policy decision support in Singapore. *Environ. Health Perspect.* 124 (9), 1369–1375.
- Simmons, C.P., Farrar, J.J., Nguyen, V.V., Wills, B., 2012. Dengue. *N. Engl. J. Med.* 366, 1423–1432.
- Slingo, J.M., Annamalai, H., 2000. 1997: The El Niño of the century and the response of the Indian summer monsoon. *Mon. Weather Rev.* 128 (6), 1778–1797.
- Steinhauer, D.A., Domingo, E., Holland, J.J., 1992. Lack of evidence for proofreading mechanisms associated with an RNA virus polymerase. *Gene* 122, 281–288.
- Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58, 267–288.
- U.S. Census Bureau, 2014. International database.
- Viboud, C., *et al.*, 2006. Synchrony, waves, and spatial hierarchies in the spread of influenza. *Science* 312 (5772), 447–451.
- Webster, P.J., Palmer, T.N., 1997. The past and the future of El Niño. *Nature* 390 (12), 562–564.
- Yew, Y.W., Ye, T., Ang, L.W., *et al.*, 2009. Seroepidemiology of dengue virus infection among adults in Singapore. *Ann. Acad. Med. Singap.* 38, 667–675.
- Zhang, X., *et al.*, 2013. Cryo-EM structure of the mature dengue virus at 3.5-Å resolution. *Nat. Struct. Mol. Biol.* 20, 105–110.

Biographical Sketch



Dr. Vijayaraghava Seshadri Sundararajan is an advisor of Bioclues.org who is organizing international conferences/workshops.

Identification of Homologs

William R Pearson, University of Virginia School of Medicine, Charlottesville, VA, United States

© 2019 Elsevier Inc. All rights reserved.

Background/Fundamentals

Modern genome biology relies on the inference of homology from protein and DNA sequence similarity searching. As the number of sequenced genomes has grown from the dozens to the thousands and tens of thousands today, except for a small number of experimental model organisms, most functional annotations on newly sequenced genomes are transferred based on similarity searches. This section examines the philosophical and statistical basis for inferring homology from excess similarity.

The inference of homology – common evolutionary ancestry – is a conclusion about relationships based on biological events that often occurred from 0.5 to 3 billion years ago. For example, based on excess protein sequence similarity, about 40% of human enzymes have easily identified homologs in *E. coli* (Fig. 1). This observation of excess similarity implies that large fraction of human and *E. coli* proteins existed more than 3 billion years ago in the organism that was the last common ancestor of mammals and bacteria. And 40% is a very conservative estimate; it is likely that a much larger fraction of human enzymes have homologs in *E. coli*, but 40% are easily detected in a BLASTP search.

The ability to identify such distant homologs was unexpected. In the late 1960s, as the first protein sequences became available, one of the first uses of “homology” to denote excess similarity (Winter *et al.*, 1968) explicitly denied that the traditional evolutionary meaning of the word (common ancestry) was appropriate in the protein context, suggesting that excess similarity might be the result of very similar protein sequences emerging independently, analogous to the emergence of the ability to fly in insects and vertebrates. But this argument was short-lived, as early researchers came to understand that the space of possible protein sequences is astronomically large (McLachlan, 1971; Altschul *et al.*, 1994). Thus, faced with the two logical alternatives for excess similarity – common ancestry or independent origins – the more parsimonious explanation, in the presence of excess similarity, is common ancestry.

The “homology:excess similarity” relationship is also confused by the use of the term “homology” in place of “identity,” as in the phrase “these sequences are 30% homologous.” Because homology is a qualitative inference, two proteins are either homologous, or they are not homologs (Reeck *et al.*, 1987). Homology and identity are not interchangeable; homology is a *qualitative* statement about evolutionary ancestry, while identity is a *quantitative* statement about sequence similarity. While sequence identity can be a

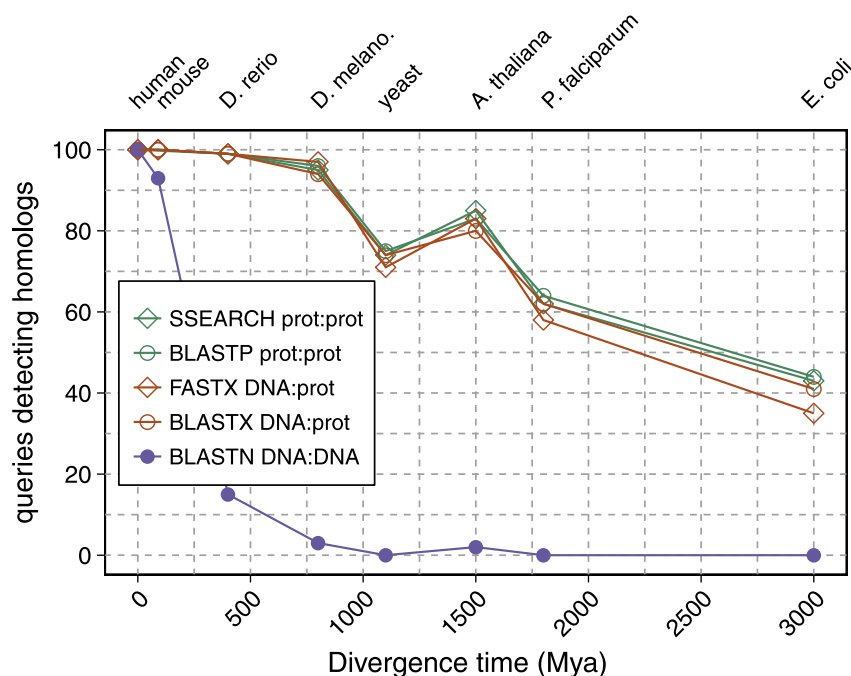


Fig. 1 Sensitivity of protein and DNA sequence comparison – One hundred human RefSeq proteins and their corresponding mRNA sequences from proteins with E.C. numbers (enzymes) were compared to proteomes and mRNA sequence sets from the indicated genomes. Plotted are the number of queries that had at least one homolog with enough excess similarity that the score would be expected in fewer than 1000 searches by chance ($E < 0.001$) in searches against the indicated proteomes/RNA-sets using SSEARCH and BLASTP for protein:protein comparisons, FASTX and BLASTX for DNA:protein comparisons, and BLASTN for DNA:DNA comparisons. Divergence times from timetree.org.

reliable (though insensitive) measure of excess similarity, and identity is a useful indicator of evolutionary distance (homologous sequences that are 90% identical are much more closely related than sequences that are 30% identical), identity (or similarity) and homology are fundamentally different concepts. Amino-acid residues can be homologous (share a common evolutionary history) without being identical, and residues may be identical simply by chance (thus not sharing common ancestry).

Homology inference is often based on excess pair-wise sequence similarity found in a similarity search, using a program like BLASTP (Altschul *et al.*, 1990; Altschul *et al.*, 1997; Camacho *et al.*, 2009). Pair-wise sequence similarity is effective because modern sequence databases are comprehensive – complete proteomes are known for thousands of organisms. However, pair-wise sequence comparison is less sensitive than comparison based on excess structural similarity (Pearson and Sierk, 2005). Excess structural similarity is considered the “gold standard” for inferring homology, but the argument for homology from excess similarity is the same. If two structures share much more similarity than is expected by chance, the simplest explanation is that the excess similarity reflects common ancestry. Methods to improve sequence-based homology inference often rely on homology inferences based on excess structural similarity. While structure comparison is far more sensitive than pair-wise sequence comparison, only a fraction of protein sequences are homologous to proteins with known structures. Because there are so many complete proteomes, sequence comparison can be used to identify homologs for almost every protein in a newly sequenced genome.

Intermediate in sensitivity between pair-wise sequence comparison and pair-wise structure comparison are sequence based methods that use sequence models, using either Position-Specific Scoring Matrices (PSSMs, Altschul *et al.*, 1997; Schaffer *et al.*, 2001; Altschul and Koonin, 1998), or Hidden Markov Models (HMMs, Sonnhammer *et al.*, 1998; Finn *et al.*, 2011; Eddy, 2011). Model-based methods (PSSMs, HMMs) can be 5–10-fold more sensitive than pair-wise alignment (Pearson *et al.*, 2017). Model-based methods provide an automated approach to transitive similarity searching – if A shares significant similarity with B, and B with C, then A can be inferred to be homologous to C without sharing significant pair-wise similarity (Gerstein, 1998).

Application

The inference of homology from excess similarity requires both (1) algorithms to align sequences and produce similarity scores and (2) an accurate model of how much similarity is expected by chance. The first algorithm used to align protein sequences (Needleman and Wunsch, 1970) produced a *global* sequence alignment – sequences were aligned from end-to-end – and accommodated a variety of match/mismatch penalties, from simple identities to more sophisticated measures of amino-acid similarity. The Needleman-Wunsch algorithm used a penalty-*per-gap* strategy; a gap of one residue, or two, or five would all have the same cost.

Today, most similarity searching programs, including BLAST, calculate a local alignment score – one in which the alignment can begin and end anywhere in the sequences as long as it produces a maximum score. Local alignments can identify homologous domains in different sequence contexts, and work well with partial protein sequences produced by exons in genome assemblies. The first rigorous local alignment algorithm was described by Smith and Waterman (1981). Modern algorithms use a more efficient form described by Gotoh (1982), vectorized versions of the rigorous Smith-Waterman-Gotoh local alignment algorithm (Farrar, 2007) can speed searches 10–20-fold.

In addition to an algorithm to calculate alignment scores, homology inference requires an understanding of alignment score statistics. To recognize *excess* similarity, it is critical to understand how much similarity is expected by chance (between random or unrelated sequences), particularly in the context of a database search, where tens of thousands to tens of millions of sequences are compared. Early protein sequence analysts understood the importance of evaluating statistical significance using unrelated (typically shuffled) sequences (McLachlan, 1971), but without a statistical model of the distribution of unrelated scores, it was impossible to accurately estimate probabilities. The recognition that local similarity scores between random protein or DNA sequences are accurately described by the extreme-value distribution (Arratia *et al.*, 1986; Karlin and Altschul, 1990), even for gapped alignments (Mott and Tribe, 1999), allows excess similarity to be measured and thus homology to be identified routinely. These statistical foundations enabled the BLAST programs (Altschul *et al.*, 1990, 1997) to provide the first accurate statistical estimates for similarity scores. Today, all widely-used similarity search programs provide accurate statistical estimates (Brenner *et al.*, 1998; Pearson and Sierk, 2005).

The most widely used sequence comparison algorithm, BLAST (Altschul *et al.*, 1990, 1997; Camacho *et al.*, 2009), uses the Karlin and Altschul statistical model (Karlin and Altschul, 1990) to set heuristic thresholds for the inclusion of *k*-letter amino-acid “words” (typically *k*=3 for proteins) to dramatically accelerate sequence comparison by not examining alignments that are unlikely to reflect homology. Since homologous sequences are rarely more than 1.0% of a comprehensive sequence database (in the human proteome, only 6 of 6116 Pfam domain families are present in more than 1.0% of proteins, with the most abundant domains, protein kinases and Zn-fingers, appearing in 2% of proteins), focusing on sequences that could produce significant alignments can speed up similarity searches 100-fold.

Local sequence alignment scores between random sequences are well described by the extreme value distribution:

$$p(S_{\text{raw}}) \leq 1 - \exp(-Kmn e^{-\lambda S_{\text{raw}}}) \quad (1)$$

where S_{raw} is the calculated alignment score, m, n are the lengths of the two sequences being aligned, λ is a parameter that captures the “scaling” of the scoring matrix used to produce the alignment score, and K is a parameter that captures how the similarity scores of random sequence alignments increase with sequence length. For alignment scores with $p < 0.1$, this equation can be simplified to:

$$p(S_{\text{raw}}) \leq K m n e^{-\lambda S_{\text{raw}}} \quad (2)$$

which can also be expressed in terms of a “bit” score as

$$p(S_{\text{bits}}) \leq m n 2^{-S_{\text{bits}}} \quad (3)$$

where $S_{\text{bits}} = (\lambda S_{\text{raw}} - \ln(K)) / \ln(2)$

The “bit” score has the advantage that Eq. (3) can be used to calculate the probability of any S_{bit} score, regardless of the scoring matrix, and “bit” scores are available for most similarity searching programs (BLASTP, [Camacho et al., 2009](#); FASTA, [Pearson, 2016](#)); and HMMER3, [Eddy, 2011](#)).

The $p(S)$ value calculated in Eqs. (1)–(3) reflects the probability of obtaining a score $\geq S$, in a *single* sequence alignment between two unrelated proteins. Because it assumes that only one alignment score has been calculated, $p(S)$ is not the appropriate measure of statistical significance in a similarity search of a protein or DNA database, where hundreds of thousands to tens of millions of sequences have been examined. To account for the thousands to millions of multiple tests (alignments) that were considered in a search, the p -value is converted to an expectation value using a Bonferroni correction:

$$E(S) \leq p(S) D \quad (4)$$

where D is the number of alignments performed. Equivalently:

$$E(S_{\text{bit}}) \leq D m n 2^{-S_{\text{bit}}} \quad (5)$$

or, for BLAST where the length of the database is: $N = Dn$

$$E(S_{\text{bit}}) \leq m N 2^{-S_{\text{bit}}} \quad (6)$$

Because the Bonferroni correction (Eqs. (4)–(6)) changes the expectation value depending on the size of the database (the number of comparisons, or tests), the same similarity score can be statistically significant in some contexts (e.g., a single proteome with 20,000 sequences) but not in others (e.g., comprehensive protein databases with tens of millions of sequences). For example, alignment of human RefSeq protein NP_000444 (a sodium/iodide cotransporter, 643 amino acids) with *E. coli* NP_418491 (an acetate transporter, 549 amino acids) produces a local alignment score of 45 bits. If the *E. coli* sequence had been identified in a search of human proteins ($\sim 40,000$ sequences), the relationship would be statistically significant, with $E(S_{\text{bits}} = 45, D = 40,000) \leq 40,000 \times 643 \times 549 \times 2^{-45} = 4.2 \times 10^{-4}$. In contrast, the same alignment would not be significant if the *E. coli* protein had been compared to UniProt ($\sim 40,000,000$ sequences, $E() \leq 0.42$) or RefSeq ($\sim 80,000,000$ sequences, $E() \leq 0.84$).

This loss of statistical significance with increased database size can be disconcerting, because it suggests that two sequences can be homologous in searches of small proteome databases but not in searches of large comprehensive sequence sets. The discrepancy highlights the fundamental asymmetry when inferring homology using statistical significance; a *significant* similarity score *can* be used to *infer* homology, but a *non-significant* similarity score *cannot* be used to infer *non-homology*.

In this example, the alignment does reflect homology; both proteins contain a sodium:solute symporter family domain (PF00074, [Finn et al., 2016](#)) but the alignment score (45 bits) is not large enough to guarantee statistical significance after the Bonferroni correction for large datasets. In databases with tens of millions of sequences, a score of 45 bits will often occur by chance.

The relationship between database size and homology inference is key to interpreting similarity search results. Homology inference is asymmetrical; sequences that share statistically significant similarity can be reliably inferred to be homologous, and non-homologous sequences should never share statistically significant similarity after the significance has been corrected for multiple testing, but sequences that do not share significant similarity are not always non-homologous.

The expectation value $E(D)$ reports the number of times (on average) that an alignment score would be expected by chance in a database of D unrelated sequences. Thus, an alignment with an $E()$ -value of 0.001 would be expected to occur by chance once in 1000 searches of the database. $E() \leq 0.001$ is frequently used to infer homology in a single similarity search (but it *cannot* be used to infer non-homology). As the example shows, there may be more distantly related homologs with $E()$ -values ~ 1 or even higher, mixed among alignments of unrelated sequences with $E() \sim 1$.

The relationship between bit scores and statistical significance (Eqs. (4)–(6)) can be used to calculate the minimum bit score that would be statistically significant in searches of databases with different numbers of sequences. For average length proteins (400 amino-acids), to achieve $E() \leq 0.001$, a score of:

$$0.001 \leq D \times 400 \times 400 \times 2^{-S_{\text{bits}}} \quad (7)$$

$$D \geq 0.001 / (400 \times 400 \times 2^{-S_{\text{bits}}}) \quad (8)$$

$$160,000 \times D / 0.001 \leq 2^{S_{\text{bits}}} \quad (9)$$

the $-S_{\text{bits}}$ exponent is inverted

$$\log_2(160,000,000 \times D) \leq S_{\text{bits}} \quad (10)$$

$$27.25 + \log_2(D) \leq S_{\text{bits}} \quad (11)$$

For *E. coli*, $\log_2(4000) = 12$ so a 40-bit score is significant ($E(S_{\text{bit}} = 40, D = 4000) < 0.001$); for human $\log_2(40,000) \simeq 15$, so 43 bits are required. For large protein databases like UniProt ($\log_2(40 \times 10^6) \simeq 25$), so ~ 50 –55 bits are required for statistical significance.

Protein sequence similarity searching can routinely identify homologs – proteins sharing a common ancestor – that diverged > 2 billion years ago, using a single protein query sequence. More sophisticated methods, such as PSI-BLAST (Altschul *et al.*, 1997) and HMMER3/jackhmmmer (Eddy, 2011) are dramatically more sensitive. These approaches build an ancestral model of a protein family or domain that typically can identify 5–10 times as many homologs as a single sequence BLASTP search. For PSI-BLAST, this model is a Position Specific Scoring Matrix (PSSM) that captures the specific sequence changes possible at each residue across the length of the protein or domain. HMMER3/jackhmmmer uses a more sophisticated approach to build a Hidden Markov Model (HMM) that captures both the substitution frequencies at each position across the model and the variation in gap-frequency at different locations. However, despite their much higher sensitivity compared with pairwise-sequence comparison, PSI-BLAST and HMMER3 can also fail to identify distant homologs that can be identified using structure comparison (Pearson and Sierk, 2005). Today, there is no sequence-based method that is as sensitive as structure comparison, though methods that use HMM-HMM comparison approach this ideal (Sadreyev *et al.*, 2003; Remmert *et al.*, 2012).

Case Studies

The discussion above has focused on the inference of homology from protein sequence comparison. Protein sequence comparison provides dramatically greater evolutionary look-back time than DNA sequence comparison for several reasons: (1) protein sequences change more slowly than DNA sequences, because many DNA changes are silent – they do not change the encoded amino acid; (2) protein sequence comparison can take advantage of an informative scoring matrix, e.g., the BLO-SUM62 matrix gives similar, but non-identical (e.g., arginine, lysine, histidine, or leucine, isoleucine, and valine) positive alignment scores, and gives amino-acids that are rarely substituted negative scores; (3) the amino-acid alphabet is larger (20 vs 4 residues), so chance alignments between protein sequences occur at much lower percent identity; (4) protein sequence databases are much smaller than DNA sequence databases, since 95% or more of metazoan genomes do not code for protein; and (5) while unrelated protein sequences are indistinguishable from random protein sequences, unrelated DNA sequences can have local composition and other biases that make it difficult to produce random DNA sequences that look like real, unrelated DNA sequences. As a result, while expectation values with $E() < 0.001$ will occur by chance once in 1000 searches for protein sequences (Pearson and Sierk, 2005), they occur much more often in DNA sequences, so $E() < 10^{-10}$ is often used as a threshold for inferring homology with DNA:DNA comparison.

The dramatic difference in sensitivity between protein and DNA comparison is illustrated in Fig. 1, which shows how the sensitivity of sequence similarity searches drops with evolutionary distance, using protein:protein, DNA:protein, and DNA:DNA comparisons. The figure illustrates that protein:protein (SSEARCH, BLASTP) and translated-DNA:protein (FASTX, BLASTX) can identify homologs in 99%–100% of the queries among the vertebrates (500 Mya), in more than 95% of the queries among metazoa (800 Mya), in more than 80% of queries between humans and plants (1500 Mya), and in ~40% of queries between humans and *E. coli*. Moreover, there is relatively little loss in sensitivity with translated-DNA:protein vs protein:protein alignments. In contrast, while DNA:DNA searches (BLASTN) can find homologs for 93% of the queries between human and mouse (and this number is high because the DNA sequences encode protein; non-coding sensitivity would be much lower), comparison of human queries find homologs for only 15% of queries in fish, and less than 5% of queries between humans and *Drosophila*. Protein:protein and translated-DNA:protein comparisons allow homologs to be identified 10-times earlier in evolutionary time than DNA:DNA comparisons.

Discussion

The identification of biological homologs – genes, protein sequences, or protein structures that share a common ancestor – is based on the parsimonious principle that when sequences or structures share much more similarity than is expected by chance, the simplest explanation for the excess similarity is common ancestry, i.e., the sequences are similar because they are descendants (copies) of an ancient protein. Thus, the ability to recognize distant homologs depends both on a method to calculate a similarity score, e.g., BLASTP (Camacho *et al.*, 2009) or FASTA (Pearson, 2016), and an accurate model of the amount of similarity expected by chance. Today, because there are so many fully sequenced genomes and their corresponding proteomes, virtually every similarity search identifies hundreds if not thousands of homologs. But pair-wise similarity searches sometimes miss large numbers of homologs that can be detected by transitive similarity searches, model based methods (PSI-BLAST, HMMER3), and structure comparison. As a result, even the most sensitive sequence based similarity searches miss distant homologs. Despite its imperfections, homology inference from excess sequence similarity is the most powerful and reliable strategy for characterizing new protein and DNA sequences, and is most effective when comparisons use protein sequence databases.

See also: Gene Duplication and Speciation. Inference of Horizontal Gene Transfer: Gaining Insights Into Evolution via Lateral Acquisition of Genetic Material. Natural Language Processing Approaches in Bioinformatics. Phylogenetic analysis: Early evolution of life. Sequence Analysis

References

- Altschul, S.F., Boguski, M.S., Gish, W., Wootton, J.C., 1994. Issues in searching molecular sequence databases. *Nat. Genet.* 6, 119–129.
- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. A basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.
- Altschul, S.F., Koonin, E.V., 1998. Iterated profile searches with PSI-BLAST – A tool for discovery in protein databases. *Trends Biochem. Sci.* 23, 444–447.
- Altschul, S.F., Madden, T.L., Schaffer, A.A., *et al.*, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.
- Arratia, R., Gordon, L., Waterman, M.S., 1986. An extreme value theory for sequence matching. *Ann. Stat.* 14, 971–993.
- Brenner, S.E., Chothia, C., Hubbard, T.J., 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl. Acad. Sci. USA* 95, 6073–6078.
- Camacho, C., Coulouris, G., Avagyan, V., *et al.*, 2009. Blast + : Architecture and applications. *BMC Bioinform.* 10, 421.
- Eddy, S.R., 2011. Accelerated profile HMM searches. *PLOS Comput. Biol.* 7 (10), e1002195.
- Farrar, M., 2007. Striped Smith-Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics* 23, 156–161.
- Finn, R.D., Clements, J., Eddy, S.R., 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Res.* 39, W29–W37.
- Finn, R.D., Coghill, P., Eberhardt, R.Y., *et al.*, 2016. The Pfam protein families database: Towards a more sustainable future. *Nucleic Acids Res.* 44 (D1), D279–D285.
- Gerstein, M., 1998. Measurement of the effectiveness of transitive sequence comparison, through a third 'intermediate' sequence. *Bioinformatics* 14 (8), 707–714.
- Gotoh, O., 1982. An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 705–708.
- Karlin, S., Altschul, S.F., 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA* 87, 2264–2268.
- McLachlan, A.D., 1971. Tests for comparing related amino-acid sequences: Cytochrome c and cytochrome c 551. *J. Mol. Biol.* 61, 409–424.
- Mott, R., Tribe, R., 1999. Approximate statistics of gapped alignments. *J. Comput. Biol.* 6, 91–112.
- Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453.
- Pearson, W.R., 2016. Finding protein and nucleotide similarities with FASTA. *Curr. Protoc. Bioinformatics* 53, 3.9.1–3.9.25.
- Pearson, W.R., Li, W., Lopez, R., 2017. Query-seeded iterative sequence similarity searching improves selectivity 5–20-fold. *Nucleic Acids Res.* 45 (7), e46.
- Pearson, W.R., Sierk, M.L., 2005. The limits of protein sequence comparison? *Curr. Opin. Struct. Biol.* 15, 254–260.
- Reeck, G.R., de Haen, C., Teller, D.C., *et al.*, 1987. Homology in proteins and nucleic acids: A terminology muddle and a way out of it. *Cell* 50, 667.
- Remmert, M., Biegert, A., Hauser, A., Soeding, J., 2012. HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods* 9 (2), 173–175.
- Sadreyev, R.I., Baker, D., Grishin, N.V., 2003. Profile-profile comparisons by COMPASS predict intricate homologies between protein families. *Protein Sci.* 12, 2262–2272.
- Schaffer, A.A., Aravind, L., Madden, T.L., *et al.*, 2001. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.* 29, 2994–3005.
- Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197.
- Sonnhammer, E.L.L., Eddy, S.R., Birney, E., Bateman, A., Durbin, R., 1998. Pfam: Multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res.* 26, 322–325.
- Winter, W.P., Walsh, K.A., Neurath, H., 1968. Homology as applied to proteins. *Science* 162 (3861), 1433.

DNA Barcoding: Bioinformatics Workflows for Beginners

John-James Wilson, University of South Wales, Pontypridd, United Kingdom and Naresuan University, Phitsanulok, Thailand
Kong-Wah Sing, Kunming Institute of Zoology, Chinese Academy of Sciences, Yunnan, P.R. China
Narong Jaturas, Naresuan University, Phitsanulok, Thailand

© 2019 Elsevier Inc. All rights reserved.

Introduction

Just as species show differences in their morphology, ecology, and behavior, they also show differences in their DNA sequences (Wilson *et al.*, 2017). “DNA barcoding,” used in a broad sense, refers to the use of short, standardized DNA sequences as markers for the recognition of species. When used more precisely, “DNA barcoding” refers to the technique of sequencing a short fragment of the DNA sequence of the mitochondrial cytochrome c oxidase subunit I (COI) gene, the animal “DNA barcode,” from a taxonomically unknown specimen and performing comparisons with a library of DNA barcodes from taxonomically known specimens to establish a taxonomic identification.

DNA barcoding requires basic molecular biology methods (which pre-date the term DNA barcoding) to extract and amplify the DNA barcode sequence fragment from the unknown specimen (Fig. 1). Generally, this sample of amplified DNA is then passed to commercial companies for inexpensive Sanger sequencing, or, particularly in the case of mixed, bulk samples, for high-throughput (next generation) sequencing. These molecular biology methods are not covered in this article which focuses on bioinformatics workflows following the receipt of digital DNA sequences from a sequencer (Fig. 1). For a step-by-step guide to the molecular biology methods used for standard (animal) DNA barcoding see Wilson (2012) and for an approximation of costs in developing countries see Sing *et al.* (2016). Brandon-Mong *et al.* (2015) provide an example of a method (bulk extraction and bulk PCR) for use prior to high-throughput sequencing, i.e., DNA metabarcoding.

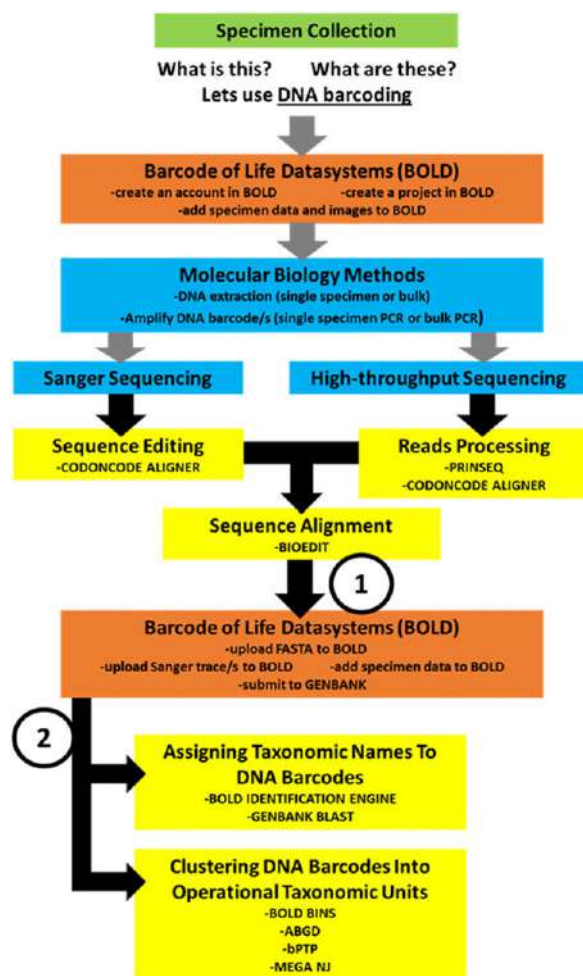


Fig. 1 The DNA barcoding workflow.

The ultimate aim of a DNA barcoding bioinformatics workflow is to (1) produce a “clean” or “reliable” digital representation of the DNA barcodes, and (2) use these DNA barcodes to obtain information about the taxonomy of the unknown specimen through algorithms enabling DNA sequence comparisons, in conjunction with DNA sequence libraries (Fig. 1).

Background/Fundamentals

It is important to realize that full exploitation of DNA barcodes for species recognition will only be possible after assembling a comprehensive library linking organisms (and Linnaean taxonomy) with their DNA barcodes (see Wilson *et al.*, 2017 for an assessment of DNA barcode library coverage for insects). With this in mind, in an effort to promote DNA barcoding research and DNA barcode library building (e.g., Wilson *et al.*, 2013), we have been organizing and facilitating DNA barcoding workshops in Southeast Asia, a mega diverse region with relatively low DNA barcode coverage in public libraries (Wilson *et al.*, 2016). We have created a website, DNA Barcoding Workshops (DBW), as a companion resource to our workshops and much of the material in this article derives from our experience running these workshops.

This article is intended as a guide for absolute beginners to DNA barcoding, and provides step-by-step instructions for basic bioinformatics workflows following receipt of DNA sequences from a DNA sequencing facility.

Application

Getting Started With DNA Barcoding: The Barcode of Life Datasystems

The Barcode of Life Datasystems (BOLD) (Ratnasingham and Hebert, 2013) is a cloud-based data storage and analysis platform developed and maintained by the Centre for Biodiversity Genomics (CBG) in Canada. It consists of four main modules, a data portal, an educational portal, a registry of Barcode Index Numbers (BINs) (putative species), and a data collection and analysis workbench. BOLD is the recommended place to manage your DNA barcoding research. Please see the BOLD Handbook available from the BOLD website (under “Resources”) and the DNA Barcoding Workshop (DBW) website for step-by-step instructions for: (i) creating an account on BOLD; (ii) preparing specimen data for BOLD; (iii) creating a project on BOLD; and (iv) adding records and images to BOLD. You will need to have completed these steps in order to upload your DNA barcode sequences to BOLD and to use the BINs system. Note that BOLD keeps all of your work (specimen records, sequences, etc.) private, accessible only to yourself and your coworkers, until you choose to make the work public.

Editing Sanger Sequences

To learn more about the process of Sanger sequencing see the resources on the DNA Learning Centre website from the Cold Spring Harbor Laboratory. The outputs of Sanger sequencers are trace files (also known as chromatograms and electropherograms), digital representations of DNA sequences. For a number of reasons, a trace may not be “clean”, it may be “messy”, and the first step is to “edit” the sequence, a process during which “messy” parts of sequence are removed and only “reliable” sequence is retained.

- a) Usually your sequences will come back from the sequencing company by email in a zip (folder) file. An example of a zip file containing 12 trace files (6 each trace files for forward and reverse) (6 unknown specimens), representing the kind of zip file you could receive from a sequencing company, is provided for download as Supplemental File 1.
- b) Unpack (extract) the zip file to your desktop. This should create a regular folder on your desktop, which you can name *Traces*. There are two sets of files for each sequence (e.g., *NUMBEROFSAMPLE_F.ab1* and *NUMBEROFSAMPLE_F.txt*). The files you are interested in have an extension .ab1 (e.g., *NUMBEROFSAMPLE_F.ab1* and *NUMBEROFSAMPLE_R.ab1*). Delete the other files in the folder.

For sequence editing we recommend the program CODONCODE ALIGNER which is used at the Canadian Centre for DNA Barcoding (CCDB). Information on CODONCODE ALIGNER, including a free trial version, can be found at the Codoncode Corporation website. The following steps describe the process of Sanger sequence editing for multiple specimens whose DNA barcodes were amplified using the primers *LCO1490* and *HCO2198* (see Wilson, 2012; Brandon-Mong *et al.*, 2015). For practice you can go through these steps using the example files in Supplemental File 1. For other primer sets adapt accordingly.

- a) Open CodonCode Aligner and choose *Create a new project* and press OK.
- b) Go to *File > Import > Add Folder...* navigate to the desktop and select the folder of traces [which should be named *Traces* if you followed the suggestion above]. Click *Open > Import*.
- c) To see the files you just imported press ► besides the *Unassembled Samples* folder.
- d) The .ab1 files should be of the form *NUMBEROFSAMPLE_F.ab1* where the second part “F” refers to the direction, i.e. Forward.
- e) Sort the files by quality by double-clicking on *Quality*. Any sequences that are of very poor quality (look for a big difference between the sequence length and the quality score; a higher quality score is better) can be deleted by highlighting the sequence and clicking *Edit > Move to Trash*.
- f) Next we will group our sequences by direction for easy editing.

- g) Make sure the *Unassembled Samples* folder is highlighted. Select the *Contig* menu and move the cursor over *Advanced Assembly*. From the options that appear select *Assemble in Groups*.
- h) A new window will appear. Click the button *Define name parts...*
- i) There are two name parts to our file names (see above). The first part of our file names refers to the number of the sample and for our purposes the option in the *Meaning* menu (first row) can be left as *Clone*. Since the sample number is followed by an underscore, choose *_ (underscore)* in the *Delimiter* menu next to *Clone* (if it is not already selected).
- j) For the second row choose *Direction* in the *Meaning* menu. We can ignore the *Delimiter* menu for the *Direction* part because there is nothing following the direction in our file names.
- k) Delete all additional name parts that may appear in the window (if any), and next click *Preview...* to check how CodonCode Aligner is interpreting the sample names.
- l) Click *Close* to exit the preview. Click *OK* to return to the *Assemble in Groups* window.
- m) We first want to assemble our samples according to direction. Choose *Direction* in the *Name part:* dropdown menu. Then click *Assemble*. You should now have two folders, one called **F** with the forward sequences and one called **R** with the reverse sequences. [Note: if you only sent your PCR products for sequencing in one direction, i.e., with one primer, then you will only have one folder.].
- n) We will deal with the reverse sequences first. The first step is to reverse complement the sequences. Highlight the **R** folder, select *Edit > Reverse complement*.
- o) Next we need to cut the primer from the sequence. Double click the **R** folder to open it. For the reverse sequences, you need to find the forward primer motif and delete it from the beginning of the consensus sequence at the bottom of the window. You will find the primer around 30 nucleotides from the end of the raw sequence. For example, you would need to delete the section of the sequence marked below in **bold** and everything to the left of it. Highlight it on the **consensus** sequence at the bottom of the window and press the **Backspace** key on the keyboard. ←AAAGATATTGGAACATTATATTATTTT...
- p) Next go to the opposite end of the consensus, the far right. Delete the consensus sequence from the point where the sequence gets messy. This will be apparent due to lots of green highlight. For example [it will not look exactly like this], delete the section marked in **bold** and everything to the right of it. Highlight it on the **consensus** sequence at the bottom of the window and press the **Delete** key on the keyboard. Close the window. ... TCITTTTTTGACCCTGCTGGTGGAGG**TTTCTAGCAGGATC**→
- q) Double click the **F** folder to open it. Go to the far right of the consensus sequence and find the reverse complement reverse primer motif at the very end. This should be around 650 bp on the raw sequence. For example, you would delete the section marked in **bold** and everything to the right of it. Highlight it on the **consensus** sequence at the bottom of the window and press the **Delete** key on the keyboard. ... CAACATTTATTTT**GCATTTTGG**→
- r) Next go to the opposite end of the consensus, the far left, and delete the consensus sequence from the point where the sequences get messy. This will be apparent due to lots of green highlight. For example [it will not look exactly like this], you would delete the sequence in **bold** and everything to the left of it. Highlight the region on the consensus sequence at the bottom of the window and press the **Backspace** key on the keyboard. Close the window. ←**ATGCTTTTTTTTKGG**TGTTTAATCAGGACTAATTGGAAC TTC
- s) Dissolve both the **F** and **R** folders by highlighting them and clicking the button marked with a red **X**.
- t) Now we are going to combine the forward and reverse sequence from each specimen into a contig. Highlight the *Unassembled Samples* folder and open the *Contig* menu. Move the cursor over *Advanced Assembly*. From the options that appear select *Assemble in Groups*. This time choose *Clone* in the *Name part:* menu, then click *Assemble*. [Note: if you only sent your PCR products for sequencing in one direction (with one primer) then you will need to check each sequence individually rather than checking a consensus (contig).][Note: specimens which only sequenced successfully in one direction will have files which remain in the *Unassembled Samples* folder.]
- u) The contigs are likely to be in reverse complement orientation. Highlight every folder (contig), select *Edit > Reverse complement*.
- v) Open each folder (contig) in turn by double-clicking. Correct ambiguous positions (shown in red, in green highlight, and/or as N) and gaps ("—") in the consensus sequence by checking the original traces. This is done by double-clicking on the **consensus sequence** at the bottom of the window. Always check both trace files (forward and reverse) and compare them. [Note: the corrected consensus sequence should have **NO** gaps.].
- w) Generally if traces conflict (i.e., different colored peaks appear in the same location on the forward and reverse chromatograms) you can decide which is more reliable based on sequence quality (e.g., less background noise, taller peaks).
- x) Check the contigs first, then check the individual single sequences in the *Unassembled Samples* folder, if any.
- y) To export the consensus sequences, highlight all the folders, go *File > Export > Consensus Sequences...*, choose *Current selection*. Open the *Options* and select *Include gaps in FASTA* but deselect all other options. Press *Export*. Save the file to the desktop as *sequences.fasta*.
- z) If necessary, to export single direction sequences, go *File > Export > Samples...*, choose *Current selection*. Press *Export*. Save the file to the desktop as *sequences_single.fasta*.

Processing High-Throughput Sequencing Reads

The files output by an Illumina MiSeq high-throughput sequencer are in FASTQ format. FASTQ is similar to FASTA but contains additional data. Note that high-throughput sequencing reads are generally demultiplexed and adapter and primer-trimmed

onboard the sequencer (e.g., onboard the MiSeq using the MiSeq Reporter software). Because the FASTQ outputs are large files, the sequencing company probably will not send the files by email but will email you a link to a website from where you can download the files, usually (like Sanger sequences), packed into a zip file. Two FASTQ files (Paired-end files) are output from each sequencing run, which you can think of as the *Forward* and *Reverse* sequences.

The following workflow describe steps taken for processing high-throughput reads for bulk arthropod samples whose DNA was amplified using the primers *mlCOIintF* and *HCO2198* (see [Brandon-Mong et al., 2015](#)). A zip file containing some example FASTQ Paired-end files are available for download as Supplemental File 2. For practice you can go through these steps using these example files. For convenience, save the files in a folder (called *Reads*) on your Desktop.

It is important to note that the steps provided below are crude methods for processing a very small number of high-throughput sequencing reads. The field of DNA metabarcoding is a relatively new field and much work is being undertaken to develop methods to reduce the number of “spurious” reads generated and retained by bioinformatics pipelines for high-throughput sequencing reads ([Brandon-Mong et al., 2015](#)) (see Future Directions below). For FASTQ files which are larger than the example provided as Supplemental File 2, CODONCODE ALIGNER is probably not a suitable program, and for beginners, it may be better to register and use applications provided on the GALAXY webserver. Considering that DNA metabarcoding is the focus of another article in this book, we do not provide additional details here.

- Open the PRINSEQ webserver ([Schmieder and Edwards, 2011](#)), click *Use PRINSEQ* and click on *Upload Data*.
- Your files are *FASTQ Paired-end* so choose that option.
- Select the two FASTQ files you have saved on your Desktop (in the *Reads* folder).
- Under *Please select the statistics you want to generate*. Choose *None* for all options then click *Continue*.
- Wait while PRINSEQ processes your data.
- Once it is finished click *Process Input Data*.
- Choose the options from [Table 1](#).
- Choose to Output the data as FASTA, *Data passing all the filters (good)*.

Table 1 Parameters used for processing high-throughput sequencer reads

<i>Program</i>	<i>Parameter</i>	<i>Option</i>
PRINSEQ	Trim #nucleotides from 5'-end	25
	Trim #nucleotides from 3'-end	25
	Trim ends by quality scores	While Mean of scores is < (less then) 10 (5'-end) and 10 (3'-end_ using window size of 5 with step size 5
	Trim poly-A/T tails from 5'-end	That are equal to or longer than 5
	Trim poly-A/T tails from 3'-end	That are equal to or longer than 5
	Minimum sequence length in bp	200
	Maximum sequence length in bp	300
	Minimum mean quality score	32
	Minimum GC content in %	25
	Maximum GC content in %	35
	Maximum allowed rate of Ns in %	0
	Maximum number of allowed Ns	0
	Remove sequences with characters other than A, C, G, T or N	Yes
	Low-complexity threshold	80 (using Entropy)
	Dereplicate data	Remove exact sequence duplicates, Remove 5' sequence duplicates, Remove 3' sequence duplicates, Remove reverse complement exact sequence duplicates, Remove reverse complement 5'/3' sequence duplicates
CODONCODE ALIGNER	Algorithm	Local alignments
	Min. percent identity	97.5
	Min overlap length	20
	Min score	20
	Max. unaligned end overlap	100.00
	Bandwidth (max. gap size)	30
	Word length	10
	Max successive failures	50
	Match score	1
	Mismatch penalty	− 2
	Gap penalty	− 2
	Additional first gap penalty	− 3

- i) Click *Generate Files*.
- j) Use right click on the file you want (all of the FASTA files) to download and select “Save Link As” to save the file into a folder on your Desktop (which you could name *Filtered Reads*).
- k) Next we will be using CODONCODE ALIGNER (see above). Go to *File > Import > Add Folder...* Import the sequences in the FASTA files in the *Filtered Reads* folder. Click *Rename Duplicates*.
- l) Select all the sequences and click *Assemble*. Make sure the settings for *Assembly* found under *Edit > Preferences* are set to match those in [Table 1](#).
- m) Select all the contigs, including the *Unassembled Samples* folder and again, *Assemble*. Repeat until the number of contigs cannot be reduced any further.
- n) Delete (*Move to Trash*) the sequences still remaining in the *Unassembled Samples* folder.
- o) Export all the remaining contigs as consensus sequences (FASTA file), highlight all the folders, go *File > Export > Consensus Sequences...*, choose *Current selection*. Make sure none of the *Options* are selected.

Sequence Alignment

The alignment of DNA barcode sequences is a necessary step before two or more DNA barcode sequences can be compared with one another. Sequence alignment is the process of lining up nucleotides which are assumed to have the same common ancestor (i.e., thought to be homologous). BIOEDIT is the most commonly used program for small-scale sequence alignment, and is free for use by any and all interested parties. BIOEDIT can be downloaded from the program website but is no longer being regularly maintained.

- a) Open your FASTA file (e.g., *sequences.fasta*) in BIOEDIT.
- b) If necessary, you can then use *File > Import > Sequence alignment file* to add additional sequences (e.g., *sequences_single.fasta*) to the alignment.
- c) Make sure *Mode*: is set to *Edit* using the dropdown menu.
- d) Another dropdown menu will become visible to the right of the *Edit* dropdown. Make sure this is set to *Insert*.
- e) Sequences that have ended up in the FASTA file in the wrong orientation (with standard primers, full length arthropod DNA barcodes should typically start AAC or TAC) may be corrected by highlighting the sequence name by clicking the cursor on it, clicking the *Sequences* menu at the top of the screen. Moving the cursor down the dropdown to *Nucleic Acid* and clicking *Reverse complement*.
- f) Sequences all need to be the same length (typically 658 bp for full length DNA barcodes; 300 bp for metabarcodes) and aligned to each other (before proceeding for further analysis or before uploading to BOLD and GenBank). This can be done by typing additional *N(s)* at the beginning and end of the sequences in the *Edit* mode. Be sure to check across the whole of the alignment of the sequences that you have added the correct number of *N(s)*.
- g) In [Fig. 2](#) featuring a 50-bp barcode for simplification, *JKN001-17* is of full length. *JKN002-17* needs 6 *Ns* adding to the left side of the sequence to become aligned, while *JKN003-17* needs 4 *Ns* adding to the right side of the sequence to be 50 bps long. *JKN005-17_F* needs to be reverse complemented to be in the same orientation as the other sequences.
- h) Sanger sequences which were not part of a consensus (i.e., when one direction failed but the single sequence is of sufficient length and quality for submission to BOLD) may appear in the FASTA still tagged with the direction. This needs to be deleted, e.g., the sequence named *JKN005-17_F* should be renamed as *JKN005-17*.
- i) If you are having trouble with the alignment, a good quality (i.e., 658[0n]) sequence can be downloaded from BOLD and imported into your alignment file as a guide, e.g., *MHAHC824-05* ([Fig. 2](#)). Be sure to delete this sequence before saving the file.
- j) Save the file (*File > Save*) (e.g., save as *sequences_aligned.fasta*).

Following these steps you can upload your FASTA file (i.e., *sequences_aligned.fasta*) (and Sanger sequence trace files) to your project on BOLD. Step-by-step instruction can be found on the DBW website. Once sequences are registered in BOLD, they can also be submitted to GenBank using the *Submit to GenBank* option under the *Publication* tab on the BOLD project console.

Assigning Taxonomic Names to DNA Barcodes

For the assignment of taxonomic names to DNA barcodes (and by extension, checking sequences for contamination) we find it is best to use online tools to ensure you are searching the most recent DNA barcode library available (the DNA barcode library is constantly growing; [Wilson et al., 2017](#)). For checking DNA barcodes for contamination and/or establishing a taxonomic identity, we would commonly “blast” our DNA barcodes against the BOLD and GenBank libraries. By statistically assessing how well library and query (unknown) DNA barcodes match, we can infer homology and transfer information (such as putative species membership) to the unknown specimen.

When there are no species-level matches in the libraries, some Linnaean taxonomic information may still be retrievable. [Wilson et al. \(2011\)](#) performed an investigation of the possibilities of assigning DNA barcodes to higher taxonomic groups (genus, family) when no species matches (i.e., with > 98% similarity) are available in the library. Based on those results we suggest using a strict tree-based approach. BOLD can provide a Neighbor-Joining tree containing the query DNA barcode and the top matches by clicking *Tree Based Identification* on the *Specimen Identification Request* page (see below).

Contig

```
JKN001-17_F  AACTNTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTT (Forward)
JKN001-17_R  AACTTTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTN (Reverse)
JKN001-17    AACTTTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTT (Consensus)
```

[A contig is two or more sequences grouped together in the same folder, usually one Forward and one Reverse, to produce a Consensus. Note: N = A, T, C, or G]

Unaligned FASTA (BIOEDIT) (*sequences.fasta*)

```
JKN001-17  AACTTTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTT
JKN002-17  ATATTTTATTTTGGAAATTTGAGCTGGATTAATTGGAACCTCAT
JKN003-17  AACTCTATATTTTATTTTGGAAATTTGACCAGGATTACTAGGAACCT
JKN004-17  TCTATATTTTATTTTGGAAATTTGACCAGGTTTAGTTGGAACCTCAT
JKN005-17_F ATGATGTTCTTAACATACCTGCTCAAATACCAAAATAAAATATAAAGTT
MHAHC82-05 AACTTTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTT
```

Aligned FASTA (BIOEDIT) (*sequences_aligned.fasta*)

```
JKN001-17  AACTTTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTT
JKN002-17  NNNNNNATATTTTATTTTGGAAATTTGAGCTGGATTAATTGGAACCTCAT
JKN003-17  AACTCTATATTTTATTTTGGAAATTTGACCAGGATTACTAGGAACCTNNNN
JKN004-17  NNNCTATATTTTATTTTGGAAATTTGACCAGGTTTAGTTGGAACCTCAT
JKN005-17  AACTTTATATTTTATTTTGGAAATTTGACCAGGATGTTAGGAACATCAT
```

Aligned FASTA (MS WORD) (*sequences_aligned.fasta*)

```
>JKN001-17
AACTTTATATTTTATTTTGGAAATTTGACCAGGAATAGTAGGAACCTCTT
>JKN002-17
NNNNNNATATTTTATTTTGGAAATTTGAGCTGGATTAATTGGAACCTCAT
>JKN003-17
AACTCTATATTTTATTTTGGAAATTTGACCAGGATTACTAGGAACCTNNNN
>JKN004-17
NNNTCTATATTTTATTTTGGAAATTTGACCAGGTTTAGTTGGAACCTCAT
>JKN005-17
AACTTTATATTTTATTTTGGAAATTTGACCAGGATGTTAGGAACATCAT
```

Fig. 2 Examples of DNA sequences viewed in BIOEDIT and MS WORD.

BOLD identification engine

- Open your aligned FASTA file (i.e., *sequences_aligned.fasta*) in MS WORD using right click *Open with*. Select *All* of the text and *Copy*.
- Click *IDENTIFICATION* on the BOLD homepage, select *All Barcode Records On BOLD*, paste the text from your FASTA file (i.e., *sequences_aligned.fasta*) into the box *Enter sequences in fasta format*.
- BOLD was designed specifically for DNA barcoding, so the *Specimen Identification Request* page displays a list of library records and their similarity to the query (i.e., the DNA barcode of the unknown specimen). A self-explanatory *Identification Summary* is also provided. An example of a BOLD search result is shown in [Fig. 3\(a\)](#) where the sequence can be conclusively identified as *Amauthuxidia amythaon*.

GenBank BLAST

- Click *Nucleotide BLAST* on the BLAST homepage, paste the text from your FASTA file into the box *Enter accession number(s), gi(s), or FASTA sequence(s)*, and make sure the *Database* selection is *Others*.
- BLAST pre-dates DNA barcoding, and is used for a variety of purposes, so the output is a little more difficult to interpret. Like BOLD, a list of library records is displayed, generally with the closest matching library sequence (i.e., the highest % *Identity*) at the top. Four other statistics are supplied: *Max score* indicates the highest alignment score (bit-score) between the query DNA barcode and the library sequence segment (the higher the better, 1000 is very good); *Total score* and *Query coverage* are generally not applicable for protein-coding genes such as the animal DNA barcode; *E-value* is the most important statistic for DNA barcoding and indicates number of alignments expected by chance with a particular score or higher (the closer to 0 the better). An example of a BLAST search result is shown in [Fig. 3\(b\)](#) where the sequence can be conclusively identified as *Amauthuxidia amythaon*.

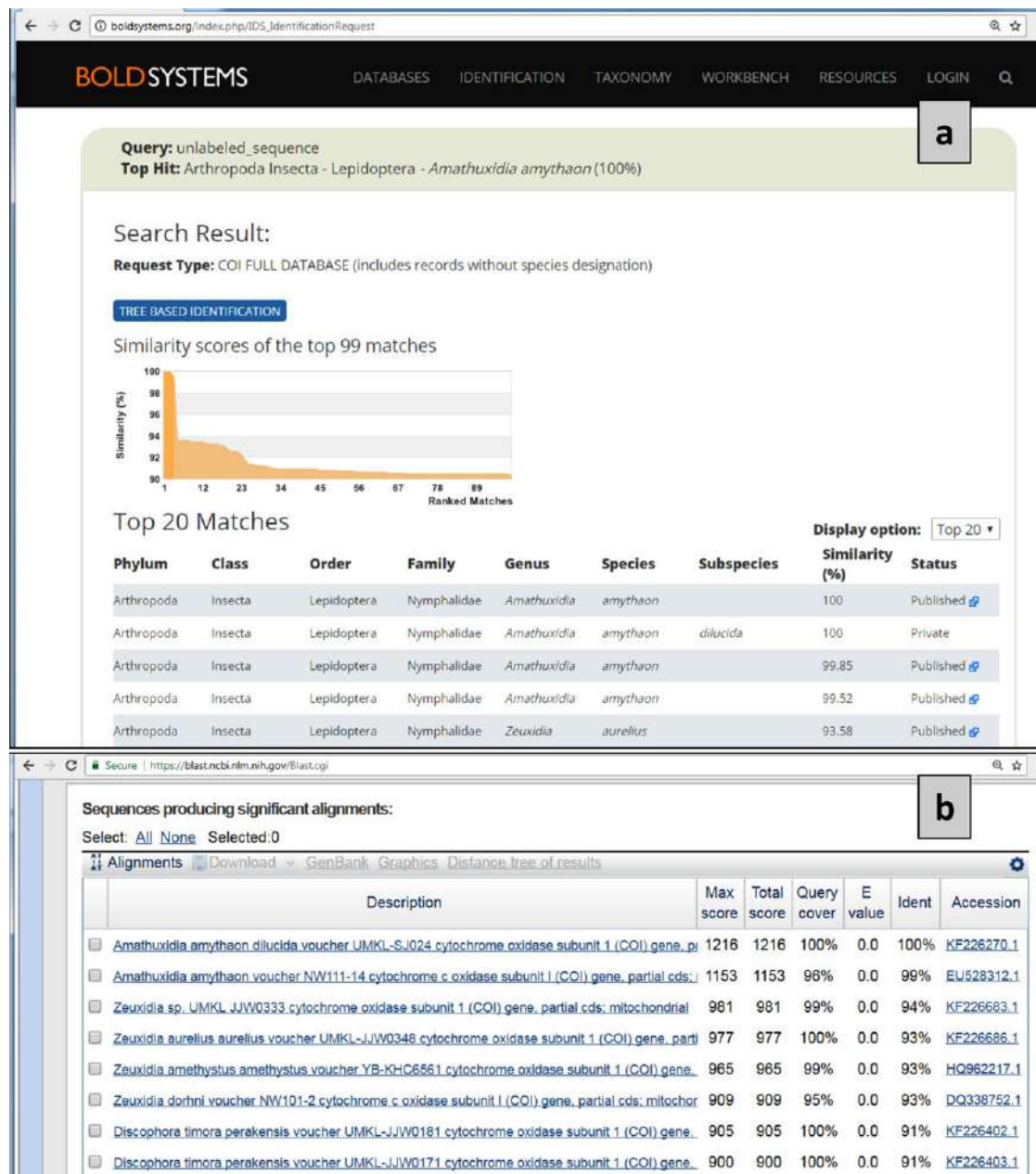


Fig. 3 Examples of the results of sequence identification requests using (a) BOLD identification engine and (b) GenBank BLAST.

Clustering DNA Barcodes Into Operational Taxonomic Units

Due to the incompleteness of DNA barcode reference libraries, and inconsistencies in the use of Linnaean names, besides assigning Linnaean taxonomy to DNA barcodes, it is also valuable and common practice to cluster DNA barcodes into Operational Taxonomic Units (OTU) (see [Sukantamala et al., 2017](#); [Casas et al., 2017](#)). OTU are groupings of similar sequences which are objective, operational, “species”-level units ([Ratnasingham and Hebert, 2013](#)). Several methods are available for clustering DNA barcodes into OTU and we provide step-by-step instructions for the most commonly used and easily accessible programs in this article.

BOLD Barcode Index Number system

DNA barcodes of a minimal length (200 bp) and quality (maximum proportion of Ns) requirement are automatically clustered into OTU known as Barcode Index Numbers (BINs) upon upload to BOLD. Such clusters, produced by refine single linkage analysis across the entire BOLD library have been shown to closely correspond to species recognized through traditional taxonomic approaches (i.e., morphology). Consequently, the alphanumeric identifiers can act as surrogate taxonomic names in the absence of full Linnaean assignments ([Wilson et al., 2016](#)).

Automatic Barcode Gap Discovery

Another popular tool to cluster DNA barcodes into OTU is Automatic Barcode Gap Discovery (ABGD) (Puillandre *et al.*, 2012). ABGD uses an automatic recursive procedure to converge on the best patterns for the dataset and clusters DNA barcodes into groups accordingly. The median number of ABDG groups can be used as the basis for OTU as this has produced good correspondence with traditional species in empirical studies. ABGD is available as a web interface.

- a) From the ABGD homepage click *Take me to ABGD Web site*.
- b) Open your aligned FASTA file (i.e., *sequences_aligned.fasta*) in MS WORD (using right click *Open with*), then copy and paste the DNA barcodes into the *Or paste your data (FASTA alignment) here* box.
- c) Change the *X (relative gap width)*: to 1, but keep all other settings as default.
- d) Click *Go* and then click to see the results.
- e) On the graph click the datapoint representing the median number of groups. If available choose the Recursive partition, rather than the Initial partition.
- f) The page that opens shows a list of groups and the DNA barcodes contained within them.

Neighbor-Joining in MEGA

Molecular Evolutionary Genetics Analysis (MEGA) program is an extremely popular program for tree-building and is free to download from the program website (Tamura *et al.*, 2013). Once a Neighbor-Joining (NJ) tree has been built, DNA barcodes can be sorted to OTU ad hoc based on the tree branching pattern (topology) and branch lengths (see the NJ tree and discussion in Sukantamala *et al.*, 2017). Note that NJ trees are not technically “phylogenetic” trees. Phylogenetic trees are constructed on the basis of synapomorphies, whereas NJ trees are phenetic trees, constructed on the basis of sequence similarity. DNA barcoding is concerned with relationships amongst sequences at the “species” boundary and not the reconstruction of phylogeny, so generally NJ trees are sufficient for the analysis of DNA barcodes.

- a) Start MEGA by double-clicking on the MEGA desktop icon.
- b) You can then choose *File > Open A File/Session...* > Select your FASTA file (i.e., *sequences_aligned.fasta*) from your Desktop > *Open > Analyze > Nucleotide Sequences > OK > Yes > Invertebrate Mitochondrial > OK*.
- c) You can then choose *Phylogeny > Construct/Test Neighbor-Joining Tree...* > *Yes*.
- d) In the *Analysis Preferences* window *Options Summary* tab choose the options: *Substitutions Type: Nucleotide; Model/Method: p-distance; Substitutions to Include: d. Transitions + Transversions; Gaps/Missing Data Treatment: Pairwise deletion*.
- e) Click *Compute* to accept the defaults for the rest of the options and begin the computations. A progress indicator will appear briefly before the tree displays in the *Tree Explorer*.
- f) The tree can then be exported as a Newick (text) file (e.g., *tree.nwk*) *File > Export Current Tree (Newick)*.
- g) To select a branch, click on it with the left mouse button. If you click on a branch with the right mouse button, you will get a small options menu that will let you flip the branch and perform various other operations on it. To edit the taxon labels, double click on them.
- h) Change the branch style by using *View > Tree/Branch Style*.
- i) Select *View > Topology Only* to display the branching pattern (without actual branch lengths on the screen).
- j) Edit your tree to create the image that you want to use in your publication or report. The tree can then be exported as an image *Image > Save as PDF file* or *Image > Save as PNG file*.

Bayesian Poisson Tree Processes

A Bayesian implementation of the Poisson Tree Processes (bPTP) (Zhang *et al.*, 2013) program for generating OTU is available through a web interface. bPTP can be used to delimit phylogenetic species in a similar way to the popular and widely used General Mixed Yule Coalescent (GMYC) approach (Pons *et al.*, 2006), but without the requirement for an ultrametric tree.

- a) From the bPTP homepage, click the *Browse...* button to locate and upload your NJ tree (select the Newick file, e.g., *tree.nwk*, not the image file).
- b) Leave all the settings as the defaults and enter your email address.
- c) Click to refresh until the results appear. Two trees are displayed (a maximum likelihood solution and a Bayesian solution) but they are likely to be the same topology.
- d) Click *Download delimitation results*. The page that opens shows a list of groups (species) and the DNA barcodes contained within them.

Illustrative Example(s) or Case Studies

We have used DNA barcoding in several biodiversity studies in Southeast Asia (Wilson *et al.*, 2016) covering a wide range of animal groups including butterflies (Jisming-See *et al.*, 2016), bats (Syaripuddin *et al.*, 2014), sandflies (Polseela *et al.*, 2016; Sukantamala *et al.*, 2017) and dragonflies (Casas *et al.*, 2017). Two illustrative examples are provided below.

Butterflies of Setiu Wetlands, Malaysia

As part of the Setiu Wetlands Scientific Expedition organized by WWF-Malaysia in 2016, we conducted a survey of butterflies at Lata Changkah, Terengganu, Malaysia. All sampled butterflies were brought back to the laboratory and subjected to standard molecular methods for DNA barcoding (Wilson, 2012; Jisming-See *et al.*, 2016) and bioinformatics methods described above (see Sections Editing Sanger Sequences and Sequence Alignment). The DNA barcodes were submitted to BOLD where they are publicly available under the project code: SETIU. Species assignments were obtained on basis of >98% sequence similarity with records in BOLD (see Section BOLD identification engine), which was possible due to the existing DNA barcode reference library for the butterflies of Peninsular Malaysia (Wilson *et al.*, 2013). However, a strict tree-based criterion (Wilson *et al.*, 2011) was used to assign three DNA barcodes that did not share >98% similarity with any BOLD records to genera. Forty-nine species were recorded suggesting Lata Changkah can currently support rare butterflies (Sing and Wilson, 2017).

Sandflies of Northern Thailand

We collected sandflies using CDC light traps from five tourist caves in Northern Thailand. Specimens were brought back to the laboratory and subjected to standard molecular methods for DNA barcoding (Wilson, 2012; Jisming-See *et al.*, 2016) and bioinformatics methods described above (see Sections Editing Sanger Sequences and Sequence Alignment). We combined our new DNA barcodes with selected publicly available DNA barcodes in BOLD, which included species reported from Thailand (includes records from Thailand, India, and Sri Lanka). The combined dataset (394 DNA barcodes) is publicly available in BOLD under the code: DS-SFNTH. Using a combination of methods (see Sections BOLD Barcode Index Number (BIN) system and Neighbor-Joining (NJ) in MEGA), the specimens were clustered into 34 OTU. Several of the taxa thought to be present in multiple caves, based on morphospecies sorting, split into cave-specific OTU which likely represented cryptic species. The resulting species checklist and DNA barcode library contributed to a growing set of records for sandflies which is useful for monitoring and vector control (Polseela *et al.*, 2016; Sukantamala *et al.*, 2017).

Discussion

DNA barcoding is being used by researchers across an increasing number of biological fields reflecting the fact that DNA sequence information can be cheap and easy to obtain and can enable assignment of taxonomic names to organisms without requiring researchers to be familiar with intricate morphological features. Likewise, molecular OTU can be suitable surrogates for partitioning diversity into interoperable units for biodiversity studies enabling researchers to obtain taxonomic data much faster than possible with traditional morphological approaches, making studies scalable across much larger taxonomic groups and wider geographical regions (Wilson *et al.*, 2017). Yet, the prospect of DNA barcoding can be daunting for beginners (Wilson *et al.*, 2016). Mastering a basic bioinformatics workflow is essential to ensure the quality and reliability of data and to generate meaningful results (Wilson and Sing, 2013).

Future Directions

High-throughput sequencers are replacing Sanger sequencers in most molecular applications. The development of the sub-discipline of DNA metabarcoding grew directly from the major advantages offered by high-throughput sequencing through circumventing the sorting and isolation of the thousands of individuals in bulk mixed samples of organisms (Brandon-Mong *et al.*, 2015). However, the short read lengths and high error rates limited the use of high-throughput sequencers for conventional, individual specimen, DNA barcoding (Hebert *et al.*, 2017), and in particular, for DNA barcode reference library construction (Liu *et al.*, 2017). The continued reliance on Sanger sequencing has constrained reductions in the cost of DNA barcoding and led to uneven DNA barcoding efforts around the world (Liu *et al.*, 2017). Recently developed approaches using the latest generation of high-throughput sequencing platforms (Pacific Biosciences SEQUEL and Illumina HiSeq 4000) have produced full length DNA barcodes of equivalent length and quality to those generated by Sanger sequencing, but with substantially reduced costs of 10-fold (Liu *et al.*, 2017) to 40-fold (Hebert *et al.*, 2017). Bioinformatics pipelines to complement these new approaches have developed concurrently. The mBRAVE webserver developed by the team behind BOLD, and with direct links to the BOLD reference libraries (Hebert *et al.*, 2017) is an important development and likely represents a landmark shift in standard DNA barcoding protocols.

Closing Remarks

In this article, we provide beginners with step-by-step instructions for converting raw DNA sequences into clean DNA barcodes (sequence editing, sequence alignment), to commonly used tools for assigning taxonomic names to DNA barcodes, and to cluster DNA barcodes into OTU. As more researchers become comfortable with such bioinformatics workflows, and the DNA barcoding community continues to grow, essential questions for society: “What is this specimen on an agricultural shipment?”, “Who eats whom in this whole food web?”, and even “How many species are there?” become answerable (Adamowicz, 2015). It is promising

that capacity for DNA barcoding is growing in the parts of the world where it is needed most, particularly among the younger generation of researchers who can easily connect with the barcoding analogy (Adamowicz, 2015; Wilson *et al.*, 2016).

Acknowledgements

Kong-Wah Sing is supported by the Chinese Academy of Sciences President's International Fellowship Initiative. We thank the BOLD team, especially Megan Milton, for their continuous support of our DNA barcoding workshops in Southeast Asia. We are grateful to CodonCode Corporation for supplying teaching licenses during our workshops. We thank previous sponsors and hosts of our DNA barcoding workshops: the Centre of Excellence in Fungal Research, the Department of Microbiology and Parasitology, and the Faculty of Medical Science at Naresuan University, Phitsanulok, Thailand; the Zoological and Ecological Research Network, and Museum of Zoology at the University of Malaya, Kuala Lumpur, Malaysia; the University of Nottingham Malaysia Campus, Selangor, Malaysia; Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia; and the Asia-Pacific Network for Global Change Research. We also thank the scientists who have helped facilitate our workshops: Paul Hebert, Brandon Mong Guo Jie, Lee Ping Shin, Evan Chin, Kharunnisa Syaripuddin, Jedsada Sukantamala, Cheah Men How, Siti Azizah M Nor, Noor Adelyna M Akib, Mr Foo, Mr Chin, Elizabeth Clare.

Appendix A Supplementary Information

Supplementary data associated with this article can be found in the online version at [10.1016/B978-0-12-809633-8.20468-8](https://doi.org/10.1016/B978-0-12-809633-8.20468-8).

See also: Computational Pipelines and Workflows in Bioinformatics. Natural Language Processing Approaches in Bioinformatics

References

- Adamowicz, S.J., 2015. International barcode of life: Evolution of a global research community. *Genome* 58, 151–162.
- Brandon-Mong, G.J., Gan, H.M., Sing, K.W., *et al.*, 2015. DNA metabarcoding of insects and allies: An evaluation of primers and pipelines. *Bulletin of Entomological Research* 105, 717–727.
- Casas, P.A.S., Sing, K.W., Lee, P.S., *et al.*, 2017. DNA barcodes for dragonflies and damselflies (Odonata) of Mindanao, Philippines. *Mitochondrial DNA Part A*, Online Early. doi:10.1080/24701394.2016.1267157.
- Hebert, P.D.N., Braukmann, T.W.A., Prosser, S.W.J., *et al.*, 2017. A sequel to sanger: Amplicon sequencing that scales. *bioRxiv*. 191619. Available at: <https://doi.org/10.1101/191619>.
- Jisming- See, S.W., Sing, K.W., Wilson, J.J., 2016. DNA barcodes and citizen science provoke a diversity reappraisal for the “ring” butterflies of Peninsular Malaysia (*Ypthima* satyrinae: Nymphalidae: lepidoptera). *Genome* 59, 879–888.
- Liu, S., Yang, C., Zhou, C., Zhou, X., 2017. Filling reference gaps via assembling DNA barcodes using high-throughput sequencing-moving toward barcoding the world. *Gigascience* 6, 1–8.
- Polseela, R., Jaturas, N., Thanwisai, A., Sing, K.W., Wilson, J.J., 2016. Towards monitoring the sandflies (Diptera: psychodidae) of Thailand: DNA barcoding the sandflies of Wiha Cave, Uttaradit. *Mitochondrial DNA Part A* 27, 3795–3801.
- Pons, J., Barraclough, T.G., Gomez-Zurita, J., *et al.*, 2006. Sequence-based species delimitation for the DNA taxonomy of undescribed insects. *Systematic Biology* 55, 595–609.
- Puillandre, N., Lambert, A., Brouillet, S., Achaz, G., 2012. ABGD, Automatic barcode gap discovery for primary species delimitation. *Molecular Ecology* 21, 1864–1877.
- Ratnasingham, S., Hebert, P.D.N., 2013. A DNA-based registry for all animal species: The Barcode Index Number (BIN) system. *PLOS ONE* 8, e66213.
- Schmieder, R., Edwards, R., 2011. Quality control and preprocessing of metagenomic datasets. *Bioinformatics* 27, 863–864.
- Sing, K.W., Syaripuddin, K., Wilson, J.J., 2016. How to rapidly accelerate biodiversity inventory in places where most of the species are unknown? *Malayan Nature Journal* 68, 131–134.
- Sing, K.W., Wilson, J.J., 2017. Butterfly diversity at a recreation hotspot in Setiu Wetlands, Terengganu, Malaysia. *Prosiding Seminar Ekspedisi Saintifik Tanah Bencah Setiu 2016*. Selangor: WWF-Malaysia. pp. 86–96.
- Sukantamala, J., Sing, K.W., Jaturas, N., Polseela, R., Wilson, J.J., 2017. Unexpected diversity of sandflies (Diptera: psychodidae) in tourist caves in Northern Thailand. *Mitochondrial DNA Part A* 28, 949–955.
- Syaripuddin, K., Kumar, A., Sing, K.W., *et al.*, 2014. Mercury accumulation in bats near hydroelectric reservoirs in Peninsular Malaysia. *Ecotoxicology* 23, 1164–1171.
- Tamura, K., Stecher, G., Peterson, D., Filipi, A., Kumar, S., 2013. MEGA6: Molecular evolutionary genetics analysis Version 6.0. *Molecular Biology and Evolution* 30, 2725–2729.
- Wilson, J.J., 2012. DNA barcodes for insects. In: Kress, W.J., Erikson, D.L. (Eds.), *DNA Barcodes: Methods and Protocols*. New York: Humana Press.
- Wilson, J.J., Rougerie, R., Shonfeld, J., *et al.*, 2011. When species matches are unavailable are DNA barcodes correctly assigned to higher taxa? An assessment using sphingid moths. *BMC Ecology* 11, 18.
- Wilson, J.J., Sing, K.W., 2013. DNA barcoding can successfully identify *Penaues monodon*, associate life cycle stages, and generate hypotheses of unrecognised diversity. *Sains Malaysiana* 42, 1827–1829.
- Wilson, J.J., Sing, K.W., Floyd, R.M., Hebert, P.D.N., 2017. DNA barcodes and insect biodiversity. In: Footitt, R.G., Adler, P.H. (Eds.), *Insect Biodiversity: Science and Society*, second ed. Oxford: Blackwell Publishing Ltd, pp. 575–592.
- Wilson, J.J., Sing, K.W., Lee, P.S., Wee, A.K.S., 2016. Application of DNA barcodes in wildlife conservation in Tropical East Asia. *Conservation Biology* 30, 982–989.
- Wilson, J.J., Sing, K.W., Sofian-Azirun, M., 2013. Building a DNA barcode reference library for the true butterflies (Lepidoptera) of Peninsula Malaysia: What about the subspecies? *PLOS ONE* 8, e79969.
- Zhang, J., Kapli, P., Pavlidis, P., Stamatakis, A., 2013. A general species delimitation method with applications to phylogenetic placements. *Bioinformatics* 29, 2869–2876.

Further Reading

Adamowicz, S.J., Chain, F.J.J., Clare, E.L., *et al.*, 2016. From barcodes to biomes: Special issues from the 6th international barcode of life conference. *Genome* 59, v–ix.

Kress, W.J., Erikson, D.L. (Eds.), 2012. *DNA Barcodes: Methods and Protocols*. New York: Humana Press.

Relevant Websites

<http://www.wabi.snv.jussieu.fr/public/abgd/>
ABGD.

www.boldsystems.org
Barcode of Life Datasystems (BOLD).

www.mbio.ncsu.edu/bioedit/bioedit.html
BIOEDIT.

<http://species.h-its.org/ptp/>
bPTP.

www.codoncode.com
CODONCODE ALIGNER.

www.barcodingasia.weebly.com
DNA Barcoding Workshops (DBW).

www.usegalaxy.org
GALAXY.

<https://blast.ncbi.nlm.nih.gov/Blast.cgi>
GenBank BLAST.

<http://www.megasoftware.net/>
Molecular Evolutionary Genetics Analysis (MEGA).

www.prinseq.sourceforge.net
PRINSEQ.

Text Mining Applications

Raul Rodriguez-Esteban, F. Hoffmann-La Roche Ltd., Basel, Switzerland

© 2019 Elsevier Inc. All rights reserved.

Glossary

Co-reference References in a text that allude to the same thing using different names. Examples of co-references can be pronouns or abbreviations.

Corpus Collection of documents.

Manual curation Review and evaluation of data or information by humans.

Natural language Language that has evolved organically and thus has not been formally designed. Most spoken human languages are natural languages.

Pharmacovigilance Tracking of potential adverse events associated to marketed drugs.

Introduction

Biomedical text mining (henceforth, text mining) is primarily an applied scientific discipline. An important focus of text mining research is to increase the efficiency and efficacy of everyday biomedical work requiring the analysis of textual information. Due to the pervasiveness of text in the biomedical realm, text mining has applications in numerous settings, such as medical facilities, pharmaceutical and academic research institutions, etc. In each of these settings, however, there is interest in different questions, which can only be answered through the information contained in particular types of textual sources. However, each type of question and textual source brings its own challenges that necessarily condition the text mining tools that are used to answer them. Thus, text mining applications can be broadly organized around both the types of questions and the sources of text that are utilized.

Sources of Text

There is considerable variation in the writing style used within different textual sources. Notable differences can be seen between text mining applications for clinical settings, which deal with medical records, and those geared toward scientific text from scientific articles or grants. Similarly, text mining applications for social media or patents occupy their own niches. The reason for this is that each of these textual sources follows grammatical and semantic rules that differentiate them. These differences cause performance degradation in text mining tools that are applied to types of text that are different from the ones they were originally designed for. For example, patents may contain complicated claim dependencies (i.e., patent claims that make reference to other claims within the patent) that do not fit the natural language patterns that appear in other types of text. Another example of unusual writing style comes from social media, which is often ungrammatical and built with its particular contextual cues. It is worthy, thus, to discuss some of the different sources of text used in text mining before delving into particular applications.

Before discussing types of textual sources it is important to briefly point out the role that availability plays. Cost and privacy issues can be significant hurdles to access valuable biomedical information, thereby affecting the development of text mining applications. Free, electronic and easily downloadable content can more easily enable the development of such applications. With that in mind some relevant types of text sources are:

- Scientific abstracts: the main freely available sources are Medline and Europe PubMed Central (Europe PMC).
- Scientific full text articles: a growing subset of Open Access full text articles is available from PMC and Europe PMC.
- Patents: US patents are provided in bulk for free by the United States Patent and Trademark Office (USPTO). Other patent authorities typically require a fee for bulk electronic access.
- Grants: the main sources are the National Institutes of Health (NIH) RePORTER in the United States and the European Bioinformatics Institute (EBI) Grist database in Europe.
- Clinical trial records: ClinicalTrials.gov, the World Health Organization (WHO), and International Clinical Trials Registry Platform (ICTRP) are extensive free repositories that can be downloaded in bulk.
- Medical records and clinical notes: due to privacy concerns only limited samples are freely available, such as the i2b2 corpus.
- Pharmacovigilance and safety: pharmacovigilance records are available from the US Food and Drug Administration (FDA) FAERS and European Molecular Biology Laboratory (EMBL) SIDER databases. US drug labels are available from FDA's DailyMed.
- Social media: multiple blogs, microblogs, and forums can be downloaded from their internet locations. However, most of these sources require customized downloading and processing.
- News: freely available RSS feeds from news services, news agencies and newspapers provide an easy way to access current news content. Access to archival news is more limited (Breiner and Rodriguez-Esteban, 2012).

Documents coming from each of these sources are split into different sections, each with its own properties. For example, the abstract of a scientific article has a different structure, style, and content than the full text (Martin *et al.*, 2004; Schuemie *et al.*, 2004; Cohen *et al.*, 2010). Moreover, within document sections one can find structures, such as tables and figures, which require specialized processing.

Building Blocks

Text mining applications are often driven by the capabilities of the algorithms that have been the focus of most research attention in text mining. Such algorithms have both enabled and limited the applications that are currently possible. Thus, we may call these algorithms the “building blocks” of text mining applications. Where adequate algorithms do not exist, or yield low performance, applications become unfeasible. Examples of applications that have had very limited success for lack of effective algorithms are synonym detection (Cohen *et al.*, 2005), co-reference resolution (Kim *et al.*, 2012), and question answering (QA) (Yang *et al.*, 2014). Thus, any discussion on text mining applications needs to make a reference to the possibilities that are enabled by the algorithms that have been successfully developed thus far.

Some of these building blocks are:

1. **Named-entity recognition (NER).** One of the most studied areas in text mining is NER. NER consists in identifying specific types of names and other entities written in text. NER in biomedicine is faced with a particular set of challenges because some types of names that are of interest in biomedicine have large, complex vocabularies and can be uncommon in other realms. Examples of these are the names of diseases, genes, chemicals, drugs, mutations, cell types, and species. The precise identification of such names enables the extraction of other, more complicated facts. Therefore, high-quality NER is a prerequisite for many other text mining algorithms. Identifying such names by solely using lists of synonyms (a.k.a. concept recognition) does not often deliver results of acceptable quality (see Section “Evaluation Metrics”). Thus, there has been significant amount of work in machine learning algorithms to improve NER.
2. **Relation extraction.** A step further from NER is finding relations between pairs of entities. Many descriptions of biological and chemical phenomena involve multiple entities, such as the descriptions of signaling pathways, biomarkers, mutation–disease links, and drug–drug interactions. Descriptions of such phenomena may be simplified into a set of relations between pairs of entities. For example, a signaling cascade may be described by a set of individual protein-binding interactions. While such type of representation might be too simple for some applications it can still be useful to highlight relevant facts that can then be further explored.
3. **Event extraction.** In order to tackle more complex biological phenomena that may not be described adequately with relations, text miners have come up with the task of “event” extraction. Within this task an event is considered rather generically as a molecular status or change of status described in text (Kim *et al.*, 2008). An example of molecular event could be a translocation or a change in gene expression. Events can be multifaceted and conditioned by temporal and contextual parameters, which complicate their identification and the extraction of their details. Thus, more complex extraction and representation approaches are necessary to capture them and their different types of dependencies. Not surprisingly, the more complex the events that need to be extracted the more challenging it is for text mining algorithms. Thus, NER and relation extraction algorithms have achieved higher performance than those algorithms specialized in event extraction.
4. **Document classification and clustering.** Assigning a document to a specific class (classification) or grouping documents according to their similarity (clustering) is useful for certain text mining applications, such as for organizing and exploring the scientific landscape around a certain topic. It is also particularly useful for the task of selecting documents that can then be further analyzed or mined. Thus, an initial classification or clustering can reduce the work or improve the quality involved in later processing steps. This can be considered a “filtering” of documents. For example, due to the cost associated to accessing the full text of scientific articles, documents can be filtered first by the content of their freely available abstracts before accessing them.
5. **Document segmentation and zone detection.** As mentioned, many documents are divided into certain standard sections. For example, scientific articles are often organized into introduction, methods, results and discussion (IMRaD). Concentrating the focus of an application on particular sections can improve the quality of the results. However, the content of a document may not be appropriately located in the correct sections (Agarwal and Yu, 2009). Thus, recognizing the different segments and discursive zones of a document can help in its analysis. For example, by identifying the zones of a patent that deal with experimental setups one may get a better idea of the results that are being reported in the patent (Rodriguez-Esteban and Bundschuh, 2016).
6. **Co-reference resolution.** Generally, co-references (see Glossary) are challenging to mine, especially those that refer to something mentioned in another sentence in the text. On the other hand, a special case of co-reference that is easy to resolve is the abbreviation. Algorithms to map abbreviations to their definitions have been highly successful (Wren and Garner, 2002; Pustejovsky *et al.*, 2001), because abbreviations are often defined in the text in typical ways, where the abbreviation is followed by the definition in parentheses, for example, natural language processing (NLP).

Evaluation Metrics

It is important to consider the metrics used to rate the performance of text mining applications in order to be able to assess the quality of the results they provide. These metrics are similar to those used in information retrieval (IR). Two of the metrics are the counts of type I errors, known as false positives (FPs), and type II errors, known as false negatives (FNs). FPs represent items that

are incorrectly included in the results, while FNs are items that are incorrectly missing from the results. Moreover, true positives (TPs) are items that appear correctly in the results and true negatives (TNs) items that, correctly, do not appear in the results. Because in IR the number of TNs is often hard to quantify, or even conceptualize, performance metrics are based on TPs, FPs, and FNs. Two typical metrics that derive from them are precision and recall:

$$\text{precision} = \frac{TP}{TP + FP}, \text{ recall} = \frac{TP}{TP + FN}$$

Even though, precision and recall are widely used, finding a single metric has been always of interest because it makes comparisons easier across text mining applications. Thus, there are single metrics that have been devised to encapsulate the value of precision and recall, the most typical of them being the F-measure. The F-measure is the harmonic mean of precision and recall:

$$F = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Even though, it is the most widely used single metric, the F-measure has flaws and several alternative measures have been proposed, such as area under the curve (AUC) and overhead (Rodríguez-Esteban, 2009, 2015).

Another aspect to consider is that the use of multiple text mining applications to solve a problem can lead to an overall reduced performance as errors are compounded. As can be seen in Fig. 1, setting several text mining applications in series produces a degradation of performance as FNs and FPs accumulate at every step. Thus, increases in complexity of a text mining application may lead to results of lower quality, which in the end hinders our ability to identify complex phenomena in text.

Applications

Biomedical questions that can be answered using textual information are the engines behind the development of text mining applications. Such questions are context dependent because they change according to the settings and the textual sources available. Some of the questions that can be addressed with text mining are:

1. Questions that can be answered with relations. Questions that can be answered by mining relations are common in multiple biomedical contexts. Therefore, extracting relations is a major focus of text mining, especially for molecular-level relationships. In particular, the relations that have drawn the most attention have been the so-called protein–protein interactions (PPIs), which involve any relations between pairs of proteins or genes, whether direct or indirect (thus, not just protein binding). PPIs are useful to build signaling pathways and describe other phenomena, such as gene–gene co-expression. Another application based on extracting relations is the identification of biomarkers. There are several types of biomarkers (e.g., mechanistic vs. descriptive biomarkers, efficacy biomarkers, safety biomarkers). They typically involve a surrogate measurement of a biological effect that cannot be measured directly or it is costly to do so. Such surrogate measurements can be extracted from relationships in text, for example, by identifying downstream proteins in a pathological pathway or by extracting protein–metabolite relations. In addition to extraction of PPIs, many other applications are based on identifying relations. Identifying drug–drug relations can be helpful in clinical settings to discern pernicious drug interactions. Gene–disease and mutation–disease relations can lead to the identification of pharmaceutical targets, especially in cases in which the relation is causal. A subset of causal gene–disease relations, for example, are called perturbations, which are gene modifications that have been shown experimentally to improve or worsen a disease phenotype (Rodríguez-Esteban *et al.*, 2009). In the realm of pharmacovigilance, signal detection, which involves the flagging of potential drug-adverse event relations, is of great importance (Harpaz *et al.*, 2014). Another area of interest relates to pharmacogenomics, which is concerned with gene–compound relations (Garten and Altman, 2009).
2. Biological processes beyond relations. A relation is often an overtly simplified representation of a biological phenomenon. Events, on the other hand, can encompass even fairly complex descriptions of biological processes. Example applications that are based on extracting such biological processes are gene regulation, gene transcription, gene-variant-disease relations, protein catabolism, and phosphorylation. Protein phosphorylation, for example, may involve the identification of a kinase,

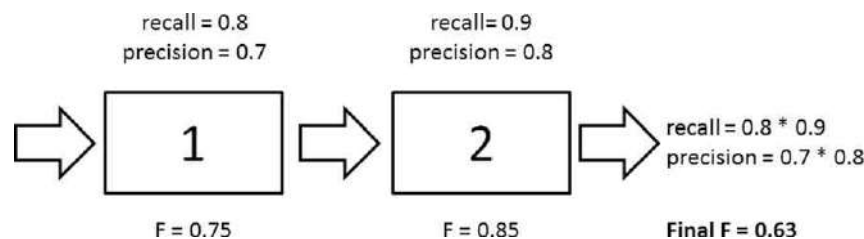


Fig. 1 Decreasing performance of serial text mining applications. When the input text needs to be processed by more than one text mining application the performance degrades.

- a substrate and a phosphorylation site (Hu *et al.*, 2005). Gene regulation, on the other hand, may involve the identification of genes that are expressed in particular cell types or anatomic locations (Gerner *et al.*, 2010). Gene-variant-disease relations present the challenge of identifying the gene associated to a specific genetic variant such as a mutation (Lee *et al.*, 2016).
3. Biomedical QA (BioQA). QA entails the answering of a natural-language question by seeking an adequate fragment of text that can function as an answer or that provides a fact that answers the question. QA is typically composed of two parts: (1) understanding the question being posed and (2) finding the fragment of text that correctly answers the question. In the biomedical setting, QA (or BioQA) can be used toward addressing questions that appear in real time, for example, questions from medical practitioners in clinical settings. Despite its potential usefulness, BioQA applications have not yet shown a level of performance that enables their practical use in such settings.
 4. Semi-automated curation. Many biological databases use text mining to filter documents or text snippets before they are evaluated by human curators and added to the database (Rodríguez-Esteban, 2015). This can significantly reduce the amount of curation effort that is spent while minimizing the number of relevant information that is missed through this filtering. The goal is to apply filters that have high recall and moderate precision. High recall ensures that there is a small number of FNs, while moderate precision leads to a reduction in the amount of items that need to be curated by avoiding potential FPs. Besides biological databases, text mining for semi-automated curation can be used for other tasks, such as for writing systematic reviews (O'Mara-Eves *et al.*, 2015). Graphical applications that enable semi-automated curation have been developed for many different curatorial settings (Rinaldi *et al.*, 2014).
 5. Biological similarity. All the text written about a biological entity can be used as a fingerprint of said entity. Because such text covers the knowledge that we have about the entity it can be used to make a number of predictions and comparisons. For example, all the documents mentioning a protein can be used together to predict the protein's function (Verspoor, 2014), its subcellular localization (Shatkay *et al.*, 2007), or its structure (Koussounadis *et al.*, 2009). Another example is similarity across diseases, which can be computed by analyzing the documents that mention each disease.
 6. Knowledge discovery. One can connect facts that have been extracted from different documents through the use of text mining. Because scientists are unable to follow all publications even in their own field they are prone to miss facts that could be relevant to their research. Thus, finding pairs or sets of related facts that are present in different publications ("connecting the dots") can lead to the discovery of knowledge that had not been evident to the individual scientists that contributed the facts. This way, for example, one may be able to explain the mechanism leading to certain adverse drug effects (Hristovski *et al.*, 2016) or anticipate future discoveries (Frijters *et al.*, 2010).
 7. Pharmacovigilance. Identifying reports of potential adverse events associated to specific drugs can be very time consuming, requiring the work of multiple human curators. Moreover, social media is becoming an increasingly relevant source of adverse event reports. Thus, there is a need to filter all the documents potentially associated to drug-adverse events to focus solely on those that are more likely to be relevant. Such filtering needs to be done carefully to minimize the number of FNs.
 8. Figure mining. Figures are integrated within many textual sources, such as patents and scientific articles. Thus, analyzing the content of figures requires mixing textual and image analysis. Applications of figure mining include figure classification (Rodríguez-Esteban and Iossifov, 2009) and linking an article's figures to text fragments within the article that explain the figure (Yu *et al.*, 2009).
 9. Landscaping and tracking biomedical knowledge. Following and landscaping biomedical trends can be useful to identify emerging technologies, areas of high research impact (Cokol and Rodríguez-Esteban, 2008; Cokol *et al.*, 2007), scientific trends (Rodríguez-Esteban and Loging, 2013) and novel discoveries, such as new pharmaceutical targets. Landscaping is of particular interest in patent mining, where it can be used to detect technological areas with low level of patenting (white spots) or to craft research strategies.
 10. Pharmacokinetics and systems biology modeling. Measurements and parameters that can be used for systems biology or pharmacokinetics modeling are difficult to identify using traditional search engines. Such numerical information can be better identified, for example, through rule-based approaches. Even more challenging is to associate these values to the relevant entity they describe. For example, to associate a protein concentration measurement to the protein that was measured.
 11. Enrichment of genomic analyses. Typically, genomic data, coming from high-throughput experiments, such as genome-wide association study (GWAS), pharmacogenomics, microarray, and RNAseq, can produce long lists of genes that need to be prioritized or grouped to further analyze their function or to decide on follow-up experiments. Text mining can help manage such lists in a high-throughput manner by parsing the literature associated to each gene (Ailem *et al.*, 2016) and providing an overall picture, for example, of the pathways and functions most often associated to a group of genes of interest. Such text mining results can be complemented with data from biological databases to create an integrated approach.

Conclusions

There are multiple ways in which text mining applications can impact biomedical work. A few have been discussed here but more can be explored within the text mining literature. See the Further Reading section for additional information on this topic.

See also: Biological Database Searching. Data-Information-Concept Continuum From a Text Mining Perspective. Natural Language Processing Approaches in Bioinformatics. Text Mining Basics in Bioinformatics. Text Mining for Bioinformatics Using Biomedical Literature

References

- Agarwal, S., Yu, H., 2009. Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics* 25, 3174–3180.
- Ailem, M., Role, F., Nadif, M., *et al.*, 2016. Unsupervised text mining for assessing and augmenting GWAS results. *Journal of Biomedical Informatics* 60, 252–259.
- Breiner, D.A., Rodriguez-Esteban, R., 2012. What's in the News? Web Scraping Technology as a Cost-Effective Solution for News Alerting. *Pharma-Bio-Med*, Lisbon, Portugal, September 2012.
- Cohen, A.M., Hersh, W.R., Dubay, C., *et al.*, 2005. Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts. *BMC Bioinformatics* 6, 103.
- Cohen, K.B., Johnson, H.L., Verspoor, K., *et al.*, 2010. The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics* 11, 492.
- Cokol, M., Rodriguez-Esteban, R., 2008. Visualizing evolution and impact of biomedical fields. *Journal of Biomedical Informatics* 41, 1050–1052.
- Cokol, M., Rodriguez-Esteban, R., Rzhetsky, A., 2007. A recipe for high impact. *Genome Biology* 8, 406.
- Frijters, R., van Vugt, M., Smeets, R., *et al.*, 2010. Literature mining for the discovery of hidden connections between drugs, genes and diseases. *PLOS Computational Biology* 6 (9), Available at: <http://dx.doi.org/10.1371/journal.pcbi.1000943>
- Garten, Y., Altman, R.B., 2009. Pharmspresso: A text mining tool for extraction of pharmacogenomic concepts and relationships from full text. *BMC Bioinformatics* 10 (Suppl. 2), S6.
- Gerner, M., Nenadic, G., Bergman, C.M., 2010. An exploration of mining gene expression mentions and their anatomical locations from biomedical text. In: *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*.
- Harpaz, R., Callahan, A., Tamang, S., *et al.*, 2014. Text mining for adverse drug events: The promise, challenges, and state of the art. *Drug Safety* 37, 777–790.
- Hristovski, D., Kastrin, A., Dinevski, D., *et al.*, 2016. Using literature-based discovery to explain adverse drug effects. *Journal of Medical Systems* 40, 185.
- Hu, Z.Z., Narayanaswamy, M., Ravikumar, K.E., *et al.*, 2005. Literature mining and database annotation of protein phosphorylation using a rule-based system. *Bioinformatics* 21, 2759–2765.
- Kim, J.D., Nguyen, N., Wang, Y., *et al.*, 2012. The genia event and protein coreference tasks of the BioNLP shared task 2011. *BMC Bioinformatics* 13 (Suppl. 11), S1.
- Kim, J.D., Ohta, T., Tsujii, J., 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics* 9, 10.
- Koussounadis, A., Redfern, O.C., Jones, D.T., 2009. Improving classification in protein structure databases using text mining. *BMC Bioinformatics* 10, 129.
- Lee, K., Lee, S., Park, S., *et al.*, 2016. BRONCO: Biomedical entity Relation ONcology COrpus for extracting gene-variant-disease-drug relations. *Database (Oxford)*. Available at: <http://dx.doi.org/10.1093/database/baw043>
- Martin, E.P.G., Bremer, E.G., Guerin, M., DeSesa, C., Jouve, O., 2004. Analysis of protein/protein interactions through biomedical literature: Text mining of abstracts vs. text mining of full text articles. In: López, J.A., Benfenati, E., Dubitzky, W. (Eds.), *Knowledge Exploration in Life Science Informatics. Lecture Notes in Computer Science*. Berlin; Heidelberg: Springer, pp. 96–108.
- O'Mara-Eves, A., Thomas, J., McNaught, J., *et al.*, 2015. Using text mining for study identification in systematic reviews: A systematic review of current approaches. *Systematic Reviews* 4, 5.
- Pustejovsky, J., Castaño, J., Cochran, B., *et al.*, 2001. Automatic extraction of acronym-meaning pairs from MEDLINE databases. *Studies in Health Technology and Informatics* 84 (Pt. 1), 371–375.
- Rinaldi, F., Clematide, S., Marques, H., *et al.*, 2014. OntoGene web services for biomedical text mining. *BMC Bioinformatics* 15, S6.
- Rodriguez-Esteban, R., 2009. Biomedical text mining and its applications. *PLOS Computational Biology* 5, e1000597.
- Rodriguez-Esteban, R., 2015. Biocuration with insufficient resources and fixed timelines. *Database (Oxford)*. Available at: <http://dx.doi.org/10.1093/database/bav116>
- Rodriguez-Esteban, R., Bundschuh, M., 2016. Text mining patents for biomedical knowledge. *Drug Discovery Today* 21, 997–1002.
- Rodriguez-Esteban, R., Iossifov, I., 2009. Figure mining for biomedical research. *Bioinformatics* 25, 2082–2084.
- Rodriguez-Esteban, R., Loging, W.T., 2013. Quantifying the complexity of medical research. *Bioinformatics* 29, 2918–2924.
- Rodriguez-Esteban, R., Roberts, P.M., Crawford, M.E., 2009. Identifying and classifying biomedical perturbations in text. *Nucleic Acids Research* 37, 771–777.
- Schuemie, M.J., Weeber, M., Schijvenaars, B.J., *et al.*, 2004. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics* 20, 2597–2604.
- Shatkay, H., Höglund, A., Brady, S., *et al.*, 2007. SherLoc: High-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics* 23, 1410–1417.
- Verspoor, K.M., 2014. Roles for text mining in protein function prediction. *Methods in Molecular Biology* 1159, 95–108.
- Wren, J.D., Garner, H.R., 2002. Heuristics for identification of acronym-definition patterns within text: Towards an automated construction of comprehensive acronym-definition dictionaries. *Methods of Information in Medicine* 41, 426–434.
- Yang, Z., Li, Y., Cai, J., *et al.*, 2014. QUADS: Question answering for decision support. In: *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'14)*, pp. 375–384.
- Yu, H., Agarwal, S., Johnston, M., *et al.*, 2009. Are figure legends sufficient? Evaluating the contribution of associated text to biomedical figure comprehension. *Journal of Biomedical Discovery and Collaboration* 4, 1.

Further Reading

- Rebholz-Schuhmann, D., Oellrich, A., Hoehndorf, R., 2012. Text-mining solutions for biomedical research: Enabling integrative biology. *Nature Reviews Genetics* 13, 829–839.
- Rodriguez-Esteban, R., 2009. Biomedical text mining and its applications. *PLOS Computational Biology* 5, e1000597.
- Rodriguez-Esteban, R., 2016. Understanding human disease knowledge through text mining: What is text mining? In: Loging, W. (Ed.), *Bioinformatics and Computational Biology in Drug Discovery and Development*. Cambridge: Cambridge University Press.
- Rodriguez-Esteban, R., 2016. Appendix. I. Additional knowledge-based analysis approaches. In: Loging, W. (Ed.), *Bioinformatics and Computational Biology in Drug Discovery and Development*. Cambridge: Cambridge University Press.
- Rzhetsky, A., Seringhaus, M., Gerstein, M., 2008. Seeking a new biology through text mining. *Cell* 134, 9–13.

Relevant Websites

<http://bionlp.org/>
 BioNLP.org.
<https://www.i2b2.org/NLP/DataSets/Main.php>
 i2b2.

Note

Page numbers suffixed by 'F', 'T', and 'B' refer to Figures, Tables, and Boxes, respectively.

A

- A-Brujn alignment method 2:271
- A-DNA 3:505
- AAARF algorithm 2:215
- ab initio*
 - gene prediction 2:195, 2:195–198
 - methods 2:182–183, 2:214, 2:475, 2:540, 3:253–254, 3:261, 3:522–523
 - prediction methods 3:233
 - protein structure prediction 1:63, 1:63–64, 1:71T, 1:261, 2:452
- AB SOLiD System 1:128–129
- abiogenesis 3:938
- ABO grouping system 3:363
- absolute error 1:622
- absolute quantification 3:872
- absorption, distribution, metabolism, excretion, and toxicity (ADMET) 2:684, 2:1093, 3:275, 3:750, 3:753
- absorption, distribution, metabolism and excretion (ADME) 1:333, 1:912
- ADME-Tox 3:66
- Abstract Index Broker class 1:884–885
- Abstract Search Engine Similarity Measure 1:885
- ABYSS 3:304
- ACB database 2:1112
- access confidentiality 1:267
- access level modifier 1:206
- access linkage problem 1:267–269, 1:268–269, 1:269–270
- access modes 1:841–842
- accessible surface area (ASA) 2:145–146, 2:447, 2:449F, 3:682
- accountability 1:254
- accuracy (Ac) 1:686, 3:5, 3:9T, 3:682–683
 - application to imbalanced classification 1:688
 - application to multi-class domains 1:686–688
 - confusion matrix 1:685–686
 - measures for comparing structures 1:69–70
 - relevant statistical tests for significance of biological results 1:688–689
- accuracy self estimate score (ASE Score) 1:71
- AceCloud 1:263
- Acellera 1:263
- acetylation 2:15, 3:741
- acidic amino acids 3:511
- acidophiles 3:945
- acquired immune deficiency syndrome (AIDS) 2:986, 3:706
- acuity UPLC system 3:472
- activating domain (AD) 2:835, 3:933
- activation functions 1:640–641
- activation induced cytosine deaminase (AID) 2:942–943
- activators 2:142
- active contour model 2:1032
- active demethylation 3:341
- active learning method 1:570
- active site 3:504, 3:516
- activity flow variant (AF variant) 2:887
- acute myeloid leukemia (AML) 3:170–171
- ad hoc retrieval 2:1101–1102
- Ad-Hoc/Shell Scripts 3:136
- Adagrad 1:640
- adaptability study of periplasmic binding proteins 3:636–638
- ADAPTABLE trial 1:155–156
- adapter-removing tools 3:296
- adaptive boosting algorithm (Adaboost algorithm) 1:531, 1:533–534, 1:533F, 1:534F, 1:539
- adaptive immune system 2:906–908, 2:931
- adaptive learning rates 1:640
- adaptive linear element (ADALINE) 1:621–623, 1:622F, 1:622–623, 1:623, 1:623–626
- adaptive sampling methods 1:466
- adaptive seeds 2:271–272
- Addison's disease 1:817
- adenine 2:701
- adenoma 1:841
- adenosine deaminase acting on RNA (ADAR) 2:328
- adenosine diphosphate (ADP) 2:439
- adenosine monophosphate (AMP) 2:439
- adenosine triphosphate (ATP) 2:439, 2:867, 3:927–928
 - ATP-dependent nucleosome repositioning 2:314
 - ATP-driven molecular machine 2:304
 - TF-mediated recruitment of ATP-dependent enzymes 2:314
- ADHDGene 2:1050–1051
- adjacency matrix of graph 1:925
- adjusted p-values 1:695, 1:696, 3:309
- adjusted rand index (ARI) 1:445, 1:451, 1:453F
- ADMIXTOOLS package 3:369
- ADMIXTURE 3:367
- admixture dating methods 3:369–370
- advanced analytics 3:67–68
- advanced visual systems (AVS) 2:527–528
- adverse drug reactions (ADRs) 2:634, 2:1047, 2:1052, 3:379, 3:780–781
- adverse outcome pathways (AOP) 3:66
- Aedes aegypti* vectors 3:9–10
- aedes mosquitos 3:975
- Aedes-dengue virus interaction, DSF in studying 3:13
- AfCS database 2:799
- affine gap model, extension to 1:26–27
- affine scoring 2:702
- affinity 1:972
- affinity capillary electrophoresis (ACE) 3:467
- affinity maps 2:692–693
- affinity maturation 2:972
- affinity purification–mass spectrometry (AP-MS) 2:835, 2:836, 2:840–841, 2:849–850, 3:284, 3:934
- "affy" package 2:366–367
- Affymetrix DMET platform 1:333
- Affymetrix exon arrays 2:224
- african clawed frog (*Xenopus laevis*) 2:262
- aGeneApart 1:910
- agent-based modeling and simulation (ABMS) 1:315, 1:317–318
- agent-based models (ABM) 2:878–879
- agent-oriented programming (AOP) 1:318–319
- agent-oriented software engineering (AOSE) 1:315, 1:318–319
- agglomerative approaches 1:442
- agglomerative clustering 1:350
- agglomerative hierarchical clustering 2:390
- aggrecan gene 3:967
- AGX service 1:253
- Akaike's information criterion (AIC) 1:492, 1:734, 2:159, 2:714, 2:984–985
- ALARM network 2:161
- alcohol dehydrogenases evolution (Adh evolution) 2:197, 3:946–947
- algae 3:1
- algorithm for reconstruction of accurate cellular networks (ARACNe) 1:1055
- algorithm(s) 1:1, 1:5
 - complexity 1:1–2
 - iterative 1:2–3
 - recursive 1:3–4
- Alien Index (AI) 3:957
- AlignACE tool 2:120
- "Alignathon" 2:274, 2:275–276, 2:276
- aligners 3:166–167, 3:345
- aligning sequences problem 1:22
- AlignMCL 1:1068

- alignment 1:104, 2:184, 2:273–275, 2:274–275, 3:504, 3:510
 alignment-based methods 3:681
 alignment-free sequence comparison 1:224–226
 experimental analysis 1:226, 1:226F, 1:227F
 file formats 3:298
 objects 1:709
 optimizations 1:225–226
 incremental in-mapper combining 1:226
 input split strategies 1:226
 alkaloid(s) 3:498–499
 data set 3:490
 relations between ring structure and pathways 3:493
 subring skeleton profiling 3:490–491, 3:491–493
 synthesis pathways 3:489–490
 alkalophiles 3:945
 all catfish species inventory (ACSI) 2:1112
 ALLEGRO 2:246–247
 allele-specific expression (ASE) 3:434
 allele-specific hybridization 3:433
 allele-specific oligonucleotide (ASO) 3:433
 allele-specific PCR (AS-PCR) 3:433
 allele-specific single-base primer extension (AS-SBE) 3:433–434
 allelic encoding 2:236
 allelic expression 3:348
 allelic frequency 3:371
 allelic inference 2:946
 allergy 2:908
 allopolyploid 2:261
 allostery 2:509
 allozymes 3:966
 a criterion 1:376
 a diversity 3:18–19
 a globins 3:966
 a-helical structure 3:645–646
 a-helices 3:514
 alphabet 1:15
 AltAnalyze tool 2:225–226
 altered-peptide ligand (APL) 3:243–244
 alternate splicing 3:968
 alternative exon (AE) 2:227
 alternative hypothesis (H_1) 1:691
 alternative open reading frames (altORFs) 3:918
 alternative RNA structures 3:580–581
 alternative splice site predictor (ASSP) 2:228
 alternative splicing (AS) 1:282, 2:187, 2:187–188, 2:221F, 2:222–223, 2:222F
 application of RNA-seq for analyzing 2:225
 bioinformatics approaches 2:223–225, 2:225–228
 eukaryotic *vs.* prokaryotic splicing 2:221
 genes 2:221
 regulation 2:223
 spliceosome and splicing mechanism 2:221–222
 tissue-specific AS in humans 2:223
 alternative splicing module (ASMs) 2:227
 Alu exclusion 2:275
 AlzGene 2:1050–1051
 Alzheimer's disease (AD) 2:344
 Alzheimer's disease sequencing project (ADSP) 2:179
Amathuxidia amythaon 3:990
 AMAZE database 2:799
 Amazon CloudFront 1:240
 Amazon Elastic MapReduce (EMR) 1:248
 Amazon Web Services (AWS) 1:160–161, 1:172–174, 1:240, 1:241F, 1:252, 1:254, 1:258, 1:334, 2:1153, 2:1154
 batch 2:1155
 Manager Plugin 1:248
 toolkit for eclipse 1:247
 Ambiguity 2:1100
 American College of Medical Genetics and Genomics (ACMG) 3:388
 American Standard Code for Information Interchange (ASCII) 1:1104–1105
 AmiGO application 3:220–221
 amino acid substitutions (AAS) 3:1
 amino acids (AA) 2:28, 2:29F, 3:1, 3:2F, 3:40, 3:310, 3:511, 3:631
 3d motif searching 3:620–621, 3:621T
 composition and molecular weight 2:28–29, 2:492
 mutations 3:655
 propensities 2:490
 properties derived from protein structural data 2:450–451
 scoring matrices 3:315
 sequences 2:53, 2:453, 3:307
 aminoacyl tRNA synthetases 3:968
 aminoisobutyric acid (Aib) 3:2
 ammonia (NH_3) 3:939
 AMPHORA2 workflow 3:188–189
 AmphoraNet webserver 3:188–189
 amplicon based 16S 3:184
 amplicon based 18S/ITS 3:184–185, 3:185T
 amplicon based metagenomic analysis 3:188
 functional analysis 3:189
 metagenomics phylogenetic analysis 3:188–189
 OTU picking and taxonomic assignment 3:188
 pre-processing of reads for amplicon analysis 3:188
 statistical analysis 3:188
 amplification refractory mutation system (ARMS) 3:432
 amplified ribosomal DNA restriction analysis (ARDRA) 3:203
 amprenavir (Agenerase) 2:585
 amyloid beta ($A\beta$) 3:2
 amyloid fibrils 2:509
 anabolic reactions 3:489
 anabolism 3:489
 analog-based designs 1:78
 analysis of splice variation (ANOSVA) 2:224
 analysis of variance (ANOVA) 1:648, 1:745, 2:237, 3:814
 analytical method validation 2:753
 anaphase-promoting complex or cyclosome (APC/C) 1:833
 ancestor(s) 1:871, 3:944–946
 diving deep into evidences 3:944–946
 ancestral recombination graph (ARG) 2:747–748, 2:749F
 ancestral repeats (ARs) 2:275
 ancestral selection graph (ASG) 2:749
 ancestral sequence prediction 2:734, 2:734F
 ancestral sequence reconstruction (ASR) 2:712, 2:750
 ancestry informative markers (AIMs) 3:375
 anchors. *see* pairwise fragments
 Anfinsen's Nobel acceptance speech 3:253
 Angelman syndrome (AS syndrome) 3:235–236
 angle-based outlier factor (ABOF) 1:459
 angle-based outliers 1:459
 angular resolution maximization 1:1088–1089
 animal model(s)
 database 2:103–104
 for human disease 3:429
 animal tissue sample preparation 3:465
 animal transcriptome databases 2:345–346
 anisotropic network model (ANM) 3:610–611
 annealing process 1:323, 3:543
 Annotare 1:139
 annotating functional importance of genes 3:456–457
 annotation and analysis 3:179–180
 annotation enrichment analysis 1:828
 annotation rule (AR) 3:222
 annotation score (AS) 3:222
 annotation(s) 1:826, 1:867, 1:871, 1:874, 3:187–188, 3:207
 repositories for variant storage and 3:169–170
 transfer approaches 2:807
 ANNOVAR tool 3:116, 3:145, 3:170, 3:299, 3:382
 anomalous scattering (AS) 2:514
 anomaly detection 1:338
 anonymity 1:254
 ANOVA model 1:702
 answer-set programming (ASP) 1:288, 1:294–295, 1:297–298, 1:298
 ant colony optimization (ACO) 2:1033–1034
 antagonistic networks. *see* food webs
 anti-citrullinated peptide antibody (ACPA) 3:729, 3:730
 anti-keratin antibodies (AKA) 3:730
 anti-perinuclear factor (APF) 3:730
 antialiasing 2:522
 antibiotics 3:927
 antibody (Ab) 2:909–910, 2:909, 2:940
 proteins 3:1
 antibody-dependent cell-mediated cytotoxicity (ADCC) 2:907
 antibody-specific epitope propensity (ASEP) 2:919
 antigen-presenting cells (APCs) 2:910
 Antijen 2:934, 2:935F
 processing and integration 2:914–915
 selection detection 2:947
 antimicrobial resistance 3:926
 bioinformatics in combating 3:928–929
 molecular mechanisms 3:926–928, 3:927T

- antiparallel β -sheets 2:489
- antisense oligonucleotides (ASOs) 2:669–670
- antisense transcription 3:842
- cis-and trans-antisense transcription 3:842–844
- differing effects 3:844–845
- Anton 1:65
- Antweb 2:1110
- Apache Hadoop 1:169–171, 1:222, 1:248
- Apache Spark 1:222, 1:248
- apo*-form PRLBD, conformations and energy landscapes of 2:558–559
- apolipoprotein B gene 3:968–969
- APOLLO 2:197–198
- apoptosis 3:71
- application programming interfaces (APIs) 1:167, 1:208, 1:209, 1:216, 1:821, 1:1052, 1:1096, 2:1065–1066, 2:1080, 2:1116–1117
- application-specific integrated circuits (ASICs) 2:1142
- application-specific kernel 1:497–498
- for documents 1:498–499
- for graphs 1:499
- for strings 1:498
- Applied Biosystem SOLiD 3:293
- approximate bayes (aBayes) 2:739
- approximate likelihood ratio tests (aLRT) 2:739
- approximation Bayesian computation (ABC) 2:749
- Apriori algorithm 1:363, 1:364
- AptaCluster clusters SELEX/HT-SELEX data 3:570
- aptamer
- analysis 3:569–570, 3:569T
- domains 3:568
- AptaPLEX 3:570
- AptaTrace process 3:569–570
- aprimidinic side (AP) 2:942–943
- Aquaria 2:533–534, 2:534F
- Aquifex aeolicus* 3:607
- Arabidopsis* miRNAs 2:125
- Arabidopsis thaliana* 2:135, 2:140, 3:472
- arabinose-binding protein (ABP) 3:637
- ARACNe algorithm 2:158
- arbitrary Lagrangian-Eulerian method (ALE method) 2:903
- arbitrary length paths (ALP) 1:804
- arbuscular mycorrhizal fungi 3:185
- Arcadia (software tool) 1:1051
- Archaea 3:18, 3:945
- ArchDB 2:467–468
- archeogenetics 3:375
- architecture for metabolomics consortium (ArMet) 1:130
- Arctos 2:1112
- area minimization 1:1088–1089
- area under curve (AUC) 1:433, 1:553–554, 1:686, 2:682, 2:919, 3:5, 3:9T, 3:333, 3:998
- area under precision-recall curve (AUCPR) 1:556
- area under precision-recall curve (AUPRC) 1:274
- area under ROC convex hull (AUCH) 1:554
- area under the receiver operating characteristic curve (A_{ROC}) 1:274, 2:955, 3:682–683, 3:683
- Arg-Glu-Arg-Glu query motif detection in database of PDB structures 3:627–628, 3:628F
- arginine 2:147, 2:840
- arginine deiminase (ADI) 3:733
- arginine/serine (RS) 2:223
- Argonaute protein (Ago) 2:165
- arithmetic logic units (ALU) 1:161
- arithmetic operators 1:180–181
- Arity 1:468
- ARNold tool 3:324
- aromatases adaptations in pigs 3:947
- aromatic amino acids 3:511
- aromatic residues 2:824–825
- Arqueobacteria* 3:945
- array 1:2, 1:178–179
- methods 3:344
- syntax 1:201
- arrayQualityMetrics package 3:117–118
- artemisinin-based combination treatment (ACT) 2:591
- arthropod and parasite sources 2:346
- artificial intelligence (AI) 1:272, 1:287, 1:294, 1:315, 1:589, 2:335, 2:1049, 3:67–68, 3:71, 3:669;
- see also* intelligent agents (IA)
- constraint satisfaction 1:290
- knowledge representation, reasoning, and logic 1:287–288
- machine learning 1:290–291
- natural language processing 1:291
- planning agents 1:290
- rational agents 1:288–289
- artificial neural network-trained NetMHCpan (ANN-trained NetMHCpan) 3:244–246
- artificial neural networks (ANN) 1:272, 1:276, 1:302, 1:342, 1:345, 1:390, 1:612, 1:621, 1:634, 1:635, 2:31, 2:121, 2:664–665, 2:669, 2:828, 2:916–917, 2:954, 2:1004, 2:1005, 3:1, 3:2, 3:42, 3:244, 3:669, 3:681;
- see also* deep neural networks (DNNs)
- artificial neuron 1:635F, 1:635
- artificial systems (ASs) 1:309
- Arvados (platform) 1:259
- Arvados 3:139
- ASEAN Centre for Biodiversity 2:1112
- ASEAN Member States (AMS) 2:1112
- Asia Pacific Bioinformatics Network (APBioNet) 2:110
- Asia Pacific Region (2015), dengue and chikungunya diagnostics in 3:13–14
- Asia Pacific Strategy for Emerging Diseases (APSED) 3:9
- aspects. *see* non-overlapping ontologies
- Aspergillus* 3:928
- A. fumigates* 3:928
- A. nidulans* 3:325
- ASprofile 2:227
- assay DNA modifications, experimental methods to 3:342–344, 3:343T
- assay for transposase-accessible chromatin-seq (ATAC-seq) 2:262
- assay ID (AID) 2:630–631
- assembled genomes, detection of repeats in 2:211–214
- assembled searchable giant arthropod read database (ASGARD) 2:346
- assembler 2:180
- assembly method 2:180, 2:359
- assignment of secondary structure in proteins (ASSP) 3:519
- assisted model building with energy refinement (Amber) 1:63–64, 2:552, 3:613
- Amber03 3:613
- Amber94 3:613
- Amber99SB 3:613
- Association Rule Mining (ARM) 1:367, 1:368–370, 1:369F, 1:370, 1:371
- association rules (AR) 1:358–359, 1:359–360, 1:397–398, 1:867
- ARM algorithms 1:368–370, 1:371
- binary attributes 1:367
- extended 1:370–371
- formalism 1:397–398
- mining 1:304, 1:358
- Search Space 1:368
- association test 3:443–444
- assortative mixing 1:965
- assortativity 1:965–966
- AStalavista tool 2:225
- ASTRO-FOLD method 1:68, 1:69F
- asymmetric alignment 1:104
- asymmetric least squares smoothing (AsLS) 3:469
- asymmetric units (ASU) 2:512, 2:836
- AT GC hydrogen bonding rule 3:546
- atazanavir 3:744T–748
- Atlas 1:1096
- Atlas for Cancer Signaling Networks (ACSN) 1:1076
- Atlas of Gene Expression in Mouse Aging Project (AGEMAP) 2:346
- atmospheric gas plasma (AGP) 3:850, 3:852
- atmospheric pressure chemical ionization (APCI) 2:411, 3:466
- atmospheric pressure photoionization (APPI) 3:466
- atom-decomposition 2:1146
- atomic non-local environment assessment (ANOLEA) 1:56
- atomic property fields (APF) 2:653
- atomistic molecular dynamics simulations 2:544
- ATP-binding cassette (ABC) 3:379, 3:927–928
- atrial septal defect 1:701–702
- attention deficit hyperactivity disorder (ADHD) 2:1050–1051
- attribute selection 1:387–388, 1:482
- attributeList 1:1064
- attributes 1:466, 1:1048–1049
- AUC-ROC plot 2:197
- audit 1:254
- AUGUSTUS 2:202
- Aurignacian culture 3:373–374

AutDB 2:1050–1051
 authenticity 1:254
 autism spectrum disorders (ASD) 2:1050–1051
 autoantibodies in RA diagnosis 3:729–730
 autoassociator. *see* autoencoders
 autocrine 1:917
 AutoDock program 3:276
 AutoDock Vina (ADV) 2:593–594, 3:673
 AutoDockCloud system 1:262–263
 AutoDockTools (ADT) 2:690, 3:276, 3:278F
 autoencoders 1:643–644, 1:644F
 automated mass spectral deconvolution and identification system (AMDIS) 2:399, 2:403
 automated modelling 3:267
 automated pipelines 3:169
 automated rRNA intergenic Spacer Analysis (ARISA) 3:203
 automated structure prediction 3:264–265
 automatic annotation 2:10–11
 automatic barcode gap discovery (ABGD) 3:991–992
 automatic pathway mining 1:1080–1081
 autonomy 1:310
 autophagy 3:106
 availabilityList element 1:1064
 available chemicals directory (ACD) 2:689
 average and mid-ranged value discretization 1:469
 average assignment method 2:454
 average gain 1:553
 average lift 1:553
 average linkage 1:351–352
 average path length of graph 1:936
 average turnover rate 2:985
 avian influenza 3:241
 7-azaindole fragment 3:757F
 Azure Machine Learning (Azure ML) 1:250

B

B cell receptors (BCRs) 2:940
 B cell(s) 2:942
 epitopes 2:953
 prediction 2:955–965, 2:962T–964
 B lymphocytes 2:972
 B-cell epitope interaction database (BEID) 2:934–935
 B-cell lymphoma-2 (Bcl-2) 2:669–670
 B-cell receptor (BCR) 2:915
 B-cells 3:243
 epitopes 2:931, 3:39
 prediction 2:910, 2:915–916, 2:915F, 2:917T, 2:919–920
 receptors 3:39
 B-factor 1:42, 2:841
Bacillus anthracis 3:926
Bacillus licheniformis 3:465
Bacillus simplex 3:942
Bacillus subtilis 2:140, 3:324
 backbone extraction and merge strategy (BEAMS) 1:1011–1012
 backbone generation 1:45

background mutation rate (BMR) 2:123–124
 Backpropagation 1:390, 1:390–392, 1:391, 1:621, 1:626–627, 1:638–640, 1:640, 3:289;
 see also Perceptron
 Backtracking technique 1:11–12
 backward approach 1:734
 backward Kolmogorov equation 2:29
 backward stepwise selection 1:493
 bacteria 2:737, 3:18, 3:927
 HGT in 3:957–959
 Bacteria-Ensembl database 2:252
 bacterial bloodstream infections (bBSI) 3:189
 Bacterial Diversity Metadatabase (Bacdiv Metadatabase) 2:1111
 bacterial vaginosis (BV) 1:277
 bacterial wilt (BW) 3:428
 Bacteroidetes 3:21–23
 bags-of-words method (BoW method) 2:1007
 balanced accuracy (BACC) 1:549
 balanced bootstrap sampling 1:526
 balanol 3:276F, 3:276
 Balaur technique 1:252
 Baliphy approach 2:712
 Ball-Hall index 1:445
 BALLDock 3:673
 BAM files 3:298
 Barabasi-Albert graph 1:971F
 barcode index numbers (BINs) 3:986, 3:991
 barcode of life data systems (BOLD systems) 2:1015–1018, 2:1117, 3:986, 3:989
 BINs system 3:991
 identification engine 3:989–990, 3:991F
 bariatric surgery 1:155
 base alignment quality score (BAQ score) 3:437
 base excision repair mechanism (BER mechanism) 2:942–943, 3:341
 base pairs (bps) 2:210, 2:376, 2:567–568, 2:942, 2:1142, 3:447, 3:510
 base quality score recalibration (BQSR) 2:185, 3:168
 base stacking 3:504
 base-base interactions, descriptions and nomenclature of 3:547
 base-pairing 3:504–505
 base-pairing probability (BPP) 2:578
 DP for 2:579–580
 BASELINE 2:947
 baseline correction 3:469
 BaseSpace (platform) 1:253
 basic graph pattern (BGP) 1:802
 basic local alignment search tool (BLAST) 1:39, 1:44, 1:263, 2:943, 3:30, 3:222, 3:233, 3:255
 basically available, soft state, eventually consistent principal (BASE principal) 2:1064
 batch effect 1:738, 3:309, 3:793, 3:815–816
 Bayes factor 1:405, 1:405T
 Bayes' theorem 1:403, 1:403–404, 1:404, 1:404–406, 1:407
 Bayesian approach 1:735, 2:834–835, 3:393
 Bayesian bootstrap imputation 1:465

Bayesian classification 1:275
 Bayesian classifiers 3:129
 Bayesian epistasis association mapping (BEAM) 2:118
 Bayesian information criterion (BIC) 1:492, 1:734, 2:159, 2:714
 Bayesian methods 2:709, 2:722, 3:443
 Bayesian networks (BN) 1:279–280, 1:1053–1054, 2:158–159, 2:877–878, 3:63, 3:822, 3:848, 3:848F, 3:849F, 3:849T
 Bayesian phylogenetic methods 2:724
 Bayesian Poisson tree processes (bPTP) 3:992
 bayesian principal component analysis method (BPCA method) 1:473
 BEAGLE 3:382
 beam-forming approaches 2:808
 BEAST program 2:709, 2:710, 2:750, 2:753
 behavior-based deanonymization attack 1:269
 Beijing genomic institute (BGI) 3:850
 beliefs-desires-intentions (BDI) 1:317, 1:318–319
 Bellman-Ford algorithm 1:940, 1:943–944, 1:944
 benchmark dataset 3:122
 benchmarking and resource portals 1:984
 bend minimization 1:1088–1089
 Benjamini-Hochberg method (BH method) 1:695–696, 1:982, 3:221–222, 3:456
 benzamide moiety 3:279
 benzoyl-arginine amide (BAA) 3:733, 3:736F, 3:736
 BEPro 2:955
 Berkeley morphometric visualisation and quantification 2:109
 Berkeley Open Infrastructure for Network Computing (BOINC) 1:232
 best linear unbiased estimator (BLUE) 1:722–723, 1:724
 best-matching unit (BMU) 1:441
 beta diversity 3:19, 3:188, 3:206
 β globins 3:966
 beta secretase 1 (BACE1) 3:236
 β -augmentation 3:5
 β -bulge 3:522
 β -sheet 3:514–516
 structure 3:646
 surfaces to boost protein stability 3:636
 β -strands 2:489, 3:514–516
 β_1 -adrenergic receptor (β_1 -AR) 3:261
 between-class scatter matrix 1:489
 betweenness centrality 1:937, 1:953, 1:954–957, 1:962, 2:823–824
 Bhageerath software 1:67–68, 1:68F
 BHV distance use 2:743
 bias 1:740
 bias-variance
 decomposition 1:525, 1:527
 trade-off 1:525–526
 biased genomic data 2:259–260
 biclique. *see* complete bipartite graph
 biclustering analysis 1:1040–1041, 3:796–797, 3:798, 3:798F
 bidirectional best-hits (BBH) 2:261
 Bienaymé-Galton-Watson (BGW) 2:23

- bifurcation analysis 2:792–793
 Big Data to Knowledge program (BD2K program) 2:1152
 big data 2:808–809, 3:66–67
 BigFoot 2:284
 BigML 1:250
 BigQuery 1:242
 bigrams 2:1102
 BIMAS 2:954
 bimolecular fluorescence complementation (BiFC) 2:824
 binary alignment map (BAM) 2:228, 2:356, 3:167
 binary classification
 evaluating obtained value of performance measure 1:556–557
 performance measures
 based on single classification threshold 1:547–549
 based on probabilistic understanding of error 1:550–552
 problems 1:512–513
 ranking measures 1:552–556, 1:554F
 binary heap data structure 1:7
 binary large objects (BLOBs) 2:602
 binary logistic regression (BLR) 1:732
 binary matrix 3:495
 BinBase 2:397–399
 binding
 affinity 3:634–635, 3:781
 free energy 2:590–591, 3:614
 interaction evaluation 3:659–660
 methods 1:468, 1:474, 3:187, 3:207, 3:469
 MOAD 2:633
 site identification 2:587–588, 3:274
 binding site comparison 2:651T, 2:655
 model principles, definitions, and criteria for similarity 2:653–654
 modeling 2:650–652, 2:652F
 off-target analysis 2:656–657
 polypharmacology 2:655
 protein–protein interactions 2:657
 selectivity profiling 2:655–656
 bindingDB 2:632, 2:1048
 binomial logistic regression. *see* binary logistic regression (BLR)
 binomial tests 1:699–700, 3:222
 “bins” 3:207
 Bio-Docklets 3:139
 bio-magnetic resonance bank (BMRB) 2:431
 bio-nano-technologies 1:1092
 bio-ontologies 1:814, 1:814–815
 bio-sequence analysis 1:230
 bio-terminologies 1:814
 bioactivity databases
 binding MOAD 2:633
 bindingDB 2:632
 ChEMBL 2:631–632
 PDBbind 2:632–633
 BioASQ 2:1105
 Bioblender (2012) 2:532
 BioCarta 2:799
 BIOCAT software 2:1006
 biochemical networks 1:1106
 biochemical oxygen demand (BOD) 3:1, 3:3
 biochemical reactions 1:147, 3:838
 biochemistry related databases 2:102
 Biochimica et Biophysica Acta (BBA) 2:1074
 bioconductor 1:333, 2:1152
 BioContainers framework 2:86
 BioCreAtIvE challenge 1:568, 2:1105
 BioCyc database 2:404, 3:317
 BioDataServer 1:1095
 BioDIG 2:1020–1021
 biodiversity databases 2:103, 2:1110, 2:1115
 access and benefit sharing 2:1119–1120
 biodiversity and environmental change 2:1117–1118
 economics and regulatory framework 2:1119
 importance of digitizing biodiversity information 2:1117
 invasive alien species and biosecurity 2:1118
 public health and wildlife disease 2:1118–1119
 biodiversity heritage library (BHL) 2:1117
 BIOEDIT 3:989
 biogenesis 2:124
 BioGraph 1:910
 BioGrid 1:171–172, 2:850, 2:1104
 bioimage analysis pipelines 3:126–128, 3:128F, 3:129
 BIOimage Classification and Annotation Tool (BIOCAT) 2:1003–1004
 bioimage(s) 1:160–161, 2:1028
 using CBIR 2:1018–1019
 databases 2:1011, 2:1022–1024
 informatics 2:1004–1006
 model procedural flow 2:1024F
 segmentation 2:1028, 2:1028–1030, 2:1029F, 2:1034, 2:1039–1040, 2:1043T
 solution for bioimage databases using ontology and CBIR approaches 2:1024–1025
 text-based query 2:1011
 bioinformatic(s) 1:30, 1:81, 1:89, 1:130, 1:211, 1:223, 1:223T, 1:871, 1:889, 1:899–900, 1:915, 1:923–924, 1:1071, 2:89, 2:123, 2:1142, 2:1151–1152, 3:1, 3:459, 3:478, 3:836–839, 3:921;
 see also translational bioinformatics (TBI)
 algorithm designing techniques 1:5
 applications 1:230–231, 1:306, 1:307F
 challenges and opportunities for pipelines and workflows 2:1157–1158
 cloud-based services 1:258–259
 clouds 1:1099, 1:1100F, 2:1154–1155
 in combating antimicrobial resistance 3:928–929
 and computational biology 1:313
 data cleaning methods
 for missing values in 1:472–473
 for noisy data in 1:474–475
 and data science in Python 1:197–198
 databases 3:393
 dedicating bioinformatics analysis hardware 2:1142
 dependency management and containerization 2:1155
 of discovery 3:106–107
 in drug design 2:614
 feature selection and extraction in 1:301–302
 gene functions using bioinformatics strategy 3:444
 to identify, annotate and analyze repeats in genomes 2:211–214
 Java for 1:208
 kernel methods in 1:515–516
 in mutational studies 2:123–124
 ontologies in 1:788
 resources for pharmacogenetics/-genomics 3:382
 semantically encoded workflows 2:1157
 systems and applications 2:1152
 tools 2:124–125, 2:1152
 to analyzing codon usage 3:329
 to optimizing codon usage in DNA 3:329
 for protein structure prediction and analysis 1:38–39
 in ribosomal SSU rRNA sequence and structure studies 3:10–11
 used for regulation of genes 2:120
 for transcriptome analysis 3:455
 workflow sharing 2:1156
 bioinformatics data models 1:110F;
 see also computable phenotyping model
 algorithms and access to data 1:113–114
 algorithms for data elaboration 1:115
 data elaboration 1:114–115
 management of information and databases 1:111–113, 1:112F
 structural bioinformatics 1:110–111
 use cases 1:115
 BIOISIS 2:465–466
 BioJava 1:208
 BioKleisli 1:1093
 biological and medical ontologies
 OBO 1:815–816
 terminologies and ontologies 1:813–814
 UMLS 1:816–819
 biological annotation 1:203–204
 biological association networks (BANs) 2:797
 biological carbon fixation evolution 3:946
 biological connection markup language (BCML) 1:1065
 biological data(bases) 1:300, 1:477, 1:1036, 2:96, 2:99, 2:100–101, 2:101–102, 2:103, 2:104, 2:109–112, 2:112–113, 3:196–197
 analysis pipeline 3:29F
 formats for 1:131–132, 1:135
 standards for 1:130–131
 and types of data stored 3:29–30, 3:34–35
 Biological General Repository for Interaction Datasets (BioGRID) 2:103, 3:1–2, 3:887
 biological layers, simulation models in 2:866
 biological magnetic resonance bank (BMRB) 2:460, 2:465, 3:472

- biological network(s) 1:102, 1:915, 1:958, 1:961*F*, 1:982, 1:1047–1048, 1:1048–1049, 2:816–817, 2:822–824
 - as expressive data models 1:1104–1105
 - inference 2:805
- Biological Networks Gene Ontology tool (BiNGO) 3:222
- biological pathway (bp) 1:982–983, 1:1047–1048, 1:1048, 1:1048–1049, 1:1067, 1:1071, 1:1071–1074, 1:1072*T*–1073, 1:1074*F*, 1:1077–1078;
 - see also* protein-protein interaction (PPI)
- alignment approaches 1:1067–1068
- analysis in genome-wide association studies 1:1069
- annotation classes based on data source 1:1071–1074
- context 1:1081
- curation 1:1079–1080
- data 1:1049
- gene regulation pathways 1:1052–1055, 1:1053*F*, 1:1057*F*
- graph mining approaches 1:1067
- heterogeneity in pathway definition 1:1079
- human proteome coverage 1:1077–1078
- ID conversion and naming 1:1078–1079
- layout algorithms 1:1089–1090
- metabolic pathways 1:1049–1051, 1:1050*F*
- paradigms for graph drawing 1:1087–1089
- representations 1:1085–1087
- signal transduction pathways 1:1056–1059
- in systems biology 1:1071
- Biological Pathway Exchange (BioPAX) 1:147, 1:147–148, 1:863, 1:1049, 1:1063–1064, 1:1076, 2:858–859
 - conding example 1:149–152
 - language 1:1027
 - level 2 and level 1 ontology structure 1:148–149
 - level 3 ontology structure 1:147–148
 - pathway databases 1:152
- biological process (BP) 1:823, 1:867, 1:889, 1:900, 2:1, 2:2, 3:821
- biological replicates 3:356
- biological sciences, KDD in 1:337
- biological systems (BSs) 1:147, 1:309, 2:789–790, 2:875–876, 3:218, 3:796–797
 - standards for structuring models 2:885–886
- biological text mining 2:1105
- biologically active enzyme 3:638
- biologically inspired models 2:808
- biology 1:1
- bioluminescence (BL) 2:1097
- BioMagResBank 3:470
- biomarker association network (BAN) 3:63
- biomarkers 2:1097, 3:60
 - efficacy 2:1052
 - identification and extraction of biomarker information 3:60
- BioMart 1:1093
- BioMe BioBank 2:1053
- biomedical corpora 1:604
- biomedical data
 - data models for 1:1100–1102
 - data storage 1:1099–1100, 1:1099*T*
- biomedical domain 2:1104
- Biomedical Entity Search Tool (BEST) 1:607
- biomedical knowledge 2:1099
- biomedical literature browsing, text mining for simplifying 1:581–583
- biomedical natural language processing (BioNLP) 1:561
- biomedical networks 1:1016, 1:1030–1031
 - annotation 1:1022–1024, 1:1026*F*
 - databases 1:1018
 - ecological networks 1:1018
 - epigenetic and GRN 1:1016
 - example of phylogeny 1:1031*F*
 - gene coexpression networks 1:1016–1018
 - layout algorithms 1:1021–1022
 - layouts 1:1018–1019, 1:1019*F*
 - metabolic networks 1:1018
 - motivation 1:1016
 - network data formats 1:1026–1027
 - network visualization software 1:1027–1030
 - phylogenetic networks 1:1018
 - protein-protein interaction networks 1:1018
 - signaling networks 1:1018
 - visualization 1:1016, 1:1017*T*, 1:1030
- biomedical package (BMDP) 1:689
- Biomedical QA (BioQA) 3:999
- biomedical realm 3:996
- biomedical text mining 2:1090, 2:1101, 3:996
 - biomedical applications, shared tasks, and current/future challenges 2:1104
 - community challenges and shared tasks 2:1104–1105
 - information retrieval and extraction tasks 2:1104
 - information retrieval and text classification 2:1101–1102
 - knowledge discovery through text and future directions 2:1105–1106
 - natural language processing and information extraction 2:1099–1101
 - tools 1:605–606, 1:605*T*
 - NER and normalization 1:605–606
 - relationship and event extraction 1:606
- biomedicine 1:89
- Biomine 1:910
- BioModels database 1:152, 1:862
- Biomolecular Interaction Network Database (BIND) 1:1105, 3:1–2
- biomolecular structures
 - base-pairing and double helical structure of DNA 3:504–505
 - based on sequence similarity 3:514
 - based on structure 3:512–514
 - central dogma of life 3:504
 - comparison in 3D 2:545
 - computational approaches for identification 3:517–519
 - assigning SSEs to 3D structures 3:519–520
 - prediction of secondary structures 3:517–519
 - distortions in regular SSEs 3:521–522
- DNA and RNA adopting different structural forms 3:505–508
- experimental approaches for identification 3:520–521
- hierarchical organization and classification 3:512–514
- polysaccharides-macromolecular biopolymer 3:523–525
- predicting nucleic acids secondary structures 3:508–510
- prediction of 3D structure 3:509–510
- protein 3D structure prediction and analyses 3:522–523
- protein structure and sequence repositories 3:522
- proteins—building blocks of life 3:510–512
- secondary structural elements 3:514
- Biomolecules 1:142
 - strings and sequences for representing 1:1102
 - structures for representing 1:1102–1104, 1:1103*F*
- BIONC based grids 1:232
- BioNetGen language (BNGL) 1:1051
- BioNLP-shared tasks (BioNLP-ST) 1:568–569
- BioPath database 1:1052
- BioPerl 1:706–707, 1:709–712;
 - see also* Perl
 - installing BioPerl on windows machines 1:707
 - installing BioPerl on Linux and Mac OSX machines 1:706–707
 - modules 1:709
 - objects management in Perl and 1:707–709
- BioPortal 1:815
- Biopython 1:198
- biosecurity 2:1118
- biosensors 3:636–638
- BioSilico 2:1020
- Biospecimen Core Resource (BCR) 2:108
- biostatistics 1:648, 1:685, 1:685*F*
 - case studies 1:669
 - clinical depression data 1:667*T*–668
 - inferential statistics 1:654–657
 - software 1:670
 - statistical analysis 1:648–651, 1:649*F*
- biotin labelling method 3:434
- Biovia Discovery Studio Visualizer (BDSV) 2:690
- BioVLAB-MMIA-NGS 1:253
- BioVLAB-NGS 1:253
- BioWarehouse 1:1096
- BioWeka 1:305, 1:306*F*
- BioXpress 2:342
- Biozon 1:1096
- Bipartite Graph Library 1:1011
- bipartite graphs 1:924, 1:924*F*, 1:925–926, 1:1048–1049, 3:489
- BIPARTMOD software 2:1135
- bird flu 3:241
- bis-tetrahydrofuran moiety (bis-THF moiety) 3:710
- Bismark software 3:356
- Bisulflighter software 3:356
- bisulfite (BS) 3:342–344
- bisulfite sequencing (BS-Seq) 3:324, 3:355

- bisulfite-converted restriction site associated DNAsequencing (BsRADseq) 2:128
 BITOLA 1:910
 Bitset-based Unbiased miRNA Functional Enrichment Tool (BUFET) 2:125
 bitwise operators 1:181
 bivariate normal distribution 1:710–713
 BLAST 2:12, 2:215, 2:475, 2:1143, 3:206, 3:981
 bit scores 1:997
 search 2:451–452
 and similar database search methods 3:295–296
 BLAST-like alignment tool (BLAT) 2:272–273, 3:233
 Blast2GO 3:222
 blastocyst 3:825
 BLASTX genomic query 2:201
 Blender Game Engine (BGE) 2:532
 blinding effect 3:833
 Blobworld image retrieval system 1:443
 block and bridge preserving (BBP) 2:640–641
 block(s) 2:39, 2:271
 BLOcks SUBstitution Matrix (BLOSUM 62 matrix) 1:41–42, 1:41F
 blood brain barrier (BBB) 2:661, 2:670
 blood oxygenation level dependent signal (BOLD signal) 2:808
 blood transcription modules (BTM) 1:982
 Bloomington Drosophila Stock Center (BDSC) 2:104
 BLOSUM matrices 1:28, 3:333, 3:393
 BLOSUM62 matrices 1:28, 2:702F, 3:983
 Bluemix dedicated 1:243
 Bluemix Hybrid 1:243
 Bluemix public 1:243
 BLUEPRINT 3:341
 epigenome 3:357–358
 BlueWaters 1:65
 BNFinder2 tool 1:1055
 body mass index (BMI) 2:171, 2:236
 body systems
 comparative analysis of transcriptome profile 3:99–100
 GO and KEGG pathway analysis of protein-coding genes 3:98F
 reference gene catalog of human primary monocytes 3:96–99
 techniques for cell preparation and transcriptome profiling 3:95–96
 Boltzmann distribution 2:576–577
 Boltzmann Likelihood method (BL method) 2:576
 Boltzmann machine 1:644–646, 1:645F
 Bond Polarization Theory (BPT) 2:561
 bond strength methods 2:825
 bonded term 3:608
 bone mineral density (BMD) 2:861–862
 Bonferroni correction 1:695, 3:456, 3:459, 3:833, 3:982
 Boolean networks (BN) 1:1053–1054, 2:159–160, 2:802–803, 2:877, 3:822
 Boolean Operation-based Screening and Testing (BOOST) 2:118
 Boolean operators 2:1102
 Boolean queries 2:1102
 boosting 1:343–344, 1:524, 1:531–533
 bootstrap (BP) 1:434, 1:435F, 1:558, 1:766, 2:738, 2:738F
 bootstrap-t confidence interval 1:769–771
 confidence intervals 1:769
 estimation of prediction error 1:771–772
 percentile confidence interval 1:558, 1:769
 resampling 1:526–527
 resubstitution error 1:772
 Bootstrapp AGGregatING (Bagging) 1:343–344, 1:376–377, 1:524, 1:525, 1:527F, 1:534
 algorithm 1:527–528, 1:528
 motivation and fundamentals 1:525–526
 and random forest 1:527–528
 Borda count approach 1:523
 Born-Oppenheimer approximation (BO approximation) 1:63–64, 2:607
Borrelia burgdorferi 2:830
 bottom up MS 2:1069–1070
 bottom-up approach 1:787, 2:801, 2:1033, 3:243, 3:311–312
 boundary element 2:303
 boundary-mutation model 2:12
 bounding box 2:1001, 2:1002F
 Bourne Again Shell (bash) 3:145
 Bourne shell (Bourne sh) 3:145
 bow-tie structure 3:485
 Bowley's skewness. *see* Galton skewness
 boxplot 1:675, 1:676F
 Boyer-Moore algorithm 1:9
 Bpipe 3:137
 BPRMeth 3:809
 BPSL1549 protein 3:624T
 brain functional connectivity 2:808
 brain transcriptome database (BrainTx) 2:345–346
 branch confidence 2:733
 branch lengths 2:727, 2:729F, 2:733
 in consensus trees 2:741
 branch point sequence (BPS) 2:222
 Branch-and-Bound technique 1:11–12
 Branching models 2:23–24
 BRAunschweig ENzyme Database (BRENDA) 2:102
 BRCA1 2:1102
 breadth first search (BFS) 1:89, 1:90, 1:91, 1:940–943, 1:941, 1:941F, 1:947
 breast cancer database (BCDB) 2:1094, 2:1095F
 breast cancer with python, predict relapse of primary 1:379–381, 1:382F
 BRENDA Tissue Ontology (BTO) 1:1024–1025
 brick motifs 1:98
 bridge motifs 1:98
 bridge PCR 3:158
 Brier score 1:551
 bring your own license model (BYOL model) 3:70
 Broad GDAC Firehose 2:109
 broadly enriched DBPs 3:114
 Bromo and Extra terminal (BET) 3:787
 5-bromo-2'-deoxyuridine (BrdU) 2:985, 2:987–988
 Brookhaven National Laboratory (BNL) 2:460
 Brownian dynamics method (BD method) 2:560, 2:868–869
 Brownian motion 1:750
 browsers 2:335–337
 brute force approach. *see* exhaustive search
 Bruton's tyrosine kinase (BTK) 3:95
 BS-Seeker2 tool 3:324
 BSMAP software 3:356
 BsmF1 enzyme 3:820
 BSmooth software 3:356
 BSPlib 1:219
Bub1 1:834
BubR1 1:834
 bucket-based-index 1:267
 building classification rules 1:388–389
 Bulk Synchronous Parallel model (BSP model) 1:219
 bulk transcriptomic analysis 3:795–796, 3:795F;
 see also single cell transcriptomic analysis
 analysis for module identification 3:796–798
 differential expression analysis 3:795–796
 functional analysis 3:800–801
 ncRNAs 3:799–800
 variant calling from RNA-seq data 3:798–799
 Burdock 2:347
 burden test 2:190, 3:115–116
 Burkholderia lethal factor 1 (BLF-1) 3:726
Burkholderia pseudomallei 3:726–727
 core hypothetical proteins in 3:957–959, 3:958T, 3:959F
 unique genes in 3:959–960, 3:960–961
 Burkholderiaceae family 3:957–959
 Burrow-Wheeler Transformation (BWT) 2:330, 3:151, 3:166–167, 3:177
 Burrow-Wheels Alignment (BWA) 1:212, 3:161
 Burrows Wheeler transform (BWT) 2:357, 2:1143
 Buruli ulcer 3:426–427
 BUSCO tool 2:203, 3:305, 3:310
 butterflies of Setiu Wetlands, Malaysia 3:992–993
 byte data type 1:206
- ## C
- C language 1:164
 C Shell (csh) 3:145
 C-Alpha, c-Beta, Side-chain (CABS) 3:609–610
 c-Jun amino-terminal kinases 12/3 (JNK12/3) 2:857
 C-terminal domain (CTD) 3:607
 C-termini 3:265
 of proteins 2:56–57
 C-value paradox 2:210
 C4.5 algorithm 1:385–387, 1:387
 CACNA1D 1:903
 CAD-score 1:71, 3:593

- CADgene 2:1049–1050
Caenorhabditis elegans 2:999, 3:47, 3:966
 caffeine 1:870
 CAGE method 3:820
 calcium dependency 3:731–732
 calibrating molecular clock 2:721–722
 calmodulin 3:607
 Cambridge Crystallographic Data Center 3:274
 Canadian Centre for DNA Barcoding (CCDB) 3:986
 cancer 2:1102, 3:54, 3:170–171
 based study 3:483–486
 cancer-specific DMRs 3:347
 models and text mining 2:1089–1090
 Raf–MEK–ERK MAPK signaling pathway in 2:856–857
 RNA-seq Nexus 2:342–343
 somatic variants detection in cancer genomes 3:392–393
 therapies 3:929
 Cancer Cell Line Encyclopedia (CCLE) 2:1079–1080
 Cancer Digital Slide Archive (CDSA) 2:109
 cancer drug resistance (CancerDR) 3:929
 Cancer Genome Characterization Initiative (CGCI) 2:1078–1079
 cancer genome interpreter (CGI) 2:1059
 cancer Genomics Hub (CGHub) 2:108
 Cancer Genomics Hub (CGHub) 2:1077
 Cancer Program Resource Gateway 2:1079–1080
 cancer stem cells (CSC) 3:53–54
 Cancer-Specific high-throughput Annotation of Somatic Mutations (CHASM) 3:1–3
 Cancers Genome Workbench (CGWB) 2:109
 CANDID 1:910
Candida albicans 2:127–128, 3:105, 3:325, 3:928
Candida glabrata 2:127–128
 CanEvolve 2:1081
 Cannings model 2:22–23
 canonical ensemble 2:550
 canSAR 2:1081
 cap analysis gene expression (CAGE) 2:120, 2:1048, 3:819–820, 3:820
 CAP theorem 2:1064, 2:1065F, 2:1066F
 capillary electrophoresis (CE) 3:465, 3:575
 CE-MS metabolomics 3:467
 capillary gel electrophoresis (CGE) 3:467
 capillary isoelectric focusing (CIEF) 3:467
 capillary isotachopheresis (CITP) 3:467
 capillary zone electrophoresis (CZE) 3:467
 CAPN10 2:286
 capsid protein (C protein) 3:763–764
 captopril 3:744T–748
 Car-Parrinello method 1:78
 Carb-builder 3:525
 carbohydrate binding proteins (CBPs) 3:667
 carbohydrate intrinsic energy functions (CHI energy functions) 3:673
 carbohydrates 3:666
 carboxy-fluorescein diacetate succinimidyl ester (CFSE) 2:987
 Carboxyfluorescein succinimidyl ester (CFSE) 3:40
 5-carboxylcytosine (5caC) 3:341, 3:342
 cardiac function 2:901–902
 modeling
 fluid dynamics and circulatory blood flow 2:903
 passive and active mechanical properties of myocardium 2:903
 spatial propagation of electrical activation 2:902–903
 multi-scale overview of cardiac electromechanics 2:903–904
 multiscale model of cardiac electromechanics 2:902F
 single-cell ionic models 2:901–902
 CardioGenBase 2:1060–1061
 Cardiovascular Disease Ontology (CVDO) 1:1025
 CAREMAP 2:1049–1050
 CARMA3 3:207
 Carrillo-Lipman algorithm 1:30
 CarrotDB database of *Daucus carota* L. 2:347
 Cas-OFFinder algorithm 2:123
 cascading algorithm 1:524, 1:540–541, 1:540F
 case-based reasoning classifier (CBR classifier) 1:415–416
 cycle 1:415–416, 1:416F
 operating details 1:417
 casein kinase II (CK2) 3:252
 CASPER program 3:525
 CAT models 2:715
 catabolic reactions 3:489
 catabolism 3:489
 catalogue of life (COL) 2:1117
 catalogue of somatic mutations in cancer database (COSMIC database) 2:123
 catalytic function addition for design of biologically active enzyme 3:638
 catalytic site atlas (CSA) 3:622
 cath-resolve-hits software (CRH software) 2:479
 CATHEDRAL algorithm 2:474, 2:475T
 Causal network (CN) 2:159
 Cavender-Farris model (CF model) 2:707
 CBA classifier builder (CBA-CB) 1:398
 CBA rule generator (CBA-RG) 1:398
 cBio cancer genomics portal 2:1079
 CBS-Pred 3:673–674
 CBTOPE 2:965
 Cell ImageLibrary-CCDB (CIL-CCDB) 2:1013, 2:1016F
 Cell Line Ontology (CLO) 1:844
 cell markup language (CellML) 1:858, 1:1065, 2:885, 2:886, 2:889
 cell sequencing (Cel-seq) 2:333
 cell(s) 1:917, 1:1047–1048, 1:1050
 cell-surface receptor 1:1057–1058
 communication analysis 3:801–802
 cycle 3:74, 3:84
 predicting abundance of nuclear proteins in cell-cycle phases 3:851–852
 differentiation 2:985, 3:825
 gene transcription during 3:825–826
 genes involving in 3:827
 division 3:84
 growth 3:825
 illustrator tool 1:1058
 modeling and simulation 2:864–865, 2:865F
 deterministic simulation and stochastic simulation 2:865–866
 dynamic modeling and static modeling 2:864–865
 mechanism-based modeling and logic-based modeling 2:866
 predictive kinetic model application 2:867–868
 simulation models in biological layers 2:866
 spatial simulation techniques 2:868–869
 tools for 2:869, 2:870T–871
 nucleus size correlating with chromatin chain length 2:296–299, 2:297F
 origin of cells and organized structure 3:940
 proliferation 3:825
 signaling 1:917
 pathways 1:147, 1:1086
 specialization during development 3:825, 3:826F
 techniques for cell preparation 3:95–96
 type identification 3:801, 3:802F
 wall biosynthesis 1:829
 CellMaps application 1:1028
 CellProfiler 2:1006
 cellular automata (CAs) 1:272, 1:1055, 2:878, 2:879F, 2:921
 cellular biology 3:891
 cellular component (CC) 1:823, 1:833, 1:867, 1:889, 2:1, 2:8–9, 3:821
 cellular lipidome 2:894
 cellular metabolic process 3:84
 cellular Potts models 2:878
 cellular signaling 1:1018
 cellular-automata models 2:901
 cellular-driven mechanisms 2:906
 Center for Biological Sequence Analysis (CBS) 3:40–41
 Center for Cancer Genomics (CCG) 2:1077, 2:1078–1079
 center for computational mass spectrometry (CCMS) 2:80, 3:863
 Center for Infectious Disease Research (CIDR) 3:932
 centiMorgan (cM) 2:242
 central processing unit (CPU) 1:209, 2:825, 3:135
 central proteomics facilities pipeline (CPFP) 3:858
 central tendency measures 1:674–676
 centrality analysis 1:950
 centrality indexes 1:83, 1:950;
 see also cohesion indexes
 centrality indicators 1:81
 centrality measures 1:936, 1:937
 degree and walk-based centralities 1:951
 shortest-paths based centralities 1:952
 centre for biodiversity genomics (CBG) 3:986

- centroid 1:352, 2:1001, 2:1002F
centroid-based approaches 1:440–441
centroid-based clustering method 1:1038
centromere repeat-associated small RNA (crasiRNA) 3:234
CEPH Reference Family Panel 2:243–244
cerebellar development transcriptome database (CD-DBT) 2:345–346
cerebral calcification 1:704
cerebrospinal fluid (CSF) 2:400
CFinder 1:96, 1:979
chain referral sampling. *see* snowball sampling (SBS)
chameleon sequences 2:490
CHAOS seed-finding method 2:271–272
Chapman-Kolmogorov Equation 1:747
char data type 1:207
character-based representation 3:447
character-string-based approach 3:449
charged couple device (CCD) 3:435
CHARMM36 3:672
CheckMyMetal server 2:545
ChemAxon tools 2:591–592
ChemBank 2:1090
ChEMBL 2:631–632, 2:1093–1094
ChemCell tool 1:1058
ChemExpr 2:1090
chemical abstract service (CAS) 2:604–605, 3:478–480
chemical and enzymatic structure probing of RNA 3:575
chemical bonding 2:601
chemical compounds (CC) 2:602
Chemical Entities of Biological Interest (ChEBI) 1:874, 2:5, 2:635–636, 2:889, 2:1090, 3:470, 3:478–480
chemical inference of RNA structures sequencing (CIRS-seq) 2:580
chemical ionisation (CI) 2:411
chemical master equation (CME) 1:750
chemical ontology 2:603
chemical oxygen demand (COD) 3:1
chemical probing, combining systematic mutation with 3:579
chemical shifts 2:515
chemical similarity and substructure searches 2:640–641, 2:646
classification of datasets 2:646
graph representations of molecules 2:641
graph theory 2:640–641
identification of similar molecules and prediction of properties 2:645–646
maximum common subgraph problems 2:642–644
substructure search 2:641–642
variations of maximum common subgraph problem 2:645
chemical space (CS) 2:604, 2:604–605
chemical syntheses and retro-syntheses 2:609, 2:609F
chemical toxicity databases 2:634–635
chemical-protein interactome (CPI) 3:780–781
Chemistry at HARvard Macromolecular Mechanics (CHARMM) 1:63–64, 2:552, 3:613
ChEMistry TRaNslator (CHMTRN) 2:611
chemoinformatics 2:601, 2:601–602, 2:603–604
bioinformatics in drug design 2:614
computer-assisted synthesis design 2:610–611
current practices and capabilities 2:614–615
between descriptors and properties 2:603
drug-likeness and druggability concept 2:614
internet resources for chemistry and 2:614
ligand based design 2:613
mapping structure to property in QSAR approach 2:613–614
operations on synthons 2:610
precision and complexity in 2:602F
in silico chemistry 2:609
in silico functionalities 2:605
storing information to advanced ontology formalism 2:604–605
structure based design 2:613
synthon nomenclature 2:610
teaching computers chemistry 2:602–603
chemokine signalling pathways 3:100
chemotaxonomy 2:1124
chemotypes for FLT3 inhibition 3:755–757
ChemSpider 2:631, 2:1090
ChemSpider Synthetic Pages (CSSP) 2:631
Chenomx 3:470
chest pain (cp) 1:379
chi-squared/square (χ^2) 2:667
based discretization 1:469
test 3:444–445
test of independence 1:704
ChIA-PET and Hi-C methods 2:301
ChIA-PET/Hi-C methods 2:301
considerations of ligation 2D mapping 2:301–302
3DFISH *vs.* ligation products 2:301
chicken (*Gallus gallus*) 2:252, 3:428
chikungunya diagnostics, EQA of 3:13–14
ChiliBot tool 2:544
Chilo suppressalis 2:346
ChiloDB 2:346
chimeric proteins 3:966–967
China Cancer Genome Database (CCGD) 2:1082
chip analysis methylation pipeline (ChAMP) 3:344, 3:809, 3:810F
methylation analysis using 3:810
ChIP-exo 3:76, 3:79F
challenges with GEM and MACE 3:78
comparing data formats and peak detection methods performance 3:80–81
data analysis of Fkh1 and Fkh2 TFs 3:78
data integration and annotation 3:78–80
increasing stringency for TF binding significance 3:82–84
investigating discrepancy between OL7 and AD10 3:81–82
read length determination 3:78
system-level insight from “common core” of retrieved target genes 3:84–85
unique sets exploration of Fkh targets unravels regulatory links 3:85–87
ChIPMunk 3:449
Chipster 2:335
chitin
metabolic process 1:829
structural features of chitin polysaccharides 3:525
Chlamydia trachomatis 3:261
chlorophyll-a 3:1, 3:2
predictive model assessment 3:4–5
chloroplastic targeting signal prediction 2:55–56
chloropleth maps 3:6–7
choanoflagellates 2:262
choice model 1:744
choosing optimal-interval SNP set (CHOISS) 2:119
ChopClose automated protocol 2:473–474, 2:475T
Chou-Fasman method 3:518–519
chromatids 2:142
chromatin 2:308
chromatin 3D interactions 2:288–292
flexibility of chromatin 2:292–294, 2:293F
local structure of nucleosomes 2:288–292
role of CTCF protein and cohesin in genome organization 2:294–295
computational models of mechanism 2:302–303
methods for evaluating chromatin interactions 2:295–299
experimental methods 2:296–299
proteins 2:311
state 3:809
estimation 3:357
structure 3:807
chromatin contact domains (CCDs) 2:295
chromatin immuno-precipitation sequencing (ChIP-seq) 1:253, 1:272, 1:898, 2:149, 2:168, 2:204, 2:262, 3:162, 3:447, 3:449
alignment 3:113–114
correlation with 3:347
data 3:113
peak calling 3:114
pipeline 3:113–114, 3:114F
quality metrics 3:114
reproducibility assessment 3:114
chromatin immunoprecipitation (ChIP) 2:145, 2:183, 2:301, 2:798, 3:75, 3:75F, 3:347, 3:355, 3:808
data formats 3:76
read mapping for 3:357
chromatin interaction
calling mid-range or long-range chromatin interactions 2:321
Hi-C
data normalization 2:320
experiment review 2:318
paired read mapping 2:319
local structure calling 2:320–321
overview of Hi-C analysis pipeline 2:318–319
read-level filtering 2:319–320
3D modeling 2:322

- chromatin interaction (*continued*)
 visualization 2:321–322
- chromatin interaction analysis by paired-end tag sequencing (ChIA-PET) 2:182, 2:300F, 2:301
- chromatin related databases (Chromatin DB) 2:129
- chromatin remodellers, ATP-dependent nucleosome repositioning by 2:314
- chromatograms 2:1064, 3:878, 3:881F, 3:986
- ChromHMM software 3:323, 3:357
- Chromopainter method 3:369–370
- Chromopainter/fineSTRUCTURE approach 3:367–368
- chromosomal crossing-over 2:242
- chromosome conformation capture carbon copy (5C) 2:149, 2:300, 2:318
- chromosome conformation capture method (3C method) 2:149, 2:288, 2:294, 2:296, 2:299–300, 2:318
- “chromosome sweeping” approach 3:116
- chromosome-centric human proteome project (C-HPP) 2:1071, 3:920
- chromosomes 1:417, 1:418F, 2:142
- chronic inflammatory disease 3:729
- chronic lymphocytic leukemia (CLL) 2:933
- chymotrypsin-like protease 3:775–776
- CIGAR string 2:356
- cimetidine 3:744T–748
- CircNet 2:344
- CIRCOS 2:197–198
- circuit topology (CT) 2:522
- circular chromosome conformation capture (4C) 2:318
- circular dichroism (CD) 3:520–521
- circular fingerprints 2:619, 2:621F
- circular layout algorithm 1:1020, 1:1089
- circular RNAs (circRNA) 2:326, 2:1081, 3:808
- circular RNA/small non-coding RNA databases 2:344–345
- circulatory blood flow 2:903
- Cis-acting lncRNAs 2:168
- cis*-acting RNA elements (CAEs) 2:223
- cis*-antisense transcription 3:842–844, 3:843F
- cis*-interactions, 2:128
- Cis*-regulatory elements 2:155
- CisGenome software 3:114
- CisOrtho 2:285
- citrullination in RA cycle 3:730–731
- citrulline 3:729
- city block distance. *see* Manhattan distance
- clade B HIV-1 3:246–247, 3:247T
- class, architecture, topology & homology database (CATH database) 2:103, 2:446, 2:467, 2:483, 2:1020, 2:1021F, 3:32, 3:33
- CATH-Gene3D resource 2:480
- class architecture topology and homology (CATH) 3:513
- class association rules (CARs) 1:397–398
- class switch recombination (CSR) 2:940
- class(es) 1:336, 2:1
- class P 2:814
- class-based frameworks 2:1154
- class-based representations and description logics 1:288
- disjointness constraints 1:794
- equivalence constraints 1:794
- imbalance 3:332–333
- classic alignment algorithms 3:295
- classical graph theory 2:802–803
- classical segmentation methods 3:128
- classical statistical tests 1:199
- classical T-test 1:694
- classification and regression trees (CART) 1:331, 1:342, 1:385–387, 1:387, 2:1004, 2:1005
- classification based methods 2:664
- classification based on association rules (CBA rules) 1:397–398, 1:398
- AR 1:397–398
- CMAR 1:398
- CPAR 1:398–399
- classification based on multiple association rules (CMAR) 1:398
- classification based on predictive association rules (CPAR) 1:398–399
- classification techniques 1:199
- classification trees 1:275
- classification-based methods 2:668
- classifier specificity 1:346, 1:548, 1:686, 2:197
- classifier(s) 1:336
- further considerations 1:385
- generalization and overfitting 1:385
- learning 1:384–385
- predictor *vs.* 1:384–385
- specificity 1:431
- testing and accuracy 1:385
- clay hypothesis 3:941
- client-centric approach 1:238
- clinical and translational science awards (CTSA) 1:154–155
- clinical data interchange standards consortium (CDISC) 1:130
- clinical data repository (CDR) 2:1049
- clinical data research networks (CRDNs) 1:154
- clinical database for kidney disease (CDKD) 2:1060–1061
- clinical genomics databases (CGD) 2:1059
- clinical interpretation of variants in cancer (CIViC) 2:186
- Clinical Pharmacogenetics Implementation Consortium (CPIC) 2:634, 2:1059–1060, 3:384
- clinical proteomics 3:911, 3:915T–917
- bottom-up and top-down proteomics 3:918
- electrospray ionization and matrix-assisted laser desorption/ionization 3:914–918
- enrichment of low abundance proteins 3:914
- human tissue samples 3:914
- multiparametric panel of proteome profiles 3:911
- personalized proteomics and precision medicine 3:918–919
- post-translational modifications in proteoforms 3:911–912
- proteoform bioinformatics 3:919–921
- single reactor in sample preparation 3:912–914
- clinical text analysis and knowledge extraction system (cTAKES) 2:1090
- clinical/health database 2:104
- ClinVar 2:104
- CLIPMERGE PGx 2:1053
- CLIPPER 2:803
- clique percolation method (CPM) 1:979
- clique(s) 1:85–86, 1:929, 1:934, 1:965, 1:978
- clique-relaxation 1:937
- clonal lineage analysis 2:946–947
- clone clustering 2:947
- clone library method 3:203
- clonidine 2:677
- Clopper-Pearson confidence interval 1:558
- closed frequent itemsets (CFI) 1:360–362, 1:361F, 1:370
- closed-reference method 3:188
- closeness centrality 1:81, 1:83–84, 1:937, 1:952–953, 1:954, 1:962
- Clostridium difficile* 3:17, 3:26, 3:926
- cloud 2:1154–1155
- cloud-based bioinformatics platforms 1:257–258
- cloud-based bioinformatics tools 1:252–254
- open challenges 1:254–255
- cloud-based molecular modeling systems
- cloud-based services for molecular modelling tasks 1:262–263
- molecular modelling tasks 1:261–262
- of clouds 1:239
- federation 1:239
- identity & access management 1:242
- IDEs 1:247
- infrastructures 1:240
- load balancing 1:242
- privacy in
- access linkage problem 1:267–269
- problem of 1:266–267
- related work 1:265–266
- services 1:257, 1:258
- storage 1:241
- tools for IntelliJ 1:248
- Cloud Bigtable 1:241
- cloud computing 1:160–161, 1:161, 1:163–164, 1:236, 1:252, 1:257, 1:261, 1:267, 1:1099, 2:1049
- cloud controller (CLC) 1:245
- Cloud Dataflow 1:242
- Cloud Datastore 1:241
- Cloud Pub/Sub 1:242
- Cloud Virtual Network 1:242
- Cloud4SNP tool 1:172–174, 1:253
- CloudAligner algorithms 1:252
- CloudBurst algorithms 1:252
- CloudDOE (platform-independent software package) 1:259
- CloudFormation 1:240
- CloudGene (platform) 1:259
- CloudRS 1:253
- CloudWatch 1:240
- Clustal Omega 3:426

- Clustal programs 1:30
 Clustal ω 1:30
 ClustalW 1:30, 3:257
 CLUSTALW 1:17
 ClustalX 1:30
 cluster controller (CC) 1:245
 cluster domains (CDs) 3:415–417
 cluster identification by connectivity kernels (CLICK) 1:441
 cluster of differentiation (CD) 2:953–954
 cluster(ing) 1:95, 1:199, 1:230, 1:278, 1:338, 1:350, 1:437, 1:437–439, 1:439–441, 1:442–443, 1:444–445, 1:449, 1:456, 1:466–467, 1:474, 1:482–483, 1:745, 1:983, 1:1037–1038, 1:1067, 2:121, 2:389–390, 2:390F, 2:405, 2:664, 2:1032, 2:1102, 2:1103, 3:128–129, 3:197, 3:205, 3:456, 3:796–797
 algorithms 1:452, 1:1036–1037
 drawbacks of traditional clustering algorithms 1:1037
 algorithms for network clustering 1:95–96
 cells 3:816
 clustering-based method 2:1036–1037
 clustering-based segmentation 2:1032–1033
 coefficient 1:936, 1:965, 1:970, 1:979, 2:816, 2:823
 compactness 1:439
 of computers 1:162
 computing 1:210
 data classification 1:1036F
 DBSCAN 2:390–391
 DNA barcodes into OTU 3:991
 ABGD 3:991–992
 BOLD barcode index number system 3:991
 bPTP 3:992
 NJ in MEGA 3:992
 evaluation and interpretation of clustering results 2:391
 external criteria 1:445
 of graphs 1:95
 hierarchical clustering 1:278, 2:390
 internal criteria 1:444–445
 isolation 1:439
 K-means clustering 1:278–279, 2:389–390
 layout algorithm 1:1021, 1:1090
 property 1:95
 quality assessment 1:449–450
 sampling 1:466, 1:483
 significance assessment 1:450
 validation criteria 1:1043–1044
 clustered regularly interspaced short palindromic repeats (CRISPR) 3:207–208, 3:635
 ClusterFlow 3:136–137
 clustering based on maximal cliques algorithm (CMC algorithm) 1:979
 clustering objects on subsets of attributes (COSA) 3:470
 ClusterOne method 1:979–980
 clusters of orthologous groups (COG) 3:18, 3:187–188
 clusters of tandemly duplicated genes (CTDGs) 3:965T
 clusterSim function 1:893
 Cn3D 2:528–529
 CNVnator 3:116
 CO-exPRESSed gene database (COXPRESdb) 1:114
 co-fractionation mass spectrometry (CF-MS) 2:835, 2:836, 2:849–850
 co-immunoprecipitation (Co-IP) 2:824
 co-occurrence analysis 1:593
 co-orthologs 2:260
 co-regulated genes 2:344
 COACH 1:979
 coalescent applications 2:749–750, 2:750–753
 designing coalescent-based experiment 2:754
 estimation of evolutionary parameters and model selection 2:753–754
 hypothesis testing 2:750–753
 simulators 2:750, 2:751T–752
 theory and extensions 2:746–747, 2:746F
 validation of analytical methods 2:753
 coalescent for varying populations 2:20–21
 CoalEvol 2:754,
 coalitional games 1:289–290
 coarse graining 2:560
 coarse-grained force fields 2:561
 coarse-grained protein structure models (CG protein structure models) 3:609–610
 COCACOLA tool 3:325
 coding chemical information into molecular descriptors 2:605
 coding chemical reactions 2:605–606
 coding coding prediction approaches 3:232–233
 coding DNA sequences (CDS) 1:132, 2:8, 2:196, 2:361, 3:187–188, 3:323–324, 3:580, 3:722
 coding gene 3:230
 coding RNA 3:230, 3:230–231, 3:232F
 coding sequence (CS) 1:133
 codon adaptation index (CAI) 3:328
 codon(s) 2:196, 3:323–324
 adaptation in composition of codon 3:328
 applications in systems and synthetic biology 3:329–330
 bioinformatics tools
 to analyze codon usage 3:329
 to optimize codon usage in DNA 3:329
 dialect 3:327
 GC content and codon-usage bias 3:327
 influence of tRNA abundance on codon bias 3:327–328
 models 2:714, 2:715
 ramp at translational initiation 3:328–329
 CODONCODE ALIGNER program 3:986, 3:988
 coefficient of determination 1:725
 coefficient of variation (CV) 1:651, 1:677, 1:685, 3:902–903
 coexpression networks 1:1016–1018, 2:158
 CoGI method 1:484
 COGNAC service 3:548, 3:551
 cognitive economy 1:878
 cognitive saliency 1:880
 COGNIZER 3:187–188, 3:207–208
 Cohen's kappa (κ) 1:552
 cohesin role in genome organization 2:294–295, 2:295F
 cohesion indexes 1:84–85;
 see also centrality indexes
 cliques 1:85–86
 components 1:86
 triads 1:85
 coiled-coils 3:504, 3:521–522
 cold deck methods 1:465
 cold glycerol-saline solution 3:465
 collaborative analysis platforms 2:1153–1154
 collaborative competition 2:311–314
 collagen triple helix 3:516–517
collect-fpkms 2:227
 collective variables (CVs) 2:554
 collision-induced dissociation (CID) 2:896, 2:1066, 3:857–858
 color histograms 2:999
 color-genetic NCut (C-GeNCut) 1:443
 colorectal cancer (CRC) 3:466, 3:471
 colored-motif 1:938
 Coloured Petri Nets (CPN) 2:922
 column family 2:1065T
 ComBat method 3:360
 combinatorial diversity 2:942
 Combinatorial Extension algorithm (CE algorithm) 2:472, 3:592
 Combined Annotation Dependent Depletion (Condel) 3:5
 Combined Annotation Dependent Depletion algorithm (CADD algorithm) 3:393
 Comma Separated Values (CSV) 1:1049
 comma-separated values (CSV) 2:696, 2:888
 command line interpreter (CLI) 2:527–528
 commercial systems for NGS 1:128–129
 ComMet software 3:356
 Common Data Model (CDM) 1:154
 common disease-common variant hypothesis (CD/CV hypothesis) 2:235, 3:374–375
 common factor analysis 1:745
 common workflow language (CWL) 2:1154, 2:1155F, 3:139
 common-neighbors-based GRAPh ALigner (C-GRAAL) 1:1004–1005
 communicating sequential processes (CSP) 1:210
 community 1:95
 challenges 1:607–608, 2:1104–1105
 cloud 1:238, 1:258
 discovery algorithms 1:96
 finding 1:982, 1:983
 mapping 1:1008
 role 3:68–69
 standards 2:853–854
 structure in networks 1:972–973, 1:972F
 community detection 1:967, 1:982, 1:1008
 assessment 1:980–981
 benchmarking and resource portals 1:984
 in biological networks 1:978–979
 biological networks 1:982
 case studies 1:982
 clustering *vs.* community finding 1:983
 complexome 1:984
 functional module 1:982–983

- community detection (*continued*)
 methodologies 1:979
 PPIN
 and dynamic data integration 1:983–984
 multispecies 1:984
 specialization 1:984
 and static data integration 1:983
 special types of communities 1:984
 COMODI ontology 2:889
 comparative binding energy analysis
 (CoMBINE) 2:669
 comparative epigenomics 2:203, 2:257–258,
 2:257F, 2:258–259, 2:265, 3:296,
 3:355–356, 3:358, 3:360, 3:425
 agricultural animals and plants 3:428–429
 animal models for human disease 3:429
 biased genomic data 2:259–260
 biological replicates 3:356
 chromatin state estimation 3:357
 comparative analysis 3:355–356
 comparison of DNA methylation
 between cell types 3:358–359
 and gene expression 3:358
 comparison of gene order 2:263
 data 3:358
 defining windows 3:356
 DNA methylation 3:356–357
 domestication and comparative animals
 2:264
 epigenetic modification 3:354–355
 evolutionary distances for comparative
 analysis 2:258–259
 experimental technology and raw data
 processing 3:355
 gene expression 3:357
 genome structure and types of change
 2:257–258
 histone modification and transcription
 factor binding 3:357
 homologous genes 2:260
 of intergenic regions 2:262–263
 methods in 3:426
 prokaryotes 3:426–428
 public databases 3:357–358
 sequence to genomes 3:425–426
 software 3:360–361
 workflow of 3:427F
 comparative modeling 2:585
 comparative molecular field analysis
 (CoMFA) 2:664, 2:669
 comparative molecular moment analysis
 (CoMMA) 2:669
 comparative molecular similarity indices
 Analysis (CoMSIA) 2:669
 comparative residue interaction analysis
 (CoRIA) 2:669
 comparative RNA Web server (CRW) 3:12,
 3:16–18
 comparative spectral analysis (CoSA) 2:669
 comparative toxicogenomics database (CTD)
 2:103, 2:175, 2:634–635, 2:1048
 comparative transcriptomics analysis (CTA)
 3:814, 3:816
 for integer numbers 3:817
 for real numbers 3:817
 statistical models 3:814
 compendium of protein lysine modifications
 (CPLM) 3:35
 competing endogenous RNAs (ceRNA)
 2:326, 2:1081
 complementarity determining region (CDR)
 2:909, 2:940, 2:953, 3:634
 complementary DNA (cDNA) 1:127, 2:364,
 2:388, 3:231–232, 3:976–977
 complementary metal-oxide-semiconductor
 (CMOS) 3:435
 complete bipartite graph 1:930
 complete chromosomal duplication 3:965
 complete gene duplication 3:965, 3:966
 complete genomics systems 3:434
 complete graphs 1:929, 1:930F
 complete linkage 1:350–351
 complex daubechies wavelet transform
 (CDWT) 2:999
 complex disease-related haplotype
 identification 3:444
 complex diseases 2:343
 complex enrichment analysis tool
 (COMPLEAT) 2:129
 complex pathways simulator (COPASI)
 2:920
 complex proteins 3:511
 complex-complex networks (CCIN) 1:984
 complexity parameter. *see* a criterion
 complexome 1:984
 COMPLIG (Fast heuristic graph match
 algorithm) 3:495
 component directed graph 1:935
 composite elements (CEs) 2:135
 composite likelihood ratio test (CLR) 3:372
 composite module analyst (CMA) 2:135
 composite motifs. *see* structured motif
 composite performance measures 1:549–550
 composite scoring 3:372
 composition based languages 1:219;
 see also object-oriented parallel languages
 composition profile of patterns (CPP) 3:670
 composition-based binning 3:187
 compound ID (CID) 2:630–631
 compound library preparation 2:588–589,
 2:589T
 comprehensive R archive network (CRAN)
 1:200, 1:332
 comprehensive species-metabolite
 relationship database 3:470
 compress protein scheme (CP scheme)
 1:483
 computable phenotyping model 1:154;
 see also bioinformatics data models
 analysis 1:155
 applications 1:154–155
 case study 1:155–156, 1:157F
 data quality 1:156–157
 health research networks 1:158
 computation cluster validation in big data
 era 1:449–450, 1:452–453
 computational algorithms 2:314
 computational analysis/methods 1:706,
 1:988, 2:825, 3:1–3, 3:1–2, 3:2F,
 3:107, 3:107–108, 3:252
 of constructing RNA supramolecules
 3:540–543
 in drug target identification 2:1094
 of HGT genes 3:954
 for identification of secondary structures in
 proteins 3:517–519
 assigning SSEs to 3D structures 3:519–
 520
 prediction of secondary structures using
 protein sequences 3:517–519
 for investigating protein-DNA interactions
 2:149–150
 of mechanism 2:302–303
 to predicting PPIs 2:840
 for studying dynamics of CRISPR-Cas9
 system 3:635
 for studying miRNAs 2:124–125
 in systems biology 2:874
 computational biology 1:30, 1:130, 1:313,
 3:247
 computational complexity theory 2:814
 computational designing of β -sheet surfaces
 to boost protein stability 3:636
 computational docking 1:78
 simulation 3:761
 computational functional analysis
 of lncRNAs 2:188
 of miRNAs 2:188
 computational immunogenetics 2:906
 computational intelligence method 2:1038–
 1039
 computational intelligence-based
 segmentation 2:1033–1034
 computational lipidomics 2:894–895,
 2:895F
 identification and quantification of lipids
 2:896
 lipids and functions 2:894
 mass spectral analysis and peak detection
 2:895–896
 next dimension 2:896–897
 Computational Modeling in Biology/
 Network (COMBINE) 1:130, 2:884
 computational models of proteins and
 assemblies 2:540–541
 computational modification of periplasmic
 binding proteins 3:638
 computational pipelines
 example 3:139–140
 GitHub 3:141
 hardware levels, project sizes and user
 experience levels 3:135
 pipeline tools 3:135–136
 computational prediction of TM segments
 and topology 2:47–48
 machine-learning based methods 2:48–49
 scale-based methods 2:48
 computational protein engineering
 approaches 3:632–634, 3:633T,
 3:638–639
 addition of novel function to existing
 proteins 3:636–638
 alteration in specificity of macromolecules
 by structural modifications 3:632–634
 computational approaches for studying
 dynamics of CRISPR-Cas9 system
 3:635

- computational design of zinc finger proteins 3:633–634
 re-engineering of proteins 3:635–636
 structure-based computational engineering of therapeutic antibody 3:634–635
 computational proteomics analysis system (CPAS) 3:866
 computational resources 1:1050
 for drug repurposing 3:780–781
 computational systems biology 3:71
 case studies in multi-dimensional data analysis 3:66–67
 controllability/network motif analysis 2:790–791
 deterministic *vs.* stochastic approaches to modeling 2:791
 digital drivers at intersection of technology and medicine 3:66–67
 models and methodologies in 2:789F
 network based approaches 2:790
 network inference 2:791
 pathway curation 2:789–790
 standards 2:884–885
 data used for model construction 2:884–885
 recording modeling results 2:888
 recording simulation environments 2:887–888
 structuring models of biological systems 2:885–886
 visualizing models 2:886–887
 computational temperature 1:54–55
 computational tools 3:874
 for drug repurposing 3:781–783
 for gene-gene interactions 2:118
 to predicting nucleosome positioning 2:312T–313
 Compute Engine 1:241
 compute management services 1:259
 compute unified device architecture (CUDA) 1:165–167, 1:166F, 1:209, 1:211
 programming languages 2:1142
 sum of two arrays in 1:168F
 computed tomography (CT) 2:993, 2:1007
 computer algorithms 3:243
 computer generated chemical names 2:606
 computer processing unit 2:525–526
 computer-aided drug design (CADD) 2:677, 3:652, 3:749–751, 3:751–752, 3:755F
 advantages 3:749–751
 for EGFR protein controlling lung cancer 3:755
 fragment-based drug design to target EphA4 kinase domain 3:757
 ligand-based drug designing 3:753
 limitations 3:757–758
 SBVS, for identification of new chemotypes for FLT3 inhibition 3:755–757
 structure-based drug discovery 3:752
 computer-aided molecular design (CAMD) 3:742
 computer-assisted synthesis design 2:610–611, 2:611F
 adopting mathematics to chemistry and chemistry to mathematics 2:612
 computer assisted knowledge discovery 2:611–612
 computer assisted molecular design 2:612
 computer assisted structure elucidation 2:611
 from data to drug candidates and drugs 2:612–613
 computer-processable molecular codes 2:605
 computing shortest path algorithms 1:940
 concentration free outlier factor (CFOF) 1:458
 concept layer 1:587
 concept mining 1:586, 1:588–589
 concept unique identifier (CUI) 1:817
 concept-driven approaches 1:593, 1:594–595, 1:597T
 conceptual ontological graph (COG) 1:595, 1:597
 concurrent languages. *see* parallel programming languages
 concurrent neural network 2:1033–1034
 conditional mutual information (CMI) 2:161–162
 conditional neural fields (CNF) 1:68–69
 conditional random fields (CRF) 1:68–69, 2:48–49, 2:1085
 conditional statement 1:181–182
 Condorcet's Jurys Theorem 1:520, 1:539
 confidence interval (CI) 1:557, 1:658, 1:664T, 1:665T, 1:669–670, 1:710–714
 critical regions of standard normal distribution 1:711F
 for difference between population means 1:659–661
 for difference between population proportions 1:662
 for population mean 1:658–659
 for population proportion 1:661–662
 for population variance 1:662
 for ratio of variances of two normally distributed populations 1:662–663
 vital capacity data 1:712T–713
 confidence of association rule 1:360
 confidentiality 1:254
 configuration model. *see* fixed degree distribution random graph (FDDRG)
 conflict resolution and overlap problem 1:390
 conformational B-cell epitopes 2:917–919
 conformational epitope database (CED) 2:935
 conformational transitions observing in experimentally determined structures 3:606–607
 conformational variability 3:606
 confusion matrix 1:685–686, 1:686T
 conical correlation analysis (CCA) 2:818
 conjugate gradient method (CG method) 1:473
 conjugation 3:953
 connected component 1:964
 connection indicators 1:81
 connectivity map (CMAP) 2:1095
 connectivity search 2:636
 connectivity-based clustering method 1:1037–1038
 consensus methods 2:322, 2:588, 2:740–741
 consensus MQAP 3:595, 3:597, 3:599–600
 consensus sequence models 3:333–334, 3:334F
 consensus tools with combined features 3:4–5
 consensus variant effect classification (COVEC) 3:5
 consensus-based methods 1:564, 2:507–508
 ConsensusPathDB 2:852, 2:853, 2:1054
 conservation of duplex structures 3:580
 conserved domain database (CDD) 2:12, 3:31
 conserved exon method (CEM) 2:203
 conserved interactions (CI) 1:1013
 conserved linkage 2:258–259, 2:263
 conserved-syteny 2:258–259
 and collinearity 2:269–270
 conserved syteny-block 2:263
 ConSite 2:285
 constraint generation method (CG method) 2:576
 constraint programming (CP) 2:644
 constraint satisfaction 1:290
 constraint satisfaction problem (CSP) 1:290, 1:294
 constraints and databases 3:581
 ConStruct 3:510
 contact area differences (CAD) 3:593
 contact map overlap (CMO) 2:818–819
 contact maps 1:564
 contact order (CO) 2:451, 2:453
 Container Engine 1:241
 containerization 2:1155–1156
 content-based image retrieval (CBIR) 2:1018, 2:1020F, 2:1022, 2:1025
 bioimage databases using 2:1018–1019, 2:1019F
 solution for bioimage databases using 2:1024–1025
 context-free grammar (CFG) 1:561, 1:564–566
 contextual genes 3:60
 contigs 2:357, 3:325
 contingency matrix 1:431
 continuity 1:748
 Continuous Automated Model EvaluatiOn (CAMEO) 3:587
 continuous binding site properties 2:653
 continuous decision outputs, combining functions for 1:523
 continuous long read (CLR) 3:436
 continuous models 1:1053–1054
 continuous probability distributions 1:654–657
 continuous time 2:704
 continuous time Markov chain (CTMC) 2:712
 continuous wavelet transforms (CWT) 1:481
 continuum Bienaymé-Galton-Watson branching model 2:24–26
 continuum WF model with mutations 2:11–12, 2:12F
 and selection 2:12–13

- continuum WF model with mutations
(*continued*)
- contour-based method 2:1035–1036
- contour-based segmentation 2:1031–1032
- contrast-limited adaptive histogram
equalization technique (CLAHE
technique) 3:128
- contrastive divergence algorithm 1:646
- Contributes_to* 2:3
- control docking method 2:693–694, 2:694F
- control effective fluxes and structural fluxes
2:442
- control ligand preparation 2:690–691
- control unit (CU) 1:161
- controllability analysis 2:790–791
- controlled vocabulary 1:813
for adding semantic information 2:888–
889
- convention on international trade in
endangered species (CITES) 2:1111
- conventional design paradigm 2:1153
- conventional MD simulations 2:553, 2:554
- conviction 1:362
- convolution neural network model (CNN
model) 1:281, 1:641–642, 2:793,
2:1006, 2:1033–1034
- convolutional neural deep networks
(ConvNets) 1:304
- coordination 1:316
- COordination of Standards in
MetabOlomicS (COSMOS) 2:399,
2:431–432
- COPASI (software application) 1:1051
- copy number aberrations (CNAs) 3:485–486
- copy number variants/variation (CNVs)
1:706, 1:1104–1105, 2:2, 2:175,
2:185, 2:1080, 3:116, 3:168, 3:176,
3:324, 3:376, 3:388–389, 3:965T
- copying model 1:971–972
- Corchorus capsularis* 3:428
- Corchorus olitorius* 3:428
- core decomposition of graph 1:979
- core eukaryotic genes mapping approach
(CEGMA) 2:181
- core hypothetical proteins in *B. pseudomallei*
3:957–959
hypothetical genes conserved in
Burkholderiaceae family 3:957–959
- core index 1:979
- core-clustering coefficient 1:979
- Core&Peel algorithm 1:980
- CoreBoost_HM tool 3:323
- CoReCG 2:342
- correlation 1:367, 1:689, 1:714–717
- correlation analysis 1:477, 1:706, 3:3;
see also functional enrichment analysis
- clinical data 1:718–719
- correlation and regression 1:714–717
- data analysis and results 1:717–718
- measurement 1:706–708
- simulated data 1:717–718
- software 1:719
- correlation coefficient(s) 1:708–709, 2:392
- Kendall-s 1:709–710
- PPMCC 1:709
- Spearman's rank 1:709
- correlation feature selection method (CFS
method) 2:1004
- correlation rules 1:362
- COrelation Spectroscopy (COSY) 2:427–
428, 2:515
- correspondence analyses (CA) 3:329
- CORUM 2:851
- COSMIC 2:1079, 3:116
- COSMIC 3D 2:468
- cross-linking immunoprecipitation
sequencing 2:333
- cost analysis 2:682
- cost function 1:440–441
- cost-based local search 1:96
- COSY technique 3:525
- COTRAN 3:667
- counts-per-million (CPM) 2:380, 3:455
- coupling cell division to metabolic pathways
3:90
- categorization according to KEGG pathway
3:90T
- ChIP data formats 3:76
- ChIP-based methodologies 3:87–90
- ChIP-exo data analysis of Fkh1 and Fkh2
TFs 3:78
- Fkh1 and Fkh2 target genes 3:91F
- interplay between DNA and proteins 3:74–
75
- metabolic enzymes in central carbon
metabolism 3:89F
- methodologies to investigate DNA-TF
interaction 3:75–76
- stack plot representing the Fkh2 target
genes 3:88F
- timing of cell division 3:74
- course-graining methods 2:901
- covalent bond terms 1:50–53
- covariance 1:706, 1:706–708
- covariance models (CMs) 1:280–281, 2:1018
- covariates 1:731
- covariation of duplex structures 3:580
- covarian models 2:715–716, 2:716F
- COXPRESdb 2:1059
- CpG Island (CGI) 1:760, 2:128, 2:1081,
3:324
regions 3:341
related databases 2:129
- CpG island Promoter Detection (CpGProD)
2:129
- CpGcluser tool 2:129, 3:324
- CpGTLBO tool 3:324
- CR Cistrome 2:129
- CRASA 2:202
- create, read, update and delete (CRUD) 2:96
- Crick's central dogma 3:504
- crisp classifier 1:546
- CRISPR associated protein 9 (Cas9) 3:635
- CRISPR recognition tool (CRT) 3:207–208
- CRISPR-Cas9 system 3:635
- critical assessment of epitope predictions
(CAEP) 2:967
- critical assessment of function annotation
(CAFA) 2:13, 2:483, 2:544
- Critical Assessment of PRedected
Interactions (CAPRI) 1:78–79, 1:232,
2:13, 2:503, 2:825, 2:842
- critical assessment of structure prediction
(CASP) 1:62–63, 1:232, 1:562, 2:13,
2:497–498, 2:500–501, 2:540, 2:967,
3:254, 3:587, 3:613
- ROLL 3:587
- Critical Micelization Concentration (CMC)
2:670
- crizotinib 3:744T–748
- crop plants 2:347
- cross entropy 1:639
- cross industry standard process for data
mining (CRISP-DM) 1:337
- cross validation 2:667, 3:333
- cross-entropy 1:551
- cross-linking coupled with mass
spectrometry (XL-MS) 2:824
- cross-network probabilistic consistency
1:1010
- cross-organism databases 2:251–252
- cross-population extended haplotype
homozygosity (XP-EHH) 3:371
- cross-validation (CV) 1:274, 1:536–537,
1:542, 3:120–121
concepts and notation 1:542
- k-fold cross-validation 1:543–544, 1:543F
- k-fold random subsampling 1:543
- LOOCV 1:544, 1:544F
- single hold-out random subsampling
1:542–543
- Crossbow tool 1:253
- crossing edge minimization 1:1088–1089
- crosslinking immunoprecipitation (CLIP)
3:50–53
CLIP-seq 2:333
- crossover connection 3:514–516
- crunching factor 1:233
- cryo-electron microscopy (Cryo-EM) 2:515,
2:835–836, 3:6, 3:6–7, 3:10, 3:12,
3:274, 3:521, 3:586, 3:613, 3:722
- cryo-electron tomography 3:521
- Cryptococcus neoformans* 3:928
- cryptoendoliths 3:945–946
- crystallins 1:870
- crystallizability 2:30
- crystallographic methods 3:620
- Csbl. go package 1:893
- cSynechococcus sp.* PCC 7002 omics studies
2:345
- CTCF protein role in genome organization
2:294–295, 2:295F
- Ctenopharyngodon idellus* (grass carp) 2:345
- culture dependent methods 3:202–203
- culture-based approach 3:17–18
- culture-independent method 3:203
- cumulative absolute frequency 1:673
- cumulative distribution function (CDF)
1:766, 1:769
- cumulative relative abundance distribution
(CRAD) 3:21, 3:22F
- curatedOvarianData* 2:342
- curators 2:3–4
- curdian sulphate 3:775
- Curoverse 3:139
- curse of dimensionality 1:466–467, 1:486
- Curves+ 3:510
- Cutadapt 3:166, 3:296

cyanobacteria, ancestors of 3:946
 CyanOmics database 2:345
 cyber security 1:336
 cyber-physical systems (CPSs) 1:309
 Cycle-Cloud 1:262
 cyclic AMP singling pathway 2:796
 cyclic citrullinated peptides (CCP) 3:730
 cyclic neural networks 1:621
 cyclic reversible termination (CRT) 3:435
 cyclin-dependent kinase 1 (Cdk1) 3:74
 cyclin-dependent kinase 2 (CDK2) 2:680
 cyclo-oxygenase 3:741
 CYP2D6*10 allele 3:384–385
 CYP2D6*4 allele 3:384–385
 Cypher 2:98–99
 cystic fibrosis transmembrane conductance
 regulator (CFTR) 2:171
 cytochrome c oxidase subunit I (COI) 3:985
 cytochrome P450 (CYP450) 3:379, 3:383,
 3:947
 CytoHiC 2:322
 cytokines 2:920
 cytoscape 1:1028–1029, 3:50–53, 3:120
 cytosine 2:701, 3:425
 methylation 3:324
 modifications 3:341, 3:342F
 determining cytosine modification
 proportions 3:346
 genomic distribution 3:347
 cytotoxonomy 2:1124
 cytotoxic necrotizing factor 1 (CNF-1)
 3:726–727
 cytotoxic T lymphocytes (CTL) 2:910
 Cyworld 1:89

D

D.E. Shaw Research's (DESRES) 2:1146
 DACH gene 2:262–263
 DALI program 3:724
 Dali server 2:472T, 3:593
 DaliLite search 2:474
 Dana-Farber Repository for Machine
 Learning in immunology (DFRMLI)
 2:935, 3:40
 dangling read pairs 2:319
 Darpa Agent Markup Language plus
 Ontology Inference Layer (DAML
 +OIL) 1:809
 DARPA's Big Mechanism Project 2:790
 darunavir 3:744T–748
 Darwinian evolution 3:938, 3:942–943,
 3:942F
 data
 acquisition 3:465
 analytics 2:793
 augmentation 1:641
 characteristics of targeted metabolomics
 experiments 2:415, 2:415F
 clustering applications 1:443–444
 compression 1:483
 conversion 3:859
 curation based approaches 2:797–798
 data-induced tree 1:458

data-centered ensemble 1:460
 data-integration techniques 1:1104
 data-mapping directives 1:217
 data-structured sources 1:586
 dimension 2:404
 discretization 1:468–469
 frame syntax 1:202
 in life science 1:1104–1105, 1:1106F
 input 2:404
 layer 1:587, 1:883–884
 management services 1:259
 management system 2:397–399
 models 2:1073
 normalization 1:467, 2:404–405, 2:418–
 419
 preparation 1:337–338, 3:441, 3:458
 pretreatment 3:469–470
 provenance 2:3
 quality assessment 1:744
 reliability 1:992
 resampling methods 1:542
 science in Python 1:197–198
 selection 1:329
 services registry 2:86
 sharing 1:1076–1077
 smoothing 1:474
 sources 1:907
 structures
 in Java 1:207
 in Python 1:195–196
 transformation 1:329, 1:358, 1:467, 1:472,
 1:477, 1:477–478, 1:480
 types in Java 1:206–207
 understanding 1:337
 visualization 2:1063, 2:1063–1064, 2:1069,
 3:887
 warehouse/warehousing 1:1092, 1:1095–
 1096, 1:1095F, 2:99–100, 2:100F
 data analysis 3:78, 3:858
 computation tools 3:858–859
 de novo and hybrid search approaches 3:863
 sample data set 3:858
 software 3:873–874
 Data analysis, visualization, and exploration
 (DAVE) 2:1079
 Data as a Service (DaaS) 1:172–174, 1:252
 data cleaning 1:328, 1:463–464, 1:472,
 2:404
 duplicate data detection 1:465–466
 handling missing value 1:464
 methods for missing values in
 bioinformatics 1:472–473
 methods for noisy data in bioinformatics
 1:474–475
 outlier detection 1:475
 data control language (DCL) 2:1073
 data coordinating center (DCC) 2:108
 data definition language (DDL) 2:1067T,
 2:1073
 data dependent acquisition (DDA) 2:1069,
 3:855
 data elaboration 1:114–115
 algorithms for 1:115
 data formats
 BioPAX 1:1063–1064
 KGML 1:1065

languages 1:1065
 metadata, controlled vocabularies, and
 ontologies for adding semantic
 information 2:888–889
 PSI-MI XML 1:1064
 SBCN 1:1065
 SBML 1:1064–1065
 standards for computational systems
 biology 2:884–885
 survey and synopsis of modeling formats
 2:889–890
 data independent acquisition (DIA) 2:84,
 2:417–418, 2:1069, 3:855, 3:872–
 874, 3:874T, 3:906
 computational tools 3:874
 data analysis software 3:873–874
 data exploration and analysis 3:880–884
 peptides selection for quantification
 3:882–884
 data extraction environment 3:877–879,
 3:879F
 fragment ion library or assay library 3:875–
 877
 and MS2 quantification 2:90
 sample dataset 3:874
 spectral identification and library
 generation 3:874–875
 targeted data extraction 3:879–880
 peak picking and scoring 3:879–880
 targeted proteins 3:877
 data integration 1:329, 1:469–470, 1:472,
 1:477, 1:988–991, 1:1092–1093,
 2:333–334
 and annotation 3:78–80
 article curation 1:991
 integrating multiple PPI databases and
 predicted PPIs 1:991
 mapping to common identifiers 1:991
 methodologies and tools for integrative
 analysis 1:477
 schemas integration 1:477
 standardizing data format and vocabulary
 1:988–991
 data manipulation language (DML) 2:1067T,
 2:1073
 data mining 1:114, 1:338, 1:358, 1:358–360,
 1:456, 1:463, 2:101, 2:121, 2:545,
 2:1005, 2:1083
 accuracy measures
 in classification 1:431–432
 in regression 1:432–433
 associative classification 1:397–398
 in bioinformatics 1:328–329, 1:329F
 classification 1:384
 by decision tree induction 1:385–387
 model construction 1:384
 by multilayer FNN 1:390
 role of features 1:384
 closed and maximal frequent itemsets
 1:360–362, 1:361F
 correlation and implication rules 1:362
 frequent itemset
 and association rules 1:359–360
 mining algorithms 1:362–363
 improving accuracy 1:434–436

- data mining (*continued*)
- learning, testing, accuracy, overfit 1:384–385
 - model selection and assessment 1:433–434
 - outlier detection 1:456, 1:456–457
 - platforms 1:331
 - program environment for 1:331–332
 - rule-based classification 1:388
 - SVM 1:393–395
 - task 1:329–331
- data mining cloud framework (DMCF) 1:249
- data mining ontology for grid programming (DAMON) 1:811
- data pre-processing 1:472
- untargeted metabolomics 2:418
- data processing 2:1067
- quality assessment 3:792–795, 3:794F
 - workflow in MS-based proteomics 2:84–86
- data processing inequality (DPI) 2:158
- data reduction 1:466, 1:472, 1:480
- dimension reduction 1:466–467
 - parametric 1:466
 - sampling 1:466
- data storage
- biological networks as expressive data models 1:1104–1105
 - biomedical data 1:1099–1100
 - data models for biological data 1:1100–1102
 - in public repositories 3:866
- data-analytics development environments 1:247
- BigML 1:250
 - microsoft Azure ML 1:250
- data-dependent acquisition (DDA) 2:84, 3:873
- and MS1 quantification 2:89–90
- data-driven approaches 1:592, 1:593, 1:596T
- data-information-concept continuum from text mining perspective
- concept mining background 1:588–589
 - concept-driven approaches 1:594–595
 - connections and relations among main research domains 1:589–591
 - information retrieval 1:591–592
 - NLP 1:589–591
 - semantic web 1:591
- data-driven approaches 1:593
- information-driven approaches 1:593–594
- Layered Multi-Granule Knowledge Model 1:586–588
- multi-layer knowledge representation 1:595–598
- database for annotation, visualization and integration discovery (DAVID) 2:129, 3:220
- knowledgebase 2:344
 - pathway databases 3:822
- database management systems (DBMS) 1:1096, 2:1066–1067
- database of arylamine N-acetyltransferases (NATs database) 3:384
- database of CpG islands and analytical tools (DBCAT) 2:129
- database of curated mutation (DoCM) 2:1059
- database of decoys tools (DUD-E tools) 2:681–682
- database of genetic variants (DGV) 2:1059
- database of genotypes and phenotypes (dbGaP) 2:104
- database of interacting proteins (DIP) 1:477, 1:1105, 3:1–2
- Database of Plant Biodiversity of West Bengal (WBPBDIVDB) 2:1112–1113
- database of small human noncoding RNA (DASHR) 2:125, 2:345
- database of transcription start sites (DBTSS) 2:120
- database of transcriptome in mouse embryos (DBTMEE) 2:345
- database(s) 2:96, 2:335–337, 2:931–932, 2:1066–1067, 3:292–293, 3:382, 3:383T
- architecture 2:109
 - and computational resources for drug repurposing 3:780–781
 - functionality 1:993
 - hosting immune epitopes 3:40–42
 - languages 2:1067T
 - models 2:109
 - search programs 2:202, 2:1068–1069, 3:860–862
 - system architectures 2:99
- DataBEL 3:382
- datagram sockets 1:218
- Datalog⁺ 1:786
- dataset clause 1:800
- and named graphs 1:801
- dataset for machine learning and performance measure 1:273–274
- dating gene duplication and computations 3:971
- case studies 3:971–972
 - tools and methods 3:972
 - unitary pseudogenes 3:971
 - unprocessed pseudogenes 3:971
- Davies-Bouldin index 1:444
- Dayhoff PAM matrices 3:333, 3:337
- DBChIP 3:115
- dbEST database 3:820
- dbPTB database 2:343
- dbSNP database 3:116, 3:169–170, 3:299
- dbWFA 2:347
- dChip-Gene and miRNA Network-based Integration (dChip-GemiNi) 1:1100
- DChIPRep 3:809
- DDBJ Sequence Read Archive (DRA) 3:30
- de Bruijn graph method (DBG method) 2:148, 2:181, 2:181F, 2:330, 2:358, 3:161, 3:177, 3:302–304, 3:304
- de novo* 1:774, 2:540, 3:188
- assembling metagenomes 3:305
 - assembly 2:181, 2:182F, 2:357, 2:1143, 3:177, 3:186–187
 - using long reads 3:301–302
 - of microbial genomes 3:305
 - to reference genome 3:437
 - using short reads 3:302–304
 - assembly of eukaryotes 3:305
- gene prediction. *see ab initio-gene prediction*
- genome assembly 3:301–302, 3:304–305
- hybrid assembly 3:304
- and hybrid search approaches 3:863
- ligand designing 3:749
- pathway discovery 1:1075–1076
- proteins 3:644, 3:644F
- modeling 3:631, 3:632F
- sequencing 2:1068
- transcriptome assembly 3:309–310
- dead-end elimination (DEE) 3:631
- Debye-Huckel model 2:143
- decimal scaling normalization 1:467, 1:478
- decipher STNs for drug effect in complex disease 2:861
- decision support system (DSS) 1:1105–1106, 2:1049
- decision trees (DTs) 1:302, 1:302F, 1:343, 1:374, 1:374–375, 1:375F, 2:913, 3:129
- induction 1:385–387
 - attribute selection measures 1:387–388
 - build tree 1:386–387
 - literature applications 1:381–382
 - training algorithm 1:374–375
 - use-cases 1:377–379
- dEclat 1:363
- decoding problem 1:758–759
- deconvolution 2:419
- deconvolving batch effects 3:796
- decoy database builder (DecoyDBB) 3:862
- DecoyPYrat by Sanger Institute 3:861
- decoys 2:167, 2:507
- decrease and conquer problem 1:6–7
- deep architectures 1:634
- deep belief networks 1:644–646
- deep Boltzmann machines (DBM) 1:646
- deep convolutional nets 1:304, 1:304F
- deep kernel strategies 1:500–501
- deep learning (DL) 1:634, 1:634–635, 1:641, 2:1006, 2:1147
- architectures 1:641–642
 - learning DNN 1:638–640
 - multi-layer networks and deep networks 1:635–638
 - Perceptron 1:634–635
 - traditional machine learning *vs.* 1:638F
- deep networks 1:635–638
- deep neural networks (DNNs) 1:634, 1:638, 1:641, 2:1033–1034
- learning 1:638–640
 - activation functions 1:640–641
 - gradient descent and Backpropagation 1:638–640
 - gradient descent optimization algorithms 1:640
 - loss functions 1:639–640
 - shallow *vs.* deep learning 1:641
 - traditional machine learning *vs.* 1:638F
- deep recurrent neural networks (DRNN) 1:281
- DeepBind 3:449
- DeepBlue Epigenomic Data Server 3:341
- DeepVariant model 2:793, 3:66
- defensins 2:906

- Define Secondary Structure of Proteins (DSSP) 2:490
 DEFINE-S algorithm 3:519
 Deformation index (DI) 3:596–597
 deformation profile (DP) 3:597
 degenerate alphabets 1:18
 degree centrality 1:83, 1:937, 1:951, 1:953–954, 1:962
 degree correlation 1:965–966
 degree distribution 1:936–937, 1:962–964, 1:964F, 2:816
 degree-aware algorithms for network-based disease gene prioritization (DA DA) 2:811
 degrees of freedom (DOFs) 2:560
Deinococcus radiodurans 3:945–946
 delayed differential equations (DDE) 2:920–921
 delayed-type hypersensitivity (DTH) 2:931
 delivery models 1:258
 delta rule 1:621, 1:621–623
 deltaviruses 3:246–247
 denaturing high-performance liquid chromatography (dHPLC) 3:433
 denaturing or temperature gradient gel electrophoresis (DGGE/TTGE) 3:203
 dendritic cells (DC) 2:910, 3:106
 dendrogram 1:352, 1:442
 dendroscope 3:370
 dengue
 diagnostics
 EQA 3:13–14
 in WHO Western Pacific Region 3:11–13
 across eight countries in Southeast Asia 3:975–976
 outbreaks 3:11
 patient's blood as substitute for DSF 3:13
 serotype-specific differences 3:10–11
 transmission 3:975
 dengue fever (DF) 3:10
 dengue hemorrhagic fever (DHF) 3:9, 3:10
 dengue shock syndrome (DSS) 3:9, 3:10
 dengue virus (DENV) 3:9, 3:11, 3:246–247, 3:247T, 3:761, 3:762F, 3:763F, 3:764F, 3:975
 DENV-1 3:9, 3:10
 infection 3:11
 population reveal mutant spectra 3:976–977
 DENV-2 3:9
 non-structural proteins
 NS1 3:765
 NS2B–NS3 protease 3:765–768
 NS3 helicase 3:768–772
 NS4 3:772
 NS5 3:772–775
 protein targets 3:761–763
 serotype surveillance 3:11
 structural proteins
 C protein 3:763–764
 E protein 3:761–763
 denial-of-service attack 1:269–270
 denoising autoencoder 1:644
 dense genetic maps 2:242
 dense graphs 1:931F, 1:931
 “density connectivity” cluster model 1:1039–1040
 density functional theory (DFT) 1:78, 1:562
 Density-based clustering algorithms 1:1039–1040, 2:391
 density-based graphs 1:931
 density-based methods 1:442
 density-based outliers 1:457–458
 density-based spatial clustering of applications with noise (DBSCAN) 1:442, 1:1039–1040, 1:1040F, 2:390–391, 2:391
 “density-reachability” cluster model 1:1039–1040
 dental follicle cells (DFCs) 1:835
 DENV Capsid (DENV C) 3:764, 3:765F
 3-deoxy-D-arabino-heptulosonate 7-phosphate-synthase (DAH7PS) 3:656–657
 deoxynucleoside triphosphates (dNTPs) 2:300–301
 deoxyribonucleic acid (DNA) 1:1100, 1:1101F, 1:1102, 2:178, 2:324, 2:567, 2:567–568, 3:230, 3:425, 3:510, 3:535, 3:731;
 see also ribonucleic acid (RNA)
 array technology 3:820
 barcoding 3:985, 3:985F, 3:986
 assigning taxonomic names to DNA barcodes 3:989–990
 butterflies of Setiu Wetlands, Malaysia 3:992–993
 clustering DNA barcodes into OTU 3:991
 editing sanger sequences 3:986–987
 processing high-throughput sequencing reads 3:987–989, 3:988T
 sandflies of Northern Thailand 3:993
 sequence alignment 3:989
 supplementary information 3:994
 binding residues prediction 2:145–147
 bioinformatics tools to optimize codon usage in 3:329
 cell division interplay between proteins and 3:74–75
 chemical modifications 2:311
 codons 3:327
 compact algorithm 1:483
 comparison 3:983
 DNA-based approach 3:18
 fragment 3:968
 helicase and lyase activity 2:1
 methodologies to investigate DNA–TF interaction 3:75–76
 methylation 1:137, 2:128, 3:324, 3:354, 3:355, 3:356–357, 3:809
 analysis 3:348
 comparison at transcription factor binding sites 3:359
 comparison between cell types 3:358–359
 comparison of gene expression and 3:358
 databases 2:128–129
 microarray 1:127, 2:121, 3:195, 3:830, 3:832, 3:832F
 data 1:542
 technologies 3:117, 3:482
 molecules 1:916, 3:447, 3:832
 nanoballs' technique 3:434
 nucleotides 3:341
 Origami 3:535
 parameterized DNA models 2:713, 2:713F
 polymerase 3:820
 replication 3:84
 sequences 2:8, 2:34, 3:323, 3:981
 composition 3:323–325
 duplication 3:965
 zinc finger proteins for specific binding to 3:633–634
 sequencing 3:176
 structure role and dynamics 2:144–145
 TF binding to DNA partially unwrapped from histone octamer 2:311–314
 transposons 2:210–211, 2:214
 dependency detection techniques 1:456
 dependency management 2:1155
 containerization 2:1155–1156
 universal package management 2:1155
 dependent variable 1:722
 deployment model 1:236, 1:237–238
 depth first sampling (DFS) 1:90, 1:91
 depth perception 2:525–526, 2:526F
 depth-first search (DFS) 1:940–943, 1:942, 1:942F, 1:947
 descendants 1:871
 description logics (DL) 1:288, 1:294, 1:591, 1:786, 1:877
 descriptive measures 1:651–654, 1:652T
 descriptive models 1:968–972
 community structure in networks 1:972–973, 1:972F
 network models 1:968–972
 descriptive statistics 1:648
 cumulative histogram 1:683F
 data analysis and results 1:681–683
 frequency of patients grouped by nationality 1:682F
 organization and visualization of data 1:672–674
 software 1:682–683
 statistical data types 1:672
 statistical measures 1:674–676
 DESeq normalization 2:368–369
 DESeq2 3:817
 DeSigN 2:1095
 design matrix 1:724, 1:725
 desktop grids (DGs) 1:232
 Desktop IDE 1:247
 detection above background (DABG) 2:225–226
 DETECTIVE protocol 2:475T
 deterministic approaches to modeling 2:791
 deterministic simulation 2:865–866
 deuterium glucose 2:987
 DGIdb database 2:1094
 diabetes mellitus (DM) 1:115
 DIANA-TarBase 2:167
 dictionaries 1:196
 dictionary of secondary structure of proteins (DSSP) 1:483, 3:519
 dietary restriction (DR) 2:346
 diethylpyrocarbonate (DEPC) 3:830–831
 difference equations 2:877

- differential binding analysis 3:114–115
 differential equations (DE) 2:875–876,
 2:876F, 2:920–921
 difference equations 2:877
 ordinary 2:875–876
 PDE 2:876–877
 differential expression 2:360, 2:373–375
 analysis 3:308–309, 3:795–796, 3:801
 deconvolving batch effects 3:796
 FDR assessment 3:796
 multi-factor 3:796
 single factor 3:796
 gene expression 2:372F
 microarray approach to gene expression
 profiling 2:375
 microarrays 2:377–379
 RNA-seq 2:379–380
 from RNA-seq 2:382–383
 between sample normalization 2:381–382
 sequencing approach to gene expression
 profiling 2:375–377
 within-sample normalization 2:380–381
 differential gene expression (DGE) 2:331–
 332
 differential methylation 3:347
 differential network analysis 3:796–797,
 3:798
 differential ontology editor (DOE) 1:807
 differentially expressed genes (DEGs) 1:910,
 2:226, 3:355, 3:456, 3:793
 identification 3:456
 2-level single factor DEG analysis 3:796
 differentially methylated cytosines (DMCs)
 3:356
 differentially methylated positions (DMPs)
 3:347
 differentially methylated regions (DMRs)
 3:347, 3:348, 3:356
 differentially spliced exons (DSEs) 2:226
 diffractometers 2:514
 DiffSplice 2:227
 diffusion limit 2:10–11, 2:15–16
 diffusion tensor imaging technique (DTI
 technique) 2:808
 digital drivers at intersection of technology
 and medicine 3:66–67, 3:67F
 community role 3:68–69
 data role 3:66–67
 devices role 3:67
 platforms role 3:68
 role of advanced analytics and machine
 intelligence 3:67–68
 digital object identifiers (DOI) 2:81, 2:631
 digital revolution 1:272, 2:1083
 digital RNA fragments 3:452
 digitalised 2D-gels 2:1064
 digraph. *see* directed graph
 dihedral angles 1:52
 dihydroxyacetone phosphate (DHAP) 3:638
 Dijkstra's algorithm 1:940, 1:943
 dimension reduction techniques 1:466–467,
 2:429–430
 dimensionality reduction 1:480, 1:486,
 1:487T, 3:366–367;
 see also numerosity reduction
 attribute selection 1:482
 Fisher linear discriminant 1:488–490
 ICA 1:480–481
 ISOMAP method 1:490–491
 linear transformation 1:480
 PCA 1:486–488
 PCA 1:480
 subset selection 1:491–493
 WT 1:481–482
 dimensions. *see* attributes
 dimethyl sulfate (DMS) 2:580, 3:575
 dimethyl sulfate sequencing (DMS-seq)
 2:580
 dimethylarginine dimethylaminohydrolase
 (DDAH) 3:733
 DIP followed by high-throughput
 sequencing (DIP-seq) 2:148, 2:149
 direct analysis in real time–mass
 spectrometry (DART–MS) 3:472
 direct branch-and-bound approaches 2:644
 direct determination of long-range
 interactions 3:579
 direct functional analysis 2:405–406
 direct index 1:267
 direct infusion mass spectrometry (DIMS)
 3:472
 direct interaction (DI) 2:810
 direct link 1:243
 direct method 1:388–389
 direct PPI 3:1
 direct readout. *see* sequence readout
 direct semantics 1:797–798
 direct to consumer companies (DTC
 companies) 1:898
 directed acyclic graph (DAG) 1:701–702,
 1:814, 1:823, 1:825F, 1:839, 1:868,
 1:878, 1:908, 1:1085–1086, 1:1097,
 2:158, 2:1152–1153, 3:137, 3:222
 directed graph 1:922, 1:923F, 1:928, 1:934F,
 1:935, 1:1085–1086
 directed network 1:85, 1:85F
 directed proteomics 2:1070
 Dirichlet mixture distribution 3:337
 discarding data 1:464
 Discotope 2.0 2:955
 Discovery Studio Visualizer 3:276, 3:279
 discrete cosine transform (DCT) 2:999
 discrete distribution of read counts 2:382–
 383
 discrete dynamical networks (DDNs) 1:1055
 discrete dynamics lab (DDLab) 1:1055
 discrete optimized protein energy statistical
 potential (DOPE) 3:588
 discrete wavelet transform (DWT) 1:481,
 2:999
 discretization techniques 1:478
 discriminative features in data driven models
 3:681–682
 discriminator. *see* thresholding value
 Discs Large Homolog (DLG) 3:224
 disease 1:1025
 biomarker discovery 3:480–481
 cancer based study 3:483–486
 computational protocols and methods
 3:478–480
 gene expression and metabolic network
 3:481–483
 metabolic networks 3:476–478, 3:476F
 disease-causing mutations 3:169
 disease-related candidate gene 3:444
 genes 1:988
 prediction methodologies 2:809–810
 metabolomics in 3:471–472
 PPIN rewiring during disease, mutation
 and evolution 3:285–286
 transmission 3:976–977
 DISeAse MModule Detection (DIAMOnD)
 2:811
 disease ontology (DO) 1:838, 1:839, 1:842–
 844, 1:844, 1:908, 1:1025, 1:1104–
 1105
 annotations in 1:868
 mapping with 1:841
 quantitative aspects 1:840–841
 relations in 1:839–840
 structure 1:839
 technological details 1:841–842
 web browser 1:842
 disease-modifying anti-rheumatic drugs
 (DMARD) 3:729
 DisGeNet 2:1051
 disjunctive common ancestors (DCA) 1:873
 disjunctive logic programming (DLP) 1:786
 Disorder B-factor Agreement (DBA) 3:594–
 595
 disordered proteins 2:31, 2:509
 dispersion measurement 1:676–677
 dissolved oxygen (DO) 3:1
 dissortative mixing 1:965
 distance-scaled finite-ideal gas reference state
 (DFIRE) 3:667
 distance(s)
 distance-based consensus 2:741
 distance-based methods 1:475, 3:367,
 3:368–369, 3:373
 distance-based outliers 1:457
 distance-based tree 3:336
 in feature space 1:511–512
 functions 1:413–414
 distant supervision (DS) 1:604
 DistiLD 2:1051
 distortions in regular SSEs 3:521–522
 distributed accelerator (DA) 1:223
 distributed access and integration (DAI)
 1:1105–1106
 distributed annotation system (DAS) 1:208
 distributed architectures 1:162–163
 bioinformatics 1:171–174
 cloud 1:163–164
 grid 1:163
 programming languages for 1:167–169
 distributed databases 2:99, 2:99F
 distributed file system 2:1049
 distributed infrastructure with remote agent
 control (DIRAC) 1:233
 distributed memory languages 1:216–217;
 see also shared memory languages
 HPF 1:217–218
 MPI 1:217
 distributed programming in Java 1:218–219
 distributed query processing (DQP) 1:1105–
 1106
 distributed share dmemory 1:215

- distributed systems 1:221
- distribution-based clustering method
1:1038–1039
- disulphide bridges 3:266–267
- diversification 3:946
- diversity
analysis 3:206
index 3:21, 3:21F
microarrays 3:203
- divide and conquer technique 1:8–9, 1:386–387, 2:277
- divisive approaches 1:442
- DMS mutational profiling with sequencing (DMS-MaPseq) 2:580
- DNA Barcoding Workshops (DBW) 3:986
- DNA Data Bank of Japan (DDBJ) 1:112, 1:132, 2:102, 3:30
- DNA immunoprecipitation followed by microarray (DIP-chip) 2:148, 2:149
- DNA methyltransferase enzymes (DNMTs) 3:341
- DNA Y-family polymerase IV (DPO4) 2:143
- DNA-binding proteins (DBPs) 1:916, 2:134, 2:835, 3:114, 3:678, 3:933
dynamics and conformational changes 2:143–144
with mixed signals 3:114
prediction 2:145–147
- DNA-binding residues prediction 3:683–684
- DNA-COMPACT method 1:484
- DNA-dependent kinase (DNA-PK_{cs}) 2:942
- DNA Nexus 1:253, 1:259, 3:139
- DNApipeTE approach 2:215
- DNase I hypersensitivity 2:204
- DNase-Seq 2:149
- DNAzip algorithm 1:483–484
- DO REST API 1:842
- DOCK. *see* UCSF DOCK v6.7
- Docker 3:138–139
- docking 1:77, 1:77–78, 1:78–79, 1:79, 2:842–844, 2:843T, 2:1144
computational approaches 1:78
methodology validation 2:595, 2:595T
parameter setup 2:693
and scoring 3:742
- docking-based virtual screening (DBVS) 2:688, 2:688F
case study 2:689–690
control docking or validation docking method 2:693–694
docking parameter setup 2:693
grid parameter setup and affinity maps calculation 2:692–693
preparation of ligand from database 2:690
preparation of receptor and control ligand 2:690–691
VS 2:694–696, 2:696–697
databases for VS 2:689
structure of directory in 2:695F
- document classification (DC) 1:578, 1:579T, 1:579
- document type definitions (DTD) 1:443, 1:826
- document/documentation 1:200
store 2:1065T
- summarization 1:498, 2:1085
in vector space 2:1102–1103
- doGoD function 1:911
- DOLite 1:843–844
- DOLPHIN algorithm 1:457, 2:399
- domain description techniques. *see* semi-supervised methods
- domain duplication 3:965–966
- domain fusion-based approaches. *see* gene fusion-based methods
- DOmain MAKer protocol (DOMAK protocol) 2:475T
- domain object model (DOM) 1:1063
- domain-domain interaction networks (DDI networks) 1:919, 3:286, 3:287
- domain-specific languages (DSL) 2:1153
- domain(s) 2:499, 3:504, 3:513
domain-based methods 3:287–288
domain-specificity of pathways 1:1076
models 1:589–590
ontologies 1:809
- DomainRBF 1:910
- domains of unknown function (DUFs) 3:958
- domestication and comparative animals 2:264
domestic organisms 2:264
GWAS and selective sweep 2:264–265
- DOQCS 2:799
- dormancy 3:926
- DORN1 3:261
- dorzolamide 2:460, 3:744T–748
- dot products 3:880
- dot-plot 2:273, 2:273F, 2:274F
- double data type 1:207
- double helical structure of DNA 3:504–505
- double strand break (DSB) 3:635
- double stranded DNA (dsDNA) 2:288, 2:289F, 3:435
- double-negative (DN) 2:978
- double-walled lipids 3:941
- DOUBLESCAN program 2:203
- doublesex (dsx) 3:968
- doubly stochastic model 2:707
- downstream analysis 3:115
- downward closure 1:360
- DPClus method 1:95
- DPCLusO algorithm 3:495
- DQPred-DBR 3:681
- Draft Secondary Structure 3:542
- drawings 1:1085
- DREAM project 2:161
- dreiding model 2:524
- DREME 3:449
- DRNAPred 3:679
- DroID 2:852
- Dropout 1:641
- Drosophila* 3:965, 3:968
Drosophila melanogaster 2:197, 3:821, 3:968
Drosophila melanogaster's Down syndrome cell adhesion molecule (Dscam) 2:222–223
- drug combination database (DCDB) 2:1093–1094
- drug discovery 1:988, 2:1061, 3:273, 3:742, 3:743F
- approaches for drug discovery/designing 3:742, 3:744T–748, 3:748F
de novo ligand designing 3:749
genetic algorithm 3:748–749
homology modelling 3:743
HTS 3:742–743
MCM simulations 3:748
molecular dynamic simulations 3:743
molecular fingerprint and similarity searches 3:749
Monte Carlo simulation 3:743–748
rational drug design 3:748
virtual screening 3:742
- clinical research 2:1052
biomarker (tracking) efficacy 2:1052
pharmacogenomics for optimal treatment 2:1052
- and design 1:231
- discovery of tyrosine kinase inhibitors 3:749
- drug molecule development 2:1052
- post-approval surveillance 2:1052
- text mining and 2:1090
- tools and web servers 2:1054T
- drug gene interaction database (DGIdb) 2:1059–1060
- drug metabolism and pharmacokinetics (DMPK) 3:750
- drug metabolism enzymes and transporters (DMET) 1:333
DMET-Analyzer 1:333
DMET-Miner 1:333
- drug target
databases 2:1093–1094, 2:1094
identification 2:1094
identification and validation in human
approaches for target validation *in silico*, *in vitro* and *in vivo* 2:1094–1096
computational approaches in drug target identification 2:1094
target identification *in silico* 2:1093–1094
and structure determination 2:586
binding site identification 2:587–588
homology modeling 2:586–587
molecular docking based VS 2:588
X-ray crystallography and NMR 2:586
structures evaluation 3:273–274
- Drug-drug interaction (DDI) 1:276, 2:1053
- drug-likeness and druggability concept 2:614
- Drug-Path 2:799
- Drug-specific Signaling Pathway Network (DSPNet) 2:861
- drug(s) 3:741, 3:741–742
candidates and drugs 2:612–613
classes 2:831
databases 2:1093–1094
decipher STNs for drug effect in complex disease 2:861
design 2:614
proteins and 2:455
development 3:273
efficacy 3:379
genetic basis of variability in drug response 3:379–380
indifference 3:926
information databases 2:633

- drug(s) (*continued*)
 interactions 2:1053
 metabolism 2:796–797
 molecule 2:1052
 molecular variation 2:1052
 target identification/rational drug discovery 2:1052
 pathway databases 2:799
 QSAR in drug designing and discovery 2:669
 related pathways 2:796–797
 repositioning 3:5–6
 repurposing 2:1052–1053, 3:782T
 challenges and solutions 3:786–787
 computational tools for 3:781–783
 computational tools for drug repurposing 3:781–783
 databases and computational resources for 3:780–781
 databases and computational resources for drug repurposing 3:780–781
 polypharmacology, benefits and drawbacks 3:780
 virtual screenings 3:783–786
 resistance
 of HIV-protease inhibitors 3:707–709
 molecular mechanism 3:711, 3:715, 3:717–718
 pathogens development 3:241
 safety monitoring 2:1052
 Drug2Gene 2:1054
 DrugBank database 2:633, 2:799, 2:1090, 2:1094, 3:384
 druggability concept 2:614
 DrugQuest 2:1090, 2:1091F
 DSGseq 2:227
 dual RNA-seq technique 3:410
 Dubin-Johnson syndrome 1:912
 Dunbar number 1:86
 duplex groups (DGs) 3:580
 duplex structures 3:580
 duplicate data detection 1:465–466
 ETL method 1:466
 knowledge-based methods 1:465–466
 dyadic scaling scheme 1:481
 dyes swap experiments 2:377
 dynamic Bayesian networks (DBN) 2:159
 dynamic branching 2:1153
 dynamic causal modelling (DCM) 2:808
 dynamic model software 2:875, 3:437
 dynamic modeling 2:864–865
 dynamic network models 2:159
 applications 2:160
 BN models 2:159–160
 constructing GRN using ODE 2:160
 DBN 2:159
 dynamic programming (DP) 1:10–11, 1:23, 1:280, 2:270, 2:577, 2:1144–1145, 3:231, 3:314
 algorithms 2:579
 for global alignment 3:314
 for local alignment 3:314
 for MFE 2:579
 for partition function 2:579
 McCaskill algorithm 2:579
 dynamics, linking to 2:1138
 dynamics proteomics (DynamProt) 2:1048
 dysbiosis 3:17
 dystrophin gene 3:965
- ## E
- Eades & Peter algorithms 1:1089
 Ear Nose and Throat (ENT) 1:157
 Early pregnancy factor (EPF) 3:871
 Ebola virus membrane-associated matrix protein (vP40) 2:821
 eccentricity centrality 1:952
 ECGSYN tool 2:1049–1050
 Eclat algorithm 1:363, 1:365
 eclipse 1:247
 Ecocyc database 1:152
 ecological communities 2:1131
 ecological networks 1:958, 1:1018, 2:1131, 2:1132F, 2:1134
 comparison of structural patterns between different types of 2:1135
 construction of highly normalized databases 2:1137–1138
 databases and network analysis tools 2:1134
 environmental factors effects on 2:1133
 evaluating significance of structural pattern 2:1134–1135
 finding keystone species 2:1133–1134
 impact of global warming on 2:1135
 linking to dynamics 2:1138
 multiplex network analysis 2:1137
 searching for non-random structural patterns 2:1131–1133
 structure–stability relationship 2:1133
 weighted network analysis 2:1136–1137
 economic trait loci (ETL) 3:428
 ecosystem monitoring through predictive modeling
 ML methods
 application in water quality prediction 3:3–4
 in water quality modeling 3:2
 visualization and Google Earth 3:6–7
 EdaFold 1:778
 EdaRose 1:779
 algorithm 1:779–781
 protocol design particularities 1:779
 steps of 1:780F
 edge based segmentation method 2:997
 edge correctness (EC) 1:998, 1:1005
 edge sampling (ES) 1:90, 1:93
 edge sampling with contraction (ESC) 1:90, 1:91
 edge-based method 2:1034
 edge-based segmentation 2:1030
 edge-matrix layout 1:1020–1021
 edge-weighted graph 1:923–924, 1:925
 edit distance 1:439, 3:295
 Edman degradation procedure 3:311, 3:311F
 effective Nc (ENC) 3:329
 Efficient Referential Genome Compressor method (ERGC method) 1:484
 EGEE/EGI grid platform 1:231–232
 ego-network 1:85
 eHive, Perl workflow management system 3:137–138, 3:138F, 3:147–149, 3:149F
 EIC. *see* extracted ion chromatograms (XIC)
 EIGENSOFT software 3:366, 3:369
 eigenvector centrality 1:84, 1:937, 1:951–952
 EKEY system 2:1023
 Elastic Beanstalk 1:240
 Elastic Compute Cloud (EC2) 1:240, 1:257
 elastic net regression 1:729
 elastic network models (ENM) 3:610–611
 electrical activation in heart tissue, modelling spatial propagation of 2:902–903
 electrocardiogram (ECG) 1:1105–1106
 electroencephalography (EEG) 2:808
 electron capture dissociation (ECD) 3:857–858
 electron crystallography 3:521
 electron impact Mass spectrum (EI-MS) 2:403
 electron ionisation (EI) 2:403, 2:411
 electron microscopy (EM) 2:460, 2:512, 2:513F, 2:515–516, 2:516–517, 2:530, 2:915–916, 3:613–614, 3:614
 Electron Microscopy Data Bank (EMDB) 2:465
 electron spin resonance (ESR) 2:403
 electron transfer dissociation (ETD) 2:1066, 3:857–858
 electronic health record (EHR) 1:265–266, 2:1049, 2:1053
 integration
 data visualization and linked data 2:1063–1064, 2:1069
 database and DBMS 2:1066–1067
 EMR 2:1063
 EMR graph model 2:1067–1068
 limitations of current study and proposed future work 2:1074
 NoSQL databases 2:1064
 relational *vs.* graph databases 2:1073
 relational *vs.* NoSQL databases 2:1069–1071
 systems 1:154
 electronic kinetics 2:864
 Electronic Medical Record (EMR) 1:265, 2:1052, 2:1053, 2:1063
 availability 2:1073
 consistency 2:1073
 data models 2:1073
 graph model
 document database development 2:1068
 EMR graph database development 2:1068–1069, 2:1068F, 2:1069F
 Neo4j EMR data visualization 2:1069, 2:1070F, 2:1071F
 NoSQL databases benefits for 2:1064–1065
 performance and storage scalability 2:1073
 electronic Medical Record and Genomics Network (eMERGE Network) 2:175
 electronic Pharmacogenomics Assistant (ePGA) 3:384
 electropherograms 3:986
 electrophoretic mobility 2:1063

- electrophoretic mobility shift assay (EMSA) 2:145, 2:149, 2:685
- electrospray (ES) 2:411
- electrospray ionisation (ESI) 1:126, 2:84, 2:894–895, 2:1064, 3:466, 3:914–918
- electrospray ionization mass spectrometry (ESI-MS) 3:466
- electrostatic potential (EP) 2:522, 2:532
- electrostatic(s) 2:1144
- free energy 2:449
- interactions 3:504
- elementary measures, composite
- performance measures derived from 1:549–550
- elementary mode (EMs) 2:441, 2:441F
- applications 2:442
- control effective fluxes and structural fluxes 2:442
- large-scale enumeration 2:441–442
- elementary performance measures 1:548–549
- Elispot assays 3:40
- ELIXIR Tools 2:86
- Ellipro 2:965
- ellipticity 3:520–521
- elongation factors adaptations 3:947–948
- elongation factors Tu family (EF-Tu) 3:947–948
- elucidation
- of potential inhibitors on DENV protein targets
- DENV non-structural proteins 3:764–765
- DENV structural proteins 3:761–763
- of protein function 2:34
- structural dynamics and mechanisms of MTB targets 3:655–657
- embedded methods 1:301, 1:331, 2:1004
- embedded-atom method (EAM) 2:561
- EMBOSS seqret 3:118
- embryoid bodies (EB) 2:327
- embryonic stem cell (ESCs) 2:327, 3:342, 3:826–827
- EMDataBank 3:522
- emerging paradigms in ML 1:304–305
- empirical contact models 2:716
- use of contact potential matrix for substitution 2:717F
- empirical distribution function (EDF) 1:767
- empirical models 2:714–715
- empirical-based functions 3:274
- empirical-energy functions. *see* statistics-based functions
- emulsion PCR (emPCR) 3:158
- Encapsulated Postscript (eps) 1:1051
- Encyclopedia of DNA Elements (ENCODE) 2:140, 3:341, 3:357–358, 3:449–450
- consortium 2:178
- database 2:252–253
- pilot project 2:275
- project 2:204, 2:344–345, 3:230, 3:381, 3:395
- Encyclopedia of Life (EoL) 2:103
- database 2:1115, 2:1116–1117, 2:1118F, 2:1119F
- taxonomy data content in 2:1121F
- textual and image content in 2:1120F
- Endeavor software 1:910, 3:444, 3:444–445
- endocrine 1:917
- endogenous silencing RNAs (endo-siRNA) 2:326
- endoliths. *see* cryptoendoliths
- endoplasmic reticulum (ER) 2:53, 2:796, 3:39
- energetically optimized structure-based pharmacophores (e-pharmacophores) 2:681
- energy
- based approach 1:47, 2:588
- functions 3:588
- models 2:576–577
- Energy Calculation and Dynamics (ENCAD) 1:63–64
- energy minimisation (EM) 3:659
- energy of highest occupied molecular orbital (EHOMO) 2:670
- energy of lowest unoccupied molecular orbital (ELUMO) 2:670
- Engyodontium albus* 3:942
- enhancer lncRNAs (elncRNAs) 2:168
- enhancer RNAs (eRNA) 2:325, 3:49, 3:231
- EnhancerPred tool 3:323
- enrichment analysis 3:816
- enrichment score (ES) 2:802
- EnrichNet databases 1:1075
- Ensembl Genome 2:251–252, 2:252
- Ensembl genome browser 2:253
- Ensembl Variation 3:299
- ensemble approach 1:460, 1:519, 1:570, 2:322
- ensemble learner. *see* multiple learner combiner
- ensemble learning methods 1:524, 1:525, 1:537, 1:539–540
- ensemble learning system, building 1:521–523, 1:522F
- methods for combining multiple learners 1:522–523
- combining class label 1:522–523
- combining functions for continuous decision outputs 1:523
- most popular ensemble learning algorithm 1:524
- enterovirus 71 (EV71) 3:786
- entity 1:603
- class 1:148
- similarity 1:874
- table 2:1068
- entity relationship variant (ER variant) 2:887
- Entity-Relationship Diagram 1:1065
- ENTPRISE
- tool 3:393
- webserver 3:4
- Entrez Genome database 3:243
- Entrez system 1:1092, 1:1093
- entropy 1:468, 2:1002
- entropy based discretization method 1:468
- entropy-based thresholding 2:1029
- envelope protein (E protein) 3:761, 3:761–763
- environment (al) 1:310–311, 1:311T
- data 2:1134
- factors effects on ecological networks 2:1133
- Environment-wide association studies (EWAS) 2:175
- Environmental Health Institute (EHI) 3:976
- environmental microbiome, mapping 3:17–18
- application 3:18–19
- CRAD 3:21, 3:22F
- diversity index 3:21, 3:21F
- multivariate analysis based on Unifrac distance 3:23–25, 3:24F
- periodic analysis 3:25
- taxonomic assignment 3:21–23, 3:23F
- enzymatic structure probing of RNA 3:575
- Enzyme Commission (EC) 2:478–479
- enzyme(s) 2:1100, 3:741
- enzyme-linked receptors 1:1057–1058
- function 2:650
- proteins 3:1
- responsible for writing or erasing PTM 2:20
- Eoulsan 1:259
- ephrin type-A receptor 3 (EphA3) 3:749, 3:750F, 3:751F
- Epidemic Resurgence of Dengue Fever in Singapore (2013–2014) 3:977–978
- epidemiology
- research studies 3:975–976
- Epidemic Resurgence of Dengue Fever in Singapore (2013–2014) 3:977–978
- intra-epidemic evolutionary dynamics 3:976–977
- region-wide synchrony and traveling waves 3:975–976
- three-month real-time dengue forecast models 3:976
- epidermal growth factor receptor (EGFR) 3:755
- protein controlling lung cancer 3:755
- EpiDOCK 2:955
- EpiFactors 2:129
- Epigenetic Module based on Differential Networks (EMDN) 2:128–129
- epigenetics 2:126
- 5caC 3:342
- allelic expression 3:348
- computational tools in 2:128–129
- chromatin related databases 2:129
- CpG island related databases 2:129
- DNA methylation databases 2:128–129
- histone related databases 2:129
- correlation
- with ChIP-Seq 3:347
- with RNA-Seq 3:347–348
- cytosine modifications 3:341, 3:342F, 3:348
- determining cytosine modification proportions 3:346
- differential methylation 3:347
- emerging technologies 3:348
- experimental methods to assay DNA modifications in genome 3:342–344, 3:343T
- factors affecting nucleosome positioning 2:309–310
- 5fC 3:342

- epigenetics (*continued*)
 genomic distribution of cytosine
 modifications 3:347
 5hmC 3:341–342
 long-read sequencing 3:348–349
 5mC 3:341
 modification 2:127–128, 2:311, 3:354–355
 quality control and mapping to genome 3:344
 single cell sequencing 3:349
 system 1:1016
- epigenomics 2:126–128
 computational tools in epigenetics 2:128–129
 DNA methylation, genome imprinting and paramutation 2:128
 tools and databases used in epigenetics 2:127T
- EpilepsyGene 2:1050–1051
- EpiMatrix 2:954
- epistasis 2:171–172, 2:172, ,, 2:173, 2:175
- epistatic genetic interaction (GI) 2:814
- epithelium (Epi) 2:392
- epitope 2:931
 databases 2:965
 identification 3:40–42
 databases hosting immune epitopes 3:40–42
 prediction systems 3:42–44
 predictions 2:952–953, 2:953–954, 2:954–955, 2:955, 2:965–966, 2:966–967, 2:967
- epitranscriptomics 3:807
- e-intensive loss function 1:509
- Epstein-Barr virus (EBV) 3:41–42, 3:425
- Epstein-Barr virus T cell antigen database (EBVdb) 3:40
- equality operators 1:181
 comparing numbers 1:181
 comparing strings 1:181
 logical operators 1:181
 other operators 1:181
- equivalence class 2:159
- equivalent of variance 1:489
- ERANGE tool 2:368
- Erdos-Renyi Network (ERN) 1:90
- ERGO 3:153
- erlotinib 2:585, 3:744T–748
- Erpin program 3:324
- error correcting output code (ECOC) 1:522
- error correction (EC) 2:1143, 3:160
- error(s)
 functions 1:622
 in PPI databases 1:991–992, 1:992F
 improving data reliability 1:992
 sources and rates of errors 1:991–992
 rates assessment 2:88–89
 sources 2:736–737
 biological sources 2:737
 modeling errors 2:737
- escape sequences 1:179, 1:179T
- Escherichia coli* 2:142, 2:161, 2:251–252, 2:866, 3:18, 3:184, 3:293, 3:323, 3:324, 3:427–428, 3:463, 3:821
- ESEfinder 2:228
- essential dynamicscoarse-graining (ED-CG) 2:560
- essential proteins 2:830
- Estimated Locations of Pattern Hits (ELPH) 3:323
- estimated melting temperature (eTm) 3:580
- estimation of distribution algorithms (EDA) 1:775, 1:776–777
- Estimation of Model Accuracy (EMA) 3:587
- estimation of recent shared ancestry (ERSA) 2:247
- estimator, evaluating 1:739–740
 bias 1:740
 consistency 1:741
 MSE 1:740–741
- estrogen receptor (ER) 3:384–385
- ESX system. *see* Type VII secretory system
- ethambutol (EMB) 3:652–653
- Eucalyptus 1:245
- EUCANEXT database 2:347–350
- Euclidean algorithm 1:5
- Euclidean distance 1:350, 1:413, 1:438, 2:158
- eukaryotes 3:13–14, 3:18, 3:325, 3:568, 3:837
 assembly 3:305
 cells 2:54–55, 3:1, 3:825
 to eukaryote 3:953
 eukaryotes organisms 3:938
 genes 2:796
 genomes 3:305
 to prokaryote 3:953–954
- eukaryotic linear motifs (ELMs) 2:18, 3:415–417
- Eukaryotic Pathogen Database Resources (EuPathDB Resources) 2:1086, 2:1087F, 2:1087T, 3:108–109
- eukaryotic promoter database (EPD) 2:102
- eukaryotic splicing 2:221
- eukaryotic telomeres 3:506–508
- Euler number 2:1000, 2:1002F
- Euler-Maruyama method 1:750
- Eulerian walk 1:929
- Europe PubMed Central (Europe PMC) 3:996
- European Bioinformatics Institute (EBI) 1:112, 1:829, 1:861, 1:1065, 1:1093, 2:80, 2:468, 2:631–632, 2:937, 3:30, 3:996
- European Genome-phenome Archive (EGA) 2:1080
- European Molecular Biology Laboratory (EMBL) 1:112, 1:133, 1:1092–1093, 2:103, 2:631–632, 3:996
- European Molecular Biology Library–European Bioinformatics Institute (EMBL-EBI) 1:825
- European Mouse Mutant Archive 2:103
- European Nucleotide Archive (ENA) 1:132, 2:102, 3:30
- eutrophication 3:1
- evading host system 3:105–106
- event and relation extraction 1:568–569, 1:568F
 BioNLP-ST 1:568–569
- event-wise testing (EWT) 3:965T
- evidence code (EC) 1:867, 1:889, 2:3
- evidence-based algorithms 2:182–183
- evidence-based methods 3:233
- Evodesign 3:647
- evolutionary algorithms (EA) 1:774–775, 1:775–776
 elements of 1:775–776
- evolutionary ancestry 3:980
- evolutionary classification of protein domains (ECOD) 2:467, 2:472T
- evolutionary conservation 3:682
- evolutionary distances for comparative analysis 2:258–259
- evolutionary information (EI) 3:670
- evolutionary models 2:712
 CAT models 2:715
 Codon models 2:715
 covarion models 2:715–716, 2:716F
 empirical contact models 2:716
 empirical models 2:714–715
 Lie-Markov models 2:714
 Markov models 2:712–713
 model selection 2:714
 parameterized DNA models 2:713, 2:713F
 parameters 2:713–714
 physicochemical models of substitution 2:716
- evolutionary parameters and model selection, estimation of 2:753–754
- evolutionary trees 2:123
 and networks 1:1106
- EX-SKIP application 2:228
- exact confidence interval for proportion 1:557–558
- exact methods 1:562
- excel-compatible format 1:993
- exhaustive bootstrap 1:769
- exhaustive search 1:5–6
- EXOFISH program 2:202
- Exome Aggregation Consortium (ExAC) 2:186, 2:189, 3:169, 3:170
- exome analysis pipelines 3:116
- exome sequencing analysis workflow 3:165
 exome capture platforms for human exome sequencing 3:166T
- exome sequencing data analysis 3:164–165
 advantages of WES 3:165
 BQSR 3:168
 clinical diagnosis 3:171
 complete automated pipelines 3:169
 discovery of rare variants in Mendelian diseases and complex disorders 3:170–171
- exome sequencing analysis workflow 3:165
- limitations 3:171
- local realignment 3:168
- post alignment processing 3:167
- quality control/pre-processing 3:165–166
- reference-based alignment 3:166–167
- removal of duplicates 3:167–168
- repositories for variant storage and annotation 3:169–170
- variant analysis 3:168
- Exome Sequencing Project (ESP) 2:186, 2:189
- exome/whole genome sequencing 3:390

ExomeCNV 3:116
 Exomiser 1:706
 exon splicing enhancers (ESEs) 3:323
 exon-centered splicing score (ECSS) 2:224
 exon(s) 2:221, 3:306, 3:967
 deletion 3:966
 duplication 3:966
 insertion 3:966
 shuffling 3:966–967, 3:967F
 exonization and pseudoexonization 3:967
 mosaic or chimeric proteins 3:966–967
 EXONERATE genomic sequence 2:199–200, 2:202
 exonization 3:967
 exonucleases 3:843
 exosomes 2:326, 3:483
 ExPASy server 1:38
 Expectation-Maximization (EM) 1:17, 1:19, 1:352–354, 1:441, 3:338, 3:448–449
 algorithm 1:464–465, 1:1038–1039, 1:1039F, 3:443
 clustering algorithm 1:593
 expectation-value (E-value) 1:44
 expected time to fixation 2:4–7
 Experimental Factor Ontology (EFO) 1:841
 experimentally verified annotation 1:867
 experimentList 1:1064
 eXpert Model method (XM method) 1:483
 explanatory subspace 1:461
 explanatory variables 1:491
 explicit DSLs 2:1153
 explicit frameworks 2:1153
 explicitly model biases 2:320
 exploratory analysis 1:337
 exploratory data analysis techniques 3:365
 exploring sequence patterns and profiles 3:35
 exponential algorithms 1:1
 exponential random graph model 1:972
 expressed sequence tags (ESTs) 2:156–157, 2:182–183, 2:196, 2:200, 2:223–225, 2:347, 2:1051, 3:29, 3:323–324, 3:428, 3:819–820, 3:820, 3:843
 expression analysis 2:360
 expression clustering 2:389F
 case study 2:392–394, 2:393F
 clustering methods 2:389–390
 practical considerations 2:391–392
 expression data for reconstructing GRN 2:156–157, 2:157F
 expression profile method 2:840T
 expression profile reliability (EPR) 2:806
 Expression Profile Viewer (ExProView) 2:121–122
 expression quantitative trait loci (eQTLs) 2:334
 extended ancestral-state reconstruction algorithm 3:189
 extended association rule mining 1:370–371
 extended finite-context models (XFCMs) 1:483
 extended haplotype homozygosity (EHH) 3:371
 extended information content (eIC) 1:880–881

extended majority-rule consensus 2:741
 extended pathway databases 1:1075
 extended phenotype 3:67
 extended secondary structures 2:575–576
 extensible Graph Markup and Modeling Language (XGMMML) 1:1027
 eXtensible Markup Language (XML)
 documents 1:443
 language 1:1065
 XML-based languages 1:1027
 eXtensible Markup Language (XML) 1:142, 1:859
 extensively drug resistant (XDR) 3:926
 extensively drug-resistant tuberculosis (XDR-TB) 3:652–653
 external clustering quality measures 1:354
 external indices 1:450–451
 external quality assessment (EQA) 3:9
 of dengue and chikungunya diagnostics in Asia Pacific Region (2015) 3:13–14
 of dengue diagnostics in WHO Western Pacific Region (2013) 3:11–13
 external reference files (ERFs) 2:569, 2:570
 External RNA control consortium (ERCC) 3:792–793
 external validation 1:1044–1045, 1:1045T, 2:668
 extracellular ATP (eATP) 3:261
 extracellular matrix (ECM) 3:912–913
 extracellular RNA (exRNA) 3:33–34
 extracellular signal-regulated kinase 1 and 2 (ERK1/2) 2:857
extract-as program 2:227
 extracted ion chromatogram (XIC) 2:89–90, 3:871
 extracting PHIs from literature and public databases 3:935–936
 extraction, transformation and load (ETL) 1:331, 1:466, 1:472, 1:1095
 extraction tasks 2:1104
 Extreme Machine Learning (ELM) 1:305
 Extreme Pathways (EPs) 2:441
 extremophiles 3:944–946

F

F-measure 1:981
 fabric controller 1:242
 facilitated diffusion mechanism 2:144, 2:144F
 facioscapulohumeral muscular dystrophy (FSHD) 3:236
 Factor Analysis (FA) 1:745
 Factor analysis Partial Least Squares (FA-PLS) 2:664
 factor syntax 1:202
 factual chemical space (FCS) 2:517, 2:604
 facultative exons 3:968
 Falco (cloud-based framework) 1:254
 false discovery rate (FDR) 1:549, 1:695, 1:735, 1:991–992, 2:22, 2:88, 2:89, 2:372, 2:431, 2:1068, 3:220, 3:221–222, 3:456, 3:495, 3:796, 3:863–864
 false hit rate (FHR) 3:419
 false negative (FN) 2:160–161, 2:682, 3:332–333, 3:682–683, 3:997–998
 false positive (FP) 2:160–161, 2:237, 2:682, 3:332–333, 3:682–683, 3:997–998
 False Positive Pruning (FPP) 2:1033
 false positive rate (FPR) 1:274, 1:555, 2:160–161
 family wise error rate (FWER) 1:695
 Fantom5 3:381
 Fast network component analysis (FastNCA) 1:1055
 FASTA format 1:131–132, 3:294, 3:861
 FASTAptamer 3:570, 3:570F
 FastHiC method 2:321
 FASTQ format 1:1104–1105, 1:1105F, 2:352, 2:354T, 3:144, 3:294–295, 3:987–988
 FastQC software tool 3:113, 3:118–119, 3:144, 3:166, 3:204–205, 3:296, 3:307, 3:455
 FastSTRUCTURE software 3:367
 FASTX-Toolkit 3:118, 3:455
 fat Specific Protein27 (FSP27) 2:1100
 FATHMM tool 3:393
 fatty acid synthase 3:967
 feature extraction 1:486, 2:998–1004, 2:1003F
 feature extraction/selection 3:129
 feature selection 1:486, 2:1004
 feature space 1:495
 feature-based model 1:877
 fecal microbiota transplantation (FMT) 3:17, 3:211
 federated databases approach 1:1092, 1:1093, 2:99, 2:100F
 feed forward loops (FFLs) 2:814, 2:857, 2:857–858, 2:859F
 feed-forward multi-layer Neural Networks 1:621, 1:621F
 feedback loops (FBLs) 2:857, 2:858, 2:859F
 Feedforward neural network (FNN) 1:390–392, 1:612, 1:619, 1:619F, 1:621, 1:637
 feedforward neural networks (FFNN) 1:276
 fermentation origins 3:946–947
 Fibonacci sequence example 1:184–185
 Fibonacci series 1:10
 FIBROCHONDROGENESIS 1 1:702
 fibronectin type III domain of protein tenascin (TNfn3) 3:636
 fibrous proteins 2:28, 2:446, 3:1, 3:511
 Field Programmable Gate Arrays (FPGAs) 2:1142
 Figure of Merit (FOM) 1:451, 1:452
 file
 formats 2:352, 2:356, 2:462
 transformation process 2:1151
 filter(ing) 2:356
 based approaches 1:301
 based methods 3:483–484
 constraints 1:802
 contamination 3:296–297
 method 1:330, 2:1004
 process 3:723

- filter(ing) (*continued*)
- Findable, Accessible, Interoperable, and Reusable guiding principles (FAIR guiding principles) 2:460
- Finding isoforms using robust multichip analysis (FIRMA) 2:224
- fine-mapping 3:381
- FineSplice 2:228
- fingerprint(ing) 2:38–39, 2:426–427, 2:654, 3:32
 - and pharmacophores 2:619–623, 2:624–625
 - case studies 2:623–624
 - circular fingerprints start at central atom 2:621F
 - fundamentals 2:619
 - ligand-based pharmacophore model generation 2:621F
 - substructure keys-based fingerprints translate presence or absence 2:620F
 - topological fingerprints exemplified on primary amino groups 2:620F
 - visualized grid for positively ionizable ligand functionalities 2:623F
 - workflow of structure-based pharmacophore model generation 2:622F
 - techniques 3:203
- fingerprints for ligands and proteins (FLAP) 2:652–653
- finite element method 2:561
- FIPA Agent Communication Language (FIPA-ACL) 1:313
- FireBrowse 2:1080
- FireHose 2:1080
- FireProt 3:647
- Firewalls 1:243
- First Order Inductive Learner (FOIL) 1:399
- First Passage Time (FPT) 1:749
- first PC(PC1) 2:392
- first principle methods 1:78
- first-generation sequencing 3:293
- first-order logic (FOL) 1:786
- first-principles molecular dynamics (FPMD) 2:558
- Fischer's randomization test 2:682
- Fish Ontology (FISHO) 2:109, 2:1126, 2:1128F
 - adoption in 2:1127F
 - inferring process 2:1128F
 - usage of classification sources comparing to 2:1129F
- FishBase 2:1110, 2:1126
- Fisher discriminant analysis 1:486, 3:63
- Fisher discriminant analysis 1:301, 1:744–745, 1:1067, 2:1004
- Fisher linear discriminant 1:488–490
- Fisher-Irwin test 1:701F, 1:701
- Fisher's Exact test based methods 3:345
- Fisher's z-transformation 1:710–713
- Fitch's algorithm 2:706, 2:708
- fitting regression models 1:734–735
- fixation
 - expected time to 2:4–7
 - probability 2:4
- fixed degree distribution random graph (FDDRG) 1:90
- fixed part method 2:560
- flap
 - dynamics 3:709
 - elbow dynamics 3:709
 - movement 3:709
- flattened index 1:267
- flavin mononucleotide-based fluorescent proteins (FbFP) 3:836
- flaviviridae* 3:761
- flaviviruses 3:246–247
- Flaviviruses database (FLAVIdB) 3:40
- flavones 3:497–498
- flavonoids 3:497–498
- flexibility 1:237
- Flexpred 3:611–612
- float data type 1:207
- flow simulation 1:96
- Floyd-Warshall algorithm 1:165F, 1:940, 1:944–945
- Floyd-Warshall's OpenMP parallel version 1:166F
- fluctuation 2:747, 3:611–612
- fluid dynamics, modelling 2:903
- fluorescence in situ hybridization (FISH) 2:296, 2:298, 2:299, 2:318
- fluorescence microscopy 2:299
- fluorescence resonance energy transfer (FRET) 2:825, 2:849–850, 2:868, 3:433
- fluorescence-based genotyping arrays 2:236
- fluorescent detection 3:636–637
- fluorescent in situ hybridization (FISH) 2:288
- fluorescent labelling method 3:434
- fluorescent proteins (FP) 2:825
- Flux Balance Analysis (FBA) 2:440, 2:801, 2:807, 2:866
- Flux Balance Constraints package 1:144
- fluxomics 2:417, 3:821
- Flybase 2:1011, 2:1012F, 2:1013F
- Flynn taxonomy 1:161–162
- FMS-like tyrosine kinase 3 (FLT3) 3:754F, 3:755–757, 3:756T
- Fokker-Planck equation 1:749, 2:10
- fold classification based on structure-structure alignment of proteins (FSSP) 3:513
- fold-change (FC) 3:456, 3:815
- fold(ing)
 - classification databases 2:467
 - comparisons 3:620–621
 - nuclei 2:452
 - recognition 2:452, 2:505
 - and sequence independent motifs in protein structures 3:621–622
 - similarity 2:476
- food storage polysaccharides, structural features of 3:524–525
- food webs 2:1131
- foot-to-foot interactions 3:539
- force field 2:550–552, 3:671–672
 - in CG MD simulation 2:560–561
 - force field-based functions 3:274
- force matching. *see* multiscale coarse-graining (MC-CG)
- force-based algorithms 1:1022
- force-decomposition 2:1146
- force-directed algorithms (FDA) 1:1089
- Ford-Fulkerson algorithm 1:940, 1:945, 1:948
- foreground model. *see* motif model
- forest fire sampling (FFS) 1:90, 1:92
- Forkhead (Fkh) 3:74, 3:78
- Formal Concept Analysis (FCA) 1:595, 1:596, 1:597
- formaldehyde-assisted isolation of regulatory elements-seq (FAIRE-seq) 2:262
- formalin-fixed paraffin-embedded tissues (FFPE) 3:914
- 5-formylcytosine (5fC) 3:341, 3:342
- forward Kolmogorov equation 2:10–11, 2:26–29, 2:27F
- forward propagation 1:637
- forward stepwise selection 1:493
- forward-backward approach 1:757
- fosamprenavir 3:744T–748
- fossil taxon 2:721
- Foundation for Intelligent Physical Agents (FIPA) 1:313, 1:315
- foundational model of anatomy (FMA) 1:840–841
- foundational ontologies. *see* top-level ontologies
- four-dimensional imaging (4D imaging) 2:993
- four-gamete test (FGT) 3:442–443
- 4D Genome 2:129
- Fourier Transform (FT) 1:481
- Fourier transform infrared spectrum (FT-IR) 2:403
- Fourier Transform Ion Cyclotron Resonance (FTICR) 2:1064, 2:1069–1070
- Fourier-transformed, and correlation coefficients 2:517
- FragGeneScan 3:207–208
- Fragment assembly of RNA with full-atom refinement (FARFAR) 3:597
- fragment ions 2:1066
 - library or assay library 3:875–877
 - spectra 3:857–858
- fragment library creation 1:781
- fragment-based approach 1:774
- fragment-based drug design to target EphA4 kinase domain 3:757
- fragment-based lead discovery 3:753
- fragment-based protein structure prediction 1:778–779
 - application to 1:778–779
 - EdaRose 1:779
 - test case 1:781–782
- fragment-based virtual screening approach 3:768
- fragment-ion indexing method 2:87
- fragmentation sequencing (Fragseq) 2:580
- fragmentation techniques 2:897, 2:1066–1067
- fragments per kilobase of transcript per million mapped reads (FPKM) 2:360, 2:380, 3:119, 3:357, 3:455

- Frame Logic (F-Logic) 1:785–786
 Framework for Evaluation of Reaction Networks (FERN) 1:1055
 framework regions (FR) 2:940
 free, online collaborative encyclopedia 2:1116–1117
 free and open access to biodiversity data 2:1115–1116
 free binding energies of protein-ligand complexes 3:659–660
 free energy
 calculation 3:614
 minimization approach 3:11–12
 Free Energy Perturbation (FEP) 3:3
 free energy surface (FES) 2:553
 free induction decay (FID) 2:427, 2:465
 free-modelling (FM) 1:261
 FM-index 3:298–299, 3:437
 frequency distribution 1:650T, 1:672–674, 1:673T, 1:676–677
 qualitative variable 1:673–674
 quantitative variable 1:674
 frequent itemsets 1:359–360
 mining algorithms 1:362–363
 Apriori 1:363, 1:364
 Eclat 1:363, 1:365
 FP-growth 1:363–366, 1:365
 frequent patterns (FP)
 FP-growth algorithm 1:363–366
 FP-tree 1:363
 mining 1:358, 1:367
 Freshwater Ecoregions of World (FEOW) 2:1111
 Fruchterman&Reingold algorithms 1:1089
 fruit fly (*Drosophila melanogaster*) 2:328, 3:103–104
 full shotgun metagenomics 3:203, 3:204
 full-length expressed stable isotope-labelled quantification (FLEXIQuant) 3:872
 fully connected HMMs (fHMMs) 2:913
 function annotation-based methods 3:288
 functional analysis 2:405–406, 3:189, 3:800–801
 diagnosis & prognosis 3:800–801
 of ncRNAs–miRNAs 2:188, 2:188T
 of non-coding regulatory elements 2:186–187
 of variants 2:185–186
 Functional Analysis through Hidden Markov Models (FATHMM) 3:3
 functional annotation 3:185
 of hypothetical protein 3:625–627
 of variants 3:382
 Functional Annotation of Animal Genomes (FAANG) 3:144
 Functional Annotation of Animal Genomes project (FAANG project) 3:138
 functional annotation of mammalian (FANTOM) 2:186–187
 functional annotation of mammalian genome project (FANTOM project) 3:195
 functional brain connectivity concept 2:807–808
 functional characterization 2:50
 functional class fingerprints (FCFP) 2:619
 functional class scoring. *see* quantitative enrichment analysis (QEA)
 functional convergence 3:968
 Functional Disease Ontology (FunDO) 1:844
 functional divergence mechanisms 2:482, 2:483F
 functional elements in genome 2:182–183
 functional enrichment analysis 1:896, 3:218–219, 3:219–220, 3:222–224, 3:224, 3:224–227;
 see also correlation analysis
 adjustment for multiple hypothesis testing 3:220
 calculation of enrichment score 3:219–220
 enrichment types 3:219
 estimation of enrichment score significance 3:220
 functional annotation
 chart 3:225F
 table page 3:226F
 Gene Ontology Consortium 3:220
 GSEA 1:896–897
 link of enrichment analysis tools 3:227T
 MEA 3:219
 MEA 1:897
 SEA 1:896
 Functional Families (FunFams) 2:483
 functional genomics 2:118
 gene regulation 2:119–120
 genetic interaction mapping 2:118
 microarray data analysis 2:120–121
 SAGE 2:121–122
 SNPs 2:118–119
 Functional Genomics Data (FGED) 1:137
 annotare 1:139
 MAGE-TAB 1:138
 MGED ontology and ontology for biomedical investigations 1:138–139
 MIAME 1:137–138
 MINSEQE 1:138
 Functional Genomics Investigation Ontology (FuGO) 1:138–139
 functional groups or functionalities (FG) 2:610
 functional magnetic resonance imaging (fMRI) 2:808
 functional module 1:982–983
 functional screens 3:184
 Functional Semantic Similarity Measure Between Gene-Products (FuSSiMeG) 1:891
 functional similarity matrix (FunSimMat) 1:891
 functional structural ensembles 2:509
 Fungi-Ensembl database 2:252
 FunRich application 3:221–222
 FunSimMat 1:892
 Fuse algorithm 2:819
 fusion
 fusion-type strategy 1:522
 genes 2:332
 fuzzy
 clustering 1:967
 reasoning process 3:3
 Fuzzy Formal Concept Analysis (FFCA) 1:582
 fuzzy logic (FL) 3:3
 Fuzzy-C-Mean algorithm (FCM algorithm) 2:1032
- ## G
- G protein-coupled receptors (GPCRs) 2:36T, 2:37F, 2:38F, 2:460, 2:633, 3:252, 3:261, 3:379
 G-quadruplexes 2:571
 G-quadruplex-forming DNA motif 3:506–508
 G-SESAME web tool 1:892
 G4 Interacting Proteins DataBase (G4IPDB) 2:572
 G4LDB database (G4LDB database) 2:571
 Gabriel's confidence intervals method 3:442–443
 GAGE 3:305
 Galaxy 2:335, 3:139
 case study with 3:145–147, 3:147F, 3:148F
 cloud-based platform 1:172–174
 Galaxy-M 2:401
 project 2:401
 GALAXY server 2:197–198
 Galaxy Tool Shed 3:146, 3:147F
 GalaxyPepDock program 3:4
 gallbladder adenocarcinoma 1:841
 Galton skewness 1:680
 Gamma diversity 3:206
 correction 2:528
 distribution 2:714
 example 1:739
 gaps 3:332
 penalties 3:314
 “Garbage In, Garbage Out” 2:397
 Garnier, Osguthorpe and Robson method (GOR method) 3:519
 Garuda Gateway 3:70
 Garuda platform 3:69–70
 gas chromatography (GC) 2:396, 2:403, 2:411, 2:894–895, 3:166, 3:327, 3:465
 gas chromatography–mass spectrometry (GC–MS) 2:396, 2:399, 2:412, 2:428, 2:635, 3:464, 3:466
 gastrointestinal tract (GI tract) 3:200
 gastrula 3:825
 GATK HaplotypeCaller 2:185
 Gauss-Markov conditions 1:724
 and diagnostics 1:726
 confidence interval 1:726, 1:726–727
 inference for model 1:726
 tests of significance and confidence regions 1:726
 Gauss-Newton method 1:422
 Gaussian distribution. *see* normal distribution
 Gaussian elimination 1:7
 Gaussian filters 2:995

- Gaussian function 2:292
- Gaussian kernel. *see* radial based function (RBF)
- Gaussian mixture clustering imputation (GMCimpute) 1:473
- Gaussian mixture model (GMM) 1:279, 1:441, 2:998, 3:611
- Gaussian polymer chain model (gpc model) 2:297–298
- Gaussianity 1:748
- GBOOST (software package) 3:928–929
- GBrowse 2:197–198, 2:251, 2:253, ,, 2:254F
- GDT-high accuracy (GDT-HA) 1:71
- GDT-template score (GDT-TS) 1:71
- GeCo algorithm 1:483
- gefitinib 3:744T–748, 3:754F
- Gen-Bank format 1:132–133
- GenABEL package 3:382
- GenBank 3:203
- GenBank BLAST 3:990, 3:991F
- GENCODE 2:167–168, 2:344–345
- GenDR 2:346
- gene association file format (GAF) 2:5
- gene co-expression networks (GCNs) 1:919–920, 1:1016–1018, 2:332, 2:800–801, 2:806
- gene expression 1:1057–1058, 2:142, 3:357, 3:481–483
- analysis 1:223, 1:224T, 2:389
- comparison of DNA methylation and 3:358
- microarrays 2:155
- profiling datasets 2:1095
- profiling technologies 2:374
- differential 2:374F
- microarray approach to 2:375
- sequencing approach 2:375–377
- Gene Expression Barcode 2:1048
- Gene Expression Omnibus (GEO) 1:114, 2:102, 2:342, 2:343, 2:388, 2:1095, 3:357–358
- Gene Expression Omnibus-NCBI (GEO-NCBI) 2:342
- Gene Expression Programming (GEP) 2:665
- gene expression regulation 3:806, 3:806F, 3:807–808, 3:809, 3:810
- chromatin
- state 3:809
- structure 3:807
- DNA methylation 3:809
- GRN 3:809
- histone modification 3:807
- identification of relationship between various factors and 3:808
- and lncRNA 3:809–810
- measurement technologies
- long read 3:809
- microarray 3:808
- short read 3:808–809
- methylation analysis using ChAMP 3:810
- miRNA 3:807, 3:809–810
- post-transcriptional regulation 3:807
- pre-transcriptional regulation 3:806
- promoter methylation 3:806–807
- RNA structure 3:809
- state identification using STAN 3:810
- TFBS 3:806
- TFBS/histone modification 3:809
- gene functions 3:179
- analysis 2:1
- using bioinformatics strategy 3:444
- gene mapping 2:242
- evolution of genetic markers and human genetic maps 2:242–243
- human linkage analysis in age of next-generation sequencing 2:247
- interpolated genetic maps for shared IBD segment analysis 2:247
- linkage analysis for mapping Mendelian disease genes 2:244
- from meiosis to centiMorgans 2:242
- in plants and animals 2:247–249
- Gene Matrix Transposed format (GMT) 1:1076
- Gene on Diseases (GoD) 1:911
- Gene Ontology (GO) 1:102, 1:130, 1:203–204, 1:474, 1:477, 1:579, 1:583, 1:700–701, 1:788, 1:810–811, 1:823, 1:823–825, 1:829, 1:832, 1:867, 1:868, 1:870–871, 1:889, 1:899, 1:908, 1:999, 1:1024, 1:1097, 1:1104–1105, 2:1, 2:1–2, 2:2F, 2:4–5, 2:5, 2:8–9, 2:20, 2:159, 2:187, 2:203, 2:331–332, 2:347, 2:463–464, 2:478–479, 2:634, 2:818, 2:840T, 2:889, 2:1046–1047, 2:1050–1051, 2:1099, 3:4, 3:84, 3:162, 3:218, 3:288, 3:317–318, 3:800, 3:821, 3:887
- accessing gene ontology project data 2:5
- annotations 2:2–4
- browsers 1:825
- database 3:444
- enrichment analysis 3:456
- evaluation measure for gene ontology coherence 1:981–982
- formats and availability 1:826
- pathways analysis 3:100, 3:100F
- relations 1:824–825
- statistics 1:825–826
- structure 1:823–825, 1:824F
- gene ontology annotation (GOA) 1:826–827, 1:827F, 1:867, 1:867–868, 1:868
- database 1:569, 1:867–868, 1:868
- repositories 1:829
- true-path-rule 1:827
- usage in computational applications 1:827–829
- gene ontology consistency (GOC) 1:999–1000
- Gene Ontology Consortium 3:220
- AmiGO 3:220–221
- BiNGO 3:222
- bioinformatics tools 3:221T
- Blast2GO 3:222
- DAVID 3:220
- FunRich 3:221–222
- GOplot 3:222
- GOzilla 3:221
- gene prediction 3:306–307
- approaches 2:182–183
- programs 3:231
- tools 3:152
- gene prioritization
- using semantic similarity 1:898–899
- computation in ontologies 1:901–903
- gene ontology 1:899
- HPO 1:899
- ontology 1:899
- semantic similarity 1:899–900
- tools 1:907, 1:907–908, 1:909, 1:910–912, 1:912
- ontologies 1:908
- semantic similarities 1:908–909
- Gene Prioritizing Approach using Network Distance Analysis (GenePANDA) 2:810
- gene product information (GPI) 2:5
- Gene Reference Into Function (GeneRIF) 1:843
- Gene regulation 1:147, 2:183
- bioinformatics tools used for regulation of genes 2:120
- databases 2:798
- pathways 1:1048, 1:1052–1055, 1:1053F, 1:1057F
- gene regulatory pathways databases 1:1055–1056
- gene regulatory pathways visualization and analysis tools 1:1054–1055
- pattern discovery 2:120
- promoter analysis 2:120
- regulatory regions 2:120
- Gene Regulation Ontology (GRO) 1:570
- gene regulatory network (GRN) 1:916–917, 1:1016, 2:155, 2:806, 2:867, 3:67–68, 3:808, 3:809, 3:822
- application for precision medicine 2:162
- central dogma and regulating elements for biological networks 2:156F
- dynamic network models and inference methods 2:159
- expression data for reconstructing 2:156–157, 2:157F
- inference 2:161–162, 3:822
- network validation and assessment of network inference methods 2:160–161
- static network models and inference methods 2:157–158
- Gene Regulatory Network Decoding Evaluations tool (GRENDEL) 1:1055
- gene regulatory pathways 2:796
- databases 1:1055–1056
- databases features comparison 1:1057T
- tools features comparison 1:1056T
- visualization and analysis tools 1:1054–1055
- gene set enrichment analysis (GSEA) 1:477, 1:896, 1:896–897, 1:1101–1102, 2:129, 2:802, 3:120, 3:219, 3:456, 3:821, 3:821–822
- gene set enrichment (GSE) 2:331–332
- gene structure-based splice variant deconvolution method (DECONV) 2:224
- Gene Transcription Regulation Database (GTRD) 2:140
- gene transfer format (GTF) 3:795

- gene-based splicing score (GBSS) 2:225
- gene-gene interactions
- analysis and interpretation 2:171–173
 - computational tools for 2:118, 2:119T
 - exploring genetic interactions and solutions 2:173–175
 - genetic interaction modeling and predict complex diseases 2:171
 - investigating additional complexity in precision medicine 2:175
- gene-gene-environment interaction models 2:175
- gene(s) 1:148, 2:221, 3:815, 3:820, 3:983
- annotation database 3:444
 - borrowing strength across 2:383
 - co-localisation-based methods 3:288
 - counting 3:308–309
 - duplication 3:965, 3:968
 - duplications as outgroup 2:732
 - events 3:965
 - elongation 3:965–966
 - finding problem 1:515
 - gain and loss 3:970
 - gene fusion-based method 2:826, 3:288
 - interaction 2:118
 - involving in cell differentiation 3:827, 3:828F
 - and isoforms 2:187–188
 - knockout techniques 2:807
 - neighbourhood-based methods 3:288
 - order 2:263
 - pattern 3:68
 - product 2:3
 - profiling experiments 2:372
 - prospector 1:910
 - reference datasets 2:946
 - set-based scoring 2:802
 - sharing 3:969–970
 - transcription 2:372
 - transcription during cell differentiation 3:825–826
- GeneAnswers 1:844
- Genecards 2:104, 2:104–105, 2:105
- Genechip 1:128
- GeneDecks 2:104
- GeneDistiller 1:910
- GeneFriends 1:910
- gene-gene co-expression 3:998
- GENEHUNTER 2:246–247
- GENEID Gene Model 2:197
- GeneMapper 3:434
- GeneMark 3:305
- GENEMARK program 2:196
- GeneMarkS-T tool 3:310
- GenePix Array List format (GAL format) 3:117
- General and Robust Alignment of Multiple Large Interaction Networks (Grmlin) 1:1010
- general feature format (GFF) 2:358
- general GWAS analysis 3:382
- general linear models (GLM) 3:346T, 3:346–347
- general LM 1:731
- General Markov Model (GMM) 2:713
- General Mixed Yule Coalescent approach (GMYC approach) 3:992
- General Public Licence (GNU) 1:176
- General Time-Reversible model (GTR) 2:708, 2:713
- general transcription factors (GTFs) 2:134
- general umbrella sampling scheme 2:553
- Generalised Association Rules 1:370
- generalized born energy for polar solvation energy calculations 2:553
- generalized HMM (GHMM) 2:197
- Generalized Linear Mixed Model Prediction of sQTL (GLiMMPs) 2:227
- generalized linear model (GLM) 1:385, 1:424–425, 1:723, 1:729, 1:731, 1:731–732, 2:237, 2:382–383
- components 1:424–425
 - estimating parameters 1:426–427, 1:427
 - fitting generalized linear models 1:427
 - implementation in R 1:733–734
 - logistic regression 1:425
 - model checking 1:427–428
 - modelling categorical data 1:732
 - modelling count data 1:732–733
 - Poisson regression 1:426
- generalized pair HMMs 2:203
- generalized Poisson model (GP model) 2:226
- GeneRanker 1:910
- Generating Vectors (GVs) 2:441
- Generative model for the alternative splicing array platform (GenASAP) 2:224
- generative models 1:279
- HMM 1:280–281
 - mixture models 1:279
 - probabilistic graphical models 1:279–280
- GeneReader application 3:435
- generic functions 1:199
- Generic Model Organism Database (GMOD Project) 2:252
- generic ontologies 1:809
- generic procedure 1:450
- Genes to Cognition database (G2Cdb) 2:1050–1051
- GeneSeeker 1:910
- genesim function 1:893
- GENESIS system 1:252–253
- genetic algorithm (GA) 1:280, 1:321, 1:323–326, 1:324, 1:774–775, 1:776, 1:1006, 2:1004, 3:2, 3:3, 3:525, 3:748–749
- classifier 1:417, 1:417–418, 1:419T
 - fitness function 1:418
 - operators 1:418–419
 - optimization problems and classification tasks 1:419
 - scheme of simple 1:419
 - vocabulary 1:417
- Genetic Algorithms Image Clustering (GA-IC) 1:443
- Genetic Algorithms Image Clustering for Document Analysis (GA-ICDA) 1:443
- genetic interaction mapping 2:118
- computational tools for gene-gene interactions 2:118
- gene interaction 2:118
- Genetic Normalized Cut (GeNCut) 1:443
- Genetic Partial Least Squares (G/PLS) 2:664
- genetic programming (GP) 1:277, 1:347
- Genetic Sensory-Response units (GENSOR units) 2:140
- genetic(s) 2:171
- basis of variability in drug response 3:379–380
 - clustering methods 3:367
 - drift 3:943–944
 - genetic bottleneck 3:943–944, 3:944F
 - process 2:2
 - epidemiology databases 2:1060–1061
 - factors affecting nucleosome positioning 2:309–310
 - genetic interaction
 - modeling and predict complex diseases 2:171
 - and solutions 2:173–175, 2:174T
 - hitchhiking 2:264
 - maps 2:242, 2:243–244
 - markers evolution 2:242–243
 - mutations impact STNs 2:860–861
 - variation 2:235–236, 3:364
- GeneTrail 3:219
- GENETREE 2:753
- GeneWanderer 1:910
- GeneWeaver 2:1048
- GENEWISE 2:199–200, 2:202
- genie 1:910
- GenInfo Identifier (GI) 1:133
- genome 2:288–289, 2:318, 3:327, 3:425
- assembly 1:224, 1:224T, 3:160–161, 3:304–305
 - experimental methods to assay DNA modifications in 3:342–344, 3:343T
 - genome-scale analyses 2:737
 - genome/transcriptome assembly 2:357–358
 - imprinting 2:128
 - mapping proteins to 3:313
 - organization 2:294–295
 - quality control and mapping to 3:344
 - sequence 2:268, 2:288–289, 3:164, 3:425–426, 3:814
 - techniques 3:819
 - wide measurement of RNA unfolding 3:579–580
- genome alignment 2:268, 2:272–273, 2:273, 2:276–277
- background/fundamentals 2:269
 - challenges on 2:270–271
 - steps in 2:271–272
- genome analysis 1:111, 3:113–114; *see also* transcriptome analysis
- ChIP-Seq pipeline 3:113–114
- mutation and genomic variation 3:115–116
- pipelines 3:116
- Genome Analysis Tool Kit (GATK) 2:357, 2:1080, 3:116
- genome annotation 2:195, 2:360–361, 3:152, 3:154, 3:179, 3:180T, 3:305–307, 3:722
- ab initio* and similarity years 2:195–198
 - comparative and homology years 2:200–202

- genome annotation (*continued*)
 curated data repositories for 3:153–154
 dynamic view 3:152
 Markov models and gene prediction 3:306–307
 population years 2:205
 RNA and regulatory years 2:204–205
 software 3:152–153
 static view 3:152
 Genome Annotation Assessment Project (GASP) 2:197
 genome browser (GBrowse) 2:251, 2:253, 2:253–255, 2:344, 2:345, 2:1051, 3:300
 Wormbase and Gbrowse 2:253
 genome characterization centres (GCCs) 2:106, 2:108
 Genome Data Analysis Centers (GDACs) 2:108
 genome databases 2:251–252, 2:253–255
 broad databases 2:252
 and browsers for cancer 2:1077, 2:1077–1079, 2:1078T
 human 2:252–253
 specific organism databases and GMOD Project 2:252
 Genome Differential Compressor method (GDC method) 1:484
 genome informatics
 genomic functional informatics 2:185–186
 genomic structural informatics 2:180–181
 genomic technologies in genomic studies 2:179–180
 population 2:188–190
 Genome Re-sequencing Encoding (GReEn) 1:484
 genome records analysis facilities (GDACs) 2:106
 Genome Reference Consortium (GRC) 2:178
 Genome Sequencing Facilities (GSCs) 2:108
 genome wide association studies (GWAS) 1:277–278, 1:763–764, 1:1047–1048, 1:1069, 2:264–265, 3:394
 Genome wide Event finding and Motif discovery (GEM) 3:76, 3:78
 Genome-analysis-toolkit package tools (GATK package tools) 3:437
 genome-scale metabolic models (GEMs) 2:440–441
 genome-wide association studies (GWAS) 2:118, 2:171, 2:190, 2:235, 2:359, 2:433, 2:861, 2:1094–1095, 3:67–68, 3:115–116, 3:381, 3:382, 3:388, 3:434, 3:441, 3:999
 CD/CV hypothesis 2:235
 experimental designs 2:236
 genetic variation and LD 2:235–236
 meta-analysis, replication and GWAS follow-up 2:238–239
 statistical analysis 2:236–238
 genome-wide haplotype association study 3:441, 3:444–445
 applications 3:444
 framework 3:441, 3:442F
 genome-wide population genetics 3:365
 genome-wide probing of RNA structure 3:577T, 3:578
 chemical and enzymatic structure probing of RNA 3:575
 constraints and databases 3:581
 emergence of single-molecule sequencing technologies 3:581–582
 genome wide measurement of RNA unfolding 3:579–580
 high-throughput structure probing and automation 3:575–576
 beyond one modification per RNA molecule 3:578–579
 phylogenetic conservation of directly observed duplexes 3:580
 rigorous computational models of high-throughput structure probing data 3:578
 RNA structure probing with deep sequencing 3:576–578
in vivo chemical probing of RNA 3:578
 genome-wide scanning of gene expression 3:452, 3:452F
 clustering analysis of transcriptome complexity 3:457F
 functional profiling with transcriptome data 3:455–456
 genomic-scale transcriptome profiling 3:452–454
 materials and methods 3:458
 transcriptome analysis with HISAT2-Cufflinks2 pipeline 3:453F
 genome-wide significance 2:237
 Genome-Wide Study (GWAS) 1:171–172
 Genome3D 2:468
 collaborative project 2:478
 Genomes Online Database (GOLD) 3:152
 GENOMESCAN 2:202
 GenomeVIP 1:254
 genomic analyzer (GA) 1:129
 Genomic and Proteomic Knowledge Base (GPKB) 1:829, 1:1096
 Genomic Data Commons (GDC) 2:1077–1079
 genomic functional informatics 2:185–186
 functional analysis
 of genes and isoforms 2:187–188
 of ncRNAs–miRNAs and lncRNAs 2:188
 of non-coding regulatory elements 2:186–187
 of variants 2:185–186
 genomic islands (GIs) 3:955
 genomic sequences analysis
de novo genome assembly 3:301–302
 genome annotation 3:305–307
 mapping sequencing data 3:297–298
 preprocessing sequencing data 3:296
 sequence alignment 3:295
 genomic structural informatics 2:180–181
 eliciting linear structure of genome 2:180–181
 identifying functional elements in genome 2:182–183
 identifying genomic variants 2:183–185
 inferring 3D structure of genome 2:181–182
 genomic(s) 1:486, 2:124, 2:808–809, 2:1059, 3:1, 3:194, 3:194–195, 3:425, 3:482–483, 3:965
 boundaries 2:310–311
 data compression 1:483–484
 distribution of cytosine modifications 3:347
 genomic nucleosome positioning 2:309
 genomic-scale transcriptome profiling 3:452–454
 inflation factor 2:237
 machine learning approaches in 3:120–121
 medicine databases 2:1059
 screening methods 3:783–786, 3:785–786
 sequences 1:221, 1:367, 2:1106, 3:293
 structural families in genomic era 2:479
 technologies in genomic studies 2:179–180
 variants identification 2:183–185, 2:184F
 GenoQuery 1:1096
 Genotype-Tissue Expression (GTEx) 2:344, 3:195, 3:457
 genotypes 2:9, 2:178
 calling 3:438
 data 3:441
 genotype-tissue and phenotype interactions 2:343–344
 genotype-phenotype associations 3:47–49
 imputation 2:239, 3:390–392, 3:391F, 3:392F
 GENSCAN 2:197
 Gentrepid 1:910
 geographic information systems (GIS) 2:99
 geometric accuracy 1:981
 geometric active contour model 2:1032
 geometric margin, functional and 1:505, 1:506
 geometric random graphs 2:816–817
 Georgian biodiversity database 2:1110–1111
 germline 2:2
 repertoire inference 2:976
 variant calling 3:168
 GERMLINE algorithm 3:368
 germline centers (GC) 2:942
 GERMLASM 2:347
 GHOST aligner 1:1007
 Gibbs free energy of molecule 3:614
 Gibbs sampler (GS) 2:285, 3:338, 3:448–449
 Gibbs sampling algorithm 1:17, 3:448–449
 giberrellanes 3:497–498
 Gillespie algorithm 1:751, 2:866
 Gini coefficient 1:554
 Gini index 1:388, 1:554
 GitHub 3:137
Glaciozyma antarctica 3:727
Glaciozyma antarctica Integrated Exploration Resource (GlacIER) 3:727
 GLAD4U 1:910
 Glazier Graner-Hogeweg models. *see* cellular Potts models
 Gleaning 2:1101
 Glimmer 3:305, 3:307
 GLIMMER program 2:196
glm function 1:733
 global aligners 1:1001

- global alignment 1:23, 1:24–26, 1:102–103, 1:103F, 1:104, 1:107F, 2:270, 3:313
 approaches 1:1068
 quality measures for 1:1068–1069
 dynamic programming for 3:314
 Global Alliance for Genomics and Health (GAGH) 1:130
 global approach-based algorithms 1:473
 Global Biodiversity Information Facility (GBIF) 2:103, 2:1117
 database 2:1114F, 2:1115, 2:1115–1116, 2:1116F, 2:1117F
 occurrences information in 2:1114F
 searchable fields in 2:1115F
 global cluster coefficient 1:936
 global computation design and evaluation 1:215
 global consensus-based methods 3:588
 Global distance test (GDT) 1:70–71, 2:502, 3:264, 3:593
 Global Distance Test High Accuracy variant (GDT_HA) 3:593
 Global Distance Test Total Score (GDT_TS) 2:502, 3:593
 Global IDDT Score 1:71
 global methods 2:810, 2:810–811
 Global Natural Products Social Molecular Networking (GNPS) 2:402
 global optimization 1:321–322
 Global Proteome Machine database (GPMdb) 2:79
 global run-on sequencing (GRO-seq) 2:333
 global sequence alignment 3:981
 global sequence alignment-based methods 3:447
 global warming on ecological networks 2:1135
 global-as-view method (GAV method) 1:1094
 GLOBETROTTER method 3:369–370
 globular proteins 2:28, 2:446, 3:511
 globus based grids 1:231–232
 EGEE/EGI grid platform 1:231–232
 Globus Toolkit 1:163, 1:171
 “glocal” alignment 2:270
 glucocorticoid receptor evolution (GR evolution) 3:946
 glucocorticoids (GCs) 3:946
 GLYCAM06 3:671–672
 glycans 2:1072
 glyceraldehyde 3-phosphate (GAP) 3:638
 glyceraldehyde 3-phosphate dehydrogenase (GADPH) 3:765
 glycerolipids 2:894
 glycerophospholipids 2:894
Glycine max 3:428
 glycogen 3:525
 glycogen synthase kinase-3 (GSK-3) 3:785
 glycopeptides 2:1072
 glycoproteomics 2:1072–1073
 glycosaminoglycan
 attachment site 2:35
 receptors 3:775
 glycosylation 2:15
 glycosylphosphatidylinositol (GPI) 2:57
 GNormPlus 1:606
 GNU Public License (GPL) 1:1058
 GO Consortium (GOC) 1:823
 GO-Causal Activity Model (GO-CAM) 2:5
 Go-plus 2:5
 gold standard 2:806, 2:1104–1105
 Goldman-Yang model 2:715
 Golm Metabolome Database 2:403, 3:470
 goodness of hit list (GH list) 2:681, 2:681–682
 Google Apps Engine 1:160–161, 1:258
 Google Cloud Platform 1:240, 1:240–242, 1:241F
 Google Earth 3:1–2, 3:6–7, 3:7F
 Google Image Search 2:1019
 Google Plugin for Eclipse 1:247
 Google’s PageRank 1:952, 1:1001
 Google’s Tensor Processing Unit (TPU) 2:1147
 GOpilot 3:222
 GOzilla application 3:221
 GOSemSim 1:892–893
 GOTHic method 2:321
 GOToolBox web server 1:892
 GOvis package 1:893
 GPCRs Database (GPCRdb) 2:468, 3:384
 GPS-MSP 2:19
 GPS-PAIL tool 2:18–19
 GPSeq 2:226
 Grabcut algorithm 2:997
 gradient clipping 1:641
 gradient descent method 1:422, 1:622, 1:638–640
 gradient descent optimization algorithms 1:640
 Gradient Outlier Factor (GOF) 1:458
 graft-versus-host disease (GVHD) 2:966
 GRAILEXP 2:202
 Gram matrix 1:496
 Gram-negative bacteria 3:105
 GRand AVerage of hydropathY (GRAVY) 2:30
 Granger Causality-based methods 2:808
 Grantham matrix 2:716
 granularity 1:233
 graph algorithms 1:940, 1:940–943
 analysis and assessment 1:947
 BFS 1:940–943, 1:941, 1:941F
 DFS 1:940–943, 1:942, 1:942F
 examples 1:948
 Ford-Fulkerson algorithm 1:948
 graph traversal algorithms 1:948
 spanning trees 1:948
 TSP 1:948
 Ford-Fulkerson algorithm 1:945
 minimum spanning tree 1:945–946
 nearest neighbour algorithm 1:946–947
 shortest path algorithms 1:943
 GRaph ALignment (GRAAL) 1:98, 1:102, 1:105–107, 1:106F, 1:107T, 1:1003–1004
 algorithms for graph alignment 1:103–104
 C-GRAAL 1:1004–1005
 global alignment 1:102–103, 1:104
 graph querying 1:103, 1:105
 H-GRAAL 1:1004
 local alignment 1:103, 1:104–105
 MI-GRAAL 1:1004
 pairwise and multiple alignment 1:102
 software tools 1:107–108
 graph drawing with intelligent placement (GRIP) 1:1022
 graph indexes/descriptors
 centrality indexes 1:83
 cohesion indexes 1:84–85
 indexes 1:83
 network representation 1:81–82
 other indexes 1:86
 graph isomorphism 1:933, 1:933–936, 1:935–936, 1:937–938
 analysis and assessment 1:937–938
 directed graph 1:934F, 1:935F
 examples 1:938
 methodologies 1:936–937
 two isomorphic graphs 1:936F
 undirected graph 1:933F
 graph modeling language (GML) 1:1027
 graph-based clustering method 1:1041–1042, 1:1041F
 MCL 1:1041–1042
 SC 1:1042–1043
 SPICi 1:1043
 graph-based exon-skipping scanner (GESS) 2:227
 graph(s) 1:922, 1:928, 1:933, 1:940, 1:959F, 1:1047–1048, 1:1048–1049, 1:1085, 1:1085F, 2:653–654
 classification 1:975
 cousins 1:99
 database 2:98–99, 2:1065T, 2:1065–1066, 2:1066, 2:1069
 drawing 1:1087–1089
 graph-based aligner 2:271
 graph-based molecular alignment 2:645–646
 graph-theoretic approaches 1:441–442
 kernels 1:1067
 mining approaches 1:1067
 modeling 1:972
 partitioning algorithms 1:1022
 pattern
 combining 1:802–803
 mining 1:1067
 querying 1:103, 1:104F, 1:105
 representations of molecules 2:641
 simplification 3:304
 theory 1:922–925, 1:935, 1:1105, 1:1105F, 2:640–641, 3:284
 methodologies 1:925–926
 undirected and directed graph 1:923F
 traversal algorithms 1:6, 1:948
 graphic processing unit (GPU) 1:162, 2:825
 accelerated MD simulations 2:559–560
 computing 1:211–212
 servers 1:243
 Graphical Gaussian Models 2:806
 graphical methods 1:199, 1:672
 Graphical User Interface (GUI) 1:65, 1:160–161, 1:176, 1:209, 1:252–253, 1:305, 1:331, 1:1051, 2:84–86, 2:401, 2:525–526, 2:527–528, 2:529, 2:530, 2:1142, 3:145
 Graphlet Correlation Distance (GCD) 2:817
 Graphlet Correlation Matrix (GCM) 2:817

- graphlet degree vector (GDV) 3:419
 graphlets 2:816–817, 2:816F
 algorithms for graphlet identification 2:817
 frequency distribution 2:816
 link network structure and biological function 2:817–818
 in network aligners 2:818–819
 GraphX 1:249
 grass carp genome database (GCGD) 2:346
 Great Internet Mersenne Prime Search (GIMPS) 1:163
 greedy algorithms 2:181
 greedy consensus. *see* extended majority-rule consensus
 Greedy equivalence search (GES) 2:159
 greedy seed-and-extend algorithm 1:1004
 green fluorescent protein (GFP) 3:568
 Green's function reaction dynamics (GFRD) 2:868–869
 Gremlin 2:98–99
 grid 1:163
 computing 1:230
 layout algorithm 1:1019–1020, 1:1090
 maps preparation and generation 3:277
 Grid Security Infrastructure (GSI) 1:171, 1:232
 Grid Services (GS) 1:231, 1:232
 GROningen MAchine for Chemical Simulations (GROMACS) 2:825, 3:613
 Groningen Molecular Simulation (GROMOS) 2:552, 3:672
 grouping gene expression patterns 3:456
 growth factor receptor-bound protein 2 (Grb2) 2:857
 growth hormone (GH) 3:898–902
 guanidinium modifying superfamily (GMSF) 3:731
 guanine 2:701, 3:425
 guanine at position 24 (G24) 1:277
 Guide to Receptors and Channels (GRAC) 2:633
 Gumbel distribution 3:315
 G–ner-Henry method (GH method) 2:681–682
 gut microbiome 3:200–202, 3:201F, 3:202–203, 3:211–212
 bioinformatics approaches 3:204–207
 core gut microbiome in obese and lean twins 3:208
 in healthcare and medicine 3:208–211
 molecular techniques from pre-ngs to NGS 3:202–203
 gut microflora 3:200
 gut profiling 3:200
 GX bulges 3:522
- H**
- H-bonding in nucleic acid structures 3:504–505
 H-Invitational database (H-InvDB) 2:252–253
 H3K4me1 3:355
 H3K4me3 3:354, 3:355
 H5 histone 2:289
 HADDOCK program 3:4
 setup and protocols 2:143
 Hadoop 1:248
 Hadoop-based framework 1:226
 MapReduce application 1:222
 YARN 1:248
 Hadoop Distributed File System (HDFS) 1:171, 1:222, 1:248, 1:258
Haemophilus influenza 3:425, 3:427–428
Haemophilus influenzae 3:176
 haemorrhagic fever virus (HFV) 3:108–109
 hail 2:1080
 hairpin connection 3:514–516
 HAL package 2:273
 Half maximal effective concentration (EC50) 2:455
 Half maximal inhibitory concentration (IC50) 2:455
 halophiles 3:945
 Halpern-Bruno style models 2:715
 Hamiltonian Path 1:297
 Hamiltonian process 1:940
 Hamiltonian replica exchange MD (HREMD) 2:553, 2:556
 Hamming distance 1:414, 2:807, 2:977
 hand-in-hand interactions 3:539
 haploid cells 2:289
 haplotype 2:178–179, 3:441
 analysis software 3:444
 frequency estimation 3:443
 Haplotype Inference by Pure Parsimony (HIPP) 1:298
 haplotypic phase 3:365
 haploview 1:764, 3:444
 HapMap 2:178–179
 HAPPI 2:852
 hard clustering 1:966, 2:1032
 hardware
 levels, project sizes and user experience levels 3:135
 tools and availability 1:452
 Hardy-Weinberg Equilibrium (HWE) 2:9, 3:115, 3:365, 3:441–442
 HarmoniMD 2:1053
 Hasegawa-Kishino-Yano rate matrix (HKY rate matrix) 2:16–17
 hash table 2:330
 hash-based methods 3:166–167
 hashing 2:655
 hat-matrix 1:725
 Hazardous Substances Data Bank (HSDB) 2:635
 Head and Neck database (HNdb) 2:342
 Head and Neck squamous cell carcinoma genes (HNSCC) 2:342
 Health Care Information System 1:112
 Health Information Systems (HIS) 1:266
 Health Insurance Portability and Accountability Act (HIPPA) 1:265
 health research networks 1:154
 health sciences 1:337
 KDD examples in 1:339–340
 healthcare 1:336
 HealthVault 2:1053
 “healthy” somatic cell 3:891
 heart failures with decision trees in R, predicting 1:377–379, 1:381F
 heart tissue, modelling spatial propagation of electrical activation in 2:902–903
 HELANAL-Plus program 3:519
 helical bundle 3:265–266
 helices (a, 3_{10} , p) 2:488–489
 3_{10} -helices 3:516
Helicobacter pylori 3:427–428
 Helicos single-molecule sequencing device (HeliScope) 1:129
 hematopoietic stem cells (HSC) 2:986
 mathematical modeling HSC differentiation 2:986–987
 hemoglobin (Hb) 2:591
 proteins 3:1
 hepatitis C virus (HCV) 3:41–42, 3:108–109, 3:289, 3:776, 3:933
 HERA 2:520–521
 “HerbIngredients” Targets (HIT) 2:1048
 Hereditary nonpolyposis colorectal cancer (HNPCC) 2:123
 HETCOR technique 3:525
 Hetero-nuclear Multiple Bond Correlation (HMBC) 2:427–428
 Hetero-nuclear Single Quantum Coherence (HSQC) 2:427–428
 heterochromatin 2:293–294
 heteroduplex analysis 3:433
 heterogeneity in pathway definition 1:1079
 heterogeneous datasets 3:457–458
 Heterogeneous metabolic databases 2:1020
 heterogeneous nuclear ribonucleoproteins (hnRNPs) 2:222, 2:223
 heterogeneous recombination (hotspots) 2:754
 Heterogeneous Value Difference Metric (HVDM) 1:475
 heterogenous nuclear RNA (hnRNA) 3:230
 heteronuclear single-quantum correlation method (HSQC method) 2:515
 heuristic algorithms 1:30, 2:645
 heuristic global optimization methods 1:322
 heuristic methods 1:482, 2:87, 3:437
 heuristic sequence alignment methods 2:703
 hexose metabolic process 1:827, 1:828F
 Hi-C 3:162
 analysis pipeline 2:318–319
 conformation capture by 2:149
 data normalization methods 2:182, 2:320
 experiment review 2:318
 ligation products 2:318
 methods 2:300–301
 paired read mapping 2:319, 2:319F
 technology 2:182
 HiCCUPS method 2:321
 HiCLib 2:319
 HiCNorm method 2:320
 HiCPlotter 2:322
 Hidden Markov Model (HMM) 1:261, 1:280–281, 1:563–564, 1:753, 1:755–757, 1:760, 1:760–761, 1:774–775, 2:12, 2:40, 2:48–49, 2:123, 2:134, 2:135–140, 2:196, 2:246–247, 2:277–278, 2:475T, 2:475, 2:483,

- 2:557–558, 2:828, 2:913, 2:946, 2:954, 2:1048, 2:1085, 3:3, 3:32, 3:231, 3:244, 3:305, 3:307, 3:346T, 3:346–347, 3:669, 3:981, 3:983
- decoding problem 1:758–759
- evaluation problem 1:757–758
- learning problem 1:759–760
- statistical inference in 1:757
- stochastic process 1:753–755
- hidden neuron 1:621, 1:627, 1:627–630, 1:631–632
- hierarchical agglomerative clustering approach (HAC) 1:982
- hierarchical clustering 1:278, 1:350, 1:449, 2:390, 2:391, 2:392, 2:827–828, 3:470
- Hierarchical Hyperbolic SOM (H2SOM) 1:1106
- hierarchical layout 1:1020
- hierarchical layout algorithm (HLA) 1:1089
- hierarchical methods 1:304, 1:442, 1:452, 1:745
- Hierarchical Model Composition package 1:144
- hierarchical structural classification 2:475–476, 2:476F
- architecture 2:476
- homologous family 2:477–478
- homologous superfamily 2:476–477
- protein class 2:475–476
- topology or fold similarity 2:476
- high content screening (HCS) 3:757
- high dimensional regression 1:728
- high performance cluster (HPC) 2:335
- High Performance Fortran (HPF) 1:217–218
- high performance liquid chromatography (HPLC) 3:465
- High PerformanceC++ (HPC++) 1:218
- high throughput direct assay evidence (HDA) 2:3
- high throughput expression pattern evidence (HEP) 2:3
- high throughput genetic interaction evidence (HGI) 2:3
- high throughput mutant phenotype evidence (HMP) 2:3
- high throughput omics data, integrative analysis of 3:196–197
- integrated omics data analysis 3:197
- integration of high-throughput data 3:196–197
- network-based models of integrative omics data analysis 3:197
- unsupervised integration and analysis of omics data 3:197
- high throughput omics datatypes and methodologies 3:195F
- genomics 3:194–195
- metabolomics 3:196
- proteomics 3:195–196
- transcriptomics 3:195
- high throughput sequencing (HTS) 1:137, 2:585, 2:613, 2:631, 2:973, 3:47, 3:453–454, 3:568, 3:808–809, 3:814, 3:814–815, 3:816–817, 3:965T, 3:987–989, 3:988T, 3:993
- high throughput systematic evolution of ligands by exponential enrichment (HT-SELEX) 2:147–148, 2:148
- high throughput-next generation sequencing platform (HT-NGS platform) 3:435
- high-dimension data (HD data) 3:67–68
- high-dimensional regression models 1:729
- high-level languages 1:215
- high-performance computing (HPC) 1:161, 1:163, 1:171–172, 1:209, 1:210, 1:257, 1:1099, 2:1142, 2:1151, 3:135
- bioinformatic applications 1:230–231
- BIONC based grids 1:232
- composition based languages 1:219
- distributed memory languages 1:216–217
- globus based grids 1:231–232
- infrastructures for
- cloud computing development environments 1:247
- cloud infrastructures 1:240
- object-oriented parallel languages 1:218
- performance 1:233
- crunching factor 1:233
- granularity 1:233
- job efficiency and latency time 1:233
- shared memory languages 1:215–216
- virtual paradigms 1:232–233
- High-quality automated and manual annotation of proteins (HAMAP) 2:10
- high-resolution isoelectric focusing (HiRIEF) 3:897–898
- high-resolution melting curve analysis (HRM curve analysis) 3:432–433
- high-resolution ribosome structures 3:13–14
- high-scoring segment pairs (HSP) 1:44
- high-throughput (HTP) 1:1071
- high-throughput structure probing and automation 3:575–576
- data 3:578
- innovative technologies 1:160–161
- methodologies 1:374, 1:463, 1:1047–1048, 1:1053–1054, 3:434, 3:482
- SNP array technology 3:441
- high-throughput data-collection techniques 1:915
- high-throughput docking (HTD) 1:78
- High-Throughput GoMiner 2:121
- high-throughput screening (HTS) 1:988, 2:688, 3:742–743, 3:783–786, 3:929
- high-throughput sequencing-CLIP (HTS-CLIP) 3:50–53
- High-throughput sequencing-fluorescent ligandinteractionprofiling (HiTS-FLIP) 2:148, 2:148–149
- higher order structure of XIST lncRNA in epigenetic silencing 3:581
- higher-energy collisional dissociation (HCD) 2:1066
- Hill diversity index 2:977–978
- Hill-climbing problem 2:708
- Hint Interaction Field Analysis (HIFA) 2:669
- HIPPIE network 1:1025, 2:852
- HISAT2-Cufflinks2 pipeline 3:455
- histidine-binding protein (HBP) 3:637
- histocompatibility complex class I allele (HLA) 3:384
- polymorphism 3:39
- transgenic animal models 3:244
- histogram
- equalization method 2:994, 2:995F
- techniques 1:482–483
- histome 2:129
- histone acetyltransferases (HATs) 2:18
- histone octamer 2:308
- intrinsic DNA sequence affinity 2:310
- TF binding to DNA partially unwrapped from 2:311–314
- TF competition with 2:311
- histone related databases (HistoneDB) 2:129
- histones 2:128, 2:155
- chemical modifications 2:311
- database 1:835
- modification 3:355, 3:357
- proteins 2:142
- HitPredict 2:852, 2:853
- HIV-1-protease (HIV-Pr) 3:706–707
- HIV-pr 3D-structure and dynamics 3:709
- flap movement and dynamics 3:709
- mutation effects on HIV-pr 3D-conformation 3:709–710
- schematic representation of structure 3:708F
- structural distribution of mutations associated with drug resistance 3:708F
- 2D structures 3:707F
- hive
- frameworks 1:248
- plot layout 1:1020
- HMRf method 2:321
- Hodgkin-Huxley formalism 2:904
- Holo-QSAR (HQSAR) 2:669
- homeostatic mechanisms 2:985
- Homo sapiens* 2:830, 3:427, 3:821
- Homo sapiens-Plasmodium falciparum* 3:107
- homodimers 2:821
- binding motifs 3:447
- homogeneity 1:1036
- homologous genes 2:260, 2:477–478
- comparative domain structure of genes 2:262
- homology search and functional prediction 2:260–261
- orthologs, paralogs, and others 2:260
- WGD 2:261–262
- homologous proteins 3:725
- detection 3:1–2, 3:2–5, 3:4F
- drug repositioning 3:5–6
- structure determination and prediction 3:6–7
- homologous sequences 2:703
- HOMologous STRucture Alignment Database (HOMSTRAD) 2:472T
- homologous superfamily level (H-level) 2:476–477
- homologous template sequences 1:43–45
- homologs 2:837, 3:255, 3:943
- identification 3:980–981, 3:980F
- application 3:981–983
- case studies 3:983
- homology 2:269, 3:426

- homology-based algorithms 2:182–183
- homology-based prediction methods 3:107
 - approach 2:187
 - based methods 2:145–146
 - based template finding 2:504–505
 - and conservation 2:837–838
 - and development 3:943
 - inference 3:981
 - mapping. *see* synteny mapping
- homology modelling (HM) 1:563–564, 1:774, 2:150, 2:451–452, 2:540, 2:586–587, 2:587T, 3:253, 3:276–277, 3:522–523, 3:743, 3:756
- proteins 1:38, 1:42–45, 1:43–45, 1:57–59
 - bioinformatics tools for protein structure prediction and analysis 1:38–39
 - energy optimization by simulated annealing 1:42–45
 - model verification 1:55–57
 - protein data bank 1:42
 - protein families and SCOP database 1:39
 - substitution matrix and sequence alignment 1:39–42
- homophily 1:86
- homoplasmy, resolving 2:734–735, 2:735F
- Homotopy optimization method (HOPE) 3:3
- horizontal database example 1:359T
- horizontal gene transfer (HGT) 3:953
 - in bacteria 3:957–959
 - detection 3:954
 - computational analysis of HGT genes 3:954
 - identification of HGT events using transcriptome data 3:957
 - pan-genome informatics 3:954–955
 - inter-kingdom 3:953–954
 - intra-kingdom 3:953
 - in plants 3:961
- host system, proliferating in 3:106
- host-pathogen genomes/proteomes retrieval 3:411, 3:411T
- host-pathogen interaction 2:830–831
 - applications 3:109
 - bioinformatic methods of discovery 3:106–107
 - databases for 3:108–109, 3:108T
 - infection stages 3:104–105
 - pathogens 3:103–104
- host-pathogen protein-protein interaction networks (HP-PPI networks) 3:410, 3:410–411, 3:418–419
 - case studies on prediction 3:415–417
 - detection 3:411–413
 - filtering 3:413–415
 - future directions 3:419
 - principles 3:412F
 - studies related to human infectious diseases 3:416T–417
 - validation 3:415
 - workflow for prediction 3:411
- host-pathogen protein-protein interactions (PHIs) 3:932, 3:932F, 3:935T
 - experimental methods to determine PHIs on large scale 3:933–934
 - PPI network-based analysis 3:934–935
 - across pathogenic species 3:934–935
- hosts miRNA transcription start sites (microTSS) 2:120
- hot-spot residues, predicting 3:4
- HotKnots* 3:510
- hpoGoD function 1:911
- HRGRN database 1:1055
- HSP90 chaperone family 3:425
- hubs 1:950, 2:822
 - proteins 3:105
- Hudson's algorithm 2:750
- HuGChip 3:203
- HUGO Gene Nomenclature Committee (HGNC) 1:833
- human (*Homo sapiens*) 2:252
- Human Ageing Genomic Resources (HAGR) 2:343–344, 2:346
- human and vertebrate analysis and annotation (HAVANA) 2:197–198
- human antibody responses
 - analysis of immunoglobulin gene sequences 2:943–946
 - analysis pipelines for repertoire data 2:947–948
 - diversification of immunoglobulin genes 2:940–942
 - sequencing of immunoglobulin gene repertoires 2:943
- Human Brain Transcriptome database (HBT database) 2:343–344
- Human Cancer Proteome Project (Cancer-HPP) 3:920
- human disease
 - animal models for 3:429
 - human disease-related databases 3:425
 - predicting nsSN vs. pathogenic effects in computational approaches 3:1–3
 - web servers as tool for predicting SNVs 3:5
- human embryonic stem cell (hESC) 3:827
- human ether-a-go-go-related (hERG) 2:623–624, 2:633
- Human Eye Proteome Project (EyeOme) 3:920
- Human footprint score (HF score) 2:1134
- Human Gene Mutation Database (HGMD) 2:123
- human genetic maps 2:242–243
- human genome 3:164
- human genome data 2:1093
- human genome database (GDB) 2:252–253, 2:1059
- Human Genome Organization (HUGO) 2:328, 3:388
- human genome project (HGP) 1:561, 1:1093, 2:118, 2:178, 2:179, 2:242–243, 2:251, 2:352
- Human Genome Variation Society (HGVS) 1:254, 3:388
- Human Histone Modification Database (HHMD) 2:129
- human immunodeficiency virus (HIV) 3:40–41, 3:108–109, 3:652, 3:706, 3:706F, 3:919, 3:926, 3:935
 - anti-viral drug 3:275
 - HIV-1 2:986, 3:44
 - proteins 3:105
- HIV-protease inhibitors 3:707–709
- molecular immunology database 2:937F, 2:938
- Human Leukocyte Antigen (HLA) 2:913, 3:39, 3:242
- human linkage analysis in age of next-generation sequencing 2:247
- human melanoma cell line 3:852
- human metabolome database (HMDb) 2:175, 2:401, 2:403–404, 2:420, 2:431, 2:635, 2:798, 2:1047, 2:1047–1048, 3:470, 3:471, 3:478
- Human Microbiome Project (HMP) 3:201, 3:478
- human microbiomes 3:17
- human oral microbiome database (HOMD) 2:343–344
- human oral taxon (HOT) 2:343
- human papillomavirus (HPV) 3:39, 3:41–42, 3:108–109
- Human Papillomavirus T cell Antigen Database (HPVdb) 3:40
- human phenotype ontology (HPO) 1:700, 1:701–702, 1:706, 1:840–841, 1:899, 1:908, 1:1104
 - access mode 1:705–706
 - applications of annotations 1:703–704
 - computable definitions 1:704
 - coverage 1:705
 - development 1:701
 - mapping with other ontologies 1:705
 - organization 1:702
 - phenotype annotation data 1:702–704
 - quantitative aspects 1:704–705
 - workflow and resources 1:705
- human platelet antigens (HPA) 2:938
- human primary monocytes, establishment of reference gene catalog of 3:96–99
- Human Protein Reference Database (HPRD) 1:1105, 2:109, 2:851, 3:1–2, 3:887
- human proteome coverage 1:1077–1078
- Human Proteome Organisation Proteomics Standards Initiative–Molecular Interactions (HUPOPSI-MI) 2:853
- Human Proteome Organization (HUPO) 1:130, 1:139–140, 1:1064, 2:1071, 3:866, 3:920
 - Mzidentml 1:140
 - mzML 1:140
 - PSI-MI XML 1:139–140
- Human Proteome Organization–Proteomics Standards Initiative (HUPO-PSI) 1:988–991
- Human Proteome Project (HPP) 3:919
- human RNAseq 3:850
- Human Splicing Finder (HSF) 2:228
- human tissue samples 3:914
- human transcriptome databases 2:341–343
- Human Variome Project 3:388
- human-bacterial protein-protein interactions 3:417
- human-parasites protein-protein interactions 3:417–418
- human-virus protein-protein interactions 3:415–417

- HumanCyc database 1:1052
humanity's war 3:103
Hungarian algorithm 1:883, 2:818
Hungarian-algorithm-based GRAPh ALigner (H-GRAAL) 1:1004
hybrid approach-based algorithms 1:474
hybrid assembly 3:304
hybrid cloud 1:238–239, 1:258
hybrid evolutionary algorithm (HEA) 3:4
hybrid model 1:1053–1054, 1:1054, 2:161–162
hybrid parallel computers cluster 1:162
hybrid pathway databases 1:1074–1075
hybrid techniques 1:881, 2:466
hybridization process 2:568
hydrogen (H₂) 3:939
hydrogen bond (H-bond) 2:567–568, 3:279, 3:280F, 3:546
hydrogen/deuterium exchange mass spectrometry (HDX-MS) 3:666
hydrogenase 2 maturation endopeptidase (hybD) 3:837, 3:838F
hydropathy 2:522
hydrophilic interaction liquid chromatography (HILIC) 3:466, 3:472
hydrophilic residues 3:332
hydrophobic interactions 2:839
hydrophobic residues 3:332
hydrophobic-hydrophilic model (HP model) 1:563
hydrophobicity 2:30–31, 2:447–449
scales 2:48
hydroxyl group (-OH) 3:535–536
hydroxyl hydroquinone (HHQ) 2:1094, 2:1096F
5-hydroxymethylcytosine (5hmC) 3:341, 3:341–342
5-hydroxymethyluracil (5hmU) 3:341
“HyperBalls” theory 2:525
hyperelastic constitutive law 2:903
hypergeometric tests 3:222
hypergraph 1:958, 1:1086, 1:1086F, 1:1087
hypertext transfer protocol (HTTP) 1:167–169, 2:1063
hyperthermophiles 3:945
Hyphenated MS-based approaches 2:434
hypoglycemia 1:704
hypotheses 2:1033
hypothesis testing 1:663–664, 1:691, 1:691–693, 1:694, 2:750–753
Benjamini and Hochberg correction 1:695–696
Bonferroni correction 1:695
errors in 1:693T
graphical representation of acceptance and rejection regions 1:692F
large-scale hypothesis testing 1:696
multiple hypothesis testing 1:695
procedures and error types 1:696
reporting results and misinterpretation of p-values 1:694–695
systems/applications 1:693–694
testing procedure step-by-step 1:694
hypothesis-driven modeling approaches 3:66
hypothetical genes conserved in Burkholderiaceae family 3:957–959
hypothetical protein 3:620, 3:625–627
Hypothetical Taxonomic Units (HTUs) 2:728B
- I**
- I-TASSER server 2:544–545
i2b2 tool 2:1053
I47V mutation effect 3:710
molecular mechanism of drug resistance 3:711
protein WT *vs.* mutant
comparing apo form 3:710
comparing complexed form 3:710–711
I50V and I50L/A71V mutations effect 3:711–714
IBD calculation method 3:368
IBM Bluemix 1:240, 1:242–243
IBM Bluemix Virtual servers 1:240
IBM Containers 1:243
IBM Eclipse Tools for Bluemix 1:247
IBNAL method 1:1005
ICGC Data Portal (DCC) 2:1079
iCluster method 3:197
ICM Web 2:312T–313
IDBA-UD 3:207
IDE Platforms 1:247
Identifiers (ID) 1:823, 2:10–11, 2:35
conversion and naming 1:1078–1079
identifying nucleosomal or linker sequences from physicochemical properties (iNuc-PhysChem) 2:312T–313
Identity by Descent (IBD) 2:189, 2:247
based method 3:368
matrix 3:366
identity link function 1:732
Identity-by-Missingness matrix (IBM matrix) 3:366
iDoComp algorithm 1:484
if-then rules 1:388
iFoldRNA 3:596
IgBLAST 2:943
IgDiscover 2:946
IGoR package 2:946
IgSCUEAL package 2:946
iHMMune-align 2:946
illumina
GA/HiSeq system 1:129
platforms 3:453–454
technology 3:293
Illumina HiSeq method 2:180
Illumina Infinium BeadChip 3:355
Illumina Nextera Mate-Pair protocol 3:305
Illumina Truseq Long-Read DNA 3:436
IMAAAGINE server 3:622
image
acquisition and preprocessing 2:994–996, 2:994F
analysis 2:1028
data 2:1011
clustering 1:443
digital libraries 2:1023
enhancement 2:994, 2:996F
processing 1:443, 2:1063–1064
registration 3:128
and annotation 2:1006–1007
segmentation 2:996–998, 2:1028, 3:128
classical segmentation methods 3:128
clustering methods 3:128–129
image data resource (IDR) 2:1020
image registration 3:128
imatinib 3:744T–748
imatinib mesylate (Gleevec) 2:585
imbalanced classification, application to 1:688
IMGT/Junction Analysis (IMGT/JA) 2:943–946
ImmersaDeskTM 3:129
immersive visualization systems 3:129
immobilised pH gradient isoelectric focussing (IPG-IEF) 3:892
application 3:897–898, 3:899T–901
to identifying differentially-expressed proteins
application of IPG-IEF 3:897–898
case studies 3:898–903, 3:902F, 3:904F
cellular behaviours 3:905
gene abbreviations 3:907
PTMs 3:906
immobilised pH gradient isoelectric focussing (IPG-IEF) 3:892
immobilized metal affinity chromatography (IMAC) 2:1072
immune cells 2:985, 2:986
immune control of invader 3:106
immune epitope
information extraction
in silico methods for epitope identification 3:40–42
in vitro methods for identification 3:39
mining for 3:35
immune epitope database (IEDB) 2:931, 2:931–932, 2:932F, 2:954, 3:35, 3:40, 3:244, 3:244–246
immune genetic algorithm (IGA) 1:97
immune system (IS) 2:906, 2:931, 2:984, 2:984F, 3:94, 3:652–653, 3:825
immune tolerance 2:907
Immuno Polymorphism Database (IPD) 2:935–938, 2:937F
IPD-ESTDAB 2:938
IPD-HPA 2:938
IPD-KIR 2:937
IPD-MHC 2:937
Immunochip 3:389
immunodepletion 3:903
immunofluorescent assay (IFA) 3:10
immunogenetics modelling approaches
antibody modelling 2:909–910
prediction of T- and B-cell epitopes 2:910
immunogenicity of epitopes 2:953–954
immunoglobulin kappa gene (IGK gene) 2:940
immunoglobulin lambda light chain gene (IGL gene) 2:940
immunoglobulins (Ig)
clonotype and ontogeny inference
B cell maturation 2:972F

- immunoglobulins (Ig) (*continued*)
 - lineage tree analysis & lessons learned 2:978
 - mutation analyses relying on lineage tree structure 2:979–980
 - repertoire analyses 2:977
 - sequencing approaches 2:973–974
- gene diversification 2:940–942
- gene sequence analysis 2:943–946
- IgG 3:730
- repertoires 2:973
 - comparison 2:947
 - sequence data pre-processing 2:974–976
 - sequencing 2:943
- immunoinformatics 3:241–242, 3:244–246
 - databases 2:931, 2:931–932
 - epitope 2:931
 - immune system 2:931
- immunological databases 2:922, 3:242
- immunological hot-spots 3:242
- immunology, modelling signalling pathways
 - in 2:920
- immunome 3:241–242
- immunomodulatory diseases 3:54
- immunoprecipitation (IP) 3:77
 - control 3:76
 - methods 2:332–333
- implication rules 1:362
- implicit DSLs 2:1153
- implicit frameworks 2:1153
- Implicit hydrogen* model 2:560
- “imprecision” medicine 3:379
- imprinting control regions (ICRs) 3:348
- Improved Gene Expression Programming (IGEP) 2:665
- imputation 1:465, 3:382
- IMPUTE* 3:382
- IMSEQ’s clone building 2:947
- in silico*
 - approaches for target validation 2:1094–1096
 - chemistry
 - chemical syntheses and retro-syntheses 2:609
 - development of product to reagents strategy 2:609–610
 - genotype imputation 3:388–389
 - homology modeling 2:585
 - identification of novel inhibitors 3:761
 - elucidation of potential inhibitors on DENV protein targets 3:761–763
 - imputation 3:390–392
 - methods 3:40–42, 3:44, 3:274
 - to enhance stability of TEV protease 3:636
 - protein engineering methods 3:638
 - structure-based lead optimization 3:753
 - studies 2:822
 - target identification 2:1093–1094
 - drugs databases and drug target databases 2:1093–1094
- in silico* functionalities
 - coding chemical information into molecular descriptors 2:605
 - coding chemical reactions 2:605–606
 - computer generated chemical names 2:606
 - computer-processable molecular codes 2:605
 - modeling 3D-structures 2:607–608
 - molecular editors 2:605
 - molecular graphics 2:608–609
 - molecular modeling 2:606–607
 - organizing chemical facts into databases 2:606
 - structure and substructure searches 2:608
 - 2D and 3D structure generators 2:607
- in silico* prediction
 - of PHIs 3:936
 - of PPIs for PPI network analysis 3:286–287, 3:287T
 - domain-based methods 3:287–288
 - function annotation-based methods 3:288
 - gene fusion-based methods 3:288
 - gene neighbourhood-based methods 3:288
 - interologue-based methods 3:286–287
 - machine learning-based methods 3:289
 - phylogenetic similarity-based methods 3:288
 - text mining-based methods 3:288–289
- in vacuo* methods 3:525
- in vitro*
 - approaches for target validation 2:1094–1096
 - assays 3:382
 - binding experiments 2:148
- in vivo*
 - approaches for target validation 2:1094–1096
 - assays 3:382
 - chemical probing of RNA 3:578
 - method 2:805
 - stability 3:631
- in vivo* click selective 2'-hydroxyl acylation and profiling experiment (icSHAPE) 2:580
- in-silico* prediction method 3:572
- In-silico Rational Design of Proteins (iRDP) 3:647
- iNaturalist network 2:1113
- InCa-SiteFinder 3:669
- incremental in-mapper combining 1:226
- indel realignment. *see* local realignment
- INDELible 2:754
- independent component analysis (ICA) 1:282, 1:480, 1:480–481, 1:500, 2:429–430
- independent ensembles 1:460
- independent two-samples T-test 1:700–701
- indexed retention time (iRT) 2:91
- indexes 1:83, 2:1102
 - index-based aligners 1:1005
 - structure 2:1102F
 - structures 2:1102
- indexing method 3:437
- indinavir 3:744T–748
- indirect readout. *see* shape readout
- individual equivalence constraints 1:794
- individual inequality constraints 1:795
- individual nucleotide resolution cross-linking immunoprecipitation sequencing (i-CLIP-seq) 2:333
- induced conserved structure (ICS) 1:998
- induced mutations 2:122–123
- induced pluripotent stem cells (iPSC) 2:327
- induced subgraph 1:923F, 2:815, 2:816F
- inductive approach 1:6
- infection stages 3:104–105
 - evading host system 3:105–106
 - immune control of invader 3:106
 - invading host system 3:105
 - proliferating in host system 3:106
- infectious diseases 3:932
- vaccine on 3:246–247
- inference
 - methods 2:157–158, 2:159
 - of secondary structure elements from 3D protein structures 2:489–490
 - DSSP 2:490
 - STRIDE 2:490
- inference of peptidofoms (IPF) 2:88
- inferential statistics 1:648, 1:654–657
 - continuous probability distributions 1:654–657
 - estimation 1:657–659
 - hypothesis testing 1:663–664
 - non-parametric statistics 1:664–666
- Inferred from Biological Aspect of Ancestor (IBA) 2:3–4
- inferred from curator code (IC code) 2:3–4
- inferred from electronic annotation (IEA) 1:826, 1:867
- inferred from physical interaction (IPI) 1:835
- inferred GRNs 2:806
- inferred molecular networks, assessment and validation of 2:807
- inferring rooting methods 2:733
- inflammatory bowel disease (IBD) 2:104, 3:17, 3:785
- influenza anti-viral drug 3:275–276
- Influenza Research Database (IRD) 3:108–109, 3:244
- Influenza virus knowledge-base (FluKB) 3:40
- Informatics for Integrating Biology and Bedside (i2b2) 2:1105
- information
 - flow analysis 2:807
 - information-driven approaches 1:592, 1:593–594, 1:596T
 - layer 1:587
 - linkage approach 1:1092, 1:1092–1093
 - mining 2:1083
 - sharing 1:774
 - theoretic approaches 1:879
- information and communication technologies (ICT) 1:1092, 3:67
- information content (IC) 1:871–872, 1:879, 1:889, 1:900, 1:908–909
 - graph represents metals classification example 1:872F
- information extraction (IE) 1:602, 2:1085, 2:1099–1101, 2:1099, 2:1101
- text classification and 2:1101–1102

- information retrieval (IR) 1:498, 1:561, 1:578, 1:578–579, 1:579T, 1:588–589, 1:589, 1:591–592, 1:877, 1:1104, 2:1085, 2:1099, 2:1104, 3:997–998
 algorithms 1:1106–1107
 data in life science 1:1104–1105, 1:1106F
 techniques and methods 1:1105–1106
 Informed-Proteomics 2:88
 Infrastructure as a Service (IaaS) 1:160–161, 1:163, 1:236, 1:236–237, 1:240, 1:241, 1:249, 1:257, 1:258
 Infrastructure for Systems Biology Europe (ISBE) 2:885
 Inh Dap 3:737, 3:738F
 inhibitory activity of peptide-based inhibitor 3:736–737, 3:737T
 inhibitory module (IM) 2:144
 initial ADAPTABLE phenotype 1:155–156, 1:156F
 initialization 1:641
 innate immune system 2:906, 2:931
 InnateDB 2:852
 inner cell mass (ICM) 3:827
 inner ear diseases 2:343
 inparalogs 2:260
 input space 1:503
 input split strategies 1:226
 insertion and deletion variants (indels) 3:168
 insertions or deletions (InDels) 2:359
 Institute for Systems Biology (ISB) 2:79, 2:1069, 3:858
 insulators 2:294
 insulin-like growth factor-I (IGF-I) 3:898–902
 int data type 1:206
 IntAct 2:850, 2:1104
 integer numbers, CTA for 3:817
 integer programming 2:581
 IntegraMiR 1:1101–1102
 Integrated Biodiversity Information System (IBIS) 2:1113
 integrated databases 2:103
 integrated development environments (IDEs) 1:247, 2:1156
 eclipse 1:247
 IntelliJ 1:248
 visual studio 1:247–248
 Integrated Functional inference of SN *vs.* in human (IFish) 3:4
 integrated genome annotation pipeline 3:153F
 Integrated genome browser (IGB) 2:321–322
 Integrated Genome Viewer (IGV) 2:321–322, 3:161, 3:300
 integrated haplotype score (iHS) 3:371
 Integrated Interaction Database (IID) 2:852
 integrated MS approaches 3:196
 integrated omics data analysis 3:197
 integrated pathway databases 1:1074–1075
 Integrated Taxonomic Information System (ITIS) 2:103
 integrated taxonomic methods 2:1125
 integrating Data for Analysis, “anonymization” and SHaring (iDASH) 1:255
 integrating multiple PPI databases and predicted PPIs 1:991
 integrative analysis of multi-omics data
 high throughput omics datatypes and methodologies 3:194–195
 integrative analysis of high throughput omics data 3:196–197
 integrative and machine learning based methods 2:12–13
 integrative bioinformatics.
see also bioinformatic(s)
 data integration approaches 1:1092–1093
 ontology-based data integration 1:1096
 statistical and network-based data integration 1:1096–1097
 of transcriptome 1:1099
 interaction databases 1:1099–1100
 mRNA, miRNA, and transcription factor interactions 1:1099
 tools and pipelines 1:1100
 integrative genomics viewer (IGV) 2:197–198, 2:335, 2:1080
 integrative techniques 2:466
 integrators 2:552
 intelligent agents (IA) 1:309–310;
see also artificial intelligence (AI), ABMS 1:317–318
 actions and interactions 1:311–312
 situated (inter)action 1:312
 social (inter)action 1:312–313
 agent-oriented software engineering 1:318–319
 agents in bioinformatics and computational biology 1:313
 environment 1:310–311
 guidelines/perspectives 1:319
 multi-agent systems 1:315–317
 intelligent machine 1:287
 intelligent platforms 3:72
 Intelligent System for Analysis, Model Building And Rational Design (ISAMBARD) 3:647
 IntelliJ 1:248
 intentional class 1:148, 1:796
 inter-annotator agreement score (IAA score) 1:570
 inter-cloud 1:239
 inter-object parallelism 1:218
 inter-process communication and synchronization 1:215
 inter-residue contacts 2:447
 interaction correctness (IC) 1:999
 interaction databases 1:1099–1100
 Interaction Fingerprint Triplets (TIFP) 2:653
 Interaction Network Fidelity (INF) 3:596
 Interaction Web DataBase 2:1134
 interactorList 1:1064
 Interactive Development Environment (IDE) 1:177–178
 interactive Pathways Explorer (iPath) 1:1051
 interactome 2:822, 3:1
 Interactome3D 2:852
 interactomics 2:1071–1072
 interactorList 1:1064
 intercalated motifs (i-Motifs) 3:508
 intercept-only model 1:728
 interconnected clouds 1:238–239
 interestingness measures 1:1067
 interestingness measures and criteria 1:370
 interface conservation (IC) 3:681
 interferon (IFN) 2:986
 IFN- γ 2:931
 modeling antiviral effect of 2:988–989
 Intergenic lncRNAs (lincRNAs) 2:167–168
 intergenic regions, comparative genomics of 2:262–263
 interlaced FASTQ file 3:295
 interleukins 2:920
 IL-1a 3:842
 IL-4 2:907, 2:931
 IL-12 2:907
 internal indices 1:451–452
 internal transcribed spacer (ITS) 3:184, 3:185, 3:188, 3:204
 internal validation 1:1043–1044, 1:1044T
 validation metrics for 2:667
 cross validation 2:667
 least square fitting 2:667
 LOO cross validation 2:667
 LSO cross validation 2:667–668
 true Q^2 and r_m^2 metrics 2:668
 χ^2 and RMSE 2:667
 International Cancer Genome Consortium (ICGC) 2:123–124, 2:179, 2:860–861, 2:1079, 3:457
 International Chemical Identifier (InChI) 2:635–636
 International Classification of Disease (ICD) 1:861, 1:1104
 ICD-9 1:838
 International Clinical Trials Registry Platform (ICTRP) 3:996
 International Genome Sample Resource (IGSR) 2:252–253, 3:138
 International HapMap Project 2:178–179, 2:235–236
 International Human Epigenome Consortium (IHEC) 3:341, 3:357–358
 International ImMunoGeneTics information system (IMGT) 2:932–933, 2:933F, 2:965
 IMGT/3Dstructure-DB 2:933
 IMGT/CLL-DB 2:933
 IMGT/GENE-DB 2:933
 IMGT/LIGM-DB 2:932
 IMGT/mAB-DB 2:933
 IMGT/MH-DB 2:933
 IMGT/PRIMER-DB 2:933
 IMGT/V-QUEST 2:943
 International Molecular Exchange (IMEx) 1:988, 1:1064, 2:829, 2:836, 2:853–854
 International Mouse Phenotyping Consortium (IMPC) 1:898
 International Mouse Strains Resource (IMSR) 2:103
 International Non-proprietary Names (INNs) 2:635–636

- International Nucleotide Sequence Database Collaboration (INSDC) 1:112, 2:80, 3:30
- International Union of Biochemistry and Molecular Biology (IUBMB) 1:99
- International Union of Pure and Applied Chemistry (IUPAC) 2:677
- Internet of Things (IoT) 1:243, 3:67
- internet resources for chemistry and chemoinformatics 2:614
- interolog mapping 2:826
- interologs 3:412
- interologue-based methods 3:286–287
- interoperability 1:861
- interpolated genetic maps for shared IBD segment analysis 2:247
- interpolated Markov model (IMM) 3:307
- interpretation 3:347
and applicability domain analysis 2:668–669
of genetic maps 2:243
- interpreter directive 1:178
- InterPro 2:40–42, 2:40T, 2:41F
- interProphet (iProphet) 3:864
- interquartile range (IQR) 1:651–653, 1:676
- interspersed repeats 2:210
- intra-cellular cytokine staining 3:40
- intra-epidemic evolutionary dynamics 3:976–977
- intra-network probabilistic consistency 1:1010
- intra-object parallelism 1:218
- intracellular loops (ICLs) 3:266
ICL3 3:266
- intracrine 1:917
- intrinsic
DNA sequence affinity of histone octamer 2:310
information content 1:880
noise 1:747
- intrinsically disordered proteins (IDPs) 2:28, 2:31, 3:113, 3:606
- Intrinsically Disordered Regions (IDRs) 3:1
- Introduction, methods, results and discussion (IMRAD) 3:997
- intron-centered splicing score (ICSS) 2:225
- introns 2:221, 3:967
intron-encoded proteins 3:968
retention 3:968
- intronsare 3:306
- invader
assay 3:433
immune control of 3:106
- invading host system 3:105
- invariant repeats 3:966
- invasive alien species 2:1118
- inverse co-expression 2:332
- Inverse Document Frequency 2:1103
- inverse functional constraints of properties 1:795–796
- ion channel-linked receptors 1:1057–1058
- ion ladders 2:1067
- ion mobility (IM) 2:1064
- Ion PGM 3:435–436
- Ion Proton 3:294
- Ion Torrent 3:293–294, 3:435
platform 3:180
sequencing technology 3:299
- ion-sensitive field-effect transistor (ISFET) 3:435
- ionic liquids (ILs) 2:670
- ionizable residues 2:29–30
- IonTorrent's personal genome sequencers 2:330
- iPA vs. pathway 1:1051
- "iprob" 3:864
- iRefWeb 2:852
- iron 3:106
- iron-sulfur world 3:941
- IS A relation 1:813
- ISMBLab_PCA 3:673–674
- isobaric tags for relative and absolute quantification (iTRAQ) 2:1070
- Isocitrate lyase (ICL) 3:661, 3:662F
- isoelectric point (pI) 2:29–30, 3:892
- isoflavones 3:497–498
- isoflavonoids 3:497–498
- isoform graph 1:938
- isoforms, functional analysis of genes and 2:187–188
- Isolation Forest (iForest) 1:458
- Isolation Tree (iTree). *see* data-induced tree
- isolation-based outliers 1:458–459
- ISOmestic feature MAPPING (ISOMAP) 1:486, 1:490–491
3D nonlinear swirl roll data 1:492F
- isoniazid (INH) 3:652–653
- isonicotinic acyl-NADH (INADH) 3:660
- IsoRank algorithms 1:1001
- IsoRank aligners 1:1001–1002, 1:1002
- IsoRank-Nibble. *see* IsoRankN
- IsoRankN 1:1002–1003
- isothermal-isobaric ensemble (NPT) 2:550
- isotope-coded affinity tags (ICAT) 2:1070
- isotopically nonstationary metabolic flux analysis 2:807
- isozymes 3:966
- iterated LLSimpute (ILLSimpute) 1:473
- iterative algorithms 1:2–3
- iterative correction and eigenvector decomposition (ICE) 2:182
- Iterative Dichotomiser 3 (ID3) 1:385–387, 1:387
- Iterative Protein Redesign and Optimization (IPRO) 3:631, 3:647
- IUPHAR/BPS guide to pharmacology 2:633–634
- ## J
- Jaccard distance 1:439
- Jaccard index 1:445
- Jaccard measure 1:874, 1:981
- jackknife method 1:544
- Japan Proteome Standard (jPOST) 2:80
- Japanese Genotype-phenotype Archive (JGA) 3:30
- Jarnac tool 1:1058
- JASPAR database 2:120, 2:135–140, 3:448
- Java Agent DEvelopment framework (JADE) 1:318
- Java Collection Framework (JCF) 1:207
- Java programming language 1:206, 1:216;
see also R software programming language
- BioJava 1:208
- concurrency in 1:207–208
- creation and running of program 1:207
- data structures in Java 1:207
- data types in Java 1:206–207
- structure of program in Java 1:206
- web programming in Java 1:207
- Java Server Pages (JSP) 1:207
- Java Virtual Machine (JVM) 1:206, 1:207, 3:137
- Java Web Start (JWS) 3:222
- Java/Groovy based pipeline tool 3:137
- JavaScript (JS) 2:532–533
- JavaScript for Cloud (JS4Cloud) 1:249
- JAX Clinical Knowledge Base (JAX CKB) 2:186
- JE-2147 binding, I47V mutation effect on 3:710
- Jenalib 2:467
- Jiang and Conrad measure (J&C measure) 1:879
- Jiang andConrath's measure 1:909
- Jmol (2004) 2:531–532
- Job Description Language (JDL) 1:231
- job efficiency and latency time 1:233
- job scheduler 3:136
- JOINSOLVER 2:943–946
- joint secondary structures prediction 2:581
- JointML 2:946
- Journal of Integrative Bioinformatics (JIB) 2:884
- jPOST database (jPOSTdb) 2:80
- jPOST repository (jPOSTrepo) 2:80
- JPRED method 3:519
- JSmol (2013) 2:532–533
- JTT matrix 2:737
- Jukes-Cantor model 2:713
- JuncBASE 2:226
- junctional diversity 2:942
- Just Enough Results Model (JERM) 2:889
- juxtacrine 1:917
- ## K
- k-fold cross-validation 1:543–544, 1:543F
- k-fold random subsampling 1:543
- k-means 1:352, 1:452
algorithm 1:440
arithmetic model 3:3
clustering 1:278–279, 1:1037, 1:1039F, 2:389–390, 2:391, 2:827, 2:998, 3:470
segmentation algorithm 2:1033F
- K-Medoid algorithm 1:440–441
- k-mer content 2:355
- k-nearest neighbors (kNN) 1:413–414, 1:413F, 1:473, 1:828–829, 2:13, 2:121, 3:63, 3:120–121, 3:289
algorithm 1:475

- classifier 2:1005
 distance functions 1:413–414
 impute 1:465
k value 1:414, 1:414F
 regression 1:414, 1:415F, 2:161–162
 rule 3:129
 KAKSI algorithm 3:519
 kallisto 3:119
 Kamada-Kawai algorithm 1:1022, 1:1089–1090
 Kannan, Vempla and Vetta Algorithm (KVV Algorithm) 1:1043
 Kaplan Meier plot (KM plot) 2:1081
 Karatsuba's algorithm 1:9
 Karlin-Altschul equation 1:44
 Karlsruhe ontology (KAON2) 1:807
 Katz centrality 1:951
 Katz index 1:975
 KEGG Markup Language (KGML) 1:1027, 1:1065
 Kendall's correlation coefficient 1:709–710
 kernel algorithms 1:500–501
 kernel composition 1:499–500
 Kernel Eigenface methods 1:516
 Kernel Fisherface methods 1:516
 kernel function 1:495–496, 2:743
 kernel ICA 1:500
 kernel machines 1:495
 application-specific kernel 1:497–498
 binary classification problems 1:512–513
 distances in feature space 1:511–512
 examples of kernels 1:496–497
 kernel algorithms 1:500–501
 kernel composition 1:499–500
 methods and pattern recognition 1:516
 methods in bioinformatics 1:515–516
 norm and distance from center of mass 1:512
 regression 1:513–514
 kernel matrix. *see* Gram matrix
 kernel methods
 function 1:504
 maximum margin hyperplane 1:503
 soft margin and slack variables 1:503–504
 in SVM algorithms 1:507–508
 kernel trick 1:396, 1:397T, 1:495, 1:500
 Key Representation of Interaction in Pockets (KRIPO) 2:653
 key-value store 2:1065T
 keys-based fingerprints 2:619, 2:620F
 KH domains 3:684
 kidney diseases 2:343
 killer-cell immunoglobulin-like receptors (KIR) 2:937
 Kimura 2-parameter model 2:713
 Kinefold tool 3:324–325
 "kinemage" 2:527
 kinesin spindle protein (KSP) 2:685
 kinetic equations 1:863
 kinetic models 2:438, 2:438–439
 application 2:867–868
 metabolic network to kinetic model 2:438F
 petri nets 2:439
 simulation and applications 2:439
 Kinetic Simulation Algorithm Ontology (KiSAO) 2:888
 Kingman coalescent and Watterson estimator 2:18–21
 coalescent for varying populations 2:20–21
 kinome-wide network module for cancer pharmacogenomics (KNMPx) 2:861
 KNAPSAcK 3:470
 Core Database 3:489–490
 Metabolite Activity DB 3:495
 knowledge
 assisted approach-based algorithms 1:474
 discovery through text and future directions 2:1105–1106
 knowledge extraction 1:582, 2:101
 knowledge-or empirical-based methods 3:666–667
 presentation 1:329
 knowledge and reasoning 1:295
 applications of ASP 1:298
 logic-based methodology to KR&R 1:296–297
 semantics 1:295–296
 syntax 1:295
 tools for executing ASP programs 1:297–298
 knowledge bases (KBs) 1:287, 1:291
 approach 1:47, 1:877–878
 constraints 2:506
 energy function 2:507
 Knowledge Discovery in Databases (KDD) 1:328, 1:336, 1:337
 business understanding 1:337
 data preparation 1:337–338
 data understanding 1:337
 deployment 1:339
 evaluation 1:338–339, 1:339F
 examples in health sciences 1:339–340
 in medical and biological sciences 1:337
 modeling 1:338
 knowledge engineering (KE) 1:899–900
 Knowledge Organization Systems (KOS) 1:870–871
 knowledge representation (KR) 1:294
 knowledge representation and reasoning (KR&R) 1:294
 logic-based methodology to 1:296–297
 Knowledge-based Decision Support System (KDSS) 2:1049
 knowledge-based functions. *see* statistics-based functions
 knowledge-driven Bayesian network (KD-BN) 1:688
 Kohonen map. *see* self-organizing feature maps (SOM)
 Kohonen SOM 3:3
 Kolmogorov backward equation 1:749
 Kolmogorov-Smirnov statistic (KS) 1:555
 Konstanz Information Miner (KNIME) 1:331, 2:401, 2:1152
 Kratky-Porod model 2:292
 krona tool 3:187–188, 3:187F
 Kruskal-Wallis one-way ANOVA 1:704–705
 Kruskal's algorithm 1:940, 1:945–946, 1:946F, 1:947
 Krzanowski and Lai Index (KL Index) 1:451
 kurtosis distributions 1:653–654, 1:655F, 1:679F, 1:681
 Kyoto Encyclopedia of Genes and Genomes (KEGG) 1:152, 1:203–204, 1:477, 1:918, 1:1051, 1:1052, 1:1097, 1:1106, 2:103, 2:350, 2:797, 2:799, 2:856–857, 2:858–859, 2:1047, 3:18, 3:470, 3:478–480
 database 1:982–983, 2:534, 3:317
 pathway
 analysis 3:100, 3:100F
 databases 1:1027, 2:404, 3:444, 3:822
 Kyoto Encyclopedia of Genes and Genomes (KEGG) 2:331–332, 2:421, 2:431
- ## L
- Label Free Quantification algorithm (MaxLFQ) 2:91
 label-based quantification 3:871
 label-cleavage method for protein sequencing 3:310
 label-free methods 2:89, 2:1070
 label-free quantification 3:871–872
 labelled quantification 2:1070
 labelling method 3:196
 Laboratory Information Management Systems (LIMS) 2:397–399
lac repressor protein 2:142
 Lagrange multipliers method 1:394
 Lagrangian Graphlet-based Aligner (L-GRAAL) 1:1005
 LAMARC 2:753
lambda repressor protein 2:142
 LaMoFinder 1:99–100
 lancelets 2:263
 Lander-Green algorithm 2:246–247
 language 2:1073–1074
 language-model framework 2:1103–1104
 of phylogenetic inference 2:700–701
 lapatinib 3:744T–748
 laplace smoothing 1:408–409
 Laplacian Regularized Least Squares for lncRNA-Disease Association (LRLSLDA) 3:237
 Large Organized Chromatin K9-modifications (LOCKs) 2:126–127
 large parsimony problem 2:708
 large ribosomal subunit (LSU) 3:10, 3:185
 rRNA 3:11
 large scale ecological modeling with viruses
 clinical outcome and genetic differences 3:10
 dengue outbreaks in Singapore and Malaysia 3:11
 dengue serotype-specific differences 3:10–11
 EQA of dengue
 and chikungunya diagnostics in Asia Pacific Region (2015) 3:13–14
 diagnostics 3:11–13
 MF of dengue patient's blood 3:13
 large-scale data aggregators
 ChemSpider 2:631
 PubChem 2:629–631
 large-scale enumeration 2:441–442

- large-scale human proteome studies 2:836
large-scale hypothesis testing 1:696
largest common connected sub-component (LCCS) 1:999
Las Vegas algorithm 1:10
laser UV cross-linking (LUCK) 2:824–825
Lasso regression 1:723, 1:728–729
Last Common Ancestor (LCA) 2:728B
last glacial maximum (LGM) 2:1135
Last Observation Carried Forward (LOCF) 1:465
last universal common ancestor (LUCA) 3:944
Late-Onset Alzheimer's Disease (LOAD) 2:179
latent Dirichlet allocation (LDA) 1:594–595, 1:828–829
latent semantic analysis (LSA) 1:594–595, 1:881–882, 1:882
latent variables (LV) 2:405
lateral gene transfer (LGT) 2:737, 3:970
Layered Multi-Granule Knowledge Model 1:586, 1:586–588, 1:587F, 1:588F
layers number 1:632
layout algorithms 1:1018–1019
Ld. pairs 1:764
LDetect 1:764
LDheatmap 1:764
LDHub 1:763
Ldlink 1:764
LDx 1:764
Leacock and Chodorow measure (L&C measure) 1:878–879
lead compound optimization 3:274–275
learning
 algorithm 1:329
 approaches 3:335–337
 learning-based detection methods 3:413, 3:414T
 problem 1:759–760
 rate 1:390
 rule 1:613
Least Absolute Shrinkage and Selection Operator (LASSO) 1:486, 3:976
least square (LS)
 approach 1:724
 estimation 1:427
 fitting 2:667
 impute 1:465, 1:473
 optimization approach 3:367
least-angle regression (LARS) 1:728–729
leave one pathogen protein out (LOPO) 3:419
leave-group-out (LGO) 2:667
leave-many-out cross validation (LMO cross validation) 2:667–668
leave-one-out bootstrap estimation 1:771–772
leave-one-out cross-validation (LOOCV) 1:544, 1:544F, 2:667, 2:667–668, 2:955, 3:667
 jackknife method 1:544
lectins 3:612–613
LeishDB database 2:346
Leishmania braziliensis 2:346
lemmatization 1:562, 1:577
Lennard-Jones potential (LJ) 1:53
 potential force 2:551
leptokurtic distribution 1:653–654
leptokurtotic distribution 1:653–654
level-set contour-based segmentation 2:1036, 2:1039F
Levenberg-Marquardt method 1:422
Levenshtein distance 1:23
Levitt-Gerstein score (LG-score) 3:592
Lexical Variant Generation (LVG) 1:820
lexicon-based approaches 1:579
Library of Congress 2:1083
Library of Pharmacologically Active Compounds (LOPAC) 3:770–772
library preparation 2:974
library-or signature-based methods 2:214
licorice model 2:524
Lie-Markov models 2:714
ligand 1:77
 based design 2:613
 from database preparation 2:690
 ligand-based approach 2:620, 2:621F
 preparation 3:276–277
 profiling 2:684
ligand binding domain (LBD) 2:558–559, 3:5
ligand efficiency (LE) 2:603
ligand excluded surface (LES) 2:524, 2:525
ligand-based drug designing (LBDD) 2:677, 3:280, 3:753
 LBVS 3:753
 molecular descriptors 3:753
 pharmacophore modelling 3:753–755
 QSAR 3:753
ligand-based pharmacophore modeling 2:679–680
ligand-based virtual screening (LBVS) 2:688, 3:753
ligand-binding site
 for different substrates and application as biosensors 3:636–638
 in MTB targets 3:657–658
LigAsite 2:1048
ligation 2D mapping 2:301–302
ligation of interacting RNA and high-throughput sequencing (LIGR-seq) 2:580
LIGSITEcsc algorithm 2:588
likelihood ratio of test 1:689
likelihood-based support values 2:739
Lin measure 1:879–880
LinCmb 1:474
lineage tree analysis and lessons learned
 from tree structure 2:978
 aging 2:978
 autoimmune diseases 2:979
 construction 2:978
 infection and response dynamics 2:978–979
 SHM and class switching 2:978
lineages coalescing 2:705
linear B-cell epitopes 2:916–917, 2:917T, 2:918T–919
linear biomarker association network (LN) 3:62
linear correlation coefficient. *see* Pearson product-moment correlation coefficient (PPMCC)
linear discriminant analysis (LDA). *see* Fisher Discriminant Analysis
linear epitope prediction approaches 2:965
linear free energy relationship model 2:661–662
linear interaction energy (LIE) 3:3, 3:659
linear mixed models 2:237
linear model (LM) 1:199, 1:731, 1:1053–1054
 Newton-Raphson algorithm 1:427
linear predictor 1:732
linear programming algorithms 2:440
linear regression 2:827
 analysis 1:716–717
 methods 1:722–723, 3:847–848
 prediction 1:420
 goodness of model 1:422–423
 linear regression predictor 1:420–421
 regression prediction methods work 1:420
 predictor 1:420–421
 multiple linear regression 1:421
 polynomial regression 1:421
linear steric energy relationships model 2:661–662
linear transformation 1:480
linearly inseparable data
 kernel trick 1:396
 soft margin 1:395–396
linearly separable data 1:393–395, 1:394F
Linguistic Data Consortium (LDC) 1:562
link
 clustering 1:97–98
 function 1:723
 prediction 1:973–975, 1:974F
 similarity 1:97
linkage disequilibrium (LD) 1:763, 2:189–190, 2:205, 2:235–236, 3:364–365, 3:371–372, 3:381, 3:390–392
 block 3:441
 identification 3:442–443
 coefficient 1:763
 tools for LD estimation 1:763–764
linkage equilibrium (LE) 1:763
linkages 3:525
 analysis 2:235
 limitations 2:246–247
 for mapping Mendelian disease genes 2:244
 linked data 2:1063, 2:1063–1064, 2:1069
 programs 2:244–246
 simulation and genotyping prior to analysis 2:244
linker histones, interaction of nucleosomes with 2:311
Linnaean taxonomy 2:1124
LINUS 1:65
Linux Shell 3:136
LIPED 2:245, 2:246
lipid metabolites and pathways strategy (LIPID-MAPS) 3:478–480
lipid world 3:941
LipidMaps Structure database (LMSD) 2:401

- lipids
 - and functions 2:894
 - identification and quantification of 2:896
- lipopolysaccharide (LPS) 2:906
- liposomes 3:940
- liquid chromatography (LC) 2:84, 2:396, 2:412, 2:894–895, 2:1064, 3:311–312, 3:464
- liquid chromatography-mass spectrometry (LC-MS) 1:474, 2:396, 2:399, 2:412, 2:413F, 2:418F, 2:428, 2:429F, 2:827, 3:478
- metabolomics 3:466–467
- liquid chromatography-tandem mass spectrometer (LC-MS/MS) 3:311–312, 3:918
- list syntax 1:201–202
- lists 1:196
- Lithium (Li) measure 1:881
- living cells 3:489, 3:891
- living organisms 3:74
- Illumina ExomeChip 3:389
- lncRNA-disease association prediction (LDAP) 3:237
- lncRNA-seq analysis 3:800
- lncRNAdb 2:167
- lncRNADisease 2:1051
- load balancing 1:243
- Load Sharing Facility (LSF) 3:136, 3:147–149
- lobsters 2:723–724
- local aligners 1:1009
 - MaWISH 1:1009
 - NetAligner 1:1009
 - NetworkBLAST 1:1009
 - PathBLAST 1:1009
- local alignment 1:23, 1:26, 1:103, 1:104–105, 1:1068, 2:270, 3:313
 - dynamic programming for 3:314
- local approach-based algorithms 1:473–474
- local cluster coefficient 1:936
- local distance difference test (LDDT) 3:593
- local docking 3:4
- local least squares impute (LLS impute) 1:465, 1:473
- local methods 2:809, 2:810
- local neighborhood density search 1:95–96
- Local Outlier Factor (LOF) 1:458
- local realignment 3:168
- local structure calling 2:320–321
 - visualization of 2D-interaction matrices of chr19 2:321F
- Local Term Frequency 2:1103
- local-as-view method (LAV method) 1:1094
- local-clock models 2:722
- Local-Global alignment (LGA) 3:592–593
- locality sensitive hashing (LSH) 1:252, 3:570
- locally weighted scatterplot smoothing (LOWESS) 2:377
- LOCUS field 1:132
- Lod scores 2:244–246
- loessPeak approach 3:76–78, 3:80F
- log-linear model 1:482
- log-log plotting 1:962–963
- log-odds
 - ratio 1:41, 1:760–761
- scores 1:28
 - statistics 3:333
- log2 transformation 1:467
- logarithmic loss (logloss) 1:551
- Logic and Heuristics Applied to Synthetic Analysis (LHASA) 2:611
- logic and non-monotonic reasoning 1:287–288
- logic programming 1:288
- logic-based methodology to KR&R 1:296–297
- logic-based modeling 2:866
- logical models 1:1053–1054
- logical operators 1:181
- logistic function 1:641
- Logistic Model Tree (LMT) 3:3
- logistic regression (LR) 1:277, 1:425, 1:426F, 1:729, 2:237
 - analysis 1:744
 - classifier 1:425
 - model 1:732
 - nominal and ordinal logistic regression 1:425–426
- London dysmorphology database (LDDb) 1:705
- long branch artifacts 2:733–734
- long branch attraction (LBA) 2:740
- long error-prone read alignment 3:300–301
- long intergenic noncoding RNAs (lincRNAs) 3:47, 3:49
- long interspersed nuclear elements (LINEs) 2:126, 3:305
- long non-coding RNAs (lncRNAs) 1:924, 2:167–168, 2:188, 2:189T, 2:325, 2:342–343, 2:1081, 3:9, 3:34, 3:47, 3:94, 3:94–95, 3:96F, 3:100, 3:231–232, 3:232F, 3:236–237, 3:237F, 3:800, 3:807, 3:809–810;
 - see also* small RNA
 - biological background 2:167
 - databases/resources 2:168
 - discovery and exploration 2:168
 - lncRNA-protein interactions 3:49
- long read technology 3:815
- long short-term memory (LSTM) 1:281
- long terminal repeats (LTRs) 2:126, 2:210–211
 - “long-indel” model 2:278
- long-range haplotype test (LHR test) 3:371
- long-range order (LRO) 2:451
- long-read sequencing 3:348–349
- long-read technologies 3:390, 3:436
- long-short term memory (LSTM) 1:643
- longest common subsequence (LCS) 1:16
 - problem 1:10
- longest continuous segments (LCS) 3:593
- Longest Continuous Segments in Torsion Angle space (LCS-TA) 3:597
- loop extruding factors (LEF) 2:303
- loop extrusion model (LE model) 2:302–303, 2:302F
- loops 1:182–184
 - or missing substructure modelling 2:506
 - structure generation 1:45–47
- lopinavir 3:744T–748
- loss function 1:496
- Lotka-Volterra equations 2:1131
- Lotus japonicas* 3:428
- low abundance proteins enrichment 3:914
- low power devices 1:212
- low resolution electrical tomography (LORETA) 2:808
- low-density lipoprotein (LDL) 2:236
- lowess normalization 2:366
- lowest common ancestor (LCA) 1:890
- LRLSLDA-lncRNA Functional Similarity Calculation Model (LNCsim) 3:237
- LTRdigest* 2:214
- LTRharvest* 2:214
- Luminescence-based Mammalian Interactome (LUMIER) 2:827
- Lung Gene Expression Analysis (LGEA) 2:1051
- LymAnalyzer 2:946
- lymphocytes 2:907, 2:987
- lymphoid cell turnover 2:985
- lymphoma 1:452, 1:453T
- lysine 2:147
- lysozyme 2:906
- LysR 3:428

M

- M-COFFEE method 1:30
- M46L mutations 3:715–717
- Mac OSX machines 1:706–707
- MACH function 3:382
- machine intelligence role 3:67–68
- machine learning (ML) 1:272, 1:290–291, 1:300, 1:300–301, 1:301F, 1:328, 1:342, 1:374, 1:474, 1:495, 1:499, 1:536, 1:604, 2:150, 2:826–827, 2:913, 2:916–917, 2:1005, 2:1085, 2:1103, 3:1–2, 3:35, 3:66, 3:113, 3:481, 3:681;
 - see also* deep learning (DL):supervised learning
- application
 - in bioinformatics and systems biology 1:306
 - in water quality prediction 3:3–4
- approaches in genomics 3:120–121
- emerging paradigms 1:304–305
- feature selection and extraction in bioinformatics 1:301–302
- hierarchical clustering 2:827–828
- HMM 2:828
- k-means clustering 2:827
- linear regression 2:827
- MD simulation with 2:558
- methods in water quality modeling 3:2
- ML based methods 1:579, 2:12–13, 2:48–49, 2:146–147, 3:289, 3:667–669
- performance assessment 1:304
- and performance measure 1:273–274
- representative ML algorithms 1:302–304
- software 1:305–306
- machine learning techniques (MLTs) 1:466–467, 1:1105–1106, 2:834–835, 2:842, 2:1005, 3:1, 3:6T–7, 3:8T

- ANN 2:664–665
 GEP 2:665
 SVM 2:665
 Macroglossa visual search 2:1019–1020
 macromolecular 2:552, 3:940
 by structural modifications 3:632–634
 structure determination methods 3:722
 visualization 1:262
 Macromolecular Structure Database (MSD)
 2:460
 macrosynteny 2:263
 MACS software 3:114
 Madison-Qingdao Metabolomics Consortium Database (MMCD)
 2:403
 MafTools 2:273
 MAGE Software 2:527
 MAGE-TAB 1:138
 MAGIA2 web tool 1:1100–1101
 MAGNA algorithm 1:1006
 magnetic resonance images (MRI) 2:993,
 2:998
 magnetoencephalography (MEG) 2:808
 Maize Genetics Database (MaizeGDB)
 3:425
 major cardiovascular disease (MCDVs)
 2:1060–1061
 major facilitator superfamily (MFS) 3:927–
 928
 major histocompatibility complex (MHC)
 2:907, 2:911F, 2:914F, 2:931, 2:965,
 3:39, 3:242, 3:392
 prediction systems for MHC binding 3:42–
 44, 3:43T
 in vitro methods for identification of MHC
 ligands 3:39
 malaria 2:591
 MALDER method 3:369
 maltose-binding protein (MBP) 3:637
 mammalian organism sources 2:345–346
 Mammalian Protein-Protein Interaction Trap
 (MAPPIIT) 2:827
 Mammalian Transcriptomic Database (MTD)
 2:345
 Manhattan distance (d_M) 1:413, 1:438
 Mann-Whitney test 1:666–669, 1:704
 mannose binding proteins (MBPs) 3:670
 mannose interacting residues (MIRs) 3:670
 mannose receptors (MR) 2:906
 mannose-6-phosphate (M6P) 2:57
 manual modelling 3:267
 manual pathway mining 1:1079–1080
 manual prokaryotic gene identification
 3:306
 Many Integrated Core (MIC) 1:212
 many-body potentials function 2:561
 MAP kinase kinase (MAP2K). *see* MAP
 kinase kinase (MAPKK)
 MAP kinase kinase (MAPKK) 2:857
 MAP kinase kinase (MEK). *see* MAP kinase
 kinase (MAPKK)
 MAP kinase kinase kinase (MAP3K). *see*
 MAP kinase kinase kinase (MAPKKK)
 MAP kinase kinase kinase (MAPKKK) 2:857
 MAP kinase kinase kinase (MEKK). *see* MAP
 kinase kinase kinase (MAPKKK)
 mapping disease-related haplotype to genes
 3:444
 mapping proteins to genome 3:313
 mapping RNA interactome *in vivo* (MARIO)
 2:580
 mapping sequencing data
 alignment file formats 3:298
 goals and problems of short reads
 alignment 3:297–298
 long error-prone read alignment 3:300–301
 QC using read alignment 3:300
 short read alignment 3:298–299
 variation calling 3:299–300
 mapping structure to property in QSAR
 approach 2:613–614
 MapReduce algorithm 1:171, 1:172F, 1:258
 alignment-free sequence comparison
 1:224–226
 computational biology by Hadoop and
 spark 1:221–222
 description, analysis and assessment
 1:222–224
 marginal probabilities of secondary
 structures 2:578
 Marine Microbial Eukaryote Transcriptome
 Sequencing Project (MMETSP) 2:34
 marker gene analysis 3:204–207
 Markov chain model 1:749, 1:753–755,
 3:449–450
 Markov clustering algorithm (MCL
 algorithm) 1:96, 1:254, 1:1041,
 1:1041–1042, 2:344
 Markov decision process (MDP) 1:304–305
 Markov models 2:712–713, 3:306–307,
 3:443
 Markov networks 1:280
 Markov process 1:748, 1:748–749, 1:749,
 1:753–755
 Brownian motion 1:750
 Poisson process 1:749–750
 Markov random fields (MRF) 1:280
 Markov State Model (MSM) 2:557–558,
 2:904
 MARS Human 7 (MARS-7) 3:902
 Marshfield genetic map 2:242–243
 Marshfield Personalized Medicine Research
 Project (PMRP) 2:175
 mas5 function 3:816
 MASCOT database 2:87
 Mascot generic format (MGF) 2:77
 MASH Suite 2:88
 mass spectrometry (MS) 1:126–127, 1:137,
 1:139, 1:171–172, 1:332, 2:76, 2:84,
 2:396, 2:410, 2:410–411, 2:411F,
 2:414, 2:426, 2:428, 2:798, 2:834,
 2:895–896, 2:1064–1065, 3:39,
 3:189, 3:196, 3:311, 3:410, 3:465,
 3:478, 3:847, 3:857, 3:871, 3:892,
 3:896F, 3:911, 3:913F;
 see also nuclear magnetic resonance spec-
 troscopy (NMR spectroscopy),
 data
 files 2:414
 from proteomics analyses 2:84
 inherent role of bioinformatics in
 proteomics 2:1065–1066
 MS based metabolomics 3:465–466
 MS data preprocessing 3:467–468
 MS-analyzer 1:332–333
 MS-based protein identification 3:855
 MS-based proteomic
 data processing workflow 2:84–86
 studies 2:91
 multiple approaches exist for MS based
 protein quantification 3:872F
 PMF 1:127
 from raw data to peak lists 2:1064–1065
 reconstructing proteins from 3:311–313,
 3:312F
 sample preparation for 2:414
 shotgun proteomics 2:1066–1067
 tandem mass spectrometry 1:126–127
 workflow of MS based protein
 quantification 3:873F
 Mass Spectrometry Interactive Virtual
 Environment (MassIVE) 2:80, 2:402
 mass spectrometry strategies 3:855
 mass spectrometry TAP (MS-TAP) 2:824
 Mass spectrometry-based metabolomics
 analysis 2:421–422, 2:422
 GC-MS and LC-MS 2:412
 instrumentation for 2:410–411
 mass spectrometer data files 2:414
 separation techniques 2:411–412
 tandem mass spectrometry-MS/MS 2:412–
 414
 Mass Spectrometry-Data Independent
 AnaLysis (MS-DIAL) 2:399
 MassBank database 3:470
 MassCascade algorithm 2:401
 Massive Parallel Processing (MPP) 2:1049
 massive parallel signature sequencing 3:820
 massively parallel sequencing (MPS) 3:157,
 3:453
 massively parallel sequencing by synthesis
 (MPSS) 2:329
 MassTRIX database 3:470, 3:821
 MASTR-MS approach 2:397–399
 matches 1:15–16
 Matching-based Integrative GRAPh Aligner
 (MI-GRAAL) 1:1004
 mate-pair (MP mapping) 2:357
 Math well Correlation Coefficient (MCC)
 3:5, 3:9T
 Mathematical Markup Language (MathML)
 1:860, 1:861F
 mathematical modeling 2:801
 mathematics to chemistry and chemistry to
 mathematics 2:612
 MatNMR software system 2:402
 Matrix Assisted Laser Desorption Ionization
 (MALDI) 1:126
 matrix syntax 1:201
 matrix-assisted laser desorption ionisation
 (MALDI) 2:1064, 3:914–918
 Matrix-assisted laser/desorption ionization
 time of flight (MALDI-TOF) 1:127
 matrix-balancing approach 2:320
 MatrixDB 2:851
 MATrixLABoratory (MATLAB) 1:115, 1:331–
 332
 MATLAB 7.0 package 3:471

- MATS tools 2:226
 Matthew's correlation coefficient (MCC)
 1:550, 2:578, 3:597, 3:669
 mature fat cells 3:354
 MAVisto, network motifs 1:99
 MaWISH aligner 1:1009, 1:1068
 maximal frequent itemsets (MFI) 1:360–362,
 1:361*F*, 1:370
 maximum a posteriori class (MAP class)
 1:406
 maximum clique 1:934
 maximum common ancestor (MCA) 1:890
 maximum common subgraph (MCS) 2:640,
 2:642–644, 2:653–654
 algorithms 2:642–644
 variations 2:645
 maximum edge length minimization
 1:1088–1089
 maximum enrichment score (MES) 1:896–
 897, 3:219
 maximum entropy (ME) 2:1085
 maximum entropy Markov model (MEMM)
 1:563–564
 maximum entropy thresholding 2:1029
 maximum expected accuracy (MEA) 2:580
 predictions based on 2:578
 Maximum Informative Common Ancestor
 (MICA) 1:889
 maximum likelihood (ML) 1:723, 1:732,
 2:707–708, 2:753–754, 3:10
 estimator 2:13–14
 maximum likelihood estimation (MLE)
 1:426–427, 1:738–739, 1:759, 2:577,
 3:346*T*, 3:346–347
 gamma example 1:739
 normal example 1:739
 Poisson example 1:739
 maximum margin hyperplane 1:503, 1:504*F*
 maximum matching ratio (MMR) 1:981
 maximum parsimony 2:706–707
 maximum weighted common subgraph
 (MWCS) 2:654
 maximum weighted matching 1:981
 MaxMod database 2:451–452
 maxPeak approach 3:76–78, 3:80*F*
 MaxSnippetModel 2:947
 MaxSub program 3:592
 May's paradox 2:1131–1133
 McCaskill algorithm 2:579
 mclusterSim function 1:893
 MCODE method 1:95, 1:979
 MCQ4Structures 3:597
 MD display Software 2:528
 mean absolute error (MAE) 1:433, 1:551
 mean average precision (MAP) 1:556, 3:333
 Mean of Circular Quantities (MCQ) 3:597
 mean ratio normalization procedure (MRN
 procedure) 2:381
 mean residue ellipticity (MRE) 3:520–521
 mean squared error (MSE) 1:526, 1:551,
 1:622, 1:740–741
 Mechanically induced trapping of molecular
 interactions (MITOMI) 2:148, 2:149
 mechanism of action (MOA) 3:71
 mechanism-based modeling 2:866
 MedGen database 2:104
 median 1:675
 median rank of first positive prediction
 (MRFPP) 3:419
 mediator-based solutions 1:1092, 1:1094–
 1095, 1:1094*F*
 Medical Image File Accessing System
 (MIFAS) 1:265
 Medical Language Extraction and Encoding
 system (MedLEE system) 2:1090
 Medical Subject Headings (MeSH) 1:579,
 1:700, 1:838, 2:634, 2:1085–1086,
 2:1099
 MeDIPS package 3:344
 medium non-coding RNA 3:235;
 see also long non-coding RNA (lncRNA):
 small RNA
 snoRNA 3:235–236
 snRNA 3:236
 tRNA 3:235
 MedSim tool 1:911
 MedusaDock program 3:4
 MegaBLAST algorithm 3:295–296
 MeilPa and Shi Multicut Algorithm 1:1043
 meiosis to centiMorgans 2:242
 MeltDB database 2:399
 Membrane Protein Data Bank (MPDB)
 2:468
 MEME program 3:338, 3:448–449, 3:450*F*
 memetic algorithms 1:777
 memory cell structure 1:643*F*
 Memory Mapped Version 1 engine
 (MMAPv1 engine) 2:1072
 memory-constrained UPGMA (MC-UPGMA)
 1:278
 MemPROT package 2:468
 MemProtMD package 2:47
 Mendelian diseases 2:171, 3:116
 diagnosis 3:180–181
 rare variants discovery in Mendelian
 diseases 3:170–171
 Mendelian Inheritance in Man (MIM) 3:170
 Mental Disease Ontology (MFO) 1:1025
 Mercator 2:272–273
 Mercer conditions 1:496
 MERGE function 1:3
 merge sort 1:8
 MergeAlign method 1:30
 MERLIN software 2:246–247
 mesenchym (Mes) 2:392
 mesenchymal stem cells 3:354
 MESHI-Score 3:594
 mesokurtic distribution 1:653–654
 mesokurtotic distribution 1:653–654, 1:681
 mesoscale 1:978
 mesoscopic lattice model 2:868
 message passing interface (MPI) 1:167,
 1:200, 1:217
 messenger RNA (mRNA) 1:113, 1:127,
 1:561, 1:833, 1:1099, 2:142, 2:165,
 2:198–200, 2:201–202, 2:221, 2:324,
 2:325, 2:364, 2:372, 2:567, 2:568,
 2:575, 3:9, 3:33–34, 3:94, 3:230,
 3:234, 3:294, 3:504, 3:546, 3:792,
 3:807, 3:820
 expression data 3:849
 folding energy
 and interaction data 3:849
 prediction using protein abundance
 3:852
 mRNA-seq quantification pipeline 3:793*F*
 meta-database management system (meta-
 DBMS) 1:1093
 meta-genetics. *see* marker gene amplification
 metagenomics
 MetABEL package 3:382
 MetaboAnalyst 3.0 package 2:401
 metabolic fingerprinting 3:463
 metabolic flux analysis (MFA) 2:801, 2:807,
 2:864–865
 metabolic footprinting 3:463
 metabolic models
 elementary mode analysis 2:441
 kinetic models 2:438
 stoichiometric models 2:439–440
 metabolic network 1:915, 1:918, 1:1018,
 2:806–807, 3:476–478, 3:476*F*,
 3:481–483
 metabolic pathways 1:147, 1:1036, 1:1048,
 1:1049–1051, 1:1050*F*, 2:796, 3:316–
 317
 analysis 3:316–317
 databases 1:1051–1052, 1:1053*T*
 metabolic databases 2:404, 3:317
 and networks 3:489
 tools features comparison 1:1052*T*
 visualization and analysis tools 1:1050–
 1051
 Metabolic Phenotype Database (MetaPhen)
 2:402
 metabolic profiling 2:433, 2:433–434, 2:434
 analytical workhorses 2:427–428
 approaches in experimental and statistical
 design 2:433
 bottlenecks in 2:431–433
 phenotyping and fingerprinting 2:426–427
 statistical methods 2:428–430
 MetaboLights repository 2:402
 metabolism 1:1050, 2:438, 2:866, 3:84–85,
 3:489
 MetaboliteDetector 2:399
 metabolites 2:396, 2:410, 3:489–490
 databases 2:402–403, 2:403
 feature list 2:419
 identification 2:420–421, 3:470
 mass spectral databases 2:403
 peak picking and deconvolution 2:419
 profiling 3:463
 quantification 2:415–416, 2:416*F*
 sets 2:405–406
 target analysis 3:463
 metabolites biological role (MBROLE)
 2:405–406
 metabolome analysis 3:463, 3:463–465,
 3:464*F*, 3:470–472, 3:478–480
 data processing
 and analysis 3:467–468
 and workflow management solutions
 2:399–402
 metabolomics experiments repositories
 and metabolite databases 2:402–403
 peaks to biological activities 2:406
 sample preparation 3:463–465

- metabolome analysis (*continued*)
 statistical analysis and functional interpretation 2:404
 workflow and LIMS solutions 2:397–399, 2:401–402
- Metabolome-Wide Association Studies (MWAS) 2:433
- MetabolomeXchange 2:403
- metabolomics 2:410, 2:806, 3:194, 3:196
 in disease 3:471–472
 experimental design 2:414
 experiments repositories 2:402–403
 in microbiology 3:472
 in plant biology 3:472
 sample preparation for mass spectrometry analysis 2:414
 transcriptome to 3:820
- Metabolomics Repository Bordeaux (MeRy-B) 2:402
- Metabolomics Standards Initiative (MSI) 1:130
- Metabolomics Workbench 2:402
- metabonomics 2:427, 3:463
- MetaCyc database 1:152, 1:1052, 2:404, 2:797, 3:478–480
- metadata 2:888–889
- metadynamics simulation 2:554–555
- MetageneAnnotator 3:154
- Metagenome Human Intestinal Tract project (MetaHit project) 3:201
- metagenomes
 analysis 3:186
 assembling 3:305
- metagenomic analysis 3:19–20, 3:20F, 3:207–208
 and applications 3:184, 3:186F, 3:189
- Metagenomics-Rapid Annotation using Subsystems Technology (MG-RAST) 3:188
- metal oxide affinity chromatography (MOAC) 2:1072
- MetAlign tool 2:399
- MetaMap tool 1:820
- metamap transfer tool (MMTx) 1:868
- metamorphic proteins 1:562
- MetamorphoSys software 1:821
- MetaMQAP tool 3:595
- metaMS software 2:399
- metaproteomics 3:185–186
 analysis 3:189
- MetaRanker 1:911
- metaSPAdes 3:207
- MetastaSim model 3:247
- metastasis associated lung adenocarcinoma transcript 1 (MALAT1) 3:236, 3:844
- metatranscriptomics 3:185–186, 3:189
- Metazoa-Ensembl database 2:252
- methanol 3:905
- MethDB database 2:128–129
- methionine 3:306
- Methodologies for Optimization and Sampling in Computational Studies (MOSAICS) 2:312T–313
- methotrexate (MTX) 3:729
- MethyCancer 2:1081
- methyl-6-Adenosine (m6A) 2:328
- methylation 1:760, 2:15, 2:155, 3:327
 arrays 3:346
 callers 3:345
 sites prediction 2:19
- methylation analysis using ChAMP 3:810
- 5-methylcytosine (5mC) 3:341
- methylmalonic aciduria associated protein A (MMAA) 2:656–657
- MethylomeDB database 2:128–129
- methylSig 3:347
- methyltransferase (MTase) 3:772
- methyltransferase-specific protein methylation motifs 2:19
- METLIN Metabolite Database 2:403, 3:470, 3:478
- Metroplis method 2:556–557
- metropolis criterion simulations (MCM simulations) 3:748
- metropolis-hastings random walk (MHRW) 1:90, 1:91–92
- MG-RAST 3:206–207
- MGED Ontology (MO) 1:138–139
- mgeneSim 1:893
- mgoSim function 1:893
- MHC-Peptide Interaction Database-TR version 2 (MPID-T2) 2:934, 2:936F
- MHCBN database 2:934, 2:936F
- micellar electrokinetic chromatography (MEKC) 3:467
- Michaelis-Menten equation 2:439
- micro canonical ensemble (NVE) 2:550
- microarray 1:127–128, 1:171–172, 2:329, 2:364, 2:377–379, 3:195, 3:389–390, 3:808, 3:814, 3:820
 applications 2:121
 approach to gene expression profiling 2:375
 computational prospective for 2:121
 data 1:342
 case study on 1:452
 normalization 2:364–366
 database 2:102
 Genechip 1:128
 implications in functional genomics 2:120–121
 microarray-based technologies 3:355, 3:452
 normalization for 3:816
 spotted arrays 1:128
 technology 1:126, 1:463, 2:156–157
- Microarray and Gene Expression Data (MGED) 1:137
 ontology and ontology for biomedical investigations 1:138–139
- Microarray Data Analysis System (MIDAS) 1:334
- Microarray Data Manager (MADAM) 1:334
- microbes 3:18, 3:184, 3:945
- microbial genomes assembly 3:305
- microbial sample preparation 3:464–465
- microbiology 1:276–277
 metabolomics in 3:472
- microbiome 3:200–201
- microRNA (miRNA) 1:230, 1:1099, 2:124, 2:155, 2:165–166, 2:167, 2:188, 2:325, 2:372, 2:575, 2:704, 2:861, 2:1081, 2:1082, 3:9, 3:33–34, 3:47, 3:94, 3:105, 3:194, 3:232F, 3:234, 3:535, 3:809–810
- biological background 2:165–166
- computational approaches 2:124–125
- databases/resources 2:166–167
- discovery 2:165–166
- genomics, biogenesis and mechanism of regulatory RNA 2:124
- miRNA/mRNA associations 1:1100–1101
- plant 2:125–126
- regulation of gene expression 3:807
- resources and tools 2:124T
- targeting to various major cancer types 3:53F
- in therapeutics 2:126
- transposons in functional genomics 2:126
- microRNA database (miRBase) 1:113
- microRNA off-set RNAs (moRNA) 2:325
- microRNA sequencing (miRNA-seq). 2:125, 2:333, 3:799–800, 3:800F
- microsatellites 2:210
- microscopic Ab-initio models 2:878
 ABM 2:878–879
 cellular automata 2:878
- microscopic lattice method 2:868
- Microsoft Azure ML 1:250
- Microsoft IDE 1:247
- Microsoft Visual Studio 1:247
- Microsoft's Azure Batch 2:1155
- microsynteny 2:263
- mid-throughput technologies requiring computational analysis 3:434
- midpoint rooting 2:730F, 2:730–731, 2:733;
see also outgroup rooting
- migration 2:747, 2:748F
- Miller-Urey experiment 3:939, 3:939F
- Million Veterans Project (MVP) 2:175
- miner-allocorticoid receptors (MR) 3:946
- miniature inverted-repeat transposable elements (MITEs) 2:210–211
- minimal free energy (MFE) 3:569
- minimal information (MI) 1:130
- minimum confidence (minconf) 1:398
- minimum free energy (MFE) 3:234, 3:571
 dynamic programming for 2:579
 prediction of MFE structures 2:577–578
- Minimum Information about Genome Sequence (MIGS) 2:885
- Minimum Information about high-throughput SEQuencing Experiment (MINSEQE) 1:137, 1:138
- Minimum Information About Microarray Experiment (MIAME) 1:137, 1:137–138, 2:885
- Minimum Information About Molecular Interaction Experiment (Mimix) 2:829
- Minimum Information About Proteomics Experiment (MIAPE) 2:885, 2:1063
- Minimum Information About Proteomics Experiment (MIAPE)
- Minimum Information About Simulation Experiment (MIASE) 2:888, 2:889

- Minimum Information for Biological and Biomedical Investigations (MIBBI) 2:885, 2:889
- Minimum Information Required In Annotation of Models (MIRIAM) 1:1065, 2:885–886, 2:889
- minimum inhibitory concentration (MIC) 3:926
- minimum norm estimates (MNE) 2:808
- minimum spanning tree algorithms 1:940, 1:945–946, 1:947
- Kruskal's algorithm 1:945–946
- Prim's algorithm 1:946
- minimum spanning tree-based clustering (MST-based clustering) 1:441–442
- MinION system 3:294, 3:436
- Minkowski distance. *see* Manhattan distance (d_M)
- min–max normalization 1:467, 1:478
- minor allele frequency (MAF) 1:763–764, 2:235, 3:169, 3:388
- minor histocompatibility antigen (MiHA) 2:966
- minorallelefrequency (MAF) 3:441–442
- miR2GO webserver 2:1081
- MiRBase database 2:125, 2:166
- mirConnX 1:1101
- miRNA Prediction From small RNA-Seq data (miR-PREfer) 2:125–126
- miRNA signature (miRsig) 2:125
- miRNA-target interactions (MTIs) 3:194
- miRNA398 (miR398) 2:125
- miRNAs encode peptides (miPEPs) 3:47
- mirror tree approach. *see* phylogenetic tree-based approach
- Missing at Random (MAR) 1:464
- Missing Completely at Random (MCAR) 1:464
- Missing data Imputation incorporating Network and adduction information in Metabolomics Analysis (MINMA) 1:474
- Missing Not at Random (MNAR) 1:464
- missMethyl package 3:344
- MITAB format 1:1064
- mitochondrial DNA 2:3
- mitochondrial genes 2:724
- mitochondrial processing peptidase (MPP) 2:55
- mitochondrial targeting signal prediction 2:55
- mitogen-activated protein kinase (MAPK) 2:856, 2:856–857, 2:858F
- dynamics 3:66
- signaling 2:789
- mitotic DNA replication 2:2
- mixed graph 1:1086, 1:1086F
- mixed linear models. *see* linear mixed models
- mixed-use schedulers 2:1155
- mixture models 1:279
- mixture of factor analysers (MFA) 1:279
- mixture of isoforms (MISO) 2:226
- mmCIF format 2:569
- mobile elements 3:955–957
- ModBase 2:1013, 2:1016F
- Model based Analysis of ChIP-Exo (MACE) 3:76
- challenges with 3:78
- model based expression index (MBEI) 2:377
- Model Building Design 1:338
- Model Explorer (ME) 1:451
- Model Quality Assessment (MQA) 3:587
- model quality assessment 2:508
- programs 3:588
- of protein structures 3:588–592
- of RNA structures 3:596
- Model Quality Assessment Program (MQAP) 2:508, 3:587, 3:589T–591
- model-based approaches 1:331, 2:1004, 3:438
- model/modelling
- analysis 2:792–793
- archives 2:466
- biological systems 2:791
- calibration and tuning 2:792
- cellular immune system dynamics 2:920–922
- construction 2:791–792
- immune system 2:908–909
- model-based approaches 1:17, 1:441, 3:483–484
- model-based clustering methods 3:365, 3:372–373
- model-based functional connectivity approaches 2:808
- modelling-based methods 3:367, 3:367–368
- refinement 2:508
- signalling pathways in immunology 2:920
- 3D-structures 2:607–608
- on web 2:452
- MODELLER software 1:38, 2:451–452, 2:506, 3:588
- modern short read alignment software 3:298–299
- ModFOLD web server 3:594–595
- ModFOLD3 3:594
- ModFOLD6 3:594–595
- ModFOLDclust 3:595–596
- ModuLand 1:96
- modular enrichment analysis (MEA) 1:896, 1:897, 3:219
- modularity 1:96, 3:485
- module identification analysis 3:796–798
- molecular
- assemblies visualization 2:526–527
- data to chemical information and chemical ontology 2:602–603
- editors 2:605
- entity 2:603, 2:635
- epidemiology 2:430–431
- fingerprint and similarity searches 3:749
- force fields 1:49–53, 2:1144
- graphics 2:608–609
- imaging techniques 2:1097, 2:1097F
- mechanisms responsible for drug resistance 3:926
- repair 3:970
- simulation methods 1:261, 2:545–546
- variation 2:1052
- weight 2:28–29
- molecular biology
- experimental techniques 1:209
- research 1:1071
- techniques 1:230
- molecular clock 2:719, 2:719–720, 2:720–721, 2:722–723, 2:728B, 2:729F, 2:733
- calibrating 2:721–722
- emergence of lobsters 2:723–724
- molecular dating 2:719F
- neutral and nearly neutral theories 2:720
- in practice 2:720
- rate variation models 2:722
- molecular descriptors 3:753
- calculation 2:663, 2:663T
- coding chemical information into 2:605
- molecular docking 2:689, 2:1144, 3:274, 3:275T, 3:276, 3:277–278, 3:672–673, 3:929
- advancement towards designing computational methods 3:673
- analysis of Inh Dap 3:737
- limitations of traditional molecular docking tools 3:673
- MD simulation with 2:557
- molecular docking based VS 2:588, 2:589
- molecular dynamics (MD) 1:211, 1:262, 1:562, 2:145, 2:150, 2:526, 2:590–591, 2:607, 2:910, 2:1144, 2:1145–1146, 3:3–4, 3:260, 3:607–609, 3:708–709
- MD based methods 3:671–672
- MD-based protein structure model refinement 3:613
- trajectories visualization 2:526
- molecular dynamics flexible fitting (MDFF) 3:614
- molecular dynamics simulation (MD simulation) 1:65, 2:292, 2:529, 2:550, 2:552, 2:559–560, 2:582, 2:681, 3:286, 3:652, 3:655–657, 3:656F, 3:660–662, 3:743, 3:749
- advanced techniques 2:553–554
- coarse graining 2:560
- conformations and energy landscapes of apo-form PRLBD 2:558–559
- elucidating structural dynamics and mechanisms of MTB targets 3:655–657
- ensemble and trajectory 2:550
- evaluation of binding interactions 3:659–660
- finite element method 2:561
- force field 2:550–552, 2:560–561
- GPU accelerated 2:559–560
- integrators 2:552
- internal coordinates 2:551F
- with machine learning 2:558
- MC simulation 2:556–557
- MC-CG 2:560
- metadynamics simulation 2:554–555
- methods to overcome limitations and decrease load 2:552
- with molecular docking 2:557
- multilevel coarse graining 2:560

- molecular dynamics simulation (MD simulation) (*continued*)
 prediction of new ligand-binding site 3:657–658
 SAC6 inter-domain peptides flexibility 2:559
 software 3:613
 systems/applications 2:557
 target structure for structure-based drug design and discovery studies 3:658–659
- molecular evolution 3:938, 3:939–940
 current model for life hypothesis 3:941
 membrane defined first cell 3:940–941, 3:941F
 neutral and nearly neutral theories of 2:720
 origin of cells and organized structure 3:940
- Molecular Evolutionary Genetics Analysis (MEGA) 3:992
- molecular function (MF) 1:823, 1:867, 1:889, 2:1, 2:8–9, 3:626–627, 3:821
- molecular interaction (MI) 1:139, 1:915, 1:988–991
- Molecular INteraction Database (MINT Database) 2:1048, 3:1–2
- Molecular Interactions database (MINT database) 1:1105
- molecular labelling method 3:914–918
- Molecular Libraries Program (MLP) 2:631
- molecular lipophilic potential (MLP) 2:532
- molecular mechanics (MM) 1:63–64, 2:607
- Molecular Mechanics energies combined with Poisson–Boltzmann and Surface Area continuum solvation method (MM/PBSA method) 3:614
- Molecular Mechanics Generalised Born Surface Area (MM-GBSA) 3:659
- Molecular Mechanics Poisson–Boltzmann Surface Area (MM-PBSA) 3:659
- molecular mechanics with Poisson–Boltzmann and surface area solvation (MM-PBSA solvation) 3:710
- Molecular Mechanics–Poisson Boltzmann Surface/Generalized Born Area (MM-PBSA/GBSA) 3:3, 3:4
- Molecular Mechanics/Generalized Born Surface Area (MM/GBSA) 2:655
- molecular modelling 1:261, 2:604, 2:606–607
 cloud-based services for 1:262–263
 method 2:840T
 tasks 1:261–262
- molecular modelling database (MMDB) 2:467, 3:31
- molecular networks inference
 assessment and validation of inferred molecular networks 2:807
 GCNs 2:806
 GRNs 2:806
 metabolic networks 2:806–807
 PPI networks 2:805–806
 signaling networks 2:807
- molecular phylogenetics 2:700, 2:701, 2:710, 2:736
 aligning sequences 2:701–703, 2:702F
 assumptions in 2:703
 Bayesian methods and MCMC 2:709
 data 2:701
 motivation 2:700
 phylogenetic inference language 2:700–701
 software 2:710
- molecular recognition features (MoRFs) 3:121, 3:126
 implementation steps of MoRFs predictor 3:123–125
 prediction framework 3:122–123
 predictor design 3:122
- molecular replacement (MR) 2:514
- Molecular Sig-natures Database (MSigDB) 2:858–859
- molecular surface (MS) 2:524–525
- molecular tinkering 2:482, 3:970
- molecular visualization software (MV software) 2:522, 2:527
 applications in 2:535
 AVS 2:527–528
 Bioblender 2:532
 Cn3D 2:528–529
 Jmol 2:531–532
 JSmol 2:532–533
 MAGE 2:527
 MD display 2:528
 MOLMOL 2:528
 NGL viewer 2:533
 PyMOL 2:530, 2:531F
 RasMol 2:527
 Raster3D 2:528
 RIBBONS 2:527
 3Dmol.js 2:533
 UCSF Chimera 2:530–531
 VMD 2:529
 Web3DMol 2:533
 WebMol 2:529–530
 YASARA view 2:533
- molecules 1:1047–1048, 2:640
 graph representations of 2:641
- MolProbity 2:517, 3:586, 3:593–594, 3:597
- MolviZ.Org 2:534–535
- momentum 1:631
- Monarch Initiative 1:706, 1:898
- monocyte chemotactic protein-1 (MCP-1) 3:634
- MonoDB 2:1012–1013, 2:1015F
 MongoDB SQL Server Importer 2:1068
 performance 2:1070–1071
 storage requirement of MS SQL server *vs.* 2:1072
- monomer 2:293, 2:821
- monophyletic dengue virus type 2 population 3:10
- monophyly 2:734–735
- monosaccharide biosynthetic process 1:827
- Monosiga brevicollis* 2:262
- Monte Carlo Markov chain (MCMC) 1:749, 2:159, 2:226, 2:709, 2:738–739, 3:338, 3:977
- Monte Carlo simulation (MC simulation) 1:9, 1:451, 1:982, 2:123, 2:507, 2:556–557, 2:582, 3:525, 3:743–748
- Moran model 2:21–22
- Morus Genome Database (MorusDB) 2:347
- Morus Notabilis* 2:347
- mosaic proteins 3:966–967
- most informative common ancestors (MICA) 1:873, 1:901
- most recent common ancestor (MRCA) 2:746
- most specific common abstraction (MSOA) 1:878
- Mothur tool 3:206
- motif 1:15, 1:16, 1:95, 2:155, 3:332, 3:504
 additional collections of binding sites and 2:140
 algorithms for search of motifs in graphs 1:98
 detection 1:937, 1:938
 finding algorithms 3:448–449
 model 3:333–334
- motif-matrices (MM) 2:913
- mouse (*Mus musculus*) 2:252
 genomics 2:1104
 mouse protein kinase A 3:276
- mouse double minute 2 (MDM2) 3:5, 3:6
- mouse embryonic stem (mES) 3:580
- Mouse Genome Database (MGD) 1:823, 1:898, 3:425
- Mouse Genome Informatics (MGI) 1:825, 2:103, 2:1104
- mouse phenome database (MPD) 2:103
- mProphet algorithm 2:89, 3:880
- MrBayes program 2:709, 2:710
- MREPS program 2:214
- MS SQL server performance
 MongoDB performance *vs.*, 2:1070–1071
 Neo4j performance *vs.*, 2:1071–1072
 storage requirement
 of MS SQL server *vs.* MongoDB 2:1072
 of MS SQL server *vs.* Neo4j 2:1072–1073
- MS-Align+ 2:88
- MSTN gene 2:264
- MTOR signaling pathway 3:100
- MUGEN mouse database (MMdb) 2:103
- Multi Dimensional Scaling (MDS) 1:490–491
- multi-agent systems (MAS) 1:310, 1:315, 1:315–317
- multi-allele neutral WF model 2:16–18
- multi-class classifiers 1:508
- multi-class domains, application to 1:686–688
- multi-databases system 1:1092, 1:1093–1094
- multi-dimensional data analysis, case studies in 3:66–67, 3:71–72
- multi-dimensional random walk (MDRW) 1:90, 1:92
- multi-domain architecture (MDA) 2:473, 2:482
- multi-domain context 2:482–483
- multi-factor differential expression analysis 3:796
- multi-FASTA (MFA) 2:273
- Multi-granularity DEviation Factor (MDEF) 1:458
- multi-layer artificial neural networks 1:621
- multi-layer knowledge representation 1:595–598

- multi-layer networks 1:635–638
- multi-layer neural networks 1:619–620, 1:619F
- multi-layer perceptron (MLP). *see* Backpropagation
- multi-level single factor DEG analysis 3:796
- multi-loop subcompartment model (MLS model) 2:302
- multi-omics data 3:452
 - analysis 3:196–197
 - high throughput omics datatypes and methodologies 3:194–195
 - integrating transcriptome with 3:457–458
 - integration model 2:161–162
 - integrative analysis of high throughput omics data 3:196–197
- multi-scale modelling in biology 2:900–901, 2:900F
 - case study 2:901–902
 - computational implementation 2:901
 - model formulation 2:900–901
- multi-specific (MS) 2:143
- multi-threaded shared memory 1:218
- multi-variate Gaussian mutation operator 1:325
- multicategory support vector machines (MC-SVMs) 1:276
- multicellular organisms 1:916, 2:53, 3:938
- multiclass SVM 1:396–397
- multicloud model 1:239
- multicore computers 1:162
- multidigraph 1:1087
- multidimensional databases (MDD) 2:96–98
- multidimensional NMR spectroscopic techniques 2:515
- multidimensional QSAR 2:669
- multidimensional scaling (MDS) 1:278, 1:1022
- multidrug resistance efflux pumps 3:927–928
- MultiExperiment Viewer (MeV) 1:334
- multifactor dimensionality reduction
 - algorithm (MDR algorithm) 1:272, 2:174
- multifunctional protein 3:969–970
- multifurcating trees 2:728–730
 - phylogenetic networks 2:730
- multigraph 1:924, 1:924F, 1:1087
- multilayer FNN 1:390
- multilevel coarse graining 2:560
- multinomial distribution 1:970
- multinomial logistic model (MLM) 1:734
- Multi-parameter Intelligent Monitoring in Intensive Care (MIMIC) 2:1049
- multiparametric panel of proteome profiles 3:911, 3:911F
- multiple alignment format (MAF) 2:273
- multiple alignments 1:45, 1:102, 1:1009–1010;
 - see also* pairwise alignment (PWA)
 - Gr-mlin 1:1010
 - multiple alignment based approaches 3:335–337
 - NetCoffee 1:1011
 - SMETANA 1:1010–1011
- multiple alleles 2:15–18, 2:16F
- multiple array based strategy 3:816
- multiple cancer sources 2:341–343
- multiple co-inertia analysis (MCOA) 2:743
- multiple contact index (MCI) 2:451
- multiple drug resistance (MDR) 3:926
- multiple genome aligner (MGA) 2:272
- multiple hypothesis testing 1:695
- multiple independent random walkers (MIRW) 1:90, 1:92
- multiple instructions multiple data (MIMD) 1:162
- multiple instructions single data (MISD) 1:161–162
- multiple isomorphous replacement (MIR) 2:514
- multiple learners combination 1:536–537, 1:539–540
 - building ensemble learning system 1:521–523, 1:522F
 - cascading algorithm 1:540–541, 1:540F
 - cross-validation 1:536–537
 - ensemble learning 1:537, 1:539–540
 - motivation and fundamentals 1:519–521
 - error rate of three ensemble classifiers 1:521F
 - possible output's combination of 3 classifier 1:521T
 - stacking algorithm 1:537–538, 1:538F
- multiple linear regression (MLR) 1:421, 1:491, 1:725–726, 2:664, 2:670, 3:4
- multiple reaction monitoring (MRM) 3:466, 3:855, 3:896–897
- multiple regression analysis 1:744
- multiple sequence alignment (MSA) 1:253, 1:709, 2:270, 2:491, 2:703, 2:738, 2:825–826, 2:1143–1144, 3:11, 3:257, 3:315–316, 3:335, 3:426, 3:612–613
 - algorithms 1:30
 - tools for 1:30
 - MSA-based interaction prediction 2:825–826
 - and phylogenetic tree construction 3:14–15
- multiple testing corrections 3:444
- multiplex network analysis 2:1137
- MULTIPRED2 tool 2:955
- multiprocessor computers 1:162
- multipurpose internet mail extensions (MIME) 1:169
- multiscale coarse-graining (MC-CG) 2:560
- multispecies PPIN 1:984
- multivariate analysis based on Unifrac
 - distance 3:23–25, 3:24F
- multivariate analysis of variance (MANOVA) 1:745
- multivariate methods 1:742–744, 1:744
 - cluster analysis 1:745
 - data quality assessment 1:744
 - FA 1:745
 - linear and quadratic discriminant analysis 1:744–745
 - logistic regression analysis 1:744
 - MANOVA 1:745
 - multiple regression analysis 1:744
 - multivariate statistics 1:741–742
- multivariate regression 1:722
- mummichog* method 2:406
- Munich Information Center for Protein Sequences (MIPS) 1:1105
- murine cytomegalovirus (MCMV) 2:142
- Mus musculus* 3:821
- Mus-Musculus organism 1:151–152
- Muse-Gaut model 2:715
- MUST 2:214
- MUSTv2 2:214
- MutaGene 2:1080–1081
- Mutation Assessor tool 3:3
- mutation effects on 3D-structural reorganization
 - AIDS and HIV 3:706
 - HIV-Pr 3:706–707
 - HIV-pr 3D-structure and dynamics 3:709
 - I47V mutation effect on JE-2147 and TMC114 binding 3:710
 - I50V and I50L/A71V mutations effect 3:711–714
 - mutations and drug resistance of HIV-protease inhibitors 3:707–709
 - V321 and M46L mutations effect 3:715–717
- mutation-selection models. *see* Halpern-Bruno style models
- mutation(s) 2:2, 2:122–124, 2:712–713, 3:944, 3:945F
 - analyses on lineage tree structure 2:979–980
 - bioinformatics in mutational studies 2:123–124
 - continuum WF model with 2:11–12, 2:12–13
 - of HIV-protease inhibitors 3:707–709
 - protein folding rates prediction on 2:453
 - rates estimation 2:13–14
 - maximum likelihood estimator 2:13–14
 - multiple alleles 2:15–18
 - relationship to Watterson estimator 2:14–15
 - stationary distribution of multi-allele neutral WF model 2:16–18
 - stability of proteins on 2:454
 - in Wright-Fisher model 2:7–9, 2:8F
- mutational profiling (MaP) 3:579, 3:581–582
- Mutations for Functional Impact on Network Neighbors (MUFINN) 2:123–124
- MutationTaster tool 3:393
- multinomial logistic regression (MLR) 1:732
- mutual best-hits. *see* bidirectional best-hits (BBH)
- mutual information (MI) 1:355, 2:157, 2:806
- mycobacteria 3:105
- Mycobacterium tuberculosis* (MTB) 2:685, 2:689–690, 2:821, 2:830, 3:105, 3:106, 3:652, 3:653F, 3:660F, 3:928–929
 - elucidating structural dynamics and mechanisms of MTB targets 3:655–657
 - infection 3:780
- Mycobacterium ulcerans* 3:426–427
- Mycoplasma genitalium* 3:425, 3:427
- Mycoplasma pneumoniae* 2:1022F, 2:1051

MycoRegDB database 2:1054
 myGRV database 1:1056
 myocardium 2:903
 modelling passive and active mechanical properties 2:903
 myoglobin of abalone *Sulculus* 3:968
 MySQL database 3:137–138
 MyVariant. info tool 1:254
 mz5 format 2:77–78
 mzData format 2:77
 mzIdentML format 2:78
 mzML format 1:140, 2:77
 mzQuantML format 2:78
 mzTab format 2:78
 mzXML format 2:77, 2:86

N

- n-grams 2:1102
 language modeling with 1:562–563
 n-octyl- β -D-glucoside (β -OG) 3:761–762
 N-terminal domain (NTD) 3:607
 N-terminal pro-peptide of type III collagen (P-III-NP) 3:898–902
 N-termini 3:265
 N-a-benzoylarginineethyl ester (BAEE) 3:734
 N6-methyladenine (6mA) 3:341
 NADPH dependent retinol dehydrogenase/reductase (NRDR) 3:845
 Na-ve B cells 2:940–942
 Na-ve Bayes (NB) 2:1005, 2:1103, 3:289
 classifier 1:403, 1:406, 1:407–408
 learning 1:406
 methods 3:107
 Naive Bayesian-based model (NB-based model) 1:688
 NAMD program 1:262, 2:529
 named entity normalization (NEN) 1:578, 1:580, 1:605–606
 named entity recognition (NER) 1:566–568, 1:567T, 1:579–580, 1:580T, 1:603, 1:605–606, 2:634, 2:1090, 3:997
 nanoCAGE 3:820
 nanopore technology 3:348–349
 nAnTi-CAGE 3:820
 nascent RNA 2:221
 NASSAM server 3:548
 NATALIE aligner 1:1007
 National Aeronautics and Space Act (NASA) 2:1134
 National Bioscience Database Center (NBDC) 2:80, 3:30
 National Cancer Institute (NCI) 1:861, 2:109, 2:179, 2:342, 2:685, 2:689, 2:689–690, 2:1079
 National Center for Biomedical Ontology (NCBO) 1:580–581, 1:815, 1:861
 National Center for Biotechnology Information (NCBI) 1:112, 1:263, 1:1093, 2:80, 2:103, 2:104, 2:187, 2:342, 2:347, 2:388, 2:629, 2:1059, 2:1085, 3:11, 3:29–30, 3:522, 3:954
 organismal classification vocabulary 1:840–841
 taxonomy 2:889
 National Center for Macromolecular Imaging (NCMI) 2:465
 National Center for Research Resources (NCRR) 1:839
 National Health and Nutrition Examination Surveys (NHANES) 2:175
 National Health Service (NHS) 3:425
 National Human Genome Research Institute (NGHRI) 2:178, 2:179, 2:342, 2:1079, 3:293, 3:388
 National Institute of Allergy and Infectious Diseases (NIAID) 2:932, 3:108–109, 3:726
 National Institute of Genetics (NIG) 3:30
 National Institute of Health (NIH) 1:132, 1:254, 1:839, 2:105, 2:402
 National Institute of Standards and Technology (NIST) 1:236, 2:1069, 3:862
 National Institutes of Fitness 3:996
 National Library of Medicine (NLM) 1:816, 2:629, 2:1085
 National Organization for Rare Disorders (NORD) 1:838
 National Science Foundation (NSF) 2:1011, 2:1014F
 native protein structure 3:612–613
 coarse-grained protein structure models 3:609–610
 conformational transitions 3:606–607
 ENM 3:610–611
 Flexpred 3:611–612
 MD 3:607–609
 Native-ChIP (N-ChIP) 2:149
 Natural Antisense Transcripts (NATs) 2:167–168
 natural killer cells (NK cells) 2:906
 natural language processing (NLP) 1:291, 1:444, 1:561, 1:576, 1:588–589, 1:589, 1:589–591, 1:603, 1:899–900, 1:1047–1048, 1:1096, 2:634, 2:790, 2:1083, 2:1084, 2:1085, 2:1099, 2:1099–1101, 2:1104, 3:997
 applications of active learning to biomedical 1:569–571
 challenges in 2:1100–1101
 evolution 1:561
 programming 1:155
 sub-problems and applications to biomedical text 1:561–562
 syntax layer and semantics layer 1:562F
 natural language query processing 1:582
 natural selection 3:943
 natural systems (NSs) 1:309
 natural-language text 2:1099
 Naturdata database 2:1111
 Natureserve Explorer 2:1113
 Navier-Stokes equations 2:903
 NAViGaTOR 1:1028
 NCBI Entrez Protein database 3:244
 NCBI GenBank database 3:928–929
 NCBI Reference Sequence Database (RefSeq) 2:102
 NCBI Reference Sequence Database (RefSeq) 2:203
 NCMIR's ATLAS 3:129
 nearest neighbor interchange (NNI) 2:742
 nearest neighbour algorithm 1:940, 1:946–947
 nearest-neighbour clustering 2:478
 Needleman Wunsch algorithm (NW algorithm) 1:23, 2:270, 2:1144–1145, 3:316, 3:981
 Negation as failure (NAF) 1:295
 negative binomial distribution (NB distribution) 1:733, 2:227, 2:382
 negative binomial model (NBM) 1:734
 negative-predictive value (NPV) 1:549, 2:955
 Neighbor-Joining algorithm (NJ algorithm) 2:705, 2:708, 3:992
 in MEGA 3:992
 neighborhood list 2:552
 neighborhood-based entity set (NEST) 1:1075
 NeighborNets tool 2:705
 Neighbourhood Local Search (NLS) 2:817
 nelfinavir 2:585, 3:744T–748
 nematode (*Caenorhabditis elegans*) 2:252, 3:103–104, 3:429
 NeMoFinder tool 1:99
 Neo4j 1:842, 2:1063, 2:1066, 2:1068, 2:1069, 2:1070F, 2:1071F
 EMR data visualization 2:1069
 performance of MS SQL server *vs.* 2:1071–1072
 storage requirement of MS SQL server *vs.* 2:1072–1073
 NeOn toolkit 1:807–808
 Nephele project 1:254
 Nervous System Disease NcRNAome Atlas (NSDNA) 2:1050–1051
 nervous system diseases (NSDs) 2:1050–1051
 nested genes association with speciation 3:967–968
 Nestedness metric based on Overlap and Decreasing Fill (NODF) 2:1135, 2:1135F
 NetAligner aligner 1:1009, 1:1068
 NetCoffee method 1:1011
 NetCTL package 3:244–246
 NetCTLpan package 2:966
 NetMHC 4.0 package 2:955
 NetMHCII 2.2 package 2:955
 NetMHCIIpan 3.1 package 2:955
 NetMHCpan 3.0 package 2:955
 NetMHCpan package 2:966
 NetPath database 1:152, 1:1059
 NetRecodon database 2:754
 NetStruct package 3:370
 network 1:90, 1:958, 1:1106, 2:814
 algorithms for network clustering 1:95–96
 in biology 1:915–916, 1:920–921
 gene co-expression networks 1:919–920
 gene regulatory networks 1:916–917
 metabolic networks 1:918
 protein-protein interaction networks 1:918–919
 signaling networks 1:917–918
 centralities 1:950–951
 examples 1:953–954

- measures 1:951
 - cohesion 1:964–965
 - controllability 2:790
 - data formats 1:1026–1027
 - diameter 2:816
 - drawing 1:1085
 - evolution 1:974F
 - flow optimization 2:807
 - inference 2:791
 - molecular networks inference 2:805–806
 - models 1:968, 1:968–972
 - network based approaches 1:878, 2:790
 - network-based data integration 1:1096–1097
 - network-based methods 2:800–801
 - in neuroscience 2:807–808
 - representation 1:81–82
 - science 1:978, 2:1131
 - theory 3:370
 - training 1:391–392
 - validation and assessment 2:160–161
 - NETwork ALigner (NETAL) 1:1007–1008
 - network aligners, graphlets in 2:818–819
 - network analysis (NA) 1:81, 1:89, 3:370, 3:477
 - databases and tool 2:1134
 - databases on ecological networks 2:1134
 - environmental data 2:1134
 - NA-based approaches 1:81
 - NA-based methods 3:107–108
 - Network Medicine 2:808–810
 - network motif score (NMS) 1:1100
 - network motifs 1:98, 2:814–816, 2:815F
 - analysis 2:790–791
 - detection 1:936
 - network neighbourhood, disease
 - mechanisms based on 3:284–285
 - Network of Cancer Genes (NCG) 2:1080
 - network properties
 - complete directed graphs 1:930F
 - complete undirected graphs 1:930F
 - graph 1:928
 - methodologies 1:930–931
 - undirected graph 1:929F
 - network sampling. *see* snowball sampling (SBS)
 - network topology 1:958–960, 3:284
 - assessment of structural properties 1:962–964
 - biological networks 1:961F
 - dyad and triad census 1:961F
 - measures 1:936, 1:936–937
 - network topology-based analysis 2:802–803
 - tools and methodology 1:960–962
 - network visualization software 1:1027–1030
 - free visualization tool 1:1029F
 - PPI network 1:1030F
 - software packages and data formats 1:1028F
 - network-based analysis
 - for biological discovery
 - experimental methods 3:283–284
 - PPIN-based network analysis 3:284
 - insilico* prediction of PPIs 3:286–287
 - of host-pathogen interactions
 - experimental methods to determine PPIs 3:933–934
 - extracting PPIs 3:935–936
 - PPI network-based analysis of PPIs 3:934–935
 - in silico* prediction of PPIs 3:936
 - network-based models of integrative omics
 - data analysis 3:197
 - NetworkBLAST aligner 1:1009, 1:1068
 - Networks Science 1:968
 - neural network (NN) 1:281, 1:303F, 1:390, 1:390–392, 1:613, 2:48–49, 2:491, 3:231, 3:289
 - models 1:636–637
 - prediction based on two levels 2:492–493
 - synoptic table of deterministic activation functions 1:392T
 - training network 1:391–392
 - Neural Open Markup Language (NeuroML) 2:885
 - neuraminidase (NA) 3:275–276
 - neurogenetic algorithm (NGA) 3:3
 - neuroimaging techniques 2:808
 - neurological ageing 2:343–344
 - NeuroML tool 2:889
 - neuron 1:612, 1:634
 - number 1:632, 1:632F
 - neuronal progenitor cells (NPCs) 3:827, 3:827F
 - neuropsychiatric disorder *de novo* mutation
 - database (NPdenovo mutation database) 2:1060–1061
 - neuroscience, networks in 2:807–808
 - Neuroscience Information Framework
 - Standard Ontology (NIF Standard Ontology) 1:861
 - neutral and nearly neutral theories of
 - molecular evolution 2:720
 - neutral loss (NL) 2:896
 - neutron scattering technique 2:465–466
 - neutrophil extracellular traps (NET) 2:993, 3:731
 - new chemical entities (NCE) 2:679
 - Newton-Raphson algorithm 1:427
 - Newton's equation of motion 2:590–591
 - next generation sequencing (NGS) 1:22, 1:128–129, 1:142, 1:160–161, 1:171–172, 1:209, 1:252, 1:252–253, 1:253, ,, 1:259, 1:463, 1:706, 1:928, 1:1047–1048, 1:1104–1105, 2:102, 2:156–157, 2:179, 2:179–180, 2:189, 2:214–215, 2:223–225, 2:308, 2:329, 2:352, 2:367, 2:388, 2:922, 2:1142, 3:17, 3:44, 3:47, 3:62, 3:67–68, 3:113, 3:144, 3:157, 3:176, 3:201–202, 3:203, 3:231–232, 3:285–286, 3:293, 3:294, 3:344–345, 3:346–347, 3:355, 3:380, 3:389, 3:434, 3:447, 3:449, 3:452, 3:453, 3:576–578, 3:581–582, 3:819–820, 3:820, 3:822, 3:834–835, 3:843, 3:852, 3:928–929, 3:965T, 3:970;
 - see also* RNA sequencing (RNA-seq)
 - aligners and methylation callers 3:345
 - commercial systems for 1:128–129
 - data analysis workflow for 3:158, 3:160F
 - expression analysis 2:360
 - genome assembly 3:160–161, 3:161T
 - NGS-based methods 2:149
 - pipeline application for NGS preprocessing
 - automation 3:145
 - preprocessing 3:144–145
 - and quality assessment 2:352
 - primary analysis 2:356
 - process workflow 3:157–158, 3:159F
 - raw data pre-processing 3:158–160
 - read mapping 2:1143, 3:161
 - secondary analysis 2:358
 - sequence generation 3:158
 - spike in controls 3:345–346
 - tertiary analysis 3:161–162
 - toolkit 3:204–205
 - trimming 3:344–345
- next reaction method 1:751
- Nextflow database 3:137
- Neyman-Pearsonian hypothesis testing 1:405
- Ng, Jordan and Weiss algorithm (NJW algorithm) 1:442, 1:1043
- NGL Viewer database 2:533
- NGSmethDB database 2:128–129
- NIH Roadmap Epigenomics Mapping Consortium 2:178
- Nipah* virus 2:831
- NIST MS Search 2.0 package 3:471
- NIST Standard Reference Database 1A v17 (NIST 17) 2:403
- nitric oxide (NO) 3:852
- NNScore tool 2:1054
- no free lunch theorems for optimization 1:519
- Node Controller (NC) 1:240, 1:245
- node correctness (NC) 1:999
- Node Management System 2:397–399
- node sampling (NS) 1:90
- node sampling with contraction (NSC) 1:90, 1:91
- node sampling with neighborhood (NSN) 1:90, 1:90–91
- nodes 1:221, 1:966–967, 2:814
 - colors 1:99–100
- NOE technique 3:525
- noise 1:747
- NOISeq 3:817
- nominal logistic regression 1:425–426
- non linearly separable problems 1:617–619, 1:619–620
- non parametric data reduction 1:482–483
- non-alignment approaches 3:337–338
- non-biological networks 1:915
- non-bonded interactions 1:53, 2:1144, 2:1145
- non-coding prediction approaches 3:232–233
 - software tools for 3:233F
- non-coding regulatory element analysis 2:186–187
- non-covalent interactions in protein
 - structures 2:447–450, 2:450F
- non-deterministic polynomial time hard (NP-hard) 2:814
- non-DP algorithms
 - integer programming 2:581

- non-DP algorithms (*continued*)
 - stochastic sampling 2:581
- non-esterified fatty-acids (NEFAs) 2:894
- Non-Euclidean measures 1:439
- non-hierarchical method 1:745
- non-histone proteins 2:128
- non-infectious diseases, vaccine on 3:247
- non-linear classifier, difference between linear and 1:390
- non-local environment (NLE) 1:56
- non-long terminal repeats (Non-LTRs) 2:126
- non-mono-phyletic traits 2:734
- non-monotonic reasoning 1:287–288
- non-negative matrix tri-factorization (NMTF) 2:819
- non-overlapping ontologies 2:1
- non-parametric bootstrap 1:766
 - bootstrap confidence intervals 1:769
 - bootstrap estimation of prediction error 1:771–772
 - notation 1:766
 - ordinary non-parametric bootstrap 1:768–769, 1:768F
 - sampling distribution of sample statistic 1:766–768
- non-parametric model 1:374, 1:738
- non-parametric statistics 1:664–666
 - Mann–Whitney test 1:666–669
 - Wilcoxon signed-rank test for location 1:664–666
- non-physical interactions 2:850
- non-protein coding (ncRNA) 3:9
- non-random structural patterns 2:1133
 - searching for 2:1131–1133
- non-recursive PPs (nPPs) 1:803
- non-repudiation 1:254
- non-standard protein coding genes, annotation of 2:200, 2:203–204
- non-steroidal anti-inflammatory drugs (NSAID) 3:729
- non-structural proteins (NS proteins) 3:761, 3:975
 - NS1 protein 3:13, 3:765, 3:766F
 - NS4 protein 3:772
 - NS5 protein 3:772–775, 3:773F, 3:774F
- non-synonymous coding SNPs (nsSNPs) 2:1048
- non-synonymous single nucleotide variants (nsSNVs) 3:1
- non-synonymous SNPs (nsSNPs) 2:118–119
- non-tree-like evolution 2:705
- non-Watson-Crick (NWC) 3:504–505
- NONCODE database 2:1081–1082
- noncoding RNAs (ncRNAs) 1:280–281, 2:165, 2:346, 2:575, 2:1050–1051, 3:47, 3:47–53, 3:48F, 3:50F, 3:53–54, 3:230, 3:231–232, 3:799–800
 - biogenesis of miRNAs, siRNAs, piRNAs and lncRNAs 3:232F
 - characterizing and functional assignment of 3:35–36
 - databases 3:33–34, 3:34T
 - genes 3:306
 - lncRNA-seq analysis 3:800
 - miRNA-seq analysis. 3:799–800
- nonlinear regression
 - models 1:735
 - fitting regression models 1:734–735
 - general LM 1:731
 - GLM 1:731
 - GLM implementation in R 1:733–734
 - other nonlinear techniques 1:733
 - regularization methods 1:735
 - prediction 1:420
 - goodness of model 1:422–423
 - nonlinear regression 1:421–422
 - nonparametric tes 1:693, 1:694, 1:698, 1:702–703;
 - see also* parametric test
 - statistical tests
 - for *k* samples 1:704–705
 - for “one-sample” case 1:702–703
 - for “two-sample” case 1:703–704
- nonpolar amino acids 3:511
- nonsense mediated decay (NMD) 2:187, 2:227
- nonsynonymous single-nucleotide polymorphisms 3:928–929
- norepinephrine 2:677
- normal mode analysis (NMA) 3:610
- normal phase liquid chromatography (NPLC) 3:466
- normalised mutual information (NMI) 1:355
- normalised spectral abundance factors (NSAF) 3:904
- normalization 1:478, 1:603, 1:605–606, 2:360, 3:365, 3:469, 3:815
 - for HTS 3:816–817
 - for microarray 3:816
 - between sample 2:381–382
 - within-sample 2:380–381
- normalized databases construction 2:1137–1138
- normalized gene ontology consistency (NGOC) 1:1000
- Normalized Google Distance measure (NGD measure) 1:882
- normalized mutual information (NMI) 1:445
- normalized relative quantities (NRQ) 3:831
- NormSys registry 2:889–890, 2:890F
- Not Only SQL databases (NoSQL databases) 2:98–99, 2:1049, 2:1063, 2:1064, 2:1069–1071
 - BASE principal 2:1064
 - benefits for EMR systems 2:1064–1065, 2:1066T
 - graph database 2:1065–1066
 - property graph model 2:1066
 - relational databases *vs.* 2:1069–1071
 - types 2:1064, 2:1065T
- NP-hard problem 1:967
- NS2B–NS3 protease 3:765–768, 3:767F, 3:768F, 3:769F, 3:770F, 3:771T–772
- NS3protease (NS3pro) 3:765–766
- nuclear export signal (NES) 2:54
- nuclear export signals 2:56
- nuclear localization signal (NLS) 2:54, 3:731
 - prediction 2:56
- nuclear magnetic resonance spectroscopy (NMR spectroscopy) 1:38, 1:562, 2:143, 2:145, 2:396, 2:399, 2:403, 2:410, 2:426, 2:460, 2:512, 2:513F, 2:514–515, 2:516, 2:528, 2:541, 2:585, 2:586, 2:631, 2:680, 2:798, 2:835–836, 3:6, 3:6–7, 3:7, 3:10, 3:252, 3:274, 3:465, 3:478, 3:521, 3:586, 3:606, 3:722, 3:764;
 - see also* mass spectrometry (MS)
 - data preprocessing 3:468–469
 - NMR based metabolomics 3:465
 - spectroscopy 2:427–428
- Nuclear Overhauser effect spectroscopy technique (NOESY technique) 2:515, 3:525
- nuclear protein predicting protein abundance of 3:848–850
 - comparing accuracy of available prediction models 3:850–851, 3:851F
- data resources to develop models 3:849–850
- predicting abundance of nuclear proteins 3:851–852
- predicting mRNA folding energy 3:852
- predicting protein abundance 3:852
- nuclear receptor (NR) 3:4
- nuclear ribosomal gene cluster 3:185
- nucleases 3:575
- nucleic acid
 - binding residues 3:683–684, 3:685–686
 - binding site prediction tools 3:679–681
 - computational tools for 3:678–679
 - evaluating confidence levels of predictions 3:682–683
 - secondary structure prediction 3:508–510
 - sequencing
 - DNA sequencing technologies 3:293
 - Rna-Seq 3:294
 - sequencing data formats 3:294–295
- Nucleic acid builder (NAB) 2:128, 2:527–528
- nucleic acid database (NDB) 2:567–568, 2:568–569, 2:570–572, 2:572T, 3:504–505, 3:522, 3:540–541
 - applications 2:570
 - curator workflow 2:570F
 - entry format 2:569
 - sources and number of NDB data 2:568–569
 - user interface 2:569–570
- Nucleic Acid Programmable Protein Array assays (wNAPPA assays) 2:827
- nucleic magnetic resonance (NMR) 2:915–916
- nucleic sequence alignment algorithms 3:296
- nucleolus 2:53
- Nucleosome energetics (NucEnerGen) 2:312T–313
- nucleosome positioning 2:308
 - ATP-dependent nucleosome repositioning 2:314
 - binding of transcription factors and chromatin proteins 2:311
 - chemical modifications of DNA or histones 2:311

- genetic and epigenetic factors affecting nucleosome positioning 2:309–310
- intrinsic DNA sequence affinity of histone octamer 2:310
- nucleosomes interaction 2:311
- parameters characterizing 2:308, 2:309F
- statistical positioning of nucleosomes 2:310–311
- theoretical approaches to predict nucleosome positioning 2:314
- Nucleosome Positioning Prediction Engine (NuPop) 2:312T–313
- Nucleosome Repeat Length (NRL) 2:289, 2:308–309
- Nucleosome-Occupancy Study for Cis-elements Accurate Recognition (Nu-OSCAR) 2:312T–313
- nucleosome-positioning score calculator (nuScore) 2:312T–313
- nucleosomes (nuc) 2:293, 2:301
- local structure of 2:288–292
- nucleotide 2:567–568
- changes 3:388
- nucleotide sequences 2:754
- resources 3:153
- sequence databases 2:102
- substitutions 2:721
- Nucleotide-nucleotide BLAST (BLASTN) 2:1143
- nucleus HAS PART chromosome 1:824
- null hypothesis (H_0) 1:688–689, 1:691, 3:821–822
- numeric based representation 3:447–448
- numeric prediction 1:639
- Numerical Markup Language (NuML) 2:888
- numerosity reduction 1:482;
- see also* dimensionality reduction
- non parametric data reduction 1:482–483
- parametric data reduction 1:482
- Numerous Sequence Alignment (MSA) 3:3
- NusB-NusE interactions 2:831
- NVIDIA C compiler (*nvcc*) 1:166
- NxTrim tool 3:296
- O**
- O-phosphorylation 2:1072
- OASES 2:358
- obesity 3:17, 3:54
- object-oriented parallel languages 1:216–217, 1:218;
- see also* composition based languages
- distributed programming in Java 1:218–219
- HPC++ 1:218
- object-oriented programming (OOP) 1:199, 1:707–708
- in Python 1:197
- objects management in Perl and BioPerl 1:707–709
- OBO Foundry 1:815
- OBO-Edit 1:702
- OCA 2:467
- Oceanic Ni-o Index (ONI) 3:976
- octamer 2:288
- oestrogen receptor (ER) 3:796
- off-target analysis 2:656–657
- Office of Cancer Genomics (OCG) 2:1078–1079
- olfactory receptors (ORs) 3:252, 3:456–457
- oligo probes. *see* DNA molecules
- oligonucleotide 3:50–53
- sequences 3:13
- oligonucleotide ligation assay (OLA) 3:434
- omics 1:126, 3:911
- bridging issues 2:1070–1071
- data 1:254
- datasets 3:194
- integrator 3:821
- resources 2:347
- strategies 3:463
- OmicsDI database 2:403
- OMIM database 3:170
- OmniPath database 1:1059
- on-line analytical processing (OLAP) 2:96–98
- oncogene B-cell lymphoma 2 3:6
- oncogenic miRNAs 3:53–54
- Oncorhynchus tshawytscha* 2:1124
- one sample T-test 1:699
- one-class classifiers. *see* semi-supervised methods
- one-dimensional Gaussian distribution 1:457
- one-dimensional proton NMR spectroscopy (1D NMR) 3:465
- one-drug-one-gene-one-disease 1:339
- one-sample chi-square test 1:703
- one-sample Kolmogorov-Smirnov test 1:703
- one-way ANOVA 1:702
- OneDep system 2:460–461
- Online Genes to Cognition (G2C) 1:113
- Online Mendelian Inheritance in Man (OMIM) 1:700, 1:898, 1:903, 2:103, 2:1090, 3:169, 3:444, 3:919
- online taxonomy databases, impact of 2:1125
- OntoDLP database 1:786
- OntoEdit tools 1:807
- ontology 1:569, 1:785, 1:858, 1:1063, 1:1104, 1:1104–1105, 1:1107F, 2:1, 2:1126–1129
- for adding semantic information 2:888–889
- in bioinformatics 1:809
- and biology 1:788
- languages for modeling ontologies 1:809–810
- for life sciences 1:810
- design 1:786–787
- languages 1:785–786
- management tools 1:806
- ontology based databases 2:98–99
- ontology-based annotation methods 1:867, 1:867–868
- ontology-based data integration 1:1096
- ontology-based workflow editor 1:333
- RDF 1:790–792
- solution for bioimage databases using 2:1024–1025
- tools for ontology management 1:787–788
- Ontology Based Information Retrieval (OBIR) 2:1025
- Ontology for Biomedical Investigations (OBI) 1:138–139
- Ontology of General Medical Science (OGMS) 1:839
- Ontology of Genes and Genomes (OGG) 1:583
- Ontology of Physics for Biology (OPB) 1:788
- ontology specification 1:785
- Ontology Web Language (OWL) 1:147, 1:288, 1:786, 1:790, 1:794–798, 1:798T, 1:809, 1:815–816, 1:826, 1:838, 1:859, 1:1063, 2:1157
- opcode 1:161
- Open Bioinformatics Foundation (OBF) 1:706
- open biological and biomedical ontology (OBO) 1:130, 1:700–701, 1:788, 1:815, 1:815–816, 1:832, 1:838, 1:858, 1:1064, 2:5
- Open Cloud Computing Interface (OCCI) 1:238
- Open Grid Services Architecture (OSGA) 1:1105–1106
- Open Proteomics Database (OPD) 3:866
- Open Provenance Model (OPM) 2:1157
- Open Provenance Model for Workflows (OPMW) 2:1157
- open reading frame (ORF) 2:182–183, 2:346, 2:361, 3:47, 3:154, 3:233, 3:283, 3:323–324, 3:722, 3:849, 3:968
- open reference method 3:188
- Open Science Grid (OSG) 1:163
- Open source 1:243
- Open Source Protein REdesign for You (OSPREY) 3:647
- Open Targets 2:468
- Open World Assumption (OWA) 1:793
- OpenBabel tool 2:591–592
- OpenCL programming languages 2:1142
- OpenMP database 1:164, 1:216, 2:401, 2:1065
- OpenNebula database 1:244–245, 1:244F
- OpenStack database 1:243–244, 1:244F
- OpenSWATH workflow 2:91
- operands 1:161
- operating system (OS) 3:138
- operational taxonomic unit picking (OTU picking) 3:188, 3:205
- Operational Taxonomic Units (OTU) 2:728B, 3:18–19, 3:184, 3:207, 3:991
- clustering DNA barcodes into 3:991
- OPLS-AA-SEI method 3:672
- optical approach 2:299
- Optical Character Recognition tools (OCR tools) 1:576
- optimal evolutionary algorithm 2:160
- optimal solutions range 2:440
- optimal treatment, pharmacogenomics for 2:1052
- optimization based methods 3:2
- optimize codon usage in DNA 3:329

- optimized potential molecular dynamics (OPMD) 3:4
 Optimized Potentials for Liquid Simulations (OPLS) 1:63–64
 Optnetalign approach 1:1006–1007
 Orchidaceae of Central Africa 2:1012, 2:1015F
 ordinal logistic regression 1:425–426
 ordinary differential equation (ODEs) 1:142, 1:917–918, 1:918, 1:1058–1059, 2:160, 2:791, 2:801, 2:864, 2:920, 2:920–921, 2:985, 3:838;
 see also stochastic differential equations (SDE)
 constructing GRN using 2:160
 model 2:875–876
 ordinary graph 1:1086
 ordinary least squares (OLS) 1:727–728, 2:429
 ORFFinder tool 3:323–324
 Organ System Heterogeneity DB 2:1051
 organelles 3:13–14
 Organization for Economic Cooperation and Development (OECD) 2:665–667
 organizational networks 1:81
 organizing chemical facts into databases 2:606
 Orientations of Proteins in Membranes (OPM) 2:47
 original algorithm (OA) 1:223
 original engineered algorithm (OEA) 1:223
 orotidine 5'-monophosphate decarboxylase (OMP decarboxylase) 1:43
 orphamizer 1:706
 orthogonal drawing 1:1087, 1:1088F
 Orthogonal Partial Least Squares Discriminant Analysis (OPLS-DA) 2:420, 3:470
 Orthogonal Projections to Latent Structures (OPLS) 2:430
 Orthogonal Signal Correction (OSC) 2:430
 Orthogonal Signal Correction Partial Least Squares (OSC-PLS) 2:664
 orthologs 2:260, 2:840T, 3:943, 3:943F
 ortholog-based sequence approach 2:806
 orthologous genes 3:426, 3:428
Oryza longistaminata 3:821
 oseltamivir (OTV) 3:275–276, 3:744T–748
 Otsu technique 2:1029, 3:128
 Out Of Bag (OOB) 1:377
 error estimate 1:377
 Out-Of-Bag error (OOB error) 1:530
 Outer Membrane Protein F (OmpF) 3:842
 outgroup rooting 2:731–732, 2:731F;
 see also midpoint rooting
 adding 2:732
 using gene duplications as 2:732
 selecting 2:731–732, 2:732F
 outlier detection 1:456, 1:456–457, 1:475
 angle-based outliers 1:459
 density-based outliers 1:457–458
 distance-based outliers 1:457
 isolation-based outliers 1:458–459
 outlier ensembles 1:460
 outlier explanation 1:460–461
 statistical-based outliers 1:456–457
 subspace-based outliers 1:459–460
 Outlier Detection using Indegree Number algorithm (ODIN algorithm) 1:458
 outlier mining 1:456
 outparalogs 2:260
 output neurons 1:627
 ovarian cancer, transcriptomic database of 2:342
 over-representation analysis (ORA) 2:405–406, 2:801–802
 overfitting 1:273–274, 1:273F, 1:723, 3:333
 overlap graph method 3:177
 Overlap-Layout-Consensus (OLC) 2:181, 2:357–358, 3:207, 3:301–302
 Overlap/Layout/Consensus approach (OLC approach) 3:161
 overlapping genes with speciation 3:967–968
 alternate splicing 3:968
 functional convergence 3:968
 gene sharing 3:969–970
 molecular repair 3:970
 overlapping genes 3:967–968
 RNA editing 3:968–969, 3:969F
 OWL Description Logics (OWL DL) 1:786, 1:1104–1105
 Oxford BSPlib 1:219
 Oxford Nanopore Technologies (ONT) 2:330, 3:180, 3:436, 3:453–454
 oxidative bisulfite sequencing 3:342–344
 OxyBS package 3:346
 oxygen 3:1
- ## P
- P-CURVE algorithm 3:519
 P-glycoprotein (P-gp) 3:928
 P-SEA algorithm 3:519
p-value 1:693, 2:237
 approach 1:714
 reporting results and misinterpretation of 1:694–695
 threshold 1:896
 Pacific Biosciences (PacBio) 2:181, 3:204, 3:436, 3:453–454
 sequencing 3:294
 Pacific Biosystems 2:330
 packaging RNA (pRNA) 3:539
 PAD type 4 (PAD4) 3:729
 PageRank algorithm 2:1133–1134
 paintomics 3:821
 pair interaction frequency (PIF) 2:302
 pair-potentials function 2:561
 pair-wise sequence alignment 2:270, 2:1144–1145
 paired end sequencing (PE sequencing) 2:185, 2:329, 2:357, 3:169
 paired end tag (PET) 2:301
 paired two-samples T-test 1:700
 pairwise alignment (PWA) 1:22–23, 1:102, 1:1001;
 see also multiple alignments
 global aligners 1:1001
 GRAAL 1:1003–1004
 index-based aligners 1:1005
 IsoRank aligners 1:1001–1002
 local aligners 1:1009
 methodologies 1:23–26
 other aligners 1:1007
 swap-based aligners 1:1006
 Pairwise Term SS measures 1:890
 palaeogenetics 3:375
 paleobiochemistry techniques 3:946–947
 “paleolog” 2:261
 PALSSE algorithm 3:519
 Pan Assay Interference Compounds (PAINS) 2:594–595
 pan-cancer analysis of dysregulated TFs–miRNA FFLs 2:861
 Pan-European Species-Directories Infrastructure (PESI) 2:1111
 pan-genome informatics 3:954–955, 3:955F, 3:956T
 atypical sequence signatures 3:954–955
 GIs 3:955
 mobile elements 3:955–957
 pancreatic cancer 3:26
 pandrug-resistant (PDR) 3:926
 panspermia 3:942
 paracrine 1:917
 ParallABEL tool 3:382
 parallel analysis of RNA structure (PARS) 2:580
 Parallel Analysis of RNA structures with Temperature Elevation (PARTE) 2:580, 3:580
 parallel architectures 1:161–162
 bioinformatics 1:171–174
 parallel computing platforms 1:210
 parallelization paradigms 1:209–210
 programming languages for 1:164–165
 parallel computing platforms 1:200, 1:210
 cluster computing 1:210
 GPU computing 1:211–212
 low power devices 1:212
 supercomputing 1:212–213
 virtual clusters 1:210–211
 XeonPhi computing 1:212
 parallel programming languages 1:215
 parallel reaction monitoring (PRM) 2:91, 3:855
 parallel tempering metadynamics (PT metadynamics) 2:554
 parallel virtual machine (PVM) 1:200
 parallel β -sheets 2:489
 parallel-processing development environments 1:247
 Hadoop 1:248
 Spark 1:248–249
 parallelization paradigms 1:209–210
 paralog edge deletion 1:1010
 paralogous verification method (PVM) 2:806
 paralogs 2:260, 2:837, 3:943, 3:943F
 parameter estimation 1:464–465
 parametric active contour model 2:1032
 parametric approach on transformed counts 2:383–384
 parametric data reduction 1:466, 1:482
 parametric test 1:693, 1:693–694, 1:698, 1:698–699;

- see also* nonparametric test
 evaluating estimator 1:739–740
 MLE 1:738–739
 parametric statistical inference 1:738
 statistical tests
 for k samples 1:701–702
 for “one-sample” case 1:698–699
 for “two-sample” case 1:700
 paramutation 2:128
 Pareto dominance 1:1006
 ParkDB database 2:343
 Parkinson’s disease (PD) 2:343, 2:1096
 parse tree 1:564, 1:565F
 parsimonious FBA (pFBA) 2:440
 parsimony
 algorithm for calculating Parsimony length
 of tree 2:706B
 length 2:706
 methods 2:727
 parsimony-generated phylogenetic tree
 root 3:946
 part of speech tagging (POS tagging) 2:1100
 Part-of-speech (POS) 1:563, 1:592, 1:603
 tagging 1:563–564, 1:577
 partial chromosomal duplication 3:965
 partial community analysis approaches
 3:203
 Partial differential equations (PDE) 2:801,
 2:864, 2:876–877, 2:920–921
 partial gene duplication 3:965
 partial least squares discriminant analysis
 (PLS-DA) 2:405, 2:420, 3:470
 partial least squares method (PLS method)
 2:430, 2:664, 2:669
 partial subgraph 2:815, 2:816F
 Partially Intronic RNAs (PINs) 2:167–168
 Particle Mesh Ewald summation (PME
 summation) 2:553, 2:1145–1146
 particle-particle, particle-mesh method
 (PPPM method) 2:553
 partis package 2:946
 partitional clustering algorithms 1:449
 partitioning methods 1:303, 1:440–441
 partner redundancy method 2:840T
 Pascal (pathway scoring algorithm) 2:860
 passive demethylation 3:341
 passive learning method 1:570
 PASTEC program 2:214
 PathBLAST aligner 1:1009, 1:1068
 PathCards tool 1:1079
 PathExpand databases 1:1075
 PATHOgenic YEAs Search for Translational
 Regulators and Consensus Tracking
 (PathoYeast) 2:127–128
 pathogenicity Prediction through Logistic
 Model tree (APOGEE server) 3:3
 pathogens 3:103–104, 3:104T, 3:104–105
 Pathosystems Resource Integration Center
 (PATRIC) 3:928–929
 pathway alignment approaches 1:1067–1068
 pathway annotation classes based on data
 source 1:1071–1074
 de novo pathway discovery 1:1075–1076
 design, services, and data sharing 1:1076–
 1077
 domain-specificity of pathways 1:1076
 extended pathway databases 1:1075
 integrated and hybrid pathway databases
 1:1074–1075
 level of pathway details 1:1076, 1:1077F
 platforms for pathway reconstruction
 1:1077, 1:1078T
 primary pathway databases 1:1074
 pathway context 1:1081
 pathway curation 2:789–790
 automatic pathway mining 1:1080–1081
 manual pathway mining 1:1079–1080
 pathway databases 2:797–798
 using BioPAX 1:152
 drug pathway databases 2:799
 gene regulation databases 2:798
 for metabolic pathway 2:797–798
 Reactome pathway database 2:800
 signaling pathway databases 2:798–799
 SMPDB 2:800
 WikiPathways 2:799–800
 pathway informatics 2:796, 2:801–802
 drug related pathways 2:796–797
 gene set-based scoring 2:802
 metabolic pathways 2:796
 network topology-based analysis 2:802–
 803
 ORA 2:801–802
 pathway construction
 data curation based approaches 2:797–
 798
 quantitative approaches 2:800–801
 regulatory pathways 2:796
 signaling pathways 2:796
 pathway visualization 3:470
 patient similarity networks 1:1016
 patient sub-typing, example of cluster
 analysis application to 1:355–356
 Patient-centered Research Institute (PCORI)
 1:154
 patient-specific treatment regimes 2:1053
 personalized medicine and
 pharmacogenomics 2:1053
 real-time health monitoring 2:1053
 patter confidentiality 1:267
 pattern
 discovery 2:120
 evaluation 1:329
 identification 2:120
 pattern-based methods 3:244
 recognition 1:516, 2:1085
 space. *see* input space
 pattern-based representation. *see* character-
 based representation
 Paxtools 1:1063
 Pcomb program 3:596
 Pcons program 3:595
 PDBbind database 2:632–633
 PDBeFold database 2:463, 2:472, 2:472T
 PDBePISA database 2:463
 PDBsum database 2:466–467, 2:534
 PDGene database 2:1050–1051
 peak calling 3:114
 broadly enriched DBPs 3:114
 DBPs with mixed signals 3:114
 point-source DBPs 3:114
 peak detection 2:895–896
 protocol establishment 3:76
 peak expiratory flow rate (PEFR) 1:718–719
 peak list formats 2:76–77
 peak picking 3:468
 PeakRanger (software package) 1:253
 Pearson correlation coefficient 1:725, 1:763,
 1:920, 2:157, 3:456
 Pearson product-moment correlation
 coefficient (PPMCC) 1:706, 1:708–
 709, 1:709
 Pearson’s chi-square test 3:443
 Pedant-Pro system 3:153
 Peer-To-Peer network (P2P network) 1:268
 penicillin 3:926
 Penn Treebank tokenization standard 1:562
 people, public and private partnership (3P
 partnership) 3:978
 PEP-3D-Search 2:965
 peptide 3:729;
 see also protein-peptide interactions
 bond 1:562, 3:511
 identification strategies 2:87
 peptide-based inhibitor 3:736–737
 peptide-based PAD4 inhibitors
 activity 3:734–736, 3:735F, 3:736F
 design 3:734, 3:734T, 3:735F
 peptide-MHC binding prediction 2:910–
 914, 2:914–915
 selection for quantification 3:882–884
 data visualization 3:887
 statistical analysis for quantification
 3:884
 sequence 3:312F
 peptide mass finger printing (PMF) 1:127,
 2:1065, 3:857–858
 peptide nucleic acid (PNA) 2:145, 3:941–
 942
 “peptide sequence tag” algorithm 2:1068
 peptide spectrum matches (PSMs) 2:1067
 peptide-spectra matching (PSM) 2:78,
 2:1069, 3:861, 3:892, 3:921
 validation 3:863–864
 PeptideAtlas database 2:79, 3:866
 PeptideAtlas SRM Experiment Library
 (PASSEL) 2:79–80
 PeptideProphet 3:863–864
 peptidyl transferase center (PTC) 3:11
 pepXML format 2:78
 percent accepted mutation (PAM) 1:28
 Percent of Applied Dose Dermal Absorbed
 (PADA) 2:664
 percentile bootstrap 1:558
 Perceptron 1:390–392, 1:612, 1:612–614,
 1:621–622, 1:634–635;
 see also Backpropagation
 artificial neuron 1:635F
 biological neuron and synapsis 1:612F
 convergence theorem 1:617
 learning rule application 1:614–617
 linear classification example 1:636F
 multiple class classification 1:617
 non linearly separable problems 1:617–619
 and multi-layer neural networks 1:619–
 620
 single layer 1:612–614

- Perceptron (*continued*)
percolation graph matching (PGM) 1:1008–1009
periodic analysis 3:25
periodic boundary check (PBC) 2:552
periodicity discrimination method (PDM) 3:25
periodontal ligament cells (PDLs) 1:835
periplasmic binding proteins (PBPs) 3:636–638, 3:637
 adaptability study 3:636–638
 computational modification 3:638
Perl 1:176, 1:176–177, 1:178–181;
 see also BioPerl
 arithmetic operators 1:180–181
 bitwise operators 1:181
 conditional statement 1:181–182
 equality operators 1:181
 Fibonacci sequence example 1:184–185
 install Perl on machine 1:176–177
 loops 1:182–184
 write and execute Perl program by using IDE 1:177–178
 write and execute Perl program by using terminal 1:178
Perl Package Manager Index (PPM Index) 1:707
Perl workflow management system 3:137
perldoc command 1:176, 1:177F
permutational multivariate analysis of variance (PERMANOVA) 3:19
peroxisomal targeting signal 1 (PTS1) 2:54
peroxisome proliferator-activated receptor gamma 3:6
persistence length/Kuhn length 2:292–294, 2:293F
persistence-of-vision ray (POV ray) 2:525
personal genome machine (PGM) 3:204, 3:435–436
Personal Health Records (PHR) 1:266
personalized cancer therapy (PCT) 2:1059
personalized medicine 2:1053
 and text mining 2:1088–1089, 2:1089F
personalized proteomics 3:918–919
perturbation factor (PF) 2:803
Petri Nets (PN) 2:439, 2:877, 2:922, 3:489
Pfam program 2:39–40, 2:102, 3:522
PGMapper tool 1:911
pharmaceutical agents 3:780
pharmacodynamics (PD) 2:634
pharmacodynamics/pharmacokinetic (PKPD) 2:1097, 3:66
Pharmacogene Variation Consortium (PharmVar) 3:383
pharmacogenetics 3:379–380
 bioinformatics resources for 3:382
 population analysis of pharmacogenetic polymorphisms
 bioinformatics resources for
 pharmacogenetics/-genomics 3:382
 genetic basis of variability 3:379–380
 in practice 3:384–385
 workflow in 3:380F, 3:380–382, 3:381F
Pharmacogenetics of Absorption, Distribution, Metabolism & Excretion genes (PharmaADME) 3:384
Pharmacogenetics of Membrane Transporters (PMT database) 3:384
pharmacogenomics 2:1053, 3:379–380, 3:380F
 bioinformatics resources for 3:382
 databases 2:1059–1060
 for optimal treatment 2:1052
Pharmacogenomics Knowledge Base (PharmGKB) 2:186, 2:634, 2:1059–1060, 3:382–383
Pharmacogenomics Research Network (PGRN) 2:634
pharmacokinetics (PK) 2:634
 genes 3:379
 and systems biology modeling 3:999
pharmacology 2:633–634
Pharmacology Interactions Database (PhID) 3:384
Pharmacometrics Markup Language (PharmML) 2:889
pharmacophore 1:78, 2:619
 modeling 2:677, 2:683F, 2:685
 commercial tools and web servers 2:679T
 ligand profiling 2:684
 ligand-based pharmacophore modeling 2:679–680
 limitations 2:684–685
 pharmacophore-guided fragment design 2:684
 pose filtering 2:684
 prediction of pharmacological activity 2:684
 PVS 2:683–684
 role of excluded volumes 2:681
 scaffold hopping 2:683
 structure-based pharmacophore modeling 2:680–681
 3D database search 2:683
 validation 2:681
 modelling 3:753–755
 pharmacophore-guided fragment design 2:684
pharmacophore fingerprints (PF) 2:619
pharmacophore-based virtual screening (PVS) 2:683–684
pharmacovigilance 3:999
PharmGKB 2:1054, 2:1094
PhD-SNPg 3:1
PhenIX 1:706
PhenogramViz 1:706
PhenoMeNa App library 2:401
phenomizer 1:706, 1:706–707
PhenoPred 1:911
Phenotypic Quality Ontology (PATO) 1:704
phenotypic/phenotypes/phenotyping 2:178, 2:426–427, 3:821–822
 data 3:441
 grouping 3:470
 inference using GSEA 3:821
 network effects between transcriptome and 3:819, 3:819F
 screening 2:1095–1096
 imaging 2:1096–1097
 transgenic organisms 2:1096
phenyl isothiocyanate 3:311
phenylthiohydantoin (PTH) 3:311
Phevor 1:706
PHI networks (PHINs) 3:934
phosphatidylcholine (PC) 2:894, 2:894F
PhosphoPredict 2:16–18
phosphoproteomics 2:1072
phosphorylation 2:15
phosphorylation sites prediction
 ELM 2:18
 PhosphoPredict 2:16–18
PhosphoSitePlus (PSP) 3:35
photoactivatable ribonucleoside-enhanced CLIP (PAR-CLIP) 3:50–53
Phred score 1:111, 2:352
PhyloChip 3:203
phylogenetic analysis 2:700, 3:188, 3:965
 dating gene duplication and other computations 3:970–971
 domain duplication and gene elongation 3:965–966
 exon shuffling 3:966–967, 3:967F
 gene duplication 3:965
 gene gain and loss and speciation events 3:970
 historical proofs and scientific evidences 3:946
 adaptations of aromatases in pigs 3:947
 adaptations of elongation factors 3:947–948
 evolution of biological carbon fixation 3:946
 evolution of glucocorticoid receptor 3:946
 origins of fermentation, evolution of Adh 3:946–947
 nested and overlapping genes and association 3:967–968
 partial gene duplication 3:965
 stages of evolution
 chemical evolution 3:938–939
 common ancestor 3:944–946
 Darwinian evolution 3:942–943
 homology and development 3:943
 molecular evolution 3:939–940
 speciation 3:943
 tracing evolutionary steps using phylogeny 3:948
phylogenetic diversity (PD) 3:188, 3:206
phylogenetic(s) 2:102, 3:948
 conservation of directly observed duplexes 3:580
 conservation and covariation of duplex structures 3:580
 distinguishing between alternative RNA structures 3:580–581
 higher order structure of XIST lncRNA 3:581
 construction 3:14–15
 footprinting 2:284–285, 2:284F, 2:285T, 2:285–286, 3:447
 language of phylogenetic inference 2:700–701
 networks 1:1018, 2:730, 3:370
 profile method 2:840T, 3:288
 profile-based prediction 2:826
 shadowing 2:285, 2:285T, 2:286

- similarity-based methods 3:288
 studies 3:296
 tree rooting 2:727–728
 multifurcating trees 2:728–730
 recognizing rooted trees and other tree attributes 2:732–733
 root phylogenetic tree 2:734
 rooting and evolutionary change 2:728
 rooting methods 2:730
 tree rooting and topology 2:729F
 trees as graphs 2:728B
 trees 2:737
 tree-based approach 3:288
 tree-based prediction 2:826
 Phylogenomic Database for Plant Comparative Genomics (GreenPhylDB) 2:103
 phylogeny
 applications 2:736
 phylogeny-based approaches 1:17
 tracing evolutionary steps using 3:948
 phylogenetic footprinted regions (PFRs) 2:285
 PhyloSift database 3:188–189
 PhymmBL 3:207
 physical chemistry-based methods 2:582
 physical interactions 2:849–850
 yeast two hybrid system (Y2H system) 2:849–850
 PhysicalEntity class 1:148
 physicochemical models of substitution 2:716
 physics and knowledge-based integrated functions 1:64–65
 physics-based energy function 2:147
 physics-based functions 1:63–64
 physics-based scoring functions 1:63
 phytochemicals 3:472
 p-helices 3:516
 Picard Tools 3:167–168
 picotiter plate (PTP) 1:128
 PICRUSt 3:189
 pigs (*Sus scrofa*) 1:248, 3:947
 PILER 2:214
 PINCoC 1:96
 Pinta 1:911
 pipeline 2:1151, 3:135, 3:136
 challenges and opportunities for incentives 2:1159
 integration of web 2:1158
 limitations of technologies 2:1158–1159
 spark integration 2:1158
 unruly provenance 2:1157–1158
 workflow decay 2:1158
 cloud environments 3:139
 frameworks 2:401, 2:1152–1153
 class-based frameworks 2:1154
 configuration frameworks 2:1153
 CWL 2:1154
 implicit and explicit DSLs 2:1153
 workbenches 2:1153–1154
 of high throughput sequencing
 application for NGS preprocessing automation 3:145
 NGS preprocessing 3:144–145
 languages 3:139
 tools 3:135–136, 3:136F, 3:136, 3:140F
 Ad-Hoc/Shell Scripts 3:136
 Bpipe 3:137
 ClusterFlow 3:136–137
 CWL 3:139
 Docker 3:138–139
 eHive 3:137–138
 Galaxy 3:139
 Nextflow 3:137
 pipeline cloud environments 3:139
 pipeline languages 3:139
 Seven Bridges and DNANexus 3:139
 snakemake 3:137
 virtualization 3:138
 PipMaker 2:263
 PISwap aligner 1:1006
 pivot approximation 1:770
 piwi-interacting RNA (piRNA) 2:165, 2:167, 2:326, 2:346, 3:9, 3:33–34, 3:47, 3:232F, 3:234, 3:234–235
 pixel shaders 1:162
 PKNOTS 3:510
 planar drawing 1:1087, 1:1088F
 planar layout 1:1019
 Planetary Biodiversity Inventory (PBI) 2:1011, 2:1014F
 planning agents, AI 1:290
 plant
 HGT in 3:961
 metabolites 3:493–494
 metabolomics in plant biology 3:472
 miRNAs 2:125–126
 non-coding RNA database 2:167
 TFs 2:140
 tissue sample preparation 3:465
 transcriptome databases 2:346–347
 Plant Alternative Splicing database (PASD) 2:103
 Plant Bug PBI 2:1011
 Plant Metabolic Network 2:404
 Plant MicroRNA Database (PMRD) 2:167
 Plant Secretome and Subcellular Proteome Knowledgebase (PlantSecKB) 2:103
 PlantCyc 2:404
 Plants-Ensembl database 2:252
 PlantTFDB 2:140
 Plasmepsins (PM) 2:591
 PlasmoDB 2:1086–1088
Plasmodium falciparum 2:585, 3:323, 3:417–418
 PM-II case study 2:591, 2:592F, 2:593F, 2:594F
 Plasmodium Genomics Resource (PlasmoDB) 2:1086, 2:1088F
 Platform as a Service (PaaS) 1:160–161, 1:163, 1:236, 1:240, 1:249, 1:258, 2:1155
 bioinformatics platforms deployed as 1:258–259
 Platform for RIKEN Metabolomics (PRIME) 3:472
 platykurtic distribution 1:653–654, 1:681
 platykurtotic distribution 1:653–654, 1:681
 PLINK software 2:238, 3:115, 3:365, 3:382, 3:444
 “plus one prior” adjustment 3:336
 Plus Premier Pack 1:912
 PNA-assisted identification of RBPs (PAIR) 2:145
 PockDrug tool 2:544
 pocketoptimizer 3:648
 Point Accepted Mutation (PAM) 1:113–114
 point-of-view ray tracing (POV-Ray tracing) 2:531–532
 point-source DBPs 3:114
 pointwise mutual information (PMI) 1:877
 pointwise mutual information and information retrieval measure (PMI-IR measure) 1:882
 Poisson distribution 3:817
 Poisson example 1:739
 Poisson loss 1:639
 Poisson process 1:749–750
 Poisson random graph. *see* Erdos-Renyi Network (ERN)
 Poisson regression 1:426, 1:729, 1:732–733
 polar amino acids 3:511
 polar solvents 3:464
 polarizable potentials 2:561
 pollination networks 2:1131
 polluted water resources 3:1
 polony sequencing 3:434
 polony-based approaches 3:293
 POLY 2.0 3:525
 polyadenylated RNA sequencing 2:168
 Polycomb repressive complex 2 (PRC2) 2:167
 polyextremophile 3:945–946
 polyline drawing 1:1087, 1:1088F
 polymer chemistry 2:292
 polymerase chain reaction (PCR) 1:128, 1:142, 2:179, 3:432, 3:819
 polymorphism 1:199
 Polymorphism Phenotyping v2 (Polyphen-2) 3:4
 polynomial kernel 1:497
 polynomial regression 1:421
 polynomial-time algorithms for restricted graph classes 2:644–645
 polypeptide 3:1, 3:2F
 conformation 3:2–3
 polypharmacology 2:655
 benefits and drawbacks 3:780
 POLYPHEN program 2:205
 polyphen-2 tool 2:123–124, 3:393
 polyploidy 2:261
 PolyProlineII (PPII) 3:514
 PolyProlineII-helices 3:516
 polysaccharides-macromolecular biopolymer 3:523–525
 advancement 3:525
 polysaccharide 3D structure prediction and analyses 3:525
 structural features
 of chitin polysaccharides 3:525
 of food storage polysaccharides 3:524–525
 PolySearch 1:911
 polysemy 2:1100
 POLYVIEW 2:535
 POLYVIEW-2D 2:535
 POLYVIEW-3D 2:535

- POLYVIEW (*continued*)
 POLYVIEW-MM 2:535
 PON-P 3:4
 PON-P2 3:4
 PON-PS 3:4
 population
 coalescent with population structure and migration 2:747, 2:748F
 coalescent with variable population size 2:747, 2:747F
 correlation matrix 1:742–743
 differentiation 3:372
 dynamics models 2:984–985
 genome informatics 2:188–190
 meta-heuristics approach 1:774–775
 population-wide approach 3:379
 sampling methods 1:774–775
 stochastic search 1:97
 stratification 2:237
 population genetics 3:363, 3:364, 3:364–365, 3:375
 alternatives to Wright-Fisher 2:21–22
 analytic methods and strategies 3:365–366
 application in disease 3:374–375
 assessment and considerations 3:372–373
 continuum Bienaymé-Galton-Watson branching model 2:24–26
 continuum WF model with mutations 2:11–12
 and selection 2:12–13
 derivation of forward Kolmogorov equation 2:26–29
 diffusion limit and forward Kolmogorov equation 2:10–11
 estimation of mutation rates 2:13–14
 haplotypic phase 3:365
 Kingman coalescent and Watterson estimator 2:18–21
 linkage disequilibrium 3:364–365
 mutations in WF model 2:7–9
 peopling of Europe as illustrative case 3:373–374
 quantifying genetic variation and perspectives 3:364
 scanning for selection 3:370–371
 selection in WF model 2:9–10
 small portion of multiple genomic alignment across entire population 2:2F
 Wright-Fisher model 2:3–4
 pore loop 1:832
Porphyromonas gingivalis 3:730–731
 portability 1:237
 Portable Batch System (PBS) 3:136
 Portable Document Format (PDF) 1:576, 1:1051
 Portable Network Graphics (png) 1:1051
 portals 2:335–337
 PorthoMCL 1:254
 POSA 3:612
 pose filtering 2:684
 Position Specific Iterated (PSI) 1:261
 position specific scoring matrices (PSSMs)
 1:16, 2:12, 2:134, 2:147, 3:42, 3:335, 3:447–448, 3:448–449, 3:670, 3:682, 3:983
 position specific weight matrices (PSWM). *see* position specific scoring matrix (PSSMs)
 position specific weights 3:336
 Position-Specific Evolutionary Conservation (sub-PSEC) 3:3
 Position-Specific Independent Count (PSIC) 3:4
 Position-Specific Scoring Matrices (PSSMs) 3:981
 position-specific scoring matrix (PSSM) 2:145–146, 2:491, 2:491F
 positional orthology. *see* topoorthology
 positional weight matrices (PWM). *see* position specific scoring matrix (PSSMs)
 PositionSpecific Iterative blast program (PSI blast program) 3:669
 positive likelihood ratio (LR₊) 1:549
 positive predicted value (PPV) 1:407, 1:548, 1:981, 2:578, 2:955, 3:667
 positive-unlabeled learning (PU learning) 1:571
 positron emission tomography (PET) 1:160–161, 2:808, 2:1028
 POSIX thread library 1:164
 Posmed 1:911
 post alignment processing 2:184, 3:167, 3:437
 post-approval surveillance 2:1052
 drug repurposing 2:1052–1053
 drug safety monitoring 2:1052
 Post-Bisulfite Adapter Tagging libraries (PBAT libraries) 3:344–345
 post-docking evaluation and analysis 2:590
 post-genomic era 3:103
 post-processing method 2:560
 post-transcriptional gene silencing (PTGS) 3:235
 post-transcriptional regulation 3:807
 post-translational modifications (PTM)
 1:835, 2:1063, 3:912
 definite potential for elucidating PTM cross-talk 2:1074
 detection and mapping 2:1072
 identification and localisation 2:1068
 post-translational modifications (PTMs)
 2:15, 2:17T, 2:87–88, 2:155, 3:196, 3:283, 3:857, 3:897, 3:906, 3:933
 biological questions explored through prediction 2:19–20
 prediction tool 2:16, 2:16–18, 2:17F
 of acetylation sites 2:18–19
 of methylation sites 2:19
 of phosphorylation sites 2:16–18
 in proteoforms 3:911–912
 post-translational regulation 3:821
 posterior decoding for maximum expected gain prediction 2:580
 Posterior Probabilities (PP) 2:738–739
 potassium (K) 1:634
 potential energy minimization (PEM) 3:260
 potentials of mean force (PMF) 2:560
 power-law distribution 1:962–963
 power-law network (PLN) 1:90, 1:963–964
 POWER_SAGE software 2:121–122
 Prader-Willi syndrome (PWS) 3:235–236
 pragmas 1:164
 pre-miRNA (precursor miRNA) 2:166, 2:221, 2:325
 Pre-processed and prepared spectra repository (PPSR) 1:333
 Pre-processed spectra repository (PSR) 1:333
 pre-processing 1:160–161, 1:463, 1:561–562, 3:128, 3:165–166
 CFG and syntactic parsing 1:564–566
 data cleaning 1:463–464
 data integration 1:469–470
 data reduction 1:466
 data transformation and normalization 1:467
 language modeling with N-grams 1:562–563
 MS data 2:86
 POS tagging 1:563–564
 of reads for amplicon analysis 3:188
 of sequence reads for metagenomics 3:186
 of sequencing data 3:204
 adapter and quality trimming 3:296
 filtering contamination 3:296–297
 quality control of sequencing data 3:296
 steps for gene expression data analysis 1:464F
 text chunking 1:564
 text normalization 1:561–562
 pre-transcriptional regulation 3:806
 pre-variant calling processing 3:437
 prebiotic Earth 3:938–939, 3:938F
 prebiotics 3:211
 precision (Pr). *see* classifier specificity
 precision medicine 1:355, 3:918–919
 GRN application for 2:162
 precision medicine knowledge base (PMKB) 2:1059
 precision medicine target-drug selection (PMTDS) 2:162
 precision recall product 1:981
 precision-recall curves (PR curves) 2:807
 precursor of membrane (prM) 3:9
 precursor RNA (pre-miRNA) 2:165
 predict protein-protein interactions (PrePPI) 2:840
 Predict SNP1.0 3:4–5
 PredictABEL 3:382
 Predicted Residual Sum of Squares (PRESS) 2:667
 prediction
 algorithms 3:681
 based on amino acid composition statistics 2:492
 based on two levels of neural networks 2:492–493
 bootstrap estimation of prediction error 1:771–772
 making 1:250
 of new ligand-binding site in MTB targets 3:657–658
 performance estimates 2:493–494
 Prediction Algorithm for Proteasomal Cleavages (PAProC) 3:42

- predictive, personalized, preventive,
 participatory-medicine (P4-medicine)
 1:330
 predictive models 1:968, 1:973–975
 assessment for chlorophyll-a concentration
 3:4–5
 graph classification 1:975
 link prediction 1:973–975, 1:974F
 Predictive Rule Mining algorithm (PRM
 algorithm) 1:399
 predictive water quality modeling 3:1
 predictors 1:336, 1:563
 PREKIN 2:527
 prepilin 1 (*PLIN1*) 2:1100
 Pribnow Box. *see* TATAAT
 PRIDE PRoteomics IDentifications (PRIDE)
 2:102
 PRIDE/PRIDE Archive 2:80
 Prim's algorithm 1:940, 1:946, 1:947F,
 1:947
 primary analysis 2:356
 primary databases 2:102, 2:850–851, 2:851T
 primary metabolites 3:489–490
 primary miRNAs (pri-miRNAs) 2:325, 3:47
 primary pathway databases 1:1074
 primary structure information 2:569
 primer sets 2:974
 primitive data types 1:206
 PRIMO 2:451–452
 principal canonical correlation analysis
 (PCCA) 2:558
 principal combination analysis (PCA)
 2:999–1000
 principal component analysis (PCA) 1:278,
 1:282, 1:301, 1:466–467, 1:480, ,,
 1:480–481, 1:486, 1:486–488,
 1:488F, 1:490F, 1:500, 1:745, 1:1067,
 2:320, 2:392, 2:405, 2:419, 2:420F,
 2:429, 2:554, 2:743, 2:1004, 3:3,
 3:60, 3:119–120, 3:366, 3:367, 3:470,
 3:661–662, 3:887
 principal components (PCs) 1:486–487,
 2:392, 2:405
 principle coordinate analysis (PCoA) 1:301,
 3:19, 3:188, 3:206–207
 principle of optimality 1:10
 PRINTS database 2:38–39, 3:32
 prion disease database (PDDb) 2:343
 prions 3:103
 disease 2:343
 PRioritizationN and Complex Elucidation
 (PRINCE) 1:911, 2:811
 private cloud 1:238, 1:258
 Private Information Retrieval techniques
 (PIR techniques) 1:267
 PRO association file (PAF) 1:833
 PRO-CHECK 2:517
 Probabilistic Context-Free Grammar (PCFG)
 1:566
 probabilistic latent semantic analysis (pLSA)
 1:828–829
 probabilistic/probability
 approaches. *see* model-based approaches
 clustering 1:966
 of fixation 2:4
 graph layout 1:1022
 graphical models 1:68–69, 1:279–280
 Bayesian networks 1:279–280
 Markov networks 1:280
 matrix 2:707
 modeling 1:975
 models of retrieval 2:1103–1104
 network approaches 2:807
 probabilistic of error, performance
 measures based on 1:550–552
 probability density function (PDF) 1:766–
 767, 2:158
 probability of loss of function intolerance
 (pLI) 2:189
 probably approximately correct learnability
 (PAC-learnability) 1:531
 proBAM format 2:1073
 probiotics 3:208–211
 ProbKnot 3:510
 problem representation 1:775
 process Algebras (PA) 2:879
 process calculi 2:879–880
 Process Description Diagram 1:1065
 process description variant (PD variant)
 2:886
 process-to-processor mapping 1:215
 processing elements (PEs) 2:1145
 PROCHECK 3:523, 3:586, 3:588
 PROCRUSTES 2:199–200
 Prodigal 3:154
 ProDiGe 1:911, 2:811
 ProDom 2:39, 3:32
 PRODORIC hosts data 2:120, 2:140
 ProDrg tools 2:123
 Profile Hidden Markov Models (pHMM)
 2:828, 3:335
 profile/profiling 3:335
 platforms 3:117
 profile-based representation. *see* numeric
 based representation
 profile-profile matching 2:505
 transcriptome landscape by RNA
 sequencing 3:454–455
 ProFit 3:592
 progesterone receptor (PR) 2:558–559
 Program Composition Notation (PCN)
 1:219
 programmability (PP) 1:223
 programming languages 3:372
 progressive alignment 2:1143–1144
 ProHits 2:1072
 Project Management System 2:397–399
 projection matrix. *see* hat-matrix
 prokaryotes 2:183, 3:13–14, 3:426–428,
 3:568, 3:837, 3:944
 to eukaryote 3:953–954
 genome 2:182–183, 3:815
 to prokaryote 3:953
 prokaryotic
 organisms 3:938
 splicing 2:221
 Prokaryotic Genome Annotation Pipeline
 (PGAP) 3:954
 prokaryotic transcription factors, TFDBs on
 2:140
 proliferating in host system 3:106
 proliferation assays 3:40
 PROMISCUOUS database 2:1054, 3:780
 promoter analysis 2:120
 promoter enhancer predictor (PEP) 2:187
 promoter lncRNAs (plncRNAs) 2:168
 promoter methylation 3:806–807
 propensity score method 1:465
 property
 and cardinality restrictions 1:796
 disjointness constraints 1:794
 equivalence constraints 1:794
 functional constraints 1:795
 graph model 2:1066, 2:1067F
 inverse constraints 1:795
 symmetric constraints 1:795
 transitive constraints 1:795
 property paths (PPs) 1:803
 ProphNet 1:911
 proprietary formats 2:76
 ProQ 3:594
 ProQ2 3:594, 3:599
 ProQ3 3:594, 3:599
 ProQ3D 3:594
 ProRank+ 1:980
 ProRules 3:32
 PROSA II 2:518, 3:588
 prosecutor's fallacy 1:693
 ProSight 2:1070
 ProSight Lite 2:88
 Prosit 2:102
 Prosit database (ProDoc) 3:334
 PROSITE database 2:34–36, 2:35T, 2:36T,
 3:338–339
 profile library 2:36–38
 ProtComp algorithm 1:483
 ProtCompSecS method 1:483
 protease inhibitor (PI) 3:710
 proteasome cleavage, prediction systems for
 3:42–44
 Protected Health Information (PHI) 1:155
 protective haplotype 3:444
 Protégé tools 1:806
 Protein ANALysis THrough Evolutionary
 Relationships (PANTHER) 1:1060,
 2:123, 2:1048, 3:3, 3:32
 Protein Arginine Deiminase (PAD) 3:729
 Protein Arginine Deiminase Type IV (PAD4)
 3:731, 3:732F, 3:733F, 3:738F
 activity of peptide-based PAD4 inhibitors
 3:734–736
 autoantibodies in RA diagnosis 3:729–730
 calcium dependency 3:731–732
 citullination in RA cycle 3:730–731
 design of peptide-based PAD4 inhibitors
 3:734
 enzymatic conversion of peptidylarginine
 into peptidyl citrulline 3:730F
 inhibitory activity of peptide-based
 inhibitor 3:736–737
 molecular docking analysis of Inh Dap
 3:737
 substrate recognition 3:732–734
 protein attached chromatin capture (PATCh-
 Cap) 3:76
 protein binding microarray (PBM) 2:148
 Protein Complementation Assay (PCA)
 2:827

- protein complex (pc) 1:982–983, 3:2
evaluation measures for prediction 1:980–981
- protein data analysis 3:310–311;
see also metabolic pathways analysis
mapping proteins to genome 3:313
protein sequencing technologies 3:310–311
protein-protein alignment 3:313–314
reconstructing proteins from mass-spectra 3:311–313
- Protein Data Bank (PDB) 1:42, 1:62, 1:111, 1:261, 1:262, 1:561–562, 1:706, 2:9, 2:288, 2:445, 2:446, 2:460–461, 2:462–463, 2:472, 2:488, 2:497, 2:520, 2:540, 2:568–569, 2:585, 2:586F, 2:632, 2:680, 2:835–836, 2:836, 2:909, 2:934, 2:1104, 3:3, 3:32, 3:32–33, 3:252, 3:274, 3:313, 3:504–505, 3:586, 3:606, 3:620, 3:644, 3:673, 3:722, 3:761, 3:766, 3:919, 3:977
BMRB 2:465
curation and validation 2:461–462
depositing data 2:460–461
file formats 2:462
RCSB 2:464–465, 2:464F
websites of wwPDB members 2:462–463
- Protein Data Bank in Europe (PDBe) 2:460, 2:463–464
- Protein Data Bank Japan (PDBj) 2:460, 2:464
- Protein Data Bank of Trans membrane Proteins (PDBTM Proteins) 2:446, 3:32–33
- Protein *de novo* design 3:646–647
Evodesign 3:647
FireProt 3:647
IPRO 3:647
iRDP 3:647
ISAMBARD 3:647
OSPREDY 3:647
pocketoptimizer 3:648
Protein Wisdom 3:647
Rosetta and Rosetta design 3:646–647
scaffold selection 3:647–648
- protein design
 α -helical structure 3:645–646
 β -sheet structure 3:646
fundamentals 3:645
helix stabilizing properties 3:645F
- protein domain (ProDom) 3:514
identification 3:152
resources 3:153
- protein family (Pfam) 1:39, 3:514
databases 3:30–32, 3:31T
blocks 2:39
InterPro 2:40–42
Pfam 2:39–40
PRINTS 2:38–39
ProDom 2:39
PROSITE 2:34–36
- protein folding 1:561, 3:614
energy landscape 3:614
evolution 2:479–480, 2:480F
kinetics 2:452
folding nuclei 2:452
- prediction from amino acid sequence 2:453
prediction on mutation 2:453
protein folding rates 2:452
relating protein folding rates 2:452–453
- secondary structure elements in 2:488–489, 2:489F
helices (a, 3_{10} , p) 2:488–489
other/loops 2:489
strands and sheets 2:489
- protein folding problem (PFP) 1:561, 1:561–562, 2:451
- protein functional annotation
annotation process in UniProt 2:9–10
critical assessment 2:13
foundations 2:9, 2:9F, 2:10T
gene ontology 2:8–9
integrative and machine learning based methods 2:12–13
- methods
based on sequence profiles and motifs 2:12
based on sequence similarity 2:12
based on structure similarity 2:11–12, 2:11T
UniProt 2:8
- Protein Homology/Analogy Recognition Engine (Phyre) 1:262
Phyre2 1:262
- Protein Identification Resource (PIR) 1:112, 3:313, 3:919
- protein information and knowledge extractor (PIKE) 2:101
- Protein Information Resource (PIR). *see* Protein Identification Resource (PIR)
- protein interaction information extraction search (PIE search) 1:578
- Protein Interaction Network (PIN) 1:938, 2:805, 2:822
- Protein Interaction Network Alignment (PINALOG) 1:1008
- Protein Interactions and Ontology (ProteinOn) 1:891–892, 1:891
- protein kinase R (PKR) 3:574
- protein kinaseC e (PKCe) 3:276
- Protein Knot Server 2:545
- Protein Model Portal 2:466
- PRotein Ontology (PRO) 1:583, 1:832, 1:834F, 1:835, 1:1024
content and structure 1:832–833
development and statistics 1:832
tools and annotations 1:833
- Protein Research Foundation (PRF) 3:313
- Protein Separation (PS) 1:139
- Protein Sequence Activity Relationships (ProSAR) 3:631
- Protein Sequence Database (PSD) 3:313
- protein sequences/sequencing 3:681, 3:821, 3:983
analysis 3:122
under population structure with migration 2:754
prediction of secondary structures 3:517–519
under selection 2:754
- Protein SIMilarity (PSIM) 2:653
- protein standard absolute quantification (PSAQ) 3:872
- protein structural bioinformatics
databases and servers for protein structures 2:448T
levels of protein structures 2:446F
prediction of protein tertiary structures 2:451–452
protein 3D structures analysis 2:446–447
protein folding kinetics 2:452
protein interactions 2:454
protein stability 2:453
proteins and drug design 2:455
proteins classification based on structure and function 2:446
structural organization of proteins 2:445–446
structure databases for proteins and complexes 2:446
- protein structural database (PDB) 1:38
- protein structure 2:472–474, 3:252, 3:258T, 3:259T, 3:260T, 3:260–261, 3:983
automated structure prediction 3:264–265
automated *vs.* manual modelling 3:267
case study 3:261
current status of protein structural universe 2:478–479
datasets 3:679
fold and sequence independent motifs in 3:621–622
model building 3:257
model building, manual way 3:263
and nearest-neighbour resources 2:472T
obtaining target sequence 3:261
protein domains and methods 2:472–474
protocol for protein structure modelling 3:254–255
quality analysis 3:257
refinement 3:257–260
sequence 3:261–262
analysis 3:254–255
and sequence repositories 3:522
structural families in genomic era 2:479
target-template alignment 3:256–257, 3:263
template identification 3:255–256, 3:263
template selection 3:256, 3:263
validation 3:263–264
- protein structure analysis and validation (ProTSAV) 3:257
experimental methods to determine structure 2:512–514
structure validation methods 2:516
- protein structure databases 3:32–33, 3:33T
access to PDB data 2:462–463
challenges for future 2:468
derived protein structure databases 2:466–467
fold classification databases 2:467
integrating structure into bigger picture 2:468
modelling archives 2:466
outside of wwPDB 2:465
PDB 2:460–461
subject-specific resources 2:468

- value of understanding protein structure 2:460
- URLs of resources 2:461T
- Protein Structure Initiative Modification Ontology (PSI-MOD) 1:835
- protein structure prediction 1:62–63, 1:774; *see also* protein three-dimensional structure prediction
 - application to fragment-based 1:778–779
 - EA 1:775–776
 - EDA 1:776–777
 - exploration *vs.* exploitation 1:777–778
 - GA 1:776
 - memetic algorithms 1:777
 - protein structure methods for 3:252–253
- protein structure validation software suite (PSVS) 3:257
- protein structure visualization
 - additional insights from structure 2:533–534
 - Aquaria 2:533–534, 2:534F
 - MolviZ. Org 2:534–535
 - PDBsum 2:534
 - POLYVIEW 2:535
 - Proteopedia 2:534
 - applications in molecular visualization 2:535
 - approaches for protein topology analysis 2:521–522
 - evolution of molecular visualization software 2:527
 - HERA 2:520–521
 - structural topology 2:520
 - 3D structure visualization 2:522–524
 - TOPS 2:521
- protein three-dimensional structure
 - prediction 2:497F, 2:502; *see also* protein structure prediction
 - accuracy 2:503–504
 - assessing quality of structure prediction methods 2:499–501
 - gene sequences 2:502–503
 - problem 2:497
 - selecting and refining models from structure prediction 2:508
 - single native fold 2:509
 - template based protein structure modelling 2:504–505
 - template-free protein structure modelling 2:506
- Protein Topology Graph Library (PTGL) 2:522
- Protein Variation Effect Analyzer (PROVEAN) 3:3
- Protein Wisdom 3:647
- protein-binding sites prediction in DNA sequences 3:447
 - algorithms of motif finding 3:448–449
 - IUPAC codes for nucleotides 3:448T
 - motifs and false positive problem 3:447–448
 - variations of theme 3:449–450
- protein-carbohydrate complex structures (PROCARB) 2:455
- protein-carbohydrate docking 3:673
- protein-carbohydrate interactions 2:455, 3:666, 3:673–674, 3:674
 - knowledge-or empirical-based methods 3:666–667
 - ML-based methods 3:667–669
 - molecular dynamics based methods 3:671–672
 - sequence-based prediction methods 3:669–671
 - structure-based prediction methods 3:667–669
- protein-chemical compound interactions (PCIs) 3:194
- protein-coding genes 2:195
- protein-DNA interactions 1:111, 2:142
 - computational methods for investigating 2:149–150
 - DNA-binding protein structures, dynamics and conformational changes 2:143–144
 - experimental methods to investigating 2:147–148, 2:148F
 - issues in protein-DNA interactions predictions 2:145–147
 - methods 2:149
 - NGS-based methods 2:149
 - representative systems of protein-DNA interactions 2:142–143
 - role of DNA structure and dynamics 2:144–145
- protein-encoding gene 3:307
- protein-ligand
 - binding free energy 3:614
 - complexes 3:659–660
 - docking 2:455
 - interactions 2:455
- protein-peptide interactions 3:2
 - bivariate kernel density 3:5F
 - case studies 3:4–5
 - computational approaches 3:1–2
 - polypeptide conformation 3:2–3
 - PPI 3:1
 - prediction and modelling
 - predicting hot-spot residues 3:4
 - protein-peptide docking 3:3–4
 - statistical mechanical formulation of molecular association 3:3
 - structural intricacies of PPIs 3:1
- protein-protein alignment 3:313–314
 - amino acid scoring matrices 3:315
 - dynamic programming
 - for global alignment 3:314
 - for local alignment 3:314
 - gap penalties 3:314
 - multiple sequence alignment 3:315–316
 - scoring matrices 3:314–315
 - statistical significance of sequence alignment 3:315
- Protein-protein BLAST (BLASTP) 2:1143
- protein-protein interaction (PPI) 1:77, 1:569, 1:747, 1:835, 1:973, 1:997, 1:1016, 1:1036, 1:1075, 2:454, 2:657, 2:814, 2:849, 2:849–850, 2:854, 3:1, 3:194, 3:283, 3:284, 3:934, 3:998; *see also* biological pathway (bp)
- data retrieval 3:411, 3:411T
- databases 1:988, 2:1060–1061, 3:108–109
 - context 1:992–993
 - data integration 1:988–991
 - database functionality 1:993
 - errors in 1:991–992
 - human-bacterial 3:417
 - human-parasites protein 3:417–418
 - human-virus 3:415–417
 - prediction 2:836–838, 2:840–841
 - in silico* prediction for PPI network analysis 3:286–287, 3:286T
 - structural intricacies 3:1
- PROtein-protein interaction network
 - alignment based on PERcolation algorithm (PROPER) 1:1008–1009
- protein-protein interaction networks (PPINs) 1:918–919, 1:978, 1:997–998, 1:1018, 1:1048–1049, 1:1105, 2:162, 2:805–806, 2:810, 2:819, 2:821, 2:824–825, 2:828, 3:194, 3:196, 3:283, 3:934
- alignment 1:997, 1:1000
 - assessment methods 1:998
 - functional mapping 1:1000
 - multiple alignments 1:1009–1010
 - pairwise alignments 1:1001
 - performance comparison 1:1012–1013
 - topological mapping 1:1000–1001
- analysis 3:284, 3:285F
 - context-specificity of protein interactomes 3:285
 - disease mechanisms based on network neighbourhood 3:284–285
 - network topology 3:284
 - rewiring during disease, mutation and evolution 3:285–286
- analysis of PHIs 3:934–935
- applications 2:829
- biological interaction 2:821
- biological network characteristics 2:822–824
- community standards 2:853–854
- computational methods 2:825
- databases 2:850–851
- and dynamic data integration 1:983–984
- essential proteins 2:830
- experimental methods 2:824–825
- G Prorank+ 1:980
- host-pathogen interaction 2:830–831
- interaction representation 2:822
- interactions types 2:821–822
- interolog mapping 2:826
- machine learning 2:826–827
- Mimix 2:829
- MSA-based interaction prediction 2:825–826
- multispecies 1:984
- new drug classes 2:831
- PIN 2:822
- protein annotation 2:829–830
- public databases 2:828–829
- specialization 1:984
- and static data integration 1:983
- system biology 2:830
- text mining 2:826
- 3D structure-based interaction prediction 2:825

- protein-protein interaction networks (PPINs) (*continued*)
- protein-small molecule interactions 2:543–544
- protein-membrane assembly 2:540
- protein-nucleic acid
assembly 2:540
contact data 3:679–681
- protein-nucleic acid interactions (ProNIT)
2:454–455, 3:678
- proteinoids 3:939–940
- protein-protein assembly 2:540
- Protein-RNA Interface Database (PRIDB)
3:681
- proteins 1:561, 1:562–564, 1:564–565,
1:774, 1:1071, 1:1102, 2:198–200,
2:201–202, 2:311, 2:445, 2:445F,
2:512, 2:814, 2:821, 2:1101, 3:196,
3:606, 3:620–621, 3:631, 3:966,
3:983
- abundance prediction
determinants 3:847
experimental technologies for measuring
3:847
linear regression methods to predicting
3:847–848
predicting protein abundance of nuclear
proteins 3:848–850
probabilistic approaches to predicting
3:848
- addition of novel function to existing
3:636–638
- computational modification of
periplasmic binding proteins 3:638
- study of adaptability of periplasmic
binding proteins 3:636–638
- annotation 2:829–830
- binding site-based pharmacophore models
2:622
- cell division interplay between DNA and
3:74–75
- class 2:475–476
- coding SNPs 2:118–119
- context-specificity of protein interactomes
3:285
- core 2:837–838
- dynamics 2:544
- entity 1:832
- expression profiling datasets 2:1095
- flexibility prediction with FlexPred 3:614
- identification 3:855
data analysis 3:858
fragmentation 3:857–858
mass spectrometry strategies 3:855
mass spectrometry-based protein
identification 3:855
sample preparation 3:855–857
- identification of protein post-translational
modifications 3:35
- inference and validation 2:1067–1068,
3:864–866
- localization prediction
prediction of subcellular localization
sites 2:53–54
targeting signal prediction 2:54–55
- microarray 3:847
- molecules 1:997, 2:28
- motifs 3:337
- peptide method 2:840T
- precipitation 3:464
- properties
amino acid composition and molecular
weight 2:28–29
amino acids 2:28
disordered proteins 2:31
hydrophobicity 2:30–31
isoelectric point 2:29–30
of protein surfaces and cores 2:543
quaternary structure 2:31
resources 2:31–32
solubility and crystallizability 2:30
- protein/peptide separation techniques
3:892, 3:893T–895
- PTM prediction 2:20–23
biological questions exploration 2:19–
20
case studies on PTM prediction tools
2:16–18
criteria for evaluating and selecting PTM
prediction tool 2:16
reversible tyrosine phosphorylation
regulates 2:16F
- quantification 3:871
absolute quantification 3:872
through DIA 3:872–874
label-based quantification 3:871
label-free quantification 3:871–872
relative quantification 3:872
- quantification techniques 3:847
- quaternary structures 1:1103–1104, 1:1104T,
1:1104F
- remote homology detection 1:261
- resources 3:153
- secondary structure 2:520
- specific amino acid within 2:20
- structural analysis
alignments in structural context 2:541
computational models of proteins and
assemblies 2:540–541
modeling unknown structures 2:539–
540
power of molecular simulation methods
2:545–546
retrieving experimental protein
structures 2:539–540
sources of protein sequences 2:539T
tools for 2:541–542, 2:542T
- subcellular localization 2:53
- synthesis 1:319, 1:823
- targeted 3:877
- tertiary structures prediction 1:63, 2:451–
452
- threading problem 3:4–5
- 3D structure prediction and analyses
3:522–523
- WT *vs.* mutant
comparing apo form 3:710, 3:712–714
comparing complexed form 3:710–711,
3:714–715, 3:717
- proteins-building blocks of life 3:510–512
- Proteobacteria 3:21–23
- ProteoCloud 1:172–174, 1:253
- proteoforms 3:911
bioinformatics 3:919–921
post-translational modifications in 3:911–
912
- proteogenomics 2:1070–1071
- proteome 3:911, 3:912F
functional role of PTM in 2:20
profiles 3:911
proteome-scale interaction maps 3:283–
284
- proteome informatics 2:84, 2:1073
de novo sequencing 2:1068
identification and localisation of post-
translational modifications 2:1068
mass spectrometry 2:1064–1065
omics bridging issues 2:1070–1071
PSM scoring and protein inference
validation 2:1067–1068
separation techniques 2:1063–1064
- Proteome Standards Initiative (PSI) 1:130,
1:139
- Proteome Standards Initiative-Molecular
Interactions XML (PSI-MI XML)
1:139–140, 1:1064
- ProteomeXchange Consortium (PX
Consortium) 2:80–81, 2:86
- proteomic(s) 1:62, 2:84, 3:1, 3:194, 3:195–
196, 3:313, 3:891
- bottom-up and top-down 3:918
- dark matter 2:1068
- data 3:850
compression 1:483
types and standard formats 2:76, 2:77T
- databases and repositories 2:79, 2:79T
- mass spectrometry data analysis tools
data processing workflow 2:84–86
identification by tandem mass
spectrometry 2:86–88
preprocessing MS data 2:86
from proteomics analyses 2:84
proteomics studies 2:91
quantification 2:89–90
- PX Consortium 2:80–81
- sequences 1:221
- technical issues bearing on data processing
2:1063–1064
- transcriptome to 3:820
- variation and quantitative error 3:898F
- workflows 2:1068–1069, 3:121–122
fundamentals of protein sequence
analysis 3:122
implementation steps of MoRFs
predictor 3:123–125
MoRFs predictor design 3:122
necessity of molecular recognition
feature prediction 3:121–122
state-of-the-art MoRFs predictors 3:122
test step 3:125–126
- PRoteomics IDentifications (PRIDE) 2:80,
3:866, 3:920
- Proteomics Informatics (PI) 1:139
- Proteomics Standards Initiative (PSI) 2:77
- Proteomics Standards Initiative-Molecular
Interaction (PSI-MI) 1:1027, 2:853
- Proteomics Standards Initiative-Molecular
Interactions (PSI-MITAB) 1:1076

Proteopedia 2:467, 2:534
 ProteoWizard toolkit 2:86
 Protist-Ensembl database 2:252
 protocells 3:939–940
 ProtOntology 1:333
 protXML format 2:78
 PROVEAN webserver 2:123–124
 provenance 2:1151
 provenance, consisting of data models (PROV-DM) 2:1157
 provenance-ontology (PROV-O) 2:1157
 provider-centric approach 1:238
 proximity assays 3:847
 proximity ligation assay (PLA) 2:149
 pruning algorithm 2:708
 pruning criterion 1:375–376
 PSAR-Align 2:276
 PSAR-precision 2:276
 Pseudo Amino Acid Composition (PaPI) 3:4
 Pseudo F-Score 2:276
 Pseudo Position-Specific ScoringMatrix (Pse-PSSM) 2:146–147
 pseudo-ancestral material 2:748
 pseudo-counts 3:447–448
 pseudoaligners 3:119
 pseudocode 1:2
 pseudoexonization 3:967
 pseudoxons 3:967
 pseudogenes 2:200, 3:971
 unprocessed 3:971
 pseudoknots 2:575
 programs for predicting secondary structures with 3:510
 programs for predicting secondary structures without 3:509–510
Pseudomonas fluorescens 3:465
 PSI Common Query InterfaCe (PSICQUIC) 2:853
 PSI-BLAST 2:505
 PSIPRED method 3:519
 PSIVER 2:841
 PSM scoring 2:1067–1068
 psoralen analysis of RNA interactions and structures (PARIS) 2:580
 PSSM Relation Transformation (PSSM-RT) 2:145–146
 PsychoProt server 2:545
 PsychoProt webserver 2:541
 psychrophiles 3:945–946
 psychrophilic yeast 3:725
 PubAngioGen 2:1049–1050
 PubChem 2:544, 2:629–631, 2:631F, 2:1090, 3:470, 3:478–480
 database 2:1093–1094
 public chemical databases 2:629F
 bioactivity databases 2:631–632
 ChEBI 2:635–636
 of chemical toxicity 2:634–635
 of drug information 2:633
 HMDB 2:635
 large-scale data aggregators 2:628–631
 number of chemical structures contained in databases 2:630F
 UniChem 2:636
 public cloud 1:237, 1:238F, 1:240, 1:258

public databases 2:828–829, 2:836, 3:357–358
 public health and wildlife disease 2:1118–1119
 PubMed 2:1101
 PubMed Health 2:104
 pupal cuticle protein gene 3:968
 purine group 2:567–568
 purity 1:354–355
 PyIgClassify 2:468
 PyMod 2:451–452
 PyMOL (1999) 2:530, 2:531F, 2:543
 program
 tools 2:123
 PyNAST 3:206
 pyrazinamide (PZA) 3:652–653
 pyrimidine group 2:567–568
Pyrococcus fumaris 3:945
 454 Pyrosequencer 3:435
 pyrosequencing 2:179–180, 3:435
 pyrrolysine 2:200
 Pythagoras' theorem 2:1003
 Python 1:195, 3:69, 3:145–146
 bioinformatics and data science in 1:197–198
 data structures in 1:195–196
 functions in 1:196–197
 object oriented programming in 1:197
 predict relapse of primary breast cancer with 1:379–381, 1:382F
 and R 1:332
 statements 1:195
 Python Package Index (PyPI) 1:197

Q

q-value 3:456
 Qiagen 3:435
 QikProp 3:275
 QMEANclust 3:595
 qpGraph 3:369
 Qprob/MULTICOM-NOVEL 3:595
 quadratic discriminant analysis 1:744–745
 quadratic programming (QP) 1:344–345
 quadruplexes 3:506–508
 quadrupole mass analyzers (QqQ) 2:428
 qualifier 2:3
 qualitative annotations 1:1023–1024
 biological function 1:1023–1024
 disease 1:1025
 subcellular localization 1:1024
 tissue annotation 1:1024–1025
 qualitative model energy analysis (QMEAN) 3:595
 qualitative models 1:144, 2:680
 qualitative variable 1:673–674
 quality
 assessment and scoring 2:852–853
 checking of experimental and modeled structures 2:541–542
 metrics 3:114
 values 2:352–353
 of water 3:1
 quality control (QC) 2:184, 2:376, 2:404, 3:165–166, 3:365–366, 3:441–442, 3:455, 3:793–794, 3:920
 and mapping to genome 3:344
 of raw data 2:353–355
 using read alignment 3:300
 of Rna-Seq 3:307
 score 1:71
 of sequencing data 3:296
 quantification 2:89–90
 DDA and MS1 2:89–90
 DIA and MS2 2:90
 SRM 2:90–91
 quantification Concatemers (QconCAT) 3:872
 quantile normalization 1:467–468, 2:366–367
 quantile regression 1:728
 quantile-quantile plot (QQ plot) 2:237
 quantitative affinity matrices (QAMs) 2:913
 quantitative annotations 1:1025;
 see also qualitative annotations
 quantitative approaches 2:800–801
 Quantitative Association Rules 1:371
 quantitative enrichment analysis (QEA) 2:405–406
 quantitative immunology by data analysis 2:984, 2:984–985, 2:985, 2:986–987, 2:989, 2:989–990, 2:990
 Quantitative Insights Into Microbial Ecology (QIIME) 3:185, 3:188, 3:206–207
 quantitative matrices (QM) 2:954
 quantitative modelling approaches 2:874–875
 Bayesian networks 2:877–878
 BN 2:877
 differential equations 2:875–876
 mathematical or computational modelling 2:875F
 microscopic Ab-initio models 2:878
 PN 2:877
 process calculi 2:879–880
 quantitative polymerase chain reaction (qPCR) 2:388, 3:816, 3:830–831, 3:830F, 3:831T
 quantitative reverse transcription-polymerase chain reaction (qRT-PCR) 2:372
 Quantitative Sequence Enrichment Analysis (QSEA) 2:128–129
 quantitative structure activity relationship (QSAR) 1:78, 2:455, 2:613, 2:613–614, 2:640, 2:661, 2:661–662, 2:662F, 2:677, 2:688, 2:913, 2:954–955, 3:751–752, 3:753, 3:776
 case studies 2:669–671
 classification based methods 2:664
 databases in QSAR studies 2:665, 2:667T
 in drug designing and discovery 2:669
 interpretation & applicability domain analysis 2:668–669
 machine learning techniques 2:664–665
 methodologies 2:662
 model construction 2:662–663
 multidimensional 2:669
 regression based methods 2:664

quantitative structure activity relationship (QSAR) (*continued*)
 software for QSAR studies and modeling 2:665, 2:666T
 tool 2:1054
 validation of models 2:665–667
 quantitative structure-binding affinity relationship (QSBR) 2:662
 quantitative structure-metabolism relationship (QSMR) 2:662
 quantitative structure-property relationships (QSPR) 2:640, 2:661, 2:677
 quantitative structure-retention relationship (QSRR) 2:662
 quantitative structure-toxicity relationship (QSTR) 2:662
 quantitative systems pharmacology (QSP) 3:69
 quantitative trait locus (QTL) 2:248–249, 3:428
 quantitative variable 1:674
 quantum mechanical calculations (QM calculations) 1:63–64
 quantum/molecular mechanics method (QM/MM method) 3:776
 QUARK method 1:66–67, 1:67F, 1:261
 quartiles 1:675
 quasi steady state assumption 2:440
 QUASt 3:305
 quaternary structure 2:31
 query
 clause 1:800, 1:801–802
 combining graph patterns 1:802–803
 filter constraints 1:802
 PPs 1:803
 SPARQL semantics 1:803–804
 triple patterns and basic graph patterns 1:802
 language 2:1067T
 query-based paradigm 1:591
 SPARQL to query ontologies 1:804–806
 test 2:1072
 types 1:801
 querying languages and development
 ontology management tools 1:806
 SPARQL language 1:800–801
 question answering (QA) 3:997
 Quick and Quantitative ChIP (Q2 ChIP) 2:149
 quiver graph 1:1087
 Quote Management System 2:397–399

R

R Bioconductor 2:366–367
 R factor 3:274
 R implementation 1:410–411
 R software programming language 1:199, 1:689, 3:69, 3:117, 3:222;
see also Java programming language
 analysis and assessment
 documentation 1:200
 OOP 1:199

support for parallel computing 1:200
 visualization capability 1:200
 world wide web connectivity 1:199–200
 biological annotation 1:203–204
 environment 1:199
 functions 1:202–203
 GLM implementation in 1:733–734
 installation 1:200
 packages 1:199, 1:203
 R statistical package 2:89
 R statistical software 1:683, 1:719
 syntax 1:200–201
 array 1:201
 data frame 1:202
 factor 1:202
 list 1:201–202
 matrix 1:201
 vector 1:200–201
 Raccoon tool 2:690, 2:691F
 Rada measure 1:878
 radial based function (RBF) 1:497, 3:5
 classifier 2:1005
 rbf-SVM 3:63
 radial layout 1:1021
 radiation-sensitive 4 (Rad4) 2:143
 radio frequency (RF) 2:427
 radio-magnetic indicator (RMI) 1:160–161
 radiometric dating methods 2:722
 radioresistant bacteria 3:945–946
 Raf–MEK–ERK MAPK signaling pathway in cancer 2:856–857
Ralstonia solanacearum lectin (RSL) 3:674
 Ramachandran map 1:55–56, 3:511–512
 RANCoC 1:96
 Rand index 1:354
 random
 graph. *see* Erdos-Renyi Network (ERN)
 layout algorithm 1:1019, 1:1089
 mutation 2:7
 network model 2:815
 protein 3:981
 rooting 2:730
 subsampling methods 1:542, 1:543
 subspace selection 1:376–377
 testing with random permutation tests 1:557
 Random Boolean Networks (RBNs) 1:1055
 random breaking read pairs 2:319
 random component 1:731
 random first sampling (RFS) 1:90, 1:91
 random forests (RF) 1:272, 1:275–276, 1:277, 1:302, 1:344, 1:376–377, 1:528–529, 1:529, 2:1004, 2:1103, 3:1, 3:2, 3:289, 3:669
 out of bag error estimate 1:377
 bagging algorithm 1:527–528
 bagging and random subspace selection 1:376–377
 classifier 3:1–3
 literature applications 1:381–382
 model 1:436
 OOB error 1:530
 randomization 1:528
 use-cases 1:377–379
 variable importance 1:377
 random walk (RW) 1:89, 1:90, 1:91, 1:975

graph kernel 1:499
 random walk with escaping (RWE) 1:90, 1:92
 random walk with restart (RWR) 2:810–811
 randomization 1:9–10, 1:528
 ranker classifier 1:546
 ranking
 algorithms 1:950
 and filtering methods 3:62
 measures 1:552–556, 1:554F
 Rapid Annotation interfaCE for PRotein Ontology (RACE PRO) 1:833
 Rapid Miner 1:331
 RaptorX-FM method 1:68–69
 Rare Exome Variant Ensemble Learner (REVEL) 3:5
 rare variant association studies (RVAS) 3:393, 3:393–395
 rare variants
 annotation 3:393
 detection 3:388–389
 future directions 3:395–396
 rarefaction analysis 3:188
 raster molecules (RasMol) 2:527
 Raster3D (1996) 2:528
 Rat Resource and Research Center 2:103
 rate variation models 2:722
 rate-adjusted midpoint rooting 2:731
 rational agents, AI 1:288–289
 coalitional games 1:289–290
 mechanism design 1:290
 strategic agents 1:289
 rational design with natural RNA motifs 3:539
 rational drug design 3:748
 rational SBDD 2:585
 drug target and structure determination 2:586
 materials and methods 2:591–593
Plasmodium falciparum PM-II case study 2:591
 structure-based virtual screening 2:595–597
 validation of docking methodology 2:595
 rationality 1:310
 raw counts 3:816–817
 raw data 2:76
 mz5 format 2:77–78
 mzData and mzXML formats 2:77
 mzML format 2:77
 peak list formats 2:76–77
 proprietary formats 2:76
 raw reads 3:186
 archives 2:344–345
 raw spectra repository (RSR) 1:333
 ray tracing 2:525, 2:525F
 ray-tracing technique 2:522
 Raynaud's syndrome 2:1105–1106
 Rclick 3:597
 RCSB PDB Protein Comparison Tool 2:472T
 RDF Schema (RDFS) 1:786, 1:790, 1:792–794, 1:809
 RDF/JSON technologies 2:1063
 RDXplorer | CNV detection 3:965T
 re-engineering of proteins 3:635–636

- computational designing of β -sheet surfaces 3:636
 in silico approach to enhance stability of TEV protease 3:636
 re-rooting trees 2:732
 re-weighted random walk (RWRW). *see* respondent driven sampling (RDS)
 reaction motifs 1:99
 reactive oxygen/nitrogen species (RONS) 3:852
 Reactome 1:1059–1060
 database 1:152, 3:317
 pathway
 database 2:800
 knowledgebase 2:404
 viewer 1:151F
 read alignment 3:297
 QC using 3:300
 read depth (RD) 2:360, 3:965T
 read mapping 2:357, 3:161
 read-level filtering 2:319–320
 reading frames (RFs) 2:942
 reads 2:329
 reads per kilo base per million mapped reads (RPKM) 2:360, 2:367–368, 2:380, 3:455, 3:816–817, 3:817, 3:833
 ready-made pipelines 2:1152
 real numbers, CTA for 3:817
 real-life biomedical applications 1:209
 real-time dengue forecast models 3:976
 real-time health monitoring 2:1053
 real-time quantitative PCR (RT-qPCR) 2:224
 rearranged immunoglobulin gene 2:940
 partitioning of 2:943–946
 rearrangement events 3:296
 recall 2:197
 receiver operating characteristic curve (ROC curve) 1:274, 1:346, 1:347F, 1:433, 1:434F, 1:553, 1:686, 2:160–161, 2:681, 2:682, 2:807, 2:955, 3:333, 3:682–683
 receptor preparation 2:592–593, 2:690–691, 3:277
 AutoDock VINA 2:593–594
 UCSF DOCK 2:592–593
 receptor tyrosine kinases (RTKs) 2:857
 receptor-operated channels (ROCs) 2:796
 receptors 1:917, 1:1057–1058
 reciprocal best-hits. *see* bidirectional best-hits (BBH)
 recognition motifs 3:2–3
 recognition-based method 2:1037–1038
 recognition-based segmentation 2:1033
 recombination activation gene (RAG) 2:942
 recombination signal sequences (RSS) 2:942
 RECON 2:214, 2:312T–313
 reconstructing homologous collinearity. *see* synteny mapping
 REctified Linear Unit (ReLU) 1:641
 recurrent artificial neural network (RANN) 3:4
 recurrent neural networks (RNNs) 1:642–643, 1:643F
 recursion-tree method 1:8
 recursive algorithms 1:3–4, 1:8
 recursive elimination function (REF) 1:330
 recursive feature elimination (RFE) 1:515
 ReD tandem program 2:214
 redictors 1:731
 reduced graph models 2:641
 reduced representation bisulfite sequencing (RRBS) 3:324, 3:342–344
 reentrancy 2:1153
 reference 2:4
 reference based assembly 2:180
 reference-based alignment 3:166–167, 3:167T
 reference-based approach 3:310
 Reference SNP (refSNP) 3:170
 RefinedIBD algorithm 3:368
 region-based method 2:1034–1035
 region-based segmentation 2:1030–1031
 region-wide synchrony and traveling waves of dengue 3:975–976
 RegionMoRF method 3:125, 3:126
 regions of interest (ROIs) 2:399, 2:808
 RegNetwork 1:1056
 regression 1:277, 1:338, 1:367, 1:465, 1:482, 1:731
 accuracy measures in 1:432–433
 algorithms 1:300–301
 coefficients 1:722–723
 imputation 1:465
 prediction methods 1:420
 regression analysis 1:689, 1:706, 1:714–717, 1:722–723, 1:723–725
 advanced approaches 1:728
 outliers and influential observations 1:727
 reporting results 1:727
 transformations 1:727–728
 regression based methods 2:158
 MLR 2:664
 PLS 2:664
 validation metrics for 2:667
 for external validation 2:668
 for internal validation 2:667
 RegRNA 2:228
 regular expression 3:334–335
 REGULATES relation 1:825
 Regulatory Sequence Analysis Tools (RSAT) 3:449
 regulatory/regularization 1:641, 3:337
 methods 1:735
 molecules 3:470
 motifs 3:1
 networks 1:1105–1106
 parameter 1:728
 pathways 2:796
 regions 2:120
 RNA mechanism 2:124
 RegulonDB 2:140
 reinforcement learning 1:304–305, 1:305F
 relational database management system (RDBMS) 1:1105–1106, 2:1063
 Relational Database Service (RDS) 1:240
 relational databases 2:96
 graph databases *vs.* 2:1073
 NoSQL databases *vs.* 2:1069–1071
 relational databases 2:1073
 Relational Formal Context (RCA) 1:595
 relations 1:839
 Relationship Extraction (RE) 1:603
 relative absolute error (RAE) 1:433
 relative accessible surface area (RSA) 3:666–667
 relative frequency 1:673
 relative log-expression procedure (RLE procedure) 2:381
 relative quantification 3:872
 relative solvent accessibilities (RSA) 2:535
 relative squared error (RSE) 1:433
 relaxed-clock models 2:722
 relevance networks 2:157–158
 Relief (RF) 3:61
 remote homology
 detection 1:261
 homologous proteins 3:3
 Remote Method Invocation toolkit (RMI toolkit) 1:219
 Renal Gene Expression Database (RGED) 2:343
 RepARK tool 2:215
 REPClass 2:214
 REPDenovo tool 2:215
 repeat associated silencing RNAs (rasiRNA) 2:326
 repeat copy number variation identification 2:215–216
 repeated random walks (RRW) 1:96
 RepeatExplorer pipeline 2:215
 RepeatMasker 2:214
 RepeatModeler 2:214
 repeats 2:210–211, 2:210F
 bioinformatic approaches 2:211–214
 from junk to funk 2:211
 repertoire analyses 2:977
 repertoire diversity 2:977–978
 repertoire measures and coping with inherent biases 2:977
 Repertoire data analysis, analysis pipelines for 2:947–948
 repertoire dissimilarity index (RDI) 2:947
 repertoire sequencing 2:922–923
 REPET 2:214
 repetitions 1:15–16
 replay attack 1:270
 replica exchange MD (REMD) 2:553, 2:555–556, 2:556F
 “replication organelle” 3:934
 reporter proteins 3:836
 REpresentational State Transfer REST 1:167–169
 repressors 2:142
 Reptiles Database 2:1113
 resampling techniques 1:526–527
 RESCUE-ESE 2:228
 research and development activities (R&D activities) 2:677
 Research Collaborator for Structural Bioinformatics (RCSB) 2:460, 2:464–465, 2:464F, 3:977
 Research Data Alliance (RDA) 1:131
 residual analysis 1:423
 residual sum-of-squares (RSS) 1:420
 Residual Variance Intolerance Score (RVIS) 2:189
 residue scanning 3:4

- ResidueMoRF method 3:124–125, 3:125–126
- residues
- binding site modeling based on 2:650–652
 - residue-residue contacts 2:447
- Resilient Distributed Dataset (RDD) 1:222, 2:1158
- resistance-nodulation-division family (RND family) 3:927–928
- Resnik measure 1:879, 1:909
- Resnik's semantic similarity 1:999
- Resource Description Framework (RDF)
- 1:149–151, 1:786, 1:790, 1:790–792, 1:809, 1:826, 1:835, 2:464, 2:885–886, 2:886, 2:888, 2:1063, 2:1157
 - entailment 1:792
 - RDF-based semantics 1:797–798
 - semantics 1:792–794
- Resource for Biocomputing, Visualization, and Informatics (RBVI) 1:262
- respondent driven sampling (RDS) 1:90, 1:92
- response variable or outcome. *see* dependent variable
- RESTful APIs 1:167–169
- Restful model 1:167–169
- restricted Boltzmann machine (RBM) 1:281, 1:282, 1:646
- restricted fragment length polymorphism (RFLP) 3:433
- restriction enzyme mapping 1:6
- Restriction Fragment Length Poly-morphism (RFLP) 2:118–119
- restriction mapping problem 1:6
- retainment process, CBR cycle 1:417
- retention time (RT) 2:1070, 3:468
- filtering 3:878
 - prediction 2:1064
- retrieval process, CBR cycle 1:416
- retro-syntheses 2:609
- retrotransposons 2:126, 2:210–211, 3:356
- reuse process, CBR cycle 1:416
- reverse engineering. *see* biological networks inference
- reverse phase liquid chromatography (RPLC) 3:466
- reverse transcriptase polymerase chain reaction (RT-PCR) 3:13, 3:14, 3:568
- reverse transcription (RT) 3:574, 3:830–831
- reverse vaccinology 3:243
- immunoinformatics approaches 3:246
- reversibility 2:709
- revision process, CBR cycle 1:416–417
- RfaH 3:607
- Rfam 2:167, 2:1018, 2:1018F
- Rhea database 1:152
- rheumatoid arthritis (RA) 3:729
- autoantibodies in RA diagnosis 3:729–730
 - citruination in RA cycle 3:730–731
- rheumatoid factor (RF) 3:729
- Rhodnius montenegrensis* 3:821
- Rhodnius robustus* 3:821
- RIBBONS (1987) 2:527
- ribonuclease P (RNase P) 3:230
- ribonucleic acid (RNA) 1:1100, 2:178, 2:324, 2:372, 2:567, 2:567–568, 3:9, 3:29, 3:230, 3:535–539, 3:536F, 3:537F, 3:574, 3:576F;
- see also* deoxyribonucleic acid (DNA)
- algorithms on RNA secondary structures
- DP for base-pairing probability 2:579–580
 - dynamic programming algorithms 2:579
 - posterior decoding for maximum expected gain prediction 2:580
- biological sources 2:326–327
- chemical and enzymatic structure probing 3:575
- classes 2:324–326
- coding RNA 3:230–231
- editing 3:968–969, 3:969F
- genome wide measurement of RNA unfolding 3:579–580
- information on secondary structure prediction
- joint secondary structures prediction 2:581
 - non-DP algorithms 2:581
 - RNA secondary structure prediction 2:580–581
 - secondary structure prediction 2:581
- lncRNAs 3:236–237
- medium non-coding RNA 3:235
- modifications 2:327–328
- molecules 3:792
- nanostuctures construction 3:539
- via foot-to-foot interactions 3:539
 - via hand-in-hand interactions 3:539
 - via rational design with natural RNA motifs 3:539
 - via RNA architectonics 3:539–540
 - via RNA origami 3:540
- nanotechnology 3:535
- ncRNAs 3:231–232
- beyond one modification per RNA molecule
- combining systematic mutation 3:579
 - determination of long-range interactions 3:579
 - MaP 3:579
- and regulatory years 2:204–205
- RNA-binding domains 3:678
- RNA-binding residue prediction 3:684–685
- RNA-specific cancer database 2:341–343
- RNA-specific methods 3:598–599
- secondary structure 3:536
- sequence and structure analysis 3:9–10
- small RNA 3:233–234
- structure 3:809
- annotation programs 3:548
 - probing with deep sequencing 3:576–578
- structure prediction
- based on maximum expected accuracy 2:578
 - energy models 2:576–577
 - marginal probabilities of secondary structures 2:578
 - of minimum free energy structures 2:577–578
 - reliability of predicted structures 2:578
- of RNA three dimensional structure 2:581–582
- secondary structure 2:575, 2:575F, 2:576F
- secondary structure prediction 2:577–578
- stochastic models 2:577
- supramolecule construction 3:540–543, 3:541F
- synthesis 2:165
- tertiary motifs 3:546
- 3D motifs 3:546
- atlas 2:1021–1022, 2:1023F
- transcriptomics 2:328–329
- universe 3:230–231
- View Image 2:569
- in vivo* chemical probing 3:578
- world 3:941–942
- Ribonucleic Acids Statistical Potential (RASP) 3:597–598
- ribonucleoprotein (RNP) 2:221, 3:231
- ribonucleotides 2:568
- riboregulators 3:568, 3:572
- ribose-binding protein (RBP) 3:637
- ribosomal binding site (RBS) 3:306, 3:323, 3:568, 3:569
- calculator 3:323
- Ribosomal Database Project (RDP) 3:203
- classifier 3:206
- Ribosomal Database Project Pipeline (RDPipeline) 3:188
- ribosomal RNA (rRNA) 2:324, 2:325, 2:372, 2:567, 2:568, 2:575, 3:9, 3:137, 3:18, 3:230, 3:231, 3:294
- 16S rRNA 3:18–19, 3:19F, 3:188, 3:189, 3:203
- 18S rRNA 3:185
- ribosomal rRNA 3:10, 3:10T
- ribosomal SSU rRNA sequence 3:10–11
- Ribosome sequencing (Ribo-seq) 2:333
- ribozymes 2:325, 3:535, 3:792, 3:941–942
- rice (*Oryza sativa*) 3:428–429
- ridge regression 1:514, 1:728
- rifampicin (RIF) 3:652–653
- right-hand side (RHS) 1:388
- RIKEN MSn spectral database for phytochemicals (ReSpect) 2:403–404
- ritonavir 3:744T–748
- RLZ algorithm 1:484
- RM2TS method 1:69, 1:70F
- RNA binding proteins (RBPs) 2:145
- RNA guanine (rG) 2:974
- RNA Immunoprecipitation sequencing (RIP-seq) 2:333
- RNA interference (RNAi) 2:125, 2:126, 3:844–845
- RNA polymerase (RNAP) 1:916, 1:1100–1101, 2:178, 3:7, 3:9
- RNA polymerase II (Rpol2) 2:288, 3:230–231
- RNA proximity ligation (RPL) 2:580
- RNA recognition motif (RRM) 2:223
- RNA-dependent RNA polymerase (RdRp) 3:772, 3:776
- RNA-DNA difference (RDD) 2:328

RNA-induced silencing complex (RISC) 3:535, 3:842

RNA-RNA interaction prediction in 3D structures 3:546, 3:546–547

analysis approach and utility of base arrangement annotation 3:547–548

base-base interactions 3:547

illustrative example(s) or case studies 3:548–551

systems and/or applications 3:547

RNA-sequencing (RNA-seq) 1:115, 1:272, 2:155, 2:156–157, 2:225, 2:332–333, 2:364, 2:372, 2:379–380, 2:379F, 2:388, 3:50–53, 3:94, 3:94–95, 3:97F, 3:113, 3:195, 3:236–237, 3:294, 3:452, 3:454–455, 3:814, 3:820, 3:821, 3:830, 3:833, 3:834F, 3:835T, 3:843

correlation with 3:347–348

data 2:166, 2:332, 3:798–799

normalization 2:367

preprocessing 2:376–377, 2:376F

processing 3:118–119

for variant calling 3:799, 3:799F

differential expression from 2:382–383

borrowing strength across genes 2:383

discrete distribution of read counts 2:382–383

parametric approach on transformed counts 2:383–384

pipeline 3:458–459

quality control 3:307

sequence alignment 3:307–308

RNA-STAR 3:510

RNABindR 3:679

RNABindRPlus 3:679

RNAfold program 3:324, 3:570–571, 3:571

RNAHelix 3:510

RNAnalyzer 3:596

RNAmotif program 3:324

RNAproBR 3:809

rnaQUAST tool 3:310

RNAAssess 3:596–597

RNAstructure software package 3:509–510

RNAz non-coding RNA prediction pipeline 3:234F

Roadmap Epigenome Project 3:395

Robust Multiarray Analysis (RMA) 3:389–390

robust multichip average (RMA) 2:377

robustness 2:737–738, 3:484–485

estimation 2:736

outliers in data 2:739–740

support values

BP 2:738

likelihood-based support values 2:739

PP 2:738–739

taxon sampling 2:740

Roche Nimblegen SeqCap 3:342–344

ROLLOFF software 3:369

root mean square deviation (RMSD) 1:69–70, 1:79, 1:564, 2:501–502, 2:528, 2:590–591, 2:682, 2:693–694, 2:909, 2:955, 3:264, 3:588–592, 3:596, 3:658–659, 3:785

root mean square error (RMSE) 1:433, 1:551, 2:667, 3:4

root mean square fluctuation (RMSF) 2:590–591, 3:659, 3:710, 3:712F, 3:717

root phylogenetic tree 2:734

ancestral sequence prediction 2:734

resolving monophyly and homoplasy 2:734–735

ultrametric phylogenetic methods 2:734

root-finding method 1:739

rooted tree recognition 2:732–733

inferring rooting methods 2:733

long branch artifacts 2:733–734

unrooted tree 2:732–733

rooting 2:727

and evolutionary change 2:728

midpoint 2:730–731

outgroup 2:731–732

random 2:730

unrooting and re-rooting trees 2:732

Rosetta program 1:66, 2:203, 3:510, 3:614

Rosetta algorithm 2:452

Rosetta and Rosetta design 3:646–647

Rosetta macromolecular modeling software 1:774

Rosetta Relax protocol 1:779–780

Rosetta software suite 1:262

Rosetta Stone 3:288

Rosetta:cloud 1:262

Rosetta:home 1:232

Royal Society of Chemistry (RSC) 2:631

rule-based classification 1:388, 1:397

building classification rules 1:388–389

if-then rules 1:388

rule-based expert systems (RBES) 1:415

rule-based taggers 1:577

S

S-adenosyl-L-homocysteine (SAH) 3:772

S-adenosyl-L-methionine (SAM) 3:772

S-Adenosylmethionine Synthetase 1:1103

S. cerevisiae Promoter Database (SCPD) 2:120

S2G 1:911

SAbDab 2:468

SABIO-RK database 1:1052

SAC6 inter-domain peptides flexibility 2:559

Saccharomyces cerevisiae 2:129, 2:161, 2:251–252, 3:74, 3:325, 3:327–328, 3:465, 3:821

Saccharomyces Genome Database (SGD) 1:823, 2:103, 3:874

salinization process 3:1

Salmonella Typhimurium LT2 3:427

salt bridges 1:53

sampling 1:89

analysis 3:465

approaches 1:90, 1:482–483

taxonomies 1:90

distribution 1:766

sandflies of Northern Thailand 3:993

Sanger sequencing 2:179, 3:311, 3:986–987

DNA sequencing methods 3:293

Sankoff's algorithm 2:706, 2:708

sauquinar 3:273, 3:744T–748

saturation transfer difference (STD) 3:658

scaffolding 3:186

scaffolds 2:167

hopping 2:683

selection 3:647–648

scalability 1:237

Scalable Protein Interaction Network

ALIGNment (SPINAL) 1:1008

Scalable Vector Graphics (SVG) 1:1051

scale-based methods 2:48

scale-free distribution 2:823

scale-free network. *see* power-law network (PLN)

scale-free topology 3:485

scale-invariant feature transform (SIFT) 2:999, 2:1037–1038, 2:1041F

Scalenet 2:1111

ScanProsite tool 3:32

Schizophrenia Gene Resource (SZGR) 2:1050–1051

Schizosaccharomyces pombe 3:325

science citation indexing (SCI) 2:1084

Scientific Review Panel (SRP) 2:635

scientific workflow management systems (SWfMS). *see* pipeline frameworks

Scikit-learn, Python 1:197–198

SCnorm 2:369

SCOPextended database (SCOPE database) 2:467

scores plot 2:405

scoring 2:87

classifier 1:546

docking 3:742

tools 2:590T

functions 1:23, 1:27–28, 1:63–64, 1:78, 2:589–590

knowledge-based functions 1:64

physics and knowledge-based integrated functions 1:64–65

physics-based functions 1:63–64

matrices 3:314–315

scoring matrix. *see* substitution matrix

Scran 2:369

screening for non-acceptable polymorphisms (SNAP) 2:119, 3:4

search algorithm 2:589

Search Engine Assessor 1:885

search engines 2:1101

similarity measures 1:881–882

Search Space 1:368

Search Tool for the Retrieval of Interacting Genes (STRING) 2:103, 3:1–2

SearchGUI database 2:87

searching

motifs/sites databases with structure query 3:622–624

tree space 2:708–709

Seattle Structural Genomics Center for Infectious Disease (SSGICD) 3:726

second generation sequencing technologies 2:179–180

second-generation sequencing 3:293–294

secondary protein structure atlases 2:466–467

- secondary structure elements (SSEs) 2:520, 3:267, 3:512, 3:517
 diagram of 16S SSU RNA 3:15, 3:17F
 in protein folding 2:488–489
 in proteins 3:514
- secondary structure prediction 2:490–492, 2:492, 2:492T, 2:494, 2:508–509, 3:570–571
 amino acid propensities 2:490
 assessing 2:490
 based on amino acid composition statistics 2:492
 based on approaches 2:493
 based on deep learning 2:493
 based on two levels of neural networks 2:492–493
 estimates of prediction performance 2:493–494
 inference of secondary structure elements 2:489–490
- secondary structure states (SS states) 2:535
- security 1:237, 2:1074
 analysis 1:269–270
 software 1:243
- “seed-and-chain” strategy 2:272
- “seed-and-extend” strategy 2:270
- Seeded Region Growing (SRG) 2:1030
- seeds 2:270
 seed-dispersal networks 2:1131
 sequences 3:30–31
- segment overlap (SOV) 2:490
- segmentation 2:996–997, 2:1028–1030, 2:1029F, 2:1034, 2:1039–1040, 2:1043T
 clustering-based 2:1032–1033, 2:1036–1037
 computational intelligence-based 2:1033–1034, 2:1038–1039
 contour-based 2:1031–1032, 2:1035–1036
 edge-based 2:1030, 2:1034
 recognition-based 2:1033, 2:1037–1038
 region-based 2:1030–1031, 2:1034–1035
 thresholding-based 2:1028–1030, 2:1034, 2:1036F
- segmentation-based fractal texture analysis (SFTA) 2:999
- selected reaction monitoring (SRM) 2:89, 2:90–91, 2:1070, 3:466, 3:855
- selected/parallel reaction monitoring (SRM) 3:920
- selective 2'-hydroxyl acylation analyzed via primer extension and sequencing (SHAPE-seq) 2:333, 2:580
- selective 2'-hydroxyl acylation and primer extension (SHAPE) 3:575, 3:586
- selective sweep 2:264–265
- selectivity profiling 2:655–656
- selenocysteine insertion sequence (SECIS) 2:200
- SELEX/HT-SELEX analysis programs 3:571–572, 3:571T
- self organizing feature maps. *see* self-organizing maps (SOM)
- self-ligation read pairs 2:319
- self-organized systems 1:316
- self-organizing maps (SOM) 1:441, 2:121, 2:405, 3:2, 3:3, 3:470
- self-splicing introns 3:967
- semantic similarity 1:870, 1:899–900, 1:908–909
 GO 1:903–904
 HPO 1:901–903
 information content 1:871–872
 SB 1:871
 shared ancestors 1:872–873
 shared information 1:873
 similarity measure 1:873–874
- semantic similarity measure (SSM) 1:203–204, 1:873, 1:889–890, 1:908, 1:981
 analysis and assessment 1:891
 functions and measures 1:877, 1:885–886
 features and information content 1:881
 Similarity Library 1:883–884
 similarity measures between words 1:877–878
 strategy to computing similarity between sentences 1:882–883
 systems and/or applications 1:890–891
- Semantic Web (SW) 1:589, 1:591, 1:877
 technologies yield knowledge 1:591
- Semantic-Base (SB) 1:870–871, 1:871
- semantic(s) 1:295–296
 heterogeneity 1:1096
 networks 1:813, 1:825
 SBML 1:863
- semantical layer
 event and relation extraction 1:568–569, 1:568F
 named entity recognition 1:566–568, 1:567T
 text summarization 1:569
 WSD 1:566
- semantically encoded workflows 2:1157
- Semi-Markov random walks cores Enhanced by consistency Transformation for Accurate Network Alignment (SMETANA) 1:1010–1011
- semi-structured data clustering 1:443
- semi-supervised learning 1:301
- semi-supervised methods 1:456, 1:593
- semiempirical approaches 1:78
- SemXClust 1:443
- sensitivity (SE) 1:686, 1:981, 2:578, 2:955, 3:682–683
 analysis 2:793
- Sensitivity Analysis Tool (SAT) 1:1058–1059
- “sensory perception of smell” 3:456–457
- Sentence Boundary Detection (SBD). *see* sentence detection
- sentence detection 1:576–577
- sentence segmentation task 1:562
- Sentence Similarity Measure interface 1:884
- Sentence splitting 1:603
- Sentence Utilities class 1:884
- separation techniques 1:1036, 2:411–412, 2:1063–1064
- SEPIa 2:965
- SeqGraphR program 2:215
- SeqMule 3:169
- sequence alignment 1:30, 1:39–42, 2:356–357, 3:989
- algorithms 1:30, 2:271
- BLAST and similar database search methods 3:295–296
- classic alignment algorithms 3:295
- comparative genomics 3:296
- problem 3:2
- statistical significance 3:315
- Sequence Alignment Map (SAM) 1:133–134, 2:356, 3:161, 3:167
 files 3:298
 tools 3:116, 3:145, 3:167–168, 3:298, 3:437, 3:438
- sequence analysis 3:254–255, 3:292–293
 analyzing genomic sequences 3:295
 analyzing transcriptomic sequences 3:307
 conservation of residues for structure and function 3:255
- de novo transcriptome assembly 3:309–310
- domain analysis 3:255
- GO 3:317–318
- metabolic pathways analysis 3:316–317
- motifs and patterns 3:255
- nucleic acid sequencing 3:293
- protein data analysis 3:310–311
- protein family analysis 3:255
- Sequence analysis method (SAM) 2:10–11
- Sequence and Phenotype Integration Exchange (SPHINX) 3:384
- sequence motifs (SM) 2:913
- Sequence Order Independent Profile-Profile Alignments (SOIPPA) 2:652
- sequence read archive (SRA) 2:335, 2:344, 2:347, 2:1142, 3:150, 3:165, 3:357–358, 3:425
- Sequence Retrieval System (SRS) 1:1092, 1:1092–1093
- sequence self-alignment (SSA) 2:214
- sequence signatures 3:954–955
- sequence-based methods 2:145–146, 2:146F, 2:587–588, 2:965
 prediction methods 3:669–671
 for recognising domains in proteins 2:473–474
- sequence/sequencing
 approach to gene expression profiling 2:375–377
 comparison 1:223, 1:224T
 composition 3:325
 DNA 3:323–325
 databases 3:29–30, 3:30T
 evolution 2:703–704
 features 3:682
 generation 3:158
 homology 2:850
 logos 3:335
 methods based on sequence similarity 2:12
 mRNA 2:360
 neighborhood 3:682
 objects 1:709
 profiles and motifs 2:12
 readout 2:144–145
 reads mapping 1:224, 1:225T
 RNA-seq data preprocessing 2:376–377
 to sequence base pairing 2:188
 sequence patterns, motifs and domains identification 3:332

- sequence-based approaches 3:1–3
- sequence-based bioinformatics 2:1142, 2:1146
- sequence-based predictors 2:841
- sequence-based screens 3:184
- technologies 1:22
- Sequences Annotated by Structure (SAS) 2:534
- sequencing by ligation (SBL) 3:434
- sequencing of psoralen crosslinked, ligated and selected hybrids (SPLASH) 2:580
- sequencing-by-synthesis (SBS) 2:179–180, 2:1151, 3:434–435, 3:834–835
- Sequenom MassARRAY platform 3:434
- sequential covering method 1:388–389, 1:390F
- sequential ensembles 1:460
- Sequential LLSimpute (LLSimpute) 1:473
- Sequential Window Acquisition of all Theoretical fragment ion spectra coupled to Tandem Mass Spectrometry (SWATH-MS) 3:873
- sequential window acquisition of all theoretical mass spectra (SWATH) 3:906
- SEQUEST database 2:87
- SeRenDIP 2:841
- serendipitous discovery 3:741
- serial analysis of gene expression (SAGE) 2:121–122, 2:372, 2:373, 2:376, 3:819–820, 3:820
 - bioinformatics packages 2:121–122
 - epigenomics 2:126–128
 - general tools implicated in functional genomics 2:129
 - miRNAs/transposons 2:124
 - mutations 2:122–124
 - SAGE300 2:121–122
 - SAGEnet catalogs tags 2:121–122
 - tools implicated in microarray 2:122T
- serial/parallel job processing 3:136
- serine-rich region (SRR) 2:144
- serine/arginine rich proteins (SR proteins) 2:222
- Sertoli cells 3:456
- serverless computing 2:1155
- Service Level Agreements (SLA) 1:257
- service model 1:236, 1:236–237, 1:257–258
 - cost 1:237
 - flexibility 1:237
 - maintenance 1:237
 - portability 1:237
 - scalability 1:237
 - security 1:237
- Service-Oriented Architecture (SOA) 1:163
- set 1:196
- setosa* 1:742
- SetupX 2:397–399
- Seven Bridges (pipeline tools) 3:139
- seven-transmembrane spanning receptors (TM7) 2:906
- severe acute respiratory syndrome (SARS) 3:241
- severe dengue (SD) 3:9, 3:10
- Sewall Wright effect 3:943
- Sexlethal (*sxl*) 3:968
- SFmap server 2:228
- Sfold program 3:569–570
- SH-corrected aLRT (SH-aLRT) 2:739
- shallow annotation problem 1:867
- shallow learning 1:641
- Shannon's information entropy 3:21
- SHAPE and mutational profiling (SHAPE-MaP) 2:580
- shape readout 2:144–145
- Shapeit 3:444
- shared ancestors 1:872–873
- shared epitope genes (SE genes) 3:730
- Shared Harvard Inner Ear Laboratory Database (SHIELD) 2:343
- shared IC 1:873
- shared information 1:873
- shared memory 1:164–165
- shared memory languages 1:215–216; *see also* distributed memory languages
 - Java 1:216
 - OpenMP 1:216
- shared memory multiprocessor (SMP) 1:162
- shared tasks 2:1104–1105
- shared-memory model 1:209–210
- Shell scripting, case study with 3:145
- SHH* gene 2:262–263
- Shi and Malik algorithm (SM algorithm) 1:443
- Shi-Malik Algorithm (SM Algorithm) 1:1043
- Shigella flexneri* 3:184
- shikimate kinase (SK) 3:658
- shikimic acid (SKM) 3:657
- short chain fatty acid (SCFAs) 3:200
- short inter-spersed nuclear elements (SINEs) 2:126
- short interfering RNAs. *see* Small interfering RNAs (siRNAs)
- short interspersed nuclear elements (SINEs) 3:305
- short linear motifs (SLiMs) 2:18, 3:1, 3:2–3, 3:105, 3:106
- Short Oligonucleotide Alignment Program (SOAP2) 3:161
- short tandem repeats (STR). *see* simple sequence repeats (SSR)
- short-chain dehydrogenase/reductase family member 4 (DHRs4) 3:845
- short-read technologies 3:454
- Short-time Fourier Transform (STFT) 1:481
- shortest path algorithms 1:943, 1:947
 - Bellman Ford algorithm 1:943–944
 - Dijkstra's algorithm 1:943
 - Floyd-Warshall algorithm 1:944–945
- shortest-paths based centralities 1:952
 - betweenness centrality 1:953
 - closeness centrality 1:952–953
 - eccentricity centrality 1:952
- shotgun lipidomics 2:894–895
- shotgun metagenomics 3:184, 3:186
 - annotation 3:187–188
 - binning 3:187
 - de novo assembly 3:186–187
 - pre-processing of sequence reads 3:186
- shotgun proteomics 2:89–90, 2:1066–1067
- SHOWBOX program 2:593
- SHRiMP 3:437
- shrink-hole effect 1:230
- shrinkage 3:366
- shuffle indexes 1:267
- Side Effect Resource (SIDER) 2:634, 2:1048, 2:1090
- siDirect 2:125
- Sig2GRN tool 1:1055
- sigmoids 1:641
 - function 1:390, 1:624
- sign test 1:702–703
- signal recognition particle (SRP) 2:54–55
- signal sequence prediction 2:54–55
- signal transduction 2:796, 2:856, 3:84
 - pathways 1:1048, 1:1056–1059
 - database comparison 1:1060T
 - databases 1:1059–1060
 - tools features comparison 1:1059T
 - visualization and analysis tools 1:1058–1059
- signal transduction networks (STNs) 1:1106, 2:856
 - computational modeling of STNs and applications 2:860–861
 - databases and enrichment analysis tools 2:858–860, 2:860T
 - Raf-MEK-ERK MAPK signaling pathway in cancer 2:856–857
 - structural motifs in 2:857–858
- signaling 2:866–867
 - gateway 1:1060
 - networks 1:917–918, 1:1018, 2:807
 - noise 1:747
 - pathways 1:1048–1049, 2:796, 2:801
 - databases 2:798–799, 2:799T
- SIGnaling Network Open Resource (SIGNOR) 1:1060
- signaling pathway impact analysis (SPIA) 2:803
- Signaling Pathway Integrated Knowledge Engine (SPIKE) 1:1060
- Signalink2 1:1060
- signed rank test 1:703
- Significance Analysis of Microarrays (SAM) 3:785, 3:817
- sigPath 2:799
- silencing RNAs. *see* Small interfering RNAs (siRNAs)
- Silhouette coefficient 1:444
- Sim Pack 1:886
- Sim Triplex model 3:247
- SIM4 2:199–200, 2:202
- SimBoolNet tool 1:1059
- SIMCA-P software 3:471
- SimCoal2 2:754
- simGIC 1:890, 1:909
- simian immunodeficiency virus (SIV) 2:985
- similarity
 - entity 1:874
 - layer 1:884
 - measure 1:871, 1:873–874
 - scoring function 1:23
 - similarity-based algorithms 2:182–183
 - similarity-based binning 3:187
 - similarity-based detection methods 3:412–413
- similarity ensemble approach (SEA) 2:623

- Similarity Library 1:881–882, 1:883–884
 simJC 1:890
 simLin 1:890
 simple graphs 1:923, 1:1086
 simple interaction format (SIF) 1:1027
 simple language model 2:1103–1104
 simple linear iterative clustering on
 multichannel microscopy with edge
 detection method (SLIC-MMED
 method) 2:993–994
 simple linear regression 1:421, 1:723–725
 Simple Linux Utility for Resource
 Management (SLURM) 3:136
 simple modular architecture research tool
 (SMART) 3:514
 simple motif 1:16, 1:18
 simple proteins 3:511
 simple random sampling 1:483
 without replacement 1:466
 with replacement 1:466
 simple sequence repeats (SSRs) 2:118–119,
 2:210
 Simple Storage Service (S3) 1:240, 1:257
 SIMPLEX 3:169
 Simplified Molecular Input Line Entry
 (SMILES) 2:605, 2:635–636
 Simulated annealing (SA) 1:78, 1:321,
 1:322, 1:322–323
 algorithm 1:1090
 energy optimization by 1:42–45
 Simulated Quantum Annealing (SQA) 1:323
 simulated tempering (ST) 2:553
 simulation environments and setups,
 standard for recording 2:887–888
 Simulation Experiment Description Markup
 Language (SED-ML) 2:887–888,
 2:889
 simulation models in biological layers 2:866
 GRN 2:867
 metabolism 2:866
 signaling 2:866–867
 simulation of reacting systems 1:751
 SINEDR 2:214
 single cancer-specific sources 2:341–343
 single cell analysis (SCA) 3:816
 single cell RNA sequencing (scRNA-Seq)
 3:140
 single cell sequencing 1:898, 2:974, 3:349
 single cell transcriptomic analysis 3:801;
 see also bulk transcriptomic analysis
 cell communication analysis 3:801–802
 cell type identification 3:801
 differential expression analysis 3:801
 Single Cell-2-Cell Communicator (SC2CC)
 3:801–802, 3:802F
 single chain fragment variable (scFv) 2:933
 single classification threshold
 composite performance measures 1:549–
 550
 elementary performance measures 1:548–
 549
 performance measured based on 1:547–
 549
 single factor differential expression analysis
 3:796
 2-level single factor DEG analysis 3:796
 multi-level single factor DEG analysis 3:796
 single hold-out method 1:542–543
 single hold-out random subsampling 1:542–
 543
 single instruction multiple data (SIMD)
 1:161
 single instruction single data (SISD) 1:161
 single layer Perceptron (SLP) 1:391, 1:612–
 614, 1:617F
 Perceptron learning rule 1:613–614
 single neuron Perceptron 1:613F
 single linkage 1:350
 single microarray based normalization 3:816
 single molecule level models 1:916, 1:1053–
 1054, 1:1054
 single molecule real-time (SMRT) 2:180,
 2:330, 3:204
 long reads 3:436
 sequencing 3:294
 single molecule sequencing (SMS) 3:436
 single nucleotide polymorphism database
 (dbSNP) 3:169
 single nucleotide polymorphisms (SNPs)
 1:128, 1:382, 1:483–484, 1:763,
 1:764, 1:898, 1:1093, 1:1104–1105,
 2:1, 2:118, 2:118–119, 2:123, 2:171,
 2:178–179, 2:235–236, 2:242–243,
 2:332, 2:359, 2:946, 2:1047–1048,
 2:1051, 3:115, 3:145, 3:166–167,
 3:176, 3:296, 3:299, 3:324, 3:381,
 3:388, 3:428, 3:432, 3:441
 analysis of SNP data generated by
 sequencers 3:436–437
 ARMS 3:432
 AS-SBE 3:433–434
 calling 3:179, 3:437–438
 heteroduplex analysis 3:433
 HRM curve analysis 3:432–433
 identification 1:223, 1:224T
 invader assay 3:433
 mid and high-throughput technologies
 3:434
 OLA 3:434
 RFLP 3:433
 SNP-microarray 1:1047–1048
 SSCP 3:433
 tools implicated in analysis 2:119
 single nucleotide variants (SNVs) 2:183–
 184, 2:186, 2:358, 2:359, 3:1, 3:144,
 3:168, 3:364, 3:380, 3:388–389
 web servers as tool for predicting 3:5
 single particle tracking 1:747
 Single Program Multiple Data (SPMD) 1:217
 single reaction monitoring (SRM) 3:906–907
 single reactor in sample preparation 3:912–
 914
 single sequence approach 3:670
 “single species, multiple genes” approach
 2:285
 single template, homology modelling using
 3:260–261
 Single Value Decomposition method (SVD
 method) 1:473
 single value imputation 1:465
 single-cell ionic models 2:901–902
 single-cell RNA-seq (scRNA-seq) 2:369,
 2:392, 3:792
 single-cell transcriptomics technology 2:53
 single-model and quasi-single-model MQAP
 3:593–594, 3:597
 single-model programs 3:588
 single-molecule real-time sequencing
 (SMRT) 3:348–349, 3:436
 single-molecule sequencing technologies
 3:581–582
 single-nucleotide addition (SNA) 3:435–436
 single-particle cryo-EM 3:521
 single-particle tracking (SPT) 2:868
 single-photon emission computed
 tomography (SPECT) 2:1028
 single-strand conformation polymorphism
 (SSCP) 3:203, 3:433
 single-stranded DNA (ssDNA) 3:433
 single-stranded nucleicacids 3:819–820
 singular enrichment analysis (SEA) 1:896,
 3:219
 singular value decomposition (SVD) 1:480,
 1:488, 1:881–882, 2:429
 SIOMICS 3:449–450
 Site-Directed Mutate (SDM) 3:3
 situated (inter)action 1:312
 SKAT 3:394
 SKAT-O 2:190
 skull classification 2:1006
 Skyline software 3:874, 3:875F, 3:876F,
 3:877F, 3:878F, 3:878, 3:880F, 3:880
 spectral library using 3:875
 tool 2:86, 2:91
 slack variables 1:503–504
 SLAM program 2:203
 slave nodes 1:222
 SLiMs. *see* eukaryotic linear motifs (ELMs)
 Slow Growth (SG) 3:3
 SMAD4 1:903
 Small Angle Scattering Biological Databank
 (SASBDB) 2:465–466
 small angle scattering of neutrons (SANS)
 3:586
 small angle x-ray scattering (SAXS) 2:289,
 2:514, 2:845, 3:586
 small cytoplasmic RNA (scRNA) 3:235
 small interfering RNAs (siRNAs) 2:124,
 2:125, 2:165, 3:9, 3:33–34, 3:47,
 3:94, 3:234, 3:235
 small molecule drug design 3:741
 advantages 3:741
 approaches for drug discovery/designing
 3:742
 CADD 3:749–751
 characteristics of drug 3:741–742
 drug 3:741
 drug discovery 3:742
 Small Molecule Pathway Database (SMPDB)
 2:420, 2:799, 2:800, 3:470
 small non-coding RNA (snRNA) 2:125
 small nuclear RNAs (snRNAs) 2:165, 2:325,
 3:33–34, 3:230, 3:235, 3:236
 small nuclear RNPs (snRNPs) 2:221
 small nucleolar RNAs (snoRNA) 2:165,
 2:167, 2:326, 3:33–34, 3:230, 3:235,
 3:235–236

- small parsimony problem 2:708
- small regulatory RNA (sRNAs) 3:535
- small RNA 3:233–234;
 see also medium non-coding RNA
 miRNA 3:234
 piRNA 3:234–235
 siRNA 3:235
- small subunit (SSU) 3:185
 16S SSU RNA 3:15
- small-interfering RNA (siRNA) 3:230
- small-molecule pocket interactions 2:543–544
- small-world model 1:964, 1:970, 1:971F
- small-world network (SMN) 1:90
- small-world-effect 2:823
- Smith-Waterman algorithm (SW algorithm)
 1:23, 1:113–114, 2:270, 2:1144–1145, 3:237
- Smith-Waterman-Gotoh local alignment algorithm 3:981
- Smoking Effects on Gene Expression (SEGEL) 2:1051
- smooth support vector machine (SSVM) 3:3
- snakemake (pipeline tools) 3:137
- SNOMED Clinical Terms (SNOMED CT) 1:1104
- snoSeeker program 3:236
- snowball sampling (SBS) 1:90, 1:91
- SNPdryad* 3:3
- SNPeff 3:299
- SnpEff* 3:382
- SNPs3D 1:911
- SNPsim 2:754
- SOAPdenovo 3:304
- SOAPDENVO-TRANS 2:358, 3:310
- social (inter)action 1:312–313
- social networks 1:81, 1:89
 analysis tools 1:984
- sockets 1:218
- SoDA 2:946
- sodium (Na) 1:634
- sodium borohydride 3:342–344
- soft clustering 1:966, 2:1032
- soft confidence-weighted learning methods (SCW learning methods) 3:419
- soft margin 1:395–396, 1:503–504
- Soft Margin Classifier 1:503–504, 1:504
- Software as a Service (SaaS) 1:160–161, 1:163, 1:172–174, 1:236, 1:240, 1:249, 1:250, 1:258
 installations 2:1153
- software development kit (SDK) 1:160–161
- software standardization 3:69
- Solexa Genome Analyzer 3:435
- SOLiD systems 3:434
- solid-phase extraction (SPE) 3:464
- solid-phase micro-extraction (SPME) 3:464
- solubility 2:30
- solute carrier transporters (SLC transporters) 3:379
- solution modifiers 1:800, 1:804
- solution scattering databases 2:465–466
- solvent accessibility 3:682
- solvent accessible surface (SAS) 2:524, 2:524–525, 2:528
- solvent excluded surface (SES) 2:524, 2:525
- solvent-accessible surface area (SASA) 2:841, 3:657–658
- SomamiR 2:1081
- SomamiR 2.0 2:1081
- somatic cells 2:289
- somatic hypermutation (SHM) 2:940, 2:942–943, 2:972
- somatic mutations 2:860–861
- somatic variant
 calling 3:168–169
 detection in cancer genomes 3:392–393
- son of sevenless (SOS) 2:857
- sorafenib 3:744T–748
- S-rensen–Dicesimilarity coefficient. *see* F_1 score
- SorghumFDB database 2:347
- Sorting Intolerant From Tolerant (SIFT) 2:123, 2:205, 3:3
- source knowledge base (SKB) 1:1095
- Southeast Asia, dengue across eight countries in 3:975–976
- Sox2 proteins 2:144–145
- Soybean (*Glycine max* L. (Merrill)) 3:428
- soybean mosaic virus (SMV) 3:428
- SP-score 3:667
- SP2TFBS database 2:119
- space complexity 1:1–2
- space filling model 2:524
- space-decomposition 2:1146
- “space-homogeneous” models 2:277
- SPAdes 3:304
- spanning trees 1:948
- Spark 1:248–249
 Core 1:248
 integration 2:1158
 SQL 1:249
 Streaming 1:249
- SPARQL language 1:800–801, 1:835, 2:98–99
 dataset clause and named graphs 1:801
 query clause 1:801–802
 to query ontologies 1:804–806
 query types 1:801
 semantics 1:803–804
 solution modifiers 1:804
 SPARQL 1.1 standard 1:800
- sparse graphs 1:931F, 1:931
- sparse logistic regression 1:731
- sparse non-negative matrix factorization algorithm (sparse NMF algorithm) 3:367
- sparsity coefficient 1:960
- spatial data types (SDTs) 2:99
- spatial databases 2:99
- spatial simulation techniques 2:868–869, 2:869F
- spatio-temporal gene expression analysis 3:456
- SPDP-mode 2:1146
- Spearman correlation 2:157
- Spearman’s rank correlation coefficient 1:709
- SpecDB 1:333
- specialised browsers 2:253
- specialised databases 2:103
- speciation 3:943
 events 3:970
 genetic drift 3:943–944
 mutation 3:944, 3:945F
 natural selection 3:943
 nested and overlapping genes and association with 3:967–968
- species
 keystone 2:1133–1134
 in SBML 2:885–886
 species-specific metabolite databases 2:403–404
 species+database 2:1111
- specific amino acid within protein 2:20
- specific organism databases 2:252
- specificity (SP) 1:686, 2:955, 3:682–683
- Specimen ImageDatabase (SID) 2:1011–1012, 2:1014F
- SPECTRA network 1:1025
- spectral approaches 1:442
- spectral clustering algorithm (SC algorithm) 1:1042–1043, 1:1042F
- spectral counting technique 2:89
- spectral database for organic compounds (SDBS) 2:403, 3:470
- spectral identification and library generation 3:874–875
 spectral library using Skyline 3:875
- spectral libraries search 3:862
- Spectral Repeat Finder (SRF) 2:126
- SpectraST 3:862, 3:874
- spectrum identification and library generation 3:874–875
 spectral library using Skyline 3:875
- spectrum kernel 1:498
- spectrum-to-spectrum matches (SSMs) 2:1069
- speed and performance in clustering (SPiCi) 1:1041, 1:1043
- spermatogenesis 3:452
- spermatogonia (SG) 3:456–457
- SPFP-mode 2:1146
- Sphere Grinder Score (SG Score) 1:71
- SPHGEN program 2:593
- SPiCi method 1:979–980
- SPIDEY 2:199–200, 2:202
- spike in controls 3:345–346
- spin 1:562
 magnetization 2:515
- spin-spin coupling 2:427
- spindle checkpoint 1:833–834, 1:835
- SPINE method 3:519
- SPINE-X algorithm 3:519
- Spirocheta Americana* 3:945
- Splice Complexity 2:197
- “Splice graph” 2:224
- spliced alignment 2:202
- spliced leader sequence (SL sequence) 2:200
- spliced mapping tools 3:455
- SpliceGrapher 2:204
- spliceosome 2:221–222
- spliceosome apparatus (SA) 2:288
- SpliceR 2:227
- SpliceSeq 2:226
- SpliceTrap 2:226
- splicing 2:221, 2:222
 code prediction 2:228, 2:229T

- splicing (*continued*)
 - graphs 2:204
 - index. *see* splicing efficiency (SE)
 - mechanism 2:221–222
 - score. *see* splicing efficiency (SE)
- splicing efficiency (SE) 2:224
- splicing prediction and concentration
 - estimation (SPACE) 2:224
- splicing regulatory elements (SREs) 2:228
- SplicingViewer 2:226–227
- split read (SR) 2:185, 2:359
- SplitsTree 3:370
- spoofing attack 1:270
- SPOT-Struc 3:667
- SpotFinder 1:334
- spotted arrays 1:128
- SPRINT-CBH 3:673–674
- SPXP-mode 2:1146
- SQL2Mongo 2:1068
- SRCPred 3:678
- SROOGLE 2:228
- sry-related high mobility group box 4 (Sox4) 3:844–845
- SSAlign method 1:1005–1006
- SST algorithm 3:519
- SSU rRNA
 - from prokaryotes, eukaryotes, and organelles
 - high-resolution ribosome structures 3:13–14
 - multiple sequence alignment and phylogenetic tree construction 3:14–15
 - secondary structure diagram of 16S SSU RNA 3:15
 - secondary structure predictions 3:11–12
 - sequence studies 3:11
 - 3D structure 3:15–16
 - studies 3:12–13
- stability and free binding energies of
 - protein-ligand complexes 3:659–660
- Stabilized matrix method (SMM) 3:43
- Stable Isotope Labelling with Amino Acid in Cell Culture (SILAC) 2:22, 2:1070, 3:872
- Stackdriver Logging 1:242
- Stackdriver Monitoring 1:242
- stacked autoencoder 1:644
- stacking 1:519, 1:536
 - algorithm 1:537–538, 1:538F
 - method 1:524
- STAN 3:809
 - state identification using 3:810, 3:811F
- standard deviation (SD) 1:685
- Standard Deviation of Error of Prediction (SDEP) 2:667
- Standard operating procedure (SOP) 2:10
- standard upper ontologies (SUO). *see* top-level ontologies
- standardized markup language (SBGN-ML) 2:887
- standardizing data format and vocabulary 1:988–991
- Stanford CoreNLP 1:604–605
- Stanford Network Analysis Platform (SNAP) 1:318
- Stanford Research Institute Problem Solver (STRIPS) 1:290
- Staphylococcus aureus* 2:161, 3:472, 3:926
- Staphylococcus pasteurii* 3:942
- STAR algorithm 2:214
- STAR software 3:357
- star-tree paradox 2:739
- stars 1:930
- STATA software 1:689
- state-of-the-art genome assembly tools 3:304
- state-of-the-art MoRFs predictors 3:122, 3:127T
- state-of-the-art multiple network alignment algorithm 2:819
- “state-transition” steps 2:37
- static data integration, PPIN and 1:983
- static modeling 2:864–865
- static network models 2:157–158, 2:159
 - BN 2:158–159
 - coexpression networks and relevance networks 2:157–158
 - regression based approaches 2:158
- static partial charges 2:561
- stationarity 1:748
- stationary distribution of multi-allele neutral WF model 2:16–18
- “statistical alignment” 2:277–278
- statistical analysis 1:648–651, 1:649F, 2:681, 3:188, 3:206, 3:470
 - collection, organization and visualization of data 1:649–651
 - descriptive measures 1:651–654, 1:652T
- statistical analysis system (SAS) 1:689
- statistical classification model 1:686
- statistical computing and graphics 1:199
- statistical data analysis 2:419–420
 - further data analysis 2:420
- statistical data integration 1:1096–1097
- Statistical genetics 2:189
- statistical hypothesis 1:691
 - test 1:693
- statistical inference 1:738
 - in HMMs 1:757
 - techniques 1:698
 - experimental data in one sample design 1:698F
 - hypothesis testing procedures 1:699T
 - nonparametric test 1:702–703
 - parametric tests 1:698–699
- statistical measures 1:96–97, 1:674–676
 - central tendency measurement 1:674–676
 - dispersion measurement 1:676–677
 - symmetry measurement 1:677–681
- statistical methods 2:428–430
 - classification algorithms 2:430
 - dimension reduction techniques 2:429–430
 - molecular epidemiology 2:430–431
- statistical modeling 3:631, 3:632F
 - batch effect 3:815–816
 - enrichment analysis 3:816
 - fold change 3:815
 - HTS 3:814–815
 - microarray 3:814
 - multiple comparisons 3:815
 - normalization 3:815
- qPCR 3:816
- SCA 3:816
- statistical package for the social sciences (SPSS) 1:689
- statistical significance of sequence alignment 3:315
- statistical software 1:689
- statistical TOCSY (STOCSY) 2:431
- statistical unit 1:648–649
- Statistical-Algorithmic Method for Biclusteral Analysis (SAMBA) 1:1041
- statistical-based outliers 1:456–457
- statistics 1:651, 1:672, 1:674, 1:738
- statistics-based functions 1:63, 1:64, 1:465–466, 3:274
- steepest descent method. *see* gradient descent method
- Steiner tree approaches 2:807
- stemming 1:577
 - task 1:562
- stepwise methods 1:492
- stereochemical constraints 3:634
- stereochemical parameters 2:517F
- Stevens–Johnson syn-drome (SJS) 3:384
- STKE 2:799
- STM 1:96
- stochastic approaches to modeling 2:791
- stochastic context free grammar (SCFG) 2:577
- Stochastic Context-Free Grammar PCFG (SCFG). *see* Probabilistic Context-Free Grammar (PCFG)
- stochastic differential equations (SDE) 1:747–748
 - methods for 1:750–751
- stochastic differential equations (SDE) 2:920–921
- stochastic gradient descent (SGD) 1:640
 - learning methods 3:419
- stochastic methods
 - GAs 1:323–326
 - global optimization 1:321–322
 - SA 1:322–323
 - stochastic optimization algorithms 1:322
- stochastic models 2:577
- stochastic optimization algorithms 1:322
- stochastic processes 1:747–748, 1:748, 1:751, 1:753–755
 - continuity 1:748
 - Gaussianity 1:748
 - independence process 1:748
 - Markov process 1:748, 1:748–749
 - methodologies for numerical investigation 1:750–751
 - stationarity 1:748
- stochastic proximity embedding (SPE) 2:680
- stochastic sampling 2:581
- stochastic simulation 2:865–866
- stochastic simulation algorithm (SSA) 1:751
- Stochastic Tunneling approach (STUN approach) 1:323
- stochastic variation operators 1:775–776
- stochasticity 1:747, 2:791
- stoichiometric coefficients of metabolic reactions 2:439
- stoichiometric matrix 2:439–440

- stoichiometric models
 flux balance analysis 2:440
 GEMs 2:440–441
 quasi steady state assumption 2:440
 range of optimal solutions 2:440
 stoichiometric matrix 2:439–440
 StochSim. *see* stochastic simulation
 algorithm (SSA)
 Storage Controller (SC) 1:240, 1:245
 straight line drawing 1:1087, 1:1088F
 strand-specific RNA sequencing (ssRNA-seq)
 3:843
 strands and sheets 2:489
 strategic agents 1:289
 stratified random sampling 1:543
 stratified sampling 1:466, 1:483
 stream sockets 1:218
Streptomyces coelicolor 3:465
 streptomycin 3:569
 streptomycin aptamer 3:571
 strict clock models 2:722
 strict consensus tree 2:740–741
 STRIDE algorithm 2:490, 3:519
 STRING 2:851, 2:853
 String graph 3:304
 string matching 1:16
 STRING software 2:91
 string-based approaches 1:17
 strings and binders switch model (SBS
 model) 2:303–304, 2:303F
 strings and sequences 1:15
 approaches to motif search 1:16–17, 1:17–
 18, 1:18
 assessing motif relevance 1:18–19
 matches and repetitions 1:15–16
 motifs 1:16
 strings and sequences for representing
 biomolecules 1:1102
 strong nucleosomes 2:310, 2:312T–313
 structural bioinformatics 1:77, 1:110–111,
 2:1142
 structural biology 3:586
 structural classification of proteins (SCOP)
 1:112, 1:261, 1:561–562, 2:102,
 2:467, 2:472T, 3:513
 database 1:39, 3:32
 integration of SCOP and CATH resources
 2:478
 SCOP2 database 1:39, 2:102, 2:472T, 3:33
 Structural Classification of RNA (SCOR)
 3:539–540, 3:540–541
 structural fluxes 2:442
 structural genomics 3:722
 analysis and assessment 3:725
Burkholderia pseudomallei 3:726–727
 extent and realities of structural genomics
 3:722–723, 3:723F
Glaciozyma antarctica 3:727
 systems and/or applications 3:723–725,
 3:724T
 structural heterogeneity 1:1096
 structural variability in protein families
 3:612–613
 structural variants (SVs) 2:183–184, 3:168,
 3:380
 calling (SV calling) 2:185
 structural variations (SVs) 2:358, 2:359–360,
 2:1080, 3:116
 structurally resolved TM domains databases
 2:47
 structurally similar groups (SSGs) 2:478–479
 structure activity relationship 3:493–495
 proposed method to assessing 3:495
 relation between structure and biological
 activities 3:496–499
 selection of statistically significant
 relationships 3:495–496
 structure always affects function 1:950
 structure based design 2:613
 structure based model (SBM) 2:143
 structure database using 3D motif query
 3:624–625
 structure databases for proteins (SCOP)
 2:446
 and complexes 2:446
 structure group-biological activity pairs (SG-
 BA pairs) 3:496
 structure groups (SGs) 3:495
 structure integration with function,
 taxonomy and sequence (SIFTS)
 2:463–464
 structure learning 2:159
 structure optimization 3:260
 structure quality analysis. *see* structure
 quality assessment
 structure quality assessment 3:257, 3:586
 analysis and assessment 3:598–599
 consensus MQAP 3:599–600
 single-model and quasi-single-model
 MQAP 3:599
 validation against reference structure
 3:598–599
 applications 3:588–592
 background/fundamentals 3:587–588
 structure refinement 3:257–260
 structure similarity, methods based on 2:11–
 12, 2:11T
 STRUCTURE software 3:367
 structure validation 2:516, 3:263–264
 consistency with dictionary values 2:517–
 518
 electron microscopy 2:516–517
 NMR spectroscopy 2:516
 X-ray crystallography 2:516
 structure-activity relationship study (SAR
 study) 2:683
 structure-based bioinformatics 2:1142
 structure-based computational engineering
 of therapeutic antibody 3:634–635
 structure-based design 1:78
 structure-based drug design (SBDD) 2:585,
 2:677, 3:273, 3:279, 3:279–280
 binding site identification 3:274
 case study 3:276–277
 analysis 3:278–279
 grid maps preparation and generation
 3:277
 ligand preparation 3:276–277
 molecular docking 3:277–278
 receptor preparation 3:277
 and discovery studies 3:658–659
 drug target selection 3:273
 evaluation of drug target structures 3:273–
 274
 lead compound optimization 3:274–275
 molecular docking 3:274
 practical applications in pharmaceutical
 industry 3:275
 workflow 3:273F
 structure-based methods 2:145–146, 2:146F,
 2:147, 2:588, 2:954–955
 for recognising domains in proteins 2:473–
 474
 algorithms used in CATH Update
 protocol 2:475T
 structure-based pharmacophore modeling
 2:680–681
 structure-based prediction methods 3:667–
 669
 structure-based sequence alignment (SSA)
 3:257
 structure-based virtual high-throughput
 screening (SB-vHTS) 3:752
 structure-based virtual screening (SBVS)
 2:595–597, 2:688, 3:752–753,
 3:755–757
 potential PM II inhibitors lead molecules
 2:596F
 predicted PM II lead compounds 2:595T
 structure-optimized HMMs (soHMMs) 2:913
 structure-based drug discovery 3:752
 ADMET modelling 3:753
 fragment-based lead discovery 3:753
 SB-vHTS 3:752
 in silico structure-based lead optimization
 3:753
 structure-based virtual screening 3:752–
 753
 structured motif 1:16, 1:18
 Structured Query Language (SQL) 1:800,
 1:1096, 2:96, 2:1063, 2:1073
 language elements 2:1074T
 structure–stability relationship 2:1133
 subcellular localization 1:1024
 subcellular localization site
 prediction methods 2:54
 prediction scheme 2:53–54
 subgraph centrality 1:952
 subgraph isomorphism problem 1:938
 subject-specific resources 2:468
 subnetwork 1:84
 subring skeleton profiling (SRS profiling)
 3:490–491
 in alkaloids 3:491–493
 subsequence 1:15
 subset linear regression 1:727
 subset selection 1:491–493
 subspace clustering method 1:1040–1041
 subspace-based outliers 1:459–460
 Substance ID (SID) 2:630–631
 substitution 2:712–713
 matrix 1:27–28, 1:39–42, 3:314–315
 rate 2:7–8
 substrate recognition 3:732–734, 3:733F
 substring 1:15
 substructure search 2:641–642
 suffix/prefix trie 3:437

- Sugar-Lectin Interactions and DoCKing (SLICK) 3:673
- sugars 3:666
- folding 3:525
- suggested upper merged 50 ontology (SUMO) 1:809
- sulfides 3:941
- Sulfolobus solfataricus* 2:143
- Sum-of-Squared-Error (SSE) 1:439
- summarization (SUM) 1:578, 1:580, 1:580T
- Sundarban Mangrove ecosystem 2:1112–1113
- super-secondary structure 3:504, 3:512
- SuperCYP 3:384
- superfamily 3:966
- SUPERFAMILY database 3:32
- superfolder green fluorescent protein (sfGFP) 3:836
- SuperIDR 2:1023
- SuperMap 2:272–273
- supervised approaches 2:405
- supervised categorization 2:1103
- supervised data discretization 1:468–469
- supervised learning 1:300–301, 1:542, 3:2, 3:120–121;
 see also machine learning (ML)
- artificial neural networks 1:276, 1:345
- bagging and boosting 1:343–344
- bioinformatics 1:275
- decision tree 1:343
- performance evaluation 1:345–346
- random forest 1:344
- research and issues 1:346–347
- support vector machine 1:344–345
- support
- of association rule 1:360
- of itemset 1:359
- monotonicity property 1:360
- support vector (SV)
- application 3:5–6
- detection 2:185, 2:186F
- support vector classification (SVC) 3:2–3
- support vector clustering (SVC) 1:508–509, 1:515
- support vector machine (SVM) 1:276, 1:302, 1:303F, 1:342, 1:344–345, 1:393–395, 1:394F, 1:503, 1:514–515, 1:564, 1:589, 1:828–829, 2:13, 2:31, 2:48–49, 2:121, 2:125, 2:405, 2:665, 2:828, 2:842, 2:913, 2:916–917, 2:954, 2:955, 2:1004, 2:1085, 2:1103, 3:1, 3:2, 3:2–3, 3:42, 3:61, 3:107, 3:120–121, 3:129, 3:234, 3:244, 3:289, 3:594, 3:669, 3:681
- extension algorithms 1:508
- multi-class and one-versus-all classifiers 1:508
- SV regression 1:509–510
- SVC 1:508–509
- formulation for classification problems 1:504–505
- functional and geometric margin 1:505
- kernels in SVM algorithms 1:507–508
- non-separable data 1:507
- separable data 1:504–505
- SVM–dual optimization problem 1:506–507
- SVM–primal optimization problem 1:505–506
- linearly inseparable data 1:395–396
- linearly separable data 1:393–395
- maximum margin hyperplane 1:503
- multiclass 1:396–397
- soft margin and slack variables 1:503–504
- support vector machine–sequential minimal optimization algorithm (SVM-SMO algorithm) 2:146–147
- support vector machines with both linear kernels (linear-SVM) 3:63
- support vector regression (SVR) 1:474, 3:3, 3:611–612
- supramolecular RNA structures 3:535
- computational methods of constructing RNA supramolecules 3:540–543
- properties of ribonucleic acids 3:535–539
- techniques for constructing RNA nanostructures 3:539
- surface properties 2:653
- Surrogate Variable Analysis (SVA) 3:360, 3:796
- SVMQA 3:594
- swap-based aligners 1:1006
- sweet pea (*Lathyrus odoratus*) 2:171–172
- SWEt software 3:525
- SweetUnityMol 3:525
- Swift 1:249
- Swift-Turbine Compiler (STC) 1:249
- Swiss Institute of Bioinformatics (SIB) 1:112–113
- Swiss PDB viewer 1:38
- SWISS-MODEL 2:452, 3:253, 3:725, 3:725F, 3:726F
- repository 2:466, 2:1013–1015, 2:1017F
- server 1:38
- SwissADME 3:275
- SwissProt 3:681
- switch RNA 3:569
- Swoop tools 1:806–807
- SYFPEITHI (binding motifs) 2:933–934, 2:934F, 2:954, 2:965, 3:244
- SYFPEITHY 2:965
- symbols 1:15
- SymCurv 2:312T–313
- symmetric/symmetries 1:1088–1089
- function 1:496
- measurement 1:677–681
- rate matrix 2:707
- substructure score 1:998–999
- symmetrical exon 3:967
- synapsis 1:612, 1:634
- synchrotron radiation circular dichroism (SRCD) 3:520–521
- SYNLET 2:104–105, 2:106F
- synonymous SNVs. *see* single nucleotide polymorphisms (SNPs)
- syntactic
- categories. *see* Part-of-speech (POS)
- heterogeneity 1:1096
- layer 1:561–562
- parsing 1:564–566
- structure 1:577–578
- syntax 1:295
- tree 1:564
- “syntelogs” 2:261
- Syntenic Gene Prediction (SGP1) 2:203
- synteny 2:263
- blocks 3:296
- long-term conservation 2:263–264
- mappers 2:272
- mapping 2:269–270, 2:271
- synthetic biology 3:329–330
- Synthetic Biology Open Language (SBOL) 2:889
- synthetic long-reads 3:436
- synthetic riboregulators 3:568–569
- analysis and assessment 3:569–570
- future directions 3:571–572
- illustrative examples and case studies 3:571
- systems/applications 3:569
- synthetic riboswitches 3:569
- analysis and assessment 3:569–570
- future directions 3:571–572
- illustrative examples and case studies 3:571
- systems/applications 3:569
- synthon
- nomenclature 2:610
- operations on 2:610
- SysKid 2:105
- SYSTAT software 1:689
- system biology 2:830
- networks 1:1028
- System of Systems (SoSs) 1:309
- Systematic Evolution of Ligands by Exponential enrichment (SELEX) 3:568
- Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) 1:838
- systems biology 1:978, 3:74
- multi-omics data analysis 3:194
- pathways in 1:1071
- standards 2:884
- Systems Biology Graphical Notation (SBGN) 1:858, 1:1051, 1:1059–1060, 1:1065, 1:1076, 2:886–887, 2:887F, „ 2:889
- Systems Biology Markup Language (SBML) 1:142, 1:142–144, 1:858, 1:863F, 1:1027, 1:1058–1059, 1:1064–1065, 1:1076, 2:858–859, 2:885, 2:885–886, 2:889, 3:69, 3:477–478
- packages 1:144
- SBML squeezer 1:863
- SBML-SAT tool 1:1058–1059
- specification differences 1:144–145
- structure 1:143–144
- Systems Biology Ontology (SBO) 1:788, 1:858, 1:858–859, 1:859–860, 1:859F, 1:862–863, 1:864F, 2:888
- interoperability and quantitative aspects 1:861
- relationships 1:860–861
- technological details 1:861–862
- term fields 1:860, 1:860T
- use case 1:863–864
- Systems Biology Pathway Exchange (SBPAX) 1:863
- Systems Biology Results Mark-up Language (SBRML) 2:888
- systems-on-chip (SoC) 1:212, 2:1146

T

- T box riboswitch 3:548
 T cell epitopes 2:952–953, 2:952*F*, 2:953*F*
 prediction 2:955, 2:956*T*–961
 in vitro methods for identification 3:39
 T cell reactivity testing 3:40
 T cell receptor (TCR) 2:907, 2:940, 3:44
 T cytotoxic cells (Tc cells) 2:931
 T helper (Th cells) 2:931
 T lymphocytes 2:972
 T-cell epitopes 2:931
 prediction 2:910, 2:911*F*, 2:912*T*, 2:915
 of antigen processing and integration 2:914–915
 of peptide-MHC binding 2:910–914
 T-cell receptors (TCRs) 2:952–953, 3:39
 T-Coffee
 method 1:1011, 3:257
 suite 2:505–506
 tool 3:426
 t-distributed stochastic neighbor embedding
 analysis (t-SNE analysis) 3:456
 T-HOD 2:1049–1050
 t-test 1:693, 3:814
 T1Dbase 2:1051
 T4 DNA ligase 3:820
 ta-siRNAs-producing locus (TAS) 2:125–126
 tag SNPs 2:178–179, 2:236
 TAMBIS Ontology (TaO) 1:810
 tamoxifen (TAM) 3:384–385
 tandem affinity purification (TAP) 2:824, 2:1072
 tandem affinity purification-mass spectroscopy (TAP-MS) 2:805
 tandem duplication events in DNA sequences 3:965
 tandem mass spectrometry (MS/MS) 1:126–127, 2:399, 2:412–414, 2:413*F*, 2:635, 2:1066
 identification 2:86–88
 error rates assessment 2:88–89
 PTMs 2:87–88
 tandem mass tags (TMT) 2:1070
 Tandem RepeatFinder program 2:214, 2:215
 tandem repeats 2:210
 Taqman Assay 3:433
 Tarceva 3:754*F*
 TAREAN pipeline 2:215
 target graph 1:103
 target molecule (TM) 1:77, 2:610
 target sequence 2:497–498
 target template alignment 3:263
 target-based drug design 1:78
 “target-decoy” approach (TDA) 2:88, 2:1068
 target-template alignment 2:505–506, 3:256–257
 gaps
 in alignment 3:257
 in secondary structures 3:257
 residue conservation 3:257
 targeted metabolomics 2:414–415, 3:463;
 see also untargeted metabolomics
 data analysis 2:416–417
 data characteristics 2:415
 fluxomics 2:417
 metabolite quantification 2:415–416
 Targeted Node Processing (TNP) 2:817
 TargetFinder method 2:187
 targeting signal prediction
 chloroplastic targeting signal prediction 2:55–56
 mitochondrial targeting signal prediction 2:55
 prediction of nuclear localization signals and nuclear export signals 2:56
 signal sequence prediction 2:54–55
 signals 2:56–57
 TargetMine 1:911, 3:197
 TargetOrtho 2:285
 TasiRNAdb database 2:125–126
 TATA-binding protein 3:447
 TATAAT 2:183, 3:447
 Taverna Workflow Management System 2:401
 tax2tree 3:206
 Taxon Inuence Index (TII) 2:740
 taxon sampling 2:740
 taxonomic assignment 3:21–23, 3:23*F*, 3:188
 of OTUs and alignment 3:206
 taxonomic classification, creating online
 biodiversity repositories with
 benefits and risks of data online 2:1125
 comparison between databases 2:1125–1126
 differences between online databases in handling and displaying 2:1127*T*
 impact of issues to online taxonomy databases 2:1125
 ontology and 2:1126–1129
 taxonomy related issues 2:1124–1125
 Taxonomic Data Working Group (TDWG) 2:103
 taxonomic identification and phylogenetic profiling (TIPP) 3:188
 taxonomic/taxonomy 1:785, 2:1124
 databases. *see* biodiversity databases
 diversity 3:188
 hierarchy 2:1124
 information 3:187–188
 names to DNA barcodes 3:989–990
 BOLD identification engine 3:989–990, 3:991*F*
 GenBank BLAST 3:990, 3:991*F*
 reflect vocabulary properties 1:591
 tree in Newick format 3:188
 teaching-learning-based-optimization (TLBO) 3:324
 TEclass 2:214
 tectorna 3:539–540
 Tedna tool 2:215
 Teleost fish 3:947
 teleost genome duplication (TGD) 2:262
 telomerase guide RNA (tmRNA) 2:165
 telomerase RNA (TER) 3:230
 temperature factor. *see* B-factor
 temperature-jump spectroscopy 2:143
 template 3:206
 based methods 2:844–845
 based protein structure modelling 2:504–505
 DNA 3:435
 free modelling 3:252–253
 identification 3:255–256
 criteria for defining homologue 3:255–256
 and selection 3:263
 targets without homologues 3:256
 selection 3:256
 bounded ligands 3:256
 missing residues 3:256
 multiple templates 3:256
 resolution of template structure 3:256
 sequence similarity 3:256
 similarity in secondary structure 3:256
 template-based methods/modeling 1:261–262, 2:581–582, 3:252–253
 template-free methods 3:586–587
 template-free protein structure modelling 2:506
 template library of 523 (T523) 3:667
 template modeling score (TM score) 1:71, 1:71*T*, 3:592
 temporal motifs 2:814–815
 ten-eleven translocation (TET) 3:341
 TENET methodology 1:1059
 TensorFlow implementation 2:947
 TepiTool pipeline 2:965
 term frequency/inverse document frequency (tf/idf) 1:592
 Term Information Content (IC) 1:889
 term–document matrix 1:499
 terminal deoxynucleotidyl transferase (TdT) 2:942
 terminal inverted repeats (TIRs) 2:210–211
 terminal restriction fragment length polymorphism (T-RFLP) 3:203
 TErminology for Description of DYnamics (TEDDY) 2:889
 tertiary analysis 3:161–162
 tertiary interactions 3:546
 test statistic 1:692
 testing hypothesis 1:710–714
 tetracycline (TetR) 3:569, 3:958
 aptamer 3:571
Tetraodon nigroviridis 2:202
 text categorization 2:1103–1104, 2:1103*F*
 text chunking 1:564
 text classification 2:1099, 2:1101–1102
 Boolean queries and index structures 2:1102
 documents in vector space 2:1102–1103
 text categorization 2:1103–1104
 text disambiguation 1:578
 text mining (TM) 1:444, 1:575, 1:578–579, 1:580–581, 2:790, 2:826
 applications
 applications 3:997–998
 building blocks 3:997
 evaluation metrics 3:997–998, 3:998*F*
 sources of text 3:996–997
 for bioinformatics using biomedical literature 1:606–607, 1:607*T*
 approaches 1:603–604
 biomedical corpora 1:604
 biomedical text mining tools 1:605–606
 community challenges 1:607–608

- text mining (TM) (*continued*)
 NLP concepts 1:603
 text mining tasks 1:603
 DC 1:579
 and drug discovery 2:1090
 evaluation techniques and metrics 1:581
 IR 1:578–579, 1:579T
 NER 1:579–580, 1:580T
 number of publications in PubMed 1:575F, 1:576F
 PoS tagging 1:577
 resources for bioinformatics 2:1083, 2:1084–1085, 2:1090
 accessing information from database 2:1086–1088
 cancer models 2:1089–1090
 deconvolution of unstructured data 2:1084F
 document summarization 2:1085
 from experiments to articles to models 2:1085–1086
 IE 2:1085
 IR 2:1085
 ML 2:1085
 NLP 2:1085
 pattern recognition 2:1085
 personalized medicine 2:1088–1089
 trends in text mining 2:1083–1084
 sentence detection 1:576–577
 for simplifying biomedical literature browsing 1:581–583
 stemming and lemmatize 1:577
 SUM 1:580, 1:580T
 syntactic structure 1:577–578
 task 2:1099
 techniques 2:1083–1084, 2:1085
 text disambiguation 1:578
 text mining-based methods 3:288–289
 tokenization 1:577
 toolkits 1:604–605
 word-cloud creation 1:576F
 text normalization 1:561–562
 text retrieval conferences (TREC) 2:1105
 text summarization 1:569
 text-based query 2:1011
 BOLD systems 2:1015–1018
 CIL-CCDB 2:1013, 2:1016F
 Flybase 2:1011
 ModBase 2:1013, 2:1016F
 MonoDb 2:1012–1013, 2:1015F
 Orchidaceae of Central Africa 2:1012, 2:1015F
 PBI 2:1011, 2:1014F
 Rfam 2:1018
 SID 2:1011–1012, 2:1014F
 SWISS-MODEL Repository 2:1013–1015, 2:1017F
 Tez (frameworks) 1:248
 Tgex 1:253
 the Arabidopsis Information Resource (TAIR) 2:103
 The Cancer Genome Atlas (TCGA) 2:105–109, 2:107F, 2:108F, 2:108–109, 2:109, 2:342, 2:860–861, 2:1077, 2:1078–1079, 2:1079
 THE Economics of Ecosystems and Biodiversity (TEEB) 2:1119
 The Institute for Genomic Research (TIGR) 1:334
 The OpenMS Proteomics Pipeline (TOPP) 3:858
 thematic maps 3:6–7, 3:6F
 theophylline aptamer 3:571
 Theory of Evolution 1:775
 “theory of surprise” 1:19
 Therapeutic Target Database (TTD) 2:1048, 2:1090, 2:1094
 Therapeutically Applicable Research to Generate Effective Treatments (TARGET) 2:1077, 2:1078–1079
 therapeutics, miRNA in 2:126
 thermodynamic database for proteins and mutants 2:453
 thermodynamic hypothesis 2:452
 Thermodynamic Integration (TI) 3:3, 3:4
 thermodynamic scales 2:48
 thermophiles 3:945
 thermophilic proteins stability 2:453–454
Thermus brockianus 3:945
 thiamine pyrophosphate (TPP) 2:482
 thiamine pyrophosphate aptamer domain (TPP aptamer domain) 3:568
 third generation sequencing technologies 2:180
 third-generation sequencing 3:294
 “threaded block-set” 2:271
 threading 2:505, 3:253, 3:522–523
 threading-based homology modelling 3:261
 Threading Building Blocks (TBB) 1:209–210
 three billion base pairs (3 Gbp) 2:289
 three-dimension (3D)
 amino acid arrangements in protein structures
 analysis and assessment 3:622–624
 detection of 3D motif 3:625–627
 detection of Arg-Glu-Arg-Glu query motif 3:627–628
 fold and sequence independent motifs 3:621–622
 fold comparisons *vs.* amino acid 3d motif searching 3:620–621
 systems and/or applications 3:622, 3:623T
 base motifs 3:546
 database search 2:683
 Genome Browser 2:322
 layout 1:1021, 1:1090
 modeling 2:322
 motif detection in protein structure 3:625–627, 3:627F
 motif query 3:624–625
 points in 2:654–655
 PPIs 2:843F, 2:844
 protein structures 2:489–490
 space 3:252
 structure 2:488, 2:586, 2:688, 2:835–836, 3:620, 3:666–667, 3:722
 generators 2:607
 of genome 2:181–182
 interaction prediction 2:825
 prediction 3:509–510
 RNA structures 2:576
 of SSU rRNAs 3:15–16, 3:18F
 visualization 2:522–524
 3D-conformational search 2:680
 3D-QSAR pharmacophore models 2:680, 2:684
 3dRNA score 3:598
 3D electron microscopy (3DEM) 3:521
 3DFISH methods 2:301
 3Dmol.js 2:533
 threshold cycle (Ct) 3:831
 thresholding 3:128
 based method 2:997, 2:1034, 2:1036F
 based segmentation method 2:996–997
 segmentation 2:1028–1030
 value 2:1029
 thymidylate kinase inhibitor (TMPK inhibitor) 2:689–690
 thymidylate monophosphate analog (TMP analog) 2:689–690
 thymine 2:701, 3:425
 thymine DNA glycosylase (TDG) 3:341
 thyroid-stimulating hormone receptor (TSHR) 2:265
 Tianhe-2 1:213
 TIgGER 2:946
 tight-binding second moment approximation (TBSMA) 2:561
 TIGRFAM 3:31
 time 2:709
 complexity 1:1–2
 time to MRCA (TMRCA) 2:746–747
 time-of-flight (TOF) 2:428, 2:1064
 time-reversibility 2:704
 time-series analysis 1:199
 tipranavir 3:744T–748
 tissue annotation 1:1024–1025
 tissue plasminogen activator (tPA) 3:631
 tissue-specific AS in humans 2:223
 TMC114 binding
 energy components 3:714F
 I47V mutation effect on 3:710
 I50V and I50L/A71V mutations effect on 3:711–714
 molecular mechanism of drug resistance 3:715
 protein WT *vs.* mutant
 comparing apo form 3:712–714
 comparing complexed form 3:714–715, 3:717
 V32I and M46L mutations effect 3:715–717
 comparative schematic view 3:718F
 energy components 3:716F
 molecular mechanism of drug resistance 3:717–718
 TMC114 inhibitor 3:710F, 3:710
 TMEM45B 3:486
 tobacco etch virus (TEV) 3:636
 protease 3:636
 TOCSY technique 3:525
 tokenization 1:577
 task 1:561–562
 tokens 2:1100
 Toll like receptors signalling pathway 3:100
 Toll-like receptors (TLR) 2:906

- top-down MS 2:1069–1070
 top-down approaches 1:787, 2:801
 top-down proteomics 2:88
 TopHat2 software 3:357
 topological fingerprints 2:619, 2:620F
 topologically associated domains (TADs) 2:178, 2:295, 2:300, 2:320
 “topologically orthologous” functions 2:818
 topology 2:476, 2:727, 2:733
 cartoons or diagrams 2:521
 strings 2:522
 structural 2:520
 topomyosin I (TPM1) 3:842
 Toponome Imaging System (TIS) 1:1106
 topoorthology 2:269
 ToppGene 1:911
 TopPIC 2:88
 TOPS 2:521
 TTotal Correlation Spectroscopy (TOCSY) 2:427–428, 2:515
 total count normalization (TC normalization) 2:367
 total edge length minimization 1:1088–1089
 total integrated archive of short-read and array database (TIARA database) 2:344
 total internal reflection fluorescence (TIRF) 2:868
 total plasma protein mass 3:891–892
 total probability theorem 1:404
 totally drug-resistant tuberculosis (TDR-TB) 3:652–653
 Totally Intronic RNAs (TINs) 2:167–168
 TOUCHSTONE approach 1:67
 toxic epidermal necrolysis (TEN) 3:384
 Toxin and Toxin Target Database (T3DB) 2:1048
 traceback process 1:25
 traditional biology 2:796
 traditional clustering algorithms 1:1037
 Traditional Korean Medicine (TKM) 3:781
 traditional machine learning 1:638F, 1:639
 traditional methods of phylogenetic analysis 3:366
 traditional Sanger sequencing 3:293
 training datasets 1:274, 3:679–681
 TraML format 2:78–79
 Trans-ABYSS tool 3:310
 trans-acting lncRNAs 2:168
 trans-acting siRNAs (ta-siRNAs) 2:125–126
 trans-antisense transcription 3:842–844, 3:843F
 Trans-Omics approach 3:66
 Trans-Omics for Precision Medicine (TOPMed) 2:179
 trans-proteomic pipeline (TPP) 2:86, 3:858
 tools 3:874
trans-regulatory elements 2:155
 Transaldolase deficiency 1:702
 TRANSCoMpel database 2:120
 transcript quantification 2:377
 transcription 2:324, 3:504, 3:842–844, 3:843F
 transcription factor binding sites (TFBSs) 2:119, 2:134, 2:798, 3:324, 3:332, 3:338, 3:447, 3:458, 3:806
 comparison of DNA methylation at 3:359
 prediction 2:147
 TFBS/histone modification 3:809
 tools 3:809
 transcription factor databases (TFDB) 2:134, 2:135F, 2:136T–139
 aims 2:134
 collections of binding sites and motifs 2:140
 JASPAR 2:135–140
 other model organisms 2:140
 plant TFs 2:140
 on prokaryotic transcription factors 2:140
 TRANSFAC 2:134–135
 Transcription Factor Flexible Models (TFFMs) 2:135–140
 transcription factors (TFs) 2:134, 2:142, 2:147, 2:155, 2:311, 2:347, 2:796, 2:857–858, 2:861, 2:867, 3:97, 3:194, 3:447, 3:454, 3:806
 binding 2:311, 3:357
 to DNA partially unwrapped from histone octamer 2:311–314
 competition with histone octamer 2:311
 interactions 1:1099
 TF-mediated recruitment of ATP-dependent enzymes 2:314
 transcription start site (TSS) 1:272, 2:102, 2:134, 2:183, 2:186–187, 3:75, 3:348, 3:356, 3:808
 transcriptional activation domain (TA) 2:824
 transcriptional activities
 bioinformatics analysis/applications 3:836–839
 experimental techniques 3:830–836
 Transcriptional Regulatory Network (TRN) 3:76
 transcriptional regulatory networks. *see* gene regulatory networks (GRNs)
 transcriptional regulatory relationships
 unraveled by sentence-based text-mining (TRUSST) 2:798
 transcriptional start sites (TSSs) 3:447
 transcriptome 3:94, 3:819–820
 of cell differentiation 3:826–827
 cell specialization during development 3:825
 gene transcription during cell differentiation 3:825–826
 genes involving in cell differentiation 3:827
 data
 analysis 2:331–332
 HGT events using 3:957
 informatics 2:324
 making sense of 3:820–822
 mapping 2:330–331
 measurements 2:329–330
 output and normalization 2:331
 profiling techniques 3:95–96
 reconstruction 2:358
 sequencing platforms 2:329–330
 transcriptome analysis 3:117
 bulk 3:795–796
 differentially expressed genes and visualization 3:119–120
 DNA-chip data processing 3:117–118
 experimental design, data processing and quality assessment 3:792–795
 functional, pathway, and network analysis 3:120
 machine learning approaches in genomics 3:120–121
 profiling platforms 3:117
 RNA-sequencing data processing 3:118–119
 single cell 3:801
 Transcriptome Browser (TBrowse) 2:344
 transcriptome encyclopedia of rice (TENOR) 2:347
 transcriptome-phenotype correlations 3:819
 transcriptomics 1:111, 2:364, 2:373, 3:1, 3:194, 3:195, 3:294
 data generation 3:803
 databases 2:341, 2:348T–349
 animal 2:345–346
 human 2:341–343
 plant 2:346–347
 microarray data normalization 2:364–366
 RNA-seq data normalization 2:367
 sequences
 aligning RNA sequences 3:307–308
 gene counting and differential expression 3:308–309
 quality control of Rna-Seq 3:307
 transcripts per million (TPM) 3:357, 3:455, 3:833
 transducer strand. *see* switch RNA
 transduction 3:953
 TRANSFAC database 1:1100, 2:120, 2:134–135, 2:798, 3:448
 transfer RNA (tRNA) 2:325, 2:372, 2:567, 2:568, 2:575, 3:9, 3:230, 3:231, 3:235, 3:294
 abundance on codon bias 3:327–328
 transformation-based approach 1:577
 transgenic organisms 2:1096
 translating ribosome affinity purification sequencing (TRAP-seq) 2:333
 translation 1:823, 3:504, 3:510
 translational bioinformatics (TBI) 2:1058;
 see also bioinformatic(s)
 components 2:1046F
 computational tools implicated in 2:1053–1054
 databases 2:1059, 2:1060T
 CGDs 2:1059
 genetic epidemiology databases 2:1060–1061
 genomic medicine databases 2:1059
 pharmacogenomics databases 2:1059–1060
 datasets 2:1046–1048
 drug discovery 2:1051–1052
 drug interactions 2:1053
 integration and analysis of data for diagnosis and treatment 2:1049–1051
 patient-specific treatment regimes 2:1053
 translational initiation, codon ramp at 3:328–329
 transmembrane domains (TM domains) 2:38–39

- computational prediction of TM segments and topology 2:47–48
 - databases 2:47
 - functional characterization 2:50
 - 3D reconstruction 2:49–50
 - transmembrane proteins (TAP) 2:46, 2:46F, 2:47T, 2:49F, 2:49T, 2:50T, 3:42
 - prediction systems for TAP binding 3:42–44
 - TM segments and topology 2:47–48
 - transmembrane receptors 1:1057–1058
 - transmission electron microscopy (TEM) 2:515
 - transmission networks 1:81
 - Transparent Access to Multiple Bioinformatics Information Sources (TAMBIS) 1:810, 1:1094
 - TRANSP0 2:214
 - transporter associated with antigen processing (TAP) 2:952–953, 3:39
 - transposable elements (TEs) 2:126, 2:210–211, 2:347, 3:49
 - Transposome* program 2:215
 - Transposon insertion sequencing (TN-seq) 2:333
 - transposons in functional genomics 2:126
 - Transrate tool 3:310
 - trapezoidal estimators 1:556
 - Traveling Salesman Problem (TSP) 1:12, 1:324, 1:775–776, 1:946, 1:947–948, 1:948
 - traversal based sampling (TBS) 1:90
 - traversing/searching/sampling graphs
 - analysis and assessment 1:92–93
 - network 1:90
 - sampling approaches 1:90
 - TREC-Genomics 2:1105
 - TRED 1:1056, 2:214
 - tree 2:705
 - build 1:386–387
 - as graphs 2:728B
 - layout algorithm 1:1020, 1:1090
 - selection 2:705–707
 - space 2:742–743
 - species 2:347–350
 - tree-like evolution 2:704–705
 - tree-like hierarchy 2:390
 - ultrametric 2:731
 - tree evaluation and robustness testing
 - consensus methods 2:740–741
 - detection of conflicting phylogenetic signals 2:741–743
 - error sources 2:736–737
 - phylogenies applications 2:736
 - robust estimate 2:736
 - robustness 2:737–738
 - validating tree 2:736
 - Tree-Based Epistasis Association Mapping (TEAM) 2:118
 - TreeBASE 2:103
 - TreeMix method 3:369
 - TrEMBL 3:313
 - triacylglycerols (TAGs) 2:894
 - triads 1:85
 - trimethylsilyl-(²H₄)-propionate (TSP) 2:427
 - trimmed mean of M-values (TMM) 2:368, 2:381
 - trimming 2:355–356, 3:344–345
 - Trimmomatic 3:166, 3:296
 - trinitrobenzene (TNB) 3:637
 - trinitrotoluene (TNT) 3:637
 - trinity 2:215, 2:358
 - triose phosphate isomerase (TIM) 3:638
 - triple patterns 1:802
 - triplexes 3:505–506
 - Triticum aestivum* 2:347
 - tRNA adaptation index (tAI) 3:327–328, 3:848, 3:849
 - tRNA-derived stress-induced RNA (tiRNA) 3:234
 - trophic networks. *see* food webs
 - TRRUS database 1:1056
 - true hit rate (THR) 3:419
 - true negative rate (TNR) 1:548
 - true negatives (TNs) 2:160–161, 2:682, 3:332–333, 3:682–683, 3:997–998
 - true positive rate (TPR) 1:274, 1:548, 1:555, 2:160–161
 - true positives (TPs) 2:160–161, 2:682, 3:332–333, 3:682–683, 3:997–998
 - true-path rule 1:702, 1:827
 - truncated average Kolmogorov-Smirnov statistic (taKS) 1:555, 1:556
 - tryptophan 2:840
 - TSDfinder* 2:214
 - tuberculosis (TB) 3:652, 3:655T
 - tumor suppressor miRNAs 3:53–54
 - TumorPortal 2:1080
 - tumour necrosis factor “ (TNF-”) 2:931
 - tuples 1:195
 - turbidity 3:1
 - turbine runtime 1:249
 - TurboFold* 2 3:510
 - TurboKnot* 3:510
 - turkey 3:428
 - turns 3:517
 - Tute Genomics (platform) 1:253
 - Tversky contrast model 1:881
 - Tversky ratio-model 1:881
 - TWINSKAN program 2:203
 - two round WGD (2R WGD) 2:261
 - Two Stage-Grouped Sure Independence Screening (TS-GSIS) 2:118
 - two-dimension (2D)
 - isualization 2:520
 - NMR spectroscopy techniques 2:427–428
 - pharmacophore model 2:677
 - structure generators 2:607
 - 2D-gel electrophoresis 2:1063–1064
 - two-dimensional electrophoresis (2DE) 3:897
 - two-dimensional proton J-resolved NMR spectroscopy (2D Jres NMR) 3:465
 - two-sample Kolmogorov-Smirnov test 1:702
 - two-way ANOVA model 1:702
 - type 2 diabetes (T2D) 2:861, 3:17, 3:466–467
 - type 2 diabetes database (T2D-Db) 2:1051
 - type 2 diabetes genetic association database (T2DGADB) 2:1060–1061
 - Type 2 Diabetes Knowledge Portal 2:1051
 - type I error 1:694
 - Type II Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) 2:123
 - type VI secretion system (T6SS) 3:960–961
 - TYPE VII secretory system 3:105
 - type-1 diabetes (T1D) 2:1051
 - tyrosine 2:840
 - kinase inhibitors 3:749
- ## U
- U-matrix 3:3
 - U.S. National institutes of Health (NIH) 2:629
 - U.S. National Institutes of Standards and Technology (NIST) 2:1105
 - UBE2C 3:486
 - UBE2T 3:486
 - ubiquitin carboxy-terminal hydrolase L1 (Uchl1) 3:842
 - ubiquitination 2:15
 - sites prediction 2:19
 - UBViz (software tool) 1:1051
 - UCHIME algorithm 3:188
 - UCLUST program 3:188, 3:206
 - UCSC 3:300
 - Genome Browser 2:197–198, 2:199F, 2:253
 - Genome Browser Database 3:425
 - Xena 2:1081
 - UCSD-Nature Signaling Gateway 1:1060
 - UCSF Chimera 1:262, 2:530–531, 2:591–592
 - UCSF DOCK 2:591–592, 2:592–593
 - UDP-galactofuranose (UDP-Galf) 3:658
 - UDP-galactopyranose (UDP-Galp) 3:658
 - UDP-Galactopyranose Mutase (UGM) 3:658
 - UDP-glucuronosyltransferase Allele Nomenclature (UGT-allele database) 3:384
 - ultra-performance liquid chromatography (UPLC) 3:465
 - ultrametric phylogenetic methods 2:734
 - ultrametric trees 2:731
 - umbrella sampling 2:553–554
 - UMLS Concept Unique Identifiers (CUIs) 1:841
 - UMLS Terminology Services (UTS) 1:821
 - UN Environment World Conservation Monitoring Centre 2:1111
 - unaligned BAM 2:356
 - unassembled genomes, detection of repeats in 2:214–215
 - uncorrelated relaxed clocks 2:722
 - undirected graph 1:102, 1:922, 1:923F, 1:933F, 1:935, 1:1085–1086
 - undirected networks 1:85, 1:85F
 - unicellular organisms 1:916
 - UniChem 2:636
 - UniCon3D method 1:69, 1:70F
 - Unified Medical Language System (UMLS) 1:579, 1:580–581, 1:606, 1:700, 1:816–819, 1:817F, 1:838, 1:841, 3:62
 - availability 1:820–821
 - knowledge sources 1:819–820

lexical tools 1:820
 Metathesaurus 1:819, 1:819–820
 Semantic Network 1:819, 1:820
 SPECIALIST Lexicon 1:819, 1:820
 Unified Modeling Language (UML) 1:144–145, 1:333
 unified nucleic acid folding (UNAFold) 3:324–325, 3:509–510
 unified resource identifier (URI) 1:167–169
 uniform bend minimization 1:1088–1089
 uniform edge length minimization 1:1088–1089
 uniform resource identifiers (URIs) 1:790, 1:832, 1:1065, 2:1063, 2:1157
 uniform resource locators (URLs) 2:628
 Unifrac distance, multivariate analysis based on 3:23–25, 3:24F
 Unifrac metrics 3:206
 unigrams 2:1102
 UniProt 2:1104, 3:153, 3:313, 3:522
 UniProt archive (UniParc) 2:102, 3:313
 UniProt Gene Ontology Annotation (UniProt-GOA) 2:187
 UniProt KnowledgeBase (UniProtKB) 1:112–113, 1:868, 2:8, 2:8T, 2:47, 2:102, 3:30, 3:313
 UniProtKB-derived terms 1:835
 UniProtKB/SwissProt 3:153
 UniProtKB/TrEMBL 3:153
 UniProt reference clusters (UniRef) 2:102, 3:313
 unique molecular identifiers (UMI) 2:943, 2:973–974
 unitary pseudogenes 3:971
 UNITE database 3:185
 United inTackling Epidemic Dengue (UNITEDengue) 3:9
 UNited RESidue (UNRES) 1:63–64, 3:609–610
 United States Food and Drug Administration (FDA) 2:633, 2:1061, 2:1094, 2:1099, 3:379, 3:780, 3:996
 Pharmacogenomic Biomarkers 3:384
 United States Patent and Trademark Office (USPTO) 3:996
 united-residue force field (UNRES) 1:65
 univariate filter method 3:61
 univariate regression 1:722
 universal approximation theorem 1:627
 Backpropagation
 algorithm 1:630–631
 parameter setting 1:631
 weights update
 hidden neurons 1:627–630
 output neurons 1:627
 universal package management 2:1155
 Universal PBM Resource for Oligonucleotide Binding Evaluation (UniPROBE) 2:140
 Universal Protein Resource (UniProt) 2:8
 annotation process in 2:9–10
 automatic annotation 2:10–11
 manual annotation 2:10
 universally unique identifiers (UUIDs) 2:1153
 UNIX system 3:145–146

unlinkability 1:255
 unpaired biomarker identifier 3:61
 unpaired null hypothesis testing 3:60
 unrooting trees 2:732, 2:732–733
 Unseeded Region Growing (UsRG) 2:1030
 unsharp filters 2:995
 Unstructured Information Management Architecture (UIMA) 1:605
 unsupervised data discretization 1:469
 unsupervised learning 1:278, 1:301, 3:2, 3:120
 clustering 1:278
 unsupervised methods 1:456, 2:405, 3:470
 untargeted metabolomics 2:417, 3:463;
 see also targeted metabolomics
 data pre-processing 2:418
 experimental considerations 2:417
 metabolite detection 2:419
 metabolite identification 2:420–421
 pathway analysis 2:421
 statistical data analysis 2:419–420
 workflow 2:417–418, 2:417F
 untranslated regions (UTRs) 1:272, 2:125, 2:188, 2:227
 unweighted network 1:82F
 Unweighted Pair Group Method with Arithmetic Mean (UPGMA) 1:278, 1:1038, 2:734
 upper quartile normalization (UQ normalization) 2:367
 Upstream Activation Sequence (UAS) 2:824
 upstream regulators (URs) 1:1099
 uridine monophosphate (dUMP) 2:821
 US National Institute of Health (NIH) 1:815, 1:816
 use case 1:833–835, 1:844–845
 USEARCH algorithm 3:188, 3:204–205
 User Management System 2:397–399

V

V32I mutations 3:715–717
 vaccination 3:241
 vaccine design 3:109, 3:241
 vaccine informatics 3:241–242
 and future vaccines 3:247
 vaccine target discovery 3:242–243
 computational framework for 3:243–244, 3:245F
 immunoinformatics 3:244–246
 tools and databases for 3:246F
 vaccine informatics and future vaccines 3:247
 vaccine on infectious diseases and non-infectious diseases 3:246–247
 validation 3:338–339
 criteria 1:1043–1044
 external 1:1044–1045
 internal 1:1043–1044
 docking method 2:693–694
 indices 1:450–451
 external indices 1:450–451
 internal indices 1:451–452
 internal validation measures 1:452
 metrics for regression-based methods 2:667
 of models 2:665–667
 of pharmacophore models 2:681
 cost analysis 2:682
 Fischer's randomization test 2:682
 GH method 2:681–682
 ROC curve 2:682
 statistical analysis 2:681
 set 1:543
 in silico, *in vitro* and *in vivo*, approaches 2:1094–1096
 phenotypic screening 2:1095–1096
 tree 2:736
 validation against reference structure
 protein structures 3:588–592
 RNA structures 3:596
 van der Waals (VDW) 2:524, 3:712–714
 energy 2:842–844
 interaction 2:147, 2:1144, 2:1145–1146, 3:504
 radius 1:53
 surface 2:524, 2:528
 variance
 component test 3:115–116
 function 1:732
 stabilization 2:367
 variance stabilizing transformations (vst) 2:383–384
 variant
 annotation 3:116, 3:169
 calling 2:185, 2:1143, 3:168, 3:178–179, 3:179T, 3:299–300
 from RNA-seq data 3:798–799
 discovery 2:358–359
 filtration 2:185
 and prioritization 3:169
 functional analysis of 2:185–186
 identification 3:116
 repeats 3:966
 variant analysis 3:168
 germline variant calling 3:168
 somatic variant calling 3:168–169
 variant annotation 3:169
 variant calling 3:168
 variant filtration and prioritization 3:169
 Variant Annotation Tool (VAT) 1:253
 variant call format (VCF) 1:134–135, 2:358, 3:165
 variant effect predictor (VEP) 3:382, 3:393
 variants of unknown significance (VUS) 3:395–396
 VarScan tool 3:389–390
 VarScan2 approach 3:116, 3:393, 3:437
 Vector Alignment Search Tool (VAST) 2:472T
 Vector Alignment Search Tool Plus (VAST+) 2:472T
 vector space 2:1101–1102
 documents in 2:1102–1103
 kernel 1:499
 vector space model (VSM) 1:498, 1:589, 1:889
 vector syntax 1:200–201
 vector-similarity measure 2:1103
 VectorBase 3:108–109
 vectors number in training set 1:632
 vegetable species 2:347–350

- Velvet 3:304
 Venn diagrams 3:85F
 VERIFY_3D 2:518, 3:523, 3:588
 Verlet algorithm 3:607–608
versicolor 1:742
 Vertebrate Genome Annotation database (VEGA) 2:197–198
 vertebrate hemoglobin α and β chains 3:966
 vertex-weighted graph 1:923–924
 vertices. *see* nodes
 Vertnet 2:1112
 Very Important Pharmacogenes (VIP) 2:634, 3:382–383
 very low density lipoproteins (VLDL) 3:764
Vibrio harveyi CAIM 1792 3:428
 Vienna RNA Websuite 3:509–510
 vignettes 1:200
 VINA. *see* AutoDock Vina
 Vina-Carb (VC) 3:673
 VIPERdb 2:468
 viral metagenomics 3:189
 viral pathogens 3:105
 viral RNA 3:13
 Viral strains, dengue outbreaks in Singapore and Malaysia caused by 3:11
virginica 1:742
 VirHostNet 2.0 3:108–109
 virological and entomological perspective 3:977–978
 virtual atom molecular mechanics (VAMM) 2:561
 virtual chemical space (VCS) 2:604
 virtual clusters 1:210–211
 virtual docking 2:544, 3:70
 virtual high-throughput screening (vHTS) 3:751–752, 3:783, 3:784F, 3:784T
 virtual machine (VM) 1:241, 1:249, 1:257
 virtual molecular models in silico 2:607
 virtual paradigms 1:232–233
 virtual reality (VR) 2:1113
 virtual screening (VS) 2:544, 2:587, 2:683, 2:688, 2:694–696, 2:696–697, 3:742, 3:783–786, 3:929
 databases for 2:689
 molecular docking based VS 2:588
 Virtual servers 1:243
 virtual shared memory 1:215
 virtual tour database system of Rimba Ilmu Botanic Garden 2:1113–1115
 virtualization 1:236, 3:138
 of networks 1:236
 technology 1:160–161
 virus 3:928
 virus dynamics 2:985–986
 virus–host interactions (VHIs) 3:935
 VISTA 2:263
 Visual Language for Cloud (VL4Cloud) 1:249
 Visual Molecular Dynamics (VMD) 1:262, 2:529, 2:543
 visual studio 1:247–248
 visualization 1:1089, 2:101, 2:321–322, 2:334–335, 3:6–7, 3:7F, 3:116–117, 3:119–120, 3:129, 3:206
 of annotation 1:1025–1026
 of biomolecular structures and models 2:542–543
 model extraction and 1:250
 of specific SBO term 1:861–862
 standard for visualizing models 2:886–887
 visualization applet for RNA (VARNA) 2:569
 Viterbi algorithm 1:758, 2:196
 Vivar 1:254
 VNTRseek tool 2:215
 volatile organic compounds (VOCs) 3:493–494
 Volcano plot 3:456
 voltage-operated channels (VOCs) 2:796
 volume, variety, velocity and variability (4 'V's) 3:66–67
 volume, variety and velocity (3V-s) 2:1049
 volume visualization 3:6–7
 voluntary computing model 1:163
 voom method 2:383–384
 vorinostat (Zolinza) 2:585
 Voronoi diagram 3:595
 Voronoi diagram of atoms 2:525
- ## W
- Waikato Environment for Knowledge Analysis (WEKA) 1:305, 1:331
 Wald interval 1:557
 walk-based centralities 1:951
 Wallner in CASP11. *see* Pcomb
 Walrus Storage Controller (WS3) 1:245
 Wang-Landau method 2:553
 Warburg effect 3:476
 Ward's method 2:390
 warming velocity 2:1135
 warning signs (WS) 3:11
 WashU Epigenome Browser 2:322
 water (H₂O) 3:946
 bodies 3:1
 quality 3:1
 modeling 3:4
 water quality index (WQI) 3:3
 water quality modeling, ML methods in 3:2
 ANNs 3:2
 FL 3:3
 GA 3:3
 RF 3:2
 SVM 3:2–3
 watershed 2:1031F, 2:1031
 method 2:998
 region-based method 2:1035, 2:1038F
 watershed-based algorithms 2:1031
 Watson-Crick (WC) 3:504–505, 3:506F
 base triple 3:548–551
 edges 2:575
 Watterson estimator 2:14–15, 2:18–21
 wavelet transforms (WT) 1:481–482
 Web Atlas of Murine genomic Imprinting and Differential Expression (WAMIDEX) 2:128
 Web Graphics Library (WebGL) 2:533
 web integration 2:1158
 web programming in Java 1:207
 web servers as tool for predicting SNVs 3:5
 web services 3:292–293
 web-based license 1:821
 Web-of-Life Database 2:1134
 web-servers 3:520–521
 Web3DMol (2017) 2:533
 WebGeSTer DB 3:324
 Webmev 1:333–334
 WebMol (1997) 2:529–530
 WebODE tools 1:807
 WebRASP 3:597–598
 weighted co-expression network analysis (WGCNA) 3:67–68
 Weighted Gene Expression Tool (WeGET) 2:1048
 weighted graph 1:923–924, 1:1086, 1:1086F
 weighted histogram analysis method (WHAM) 2:553
 weighted majority voting 1:523
 Weighted Neighbor Distances-Compound Hierarchy of Algorithms Representing Morphology (WND-CHARM) 2:1003–1004
 weighted network analysis 2:1136–1137
 weighting schemes 2:1103
 weights update 1:627, 1:627–630
 Weizmann Institute of Science Crown Human Genome Center 2:104
 WekaOntology 1:333
 well-tempered metadynamics (WT metadynamics) 2:554
 Wellcome Trust Sanger Institute 2:197–198
 WES analysis pipeline 3:169
 West Nile virus (WNV) 3:246–247, 3:247T
 WF model selection 2:9–10
 WHAT_CHECK 3:523, 3:586, 3:588
 whisker diagram 1:675, 1:676F
 “white noise” conditions 1:722–723
 whole community analysis approaches 3:203–204
 whole exome sequencing (WES) 2:359, 3:113, 3:164, 3:165, 3:171, 3:176, 3:798–799
 whole genome bisulfite sequencing (WGBS) 3:324
 whole genome duplication (WGD) 2:261–262
 whole genome sequencing (WGS) 1:272, 2:251, 2:359, 2:1143, 3:113, 3:164, 3:165, 3:363, 3:798–799
 alignment and assembly 3:177–178
 annotation and analysis 3:179–180
 clinical workflow and application 3:180–181, 3:181F
 data preparation 3:176–177
 limitations 3:181–182
 typical computational pipeline 3:176, 3:177F
 variant calling 3:178–179
 whole human genome 1:147
 whole transcriptome 3:50–53
 whole-genome shotgun (WGS) 3:29–30
 whole-genomes 2:202–203
 “Widom 601” 2:310
 WikiPathways 2:799–800
 Wikispecies 2:1111–1112

Wilcoxon signed-rank test for location 1:664–666
 Wiley Registry of Mass Spectral Data 2:403
 WiredTiger 2:1072
 wireframe models 2:520
 within clusters sum of square (WCSS) 1:352, 1:451
 within print-tip group normalization 2:366
 within set sum of square error (WSSSE) 1:278–279
 within-class scatter matrix 1:489
 within-class scatter of projected samples 1:489
 WND-CHARM 2:1006
 Women's Health Initiative (WHI) 2:175
 Wood–Ljungdahl pathway 3:946
 Word Based Tagged Code (WBTC) 1:483
 word classes. *see* part-of-speech (POS)
 word sense disambiguation (WSD) 1:566
 Word Similarity Calculator class 1:884
 Wordind 1:820
 WordNet (WordNet) 1:589
 “work horses” in metabolomics 2:396
 workbenches 2:1153–1154
 Worker Nodes on Demand Services (WNoDes) 1:233
 Workflow Instance Generation and Specialization (WINGS) 2:1157
 workflow management systems (WMS). *see* pipeline frameworks
 Workflow4metabolomics 3.0 2:401
 workflows 2:1151
 challenges and opportunities for 2:1157–1158
 decay 2:1158
 development environments 1:247
 DMCF 1:249
 Swift 1:249
 in metabolomics 2:401–402
 sharing 2:1156
 World Health Organization (WHO) 3:247, 3:932, 3:996
 dengue diagnostics in WHO Western Pacific Region (2013) 3:11–13
 world wide web (WWW) 1:163
 connectivity 1:199–200
 World Wide Web Consortium (W3C) 1:288, 1:786, 1:800, 1:1104–1105
 WorldClim database 2:1134
 Worldwide Protein Data Bank (wwPDB) 2:460, 3:586

protein structure databases outside of 2:465
 websites of 2:462–463
 worm-like-chain model 2:292
 Wormbase 2:253, 2:254F
 worms (*Caenorhabditis elegans*) 2:328
 wrapper approaches. *see* model-based approaches
 wrapper methods. *see* model-based approaches
 wrapping layer 1:884
 Wright-Fisher model (WF model) 2:3–4
 alternatives 2:21–22
 Branching models 2:23–24
 Cannings model 2:22–23
 Moran model 2:21–22
 expected time to fixation 2:4–7
 probability of fixation 2:4
 write and execute Perl program
 by using IDE 1:177–178
 by using terminal 1:178, 1:178F
 Write Once, Run Anywhere (WORA) 1:206
 Wu & Palmer measure (W&P measure) 1:879

X

X-Cephalometric images 2:995–996
 X-linked agammaglobulinemia (XLA) 3:95
 x-ray 2:835–836
 diffractometry 2:46, 2:585
 scattering technique 2:465–466
 x-ray crystallography (XRC) 1:562, 2:143, 2:512, 2:512–514, 2:513F, 2:516, 2:585, 2:586, 3:6, 3:6–7, 3:10, 3:12, 3:252, 3:274, 3:521, 3:722
 analyses 2:30
 device 3:7
 x-ray free electron laser (XFEL) 2:514
 XCMS 2:399
 XCMS Online 2:401
 Xenbase 2:345
 XeonPhi computing 1:212
 XGBoost 1:302
 ximelagatran (Exanta) 2:585
 XIST lncRNA in epigenetic silencing 3:581
 XOR function 1:617–618
 XP. *see* x-ray crystallography (XRC)
 XSTREAM program 2:214

Xtlsstr algorithm 3:519

Y

YASARA view (2015) 2:533
 yeast 3:874, 3:946–947
 yeast one-hybrid assay (Y1H) 2:149
 Yeast Proteome Database (YPD) 1:1105
 yeast two-hybrid system (Y2H system) 1:919, 2:805, 2:824, 2:827, 2:834, 2:835, 2:836, 3:196, 3:283, 3:410, 3:933
 yellow fluorescent protein (YFP) 2:827
 Yet Another Learning Environment (YALE) 1:305
 Yet Another Resource Negotiator (YARN) 1:222
 Yin Yang 1 (YY1) 2:295
 Youden index 1:549
 Youden's *J* statistic 1:549
 Yule model 2:709

Z

“Z buffer” algorithms 2:528
 Z-DNA 3:505
 Z-score 3:253, 3:593
 normalization 1:467
 zanamivir (Relenza) 2:585, 3:275–276, 3:744T–748
 Zebra International Resource Center 2:104
 zebrafish (*Danio rerio*) 2:252, 3:103–104
 zebrafish models (ZF-Models) 2:104
 Zeng's approach 2:18
 Zernike moment (ZM) 2:999
 zero inflation 2:369
 zero mean normalization 1:478
 zero-one “loss function” 1:496
 Zif-Predict 3:634
 Zika virus 3:10
 Zinc 2:544, 2:1093–1094
 zinc finger (ZF) 3:633
 Zinc finger nucleases (ZFNs) 3:633
 zinc finger protein (Zif268) 2:447, 2:449F
 for specific binding to DNA sequences 3:633–634
 Zoo Blot 2:200–201
 Zuker algorithm 3:509–510
 zygote 3:825